Comparativa de modelos predictivos en la demanda de GNV en la región central de Colombia (2025)

Daniel Adolfo Vásquez Guaje

Asesor

Eduardo Sánchez Sandoval

Universidad Nacional Abierta y a Distancia UNAD

Escuela de Ciencias Básicas Tecnología e Ingeniería ECBTI

Especialización en Ciencia de Datos y Analítica

2025

Resumen

Este estudio tiene como objetivo comparar la precisión de tres modelos de machine learning: regresión lineal, k-nearest neighbors (k-NN) y árboles de decisión, para predecir la demanda de Gas Natural Vehicular (GNV) en Bogotá, Cundinamarca y Boyacá en 2025. Utilizando la metodología SAMPLE (Selección, Análisis, Modelado, Prueba y Evaluación), se tomó los datos históricos de la demanda de GNV, incluyendo variables como número de estaciones activas por ciudad, el número de ventas por día en cada estación y la cantidad de volumen suministrado en las ventas por semana. Los datos fueron limpiados y preprocesados con Python en Google Colab para asegurar su calidad y consistencia. Los modelos fueron entrenados y posteriormente evaluados utilizando las métricas MAE, MSE, R^2 y la visualización de predicciones. Los resultados obtenidos permitieron identificar el modelo más adecuado para la predicción de la demanda de GNV, considerando la precisión de las predicciones y su implementación práctica.

***Palabras claves*:** Machine Learning, Regresión Lineal, k-Nearest Neighbors (k-NN), Árboles de Decisión, Gas Natural Vehicular (GNV)

Abstract

This study aims to compare the accuracy of three machine learning models: linear regression, k-nearest neighbors (k-NN), and decision trees, to predict the demand for Natural Gas Vehicle (NGV) in Bogotá, Cundinamarca, and Boyacá in 2025. Using the SAMPLE methodology (Selection, Analysis, Modeling, Testing, and Evaluation), historical NGV demand data were taken, including variables number of active stations per city, the number of sales per day in each station and the amount of volume supplied in sales per week.. The data was cleaned and preprocessed using Python in Google Colab to ensure its quality and consistency. The models were trained and after evaluated using the metrics MAE, MSE, R2 and prediction visualization. The results obtained made it possible to identify the most suitable model for predicting NGV demand, considering both the accuracy of the predictions and their practical implementation.

Keywords**:** Machine Learning, Linear regresion, k-Nearest Neighbors (k-NN), Decision trees, Natural Gas Vehicle (NGV)

Tabla de Contenido

Lista de Tablas

Lista de Figuras

Lista de Apéndices

Introducción

El presente estudio aborda la predicción de la demanda de Gas Natural Vehicular (GNV) en Bogotá, Cundinamarca y Boyacá para el año 2025, en un contexto donde los combustibles fósiles líquidos predominan, a pesar de sus altos costos y su significativo impacto ambiental. El GNV emerge como una alternativa económica y ecológica, ofreciendo altos ahorros en comparación con la gasolina y reduciendo las emisiones de material particulado. Sin embargo, su adopción se ve limitada por una infraestructura de estaciones de servicio insuficiente y la persistencia de mitos sobre su uso. Este trabajo tiene como objetivo principal comparar la precisión de tres modelos de Machine Learning (Regresión Lineal, k-Nearest Neighbors y Árboles de Decisión) para predecir la demanda de GNV, utilizando la metodología SAMPLE (Selección, Análisis, Modelado, Prueba y Evaluación) sobre datos históricos acerca de la venta de este combustible entre los años 2020 y 2025, con el fin de aportar en la construcción de unas bases sólidas que puedan ser usadas para futuros estudios y predicciones eficientes de la demanda del GNV, ayudando así a dar soporte a las decisiones estratégicas que puedan tomar los inversionistas para la expansión de la infraestructura y fomento de este como combustible.

Descripción del Problema

El uso de combustibles fósiles líquidos, como la gasolina y el diésel, domina el mercado de combustibles en Bogotá, Cundinamarca y Boyacá. Estos combustibles no solo tienen un alto costo, sino que también contribuyen significativamente a la contaminación ambiental. En contraste, el Gas Natural Vehicular (GNV) se presenta como una alternativa más económica y ecológica. Sin embargo, la adopción del GNV está limitada por la escasa disponibilidad de estaciones de servicio, lo que genera desconfianza entre los consumidores y un riesgo percibido de baja calidad del producto.

La falta de infraestructura adecuada para el suministro de GNV impide que los conductores consideren seriamente la conversión de sus vehículos a este tipo de combustible. Según Camilo Guzmán "si existiera un mercado competitivo, los precios reflejarían mejor las condiciones globales, incentivando la eficiencia y la reducción de costos" (Guzman, 2023, p. 1). La Asociación Colombiana de Gas Natural - Naturgas destaca que el GNV puede "representar un ahorro de hasta el 50% en comparación con la gasolina" (Naturgas, 2024, p.1), lo que tendría un impacto positivo en la economía de los conductores.

Además del beneficio económico, el uso de GNV podría reducir significativamente las emisiones de material particulado en la región. El Registro Único Nacional de Tránsito (Runt) indica que "el parque automotor de Colombia está compuesto por más de 6,8 millones de vehículos y 10,2 millones de motocicletas" (Registro Único Nacional de Tránsito, 2023, p. 1), una fuente importante de contaminación. Según estudios realizados "las emisiones de material particulado de un vehículo a gas son un 95% menores que las de un vehículo a diésel" (Observatorio Ambiental de Bogotá, 2017, p. 1), lo que subraya la relevancia de promover el uso de GNV para mejorar la calidad del aire.

Dado este contexto, es importante la predicción de la demanda de GNV considerando que los precios de este tipo de combustibles fósiles pueden ser influenciados por “elementos tales como el clima, la economía, la geopolítica, los cambios tecnológicos, inversión de capital, liquidez de los mercados financieros, mercados de futuros, entre otros”. (Morales et al., 2023, p. 10).

Planteamiento del Problema

Este análisis comparativo de modelos de machine learning permitirá identificar el modelo más adecuado para predecir la demanda de GNV, lo que a su vez puede aportar como información base para decisiones estratégicas sobre la expansión de la infraestructura de estaciones de servicio. Al mejorar la disponibilidad de GNV, se espera no solo fomentar su adopción entre los conductores, sino también contribuir a la reducción de costos y la mejora de la calidad del aire en la región. La pregunta central de este estudio es: ¿Cuál de los modelos de machine learning: regresión lineal, k-nearest neighbors (k-NN) o árboles de decisión, proporciona la predicción más precisa y eficiente de la demanda de Gas Natural Vehicular – GNV en esta región en el 2025?

Justificación

El mercado de combustibles en Bogotá, Cundinamarca y Boyacá está dominado por los combustibles líquidos de origen fósil, como la gasolina y el diésel, que no solo tienen un alto costo, sino que también contaminan liberando compuestos resultado de la combustión como "monóxido de carbono, hidrocarburos sin quemar, óxidos de nitrógeno, compuestos de plomo y partículas, y una pequeña cantidad de óxidos de azufre" (Parker, 2021, p. 656). En contraste, el Gas Natural Vehicular "no genera las emisiones más contaminantes, como las partículas en suspensión, y permite ahorros del 57% respecto a la gasolina" (Ferro Veiga, 2019, p. 85).

Uno de los retos a superar es el poco conocimiento que se tiene de la potencial demanda de GNV y la gran cantidad de mitos relacionados con su uso en los automotores. Las empresas inversionistas desconocen el alto potencial que puede llegar a tener la comercialización de este combustible, ya que los mitos existentes hacen que las personas tengan cierta apatía a adquirir vehículos que funcionen con gas o adaptar los que ya tienen. Los mitos más comunes son: afectaciones mecánicas, nula disposición de estaciones de servicio, los vehículos pueden explotar, los vehículos pierden casi la totalidad de su fuerza.

El desconocimiento de las tendencias emergentes del uso del gas natural en los vehículos genera un miedo el cual "es la respuesta típica del ser humano ante la incertidumbre" (Cruz Tapia, 2024, p. 5), esto por parte de los inversionistas para disponer puntos de venta de este combustible. Sin embargo, con la demostración de un posible tendencia al alza en la demanda del GNV y la progresiva aclaración de los mitos referentes a su uso, es posible motivar a las personas a utilizar este combustible ya que "si se pudieran manejar todos los motivos que llevan a la gente a consumir, podría reducirse la probabilidad de que falle la demanda" (Nivia Gil,

2024, p. 299) y, a su vez, se incentivaría a los inversionistas a aumentar el número de puntos de venta.

Mediante un análisis para identificar el modelo más adecuado para predecir la demanda de GNV, con los resultados obtenidos se busca aportar a los fundamentos que puedan servir para impulsar la inversión en la expansión de la infraestructura de estaciones de servicio, ya que las compañías buscan “tomar decisiones informadas basadas en los datos interpretados y monitorear el resultado para informar decisiones futuras”(Reza & Kamrouz, 2024, p. 2).

Este estudio es fundamental para abordar el problema de la escasa adopción del GNV debido a la falta de conocimiento y los mitos asociados. Al proporcionar predicciones precisas sobre la demanda de GNV, se espera fomentar su adopción entre los conductores, reducir los costos de combustible y mejorar la calidad del aire en la región. Además, este estudio puede servir como base para futuras investigaciones y políticas que promuevan el uso de combustibles más limpios y sostenibles.

Objetivos

Objetivo General

Comparar la precisión de los modelos de machine learning: regresión lineal, k-nearest neighbors (k-NN) y árboles de decisión, mediante las métricas de precisión de MAE, MSE y R^2, para la predicción de la demanda de Gas Natural Vehicular – GNV en Bogotá, Cundinamarca y Boyacá durante el 2025

Objetivos Específicos

Preparar el conjunto de datos de ventas de GNV mediante técnicas de limpieza para un adecuado desempeño de los algoritmos de machine learning.

Ajustar los modelos de machine learning seleccionados utilizando Python y Google Colab, explorando diferentes configuraciones de hiperparámetros para optimizar el rendimiento de cada modelo.

Evaluar la precisión de los modelos de machine learning implementados utilizando las métricas de MAE, MSE, R^2 y la visualización de predicciones, para seleccionar el modelo con la mejor predicción de la demanda de GNV en el segundo semestre del año 2025.

Marco de Referencia

Marco Contextual

La predicción de la demanda de Gas Natural Vehicular (GNV) es crucial para la planificación y gestión de recursos en Bogotá, Cundinamarca y Boyacá. Factores como el precio del combustible, el número de vehículos, las políticas gubernamentales y las tendencias económicas pueden influir en la demanda de GNV. El uso de modelos matemáticos para la identificación de patrones de comportamiento y la predicción de valores basados en los registros históricos de datos, es una herramienta muy útil en la ciencia de datos, a esta técnica se le conoce como Machine Learning que es "la aplicación y la ciencia de algoritmos que dan sentido a los datos, es el campo más apasionante de todas las ciencias informáticas" (Raschka & Mirajalili, 2018, p. 1). Utilizar modelos de machine learning para predecir esta demanda puede ayudar a las autoridades y empresas a tomar decisiones informadas y optimizar la distribución y el suministro de GNV.

Para la predicción de la demanda de Gas Natural Vehicular (GNV), este estudio se centra en tres modelos específicos: regresión lineal, k-Nearest Neighbors (k-NN) y árboles de decisión, estos modelos hacen parte de los modelos de aprendizaje supervisado más utilizados, y que se caracterizan por "pronosticar las etiquetas de clase categóricas de las nuevas instancias o puntos de datos basándose en observaciones anteriores" (Raschka Sebastian, 2023, p. 3).

Marco Teórico

Regresión Lineal

La regresión lineal es uno de los métodos más simples y ampliamente utilizados en el análisis predictivo. Se basa en la relación lineal entre una variable dependiente y una o más variables independientes. Esta "consiste en ajustar una línea recta a un conjunto de observaciones" (Kane, 2017, p. 134). La fórmula general de la regresión lineal es:

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n + \in$$

Donde "y" es la variable dependiente, $x_1, x_2, \ldots x_n$ son las variables independientes, β_0 es la intersección, $\beta_1 + \beta_2, \ldots, \beta_n$ son los coeficientes de regresión y $\in$ es el término de error.

Es un método muy útil por su simplicidad y facilidad de interpretación, eficiencia computacional y buen rendimiento con datos lineales; sin embargo, es limitado a relaciones lineales, no captura interacciones complejas entre variables y es sensible a outliers o valores atípicos que son "datos muy diferentes a la mayoría de los valores de datos" (Bruce et al., 2022, p. 15). Según Castro Zuluaga, es conveniente utilizarlo "cuando existe un patrón de comportamiento con tendencia de los datos históricos y las fluctuaciones aleatorias de los mismos son pequeños" (Castro Zuluaga, |C. A., 2020, p. 143)

K-Nearest Neighbors (k-NN)

El algoritmo k-NN es un método de aprendizaje supervisado que se utiliza tanto para clasificación como para regresión. En este se "realiza predicciones basadas en el valor de salida medio entre los k vecinos más cercanos" (Prathmesh Yelne, 2023, p. 190), el valor de la variable dependiente se puede predecir como el promedio de los valores de los k vecinos más cercanos. La distancia entre los puntos se calcula generalmente utilizando la distancia euclidiana.

Este modelo es fácil de entender e implementar, no hace suposiciones sobre la distribución de los datos, es flexible y puede capturar relaciones no lineales; no obstante, “el valor de k puede afectar significativamente el rendimiento del algoritmo, ya que un valor muy bajo puede llevar a una clasificación o predicción demasiado sensible al ruido, mientras que un valor muy alto puede suavizar demasiado las fronteras de decisión o las tendencias en el caso de regresión” (Velasco Rebolledo, 2024, p. 75). En este modelo es importante el escalado porque el algoritmo K-NN funciona midiendo la distancia entre los puntos de datos, y asegura que todas las características o variables contribuyan equitativamente a la distancia. Si se tienen múltiples variables predictoras con rangos de valores muy diferentes, la variable con el rango más grande dominaría el cálculo de la distancia, haciendo que la otra variable sea prácticamente irrelevante. En el caso que exista una única variable, escalarla es una buena práctica general.

Árboles de Decisión

Los árboles de decisión son modelos predictivos que dividen los datos en subconjuntos basados en valores de las variables independientes. Frank Kane afirma que se observa “básicamente como un diagrama de flujo de cómo tomar una decisión” (Kane, 2017, p. 183). Cada nodo interno representa una prueba en una variable, cada rama representa el resultado de la prueba y cada hoja representa una predicción del valor de la variable dependiente. Los árboles de decisión pueden manejar tanto variables categóricas como continuas, estos “proporcionan una forma más gráfica de representar las alternativas o eventos que surgen a partir de la elección de una opción” (Tito et al., 2023, p. 87).

El uso de este modelo es muy común gracias a que es fácil de interpretar y visualizar, no requiere normalización de datos y maneja bien datos no lineales y relaciones complejas; aunque para su implementación es necesario considerar que es propenso al sobreajuste, es sensible a

pequeñas variaciones en los datos, y puede ser ineficiente con datos muy grandes. Este modelo es propenso al sobreajuste si se les permite crecer demasiado. Con una sola variable o característica el árbol podría aprender los ruidos específicos de los datos de entrenamiento, por eso es necesario ajustar hiperparámetros como profundidad máxima del árbol (max_depth) y el número mínimo de muestras requeridas para dividir un nodo interno (min_samples_split), si se observa que las predicciones son muy erráticas o si el R^2 en datos de prueba es mucho menor que en entrenamiento.

Para el entrenamiento de los datos se realiza una división de los mismo que consiste en separar el conjunto original en distintas partes para entrenar y evaluar un modelo de aprendizaje automático. Esta separación suele hacerse dividiendo los datos en un conjunto de entrenamiento, que se utiliza para construir el modelo, y un conjunto de prueba, que sirve para evaluar su desempeño con datos que no ha visto antes. Durante esta división, es común establecer un parámetro llamado random_state, y un valor muy utilizado es random_state=42. Este número no tiene un significado especial en sí mismo, pero al fijarlo, se asegura que la separación de los datos se haga siempre de la misma manera cada vez que se ejecute el código.

Al monitorear el rendimiento de los conjuntos de entrenamiento y prueba para cada combinación, podría darse diferentes escenarios:

Subajuste: Cuando el modelo es demasiado simple (por ejemplo, max_depth muy bajo o min_samples_split muy alto), el rendimiento es pobre tanto en el conjunto de entrenamiento como en el de prueba. El modelo no aprende lo suficiente de los datos.

Sobreajuste: Cuando el modelo es demasiado complejo (por ejemplo, max_depth muy alto y min_samples_split muy bajo), el rendimiento en el conjunto de entrenamiento es

excelente, pero el rendimiento en el conjunto de prueba es significativamente peor. El modelo ha memorizado el ruido de los datos de entrenamiento.

Modelo Óptimo: El objetivo es encontrar la combinación de hiperparámetros donde el modelo tiene un alto rendimiento en el conjunto de prueba y la diferencia entre el rendimiento de entrenamiento y prueba es aceptable, es decir que no hay una brecha demasiado grande que indique sobreajuste.

Feature Engineering

Es un proceso también conocido como ingeniería de características, consiste en transformar o crear nuevas variables a partir de los datos originales con el objetivo de mejorar el rendimiento de los modelos de machine learning. Esta tarea no solo se enfoca en usar los datos tal como vienen, sino en reinterpretarlos, combinarlos o derivar información más relevante que pueda ayudar al modelo a entender mejor los patrones subyacentes.

En esencia, el feature engineering es una forma de añadir conocimiento y contexto al modelo, ya que permite representar la información de manera que sea más comprensible y aprovechable por los algoritmos. Este paso requiere tanto conocimiento técnico como comprensión del dominio del problema, ya que el valor de las nuevas variables depende de qué tan bien capturen la lógica real detrás de los datos.

Comparación de Modelos

Para determinar el modelo más adecuado para la predicción de la demanda de GNV, es esencial comparar el rendimiento de los modelos de regresión lineal, k-NN y árboles de decisión. Existen diversas métricas de evaluación de modelos, lo cual es un proceso importante en la comparación de estos, tal como lo afirma Cuevas, “la evaluación de un modelo entrenado resulta vital para determinar si su funcionamiento es excelente o regular, al realizar predicciones con

nuevos datos o instancias. Debido a que las futuras instancias tienen valores desconocidos para nuestro modelo, es necesario emplear métricas sobre el funcionamiento del modelo de Machine Learning, para determinar si este tendrá la capacidad de generalizar exitosamente datos con los que no fue entrenado" (Cuevas et al., 2021, p. 5), sin embargo, los más comunes para predicción de valores son:

- R^2 (Coeficiente de Determinación): Este varia de 0 a 1 y significa "la diferencia entre las muestras del conjunto de datos y las predicciones realizadas por el modelo. Un valor alto de R al cuadrado determina la menor correlación entre las características dependientes e independientes, por lo que representa un buen modelo de predicción."(Hossain et al., 2023, p. 265)

- Error Cuadrático Medio (MSE): "Es la medida de los errores al cuadrado entre los valores observados y los valores predichos por el modelo. Mide el promedio por el cual las predicciones del modelo difieren de los valores reales" (Velas Rebolledo, 2024, p. 121)

- RMSE (Raíz del Error Cuadrático Medio): Es el valor obtenido al sacar raíz cuadrada al promedio de los errores que un modelo de predicción comete, indicando qué tan cerca están las predicciones del modelo de los valores reales, según el rango que tengan los datos.

- MAE (Mean Absolute Error): Mide la magnitud promedio de los errores de predicción de un modelo, sin tener en cuenta la dirección del error, es decir, si el modelo sobreestimó o subestimó el valor real. Indica, en promedio, cuánto se desvían las predicciones de los valores reales.

La elección del modelo de machine learning adecuado para la predicción de la demanda de GNV depende de varios factores, incluyendo la naturaleza de los datos, los requisitos de precisión y la interpretabilidad del modelo, esto se puede realizar empleando diversas

herramientas de programación, sin embargo, una de las más idóneas es Python, el cual, es “un lenguaje de programación que se destaca por su código legible y limpio. Una de las razones de su éxito es que cuenta con una licencia de código abierto que permite su utilización en cualquier escenario” (Contreras et al., 2023, p. 89).

En estudios realizados referentes a la predicción de la demanda de GNV, muestran que los modelos de previsión energética “forman parte integral de las operaciones de energía y gas natural desde hace décadas. Los esfuerzos por ofrecer alternativas y nuevos métodos para aumentar la eficiencia de los sistemas energéticos reducen los costes de explotación de un sistema energético” (Sharma et al., 2021, p. 9). 9. Según concluye León & Di Scipio-Cimetta (2022) “las energías renovables no podrán satisfacer el 100 % de la demanda para 2040” (p. 99.).

La comparación de modelos predictivos de machine learning para determinar el más adecuado al hacer una predicción, es un paso importante en las predicciones de datos, ya que algunos modelos arrojan mejores resultados según sea el caso, por ejemplo, en el estudio hecho por Tito et al. (2023) se concluye que “a pesar de la eficacia de las técnicas de aprendizaje automático, algunas de ellas arrojaron resultados inferiores en comparación con otras. Este es el caso de las técnicas Árbol de decisión y Naive Bayes, que obtuvieron los mejores resultados en términos de precisión, pero los peores en términos de precisión y exhaustividad” (p.96.).

Para el caso de predicciones donde el objetivo es un resultado categórico, es de resaltar que existen modelos que, en comparación con la Regresión logística, pueden ser más eficientes, como es el caso del estudio realizado por Daniel & Porras (2023) donde se efectuó una comparación y se concluyó que “Random Forest logró una precisión excepcional del 99.9% en datos de prueba. Este superó a los Árboles de Decisión y la Regresión Logística en precisión” (p. 50.).

Marco Normativo

El gas natural vehicular se enmarca en la regulación del sector de hidrocarburos y, más específicamente, del gas natural. La institucionalidad y normatividad colombiana en esta materia define aspectos clave para la producción, transporte, distribución y comercialización del gas. El Ministerio de Minas y Energía, como cabeza del sector, define las políticas públicas en materia energética, incluyendo las relacionadas con la diversificación de la matriz energética y el fomento de combustibles alternativos como el GNV.

Ley 142 de 1994 (Ley de Servicios Públicos Domiciliarios): Esta ley establece el marco general para la prestación de los servicios públicos domiciliarios, incluyendo el servicio de gas combustible. Regula la actuación de las empresas prestadoras, los derechos de los usuarios y la función de las autoridades de regulación y control sobre la comercialización del GNV como un servicio público. (Ley 142 de 1994 - Gestor Normativo, s. f.)

Decreto 1073 de 2015 (Decreto Único Reglamentario del Sector de Minas y Energía): Este decreto compila la normatividad vigente del sector de minas y energía, incluyendo lo relativo a la distribución y comercialización de gas combustible, lo que es fundamental para entender la cadena de suministro del GNV. (Decreto 1073 de 2015 Ministerio de Minas y Energía - Gestor Normativo. s. f.).

Comisión de Regulación de Energía y Gas (CREG): La CREG es el principal ente regulador del sector de energía y gas en Colombia. A través de sus resoluciones, la CREG establece las metodologías tarifarias, las condiciones de calidad del servicio, las obligaciones de información para los agentes del mercado, y otras disposiciones que impactan directamente la viabilidad y expansión del GNV.

El componente ambiental es fundamental en este estudio, ya que el GNV es promocionado como una alternativa más limpia. La normativa ambiental establece los límites de emisiones y los planes para mejorar la calidad del aire.

Ley 99 de 1993 (Ley General Ambiental): Crea el Sistema Nacional Ambiental (SINA) y sienta las bases de la política ambiental en Colombia. (Ley 99 de 1993 - Gestor Normativo, s. f.)

Decreto 1076 de 2015 (Decreto Único Reglamentario del Sector Ambiente y Desarrollo Sostenible): Este decreto compila la normativa ambiental, incluyendo la que regula la calidad del aire y las emisiones contaminantes de fuentes móviles. (Decreto 1076 de 2015 Ministerio de Ambiente y Desarrollo Sostenible - Gestor Normativo. s. f.)

Resoluciones del Ministerio de Ambiente y Desarrollo Sostenible: Se deben considerar las resoluciones que establecen los estándares de calidad del aire y los límites máximos permisibles de emisiones para vehículos automotores. El impulso del uso de GNV se alinea con los esfuerzos por cumplir con estos estándares y mejorar la salud pública.

Planes de Descontaminación Atmosférica: Las principales ciudades de la región central, como Bogotá, cuentan con planes de descontaminación atmosférica que buscan reducir los niveles de material particulado y otros contaminantes. El fomento del GNV es una estrategia contemplada en estos planes para alcanzar las metas de calidad del aire.

Colombia ha expresado un compromiso con la transición energética y la diversificación de su matriz de combustibles. Esto se traduce en políticas y planes que buscan incentivar el uso de alternativas al petróleo y sus derivados.

Plan Energético Nacional (PEN): Si bien es un documento de planificación, el PEN establece las directrices y prioridades del país en materia energética, donde la diversificación de la canasta energética y la promoción de combustibles más limpios suelen ser pilares.

Incentivos para la Conversión a GNV: En el pasado, han existido programas o incentivos fiscales, financieros, entre otros, para la conversión de vehículos a GNV. Aunque pueden variar con el tiempo, la existencia de estos incentivos forma parte del marco de política que busca estimular la demanda.

Metodología

Tipo de Estudio

El presente trabajo sigue la metodología Sampling, Analysis, Model Proccesing, Learning, Evaluation – SAMPLE, ya que contempla los aspectos claves en el objetivo del presente estudio, basándose en los procesos de selección y análisis de los datos, construcción de los modelos predictivos, prueba y evaluación de los modelos. Esta metodología tiene un enfoque adecuado para este estudio ya que es sistemática, flexible, orientada a resultados y enfatizada en la calidad de los datos, alineada idealmente con las fases que se contemplan en el estudio: recolección de datos, entrenamiento de los modelos y el análisis de su rendimiento mediante diversas métricas.

Recolección de Datos

En la fase de recolección de datos, se obtuvo datos históricos sobre la demanda de GNV en Bogotá, Cundinamarca y Boyacá, de la página de datos abiertos del Gobierno de Colombia. Estos datos incluyen variables relevantes como el número de estaciones activas por ciudad, el número de ventas por día en cada estación y la cantidad de volumen suministrado en las ventas del día. Posteriormente, los datos recolectados se inspeccionaron para estimar si era necesario una limpieza y preprocezamiento con Python en Google Colab para asegurar su calidad y consistencia, lo que implicó identificar la necesidad de eliminar valores atípicos, el manejo de datos faltantes y la normalización de las variables.

Método

Para el entrenamiento de los modelos, los datos preprocesados se dividieron en conjuntos de entrenamiento y prueba, utilizando una proporción adecuada para asegurar la validez de los resultados. Se entrenaron con Python en Google Colab empleando la librería sklearn: un modelo

de regresión lineal para predecir la demanda de GNV, un modelo k-NN ajustando el valor de k para optimizar su rendimiento, y un modelo de árboles de decisión ajustando los hiperparámetros necesarios para mejorar su precisión.

En la fase de evaluación los modelos utilizaron las métricas R^2 (Coeficiente de Determinación) para medir la proporción de la varianza explicada por el modelo, MAE (Mean Absolute Error) para medir la precisión de las predicciones, MSE (Mean Squared Error) para penalizar más los errores grandes, y la visualización de gráficas que comparan los valores reales con los predichos.

Finalmente, se compararon los resultados obtenidos por cada modelo en términos de las métricas de evaluación y precisión, y se identificó el modelo más adecuado para la predicción de la demanda de GNV, considerando tanto la precisión de las predicciones y su implementación práctica.

Resultados

Los datos trabajados requirieron una exploración inicial para entenderlos a fondo, identificando su forma y presentación. Los datos fueron recolectados por la Dirección de Hidrocarburos del Ministerio de Minas y Energía de Colombia, en el periodo comprendido entre el primero de enero de 2020 y el diez de junio de 2025, el tamaño inicial de la base denominada "Ventas de Gas Natural Comprimido Vehicular", tomada de la página de datos abiertos del Gobierno de Colombia, contenía una cantidad de 161.102 registros y 15 variables, sin embargo al filtrarlo para Bogotá, Cundinamarca y Boyacá, según el objetivo del presente estudio, se obtuvo una base de datos de 15 variables y 36.729 registros. A continuación se describe la estructura y el contenido conjunto de datos.

Tabla 1

Listado y Tipo de Variables de la Base de Datos

Variable	Tipo
FECHA_VENTA	Fecha
ANIO_VENTA	Número
MES_VENTA	Número
DIA_VENTA	Número
CODIGO_MUNICIPIO_DANE	Número
DEPARTAMENTO	Texto
MUNICIPIO	Texto
LATITUD	Número
LONGITUD	Número
TIPO_AGENTE	Texto

Variable	Tipo
TIPO_DE_COMBUSTIBLE	Texto
EDS_ACTIVAS	Número
NUMERO_DE_VENTAS	Número
VEHICULOS_ATENDIDOS	Número
CANTIDAD_VOLUMEN_SUMINISTRADO	Número

Nota. Nombres de las variables y su tipo.

Tabla 2

Descripción de Variables de la Base de Datos

Variable	Descripción
FECHA_VENTA	Fecha de la venta
ANIO_VENTA	Año de la venta
MES_VENTA	Mes de la Venta
DIA_VENTA	Dia de la venta
CODIGO_MUNICIPIO_DANE	Código DANE único del municipio de venta
DEPARTAMENTO	Nombre del Departamento de la venta
MUNICIPIO	Nombre del Municipio de la venta
LATITUD	Latitud punto de la venta
LONGITUD	Longitud punto de la venta
TIPO_AGENTE	Tipo de agente vendedor
TIPO_DE_COMBUSTIBLE	Tipo de combustible vendido

Variable	Descripción
EDS_ACTIVAS	Número de Estaciones de Servicio Activas incluidas en el reporte del día
NUMERO_DE_VENTAS	Número de ventas de GNV por día
VEHICULOS_ATENDIDOS	Número de vehículos abastecidos con GNV por día
CANTIDAD_VOLUMEN_SUMINISTRADO	Número de Metros Cúbicos de GNV vendidos por día

Nota. Nombres de las variables y descripción.

Para observar cómo se encuentran los datos se realizó una inspección de las principales características de estadística descriptiva a las variables numéricas con información relevante, observando su rango, media y desviación estándar. A continuación, se muestra resumen estadístico obtenido.

Tabla 3

Resumen Estadístico de las Variables Numéricas

	EDS_ACTIVAS	NUMERO_DE_VENTAS	VEHICULOS_ATENDIDOS	CANTIDAD_VOLUMEN_SUMINISTRADO
Cantidad	36729	36729	36729	36729
Media	6,41	1,62	1,09	1,43
mínimo	1	1	1	1
0,25	1	94	81	5179

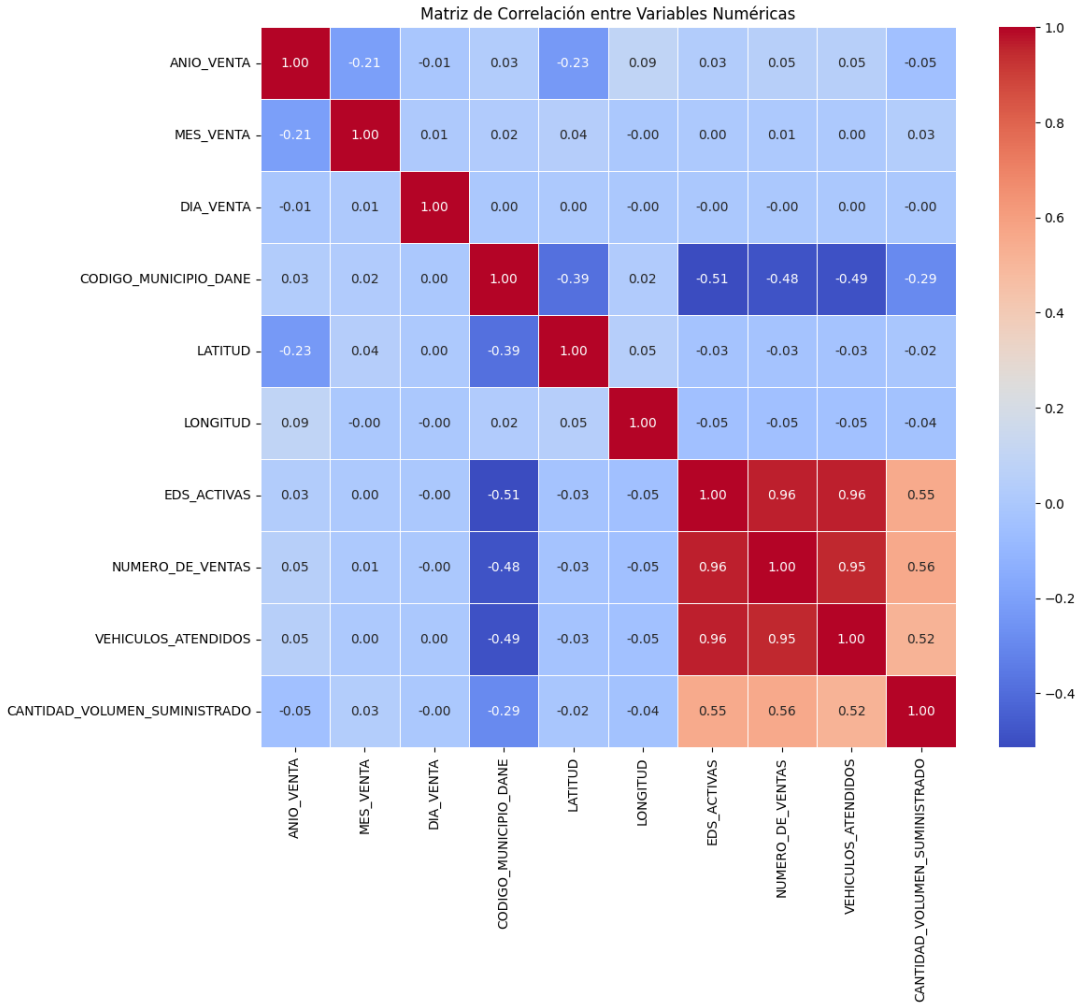
	EDS_ACTIVAS	NUMERO_DE_VENTAS	VEHICULOS_ATENDIDOS	CANTIDAD_VOLUMEN_SUMINISTRADO
0,5	1	186	150	102426
0,75	2	394	287	285522
Máximo	118	59792	29474	570835762
Desviación estándar	2,05	6,36	4,12	9,24

Nota. Resumen estadístico variables numéricas relevantes.

La correlación entre las variables se realizó empleando las librerías pandas, matplotlib y seaborn, se logró observar las relaciones lineales existentes, tomando como variable objetivo CANTIDAD_VOLUMEN_SUMINISTRADO, se encontró cuáles son las variables más influyentes sobre esta. En la Figura 1 se muestra el resultado de la correlación de las variables numéricas entre sí.

Figura 1

Matriz de Correlación de Todas Variables Numéricas



Se observa que existen tres variables que resaltan más que las demás, lo cual según el origen de los datos evidencia la relación existente entre ellas, esto confirma la proporcionalidad que existe, ya que, a mayor cantidad de estaciones activas, mayor número de vehículos atendidos y por esto un mayor volumen de GNV suministrado. Los valores de correlación de las variables más influyentes sobre la variable objetivo se describen a continuación:

- NUMERO_DE_VENTAS (Correlación: 0.56): Existe una correlación positiva moderada-fuerte entre el número de ventas y la cantidad de volumen suministrado. Esto significa que a medida que aumenta el número transacciones de venta, también aumente la cantidad de combustible que se suministra.
- EDS_ACTIVAS (Correlación: 0.55): Hay una correlación positiva moderada-fuerte entre el número de Estaciones de Servicio (EDS) activas y la cantidad de volumen suministrado. Esto sugiere que en los lugares donde hay más estaciones de servicio operando, se tiende a suministrar un mayor volumen de combustible. Esto podría deberse a una mayor demanda o a una mayor capacidad de oferta en esas áreas.
- VEHICULOS_ATENDIDOS (Correlación: 0.52): Se observa una correlación positiva moderada entre la cantidad de vehículos atendidos y el volumen de combustible suministrado. Se confirma que, cuantos más vehículos son abastecidos, mayor es el volumen total de combustible que se despacha.

Con la exploración inicial del conjunto de datos se logró entender la forma en que se presentaban los datos, y así identificar las variables que se debían preparar para incluir en los modelos según el objetivo del estudio.

Resultados Objetivo 1

Las variables de interés para el estudio fueron la CANTIDAD_VOLUMEN_ SUMINISTRADO y FECHA_VENTA, las cuales se identificaron como fundamento para el cumplimiento del objeto. Empleando la función .info() de la biblioteca Pandas se obtuvo la información resumen del dataset. En la Figura 2 se observa que existen 36729 registros no nulos en todas las variables, lo que evidencia que todos los valores son tentativamente útiles para el

estudio. Adicionalmente se observa que existen 4 variables de tipo object, 10 tipo int64 y 1 datetime64.

Figura 2

Información General de las Variables y la Cantidad de Registros No Nulos

```
1 df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 36729 entries, 3 to 161099
Data columns (total 15 columns):
 #   Column                        Non-Null Count  Dtype
---  ------                        --------------  -----
 0   FECHA_VENTA                   36729 non-null  datetime64[ns]
 1   ANIO_VENTA                    36729 non-null  int64
 2   MES_VENTA                     36729 non-null  int64
 3   DIA_VENTA                     36729 non-null  int64
 4   CODIGO_MUNICIPIO_DANE         36729 non-null  int64
 5   DEPARTAMENTO                  36729 non-null  object
 6   MUNICIPIO                     36729 non-null  object
 7   LATITUD                       36729 non-null  int64
 8   LONGITUD                      36729 non-null  int64
 9   TIPO_AGENTE                   36729 non-null  object
 10  TIPO_DE_COMBUSTIBLE           36729 non-null  object
 11  EDS_ACTIVAS                   36729 non-null  int64
 12  NUMERO_DE_VENTAS              36729 non-null  int64
 13  VEHICULOS_ATENDIDOS           36729 non-null  int64
 14  CANTIDAD_VOLUMEN_SUMINISTRADO 36729 non-null  int64
dtypes: datetime64[ns](1), int64(10), object(4)
memory usage: 4.5+ MB
```

Como proceso inicial de la limpieza de los datos fue necesaria la observación de los tipos de registros que contiene cada variable, donde se evidenció que las variables no contienen datos nulos o faltantes y se pueden trabajar.

Teniendo en cuenta que la predicción que se buscó realizar se fundamenta en las ventas realizadas por fechas, se hizo necesario realizar un proceso de Feature engineering, en este caso la columna FECHA_VENTA, utilizando la librería sklearn y matplotlib, se extrajo el día de la semana, el mes y el año. También, dado que los registros de ventas se tenían por día, se procedió

a realizar la suma de los valores de la variable CANTIDAD_VOLUMEN_SUMINISTRADO por semana, para ello se creó un par de variables denominadas SEMANA_ANIO y ANIO.

Figura 3

Creación de Variables de Número de Semana del Año y Año

```
#Importar Librerías necesarias
from sklearn.preprocessing import StandardScaler
from matplotlib.dates import MonthLocator, WeekdayLocator, DateFormatter

# Crear una columna de semana del año para agrupar
df['SEMANA_ANIO'] = df['FECHA_VENTA'].dt.isocalendar().week.astype(int)
df['ANIO'] = df['FECHA_VENTA'].dt.year

# Agrupar por año y semana del año para obtener la suma semanal de CANTIDAD_VOLUMEN_SUMINISTRADO
# Guardar en nuevo DataFrame 'df_semanal' que contiene todos los datos históricos agregados por semana.
df_semanal = df.groupby(['ANIO', 'SEMANA_ANIO'])['CANTIDAD_VOLUMEN_SUMINISTRADO'].sum().reset_index()

# Crear una representación numérica de la semana para el modelo, considerando 52 semanas por año
# Este 'SEMANA_INDEX' es la variable predictora principal basada en el tiempo.
df_semanal['SEMANA_INDEX'] = (df_semanal['ANIO'] - df_semanal['ANIO'].min()) * 52 + df_semanal['SEMANA_ANIO']

print(df.columns)
print(df_semanal.head())
```

```
Index(['FECHA_VENTA', 'ANIO_VENTA', 'MES_VENTA', 'DIA_VENTA',
       'CODIGO_MUNICIPIO_DANE', 'DEPARTAMENTO', 'MUNICIPIO', 'LATITUD',
       'LONGITUD', 'TIPO_AGENTE', 'TIPO_DE_COMBUSTIBLE', 'EDS_ACTIVAS',
       'NUMERO_DE_VENTAS', 'VEHICULOS_ATENDIDOS',
       'CANTIDAD_VOLUMEN_SUMINISTRADO', 'SEMANA_ANIO', 'ANIO'],
      dtype='object')
   ANIO  SEMANA_ANIO  CANTIDAD_VOLUMEN_SUMINISTRADO  SEMANA_INDEX
0  2020           1                        6687888             1
1  2020           2                       18273691             2
2  2020           3                       24946238             3
3  2020           4                       23568168             4
4  2020           5                       46362617             5
```

Como se observa en la Figura 3, se creó un nuevo dataframe denominado df_semanal y para un mejor manejo de la información, se le agregó una nueva variable denominada SEMANA_INDEX que representa el número consecutivo de la semana en todo el periodo estudiado; esto permitió asignar la suma del volumen suministrado en total de los días de la semana al número de la semana.

Se asignó a la variable independiente el nombre de “X” que contiene la información de SEMANA_INDEX del dataframe df_semanal, y a la variable dependiente el nombre de “y” que

contiene la información de CANTIDAD_VOLUMEN_SUMINISTRADO del dataframe df_semanal.

Para la correcta ejecución y predicción del modelo de k-NN se hizo necesario un proceso de escalado de datos, el cual consistió en ajustar los valores de la variable independiente para que tenga un escala o rango estándar. En este caso al tener SEMANA_INDEX como única variable, no era estrictamente necesario, pero se realizó como buena práctica para la óptima ejecución del modelo. A continuación, en la Figura 4, se observa el proceso de escalado que se realizó a los datos.

Figura 4

Escalado de Variable Independiente

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

# Definición la variable independiente (X) y la dependiente (y)
X = df_semanal[['SEMANA_INDEX']]
y = df_semanal['CANTIDAD_VOLUMEN_SUMINISTRADO']

# Proceso de escalado de variables
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Mostrar un resumen estadístico de la variable escalada
print("Resumen estadístico de X_scaled:")
print(pd.DataFrame(X_scaled, columns=['SEMANA_INDEX_ESCALADA']).describe())
```

```
Resumen estadístico de X_scaled:
       SEMANA_INDEX_ESCALADA
count           2.860000e+02
mean            4.968830e-17
std             1.001753e+00
min            -1.722376e+00
25%            -8.645823e-01
50%            -6.788405e-03
75%             8.632161e-01
max             1.733221e+00
```

Para evaluar de manera objetiva los comportamientos de los modelos cuando se enfrentaron a datos nuevos, que no había visto durante su entrenamiento, se realizó el proceso de

división de los datos, utilizando la librería sklearn, se estableció con un 70 % para entrenamiento y un 30% para evaluación. A continuación, se puede observar en la Figura 5 el proceso de división aplicado a los datos.

Figura 5

Proceso de División de los Datos

```python
import pandas as pd
from sklearn.model_selection import train_test_split

# Division de los datos en conjuntos de entrenamiento (70%) y prueba (30%)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

# Calcular tamaños
total = len(X)
train_size = len(X_train)
test_size = len(X_test)

# Crear tabla resumen
tabla_ajuste = pd.DataFrame({
    'Conjunto': ['Entrenamiento', 'Prueba', 'Total'],
    'Número de muestras': [train_size, test_size, total],
    'Porcentaje': [f"{train_size / total * 100:.1f}%", f"{test_size / total * 100:.1f}%", "100%"]
})

print("\nTabla de distribución tras la división:")
print(tabla_ajuste.to_string(index=False))

```

```
Tabla de distribución tras la división:
     Conjunto  Número de muestras Porcentaje
Entrenamiento                 200      69.9%
       Prueba                  86      30.1%
        Total                 286       100%
```

Como parte de la preparación de los datos se realizó el proceso de entrenamiento de los modelos, para el caso del modelo de regresión lineal se empleó la librería sklearn, aquí se tomó la clase LinearRegression de la librería, digitando LinearRegression() y se guardó en la variable u objeto llamado “model”, enseguida con la función .fit() de la clase LinearRegression se entrenó el modelo con los datos guardados en las variables “X” y “y”. El algoritmo de regresión lineal realizó los cálculos necesarios para encontrar la línea recta que mejor se ajustó a los datos (X, y),

calculando la pendiente y el intercepto de la ecuación de la línea (Y=mX+b) que minimizan la suma de los errores cuadrados entre las predicciones del modelo y los valores reales de “y”. Una vez que fit() se ejecutó, el objeto “model” aprendió esta relación y quedó listo para hacer las predicciones sobre nuevos datos.

Figura 6

Proceso de Entrenamiento de los Datos para el Modelo de Regresión Lineal

```
# Entrenamiento del Modelo de Regresión Lineal

from sklearn.linear_model import LinearRegression

# Inicializar el modelo
model = LinearRegression()

# Entrenamiento el modelo con todos los datos disponibles hasta el 10 de junio de 2025
# df_semanal ya contiene los datos agregados hasta esa fecha
model.fit(X, y)

# Imprimir un resumen del entrenamiento del modelo
print("Resumen del Entrenamiento del Modelo de Regresión Lineal")
print(f"Número de características de entrada: {X.shape[1]}")
print(f"Intercepto del modelo: {model.intercept_:.4f}")
print("\nCoeficientes del modelo por característica:")
for i, col in enumerate(X.columns):
    print(f"  {col}: {model.coef_[i]:.4f}")
```

```
Resumen del Entrenamiento del Modelo de Regresión Lineal
Número de características de entrada: 1
Intercepto del modelo: 232336311.0677

Coeficientes del modelo por característica:
  SEMANA_INDEX: -341831.5812
```

El entrenamiento del modelo K-NN se realizó empleando la clase KNeighborsRegressor de la librería sklearn, digitando KNeighborsRegressor(n_neighbors=5) y se guardó en la variable u objeto llamado “knn_model”, enseguida con la función .fit() de la clase KNeighborsRegressor se entrenó el modelo con los datos guardados en las variables “X_scaled” y “y”. El

hiperparámetro n_neighbors que define el número de vecinos más cercanos que el modelo debe considerar cuando hace una predicción se estableció inicialmente en 5. Ver figura 7.

Figura 7

Proceso de Entrenamiento de los Datos para el Modelo de K-NN

```
# Entrenamiento del Modelo K-NN para Regresión
from sklearn.neighbors import KNeighborsRegressor

# Inicializar el modelo K-NN para regresión
# Se define un n_neighbors=5 como valor inicial
knn_model = KNeighborsRegressor(n_neighbors=5)

# Entrenar el modelo con todos los datos históricos escalados
knn_model.fit(X_scaled, y)

# Imprimir un resumen de lo que se hizo
print("Resumen del Entrenamiento del Modelo K-NN para Regresión:")
print(f"- Modelo: KNeighborsRegressor")
print(f"- Número de Vecinos (n_neighbors): {5}")
print(f"- El modelo ha sido entrenado exitosamente con los datos escalados (X_scaled, y).")

```

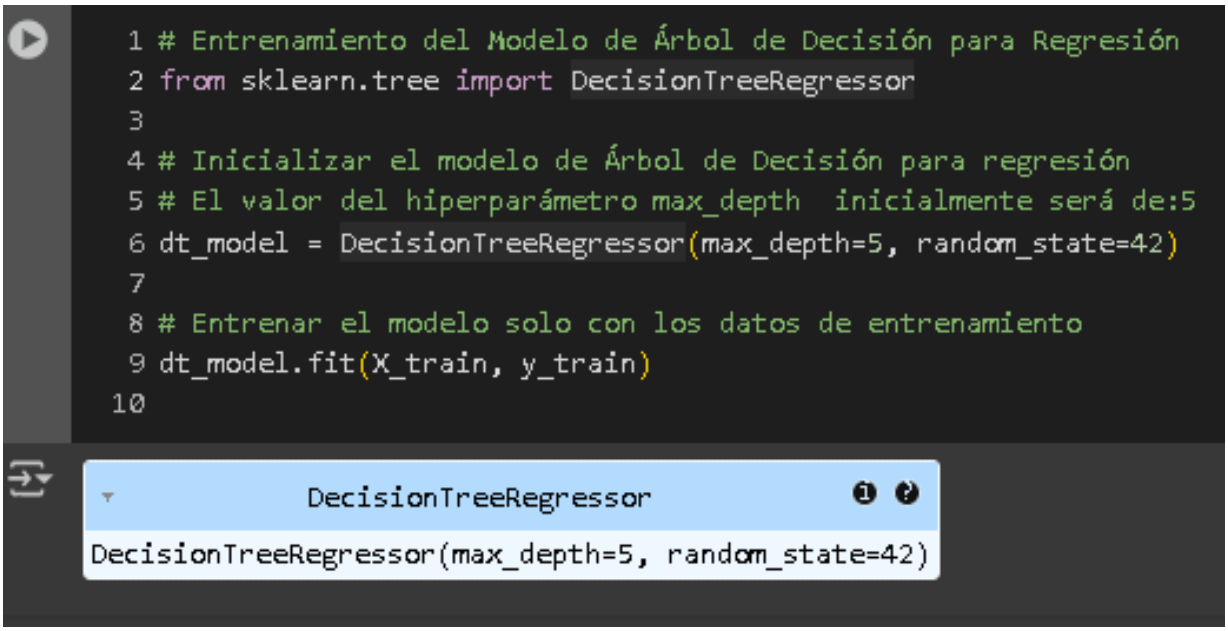
```
Resumen del Entrenamiento del Modelo K-NN para Regresión:
- Modelo: KNeighborsRegressor
- Número de Vecinos (n_neighbors): 5
- El modelo ha sido entrenado exitosamente con los datos escalados (X_scaled, y).
```

El entrenamiento inicial del modelo árboles de decisión se realizó empleando la clase DecisionTreeRegresso de la librería sklearn, digitando DecisionTreeRegressor(max_depth=5, random_state=42) y se guardó en la variable u objeto llamado “dt_model”, enseguida con la función .fit() de la clase DecisionTreeRegressor se entrenó el modelo con los datos guardados en las variables “X_train”, y “y_train”. El valor de random_state se estableció en 42 para controlar la aleatoriedad del proceso de construcción del árbol, el valor del hiperparámetro max_depth se definió inicialmente en 5, el cual establece el nivel de profundidad del árbol hasta 5 niveles máximo. El algoritmo examinó “X_train” y “y_train” para encontrar las mejores divisiones o

nodos que minimicen el error de predicción en cada paso, construyendo así el árbol hasta la profundidad máxima especificada.

Figura 8

Proceso de Entrenamiento para el Modelo de Árboles de Decisión



Después de realizar las acciones de preparación del conjunto de datos, éste quedó listo para realizar el proceso de búsqueda y definición de hiperparámetros óptimos para hacer predicciones.

Resultados Objetivo 2

Dado que para optimizar el rendimiento de los modelos se requiere establecer configuraciones e hiperparámetros propios de cada modelo, se procedió a buscar que ajustes y valores de los hiperparámetros entregaban mejores resultados según cada modelo.

Para el caso del modelo de regresión lineal, se tuvo en cuenta que este busca la única línea recta que minimiza la suma de los errores cuadrados entre los puntos de datos y la línea, arrojando una solución matemática directa y única para encontrar la pendiente y la intersección.

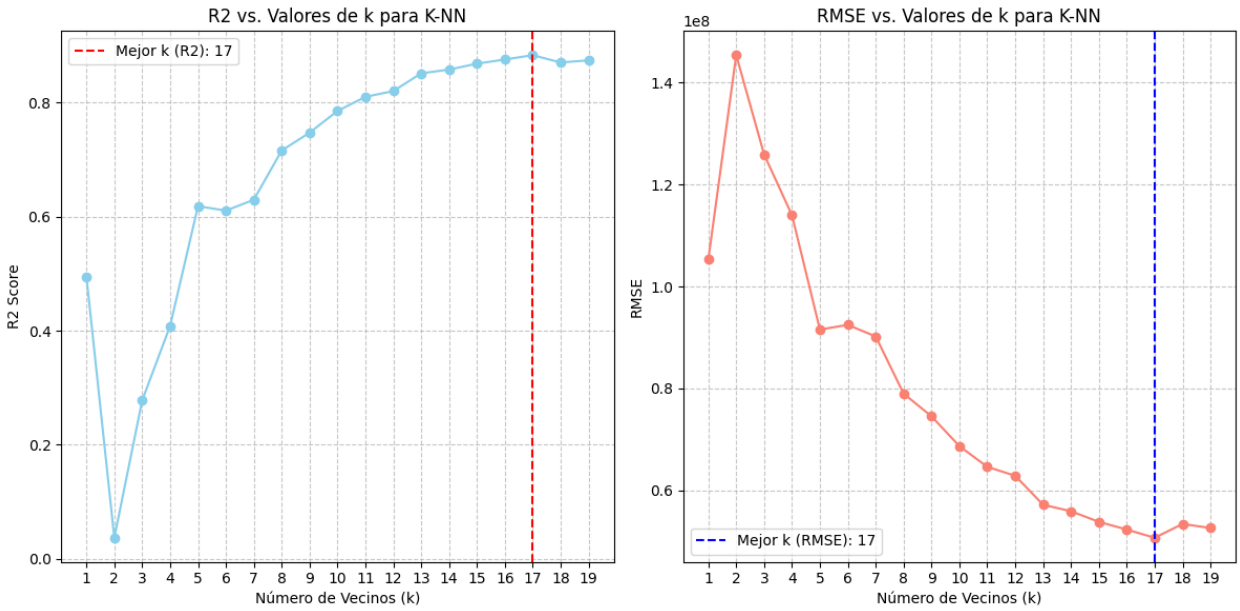
Por lo tanto, no se contó con configuración que el algoritmo tome de forma arbitraria influenciada con algún hiperparámetro.

Estimación de Hiperparámetros del Modelo K-NN

El modelo k-NN tienen como principal hiperparámetro el valor que se le da a k, cuando el modelo necesitó predecir la CANTIDAD_VOLUMEN_SUMINISTRADO para una semana futura, buscó las k semanas históricas en el conjunto de entrenamiento, que fueron las más similares a esa semana futura, basado en la variable SEMANA_INDEX y la distancia calculada. Luego, teniendo como configuración por defecto el hiperparámetro weights=’uniforrm’ tomó el promedio de los valores de CANTIDAD_VOLUMEN_SUMINISTRADO de esas k semanas más cercanas como la predicción. El hiperparámetro algorithm se configuró como 'auto' para seleccionar la mejor distancia entre los vecinos cercanos. Para encontrar cual es el valor de k optimo se empleó la técnica de Elbow Method o método gráfico del codo, que compara los valores de la suma de cuadrados en el eje Y, y diferentes valores de K en el eje X; para su definición se estableció un rango de valores de k de 1 a 10. Ver Figura 8.

Figura 9

Estimación del Hiperparámetro k en el Modelo K-NN



Según los resultados del método gráfico Elbow, se observó que para k iniciando en 2 se obtuvo un valor de R^2 de 0,03 y al variar k hasta 17 el valor de R^2 llegó a su valor máximo de 0,88, al dar a k un valor superior a 17, el valor de R^2 inició a descender; entonces se definió como valor óptimo de k el valor de 17, validado por la gráfica del RMSE para el mismo rango de k.

Estimación de Hiperparámetros del Modelo Árboles de Decisión

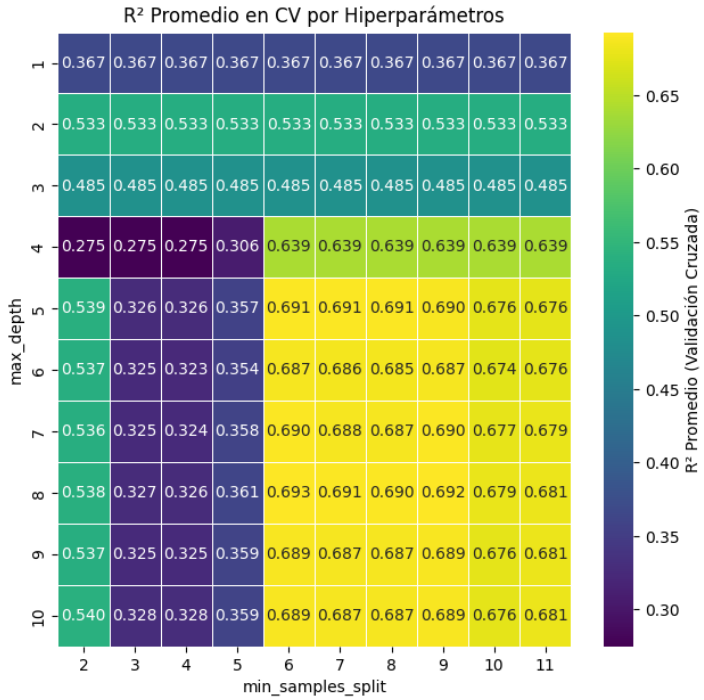
Para el modelo de árboles de decisión el valor óptimo de los hiperparámetros a menudo se encuentra mediante experimentación o técnicas como la validación cruzada, la cual en lugar de dividir el conjunto de datos una sola vez en entrenamiento y prueba, divide el conjunto de entrenamiento en “k” subconjuntos o “folds”, y promedia las métricas de rendimiento de todas las "k" iteraciones para obtener una estimación del rendimiento del modelo.

El modelo de árboles de decisión tiene como principales hiperparámetros “max_depth” y “min_samples_split”, en el caso de “max_depth”, este controla la longitud del camino más largo desde la raíz hasta una hoja. Al establecer “max_depth=x”, se le indica al modelo que no construya un árbol con más de x niveles de divisiones, controlando la complejidad del modelo y reduciendo el riesgo de sobreajuste. Un valor bajo crea un árbol simple y robusto. Un valor alto permite un árbol más complejo que puede capturar relaciones más intrincadas en los datos, pero corre el riesgo de sobreajuste. Y para el caso del “min_samples_split”, este determina cuántas muestras debe tener un nodo para que el árbol intente dividirlo, un valor bajo permite que el árbol haga muchas divisiones, pudiendo dar un sobreajuste si el árbol es muy profundo, y un valor alto restringe las divisiones generando un subajuste, de ahí la importancia de seleccionar los valores óptimos para estos hiperparámetros.

Para encontrar el valor óptimo de cada hipeparámetro, se empleó la técnica de la validación cruzada o Cross Validation(CV), buscando la combinación que produce el mejor equilibrio entre un árbol que sea lo suficientemente complejo para aprender de los datos, pero no tanto como para memorizarlos y perder capacidad de generalización. Como se observa en la Figura 10, esta comparó el rendimiento del modelo para diferentes valores de los hiperparámetros, en un rango de 1 a 10 para “max_depth” y de 2 a 12 para “min_samples_split”.

Figura 10

Estimación del Hiperparámetros del Modelo de Árboles de Decisión

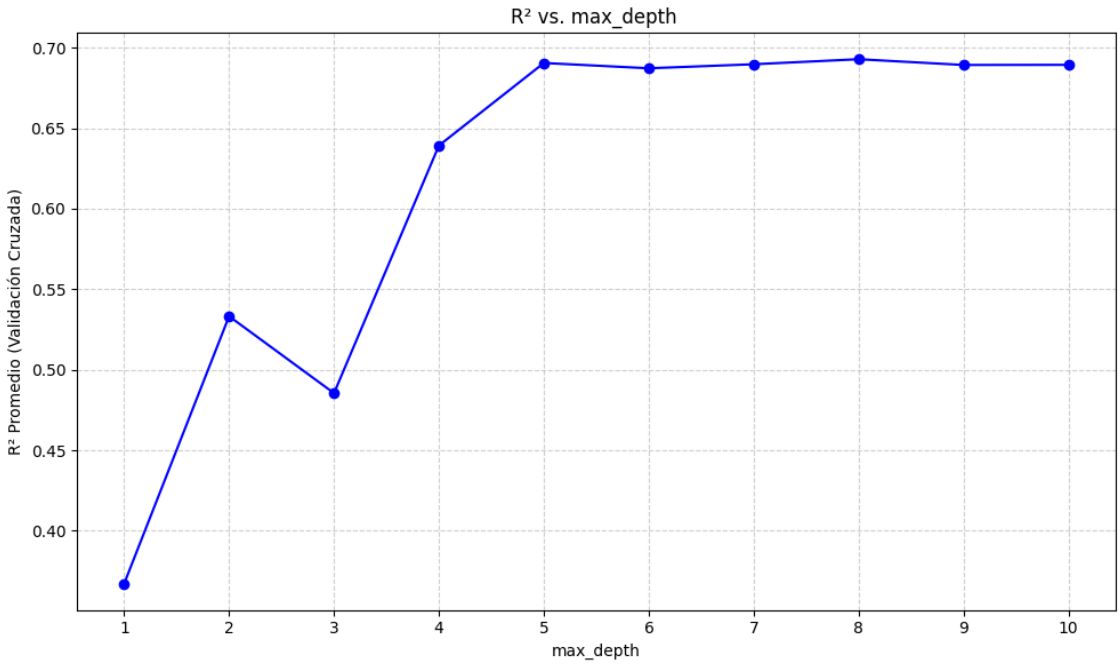


Al observar los resultados mostrados en la Figura 10, se identifica que, para una profundidad de 1, el R^2 es bajo, lo que indica que con una sola división no es suficiente para capturar la complejidad de los datos. Para un "max_depth" de 8, el R^2 alcanza su valor más alto con 0,689, en "min_samples_split" óptimo de 6, mostrando que una vez que el árbol tiene una profundidad de 8, el número mínimo de muestras para dividir un nodo ya no mejora significativamente el rendimiento del R^2 en el conjunto de datos dentro del rango explorado. A medida que "max_depth" aumenta más allá de 8, el R^2 comienza a disminuir, lo que permitió definir como valore óptimo de 8 para "max_depth" y 6 para "min_samples_split".

Considerando la influencia del parámetro “max_depth” en el rendimiento del modelo, se calculó el R^2 y el RMSE del modelo para diferentes valores en un rango de 1 a 10 manteniendo el “min_samples_split” óptimo definido. Ver Figura 11.

Figura 11

Rendimiento del Modelo de Árboles de Decisión



De la gráfica anterior es posible confirmar que el valor de “max_depth” que mejor rendimiento da al modelo es 8, ya que partiendo de 1, el R^2 fue menor a 0,4, y al llegar a 8 arrojó el mejor resultado, y al incrementar su valor, el rendimiento disminuyó y varió sin ser mejor que el obtenido en 8.

Luego de realizar la búsqueda de los valores óptimos de los hiperámetros más influyentes en cada modelo, se obtuvo la configuración óptima de los algoritmos, los cuales quedaron listos para realizar predicciones de la mejor manera según su naturaleza y desarrollo original.

Resultados Objetivo 3

Para evaluar la precisión de los modelos de machine learning trabajados se empleó las métricas de medición de MAE, MSE, R^2 y gráficas de comparación entre valores reales y valores predichos.

Evaluación del Modelo Regresión Lineal

Para el modelo de regresión lineal, basado en que la variable "model" contiene el modelo entrenado con los datos apartados para ello, se procedió a realizar las predicciones, allí se creó la variable "y_pred" para almacenar los valores a predecir empleando la función .predict() la cual tomó las nuevas entradas de datos y, basándose en lo que el modelo aprendió durante el entrenamiento, hizo las predicciones correspondientes, todo esto sobre los datos de entrenamiento almacenados en la variable "X_train" que utilizó para aprender. En este paso el modelo se ajustó a los datos que ya había visto, posteriormente con los valores reales de la variable dependiente "y_train" y las predicciones del modelo "y_pred", calculó el coeficiente de determinación R^2 y el MSE.

Figura 12

Proceso de Evaluación del Modelo de Regresión Lineal

```python
# Evaluacion Modelo Regresion Lineal
from sklearn.metrics import r2_score, mean_squared_error, mean_absolute_error
import numpy as np # Import numpy for sqrt

# Realizar Predicciones sobre los datos de entrenamiento para la evaluación
y_pred_train = model.predict(X_train)
# Realizar Predicciones sobre los datos de prueba para la evaluación
y_pred_test = model.predict(X_test)

# Cálculo de las Métricas de Evaluación para el conjunto de entrenamiento
# Coeficiente de Determinación (R²)
r_squared_train = r2_score(y_train, y_pred_train)

# Error Cuadrático Medio (MSE)
mse_train = mean_squared_error(y_train, y_pred_train)

# Raíz del Error Cuadrático Medio (RMSE)
rmse_train = np.sqrt(mse_train)

# Calcular MAE (Mean Absolute Error)
mae_train = mean_absolute_error(y_train, y_pred_train)
```

Como resultado de la evaluación del modelo de regresión lineal se obtuvo los siguientes resultados:

Tabla 4

Resultados de la Evaluación del Modelo de Regresión Lineal

Métrica de Evaluación	Resultado Obtenido
R^2	0,0196
MSE	39168458341035600
RMSE	197910227,98
MAE	145053532,20

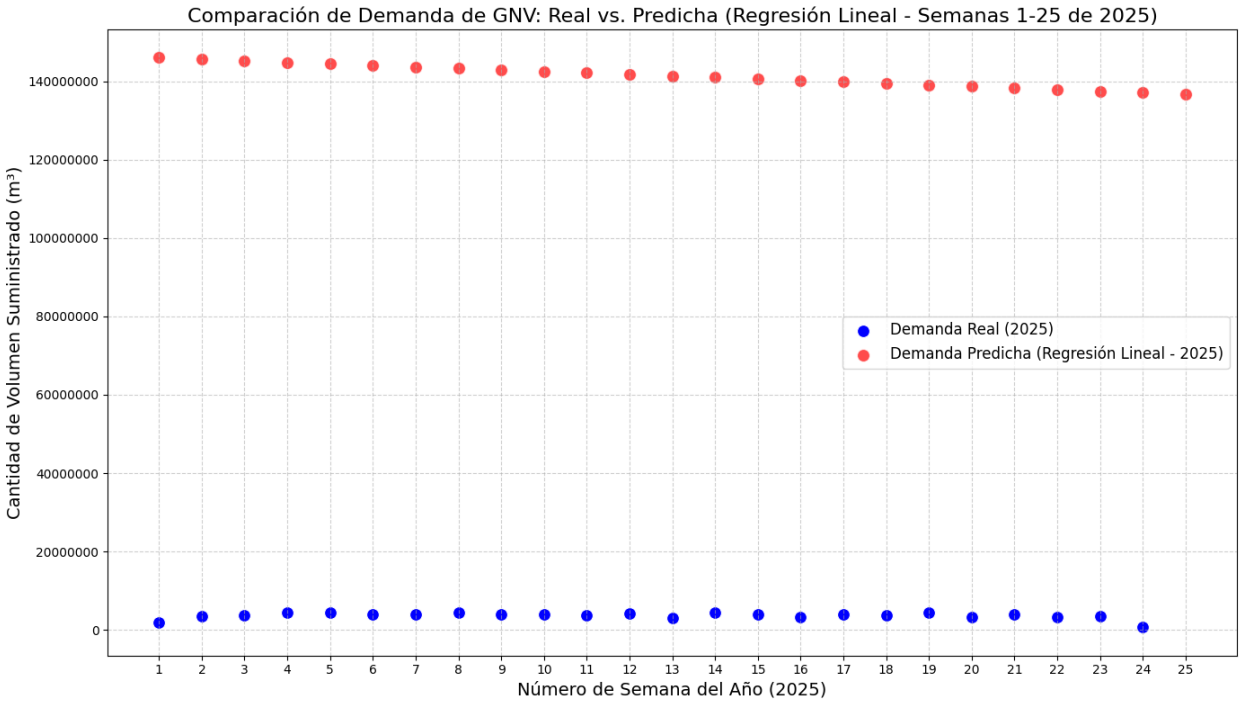
Nota. Valores obtenidos de las métricas de evaluación del modelo de regresión lineal.

Estos resultados muestran que según el valor de R^2, el modelo de regresión lineal solo explica aproximadamente el 1,91% de la variabilidad en CANTIDAD_VOLUMEN _SUMINISTRADO basándose en la variable "SEMANA_INDEX", indicando que el modelo no capturó de manera efectiva los patrones que influyen en el volumen de combustible suministrado. Los valores del MAE, MSE y el RMSE confirman que el modelo hace predicciones que se desvían de los valores reales, demostrando que, dada la variabilidad en el volumen de combustible suministrado por semanas, la relación que hay entre las variables estudiadas no es lineal.

Para observar gráficamente la precisión de las predicciones hechas por el modelo de regresión lineal, se procedió a tomar los datos originales del volumen suministrado entre las semanas 1 y 25 del 2025, y se graficó junto con los datos que el modelo entrenado predijo para este mismo rango de fechas. Así, con la función scatterplot() se elaboró un gráfico de dispersión usando la librería Seaborn que está construida sobre Matplotlib. Ver Figura 13.

Figura 13

Comparación Gráfica de Predicciones del Modelo de Regresión Lineal



La gráfica de dispersión muestra dos series de puntos, la primera con puntos azules que representan la demanda real de la cantidad de volumen suministrado para las semanas 1 a 25 del año 2025, y la segunda con puntos rojos que representan los valores de la demanda predicha por el modelo de regresión lineal para el mismo rango de semanas. Se observa una gran diferencia entre los valores reales y los predichos. Los valores de demanda real que se mantienen en un rango de valores entre 1.5 y 4.5 millones, mientras que los valores de la demanda predicha se sitúan en un rango de valores mucho más altos, por encima de los 140 millones de metros cúbicos. Esto confirma que el modelo hace una predicción de forma lineal en estos datos con una variabilidad considerable, y se ajusta al comportamiento real de estos, dejando claro que su precisión fue muy baja.

Evaluación del Modelo K-NN

El entrenamiento definitivo del modelo k-Nearest Neighbors se realizó con el valor de k igual a 17 el cual fue el óptimo hallado. Conjuntamente se empleó la clase KNeighborsRegressor() de la biblioteca Scikit-learn, y la función .fit() con los datos guardados en las variables “X_scaled” y “y”.

Las predicciones se realizaron empleando la función .predict(), la cual tomó las nuevas entradas de datos y basándose en lo que el modelo aprendió durante el entrenamiento, hizo las predicciones correspondientes, todo esto sobre los mismos datos de entrenamiento almacenados en la variable “X_scaled” que utilizó para aprender. Allí la función usó la variable “knn_model” que contiene el modelo entrenado con el algoritmo k-NN, y los guardó en la variable “y_pred_knn”, la cual se usó para almacenar los valores predichos. Posteriormente con los valores reales de la variable “y” que contiene los volúmenes de GNV suministrados históricos y “y_pred_knn” que son los valores predichos por el modelo, se calculó el coeficiente de determinación R^2 y el MSE. Ver Figura 14.

Figura 14

Proceso de Evaluación del Modelo de K-Nearest Neighbors (K-NN)

```python
# Evaluación de Métricas del modelo de k-NN

from sklearn.neighbors import KNeighborsRegressor
from sklearn.metrics import r2_score, mean_squared_error, mean_absolute_error

# usamos el valor optimo de k
knn_model = KNeighborsRegressor(n_neighbors=17)
knn_model.fit(X_train_scaled, y_train)
y_pred_knn = knn_model.predict(X_test_scaled)

# Calcular R2
r2 = r2_score(y_test, y_pred_knn)

# Calcular MSE (Mean Squared Error)
mse = mean_squared_error(y_test, y_pred_knn)

# Calcular RMSE (Root Mean Squared Error)
rmse = np.sqrt(mse)

# Calcular MAE (Mean Absolute Error)
mae = mean_absolute_error(y_test, y_pred_knn)
```

Como resultado de la evaluación del modelo de k-NN se obtuvo los siguientes resultados:

Tabla 5

Resultados de la Evaluación del Modelo K-NN

Métrica de Evaluación	Resultado Obtenido
R^2	0,8831
MSE	2567433026128827
RMSE	50669843,36
MAE	33951088,04

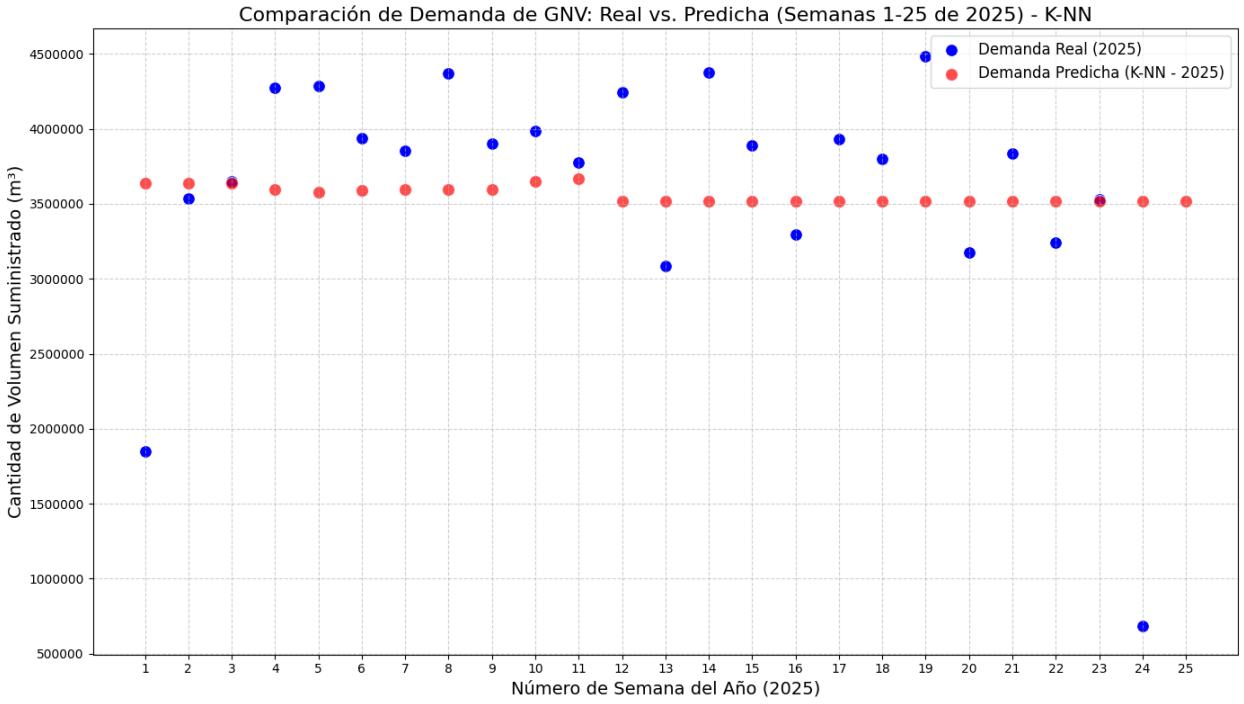
Nota. Valores obtenidos de las métricas de evaluación del modelo k-NN.

Estos resultados muestran que el 88,31% de la variabilidad en el volumen de GNV suministrado es bien expuesto por el modelo según la variable de tiempo en semanas dada. Se observa que el modelo K-NN es capaz de capturar una gran parte de los patrones y tendencias en los datos históricos de ventas de GNV, todo esto confirmado por los valores del MAE, MSE y el RMSE aceptable considerando el rango de la variable objetivo.

Para observar gráficamente la precisión de las predicciones hechas por el modelo k-NN, se procedió a tomar los datos originales del volumen suministrado entre las semanas 1 y 25 del 2025, y se graficó junto con los datos que el modelo entrenado predijo para este mismo rango de fechas. Así, con la función scatterplot() se elaboró un gráfico de dispersión usando la librería Seaborn. Ver Figura 15.

Figura 15

Comparación Gráfica de las Predicciones del Modelo K-NN



La gráfica de dispersión muestra dos series de puntos, la primera con puntos azules que representan la demanda real de la cantidad de volumen suministrado para las semanas 1 a 25 del año 2025, y la segunda con puntos rojos que representan los valores de la demanda predicha por el modelo k-NN para el mismo rango de semanas. Se observa que, en muchas semanas, las predicciones están muy cerca de los valores reales, lo que indica que el modelo captura bien la tendencia general de la demanda, esto evidenció que el modelo tiene un buen desempeño, y realiza predicciones con una precisión buena.

Evaluación del Modelo Árboles de Decisión

El entrenamiento definitivo del modelo de árboles de decisión se realizó utilizando los valores óptimos encontrados, max_depth = 8 y min_samples_split = 6. Conjuntamente se empleó la clase DecisionTreeRegressor de la librería sklearn.tree de scikit-learn, y la función .fit(), el cual al ser ejecutado usa el algoritmo del árbol de decisión y comienza a construir su estructura de nodos, ramas y hojas, de manera iterativa; este buscó la mejor manera de dividir los datos en cada nodo para reducir la varianza y hacer que las predicciones en los nodos hoja sean lo más precisas posible. Esta función se ejecutó con los datos guardados en las variables “X_train” y “y_train” que contienen la parte de los datos definida de cada variable para entrenar al modelo, y su resultado se almacenó en la variable “dt_model”.

Las predicciones se realizaron empleando la función .predict() la cual tomó los datos que no había visto antes almacenados en la variable “X_test”, basándose en lo que el modelo “dt_model” aprendió durante el entrenamiento, y el resultado de estas predicciones se guardó en la variable “y_pred_dt_test”.

Posteriormente con los valores de la variable “y_test” que contiene los valores observados para la variable objetivo y la variable “y_pred_dt_test”, se calculó el coeficiente de determinación R^2 y el MSE.

Figura 16

Proceso de Evaluación del Modelo de Árboles de Decisión

```
# Entrenamiento y Evaluacion del modelo Arboles de decision
from sklearn.tree import DecisionTreeRegressor
from sklearn.metrics import r2_score, mean_squared_error

# Entrenamiento del Modelo de Árbol de Decisión con los datos de entrenamiento
dt_model = DecisionTreeRegressor(max_depth=8, random_state=42, min_samples_split=6)
dt_model.fit(X_train, y_train)

# Usar el modelo entrenado para predecir los valores en el conjunto de prueba
y_pred_dt_test = dt_model.predict(X_test)

# Cálculo de Métricas de Evaluación para el Árbol de Decisión en el Conjunto de Prueba

# R² (Coeficiente de Determinación)
r2_dt_test = r2_score(y_test, y_pred_dt_test)

# MSE (Error Cuadrático Medio)
mse_dt_test = mean_squared_error(y_test, y_pred_dt_test)

# RMSE (Raíz del Error Cuadrático Medio)
rmse_dt_test = np.sqrt(mse_dt_test)

# Calcular MAE (Mean Absolute Error)
mae = mean_absolute_error(y_test, y_pred_dt_test)
```

Como resultado de la evaluación del modelo de árboles de decisión se obtuvo los siguientes resultados:

Tabla 6

Resultados de la Evaluación del Modelo Árboles de Decisión

Métrica de Evaluación	Resultado Obtenido
R^2	0,5555
MSE	9760370388537516
RMSE	98794586,84
MAE	33939645,99

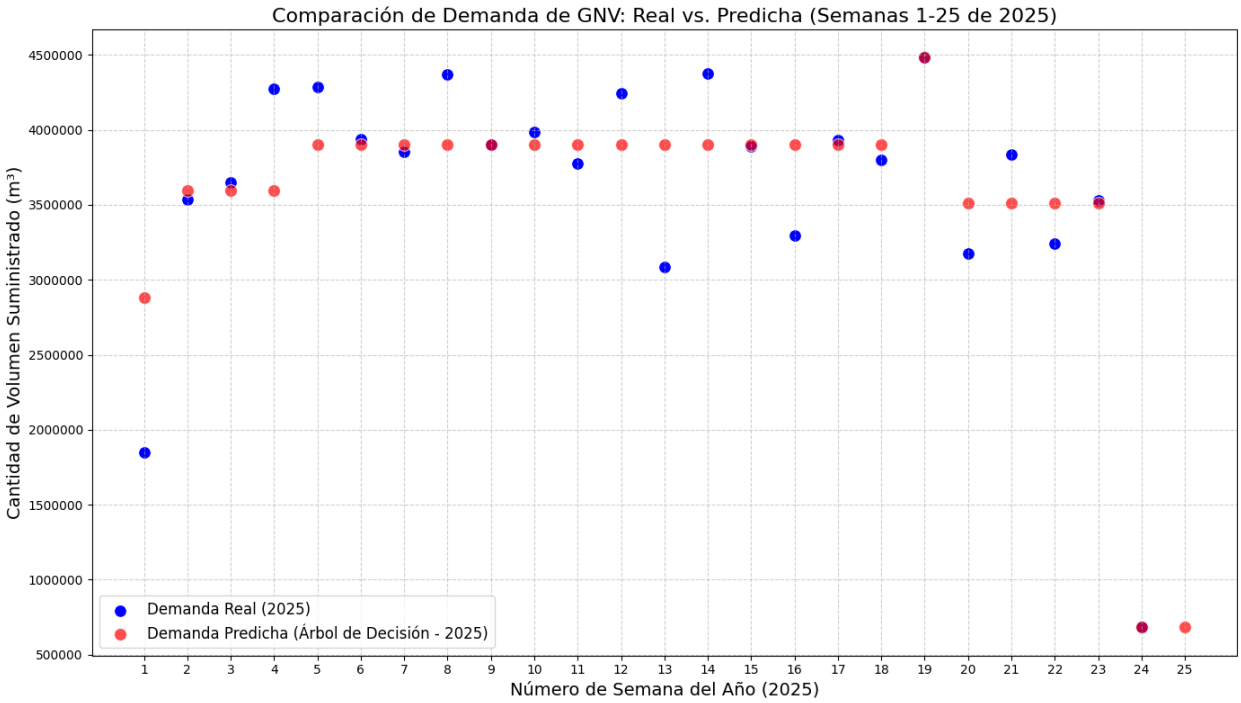
Nota. Valores obtenidos de las métricas de evaluación del modelo árboles de decisión.

Estos resultados muestran que según el valor de R^2, el modelo de regresión lineal explica aproximadamente el 55,5% de la variabilidad en cantidad de volumen de GNV suministrado basándose en la variable "SEMANA_INDEX", indicando que el modelo es capaz de capturar una porción significativa de la tendencia en el volumen de ventas semanales. Los valores del MAE, MSE y el RMSE confirman que el modelo hace predicciones que se desvían moderadamente de los valores reales, aunque el modelo capta la tendencia, las predicciones individuales tienen desviaciones considerables.

Para observar gráficamente la precisión de las predicciones hechas por el modelo árboles de decisión, se procedió a tomar los datos originales del volumen suministrado entre las semanas 1 y 25 del 2025, y se graficaron junto con los datos que el modelo entrenado predijo para este mismo rango de fechas. Así, con la función scatterplot() se elaboró un gráfico de dispersión usando la librería Seaborn. Ver Figura 17.

Figura 17

Comparación Gráfica de las Predicciones del Modelo Árboles de Decisión



La gráfica de dispersión muestra dos series de puntos, la primera con puntos azules que representan la demanda real de la cantidad de volumen suministrado para las semanas 1 a 25 del año 2025, y la segunda con puntos rojos que representan los valores de la demanda predicha por el modelo árboles de decisión para el mismo rango de semanas. Se observa que el modelo no captura las fluctuaciones semanales y por ello las predicciones no siguen bien la variabilidad de los datos reales, algunos valores predichos aparecen muy cerca de los reales, pero muchas se desvían, lo que evidencia que el modelo tiene una precisión no tan buena.

Luego de realizar las predicciones y evaluación de cada modelo objeto de este estudio, se observó que el modelo que mejor se ajustó a la naturaleza de los datos y realizó predicciones más acertadas fue el de k-Nearest Neighbors (k-NN), el cual mostró un buen rendimiento por encima

de los modelos de árboles de decisión que tubo un rendimiento medianamente bueno, y el modelo de regresión lineal obtuvo un rendimiento muy bajo.

Conclusiones

La limpieza y preprocesamiento de los datos históricos de ventas de GNV, incluyendo la creación de variables como SEMANA_ANIO, ANIO, y SEMANA_INDEX, fue fundamental para asegurar la calidad y consistencia del dataset, lo que permitió un adecuado desempeño de los algoritmos de machine learning.

La aplicación de la metodología SAMPLE, que incluyó la división de datos en conjuntos de entrenamiento (70%) y prueba (30%), permitió una evaluación objetiva del comportamiento de los modelos frente a datos no vistos, asegurando la validez de los resultados obtenidos.

La exploración de diferentes configuraciones de hiperparámetros para k-NN y Árboles de Decisión junto con el escalado de la variable independiente SEMANA_INDEX para k-NN, fue un paso importante para lograr obtener un buen rendimiento y precisión de cada modelo.

La aplicación de modelos de machine learning para la predicción de la demanda de GNV demuestra su utilidad como herramienta estratégica para la planificación y gestión de recursos, permitiendo aportar a las autoridades y empresas para la toma decisiones informadas en pro de optimizar la distribución y el suministro de GNV, y así contribuir a la transición energética del país.

La evaluación de los modelos mediante métricas como MAE, MSE y R^2 permitió una comprensión integral de su precisión. Estas son un fundamento clave en la selección del modelo más eficiente para hacer predicciones, con el apoyo de la visualización comparativa de los valores reales versus los predichos se logró concluir cual modelo fue el mejor.

El modelo de regresión lineal es fácil de utilizar, pero demostró que, por su predicción de forma lineal, que no se ajustó a la variabilidad considerable que tienen los datos, y entregó una precisión muy baja.

El modelo de árboles de decisión, a pesar de sus ventajas en interpretabilidad, mostró limitaciones para capturar las fluctuaciones en la demanda real de GNV, indicando que, si bien puede identificar tendencias, la precisión en predicciones individuales podría ser un desafío debido a desviaciones considerables entre valores predichos y reales.

El modelo k-Nearest Neighbors fue el que mejor se ajustó a la naturaleza de los datos, mostrando que reconoció relativamente bien la tendencia general de la demanda de GNV, obteniendo el mejor resultado en las métricas de evaluación entre los tres modelos estudiados.

Recomendaciones

Para la obtención de resultados más precisos en la demanda del GNV, se podría integrar variables externas más dinámicas y predictivas, como pronósticos climáticos a corto y mediano plazo, precios futuros de combustibles líquidos, eventos económicos o sociales que puedan influir en la movilidad, y campañas de promoción del GNV, lo que mejoraría significativamente la capacidad de pronóstico.

Para mejor la precisión del modelo de k-NN se sugiere establecer un proceso de optimización continuo de hiperparámetros y reentrenamiento periódico de los modelos, con el fin de mantener su relevancia y precisión a medida que cambian las condiciones del mercado y se acumulan nuevos datos.

Realizar un análisis de sensibilidad para entender cómo pequeñas variaciones en los datos de entrada o en los parámetros del modelo afectan las predicciones. Esto permitirá evaluar la robustez del modelo y comprender mejor los rangos de confianza de las proyecciones de demanda.

Para tomar las predicciones del modelo como fundamento para la toma de decisiones se recomienda desarrollar un sistema de monitoreo en tiempo real que compare las predicciones del modelo con la demanda real observada. Este sistema permitiría identificar rápidamente desviaciones significativas, activar alertas y facilitar el ajuste oportuno de los modelos y la estrategia en ejecución.

Dada la complejidad de la demanda de GNV y las limitaciones observadas en modelos incluidos en este estudio, se recomienda explorar la implementación de modelos híbridos o métodos de ensamble, que podrían combinar las fortalezas de diferentes algoritmos para mejorar la precisión y robustez de las predicciones.

Fomentar la colaboración entre entidades gubernamentales como el Ministerio de Minas y Energía, CREG, Runt, Ministerio de Ambiente, empresas distribuidoras de GNV y centros de investigación para compartir datos, validar modelos y coordinar estrategias de expansión de infraestructura y políticas de fomento del GNV, asegurando una visión integral y un impacto sinérgico en pro de medio ambiente y la economía.

Referencias Bibliográficas

Bruce, P. C., Bruce, A., & Gedeck, P. (2022). Estadística práctica para ciencia de datos con R y Python.

Castro Zuluaga, C. A. (2020). Planeación de la producción. Colombia: Universidad EAFIT.

Contreras, L., Tarazona Bermúdez, G., & Alemán Cardona, A. P. (2023). Machine Learning aplicado al rendimiento académico en educación superior: factores, variables y herramientas. Google Books https://www.google.com.pe/books/edition/Machine_Learning_aplicado_al_rendimiento/J_XBEAAAQBAJ?hl=es-419&gbpv=0

Cruz Tapia, J. C. (2024). 100 preguntas para entender sobre inversiones (Ecoe Ediciones, Ed.; 1a ed.).

Cuevas, Erik., Avalos, Omar., Emanuel, Primitivo., Valdivia, Arturo., & Pérez, M. Antonio. (2021). Introducción al machine learning con MATLAB.

Daniel, E., & Porras, S. (2023). Un método para la asignación de cupos de crédito de entidades del sector financiero colombiano empleando técnicas de machine learning.

Ferro Veiga, J. M. (2019). Generación Terrorismo Medioambiental (I. Lulu Press, Ed.).

González-De León, M. A. & Di Scipio-Cimetta, S. (2022). The role of natural gas in today's energy transition. DYNA, 89(221), 92–100. https://doi.org/10.15446/dyna.v89n221.99347

Guzmán, C. (2024, 11 de septiembre). Gasolina a precio justo. Periódico La República. https://www.larepublica.co/analisis/camilo-guzman-3193497/gasolina-a-precio-justo-3949878

Hossain, M., Ho, R. C., & Trajkovski, G. (Eds.). (2023). Handbook of Research on AI and Machine Learning Applications in Customer Support and Analytics. IGI Global Scientific Publishing. https://doi.org/10.4018/978-1-6684-7105-0

Kane, F. (2017). Hands-On Data Science and Python Machine Learning.

Ley 99 de 1993 (1993) Ley General Ambiental de Colombia Gestor Normativo. (s. f.). Función Pública.

https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=297

Ley 142 de 1994 (1994) Régimen de los Servicios Públicos Domiciliarios Gestor Normativo. (s. f.). Función Pública. https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=2752

Ministerio de Ambiente y Desarrollo Sostenible (2015). Decreto 1076 de 2015 Por medio del cual se expide el Decreto Único Reglamentario del Sector Ambiente y Desarrollo Sostenible Bogotá D.C.: Ministerio de Ambiente y Desarrollo Sostenible https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=78153

Ministerio de Minas y Energía de Colombia (2015). Decreto 1073 de 2015 Por la cual medio del cual se expide el Decreto Único Reglamentario del Sector Administrativo de Minas y Energía. Bogotá D.C.: Ministerio de Minas y Energía de Colombia https://www.funcionpublica.gov.co/eva/gestornormativo/norma.php?i=77887

Morales, A. C., Adrián, C., Flórez, C., Andrés, M., Orozco, P., Herrera, B., Paula, J., & Sánchez García, L. (2025, 10 de abril de 2025). Proyección de Demanda de Combustibles Líquidos 2024-2040 Unidad de Planeación Minero-Energética UPME https://docs.upme.gov.co/DemandayEficiencia/Documents/UPME_Proyecciones_demanda_comb_liquidos_2024-2040_Para_comentarios_4-4-2025.pdf

Nivia Gil, J. A. (2024). Lecciones básicas de economía.

Observatorio Ambiental de Bogotá. (2017, 20 de noviembre). El material particulado en un carro a gas es 95 % menor que a diésel. Observatorio Ambiental de Bogotá. https://oab.ambientebogota.gov.co/el-material-particulado-en-un-carro-a-gas-es-95-menor-que-a-diesel/

Parker, A. (2021). Contaminación del aire por la industria. España: Reverte Google Books https://www.google.com.co/books/edition/Contaminaci%C3%B3n_del_aire_por_la_industria/VdMfEAAAQBAJ?hl=es-419&gbpv=0

Prathmesh Yelne. (2023). Machine Learning & AI (Codegyan, Ed.). codegyan.in. Google Books https://www.google.com.co/books/edition/Machine_Learning_AI/quXAEAAAQBAJ?hl=es-419&gbpv=1

Raschka Sebastian. (2023). Machine Learning con PyTorch y Scikit-Learn.(Marcombo. Ed). Google Books https://www.google.com.co/books/edition/Machine_Learning_con_PyTorch_y_Scikit_Le/NumwEAAAQBAJ?hl=es-419&gbpv=0

Raschka, Sebastian., & Mirajalili, Vahid. (2018). Python machine learning : machine learning and deep learning with Python, scikit-learn, and TensorFlow. Packt Publishing.

Registro Único Nacional de Tránsito. (2023). Cifras RUNT. Registro Único Nacional de Tránsito, 1. https://www.runt.gov.co/sites/default/files/CIFRAS%20RUNT%20%281%29.pdf

Reza, N., & Kamrouz, B. (2024). Estadísticas para Principiantes (Mathlibros.com, Ed.).

Sharma, V., Cali, Ü., Sardana, B., Kuzlu, M., Banga, D., & Pipattanasomporn, M. (2021). Data-driven short-term natural gas demand forecasting with machine learning techniques.

Journal of Petroleum Science and Engineering, 206. https://doi.org/10.1016/j.petrol.2021.108979

Tito, A. E. A., Condori, B. O. H., & Vera, Y. P. (2023). Comparative analysis of Machine Learning Techniques for the prediction of cases of university dropout. RISTI - Revista Iberica de Sistemas e Tecnologias de Informacao, 2023(e51), 84–98. https://doi.org/10.17013/risti.51.84-98

Velasco Rebolledo, Jacinto. (2024). Machine Learning: Fundamentos, Algoritmos y Aplicaciones para los Negocios, Industria y Finanzas.

Apéndices

Apéndice A

Código Desarrollado para el Proyecto

A continuación, se presenta en enlace en donde se encuentra ubicado el código ejecutado para el desarrollo de este proyecto aplicado:https://drive.google.com/drive/folders/1kIiFZDzFChU-_qbXb9lZXeZuN1u8nSzz?usp=drive_link