import Libraries

In [1]:

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

import Linear Regression

In [2]:

```
from sklearn.linear_model import LogisticRegression
```

In [3]:

```
lgr=LogisticRegression()
```

Select Required data from certain columns

In [4]:

```
a=pd.read_csv("cars.csv")
a
```

Out[4]:

	Unnamed: 0	model	year	price	transmission	mileage	fuelType	tax	mpg	engineSiz
0	0	T-Roc	2019	25000	Automatic	13904	Diesel	145	49.6	2
1	1	T-Roc	2019	26883	Automatic	4562	Diesel	145	49.6	2
2	2	T-Roc	2019	20000	Manual	7414	Diesel	145	50.4	2
3	3	T-Roc	2019	33492	Automatic	4825	Petrol	145	32.5	2
4	4	T-Roc	2019	22900	Semi-Auto	6500	Petrol	150	39.8	1
...	...	...	...	...	...	...	...	...	...	
99182	10663	A3	2020	16999	Manual	4018	Petrol	145	49.6	1
99183	10664	A3	2020	16999	Manual	1978	Petrol	150	49.6	1
99184	10665	A3	2020	17199	Manual	609	Petrol	150	49.6	1
99185	10666	Q3	2017	19499	Automatic	8646	Petrol	150	47.9	1
99186	10667	Q3	2016	15999	Manual	11855	Petrol	150	47.9	1

99187 rows × 11 columns

In [5]:

```
c=a.dropna()
c
```

Out[5]:

	Unnamed: 0	model	year	price	transmission	mileage	fuelType	tax	mpg	engineSiz
0	0	T-Roc	2019	25000	Automatic	13904	Diesel	145	49.6	2
1	1	T-Roc	2019	26883	Automatic	4562	Diesel	145	49.6	2
2	2	T-Roc	2019	20000	Manual	7414	Diesel	145	50.4	2
3	3	T-Roc	2019	33492	Automatic	4825	Petrol	145	32.5	2
4	4	T-Roc	2019	22900	Semi-Auto	6500	Petrol	150	39.8	1
...	...	...	...	...	...	...	...	...	...	
99182	10663	A3	2020	16999	Manual	4018	Petrol	145	49.6	1
99183	10664	A3	2020	16999	Manual	1978	Petrol	150	49.6	1
99184	10665	A3	2020	17199	Manual	609	Petrol	150	49.6	1
99185	10666	Q3	2017	19499	Automatic	8646	Petrol	150	47.9	1
99186	10667	Q3	2016	15999	Manual	11855	Petrol	150	47.9	1

99187 rows × 11 columns

In [6]:

```
c.columns
```

Out[6]:

```
Index(['Unnamed: 0', 'model', 'year', 'price', 'transmission', 'mileage',
       'fuelType', 'tax', 'mpg', 'engineSize', 'Make'],
      dtype='object')
```

In [88]:

```
fm=c[['Unnamed: 0','year', 'price','mileage','tax', 'mpg', 'engineSize']]
tv=c['Make']
```

Shape

In [89]:

```
fm.shape
```

Out[89]:

```
(99187, 7)
```

In [90]:

```
tv.shape
```

Out[90]:

```
(99187,)
```

To make the data in order (feature matrix)

In [91]:

```
from sklearn.preprocessing import StandardScaler
```

In [92]:

```
fs=StandardScaler().fit_transform(fm)
```

Imply Logistic Regression

In [93]:

```
lgr.fit(fm,tv)
```

```
C:\ProgramData\Anaconda3\lib\site-packages\sklearn\linear_model\_logistic.
py:763: ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown i
n:
    https://scikit-learn.org/stable/modules/preprocessing.html (https://sc
ikit-learn.org/stable/modules/preprocessing.html)
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-reg
ression (https://scikit-learn.org/stable/modules/linear_model.html#logisti
c-regression)
  n_iter_i = _check_optimize_result(
```

Out[93]:

```
LogisticRegression()
```

Prediction

In [96]:

```
ab=[[3,90,45,34,12,43,56]]
```

In [97]:

```
pre=lgr.predict(ab)
```

In [98]:

```
print(pre)
```

```
['vauxhall']
```

To check the output var we have got

In [99]:

```
lgr.classes_
```

Out[99]:

```
array(['Audi', 'BMW', 'VW', 'ford', 'hyundi', 'merc', 'skoda', 'toyota',
       'vauxhall'], dtype=object)
```

Prediction in Probablity value

In [100]:

```
lgr.predict_proba(ab)[0][1]
```

Out[100]:

```
0.09312443674147687
```

In [ ]:

```

```