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

In [2]:
```
#loading csv file as a data frame using pandas
df = pd.read_csv('cars.csv')
```

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

In [5]:
```
df.columns
```

Out[5]:
```
Index(['passedemissions', 'mpg', 'displacement', 'horsepower', 'cylinders',
       'weight', 'acceleration', 'modelyear', 'carname'],
      dtype='object')
```

In [6]:
```
passed_emissions=pd.get_dummies(df['passedemissions'],drop_first=True)
```

In [7]:
```
passed_emissions
```

Out[7]:

	True
0	0
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	1
15	1
16	0
17	1
18	1
19	1
20	1
21	1
22	1
23	1
24	1
25	0
26	0
27	0
28	0
29	1
...	...
362	1
363	1
364	1
365	1
366	1
367	1
368	1
369	1
370	1
371	1
372	1
373	1
374	1
375	1
376	1
377	1
378	1
379	1
380	1
381	1
382	1
383	1
384	1
385	1
386	1
387	1
388	1
389	1
390	1
391	1

392 rows × 1 columns

In [8]:
```
passed_emissions.rename(columns={True:'Emission_test'},
                        inplace=True)
```

In [9]:
```
passed_emissions
```

Out[9]:

	Emission_test
0	0
1	0
2	0
3	0
4	0
5	0
6	0
7	0
8	0
9	0
10	0
11	0
12	0
13	0
14	1
15	1
16	0
17	1
18	1
19	1
20	1
21	1
22	1
23	1
24	1
25	0
26	0
27	0
28	0
29	1
...	...
362	1
363	1
364	1
365	1
366	1
367	1
368	1
369	1
370	1
371	1
372	1
373	1
374	1
375	1
376	1
377	1
378	1
379	1
380	1
381	1
382	1
383	1
384	1
385	1
386	1
387	1
388	1
389	1
390	1
391	1

392 rows × 1 columns

In [10]:
```
df = pd.concat([df,passed_emissions],axis=1)
```

In [11]:
```
df.head()
```

Out[11]:

	passedemissions	mpg	displacement	horsepower	cylinders	weight	acceleration	modelyear	carname	Emission_test
0	False	18.0	307.0	130.0	8.0	1.7520	12.0	70.0	chevrolet chevelle malibu	0
1	False	15.0	350.0	165.0	8.0	1.8465	11.5	70.0	buick skylark 320	0
2	False	18.0	318.0	150.0	8.0	1.7180	11.0	70.0	plymouth satellite	0
3	False	16.0	304.0	150.0	8.0	1.7165	12.0	70.0	amc rebel sst	0
4	False	17.0	302.0	140.0	8.0	1.7245	10.5	70.0	ford torino	0

In [13]:
```
df.drop(['passedemissions','carname'],axis=1,inplace=True)
```

In [14]:
```
df.head()
```

Out[14]:

	mpg	displacement	horsepower	cylinders	weight	acceleration	modelyear	Emission_test
0	18.0	307.0	130.0	8.0	1.7520	12.0	70.0	0
1	15.0	350.0	165.0	8.0	1.8465	11.5	70.0	0
2	18.0	318.0	150.0	8.0	1.7180	11.0	70.0	0
3	16.0	304.0	150.0	8.0	1.7165	12.0	70.0	0
4	17.0	302.0	140.0	8.0	1.7245	10.5	70.0	0

In [15]:
```
X=df.drop('Emission_test',axis=1)
y=df['Emission_test']
```

In [16]:
```
from sklearn.model_selection import train_test_split
```

In [17]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=101)
```

In [18]:
```
from sklearn.linear_model import LogisticRegression
```

In [19]:
```
logm = LogisticRegression()
```

In [20]:
```
logm.fit(X_train,y_train)
```

```
C:\Users\Chandan\Anaconda3\lib\site-packages\sklearn\linear_model\logistic.py:433: FutureWarnin
g: Default solver will be changed to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)
```

Out[20]:
```
LogisticRegression(C=1.0, class_weight=None, dual=False, fit_intercept=True,
                   intercept_scaling=1, max_iter=100, multi_class='warn',
                   n_jobs=None, penalty='l2', random_state=None, solver='warn',
                   tol=0.0001, verbose=0, warm_start=False)
```

In [21]:
```
estimations = logm.predict(X_test)
```

In [24]:
```
from sklearn.metrics import classification_report
```

In [26]:
```
from sklearn.metrics import confusion_matrix
```

In [28]:
```
print(confusion_matrix(y_test,estimations))
```

```
[[36  4]
 [ 5 73]]
```

In [29]:
```
# 36 = True negative Predicted No and in actual also No
# 4 = false positive predicted yes but in actual was no
```