SEP
SECRETARÍA DE EDUCACIÓN PÚBLICA

TECNOLÓGICO NACIONAL DE MÉXICO

Proyecto final

Aprendizaje Automático II

Prof. Eduardo Antonio Hinojosa Palafox

Alumno:
Salgado Mahuari Julio Cesar - 18330500

Hermosillo, Sonora a 15 de diciembre de 2022

Tabla de contenido

INTRODUCCIÓN

La minería de datos es la práctica de examinar datos que ya fueron recolectados, utilizando diversos tipos de algoritmos, normalmente de forma automática, con la finalidad de generar nuevas informaciones y encontrar patrones. Esto es lo que se presenta en este documento, se utilizará un conjunto de datos recolectado en un una página preseleccionados en las instrucciones de la tarea. Se investigará un problema al que se propondrá por medio de un algoritmo estadístico solucionarlo.

Antes de comenzar con el apartado del modelo estadístico se tendrá que realizar una limpieza del conjunto de datos, esto se documentará buscando qué valores son los que fallaron desde el datahouse en el que se previene.

Finalmente, con propósito de poder demostrar en un futuro nuestras habilidades de ciencias de los datos, se tendrá que cargar el proceso realizado para la solución del problema y su correspondiente documentación en la plataforma de GitHub que es es un portal creado para alojar el código de las aplicaciones de cualquier desarrollador.

DESCRIPCIÓN DEL PROBLEMA A DESARROLLAR

El proyecto que se plantea en este documento es conocer el precio de un auto dando solamente características de estas, el número de características es demasiado grande como para que una persona experta en la mercancía de autos pueda predecir el precio. Por lo que, se necesita un modelo estadístico que sea capaz de al ingresar información de un auto, pueda predecir un precio estimado de un auto con esas características. Las preguntas que nos competen responder serían “Según estas características, ¿qué precio tiene un auto con estas características?”.

Para esto se hará uso de algún algoritmo de aprendizaje supervisado que es una rama de aprendizaje automático, un método de análisis de datos que utiliza algoritmos que aprenden iterativamente de los datos para permitir que las computadoras encuentren información escondida sin tener que programar de manera explícita dónde buscar.

CONJUNTO DE DATOS

Este conjunto de datos es un conjunto de datos popular para estudiar técnicas de regresión del repositorio de aprendizaje automático de UCI que se puede encontrar aquí: http://archive.ics.uci.edu/ml/datasets/Automobile

Este conjunto de datos consta de tres tipos de entidades:

- La especificación de un automóvil en términos de varias características.
- Su calificación de riesgo de seguro está asignada.
- Sus pérdidas normalizadas en uso en comparación con otros automóviles.

La segunda calificación corresponde al grado en que el auto es más riesgoso de lo que indica su precio. A los automóviles se les asigna inicialmente un símbolo de factor de riesgo asociado con su precio. Luego, si es más arriesgado (o menos), se ajusta este símbolo moviéndolo hacia arriba (o hacia abajo) en la escala.

El tercer factor es el pago de pérdida promedio relativo por año de vehículo asegurado. Este valor está normalizado para todos los autos dentro de una clasificación de tamaño particular (pequeño de dos puertas, familiar, deportivo/especial, etc.), y representa la pérdida promedio por auto por año.

DESCRIPCIÓN DE LA SOLUCIÓN PROPUESTA

Para la solución de este proyecto se ha descubierto que es un problema que se puede solucionar con una regresión que es un problema de aprendizaje supervisado que le pide al modelo que prediga un número. Se planea utilizar diferentes tipos de regresión para poder conocer el algoritmo con el mejor desempeño.

Se planea utilizar alguno de los siguientes algoritmos:

- Regresión lineal: es una técnica de modelado estadístico que se emplea para describir una variable de respuesta continua como una función de una o varias variables predictoras.
- Decision Tree Regression: son una técnica de aprendizaje supervisado que predice valores de respuestas mediante el aprendizaje de reglas de decisión derivadas de características.

Antes de realizar lo anterior, se realizará un análisis del conjunto de datos para poder conocer valores perdidos que son aquellos en los que no se almacena ningún valor de datos en una observación. Después solucionaremos este problema con algún método lógico.

DESCRIPCIÓN DEL CÓDIGO UTILIZADO

Importamos las librerías necesarias para realizar este proyecto y serán los siguientes:

- matplotlib.pyplot: es una interfaz basada en estado para matplotlib.
- pandas: es una librería en Python que se especializa en el manejo, análisis y procesamiento de datos.
- sklearn: ofrece unas estructuras muy poderosas y flexibles que facilitan la manipulación y tratamiento de datos.

```
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn import preprocessing
import warnings
warnings.filterwarnings('ignore')
%matplotlib inline
```

Después utilizamos la librería pandas para poder recoger el conjunto de datos csv. Después imprimimos los primeros 5 registros para conocer sus datos.

```
cars = pd.read_csv('/content/Automobile_data.csv')
cars.head()
```

	symboling	normalized-losses	make	fuel-type	aspiration	num-of-doors	body-style	drive-wheels
0	3	?	alfa-romero	gas	std	two	convertible	rwd
1	3	?	alfa-romero	gas	std	two	convertible	rwd
2	1	?	alfa-romero	gas	std	two	hatchback	rwd
3	2	164	audi	gas	std	four	sedan	fwd
4	2	164	audi	gas	std	four	sedan	4wd

Podemos observar que dentro de esta muestra de datos existen valores erróneos como el “?”

que son valores que fallaron a la hora de cargarse o desde la creación de este. Por lo que tendremos que cambiarlos. Para esto utilizaremos una característica de la función desde donde leemos el conjunto de datos. Conociendo de antemano que tipo de valor es el que se registra cuando es inexistente, podemos utilizar la propiedad na_values para darle su correspondiente valor nulo a cada registro fallido.

```
valores_nulos = ['?']
cars = pd.read_csv('/content/Automobile_data.csv', na_values=valores_nulos)
cars.head()
```

	symboling	normalized-losses	make	fuel-type	aspiration	num-of-doors	body-style	drive-wheels
0	3	NaN	alfa-romero	gas	std	two	convertible	rwd
1	3	NaN	alfa-romero	gas	std	two	convertible	rwd
2	1	NaN	alfa-romero	gas	std	two	hatchback	rwd
3	2	164.0	audi	gas	std	four	sedan	fwd
4	2	164.0	audi	gas	std	four	sedan	4wd

Después utilizamos una función del mismo pandas que nos ayuda a contar los números nulos dentro de un arreglo, en este caso nos muestra de cada columna sus valores nulos, podemos ver que la característica “normalized-losses” tiene el mayor número de valores erróneos.

```
cars.isna().sum()
```

```
symboling            0
normalized-losses    41
make                 0
fuel-type            0
aspiration           0
num-of-doors         2
body-style           0
drive-wheels         0
engine-location      0
wheel-base           0
length               0
width                0
height               0
curb-weight          0
engine-type          0
num-of-cylinders     0
engine-size          0
fuel-system          0
bore                 4
stroke               4
compression-ratio    0
horsepower           2
peak-rpm             2
city-mpg             0
highway-mpg          0
price                4
dtype: int64
```

Ahora graficamos con plt estos valores erróneos en una gráfica de barras para ver desde otra forma estos valores.

```
plt.subplots(0,0, figsize = (18,5))
ax = (cars.isnull().sum()).sort_values(ascending = False).plot.bar(color = 'blue')
plt.title('Valores perdidos por columna', fontsize = 20);
```

Ahora haremos el proceso correspondiente para poder eliminarlos o en este caso, utilizar un modelo estadístico para poder intentar determinar su valor con una regresión. Para poder realizar lo anterior existe el método interpolate de la propiedad de una columna seleccionada

previamente, este utiliza una propiedad llamada method para seleccionar el tipo de regresión que se utilizará y el limit_direction que utilizaremos “both”.

```
cars['normalized-losses'] = cars['normalized-losses'].interpolate(method="linear", limit_direction="both")
cars['price'] = cars['price'].interpolate(method="linear", limit_direction="both")
cars['stroke'] = cars['stroke'].interpolate(method="linear", limit_direction="both")
cars['bore'] = cars['bore'].interpolate(method="linear", limit_direction="both")
cars['peak-rpm'] = cars['peak-rpm'].interpolate(method="linear", limit_direction="both")
cars['horsepower'] = cars['horsepower'].interpolate(method="linear", limit_direction="both")
cars['num-of-doors'] = cars['num-of-doors'].fillna('four')
```

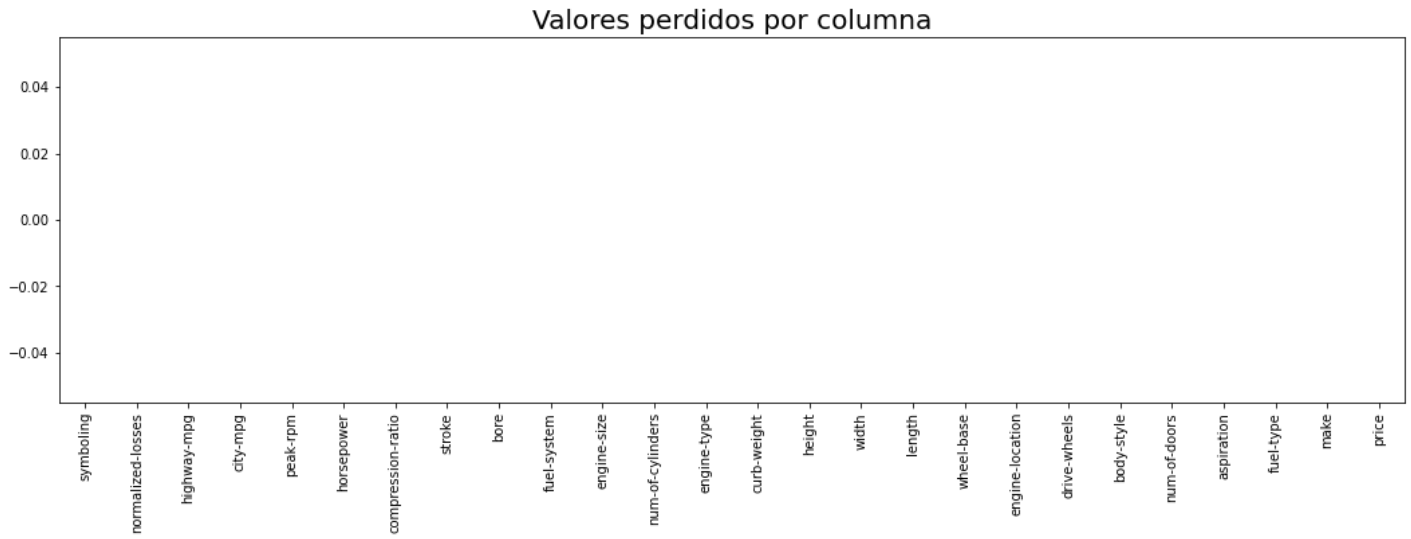
Volvemos a observar los valores nulos y podemos ver como el conjunto de datos ya no tiene ningún valor nulo.

```
cars.isna().sum()
```

```
symboling           0
normalized-losses   0
make                0
fuel-type           0
aspiration          0
num-of-doors        0
body-style          0
drive-wheels        0
engine-location     0
wheel-base          0
length              0
width               0
height              0
curb-weight         0
engine-type         0
num-of-cylinders    0
engine-size         0
fuel-system         0
bore                0
stroke              0
compression-ratio   0
horsepower          0
peak-rpm            0
city-mpg            0
highway-mpg         0
price               0
dtype: int64
```

Podemos volver a observar esto en nuestra gráfica, no se crea ninguna barra por lo mismo, observamos que no existe ningún valor nulos.

```
plt.subplots(0,0, figsize = (18,5))
ax = (cars.isnull().sum()).sort_values(ascending = False).plot.bar(color = 'blue')
plt.title('Valores perdidos por columna', fontsize = 20);
```



Para poder ejecutar un algoritmo de regresión como es en este caso, se tendrá que utilizar los valores continuos, en este caso se encuentran bastantes datos de los cuales tendremos que cambiar con los siguientes códigos. Desde la librería de sklearn utilizamos el método LabelEncoder() que nos ayuda a encontrar estos valores categóricos y los convierte en continuos.

```
leMake = preprocessing.LabelEncoder()
leMake.fit(cars["make"])
cars["make"] = leMake.transform(cars["make"])+1
leFT = preprocessing.LabelEncoder()
leFT.fit(cars["fuel-type"])
cars["fuel-type"] = leFT.transform(cars["fuel-type"])+1
leA = preprocessing.LabelEncoder()
leA.fit(cars["aspiration"])
cars["aspiration"] = leA.transform(cars["aspiration"])+1
leNOD = preprocessing.LabelEncoder()
leNOD.fit(cars["num-of-doors"])
cars["num-of-doors"] = leNOD.transform(cars["num-of-doors"])+1
leBS = preprocessing.LabelEncoder()
leBS.fit(cars["body-style"])
cars["body-style"] = leBS.transform(cars["body-style"])+1
leDW = preprocessing.LabelEncoder()
leDW.fit(cars["drive-wheels"])
cars["drive-wheels"] = leDW.transform(cars["drive-wheels"])+1
leEL = preprocessing.LabelEncoder()
leEL.fit(cars["engine-location"])
cars["engine-location"] = leEL.transform(cars["engine-location"])+1
leET = preprocessing.LabelEncoder()
leET.fit(cars["engine-type"])
cars["engine-type"] = leET.transform(cars["engine-type"])+1
leNC = preprocessing.LabelEncoder()
leNC.fit(cars["num-of-cylinders"])
cars["num-of-cylinders"] = leNC.transform(cars["num-of-cylinders"])+1
leFS = preprocessing.LabelEncoder()
leFS.fit(cars["fuel-system"])
cars["fuel-system"] = leFS.transform(cars["fuel-system"])+1
```

Ahora para cuestiones de cualquier proceso de ciencias de datos en una tarea de aprendizaje supervisado, se tendrá que dividir el conjunto de datos en la variable dependiente y la variable independiente.

```
x = cars.drop('price', axis=1)
y = cars['price']
```

Ahora dividimos este conjunto de datos de datos en los valores de entrenamiento y los valores de prueba, utilizamos el 0.7 para dividir entre 70% y un 30%, estos valores serán para el conjunto de entrenamiento y de prueba respectivamente.

```
x_train, x_test, y_train, y_test = train_test_split(np.array(x), np.array(y), test_size=0.7, random_state=209)
```

Ahora utilizaremos el modelo de regresión lineal para poder entrenarlo con los datos de los autos y sus respectivos precios.

```
regression = LinearRegression()
regression.fit(x_train, y_train)
```

La misma función que nos permite entrenar estos valores, también nos ayuda a conocer el score que se encontró.

```
regression.score(x_test, y_test)
```

Y recordando que también que en la propuesta de la solución se pedía utilizar el árbol de regresión, también se realizará.

```
DTregression = DecisionTreeRegressor()
DTregression.fit(x_train, y_train)
```

La misma función que en la regresión lineal, se usará el score para conocer su rendimiento.

```
DTregression.score(x_test, y_test)
```

RESULTADOS

Los resultados son casi similares y son los siguientes.

Para la regresión lineal, podemos observar que en rendimiento al utilizar el método proveniente de la misma librería del modelo estadístico, se encuentra muy alto. Se encontró un 85% de precisión.

```
regression.score(x_test, y_test)
```

```
0.8577789918774089
```

También podemos observar el conjunto de datos entrenado con nuestro modelo estadístico de árbol de regresión, el cual se encontró un buen porcentaje de precisión, aunque no sea mejor que el de la regresión lineal, nos podría servir en dado caso de que no hayamos encontrado el anterior. Por lo tanto, podemos observar que el árbol de regresión obtuvo un 80%.

```
DTregression.score(x_test, y_test)
```

```
0.8028089324967764
```

CONCLUSIONES

En conclusión, este proyecto me ha ayudado a conocer un repositorio de datos donde la gente lo puede utilizar como un portafolio de evidencias para poder presentarlo ante una empresa. Esto con el propósito de demostrar nuestras habilidades en ciertas áreas. Por lo tanto, con esta práctica, me será de gran utilidad para poder demostrar mis conocimientos sobre el área.

También hemos conocido como realizar un proyecto documentado desde inicio a fin, este proyecto que tuvo la finalidad de conocer los precios de autos, suena tan sencillo, sin embargo, la dificultad del conjunto de datos se me presentó en solamente la transformación de esta, en la limpieza de este. Me reforzó la idea de que la tarea más importante en un proyecto de ciencias de datos es el transformado de los mismos.

Finalmente, reconocer que no existe un solo modelo estadístico que sea capaz de generalizar la información que se le presenta, sino, tener en cuenta que existen diversos modelos que trabajando en conjunto podemos seleccionar el que mejor rendimiento nos presente.

Agradezco este curso ya que me ayudó a reforzar conocimiento que aprendimos anteriormente como las métricas y nuevos modelos estadísticos, sino que, además nos ayudó a conocer nuevos de estos. Me ayudó a conocer los pequeños detalles de los nuevos modelos estadísticos siendo capaz de transmitir conocimiento para poder utilizarlos adecuadamente en nuestra labor como científico de datos.

REFERENCIAS

¿Qué es el aprendizaje supervisado? (s. f.). TIBCO Software. https://www.tibco.com/es/reference-center/what-is-supervised-learning

¿Qué es la regresión lineal? (s. f.). MATLAB & Simulink. https://la.mathworks.com/discovery/linear-regression.html

UCI Machine Learning Repository: Automobile Data Set. (s. f.). http://archive.ics.uci.edu/ml/datasets/Automobile