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

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

In [3]:
```
df = pd.read_csv(r"C:\Users\user\Downloads\c7_used_cars.csv")
df
```

Out[3]:

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

99187 rows × 11 columns

In [4]:
```
df.columns
```

Out[4]:
```
Index(['Unnamed: 0', 'model', 'year', 'price', 'transmission', 'mileage',
       'fuelType', 'tax', 'mpg', 'engineSize', 'Make'],
      dtype='object')
```

In [5]:
```
feature_matrix = df[['Unnamed: 0', 'year', 'price', 'mileage', 'tax', 'mpg', 'e
target_vector = df['fuelType']
```

In [6]:
```
feature_matrix.shape
```

Out[6]:
```
(99187, 7)
```

In [7]:
```
from sklearn.preprocessing import StandardScaler
```

In [8]:
```
fs = StandardScaler().fit_transform(feature_matrix)
```

In [9]:
```
logs = LogisticRegression()
logs.fit(fs,target_vector)
```

```
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 in:
    https://scikit-learn.org/stable/modules/preprocessing.html (https://scikit-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-regression (https://scikit-learn.org/stable/modules/linear_model.html#logistic-regression)
  n_iter_i = _check_optimize_result(
```

Out[9]:
```
LogisticRegression()
```

In [10]:
```
observation = [[1.4,1.5,1.6,1.2,3,4,3]]
prediction = logs.predict(observation)
```

In [11]:
```
print(prediction)
```

```
['Hybrid']
```

In [12]:
```
logs.classes_
```

Out[12]:
```
array(['Diesel', 'Electric', 'Hybrid', 'Other', 'Petrol'], dtype=object)
```

In [13]:
```
logs.predict_proba(observation)[0][0]
```

Out[13]:
```
0.2332564271655534
```

In [ ]:
```

```