Student Name: Tushar Panchal

En. No. : 21162101014

Branch: CBA

Batch: 71

Subject: ML (Machine Learning)

EXPERIMENT 1

Exploratory data analysis in car price prediction dataset

1. **Import dataset available at following url:**

https://archive.ics.uci.edu/ml/machine-learning-databases/autos/imports-85.data

```
import pandas as pd
path = "https://archive.ics.uci.edu/ml/machine-learning-databases/autos/imports-85.data"
df = pd.read_csv(path)
print("The First 5 rows of the dataframe")
headers = ["symboling","normalized-losses","make","fuel-type","aspiration", "num-of-doors","body-style",
         "drive-wheels","engine-location","wheel-base", "length","width","height","curb-weight","engine-type",
         "num-of-cylinders", "engine-size","fuel-system","bore","stroke","compression-ratio","horsepower",
         "peak-rpm","city-mpg","highway-mpg","price"]
df.columns = headers
df.head(5)
```

```
The First 5 rows of the dataframe
```

	symboling	normalized-losses	make	fuel-type	aspiration	num-of-doors	body-style	drive-wheels	engine-location	wheel-base	...	engine-size	fuel-system	bore	stroke
0	3	?	alfa-romero	gas	std	two	convertible	rwd	front	88.6	...	130	mpfi	3.47	2.68
1	1	?	alfa-romero	gas	std	two	hatchback	rwd	front	94.5	...	152	mpfi	2.68	3.47
2	2	164	audi	gas	std	four	sedan	fwd	front	99.8	...	109	mpfi	3.19	3.40
3	2	164	audi	gas	std	four	sedan	4wd	front	99.4	...	136	mpfi	3.19	3.40
4	2	?	audi	gas	std	two	sedan	fwd	front	99.8	...	136	mpfi	3.19	3.40

5 rows × 26 columns

2. **Clean dataset.(hint: Replace missing values):**

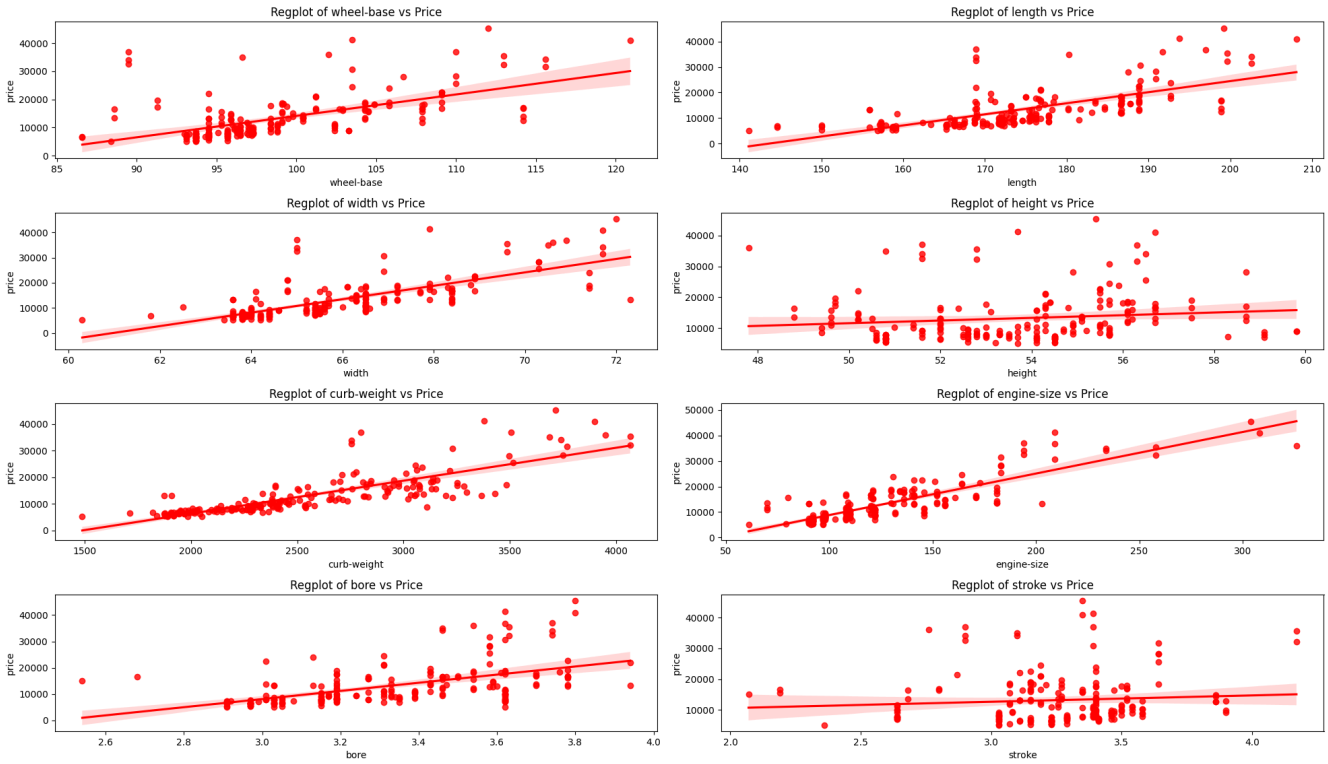
```
import pandas as pd
path = "https://archive.ics.uci.edu/ml/machine-learning-databases/autos/imports-85.data"
df = pd.read_csv(path, header=None, na_values="?")
headers=["symboling","normalized-losses","make","fuel-type","aspiration", "num-of-doors","body-style",
           "drive-wheels","engine-location","wheel-base", "length","width","height","curb-weight","engine-type",
           "num-of-cylinders", "engine-size","fuel-system","bore","stroke","compression-ratio","horsepower",
           "peak-rpm","city-mpg","highway-mpg","price"]
df.columns = headers
numeric_columns = df.select_dtypes(include=['float64', 'int64']).columns
df[numeric_columns] = df[numeric_columns].interpolate(method='linear',limit_direction='forward',axix=0)
categorical_columns = df.select_dtypes(include=['object']).columns
for column in categorical_columns:
  df[column].fillna(df[column].mode()[0], inplace=True)
print("\nNumber of missing values after Cleaning:")
print(df.isnull().sum())
print("\nThe first 10 rows of the cleaned dataframe:")
print(df.head(50))
```

```
Number of missing values after Cleaning:
symboling           0
normalized-losses    3
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

The first 10 rows of the cleaned dataframe:
    symboling  normalized-losses         make fuel-type aspiration  \
0           3                NaN  alfa-romero       gas        std
1           3                NaN  alfa-romero       gas        std
2           1                NaN  alfa-romero       gas        std
3           2         164.000000         audi       gas        std
4           2         164.000000         audi       gas        std
5           2         161.000000         audi       gas        std
6           1         158.000000         audi       gas        std
7           1         158.000000         audi       gas        std
8           1         158.000000         audi       gas      turbo
9           0         175.000000         audi       gas      turbo
10          2         192.000000          bmw       gas        std
11          0         192.000000          bmw       gas        std
12          0         188.000000          bmw       gas        std
13          0         188.000000          bmw       gas        std
14          1         174.600000          bmw       gas        std
15          0         161.200000          bmw       gas        std
16          0         147.800000          bmw       gas        std
17          0         134.400000          bmw       gas        std
18          2         121.000000    chevrolet       gas        std
19          1          98.000000    chevrolet       gas        std
20          0          81.000000    chevrolet       gas        std
21          1         118.000000        dodge       gas        std
22          1         118.000000        dodge       gas        std
23          1         118.000000        dodge       gas      turbo
24          1         148.000000        dodge       gas        std
25          1         148.000000        dodge       gas        std
```

3. **Feature extraction using visualization (hint: matplotlib or seaborn graphs)Only regplots of continuous attributes**

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

```
path = "https://archive.ics.uci.edu/ml/machine-learning-databases/autos/imports-85.data"
df = pd.read_csv(path, header=None, na_values="?")

headers = ["symboling","normalized-losses","make","fuel-type","aspiration", "num-of-doors","body-style",
           "drive-wheels","engine-location","wheel-base", "length","width","height","curb-weight","engine-type",
           "num-of-cylinders", "engine-size","fuel-system","bore","stroke","compression-ratio","horsepower",
           "peak-rpm","city-mpg","highway-mpg","price"]

df.columns = headers

numeric_columns = df.select_dtypes(include=['float64', 'int64']).columns
for column in numeric_columns:
    df[column].fillna(df[column].mean(), inplace=True)

categorical_columns = df.select_dtypes(include=['object']).columns
for column in categorical_columns:
    df[column].fillna(df[column].mode()[0], inplace=True)

continuous_attributes = ["wheel-base", "length", "width", "height", "curb-weight", "engine-size",
                         "bore", "stroke", "compression-ratio", "horsepower", "peak-rpm", "city-mpg", "highway-mpg", "price"]

plt.figure(figsize=(20, 20))
for i, attribute in enumerate(continuous_attributes):
    plt.subplot(7, 2, i + 1)
    sns.regplot(x=attribute, y='price', data=df,color='#FF0000')
    plt.title(f'Regplot of {attribute} vs Price')

plt.tight_layout()
plt.show()
```


Regplot of wheel-base vs Price
price
wheel-base
Regplot of length vs Price
price
length
Regplot of width vs Price
price
width
Regplot of height vs Price
price
height
Regplot of curb-weight vs Price
price
curb-weight
Regplot of engine-size vs Price
price
engine-size
Regplot of bore vs Price
price
bore
Regplot of stroke vs Price
price
stroke