UNIVERSIDAD NACIONAL AUTÓNOMA

DE MÉXICO

FACULTAD DE CIENCIAS

APLICACIÓN DE LAS MATEMÁTICAS EN EL ANÁLISIS DE GRANDES VOLÚMENES DE DATOS

REPORTE DE TRABAJO PROFESIONAL

QUE PARA OBTENER EL TÍTULO DE:

MATEMÁTICO

P R E S E N T A :

LUIS DONALDO ROMERO TAPIA

TUTORA

DRA. MARÍA DEL PILAR ALONSO REYES

Ciudad Universitaria, CD. MX., 2024

Datos del jurado

1.-Datos del alumno:

- Luis Donaldo Romero Tapia.
- 5528599618.
- Universidad Nacional Autónoma de México Facultad de Ciencias.
- Matemáticas.
- 314322724.

2.-Datos del tutor:

- Dra. María del Pilar Alonso Reyes.

3.-Datos del sinodal 1:

- Dra. Ursula Xiomara Iturrarán Viveros.

4.-Datos del sinodal 2:

- Dr. Fernando Baltazar Larios.

5.-Datos del sinodal 3:

- Dra. Amparo López Gaona.

6.-Datos del sinodal 4:

- Dra. Verónica Esther Arriola Ríos.

7.-Datos del trabajo escrito:

- APLICACIÓN DE LAS MATEMÁTICAS EN EL ANÁLISIS DE GRANDES VOLÚMENES DE DATOS.
- 100 p.
- 2024.

Dedicatoria

A mi familia: Mi refugio, los que me han ayudado mucho en todos los momentos, ya que han sido una fuente de amor incondicional y comprensión que ha sostenido mi espíritu en cada desafío. Gracias por proveerme no solo de lo necesario, sino de sabiduría, y por ser aquellos que me han permitido crecer como persona y como profesional.

A mis amigos: Compañeros de vida, ustedes han sido aquellos que han soportado mis caídas y celebrado mis logros. Cada conversación y cada momento compartido han sido una parte fundamental en la construcción de mi ser. En los momentos difíciles, su presencia ha sido un faro de esperanza y alegría.

A mis líderes: Visionarios que han trazado el sendero que ahora tránsito, su influencia me ha enseñado a mirar más allá de lo conocido y a desafiar mis propios límites. Las lecciones aprendidas han sido invaluables, pues me han enseñado a capturar las oportunidades con agilidad y a enfrentar los desafíos con una perspectiva estratégica.

A mis profesores: Ustedes han sido aquellos que construyeron mi conocimiento, los primeros en reconocer y fomentar mi curiosidad por las matemáticas. Gracias por mostrarme las distintas perspectivas de la vida y por confiar en mi potencial. El que creyeran en mí ha sido el cimiento sobre el cual he construido mi sueño y mi realidad.

Agradecimientos

Mi sincero reconocimiento a la Universidad Nacional Autónoma de México, institución que ha sido la cuna de mi formación académica y personal. Agradezco infinitamente el vasto conocimiento que me ha brindado, así como el uso de sus instalaciones, las cuales han sido escenario de innumerables horas de aprendizaje y reflexión, sobre todo durante las tardes que he pasado en los pasillos de la Facultad de Ciencias junto con mis compañeros, descubriendo las maravillas de las matemáticas. Todo lo anterior me ha dado momentos inolvidables que han moldeado mi carácter y mi entendimiento del mundo.

Extiendo mi gratitud a cada una de las instituciones que me han acogido en mi camino profesional. En ellas he tenido el privilegio de colaborar con colegas excepcionales cuya maestría y generosidad al compartir sus conocimientos han conformado un pilar importante en mi crecimiento como profesional. Estoy profundamente agradecido por la confianza depositada en mí y por las oportunidades que se me brindaron, las cuales me impulsaron a alcanzar horizontes que nunca imaginé posibles.

Un agradecimiento especial a los diversos cursos que han complementado mi educación, en particular a los ofrecidos por Coursera. Esta plataforma ha sido un elemento decisivo en el perfeccionamiento de mis habilidades y en la expansión de mi conocimiento, me han abierto puertas y presentado caminos que han enriquecido mi visión profesional. La educación en línea ha demostrado ser un recurso invaluable, y estoy eternamente agradecido por el acceso a tal riqueza de aprendizaje.

Resumen

Este trabajo se centra en la implementación práctica de las matemáticas y la ciencia de datos en una variedad de entornos profesionales, que profundiza en la comprensión de los procesos y esfuerzos subyacentes a la ejecución de proyectos en escenarios del mundo real; explorando cómo la teoría matemática y los métodos analíticos se aplican para resolver problemas complejos; integrando nuevos procesos al negocio; automatizando la toma de decisiones y generando valor continuo.

Se estudiará la importancia de las herramientas para la realización del trabajo de un científico de datos (las necesarias para los proyectos mostrados), con un énfasis en la identificación y manejo de valores atípicos. Por otra parte, se tendrá una introducción a la etapa de modelado que involucra una variedad de técnicas, como los modelos de la familia SARIMA y con enfoques en aprendizaje profundo, teniendo en cuenta si es necesario la predicción de valores reales o de categorías. Aquí el conocimiento técnico es fundamental, pero también la manera de aplicar estos saberes en el mundo real, considerando limitaciones como el entorno del desarrollo y el tiempo de entrega.

El primero de estos entornos profesionales se relaciona con la investigación de mercados, donde la precisión, relevancia y actualidad de los datos son fundamentales para tomar decisiones empresariales acertadas y establecer relaciones sólidas con las empresas que dependen de esta información, ya que permite la personalización de la venta de productos basada en una comprensión profunda de las necesidades y preferencias de los consumidores. En este sentido, el mantener la calidad de la información recolectada es primordial.

También se aborda la gestión de la logística para la satisfacción del cliente, el cual es un aspecto crucial para el éxito de cualquier empresa, ya que una cadena de suministro bien coordinada junto con una gestión eficiente de inventarios, contribuye a la disponibilidad oportuna de productos, reducción de costos operativos y mejora en la eficiencia, lo que beneficia tanto a la empresa como a los clientes. Es este sentido permite junto con ayuda de la información generada por la empresa tomar las decisiones correctas para el correcto funcionamiento de la logística en la empresa.

Por último, se estudiará lo que implica el mantenimiento predictivo en donde, mediante el análisis exhaustivo de grandes volúmenes de datos operativos a lo largo del tiempo, es posible identificar anticipadamente potenciales fallos, lo que permitirá adoptar medidas preventivas, evitar paradas no programadas, además de costos elevados relacionados con un proceso, lo cual mejorará la eficiencia y reducirá los tiempos muertos.

Contenido

Introducción

En este trabajo, se plantea ir más allá de la teoría académica. Se pretende conseguir la sumersión en el campo de la ciencia de datos, a través del desarrollo de soluciones a desafíos complejos que cumplan los objetivos de las empresas, con lo cual se demostrará cómo la ciencia de datos puede ser una fuerza transformadora en el mundo empresarial.

Este informe es el reflejo de un viaje profesional, donde las habilidades y conocimientos adquiridos no sólo han resuelto problemas teóricos, sino que han generado un valor real y mensurable en distintas empresas y situaciones. Es crucial señalar que, aunque los problemas discutidos en este documento están inspirados en situaciones reales de negocio, se han modificado por razones de confidencialidad. Estos escenarios, si bien no replican los desafíos exactos de las empresas, reflejan fielmente el tipo de problemas que estas enfrentan. La metodología aplicada aquí, por tanto, es profundamente relevante y directamente aplicable en contextos profesionales.

La aplicación de las matemáticas y la ciencia de datos en entornos empresariales exige un enfoque que va más allá del mero análisis numérico, ya que requiere la comprensión profunda de los problemas específicos de una empresa, la capacidad de adaptarse a entornos en constante cambio y la visión para implementar soluciones que puedan también anticipar necesidades futuras al destacar la aplicación práctica, que demuestra cómo la ciencia de datos, en manos de alguien experimentado, puede convertirse en una herramienta vital para la toma de decisiones estratégicas, la optimización de procesos y el impulso del crecimiento empresarial.

Se examinará el rol indispensable de los matemáticos en entornos multidisciplinarios a través de ejemplos concretos de proyectos realizados. Se evidencia que la integración de estas técnicas avanzadas de ciencia de datos, como los modelos SARIMAX y las redes neuronales, ayudan a transformar datos en estrategias y soluciones operativas eficientes.

El siguiente reporte sirve como un testimonio del impacto significativo que un científico de datos puede generar en el mundo empresarial relacionado a sectores involucrados con la investigación de mercados, logística y el mantenimiento predictivo al destacar la importancia de la ciencia de datos no solo como una disciplina académica, sino como una habilidad esencial en la toma de decisiones y la innovación en el mundo real.

Para ello se comenzará estableciendo un marco teórico robusto, donde se profundiza en la detección de valores atípicos, un elemento crucial en el análisis de datos. Métodos como la prueba de hipótesis, rango intercuartil, puntuación Z, percentiles, DBSCAN y el bosque de aislamiento son explorados para entender mejor cómo tratar con valores atípicos en conjuntos de datos.

Se avanza hacia la exploración de técnicas de modelado para regresión y clasificación, enfatizando la relevancia de los modelos SARIMAX en el análisis de series temporales y la contribución las CNN y LSTM son discutidos por su capacidad para manejar complejas estructuras de datos.

La atención se dirige luego a los aspectos clave durante el entrenamiento de modelos en ciencia de datos. Esta sección se enfoca en optimizar parámetros y elegir algoritmos apropiados para garantizar la efectividad y eficiencia de los modelos desarrollados.

El enfoque práctico del científico de datos en la gestión de proyectos se trata a continuación. Se destaca la importancia de la aplicación práctica de teorías en desafíos reales del mundo empresarial, resaltando las habilidades necesarias para llevar la teoría a la práctica.

La calidad de la información en la investigación de mercados se examina minuciosamente, destacando la detección de precios atípicos, control de datos inexistentes y errores en la recolección de datos. Además, se discute cómo los descubrimientos pueden retroalimentar y mejorar los procesos de investigación.

En lo que respecta a la logística y la satisfacción del cliente, se analiza cómo la generación eficiente de flujos de información y la adopción de nuevas tecnologías pueden optimizar la planificación logística y ofrecer soluciones ante situaciones de urgencia.

El papel de la inteligencia artificial en el mantenimiento predictivo se aborda, subrayando cómo puede ser utilizada para mejorar proactivamente la gestión de infraestructuras y sistemas.

Finalmente, se delinea un camino para la implementación de soluciones de ciencia de datos en ambientes productivos. Este capítulo cubre desde la conceptualización hasta el despliegue operativo, enfatizando la importancia de integrar soluciones de manera efectiva en entornos operativos reales.

Objetivos

El propósito general es poder aplicar y demostrar la eficacia de las técnicas de la ciencia de datos en entornos empresariales para transformar datos en decisiones estratégicas, optimizar procesos y fomentar la innovación tecnológica. De manera específica, se logrará con los siguientes objetivos:

1. Aplicar y adaptar técnicas de ciencia de datos en contextos empresariales:

   Implementar y adaptar teorías matemáticas y métodos de ciencia de datos en sectores relacionados con la investigación de mercados, logística y mantenimiento predictivo.

2. Transformar datos en decisiones y estrategias empresariales:

   Utilizar el análisis de datos para mejorar las decisiones empresariales al transformar la información en estrategias operativas que permitan resoluciones informadas.

3. Desarrollar el manejo efectivo de proyectos de ciencia de datos:

   Mostrar la forma de gestionar proyectos de ciencia de datos, abarcando desde la identificación y comprensión del problema hasta la implementación de soluciones.

4. Explorar el impacto de la ciencia de datos en distintos sectores:

   Reflexionar sobre la manera en la que experiencia profesional en ciencia de datos podría influir en distintos sectores empresariales.

5. Fomentar la adopción de tecnologías en el procesamiento y análisis de datos:

   Promover el uso de tecnologías avanzadas como Python y BigQuery de Google Cloud para el análisis de información, destacando la importancia de estas tecnologías en la mejora del procesamiento y análisis de datos.

Inventario de recursos y herramientas utilizadas en el análisis

Lo siguiente es un inventario de los recursos utilizados con el propósito de tener una visión clara y completa de las herramientas metodológicas y tecnológicas para el desarrollo del presente trabajo.

1.- Conjuntos de datos:

- Datos "Global Food Prices" de Kaggle (Boysen, 2017), como su nombre lo indica, contienen precios de productos globales, incluyendo el país, mercado, precio en moneda local y el mes de registro.
- Datos de la competencia "M5 Forecasting – Accuracy" de Kaggle (Howard, Makridakis & Vangelis, 2020), que son datos en ventas jerárquicos de Walmart.
- Datos "House Sales in King County, USA" de Kaggle (Harlfoxem, 2016), que incluyen precios de casas del condado de King como precio en dólares, número de cuartos, año de renovación, etc.
- Datos "PM_dataset" de Kaggle (Reddy, 2020), que incluyen un conjunto de sensores y configuraciones simuladas de motores de turbinas de gas de aviones.

2.- Herramientas y tecnologías:

- Python: Se usó el lenguaje Python 3.9.13 de programación principal para el análisis de datos y el modelado, las librerías utilizadas se encontrarán en el archivo de texto llamado "requirements"[1], el cual está alojado en un repositorio de código de Zenodo,

[1] El objetivo de este archivo es documentar los requisitos necesarios para llevar a cabo un proceso, en este caso contiene las librerías de Python que se usaron en el desarrollo de los programas presentados en el presente escrito.

creado con el fin de guardar el código utilizado en el presente trabajo. (Romero, 2023).

- BigQuery de Google Cloud para el estudio de adopción de nuevas tecnologías en el procesamiento de datos.

3.- Modelos de análisis y aprendizaje automático:

- Modelos SARIMAX, Redes Neuronales Convolucionales (CNN), y Redes de Memoria a Corto y Largo Plazo (LSTM) utilizados para la predicción y análisis de datos.
- Métodos para transformación y preparación de datos para estos modelos, métodos para la optimización de modelos en función de sus parámetros y criterios específicos del problema a tratar.

4.- Desarrollo y gestión de software:

- El proyecto se realizó por módulos para facilitar la mantenibilidad y colaboración, en donde se tratará la creación de módulos, archivo de configuración, y generación de logs como parte del proceso.

1. Marco Teórico

Para que el trabajo de un científico de datos se pueda llevar a cabo es esencial contar con las herramientas adecuadas. Una tarea clave en este proceso es identificar y manejar los valores atípicos, ya que pueden cambiar la forma en que se ve y entienden los resultados de un análisis, aunque esto es sólo una pequeña parte del tratamiento de los datos, se le dará especial énfasis en esta parte.

Una vez que se han tratado los datos, están listos para ser analizados o para entrar en la fase de modelado. En la etapa de modelado, hay muchas técnicas a disposición, desde métodos relacionados con los modelos de la familia de SARIMA, hasta enfoques relacionados con el aprendizaje profundo. Esto es dependiendo del problema que se está tratando, ya sea predecir valores (regresión) o categorizar datos (clasificación). Es donde el conocimiento técnico adquirido se pone a prueba para encontrar la mejor solución.

Sin embargo, hay que tener en cuenta que la teoría no es todo. Cuando se aplican estos métodos en situaciones reales, se enfrenta a desafíos como limitaciones de presupuesto, tiempos de entrega y el tipo de ambiente donde se pueden desplegar los procesos. Un buen científico de datos no solo tiene un dominio de la teoría, sino también tiene que conocer cómo aplicar estos conocimientos en el mundo real.

1.1. Manejo de valores atípicos en ciencia de datos

En esta sección se aborda un aspecto crucial que es la detección de valores atípicos. Estos son datos que se desvían significativamente de lo que se espera, y su presencia puede tener varias implicaciones, ya que si no se manejan correctamente pueden distorsionar los resultados, conduciendo a interpretaciones erróneas.

El primer objetivo al identificar valores atípicos es entender su origen, que puede deberse a distintas razones, como son:

1. Errores al momento de recolectar datos.

2. Falla del instrumento de medición.
3. Consecuencia de una medición natural.

Esta comprensión puede abrir puertas a análisis más profundos o incluso indicar cómo mejorar los métodos de recolección y tratamiento de datos.

Por lo tanto, es esencial considerar cómo los valores atípicos afectan los modelos que se aplican. Es fundamental decidir cómo detectarlos y tratarlos antes de la fase de modelado. A continuación, se expondrán distintos métodos para detectar y tratar valores atípicos.

1.1.1. Prueba de hipótesis – prueba de Grubbs

La prueba de hipótesis es una técnica estadística utilizada para evaluar si una hipótesis sobre un conjunto de datos es válida. En el contexto de la prueba de Grubbs se utiliza para determinar si existe un valor atípico, en donde la hipótesis nula en esta prueba supone que no hay valores atípicos, mientras que la hipótesis alternativa sugiere que sí los hay. Para ello se compara el estadístico de Grubbs con respecto al valor crítico y si es mayor se concluye que hay un valor atípico en los datos (Grubbs's test, s.f.).

$$H_0: No\ hay\ valores\ atípicos\ en\ el\ conjunto\ de\ datos$$

$$H_1: Hay\ exactamente\ un\ outlier\ en\ los\ datos$$

$$G_{calculada} = \frac{\text{máx } |X_i - \bar{X}|}{SD}$$

$\bar{X}\ y\ SD$ son la media y la desviación estándar respectivamente de los datos.

$$G_{crítica} = \left(\frac{\mathrm{N}-1}{\sqrt{N}}\right)\sqrt{\frac{\left(t_{\frac{\alpha}{2N},N-2}\right)^2}{N-2+\left(t_{\frac{\alpha}{2N},N-2}\right)^2}}$$

Si $G_{calculada}$ es más grande que, $G_{crítica}$ entonces se puede rechazar la hipótesis nula, donde $t_{\frac{\alpha}{2N},N-2}$ es un valor crítico superior de la distribución de T-Student con N-2 grados de libertad y un nivel de significancia de $\frac{\alpha}{2N}$.

1.1.2. Método rango intercuartil

Una forma de detectar valores atípicos es con el uso del rango intercuartil (iqr). Para entender el concepto, primero se ordenan las observaciones de menor a mayor. Así, se define el primer cuartil como la mediana de la mitad inferior de los datos q_1, la mediana de todos los datos es conocida como el segundo cuartil y se denotará como q_2, y el tercer cuartil es la mediana de la mitad superior q_3. El rango intercuartil, obtenido como la diferencia entre $q_3\ y\ q_1$, es importante ya que su cálculo no es afectado por valores atípicos. Teniendo en cuenta lo anterior, cualquier observación que se sitúe a más o menos de 1.5 veces el iqr del q_1 y del q_3 respectivamente se considera como un valor atípico (Devore, Berk & Carlton, 2021). Matemáticamente se diría que *x* es un valor atípico si:

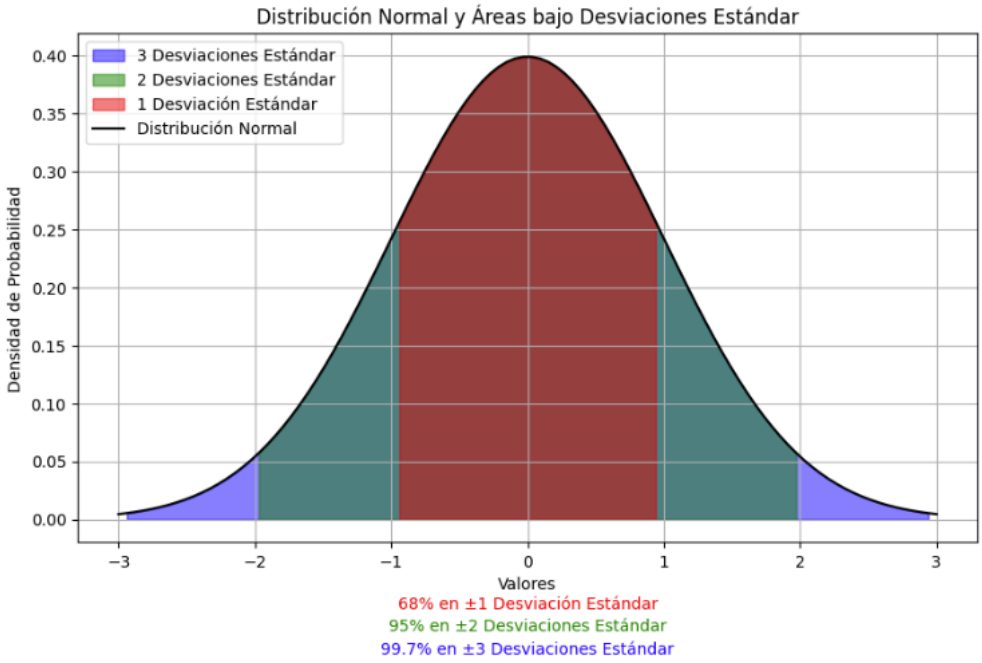
$$x < q_1 - 1.5 * iqr \quad \text{ó} \quad x > q_3 + 1.5 * iqr$$

1.1.3. Método de puntuación Z

La desviación estándar es una medida que ayuda a entender cuán dispersos están los datos en torno a su media. Para detectar valores atípicos utilizando esta medida, primero se calcula la media y la desviación estándar de los datos, luego se multiplica la desviación estándar por tres y se suma o se resta este valor a la media para obtener el límite superior e inferior respectivamente. Cualquier punto de datos que se encuentre por encima del límite superior o por debajo del límite inferior se considera un potencial valor atípico (Moore, McCabe & Craig, 2014). Sin embargo, es esencial recordar que esta regla se basa en la suposición de que los datos siguen una distribución normal y puede no ser adecuada para todas las situaciones.

Esta regla ayuda a destacar puntos anómalos o errores en los datos. Es un enfoque rápido y sencillo, pero es importante aplicarlo con cuidado y considerar otras técnicas cuando los datos no siguen una distribución normal.

Figura 1. Área comprendida por desviaciones estándar en la distribución normal

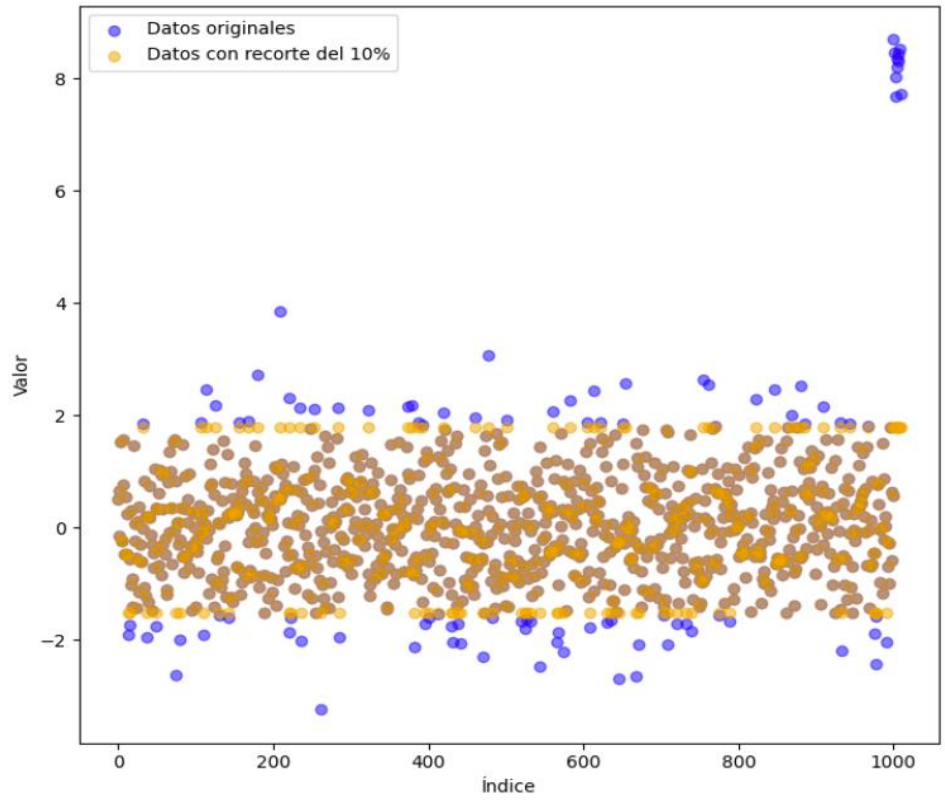


Visualización del área acumulada dentro de múltiples rangos de desviación estándar.

1.1.5. Método de percentiles

Este método es una técnica utilizada para tratar con valores atípicos en un conjunto de datos. Se puede aplicar un recorte del 1% superior e inferior de los datos, es decir, cualquier valor que esté por encima del percentil 99 se eliminará (Moore, McCabe & Craig, 2014) o se reemplazaría por el valor del percentil 99, en este caso se llama el método de *winsorización* (Winsorizing, s.f.), lo mismo con el percentil 1. Esta técnica es útil cuando se quiere reducir el impacto de los valores atípicos en un análisis estadístico.

Figura 2. Valores atípicos detectados por percentiles



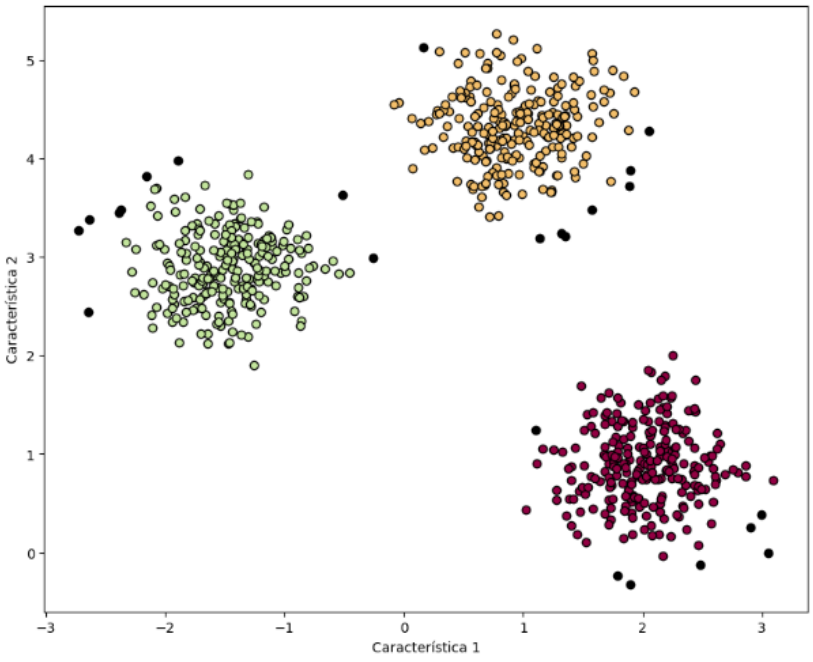
Representación visual por índice de la detección y del recorte de datos simulados con distribuciones normales.

1.1.6. Método DBSCAN

DBSCAN es un algoritmo de agrupamiento que identifica grupos de puntos de datos basándose en su densidad. Esto significa que puede encontrar clústeres de formas arbitrarias y es resistente a la presencia de valores atípicos.

El funcionamiento de DBSCAN se basa en la noción de "densidad alcanzable". El algoritmo comienza seleccionando un punto y explorando su vecindario en busca puntos adicionales en el vecindario de forma recursiva, construyendo un clúster si encuentra suficientes puntos cercanos dado un umbral predefinido. Los puntos que no se pueden conectar a ningún clúster se consideran valores atípicos y se consideran en una categoría separada (Ester, Kriegel, Sander & Xu, 1996).

Figura 3. Valores atípicos detectados por DBSCAN



Representación de clústeres encontrados por DBSCAN en datos simulados con la función make_blobs[2], la cual genera clústeres de n dimensiones.

1.1.7. Bosque de aislamiento

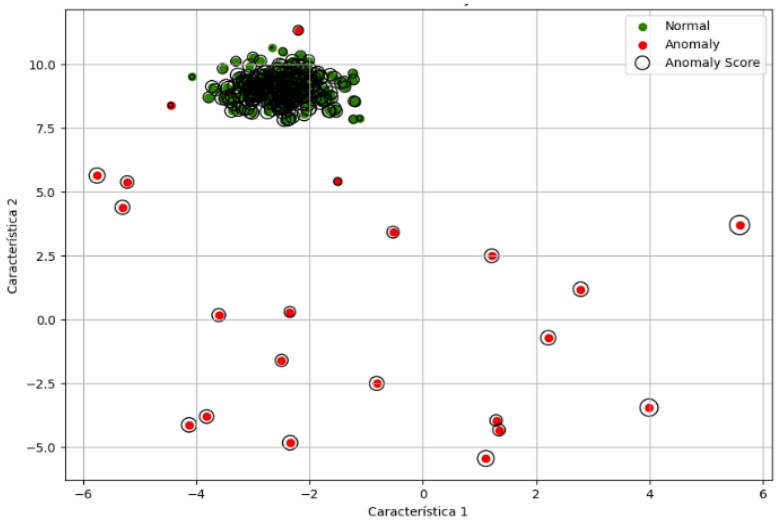
Los bosque de aislamiento son una técnica eficaz para detectar valores atípicos en conjuntos de datos, debido a su enfoque único en la separación de valores normales de los valores atípicos. El algoritmo construye una serie de árboles de decisión aislados, lo que significa que cada árbol se entrena de manera independiente. Luego se selecciona aleatoriamente un atributo y un umbral en cada paso, donde la idea fundamental es que los valores atípicos requieren menos divisiones para ser aislados, es decir, terminan siendo ubicados más cerca de la raíz del árbol en comparación con los valores normales (Liu, Ting, & Zhou, 2008).

Cuando se evalúa la profundidad en la que se encuentra un punto en los árboles, se puede medir cuán inusual es ese punto en el conjunto de datos. Una de las ventajas de este algoritmo es su capacidad para manejar conjuntos de datos de alta dimensionalidad y su enfoque

[2] La función make_blobs es de la librería de sklearn en Python, la cual tiene como objetivo generar conjuntos de datos de n dimensiones.

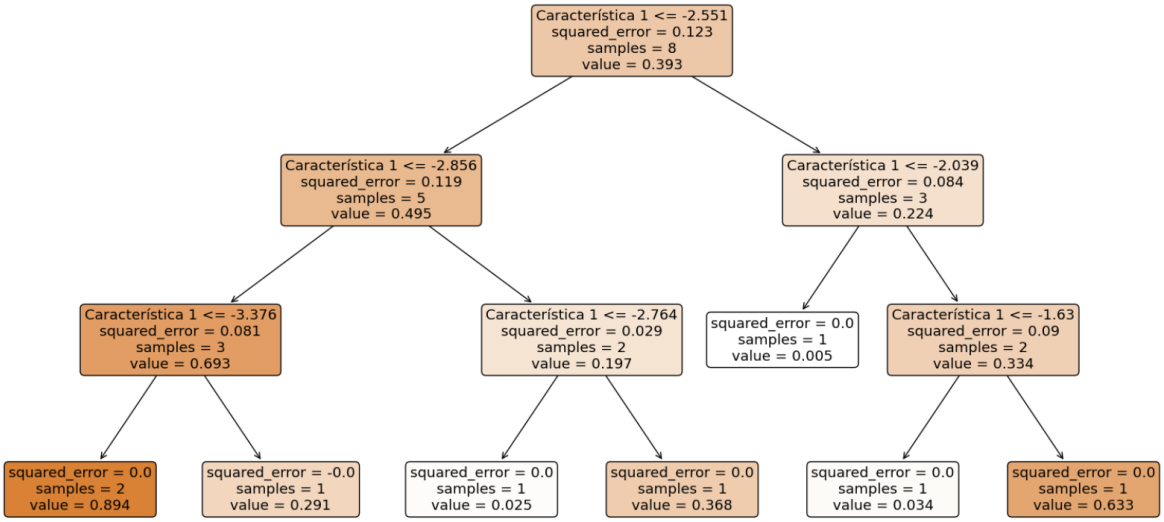
aleatorio de la selección de atributos y umbrales en cada árbol contribuye a su robustez y capacidad para generalizar a diferentes tipos de datos.

Figura 4. Detección de valores atípicos con el método "bosque de aislamiento"



Representación de valores anómalos encontrados con el bosque de aislamiento en datos simulados con la función make_blobs, solo se generó un clúster.

Figura 5. Árbol del "bosque de aislamiento"



Representación de un árbol del bosque de aislamiento que logra encontrar valores atípicos en datos generados con la función make_blobs, en donde se agregan valores atípicos con ayuda de una distribución uniforme.

1.2. Técnicas de modelado para la regresión y clasificación

El modelado de series de tiempo representa un desafío complejo en el campo de la ciencia de datos, especialmente del lado de la regresión y clasificación, además de tener una gran relevancia en diversos sectores como finanzas, comercio minoristas, entre otros. Es en este marco, los modelos multivariados han probado ser de gran utilidad, específicamente, el desarrollo de modelos desde AR hasta SARIMAX, ha sido fundamental para entender y aplicar estas técnicas en el análisis de series temporales. SARIMAX, en particular, ha sido la piedra angular de muchos análisis por su capacidad de incorporar múltiples variables exógenas para mejorar el modelado, las siguientes secciones se basarán en el artículo "Time Series Forecasting with ARIMA, SARIMA and SARIMAX" (Artley, 2022).

Sin embargo, es importante explorar diferentes enfoques, como las redes neuronales, las cuales han emergido como una alternativa potente. De manera específica, las redes convolucionales y las redes neuronales recurrentes ofrecen perspectivas para el modelado de problemas de regresión o clasificación, ofreciendo características únicas que podría mejorar la parte del modelado.

En esta sección, se exploran las capacidades y limitaciones de estas formas de modelado, teniendo así un panorama de cómo los diferentes métodos pueden converger para ofrecer soluciones más robustas y precisas en el estudio de series de tiempo.

1.2.1 Explorando SARIMA(X) en el modelado de datos

Como bien se ha mencionado para las series temporales se cuentan con modelos para analizar patrones y tendencias en los datos. Particularmente, SARIMAX se ha establecido como un pilar en este dominio por tener el poder de integrar factores diversos relacionados con la misma serie, así como variables externas. Este modelo extiende los ARIMA tradicionales al incluir componentes estacionales y variables exógenas, permitiendo así modelar influencias externas directas sobre la serie temporal.

Si bien detrás de ellos existe una extensa teoría detrás de ellos que es esencial comprender, es necesario recordar que el objetivo principal que se tiene en este trabajo no es adentrarse profundamente en la teoría, sino entender a grandes rasgos cómo se componen estos modelos y cómo pueden ser aplicados de manera práctica en problemas reales.

1.2.1.1. Componente autorregresivo – AR

Primero, se necesita introducir el componente autorregresivo "AR(p)", donde "p" indica el número de valores rezagados de la serie que se utilizan en el modelado de datos. Si "p" es 1, se toma el valor anterior multiplicado por un peso para predecir el siguiente valor. El anterior modelo estudia las caminatas aleatorias (ya que son influenciadas por el ruido blanco), hacer el parámetro "p" más grande se traduce en agregar más valores anteriores de la serie multiplicados por varios pesos.

$$y_t = c + \sum_{n=1}^{p} \alpha_n * y_{t-n} + \epsilon_t$$

Donde y_t es el valor de la serie de tiempo en el tiempo t, ϵ_t es ruido blanco y $\epsilon_t \sim N(0, \sigma^2)$, teniendo a c y α_n como números reales.

1.2.1.2. Media móvil – MA

El componente "MA(q)", en donde "q" es el número de términos de errores anteriores asociados a la serie de tiempo se llama de media móvil, se utilizará en el modelado de datos. Si "q" es 1, el pronóstico será influenciado por el término del ruido blanco anterior multiplicado por un peso. Hacer el parámetro "q" más grande toma más valores de ruido blanco anteriores para el pronóstico.

$$y_t = c + \sum_{n=1}^{q} \gamma_n * \varepsilon_{t-n} + \epsilon_t$$

Donde y_t es el valor de la serie de tiempo en el tiempo t, ϵ_t es ruido blanco y $\epsilon_t \sim N(0, \sigma^2)$, teniendo a c y γ_n como números reales.

1.2.1.3. Modelos ARMA, ARIMA y SARIMA

El modelo ARMA(p, q) es la combinación de los modelos AR(p) y MA(q) sumados, mientras que el modelo ARIMA$(p\ , d, q)$ incluye un paso de procesamiento de la serie de tiempo para volverla estacionaria, aplicando diferencias entre el valor actual y el anterior (también llamadas diferencias regulares) con ayuda del parámetro "d"[3]. Por último, se incluye la estacionalidad del modelo como componentes autorregresivos y de media móvil con un rezago igual a la frecuencia de la estacionalidad, denotado por "s", con la misma capacidad de realizar diferenciación de los datos con la misma frecuencia estacional, denotado por "D", teniendo como resultado los modelos SARIMA$(p\ , d, q)(P, D, Q)_s$.

$$y_t = c + \sum_{n=1}^{p} \alpha_n * y_{t-n} + \sum_{n=1}^{q} \gamma_n * \varepsilon_{t-n} + \sum_{n=1}^{P} \emptyset_n * y_{t-sn} + \sum_{n=1}^{Q} \varphi_n * \varepsilon_{t-sn} + \epsilon_t.$$

Donde y_t es el valor de la serie de tiempo en el tiempo t, ϵ_t es ruido blanco y $\epsilon_t \sim N(0, \sigma^2)$, teniendo a c, $\alpha_n, \gamma_n, \emptyset_n, \varphi_n$ como números reales.

1.2.1.4. Modelo SARIMAX

Por último, el modelo SARIMAX$(p\ , d, q)(P, D, Q)_s$ toma las cualidades del modelo SARIMA, considerando adicionalmente una cantidad "r" de variables externas. Esto es relevante ya que la variable a estudiar puede ser afectada por factores externos, es decir, los valores de la serie de tiempo principal puede ser consecuencia de estas variables externas. El modelo se denota de la siguiente manera:

$$y_t = c + \sum_{n=1}^{p} \alpha_n * y_{t-n} + \sum_{n=1}^{q} \gamma_n * \varepsilon_{t-n} + \sum_{n=1}^{P} \emptyset_n * y_{t-sn} + \sum_{n=1}^{Q} \varphi_n * \varepsilon_{t-sn} + \sum_{n=1}^{r} \beta_n * x_{n_t} + \epsilon_t.$$

Donde y_t es el valor de la serie de tiempo en el tiempo t, ϵ_t es ruido blanco y $\epsilon_t \sim N(0, \sigma^2)$, x_{n_t} es el valor de la variable externa n-ésima en el tiempo t, y donde $c, \alpha_n, \gamma_n, \emptyset_n, \varphi_n, \beta_n$ son números reales.

[3] Es importante mencionar que los parámetros que hacen referencia a la diferenciación (d y D) no aparecen explícitamente en la fórmula, ya que representan el número de diferencias regulares y estacionales que se aplican a los datos para hacer la serie de tiempo estacionaria, posteriormente se emplean los componentes autorregresivos y de media móvil de los modelos SARIMA.

Como se puede observar este modelo da la oportunidad de agregar influencias externas para obtener resultados más precisos en el modelado. Sin embargo, al momento de utilizar este tipo de modelos, se tienen que tomar en cuenta otros aspectos como el peso del modelo al guardarlo en memoria, el tiempo que tarda la computadora en entrenar el modelo y la forma de obtener los valores de las variables externas futuras para realizar la predicción de manera correcta. Estas cuestiones se debatirán durante el desarrollo de los problemas del presente trabajo.

Para el desarrollo de los proyectos se utilizará la librería de Statsmodels en Python, que provee distintas herramientas para el análisis estadístico y la modelización predictiva. Especialmente a través de su implementación de SARIMAX, que como hemos visto, integra componentes estacionales, autorregresivos, media móvil, y junto con las variables exógenas, permite a los científicos de datos modelar series temporales complejas con una precisión notable (Statsmodels, s.f.).

1.2.2. Aprendizaje profundo en el modelado de datos

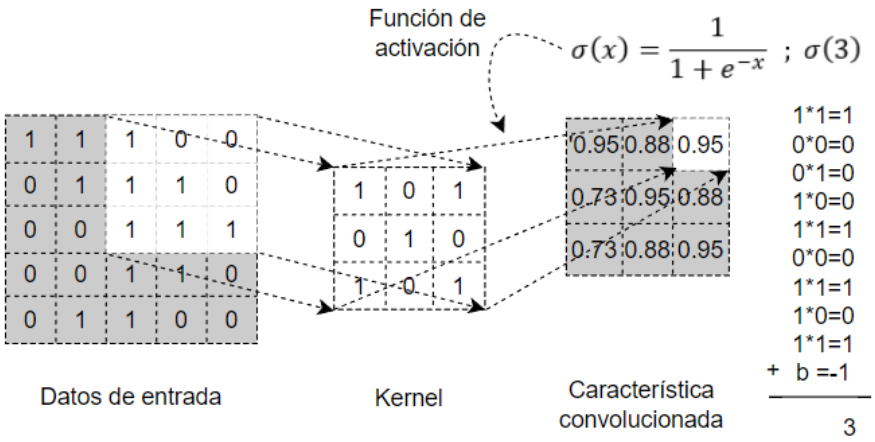
Aunque en el mundo del análisis de series temporales los métodos tradicionales han demostrado ser robustos, la estadística ofrece más herramientas para este análisis de series, destacando el uso del aprendizaje profundo, que ha emergido como una solución poderosa para enfrentar desafíos complejos en este campo. Específicamente, las redes neuronales convolucionales (CNN) y las redes de memoria a corto y largo plazo (LSTM) han mostrado ser particularmente eficaces para este propósito. En esta sección, se ofrece una introducción sobre el funcionamiento de estas técnicas de modelado y cómo pueden ser aplicadas en el estudio de series temporales.

1.2.2.1 Redes neuronales convolucionales CNN

Las redes neuronales convolucionales fueron diseñadas para usarse en aplicaciones de visión por computadora. Se hicieron más famosas después de los logros en el concurso de 2012 para el conjunto de datos ImageNet. El objetivo de estos modelos es transformar imágenes en algo más fácil de procesar a través de operaciones llamadas convoluciones, las cuales funcionan

mediante un núcleo. Al aplicar este núcleo a la imagen de entrada, se obtiene una característica convolucional. Esta característica, a través de una función de activación genera un valor de activación, y en conjunto estos valores se consideran como un mapa de activación.

Figura 6. Aplicación de un núcleo

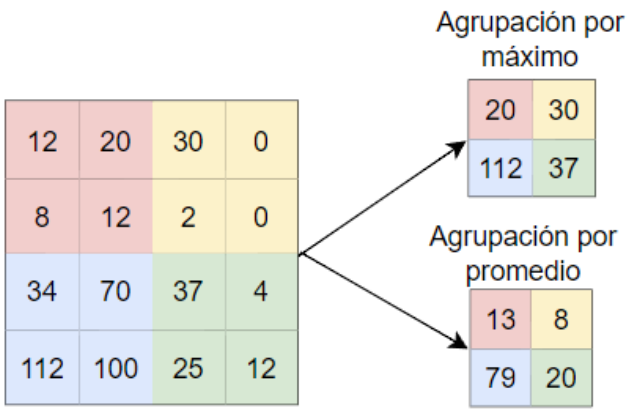


Representación del resultado de aplicar un núcleo en los datos de entrada para la obtención de una característica convolucional (Mandal, 2021; Venkatesan, 2017).

Generar varios de estos mapas de manera consecutiva con ayuda de un núcleo permite a una red neuronal convolucional la habilidad de aprender muchos más detalles de los datos de entrada. Comienza aprendiendo formas simples como líneas o diagonales en las capas más superficiales de la red, y continúa con algunos conceptos más complejos como objetos geométricos en las capas más internas.

Otro aspecto importante son las capas de agrupación, responsables de reducir el tamaño de las características convolucionadas. Esta reducción se realiza mediante un promedio o tomando el máximo de la parte de la imagen cubierta por el núcleo de la agrupación.

Figura 7. Aplicación de un núcleo de agrupación

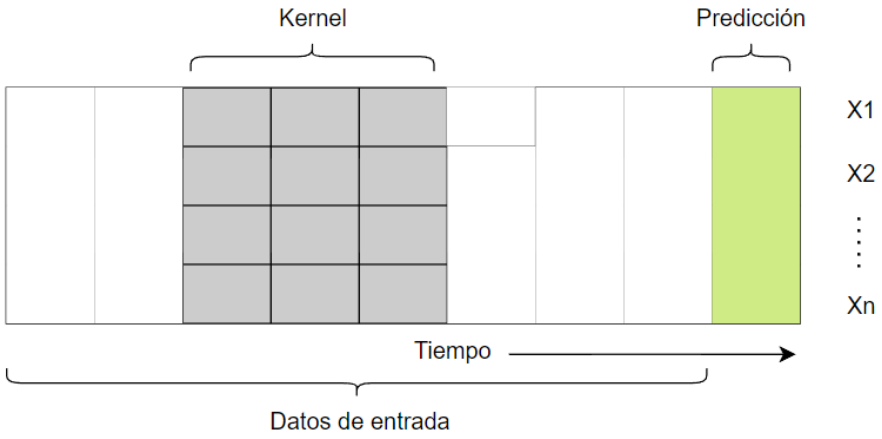


Representación del resultado de aplicar un núcleo de agrupación en los datos de entrada, con el objetivo de reducir características y el tiempo de entrenamiento manteniendo prácticamente la misma información (Mandal, 2021; Venkatesan, 2017).

Los núcleos antes descritos se conocen también como filtros. Es importante considerar el uso de una mayor cantidad de filtros en cada capa convolucional, ya que esto permite un mejor análisis de los datos de entrada, lo cual se traduce en la generación de un conjunto de mapas de activación. Es crucial saber esto porque el parámetro de filtro es importante al definir un modelo de tipo CNN. Tras haber explorado el funcionamiento y los componentes fundamentales de las redes neuronales convolucionales (CNN), podría surgir la duda sobre cómo aplicarlas en el análisis de series temporales, contrario a la percepción inicial, estas redes pueden ser útiles en este ámbito, la clave radica en el adecuado preprocesamiento de los datos, permitiendo a su vez la capacidad de manejar tanto variables exógenas como componentes de la misma serie temporal.

Para ello, se debe explicar algunos conceptos adicionales. Se supone que se tienen ocho valores del pasado para predecir el siguiente paso. Entonces, se deben convertir los datos en una matriz, donde los renglones representan el número de variables que se utilizarán y el número de columnas es el número de valores en el tiempo con el que se realizará la predicción. A cada uno de estos arreglos se les debe asociar con el número de valores en el futuro a predecir. Este procedimiento se realiza por cada unidad en el tiempo de la serie de tiempo, transformando los datos de esta manera para que puedan ser utilizados por las CNN

Figura 8. Transformación de datos para las redes neuronales



Representación visual de la forma de transformar los datos de entrada en arreglos que puedan ser procesados por las redes neuronales (Sefidian,2019; TensorFlow,2023).

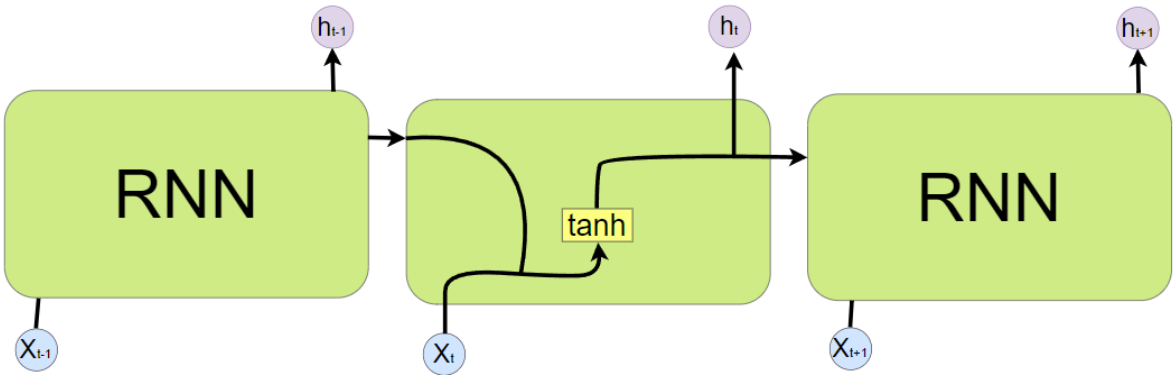
Teniendo en cuenta lo anterior, se tendrán todos los elementos y entendimiento necesarios para construir distintas arquitecturas de modelos de redes neuronales con la ayuda de las CNN, entre los cuales pueden estar tanto el modelado de imágenes o el modelado de secuencias con un objetivo de regresión o clasificación.

1.2.2.2. Modelos de memoria a corto y largo plazo LSTM

Ahora es importante revisar otro tipo de modelos llamados LSTM. Primero, se introducen las redes neuronales recurrentes (RNN), las cuales son una piedra angular en el análisis de datos secuenciales. Se caracterizan por su habilidad para procesar información considerando el valor actual con relación a los valores pasados. A diferencia de las redes neuronales tradicionales, las RNN tienen la capacidad de retener memoria de lo que han visto previamente, lo que les permite establecer conexiones temporales, una característica fundamental en tareas como el procesamiento del lenguaje natural y predicciones en series temporales.

La arquitectura de una RNN recibe una entrada "x" que produce una salida "h". Esto se repite en un bucle, permitiendo pasar la información de un paso a otro de la red neuronal, simulando tener una memoria durante la secuencia (Karpathy, 2015; Goodfellow, Bengio, & Courville, 2016). Sin embargo, estas redes enfrentan dificultades para conectar información que está muy alejada entre sí. Para solucionar esta problemática, se debe considerar otro tipo de arquitectura, como las LSTM.

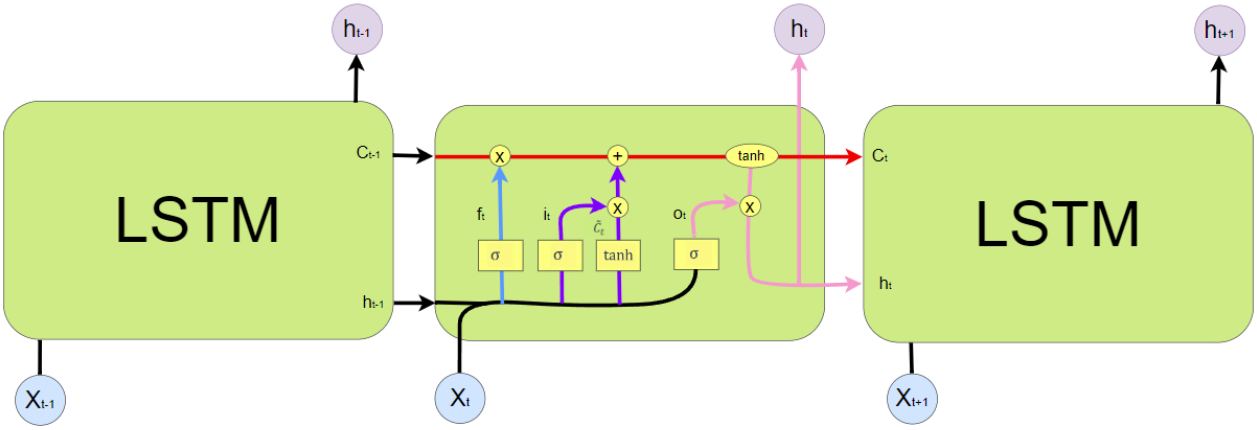
Figura 9. Representación de una red neuronal recurrente



Representación de los componentes de una red neuronal recurrente. Este tipo de redes son precursoras de las redes LTSM.

Las LSTM cuentan con cuatro estructuras importantes que interactúan de una forma especial para proveer a la red neuronal de memoria a largo plazo. A continuación, se revisarán cada una de estas estructuras para entender completamente su funcionalidad.

Figura 10. Representación de una red neuronal de memoria a corto y largo plazo



CAPA DE OLVIDO	$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$
CANDIDATO DE ESTADO	$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$ $\tilde{C}_t = tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$
ACTUALIZACIÓN ESTADO	$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$
SALIDA DE LA CELDA	$o_t = \sigma(W_o \cdot [h_{t-1}, x_t] + b_o)$ $h_t = o_t * \tanh(C_t)$

Representación de los componentes y flujo de información de una LSTM para lograr retener de mejor manera información en secuencias más largas (Olah, 2015; Filzinger,2023; Goodfellow, Bengio, & Courville, 2016).

1. ***Capa de olvido***: Esta parte decide qué tanto olvidar del estado anterior C_{t-1}, ya que toma la salida anterior h_{t-1} y x_t multiplicado por pesos específicos W_f. Esta capa es evaluada en una función de activación, teniendo como resultado un valor entre 0 y 1, que afectará al valor del estado anterior de la celda.

2. ***Candidato de estado***: El siguiente paso es crear un nuevo vector que contendrá los nuevos valores candidatos $\tilde{C}_t$, siendo controlados por el resultado i_t que es la salida de otra función de activación para saber qué parte de estos valores dejará pasar.

3. ***Actualización de estado***: Primero debe notarse que la información del estado anterior C_{t-1} puede pasar sin problemas, ya que sólo es afectada de manera lineal por f_t, este resultado se combina con el candidato de estado $\tilde{C}_t$ ponderado con i_t para obtener el nuevo estado de la celda siguiente C_t.

4. ***Salida de celda***: Se tiene que decidir cuál será la salida de la celda h_t, la cual se basará en el estado de la celda C_t valuado en una función de activación multiplicado por un valor de activación calculado O_t, con ayuda del valor anterior h_{t-1} y el valor actual x_t.

Como se ha observado, las LSTM parecen ser de gran utilidad en distintos ámbitos, especialmente en problemas relacionados con secuencias. Esto es un beneficio importante, ya que en este trabajo varios de los problemas analizados serán series de tiempo, lo cual se adecúa mucho a lo que nos pueden proveer estos modelos.

1.3. Aspectos importantes al momento de entrenar un modelo

Es crucial saber que la construcción de modelos robustos y confiables va más allá de la mera selección del algoritmo. Esto implica una revisión meticulosa de múltiples aspectos que afectan directamente su desempeño. Entre estos aspectos, se revisa la optimización de parámetros, la cual puede significar la diferencia entre un modelo bueno o uno malo. Para los modelos SARIMAX, se deben optimizar los parámetros explicados anteriormente. A menudo, se busca el mejor modelo con ayuda del criterio BIC o AIC[4], donde lo más complicado de obtener es el periodo estacional (denotado por s) que puede presentar la serie. Para los modelos de aprendizaje profundo, hay una variedad de parámetros por optimizar, entre los cuales están:

1. N_input: Tamaño de la secuencia que entrará al modelo para realizar una predicción.

[4] Tanto BIC (Criterio de Información Bayesiano) y el AIC (Criterio de Información de Akaike) son herramientas estadísticas fundamentales para la selección de modelos, ambos consideran la adecuación del modelo a los datos y penalizan sobre la complejidad.

2. N_filters: Número de dimensiones que tiene como salida cada parte del modelo.
3. N_kernels: Tamaño del filtro que utilizaran las CNN para obtener la convolución.
4. N_epoch: Número de veces que se entrenará el modelo sobre el total de los datos de entrenamiento para reducir la función de pérdida.
5. N_batch: Número de bloques de muestras con el cual se entrenará el modelo por cada época.
6. Optimizer: La forma en que se logrará minimizar la función de costo.
7. Learn_rate: Valor que representa cuánto va a aprender en cada iteración el modelo.
8. Init_mode: Forma de inicializar los pesos del modelo.
9. Activación: Función que opera la combinación lineal de los pesos en las capas del modelo.

Es importante que para encontrar los mejores y más significativos parámetros se realice un procedimiento llamado validación cruzada. Este proceso tiene como objetivo correr bajo distintas particiones de los datos de entrenamiento para verificar cuál sería, en promedio, el verdadero desempeño del modelo (Brownlee, 2020). Dado que interviene cierta parte aleatoria al entrenar un modelo, es necesario evitar aquellos modelos que tengan sobreajuste en los datos, ya que esto impediría que el modelo pueda generalizar de manera correcta.

En los datos, es necesario verificar si existen variables con escasa varianza, ya que éstas pueden no aportar información significativa al modelo. Éstas son posibles "variables constantes" que introducen más ruido al modelo. Por otra parte, se podría tener un problema de datos desbalanceados, lo cual ocurre cuando se cuentan con pocas muestras de las variables categóricas a predecir. Los modelos tienden a predecir de manera correcta a la mayoría de los datos posibles y, si nuestra clase minoritaria es importante, este aspecto podría estar en nuestra contra, ya que puede afectar el poder de generalización del modelo.

Para resolver el problema anterior, se pueden utilizar varios métodos, uno de ellos es el balanceo de los datos, donde mediante distintas técnicas se logra nivelar el número de muestras por categoría, dándoles la misma importancia al optimizar nuestro modelo y que pueda realizar las predicciones correctas. Otra manera es mediante la elección de la métrica adecuada, que puede medir distintos aspectos, como la cantidad de predicciones correctas de

una de las categorías o cuántos falsos positivos se cometieron. La elección de la métrica correcta debe alinearse con los objetivos específicos del proyecto.

Por último, cabe mencionar que la función de activación es un elemento importante al modelar datos, ya que estos modelos pueden funcionar tanto para problemas de regresión (predecir un número real) como de clasificación (predecir una clase). Esto se puede lograr no usando en la última capa una función de activación, lo cual ayudaría a obtener un número real, o utilizando una función sigmoide que nos daría una probabilidad. También es importante considerar las dimensiones de salida, lo cual nos ayudaría a predecir de manera simultánea varias series de tiempo o múltiples clases.

Los elementos que se discutieron anteriormente, junto con una comprensión profunda del problema, son parte esencial para el éxito al momento de desarrollar proyectos basados en ciencia de datos.

1.4. Desarrollo de proyectos como científico de datos

Hacer ciencia de datos es una disciplina que va más allá de realizar manipulaciones numéricas, ya que es un campo desafiante que requiere una comprensión profunda de los problemas reales, una gran capacidad analítica y una aplicación cuidadosa de la tecnología y las matemáticas para convertir datos en conocimiento. Es así como se explora la complejidad de los proyectos en escenarios del mundo real, donde los problemas raramente se presentan de manera ordenada y estructurada.

Iniciar con la comprensión del problema de negocio es crucial, ya que en esta parte se conoce de cerca todo lo que implica de manera profunda y de esta manera poder hacer un plan lógico de acuerdo con las necesidades de éste. Ahora es indispensable obtener un conocimiento profundo de las dependencias del proyecto, lo cual a menudo implica navegar por la complejidad de los sistemas que proveen los datos y las relaciones subyacentes. Es importante conocer si existe un impedimento y la forma de poder resolverlo.

La información anterior conduce a la formalización del problema, donde las preguntas difusas se transforman en hipótesis y objetivos medibles que definirán el éxito del proyecto. También es importante proveer fechas estimadas de los distintos entregables y realizar el diseño de la arquitectura de la solución. Es crucial que, antes de avanzar a las siguientes fases, se pueda verificar con el equipo si esto es lo que se busca, porque el trabajo de aquí en adelante se basará en los resultados de este paso.

Con el conocimiento previo del asunto, es necesario proceder a la recolección de los datos. En realidad, esta tarea es compleja debido a la diversidad de fuentes de datos, lo que dificulta la centralización de estos, seguido del procesamiento y transformación de los datos. Esta tarea es ardua porque los datos en general no están listos para su uso.

El siguiente paso es el modelado de datos, donde se aplican técnicas estadísticas y algoritmos de aprendizaje automático para descubrir patrones y hacer predicciones. Es importante entender que no hay un modelo mejor para cualquier tipo de datos. En este sentido, se tienen que comparar varios tipos de modelos para encontrar el óptimo. Se debe mencionar que el trabajo no termina con la construcción de un modelo; su evaluación en escenarios del mundo real es fundamental para medir la eficacia y realizar los ajustes necesarios que aseguren su relevancia y exactitud.

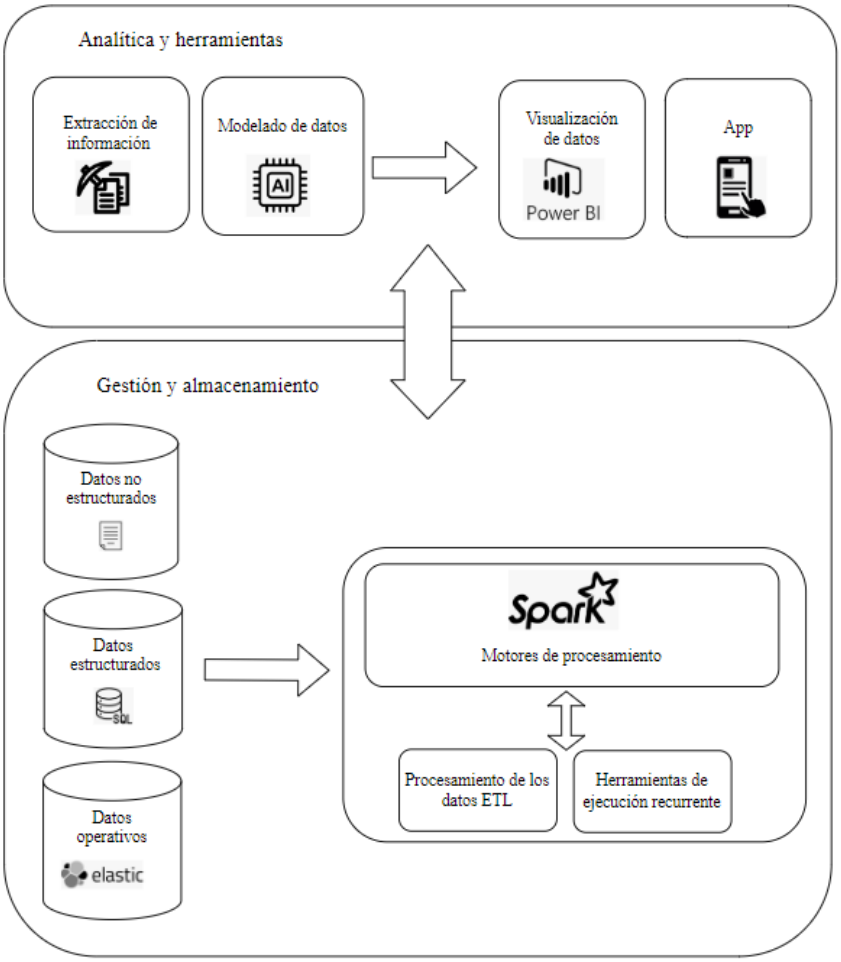
Finalmente, el proyecto se consolida con la puesta en producción, donde el modelo se integra en procesos de negocio existentes, pudiendo automatizar la toma de decisiones y facilitar la generación de valor continuo (Ulyanov, Guschin, Trofimov, Altukhov, & Michailidis, 2018). De esta manera, los pasos en un proyecto de ciencia de datos se resumen en lo siguiente:

1. Comprensión del problema de negocio.
2. Entender las dependencias del proyecto.
3. Formalización del problema.
4. Recolección, procesamiento y transformación de los datos.
5. Modelación de los datos.

6. Validación de los datos.
7. Poner en producción el proyecto.

En la parte de la formalización del proyecto, un aspecto crítico es el desarrollo del diseño de la arquitectura. Este diseño debe responder no solo a las necesidades inmediatas, sino también proyectarse como una solución sostenible y escalable. Paralelamente, es necesario construir un cronograma detallado que incluya la fecha de un primer lanzamiento para evaluar la alineación del proyecto con los objetivos estratégicos y una fecha subsiguiente que marque la culminación del producto en su totalidad.

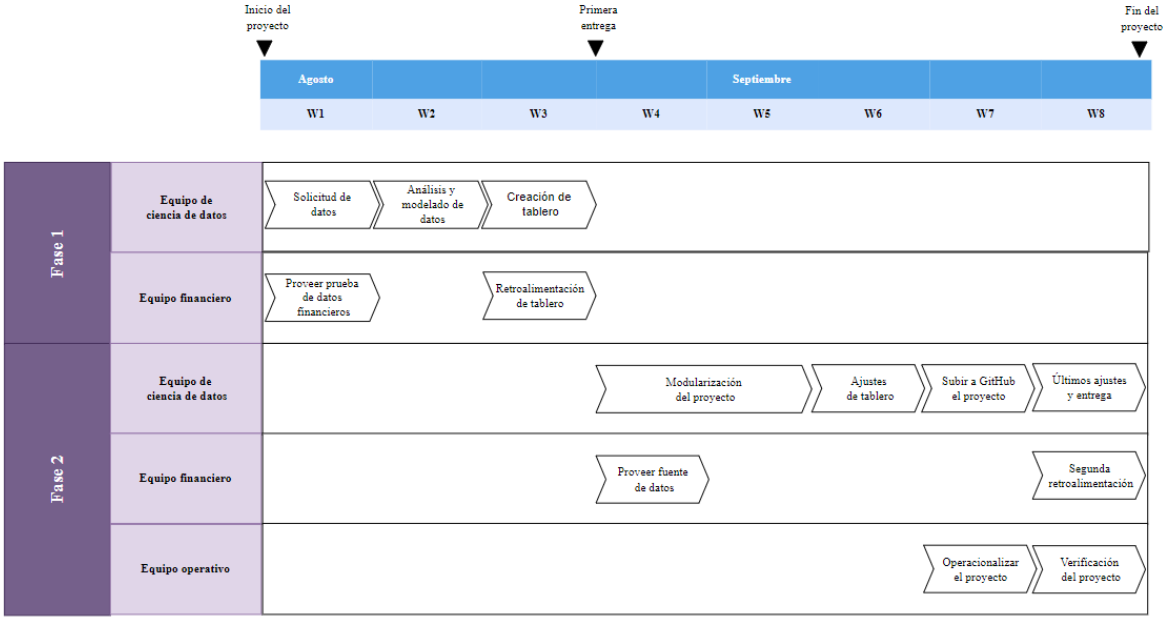
Figura 11. Diseño de la arquitectura de un proyecto



Visualización simplificada de la arquitectura de un proyecto, en el cual se mencionan las fuentes de información, en donde se centrará el proceso de los datos, la parte analítica del proceso y cuál va a ser el producto final (Solis, 2020).

Mientras se crea un cronograma también es importante especificar los intervalos para la revisión constante de los avances del proyecto. Es importante tener en cuenta en esta planificación los riesgos del proyecto, considerando el impacto de posibles contratiempos y delineando estrategias de mitigación. La importancia de estos pasos es poder mantener la transparencia y la comunicación continua que fortalece este proceso, permitiendo ajustes dinámicos y la gestión proactiva de cualquier desafío.

Figura 12. Cronograma para la planificación de un proyecto



Visualización de un cronograma y las interacciones necesarias entre equipo para el desarrollo de un proyecto

Con lo anterior podemos tener una comprensión de lo que implica un proyecto en el campo de la ciencia de datos, en donde se da claridad de cómo enfocar esfuerzos, que se convierten en soluciones prácticas y efectivas para enfrentar los retos del ámbito empresarial.

2. Calidad de información en la investigación de mercados

En la era de la información en que se vive, la investigación de mercados se ha convertido en una herramienta crucial para comprender las necesidades y preferencias de los consumidores. En este contexto, la calidad de la información adquiere una relevancia importante, ya que, sin datos precisos, confiables y relevantes, las decisiones empresariales pueden no ser certeras. Esto afecta directamente la relación con las empresas que requieren de estos servicios.

La calidad de la información no sólo implica que los datos sean verídicos, sino también su relevancia y actualidad. Las tendencias del mercado cambian constantemente, lo que se traduce directamente en la evolución de los gustos y preferencias del cliente. La personalización se ha convertido en una expectativa básica de los consumidores; los clientes quieren sentir que las empresas los entienden. La información de calidad proporciona la base para esta personalización. En las siguientes secciones, estudiaremos las formas en que podemos asegurar esta calidad en los datos.

2.1. Detección de precios atípicos

Una de las partes relacionadas con la calidad de la información es la detección de precios atípicos, cuyo objetivo es identificar valores anormales que pueden distorsionar los análisis y llevar a conclusiones erróneas. En este sentido, las técnicas para detectar valores atípicos juegan un papel fundamental. Es importante mencionar que los métodos que se utilicen deben ser escalables a la cantidad de información que se maneja durante el estudio de mercado, asegurando así que la información utilizada para la toma de decisiones sea de la máxima calidad.

La causa de estos precios atípicos comúnmente puede ser el resultado de errores en la recopilación de datos o de variaciones extremas del mercado. Si no se realiza, sobre todo,

una detección oportuna de la primera forma en que se obtienen precios atípicos, estos pueden afectar negativamente la interpretación de los datos. Como ejemplo, podemos ver cómo en los siguientes datos afecta negativamente al promedio.

Cuadro 1. Influencia de valores atípicos en el promedio

									Promedio
Datos con errores	15	18	24	99	18	17	56	22	33.6
Datos sin errores	15	18	24	19	18	17	16	22	18.6

Influencian de errores en la colecta, en este caso se insertó de manera incorrecta un dígito, lo cual impacta de manera significativa el promedio final.

Como se puede observar, la aparición de dos valores atípicos hizo que el promedio fuera prácticamente el doble del real. Se puede notar que tampoco el promedio está dentro del rango normal de los precios correctos. Para ejemplificar esto con datos reales, se procede a hacer un análisis en una base de datos pública que simula la recopilación de esta variable en un estudio de mercado.

Esta base de datos se obtuvo de Kaggle (Boysen, 2017) y es sobre los precios de productos globales, la cual contiene precios obtenidos en los mercados del mundo para diversos productos, incluyendo información sobre el país, el mercado, el precio en moneda local y el mes de registro. Aunque estos datos podrían utilizarse para relacionarlos con fluctuaciones monetarias, patrones climáticos, entre otros, en este caso los utilizaremos para encontrar valores atípicos aplicando los distintos procedimientos que ya hemos discutido. Se comienza explorando un poco los datos, donde se observa lo siguiente.

Cuadro 2. Exploración de los datos de precio de productos globales

Preguntas por resolver	Resultado
¿Cuáles son las dimensiones de los datos?	Los datos cuentan con 743914 registros y 18 columnas
¿Los datos cuentan con valores nulos?	Si, la única columna que tiene valores nulos es localidad, la cual tiene 13949
¿Cuál es el número de productos únicos?	Se tienen 321 productos únicos
¿El número de divisas únicas?	Se cuentan con 61 divisas únicas
¿Cuál es el número de productos registrados con una unidad de KG o G?	El 86.34 % de los datos
¿En qué años se recolectaron los datos?	Desde 1992 hasta 2017
¿Cuál es el porcentaje de datos recolectados después del 2000?	El 98.91 % de los datos
¿Cuál es el tamaño final de los datos tomando productos registrados con una unidad de peso, recolectados después del 2000 y sin las divisas "NIS" y "Somaliland"?	El tamaño final es de 619628 registros
¿Es constante la cantidad de productos que se registran en la base de datos?	No, crece el número de productos en un promedio de 18 productos más por año

Resumen de los datos más importante relacionados a los precios de productos globales, la cual nos da una idea clara del tipo de datos con los que estamos tratando y que tipo de tratamiento les tenemos que dar para su posterior análisis.

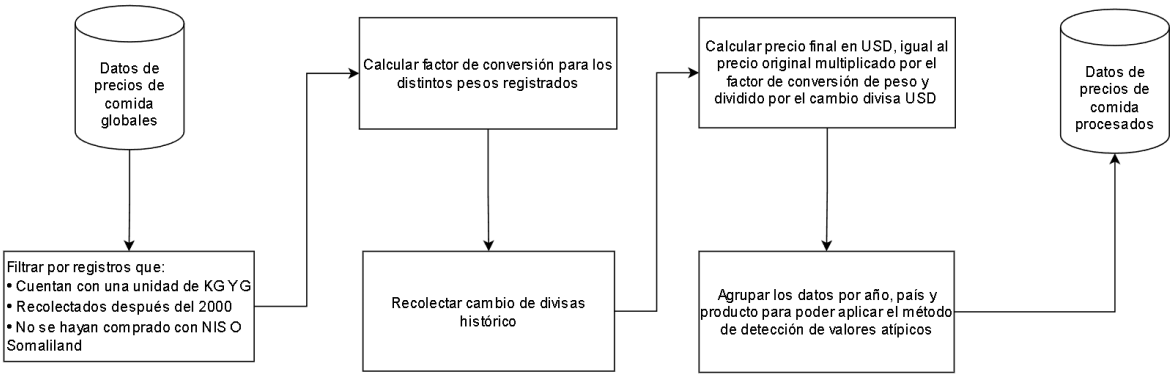
A partir de estos primeros resultados se pueden hacer las siguientes consideraciones que ayudan a tener datos limpios y lo más parecido a un problema real:

1. Los precios recolectados se basan en la cantidad comprada en distintas unidades de medida, donde el 86.34 % de los datos tienen una unidad basada en el peso (KG y G). Entonces se debe restringir a esos datos, además se realizarán las transformaciones correspondientes para que todos los precios estén por kilogramos.

2. Por otra parte, los datos contienen distintas divisas, por lo que se tienen que transformar en una divisa comparable, la cual será “USD”. Adicional como se necesita el tipo de cambio histórico y como conseguirlo antes del 2000 es complicado para todas las divisas, entonces se tomarán todos los datos después de ese año. Cabe mencionar que no se pudo encontrar de forma histórica dos divisas las cuales eran “NIS” y “Somaliland Shilling”. Tomando todas estas consideraciones se trabajará con el 83 % de los datos.

Es importante hacer notar que el conjunto de datos donde se aplicarán los métodos que busquen valores atípicos serán agrupados por año, país y producto. Se tomó esta agrupación, ya que el tamaño de los grupos con menos de 10 registros es del 10% de los grupos. No se añaden más campos para agrupar porque la información resultante queda muy granulada, lo cual afecta a algunos métodos haciéndolos menos eficaces. Después de tomar todas estas consideraciones, este será el recorrido que tendrá la información para poder aplicarle las pruebas de detección de valores atípicos.

Figura 13. Flujo de datos de precios de productos globales



Visualización de los distintos pasos que se realizaron para poder trabajar con los datos de precios de productos globales.

Teniendo la información ya preparada, ahora la propuesta es que primero se aplique la prueba de Grubbs con una regla adicional. Esta regla será que todos los grupos que cuenten con una varianza inferior a 0.01 en la variable de precio se tomarán como que no tienen un valor

atípico. Tomando esta regla, ya que esos grupos presentan precios colectados que no distan en más de un dólar. Posteriormente, a aquellos grupos que la prueba anterior mencione que hay un valor atípico, se les aplicarán los métodos de puntuación z (con un preprocesamiento de recorte con percentiles), IQR, DBSCAN y el bosque de aislamiento. De estos métodos se obtuvieron los siguientes resultados.

Cuadro 3. Número de valores atípicos por método aplicado

Método	Número de valores atípicos
Puntuación z	5321
IQR	5942
ISO	26578
DBSCAN	779
Todos	750
Puntuación z > 5	1845
ISO < -0.2	2547

Resultados de la aplicación de los distintos métodos para la detección de valores atípicos. De esta manera se puede observar a un alto nivel el efecto que tendrá en la tarea a resolver.

Se puede notar que los distintos métodos son más o menos estrictos al momento de encontrar los valores atípicos. Es esta la parte donde uno tiene que ser lo más cuidadoso posible, ya que

se tiene que escoger el método o una combinación, de tal manera que se logre detectar la mayor cantidad de valores atípicos y los que más podrían afectar sin dar tantos falsos positivos. La revisión y corrección de cada uno de estos precios requiere de tiempo y pruebas.

Adicionalmente, se puede notar que para los métodos de puntuación z y el bosque de aislamiento, se pueden utilizar distintos umbrales para que de esta manera sean más o menos estrictos al momento de encontrar valores atípicos y que, se encuentren los que se buscan. En este caso, la mayoría de los métodos arroja un número de valores atípicos que no representa más del 0.01% de los precios colectados totales. Se menciona esto, ya que cada uno de los valores que se toman como atípicos necesitarán una revisión para su posible corrección.

Para la siguiente exploración de datos, se debe tener en cuenta que la detección de valores atípicos se hará con todos los métodos. Se exploran las siguientes preguntas para conocer de manera más precisa el origen de los precios atípicos y, de esta manera, saber si están relacionados con un país, moneda o producto especial. El objetivo es conocer si es que se debieron, por ejemplo, a errores de recopilación de una región específica o si los valores atípicos están relacionados con un cambio normal de la zona.

De lo cual se observan algunas primeras cosas, como que el arroz es un producto que suele presentar muchos valores atípicos, o que el azúcar, los tomates y la carne de pollo presentan valores atípicos realmente grandes. Esto nos puede llevar a pensar por qué precisamente estos productos presentan estos valores tan elevados para resolver el problema de fondo.

Es importante mencionar que los promedios totales podrían ser mayores a los promedios de los productos atípicos, ya que el precio puede variar año con año (o entre momentos de recolección), por lo que se toma como un valor de referencia.

Por otra parte, en los países que presentan más valores atípicos las diferencias si son muy marcadas, los cuales son Arabia, Rwanda, Gambia, Lébano y Congo. Esto de alguna manera nos podría indicar que el problema, más que con un producto, podría estar relacionado con la localidad donde se recolecta. De esta manera, se podría dar seguimiento puntual a cómo se recolectan los datos en esa parte geográfica, para poder entender la raíz del problema desde este punto de vista.

Cuadro 4. Resultados de valores atípicos por producto y país

Producto	Valores atípicos	Máximo atípico	Promedio atípico	Promedio total
Azúcar	25	923.86	43.39	0.93
Arroz (importado	23	5.27	1.8	0.99
Harina de trigo	21	16.11	2.8	0.85
Carne de res	20	7.99	4.14	7.25
Tomates	20	923.86	48.73	2.36
Arroz	19	18.41	4.44	0.78
Carne de pollo	16	465.48	55.76	13.14
Zanahorias	16	3.97	1.92	0.63
Arroz local	16	4.62	2.55	0.82
Trigo	13	1.64	1.19	0.4
Repollo	13	2.86	1.87	0.47
Mangos	13	5.35	3.31	0.84

País - divisa	Valores atípicos	Máximo atípico	Promedio atípico	Promedio Total
Syrian Arab Republic - SYP	79	161.13	22.99	3.37
Rwanda - RWF	77	58.4	3.37	0.81
Gambia - GMD	61	23.88	3.16	0.84
Lebanon - LBP	60	18.36	5.84	3.08
Democratic Republic of the Congo -	45	26.42	5.03	2.8
Mozambique - MZN	35	5.33	2.49	0.89
Myanmar - MMK	33	487.9	178.99	48.42
Tajikistan - TJS	30	5.22	1.74	1.33
Colombia - COP	29	34.39	4.4	1.86
Yemen - YER	22	5.76	2.49	1.13
Philippines - PHP	20	4.38	2.93	2.02
Jordan - JOD	20	13546.29	1149.73	29.37

Cuadro con los productos y países que más cuentan con valores atípicos con el objetivo de conocer el origen de estos, de esta manera se les puede dar una solución más específica.

Adicionalmente a los resultados anteriores, es importante mencionar que correr el proceso completo toma 5 minutos en total. Sin embargo, es posible que ante un escenario con mayor cantidad de datos el proceso podría tardar mucho más tiempo, tiempo que se tiene que considerar para no afectar los procesos que siguen a la detección de valores atípicos.

Como se ha visto, la detección de precios atípicos durante un estudio de mercado a través del uso de los distintos métodos descritos es un componente esencial en la garantía de la calidad de la información recolectada. Al tratar estos valores anormales, se puede asegurar que las decisiones se toman con base en información precisa y confiable.

2.2. Control de inexistentes y detección de medidas incorrectas

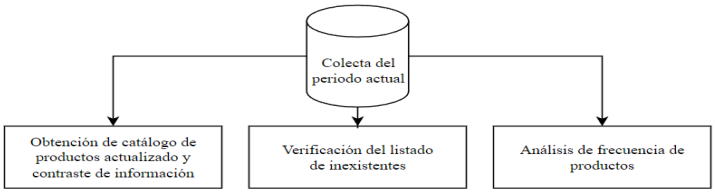
Este subcapítulo profundiza en el estudio y análisis de la aplicación de controles de calidad en la información recolectada en campo, de manera más específica en el control de productos inexistentes y en la detección de medidas incorrectas de acuerdo con ciertas reglas de la recolección. Es importante mencionar que no siempre es necesario un sistema complejo para resolver problemas del negocio, siempre y cuando se logre el objetivo de tener la mayor calidad de información posible.

Existe una manera preventiva para verificar la existencia de productos, que consiste en pedir o ir directamente al catálogo en línea de las distintas empresas que requieren del servicio de estudio de mercado. Esto con el objetivo de tener un listado de productos actualizado con los cuales se pueda contrastar directamente con la recolección, asegurando así la correcta recolección de estos.

Una vez aplicado este contraste con los productos de un catálogo conocido, se tiene que comparar la información recolectada en campo con una lista establecida de productos inexistentes. Esto permite la corrección de aquellos productos que se recolectaron de manera errónea en campo.

Adicionalmente, por cada periodo de recolección se realiza un análisis para comprender si un producto existe, estudiando su frecuencia a través del tiempo. Siendo candidatos a inexistentes aquellos productos con inconsistencias en el tamaño de recolección, teniendo como ejemplo productos que son muy poco frecuentes. Una vez detectado y confirmado que se recolectó de manera incorrecta, se corrige la información al producto correcto.

Figura 14. Métodos para la detección de productos inexistentes



Visualización de los métodos que se aplican para la detección de productos inexistentes con fines de prevención y corrección durante la colecta.

Respecto al control de medidas incorrectas, se procede a comparar la información recolectada con las normas establecidas en manuales de procedimientos. Aquéllas que no las cumplen se identifican y corrigen. Este control resulta necesario porque a las empresas que usan esta información necesitan asegurarse de que sus productos están exhibidos de manera acordada en los distintos puntos de venta y el comunicar información con medidas incorrectas podría llevar a conclusiones incorrectas.

Con lo anterior discutido, se puede dar uno cuenta de que, con un sistema de control de calidad de la información no tan complejo, pero efectivo, se puede asegurar la obtención de la realidad del mercado de manera confiable y precisa, manteniendo de esta manera nuestra relación con el cliente.

2.3. Detección de errores de la colecta en campo

La información recolectada tiene el objetivo de identificar y prevenir malas prácticas durante la recolección en campo. Ya que presentar información errónea con pequeñas variaciones se traduciría en que la estimación de la información para la zona que representa la recolección presentaría grandes sesgos, y en última instancia, que el cliente tome malas decisiones con esta información.

Para poder llevar a cabo una detección óptima de estos casos, se cuentan con distintos métodos que ayudarán tanto a conocer de manera oportuna como preventiva los distintos errores al momento de la recolección. Estos métodos son:

1. ***Análisis de variabilidad entre las colectas semanales***: Para detectar aquellas colectas que no se estén realizando de manera correcta.
2. ***Aplicar un detector de valores atípicos a los distintos productos por zona***: Aunque es posible que se den valores atípicos en las variables recolectadas de manera natural,

el verificar por zona la información recolectada y asociar esta verificación con las distintas colectas podría darnos un indicio de posibles errores de recolección.

3. ***Realizar revisiones de manera periódica***: Es necesario que los auditores más experimentados y de confianza realicen de manera usual revisiones a las demás colectas para corroborar la calidad de la información. De igual manera, en aquellas que se hayan presentado algún error en la colecta, se les puede dar un seguimiento de esta manera.

4. ***Mediante un llamado del cliente***: A veces los clientes piden una revisión específica de alguna zona, ya que la información entregada no empata con lo que ellos esperaban. Por lo tanto, se hace una revisión con ayuda de auditores experimentados para corroborar los datos colectados.

En resumen, el análisis cuidadoso de los datos y la prevención proactiva de las malas prácticas son fundamentales para garantizar la integridad de los datos colectados. Ya que así se puede proveer de información confiable al cliente, permitiéndole tomar decisiones correctas basadas en la realidad del mercado.

2.4. Retroalimentación del periodo a partir de los descubrimientos

Se puede notar con lo anterior que el análisis técnico ha sido fundamental para poder obtener resultados que mejoren la calidad de la información. Sin embargo, sería un error considerar que el trabajo culmina únicamente con la presentación de datos y hallazgos. Más bien, se revela una nueva etapa crucial que se inicia a partir de los descubrimientos obtenidos: la generación de un plan de acción.

Lo primero que se tiene que hacer es una retroalimentación mensual a partir de los hallazgos durante la colecta en campo, destacando lo que se tiene que corregir en la siguiente colecta. Se intenta generar un impacto significativo en los auditores con mensajes claros, y

posteriormente el seguimiento de estos en la nueva colecta. En esta retroalimentación se revisan temas entre los que están:

1. Buenas prácticas al momento de la colecta
2. Revisión de productos inexistentes
3. Revisión de errores comunes durante la colecta
4. Exploración de las novedades en la colecta del siguiente periodo

Adicional al mensaje general que se manda a los auditores, se podrían revisar áreas de mejora con los auditores que presentaron algún tipo de error durante la colecta, además de darles el seguimiento necesario para verificar que ya no se cometa el mismo error durante la siguiente colecta.

Es importante notar que la colaboración entre diferentes actores involucrados, así como el liderazgo y compromiso, son factores claves para la mejora de la calidad de la colecta. En donde como profesionales se tiene la responsabilidad de no solo descubrir la verdad, sino también de ser agentes del cambio.

3. Logística efectiva para la satisfacción del cliente

En este capítulo, se mostrará un aspecto esencial en el éxito de cualquier empresa: la logística. Ya que detrás de la entrega de un producto o servicios se encuentra una serie de pasos que, cuando son ejecutados de manera correcta, mejoran la satisfacción del cliente. Aquí se acentúa la visión en la gestión de inventarios, la solución ágil de problemas imprevistos y el manejo de grandes cantidades de información generada por el negocio.

Lo anterior mencionado forma parte de los procesos que garantizan la disponibilidad de productos en el tiempo y lugar precisos. Así mismo, aporta al negocio en la reducción de costos operativos y mejora de la eficiencia, beneficiando a ambos. Una cadena de suministro bien coordinada permite la precisión en cada paso del proceso, desde el origen hasta el destino final, permitiendo a las empresas forjar relaciones sólidas con sus clientes y asegurar un crecimiento sostenible en el mercado competitivo actual.

3.1. Generación de flujos de información para la planeación logística

La correcta administración y optimización de la logística en una empresa depende en gran medida de la calidad y la disponibilidad de las métricas generadas por el mismo. De manera particular, es importante la generación de flujos de información que muestren y procesen los datos relativos a los distintos productos que obtiene la empresa de sus distintos proveedores.

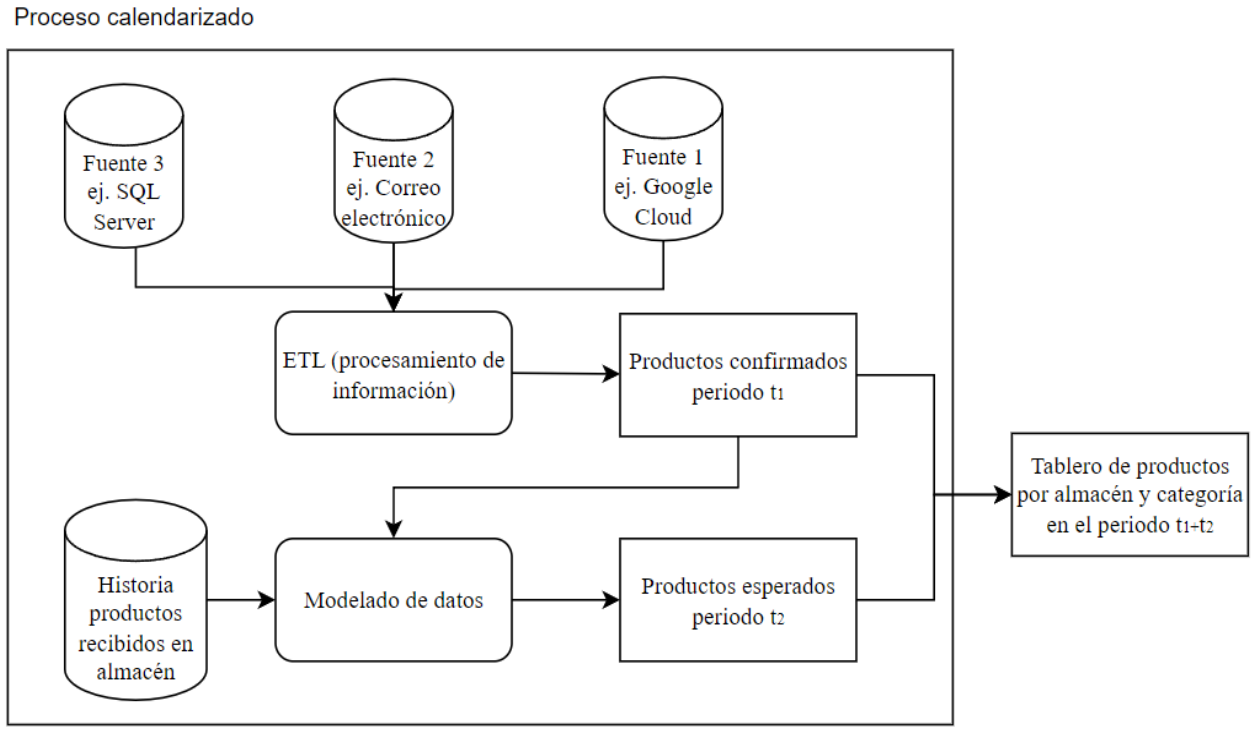
La generación de estos flujos de información tiene como objetivo final optimizar la planificación en las bodegas. Al contar con antelación qué productos se recibirán, se puede preparar y organizar las bodegas de manera que se minimice el tiempo de procesamiento de los productos, se optimice el espacio disponible y se mejore la eficiencia general de la operación. De esta manera, la empresa puede asegurarse de tener siempre los productos correctos en el lugar correcto y en el momento correcto, lo que además genera una reducción

de costos, ya que el producto no se queda detenido por mucho tiempo. Esto a su vez da como resultado una mejor experiencia para el cliente.

Hay que notar que esta tarea tiene una alta complejidad. A continuación, se tiene que centralizar las distintas fuentes de información que contienen lo necesario para llevar a cabo el proyecto, tomando en cuenta la frecuencia con la que se actualiza la información, quién la provee, en qué formato, etcétera. Una vez centralizada la información, sigue el tratamiento de los datos, en el cual es necesario seguir una serie de pasos para generar conocimiento concreto de la operación. Cabe mencionar que la parte del procesamiento de datos de productos confirmados requiere de muchas reglas que podrían estar constantemente actualizándose y que además tienen una gran complejidad dependiendo del tipo de información que se esté tratando. Lo anterior es necesario porque la mayoría de la información generada por la operación necesita ser procesada para que pueda generar un valor real y positivo en el negocio.

Hasta el momento se ha podido tratar con los datos conocidos, que da una visión de la mercancía acordada en un periodo dado. Sin embargo, para poder proveer al almacén de la información suficiente para que pueda realizar de manera correcta su planeación necesitan tener una visión más amplia. Por lo que la cantidad de productos que llegarán por categoría y por almacén que se presentarán, estará compuesta de una primera parte de productos ya confirmados por llegar en un periodo t_1 y una segunda parte desconocida que se resolverá con la predicción de productos por categoría para un periodo t_2 con ayuda de la información histórica.

Figura 15. Proceso para la planificación de bodegas



Visualización del desarrollo de un proyecto en ciencia de datos para la planificación de bodegas, en donde la parte del modelado solo es una parte del trabajo que se realiza.

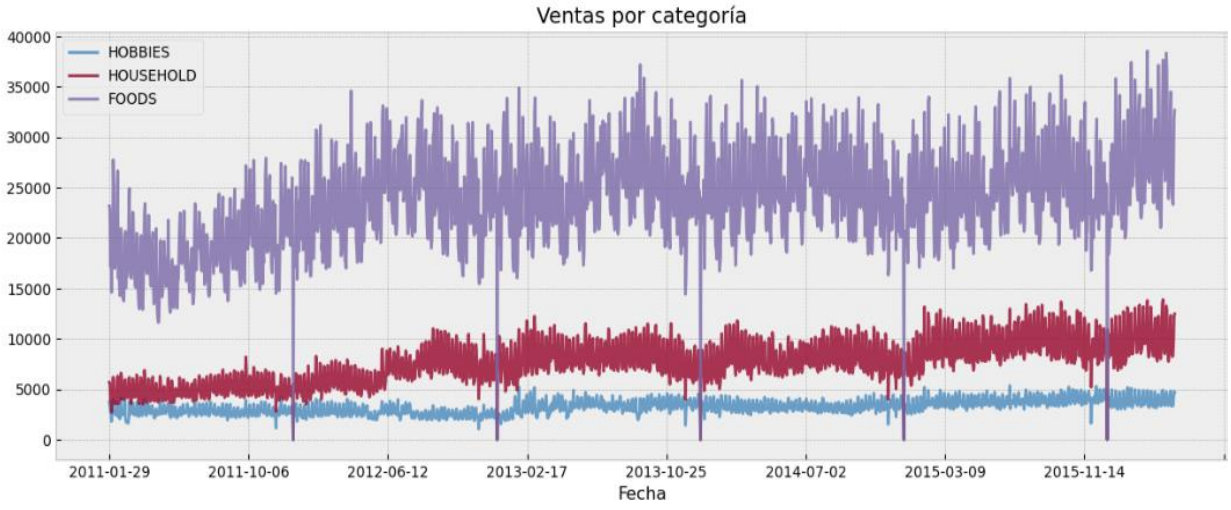
Los datos que se utilizarán serán los de la competencia "M5 Forecasting – Accuracy" de la plataforma de Kaggle, los cuales son los datos en ventas jerárquicos de Walmart. Esto con el objetivo de realizar las mejores predicciones comerciales para poder salvar costos y entender las mejores oportunidades para surtir las tiendas correctas de los productos que se necesiten. En este caso, sólo se enfocó únicamente en la predicción de venta de artículos por categoría para simular lo que se viviría en el problema descrito de planeación de bodegas para la predicción de productos que se piensa que llegarán por categoría. Explorando los datos y visualizando la serie, se obtiene lo siguiente.

Cuadro 5. Exploración de los datos de ventas

Preguntas por resolver	Resultado
¿Cuál es el tamaño de la serie de tiempo?	Se cuentan con 1913 días de historia desde el 2011 al 2016
¿Cuántos son los productos únicos por categoría?	Son 14370 de comida, 10470 para el hogar y 5650 de pasatiempos
¿Cuántos eventos únicos se consideran en los datos?	Cuenta con 30 eventos únicos como lo es el Super Bowl, el día de san Valentín, el 5 de mayo, etc.
¿Cuáles son las variables externas que se usaran?	Semana en curso, Día, mes y año, eventos ocurridos
¿Con qué retraso se encuentran más correlacionados los datos?	Con 7 días de retraso los datos se correlacionan más

Visualización general del tipo de datos con los que se trabajara en la predicción de ventas, de esta manera se observa el tipo de variables externas que se podrán utilizar, así como cuál es el total de datos con los que se cuenta para el modelado de datos.

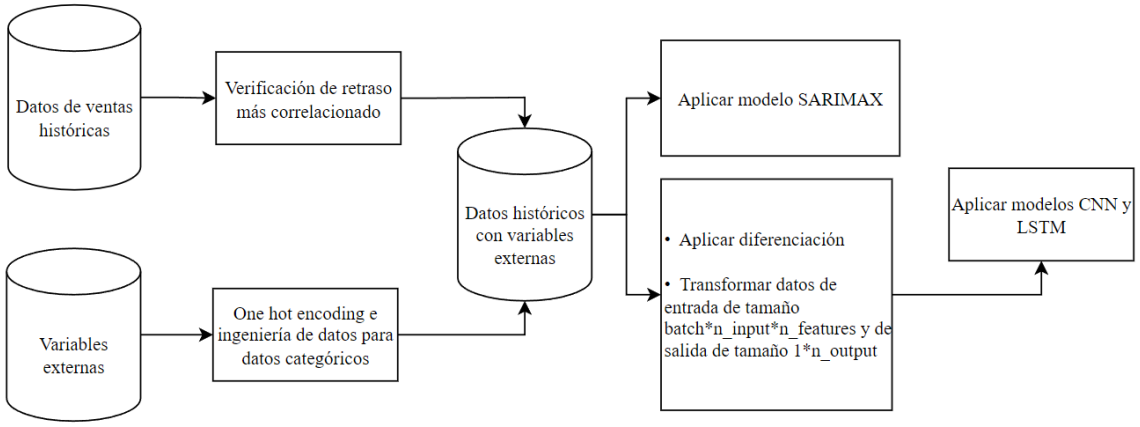
Figura 16. Ventas por categoría de productos



Visualización a alto nivel de las series de tiempo con las que se quiere trabajar, entendiendo la tendencia, posible estacionalidad y valores típicos presentados en las mismas, entendimiento que se utilizara para su posterior modelado y predicción.

Para lograr realizar la predicción de las categorías anteriormente expuestas, se tiene que seguir una serie de pasos que serán la preparación de los datos que anteriormente se han limpiado. Esto ya que los distintos modelos que se utilizan requieren de distintas preparaciones. Después, se probará con los modelos anteriormente expuestos, los cuales serán SARIMAX, CNN y LSTM. Por último, se verá una pequeña explicación de lo que se tendría que considerar para llevarlo a un entorno en producción y que automáticamente pueda llevar a cabo un reentrenamiento.

Figura 17. Flujo para el modelado de ventas



Comprensión del tratamiento de datos requerido para el entrenamiento de los distintos modelos

Como se acaba de ver de manera muy resumida, primero se tienen que tratar los datos, transformando todas las variables categóricas de tal manera que los modelos puedan comprender los datos. De esta manera, se obtiene un total de 41 variables externas. A partir de esto, se obtienen los datos con los cuales se entrenará la parte de SARIMAX. Sin embargo, para poder entrenar las redes neuronales, se tienen que transformar los datos de tal manera que estos modelos los puedan procesar.

Para ello se tienen que transformar los datos de un tamaño, $\text{batch} * \text{n\_input} * \text{n\_features}$ donde *batch* es el número de muestras de los datos con el que se optimizará la función de pérdida del modelo, n_input el número de valores en la historia con el que se realizará la

predicción y $n\_features$ el número de variables con las que se intenta explicar la serie de tiempo. La salida tiene un tamaño,$1 * n\_output$ donde n_output es el número de pasos siguientes que se quieren predecir.

Ahora que ya se tienen los datos listos para cada tipo de modelo, es momento de encontrar el mejor modelo encontrando los mejores parámetros a través de un método llamado "Grid search". Este método tiene como objetivo probar con un conjunto de parámetros y verificar cuál es el que mejor se ajusta a los datos con respecto a una métrica en especial, en este caso se usará el error cuadrático medio. Teniendo el mejor modelo para cada tipo, se compararán entre ellos y el mejor será el modelo final. Esto se hará por cada serie de tiempo relacionada con cada categoría. Los resultados se pueden ver en el siguiente cuadro.

Cuadro 6. Optimización de parámetros para modelos de regresión

	SARIMAX	CNN	LSTM
Búsqueda de parámetros	p, q, d, P, Q, D pueden tomar el valor 1 o 2 y s es la estacionalidad de la serie (en este caso es 7)	{ n_input:[7, 14], n_filters: [20, 50, 100], n_kernels:[3,5], n_epochs:[18], n_batch:[32, 128, 256], n_diff: [0, 7], optimizer:['adam', 'Adamax', 'Adagrad'], learn_rate:[0.001, 0.1], init_mode:['uniform', 'lecun_uniform', 'normal', 'zero'], activation:['linear', 'relu', 'softplus', 'tanh'] }	
Resultados por categoría sin variables externas	Combinaciones por serie 64 Duración:10 min Household - (1,1,1,1,2,2) RMSE 899 Hobbies - (2,1, 2, 2, 2, 2) RMSE 574 Foods - (1,1,2,2,1,1) RMSE 2467	Combinaciones por serie 100 Duración: 15 min Household (14, 50, 5, 6000*, 128, 7, "Adagrad", 0.1, "normal", "softplus") RMSE 543 Hobbies (14, 100, 5, 40, 6000*, 0, 'adam', 0.001, 'normal', 'linear') RMSE 437 Food (14, 50, 5, 40, 6000*, 7, 'Adagrad', 0.1, 'uniform', 'softplus') RMSE 1826	Combinaciones por serie 255 Duración: 129 min Household (14, 50, 5,nan,18, 128, 7, "Adagrad", 0.1, "normal", "softplus") RMSE 805 Hobbies (7, 100, 5,nan,18, 32, 7, "Adamax", 0.1, "normal", "relu") RMSE 420 Food (7,50,5,nan,18,5,7, "Adagrad", 0.1, "normal", "softplus") RMSE 3051
Resultados por categoría con variables externas	Combinaciones por serie 64 Duración: 62 min Household - (2,2,2,2,1,2) RMSE 616 Hobbies - (1,1,1,2,1,1) RMSE 356 Foods - (1,1,1,1,1,2) RMSE 2337	Combinaciones por serie 350 Duración: 40 min Household (14,100,5,40,15000*,500,7, "Adamax",0.1,"lecun_uniform","linear") RMSE 320 Hobbies (14, 50, 5, 15000*, 128, 7, 'Adagrad', 0.1, 'uniform', 'softplus') RMSE 141 Food (14,100,5,15000*,32,7, "Adagrad",0.1,"normal","relu") RMSE 1100	174 combinaciones - Duración: 84 min Household (7,100,nan,700*,128,7, "Adamax",0.1,"uniform","softplus") RMSE 287 Hobbies (7,100,nan,500*,128,7, "Adamax",0.001,"uniform","softplus") RMSE 105 Food (7,100,nan,15000*,32,7, "Adamax",0.001,"uniform","linear") RMSE 2064

Comparación de resultados con bajo distintos escenarios para la elección final de modelos con respecto a la métrica de raíz del error cuadrático medio, el número de combinaciones que se seleccionaron se basa en el tiempo de procesamiento de los distintos modelos, para el entrenamiento de la CNN se utilizó una GPU y de esta manera se aceleró el entrenamiento.

De los anteriores resultados, se pueden notar varias cosas como: Lo primero es tomar en cuenta que los modelos SARIMAX, suelen pesar más cuando se guardan; en este caso, pesaron 150 MB, mientras que los modelos CNN o LSTM, cuando se guardaron, sólo pesaron 30 kB. Esto es algo a tomar en cuenta porque al momento de escalar el proyecto a varias series de tiempo, el espacio que ocupan los modelos puede ser realmente grande, lo cual podría afectar dependiendo de las características en memoria del ambiente productivo donde se localice el proyecto.

Por otra parte, es importante pensar en los tiempos para encontrar los mejores parámetros para cada tipo de modelo. En este caso, los modelos SARIMAX suelen ser más rápidos al momento de encontrar los mejores parámetros, ya que el tiempo de entrenamiento es realmente corto. En este caso, solo se tardó 10 y 62 minutos para encontrar los mejores parámetros cuando se tienen o no variables externas respectivamente, mientras que para encontrar los mejores parámetros en las redes neuronales depende de cuántas combinaciones de las totales se esté probando, del número de épocas que se utilice para entrenar los modelos y si se están tomando en cuenta las variables externas o no. En este caso, el tiempo para encontrar los mejores parámetros podría multiplicarse por el tiempo de entrenamiento, sólo si se tiene a disposición una GPU y se cumplen ciertos criterios la búsqueda de parámetros puede ser realmente rápida como se observa con las CNN.

Otra cosa que se puede notar, es que en general los modelos que consideran variables externas tienen mejores resultados. Aunque es importante mencionar que cuando se consideran variables externas, es necesario que al momento de realizar la predicción a futuro se utilicen igualmente los valores a futuro de estas variables externas, lo cual se puede obtener modelando a su vez esas variables. En este caso no es necesario porque las variables externas utilizadas son relacionadas con el tiempo y fechas especiales, por lo que fácilmente se podrían obtener los valores futuros para utilizarlos.

Por otra parte, los modelos LSTM tuvieron el mejor desempeño para “Household” y “Hobbies”, mientras que las CNN lo hicieron bien en “Foods”. Sin embargo, estos modelos tienen otras contras como el tiempo de entrenamiento, ya que se optó por tomar un gran número de “épocas” para que el modelo pudiera entrenarse de mejor manera (en los

parámetros están señaladas por *), pero a costa de que tarda mucho tiempo en entrenarse. Mientras que entrenar una SARIMAX teniendo los mejores parámetros es prácticamente instantáneo, esta parte depende mucho de las características del CPU y GPU del ambiente productivo donde se localice el proyecto.

Por último, la selección entre los parámetros que pueden tomar los modelos para SARIMAX se basa en los valores más comunes que presentan este tipo de modelos, teniendo en cuenta que a veces los modelos no tan complejos se desempeñan mejor en la mayoría de los casos. Mientras que los valores escogidos para las redes neuronales se basan en las distintas posibilidades con las que el modelo logra reducir la función de pérdida, entre los que están el periodo de diferenciación para volver una serie estacionaria, el número de valores que mira hacia atrás, y las épocas en las que se entrenará el modelo.

Es importante, por último, realizar un sistema que tenga como objetivo el reentrenamiento de los modelos en caso de no cumplir con ciertos criterios relacionados con qué tan certeros son los modelos para realizar predicciones. La consecuencia de que los modelos reduzcan su poder predictivo será que se volverá a correr el proceso anteriormente descrito para encontrar el mejor modelo, de tal manera que se puedan seguir cumpliendo los objetivos buscados con la mejor calidad posible. Es importante mencionar que esta parte en sí misma tiene una alta complejidad para su desarrollo, pero es un sistema que todo proyecto de ciencia de datos debe contemplar.

Es realmente complejo encontrar el mejor modelo para los datos ya procesados, esto siempre pensando en encontrar la solución más adecuada que esté en línea con el ambiente productivo en donde se aloje el proyecto, así como con el objetivo que se plantea alcanzar en cuestión de precisión buscada del modelo. De igual manera, es importante pensar en caso de ser necesario que la solución sea escalable para que el proyecto se pueda mantener de una manera sencilla y a largo plazo.

Con el ejemplo anteriormente desarrollado, es necesario realizar un proceso similar para poder predecir la parte esperada de productos que llegarán a los almacenes. De esta manera, juntaremos esta información con la información de los productos ya confirmados, y de esta

manera poder proveer al almacén de la visibilidad necesaria para que puedan realizar de manera correcta su planeación. Esta interacción se realizará mediante la creación de un tablero que recibirá esta información y la cual se actualizará cada cierto periodo de acuerdo con la naturaleza de los datos y que esté alineado a las necesidades del área que utilice estos datos. De esta manera es que resolvemos este problema.

3.2. Soluciones ante situaciones de urgencia

Dentro de los procesos de la empresa, pueden surgir de forma imprevista situaciones de emergencia, como el traslado inmediato de productos de una bodega clave a otra, sin interrumpir la entrega regular. Para hacer frente a estas situaciones, es necesario el diseño de planes logísticos de manera eficaz. Para llevar a cabo este plan, se necesita transportar sólo los productos necesarios para ahorrar costos. De esta manera, el objetivo es proveer el inventario de los productos que se quieren mover en el tiempo solicitado para que se pueda desarrollar este plan.

La mayor complicación es que no siempre se cuenta con la información necesaria en una base de datos. Por ejemplo, se podría contar sólo con las entradas y salidas de los productos, lo cual es un problema, ya que, sin tener un valor de inventario inicial en un tiempo determinado, no se puede obtener el inventario buscado. Es en este punto donde podría ser necesario desarrollar alguna solución haciendo supuestos razonables.

> Supuesto 1.- Primeras entradas, primeras salidas, es decir, todo lo que entra primero es lo que sale y se vende.
>
> Supuesto 2.- Se intentará nunca tener rezagos de producto

Para entender, un primer paso es ver cómo se vería esto en un ejemplo. Hay que suponer que en un tiempo t_0 se tiene un inventario inicial i_0. Entonces si en un tiempo t_1 se tienen entradas e_1 y salidas s_1 se tendría que el inventario en este tiempo sería $i_1 = i_0 + e_1 - s_1$. Así en un tiempo t_n se tendría que el inventario es $i_n = i_{n-1} + e_n - s_n = i_0 + e_1 + \cdots + e_n - (s_1 + \cdots + s_n)$. Si ha pasado un tiempo suficientemente largo, entonces se puede decir con ayuda

de los supuestos que $i_0 - (s_1 + \cdots + s_i) \approx 0$ para una *i*, lo que se traduce a que los productos que han entrado salgan en algún momento.

Lo anterior ayudará en el desarrollo de la estrategia para la obtención del inventario buscado. Para ello en el tiempo t_1 se toma como $i_1 = e_1 - s_1$, es decir, la suma de las entradas y salidas en ese tiempo. Entonces para el tiempo t_n se tienen que $i_n = i_{n-1} + e_n - s_n$. En esta estrategia se tomará en cuenta que si $i_n < 0$, entonces se toma ese paso como, $i_n = 0$; en otro caso, se dejará el inventario calculado. Ahora sea el conjunto i_{ns} el conjunto de los pasos $i_n < 0$. Así entre más historia se tenga se esperaría por los supuestos que $\sum_{i \in i_{ns}} |i| \approx i_0$, esto debido a que el inventario calculado de un momento t siempre tiene que ser positivo y el hecho de que $i_n < 0$ quiere decir que al menos se tenía esa cantidad de inventario inicial, lo cual ayuda a que en los últimos pasos se tenga el valor real del inventario. Esta aproximación de inventario en un tiempo t_n diferirá en a lo más i_0.

A partir de lo anterior, para la propuesta de calcular el inventario actual se necesitan las entradas y salidas de los productos en un tiempo lo suficientemente largo. Por lo mismo para conocer la efectividad del método anterior descrito se realizó una simulación donde se repite 10000 veces el método anterior descrito con el objetivo de medir la diferencia entre el valor del inventario actual conociendo y sin conocer el inventario inicial.

Suponiendo que las salidas sean similares al inventario en un tiempo y las entradas similares al valor del inventario anterior, se decidió simular las entradas y salidas de tres maneras, el primero con ayuda de una distribución Uniforme (0, valor inicial), el segundo con ayuda de Normal (valor inicial, valor inicial/8), el tercero es igual una distribución normal con la diferencia de que la media se adapta al valor inicial de cada tiempo.

A partir de lo anterior expuesto, se observó que para la información de entradas y salidas se necesitan más de 20 unidades de tiempo para obtener mediante este algoritmo un 55% del inventario con el valor real del inventario actual, un 15% con a lo más la mitad de i_0 más el valor del inventario real, y el 30% con a lo más i_0 más el inventario real. Estos resultados son consecuencia de que al tomar esa cantidad de historia aparece un valor $i \in i_{ns}$ tal que $i \approx i_0$ o que el conjunto i_{ns} es relativamente grande y que $\sum_{i \in i_{ns}} |i| \approx i_0$.

Adicionalmente, es importante mencionar que, si se toman más de esas unidades de tiempo, el resultado no mejora significativamente. Y es necesario mencionar que, aunque pueda ser un método que parece en una primera instancia algo sencillo, con la cantidad de información que se manejó en cuestión de productos y unidades de tiempo que se utilizaron, el algoritmo finalizó en varias horas para que pudiera dar a conocer la información con el detalle requerido.

Como se puede notar, se puede calcular en general con una gran precisión el inventario del tiempo requerido. Información que, como se ha mencionado, tiene como objetivo poder llevar a cabo la migración del almacén, dejando lo que ya es necesario para la operación. De esta manera, se puede notar que en general se puede proporcionar una solución tratando de manera correcta los elementos con los que el negocio puede proporcionar.

3.3. Adopción de nuevas tecnologías para la centralización de procesos

La digitalización empresarial demanda herramientas efectivas para la gestión de datos, como BigQuery de Google Cloud, una plataforma para análisis de datos en la nube. Ofrece consultas rápidas en grandes volúmenes de datos y se escala según la necesidad, lo que facilita el acceso y análisis de la información de forma segura y confiable (Google Cloud, 2024).

Las soluciones en la nube de Google Cloud ofrecen ventajas significativas en términos de seguridad, escalabilidad, accesibilidad, administración y costos operativos. Además, permiten analizar en tiempo real petabytes de información y facilitan el uso de herramientas de análisis de datos y aprendizaje de máquina. Para analizar nuestros datos, debemos seguir los siguientes pasos.

1. ***Crear o utilizar un proyecto existente***: Para poder crear un proyecto se necesita proveerlo de un nombre y del grupo de personas que tendrán acceso al mismo. De esta manera, se puede segmentar de manera precisa la información a los equipos que lo requieran.

2. ***Crear o utilizar un conjunto de datos en un proyecto***: Se asemeja a una base de datos y se crea con un nombre y un lugar donde alojar el conjunto de datos.

3. ***Crear o utilizar una tabla en un conjunto de datos***: Se crea con un nombre, un lugar donde alojar la tabla, un esquema, con respecto a que se realizara la partición de datos, etcétera.

Los datos que se utilizaron son "House Sales in King County, USA" de Kaggle, el cual contiene precios de casas para el condado de King e incluye Seattle. Se incluye información entre mayo de 2014 y 2015, los datos tienen columnas como precio en dólares, número de cuartos, si cuenta con vista al mar, año de renovación, etc. Para conocer un poco el poder de BigQuery, se realizaron las siguientes consultas.

Cuadro 7. Rendimiento para distintos tipos de consultas

Consulta	Resultado	Tiempo transcurrido	Tiempo de ranura consumido	Bytes mezclados
Precio máximo por código postal y traer toda la información relacionada	Tabla	557 ms	322 ms	646 B
Promedio de precios por número de cuartos por encima del promedio general	Con 8 cuartos cuestan 1105076, etc.	550 ms	322 ms	646 B
Seleccionar primeros 5 elementos	Tabla	475 ms	264 ms	910 B
Número de casas que tienen cierto número de cuartos	Con 3 cuartos se tiene 9824 casas, etc.	330 ms	140 ms	702 B
Número de casas con vista al mar	163	236 ms	30 ms	18 B
Precio promedio de casas renovadas y no renovadas	760379 y 530360 respectivamente	229 ms	33 ms	52 B
Precio promedio	540088	221 ms	32 ms	27 B
Petición 1 - Tabla con 19176586 renglones x 60 columnas	Tabla	1 min 48 seg	29 min 24 seg	33.54 GB
Petición 2	Tabla	4 hr 26 min	1156 días 11 hr	83.3 TB

En esta tabla se observan el desempeño que tuvieron distintas consultas en datos de precios en USA. Para la petición 1 y petición 2 son dos consultas que se tomaron de un entorno con una cantidad de datos que simula un entorno real (Ahmad Kanani, 2023).

Hay que notar que el tiempo transcurrido, aunque hace referencia al tiempo en que se realizó la consulta, por detrás, la plataforma trabajó de manera paralela con distintas ranuras. Una ranura en BigQuery es una unidad de CPU virtual. Entonces, lo que nos dice la columna de "tiempo de ranura consumido" es la suma de tiempo de las ranuras que usó BigQuery en paralelo para poder realizar la petición. Por último, los "Bytes mezclados" es el total de datos que la consulta tuvo que procesar para llegar al resultado deseado.

Como se observa, en una escala algo pequeña los resultados no son diferentes a lo que se podrían obtener, por ejemplo, con SQL Server; hasta podrían ser un poco más tardados porque por detrás, BigQuery toma decisiones como la cantidad de CPU virtuales que utilizará en el proceso. Sin embargo, se tienen dos ejemplos más que son el 1 y el 2, el primero da una visión de lo que sería trabajar con una cantidad considerable de datos, en donde BigQuery termina el trabajo en sólo 2 minutos, mientras que si se procesara en una sola CPU se tardaría alrededor de 30 minutos. Podemos ver cómo esto ya se lleva al límite con terabytes de datos, teniendo el resultado de la consulta en 4 horas y media en BigQuery, mientras que el tiempo consumido de ranuras fue de 1156 días. En este punto, se puede ver lo asombroso que son los resultados.

Con lo anterior visto, que las tecnologías en la nube tienen un impacto significativo en los procesos de la empresa. Ya que centralizan la información, la cual no tiene problemas de escalabilidad, facilitan el acceso a la información y así optimizan el proceso de toma de decisiones. Esta transformación no solo ha permitido explotar más eficazmente los datos, sino que también proporciona una base sólida para la adopción de futuras tecnologías y la mejora continua de los procesos empresariales.

4. Mantenimiento predictivo con ayuda de la inteligencia artificial

El mantenimiento predictivo, con ayuda de la inteligencia artificial, es un elemento crucial en la industria moderna. A través del análisis de grandes volúmenes de datos del funcionamiento de un proceso a lo largo del tiempo, se puede revelar cuándo y dónde podrían ocurrir fallos en los mismos. Esta anticipación posibilita actuar preventivamente, evitando caídas del proceso no programadas que pueden conllevar costos elevados. De esta manera, se mantiene una operación más eficiente, reduciendo los tiempos muertos y ayudando a comprender de manera profunda las causas de los fallos.

Los datos que se utilizaron en este caso son de Kaggle y se llaman “PM_dataset”, son un conjunto de sensores y configuraciones simuladas de motores de turbinas de gas de aviones como una serie de tiempo multivariada, donde cada fila representa una instancia de datos tomados durante un único ciclo operativo. Cabe mencionar que las redes neuronales, anteriormente expuestas, son herramientas muy útiles porque son muy buenas para aprender a partir de secuencias. Aunque el problema se pueda abordar tanto como un enfoque de regresión como de clasificación, todo esto se basa en la definición que se haga de lo que es un fallo en un proceso. En este caso, se enfoca el problema desde el punto de vista de clasificación. Para comprender un poco más la naturaleza de los datos, se realizó una exploración de estos.

Cuadro 8. Número de combinaciones de sensores anómalos relacionados con una falla

Número de combinaciones de sensores anómalos relacionados a una falla			
2 sensores	3 sensores	4 sensores	5 sensores
s2, s8 - 17	s2, s8, s15 - 20	s2, s3, s8, s15 - 20	s2, s3, s8, s15, s17 - 25
s8, s15 - 17	s3, s14, s15 - 7	s2, s8, s15, s17 - 14	s2, s3, s14, s15, s17 - 8
s3, s14 - 10	s2, s3, s14 - 7	s2, s3, s14, s15 - 10	s2, s3, s8, s14, s15 - 8
s14, s15 - 10	s2, s3, s8 - 7	s2, s8, s15, s21 - 5	s2, s8, s15, s17, s21 - 7

Combinaciones de valores anómalos de sensores justo antes de la presencia de una falla.

Cuadro 9. Exploración de los datos para el mantenimiento predictivo

Preguntas por resolver	Resultado
¿Cuál es el tamaño de los datos?	Los datos cuentan con 20631 registros con 26 variables
¿Con qué información cuenta los datos?	Con 3 ajustes y 21 sensores
¿Existen variables que no proveen información?	Se tienen 5 variables con varianza 0, las cuales son setting3, s1, s5, s10, s16, s18, s19
¿Existen variables altamente correlacionadas?	Se tienen 6 columnas altamente correlacionadas, las cuales son s11, s4, s12, s7, s13, s9
¿Cuál es el tiempo entre fallos del proceso?	El tiempo de vida promedio es de 206 días, con un mínimo de 128 a 362 días

En el cuadro se puede observar una visión resumida de la relación que tienen las distintas variables, que tipo de información nos pueden proporcionar y el comportamiento que suele presentar el proceso antes de un fallo.

Como se puede observar, los datos contienen información que no provee nada al modelo. El hecho de que una columna tenga varianza cercana a cero significa que es prácticamente constante. De igual manera, se removieron las columnas correlacionadas, ya que los modelos en general empeoran cuando tienen información de este estilo. También informa sobre cada cuánto sucede un fallo en el proceso y por tanto proporciona la idea de la distribución relacionada con los mismos. A continuación, se explora si existe verdaderamente una diferencia entre los sensores presentados en un estado normal del dispositivo y momentos antes del fallo, con lo cual se observa lo siguiente.

Cuadro 10. Comparación de los valores en los sensores antes del fallo

Variable	Valor normal	Valor anómalo	Apariciones
s2	642.58	643.59	60
s15	8.43	8.51	59
s8	2388.09	2388.24	51
s14	8149.03	8208.77	38
s3	1589.66	1602.73	31
s17	393.12	396.24	27
s21	23.24	23.06	18
s20	38.75	38.44	16

Con la tabla anterior se puede empezar tanto a entender las diferencias entre estados normal y anormal de los sensores, la información de sensores anormales se obtuvo tomando el promedio de los sensores que se presentan 7 unidades de tiempo antes de presentar el error, mientras que la información de los sensores normales se obtuvo con la información antes de la información anterior.

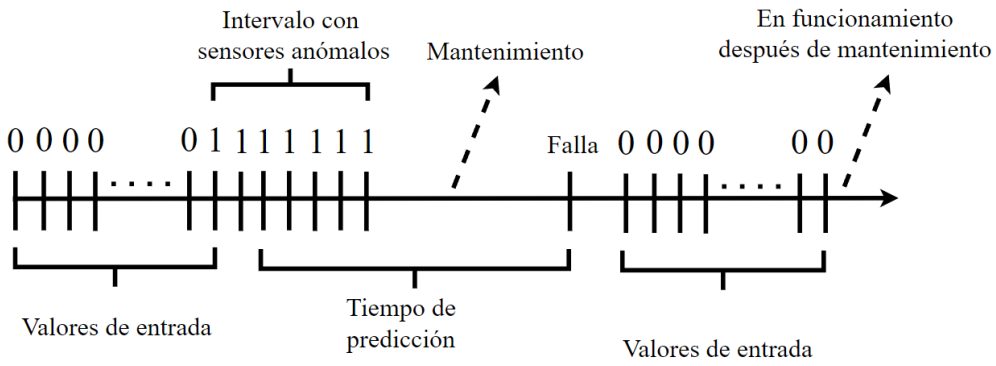
En el primer cuadro, se puede observar las distintas combinaciones de sensores anómalos que se presentan antes de una falla, lo que apoyaría a poder clasificar de mejor manera los distintos casos por los que se genera una falla y el posible plan de acción que se debería de tomar relacionado con ello. En este caso, sería necesario desarrollar un sistema que, en adición a la alerta que se genere, se acompañe de la combinación de sensores que presentaron los valores anormales para poder responder con una solución óptima.

En el segundo cuadro, se observa cómo de manera individual los sensores presentan valores anómalos relacionados con un fallo. En donde, de manera muy marcada, se presenta un aumento en el valor del sensor. Sólo en los casos del sensor s21 y s20, cuando disminuye el sensor, está relacionado con una falla. De esta manera, se puede notar qué tipo de valores

podrían considerarse anormales para cada uno de los sensores y, junto con el entendimiento adecuado de los significados de estos, se puede prever de mejor manera tanto la falla, así como la mejor forma en que se podría dar una solución de manera preventiva.

De esta manera, solo se seleccionarán los sensores del cuadro anterior, ya que en la mayoría de los casos son las que dictan cuándo se presentará un problema. Esto es beneficioso para el modelo, porque será menos complejo y, en general, es mejor crear modelos con menos variables, pero mucho más significativas para la predicción. Es así como se dejarán fuera las variables "s6, setting1, setting2, setting3". A partir de toda la exploración de datos realizada, se propone la siguiente solución presentada en el siguiente diagrama.

Figura 18. Funcionamiento del proceso para el mantenimiento predictivo



Visualización del funcionamiento del modelo de manera productiva en donde se ven todas las facetas por las que tiene que pasar la información y el mantenimiento.

La anterior solución será explicada por partes. Los valores de entrada son el número de valores de los sensores que tomará el modelo para poder realizar una predicción. El tiempo de predicción es el número de días antes que se utilizarán para marcar en los datos como la clase positiva "1", lo cual representará que se requiere de un mantenimiento predictivo. La clase negativa "0" indicará que no se requiere de mantenimiento predictivo.

En este caso, se utilizarán siete unidades de tiempo; sin embargo, como el número de elementos de la clase positiva con respecto a la clase negativa es mucho más pequeño, se

necesitará de alguna manera aumentar la clase positiva. Para ello, se tomará un intervalo alrededor de siete días antes de la falla representando la clase positiva, los cuales estarán relacionados con sensores anómalos, los cuales serán la representación que se necesita para un mantenimiento predictivo. En este caso, como el tiempo utilizado es relativamente poco y como en la presencia de una falla interviene un factor aleatorio, entonces, si el modelo realiza una predicción positiva, el aviso relacionado con una predicción positiva será que se necesita lo antes posible de un mantenimiento preventivo para evitar la falla en el proceso.

Una vez que el modelo realice una predicción mencionando que necesita un mantenimiento preventivo, el modelo se apagará. Durante las unidades de tiempo en las que el modelo esperaría que hubiera una falla, a partir de ahí tendrá que pasar las unidades de tiempo necesarias para que el modelo pueda seguir realizando predicciones. Para poder alcanzar este objetivo, se utilizan los métodos de aprendizaje profundo, anteriormente explicados, enfocados en la clasificación. De la misma manera, se realizó una optimización de parámetros, obteniendo los siguientes resultados.

Cuadro 11. Optimización de parámetros para los modelos de clasificación

	CNN	LSTM
Búsqueda de parámetros	{ n_input:[2, 8, 16], n_filters: [25, 50, 100, 200], n_kernels:[3,5], n_epochs:[15, 20, 30], n_batch:[500, 2000, 8000], n_diff: [0], optimizer:['adam', 'Adamax', 'Adagrad', "SGD"], learn_rate:[0.001, 0.1], init_mode:['uniform', 'lecun_uniform', 'normal', 'glorot_normal'], activation:['linear', 'relu', 'softplus', 'tanh'] }	
Resultados 7 días	Combinaciones por serie 100 Duración: 15 min (16, 100, 3, 15, 500, 0, 'Adamax', 0.001, 'uniform', 'relu') Train 0.97 - Test 0.85 (16, 50, 3, 20, 500, 0, 'SGD', 0.1, 'uniform', 'relu') Train 0.97 Test 0.83 (16, 100, 3, 30, 8000, 0, 'adam', 0.001, 'lecun_uniform', 'linear') Train 0.91 - Test 0.78	Combinaciones por serie 255 Duración: 129 min (16, 200, 3, 30, 500, 0, 'Adamax', 0.001, 'uniform', 'linear') Train 0.96 - Test 0.83 (8, 25, 3, 50, 500, 0, "SGD", 0.1, "uniform", "tanh") Train 0.96 - Test 0.83 (8, 50, 3, 15, 2000, 0, "Adagrad", 0.001, "glorot_normal", "softplus") Train 0.92 - Test 0.52

Mejores parámetros obtenidos después de la aplicación de distintos modelos con el objetivo de alertar de manera temprana un posible falla en el proceso.

La elección de siete días tiene como objetivo que, en un entorno real, se pueda dar un aviso al equipo de mantenimiento del requerimiento de un chequeo en ese mismo instante,

acompañando el aviso de que se tienen pocos días para solucionarlo y de información adicional de los sensores que podrían estar indicando el fallo.

Como se observa en los resultados anteriores, el mejor modelo se encontró con las CNN, teniendo tanto un buen desempeño en los datos de entrenamiento como en los datos de comprobación. Es importante mencionar que se utilizó la métrica f1, la cual tuvo como objetivo ser precisa en la predicción correcta de la necesidad de un mantenimiento predictivo, así como no obtener falsos positivos. Lo anterior se podría traducir en no predecir mucho antes de que suceda un fallo o que prediga muchos más mantenimientos predictivos de los necesarios.

De igual manera, es importante mencionar que, para los modelos LSTM en esta ocasión, aunque parecieran tener métricas muy parecidas, fueron muy sensibles al número de épocas en el que se entrenaban. Por lo que se tenía que seleccionar de manera cuidadosa para obtener buenos resultados, haciendo que los CNN en esta ocasión fueran superiores y se tomen como el modelo final de este ejercicio.

De esta manera, se observa que el mantenimiento predictivo con inteligencia artificial es fundamental en la industria moderna, ofreciendo una herramienta clave para anticipar y mitigar fallos en los procesos y equipos de manera proactiva, mejorando la eficiencia del proceso. Además, conduce a una reducción significativa de los costos asociados a paradas imprevistas y mantenimientos de emergencia. La implementación representa un avance estratégico en la gestión y mantenimiento de los procesos, y está relacionada con prácticas de operación más inteligentes y resilientes.

5. Pasos para llevar una solución a un ambiente productivo

La transición de una solución desde un entorno de desarrollo hasta su implementación en producción es un paso crucial en el desarrollo de un proyecto de ciencia de datos. A continuación, se desglosarán los procedimientos necesarios para llevar las soluciones al escenario productivo, ya que hasta ahora han sido desarrolladas en un entorno de pruebas local. Esto con el objetivo de desempeñar su función de manera continua y eficiente, garantizando que la aplicación sea robusta, estable, escalable y mantenible a largo plazo.

Se revisará la "modularización" de los procesos, una estrategia que implica la división del código en componentes independientes. Paralelamente, se implementará un sistema de generación de logs, esencial para el monitoreo y la resolución de problemas en tiempo real. Adicionalmente, se explicará cómo realizar actualizaciones periódicas en un repositorio de GitHub, garantizando que la documentación y el código sean accesibles para los equipos de desarrollo y operaciones. Facilitando la colaboración entre equipos, la modularización requiere de los siguientes pasos.

1. ***Creación del main***: El objetivo es tener un script que pueda a muy alto nivel correr el programa completo y que dé una idea muy buena al desarrollador del flujo de los distintos módulos que componen tu solución.
2. ***Creación de módulos***: En esta parte lo que se tiene que hacer es dividir en componentes el código ya desarrollado, en los cuales se desarrollarán clases que contengan las funcionalidades de la solución. Intentando generar los componentes necesarios, permitiendo la fácil mantenibilidad de estos.
3. ***Creación del archivo de configuración***: Este archivo tiene como objetivo que de manera sencilla se pueda proveer de los parámetros necesarios para que el proyecto funcione, y que en este mismo sentido puedan cambiarse de manera sencilla sin

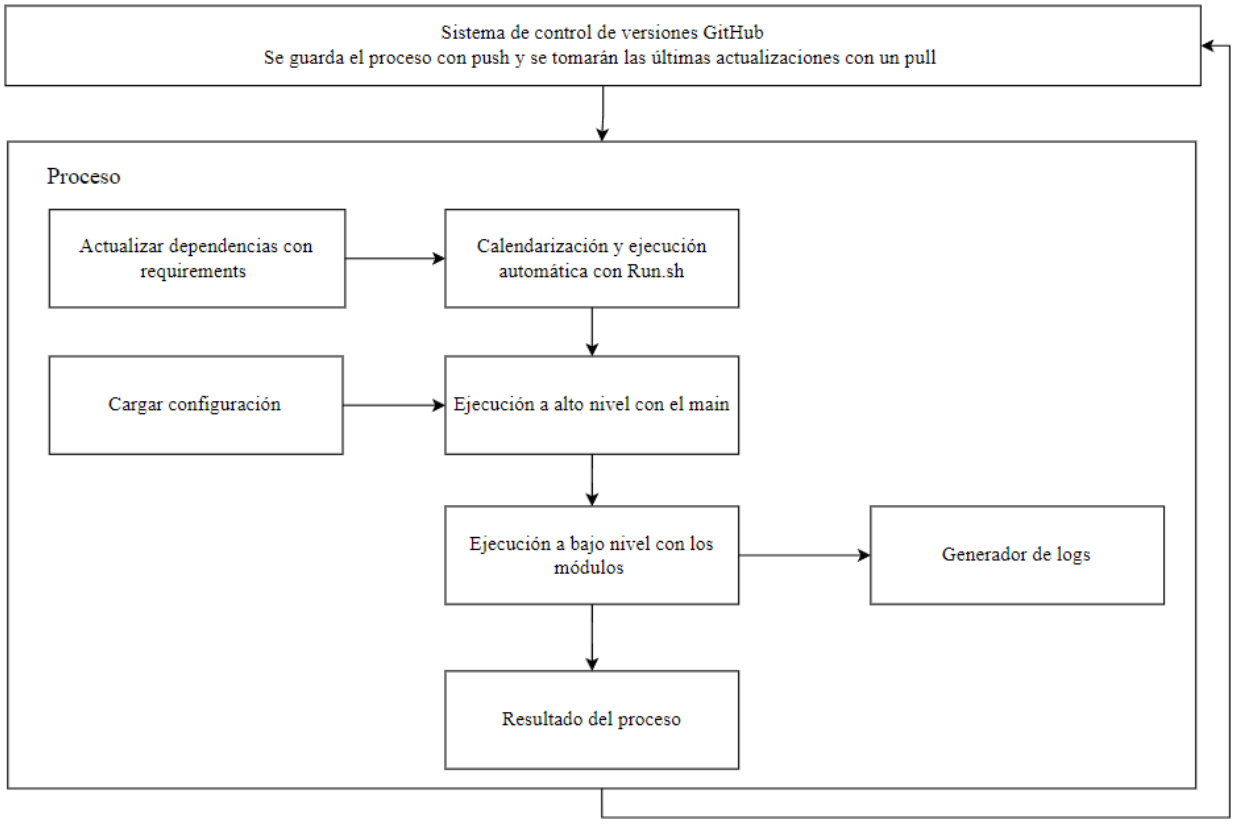
modificar el código directamente. Para que esta configuración pueda ser tomada por el programa, se pasará a los distintos módulos mediante un diccionario.

4. ***Creación de logs***: La importancia de este paso es poder tener control sobre el buen funcionamiento de los distintos módulos dentro del proyecto. De esta manera, todas las veces que el proceso corra, se puede saber a detalle los posibles errores que salgan, los recursos consumidos, excepciones, etcétera.

Adicional a los pasos anteriores, se tiene que agregar una parte llamada “requirements”, que contiene todas las librerías necesarias con las que funciona el proyecto. Esto con el objetivo de tener el mismo ambiente en donde corra el código y donde se desarrolló el proyecto. En caso de ser necesario, se agregará un binario que permite la automatización de la ejecución de un programa. Con lo anterior, se puede decir que ya se ha modularizado la aplicación.

Por otra parte, por cada actualización que se realice, se debe considerar que todos los cambios en el código deben alojarse en un sistema de control de versiones, en este caso, se utiliza GitHub. A muy alto nivel, lo que se tiene que hacer es crear un proyecto en GitHub, donde, por cada actualización que se realice, se utilice el comando push junto con un comentario dejando constancia de los distintos cambios que se realizaron. Al tener los últimos cambios, se facilita que el proceso operativo en ejecución tome los últimos cambios con un pull y de esta manera funcione con los últimos cambios sin ningún problema.

Figura 19. Modularización de un proyecto



En el anterior esquema podemos tanto visualizar los distintos componentes que forman parte de una correcta visualización, así como el orden de la ejecución de estos, dando una idea de cómo se da este ciclo de actualizaciones del proyecto en un ambiente productivo.

Conclusiones

A lo largo de mi trayectoria en la ciencia de datos, he experimentado el modo en que la teoría y la práctica se entrelazan para transformar información en estrategias de negocio concretas y eficaces. Este viaje ha estado marcado por un aprendizaje y adaptación constantes mientras se ha resaltado la visión estratégica y la innovación en diversas áreas tecnológicas. Los modelos y algoritmos que he diseñado reflejan una gran complejidad que está orientada a una integración efectiva en sistemas empresariales complejos.

La transición de un modelo desde su concepción inicial hasta su implementación en sistemas de producción llenos de retos, la modularización y la adaptabilidad son cruciales en un entorno empresarial en evolución, por lo que, en este trabajo se subraya la relevancia de la ingeniería de software y la gestión de proyectos como elementos clave para el éxito de las soluciones en proyectos de ciencia de datos.

He observado cómo el rol del científico de datos se expande más allá del mero conocimiento técnico, ya que es necesario ser un comunicador eficaz, alguien un tanto visionario y un líder. Donde los modelos que construimos deben ser técnicamente sólidos y explicables, ya que un modelo en producción depende de su capacidad para adaptarse y evolucionar. Esta dedicación a la mejora iterativa y la innovación sostiene la vitalidad de los proyectos manejados por científicos de datos.

Durante esta trayectoria he sido testigo de la convergencia de grandes volúmenes de datos, poder computacional y la innovación. Esta confluencia impulsa el progreso y el cambio organizacional. Este trabajo es una reafirmación de que los datos, manejados con habilidad y responsabilidad, son la fuente de verdades poderosas y decisiones transformadoras.

Con miras hacia el futuro, tengo la visión de conservar mi curiosidad insaciable y el rigor intelectual, teniendo en cuenta que la responsabilidad de nosotros, los científicos de datos, es asegurar que nuestra contribución al mundo sea positiva, significativa y duradera.

Más que un reporte, este trabajo escrito es una forma de reconocer el potencial de la ciencia de datos para informar, influir e inspirar el mundo.

Anexos

En la siguiente sección se detallará de manera más precisa el código utilizado para las distintas figuras o secciones del presente trabajo. Es importante mencionar que solo se mostrará en imágenes aquel código que pueda ser totalmente visible en pocas imágenes, mientras que aquel que tenga mucha más complejidad se abordara a un mayor nivel explicando el código realizado, en caso de ser necesario también se explicaran las funciones utilizadas en el mismo.

Generación de figuras

Para la figura 1 que representa el puntaje z se utilizó el siguiente código.

Figura 20. Código puntaje z

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.stats import norm

# Configurar el tamaño de la figura
plt.figure(figsize=(10, 6))
# Parámetros de la distribución normal
mu = 0
sigma = 1
x = np.linspace(mu - 3 * sigma, mu + 3 * sigma, 100)
pdf = norm.pdf(x, mu, sigma)
colors = ['blue', 'green', 'red']

# Rellenar el área bajo la curva
plt.fill_between(x, pdf, where=[abs(i - mu) < 3 * sigma for i in x], color=colors[0], alpha=0.5, label='3 Desviaciones Estándar')
plt.fill_between(x, pdf, where=[abs(i - mu) < 2 * sigma for i in x], color=colors[1], alpha=0.5, label='2 Desviaciones Estándar')
plt.fill_between(x, pdf, where=[abs(i - mu) < 1 * sigma for i in x], color=colors[2], alpha=0.5, label='1 Desviación Estándar')

# Crear la gráfica de la distribución normal
plt.plot(x, pdf, label='Distribución Normal', color='black')
plt.text(mu, pdf.max() * -0.2, '68% en ±1 Desviación Estándar', horizontalalignment="center" ,verticalalignment="bottom",  color=colors[2])
plt.text(mu, pdf.max() * -0.25, '95% en ±2 Desviaciones Estándar', horizontalalignment="center" ,verticalalignment="bottom", color=colors[1])
plt.text(mu, pdf.max() * -0.3, '99.7% en ±3 Desviaciones Estándar', horizontalalignment="center",verticalalignment="bottom", color=colors[0])

# Configuración de la gráfica
plt.xlabel('Valores')
plt.ylabel('Densidad de Probabilidad')
plt.title('Distribución Normal y Áreas bajo Desviaciones Estándar')
plt.legend(loc='upper left')
plt.grid(True)
plt.subplots_adjust(bottom=0.15)
plt.show()
```

En la imagen se muestra el código utilizado para la generación de los valores que representaran la función de densidad de probabilidad de una distribución normal, para después con ayuda de librerías de visualización como mathplotlib y scipy poder generar una visualización relacionada al puntaje z.

En el código se observa la construcción de la distribución normal, posteriormente se pinta el área de la curva correspondiente a distintas desviaciones estándar, por último, se agregan los porcentajes de población en cada rango de desviaciones estándar.

Para la figura 2, asociada a los valores atípicos detectados por percentiles se utilizó el siguiente código.

Figura 21. Código de detección por percentiles

```
from scipy.stats.mstats import winsorize
import numpy as np
import matplotlib.pyplot as plt

# Generar un conjunto de datos con algunos valores extremos
np.random.seed(42)
data = np.concatenate((np.random.normal(0, 1, 1000), np.random.normal(8, 0.5, 10)))
# Aplicar detección de valores atípicos por percentiles al conjunto de datos, limitando los valores a los percentiles 5 y 95
winsorized_data = winsorize(data, limits=[0.05, 0.05])
plt.figure(figsize=(14, 7))
# Diagrama de dispersión del conjunto de datos original vs datos sin valores atípicos
plt.subplot(1, 2, 2)
plt.scatter(range(len(data)), data, alpha=0.5, label='Datos originales', c='blue')
plt.scatter(range(len(winsorized_data)), winsorized_data, alpha=0.5, label='Datos con recorte del 10%', c='orange')
plt.title('Datos originales vs datos sin valores atípicos')
plt.xlabel('Índice')
plt.ylabel('Valor')
plt.legend()
plt.tight_layout()
plt.show()
```

Se muestra el código para la generación de la figura relacionada a la detección por percentiles, primero se generan datos sintéticos, los cuales se les aplica un método por percentiles para la detección por valores atípicos, por último, se utiliza funciones para la visualización de datos.

En la imagen se puede observar cómo primero se crean dos conjuntos de datos de una distribución de datos normales, con ayuda de la función winsorize se logran aislar los datos que están dentro de un intervalo del 90%, los cuales posteriormente se grafican para tener una representación visual.

Para la figura 3 de valores atípicos detectados por DBSCAN se utilizó el siguiente código.

Figura 22. Código DBSCAN

```
import matplotlib.pyplot as plt
from sklearn.cluster import DBSCAN
import numpy as np
from sklearn.datasets import make_blobs

X, _ = make_blobs(n_samples=750, centers=3, cluster_std=0.4, random_state=0)

# Aplicamos DBSCAN al nuevo dataset
db = DBSCAN(eps=0.3, min_samples=10)
db.fit(X)
labels = db.labels_

# Número de clusters en labels, ignorando ruido si está presente.
n_clusters_ = len(set(labels)) - (1 if -1 in labels else 0)

# Graficamos los resultados
unique_labels = set(labels)
colors = [plt.cm.Spectral(each) for each in np.linspace(0, 1, len(unique_labels))]
plt.figure(figsize=(10, 8))

for k, col in zip(unique_labels, colors):

    class_member_mask = (labels == k)

    xy = X[class_member_mask]
    plt.plot(xy[:, 0], xy[:, 1], 'o', markerfacecolor=tuple(col), markeredgecolor='k', markersize=6)

plt.title('Número estimado de clusters: %d' % n_clusters_)
plt.xlabel('Característica 1')
plt.ylabel('Característica 2')
plt.show()
```

En la imagen se muestra primero la generación de valores con ayuda de la función make_blobs, para después aplicarle el método DBSCAN para la detección de valores atípicos y por último visualizar los resultados con mathplotlib.

Aquí se puede observar que primero se crean tres agrupaciones de datos, a los cuales se les aplica el método DBSCAN para conocer los clústeres que se observan en los datos, a los cuales se les asocia un color para poder presentarlo de manera gráfica con las librerías usuales de visualización en Python como lo es mathplotlib.

Para las figuras 4 y 5 relacionados a los bosques de aislamiento se utilizó el siguiente código.

Figura 23. Código bosques de aislamiento

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.ensemble import IsolationForest
from sklearn.datasets import make_blobs

# Generar un conjunto de datos sintético
np.random.seed(42)
X, _ = make_blobs(n_samples=200, centers=1, cluster_std=0.6, random_state=42)
X_outliers = np.random.uniform(low=-6, high=6, size=(20, 2))
X = np.r_[X, X_outliers]
# Entrenar el modelo Isolation Forest
clf = IsolationForest(random_state=42, contamination=0.1)
clf.fit(X)
# Predecir si un punto es atípico o no (1 para típicos, -1 para atípicos)
y_pred = clf.predict(X)
# Obtener la puntuación de anomalía (cuanto más bajo, más anómalo)
scores_pred = clf.decision_function(X)
# Visualizar el conjunto de datos y la puntuación de anomalía
plt.figure(figsize=(10, 7))
plt.scatter(X[y_pred == 1][:, 0], X[y_pred == 1][:, 1], c='green', label='Normal')
plt.scatter(X[y_pred == -1][:, 0], X[y_pred == -1][:, 1], c='red', label='Anomaly')
plt.scatter(X[:, 0], X[:, 1], s=1000 * np.abs(scores_pred), edgecolors='k',
            facecolors='none', label='Anomaly Score')
plt.legend()
plt.title("Isolation Forest Anomaly Detection")
plt.xlabel('Feature 1')
plt.ylabel('Feature 2')
plt.grid(True)
plt.show()

from sklearn.tree import plot_tree

clf_single_tree = IsolationForest(n_estimators=3, max_samples=10, bootstrap=False,
                                  random_state=42, contamination=0.1,max_features=1)
clf_single_tree.fit(X)
# Obtener el árbol único del modelo Isolation Forest
estimator = clf_single_tree.estimators_[3]
# Visualizar el árbol único
plt.figure(figsize=(20, 10))
plot_tree(estimator, filled=True, feature_names=['Característica 1', 'Característica 2'], rounded=True, proportion=False)
plt.title('Single Tree Visualization from Isolation Forest')
plt.show()
```

Para poder generar las figuras 4 y 5 se generan un conjunto de datos sintéticos con la función make_blobs, con los cuales se entrena el método de detección de valores atípicos llamado bosque de aislamiento con la función “IsolationForest”, para después poder visualizarlos con mathplotlib.

Aquí nuevamente se crea agrupamientos de datos y se agregan algunos valores atípicos, posteriormente se aplica el método de bosques de aislamiento para detectar los valores atípicos y sus puntuaciones de anomalía, con esta información se procede a representarlo de manera gráfica. Por otra parte, para tener una representación de un bosque de aislamiento se utilizó uno de sus árboles para poder visualizarlo.

Desarrollo del problema de precios atípicos

Para la sección en la que se habló sobre “detección de precios atípicos” se utilizan datos relacionados a productos globales, en donde para poder tener una visión general de los datos se respondieron una serie de preguntas del cuadro dos con el siguiente código.

Figura 24. Código de la exploración de los precios atípicos

```
#Revisión de algunas preguntas
rows=dat_food.shape[0]
print("Los datos tienen "+str(dat_food.shape[0])+" muestras y tienen "+str(dat_food.shape[1])+" columnas")
print("La unica columna que tiene nulos es localidad, la cual tiene "+str(dat_food.isnull().sum().sum()))
print("El número de productos unicos es "+str(len(dat_food.commodity_purchase_id.unique())))
print("El número de monedas unicas es "+str(len(dat_food.name_of_currency.unique())))
print("El porcentaje de productos comprados en unidad de peso es "+format(dat_food[dat_food.unit_of_goods_measurement.str.\
contains("(?<=\s)?KG(?=\s)?|(?<=\s)G(?=\s)?")].shape[0]/rows,".2%"))
print("Los datos fueron tomados desde "+str(dat_food.year_recorded.min())+" hasta "+str(dat_food.year_recorded.max()))
print("El porcentaje de datos tomados despues del 2000 es "+format(dat_food[dat_food.year_recorded>1999].shape[0]/rows,".2%"))

dat_food.unit_of_goods_measurement.value_counts()[:20]
```

En el código se muestra la exploración de datos realizada con ayuda de la librería pandas, para tener la visualización de la información de manera más sencilla se imprimió en pantalla junto a un texto en específico.

En el anterior código se utilizan funciones asociadas a la exploración del conjunto de datos para saber su tamaño, la cantidad de valores nulos, los valores únicos de columnas, adicionalmente se realizó un análisis de textos y estadísticos por columnas. Para las preguntas restantes del “cuadro dos” se cuenta por año el número de productos únicos para conocer si siempre se registraron el mismo número de productos por año, y por último se realizan una serie de filtros para obtener una base de datos lo más completa y fácil de manejar posible, estos filtros toman en cuenta los datos después del 2000, que fueron recolectados con unidades de medida de “KG” y “G”, y que no se colectaron con las divisas NIS o Somaliland.

Para que los datos fueran comparables se calculó un factor con respecto al peso para saber cuánto costaría un KG de producto, por otra parte, se utilizaron datos históricos de los cambios de divisa a dólar después del 2000 para tener el precio final por un KG de producto.

Posteriormente se crearon las siguientes funciones para aplicar los métodos de detección de valores atípicos antes mencionados.

1.-Grubbs_test: En la función se realiza la prueba de Grubbs, adicionalmente aquellos grupos que tengas una desviación estándar menor a 0.1 se descartan, ya que eso quiere decir que la variabilidad en los datos en tan pequeña que los posibles valores atípicos que presente no presentarían un problema.

2.-Zscore_outlier: Teniendo ordenados los datos de precio, se extraen los datos pertenecientes al intervalo del percentil 5 y 95, esto con el objetivo de reducir la influencia de valores atípicos en el cálculo de la media y la desviación estándar, posteriormente se aplica la metodología de la puntuación z para conocer los valores atípicos.

3.-iqr_outliers: En esta función se calculan los percentiles 1 y 99, posteriormente se cataloga como un valor atípico si no pertenecen a este intervalo.

4.-DB_outliers: En esta función se utiliza el método DBSCAN con la métrica euclidiana y un mínimo de muestras por grupo de tres.

5.-Iso_outliers: Aquí solamente se utiliza el método de bosque de aislamiento.

Una vez teniendo estas funciones se genera una lista de registros únicos compuestos por año, país y producto comprado, de esta manera se consiguen grupos con una cantidad suficiente de precios colectados para que funcionen de mejor manera los métodos. Posteriormente se aplica la prueba de Grubbs para descartar grupos con precios normales, y los que puedan contar con algún valor atípico se le aplican los métodos restantes para catalogar todos los precios. Teniendo catalogados los precios de la manera anterior descrita se pueden obtener los productos o países que presentan una mayor cantidad de valores atípicos.

Proceso sobre la generación de flujos de información para la planeación logística

Para la sección de “generación de flujos de información para la planeación logística” se utilizaron datos de ventas jerárquicos de Walmart, en los cuales se encontraron varias características asociadas a los datos, como el tamaño de estos, productos únicos por categoría, los eventos que contiene los datos, las variables externas a la serie que se van a usar y la correlación entre los datos de la serie de tiempo.

Debido al comportamiento de las ventas por productos, en donde muchas de ellas presentaban ventas cero y en donde la cantidad de series de tiempo por productos es realmente grande, se tomó la decisión de enfocarse en estudiar las series de tiempo por categoría para que se parezca más al problema presentado en este trabajo, de esta manera lo primero que se realizo fue la agrupación de ventas por categoría.

Figura 25. Código del tratamiento de variables externas de la planeación logística

```
from sklearn.preprocessing import MinMaxScaler

dum=pd.get_dummies(cal[["event_type_1","event_type_2"]].fillna("Normal"),dtype="int")
"cultural normal religius"
for i in ["Cultural","Normal","Religious"]:
    dum["event_type_1_"+i]=dum["event_type_1_"+i]+dum["event_type_2_"+i]
    dum=dum.drop("event_type_2_"+i,axis=1)

scaler = MinMaxScaler()

dum = pd.DataFrame(scaler.fit_transform(dum),columns=dum.columns)
cal1=pd.concat([cal.drop(["d","weekday","event_name_1","event_type_1","event_name_2","event_type_2","snap_CA","snap_TX","snap_WI"],axis=1),dum],axis=1)
scaler = MinMaxScaler()

dum2=cal1[["wm_yr_wk","wday","month","year"]]
dum2=pd.DataFrame(scaler.fit_transform(dum2),columns=dum2.columns)

cal2=pd.concat([cal1.drop(["wm_yr_wk","wday","month","year"],axis=1),dum2,pd.get_dummies(cal["event_name_1"].fillna("Normal1"),dtype="int")],axis=1)
cal2.head()
```

Para el tratamiento de las variables externas se aplicaron funciones relacionadas al tipo de variables categóricas y para la normalización de los datos, adicionalmente se crearon nuevas variables para representar las interacciones que tienen los días festivos durante el año.

Por otra parte, se tienen que tratar las variables externas, entre las cuales se tienen dos columnas llamadas “evento_type_1” y “evento_type_2”, como estas son texto primero las juntaremos para hacer una llave única y de esta manera simular la interacción de estas dos variables, los cuales se les aplicara la función get_dummies para conseguir un valor numérico por cada combinación de días especiales asociados a la serie de tiempo.

De igual manera, las variables de día, semana, mes y año, se les aplicara una normalización para que ayuden de mejor manera al modelo con la función MinMaxScaler, adicionalmente se agregan al conjunto de datos final de variables externas los datos sin modificar asociados al tiempo para poder manejar de manera correcta la serie de tiempo.

Figura 26. Código para la búsqueda de los mejores parámetros SARIMAX

```
import pandas as pd
import numpy as np
from statsmodels.tsa.statespace.sarimax import SARIMAX
import itertools
import warnings

cat='FOODS'
past_sales
items_col = [c for c in past_sales.columns if cat in c]
past_cat=past_sales[items_col] \
        .sum(axis=1)
warnings.filterwarnings('ignore')
combinations = list(itertools.product([1,2], repeat=6))
score=[]
data_exo=pd.DataFrame(past_cat,columns=[cat]).merge(cal2.set_index("date"),left_index=True, right_index=True)

for combo in combinations:
    order = combo[:3]
    seasonal_order = tuple(list(combo[3:])+[order_max])
    model = SARIMAX(endog=data_exo[cat],exog=data_exo.drop(cat,axis=1), order=order, seasonal_order=seasonal_order)
    model.initialize_approximate_diffuse()
    results = model.fit(disp=False)
    score.append(results.bic)

score_f=score[score.index(min(score))]
comb_f=combinations[score.index(min(score))]
order =combo[:3]
seasonal_order = tuple(list(combo[3:])+[7])
```

Se muestra que para un conjunto de datos de una categoría especifica se le aplica modelos de tipo SARIMAX con el objetivo de encontrar el mejor de ellos con respecto a métrica de BIC.

Teniendo los datos listos, pasemos a encontrar los mejores parámetros para modelar las series de tiempo de venta por categoría con SARIMAX, para ello se utilizó la función que provee la librería statsmodels, en donde los parámetros del modelo pueden tomar valores de uno a tres. Para encontrar el mejor modelo se utilizó la métrica BIC, es importante mencionar que, para realizar las predicciones con variables externas, tienes que también tener la información futura de estas, sin embargo, como todas las variables externas están asociadas al tiempo no tienen mayor complicación en conseguirlas.

Para buscar los mejores parámetros en las redes neuronales tenemos que tomar en cuenta muchas más cosas, en especial un tratamiento para las series de tiempo, para ello se utilizaron una serie de funciones diseñadas para este tipo de problemas, permitiendo manejarlos de mejor manera con una mayor personalización.

1.-train_test_split: Esta función toma en cuenta que los datos son una serie de tiempo, entonces para generar un conjunto de entrenamiento y de prueba se realiza a partir de un punto especifico en el tiempo, partiendo los conjuntos de datos en entrenamiento y de prueba antes y después del punto en el tiempo seleccionado respectivamente.

2.-difference: Aplicar diferencias en las series de tiempo puede tener muchos beneficios, esta función las aplica con respecto a un parámetro dado.

3.-dataset_train: Antes de crear el conjunto de datos con el que el modelo interpretara el problema se tiene que manejar los datos de una manera especial transformándolos en arreglos, y adicionalmente como queremos que los mismos datos de la serie en un tiempo pasado describan el futuro es aquí donde se ajustan los datos de una manera concreta.

4.-split_sequences: Ya que en las redes neuronales necesitamos una cantidad de información de entrada y una cantidad de información de salida, entonces tenemos que transformar el conjunto de datos que tenemos a algo que pueda ser interpretado por el modelo y que cumpla los objetivos buscados.

5.-model_fit: Tiene la función de ajustar el modelo, con base a los parámetros de entrada, verifica si es necesaria una diferenciación con ayuda de la función difference, posteriormente da un preprocesamiento a los datos con dataset_train, es entonces que genera un conjunto de datos interpretable por el modelo con la función split_sequences, y entrena un modelo ya sea LSTM o CNN según sea el caso tomando en cuenta los parámetros de entrada, teniendo como capa final en este caso un valor real para resolver este problema de series de tiempo, regresando el modelo entrenado.

6.-model_predict: Teniendo en cuenta que solo se conoce los datos de entrada esta es una función auxiliar que predice los siguientes pasos tomando en cuenta si se realizó una diferenciación o no.

7.-walk_forward_validation: En esta función genera el conjunto de datos de entrenamiento y de testeo, posteriormente ajusta el modelo con la función model_fit, para después evaluar el modelo con el error cuadrático medio, utilizando la función model_predict paso por paso.

8.-repeat_evaluate: Con base a lo exhaustivo que quieres que sea la búsqueda y la fiabilidad de tus resultados es que en esta función por cada conjunto de parámetros se puede repetir el entrenamiento del modelo, ya que pueden intervenir factores estocásticos, entonces obtiene en promedio el desempeño del modelo, aquí se escoge el modelo que se quiere entrenar (entre LSTM o CNN).

9.-grid_search: Es la función que maneja las puntuaciones por tipo de configuración de los modelos, es la función principal de la búsqueda de los mejores parámetros.

10.-model_configs: Es la función que guarda todas las configuraciones de los modelos y la cual permite ser más exhaustivos en la búsqueda de parámetros.

De esta manera utilizando la función grid_search es que se consiguen los mejores parámetros tomando en consideración la información anterior descrita por tipo de modelos y por serie de tiempo para conseguir los resultados finales, los cuales se tomaran en cuenta y se compararan para escoger el mejor modelo que cumpla con los objetivos de la empresa.

Simulación aplicada en soluciones ante situaciones de urgencia

Para la sección que habla sobre las “soluciones ante situaciones de urgencia” se utilizó el siguiente código para la simulación del problema propuesto en este trabajo.

Figura 27. Código para soluciones ante situaciones de urgencia

```
import pandas as pd
import numpy as np
n_repeat=10000
hist_n=24
dif_list,c_ci_li,c_c_li=[[],[],[]]
for j in range(n_repeat):
    correction=0
    salidas_new,corr_sal=[[],[]]
    entradas=np.random.randint(1, 100, size=hist_n)
    salidas=np.random.randint(1, 100, size=hist_n)
    init,sin_init,count_ci,count_c=[100,0,0,0]
    #corrección de los valores aleatorios
    for r in range(hist_n):
        init=init+entradas[r]-salidas[r]
        if init<0:
            salidas_new.append(salidas[r]+init)
        else:
            salidas_new.append(salidas[r])
    salidas=salidas_new
    init=100
    for i in range(hist_n):
        init=init+entradas[i]-salidas[i]
        if init<0:
            init=0
            count_ci=count_ci+1
        else :
            pass
        sin_init=sin_init+entradas[i]-salidas[i]
        if sin_init<0:
            sin_init=0
            count_c=count_c+1
        else:
            pass
    c_ci_li.append(count_ci)
    c_c_li.append(count_c)
    dif_list.append(init-sin_init)
z=pd.DataFrame({"dif":dif_list,"count_ci":c_ci_li,"count_c":c_c_li})
print("media :"+str(z.mean()))
print("varianza :"+str(z.var()))
z.hist()
```

Bajo distintas distribución, en este ejemplo bajo una distribución normal, se generaron los valores de entradas y salidas, para después calcular la diferencia entre los valores del inventario final considerando y sin considerar el inventario inicial.

Para generar esta simulación se consideran algunos parámetros, como n_repeat que es el número de repeticiones del experimento para tener resultados más confiables, hist_n que es el número de unidades de tiempo que se utilizan para conseguir el inventario actual, el

inventario inicial que en principio es desconocido, pero que se considera para conocer la diferencia entre el cálculo del inventario final considerándolo y sin considerarlo.

Lo primero que se realiza depende del tipo de distribución que genera los valores aleatorios de entradas y salidas, ya que estos valores son aleatorios lo primero que se tiene que hacer es que con respecto al inventario inicial tengan sentido estas entradas y salidas generadas , por lo que en caso de que el inventario en un tiempo dado sea negativo se ajustan estos valores para tener un inventario cero, ya teniendo las cantidades correctas se aplica el algoritmo descrito en el trabajo y se mide la diferencia entre el inventario final tomando él cuenta el inventario inicial y sin contarlo, lo cual genera los resultados expuestos en el trabajo.

Detalle de consultas realizadas en BigQuery

Para la construcción del cuadro siete relacionado al rendimiento de distintos tipos de consultas, la cual está en la sección que habla sobre la adopción de nuevas tecnologías se utilizaron las siguientes consultas.

Figura 28. Consultas realizadas en BigQuery

```
#Seleccionar primeros 5 elementos
SELECT * FROM `analisis-big-query.Datos_prueba.Precios_casas` LIMIT 5;
#Precio promedio
SELECT AVG(price) as precio_promedio FROM `analisis-big-query.Datos_prueba.Precios_casas`;
#Número de casas con vista al mar
SELECT COUNT(*) as casas_con_vista_al_agua FROM `analisis-big-query.Datos_prueba.Precios_casas` WHERE waterfront = 1;
#Número de casas que tienen cierto número de cuartos
SELECT bedrooms, COUNT(*) as cantidad_de_casas FROM `analisis-big-query.Datos_prueba.Precios_casas` GROUP BY bedrooms ORDER BY bedrooms;
#Promedio de precios por número de cuartos por encima del promedio general
SELECT bedrooms, AVG(price) as precio_promedio FROM `analisis-big-query.Datos_prueba.Precios_casas` GROUP BY
bedrooms HAVING AVG(price) > (SELECT AVG(price) FROM `analisis-big-query.Datos_prueba.Precios_casas`) ORDER BY bedrooms;
#Precio máximo por código postal
WITH casas_caras AS (SELECT zipcode, MAX(price) as precio_maximo FROM `analisis-big-query.Datos_prueba.Precios_casas` GROUP BY zipcode)
SELECT t.* FROM `analisis-big-query.Datos_prueba.Precios_casas` t JOIN casas_caras c ON t.zipcode = c.zipcode AND t.price = c.precio_maximo;
#Precio promedio de casas renovadas y no renovadas
SELECT AVG(CASE WHEN yr_renovated > 0 THEN price ELSE 0 END) as precio_promedio_renovadas,AVG(CASE WHEN yr_renovated = 0 THEN price ELSE 0 END) as
precio_promedio_no_renovadas FROM `analisis-big-query.Datos_prueba.Precios_casas`;
```

Las consultas mostradas se crearon con ayuda del lenguaje SQL, en donde la información obtener se puede apreciar como comentario.

Desarrollo de la solución aplicada en el mantenimiento predictivo

Para la sección de que habla sobre el “mantenimiento predictivo” y en especial para el cuadro nueve se realizó una exploración de datos relacionada a el tamaño asociado a los datos, las variables más importantes, se realizó un análisis de varianza para conocer cuáles de ellas no presentaban información útil al estudio, de igual manera se buscó cuales variables estaban altamente correlacionadas, y se empezó a explorar la vida promedio del proceso para ver si tenía algún tipo de patrón.

Figura 29. Código mantenimiento predictivo – preparación de datos

```
#Preparación de datos
rul = pd.DataFrame(train_df.groupby('id')['cycle'].max()).reset_index()
rul.columns = ['id', 'max']
train_df = train_df.merge(rul, on=['id'], how='left')
train_df['RUL'] = train_df['max'] - train_df['cycle']
train_df.drop('max', axis=1, inplace=True)
w1 = 7
w0 = 90
train_df['label1'] = np.where((train_df['RUL'] <= w1+2) & (train_df['RUL'] >= w1-4), 1, 0 )
train_df['label2'] = np.where(train_df['RUL'] <= w0, 1, 0 )
train_df["aux"]=np.where((train_df['RUL'] < w1-4) , 1, 0 )
train_df=train_df[train_df["aux"]==0].drop("aux",axis=1)
# MinMax normalization (from 0 to 1)
train_df['cycle_norm'] = train_df['cycle']
explor=train_df.copy()
cols_normalize = train_df.columns.difference(['id','cycle','RUL','label1','label2'])
min_max_scaler = preprocessing.MinMaxScaler()
norm_train_df = pd.DataFrame(min_max_scaler.fit_transform(train_df[cols_normalize]),
                             columns=cols_normalize,
                             index=train_df.index)
join_df = train_df[train_df.columns.difference(cols_normalize)].join(norm_train_df)
train_df = join_df.reindex(columns = train_df.columns)

train_df=train_df[["id","label1","cycle_norm",'s15', 's2', 's8', 's17', 's14', 's20', 's3', 's21']]
train_df.shape
```

En esta preparación de datos se asigna la etiqueta relacionada a la predicción de mantenimiento predictivo, además se la selección de variables con las cuales se hará la predicción.

Figura 30. Código mantenimiento predictivo – Análisis de datos

```
#preparación para importancia de valores
explorz=pd.concat([explor[explor["label1"]==1].groupby("id",as_index=False).mean(),explor[explor["label1"]==0].groupby("id",as_index=False).mean()]).sort_values("id")
explorz=explorz.drop(["cycle_norm"],axis=1)
explorz1=explorz[explorz["label1"]==1].melt("id",explorz.drop(["id","label1"],axis=1)).merge(explorz[explorz["label1"]==0].melt("id",explorz.drop(["id","label1"],axis=1)),on=["id","variable"])
explorz1["change"]=abs(1-explorz1["value_x"]/explorz1["value_y"])
result1=explorz1.sort_values(["id","change"],ascending=False)

explorz=pd.concat([train_df[train_df["label1"]==1].groupby("id",as_index=False).mean(),train_df[train_df["label1"]==0].groupby("id",as_index=False).mean()]).sort_values("id")
explorz=explorz.drop(["cycle_norm"],axis=1)
explorz1=explorz[explorz["label1"]==1].melt("id",explorz.drop(["id","label1"],axis=1)).merge(explorz[explorz["label1"]==0].melt("id",explorz.drop(["id","label1"],axis=1)),on=["id","variable"])
explorz1["change"]=abs(1-explorz1["value_x"]/explorz1["value_y"])
result2=result1.merge(explorz1,on=["id","variable"]).sort_values(["id","change_y"],ascending=False)
top_var=result2.groupby(["id"]).head(3)
top_var["variable"].value_counts()
```

En esta parte se exploraron el tipo de valores que presentan los distintos sensores antes y durante la falla con ayuda del etiquetado anteriormente echo, con el fin de entender de una manera más profunda el comportamiento de estos.

Siguiendo esta misma línea, se planteaba explorar las diferencias que presentan los datos de los sensores antes y después de presentar un fallo, por lo que tomando una versión normalizada de los sensores, se calcula el promedio de los valores siete días antes del fallo y se calcula un promedio de los valores normales, se comparan porcentualmente por cada uno de los ciclos de fallo los tres que presentan un mayor cambio y se cuenta el número de apariciones tanto de manera individual como conjunta, adicionalmente se toma el promedio de cambio de los valores de los sensores, este análisis se realizó con los objetivos ya mencionados en el presente trabajo.

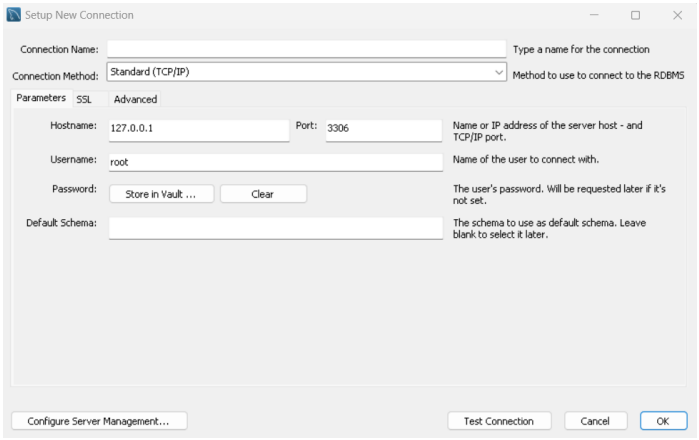
Debido a la naturaleza parecida del problema como una secuencia de datos que tiene como resultado un siguiente valor a predecir, el código utilizado para modelar que se utilizó es prácticamente el mismo explicado para la sección de "generación de flujos de información para la planeación logística", con la excepción más importante del cambio de enfoque en clasificación, lo cual se hace añadiendo una función de activación en la parte final de la red neuronal y ya en un caso real el enfoque diferente que se tendría que tener para que un modelo así funcione de manera productiva.

Desarrollo de los componentes del proyecto desde la perspectiva de producción

Por último, revisaremos la propuesta del desarrollo de un proyecto simulando un ambiente profesional, para ello, abordaremos de manera más detallada el problema "logística efectiva para la satisfacción del cliente". Primero tendremos que generar fuente de datos, de donde extraeremos los datos para el desarrollo del modelo, para ello utilizaremos MySQL Workbench para el almacenamiento de nuestros datos.

Primero se generará una nueva conexión en la cual se deben proporcionar algunos datos, como el nombre de la conexión, usuario, contraseña, puerto de conexión, etc.

Figura 31. Creación de nueva conexión con MySQL Workbench



En la imagen se presenta la interfaz de MySQL Workbench para la creación de una conexión a la base de datos que se utilizará en el proyecto.

Ahora realizaremos la creación de una tabla con respecto a un esquema que se ve más en un sentido productivo, en donde se tiene el detalle por fecha de las ventas por categoría y su información relacionada, de igual manera se generara la tabla relacionada a la información por fecha, la cual contiene la información relacionada a las fechas y la cual se utilizará como variables externas.

Figura 32. Creación de tabla y carga de datos – ventas por categoría

```
CREATE DATABASE IF NOT exists demanda_productos;
USE demanda_productos;
CREATE TABLE prueba_ventas
(
ID VARCHAR(255),
ITEM_ID VARCHAR(60),
DEPT_ID VARCHAR(60),
CAT_ID VARCHAR(60),
STORE_ID VARCHAR(60),
STATE_ID VARCHAR(60),
TEXT_DAY VARCHAR(60),
VALUE INT
);
LOAD DATA INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/dat_dem_sql.CSV"
INTO TABLE prueba_ventas
FIELDS TERMINATED BY ","
IGNORE 1 LINES;
```

Se muestran los pasos a seguir para la creación y carga de la tabla que contendrá la información relacionada a las ventas, se hizo de esta manera ya que el archivo que contenía los datos a cargar tenía un peso relativamente alto.

Figura 33. Creación de tabla y carga de datos – variables externas

```
CREATE TABLE caracteristics_dates
(
DATES DATE,
WM_YR_WK int,
WEEKDAY VARCHAR(60),
WDAY int,
MONTH int,
YEAR int,
TEXT_DAY VARCHAR(60),
EVENT_NAME_1 VARCHAR(60),
EVENT_TYPE_1 VARCHAR(60),
EVENT_NAME_2 VARCHAR(60),
EVENT_TYPE_2 VARCHAR(60),
SNAP_CA int,
SNAP_TX int,
SNAP_WI int
);
LOAD DATA INFILE "C:/ProgramData/MySQL/MySQL Server 8.0/Uploads/calendar.csv"
INTO TABLE caracteristics_dates
FIELDS TERMINATED BY ","
IGNORE 1 LINES;
```

Se muestra la creación de la tabla que contendrá las variables externas relacionadas a las fechas importantes.

Teniendo las tablas en nuestro gestor de base de datos generaremos una consulta con la cual podremos extraer la información de ventas por categoría totales por fecha, además de tener la información externa relacionada a la fecha, esta consulta devolverá distinta información de acuerdo con la categoría que estemos estudiando.

Figura 34. Obtención de ventas totales de categoría por día con variables externas

```
select AA.CAT_ID,AA.TEXT_DAY,AA.VENTAS,BB.* FROM
(select CAT_ID,TEXT_DAY,SUM(VALUE) as VENTAS
from (select * from prueba_ventas WHERE CAT_ID="FOODS" limit 10000000) as a
GROUP BY CAT_ID,TEXT_DAY limit 1000) AS AA
LEFT JOIN
(select * from caracteristics_dates) AS BB
ON AA.TEXT_DAY=BB.TEXT_DAY
ORDER BY CAT_ID,DATES LIMIT 100000;
```

Se muestra la consulta final que extrae por categoría el total de ventas y asocia todas las variables externas en un solo conjunto de datos.

Teniendo los datos que necesitamos en una base de datos y habiendo desarrollado el problema en la sección tres, es necesario modularizar el proyecto, el cual tendrá la ventaja de poder ser fácilmente configurable, más escalable, fácil de mantener, mejora el control del proceso, mejora la comprensión del código y facilita la colaboración para el desarrollo ágil de nuevas funcionalidades.

Primero se desarrollará todo lo relacionado al código fuente, en donde el programa principal llamado “main”, el cual tendrá las instrucciones generales que seguirá el proceso representadas a un alto nivel.

Figura 35. Programa principal del proceso de demanda de productos

```
47      log.info('Iniciando el proceso de analisis por categorias')
48      for i in range(0,len(parameters["cat"])):
49          log.info('categoria: '+str(parameters["cat"][i]))
50          cat= parameters["cat"][i]
51          log.info("Iniciando extraccion de informacion")
52          data=ed.extraccion().extrac_info(parameters=parameters,num=i)
53          log.info("Extraccion de informacion de categoria completado")
54          data_v=data.set_index("DATES")[["VENTAS"]].rename({"VENTAS":cat},axis=1)
55          data_cat=data.drop(["CAT_ID","VENTAS"],axis=1)
56          data_cat=data_cat.rename(dict(zip(list(data_cat.columns),parameters["rename_ve"])),axis=1)
57
58          log.info("Extraccion y tratamiento de variables externas")
59          data_cat=tv.pre_var_ex().transformation(parameters,data_cat).set_index("date")
60          log.info("Tratamiento de variables externas completado")
61          data_cat=pd.concat([data_v,data_cat],axis=1)
62
63          log.info("Generando el mejor modelo")
64          mod_ind,sco = tp.training_model().search_best_model(parameters,data_cat,cat)
65          log.info("Mejor modelo generado")
66
67          log.info("Calculando las predicciones del periodo")
68          datos_predf=pdi.predict_f().predict_pass(parameters,data_cat,cat,sco,mod_ind)
69          log.info("Predicciones calculadas")
```

Se muestra la parte más importante del programa principal, en donde se puede observar en que orden se usan los distintos módulos del proceso de demanda de productos.

El programa principal a su vez llamara a los distintos módulos, los cuales contendrán las instrucciones detalladas de cada una de las instrucciones generales que realizara el programa principal, en donde los módulos a desarrollados son:

- extrac_data: En este módulo se genera la conexión a la base de datos, y se realiza la consulta expuesta anteriormente.

Figura 36. Módulo para la extracción de información

```
from sqlalchemy import create_engine
import pandas as pd


class extraccion:
    def __init__(self):
        self.init = "extract"

    def extrac_info(self,parameters,num=1):
        db_connection_str = "mysql+pymysql://{0}:{1}@{2}/demanda_productos".format(parameters["user"],parameters["pass"],parameters["ip"])
        db_connection = create_engine(db_connection_str)
        sql_s="""select AA.CAT_ID,AA.VENTAS,BB.* FROM
        (select CAT_ID,TEXT_DAY,SUM(VALUE) as VENTAS  from (select * from prueba_ventas WHERE CAT_ID=\"""+parameters["cat"][num]+"""\" limit 10000000) as a

        GROUP BY CAT_ID,TEXT_DAY limit 100000) AS AA
        LEFT JOIN
        (select * from caracteristics_dates) AS BB
        ON AA.TEXT_DAY=BB.TEXT_DAY
        ORDER BY CAT_ID,DATES LIMIT 100000;"""


        df = pd.read_sql(sql_s, con=db_connection)
        return df
```

Este módulo está constituido por la conexión a la base de datos y la consulta a realizar.

- trans_var_ext: Este módulo tiene como objetivo procesar las variables externas relacionadas a las fechas, se utilizó el procesamiento de variables externas antes explicado.

Figura 37. Módulo para el procesamiento de variables externas

```
import pandas as pd
from sklearn.preprocessing import MinMaxScaler

class pre_var_ex:

    def __init__(self):
        self.init = "extract"

    def transformation(self,parameters,data_cat):
        cal=data_cat

        dum=pd.get_dummies(cal[["event_type_1","event_type_2"]].replace('',"Normal"),dtype="int")
        "cultural normal religius"
        print(dum.head())
        for i in ["Cultural","Normal","Religious"]:
            dum["event_type_1_"+i]=dum["event_type_1_"+i]+dum["event_type_2_"+i]
            dum=dum.drop("event_type_2_"+i,axis=1)

        scaler = MinMaxScaler()

        dum = pd.DataFrame(scaler.fit_transform(dum),columns=dum.columns)
        cal1=pd.concat([cal.drop(["d","weekday","event_name_1","event_type_1","event_name_2","event_type_2","snap_CA","snap_TX","snap_WI"],axis=1),dum],axis=1)
        scaler = MinMaxScaler()

        dum2=cal1[["wm_yr_wk","wday","month","year"]]
        dum2=pd.DataFrame(scaler.fit_transform(dum2),columns=dum2.columns)

        cal2=pd.concat([cal1.drop(["wm_yr_wk","wday","month","year"],axis=1),dum2,pd.get_dummies(cal["event_name_1"].fillna("Normal1"),dtype="int")],axis=1)
        return cal2
```

En este módulo se utilizó el mismo código para el procesamiento de variables externas, con algunos pequeños cambios para poder tenerlo dentro de una clase.

- train_pass: Es el módulo encargado de llevar a cabo el entrenamiento de los modelos para encontrar el mejor por categoría.

Figura 38. Módulo para la búsqueda del mejor modelo

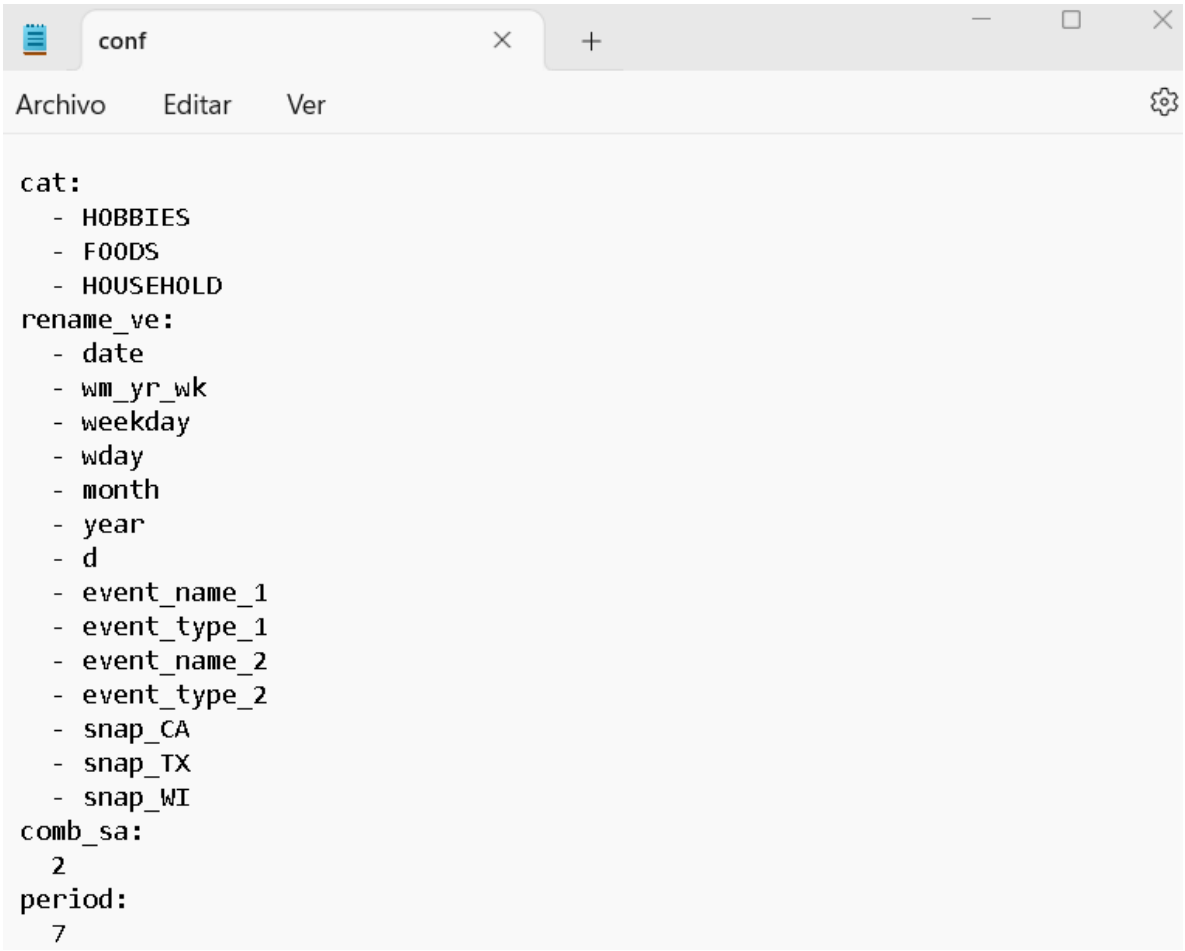
```
def search_best_model(self,parameters,data_cat,cat):

    score_sar,result=self.tra_sar(parameters,data_cat,cat)

    n_test = int(parameters["test_len"])#7
    # model configs

    cfg_list = et.utils_train().model_configs(parameters)#model_configs()
    # grid search
    scoresCNN= self.grid_search(parameters, data_cat.rename({cat:"CAT"},axis=1), cfg_list, n_test, tip_mod=1)
    scoresLSTM= self.grid_search(parameters, data_cat.rename({cat:"CAT"},axis=1), cfg_list, n_test, tip_mod=2)

    #LSTM
    sco=list(scoresLSTM[0][0])
    sco[3]=parameters["train_LSTM"]
    model = et.utils_train().model_fit_lstm(data_cat.rename({cat:"CAT"},axis=1), sco)

    model_LS,train_fx,train_fy=model[0],model[1],model[2]
    predict=model_LS.predict(train_fx)
    datos_predf=pd.DataFrame({"Y":list(train_fy.reshape(len(train_fy))),"Y_pred":list(predict.reshape(len(predict)))})
    score_LSTM=rmse(datos_predf["Y"],datos_predf["Y_pred"])

    #CNN
    sco=list(scoresCNN[0][0])
    sco[3]=parameters["train_CNN"]
    model = et.utils_train().model_fit(data_cat.rename({cat:"CAT"},axis=1), sco)
    model_CN,train_fx,train_fy=model[0],model[1],model[2]
    predict=model_CN.predict(train_fx)
    datos_predf=pd.DataFrame({"Y":list(train_fy.reshape(len(train_fy))),"Y_pred":list(predict.reshape(len(predict)))})
    score_CNN=rmse(datos_predf["Y"],datos_predf["Y_pred"])
    tot_scores=[score_sar,score_LSTM,score_CNN]
    mod_ind=tot_scores.index(min(tot_scores))

    print(tot_scores)
    if mod_ind==0:
        result.save(".\\modelos\\SARIMAX_"+cat+".pkl")
    elif mod_ind==1:
        model_LS.save(".\\modelos\\LSTM_"+cat+".h5")
    else:
        model_CN.save(".\\modelos\\CNN_"+cat+".h5")

    return mod_ind
```

En este módulo la función principal entrena el mejor de cada uno de los tipo de modelos y escoge el mejor con respecto a la métrica de error cuadrático medio.

- extra_train: Se compone de las funciones auxiliares para el tratamiento de datos antes de entrenar los modelos de aprendizaje profundo, se realizó con las funciones explicadas para el entrenamiento de redes neuronales.
- predict: Genera la predicción del modelo que haya tenido el mejor desempeño.
- extra_log: Tiene las indicaciones necesarias para generar el archivo de log en un formato especifico.

Teniendo desarrollada la parte del código fuente, es momento de generar la carpeta que contendrá un archivo con las configuraciones necesarias para que funcione el programa en formato YAML, entre las que están, las categorías a las cuales se les aplicara el proceso, renombramiento de algunas variables, el conjunto de hiperparámetros entre los cuales se buscaran los mejores para los modelos,etc. De esta manera se podrán realizar cambios de manera sencilla sin tener que modificar directamente el código fuente.

Figura 39. Archivo de configuración del proceso de demanda de productos

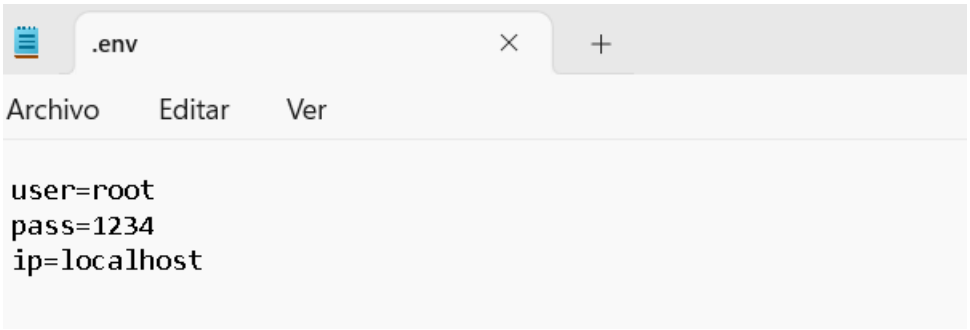


En la anterior figura se muestra cómo se deben de agregar los diferentes parámetros en el archivo de conf.

Adicionalmente es necesario guardar de manera correcta las credenciales con las cuales se podrá conectar a una API, base de datos, etc. Por ello de manera local se creara un archivo llamado ".env" el cual contendrá las mismas, ya que en caso de utilizar un sistema de control de versiones como GitHub será necesario añadir ".env" en los archivos que se omitan al

momento de subir tu código en este sistema, con el objetivo de mantener seguras las credenciales que contienen información sensible.

Figura 40. Variables de entorno



En la anterior figura se muestra un ejemplo del tipo de credenciales que se pueden utilizar en un proyecto.

Es importante mencionar que en caso de que se necesiten otro tipo de variables externas que modifiquen el funcionamiento de nuestro programa se utilizara la biblioteca de python llamada argparse, un ejemplo de uso es el poder correr el programa en un ambiente de desarrollo, preproductivo o productivo (hace referencia a los distintos tipos de credenciales que se usan en el proceso) agregando el argumento "--env", también se pueden agregar la localización de recursos importantes del proyecto.

Figura 41. Comando para iniciar el proceso de predicción de ventas

```
C:\\Users\\Escritorio\\deep_enviroment2\\Scripts\\python.exe "src
\\main.py" --project_directory "C:\\Users\\Escritorio\\Trabajo profesional
\\Revisión demanda de productos\\Modularizacion\\demanda_productos" --
path_config "C:\\Users\\Escritorio\\Trabajo profesional\\Revisión demanda de
productos\\Modularizacion\\demanda_productos\\config" --path_log "C:\\Users
\\Escritorio\\Trabajo profesional\\Revisión demanda de productos
\\Modularizacion\\demanda_productos\\log\\log.log" --env "dev"
```

Lo anterior muestra los posibles parámetros que podrían tener el comando para iniciar el proceso, donde uno de los más importantes sería el tipo de ambiente representado por "--env".

De igual manera se agregará una carpeta de "logs", el cual tiene como objetivo darnos un vistazo del correcto funcionamiento del proceso, en caso de que exista un problema durante el proceso se podrá ver de manera sencilla la causa del mismo, de esta manera se podrá actuar de manera eficiente para la solución de los posibles problemas, adicionalmente se puede

incluir información relevante del proceso para una posterior revisión del comportamiento en producción.

Figura 42. Archivo de logs del proceso de predicción de ventas

En la anterior figura se puede observar el tipo de mensajes que se pueden guardar en los archivos de log, para que de esta manera se pueda llevar un seguimiento del funcionamiento del proceso.

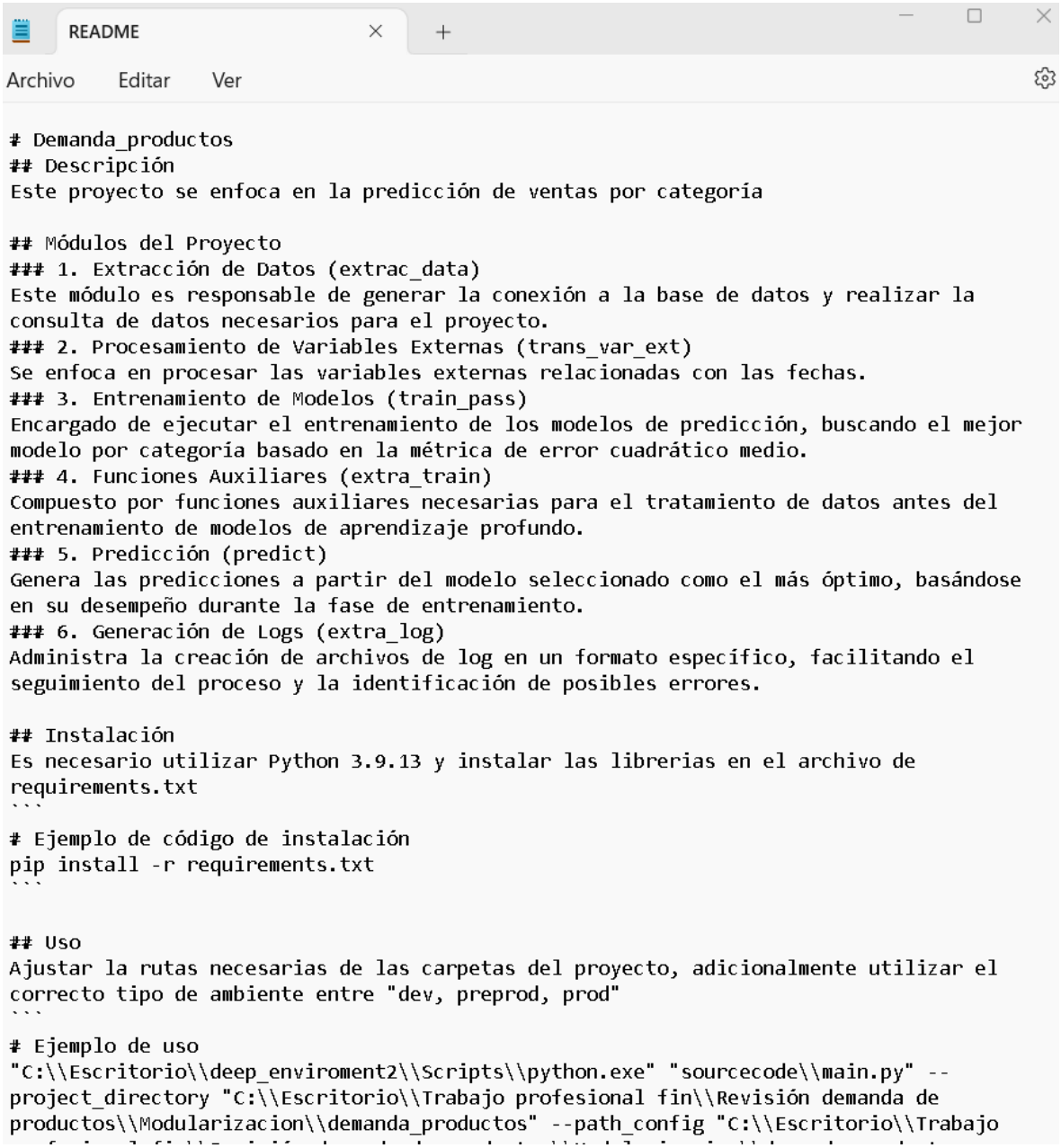
Por otra parte, es importante generar un archivo llamado “requirements” el cual contendrá todas las dependencias necesarias y suficientes que necesita el proyecto para que pueda funcionar y de esta manera poder replicar tus resultado en cualquier ambiente que se configure de manera correcta.

Figura 43. Requirements del proyecto de demanda de productos

En el anterior archivo se puede observar la estructura que tiene un archivo de tipo “requirements”.

Por último, es necesario generar un archivo “README.md”, el cual contendrá toda la información necesaria relacionada con el proyecto para que cualquier persona que necesite entender el proyecto pueda hacer referencia a este archivo.

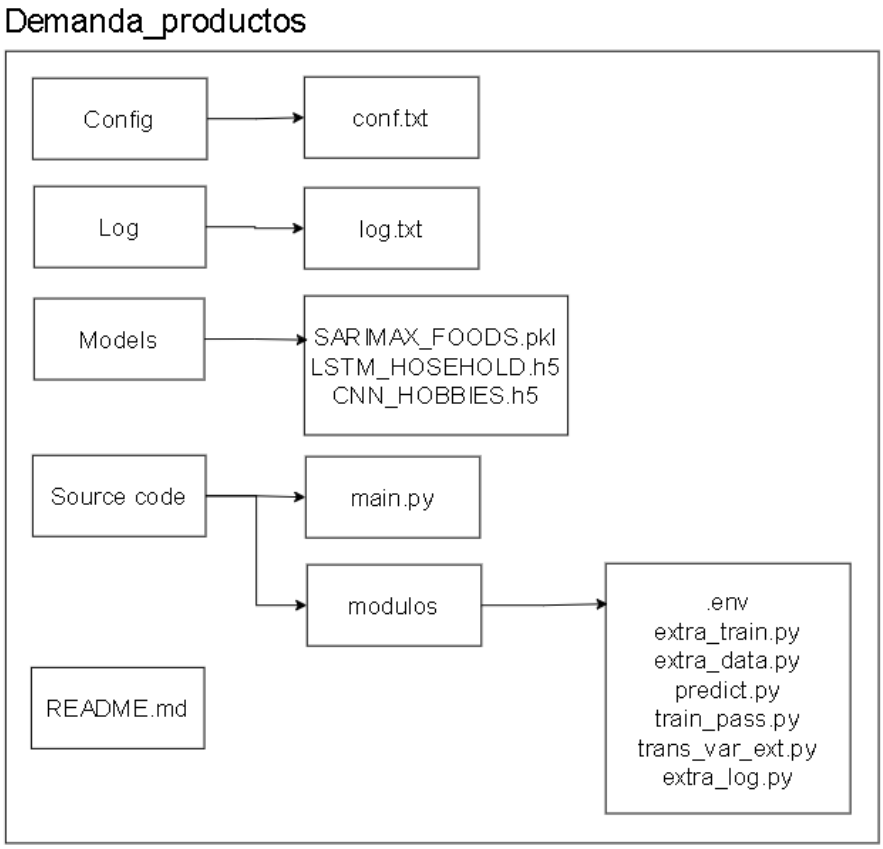
Figura 44. README del proyecto de demanda de productos



En los archivos de tipo “README” se tiene que agregar todo lo que necesite saber las personas que harán uso de tu código entre lo que esta la descripción del proyecto, el código que contiene, los requerimientos del proyecto, como usarlo, etc.

Teniendo en cuenta la explicación anterior, la distribución de los archivos en el proceso de “Demanda_productos” tiene la siguiente forma.

Figura 45. Estructura de archivos en el proceso de modularización



En el anterior diagrama se puede observar tanto los tipos de archivos que contienen este tipo de proyectos y la forma en que están distribuidos los mismos.

Una vez que el proceso termina de generar los valores actuales y las predicciones correspondientes, las cuales se deberán de guardar en algún lugar que pueda gestionar estos datos, en este caso utilizaremos Google drive para guardar la salida del proceso, con el propósito poder utilizar los datos para la generación de un tablero que cumpla el objetivo establecido.

Figura 46. Estructura de archivos en el proceso de modularización

Se puede observar una pequeña visualización del tipo de reportes que se podrían crear con los resultados de un proceso de este tipo, lo que al final tiene que cumplir con el objetivo propuesto en el problema a resolver.

Referencias

- Artley, B. (2022, 26 de abril). Time Series Forecasting with ARIMA, SARIMA and SARIMAX. Towards Data Science. Recuperado de https://towardsdatascience.com/time-series-forecasting-with-arima-sarima-and-sarimax-ee61099e78f6.
- Boysen, J. (2017). Global Food Prices. Kaggle. Recuperado de https://www.kaggle.com/datasets/jboysen/global-food-prices/data.
- Brownlee, J. (2020, 28 de agosto). How to Grid Search Deep Learning Models for Time Series Forecasting. Machine Learning Mastery. Recuperado de https://machinelearningmastery.com/how-to-grid-search-deep-learning-models-for-time-series-forecasting/.
- Devore, J. L., Berk, K. N., & Carlton, M. A. (2021). Modern mathematical statistics with applications (3.ª ed.). Springer.
- Ester, M., Kriegel, H.-P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. En E. Simoudis, J. Han, & U. M. Fayyad (Eds.), Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD-96) (pp. 226-231). AAAI Press. ISBN 1-57735-004-9.
- Filzinger, T. (2023, 19 de marzo). LSTM Long Short-Term Memory - Memoria a corto plazo de larga duración. Konfuzio. Recuperado de https://konfuzio.com/es/lstm/.
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. MIT Press.
- Google Cloud. (2024). Descripción general de BigQuery. Recuperado de https://cloud.google.com/bigquery/docs/introduction.
- Grubbs's test. (s.f.). En Wikipedia, Recuperado de https://en.wikipedia.org/wiki/Grubbs%27s_test.
- Howard, A., Makridakis, S., & Vangelis. (2020). M5 Forecasting - Accuracy. Kaggle. https://kaggle.com/competitions/m5-forecasting-accuracy.

- Kanani, A. [Ahmad Kanani]. (2023, 4 de junio). 16.5. How Fast is BigQuery? Looker Studio Masterclass - Full Beginner to Advanced Course [Video]. YouTube. https://www.youtube.com/watch?v=NqTd0fXsROk.
- Karpathy, A. (2015, 21 de mayo). The unreasonable effectiveness of recurrent neural networks. Andrej Karpathy blog. Recuperado de https://karpathy.github.io/2015/05/21/rnn-effectiveness/.
- Liu, F. T., Ting, K. M., & Zhou, Z.-H. (2008). Isolation-based anomaly detection. ACM Transactions on Knowledge Discovery from Data, 6(3), 1-39. https://doi.org/10.1145/2133360.2133363.
- Mandal, M. (2021, mayo). Introduction to Convolutional Neural Networks (CNN). Analytics Vidhya. Recuperado de https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/.
- Michailidis, M. (2018). How to Win a Data Science Competition: Learn from Top Kagglers. National Research University Higher School of Economics. Coursera.
- Moore, D. S., McCabe, G. P., & Craig, B. A. (2014). Introduction to the practice of statistics (8th ed.). W. H. Freeman and Company.
- Olah, C. (2015, 27 de agosto). Understanding LSTM Networks. Colah's blog. Recuperado de http://colah.github.io/posts/2015-08-Understanding-LSTMs/.
- Reddy, V. K. (2020). PM Dataset. Kaggle. Recuperado de https://www.kaggle.com/datasets/vamshikreddy/pm-dataset.
- Romero, L. (2024). Códigos trabajo profesional (v3.0).Zenodo. https://doi.org/10.5281/zenodo.10822847.
- Sefidian, A. M. (2019, 24 de febrero). Understanding 1D, 2D, and 3D convolutional layers in deep neural networks. Recuperado de http://www.sefidian.com/2019/02/24/understanding-1d-2d-and-3d-convolutional-layers-in-deep-neural-networks/.
- Solis Moreno, I. (2020, 13 de julio). First steps to define a Big Data and Analytics Architecture. IBM Community. https://community.ibm.com/community/user/ai-

datascience/blogs/ismael-solis-moreno1/2020/07/13/first-steps-to-define-a-big-data-and-analytics-arc.

- Statsmodels. (s.f.). statsmodels.tsa.statespace.sarimax.SARIMAX. Statsmodels. Recuperado de https://www.statsmodels.org/dev/generated/statsmodels.tsa.statespace.sarimax.SARIMAX.html.
- TensorFlow. (2023, 27 de octubre). Time series forecasting. Tensor Flow Core. Recuperado de https://www.tensorflow.org/tutorials/structured_data/time_series.
- Venkatesan, R., & Li, B. (2017). Convolutional neural networks in visual computing: A concise guide. CRC Press.
- Winsorizing. (s.f.). En Wikipedia, Recuperado de https://en.wikipedia.org/wiki/Winsorizing.

Índice de figuras

Índice de cuadros