MODELO DE CLASIFICACIÓN PARA PREDECIR TIPO DE COMBUSTIBLE DE VEHÍCULOS

Integrantes:

Tomás Gómez Cardona

Juan Pablo Chalarca Jaramillo

Juan Pablo Barón Vera

Juan José Echeverri Rodriguez

Bootcamp Análisis de datos - Avanzado

2025

1. Introducción

La transición hacia la movilidad sostenible en Colombia enfrenta retos importantes, tanto sociales como económicos, que condicionan la adopción de vehículos eléctricos e híbridos. Factores como el alto costo de adquisición, la limitada infraestructura de carga en regiones fuera de las principales ciudades y la percepción del consumidor sobre autonomía y confiabilidad han generado una adopción desigual de estas tecnologías en el país. Esta situación refleja un mercado fragmentado, donde las barreras de acceso y las diferencias regionales limitan la masificación de los vehículos electrificados, a pesar de los esfuerzos en incentivos gubernamentales y campañas de concienciación ambiental.

En este contexto, el presente proyecto tiene como objetivo analizar y predecir la adopción de vehículos electrificados a nivel departamental en Colombia, mediante el desarrollo de modelos de clasificación que permitan identificar el tipo de combustible utilizado en los registros vehiculares (eléctrico, híbrido gasolina, híbrido diésel). Para ello, se trabaja con un conjunto de datos que combina variables categóricas y numéricas, evaluando su significancia y aplicando técnicas de codificación y preprocesamiento. El uso de modelos predictivos permite detectar patrones, evaluar la influencia de factores regionales y técnicos, y generar insumos que faciliten la toma de decisiones tanto para autoridades públicas como para actores del sector automotor en el marco de la transición energética.

2. Desarrollo del proyecto

2.1. Objetivo General

Analizar y predecir la adopción de vehículos electrificados a nivel departamental en Colombia y desarrollar un modelo de clasificación que, a partir de variables categóricas y numéricas, prediga el tipo de combustible (eléctrico, híbrido gasolina, híbrido diésel), con el fin de identificar patrones y tendencias en los registros vehiculares que orienten la toma de decisiones.

2.2. Objetivos específicos

2.2.1. Procesar y depurar los datos del dataset, identificando y tratando valores faltantes mediante imputación o eliminación según su relevancia, además de seleccionar las variables más significativas

para evitar redundancia y alta correlación, garantizando así la calidad de la información para el modelado.

2.2.2. Codificar adecuadamente las variables categóricas aplicando Label Encoding para algoritmos basados en árboles y One-Hot Encoding para modelos lineales o de distancia, con el fin de transformar los datos en un formato óptimo que preserve la información y facilite el aprendizaje automático.

2.2.3. Desarrollar, entrenar y evaluar un modelo de clasificación que utilice tanto variables categóricas como numéricas para predecir el tipo de combustible de los vehículos, analizando el desempeño del modelo en términos de precisión, robustez y capacidad de generalización a nuevos registros.

2.3. Descripción del Proyecto

2.3.1. Identificación de la problemática

En Colombia, la adopción de vehículos eléctricos e híbridos enfrenta múltiples desafíos que afectan tanto la demanda como la disponibilidad de estos vehículos. La transición hacia transporte más limpio no solo depende de factores tecnológicos, sino también de condiciones sociales, económicas y geográficas que influyen directamente en la decisión de compra.

El alto costo de adquisición de vehículos eléctricos e híbridos representa una barrera importante para gran parte de la población. Sectores con ingresos bajos y medios tienen acceso limitado a estas tecnologías, lo que genera una concentración de la demanda en segmentos de mayor poder adquisitivo. Además, los incentivos gubernamentales y subsidios son todavía insuficientes o focalizados, lo que limita la penetración del mercado.

La percepción del consumidor respecto a la confiabilidad, mantenimiento y autonomía de los vehículos eléctricos influye significativamente en la decisión de compra. Muchos usuarios aún presentan reticencia hacia nuevas tecnologías o desconocen los beneficios económicos y ambientales de estos vehículos.

La infraestructura de recarga es desigual en el territorio colombiano. Ciudades principales como Bogotá, Medellín y Cali

cuentan con más estaciones de carga, mientras que zonas rurales o regiones periféricas presentan escasa o nula disponibilidad, limitando la viabilidad de la adopción. La geografía del país, con zonas montañosas y de alta altitud, también influye en la autonomía y rendimiento de los vehículos eléctricos.

La combinación de bajos ingresos, infraestructura desigual, altos costos de adquisición y percepción social limitada genera un mercado fragmentado y desbalanceado. Esto dificulta predecir la demanda de vehículos eléctricos e híbridos por región y por segmento socioeconómico, lo que representa un desafío para fabricantes, distribuidores y planificadores urbanos que buscan promover la movilidad sostenible.

El uso de modelos predictivos y análisis avanzado de datos puede ayudar a identificar patrones de adopción, segmentar el mercado, evaluar la influencia de variables socioeconómicas y regionales, y optimizar estrategias de incentivo y distribución de infraestructura. Esto permitirá generar políticas más efectivas y decisiones empresariales fundamentadas, orientadas a impulsar la transición hacia vehículos eléctricos e híbridos en Colombia.

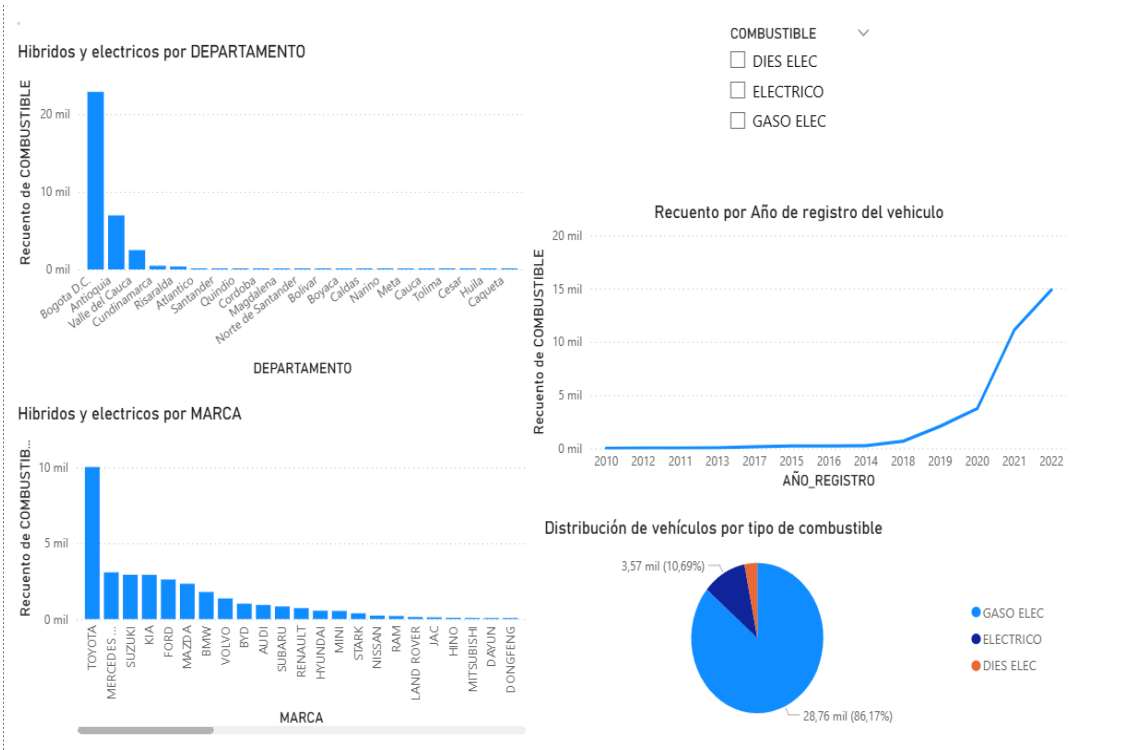
2.3.2. Limpieza de la base de datos y Análisis descriptivo

Antes de proceder con la visualización de los valores faltantes o iguales a cero en algunos registros de la base de datos, se decidió excluir ciertas variables que, por su naturaleza, no aportan información relevante para la predicción. Esta selección se realizó en conjunto con nuestros docentes, considerando el contexto del problema. Las variables excluidas fueron:

- AÑO_REGISTRO y FECHA_REGISTRO: No son necesarias para el entrenamiento de los modelos de clasificación, ya que corresponden únicamente a datos temporales.
- MARCA y LÍNEA: Ambas variables son multiclase con una cardinalidad muy alta (más de 100 clases en MARCA y más de 400 en LÍNEA). Transformarlas a variables dummies generaría más de 500 columnas adicionales, lo cual resultaría poco eficiente y contraproducente para el entrenamiento de los modelos.

- MODALIDAD: Es propia de vehículos grandes, lo que podría generar asociación con otras variables del conjunto de datos. Además, presenta una gran cantidad de valores faltantes.
- CAPACIDAD_CARGA y CAPACIDAD_PASAJEROS: Exclusivas de buses y camiones, con numerosos registros incompletos, razón por la cual se excluyen.
- CANTIDAD: Representa el número de vehículos registrados, pero en todos los casos su valor es igual a 1, por lo tanto carece de utilidad.
- ORGANISMO_TRANSITO y MUNICIPIO: Ambas variables presentan redundancia, ya que están directamente relacionadas entre sí y con la variable DEPARTAMENTO.
- CILINDRAJE: Genera fuga de información, pues los vehículos eléctricos tienen un valor de 0 en esta variable, lo que permitiría al modelo clasificarlos de manera trivial. Por esta razón se omite.

Ya con el dataset preparado, empezamos a borrar lo registros con variables nulo e iguales a cero para la limpieza



📊 Conclusiones

1. La **adopción de híbridos y eléctricos está concentrada en Bogotá y grandes ciudades**.

2. Existe un **crecimiento exponencial a partir de 2019**, lo que indica una tendencia fuerte hacia la transición energética.

3. El **mercado lo lidera Toyota**, lo que muestra la fuerza de las marcas que apuestan por los híbridos accesibles.

4. Los **eléctricos puros aún son minoría (10%)**, lo que evidencia que la transición plena aún necesita más políticas públicas y desarrollo de infraestructura

2.3.3. Verificación de significancia de las predictoras

Se construyó una tabla de Chi-cuadrado junto con su respectivo valor p para analizar la relación directa entre la variable dependiente “COMBUSTIBLE” y las variables categóricas predictoras. Los resultados muestran que en todos los casos el valor p fue menor al 5%, lo que nos permite concluir que dichas variables son estadísticamente significativas.

De manera complementaria, se aplicó una prueba ANOVA sobre las variables numéricas “PESO”, “POTENCIA” y “EJES”, obteniendo el mismo resultado: todas resultaron significativas, por lo cual no se descartó ninguna de ellas. Sin embargo, todavía es necesario examinar la correlación existente entre estas variables numéricas para descartar posibles problemas de multicolinealidad.

2.3.4. Verificación de correlación y asociación entre predictoras

Para evaluar la significancia de las variables predictoras numéricas ("PESO", "POTENCIA", “EJES”) y categóricas ("MARCA", "CLASE", "SERVICIO", "DEPARTAMENTO", "CLASIFICACION", "CARROCERIA") en la predicción de "COMBUSTIBLE", realizamos un análisis exhaustivo de correlaciones y asociaciones. Para las variables numéricas, calculamos las correlaciones de Pearson y Spearman para detectar relaciones lineales y no lineales,

respectivamente. Los resultados mostraron correlaciones débiles a moderadas (máximo Spearman 0.731 entre PESO y POTENCIA), indicando ausencia de multicolinealidad significativa. Además, empleamos el Factor de Inflación de la Varianza (VIF), obteniendo valores bajos (VIF < 1.4 para todas), lo que confirmó que no había redundancia entre las numéricas, permitiendo mantenerlas todas. Este análisis aseguró que "PESO" y "POTENCIA" aportan información independiente para la clasificación.

Para las variables categóricas, utilizamos Cramér's V para medir asociaciones entre pares, identificando correlaciones fuertes (V > 0.5) entre CLASIFICACION-CLASE (V=1.0), CLASE-CARROCERIA (V=0.725), SERVICIO-CARROCERIA (V=0.674) y CLASE-SERVICIO (V=0.609), todas con p-valores < 0.05, lo que llevó a eliminar "CLASIFICACION" y "CARROCERIA" por redundancia. Para evaluar la significancia con la variable objetivo, usamos pruebas de chi-cuadrado, que mostraron asociaciones significativas (p < 0.05) para todas las categóricas, con "CARROCERIA" (Chi2=37107) y "DEPARTAMENTO" (Chi2=3029) como las más relevantes. Este enfoque aseguró la selección de variables relevantes, minimizando redundancias y abordando el desbalance de clases en "COMBUSTIBLE".

2.3.5. Desarrollo de Modelos Predictivos

Para abordar la problemática de predicción del tipo de combustible de vehículos en Colombia, se emplearon diversas librerías de Python especializadas en machine learning, principalmente scikit-learn, que permitió implementar modelos de clasificación, calcular métricas de rendimiento y gestionar el preprocesamiento de variables categóricas. Las librerías utilizadas fueron:

- sklearn.svm para Support Vector Machines (SVC)
- sklearn.neighbors para K-Nearest Neighbors (KNN)
- sklearn.ensemble para Random Forest y Gradient Boosting
- sklearn.linear_model para Logistic Regression
- sklearn.tree para Decision Tree
- sklearn.metrics para accuracy_score, precision_score, recall_score y f1_score

Se seleccionaron estos seis modelos de clasificación por sus enfoques distintos, lo que permite evaluar cuál se adapta mejor al

problema. Los modelos basados en árboles (Árbol de Decisión, Random Forest y Gradient Boosting) se eligieron por su capacidad para manejar relaciones no lineales y datos mixtos. Se incluyeron modelos lineales y de distancia (Regresión Logística, KNN y SVC) para contrastar su desempeño con métodos más tradicionales y analizar cómo responden al desbalance de clases y a la codificación de las variables.

2.3.5.1. Tratamiento de variables categóricas

Dado que las variables predictoras son nominales, se utilizó Label Encoding para los modelos basados en árboles, ya que pueden interpretar los valores numéricos de las categorías sin asumir un orden. Por otro lado, se implementó One-Hot Encoding para los modelos lineales y basados en distancia (Regresión Logística, KNN, SVC), para evitar que estos interpreten una relación ordinal inexistente en los datos. Si bien este enfoque permite la compatibilidad de los datos con cada modelo, presenta desafíos: Label Encoding puede inducir relaciones ordinales erróneas, mientras que One-Hot Encoding aumenta significativamente la dimensionalidad del dataset, afectando el rendimiento y el tiempo de entrenamiento en variables con alta cardinalidad.

2.3.5.2. Métricas de evaluación

Dado que la variable objetivo "COMBUSTIBLE" está fuertemente desbalanceada:

- GASO ELEC: 84,7%
- ELECTRICO: 12,8%
- DIES ELEC: 2,5%

Se priorizó el F1-score ponderado como métrica principal, ya que combina precisión y recall, equilibrando la evaluación entre clases mayoritarias y minoritarias. Además, se calcularon accuracy, precision y recall usando average='macro' para obtener un promedio simple de desempeño entre las tres clases y reflejar mejor la efectividad de los modelos en las clases minoritarias.

Este enfoque asegura que la selección del modelo no dependa únicamente de la clase mayoritaria, evitando sesgos y permitiendo identificar cuál algoritmo generaliza mejor y predice con mayor fidelidad los vehículos eléctricos e híbridos.

2.3.6. Optimización y Validación

2.3.7. Integración y Automatización

Para presentar resultados y facilitar la experimentación reproducible, desarrollamos una interfaz en Streamlit y PowerBI que integra un análisis previo a los datos, limpieza de datos, visualización y entrenamiento de modelos. La aplicación permite seleccionar el periodo temporal (mes/año), explorar series históricas y visualizar forecasts con intervalos de confianza. Además añadimos un panel interactivo de modelado donde se pueden entrenar y comparar algoritmos supervisados (Random Forest, Gradient Boosting, KNN, Regresión Logística, SVM), ajustar hiperparámetros con GridSearch y evaluar desempeño mediante métricas (Accuracy, Precision, Recall, F1, ROC AUC), matriz de confusión y visualización de importancia de variables. Esta integración facilita la iteración rápida entre preprocesado, modelos y visualizaciones, y permite exportar resultados para su automatización en pipelines posteriores.Para esto se usó la plataforma Streamlit.

3. Resultados esperados

3.1. Árbol de decisión

```
dt = DecisionTreeClassifier(max_depth=10, class_weight="balanced")
dt.fit(x_train, y_train)
dt_score = dt.score(x_test, y_test)

y_bar_dt_train = dt.predict(x_train)
y_bar_dt_test = dt.predict(x_test)

print(f'F1 score training set: {f1_score(y_train, y_bar_dt_train, average="macro")}')
print(f'F1 score testing set: {f1_score(y_test, y_bar_dt_test, average="macro")}\n')

print(f'Accuracy training set: {accuracy_score(y_train, y_bar_dt_train)}')
print(f'Accuracy testing set: {accuracy_score(y_test, y_bar_dt_test)}\n')

print(f'Recall training set: {recall_score(y_train, y_bar_dt_train, average="macro")}')
print(f'Recall testing set: {recall_score(y_test, y_bar_dt_test, average="macro")}\n')

print(f'Precision training set: {precision_score(y_train, y_bar_dt_train, average="macro")}')
print(f'Precision testing set: {precision_score(y_test, y_bar_dt_test, average="macro")}')
```

[117] ✓ 0.2s Python

```
F1 score training set: 0.9940504505757634
F1 score testing set: 0.9950415266429607

Accuracy training set: 0.996686269018949
Accuracy testing set: 0.9974799327982079

Recall training set: 0.9983926457217316
Recall testing set: 0.9985955644621632

Precision training set: 0.989813415124449
Precision testing set: 0.9915436842975471
```

3.2. Bosques aleatorios

```
rf = RandomForestClassifier(n_estimators=1, class_weight="balanced")
rf.fit(x_train, y_train)

y_bar_rf_train = rf.predict(x_train)
y_bar_rf_test = rf.predict(x_test)

print(f'F1 score training set: {f1_score(y_train, y_bar_rf_train, average="macro")}')
print(f'F1 score testing set: {f1_score(y_test, y_bar_rf_test, average="macro")}\n')

print(f'Accuracy training set: {accuracy_score(y_train, y_bar_rf_train)}')
print(f'Accuracy testing set: {accuracy_score(y_test, y_bar_rf_test)}\n')

print(f'Recall training set: {recall_score(y_train, y_bar_rf_train, average="macro")}')
print(f'Recall testing set: {recall_score(y_test, y_bar_rf_test, average="macro")}\n')

print(f'Precision training set: {precision_score(y_train, y_bar_rf_train, average="macro")}')
print(f'Precision testing set: {precision_score(y_test, y_bar_rf_test, average="macro")}')
```

[122] ✓ 0.1s Python

```
F1 score training set: 0.9973359575604267
F1 score testing set: 0.997211841982803

Accuracy training set: 0.9994165966582657
Accuracy testing set: 0.9993466492439799

Recall training set: 0.9977883753498649
Recall testing set: 0.9985044838962595

Precision training set: 0.9968848491984331
Precision testing set: 0.9959421184002872
```

3.3. Gradient boost

```
gb = GradientBoostingClassifier(learning_rate=0.1)
gb.fit(x_train, y_train)

y_bar_gb_train = gb.predict(x_train)
y_bar_gb_test = gb.predict(x_test)

print(f'F1 score training set: {f1_score(y_train, y_bar_gb_train, average="macro")}')
print(f'F1 score testing set: {f1_score(y_test, y_bar_gb_test, average="macro")}\n')

print(f'Accuracy training set: {accuracy_score(y_train, y_bar_gb_train)}')
print(f'Accuracy testing set: {accuracy_score(y_test, y_bar_gb_test)}\n')

print(f'Recall training set: {recall_score(y_train, y_bar_gb_train, average="macro")}
print(f'Recall testing set: {recall_score(y_test, y_bar_gb_test, average="macro")}\n'

print(f'Precision training set: {precision_score(y_train, y_bar_gb_train, average="ma
print(f'Precision testing set: {precision_score(y_test, y_bar_gb_test, average="macro
```

[39] ✓ 6.2s

```
F1 score training set: 0.9934310858841782
F1 score testing set: 0.9932745271512359

Accuracy training set: 0.9965229160832634
Accuracy testing set: 0.9963599029307448

Recall training set: 0.9909085620700101
Recall testing set: 0.9899253715219927

Precision training set: 0.9960424001094693
Precision testing set: 0.9967233918218225
```

3.4. KNN

```python
# KNN
knn = KNeighborsClassifier(n_neighbors=10)
knn.fit(x_train, y_train)

y_bar_knn_train = knn.predict(x_train)
y_bar_knn_test = knn.predict(x_test)

print(f'F1 score training set: {f1_score(y_train, y_bar_knn_train, average="macro")}')
print(f'F1 score testing set: {f1_score(y_test, y_bar_knn_test, average="macro")}\n')

print(f'Accuracy training set: {accuracy_score(y_train, y_bar_knn_train)}')
print(f'Accuracy testing set: {accuracy_score(y_test, y_bar_knn_test)}\n')

print(f'Recall training set: {recall_score(y_train, y_bar_knn_train, average="macro")}
print(f'Recall testing set: {recall_score(y_test, y_bar_knn_test, average="macro")}\n

print(f'Precision training set: {precision_score(y_train, y_bar_knn_train, average="ma
print(f'Precision testing set: {precision_score(y_test, y_bar_knn_test, average="macro
```

[ ]

```
F1 score training set: 0.9943297834715357
F1 score testing set: 0.9964087519770038

Accuracy training set: 0.9985064874451601
Accuracy testing set: 0.9984132910210939

Recall training set: 0.994835017606353
Recall testing set: 0.9963062884504087

Precision training set: 0.993825493446676
Precision testing set: 0.9965113057087991
```

3.5. Regresión logística

```python
# Regresion Logistica
lr = LogisticRegression(multi_class='multinomial', solver='lbfgs', max_iter=10000, ra
lr.fit(x_train, y_train)

y_bar_lr_train = lr.predict(x_train)
y_bar_lr_test = lr.predict(x_test)

print(f'F1 score training set: {f1_score(y_train, y_bar_lr_train, average="macro")}')
print(f'F1 score testing set: {f1_score(y_test, y_bar_lr_test, average="macro")}\n')

print(f'Accuracy training set: {accuracy_score(y_train, y_bar_lr_train)}')
print(f'Accuracy testing set: {accuracy_score(y_test, y_bar_lr_test)}\n')

print(f'Recall training set: {recall_score(y_train, y_bar_lr_train, average="macro")}
print(f'Recall testing set: {recall_score(y_test, y_bar_lr_test, average="macro")}\n'

print(f'Precision training set: {precision_score(y_train, y_bar_lr_train, average="ma
print(f'Precision testing set: {precision_score(y_test, y_bar_lr_test, average="macro
```

[144] ✓ 3m 12.6s

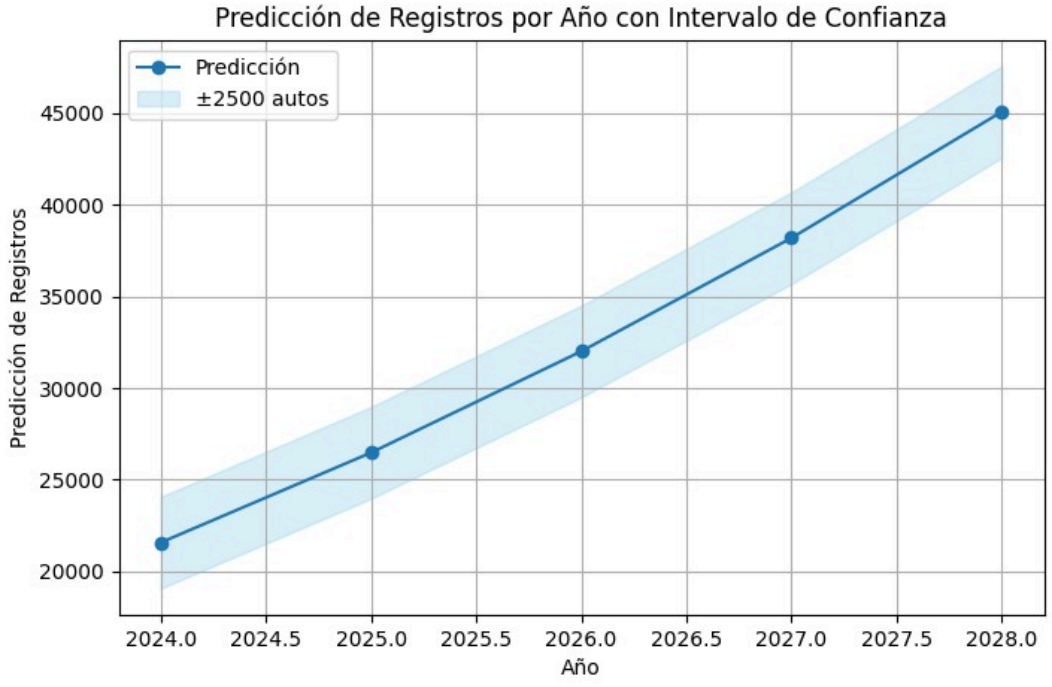
```
C:\Users\Tomás Gómez\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.10_qbz5n2k
  warnings.warn(
F1 score training set: 0.82078153308806
F1 score testing set: 0.8126865606007089

Accuracy training set: 0.9426864557080183
Accuracy testing set: 0.9425051334702259

Recall training set: 0.7683202994908976
Recall testing set: 0.7648211522082157

Precision training set: 0.8931800530498286
Precision testing set: 0.8771229640922483
```

3.6. Predicción de futuros registros



La predicción de futuros registros constituye una herramienta fundamental en el marco de la transición hacia la movilidad sostenible, ya que posibilita anticipar la evolución de la demanda de vehículos electrificados y sus implicaciones en el mercado. Contar con proyecciones confiables permite a las entidades gubernamentales, fabricantes y distribuidores planificar de manera estratégica la expansión de la infraestructura de recarga, la formulación de políticas públicas orientadas a la adopción tecnológica y el diseño de estrategias comerciales ajustadas a las tendencias emergentes. De este modo, los modelos predictivos no solo reducen la incertidumbre en la toma de decisiones, sino que también garantizan una asignación más eficiente de los recursos, contribuyendo al fortalecimiento de la transición energética y a la consolidación de un sistema de transporte más sostenible y equitativo.

4. Conclusiones

En el desarrollo de los modelos de clasificación para predecir el tipo de combustible ("COMBUSTIBLE": ELÉCTRICO, HÍBRIDO GASOLINA, HÍBRIDO DIESEL), se evaluaron Decision Tree, Random Forest, Gradient Boosting, KNN, MLPClassifier (Red Neuronal) y Logistic Regression. Todos los modelos, excepto Logistic Regression, exhibieron excelentes resultados tanto en el conjunto de entrenamiento como en el de prueba, con F1-scores superiores al 99% en la mayoría de los casos. Este desempeño excepcional se atribuye a una selección cuidadosa de variables predictoras, que incluyó el análisis de significancia mediante chi-cuadrado para categóricas y ANOVA para numéricas, eliminando redundancias (como CLASIFICACION y CARROCERIA por altas correlaciones en Cramér's V) y manejando la alta cardinalidad (agrupando categorías raras en "OTROS"). Estas variables, en conjunto, capturan patrones robustos, permitiendo una predicción precisa incluso con desbalance de clases, lo que demuestra la efectividad del preprocesamiento y la elección de modelos no lineales.

El único modelo que no alcanzó un buen desempeño, con F1-scores inferiores (tanto en entrenamiento como en prueba), fue Logistic Regression, a pesar de usar variables categóricas transformadas en dummies (One-Hot Encoding). La razón probable es que, como modelo lineal, asume relaciones lineales entre las features y la variable objetivo, lo que no se adapta bien a patrones no lineales complejos en los datos (ej. interacciones entre categóricas de alta cardinalidad y numéricas como CILINDRAJE=0 para eléctricos). Además, la alta dimensionalidad generada por dummies puede causar problemas de

convergencia o multicolinealidad residual, a diferencia de modelos basados en árboles (Decision Tree, Random Forest, Gradient Boosting) o KNN, que manejan estas complejidades de manera más flexible. El F1-score, utilizado como método de evaluación principal, es la media armónica de precisión y recall, diferenciándose de métricas como accuracy (que ignora desbalance) o precisión/recall individuales al equilibrar falsos positivos y negativos. Elegimos `average='macro'` para promediar el F1 por clase sin ponderar por soporte, tratando todas las clases por igual y mitigando el desbalance, aunque también evaluamos accuracy, precisión y recall, obteniendo resultados excelentes en todas. Dado el alto rendimiento general con leves diferencias entre los modelos (excepto Logistic Regression), no nos decantamos por uno en particular, cumpliendo el objetivo de crear un sistema robusto para clasificar combustibles de vehículos en Colombia.