

# **Car Price Prediction using Machine Learning**

## **Code Implementation**

### **Import Various Libraries**

```
import datetime
import pandas as pd
import numpy as np
import math
from sklearn.model_selection import train_test_split
from matplotlib import pyplot as plt
from pandas import DataFrame
```

### **Import Dataset**

```
train = pd.read_excel('/content/Data_Train.xlsx')
```

### **Cleaning the Dataset**

**Returns the number of missing values in the data set**

```
train.isnull().sum()
```

```
train.describe()
print(train['Location'].unique())
print(train['Fuel_Type'].unique())
print(train['Transmission'].unique())
print(train['Owner_Type'].unique())
```

**Let's Drop some Rows which contain NULL values.**

```
print("Shape of train data Before dropping any Row: ",train.shape)
```

```
train = train[train['Mileage'].notna()]
```

```
print("Shape of train data After dropping Rows with NULL values in Mileage:  
",train. shape)
```

```
train = train[train['Engine'].notna()]
```

```
print("Shape of train data After dropping Rows with NULL values in Engine :  
",train. shape)
```

```
train = train[train['Power'].notna()]
```

```
print("Shape of train data After dropping Rows with NULL values in Power :  
",train. shape)
```

```
train = train[train['Seats'].notna()]
```

```
print("Shape of train data After dropping Rows with NULL values in Seats :  
",train.shape)
```

**Bringing Company and model column to the beginning**

```
Col_name = 'Model'
```

```
First_col = train.pop(col_name)
```

```
train.insert(0,col_name,first_col)
```

```
col_name='Brand'
```

```
First_col = train.pop(col_name)
```

## Converting mileage and engine values to float

```
train['Mileage'] = train['Mileage'].astype(float)
```

```
train['Engine'] = train['Engine'].astype(float)
```

## Data Visualization

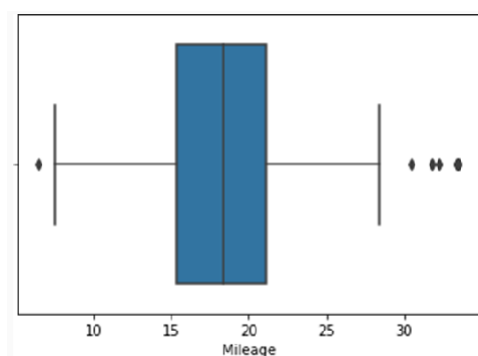
### Data Visualization Concept

Data Visualization is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data Visualization tools provide an accessible way to see and understand trends, outliers, and patterns in data. Additionally, it provides an excellent way for employees or business owners to present data to non-technical audiences without confusion.

### Data Visualization Code and Graphs

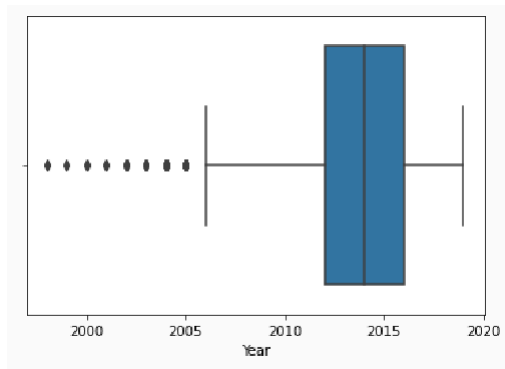
```
import seaborn as sns
```

```
sns.boxplot(x=train['Mileage'])
```



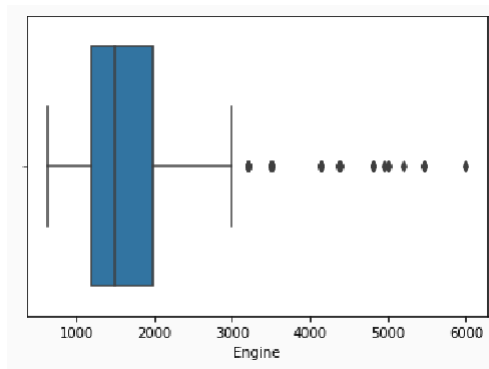
**Figure 10 : Mileage Visualization**

```
sns.boxplot(x=train['Year'])
```



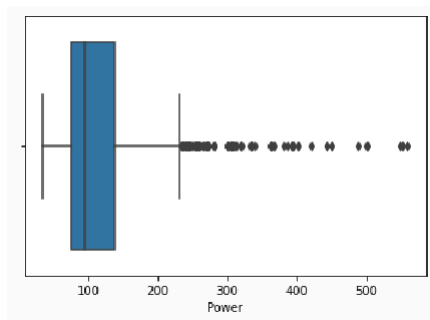
**Figure 11 : Year Visualization**

```
sns.boxplot(x=train['Engine'])
```



**Figure 12 : Engine Visualization**

```
sns.boxplot(x=train['Power'])
```



**Figure 13 : Power Visualization**

## EXPLORATORY DATA ANALYSIS

`train.dtypes`

▶	<code>train.dtypes</code>
Brand	object
Model	object
Location	object
Year	int64
Kilometers_Driven	int64
Fuel_Type	object
Transmission	object
Owner_Type	object
Mileage	float64
Engine	float64
Power	float64
Seats	float64
Price	float64
dtype: object	

`train.isnull().sum()`

▶	<code>train.isnull().sum()</code>
Brand	0
Model	0
Location	0
Year	0
Kilometers_Driven	0
Fuel_Type	0
Transmission	0
Owner_Type	0
Mileage	0
Engine	0
Power	0
Seats	0
Price	0
dtype: int64	

## Pie Chart Visualization

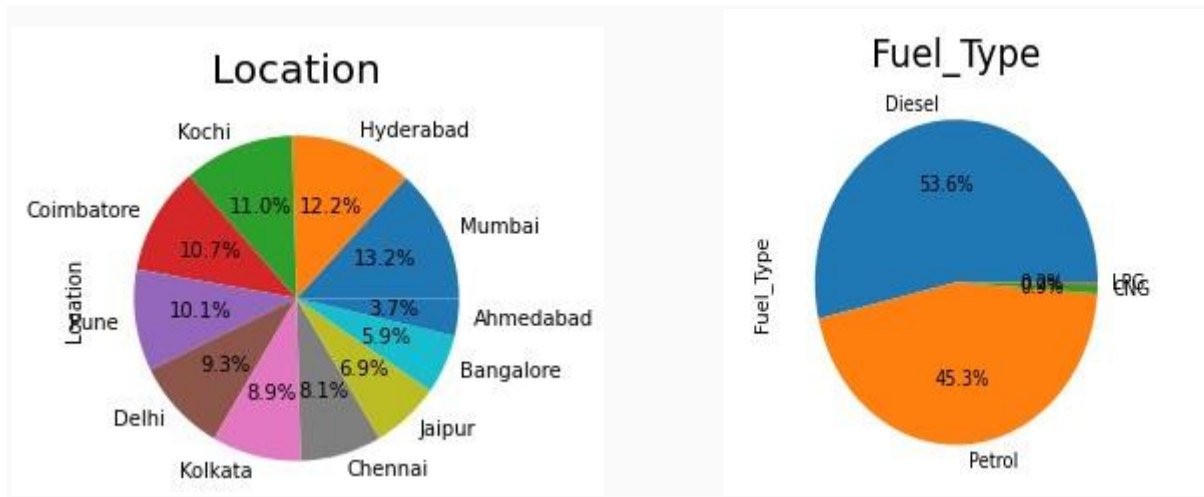


Figure 14 : Visualization

Figure 15 : Visualization

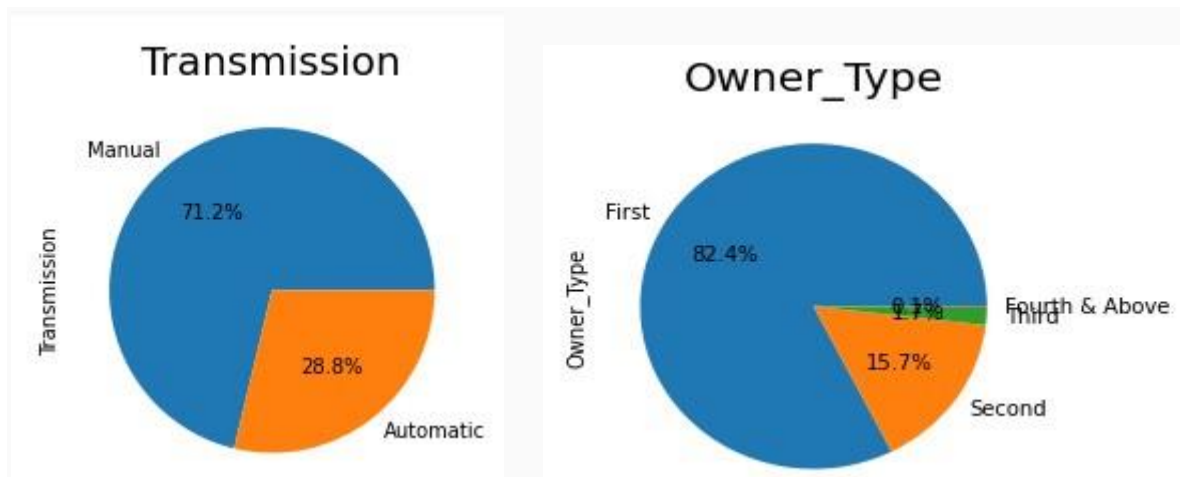


Figure 16 : Visualization

Figure 17 : Visualization

Bar GraphVisualization

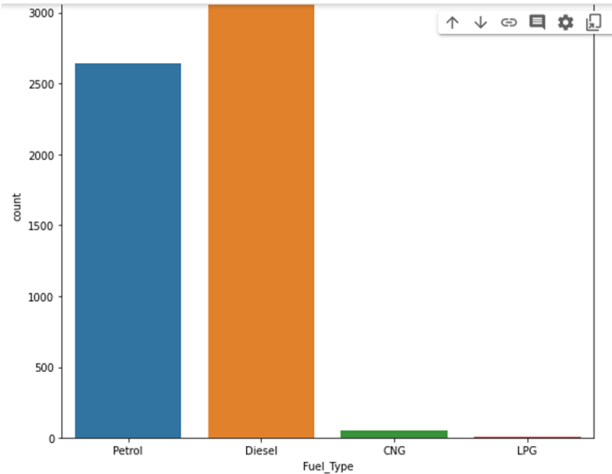


Figure 18 : Fuel Type Bar Graph

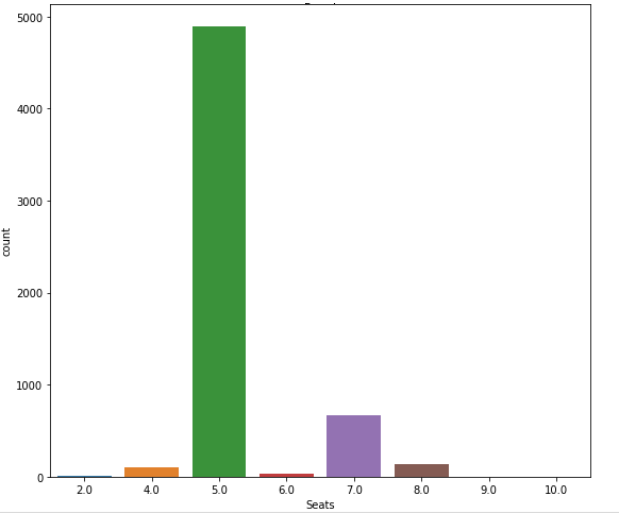


Figure 19 : Seats in Bar Graph Graph

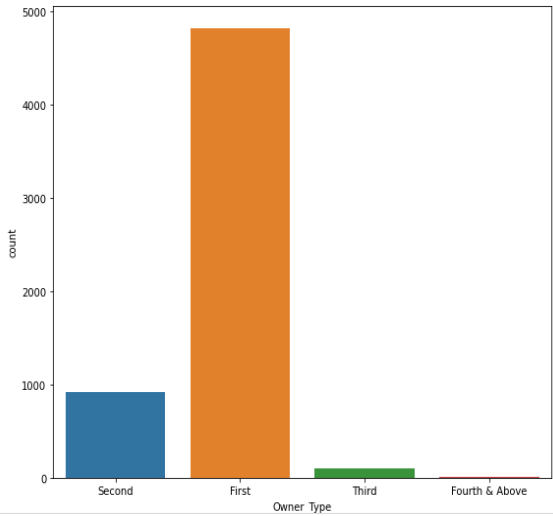


Figure 20 : Owner Type Bar