

In [44]:

```
import os
```

In [45]:

```
os.getcwd()
```

Out[45]:

```
'C:\\Users\\saima\\Desktop\\datascience'
```

In [46]:

```
os.chdir("C:/Users/saima/Desktop/datascience")
```

In [47]:

```
os.getcwd()
```

Out[47]:

```
'C:\\Users\\saima\\Desktop\\datascience'
```

In [48]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sb
import scipy
from scipy.stats.stats import pearsonr
```

In [49]:

```
dia=pd.read_excel("diabetes2.xlsx")
```

In [50]:

```
dia
```

Out[50]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeI
0	6	148	72	35	0	33.600	
1	1	85	66	29	0	26.600	
2	8	183	64	0	0	23.300	
3	1	89	66	23	94	28.100	
4	0	137	40	35	168	43.100	
5	5	116	74	0	0	25.600	
6	3	78	50	32	88	31.000	
7	10	115	0	0	0	35.300	
8	2	197	70	45	543	30.500	
9	8	125	96	0	0	0.000	
10	4	110	92	0	0	37.600	
11	10	168	74	0	0	38.000	
12	10	139	80	0	0	27.100	
13	1	189	60	23	846	30.100	
14	5	166	72	19	175	25.800	
15	7	100	0	0	0	30.000	
16	0	118	84	47	230	45.800	
17	7	107	74	0	0	29.600	
18	1	103	30	38	83	43.300	
19	1	115	70	30	96	34.600	
20	3	126	88	41	235	39.300	
21	8	99	84	0	0	35.400	
22	7	196	90	0	0	39.800	
23	9	119	80	35	0	29.000	
24	11	143	94	33	146	36.600	
25	10	125	70	26	115	31.100	
26	7	147	76	0	0	39.400	
27	1	97	66	15	140	23.200	
28	13	145	82	19	110	22.200	
29	5	117	92	0	0	34.100	
...	
738	2	99	60	17	160	36.600	
739	1	102	74	0	0	39.500	
740	11	120	80	37	150	42.300	
741	3	102	44	20	94	30.800	
742	1	109	58	18	116	28.500	
743	9	140	94	0	0	32.700	

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeI
744	13	153	88	37	140	40.600	
745	12	100	84	33	105	30.000	
746	1	147	94	41	0	49.300	
747	1	81	74	41	57	46.300	
748	3	187	70	22	200	36.400	
749	6	162	62	0	0	24.300	
750	4	136	70	0	0	31.200	
751	1	121	78	39	74	39.000	
752	3	108	62	24	0	26.000	
753	0	181	88	44	510	43.300	
754	8	154	78	32	0	32.400	
755	1	128	88	39	110	36.500	
756	7	137	90	41	0	32.000	
757	0	123	72	0	0	36.300	
758	1	106	76	0	0	37.500	
759	6	190	92	0	0	35.500	
760	2	88	58	26	16	28.400	
761	9	170	74	31	0	44.000	
762	9	89	62	0	0	22.500	
763	10	101	76	48	180	32.900	
764	2	122	70	27	0	36.800	
765	5	121	72	23	112	26.200	
766	1	126	60	0	0	30.100	
767	1	93	70	31	0	30.400	

768 rows × 9 columns



In [51]:

```
dia.head()
```

Out[51]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFu
0	6	148	72	35	0	33.600	
1	1	85	66	29	0	26.600	
2	8	183	64	0	0	23.300	
3	1	89	66	23	94	28.100	
4	0	137	40	35	168	43.100	

In [52]:

```
pd.set_option('display.float_format', lambda x: '%.3f' % x)
```

In [53]:

```
dia.describe()
```

Out[53]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedig
count	768.000	768.000	768.000	768.000	768.000	768.000	
mean	3.845	120.895	69.105	20.536	79.799	31.993	
std	3.370	31.973	19.356	15.952	115.244	7.884	
min	0.000	0.000	0.000	0.000	0.000	0.000	
25%	1.000	99.000	62.000	0.000	0.000	27.300	
50%	3.000	117.000	72.000	23.000	30.500	32.000	
75%	6.000	140.250	80.000	32.000	127.250	36.600	
max	17.000	199.000	122.000	99.000	846.000	67.100	

In [54]:

```
dia.dtypes
```

Out[54]:

```
Pregnancies      int64
Glucose           int64
BloodPressure     int64
SkinThickness     int64
Insulin           int64
BMI               float64
DiabetesPedigreeFunction float64
Age               int64
Diabetes          object
dtype: object
```

In [55]:

```
dia.apply(lambda x:sum(x.isnull()),axis=0)
```

Out[55]:

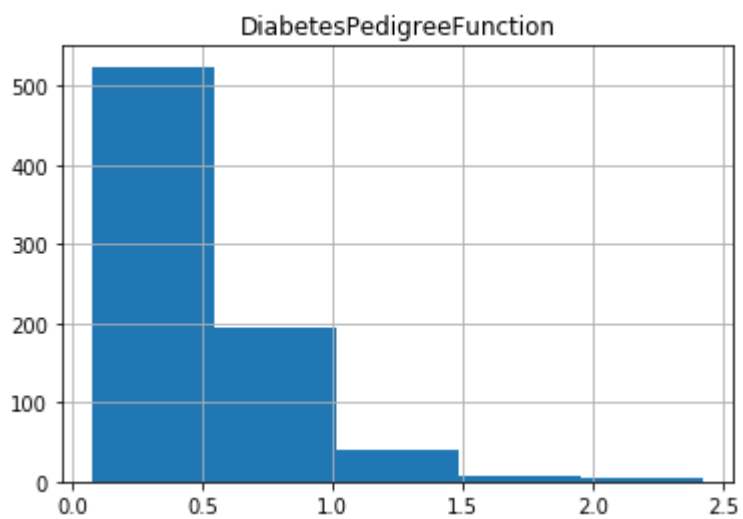
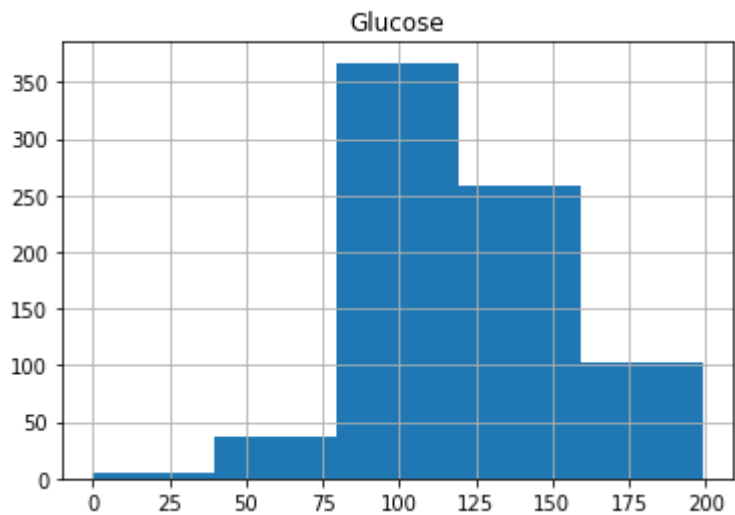
Pregnancies	0
Glucose	0
BloodPressure	0
SkinThickness	0
Insulin	0
BMI	0
DiabetesPedigreeFunction	0
Age	0
Diabetes	0
dtype:	int64

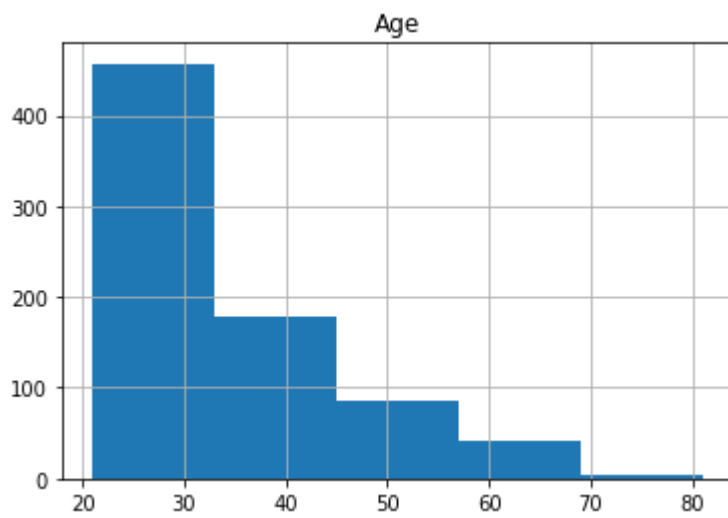
In [56]:

```
dia.hist('Glucose',bins=5),dia.hist('DiabetesPedigreeFunction',bins=5),dia.hist('Age',bins=5)
```

Out[56]:

```
(array([[<matplotlib.axes._subplots.AxesSubplot object at 0x0000013919DDFB00>]],  
      dtype=object),  
 array([[<matplotlib.axes._subplots.AxesSubplot object at 0x0000013919DE52B0>]],  
      dtype=object),  
 array([[<matplotlib.axes._subplots.AxesSubplot object at 0x0000013919DFD898>]],  
      dtype=object))
```



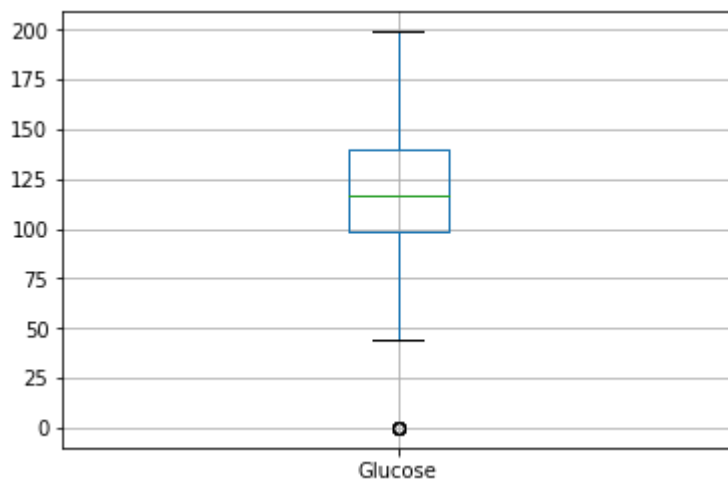


In [57]:

```
dia.boxplot(column='Glucose')
```

Out[57]:

<matplotlib.axes._subplots.AxesSubplot at 0x13919f5f4a8>

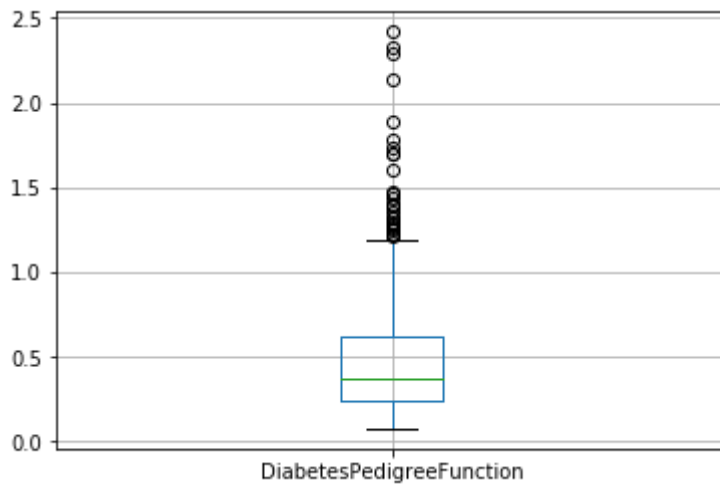


In [58]:

```
dia.boxplot(column='DiabetesPedigreeFunction')
```

Out[58]:

<matplotlib.axes._subplots.AxesSubplot at 0x13919f827f0>



In [59]:

```
IQR_SP=1779000-429000  
IQR_SP
```

Out[59]:

1350000

In [60]:

```
Lower_bound=429000-(1.5*IQR_SP)  
Upper_bound=1779000+(1.5*IQR_SP)  
print(Lower_bound,Upper_bound)
```

-1596000.0 3804000.0

In [61]:

```
dia.dtypes
```

Out[61]:

```
Pregnancies      int64
Glucose           int64
BloodPressure     int64
SkinThickness     int64
Insulin           int64
BMI               float64
DiabetesPedigreeFunction float64
Age               int64
Diabetes          object
dtype: object
```

In [97]:

```
dia.corr(method='spearman')
```

Out[97]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI
Pregnancies	1.000	0.131	0.185	-0.085	-0.127	0.000
Glucose	0.131	1.000	0.235	0.060	0.213	0.231
BloodPressure	0.185	0.235	1.000	0.126	-0.007	0.293
SkinThickness	-0.085	0.060	0.126	1.000	0.541	0.444
Insulin	-0.127	0.213	-0.007	0.541	1.000	0.193
BMI	0.000	0.231	0.293	0.444	0.193	1.000
DiabetesPedigreeFunction	-0.043	0.091	0.030	0.180	0.221	0.143
Age	0.607	0.285	0.351	-0.067	-0.114	0.133

In [98]:

```
dia.describe()
```

Out[98]:

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction
count	768.000	768.000	768.000	768.000	768.000	768.000	768.000
mean	3.845	120.895	69.105	20.536	79.799	31.993	0.332
std	3.370	31.973	19.356	15.952	115.244	7.884	0.332
min	0.000	0.000	0.000	0.000	0.000	0.000	0.000
25%	1.000	99.000	62.000	0.000	0.000	27.300	0.167
50%	3.000	117.000	72.000	23.000	30.500	32.000	0.332
75%	6.000	140.250	80.000	32.000	127.250	36.600	0.332
max	17.000	199.000	122.000	99.000	846.000	67.100	0.674

In [70]:

```
x=dia.iloc[:,[1,4,6]].values  
y=dia.iloc[:,[8]].values
```

In [71]:

```
x
```

Out[71]:

```
array([[148.    ,  0.    ,  0.627],  
       [ 85.    ,  0.    ,  0.351],  
       [183.    ,  0.    ,  0.672],  
       ...,  
       [121.    , 112.    ,  0.245],  
       [126.    ,  0.    ,  0.349],  
       [ 93.    ,  0.    ,  0.315]])
```

In [72]:

```
y
```

Out[72]:

```
array(['pos'],
      ['neg'],
      ['pos'],
      ['neg'],
      ['pos'],
      ['neg'],
      ['pos'],
      ['neg'],
      ['pos'],
      ['pos'],
      ['neg'],
      ['pos'],
      ['neg'],
      ['pos'],
      ['pos'],
      ['pos'],
      ['pos'],
      ['pos'],
      ['neg'],
      ['pos'],
      ['neg'],
      ['neg'],
      ['neg'],
      ['pos'],
      ['neg'],
      ['neg'],
      ['neg'],
      ['neg'],
      ['pos'],
      ['neg'],
      ['neg'],
      ['neg'],
      ['neg'],
      ['pos'],
      ['pos'],
      ['pos'],
      ['neg'],
      ['neg'],
      ['neg'],
      ['pos'],
      ['neg'],
      ['neg'],
      ['neg'],
      ['pos'],
      ['neg'],
      ['neg'],
      ['neg'],
      ['pos'],
      ['neg'],
      ['pos'],
      ['neg'],
      ['neg'])
```

```
['neg'],
['neg'],
['pos'],
['neg'],
['neg'],
['pos'],
['neg'],
['pos'],
['neg'],
['neg'],
['neg'],
['pos'],
['neg'],
['pos'],
['neg'],
['neg'],
['neg'],
['neg'],
['neg'],
['pos'],
['neg'],
['neg'],
['neg'],
['neg'],
['neg'],
['pos'],
['neg'],
['neg'],
['neg'],
['pos'],
['neg'],
['neg'],
['neg'],
['neg'],
['pos'],
['pos'],
['neg'],
['neg'],
['neg'],
['neg'],
['neg'],
['neg'],
['neg'],
['pos'],
['pos'],
['pos'],
['neg'],
['neg'],
['pos'],
['pos'],
['pos'],
['neg'],
['neg'],
['pos'],
['pos'],
['pos'],
['neg'],
['neg'],
['neg'],
```

16/42

localhost:8888/nbconvert/html/diabetes.ipynb?download=false

localhost:8888/nbconvert/html/diabetes.ipynb?download=false

19/42

```
['neg'],
['neg'],
['pos'],
['neg'],
['neg'],
['pos'],
['pos'],
['neg'],
['neg'],
['neg'],
['neg'],
['pos'],
['neg'],
['neg'],
['pos'],
['neg'],
['neg'],
['neg'],
['neg'],
['neg'],
['pos'],
['pos'],
['pos'],
['neg'],
['neg'],
['pos'],
['neg'],
['neg'],
['pos'],
['neg'],
['pos'],
['pos'],
['pos'],
['neg'],
['pos'],
['neg'],
['pos'],
['neg'],
['pos'],
['neg'],
['pos'],
['pos'],
['pos'],
['neg'],
['neg'],
['neg'],
['neg'],
['pos'],
['pos'],
['neg'],
['pos'],
['neg'],
['neg'],
['neg'],
['neg'],
['pos'],
['pos'],
['neg'],
['pos'],
['neg'],
['neg'],
['neg'],
['neg'],
['pos'],
```

21/42

```
['neg'],
['neg'],
['neg'],
['neg'],
['neg'],
['neg'],
['neg'],
['pos'],
['neg'],
['neg'],
['neg'],
['neg'],
['pos'],
['neg'],
['neg'],
['neg'],
['pos'],
['neg'],
['neg'],
['neg'],
['pos'],
['neg'],
['neg'],
['neg'],
['pos'],
['neg'],
['neg'],
['neg'],
['pos'],
['neg'],
['neg'],
['neg'],
['pos'],
['neg'],
['neg'],
['neg'],
['neg'],
['pos'],
['neg'],
['neg'],
['neg'],
['neg'],
['neg'],
['pos'],
['neg'],
['neg'],
['neg'],
['neg'],
['neg'],
['pos'],
['pos'],
['pos'],
['pos'],
['neg'],
['neg'],
['pos'],
['pos']
```

23/42

24/42

localhost:8888/nbconvert/html/diabetes.ipynb?download=false

```
['pos'],
['pos'],
['pos'],
['neg'],
['neg'],
['neg'],
['neg'],
['neg'],
['neg'],
['pos'],
['pos'],
['neg'],
['neg'],
['pos'],
['neg'],
['neg'],
['pos'],
['neg'],
['pos'],
['neg'],
['pos'],
['pos'],
['pos'],
['neg'],
['pos'],
['neg'],
['pos'],
['pos'],
['pos'],
['neg'],
['pos'],
['neg'],
['pos'],
['neg'],
['neg'],
['neg'],
['pos'],
['neg']] , dtype=object)
```

In [66]:

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.4,random_state=0)
```

In [67]:

```
x_test
```

Out[67]:

```
array([[199],
       [107],
       [ 76],
       [166],
       [111],
       [ 81],
       [152],
       [176],
       [127],
       [103],
       [158],
       [187],
       [114],
       [ 82],
       [ 65],
       [108],
       [179],
       [ 73],
       [132],
       [ 99],
       [144],
       [118],
       [ 87],
       [ 61],
       [ 97],
       [124],
       [104],
       [153],
       [112],
       [110],
       [136],
       [102],
       [102],
       [122],
       [119],
       [168],
       [136],
       [117],
       [119],
       [180],
       [119],
       [ 90],
       [ 99],
       [193],
       [197],
       [102],
       [108],
       [129],
       [109],
       [125],
       [133],
       [108],
       [151],
       [137],
       [ 91],
       [  0],
       [ 85],
       [115],
       [134],
```

```
[115],  
[167],  
[147],  
[125],  
[138],  
[182],  
[197],  
[165],  
[122],  
[106],  
[ 72],  
[108],  
[119],  
[ 86],  
[183],  
[171],  
[137],  
[ 68],  
[154],  
[ 83],  
[ 97],  
[100],  
[136],  
[110],  
[ 97],  
[ 95],  
[ 92],  
[139],  
[103],  
[181],  
[ 94],  
[124],  
[129],  
[113],  
[103],  
[164],  
[111],  
[ 97],  
[152],  
[111],  
[115],  
[114],  
[ 99],  
[110],  
[ 95],  
[128],  
[108],  
[126],  
[137],  
[ 99],  
[183],  
[ 88],  
[112],  
[143],  
[128],  
[173],  
[161],  
[108],  
[193],  
[111],  
[120],
```

```
[112],  
[103],  
[115],  
[121],  
[143],  
[ 90],  
[127],  
[116],  
[135],  
[175],  
[139],  
[105],  
[ 81],  
[172],  
[ 94],  
[102],  
[107],  
[103],  
[141],  
[120],  
[ 87],  
[154],  
[105],  
[ 62],  
[114],  
[ 99],  
[ 88],  
[ 80],  
[ 97],  
[105],  
[ 85],  
[142],  
[116],  
[107],  
[150],  
[ 97],  
[165],  
[109],  
[120],  
[135],  
[161],  
[194],  
[ 71],  
[122],  
[179],  
[119],  
[ 76],  
[111],  
[125],  
[105],  
[ 89],  
[ 74],  
[107],  
[102],  
[142],  
[ 91],  
[ 95],  
[ 77],  
[110],  
[ 91],  
[102],
```

```
[158],  
[ 83],  
[180],  
[136],  
[ 44],  
[131],  
[124],  
[144],  
[109],  
[128],  
[ 88],  
[ 89],  
[118],  
[ 91],  
[140],  
[145],  
[154],  
[102],  
[ 99],  
[111],  
[139],  
[103],  
[136],  
[113],  
[ 77],  
[ 95],  
[ 99],  
[155],  
[124],  
[ 95],  
[ 88],  
[112],  
[151],  
[112],  
[ 81],  
[119],  
[131],  
[161],  
[ 96],  
[ 95],  
[101],  
[128],  
[134],  
[147],  
[140],  
[104],  
[109],  
[ 95],  
[106],  
[122],  
[151],  
[ 83],  
[122],  
[146],  
[123],  
[104],  
[152],  
[100],  
[170],  
[ 88],  
[132],
```

[113],
[109],
[92],
[100],
[90],
[139],
[102],
[158],
[107],
[145],
[90],
[128],
[105],
[163],
[126],
[100],
[95],
[117],
[91],
[93],
[97],
[122],
[111],
[56],
[100],
[184],
[115],
[78],
[114],
[68],
[85],
[119],
[136],
[125],
[80],
[121],
[137],
[75],
[85],
[169],
[123],
[111],
[100],
[111],
[80],
[84],
[85],
[145],
[170],
[122],
[79],
[143],
[102],
[197],
[196],
[83],
[92],
[120],
[106],
[162],
[107],


```
[101],  
[130],  
[148],  
[146],  
[134]], dtype=int64)
```

In [77]:

```
from sklearn.linear_model.logistic import LogisticRegression  
  
lgshr=LogisticRegression(random_state=0)  
lgshr.fit(x_train,y_train)
```

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

Out[77]:

```
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=0, solver='warn',  
tol=0.0001, verbose=0, warm_start=False)
```

In [78]:

```
y_pred=lgsr.predict(x_test)
y_pred
```

Out[78]:

```
array(['pos', 'neg', 'neg', 'pos', 'neg', 'neg', 'pos', 'pos', 'neg',
       'neg', 'pos', 'pos', 'neg', 'neg', 'neg', 'neg', 'pos', 'neg',
       'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg',
       'pos', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'pos',
       'neg', 'neg', 'neg', 'pos', 'neg', 'neg', 'neg', 'pos', 'pos',
       'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'pos', 'neg',
       'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'pos', 'pos', 'neg',
       'neg', 'pos', 'pos', 'pos', 'neg', 'neg', 'neg', 'neg', 'neg',
       'neg', 'pos', 'pos', 'neg', 'neg', 'pos', 'neg', 'neg', 'neg',
       'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'pos', 'neg',
       'neg', 'neg', 'neg', 'neg', 'pos', 'neg', 'neg', 'pos', 'neg',
       'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg',
       'neg', 'pos', 'neg', 'neg', 'neg', 'neg', 'neg', 'pos', 'pos',
       'pos', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg',
       'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg',
       'neg', 'pos', 'neg', 'pos', 'neg', 'neg', 'neg', 'neg', 'neg',
       'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg',
       'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'pos', 'pos',
       'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg',
       'neg', 'pos', 'neg', 'neg', 'neg', 'neg', 'neg', 'pos', 'neg',
       'neg', 'neg', 'pos', 'neg', 'neg', 'neg', 'neg', 'neg', 'pos',
       'pos', 'neg', 'neg', 'neg', 'neg', 'neg', 'pos', 'neg', 'neg',
       'pos', 'neg', 'neg', 'pos', 'neg', 'pos', 'neg', 'neg', 'neg',
       'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'pos', 'neg', 'pos',
       'neg', 'neg', 'neg', 'pos', 'neg', 'neg', 'neg', 'neg', 'neg',
       'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg',
       'neg', 'neg', 'pos', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg',
       'neg', 'pos', 'pos', 'neg', 'neg', 'neg', 'neg', 'neg', 'pos',
       'pos', 'neg', 'neg', 'neg', 'neg', 'pos', 'neg', 'neg', 'pos',
       'pos', 'neg'], dtype=object)
```

In [79]:

```
from sklearn.metrics import confusion_matrix
conf_matrix=confusion_matrix(y_pred,y_test)
conf_matrix
```

Out[79]:

```
array([[187,  61],
       [ 18,  42]], dtype=int64)
```

In [80]:

```
from sklearn.metrics import accuracy_score
acc_sco=accuracy_score(y_test,y_pred)
acc_sco
```

Out[80]:

0.7435064935064936

In [81]:

```
#Support vector classification
from sklearn.svm import SVC
sc =SVC(kernel='rbf')
sc_classifier=sc.fit(x_train,y_train)
```

C:\Users\saima\Anaconda3\lib\site-packages\sklearn\svm\base.py:196: FutureWarning: The default value of gamma will change from 'auto' to 'scale' in version 0.22 to account better for unscaled features. Set gamma explicitly to 'auto' or 'scale' to avoid this warning.
"avoid this warning.", FutureWarning)

In [82]:

```
#Predicting Test and Train data

svc_y_test=sc_classifier.predict(x_test)
print(svc_y_test)
```

```
[ 'pos' 'neg' 'neg' 'neg' 'neg' 'neg' 'pos' 'pos' 'neg' 'neg' 'pos' 'pos'
'neg' 'neg' 'neg' 'neg' 'pos' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg'
'neg' 'pos' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'pos'
'neg' 'neg' 'neg' 'pos' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg'
'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg'
'pos' 'pos' 'neg' 'neg' 'pos' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg'
'neg' 'pos' 'pos' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg'
'neg' 'neg' 'neg' 'neg' 'pos' 'neg' 'pos' 'neg' 'neg' 'neg' 'pos' 'neg'
'neg' 'pos' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg'
'neg' 'pos' 'neg' 'neg' 'neg' 'neg' 'pos' 'pos' 'neg' 'neg' 'neg' 'neg'
'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'pos' 'neg' 'neg'
'neg' 'pos' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg'
'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg'
'neg' 'neg' 'neg' 'neg' 'pos' 'pos' 'neg' 'neg' 'pos' 'neg' 'neg' 'neg'
'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg'
'neg' 'pos' 'neg' 'pos' 'neg' 'neg' 'neg' 'pos' 'neg' 'neg' 'neg' 'neg'
'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg'
'neg' 'neg' 'neg' 'neg' 'pos' 'pos' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg'
'neg' 'neg' 'pos' 'neg' 'neg' 'neg' 'neg' 'neg' 'pos' 'neg' 'neg' 'neg'
'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'pos' 'neg' 'neg' 'pos' 'neg' 'pos'
'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'pos' 'neg' 'neg'
'neg' 'neg' 'neg' 'pos' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg'
'neg' 'neg' 'neg' 'pos' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg'
'neg' 'neg' 'neg' 'neg' 'neg' 'pos' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg'
'neg' 'neg' 'pos' 'neg' 'neg' 'neg' 'neg' 'neg' 'pos' 'neg' 'neg' 'neg'
'neg' 'pos' 'neg' 'neg' 'neg' 'pos' 'pos' 'neg' ]
```

In [83]:

```
#Build a confusion matrix on test and train  
from sklearn.metrics import confusion_matrix  
svc_cm_test=confusion_matrix(y_test,svc_y_test)  
svc_cm_test
```

Out[83]:

```
array([[191, 14],  
       [ 72, 31]], dtype=int64)
```

In [84]:

```
from sklearn.metrics import accuracy_score  
acc_sco=accuracy_score(y_test,y_pred)  
acc_sco
```

Out[84]:

```
0.7435064935064936
```

In [85]:

```
#Support vector classification  
from sklearn.svm import SVC  
sc =SVC(kernel='rbf')  
sc_classifier=sc.fit(x_train,y_train)
```

C:\Users\saima\Anaconda3\lib\site-packages\sklearn\svm\base.py:196: FutureWarning: The default value of gamma will change from 'auto' to 'scale' in version 0.22 to account better for unscaled features. Set gamma explicitly to 'auto' or 'scale' to avoid this warning.
"avoid this warning.", FutureWarning)

In [86]:

#Predicting Test and Train data

```
svc_y_test=sc_classifier.predict(x_test)
print(svc_y_test)
```

```
[ 'pos' 'neg' 'neg' 'neg' 'neg' 'neg' 'pos' 'pos' 'neg' 'neg' 'pos' 'pos'
 'neg' 'neg' 'neg' 'neg' 'pos' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg'
 'neg' 'pos' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'pos'
 'neg' 'neg' 'neg' 'pos' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg'
 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg'
 'pos' 'pos' 'neg' 'neg' 'pos' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg'
 'neg' 'pos' 'pos' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg'
 'neg' 'neg' 'neg' 'neg' 'pos' 'neg' 'pos' 'neg' 'neg' 'neg' 'neg' 'pos'
 'neg' 'pos' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg'
 'neg' 'pos' 'neg' 'neg' 'neg' 'neg' 'pos' 'pos' 'neg' 'neg' 'neg' 'neg'
 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'pos' 'neg' 'neg'
 'neg' 'pos' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg'
 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg'
 'neg' 'neg' 'neg' 'neg' 'pos' 'pos' 'neg' 'neg' 'pos' 'neg' 'neg' 'neg'
 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg'
 'neg' 'pos' 'neg' 'pos' 'neg' 'neg' 'neg' 'pos' 'neg' 'neg' 'neg' 'neg'
 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg'
 'neg' 'neg' 'neg' 'neg' 'pos' 'pos' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg'
 'neg' 'neg' 'pos' 'neg' 'neg' 'neg' 'neg' 'neg' 'pos' 'neg' 'neg' 'neg'
 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'pos' 'neg' 'neg' 'pos' 'neg' 'pos'
 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'pos' 'neg' 'neg'
 'neg' 'neg' 'neg' 'pos' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg'
 'neg' 'neg' 'neg' 'pos' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg'
 'neg' 'neg' 'neg' 'neg' 'neg' 'pos' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg'
 'neg' 'neg' 'pos' 'neg' 'neg' 'neg' 'neg' 'neg' 'neg' 'pos' 'neg' 'neg'
 'neg' 'pos' 'neg' 'neg' 'neg' 'pos' 'pos' 'neg']
```

In [87]:

```
#Build a confusion matrix on test and train
from sklearn.metrics import confusion_matrix
svc_cm_test=confusion_matrix(y_test,svc_y_test)
svc_cm_test
```

Out[87]:

```
array([[191, 14],
       [ 72, 31]], dtype=int64)
```

In [88]:

#Accuracy classification score

```
from sklearn.metrics import accuracy_score
svc_acc_test=accuracy_score(y_test,svc_y_test)
svc_acc_test
```

Out[88]:

```
0.7207792207792207
```

In [89]:

```
#DECISION TREE CLASSIFIER

from sklearn.tree import DecisionTreeClassifier
dtc_clf=DecisionTreeClassifier()
dtc_clf.fit(x_train,y_train)
```

Out[89]:

```
DecisionTreeClassifier(class_weight=None, criterion='gini', max_depth=None,
e,
                        max_features=None, max_leaf_nodes=None,
                        min_impurity_decrease=0.0, min_impurity_split=None,
                        min_samples_leaf=1, min_samples_split=2,
                        min_weight_fraction_leaf=0.0, presort=False, random_state=None,
e,
                        splitter='best')
```

In [90]:

#Predicting on test data

```
dtc_y_test=dtc_clf.predict(x_test)
dtc_y_test
```

Out[90]:

```
array(['pos', 'neg', 'neg', 'neg', 'neg', 'neg', 'pos', 'pos', 'neg',
       'neg', 'pos', 'pos', 'neg', 'neg', 'neg', 'neg', 'pos', 'neg',
       'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'pos', 'pos',
       'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg',
       'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg',
       'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'pos', 'pos', 'neg',
       'neg', 'pos', 'neg', 'neg', 'pos', 'neg', 'neg', 'neg', 'neg',
       'neg', 'pos', 'pos', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg',
       'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'pos', 'neg',
       'pos', 'neg', 'neg', 'neg', 'pos', 'neg', 'neg', 'pos', 'neg',
       'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg',
       'neg', 'pos', 'neg', 'neg', 'neg', 'neg', 'neg', 'pos', 'pos',
       'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg',
       'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg',
       'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg',
       'neg', 'pos', 'pos', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg',
       'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg',
       'neg', 'pos', 'pos', 'neg', 'neg', 'neg', 'pos', 'neg', 'neg',
       'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg',
       'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg',
       'neg', 'neg', 'pos', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg',
       'neg', 'neg', 'pos', 'pos', 'neg', 'neg', 'neg', 'neg', 'pos',
       'neg', 'neg', 'neg', 'neg', 'pos', 'neg', 'neg', 'neg', 'pos',
       'pos', 'neg'], dtype=object)
```

In [91]:

Out[91]:

```
array([[181, 24],
       [ 71, 32]], dtype=int64)
```

In [92]:

```
#Accuracy score on test

from sklearn.metrics import accuracy_score
dtc_accu_test=accuracy_score(y_test,dtc_y_test)
dtc_accu_test
```

Out[92]:

0.6915584415584416

In [93]:

```
#build random forest classifier

from sklearn.ensemble import RandomForestClassifier
rmf=RandomForestClassifier(max_depth=3,random_state=0)
rf_classi=RandomForestClassifier()
rf_classi.fit(x_train,y_train)
```

C:\Users\saima\Anaconda3\lib\site-packages\sklearn\ensemble\forest.py:246:
FutureWarning: The default value of n_estimators will change from 10 in ve
rsion 0.20 to 100 in 0.22.

"10 in version 0.20 to 100 in 0.22.", FutureWarning)

Out[93]:

```
RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gin  
i',  
                        max_depth=None, max_features='auto', max_leaf_nodes=None,  
                        min_impurity_decrease=0.0, min_impurity_split=None,  
                        min_samples_leaf=1, min_samples_split=2,  
                        min_weight_fraction_leaf=0.0, n_estimators=10, n_jobs=None,  
                        oob_score=False, random_state=None, verbose=0,  
                        warm_start=False)
```


In [94]:

```
rf_classi_y_test=dtc_clf.predict(x_test)

rf_classi_y_test
```

Out[94]:

```
array(['pos', 'neg', 'neg', 'neg', 'neg', 'neg', 'pos', 'pos', 'neg',
       'neg', 'pos', 'pos', 'neg', 'neg', 'neg', 'neg', 'pos', 'neg',
       'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'pos', 'pos',
       'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'pos', 'neg', 'pos',
       'neg', 'neg', 'neg', 'pos', 'neg', 'neg', 'neg', 'neg', 'neg',
       'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'pos', 'pos', 'neg',
       'neg', 'pos', 'neg', 'neg', 'pos', 'neg', 'neg', 'neg', 'neg',
       'neg', 'pos', 'pos', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg',
       'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'pos', 'neg',
       'pos', 'neg', 'neg', 'neg', 'pos', 'neg', 'neg', 'pos', 'neg',
       'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg',
       'neg', 'pos', 'neg', 'neg', 'neg', 'neg', 'pos', 'pos', 'neg',
       'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'pos', 'neg',
       'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg',
       'neg', 'pos', 'neg', 'pos', 'neg', 'neg', 'neg', 'pos', 'neg',
       'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'pos', 'neg', 'neg',
       'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg',
       'neg', 'pos', 'pos', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg',
       'neg', 'neg', 'pos', 'neg', 'neg', 'neg', 'pos', 'neg', 'neg',
       'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg',
       'neg', 'neg', 'pos', 'neg', 'neg', 'neg', 'neg', 'neg', 'neg',
       'neg', 'neg', 'pos', 'pos', 'neg', 'neg', 'neg', 'neg', 'pos',
       'neg', 'neg', 'neg', 'neg', 'pos', 'neg', 'neg', 'neg', 'pos',
       'pos', 'neg'], dtype=object)
```

In [95]:

```
from sklearn.metrics import confusion_matrix
rf_cm_test=confusion_matrix(y_test,rf_classi_y_test)
rf_cm_test
```

Out[95]:

```
array([[181, 24],
       [ 71, 32]], dtype=int64)
```

In [96]:

```
#Accuracy score on test data
```

```
from sklearn.metrics import accuracy_score  
rf_accu_test=accuracy_score(y_test,rf_classi_y_test)  
rf_accu_test
```

Out[96]:

0.6915584415584416

In []: