

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

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

```
In [3]: df=pd.read_csv("c7 used csv").dropna()

df
```

Out[3]:

	Unnamed: 0	model	year	price	transmission	mileage	fuelType	tax	mpg	engineSize
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

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

In [5]: 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]: feature\_matrix = df[['Unnamed: 0', 'year', 'price', 'mileage', 'tax', 'mpg', 'engineSize', 'fuelType', 'transmission']]  
target\_vector = df['transmission']

In [7]: feature\_matrix.shape

Out[7]: (99187, 7)

In [8]: target\_vector.shape

Out[8]: (99187,)

In [9]: from sklearn.preprocessing import StandardScaler

In [10]: fs = StandardScaler().fit\_transform(feature\_matrix)

In [11]: logr = LogisticRegression()  
logr.fit(fs, target\_vector)

Out[11]: LogisticRegression()

In [12]: feature\_matrix.shape

Out[12]: (99187, 7)

In [13]: target\_vector.shape

Out[13]: (99187,)

In [14]: from sklearn.preprocessing import StandardScaler

```
In [15]: fs = StandardScaler().fit_transform(feature_matrix)
```

```
In [16]: logr = LogisticRegression()  
logr.fit(fs,target_vector)
```

```
Out[16]: LogisticRegression()
```

```
In [17]: observation=df[['Unnamed: 0','year','price','mileage','tax','mpg','engineSize'
```

```
In [18]: prediction = logr.predict(observation)  
prediction
```

```
Out[18]: array(['Semi-Auto', 'Semi-Auto', 'Semi-Auto', ..., 'Automatic',  
               'Automatic', 'Automatic'], dtype=object)
```

```
In [19]: logr.classes_
```

```
Out[19]: array(['Automatic', 'Manual', 'Other', 'Semi-Auto'], dtype=object)
```

```
In [20]: logr.predict_proba(observation)[0][1]
```

```
Out[20]: 0.0
```