In [1]:
```python
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
```

In [2]:
```python
from sklearn.linear_model import LogisticRegression
```

In [3]:
```python
df=pd.read_csv("c7_used_cars.csv")
df
```

Out[3]:

	Unnamed: 0	model	year	price	transmission	mileage	fuelType	tax	mpg	engineSize	Make
0	0	T-Roc	2019	25000	Automatic	13904	Diesel	145	49.6	2.0	VW
1	1	T-Roc	2019	26883	Automatic	4562	Diesel	145	49.6	2.0	VW
2	2	T-Roc	2019	20000	Manual	7414	Diesel	145	50.4	2.0	VW
3	3	T-Roc	2019	33492	Automatic	4825	Petrol	145	32.5	2.0	VW
4	4	T-Roc	2019	22900	Semi-Auto	6500	Petrol	150	39.8	1.5	VW
...	...	...	...	...	...	...	...	...	...	...	...
99182	10663	A3	2020	16999	Manual	4018	Petrol	145	49.6	1.0	Audi
99183	10664	A3	2020	16999	Manual	1978	Petrol	150	49.6	1.0	Audi
99184	10665	A3	2020	17199	Manual	609	Petrol	150	49.6	1.0	Audi
99185	10666	Q3	2017	19499	Automatic	8646	Petrol	150	47.9	1.4	Audi
99186	10667	Q3	2016	15999	Manual	11855	Petrol	150	47.9	1.4	Audi

99187 rows × 11 columns

In [4]:
```python
df=df.dropna()
df
```

Out[4]:

	Unnamed: 0	model	year	price	transmission	mileage	fuelType	tax	mpg	engineSize	Make
0	0	T-Roc	2019	25000	Automatic	13904	Diesel	145	49.6	2.0	VW
1	1	T-Roc	2019	26883	Automatic	4562	Diesel	145	49.6	2.0	VW
2	2	T-Roc	2019	20000	Manual	7414	Diesel	145	50.4	2.0	VW
3	3	T-Roc	2019	33492	Automatic	4825	Petrol	145	32.5	2.0	VW
4	4	T-Roc	2019	22900	Semi-Auto	6500	Petrol	150	39.8	1.5	VW
...	...	...	...	...	...	...	...	...	...	...	...
99182	10663	A3	2020	16999	Manual	4018	Petrol	145	49.6	1.0	Audi

	Unnamed: 0	model	year	price	transmission	mileage	fuelType	tax	mpg	engineSize	Make
99183	10664	A3	2020	16999	Manual	1978	Petrol	150	49.6	1.0	Audi
99184	10665	A3	2020	17199	Manual	609	Petrol	150	49.6	1.0	Audi
99185	10666	Q3	2017	19499	Automatic	8646	Petrol	150	47.9	1.4	Audi
99186	10667	Q3	2016	15999	Manual	11855	Petrol	150	47.9	1.4	Audi

99187 rows × 11 columns

In [5]:
```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 99187 entries, 0 to 99186
Data columns (total 11 columns):
 #   Column       Non-Null Count  Dtype
---  ------       --------------  -----
 0   Unnamed: 0   99187 non-null  int64
 1   model        99187 non-null  object
 2   year         99187 non-null  int64
 3   price        99187 non-null  int64
 4   transmission 99187 non-null  object
 5   mileage      99187 non-null  int64
 6   fuelType     99187 non-null  object
 7   tax          99187 non-null  int64
 8   mpg          99187 non-null  float64
 9   engineSize   99187 non-null  float64
 10  Make         99187 non-null  object
dtypes: float64(2), int64(5), object(4)
memory usage: 9.1+ MB
```

In [6]:
```python
df.columns
```

Out[6]:
```
Index(['Unnamed: 0', 'model', 'year', 'price', 'transmission', 'mileage',
       'fuelType', 'tax', 'mpg', 'engineSize', 'Make'],
      dtype='object')
```

In [7]:
```python
feature_matrix=df[['Unnamed: 0','year', 'price',  'mileage',
        'tax', 'mpg', 'engineSize']]
target_vector=df['Make']
```

In [8]:
```python
feature_matrix.shape
```

Out[8]: (99187, 7)

In [9]:
```python
target_vector.shape
```

Out[9]: (99187,)

In [10]:
```python
from sklearn.preprocessing import StandardScaler
```

In [11]:
```python
fs=StandardScaler().fit_transform(feature_matrix)
```

In [12]:
```python
logr=LogisticRegression()
logr.fit(fs,target_vector)
```

Out[12]: LogisticRegression()

In [13]:
```python
observation=[[1,2,3,4,5,6,7]]
```

In [14]:
```python
prediction=logr.predict(observation)
print(prediction)
```

['BMW']

In [15]:
```python
logr.classes_
```

Out[15]: array(['Audi', 'BMW', 'VW', 'ford', 'hyundi', 'merc', 'skoda', 'toyota',
       'vauxhall'], dtype=object)

In [16]:
```python
logr.predict_proba(observation)[0][0]
```

Out[16]: 2.7412293064875042e-05

In [17]:
```python
logr.predict_proba(observation)
```

Out[17]: array([[2.74122931e-05, 9.36836737e-01, 2.51395992e-08, 5.85008303e-09,
        3.09237182e-12, 6.31357545e-02, 6.44018883e-09, 5.85474765e-08,
        7.49581427e-16]])