PREDICCIÓN DE VENTA DE AUTOS

Facundo Herrera.

Proyecto de Data Science para Coderhouse.

Enero,2023.

ÍNDICE:

INTRODUCCIÓN

En este proyecto, nos proponemos desarrollar un modelo de predicción de ventas de automóviles basado en datos históricos de ventas y características de los vehículos. Utilizamos un conjunto de datos recopilados de una concesionaria automotriz. Los objetivos son entender qué factores influyen en las ventas de automóviles y desarrollar un modelo que pueda predecir las ventas futuras con un alto grado de precisión.

El rango de registros que estamos trabajando datan de los últimos 5 años de su compañía. La informaciòn brindada cuenta los siguientes datos: ID, precio, impuesto por importación o exportación(Levy), fabricante, modelo, año de fabricación, categoría, cuero interior, tipo de combustible, Volumen de motor, Millaje, cilindros, tipo de caja, tracción, puertas, rueda, color, airbags. Son 18 variables y 18924 registros.

OBJETIVOS

Desarrollando el desafío, se espera poder alcanzar una precisión mayor al 60% con algún modelo de regresión.

METODOLOGÍA

En primer lugar, limpiamos y procesamos los datos, eliminando registros duplicados y valores faltantes. Luego, utilizamos técnicas de análisis exploratorio de datos para entender la relación entre las características de los vehículos y las ventas. Utilizamos un conjunto de entrenamiento y prueba para entrenar y evaluar diferentes modelos de regresión.

PRIMERAS HIPÓTESIS

-Futuro precio de los coches depende de ciertas características.

-Existe una predominancia en ventas de coches sedán.

-¿La seguridad es un factor importante a la hora de la compra de un automóvil?

-El valor del Levy (impuesto por importación/exportación) depende del año del automóvil.

DICCIONARIO DE VARIABLES:

ID: Número único de identificación de venta.

Price: Precio de venta de cada auto.

Levy: Gasto de importación/exportación del vehículo.

Manufacturer: Empresa de manufactura.

Model: Modelo del auto.

Prod. Year: Año de salida del auto.

Category: Categoría en la que se encuentra el vehículo, como Sedan, Jeep, Coupé, etc.

Leather interior: Si tiene asientos de cuero o no.

Fuel type: Tipo de combustible que consume el vehículo.

Engine volume: Volumen del motor, pueden ser motores 1.8, 2.0, etc.

Mileage: Kilometraje del vehículo.

Cylinders: Cantidad de cilindros de acuerdo al motor.

Gear box type: Tipo de caja de cambios.

Drive wheels: Tipo de tracción del auto.

Doors: Número de puertas.

Wheel: Lado en el que está el volante.

Color: Color del vehículo.

Airbags: Número de airbags.

EDA (EXPLORATORY DATA ANALYSIS)

PRIMERA LIMPIEZA

Como primer paso vemos una vista de la data cargada en primer lugar.

	ID	Price	Levy	Manufacturer	Model	Prod. year	Category	Leather interior	Fuel type	Engine volume	Mileage	Cylinders	Gear box type	Drive wheels	Doors	Wheel	Color	Airbags
0	45654403	13328	1399	LEXUS	RX 450	2010	Jeep	Yes	Hybrid	3.5	186005 km	6.0	Automatic	4x4	04-May	Left wheel	Silver	12
1	44731507	16621	1018	CHEVROLET	Equinox	2011	Jeep	No	Petrol	3	192000 km	6.0	Tiptronic	4x4	04-May	Left wheel	Black	8
2	45774419	8467	-	HONDA	FIT	2006	Hatchback	No	Petrol	1.3	200000 km	4.0	Variator	Front	04-May	Right-hand drive	Black	2
3	45769185	3607	862	FORD	Escape	2011	Jeep	Yes	Hybrid	2.5	168966 km	4.0	Automatic	4x4	04-May	Left wheel	White	0
4	45809263	11726	446	HONDA	FIT	2014	Hatchback	Yes	Petrol	1.3	91901 km	4.0	Automatic	Front	04-May	Left wheel	Silver	4
5	45802912	39493	891	HYUNDAI	Santa FE	2016	Jeep	Yes	Diesel	2	160931 km	4.0	Automatic	Front	04-May	Left wheel	White	4
6	45656768	1803	761	TOYOTA	Prius	2010	Hatchback	Yes	Hybrid	1.8	258909 km	4.0	Automatic	Front	04-May	Left wheel	White	12
7	45816158	549	751	HYUNDAI	Sonata	2013	Sedan	Yes	Petrol	2.4	216118 km	4.0	Automatic	Front	04-May	Left wheel	Grey	12
8	45641395	1098	394	TOYOTA	Camry	2014	Sedan	Yes	Hybrid	2.5	398069 km	4.0	Automatic	Front	04-May	Left wheel	Black	12
9	45756839	26657	-	LEXUS	RX 350	2007	Jeep	Yes	Petrol	3.5	128500 km	6.0	Automatic	4x4	04-May	Left wheel	Silver	12

En esta sección dictaremos los pasos que aplicamos:

- Se cambiaron los nombres de las columnas para minimizar la cantidad de errores de código con los espacios, todos los espacios vacíos se cambiaron a "_"
- Se reemplazaron todos los valores de "-" de los registros a NaN para tratarlos luego en conjunto.
- Se buscaron duplicados por la columna "ID" y se encontraron 313 registros duplicados. Fueron eliminados.
- Se eliminó la columna ID ya que no nos era de utilidad para los pasos siguientes
- Se cambiaron los valores de "Mileage", se eliminó la palabra "km" de todos los registros de la variable y la convertimos a una variable numérica.
- En la columna "Door" la data fue cargada como fecha, pero se necesitaba un número de puertas para cada vehículo, siendo '2', 2-3 puertas, '4': 4-5 puertas y '5': más de 5 puertas.
- Se encontraron 5709 valores nulos en la variable "Levy":

```
Price                0
Levy              5709
Manufacturer         0
Model                0
Prod_year            0
Category             0
Leather_interior     0
Fuel_type            0
Engine_volume        0
Mileage              0
Cylinders            0
Gear_box_type        0
Drive_wheels         0
```

- En la columna “Engine_volume” se encontraron tanto tipos de motores con Turbo y sin él.

```
Engine_volume:
['3.5' '3' '1.3' '2.5' '2' '1.8' '2.4' '4' '1.6' '3.3' '2.0 Turbo'
 '2.2 Turbo' '4.7' '1.5' '4.4' '3.0 Turbo' '1.4 Turbo' '3.6' '2.3'
 '1.5 Turbo' '1.6 Turbo' '2.2' '2.3 Turbo' '1.4' '5.5' '2.8 Turbo' '3.2'
 '3.8' '4.6' '1.2' '5' '1.7' '2.9' '0.5' '1.8 Turbo' '2.4 Turbo'
 '3.5 Turbo' '1.9' '2.7' '4.8' '5.3' '0.4' '2.8' '3.2 Turbo' '1.1' '2.1'
 '0.7' '5.4' '1.3 Turbo' '3.7' '1' '2.5 Turbo' '2.6' '1.9 Turbo'
 '4.4 Turbo' '4.7 Turbo' '0.8' '0.2 Turbo' '5.7' '4.8 Turbo' '4.6 Turbo'
 '6.7' '6.2' '1.2 Turbo' '3.4' '1.7 Turbo' '6.3 Turbo' '2.7 Turbo' '4.3'
 '4.2' '2.9 Turbo' '0' '4.0 Turbo' '20' '3.6 Turbo' '0.3' '3.7 Turbo'
 '5.9' '5.5 Turbo' '0.2' '2.1 Turbo' '5.6' '6' '0.7 Turbo' '0.6 Turbo'
 '6.8' '4.5' '0.6' '7.3' '0.1' '1.0 Turbo' '6.3' '4.5 Turbo' '0.8 Turbo'
 '4.2 Turbo' '3.1' '5.0 Turbo' '6.4' '3.9' '5.7 Turbo' '0.9' '0.4 Turbo'
 '5.4 Turbo' '0.3 Turbo' '5.2' '5.8' '1.1 Turbo']
```

  Decidimos dividir estos registros en una variable booleana distinta llamada “Turbo” y cortar el texto para que la variable pase a ser numérica.
- Se cambiaron todos los valores nulos a valor “0” (todos de ‘Levy’) ya que hay autos que no necesariamente deberían tener impuestos de importación y exportación.
- Se apartaron todos los registros con valores menores a U$D500, los valores de motor iguales a ‘0’ para tratarlos luego del modelo y no perder data.
- Hasta antes de tratar los outliers, la data se muestra de la siguiente manera:

```
Int64Index: 18924 entries, 0 to 19236
Data columns (total 18 columns):
 #   Column            Non-Null Count  Dtype
---  ------            --------------  -----
 0   Price             18924 non-null  int64
 1   Levy              18924 non-null  int64
 2   Manufacturer      18924 non-null  object
 3   Model             18924 non-null  object
 4   Prod_year         18924 non-null  int64
 5   Category          18924 non-null  object
 6   Leather_interior  18924 non-null  object
 7   Fuel_type         18924 non-null  object
 8   Engine_volume     18924 non-null  float64
 9   Mileage           18924 non-null  int64
 10  Cylinders         18924 non-null  float64
 11  Gear_box_type     18924 non-null  object
 12  Drive_wheels      18924 non-null  object
 13  Doors             18924 non-null  int64
 14  Wheel             18924 non-null  object
 15  Color             18924 non-null  object
 16  Airbags           18924 non-null  int64
 17  Turbo             18924 non-null  bool
dtypes: bool(1), float64(2), int64(6), object(9)
```

En la siguiente sección sacamos algunos primeros tipos de visualización para entender un poco más cómo se comportan las variables y cómo tratar los outliers.

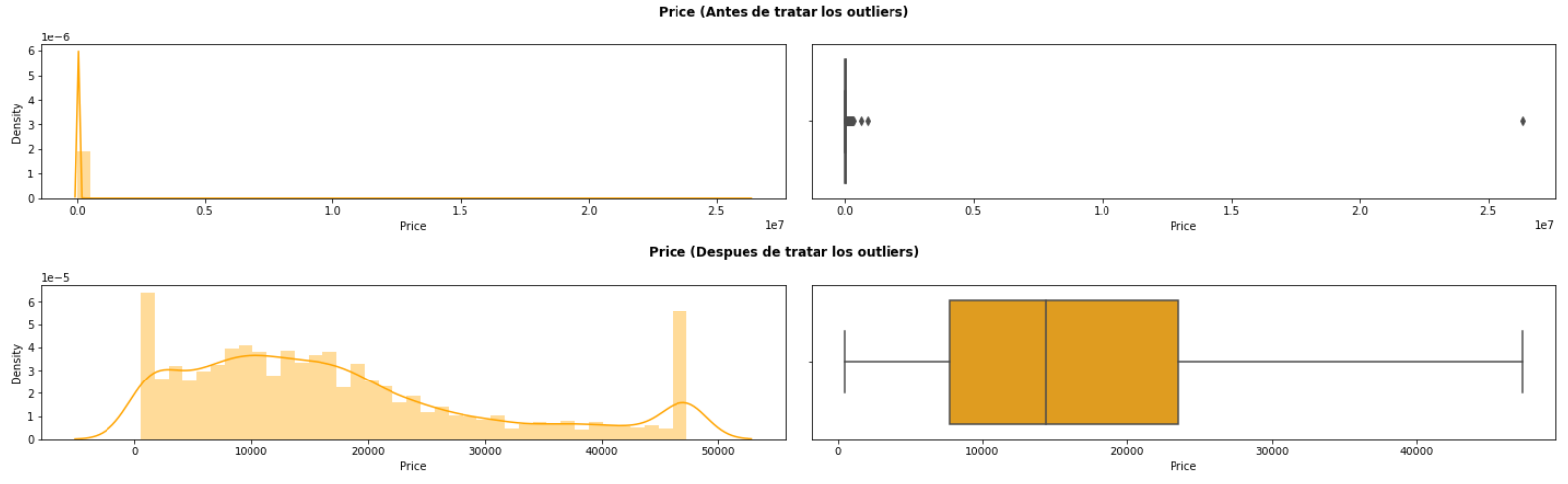
OUTLIERS Y PRIMERAS VISUALIZACIONES

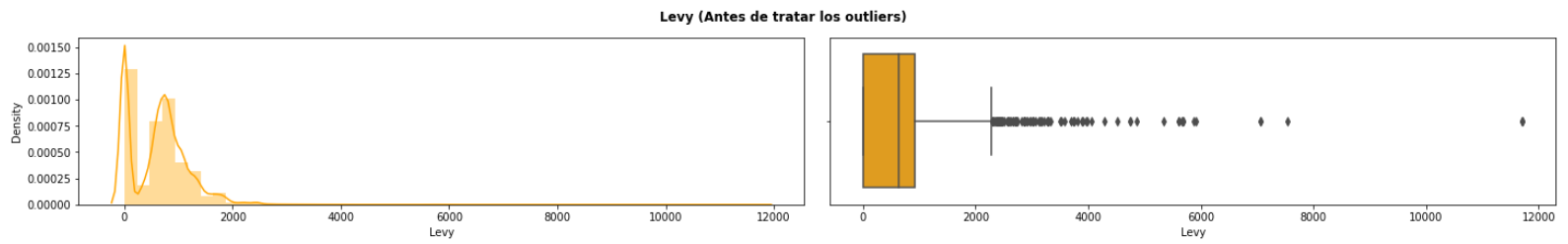
Como tratamiento de los outliers decidimos aplicar Rango Interquartil a todas las variables y separar los registros en un listado aparte llamado “Outliers”, lo guardamos en un dataframe fuera del que vamos a estar trabajando para no tener complicaciones al momento de implementar los modelos.

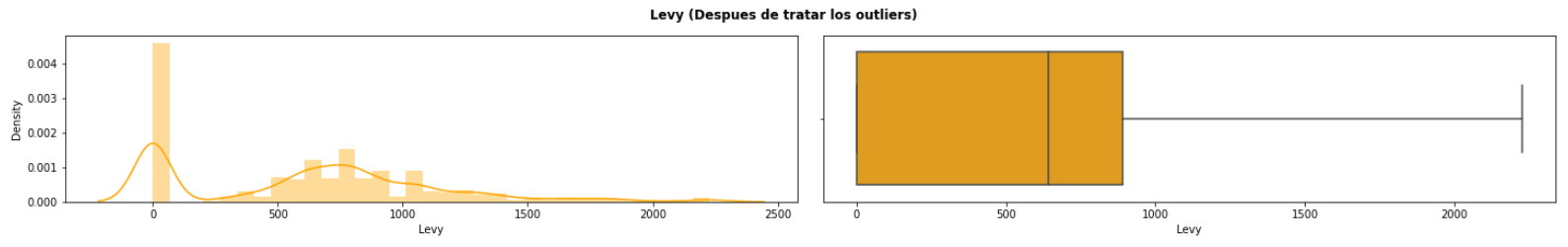
Un estimado de los outliers por cada variable es:
Outliers en Price son: 1043 - 6.03%.
Outliers en Levy son: 133 - 0.77%.
Outliers en Engine_volume son: 1172 - 6.78%.
Outliers en Mileage son: 626 - 3.62%.

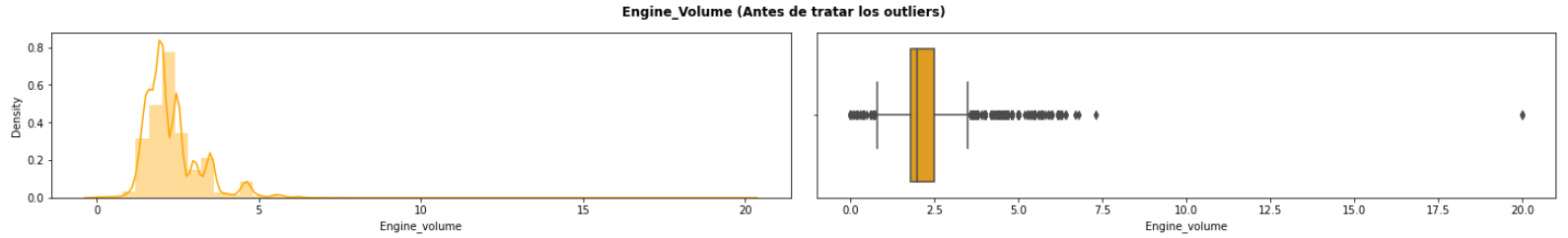
Muchos de estos outliers están concatenados entre variables.
La estructura de la data filtrada es: (17291, 18).

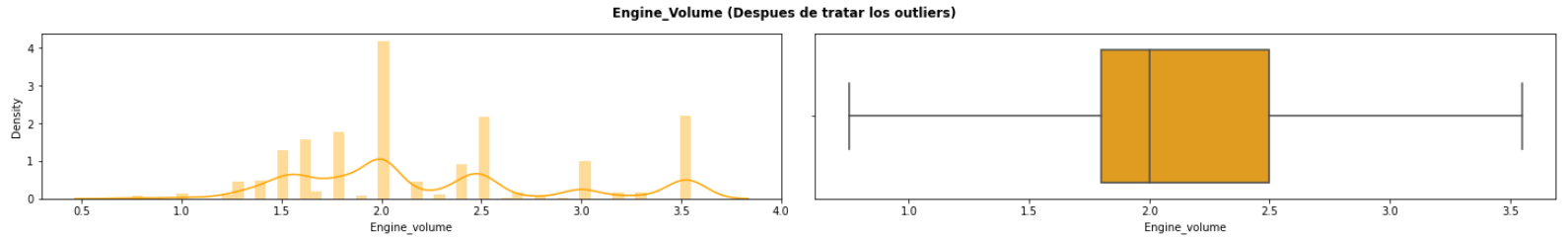
Ahora haremos un listado de visualizaciones de todas las variables numéricas antes y después de tratar los outliers:

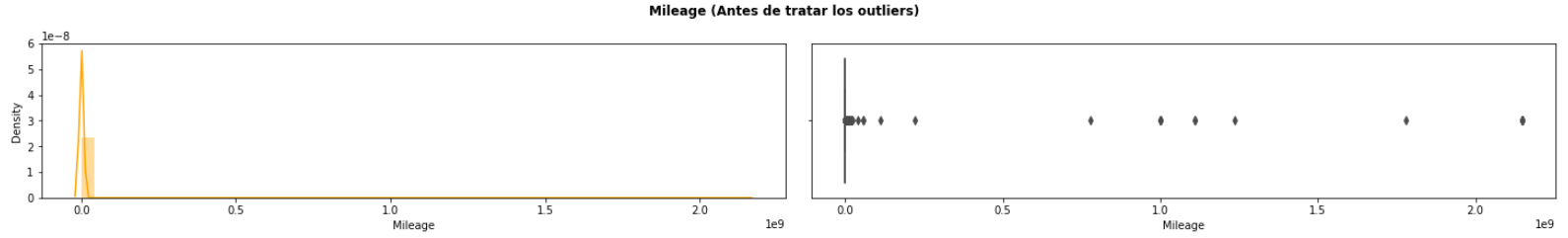


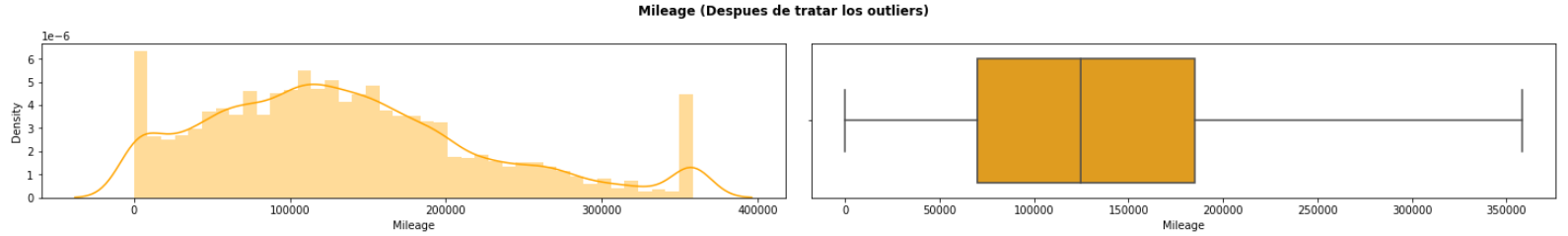






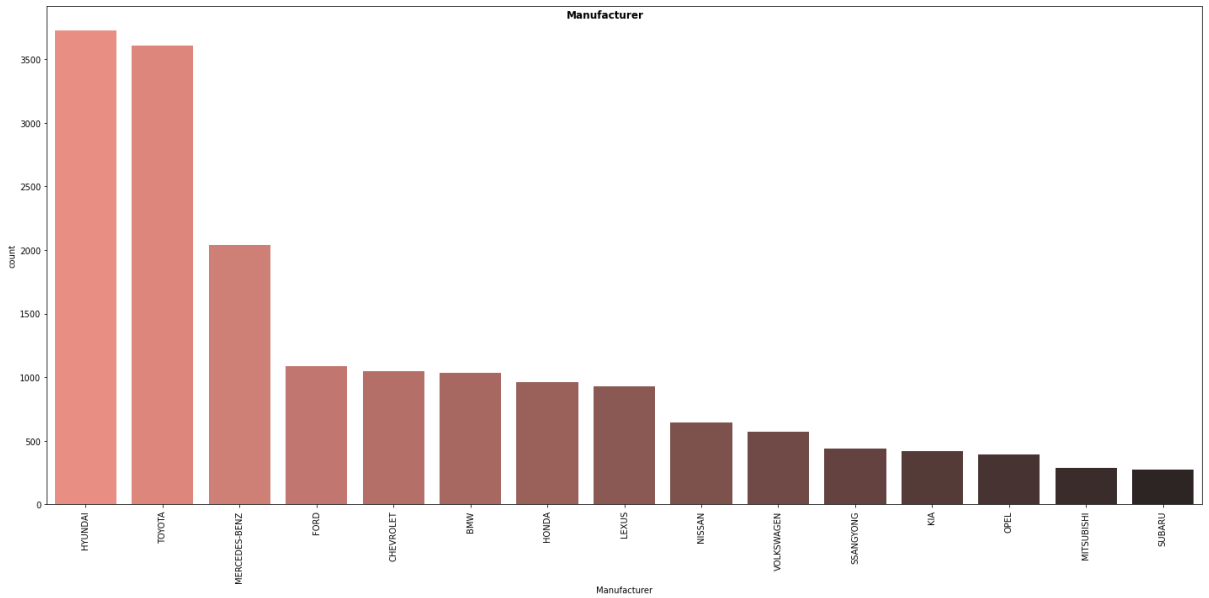


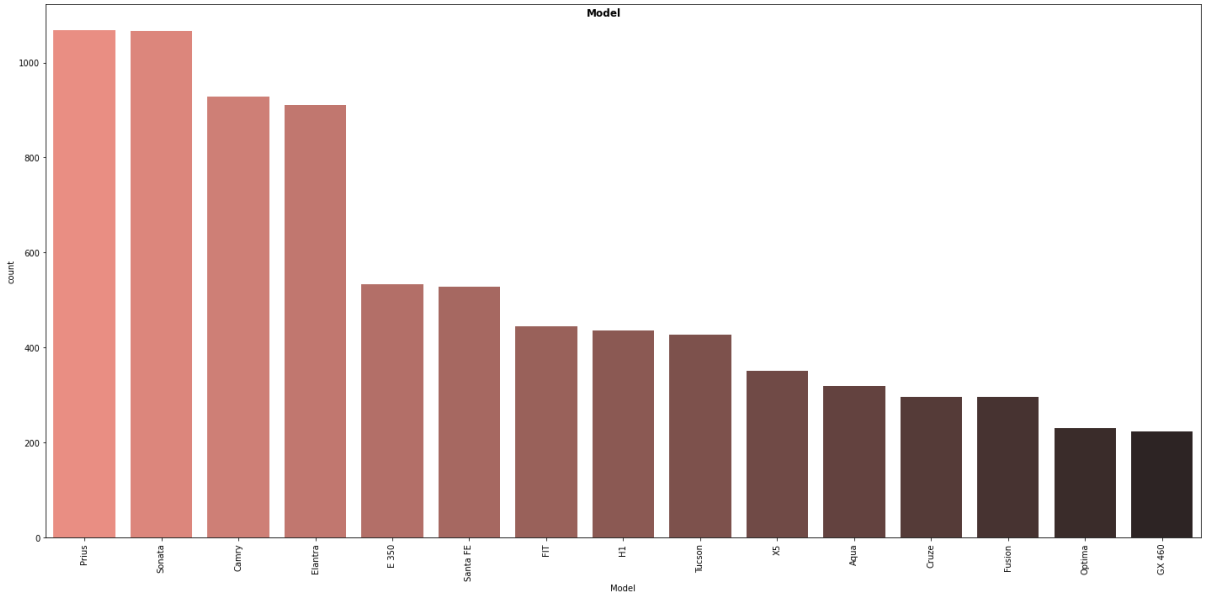


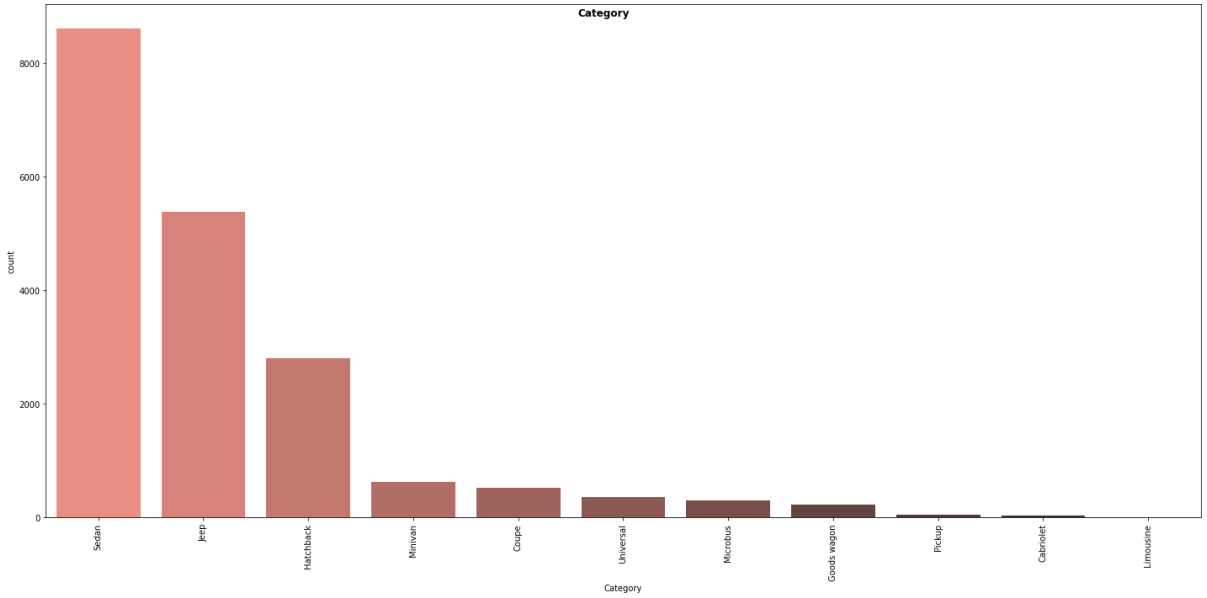


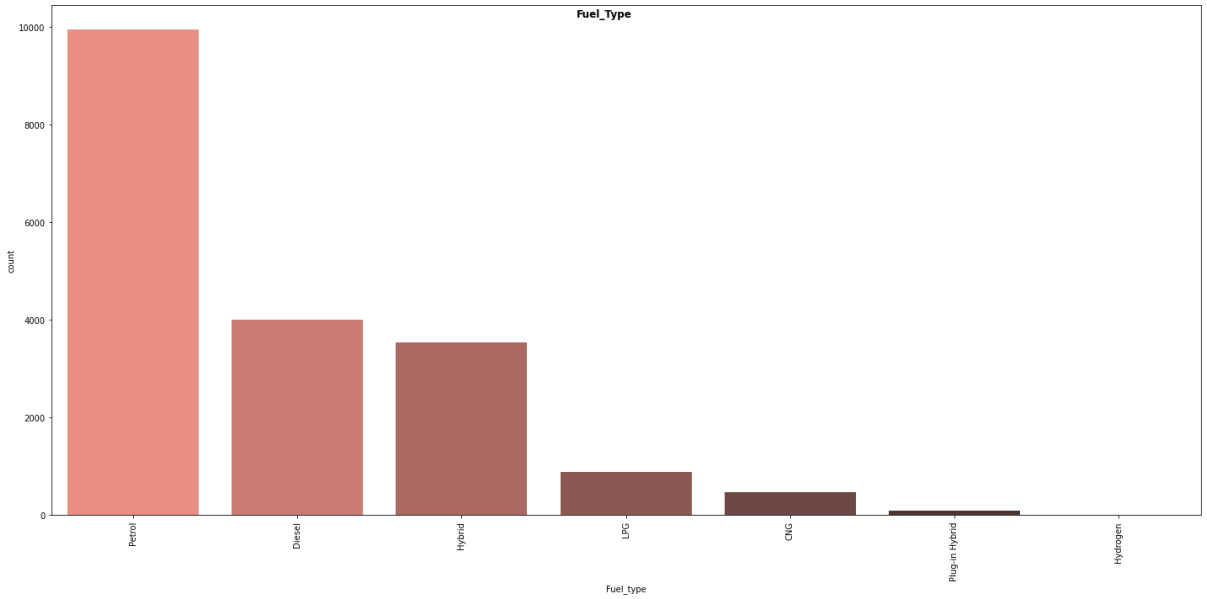
Tanto en “Price” como en “Mileage” se encontraron outliers exageradamente elevados o sumamente chicos que con este método fueron tratados correctamente, se puede ver más a fondo en el código.

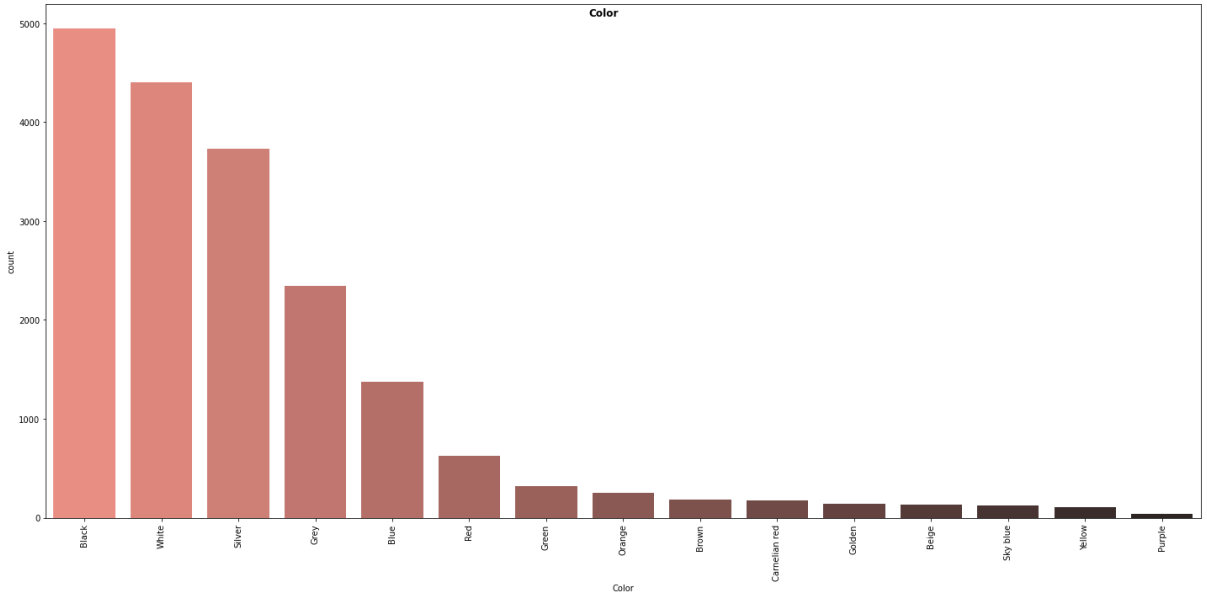
Para seguir describiendo un poco las variables categóricas se hicieron las siguientes visualizaciones:




Model
count
Prius
Sonata
Camry
Elantra
E 350
Santa FE
FIT
H1
Tucson
X5
Aqua
Cruze
Fusion
Optima
GX 460
Model


Category
count
Sedan
Jeep
Hatchback
Minivan
Coupe
Universal
Microbus
Goods wagon
Pickup
Cabriolet
Limousine
Category


Fuel_Type
count
Petrol
Diesel
Hybrid
LPG
CNG
Plug-in Hybrid
Hydrogen
Fuel_type



PRIMEROS INSIGHTS

-Hyunday, Toyota y Mercedes-Benz encabezan la lista de las marcas con más liquidez.

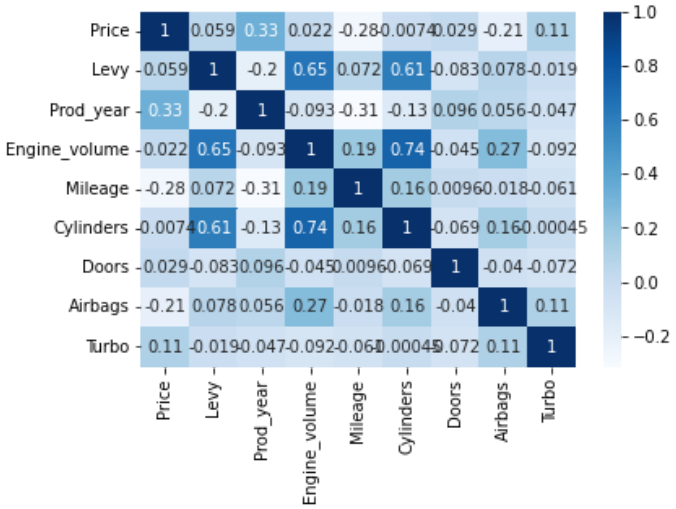
-Los coches Sedán, como era de esperarse, son los tipos de coches más elegidos por el cliente seguidos por los Jeep y los Hatchback.

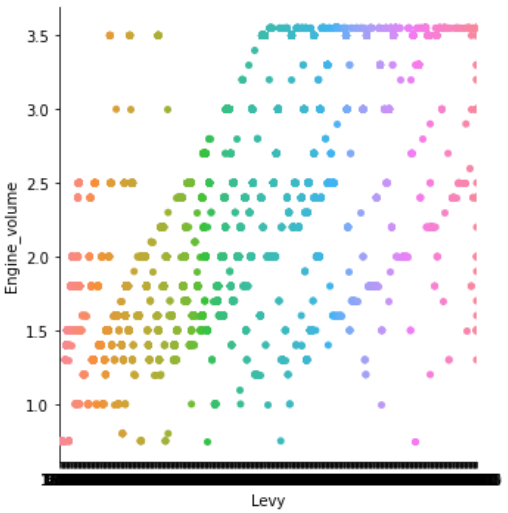
-Los combustibles a base de petróleo siguen siendo de los más comprados, seguidos por Diesel y los tipo Híbrido. Aunque entre estos dos tipos la diferencia no es tan grande.

-Los autos de caja automática lideran con fuerza las ventas.

-Tenemos buenos insights del color y de los tipos de tracción de los autos, pero entendiendo un poco el mercado y cómo los vamos a utilizar a futuro, no creo que aporten mucho a nuestro modelo.

-Notamos una relación entre “Engine_volume” y “Cylinders” con Levy.





Como podemos ver, depende del volumen del motor y los cilindros del auto, el Levy puede cambiar entre 800 y 1200 dólares. Seguimos buscando más información sobre esto y pudimos ver como los valores de Levy y Price de los vehículos HYUNDAI Sonata y otros tienen cambios mucho más claros. De acuerdo con el Año del vehículo y el Engine_volume el precio va modificándose gradualmente. Solo en casos extremos de Engine Volume el valor de Levy cambia drásticamente.

Engine_volume = 3.3 / Levy = 2227.5
Engine_volume = 2.0 / Levy = 891
Pero en una situación más clara, podemos ver que:

Prod_year = 2018 / Levy = 1079
Prod_year = 2017 / Levy = 1017
Prod_year = 2016 / Levy = 891
Prod_year < 2015 sigue bajando progresivamente.

Después de estos insight se eliminó la variable color ya que no nos será útil para el modelado y tenemos toda la información que necesitamos.

ENCODING

-Ordinal Encoder
Usamos esta librería para enumerar los valores de los registros categóricos:

-One Hot Encoder
Las variables booleanas las transformamos con esta librería. Aplicamos “drop='if_binary'” para que la única columna que quede sea las de valores positivos.

-Aplicamos MinMaxScaler para probar PCA y escalar los valores para los siguientes modelos. Al tener poca varianza escogimos aplicar este proceso de escalado.

MODELADO

ALGORITMOS ELEGIDOS

Los algoritmos que vamos a utilizar son supervisados y se listan aquí:

- Linear Regression
- DecisionTreeRegressor
- RandomForestRegressor
- AdaBoostRegressor
- GradientBoostingRegressor
- XGBRegressor

El modelo de regresión con árboles de decisión resultó ser el mejor modelo, con una precisión del 60% en el conjunto de prueba. Los resultados también mostraron que los modelos Gradient Boosting Regressor y XGBoost Regressor retornaban buenos resultados sin hyperparámetros.

-Se usó un 70% de la data para entrenamiento y un 30% para evaluación.
-Se probó usar PCA para reducir la cantidad de variables pero el nivel de varianza era bajo muy bajo para utilizarlo.

R2 Score:

Random Forest Train: 95 %

Random Forest Test: 60 %

—----------

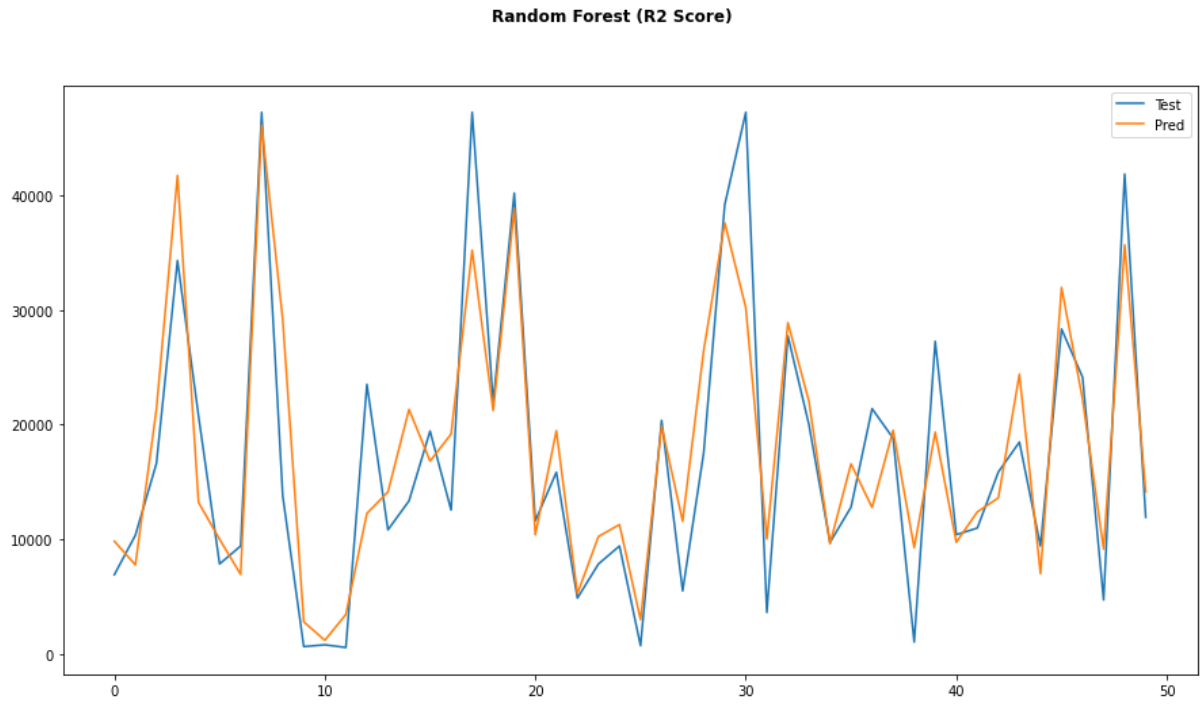
Gradient Boosting Regressor Train: 58 %

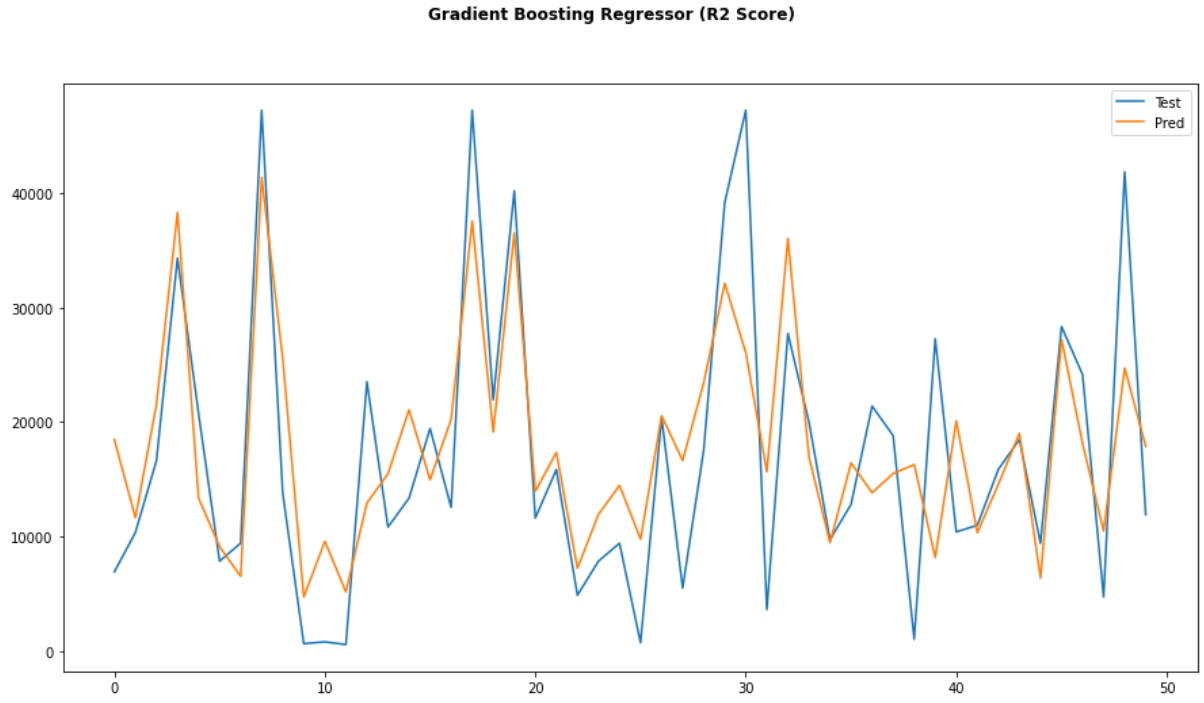
Gradient Boosting Regressor Test: 56 %

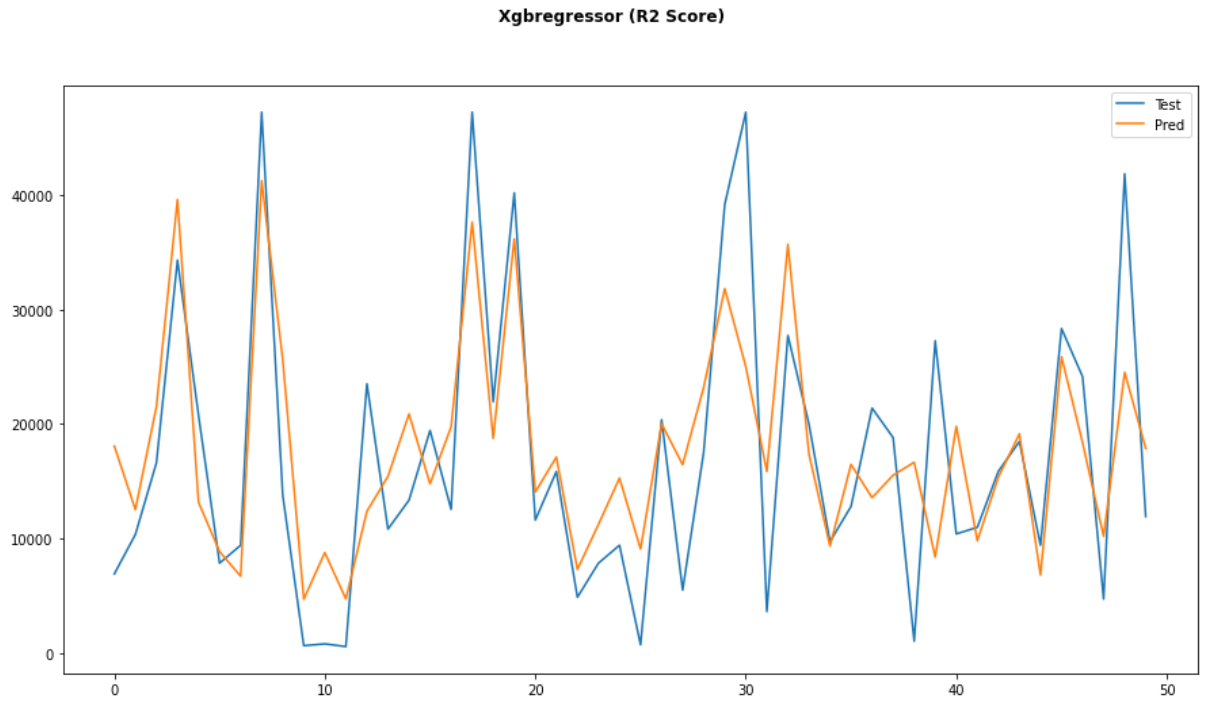
—----------

XGBRegressor Train: 58 %

XGBRegressor Test: 56 %







Los demás modelos estaban muy por debajo de estos últimos tres.

HYPERTUNING

Se logró un aumento del 4% en la precisión luego de llevar a cabo el proceso de hypertuning.

Hyperparámetros en cada GridSearchCV:

##{'max_depth': 16, 'n_estimators': 200}
##{'max_depth': 20, 'n_estimators': 250}
##{'max_depth': 25, 'n_estimators': 300}
##{'max_depth': 30, 'n_ estimators: 300} max_depth estaba en 25-30-35. Encontramos tope en 30
##{'max_depth': 35, 'n_estimators': 400}

Distintos resultados después de los hyper tunings.

#Primer R2_Score: 61
#Segundo R2_Score: 62.1
#Tercer R2_Score: 62.9
#Cuarto R2_Score: 64

CONCLUSIÓN

El modelo de Random Forest Regressor resultó ser el mejor modelo, con una precisión del 64% en el conjunto de pruebas. Los resultados también mostraron que las características de los vehículos como el modelo, el año y el tipo de combustible son los factores más importantes en la determinación de las ventas.

En este proyecto, desarrollamos un modelo eficaz para predecir las ventas de automóviles utilizando técnicas de aprendizaje automático. Este modelo puede ser evaluado por ustedes y estamos abiertos a cualquier tipo de cambio en el proyecto que crean necesario.