Proyecto final de Aprendizaje Automático

Hernández García, Gerardo de Jesús

23 de julio de 2024

1. Introducción

Cuando se vende un auto de segunda mano, no se sabe on certeza el precio exacto con el que se debe vender, muchas de las veces por la necesidad de venderlo lo más rápido posible se venden a precios muy por debajo de lo habitual, y ésta como algunas otras situaciones impactan financieramente, tanto a los vendedores como a los compradores del vehículo.

Entonces, lo que se busca con este proyecto es poder encontrar algún modelo que nos pueda dar el valor adecuado de un auto, de acuerdo a las características del mismo. Y a su vez, reconocer que características son las que hay que tomar en consideración.

2. Descripción de los datos

Los datos que se utilizaron, es una lista de registros de autos vendidos con 12 columnas, incluida una para del precio al que se vendió el auto.

2.1. Origen de los datos

La información que se utiliza para este proyecto fue tomada de la siguiente liga, en la página de Kaggle:

https://www.kaggle.com/datasets/zeeshanlatif/used-car-price- prediction-dataset?select=train.csv

Trabajaremos más adelante la limpieza y creación de más información a partir de columnas ya establecidas.

También se está tomando como referencia el siguiente proyecto para la estructura de este documento para cumplir con la tarea 5 de la materia de Aprendizaje Automático:

https://github.com/suhasmaddali/Car-Prices-Prediction/blob/main/README.md

2.2. Librerías

Las librerías y funciones que se tendrán que cargar al procesador de python serán las siguientes:

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import plotly.graph_objects as go
import seaborn as sns
import scipy.stats as stats
import statsmodels.api as sm
import warnings
```

Figura 1: Librerías

```
from matplotlib import pyplot as plt
from mlxtend.feature_selection import ExhaustiveFeatureSelector as EFS
from mlxtend.feature_selection import SequentialFeatureSelector as SFS
from scipy.cluster.hierarchy import fcluster
from scipy.cluster.hierarchy import linkage, dendrogram
from scipy.stats import multivariate_normal
from scipy.stats import normaltest
from scipy.stats import shapiro
from scipy.spatial.distance import pdist
from sklearn.decomposition import PCA
from sklearn.ensemble import RandomForestClassifier
from sklearn.feature_selection import f_regression
from sklearn.feature_selection import r_regression
from sklearn.feature_selection import VarianceThreshold
from sklearn.feature_selection import mutual_info_regression
from sklearn.feature_selection import SelectKBest, chi2,f_regression,f_classif
from sklearn.preprocessing import LabelEncoder
from sklearn.preprocessing import MinMaxScaler
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error,mean_squared_log_error
from sklearn.metrics import r2_score, mean_squared_error, accuracy_score
from sklearn.metrics import silhouette_score
from sklearn.metrics import mean_absolute_percentage_error as mape
from sklearn.model_selection import train_test_split
from sklearn import linear_model
from sklearn.svm import SVR
from sklearn.neighbors import KNeighborsRegressor
from sklearn.tree import DecisionTreeRegressor
```

Figura 2: Funciones

2.3. Estructura de los datos

Los datos constan de 54,273 refistros (filas) y 12 variables (columnas).
A continuación, se describen las variables:

Nombre	ID_Variable	Descripción	Ejemplo
ID	id	Es solo un identificador de los registros.	11
Marca	brand	Es el nombre de la marca del auto.	Chevrolet
Modelo	model	Es el nombre del modelo del auto.	Tahoe Premier
Año del modelo	model_year	Es el año en que se lanzó el auto a la venta.	2017
Kilometraje	milage	Es el kilometraje que tenía el auto en el momento de la venta.	92728
Tipo de combustible	fuel_type	Es el tipo de combustible que maneja el auto.	Gasoline
Motos	engine	Es el motor que tiene el auto.	355.0HP 5.3L 8 Cylinder Engine Gasoline Fuel
Transmisión	transmission	Es el tipo de transmisión que tiene el auto	A/T
Color exterior	ext_col	Es el color de la parte externa del auto.	Silver
Color interior	int_col	Es el color de los interiores del auto.	Silver
Accidente	accident	Es el registro de si el auto sufrio algun accidente o no.	None reported
Titulo limpio	clean_title	Desconozco la variable, pero más adelante se eliminará ya que es el mismo dato para todos los registros.	Yes
Precio	price	Es el precio al que se vendió el auto.	49900

Como podemos observar, en el ejemplo de los datos, la mayoría de las variables son de tipo categórico, por lo que hay que tenerlo en consideración para hacer los análisis, ya sea que debamos transformarlos en índices, crear clasificaciones diferentes u obtener datos númericos de esas variables.

2.4. Análisis exploratorio de los datos

Una vez cargados los datos al procesador de python, eliminamos la columna “Título limpio” `ya` que contiene el mismo valor para todos los registros por lo que no sera significativa para los análisis posteriores y el “ID” `ya` que no representa nada más que el identificador de cada registro.

Al hacer una lectura de valores unicos de cada variable, obtenemos los siguientes resultados:

```
[ ]  autos.nunique()

id                54273
brand                53
model              1827
model_year           34
milage             3212
fuel_type             7
engine             1061
transmission         46
ext_col             260
int_col             124
accident              2
price              1481
horsepower          342
litre-capacity       61
dtype: int64
```

Donde podemos observar que no hay ninguna variable que tenga valores únicos tan grandes como los registros, por lo que podemos continuar con las variables.

En la Figura 3, se muestra el top 10 de marcas de auto que contiene la base de datos y en la Figura 4 el top 10 de modelos de auto. Esto nos hace dar cuenta que la base de datos está orientada a los autos de gama alta en el mercado, como lo es la BMW, Audi, Lexus, etc.

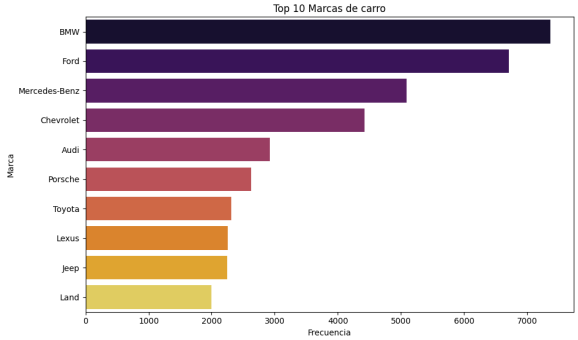

Figura 3: Estas son las 10 marcas de autos que más contiene la base de datos.

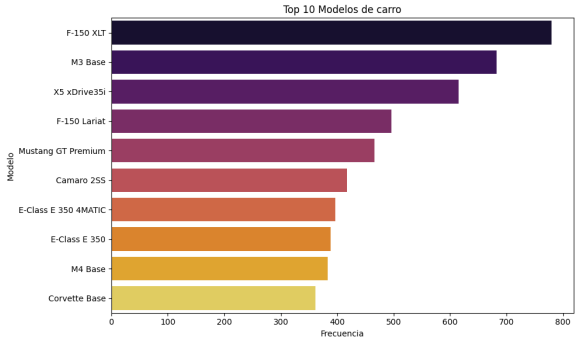

Figura 4: Estos son los 10 modelos que más contiene la base de datos.

2.5. Preprocesamiento

En el procesamiento, se crearon 2 nuevas variables para la información, las cuales fueron extraidas de la variable "Motor". Estas variables nuevas son "Caballos de fuerza " y "Capacidad de litros", los caballos de fuerza miden la potencia de un auto, entre mayor sea este número, mayor es la potencia del auto, y la capacidad de litros es una medida que nos indica cuánta cantidad de aire y combustible puede consumir el motor en cada ciclo de funcionamiento.

Por ejemplo, tenemos un registro con el motor 355.0HP 5.3L 8 Cylinder Engine Gasoline Fuel, de este dato se extrae el número 355, que está antes de los caracteres 'HP', y el número 5.3, que está antes del caracter 'L', dandonos así los datos de los caballos de fuerza y la capacidad de litros, respectivamente.

Y para los registros que en su motor no tengan estos datos, se utilizó el promedio general de cada variable de los datos obtenidos. Enriqueciendo nuestra base de datos con 2 variables numéricas más en la base de datos.

Ya que hemos agregado las variables explicadas previamente, la base de datos cuenta ahora con 5 variables numéricas que son, "Año del modelo", "Kilometraje","Precio", "Caballos de fuerza" y "Capacidad de litros".

A continuación se obtiene el siguiente resumen de los estadísticos básicos de las variables numéricas:

	count	mean	std	min	25%	50%	75%	max	Sesgo	Moda
model_year	54273.0	2015.091979	5.588909	1974.00	2012.0	2016.0	2019.0	2024.0	-0.940515	2018.000000
milage	54273.0	72746.175667	50469.490448	100.00	32268.0	66107.0	102000.0	405000.0	0.856366	60000.000000
price	54273.0	39218.443333	72826.335535	2000.00	15500.0	28000.0	45000.0	2954083.0	23.628974	15000.000000
horsepower	54273.0	331.698323	103.936245	76.00	261.0	329.0	395.0	1020.0	0.659195	331.698323
litre-capacity	54273.0	3.717009	1.328685	0.65	3.0	3.5	4.6	8.4	0.531840	3.000000

Figura 5: Tabla de estadísticos básicos

De estas variables, podemos observar que las variables "Kilometraje" y "Precio"tienen una desviación estandar amplia, y al ver los cuartiles vemos que puede haber valores atípicos en la variable "Precio".

2.6. Correlación

La correlación nos puede proporcionar información inicial de si alguna de las variables está relacionada con alguna otra, pero más importante con la variable a predecir.

En al Figura 6, podemos observar que la correlación entre las variables "Año de modelo"vs "Kilometraje" y "Precio"vs "Kilometraje" es alta pero negativa, es decir que cuando una variable sube, la otra baja. Pero el "Año de modelo"vs "Precio"tienen una correlación alta y positiva.

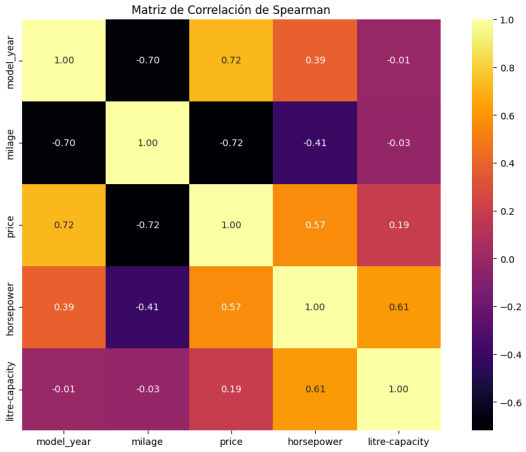

Figura 6: Correlación de Spearman sobre las variables numéricas.

2.7. Codificación de los datos

Para crear mejores análisis, vammos a transformar las variables categóricas en variables numéricas.

Con la paquetería de python "category encoders"será muy sencillo, ya que en esta paquetería ya se encuentran funciones que codifican las variables categóricas, solamente hay que seleccionar los mejores métodos para hacerlo. Por el tipo de datos que tenemos, vamos a utilizar métodos:

- El método *Target Encoding*
- El método *Base N Encoding*

El primero, se utiliza comunmente para las variables que tienen un gran número de categorías, como lo son las variables "Marca", "Modelo" y "Motor", lo que realiza este método de codificación es asignar a cada categoría el valor del promedio de la variable objetivo, que en este caso es el "Precio". Y el segundo método, asigna un código basado en el número de digitos que quieras que lo explique, por ejemplo, si utilizamos 2 dígitos, las categorías serán remplazadas por ceros y unos, pero en este caso como tenemos un gran número de categorías para algunas variables, utilizaremos un numero base mayor, 7 números, ya que si utilizamos bases más pequeñas, se incrementarían las columnas que alojan el código.

Ahora que ya tenemos "limpia"la información y las variable categóricas codificadas, es hora de navegar en algunos análisis.

3. Antecedentes

A lo largo del proyecto, se dedicó una gran parte dde tiempo en buscar documentación y preoyectos relacionados como este, de lo cual no fue dificil encontrar. Hay una variedad de ejercicios parecidos, lo diferente que note de esos casos investigados fue la información. Ya que la información que se utilizaba era más precisa y con más variables númericas que podiamos utilizar a nuestro favor, como el precio al que salió a la venta el carro cuando era nuevo, el número de cilindros del auto, los años de antigüedad que tenía cuando se vendió, entre otras variables numéricas y categóricas.

Por lo cual, decidí tomar este reto de utilizar la información que tenemos a la mano, ya que en la práctica muchas de las empresas no pueden conseguir más información específica o más detallada y se tiene que utilizar lo que hay.
Lo que me llamó la atención de todos esos ejemplos de proyectos sobre la predicción de precios del auto, es que los modelos que utilizan se ajustan bien a lo que buscan, a diferencia de este, spoiler alert, que no sale tan bien como se espera, pero se hizo un gran esfuerzo tanto de investigación como de programación y aprendizaje.

4. Metodología

Previo a realizar los análisis que se presentarán, aplicaremos algunos métodos de filtro para seleccionar las características de las mejores variables y reducir la dimensión de la base de datos.

4.1. Selección de características

4.1.1. Valor F

Con este primer filtro, vamos a revisar que variables son las que muestran una relación lineal alta con las demás variables, una vez realizado el análisis se grafican los resultados.

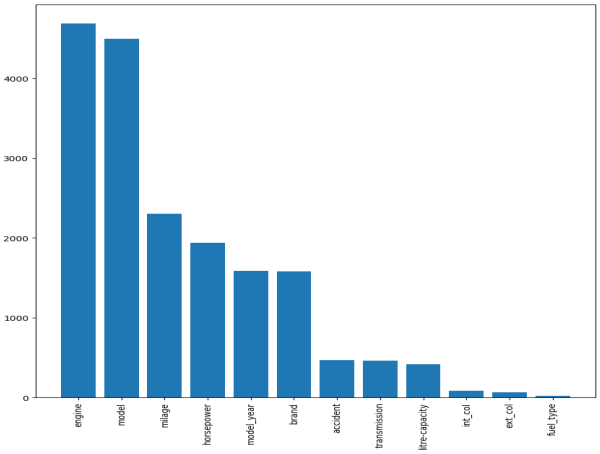

Figura 7: Resultado de los valores F de las variables.

En la Figura 7, podemos observar que las variables “Motor” y “Modelo”son las variables con una alta relación lineal.

4.1.2. R de correlación

Revisaremos la correlación que tienen las variables, como lo visto anteriormente, y gráficarlo para definir que variables están más correlacionadas.

Se puede observar en la gráfica de la Figura 8, que de nuevo dominan las variables “Motor” y “Modelo”, por lo que se encamina a que estas 2 variables si o si aparezcan en nuestros modelos, y la variable “Caballos de fuerza”tienen una correlación negativa.

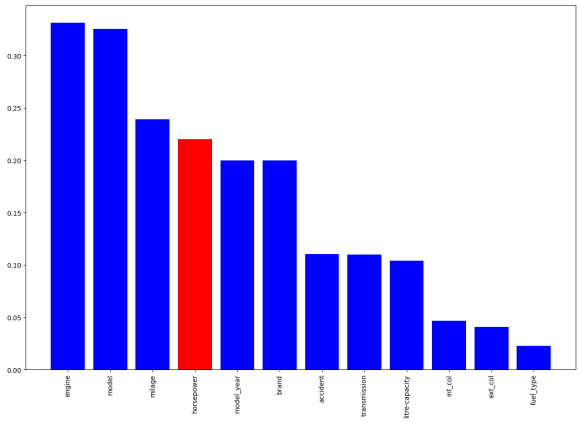

Figura 8: Resultados de correlación.

4.1.3. Umbral de varianza

Para este método, necesitaremos estandarizar las variables y despues obtener los indices de varianza para poder concluir a cerca de que variables son las que aportan mejor variabilidad a los datos.

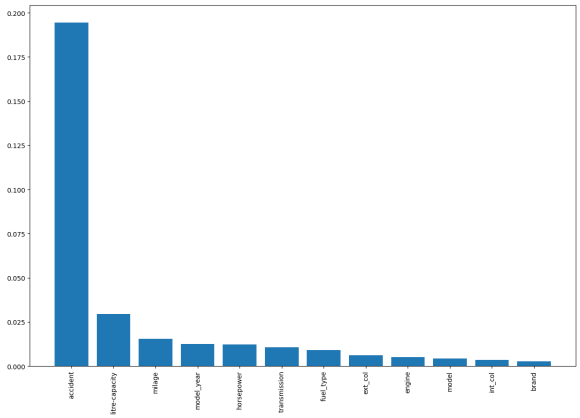

Figura 9: Umbral de varianza de los datos.

Para la consideración de este filtro, si las varianzas están por debajo de 0.2 son variables que no aportan información al modelo, dado este criterio se tendrían que descartar todas, como se observa en la Figura 9, es por eso que no solo podemos considerar un solo filtro para las variables.

4.1.4. Información Mutua

El último método de filtro que se aplica es la Información Mutua, que nos dara un panorama de que variables pueden ser independientes.

Los resultados de este filtro nos proporciona si hay dependencia entre las variables, en este caso todas son mayor a 0, por lo cual no son independientes.

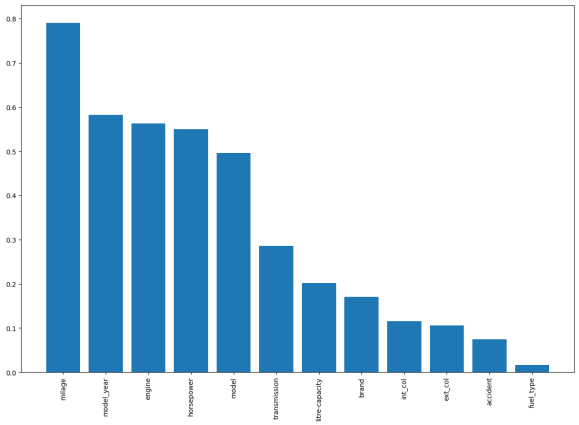

Figura 10: Información mutua

Y dando como resultado, la variable “Kilometraje" con mayor dependencia, siguiendo con poca diferencia entre ellas, las variables “Año del modelo”, “Motor”, “Caballos de fuerza” y “Modelo”.

4.1.5. Estandarización de resultados

Al hacer una estandarización de los resultados de los filtros, podemos otener una métrica general, al obtener la media de cada resultado, que aporte todos los resultados de cada filtro para poder seleccionar las variables con mejores rendimientos.
Hecho lo anterior, la siguiente gráfica muestra los resultados, y se puede observar que hay un rendimiento muy parejo entre las variables.

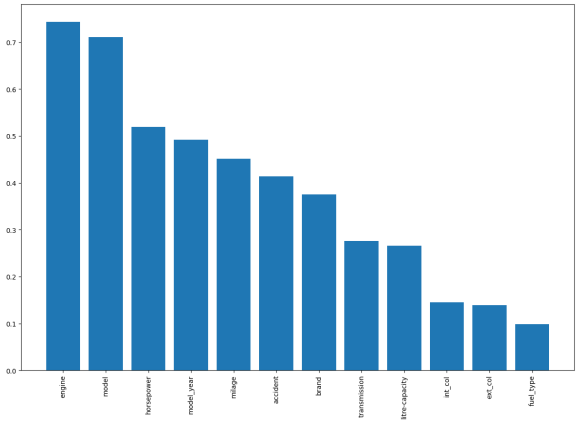

Figura 11: Indices de filtro estandarizados.

Las variables “Motor” y “Modelo”deben ser variables que formen parte del modelo, ya que tienen los resultados más altos, y de las demas variables se seleccionan las variables “Caballos de fuerza”, “Año del modelo”, “Kilometraje”,“Accidentes” y “Marca”, ya que en las variables restantes, al pasar del resultado de la variable “Marca” a “Transmisión”se nota una mayor diferencia.

Por lo tanto, las variables a las que se les dará más importancia en los modelos serán:

- "Motor"
- "Modelo"
- "Caballos de fuerza"
- "Año del modelo"
- "Kilometraje"
- "Accidentes"
- "Marca"

4.2. Ánalisis de modelos de regresión

Para este proyecto, ya que tenemos los datos para hacer una predicción del precio de los autos, y tenemos directamente conocimiento de la variable predictora, aplicaremos una serie de modelos de aprendizaje supervisado para determinar que modelo se ajusta mejor a la predicción del precio considerando solamente las mejores variables seleccionadas de la sección Selección de características.

Los modelos que utilizaremos son:

- Modelo de regresión lineal
- Modelo Regresión por Vectores de Soporte (SVR)
- Modelo Regresor de Vecinos más Cercanos (Neighbors Regressor)
- Regresor de Árbol de Decisión

Las métricas de error que se medirán para cada modelo para compararlos son:

- MAPE (Mean Absolute Percentaje Error)
- MSE (Mean Square Error)
- MAE (Mean Absolute Error)
- R^2 (Coefficient of Determination)

NOTA: Estos análisis se han realizado con los datos de entrenamiento y de prueba estandarizados.

4.2.1. Modelo de regresión lineal

Un modelo de regresión lineal es un modelo que se utiliza para entender la relación entre una variable dependiente y una o más variables independientes.

Es uno de los modelos más simples y ampliamente utilizados en el análisis de regresión, es por eso que debe de estar en nuestros modelos a analizar.

En su forma más básica, con una sola variable predictora, el modelo de regresión lineal se puede expresar como:

$$y = \beta_0 + \beta_1 x + \epsilon$$

donde:

- y es la variable dependiente,
- x es la variable independiente (predictora),
- β_0 es el término de intersección (intercepto),
- β_1 es el coeficiente de regresión (pendiente),
- ϵ es el término de error aleatorio.

4.2.2. Modelo de Regresión por Vectores de Soporte

El Modelo de Regresión por Vectores de Soporte (SVR, por sus siglas en inglés: Support Vector Regression) es una técnica para predecir valores numéricos continuos.

Este modelo se aplica ya que busca encontrar una función que se ajuste a los datos de entrenamiento mientras mantiene el margen más amplio posible y busca encontrar una función que pase lo más cerca posible de la mayoría de los puntos de datos.

En SVR, la función de regresión busca predecir un valor continuo y para un vector de características x.
El modelo de Regresión por Vectores de Soporte (SVR) se expresa como:

$$y = \langle w, x \rangle + b$$

sujeto a las restricciones:

$$\begin{cases} y_i - \langle w, x_i \rangle - b \leq \epsilon + \xi_i \\ \langle w, x_i \rangle + b - y_i \leq \epsilon + \xi_i \\ \xi_i \geq 0, \quad \sum_{i=1}^{n} \xi_i \leq C \end{cases}$$

donde:

- y es la variable dependiente que se intenta predecir,
- x es el vector de características (variables independientes),
- $\langle w, x \rangle$ es el producto punto entre el vector de pesos w y x,
- b es el término de sesgo (intercepto),

- ϵ es el margen epsilon-insensitive,
- ξ_i son variables de holgura que permiten ciertos errores de predicción,
- C es un parámetro de regularización que controla la penalización de las variables de holgura.

4.2.3. Modelo Regresor de Vecinos más Cercanos (Neighbors Regressor)

El Modelo Regresor de Vecinos más Cercanos es una técnica de aprendizaje automático utilizada para resolver problemas de regresión. A diferencia de los modelos paramétricos como la regresión lineal, el regresor de vecinos más cercanos es un modelo no paramétrico, lo que significa que no hace suposiciones explícitas sobre la forma funcional de los datos. En su lugar, aprende directamente de los datos de entrenamiento.

El algoritmo se basa en la idea de que los puntos de datos similares deben tener respuestas similares. Para predecir el valor de una nueva observación, el modelo encuentra los puntos de datos más cercanos (vecinos) en el espacio de características.

La predicción del modelo de Vecinos más Cercanos se calcula como:

$$\hat{y}_{\text{new}} = \frac{1}{k} \sum_{i=1}^{k} y_i$$

donde:

- $\hat{y}_{\text{new}}$ es el valor predicho para la nueva observación,
- y_i son los valores de la variable objetivo de los k vecinos más cercanos.

4.2.4. Regresor de Árbol de Decisión

El Modelo Regresor de Árbol de Decisión es una técnica utilizada para resolver problemas de regresión. A diferencia de los modelos lineales que intentan ajustar una función lineal a los datos, los árboles de decisión dividen iterativamente el espacio de características en regiones más pequeñas y predictivas, permitiendo así modelar relaciones no lineales y complejas entre las variables de entrada y la variable objetivo.

Un árbol de decisión se compone de nodos y ramas. Cada nodo interno representa una característica (o atributo) y una regla de decisión que divide los datos en función de esa característica. Los nodos hoja representan valores numéricos que corresponden a la variable objetivo en la regresión.

El modelo regresor de Árbol de Decisión se puede expresar de manera simplificada como:

$$\hat{y} = \sum_{m=1}^{M} c_m \cdot I(x \in R_m)$$

donde:

- $\hat{y}$ es la predicción del modelo para una nueva observación x,
- M es el número total de nodos terminales (hojas) en el árbol,
- R_m es la región de espacio de características correspondiente al nodo terminal m,
- c_m es la constante que representa el valor predicho en la hoja m,

- $I(x \in R_m)$ es una función indicadora que toma el valor 1 si x pertenece a la región R_m, y 0 en otro caso.

5. Resultados

Para evaluar el desempeño de los modelos de regresión utilizados, se decidió utilizar las metricas de error MAPE, MAE, MSE y R^2 que se mostrarán más adelante.

5.1. Modelo de regresión lineal

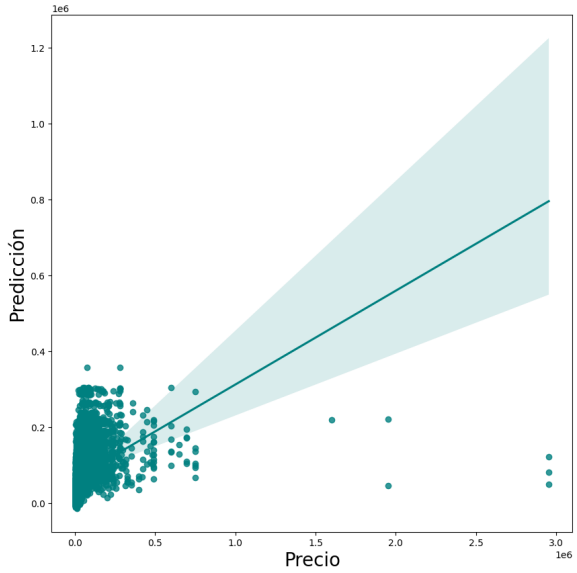

Figura 12: Predicción Modelo de regresión lineal

Como se observa en la Figura 12, la mayoría de los resultados se agrupan en la parte baja de la línea de predicción que representa que los datos estan siendo predecidos correctamente.

5.2. Modelo de Regresión por Vectores de Soporte

Al igual que el modelo anterior, vemos que los datos están muy arupados sin ajustarse a la línea de predicción,Figura 13, pero aquí se logra ver que los datos que proyecta son muy bajos a comparación de los reales.

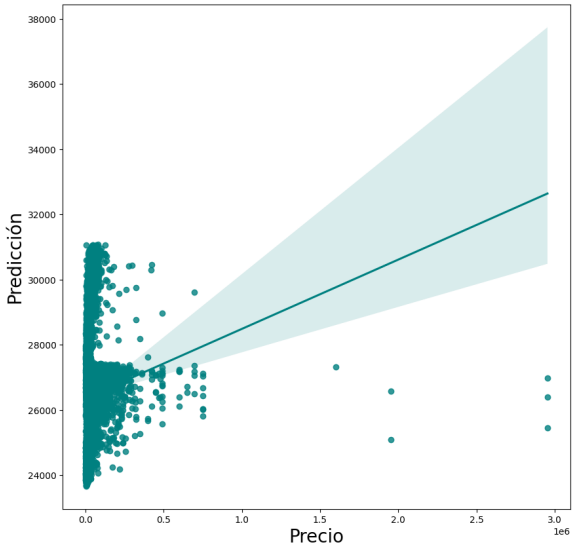

Figura 13: Predicción Modelo de Regresión por Vectores de Soporte

5.3. Modelo Regresor de Vecinos más Cercanos (Neighbors Regressor)

Los resultados para este modelo son muy similares al anterior, no se ajustan los precios proyectados.

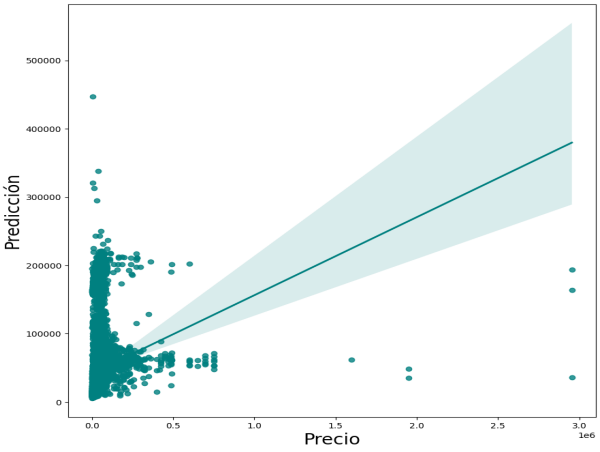

Figura 14: Predicción Modelo Regresor de Vecinos más Cercanos

5.4. Regresor de Árbol de Decisión

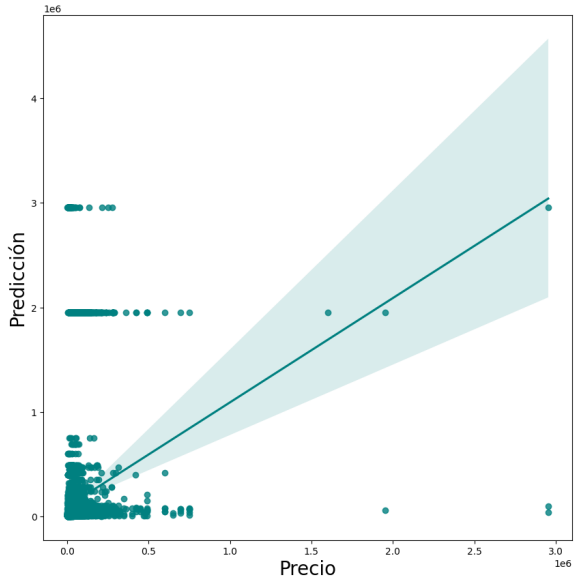

Figura 15: Predicción Regresor de Árbol de Decisión

Aquí ya se ve un poco diferentes los resultados, pero de la misma manera deficientes. Con este modelo se pueden apreciar algunos puntos dentro de la región de la línea de predicción pero la mayoría, por no decir todos, dan proyecciones más altas que los valores reales.

5.5. Errores

A continuación, se muestra una tabla con los resultados obtenidos de los errores de los modelos.

Modelo	MAPE	MSE	MAE	R^2
Regresión Lineal	0.837	5.52E+09	45,233	-1.713
SVR	0.857	4.32E+09	22,789	-1601.465
Vecinos cercanos	0.473	4.45E+09	24,524	-2.836
Árbol de desición	0.902	1.71E+11	112,755	-0.028

Como podemos observar de primera impresión, el MSE está extremadamente alto, dado que los datos predecidos están muy alejados de lo real.

Así como tambien vemos que los valores de R^2 son todos negativos, lo que nos hace suponer 2 cosas:

1. El modelo de regresión no es adecuado para los datos o no captura correctamente la estructura subyacente de la relación entre las variables.

2. La presencia de errores de medición significativos o valores atípicos en los datos puede afectar negativamente la capacidad del modelo para ajustarse adecuadamente.

Lo que en este proyecto el segundo punto sería un área de oportunidad para mejorar el análisis de los datos.

Por lo que almenos con estos análisis y modelos, la infromación no se ajusta nada bien y su predicción es altamente deficiente.

6. Discusión

En conclusión primeramente a este proyecto, vemos que despues de una gran limpieza y entendimiento de los datos, al aplicar los modelos de regresión estudiados ninguno se pudo ajustar, ni siquiera de una manera que nos que nos haga dudar, ya que directamente nos dimos cuenta de su deficiencia, lo cual puede deberse a 2 cosas principales, primeramente que el modelo no es adecuado para los datos que se analizan, y segundo que hace falta estudiar y procesar más la información, tratando de encontrar datos atípicos u otra manera de ver la información de los datos, es decir, cambiar las métricas o calificaciones de cierta manera que puedan dar resultados diferentes y mejores a los modelos.

Personalmente, lo largo de este proyecto, hubo altas y bajas, ya que fue un gran reto y una satisfacción investigar y aprender sobre todos los modelos que se pueden aplicar a las regresiones lineales y no lineales, pero cada vez que análizabamos uno, no se ajustaba. Por lo que, estos resultados me llevarán a estudiar temas más especificos y probablemente complejos sobre el procesamiento de la información previa a un análisis y temas relacionados con el entendimiento de los modelos de regresión además de los vistos en este proyecto.

Muchas gracias por la oportunidad de permitirme realizar este proyecto, me llevo muchas cosas aprendidas y con un gran interes de seguir mejorando mis conocimientos y habilidades matemáticas y de programación.

Referencias