Portadilla

Universidad Nacional Autónoma de México

Facultad de Contaduría y Administración

Aprendizaje de máquina para toma de decisiones en reclutamiento y selección, caso de la NFL

Tesis

Que para obtener el título de:

Licenciado en Administración

Presenta:

José Mauricio Mani Yáñez

Asesor:

Dr. Ricardo Alfredo Varela Juárez

Cd. Mx. **2020**

Universidad Nacional Autónoma de México
UNAM
Biblioteca Central
Dirección General de Bibliotecas de la UNAM

UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

Portada

Universidad Nacional Autónoma de México

Facultad de Contaduría y Administración

Aprendizaje de máquina para toma de decisiones en reclutamiento y selección, caso de la NFL

Tesis

José Mauricio Mani Yáñez

Cd. Mx **2020**

Contenido

Índice de figuras

Índice de cuadros

Agradecimientos

A mi mamá Elsa Gabriela y a mi padre Mauricio por apoyarme en todos mis emprendimientos, por sus consejos y enseñanzas. A mi hermano Antonio por ayudarme en lo que fuera necesario e intentar resolver mis dudas. A mi abuela Elsa por protegerme siempre y a mi abuelo Antonio por hacerme resiliente. A mis tías, tíos y primos por hacerme sonreír, estar cerca de mí y permitirme expresarme libremente.

A la Universidad Nacional Autónoma de México y a la Facultad de Contaduría y Administración por brindarme infinitas oportunidades y enseñarme a seguir aprendiendo siempre. A todos los profesores, por enseñarme que el conocimiento debe ser compartido y que hay que adoptar una postura crítica ante la vida. Y en especial al Dr. Ricardo por creer en este tema y estar siempre al pendiente de mi trabajo, guiándome a través de este proyecto.

Al Department of Management Studies de la Universidad Tecnológica de la India en Delhi Y en especial a Swati Garg por enseñarme la importancia de la investigación junto con la Dra. Shuchi, y el Dr. Arpan. Igualmente, al Dr. Vigneswara Ilavarasan por confiar en mí. Al departamento de movilidad de la Facultad de Contaduría y Administración por su apoyo incondicional.

A las plataformas de aprendizaje en línea, a los desarrolladores de los proyectos de licencia libre que fueron utilizados en este trabajo y a Fundación Carlos Slim por introducirme al camino de la cuarta revolución industrial y hacerme perder el miedo a la programación y enseñarme que siempre se pueden resolver las cosas con esfuerzo. Además, a todas aquellas personas que invierten su tiempo en compartir sus hallazgos y conocimientos en sitios de preguntas y respuestas, en foros y redes sociales.

A todos mis amigos por siempre escucharme, integrarme y aceptarme como soy, por tantas risas y aventuras.

A todos gracias por permitirme cada día ser mejor persona.

Introducción

A partir de los trabajos de Gary Becker aplicados a la economía, donde enfrenta problemas de la vida real entendidos desde su campo de estudio; surgen trabajos como *Freakonomics* de Steven Levitt y Stephen Dubner donde se intenta entender problemas sociales con teorías económicas. Aplicar a los problemas cotidianos aquello que se aprendió durante los estudios universitarios, pero no únicamente problemas complejos, sino nimiedades y curiosidades, que se pueden entender con la aplicación de lo estudiado; poniendo en marcha nuestra parte más crítica y curiosa. Es por ello que en este trabajo se propone resolver un problema del área de recursos humanos en selección de personal, mediante una aproximación *Big Data* al problema de reclutamiento masivo y de inteligencia artificial en la selección de candidatos aplicado a la NFL, debido a la cantidad de información que existe y su disponibilidad.

De acuerdo al existencialismo filosófico y sus estudios sobre la libertad, los seres humanos somos libres, por lo que no tenemos un destino, sino la serie de decisiones que tomamos son las que forjan nuestro futuro. Resulta ser lo mismo para una organización, y son, en sus etapas de planeación y organización donde se decide que es lo que se realizará durante un tiempo determinado. Posteriormente en su fase de ejecución se accionará en función de las decisiones tomadas en los dos pasos anteriores. Por ello la importancia de una adecuada planeación, ya que será la que defina los rumbos de la organización. Por lo anterior, la administración debe de buscar las mejores formas de tomar decisiones, porque inclusive el no tomar una decisión es la decisión en sí misma. Es por ello que en esta investigación se propone un modelo para ayudar a la toma de decisiones de reclutamiento y selección. Donde se busca resolver el manejo masivo de información de mariscales de campo profesionales y sus datos colegiales, además, prediciendo su desempeño tanto de forma continua como de forma dicotómica dada su trayectoria en el fútbol americano colegial.

En este trabajo se propone una aplicación bot-araña para extraer información estructurada y semiestructurada automáticamente de distintas páginas web con información tanto profesional como colegial, junto con los procesos necesarios de almacenamiento y transformación de datos. Se estudió el flujo de ciencia de datos para la creación de un producto. Empezando por el análisis exploratorio y el preprocesamiento de datos, para obtener un arreglo de entrada para las distintas técnicas de aprendizaje de máquina que se utilizaron. Se entrenó un modelo supervisado que funcionará como referencia tanto para las aproximaciones de regresión como

para las de clasificación, buscando que estos cumplan con los supuestos estadísticos, para los subsecuentes algoritmos regularizados, arboles de decisión y perceptrones multicapa. Buscando que estos no subajusten o sobreajusten los datos y por lo tanto puedan generalizar bien ante observaciones nunca antes vistas, por ello se realizó la división del arreglo en entrenamiento y prueba, para conocer el error práctico del modelo y compararlo con el error teórico. Además, los algoritmos de clasificación nos permiten personalizar la cantidad de error dispuestos a aceptar en distintas etiquetas. Debido a la naturaleza del problema se aplicó aprendizaje no supervisado de agrupación para balancear el arreglo de mariscales de campo sin desempeño esperado positivo. Al mismo tiempo, se incluyen las características que permiten que se puedan realizar predicciones al mercado laboral del fútbol americano profesional junto con lo que se considera son las características necesarias del modelo para ser aplicado a un equipo de la liga de fútbol americano profesional y se adapte a las características que requiere la industria.

La estructura capitular del trabajo inicia con un marco teórico en donde se introducen temas de recursos humanos, ciencia de datos y big data, a continuación, se realiza una revisión de la literatura de los temas del marco teórico, luego, se introduce al fútbol americano. Posteriormente, se detalla la creación del bot araña para la extracción de datos, su exploración, transformación y modelamiento. Luego, se muestran los resultados y por último la conclusión.

Planteamiento del problema

Con el crecimiento de la información nuevas oportunidades en los procesos de toma de decisiones generarán cambios vertiginosos en las herramientas organizacionales de la administración. Por lo anterior, tecnologías de aprendizaje de máquina pueden ser aplicadas a lo largo de las distintas áreas estructurales que conforman las organizaciones. Nuevas técnicas de aprendizaje de máquina son desarrolladas con el objetivo de mejorar las predicciones a los problemas a los que estas se enfrentan. De acuerdo con Aditi Jain (2016) las tendencias en las prácticas de administración de recursos humanos estarán relacionadas con control y medición de resultados, individualización, que permite que los sujetos que conforman las organizaciones sean tratados y atendidos individualmente gracias al análisis Big Data, que permite extraer conocimiento más personalizados de los individuos; la inteligencia artificial y la madurez de la analítica de personas son algunas de las tendencias de administración de recursos humanos.

Dentro de las áreas estudiadas por la administración de recursos humanos se encuentran el reclutamiento, selección y retención del personal, que ha sido analizado por McCracken, Currie & Harrison (2015) en procesos para recién graduados y administración de talentos, teniendo como objetivo encontrar a los más destacados. Nuevos retos en la automatización de los procesos de reclutamiento con el uso de Inteligencia Artificial (Gupta, Fernandes, & Jain, 2018). La importancia de políticas y procedimientos en el reclutamiento y selección del personal es estudiada por Uzair, Majeed & Shakeel (2017) en la industria financiera mediante cuestionarios. Ng & Sears (2017) buscan investigar prácticas de proceso empleo desde el enfoque de la problemática de género, sus alcances y su importancia. Por lo anterior, la mejora en la toma de decisiones en reclutamiento y selección será vital para una adecuada retención del personal y gestión de talento.

En el presente trabajo se pretende desarrollar un modelo para la automatización de reclutamiento y semi-automatización de selección de personal, mediante la predicción de desempeño por medio de flujos de ciencia de datos y algoritmos de inteligencia artificial, que sirvan para la toma de decisiones en la selección del aspirante, tomando en cuenta información Big Data disponible del individuo a reclutar. Para mostrar la capacidad predictiva en los procesos de reclutamiento, el presente trabajo pretende predecir el desempeño de jugadores

en la NFL[1], dada la información del rendimiento en el fútbol americano colegial. Se ha elegido la aplicación al caso de reclutamiento y selección en la NFL ya que permite obtener y analizar grandes volúmenes de información sobre la actuación pasada de un jugador y con esta información poder predecir su actuación esperada en su futuro profesional.

Objetivo General

Proponer un modelo como una alternativa a las técnicas y métodos empleados actualmente usados en el campo del conocimiento de los recursos humanos en la función de administración del reclutamiento y selección del personal. Describiendo como el uso de herramientas Big Data, algoritmos inteligentes y flujos de ciencia de datos pueden ser aplicados para tomar decisiones para el caso de la NFL. Estudiando la identificación de los mariscales de campo más aptos para la liga profesional mediante los datos de su trayectoria en el fútbol americano colegial.

Además, buscamos que este trabajo sea útil e inspirador para todos aquellos interesados en el estudio de la toma de decisiones en las organizaciones, la administración Big Data, los flujos de ciencia de datos y la inteligencia artificial. Haciendo énfasis en la importancia de iniciar la discusión respecto a la automatización del trabajo en escuelas de administración.

Objetivos Específicos

- Definir procedimientos de ciencia de datos que logren mejores ajustes en algoritmos inteligentes, para optimizar la toma de decisiones en la administración de reclutamiento y selección de personal.
- Exponer los usos de Big Data mediante la recopilación masiva de datos para reclutamiento y el uso de predicciones como resultado de algoritmos con inteligencia computacional para la toma de decisiones de selección de personal en las organizaciones.
- Desarrollar un modelo de predicción de desempeño esperado para el caso de reclutamiento y selección de jugadores en la NFL, usando su trayectoria de juego en el fútbol americano colegial de todos los quarterbacks que aspiraban a ser profesionales.

[1] Del inglés, *National Football League*: Liga Nacional de Fútbol.

Hipótesis

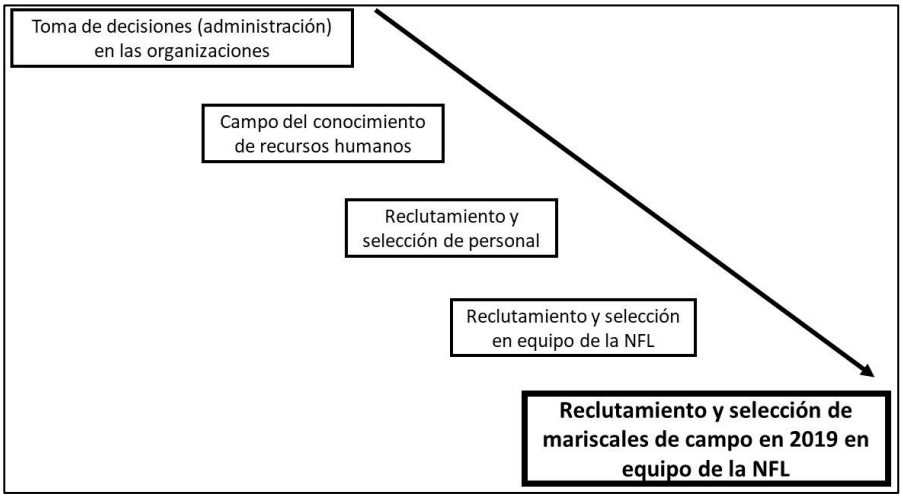
H_1 El aprendizaje de máquina permite mejorar las decisiones sobre el reclutamiento y selección de personal basándose en algoritmos.

H_2 La estructura que ofrece la liga profesional de fútbol americano y el empleo de técnicas big data para impulsar algoritmos machine learning, resulta conveniente la utilización de información colegial (a priori) para predecir el desempeño esperado de mariscales de campo entre las temporadas de 1999 y 2019.

Metodología

La intención del trabajo es integrar distintos temas tomando en cuenta los intereses personales del investigador que son el fútbol americano y la NFL. Además de temas científicos de interés personal como metodologías de ciencia de datos y algoritmos de inteligencia computacional para resolver problemas contemporáneos de administración, en este caso el reclutamiento y selección de personal. El problema inicia con la explosión de datos en todas las áreas de la administración y los recursos humanos no son la excepción. De acuerdo con Marga Salvador y Poyen Ramos (2016), la empresa Google recibe alrededor de 1,000,000 de currículos al año, contratando solo alrededor de 0.05 por ciento de ellos. La cantidad impresionante de aspirantes se debe, en parte, a su reputación como gran empresa para trabajar y ampliamente poderosa. Lo anterior nos permite reflexionar acerca de la cantidad de datos a los que se tiene acceso con una publicación de oferta de trabajo en la red, pero mediante procesos activos con el uso de redes sociales laborales[2]. Donde se vuele factible explorar un número muy cercano a toda la oferta de candidatos, pero surgen problemas ya que no es solo atraer, sino que

Figura 0.1: Delimitación del trabajo.



Fuente: Elaboración propia.

[2] Ver apartado 1.1.3. Administración de Recursos Humanos.

también es importante conocer y elegir buenos candidatos. Para el fútbol americano, lo anterior es vital, ya que durante el Draft[3] se invierten muchos recursos por parte de todos los equipos para lograr una plantilla de jugadores talentosos y por lo tanto un equipo exitoso[4]. Este proceso inclusive es televisado.

La delimitación del trabajo se muestra en la Figura 0.1 partiendo de la investigación a nivel general de la toma de decisiones en las organizaciones, pasando por el área de recursos humanos y su función de reclutamiento y selección. Se pretende estudiar el caso específico de las organizaciones de la NFL (equipos) para encontrar y elegir mejores jugadores. Al ser posiciones especializadas se ha decidido investigar el proceso de toma de decisiones de reclutamiento y selección de mariscales de campo, demostrando sus resultados en la temporada 2019 en organizaciones de la NFL. A pesar de que se tomará en cuenta la temporada 2019 para la validación de resultados, usaré los últimos 20 años de datos para predecir el desempeño de la NFL en el año 2019.

La Metodología de investigación es realizada conforme a los trabajos de Hernández Sampieri, Fernández Collado & Baptista Lucio (2014) de acuerdo con estos, el enfoque cuantitativo es un conjunto de procesos donde los pasos y su orden debe de ser riguroso. Por ello el enfoque propuesto es el cuantitativo, haciendo hincapié en lo escrito por Creswell (2013) y Niglas (2010) citado por Sampieri, Collado y Lucio (2014) que no se debe de realizar una investigación con un enfoque binario, pero entender que todo problema se puede estudiar en un espectro entre la rigurosidad cuantitativa y la libertad creativa que ofrece el enfoque cualitativo. El tipo o alcance de la investigación será correlacional, iniciando con una amplia descripción del fenómeno de reclutamiento y selección bajo el paradigma Big Data y los métodos que este propone, como "*machine learning*" y ciencia de datos. Para dar respuesta a la pregunta sobre el efecto que tienen las variables a priori (colegiales, antes de profesional) para predecir el desempeño en la liga profesional de fútbol americano. Que logre proponer una solución para la predicción del desempeño de los jugadores en sus años como novato en la liga profesional de fútbol americano tomando en consideración variables que creemos afectan el desempeño esperado, dados los datos de su paso por el fútbol americano colegial.

[3] Proceso de reclutamiento y selección de la NFL. Ver el apartado de Draft de la sección 4.4. Fútbol Americano Profesional
[4] Ver apartado de Draft de la sección 4.4. Fútbol Americano Profesional

La población a usar para comprobar la hipótesis son mariscales de campo profesionales de todos los equipos que conforman la organización de la NFL de los últimos 20 años, debido al cambio de juego 1980-2000, consecuencia de la evolución de las posiciones[5]. La recolección de datos utilizará métodos Big Data, obteniendo la información necesaria mediante un "*crawler*" a un sitio web de estadísticas de fútbol americano (www.sports-reference.com), tanto colegial como profesional escrito en el lenguaje de programación orientado a objetos Python. El procesamiento de datos se realizará en *Python Jupyter Notebooks* y serán parte de la descripción de flujos de ciencia de datos y algoritmos inteligentes.

[5] Ver apartado de Posiciones de la sección 4.4. Fútbol Americano Profesional.

1 Marco Teórico

En este capítulo se conceptualizará en el campo del conocimiento de los recursos humanos en reclutamiento y selección. Asimismo, se explicarán nociones de Big Data y flujos de ciencia de datos (estrategias para resolución de problemas), así como enlistar y definir algoritmos de inteligencia computacional y las bases teóricas de lo anterior. Se hará una revisión de la literatura en aproximaciones de inteligencia artificial en la toma de decisiones para reclutamiento y selección en el campo de conocimiento de los recursos humanos. Además, se hará una descripción del fútbol americano de la liga profesional de la NFL y la liga colegial de fútbol americano, sus procesos de reclutamiento y selección y la estructura organizacional.

1.1 Recursos humanos y la incorporación de talento

En esta capitulo se estudiará el concepto de administración para el mejor entendimiento de la administración de RH como área funcional de las organizaciones que servirá como introducción a los procesos para incorporar personas, buscando dar un panorama de los objetivos, actividades, características y retos del área funcional de recursos humanos y del reclutamiento y selección del personal.

1.1.1 Definiciones de administración

Stephen Robbins y David deCenzo (2002) aseguran que la administración se encarga de conseguir que se lleve a cabo lo planeado de manera correcta, como fue definido en un principio (eficacia) y maximizando la relación insumo-producto (eficiencia) mediante un proceso realizado por aquel o aquella que sustenta el cargo organizacional de gerente. Este proceso se conoce como proceso administrativo, que consta de cuatro actividades, planificar, organizar, dirigir y controlar. Idalberto Chiavenato (2019) agrega la necesidad del administrador de una visión crítica e innovadora. Y estudia distintos conceptos de administración, donde se destaca la importancia de la consecución de los objetivos y la utilización de recursos: humanos, financieros, materiales y tecnológicos a través de otros en una organización, mediante un proceso como el de planeación, organización, liderazgo (dirección) y control. Y enfatiza la importancia de la toma de decisiones, la acción y la coordinación, sin embargo, no incluye la gestión. Los estudios de los autores anteriores fueron traducidos al español, escritos en un contexto de organizaciones de países desarrollados y conflictos de traducción (Sanabria, 2007). Sergio Hernández y Rodríguez & Alejandro Pulido (2011) ambos hispanoparlantes completan su estudio conceptual de la administración mediante el concepto de gestión y de

gerente. La primera ayuda a los que la usan al logro de lo anhelado, guiado por las estrategias fijadas previamente, mediante un proceso crítico, creativo, de análisis y reflexión. Sin embargo, se basan en la definición de administración usando el marco conceptual internacional como el de los primeros autores.

1.1.2 Administración y su relación con la toma de decisiones

Omar Guerrero (2004) en un análisis de administración pública examina la palabra "management" a través de su traducción al español. Management proviene de manipular o manejar (managery) y posteriormente se operó como un sinónimo de administración o de gerencia. El uso de "public administration" y "public management" se han usado de forma indistinta a lo largo de la historia, pero diferenciando su uso en Europa y Norte América, siendo administración en la primera lo que se está ejecutando y en la segunda un estado jerárquico en la organización; mientras el management en la primera se entendía como manipulación y en la segunda un concepto confiable y dinámico. Además, de acuerdo con Sanabria, en Europa se ligaba el management con organizaciones con fines de lucro y a la administración con lo público. Por último, asegura que management apela a la dirección o conducción de una organización por un "manager", que trabaja especialmente en la decisión. En una investigación (Lara, 2015 citado en Chávez García, Arguello Pazmiño, Viscarra Armijos, Aro Sosa, & Albarrasín Reinoso, 2018, p. 8) declara que "cuando se estudia la gerencia se estudia la toma de decisiones". Dentro de las teorías enlistadas por Chiavenato (p. 17), la teoría de las matemáticas y la tecnología en la administración, enfoca su estudio en la decisión priorizando esta sobre la acción; siendo el administrador aquel que toma las decisiones en la organización para la resolución de los problemas que afecta a la misma. No obstante, la aproximación matemática sintetiza el problema mediante el uso de números y variables perdiendo de vista el enfoque global, por lo que en ocasiones solo se usa para ciertas actividades. En la obra Comportamiento Administrativo, Herbert Simon (1964) asegura que el proceso administrativo es un proceso decisorio que posteriormente será comunicados a aquellos colaboradores afectados, siendo la actividad de dirección el empleo de la autoridad administrativa, donde el trabajador no tiene capacidad de elección. Por ello la administración se encarga de solucionar (mediante una adecuada toma de decisiones) y hacer que se lleven a cabo (actuar); la primera por el administrador, la segunda en conjunto con el colaborador o trabajador. En la coyuntura actual, la teoría clásica y científica (Chiavenato, p. 17) estudiada por Simon deberá ser

replanteada por la automatización de ciertas posiciones y el remplazo por robots en algunas industrias (Puntoni, 2019 & Connley, 2017; Cerullo, 2020). Al no haber colaborador al que coordinar, gestionar, manejar o dirigir la administración se centrará en tomar las mejores decisiones.

1.1.3 Administración de recursos humanos

Gary Dessler y Ricardo Varela (2017) definen la administración de recursos humanos de acuerdo a las actividades relacionadas con las personas y aquellos procesos que son atribuibles al área funcional de recursos humanos. Siendo las actividades más importantes el contratar, capacitar, evaluar y remunerar empleados. Llevar acabo las actividades de RH es importante para las organizaciones capitalistas ya que permite realizar las operaciones de manera más eficiente y eficaz, buscando la racionalidad en el puesto de trabajo teniendo como máxima elevar la productividad enfatizando en la importancia de contratar al personal idóneo. Por ello, en las áreas de Recursos Humanos existe un puesto de reclutador, encargado de buscar a los mejores candidatos para los puestos que requiere la organización. Scott Snell y George Bohlander (2013) hacen énfasis en las distintas maneras en las que se llama a la gestión del personal en las distintas organizaciones: recursos humanos, capital humano, activos intelectuales y gestión de talento. Todas las anteriores tienen un solo objetivo ser competitivo usando el talento contratado por la organización, el capital humano que debe de buscar alcanzar las metas y objetivos generales de la organización. Dessler et al. Incluyen en su estudio la administración de recursos humanos en las pequeñas empresas, siendo las actividades de la primera importantes para una sana consecución de los objetivos. Dando espacio a imaginar las funciones de RH fuera de la burocratización debido a la cantidad de empleados en las pymes, permitiendo a los encargados de las actividades de RH estar más cerca del personal. Recordando que por las características de dichas organizaciones es complicado seguir las tendencias y afrontar los retos constantes, como puede ser la adopción de la tecnología o la globalización de los procesos. Snell y Bohlander aseguran que son diversos los retos que el área de recursos humanos debe afrontar para mantenerse competitiva:

- Los cambios en el negocio y mercado a mediano y largo plazo rumbo a la adopción de mejores prácticas organizacionales y administrativas, debido a los desafíos demográficos y la globalización.

- Procesos enormes de reclutamiento y selección debido a la coyuntura actual de globalización, que repercute en búsqueda de mejores prácticas.
- Uso de la tecnología en toda la organización, tanto para la administración de las actividades inherentes del área de recursos humanos[6] para mejores prácticas laborales.
- Minimización de costos y maximización de la productividad a través del mejor talento.

De acuerdo con Varela y Lerma (2016), las tareas del gerente del área de talento deben ser vistas de manera holística y no como una función separada, de esta manera los objetivos organizacionales estarán ligados con las acciones emprendidas en cada área y se podrán entender como procesos lineales, interrelacionados. Por lo tanto, si no se alcanzan los objetivos de la primera función, no permitirá un adecuado flujo para las funciones que le siguen. Si no se realiza un adecuado reclutamiento, la selección de candidatos no será adecuada y repercutirá en el desempeño de este o esta con la organización. Las funciones del área deben ser integradas, por cada uno de los gerentes que conforman el área de gestión de talento, de esta manera se accederá a un panorama amplio de la alineación de objetivos, perfiles y tareas que se necesitan desarrollar para alcanzar los objetivos fijados. Por ejemplo, una correcta gestión del área de talento, se encargará de segmentar a los empleados clave para la consecución de la función organizacional, para ello se deben de establecer métricas claras, conocidas por todos, generalizables y replicables. Por ejemplo, un empleado clave debe ser hábil en la comunicación y dispersión de conocimiento, este perfil lo deben de cumplir todos los empleados clave y deberá ser medido de la misma manera. Pero empieza desde la necesidad de empleados clave o necesidad de contratación, luego, es necesario que en el proceso de reclutamiento se busquen candidatos con habilidades comunicacionales usando las mismas métricas de toda el área; después, en la selección se deberá estar seguro que el candidato cumple con los mínimos requerimientos para que sea contratado y posteriormente pueda ser evaluado, toda el área bajo un mismo marco de trabajo.

1.1.4 Reclutamiento y selección

El talento debe de ser identificado buscando gran precisión en el momento adecuado y esto se puede lograr gracias a un reclutamiento y selección de personal apropiada. Para explicar el

[6] Sistemas de información de Recursos Humanos, HRIS. Del inglés, Human Resource Information System

reclutamiento y selección, nos basaremos en lo publicado por William Werther, Keith Davis y Martha Guzmán (2014). El reclutamiento puede ser tanto interno (en la organización) como externo, esta decisión limitará el alcance de búsqueda y se deberá tomar en cuenta los recursos financieros, la complicación, el tiempo que tomará, las políticas organizacionales, requisitos del puesto, etc. Las políticas de compensación son aquellas que establecerán los parámetros de acción en términos de sueldos y salarios. Las políticas de contratación establecerán los alcances de la acción en la contratación de los aspirantes. Los canales de reclutamiento externo son:

- Sitio web de la organización a contratar
- Sitios de reclutamiento en la web
- Agencias de empleo
- "*Headhunters*"[7]
- Instituciones educativas
- Asociaciones profesionales
- Ferias de empleo

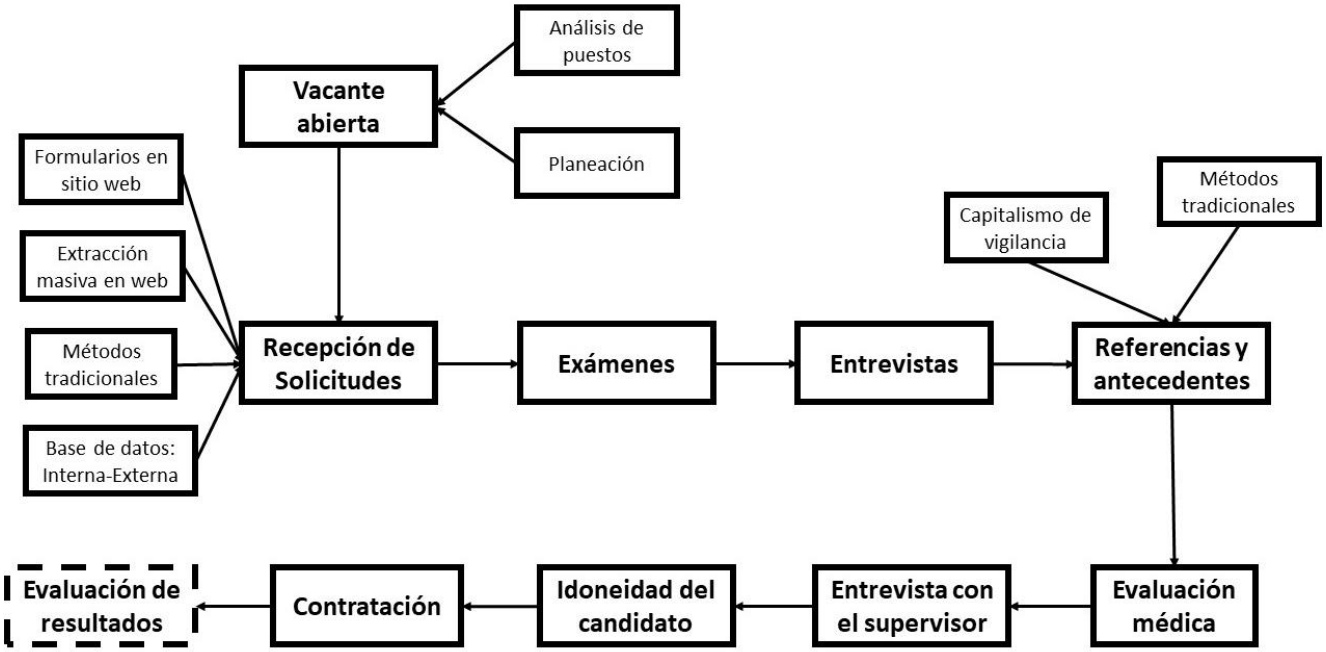
En el proceso de reclutamiento se recaban datos del solicitante como su situación laboral, preparación académica, antecedentes laborales, pasatiempos y personales. Estos datos pueden ser recabados en sitios de reclutamiento web de forma masiva sumando datos de habilidades y experiencia para la identificación de talento (Bastian, y otros, 2014; Lops, de Gemmis, Semeraro, Narducci, & Musto, 2011; Garg, Rani, & Miglani, 2015) o formularios en el sitio web de la organización donde se recauda información de aquellos candidatos interesados, incluyendo las ligas de referencia a la cuenta del sitio de reclutamiento web del interesado (Paz & Reina, 2004).

Werther et al. sostienen que el objetivo de la selección es detectar, desarrollar y retener talento. La actividad de detección de talento puede ser tercerizada inclusive a nivel internacional, donde el desarrollo y la retención también puede ser externalizada; sin embargo, estas son tareas específicas. Para una correcta detección del personal adecuado es necesario tener análisis de puestos, para cumplir con la rápida atención de las demandas de vacantes, con

[7] Se traduce como cazadores de talento.

aquellos candidatos que se ajustan mejor a lo detallado en el análisis de puestos y con capacidad de desarrollo y retención potencial de los aspirantes.

Figura 1.1: Proceso de reclutamiento y selección.



Fuente: Elaboración propia.

Como lo muestra la figura 1.1, el primer paso, una vez que se tiene conocimiento de la vacante abierta y se ha planeado el reclutamiento (objetivos, estrategias, políticas, programas, procesos, canales) y se conocen las necesidades del puesto (análisis del puesto) se recibirán las solicitudes por el canal elegido. La segunda fase tiene por objeto asegurar que el candidato tiene las competencias necesarias, que será vital para una futura selección del candidato:

- Conocimiento: Información teórica que posee el candidato.
- Habilidad: Es la experiencia que le da la capacidad al individuo de resolver y hacer una tarea.
- Actitud: Características del individuo que lo mueven a realizar una acción.

Se deben de encontrar medios adecuados para detectar las competencias de los solicitantes, como pruebas de inteligencias tratando de lograr la mayor objetividad posible. En la entrevista se debe saber si el entrevistado podrá desarrollar las funciones y actividades detalladas en el análisis de puesto y mantener la moderación para comparar con los candidatos pasados y los

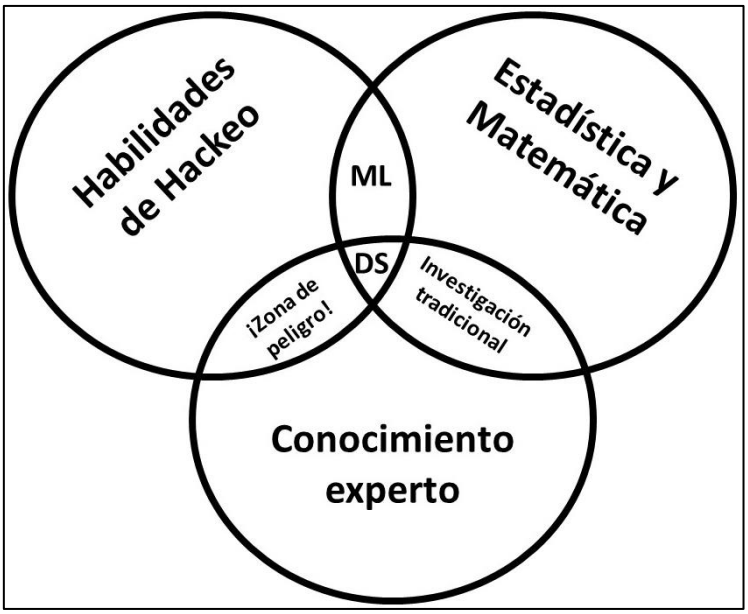
que vienen. En las entrevistas también se puede conocer si el candidato posee con las competencias necesarias para el puesto. Posteriormente se debe comprobar la integridad de los datos, mediante llamadas telefónicas o contactos en la industria, pero también con medios de capitalismo de vigilancia y almacenamiento masivo de datos personales (Sherman, 2019). El proceso de reclutamiento y selección requiere de integridad en la información y de un protocolo ético y social. Posteriormente se realizará una entrevista con el superior jerárquico inmediato de la vacante donde se examinarán los habilidad, experiencia y conocimientos técnicos del candidato. Consecutivamente se analizará la idoneidad del candidato sintetizando y analizando la información de cada una de las fases del proceso, priorizando aquellos con experiencia, habilidad, conocimiento y actitud, esta última ayudará a maximizar la retención del candidato y su desarrollo profesional una vez que allá sido contratado. Dessler y Varela hacen énfasis en la importancia de saber sobre candidatos tanto en la misma organización, como fuera de ella que puedan desempeñar la labor detallada en el análisis de puestos. Además de la importancia de dar seguimiento a la contratación, mediante una inducción y capacitación constante para un adecuado desarrollo, pudiendo medir, en un tiempo previamente determinado, los resultados del candidato seleccionado. De esta manera se pueden tomar decisiones adecuadas en otras áreas, como aumento salariales o crecimiento en la escala jerárquica.

1.2 Ciencia de datos e inteligencia artificial

En esta sección se estudiará la ciencia de datos y sus procesos para entender las distintas actividades, herramientas y conceptos que la componen: Big Data como motor de metodología de ciencia de datos, flujos habilitadores de algoritmos de inteligencia computacional para extracción y preprocesamiento de datos e inteligencia computacional para extracción de conocimiento en arreglos masivos de datos. Para un claro entendimiento de estos, se establecerán las bases teóricas de los medios usados en el flujo de ciencia de datos. Se entenderá *Machine Learning* por inteligencia computacional, inteligencia de máquina o mediante el acrónimo ML y ciencia de datos por su acrónimo DS[8]. Por API[9] se entenderá interfaz de programación de aplicaciones. Se usará de manera indistinta variables (estadística clásica) y atributos (estadística computacional).

[8] Del inglés, "*Data Science*".
[9] Del inglés, "*Application Program Interface*".

Figura 1.2: Diagrama de Venn de ciencia de datos.



Fuente: Conway, D. (30 de septiembre de 2010). *The Data Science Venn Diagram*.

De acuerdo con John Kelleher & Brendan Tierney (2018) minería de datos, *machine learning* y ciencia de datos han sido usados de manera indistinta, sin embargo, cada uno de los conceptos se adapta a la resolución de problemas específicos y se han moldeado a través del tiempo. La ciencia de datos es constituida por una serie de actividades desafiantes como la extracción, limpieza y transformación de datos que incluyen el uso de tecnologías Big Data para la predicción o extracción de patrones mediante algoritmos de inteligencia computacional para una rápida acción y revelación de información. Con enfoque en manipulación de arreglos de datos extensos y algoritmos computacionales complejos. Se compone de 2 tareas principales, recopilar datos y analizar datos de forma descriptiva y predictiva o prescriptiva. Estas actividades requerirán de habilidades de hackeo, matemáticas y estadística y conocimiento experto (Conway, 2010) mediante un diagrama de Venn donde cada habilidad es un conjunto y la ciencia de datos es la intersección de estos tres conjuntos. Kelleher y Tierney detallan las habilidades, añadiendo la comunicación como una habilidad para detallar los procesos y resultados obtenidos; visualización de datos, que se encuentra relacionado con la comunicación ya que nos ayudará a expresar nuestras ideas de manera más clara; transformación de datos crudos a conocimiento mediante el manejo de bases de datos, además permitirá ajustar los datos a algoritmos computacionales, que junto con el almacenamiento de datos debe de regularse y cuestionarse de manera ética.

Son tres áreas principales las que permiten un adecuado flujo de ciencia de datos, tomando en consideración si estas áreas incluyen tecnologías Big Data o tradicionales:

- Fuentes de datos
- Almacenamiento de datos

- Aplicaciones

Las tecnologías tradicionales extraen información de bases de datos o aplicaciones con estructuras de datos bien definidas, para almacenar y analizar datos mediante almacenes de datos y sistemas administradores de bases de datos relacionales para aplicaciones de analítica de negocio y específicas para el negocio. Las tecnologías Big Data extraen datos de fuentes de internet de las cosas (IoT), de la web, etc. El almacenamiento y análisis se realiza en sistemas especializados como *Hadoop* para ser aplicado en procesos *Back-End* o en aplicación.

1.2.1 Almacenamiento de datos y big data

Para tomar mejores decisiones es necesario tener datos, es por ello que el almacenamiento y administración de los mismos es vital para una adecuada administración de las organizaciones. En el principio los sistemas de base de datos eran realizados para organizaciones buscando que fueran eficientes, a prueba de errores e imprevistos, con acceso a múltiples usuarios y constantes en el tiempo. Las bases de datos relacionales son aproximadas mediante modelos de datos, que buscan abstraer la realidad operacional de los datos. Donde se deben representar las relaciones mediante tablas de datos estructuradas. Además, es necesario poder consultar los datos de los administradores de base de datos relacionales mediante un lenguaje de consultas de alto nivel, este tiene como fin hacer accesible el acceso a los datos. Al lenguaje estructurado de consultas para base de datos se conoce como SQL [10]. Posteriormente se agregan al desarrollo las teorías de normalización para evitar la redundancia en datos ya que el problema enfrentado era el espacio de almacenamiento y construcción de índices para consultas rápidas (Silberschatz, Stonebraker, & Ullman, 1991). Kelleher y Tierney definen los almacenes de datos como zonas de datos que tienen como objetivo técnicas de agregación de datos para la toma de decisiones y requieren de un proceso ETL[11] que mueve datos entre distintas bases de datos. A estas operaciones se le conocen como OLAP[12]: Procesamiento Analítico en línea que tienen como objetivo hacer más rápidas las consultas ya que la descarga de datos es menor, puesto que solo se tienen que descargar datos agregados y previamente transformados.

[10] Del inglés, *Structured Query Language*
[11] Del inglés, *Extract, Transform, Load*: Extracción, transformación y carga.
[12] Del inglés, *Online Analytical Processing*

La era Big Data surge debido al crecimiento masivo y la democratización de los datos y para dar nombre a los arreglos de datos masivos que no podían ser procesados por una hoja de cálculo. Poder procesar datos en herramientas tradicionales de procesamiento varía en el dispositivo que se ponga a prueba, por lo que definir un arreglo de datos como Big Data resulta complicado. Son otros sistemas los que deben manejar estos datos, ya que la forma de conseguirlos, almacenarlos y administrarlos debe de ser distinta a la tradicional (Chen, Mao, Zhang, & Leung, 2014). De acuerdo con García et al. (2018) no son solo los sistemas o el tamaño de los datos, pero también la variedad, veracidad y valor. Estas junto con la velocidad y la variedad son conocida como las 5V's de Big Data.. La velocidad en Big Data está relacionada con la capacidad de analizar datos en tiempo real, así como procesar grandes volúmenes de datos. El volumen concierne a la capacidad de almacenar grandes cantidades de datos de distintas fuentes. La variedad son las distintas fuentes de las que proviene la data; por ejemplo, de redes sociales, de dispositivos de internet de las cosas o datos históricos. La veracidad es vital para sacar el mejor provecho de los datos. Los datos administrados deben de añadir valor y ser un medio para alcanzar los objetivos fijados.

De acuerdo con Keith Gordon (2014) debido a la incapacidad por parte de las bases de datos relacionales de escalar y de almacenar datos semiestructurados. Para resolver este problema modelos alternativos de base de datos conocidos como NoSQL. Algunas categorías son:

- Bases de datos clave-valor: Carecen de un esquema definido como en los modelos relacionales (entidad relación).
- Almacenamiento de documentos: Almacena documentos, por ejemplo, de texto.
- Almacenamiento de columna ancha: No es necesario que los datos sean definidos con anterioridad y contienen una mezcla de atributos.
- Base de datos orientada a grafos: Es construida mediante conexiones de elementos para representar relaciones sociales principalmente.

Chen et al. Delimita la tecnología usada en la administración Big Data, siendo *Hadoop* una solución de almacenamiento y procesamiento de datos construida mediante una infraestructura de módulos. *Hadoop* consiste de un sistema de archivos distribuidos y un *Framework MapReduce*. Los datos se distribuyen en nodos en un clúster (computadora) que dan paso al cómputo en paralelo. Estas tecnologías permiten el uso de consultas SQL mediante módulos como "*Hive*". *Hadoop* permite escalar en datos y ser muy flexibles para

organizaciones que requieren de tecnologías Big Data, pero no son gigantes de datos como Google o Facebook. Debe ser capaz de soportar errores ya que debe de administrar múltiples nodos que pueden congestionarse.

Tecnologías de cómputo en la nube pueden mejorar la administración de los datos ya que la infraestructura de hardware es tercerizada, reduciendo costos y aprovechando el conocimiento experto. En la era Big Data el volumen necesario de almacenamiento y procesamiento crece constantemente por lo que es necesario dar solución a los problemas constantes que surgen y ser capaces de extraer y dar el mejor uso a los nuevos datos que emergen que no son capaces de ser analizados con las tecnologías convencionales. Por ello la necesidad de individuos en las organizaciones que sean capaces de aprovechar mejor estos datos mediante tecnologías como Hadoop.

1.2.2 Flujos de trabajo en ciencia de datos

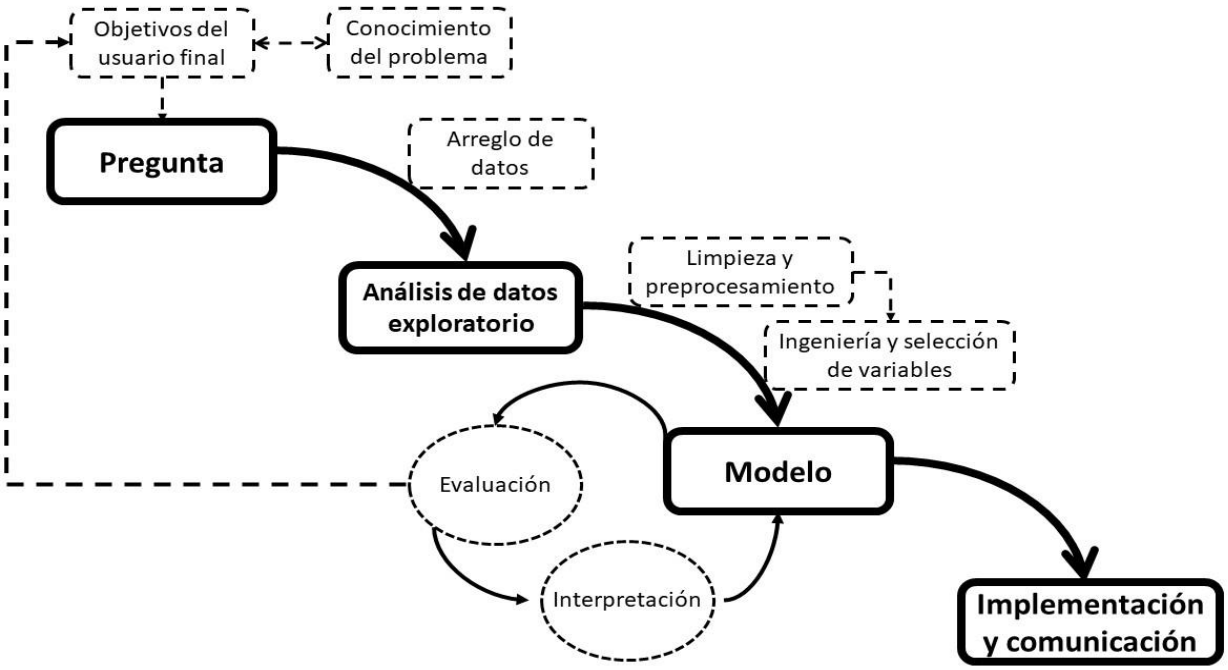
En este apartado se estudiarán los procesos de ciencia de datos de distintos autores para la puesta en producción de productos de datos que sean capaces de agregar valor a las organizaciones. Se detallarán las actividades que componen el flujo y algunas de las técnicas y métodos con que cuentan estas actividades. Por último, se pondrá como ejemplo una herramienta para flujos de ciencia de datos Big Data de licencia libre.

Con la creciente aparición de datos de manera exponencial, aplicaciones tecnológicas basadas en datos son posibles y por su naturaleza se pueden considerar como productos de datos. La ciencia de datos permite la creación de productos de datos que además sean capaces de contar historias (Loukides, 2010), a través de los datos mediante un proceso: El ciclo de ciencia de datos. De acuerdo con Roger Peng y Elizabeth Matsui (2015) el ciclo de ciencia de datos inicia con un proceso iterativo para encontrar una pregunta adecuada capaz de ser resuelta mediante datos como lo muestra la figura 1.3.

Por la explosión Big Data más preguntas pueden ser resueltas mediante datos, por lo que es posible hacer cada vez preguntas más interesantes y únicas, por ello es importante investigar si la pregunta que nos hacemos ya ha sido respondida; en caso de que ya exista respuesta a la pregunta esta primera iteración nos servirá de inspiración para preguntas posteriores. El análisis exploratorio de datos es el proceso mediante el cual se conoce el arreglo de datos usando la descripción de sus variables y como interactúan unas con otras. Este proceso es

intensivo en visualización mediante la representación gráfica de los datos y servirá para estar seguro de que el arreglo de datos con que contamos sirve para responder la pregunta. El siguiente paso es el uso de modelos, los cuales explican las relaciones que existen entre diversos factores para predecir un suceso mediante el uso de la estadística. El modelamiento tendrá por objetivo ayudarnos a responder la pregunta sabiendo que tenemos la data adecuada ya sea mediante inferencias o predicciones. La primera intenta estimar la relación entre variables de interés x, y; la segunda tiene como objetivo encontrar el mejor modelo en términos de predicción. Le sigue la interpretación, en esta fase debemos asegurarnos que hayamos respondido la pregunta y cómo es que la respondimos. La última fase incluye la tarea de comunicar el flujo de ciencia de datos llevado a cabo de la forma más clara de acuerdo al público objetivo.

Figura 1.3: Flujo de ciencia de datos.



Fuente: Elaboración propia.

García et al. Definen el flujo de ciencia de datos mediante etapas Big Data. En la primera etapa se debe definir los objetivos del usuario final. Posteriormente se debe crear una colección de datos y explorarla. Le sigue la limpieza y preprocesamiento de datos, ingeniería de variables y reducción de datos, el modelamiento y, por último, la interpretación. El modelamiento consiste de una clara idea de la aproximación al problema del modelo, se debe de conocer los distintos

algoritmos que se ajustan a la aproximación elegida y entrenar el modelo. Foster Provost y Tom Fawcett (2013) utilizan el proceso de minería de datos dentro del flujo necesario de ciencia de datos. Primero se debe conocer el negocio para poder proponer una solución mediante un proceso iterativo. Dicha solución incluirá datos, por lo que es necesario entender estos para saber cómo prepararlos para la fase de modelamiento. La fase de evaluación permitirá poner en producción[13] el mejor modelo mediante un proceso iterativo para la selección del modelo. En este proceso iterativo se asegurará un desempeño adecuado del modelo y una congruencia con las especificaciones de la solución definida por el negocio. La puesta en producción es la fase final que busca obtener ganancias del modelo desarrollado, por ejemplo, la predicción en tiempo real de ingresos no autorizados a un sistema informático. La implementación del modelo tiene por objetivo automatizar todo el proceso desde la obtención de datos hasta el producto final.

De acuerdo con Salvador García, Julian Luengo y Francisco Herrera (2015) la preparación de datos debe de asegurar datos de calidad y permitir que estos sirvan como entrada (*input*) del modelo, en ocasiones se necesitará integrar distintos arreglos de datos, se deberá limpiar, normalizar, transformar y reducir el conjunto de datos. Al integrar datos se debe asegurar que estos no sean redundantes. La redundancia se puede detectar mediante pruebas de correlación o la prueba Ji-cuadrado de Pearson. La limpieza de datos se encarga de manejar los valores faltantes en el conjunto de datos. Es vital conocer la naturaleza de los valores nulos como nulo aleatorio, nulo completamente aleatorio y nulo no aleatorio, para poder aplicar los mejores métodos de imputación. La primera aproximación es no imputar los datos, también, si los algoritmos *Machine Learning* a usar lo permiten, imputar datos mediante algoritmos de inteligencia computacional como vecinos más cercanos o K-medias. Además, Wes Mckinney (2011) con la librería de licencia libre Pandas de Python permite imputar datos usando el valor cero, con la media, la mediana, la moda o un valor indicado por conocimiento experto; permite la imputación por el valor anterior o el valor posterior al nulo, permite la eliminación de observaciones con valores nulos o columnas con nulos.

García et al. En su estudio de preprocesamiento para mejorar la capacidad predictiva de modelos de inteligencia computacional se propone normalizar los atributos, mediante

[13] Comúnmente usada la palabra del inglés, deployment.

funciones matemáticas. Consideremos el vector $x = [x_1, x_2, \dots, x_n]$ de tamaño n. Definamos el algoritmo para normalizar datos MIN-MAX para cada $a_i,\ i = 1$ hasta n como el vector auxiliar:

$$a_i = \frac{x_i - \min(x)}{\max(x) - \min(x)}$$

Donde la función min obtiene el valor mínimo y la función max el valor máximo de los elementos del vector x.

La normalización MIN-MAX modifica el valor a un intervalo entre 0 y 1. La normalización valor-z se aplica cuando es imposible realizar la normalización MIN-MAX o se teme que valores atípicos puedan repercutir en el valor normalizado. Definamos el algoritmo valor-z para cada $b_i, i = 1$ hasta n como:

$$b_i = \frac{x_i - \bar{x}}{\sigma}$$

Donde $\bar{x}$ es la media del vector x y σ es la desviación estándar del mismo.

En ocasiones algoritmos de inteligencia computacional requieren de ciertas condiciones para ser aprovechados de la mejor manera. Las transformaciones son la aplicación de funciones o de fórmulas matemáticas a los atributos de la data. Por ejemplo, transformaciones logarítmicas, raíces cuadráticas, cúbicas, aproximaciones polinómicas, datos nominales a binarios, etc. La reducción de datos es necesaria en ocasiones ya que incrementan la complejidad computacional de los algoritmos *machine learning*, por lo que reducir las variables es vital, la realización se puede lograr mediante *machine learning* no supervisado.

"*Apache Spark*" es uno de los "*frameworks*" Big Data usados para el flujo de ciencia de datos, ya que reduce la dependencia del cómputo distribuido (*Hadoop*) mediante el aumento en memoria en el clúster. En *Spark* se puede desarrollar usando lenguajes de programación como Scala, Python o R y permitir la colección y preprocesamiento de datos mediante SQL o cualquiera de los API's anteriores. Se puede trabajar con datos en tiempo real y contiene librerías de inteligencia computacional y teoría de graficas. Incluye flujos *machine learning* (pipelines) para preprocesamiento, entrenamiento y evaluación de modelos (Salloum, Dautov, Chen, Xiaogang Peng, & Zhexue Huang, 2016).

1.2.3 Machine Learning proceso iterativo

En este apartado se definirá *machine learning* y sus aproximaciones al aprendizaje de big data. Se estudiarán algoritmos de inteligencia computacional para darle al lector un panorama de las capacidades del aprendizaje de máquina al resolver distintos problemas mediante su forma matemática.

En el estudio de Shai Shalev-Shartw y Shai Ben-David (2014) para entender los algoritmos de inteligencia computacional donde concluyen que estos consisten en enseñar a una máquina cierta actividad para que logre aprenderla mediante un meta-algoritmo de aprendizaje estadístico. Estas actividades deben ser complejas, ya que el tiempo de aprendizaje puede ser extenso y de no ser compleja se pueden considerar otras estrategias para la solución de problema. Son dos los principales tipos de aprendizaje, supervisado y no supervisado. En el primero, se le otorga al meta-algoritmo un conjunto de datos de entrenamiento con una variable claramente etiquetada, que será la variable objetivo, mediante un proceso de aprendizaje computacional se aprenderán los parámetros del algoritmo de aprendizaje estadístico; por ejemplo, detección de ordenes fraudulentas. Por el contrario, en el aprendizaje no supervisado no existe arreglo de entrenamiento, todo el conjunto de datos es ingresado y el algoritmo debe detectar patrones o anomalías en la data; por ejemplo, anomalías en transacciones para detección de fraude. Las variables que servirán de entrada al modelo deben ser numéricas, sin embargo, estas pueden ser de tipo cuantitativo o cualitativo, el preprocesamiento de datos se encargará de asegurar que las variables cualitativas se representen numéricamente de la mejor manera. El aprendizaje supervisado se divide en dos grandes ramas la regresión y la clasificación. La regresión se ocupa de una variable objetivo Y continua. La Clasificación se encarga de predecir una variable Y discreta, ya sea dicotómica, fraude o no fraude o múltiple, positivo, neutral, negativo.

Los algoritmos supervisados deben ser capaces de generalizar, por ello la importancia de tener un conjunto de datos de entrenamiento para aprender el modelo $f(x)$ y un conjunto de datos para prueba, con ello podremos saber el error de predicción en la prueba o la capacidad del modelo para generalizar, comparado con tener solo el error de entrenamiento. Cuando el algoritmo *machine learning* tiene buen desempeño en el conjunto de entrenamiento, pero mal desempeño en el conjunto de prueba se le llama sobreajuste y tiene que ver con la imposibilidad del modelo de generalizar. No hay modelo $f(x)$ que pueda generalizar en todos

los problemas de entrenamiento. De acuerdo con Hastie, Tibshirani, & Friedman (2008), mediante la hiperparametrización [14] del modelo de inteligencia de máquina, el error de entrenamiento puede ser reducido. Para definir los parámetros de afinación, se usa un conjunto de datos de validación, así como estimar el error en prueba buscando mejorar el error de entrenamiento. Los conjuntos de entrenamiento y de prueba (*train-test*) deben ser mutuamente excluyentes y generados mediante un proceso aleatorio. La validación cruzada permite estimar el error de prueba, la validación cruzada con K-iteraciones busca resolver el problema de datos escasos, mediante la división de datos en K partes iguales, una de estas partes servirá como arreglo de validación, mientras los demás servirán como arreglo de entrenamiento $K-1$. A través de iteraciones se cambiará la división usada para validación, de esta manera todos los datos serán usados como arreglo de entrenamiento y se tendrá una predicción del error de prueba, mediante los K arreglos de validación usados en K-iteraciones de validación cruzada.

En el aprendizaje supervisado se usan variables de entrenamiento o variables independientes para predecir una variable objetivo discreta o continua, variable dependiente. Consideremos la siguiente matriz X como el arreglo de datos que se posee, con m variables o atributos. Donde Y es la variable objetivo o a predecir.

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix} \; X = \begin{bmatrix} x_{1,1} & x_{1,2} & \cdots & x_{1,m} \\ x_{2,1} & x_{2,2} & \cdots & x_{2,m} \\ \vdots & \vdots & \cdots & \vdots \\ x_{n,1} & x_{n,2} & \cdots & x_{n,m} \end{bmatrix}$$

Supongamos un modelo $y = f(x) + \epsilon$, donde ϵ es el error del modelo. El aprendizaje supervisado busca aprender f a través del ejemplo. Una de las posibles $f's$ a aprender es la regresión lineal, que tiene la siguiente forma matemática, un vector de entrada $d_i = \{x_{1,i}, x_{2,i}, \dots, x_{m,i}\}$ que representa las observaciones para predecir una variable y_i continua. La regresión lineal múltiple es modelada de la siguiente manera:

$$f(d) = \beta_0 + \sum_{i=1}^{n} d_i \beta_i$$

Donde β_0 es la intercepción de la regresión lineal y β_i representa el parámetro para cada i del vector de entrada. En este modelo se asume que la relación entre el vector de entrada y la

[14] Realizada en los parámetros de afinación del modelo o *tuning parameters* en inglés.

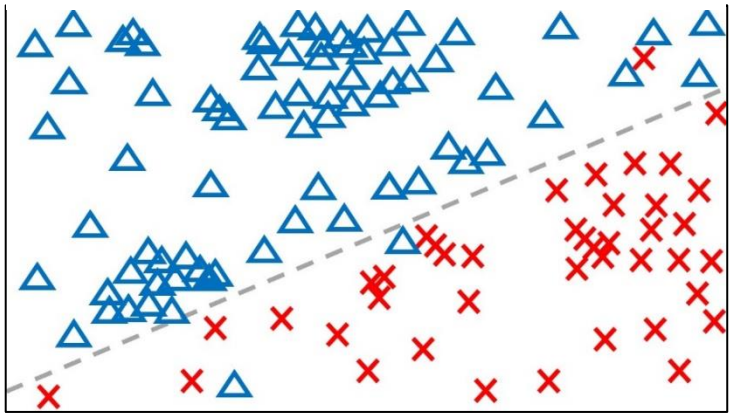
variable a predecir es lineal. Los parámetros β_i no se conocen y para estimarlos se debe minimizar la suma de diferencias cuadráticas que es la función de pérdida o de riesgo. Siendo X la matriz (n, m) de entrada, compuesta por los vectores d_i desde 1 hasta n. Y y_i la variable objetivo o dependiente, desde $i = 1$ hasta n. Se define la suma de diferencias cuadráticas, SSE[15] como:

$$SSE(\beta) = \sum_{i=1}^{n} (y_i - f(d_i))^2$$

Se puede reducir el número de atributos del conjunto de datos, que como veíamos en el apartado anterior es una buena aproximación para una mayor eficiencia computacional, mediante regresión lineal. En la selección de un subconjunto de datos, se pretende mejorar la exactitud del modelo y mejorar su interpretación, ya que los coeficientes del modelo adoptan valor cero o son eliminados y por lo tanto las variables del arreglo de datos no pertenecen más al modelo, haciendo este más simple. Las aproximaciones a este problema mediante regresión lineal es la selección hacia adelante y hacia atrás por pasos. Otra aproximación es la regularización de Tijonov[16] donde se penaliza los coeficientes para que estos adopten valores cero en la suma de diferencias cuadráticas. Donde $x_{i,j}$ es un elemento de la matriz (n, m):

$$\hat{\beta}^{Ridge} = \min_{\beta} \left(\sum_{i=1}^{n} (y_i - \beta_0 - \sum_{j=1}^{m} x_{i,j}\beta_j)^2 + \lambda \sum_{j=1}^{m} \beta_j^2 \right)$$

Figura 1.4: Problema de clasificación



Fuente: Elaboración propia.

La anterior es la función de costo a minimizar, comparada con la suma del error cuadrático, en esta existe un coeficiente de complejidad λ para reducir el valor de los parámetros. Un valor λ mayor, penalizará más los valores, generando que más parámetros adopten valor cero. El coeficiente de complejidad

[15] Por sus siglas en ingles "*Sum of Squared Errors*".
[16] Regresión Ridge o regularización L2

es definido mediante un proceso iterativo de acuerdo con la selección del modelo.

Siguiendo con Hastie et al. (2008), los problemas de clasificación tienen un conjunto de datos finito a predecir, por lo que se puede dividir el espacio dimensional en la cantidad de etiquetas a clasificar. Estas divisiones se pueden realizar mediante barreras de decisión lineales, problemas lineales, como lo muestra la figura 1.4. A diferencia de la regresión lineal, la regresión logística se asegura de tener un valor entre 0 y 1 para la variable a predecir y_i, pertenece a {0, 1}, dadas las observaciones d_i de la matriz X, mediante la máxima verosimilitud. La función a aprender es la log-verosimilitud del vector de observaciones d_i, sujetas a los parámetros β dada la variable a predecir:

$$f(\beta) = \sum_{i=1}^{n} \log p_{y_i}(d_i; \beta),$$

Los problemas dicotómicos para encontrar la recta que mejor ajuste los datos se pueden rescribir de la siguiente manera, suponiendo que el vector d_i contiene la constante 1 para β_0. Donde β^T es el vector transpuesto para lograr una adecuada multiplicación de vectores y e es la contante de Euler (≈ 2.7182):

$$f(\beta) = \sum_{i=1}^{n} \{y_i \beta^T d_i - \log(1 + e^{\beta^T d_i})\}$$

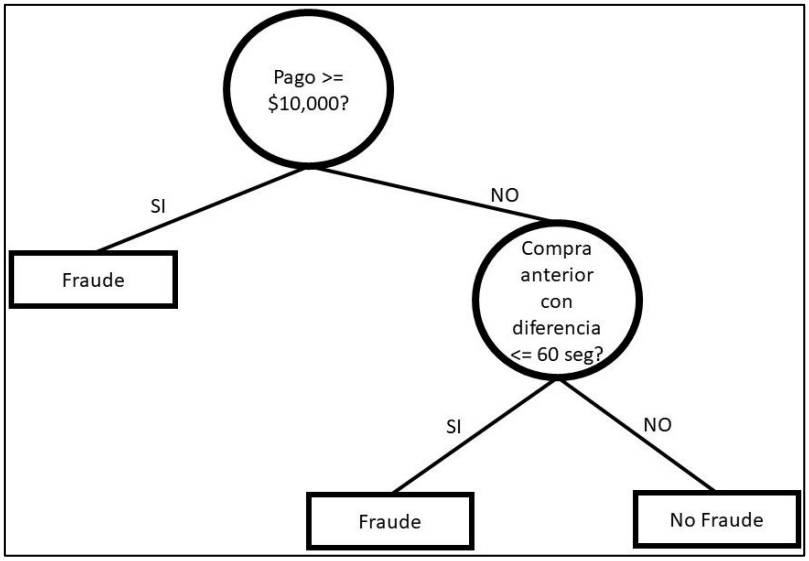
En la regresión logística las barreras de decisión se ajustan mediante la maximización de la log-verosimilitud. Esto se logra fijando su derivada a cero:

$$\frac{\delta f(\beta)}{\delta \beta} = \sum_{i=1}^{n} d_i \big(y_i - p(d_i; \beta)\big) = 0$$

Donde a la suma del producto de los m elementos del vector d_i por de la diferencia entre el valor objetivo observado y la probabilidad predicha se iguala a cero, sujeto a los parámetros β. Se debe de fijar un umbral para la barrera de decisión del valor de salida {0, 1} eligiendo así que valores adoptará la clase. Los coeficientes aprendidos β se pueden interpretar tomando en cuenta las modificaciones logarítmicas para el ajuste de las probabilidades condicionales. Al igual que con la regresión lineal (algoritmo estadístico con valor de salida continuo), la

regresión logística tiene variaciones de normalización L1, Lasso, para penalizar coeficientes muy altos.

Figura 1.5: Ejemplo árbol de decisión.



Fuente: Elaboración propia.

Shalev-Shwartz y Ben-David (2014) proponen otro algoritmo, los aboles de decisión, que sirve tanto para problemas de clasificación como de regresión, pero más comúnmente usados para problemas de clasificación. Los árboles de decisión se basan en la división mediante las variables del arreglo de entrenamiento. En la raíz se toma la primera decisión, donde se elige el atributo óptimo para dividir los datos, este proceso se repite de acuerdo a los parámetros del algoritmo de inteligencia de máquina como muestra la figura 1.5. Para evitar que el algoritmo sobreajuste se puede limitar el largo de la rama. Para elegir la división óptima se busca maximizar la medida de ganancia de datos como el algoritmo de ganancia de información o el de entropía, que buscan hacer la división en el mejor atributo o aquel donde exista mayor información, esto, cuando no todos los datos pertenezcan a la misma clase, de ser así, esa rama pertenece a dicha clase. Supongamos un vector de variable objetivo Y , con C clases distintas. Se define la entropía como:

$$E = \sum_{i}^{C} p_i log_2 p_i$$

Donde p_i es la proporción de la división del arreglo de datos con elementos de la clase i. El algoritmo de información de ganancia usa nociones de entropía y se debe de calcular para todos los atributos posibles m. Siendo p_j la proporción de los datos del arreglo restante con los datos obtenido de hacer la división en el atributo j por la entropía del mismo, buscando que nuestras divisiones maximicen el algoritmo de información de ganancia:

$$IG = \sum_{j}^{m} E_j p_j$$

Cuando se encuentra el mejor atributo, este se tomará como nodo para la división del árbol y se iterará este proceso, buscando que no sobreajuste. Para evitar el sobreajuste se puede recurrir a métodos de ensamble como bosques aleatorios (Random Forest), que son conjuntos de árboles de decisión. La decisión final se basará mediante la votación de todas las salidas de los n árboles de decisión de forma unitaria, alimentados con una muestra aleatoria del conjunto de datos de entrenamiento.

Además, proponen otro algoritmo, las redes neuronales artificiales (*artificial neural networks*) que están inspiradas en el funcionamiento del cerebro. Las redes neuronales artificiales consisten de neuronas que reciben pesos, calculados de las neuronas anteriores y calculan pesos para las neuronas posteriores, este proceso se conoce como propagación hacia adelante, para optimizar los modelos $f(x)$ de cada neurona y por lo tanto los pesos, se usa un proceso conocido como propagación hacia atrás mediante gradiente descendiente estocástico, SGD[17]. SGD tiene como objetivo minimizar la función de pérdida en problemas de aprendizaje convexo. Mediante un proceso iterativo, a través de pasos en dirección aleatoria toma el negativo del gradiente de la función de perdida a minimizar.

Las métricas de los sistemas clasificadores que utilizaremos para conocer la efectividad de los modelos serán cuatro, exactitud, precisión, exhaustividad y valor F1. La exactitud sigue la siguiente forma:

$$\frac{VP + VN}{VP + VN + FP + FN}$$

Donde:

- VP: Son los verdaderos positivos, aquellas observaciones que fueron predichas como positivas (1) y en realidad son positivas.
- VN: Son los verdaderos negativos, aquellas observaciones que fueron predichas como negativas (0) y en realidad son negativas. Estas primeras dos representan lo ideal, lo esperado al entrenar un modelo.

[17] Del inglés, Stochastic Gradient Descent

- FP : Son las observaciones Falsas Positivas, las cuales representan aquellas observaciones que se han marcado como positivas, pero en realidad no lo son.
- FN: Son los falsos negativos, aquellas observaciones que fueron marcadas como negativas, sin embargo, han sido positivas.

La precisión permite conocer que tan bien identificamos las observaciones positivas sobre aquello predicho como positivo, mediante la siguiente fórmula:

$$\frac{VP}{VP + FP}$$

La exhaustividad, muestra que tan bien se identifican los positivos reales del número total de positivos. Por medio de la siguiente fórmula:

$$\frac{VP}{VP + FN}$$

Una exhaustividad alta quiere decir que encuentra todos los datos positivos, sin embargo, no quiere decir que no esté considerando datos que no son positivos. Mientras que una precisión elevada predice como positivos a los que realmente son positivos, bajo el costo de dejar afuera muchos positivos. El valor F1 es una medida que considera tanto la precisión como la exhaustividad, mediante la siguiente fórmula:

$$F_1 = * \frac{precisión * exhaustividad}{precisión + exhaustividad}$$

Uno de los problemas resueltos por algoritmos no supervisados es la agrupación[18]. Estos algoritmos nos pueden ayudar a segmentar usuarios con transacciones fraudulentas, dicha segmentación agrupará transacciones similares o próximas y aquellas que no sean iguales terminaran en otros grupos. K-Medias[19] tiene por objetivo encontrar el conjunto de *clusters* que minimice la función de costo. Dicha función toma en consideración la distancia d entre el centro del *cluster* c (centroide), del vector de clusters $C = [C_1, C_2 \dots C_n]$, representados por C_j y la observación *x* que pertenece al conjunto de datos de entrada *X*. Sin embargo, el cálculo de

[18] También conocido por su forma en inglés, Clsutering.
[19] También conocido por su forma en inglés, K-Means.

todas las distancias para su minimización puede ser complejo en términos computacionales (NP complejo). La función objetivo de K-medias tiene la siguiente forma:

$$= \min_{c \epsilon X} \sum_{x \epsilon C_j} d(x, c)^2$$

El centro de los clusters se inicia aleatoriamente y mediante un proceso iterativo se busca minimizar la función objetivo, moviendo el centro de los clusters en la dirección de convergencia.

Otra familia de algoritmos no supervisados son los de reducción de dimensiones, su objetivo es reducir un arreglo de múltiples dimensiones D y mapearlo a un nuevo arreglo de d dimensiones, donde $D > d$. Análisis de componentes principales, PCA[20]; es un algoritmo de reducción de dimensiones a través de transformaciones lineales. Estos algoritmos sirven para mejorar el rendimiento de los algoritmos principales *Machine Learning* o tener la capacidad de visualizar en dos o tres dimensiones gráficamente.

Todos los algoritmos anteriormente detallados han sido desarrollados por el proyecto de licencia libre scikit-learn. Esta es una librería de algoritmos de inteligencia de máquina codificada en Python, permitiendo evaluar y seleccionar modelos, transformar arreglos de datos mediante pipelines (flujos) predefinidos, acceder a todos los recursos de máquina con la capacidad de procesar en paralelo e implementar algoritmos machine learning supervisados y no supervisados; permitiendo como entrada tipos de datos Pandas (Pedregosa, y otros, 2011). Tambien han sido implementados algoritmos de inteligencia computacional en el framework de Spark en Python.

1.3 Revisión de la literatura

En esta sección analizaremos una serie de trabajos relacionados con el problema de reclutamiento y selección e Inteligencia Artificial, donde se tomó en cuenta términos de inteligencia artificial como Big Data, Machine Learning, data science, data mining o estadística; y los terminos de reclutamiento y selección en la administración de recursos humanos, mediante la siguiente regla de búsqueda:

[20] También conocido por su forma en inglés, *Principal Component Analysis*.

("big data" OR "machine learning" OR "artificial intelligence" OR "data science" OR "data mining" OR statistic) AND ((recruitment OR selection) AND ("human resource" OR "people management"))

Un cambio en las habilidades de los administradores será necesario para una correcta adopción de técnicas de ciencia de datos en el área de recursos humanos, ya que por ahora es escasa y se debe a bases de datos pequeñas y problemas definidos vagamente, por lo que un proceso de ciencia de datos bien estructurado es necesario (Tambe, Cappelli, & Yakubovich, 2019). Oswald, Behrend, Putka, y Sinar (2020) abordan los problemas del area de recursos humanos en la adopción de Big Data, destacando los problemas en infraestructura (falta de empleados), manejo de datos, fuentes de datos y la importancia de la información pública y abierta en la red y su aprovechamiento mediante herramientas de inteligencia artificial, añadiendo la importancia de bases éticas para su correcto aprovechamiento. Adrián Todolí-Signes (2019) y Dena Mujtaba & Nihar Mahapatra (2019) estudian la importancia de conocer los metodos de los algoritmos de inteligencia de máquina para la toma de decisiones referentes a los empleados y su protección necesaria de datos, mediante ejemplos de monitoreo de empleados y la automatización de procesos; asi como la idea de justicia en los procesos de reclutamiento y selección. Los procesos de automatización requieren toma de decisiones no controladas por humanos y puede tener concecuencias discriminatorias difíciles de regular por la posición de un aspirante en la búsqueda de un empleo, por ello la importancia de un marco que otorgue al aspirante la capacidad de conocer las razones y procesos de toma de decisiones realizadas por el algoritmo o el uso de técnincas en el proceso de inteligencia computacional para mitigar sesgos. Karolina Rab-Kettler y Bada Lehnervp (2019) proponen la figura de gestión humanística (humanistic management) y su derivado reclutamiento humanístico a favor de procesos de reclutamiento machine learning, siendo favorable para el reclutado al ser un proceso más rápido y menos sesgado y al reclutador le permitirá realizar actividades de mayor valor y significantes. Hung-Yue Suen, Mavis Yi-Ching Chen y Shih-Hao Lu (2019) estudian los efectos que tiene en el aspirante los procesos de reclutamiento por video sincrónicos (en tiempo real) asincrónicos y el uso de herramientas de inteligencia artificial, encontrando que los vídeos asincrónicos tienen un efecto negativo en los aspirantes, sin embargo, provocan menos sesgos iniciales por parte del reclutador; mientras que no se encontró perpceción de injusticia en los aspirantes con el uso de inteligencia artificial.

Los algoritmos de recomendación y emparejamiento de lenguaje natural de información del candidato con lo datos requeridos por el puesto son propuestos por (Truică & Barnoschi, 2014; Coelho, Costa y Gonçalves, 2016; Celik, y otros, 2013). Duncan Dickson y Khaldoon Nusair (2010) estudian el procesamiento, almacenamiento y técnicas de inteligencia atificial en datos de candidatos en la industria de la hospitalidad. Qing Xie (2019) añade a los problemas de sistema de recomendación de candidatos, el enfrentamiento de recolección y procesamiento de datos, para la creación de una base de datos profesional. El algoritmo usado busca relacionar capacidades del candidato con sus deseos o necesidades, tomando en cuenta donde vive, donde ha trabajado, datos personales y su relación en redes sociales. Prafulla Bafna, Shailaja Shirwaikar y Dhanya Pramod (2019) también proponen un algoritmo de recomendación basado en clusterings semánticos, el sistema es capaz de detectar habilidades en curriculumns mediante procesamiento de texto comparadas con las habilidades necesarias, que seran agrupadas. Por otro lado, se propone un modelo con redes neuronales (Ye, y otros, 2019) para detección de talento mediante interacciones sociales modeladas utilizando grafos. Las interacciones sociales durante los estudios universitarios del aspirante pueden ser usadas para predecir el crecimiento de un individuo en la organización (Liu, Li, Wang, & He, 2019; Chen & Chien, 2011), mediante modelos clasificados como regresión logistica, bosques aleatorios y redes neuronales artificiales. Youzheng Chang y Ming Guan (2008) utilizan técnicas de agrupación, árboles de decisión y redes neuronales artificiales para la evaluación del personal en la industria de la construcción. Ion Ivan, Eduard Budacu y Mihai Liviu Despa (2019) estudian la identificación de talentos mediante una página especializada para agrupar candidatos que se ajustarían al equipo colaborativo de manera más efectiva mediante K vecinos cercanos, enfrentandose al problema de imputación de datos con técnicas Machine learning. Bafna, Pillai y Pramod (2016) tambien estudian parámetros de desempeño desde técnicas de aprendizaje no supervisado, clasificación y selección de atributos. Qiangwei Wang, Boyang Li y Jinglu Hu (2009) también estudian la selección de atributos para la mejora de la predicción del algoritmo de soporte de máquina para la selección de candidatos. Jean Rabcan, Monika Vaclavkova & Rudolf Blasko (2017) y Peng Ye (2011) estudian la selección de candidatos internos utilizando arboles de decisiones. También se puede automatizar el proceso de reclutamiento y selección, reduciendo el conjunto de candidatos mediante patrones extraídos de conocimiento experto con agentes basado en reglas (Ali & Rajamani, 2012). El problema con la evaluación del desempeño radica en lo vago que es este y su medición, por

lo que (Doctor, Hagras, Roberts, & Victor, 2009) proponen uso de matemática difusa, y (Jing, 2009) añade el uso de un agente basado en reglas.

Para la selección nacional de jugadores en equipos de fútbol soccer (Shahriar, Islam, & Amin, 2019) se utilizan distintos modelos de clasificación: "*Naive Bayes"*, árboles de decisión, vectores de soporte de máquina, K vecinos cercanos y bosques aleatorios. Usando variables como número de juegos, asistencias de goles, tarjetas amarillas, tarjetas rojas, minutos jugados, goles anotados; categorizando por posiciones. También se pueden aplicar métodos no supervisados de agrupación para el mercado de transferencias, junto con regresión logística para predicción del desempeño, usando además técnicas de reducción de variables de entrada (Kim, Bui, & Jung, 2019). Utsav Jagdishbhai Solanki y Jay Vala (2017) estudian la selección de jugadores para balancear las habilidades de equipos de cricket, usando reglas de asociación, encontrando que el desempeño de habilidades especificas es posible de encontrar, comparado con la selección de un equipo total y la importancia de conocimiento experto en el juego a estudiar.

Nishad Nawaz (2019) estudian la toma de decisiones en empresas que utilizan técnicas de inteligencia artificial comparadas con aquellas organizaciones que no las utilizan en los procesos de reclutamiento, encontrando que las primeras tienen mejores desempeños y principalmente son usadas al principio del proceso de reclutamiento y selección. Mediante un caso de estudio, Pooja Gupta, Semila Fernandes y Manish Jain (2018) se plantean la preparación de las organizaciones para incluir herramientas de inteligencia artificial y automatización y que oportunidades se bosquejan en un futuro. Enfocándose en el área de recursos humanos y la posibilidad de perder el toque humano o deshacerse de los empleos de esta área. Sarah Guilfoyle, Shawn Bergman, Cristopher Hartwell y Jonathan Powers (2016) revisan la literatura de reclutamiento y selección mediante el uso de datos, enfocándose en redes sociales; encontrando problemas con la privacidad y consentimiento por parte de los implicados, inconsistencia, rapidez y verificación de asertividad de datos. Kirstie Ball (2001) estudian el uso de sistemas de información de recursos humanos en distintas organizaciones y como se relaciona la cantidad de colaboradores con los análisis realizados. Chen-Fu Chien y Li-Fei Chen estudian las técnicas de árboles de decisión de otros trabajos para la selección de personal, así como sus procesos de ciencia de datos.

Se ha estudiado la detección de personalidad del modelo de cinco factores, 16 factores de personalidad y prueba Eneagrama, mediante texto usando redes neuronales profundas (Yılmaz, Ergil, & İlgen, 2020; Sanchez, Capel, Jiménez, Rodriguez-Fraile, & Pegalajar, 2018). Faliagka, Tsakalidis y Tzimas (2012) añaden la extraversión para mejorar la clasificación de candidatos, a través de textos en redes sociales. Khosla, Chu y Nguyen (2016) detallan el modelamiento de comportamiento y emociones mediante audio y video durante el proceso de reclutamiento, para una mejor selección. Eduardo Santiago y Glenn Paul Gara (2018) estudian la predicción de abandono de un candidato a seleccionar, enfocándose en el producto de datos final. Usando “naive bayes” y reglas de asociación, entrenando el modelo usando como entrada un cuestionario de aquellos empleados que llevaban más de 2 años en la empresa estudiada, asociándolos con los candidatos más parecidos.

2 Fútbol Americano Profesional

En este capítulo se dará una breve introducción al fútbol americano profesional, explicando cómo se juega, la importancia de las posiciones de cada jugador, la dinámica en la *National Football League* y el fútbol americano colegial en los Estados Unidos de Norteamérica y sus procesos de reclutamiento y selección[21]. Además, se realizará un estudio de las características que habilitan a la NFL a ajustarse a estos meta-algoritmos mediante un análisis económico del mercado laboral o de jugadores.

2.1 El juego

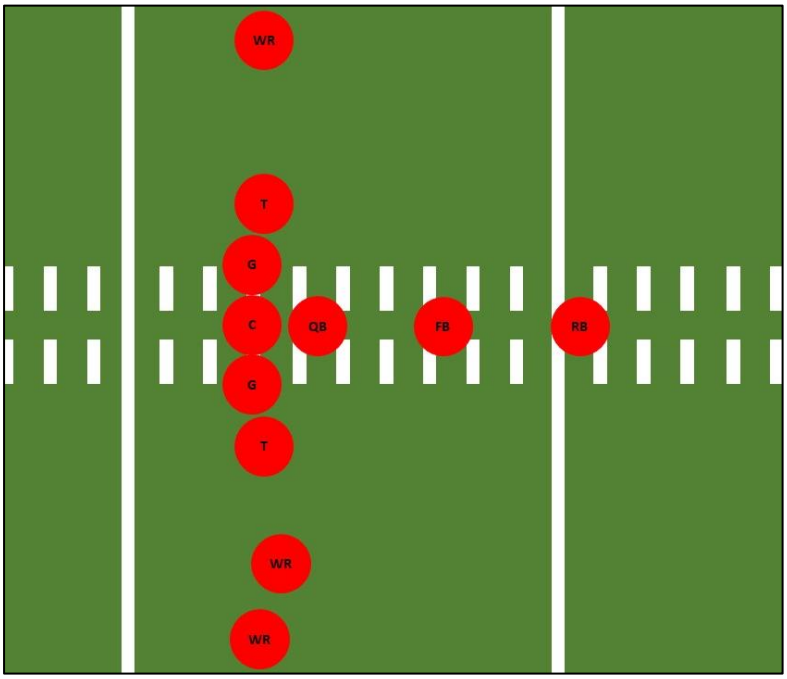
El fútbol americano profesional tiene por objetivo desplazar el balón a la zona final opuesta, ya sea corriendo con el balón o pasando este. El desplazamiento puede ser detenido mediante una tacleada o cuando el balón, al ser pasado, cae al suelo en lugar de a las manos de la persona que lo recibe. Los que desplazan el balón se conocen como la ofensiva y tienen 4 oportunidades[22] para avanzar 10 yardas, una vez avanzadas se tienen otras 4 oportunidades hasta que son detenidos o llegan a la zona final opuesta, anotando 6 puntos, a esto se le llama "touchdown". Una vez que se llega a la zona de anotación se tiene derecho a un punto extra, ya sea mediante una patada, con valor a un punto o intentando llegar a la zona de anotación por segunda vez, siendo acreedores de dos puntos. De ser detenidos, se patea al balón para anotar 3 puntos o para alejar al equipo opuesto de su zona final mediante un "*punt*", comúnmente en la última oportunidad. La patada (*kick*) para anotar 3 puntos o el punto extra debe pasar en medio de dos postes. El equipo ganador es aquel que tiene más puntos al final del juego. Este consta de 4 cuartos de 15 minutos cada uno y en caso de empate se jugarán otros 15 minutos. Cada equipo debe tener 11 jugadores en campo, ya sea en defensiva u ofensiva.

2.1.1 Posiciones

La ofensiva consiste de una serie de posiciones que tienen actividades distintas para alcanzar un único objetivo, llegar a la zona de anotación. El centro tiene como trabajo entregar el balón

[21] Información extraída de sitios web oficiales: www.operations.nfl.com, www.nfl.com y www.ncaa.com
[22] También conocidas por su palabra en inglés "downs".

Figura 2.1: Ejemplo formación ofensiva.



Fuente: Elaboración propia.

muerto (listo para juego) y pertenece a lo que se conoce como línea, que consta de guardias y tacles, los guardias son dos y se encuentran uno a cada lado del centro y los tacles también son dos y se localizan uno a cada lado de los guardias. Los dos últimos tienen como objetivo contener a los jugadores de la defensiva, que pretenden detener al mariscal de campo, corredor o pateador. Otras posiciones que tienen como objetivo detener a la defensiva son los corredores de poder y las alas cerradas[23], pero también para desplazar el balón y recibir este, respectivamente. El mariscal de campo es el líder de la ofensiva, reúne a su equipo para que todos conozcan la jugada y se mantiene en constante comunicación con los entrenadores. Al recibir el ovoide del centro, puede correr con el balón, asignárselo al corredor[24], corredor de poder o pasarlos a los receptores. Los receptores pueden ser abiertos, de Slot, o alas cerradas. Cada posición tiene un trabajo específico y por ello las variaciones en las posiciones generales.

La defensiva busca detener las anotaciones del otro equipo (la ofensiva), permitiéndoles avanzar lo menos posible, a través de tacleadas, intercepciones (atrapando el balón lanzado por el mariscal de campo) o hacer que suelten los ovoides de jugadores[25] corriendo (en el equipo de la ofensiva). La defensiva se divide en primaria y secundaria, la primaria busca detener a corredores y taclear al mariscal de campo antes de que pueda pasar el balón, la secundaria se encarga de evitar que los receptores atrapen el ovoide lanzado por el quarterback; sin embargo, hay jugadores dinámicos que buscan detener las corridas o interrumpir los pases, esto no quiere decir que no sepan su rol o su tarea, simplemente es más

[23] "*Full-back*": FB y "*tight-end*" respectivamente
[24] *Running Back*: RB
[25] Conocido por su forma en inglés "fumble"

completa su labor a desempeñar. Al igual que en la ofensiva, cada posición tiene una tarea específica a desarrollar y en ocasiones un jugador espejo en la ofensiva, cerrando la oportunidad del otro equipo de anotar puntos. Por último, se encuentran los equipos especiales, encargados de patear y devolver el balón al inicio del juego, después de cada touchdown, después del medio tiempo, en patadas para tres puntos y en patadas "*punts*".

Los jugadores y por lo tanto las características de las posiciones están en constante evolución. En el principio del juego, eran pocos los jugadores que se dedicaban al fútbol americano de tiempo completo, teniendo otros trabajos, poco entrenamiento específico y nula experiencia en el fútbol americano colegial (como hoy en día). La especialización se dio gracias a cambios en las reglas, limitando substituciones en los jugadores o que estos jugaran en múltiples posiciones tanto de ofensiva como de defensiva. Teniendo como consecuencias óptimos pesos, alturas y habilidades más especializadas como velocidad, empuje, arrastre, etc. para cada variación de posición, y así ajustar las estrategias tanto ofensivas como defensivas. Dando como resultado que en la actualidad sean cuarenta y seis los jugadores activos en cada equipo. Las características físicas de los jugadores son diversas tanto en la defensiva como en la ofensiva, por ejemplo, la línea, tanto defensiva como ofensiva, es muy alta y pesada; por el contrario, los corredores son medianamente pesados y de estatura por debajo del promedio. Estas características se han ido ajustando a lo largo del tiempo, adaptándose a los estilos de juego de cada época.

2.1.2 Estructura de la liga profesional y colegial

Son dos las conferencias de equipos de fútbol americano profesional que conforman los treinta y dos equipos de la liga nacional de fútbol americano, la americana y la nacional[26] ambas con cuatro divisiones y cuatro equipos en cada división. Son cuatro enfrentamientos de pretemporada para preparar la temporada regular, que consta de dieciséis juegos. Seis enfrentamientos se deben de jugar contra los otros equipos de la división, seis juegos contra equipos de la misma conferencia, cuatro enfrentamientos contra equipos de la conferencia a la que no se pertenece. Son seis las organizaciones que participan en la postemporada. Los cuatro equipos con mejor récord de cada división y dos comodines[27] con el quinto y sexto mejor récord. En postemporada, se juegan cuatro juegos de comodines y dos enfrentamientos más

[26] *Americal Football Conference*: AFC; y *National Football Conference*: NFC, respectivamente.
[27] Conocidos en inglés como "*wildcards*".

para llegar al "*Superbowl*"[28]. El fútbol americano colegial, NCAA[29], también está dividido en conferencias:

- Atlantic Coast
- American Athletic Conference
- Big 12
- Big Ten
- Conference USA
- Independents (FBS)
- Mid American
- Mountain West
- Pac-12
- Southeastern
- Sun Belt

Algunas de éstas, se fragmentan en divisiones. Siendo un total de 130 equipos colegiales pertenecientes a la NCAA. Los equipos que jugarán en las semifinales son definidos por un comité de trece personas con experiencia como entrenadores, jugadores, administradores, directores atléticos y periodistas. Estos decidirán los mejores veinticinco equipos y los enfrentamientos en los tazones. Se juegan dos semifinales y un campeonato nacional, sumado a cuatro tazones con los mejores equipos, un total de seis tazones: *Cotton, Fiesta, Orange, Peach, Rose* y *Sugar*. Se tomarán en cuenta factores como juegos ganados, competitividad del calendario, si se gana o no la conferencia, videos de enfrentamientos, estadísticas, etc. Mediante un sistema de votación iterativo, buscando coincidencias entre los votos.

La NFL y la NCAA están en constante comunicación para desarrollar mejores jugadores, que conozcan y se adapten a las reglas de la NFL en caso de ser elegibles. Los jugadores son seguidos durante el bachillerato y el fútbol americano colegial, proveyéndoles constante retroalimentación. Los reclutadores analizan a miles de aspirantes en cientos de universidades recabando información específica de los candidatos. Tanto en la liga de fútbol americano colegial, como en la profesional; las estadísticas están presentes para un mejor entendimiento

[28] Tazón de fútbol americano profesional, donde se define el ganador anual de la liga.
[29] Del inglés, *National Collegiate Athletic Association*

de la liga por parte de los equipos, la administración y los espectadores. En “Kaggle”[30], se planteó el reto de predicción de avance de un jugador una vez que se ha recibido el balón. Tomando en cuenta la posición inicial del receptor, posición de recepción del balón, posición, años de experiencia, yardas promedio, etc.

2.1.3 Draft

La principal plataforma de reclutamiento y selección en la NFL se conoce como “draft”. El cual es un proceso anual que dura tres días para seleccionar jugadores del fútbol americano colegial en EE.UU. Permitiendo a los equipos hacerse de jugadores que pueden tener mucho talento en el futuro, por ello la importancia de su identificación ya que separa a equipos exitosos de aquellos que no lo son. El primer problema radica en la cantidad de jugadores colegiales que hay que evaluar. La evaluación mediante el combine es costosa por la cantidad de tiempo que se invierte y los recursos monetarios que necesita por la cantidad de pruebas médicas, físicas y psicológicas que se realizan. A lo largo del tiempo, más dinero ha sido alocado en la NFL, por lo que sus procesos han ido mejorando como el combine y el draft. El *combine* nacional es una serie de actividades físicas, médicas e interacciones centralizadas en un solo lugar para que todos los equipos puedan evaluar a los candidatos elegibles al draft. El *combine* nacional es muy selectivo y son menos de cuatrocientos jugadores los que son evaluados a lo largo de cuatro días. En caso de que un jugador no haya sido elegido al combine, se le puede invitar a las instalaciones del equipo para llevar a cabo pruebas físicas y mentales.

El draft tiene por objetivo dar a los equipos la oportunidad de contar con talentos nuevos mediante un sistema justo y regulado, compitiendo en una serie de rondas. Son siete las rondas que componen al draft, en principio cada equipo tiene derecho a una selección en cada ronda. El primer equipo es definido como tal por ser el último en la tabla de posiciones la temporada anterior y el último equipo es aquel que ganó el Super Bowl, los equipos se definen en orden inverso a dicha tabla de posiciones, tomando en cuenta si se jugó más allá de temporada regular. Puede haber elecciones compensatorias debido a la perdida de jugadores durante el proceso de agencia libre (traspaso de jugadores de equipo a equipo), la cantidad es determinada por el salario del jugador perdido, la experiencia del mismo, su tiempo de juego y su experiencia en postemporada. Al igual que las elecciones del draft regular, estas pueden

[30] Página web comunitaria para ciencia de datos, donde se puede acceder a retos planteados mediante el uso de datos.

ser intercambiadas. Los intercambios se pueden realizar antes o durante el evento, por otras posiciones en drafts futuros o por jugadores de los equipos implicados. Estos intercambios se deben de notificar a los representantes del draft, que se aseguraran que se encuentre en el marco de las reglas.

Los Equipos tienen un representante en el evento, que hace llegar las decisiones de cada elección para que se pueda llevar a cabo el proceso de rondas lineal de los treinta y dos equipos. Una vez que los representantes del draft reciben la información sobre la elección del equipo, se registra y se notifica a todos los equipos de la selección. Las organizaciones tienen como máximo dos minutos para elegir en la primera ronda, siete minutos para la segunda ronda, cinco minutos para la elección en la ronda tercera, cuarta, y quinta, y cuatro minutos para la última ronda. No es necesario tomar una decisión en este intervalo de tiempo, sin embargo, se debe de esperar a que todos los equipos hayan tenido la oportunidad de elegir en esa ronda. Por lo tanto, se pueden ir aquellos jugadores más talentosos. Son candidatos elegibles al draft, aquellos que tengan al menos tres años de haber concluido el bachillerato y ser elegido por el equipo de fútbol americano colegial por lo menos por una temporada y haber finalizado con su elegibilidad colegial. En caso de haberse graduado se puede meter la petición al comité del draft para definir elegibilidad del candidato. Los contratos de un jugador novato son por un mínimo de 4 años sin posibilidad de negociar su contrato durante ese tiempo.

Además del draft, los equipos de la NFL cuentan con la agencia libre para mejorar las plantillas, donde a través de contratos se pueden buscar mejores combinaciones de jugadores siguiendo el acuerdo colectivo de negociación, CBA[31], el cual es un documento con una serie de reglas. Al igual que el draft, este tiene un inicio y un final; el día de inicio es igual al de la liga anual de la NFL. A partir de dos días antes del inicio los equipos pueden comenzar con las negociaciones con los agentes o los jugadores con o sin restricciones de agencia libre. Aquellos sin restricciones son los que tienen un contrato expirado y más de tres años totales en el equipo. Los agentes libres con restricciones son aquellos con tres años acumulados y un contrato vencido, sin embargo, estos deben ser aprobados por el equipo original, en ocasiones con una compensación de elección en el draft y dependiendo de la ronda de selección en la que fue elegido el jugador, donde además se fija el salario anual del jugador. Lo cual es importante debido a que los equipos se ven limitados en el pago de salarios ya que tienen un

[31] Del inglés, *Collective Bargaining Agreement*

tope salarial anual[32]. Si el jugador tiene menos de tres años acumulados y un contrato vencido tiene que ser aceptado un contrato por un año más en caso que el equipo original así lo quiera. El jugador también puede tener una etiqueta de franquicia, convirtiéndolo en un jugador seguro para el equipo. Los salarios pueden ser definidos por el tipo de agente libre y las habilidades del jugador o por el número de elección en las rondas del draft, fijado con base en el tope salarial de todos los equipos.

2.2 Análisis económico del mercado de jugadores

En esta sección se estudiarán los factores que habilitan la implementación de modelos automatizado o semiautomatizados para el reclutamiento y selección con el uso de *machine learning* en la NFL. Empezando por los aparatos de la liga para promover la competitividad, seguido por las características que envuelven el mercado de jugadores con el objetivo de una competencia justa.

El libre mercado promueve como máxima la capacidad de elección de los individuos y las organizaciones, junto con la capacidad de propiedad privada. En el libre mercado, el gobierno no controla los intercambios "voluntarios" entre individuos que tienen sus propios intereses, ya sea bajo el supuesto de la mano invisible de Adam Smith o del orden natural y por lo tanto inamovible de los fisiócratas, donde se apela por la autorregulación de la economía de una nación. De acuerdo con los liberales, la producción y la distribución de bienes están íntimamente relacionadas. En caso de que se busque ajustar la distribución del mercado, la producción de este puede conllevar a puntos no óptimos. Tomasi (2012 citado en Stilz, 2014) asegura que la equidad de libre mercado es la mejor interpretación moral de la justicia o puede ser definido el mercado como justicia procedimental. Zimmerling (1995) asegura que el mercado como impartidor de justicia es el mercado de competencia perfecta, sin embargo, los mercados no tienen las características necesarias para ser considerados de competencia perfecta y, por el contrario, normalmente se generan condiciones cuasi-monopólicas. En un mundo globalizado la competencia es aún más feroz, provocando que esta no sea justa ni libre, creando que los grandes, por las economías de escala, sean los ganadores y aplasten a las pequeñas empresas locales. Por ello, Pfleiderer (2018) expone las razones a favor de la protección de la economía, apelando a que es una obligación moral de las autoridades permitir

[32] En 2020 el tope asciende a $198.2 millones de dólares anuales.

a todos competir de manera sostenible para frenar el crecimiento del distanciamiento entre ricos y pobres.

La competitividad de un mercado es vital para el sano desarrollo de la economía, pero también para atraer el interés de fanáticos en una liga de fútbol americano profesional (Pivovarnik, Lamb, Zuber, & Gandar, 2008), serán mas atractivos aquellos enfrentamientos en donde ambos equipos tengan capacidad de ganar; comparado con la idea general de negocios, donde a la competencia se le debe de eliminar. Lo que ha generado que la liga realice cambios estructurales como el draft, en donde los equipos más débiles tienen la oportunidad de ser más fuertes y los equipos más fuertes volverse más débiles, balanceando así las fuerzas y poderes de los equipos. "*NFL OPS*" es la división de la liga de fútbol americano profesional que se encarga de balancear el juego mediante una cultura de claridad, equidad, consistencia y credibilidad mediante un proceso de revisión sistemática y basado en consensos de reglas claras, tecnología, regulación de plantillas y seguridad de los jugadores. Mejorando a través de un proceso iterativo, comenzando con la retroalimentación de todos los equipos, que será reforzada durante el proceso del combine nacional, donde se incluye al equipo médico y a la NCAA.

Según Michael Schotty (2013) la paridad entre los equipos de la NFL se debe a su sistema de tope salarial basado en las ganancias de la liga, la repartición pareja de ganancias y un sistema de talentos colegial, donde el éxito se encuentra en la función de las habilidades y no de otros factores. Por lo último, es necesario un mercado de jugadores o "laboral" con reglas claras y pensado para impulsar a los débiles como en la NFL, en comparación con la NBA, MLB, NHL[33] o la liga europea de fútbol soccer, donde se puede reclutar a candidatos de distintas escolaridades y edades. Un sistema de reclutamiento y selección donde se comparte información entre los reclutadores de los equipos de la NFL y los entrenadores de fútbol americano colegial. En donde se impulsa a los débiles, pero también se crean oportunidades para los fuertes, mediante un flujo constante de jóvenes jugadores en compensación por los intercambios de grandes estrellas en la agencia libre. El hecho de que exista un flujo constante de personal no quiere decir que la NFL impulse un sistema de cambios constantes y permanentes, pero permite la capacidad de planeación a largo plazo. El tope salarial es el otro factor que contribuye a evitar equipos fuertes por un periodo largo o constante, ya que un solo

[33] National Basketball Association, Major League Baseball y National Hockey League

equipo no podrá poseer a todos los mejores jugadores, pero, por el contrario, estarán distribuidos. Pero tampoco los equipos podrán invertir en jugadores baratos para maximizar ganancias, puesto que existe un mínimo salarial y el objetivo principal no es maximizar ganancias ya que estas se reparten de manera idéntica entre todos los equipos que conforman la NFL.

Las acciones voluntarias de reclutamiento, trabajo, selección y salarios por parte de empleadores y candidatos se le considera un mercado: el mercado laboral. Siendo el mercado de jugadores de la NFL un mercado controlado o cerrado (Abela, 2020) que logra promover la competitividad, mientras que los libres mercados la frenan. Por ejemplo, la liga de fútbol soccer de Inglaterra, en esta las ganancias de la liga se ponderan por el lugar de la tabla en el que estés, ganando más dinero aquellos que estén más arriba. Provocando que los buenos siempre sean buenos, solo pudiendo romper el ciclo si se invierten grandes sumas de dinero con la esperanza de atraer mejores jugadores.

Sin embargo, este control de la NFL, permite que se establezca un mercado laboral con características similares a las de competencia perfecta. De acuerdo con Michael Parkin (1993) un mercado de competencia perfecta tiene las siguientes características:

- Muchas empresas y un producto idéntico.
- Muchos compradores.
- No hay restricciones a la entrada.
- No se goza de ninguna ventaja sobre los participantes.
- Tanto compradores como vendedores están completamente informados.

Para el caso laboral, se entiende al trabajador, en este caso jugador, como oferente y a los equipos, las organizaciones de la NFL, como demandantes. Son muchos los jugadores provenientes del fútbol americano colegial, de universidades igual de competentes, con experiencias muy similares y características físicas equivalentes. Son treinta y dos equipos a los que se puede ingresar una vez que se es apto, de acuerdo con las reglas de la NFL. Existen restricciones de entrada a la oferta, sin embargo, estas son muy claras y requieren principalmente de experiencia en el fútbol americano colegial. Gracias a las reglas establecidas no se puede gozar de una ventaja por ser el equipo ganador, tener capacidades de inversión muy grandes, establecerse en una cierta región geográfica o contar con contactos, como en

otras ligas deportivas. Por último, la liga busca que sean trasparentes las reglas, decisiones, juegos y estadísticas de los jugadores y los equipos. Estas últimas, están disponibles y se invita a los aficionados que las exploren a través de mecanismos como el fútbol americano de fantasía o el tazón de *big data*. Este último, es una competencia anual de analítica de datos para comprender mejor cómo funciona la liga, la cual se asegura de que existan datos suficientes para entender el juego. Contando con una página web[34] exclusiva para analizar estadísticas agregadas de los enfrentamientos. Estas características habilitan la competitividad en el juego, pero también la posibilidad de tener sistemas automatizados de reclutamiento y selección retadores.

[34] www.operations.nfl.com/stats-central/.

3 Desarrollo del Modelo

En este capítulo inicia la investigación propuesta, donde se expondrá el experimento realizado; mediante una clara metodología de ciencia de datos para la creación de un sistema semiautomatizado de reclutamiento y selección de mariscales de campo. Se estudiarán las técnicas Big Data para ciencia de datos, donde se detallará la extracción de jugadores profesionales, junto con su información colegial y de liga, así como la integración del producto interno Bruto por estado y su flujo de carga y transformación, ELT, para la creación de una tabla final que sirva como entrada para el análisis de datos exploratorio. El cual tendrá como objetivo entender los datos y obtener patrones, así como saber que transformaciones serán necesarias en el arreglo, como valores faltantes o atípicos, construcción de variables, etc. En donde se debe asegurar que el arreglo sirva como entrada al modelo. Se aproximará el modelo desde dos perspectivas de regresión y clasificación, donde se utilizarán modelos de referencia y a través de un proceso iterativo, apelar por un modelo que generalice mejor. Se utilizará regresión lineal, regresión logística, métodos de regularización, arboles de decisiones y perceptrones multicapa; buscando construir un modelo que obtenga mejores resultados en su arreglo de prueba, a través de la hiperparametrización del algoritmo de inteligencia de máquina. Por último, se busca entender las necesidades teóricas de la liga de fútbol americano profesional en torno a la selección de jugadores.

3.1 Extracción big data

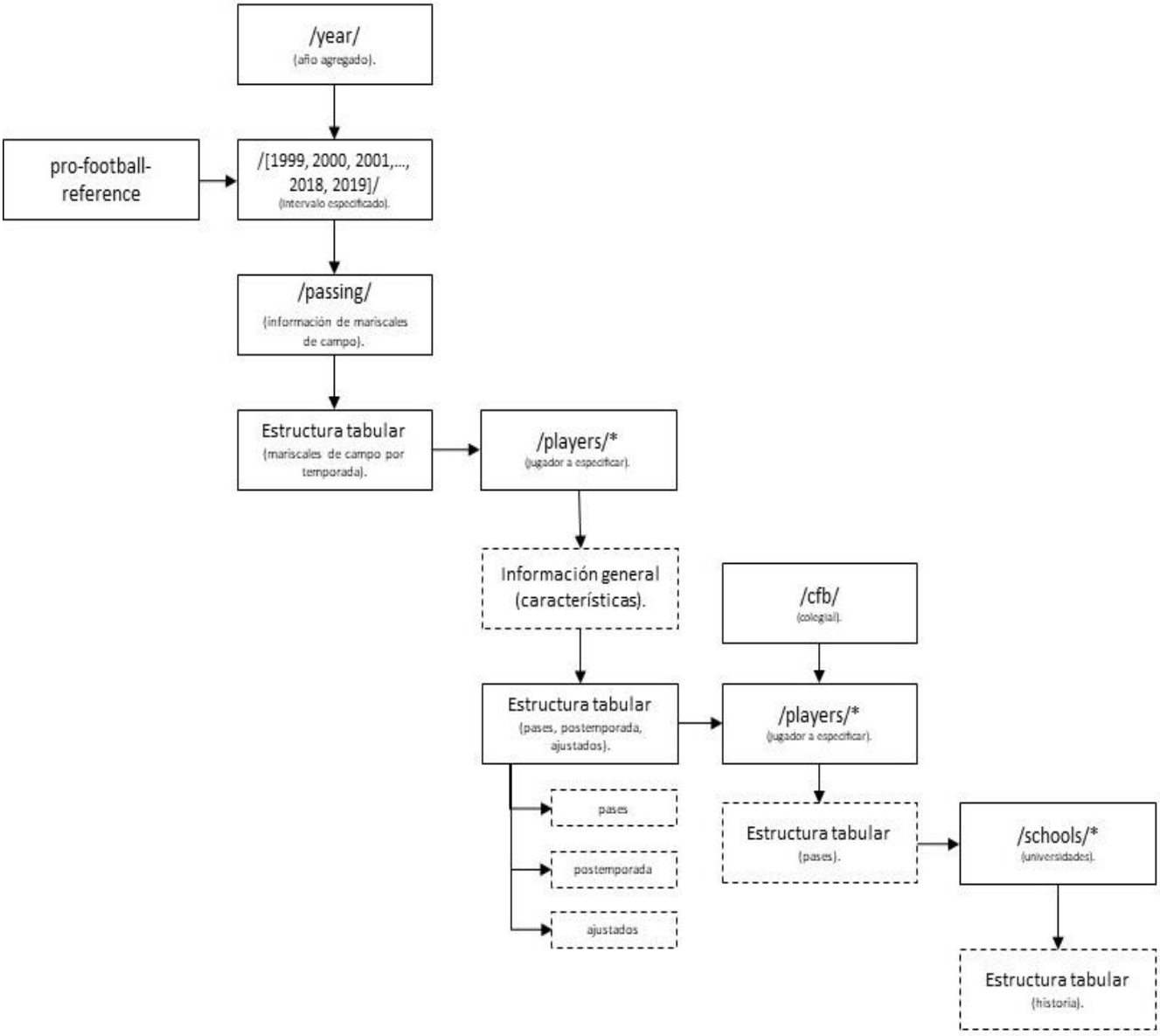
En este apartado se explicará la extracción web Big Data mediante una aplicación bot-araña, desarrollando el proceso de entendimiento de la estructura de la página web. Se detallará gran parte del primer y segundo paso del modelo ELT, y será completado en el capítulo que sigue, mediante el proceso de entendimiento de las páginas web a extraer, la explicación de los métodos usados para extraer la información de forma automática y la carga local de las tablas

Para la extracción de datos web de manera automatizada se realizan bot-arañas o telarañas web[35]. La web superficial está conectada mediante hipervínculos, estos hipervínculos son la puerta para ingresar a otras páginas web. Un rastreo o telaraña web inicia localizando la fuente de la data web, el siguiente paso consiste en el estudio de la estructura de la página web, seleccionar los métodos adecuados de consulta o extracción y por último retornando los

[35] Del inglés, "*spyder bot*" y "*web crawling*".

resultados mediante la extracción web (Wang & Lu, 2016). El objetivo es la creación de una tabla final mediante la extracción de datos, consiguiendo de forma automática y contínua información y estadísticas de mariscales de campo colegiales, junto con su respectiva información ex post en el campo profesional de aquellos que hayan jugado en los último 20 años. Se ha tomado la decisión de extraer información con un intervalo de 20 años con base en lo recabado en la sección de futbol americano profesional sobre la constante evolución de las características de los jugadores. Se utilizó www.pro-football-reference.com como página web base para realizar la extracción de estadísticas colegiales y profesionales de mariscales de campo. La cual incluye estadísticas de deportes de EE.UU. teniendo como objetivo ser fácil de usar, rápida y muy completa. Con una rama especializada en fútbol americano, tanto profesional como colegial; con estadísticas y resultados de juegos, incluyendo datos de jugadores, entrenadores y equipos para cada temporada, así como todas las selecciones del draft.

Figura 3.1: Diagrama de extracción propuesto y flujo de www.pro-football-reference.com.



Fuente: Elaboración propia.

En la Figura 2.1 se muestra el flujo necesario para la extracción de datos profesionales y colegiales de mariscales de campo mediante pro-football-reference, que será detallado a continuación. Cada cambio horizontal en la Figura 2.1 representa una distinta estructura de la página web. Por lo que la casilla de *"/year/*" junto con *"/passing/*"[36] (Figura 2.2) representan el

[36] "/year/" y "/passing/" representan año y pases en español. Se han mantenido en su idioma de origen, ya que los archivos de la página web están en inglés.

archivo de años profesionales de cada uno de los mariscales de campo, donde se encuentra una tabla con hipervínculos al folder de "/players/", con estadísticas profesionales (Figura 2.3 y 2.4), que también conlleva al archivo "/cfb/"[37], que es el archivo que contiene las tablas con datos colegiales (Figura 2.5). Mientras que "/school/*" es el folder que contiene la información de universidades, se ha añadido el asterisco, ya que se debe especificar la escuela a extraer (Figura 2.6). Las casillas punteadas representan información que se extraerá de cada archivo web. La página semilla o página web fuente utilizada fue www.pro-football-reference.com/years/1999/passing.htm. La cual contiene información mediante una estructura

Figura 3.2: Tabla a 19 observaciones de pases por jugador.

Passing * Selected to Pro Bowl, + First-Team All-Pro Share & more ▾ Hide non-qualifiers for rate stats Glossary Toggle Per-Game Stats

Rk	Player	Tm	Age	Pos	G	GS	QBrec	Cmp	Att	Cmp%	Yds	TD	TD%	Int	Int%	1D	Lng	Y/A	AY/A	Y/C	Y/G	Rate	Sk	Yds	NY/A	ANY/A	Sk%	4QC	GWD
1	Brett Favre	GNB	30	QB	16	16	8-8-0	341	595	57.3	4091	22	3.7	23	3.9	196	74	6.9	5.9	12.0	255.7	74.7	35	223	6.14	5.20	5.6	3	3
2	Steve Beuerlein*	CAR	34	QB	16	16	8-8-0	343	571	60.1	4436	36	6.3	15	2.6	208	88	7.8	7.8	12.9	277.3	94.6	50	280	6.69	6.76	8.1	1	1
3	Drew Bledsoe	NWE	27	QB	16	16	8-8-0	305	539	56.6	3985	19	3.5	21	3.9	184	68	7.4	6.3	13.1	249.1	75.6	55	342	6.13	5.18	9.3	2	2
4	Peyton Manning*	IND	23	QB	16	16	13-3-0	331	533	62.1	4135	26	4.9	15	2.8	197	80	7.8	7.5	12.5	258.4	90.7	14	116	7.35	7.06	2.6	6	7
5	Brad Johnson*	WAS	31	QB	16	16	10-6-0	316	519	60.9	4005	24	4.6	13	2.5	178	65	7.7	7.5	12.7	250.3	90.0	29	177	6.99	6.79	5.3	3	4
6	Rich Gannon*	OAK	34	QB	16	16	8-8-0	304	515	59.0	3840	24	4.7	14	2.7	195	50	7.5	7.2	12.6	240.0	86.5	49	241	6.38	6.12	8.7	3	3
7	Elvis Grbac	KAN	29	QB	16	16	9-7-0	294	499	58.9	3389	22	4.4	15	3.0	163	86	6.8	6.3	11.5	211.8	81.7	26	170	6.13	5.68	5.0	1	2
8	Kurt Warner*+	STL	28	QB	16	16	13-3-0	325	499	65.1	4353	41	8.2	13	2.6	197	75	8.7	9.2	13.4	272.1	109.2	29	201	7.86	8.31	5.5		
9	Jon Kitna	SEA	27	QB	15	15	8-7-0	270	495	54.5	3346	23	4.6	16	3.2	168	51	6.8	6.2	12.4	223.1	77.7	32	198	5.97	5.48	6.1	2	2
10	Doug Flutie	BUF	37	QB	15	15	10-5-0	264	478	55.2	3171	19	4.0	16	3.3	157	54	6.6	5.9	12.0	211.4	75.1	26	176	5.94	5.27	5.2	2	2
11	Brian Griese	DEN	24	QB	14	13	4-9-0	261	452	57.7	3032	14	3.1	14	3.1	139	88	6.7	5.9	11.6	216.6	75.6	27	176	5.96	5.23	5.6	1	2
12	Troy Aikman	DAL	33	QB	14	14	7-7-0	263	442	59.5	2964	17	3.8	12	2.7	129	90	6.7	6.3	11.3	211.7	81.1	19	130	6.15	5.71	4.1	1	1
13	Mark Brunell*	JAX	29	QB	15	15	13-2-0	259	441	58.7	3060	14	3.2	9	2.0	154	62	6.9	6.7	11.8	204.0	82.0	29	174	6.14	5.87	6.2	2	3
14	Jim Harbaugh	SDG	36	QB	14	12	6-6-0	249	434	57.4	2761	10	2.3	14	3.2	130	80	6.4	5.4	11.1	197.2	70.6	37	208	5.42	4.51	7.9	2	2
15	Tim Couch	CLE	22	QB	15	14	2-12-0	223	399	55.9	2447	15	3.8	13	3.3	110	78	6.1	5.4	11.0	163.1	73.2	56	359	4.59	3.96	12.3	2	2
16	Jeff Blake	CIN	29	QB	14	12	3-9-0	215	389	55.3	2670	16	4.1	12	3.1	125	76	6.9	6.3	12.4	190.7	77.6	30	168	5.97	5.45	7.2		
17	Jake Plummer	ARI	25	QB	12	11	3-8-0	201	381	52.8	2111	9	2.4	24	6.3	110	63	5.5	3.2	10.5	175.9	50.8	27	152	4.80	2.60	6.6	2	2
18	Jeff Garcia	SFO	29	QB	13	10	2-8-0	225	375	60.0	2544	11	2.9	11	2.9	134	62	6.8	6.1	11.3	195.7	77.9	15	104	6.26	5.55	3.8		
19	Dan Marino	MIA	38	QB	11	11	5-6-0	204	369	55.3	2448	12	3.3	17	4.6	122	62	6.6	5.2	12.0	222.5	67.4	9	66	6.30	4.91	2.4	2	2

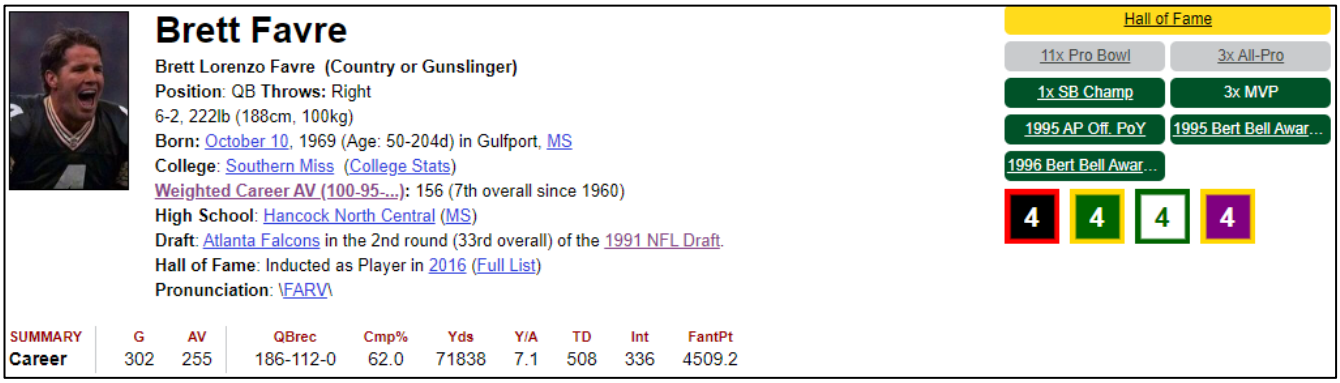
Fuente: www.pro-football-reference.com/years/1999/passing.htm

tabular clasificada de todos los mariscales de campo profesionales que hayan juagado fútbol americano profesional durante la temporada de 1999. Donde "*Rk*" es el identificador (Id) de la tabla. Por medio del estudio de la estructura de la página web, se encontró que mediante el simple cambio de fecha en la liga HTML se puede acceder a la información de las temporadas a extraer. Por lo que la dirección web utilizada para extraer la información de la última temporada será www.pro-football-reference.com/years/2019/passing.htm.

[37] "/players/" y "/cfb/" representan año y acrónimo de "college football", fútbol colegial en español. Se han mantenido en su idioma de origen, ya que los archivos de la página web están en inglés.

Estas páginas web, además, contienen información del equipo al que cada quarterback pertenece, su edad, número de juegos iniciados, récord de juegos ganadas, perdidos y empatados, pases completados e intentados, yardas totales, anotaciones, intercepciones, primeros y dieces, junto con más estadísticas, sin embargo, no serán detalladas ya que no fueron utilizadas en ningún paso de extracción de datos. El único campo utilizado fue el de jugador (Player en la Figura 2.2), el cual, contiene la dirección de la página que almacena las estadísticas y características del jugador durante su trayectoria en la NFL. Es por ello, que en la Figura 2.2, se ven los nombres del jugador en azul o morado indicando que contiene un hipervínculo.

Figura 3.3: Información general de Brett Favre.



Fuente: www.pro-football-reference.com/players/F/FavBr00.htm

La página web del jugador a la que se accede mediante los hipervínculos, previamente detallados, contiene de forma semiestructurada información general del jugador, esta estructura es dinámica y cambia dependiendo de las características del jugador. Las variables generales se detallan en el cuadro 2.1 y la figura 2.3:

Variable	Descripción
Name	Nombre y apodo(s) del jugador.
Position	Posición del jugador: QB.
Weight and Height	Peso y altura.
Throws	Con que extremidad lanza el balón: Derecha o izquierda.
Born	Fecha y lugar de nacimiento.
College	Universidad a la que asistió.

High School	Bachillerato al que asistió.
Draft	Equipo, ronda y selección del draft especificado.
Hall of Fame	Año de entrada al salón de la fama.
Pronunciation	Pronunciación del apellido del jugador.

Cuadro 3.1: Descripción de la información general del jugador.

La página también contiene información agregada por temporada y su agregado total de las siguientes estadísticas:

- Pases en temporada regular.
- Pases en postemporada.
- Pases ajustados.
- Corridas y recepciones.
- Corridas y recepciones en postemporada
- Defensa y balones sueltos.
- Defensa y balones sueltos en postemporada
- Resumen de anotaciones
- Resumen de anotaciones en postemporada.

La tabla de pases en temporada regular contiene 32 columnas, siendo "*Year"* la variable de identificación detalladas en el cuadro 2.2 y la figura 2.4:

Variable	**Descripción**
Year	Año de la temporada.
Age	Edad del jugador.
Tm	Equipo en el que jugaba.
Pos	Posición: QB.
No.	Número en playera.
G	Enfrentamientos jugados.
Gs	Juegos iniciados.
QBrec	Récord en temporada W-L-T.
Cmp	Pases completados.

Att	Intentos de pase totales.
Cmp%	Cmp / Att.
Yds	Yardas en pases.
TD	Pases de anotación.
TD%	Porcentaje de touchdowns cuando pase.
Int	Intercepciones.
Int%	Int / Att.
1D	Pases de primera y diez.
Lng	Pase más largo completado.
Y/A	Yardas por intento de pase.
AY/A	Yardas ajustadas por intento de pase.
Y/C	Yardas por pase completo.
Y/G	Yardas por juego.
Rate	"Rate" de quarterback.
QBR	"Rate" de quarterback calculado por ESPN.
Sk	Sacks totales.
Yds	Yardas perdidas por "sack". Referencia: Yds(sack).
NY/A	Yardas netas por intento de pase.
ANY/A	Yardas ajustadas netas por intento de pase.
Sk%	Porcentaje de "sacks" en intentos de pase.
4QC	Remontadas en 4to cuarto lideradas por quarterback.
GWD	Juegos ganados por remontada en 4to cuarto.
AV	Valor aproximado de cada jugador.

Cuadro 3.2: Variables de tabla de pases.

"QBRec" contiene el número de juegos ganados, perdidos y empatados (W-L-T) en la temporada especificada. La fórmula de yardas ajustadas por intento de pase (1), yardas netas por intento de pase (2) y yardas ajustadas netas por intento de pase (3), es la siguiente:

$$(1)\ \frac{Yds + 20 * TD - 45 * Int}{Att}$$

$$(2)\ \frac{Yds - Yds(sack)}{Att - Sk}$$

$$(3)\ \frac{Yds - Yds(sack) + (20 * TD) - (45 * Int)}{Att - Sk}$$

Figura 3.4: Tabla de pases de Brett Favre.

Passing * Selected to Pro Bowl, + First-Team All-Pro Share & more ▾ Glossary Toggle Per-Game Stats

Year	Age	Tm	Pos	No.	G	GS	QBrec	Cmp	Att	Cmp%	Yds	TD	TD%	Int	Int%	1D	Lng	Y/A	AY/A	Y/C	Y/G	Rate	QBR	Sk	Yds	NY/A	ANY/A	Sk%	4QC	GWD	AV
1991	22	ATL		4	2	0		0	4	0.0	0	0	0.0	2	50.0		0	0.0	-22.5		0.0	0.0		1	11	-2.20	-20.20	20.0			0
1992*	23	GNB	QB	4	15	13	8-5-0	302	471	64.1	3227	18	3.8	13	2.8		76	6.9	6.4	10.7	215.1	85.3		34	208	5.98	5.53	6.7	3	3	12
1993*	24	GNB	QB	4	16	16	9-7-0	318	522	60.9	3303	19	3.6	**24**	4.6		66	6.3	5.0	10.4	206.4	72.2		30	199	5.62	4.36	5.4	**3**	3	13
1994	25	GNB	QB	4	16	16	9-7-0	363	582	62.4	3882	33	5.7	14	2.4	200	49	6.7	6.7	10.7	242.6	90.7		31	188	6.03	6.08	5.1	1	2	16
1995*+	26	GNB	QB	4	16	16	11-5-0	359	570	63.0	**4413**	**38**	**6.7**	13	2.3	223	**99**	7.7	8.0	12.3	275.8	99.5		33	217	6.96	**7.25**	5.5	0	1	18
1996*+	27	GNB	QB	4	16	16	13-3-0	325	543	59.9	3899	**39**	**7.2**	13	2.4	194	80	7.2	7.5	12.0	243.7	95.8		40	241	6.27	6.61	6.9	1	1	17
1997*+	28	GNB	QB	4	16	16	13-3-0	304	513	59.3	3867	**35**	**6.8**	16	3.1	189	74	7.5	7.5	12.7	241.7	92.6		25	176	6.86	6.82	4.6			17
1998	29	GNB	QB	4	16	16	11-5-0	**347**	551	**63.0**	**4212**	31	5.6	23	4.2	204	84	7.6	6.9	12.1	263.3	87.8		38	223	6.77	6.07	6.5	1	2	16
1999	30	GNB	QB	4	16	16	8-8-0	341	**595**	57.3	4091	22	3.7	23	3.9	196	74	6.9	5.9	12.0	255.7	74.7		35	223	6.14	5.20	5.6	3	3	13
2000	31	GNB	QB	4	16	16	9-7-0	338	580	58.3	3812	20	3.4	16	2.8	194	67	6.6	6.0	11.3	238.3	78.0		33	236	5.83	5.31	5.4	0	4	12
2001*	32	GNB	QB	4	16	16	12-4-0	314	510	61.6	3921	32	6.3	15	2.9	187	67	7.7	7.6	12.5	245.1	94.1		22	151	7.09	7.02	4.1	1	2	15
2002*	33	GNB	QB	4	16	16	12-4-0	341	551	61.9	3658	27	4.9	16	2.9	189	85	6.6	6.3	10.7	228.6	85.6		26	188	6.01	5.70	4.5	3	3	13
2003*	34	GNB	QB	4	16	16	10-6-0	308	471	65.4	3361	**32**	**6.8**	21	4.5	168	66	7.1	6.5	10.9	210.1	90.4		19	137	6.58	5.96	3.9	1	3	14
2004	35	GNB	QB	4	16	16	10-6-0	346	540	64.1	4088	30	5.6	17	3.1	205	79	7.6	7.3	11.8	255.5	92.4		12	93	7.24	6.94	**2.2**	3	4	14
2005	36	GNB	QB	4	16	16	4-12-0	**372**	**607**	61.3	3881	20	3.3	**29**	4.8	202	59	6.4	4.9	10.4	242.6	70.9		24	170	5.88	4.45	3.8	1	1	9
2006	37	GNB	QB	4	16	16	8-8-0	343	**613**	56.0	3885	18	2.9	18	2.9	183	82	6.3	5.6	11.3	242.8	72.7	43.8	21	134	5.92	5.21	3.3	1	1	9
2007*	38	GNB	QB	4	16	16	13-3-0	356	535	66.5	4155	28	5.2	15	2.8	198	82	7.8	7.6	11.7	259.7	95.7	70.6	15	93	7.39	7.18	2.7	2	**4**	14
2008*	39	NYJ	QB	4	16	16	9-7-0	343	522	65.7	3472	22	4.2	**22**	4.2	186	56	6.7	5.6	10.1	217.0	81.0	46.0	30	213	5.90	4.91	5.4	1	2	12
2009*	40	MIN	QB	4	16	16	12-4-0	363	531	68.4	4202	33	6.2	7	**1.3**	214	63	7.9	8.6	11.6	262.6	107.2	75.6	34	247	7.00	7.61	6.0	2	2	16
2010	41	MIN	QB	4	13	13	5-8-0	217	358	60.6	2509	11	3.1	19	5.3	112	53	7.0	5.2	11.6	193.0	69.9	35.2	22	139	6.24	4.57	5.8	1	2	5
Career					**302**	**298**	**186-112-0**	**6300**	**10169**	**62.0**	**71838**	**508**	**5.0**	**336**	**3.3**	**3244**	**99**	**7.1**	**6.6**	**11.4**	**237.9**	**86.0**		**525**	**3487**	**6.39**	**5.93**	**4.9**	**28**	**43**	**255**
16 yrs		**GNB**			**255**	**253**	**160-93-0**	**5377**	**8754**	**61.4**	**61655**	**442**	**5.0**	**286**	**3.3**	**2732**	**99**	**7.0**	**6.6**	**11.5**	**241.8**	**85.8**		**438**	**2877**	**6.39**	**5.96**	**4.8**	**24**	**37**	**222**
2 yrs		**MIN**			**29**	**29**	**17-12-0**	**580**	**889**	**65.2**	**6711**	**44**	**4.9**	**26**	**2.9**	**326**	**63**	**7.5**	**7.2**	**11.6**	**231.4**	**92.2**		**56**	**386**	**6.69**	**6.39**	**5.9**	**3**	**4**	**21**
1 yr		**ATL**			**2**	**0**		**0**	**4**	**0.0**	**0**	**0**	**0.0**	**2**	**50.0**	**0**	**0**	**0.0**	**-22.5**		**0.0**	**0.0**		**1**	**11**	**-2.20**	**-20.20**	**20.0**			**0**

Fuente: www.pro-football-reference.com/players/F/FavBr00.htm

Las tablas de pases de postemporada y pases ajustados contienen las variables del cuadro 2.2, para los casos que corresponda. El caso de pases ajustados muestra aquellos mariscales de campo que se encuentran por encima o por debajo del promedio.

Como se muestra en la figura 2.3, en la sección de información general del jugador se encuentra el hipervínculo que direcciona a las estadísticas colegiales (*College Stats*), con la siguiente liga HTML, www.sports-reference.com/cfb/players. La cual contiene tablas de pases, recepciones y corridas y puntos agregados por año y totales. La tabla de pases, que será la única a extraer, contiene las variables mostradas en el cuadro 2.3 y la figura 2.5:

Variable	**Descripción**
Year	Año de temporada colegial.
School	Universidad del jugador.
Conf	Conferencia colegial.

Class	Categoría colegial.
Pos	Posición: QB.
G	Enfrentamientos jugados.
Cmp	Pases completados.
Att	Pases intentados.
Pct	Porcentaje de pases completados.
Yds	Yardas en pases.
Y/A	Yardas en pase por intento.
AY/A	Yardas de pase por intento ajustadas.
TD	Pases de anotación.
Int	Intercepciones.
Rate	Razón de eficiencia.

Cuadro 3.3: Variables de tabla de pases colegial.

La categoría colegial incluye las cuatro etapas por las que pasa un estudiante universitario en EE.UU. siendo "*freshman*" o primer año, "*sophomore*" o segundo año, "*junior*" o tercer año y "*senior*" o cuarto año. La razón de eficiencia se calcula de la siguiente manera:

$$\frac{8.4 * Yds + 330 * TD - 200 * Int + 100 * Cmp}{Att}$$

Figura 3.5: Tabla de pases colegial de Brett Favre.

Passing * indicates bowl stats included Share & more ▼ Glossary

						Passing								
Year	School	Conf	Class	Pos	G	Cmp	Att	Pct	Yds	Y/A	AY/A	TD	Int	Rate
1987	Southern Mississippi	Ind		QB	11	79	194	40.7	1264	6.5	5.0	15	13	107.6
1988	Southern Mississippi	Ind		QB	11	178	319	55.8	2271	7.1	7.4	16	5	129.0
1989	Southern Mississippi	Ind		QB	11	206	381	54.1	2588	6.8	6.3	14	10	118.0
1990	Southern Mississippi	Ind		QB	11	150	275	54.5	1572	5.7	5.2	7	6	106.6
Career	**Southern Mississippi**					**613**	**1169**	**52.4**	**7695**	**6.6**	**6.2**	**52**	**34**	**116.6**

Fuente: www.sports-reference.com/cfb/players/brett-favre-1.htm

La figura 2.5 muestra hipervínculos en las tablas de pase en las columnas de "*year*", "*school*" y "*Conf*". La dirección en la columna de universidad dirige a la temporada colegial del año especificado en la columna "Year", mediante la liga https://www.sports-reference.com/cfb/schools/southern-mississippi/1987.html, en este caso el año 1987. Con la liga https://www.sports-reference.com/cfb/schools/southern-mississippi se obtienen las estadísticas de todas las temporadas que ha jugado la universidad, mediante una tabla que es detalla en el cuadro 2.4 y la figura 2.6:

Variable	**Descripción**
Rk	Posición de observación en tabla.
Year	Año en que se jugó la temporada.
Conf	Conferencia a la que pertenece.
W	Partidos ganados.
L	Juegos perdidos.
T	Enfrentamientos empatados.
Pct.	Porcentaje ganados-perdidos.
SRS[38]	Sistema simple de rating.
SOS[39]	Fortaleza del calendario de juego.
AP Pre	Posición en la encuesta de pretemporada.
AP High	Posición más alta durante temporada.
AP Post	Posición final de encuesta.
Coach(es)	Entrenador.
Bowl	Tazón jugado y especificado.
Notes	Notas.

Cuadro 3.4: Variables de temporadas de fútbol colegial.

SRS toma en consideración los puntos promedios y la fortaleza del calendario, donde cero es el promedio y puede adoptar valores negativos en caso que sea menor al promedio y positivo si es mayor al promedio. Al igual que SRS, SOS muestra el promedio en cero.

[38] Del inglés, "*Simple Rating System*"
[39] Del inglés, "*Strength of Schedule*"

Figura 3.6: Tabla de estadísticas de escuela del sur de Mississippi Águilas Doradas.

57 Years 57 Players Share & more ▾ Glossary

Rk	Year	Conf	W	L	T	Pct	SRS	SOS	AP Pre	AP High	AP Post	Coach(es)	Bowl	Notes
1	2019	CUSA	7	6	0	.538	-2.90	-3.67				Jay Hopson (7-6)	Armed Forces Bowl-**L**	
2	2018	CUSA	6	5	0	.545	-4.18	-7.36				Jay Hopson (6-5)		
3	2017	CUSA	8	5	0	.615	-5.49	-8.18				Jay Hopson (8-5)	Independence Bowl-**L**	
4	2016	CUSA	7	6	0	.538	-3.73	-5.04				Jay Hopson (7-6)	New Orleans Bowl-**W**	
5	2015	CUSA	9	5	0	.643	2.13	-6.22				Todd Monken (9-5)	Heart of Dallas Bowl-**L**	
6	2014	CUSA	3	9	0	.250	-10.91	0.17				Todd Monken (3-9)		
7	2013	CUSA	1	11	0	.083	-21.13	-4.55				Todd Monken (1-11)		
8	2012	CUSA	0	12	0	.000	-18.04	-1.29				Ellis Johnson (0-12)		
9	2011	CUSA	12	2	0	.857	9.18	-4.60		20	20	Larry Fedora (12-2)	Hawaii Bowl-**W**	
10	2010	CUSA	8	5	0	.615	1.79	-4.68				Larry Fedora (8-5)	Beef O'Brady's Bowl-**L**	
11	2009	CUSA	7	6	0	.538	-0.67	-4.36				Larry Fedora (7-6)	New Orleans Bowl-**L**	
12	2008	CUSA	7	6	0	.538	0.46	-3.24				Larry Fedora (7-6)	New Orleans Bowl-**W**	
13	2007	CUSA	7	6	0	.538	-2.09	-4.47				Jeff Bower (7-6)	PapaJohns.com Bowl-**L**	
14	2006	CUSA	9	5	0	.643	3.65	-1.49				Jeff Bower (9-5)	GMAC Bowl-**W**	
15	2005	CUSA	7	5	0	.583	1.26	-4.49				Jeff Bower (7-5)	New Orleans Bowl-**W**	
16	2004	CUSA	7	5	0	.583	-1.09	-1.25		21		Jeff Bower (7-5)	New Orleans Bowl-**W**	
17	2003	CUSA	9	4	0	.692	3.58	-1.12				Jeff Bower (9-4)	Liberty Bowl-**L**	
18	2002	CUSA	7	6	0	.538	-0.24	-1.93				Jeff Bower (7-6)	Houston Bowl-**L**	
19	2001	CUSA	6	5	0	.545	0.84	-2.71				Jeff Bower (6-5)		
20	2000	CUSA	8	4	0	.667	7.44	-0.14	23	13		Jeff Bower (8-4)	Mobile Alabama Bowl-**W**	
Rk	**Year**	**Conf**	**W**	**L**	**T**	**Pct**	**SRS**	**SOS**	**AP Pre**	**AP High**	**AP Post**	**Coach(es)**	**Bowl**	**Notes**
21	1999	CUSA	9	3	0	.750	9.75	1.00		14	14	Jeff Bower (9-3)	Liberty Bowl-**W**	
22	1998	CUSA	7	5	0	.583	6.34	-0.16	21	21		Jeff Bower (7-5)	Humanitarian Bowl-**L**	

Fuente: www.sports-reference.com/cfb/schools/southern-mississippi.

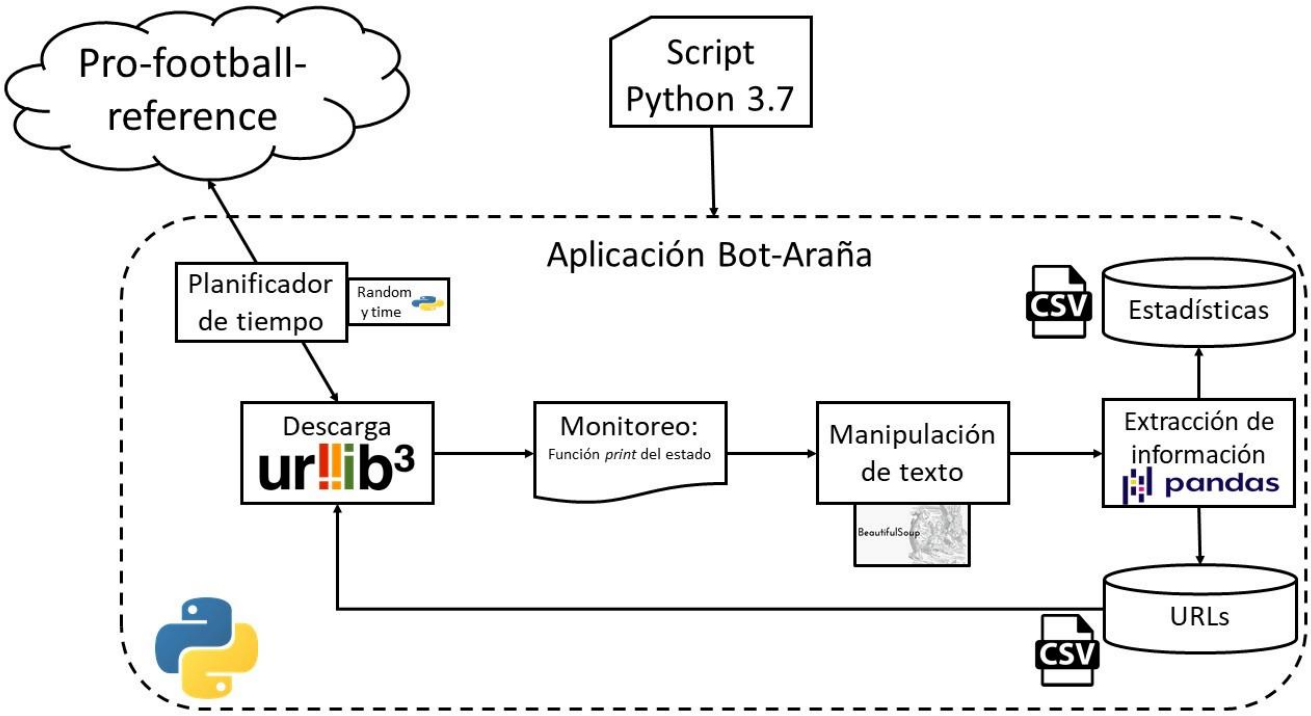
El método de consulta y extracción utilizados es un script ELT de programación funcional bajo el administrador de ambientes Anaconda en Python 3.7.6 y gestionado en Git, un sistema distribuido de control de versiones, mostrado en el apéndice A para su reproductibilidad. Utilizando las librerías *BeautifulSoup, random, time, urllib3* y *pandas*[40]. La primera es una librería para extraer datos en formato HTML y XML para navegar, buscar y modificar arboles estructurados. Las siguientes dos son paquetes que pertenecen a la librería estándar de Python, la primera genera números pseudoaleatorios, la segunda permite acceder a objetos de tiempo y su manipulación; la cuarta librería es un cliente HTTP para Python, la última librería se encarga de la manipulación y análisis de datos. HTML[41] es un lenguaje estructurado en árbol, que permite la creación de páginas web que se conectan entre ellas. La estructura del lenguaje es dada por las marcas y etiquetas para la identificación de elementos, que permitirá

[40] Por su orden de aparición: www.crummy.com/software/BeautifulSoup/bs4/doc/, www.docs.python.org/3/library/random.html, www.docs.python.org/3/library/time.html, www.urllib3.readthedocs.io/en/latest/, www.pandas.pydata.org/

[41] Del inglés, "Hypertext Markup Language".

que las búsquedas sean más rápidas y los formatos de páginas se puedan replicar[42]. Dichas características habilitan la posibilidad de extraer información de manera automatizada por la facilidad de búsqueda de ciertos elementos a extraer.

Figura 3.7: Estructura propuesta de aplicación bot-araña.



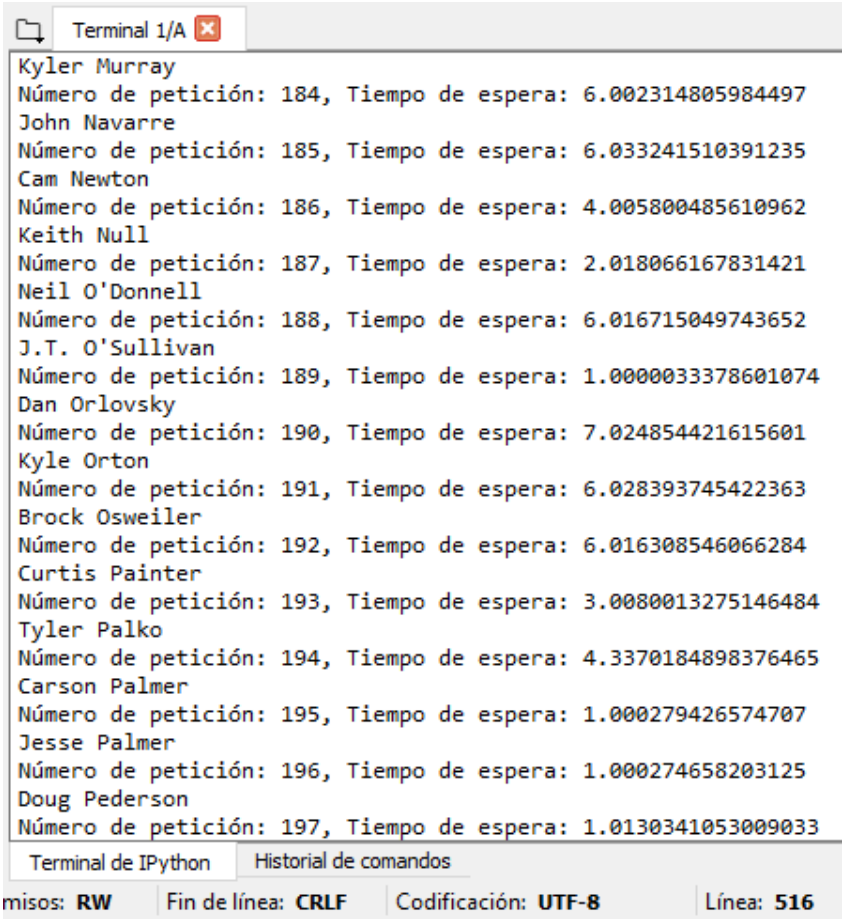
Fuente: Elaboración propia.

La Figura 2.7 muestra el flujo general propuesto de la aplicación Bot-Araña y su aplicación en Python, donde se detallan las librerías utilizadas en el script y sus funciones. La aplicación Bot-Araña fue programada utilizando el paradigma de programación funcional, donde además de crear un "*web crawler*" limitado y definido, el cual extrae y almacena las ligas necesarias para continuar con el flujo de extracción de ligas, esta incluye la extracción de información estructurada y semiestructurada de estadísticas profesionales y colegiales de mariscales de campo. La librería urllib3 permite acceder y descargar el archivo HTML, mediante peticiones HTTP, al proveer la liga que se busca descargar. HTTP es un protocolo para compartir archivos HTML o hipertextos. Para que el cliente (la aplicación) no sature de peticiones el servidor y como consecuencia ser expulsados, se usa una combinación de la paquetería "*random*" y

[42] https://developer.mozilla.org/es/docs/Web/HTML

"*time*" para pausar las consultas con un intervalo pseudoaleatorio, de un segundo a siete segundos que sigue una distribución normal. Posteriormente se imprime el estado de la aplicación, donde se detalla el tiempo de espera asignado, el número de peticiones y el jugador en proceso de descarga, como se muestra en la figura 2.8.

Figura 3.8: Ejemplo de estado de la aplicación.

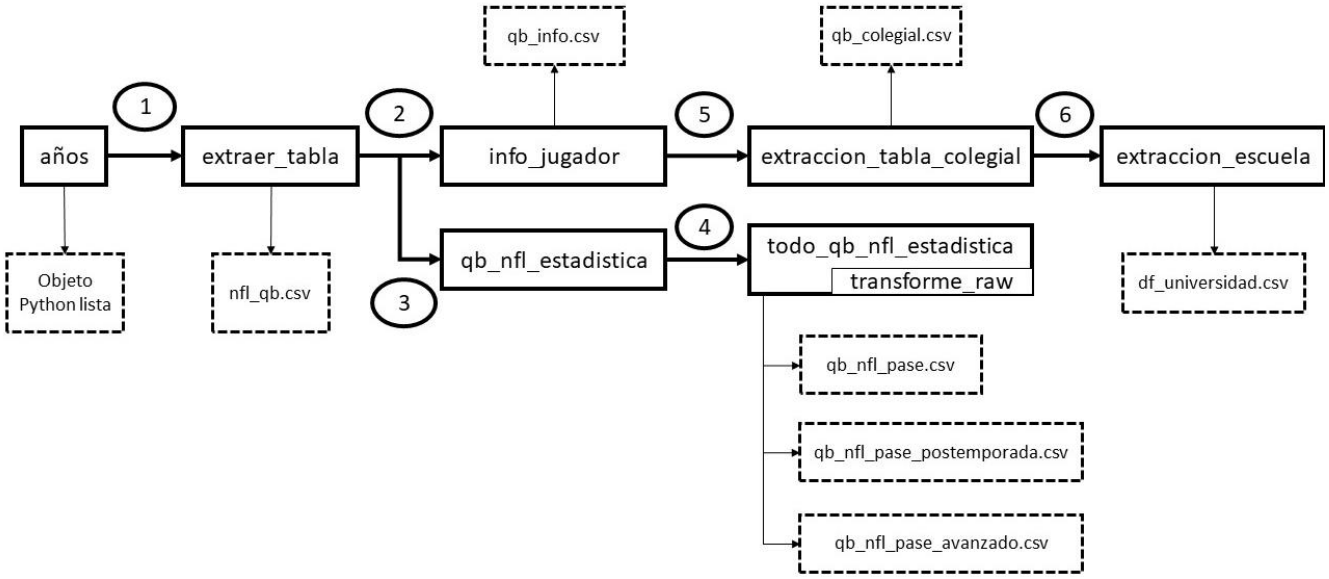


Fuente: Elaboración propia.

El apéndice C muestra un ejemplo de la estructura de las páginas de pro-football-reference, las cuales fueron manipuladas por la aplicación Bot-Araña mediante la librería *BeautifulSoup* utilizando sus funciones de búsqueda para encontrar las etiquetas especificadas de la página web, permitiendo agilizar la búsqueda y automatizar el proceso para la página de cada jugador. Donde se puede apreciar, la importancia de las etiquetas para referencias, pero también la poca capacidad de un humano para procesar dicha información.

Para obtener los datos estructurados de las páginas web del jugador se utilizaron las etiquetas de tabla HTML, aislando su estructura tabular y asignando un valor de identificación con autoincremento para la fácil anexión de datos en el proceso de trasformación. La posesión de la estructura de tabla en HTML permite la lectura y manipulación del objeto como un "dataframe" para ser manipulada por pandas.

Figura 3.9: Flujo de funciones de aplicación Bot-Araña.



Fuente: Elaboración propia.

La figura 2.9 muestra el flujo de las funciones de la aplicación bot-araña, detalladas a continuación. La función “años” de la aplicación bot-araña escribe en memoria las ligas semillas necesarias para iniciar la descarga de la página de pases por año y se usará como entrada de la función “extraer_tabla”, que regresa como salida un objeto-lista, teniendo en cada elemento un objeto pandas arreglo de datos. Dicho objeto es concatenado en una sola tabla y descargado en archivo separado por comas como “nfl_qb.csv”. La última tabla servirá como entrada de la función “qb_nfl_estadistica”, la cual resuelve problemas particulares, mediante condicionales “if-else”. Para la extracción de la tabla de pases, pases en postemporada y pases avanzados se concatenarán y descargarán con la función “todo_qb_nfl_estadistica”, además planifica el tiempo e imprime el monitoreo de control. La función “info_jugador”, almacena en un objeto lista las variables del cuadro 2.1. Las tablas anteriores serán cargadas en disco con los nombres “qb_nfl_pase.csv”, “qb_nfl_pase_postemporada.csv”, “qb_nfl_pase_avanzado.csv” y “qb_info.csv”. La función “extracción_tabla_colegial” descarga los datos de pases colegiales de todos los jugadores que aparecen en la tabla de información del jugador, con planificador de tiempo e impresión de control. La función “extracción_escuela” usa como entrada la salida de la función “extracción_tabla_colegial” y descarga los datos de todas las universidades implicadas.

3.2 Modelo Extracción-Carga-Transformación

En esta sección se detalla la transformación de variables que fue realizada como predecesora de la creación del modelo.

De acuerdo a lo detallado en el apartado 1.2.1 Almacenamiento de datos y Big Data, el clásico modelo ETL se caracteriza por una carga de datos que requiere de una cierta forma, además, demanda que la transformación se realice en un mismo proceso finito y que los datos de entrada sean estructurados y completos. Por lo anterior, este modelo no funciona para datos web y es necesario un modelo ELT o extracción-carga-transformación. En el cual, los esfuerzos se focalizan en la extracción y la carga para asegurar las características Big Data de velocidad, variedad y volumen y posteriormente concentrase en la veracidad y valor de los datos, que será reforzada en el análisis exploratorio. La extracción de datos web con la aplicación Bot-Araña fue detallada en la sección anterior. Para asegurar un solo proceso bajo el modelo ELT, se construyó todo el flujo en un solo script. Para la automatización de modelos ELT es necesario de un proceso iterativo de depuración[43] de código. Por lo cual, la sección de transformación preliminar fue realizado en "*cuadernos jupyter*", que permiten un ambiente visual por celdas y posteriormente añadidos al script único, teniendo como resultado, la posibilidad de automatizar el proceso de extracción y transformación de datos que sirvan como entrada de modelos de inteligencia de máquina ya entrenados.

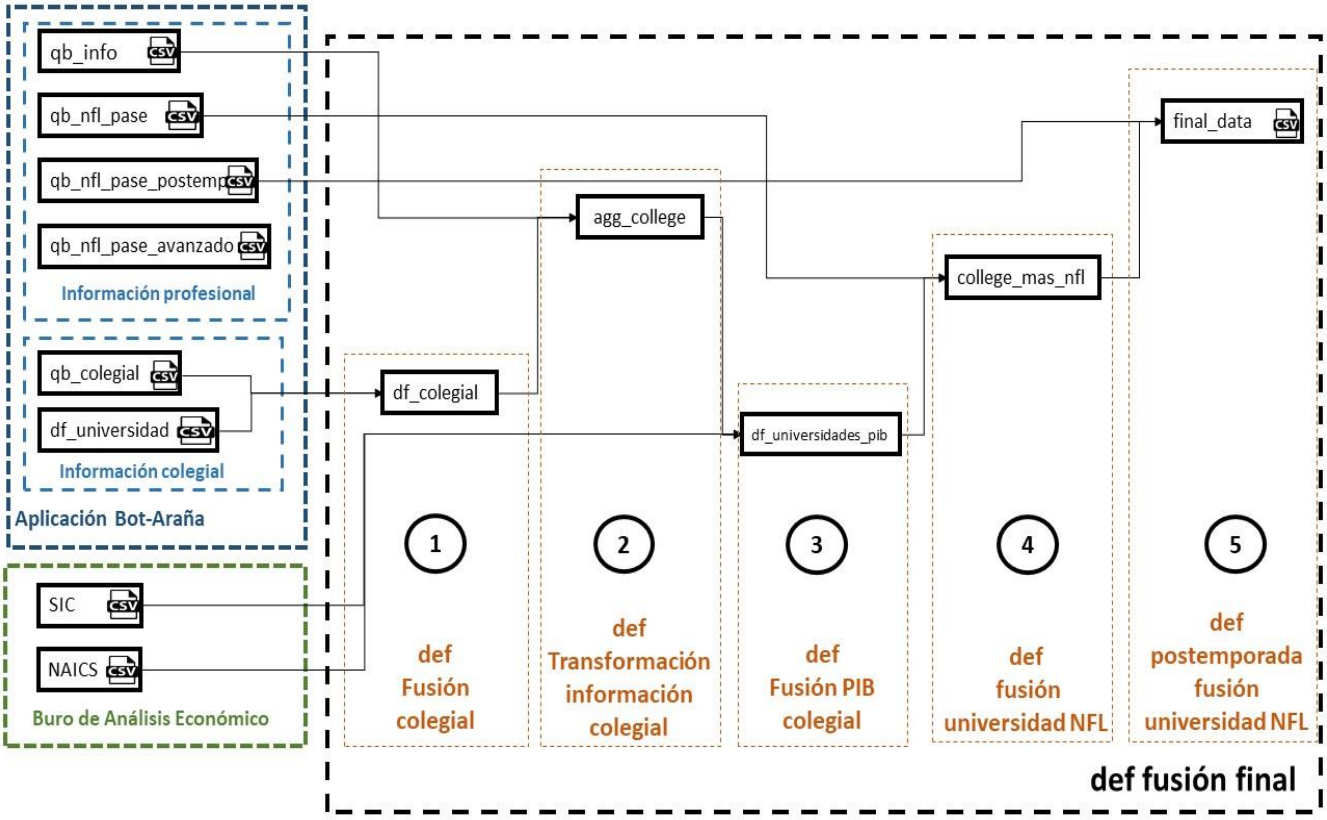
Para asegurar la variedad de datos, la agregación de valor y bajo el intento de hacer más robusto el modelo machine learning, se extrajeron datos del Producto Interno Bruto en dólares actuales por estado de EE. UU. con granularidad anual (SAGDP2). Dicha información se obtuvo de manera manual del buro de análisis económico del departamento de comercio de los Estados Unidos de Norte América[44]. Dónde SIC es la información de 1963 a 1997 y NAICS es la información de 1997 a 2019. A pesar de que existe una API del buró de análisis económico para la extracción automática de los datos, se optó por la descarga manual, debido al intervalo de tiempo en que se realiza la obtención de datos, sumado a lo fácil que es descargar dichos datos agregados en solo dos archivos separados por comas. Dichos archivos requieren de un proceso de verificación de nomenclatura de los estados para poder hacer una

[43] Comúnmente conocido por su versión en inglés, "*debugging*".
[44] https://apps.bea.gov/itable/iTable.cfm?ReqID=70&step=1#reqid=70&step=1&isuri=1

unión efectiva con la tabla de información de los jugadores, así como almacenarlos en un espacio específico.

Como se muestra en la figura 2.7, el proceso que sigue al almacenamiento en memoria de un objeto tipo *pandas dataframe* es almacenar o cargar este en disco local en formato separado por comas, esta tarea será el último paso a realizar de la aplicación bot-araña, como se muestra en el apéndice B, los archivos creados en el proceso ELT y la importancia de mantener los archivos separados por coma en una sola carpeta, para propósitos de reproductibilidad. La Figura 2.10 muestra el proceso de transformación de datos, en el cual se destaca la importancia del orden de los pasos. Iniciando por la función "*fusión colegial*", en la figura bajo el nombre "def" al ser la sintaxis para definir una función en Python. Dicha función tiene por objetivo anexar la información colegial de los jugadores y la información de las temporadas universitarias en las que jugaron. La siguiente función, "transformación información colegial", recibe como entrada la salida de la función anterior más la tabla de características de la información profesional y se encarga de asignar las características del jugador a su información colegial de pases y resultados de temporada. Una vez realizado este paso, "fusión PIB colegial" se enfrenta a la anexión de los datos del producto interno Bruto, pivoteando la información SIC y NAICS para la creación de una columna de estados y que la unión con el primer año de universidad del jugador y la región del bachillerato coincidan y se le pueda asignar un valor a cada jugador. La función número cuatro busca unir las tablas mediante el "Id" para tener la información ex post o profesional de los pases en temporada regular del jugador colegial. El último paso tiene por objetivo unir la información de los pases en postemporada con el arreglo de datos proveniente de la función "fusión universidad NFL". Todas las funciones anteriormente mencionadas están anidadas a la función "fusión final", la cual se asegura del correcto flujo por pasos.

Figura 3.10: Flujo de funciones de transformación.



Fuente: Elaboración propia.

La tabla final cuenta con trescientas tres observaciones, cada una pertenece a un jugador distinto. El cuadro 2.5 muestra las ochenta y dos variables del arreglo de salida de la función “fusión final” que será el último paso del proceso ELT, la cual se divide en dos, información colegial e información profesional a nivel jugador, de la última tenemos información en temporada regular y en postemporada, en caso de que exista. Si la variable pertenece a los datos de la NFL, el nombre de la variable contendrá “*nfl_...*” al principio y se encuentran al final de la tabla. El arreglo de datos de pases colegial, “qb_colegial”, contiene en cada observación la información de pases colegiales de cada temporada por cada mariscal de campo extraído, mientras que “final_data” contiene en cada observación un jugador, por lo que se agrupó dicha información a nivel de jugador. Teniendo como resultado que fuera necesario transformar los datos de su trayectoria colegial, que comúnmente tiene una duración de tres a cuatro años, a variables que nos permitan entender dicha trayectoria. Por ejemplo, el número de juegos totales con participación, en contraste con el número de juegos por temporada. Otro ejemplo puede ser el mínimo de enfrentamientos jugados en una temporada, la media de pases por

temporada, etc. Sin embargo, estas estadísticas no permiten conocer la evolución del jugador durante su experiencia en el fútbol americano colegial, por lo que variables como la diferencia entre el mínimo y el máximo de los valores en temporada, la desviación estándar del número de anotaciones o la pendiente que ajuste los valores mediante mínimos cuadrados, la cual permite concluir si su trayectoria fue en declive o por el contrario fue mejorando a través de las temporadas colegiales.

Las estadísticas de agregación usadas se pueden dividir en 4, donde $T = [t_1, t_{2,\dots,} t_n]$ vector de tamaño n. $\perp t_i$ desde $i = 1$ hasta n. Siendo t la observación de cada temporada de los jugadores, teniendo i como la temporada 1 y n como la última temporada colegial.

- Valor mínimo en temporada colegial:

$$\min_{t \in T} T$$

- Valor máximo en temporada colegial:

$$\max_{t \in T} T$$

- Media de la diferencia de valores entre temporadas colegiales:

$$\frac{1}{n-1} \sum_{i=1} (t_{i+1} - t_i)$$

- Coeficiente de la pendiente ajustada por mínimos cuadrados de los valores del vector T. Se eligió la pendiente sobre el coeficiente de correlación, ya que la última muestra la relación entre dos variables y la primera muestra como es esta relación y el tamaño de su coeficiente nos permite conocer la dimensión de sus cambios y avances ya sea de manera positiva o negativa. Donde los años x son la variable independiente y los valores del vector T la variable dependiente.

Variable	**Descripción**
Id	Identificador único del jugador.
Nombre	Nombre del jugador.
años_colegial	Años de experiencia en fútbol colegial.

universidad_distinta	Número de universidades distintas jugando fútbol.
juegos_totales	Número total de enfrentamientos colegiales jugados.
min_juegos	Mínimo de enfrentamientos jugados en una temporada colegial.
max_juegos	Máximo de enfrentamientos jugados en una temporada colegial.
dif_juegos_media	Media de la diferencia entre temporadas de juegos colegiales.
pendiente_juego	Pendiente del número de enfrentamientos.
suma_cmp	Número total de pases colegiales completados.
min_cmp	Mínimo de pases completados en una temporada colegial.
max_cmp	Máximo de pases completados en una temporada colegial.
dif_cmp_media	Media de la diferencia de pases colegiales completados entre temporadas.
pendiente_cmp	Pendiente del número de pases colegiales completados.
suma_intentos	Número total de pases colegiales intentados.
min_intentos	Mínimo de intento de pases en una temporada colegial.
max_intentos	Máximo de intento de pases en una temporada colegial.
dif_intentos_media	Media de la diferencia de intentos de pase colegiales entre temporada.
pendiente_intentos	Pendiente del número de intentos de pases colegiales.

max_por	Porcentaje máximo de pases colegiales completados.
min_por	Porcentaje mínimo de pases colegiales completados.
media_por	Media de la diferencia de porcentaje de pase colegiales completados entre temporada.
desv_por	Desviación estándar de la diferencia del porcentaje de pases colegiales completados entre temporada.
total_yds	Suma total de yardas colegiales.
min_yds	Mínimo de yardas por pase colegiales en una temporada.
max_yds	Máximo de yardas por pase colegiales en una temporada.
dif_yds_media	Media de la diferencia de yardas colegiales por pase entre temporadas.
pendiente_yds	Pendiente del número de yardas de pases colegiales.
total_td	Suma de anotaciones colegiales totales.
min_td	Mínimo de anotaciones colegiales en una temporada.
max_td	Máximo de anotaciones colegiales en una temporada.
dif_td_media	Media de la diferencia de anotaciones colegiales entre temporadas.
pendiente_td	Pendiente del número de anotaciones colegiales.
total_int	Número total de intercepciones colegiales.
min_int	Número mínimo de intercepciones colegiales en una temporada.

max_int	Número máximo de intercepciones colegiales en una temporada.
dif_int_media	Media de la diferencia de intercepciones colegiales entre temporadas.
pendiente_int	Pendiente del número de intercepciones colegiales.
rate_q1	Cuartil 1 de la razón de eficiencia colegial.
rate_q2	Mediana de la razón de eficiencia colegial.
rate_q3	Cuartil 3 de la razón de eficiencia colegial.
max_yds_intentos	Máximo de yardas colegiales por intento.
min_yds_intentos	Mínimo de yardas colegiales por intento.
media_yds_intentos	Media de yardas colegiales por intento.
desv_yds_intentos	Desviación estándar de yardas colegiales por intento.
min_año	Primer año de fútbol colegial.
bowls_ganados	Número de tazones colegiales ganados.
bowls_jugados	Número de tazones colegiales jugados.
juegos_ganados	Número de juegos colegiales ganados.
juegos_perdidos	Número de juegos colegiales perdidos.
pct_juegos	Media del Porcentaje de juegos colegiales ganados.
pct_juegos_desv	Desviación estándar del porcentaje de juegos colegiales ganados.
Throws	Extremidad con la que lanza el balón.
Height	Altura del jugador.
Weight	Peso del jugador.
state_high_school	Región del bachillerato.
state_ab	Región de la universidad.
Year	Año de tabla PIB.
Gdp	Producto interno bruto del estado del bachillerato.

nfl_tot_juegos	Juegos Totales en los primero cuatro años profesionales.
nfl_primeros_juegos	Juegos totales en la primera temporada.
nfl_tot_juegos_iniciados	Enfrentamientos iniciados totales en las primeras cuatro temporadas profesionales.
nfl_primeros_jug_vs_iniciados	Número de juegos no iniciados en la primera temporada.
nfl_tot_cmp	Número de pases completados en las primeras cuatro temporadas profesionales.
nfl_primero_cmp	Número total de pases completados en la primera temporada.
nfl_tot_intentos	Número de pases intentados en las primeras cuatro temporadas profesionales.
nfl_primero_intentos	Número total de pases intentados en la primera temporada.
nfl_tot_jugados_vs_iniciados	Número de juegos no iniciados en las primeras cuatro temporadas.
nfl_tot_td	Número total de anotaciones en las primeras cuatro temporadas.
nfl_primero_td	Número total de anotaciones en la primera temporada.
nfl_tot_yds	Número total de yardas por pase en las primeras cuatro temporadas profesionales.
nfl_primeros_yds	Número total de yardas por pase en la primera temporada profesional.
Draft	Posición numérica en el draft.
año_draft	Año de ingreso draft.
nfl_primer_año	Primer año de juego profesional.
nfl_primero_ganado	Número total de juegos ganados en la primera temporada profesional.
nfl_primero_perdido	Número total de juegos perdidos en la primera temporada profesional.

nfl_total_ganado	Número total de juegos ganados en las primeras cuatro temporadas profesionales.
nfl_total_perdido	Número total de juegos perdidos en la primera temporada profesional.
('Year', 'count')	Número total de postemporadas en su trayectoria profesional.
('Year', min')	Primer año en postemporada profesional.
('tot_postemporada', 'sum')	Postemporadas totales en los primeros 4 años profesionales.

Cuadro 3.5: Variables de la tabla final del flujo ELT propuesto.

3.3 Flujo de ciencia de datos

En esta sección se busca reproducir los pasos de la figura 1.3 del apartado de flujo de trabajo en ciencia de datos. Donde los objetivos y el conocimiento del problema fueron detallados previamente, a lo largo de este trabajo. Además, en las dos secciones anteriores se detalló la creación del arreglo de datos. El aparatado de análisis exploratorio, preprocesamiento y modelado de datos detallaran el proceso iterativo y continuo que permitirá obtener, primero, un arreglo capaz de servir como entrada a cualquier algoritmo de inteligencia computacional y, segundo, un modelo capaz de predecir de manera eficaz el desempeño de un mariscal de campo. El EDA, además, presentará de manera gráfica los datos, tratando de entender de modo analítico el estado actual estudiado. El preprocesamiento y la creación de modelos de datos trabajan a la par en la búsqueda de un modelo que ajuste de manera adecuada los datos, a través de un proceso de pruebas, donde se apelará por la obtención del menor error en el arreglo de prueba. El preprocesamiento de datos buscará la mejor forma de manejar valores atípicos, normalizar atributos y eliminar variables en un ciclo de entrenamiento y prueba de modelos.

3.3.1 Análisis de datos exploratorio

El análisis de datos exploratorio fue realizado en “jupyter” laboratorio (lab) donde se junta el código Python y Markdown para un claro flujo del proceso de entendimiento, como se muestra en el apéndice D, apreciando la cantidad de código que se destina en dichas tareas y a constante repetición de estructura de código. Utilizando las librerías de pandas, matplotlib y

scipy[45]. La segunda es una librería para crear visualizaciones y la tercera es un ecosistema matemático y científico para análisis de datos. El objetivo del EDA será explorar las estadísticas descriptivas de las variables, junto con sus distribuciones y representaciones gráficas, si se considera pertinente. Además, se conocerán las relaciones entre variables, buscando encontrar patrones y la detección de relación con la variable objetivo. El EDA es el medio usado por el científico de datos para entender mejor el arreglo y la estructura de datos y realizar una adecuada selección de modelo. El análisis no se utilizará para comunicar resultados, a menos que se esté trabajando en un equipo de desarrollo ágil y se busque mejorar la transparencia del equipo. Como último paso del flujo de ciencia de datos se debe comunicar los resultados, tanto los analíticos (análisis de datos exploratorio) como los datos de la construcción del modelo.

Primero se revisó la variable "*id*", asegurando que el número distinto de valores en la misma fuera igual a la cuenta de observaciones. Para comprender que variables profesionales son relevantes y cómo fue su trayectoria colegial, se utilizó la lista de Elliot Harrison (2019) y de Mike Florio (2019) a los que se hará referencia como élite:

- Warren Moon
- Ben Roethlisberger
- Russell Wilson
- Steve Young
- Troy Aikman
- Brett Favre
- Aaron Rodgers
- Dan Marino
- Drew Brees
- Peyton Manning
- Tom Brady
- Cam Newton
- Matt Ryan
- Andrew Luck

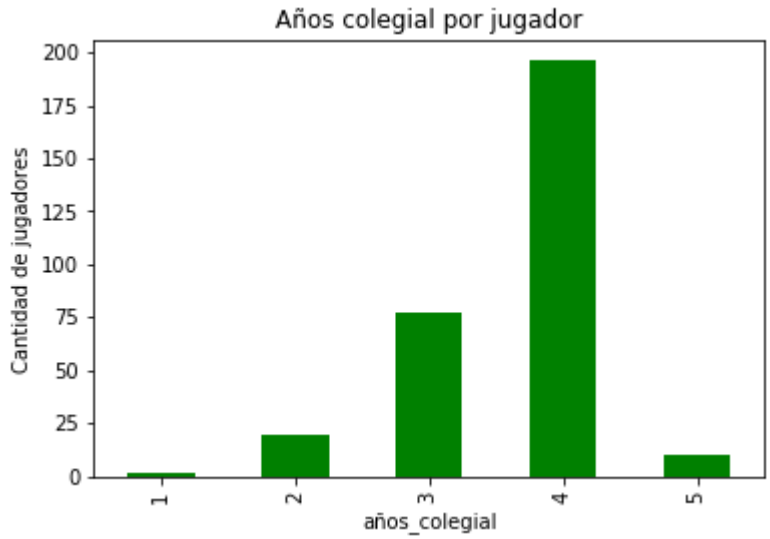
[45] https://matplotlib.org/ y https://www.scipy.org/ respectivamente.

- Philip Rivers

Se revisó el tiempo que pasan los jugadores en la universidad, como se puede ver en la Figura 2.11, se encontró que los datos de entrada propuestos para el modelo son, en su mayoría, jugadores con tres a cuatro años de experiencia colegial. 280 de los jugadores en el arreglo asistieron a una sola universidad, mientras que los 23 restantes asistieron a dos universidades. Nueve de los quince mariscales de campo élite tuvieron cuatro años de experiencia colegial, uno de ellos dos años y los restantes tres años de trayectoria. Tres de los mariscales de campo élite asistieron a 2 universidades distintas. 94% de los jugadores lanzan con la mano derecha, dentro del conjunto élite solo uno lanza con la mano izquierda.

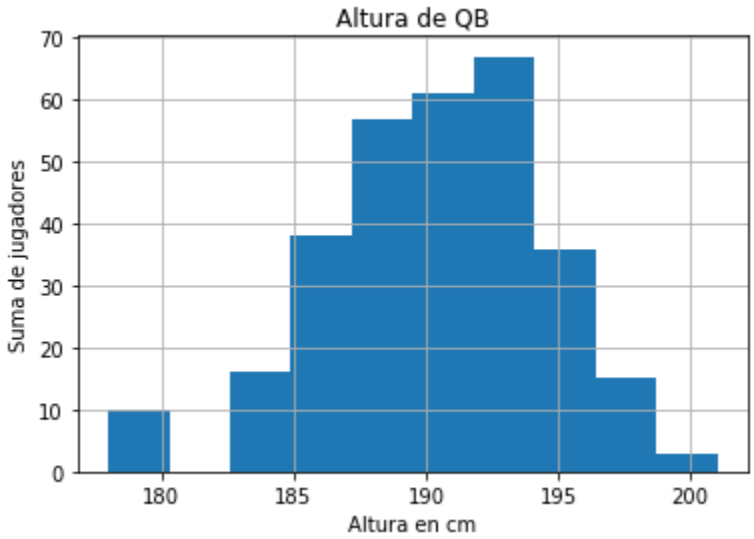
La figura 2.12 y 2.13 muestran la altura y el peso de los mariscales de campo respectivamente. Ambas distribuciones tienen un patrón simétrico. Todos los QB del conjunto élite miden más de 180cm. 60% del mismo conjunto miden más de 190 cm y ninguno supera los 200cm de altura. De los mariscales de campo élite todos superan los 94kg de peso y 53% pasan los 100kg, no obstante, todos se mantienen por debajo de los 110kg. Concluyendo que un QB que aspira a ser élite no tiene características físicas (peso y altura) que contribuyan a pertenecer a esta lista.

Figura 3.11: Cuenta de la experiencia universitaria de QB.



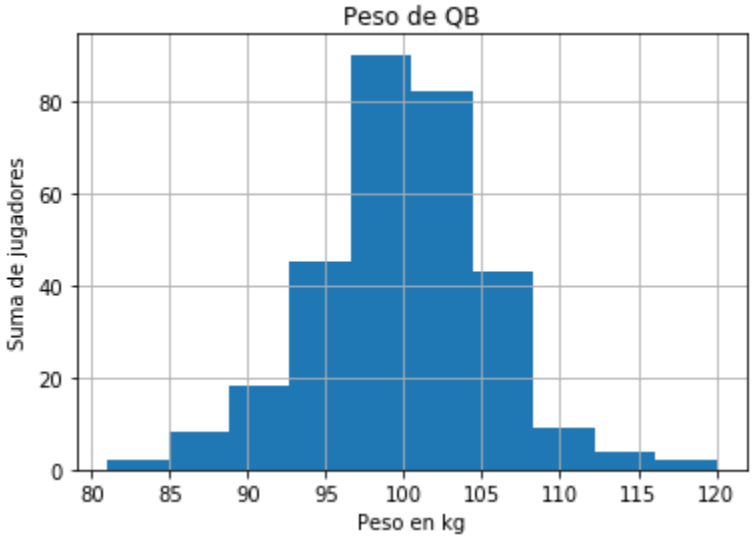
Fuente: Elaboración propia.

Figura 3.12: Histograma de altura del QB.



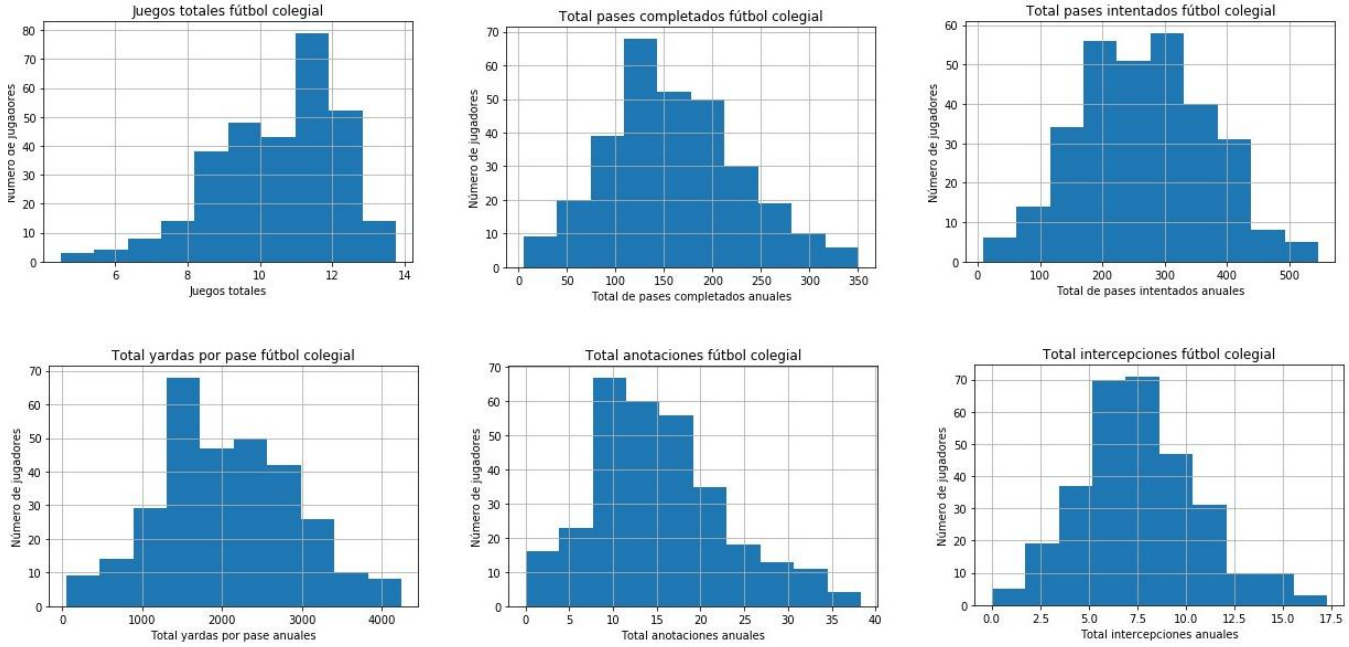
Fuente: Elaboración propia.

Figura 3.13: Histograma de pesos de QB.



Fuente: Elaboración propia.

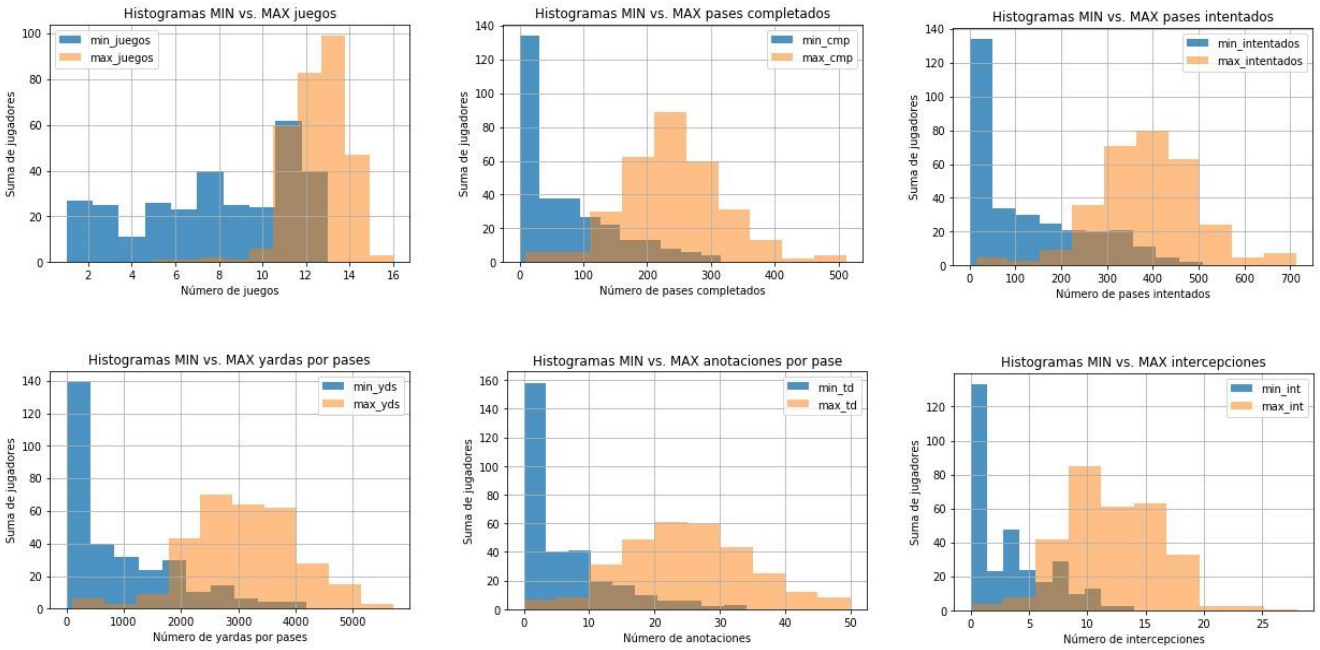
Figura 3.14: Histograma de valores agregados de QB.



Fuente: Elaboración propia.

La figura 2.14 muestra los histogramas de los valores agregados de los mariscales de campo a nivel anual de los juegos, pases completados, pases intentados, yardas por pases, anotaciones e intercepciones. La figura 2.15 muestra las distribuciones comparadas mínimo contra máximo utilizando las mismas variables de la figura 2.14. Mostrando Sesgos positivos en todas las variables, excepto la de juegos, pudiendo concluir que la trayectoria colegial de los jugadores no es estática, pero, por el contrario, tiene altos y bajos y se espera que los mínimos se encuentren en el primer año. Mientras que los máximos se distribuyen más simétricamente. Estas figuras servirán para la decisión de transformaciones en los atributos, bajo el intento de mejora de las variables predictivas en el modelo. Como complemento de la figura 2.15 se debe utilizar las variables de pendientes detalladas en el cuadro 2.6.

Figura 3.15: Histogramas de valores comparados mínimo y máximo de QB.



Fuente: Elaboración propia.

	Min	Cuartil 1	Media	Mediana	Cuartil 3	Max	Desv. Est.
Juegos	-11	0	2.97	2.05	5.85	13	3.77
Completados	-198	74.4	141.56	136.8	196.05	465.3	99.15
Intentados	-272	98.93	217.56	219.2	318.23	652	156.61
Yardas	-2559	959.1	1836.65	1850.5	2688.83	5455.8	1243.29
Anotaciones	-34	6.9	14.64	14.4	21.6	49.71	11.36
Intercepciones	-11.4	1	4.65	4.65	8.7	18	5.59

Cuadro 3.6: Estadísticas descriptivas de pendientes QB.

Donde se muestra que más del 75% de las observaciones tienen una trayectoria que mejora gradualmente a través de los años en todas las variables del cuadro. Además, se destaca la similitud de todas las variables entre su media y mediana, junto con la distancia entre el cuartil uno y la mediana y esta y el cuartil tres. La variable de la media de las diferencias entre temporada y la pendiente son atributos que pretenden explicar lo mismo, el cuadro 2.7 muestra la correlación entre ellas.

Variable	Correlación
Juegos	0.91
Completados	0.85
Intentados	0.85
Yardas	0.84
Anotaciones	0.86
Intercepciones	0.89

Cuadro 3.7: Correlación entre pendiente y media de las diferencias.

Encontrando que estas variables están correlacionadas debido, en parte, a la naturaleza de la trayectoria. La relación entre los atributos de trayectoria se realizará utilizando las variables de pendiente.

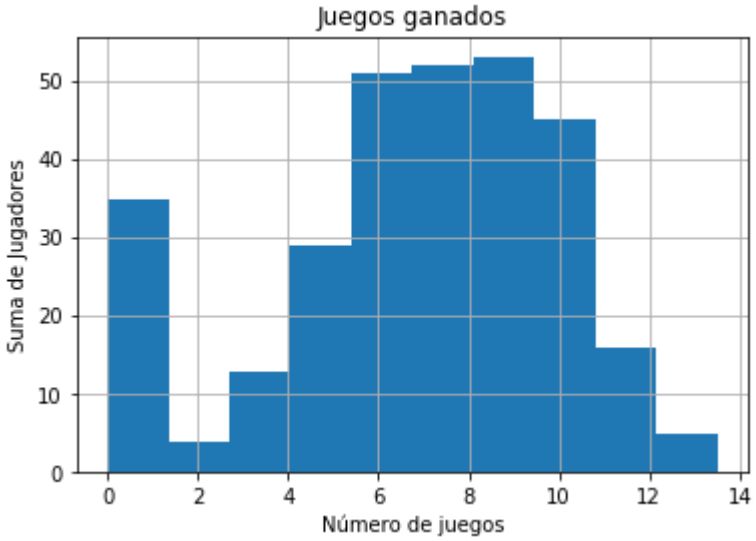
Figura 3.16: Mapa de calor correlacional de las variables de pendiente.

	pendiente_juego	pendiente_cmp	pendiente_intentos	pendiente_yds	pendiente_td	pendiente_int
pendiente_juego	1	0.631275	0.657204	0.643018	0.521424	0.473501
pendiente_cmp	0.631275	1	0.975579	0.958772	0.818419	0.631074
pendiente_intentos	0.657204	0.975579	1	0.944252	0.765079	0.718124
pendiente_yds	0.643018	0.958772	0.944252	1	0.879117	0.599737
pendiente_td	0.521424	0.818419	0.765079	0.879117	1	0.430064
pendiente_int	0.473501	0.631074	0.718124	0.599737	0.430064	1

Fuente: Elaboración propia.

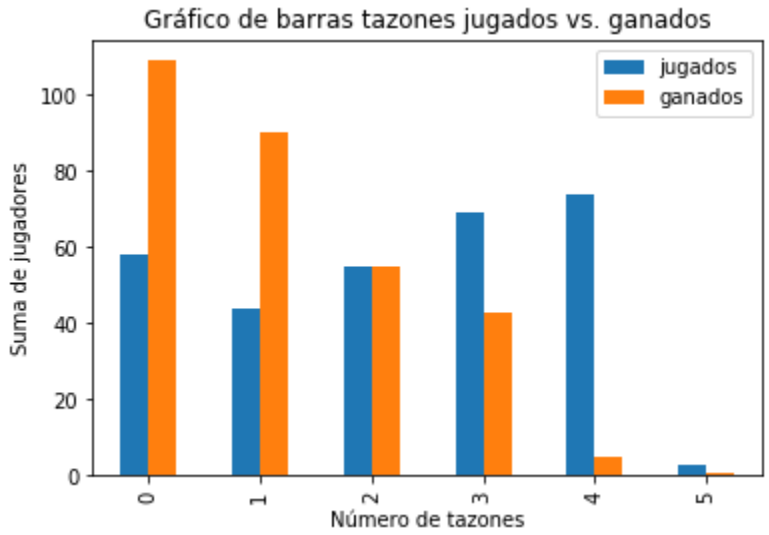
La figura 2.16 es un mapa de calor correlacional, el cual muestra que ninguna variable es negativa, no obstante, el atributo de intercepciones presenta coeficientes muy débiles; pudiendo concluir que, a mayor juegos o yardas, incrementan las intercepciones, pero es cuando intenta más pases que las intercepciones aumentan. Sin embargo, se deberá ser muy cuidadoso al no concluir que mayor número de intercepciones quiere decir una mejor trayectoria. Además, la variable de pases completados ayuda a entender toda la trayectoria colegial ya que es la que cuenta con los coeficientes de correlación más altos.

Figura 3.17: Histograma de juegos ganados de QB.



Fuente: Elaboración propia.

Figura 3.18: Tazones jugados y ganados de jugadores.



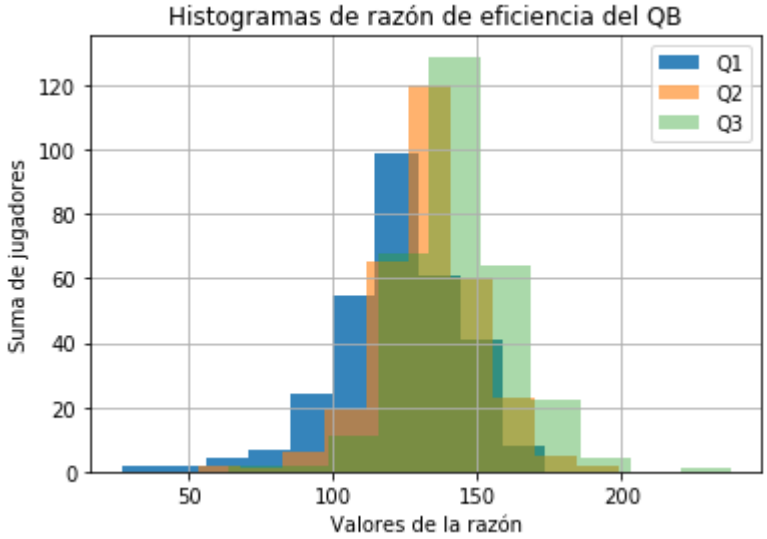
Fuente: Elaboración propia.

La figura 2.17 muestra el histograma del número de juegos ganados por temporada, habiendo treinta y cinco jugadores que ganan menos de un partido en la temporada colegial, los cuales pueden ser valores atípicos. La figura 2.18 muestra los tazones ganados contra los jugados, encontrando que un gran número de jugadores tienen cero o un tazón ganado. Del conjunto de mariscales de campo élite, solo uno (Dan Marino) tiene cero tazones jugados y exceptuando

a este, todos los demás tienen por lo menos un tazón ganado. 40% de ellos tienen cuatro tazones jugados, sin embargo, solo uno (Matt Ryan) ha ganado los cuatro juegos. Los demás ganaron más de dos tazones de los cuatro jugados.

La figura 2.19 muestra las distribuciones de la razón de eficiencia del cuartil uno, la mediana y el cuartil 3. Pudiendo concluir que la distribución es muy similar y con coeficientes de correlación superiores a 0.69 entre las variables, por lo que se puede optar por usar únicamente la mediana en los modelos de predicción.

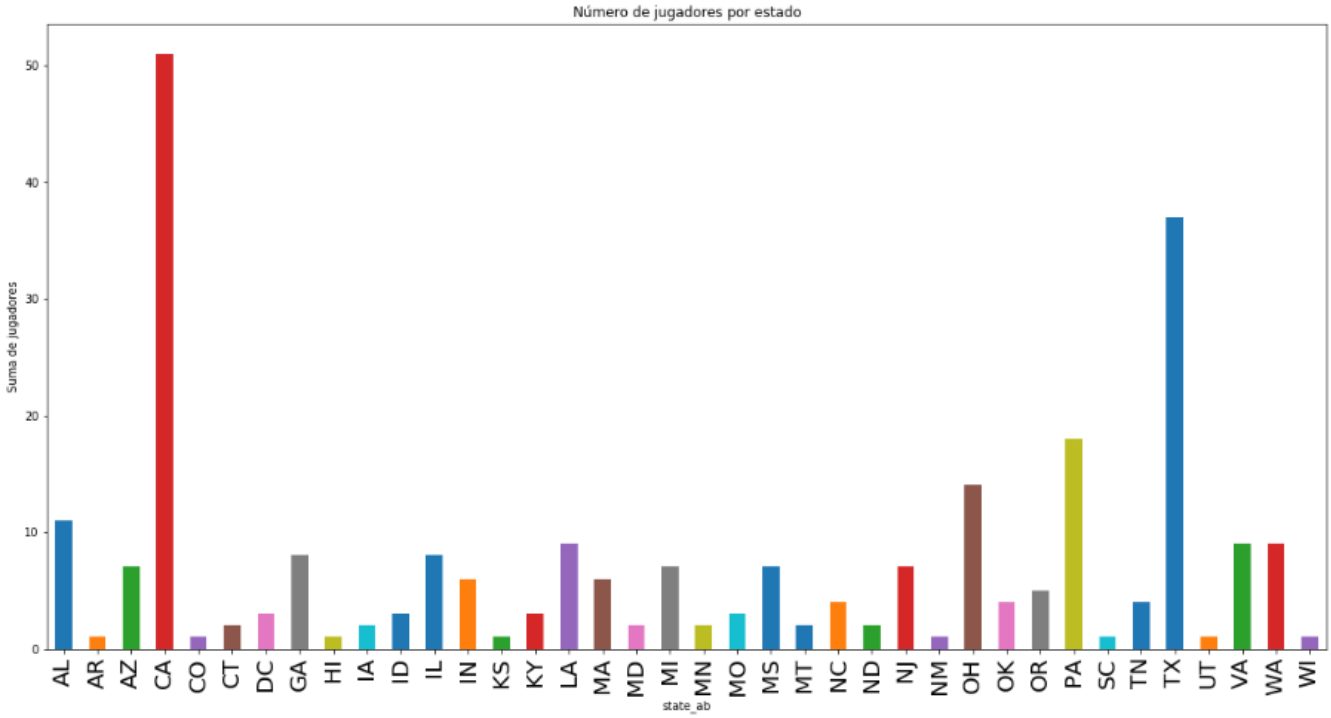
Figura 3.19: Histogramas de razón de eficiencia de QB.



Fuente: Elaboración propia.

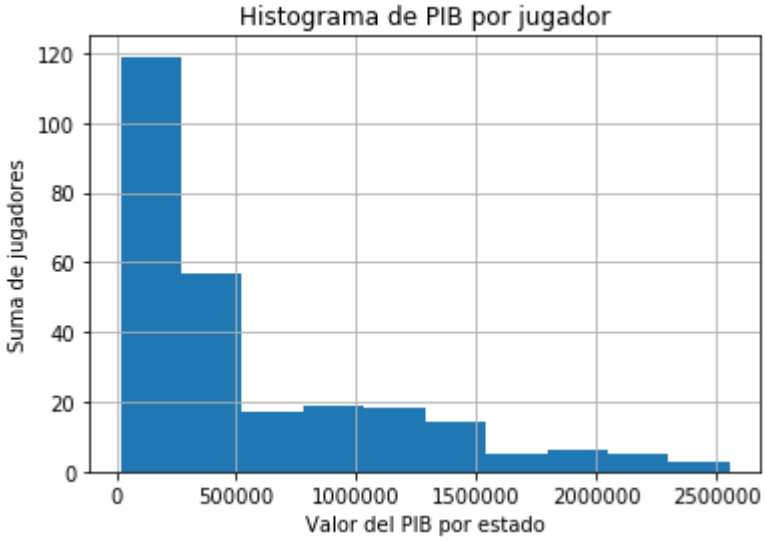
La figura 2.20 muestra la cantidad de jugadores por estado de EE.UU. en el bachillerato. Donde el estado de California y el de Texas son los mayores productores de talento durante el bachillerato. Sin embargo, de la lista de jugadores élite, tres vienen de California y dos de Texas.

Figura 3.20: Número de jugadores por estado de EE.UU.



Fuente: Elaboración propia.

Figura 3.21: Histograma del PIB per cápita por jugador.

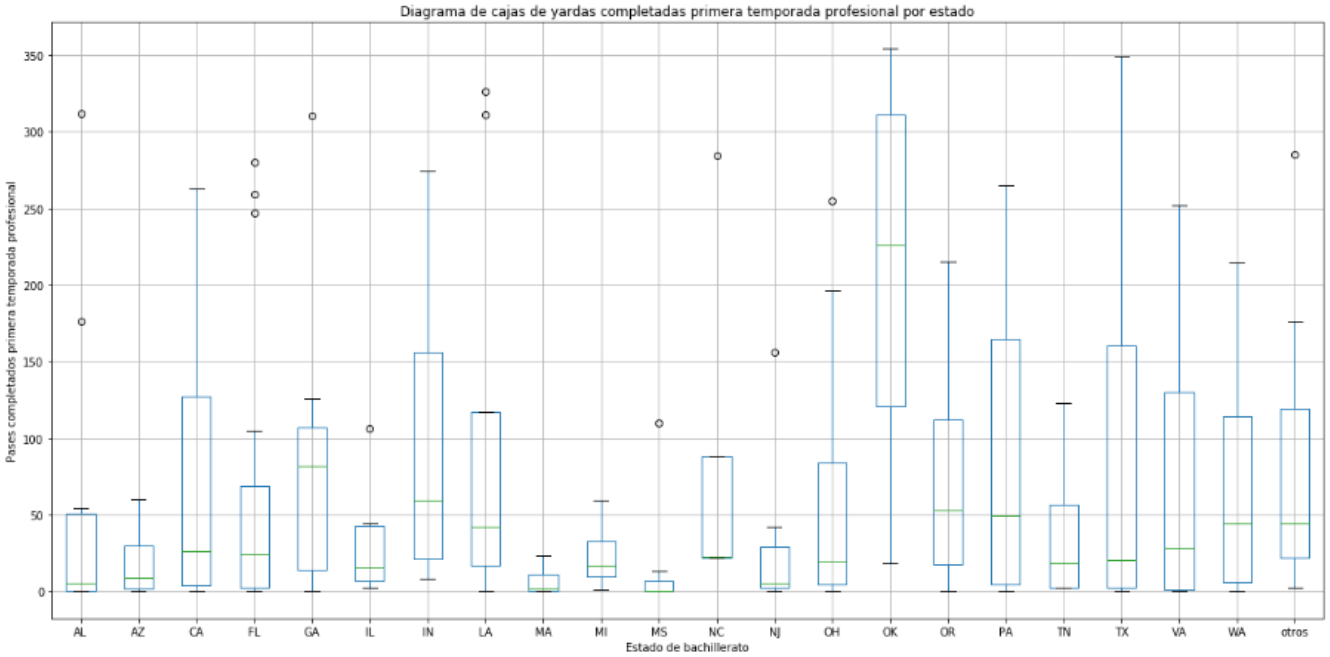


Fuente: Elaboración propia.

A cada estado del bachillerato se le asignó el producto interno bruto del año que corresponde. Esto se realizó en la búsqueda de más variables que expliquen el desempeño esperado de un mariscal de campo en el ámbito profesional. Buscando una relación entre un estado más rico (en términos de PIB per cápita) y mejor o peor desempeño en el futuro. La figura 2.21 muestra la distribución del producto interno bruto per cápita de los jugadores. Encontrando que la mayoría de jugadores provienen de estados con un PIB per cápita inferior a los $500,000 dólares anules. El conjunto de QB élite tiene una media de $440,694. Como se detalló en la sección 1.3, revisión de la literatura, ciertas variables pueden generar sesgos, como es la riqueza de los estados en donde se jugó el bachillerato. Como parte del entendimiento de modelos se incluirá esta variable y se detallará si tiene efecto en la toma de decisiones del agente inteligente.

La figura 2.22 es un diagrama de cajas que muestra la distribución de los pases completados en la primera temporada de cada uno de los estados con más de 3 observaciones. Las medianas de los estados de Minnesota (MN), Misuri (MO), Dakota del Norte (ND) y Oklahoma (OK) son superiores a las de los demás estados, estando por encima del tercer cuartil del resto de regiones, sin embargo, esto se debe a la cantidad de observaciones con las que cuenta dicho estado, por lo que no se pueden considerar como representativas, teniendo todos menos de 5 observaciones.

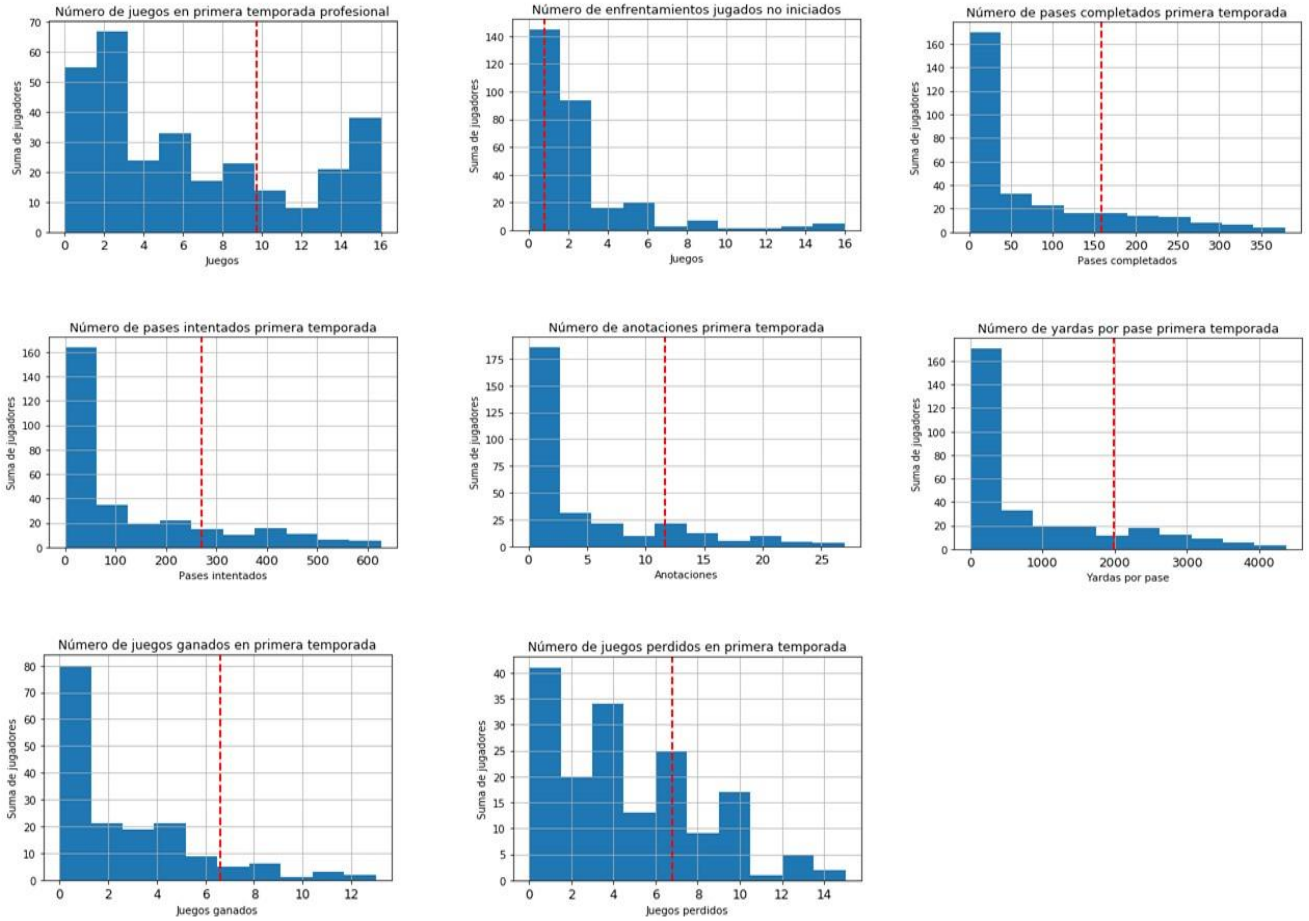
Figura 3.22: Diagrama de cajas de yardas completadas.



Fuente: Elaboración propia.

Para generar un valor dicotómico de aquellos mariscales de campo exitosos, se comparó la media de aquellos que pertenecen al conjunto élite. Primero, de manera visual las estadísticas de la primera temporada, mostradas en la figura 2.23. Con el número de enfrentamientos jugados, número de enfrentamientos jugados no iniciados, pases completados, pases intentados, anotaciones por pase, yardas por pase, juegos ganados y juegos perdidos. Las figuras muestran que las medias de los QB élite no se encuentran en la moda y pueden servir como puntos de corte para el valor binomial de selección. La figura 2.24 es un mapa de correlación de calor entre variables de la primera temporada, encontrando que tanto pases completados como yardas por pase son atributos que permiten explicar el desempeño de un jugador en el ambiente profesional. Sin embargo, los valores de yardas completadas y juegos perdidos tienen un coeficiente de correlación positivo muy alto, contrario a lo que se esperaría, a mayor pases completados más posibilidad de ganar un juego.

Figura 3.23: Histogramas de primera temporada profesional y su media del conjunto élite.



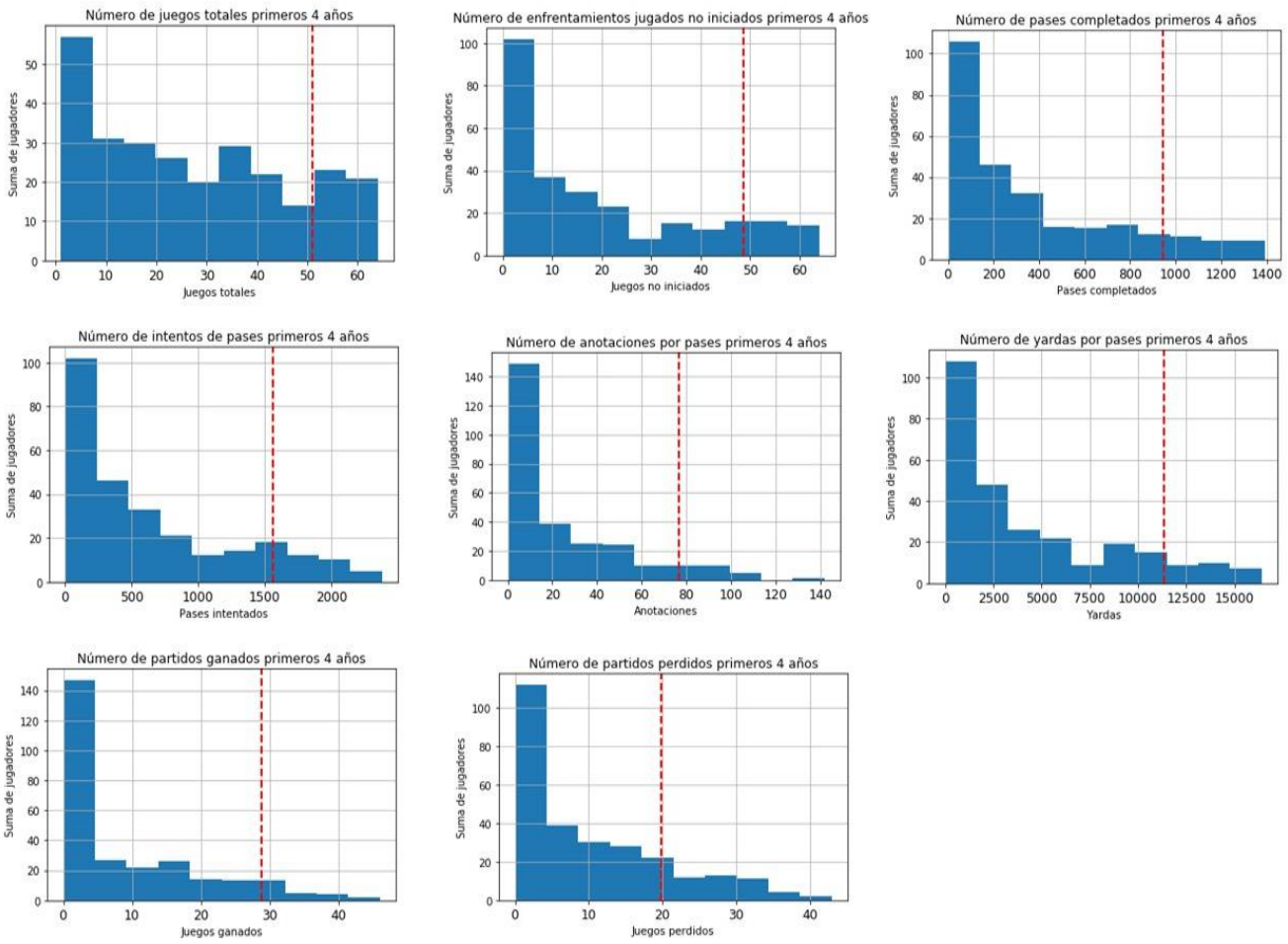
Fuente: Elaboración propia.

Figura 3.24: Mapa correlacional de variables de primera temporada profesional.

	juegos	no_iniciados	cmp	intentos	TD	yds	ganado	perdido
juegos	1	0.236102	0.802897	0.810211	0.745932	0.802444	0.73985	0.783047
no_iniciados	0.236102	1	-0.33563	-0.334407	-0.303382	-0.334919	-0.263295	-0.352171
cmp	0.802897	-0.33563	1	0.994571	0.930783	0.991756	0.724276	0.826226
intentos	0.810211	-0.334407	0.994571	1	0.917424	0.987231	0.705174	0.856557
TD	0.745932	-0.303382	0.930783	0.917424	1	0.946715	0.742157	0.645133
yds	0.802444	-0.334919	0.991756	0.987231	0.946715	1	0.759622	0.79437
ganado	0.73985	-0.263295	0.724276	0.705174	0.742157	0.759622	1	0.327392
perdido	0.783047	-0.352171	0.826226	0.856557	0.645133	0.79437	0.327392	1

Fuente: Elaboración propia.

Figura 3.25: Histogramas de primeras cuatro temporadas profesionales y su media del conjunto élite.



Fuente: Elaboración propia.

Debido a que ciertas observaciones no cumplen con cuatro años o más de experiencia en la liga profesional el arreglo se tiene que reducir a 273 observaciones. Las distribuciones agregadas a cuatro años están sesgadas positivamente, y la media de los mariscales de campo élite, marcada con una línea roja punteada, no se encuentra en la moda de la distribución como lo muestra la figura 2.25.

El cuadro 2.8 muestra el coeficiente de correlación de los pases completados y las yardas por pase en la primera temporada y en los primeros cuatro años profesionales con las variables de su trayectoria colegial. Encontrando que la relación lineal entre variables colegiales y profesionales es prácticamente nula, siendo el coeficiente más elevado 0.37 en la variable de anotaciones máximas durante temporadas colegiales.

	NFL Tot Cmp	***NFL Tot Yds***	***NFL Primero cmp***	***NFL Primero Yds***
bowls_ganados	0.00	0.00	0.05	0.04
juegos_ganados	-0.02	-0.03	0.03	0.03
height	0.19	0.18	0.17	0.16
weight	0.16	0.16	0.16	0.16
juegos_totales	0.04	0.03	-0.02	-0.03
min_juegos	0.08	0.09	0.02	0.03
max_juegos	0.09	0.08	0.24	0.23
dif_juegos_media	-0.04	-0.04	0.03	0.03
pendiente_juego	-0.02	-0.03	0.06	0.06
suma_cmp	0.09	0.08	0.14	0.12
min_cmp	0.07	0.07	0.08	0.09
max_cmp	0.15	0.14	0.26	0.25
dif_cmp_media	0.06	0.06	0.16	0.15
pendiente_cmp	0.09	0.09	0.18	0.17
suma_intentos	0.07	0.07	0.10	0.08
min_intentos	0.07	0.07	0.07	0.08
max_intentos	0.13	0.12	0.22	0.21
diff_intentos_media	0.04	0.03	0.12	0.12
pendiente_intentos	0.06	0.06	0.14	0.13
max_per	0.10	0.08	0.17	0.16
min_per	0.06	0.06	0.12	0.11
media_per	0.11	0.09	0.21	0.19
desv_per	0.00	-0.01	0.00	0.00
total_yds	0.10	0.10	0.16	0.14
min_yds	0.09	0.09	0.08	0.09
max_yds	0.17	0.16	0.32	0.30
dif_yds_media	0.04	0.04	0.17	0.16
pendiente_yds	0.09	0.08	0.21	0.19
total_td	0.14	0.13	0.21	0.19

min_td	0.07	0.08	0.06	0.07
max_td	0.23	0.23	0.37	0.35
dif_td_media	0.08	0.07	0.20	0.19
pendiente_td	0.14	0.14	0.28	0.27
total_int	0.03	0.04	-0.04	-0.03
min_int	0.02	0.02	0.01	0.02
max_int	0.08	0.09	0.00	0.02
dif_int_media	0.01	0.03	0.02	0.04
pendiente_int	0.04	0.05	0.03	0.04
rate_q1	0.10	0.09	0.21	0.20
rate_q2	0.13	0.12	0.27	0.26
rate_q3	0.11	0.10	0.26	0.25
max_yds_intentos	0.02	0.02	0.12	0.12
min_yds_intentos	0.11	0.12	0.11	0.11
media_yds_intentos	0.11	0.12	0.21	0.21
desv_yds_intentos	-0.07	-0.08	0.00	0.01
gdp	0.00	-0.01	0.11	0.10

Cuadro 3.8: Correlación variables NFL y trayectoria colegial. Elaboración propia.

El cuadro 2.9 muestra las estadísticas descriptivas de la variable draft, que muestra la posición en la que fue elegido el jugador. Encontrando que el 50% de los jugadores son elegidos en las primeras 3 rondas y que los datos están sesgados a la derecha, ya que el coeficiente de asimetría de Pearson es mayor a cero. El coeficiente de correlación de la variable draft con las variables de pases completados y yardas por pase en la primera temporada de la NFL es negativo con valor de -0.45 para ambas variables. Por ello, la importancia de una adecuada detección de desempeño esperado, ya que aquellos mariscales de campo de primera ronda no siempre terminan teniendo las mejores estadísticas.

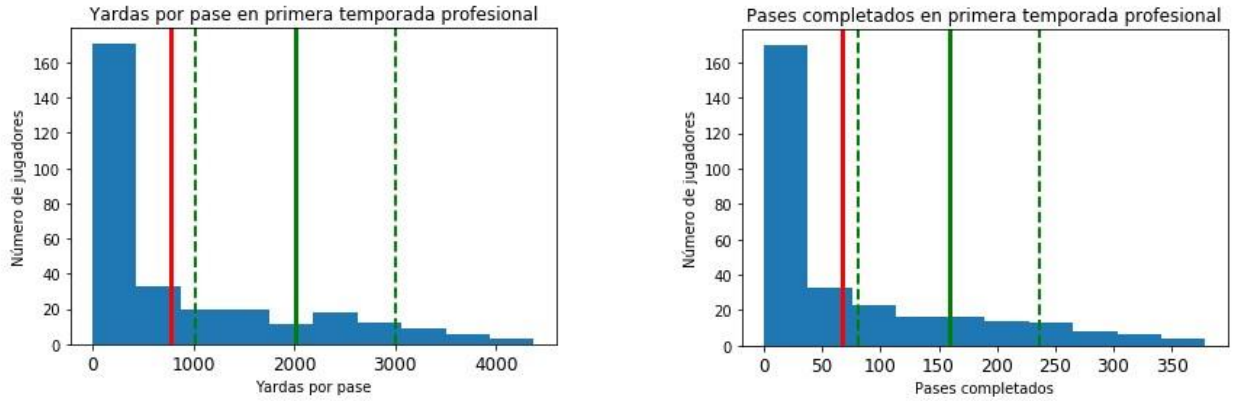
	Valor
Min	1
Media	100.17
Desviación Estándar	78.16
Q1	25

Mediana	93
Q3	171
Max	285
Asimetría Pearson	0.28

Cuadro 3.9: Estadísticas descriptivas variable draft.

Se definirá la etiqueta dicotómica de desempeño esperado, 0 no satisfactorio y 1 satisfactorio, como la unión de los conjuntos de yardas por pase y pases completados de la primera temporada cuyos elementos sean mayores o iguales a la media de los mariscales de campo de las variables respectivas. Creando un nuevo atributo en el arreglo de datos, bajo el nombre “objetivo”. La figura 2.26 muestra los intervalos en los que la media del conjunto de mariscales de campo élite podría caer por aleatoriedad en las variables profesionales de primera temporada de yardas por pase y pases completados respectivamente. Donde la línea roja representa la media de la población, la línea verde la media de la lista élite y las líneas punteadas los intervalos con 95% de confianza de encontrar la media utilizando la distribución t de student. En ambas imágenes se puede concluir que la diferencia en las medias no se debe a la aleatoriedad y por lo tanto fijar la media del conjunto élite como punto de corte, para la decisión binomial regresará aquellas observaciones que destaquen por sus estadísticas. Sin embargo, cabe aclarar que no se cumple con el supuesto de los datos que provengan de una distribución normal, ya que a diferencia de en una prueba de hipótesis convencional, se conoce la distribución de la población.

Figura 3.26: Medias de yardas por pase y pases completados primera temporada NFL.



Fuente: Elaboración propia.

3.3.2 Preprocesamiento de datos para entrada

En este apartado se realizarán actividades de ingeniería de atributos para obtener un arreglo de datos de entrada para los algoritmos de inteligencia de máquina. Se verificará la existencia de valores nulos o atípicos y su imputación o transformación. Según corresponda, se realizarán transformaciones a los datos para hacer más simétricas las distribuciones. Además, se modificarán las variables que contengan cadenas de caracteres a numéricas para que sirvan como entrada del modelo. Dichas actividades se realizarán utilizando la librería de pandas en Python, como se muestra en el apéndice E, pudiendo apreciar el desarrollo de funciones y uso de iteraciones para la estandarización de datos y normalización de variables. El procesamiento de datos es un recurso iterativo en el ciclo de preprocesamiento y modelado, donde ambas actividades apelarán por un mejor comportamiento en los resultados adquiridos del modelo.

Se evitará la eliminación de observaciones por valores nulos debido a la cantidad de información con la que se cuenta. El cuadro 2.10 muestra la cantidad de valores nulos por variable. Una vez que se ha encontrado la observación con valor nulo se examinó el caso para valores inferiores a 11 observaciones nulas por atributo. Los nulos de las variables de diferencias entre temporadas fueron remplazados por cero. Los valores nulos de los atributos de pendiente de juego fueron remplazados con los atributos de las diferencias de temporada. Los datos de postemporada con nulos fueron remplazados por ceros, puesto que son aquellos sin información de postemporada, similar a estos, "*nfl_primero_ganado"* y "*nfl_primero_perdido*" fueron remplazados por cero ya que no hay datos de partidos ganados por que no fueron jugados. El atributo con información de los estados de los bachilleratos tiene 17 valores nulos, los cuales son generados debido a la imposibilidad de hacer el cruce con los valores de PIB por información corrupta en la variable de "*state_high_school*", retornando información extra del estado o información de las rondas del draft por falta de características en el proceso de extracción de datos semiestructurados. Por lo que las variables "*state_ab*", "*year*" y "*gdp*" son afectadas. Las variables de "*state_high_school*" que no contengan información del estado, serán remplazadas por la cadena de caracteres "*otros*", dando como resultado 9 observaciones con valor "otros", el producto interno bruto per cápita de dichas observaciones fue imputado usando la mediana de la variable, debido a la distribución de la misma. Las variables "nfl_primero_ganado" y "*nfl_primero_perdido*" están relacionadas y el hecho de que están nulas se debe a que no participaron en el marcador del juego, por lo que

serán remplazadas por valor cero, al igual que los atributos "*nfl_primeros_juegos*" y "*nfl_primeros_jug_vs_iniciados*". De la misma manera, *"('Year', 'count')", "('Year', 'min')*" y *"('tot_postemporada', 'sum')"* son las variables relacionadas a postemporada por lo que valores nulos quiere decir la usencia de partidos fuera de temporada regular, por lo que las tres variables serán remplazadas con valor cero.

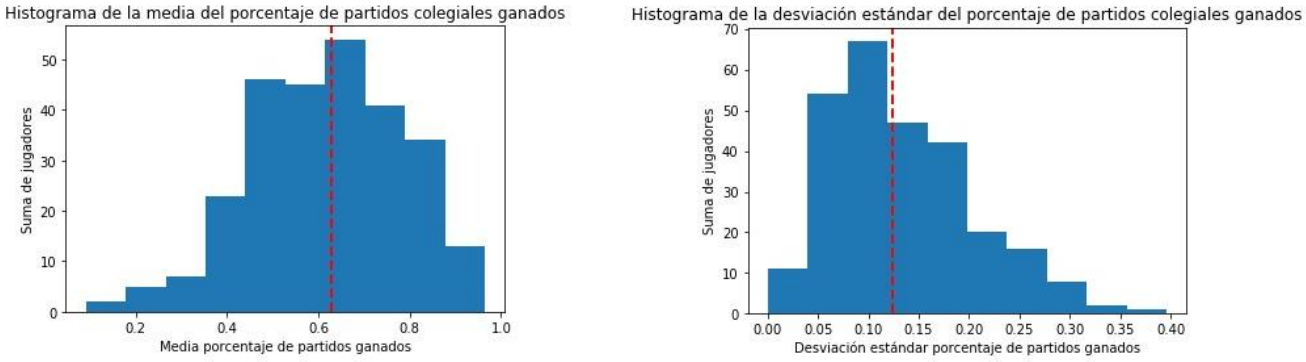
	Valores nulos
Dif_juegos_media	1
Pendiente_juego	3
Dif_cmp_media	1
Pendiente_cmp	9
diff_intentos_media	1
pendiente_intentos	9
Desv_per	2
Dif_yds_media	1
Pendiente_yds	9
Dif_td_media	1
Pendiente_td	9
Dif_int_media	1
Pendiente_int	9
desv_yds_intentos	2
Pct_juegos	33
Pct_juegos_desv	35
State_ab	40
Year	40
Gdp	40
nfl_primeros_juegos	3
nfl_primeros_jug_vs_iniciados	8
nfl_primero_ganado	136
nfl_primero_perdido	136
('Year', 'count')	178

('Year', min)	178
('tot_postemporada', 'sum')	178

Cuadro 3.10: Suma de nulos por variables.

La figura 2.27 muestra la distribución del porcentaje de los juegos colegiales ganados, tanto la media como la desviación estándar. Mostrando que la desviación estándar de juegos colegiales se acumula cerca del cero, con una distribución positiva y serán remplazados los valores nulos con la mediana, línea roja. Para la variable "*pct_juegos*" se remplazarán los valores nulos con el cociente de datos de la variable "*juegos_ganados*" y la suma de la variable "*juegos perdidos*" y "*juegos_ganados*". Para los atributos, "*desv_per*" y "*desv_yds_intentos*" debido a la cantidad de observaciones nulas (dos) se remplazaron con la mediana. Ya que esta es más robusta que la media y para los dos últimos atributos, el hecho de tener tan pocos valores nulos, permite realizar esta aproximación. Otra aproximación posible a estas dos últimas variables es la eliminación de las dos observaciones que tienen nulos o la eliminación de la columna.

Figura 3.27: Histogramas de porcentaje de partidos colegiales ganados.

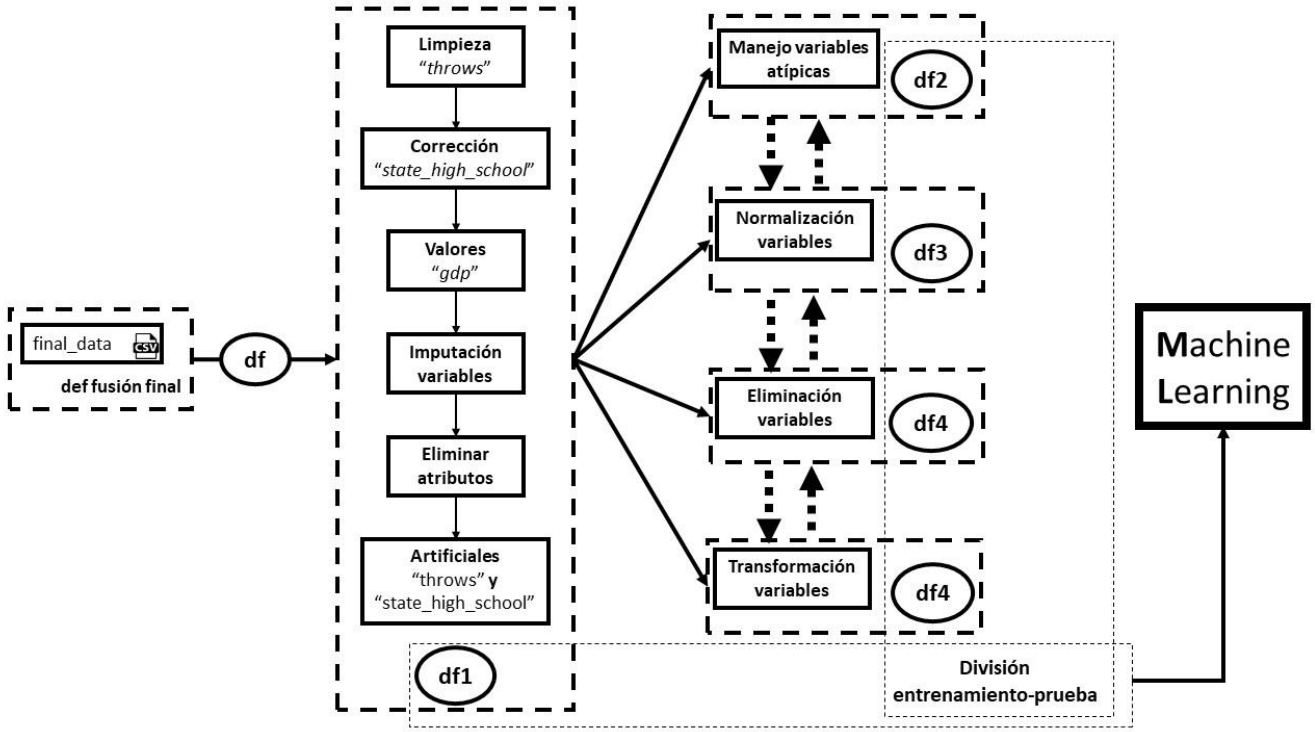


Fuente: Elaboración propia.

La variable de "*throws*", mano con la que lanza, no posee datos nulos, sin embargo, debido a la naturaleza de la extracción ciertos datos son incorrectos y se remplazarán con la moda de los valores: "*Right*". Las variables que no servirán como entrada al modelo son, "*id*" ya que no debería ser predictora, junto con "*nombre*". "*rate_q1*" y "*rate_q3*" de acuerdo a lo señalado en el apartado anterior. Los atributos referentes a años no deben funcionar como predictoras, siendo "*min_año*", "*year*", "*año_draft*" y *"('Year', 'min')*". También fue eliminada la variable de estado que funcionó como "*join*" proveniente del arreglo de datos de buró de análisis

estadístico. Las variables de "*throws*" y "*state_high_school*" son cadenas de caracteres y como tal no sirven como entrada para cualquier algoritmo ML. La primera, es una variable dicotómica y se remplazara "*Right*" por el valor uno y "*Left*" por el valor cero. Mientras que en la segunda variable se hizo uso de la técnica de "variable ficticia" de $n-1$ elementos. Siendo n el número de valores distintos en el atributo, por lo que la estructura del arreglo de datos final, *df1*, es (303, 112). El proceso anterior es representado en la figura 2.28, señalando las distintas salidas del flujo de preprocesamiento, haciendo énfasis en la iteración de normalización de variables, manejo de valores atípicos y eliminación de las mismas como entrada para modelos.

Figura 3.28: Flujo de preprocesamiento de datos.



Fuente: Elaboración propia.

La división aleatoria en arreglo de entrenamiento y arreglo de prueba será realizada durante la fase de modelamiento de inteligencia de máquina debido a que existe una buena calidad en los datos y la imputación de los mismos se debe principalmente a errores de lógica que pueden ser corregidos, como ha sido expuesto anteriormente.

Para el estudio de valores atípicos se consideraron dos aspectos, aquellos valores que están por arriba o por debajo de tres desviaciones estándar de cada atributo y la fórmula utilizada en los diagramas de caja. Donde i es un elemento de un atributo x; $q1$ y $q3$ son el cuartil uno y

cuartil tres, respectivamente, de x; (a) hace referencia a valores atípicos muy por debajo de los "normales" y (b) hace referencia a aquellos que están por arriba de lo que se considera normal.

$$(a) \quad q1 - 1.5 * (q3 - q1)$$

$$(b) \quad q1 + 1.5 * (q3 - q1)$$

Para la decisión de observaciones atípicas, se tomó en consideración la cantidad de variables que posee siendo marcadas como anormales. Debido a la cantidad de datos con los que se cuenta, la eliminación de observaciones es poco deseable. Se optó por eliminar tres variables, mostradas en el cuadro 2.11. Ya que son las observaciones con más valores nulos de acuerdo al aspecto 1 (por encima o por debajo de 3 desviaciones estándar) y se encuentran por encima de la media de suma de valores del aspecto 2 (fórmula de diagrama de cajas).

Índice	**Aspecto 1**	**Aspecto 2**
20	10	16
131	7	9
201	7	13

Cuadro 3.11: Valores por aspecto de observaciones eliminadas.

Se busca que los valores continuos del arreglo de entrenamiento sigan una distribución normal. Para ello se utilizaron las pruebas Anderson-Darling y Shapiro-Wilk. De acuerdo con dichas pruebas, las variables que no rechazan la hipótesis nula h_0, lo cual quiere decir que la variable sigue una distribución normal con un nivel de confianza del 95% son "*pendiente_cmp*", "*suma_cmp*", "*total_int*", "*suma_intentos*", "*pendiente_intentos*", "*total_yds*", "*max_yds*", "*pendiente_yds*", "*max_td*", "*pendiente_td*", "*max_int*", "*pendiente_int*" y "*media_yds_intentos*". Mediante un bucle se probaron las siguientes trasformaciones de datos, apelando a una distribución más normal mediante las pruebas anteriores, eligiendo el p-valor más alto. Donde x es la columna del arreglo de datos y dicha transformación se mapea en todos los elementos de la columna:

- $x^2;\ x^3; x^4; x^5;$
- $x^{1/2};$
- $x^{1/3};$

- $\ln(x)$;
- $1/x$;
- $1/x^2$;
- $1/x^{\frac{1}{2}}$

El cuadro 2.12 muestra la transformación que se ha aplicado a cada una de las variables del arreglo de entrenamiento, mismas que deberán ser aplicadas al arreglo de prueba. Únicamente se incluyeron aquellas variables en donde el mapeo de una transformación afecta el p-valor de las pruebas de normalidad.

Variable	**Transformación**
"juegos_totales"	x^4
"min_juegos"	x^3
"max_cmp"	x^4
"min_intentos"	$x^{1/3}$
"max_per"	$\ln(x)$
"desv_per"	$x^{1/3}$
"total_td"	x^4
"min_td"	$x^{1/2}$
"min_int"	$x^{1/2}$
"max_yds_intentos"	$1/x^{1/2}$
"desv_yds_intentos"	$\ln(x)$
"juegos_ganados"	x^4
"pct_juegos_desv"	$x^{1/2}$
"gdp"	$\ln(x)$

Cuadro 3.12: Transformaciones realizadas por variables.

La eliminación de variables está muy relacionada con los resultados del modelo. Sin embargo, se pueden sugerir algunas variables dados los resultados del apartado de análisis exploratorio de datos. En términos de correlación se puede pensar en las variables de la media de diferencias a través de temporadas colegiales, ya que se cuenta con los datos de pendiente.

Las variables de intercepción o las de pases intentados pueden tener un efecto negativo en el modelo al no ser claro un mejor desempeño dados los pases que se intentan o tener menos o más intercepciones. Se reflexionó sobre la eliminación de las variables referentes a las características físicas, manos con las que se lanza el balón, peso, altura. Y considerar retirar ciertas variables artificiales del estado en que se jugó el bachillerato. Durante el estudio de resultados se puede utilizar el valor-p de la estadística t de la distribución de Student, ya que nos permitirá comprender que tan relevante es el coeficiente otorgado a cierta variable. Además, se pueden utilizar técnicas de regularización, de acuerdo a lo explicado en la sección de ciencia de datos e inteligencia artificial.

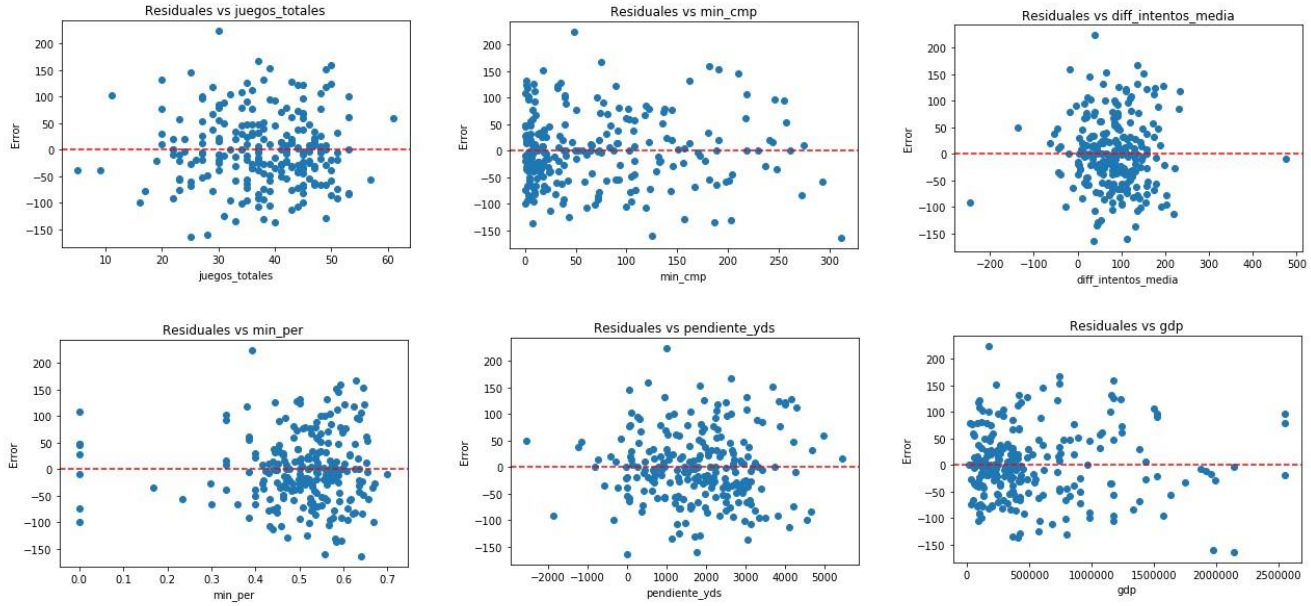
3.3.3 Entrenamiento de modelos

En este apartado se detallará el proceso de entrenamiento que se llevó a cabo para la creación de modelos, para asegurar un buen proceso iterativo y los mejores resultados predictivos. Se entrenará sobre más de un modelo, sin embargo, en este apartado se detallarán las estadísticas principales utilizando un algoritmo ML de regresión lineal y un modelo de clasificación de regresión logística, que servirán como punto de referencia, mediante las librerías de *stastmodels* y *scikit-learn*, como se muestra en el apéndice F, haciendo énfasis en la importancia de las librerías de código abierto y la facilidad de implementación de herramientas machine learning. La paquetería de *statsmodels* fue manipulada para el uso de modelos de regresión lineal múltiple, la prueba de los posteriores modelos fue realizada con *scikit-learn*. Utilizando regularizaciones, arboles de decisión y perceptrones multicapa para ambas aproximaciones, regresión y clasificación.

Se inició por dividir el arreglo de datos de manera aleatoria en entrenamiento y prueba, bajo una estructura 80-20. Se ha elegido dicha estructura debido a la relativa poca cantidad de observaciones con las que se cuenta. Primero se entrenará un modelo de salida continua (regresión) para la variable *“nfl_primero_cmp”*. Se ha elegido dicha variable sobre “*nfl_primeros_yds*” ya que ambas están correlacionadas y se ha mostrado anteriormente que encapsulan la trayectoria profesional del jugador. Asimismo, la última tiene la característica de almacenar aquellas yardas compartidas con los receptores una vez que el balón ha sido recibido, mientras que la primera mostrará la capacidad del mariscal de campo para encontrar receptores abiertos y la calidad del pase ya que fue capturado. Siguiendo a Matthew Mayo (2017) la estadística clásica y la inteligencia de máquina tienen fines distintos, la primera se

enfoca en la abstracción teórica, mientras que la segunda tiene como fin un modelo práctico que se ponga en producción para la toma de decisiones. A esto se le puede sumar el uso de arreglos de prueba para comprobar resultados, lo cual es posible por la cantidad de datos, donde la importancia reside en la capacidad de generalizar del modelo en dicho arreglo. No obstante, es importante asegurarse que nuestro modelo de regresión lineal cumpla con los supuestos de generalización. Como se muestra en el cuadro 2.8 no hay relación lineal entre la variable objetivos y las variables predictoras y si existe multicolinealidad entre las últimas; sin embargo, utilizaremos un modelo lineal como "*benchmark*" para los modelos subsecuentes. Utilizando los atributos colegiales, los cuales suman noventa variables, se entrenará una primera versión de regresión lineal múltiple, sobre 242 observaciones. Teniendo como resultados una desviación de la raíz cuadrada media (RMSE) de 66.11 en el arreglo de entrenamiento y una R-cuadrada de 0.503, como se muestra en el apéndice G. Considerando que la media de pases completados es de 72 pases, por lo que una RMSE tan grande no ayudará para una predicción precisa del desempeño. Para comprobar el supuesto de varianza constante y como ayuda del supuesto de independencia de los errores residuales, los últimos fueron graficados contra todas las variables, esperando que sigan un patrón aleatorio. La figura 2.29 muestra algunas de las gráficas de residuales contra las variables predictoras. Mostrando que la variable de la suma de juegos totales, la pendiente de las yardas por pase y la media de la diferencia de intentos en temporadas parecen estar aleatoriamente distribuidas. Mientras que el valor mínimo de pases completados, el mínimo de porcentaje de pases completados y el PIB per cápita de estado de bachillerato están agrupados a la izquierda y derecha, lo cual apela en contra del supuesto de independencia y de varianza constante. Como parte de la prueba de independencia, se utilizó Durbin-Watson (DW), con el fin de medir la autocorrelación de los residuales. Donde la estadística DW oscila entre el cero y el número cuatro, siendo el valor dos, la aceptación de la hipótesis nula h_0, indica no correlación serial. Valores cercanos a cuatro quiere decir correlaciones negativas, mientras que cercanos a cero correlaciones positivas. El valor DW obtenido es 2.007, por lo que se puede concluir que no hay correlación entre los errores residuales y se cumple el supuesto de independencia.

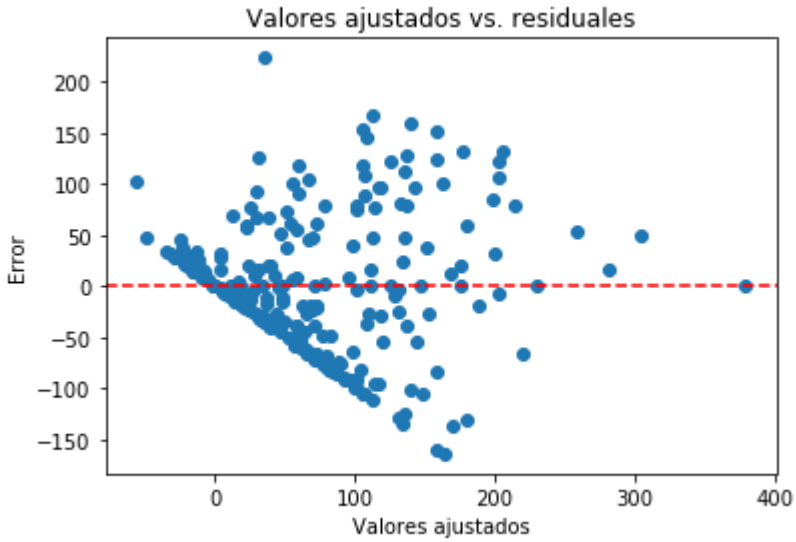
Figura 3.29: Residuales de variables modelo primer año profesional.



Fuente: Elaboración propia.

La figura 2.30 muestra la gráfica de los residuales contra los valores ajustados para asegurarse que la varianza de los errores sea constante a lo largo de las predicciones y las variables predictoras, homocedasticidad. Se puede apreciar que a lo largo de las predicciones la varianza aumenta. Además, para determinar el supuesto de homocedasticidad se usará la prueba Breusch-Pagan, donde la variable nula asume varianza constante. Encontrando que el valor-p de la estadística-F rechaza la hipótesis nula, siendo $1.0667e^{-8}$ y 2.8029 respectivamente, por lo que nuestros datos tienen heteroscedasticidad.

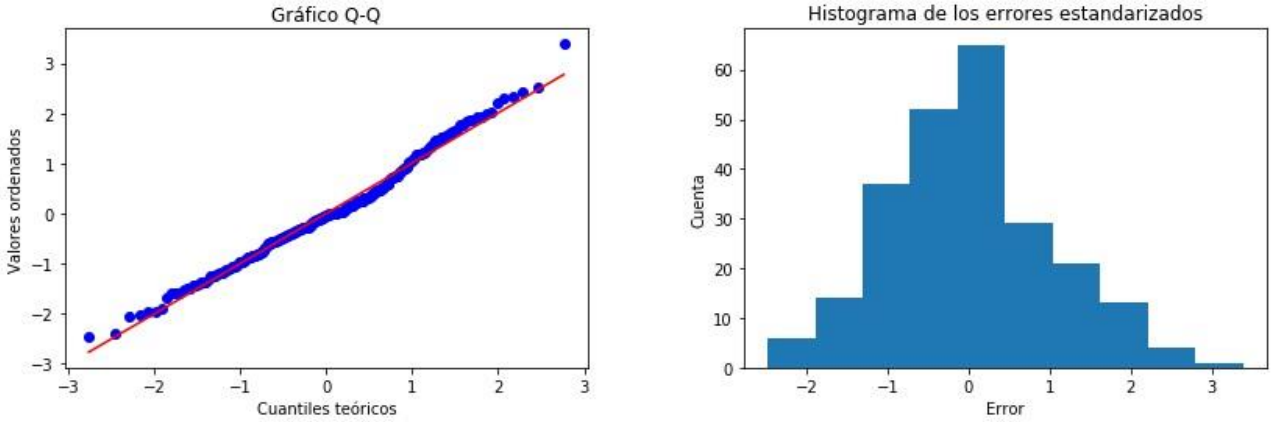
Figura 3.30: Valores residuales y valores ajustados. Primer año profesional.



Fuente: Elaboración propia.

Se debe verificar que los errores sigan una distribución normal, para ello utilizaremos un gráfico Q-Q y un histograma de los residuales, como lo muestra la figura 2.31 respectivamente. La línea de mejor ajuste, mostrada en rojo, tienen una R-cuadrado de 0.988, por lo que los cuantiles que siguen una distribución normal están estrechamente relacionados con los cuantiles de los errores observados. Asimismo, se realizó la prueba Shapiro-Wilk y Anderson-Darling para comprobar normalidad, sin embargo, en la primera prueba la hipótesis nula fue rechazada, con un nivel de confianza de 95%. Mientras que la prueba Anderson-Darling tiene un nivel de confianza superior a 99% de aceptar la hipótesis nula.

Figura 3.31: Normalización de los residuales primer año profesional.



Fuente: Elaboración propia.

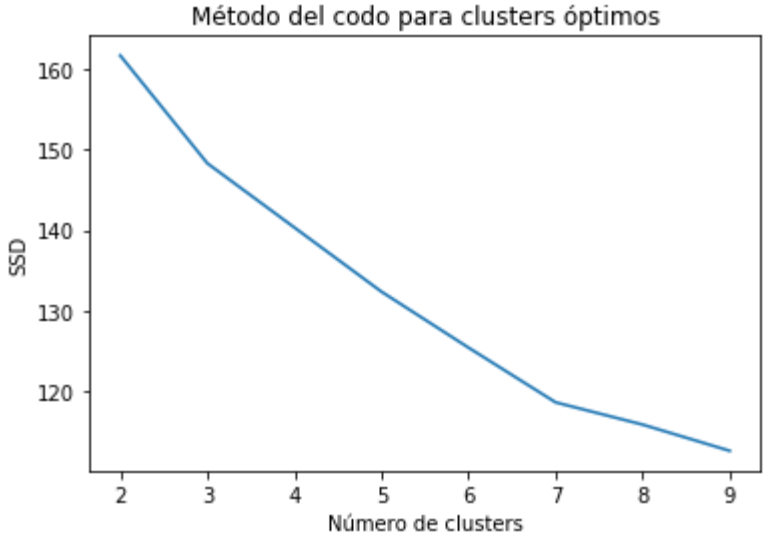
Se buscará que nuestro modelo cumpla los supuestos de regresión lineal, como hemos visto anteriormente, no se cumple la varianza constante en los errores residuales, ni la linealidad entre las variables dependientes e independientes, lo cual es la base de modelos lineales. No obstante, y bajo la idea del empleo del modelo anterior como punto de referencia, se realizará un proceso de pruebas que buscará minimizar la RMSE y maximizar R-cuadrado, la cual es una medida estadística (0, 1) que revela la capacidad del modelo para explicar la relación entre las variables predictoras y la variable objetivo.

También se entrenó un modelo de regresión con los primeros 4 años de trayectoria profesional, utilizando la variable "*nfl_tot_cmp*". Ya que es el tiempo mínimo de duración de un contrato de novato y permite conocer los inicios de la carrera profesional de un mariscal de campo. Primero, se redujo el arreglo a la cantidad de mariscales de campo con 4 años o más de trayectoria, obteniendo 273 observaciones, después de la división entrenamiento-prueba 218 observaciones para prueba. Se obtuvo una R-cuadrado de 0.574 y una RMSE de 247.46, recordando que la media de la variable objetivo es 361.68. La prueba Durbin-Watson tiene un resultado de 2.021, por lo que los errores residuales son independientes, de acuerdo a lo señalado en el apéndice H. A diferencia de pases completados en el primer año de la temporada profesional, la prueba Breusch-Pagan para homocedasticidad, acepta la hipótesis nula, h_0, con un nivel de confianza de 95%. Para comprobar la normalidad de los errores residuales se utilizó la prueba Shapiro-Wilk y Anderson-Darling. Tanto en la primera prueba, como en la segunda se puede rechazar la h_0 con un nivel de confianza de 95%, afirmando que los errores se distribuyen normalmente.

De acuerdo a la decisión binomial detallada en el apartado de análisis de datos exploratorio, se tomará como punto de referencia de problemas de clasificación la regresión logística. Se tienen 56 observaciones positivas (marcados como 1), lo cual quiere decir que cumplen con las características superiores a la media del conjunto élite. Sobre 303 observaciones totales del arreglo, por lo que no está balanceada la variable objetivo. Se realizará un sobremuestreo con "*bootstrap*", la última es una técnica estadística que permite crear observaciones a partir de una muestra del arreglo. No se considerará el submuestreo, puesto que son pocas las observaciones con las que se cuenta. Se utilizará agrupación de K-medias para dividir en colecciones similares las observaciones, con lo que lograremos que nuestro muestreo sea más representativo. Se transformarán los datos con el algoritmo *MinMax* para un mejor desempeño

de K-medias. La muestra utilizada en la técnica “*bootstrap*” será el 50% de las observaciones del cluster. Para elegir el número adecuado de grupos se utilizará el método del codo y de silueta. Estos permiten encontrar un punto óptimo, mediante la visualización de gráficos. El primero muestra el número de clusters y la suma cuadrática de las distancias (SSD) al centro de los clusters, por lo que debemos de seleccionar el primer punto después de la caída drástica del SSD, en este caso el número cuatro, como se muestra en la figura 2.32.

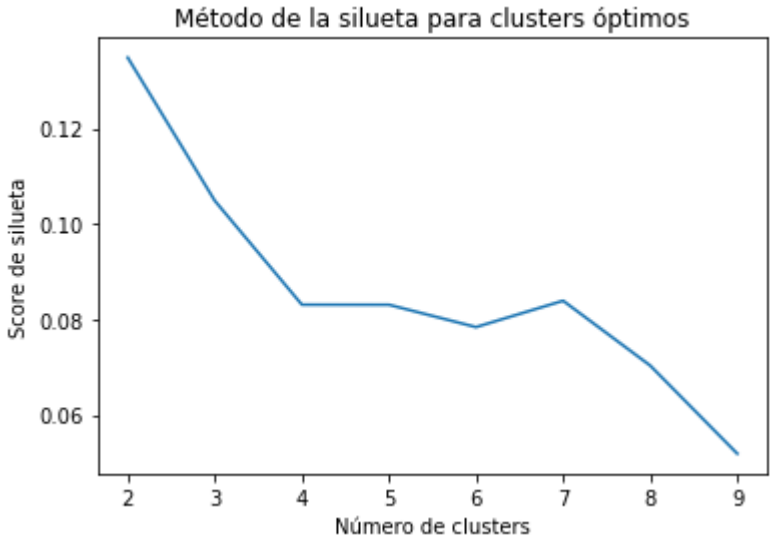
Figura 3.32: Selección k óptima, método del codo para primer año profesional.



Fuente: Elaboración propia.

El segundo, permite comprender que tan parecidos son los puntos en un cluster ya que toma en consideración las distancias euclidianas entre los puntos que pertenecen al mismo cluster. Y es un coeficiente entre menos uno y uno (-1, 1) que buscamos maximizar. En este caso el número 2, como lo muestra la figura 2.33, sin embargo, por la cantidad de observaciones con las que se cuenta se optó por el número cuatro como punto óptimo de cantidad de grupos.

Figura 3.33: Selección k óptima, método de la silueta para primer año profesional.



Fuente: Elaboración propia.

Para la creación de nuevas observaciones mediante muestras se utilizó la técnica estratificación, la cual consiste en tomar en cuenta la proporción de la población, en este caso de los grupos. El cuadro 2.13 muestra la cantidad de observaciones en cada cluster y su proporción.

k	Número de observaciones	Proporción
0	14	25%
1	13	23%
2	16	29%
3	13	23%

Cuadro 3.13: Observaciones y proporción por cluster k.

Se buscó crear 191 observaciones "*bootstrap*", para balancear el arreglo 50-50. Se obtuvo un arreglo final de 494 observaciones. El algoritmo utilizado para resolver el problema de optimización es "*saga*" con 10,000 iteraciones para llegar al punto de convergencia. La exactitud de nuestro modelo, que está representada por el número de predicciones correctas (0=0 y 1=1) entre el número total de predicciones es de 0.58.

En nuestro caso, un falso positivo indicaría un jugador que fue predicho con desempeño esperado prometedor, sin embargo, no lo tuvo. El caso contrario, los falsos negativos, indicarían un jugador que fue marcado como desempeño bajo, mientras que en la realidad su

desempeño fue muy alto. Una matriz de confusión, permite ver los datos de las clases negativas y positivas, como lo muestra la figura 2.34.

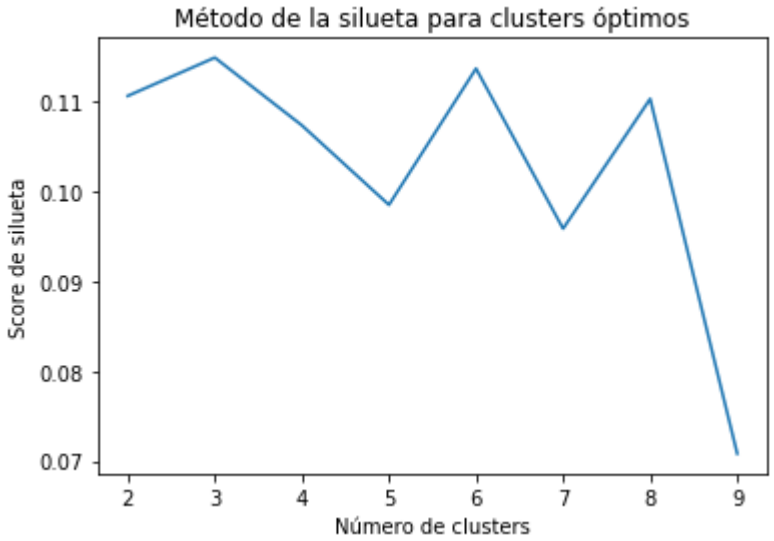
Figura 3.34: Matriz de confusión primer año profesional.

		Actuales	
		Negativos (0)	*Positivos (1)*
Predicción	**Negativos (0)**	93	54
	Positivos (1)	112	136

Fuente: Elaboración propia.

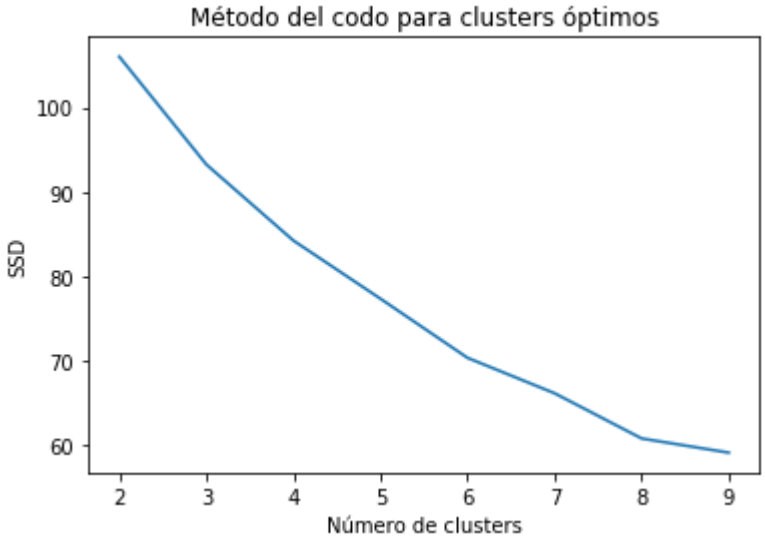
Otras medidas que nos pueden ayudar son la precisión y la exhaustividad, obteniendo 0.55 y 0.72 respectivamente. El modelo de clasificación binomial para los primeros cuatro años, también tomará como punto de referencia la regresión logística. Se tienen 273 observaciones, de las cuales 33 están categorizadas como positivas, se buscó balancear el arreglo, al igual que en el modelo anterior, utilizando agrupación de las variables predictoras; obteniendo la figura 2.35 y 2.36. Debido al número de observaciones positivas y por los resultados obtenidos mediante el método de la silueta, se utilizarán tres grupos, de 10, 5 y 18 observaciones respectivamente. Buscando crear 207 nuevas observaciones.

Figura 3.35: Selección k óptima, método del codo para primeros cuatro años profesionales.



Fuente: Elaboración propia.

Figura 3.36: Selección k óptima, método de la silueta para primeros cuatro años profesionales.



Fuente: Elaboración propia.

Se obtuvo una exactitud de 0.542 después de 10,000 iteraciones, con precisión de 0.523 y exhaustividad de 0.995. Obteniendo 191 verdaderos positivos, un falso negativo, 175 falsos positivos y 17 verdaderos negativos.

Se entrenó los siguientes modelos regresores, los cuales serán comparados en la sección de resultados. Empezando sobre la base de regresión lineal múltiple en el primer año profesional

y en los primeros cuatro años profesionales, mediante la transformación del arreglo de entrada (de acuerdo al apartado de preprocesamiento de datos). Primero se eliminaron los valores atípicos (A), seguido por la normalización de las variables (B), normalización de variables óptima (C), mostradas en el cuadro 2.14, de lado derecho el primero año profesional, mientras que del lado izquierdo están los primeros cuatro años profesionales:

Primera temporada	**Primeros 4 años**
juegos_totales	*min_juegos*
max_cmp	*max_cmp*
min_intentos	*min_intentos*
total_td	*juegos_ganados*
min_int	*pct_juegos_desv*
max_yds_intentos	*gdp*
desv_yds_intentos	

Cuadro 3.14: Normalización de variables óptimas.

Posteriormente se eliminaron las variables utilizando el p-valor de la estadística t de los coeficientes (D), manteniendo únicamente *"diff_intentos_media", "max_td", "dif_td_media", "pct_juegos", "AR", "CT", "IA", "MO", "ND", "NJ", "OR", "PA", "SC"* para el primer año profesional; y para los primeros cuatro años profesionales, *"años_colegial", "universidad_distinta", "CA", "FL", "IN", "LA", "MO", "PA", "TX"*. Después se probó eliminar las variables propuestas en el apartado de preprocesamiento (E). Por último, agregar las variables entrenadas del modelo (D), que no estuvieran en el arreglo de (E), siendo únicamente las variables artificiales, a las cuales llamaremos (F). No se ha usado un arreglo de validación para la hiperparametrización de los meta-algoritmos, puesto que se usará la técnica de validación cruzada con cuadrícula de búsqueda. La última servirá para encontrar el valor *alpha* adecuado en *lasso* y *ridge* sin valores atípicos y con normalización en las variables óptimas (C). Todo el arreglo de entrenamiento será estandarizado con algoritmo *MinMax*, ya que para la optimización en *scikit-learn* la convergencia es más rápida. Para el primer año de temporada colegial, los hiperparámetros que mejor ajustan en *lasso* y *ridge* es un *alpha* de 0.9 y selección aleatoria en *lasso*; lo cual quiere decir, que se actualiza de manera aleatoria un coeficiente, comparado con, hacerlo de manera cíclica. Mientras que en los modelos entrenados para la predicción de los primeros cuatro años de trayectoria profesional se obtuvo selección cíclica y valores iguales

en *alpha*. Tanto para el primero año, como para los primeros cuatro, se entrenaron modelos polinomiales de grado dos y tres. También se realizó un árbol de decisión regresor, obteniendo los mismos resultados en ambos. Donde, los mejores parámetros fueron profundidad de dos, mínimo cinco elementos en cada nodo y pudiendo dividir aquellos nodos que tengan 2 elementos. Al utilizar la cuadricula de búsqueda, se apela a minimizar el RMSE en el arreglo de entrenamiento, sin embargo, se puede caer en sobreajuste. Por último, se entrenará un regresor de perceptrón multicapa, obteniendo función de activación "*relu*", la cual consiste en regresar los resultados mayores a cero, aquellos que sean igual o inferiores a cero, se les asignará el valor cero, con un *alpha* para regularización de 0.001, con cincuenta capas ocultas y 1000 iteraciones como máximo. Mientras que, para los primeros cuatro años profesionales, se usó la misma función de activación y se elevó a cien capas ocultas y quinientas iteraciones.

Para los modelos de clasificación, los dos valores atípicos que pertenecen al conjunto negativo (0) fueron eliminados, mientras el que pertenece al positivo fue mantenido (A). Se realizó normalización en todas las variables (B) y sobre las variables óptimas (C). Se eliminaron las variables especificadas en el apartado de preprocesamiento de datos (E). Se entrenó con regularización lasso y ridge en cuadrícula de búsqueda para automatización sobre F1.

Obteniendo en *lasso, alpha* 0.4 con máximo de 500 iteraciones; mientras que con *ridge* se obtiene 0.1 en *alpha*, convergiendo con mayor rapidez en un máximo de 100 iteraciones. Las variables con coeficientes superiores a cero después de regularización *lasso* y *ridge* (D) serán utilizados para entrenar arboles de decisión y perceptrones neuronales multicapa. Para los primeros cuatro años profesionales, en regularización L1 se obtuvo un *alpha* de 0.5 y un máximo de 500 iteraciones. Para *ridge*, se obtuvo un *alpha* de 0.3 y máximo de 100 iteraciones. Comparado con la aproximación del primer año de trayectoria profesional, se utilizarán las variables con coeficientes superiores a cero después de regularización *lasso* (D). Se usará *lasso* ya que *ridge* en la búsqueda de optimización obtuvo un parámetro *alpha* de regularización que penaliza muy poco. El árbol de decisión con arreglo de entrada (E) utiliza el criterio de división de entropía, sobre "*gini*", con profundidad máxima de seis ramas, mínimo cinco observaciones para división y mínimo dos observaciones por nodo. Mientras que con conjunto de entrada (D), la máxima profundidad es quince, con un mínimo de dos observaciones para división y uno para hoja. La entrada del perceptrón multicapa fue regularizada con *MinMax* en el arreglo (E) utilizando la función logística como activación y

ochenta capas ocultas con *alpha* de regularización de 0.001. Utilizando el arreglo de entrada (D) se utilizan cien capas ocultas. Para los cuatro primeros años profesionales el árbol de decisión entrenado con arreglo (E) obtuvo una profundidad máxima de seis, mínimo cinco observaciones en un nodo y dos como mínimo para proceder a una división. Mientras que utilizando las variables *lasso*, se obtiene una profundidad máxima de ocho y dos observaciones mínimas, tanto en nodo como para decisión de división. Para el perceptrón multicapa también estandarizó las variables con *MinMax*. Se utilizó "*relu*" como función de activación. Con ochenta capas ocultas y máximo 500 iteraciones para llegar al punto de convergencia y un *alpha* de regularización de 0.01. Para las variables *lasso* (D) se redujo el parámetro de regularización a 0.001.

4 Resultados

En esta sección se compararán las estadísticas de los modelos de regresión y clasificación en su proceso de entrenamiento y de prueba, creados en el capítulo anterior. Aquellos modelos con mejor desempeño en el arreglo de prueba, serán estudiados para conocer si el modelo sobreajusta. Finalmente, se realizará una propuesta funcional a la liga de fútbol americano profesional en materia de mariscales de campo.

4.1 Comparación de modelos

En este apartado se detallarán los resultados de los modelos entrenados en el apartado de entrenamiento de modelos. Donde se priorizará por la practicidad del paradigma *machine learning*, ya que los supuestos estadísticos se han estudiado en el apartado anteriormente mencionado. Se comparó la RMSE y R-cuadrado de los modelos entrenados bajo regresión lineal múltiple. El cuadro 2.15 muestran las estadísticas de los modelos regresores tanto para el primer año de temporada profesional como de los primeros 4 años de temporada profesional. Se muestran los resultados de entrenamiento y prueba. La última nos dará una idea de que tan bien ajusta el modelo con datos que no han sido vistos. Los primeros 6 modelos, tanto del año uno, como de los primeros cuatro años de trayectoria profesional son regresiones lineales múltiples, entrenados con variaciones del arreglo de datos, como se ha especificado en el apartado de entrenamiento de modelos. Se puede concluir para los modelos entrenados con el primer año profesional, que el mejor arreglo de entrada para entrenar es el (F) con una R-cuadrado de 0.405, que es baja ya que explica solo el cuarenta por ciento de la variación entre variables predictoras y la variable objetivo. Sin embargo, obtiene la RMSE en entrenamiento más baja. Además, es aquel en donde se eliminan aquellas variables que pueden presentar multicolinealidad. Con los modelos de regularización hay un brinco substancial en la RMSE y ligera diferencia con el arreglo de prueba, comparado con los ajustes polinómicos, que ajustan perfectamente los datos de entrenamiento, pero tienen muy mal desempeño en los de prueba, por lo que se puede concluir que están sobre ajustando. Tanto el árbol de decisión como el perceptrón neuronal multicapa tienen peores resultados tanto en entrenamiento como en prueba que los métodos de regularización L1 y L2. El PNM tiene el mejor RMSE en entrenamiento, sin embargo, en prueba es superado por *lasso* y *ridge*.

	Modelos		RMSE	R-cuadrado
Primera temporada profesional	Atípicos (A)	Entrenamiento	65.47	0.51
		Prueba	117.57	
	Normalización (B)	Entrenamiento	66.35	0.497
		Prueba	111.2	
	Norm óptima (C)	Entrenamiento	64.73	0.522
		Prueba	115.37	
	Eliminación t (D)	Entrenamiento	82.88	0.216
		Prueba	100.51	
	Eliminación corr (E)	Entrenamiento	74.63	0.364
		Prueba	102.46	
	Eliminación unión (F)	Entrenamiento	72.18	0.405
		Prueba	107.12	
	Lasso	Entrenamiento	80.54	0.259
		Prueba	89.29	
	Ridge	Entrenamiento	80.53	0.2954
		Prueba	89.29	
	Grado 2	Entrenamiento	0	1
		Prueba	124.94	
	Grado 3	Entrenamiento	0	1
		Prueba	115.31	
	Árbol Decisión	Entrenamiento	82.82	-
		Prueba	98.46	
	PNM	Entrenamiento	79.51	-
		Prueba	91.48	
Primeros cuatro años profesionales	Atípicos (A)	Entrenamiento	268.19	0.483
		Prueba	501.72	
	Normalización (B)	Entrenamiento	273.57	0.463
		Prueba	511.05	
	Norm óptima (C)	Entrenamiento	266.99	0.488
		Prueba	495.73	

	Eliminación t (D)	Entrenamiento	356.19	0.089
		Prueba	422.94	
	Eliminación corr (E)	Entrenamiento	322.52	0.333
		Prueba	473.32	
	Eliminación unión (F)	Entrenamiento	299.36	0.373
		Prueba	469.52	
	Lasso	Entrenamiento	296.07	0.371
		Prueba	407.12	
	Ridge	Entrenamiento	293.5	0.381
		Prueba	404.59	
	Grado 2	Entrenamiento	0	1
		Prueba	623.48	
	Grado 3	Entrenamiento	0	1
		Prueba	594.08	
	Árbol Decisión	Entrenamiento	334.47	-
		Prueba	443.49	
	PNM	Entrenamiento	364.99	-
		Prueba	403.77	

Cuadro 4.1: Resultados de modelos de regresión.

Para los primeros cuatro años de trayectoria profesional, la R-cuadrado es inferior que, en los ajustes de primer año profesional, por lo que es más complicado predecir un valor continuo a cuatro años que a un año. Siendo la normalización de las variables óptimas la que mejor R-cuadrado arroja. Mientras que la eliminación de las variables no significativas de acuerdo a la estadística *t*, conlleva a un R-cuadrado inferior a 0.1. Con regularización *lasso* y *ridge* se obtiene resultados cercanos entre entrenamiento y prueba. También en este caso, la regresión polinómica en ambos grados sobre ajusta. El perceptrón multicapa obtiene los mejores resultados, muy parecidos a los de la regularización, pero desempeñándose peor en el entrenamiento.

Los modelos de clasificación en el primero año profesional están detallados en el cuadro 2.16, obteniendo lo siguiente, eliminado los valores atípicos se tienen mejores resultados en entrenamiento que la simple regresión logística, sin embargo, son peores en prueba. La

normalización de variables optimas eleva la exactitud del modelo en 25% y en el arreglo de prueba en 5% y se eleva la precisión en 15%. Tanto *lasso* como *ridge* parecen sobre ajustar, ya que se tiene mejores valores en entrenamiento que con (C) pero peores en prueba para precisión y muy similares para exactitud. El árbol de decisión con las variables *lasso* y *ridge* tiene mejores estadísticas que si se usarán todas las variables, sin embargo, por los valores en entrenamiento, puede haber problemas de sobreajuste. Mientras que el perceptrón neuronal multicapa obtiene mejores resultados usando todas las variables y es el modelo con mejor exactitud, precisión y exhaustividad de todos. No obstante, su elevada precisión permite concluir que deja afuera ciertos jugadores con alto desempeño, positivos, como negativos.

		Exactitud	**Precisión**	**Exhaustividad**
Regresión Logística	Entrenamiento	0.579	0.548	0.716
	Prueba	0.606	0.636	0.737
Atípicos (A)	Entrenamiento	0.58	0.541	0.841
	Prueba	0.65	0.65	0.897
Normalización (B)	Entrenamiento	0.517	0.499	0.963
	Prueba	0.616	0.611	0.948
Normalización (C)	Entrenamiento	0.781	0.751	0.814
	Prueba	0.758	0.804	0.776
Eliminación (E)	Entrenamiento	0.779	0.728	0.862
	Prueba	0.697	0.741	0.741
Lasso	Entrenamiento	0.858	0.812	0.915
	Prueba	0.778	0.781	0.862
Ridge	Entrenamiento	0.86	0.816	0.915
	Prueba	0.768	0.778	0.844
Árbol de Decisión	Entrenamiento	0.941	0.941	0.937
	Prueba	0.778	0.821	0.793
Árbol de Decisión (D)	Entrenamiento	1	1	1
	Prueba	0.788	0.878	0.741
PNM	Entrenamiento	0.992	0.995	0.989

	Prueba	0.818	0.955	0.724
PNM (D)	Entrenamiento	0.776	0.715	0.889
	Prueba	0.737	0.767	0.793

Cuadro 4.2: Resultados modelos de clasificación primer año profesional.

Para el valor dicotómico a cuatro años, los resultados son mostrados en cuadro 2.17. Donde mantener o remover los valores atípicos debe ser irrelevante, pero se debe optar por normalización óptima, ya que mejora exactitud y precisión tanto en entrenamiento como en prueba. La exhaustividad contra la exactitud de la regresión logística simple indica que se están clasificando a muchos jugadores como positivos. Al regularizar L1 y L2 las estadísticas de clasificación dan un gran brinco, resaltando la exactitud en *lasso*, sin embargo, en ambos, el modelo es laxo al momento de decidir sobre valores positivos y prefiere incluirlos que dejarlos afuera. Mientras que el árbol de decisión con todas las variables realiza lo contrario, obteniendo una precisión en prueba de 0.91. Siendo el perceptrón neuronal multicapa de las variables *lasso* el que mejor desempeño tiene en el arreglo de prueba, balanceando la precisión y la exhaustividad.

		Exactitud	**Precisión**	**Exhaustividad**
Regresión Logística	Entrenamiento	0.542	0.522	0.995
	Prueba	0.552	0.527	1
Atípicos (A)	Entrenamiento	0.542	0.524	0.984
	Prueba	0.552	0.527	1
Normalización (B)	Entrenamiento	0.517	0.509	1
	Prueba	0.531	0.516	1
Normalización (C)	Entrenamiento	0.605	0.585	0.734
	Prueba	0.583	0.569	0.688
Eliminación (E)	Entrenamiento	0.602	0.583	0.729
	Prueba	0.583	0.569	0.688
Lasso	Entrenamiento	0.877	0.834	0.943
	Prueba	0.823	0.772	0.917
Ridge	Entrenamiento	0.809	0.755	0.917
	Prueba	0.76	0.719	0.854

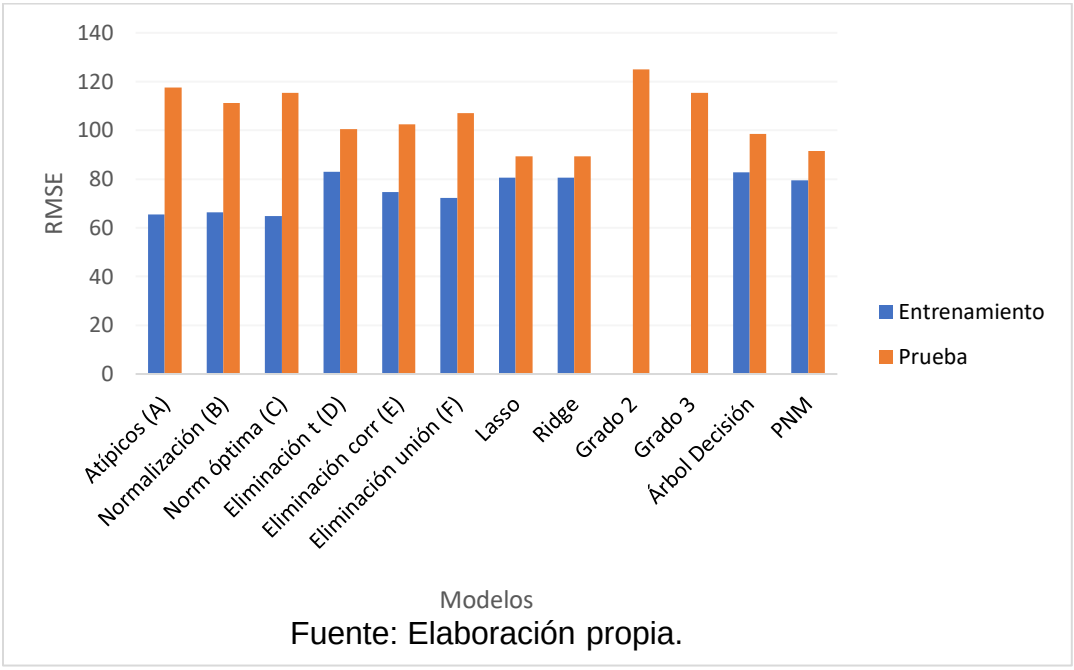
Árbol de Decisión	Entrenamiento	0.95	0.958	0.9427
	Prueba	0.875	0.909	0.833
Árbol de Decisión (D)	Entrenamiento	0.976	0.995	0.958
	Prueba	0.833	0.881	0.771
PNM	Entrenamiento	0.997	1	0.995
	Prueba	0.896	0.865	0.938
PNM (D)	Entrenamiento	0.974	0.984	0.964
	Prueba	0.896	0.896	0.896

Cuadro 4.3: Resultados modelos de clasificación primeros cuatro años profesionales.

4.2 Selección de modelo.

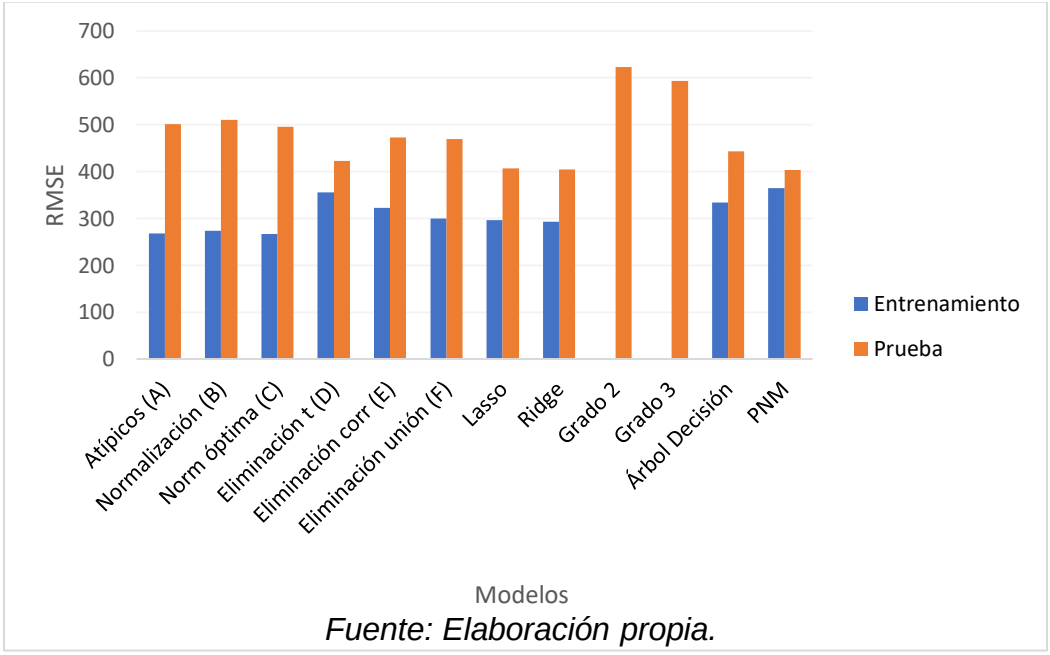
En este apartado se buscará encontrar el mejor modelo tomando en cuenta la información anterior. Para los problemas de regresión se propone la utilización de *lasso, ridge* y perceptrón neuronal multicapa como lo muestra la figura 2.37 y 2.38. Mientras que para los problemas de clasificación se proponen los árboles de decisión y perceptrón neuronal multicapa de acuerdo con la figura 2.39 y 2.40. En los problemas de regresión, buscamos una RMSE de prueba cercana a cero. Sin embargo, una RMSE de entrenamiento baja no quiere decir necesariamente una contraparte de prueba baja. Por ello, es necesario estudiar la disyuntiva sesgo-varianza. Ya que de acuerdo al principio de *lex parsimoniae* o la navaja de Ockham un modelo más sencillo puede ser mejor para predecir la realidad. La figura 2.41 muestra una gráfica con el RMSE de entrenamiento y el de prueba a diferentes coeficientes de regularización. Donde se puede ver que entre más cerca de uno este el coeficiente menor es el RMSE en prueba. No obstante, este coeficiente ayuda a realizar un modelo menos complejo, al no tomar en cuenta algunos coeficientes para ciertas variables. Un *alpha* de uno indica cero penalizaciones de coeficientes por lo que debería de ser un valor de mínimos cuadrados. Sin embargo, como *scikit-learn* utiliza en su función *linear_model.lasso* gradiente estocástico para encontrar el mínimo local, podríamos considerar que generaliza sobre el ajuste de mínimos cuadrados, al no optimizar sobre el mínimo global. Logrando un RMSE en prueba de 88.64.

Figura 4.1: RMSE primera temporada profesional.



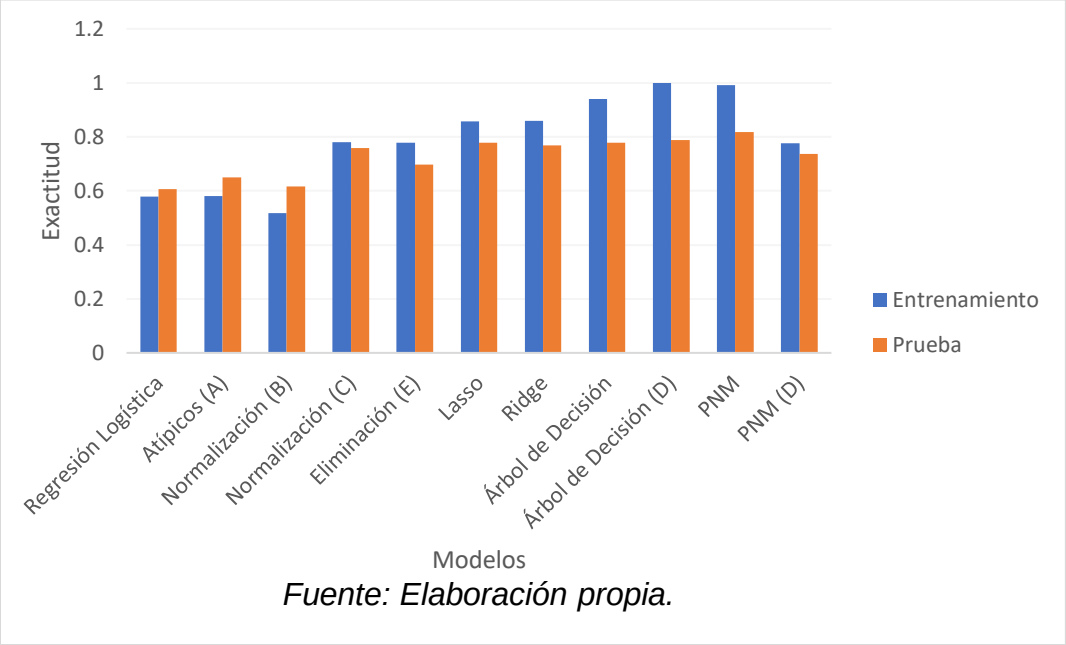
Fuente: Elaboración propia.

Figura 4.2: RMSE de primeros cuatro años de trayectoria profesional.



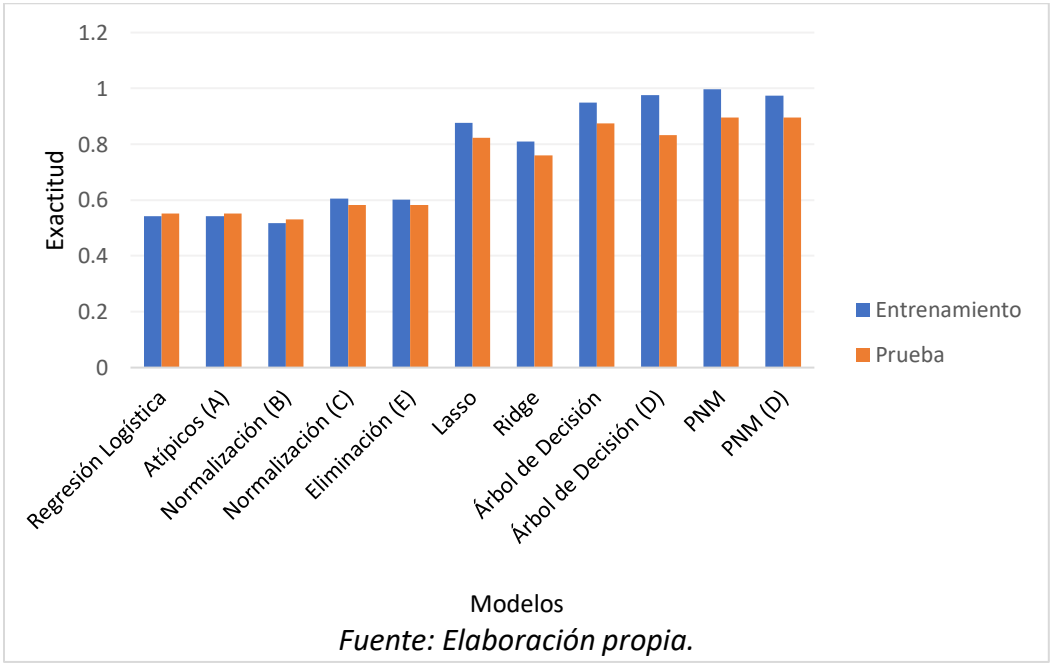
Fuente: Elaboración propia.

Figura 4.3: Exactitud de primera temporada profesional.



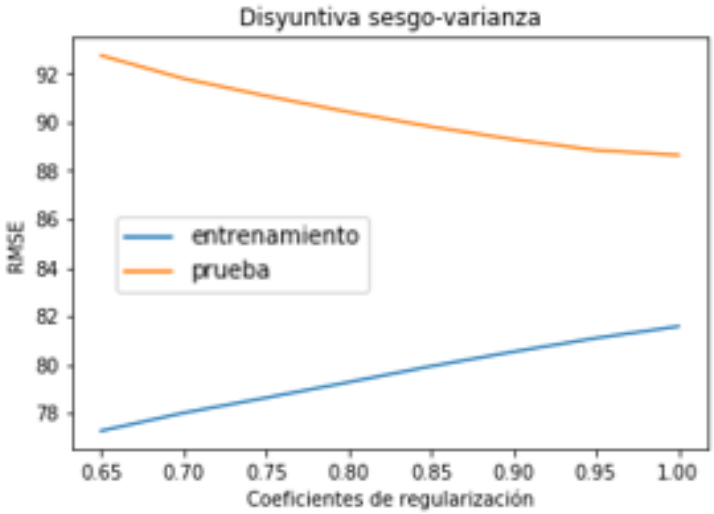
Fuente: Elaboración propia.

Figura 4.4: Exactitud primeros cuatro años profesionales.



Fuente: Elaboración propia.

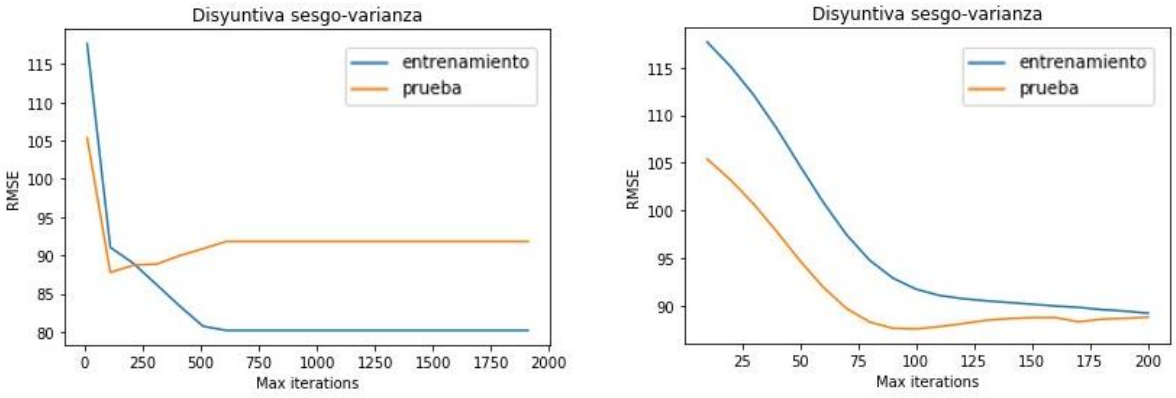
Figura 4.5: Prueba vs. Entrenamiento RMSE primera temporada profesional.



Fuente: Elaboración propia.

Otro modelo que presenta RMSE más bajos que sus contrapartes para el primer año profesional es el perceptrón neuronal multicapa. La figura 2.42 muestra la disyuntiva sesgo-varianza para un máximo de 2000 iteraciones, además se muestra un acercamiento en el punto óptimo, entre diez y doscientas iteraciones con una razón de aprendizaje de 0.001. Encontrando que, a 100 iteraciones con 80 capas ocultas no se llega al punto de convergencia, no obstante, se obtiene la RMSE de prueba más bajo, con un valor de 87.52.

Figura 4.6: Prueba vs. Entrenamiento en iteraciones primera temporada profesional.

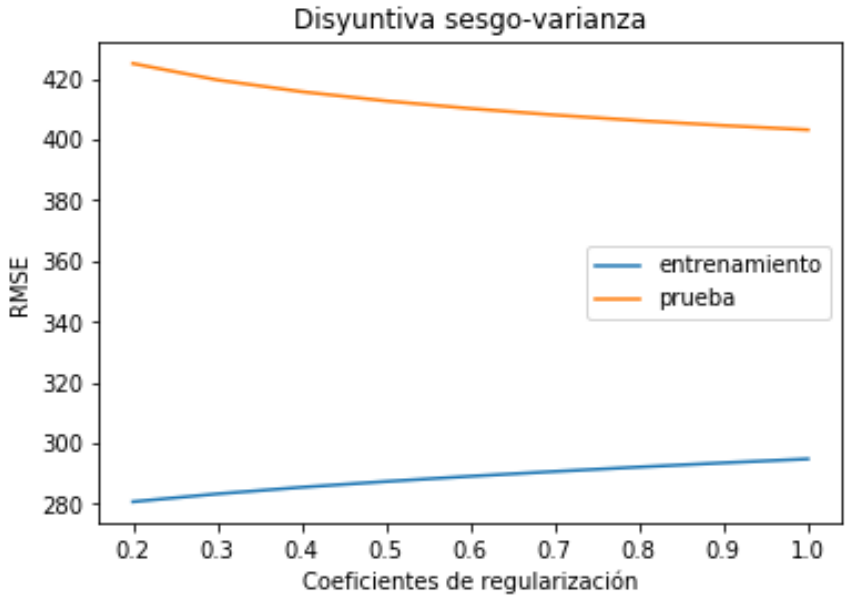


Fuente: Elaboración propia.

Para los primeros cuatro años de trayectoria profesional usando regularización L2 a distintos niveles de *alpha* se puede conocer los valores de entrenamiento y prueba, esperando conocer

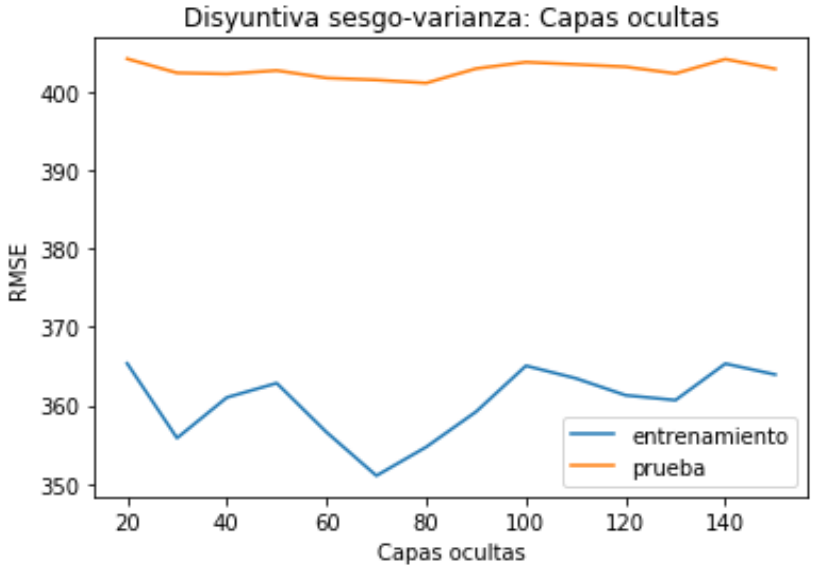
si hay sobreajuste. La figura 2.43 muestra lo anterior, donde la RMSE de entrenamiento es muy inferior a la de prueba, sin embargo, a mayor penalización de las variables, menor la RMSE de prueba con 403.17, por lo que el modelo no hace sobreajuste. Para el regresor neuronal multicapa se iteró sobre la cantidad de capas ocultas, mostrando movimientos en los valores de entrenamiento, mientras que los valores de prueba son prácticamente estáticos. Hallando que con ochenta capas ocultas la RMSE es de 401.13, sin un aparente subajuste, no obstante, no ajustando a los datos como se espera (RMSE más baja) por lo que el algoritmo de inteligencia de máquina no es problemático, de acuerdo a la figura 2.44.

Figura 4.7: Prueba vs. Entrenamiento, RMSE primeras cuatro temporadas profesionales.



Fuente: Elaboración propia.

Figura 4.8: Prueba vs. Entrenamiento, RMSE de perceptrón neuronal multicapa.



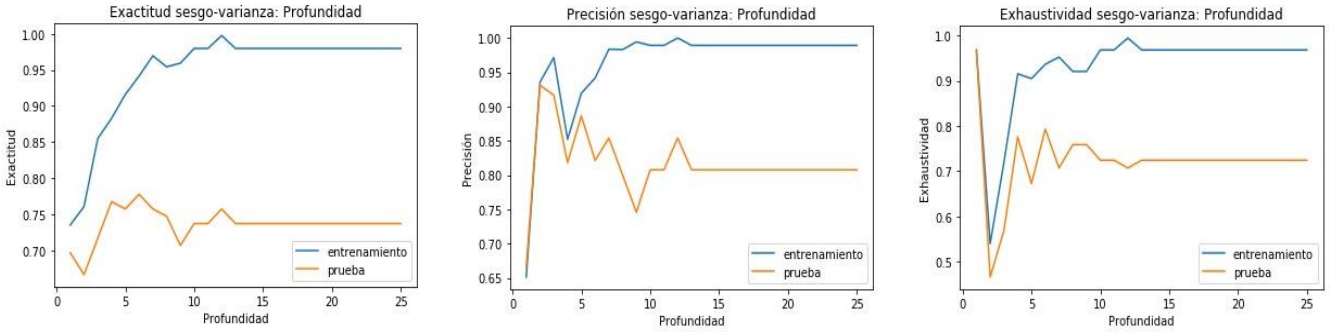
Fuente: Elaboración propia.

Para el modelo de predicción de desempeño de mariscales de campo mediante clasificación, se busca que estos sean altamente precisos, sobre altamente exhaustivos. Buscando que los falsos positivos retornados del modelo sean los menos, a costo de mayor numero de falsos negativos. Para el objetivo de primer año profesional con árbol de decisión teniendo profundidad de dos se maximiza la precisión, asegurándose que por la longitud del árbol el modelo no sobreajusta. Con una exactitud de 0.76 en entrenamiento y 0.66 en prueba, una precisión de 0.94 y 0.93 respectivamente, junto con una exhaustividad de 0.54 y 0.47. Se obtuvo 29 jugadores con desempeño esperado alto, de 99 jugadores que fueron usados como arreglo de prueba y solo dos falsos positivos. Mientras que la mayor exactitud en prueba se obtiene con una profundidad de seis, como se muestra en la figura 2.45, obteniendo 56 jugadores con alto desempeño, 10 de ellos son falsos positivos. Esos valores son obtenidos con un umbral de 0.5, lo cual quiere decir que si el valor continúo obtenido es mayor a 0.5 se le asigna la clase positiva, de lo contrario la negativa. Al subir o bajar el umbral, la cantidad de categorías falsas se ve afectada. La figura 2.46 muestra la curva ROC[46] para un árbol de decisión de profundidad 6, la cual muestra a distintos umbrales la disyuntiva entre falsos positivos y verdaderos positivos, la línea punteada representa un modelo aleatorio y el punto rojo es el umbral 0.791. Donde se obtiene una exactitud y precisión en entrenamiento de 0.94

[46] Característica Operativa del Receptor, del inglés *Receier Operating Characteristic*

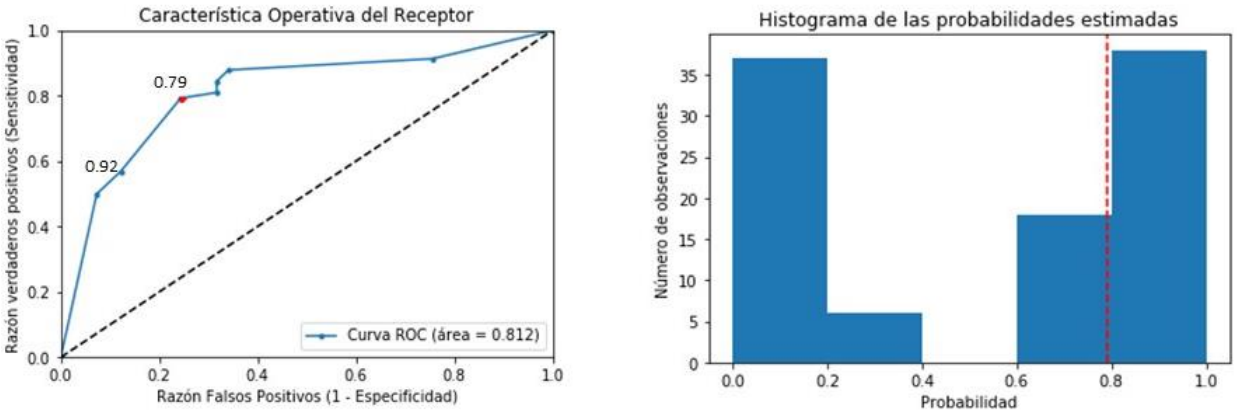
y exhaustividad de 0.94. Mientras que en prueba se obtiene una exactitud de 0.78, una precisión de 0.82 y una exhaustividad de 0.79. También se muestra el histograma de las probabilidades estimadas de desempeño, la línea roja punteada es la mediana, la cual puede servir como punto de referencia para fijar el umbral, ya que esperas que los datos se encuentren balanceados entre clases. A un umbral de 0.8 los positivos retornados son 38 con cinco falsos positivos, mientras que a un umbral de 0.92, los positivos son 32 con tres falsos positivos.

Figura 4.9: Exactitud, Precisión Exhaustividad primer año profesional.



Fuente: Elaboración propia.

Figura 4.10: Curva ROC y probabilidades predichas, primer año profesional.

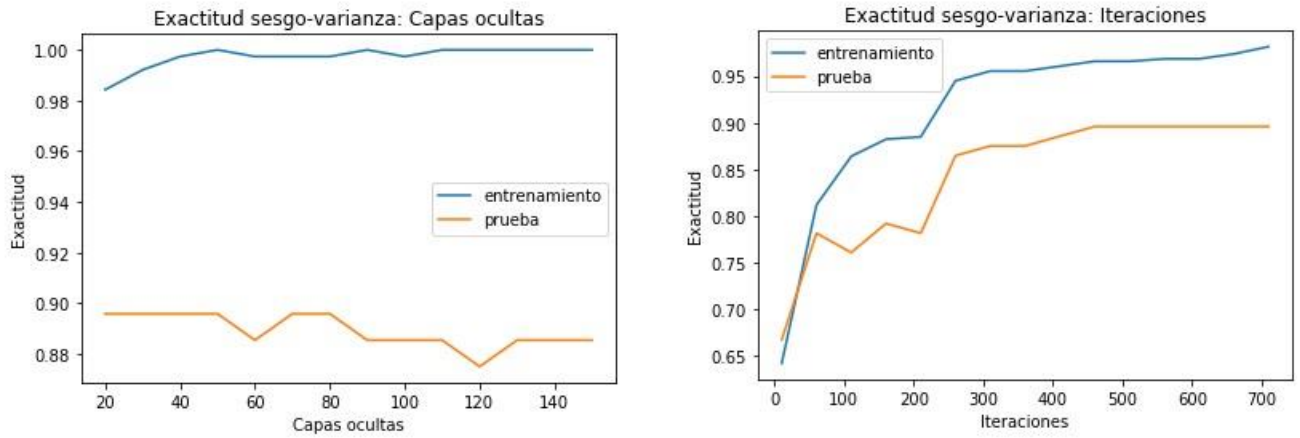


Fuente: Elaboración propia.

Para comprobar la predicción de los primeros cuatro años se utilizó un perceptrón multicapa, iterando a través de distintos valores de capas ocultas, mostrado en la figura 2.47. Veinte

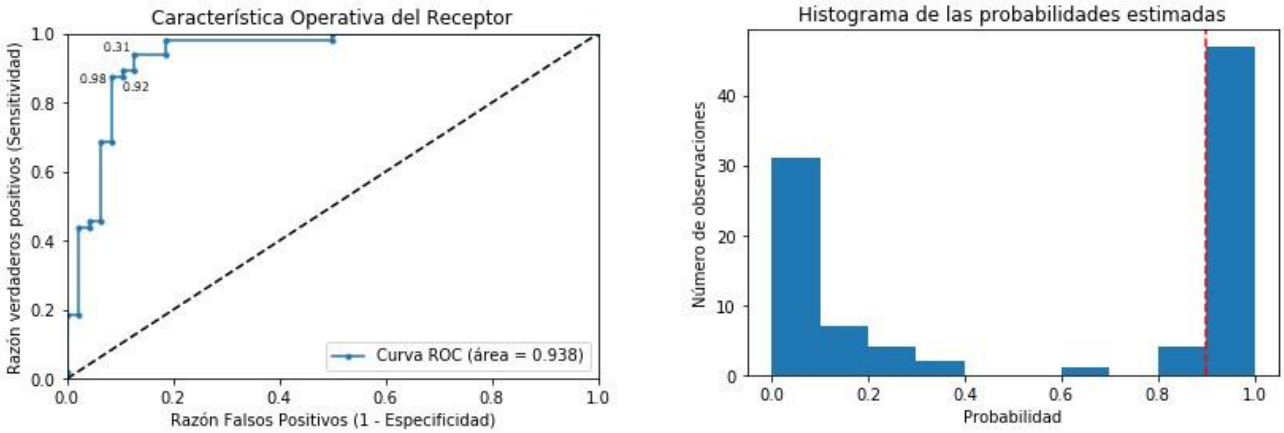
capas ocultas generan valores de exactitud en prueba altos, comparados con más capas. Por lo que se mantendrá el modelo lo más simple posible con 600 iteraciones. Obteniendo una exactitud en entrenamiento de 0.97, una precisión de 0.98 y una exhaustividad de 0.97. Mientras que en prueba se obtuvo 0.9, 0.87 y 0.94 respectivamente. La figura 2.48 muestra la curva ROC y el histograma de las probabilidades en el conjunto de prueba. Ambas figuras ayudaron a fijar el umbral de decisión. La primera muestra grandes incrementos en la tasa de verdaderos positivos sin tener grandes movimientos en la tasa de falsos positivos. Por lo que la curva se encuentra cercano al eje de las ordenadas, lo cual indica un modelo que predice bien, como podemos ver, también está alejada de la línea de punteada de referencia sobre toma de decisiones aleatorias. Al igual que con el objetivo de primer año profesional, buscamos una alta precisión del modelo, evitando los falsos positivos. Bajo un umbral de 0.5 se obtuvieron 52 positivos estimados de un arreglo de 96, de los cuales siete son falsos positivos. Mientras que bajo un umbral de 0.92, los positivos estimados son 45 con cuatro falsos positivos.

Figura 4.11: Exactitud perceptrón neuronal multicapa de primeros cuatro años profesionales.



Fuente: Elaboración propia.

Figura 4.12: Curva ROC y probabilidades predichas, primeros cuatro años profesionales.



Fuente: Elaboración propia.

4.3 Aplicación a la NFL

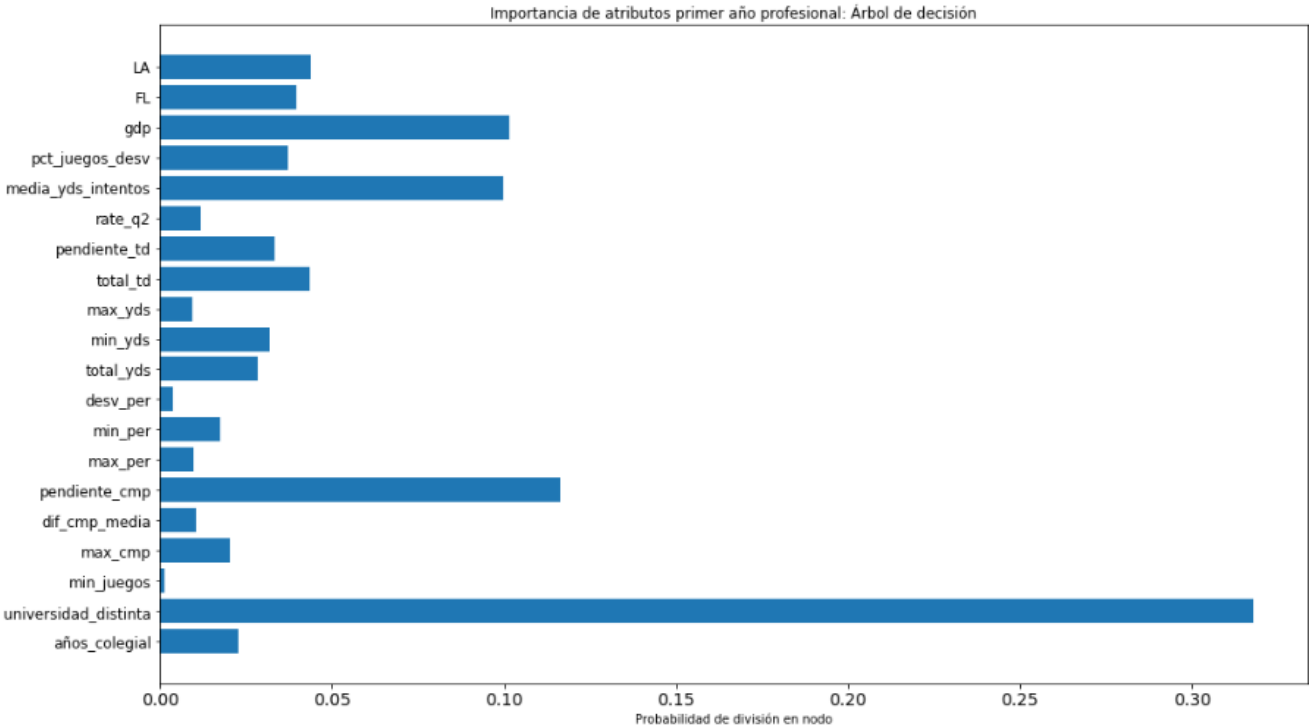
Conocer el desempeño profesional de un jugador dada su información de trayectoria colegial es un proceso que debe ser supervisado por un ser humano y este último también ayudará en la decisión final del modelo. Ya que la tarea de selección de personal requiere de información complementaria, de evaluación de distintas características y en ocasiones de personalización. Por lo tanto, otras habilidades del jugador podrían ser consideradas, como responsabilidad, trabajo en equipo, liderazgo, comunicación, etc. Las cuales son complicadas de integrar a un modelo de inteligencia de máquina. Junto con aquellas características de juego del mariscal de campo, como si prefiere correr, una mente ágil para seleccionar receptores abiertos o pases precisos. Se puede adoptar una estrategia de datos, la cual deberá tener como funciones principales el análisis y la creación de modelos predictivos, todo bajo un fuerte enfoque big data. El motor principal de la toma de decisiones deberán ser los datos, por lo que se buscará que más tareas sean ayudadas por el área de datos. Lo cual no quiere decir que los reclutadores que viajan a distintas universidades para conocer en persona al mariscal de campo, saber si sus calificaciones son buenas, si le gusta salir de fiesta, la opinión de sus compañeros y entrenadores y verlo jugar en vivo, desaparecerán. Pero su trabajo será más eficiente, visitando solo aquellos que tienen grandes posibilidades de tener un alto desempeño en su vida profesional. Podrá estar más tiempo con cada jugador, asegurándose que los datos sean veraces, realizando más pruebas, concentrando esfuerzos y recursos en candidatos con alto desempeño esperado.

De acuerdo a los resultados del apartado anterior, los modelos de regresión no ayudarán a representar la realidad por los altos valores de la desviación de la raíz cuadrada media, tanto en la predicción del primer año profesional y aún más en los primeros cuatro años de trayectoria profesional. Para la detección del personal, como ayuda en la toma de decisiones o para disminuir el conjunto de candidatos se puede utilizar los modelos de clasificación. Sin embargo, los algoritmos de máquina de regresión bien ajustados, con bajos niveles de RMSE, pueden servir para modelos de optimización, donde se busca maximizar las yardas totales o los pases completados de los mariscales de campo, junto con otras estadísticas de otras posiciones bajo la restricción salarial. O la minimización salarial bajo la restricción de mínimas yardas totales esperadas. Además, se puede utilizar programación en enteros con los valores retornados de los algoritmos de clasificación. En caso de usar los algoritmos ML de regresión, surge el problema de que jugadores considerar para analizarlos más profundamente, en pocas palabras, como proponer a dichos jugadores al usuario para que este pueda tomar mejores decisiones; pueden ser los primeros diez, veinte, treinta, cuarenta o más con mayores pases completados, por lo que se vuelve un problema dicotómico, considerar o no considerar a estos jugadores para un mayor análisis. Asimismo, se pueden incorporar actitudes de aceptación de riesgo, mediante la esperanza matemática. Sabiendo cuanto se le pagará al jugador por ser reclutado y la probabilidad de éxito en la liga de fútbol americano profesional, se puede calcular cual será el valor real del jugador.

Por lo tanto, los modelos de clasificación son los escogidos sobre los de regresión. Los valores esperados en el primer año y los primeros cuatro años arrojan resultados muy similares, a pesar, que debería ser más complicado realizar predicciones a mayor tiempo. Debido a los resultados teóricos de ambos modelos (primer año y primeros cuatro años), se puede considerar primero aquellos jugadores que se encuentren en la intersección de ambos modelos con gran desempeño esperado. Para el caso de predicción de desempeño esperado, donde se tiene un gran número de candidatos, la precisión del modelo será vital. Buscando mostrar aquellos mariscales de campo que realmente tendrán altos valores de rendimiento, siendo muy estricto y no permitiendo que jugadores con bajo rendimiento esperado aparezcan como alto rendimiento (positivos) bajo la posible pérdida de mariscales de campo que también tendrán altos valores de rendimiento en el futuro. Lo cual puede ser potencializado con el umbral en la predicción para la decisión dicotómica. Por lo que se preferirán los falsos negativos sobre los falsos positivos, pues el proceso de reclutamiento de la liga profesional de fútbol americano es

muy costoso. Obteniendo 3 falsos positivos en un conjunto de prueba de 99 observaciones para el primer año de temporada profesional y cuatro falsos positivos en 96 observaciones para los primeros cuatro años profesionales.

Figura 4.13: Importancia de variables, árbol de decisión primer año profesional.



Fuente: Elaboración propia.

El algoritmo de árbol de decisión de *scikit-learn* permite conocer la importancia de los atributos, por lo que se puede entender como el modelo toma decisiones mediante las variables que este considera. Aquellas variables con una división que implique más observaciones tendrán una mayor importancia. También servirán para saber cómo se toman decisiones en la actualidad, aunque la probabilidad mostrada en la figura 2.49 permite saber la importancia de una variable específica sobre las demás, no se debe interpretar de acuerdo a las probabilidades mostradas, pero entender que variables tienen efecto en el modelo, ya que no permiten conocer de qué manera afecta la variable dicotómica. La pendiente de pases colegiales completados y su valor máximo son importantes, junto con el máximo, mínimo y total de yardas colegiales, así como la pendiente y el total de anotaciones.

5 Conclusión

El problema de predicción del desempeño esperado de mariscales de campo de la liga de fútbol americano profesional dada su información colegial es factible y conveniente de acuerdo a los resultados teóricos obtenidos en el arreglo de prueba de los distintos modelos de clasificación tanto para la primera temporada colegial como para los consecuentes tres años, acertando en alrededor de 85% de las observaciones de prueba a un año y 90% de las observaciones de prueba en los consecuentes tres años mediante el uso de perceptrones neuronales multicapa, siendo altamente precisos, por lo que aquellos jugadores que el modelo clasifica como positivos (1, desempeño alto), son mínimo 90% probables de ser positivos efectivamente para ambos casos. A pesar que los pronósticos a largo plazo son más complicados, puesto que existe mayor incertidumbre. Sin embargo, conocer los pases completados esperados de los mariscales de campo, mediante problemas de regresión, es complicado y poco preciso por los valores de la desviación cuadrática media y los de RMSE, 91 y 403 respectivamente; por lo que no son concluyentes de la capacidad predictiva para el primer año profesional y en mayor medida para los primeros cuatro años en la NFL. Además, estos modelos de regresión tienen una R-cuadrado inferior a 0.5, por lo que los movimientos en las variables predictoras no explican bien los cambios en la variable objetivo. Siendo solo los modelos basados en perceptrones multicapa, “cajas negras”, los que obtienen mejor RMSE, pero estos modelos no permiten la interpretación de los coeficientes. La estructura de la liga y la información con la que se cuenta en la habilitan y hacen beneficioso el uso de algoritmos dicotómicos ya que presentan altas métricas de exactitud, precisión y exhaustividad bajo una forma semi-automatizada de extracción de datos.

Mientras que la propuesta de la aproximación binomial es una nueva forma, que suma a los trabajos de Wolfson, Addona y Schmicker (2011) para resolver el problema de predicción de mariscales de campo, que descubren ciertos patrones en la relación fútbol colegial-fútbol profesional, más se ven inhabilitados en encontrar un buen predictor de desempeño. Midiendo este último como la cantidad de enfrentamientos jugados y el número total de puntos. En esta investigación se midió el desempeño como el total de pases completados por su relación con las demás variables, ya que nos permitía explicar variables como intentos de pase, intercepciones, yardas y anotaciones. Para lograr predecirlo, se usaron sus estadísticas colegiales de juegos totales, cantidad de universidades a las que se asistió, número de años

en la universidad, puntos anotados, tazones jugados, anotaciones, yardas por pases, lanzamientos completados, pases intentados, intercepciones, producto interno bruto del estado donde jugó fútbol americano en el bachillerato, altura, peso, mano con la que lanza el balón y variables construidas, que nos ayudaran a entender toda la trayectoria colegial del jugador. Sin embargo, más variables podrían ser agregadas al modelo, de acuerdo con las investigaciones de (Williams, Park y Wieling, 2010; Wampole, 2012), como cantidad de testosterona, simetría en la cara y atractividad medida por el género opuesto. Así como las variables resultado del *combine*, de los estudios psicológicos y los exámenes médicos, en busca de un incremento substancial en el error de predicción. Mediante la descomposición dicotómica del problema entre ser considerado y no ser considerado, positivo y negativo respectivamente, se logran menos de cuatro falsos positivos y más de 80 por ciento de exactitud tanto para las predicciones a un año como para aquellas a cuatro años. Pudiendo reducir el conjunto de jugadores a ser observados, logrando de esta manera un conocimiento más profundo de aquellos jugadores con etiqueta positiva, por lo tanto, tomando mejores decisiones ante un problema masivo en datos. Por lo que la decisión final continúa siendo tomada por una persona, si así se prefiere. Es por ello, que el proceso de selección es semi-automatizado, mientras que el de reclutamiento es posible de automatizar.

Por lo anterior y basado en los resultados de los modelos aplicados, aceptamos las Hipótesis

H_1 El aprendizaje de máquina permite mejorar las decisiones sobre el reclutamiento y selección de personal basándose en algoritmos.

H_2 La estructura que ofrece la liga profesional de fútbol americano y el empleo de técnicas big data para impulsar algoritmos machine learning, resulta conveniente la utilización de información colegial (a priori) para predecir el desempeño esperado de mariscales de campo entre las temporadas de 1999 y 2019.

Encontramos que la analítica de grandes datos y machine learning puede ser utilizada como una herramienta adicional en el Reclutamiento y Selección de personal, pero también en la predicción del desempeño del personal, ya que entre más información se tenga mejor los resultados que puede arrojar un algoritmo machine learning. De esta forma se puede agregar a las herramientas tradicionales que los administradores de Recursos Humanos utilizan y les

pueden ayudar a desarrollar trabajos más creativos y que generen valor, acciones que no puede realizar una computadora, por el tiempo que no se invierte en las tareas automatizadas.

No obstante, el hecho de tener tanta información disponible para lograr predicciones basadas en su trayectoria colegial tiene sus desventajas, ya que deben de dejar sus estudios académicos para concentrarse en los entrenamientos y partidos, siendo su trabajo más que un lugar donde aprender y mejorar su rendimiento, por lo que las predicciones no deberían ser tan distantes del fútbol americano profesional pues no solo se conoce sus características como jugador, pero también se conoce el compromiso de cada uno. Las reglas fijadas por la NCAA y la NFL ayudan a tener un mejor mercado laboral, sin embargo, esto se debe al modelo de negocio de la NCAA al restringir la posibilidad de ganar dinero de un jugador amateur o de hacer uso de su imagen de manera comercial. Sumado al hecho de tener que dejar la universidad inconclusa para ser parte del draft, que te llevará a la liga profesional. Y todos aquellos jugadores que no son reclutados durante el draft, podrán firmar contratos profesionales, sin embargo, no tienen nada asegurado, como lo tiene un jugador elegido en draft. Además, el costo de ser jugador elegible durante el draft no es financiado por la universidad, pero por el jugador, en ocasiones siendo por encima de los cien mil dólares (Geeter & Sigalos, 2019), sin la seguridad de que serán elegidos. Mientras que, en la NFL el no poder elegir para que equipo jugar o ganar más dinero tiene un efecto positivo en el resultado general, no obstante, es perjudicial, en ocasiones, para los jugadores profesionales. Sumado a las lesiones que tienen que enfrentar, en ocasiones sin seguro y sin contracto, sabiendo que son completamente desechables (Bradley, 2015). Enfrentando problemas ante el dolor que sienten al final de cada partido, poniendo en riesgo su salud para sentirse mejor, sin tener nadie una organización que se preocupe por su salud, pero solo por el hecho de que sigan jugando y entrenando, día tras día (Gibson , Kurland, & Wyatt, 2015).

Durante el proceso de ciencia de datos, se enfrenta la disyuntiva entre un modelo complejo y el costo, relacionado con la rapidez o lentitud en la entrega del producto. Por lo que se pueden proponer mejores métodos para la realización de un arreglo que sirva como entrada al modelo, pero también modelos más complicados, mejor segmentados, etc. Que tengan como resultado predicciones con menos errores, por lo que se debe saber cuánto valoramos una ínfima mejora en la exactitud de un modelo de clasificación, sobre muchas horas de trabajo. Por ejemplo, para resolver problemas de valores atípicos o de observaciones faltantes se pueden utilizar

métodos no supervisados de inteligencia de máquina como selvas de aislamiento, K-medias y DBSCAN[47]. También se pueden utilizar recursos de aprendizaje profundo especializado, como Tensorflow o Keras [48] que permiten manipular redes neuronales en una plataforma de inteligencia de máquina personalizada. Sumado a las nuevas técnicas de cómputo distribuido que permiten mejorar los tiempos de procesamiento y la capacidad de análisis y manejo de datos[49]. Logrando que más personas tengan acceso mediante el cómputo en la nube, que también admite distribuir de manera remota las actividades para la consecución de un objetivo.

Como se ha visto a lo largo de este trabajo, la creación de un producto de datos es un proceso iterativo, donde se busca un predictor que ajuste bien con datos nunca antes vistos. Por lo que cada industria y cada organización deberá diseñar de manera personalizada sus productos de datos. No existe una aproximación que ajuste para todo tipo de problema, por lo que se debe ser muy creativo al momento de enfrentar los problemas del área de recursos humanos. La extracción de datos de currículos se puede realizar mediante el uso de procesamiento natural de lenguaje, conocer su experiencia profesional y su personalidad mediante las distintas redes sociales disponibles. Debido a la naturaleza de ciertos trabajos, la aplicación de algoritmos inteligentes es superior. Por ejemplo, un académico, se sabe el número de veces que sus trabajos son citados, el número de publicaciones con las que cuenta, la posición a nivel mundial en el departamento universitario en el que trabaja, etc. Mientras que otras profesiones son más complicadas de medir y por lo tanto intentar predecir el desempeño. Como el diseño, no obstante, si se conoce el público objetivo se puede buscar una persona que conecte con dicho público mediante su portafolio de trabajo, se puede calcular su estado de ánimo de acuerdo a sus publicaciones (Behrani, Abbasi & Bhuto, 2017) y se puede conocer cuantas personas la siguen por redes sociales. En caso que el problema no se pueda automatizar o semi-automatizar la aplicación de algoritmos de aprendizaje no tendrá sentido, puesto que durante la búsqueda y manejo de la información se puede conocer cada una de las observaciones. Si los datos recabados son demasiados, se tendrá que pensar la forma de recabarlos automáticamente. Un buen ejemplo de esto, son las páginas web de reclutamiento de candidatos, que mediante un formulario permiten la obtención estructurada de una serie de

[47] Agrupamiento espacial basado en densidad de aplicaciones con ruido. Del inglés "*Density-Based Spatial Clustering of Applications with Noise".*
[48] www.tensorflow.org, www.keras.io.
[49] www.aws.amazon.com/es/sagemaker.

información considerada como relevante para la organización. Sin embargo, para el problema de reclutamiento y selección en la liga profesional de fútbol americano, son tantos los datos con los que se cuenta, que se puede tener una plataforma de inteligencia artificial maestra, que coordine el entrenamiento y prueba de datos para las distintas posiciones. Sumado a la cantidad de esfuerzo y dinero que se destina en encontrar jóvenes estrellas.

Se ha elegido la NFL sobre otros deportes o sobre la aplicación a una industria distinta por la cantidad de datos con los que se cuenta en el fútbol americano profesional. Siendo estos una gran ventaja para la aplicación de inteligencia artificial, sin embargo, un problema que enfrentan las pequeñas y medianas empresas, puesto que su capacidad de manejar datos es limitada y por lo tanto de incorporar nuevas tecnologías para la ayuda en la toma de decisiones. Además, en el fútbol americano se tiene una idea muy clara de las posiciones de cada jugador y que requiere realizar cada uno de estos para alcanzar objetivos a corto y largo plazo. Junto con la capacidad de obtener información sobre su desempeño en el juego sin necesidad de ser contratado; otra gran desventaja que tienen industrias distintas, al no saber cómo se desempeñaran los recién egresados. No obstante, se podría aplicar en personas con experiencia o en aquellas profesiones donde a lo largo de su trayectoria académica se desarrollen portafolios o proyectos. El trabajo de recabar las mejores estadísticas en la NFL y el fútbol americano colegial, no ha surgido de la noche a la mañana, pero ha sido alrededor de mucho tiempo de analizar y comprender el juego para saber que estadísticas son las más importantes. Donde a pesar que ciertos equipos requieren de habilidades específicas, las tareas que debe desempeñar un mariscal de campo están bien acotadas y son las mismas para todos los equipos, por lo que un buen modelo funciona para todos.

Además del poder de predicción de desempeño que tienen estos modelos, pueden funcionar para eliminar sesgos humanos, pero también para comprobar y saber si existen sesgos en los procesos actuales en el área de recursos humanos. Como se puede observar en el apartado anterior, se obtiene la importancia de las variables de ciertos modelos. A través del estudio de la interpretabilidad de inteligencia artificial, mediante librerías como *Shap, LIME* y *eli5*[50]. Se puede conocer la importancia global de las variables, así como la importancia local. Logrando explicar el resultado para cada observación, de prácticamente toda clase de modelos que existen en la actualidad. Por lo que, para detectar sesgos presentes, por ejemplo, en los

[50] www.github.com/slundberg/shap, www.github.com/marcotcr/lime, www.eli5.readthedocs.io/en/latest/.

procesos de reclutamiento y selección; se modela utilizando una serie de variables que nos ayuden a explicar los procesos, junto con variables que expongan si se está discriminando, como género o raza, con el fin de darse cuenta si el modelo está utilizando esas variables para tomar decisiones.

5.1 Futuras investigaciones

Tanto en el campo de los recursos humanos, como en la aplicación que se ha trabajado en esta investigación existen una serie de posibles futuras investigaciones. Como la predicción de atractividad por parte de los fanáticos hacia un jugador en particular mediante análisis de texto en redes sociales. Ya que más venta de artículos de un jugador, son más ingresos a la liga y más personas pueden pronosticar de manera más certera el desempeño esperado de un jugador. Para empresas donde no se tenga tanta información se pueden profundizar en la investigación de algoritmos no supervisados de detección de anomalías, buscando jugadores o colaboradores con desempeño muy alto o muy bajo y solo contratando a los primeros.

Para mantener control alimenticio de los jugadores se podría profundizar en investigaciones de cálculo de calorías mediante visión computacional, teniendo que tomar una fotografía a lo que consumen y registrando información. La última, es otra área de oportunidad para los recursos humanos sabiendo como se puede administrar información de forma eficiente y efectiva en dicha área. Estando muy consciente de los alcances y los retos que presenta esta en términos técnicos, financieros y éticos. También se debe profundizar en la adopción de inteligencia artificial en México. Esta investigación nos permitirá seguir profundizando en el uso de chatbots para comunicación interna en las organizaciones. Pudiendo proveer información a recién ingresados y aquellos colaboradores con más experiencia sobre procesos y estructuras. Además, dicha comunicación con inteligencia artificial o con los demás colaboradores puede funcionar como entrada de un modelo de detección de sentimientos, que debe de ser investigado a nivel de desarrollo y a nivel ético.

Por último, se debe de profundizar la investigación cualitativa en la adopción y entrenamiento de colaboradores en herramientas de inteligencia artificial que sirvan para mejorar la toma de decisiones en las organizaciones. Se debe de buscar que perfiles se ajustan más, que empresas lo están haciendo mejor y que están haciendo. Como dirigir equipos desarrolladores, como mantenerlos motivados y creativos para que se propongan los mejores proyectos.

Apéndice A. Script ELT

```
# -*- coding: utf-8 -*-
"""
Creado en Marzo 25 19:28:43 2020

@author: mauriciomani

Crawler de página web pro-football version en español
"""

#librerías para proceso ELT
from bs4 import BeautifulSoup
import numpy as np
import pandas as pd
# Regresa enteros aleatorios para no generar actividad sospechosa
from random import randint
# Calcular diferenicas con regresion lineal
from scipy.stats import linregress
#Detener tiempo para no ser detectado y llevar control
from time import sleep, time
import urllib3


def años(x = 1999, y=2020):
    """ Reliza una lista de todos los links de QB de 1999 a 2020"""
    qb_urls = dict()
    for i in range(x, y):
        qb = "https://www.pro-football-reference.com/years/" + str(i) +
"/passing.htm"
        qb_urls[i] = qb
    return(qb_urls)

#clase para la extracción de data usando bs4
http = urllib3.PoolManager()

#column_name = data.find('thead').text.split()
#name variables, no considera todas las columnas de la tabla html
#Se debe de considerar reducir este espacio
name_variables = ['player', 'team', 'age', 'pos', 'g', 'gs', 'qb_rec', 'pass_cmp',
                  'pass_att', 'pass_cmp_perc', 'pass_yds', 'pass_td', 'pass_td_perc',
                  'pass_int', 'pass_int_perc', 'pass_first_down', 'pass_long',
                  'pass_yds_per_att', 'pass_adj_yds_per_att', 'pass_yds_per_cmp',
                  'pass_yds_per_g', 'pass_rating', 'pass_sacked', 'pass_sacked_yds',
                  'pass_net_yds_per_att', 'pass_adj_net_yds_per_att',
                  'pass_sacked_perc', 'comebacks', 'gwd']

def extraer_tabla(qb_urls):
    """Extrae tabla de juego de QB en la nfl usando la lista de años"""
    tablas = []
    for o, i in qb_urls.items():
        #obtener html
        sleep(randint(1, 7))
        qb_page = http.request('GET', i)
```

```
        soup = BeautifulSoup(qb_page.data, 'lxml')
        data = soup.find('table', attrs = {'class':'per_match_toggle sortable
stats_table'})
        #Crear un diccionario vacio
        all_qb = dict()
        #Crear una lista vacia para cada llave en el diccionario
        for i in name_variables:
            all_qb[i] = []
        #Añadir una llave vacia y una lista llamada link para cada link del jugador
        all_qb['link'] = []
        ranking = []
        for i in data.find('tbody').find_all('tr'):
            #th define la primera columna, index
            for e in i.find('th'):
                #Si no encuentra información releavante
                if e == 'Rk':
                    pass
                else:
                    ranking.append(e)
            #td Para extraer fila por fila
            for a in i.find_all('td'):
                #loop a traves de columnas de la tabla
                for u in name_variables:
                    #Extraer data por coda elemento en la lista columna
                    if a['data-stat'] == u:
                        #Extraer el link del jugador para aplicación araña de la
web y anexar el link a la llave
                        if u == 'player':
                            all_qb['link'].append(a.find('a', href=True)['href'])
                        #Para las demas columnas de información
                        all_qb[u].append(a.text)
        #Añadir rankinfg como sistema de indice al pandas dataframe
        df = pd.DataFrame(all_qb , index = ranking)
        #o es la llave del diccionario de año
        df['year'] = o
        tablas.append(df)
    return(tablas)

toda_tabla = extraer_tabla(años())
#Si se maneja menor espacio, considerar reducir el espacio dimensional
una_tabla = pd.concat(toda_tabla, axis = 0, ignore_index = True)
una_tabla.to_csv('nfl_dissertation/nfl_qb.csv', index=False)
una_tabla= pd.read_csv('nfl_dissertation/nfl_qb.csv')

def qb_nfl_estadistica(link_qb_page):
    """Almacena tablas con datos de pases tomand en cuenta el link de
extraer_tabla"""
    data_pase = []
    #ids de tablas a extraer
    ids = ['passing', 'passing_playoffs', 'passing_advanced',
'ks_passing_detailed_air_yards']
    qb_pagina = http.request('GET', link_qb_page)
    #elimiar etiqueta comentada
    qb_data = qb_pagina.data.decode('utf-8').replace('<!--', '').replace('-->', '')
```

```
    soup = BeautifulSoup(qb_data, 'lxml')
    #Anexar función de información del jugador salida de data de pases: indice: 0
    data_pase.append(info_jugador(soup))
    for i in ids:
        #Encontrar tablas queridas
        data = soup.find('table', attrs = {'id': i})
        if i == 'passing_playoffs' or i == 'passing':
            #Si se encuenta la estructura en html purp anexa pandas a una lista
(para método pd.concat despues)
            try:
                data_pase.append(pd.read_html(str(data)))
            except ValueError:
                data_pase.append(None)
        #Si la tabla no es pase o pase_postemporada se puede encontrar una distinta
estructura (reciente drafted qb)
        else:
            try:
                data_pase.append(pd.read_html(str(data)))
            except ValueError:
                pass
    return(data_pase)


def info_jugador(soup):
    """Información del jugador en esta función, parte superior de la pagina web"""
    name = soup.find('h1', attrs={'itemprop':'name'}).text.strip()
    print(name)
    #Extrae data html con información del jugador
    data = soup.find('div', attrs = {'id': 'meta'})
    #Romper problema Big Data en algo mas pequeño y mas sencillo
    info = data.find_all('p')
    #Extare mano de tiro
    transforme = transforme_raw(info[1])
    posicion = transforme.find('Throws:')
    throws = transforme[posicion + 7:]
    #extrae altura y peso
    transforme = transforme_raw(info[2])
    posicion = transforme.find('(')
    height_weight = transforme[posicion:]
    posicion = height_weight.find('cm')
    #extrae altura
    height = height_weight[1:posicion]
    posicion2 = height_weight.find('kg')
    #extrae peso
    weight = height_weight[posicion + 3 : posicion2]
    transforme = transforme_raw(info[3])
    #Jugadores que siguen jugando tendran equipos como parte de su informacion
    #Si encontrado, usar otra estructura
    if transforme.find('Team') >= 0:
        # Lugar de nacimiento
        transforme = transforme_raw(info[4])
        posicion = transforme.find('in')
        posicion2 = transforme.find(',', posicion)
        place = transforme[posicion + 2 : posicion2]
```

```
        #Estado de nacimiento
        state = transforme[posicion2 + 1 : ]
        posicion = transforme.find(',')
        #Año de nacimiento
        year = transforme[posicion + 1 : posicion + 5]
        #Link a estadisticas colegiales
        college_stats = info[5].find_all('a')[-1]['href']
        transforme = transforme_raw(info[5])
        posicion = transforme.find('College:')
        posicion2 = transforme.find('(')
        #Extraccion colegial
        college = transforme[posicion + 8 : posicion2].strip()
        transforme = transforme_raw(info[7])
        posicion = transforme.find('High School:')
        posicion2 = transforme.find('(')
        #Bachillerato
        high_school = transforme[posicion+12:posicion2].strip()
        #Estado del bachillerato
        state_high_school = transforme[posicion2 + 1 :].replace(')', '')
        posicion = transforme.find('Draft')
        #Si no se encuentra información del drat, entonces una arreglo de texto
vacio
        try:
            transforme = transforme_raw(info[8])
            draft = transforme[posicion + 7 :].strip()
        except IndexError:
            transforme = ''
            draft = ''
    #Si equipo no es encontrado, entonces seguir la estructura siguiente
    else:
        # Lugar de nacimiento
        transforme = transforme_raw(info[3])
        posicion = transforme.find('in')
        posicion2 = transforme.find(',', posicion)
        place = transforme[posicion + 2 : posicion2]
        #Estado de nacimiento
        state = transforme[posicion2 + 1 : ]
        posicion = transforme.find(',')
        #Año de nacimiento
        year = transforme[posicion + 1 : posicion + 5]
        #Link a estadisticas colegiales
        college_stats = info[4].find_all('a')[-1]['href']
        transforme = transforme_raw(info[4])
        posicion = transforme.find('College:')
        posicion2 = transforme.find('(')
        #Extraccion colegial
        college = transforme[posicion + 8 : posicion2].strip()
        transforme = transforme_raw(info[6])
        posicion = transforme.find('High School:')
        posicion2 = transforme.find('(')
        #Bachillerato
        high_school = transforme[posicion+12:posicion2].strip()
        #Estado de bachillerato
        state_high_school = transforme[posicion2 + 1 :].replace(')', '')
```

```
        posicion = transforme.find('Draft')
        try:
            transforme = transforme_raw(info[7])
            draft = transforme[posicion + 7 :].strip()
        except IndexError:
            transforme = ''
            draft = ''
    #regresa lista con observaciones
    return([name, throws, height, weight, place, state, year, college_stats,
college,
            high_school, state_high_school, draft])

def transforme_raw(raw):
    """trasnforma raw html para lograr extracción de jugadores"""
    info = raw.text.replace('\n', '').replace('\t', '').replace('\xa0', '').strip()
    return(info)

def todo_qb_nfl_estadistica(tab):
    """Crea tabla pandas de info_jugador y qb_nfl_estadistica. Extracción de datos
inicia aqui"""
    #ID para cada jugador, iniciar con 1
    numero_identificacion = 1
    info = []
    passing = []
    passing_playoffs = []
    passing_advanced = []
    #La información extraida no incuye todo el link
    pre_link = "https://www.pro-football-reference.com/"
    for i in tab:
        i = pre_link + i
        inicio = time()
        sleep(randint(1, 7))
        #Mantener control del tiempo esperado
        waiting = time() - inicio
        #Intentar extraer la web del enesimo qb
        try:
            lista_estadistica = qb_nfl_estadistica(i)
            #Saber cuantos QB se han extraido
            print("Número de petición: {}, Tiempo de espera:
{}".format(numero_identificacion, waiting))
            #Mantener numero de identificacion para join
            lista_estadistica[0].append(numero_identificacion)
            #print(lista_estadistica[0][0])
            info.append(lista_estadistica[0])
            lista_estadistica[1][0]['id'] = numero_identificacion
            if type(lista_estadistica[1][0].columns) == pd.MultiIndex:
                lista_estadistica[1][0].columns =
lista_estadistica[1][0].columns.droplevel()
            else:
                pass
            passing.append(lista_estadistica[1][0])
            #Usar try y except en caso de que no se encuentre informacion de
postemprada o pases avanzados
            try:
```

```
                lista_estadistica[2][0]['id'] = numero_identificacion
                #Revisar si dataframe tiene multi-indice, de ser asi eliminar, de
otra manera pasar
                if type(lista_estadistica[2][0].columns) == pd.MultiIndex:
                    lista_estadistica[2][0].columns =
lista_estadistica[2][0].columns.droplevel()
                else:
                    pass
                passing_playoffs.append(lista_estadistica[2][0])
            except:
                pass
            #Lo mismo, try y except en caso de que no se encuentre pase avanzado
            try:
                lista_estadistica[3][0]['id'] = numero_identificacion
                #Revisar si dataframe tiene multi-indice, de ser asi eliminarlo, de
otra manera pasar
                if type(lista_estadistica[3][0].columns) == pd.MultiIndex:
                    lista_estadistica[3][0].columns =
lista_estadistica[3][0].columns.droplevel()
                else:
                    pass
                passing_advanced.append(lista_estadistica[3][0])
            except:
                pass
            #Añadir 1 para cada jugador
            numero_identificacion += 1
        #Si la página tiene un error (esta vacia): pasar
        except:
            pass
    #Concatenar las listas del dataframe antes de ser creado
    df_pase = pd.concat(passing, axis = 0, ignore_index=True, sort= False)
    df_pase_postemporada = pd.concat(passing_playoffs, axis = 0, ignore_index=True,
sort= False)
    df_pase_advanzado = pd.concat(passing_advanced, axis = 0, ignore_index=True,
sort= False)
    columns = ['name', 'throws', 'height', 'weight', 'place', 'state',
'birth_year','link_college', 'college', 'high_school', 'state_high_school','draft',
'id']
    df_info = pd.DataFrame(info, columns = columns)
    return(df_info, df_pase, df_pase_postemporada, df_pase_advanzado)

qb_unico = np.unique(una_tabla[(una_tabla['pos'] == '') | (una_tabla['pos'] =='QB')
| (una_tabla['pos'] =='qb') | (una_tabla['pos'] == 'fl/QB') | (una_tabla['pos'] ==
'fb/QB')]['link'])
df_info, df_pase, df_pase_postemporada, df_pase_advanzado =
todo_qb_nfl_estadistica(qb_unico)

df_info.to_csv('nfl_dissertation/qb_info.csv', index=False)
df_pase.to_csv('nfl_dissertation/qb_nfl_pase.csv', index=False)
df_pase_postemporada.to_csv('nfl_dissertation/qb_nfl_pase_postemporada.csv',
index=False)
df_pase_advanzado.to_csv('nfl_dissertation/qb_nfl_pase_avanzado.csv', index=False)
df_info = pd.read_csv('nfl_dissertation/qb_info.csv')
pase = pd.read_csv('nfl_dissertation/qb_nfl_pase.csv', dtype = {'QBrec':object})
```

```
postemporada = pd.read_csv('nfl_dissertation/qb_nfl_pase_postemporada.csv')

def extraccion_tabla_colegial(df_info):
    """Extraer informacion de la universidad tomando data de df_info, de
todo_qb_nfl_estadistica"""
    pase_college = []
    link_id = df_info[df_info['link_college'].str.contains("www.sports-
reference.com/cfb/players/")][['link_college', 'id']]
    for i in link_id.iterrows():
        #Mantener contol de tiempo como arriba
        inicio = time()
        sleep(randint(1, 7))
        #Monitorear tiempo de espera
        waiting = time() - inicio
        #obtener link con urllib3 y bs4
        qb_college = http.request('GET',  i[1][0])
        soup = BeautifulSoup(qb_college.data, 'lxml')
        data = soup.find('table', attrs = {'id':'passing'})
        #Encontrar el nombre del qb
        name = soup.find('h1', attrs ={'itemprop':'name'}).text
        #Si no qb, información de pase no será disponible
        try:
            pase = pd.read_html(str(data))[0]
            if type(pase.columns) == pd.MultiIndex:
                pase.columns = pase.columns.droplevel()
            else:
                pass
            #Añadir el id al dataframe
            pase['id'] = int(i[1][1])
            #Añadir el nombre a todas las observaciones
            pase['name'] = name
            pase_college.append(pase)
            print(name)
            print("Request number: {}, Time elapsed: {}".format(i[1][1], waiting))
        except ValueError:
            pass
    df_colegial = pd.concat(pase_college, axis = 0, ignore_index = True, sort =
False)
    return(df_colegial)

df_colegial = extraccion_tabla_colegial(df_info)
df_colegial.to_csv('nfl_dissertation/qb_colegial.csv', index = False)
df_colegial = pd.read_csv('nfl_dissertation/qb_colegial.csv')

def extraccion_escuela(df_colegial):
    """"Extraer tabla de universidadExract table of school (college) all years"""
    info_college = []
    #Only extracts necessary college info
    universidades = df_colegial.School.unique()
    pre_link = 'https://www.sports-reference.com/cfb/schools/'
    for i in universidades:
        #Replace unecessary characters for school link fit
        school_link = i.lower().replace('(', '').replace(')','').replace(' ', '-')
        #Process of extraction follows former processes
```

```
        sleep(randint(1, 7))
        school_page = http.request('GET', pre_link + school_link)
        soup = BeautifulSoup(school_page.data, 'lxml')
        try:
            #Extract all college history information
            data = soup.find('table', attrs = {'id':  school_link })
            print("Extracting: " + school_link)
            #To pandas
            school = pd.read_html(str(data))[0]
            school['name'] = i
            info_college.append(school)
        except ValueError:
            pass
    #Concat all schools (college) information
    all_schools = pd.concat(info_college, axis = 0)
    return(all_schools)

df_universidad =  extraccion_escuela(df_colegial)
df_universidad.to_csv('nfl_dissertation/df_universidad.csv', index = False)
df_universidad = pd.read_csv('nfl_dissertation/df_universidad.csv')

def fusion_colegial(df_universidades, df_colegial):
    """Transforma información de universidad y fusiona con estadisticas qb
colegiales"""
    #Extrae si gana tazon durante universidad
    df_universidades['win_bowl'] = df_universidades['Bowl'].str.extract('\-(.)')
    df_universidades['bowl'] = df_universidades['Bowl'].str.extract('(.*)\-')
    #Elimina información inecesaria
    df_universidades.drop(['Rk', 'Conf', 'Coach(es)', 'Bowl', 'Notes'], axis = 1,
inplace = True)
    df_universidades.drop(df_universidades[df_universidades.Year == 'Year'].index,
axis = 0, inplace = True)
    #Cambia el tipo de columna
    df_universidades['Year'] = df_universidades['Year'].astype('int')
    df_universidades['Pct'] = '0' + df_universidades['Pct'].str[:]
    df_universidades['Pct'] = df_universidades['Pct'].astype(float)
    df_universidades[['W', 'L']] = df_universidades[['W', 'L']].astype(int)
    #Elimina filas corruptas en la columna de años
    df_colegial.drop(df_colegial[df_colegial.Year == 'Career'].index, inplace =
True)
    df_colegial.dropna(subset = ['Year'], axis = 0, inplace = True)
    df_colegial['year'] = df_colegial['Year'].str.replace(r'*', '').astype(int)
    #Fusiona estadisticas colegiales y universidad
    df_colegial = pd.merge(df_colegial, df_universidades,  how='left',
left_on=['School','year'], right_on = ['name','Year'])
    return(df_colegial)


def transformacion_informacion_colegial(df_colegial, df_info):
    """Merge info from nfl, like weight and height to aggregated information from
college"""
    #Lista vacia
    college_agg = []
    for i in df_colegial.id.unique():
```

```
        df = df_colegial[df_colegial['id'] == i]
        #Escalar los años para un ajuste apropiado de la regresion lineal
        scaled_year = (df.year - df.year.min())/ (df.year.max() - df.year.min())
        percentage = df.Pct_x / 100
        #Crear la regresion lineal de ciertas columnas
        reg_g = linregress(scaled_year, df['G'])[0]
        reg_cmp = linregress(scaled_year, df['Cmp'])[0]
        reg_att = linregress(scaled_year, df['Att'])[0]
        reg_yds = linregress(scaled_year, df['Yds'])[0]
        reg_td = linregress(scaled_year, df['TD'])[0]
        reg_int = linregress(scaled_year, df['Int'])[0]
        #Crear lista con elementos (alrededor) 4 años
        college_agg.append([df.id.iloc[0], df.name_x.iloc[0], df.year.count(),
len(df.School.unique()), df.G.sum(), df.G.min(), df.G.max(),df.G.diff().mean(),
reg_g, df.Cmp.sum(), df.Cmp.min(), df.Cmp.max(), df.Cmp.diff().mean(), reg_cmp,
df.Att.sum(), df.Att.min(), df.Att.max(), df.Att.diff().mean(), reg_att,
percentage.max(), percentage.min(), percentage.mean(), percentage.std(),
df.Yds.sum(), df.Yds.min(), df.Yds.max(),df.Yds.diff().mean(), reg_yds,
df.TD.sum(), df.TD.min(), df.TD.max(), df.TD.diff().mean(), reg_td, df.Int.sum(),
df.Int.min(), df.Int.max(), df.Int.diff().mean(), reg_int, df.Rate.quantile(0.25),
df.Rate.median(), df.Rate.quantile(0.75), df['Y/A'].max(), df['Y/A'].min(),
df['Y/A'].mean(),df['Y/A'].std(), df.year.iloc[0], (df.win_bowl == 'W').sum(),
df.bowl.count(), df.W.sum(),df.L.sum(), df.Pct_y.mean(), df.Pct_y.std()])
    #Lista de columnas para entrada de parametros dataframe
    columns = ['id', 'nombre', 'años_colegial', 'universidad_distinta',
'juegos_totales', 'min_juegos', 'max_juegos', 'dif_juegos_media',
'pendiente_juego', 'suma_cmp', 'min_cmp', 'max_cmp', 'dif_cmp_media',
'pendiente_cmp', 'suma_intentos', 'min_intentos', 'max_intentos',
'diff_intentos_media', 'pendiente_intentos', "max_per", "min_per", "media_per",
"desv_per", 'total_yds', 'min_yds', 'max_yds', 'dif_yds_media', 'pendiente_yds',
'total_td', 'min_td', 'max_td', 'dif_td_media', 'pendiente_td', 'total_int',
'min_int', 'max_int', 'dif_int_media', 'pendiente_int', 'rate_q1', 'rate_q2',
'rate_q3', 'max_yds_intentos', 'min_yds_intentos', 'media_yds_intentos',
'desv_yds_intentos', 'min_año', 'bowls_ganados', 'bowls_jugados', 'juegos_ganados',
'juegos_perdidos', 'pct_juegos', 'pct_juegos_desv']
    agg_college = pd.DataFrame(college_agg,  columns=columns)
    #Fusionar con info de nfl (throws, height, weight and state high school)
    agg_college = pd.merge(agg_college, df_info[['id', 'throws', 'height',
'weight', 'state_high_school']], on = 'id', how = 'left')
    return(agg_college)


def fusion_pib_colegial(agg_college):
    """Fusiona PIB de agregado colegial. Informacion descargada de:

https://apps.bea.gov/itable/iTable.cfm?ReqID=70&step=1#reqid=70&step=1&isuri=1"""
    #Data modificada en editor de texto para unir con abreviacion de estado y
eliminación de columna (GeoFips) y filas para facilitar el join
    #eliminado 1997 de 1963-1997 archivo csv
    #importar info PIB
    gdp1 = pd.read_csv('nfl_dissertation/pib_estado_1963_1997.csv')
    gdp2 = pd.read_csv('nfl_dissertation/pib_estado_1997_2018.csv').drop('GeoName',
axis = 1)
    #Concatena ambos PIB para mejor manipulacion
```

```
    gdp = pd.concat([gdp1, gdp2], axis = 1)
    #Pivotea data. Una columna año, otra columna estado
    gdp = pd.melt(gdp, id_vars = ['GeoName'], value_vars = list(gdp.columns[1:]))
    gdp['variable'] = gdp.variable.astype('int64')
    gdp.columns = ['state_ab', 'year', 'gdp']
    #fusiona colegio e informacion de PIB
    universidad_final = pd.merge(agg_college, gdp, how = 'left',
left_on=['state_high_school','min_año'], right_on = ['state_ab','year'])
    return(universidad_final)


def fusion_universidad_nfl(passing, universidad_final):
    """Fusiona informacion colegial e informacio de la para prediccion"""
    #eliminar filas inecesarias
    passing.drop(passing[passing.Year == 'Career'].index, inplace = True)
    passing.dropna(subset = ['Year'], axis = 0, inplace = True)
    passing = passing[~passing.Year.str.contains('(yr)')]
    #Extraer informacion de ganados y perdidos usando QBRec
    passing['Year'] = passing['Year'].str.replace(r'*', '').str.replace(r'+',
'').astype(int)
    passing['QBrec'] = passing['QBrec'].str.replace(r'/', '-').astype(str)
    passing['win'] = passing.QBrec.str.extract('(\d{1,2})-').astype(float, errors =
'ignore')
    passing['lose'] = passing.QBrec.str.extract('\d{1,2}-(\d{1,2})-').astype(float,
errors = 'ignore')
    nfl_stat = []
    for i in passing.id.unique():
        #Monitorea informacion de id
        stats = passing[passing['id'] == i]
        info = df_info[df_info['id'] == i]
        try:
            #Extrae información del draft
            draft = int(info.draft.str.extract('\((\d*)').iloc[0, 0])
            year_draft = int(info.draft.str.extract('(\d{4})').iloc[0, 0])
        except ValueError:
            pass
        #Mantiene los primeros cuatro años, a menos que sean menos
        if len(stats) < 4:
            ff = stats.iloc[0:len(stats), :]
        else:
            ff = stats.iloc[0:4, :]
        #Anexa data
        nfl_stat.append([ff.G.sum(), ff.G.iloc[0], ff.GS.sum(), ff.G.iloc[0] -
ff.GS.iloc[0], ff.Cmp.sum(), ff.Cmp.iloc[0], ff.Att.sum(), ff.Att.iloc[0],
ff.G.sum() - ff.GS.sum(), ff.TD.sum(), ff.TD.iloc[0], ff.Yds.sum(),ff.Yds.iloc[0],
i, draft, year_draft, ff.Year.iloc[0], ff.win.iloc[0], ff.lose.iloc[0],
ff.win.sum(), ff.lose.sum()])
    #Lista de columnas para dataframe
    columns = ['nfl_tot_juegos_totales', 'nfl_primeros_juegos',
'nfl_tot_juegos_iniciados', 'nfl_primeros_jug_vs_iniciados', 'nfl_tot_cmp',
'nfl_primero_cmp', 'nfl_tot_intentos', 'nfl_pimero_intentos',
'nfl_tot_jugados_vs_iniciados', 'nfl_tot_TD', 'nfl_primeros_TD', 'nfl_tot_yds',
'nfl_primeros_yds', 'id', 'draft', 'año_draft','nfl_primer_año',
```

```
'nfl_primero_ganado', 'nfl_primero_perdido', 'nfl_total_ganado',
'nfl_total_perdido']
    nfl_qb_data = pd.DataFrame(nfl_stat, columns = columns)
    #Fusiona data colegial y data universidades (4 años)
    college_mas_nfl = pd.merge(universidad_final, nfl_qb_data, how = 'left', on =
'id')
    return(college_mas_nfl)

def postemporada_fusion_universidad_nfl(college_mas_nfl, postemporada):
    """informacion postemporada. Si QB tiene juegos de postemporada en los primeros
4 años"""
    #Elimina filas no necesarias (pases, universidades y pases postemporada)
    postemporada.drop(postemporada[postemporada.Year == 'Career'].index, inplace =
True)
    postemporada.dropna(subset = ['Year'], axis = 0, inplace = True)
    postemporada = postemporada[~postemporada.Year.str.contains('(yr)')]
    postemporada['Year'] = postemporada['Year'].str.replace(r'*',
'').str.replace(r'+', '').astype(int)
    #Extrae informacion de cuenta en año
    postemporada = pd.merge(postemporada, college_mas_nfl[['id',
'nfl_primer_año']], how='left', on = 'id')
    postemporada['primeros_cuatro_años'] = postemporada.nfl_primer_año + 4
    postemporada['tot_postemporada'] = (postemporada['Year'] <
postemporada['primeros_cuatro_años']).astype(int)
    postemporada = postemporada.groupby('id', as_index = False).agg({'Year':
['count', 'min'], 'tot_postemporada': 'sum'})
    #Fusiona postemporada y data de fusion_nfl_universidad
    final_dataset = pd.merge(college_mas_nfl, postemporada, how = 'left', on='id')
    return(final_dataset)

def fusion_final(df_universidades, df_colegial, df_info, pase, postemporada):
    """Workflow de las fuciones anteriores. LLamada a todas las funciones"""
    df_colegial = fusion_colegial(df_universidades, df_colegial)
    agg_college = transformacion_informacion_colegial(df_colegial, df_info)
    df_universidades_pib = fusion_pib_colegial(agg_college)
    college_mas_nfl = fusion_universidad_nfl(pase, df_universidades_pib)
    final_data = postemporada_fusion_universidad_nfl(college_mas_nfl, postemporada)
    return(final_data)

final_data = fusion_final(df_universidad, df_colegial, df_info, pase, postemporada)
final_data.to_csv('nfl_dissertation/final_datas.csv', index = False)
```

Apéndice B. Árbol de sistema de archivos

```
nfl_dissertation
        |--- final_data.csv
        |--- pib_estado_1963_1997.csv
        |--- pib_estado_1997_2018.csv
        |--- nfl_qb.csv
        |--- qb_colegial.csv
        |--- qb_nfl_info.csv
        |--- qb_nfl_pase.csv
        |--- qb_nfl_pase_avanzado.csv
        |--- qb_nfl_pase_postemporada.csv
        |--- README.md
        |--- df_universidad.csv
        |--- scouting.yaml
        |--- elt.py
        |--- scraping_qb_nfl.PNG
        |--- EDA.ipynb
        |--- clasificacion.ipynb
        |--- clasificacion_cuatro.ipynb
        |--- regresion.ipynb
        |--- regresion_cuatro.ipynb
        |--- tesis
                |--- Data_science_ai
                |--- inspiracion
                |--- papers
                |--- RH
                |--- Scopus_hr_ai
                |--- metodologia_de_la_investigacion_sampieri.pdf
                |--- tesis.pptx
                |--- trabajo_escrito.docx
                        |--- papers
                        |--- scopus_hr_ai.csv
                        |--- scopus_hr_ai.xlsx
```

Apéndice C. Código HTML

Ejemplo Brett Favre de https://www.pro-football-reference.com/players/F/FavrBr00.htm

HTML información general semiestructurada.

```
<html>
<body>
[…]
</script><div class="search" role="search" aria-label="Site Search for players,
teams and sections">
<form method="get" name="f_big"  action="/search/search.fcgi">
<div class="ac-outline">
  <div class="ac-wrapper"><input type="search" tabindex="-1" class="ac-hint"
name="hint" placeholder="" autocomplete="off" autocorrect="off"
autocapitalize="off" spellcheck="false" dir="auto">
<input tabindex="1" type="search" class="ac-input completely" name="search"
placeholder="Enter Person, Team, Section, etc" aria-label="Enter a player, team
or section name" autocomplete="off" autocorrect="off" autocapitalize="off"
spellcheck="false" dir="auto" />
    <div class="ac-dropdown"></div>
  </div>
</div>
<input type="submit" value="Search" tabindex="2" />
<input type="hidden" name="pid" value="" data-search-id>
<input type="hidden" name="idx" value="" data-search-idx>
</form>
</div><!-- div.search -->
</div><!-- div#header -->
<div id="info" class="players">
       <div id="meta">
<div class="media-item"><img itemscope="image"
src="https://d395i9ljze9h3x.cloudfront.net/req/20180910/images/headshots/FavrBr00
_2017.jpg" alt="Photo of Brett Favre">
</div><!-- div.media-item --><div itemscope itemtype="https://schema.org/Person"
>
       <h1 itemprop="name">Brett Favre </h1>
<p>
  <strong>
    Brett Lorenzo Favre

      (<span itemprop="nickname">Country or  Gunslinger</span>)

  </strong>
</p>
<p>
       <strong>Position</strong>: QB
                    <strong>Throws:</strong>
              Right
</p>
<p><span itemprop="height">6-2</span>, <span
itemprop="weight">222lb</span> (188cm, 100kg) </p>
<p>
  <strong>Born:</strong>
```

```
    <span itemprop="birthDate" id="necro-birth" data-birth="1969-10-10">
                <a href="/friv/birthdays.cgi?month=10&day=10">October10
</a>, 1969
    </span>
  <span itemprop="birthPlace">
    in Gulfport, <a href="/friv/birthplaces.cgi?state=MS">MS</a></span>
</p>
<p><strong>College</strong>:
                <a href="/schools/somississippi/">Southern Miss</a>

                 (<a href="https://www.sports-
reference.com/cfb/players/brett-favre-1.html">College Stats</a>)
<p><strong><a href="/blog/?page_id=518">Weighted Career AV (100-95-
...)</a>:</strong> 156 (7th overall since 1960)
</p>
<p><strong>High School</strong>:
                <a href="/schools/high_schools.cgi?id=93bbd846">Hancock North
Central</a> (<a href="/schools/high_schools.cgi?hs_state=MS">MS</a>)
</p>
<p><strong>Draft</strong>: <a href="/teams/atl/1991_draft.htm">Atlanta
Falcons</a> in the 2nd round (33rd overall) of the <a
href="/years/1991/draft.htm">1991 NFL Draft</a>.</p>
<p><strong>Hall of Fame</strong>: Inducted as
                Player in <a href="/hof/2016-ballot.htm">2016</a>
(<a href="/hof/">Full List</a>)</p>
  <p><strong>Pronunciation</strong>: \<a title="Click for all players'
pronunciations" href="/friv/pronunciation-guide.htm">FARV</a>\</p>
<button id="meta_more_button" class="opener" data-type="hide_after" data-
class="open" data-id="info">More bio, uniform, draft, salary info</button>
</div>
</div><!-- div#meta -->
 […]
```

HTML tabla pases información estructurada.

```
[…]
<div class="table_outer_container">
      <div class="overthrow table_container" id="div_passing">
  <table class="row_summable per_match_toggle sortable stats_table" id="passing"
data-cols-to-freeze="2"><caption>Passing Table</caption>
<colgroup><col><col><col><col><col><col><col><col><col><col><col><col><col><col><
col><col><col><col><col><col><col><col><col><col><col><col><col><col><col><col><c
ol><col></colgroup>
   <thead>
      <tr>
         <th aria-label="Year" data-stat="year_id" scope="col" class=" poptip
sort_default_asc left" >Year</th>
         <th aria-label="Player's age on December 31st of that year" data-
stat="age" scope="col" class=" poptip sort_default_asc center" data-tip="Player's
age on December 31st of that year" >Age</th>
         <th aria-label="Tm" data-stat="team" scope="col" class=" poptip
sort_default_asc show_partial_when_sorting left" >Tm</th>
```

```
          <th aria-label="PositionIn player and team season stats,Capitals
indicates primary starter.Lower-case means part-time starter." data-stat="pos"
scope="col" class=" poptip sort_default_asc left" data-tip="<b>Position</b><br>In
player and team season stats,<br>Capitals indicates primary starter.<br>Lower-
case means part-time starter." >Pos</th>
          <th aria-label="Uniform number" data-stat="uniform_number" scope="col"
class=" poptip sort_default_asc center" data-tip="Uniform number" >No.</th>
          <th aria-label="Games" data-stat="g" scope="col" class=" poptip center"
data-tip="Games played" >G</th>
          <th aria-label="Games Started" data-stat="gs" scope="col" class=" poptip
center" data-tip="Games started as an offensive or defensive player<br />Numbers
are complete for 1920-59<br />and 1970-present but are incomplete otherwise"
>GS</th>
          <th aria-label="QB Record" data-stat="qb_rec" scope="col" class=" poptip
center" data-tip="Team record in games started by this QB (regular season)"
>QBrec</th>
          <th aria-label="Passes Completed" data-stat="pass_cmp" scope="col"
class=" poptip center" data-tip="Passes completed" >Cmp</th>
          <th aria-label="Pass Attempts" data-stat="pass_att" scope="col" class="
poptip center" data-tip="Passes attempted" >Att</th>
          <th aria-label="Pass Completion %" data-stat="pass_cmp_perc" scope="col"
class=" poptip hide_non_quals center" data-tip="Percentage of Passes
Completed<br>Minimum 14 attempts per scheduled game to qualify as leader.<br
/>Minimum 1500 pass attempts to qualify as career leader." data-filter="1" data-
name="Pass Completion %" >Cmp%</th>
          <th aria-label="Passing Yds" data-stat="pass_yds" scope="col" class="
poptip center" data-tip="Yards Gained by Passing<br>For teams, sack yardage is
deducted from this total" >Yds</th>
          <th aria-label="Passing TD" data-stat="pass_td" scope="col" class="
poptip center" data-tip="Passing Touchdowns" >TD</th>
          <th aria-label="Passing TD %" data-stat="pass_td_perc" scope="col"
class=" poptip hide_non_quals center" data-tip="Percentage of Touchdowns Thrown
when Attempting to Pass <br>Minimum 14 attempts per scheduled game to qualify as
leader.<br />Minimum 1500 pass attempts to qualify as career leader" data-
filter="1" data-name="Passing TD %" >TD%</th>
          <th aria-label="Passes Intercepted" data-stat="pass_int" scope="col"
class=" poptip center" data-tip="Interceptions thrown" >Int</th>
          <th aria-label="Pass Intercept. %" data-stat="pass_int_perc" scope="col"
class=" poptip sort_default_asc hide_non_quals center" data-tip="Percentage of
Times Intercepted when Attempting to Pass <br>Minimum 14 attempts per scheduled
game to qualify as leader.<br />Minimum 1500 pass attempts to qualify as career
leader.<br />" data-filter="1" data-name="Pass Intercept. %" >Int%</th>
 […]
</body>
</html>
```

Apéndice D. Código Análisis de datos exploratorio.

```
# coding: utf-8

# # Análisis de datos exploratorio en Arreglo final
# Correr scrape.py para extraer los archivos separados por coma.
# * nfl_qb;
# * qb_college;
# * qb_nfl_info;
# * qb_nfl_passing_advanced;
# * qb_nfl_passing;
# * schools and
# * final_dataset.<br>

# Para más información ir a README.md

import pandas as pd
import matplotlib.pyplot as plt
from scipy.stats import sem, t
from scipy import mean
import seaborn as sns
from scipy.stats import skew

df = pd.read_csv('final_data.csv')
df.columns

# ### Gráfico de dispersión entre todas las variables
#sns.pairplot(df[['juegos_totales', 'min_juegos', 'max_juegos', 'pendiente_juego',
'nfl_primeros_yds']])
#sns.pairplot(df[['suma_cmp', 'min_cmp', 'max_cmp', 'dif_cmp_media',
'pendiente_cmp', 'nfl_primeros_yds']])
#sns.pairplot(df[['suma_intentos', 'min_intentos', 'max_intentos',
'diff_intentos_media', 'pendiente_intentos', 'nfl_primeros_yds']])
#sns.pairplot(df[['total_yds', 'min_yds', 'max_yds', 'dif_yds_media',
'pendiente_yds', 'nfl_primeros_yds']])

pd.set_option('display.max_columns', None)
df.head()

#Revisar que cada id sea distinto
df.id.count() == len(df.id.unique())

#nombre de QB élite
top_qb = ['Warren Moon', 'Ben Roethlisberger', 'Russell Wilson', 'Steve Young',
'Troy Anikman', 'Brett Favre', 'Aaron Rodgers', 'Dan Marino', 'Drew Brees', 'Peyton
Manning', 'Tom Brady', 'Cam Newton', 'Matt Ryan', 'Andrew Luck', 'Philip Rivers']

df[df.nombre.isin(top_qb)]

#Problemas con mano lanzada
df.groupby('throws')['id'].count()

# ### La mayoría de QB solo tienen una universidad.
df.groupby('universidad_distinta')['id'].count()
```

```
#df.groupby('throws')['id'].count()
226/(226+15)

# ### Probablemente no tengamos demasiada información si el jugador solo ha jugado durante uno o dos años y el modelo no ajuste bien para dichos jugadores
df.groupby('años_colegial')['id'].count().plot.bar(color='green')
plt.title('Años colegial por jugador')
plt.ylabel('Cantidad de jugadores')
plt.show()

#Altura QB
df.height.hist()
plt.title('Altura de QB')
plt.ylabel('Suma de jugadores')
plt.xlabel('Altura en cm')
plt.show()

#Peso QB
df.weight.hist()
plt.title('Peso de QB')
plt.ylabel('Suma de jugadores')
plt.xlabel('Peso en kg')
plt.show()

(df.total_td/df.años_colegial).hist()
plt.title('Total anotaciones fútbol colegial')
plt.xlabel('Total anotaciones anuales')
plt.ylabel('Número de jugadores')
plt.show()

(df.total_int/df.años_colegial).hist()
plt.title('Total intercepciones fútbol colegial')
plt.xlabel('Total intercepciones anuales')
plt.ylabel('Número de jugadores')
plt.show()

(df.total_yds/df.años_colegial).hist()
plt.title('Total yardas por pase fútbol colegial')
plt.xlabel('Total yardas por pase anuales')
plt.ylabel('Número de jugadores')
plt.show()

(df.suma_intentos/df.años_colegial).hist()
plt.title('Total pases intentados fútbol colegial')
plt.xlabel('Total de pases intentados anuales')
plt.ylabel('Número de jugadores')
plt.show()

(df.suma_cmp/df.años_colegial).hist()
plt.title('Total pases completados fútbol colegial')
plt.xlabel('Total de pases completados anuales')
plt.ylabel('Número de jugadores')
plt.show()
```

```
(df.juegos_totales/df.años_colegial).hist()
plt.title('Juegos totales fútbol colegial')
plt.xlabel('juegos totales')
plt.ylabel('Número de jugadores')
plt.show()

# ### Diferencias min_max de juegos colegiales
#plot parallel min and max games
df.min_juegos.hist(alpha=0.8, label='min_juegos')
df.max_juegos.hist(alpha=0.5, label ='max_juegos')
plt.legend(loc='upper left')
plt.title('Histogramas MIN vs. MAX juegos')
plt.ylabel('Suma de jugadores')
plt.xlabel('Número de juegos')
plt.show()

#plot parallel min and max games
df.min_cmp.hist(alpha=0.8, label='min_cmp')
df.max_cmp.hist(alpha=0.5, label ='max_cmp')
plt.legend(loc='upper right')
plt.title('Histogramas MIN vs. MAX pases completados')
plt.ylabel('Suma de jugadores')
plt.xlabel('Número de pases completados')
plt.show()

#plot parallel min and max games
df.min_intentos.hist(alpha=0.8, label='min_intentados')
df.max_intentos.hist(alpha=0.5, label ='max_intentados')
plt.legend(loc='upper right')
plt.title('Histogramas MIN vs. MAX pases intentados')
plt.ylabel('Suma de jugadores')
plt.xlabel('Número de pases intentados')
plt.show()

#plot parallel min and max games
df.min_yds.hist(alpha=0.8, label='min_yds')
df.max_yds.hist(alpha=0.5, label ='max_yds')
plt.legend(loc='upper right')
plt.title('Histogramas MIN vs. MAX yardas por pases')
plt.ylabel('Suma de jugadores')
plt.xlabel('Número de yardas por pases')
plt.show()

#plot parallel min and max games
df.min_td.hist(alpha=0.8, label='min_td')
df.max_td.hist(alpha=0.5, label ='max_td')
plt.legend(loc='upper right')
plt.title('Histogramas MIN vs. MAX anotaciones por pase')
plt.ylabel('Suma de jugadores')
plt.xlabel('Número de anotaciones')
plt.show()

#plot parallel min and max games
```

```
df.min_int.hist(alpha=0.8, label='min_int')
df.max_int.hist(alpha=0.5, label ='max_int')
plt.legend(loc='upper right')
plt.title('Histogramas MIN vs. MAX intercepciones')
plt.ylabel('Suma de jugadores')
plt.xlabel('Número de intercepciones')
plt.show()

df[['dif_juegos_media', 'dif_cmp_media', 'diff_intentos_media', 'dif_yds_media',
'dif_td_media', 'dif_int_media']].describe()
df[['dif_juegos_media', 'pendiente_juego', 'dif_cmp_media', 'pendiente_cmp',
'diff_intentos_media', 'pendiente_intentos', 'dif_yds_media', 'pendiente_yds',
'dif_td_media', 'pendiente_td', 'dif_int_media', 'pendiente_int']].corr()
cor = df[['pendiente_juego', 'pendiente_cmp', 'pendiente_intentos',
'pendiente_yds', 'pendiente_td', 'pendiente_int']].corr()
cor.style.background_gradient(cmap='Oranges')

(df.juegos_ganados/df.años_colegial).hist()
plt.title('Juegos ganados')
plt.xlabel('Número de juegos')
plt.ylabel('Suma de Jugadores')
plt.show()

#df.groupby('bowls_ganados')['id'].count()
pd.DataFrame({'jugados':[58,44,55,69,74,3],'ganados':[109, 90, 55, 43, 5, 1]},
index = [0,1,2,3,4,5]).plot.bar()
#plot.bar()
#(label='Bowls jugados')
#df.bowls_ganados.plot.bar(label ='Bowls ganados')
#plt.legend(loc='upper right')
plt.title('Gráfico de barras tazones jugados vs. ganados')
plt.ylabel('Suma de jugadores')
plt.xlabel('Número de tazones')
plt.show()

df.rate_q1.hist(alpha=0.9, label='Q1')
df.rate_q2.hist(alpha=0.6, label ='Q2')
df.rate_q3.hist(alpha=0.4, label ='Q3')
plt.legend(loc='upper right')
plt.title('Histogramas de razón de eficiencia del QB')
plt.ylabel('Suma de jugadores')
plt.xlabel('Valores de la razón')
plt.show()

df[['rate_q1', 'rate_q2', 'rate_q3']].corr()

df.groupby('state_ab')['pendiente_cmp'].count().plot.bar(figsize=(20,10))
plt.title('Número de jugadores por estado')
plt.ylabel('Suma de jugadores')
plt.rc('xtick', labelsize=20)
plt.show()

df.groupby('state_high_school')['id'].count()
```

```
df[(df['state_high_school'] != 'ND') & (df['state_high_school'] != 'MN') &
(df['state_high_school'] != 'MO') & (df['state_high_school'] != 'MT') &
(df['state_high_school'] != 'KY') & (df['state_high_school'] != 'MD') &
(df['state_high_school'] != 'CT') & (df['state_high_school'] != 'DC') &
(df['state_high_school'] != 'IA') & (df['state_high_school'] != 'ID') &
(df['state_high_school'] != 'KS') & (df['state_high_school'] != 'AR') &
(df['state_high_school'] != 'CO') & (df['state_high_school'] != 'HI') &
(df['state_high_school'] != 'NM') & (df['state_high_school'] != 'SC') &
(df['state_high_school'] != 'UT') & (df['state_high_school'] !=
'WI')][['state_high_school', 'nfl_primero_cmp']].boxplot(by='state_high_school',
figsize=(20,10))
plt.title('Diagrama de cajas de yardas completadas primera temporada profesional
por estado')
plt.ylabel('Pases completados primera temporada profesional')
plt.xlabel('Estado de bachillerato')
plt.show()

df.gdp.hist()
plt.title('Histograma de PIB por jugador')
plt.ylabel('Suma de jugadores')
plt.xlabel('Valor del PIB por estado')
plt.show()

df[df.nombre.isin(top_qb)]['nfl_primeros_juegos'].mean()

# ### Entender la diferencia entre QB élite y todos los datos
plt.rc('xtick', labelsize=12)
df['nfl_primeros_juegos'].hist()
plt.axvline(df[df.nombre.isin(top_qb)]['nfl_primeros_juegos'].mean(), color='r',
linestyle='dashed', linewidth=2)
plt.title('Número de juegos en primera temporada profesional')
plt.xlabel('Juegos')
plt.ylabel('Suma de jugadores')
plt.show()

plt.rc('xtick', labelsize=12)
df['nfl_primeros_jug_vs_iniciados'].hist()
plt.axvline(df[df.nombre.isin(top_qb)]['nfl_primeros_jug_vs_iniciados'].mean(),
color='r', linestyle='dashed', linewidth=2)
plt.title('Número de enfrentamientos jugados no iniciados')
plt.xlabel('Juegos')
plt.ylabel('Suma de jugadores')
plt.show()

plt.rc('xtick', labelsize=12)
df['nfl_primero_cmp'].hist()
plt.axvline(df[df.nombre.isin(top_qb)]['nfl_primero_cmp'].mean(), color='r',
linestyle='dashed', linewidth=2)
plt.title('Número de pases completados primera temporada')
plt.xlabel('Pases completados')
plt.ylabel('Suma de jugadores')
plt.show()

plt.rc('xtick', labelsize=12)
```

```
df['nfl_pimero_intentos'].hist()
plt.axvline(df[df.nombre.isin(top_qb)]['nfl_pimero_intentos'].mean(), color='r',
linestyle='dashed', linewidth=2)
plt.title('Número de pases intentados primera temporada')
plt.xlabel('Pases intentados')
plt.ylabel('Suma de jugadores')
plt.show()

plt.rc('xtick', labelsize=12)
df['nfl_primeros_TD'].hist()
plt.axvline(df[df.nombre.isin(top_qb)]['nfl_primeros_TD'].mean(), color='r',
linestyle='dashed', linewidth=2)
plt.title('Número de anotaciones primera temporada')
plt.xlabel('Anotaciones')
plt.ylabel('Suma de jugadores')
plt.show()

plt.rc('xtick', labelsize=12)
df['nfl_primeros_yds'].hist()
plt.axvline(df[df.nombre.isin(top_qb)]['nfl_primeros_yds'].mean(), color='r',
linestyle='dashed', linewidth=2)
plt.title('Número de yardas por pase primera temporada')
plt.xlabel('Yardas por pase')
plt.ylabel('Suma de jugadores')
plt.show()

plt.rc('xtick', labelsize=12)
df['nfl_primero_ganado'].hist()
plt.axvline(df[df.nombre.isin(top_qb)]['nfl_primero_ganado'].mean(), color='r',
linestyle='dashed', linewidth=2)
plt.title('Número de juegos ganados en primera temporada')
plt.xlabel('Juegos Ganados')
plt.ylabel('Suma de jugadores')
plt.show()

plt.rc('xtick', labelsize=12)
df['nfl_primero_perdido'].hist()
plt.axvline(df[df.nombre.isin(top_qb)]['nfl_primero_perdido'].mean(), color='r',
linestyle='dashed', linewidth=2)
plt.title('Número de juegos perdidos en primera temporada')
plt.xlabel('Juegos perdidos')
plt.ylabel('Suma de jugadores')
plt.show()

df[['nfl_primeros_juegos', 'nfl_primeros_jug_vs_iniciados', 'nfl_primero_cmp',
'nfl_pimero_intentos', 'nfl_primeros_TD', 'nfl_primeros_yds', 'nfl_primero_ganado',
'nfl_primero_perdido']].rename(columns={'nfl_primeros_juegos': 'juegos',
'nfl_primeros_jug_vs_iniciados': 'no_iniciados', 'nfl_primero_cmp': 'cmp',
'nfl_pimero_intentos': 'intentos', 'nfl_primeros_TD': 'TD', 'nfl_primeros_yds':
'yds', 'nfl_primero_ganado': 'ganado','nfl_primero_perdido':
'perdido'}).corr().style.background_gradient(cmap='coolwarm')

df1 = df[df['nfl_primer_año']<=2016]
```

```
plt.rc('xtick', labelsize=12)
df1['nfl_juegos_totales'].hist()
plt.axvline(df1[df1.nombre.isin(top_qb)]['nfl_juegos_totales'].mean(), color='r',
linestyle='dashed', linewidth=2)
plt.title('Número de juegos totales primeros 4 años')
plt.xlabel('Juegos')
plt.ylabel('Suma de jugadores')
plt.show()

plt.rc('xtick', labelsize=12)
df1['nfl_tot_juegos_iniciados'].hist()
plt.axvline(df1[df1.nombre.isin(top_qb)]['nfl_tot_juegos_iniciados'].mean(),
color='r', linestyle='dashed', linewidth=2)
plt.title('Número de enfrentamientos jugados no iniciados primeros 4 años')
plt.xlabel('Juegos no iniciados')
plt.ylabel('Suma de jugadores')
plt.show()

plt.rc('xtick', labelsize=12)
df1['nfl_tot_cmp'].hist()
plt.axvline(df1[df1.nombre.isin(top_qb)]['nfl_tot_cmp'].mean(), color='r',
linestyle='dashed', linewidth=2)
plt.title('Número de pases completados primeros 4 años')
plt.xlabel('Pases completados')
plt.ylabel('Suma de jugadores')
plt.show()

plt.rc('xtick', labelsize=12)
df1['nfl_tot_intentos'].hist()
plt.axvline(df1[df1.nombre.isin(top_qb)]['nfl_tot_intentos'].mean(), color='r',
linestyle='dashed', linewidth=2)
plt.title('Número de intentos de pases primeros 4 años')
plt.xlabel('Pases intentados')
plt.ylabel('Suma de jugadores')
plt.show()

plt.rc('xtick', labelsize=12)
df1['nfl_tot_TD'].hist()
plt.axvline(df1[df1.nombre.isin(top_qb)]['nfl_tot_TD'].mean(), color='r',
linestyle='dashed', linewidth=2)
plt.title('Número de anotaciones por pase primeros 4 años')
plt.xlabel('Anotaciones')
plt.ylabel('Suma de jugadores')
plt.show()

plt.rc('xtick', labelsize=12)
df1['nfl_tot_yds'].hist()
plt.axvline(df1[df1.nombre.isin(top_qb)]['nfl_tot_yds'].mean(), color='r',
linestyle='dashed', linewidth=2)
plt.title('Número de yardas por pase primeros 4 años')
plt.xlabel('Yardas')
plt.ylabel('Suma de jugadores')
plt.show()
```

```
plt.rc('xtick', labelsize=12)
df1['nfl_total_ganado'].hist()
plt.axvline(df1[df1.nombre.isin(top_qb)]['nfl_total_ganado'].mean(), color='r',
linestyle='dashed', linewidth=2)
plt.title('Número de juegos ganados primeros 4 años')
plt.xlabel('Juegos Ganados')
plt.ylabel('Suma de jugadores')
plt.show()

plt.rc('xtick', labelsize=12)
df1['nfl_total_perdido'].hist()
plt.axvline(df1[df1.nombre.isin(top_qb)]['nfl_total_perdido'].mean(), color='r',
linestyle='dashed', linewidth=2)
plt.title('Número de partidos perdidos primeros 4 años')
plt.xlabel('Juegos perdidos')
plt.ylabel('Suma de jugadores')
plt.show()

df[[ 'bowls_ganados', 'juegos_ganados', 'height', 'weight',  'juegos_totales',
'min_juegos', 'max_juegos', 'dif_juegos_media', 'pendiente_juego', 'suma_cmp',
'min_cmp', 'max_cmp', 'dif_cmp_media','pendiente_cmp', 'suma_intentos',
'min_intentos', 'max_intentos', 'diff_intentos_media', 'pendiente_intentos',
'max_per', 'min_per','media_per', 'desv_per', 'total_yds', 'min_yds', 'max_yds',
'dif_yds_media', 'pendiente_yds', 'total_td', 'min_td', 'max_td', 'dif_td_media',
'pendiente_td', 'total_int', 'min_int', 'max_int', 'dif_int_media',
'pendiente_int', 'rate_q1', 'rate_q2', 'rate_q3', 'max_yds_intentos',
'min_yds_intentos', 'media_yds_intentos', 'desv_yds_intentos', 'gdp',
'nfl_tot_cmp', 'nfl_tot_yds', 'nfl_primero_cmp',
'nfl_primeros_yds']].corr()[['nfl_tot_cmp', 'nfl_tot_yds', 'nfl_primero_cmp',
'nfl_primeros_yds']].to_csv('C:/Users/mauri/Desktop/Trabajos_LicAdm/proyecto_titula
cion/nfl_dissertation/tesis/eda/correlacion.csv')

df.draft.describe()
skew(df.draft)

df[['draft', 'nfl_primeros_yds', 'nfl_primero_cmp']].corr()

confidence = 0.95
data = df[df.nombre.isin(top_qb)]['nfl_primeros_yds']
n = len(data)
m = data.mean()
std_err = sem(data)
h = std_err * t.ppf((1 + confidence) / 2, n-1)
df.nfl_primeros_yds.plot.hist()
plt.axvline(df[~df.nombre.isin(top_qb)]['nfl_primeros_yds'].mean(), color='r',
linewidth=3)
plt.axvline(df[df.nombre.isin(top_qb)]['nfl_primeros_yds'].mean(), color='g',
linewidth=3)
plt.axvline(m-h, color='g', linestyle='dashed', linewidth=2)
plt.axvline(m+h,  color='g', linestyle='dashed', linewidth=2)
plt.title('Yardas por pase en primera temporada profesional')
plt.ylabel('Número de jugadores')
plt.xlabel('Yardas por pase')
plt.show()
```

```
data = df[df.nombre.isin(top_qb)]['nfl_primero_cmp']
n = len(data)
m = data.mean()
std_err = sem(data)
h = std_err * t.ppf((1 + confidence) / 2, n-1)
df.nfl_primero_cmp.plot.hist()
plt.axvline(df[~df.nombre.isin(top_qb)]['nfl_primero_cmp'].mean(), color='r',
linewidth=3)
plt.axvline(df[df.nombre.isin(top_qb)]['nfl_primero_cmp'].mean(), color='g',
linewidth=3)
plt.axvline(df[df.nombre.isin(top_qb)]['nfl_primero_cmp'].mean()-h, color='g',
linestyle='dashed', linewidth=2)
plt.axvline(df[df.nombre.isin(top_qb)]['nfl_primero_cmp'].mean()+h,  color='g',
linestyle='dashed', linewidth=2)
plt.title('Pases completados en primera temporada profesional')
plt.ylabel('Número de jugadores')
plt.xlabel('Pases completados')
plt.show()

df['pos_primera'] = ((df["('Year', 'min')"] == df['año_draft'])).astype(int)
df.groupby('año_draft')['nfl_primero_cmp'].mean().plot()
plt.axhline(df[(df['año_draft']>=1980) &
(df['año_draft']<1990)]['nfl_primero_cmp'].mean(), color='r', linestyle='dashed',
linewidth=2)
plt.axhline(df[(df['año_draft']>=1990) &
(df['año_draft']<2000)]['nfl_primero_cmp'].mean(), color='g', linestyle='dashed',
linewidth=2)
plt.axhline(df[(df['año_draft']>=2000) &
(df['año_draft']<2010)]['nfl_primero_cmp'].mean(), color='y', linestyle='dashed',
linewidth=2)
plt.axhline(df[(df['año_draft']>=2010) &
(df['año_draft']<2020)]['nfl_primero_cmp'].mean(), color='c', linestyle='dashed',
linewidth=2)
plt.show()
#plt.plot(df.año_draft, df.max_yds)

df.pct_juegos_desv.plot.hist()
plt.title('Histograma de la desviación estándar del porcentaje de partidos
colegiales ganados')
plt.ylabel('Suma de jugadores')
plt.xlabel('Desviación estándar porcentaje de partidos ganados')
plt.axvline(df.pct_juegos_desv.median(), color='r', linestyle='dashed',
linewidth=2)
plt.plot()
```

Apéndice E. Código Transformación de datos.

```
import pandas as pd
import numpy as np

df = pd.read_csv('final_data.csv')
top_qb = ['Warren Moon', 'Ben Roethlisberger', 'Russell Wilson', 'Steve Young',
'Troy Anikman', 'Brett Favre', 'Aaron Rodgers', 'Dan Marino', 'Drew Brees', 'Peyton
Manning', 'Tom Brady', 'Cam Newton', 'Matt Ryan', 'Andrew Luck', 'Philip Rivers']
df['objetivo'] = ((df['nfl_primeros_yds'] >=
df[df.nombre.isin(top_qb)]['nfl_primeros_yds'].mean()) | (df['nfl_primero_cmp'] >=
df[df.nombre.isin(top_qb)]['nfl_primero_cmp'].mean())).astype(int)

df.replace({'on: FB': 'Right', 'on: QB': 'Right', 'on: QB/TE': 'Right', 'on: RB':
'Right', 'on: WR': 'Right'}, inplace = True)

remplazo = ['TX', 'otros', 'otros', 'otros', 'VA', 'CA', 'VA', 'otros', 'OR',
'otros', 'otros', 'KS', 'otros', 'otros', 'VA', 'CA', 'GA']

df.loc[df[df.state_ab.isnull()].index, "state_high_school"] = remplazo
df.loc[df[df.state_ab.isnull()].index, "year"] =
df.loc[df[df.state_ab.isnull()].index, "min_año"]
df.loc[df[df.state_ab.isnull()].index, "state_ab"] = remplazo
df.loc[37, "gdp"] = 1163401.3
df.loc[104, "gdp"] = 293275.9
df.loc[108, "gdp"] = 426143.2
df.loc[135, "gdp"] = 455070
df.loc[188, "gdp"] = 81553.7
df.loc[215, "gdp"] = 40658.3
df.loc[272, "gdp"] = 71465.1
df.loc[278, "gdp"] = 1920061.8
df.loc[301, "gdp"] = 147759.8
df.gdp.fillna(df.gdp.median(), inplace = True)
df.loc[df[df['pct_juegos'].isnull()].index, 'pct_juegos'] = 0
df.pct_juegos_desv.fillna(df.pct_juegos_desv.median(), inplace = True)
df.desv_per.fillna(df.desv_per.median(), inplace = True)
df.desv_yds_intentos.fillna(df.desv_yds_intentos.median(), inplace = True)

for i in ["nfl_primeros_jug_vs_iniciados", "nfl_primeros_juegos",
"nfl_primero_ganado", "nfl_primero_perdido", "('Year', 'count')", "('Year',
'min')", "('tot_postemporada', 'sum')", "dif_juegos_media", "dif_cmp_media",
"diff_intentos_media", "dif_yds_media", "dif_td_media", "dif_int_media"]:
    df[i].fillna(0, inplace = True)

df.loc[df[df.pendiente_juego.isnull()].index, "pendiente_juego"] =
df.loc[df[df.pendiente_juego.isnull()].index, "dif_juegos_media"]
df.loc[df[df.pendiente_juego.isnull()].index, "pendiente_juego"] =
df.loc[df[df.pendiente_juego.isnull()].index, "dif_juegos_media"]
df.loc[df[df.pendiente_cmp.isnull()].index, "pendiente_cmp"] =
df.loc[df[df.pendiente_cmp.isnull()].index, "dif_cmp_media"]
df.loc[df[df.pendiente_intentos.isnull()].index, "pendiente_intentos"] =
df.loc[df[df.pendiente_intentos.isnull()].index, "diff_intentos_media"]
df.loc[df[df.pendiente_yds.isnull()].index, "pendiente_yds"] =
df.loc[df[df.pendiente_yds.isnull()].index, "dif_yds_media"]
```

```
df.loc[df[df.pendiente_td.isnull()].index, "pendiente_td"] =
df.loc[df[df.pendiente_td.isnull()].index, "dif_td_media"]
df.loc[df[df.pendiente_int.isnull()].index, "pendiente_int"] =
df.loc[df[df.pendiente_int.isnull()].index, "dif_int_media"]

#Elminicacion de observaciones atipicas
df = df.drop(20).drop(131).drop(201)

#Transformacion de variables. Mal y bien para optimizacion de variables.
df["juegos_totales"] = df["juegos_totales"]**4 #bien #mal
df["min_juegos"] = df["min_juegos"]**3 #mal #bien 0.476
df["max_cmp"] = df["max_cmp"]**4 #bien 0.514 #bien
df["min_intentos"] = df["min_intentos"]**(1/3) #bien #bien
df['max_per'] = np.log(df['max_per']) #mal #mal
df["desv_per"] = df["desv_per"]**(1/3) #mal #mal
df["total_td"] = df["total_td"]**4 #bien #mal
df["min_td"] = df["min_td"]**(1/2) #mal #mal
df["min_int"] = df["min_int"]**(1/2) #bien #mal
df['max_yds_intentos'] = 1 / (df['max_yds_intentos']**(1/2)) #mal 0.515 #mal
df['desv_yds_intentos'] = np.log(df['desv_yds_intentos']) #mal 0.522 #mal
df["juegos_ganados"] = df["juegos_ganados"]**4 #mal #bien
df["pct_juegos_desv"] = df["pct_juegos_desv"]**(1/2) #mal #bien 0.478
df['gdp'] = np.log(df['gdp']) #mal #bien

#En caso de correr para primer año profesional comentar la siguiente linea.
df1 = df[df['nfl_primer_año']<=2016]
df1 = df1.drop(['id', 'nombre', 'rate_q1', 'rate_q3', 'min_año', 'state_ab',
'year', 'año_draft', "('Year', 'min')"], axis = 1)

df1.replace({'Right':1, "Left":0}, inplace=True)

#Variables artificiales
dummies = pd.get_dummies(df1.state_high_school, drop_first=True)
df1 = pd.concat([df1.drop('state_high_school', axis = 1), dummies], axis = 1)


#Seleccion de variables
X = df1[['años_colegial', 'universidad_distinta', 'juegos_totales', 'min_juegos',
'max_juegos', 'dif_juegos_media', 'pendiente_juego', 'suma_cmp', 'min_cmp',
'max_cmp', 'dif_cmp_media', 'pendiente_cmp', 'suma_intentos', 'min_intentos',
'max_intentos', 'diff_intentos_media', 'pendiente_intentos', 'max_per', 'min_per',
'media_per', 'desv_per', 'total_yds', 'min_yds', 'max_yds', 'dif_yds_media',
'pendiente_yds', 'total_td', 'min_td', 'max_td', 'dif_td_media', 'pendiente_td',
'total_int', 'min_int', 'max_int', 'dif_int_media', 'pendiente_int', 'rate_q2',
'max_yds_intentos', 'min_yds_intentos', 'media_yds_intentos', 'desv_yds_intentos',
'bowls_ganados', 'bowls_jugados', 'juegos_ganados', 'juegos_perdidos',
'pct_juegos', 'pct_juegos_desv', 'throws', 'height', 'weight', 'gdp', 'AZ', 'CA',
'CO', 'CT', 'DC', 'FL', 'GA', 'HI', 'IA', 'ID', 'IL', 'IN', 'KS', 'KY', 'LA', 'MA',
'MD', 'MI', 'MN', 'MO', 'MS', 'MT', 'NC', 'ND', 'NJ', 'NM', 'OH', 'OK', 'OR', 'PA',
'SC', 'TN', 'TX', 'UT', 'VA', 'WA', 'WI', 'otros']]
#añadir AR
#comentar la y necesaria para primer o primeras cuatro temporadas profesioanles
y = df1['nfl_primero_cmp']
y = df1['nfl_tot_cmp']
```

```
y = df1['objetivo']
scaler = MinMaxScaler()
X_stand = scaler.fit_transform(X)

#Seleccion de variables con conocimiento experto
X = df1[['años_colegial', 'universidad_distinta', 'juegos_totales', 'min_juegos',
'max_juegos', 'pendiente_juego', 'suma_cmp', 'min_cmp', 'max_cmp', 'dif_cmp_media',
'pendiente_cmp', 'max_per', 'min_per', 'media_per', 'desv_per', 'total_yds',
'min_yds', 'max_yds', 'pendiente_yds', 'total_td', 'min_td', 'max_td',
'pendiente_td', 'rate_q2', 'max_yds_intentos', 'min_yds_intentos',
'media_yds_intentos', 'desv_yds_intentos', 'bowls_ganados', 'bowls_jugados',
'juegos_ganados', 'juegos_perdidos', 'pct_juegos', 'pct_juegos_desv', 'gdp', 'CA',
'FL', 'IN', 'LA', 'MO', 'PA', 'TX']]
añadir AR
y = df1['nfl_primero_cmp']
y = df1['nfl_tot_cmp']

#Seleccion de variables por valor t
X = df1[['diff_intentos_media', 'max_td', 'dif_td_media', 'pct_juegos', 'AR', 'CT',
'IA', 'MO', 'ND', 'NJ', 'OR', 'PA', 'SC']]
y = df1['nfl_primero_cmp']
X = df1[['años_colegial', 'universidad_distinta', 'CA', 'FL', 'IN', 'LA', 'MO',
'PA', 'TX']]
y = df1['nfl_tot_cmp']
```

Apéndice F. Código entrenamiento de modelos y prueba.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import anderson, shapiro, normaltest, kstest, skew, zscore, boxcox
from sklearn.preprocessing import StandardScaler, MinMaxScaler, PolynomialFeatures
from sklearn.linear_model import LinearRegression, Ridge, LogisticRegression, Lasso
from sklearn.metrics import mean_squared_error, roc_curve, auc, roc_auc_score,
confusion_matrix, precision_score, recall_score
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.tree import DecisionTreeRegressor, DecisionTreeClassifier
from sklearn.neural_network import MLPRegressor, MLPClassifier
from statsmodels.tools.eval_measures import rmse
from statsmodels.stats.diagnostic import het_breuschpagan, het_white
from statsmodels.stats.stattools import durbin_watson
import statsmodels.tsa.api as tsm
import statsmodels.api as sm
import math
from sklearn.cluster import KMeans
from sklearn.metrics import silhouette_score
pd.set_option('display.max_columns', None)

scaler = MinMaxScaler()
X_kmeans = df1[df1.objetivo == 1][['años_colegial', 'universidad_distinta',
'juegos_totales', 'min_juegos', 'max_juegos', 'dif_juegos_media',
'pendiente_juego', 'suma_cmp', 'min_cmp', 'max_cmp', 'dif_cmp_media',
'pendiente_cmp', 'suma_intentos', 'min_intentos', 'max_intentos',
'diff_intentos_media', 'pendiente_intentos', 'max_per', 'min_per', 'media_per',
'desv_per', 'total_yds', 'min_yds', 'max_yds', 'dif_yds_media', 'pendiente_yds',
'total_td', 'min_td', 'max_td', 'dif_td_media', 'pendiente_td', 'total_int',
'min_int', 'max_int', 'dif_int_media', 'pendiente_int', 'rate_q2',
'max_yds_intentos', 'min_yds_intentos', 'media_yds_intentos', 'desv_yds_intentos',
'bowls_ganados', 'bowls_jugados', 'juegos_ganados', 'juegos_perdidos',
'pct_juegos', 'pct_juegos_desv', 'throws', 'height', 'weight', 'gdp', 'AZ', 'CA',
'CO', 'CT', 'DC', 'FL', 'GA', 'HI', 'IA', 'ID', 'IL', 'IN', 'KS', 'KY', 'LA', 'MA',
'MD', 'MI', 'MN', 'MO', 'MS', 'MT', 'NC', 'ND', 'NJ', 'NM', 'OH', 'OK', 'OR', 'PA',
'SC', 'TN', 'TX', 'UT', 'VA', 'WA', 'WI', 'otros']]
#añadir 'AR'
X_kmean = scaler.fit_transform(X_kmeans.copy())

distances = []
for i in range(2, 10):
    kmean = KMeans(n_clusters = i, random_state=12).fit(X_kmean)
    distance = kmean.inertia_
   distances.append(distance)

plt.plot([i for i in range(2, 10)], distances)
plt.title('Método del codo para clusters óptimos')
plt.xlabel('Número de clusters')
plt.ylabel('SSD')
plt.show()

sil_score = []
```

```
for i in range(2, 10):
    kmeans = KMeans(n_clusters=i, random_state=12).fit(X_kmean)
    labels = kmeans.labels_    sil_score.append(silhouette_score(X_kmean, labels,
metric='euclidean'))


plt.plot([i for i in range(2, 10)], sil_score)
plt.title('Método de la silueta para clusters óptimos')
plt.xlabel('Número de clusters')
plt.ylabel('Score de silueta')
plt.show()

#Para primer año
kmean_opt = KMeans(n_clusters = 4, random_state = 12).fit_predict(X_kmean)

#Para primeros cuatro años
kmean_opt = KMeans(n_clusters = 3, random_state = 12).fit_predict(X_kmean)

np.unique(kmean_opt, return_counts = True)

#Para prediccion primer año
X_kmeans['kmean'] = kmean_opt
k_0 = []
k_1 = []
k_2 = []
k_3 = []
for i in range(48):
    k_0.append(X_kmeans[X_kmeans['kmean'] == 0].sample(7, random_state = i).mean())
for i in range(44):
    k_1.append(X_kmeans[X_kmeans['kmean'] == 1].sample(6, random_state = i).mean())
for i in range(55):
    k_2.append(X_kmeans[X_kmeans['kmean'] == 2].sample(8, random_state = i).mean())
for i in range(44):
    k_3.append(X_kmeans[X_kmeans['kmean'] == 3].sample(6, random_state = i).mean())
bal_df = pd.concat([pd.DataFrame(k_0), pd.DataFrame(k_1), pd.DataFrame(k_2),
pd.DataFrame(k_3)], ignore_index = True)

#Para prediccion cuatro años
X_kmeans['kmean'] = kmean_opt
k_0 = []
k_1 = []
k_2 = []
for i in range(62):
    k_0.append(X_kmeans[X_kmeans['kmean'] == 0].sample(5, random_state = i).mean())
for i in range(31):
    k_1.append(X_kmeans[X_kmeans['kmean'] == 1].sample(3, random_state = i).mean())
for i in range(114):
    k_2.append(X_kmeans[X_kmeans['kmean'] == 2].sample(8, random_state = i).mean())
bal_df = pd.concat([pd.DataFrame(k_0), pd.DataFrame(k_1), pd.DataFrame(k_2)],
ignore_index = True)

bal_df.drop('kmean', axis = 1, inplace = True)
bal_df['objetivo'] = 1
X = X.append(bal_df.drop('objetivo', axis = 1), ignore_index=True)
```

```
y = y.append(bal_df.objetivo, ignore_index = True)

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2,
random_state = 12)

X_train = np.array(X_train)
X_train = sm.add_constant(X_train)
y_train = np.array(y_train)
model = sm.OLS(y_train, X_train)
results = model.fit()
print(results.summary())
X_test = np.array(X_test)
X_test = sm.add_constant(X_test)
results.predict(X_test)
pred_val = results.fittedvalues.copy()
residual = y_train - pred_val
rmse(y_test, results.predict(X_test))

rmse_prueba = []
rmse_entrenamiento = []
for i in [0.65, 0.7, 0.75, 0.8, 0.85, 0.90, 0.95, 1]:
    lasso = Lasso(random_state = 12, max_iter = 50, alpha = i)
    lassogrid = GridSearchCV(lasso, param_grid = {"selection": ["cyclic",
"random"], "fit_intercept":[True, False]},
                          scoring = "mean_squared_error", n_jobs = -1)
    lassogrid.fit(X_train, y_train)
    rmse_entrenamiento.append(math.sqrt(mean_squared_error(y_train,
lassogrid.best_estimator_.predict(X_train))))
    rmse_prueba.append(math.sqrt(mean_squared_error(y_test,
lassogrid.best_estimator_.predict(X_test))))

plt.plot([0.65, 0.7, 0.75, 0.8, 0.85, 0.90, 0.95, 1], rmse_entrenamiento)
plt.plot([0.65, 0.7, 0.75, 0.8, 0.85, 0.90, 0.95, 1], rmse_prueba)
plt.xlabel('Coeficientes de regularización')
plt.ylabel('RMSE')
plt.title('Disyuntiva sesgo-varianza')
plt.show()

rmse_entrenamiento = []
rmse_prueba = []
for i in range(1, 26):
    dt = DecisionTreeClassifier(random_state = 12, max_depth = i)
    dtgrid = GridSearchCV(dt, param_grid = {"criterion":['gini', 'entropy'],
                                            "min_samples_split": [2, 5],
"min_samples_leaf": [1, 2, 5],
                                            "max_features":[None, 'auto']},
                          scoring = "f1", n_jobs = -1)
    dtgrid.fit(X_train, y_train)
    rmse_entrenamiento.append(dtgrid.best_estimator_.score(X_train, y_train))
    rmse_prueba.append(dtgrid.best_estimator_.score(X_test, y_test))
    #pred1 = dtgrid.best_estimator_.predict(X_train)
    #pred2 = dtgrid.best_estimator_.predict(X_test)
    #rmse_entrenamiento.append(recall_score(y_train, pred1))
    #rmse_prueba.append(recall_score(y_test, pred2))
```

```
plt.plot(list(range(1, 26)), rmse_entrenamiento, label='entrenamiento')
plt.plot(list(range(1, 26)), rmse_prueba, label = 'prueba')
plt.xlabel('Profundidad')
plt.ylabel('Exactitud')
plt.legend()
plt.title('Exactitud sesgo-varianza: Profundidad')
plt.show()

for i in ['universidad_distinta', 'juegos_totales', 'min_juegos', 'max_juegos',
'dif_juegos_media', 'pendiente_juego', 'suma_cmp', 'min_cmp', 'max_cmp',
'dif_cmp_media', 'pendiente_cmp', 'suma_intentos', 'min_intentos', 'max_intentos',
'diff_intentos_media', 'pendiente_intentos', 'max_per', 'min_per', 'media_per',
'desv_per', 'total_yds', 'min_yds', 'max_yds', 'dif_yds_media', 'pendiente_yds',
'total_td', 'min_td', 'max_td', 'dif_td_media', 'pendiente_td', 'total_int',
'min_int', 'max_int', 'dif_int_media', 'pendiente_int', 'rate_q2',
'max_yds_intentos', 'min_yds_intentos', 'media_yds_intentos', 'desv_yds_intentos',
'bowls_ganados', 'bowls_jugados', 'juegos_ganados', 'juegos_perdidos',
'pct_juegos', 'pct_juegos_desv', 'height', 'weight', 'gdp']:
    print(i)
    print(anderson(df1[i], dist='norm'))
    print(anderson(df1[i], dist='norm')[0] < anderson(df1[i], dist='norm')[1][3])
    print(shapiro(df1[i])[1])

normal_variables = ['universidad_distinta', 'juegos_totales', 'min_juegos',
'max_juegos', 'dif_juegos_media', 'pendiente_juego', 'suma_cmp', 'min_cmp',
'max_cmp', 'dif_cmp_media', 'min_intentos', 'max_intentos', 'diff_intentos_media',
'max_per', 'min_per', 'media_per', 'desv_per', 'min_yds', 'dif_yds_media',
'total_td', 'min_td', 'dif_td_media', 'total_int', 'min_int', 'max_int',
'dif_int_media', 'rate_q2', 'max_yds_intentos', 'min_yds_intentos',
'desv_yds_intentos', 'bowls_ganados', 'bowls_jugados', 'juegos_ganados',
'juegos_perdidos', 'pct_juegos', 'pct_juegos_desv', 'throws', 'height', 'weight',
'gdp', "pendiente_cmp", "suma_intentos", "pendiente_intentos", "total_yds",
"max_yds", "pendiente_yds", "max_td", "pendiente_td", "pendiente_int",
"media_yds_intentos"]
def normal(e):
    return(anderson(e, dist='norm')[0] < anderson(e, dist='norm')[1][3])
dic = {}
for i in normal_variables:
    values = []
    try:
        values.append(['no-mod', shapiro(X_train[i])[1], normal(X_train[i]),
anderson(X_train[i], dist='norm')[0], anderson(X_train[i], dist='norm')[1][3]])
    except:
        pass
    try:
        values.append(['cuad', shapiro(X_train[i]^2)[1], normal(X_train[i]^2),
anderson(X_train[i]^2, dist='norm')[0], anderson(X_train[i]^2, dist='norm')[1][3]])
    except:
        pass
    try:
        values.append(['tres', shapiro(X_train[i]^3)[1], normal(X_train[i]^3),
anderson(X_train[i]^3, dist='norm')[0], anderson(X_train[i], dist='norm')[1][3]])
    except:
```

```
        pass
    try:
        values.append(['cuatro', shapiro(X_train[i]^4)[1], normal(X_train[i]^4),
anderson(X_train[i]^4, dist='norm')[0], anderson(X_train[i]^4, dist='norm')[1][3]])
    except:
        pass
    try:
        values.append(['cuatro', shapiro(X_train[i]^5)[1], normal(X_train[i]^5),
anderson(X_train[i]^5, dist='norm')[0], anderson(X_train[i]^5, dist='norm')[1][3]])
    except:
        pass
    try:
        values.append(['raiz', shapiro(X_train[i]**(1/2))[1],
normal(X_train[i]**(1/2)), anderson(X_train[i]**(1/2), dist='norm')[0],
anderson(X_train[i]**(1/2), dist='norm')[1][3]])
    except:
        pass
    try:
        values.append(['raiz_cub', shapiro(X_train[i]**(1/3))[1],
normal(X_train[i]**(1/3)), anderson(X_train[i]**(1/3), dist='norm')[0],
anderson(X_train[i]**(1/3), dist='norm')[1][3]])
    except:
        pass
    try:
        values.append(['log', shapiro(np.log(X_train[i]))[1],
normal(np.log(X_train[i])), anderson(np.log(X_train[i]), dist='norm')[0],
anderson(np.log(X_train[i]), dist='norm')[1][3]])
    except:
        pass
    try:
        values.append(['1/x', shapiro(1/X_train[i])[1], normal(1/X_train[i]),
anderson(1/X_train[i], dist='norm')[0], anderson(1/X_train[i], dist='norm')[1][3]])
    except:
        pass
    try:
        values.append(['1/cuad', shapiro(1/X_train[i]^2)[1],
normal(1/X_train[i]^2), anderson(1/X_train[i]^2, dist='norm')[0],
anderson(1/X_train[i]^2, dist='norm')[1][3]])
    except:
        pass
    try:
        values.append(['1/raiz', shapiro(1/X_train[i]**(1/2))[1],
normal(1/X_train[i]**(1/2)), anderson(1/X_train[i]**(1/2), dist='norm')[0],
anderson(1/X_train[i]**(1/2), dist='norm')[1][3]])
    except:
        pass
    dic[i] = values

e = 1
for i in ['años_colegial', 'universidad_distinta', 'juegos_totales', 'min_juegos',
'max_juegos', 'dif_juegos_media', 'pendiente_juego', 'suma_cmp', 'min_cmp',
'max_cmp', 'dif_cmp_media', 'pendiente_cmp', 'suma_intentos', 'min_intentos',
'max_intentos', 'diff_intentos_media', 'pendiente_intentos', 'max_per', 'min_per',
'media_per', 'desv_per', 'total_yds', 'min_yds', 'max_yds', 'dif_yds_media',
```

```
'pendiente_yds', 'total_td', 'min_td', 'max_td', 'dif_td_media', 'pendiente_td',
'total_int', 'min_int', 'max_int', 'dif_int_media', 'pendiente_int', 'rate_q2',
'max_yds_intentos', 'min_yds_intentos', 'media_yds_intentos', 'desv_yds_intentos',
'bowls_ganados', 'bowls_jugados', 'juegos_ganados', 'juegos_perdidos',
'pct_juegos', 'pct_juegos_desv', 'throws', 'height', 'weight', 'gdp','AR', 'AZ',
'CA', 'CO', 'CT', 'DC', 'FL', 'GA', 'HI', 'IA', 'ID', 'IL', 'IN', 'KS', 'KY', 'LA',
'MA', 'MD', 'MI', 'MN', 'MO', 'MS', 'MT', 'NC', 'ND', 'NJ', 'NM', 'OH', 'OK', 'OR',
'PA', 'SC', 'TN', 'TX', 'UT', 'VA', 'WA', 'WI', 'otros']:
    #Agregar AR
    plt.scatter(X_train[:,e], residuales)
    plt.axhline(0, color ='r', linestyle='dashed')
    plt.title("Residuales vs " + i)
    plt.ylabel('Error estándar')
    plt.xlabel(i)
    plt.show()
    e += 1

pred_val = results.fittedvalues.copy()
plt.scatter(pred_val, residual)
plt.axhline(0, color ='r', linestyle='dashed')
plt.title("Valores ajustados vs. residuales")
plt.ylabel('Error')
plt.xlabel('Valores ajustados')

#durbin_watson(residuales)
acf = tsm.graphics.plot_acf(residuales, alpha = 0.05)
acf.show()

np.unique(np.where(np.abs(zscore(df1[['años_colegial', 'universidad_distinta',
'juegos_totales', 'min_juegos', 'max_juegos', 'dif_juegos_media',
'pendiente_juego', 'suma_cmp', 'min_cmp', 'max_cmp', 'dif_cmp_media',
'pendiente_cmp', 'suma_intentos', 'min_intentos', 'max_intentos',
'diff_intentos_media', 'pendiente_intentos', 'max_per', 'min_per', 'media_per',
'desv_per', 'total_yds', 'min_yds', 'max_yds', 'dif_yds_media', 'pendiente_yds',
'total_td', 'min_td', 'max_td', 'dif_td_media', 'pendiente_td', 'total_int',
'min_int', 'max_int', 'dif_int_media', 'pendiente_int', 'rate_q2',
'max_yds_intentos', 'min_yds_intentos', 'media_yds_intentos', 'desv_yds_intentos',
'bowls_ganados', 'bowls_jugados', 'juegos_ganados', 'juegos_perdidos',
'pct_juegos', 'pct_juegos_desv', 'throws', 'height', 'weight', 'gdp']]))>3)[0],
return_counts=True)

np.where(np.abs(zscore(df1[['años_colegial', 'universidad_distinta',
'juegos_totales', 'min_juegos', 'max_juegos', 'dif_juegos_media',
'pendiente_juego', 'suma_cmp', 'min_cmp', 'max_cmp', 'dif_cmp_media',
'pendiente_cmp', 'suma_intentos', 'min_intentos',  'max_intentos',
'diff_intentos_media', 'pendiente_intentos', 'max_per', 'min_per', 'media_per',
'desv_per', 'total_yds', 'min_yds', 'max_yds', 'dif_yds_media', 'pendiente_yds',
'total_td', 'min_td', 'max_td', 'dif_td_media', 'pendiente_td', 'total_int',
'min_int', 'max_int', 'dif_int_media', 'pendiente_int', 'rate_q2',
'max_yds_intentos', 'min_yds_intentos', 'media_yds_intentos', 'desv_yds_intentos',
'bowls_ganados', 'bowls_jugados', 'juegos_ganados', 'juegos_perdidos',
'pct_juegos', 'pct_juegos_desv', 'throws', 'height', 'weight', 'gdp']]))>3)

dic = {}
```

```
for a in range(301):
    dic[a] = []
for i in ['años_colegial', 'universidad_distinta', 'juegos_totales', 'min_juegos',
'max_juegos', 'dif_juegos_media', 'pendiente_juego', 'suma_cmp', 'min_cmp',
'max_cmp', 'dif_cmp_media', 'pendiente_cmp', 'suma_intentos', 'min_intentos',
'max_intentos', 'diff_intentos_media', 'pendiente_intentos', 'max_per', 'min_per',
'media_per', 'desv_per', 'total_yds', 'min_yds', 'max_yds', 'dif_yds_media',
'pendiente_yds', 'total_td', 'min_td', 'max_td', 'dif_td_media', 'pendiente_td',
'total_int', 'min_int', 'max_int', 'dif_int_media', 'pendiente_int', 'rate_q2',
'max_yds_intentos', 'min_yds_intentos', 'media_yds_intentos', 'desv_yds_intentos',
'bowls_ganados', 'bowls_jugados', 'juegos_ganados', 'juegos_perdidos',
'pct_juegos', 'pct_juegos_desv', 'throws', 'height', 'weight', 'gdp']:
    for e in range(301):
        q1 = df1[i].quantile(0.25)
        q3 = df1[i].quantile(0.75)
        iqr = q3 -q1
        dic[e].append(((df1[i] < q1 - 1.5 * iqr) | (df1[i] > q1 + 1.5 * iqr))[e])

plt.hist(residuales)
plt.title("Histograma de los errores estandarizados")
plt.xlabel('Error')
plt.ylabel('Cuenta')
plt.show()

fig, ax = plt.subplots(figsize=(6,4))
_, (__, ___, r) = sp.stats.probplot(residuales, plot=ax, fit=True)
plt.title('Gráfico Q-Q')
plt.ylabel('Valores ordenados')
plt.xlabel('Cuantiles teóricos')
r**2

pred_val = results.fittedvalues.copy()
residual = y_train - pred_val
het_breuschpagan(residual, X_train[:,1:])
anderson(residuales, dist='norm')
shapiro(residuales)


log = LogisticRegression(penalty = 'none', max_iter=10000, solver = 'saga', n_jobs
= -1, random_state=12).fit(X_train, y_train)
log.score(X_train, y_train)
log.score(X_test, y_test)
pred = log.predict(X_train)
pred = log.predict(X_test)
precision_score(y_train, pred)
precision_score(y_test, pred)
recall_score(y_train, pred)
recall_score(y_test, pred)
fpr, tpr, thresholds = roc_curve(y_train, pred)
auc(fpr, tpr)
confusion_matrix(y_train, pred)
log.score(X_test, y_test)

#Para clasificacion
```

```
lasso = LogisticRegression(random_state = 12, penalty='l1', max_iter = 500)
lassogrid = GridSearchCV(lasso, param_grid = {"C":[0.4, 0.5, 0.6, 0.7, 0.8, 0.9]},
                         scoring = "f1", n_jobs = -1)
lassogrid.fit(X_train, y_train)

diccionario = dict(zip(X_train.columns, lassogrid.best_estimator_.coef_[0]))
for i in diccionario:
    if diccionario[i] != 0:
        print(i)

ridge = LogisticRegression(random_state = 12, penalty='l2')
ridgegrid = GridSearchCV(ridge, param_grid = {"C":[0.4, 0.5, 0.6, 0.7, 0.8, 0.9]},
                         scoring = "f1", n_jobs = -1)
ridgegrid.fit(X_train, y_train)

#Para clasificacion
dt = DecisionTreeClassifier(random_state = 12)
dtgrid = GridSearchCV(dt, param_grid = {"max_depth":[2, 4, 6, 8, 10, 15, 20],
"criterion":['gini', 'entropy'], "min_samples_split": [2, 5], "min_samples_leaf":
[1, 2, 5], "max_features":[None, 'auto']}, scoring = "f1", n_jobs = -1)
dtgrid.fit(X_train, y_train)

#Para clasificacion
dt = DecisionTreeClassifier(random_state = 12)
dtgrid = GridSearchCV(dt, param_grid = {"max_depth":[2, 4, 6, 8, 10, 15, 20],
"criterion":['gini', 'entropy'], "min_samples_split": [2, 5], "min_samples_leaf":
[1, 2, 5], "max_features":[None, 'auto']}, scoring = "f1", n_jobs = -1)
dtgrid.fit(X_train[['juegos_totales', 'min_juegos', 'max_juegos',
'dif_juegos_media', 'suma_cmp','min_cmp','max_cmp', 'pendiente_cmp',
'suma_intentos', 'min_intentos', 'max_intentos', 'diff_intentos_media',
'pendiente_intentos', 'max_per', 'total_yds', 'min_yds', 'max_yds',
'dif_yds_media', 'pendiente_yds', 'total_td', 'min_td', 'max_td', 'dif_td_media',
'pendiente_td', 'total_int', 'min_int', 'max_int', 'dif_int_media',
'pendiente_int', 'rate_q2', 'min_yds_intentos', 'media_yds_intentos',
'desv_yds_intentos', 'bowls_ganados', 'bowls_jugados', 'juegos_ganados',
'juegos_perdidos', 'throws', 'height', 'weight', 'gdp', 'FL', 'LA', 'MS', 'PA']],
y_train)

#Para clasificacion
scaler = MinMaxScaler()
mlp = MLPClassifier(random_state = 12, max_iter=500)
mlpgrid = GridSearchCV(mlp, param_grid = {"hidden_layer_sizes":[(60,), (70,),
(80,), (90,), (100,)], "activation": ['tanh', 'relu', 'logistic'], "alpha": [0.01,
0.001, 0.0001]}, scoring = "f1", n_jobs = -1)
mlpgrid.fit(scaler.fit_transform(X_train), y_train)

#Para clasificacion
mlp = MLPClassifier(random_state = 12, max_iter = 500)
mlpgrid = GridSearchCV(mlp, param_grid = {"hidden_layer_sizes":[(60,), (70,),
(80,), (90,), (100,)], "activation": ['tanh', 'relu', 'logistic'], "alpha": [0.01,
0.001, 0.0001]}, scoring = "f1", n_jobs = -1)
mlpgrid.fit(scaler.fit_transform(X_train[['juegos_totales', 'min_juegos',
'max_juegos','dif_juegos_media','suma_cmp','min_cmp','max_cmp', 'pendiente_cmp',
'suma_intentos', 'min_intentos', 'max_intentos', 'diff_intentos_media',
```

```
'pendiente_intentos', 'max_per', 'total_yds', 'min_yds', 'max_yds',
'dif_yds_media', 'pendiente_yds', 'total_td', 'min_td', 'max_td', 'dif_td_media',
'pendiente_td', 'total_int', 'min_int', 'max_int', 'dif_int_media',
'pendiente_int', 'rate_q2', 'min_yds_intentos', 'media_yds_intentos',
'desv_yds_intentos', 'bowls_ganados', 'bowls_jugados', 'juegos_ganados',
'juegos_perdidos', 'throws', 'height', 'weight', 'gdp', 'FL', 'LA', 'MS', 'PA']]),
y_train)

plt.hist(pred, bins = 5)
plt.title('Histograma de las probabilidades estimadas')
plt.xlabel('Probabilidad')
plt.ylabel('Número de observaciones')
plt.axvline(np.median(pred), color = 'r', linestyle='--')
plt.show()

np.unique(y_test - dtgrid.best_estimator_.predict(X_test), return_counts = True)
(pred > 0.92).astype(int).sum()
np.unique(y_test - (pred > 0.92).astype(int), return_counts = True)

fpr, tpr, thresholds = roc_curve(y_test, pred)
roc_auc = roc_auc_score(y_test, pred)

# Graficar curva ROC.
plt.plot(fpr, tpr, label='Curva ROC (área = %0.3f)' % roc_auc, marker = '.')
plt.plot([0, 1], [0, 1], 'k--')  # random predictions curve
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.0])
plt.xlabel('Razón Falsos Positivos (1 - Especificidad)')
plt.ylabel('Razón verdaderos positivos (Sensitividad)')
plt.title('Característica Operativa del Receptor')
plt.legend(loc="lower right")

plt.figure(figsize=(17,10))
plt.barh([i[0] for i in list(zip(X_train.columns,
dtgrid.best_estimator_.feature_importances_)) if i[1]>0], [i[1] for i in
list(zip(X_train.columns, dtgrid.best_estimator_.feature_importances_)) if i[1]>0])
plt.title('Importancia de atributos primer año profesional: Árbol de decisión')
plt.xlabel('Probabilidad de división en nodo')
plt.xticks(fontsize=14)
plt.yticks(fontsize=12)
plt.show()
```

Apéndice G. Resultados punto de referencia primer año profesional.

OLS Regression Results

Dep. Variable:	y	R-squared:	0.503
Model:	OLS	Adj. R-squared:	0.217
Method:	Least Squares	F-statistic:	1.759
Date:	Sun, 07 Jun 2020	Prob (F-statistic):	0.00114
Time:	13:51:38	Log-Likelihood:	-1357.7
No. Observations:	242	AIC:	2893.
Df Residuals:	153	BIC:	3204.
Df Model:	88		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-569.9197	367.739	-1.550	0.123	-1296.421	156.582
x1	-64.1445	38.739	-1.656	0.100	-140.677	12.388
x2	-1.0393	26.896	-0.039	0.969	-54.175	52.096
x3	5.9662	3.471	1.719	0.088	-0.892	12.824
x4	-12.4505	6.172	-2.017	0.045	-24.645	-0.256
x5	-5.4350	10.090	-0.539	0.591	-25.368	14.498
x6	0.4811	13.572	0.035	0.972	-26.332	27.294
x7	-1.8043	6.783	-0.266	0.791	-15.205	11.597
x8	0.6063	0.455	1.332	0.185	-0.293	1.506
x9	0.2516	1.117	0.225	0.822	-1.955	2.458
x10	-0.8918	0.950	-0.939	0.349	-2.768	0.984
x11	1.7020	2.238	0.761	0.448	-2.719	6.123
x12	-0.3348	1.254	-0.267	0.790	-2.811	2.142
x13	-0.4589	0.292	-1.570	0.118	-1.036	0.119
x14	0.5154	0.618	0.834	0.406	-0.706	1.737
x15	0.4662	0.551	0.845	0.399	-0.623	1.556
x16	-0.0189	1.421	-0.013	0.989	-2.826	2.788
x17	-0.0083	0.752	-0.011	0.991	-1.494	1.477
x18	-303.4582	546.664	-0.555	0.580	-1383.443	776.527
x19	288.0809	539.936	0.534	0.594	-778.612	1354.774
x20	-71.2593	664.286	-0.107	0.915	-1383.616	1241.097
x21	928.6868	1038.382	0.894	0.373	-1122.730	2980.104
x22	-0.0036	0.026	-0.140	0.889	-0.055	0.048
x23	-0.0082	0.075	-0.110	0.913	-0.156	0.140
x24	0.0304	0.048	0.639	0.524	-0.064	0.124
x25	-0.0864	0.114	-0.759	0.449	-0.311	0.139
x26	0.0190	0.070	0.272	0.786	-0.119	0.157
x27	-1.6336	1.670	-0.978	0.329	-4.932	1.665
x28	-4.7060	4.380	-1.074	0.284	-13.360	3.948
x29	4.4414	2.756	1.611	0.109	-1.004	9.887
x30	-7.5579	6.948	-1.088	0.278	-21.284	6.168
x31	4.4185	4.031	1.096	0.275	-3.545	12.382
x32	3.0578	2.420	1.264	0.208	-1.722	7.838
x33	-1.2846	5.761	-0.223	0.824	-12.665	10.096
x34	-3.8886	4.171	-0.932	0.353	-12.129	4.352
x35	-0.4201	9.588	-0.044	0.965	-19.361	18.521
x36	-0.2598	4.530	-0.057	0.954	-9.209	8.690
x37	0.0886	1.260	0.070	0.944	-2.401	2.579

x38	12.1902	27.210	0.448	0.655	-41.566	65.947
x39	0.1373	31.825	0.004	0.997	-62.736	63.011
x40	2.7664	46.390	0.060	0.953	-88.880	94.413
x41	-26.9934	53.109	-0.508	0.612	-131.914	77.927
x42	3.0083	9.782	0.308	0.759	-16.316	22.333
x43	-9.6597	13.149	-0.735	0.464	-35.637	16.317
x44	2.3307	3.602	0.647	0.519	-4.785	9.447
x45	-0.9332	1.135	-0.822	0.412	-3.176	1.310
x46	-49.7858	135.943	-0.366	0.715	-318.353	218.781
x47	50.3262	107.725	0.467	0.641	-162.494	263.146
x48	7.1618	31.944	0.224	0.823	-55.946	70.269
x49	2.3205	1.897	1.223	0.223	-1.427	6.068
x50	1.9643	1.461	1.344	0.181	-0.923	4.851
x51	2.061e-05	2.59e-05	0.795	0.428	-3.06e-05	7.18e-05
x52	-39.3844	51.572	-0.764	0.446	-141.270	62.501
x53	12.2987	47.900	0.257	0.798	-82.331	106.929
x54	125.5534	103.503	1.213	0.227	-78.927	330.034
x55	-38.2641	103.751	-0.369	0.713	-243.233	166.705
x56	6.8682	63.112	0.109	0.913	-117.815	131.551
x57	-12.8209	42.285	-0.303	0.762	-96.358	70.716
x58	10.6170	46.317	0.229	0.819	-80.886	102.120
x59	173.9233	107.239	1.622	0.107	-37.936	385.783
x60	38.8217	72.603	0.535	0.594	-104.611	182.255
x61	40.5063	67.845	0.597	0.551	-93.528	174.541
x62	25.5976	51.419	0.498	0.619	-75.985	127.180
x63	8.9933	55.152	0.163	0.871	-99.965	117.951
x64	9.3587	73.683	0.127	0.899	-136.209	154.926
x65	-32.0859	67.275	-0.477	0.634	-164.994	100.822
x66	114.3457	52.718	2.169	0.032	10.196	218.495
x67	-37.5001	55.373	-0.677	0.499	-146.894	71.894
x68	-48.7503	92.387	-0.528	0.598	-231.269	133.768
x69	-34.1363	53.781	-0.635	0.527	-140.385	72.112
x70	110.0583	73.059	1.506	0.134	-34.276	254.392
x71	22.4866	73.082	0.308	0.759	-121.893	166.866
x72	-20.6019	53.868	-0.382	0.703	-127.023	85.819
x73	6.5335	100.577	0.065	0.948	-192.166	205.233
x74	7.5789	72.173	0.105	0.917	-135.006	150.164
x75	265.4988	104.683	2.536	0.012	58.687	472.310
x76	-2.1271	51.420	-0.041	0.967	-103.711	99.457
x77	-69.0173	100.076	-0.690	0.491	-266.726	128.691
x78	-10.2259	44.948	-0.228	0.820	-99.024	78.572
x79	131.6641	75.292	1.749	0.082	-17.081	280.409
x80	48.4703	52.168	0.929	0.354	-54.593	151.533
x81	26.7362	40.974	0.653	0.515	-54.211	107.683
x82	37.4028	95.108	0.393	0.695	-150.491	225.297
x83	-4.9524	70.010	-0.071	0.944	-143.263	133.359
x84	-18.9537	43.410	-0.437	0.663	-104.714	66.806
x85	-2.039e-11	1.6e-11	-1.271	0.206	-5.21e-11	1.13e-11
x86	-6.9232	44.878	-0.154	0.878	-95.583	81.737
x87	25.4176	51.063	0.498	0.619	-75.463	126.298
x88	22.3004	98.954	0.225	0.822	-173.192	217.793
x89	0.3814	52.454	0.007	0.994	-103.246	104.008

Omnibus:	5.459	Durbin-Watson:	2.007

```
Prob(Omnibus):                  0.065   Jarque-Bera (JB):                5.192
Skew:                           0.350   Prob(JB):                       0.0746
Kurtosis:                       3.156   Cond. No.                     1.40e+16
==============================================================================
```

Apéndice H. Resultados punto de referencia primeros cuatro años profesionales.

OLS Regression Results

Dep. Variable:	y	R-squared:	0.574
Model:	OLS	Adj. R-squared:	0.278
Method:	Least Squares	F-statistic:	1.941
Date:	Sun, 07 Jun 2020	Prob (F-statistic):	0.000293
Time:	13:54:21	Log-Likelihood:	-1510.8
No. Observations:	218	AIC:	3202.
Df Residuals:	128	BIC:	3506.
Df Model:	89		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-1587.6546	1493.105	-1.063	0.290	-4542.018	1366.709
x1	-387.7602	174.536	-2.222	0.028	-733.110	-42.411
x2	181.4956	116.791	1.554	0.123	-49.596	412.587
x3	46.8459	14.769	3.172	0.002	17.622	76.070
x4	-56.5825	25.956	-2.180	0.031	-107.940	-5.225
x5	-123.6025	43.858	-2.818	0.006	-210.383	-36.822
x6	88.2618	56.513	1.562	0.121	-23.559	200.083
x7	-20.6654	28.471	-0.726	0.469	-77.000	35.669
x8	-1.8079	1.913	-0.945	0.346	-5.593	1.977
x9	2.0842	4.357	0.478	0.633	-6.536	10.705
x10	-0.6862	3.878	-0.177	0.860	-8.360	6.987
x11	5.1908	9.580	0.542	0.589	-13.765	24.147
x12	1.0433	4.962	0.210	0.834	-8.775	10.862
x13	0.6104	1.154	0.529	0.598	-1.673	2.893
x14	-0.7012	2.441	-0.287	0.774	-5.530	4.128
x15	0.8217	2.170	0.379	0.706	-3.473	5.116
x16	1.3685	5.780	0.237	0.813	-10.069	12.805
x17	-2.8438	2.946	-0.965	0.336	-8.673	2.986
x18	2286.1627	2331.110	0.981	0.329	-2326.336	6898.661
x19	-3403.3622	2331.836	-1.460	0.147	-8017.298	1210.573
x20	3454.9423	2840.547	1.216	0.226	-2165.566	9075.450
x21	-4576.1717	4447.786	-1.029	0.305	-1.34e+04	4224.532
x22	0.0008	0.118	0.006	0.995	-0.234	0.235
x23	0.1293	0.301	0.430	0.668	-0.466	0.725
x24	-0.0161	0.214	-0.075	0.940	-0.440	0.407
x25	-0.5797	0.513	-1.131	0.260	-1.594	0.435
x26	0.2122	0.291	0.728	0.468	-0.364	0.789
x27	-3.1194	6.902	-0.452	0.652	-16.776	10.538
x28	-4.7814	19.228	-0.249	0.804	-42.828	33.265
x29	30.9778	11.893	2.605	0.010	7.445	54.510
x30	-11.0303	30.107	-0.366	0.715	-70.603	48.542
x31	7.4930	16.451	0.455	0.650	-25.058	40.044
x32	4.0701	10.117	0.402	0.688	-15.949	24.089
x33	-34.2823	24.705	-1.388	0.168	-83.166	14.601
x34	1.8755	17.609	0.107	0.915	-32.966	36.717
x35	14.2834	37.021	0.386	0.700	-58.969	87.536

x36	4.6661	18.671	0.250	0.803	-32.277	41.609
x37	-6.2431	5.473	-1.141	0.256	-17.073	4.587
x38	-159.6655	119.016	-1.342	0.182	-395.159	75.828
x39	216.7054	138.175	1.568	0.119	-56.697	490.107
x40	-36.3823	190.702	-0.191	0.849	-413.719	340.954
x41	321.5744	240.458	1.337	0.183	-154.213	797.362
x42	-28.9054	43.830	-0.659	0.511	-115.631	57.820
x43	16.7602	56.540	0.296	0.767	-95.113	128.633
x44	-2.1854	13.904	-0.157	0.875	-29.697	25.326
x45	-4.5049	4.720	-0.954	0.342	-13.845	4.835
x46	217.0121	500.225	0.434	0.665	-772.768	1206.792
x47	-130.7112	478.628	-0.273	0.785	-1077.759	816.337
x48	87.0159	127.860	0.681	0.497	-165.977	340.009
x49	5.3191	7.998	0.665	0.507	-10.507	21.145
x50	8.7262	5.953	1.466	0.145	-3.052	20.504
x51	-0.0003	0.000	-2.169	0.032	-0.000	-2.23e-05
x52	0.7574	216.177	0.004	0.997	-426.985	428.500
x53	641.8755	195.649	3.281	0.001	254.750	1029.001
x54	304.0194	403.992	0.753	0.453	-495.348	1103.387
x55	178.4383	425.929	0.419	0.676	-664.334	1021.211
x56	164.2408	253.052	0.649	0.517	-336.466	664.948
x57	450.1002	178.741	2.518	0.013	96.430	803.770
x58	421.1956	230.698	1.826	0.070	-35.281	877.672
x59	881.7338	430.046	2.050	0.042	30.814	1732.653
x60	595.1904	290.065	2.052	0.042	21.246	1169.134
x61	648.1848	321.033	2.019	0.046	12.965	1283.404
x62	255.1078	235.199	1.085	0.280	-210.273	720.489
x63	1001.9993	255.265	3.925	0.000	496.913	1507.085
x64	344.4439	375.650	0.917	0.361	-398.843	1087.731
x65	188.3720	322.622	0.584	0.560	-449.990	826.734
x66	862.0423	191.682	4.497	0.000	482.767	1241.317
x67	185.6691	224.802	0.826	0.410	-259.140	630.478
x68	-6.4420	398.968	-0.016	0.987	-795.869	782.985
x69	258.3642	219.386	1.178	0.241	-175.728	692.457
x70	166.5121	287.107	0.580	0.563	-401.577	734.602
x71	671.0323	249.355	2.691	0.008	177.640	1164.425
x72	55.9208	221.564	0.252	0.801	-382.481	494.322
x73	70.7999	383.502	0.185	0.854	-688.023	829.623
x74	519.2026	287.720	1.805	0.073	-50.101	1088.506
x75	1382.1079	422.584	3.271	0.001	545.952	2218.264
x76	254.0076	212.073	1.198	0.233	-165.616	673.631
x77	77.0122	391.433	0.197	0.844	-697.505	851.530
x78	145.2600	192.245	0.756	0.451	-235.129	525.649
x79	565.1711	278.462	2.030	0.044	14.187	1116.155
x80	464.9578	207.744	2.238	0.027	53.901	876.015
x81	368.5468	169.324	2.177	0.031	33.510	703.583
x82	110.1297	377.264	0.292	0.771	-636.352	856.612
x83	-232.7956	272.376	-0.855	0.394	-771.738	306.147
x84	416.6307	183.613	2.269	0.025	53.321	779.940
x85	-23.6515	434.545	-0.054	0.957	-883.473	836.170
x86	319.7265	176.417	1.812	0.072	-29.345	668.798
x87	67.9258	202.367	0.336	0.738	-332.492	468.344
x88	-64.0485	381.825	-0.168	0.867	-819.555	691.458
x89	717.3466	403.804	1.776	0.078	-81.649	1516.342

Omnibus:	13.542	Durbin-Watson:	2.021
Prob(Omnibus):	0.001	Jarque-Bera (JB):	15.717
Skew:	0.499	Prob(JB):	0.000386
Kurtosis:	3.857	Cond. No.	1.80e+08

Referencias

Abela, P. (23 de Abril de 2020). *How the NFL Encourages Fair Competition*. Recuperado el 28 de abril de 2020, de Data Driven Investor: https://www.datadriveninvestor.com/2020/04/23/how-the-nfl-encourages-fair-competition/

Ali, M., & Rajamani, L. (2012). Automation of decision making process for selection of talented manpower considering risk factor: A data mining approach. *2012 International Conference on Information Retrieval and Knowledge Management, CAMP'12* (págs. 39-44). IEEE. doi:10.1109/InfRKM.2012.6205020

Bafna, P., Pillai, S., & Pramod, D. (2016). Quantifying performance appraisal parameters: A forward feature selection approach. *Indian Journal of Science and Technology, 9*(21), 1-7. doi:10.17485/ijst/2016/v9i21/95122

Bafna, P., Shirwaikar, S., & Pramod, D. (2019). Task recommender system using semantic clustering to identify the right personnel. *VINE Journal of Information and Knowledge Management Systems, 49*(2), 181-199. doi:10.1108/VJIKMS-08-2018-0068

Ball, K. S. (2001). The use of human resource information systems: A survey. *Personnel Review, 30*(6), 677-693. doi:10.1108/EUM0000000005979

Bastian, M., Hayes, M., Vaughan, W., Shah, S., Skomoroch, P., Hyungjin, K., . . . Lloyd, C. (2014). LinkedIn Skills: Large-Scale Topic Extraction and Inference. *8th ACM Conference on Recommender Systems* (págs. 1-8). New York: Association for Computing Machinery. doi:10.1145/2645710.2645729

Behrani, S., Abbasi, S., & Bhutto, A. (2017). Predicting User Mood by Classifying Music Genres from Facebook Shares and Likes. *Asian Journal of Engineering, Science & Technology, 7*(2), 1-7.

Bradley, D. (Productor), Kurland, A., & Wyatt, M. (Dirección). (2015). *No contract, No insurance: NFL players Battle for Benefits* [Película]. Recuperado el 04 de junio de 2020, de youtube.com/watch?v=os3vITs0DM

Celik, D., Karakas, A., Bal, G., Gultunca, C., Elci, A., Buluz, B., & Alevli, M. (2013). Towards an Information Extraction System Based on Ontology to Match Resumes and Jobs. *International Computer Software and Applications Conference* (págs. 333-338). IEEE.

Cerullo, M. (9 de Enero de 2020). *Robots: Now coming to a workplace near you*. Recuperado el 23 de marzo de 2020, de CBS News: https://www.cbsnews.com/news/robots-now-coming-to-a-workplace-near-you/

Chang, Y., & Guan, M. (2008). Data mining to improve human resource in construction company. *2008 International Seminar on Business and Information Management* (págs. 275-278). IEEE. doi:10.1109/ISBIM.2008.187

Chávez García, E. M., Arguello Pazmiño, A. M., Viscarra Armijos, C., Aro Sosa, G. L., & Albarrasín Reinoso, M. V. (2018). Inteligencia Artificial en la toma de decisiones gerenciales. *Dilemas Contemporáneos: Educación, Política y Valores*, 8.

Chen, L.-F., & Chien, C.-F. (2011). Manufacturing intelligence for class prediction and rule generation to support human capital decisions for high-tech industries. *Flexible Services and Manufacturing Journal, 23*(3), 263-289. doi:10.1007/s10696-010-9068-x

Chen, M., Mao, S., Zhang, Y., & Leung, V. C. (2014). Big data : related technologies, challenges and future prospects. New York: Springer. doi:10.1007/978-3-319-06245-7

Chiavenato, I. (2019). Introducción a la teoría general de la administración : una visión integral de la moderna administración de las organizaciones. En M. del Pilar Obón, & P. Mascaró Sacristán (Edits.). McGraw-Hill Interamericana. Recuperado el 2020 de marzo de 22, de https://ebookcentral.proquest.com

Chien, C.-F., & Chen, L.-F. (2008). Data mining to improve personnel selection and enhance human capital: A case study in high-technology industry. *Expert Systems with Applications, 34*(1), 280-290. doi:10.1016/j.eswa.2006.09.003

Coelho, B., Costa, F., & Gonçalves, G. (2016). ARM: Architecture for recruitment matchmaking. *Communications in Computer and Information Science* (págs. 81-99). France: Springer. doi:10.1007/978-3-319-30222-5_4

Connley, C. (30 de Noviembre de 2017). *Robots may replace 800 million workers by 2030. These skills will keep you employed*. Recuperado el 23 de marzo de 2020, de CNBC: https://www.cnbc.com/2017/11/30/robots-may-replace-up-to-800-million-workers-by-2030.html

Conway, D. (30 de septiembre de 2010). *The Data Science Venn Diagram*. Recuperado el 23 de marzo de 2020, de Drew Conway Data Consulting: http://drewconway.com/zia/2013/3/26/the-data-science-venn-diagram

Dessler, G., & Juarez, V. (2017). Administracion de recursos humanos : enfoque lationamericano. Pearson. Recuperado el 22 de marzo de 2020, de https://bookshelf.vitalsource.com/#/books/9786073241014/

Doctor, F., Hagras, H., Roberts, D., & Victor, C. (2009). A fuzzy based agent for group decision support of applicants ranking within recruitment systems. *2009 IEEE Symposium on Intelligent Agents* (págs. 8-15). IEEE. doi:10.1109/IA.2009.4927494

Faliagka, E., Tsakalidis, A., & Tzimas, G. (2012). An integrated e-recruitment system for automated personality mining and applicant ranking. *Internet Research, 22*(5), 551-568. doi:10.1108/10662241211271545

Florio, M. (27 de Diciembre de 2019). *Top ten quarterbacks of the decade*. Recuperado el 9 de Mayo de 2020, de NBC Sports: https://profootballtalk.nbcsports.com/2019/12/27/top-10-quarterbacks-of-the-decade/

García , S., Luengo, J., & Herrera, F. (2015). Data Preprocessing in Data Mining. España: Springer. doi:10.1007/978-3-319-10247-4

García, J., Molina, J. M., Berlanga, A., Patricio, M. Á., Bustamante, Á. L., & Padilla, W. R. (2018). Ciencia de Datos. Técnicas analíticas y aprendizaje estadístico. Un enfoque practico. Bogotá: Alfaomega.

Garg, P., Rani, R., & Miglani, S. (2015). Mining Professional's Data from LinkedIn. *2015 Fifth International Conference on Advances in Computing and Communications (ICACC)* (págs. 98-101). Kochi: IEEE. doi:10.1109/ICACC.2015.35

Geeter , D., & Sigalos, M. (Productores). (2019). *How much do NFL draft picks make?* [Película]. Recuperado el 4 de junio de 2020, de youtube.com/watch?v=o0N6gwd3BOI

Gibson , S., Kurland, A., Wyatt, M. (Productores), Kurland, A., & Wyatt, M. (Dirección). (2015). *Painkillers in the NFL: Marcellus Wiley & the False Choice* [Película]. Recuperado el 04 de junio de 2020, de youtube.com/watch?v=DIZvpHbmkDk

Guerrero, O. (2004). El mito del nuevo "Management" público. *Revista Venezolana de Gerencia, 9*(25), 2-9.

Gupta, P., Fernandes, S. F., & Jain, M. (2018). Automation in Recruitment: A New Frontier. *Journal of Information Technology Teaching Cases, 8*(2), 118-125. doi:10.1057/s41266-018-0042-x

Harrison, E. (2 de Julio de 2019). *Top 25 quarterbacks of all time: Patriots' Tom Brady leads list*. Recuperado el 9 de mayo de 2020, de NFL: http://www.nfl.com/news/story/0ap3000001035041/article/top-25-quarterbacks-of-all-time-patriots-tom-brady-leads-list

Hastie, T., Tibshirani, R., & Friedman, J. (2008). The Elements of Statistical Learning Data Mining, Inference, and Prediction. California: Springer.

Herbert, S. (1964). El comportamiento administrativo. Estudio de los procesos decisorios en la organización administrativa. Madrid: Aguilar.

Jain, A. (2016). Shift in HR professionals role: critical trends in HR management practices. *International Research Journal of Management, IT and Social Sciences, 3*(5), 38-47. Obtenido de https://sloap.org/journals/index.php/irjmis/article/view/365

Jing, H. (2009). Application of fuzzy data mining algorithm in performance evaluation of human resource. *2009 International Forum on Computer Science-Technology and Applications* (págs. 343-346). IEEE. doi:10.1109/IFCSTA.2009.90

Keith, G. (2014). Big data Technologies. En *Big data : Opportunities and challenges* (págs. 4-7). BCS The chatered Institute for IT. Recuperado el 23 de marzo de 2020, de https://ebookcentral.proquest.com/lib/unam/detail.action?docID=1650370

Kelleher, J. D., & Tierney, B. (2018). Data Science. Massachusetts: The MIT Press.

Khosla, R., Chu, M.-T., & Nguyen, K. (2016). Human-robot interaction modelling for recruitment and retention of employees. *Lecture Notes in Computer Science* (págs. 302-312). Springer. doi:10.1007/978-3-319-39399-5_29

Liu, J., Li, J., Wang, T., & He, R. (2019). Will Your Classmates and Colleagues Affect Your Development in the Workplace: Predicting employees' growth based on interpersonal environment. *5th IEEE International Conference on Big Data Service and Applications, BigDataService 2019, Workshop on Big Data in Water Resources, Environment, and Hydraulic Engineering and Workshop on Medical, Healthcare, Using Big Data Technologies* (págs. 71-78). IEEE Computer Society. doi:10.1109/BigDataService.2019.00016

Lops, P., de Gemmis, M., Semeraro, G., Narducci, F., & Musto, C. (2011). Leveraging the Linkedin Social Network Data for Extracting Content-Based User Profiles. *Proceedings of the Fifth ACM Conference on Recommender Systems* (págs. 293-296). Chicago: Association for Computing Machinery. doi:10.1145/2043932.2043986

Loukides, M. (2010). *What is Data Science? The future belongs to the companies and people that turn data into products.* O'Reilly Radar.

Mayo, M. (Junio de 2017). *Is Regression Analysis Really Machine Learning?* Recuperado el 18 de mayo de 2020, de KDnuggets: https://www.kdnuggets.com/2017/06/regression-analysis-really-machine-learning.html

McCracken, M., Currie, D., & Harrison, J. (2015). Understanding graduate recruitment, development and retention for the enhancement of talent management: sharpening 'the edge' of graduate talent. *The International Journal of Human Resource Management*. doi:10.1080/09585192.2015.1102159

McKiney, W. (2011). pandas: a foundational Python library for data analysis and statistics. *Python for High Performance and Scientific Computing, 14*.

Mujtaba, D. F., & Mahapatra, N. R. (2019). Ethical Considerations in AI-Based Recruitment. *2019 IEEE International Symposium on Technology in Society (ISTAS). 1360.* IEEE Society In Social Implications of Technology. doi:10.1109/ISTAS48451.2019.8937920

Nawaz, N. (2019). Artificial intelligence is transforming recruitment effectiveness in CMMI level companies. *International Journal of Advanced Trends in Computer Science and Engineering, 8*(6), 3017-3020. doi:10.30534/ijatcse/2019/56862019

Ng, E. S., & Sears, G. J. (2017). The glass ceiling in context: the influence of CEO gender, recruitment practices and firm internationalisation on the representation of women in management. *The glass ceiling in context: the influence of CEO gender, recruitment practices and firm internationalisation on the representation of women in management, 27*, 133-151. doi:10.1111/1748-8583.12135

Oswald, F., Behrend, T., Putka, D., & Sinar, E. (2020). Big Data in Industrial-Organizational Psychology and Human Resource Management: Forward Progress for Organizational Research and Practice. *Annual Review of Organizational Psychology and Organizational Behavior, 7*, 505-533. doi:10.1146/annurev-orgpsych-032117-104553

Parkin, M. (1993). Microeconomía. Delaware: Addison-Wesley Iberoamericana.

Paz, M., & Reina, A. (2004). Nuevos procedimientos en el proceso empresarial de provisión de candidatos: el reclutamiento on line. *CUADERNOS DE CC.EE. y EE*(47), 89-110. Recuperado el 23 de marzo de 2020, de http://cuadernos. uma. es/pdfs/pdf585. pdf

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., . . . Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research, 12*, 2825-2830.

Peng, R. D., & Matsui, E. (2015). The Art of Data Science. A guide for anyone who works with data. Leanpub.

Pfleiderer, A. (Productor), Michel, H., Conzelmann, H. (Escritores), & Achtnich, T. (Dirección). (2018). *JUEGO SIN LÍMITES Las mentiras del libre comercio* [Película]. Recuperado el 28 de abril de 2020, de youtube.com/watch?v=FEdeaBjOYFs

Pivovarnik, T. P., Lamb, R. P., Zuber, R. A., & Gandar, J. M. (2008). COMPETITIVE BALANCE AND FAN INTEREST IN THE NATIONAL FOOTBALL LEAGUE. *Journal of Economics and Economic Education Research, 9*(2), 75-98. Obtenido de https://www.semanticscholar.org/paper/Competitive-Balance-and-Fan-Interest-in-the-League-Pivovarnik-Lamb/e97e7d861b2dbe8489dd9c4966987f9be70aa25e

Provost, F., & Fawcett, T. (2013). Data Science for Business. California: O'Reilly.

Puntoni, S. (30 de Agosto de 2019). *Why Most Workers Would Rather Be Replaced By A Robot*. Recuperado el 23 de marzo de 2020, de Forbes: https://www.forbes.com/sites/rsmdiscovery/2019/08/30/why-most-workers-would-rather-be-replaced-by-a-robot/#4b33cfea4cf0

Rab-Kettler, K., & Bada, L. (2019). RECRUITMENT IN THE TIMES OF MACHINE LEARNING. *Management Systems in Production Engineering, 27*(2), 105-109. doi:10.1515/mspe-2019-0018

Robbins, S., & deCenzo, D. A. (2002). Fundamentos de administración : conceptos esenciales y aplicaciones. Pearson. Recuperado el 2020 de marzo de 22, de https://bookshelf.vitalsource.com/#/books/9789702603238/

Salloum, S., Dautov, R., Chen, X., Xiaogang Peng, P., & Zhexue Huang, J. (2016). Big data analytics on Apache Spark. *International Journal of Data Science and Analytics*, 145-164. doi:https://doi.org/10.1007/s41060-016-0027-9

Salvador, M., & Poyen , R. (2016). This Is How Google Hires Thier Talent. *Kalibrr*, 6-7. Obtenido de https://www.kalibrr.com/sites/default/files/featured_images/White_Paper_How_Google_Hires_Their_Talent.pdf

Sanabria R., M. (2007). De los conceptos de administración, gobierno, gerencia, gestión y management: algunos elementos de corte epistemológico y aportes para una mayor comprensión. *Universidad & Empresa, 6*(13), 158-167.

Schottey, M. (20 de Marzo de 2013). *How the NFL Became the Most Competitive League in All of Sports*. Recuperado el 28 de Abril de 2020, de bleacher Report: https://bleacherreport.com/articles/1574285-how-the-nfl-became-the-most-competitive-league-in-all-of-sports

Sergio, H. J., & Alejandro, P. (2011). Fundamentos de gestión empresarial: enfoque basado en competencias. México: McGraw-Hill Interamericana. Recuperado el 2020 de Marzo de 22, de ProQuest Ebook Central, https://ebookcentral.proquest.com/lib/bibliodgbmhe/detail.action?docID=3215791.

Shalev-Shwartz, S., & Ben-David, S. (2014). Understanding Machine Learning. From Theory to Algorithms. New York : Cambridge University Press.

Sherman, E. (4 de Noviembre de 2019). *'Surveillance Capitalism' And The Ownership Of People's Lives*. Recuperado el 23 de marzo de 2020, de Forbes: https://www.forbes.com/sites/eriksherman/2019/11/04/surveillance-capitalism-and-the-ownership-of-peoples-lives/#45b150942545

Silberschatz, A., Stonebraker, M., & Ullman, J. (1991). Database Systems: Achievements and Opportunities. *Communications of the ACM, 34*(10), 111-113. Recuperado el 2020 de marzo de 23, de https://dl.acm.org/doi/pdf/10.1145/125223.125272

Snell, S., & Bohlander, G. (2013). Administración de recursos humanos. México D.F.: Cengage Learning. Obtenido de https://bookshelf.vitalsource.com/#/books/9786074819069/

Stilz, A. (2014). Is The Free Market Fair? *Critical Rewiew: A Journal of Politics and Society, 26*(3-4), 423-438. doi:10.1080/08913811.2014.947746

Suen, H.-Y., Chen, M.-C., & Lu, S.-H. (2019). Does the use of synchrony and artificial intelligence in video interviews affect interview ratings and applicant attitudes? *Computers in Human Behavior, 98*, 93-101. doi:10.1016/j.chb.2019.04.012

Tambe, P., Cappelli, P., & Yakubovich, V. (2019). Artificial intelligence in human resources management: Challenges and A path forward. *California Management Review, 61*(4), 15-42. doi:10.1177/0008125619867910

Todolí-Signes , A. (2019). Algorithms, artificial intelligence and automated decisions concerning workers and the risks of discrimination: the necessary collective governance of data protection. *Transfer, 25*(4), 465-481. doi:10.1177/1024258919876416

Truică, C.-O., & Barnoschi, A. (2014). Innovating HR using an expert system for recruiting IT specialists - ESRIT. *Journal of Software and System Development*, 1671-1680. doi:10.5171/2015.762987

Uzair , S., Majeed, A., & Shakeel, S. (2017). Recruitment, Selection Policies and Procedure. *International Journal of Multidisciplinary and Current Research*(5), 525-529.

Varela, R., & Lerma, A. (2016). *Gestion del Talento Humano.* México: FCA Publishing.

Wampole, K. (2012). The look of the line: An empirical investigation of the impacts of facial symmetry on salary levels of offensive linemen in the NFL. *Business and economics honors papers*, 2-36. Recuperado el 4 de junio de 2020, de www.digitalcommons.ursinus.edu/bus_econ_hon/7

Wang, Q., Li, B., & Hu, J. (2009). Feature selection for human resource selection based on affinity propagation and SVM sensitivity analysis. *2009 World Congress on Nature and Biologically Inspired Computing, NABIC 2009 - Proceedings* (págs. 31-36). IEEE. doi:10.1109/NABIC.2009.5393596

Wang, Y., & Lu, J. (2016). Challenges in crawling the Deep Web. Florida: CRC Press.

Werther, W. B., Davis, K., & Guzmán Brito, M. P. (2014). Administración de recursos humanos : gestión del capital humano. En W. B. Werther, & K. Davis. México: McGraw-Hill Interamericana. Recuperado el 23 de marzo de 2020, de https://ebookcentral.proquest.com/lib/bibliodgbmhe/detail.action?docID=3217362.

Williams, K. M., Park, J. H., & Wieling, M. B. (2010). The face reveals athletic flair: Better National Football League quarterbacks are better looking. *Personality and individual differences, 48*, 112-116. doi:10.1016/j.paid.2009.09.003

Wolfson, J., Addona, V., & Schmicker, R. H. (2011). The quarterback prediction problem: Forecasting the performance of college quarterbacks selected in the NFL draft. *Journal of quantitative analysis in sports, 7*(3), 1-20. doi:10.2202/1559-0410.1302

Xie, Q. (2019). Machine learning in human resource system of intelligent manufacturing industry. *Enterprise Information Systems*, 1-20. doi:10.1080/17517575.2019.1710862

Ye, P. (2011). The decision tree classification and its application research in personnel management. *2011 International Conference on Electronics and Optoelectronics, Proceedings* (págs. VI1372-VI1375). IEEE. doi:10.1109/ICEOE.2011.6013123

Ye, Y., Zhu, H., Xu, T., Zhuang, F., Yu, R., & Xiong, H. (2019). Identifying high potential talent: A neural network based dynamic social profiling approach. *2019 IEEE International Conference on Data Mining (ICDM)* (págs. 718-726). IEEE Computer Society. doi:DOI 10.1109/ICDM.2019.00082

Yılmaz, T., Ergil, A., & İlgen, B. (2020). Deep Learning-Based Document Modeling for Personality Detection from Turkish Texts. *Advances in Intelligent Systems and Computing, 1069*, 729-736. doi:10.1007/978-3-030-32520-6_53

Zimmerling, R. (1995). *Mercado libre y justicia social.* Recuperado el 27 de abril de 2020, de cervantesvirtual.com