Universidad Nacional Autónoma de México

Facultad de Estudios Superiores Acatlán

Machine Learning: algoritmos de clasificación y sus aplicaciones en el análisis de datos

TESIS

QUE PARA OBTENER EL TÍTULO DE:

Licenciado en Matemáticas Aplicadas y Computación

PRESENTAN:

Melisa Andrea Acevedo Núñez
Karen Elizabeth Vargas Campos

DIRECTOR DE TESIS:
M.C. Javier Rosas Hernández

Santa Cruz Acatlán, Naucalpan, Estado de México, 2017

Agradecimientos

A mi mamá, Patricia Acevedo, por ser mi ejemplo a seguir, al darme tu amor incondicional y lo mejor de ti para lograr ser una persona con valores y principios.

A mi hermana, Michelle Acevedo, por creer en mi y siempre sacarme una sonrisa.

A mi abuelita, Celestina Núñez, por cuidarme, consentirme y estar siempre para mí.

A mis tíos, Victor, Roberto, Consuelo y Alejandro por brindarme su apoyo cada vez que lo necesitaba.

A mis primos, Hameyalli y Oswaldo, por los momentos que hemos compartido.

A mi asesor, M.C. Javier Rosas Hernández, por caminar a mi lado durante todo este proceso, gracias por confiar en mí al brindarme su tiempo, paciencia, conocimientos y consejos para crecer profesionalmente.

Al Dr. Antonio Ortiz Castro y al Dr. Armando Reyes García por ayudarme a ser una mejor estudiante así como una mejor persona.

A mis mejores amigas, Vanessa y Karen, por demostrarme que la verdadera amistad existe.

A mis sinodales, Mtra. Sara Camacho, Lic. Ricardo Domínguez, Lic. Fernando González y Dr. Eduardo Loza por tomarse el tiempo de revisar este trabajo.

Melisa Andrea Acevedo Núñez

Agradecimientos

A mis padres, Adrian y Gabriela . Quienes han estado en los momentos más importantes de mi vida apoyándome incondicionalmente. Gracias por su infinito amor, esfuerzo y confianza que me han brindado en todo momento. Son mi mayor ejemplo.

A mi hermano, Adrian, por acompañarme y compartir grandes momentos juntos.

A mi abuela Ángela y a mis tías Ángeles, Laura y Araceli por estar siempre conmigo.

A la memoria de mi abuelo Magdaleno, quien me apoyo hasta el último momento.

A Melisa, a la cual agradezco por ofrecerme su amistad desde el primer día que comenzamos esta etapa y sobretodo por permitirme compartir la experiencia de este trabajo de investigación.

A mi asesor el M.C Javier Rosas, por brindarnos su apoyo, tiempo, consejos e impulsarnos en el desarrollo de esta tesis.

A la Mtra. Sara Camacho, al Lic. Ricardo Domínguez, al Lic. Fernando González y al Dr. Eduardo Loza, por tomarse el tiempo para la revisión de esta tesis.

A la UNAM, particularmente a la FES Acatlán, la cual me ha brindado las herramientas necesarias para el desarrollo de mi vida profesional.

Karen Elizabeth Vargas Campos

Índice general

Índice de figuras VII

Índice de tablas X

Introducción 1

1. Machine Learning 4
1.1. Inteligencia Artificial . . . 4
1.2. Machine Learning . . . 5
1.2.1. Disciplinas en las que se apoya Machine Learning . . . 6
1.2.2. Aplicaciones de Machine Learning . . . 7
1.3. Aprendizaje . . . 8
1.4. Aprendizaje supervisado . . . 8
1.4.1. Regresión . . . 9
1.4.2. Clasificación . . . 9
1.5. Aprendizaje no supervisado . . . 10
1.5.1. Clustering . . . 11
1.6. Metodología para Machine Learning . . . 12
1.6.1. Metodología CRISP-DM . . . 13

2. Algoritmos de aprendizaje 15
2.1. Introducción . . . 15
2.2. Regresión . . . 17
2.2.1. Regresión lineal simple . . . 17
2.2.2. Regresión no lineal . . . 18
2.2.3. Algoritmo de aprendizaje . . . 19
2.2.4. Ejemplo de aplicación . . . 20
2.2.5. Ventajas y desventajas de regresión . . . 22
2.3. Árboles de decisión . . . 23
2.3.1. Construcción de un árbol de decisión . . . 23
2.3.2. Ejemplo de aplicación . . . 23
2.3.3. Algoritmos de aprendizaje para árboles de decisión . . . 29
2.3.3.1. ID3 . . . 29
2.3.3.2. Ejemplo . . . 29

2.3.3.3. C4.5 . . . 32
2.3.3.4. Ejemplo . . . 33
2.3.4. Ventajas y desventajas de los árboles de decisión. . . . 39
2.4. Naïve Bayes . . . 40
2.4.1. Algoritmo de aprendizaje . . . 40
2.4.2. Ejemplo de aplicación . . . 41
2.4.3. Ventajas y desventajas del algoritmo Naïve Bayes . . . 43
2.5. Redes Neuronales Artificiales (RNA) . . . 44
2.5.1. Las neuronas y el cerebro . . . 44
2.5.2. Estructura de una neurona biológica . . . 44
2.5.3. Historia de las Redes Neuronales Artificiales . . . 45
2.5.4. Neurona Artificial . . . 46
2.5.5. Redes neuronales artificiales (RNA) . . . 47
2.5.6. Estructura básica de RNA . . . 48
2.5.7. Aprendizaje supervisado en Redes Neuronales Artificiales. . . . 49
2.5.7.1. Perceptrón simple . . . 50
2.5.7.2. Algoritmo de aprendizaje del perceptrón. . . . 51
2.5.7.3. Perceptrón multicapa . . . 52
2.5.7.4. Algoritmo de retropropagación (Backpropagation) . . . 53
2.5.8. Ejemplo XOR . . . 57
2.5.9. Ventajas y desventajas de las Redes Neuronales Artificiales . . . 59
2.6. Máquina de soporte de vectores . . . 60
2.6.1. Máquina de soporte de vectores no linealmente separable . . . 65
2.6.2. Función Kernel . . . 65
2.6.3. Margen suave o flexible . . . 67
2.6.4. Algoritmo de aprendizaje . . . 68
2.6.5. Ejemplo de aplicación . . . 69
2.6.6. Ventajas y desventajas de máquina de soporte de vectores . . . 71
2.7. K-medias . . . 72
2.7.1. Algoritmo de aprendizaje . . . 72
2.7.2. Medidas de calidad de clustering . . . 73
2.7.3. Ejemplo de aplicación . . . 74
2.7.4. Ventajas y desventajas de K-medias . . . 78

3. Análisis de sentimientos en redes sociales 79
3.1. Escenario de aplicación . . . 79
3.1.1. Redes sociales . . . 80
3.2. Twitter . . . 83
3.3. Juegos Olímpicos . . . 84
3.4. Recolección de datos . . . 85
3.4.1. Twitter API . . . 85
3.5. Preparación y transformación de datos . . . 89
3.5.1. Transformación . . . 90
3.6. Análisis de sentimientos . . . 90
3.6.1. Datos . . . 91
3.6.2. Integración . . . 91
3.6.3. Reconocimiento de patrones . . . 91

3.6.4. Validación 92
3.6.5. Predicción 94
3.6.6. Interpretación 95
3.6.6.1. Palabras: Tweets positivos 95
3.6.6.2. Palabras: Tweets negativos 96
3.7. Recomendador 96
3.7.1. Datos 97
3.7.2. Integración 97
3.7.3. Reconocimiento de patrones 97
3.7.4. Validación 98
3.7.5. Predicción 99
3.7.5.1. Cluster por palabra 100
3.7.5.2. Cluster por tweet 103
3.7.6. Interpretación 104
3.7.6.1. Sugerencias de usuarios, cuentas oficiales y etiquetas . . . 105
3.7.6.2. Medios de comunicación 109
3.7.6.3. Etiquetas de países participantes 111

Conclusiones **114**

A. Anexo I: Código en R **117**

B. Anexo II: Código en SQL **124**

Bibliografía **129**

Índice de figuras

1.1. Disciplinas en las que se apoya Machine Learning. 6
1.2. Aprendizaje supervisado. 9
1.3. Aprendizaje no supervisado. 10
1.4. Conjunto de datos sin valores asignados 11
1.5. Conjunto de datos divididos en cuatro grupos. 11
1.6. Metodología para Machine Learning 12
1.7. Metodología CRISP-DM 13

2.1. Matriz de confusión 16
2.2. Componentes de la recta de regresión. 17
2.3. (a) Función cúbica; (b) Función logarítmica; (c) Función exponencial. . . 18
2.4. Diagrama de dispersión. Publicaciones y seguidores en Facebook 20
2.5. Recta de regresión. Publicaciones y seguidores en Facebook 21
2.6. Error asociado a la recta de regresión. 21
2.7. Prueba del modelo. 22
2.8. Atributo salario como raíz. 25
2.9. Árbol de decisión. Compañía de cable. 28
2.10. Árbol de decisión con el algoritmo C4.5. 39
2.11. Fisiología de una neurona biológica. 44
2.12. Elementos funcionales de una neurona artificial. 46
2.13. Diagrama de una RNA monocapa. 48
2.14. Diagrama de una RNA multicapa. 49
2.15. Diagrama de un perceptrón simple. 50
2.16. Separación de dos clases mediante el perceptrón. 51
2.17. (Izquierda) Clases linealmente separables. (Derecha) Clases no linealmente separables. 52
2.18. Diagrama de un perceptrón multicapa. 53
2.19. Ejemplo perceptrón multicapa 57
2.20. Ejemplo de la máquina de soporte de vectores 60
2.21. Elementos de un hiperplano 61
2.22. Ejemplo de hiperplanos posibles 61
2.23. Conjunto de datos linealmente separable 62
2.24. Maximizar el margen 63
2.25. Conjunto no linealmente separable 65
2.26. Mapeo del espacio de características. 66
2.27. Kernel polinómico 66
2.28. Kernel radial 67
2.29. Margen flexible o suave 67

2.30. Hiperplano de clasificación 70
2.31. Distancia inter cluster e intra cluster. 73
2.32. Conjunto de datos sin valores asignados. 74
2.33. Centroides de agrupamiento aleatorios. 74
2.34. Asignación de datos a los centroides más cercanos. 75
2.35. Movimiento de los centroides de acuerdo a la media de los datos de cada clúster. 76

3.1. Las redes sociales más populares. 80
3.2. Accesibilidad de Internet a la población. 81
3.3. Porcentaje de la población con acceso a Internet. 81
3.4. Promedio del tiempo invertido en Internet durante el día. 82
3.5. Porcentaje de la población con acceso a redes sociales. 82
3.6. Porcentaje Promedio del tiempo invertido en redes sociales durante el día. 83
3.7. Crear una API en Twitter API. 86
3.8. Keys generadas por Twitter API. 86
3.9. Access Token generadas por Twitter API 87
3.10. Extracción de datos en Twitter. 88
3.11. Limpieza de tweets. 89
3.12. Construcción de la matriz binaria. 90
3.13. Matriz binaria. 90
3.14. Diagrama de proceso: Análisis de sentimientos. 91
3.15. Clasificación manual de tweets. 91
3.16. Resultado del algoritmo máquina de soporte vectorial. 1° iteración. 92
3.17. Predicción del algoritmo máquina de soporte vectorial. 1° iteración. 93
3.18. Resultado del algoritmo máquina de soporte vectorial. 2° iteración. 93
3.19. Predicción del algoritmo máquina de soporte vectorial. 2° iteración. 94
3.20. Clasificación de nuevos tweets. 94
3.21. Nube de palabras de tweets positivos. 95
3.22. Nube de palabras de tweets negativos. 96
3.23. Diagrama de proceso: Recomendador 97
3.24. Gráfica con 3 clusters. 98
3.25. Gráfica con 4 clusters. 99
3.26. Predicción algoritmo K-medias. 100
3.27. Modelo relacional: Predicción algoritmo K-medias. 100
3.28. Matriz de correlación por palabras. 101
3.29. Palabras agrupadas en el cluster de Atletismo. 101
3.30. Palabras agrupadas en el cluster de Clavados. 102
3.31. Palabras agrupadas en el cluster de Nado sincronizado. 102
3.32. Palabras irrelevantes agrupadas en el cluster cuatro. 103
3.33. Tweets relacionados al Atletismo. 103
3.34. Tweets referentes a la disciplina de Clavados. 103
3.35. Tweets con contenido relativo al Nado sincronizado. 104
3.36. Porcentaje de tweets por disciplina. 104
3.37. Usuarios por cluster. 105
3.38. Recomendaciones de usuarios para los seguidores de atletismo. 105
3.39. Recomendaciones de usuarios para los seguidores de clavados. 105

3.40. Recomendaciones de usuarios para los seguidores de nado sincronizado. . 106
3.41. Cuentas oficiales para atletismo. 106
3.42. Cuentas oficiales para clavados. 107
3.43. Cuentas oficiales para nado sincronizado. 107
3.44. Etiquetas para atletismo. 108
3.45. Etiquetas para clavados. 108
3.46. Etiquetas para nado sincronizado. 109
3.47. Principales medios de comunicación para atletismo. 110
3.48. Principales medios de comunicación para clavados. 110
3.49. Principales medios de comunicación para nado sincronizado. 111
3.50. Países participantes en atletismo. 112
3.51. Países participantes en clavados. 112
3.52. Países participantes en nado sincronizado. 113

Índice de tablas

2.1. Transformaciones de las funciones logarítmica y exponencial. 19
2.2. Conjunto de entrenamiento. Publicaciones y seguidores en Facebook . . . 20
2.3. Conjunto de datos de entrenamiento. Clientes de la compañía de televisión por cable. 23
2.4. Particiones del atributo salario. 24
2.5. Particiones del atributo edad. 24
2.6. Particiones del atributo hijos. 24
2.7. Rama de salario alto 25
2.8. Atributo edad para la rama de salario alto. 25
2.9. Atributo hijos para la rama de salario alto. 26
2.10. Rama de salario medio 26
2.11. Rama de salario bajo 27
2.12. Atributo edad para la rama de salario bajo. 27
2.13. Atributo hijos para la rama de salario bajo. 27
2.14. Caso de prueba 28
2.15. Total de clientes positivos y negativos. 30
2.16. Particiones del atributo salario (ID3). 30
2.17. Particiones del atributo edad (ID3). 31
2.18. Particiones del atributo hijos (ID3). 31
2.19. Conjunto de entrenamiento con un dato desconocido. 33
2.20. Total de clientes positivos y negativos descartando la tupla con valor desconocido. 33
2.21. Partición del atributo salario (C4.5). 34
2.22. Particiones del atributo edad (C4.5). 35
2.23. Particiones del atributo hijos (C4.5). 36
2.24. Comparación de ganancia y proporción de ganancia. 37
2.25. Rama de salario medio (C45). 37
2.26. Atributo edad para la rama de salario medio (C4.5). 37
2.27. Atributo hijos para la rama de salario medio (C4.5). 38
2.28. Comparación de ganancia y proporción de ganancia 2 iteración. 38
2.29. Conjunto de datos de entrenamiento. Correos electrónicos. 41
2.30. Valores de $P(x_i|k)$ 42
2.31. Caso de prueba (test) 43
2.32. Funciones de activación 47
2.33. Conjunto de entrenamiento. Función XOR. 57
2.34. Conjunto de entrenamiento. Fotografías. 69
2.35. Registro de prueba. Fotografías. 71
2.36. Agrupamiento de los datos. 75

Introducción

En los últimos años el crecimiento exponencial de los datos ha permitido extraer información valiosa de ellos, impulsando el resurgimiento de una de las disciplinas de la Inteligencia Artificial conocida como Machine Learning (Aprendizaje Automático), que tiene el propósito de imitar el aprendizaje humano al generar conocimiento con la ayuda de ejemplos y experiencia.

Una de las mayores ventajas de Machine Learning es que permite entrenar un algoritmo de aprendizaje para que se adapte y vaya cambiando de acuerdo a las características de un histórico de datos que es suministrado ya sea con datos previamente etiquetados (Aprendizaje supervisado) o con datos desconocidos para el analista y la computadora (Aprendizaje no supervisado). Así que, el proceso de aprendizaje al que se someten las computadoras es iterativo y esta basado en el reconocimiento de patrones, logrando la construcción de modelos automáticos.

Machine Learning ha logrado la construcción de numerosas aplicaciones que han impactado de forma positiva diversos campos de la ciencia y tecnología, tales como: motores de búsqueda (Google, Yahoo, Bing), redes sociales (Twitter, Facebook), recomendadores de servicio, productos o entretenimiento (Amazon, YouTube, Netflix), filtro antispam en correo electrónico (Gmail) e inclusive casos como la detección de enfermedades, fraudes o la evaluación de riesgos en créditos bancarios.

Ante la situación planteada anteriormente, la parte fundamental de este trabajo de investigación se centra en abordar los conceptos básicos para el desarrollo de modelos de Machine Learning, analizando los algoritmos de aprendizaje para la descriptiva de datos y problemas predictivos, los cuales serán implementados para sustentar la hipótesis de esta tesis: Utilizando técnicas de Machine Learning en la clasificación de comentarios realizados en redes sociales referentes a los Juegos Olímpicos se detectaran patrones que ayuden a la toma de decisiones.

Se decidió recolectar comentarios hechos en Twitter referentes a la participación de México en los Juegos Olímpicos debido a que se han convertido en el evento con mayor

impacto durante los días en los que se llevan a cabo, logrando captar la atención de millones de personas alrededor del mundo y ante la situación del uso excedente de redes sociales por usuarios mexicanos.

Twitter es una de las redes sociales más usadas en México y en el mundo, en la que sus usuarios comparten sus ideas, sentimientos y preferencias mediante el uso de menciones (@) y etiquetas (#). Por ello, se evaluarán las opiniones en Twitter mediante un análisis de sentimientos para determinar el agrado o desagrado acerca de las disciplinas, atletas y televisoras. Asimismo, se harán sugerencias personalizadas a los usuarios con respecto a cuentas y etiquetas de medios de comunicación, atletas y usuarios cohesivos a través de un recomendador.

Por lo tanto, el objetivo principal es resolver problemas de Machine Learning a través de algoritmos de clasificación y agrupamiento para así comparar y seleccionar el algoritmo que sea más apropiado para el análisis de sentimientos en redes sociales, dependiendo de la cantidad y naturaleza de los datos.

Actualmente, es posible realizar análisis de sentimientos mediante herramientas que monitorean los comentarios que se realizan en redes sociales, por ejemplo: *Chatterscope, Twitter Sentiment, TweetFeel, Twitrrart, Twendz, SentiText, Social Mention* [35][46]. Sin embargo para las aplicaciones de esta tesis se realizara desde la recolección y preparación de datos hasta la generación de conocimiento utilizando el lenguaje de programación R y el lenguaje SQL (*Structured Query Languaje*).

Esta investigación sigue siendo vigente a pesar de que los Juegos Olímpicos se llevaron a cabo en el año 2016 ya que el análisis de sentimientos en Twitter se puede realizar sobre cualquier tema en el que los usuarios expresen sus opiniones como: entretenimiento, productos, eventos, política, personas, celebraciones, lugares, entre otros. También de que podría ser del interés de los próximos organizadores y patrocinadores de este evento en 2020.

Este trabajo va dirigido a aquellas personas que cuentan con conocimientos en matemáticas, estadística, lenguajes de programación y bases de datos, debido a que se abordan temas sobre: pronósticos, probabilidad, hiperplanos, vectores, matrices, ecuaciones lineales y no lineales, optimización lineal y no lineal, Teorema de Bayes, correlación, bondad de ajuste, estadística descriptiva, minería de datos, inteligencia artificial, algoritmos, mapeo, datos estructurados y no estructurados.

La tesis se encuentra dividida en tres capítulos:

En el capítulo I, se desarrolla el tema de Inteligencia Artificial como antecedente para abordar el tema de Machine Learning y su metodología utilizando algoritmos de aprendizaje supervisado y no supervisado.

En el capítulo II, se presentan los principales algoritmos de aprendizaje supervisado como son: regresión, árboles de decisión, Naïve Bayes, redes neuronales artificiales, máquina de soporte vectorial, y el algoritmo de aprendizaje no supervisado K-medias. Cada algoritmo incluye las ventajas y desventajas en la solución de problemas así como un ejemplo de aplicación.

En el capítulo III, se aplicarán algunas técnicas descritas previamente de Machine Learning (Capítulo I y II) utilizando los algoritmos de máquina de soporte de vectores y K-medias para el análisis de sentimientos y recomendaciones respectivamente, de acuerdo a los comentarios realizados en Twitter durante los Juegos Olímpicos del 2016.

Capítulo 1

Machine Learning

1.1. Inteligencia Artificial

La Inteligencia Artificial es la rama de las ciencias computacionales que tiene como objetivo construir sistemas artificiales tanto hardware como software capaces de emular alguna habilidad que denote inteligencia similar a la de los seres vivos para la resolución de problemas específicos. La inteligencia en los seres vivos se refiere a los procesos de percepción sensorial y de reconocimiento de patrones.

Para que un sistema de inteligencia artificial sea capaz de resolver un determinado problema requiere de una serie de instrucciones que le especifiquen las acciones que tiene que ejecutar, en otras palabras, requiere de un algoritmo.

La inteligencia artificial se compone de varias áreas de estudio que se centran en el tratamiento de los datos y la identificación de patrones, por lo que a lo largo de la historia algunas áreas han tomado mayor importancia en su investigación. A continuación, se describen algunas [7][17][24]:

- *Procesamiento de lenguaje natural*: Son sistemas capaces de reconocer, procesar y emular el lenguaje humano. El surgimiento de esta área fue gracias al test de Turing. [1]

- *Sistemas expertos*: Son sistemas que imitan el comportamiento de un experto humano realizando tareas de toma de decisiones a partir de información suministrada

[1] Test de Turing. Propuesto por Alan Turing en un artículo de 1950 publicado en la revista Mind y titulado Computing Machinery and Intelligence. Se propone un test de inteligencia para máquinas según el cual una máquina presentaría un comportamiento inteligente en la medida en que fuese capaz de mantener una conversación con un humano sin que otra persona pueda distinguir quién es el humano y quién es la computadora.

por los mismos expertos.

- *Robótica*: Es el diseño e implementación de máquinas capaces de percibir el entorno y desempeñar actividades que requieren de inteligencia humana.

- *Machine Learning*: Esta área da la capacidad a las computadoras de "aprender" imitando el aprendizaje humano.

    - *Algoritmos genéticos*: Son sistemas que buscan soluciones a problemas complejos emulando el proceso de evolución natural al seleccionar de una población a los individuos mejor adaptados para que se reproduzcan y surjan mejores individuos. Por lo tanto, su función es producir una solución óptima a partir de la combinación de las mejores soluciones de un problema específico.

En la actualidad uno de los principales intereses de la inteligencia artificial es analizar grandes cantidades de datos que se generan a diario por los diferentes medios como redes sociales, compras en línea, televisión online, conversaciones, correos electrónicos, telefonía móvil, fotografía digital, entre otros. Este análisis de datos en la inteligencia artificial ha dado paso a la disciplina conocida como Machine Learning en la cual se concentra este trabajo de investigación.

1.2. Machine Learning

Machine Learning también conocido en español como Aprendizaje Automático nace como un campo de estudio a partir de la Inteligencia Artificial. Es la ciencia que da a las computadoras la capacidad de "aprender" e ir mejorando su desempeño en una determinada tarea de manera similar como lo hace el aprendizaje humano. Su principal objetivo es convertir datos en conocimiento. Machine Learning es capaz de conservar y generar conocimiento a través de ejemplos y experiencia [19].

El proceso de Machine Learning es iterativo pues consiste en tener una base de datos que se coloca como entrada a un algoritmo, el uso de algoritmos es el medio por el cual la computadora aprende, ya que este se encarga de analizar los datos, encontrar patrones y predecir resultados. Dichos resultados son la medida de aprendizaje de la máquina, guiando la toma de decisiones.

1.2.1. Disciplinas en las que se apoya Machine Learning

Los factores que han impulsado el crecimiento de Machine Learning son la gran variedad de datos que se generan a diario, el procesamiento computacional que se ha vuelto más potente y el almacenamiento de los datos es más accesible, todos esto se vuelve posible ya que el estudio de Machine Learning se nutre de diferentes disciplinas como se muestra en la Figura 1.1

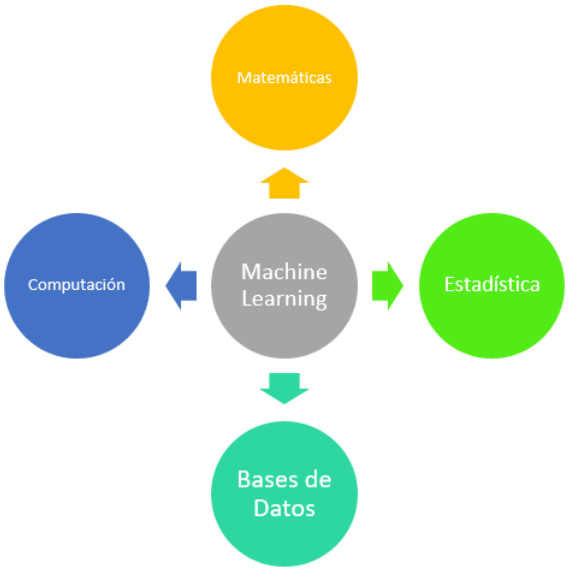


FIGURA 1.1: Disciplinas en las que se apoya Machine Learning.

Machine Learning se apoya en distintas disciplinas para su estudio como lo son [30]:

- Matemáticas: Dan fundamento a los conceptos y al desarrollo de los algoritmos de aprendizaje para así obtener un modelo matemático que detecte patrones en los datos y realice predicciones.

- Estadística: La representación y el análisis de los resultados, permite validar los modelos matemáticos.

- Bases de datos: La capacidad de almacenamiento de los datos y su administración favorecen para tener un performance alto al momento de realizar consultas SQL.

- Computación: Ofrece las herramientas de recolección de datos, los lenguajes de programación para la implementación de los algoritmos de aprendizaje y una mayor capacidad de procesamiento en el análisis de datos.

1.2.2. Aplicaciones de Machine Learning

Machine Learning se ha vuelto una herramienta potente por su capacidad de hacer cálculos con grandes volúmenes de datos para dar soluciones a problemas complejos del mundo real. Los recientes progresos de Machine Learning han impulsado cada vez más a que áreas como la tecnología, medicina, finanzas y muchas otras, se interesen en aplicar algoritmos de Machine Learning para obtener soluciones que les ayuden al momento de tomar decisiones [10].

A continuación, se enlistan algunos ejemplos de diferentes áreas donde se está aplicando Machine Learning.

Finanzas:

- Detección de fraudes en tarjetas de crédito.

Marketing:

- Campañas publicitarias.
- Análisis de fidelidad de los clientes.
- Predecir pérdidas en una compañía.
- Impulsar las ventas de productos.
- Análisis de sentimientos en redes sociales.

Medicina:

- Diagnóstico de enfermedades.

Biología:

- Descubrir secuencias en genes.

Seguridad informática:

- Detección de correo SPAM.
- Detección de malware.
- Evita el robo de identidad.

Recomendadores:

- Páginas web de compra en línea como Amazon.
- Redes sociales como Facebook, Twitter y YouTube.
- Televisión online como Netflix.

Tecnología:

- Reconocimiento de rostros.

1.3. Aprendizaje

El aprendizaje humano es un proceso donde se adquieren conocimientos mediante el estudio o la experiencia. De manera que el aprendizaje es un elemento clave para el progreso de la inteligencia artificial ya que sin aprendizaje no es posible la inteligencia.

Para que un sistema tenga la capacidad de comportarse de manera inteligente requiere poseer una gran cantidad de conocimientos que sólo los podrá adquirir si el sistema aprende a partir de ejemplos, experiencia o por sí mismo [4].

En este sentido, existen dos tipos de aprendizaje con los que un sistema obtiene conocimiento: el aprendizaje supervisado y el aprendizaje no supervisado, mismos que se describirán en las siguientes secciones.

1.4. Aprendizaje supervisado

El aprendizaje supervisado hace predicciones a partir de datos compuestos por vectores de entrada $x = (x_1, x_2, ..., x_n)$ con sus correspondientes vectores de salida $y = (y_1, y_2, ..., y_n)$, a esto se le conoce como conjunto de datos históricos $S = (x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$.

El conjunto de datos históricos se divide en dos grupos: datos de entrenamiento (70 %-80 %) y datos de prueba (30 %-20 %), donde la separación de porcentajes se basa en el principio de Pareto. [2] Los datos de entrenamiento sirven como entrada a un algoritmo de aprendizaje supervisado. Los algoritmos de aprendizaje supervisado generan un modelo de reconocimiento de patrones a partir del comportamiento o características que se han visto en los datos de entrenamiento. El modelo generado se evalúa a través del suministro de los datos de prueba, obteniendo un porcentaje de predicción. Este proceso llega a ser iterativo, pues se busca generar un modelo que satisfaga las necesidades del problema. En la Figura 1.2 se muestra la manera en la que trabaja el aprendizaje supervisado.

[2] Principio de Pareto. También conocido como la regla 80-20, es un principio de análisis y de apoyo a la toma de decisiones formulado por Vilfredo Pareto en 1987 e indica que en muchos casos el 80 % de los efectos son el producto del 20 % de las causas.

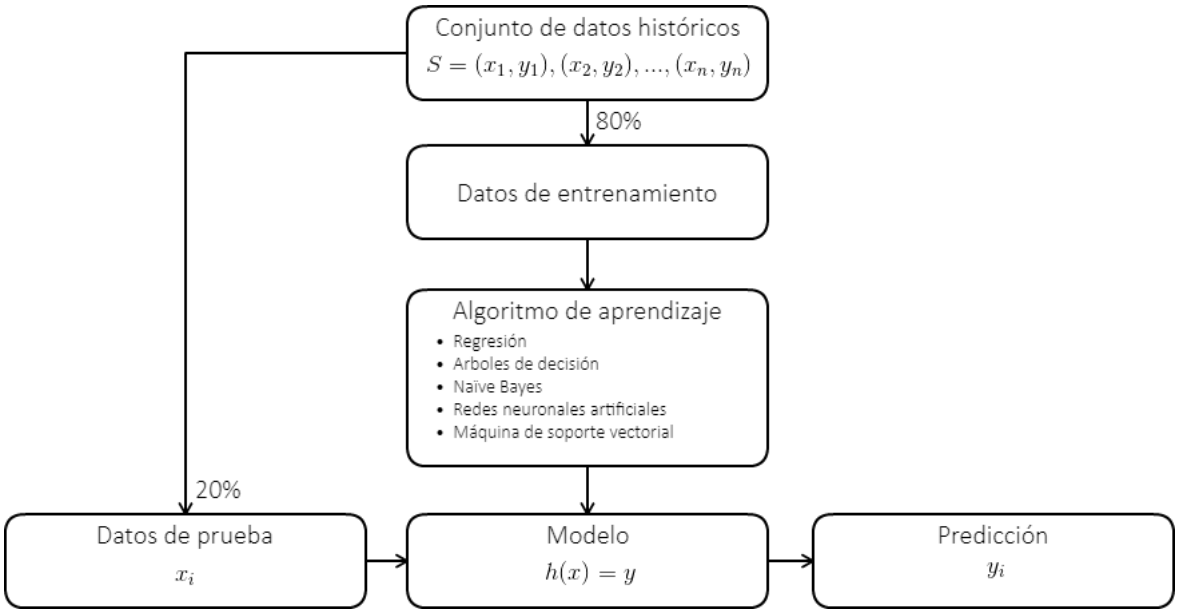


FIGURA 1.2: Aprendizaje supervisado.

Un ejemplo de este tipo de aprendizaje en la vida cotidiana es un grupo de alumnos en una clase, el profesor imparte un tema con ejemplos, da ejercicios y los alumnos intentan reproducir lo aprendido para llegar al mismo resultado.

El aprendizaje supervisado realiza tareas predictivas las cuales se dividen en dos tipos: regresión y clasificación.

1.4.1. Regresión

El objetivo del aprendizaje supervisado de tipo regresión es predecir valores numéricos. En ese tipo de tarea los atributos son valores continuos.

$$y \in \mathbb{R}$$

Por ejemplo, estimar el precio de la gasolina a partir del histórico de precios del petróleo y la demanda del combustible registrado en años anteriores.

1.4.2. Clasificación

El aprendizaje supervisado de tipo clasificación consiste en predecir la etiqueta o categoría de un elemento o individuo a partir de los atributos que los distinguen, se trata de eventos discretos. El problema de clasificación se divide en binaria y multiclase.

Binaria: sólo existen dos categorías.

$$y \in \{0, 1\}$$

Un ejemplo de este tipo de clasificación es el correo electrónico que con su filtro etiqueta los correos como “spam” y “not spam”

Multiclase: existen más de dos categorías.

$$y \in \{0, 1, 2, 3, ..., n\}$$

Un ejemplo de este tipo de clasificación sería al categorizar una noticia si es de cultura, entretenimiento, política o deportes.

1.5. Aprendizaje no supervisado

El aprendizaje no supervisado trabaja con datos compuestos solamente por el vector de entrada $x = (x_1, x_2, ..., x_n)$, no existe un vector de salida y, es decir, no se cuenta con un conjunto de datos de entrenamiento, por lo que no existe un porcentaje de predicción. Como se muestra en la figura 1.3, a partir del suministro de los datos el algoritmo de aprendizaje no supervisado va a generar un modelo que identifique alguna estructura o patrón en los datos que se usará al momento de ingresar un nuevo caso x_i.

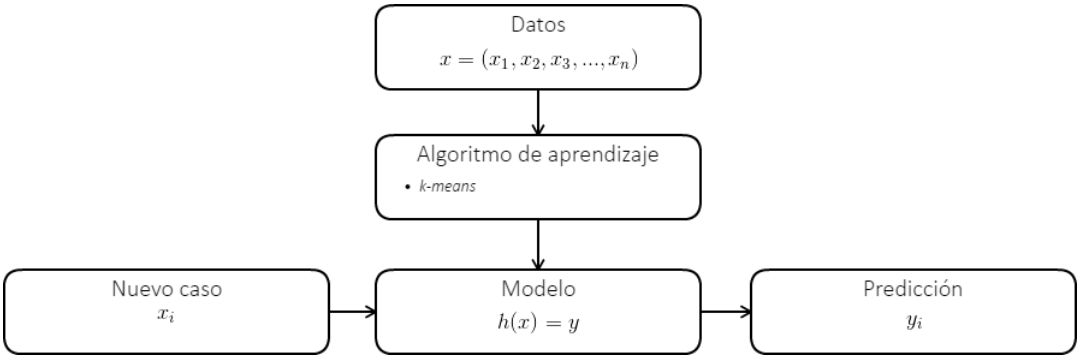


FIGURA 1.3: Aprendizaje no supervisado.

El aprendizaje no supervisado es útil para tareas de clustering (agrupamiento).

1.5.1. Clustering

En este tipo de problemas se le proporciona a la máquina un conjunto de datos y no se le dice explícitamente que hacer con ellos y tampoco cual es el punto de partida (Figura 1.4).

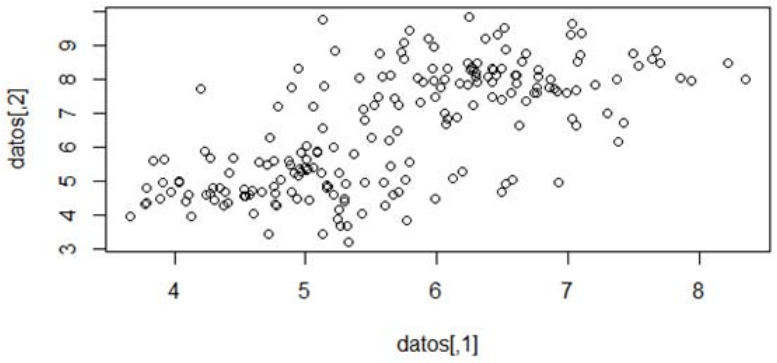


FIGURA 1.4: Conjunto de datos sin valores asignados

Con los datos de la Figura 1.4 y un algoritmo de aprendizaje no supervisado se decide dividir los datos en cuatro grupos, a esto se le conoce como problema de agrupamiento (Figura 1.5). Los grupos se logran diferenciar ya que sus datos comparten las mismas características entre ellos y son ajenos a los datos de los demás grupos.

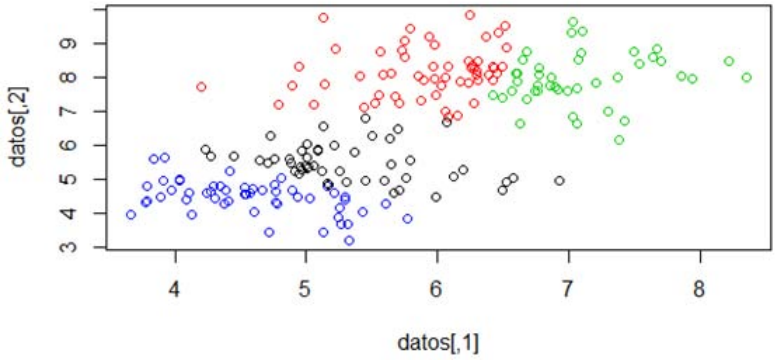


FIGURA 1.5: Conjunto de datos divididos en cuatro grupos.

Un ejemplo donde se ocupa el agrupamiento es en Google News. La finalidad de Google News es buscar nuevas historias dentro de millones de datos que constantemente se generan en Internet. Cada noticia encontrada se agrupa de manera automática en historias cohesivas a través de palabras clave que servirán como patrón de entrenamiento para generar bloques de noticias del mismo tema.

1.6. Metodología para Machine Learning

A continuación, se describe la metodología a implementar en este trabajo de investigación, planteada en una serie de pasos que involucra desde la recolección de datos hasta la generación de conocimiento.

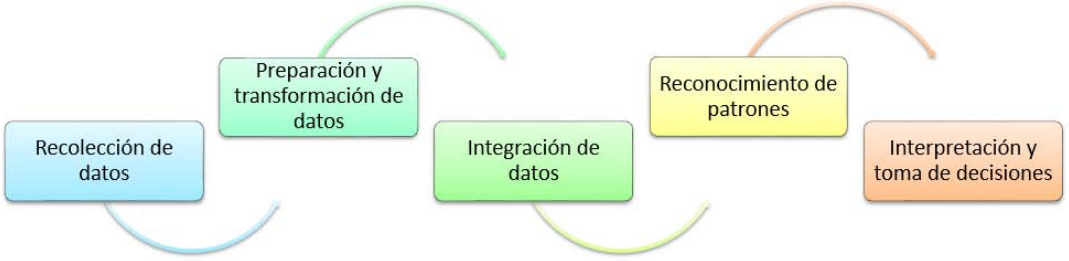


FIGURA 1.6: Metodología para Machine Learning

1. Recolección de datos. En esta etapa se identifican las fuentes de información que cumplan con los requerimientos del problema. Se extraen los datos relevantes de su medio ambiente origen desde una o varias fuentes como bases de datos públicas o bases de datos privadas.

2. Preparación y transformación de datos. Se le da tratamiento a los datos faltantes, anómalos o inconsistentes al aplicar técnicas de análisis de datos que incluyen medidas estadísticas y gráficas utilizadas para visualizar el comportamiento y mejorar la calidad en los datos. Los datos son transformados en forma apropiada para facilitar el uso de algoritmos de aprendizaje ya sea reemplazar valores continuos por discretos o valores discretos por continuos.

3. Integración de datos. Los datos que provienen de múltiples fuentes con frecuencia son combinados en un almacén de datos (*Data Warehouse*) para que todos los datos tengan un formato en común. Aunque no es imprescindible. En algunos casos, se suele trabajar con los datos originales en archivos de texto plano u hojas de cálculo.

4. Reconocimiento de patrones. Este paso es fundamental es donde se decide el tipo de tarea de aprendizaje a realizar (regresión, clasificación o agrupamiento) así como el algoritmo de Machine Learning más apropiado para la construcción del modelo.

5. Interpretación y toma de decisiones. Al obtener un modelo que sea óptimo para resolver el problema se pasa a la etapa de interpretación. En esta etapa se hace uso de la estadística descriptiva para presentar los datos de forma gráfica, facilitado el análisis del conocimiento que se ha generado y ponerlo a nivel de usuario. Una

vez que el modelo está en uso el científico de datos [3] elige soluciones estratégicas basadas en los resultados obtenidos en la etapa anterior.

1.6.1. Metodología CRISP-DM

Existen otras metodologías similares para extraer conocimiento de los datos como lo es la Metodología CRISP-DM.

La Metodología CRISP-DM (*Cross-Industry Standard Process for Data Mining*) es una de las principales metodologías utilizada en trabajos de minería de datos, fue desarrollada en 1996 por el consorcio de empresas europeas DaimlerChrysler, SPSS (*Statistical Product and Service Solutions*) y NCR Systems Engineering Copenhagen.

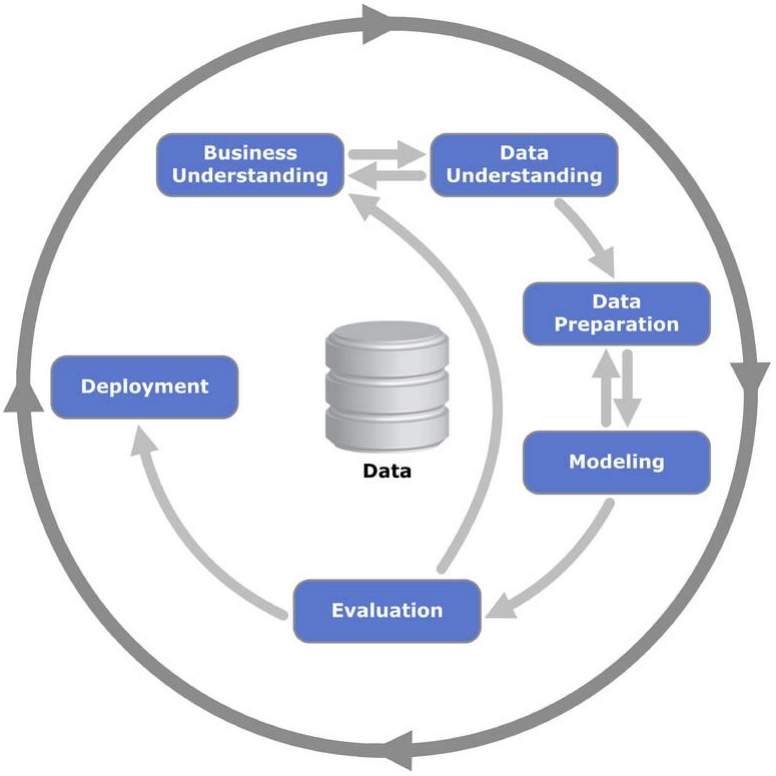


FIGURA 1.7: Metodología CRISP-DM
Obtenido de:https://es.wikipedia.org/wiki/Cross_Industry_Standard_Process_for_Data_Mining

Esta metodología consta de 6 etapas que son:

[3]Científico de datos. Es el encargado de extraer conocimiento tanto de datos estructurados como no estructurados. Teniendo sólidas bases en matemáticas, estadística y computación. Además de tener la habilidad de transmitir recomendaciones a los responsables del negocio.

1. Análisis del negocio. En esta fase se debe de conocer los objetivos del negocio y lo que el cliente desea saber, la finalidad es elaborar un plan de actividades que guíe el desarrollo del proyecto para cumplir con los objetivos del negocio, especificando el tipo de herramientas y las técnicas a usar, así como los recursos, limitaciones y riesgos.

2. Análisis de datos. Consiste en la recolección y valoración de los datos, para ver si cumplen con las necesidades del negocio. Además, se evalúan los requerimientos de la técnica de minería de datos seleccionada para hacer uso de esos datos.

3. Preparación de los datos. Se seleccionan lo datos con los que se va a construir el modelo. Esta etapa implica la limpieza de los datos, la generación de variables adicionales y los cambios de formato. Si es necesario se realiza la integración de la información en un almacén de datos desde múltiples fuentes de datos como base de datos, registros o tablas.

4. Modelado. Se seleccionan las técnicas apropiadas para el desarrollo del trabajo de minería de datos. Es posible construir más de un modelo y ajustar sus parámetros cuantas veces sea necesario hasta satisfacer las necesidades del negocio, es por ello, que se necesitan poseer los conocimientos necesarios para aplicar las técnicas seleccionadas, generalmente es una tarea asignada a los científicos de datos.

5. Evaluación. Se evalúa que el modelo construido cumpla con todos los requerimientos iniciales. Cada uno de los modelos aprobados se someten a una revisión para garantizar su calidad, por ejemplo, si se construyó correctamente o para determinar si los atributos están disponibles para hacer análisis futuros. De acuerdo a los resultados de la revisión, el equipo decide si el proyecto pasa a implementación o retrocede algunas etapas.

6. Implementación. Si el modelo generado cumple con todos los requerimientos se procede a su implementación, documentando los pasos necesarios y cómo realizarlos, así mismo se planifica una estrategia de monitoreo y mantenimiento como parte del uso correcto de los resultados de la minería de datos. Adicionalmente, se redacta un informe final con los resultados y experiencias adquiridas durante el trabajo, presentando los resultados al cliente y finalmente evaluando las actividades que salieron bien y mal, lo que se hizo bien y lo que hay que mejorar.

Capítulo 2

Algoritmos de aprendizaje

2.1. Introducción

En este capítulo se hablará de las características de los algoritmos de aprendizaje supervisado y no supervisado. Incluyendo ejemplos de aplicación de cada uno de ellos, así como las ventajas y desventajas de su uso.

En seguida se da una introducción a cada uno de los algoritmos que se emplearon en este trabajo de investigación

Algoritmos de aprendizaje supervisado:

Regresión. Consiste en predecir el valor de una variable dependiente o de respuesta (y) con base en una o más variables independientes (x). En este algoritmo se abordará la regresión lineal y no lineal para la solución de problemas con valores continuos [26].

Árboles de decisión. Este es un algoritmo de clasificación que tiene una estructura similar a la de un árbol debido a que va dividiendo los datos de manera iterativa desde la raíz hasta convertirlos en hojas o nodos terminales. Cada rama del árbol es el valor del atributo que se colocó como raíz o nodo interno y brinda los posibles caminos a seguir para llegar a un nodo terminal. Para su construcción se desarrollarán los algoritmos de aprendizaje ID3 y C4.5 propuestos por Ross Quinlan [2].

Clasificador Naïve Bayes. Es uno de los algoritmos más populares de Machine Learning por su rapidez y exactitud. Este algoritmo está basado en el Teorema de Bayes y simplifica el cálculo de las probabilidades condicionales al asumir la independencia de los atributos. La construcción de modelos con Naïve Bayes es considerada fácil y es particularmente útil en grandes conjuntos de datos históricos [19].

Redes neuronales artificiales (RNA). Son un algoritmo que intenta emular el funcionamiento del cerebro humano en procesos de aprendizaje por medio de la interconexión de neuronas. Gráficamente se representan por un conjunto de nodos conectados por arcos, que dependiendo del tipo de linealidad vista en el conjunto de datos se modela con una arquitectura del perceptrón simple o perceptrón multicapa. Las RNA son capaces de clasificar los datos suministrados al aplicar el algoritmo de aprendizaje que corresponda al número de capas que se tienen, por lo que se proponen los algoritmos del perceptrón y de retropropagación [6][11].

Máquina de soporte de vectores (SVM, del inglés *Support Vector Machine*). Los fundamentos teóricos de la máquina de soporte de vectores tienen origen en los trabajos sobre la teoría del aprendizaje estadístico de Vladimir Vapnik. La máquina de soporte de vectores surgió como un método de clasificación binaria que consiste en separar un conjunto de datos en dos clases (positiva y negativa) mediante el uso de un hiperplano que de la máxima separación entre los datos pertenecientes a cada clase, este hiperplano queda definido por los datos que se encuentren más cercanos a él, a estos datos se les conoce como vectores de soporte [2][54].

En algoritmos como arboles de decisión, Naïve Bayes y Máquina de soporte de vectores se utiliza la matriz de confusión o tabla de contingencia (Figura 2.1) para conocer el número de datos que se clasificaron correctamente e incorrectamente. Donde los valores de la diagonal representan los datos que fueron clasificados correctamente (verdadero positivo y verdadero negativo) y los valores fuera de la diagonal son los datos que se clasificaron incorrectamente (falso positivo y falso negativo).

		clase real	
		positiva	negativa
predicción	positiva	verdadero positivo (tp)	falso positivo (fp)
	negativa	falso negativo (fn)	verdadero negativo (tn)

FIGURA 2.1: Matriz de confusión
Obtenida de: Benitez, R. (2014). *Inteligencia artificial avanzada.* (1ª ed.). Barcelona: UOC.Página 158.

Algoritmo de aprendizaje no supervisado:

K-medias. Es un algoritmo de agrupamiento que fue propuesto por James McQueen en el año de 1967, es el algoritmo de aprendizaje no supervisado más utilizado por ser simple y eficaz en su aplicación, consiste en dividir un conjunto de n datos en un determinado número de K clusters. Cada cluster se caracteriza por tener un centro o centroide que representa la media ponderada de los datos que lo componen, los datos de cada cluster comparten elementos o características similares entre sí. El objetivo del

algoritmos K-medias es minimizar la distancia euclideana entre los valores de x y el centroide μ de cada cluster [8][28].

2.2. Regresión

Hay dos tipos de regresión: lineal y no lineal.

2.2.1. Regresión lineal simple

En la regresión lineal simple solo existe una variable independiente x, se busca la ecuación de una línea recta que se ajuste a un conjunto de datos $(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)$. Por lo tanto la ecuación de la recta a encontrar se define de la siguiente manera [5][24]:

$$y = \theta_0 + \theta_1 x \tag{2.1}$$

La ecuación 2.1 se conoce también como ecuación ordinaria de la recta donde θ_0 es la ordenada al origen y θ_1 es la pendiente, estos parámetros indican la posición y la inclinación de la recta como se aprecia en la figura 2.2.

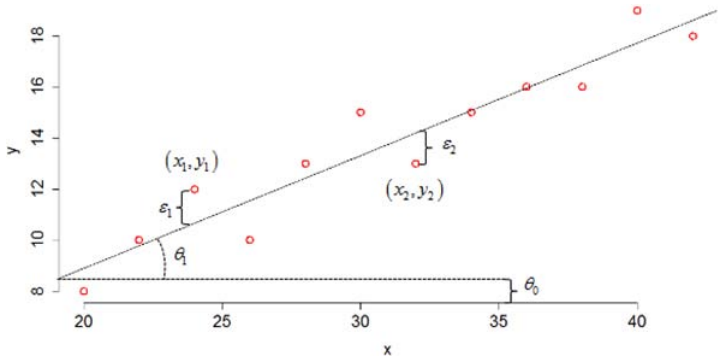


FIGURA 2.2: Componentes de la recta de regresión.

Los parámetros de la ecuación 2.1 se estiman con el método de mínimos cuadrados, estos valores servirán para trazar la línea recta que mejor se ajuste al conjunto de datos [26].

$$\theta_1 = \frac{\sum_{i=1}^{n} (x_i - \overline{x})(y_i - \overline{y})}{\sum_{i=1}^{n} (x_i - \overline{x})} \tag{2.2}$$

$$\theta_0 = \overline{y} - \theta_1 \overline{x} \tag{2.3}$$

Se dice que hay un buen ajuste en los datos cuando la suma del error al cuadrado producido entre los puntos (x, y) y la línea de regresión alcanza un promedio cercano a cero .

$$\sum e_i^2 = \sum_{i=1}^{n} (y_i - (\theta_0 + \theta_1 x_i))^2 \tag{2.4}$$

2.2.2. Regresión no lineal

En muchas ocasiones un conjunto de datos no pueden ser descritos por una línea recta, por lo que se hace uso de la regresión no lineal para encontrar una curva que de una mejor relación entre x y y [5][24].

La regresión no lineal utiliza funciones polinómicas, logarítmicas o exponenciales (Figura 2.3).

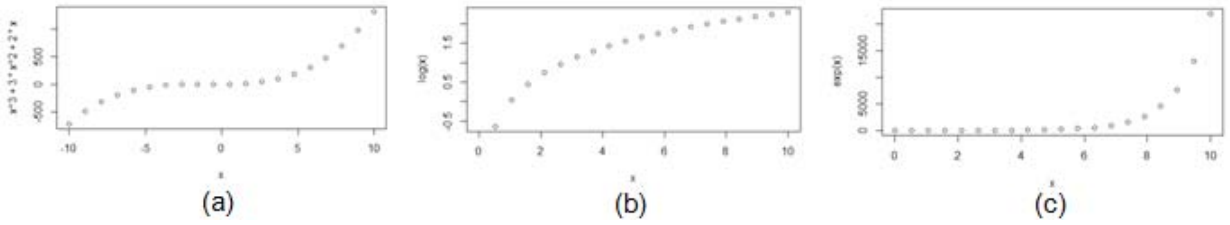


FIGURA 2.3: (a) Función cúbica; (b) Función logarítmica; (c) Función exponencial.

La ecuación para la regresión polinómica es:

$$y = \theta_0 + \theta_1 x + \theta_2 x^2 + ... + \theta_k x^k \tag{2.5}$$

Los parámetros $\theta_0, \theta_1, \theta_2, ..., \theta_k$ también se estiman mediante el uso de mínimos cuadrados, a diferencia del caso lineal se construye un sistema de $k+1$ ecuaciones lineales a partir del grado del polinomio (k). La solución del sistema de ecuaciones corresponde a la curva de mejor ajuste [5].

$$\begin{array}{ccccccccc} \theta_0 n & + & \theta_1 \sum x_i & + & \dots & + & \theta_k \sum x_i^k & = & \sum y_i \\ \theta_0 \sum x_1 & + & \theta_1 \sum x_i^2 & + & \dots & + & \theta_k \sum x_i^{k+1} & = & \sum x_i y_i \\ \vdots & & & & \vdots & & \vdots & & \\ \theta_0 \sum x_i^k & + & \theta_1 \sum x_i^{k+1} & + & \dots & + & \theta_k \sum x_i^{2k} & = & \sum x_i^k y_i \end{array} \tag{2.6}$$

La bondad de ajuste de una función polinómica a los datos de entrenamiento se calcula:

$$\sum e_i^2 = \sum_{i=1}^{n} [y_i - (\theta_0 + \theta_1 x_i + \theta_2 x_i^2 + ... + \theta_k x_i^k)]^2 \tag{2.7}$$

En el caso de las funciones logarítmica (2.8) y exponencial (2.9) con ecuaciones de regresión, se utiliza una equivalencia lineal (Tabla 2.1) debido a que es uno de los procedimientos más sencillo en la construcción de modelos de regresión no lineales, además el cálculo de los parámetros θ se hace con el método de mínimos cuadrados de la regresión lineal (Ecuación 2.1) [22].

$$y = \theta_0 + \theta_1 log(x) \tag{2.8}$$

$$y = e^{\theta_0 + \theta_1 x} \tag{2.9}$$

En la Tabla 2.1, se muestran las equivalencias de esas funciones.

Modelo	Modelo no lineal	Modelo lineal
Logarítmica	$y = \theta_0 + \theta_1 log(x)$	$y = \theta_0 + \theta_1 log(x)$
Exponencial	$y = e^{\theta_0 + \theta_1 x}$	$log(y) = \theta_0 + \theta_1 x$

TABLA 2.1: Transformaciones de las funciones logarítmica y exponencial.

2.2.3. Algoritmo de aprendizaje

El proceso del algoritmo de regresión se divide en dos fases. En la primera fase se utilizan los datos de entrenamiento para la construcción del modelo de aprendizaje a partir de identificar la mejor relación lineal o no lineal en dichos datos. Una vez que el modelo se obtiene se pasa a la segunda fase que consiste en introducir los datos de prueba en el modelo para calcular tendencias futuras.

Los pasos para la construcción el modelo de aprendizaje son los siguientes:

1. Graficar el conjunto de datos de entrenamiento. A esto se le llama diagrama de dispersión.
2. De acuerdo a la forma que tome el conjunto de datos de entrenamiento en el diagrama de dispersión se selecciona el tipo de función lineal o no lineal.
3. Calcular los parámetros θ con el método de mínimos cuadrados, utilizando las ecuaciones 2.2 y 2.3 si es lineal de lo contrario el sistema de ecuaciones 2.6 si es no lineal.

4. Calcular el error, empleando la ecuación 2.4 si es lineal o la ecuación 2.7 si es no lineal.

2.2.4. Ejemplo de aplicación

Un cine realiza un estudio para determinar el número de publicaciones semanales que necesitan postear en su cuenta oficial de Facebook para atraer nuevos seguidores ("Me gusta").

La tabla 2.2 corresponde al conjunto de entrenamiento en que aparecen el número de publicaciones que se realizaron durante 10 semanas y el número de seguidores que consiguieron por esas publicaciones.

Publicaciones	Seguidores
3	30
8	57
9	64
13	72
3	36
6	43
11	59
21	90
1	20
16	83

TABLA 2.2: Conjunto de entrenamiento. Publicaciones y seguidores en Facebook

Al graficarlos el diagrama de dispersión se ve de la siguiente manera Figura 2.4. El eje x representa las publicaciones y el eje y el total de seguidores.

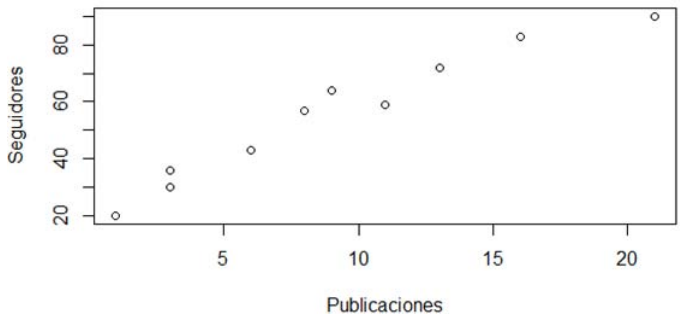


FIGURA 2.4: Diagrama de dispersión. Publicaciones y seguidores en Facebook

Cómo la relación entre las publicaciones y los seguidores parece ajustarse a una línea recta el modelo de aprendizaje usa la función 2.1, donde los parámetros θ_0 y θ_1 se estiman mediante el método de mínimos cuadrados (Ecuación 2.2 y 2.3). Al aplicar este método se obtiene como resultado $y = 23.6 + 3.5x$ que corresponde a la recta trazada sobre el conjunto de entrenamiento (Figura 2.5)

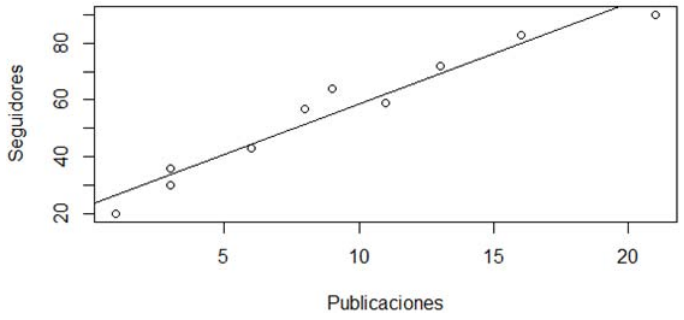


FIGURA 2.5: Recta de regresión. Publicaciones y seguidores en Facebook

En la figura 2.6 se ve como cada punto de datos en el diagrama de dispersión tiene un error asociado con su distancia respecto a la recta.

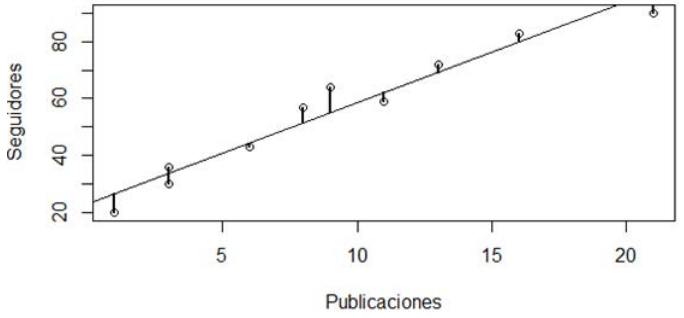


FIGURA 2.6: Error asociado a la recta de regresión.

Al aplicar la suma del error cuadrado (Ecuación 2.4) da como resultado 261.75, se considera que el error es mínimo pues la recta es la que mejor se ajusta al conjunto de datos de entrenamiento.

El modelo de aprendizaje se pone a prueba cuando el dueño del cine quiere saber cuántos seguidores tendrá en una semana si hace 15 publicaciones.

En este caso el valor de las publicaciones se sustituye en el modelo de aprendizaje

$y = 23.6 + 3.5(15)$

$y = 76.1$

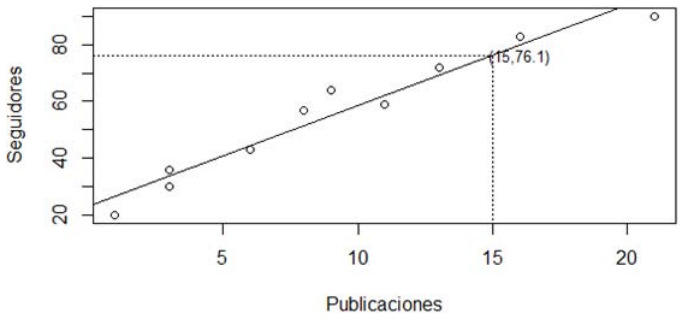


FIGURA 2.7: Prueba del modelo.

En la figura 2.7 se observa que la predicción realizada con el modelo de aprendizaje corresponde a la relación entre las publicaciones y seguidores con respecto a la linea de regresión.

Por lo que se estima que las suscripciones a la cuenta oficial de Facebook del cine serán de 76 usuarios si deciden publicar 15 comentarios a la semana.

2.2.5. Ventajas y desventajas de regresión

A continuación se describen algunas ventajas y desventajas del algoritmo de regresión [24][60].

Ventajas	**Desventajas**
•Realizar predicciones del comportamiento de una variable en un determinado punto o momento. •Proporciona un medio visual para probar si existe una relación entre dos o más variables. •Facilita y agiliza la toma de decisiones al estimar los valores por medio del modelo de aprendizaje o la gráfica de regresión. •La construcción del modelo de aprendizaje es simple porque el tipo de función seleccionada indica el número de parámetros a calcular y la posición en donde se tienen que sustituir para generar el modelo. •Es útil para predecir precios, ganancias y perdidas.	•Es sensible a los datos atípicos, es decir, valores que parecen no corresponder al conjunto de datos. •Sólo trabaja con datos numéricos.

2.3. Árboles de decisión

2.3.1. Construcción de un árbol de decisión

La construcción de un árbol de decisión es un proceso iterativo, en el que en cada iteración se selecciona el mejor atributo que particione al conjunto de datos de entrenamiento. Para realizar este proceso se tienen que efectuar los siguientes pasos [2].

1. Calcular la bondad de las particiones de los atributos. La bondad es la medida que describe el ajuste de un conjunto de datos hacia cada partición.
2. Seleccionar el mejor atributo. El mejor atributo es el que tiene mayor bondad.
3. El mejor atributo pasará a ser un nodo del árbol. Si todos los ejemplos del conjunto de entrenamiento han quedado bien clasificados se convierte en un nodo terminal. En caso contrario se convierte en un nodo interno y se aplica una nueva iteración.

2.3.2. Ejemplo de aplicación

Una compañía de televisión por cable ha solicitado un estudio de mercadotecnia para determinar qué tipo de clientes contratarían una suscripción a su servicio de telefonía e internet. El estudio se realiza con base a un histórico de datos (Tabla 2.3) que se compone de los siguientes atributos: salario, edad, hijos y si son buenos clientes.

Número de cliente	Salario	Edad	Hijos	¿Buen cliente?
1	Alto	Joven	Si	No
2	Alto	Joven	No	No
3	Medio	Joven	Si	Si
4	Bajo	Joven	Si	Si
5	Bajo	Mayor	Si	Si
6	Bajo	Mayor	No	No
7	Medio	Mayor	No	Si
8	Alto	Joven	Si	No
9	Alto	Mayor	Si	Si
10	Bajo	Mayor	Si	Si
11	Alto	Mayor	No	Si
12	Medio	Joven	No	Si
13	Medio	Mayor	Si	Si
14	Bajo	Joven	No	Si

Tabla 2.3: Conjunto de datos de entrenamiento. Clientes de la compañía de televisión por cable.

Se empieza la construcción del árbol calculando la bondad de las particiones de cada atributos. El atributo salario tiene tres particiones: alto, medio y bajo (Tabla 2.4).

Salario			
¿Buen clientes	Alto	Medio	Bajo
Si	2	4	4
No	3	0	1
Total	5	4	5

Tabla 2.4: Particiones del atributo salario.

Para calcular la bondad se suman los valores máximos de cada partición y se divide entre el número total de clientes.

$$bondad(Salario) = \frac{3+4+4}{14} = 0.78$$

Se repite este procedimiento para los atributos Edad (Tabla 2.5) e Hijos (Tabla 2.6)

Edad		
¿Buen clientes	Joven	Mayor
Si	4	6
No	3	1
Total	7	7

Tabla 2.5: Particiones del atributo edad.

$$bondad(Edad) = \frac{4+6}{14} = 0.71$$

Hijos		
¿Buen clientes	Si	No
Si	6	2
No	4	2
Total	10	4

Tabla 2.6: Particiones del atributo hijos.

$$bondad(Hijos) = \frac{6+2}{14} = 0.57$$

El mejor atributo resultó ser salario ya que es el de mayor bondad. Por lo tanto, se coloca como raíz y sus ramas son alto, medio y bajo (Figura 2.8).

FIGURA 2.8: Atributo salario como raíz.

Para seleccionar el nodo que pertenece a cada rama se realiza una segunda iteración. Empezando por la rama de salario alto (Tabla 2.7).

Salario Alto			
Salario	Edad	Hijos	¿Buen cliente?
Alto	Joven	Si	No
Alto	Joven	No	No
Alto	Joven	Si	No
Alto	Mayor	Si	Si
Alto	Mayor	No	Si

TABLA 2.7: Rama de salario alto

Se calcula la bondad del atributo Edad (Tabla 2.8) e Hijos (Tabla 2.9) de los clientes que tienen un salario alto.

Alto,Edad		
¿Buen cliente?	Joven	Mayor
Si	0	2
No	3	0
Total	3	2

TABLA 2.8: Atributo edad para la rama de salario alto.

$$bondad(Edad) = \frac{3+2}{5} = 1$$

Alto,Hijos		
¿Buen clientes	Si	No
Si	1	2
No	1	1
Total	2	3

Tabla 2.9: Atributo hijos para la rama de salario alto.

$$bondad(Hijos) = \frac{2+1}{5} = 0.6$$

El atributo de mayor bondad es Edad, se coloca como un nodo interno con dos ramas "Joven" y "Mayor", ya que el atributo "¿Buen cliente?" de la Tabla 2.7 no pertenece a una sola clase. Se requiere de otra iteración hasta encontrar un nodo terminal.

Ahora utilizando las tuplas correspondientes a salario medio.

Salario Medio			
Salario	Edad	Hijos	¿Buen cliente?
Medio	Joven	Si	Si
Medio	Mayor	No	Si
Medio	Joven	No	Si
Medio	Mayor	Si	Si

Tabla 2.10: Rama de salario medio

Como se muestra en la Tabla 2.10 todos los clientes con salario medio pertenecen a la clase de buenos clientes por lo que en la rama de valor medio se coloca un nodo terminal "Si".

Y finalmente con las tuplas de salario bajo.

Salario Bajo			
Salario	Edad	Hijos	¿Buen cliente?
Bajo	Joven	Si	Si
Bajo	Mayor	Si	Si
Bajo	Mayor	No	No
Bajo	Mayor	Si	Si
Bajo	Joven	No	Si

Tabla 2.11: Rama de salario bajo

Se calcula la bondad del atributo Edad (Figura 2.12) e Hijos (Figura 2.13)

Bajo,Edad		
¿Buen clientes	Joven	Mayor
Si	2	2
No	0	1
Total	2	3

Tabla 2.12: Atributo edad para la rama de salario bajo.

$$bondad(Edad) = \frac{2+2}{5} = 0.8$$

Bajo,Hijos		
¿Buen clientes	Si	No
Si	3	0
No	1	1
Total	4	1

Tabla 2.13: Atributo hijos para la rama de salario bajo.

$$bondad(Edad) = \frac{3+1}{5} = 0.8$$

Existe un empate entre estos dos atributos. Se elige como nodo interno el atributo Hijos ya que el atributo Edad ya se utilizo para la rama Alto.

Se repite este procedimiento hasta que todos los nodos del árbol sean nodos terminales. En la Figura 2.9 se presenta el árbol de decisión terminado.

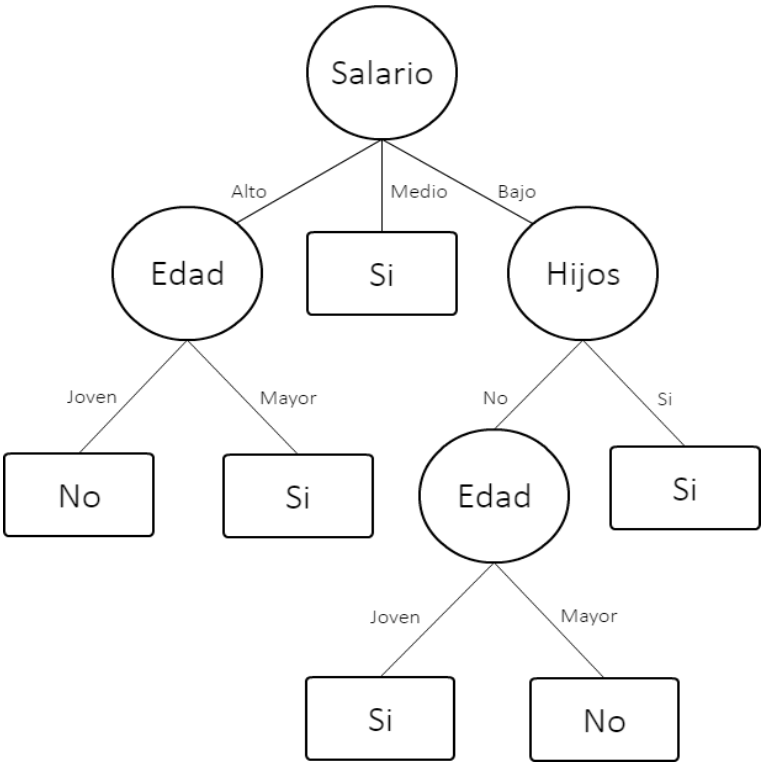


FIGURA 2.9: Árbol de decisión. Compañía de cable.

Para poner a prueba el árbol de decisión construido se ingresa una tupla del conjunto de datos de prueba (Tabla 2.14).

Número de cliente	Salario	Edad	Hijos	¿Buen cliente?
15	Alto	Joven	No	No

TABLA 2.14: Caso de prueba

En este caso se recorre el árbol bajando por la rama de salario alto llegando al atributo edad y se baja por la rama de edad joven hasta llegar al nodo terminal que indica que este no es un buen cliente. De modo que, el árbol construido funciona correctamente.

A partir de la construcción del árbol de decisión la compañía de televisión por cable toma la decisión de invertir en promociones para aquellos clientes que cumplan con ciertas características.

- Clientes con salario bajo y con hijos.
- Clientes jóvenes con salario bajo.
- A todos los clientes con salario medio.
- Clientes de edad mayor con salario alto.

2.3.3. Algoritmos de aprendizaje para árboles de decisión

Existen diferentes técnicas de construcción de árboles de decisión, pero las más empleadas en Machine Learning son los algoritmos ID3 y C4.5 estos algoritmos utilizan la ganancia de información y la entropía para calcular la bondad de cada atributo y de esta manera seleccionar el mejor atributo para comenzar la construcción del árbol hasta llegar a los nodos terminales.

2.3.3.1. ID3

El ID3 es un algoritmo matemático para la construcción de árboles de decisión que fue desarrollado en 1986 por Ross Quinlan. El conjunto de datos de entrenamiento para este algoritmo requiere de valores discretos y consistentes, es decir, la no existencia de datos faltantes. La bondad de cada atributo se obtiene mediante una prueba estadística que se conoce como ganancia de la información. Dentro del concepto de ganancia de la información es importante definir una medida utilizada comúnmente en la teoría de la información "la entropía" [19].

Dado un conjunto de entrenamiento, la entropía está dada por:

$$Entropia(S) = -(\frac{p}{p+n})\log_2(\frac{p}{p+n}) - (\frac{n}{p+n})\log_2(\frac{n}{p+n}) \tag{2.10}$$

Donde p es la cantidad de ejemplos positivos en S y n es la cantidad de ejemplos negativos en S.

La entropía será mínima (0) cuando todos los ejemplos de S están clasificados correctamente y será máxima (1) cuando los ejemplos de S estén mal clasificados.

La fórmula de la ganancia de información para el conjunto de entrenamiento S y el atributo x queda como:

$$Ganancia(S, x) = Entropia(S) - \sum_{i=1}^{\#particiones} p(x_i)Entropia(x_i) \tag{2.11}$$

2.3.3.2. Ejemplo

Retomando el ejemplo de aplicación 2.3.2, pero ahora aplicando el algoritmo ID3.

Primero se calcula la entropía (Ecuación 2.10) del conjunto de entrenamiento (Tabla 2.3), correspondiente al número total de clientes positivos y negativos.

¿Buen cliente?	
Si	10
No	4
Total	14

Tabla 2.15: Total de clientes positivos y negativos.

$E(S) = -(\frac{10}{14})\log_2(\frac{10}{14}) - (\frac{4}{14})\log_2(\frac{4}{14}) = 0.8631$

El algoritmo ID3 analiza todas las posibles divisiones según los distintos atributos y calcula la ganancia. Se comienza analizando el atributo de Salario.

Salario			
¿Buen clientes	Alto	Medio	Bajo
Si	2	4	4
No	3	0	1
Total	5	4	5

Tabla 2.16: Particiones del atributo salario (ID3).

Se calcula la entropía del atributo Salario (x) en relación al atributo "¿Buen cliente?"(S). Para ello se multiplica la probabilidad de cada partición (i: Alto, Medio, Bajo) con su correspondiente entropía.

$$\begin{aligned} E(S, Salario) &= \sum P(Salario_i)E(Salario_i) \\ &= P(Alto)E(Alto) + P(Medio)E(Medio) + P(Bajo)E(Bajo) \\ &= \frac{5}{14}(-\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5}) + \frac{4}{14}(-\frac{4}{4}\log_2\frac{4}{4} - \frac{0}{4}\log_2\frac{0}{4}) + \frac{5}{14}(-\frac{4}{5}\log_2\frac{4}{5} - \frac{1}{5}\log_2\frac{1}{5}) \\ &= 0.6046 \end{aligned}$$

Enseguida se calcula la ganancia (Ecuación 2.11) restando la entropía del atributo Salario de la entropía del total de clientes positivos y negativos.

$$\begin{aligned} G(S, Salario) &= E(S) - E(S, Salario) \\ &= 0.8631 - 0.6046 \\ &= 0.258 \end{aligned}$$

De la misma manera en que se calcula la entropía y la ganancia para el atributo Salario, se calcula para el atributo Edad (Tabla 2.17) e Hijos (Tabla 2.18) obteniendo los siguientes resultados.

Edad		
¿Buen cliente?	Joven	Mayor
Si	4	6
No	3	1
Total	7	7

Tabla 2.17: Particiones del atributo edad (ID3).

$$\begin{aligned} E(S, Edad) &= P(Joven)E(Joven) + P(Mayor)E(Mayor) \\ &= \frac{7}{14}(-\frac{4}{7}\log_2\frac{4}{7} - \frac{3}{7}\log_2\frac{3}{7}) + \frac{7}{14}(-\frac{6}{7}\log_2\frac{6}{7} - \frac{1}{7}\log_2\frac{1}{7}) \\ &= \frac{7}{14}(0.9852) + \frac{7}{14}(0.5916) \\ &= 0.7884 \end{aligned}$$

$$\begin{aligned} G(S, Edad) &= E(S) - E(S, Edad) \\ &= 0.8631 - 0.7884 \\ &= 0.0746 \end{aligned}$$

Para el atributo Hijos se obtienen los siguientes resultados:

Hijos		
¿Buen cliente?	Si	No
Si	6	2
No	4	2
Total	10	4

Tabla 2.18: Particiones del atributo hijos (ID3).

$$\begin{aligned} E(S, Hijos) &= P(Si)E(Si) + P(No)E(No) \\ &= \frac{8}{14}(-\frac{6}{8}\log_2\frac{6}{8} - \frac{2}{8}\log_2\frac{2}{8}) + \frac{6}{14}(-\frac{4}{6}\log_2\frac{4}{6} - \frac{2}{6}\log_2\frac{2}{6}) \\ &= \frac{8}{14}(0.8112) + \frac{6}{14}(0.9183) \\ &= 0.8571 \end{aligned}$$

$$
\begin{aligned}
G(S, Hijos) &= E(S) - E(S, Hijos) \\
&= 0.8631 - 0.8571 \\
&= 0.0059
\end{aligned}
$$

Una vez que se han calculado la entropía y ganancia para todos los atributos disponibles, se elige el atributo que dividirá el conjunto de datos que es aquel que tenga la mayor ganancia. Esto significa que el nodo raíz del árbol será el atributo Salario. Al igual que el ejemplo 2.3.2, se requiere de un proceso iterativo para encontrar los nodos terminales, esto se determina al obtener el valor de cero en la entropía. Finalmente se obtiene el mismo árbol de la Figura 2.9.

2.3.3.3. C4.5

El algoritmo C4.5 fue desarrollado por Quinlan en 1993, es una versión mejorada del algoritmo ID3 tienen la misma estructura ya que calculan la bondad de los atributos con los criterios de entropía y ganancia de la información. Su principal diferencia es que el algoritmo C4.5 permite trabajar con un conjunto de datos de entrenamiento tanto con valores discretos como continuos y maneja atributos con valores faltantes.

El algoritmo C4.5 utiliza la proporción de ganancia como prueba de bondad en cada atributo y es una modificación de la ganancia de información (Ecuación 2.11), la cual reduce el sesgo al seleccionar los atributos, es decir, construye árboles de decisión más pequeños [36].

La proporción de ganancia considera el número y tamaño de las ramas al escoger un atributo y se define como:

$$
Proporcion\_de\_ganancia(S, x) = \frac{Ganancia(S, x)}{I\_division(S, x)} \tag{2.12}
$$

donde la división de la información reduce la proporción de ganancia y representa la información generada al dividir el conjunto de entrenamiento S en k particiones, lo que corresponde a la información necesaria para asignar una rama a un atributo. La división de información está dada por:

$$
I\_division(S, x) = -\sum_{k=1}^{n} P(\frac{k}{S}) \log_2 P(\frac{k}{S}) \tag{2.13}
$$

2.3.3.4. Ejemplo

Se retoma el ejemplo de aplicación 2.3.2 pero ahora con un dato desconocido en el atributo Salario.

Número de cliente	Salario	Edad	Hijos	¿Buen cliente?
1	?	Joven	Si	No
2	Alto	Joven	No	No
3	Medio	Joven	Si	Si
4	Bajo	Joven	Si	Si
5	Bajo	Mayor	Si	Si
6	Bajo	Mayor	No	No
7	Medio	Mayor	No	Si
8	Alto	Joven	Si	No
9	Alto	Mayor	Si	Si
10	Bajo	Mayor	Si	Si
11	Alto	Mayor	No	Si
12	Medio	Joven	No	Si
13	Medio	Mayor	Si	Si
14	Bajo	Joven	No	Si

Tabla 2.19: Conjunto de entrenamiento con un dato desconocido.

El algoritmo C4.5 analiza cada uno de los atributos candidatos para convertirse en raíz.

PRIMERA ITERACIÓN. Se requiere del cálculo de la entropía (Ecuacion 2.10) del atributo ¿Buen cliente? Del conjunto de entrenamiento (Tabla 2.19), sin tomar en cuenta la tupla del valor desconocido (?). Así que, se trabaja con un total de 13 casos, de los cuales 10 son positivos y 3 negativos.

¿Buen cliente?	
Si	10
No	3
Total	13

Tabla 2.20: Total de clientes positivos y negativos descartando la tupla con valor desconocido.

$$E(S) = -\frac{10}{13}\log_2\frac{10}{13} - \frac{3}{13}\log_2\frac{3}{13} = 0.7793$$

Se estima la entropía de la división resultante del atributo Salario.

Salario				
¿Buen clientes	Desconocido	Alto	Medio	Bajo
Si	1	2	4	4
No	0	2	0	1
Total	1	4	5	5

Tabla 2.21: Partición del atributo salario (C4.5).

$$\begin{aligned} E(S, Salario) &= P(Alto)E(Alto) + P(Medio)E(Medio) + P(Bajo)E(Bajo) \\ &= \frac{4}{13}(-\frac{2}{4}\log_2\frac{2}{4} - \frac{2}{4}\log_2\frac{2}{4}) + \frac{4}{13}(-\frac{4}{4}\log_2\frac{4}{4} - \frac{0}{4}\log_2\frac{0}{4}) + \frac{5}{13}(-\frac{4}{5}\log_2\frac{4}{5} - \frac{1}{5}\log_2\frac{1}{5}) \\ &= 0.5853 \end{aligned}$$

Como siguiente paso se utiliza la ecuación de ganancia de información (Ecuación 2.11), a la que adicionalmente se multiplicara la proporción del total de las particiones conocidas (Alto, Medio y Bajo).

$$\begin{aligned} Ganancia(S, Salario) &= E(S) - E(S, Salario) \\ &= \frac{13}{14}(0.7793 - 0.5853) \\ &= 0.180 \end{aligned}$$

Para calcular la división de la información (Ecuación 2.13) se toma en cuenta la categoría del valor desconocido para el atributo Salario.

$$\begin{aligned} I\_division(S, Salario) &= -\frac{4}{14}\log_2\frac{4}{14} - \frac{4}{14}\log_2\frac{4}{14} - \frac{5}{14}\log_2\frac{5}{14} - \frac{1}{14}\log_2\frac{1}{14} \\ &= 1.835 \end{aligned}$$

Y finalmente, se calcula la proporción de ganancia (Ecuación 2.12).

$$\begin{aligned} Proporción\_de\_ganancia(S, Salario) &= \frac{Ganancia(S, Salario)}{I\_division(S, Salario)} \\ &= \frac{0.180}{1.835} \\ &= 0.098 \end{aligned}$$

De la misma manera se calcula la ganancia y la proporción de ganancia para el atributo Edad (Tabla 2.22) que contiene las particiones “Joven” y “Mayor”.

Edad			
¿Buen clientes	Desconocido	Joven	Mayor
Si	0	4	6
No	0	3	1
Total	0	7	7

Tabla 2.22: Particiones del atributo edad (C4.5).

Debido a que este atributo no contiene particiones desconocidas (?), se vuelve a calcular la entropía (Ecuación 2.10) de ¿Buen Cliente? (S). Obteniendo los siguientes resultados:

$$E(S) = -\frac{10}{14}\log_2\frac{10}{14} - \frac{4}{14}\log_2\frac{4}{14} = 0.8631$$

$$\begin{aligned} E(S, Edad) &= P(Joven)E(Joven) + P(Mayor)E(Mayor) \\ &= \frac{7}{14}(-\frac{4}{7}\log_2\frac{4}{7} - \frac{3}{7}\log_2\frac{3}{7}) + \frac{7}{14}(-\frac{6}{7}\log_2\frac{6}{7} - \frac{1}{7}\log_2\frac{1}{7}) \\ &= 0.7885 \end{aligned}$$

$$\begin{aligned} Ganancia(S, Edad) &= E(S) - E(S, Edad) \\ &= 0.8631 - 0.7885 \\ &= 0.0747 \end{aligned}$$

$$\begin{aligned} I\_division(S, Edad) &= P(Joven)\log_2 P(Joven) - P(Mayor)\log_2 P(Mayor) \\ &= -\frac{7}{14}\log_2\frac{7}{14} - \frac{7}{14}\log_2\frac{7}{14} \\ &= 1 \end{aligned}$$

$$\begin{aligned} Proporcion\_de\_ganancia(S, Edad) &= \frac{Ganancia(S, Edad)}{I\_division(S, Edad)} \\ &= \frac{0.0747}{1} \\ &= 0.0747 \end{aligned}$$

Y por último se analiza el atributo Hijos (Tabla 2.23) con las particiones “Si” y “No”. Se tienen los siguientes resultados:

Hijos			
¿Buen clientes	Desconocido	Si	No
Si	0	6	2
No	0	4	2
Total	0	10	4

Tabla 2.23: Particiones del atributo hijos (C4.5).

$$E(S) = -\frac{10}{14}\log_2\frac{10}{14} - \frac{4}{14}\log_2\frac{4}{14} = 0.8631$$

$$\begin{aligned} E(S, Hijos) &= P(Si)E(Si) + P(No)E(No) \\ &= \frac{8}{14}(-\frac{2}{8}\log_2\frac{2}{8} - \frac{6}{8}\log_2\frac{6}{8}) + \frac{6}{14}(-\frac{2}{6}\log_2\frac{2}{6} - \frac{4}{6}\log_2\frac{4}{6}) \\ &= 0.8571 \end{aligned}$$

$$\begin{aligned} Ganancia(S, Hijos) &= E(S) - E(S, Hijos) \\ &= 0.8631 - 0.8571 \\ &= 0.0060 \end{aligned}$$

$$\begin{aligned} I\_division(S, Hijos) &= P(Si)\log_2 P(Si) - P(No)\log_2 P(No) \\ &= -\frac{8}{14}\log_2\frac{8}{14} - \frac{6}{14}\log_2\frac{6}{14} \\ &= 0.9852 \end{aligned}$$

$$\begin{aligned} Proporcion\_de\_ganancia(S, Hijos) &= \frac{Ganancia(S, Hijos)}{I\_division(S, Hijos)} \\ &= \frac{0.0060}{0.9852} \\ &= 0.0061 \end{aligned}$$

Una vez aplicada la fórmula de proporción de ganancia a cada uno de los atributos se analizan los resultados comparándolos con la ganancia, debido a que la selección del mejor atributo se hace al que ofrece una mejora es decir, se elige el atributo en que su ganancia y proporción de ganancia sean diferentes.

Atributo	Ganancia	Proporción de ganancia
Salario	0.18	0.09
Edad	0.07	0.07
Hijos	0.006	0.006

Tabla 2.24: Comparación de ganancia y proporción de ganancia.

El atributo Salario es el que se elige como raíz del árbol con tres ramas: Alto, Medio y Bajo. Para obtener los nodos que corresponden a cada rama se realiza una segunda iteración.

SEGUNDA ITERACIÓN. Se analiza la partición del valor Medio del atributo Salario añadiendo el valor desconocido (?) como se muestra en la Tabla 2.25. En esta iteración se agrega el atributo peso con el fin de contabilizar el número de casos para analizar las particiones, agregando la proporción del número de tuplas de salario Medio (4) entre del número total de tuplas con particiones conocidas (13).

Salario	Edad	Hijos	¿Buen cliente?	Peso
?	Joven	Si	No	4/13
Medio	Joven	Si	Si	1
Medio	Mayor	No	Si	1
Medio	Joven	No	Si	1
Medio	Mayor	Si	Si	1

Tabla 2.25: Rama de salario medio (C45).

La distribución de los datos para el atributo Edad se realiza a partir de la suma de los pesos de la Tabla 2.25. Es decir, para el caso que corresponde a:

- No es Buen cliente y es joven= $\frac{4}{13} = 0.3$
- Es buen cliente y es joven= $(1 + 1) = 2$

Edad			
¿Buen clientes	Desconocido	Joven	Mayor
Si	0	2	2
No	0	4/13	0
Total	0	2.3	2

Tabla 2.26: Atributo edad para la rama de salario medio (C4.5).

Con los datos de la Tabla 2.26 se obtiene la ganancia y la proporción de ganancia para el atributo Edad, siguiendo el mismo procedimiento de la primera iteración.

$Ganancia(S, Edad) = 0.068$
$Proporcion\_de\_ganancia(S, Edad) = 0.068$

Ahora se obtienen los datos para el atributo Hijos (Tabla 2.27) nuevamente contabilizando la combinación de casos para la construcción de la tabla.

Hijos			
¿Buen clientes	Desconocido	Si	No
Si	0	0	2
No	0	2	4/13
Total	0	2	2.3

Tabla 2.27: Atributo hijos para la rama de salario medio (C4.5).

$Ganancia(S, Hijos) = 0.69$
$Proporcion\_de\_ganancia(S, Hijos) = 0.69$

Se comparan los resultados de ganancia y proporción de ganancia del atributo Edad e Hijos (Tabla 2.28).

Atributo	Ganancia	Proporción de ganancia
Edad	0.068	0.068
Hijos	0.069	0.069

Tabla 2.28: Comparación de ganancia y proporción de ganancia 2 iteración.

Se concluye que estas particiones no ofrecen ninguna mejora por lo que no se necesita otra iteración. De esta manera se elige el atributo que tiene el mayor número de casos del atributo "¿Buen cliente?" de la Tabla 2.25. Por lo tanto, "Si" se convierte en un nodo terminal de la rama de salario medio.

El procedimiento se repite hasta encontrar los nodos terminales. Después de analizar todos los atributos, la construcción del modelo de árbol de decisión queda de la siguiente manera:

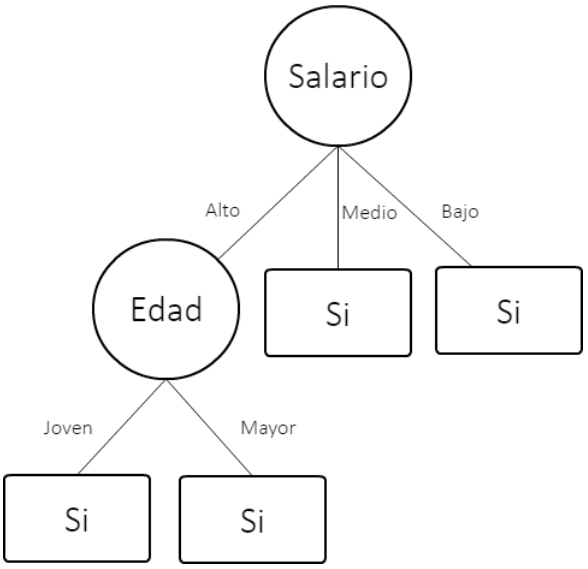


FIGURA 2.10: Árbol de decisión con el algoritmo C4.5.

2.3.4. Ventajas y desventajas de los árboles de decisión.

A continuación se describen algunas ventajas y desventajas de los árboles de decisión [10][42].

Ventajas	**Desventajas**
•La interpretación del modelo de aprendizaje es fácil, ya que se identifica inmediatamente el orden de las condiciones y acciones en cada rama del árbol. •Capaz de manejar datos numéricos y categóricos. •Todas las soluciones del problema son visualizadas. •Información clara para la toma de decisiones.	•Los arboles con múltiples ramas dificultan la toma de decisiones. •Alto costo computacional •Los nodos terminales dan poca cobertura a ejemplos de entrenamiento produciendo estimaciones poco fiables. •No es apropiado para problemas de procesamiento de lenguaje natural.

2.4. Naïve Bayes

El algoritmo de Naïve Bayes clasifica los datos de prueba $x = (x_1, x_2, .., x_m)$ asignándole la clase k que maximiza la probabilidad condicional de la clase dada la secuencia observada en los atributos de los datos de entrenamiento. Es decir,

$$arg\,\max_k P(k|x_1, x_2, ..., x_m) = arg\,\max_k \frac{P(x_1, x_2, ..., x_m|k)P(k)}{P(x_1, x_2, ..., x_m)} \approx arg\,\max_k P(k) \prod_{i=1}^{m} P(x_i|k) \tag{2.14}$$

donde $P(k)$ y $P(x_i|k)$ se estiman a partir de los datos de entrenamiento, utilizando las frecuencias relativas (estimación de la máxima verosimilitud[1]).[2]

2.4.1. Algoritmo de aprendizaje

El algoritmo de Naïve Bayes se caracteriza por dividir su algoritmo en dos etapas. La primera etapa es la construcción del modelo de aprendizaje utilizando los datos de entrenamiento, mientras que la segunda etapa se clasifican los datos de prueba con el modelo creado.

Etapa de entrenamiento

Los datos de entrenamiento se utilizan para la construcción del modelo de aprendizaje. Esta etapa consta de dos pasos que se basan en el Teorema de Bayes[2].

1. Calcular la probabilidad para cada una de las clases.

$$P(k_i) = \frac{N(k_i)}{m} \tag{2.15}$$

donde:
k_i : clase
m : número de tuplas
$N(k_i)$: número de tuplas que corresponden a la clase k_i

[1] Máxima verosimilitud. Se selecciona el valor del parámetro para el cuál la probabilidad sea máxima.
[2] Probabilidad condicional: la probabilidad de que un evento A ocurra dado que ha ocurrido un evento B.

2. Calcular la probabilidad para todos los valores de cada atributo dado la clase k

$$P(x_i|k) = \frac{N(x_i|k_i)}{N(k_i)} \tag{2.16}$$

donde:
x_i : valor del atributo i.
k_i : clase

Etapa de clasificación

Al obtener el modelo final se pasa a la segunda fase. El modelo se verifica con datos de prueba aplicando la fórmula de Naïve Bayes para cada una de las clases. Finalmente, el algoritmo de Naïve Bayes clasifica al dato de prueba con el criterio de seleccionar la clase que maximiza la fórmula de las probabilidades. Los pasos se detallan a continuación:

1. Aplicar la fórmula 2.17. El proceso consiste en multiplicar las probabilidades entre sí para cada una de las clases.

$$arg\,\max_{k} P(k) \prod_{i=1}^{m} P(x_i|k) \tag{2.17}$$

2. Elegir la clase con el mayor valor.

2.4.2. Ejemplo de aplicación

La gran cantidad de correos electrónicos basura (SPAM) que entran en las bandejas de los usuarios han pasado de ser solo una molestia a comprometer su seguridad debido a que el SPAM ha llegado a ser el principal medio para el robo de identidad, así como la proliferación de malwares. Para solucionar este problema se ha decidido hacer un filtro antispam utilizando un conjunto de correos electrónicos (Tabla 2.29), identificando las palabras que estos contengan para clasificarlos como SPAM o NOT SPAM.

Clase	Palabra 1	Palabra 2	Palabra 3	Palabra 4
Not Spam	Sugerencia	Disfruta	Último acceso	Revisa
Spam	Sugerencia	Aprovecha	Mensaje nuevo	Revisa
Spam	Compra	Bienvenido	Mensaje nuevo	Urgente
Not Spam	Sugerencia	Bienvenido	Último acceso	Urgente
Spam	Sugerencia	Aprovecha	Mensaje nuevo	Urgente
Spam	Compra	Bienvenido	Mensaje nuevo	Urgente
Not Spam	Sugerencia	Bienvenido	Último acceso	Conoce

Tabla 2.29: Conjunto de datos de entrenamiento. Correos electrónicos.

Etapa de entrenamiento

Se empieza la etapa de entrenamiento calculando la probabilidad de cada clase: Spam y Not Spam.

$$P(Not\ Spam) = \frac{3}{7} = 0.43$$
$$P(Spam) = \frac{4}{7} = 0.57$$

Para finalizar el entrenamiento calculamos $P(x_i|k)$ para todos los valores de cada atributo observando la Tabla 2.29. Al asumir la independencia entre los sucesos, la probabilidad se calcula tomando el número de veces que aparece la palabra "Sugerencia" dado que es "Not Spam" entre el número de veces que aparece el valor "Not Spam" en los datos de entrenamiento.

$$P(Palabra : Sugerencia|Not\ Spam) = \frac{3}{3} = 1$$

El mismo proceso se emplea para obtener la probabilidad de la palabra "Sugerencia" dado que pertenece a la clase "Spam".

$$P(Palabra : Sugerencia|Spam) = \frac{2}{4} = 0.5$$

La tabla 2.30 muestra los resultados de cada atributo y de cada clase.

Atributo-valor	Not spam	Spam
Palabra 1: Sugerencia	1	0.5
Palabra 1: Compra	0	0.5
Palabra 2: Disfruta	0.33	0
Palabra 2: Aprovecha	0	0.5
Palabra 2: Bienvenido	0.67	0.5
Palabra 3: Último acceso	1	0
Palabra 3: Mensaje nuevo	0	1
Palabra 4: Revisa	0.33	0.25
Palabra 4: Urgente	0.33	0.75
Palabra 4: Conoce	0.33	0

TABLA 2.30: Valores de $P(x_i|k)$

Etapa de clasificación

Se suministra un caso de prueba (Tabla 2.31).

Clase	Palabra 1	Palabra 2	Palabra 3	Palabra 4
Spam	Sugerencia	Aprovecha	Mensaje nuevo	Urgente

Tabla 2.31: Caso de prueba (test)

Aplicando la Ecuación 2.17 y con base en los resultados de la Tabla 2.30 se calcula la probabilidad de que el caso de prueba (Tabla 2.31) sea "Spam". Multiplicando entre sí las probabilidades: $P(spam)$, $P(Palabra\ 1{-}sugerencia|Spam)$, $P(Palabra2{-}aprovecha|Spam)$, $P(Palabra3 - mensajenuevo|Spam)$, $P(Palabra4 - urgente|Spam)$, se obtiene:

$$argmaxP(spam|sugerencia, aprovecha, mensaje\ nuevo, urgente) = (0.57)(0.5)(0.5)(1)(0.75)$$
$$= 0.10$$

Se repite este último procedimiento, pero ahora calculando la probabilidad de que este caso sea "Not Spam"

$$argmaxP(Not\ spam|sugerencia, aprovecha, mensaje\ nuevo, urgente) = (0.43)(1)(0)(0)(0.75)$$
$$= 0$$

En conclusión, la clase con mayor valor es "Spam", esto quiere decir que el método clasificó correctamente este caso de prueba en base al conjunto de datos de entrenamiento.

2.4.3. Ventajas y desventajas del algoritmo Naïve Bayes

A continuación se describen algunas ventajas y desventajas de Naïve Bayes [10][14].

Ventajas	**Desventajas**
•Fácil de implementar. •Converge de manera rápida •Alta precisión en sus predicciones •Manejo de datos discretos y continuos.	•Asume la independencia de los atributos.

2.5. Redes Neuronales Artificiales (RNA)

2.5.1. Las neuronas y el cerebro

Las redes neuronales artificiales fueron desarrolladas como una simulación de las neuronas o de las redes de neuronas en el cerebro.

Se calcula que en cada cerebro existen alrededor de 100,000 millones de neuronas, conectadas cada una de ellas con alrededor de 10,000, es decir que cada actividad neuronal afecta a otras 10,000 neuronas, la cual forma una red de un tamaño enorme [3].

2.5.2. Estructura de una neurona biológica

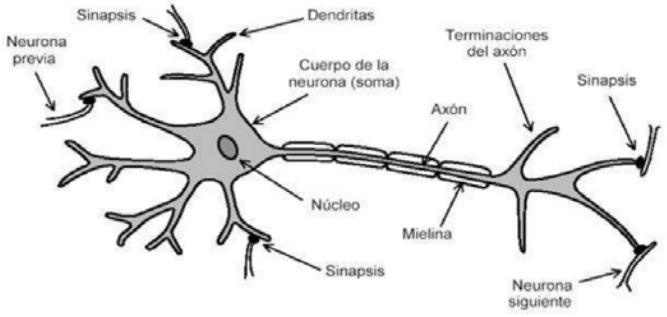


FIGURA 2.11: Fisiología de una neurona biológica, obtenida de https://tinyurl.com/y8hb3gh7

Una neurona biológica tiene los siguientes elementos (Figura 2.11):

1. Un cuerpo celular.
2. Cables de entrada llamados dendritas, son órganos de recepción y propagan la señal al interior de la neurona.
3. Un núcleo, encargado de procesar las señales.
4. Un cable de salida llamado axón, se utiliza para enviar señales o mensajes a otras neuronas a través de la propagación de impulsos electro-químicos.
5. La sinapsis son los elementos de unión entre axón y dendritas, permite a la información ser transmitida desde una neurona a la próxima.

[3]Pedro Isasi Viñuela, Inés M. Galván León. Redes de neuronas artificiales. Un enfoque práctico. Madrid, 2004. Pearson Educación. Página 4.

2.5.3. Historia de las Redes Neuronales Artificiales

1936. Alan Turing fue el primero en estudiar el cerebro humano de manera computacional.

1943. El neurólogo Warren McCulloch y el matemático Walter Pitts fueron los pioneros en la construcción de modelos matemáticos que imitan el comportamiento de las neuronas biológicas. Realizaron el primer modelo matemático de Redes Neuronales Artificiales (el modelo McCulloch-Pitts) basado en la idea de que las neuronas operan mediante impulsos binarios (0,1).

1949. Donald Hebb desarrolló un procedimiento matemático de aprendizaje (aprendizaje Hebbiano), este aprendizaje se convierte en el antecesor de las técnicas modernas de entrenamiento de las Redes Neuronales Artificiales.

1950. Varios investigadores entre ellos Holland, Haibt y Duda combinaron los resultados obtenidos por los matemáticos, biólogos y psicólogos para desarrollar modelos de simulación en computadora de neuronas y redes neuronales dando lugar a la forma actualmente más generalizada de trabajar con estos sistemas.

1951. Marvin Minsky obtuvo los primeros resultados prácticos en Redes Neuronales Artificiales al construir el primer sistema electrónico de aprendizaje llamado SNARC.

1959. Frank Rosenblatt generalizó el modelo de McCulloch- Pitts añadiéndole aprendizaje; llamo a este modelo Perceptrón que fue la primera red neuronal artificial.

1961. Bernard Widrow y Hoff desarrollaron el modelo ADALINE (Adaptive Linear Element) esta fue la primer red neuronal aplicada a un problema real que utiliza una neurona similar a la del perceptrón, pero de respuesta lineal, cuyas entradas pueden ser continuas.

1964. Stephen Grossberg realizó importantes estudios sobre procesos y fenómenos psicológicos y biológicos de procesamiento humano de la información e intentó juntar los dos (mente y cerebro) en una teoría unificada.

1968. James Anderson realizó un modelo de memoria asociativa lineal, creando un nuevo modelo llamado Brain-state-in-a-box (BSB).

1969. Minsky y Pappert publicaron el libro Perceptrons donde se hacían patentes las limitaciones de la simulación de las compuertas lógicas de tipo XOR con perceptrones (redes neuronales).

1971. Teuvo Kohonen quien interesado en comprender la clasificación natural que hace el cerebro, ideó el algoritmo Mapas auto-organizativos (Self-Organizing Map, SOM).

1984. Terence Sejnowski contribuyo con la primera Red Neuronal Artificial que reconocía un algoritmo de aprendizaje para una red de tres niveles, así como en la aplicación del algoritmo de Retropropagación (Backpropagation) en el reconocimiento de voz.

2.5.4. Neurona Artificial

La neurona artificial es una unidad procesadora de información con cuatro elementos funcionales: receptor, sumador, función activadora y salida, los cuales son descritos a continuación.(Figura.2.12)

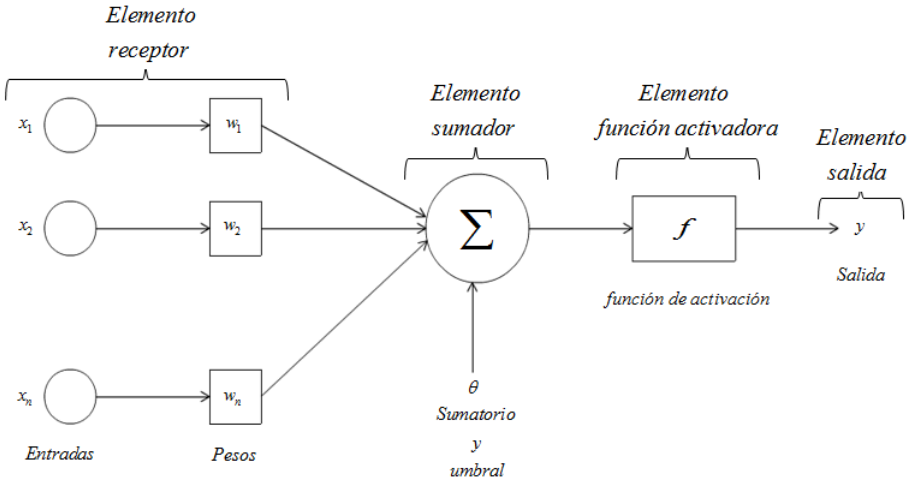


FIGURA 2.12: Elementos funcionales de una neurona artificial.

Elemento receptor: Un grupo de entradas $x_1, x_2, ..., x_n$ son introducidas en una neurona artificial que corresponden a las señales de sinapsis de una neurona biológica, definida por un vector X. Cada señal se multiplica por un peso asociado $w_1, w_2, ..., w_n$, cada peso corresponde a la fuerza de una conexión sináptica y es representado por un vector W.

Elemento sumador: Efectúa la suma algebraica ponderada de las señales de entrada de acuerdo con su peso. Esta suma representa el cuerpo de la neurona y produce una salida denominada S.

$$S = w_1x_1 + w_2x_2 + ... + w_nx_n$$

$$S = \sum_{i=1}^{n} w_ix_i$$

De manera vectorial se define como:

$$S = WX^T$$

Elemento de función activadora: Aplica una función no lineal a la salida de la suma S para decidir si la neurona se activa o no se activa, cuyo rango va desde (0 a 1) o de (-1 a 1). La neurona se activará si el resultado es superior a un determinado límite o umbral. Esto significa que se enviará una señal (en forma de una onda ionizada) a lo largo de su información, pasando de una parte de la red de neuronas a otra.

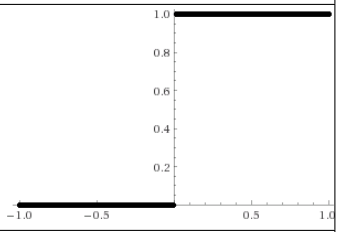
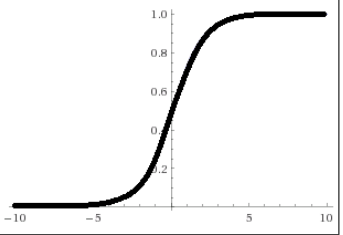
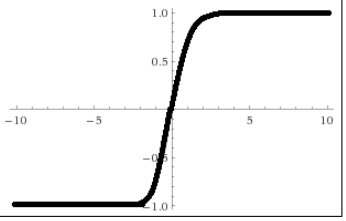
Función de activación			
	Función	Rango	Gráfica
Escalón	$y = \begin{cases} 1 & Si \quad x \geqslant 0 \\ -1 & Si \quad x < 0 \end{cases}$	$[-1, 1]$	
Sigmoidal	$y = \frac{1}{1+e^{-x}}$	[0,1]	
Tangente Hiperbólica	$y = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$	[-1,1]	

Tabla 2.32: Funciones de activación

Elemento de salida: Produce la señal de acuerdo a la función activadora y constituye la salida de una neurona artificial.

$$y = f(\sum_{i=1}^{n} w_i x_i) = f(w, x) = f(w^T x)$$

Donde f es la función de activación.

2.5.5. Redes neuronales artificiales (RNA)

Una red neuronal es una colección de neuronas, todas con las mismas escalas de tiempos, donde sus salidas están conectadas a las entradas de otras neuronas [McCulloch-Pitts,1943].

Las redes neuronales artificiales son redes interconectadas masivamente en paralelo de elementos simples y con organización jerárquica, las cuales intentan interactuar con los objetos del mundo real del mismo modo que lo hace el sistema nervioso biológico [Kohonen, 1988].

2.5.6. Estructura básica de RNA

En una RNA existe una capa de entrada con n neuronas y una capa de salida con m neuronas. A la manera en que las neuronas se conectan entre si se le denomina arquitectura de red, las cuales son: monocapa (no tiene capa oculta) o multicapa (tiene una o más capas ocultas).

En la Figura 2.13 se presenta una red neuronal en la que su capa de entrada tiene tres neuronas (1, 2 y 3) y una capa de salida con una neurona (4).

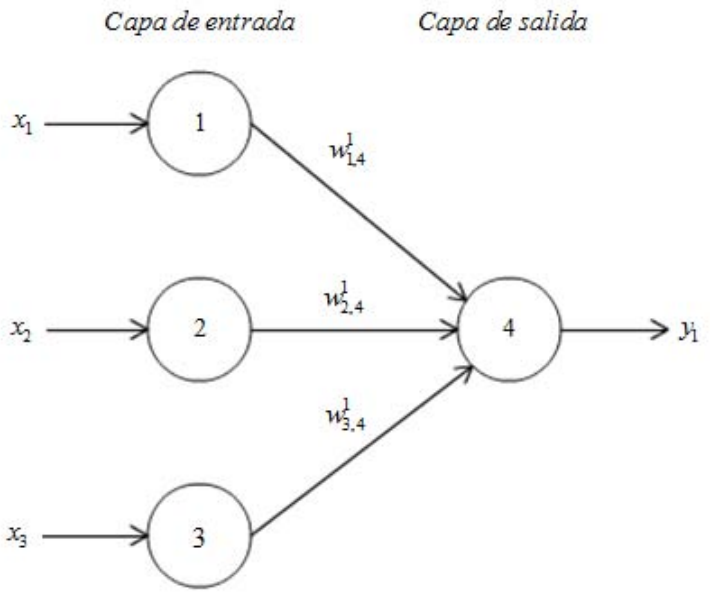


FIGURA 2.13: Diagrama de una RNA monocapa.

En la Figura 2.14 se muestra una red neuronal que en su capa de entrada hay tres neuronas (1, 2 y 3), una capa oculta que igual contiene tres neuronas (4, 5 y 6) y una capa de salida que consta de una neurona (7).

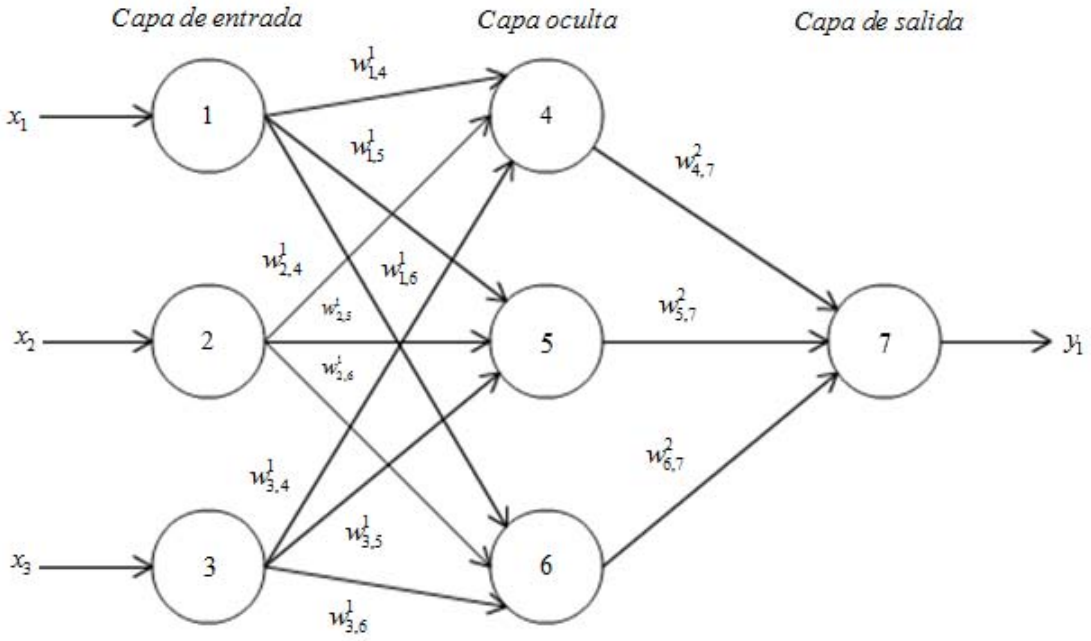


FIGURA 2.14: Diagrama de una RNA multicapa.

Por lo tanto, a partir de la distribución de las neuronas en la arquitectura de red, se distinguen tres tipos de capas:

Entrada: Capa que recibe directamente la información proveniente de fuentes externas de la red o de otras neuronas de la capa anterior. Este primer nivel lo constituyen las células de entrada y reciben valores representados en forma de vectores.

Oculta: Llamadas intermedias, son capas internas a la red por lo que no tienen contacto con el exterior. Las neuronas de la capa oculta pueden estar conectadas a otra capa oculta o a las neuronas de la capa de salida.

Salida: Proporciona el resultado de estas unidades que sirve como salida total de la red.

No hay conexiones hacia atrás ni laterales entre neuronas de la misma capa. Cada interconexión entre unidades de proceso actúa como una ruta de comunicación, donde viajan los valores numéricos de una célula a otra.Estos valores son evaluados por los pesos de las conexiones y son los que se ajustan durante el proceso de aprendizaje para producir una red neuronal artificial final.

2.5.7. Aprendizaje supervisado en Redes Neuronales Artificiales.

El aprendizaje de una red neuronal artificial consiste en modificar sus pesos w_i a partir del conjunto de entrenamiento $(x_i, d(x_i))$ es decir, al introducir a la RNA el elemento

receptor x_i los pesos se ajustaran en función de cuán parecida es la salida producida por la RNA(y_i) a la salida deseada ($d(x_i)$).

2.5.7.1. Perceptrón simple

El perceptrón simple tiene una estructura monocapa, por lo tanto no tiene capa oculta, en la que hay varias neuronas de entrada y una o más de salida [6][12].

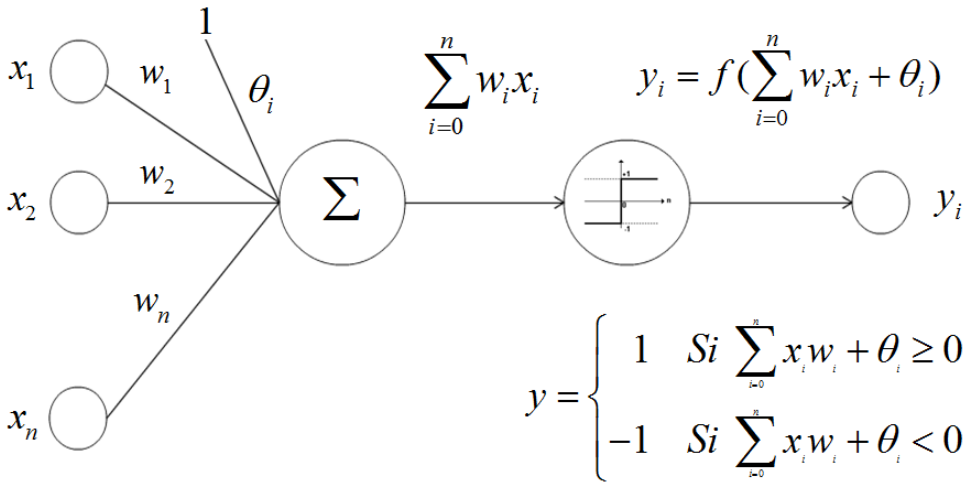


FIGURA 2.15: Diagrama de un perceptrón simple.

En la Figura 2.15 las entradas de la neurona son $x_1, x_2, ..., x_n$ y la salida y . Los pesos son $w_1, w_2, ..., w_n$. Existe un parámetro adicional llamado umbral que está denotado por θ. El umbral se utiliza como factor de comparación para producir la salida y habrá uno por cada neurona de salida.

La salida de la red se obtiene calculando la activación de la neurona mediante la suma ponderada por los pesos de todas las entradas.

$$y = \sum_{i=1}^{n} w_i x_i$$

Pasa el resultado a una función de salida al nivel de activación de la neurona. La función de activación para el perceptrón es de tipo escalón (Tabla 2.32).

La regla de decisión es responder 1 si el patrón presentado pertenece a la clase A o -1 si el patrón pertenece a la clase B.

La ecuación de un perceptrón simple se escribe

$$y = f(\sum_{i=1}^{n} w_i x_i + \theta) \tag{2.18}$$

En el caso de un perceptrón con dos entradas x_1 y x_2 la Ecuación 2.18 se transforma en:

$$w_1 x_1 + w_2 x_2 + \theta = 0$$

Que es la ecuación general de la recta, con pendiente $-\frac{w_1}{w_2}$ y ordenada $-\frac{\theta}{w_1}$.

La red define una recta en la cual clasifica los datos en dos categorías A o B. La Figura 2.16 muestra como esta recta separa los datos mediante un perceptrón.

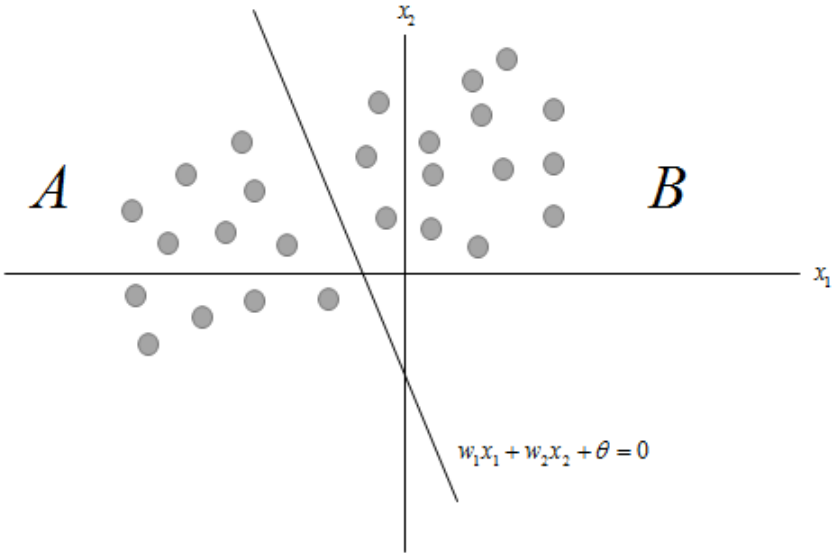


FIGURA 2.16: Separación de dos clases mediante el perceptrón.

2.5.7.2. Algoritmo de aprendizaje del perceptrón.

1. Se asignan valores aleatorios a cada uno de los pesos (w_i) y el umbral (θ). Los cuales poseen valores entre -1 y 1.

2. Presentar un vector de entrada $x = x_1, x_2, ..., x_n$ a partir del conjunto de ejemplos de entrenamiento.

3. Se propaga la activación hacia adelante a través de los pesos de la red para calcular la salida aplicando la función escalón.

$$y = f(\sum_{i=1}^{n} w_i x_i + \theta)$$

4. Si $y \neq d(x)$ la red da una respuesta incorrecta. Modificar w_i y θ de acuerdo con:

$$w_i(nuevo) = w_i(anterior) + d(x)x_i \tag{2.19}$$

$$\theta(nuevo) = \theta(anterior) + d(x) \tag{2.20}$$

   Donde $d(x)$ es la salida deseada.

5. Si los pesos no se han cambiado $w_i(nuevo) = w_i(anterior)$ entonces el proceso finaliza. En otro caso volver al paso 3.

2.5.7.3. Perceptrón multicapa

El perceptrón multicapa es una generalización del perceptrón simple, que surgió como solución a problemas de separabilidad no lineal (Figura 2.17). Hoy en día es posible mostrar que muchos conjuntos de datos son modelados mediante el empleo del perceptrón multicapa debido a la habilidad de aprender a partir de un conjunto de ejemplos, aproximar relaciones no lineales y filtrar el ruido en los datos , por lo que es considerada una de las arquitecturas más utilizadas en la solución de problemas reales [6][12].

La principal diferencia entre el perceptrón simple y el perceptrón multicapa es que el perceptrón simple solo se puede aplicar a casos linealmente separables [Minsky & Papert,1969]. En el caso de problemas no linealmente separables se emplea el perceptrón multicapa.

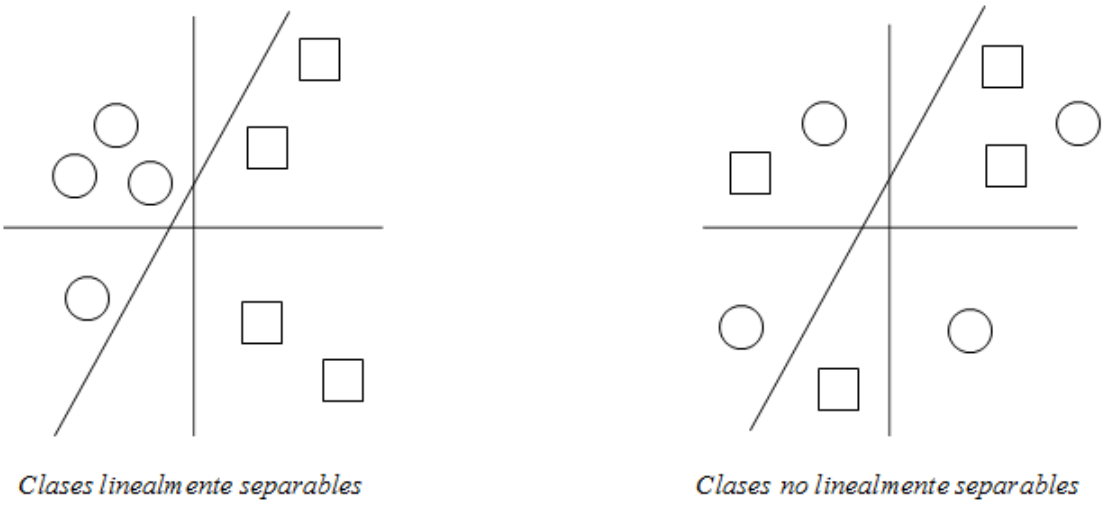


FIGURA 2.17: (Izquierda) Clases linealmente separables. (Derecha) Clases no linealmente separables.

El perceptrón multicapa es una red neuronal en forma de cascada con una o más capas ocultas.

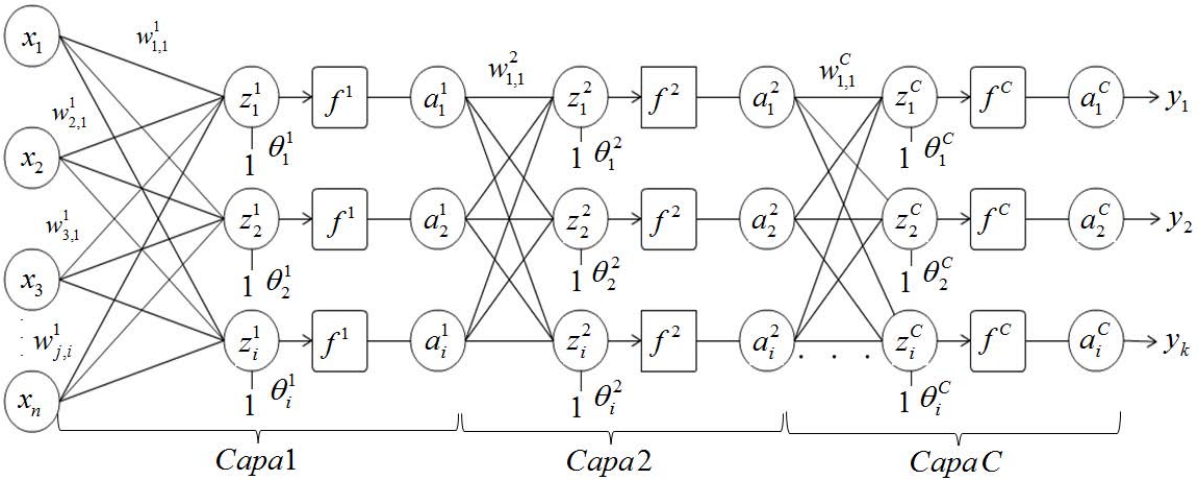


FIGURA 2.18: Diagrama de un perceptrón multicapa.

Donde:

x vector de entrada a partir de un conjunto de datos de entrenamiento $x = (x_1, x_2, ..., x_n)$

w_{ij}^c peso de la conexión de la neurona i de la capa c a la neurona j.

z_i^c suma ponderada de las entradas.

θ_i^c umbral de la neurona i en la capa c.

f función de activación $f = (f^1, f^2, ..., f^C)$.

a_i^c activación de la neurona i en la capa c.

y vector de salida de la red $y = (y_1, y_2, ..., y_k)$

En la Figura 2.18 se encuentra una red neuronal con x_n neuronas de entrada, i neuronas en cada capa oculta c y finalmente la capa de salida y_k , donde en cada capa oculta se calcula la suma ponderada de las entradas z_i^c y se aplica la función de activación a_i^c hasta llegar a la capa de salida donde el valor de salida que sea lo más próxima a la salida deseada.

2.5.7.4. Algoritmo de retropropagación (Backpropagation)

El algoritmo de retropropagación tiene dos fases:

- Hacia adelante: El patrón de entrada es presentado a la red $(x_1, x_2, ..., x_n)$ propagado a través de las capas hasta llegar a la capa de salida.

- Hacia atrás: Se comparan los valores de salida de la red con los valores de salida deseada para obtener el error. Se ajustan los valores de los pesos de la última capa proporcionalmente al error dependiendo de la función de activación usada y finalmente se pasa a la capa anterior con la retropropagación del error ajustando los pesos hasta llegar a minimizar dicho error, continuando este proceso hasta llegar a la primera capa.

La retropropagación está calculando el error de activación que se obtuvo para la neurona en la capa C.

El algoritmo de retropropagación sirve para establecer los parámetros de la red neuronal de un conjunto de entrenamiento. Como se mencionó anteriormente en el aprendizaje de una RNA consiste en adaptar y modificar todos los parámetros de la red para que la salida de la red sea lo más próxima a la salida deseada.

Para un modelo de perceptrón multicapa con un total de C capas, de las cuáles $C-2$ son capas ocultas y n_c el número de neuronas en la capa c, donde $c = 1, 2, 3, ..., C$. Los pasos del algoritmo de retropropagación son los siguientes:

1. Se asignan valores aleatorios a los pesos y umbrales de la red. Generalmente con valores positivos y negativos.

2. Presentar un vector de entrada x=$\{x_1, x_2, ..., x_n\}$ a partir del conjunto de entrenamiento propagando este vector hacia la salida de la red.

3. Activar las neuronas de la capa de entrada

$$a_i^1 = x_i. \tag{2.21}$$

   $i = 1, 2, 3, ..., n_1$ número de neuronas en la capa 1

4. Activar las neuronas de las capas ocultas

$$z_i^c = \sum_{j=1}^{c-1} W_{ji}^{c-1} a_j^{c-1} + \theta_i^c \tag{2.22}$$

   $c = 2, 3, ..., C-1$
   $i = 1, 2, ..., n_c$

   a) Aplicar la función de activación para calcular la señal de salida.

$$a_i^c = f(z_i^c) \tag{2.23}$$

$c = 2, 3, ..., C-1$
$i = 1, 2, ..., n_c$

5. Activar las neuronas de la capa de salida.

$$y_k = \sum_{j=1}^{n_{C-1}} W_{ji}^{C-1} a_j^{C-1} + \theta_i^C \tag{2.24}$$

$i = 1, 2, 3, ..., n_C$

a) Aplicar la función de activación para calcular la señal de salida.

$$a_i^C = f(y_k) \tag{2.25}$$

$i = 1, 2, 3, ..., n_C$

6. Se evalúa el error cuadrático cometido por la red para el patrón n.

$$e(n) = \frac{1}{2} \sum_{k,i=1}^{n} (d_i - y_k)^2 \tag{2.26}$$

Donde
$y(n) = \{y_1, y_2, ..., y_k\}$,vector de salida de la red
$d(n) = \{d_1, d_2, ..., d_i\}$,vector de salida deseada para el patrón n

7. Se aplica la regla delta generalizada para modificar los pesos y umbrales de la red. Para ello se siguen los siguientes pasos:

a) Se calculan los valores δ para todas las neuronas de la capa de salida C.

$$\delta_i^C = -(d_i - y_i) f'(y_k) \tag{2.27}$$

$i = 1, 2, ..., n_C$
f'=derivada de la función de activación.

b) Se calculan los valores δ para el resto de las neuronas de la red empezando desde la última capa oculta y retropropagando dichos valores hacia la capa de entrada.

$$\delta_j^{c+1} = f'(z_j^{c+1}) \sum_{i=1}^{n_{c+2}} \delta_i^{c+2} w_{ji}^{c+1} \tag{2.28}$$

$c = 1, 2, ..., C-2$
$j = 1, 2, ..., n_{c+1}$

8. Se actualizan los pesos, utilizando el algoritmo recursivo (Gradiente de descenso)[4], comenzando por las neuronas de salida y trabajando hacia atrás hasta llegar a la capa de entrada. La variable α representa la velocidad de aprendizaje de la red, normalmente se elige algún valor entre 0.05 a 0.25.

   a) Modificar los pesos y umbrales de las neuronas de la capa de salida.

$$W_{ji}^{C-1}(nuevo) = W_{ji}^{C-1}(anterior) + \alpha\delta_i a_j^{C-1} \tag{2.29}$$

$$\theta_i^C(nuevo) = \theta_i^C(anterior) + \alpha\delta_i \tag{2.30}$$

   $j = 1, 2, ..., n_{C-1}$
   $i = 1, 2, ..., n_C$
   α:Razón de aprendizaje.

   b) Modificar los pesos y umbrales de las capas ocultas.

$$W_{kj}^c(nuevo) = W_{kj}^c(anterior) + \alpha\delta_j^{c+1} a_k^c \tag{2.31}$$

$$\theta_j^{c+1}(nuevo) = \theta_j^{c+1}(anterior) + \alpha\delta_j^{c+1} \tag{2.32}$$

   $c = 1, 2, ..., C-2$
   $k = 1, 2, ..., n_c$
   $j = 1, 2, ..., n_{c+1}$

9. Se repiten los pasos 2,3,4,5,6,7 y 8 para todos los vectores de entrada.

10. Se evalúa el error total cometido por la red.

$$E = \frac{1}{N}\sum_{n=1}^{N} e(n) \tag{2.33}$$

    N: número de vectores de entrada
    $e(n)$: error cometido por la red para el vector de entrada n

[4]Gradiente de descenso. Algoritmo de optimización que consiste en localizar un mínimo en una función de manera iterativa.

2.5.8. Ejemplo XOR

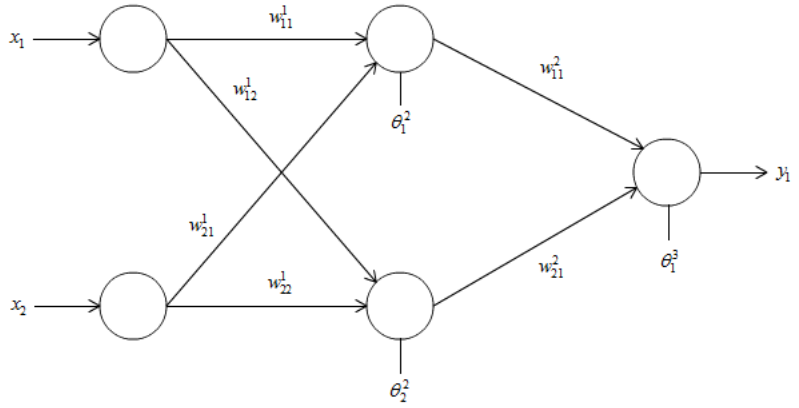


FIGURA 2.19: Ejemplo perceptrón multicapa

Entrada	Salida
(0,0)	0
(0,1)	1
(1,0)	1
(1,1)	0

TABLA 2.33: Conjunto de entrenamiento. Función XOR.

1. Se asignan valores aleatorios a los pesos y umbrales de la red.

$$W_{11}^1 = 5.1911 \qquad W_{22}^1 = 2.7695 \qquad \theta_1^2 = -1.9028$$

$$W_{12}^1 = 2.7586 \qquad W_{11}^2 = 5.8397 \qquad \theta_2^2 = -4.1270$$

$$W_{21}^1 = 5.4730 \qquad W_{21}^2 = -6.1868 \qquad \theta_1^3 = -2.5705$$

2. Se presenta el primer vector de entrada x a partir del conjunto de entrenamiento (Tabla 2.33).

3. Se activan las neuronas de la capa de entrada

$$a_1^1 = 0$$

$$a_2^1 = 0$$

4. Se activan las neuronas de las capas ocultas utilizando la Ecuación 2.22.

$$z_1^2 = W_{11}^1 a_1^1 + W_{21}^1 a_2^1 + \theta_1^2 = -1.9028$$

$$z_2^2 = W_{12}^1 a_1^1 + W_{22}^1 a_2^1 + \theta_2^2 = -4.1270$$

a) Aplicar la función de activación (Ecuación 2.23).

$$a_1^2 = f(z_1^2) = 0.1297$$

$$a_2^2 = f(z_2^2) = 0.0158$$

5. Activar las neuronas de la capa de salida (Ecuación 2.24).

$$y_1 = W_{11}^2 a_1^2 + W_{21}^2 a_2^2 + \theta_1^3 = -1.9108$$

a) Aplicar la función de activación (Ecuación 2.25). Utilizando la función sigmoidea

$$a_1^3 = f(y_1) = 0.1288$$

6. Se evalúa el error cuadrático cometido por la red con la Ecuación 2.26

$$e(n) = \frac{1}{2}[(d_1 - y_1)^2] = 0.0083$$

7. Aplicando la regla delta generalizada para modificar los pesos:

a) Se calcula el valor δ (Ecuación 2.27), para la capa de salida

$$\delta_1^3 = -(d_1 - y_1) f'(y_1) = 0.0262$$

b) Y utilizando la Ecuación 2.28 para calcular el valor δ para la capa oculta.

$$\delta_1^2 = f'(z_1^2)\delta_1^3 w_{11}^2 = 0.7788$$

8. Se actualizan los valores de los pesos y los umbrales asignando a $\alpha = 0.1$

a) Para la neurona de la capa de salida, se siguen las Ecuaciones 2.29 y 2.30

$$w_{11}^2(nuevo) = w_{11}^2(anterior) + 0.1 \cdot \delta_1^3 a_1^2 = 5.8405$$

$$w_{21}^2(nuevo) = w_{21}^2(anterior) + 0.1 \cdot \delta_1^3 a_2^2 = -6.1867$$

$$\theta_1^3(nuevo) = \theta_1^3 + 0.1 \cdot \delta_1^3 = -2.5639$$

b) Por último para la capa oculta con las Ecuaciones 2.31 y 2.32

$$w_{11}^1(nuevo) = w_{11}^1(anterior) + 0.1 \cdot \delta_1^2 a_1^1 = 5.1911$$

$$w_{12}^1(nuevo) = w_{12}^1(anterior) + 0.1 \cdot \delta_2^2 a_1^1 = 2.7586$$

$$w_{21}^1(nuevo) = w_{21}^1(anterior) + 0.1 \cdot \delta_1^2 a_2^1 = 5.4730$$

$$w_{22}^1(nuevo) = w_{22}^1(anterior) + 0.1 \cdot \delta_2^2 a_2^1 = 2.7695$$

$$\theta_1^2(nuevo) = \theta_1^2 + 0.1 \cdot \delta_1^2 = -1.7081$$

$$\theta_2^2(nuevo) = \theta_2^2 + 0.1 \cdot \delta_2^2 = -4.1276$$

9. Se repite el proceso para los vectores de entrada (0,1),(1,0) y (1,1).

10. Finalmente, se evalúa el error total de la red aplicando la Ecuación 2.33

$$E = \frac{1}{4} \sum_{n=1}^{4} e_i(n) = 0.1250$$

El algoritmo finaliza al aproximar el error a cero o cuando al hacer más iteraciones el error no disminuye considerablemente.

2.5.9. Ventajas y desventajas de las Redes Neuronales Artificiales

A continuación se describen algunas ventajas y desventajas de las Redes Neuronales Artificiales [11][12][16].

Ventajas	**Desventajas**
•Permite modelar problemas no lineales. •Se puede aplicar para problemas de clasificación, regresión y agrupamiento. •Las RNA son útiles en diversos campos como: la predicción de mercados financieros, el control de robots y la teledetección. •Capaces de reconocer patrones en datos con ruido o incompletos.	•No es sencillo determinar el número de capas ocultas. •El tiempo de entrenamiento puede ser excesivo. •Algunas personas consideran complicado su entendimiento o uso.

2.6. Máquina de soporte de vectores

La máquina de soporte de vectores es un algoritmo que divide un conjunto de datos en dos clases: positiva y negativa mediante la implementación de un hiperplano con margen máximo [48].

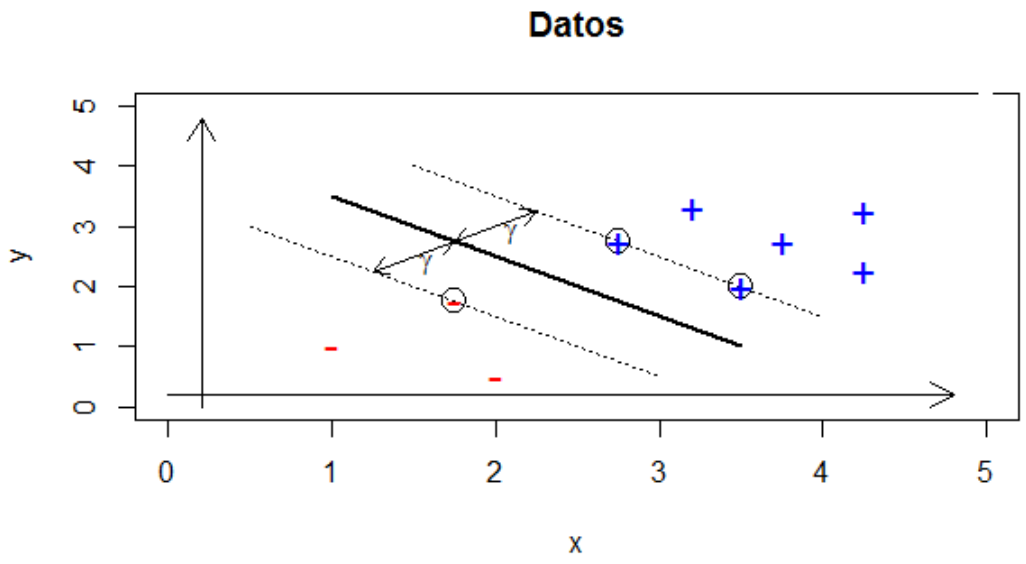


FIGURA 2.20: Ejemplo de la máquina de soporte de vectores

La máquina de soporte de vectores (Figura 2.20) se compone por un hiperplano (línea recta continua), vectores de soporte (denotados por un círculo), que son los datos más cercanos al hiperplano, y un margen (líneas discontinuas) que es la distancia entre el hiperplano y los vectores de soporte.

En este sentido un hiperplano, se define como una figura geométrica que tiene una dimensión menos que el espacio en el que está ubicado. Al ser un clasificador binario su separación se hace con una línea recta en $\mathbb{R}^2$ (Ecuación 2.34) y con un plano en $\mathbb{R}^3$ (Ecuación 2.35).

$$ax + by = d \tag{2.34}$$

$$ax + by + cz = d \tag{2.35}$$

De este modo un hiperplano queda expresado por un vector de pesos w (orientación) y un umbral b (distancia al origen) como se muestra en la Figura 2.21.

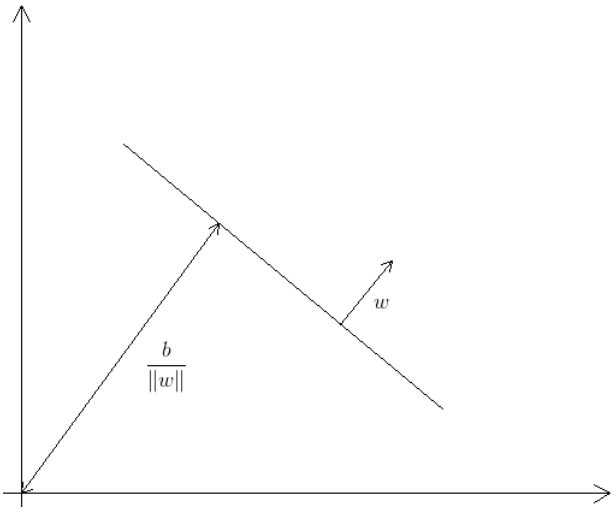


FIGURA 2.21: Elementos de un hiperplano

Por lo tanto la ecuación del hiperplano se expresa como

$$h = w^T x_i + b \tag{2.36}$$

Existen n hiperplanos capaces de clasificar un conjunto de datos (Figura 2.22).

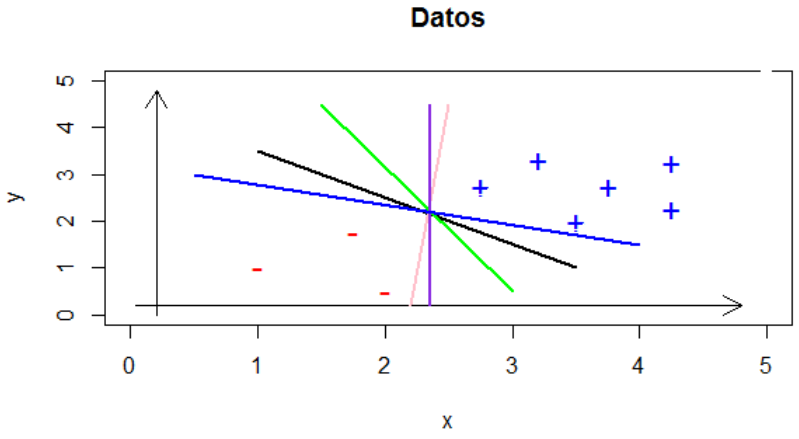


FIGURA 2.22: Ejemplo de hiperplanos posibles

Sin embargo, el hiperplano óptimo es aquel con mayor distancia a todos los datos de las dos clases, es decir, el que tiene mayor margen (γ).

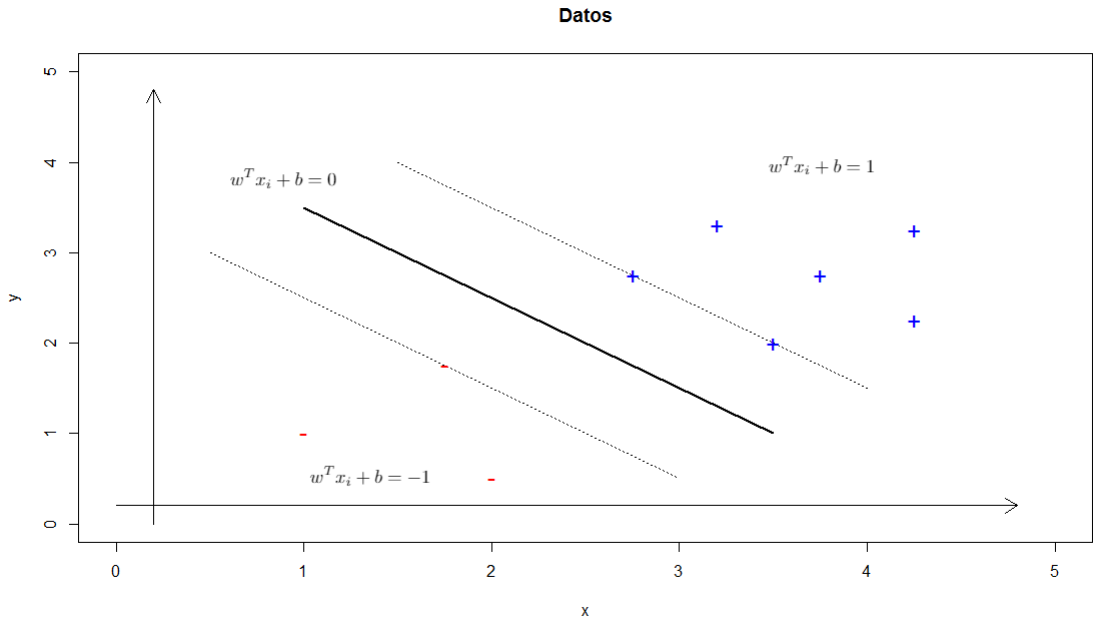


FIGURA 2.23: Conjunto de datos linealmente separable

Al elegir el hiperplano óptimo se crea la regla de clasificación (Ecuación 2.37) asignando la clase $+1$ a los datos de entrenamiento si $h \geq 0$ y la clase -1 si $h < 0$.

$$y = signo(b + \sum_{i=1}^{N} w^T x_i) \tag{2.37}$$

donde

N: número de tuplas

w: vector ortogonal

b: umbral

x: datos de entrenamiento

La solución a la SVM se plantea como un problema de optimización, cuya función objetivo es maximizar el margen.

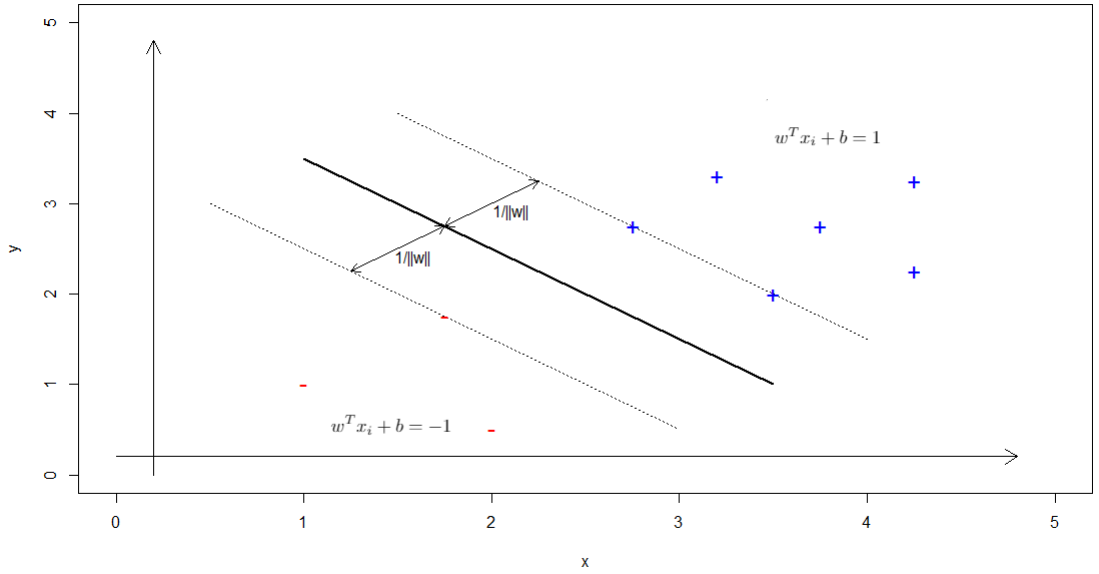


FIGURA 2.24: Maximizar el margen

$$\begin{aligned} Maximizar \quad & \frac{2}{||w||} \\ s.a \quad & y_{i(}w_i x_i + b) \geq 1 \end{aligned} \tag{2.38}$$

El problema se transforma a otro que es matemáticamente equivalente.

$$\begin{aligned} Minimizar \quad & \frac{1}{2}||w||^2 \\ s.a \quad & y_{i(}w_i x_i + b) \geq 1 \end{aligned} \tag{2.39}$$

Al obtener un modelo de optimización con una función cuadrática y que está sujeta a restricciones lineales (Ecuación 2.39), el problema se resuelve por medio de un método de optimización cuadrática realizando los siguientes pasos [49][54]:

1. Construcción del Lagrangiano [5]

$$L(x, \alpha) = f(x) - \sum \alpha_i g_i(x)$$

α: Multiplicadores de Lagrange
$f(x)$: Función a optimizar
$g_i(x)$: Restricciones

[5]Lagrangiano. Función que transforma un problema de optimización con restricciones de igualdad en uno con restricciones simples mediante multiplicadores de Lagrange.

A partir del modelo de optimización de la Ecuación 2.39 se obtiene:

$$L(x, b, w) = \frac{1}{2} w \cdot w - \sum_i \alpha_i (y_i (w_i x_i + b) \geq 1) \tag{2.40}$$

2. Se calculan las primeras derivadas parciales y se igualan a cero para minimizar el Lagrangiano

   a) Con respecto al vector ortogonal

$$\frac{\partial L_{w,x,b}}{\partial w} = 0$$
$$w = \sum_i \alpha_i y_i x_i \tag{2.41}$$

   b) Con respecto al umbral

$$\frac{\partial L_{w,x,b}}{\partial b} = 0$$
$$\sum_i \alpha_i y_i = 0 \qquad \forall \alpha > 0 \tag{2.42}$$

   Los valores de la Ecuación son los llamados vectores de soporte.

3. Se sustituyen las expresiones de las derivadas parciales (Ecuación 2.41 y 2.42) en el Lagrangiano (Ecuación 2.40).

$$L(x, b, w) = \frac{1}{2}(\sum_i \alpha_i y_i x_i \cdot \sum_j \alpha_j y_j x_j) - (\sum_i \alpha_i y_i x_i \cdot \sum_j \alpha_j y_j x_j) - b \sum_i \alpha_i y_i + \sum_i \alpha_i$$

$$L(x, b, w) = \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i \cdot x_j$$

4. A partir del modelo primal (Ecuación 2.39) se obtiene un modelo dual(equivalente) cuya solución son los vectores de soporte.

$$\begin{array}{cc} Maximizar & \sum_i \alpha_i - \frac{1}{2} \sum_i \sum_j \alpha_i \alpha_j y_i y_j x_i \cdot x_j \\ s.a & \sum_i \alpha_i y_i = 0 \qquad \forall \alpha > 0 \end{array} \tag{2.43}$$

La solución del problema da los valores óptimos de α, por lo que el hiperplano se define como la combinación lineal del conjunto de entrenamiento.

$$w = \sum \alpha_i y_i x_i \tag{2.44}$$

$$b = \sum y_i - w \cdot x_i \tag{2.45}$$

Cuya función de clasificación es:

$$y = signo(w \cdot x_i + b)$$

2.6.1. Máquina de soporte de vectores no linealmente separable

Cuando un conjunto de datos no se puede separar por una linea recta ya que datos positivos y negativos están combinados se le conoce como no linealmente separable.

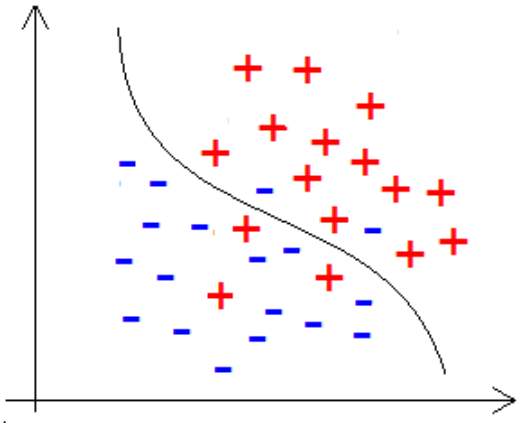


FIGURA 2.25: Conjunto no linealmente separable

Máquina de soporte vectorial aborda dos técnicas para el tratamiento de problemas no linealmente separable, que normalmente son los más habituales en un conjunto de datos reales, las cuáles son [2]:

1. Función Kernel
2. Margen suave o flexible

2.6.2. Función Kernel

La función kernel se utiliza para mapear [6] un espacio de características a una dimensión superior en la que los datos son linealmente separables.

[6]Mapeo o transformación. Se estudia como es transformada una región específica del plano z (que puede ser una recta, un círculo, etc.) en otra región del plano w cuando se aplica $w = f(z)$. Esto con la finalidad de tener una mejor visualización

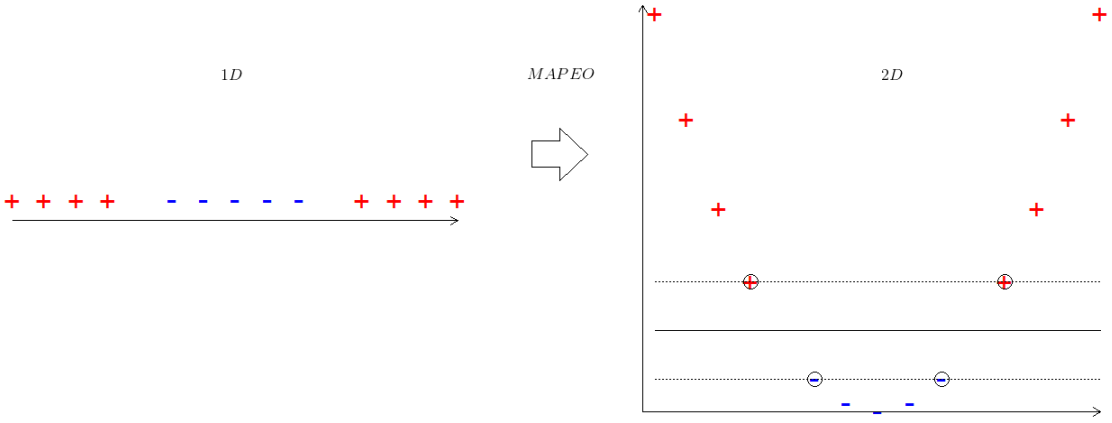


FIGURA 2.26: Mapeo del espacio de características.

Se aprovecha la propiedad de los productos escalares $x_i \cdot x_j$ del modelo dual (Ecuación 2.43) para la introducción de la función kernel (K).

$$\begin{array}{cc} Maximizar & \sum_i \alpha_i - \frac{1}{2}\sum_i \sum_j \alpha_i \alpha_j y_i y_j K(x_i \cdot x_j) \\ s.a & \sum_i \alpha_i y_i = 0 \qquad \forall \alpha > 0 \end{array} \tag{2.46}$$

cuyo clasificador se define como:

$$y = signo(\sum_i y_i \alpha_i K(x_i \cdot x_j) + b) \tag{2.47}$$

A continuación se mencionan dos de los principales tipos de kernel capaces de dividir a un conjunto de datos.

1. Kernels polinómicos

$$K(x_i, x_j) = (x_i x_j + 1)^d \tag{2.48}$$

d:grado del polinomio

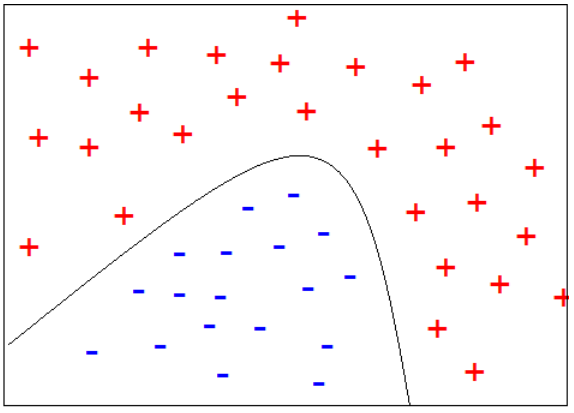


FIGURA 2.27: Kernel polinómico

2. Kernels radiales

$$K(x_i, x_j) = e^{-\gamma||x_i - x_j||^2} \tag{2.49}$$

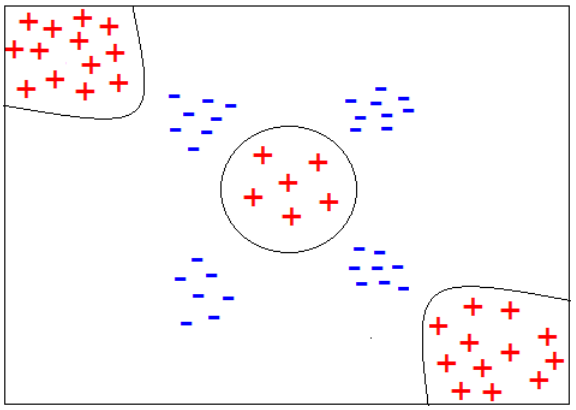


FIGURA 2.28: Kernel radial

2.6.3. Margen suave o flexible

El margen flexible permite tener cierta tolerancia a errores esto implica que algunos datos se encuentren dentro de la zona del margen, es decir, se clasifiquen erróneamente.

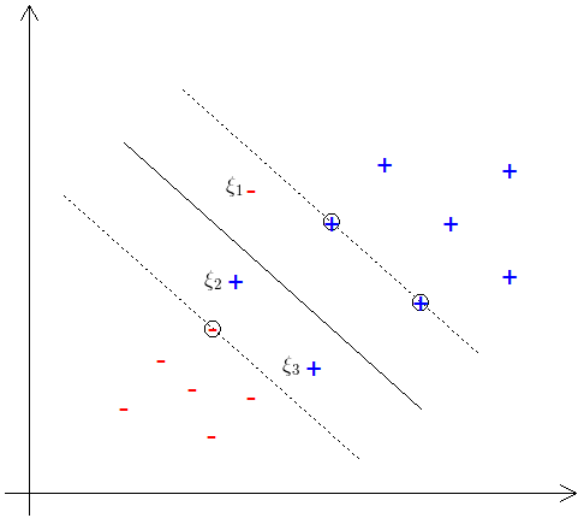


FIGURA 2.29: Margen flexible o suave

Por lo tanto, el modelo de optimización para el caso de margen flexible o suave se implementa a partir de la introducción de variables de holgura (ξ) partiendo del modelo primal (Ecuación 2.39).

$$\begin{array}{cc} Minimizar & \frac{1}{2}\langle w, w\rangle + C\sum_{i=1}^{N}\xi_i \\ s.a & y_{i(}w_i x_i + b) \geq 1 - \xi_i \end{array} \tag{2.50}$$

Donde:
$\xi_i \geq 0$
$1 \leq i \leq N$ N: Total de datos de entrenamiento.

Las variables de holgura ξ_i permiten que los datos estén dentro del margen. El parámetro C regulariza la maximizacion del margen y minimiza los datos dentro del margen. Mientras el valor del parámetro C sea pequeño más flexible sera el margen, este parámetro se escoge normalmente como una potencia de diez ($10^{\pm x}$).

2.6.4. Algoritmo de aprendizaje

El algoritmo de aprendizaje de máquina de soporte vectorial se conforma de una etapa de entrenamiento que consiste en deducir el hiperplano solución al emplear el conjunto de datos ya sea lineal o no lineal. Al obtener dicho hiperplano, se pasa a la etapa de prueba donde se clasifican nuevos registros asignado la clase $+1$ o -1.
Los pasos de las etapas se mencionan a continuación:

1. Construir la función objetivo (Lagrangiano) y las restricciones del modelo de optimización a partir del conjunto de entrenamiento, si el problema es linealmente separable se utiliza la ecuación 2.43 y para el caso no linealmente separable la ecuación 2.46 o 2.50

2. Calcular las primeras derivadas parciales de la función objetivo con respecto a cada α_i, formando un sistema de ecuaciones cuya solución son los valores númericos de α.

3. Obtener el hiperplano solución calculando el vector ortogonal (ecuación 2.44) y el umbral (ecuación 2.45), sustituyendo los valores del conjunto de entrenamiento y de α.

4. Asignar una clase a un registro de prueba utilizado la función de clasificación lineal 2.37 o no lineal 2.47 con el hiperplano deducido en el paso 3.

2.6.5. Ejemplo de aplicación

Se tiene un conjunto de fotografías las cuales se desean clasificar en dos categorías: de perfil (+1) y panorámicas (-1). Tomando en cuenta los píxeles de cada fotografía se etiquetan como se muestra en la Tabla 2.34.

Fotografía	Clase	Color	Forma	Tamaño	Detalle
1	+1	5.1	3.5	1.4	0.2
2	+1	4.9	3.0	1.4	0.2
3	-1	6.1	2.9	4.7	1.4
4	-1	5.6	2.9	3.6	1.3

TABLA 2.34: Conjunto de entrenamiento. Fotografías.

Para encontrar el hiperplano que separa las fotografías se realizan los siguientes pasos:

1. Calcular el lagrangiano.

$$
\begin{aligned}
L &= \sum_{i=1}^{4} \alpha_i - \frac{1}{2} \sum_{i=1}^{4} \sum_{j=1}^{4} \alpha_i \alpha_j y_i y_j x_i \cdot x_j \\
&= \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 - \frac{1}{2}[\alpha_1^2 y_1^2 x_1^2 + 2\alpha_1\alpha_2 y_1 y_2 x_1 \cdot x_2 + 2\alpha_1\alpha_3 y_1 y_3 x_1 \cdot x_3 \\
&+ 2\alpha_1\alpha_4 y_1 y_4 x_1 \cdot x_4 + \alpha_2^2 y_2^2 x_2^2 + 2\alpha_2\alpha_3 y_2 y_3 x_2 \cdot x_3 + 2\alpha_2\alpha_4 y_2 y_4 x_2 \cdot x_4 + \alpha_3^2 y_3^2 x_3^2 \\
&+ 2\alpha_3\alpha_4 y_3 y_4 x_3 \cdot x_4 + \alpha_4^2 y_4^2 x_4^2] \\
&= \alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 - 20.13\alpha_1^2 - 37.49\alpha_1\alpha_2 + 48.12\alpha_1\alpha_3 + 44.01\alpha_1\alpha_4 - 17.50\alpha_2^2 \\
&+ 45.45\alpha_2\alpha_3 + 41.44\alpha_2\alpha_4 - 34.83\alpha_3^2 - 61.31\alpha_3\alpha_4 - 27.21\alpha_4^2
\end{aligned}
$$

2. Construir el sistema de ecuaciones con las primeras derivadas parciales de L respecto a α_i.

$$
\begin{aligned}
\frac{\partial L}{\partial \alpha_1} &= 1 - 40.26\alpha_1 - 37.49\alpha_2 + 48.12\alpha_3 + 44.01\alpha_4 = 0 \\
\frac{\partial L}{\partial \alpha_2} &= 1 - 37.49\alpha_1 - 35.00\alpha_2 + 45.45\alpha_3 + 41.44\alpha_4 = 0 \\
\frac{\partial L}{\partial \alpha_3} &= 1 + 48.12\alpha_1 + 44.45\alpha_2 - 69.66\alpha_3 - 61.31\alpha_4 = 0 \\
\frac{\partial L}{\partial \alpha_4} &= 1 + 44.01\alpha_1 + 41.44\alpha_2 - 61.31\alpha_3 - 54.42\alpha_4 = 0
\end{aligned}
$$

Solución del sistema de ecuaciones:

$$\alpha_1 = 4.22 \qquad \alpha_3 = -16.08$$

$$\alpha_2 = 1.40 \qquad \alpha_4 = 22.63$$

3. Obtener el hiperplano

   a) Vector ortogonal

$$\begin{aligned} w &= \sum_{i=1}^{4} \alpha_i y_i x_i \\ &= \alpha_1 y_1 x_1 + \alpha_2 y_2 x_2 + \alpha_3 y_3 x_3 + \alpha_4 y_4 x_4 \\ &= 4.22(1)(5.1, 3.5, 1.4, 0.2) + 1.40(1)(4.9, 3.0, 1.4, 0.2) - 16.08(-1)(6.1, 2.9, 4.7, 1.4) \\ &+ 22.63(-1)(5.6, 2.9, 3.6, 1.3) \\ &= (-0.26, -0.02, 1.96, -5.78) \end{aligned}$$

   b) Umbral

$$\begin{aligned} b &= y_1 - w \cdot x_1 + y_2 - w \cdot x_2 + y_3 - w \cdot x_3 + y_4 - w \cdot x_4 \\ &= 1 - (-0.26, -0.02, 1.96, -5.78) \cdot (5.1, 3.5, 1.4, 0.2) \\ &+ 1 - (-0.26, -0.02, 1.96, -5.78) \cdot (4.9, 3.0, 1.4, 0.2) \\ &- 1 - (-0.26, -0.02, 1.96, -5.78) \cdot (6.1, 2.9, 4.7, 1.4) \\ &- 1 - (-0.26, -0.02, 1.96, -5.78) \cdot (5.6, 2.9, 3.6, 1.3) \\ &= 2.05 \end{aligned}$$

A partir de Ecuación 2.36, se obtiene el hiperplano de clasificación:

$$h(x) = (-0.26, -0.02, 1.96, -5.78) \cdot \vec{x} + 2.05$$

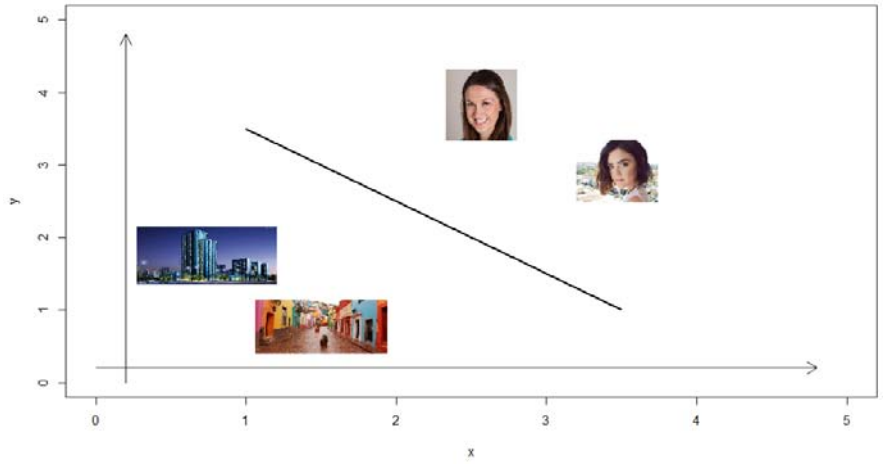


FIGURA 2.30: Hiperplano de clasificación

4. Clasificar un registro de prueba. Haciendo uso del hiperplano de clasificación encontrado.

Fotografía	Color	Forma	Tamaño	Detalle
5	4.9	3.1	1.5	0.1

Tabla 2.35: Registro de prueba. Fotografías.

$$\begin{aligned} y &= signo[(-0.26, -0.02, 1.96, -5.78) \cdot (4.9, 3.1, 1.5, 0.1) + 2.05] \\ &= signo(3.07) \end{aligned}$$

Ya que es un valor positivo, este caso de prueba pertenece a las fotografías de perfil.

$$y = +1$$

2.6.6. Ventajas y desventajas de máquina de soporte de vectores

A continuación se describen algunas ventajas y desventajas de la máquina de soporte de vectores [2][10][37].

Ventajas	**Desventajas**
•Permite modelar problemas no lineales. •Los campos donde SVM ha sido aplicada con éxito incluyen, el procesamiento de lenguaje natural y el análisis de series temporales •Permite una eficiencia razonable para problemas con miles de registros y atributos.	•Un requisito básico para aplicar con exito las SVM a un problema real es la elección de una función núcleo adecuada.

2.7. K-medias

En el año de 1967 James McQueen propuso el algoritmo de agrupamiento K-medias, es el algoritmo de aprendizaje no supervisado más utilizado por ser simple y eficaz en su aplicación, consiste en dividir un conjunto de n datos con elementos o características similares entre sí en un determinado número de K clusters, cada cluster se caracterizan por tener un centro o centroide que representa la media ponderada de los datos que lo componen. El objetivo del algoritmo K-medias es minimizar la distancia euclidiana[7] entre los valores de x y el centroide μ de cada cluster [47][56].

Notacion

- $(x_1, x_2, ..., x_n)$ conjunto de datos.
- K número de clusters.
- μ promedio de los datos asignados al cluster.

2.7.1. Algoritmo de aprendizaje

1. Elegir aleatoriamente los centroides de los agrupamientos K del conjunto de datos.

$$\mu_1, \mu_2, ..., \mu_k$$

2. Calcular la distancia entre cada uno de los datos a cada centroide. Los datos se clasifican de acuerdo a aquellos grupos cuya distancia sea mínima con respecto a todos los centroides.

$$||x_i - \mu_k|| = \sqrt{\sum_{k=1}^{K}\sum_{i=1}^{n}(x_i - \mu_k)^2} \tag{2.51}$$

3. Una vez que los datos son clasificados se calcula los nuevos centroides con la media ponderada de los datos actuales de cada cluster.

$$\mu_k = \frac{\sum_{i \in n_k} x_i}{n_k} \tag{2.52}$$

4. Se repite el paso 2 y 3 hasta que ya no se desplacen los centroides.

[7] Distancia euclidiana. Distancia en linea recta o trayectoria más corta posible entre dos puntos.

2.7.2. Medidas de calidad de clustering

Adicionalmente, se puede determinar la mejor elección del agrupamiento de los datos por medio de las técnicas de clustering llamadas Medidas de calidad de Clustering, las cuales consisten en identificar grupos con una similitud alta entre sí (intra clúster) y similitud baja entre grupos (inter clúster) [47].

La distancia inter clúster es la distancia máxima entre cada conjunto de datos, en otras palabras, los datos de un clúster tienen que ser lo más diferentes posible a los datos de otro grupo. Se dice que son diferentes mientras más alejados se encuentran de los otros grupos.

$$inter - C = \frac{\sum_{i=1}^{K-1} \sum_{j=i+1}^{K} ||\mu_i - \mu_j||}{\sum_{i=1}^{K-1} i} \tag{2.53}$$

K es el número de centroides seleccionados $||\mu_i - \mu_j||$ es la distancia entre los centroides.

La distancia intra clúster es la distancia mínima entre todos los datos dentro de un clúster, como se muestra en la Figura 2.31, las distancias se calculan entre los datos del mismo color. Al obtener distancias pequeñas se considera que comparten características en común los datos en cada grupo.

$$intra - C = \frac{\sum_{i=1}^{K} (\sum_{x \epsilon \mu_i} \frac{||x - \mu_i||}{|\mu_i|})}{K} \tag{2.54}$$

$|\mu_i|$ es el número de elementos en el centroide μ_i y $||x - \mu_i||$ es la distancia de los datos que conforman al centroide μ_i hacia el mismo μ_i.

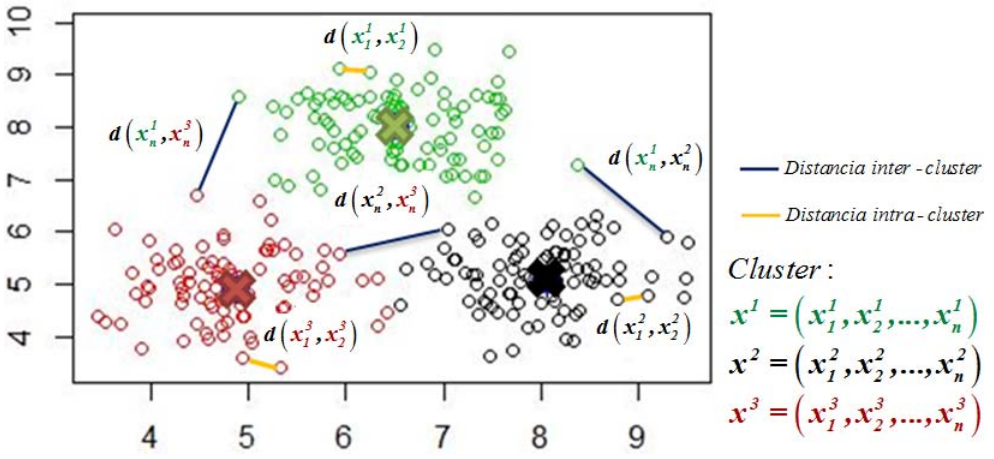


FIGURA 2.31: Distancia inter cluster e intra cluster.

2.7.3. Ejemplo de aplicación

Una empresa dedicada al entretenimiento en linea cuenta con diversos títulos de películas y series. Por lo que desea realizar recomendaciones a sus usuarios con base en la opinión de otros usuarios que comparten gustos similares sobre dos tipos de géneros: drama y terror.El procedimiento para realizar las recomendaciones se muestra a continuación.

En la Figura 2.32 se muestra el conjunto de datos sin valores asignados. Se deciden agrupar en dos Clusters ($K = 2$).

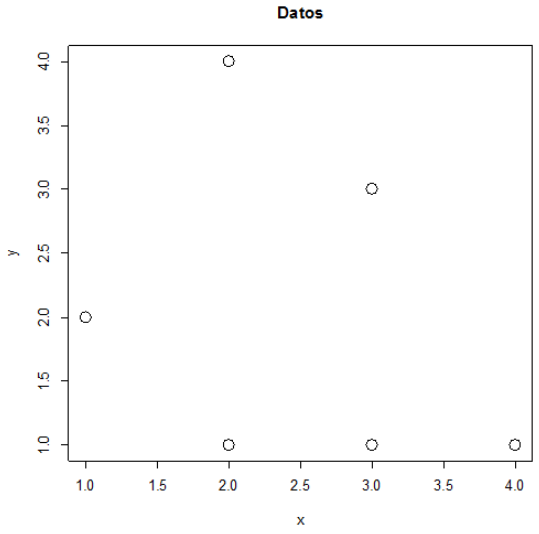


FIGURA 2.32: Conjunto de datos sin valores asignados.

Se ejecuta el algoritmo de K-medias, se eligen aleatoriamente los centroides del agrupamiento. En la Figura 2.33 los centroides se representan por una cruz roja (μ_1) y una cruz azul (μ_2).

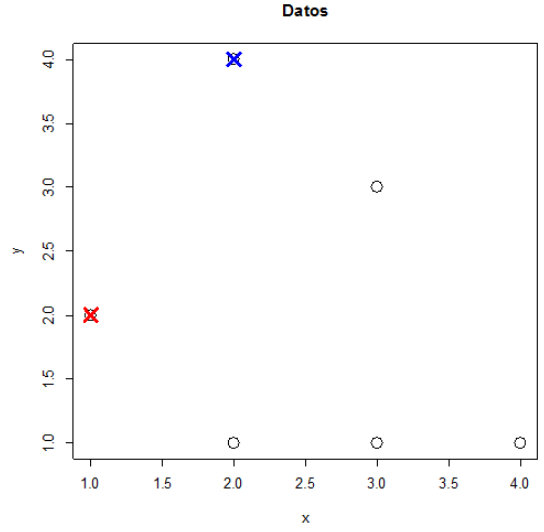


FIGURA 2.33: Centroides de agrupamiento aleatorios.

Para agrupar cada uno de los datos se calcula la distancia entre cada punto hacia cada clúster, eligiendo el cluster que minimice la distancia. De acuerdo a la Figura 2.33 los clúster seleccionados corresponden a los valores de $\mu_1 = (1, 2)$ y $\mu_2 = (2, 4)$.

Utilizando la Ecuación 2.51 se calcula la distancia del punto $x_1 = (2, 1)$ al primer clúster μ_1.

$$||x_1 - \mu_1|| = \sqrt{(2-1)^2 + (1-2)^2} = 1.4$$

Y después al segundo clúster μ_2

$$||x_1 - \mu_2|| = \sqrt{(2-2)^2 + (1-4)^2} = 3$$

Al comparar los resultados se observa que la distancia mínima pertenece al clúster μ_1 así que x_1 se agrupa en la clase 1.

El proceso se realiza para cada uno de los puntos x_i, los resultados se presentan en la Tabla 2.36.

x_i	x	y	$\|\|x - \mu_1\|\|$	$\|\|x - \mu_2\|\|$	Cluster
x_1	2	1	**1.4**	3	1
x_2	2	4	2.2	**0**	2
x_3	3	1	**2.2**	3.1	1
x_4	3	3	2.2	**1.4**	2
x_5	4	1	**3.1**	3.6	1
x_6	1	2	**0**	2.2	1

TABLA 2.36: Agrupamiento de los datos.

En la Figura 2.34, se muestra la primera agrupación realizada por el algoritmo k-means

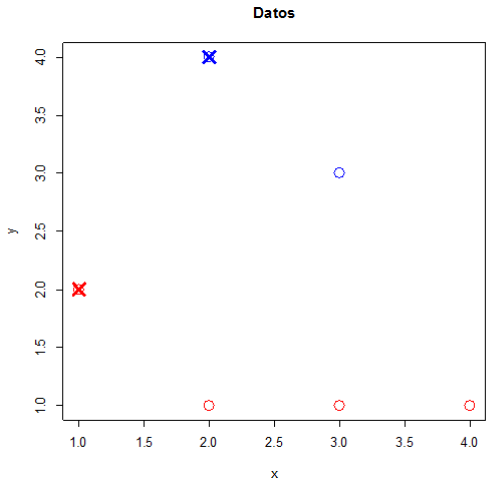


FIGURA 2.34: Asignación de datos a los centroides más cercanos.

Una vez que se llevó a cabo la primera agrupación, se toman los centroides de cada agrupamiento y se moverán al promedio de los datos pintados del mismo color, es decir, se calcula el promedio de la ubicación de todos los datos de color rojo y se moverá el centroide a esa ubicación, lo mismo se hará para el centroide de color azul (Figura 2.35).

Aplicando la Ecuación 2.52 para determinar la nueva posición:

$$\mu_k = \frac{\sum_{i \in n_k} x_i}{n_k}$$

$\mu_1 = (\frac{2+3+4+1}{4}, \frac{1+1+1+2}{4})$ =(2.5,1.25)
$\mu_2 = (\frac{2+3}{2}, \frac{4+3}{2})$ =(2.5,3.5)

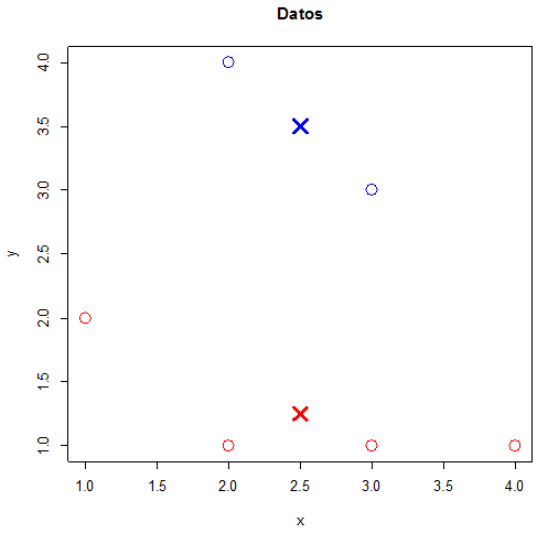


FIGURA 2.35: Movimiento de los centroides de acuerdo a la media de los datos de cada clúster.

En este sentido, si se ejecutan más iteraciones a partir de los nuevos centroides (Figura 2.35) no cambiaran más y los colores de los datos tampoco. Además, se verifica que los grupos cumplan con los criterios de distancia inter cluster (Ecuación 2.53) y distancia intra cluster (Ecuación 2.54) para garantizar que los datos han sido agrupados de manera óptima.

Aplicando las medidas de calidad de clustering (Sección 2.7.2) distancia intra-cluster y distancia inter-cluster para los primeros centroides seleccionados $\mu_1 = (1, 2)$ y $\mu_2 = (2, 4)$ se obtiene que la distancia inter-cluster (Ecuación 2.53) es:

$$\begin{aligned} inter - C &= \frac{||\mu_1 - \mu_2||}{1} \\ &= \sqrt{\frac{(1-2)^2 + (2-4)^2}{1}} \\ &= 2.23 \end{aligned}$$

Y para la distancia intra cluster (Ecuación 2.54)

$$
\begin{aligned}
intra - C &= \frac{\frac{||x_{(2,1)}-\mu_1||}{4} + \frac{||x_{(3,1)}-\mu_1||}{4} + \frac{||x_{(4,1)}-\mu_1||}{4} + \frac{||x_{(1,2)}-\mu_1||}{4} + \frac{||x_{(2,4)}-\mu_2||}{2} + \frac{||x_{(3,3)}-\mu_2||}{2}}{2} \\
&= \frac{\frac{1.4}{4} + \frac{2.2}{4} + \frac{3.1}{4} + \frac{0}{4} + \frac{0}{2} + \frac{1.4}{2}}{2} \\
&= 1.19
\end{aligned}
$$

Mientras que para el segundo conjunto $\mu_1 = (2.5, 1.25)$ y $\mu_2 = (2.5, 3.5)$ se tiene:

$$
\begin{aligned}
inter - C &= \frac{||\mu_1 - \mu_2||}{1} \\
&= \sqrt{\frac{(2.5-2.5)^2 + (1.25-3.5)^2}{1}} \\
&= 2.25 > 2.23
\end{aligned}
$$

En la segunda asignación de centorides la distancia inter-cluster es mayor que en el primer conjunto ($\mu_1 = (1, 2)$ y $\mu_2 = (2, 4)$), por lo tanto indica que los grupos en una segunda agrupación están más alejados entre sí.

$$
\begin{aligned}
intra - C &= \frac{\frac{||x_{(2,1)}-\mu_1||}{4} + \frac{||x_{(3,1)}-\mu_1||}{4} + \frac{||x_{(4,1)}-\mu_1||}{4} + \frac{||x_{(1,2)}-\mu_1||}{4} + \frac{||x_{(2,4)}-\mu_2||}{2} + \frac{||x_{(3,3)}-\mu_2||}{2}}{2} \\
&= \frac{\frac{0.5}{4} + \frac{0.5}{4} + \frac{1.5}{4} + \frac{1.6}{4} + \frac{0.7}{2} + \frac{0.7}{2}}{2} \\
&= 0.87 < 1.19
\end{aligned}
$$

Mientras que al comparar las dos distancias inter-cluster obtenidas (0.87 y 1.19), también el segundo conjunto de centroides seleccionado son los mejores debido a que la distancia es la más pequeña. Esto significa que los datos de cada grupo están mas cercanos y comparten más características en común.

Se concluye que los nuevos centroides encontrados $\mu_1 = (2.5, 1.25)$ y $\mu_2 = (2.5, 3.5)$ han agrupado correctamente los datos dado que una nueva iteración no cambiaría sus valores, además al comparar las distancias intra clúster e inter clúster siguiendo los criterios expuestos en Medidas de calidad de Clustering (Sección 2.7.2), es el mejor agrupamiento de los datos.

En este sentido, la empresa de entretenimiento en base a los resultados del agrupamiento (Figura. 2.35) tomara la decisión de recomendar los dos títulos de películas de terror o los cuatro títulos de drama de acuerdo a la preferencia de sus usuarios.

2.7.4. Ventajas y desventajas de K-medias

A continuación se describen algunas ventajas y desventajas de K-medias [2][47][64].

Ventajas	**Desventajas**
•Es sencillo y eficiente. •Aprendizaje no supervisado.	•Al ser un algoritmo iterativo suele ser lento al converger. •Se necesita especificar el número de k clusters desde un principio ya que el comportamiento del algoritmo depende del valor elegido para k.

Capítulo 3

Análisis de sentimientos en redes sociales

3.1. Escenario de aplicación

En el presente capítulo se explican los dos casos de aplicación de este trabajo de investigación. Con la finalidad de conocer como funcionan los dos tipos aprendizaje y encontrar patrones en los comentarios realizados a través de Twitter asociados a los Juegos Olímpicos de Río de Janeiro 2016 se empleara un algoritmo de aprendizaje supervisado y un algoritmo de aprendizaje no supervisado retomando las ideas descritas previamente en el capítulo 1 y en el capítulo 2.

De la misma manera, se justifica la importancia que han tenido las principales redes sociales como Facebook, Twitter, YouTube, entre otras alrededor del mundo, especialmente en México. Por ejemplo, considerando que las redes sociales se han convertido en una herramienta útil y menos costosa para realizar encuestas en línea acerca del impacto que tienen sobre las personas los eventos deportivos a analizar.

Ante la situación planteada, se realizó la extracción de 2000 tweets en español con la etiqueta #MEX relacionados a los días de participación de los atletas mexicanos en las disciplinas de atletismo, clavados y nado sincronizado para la construcción de dos tipos de aplicaciones, una de ellas enfocada al análisis de sentimientos (Aprendizaje supervisado) y la otra en recomendaciones de usuarios (@) y etiquetas (#) (Aprendizaje no supervisado).

En el análisis de sentimientos se hace uso del algoritmo de máquina de soporte vectorial con el objetivo de clasificar las palabras relevantes contenidas en los tweets de acuerdo

al tipo de polaridad detectada, preliminarmente se definieron dos categorías: positivas y negativas. [Ver Anexo I]

Con respecto al recomendador, se aplica el algoritmo K-medias para analizar los comentarios de cada usuario, identificando sus preferencias hacia cada una de las disciplinas antes mencionadas. De modo que se sugerirá de forma personalizada cuentas oficiales y etiquetas sobre atletas, medios de comunicación e inclusive usuarios cohesivos. [Ver Anexo I y Anexo II]

3.1.1. Redes sociales

El aumento de usuarios con acceso a internet ha permitido el crecimiento de las redes sociales, convirtiéndolas en las principales protagonistas para que millones de personas alrededor del mundo realicen críticas, opiniones, expectativas e inclusive expresen sus sentimientos hacia personas, eventos, productos, lugares y marcas. Las redes sociales han cambiado la forma de comunicación y de difusión de la información, ahora las personas que ni siquiera están siguiendo un determinado tema terminan por enterarse hasta llegar a tener una opinión al respecto. En este trabajo de investigación son de suma importancia los comentarios generados en las redes sociales, especialmente en Twitter, para realizar un análisis de sentimientos y un recomendador.

En un estudio realizado por la agencia de marketing y comunicación online *We are social*, las redes sociales más populares a nivel mundial se muestran en la Figura 3.1:

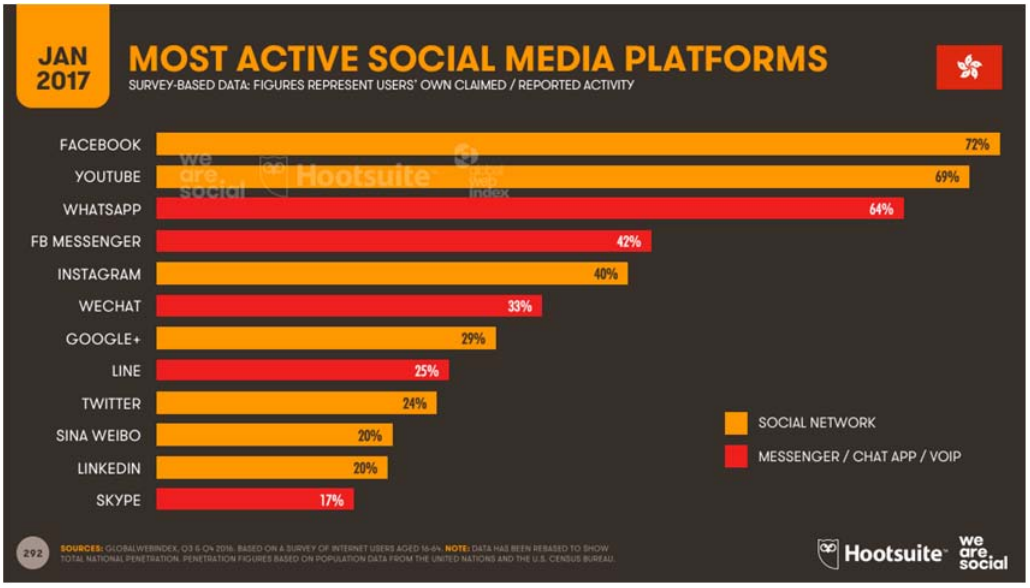


FIGURA 3.1: Las redes sociales más populares, obtenido de https://wearesocial.com/blog/2017.

Se estima que el total de la población mundial en 2017 es de 7.476 billones de personas de las cuales el 50 % tienen acceso a Internet, el 37 % son usuarios activos en redes sociales y el 34 % utilizan dispositivos móviles para acceder a sus redes sociales (Figura 3.2).

FIGURA 3.2: Accesibilidad de Internet a la población, obtenido de https://wearesocial.com/blog/2017.

En la figura 3.3 se aprecia el porcentaje de personas que tienen acceso a Internet por cada región del mundo de los cuales Norte América y la Europa Occidental tienen un mayor acceso a Internet con 88 % y 84 %, respectivamente, mientras Centro América cuenta con un 53 %.

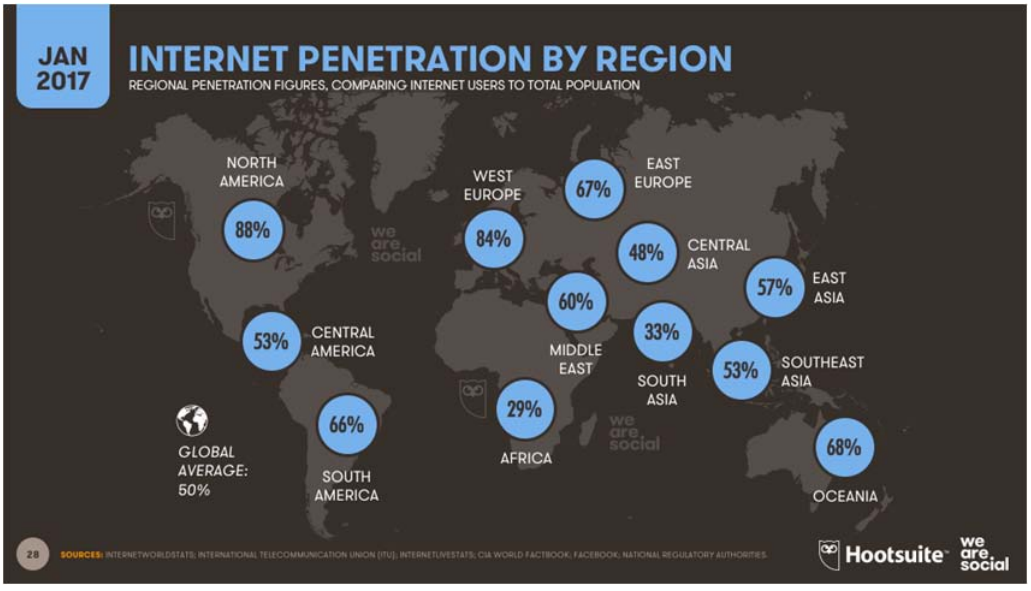


FIGURA 3.3: Porcentaje de la población con acceso a Internet, obtenido de https://wearesocial.com/blog/2017.

En la figura 3.4 el tiempo que invierten las personas en cada país en Internet mediante equipo de cómputo es de máximo 5 horas al día y mediante dispositivos móviles es de 4 horas al día. En México el máximo de horas en equipo de cómputo es de 4:47 horas y en dispositivos móviles 3:35 horas.

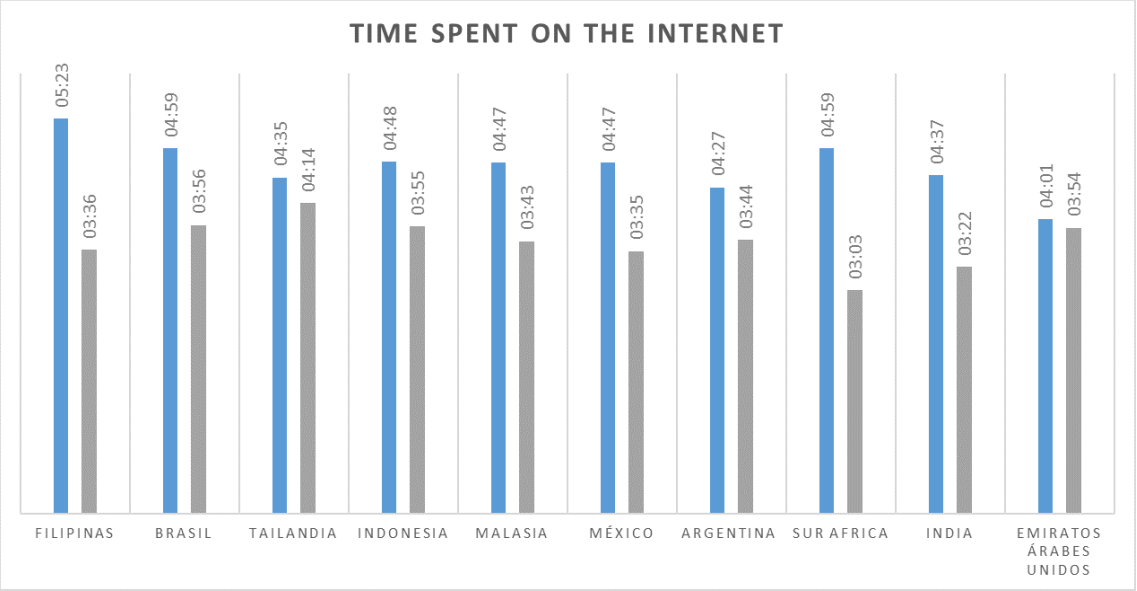


FIGURA 3.4: Promedio del tiempo invertido en Internet durante el día, obtenido de https://wearesocial.com/blog/2017.

En la figura 3.5 se muestra el porcentaje de personas que usan redes sociales por cada región del mundo. En la cual Centroamérica tiene un 51 % en uso de redes sociales.

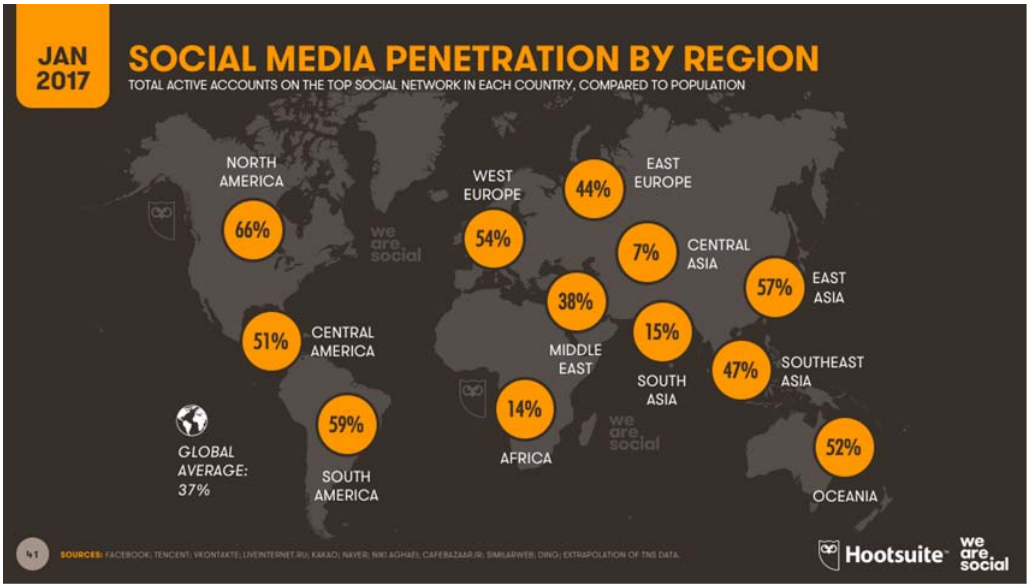


FIGURA 3.5: Porcentaje de la población con acceso a redes sociales, obtenido de https://wearesocial.com/blog/2017.

En la figura 3.6 se muestra el promedio de horas al día que los usuarios en cada país utilizan para estar conectados a sus redes sociales donde el máximo es de 4:17 horas al día.

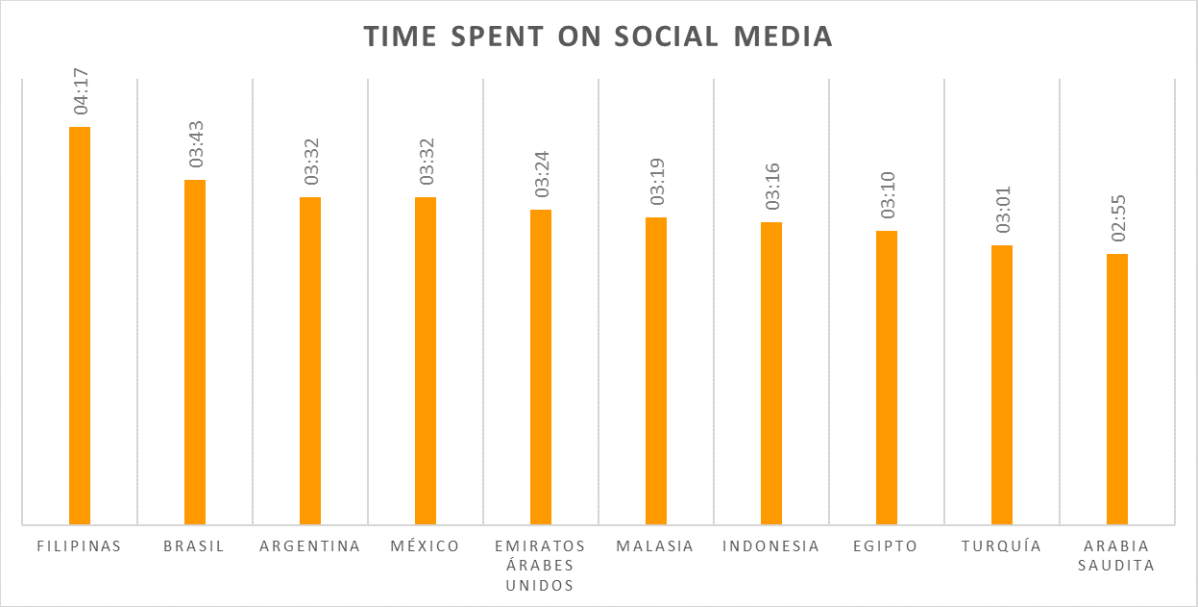


FIGURA 3.6: Porcentaje Promedio del tiempo invertido en redes sociales durante el día, obtenido de https://wearesocial.com/blog/2017.

México es el cuarto país en el mundo con mayor penetración de redes sociales, con 3:32 horas al día en la que los internautas están conectados a una o más redes sociales.

3.2. Twitter

Twitter es una de las redes sociales más populares a nivel mundial. Fue fundada en el año 2006 por Jack Dorsey, Biz Stone y Evan Williams. En un inicio Twitter fue creado como un proyecto interno de la empresa de podcast Odeo en San Francisco, cuatro meses después se independizó para hacer su presentación oficial a todos los usuarios de internet y desde entonces Twitter ha crecido y tomado relevancia porque cuenta con más de 320 millones de usuarios activos que generan 500 millones de mensajes a diario.

Twitter es una plataforma de microblogging (tipo de servicio que consiste en el envío y publicación de mensajes breves) que permite a los usuarios comunicarse por mensajes cortos de 140 caracteres como máximo, estos mensajes son conocidos como tweets. Además del contenido textual un tweet suele contener las siguientes abreviaturas:

- Etiquetas o Hashtag (#): clasifica los tweets de un determinado tema.

- Menciones (@): hacen referencia a un usuario o a un lugar.
- Retuit (RT): un usuario comparte el tweet de otros usuarios.
- Tendencias: (*trending topic*): son las etiquetas más comentadas por los usuarios en un momento determinado.

El bloque donde se publican los tweets se llama *Timeline*. Los tweets van apareciendo del más reciente al más antiguo con la indicación de los días, horas, minutos, segundos de publicación. Twitter deja abierta la posibilidad de conectarse con una u otra persona sin necesidad de tener que validar una invitación a seguir a alguien (*following*) o invitar a determinados usuarios a seguirnos (*followers*), de modo que los usos sociales que se han ido dando a esta herramienta la han llevado a ser más bien un canal de noticias en tiempo real dirigido a personas que normalmente comparten intereses similares.[1]

Twitter ha demostrado ser una red social útil y eficaz, pero presenta ciertos inconvenientes como tener que resumir los comentarios a 140 caracteres dejando algunas ideas inconclusas, los perfiles falsos que cualquier usuario puede crear ocasiona desconfianza en los demás usuarios y el mal uso de las etiquetas (#) dificulta la búsqueda de información que beneficie al análisis que se desee realizar.

3.3. Juegos Olímpicos

Los juegos olímpicos es uno de los eventos deportivos más importantes a nivel mundial que se organizan cada cuatro años en una sede diferente donde atletas de diversas disciplinas deportivas compiten de manera individual o grupal durante un periodo de quince días. Desde los juegos olímpicos de Pekín 2008, donde ya existían las redes sociales, hubo un crecimiento importante en los juegos olímpicos de Londres 2012 donde las redes sociales fueron el principal medio para comunicar noticias, controversias y emociones en torno a esas Olimpiadas, dicha razón las llevó a considerarlas como los primeros juegos olímpicos sociales.

Cada vez más las redes sociales están implementando estrategias para que los usuarios interactúen de manera más activa durante el periodo de las competencias. Twitter preparó cambios en su plataforma para los Juegos Olímpicos 2016 al diseñar algunas funciones, por ejemplo, al escribir las tres primeras letras del código en inglés, francés, portugués o español de cada país se activaba una bandera *emoji* para cada equipo participante. También tras escribir el deporte que se estaba apoyando mostraba un logo alusivo a éste.

[1] Rueda Mancera, Ana. El discurso político en twitter: análisis de mensajes que "trinan". Anthropos. Barcelona 2013

A manera de complementar esta investigación, se seleccionaron los juegos olímpicos aprovechando la reciente edición y el interés del sector mexicano en la red social Twitter.

En las siguientes secciones se aplicará la metodología de Machine Learning descrita en la sección 1.6 para hacer el análisis de sentimientos y el recomendador.

3.4. Recolección de datos

El primer paso en el proceso de análisis es la recolección de datos. Para ello, se utilizó la API pública de Twitter (REST APIs)[2] y el lenguaje de programación R [3] (RStudio 1.0.153).

3.4.1. Twitter API

Las APIs son un conjunto de funciones y protocolos informáticos que utilizan las redes sociales para comunicarse con otros softwares o aplicaciones. Las APIs evitan crear código desde cero pues utilizan funciones ya existentes.

Twitter brinda a los desarrolladores tres tipos de APIs:

- Streaming APIs: permite el acceso a datos públicos al mantener una conexión HTTP abierta con el mayor tiempo posible (conexión persistente).
- REST APIs: hace peticiones a Twitter y regresa como resultado los datos en formato XML o JSON.
- Ads API: permite a las empresas crear y administrar campañas publicitarias.

Es indispensable tener una cuenta en Twitter para hacer uso de las APIs.

Para crear una aplicación se inicia por ingresar el nombre, la descripción y la IP de acceso de la misma. (Figura 3.7)

[2] https://apps.twitter.com/

[3] R es un lenguaje de programación, de distribución libre, bajo licencia GNU, y se mantiene en un ambiente de computo en el que se aplican técnicas estadísticas para el tratamiento, análisis y representación gráfica de datos.

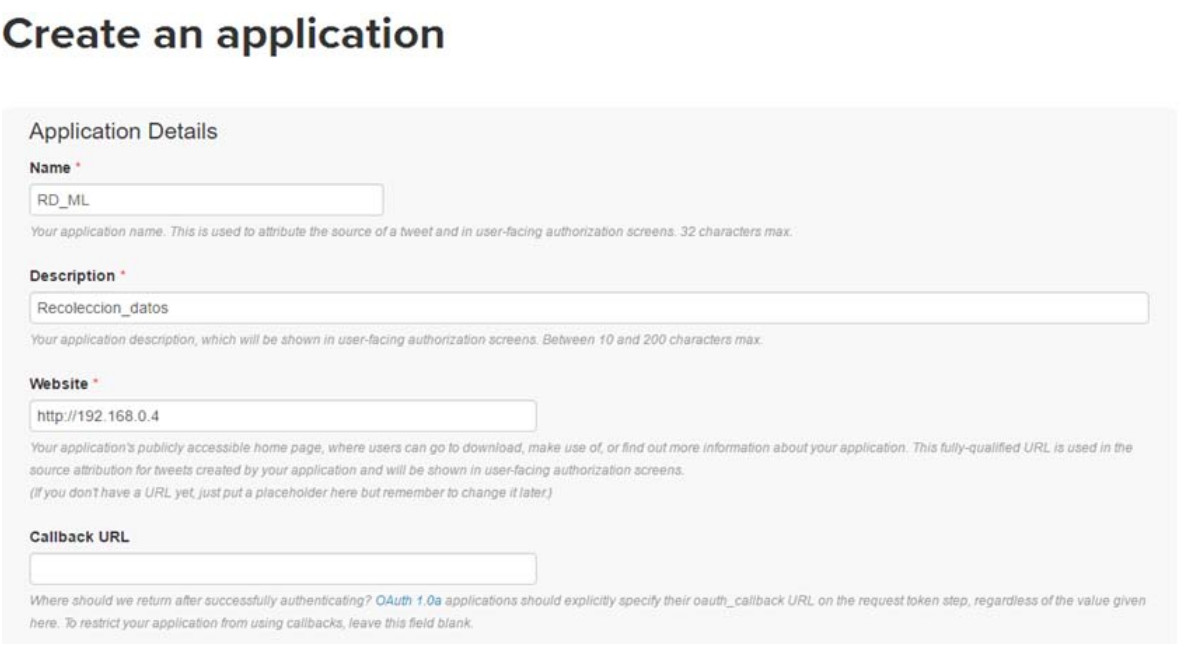


FIGURA 3.7: Crear una API en Twitter API.

Al momento de registrar la aplicación se generan una serie de claves (Keys and Access Tokens). Las claves que la plataforma proporciona a cada desarrollador son cuatro: Consumer Key (API Key), Consumer Secret (API Secret), Access Token y Access Token Secret.(Figura 3.8 y 3.9)

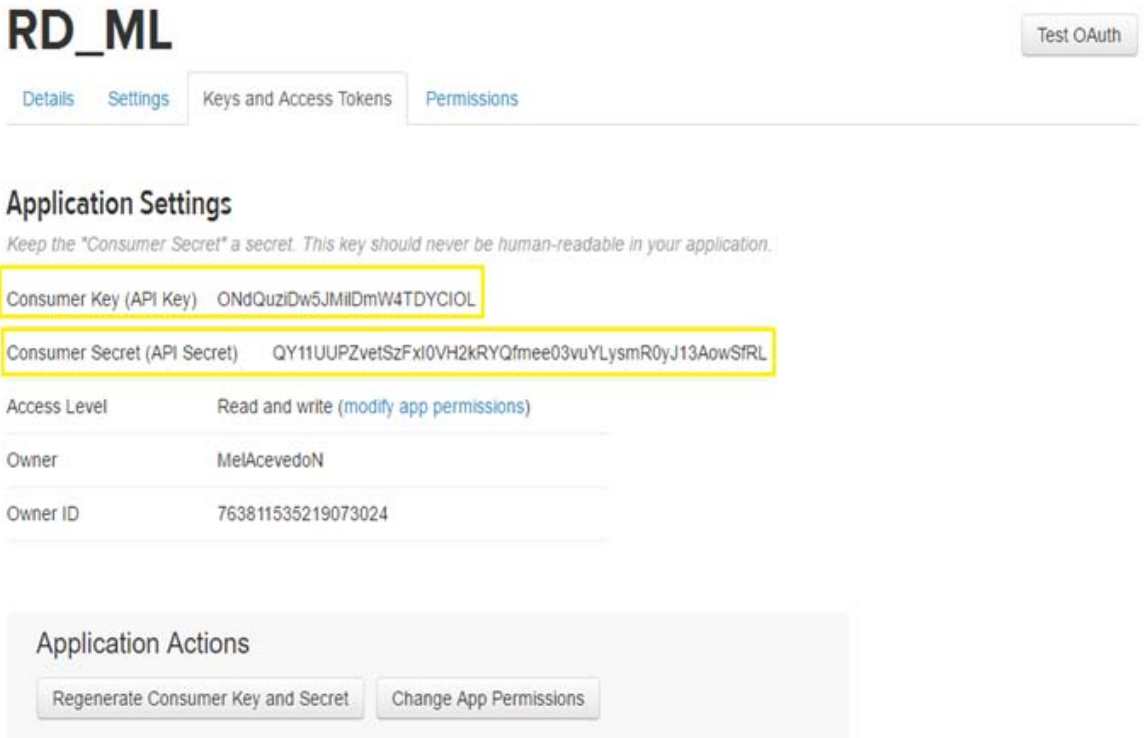


FIGURA 3.8: Keys generadas por Twitter API.

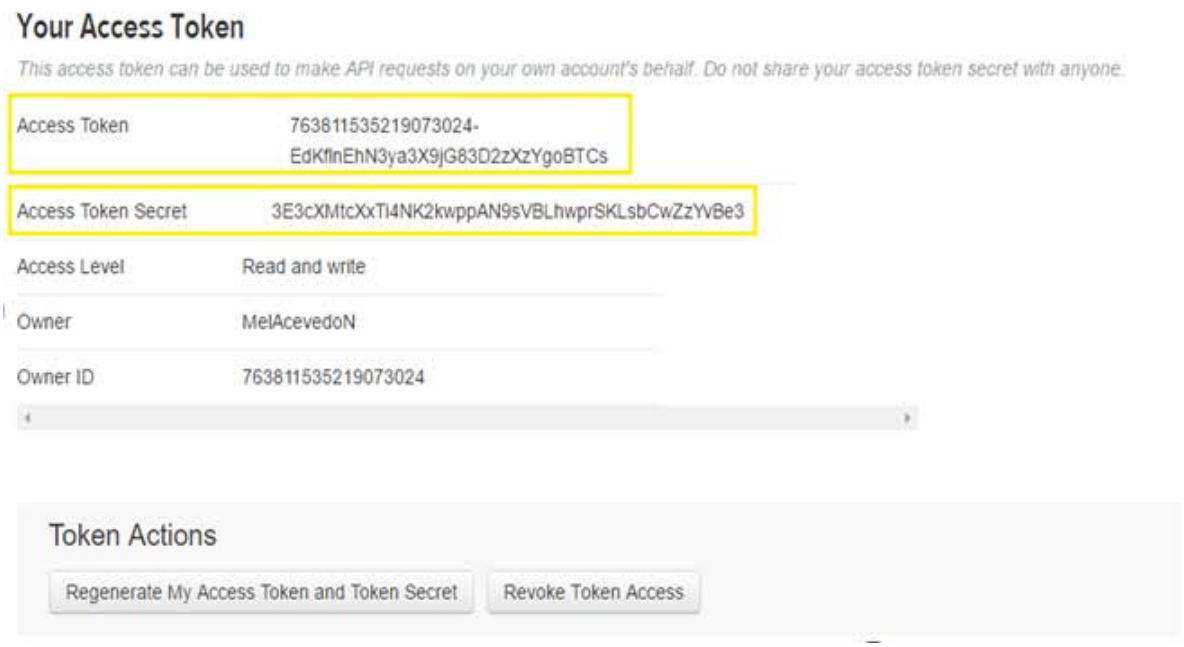


FIGURA 3.9: Access Token generadas por Twitter API

Las Keys y Access Tokens que genera Twitter servirán para comunicarse e interactuar con R, aunque el usuario no ve el proceso que se realiza internamente para la extracción de los datos.

Se ingresan las claves de acceso junto con el siguiente código:

```
#Librerías
library(twitteR)
library(ROAuth)

#Keys and Access Tokens generados por la API de Twitter
consumer_key=‘‘ONdQuziDw5JMilDmW4TDYCIOL’’
consumer_secret=‘‘QY11UUPZvetSzFxI0VH2kRYQfmee03vuYLysmR0yJ13AowSfRL’’
access_token=‘‘763811535219073024-EdKflnEhN3ya3X9jG83D2zXzYgoBTCs’’
access_secret=‘‘ 3E3cXMtcXxTi4NK2kwppAN9sVBLhwprSKLsbCwZzYvBe3’’

setup_twitter_oauth(consumer_key, consumer_secret, access_token, access_secret)
```

El paquete ROAuth de R verifica que la conexión a Twitter sea segura, es decir, la autenticación desde la API se realiza a través del protocolo OAuth para que los usuarios establezcan una comunicación hacia Twitter sin compartir completamente su identidad.

La función setup_twitter_oauth lee las claves de acceso que proporciona la API de Twitter. Si dicha autenticación fue correcta, la consola de R mostrará el mensaje

"Using direct authentication" y en dicho momento las funciones del paquete TwitteR se vuelven disponibles para el uso de esa sesión.

Se comienzan a descargar 2000 tweets relacionados con el #MEX con el siguiente código:

```
#Búsqueda y descarga de tweets
a.list<-searchTwitter('#MEX',n=2000)
a.df=twListToDF(a.list)
write.csv(a.df,file='C:/Users/Desktop/Rio2016.csv',row.names=F)
```

La paquetería *TwitteR* proporciona la función *searchTwitter* para la búsqueda de tweets basadas en cadenas que incluyen etiquetas o menciones. Cada búsqueda da la posibilidad de filtrar la información de acuerdo a lo que se quiera analizar como lo son: la etiqueta o usuario y el número de tweets a descargar.

Una vez que se obtiene la lista de tweets solicitados en Twitter, la función *twListToDF* convertirá dichos datos en un data frame, el cual facilita el almacenamiento de los tweets en archivos de tipo CSV.

Los tweets son extraídos a partir de la fecha más reciente de las etiquetas que se le indiquen, mostrando el id del tweet, el tweet, si está marcado como favorito, el número de favoritos, la fecha de publicación, quien lo publicó, desde que tipo de dispositivo se publicó, si tiene retuits, el número de retuits e inclusive la localización geográfica donde fueron publicados (Figura 3.10).

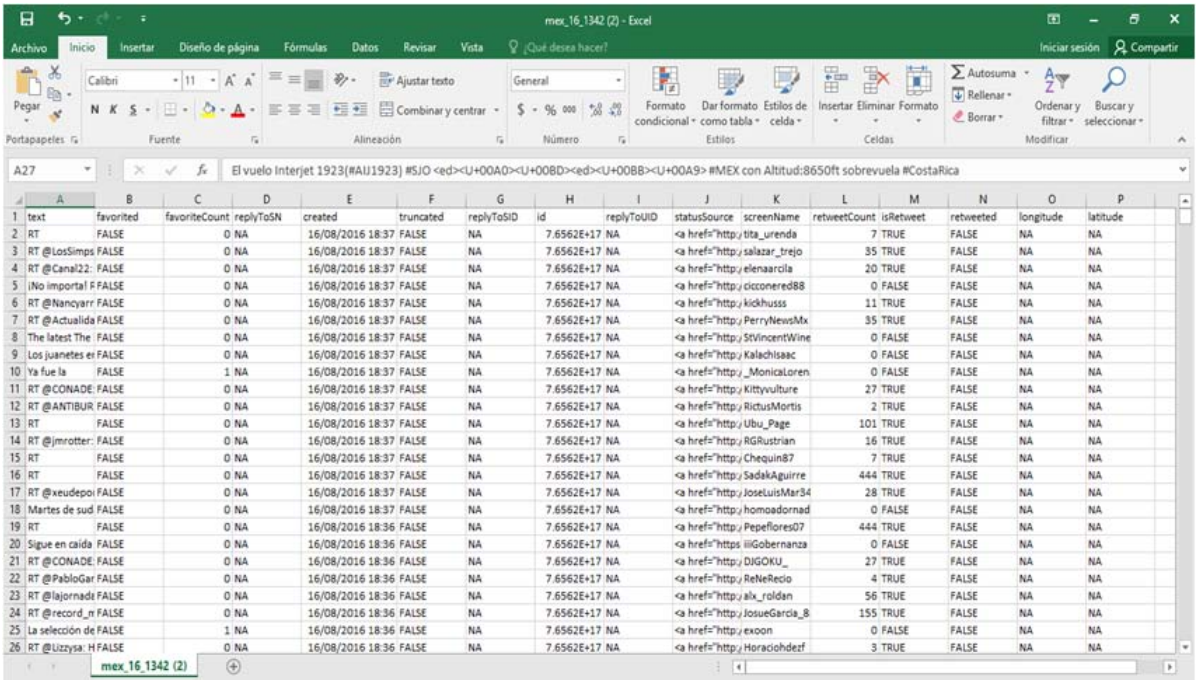


FIGURA 3.10: Extracción de datos en Twitter.

3.5. Preparación y transformación de datos

En esta etapa se someterán los datos a un tratamiento para convertirlos de datos no estructurados [4] a datos estructurados [5].

El lenguaje de programación R también permite realizar la limpieza de los tweets. La limpieza de tweets trata de remover o sustituir ciertos caracteres que son innecesarios o no aportan mucho valor al análisis.

En la figura 3.11 se muestra el proceso de la limpieza de los tweets.

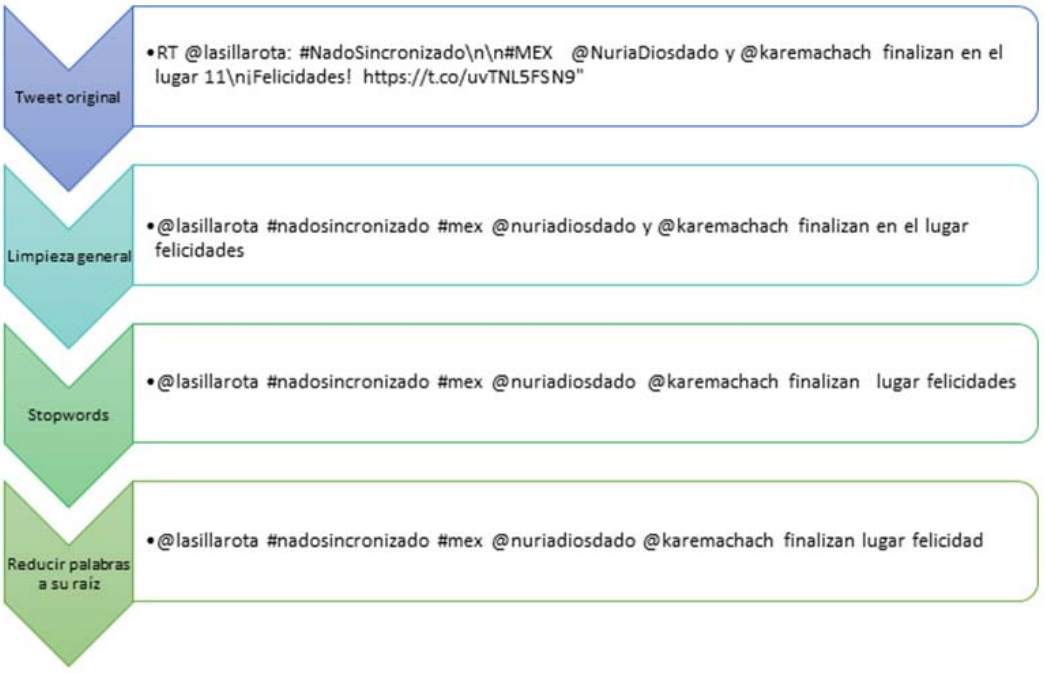


FIGURA 3.11: Limpieza de tweets.

La limpieza general consiste en convertir el texto en minúsculas, remover: las direcciones electrónicas, *retweets*, signos de puntuación, números y espacios en blanco extras. Después de este proceso, se eliminan las *stopwords* que son las preposiciones, artículos y conjunciones que tenga el texto debido a que no aportan ningún significado en el análisis. Finalmente, cada palabra se reduce a su raíz con el fin de mejorar la velocidad de búsqueda en todo el documento.

[4]Datos no estructurados. Son datos que existen en su estado original (sin refinar), es decir, en el formato en el que se recolectaron. Por ejemplo, hojas de calculo, redes sociales, musica, imágenes, archivos de texto y correo electrónico.

[5]Datos estructurados. Son el resultado de tomar datos no estructurados y formatearlos (estructurarlos) para facilitar el almacenamiento, uso y generación de información. Por ejemplo, bases de datos relacionales.

3.5.1. Transformación

Para facilitar el uso de los algoritmos de aprendizaje, se reemplazaron las palabras por valores numéricos a través de una matriz binaria. En la construcción de la matriz binaria se aplica el proceso de transformación a cada uno de los tweets, el cual consiste en asignar un 0 si la palabra no se encuentra en ese tweet y 1 si la palabra es mencionada en dicho tweet (Figura 3.12).

[1] " @lasillarota #nadosincronizado #mex @nuriadiosdado @karemachach finalizan lugar felicidad"

#mex	#nadosincronizado	orgullo	#nikeplus	lugar	felicidad	#rio
1	1	0	0	1	1	0

FIGURA 3.12: Construcción de la matriz binaria.

Este proceso se realiza para todos los tweets. La matriz resultante se muestra en la Figura 3.13. Donde cada fila representa el número de tweet y cada columna corresponde a las palabras más relevantes de la base de datos.

	row.names	#juegosolimpicosrio	#lajornadaol◆mpica	#londr	#m◆	#mex	#mex◆	#nadosincronizado	#nikeplus
1	character(0)	0	0	0	0	1	0	1	0
2	character(0)	0	0	0	0	1	0	1	0
3	character(0)	0	0	0	0	1	0	1	0
4	character(0)	0	0	0	0	1	0	0	0

FIGURA 3.13: Matriz binaria.

Por último, se remueven las palabras con frecuencia baja para reducir la dimensionalidad de la matriz binaria [Anexo 1. Transformación de datos].

3.6. Análisis de sentimientos

El análisis de sentimientos es útil para clasificar las opiniones en positivas o negativas de acuerdo a su contenido textual generado sobre un determinado tema, por ejemplo, los comentarios en redes sociales sobre los Juegos Olímpicos.

Los pasos a seguir para la realización del análisis de sentimientos en esta investigación se muestran en la Figura 3.14.

FIGURA 3.14: Diagrama de proceso: Análisis de sentimientos.

3.6.1. Datos

En el análisis de sentimientos es necesario "enseñarle" a la computadora con base en ejemplos cuales son las palabras relacionadas con tweets positivos o negativos.

Para esta investigación se decidió clasificar manualmente los tweets, considerando que la computadora tendría un mayor porcentaje de aprendizaje si se le proporcionaban ejemplos clasificados por una persona que por una computadora. Existe la posibilidad de clasificar los tweets con la ayuda de R utilizando la paquetería *Sentiment*.

De acuerdo al contenido de cada uno de los tweets se le asignó el número 1 si el comentario es positivo o -1 si es negativo (Figura 3.15).

text	
¡No importa! Representaron a nuestra nación, un aplauso para ustedes campeonas @NuriaDiosdado , @karemachach #MEX	1
RT @record_mexico: El velocista de #MEX José Carlos Herrera reprochó la falta de apoyo: "En México todo es futbol'	-1
RT @FOXSportsMX: QUÉ BUENO	1
RT @PabloGamboa_MX: Todo #Yucatán está orgulloso de ti @karemachach. Hiciste una gran rutina, eres una campeona y gra	1
Hacer el intento es la peor actitud de un deportista mexicano y que es igual a mediocridad #MEX #NadoSincronizado #Rio2016	-1
@Rommel_Pacheco Éxito!!!#Rio2016 #MEX #Clavados	1

FIGURA 3.15: Clasificación manual de tweets.

3.6.2. Integración

En el análisis de sentimientos, los datos no se insertaron en otra base de datos o en Data Warehouse. Se trabajo con la matriz binaria que se generó en R (Figura 3.13), fue indispensable agregar un atributo sentimiento. [Anexo 1. Integración]

3.6.3. Reconocimiento de patrones

Una vez que se tienen integrados los tweets clasificados en la matriz binaria se implementa el algoritmo de aprendizaje. Para el análisis de sentimientos se decidió utilizar el algoritmo de máquina de soporte de vectores (SVM), ya que es uno de los algoritmos más eficientes

en la solución de problemas de clasificación y en el procesamiento de lenguaje natural como se mencionó en la sección 2.6.6.

Máquina de soporte de vectores al ser un algoritmo de aprendizaje supervisado, así que se tiene que definir un conjunto de entrenamiento y un conjunto de prueba. Para ello, los datos se dividen de acuerdo a la ley de Pareto (Sección 1.4. Aprendizaje supervisado)

3.6.4. Validación

Se realizaron dos iteraciones con el objetivo de medir el porcentaje de aprendizaje de la computadora. En la primera iteración se utilizaron 1000 tweets y en la segunda iteración 2000 tweets.

La primera iteración se entrenó el algoritmo de aprendizaje SVM utilizando los tweets de entrenamiento. En la figura 3.16 se presentan los resultados del algoritmo de aprendizaje que nos proporciona R.

Los resultados que se muestran en la Figura 3.16 son los parámetros que se utilizan para obtener el clasificador empezando por la función kernel que en este caso es de tipo radial (Sección 2.6.2. Función Kernel), un costo de 1 por ser un costo menor representa la maximización del margen con tolerancia a errores, el valor gamma de 0.003802281 determina la curva de separación y finalmente se indica el número de vectores de soporte que es de 275 de los cuales 201 son tweets negativos y 74 son tweets positivos.

```
Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  1
      gamma:  0.003802281

Number of Support Vectors:  275

 ( 201 74 )


Number of Classes:  2

Levels:
 -1 1
```

FIGURA 3.16: Resultado del algoritmo máquina de soporte vectorial. 1° iteración.

En la figura 3.17 se muestra la matriz de confusión. Su diagonal representa los tweets que se clasificaron correctamente y los valores fuera de la diagonal son los tweets que se clasificaron incorrectamente, es decir, los 12 tweets indican que se etiquetaron como negativos, pero en realidad pertenecen a la clase positiva, por el contrario, ningún tweet

se clasificó como positivo siendo negativo. En este sentido, se tiene un modelo que permite clasificar los tweets con un 94 % de exactitud.

```
Confusion Matrix and Statistics

          Reference
Prediction  -1   1
        -1  34   0
        1   12 158

               Accuracy : 0.9412
                 95% CI : (0.8995, 0.9692)
    No Information Rate : 0.7745
    P-Value [Acc > NIR] : 8.402e-11
```

FIGURA 3.17: Predicción del algoritmo máquina de soporte vectorial. 1° iteración.

En la segunda iteración se repite el mismo procedimiento que en la primera iteración, pero ahora utilizando 2000 tweets. En la figura 3.18 se presentan los resultados del algoritmo de aprendizaje. Los cuales muestran la función kernel de tipo radial, el costo con valor 1, el valor de gamma de 0.003484321 y el número de vectores de soporte que es de 445 los cuales 307 son negativos y 138 son positivos.

```
Parameters:
   SVM-Type:  C-classification
 SVM-Kernel:  radial
       cost:  1
      gamma:  0.003484321

Number of Support Vectors:  445

 ( 307 138 )


Number of Classes:  2

Levels:
 -1 1
```

FIGURA 3.18: Resultado del algoritmo máquina de soporte vectorial. 2° iteración.

En la figura 3.19 se muestra la matriz de confusión. En la diagonal se muestra que se han clasificado correctamente 51 tweets negativos y 331 tweets positivos. Se obtuvo un aumento en el porcentaje de aprendizaje a un 95 %.

```
Confusion Matrix and Statistics

          Reference
Prediction  -1   1
        -1  51   1
        1   19 331

               Accuracy : 0.9502
                 95% CI : (0.9242, 0.9693)
    No Information Rate : 0.8259
    P-Value [Acc > NIR] : 4.884e-14
```

FIGURA 3.19: Predicción del algoritmo máquina de soporte vectorial. 2° iteración.

Al comparar la predicción realizada en ambos modelos se concluye que al proporcionar más datos de entrenamiento el algoritmo de máquina de soporte vectorial mejora el porcentaje de aprendizaje y por consiguiente las predicciones son más fiables. Por lo tanto, se utilizó el segundo modelo para hacer las predicciones.

3.6.5. Predicción

A continuación, se muestran las predicciones que realiza el algoritmo de aprendizaje utilizando 2 tweets del conjunto de prueba.

1. RT @record_mexico: El velocista de #MEX José Carlos Herrera reprochó la falta de apoyo: En México todo es fútbol.
2. Que orgullo @NuriaDiosdado @karemachach lo hicieron excelente! Muchas felicidades. #NadoSincronizado #Mex

Los tweets de prueba son suministrados al modelo final de SVM por medio de la creación de una nueva matriz binaria a partir de la matriz binaria de los tweets de entrenamiento. Esto ayudará a reconocer las palabras clave en los tweets de prueba y generar una predicción acertada.

El algoritmo automáticamente asignará cada tweet en alguno de los dos sentimientos ya que ha aprendido cómo hacerlo.

Tweet	Clasificación
RT @record_mexico: El velocista de #MEX José Carlos Herrera reprochó la falta de apoyo: "En México todo es futbol'	-1
Que orgullo @NuriaDiosdado @karemachach lo hicieron excelente! Muchas felicidades. #NadoSincronizado #Mex	1

FIGURA 3.20: Clasificación de nuevos tweets.

Como se muestra en la figura 3.20, el algoritmo de máquina de soporte de vectores, clasificó el primer tweet como negativo mientras que el segundo tweet como positivo.

3.6.6. Interpretación

A partir de los resultados del modelo, se hizo un análisis de las palabras contenidas en los tweets positivos y negativos mediante nubes de palabras. Para tener una mejor visualización se decidió colocar en las nubes las palabras con una frecuencia mayor a 50, es decir, las palabras y etiquetas más mencionadas en los tweets.

3.6.6.1. Palabras: Tweets positivos

La nube de palabras para los tweets positivos se obtuvo al filtrar de la matriz binaria aquellas tuplas asociadas con el sentimiento igual a 1. Dando como resultado 100 palabras que tuvieron más menciones en los tweets positivos.

FIGURA 3.21: Nube de palabras de tweets positivos.

Las palabras obtenidas en los tweets positivos (Figura 3.21), sobresalen términos como: esperanza, mejor, esfuerzo, bien, felicidad, perfecta, hermosa, satisfecha, venga, orgullo, excelente. Todas estas palabras están relacionadas principalmente a la etiqueta #nadosincronizado y a las atletas mexicanas Nuria Diosdado y Karen Machado. Otra de las disciplinas que destaca por tener opiniones positivas fue la de clavados, en la que Rommel Pacheco fue de los atletas con más apoyo durante su participación. Además, los usuarios manifestaron su agrado por los medios de comunicación como: FOX Sport y ESPN, los cuales mencionaron la obtención de la primera medalla de bronce en boxeo y el pase a semifinales del velocista José Carlos Herrera.

3.6.6.2. Palabras: Tweets negativos

La nube de palabras para los tweets negativos se realiza con el mismo proceso de la nube de palabras positivas. En este caso, se filtran los tweets con polaridad -1 a fin de obtener los términos más sobresalientes en los comentarios negativos, los cuales fueron 23.

FIGURA 3.22: Nube de palabras de tweets negativos.

Los tweets negativos (Figura 3.22) contienen palabras relacionadas al evento de Atletismo y a su representante José Carlos Herrera. Estos malos comentarios son debido a la falta de apoyo por parte de la CONADE y su dirigente Alfredo Castillo, este suceso se identificó como el que más desagrado a los usuarios en Twitter.

Adicionalmente, clavados y nado sincronizado tuvieron comentarios negativos en menor cantidad comparado con la situación de los tweets positivo.

3.7. Recomendador

El objetivo principal de un recomendador es identificar las preferencias y hacer sugerencias personalizadas a los usuarios de algún tema que pueda ser de su interés o de cuentas que comparten ideas semejantes como por ejemplo deportes, atletas y marcas. Estas sugerencias se realizarán mediante el uso de las etiquetas (#) y las menciones (@). En la Figura 3.23 se describe el proceso para elaborar un recomendador.

FIGURA 3.23: Diagrama de proceso: Recomendador

3.7.1. Datos

El recomendador se construyó a partir de las opiniones hechas por los usuarios a través de sus tweets. Se extrajeron 2000 tweets que corresponden al día en que atletas mexicanos tuvieron participación en las siguientes categorías: atletismo, clavados y nado sincronizado.

3.7.2. Integración

Al igual que el análisis de sentimientos, fue indispensable concentrar los mensajes de twitter en una matriz binaria debido a la facilidad de procesamiento en R para datos de tipo texto.

3.7.3. Reconocimiento de patrones

Se propone utilizar el algoritmo K-medias, ya que es uno de los algoritmos de aprendizaje no supervisado (Sección 1.5) que divide los datos en grupos que comparten características similares. Esto ayudará a realizar las recomendaciones adecuadas y de manera rápida a los usuarios.

Con la intención de obtener mejores resultados se decide realizar 2 iteraciones. La primera iteración con 3 clusters y la segunda iteración con 4 clusters.

3.7.4. Validación

Para validar que se están usando el número de clusters óptimo se analiza la gráfica con 3 clusters (Figura 3.24). En el cluster 1 se mezclan palabras relacionadas a las disciplinas de nado sincronizado y clavados. En el cluster 2 aparecen palabras referentes a atletismo y en el cluster 3 las palabras sobre la CONADE.

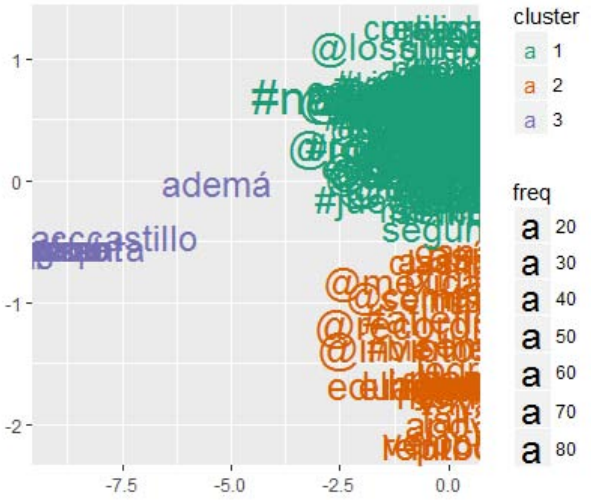


FIGURA 3.24: Gráfica con 3 clusters.

Por otra parte, se calculan las medidas de calidad del clustering (Sección 2.7.2) para determinar que la elección del número de clusters es la adecuada.

Distancia inter cluster: 1579.717

Distancia intra cluster: 16374.1

Se tiene una distancia máxima entre cada grupo de 1579.717 y una distancia mínima de 16374.1 entre los datos de cada grupo.

Por otro lado, en la gráfica con 4 clusters (Figura 3.25) se muestran las palabras con una frecuencia mayor o igual a 50, esto con el propósito de tener una mejor visualización de las palabras que pertenecen a cada cluster.

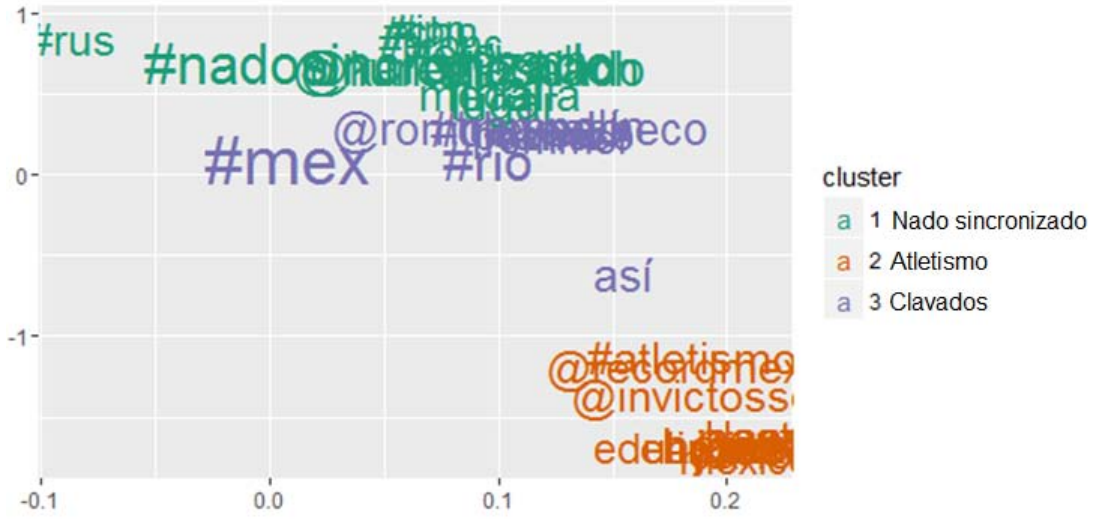


FIGURA 3.25: Gráfica con 4 clusters.

En la Figura 3.25 se presentan 3 clusters, el primer cluster esta formado por palabras que aparecen en los comentarios referentes a nado sincronizado, el segundo cluster se conforma por palabras sobre atletismo y el tercer cluster por palabras sobre clavados. El cuarto cluster no aparece en la gráfica debido a que las palabras en el son de poca frecuencia.

Se calculan las medidas de calidad del clustering

Distancia inter cluster: 2494.953

Distancia intra cluster: 15529.62

Al implementar el algoritmo K-medias con cuatro clusters se tiene una distancia máxima entre cada grupo de 2494.953 y una distancia mínima de 15529.62 entre los datos de cada grupo.

Por lo tanto, se decide que el número de clusters sea 4 ya que la distancia inter cluster aumento y la distancia intra cluster disminuyo. Además, cada una de las disciplinas tiene su propio cluster y uno adicional para las palabras que no tienen relación con estas disciplinas.

3.7.5. Predicción

Ya que se verificó que el número de cluster a usar es 4, se aplica el algoritmo de K-medias a la matriz binaria (Sección 3.5.1), donde cada uno de los tweets serán clasificados en alguno de los grupos ya definidos.

text	cluster
RT@lasillarota:#NadoSincronizado	3
¡No importa! Representaron a nuestra nación- un aplauso para ustedes campeonas @NuriaDiosdado - @kar	3
Ya fue la semifinal! La final con @Rommel_Pacheco es a las 4pm hora del Centro de México!	2
RT @record_mexico: El velocista de #MEX José Carlos Herrera reprochó la falta de apoyo: "En México todo es	1
RT @InvictosSomos: ¿Ganó su heat eliminatorio en 200m planos? ¡DAB! #MEX	1
RT @Lizzysa: Hace 16 años que #Mex no pasaba a finales en #NadoSincronizado en los Olímpicos. ¡Felicidade	3
@Rommel_Pacheco Éxito!!!#Rio2016 #MEX #Clavados	2
RT @InvictosSomos: #MEX. @Rommel_Pacheco y @Rorro_clavados avanzan a la semifinal del Trampolín de	2
RT @juanfutbol: #MEX al momento en #Rio2016: José Carlos Herrera correrá las semifinales de 200 metros r	1

FIGURA 3.26: Predicción algoritmo K-medias.

Para facilitar la visualización así como las recomendaciones personalizadas para los usuarios, se creo una base de datos en el DBMS PostgreSQL 9.6.3 con tres tablas en las que se cargaron las predicciones del algoritmo K-medias (Figura 3.27).

FIGURA 3.27: Modelo relacional: Predicción algoritmo K-medias.

En la tabla Palabra se encuentran las palabras que aparecen en los tweets, los atributos c1, c2, c3, c4 son las correlaciones que tiene la palabra con cada cluster y el atributo grupo es el número de cluster al que pertenece cada palabra. En la tabla Tweet sus atributos corresponden al tweet que se publico, el número de cluster al que pertenece y el usuario que lo publico, este último atributo será de utilidad para realizar recomendaciones personalizadas. La tabla TipoCluster tiene el número de cluster y el nombre del cluster (Atletismo, Clavados y Nado Sincronizado).

3.7.5.1. Cluster por palabra

Para determinar el cluster al que pertenece cada palabra se utilizó la matriz de correlación (Figura 3.28), seleccionando la columna de mayor grado de relación que hay entre la palabra y el cluster.

palabra text	c1 numeric	c2 numeric	c3 numeric	c4 numeric	cluster integer
#boxeo	0	.008873114	0	.693877551	4
#nadosincronizado	0	.203194321	.950850662	0	3
#rio	.274576271	.336291038	.168241966	.346938776	4
@karemachach	0	.010647737	.506616257	0	3
@nuriadiosdado	0	.007985803	.516068053	0	3
@rommelpacheco	0	.176574978	0	0	2
carlo	.996610169	.005323869	0	0	1
final	.003389831	.268855368	0.5463138	0	3
heat	.501694915	.007985803	0	0	1
herrera	.993220339	.003549246	0	.224489796	1
josé	.989830508	0	0	.224489796	1
lugar	.074576271	.036379769	0.31758034	0	3
medalla	0	.149068323	.028355388	.448979592	4
misael	0	.005323869	0	.979591837	4
trampolín	0	.139307897	0	0	2

FIGURA 3.28: Matriz de correlación por palabras.

A continuación, se muestran algunas de las palabras empleadas en los tweets de cada cluster.

En la figura 3.29 se observan las palabras utilizadas en el evento de atletismo que corresponden al cluster 1. En las que sobresalen el nombre del atleta que representó a México en esta disciplina: José Carlos Herrera.

palabra text	cluster integer	Nombre text
#atletismo	1	Atletismo
@invictossomo	1	Atletismo
carlo	1	Atletismo
eliminatorio	1	Atletismo
ganó	1	Atletismo
heat	1	Atletismo
herrera	1	Atletismo
josé	1	Atletismo
plano	1	Atletismo
velocista	1	Atletismo

FIGURA 3.29: Palabras agrupadas en el cluster de Atletismo.

La figura 3.30 muestra algunas de las palabras contenidas en el cluster 2 referentes a la disciplina de clavados, como el nombre del atleta mexicano Rommel Pacheco y el canal de televisión que dio más apoyo a dicha disciplina Fox Sport.

palabra text	cluster integer	Nombre text
#rioxfox	2	Clavados
#rommelpacheco	2	Clavados
@can	2	Clavados
@foxsposmx	2	Clavados
@rommelpacheco	2	Clavados
gran	2	Clavados
metro	2	Clavados
trampolin	2	Clavados
vamo	2	Clavados

FIGURA 3.30: Palabras agrupadas en el cluster de Clavados.

En la figura 3.31 se muestran ciertas palabras del cluster 3 para el evento de nado sincronizado, en las que resaltan las cuentas de las mexicanas Karen Machado y Nuria Diosdado así como las etiquetas de otros países participantes como Rusia.

palabra text	cluster integer	Nombre text
#nadosincronizado	3	Nado_sincronizado
#oro	3	Nado_sincronizado
#plata	3	Nado_sincronizado
#rus	3	Nado_sincronizado
@espnmx	3	Nado_sincronizado
@karemachach	3	Nado_sincronizado
@nuriadiosdado	3	Nado_sincronizado
dueto	3	Nado_sincronizado
final	3	Nado_sincronizado
lugar	3	Nado_sincronizado

FIGURA 3.31: Palabras agrupadas en el cluster de Nado sincronizado.

Y finalmente para el cluster número 4, (Figura 3.32) está conformado de aquellas palabras que son irrelevantes para el análisis. Debido a que son palabras que no están relacionadas con las tres disciplinas y su frecuencia es menor a 50. Gran número de estos tweets corresponden a mensajes con SPAM.

palabra text	cluster integer
conad	4
dar	4
delegación	4
derrotar	4
detall	4
dinero	4
eduaubdedubua	4
eduaubdedububd	4
egipcio	4
ello	4

FIGURA 3.32: Palabras irrelevantes agrupadas en el cluster cuatro.

3.7.5.2. Cluster por tweet

En las siguientes figuras, se presentan algunos de los tweets a manera de ejemplo junto con su respectivo cluster. En la figura 3.33 se muestran los tweets referentes a atletismo, en la figura 3.34 los tweets de clavados y en la figura 3.35 los tweets de nado sincronizado.

tweet text	Nombre text
"Clasificatoria a semifinales de 200m en #atletismo:20.28 segundos: Usain Bolt #JAM20.29 segundos: José Carlos Herrera #MEX<e...	Atletismo
"RT @CarolinaPadron: #atletismoSe clasifican a semis de los 200mts planos#pan @AlonsoEdward#mex José Herrera #crc @nerrybr...	Atletismo
"RT @cmrivass: #ASenRio | Clasificatoria a semifinales de 200m en #atletismo:20.28 segundos: Usain Bolt #JAM20.29 segundos: Jo...	Atletismo
"RT @tmhaidee: Clasificatoria a semifinales de 200m en #atletismo:20.28 segundos: Usain Bolt #JAM 20.29 segundos: José Carlos ...	Atletismo
"RT @tmhaidee: Clasificatoria a semifinales de 200m en #atletismo:20.28 segundos: Usain Bolt #JAM20.29 segundos: José Carlos H...	Atletismo
RT @90minutosMex: José Carlos Herrera #MEX termina en primer lugar de su hit en los 200m de #atletismo y avanza a la Semifina...	Atletismo

FIGURA 3.33: Tweets relacionados al Atletismo.

tweet text	Nombre text
RT @EstefanChidiac: ¡Éxito en la final de #Clavados @Rommel_Pacheco! Gracias @Rorro_clavados por tu esfuerzo- i...	Clavados
Rommel Pacheco va por medalla; avanza a final de trampolín tres metros #AlmomentoMX #Rio2016 #MEX #clavados...	Clavados
RT @LegionDeportes: Rommel Pacheco terminó 2do en #clavados y está en la final. A las 4:00 pm #MEX va por med...	Clavados
GobMx: RT OnceNoticiasTV: #Atentos Esta tarde- Rommel_Pacheco #MEX peleará por la medalla en #clavados tram...	Clavados
"RT @RaulOrvananos: #Rio2016xFOX #MEX Rommel Pacheco clasificó a la Final de #clavadosHOY a las 4PM va por el...	Clavados

FIGURA 3.34: Tweets referentes a la disciplina de Clavados.

tweet text	Nombre text
"RT @ESPNmx: ¡Ya ves al dueto #MEX en #nadosincronizado"	Nado_sincronizado
RT @angelasaju: Dueto nado sincronizado #MEX finaliza en lugar 11 en #Rio2016 @karemachach ...	Nado_sincronizado
La pareja de #MEX concluye en 11vo. Lugar en la final de #nadosincronizado en #Rio2016 https://...	Nado_sincronizado
Dueto nado sincronizado #MEX finaliza en lugar 11 en #Rio2016 @karemachach @NuriaDiosdado ...	Nado_sincronizado
RT laaficion: #laAficiónEnRío: Arranca la rutina de Nuria Diosdado y Karem Achach en la final de #n...	Nado_sincronizado

FIGURA 3.35: Tweets con contenido relativo al Nado sincronizado.

3.7.6. Interpretación

Para la implementación del recomendador se hizo uso del cluster por palabra y del cluster por tweet, que previamente se almacenaron en una base de datos para hacer consultas más eficientes. Adicionalmente, a la información del cluster por tweet se agregó un campo de usuarios correspondiente al nombre de las cuentas en twitter con la finalidad de realizar recomendaciones personalizadas.

El recomendador presenta información a un usuario con base en sus gustos, necesidades e intereses. Estas sugerencias están relacionadas al proceso de toma de decisiones, debido a que se tiene que elegir entre varias opciones de etiquetas, cuentas de usuarios y cuentas oficiales que provoquen un mayor interés a los usuarios a partir del contenido de sus tweets.

Con respecto a los resultados de la etapa de predicción de cluster por tweet (Sección 3.7.5.2), el tipo de recomendación se hace en relación al agrupamiento de los usuarios en alguno de los tres clusters. Por lo que se analiza el porcentaje de tweets para cada grupo.

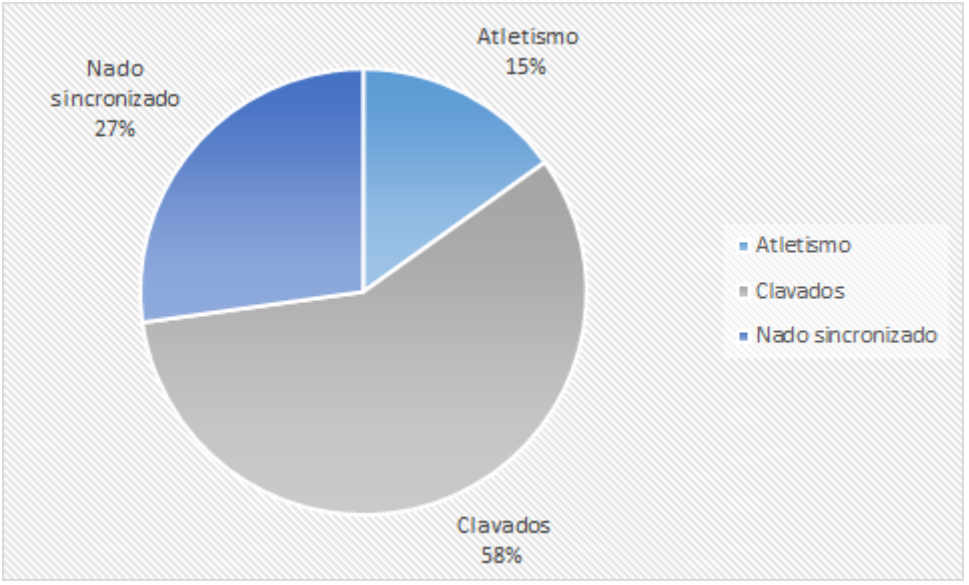


FIGURA 3.36: Porcentaje de tweets por disciplina.

En la figura 3.36 se muestra que el evento con más tweets fue Clavados con un 58 %, seguida por Nado sincronizado con 27 % y finalmente Atletismo con 15 %.

3.7.6.1. Sugerencias de usuarios, cuentas oficiales y etiquetas

Se eligió un usuario de cada cluster para llevar a cabo las recomendaciones. En el caso de atletismo se seleccionó al usuario "Oscar_CasPer05", para clavados a "EuroBarajas" y para nado sincronizado a "AriasViri". (Figura 3.37)

tweet text	cluster integer	usuario text
#Atletismo El mexicano Jose Carlos Herrera competirá mañana en las semifinales de los 200 mts planos. ...	1	Oscar_CasPer05
@Rommel_Pacheco Éxito!!!#Rio2016 #MEX #Clavados	2	EuroBarajas
"#MEX Diosdado y Achach quedan el el onceavo lugar en el #NadoSincronizado	3	AriasViri

FIGURA 3.37: Usuarios por cluster.

De acuerdo al comentario que realizó cada uno de ellos, se les ha recomendado seguir a ciertos usuarios que comparten ideas sobre el mismo tema, llevando a cabo los querys del Anexo II.

tweet text	usuario text
RT @LegionDeportes: El #MEX José Carlos Herrera ganó su hit de 200m. Podría enfrentarse al hombre más rápido Usain Bol...	jjos_a
"RT @InvictosSomos: ¿Ganó su heat eliminatorio en 200m planos? ¡DAB! #MEXAsí festejó José Carlos Herrera en #Rio2016. ...	ManeCervantes
"RT @record_mexico: El velocista de #MEX José Carlos Herrera reprochó la falta de apoyo: ""En México todo es futbol' <ed>...	Cecy_panbolera
"RT @InvictosSomos: ¿Ganó su heat eliminatorio en 200m planos? ¡DAB! #MEXAsí festejó José Carlos Herrera en #Rio2016. ...	DiegoVnt
"RT @record_mexico: El velocista de #MEX José Carlos Herrera reprochó la falta de apoyo: ""En México todo es futbol' <ed>...	Puentesnana
"RT @record_mexico: El velocista de #MEX José Carlos Herrera reprochó la falta de apoyo: ""En México todo es futbol' <ed>...	JavierFCPyS
RT @soymikehattan: Mañana José Carlos Herrera #MEX en la semifinal de los 200m. #Atletismo https://t.co/eMGG9b19d5	justjuampi
"RT @InvictosSomos: ¿Ganó su heat eliminatorio en 200m planos? ¡DAB! #MEXAsí festejó José Carlos Herrera en #Rio2016. ...	maugaraz6
"RT @InvictosSomos: ¿Ganó su heat eliminatorio en 200m planos? ¡DAB! #MEX..."	bryanhoil97
"RT @record_mexico: El velocista de #MEX José Carlos Herrera reprochó la falta de apoyo: ""En México todo es futbol"	C_iCarpizo

FIGURA 3.38: Recomendaciones de usuarios para los seguidores de atletismo.

tweet text	usuario text
RT @xeudeportes: ¡@Rommel_Pacheco está en la final de #clavados trampolín tres metros! ...	JoseLuisMar34
RT @InvictosSomos: #MEX. @Rommel_Pacheco y @Rorro_clavados avanzan a la semifinal d...	debanhmichel
RT @rgarciaochoa: La primer medalla de #MEX llega hoy. Rommel Pacheco se clasificó 2do a ...	jaimemorelos
RT @edgarzamoraMX: Termina la semifinal #Clavados trampolín 3 M- Rommel Pacheco clasifi...	Rosac96
"RT @vanehupp: ClavadosRommel Pacheco #MEX termina la semifinal trampolín 3m en 2º lu...	Roouusz

FIGURA 3.39: Recomendaciones de usuarios para los seguidores de clavados.

tweet text	usuario text
La pareja de #MEX concluye en 11vo. Lugar en la final de #nadosincronizado en #Rio2016 ...	elmercuriotam
RT @angelasaju: Dueto nado sincronizado #MEX finaliza en lugar 11 en #Rio2016 @karema...	espitiaeric
Dueto nado sincronizado #MEX finaliza en lugar 11 en #Rio2016 @karemachach @NuriaDios...	angelasaju
RT laaficion: #laAficiónEnRío: Arranca la rutina de Nuria Diosdado y Karem Achach en la final ...	elchanfle71
"#Rio2016 #NadoSincronizadoComienza la final de dueto- Nuria Diosdado y Karem Achach v...	AmorNatural10

FIGURA 3.40: Recomendaciones de usuarios para los seguidores de nado sincronizado.

De igual manera en las figuras 3.41, 3.42 y 3.43, se hacen las recomendaciones de las diez cuentas más sobresalientes de cada de una de las disciplinas. En la que se destacan los nombres de los atletas José Carlos Herrera (@mexicancharles), Rommel Pacheco (@rommelpacheco), Nuria Diosdado (@nuriadiosdado) y Karen Achach (@karenachach), medios de comunicación como revista de deportes (@invictosomos), periodicos (@recordmexico), programas y canales de televisión (@comexmasters, @legiondeportiva, @foxsportsmx y @clarosport).

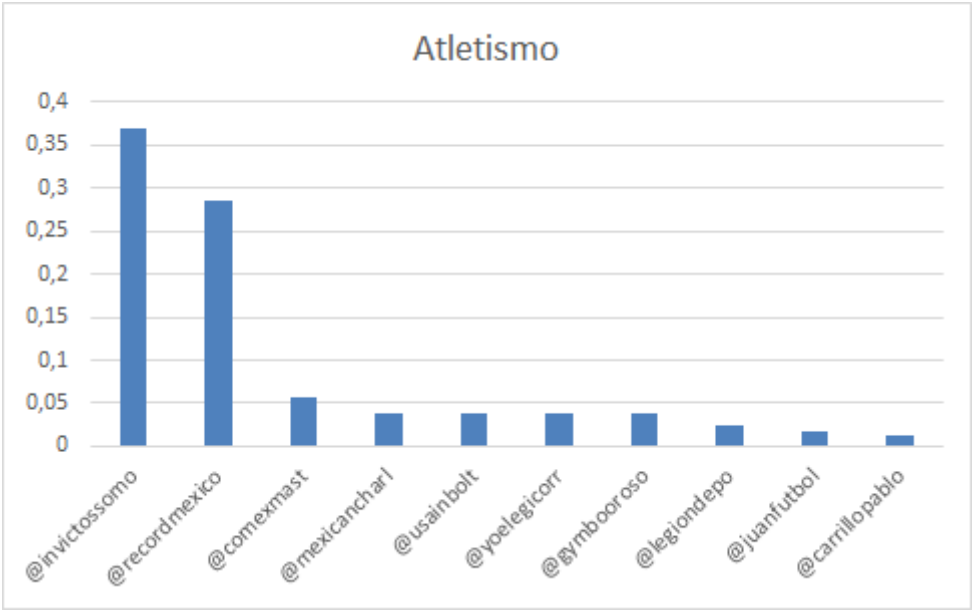


FIGURA 3.41: Cuentas oficiales para atletismo.

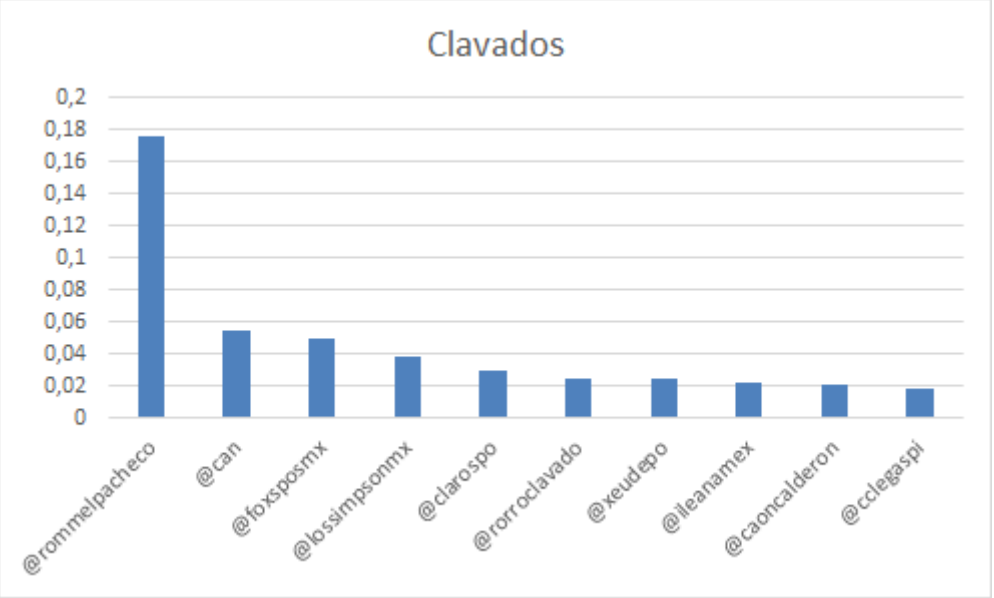


FIGURA 3.42: Cuentas oficiales para clavados.

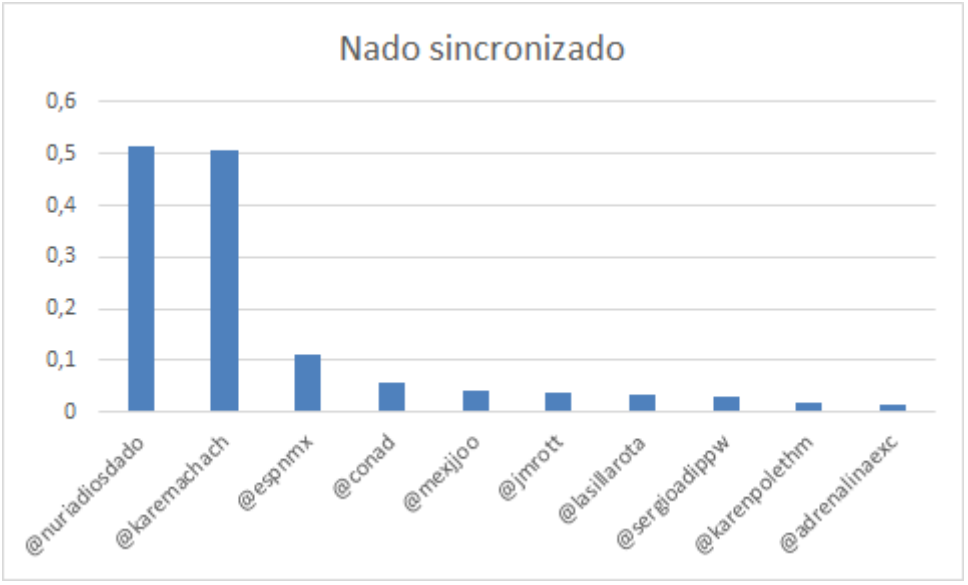


FIGURA 3.43: Cuentas oficiales para nado sincronizado.

En las figuras 3.44, 3.45 y 3.46, se muestran las diez etiquetas que tuvieron más menciones en los tweets referentes a cada cluster. Los usuarios podrán hacer uso de estas etiquetas en sus comentarios e identificar el contenido de su interés de forma rápida.

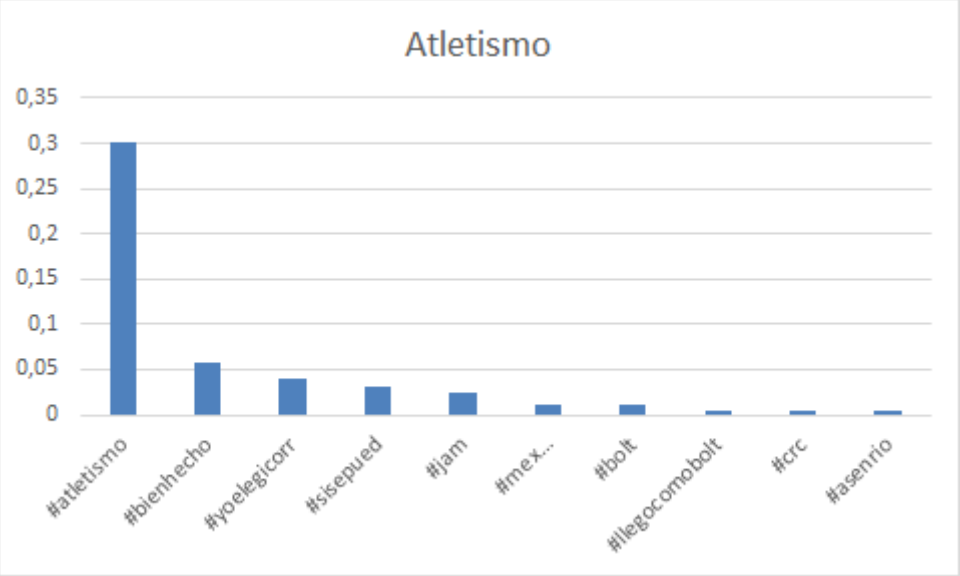


FIGURA 3.44: Etiquetas para atletismo.

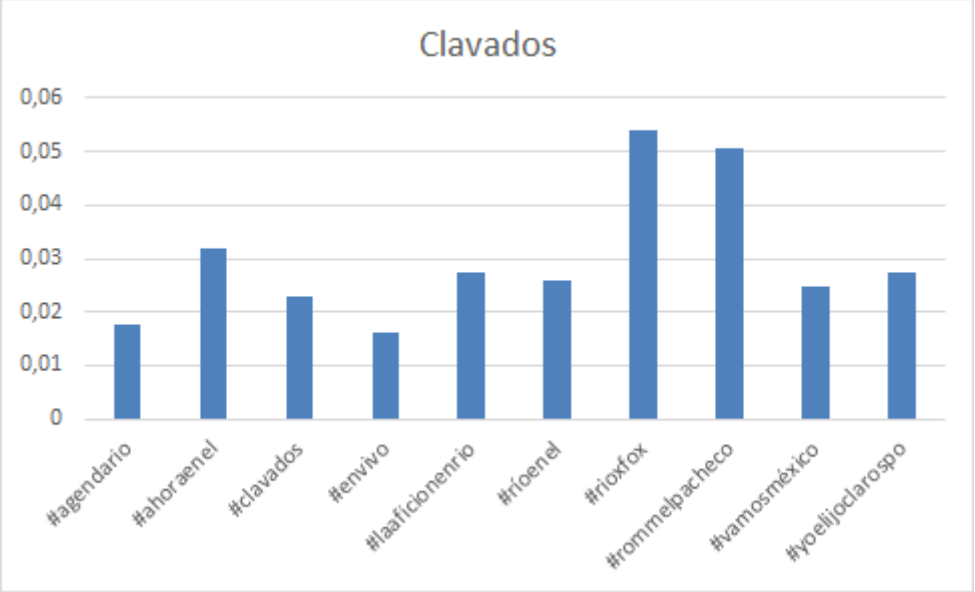


FIGURA 3.45: Etiquetas para clavados.

FIGURA 3.46: Etiquetas para nado sincronizado.

A partir del análisis de 10 las etiquetas y cuentas más mencionadas en los comentarios de Twitter como se muestran en las figuras anteriores, es posible facilitar y agilizar las recomendaciones con base en las opiniones de otros usuarios con intereses en común.

3.7.6.2. Medios de comunicación

En las gráficas siguientes se muestran los medios de comunicación que tuvieron más participación en los tweets. De la misma manera, se realizó el análisis de los principales medios de comunicación para cada cluster. El atletismo fue el deporte con menos apoyo por los usuarios de twitter, a pesar de la participación del atleta mexicano José Carlos, sólo cinco medios de comunicación se destacan por dar seguimiento a dicha disciplina, los cuales se aprecian en la figura 3.47. La revista digital Invictos somos (@InvictosSomos) atrajo la mayor atención de los usuarios por encima de programas de televisión, al tener más menciones en los tweets.

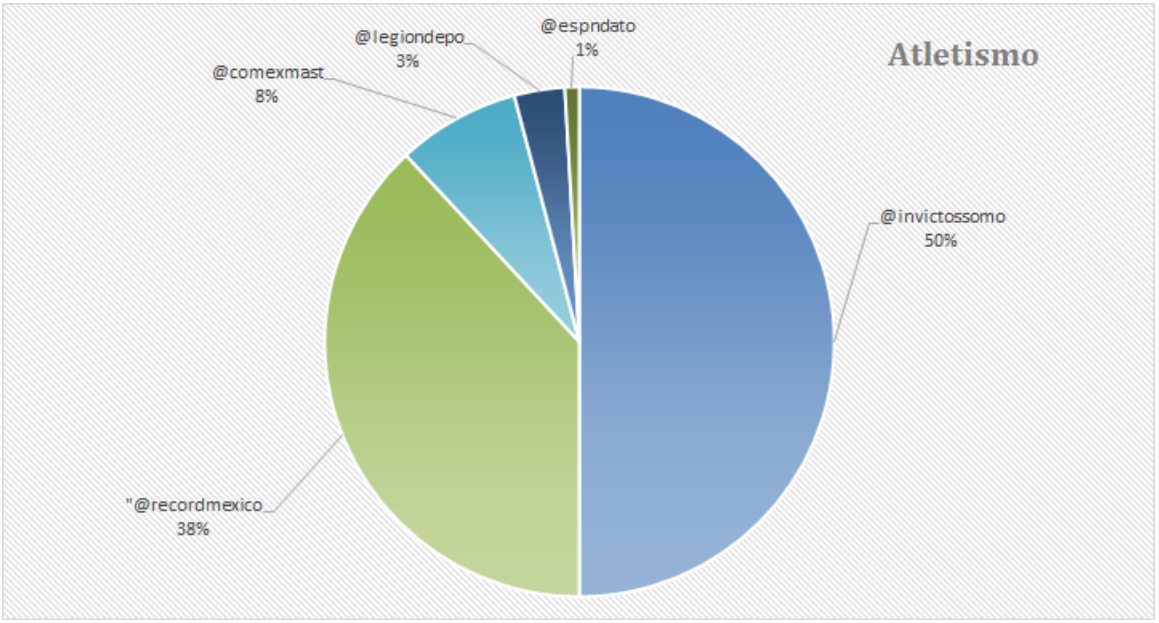


FIGURA 3.47: Principales medios de comunicación para atletismo.

Al usuario seleccionado anteriormente en la figura 3.39 que tuvo mayor interés por la disciplina de clavados se le sugiere seguir la cuenta oficial del programa deportivo Fox Sports México (@FOXSportsMX) ya que es la cuenta con más menciones en los tweets de este evento.

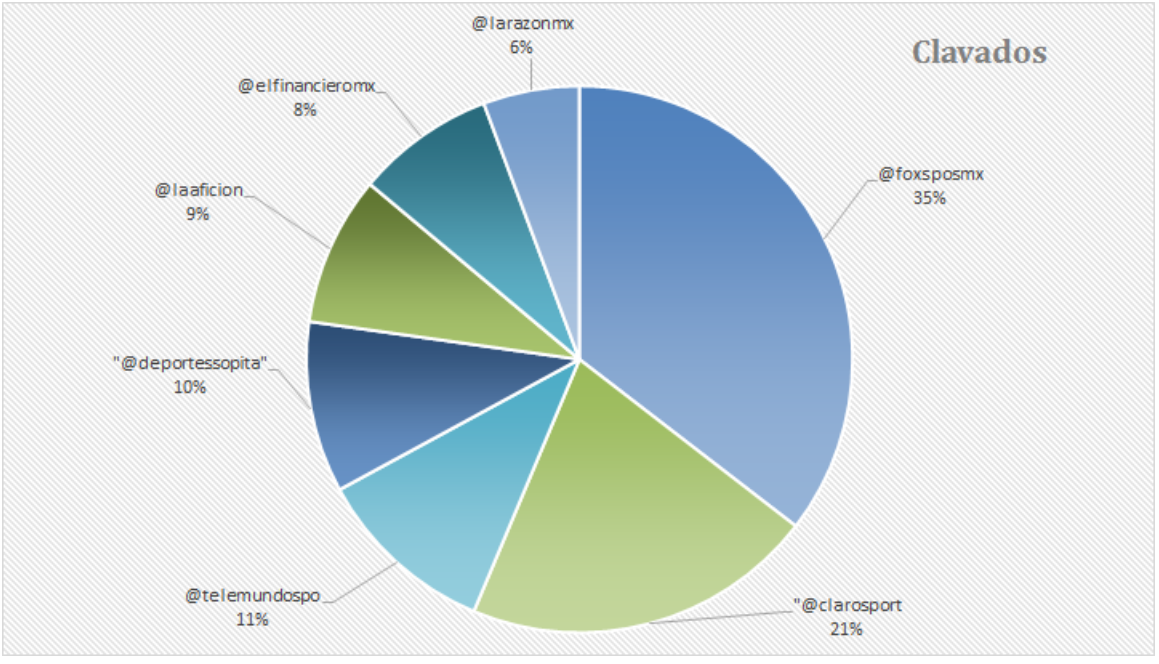


FIGURA 3.48: Principales medios de comunicación para clavados.

El programa de televisión ESPN(@ESPNmx) se recomienda a usuarios con preferencia al tema de nado sincronizado, ya que el 54 % de los usuarios de ese cluster dieron

seguimiento a noticias y resultados de la competencia a través de esta cuenta oficial. (Figura 3.49)

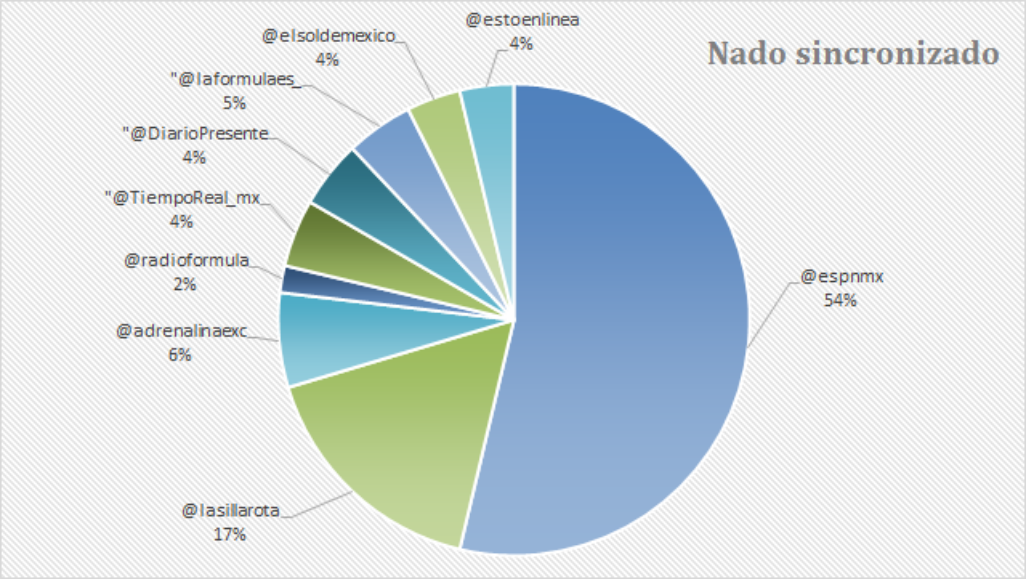


FIGURA 3.49: Principales medios de comunicación para nado sincronizado.

El análisis de los medios de comunicación que contaron con una mayor participación en los tweets durante los Juegos Olímpicos, tiene una importancia para los patrocinadores y organizadores de este evento, ya que a través de estos medios pueden captar más la atención hacia sus productos, servicios y atletas. Además de ser de utilidad entre los mismos medios de comunicación para identificar entre su competencia quien tuvo una mayor o menor popularidad y de esta manera mejorar su contenido para los próximos Juegos Olímpicos.

3.7.6.3. Etiquetas de países participantes

Durante los juegos olímpicos el uso de # con el nombre del país fue de las etiquetas con mayor uso en los comentarios. Como se puede ver en la gráfica de cada cluster (Figuras 3.50, 3.51 y 3.52), el #MEX fue el más utilizado por usuarios y cuentas oficiales en las que expresaron el apoyo hacia lo atletas mexicanos. Por lo tanto, para los usuarios seleccionados previamente (Figura 3.37) se les recomienda el uso de #MEX en sus tweets.

Otra etiqueta que se recomienda utilizar para atletismo es la que corresponde a Jamaica (#JAM), para Clavados y Nado sincronizado la de Rusia (#RUS).

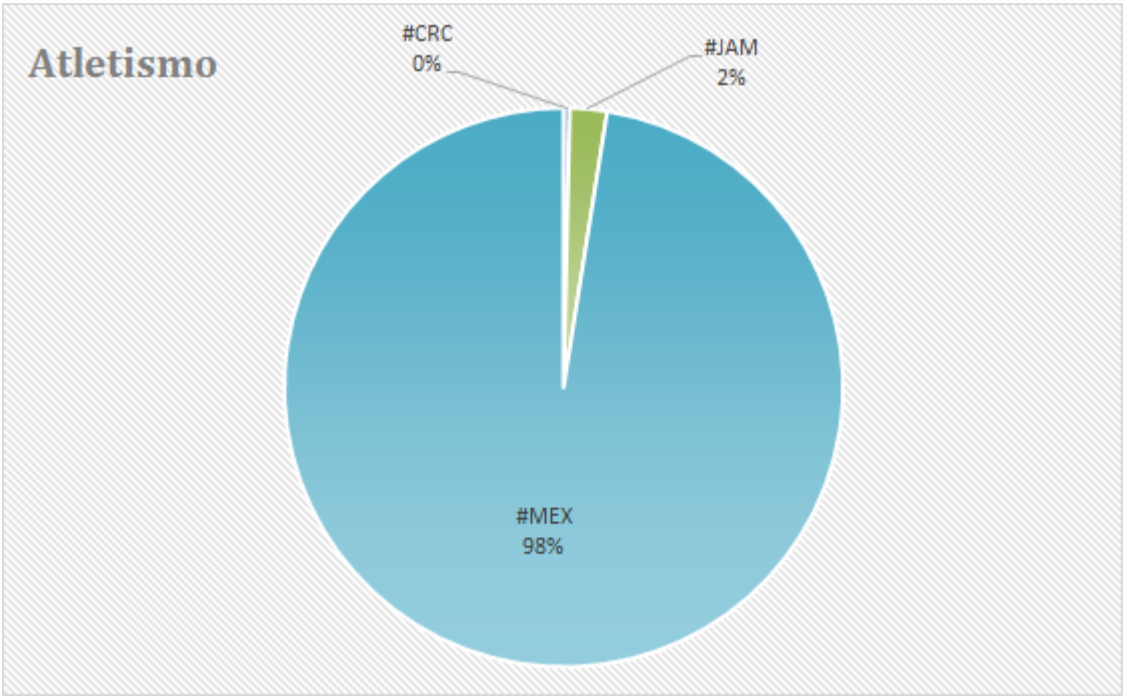


FIGURA 3.50: Países participantes en atletismo.

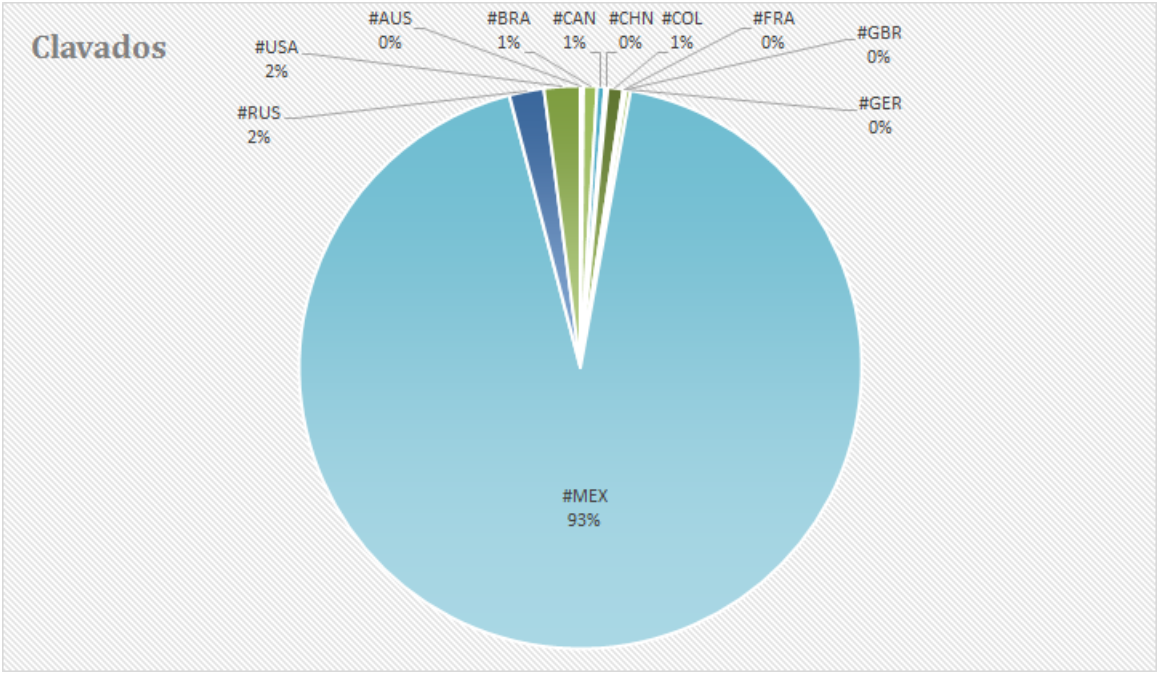


FIGURA 3.51: Países participantes en clavados.

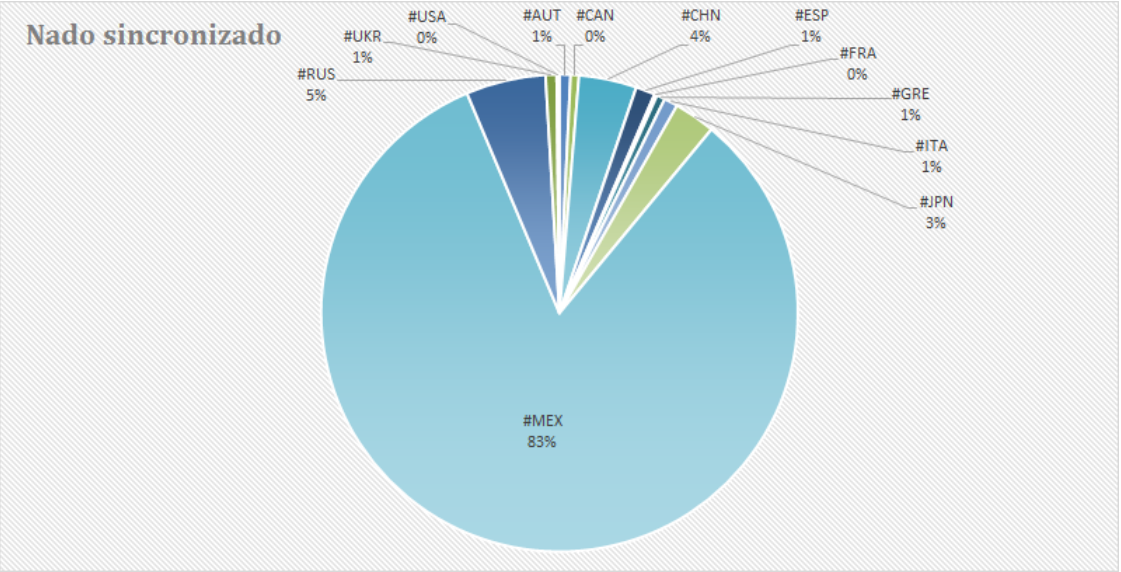


FIGURA 3.52: Países participantes en nado sincronizado.

Los datos que ofrecen las redes sociales ya sea pagando por ellos (Facebook) o gratuitamente (Twitter), y los algoritmos de aprendizaje que logran adaptarse de manera "automática" han permitido realizar diversos análisis en diferentes idiomas, por ejemplo: análisis de sentimientos y recomendaciones sobre eventos, personas, productos, lugares, servicios y marcas. Facilitando a investigadores y organizaciones información para la toma de decisiones.

Conclusiones

Actualmente, la capacidad de obtener información valiosa a partir de bases de datos robustas como son las redes sociales han impulsado el uso de las técnicas de Machine Learning (campo de estudio que se deriva de la Inteligencia artificial) en aplicaciones de análisis de texto, debido a que el reconocimiento de patrones para la solución de problemas está basado en el proceso de aprendizaje humano

Cabe señalar que el desarrollo de algoritmos de aprendizaje ayudó a valorar las ventajas y desventajas en la solución de problemas de clasificación y agrupamiento, por esta razón al analizar la naturaleza de los datos extraídos referentes a los Juegos Olímpicos se decidió utilizar los algoritmos de máquina de soporte de vectores y *k-means* debido a que permiten resolver problemas sobre el procesamiento de lenguaje natural.

En este sentido en el análisis de sentimientos se realizaron dos iteraciones. La primera iteración se entrenó un conjunto de 1000 tweets dando una precisión del 93 % al clasificar las palabras en dos categorías: positivas y negativas. En la segunda iteración se agregaron otros 1000 tweets al conjunto de entrenamiento, obteniendo una mejora en el porcentaje de precisión a un 95 %. Por lo tanto, entre más tweets de entrenamiento se le proporcione a la computadora mayor será el aprendizaje.

Debido a que el número total de términos es lo que dificulta el poder visualizar todas las palabras en las nubes, por esta razón se decidió que sólo aparecieran las palabras que se repitiera más de 50 veces en los comentarios. Se observó que en la matriz de confusión como en la nube de palabras predominan los comentarios positivos.

Por otro lado, el recomendador sugerido fue elaborado suministrando los 2000 tweets al algoritmo k-means dando un valor inicial de tres clusters ($k = 3$), por cada una de las disciplinas que participaron, los tres clusters resultantes contenían palabras no relacionadas a las disciplinas, así que se realizó una segunda iteración dando un valor de cuatro clusters ($k = 4$), un cluster por cada disciplina y el cuarto para detectar los comentarios no relacionados. Adicionalmente se utilizó la matriz de correlación de palabras que el mismo algoritmo proporcionó, este paso fue primordial para reconocer el

tipo de etiquetas (#) y cuentas (@) que correspondian a cada cluster. Finalmente, para agilizar la visualización de las preferencias de un cierto usuario mediante el lenguaje SQL fue necesario exportar los resultados de la agrupación por tweet añadiendo la variable de nombre de usuario y la correlación por palabras a una base de datos, de esta manera lograr la construcción del recomendador sugerido

Del análisis realizado, se concluye que optar por la participación de México en los Juegos Olímpicos el día 18 de Agosto de 2016 fue una buena decisión para realizar la Tesis, ya que como fuente de datos facilitó la recolección de los comentarios por la constante interacción que hubo sobre el tema en redes sociales. Además, las diversas disciplinas facilitaron la segmentación de las categorías y el análisis de las reacciones en cada una de ellas.

El lenguaje de programación R fue fundamental para la elaboración de ambas aplicaciones. Investigar las librerías y funciones que ofrece sobre Machine Learning y procesamiento de lenguaje natural representó un reto debido a que el enfoque visto durante la licenciatura fue diferente para el desarrollo de la metodología mencionada en el capítulo I de este trabajo. La API de Twitter fue una herramienta que igual requirió investigación para la conexión con R pero no causó una mayor dificultad por las ventajas que ofrece Twitter a los desarrolladores.

Después de la investigación y aplicaciones que se realizaron en este trabajo, se comprueba que al utilizar técnicas de Machine Learning en la clasificación de comentarios realizados en redes sociales referentes a las Juegos Olímpicos se detectaran patrones de datos que ayuden a la toma de decisiones.

Los conocimientos adquiridos a lo largo de la licenciatura de Matemáticas Aplicadas y Computación nos permitió abordar el tema de Machine Learning ya que engloba diversos tópicos sobre matemáticas, estadística y lenguajes de programación. Países como España y Estados Unidos son los de mayor auge en investigaciones de este tema mientras que en México las aportaciones todavía son mínimas. El análisis de datos es una de las múltiples aplicaciones relacionadas con nuestro perfil profesional lo que ha permitido que los egresados de nuestra licenciatura puedan profundizar en temas innovadores para la industria como *Machine Learning, Deep Learning, Big Data, Data Science* y *Business Intelligence*.

Una tesis conjunta necesita de una investigación más profunda en diferentes medios como libros, artículos, vídeos, cursos online, noticias y páginas web, la mayoría en el idioma inglés, debido a que tienes que estar al mismo nivel de conocimientos que tu compañero en el tema y tienes que saber algo adicional para compartirlo y enriquecer la tesis. Seleccionar información útil fue complicado debido a que es un tema extenso pero con

información escasa. A pesar de ser un trabajo conjunto no reduce las responsabilidades, se requiere de un doble esfuerzo de trabajo y compromiso para lograr el objetivo de una investigación que plasme las ideas de ambas personas dando valiosas aportaciones.

Anexos A

Anexo I: Código en R

Recopilación de datos

#Librerías

```
library(twitteR)
library(ROAuth)
```

#Keys and Access Tokens generados por la API de Twitter

```
consumer_key= ‘‘ONdQuziDw5JMilDmW4TDYCIOL’’
consumer_secret=‘‘QY11UUPZvetSzFxI0VH2kRYQfmee03vuYLysmR0yJ13AowSfRL’’
access_token=‘‘763811535219073024-EdKflnEhN3ya3X9jG83D2zXzYgoBTCs’’
access_secret=‘‘3E3cXMtcXxTi4NK2kwppAN9sVBLhwprSKLsbCwZzYvBe3’’
setup_twitter_oauth(consumer_key, consumer_secret, access_token, access_secret)
```

#Búsqueda y descarga de tweets

```
a.list<-searchTwitter(‘MEX’,n=2000)
a.df=twListToDF(a.list)
write.csv(a.df,file=‘C:/Users/Desktop/Rio2016.csv’,row.names=F)
```

Preparación de datos

```
library(httr)
```

#Cargar datos en R

```
tweets<-read.csv(‘‘C:/Users/Desktop/Rio2016.csv’’)
View(tweets)
```

```
library(NLP)
library(tm)
library(SnowballC)
```

```
library(httr)
library(ggplot2)

#Crear un corpus (base de datos)
corpus=Corpus(VectorSource(tweets$text))
#Cantidad de tweets en el corpus
length(corpus)
#Revisar el texto que contiene el corpus
content(corpus[[1]])
```

Limpieza de datos

```
#Convierte a minúsculas
corpus<-tm_map(corpus,tolower)

#Convierte el corpus a texto para visualizarlo
corpus <- tm_map(corpus, PlainTextDocument)

#Sustituye # por Hstg
removeh<-function(x) gsub(‘‘#’’,‘‘Hstg’’, x)
corpus<-tm_map(corpus,content_transformer(removeh))

#Sustituye @ por arro
removea<-function(x) gsub(‘‘@’’, ‘‘arro’’, x)
corpus<-tm_map(corpus,content_transformer(removea))

#Remueve URL
removeURL<-function(x) gsub(‘‘http [^[:space:]]*’’, ‘‘’’, x)
corpus<-tm_map(corpus,content_transformer(removeURL))

#Remueve RT
removeRT<-function(x) gsub(‘‘rt’’,‘‘’’",x)
corpus<-tm_map(corpus,content_transformer(removeRT))

#Remueve puntuación
removepun<-function(x) gsub(‘‘[[:punct:]]’’, ‘‘’’,x)
corpus<-tm_map(corpus,content_transformer(removepun))

#Remueve números
removenum<-function(x) gsub(‘‘[[:digit:]]’’, ‘‘’’,x)
corpus<-tm_map(corpus,content_transformer(removenum))

#Remueve las palabras más frecuentes en español
corpus<-tm_map(corpus,removeWords,c(stopwords(‘‘spanish’’)))
```

#Reduce las palabras a su raíz

```
corpus<-tm_map(corpus,stemDocument)
```

#Remueve espacios en blanco extras

```
corpus<-tm_map(corpus,stripWhitespace)
```

#Sustituye Hstg por #

```
removeh<-function(x) gsub(‘‘Hstg’’, ‘‘#’’, x)
corpus<-tm_map(corpus,content_transformer(removeh))
```

#Sustituye arro por @

```
removeh<-function(x) gsub(.ªrro", "@", x)
corpus<-tm_map(corpus,content_transformer(removeh)) content(corpus[[1]])
```

Transformación de datos

```
frecuencias <-DocumentTermMatrix(corpus)
frecuencias
sparse <- removeSparseTerms(frecuencias,0.995)
sparse
```

Análisis de sentimientos

Datos

```
tweetSparse <- as.data.frame(as.matrix(sparse),
row.names=as.character(tweets$text))
```

Integración

#Agregar la variable de sentimiento de los datos de entrenamiento

```
tweetSparse$sentiment <- as.factor(tweets$sentiment)
```

Reconocimiento de patrones

```
library(caTools)
set.seed(12)
```

#División en datos de entrenamiento y prueba

```
split<-sample.split(tweetSparse, SplitRatio= 0.8)
trainSparse = subset(tweetSparse, split==TRUE)
testSparse = subset(tweetSparse, split==FALSE)
```

Máquina de soporte de vectores

```
library(lattice)
library(ggplot2)
```

```
library(caret)
library(e1071)
library(kernlab)

SVM = svm(sentiment~.,data=trainSparse)
```

Validación

```
#Parámetros del modelo
summary(SVM)

#Porcentaje de aprendizaje del modelo
predecirSVM<-predict(SVM,newdata=testSparse)
confusionMatrix(predecirSVM,testSparse$sentiment)
```

Predicción

```
library(SparseM)
library(RTextTools)
#Cambiar Acronym por acronym
trace(''create_matrix'',edit=T)
#Ingresar datos
predictionData <- list(''RT @record_mexico: El velocista de #MEX José Carlos
Herrera reprochó la falta de apoyo:En México todo es futbol'' ,''Que orgullo
@NuriaDiosdado @karemachach lo hicieron excelente! Muchas felicidades.
#NadoSincronizado #Mex'')

predMatrix <- create_matrix(predictionData, originalMatrix=tweetSparse)
predMatrix <- as.data.frame(as.matrix(predMatrix), row.names=
as.character(predictionData))

predecirNuevoEjemplo<-predict(SVM,newdata =predMatrix)
predecirNuevoEjemplo
```

Interpretación

Nubes de palabras

```
library(RColorBrewer)
library(wordcloud)

##Palabras positivas
positivos<-subset(tweetSparse,tweetSparse$sentiment==1)
##eliminar variable sentiment (en data frame)
positivos$sentiment<-NULL
```

##Generar gráfica

```
palabrasPositivas<-as.data.frame(colSums(positivos))
palabrasPositivas$words<-row.names(palabrasPositivas)
colnames(palabrasPositivas)<-c(‘‘freq’’,‘‘word’’)
wordcloud(palabrasPositivas$word,palabrasPositivas$freq,random.order=FALSE,
colors=brewer.pal(8,‘‘Dark2’’),max.words=100)
```

##Palabras negativas

```
negativos<-subset(tweetSparse,tweetSparse$sentiment==-1)
```

##eliminar variable sentiment (en data frame)

```
negativos$sentiment<-NULL
```

##Generar gráfica

```
palabrasNegativas<-as.data.frame(colSums(negativos))
palabrasNegativas$words<-row.names(palabrasNegativas)
colnames(palabrasNegativas)<-c(‘‘freq’’,‘‘word’’)
wordcloud(palabrasNegativas$word,palabrasNegativas$freq,random.order=FALSE,
colors=brewer.pal(8,‘‘Dark2’’),max.words=100)
```

Recomendador

Integración

#Crear una matriz de documentos

```
tdm = TermDocumentMatrix(corpus)
```

#Convertir a una matriz

```
m = as.matrix(tdm)
```

#Remueve palabras poco frecuentes

```
wf = rowSums(m)
m1 = m[wf>quantile(wf,probs=0.9), ]
```

#Remueve columnas con ceros

```
m1 = m1[,colSums(m1)!=0]
```

#Matriz binaria

```
m1[m1 >1] = 1
```

Reconocimiento de patrones

```
library(NLP)
library(tm)
```

```
library(cluster)
library(FactoMineR)
library(RColorBrewer)
library(ggplot2)
```

K-medias

#Número de clusters
```
k = 4
```

Validación

##Gráfica de clusters
```
tweets_pam = pam(tweets_ca$row$coord[,1:2], k)
# Obtener clusters
clusters = tweets_pam$clustering
# Crear data frame
tweets_words_df = data.frame(words = rownames(m1),
dim1 = tweets_ca$row$coord[,1],dim2 = tweets_ca$row$coord[,2],
freq = rowSums(m1),cluster = as.factor(clusters))
#Definir el color
gbrew = brewer.pal(8, ‘‘Dark2’’)
gpal = rgb2hsv(col2rgb(gbrew))
#Colores: matiz, saturación,transparencia
gcols = rep(‘‘’’, k)
for (i in 1:k)
{
gcols[i] = hsv(gpal[1,i], gpal[2,i], gpal[3,i], alpha=0.6)
}
```

#Grafica de clusters
```
ggplot(subset(tweets_words_df, freq>50), aes(x=dim1, y=dim2, label=words))
+ geom_text(aes(size=freq, colour=cluster), alpha=1)
+scale_size_continuous(breaks=seq(20,80,by=10), range=c(6,10))
+scale_colour_manual(values=brewer.pal(6, ‘‘Dark2’’)) + labs(x=‘‘’’, y=‘‘’’)
```

##Medidas de calidad de cluster
Distancia inter cluster (máxima)
```
km$betweenss
```

#Distancia intra cluster (mínima)

```
km$tot.withinss
```

Predicción

```
km <- kmeans(t(tdm), k)
```

##Matriz de correlación por palabras

```
km$centers
```

##Cluster por tweet

```
km$cluster
```

Anexos B

Anexo II: Código en SQL

Modelo Relacional

Diccionario de Datos

Palabra				
Llave	Nombre_campo	Tipo de dato	Tamaño	Descripción
PK	Palabra	Text	50	Palabras que aparecen en los tweets
	c1	Numeric	(10,9)	Correlación con el cluster 1
	c2	Numeric	(10,9)	Correlación con el cluster 2
	c3	Numeric	(10,9)	Correlación con el cluster 3
	c4	Numeric	(10,9)	Correlación con el cluster 4
FK	Grupo	Smallint	2	Número de cluster (1, 2, 3 o 4)

TipoCluster				
Llave	Nombre_campo	Tipo de dato	Tamaño	Descripción
PK	Grupo	Smallint	2	Número de cluster (1, 2, 3 o 4)
	Nombre	Text	20	Nombre del cluster

Tweet				
Llave	Nombre_campo	Tipo de dato	Tamaño	Descripción
PK	Mensaje	Text	150	Tweet
FK	Grupo	Smallint	2	Número del cluster (1, 2, 3 o 4)
	Usuario	Text	50	Nombre del usuario que publicó el tweet

#Crear Base de Datos

```
CREATE DATABASE "Twitter"
```

#Tabla TipoCluster

```
CREATE TABLE public."TipoCluster"
(
"Grupo"smallint,
"Nombre"text,
PRIMARY KEY ("Grupo")
)
```

#Tabla Palabra

```
CREATE TABLE public."Palabra"
(
"Palabra"text,
c1 numeric(10, 9),
c2 numeric(10, 9),
c3 numeric(10, 9),
c4 numeric(10, 9),
"Grupo"smallint,
CONSTRAINT "Palabra_pkey"PRIMARY KEY ("Palabra"),
CONSTRAINT "PalabraCluster"FOREIGN KEY ("Grupo")
REFERENCES public."TipoCluster"("Grupo")
)
```

#Tabla Tweet

```
CREATE TABLE public."Tweet"
(
"Mensaje"text,
"Grupo"smallint,
"Üsuario"text,
CONSTRAINT "Tweet_pkey"PRIMARY KEY ("Mensaje"),
CONSTRAINT "TweetCluster"FOREIGN KEY ("Grupo")
REFERENCES public."TipoCluster"("Grupo")
)
```

#Copiar el archivo txt a Postgresql

```
COPY Palabra
```

```
FROM ‘C:/Temp/palabra_cluster.txt’ using delimiters ‘,’ ;
```

#Copiar el archivo txt a Postgresql

```
COPY Tweet
FROM ‘C:/Temp/tweet_cluster.txt’ using delimiters ‘,’ ;
```

Interpretación (SQL)

#Número de tweets por disciplina

```
SELECT tc.nombre, count(tc.nombre) as "NumeroTweets"
FROM public."TipoCluster" tc
INNER JOIN tweet t ON t.grupo=tc.grupo
GROUP BY tc.nombre
ORDER BY count(tc.nombre) desc;
```

#Sugerencias de usuarios, cuentas oficiales y etiquetas

```
SELECT mensaje as tweet, grupo as Cluster, usuario
FROM public."tweet"
WHERE usuario IN (‘Oscar_CasPer05’,‘EuroBarajas’, ‘AriasViri’)
ORDER BY cluster;
```

#Recomendación de usuarios para seguidores de atletismo

```
SELECT mensaje as tweet, usuario
FROM public."tweet"
WHERE grupo=1;
```

#Recomendación de usuarios para seguidores de clavados

```
SELECT mensaje as tweet, usuario
FROM public."tweet"
WHERE grupo=2;
```

#Recomendación de usuarios para seguidores de nado sincronizado

```
SELECT mensaje as tweet, usuario
FROM public."tweet"
WHERE grupo=3;
```

#Primeras diez cuentas oficiales para atletismo

```
SELECT p.palabra, p.c1
FROM public.‘‘Palabra’’p
WHERE p.grupo=1 AND p.palabra LIKE ‘@%’
```

```
ORDER BY p.c1 desc
LIMIT 10;

#Primeras diez cuentas oficiales para clavados
SELECT p.palabra, p.c2
FROM public.‘‘Palabra’’ p
WHERE p.grupo=2 AND p.palabra LIKE ‘@%’
ORDER BY p.c2 desc
LIMIT 10;

#Primeras diez cuentas oficiales para nado sincronizado
SELECT p.palabra, p.c3
FROM public.‘‘Palabra’’ p
WHERE p.grupo=3 AND p.palabra LIKE ‘@%’
ORDER BY p.c3 desc
LIMIT 10;

#Diez etiquetas más mencionadas para atletismo
SELECT p.palabra, p.c3
FROM public.‘‘Palabra’’ p
WHERE p.grupo=3 AND p.palabra LIKE ‘@%’
ORDER BY p.c3 desc
LIMIT 10;

#Diez etiquetas más mencionadas para clavados
SELECT p.palabra, p.c2
FROM public.‘‘Palabra’’ p
WHERE p.grupo=2 AND p.palabra LIKE ‘#%’
ORDER BY p.c2 desc
LIMIT 10;

#Diez etiquetas más mencionadas para nado sincronizado
SELECT p.palabra, p.c3
FROM public.‘‘Palabra’’ p
WHERE p.grupo=3 AND p.palabra LIKE ‘#%’
ORDER BY p.c3 desc
LIMIT 10;

#Principales medios de comunicación para atletismo.
SELECT count(t.grupo) AS ‘‘Menciones’’
FROM public.‘‘tweet’’ t
```

```
WHERE t.grupo=2 and t.mensaje LIKE ‘%@FOXSportsMX%’
GROUP BY t.grupo;
```

#Principales medios de comunicación para clavados

```
SELECT count(t.grupo) AS ‘‘Menciones’’ FROM public.‘‘tweet’’ t
WHERE t.grupo=3 and t.mensaje LIKE ‘%@ESPNmx%’
GROUP BY t.grupo;
```

#Principales medios de comunicación para nado sincronizado.

```
SELECT count(t.grupo) AS ‘‘Menciones’’
FROM public.‘‘tweet’’ t
WHERE t.grupo=3 and t.mensaje LIKE ‘%@ESPNmx%’
GROUP BY t.grupo;
```

#Etiquetas de países participantes para atletismo

```
SELECT p.palabra as .‘‘Etiqueta #’’
FROM public.‘‘Palabra’’ p
INNER JOIN public.‘‘tweet’’ t ON t.grupo=p.grupo
WHERE p.grupo=1 AND p.palabra LIKE ‘#%’ AND length(p.palabra)=4
GROUP BY p.palabra;
```

```
SELECT count(t.grupo) as ‘‘Menciones’’
FROM public.‘‘tweet’’ t
WHERE t.grupo=1 and t.mensaje LIKE ‘%#MEX%’
GROUP BY t.grupo;
```

#Etiquetas de países participantes para clavados

```
SELECT count(t.grupo) as ‘‘Menciones’’
FROM public.‘‘tweet’’ t
WHEREt.grupo=2 and t.mensaje LIKE ‘%#MEX%’
GROUP BY t.grupo;
```

#Etiquetas de países participantes para nado sincronizado

```
SELECT count(t.grupo) as ‘‘Menciones’’
FROM public.‘‘tweet’’t
WHERE t.grupo=3 and t.mensaje LIKE ‘%#MEX%’
GROUP BY t.grupo;
```

Bibliografía

1. Aubert, J. & Schomberg, R. (1986). *Inteligencia artificial*. (1ª ed). Madrid: Paraninfo.
2. Benitez, R. (2014). *Inteligencia artificial avanzada*. (1ª ed.). Barcelona: UOC.
3. Britos, P. (2005). *Minería de datos basada en sistemas inteligentes*. (1ª ed.). Buenos Aires: Nueva Libreria.
4. Coppin, B. (2004). *Artificial intelligence illuminated*. (1ª ed.). Boston: Jones and Bartlett Publishers.
5. Devore, J., Gonzalez Pozo, V. & Romo, J. (2003). *Probabilidad y estadistica para ingenieria y ciencias*. (1ª ed.). Mexico: Internacional Thomson Editores/Learning.
6. Fausett, L. (2006). *Fundamentals of neural networks*. (1ª ed.). Englewood Cliffs, NJ: Prentice-Hall.
7. Firebaugh, M. (1989). *Artificial Intelligence: A Knowledge-Based Approach*. (1ª ed.). Boston, Massachusetts: PWS-KENT.
8. Gironés Roig, J. (2013). *Algoritmos*. (1ª ed.). Barcelona: UOC.
9. Han, J. & Kamber, M. (2006). *Data mining: concepts and techniques*. (2ª ed.). San Francisco, California: Morgan Kaufmann.
10. Hernández Orallo, J., Ramírez Quintana, M. & Ferri Ramírez, C. (2008). *Introducción a la minería de datos*. (1ª ed.). Madrid: Pearson Prentice Hall.
11. Hilera González, J. & Martínez Hernando, V. (1995). *Redes neuronales artificiales. Fundamentos, modelos y aplicaciones*. (1 ª ed.). USA: RA-MA.
12. Isasi Viñuela, P. & Galván León, I. (2004). *Redes de neuronas artificiales*. (1ª ed.). Madrid: Pearson Educacion.
13. Kuri Morales, A. & Galaviz Casas, J. (2007). *Algoritmos genéticos*.(1ª ed.). México: Sociedad Mexicana de Inteligencia Artificial.
14. Lantz, B. (2015). *Machine learning with R*. (2ª ed.). Birmingham: Packt Publishing.

15. Marin Morales, R. & Palma Méndez, J. (2008). *Inteligencia artificial: Técnicas, métodos y aplicaciones.* (1ª ed.). Madrid: McGraw-Hill.

16. Martín del Brío, B. & Sanz Molina, A. (2002). *Redes Neuronales y Sistemas Difusos.* (2ª ed.). México: Alfaomega.

17. McAllister, J. (1991). *Inteligencia artificial y PROLOG en microordenadores* (Silvia Vernetti-Blina y Ángel Toribio González,trad.). (1ª ed.). Barcelona: MARCOMBO.

18. Miranda Raya, A. (2015). *Big intelligence.* (1ª ed.). Madrid: Fundación EOI.

19. Mitchell, T. (1997). *Machine Learning.* (1ª ed.). New York: McGraw-Hill.

20. Pajares, G. & De la cruz, J. (2010). *Aprendizaje automático. Un enfoque práctico.* (1ª ed). Madrid: Ra-Ma.

21. Rich, E., Knight, K., Gonzalez Calero, P. & Trescastro Bodega, F. (1998). *Inteligencia artificial.* (1ª ed.). Madrid: McGraw-Hill.

22. Russell, S. & Norving, P. (2003). *Artificial Intelligence. A modern Approach.* (2ª ed.). USA: Pearson Education.

23. Sánchez Alberca, A. (2014). *Bioestadística aplicada con R y RKTeaching.* (1ª ed.). Madrid: Creative commons.

24. Spiegel, M. & Schiller, J. (2014). *Probabilidad y estadistica.*(1ª ed.). Mexico: McGraw Hill.

25. Turban, E. (1992). *Expert Systems and Applied Artificial Intelligence.* (1ª ed.). USA: Macmillan Publishing Company.

26. Winston, W. (2005). *Investigación de operaciones.* (4ª ed.). México: CENGAGE learning Editores.

27. Witten, I., Frank, E., Hall, M. & Pal, C. (2011). *Data mining: practical machine learning tools and techniques.* (3ª ed.). USA: Morgan Kaufmann.

28. Wu, X. & Kumar, V. (2009). *The Top Ten Algorithms in Data Mining.* (1ª ed.). USA: Taylor Francis Group, LLC.

Artículos

29. Amirtha, T. (2014). *Why the R programming language is good for business.*[online]. Fast Company. Disponible en: https://tinyurl.com/y9d6le7p

30. Briega, R. (2015). *Machine Learning con Python.* [online]. Raul E. Lopez Briega: Matemáticas, análisis de datos y python. Disponible en: http://relopezbriega.github.io/blog/2015/10/10/machine-learning-con-python/

31. Chang, A. (2012). *R for Machine Learning.* [online]. MIT OpenCourseWare. Disponible en: https://tinyurl.com/y7op6mcv

32. Chen, E. (2011). *Choosing a Machine Learning Classifier.* [online]. Edwin Chen's Blog. Disponible en: http://blog.echen.me/2011/04/27/choosing-a-machine-learning-classifier/

33. Edwards, J. (2015). *Data Science: Future Or Fiction?.* [online]. Forbes Business. Disponible en: https://www.forbes.com/sites/teradata/2015/03/25/data-science-future-or-fiction/

34. Escudero, J. (2015). *Data Science: qué es y qué necesito para ser un científico de datos.* [online]. La métrica. Disponible en: http://lametrica.com/data-science-que-es-y-que-necesito-para-ser-un-cientifico-de-datos/

35. Guadián, C. (2016). *+100 herramientas para el análisis de redes sociales sna ars.* [online]. K-Government. Disponible en: http://www.k-government.com/2016/06/28/100-herramientas-analisis-redes-sna-ars/

36. Guerra, A. (2004). *Aprendizaje Automático: Árboles de Decisión.* [online]. México: Universidad Veracruzana, Facultad de Física e Inteligencia Artificial. Disponible en:https://tinyurl.com/yaol62jd

37. Guyon, I. *Data Mining History: The Invention of Support Vector Machines.* [online]. KDnuggets. Disponible en:https://tinyurl.com/yac55yl9

38. Lara, F. *Fundamentos de redes neuronales artificiales.* [online] Instituto de Investigaciones Sociales-UNAM. Disponible en: http://conceptos.sociales.unam.mx/conceptos_final/598trabajo.pdf

39. Mittal, A. (2017). *Introduction to Machine Learning.* [online]. eLearning Industry. Disponible en: https://elearningindustry.com/introduction-machine-learning

40. Murphy, K. (2014). *Machine Learning. A Probabilistic Perspective.* [online]. England. Disponible en: https://www.cs.ubc.ca/ murphyk/MLbook/pml-intro-22may12.pdf

41. Olson, R. (2015). *How Machine Learn (And You Win).* [online]. Harvard Business Review. Disponible en: https://hbr.org/2015/11/how-machines-learn-and-you-win

42. Patiño, M. (2015). *Algoritmo c45 - Arboles de Decisión - Documents.* [online]. Documents.mx. Disponible en: http://documents.mx/documents/algoritmo-c45-arboles-de-decision.html

43. Ramírez, A. (2016). *Las redes sociales en los Juegos Olímpicos.* [online]. The-emag.com. Disponible en: https://www.the-emag.com/theitprofile/2016/08/08/redes-sociales-juegos-olimpicos

44. Semana. (2016). *Así podrá vivir los Juegos Olímpicos en las redes sociales.* [online]. Semana. Disponible en: https://tinyurl.com/y8wc2z7g

45. Tendencias Digitales. (2012). *Las redes sociales compiten en los Juegos Olímpicos.* [online]. Tendencias digitales. Disponible en: http://tendenciasdigitales.com/las-redes-sociales-compiten-el-los-juegos-olimpicos/

46. Velasco, J. (2010). *Cinco herramientas para analizar los sentimientos de los tweets.* [online]. Hipertextual. Disponible en:https://tinyurl.com/yabse2vv

47. Villagra, A., Guzmán, A. & Pandolfi, D. (2009). *Análisis de medidas no-supervisadas de calidad en clusters obtenidos por K-means y Particle Swarm Optimization.* [online]. Argentina: Universidad de Palermo. Disponible en: https://dspace.palermo.edu:8443/xmlui/handle/10226/454

Cursos

48. Ng, A. (2011). *Aprendizaje Automático.* [online]. Coursera. Disponible en: https://tinyurl.com/oljwzey

49. Valveny, E., González, J. & Baldrich, R. (2017.). *Clasificación de imágenes: ¿cómo reconocer el contenido de una imagen?* [online]. Coursera. Disponible en: https://www.coursera.org/learn/clasificacion-imagenes/home/welcome

Videos

50. Gibaja Martíns, J. [Juan José Gibaja Martíns]. (2011). *Ejemplo del algoritmo k-means con R.* Disponible en: https://www.youtube.com/watch?v=SWOujaF7l2ot=178s

51. Lavrenko, V.[Victor Lavrenko]. (2014). *IAML5.12: Naive Bayes for spam detection.* Disponible en: https://www.youtube.com/watch?v=8aZNAmWKGfs

52. Martínez, J. [BD Guidance]. (2016). *Primer Taller de Análisis de Sentimiento en Twitter con R.* Disponible en: https://www.youtube.com/watch?v=nOIZnYLlPBot=146s

53. Meza Ruiz, I. [Ivan Vladimir Meza Ruiz]. (2014). *Máquinas de Soporte Vectorial.* Disponible en: https://www.youtube.com/watch?v=_nu_vY_UAnU

54. Winston, P. [MIT OpenCourseWare]. (2010). *Learning: Support Vector Machines.* Disponible en: https://www.youtube.com/watch?v=_PwhiWxHK8o

Sitios web

55. Bases Teóricas y Sistemas Biométricos. UNAM-Facultad de Ingeniería. http://redyseguridad.fi-p.unam.mx/proyectos/biometria/basesteoricas/reconocimiento.html

56. Clasificadores de patrones por funciones de distancia. http://profesores.fi-b.unam.mx/ana/APUNTES_RP/Capitulo3.pdf

57. Cluster for some short texts in R. https://www.linkedin.com/pulse/cluster-some-short-texts-r-beibei-kong

58. Machine Learning: What it is and why it matters. https://www.sas.com/en_us/insights/analytics/machine-learning.html

59. Mining twitter with R. https://sites.google.com/site/miningtwitter/questions/talking-about/given-users

60. Regresión Lineal como Técnica más Eficiente para le Previsión de la Demanda. https://tinyurl.com/y7uufah2

61. SVM tutorial: How to classify text in R. https://www.svm-tutorial.com/2014/11/svm-classify-text-r/

62. Text mining de Twitter usando R. http://www.webmining.cl/2012/07/text-mining-de-twitter-usando-r-parte-2/

63. Text mining example codes (tweets). https://tinyurl.com/y8d5zcu9

64. Text mining with R-Twitter data analysis. http://www.rdatamining.com/docs/text-mining-with-r

65. Top 10 Machine Learning Algorithms. http://www.datasciencecentral.com/profiles/blogs/top-10-machine-learning-algorithms

66. Twitter Developers. https://dev.twitter.com/

67. UCI. Machine Learning Repository. Center for Machine Learning and Intelligent Systems. http://archive.ics.uci.edu/ml/