In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

In [2]:
```
df = pd.read_csv('C:\\Users\\Atif\\Desktop\\python learning\\Car_details_v3.csv')
df.head()
```

Out[2]:

	name	year	selling_price	km_driven	fuel	seller_type	transmission	owner	mileage	engine	max_pc
0	Maruti Swift Dzire VDI	2014	450000	145500	Diesel	Individual	Manual	First Owner	23.4 kmpl	1248 CC	74
1	Skoda Rapid 1.5 TDI Ambition	2014	370000	120000	Diesel	Individual	Manual	Second Owner	21.14 kmpl	1498 CC	103.52
2	Honda City 2017-2020 EXi	2006	158000	140000	Petrol	Individual	Manual	Third Owner	17.7 kmpl	1497 CC	78
3	Hyundai i20 Sportz Diesel	2010	225000	127000	Diesel	Individual	Manual	First Owner	23.0 kmpl	1396 CC	90
4	Maruti Swift VXI BSIII	2007	130000	120000	Petrol	Individual	Manual	First Owner	16.1 kmpl	1298 CC	88.2

In [3]:
```
df.shape
```

Out[3]:
```
(8128, 13)
```

In [4]:
```
df.groupby('name')['name'].agg('count')
```

Out[4]:
```
name
Ambassador CLASSIC 1500 DSL AC        1
Ambassador Classic 2000 DSZ AC PS     1
Ambassador Grand 1500 DSZ BSIII       1
Ambassador Grand 2000 DSZ PW CL       1
Ashok Leyland Stile LE                1
                                     ..
Volvo V40 D3 R-Design                29
Volvo XC40 D4 Inscription BSIV       30
Volvo XC40 D4 R-Design                2
Volvo XC60 Inscription D5 BSIV        1
Volvo XC90 T8 Excellence BSIV         1
Name: name, Length: 2058, dtype: int64
```

In [5]:
```
df1 = df.drop(['torque'],axis='columns')
```

In [6]:
```
df1.isnull().sum()
```

Loading [MathJax]/extensions/Safe.js

Out[6]:
```
name              0
year              0
selling_price     0
km_driven         0
fuel              0
seller_type       0
transmission      0
owner             0
mileage         221
engine          221
max_power       215
seats           221
dtype: int64
```

In [7]:
```
df2 = df1.dropna()
df2.isnull().sum()
```

Out[7]:
```
name            0
year            0
selling_price   0
km_driven       0
fuel            0
seller_type     0
transmission    0
owner           0
mileage         0
engine          0
max_power       0
seats           0
dtype: int64
```

In [8]:
```
print(df2.fuel.value_counts())
print(df2.seller_type.value_counts())
print(df2.transmission.value_counts())
print(df2.owner.value_counts())
```

```
Diesel    4299
Petrol    3520
CNG         53
LPG         35
Name: fuel, dtype: int64
Individual          6564
Dealer              1107
Trustmark Dealer     236
Name: seller_type, dtype: int64
Manual       6866
Automatic    1041
Name: transmission, dtype: int64
First Owner             5215
Second Owner            2017
Third Owner              510
Fourth & Above Owner     160
Test Drive Car             5
Name: owner, dtype: int64
```

In [9]:
```
#Encoding columns of fuel,seller_type,transmission,owner
df2.replace({'fuel':{'Diesel':0,'Petrol':1,'CNG':2,'LPG':3}},inplace=True)
df2.replace({'seller_type':{'Individual':0,'Dealer':1,'Trustmark Dealer':2}},inplace=Tru
df2.replace({'transmission':{'Manual':0,'Automatic':1}},inplace=True)
df2.replace({'owner':{'First Owner':0,'Second Owner':1,'Third Owner':2,'Fourth & Above O
```

Loading [MathJax]/extensions/Safe.js

```
C:\Users\Atif\AppData\Local\Temp\ipykernel_6460\3100669488.py:2: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
  df2.replace({'fuel':{'Diesel':0,'Petrol':1,'CNG':2,'LPG':3}},inplace=True)
C:\Users\Atif\AppData\Local\Temp\ipykernel_6460\3100669488.py:3: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
  df2.replace({'seller_type':{'Individual':0,'Dealer':1,'Trustmark Dealer':2}},inplace=T
rue)
C:\Users\Atif\AppData\Local\Temp\ipykernel_6460\3100669488.py:4: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
  df2.replace({'transmission':{'Manual':0,'Automatic':1}},inplace=True)
C:\Users\Atif\AppData\Local\Temp\ipykernel_6460\3100669488.py:5: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
  df2.replace({'owner':{'First Owner':0,'Second Owner':1,'Third Owner':2,'Fourth & Above
Owner':3,'Test Drive Car':4}},inplace=True)
```

In [10]:

```
df2['CC'] = df2['engine'].apply(lambda x: int(x.split(' ')[0]))
df2.head()
```

```
C:\Users\Atif\AppData\Local\Temp\ipykernel_6460\940960152.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
  df2['CC'] = df2['engine'].apply(lambda x: int(x.split(' ')[0]))
```

Loading [MathJax]/extensions/Safe.js

Out[10]:

	name	year	selling_price	km_driven	fuel	seller_type	transmission	owner	mileage	engine	max_powe
0	Maruti Swift Dzire VDI	2014	450000	145500	0	0	0	0	23.4 kmpl	1248 CC	74 bh
1	Skoda Rapid 1.5 TDI Ambition	2014	370000	120000	0	0	0	1	21.14 kmpl	1498 CC	103.52 bh
2	Honda City 2017-2020 EXi	2006	158000	140000	1	0	0	2	17.7 kmpl	1497 CC	78 bh
3	Hyundai i20 Sportz Diesel	2010	225000	127000	0	0	0	0	23.0 kmpl	1396 CC	90 bh
4	Maruti Swift VXI BSIII	2007	130000	120000	1	0	0	0	16.1 kmpl	1298 CC	88.2 bh

In [11]:
```
df2.head()
```

Out[11]:

	name	year	selling_price	km_driven	fuel	seller_type	transmission	owner	mileage	engine	max_powe
0	Maruti Swift Dzire VDI	2014	450000	145500	0	0	0	0	23.4 kmpl	1248 CC	74 bh
1	Skoda Rapid 1.5 TDI Ambition	2014	370000	120000	0	0	0	1	21.14 kmpl	1498 CC	103.52 bh
2	Honda City 2017-2020 EXi	2006	158000	140000	1	0	0	2	17.7 kmpl	1497 CC	78 bh
3	Hyundai i20 Sportz Diesel	2010	225000	127000	0	0	0	0	23.0 kmpl	1396 CC	90 bh
4	Maruti Swift VXI BSIII	2007	130000	120000	1	0	0	0	16.1 kmpl	1298 CC	88.2 bh

In [12]:
```
df2.mileage.dtypes
```

Out[12]:
```
dtype('O')
```

In [13]:
```
df2['kmpl'] = df2['mileage'].str.replace('kmpl','').str.replace(',','')
```

Loading [MathJax]/extensions/Safe.js

```
C:\Users\Atif\AppData\Local\Temp\ipykernel_6460\1799905516.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
  df2['kmpl'] = df2['mileage'].str.replace('kmpl','').str.replace(',','')
```

In [14]:
```
df2.head()
```

Out[14]:

	name	year	selling_price	km_driven	fuel	seller_type	transmission	owner	mileage	engine	max_powe
0	Maruti Swift Dzire VDI	2014	450000	145500	0	0	0	0	23.4 kmpl	1248 CC	74 bh
1	Skoda Rapid 1.5 TDI Ambition	2014	370000	120000	0	0	0	1	21.14 kmpl	1498 CC	103.52 bh
2	Honda City 2017-2020 EXi	2006	158000	140000	1	0	0	2	17.7 kmpl	1497 CC	78 bh
3	Hyundai i20 Sportz Diesel	2010	225000	127000	0	0	0	0	23.0 kmpl	1396 CC	90 bh
4	Maruti Swift VXI BSIII	2007	130000	120000	1	0	0	0	16.1 kmpl	1298 CC	88.2 bh

In [15]:
```
df2['bhp'] = df2['max_power'].str.replace('bhp','').str.replace(',','')
```

```
C:\Users\Atif\AppData\Local\Temp\ipykernel_6460\3703754681.py:1: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

See the caveats in the documentation: https://pandas.pydata.org/pandas-docs/stable/user_
guide/indexing.html#returning-a-view-versus-a-copy
  df2['bhp'] = df2['max_power'].str.replace('bhp','').str.replace(',','')
```

In [16]:
```
df2.head()
```

Loading [MathJax]/extensions/Safe.js

Out[16]:

	name	year	selling_price	km_driven	fuel	seller_type	transmission	owner	mileage	engine	max_powe
0	Maruti Swift Dzire VDI	2014	450000	145500	0	0	0	0	23.4 kmpl	1248 CC	74 bh
1	Skoda Rapid 1.5 TDI Ambition	2014	370000	120000	0	0	0	1	21.14 kmpl	1498 CC	103.52 bh
2	Honda City 2017-2020 EXi	2006	158000	140000	1	0	0	2	17.7 kmpl	1497 CC	78 bh
3	Hyundai i20 Sportz Diesel	2010	225000	127000	0	0	0	0	23.0 kmpl	1396 CC	90 bh
4	Maruti Swift VXI BSIII	2007	130000	120000	1	0	0	0	16.1 kmpl	1298 CC	88.2 bh

In [17]:
```
df3 = df2.drop(['mileage','engine','max_power'],axis='columns')
df3.head()
```

Out[17]:

	name	year	selling_price	km_driven	fuel	seller_type	transmission	owner	seats	CC	kmpl	bhp
0	Maruti Swift Dzire VDI	2014	450000	145500	0	0	0	0	5.0	1248	23.4	74
1	Skoda Rapid 1.5 TDI Ambition	2014	370000	120000	0	0	0	1	5.0	1498	21.14	103.52
2	Honda City 2017-2020 EXi	2006	158000	140000	1	0	0	2	5.0	1497	17.7	78
3	Hyundai i20 Sportz Diesel	2010	225000	127000	0	0	0	0	5.0	1396	23.0	90
4	Maruti Swift VXI BSIII	2007	130000	120000	1	0	0	0	5.0	1298	16.1	88.2

In [18]:
```
df3.shape
```

Out[18]:
```
(7907, 12)
```

In [19]:
```
df3.CC.dtypes
```

Out[19]:
```
dtype('int64')
```

In [20]:
```
def is_float(x):
    try:
```

Loading [MathJax]/extensions/Safe.js

```
        float(x)
    except:
        return False
    return True
```

In [21]:
```
df3[~df3['bhp'].apply(is_float)].head(10)
```

Out[21]:

name	year	selling_price	km_driven	fuel	seller_type	transmission	owner	seats	CC	kmpl	bhp

In [22]:
```
df3['bhp'] = df3['bhp'].astype(float)
df3
```

Out[22]:

	name	year	selling_price	km_driven	fuel	seller_type	transmission	owner	seats	CC	kmpl	b
0	Maruti Swift Dzire VDI	2014	450000	145500	0	0	0	0	5.0	1248	23.4	74.
1	Skoda Rapid 1.5 TDI Ambition	2014	370000	120000	0	0	0	1	5.0	1498	21.14	103.
2	Honda City 2017-2020 EXi	2006	158000	140000	1	0	0	2	5.0	1497	17.7	78.
3	Hyundai i20 Sportz Diesel	2010	225000	127000	0	0	0	0	5.0	1396	23.0	90.
4	Maruti Swift VXI BSIII	2007	130000	120000	1	0	0	0	5.0	1298	16.1	88.
...	...	...	...	...	...	...	...	...	...	...	...	...
8123	Hyundai i20 Magna	2013	320000	110000	1	0	0	0	5.0	1197	18.5	82.
8124	Hyundai Verna CRDi SX	2007	135000	119000	0	0	0	3	5.0	1493	16.8	110.
8125	Maruti Swift Dzire ZDi	2009	382000	120000	0	0	0	0	5.0	1248	19.3	73.
8126	Tata Indigo CR4	2013	290000	25000	0	0	0	0	5.0	1396	23.57	70.
8127	Tata Indigo CR4	2013	290000	25000	0	0	0	0	5.0	1396	23.57	70.

7907 rows × 12 columns

In [23]:
```
df3['bhp'] = df3['bhp'].astype(int)
df3
```

Loading [MathJax]/extensions/Safe.js

Out[23]:

	name	year	selling_price	km_driven	fuel	seller_type	transmission	owner	seats	CC	kmpl	bhp
0	Maruti Swift Dzire VDI	2014	450000	145500	0	0	0	0	5.0	1248	23.4	74
1	Skoda Rapid 1.5 TDI Ambition	2014	370000	120000	0	0	0	1	5.0	1498	21.14	103
2	Honda City 2017-2020 EXi	2006	158000	140000	1	0	0	2	5.0	1497	17.7	78
3	Hyundai i20 Sportz Diesel	2010	225000	127000	0	0	0	0	5.0	1396	23.0	90
4	Maruti Swift VXI BSIII	2007	130000	120000	1	0	0	0	5.0	1298	16.1	88
...	...	...	...	...	...	...	...	...	...	...	...	...
8123	Hyundai i20 Magna	2013	320000	110000	1	0	0	0	5.0	1197	18.5	82
8124	Hyundai Verna CRDi SX	2007	135000	119000	0	0	0	3	5.0	1493	16.8	110
8125	Maruti Swift Dzire ZDi	2009	382000	120000	0	0	0	0	5.0	1248	19.3	73
8126	Tata Indigo CR4	2013	290000	25000	0	0	0	0	5.0	1396	23.57	70
8127	Tata Indigo CR4	2013	290000	25000	0	0	0	0	5.0	1396	23.57	70

7907 rows × 12 columns

In [24]:
```
df3['kmpl'] = df3['kmpl'].astype(float)
df3
```

Loading [MathJax]/extensions/Safe.js

Out[24]:

	name	year	selling_price	km_driven	fuel	seller_type	transmission	owner	seats	CC	kmpl	bhp
0	Maruti Swift Dzire VDI	2014	450000	145500	0	0	0	0	5.0	1248	23.40	74
1	Skoda Rapid 1.5 TDI Ambition	2014	370000	120000	0	0	0	1	5.0	1498	21.14	103
2	Honda City 2017-2020 EXi	2006	158000	140000	1	0	0	2	5.0	1497	17.70	78
3	Hyundai i20 Sportz Diesel	2010	225000	127000	0	0	0	0	5.0	1396	23.00	90
4	Maruti Swift VXI BSIII	2007	130000	120000	1	0	0	0	5.0	1298	16.10	88
...	...	...	...	...	...	...	...	...	...	...	...	...
8123	Hyundai i20 Magna	2013	320000	110000	1	0	0	0	5.0	1197	18.50	82
8124	Hyundai Verna CRDi SX	2007	135000	119000	0	0	0	3	5.0	1493	16.80	110
8125	Maruti Swift Dzire ZDi	2009	382000	120000	0	0	0	0	5.0	1248	19.30	73
8126	Tata Indigo CR4	2013	290000	25000	0	0	0	0	5.0	1396	23.57	70
8127	Tata Indigo CR4	2013	290000	25000	0	0	0	0	5.0	1396	23.57	70

7907 rows × 12 columns

In [25]:
```
df3['kmpl'] = df3['kmpl'].astype(int)
df3
```

Loading [MathJax]/extensions/Safe.js

Out[25]:

	name	year	selling_price	km_driven	fuel	seller_type	transmission	owner	seats	CC	kmpl	bhp
0	Maruti Swift Dzire VDI	2014	450000	145500	0	0	0	0	5.0	1248	23	74
1	Skoda Rapid 1.5 TDI Ambition	2014	370000	120000	0	0	0	1	5.0	1498	21	103
2	Honda City 2017-2020 EXi	2006	158000	140000	1	0	0	2	5.0	1497	17	78
3	Hyundai i20 Sportz Diesel	2010	225000	127000	0	0	0	0	5.0	1396	23	90
4	Maruti Swift VXI BSIII	2007	130000	120000	1	0	0	0	5.0	1298	16	88
...	...	...	...	...	...	...	...	...	...	...	...	...
8123	Hyundai i20 Magna	2013	320000	110000	1	0	0	0	5.0	1197	18	82
8124	Hyundai Verna CRDi SX	2007	135000	119000	0	0	0	3	5.0	1493	16	110
8125	Maruti Swift Dzire ZDi	2009	382000	120000	0	0	0	0	5.0	1248	19	73
8126	Tata Indigo CR4	2013	290000	25000	0	0	0	0	5.0	1396	23	70
8127	Tata Indigo CR4	2013	290000	25000	0	0	0	0	5.0	1396	23	70

7907 rows × 12 columns

In [26]:
```
df3.CC.describe()
```

Out[26]:
```
count    7907.000000
mean     1458.625016
std       503.916303
min       624.000000
25%      1197.000000
50%      1248.000000
75%      1582.000000
max      3604.000000
Name: CC, dtype: float64
```

In [27]:
```
min_thresold , max_thresold = df3.bhp.quantile([0.1,0.9])
min_thresold , max_thresold
```

Out[27]:
```
(60.0, 140.0)
```

Loading [MathJax]/extensions/Safe.js

In [28]:
```
df4 = df3[(df3.bhp < max_thresold) & (df3.bhp > min_thresold)]
df4.shape
```

Out[28]:
```
(6231, 12)
```

In [29]:
```
min_thrsold , max_thrsold = df4.kmpl.quantile([0.01,0.99])
min_thrsold , max_thrsold
```

Out[29]:
```
(11.0, 28.0)
```

In [30]:
```
df5 = df4[(df4.kmpl < max_thrsold) & (df4.kmpl > min_thrsold)]
df5.shape
```

Out[30]:
```
(6028, 12)
```

In [31]:
```
min_thsold , max_thsold = df4.CC.quantile([0.05,0.95])
min_thsold , max_thsold
```

Out[31]:
```
(998.0, 2494.0)
```

In [32]:
```
df6 = df5[(df5.CC < max_thsold) & (df5.CC > min_thsold)]
df6.shape
```

Out[32]:
```
(5081, 12)
```

In [33]:
```
X = df6.drop(['name','selling_price'],axis='columns')
```

In [34]:
```
Y = df6.selling_price
```

In [35]:
```
from sklearn.model_selection import train_test_split
X_train,X_test,Y_train,Y_test = train_test_split(X,Y,test_size=0.2,random_state=10)
```

In [36]:
```
from sklearn import linear_model
reg = linear_model.LinearRegression()
reg.fit(X_train,Y_train)
reg.score(X_test,Y_test)
```

Out[36]:
```
0.6942284978347655
```

In [37]:
```
reg.predict([[2014,145500,0,0,0,0,5.0,1248,23,74]])
```

Out[37]:
```
C:\Users\Atif\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\base.py:
450: UserWarning: X does not have valid feature names, but LinearRegression was fitted w
ith feature names
  warnings.warn(
array([420455.8832303])
```

In [39]:
```
reg.predict([[2007,120000,1,0,0,0,5.0,1298,16,88]])
```

Out[39]:
```
C:\Users\Atif\AppData\Local\Programs\Python\Python310\lib\site-packages\sklearn\base.py:
450: UserWarning: X does not have valid feature names, but LinearRegression was fitted w
ith feature names
  warnings.warn(
array([104698.96716024])
```

In [ ]:

Loading [MathJax]/extensions/Safe.js