Maestría en

CIENCIA DE DATOS Y MÁQUINAS DE APRENDIZAJE CON MENCIÓN EN INTELIGENCIA ARTIFICIAL

Trabajo previo a la obtención de título de Magister en Ciencia de datos y Máquinas de Aprendizaje con Mención en Inteligencia Artificial

AUTOR/ES:

Ricardo Alfredo Borja Quinteros

Erick Paul Changoluisa Simbaña

Danny Paul Cali Gusqui

Darwin Francisco Tapia Venzo

TUTOR/ES:

Msc. Alejandro Cortés

Msc. Iván Reyes Chacón

TEMA:

Implementación de modelos de aprendizaje automático para pronóstico de ventas de vehículos en ecuador

Quito – Ecuador

Julio - 2025

REINVENTEMOS EL FUTURO

Certificación de autoría

Nosotros, **Danny Paul Cali Gusqui, Erick Paul Changoluisa Simbaña, Darwin Francisco Tapia Venzo, Ricardo Alfredo Borja Quinteros,** declaramos bajo juramento que el trabajo aquí descrito es de nuestra autoría; que no ha sido presentado anteriormente para ningún grado o calificación profesional y que se ha consultado la bibliografía detallada.

Cedemos nuestros derechos de propiedad intelectual a la Universidad Internacional del Ecuador (UIDE), para que sea publicado y divulgado en internet, según lo establecido en la Ley de Propiedad Intelectual, su reglamento y demás disposiciones legales.

Firma del graduando

Danny Paul Cali Gusqui

Firma del graduando

Erick Paul Changoluisa Simbaña

Firma del graduando

Darwin Francisco Tapia Venzo

Firma del graduando

Ricardo Alfredo Borja Quinteros

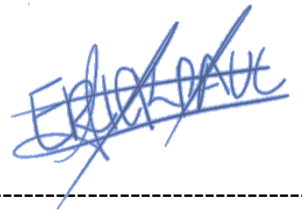
Autorización de Derechos de Propiedad Intelectual

Nosotros, **Danny Paul Cali Gusqui, Erick Paul Changoluisa Simbaña, Darwin Francisco Tapia Venzo, Ricardo Alfredo Borja Quinteros**, en calidad de autores del trabajo de investigación titulado I***mplementación de modelos de aprendizaje automático para pronóstico de ventas de vehículos en ecuador***, autorizamos a la Universidad Internacional del Ecuador (UIDE) para hacer uso de todos los contenidos que nos pertenecen o de parte de los que contiene esta obra, con fines estrictamente académicos o de investigación. Los derechos que como autores nos corresponden, lo establecido en los artículos 5, 6, 8, 19 y demás pertinentes de la Ley de Propiedad Intelectual y su Reglamento en Ecuador.

D. M. Quito, (julio 2025)

Firma del graduando

Danny Paul Cali Gusqui



Firma del graduando

Erick Paul Changoluisa Simbaña

Firma del graduando

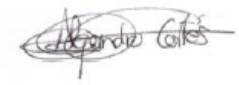
Darwin Francisco Tapia Venzo

Firma del graduando

Ricardo Alfredo Borja Quinteros

Aprobación de dirección y coordinación del programa

Nosotros, **Msc. Alejandro Cortés y Msc. Iván Reyes,** declaramos que los graduandos: **Danny Paul Cali Gusqui, Erick Paul Changoluisa Simbaña, Darwin Francisco Tapia Venzo, Ricardo Alfredo Borja Quinteros**, son los autores exclusivos de la presente investigación y que ésta es original, auténtica y personal de ellos.



Msc. Alejandro Cortés

Director/a de la

Maestría en Ciencia de Datos y Maquinas de Aprendizaje con Mención en Inteligencia Artificial

---

Msc. Iván Reyes

Coordinador/a de la

Maestría en Ciencia de Datos y Maquinas de Aprendizaje con Mención en Inteligencia Artificial

DEDICATORIA

Dedico este trabajo a mis padres. A mi madre, que en paz descanse, y que desde la gloria de Dios me ha brindado su bendición. A mi padre, por su apoyo incondicional y por estar presente en cada etapa de mi vida. También extiendo un agradecimiento especial a la Universidad Internacional del Ecuador, por el conocimiento impartido, las clases interactivas y la excelencia de su cuerpo docente. **Darwin Tapia.**

Este trabajo lo dedico a mi familia. A mis padres, que en todo momento me han sabido apoyar en todo este trayecto académico y profesional. A mis hermanos, que me dan ánimo y motivos para ser una mejor persona y profesional. A mis sobrinas, que con sus ocurrencias y cariño que me brindan me motivan a ser un pilar y apoyo en su vida. Y, por último, a mi pareja que me ha acompañado en todo mi trayecto de vida, que me ha sabido brindar su apoyo y pude encontrar un lugar de paz cuando me encuentro en caos. **Erick Changoluisa**

Este trabajo se lo dedico a la memoria de mis abuelos maternos, cuya guía espiritual y bendición me han acompañado a lo largo de este proceso. A mi madre, por su admirable esfuerzo, por haberme sostenido y alentado aún en soledad, y por creer siempre en mi capacidad de alcanzar lo más alto. A mi pareja, con quien comparto no solo este logro académico, sino también un camino de crecimiento conjunto. Y a mis hermanos y amigos, por su constante apoyo y afecto en cada etapa, tanto en los momentos de alegría como en los desafíos más duros. **Danny Cali**

Dedico este trabajo a mis padres, a mis hermanos, sobrinos, familiares y amigos que han sabido darme ánimo en tiempos difíciles, la motivación y el deseo de superación lo he heredado de mis padres; he sabido cumplir la promesa que le hice a mi abuela materna que en paz descanse de llegar a ser un profesional, pienso que estoy cumpliendo con ser un buen hijo, hermano, tío y un buen ser humano que busca superarse día a día. **Ricardo Borja**

AGRADECIMIENTOS

Quiero expresar mi profundo agradecimiento a mis padres. A mi madre, que en paz descanse, por su guía espiritual y por inculcarme el valor del trabajo honesto y perseverante. A mi padre, por su constante apoyo y por acompañarme en cada paso de mi vida. Asimismo, agradezco a la Universidad Internacional del Ecuador por brindarme una formación de calidad, con clases dinámicas y un cuerpo docente de alto nivel, que han sido fundamentales en mi desarrollo académico y profesional. **Darwin Tapia.**

Quiero agradecer a mi familia, por su apoyo constante, su motivación y consejos que me han sabido brindar para ser una mejor persona y un gran profesional. A Dios y a la Virgen de Guadalupe que me acompañan en cada paso y logro que consigo. A mi pareja por ser un apoyo incondicional en toda esta trayectoria de la maestría. A su vez, agradezco a la Universidad Internacional del Ecuador por la calidad de estudios que brinda en cada una de sus clases con docentes de alto nivel que ha logrado brindar una enseñanza de primera. **Erick Changoluisa**

Agradezco profundamente a Dios por su guía constante y por permitirme llegar hasta aquí con fe y determinación. A mi madre y hermanos, por su amor, comprensión y por convertirse en mi fuerza diaria; saber que soy un modelo a seguir para ellos me llena de responsabilidad y gratitud. A mi pareja, por su apoyo incondicional, por acompañarme en cada etapa de este proceso académico, y por creer siempre en mi potencial. **Danny Cali**

Me permito agradecerle en primer lugar a Dios, a mi familia, amigos y a cada una de las personas que conforman mi vida, ya que todo y todos tenemos un propósito, mismo que me ha permitido llegar hasta aquí, superando obstáculos de todo tipo, impensados, de la nada, sin embargo, nada ocurre por casualidad sino por causalidad. **Ricardo Borja**

RESUMEN

El presente estudio tiene como objetivo principal desarrollar un sistema de pronóstico de ventas de vehículos nuevos en Ecuador, utilizando técnicas avanzadas de análisis de series temporales, a partir de un dataset proporcionado por el Servicio de Rentas Internas (SRI) correspondiente al año 2024, para abordar el pronóstico se implementaron modelos estadísticos y modelos de machine learning como Prophet, SARIMAX y Holt-Winters, con la finalidad de analizar tendencias, estacionalidades, comportamientos cíclicos de ventas.

Todas estas herramientas proporcionan una perspectiva estratégica para distribuidores, concesionarios y reguladores del sector automotriz, ya que permite obtener mayor precisión en predicciones de ventas de vehículos en el mercado.

Este trabajo implica el análisis exploratorio de datos, el tratamiento de series temporales no estacionarias, comparación de métricas de error y visualización de resultados.

Los resultados muestran que estos modelos son aplicables en el mundo real, de manera favorable en cuanto al tema de precisión y usabilidad en el campo práctico.

Los hallazgos del estudio contribuyen para la toma de decisiones en ámbitos como la planificación y las técnicas de análisis de datos en el mercado de venta de vehículos ecuatorianos.

Palabras Claves: predicción de ventas de vehículos, análisis de datos de ventas, industria automotriz, Prophet, SARIMAX, modelos estadísticos.

ABSTRACT

The main objective of this study is to develop a system for forecasting new vehicle sales in Ecuador, using advanced time series analysis techniques, based on a dataset provided by the Internal Revenue Service (SRI) for the year 2024. To address the forecasting, statistical models and machine learning models such as Prophet, SARIMAX, and Holt-Winters were implemented, with the aim of analyzing trends, seasonality, and cyclical sales behaviors.

All these tools provide a strategic perspective for distributors, dealerships, and regulators in the automotive sector, as they allow for greater accuracy in vehicle sales predictions in the market. This work involves exploratory data analysis, the treatment of non-stationary time series, comparison of error metrics, and visualization of results.

The results show that these models are applicable in the real world, favorably regarding precision and usability in the practical field.

The findings of the study contribute to decision-making in areas such as planning and data analysis techniques in the Ecuadorian vehicle sales market.

Keywords: vehicle sales forecasting, sales data analysis, automotive industry, Prophet, SARIMAX, statistical models.

LISTA DE FIGURAS (Índice de figuras)

LISTA DE FIGURAS (Índice de figuras)

CAPÍTULO 1:

1. INTRODUCCIÓN

En este apartado se ofrece una explicación detallada del proyecto desarrollado para analizar la predicción del comportamiento en la venta de autos nuevos en Ecuador utilizando modelos avanzados de aprendizaje supervisados orientados a series temporales. El alcance del trabajo está precisamente definido, junto con los objetivos generales y específicos que son esenciales para desarrollar pronósticos sólidos, claros y fiables. Asimismo, se presenta una justificación que respalda el desarrollo de este proyecto, teniendo en cuenta la importancia de contar con herramientas analíticas de alto nivel que contribuyan a la planificación estratégica, la optimización del uso de recursos mejorando la competitividad en el sector automotriz, que está en constante evolución en Ecuador.

1.1. Definición del proyecto

El proyecto explora como usar técnicas de análisis de series temporales mediante la aplicación de modelos avanzados de aprendizaje supervisado para predecir como se comportarán las ventas de vehículos nuevos en Ecuador. Para este análisis, se emplea un conjunto de datos del Servicio de Impuestos Internos (SRI), con información histórica sobre las ventas de automóviles en el país durante 2024, procesada mendiante herramientas científicas de datos y modelado estadístico, incluyendo los siguientes modelos:

- Facebook Prophet.
- Sarimax
- Holt-Winters

El proyecto también compara el desempeño de estos modelos usando métricas de error y precisión, para identificar cual ofrece los mejores pronósticos, apoyando así las decisiones

estratégicas en el sector automotriz. Además, incorpora visualizaciones interactivas que facilitan el análisis de los resultados haciéndolos accesibles tanto para los usuarios técnicos y no técnicos.

1.2. Justificación e importancia del trabajo de investigación

El pronóstico de la demanda de vehículos es clave para planificar estratégicamente en la industria automotriz de Ecuador, en un contexto donde las decisiones suelen basarse en criterios subjetivos o datos incompletos la falta de herramientas de análisis representa un gran obstáculo para tomar decisiones efectivas y fundamentadas este desafío es común en el sector automotriz dado el mercado más complejo y la necesidad de optimizar procesos esenciales como la producción la distribución y el marketing.

Este proyecto propone abordar esta problemática mediante el desarrollo de un método integral, reproducible y fundamentado científicamente para evaluar la demanda futura de automóviles. El enfoque incorpora técnicas avanzadas de aprendizaje supervisado y no supervisado, para el análisis de datos complementadas con el modelado estadístico más robusto. Este planteamiento permite generar los pronósticos sean precisos y confiables, mediante clasificación eficiente y óptima de los datos. Además, para facilitar la interpretación y la comunicación de los resultados se utilizan interfaces interactivas diseñadas con tecnologías innovadoras adaptadas al entorno actual.

1.3. Alcance

El estudio actual se centra en el análisis y los pronósticos de ventas de automóviles, para un óptimo desarrollo modelos confiables las principales etapas del trabajo son:

- Limpieza y transformación de un conjunto de datos históricos sobre la venta de automóviles cubiertos por el proceso de purificación, formato y transformación.

- El uso de modelos de aprendizaje supervisado y no supervisado, donde se introducen varios métodos para predicción y clasificación.
- En la evaluación de rendimiento esperada, los modelos se comparan utilizando técnicas ampliamente reconocidas para determinar cuantitativamente la precisión de los pronósticos.
- La visualización de resultados, se aprecian en las herramientas gráficas avanzadas, las cuales, se utilizan para representar visualmente las series temporales, tendencias futuras y patrones ocultos identificados por los modelos predictivos.

1.4. Objetivos

1.4.1. Objetivo general

Desarrollar un modelo de pronóstico y clasificación confiable utilizando métodos avanzados de aprendizaje supervisado y no supervisado para predecir la venta de vehículos nuevos, descubrir patrones ocultos dentro de los datos y proporcionar asistencia analítica en toma de decisiones estratégicas y comerciales en el campo automotriz en el Ecuador.

1.4.2. Objetivos específicos

- Utilizar la metodología KDD para creación de un modelo de predicción y clasificación que sea reutilizable y adaptable a la estructura de la información.
- Realizar la limpieza, procesamiento y modificación de datos de ventas históricas para garantizar su calidad, coherencia y adaptación al análisis de series temporales.
- Implementar modelos de aprendizaje supervisado de predicción basadas en series temporales, garantizando la capacidad de pronóstico y la durabilidad de los modelos.
- Integrar modelos de aprendizaje no supervisado para clasificar y analizar relaciones, respaldando la obtención de estructuras, patrones y dependencias ocultas dentro de los

datos

- Evaluar la precisión esperada para cada modelo diseñado, utilizando mediciones estadísticas apropiadas.
- Realizar un análisis comparativo del rendimiento de los modelos implementados para determinar su precisión y estabilidad.
- Generar visualizaciones claras e interpretativas que comuniquen efectivamente los resultados logrados, para facilitar la comprensión del comportamiento de las ventas y respaldo de las decisiones estratégicas.

CAPÍTULO 2:

2. REVISIÓN DE LITERATURA

2.1. Estado del Arte

El pronóstico de ventas ha sido objeto de estudio en múltiples industrias debido a su importancia para planificación estratégica en el sector automotriz, anticipando la demanda futura de vehículos mediante la producción, importaciones de materiales, ensamblaje y campaña de marketing para su venta (Ikome et al., 2022).

A nivel global, la combinación del uso de modelos estadísticos ARIMA con técnicas de aprendizaje automático como redes neuronales y modelos híbridos han contribuido significativamente a mejorar la precisión de las predicciones de series temporales. Zhang et al. (1998) propusieron un enfoque híbrido, lo cual abrió un nuevo campo de estudio del comportamiento no lineal de predicciones.

En los últimos años, el modelo Prophet desarrollado por Facebook ha cobrado relevancia y protagonismo por su capacidad para gestionar el manejo automático de estacionalidades y detección de cambios estructurales (Menculini et al., 2021). Su aplicación se ha extendido a diversas áreas como finanzas, epidemiología, logística y automotriz considerando su fácil y rápido uso (Ikome et al., 2022).

En Ecuador, la aplicación de modelos avanzados como el Prophet en el análisis de series temporales en el sector automotriz aún es limitada, y muchas decisiones comerciales se basan en estimaciones o datos agregados de años anteriores. Este trabajo busca aportar una propuesta práctica, escalable que integra Prophet con otros métodos basados en aprendizaje automáticos como XGBoost, para mejorar precisión del análisis y fundamentación científica (Venkatesh et al., 2025)

2.2. Marco Teórico

En el desarrollo de este proyecto se han aplicado diferentes técnicas, metodologías y conceptos aprendidos a lo largo del programa de maestría, que permiten sustentar la implementación de modelos de predicción basado en series temporales relacionados con el lenguaje de programación Python, sus estructuras internas y los componentes esenciales para el desarrollo del modelo propuesto.

2.2.1. Python

Python es un lenguaje de programación de alto nivel, multiparadigma y dinámicamente tipado, ampliamente adaptado al análisis de datos mediante su sintaxis sencilla, legible, e intuitiva haciendo que su disponibilidad sea versátil, por lo que, se ha convertido en el lenguaje global a nivel estadístico en entornos académicos e industriales (Kabir et al., 2024).

Por otro lado, otros lenguajes de programación utilizan una estructura similar como Python con datos básicos: enteros, flotantes, strings, y booleanos. Las variables en Python son tan flexibles que pueden ser consideradas como cajas para almacenar cualquier cosa sin la necesidad de etiquetarlas con tipos específicos en entornos de análisis intensivos de datos (Hongjie et al., 2021) .

2.2.2. Elementos Predefinidos

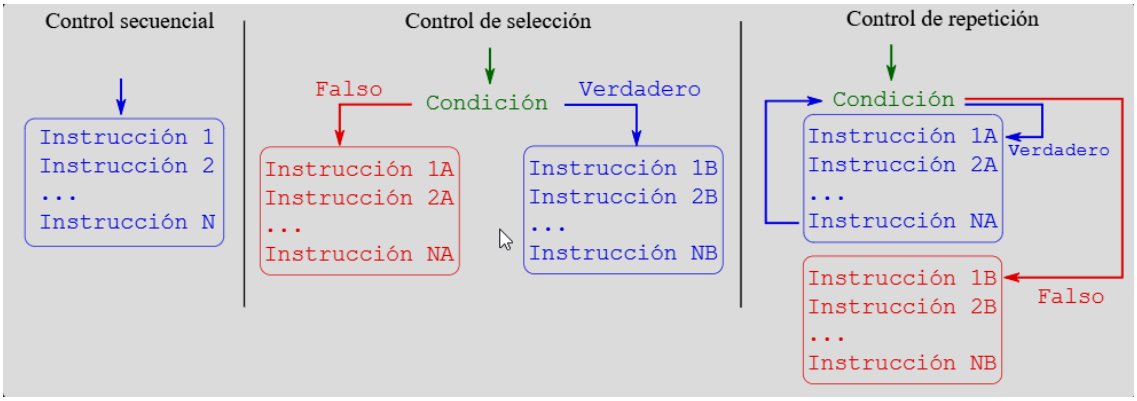
Python cuenta con elementos o tipos predefinidos propios del lenguaje que facilitan el trabajo como son las listas, y su variante las tuplas. Las litas se construyen con corchetes y se parando por comas, mientas que las tuplas tienen una estructura similar que, al ser creadas, ya no pueden ser alteradas, convirtiéndose en una herramienta para manejar datos de forma constante o fijas mientras se realicé la programación.

2.2.3. Estructuras de Control y Funciones

Las estructuras de control permiten dirigir el flujo o el orden de programación estableciendo condiciones y repeticiones de comportamiento esperados de los datos véase en Figura 1. Python incluye como estructuras de control a las condicionales lógicas (if, elif, else) que permiten procesar bloques de código según determinadas condiciones y los bucles (for, while) admiten repetir instrucciones mientras cumpla una condición específica (Buriticá & Guerrero, 2021).

Figura 1.

Diagrama de flujo de una estructura de control



Nota: Se evidencia las tres estructuras de control en programación. Fuente: Saltos.

A diferencia de otros lenguajes, Python delimita los bloques de códigos mediante indentación en lugar de llaves, con palabras como begin, o end. Esta característica, se debe alinear correctamente a estructuras de control o al definir funciones debido a que cuando se desajusta genera errores en su ejecución (Chimarro-Amaguaña et al., 2023)

Python permite experimentar con sus elementos básicos y combinarlos para crear funciones personalizadas que ayudan a simplificar y mejorar los algoritmos. Para definir una función se utiliza la palabra *def* seguida del nombre que se le quiere aplicar y entre paréntesis los

argumentos que se solicitarán, Estas funciones, se escribe su nombre seguido de paréntesis, y si se requiere de argumentos se los debe proporcionar en los paréntesis (Saabith et al., 2021)

2.2.4. Jupyter Notebooks

Los archivos de Python se guardan en formato .py y para ejecutarlo se aplica el comando *Python nombre.py;* sin embargo, la ejecución de código en entornos interactivos en el servidor de Jupyter incluyendo en la nube como Google Colab, facilita la interacción y planificación del análisis mediante celdas que combinan el texto y código, por lo que resulta más fácil su manipulación y ejecución de manera individua (Konstantin et al., 2022).

2.2.5. Librerías de Python

En programación una librería es un conjunto de funcionalidades e implementaciones que permiten codificar un lenguaje, con el objetivo de crear una interfaz independiente y llevar a cabo tareas que antes no se podían completar.

Python destaca por su versatilidad, funcionalidad, es interactivo, multiplataforma y de código abierto; por lo que, ha permitido a los usuarios crear librerías adicionales a las propias del lenguaje. Las librerías pueden ser instaladas con el módulo *pip* y el método. main*()* y de acuerdo con los objetivos existen diferentes clasificaciones en donde pueden ser usadas como son: Deep Learning, machine learning, cálculo numérico, visualización, IA, procesamiento de lenguaje, entre otros. Python ofrece un bagaje de librerías que pueden ser utilizadas para ampliar las capacidades.

- **Matplotlib:** es una librería estándar de Python que ayuda a crear visualizaciones ofreciendo la generación de gráficos estáticos, interactivos y animados, siendo una herramienta versátil (Hassan et al., 2021).

- **Numpy:** permite estructurar, manipular y analizar los datos, con sus vectores y matrices multidimensionales, a partir de operaciones matemáticas (Kadhim et al., 2022)
- **Pandas:** muy utilizada en Data Science, permite estructurar los datos en series o Data Frame, debido a que facilita el manejo de datos estructurados (Sundaram et al., 2023).
- **Seaborn:** su interfaz interactiva y de alto nivel permite al usuario una visualización avanzada de datos estadísticos para entender el data, esto genera visualizaciones conmenos código y mejor análisis
- **TensorFlow:** desarrollada por Google para el cálculo numérico, su estructura permite crear redes neuronales y sus diagramas de flujo de datos son muy utilizadas en Deep Learning.
- **Scikit-Learn:** diseñado para la construcción de modelos de aprendizaje automático o machine learning.
- **Keras:** diseñado para el desarrollo de modelos de aprendizaje profundo, con el objeto de crear prototipos de redes neuronales de manera rápida y eficaz.
- **Streamlit:** permite crear aplicaciones o interfaces web interactivas de manera rápida y sencilla.
- **Prophet:** modelo de pronóstico de series temporales desarrollado por Meta (anteriormente Facebook).

2.2.6. Aprendizaje Supervisado

El aprendizaje automático es un aspecto de la Inteligencia artificial que busca desarrollar algoritmos capaces de aprender de los datos, hacer predicciones y tomar decisiones en base la información preprocesada, para mejorar el desempeño de procesos.Un programa aprende de una

experiencia E respecto a una tarea T y una medida de rendimiento P, si su rendimiento en la tarea T, medida por P, mejora con la experiencia E (Kampezidou et al., 2023).

El aprendizaje supervisado es un elemento de machine learning en donde el modelo se entrena usando un conjunto de datos, en donde cada entrada tiene su respuesta predefinida, este proceso tiene como objetivo predecir la salida correcta de datos representativos y etiquetados para nuevas entradas que nunca ha visto antes, enfatizando la capacidad de aprender relaciones complejas entre variables de forma precisa y con calidad.

2.2.7. Aprendizaje No Supervisado

El aprendizaje no supervisado es un elemento de machine learning que se enfoca en descubrir patrones, estructuras o relaciones ocultas dentro de los datos no etiquetados, a diferencia con el aprendizaje supervisado que se basa en que el modelo aprenda de respuestas predefinidas, trabajando únicamente con los datos de entrada, para entender su estructura y su significado (Naeem et al., 2023). Por otro lado, los algoritmos no supervisados, permiten encontrar regularidades en los datos y son especialmente útiles cuando no es posible o práctico etiquetar grandes volúmenes de información.

Este aprendizaje comprende técnicas que permiten agrupar, reducir o explorar conjuntos de datos sin intervención directa de un humano, el objetivo en el aprendizaje no supervisado no es predecir una variable conocida, sino detectar patrones ocultos que faciliten la interpretación, transformación y aplicación de los datos.

2.2.8. Metodología KDD

La metodología KDD (Knowledge Discovery in Databases) representa el enfoque para extraer información de grandes volúmenes de información y darles un significado interpretativo, en el contexto actual de Machine Learning, incluye fases estructuradas que van desde la

selección de los datos hasta la heneración de modelos predictivos.El término KDD fue normalizado por Fayyad, Piatetsky-Shapiro y Smyth (1996), los cuales propusieron casos prácticos para la automatización, minería de datos y resolución de problemas complejos en el mundo real. De igual forma, sus resultados aportan significativamente al estudio ya que son datos confiables, útiles y comprensibles.

En proyectos de machine learning, la minería de datos se concreta a través de algoritmos supervisados o no supervisados. Sin embargo, la efectividad del modelo depende directamente de las fases previas del proceso KDD. Muchos errores atribuibles al modelo provienen realmente de malas decisiones en las etapas de preprocesamiento o selección de atributos que sirven para evitar el bajo desempeño de los modelos.

El proceso de KDD comprende del siguiente proceso:

Selección de datos relevantes. Consiste en identificar, extraer y reunir los datos que serán útiles para el análisis. No todos los datos disponibles son necesarios, por lo que es fundamental elegir aquellos que estén alineados con el objetivo del análisis.

Preprocesamiento y limpieza. Una vez seleccionados los datos, se procede a su limpieza. En esta fase se corrigen errores, se manejan valores faltantes, se eliminan duplicados y se detectan posibles inconsistencias.

Transformación y reducción. Se transforman los datos para hacerlos adecuados al análisis. Esto puede incluir normalización, codificación de variables categóricas, o reducción de dimensiones para simplificar los datos sin perder información relevante.

Minería de datos (data mining). Se aplican técnicas como clasificación, regresión, agrupamiento (clustering) o reglas de asociación, con el fin de encontrar patrones, relaciones o comportamientos en los datos.

Interpretación y evaluación. Los patrones descubiertos deben ser evaluados para determinar si realmente tienen valor, si son comprensibles y si pueden ser utilizados en la toma de decisiones.

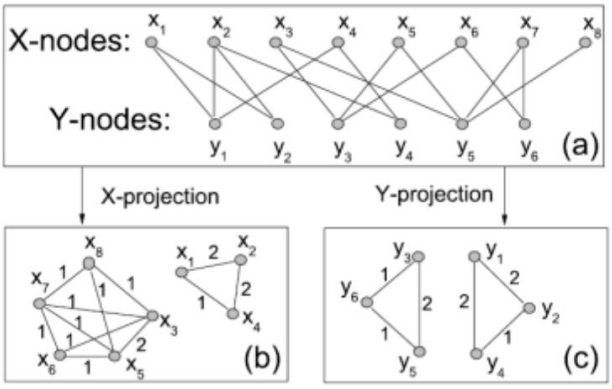
2.2.9. Redes bipartitas

Estas redes poseen nodos que a su vez pueden dividirse en 2 conjuntos de datos aislados y que solo existen conexiones entre nodos de conjuntos diferentes, pero no dentro del mismo conjunto. Estos tipos de grafos o redes que comúnmente son usadas para modelar interacciones, recomendaciones, clasificación y detección de patrones que alimentan modelos predictivos como los de clasificación binaria o multiclase

Un ejemplo simple sería una red de equipos de futbol y jugadores, estos están en conjuntos separados, sin embargo, una arista o conexión podría darse entre estos conjuntos debido a que son conjuntos que pertenecen a una entidad llamada Fútbol, ver figura 2.

Figura 2.

Red bipartita con nodos de tipo X e Y (a) y su proyección X (b) y proyección Y (c)



Nota: Ejemplo de Bipartita. Fuente: Zhou et al. (2007)

2.2.10. Series Temporales

Una serie temporal es un conjunto de observaciones tomadas en distintos puntos del tiempo, usualmente a intervalos regulares. Su análisis permite comprender patrones como:

- **Tendencia:** dirección general de la serie a largo plazo.
- **Estacionalidad:** fluctuaciones periódicas o predecibles.
- **Ciclos:** variaciones que son influenciadas por la economía.
- **Aleatoriedad:** variabilidad no explicada por los modelos.

El análisis y pronóstico de series temporales se apoya en técnicas estadísticas que permiten modelar y prever el comportamiento futuro de estas secuencias de datos.

2.2.11. Análisis de estacionariedad

Al analizar series temporales, la estacionariedad se refiere a la estabilidad de sus propiedades estadísticas una serie en el tiempo; dado esto, una serie temporal estacionaria es aquella que no muestra cambios como la media, la varianza y la autocorrelación al avanzar el tiempo, el modelado y previsión.

La estacionariedad y su análisis se vuelven más significativos en escenarios donde los valores futuros se basan sobre los valores pasados, mediante análisis de los modelos autorregresivos. En el campo de la economía, la estacionariedad es fundamental para construir y estimar modelos de pronóstico económico confiables, ya que es clave identificar tendencias sostenibles y decisiones de inversión acertadas (Ryan et al., 2025).

2.2.12. Pruebas ADF y KPSS para el análisis de estacionariedad

Si una serie temporal llega a cambiar su comportamiento de manera abrupta o gradual con el tiempo, tanto el modelado como la predicción se complican. Para seleccionar el modelo

correcto se utiliza la prueba estadística ADF (Augmented Dickey-Fuller), la cual evalúa la existencia de una raíz unitaria que a una serie temporal la vuelve en un valor estable.

Por otra parte,la prueba ADF básicamente ofrece una perspectiva complementaria o con el objetivo comprobar la existencia de una raíz unitaria en una serie temporal, debido a que la prueba sigue una distribución *t* asimétrica en donde el valor *t* calculado se compara con los valores críticos simulados y proporcionados (Bawdekar et al., 2022). Es así, como la hipótesis nula se rechaza para valores pequeños, y se acepta la hipótesis alternativa concluyendo que la serie temporal es estacionaria.

Otra de las pruebas para determinar la estacionariedad de una serie temporal, es la prueba KPSS (Kwiatkowski Phillips Schmidt y Shin), esta prueba es contraria al ADF ya que la hipótesis nula es que la serie temporal es estacionaria, y se basa en una regresión lineal que divide la serie temporal en tendencia determinista, recorrido aleatorio y error estacionario. La prueba KPSS indica si el valor *p* es mayor que el nivel de significancia, no se rechaza la hipótesis nula y por ende se concluye que la serie temporal es estacionaria.

2.2.13. Modelos de Pronóstico

Prophet. Modela series temporales de manera aditiva, por ejemplo:

$$y(t) = g(t) + s(t) + h(t) + \varepsilon t \quad (1)$$

- $g(t)$: crecimiento (lineal o logístico)
- $s(t)$: estacionalidad
- $h(t)$: feriados y eventos
- εt: error aleatorio

Su ventaja principal es la automatización del ajuste y la capacidad de incorporar múltiples estacionalidades.

SARIMAX/ARIMA. El modelo ARIMA (AutoRegressive Integrated Moving Average) es una técnica clásica para el modelado de series estacionarias. Cuando se incorpora estacionalidad, se convierte en SARIMA; y al incluir variables externas, se convierte en SARIMAX.

Holt-Winters. Es un modelo de suavizamiento exponencial o también denominado triple exponencial, fue propuesto por Winters en 1960, es útil para datos con tendencia y estacionalidad. Su versión aditiva se usa cuando la estacionalidad es constante y la multiplicativa varía proporcionalmente al nivel de la serie.

2.2.14. Métricas de Evaluación de Modelos

El rendimiento de los modelos de predicción se evaluará mediante las siguientes métricas:

MAE (Error Absoluto Medio).

$$\boldsymbol{MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|} \quad (1)$$

RMSE (Raíz del Error Cuadrático Medio).

$$\boldsymbol{RMSE = \sqrt{\frac{1}{n}(y_i - \hat{y}_i)^2}} \quad (2)$$

MAPE (Error Porcentual Absoluto Medio).

$$\boldsymbol{MAPE = \frac{100}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right|} \quad (3)$$

Estas métricas permiten determinar la precisión de los modelos y elegir el más adecuado según el contexto.

2.2.15. Clustering, usos y ventajas

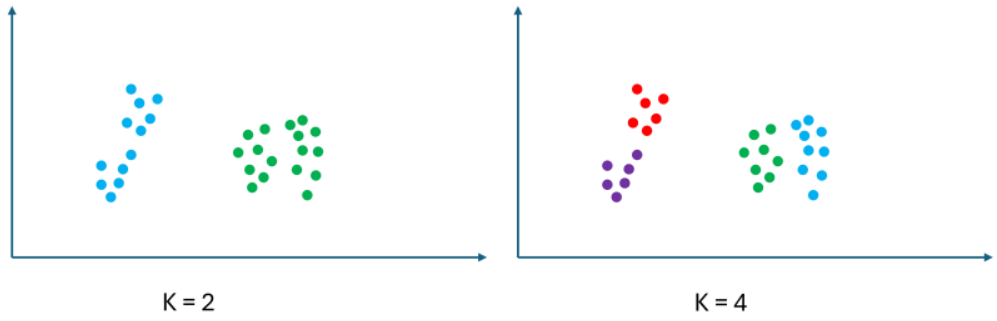
Clustering o Agrupamiento es una técnica de análisis de datos que consiste en agrupar objetos con características similares, con un enfoque para resumir la información, descubrir patrones y detectar outliers, que mejoran la obtención de resultados a partir de una toma de decisiones que promueve la segmentación del mercado con una mejor comprensión de los datos.

Este proyecto a aplicado el clustering para clasificar las marcas de vehículos y encontrar las semejanzas entre el avalúo y año de compra, lo que ayuda a encontrar dos categorías para los algoritmos esto son: los basados en distancia y los basados en densidad. Entre los algoritmos de clustering más populares tenemos:

K-means. Es el algoritmo basado en distancia más utilizado para la ciencia de datos gracias a su rapidez y eficacia en la gestión de grandes volúmenes de datos. Este algoritmo se basa en centroides y para ello buscamos un número fijo de clústeres k que representa el número de centroides que se quiere localizar, ver figura 3. Para encontrar el valor óptimo de k y obtener los mejores resultados podemos recurrir a métodos como del codo o validación cruzada.

Figura 3.

Clusterización K-means

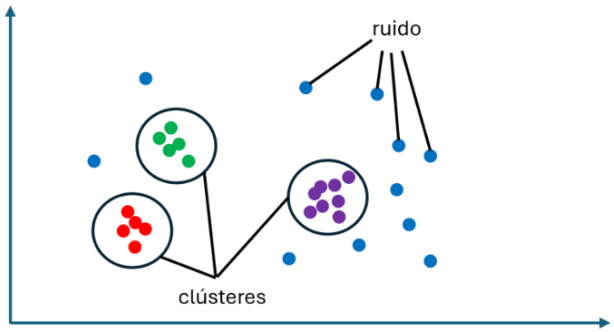


Nota: Representación del algoritmo K-meas. Lado izquierdo datos sin clasificar, lado derecho datos agrupados en base de clústeres. Fuente: Suyal & Sharma (2024)

DBSCAN. Este algoritmo se basa en la densidad de los datos, y es muy útil para identificar valores atípicos y clústeres de baja densidad en un mismo conjunto de datos. A diferencia del K-means, no se necesita especificar el número de clústeres k que se generarán.

Figura 4.

Clusterización DBSCAN



Nota: Ejemplo de agrupamiento con DBSCAN.

CAPÍTULO 3:

3. DESARROLLO

El desarrollo estructurado y eficiente de presente proyecto se fundamenta en la metodología KDD (Knowledge Discovery in Databases), la cual, destacada por su enfoque en la construcción de modelos y la posibilidad de reutilizar o adaptar soluciones a nuestros datos. Su flexibilidad metodológica la convierte en una herramienta clave para aplicar procesos de análisis y predicción, resultando especialmente adecuada para interpretar y entender grandes volúmenes de información.

El conjunto de datos seleccionado para el estudio la predicción de ventas se denomina *SRI_Vehiculos_Nuevos_2024.csv* (SRI-Servicio de Rentas Internas, 2024) el cual fue obtenido del Servicio de Rentas Internas (SRI) de Ecuador y corresponde al año 2024.

Además, para establecer el entorno de desarrollo en Colab, es fundamental instalar diversas librerías que apoyan en la recolección, análisis, modelado y visualización de datos del diseño de interfaces interactivas, mediante el uso del comando "pip install streamlit". Esto, se integra en la librería streamlit, luego se incorpora prophet usando el comando "pip install prophet". La librería networkx se instala non el comando "pip install networkx", también, se añade plotly utilizando el comando "pip install plotly", a través del comando "pip install statsmodels" y finalmente se incorpora ipywidgets mediante "pip install ipywidgets".

Una vez que se ha completado la instalación librería, el próximo paso es la incorporación de bibliotecas. Esta etapa permite obtener acceso a las características específicas que cada librería proporciona, que va desde la gestión de datos hasta la representación y el modelado.

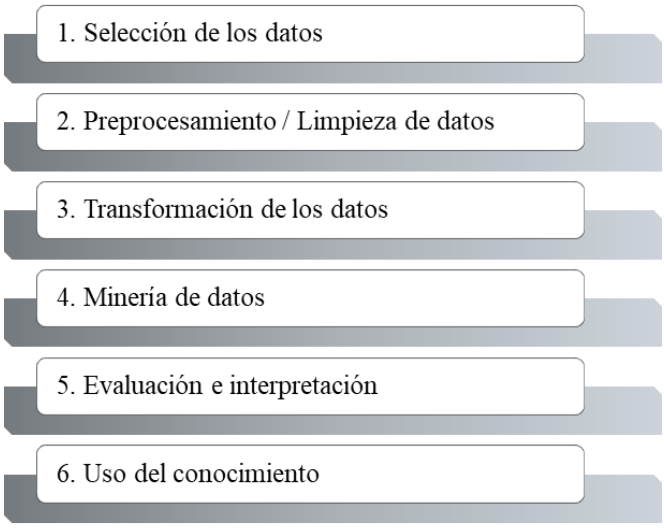
Para iniciar, se incorporan pandas y numpy, dos plataformas esenciales en el ámbito de la ciencia de datos para la limpieza y transformación, con respecto a la representación de datos se emplean tres librerías importantes: matplotlib, seaborn y plotly.

Para la modelación de series temporales y el análisis estadístico, se incorporan bibliotecas prophet y statsmodels, con el propósito de medir el desempeño de los modelos y se integran funciones de la biblioteca sklearn. De tal manera, se incorpora KMeans que permite la segmentación de datos a través de agrupamientos. Por último, se incluye networkx, una librería dedicada al análisis de grafos y redes complejas.

Este proyecto, estructurado según la metodología KDD, se compone de seis fases esenciales que se presentan y explican de manera detallada y secuencial, ver figura 5.

Figura 5.

Metodología KDD



3.1. Selección de datos

El presente estudio se fundamenta en el análisis del conjunto de datos titulado SRI_Vehiculos_Nuevos_2024.csv (SRI-Servicio de Rentas Internas, 2024), este conjunto de

datos contiene información detallada sobre la comercialización de vehículos nuevos en el país y representa una fuente valiosa para la comprensión y entendimiento del comportamiento del mercado automotriz y a su vez realizar predicciones sobre la demanda que existirá en un futuro.

Cada fila del dataset representa una venta registrada de un vehículo nuevo, e incorpora variables de carácter tanto técnico como comercial. Las columnas más relevantes para poder obtener un análisis completo son las siguientes:

- Fecha: Fecha de la transacción, en formato mes/año.
- Marca: Marca del vehículo (por ejemplo, Kia, Daytona, Chevrolet, entre otros).
- Modelo: Modelo comercial del vehículo.
- Clase: Tipo de vehículo (automóvil, SUV, moto, entre otros).
- Cilindraje: Capacidad del motor en centímetros cúbicos.
- Combustible: Tipo de energía utilizada (gasolina, diésel, híbrido, eléctrico).
- N° de ventas: Número de unidades vendidas para esa marca, modelo y fecha.

Con el objetivo de desarrollar un modelo, el cual permita pronosticar el número de vehículos nuevos vendidos en los próximos meses.

3.1.1. Carga de datos

Para iniciar el proceso de análisis y modelado, se realizó la carga del archivo de datos provisto por el Servicio de Rentas Internas (SRI), que contiene información sobre la venta de vehículos nuevos en Ecuador. Este archivo, en formato CSV, fue cargado utilizando la librería panda como se aprecia en la figura 6, especificando el tipo de codificación 'latin1' y el separador ';' con el fin de asegurar que los caracteres especiales y la estructura regional del archivo sean interpretados correctamente.

Figura 6.

Cargo de datos en pandas

```
file_path = "/content/SRI_Vehiculos_Nuevos_2024.csv"
df = pd.read_csv(file_path, encoding='latin1', sep=';')
```

3.2. Limpieza de datos

Una vez cargados los datos el equipo realizó una limpieza inicial eliminando aquellas columnas completamente vacías, así como una columna adicional sin nombre ('Unnamed: 20') que no contiene información relevante. Posteriormente, se estandarizan los nombres de las columnas para facilitar el trabajo analítico: Se optó por eliminar espacios innecesarios y se convirtieron todos los nombres a mayúsculas, lo que permite un acceso más sencillo y consistente.

En cuanto a la fecha de compra, se implementa una detección automática para localizar la columna correspondiente, ya que esta puede tener diferentes nombres en distintas versiones del archivo. Una vez identificada, la columna fue convertida al tipo de dato *datetime*, asegurando el reconocimiento correcto del formato día/mes y manejando posibles errores de conversión. De igual manera, se transformó la columna de "avalúo", que representa el valor económico del vehículo, eliminando caracteres no numéricos como comas y convirtiendo su contenido a formato numérico para su posterior análisis.

Como parte del proceso de depuración, se eliminan los registros duplicados, así como aquellos que no contaban con datos esenciales como la fecha de compra o el valor del avalúo, asegurando que el conjunto de datos contenga únicamente registros completos y únicos.

3.3. Transformación de los datos

Tras haber generado las variables temporales MES_AÑO y AÑO, se introducen filtros dinámicos para analizar únicamente los registros correspondientes a los años 2022, 2023 y 2024. Esta selección permite enfocar el análisis en los datos más recientes y relevantes para la elaboración de pronósticos y toma de decisiones actualizadas. Aunque también se incluye la opción de filtrar por marcas específicas (como TOYOTA o CHEVROLET), esta funcionalidad se deja comentada para ser utilizada según la necesidad del análisis.

Una vez aplicado el filtro por año, se agrupan los datos por el período mensual (MES_AÑO) y se contabiliza el número de vehículos vendidos por mes. Este conteo representa la serie temporal de ventas mensuales, que es copiada en una nueva variable (ventas_ts) para ser convertida a un índice de fecha con el fin de facilitar su visualización cronológica.

3.4. Minería de datos: Aprendizaje Supervisado

En el apartado de Minería de datos se realiza un análisis exhaustivo de las ventas mensuales de vehículos en Ecuador, utilizando técnicas de aprendizaje supervisado aplicadas a series temporales. Se parte de la evaluación de la estacionariedad mediante las pruebas ADF y KPSS, lo que permite determinar si se requieren transformaciones previas al modelado. Posteriormente, se lleva a cabo una descomposición estacional para identificar patrones subyacentes como tendencia, estacionalidad y ruido. Finalmente, se aplican tres métodos de pronóstico SARIMAX, Prophet y ETS para proyectar las ventas futuras. Cada técnica se adapta a distintas características de la serie, lo que permite evaluar su rendimiento bajo distintos supuestos.

3.4.1. Análisis de estacionariedad con ADF

Se aplica la prueba ADF sobre la serie “ventas_ts”, que representa las ventas mensuales de vehículos filtradas por años de interés. El resultado incluye varios componentes clave:

- **ADF Statistic**: es el valor del estadístico de prueba.
- **p-value**: representa el nivel de significancia.
- **Valores críticos**: son los umbrales a diferentes niveles de confianza (1%, 5%, 10%) para comparar con el estadístico ADF.

La salida de este análisis permite tomar decisiones importantes sobre la necesidad de diferenciar la serie o aplicar alguna transformación adicional para lograr estacionariedad antes de ajustar un modelo de predicción temporal.

3.4.2. Test de KPSS como complemento

Se evalúa la serie “ventas_ts” con regression='c', lo que indica que se prueba la estacionariedad alrededor de una constante (media constante). La salida muestra:

- **KPSS Statistic**: valor del estadístico de prueba.
- **p-value**: nivel de significancia asociado.
- **Valores críticos**: umbrales a diferentes niveles de confianza (1%, 5%, 10%) para interpretar el estadístico.

Esta prueba ayuda a decidir si se requiere transformar la serie antes del modelado.

3.4.3. Descomposición estacional

Antes de realizar esta descomposición, se establecieron condiciones para asegurar que haya una cantidad suficiente de observaciones. Si la serie contiene al menos 24 observaciones mensuales, se asume una estacionalidad anual con un período de 12 meses. En caso de tener entre 12 y 23 observaciones, se reduce el período a 6 meses, mientras que, si hay menos de 12

datos, se omite la descomposición debido a la insuficiencia de datos para capturar un patrón estacional confiable.

Cuando se cumplen las condiciones, se utiliza la función *seasonal_decompose* del paquete *statsmodels* con un modelo aditivo. Esta función separa la serie de tiempo original en tres componentes fundamentales:

- **Tendencia (Trend)**: indica el comportamiento general a largo plazo de las ventas, como crecimiento o declive.
- **Estacionalidad (Seasonal)**: muestra patrones repetitivos a lo largo de cada período (meses del año, por ejemplo).
- **Residuo (Residual)**: representa las irregularidades o variaciones no explicadas por la tendencia ni la estacionalidad.

El resultado se visualiza en una gráfica que facilita la identificación de patrones regulares o cambios estructurales en la serie.

3.4.4. Comparación por marcas (Top 5)

Primero, se verifica que la columna 'MARCA' esté presente en el conjunto de datos. Luego, agrupa todos los registros por marca para calcular el total de ventas acumuladas de cada una. A partir de este total, se seleccionan las cinco marcas con mayor número de ventas, denominadas “Top 5”, y se crea una lista con sus nombres.

Con esa lista, se filtran los registros del DataFrame original para conservar únicamente los correspondientes a esas cinco marcas. Posteriormente, se agrupan las ventas mensuales por marca, usando las columnas MES_AÑO y MARCA, lo que permite observar cómo han variado las ventas mes a mes para cada una de estas marcas destacadas. Para facilitar la visualización

temporal, se convierte la columna MES_AÑO a formato de fecha (datetime), y se restringe el análisis a los datos desde el año 2023 en adelante.

3.4.5. Comparación por año

Se utiliza el DataFrame previamente filtrado (df_filtro), que contiene únicamente los registros de los años seleccionados (por ejemplo, 2022 a 2024). A partir de este conjunto, se agrupan los datos por mes (MES_AÑO) y por año (AÑO), y se calcula la cantidad total de ventas mensuales para cada año.

Luego, se transforma la columna MES_AÑO al formato *datetime* para asegurar una correcta visualización cronológica. Esto es fundamental para que el eje X del gráfico represente de manera precisa la línea del tiempo mensual.

3.4.6. Pronóstico con ARIMA

Se define una función para realizar un pronóstico de ventas utilizando el modelo ARIMA estacional (SARIMAX), una técnica ampliamente utilizada en series temporales cuando se desea modelar y predecir datos que presentan tendencias, estacionalidad y ruido aleatorio.

La función *predecir_con_arima* recibe dos argumentos:

- “serie_ts”: la serie temporal ya preparada, que contiene los datos históricos de ventas mensuales de vehículos.
- “meses_a_predecir”: la cantidad de meses hacia el futuro que se desea pronosticar.

En su interior, se construye un modelo SARIMAX (Seasonal AutoRegressive Integrated Moving Average with eXogenous regressors), que extiende ARIMA para manejar estacionalidad. En este caso, el modelo tiene los siguientes parámetros:

- “order = (1,1,1)”: que representa el orden del componente no estacional (ARIMA):

- “seasonal_order = (1,1,1,12)”: que define el comportamiento estacional, asumiendo un ciclo anual (12 meses).

El modelo se ajusta (fit) sin mostrar mensajes de salida (disp=False), y una vez entrenado, se genera el pronóstico para el número de meses solicitado.

3.4.7. Pronóstico con Prophet

Se define una función para realizar un pronóstico de ventas utilizando el modelo Prophet, desarrollado por Meta (Facebook), el cual es especialmente útil para datos de series temporales con tendencias no lineales y estacionalidad clara, y que también puede manejar bien los valores atípicos y los huecos en los datos.

La función *predecir_con_prophet* toma dos argumentos:

- “serie_ts”: la serie temporal histórica de ventas mensuales (con índice de fechas y valores de ventas).
- “meses_a_predecir”: el número de meses que se desea proyectar hacia el futuro.

Dentro de la función, primero se transforma el DataFrame en el formato requerido por Prophet, que exige dos columnas:

- "ds": las fechas.
- "y": los valores observados (ventas en este caso).

Luego, se asegura que los datos estén en una frecuencia mensual regular (asfreq('MS')), comenzando al inicio de cada mes, lo cual es fundamental para que el modelo entienda correctamente la periodicidad de la serie. Posteriormente, se crea un objeto Prophet, se entrena (fit) con los datos históricos, y se genera un DataFrame extendido hacia el futuro (make_future_dataframe) con la cantidad de meses a predecir.

El modelo devuelve una tabla de predicción con múltiples columnas, entre ellas "yhat" que contiene los valores pronosticados. La función extrae los últimos meses_a_predecir de esta columna, que corresponden a las predicciones futuras, y los devuelve como resultado final.

3.4.8. Pronóstico con ETS

Esta función implementa un modelo de pronóstico utilizando ETS (Error-Trend-Seasonality), también conocido como Suavizamiento Exponencial, una técnica clásica pero muy eficaz para modelar series temporales con tendencia y/o estacionalidad, dependiendo de la estructura de los datos disponibles.

La función *predecir_con_ets* recibe como entrada:

- “serie_ts”: la serie temporal con las ventas históricas mensuales.
- “meses_a_predecir”: el número de pasos (meses) que se desea pronosticar.

Dentro de la función, primero se evalúa si la serie tiene al menos 24 observaciones (lo que generalmente representa dos años de datos mensuales). Si es así, se asume que puede capturarse estacionalidad anual y se configura el modelo ETS con:

- tren = 'add': incorpora una tendencia aditiva a la serie.
- seasonal = 'add': incorpora una estacionalidad aditiva (sólo si hay suficientes datos).
- seasonal_periods = 12: indica que los ciclos estacionales se repiten cada 12 meses.

Si no hay suficientes datos (menos de 24 observaciones), se utiliza el mismo modelo, pero sin estacionalidad, para evitar sobreajuste o comportamiento errático.

El modelo es ajustado con .fit(), y luego se genera el pronóstico a futuro mediante .forecast(), devolviendo la serie con los valores estimados para los meses especificados.

Este modelo es especialmente útil por su simplicidad, velocidad de cálculo y buenos resultados en contextos donde los patrones de ventas siguen una evolución suave, como puede suceder en sectores con comportamiento cíclico y predecible como el automotriz.

3.5. Evaluación e interpretación: Aprendizaje Supervisado

Para la evaluación e interpretación se genera una función, diseñada para facilitar la ejecución de modelos de pronóstico ARIMA, Prophet y ETS con una sola instrucción. La función permite seleccionar el modelo deseado y especificar cuántos meses se desean predecir. Posteriormente, se realiza una evaluación comparativa utilizando los últimos seis meses de datos reales como referencia. Para medir la precisión de cada modelo, se aplican métricas como MAE y RMSE. Luego, se genera una visualización gráfica que resume el desempeño de los modelos, facilitando la interpretación de los resultados obtenidos.

3.5.1. Predicción de ventas

Esta función *predecir_ventas* actúa como una interfaz unificada para ejecutar diferentes modelos de pronóstico, facilitando la selección y ejecución del modelo deseado con una sola línea de código.

La función recibe tres parámetros:

- “serie_ts”: la serie temporal histórica de ventas (por ejemplo, ventas mensuales de vehículos).
- “modelo”: una cadena de texto que indica qué modelo utilizar ('arima', 'prophet' o 'ets').
- “meses_a_predecir”: el número de meses hacia el futuro que se desean predecir (por defecto 6).

Dependiendo del modelo especificado, la función llama a la función correspondiente:

- predecir_con_arima()

- predecir_con_prophet()
- predecir_con_ets()

Cada una de estas funciones ha sido definida previamente y aplica el modelo seleccionado para generar el pronóstico. En caso de que el usuario introduzca un nombre de modelo no reconocido, se lanza un error indicando que debe usar uno de los nombres válidos.

3.5.2. Ejecución de predicciones

El bloque de código que se aprecia en la figura 7, ejecuta el proceso de predicción de ventas futuras utilizando tres modelos distintos (ARIMA, Prophet y ETS) y reserva los últimos seis meses de datos reales para realizar una evaluación comparativa del desempeño de cada modelo.

Figura 7.

Código para la predicción de ventas futuras

```
# Ejecución de predicciones
forecast_arima = predecir_ventas(ventas_ts[:-6], modelo='arima', meses_a_predecir=6)
forecast_prophet = predecir_ventas(ventas_ts[:-6], modelo='prophet', meses_a_predecir=6)
forecast_ets = predecir_ventas(ventas_ts[:-6], modelo='ets', meses_a_predecir=6)
test_real = ventas_ts[-6:]
```

Proceso para realizar este código:

Separación de datos de prueba. Se utiliza el conjunto de datos completo (ventas_ts), pero se excluyen los últimos 6 meses, que serán utilizados como conjunto de prueba (test_real).

Generación de pronósticos con distintos modelos.

- *forecast_arima* utiliza el modelo ARIMA entrenado con los datos hasta seis meses antes del final.
- *forecast_prophet* hace lo mismo usando Prophet.
- *forecast_ets* emplea el modelo de Suavizamiento Exponencial (ETS).

Cada uno de estos modelos genera un pronóstico de 6 pasos (meses) hacia el futuro.

Almacenamiento de datos reales. En *test_real* se almacenan los valores reales de ventas correspondientes a los últimos 6 meses del dataset, que se usarán como referencia para evaluar la precisión de los modelos al comparar estos valores reales contra los pronósticos generados.

3.5.3. Métricas de Evaluación

En este apartado se calcula e imprime las métricas de evaluación de precisión de los modelos de pronóstico aplicados previamente (ARIMA, Prophet y ETS), utilizando los datos reales de los últimos seis meses como referencia.

Se emplean dos métricas comunes en evaluación de series temporales:

MAE (Mean Absolute Error). Es útil para interpretar fácilmente el promedio de desviación en las unidades originales (ventas de vehículos).

RMSE (Root Mean Squared Error). Es una métrica sensible a valores atípicos y útil para modelos donde se busca minimizar errores grandes.

A continuación, se aprecia la forma de aplicar cada modelo:

- Se compara su pronóstico (forecast_arima, forecast_prophet, forecast_ets) con los datos reales (test_real).
- Se calculan el MAE y el RMSE utilizando funciones de sklearn.metrics.

Finalmente, se imprime un resumen comparativo con el desempeño de cada modelo, lo que permite identificar cuál de ellos ofrece menor error y por tanto mayor precisión en el pronóstico.

3.5.4. Visualización de métricas de evaluación

Con el fin de facilitar la interpretación y comparación de los resultados de los modelos de pronóstico implementados (ARIMA, Prophet y ETS), se desarrolló una visualización gráfica de las métricas de evaluación mediante gráficos de barras. Donde el Error Absoluto Medio (MAE) y la Raíz del Error Cuadrático Medio (RMSE) permiten cuantificar la precisión de las predicciones generadas por cada modelo en relación con los valores reales de ventas.

Se organiza la información en un diccionario con los nombres de los modelos y sus respectivas métricas, ver figura 8.

Figura 8.

Diccionario de los modelos y sus métricas

```
metrics = {
    'Modelo': ['ARIMA', 'Prophet', 'ETS'],
    'MAE': [mae_arima, mae_prophet, mae_ets],
    'RMSE': [rmse_arima, rmse_prophet, rmse_ets]
}
```

Este fragmento es fundamental ya que estructura los datos en un formato adecuado para graficar, lo que constituye la base del análisis visual. Luego, el diccionario se convierte en un DataFrame de pandas, llamado *metrics_df*, lo cual permite una manipulación tabular sencilla para graficar con la biblioteca *seaborn*.

La visualización se realiza mediante dos gráficos de barras (uno para MAE y otro para RMSE) Cada gráfico se genera usando *sns.barplot*, especificando el eje X como el nombre del modelo y el eje Y como la métrica correspondiente

3.6. Minería de datos: Aprendizaje no Supervisado

En esta sección se abordan distintos métodos de análisis y visualización aplicados al mercado automotor ecuatoriano. Se explora la correlación entre variables clave como avalúo,

mes y año mediante una matriz interactiva. Se identifican las 10 marcas más vendidas con un gráfico de barras, y se aplican técnicas de *clustering* (KMeans) para segmentar vehículos según su valor y año de compra. Además, se construyen redes bipartitas para analizar las relaciones entre marcas-modelos y marcas-tipos de combustible. Estas redes permiten evaluar la estructura, centralidad, resiliencia y robustez del mercado, proporcionando una visión detallada de su composición y comportamiento.

3.6.1. Matriz de correlación

Se genera una visualización interactiva de la matriz de correlación utilizando Plotly, para analizar la relación entre variables numéricas del conjunto de datos filtrado (df_filtro). Se crea un nuevo DataFrame *variables_correlacion* que contiene tres variables clave, ver figura 9.

Figura 9.

Código de la matriz de correlación.

```
variables_correlacion = df_filtro[['AVALÚO']].copy()
variables_correlacion['MES'] = df_filtro['FECHA COMPRA'].dt.month
variables_correlacion['AÑO'] = df_filtro['FECHA COMPRA'].dt.year
```

Donde:

- AVALÚO: el valor económico del vehículo.
- MES y AÑO: componentes temporales extraídos de la fecha de compra.

Estas variables permiten explorar si, por ejemplo, existe una correlación entre el valor de los vehículos y el mes o año de compra.

Luego se genera la matriz de correlación de Pearson, que mide la relación lineal entre pares de variables. Los valores van de -1 (correlación negativa perfecta) a 1 (correlación positiva perfecta), siendo 0 indicativo de ausencia de correlación. Además**,** se construye un mapa de calor

interactivo con Plotly, usando una escala de color (*Viridis*) que representa visualmente la fuerza de las correlaciones. Cada celda del gráfico muestra el valor exacto de la correlación y permite explorar interactivamente las relaciones entre variables.

3.6.2. Top 10 marcas más vendidas

Un gráfico de barras que visualiza las 10 marcas de vehículos más vendidas en Ecuador es una visualización clave para comprender la composición del mercado automotor, identificar las marcas líderes en ventas y comunicar estos hallazgos de forma clara.

Para ello, primero se verifica que la columna 'MARCA' esté presente en el conjunto de datos filtrado (df_filtro). Luego, se identifican las 10 marcas con mayor número de registros de ventas utilizando. Posteriormente, se genera una visualización horizontal mediante seaborn.barplot, que muestra de forma clara cuántas unidades ha vendido cada marca dentro del top 10.

3.6.3. Clustering con KMeans sobre avaluó y año

Se implementa una técnica de agrupamiento (clustering) utilizando el algoritmo KMeans sobre los datos de ventas de vehículos, específicamente considerando las variables AVALÚO (valor del vehículo) y AÑO (año de compra). Esta técnica permite identificar segmentos de mercado basados en el tipo de vehículo adquirido.

Se filtran los datos para trabajar únicamente con las columnas 'AVALÚO' y 'AÑO', asegurando que no existan valores nulos. Donde, estas dos variables son seleccionadas como dimensiones clave para formar los clústeres, considerando el valor del vehículo y el momento de la compra. Luego el modelo KMeans es sensible a la escala de los datos, por lo que se aplica normalización para que ambas variables tengan el mismo peso en el análisis (media 0, desviación estándar 1).

Se define el modelo KMeans con 3 grupos, que intentará encontrar patrones naturales dentro de los datos, ver figura 10. La elección de *n_clusters = 3* busca representar tres categorías posibles de vehículos según su avalúo: económico, intermedio y lujo.

Figura 10.

Modelo KMeans con 3 grupos

```
kmeans = KMeans(n_clusters=3, random_state=52, n_init=10)
clusters = kmeans.fit_predict(scaled)
```

En la figura 11 se aprecia como se convierte los clústeres numéricos en etiquetas significativas para los usuarios. Se calcula el valor promedio del avalúo en cada grupo, y según ese promedio, se asigna la categoría correspondiente.

Figura 11.

Mapero de clusters a categorías basadas en el valor medio

```
mean_values = cluster_data.groupby('CLUSTER')['AVALÚO'].mean().sort_values() # Asumiendo que df_avaluo_cat está disponible
cluster_to_label = {cluster: label for cluster, label in zip(mean_values.index, ['Económico', 'Intermedio', 'Lujo'])}
cluster_data['CATEGORÍA_AVALUO'] = cluster_data['CLUSTER'].map(cluster_to_label) # Renombrar la nueva columna de categoría
```

Finalmente se genera un gráfico de dispersión interactivo en el que se visualiza la distribución de los vehículos en función del año y el valor económico, coloreados por clúster.

3.6.4. Red Bipartita entre marca y modelo

Para representar la relación entre las marcas de vehículos más vendidas y sus respectivos modelos se implementa una red bipartita. Esta técnica de visualización resulta útil porque permite entender de manera gráfica la estructura de productos.

Primero se identifican las 5 marcas con mayor volumen de ventas para enfoca el análisis en los valores más representativos. Se filtran las combinaciones únicas de marcas y modelos dentro del top 5, ya que esta limpieza asegura que la red contenga únicamente relaciones válidas

y no repetidas. Luego se crea un grafo con *networkx*, en el que se añaden dos tipos de nodos, ver figura 12. Donde se tiene los nodos de marca (bipartite=0) y yodos de modelo (bipartite=1).

Figura 12.

Tipos de nodos

```
B.add_nodes_from(marcas, bipartite=0)
B.add_nodes_from(modelos, bipartite=1)
B.add_edges_from(edges)
```

Posteriormente, se conectan mediante aristas que indican qué modelos están asociados a qué marcas. Esto genera una estructura donde no hay conexiones entre marcas o entre modelos, sólo entre un nodo de marca y uno de modelo.

El grafo se renderiza con un diseño tipo *spring layout*, que distribuye los nodos buscando reducir cruces y mejorar la claridad. Se utilizan colores distintos para identificar marcas (azul cielo) y modelos (verde claro), facilitando la interpretación visual.

Análisis de red Marca-Modelo. Se realiza un análisis estructural de la red bipartita entre marcas y modelos de vehículos, donde se utiliza todo el conjunto de datos disponible (df), lo que proporciona una visión completa de la relación entre fabricantes y modelos. A continuación, se detallan los pasos más importantes para dicho análisis.

Construcción de la red completa. Se construye una red bipartita global a partir de todas las combinaciones únicas de marcas y modelos disponibles en el dataset. Los nodos se dividen en dos conjuntos: marcas y modelos, y las conexiones (aristas) representan relaciones reales entre ellos, ver figura 13.

Figura 13.

Construcción de la red completa:

```
B_marca_modelo = nx.Graph()
marcas = df['MARCA'].dropna().unique()
modelos = df['MODELO'].dropna().unique()
B_marca_modelo.add_nodes_from(marcas, bipartite=0)
B_marca_modelo.add_nodes_from(modelos, bipartite=1)
edges = df[['MARCA', 'MODELO']].dropna().drop_duplicates().values.tolist()
B_marca_modelo.add_edges_from([tuple(edge) for edge in edges])
```

Identificación de nodos por tipo. Se filtran los nodos del grafo para separar los que corresponden a marcas y los que representan modelos. Esto permite analizar su comportamiento por separado.

Cálculo del grado promedio. Para marcas este valor indica cuántos modelos, en promedio, ofrece cada marca registrada en el dataset. Para modelos este valor muestra cuántas marcas, en promedio, están asociadas a un mismo modelo.

Análisis de centralidades en Red MARCA–MODELO. Se realiza un análisis de centralidades en la red bipartita Marca–Modelo, herramienta utilizada en teoría de grafos para identificar los nodos más importantes o influyentes dentro de una red. A continuación, se detallan los pasos que comprenden el análisis de centralidades.

Verificación del grafo adecuado. Se asegura de que el grafo *B_marca_modelo* haya sido creado correctamente y sea una instancia válida de *networkx.Graph*.

Cálculo de centralidades. Se calculan tres tipos de métricas de centralidad para cada nodo en la red, ver figura 14.

Figura 14.

Análisis de centralidades

```
centralidad_df = pd.DataFrame({
  'Nodo': list(B_marca_modelo.nodes),
  'Grado': pd.Series(nx.degree_centrality(B_marca_modelo)),
  'Betweenness': pd.Series(nx.betweenness_centrality(B_marca_modelo)),
  'Closeness': pd.Series(nx.closeness_centrality(B_marca_modelo))
})
```

- *Grado (degree_centrality): M*ide cuántas conexiones directas tiene un nodo.
- *Betweenness (betweenness_centrality):* Mide cuántas veces un nodo actúa como puente en los caminos más cortos entre otros nodos.
- *Closeness (closeness_centrality)*. Mide qué tan cerca está un nodo de todos los demás en la red, considerando las distancias más cortas.

El DataFrame generado se ordena por la centralidad de grado y se muestra el Top 5 de nodos con mayor conectividad directa. Esto permite identificar rápidamente las marcas más diversificadas o los modelos más comunes, dependiendo del tipo de nodo.

Cálculo de resiliencia y robustez para la red Marca-Modelo. Se realiza un análisis de resiliencia y robustez para evaluar la vulnerabilidad o estabilidad del sistema ante la eliminación (intencionada o aleatoria) de nodos clave.

Función de Resiliencia. Se implementa la función "*calcular_resiliencia_robustez"* la cual evalúa qué tan bien se mantiene conectada la red cuando los nodos son eliminados de forma aleatoria. Donde un valor cercano a 1 indica alta resiliencia, mientras que valores cercanos a 0 indican que la red colapsa rápidamente al perder nodos al azar.

Evaluación de Resiliencia para nodos MARCA y MODELO. Usando las listas de nodos tipo MARCA y MODELO previamente definidas, se calcula la resiliencia de cada tipo, ver figura 15.

Figura 15.

Evaluación de Resiliencia para MARCA y MODELO

```
resiliencia_marca = calcular_resiliencia_robustez(B_marca_modelo, marca_nodes_in_graph)
resiliencia_modelo = calcular_resiliencia_robustez(B_marca_modelo, modelo_nodes_in_graph)
```

Función de Robustez. Se implementa la función "*robustez_por_centralidad"* la cual evalúa el efecto de eliminar nodos de forma dirigida, es decir, comenzando por los más importantes según su centralidad de grado. Un valor bajo indica que pocos nodos críticos pueden fragmentar la red, lo que evidencia fragilidad ante ataques dirigidos.

3.6.5. Red bipartita: Marca vs Tipo de Combustible

Se implementa una red bipartita para representa la relación entre las marcas de vehículos y los tipos de combustible que utilizan. Donde su visualización permite identificar qué marcas ofrecen vehículos con distintos tipos de combustibles. A continuación, se muestra cada paso para dicho análisis.

Filtrado y depuración de los datos. Se verifica que existan las columnas necesarias ('MARCA' y 'TIPO COMBUSTIBLE') en el DataFrame y luego se seleccionan únicamente las combinaciones válidas, eliminando duplicados y valores nulos

Creación del grafo bipartito. Se construye un grafo no dirigido con *networkx,* donde los nodos se dividen en dos grupos:

- Nodos tipo 0 (marcas), ver figura 16.

Figura 16.

Nodos tipo 0 (marcas)

```
B_marca_combustible.add_nodes_from(marcas, bipartite=0)
```

- Nodos tipo 1 (tipos de combustible), ver figura 17.

Figura 17.

Nodos tipo 1 (tipos de combustible)

```
B_marca_combustible.add_nodes_from(combustibles, bipartite=1)
```

Luego se crean las aristas, que representan las relaciones reales entre marca y combustible.

Distribución visual de los nodos. Se utiliza un *layout bipartito* que coloca los nodos de cada grupo en una línea distinta, mejorando así la legibilidad del grafo, ver figura 18.

Figura 18.

Distribución visual de los nodos

```
pos = nx.bipartite_layout(B_marca_combustible, marcas)
```

3.7. Evaluación e interpretación: Aprendizaje no supervisado

Para evaluar los métodos de aprendizaje no supervisado se realiza una clasificación automática de vehículos, como se ve continuación.

3.7.1. Clasificación automática por rango de AVALÚO (económico, intermedio, lujo)

Se lleva a cabo una categorización automática de los vehículos en tres niveles de valor económico *Económico, Intermedio y Lujo* utilizando únicamente como criterio el avalúo registrado de cada unidad, para dicha clasificación se realizan una serie de paso, los cuales se ven a continuación.

Preparación del conjunto de datos. Primero verifica que la columna 'AVALÚO' esté presente, esto garantiza que se trabaje únicamente con valores numéricos válidos del avalúo, ver figura 19.

Figura 19.

Verificación de la columna 'AVALÚO'

```
df_avaluo_cat = df_filtro[['AVALÚO']].dropna().copy()
df_avaluo_cat['AVALÚO'] = pd.to_numeric(...)
```

Normalización del valor del avalúo. Se normalizan los valores para que todos estén en la misma escala, ver figura 20. Esto es esencial para que el algoritmo de agrupamiento (KMeans) no se vea sesgado por la magnitud de los valores monetarios.

Figura 20.

Normalizan los valores

```
scaler_av = StandardScaler()
scaled_avaluo = scaler_av.fit_transform(df_avaluo_cat[['AVALÚO']])
```

Agrupamiento automático (KMeans). Se aplica un algoritmo de clustering no supervisado para identificar tres grupos naturales en los datos, ver figura 21. El número de clústeres (n_clusters=3) fue elegido para representar las tres categorías socioeconómicas clásicas del mercado automotor: bajo, medio y alto.

Figura 21.

Clustering no supervisado

```
kmeans_av = KMeans(n_clusters=3, random_state=52, n_init=10)
df_avaluo_cat['CLUSTER'] = kmeans_av.fit_predict(scaled_avaluo)
```

Asignación de etiquetas interpretables. Se calcula el valor promedio de avalúo por grupo y se asignan etiquetas significativas. Este paso convierte clústeres numéricos en categorías comprensibles y usables para el análisis.

Integración con el DataFrame principal. Se añade la clasificación al DataFrame de trabajo. Esto permite usar la nueva categoría para cruzarla con otras variables como tipo de vehículo, marca, año, provincia, etc.

Visualización final. Se genera un histograma de la distribución del avalúo dividido por categoría. Esta visualización es clave para comunicar los resultados del agrupamiento, mostrando claramente cómo se separan los grupos y cuál es su distribución relativa.

3.8. Implementación de conocimiento

Con el fin de facilitar la exploración interactiva de los resultados obtenidos en los modelos de predicción y clasificación, se implementa un dashboard dinámico utilizando Panel y Plotly. Este panel visual permite analizar las ventas de vehículos nuevos en Ecuador entre 2022 y 2024, brindando al usuario la posibilidad de filtrar por marcas y meses, y acceder a métricas clave como ventas totales, número de vehículos y precios promedio.

Además, el dashboard ofrece visualizaciones de tendencias de ventas, distribución por tipo de combustible, clase, país de origen, tipo de vehículo, así como el agrupamiento por avalúo. También integra la evaluación comparativa de los modelos de pronóstico (ARIMA, Prophet y ETS) mediante métricas de error que permiten valorar su precisión. A continuación, se muestran los pasos de mayor impacto para el desarrollo del dashboard.

Configuración de estilo y widgets. Se definen el estilo visual (fondo blanco, tamaño de los selectores) y crean widgets para que el usuario seleccione una o más marcas y rangos de meses.

Botón de actualización y control de estado. Se emplea un botón interactivo que, al hacer clic, activa la generación de gráficos y resultados. Esto evita que los análisis se ejecuten automáticamente, mejorando el rendimiento del dashboard.

Resumen y visualizaciones principales del dashboard. Dentro de la función dashboard, se genera un resumen numérico de ventas, vehículos y precios, gráfico de tendencia de ventas por marca y mes, y una distribución por tipo de combustible, clase de vehículo, país de origen y tipo de vehículo. Además de un Clustering automático por avalúo y año de compra, categorizando los vehículos en económico, intermedio y lujo.

Pronóstico de ventas y comparación de modelos. Se desarrolla una función que utiliza los datos filtrados por el usuario, separando los últimos seis meses como conjunto de prueba. A continuación, se aplican tres modelos de pronóstico ARIMA, Prophet y ETS cuyos resultados se comparan con los datos reales. Para evaluar el desempeño de cada modelo, se calculan las métricas de error MAE y RMSE. Además, los pronósticos generados se visualizan en un gráfico conjunto con líneas diferenciadas, lo que permite una comparación clara y efectiva entre los modelos.

Estructura del dashboard y despliegue final. Se define el diseño final del dashboard como una estructura en columna vertical que organiza los elementos en tres secciones principales: título y controles interactivos, área de resultados, y módulo de pronóstico. Además, el dashboard está configurado para ser funcional, permitiendo su visualización en un navegador local o su exportación como archivo HTML para compartir o integrar en otros entornos.

CAPÍTULO 4:

4. ANÁLISIS DE RESULTADOS

El presente capítulo detalla los resultados obtenidos a partir del procesamiento, análisis y modelado predictivo sobre el conjunto de datos proporcionados por el Servicio de Rentas Internas (SRI) del Ecuador, relativo a las ventas de vehículos nuevos entre los años 2022 y 2024.

4.1. Pruebas de Concepto

Las pruebas de concepto se centraron en el procesamiento de datos, análisis temporal, modelado estadístico, segmentación mediante aprendizaje supervisado y representaciones basadas en grafos.

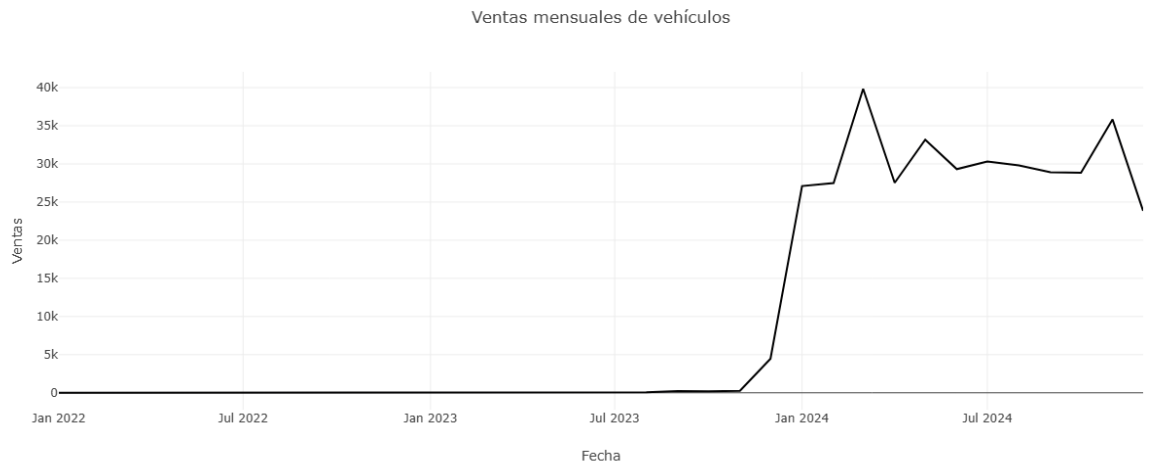
El dataset fue sometido a un proceso de limpieza que incluyó la eliminación de columnas vacías, renombramiento y estandarización de encabezados, conversión del campo “avalúo” a tipo numérico, detección automática de la fecha de compra y eliminación de registros duplicados o con datos esenciales faltantes. Se crea una columna “MES_AÑO” para análisis temporal y una columna “AÑO” para permitir filtros anuales.

4.2. Análisis Exploratorio y Visualización

Posteriormente, se aplicaron filtros dinámicos para los años 2022, 2023 y 2024, centrándose en las marcas más relevantes del mercado. Se genera una serie temporal de ventas mensuales de vehículos, representada con visualizaciones interactivas utilizando la librería Plotly. Las gráficas permiten observar variaciones estacionales marcadas especialmente al cierre de cada año, ver figura 22.

Figura 22.

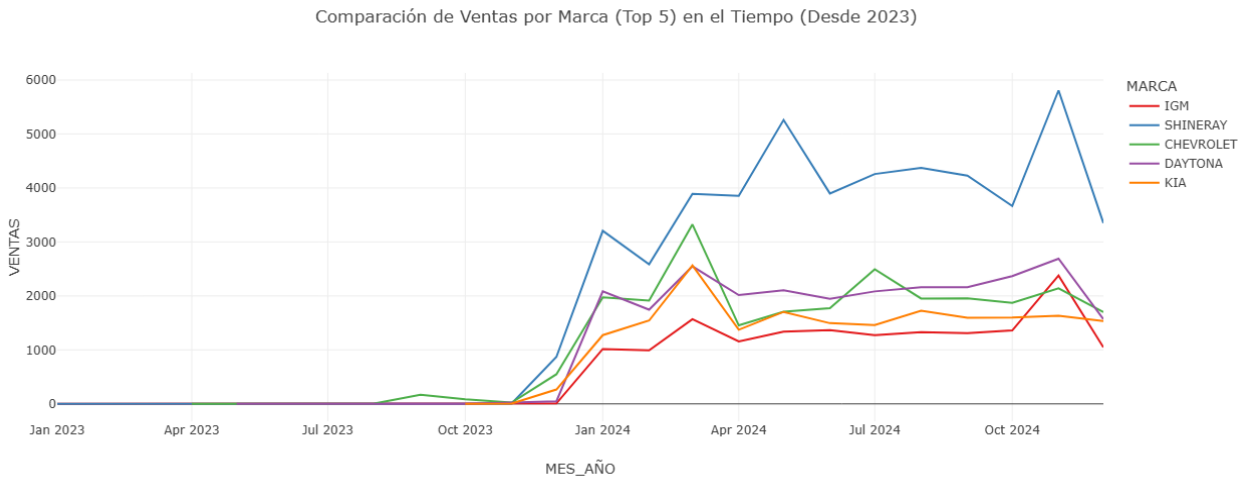
Ventas mensuales de vehículos nuevos (2022–2024)



Se genera un gráfico de línea como se puede ver en la Figura 1, que muestra la evolución mensual de ventas en el periodo 2022-2024, así como una comparación por años individuales. También se analiza la participación de las cinco marcas más vendidas desde 2023: Chevrolet, Toyota, Kia, Shineray e IGM, que se aprecia en la Figura 23. Estas visualizaciones facilitan la identificación de patrones de consumo y evolución de la demanda.

Figura 23.

Comparación de ventas por marcas (top 5) en el tiempo (desde 2023)



4.3. Estacionariedad y Descomposición Temporal

Se aplica dos pruebas estadísticas para evaluar la estacionariedad de la serie del tiempo: la prueba de Dickey-Fuller Aumentada (ADF) y la prueba KPSS. Los resultados indican que la serie no es estacionaria en nivel, lo que requirió aplicar diferenciación para modelos como SARIMA.

- ADF Estadística: -1.2652, p-valor: 0.6450
- KPSS Estadística: 0.6253, p-valor: 0.0203

La descomposición estacional (modelo aditivo) permite separar los componentes de tendencia, estacionalidad y residuales. Se identifica una tendencia creciente y una estacionalidad claramente definida con periodicidad anual, como se visualiza en la Figura 24 y Figura 25.

Figura 24.

Descomposición de la serie de tiempo (Ventas)

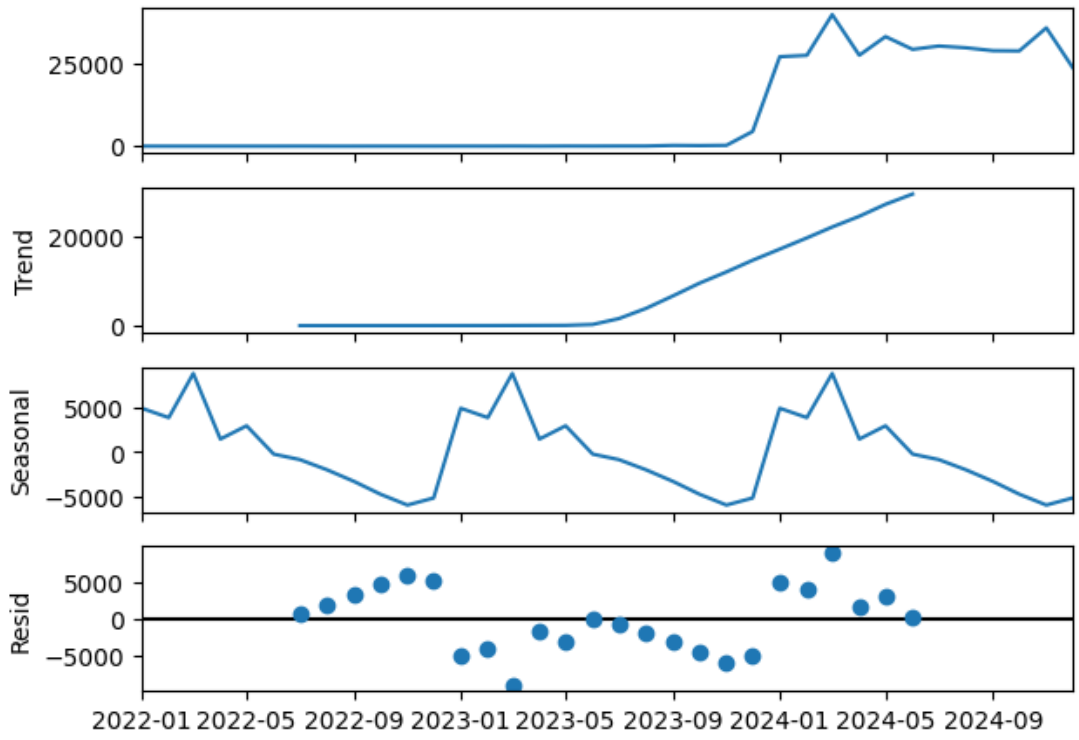
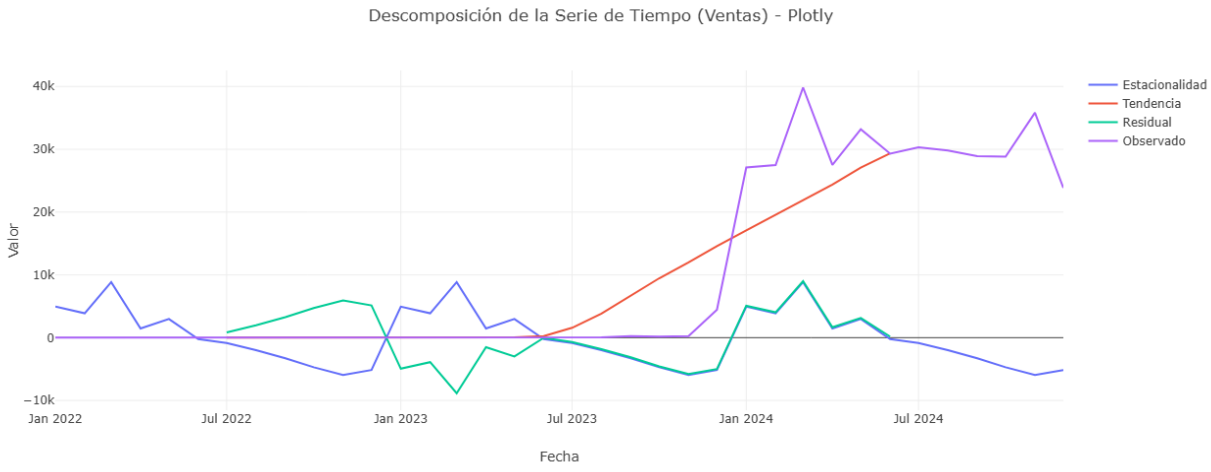


Figura 25.

Descomposición de la serie temporal de ventas (Plotly)



4.4. Aprendizaje Supervisado

El aprendizaje supervisado se basa en una técnica dentro del campo del aprendizaje automático (machine learning), consisten en entrenar modelos utilizando un conjunto de datos etiquetados, es decir, en el que cada entrada se asocia con una salida esperada. Este tipo de aprendizaje tiene como objetivo predecir una variable de salida a partir de un conjunto de variables de entrada, estableciendo una función que generaliza patrones observados en los datos En este proyecto se utiliza tres tipos de modelos los cuales se detallan en los siguientes apartados.

4.4.1. Modelado Predictivo y Evaluación

Para el pronóstico de ventas mensuales se implementaron tres modelos:

- ARIMA (SARIMAX): modelo estadístico autoregresivo con componentes estacionales.
- Prophet: modelo aditivo desarrollado por Facebook que maneja estacionalidades automáticamente.
- ETS (Error, Trend, Seasonality): modelo de suavizamiento exponencial triple.

Tabla 1

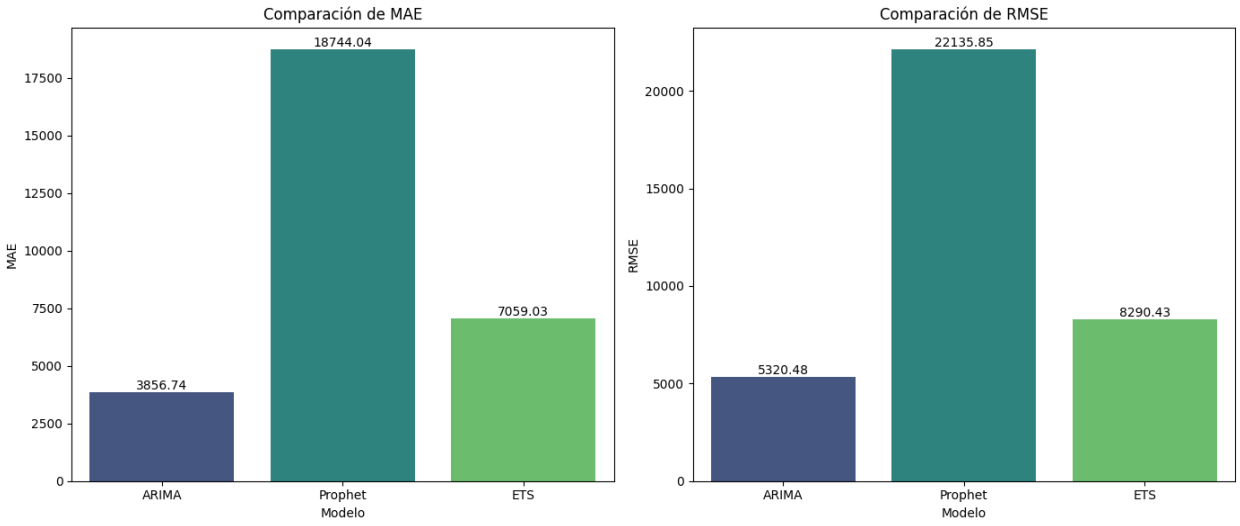
Valores obtenidos mediante modelos de pronósticos

Modelo	MAE (vehículos)	RMSE (vehículos)
ARIMA	3.857	5.320
Prophet	18.744	22.136
ETS	7.060	8.291

Cada modelo se ajusta usando los datos hasta diciembre de 2023, y evaluado en el periodo de prueba enero-junio 2024. Las predicciones se comparan con los datos reales y se calcularon las métricas MAE y RMSE, indicados en la Tabla 1 y se visualiza en la Figura 26

Figura 26.

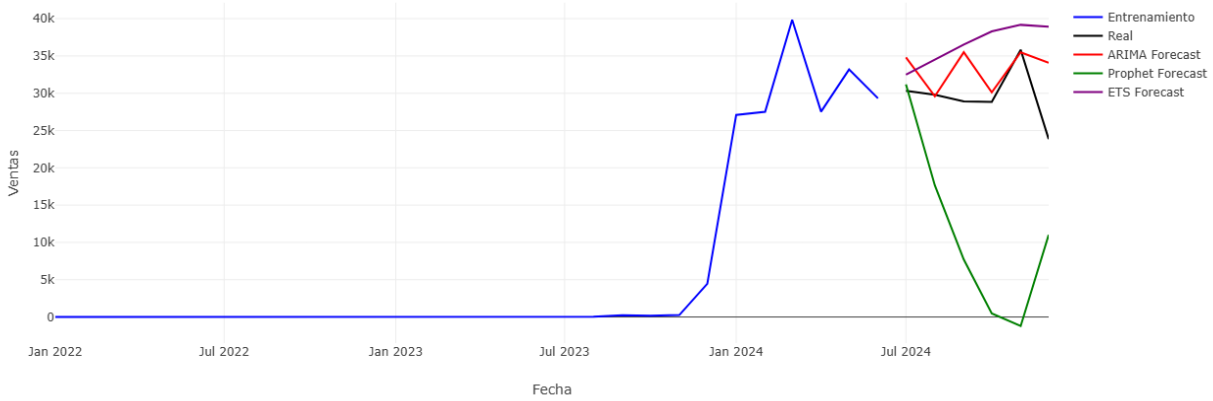
Comparación de MAE y RMSE en modelos predictivos



El modelo ETS (Exponential Smoothing), basado en suavizamiento exponencial triple, demuestra tener el mejor desempeño predictivo, al capturar adecuadamente tanto la tendencia como estacionalidad de la serie.

Figura 27.

Comparación de Pronósticos de ARIMA vs Prophet vs ETS



El modelo ETS presenta el mejor desempeño, siendo el más preciso y consistente, especialmente en la captura de la estacionalidad de la serie. Esta afirmación se respalda en la visualización de la Figura 27, donde se superponen los datos reales, los valores de entrenamiento y las predicciones generadas por cada modelo.

4.5. Aprendizaje No Supervisado

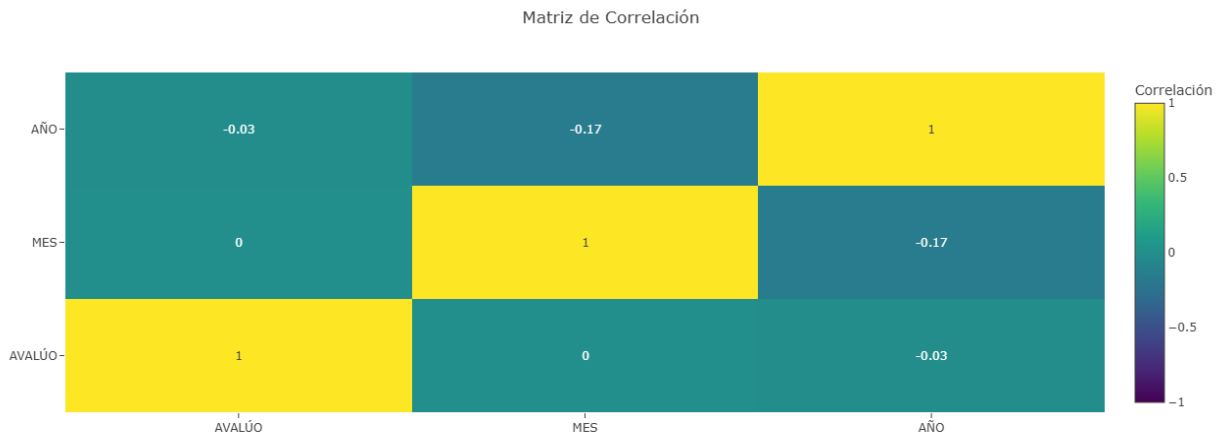
El aprendizaje no supervisado, se enfoca en métodos en los que los algoritmos analizan datos que no contienen etiquetas o salidas predefinidas. El objetivo principal se basa en descubrir estructuras ocultas, patrones o relaciones entre los datos sin intervención externa. Este enfoque es comúnmente utilizado en técnicas como correlación, Clustering con enfoque en KMeans, redes bipartitas y la reducción de dimensionalidad, las cuales resultan útiles especialmente en contextos donde el etiquetado de los datos no es factible o es costoso.

4.5.1. Correlación y Segmentación por Avalúo

Se construye una matriz de correlación entre las variables cuantitativas (avalúo, mes y año), que se aprecia en la Figura 28. Se observa una baja correlación entre el avalúo y las variables temporales, lo cual se justifica su uso como variable independiente para segmentación.

Figura 28.

Matriz de Correlación



Con base en ello, se aplica el algoritmo K-Means para clasificar los vehículos en tres categorías según su avalúo:

- Económico
- Intermedio
- Lujo

Esta clasificación automática se visualiza en histogramas que se aprecia en la Figura 29, y posteriormente se usa para segmentar y analizar el comportamiento por grupo, para consiguiente realizar un análisis de Clustering por avalúo y año de compra como se observa en la Figura 30.

Figura 29.

Clasificación automática por rango de AVALÚO (económico, intermedio, lujo)

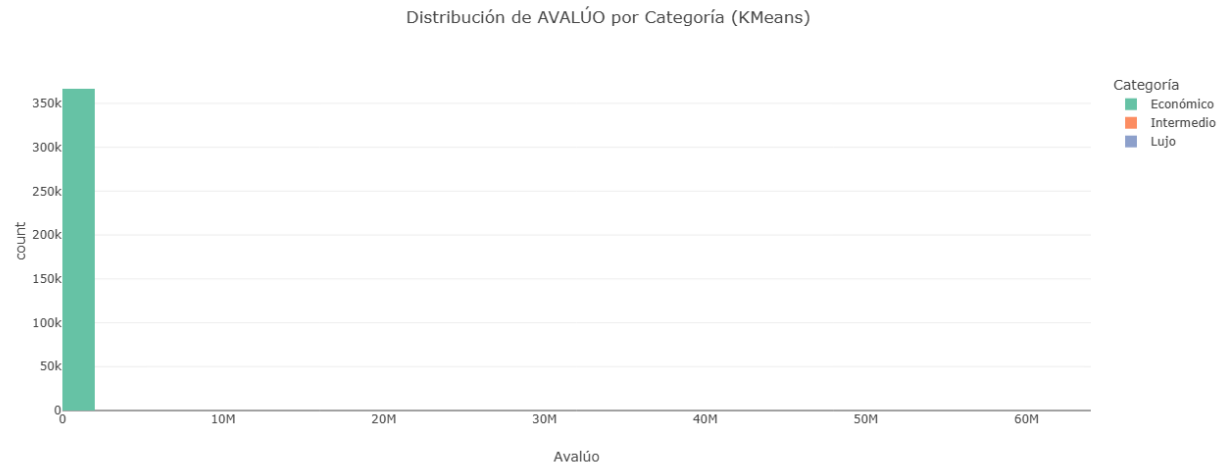
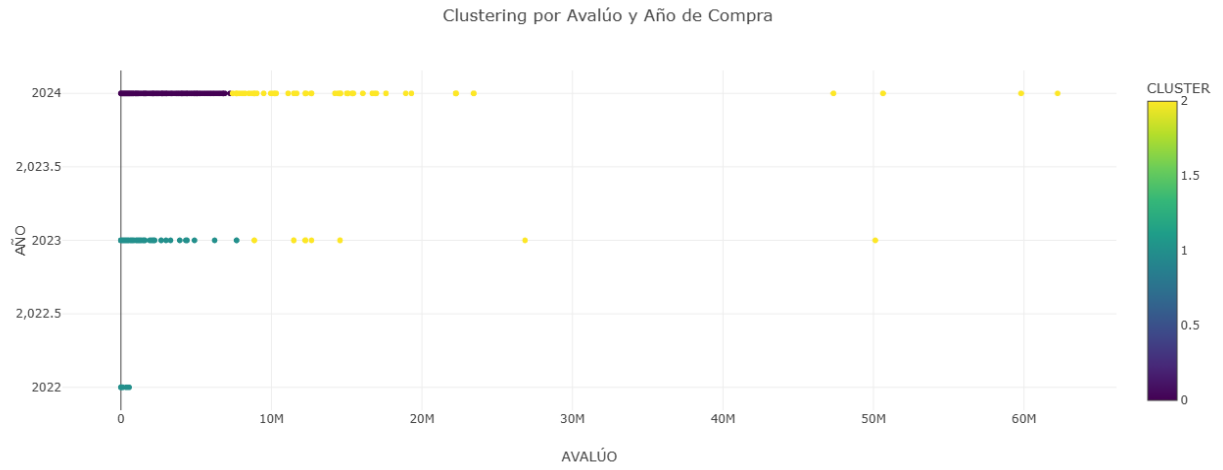


Figura 30.

Clustering por Avalúo y Año de compra



4.5.2. Análisis de Redes

Se implementa representaciones de grafos bipartitos con NetworkX para analizar relaciones entre entidades del dataset. Se genera dos tipos de red:

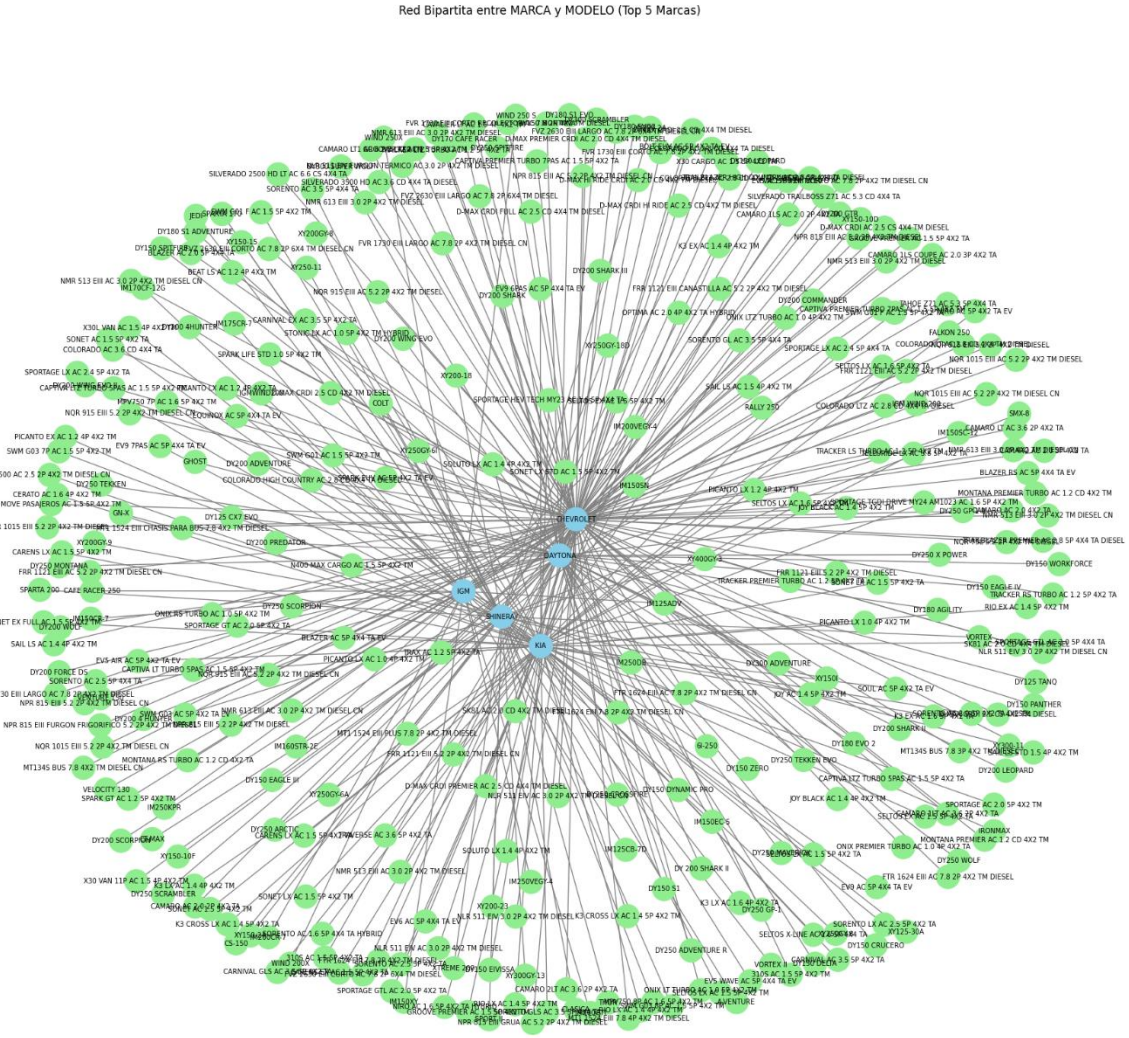
- **MARCA – MODELO:** conectividad entre marcas y modelos únicos.

- **MARCA – TIPO DE COMBUSTIBLE:** asociación entre marcas y tipos de combustible.

En la red MARCA – MODELO que se observa en la Figura 31, se identificaron 267 nodos tipo marca y 3442 nodos tipo modelo. El análisis de centralidades revela que TOYOTA, CHEVROLET y HYUNDAI tenían los grados de conectividad más altos.

Figura 31.

Red Bipartita entre MARCA y MODELO (Top 5 Marcas)



Teniendo como resultado un top 5 con mayor centralidad en la red bipartita, la cual se representa en la Tabla 2.

Tabla 2

Top 5 nodos con mayor centralidad en Red MARCA–MODELO

	Nodo	Grado	Betweenness	Closeness
TOYOTA	TOYOTA	0.038607	0.001480	0.038607
CHEVROLET	CHEVROLET	0.033747	0.001130	0.033747
HYUNDAI	HYUNDAI	0.028348	0.000796	0.028348
MERCEDEZ BENZ	MERCEDEZ BENZ	0.027268	0.000736	0.027268
JAC	JAC	0.026188	0.000679	0.026188

Además, se evaluaron métricas de robustez y resiliencia para las redes. Se encuentra que los nodos de modelo son más frágiles a ataques dirigidos, mientras que los de marca mantenían una mayor integridad ante eliminaciones aleatorias:

- Resiliencia MARCA: 0.91
- Resiliencia MODELO: 0.52
- Robustez MARCA: 0.11
- Robustez MODELO: 0.50

Estos indicadores revelan que el ecosistema de modelos es más vulnerable ante perturbaciones dirigidas, lo que sugiere una alta dependencia de ciertos modelos claves.

En la red MARCA – TIPO COMBUSTIBLE que se observa en la Figura 32, la gasolina fue el tipo con mayor grado de centralidad seguida del diésel y los híbridos, indicando una fuerte dependencia de los consumidores hacia los combustibles tradicionales.

Figura 32.

Red Bipartita: Marca vs Tipo de combustible

Posteriormente, se realiza un top 5 para la red bipartita de MARCA – TIPO COMBUSTIBLE que se representa en la Tabla 3 y Tabla 4, en las cuales se visualizan la centralidad por grado para cada nodo.

Tabla 3

Top 5 Marcas con más tipos de combustible asociados (por Centralidad de Grado)

	Nodo	Centralidad_Grado
30	FORD	0.014652
20	BMW	0.014652
22	TOYOTA	0.014652
16	JEEP	0.014652
46	KIA	0.014652

Tabla 4

Top 5 Tipos de Combustible con más marcas asociadas (por Centralidad de Grado)

	Nodo	Centralidad_Grado
268	GASOLINA	0.7010623
267	DIESEL	0.311355
270	ELECTRICO	0.164835
269	HIBRIDO_GASOLINA_BATERIAS	0.106227
272	HIBRIDO_DIESEL_BATERIAS	0.007326

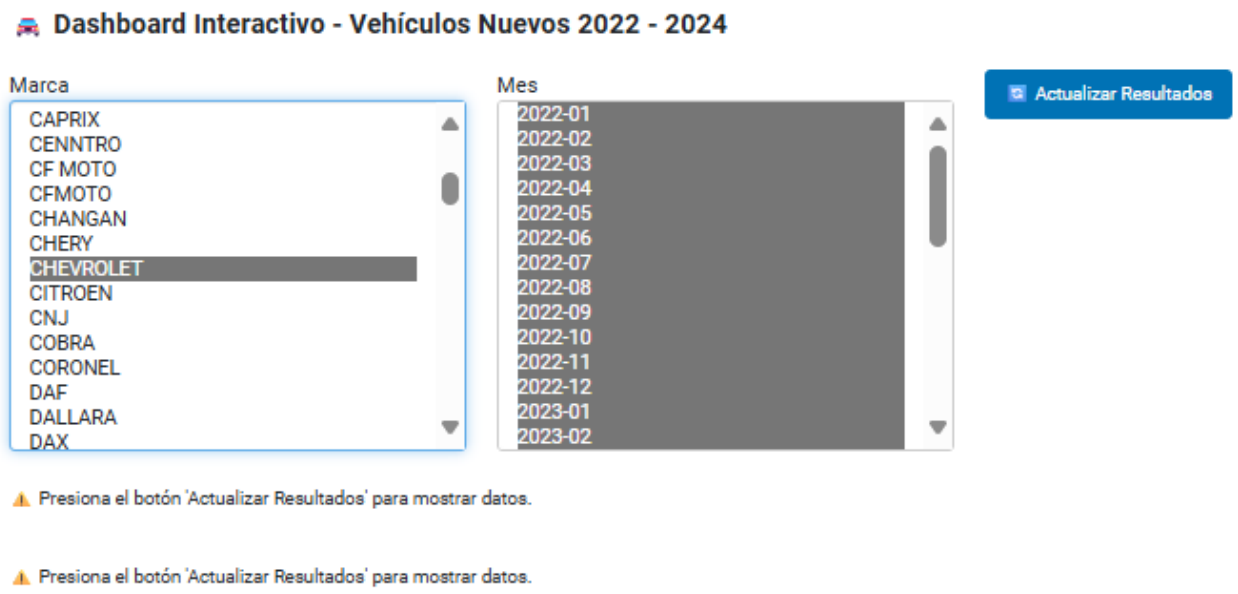
4.6. Dashboard Interactivo

Además, se desarrolla un tablero de visualización interactiva utilizando Panel (librería de Python). Este Dashboard permite al usuario aplicar filtros dinámicos por marca o periodo, observar gráficas actualizadas automáticamente y acceder a visualizaciones de ventas, Clustering y predicciones.

Finalmente, se ajusta el diseño con controles ampliados, personalización estática y componentes dinámicos de actualización que fortalecen la experiencia de análisis exploratorio interactivo tal como se visualiza en la Figura 33.

Figura 33.

Interfaz del Dashboard



CAPÍTULO 5:

5. CONCLUSIONES Y RECOMENDACIONES

5.1. Conclusiones

El riguroso preprocesamiento del dataset del Servicio de Rentas Internas (SRI) del Ecuador fue esencial para garantizar la calidad del análisis predictivo. Este incluyó la conversión de datos, eliminación de columnas vacías y estandarización de fechas, asegurando así la fiabilidad de la visualizaciones y modelos generados.

El análisis temporal reveló una marcada estacionalidad anual en las ventas de vehículos nuevos entre el año 2022 y 2024, particularmente con picos al cierre de cada año. Esta tendencia se valida mediante la descomposición estacional y visualización de la serie temporal, donde se observa una tendencia creciente, lo que refleja una recuperación o fortalecimiento del mercado automotor.

Al implementar modelos de predicción ARIMA, Prophet y ETS, muestran rendimientos predictivos diferenciados al evaluar el periodo enero – junio de 2024. En este caso el modelo ETS fue el más preciso obteniendo los menores errores:

- MAE = 7.060 vehículos
- RMSE = 8.291 vehículos

En contraste, Prophet presento un MAE mucho más alto (18.744), lo que sugiere que los métodos de suavizamiento exponencial son más adecuados para series con estacionalidad estable.

La segmentación de vehículos por avalúo, utilizando el algoritmo K-Means, permitió agrupar los datos en tres categorías: económico, intermedio y lujo. Esta clasificación es útil para

comprender la distribución del mercado por rangos de precios y planificar estrategias diferenciadas en la venta de vehículos de la industria automotriz ecuatoriana.

El uso de grafos bipartitos permitió explorar relaciones entre:

- Marca y Modelo, identificando a Toyota, Chevrolet y Hyundai como los nodos más centrales por conectividad y centralidades.
- Marca y Tipo de Combustible, donde la Gasolina dominó con una centralidad por grado de 0.70, evidenciando una fuerte dependencia del mercado hacia combustibles tradicionales.

Así mismo, los indicadores de resiliencia (0.91 para marca vs 0.52 para modelo) muestran que el ecosistema es más frágil a la pérdida de modelos que a la de marcas, por lo cual se sugiere concentración de ventas en pocos modelos clave.

Finalmente, al implementar un Dashboard interactivo con la librería Panel, se fortaleció la exploración visual y la capacidad de análisis del usuario, integrando filtros dinámicos por marca y años, así como gráficos para ventas, segmentación y predicciones. Lo cual al implementar esta herramienta logra representar una buena práctica de ciencia de datos aplicada a la toma de decisiones estratégicas.

5.2. Recomendaciones

Acorde a los resultados obtenidos, se sugiere que, para el mercado automotor ecuatoriano, el uso de técnicas de minería de datos combinadas con modelos de series temporales y visualizaciones lograría representar una poderosa herramienta para comprender y anticipar el comportamiento del consumidor.

En conjunto, este estudio demuestra cómo el análisis de datos puede proveer informaciones estratégicas y generar ventajas competitivas en el sector automotor ecuatoriano.

Entre las recomendaciones más relevantes se tiene:

- Incorporar variables externas como indicadores económicos, normativa tributarias o tasas de interés que podrían enriquecer los modelos de predicción y explicar mejor las variaciones.
- Alimentar regularmente los modelos con datos nuevos y reevaluar su rendimiento para mantener la precisión del pronóstico.
- El análisis de grafos puede utilizarse como indicador temprano de disrupciones o concentraciones excesivas en determinados modelos o marcas.
- Las entidades del sector automotor puedes ayudarse de esta herramienta para anticipar la demanda, optimizar inventarios y definir campañas publicitarias según segmento.

Referencias Bibliográficas

Bawdekar, A. A., Prusty, B. R., & Bingi, K. (2022). Sensitivity Analysis of Stationarity Tests' Outcome to Time Series Facets and Test Parameters. *Mathematical Problems in Engineering*, *2022*, 1–24. https://doi.org/10.1155/2022/2402989

Buriticá, O., & Guerrero, L. (2021). *Introducción a la programación con Python.* (Ra-Ma Editorial, Ed.).

Chimarro-Amaguaña, J., Chuqui-Barriga, F., Guamán-Cullispuma, D., & Quishpe-Farinango, C. (2023). El auge exponencial del lenguaje Python en el desarrollo tecnológico. *Revista Científica INGENIAR: Ingeniería, Tecnología E Investigación.*, *6*, 240–256.

Crespo Marquez A, G. F.-G. (2020). Maintenance Management through Intelligent Asset Management Platforms. *Emerging Factors, Key Impact Areas and Data Models*, 3762. https://doi.org/10.3390/en13153762.

Gallardo, A. (2010). *Procesamiento y limpieza de datos en bases de datos*. Editorial Técnica. Editorial Técnica.

Hassan, A., Shah, S., & Hafeez, A. (2021). Comparative Analysis of Data Visualization Libraries Matplotlib and Seaborn in Python. *International Journal of Advanced Trends in Computer Science and Engineering*, *10*(1), 277–281. https://doi.org/10.30534/ijatcse/2021/391012021

Hongjie, Y., Jun, L., Qinghe, Z., & Dong, H. (2021). Research on automatic robot charging based on infrared and ultrasonic information fusion. *Journal of Physics: Conference Series*, *1971*(1), 012060. https://doi.org/10.1088/1742-6596/1971/1/012060

Hosmani, S. M. (2023). SCDT: robust and reliable secure clustering and data transmission in vehicular ad hoc network using weight evaluation. *Ambient Intell Human Compu*, 2029–2046.https://doi.org/10.1007/s12652-021-03414-3

Ikome, J., Laseinde, O., & Kanakana, M. (2022). The Future of the Automotive Manufacturing Industry in Developing Nations: A Case Study of its Sustainability Based on South Africa's Paradigm. *Procedio Computer Science*, 1165–1173.

Kabir, M. A., Ahmed, F., Islam, M. M., & Ahmed, Md. R. (2024). Python For Data Analytics: A Systematic Literature Review Of Tools, Techniques, And Applications. *ACADEMIC JOURNAL ON SCIENCE, TECHNOLOGY, ENGINEERING & MATHEMATICS EDUCATION*, *4*(04), 134–154. https://doi.org/10.69593/ajsteme.v4i04.146

Kadhim, R., Muna, R., Mohialden, Y., & Mahmood, N. (2022). A Review of the Implementation of NumPy and SciPy Packages in Science and Math. *Turkish Journal of Computer and Mathematics Education (TURCOMAT)*, 663–667.

Kampezidou, S. I., Tikayat, A., Bhat, A. P., Pinon, O. J., & Mavris, D. N. (2023). *Fundamental Components and Principles of Supervised Machine Learning Workflows with Numerical and Categorical Data*. https://doi.org/10.20944/preprints202312.0957.v1

Kevin Riehl, M. N. (2023). Hierarchical confusion matrix for classification performance evaluation. *Journal of the Royal Statistical Society Series C: Applied Statistics,*, 1394- 1412. https://doi.org/10.1093/jrsssc/qlad057.

Konstantin, G., Sergey, T., Vladimir, S., Yaroslav, G., & Timofey, B. (2022). A Large-Scale Comparison of Python Code in Jupyter Notebooks and Scripts. *Cornell University*.

Lal, S. T. (2020). Analysis and Classification of Crime Tweets. *International Conference on Computational Intelligence and Data Science* (págs. 1911–1919. https://doi.org/10.1016/j.procs.2020.03.211). Procedia Computer Science.

Menculini, L., Marini, A., Proietti, M., Garinei, A., Bozza, A., Moretti, C., & Marconi, M. (2021). Comparing Prophet and Deep Learning to ARIMA in Forecasting Wholesale Food Prices. *Cornell University*.

Naeem, S., Ali, A., Anam, S., & Ahmed, M. M. (2023). An Unsupervised Machine Learning Algorithms: Comprehensive Review. *International Journal of Computing and Digital Systems*, *13*(1), 911–921. https://doi.org/10.12785/ijcds/130172

Rahmadi, L. H. (2023). Crop Prediction Using Machine Learning with CRISP-DM Approach. *Lecture Notes in Networks and Systems,*, https://doi.org/10.1007/978-981-99-6550- 2_31

Ridge, M. (2021). Streamlit: Dynamic dashboards for data analysis. *Data Science Today, 6(1),* , 45-58.

Rosa G. Hernández Cuacua, S. D. (2020). Análisis de la incidencia delictiva del fuero común. *Research in Computing Science,* , 451-463,

Ryan, O., Haslbeck, J. M. B., & Waldorp, L. J. (2025). Non-Stationarity in Time-Series Analysis: Modeling Stochastic and Deterministic Trends. *Multivariate Behavioral Research*, *60*(3), 556–588. https://doi.org/10.1080/00273171.2024.2436413

Saabith, S., Vinothraj, T., & Fareez, M. (2021). A review on Python libraries and Ides for Data Science. *Int. J. Res. Eng. Sci,* 36–53.

SRI-Servicio de Rentas Internas. (2024). *Estadísticas vehículos 2024 - Datos abiertos Ecuador.* . Https://Www.Datosabiertos.Gob.Ec/Dataset/Estadisticas-Vehiculos-2024.

Sundaram, J., Gowri, K., Devaraju, S., Gokuldev, S., Jayaprakash, S., Anandaram, H., Manivasagan, C., & Thenmozhi, M. (2023). *An Exploration of Python Libraries in Machine Learning Models for Data Science* (pp. 1–31). https://doi.org/10.4018/978-1-6684-8696-2.ch001

Suyal, M., & Sharma, S. (2024). A Review on Analysis of K-Means Clustering Machine Learning Algorithm based on Unsupervised Learning. *Journal of Artificial Intelligence and Systems*, *6*(1), 85–95. https://doi.org/10.33969/AIS.2024060106

Venkatesh, S., Sunil, P., & Oza, K. (2025). Sales Forecasting Prediction using Machine Learning. *InternationalJournal of Electrical, Electronics and Computer Systems*, 19–27.

Vijayaragavan, P. S. (2023). Técnicas de procesamiento del lenguaje natural para análisis de redes sociales. *Journal of Data Science and Applications, 8(3),*, 67-82.

Wankhade, M. R. (2022). A survey on sentiment analysis methods, applications, and challenges. *Artif Intell Rev*, 5731–5780 https://doi.org/10.1007/s10462-022-10144-1.

Zhang, G., Eddy Patuwo, B., & Y. Hu, M. (1998). Forecasting with artificial neural networks: *International Journal of Forecasting*, *14*(1), 35–62. https://doi.org/10.1016/S0169-2070(97)00044-7

Zhou, T., Ren, J., Medo, M., & Zhang, Y.-C. (2007). Bipartite network projection and personal recommendation. *Physical Review E*, *76*(4), 046115. https://doi.org/10.1103/PhysRevE.76.046115