

lqj1s29rd

August 1, 2023

```
[6]: import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.linear_model import LogisticRegression
from sklearn.preprocessing import StandardScaler
```

```
[7]: from google.colab import drive
drive.mount('/content/drive')
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.mount("/content/drive", force\_remount=True).

```
[8]: df=pd.read_csv("/content/drive/MyDrive/mydatasets/c7_used_cars.csv")
df
```

```
[8]:
```

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

```

99184  49.6          1.0  Audi
99185  47.9          1.4  Audi
99186  47.9          1.4  Audi

```

[99187 rows x 11 columns]

```
[9]: df.head()
```

```

[9]:   Unnamed: 0  model  year  price  transmission  mileage  fuelType  tax  mpg  \
0           0   T-Roc  2019  25000    Automatic    13904    Diesel   145  49.6
1           1   T-Roc  2019  26883    Automatic     4562    Diesel   145  49.6
2           2   T-Roc  2019  20000     Manual     7414    Diesel   145  50.4
3           3   T-Roc  2019  33492    Automatic     4825    Petrol   145  32.5
4           4   T-Roc  2019  22900   Semi-Auto     6500    Petrol   150  39.8

```

```

      engineSize Make
0           2.0   VW
1           2.0   VW
2           2.0   VW
3           2.0   VW
4           1.5   VW

```

## 1 Data Cleaning and Data Preprocessing

```
[10]: df.info()
```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 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: 8.3+ MB

```

```
[11]: df.describe()
```

```
[11]:
```

	Unnamed: 0	year	price	mileage	tax \
count	99187.000000	99187.000000	99187.000000	99187.000000	99187.000000
mean	6294.413532	2017.087723	16805.347656	23058.914213	120.299838
std	4265.588536	2.123934	9866.773417	21148.523721	63.150926
min	0.000000	1970.000000	450.000000	1.000000	0.000000
25%	2755.000000	2016.000000	9999.000000	7425.000000	125.000000
50%	5591.000000	2017.000000	14495.000000	17460.000000	145.000000
75%	9420.000000	2019.000000	20870.000000	32339.000000	145.000000
max	17964.000000	2060.000000	159999.000000	323000.000000	580.000000

  

	mpg	engineSize
count	99187.000000	99187.000000
mean	55.166825	1.663280
std	16.138522	0.557646
min	0.300000	0.000000
25%	47.100000	1.200000
50%	54.300000	1.600000
75%	62.800000	2.000000
max	470.800000	6.600000

```
[12]: df.columns
```

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

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

```
[14]: fs = StandardScaler().fit_transform(feature_matrix)
logr = LogisticRegression()
logr.fit(fs,target_vector)
```

```
/usr/local/lib/python3.10/dist-packages/sklearn/utils/validation.py:1143:
DataConversionWarning: A column-vector y was passed when a 1d array was
expected. Please change the shape of y to (n_samples, ), for example using
ravel().
y = column_or_1d(y, warn=True)
```

```
[14]: LogisticRegression()
```

```
[15]: observation=[[1,2,3,4,5,6,7]]
prediction = logr.predict(observation)
print(prediction)
```

```
['BMW']
```

```
[16]: logr.classes_
```

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

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

```
[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]])
```