Loading Dependencies

In [1]:

```
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
from sklearn.metrics import accuracy_score
```

Importing the Data Set

In [2]:

```
df=pd.read_csv("SPECTF_train.csv")
df.head()
```

Out[2]:

	1	59	52	70	67	73	66	72	61	58	...	66.3	56.1	62	56.2	72.3	62.1	74.2	74.3	64.1	67.4
0	1	72	62	69	67	78	82	74	65	69	...	65	71	63	60	69	73	67	71	56	58
1	1	71	62	70	64	67	64	79	65	70	...	73	70	66	65	64	55	61	41	51	46
2	1	69	71	70	78	61	63	67	65	59	...	61	61	66	65	72	73	68	68	59	63
3	1	70	66	61	66	61	58	69	69	72	...	67	69	70	66	70	64	60	55	49	41
4	1	57	69	68	75	69	74	73	71	57	...	63	58	69	67	79	77	72	70	61	65

5 rows × 45 columns

In [3]:

```
df.shape
```

Out[3]:

```
(79, 45)
```

In [4]:

```
target=df[['1']]
df=df.drop(labels='1',axis=1)
```

In [5]:

```
column_head=[(lambda x,y: "F"+str(x)+y) (x,y) for x in range(1,23) for y in ['R','S']]
```

In [6]:

```
df.columns=column_head
```

In [7]:

```
dft=pd.read_csv("SPECTF_test.csv")
dft.head()
```

Out[7]:

	1	67	68	73	78	65	63	67.1	60	63.1	...	61.2	56.1	76.3	75.1	74.1	77	76.4	74.2	59.1
0	1	75	74	71	71	62	58	70	64	71	...	66	62	68	69	69	66	64	58	57
1	1	83	64	66	67	67	74	74	72	64	...	67	64	69	63	68	54	65	64	43
2	1	72	66	65	65	64	61	71	78	73	...	69	68	68	63	71	72	65	63	58

	1	67	68	73	78	65	63	67.1	60	63.1	...	61.2	56.1	76.3	75.1	74.1	77	76.4	74.2	59.1	(
3	1	62	60	69	61	63	63	70	68	70	...	66	66	58	56	72	73	71	64	49	
4	1	68	63	67	67	65	72	74	72	70	...	70	70	70	67	77	71	77	72	68	

5 rows × 45 columns

In [8]:

```
dft.shape
```

Out[8]:

```
(186, 45)
```

In [9]:

```
test_target=dft[['1']]
dft=dft.drop(labels='1',axis=1)
column_head=[(lambda x,y: "F"+str(x)+y) (x,y) for x in range(1,23) for y in ['R','S']]
dft.columns=column_head
```

In [10]:

```
dft.head(2)
```

Out[10]:

	F1R	F1S	F2R	F2S	F3R	F3S	F4R	F4S	F5R	F5S	...	F18R	F18S	F19R	F19S	F20R	F20S	F21R
0	75	74	71	71	62	58	70	64	71	68	...	66	62	68	69	69	66	64
1	83	64	66	67	67	74	74	72	64	68	...	67	64	69	63	68	54	65

2 rows × 44 columns

In [11]:

```
df.head(1)
```

Out[11]:

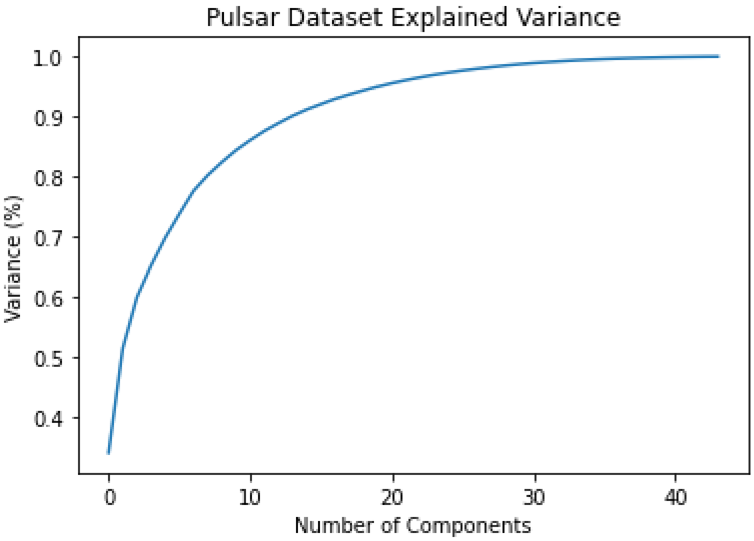
	F1R	F1S	F2R	F2S	F3R	F3S	F4R	F4S	F5R	F5S	...	F18R	F18S	F19R	F19S	F20R	F20S	F21R
0	72	62	69	67	78	82	74	65	69	63	...	65	71	63	60	69	73	67

1 rows × 44 columns

Principal Component Analysis

In [12]:

```
from sklearn.decomposition import PCA
pca = PCA().fit(df)
#Plotting the Cumulative Summation of the Explained Variance
plt.figure()
plt.plot(np.cumsum(pca.explained_variance_ratio_))
plt.xlabel('Number of Components')
plt.ylabel('Variance (%)') #for each component
plt.title('Pulsar Dataset Explained Variance')
plt.show()
```



In [13]:

```
pc=PCA(n_components=34,svd_solver='randomized').fit(df)
```

In [14]:

```
x_train=pca.transform(df)
```

In [15]:

```
x_train
```

Out[15]:

```
array([[-2.25153691e+01,  4.43400602e+00,  1.09025298e+01, ...,
         2.36547079e-01, -1.90600607e-01, -1.54656216e-01],
       [ 2.40190220e+01, -3.16216506e+01,  1.86551363e+01, ...,
        -8.22678816e-01, -1.57392166e+00, -1.01786125e+00],
       [-2.19734319e+00,  1.43263594e+01, -1.25115570e+01, ...,
         7.13284185e-01, -1.66991206e+00, -1.20327601e+00],
       ...,
       [-2.00323653e+01,  1.40177059e+01, -9.34358705e+00, ...,
        -7.95600589e-01,  5.07720674e-01,  2.11010169e-02],
       [ 6.43860906e+00, -1.74391901e+01, -6.30872656e+00, ...,
        -4.38786023e-01, -7.58102783e-01,  3.83601119e-01],
       [-1.04968178e+01,  8.34998664e+00,  3.96402167e+00, ...,
         7.85465149e-01,  2.21166304e-01, -4.88880228e-01]])
```

In [16]:

```
x_test=pca.transform(dft)
x_test
```

Out[16]:

```
array([[ -1.17530963, -10.80210189,   1.69345964, ...,   0.6992736 ,
          1.08949346,  -0.57320507],
       [  8.4969187 ,  -7.7891098 ,  -1.81119436, ...,  -4.35426022,
          3.98596719,   0.70649879],
       [ -7.50081378,  -5.72300766,  16.19286928, ...,   3.68446802,
         -1.84276515,   0.83848757],
       ...,
       [-30.19021928,  -0.3435545 ,  -8.16100715, ...,  -1.46218476,
          0.79210118,   3.20569344],
       [-26.59534356,  10.57124998,  -7.1436282 , ...,  -2.58869222,
         -1.93432059,  -0.07608877],
       [-11.28520906,  -8.83121585,   9.72864556, ...,  -3.21410934,
          1.31835611,  -0.17280859]])
```

Naive Bayes

In [17]:

```python
from sklearn.naive_bayes import GaussianNB
naive=GaussianNB().fit(x_train,target)
naive_prediction=naive.predict(x_test)
accuracy_score(test_target,naive_prediction)
```

```
C:\Users\Mansoor\anaconda3\envs\segmentation\lib\site-packages\sklearn\utils\validation.
py:63: 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().
  return f(*args, **kwargs)
```

Out[17]:

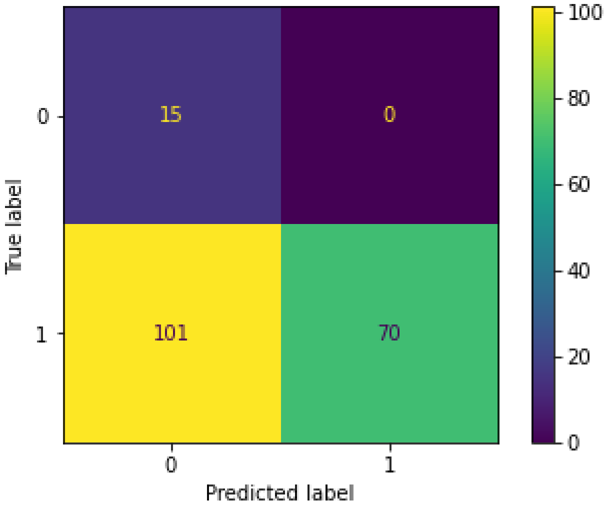
```
0.45698924731182794
```

In [21]:

```python
from sklearn.metrics import confusion_matrix,plot_confusion_matrix
import matplotlib.pyplot as plt
#confusion_matrix(test_target, pred_val)
plot_confusion_matrix(naive,x_test,test_target)
#ConfusionMatrixDisplay.from_predictions(test_target, naive_prediction)
```

Out[21]:

```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x209636f6d68>
```



In [22]:

```python
from sklearn.metrics import classification_report
target_names = ['class 0', 'class 1']
print(classification_report(test_target, naive_prediction, target_names=target_names))
```

```
              precision    recall  f1-score   support

     class 0       0.13      1.00      0.23        15
     class 1       1.00      0.41      0.58       171

    accuracy                           0.46       186
   macro avg       0.56      0.70      0.40       186
weighted avg       0.93      0.46      0.55       186
```

Support Vector Machines

In [23]:

```python
from sklearn import svm
```

Kernel: Linear Kernel ; C=10

In [24]:

```
svc=svm.SVC(kernel='linear',C=10)
svc.fit(x_train,target)
pred_val=svc.predict(x_test)
accuracy_score(test_target,pred_val)
```

```
C:\Users\Mansoor\anaconda3\envs\segmentation\lib\site-packages\sklearn\utils\validation.
py:63: 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().
  return f(*args, **kwargs)
```

Out[24]:

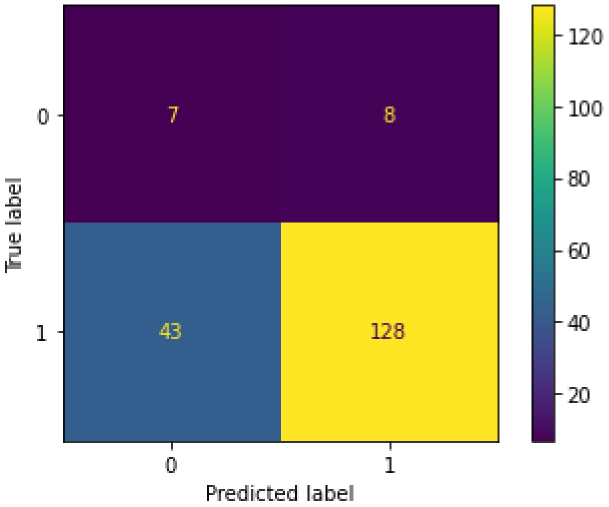
```
0.7258064516129032
```

In [25]:

```
from sklearn.metrics import confusion_matrix
#confusion_matrix(test_target, pred_val)
plot_confusion_matrix(svc,x_test,test_target)
```

Out[25]:

```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x209638b1748>
```



In [26]:

```
from sklearn.metrics import classification_report
target_names = ['class 0', 'class 1']
print(classification_report(test_target, pred_val, target_names=target_names))
```

```
              precision    recall  f1-score   support

     class 0       0.14      0.47      0.22        15
     class 1       0.94      0.75      0.83       171

    accuracy                           0.73       186
   macro avg       0.54      0.61      0.52       186
weighted avg       0.88      0.73      0.78       186
```

Kernel: RBF ; C=940 ; Gamma = 0.004

In [27]:

```
svc=svm.SVC(kernel='rbf',gamma=0.0034,C=1)
svc.fit(df,target)
```

```
pred_val=svc.predict(dft)
accuracy_score(test_target,pred_val)
```

```
C:\Users\Mansoor\anaconda3\envs\segmentation\lib\site-packages\sklearn\utils\validation.
py:63: 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().
  return f(*args, **kwargs)
```

Out[27]:
```
0.8118279569892473
```

In [31]:
```
from sklearn.metrics import confusion_matrix
confusion_matrix(test_target, pred_val)
#plot_confusion_matrix(svc,x_test,test_target)
```

Out[31]:
```
array([[ 10,   5],
       [ 30, 141]], dtype=int64)
```

In [32]:
```
from sklearn.metrics import classification_report
target_names = ['class 0', 'class 1']
print(classification_report(test_target, pred_val, target_names=target_names))
```

```
              precision    recall  f1-score   support

     class 0       0.25      0.67      0.36        15
     class 1       0.97      0.82      0.89       171

    accuracy                           0.81       186
   macro avg       0.61      0.75      0.63       186
weighted avg       0.91      0.81      0.85       186
```

Kernel: Gaussian Kernel ; C=940 ; Gamma = 0.004

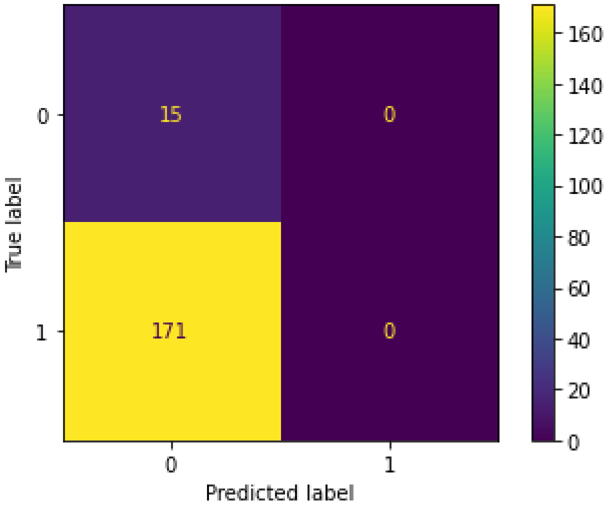
In [33]:
```
from sklearn.gaussian_process.kernels import RBF
gsvc=svm.SVC(kernel=RBF(),C=940,gamma=0.004).fit(x_train,target)
predict_gsvc=gsvc.predict(x_test)
accuracy_score(test_target,predict_gsvc)
```

```
C:\Users\Mansoor\anaconda3\envs\segmentation\lib\site-packages\sklearn\utils\validation.
py:63: 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().
  return f(*args, **kwargs)
```

Out[33]:
```
0.08064516129032258
```

In [34]:
```
from sklearn.metrics import confusion_matrix
#confusion_matrix(test_target, predict_gsvc)
plot_confusion_matrix(gsvc,x_test,test_target)
#ConfusionMatrixDisplay.from_predictions(test_target, predict_gsvc)
```

Out[34]:
```
<sklearn.metrics._plot.confusion_matrix.ConfusionMatrixDisplay at 0x20963a16390>
```



In [35]:
```
from sklearn.metrics import classification_report
target_names = ['class 0', 'class 1']
print(classification_report(test_target, predict_gsvc, target_names=target_names))
```

```
              precision    recall  f1-score   support

     class 0       0.08      1.00      0.15        15
     class 1       0.00      0.00      0.00       171

    accuracy                           0.08       186
   macro avg       0.04      0.50      0.07       186
weighted avg       0.01      0.08      0.01       186

C:\Users\Mansoor\anaconda3\envs\segmentation\lib\site-packages\sklearn\metrics\_classifi
cation.py:1248: UndefinedMetricWarning: Precision and F-score are ill-defined and being
set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control
this behavior.

C:\Users\Mansoor\anaconda3\envs\segmentation\lib\site-packages\sklearn\metrics\_classifi
cation.py:1248: UndefinedMetricWarning: Precision and F-score are ill-defined and being
set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control
this behavior.

C:\Users\Mansoor\anaconda3\envs\segmentation\lib\site-packages\sklearn\metrics\_classifi
cation.py:1248: UndefinedMetricWarning: Precision and F-score are ill-defined and being
set to 0.0 in labels with no predicted samples. Use `zero_division` parameter to control
this behavior.
```

In [ ]:

In [ ]: