In [2]:

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

In [92]:

```
a=pd.read_csv(r"C:\Users\user\Downloads\c7_used_cars.csv")
a
```

Out[92]:

| | 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 [93]:

```
a=a.head(90000)
a
```

Out[93]:

| | 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. |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | . |
| **89995** | 1476 | Q3 | 2016 | 16450 | Manual | 31000 | Diesel | 30 | 60.1 | 2. |
| **89996** | 1477 | A4 | 2017 | 15498 | Manual | 62140 | Diesel | 145 | 64.2 | 2. |
| **89997** | 1478 | Q2 | 2019 | 26998 | Semi-Auto | 156 | Diesel | 150 | 46.3 | 2. |
| **89998** | 1479 | A1 | 2019 | 17690 | Manual | 8396 | Petrol | 150 | 50.4 | 1. |
| **89999** | 1480 | Q3 | 2014 | 17998 | Semi-Auto | 23340 | Petrol | 260 | 36.7 | 2. |

90000 rows × 11 columns

In [94]:

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

In [95]:

```
a.columns
```

Out[95]:

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

In [96]:

```
b=a[['Unnamed: 0','year','mileage', 'price', 'tax', 'mpg', 'engineSize','Make']]
b
```

Out[96]:

| | Unnamed: 0 | year | mileage | price | tax | mpg | engineSize | Make |
|---|---|---|---|---|---|---|---|---|
| **0** | 0 | 2019 | 13904 | 25000 | 145 | 49.6 | 2.0 | VW |
| **1** | 1 | 2019 | 4562 | 26883 | 145 | 49.6 | 2.0 | VW |
| **2** | 2 | 2019 | 7414 | 20000 | 145 | 50.4 | 2.0 | VW |
| **3** | 3 | 2019 | 4825 | 33492 | 145 | 32.5 | 2.0 | VW |
| **4** | 4 | 2019 | 6500 | 22900 | 150 | 39.8 | 1.5 | VW |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... |
| **89995** | 1476 | 2016 | 31000 | 16450 | 30 | 60.1 | 2.0 | Audi |
| **89996** | 1477 | 2017 | 62140 | 15498 | 145 | 64.2 | 2.0 | Audi |
| **89997** | 1478 | 2019 | 156 | 26998 | 150 | 46.3 | 2.0 | Audi |
| **89998** | 1479 | 2019 | 8396 | 17690 | 150 | 50.4 | 1.0 | Audi |
| **89999** | 1480 | 2014 | 23340 | 17998 | 260 | 36.7 | 2.0 | Audi |

90000 rows × 8 columns

In [97]:

```
c=b.iloc[:,0:5]
d=b.iloc[:,-1]
```

In [98]:

```
c.shape
```

Out[98]:

```
(90000, 5)
```

In [99]:

```
d.shape
```

Out[99]:

```
(90000,)
```

In [100]:

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

In [101]:

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

In [102]:

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

In [103]:

```
logr=LogisticRegression()
logr.fit(fs,d)
```

Out[103]:

```
LogisticRegression()
```

In [104]:

```
e=[[2,5,77,7,88]]
```

In [105]:

```
prediction=logr.predict(e)
prediction
```

Out[105]:

```
array(['BMW'], dtype=object)
```

In [106]:

```
logr.classes_
```

Out[106]:

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

In [107]:

```
logr.predict_proba(e)[0][0]
```

Out[107]:

```
2.6224968197307055e-21
```

In [108]:

```
logr.predict_proba(e)[0][1]
```

Out[108]:

```
0.9999999999637039
```

In [109]:

```
import re
from sklearn.datasets import load_digits
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import sklearn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
```

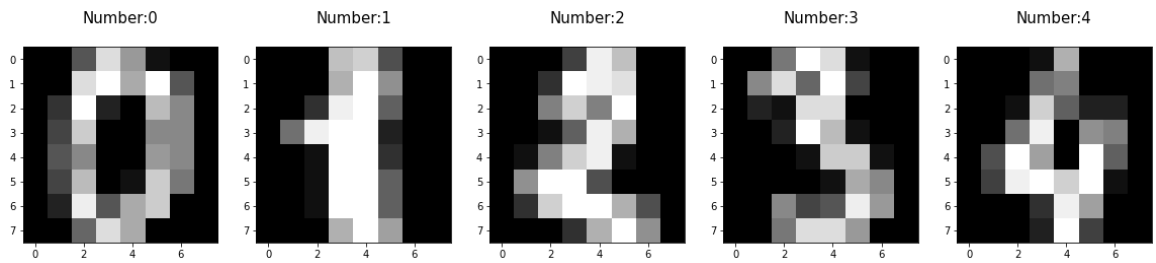
In [110]:

```
digits=load_digits()
digits
```

```
  'pixel_0_4',
  'pixel_0_5',
  'pixel_0_6',
  'pixel_0_7',
  'pixel_1_0',
  'pixel_1_1',
  'pixel_1_2',
  'pixel_1_3',
  'pixel_1_4',
  'pixel_1_5',
  'pixel_1_6',
  'pixel_1_7',
  'pixel_2_0',
  'pixel_2_1',
  'pixel_2_2',
  'pixel_2_3',
  'pixel_2_4',
  'pixel_2_5',
  'pixel_2_6',
```

In [111]:

```
plt.figure(figsize=(20,4))
for index,(image,label)in enumerate(zip(digits.data[0:5],digits.target[0:5])):
    plt.subplot(1,5,index+1)
    plt.imshow(np.reshape(image,(8,8)),cmap=plt.cm.gray)
    plt.title('Number:%i\n'%label,fontsize=15)
```




In [112]:

```
x_train,x_test,y_train,y_test=train_test_split(digits.data,digits.target,test_size=0.30)
```

In [113]:

```
print(x_train.shape)
print(x_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(1257, 64)
(540, 64)
(1257,)
(540,)
```

In [114]:

```
logre=LogisticRegression(max_iter=10000)
logre.fit(x_train,y_train)
```

Out[114]:

```
LogisticRegression(max_iter=10000)
```

In [115]:

```
logre.predict(x_test)
```

Out[115]:

```
array([5, 2, 9, 7, 8, 1, 0, 0, 2, 4, 0, 3, 2, 4, 3, 8, 4, 6, 8, 0, 2, 8,
       7, 8, 6, 0, 9, 5, 2, 4, 6, 9, 2, 9, 8, 3, 1, 1, 2, 2, 4, 7, 9, 2,
       6, 2, 9, 6, 9, 2, 5, 2, 5, 7, 9, 3, 2, 3, 7, 0, 5, 6, 8, 7, 6, 0,
       9, 8, 4, 8, 5, 1, 0, 8, 1, 4, 0, 3, 3, 6, 3, 4, 5, 7, 4, 3, 3, 4,
       9, 9, 6, 3, 1, 0, 1, 8, 1, 9, 8, 5, 8, 4, 6, 4, 8, 8, 1, 6, 4, 2,
       0, 2, 4, 0, 5, 1, 7, 2, 8, 5, 1, 1, 8, 0, 6, 4, 3, 3, 8, 3, 7, 0,
       1, 0, 5, 4, 3, 2, 9, 3, 1, 4, 0, 3, 2, 0, 0, 8, 4, 3, 1, 9, 1, 8,
       6, 5, 1, 2, 9, 5, 8, 7, 1, 7, 4, 9, 4, 9, 1, 4, 9, 5, 3, 2, 2, 7,
       7, 3, 0, 2, 9, 1, 0, 8, 0, 3, 5, 1, 1, 9, 9, 5, 3, 8, 7, 1, 4, 0,
       1, 7, 9, 0, 0, 8, 7, 0, 7, 4, 4, 2, 9, 7, 7, 7, 2, 9, 6, 6, 8, 6,
       2, 8, 9, 7, 4, 4, 6, 4, 8, 1, 6, 2, 2, 7, 6, 1, 2, 8, 8, 6, 5, 3,
       6, 7, 4, 1, 8, 1, 8, 8, 9, 5, 8, 7, 8, 0, 9, 2, 9, 2, 4, 4, 0, 0,
       3, 0, 6, 2, 0, 4, 3, 3, 3, 9, 5, 0, 6, 2, 6, 9, 4, 5, 6, 2, 1, 2,
       3, 7, 5, 9, 6, 9, 7, 4, 7, 4, 9, 9, 5, 6, 6, 6, 5, 8, 3, 4, 9, 3,
       7, 8, 3, 2, 4, 3, 5, 3, 0, 9, 5, 5, 0, 4, 9, 8, 0, 5, 5, 4, 8, 0,
       1, 4, 4, 2, 2, 3, 4, 7, 7, 0, 9, 0, 3, 7, 3, 3, 5, 8, 2, 9, 2, 0,
       2, 5, 1, 2, 7, 1, 7, 6, 6, 5, 8, 3, 6, 3, 3, 9, 3, 9, 5, 8, 7, 6,
       5, 1, 3, 3, 8, 4, 1, 4, 2, 5, 2, 7, 4, 2, 4, 1, 1, 7, 0, 1, 0, 3,
       8, 8, 7, 0, 5, 6, 8, 9, 5, 8, 1, 3, 0, 2, 6, 6, 7, 9, 4, 2, 8, 4,
       0, 9, 1, 7, 2, 6, 4, 2, 1, 9, 7, 4, 2, 2, 5, 7, 1, 7, 3, 8, 3, 8,
       7, 7, 7, 2, 6, 9, 9, 7, 5, 3, 4, 1, 1, 2, 9, 7, 1, 4, 1, 3, 0, 3,
       3, 1, 5, 5, 0, 6, 7, 5, 8, 5, 2, 8, 7, 0, 9, 0, 3, 2, 9, 0, 9, 6,
       4, 0, 8, 4, 4, 2, 2, 1, 2, 7, 2, 6, 6, 8, 2, 5, 8, 0, 4, 6, 2, 5,
       7, 6, 2, 7, 0, 6, 2, 9, 3, 7, 5, 3, 4, 2, 5, 0, 5, 8, 3, 2, 3, 1,
       6, 5, 4, 4, 1, 8, 5, 1, 7, 7, 6, 6])
```

In [116]:

```
logre.score(x_test,y_test)
```

Out[116]:

```
0.9611111111111111
```

In [ ]: