**CAR PRICE PREDICTION**

The price of a car depends on a lot of factors like the goodwill of the brand of the car, features of the car, horsepower and the mileage it gives and many more. The dataset for the used car prices is attached with this.

Split the dataset into training and testing dataset.

Develop and train the linear regression model using the training dataset.

Test the model with the test dataset.

Predict the price of a car.

**Code with respective output:**

import numpy as nm

import matplotlib.pyplot as plt

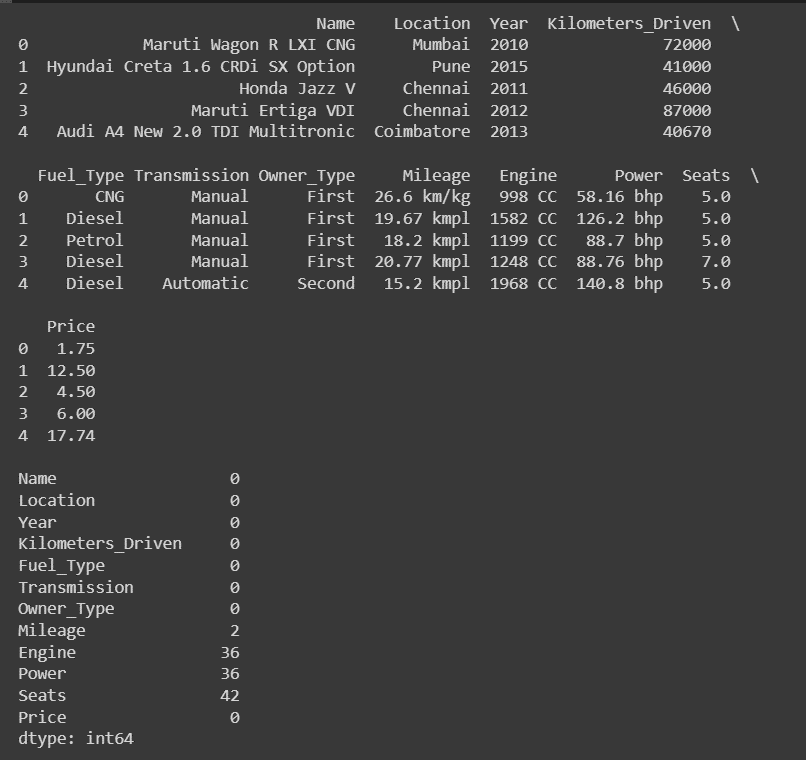
import pandas as pd

df = pd.read\_csv('cp.csv')

print(df.head(5))

#Checking for Null values

print(df.isnull().sum())

****

#removing CC from engine column

df['Engine'] = df['Engine'].apply(lambda x: str(x).replace('CC', '') if 'CC' in str(x) else str(x))

#removing kmpl and km/kg from mileage column

df['Mileage'] = df['Mileage'].apply(lambda x: str(x).replace('kmpl', '') if 'kmpl' in str(x) else str(x))

df['Mileage'] = df['Mileage'].apply(lambda x:str(x).replace('km/kg', '') if 'km/kg' in str(x) else str(x))

#removing bhp from power column

df['Power'] = df['Power'].apply(lambda x: str(x).replace('bhp', '') if 'bhp' in str(x) else str(x))

#Converting the three columns to numeric type:

df['Mileage'] = pd.to\_numeric(df['Mileage'], errors='coerce')

df['Engine'] = pd.to\_numeric(df['Engine'], errors='coerce')

df['Power'] = pd.to\_numeric(df['Power'], errors='coerce')

#Replacing null values by mode

x = df['Seats'].mode()[0]

df['Seats'].fillna(x, inplace = True)

x = df['Mileage'].mode()[0]

df['Mileage'].fillna(x, inplace = True)

x = df['Engine'].mode()[0]

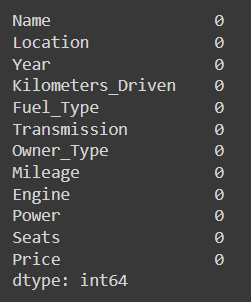
df['Engine'].fillna(x, inplace = True)

x = df['Power'].mode()[0]

df['Power'].fillna(x, inplace = True)

#Checking for Null Values

print(df.isnull().sum())

****

#Separating the brand names from the column

df['Brand\_Name'] = df['Name'].str.split(' ').str[0]

df.groupby('Brand\_Name').nunique()





#Dropping the Name and Location column

dfnew = df.drop(["Name","Location"],axis='columns')

dfnew.head(5)

****

#creating a new dataframe

dfn = dfnew.copy()

#Converting categorical columns to numeric columns

from sklearn import preprocessing

le\_Fuel\_Type=preprocessing.LabelEncoder()

le\_Transmission=preprocessing.LabelEncoder()

le\_Owner\_Type=preprocessing.LabelEncoder()

le\_Brand\_Name=preprocessing.LabelEncoder()

dfn['Fuel\_Type\_n']= le\_Fuel\_Type.fit\_transform(dfn['Fuel\_Type'])

dfn['Transmission\_n']=le\_Transmission.fit\_transform(dfn['Transmission'])

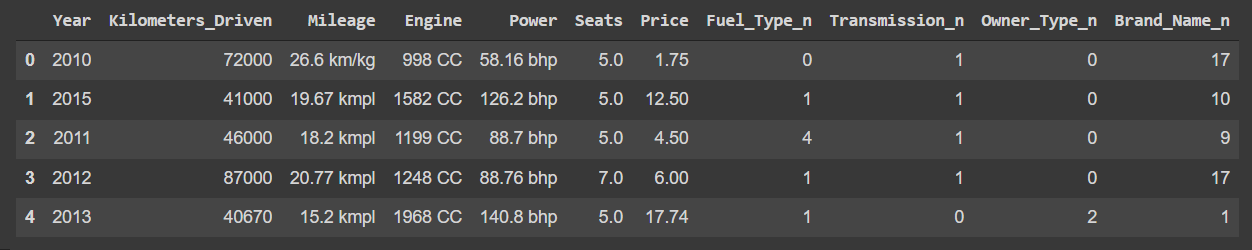
dfn['Owner\_Type\_n']=le\_Owner\_Type.fit\_transform(dfn['Owner\_Type'])

dfn['Brand\_Name\_n']=le\_Brand\_Name.fit\_transform(dfn['Brand\_Name'])

#dropping columns-Fuel Type, Transmission, Owner type and Brand Name

dfn=dfn.drop(["Fuel\_Type","Transmission","Owner\_Type","Brand\_Name"],axis='columns')

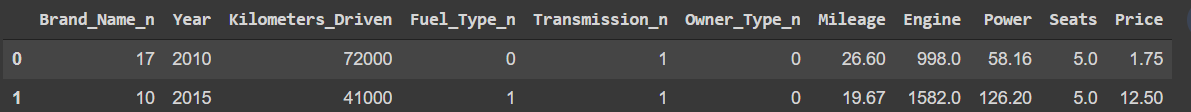
dfn.head(5)

****

#Rearranging the column names accordingly

dfn=dfn[['Brand\_Name\_n','Year','Kilometers\_Driven','Fuel\_Type\_n','Transmission\_n','Owner\_Type\_n','Mileage','Engine','Power','Seats','Price']]

dfn.head(2)

****

#Extracting and spliting the data

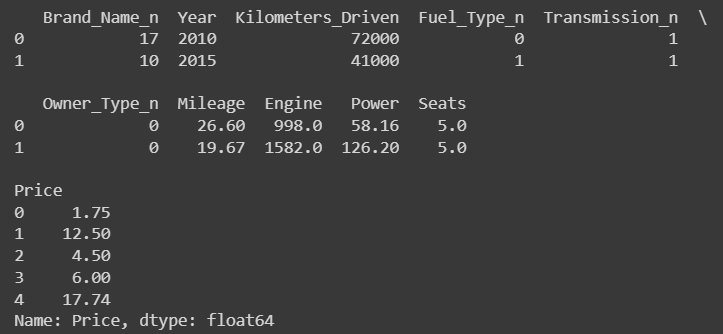
dfinput = dfn.drop(["Price"], axis='columns')

print(dfinput.head(2))

print("Price")

df3 = dfn['Price']

df3.head(5)

****

#Spliting data into test and train

from sklearn.model\_selection import train\_test\_split

x\_train, x\_test, y\_train, y\_test = train\_test\_split(dfinput, df3, test\_size = 0.25, random\_state = 0)

#Training the data using Linear Regression

from sklearn.linear\_model import LinearRegression

r = LinearRegression()

r.fit(x\_train, y\_train)

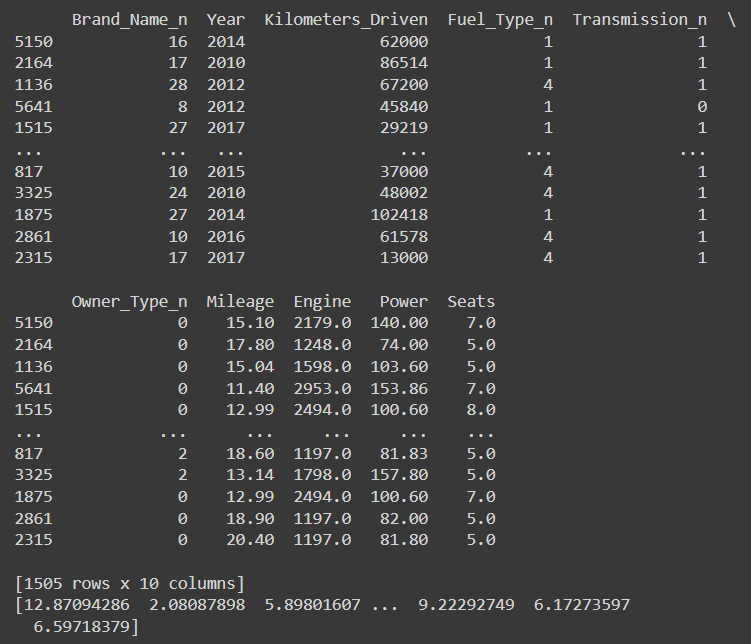
****

#Testing data

y\_pred= r.predict(x\_test)

print(x\_test)

print(y\_pred)

****

"""

Predicting data for

17- Maruti (Brand Name)

2013(Year)

10000(KM drove)

3-Petrol (Fuel Type)

1-Manual (Transmission)

0-first hand (Owner Type)

16(Mileage km/pl)

2494(Engine in CC)

100(Power in bhp)

5.0(no of seats)

"""

y\_pred = r.predict([[17, 2013, 10000, 3, 1, 0, 16, 2494, 100, 5.0]])

print("Price:", y\_pred, "lakh")

print()

