In [35]:
```
import pandas as pd
import numpy as np
import seaborn as sns
from sklearn.svm import SVC

import matplotlib.pyplot as plt
import os
print(os.getcwd())
%matplotlib inline
```

```
C:\Users\mvamd\ML
```

Load Train data

In [36]:
```
train_data  = pd.read_csv("C:/Users/mvamd/Downloads/train.csv.zip")
train_data.head()
```

Out[36]:

	id	Mean of the integrated profile	Standard deviation of the integrated profile	Excess kurtosis of the integrated profile	Skewness of the integrated profile	Mean of the DM-SNR curve	Standard deviation of the DM-SNR curve	Excess kurtosis of the DM-SNR curve	Skewness of the DM-SNR curve
0	16330	138.835938	45.453922	-0.096961	0.086516	3.082776	18.529846	7.179262	60.66034
1	5999	130.296875	51.969546	-0.005585	-0.295680	4.299331	23.557530	6.883942	52.33710
2	10228	127.328125	55.471714	0.090976	-0.444422	69.913880	73.240545	0.434744	-1.24902
3	2980	102.453125	44.412987	0.652927	1.101361	2.425585	15.501871	9.213629	106.09740
4	2472	104.921875	41.629431	0.189677	0.725700	3.275084	18.661805	7.389537	65.16969

In [37]:
```
train_data.isnull().sum()
```

Out[37]:
```
id                                              0
Mean of the integrated profile                  0
Standard deviation of the integrated profile    0
Excess kurtosis of the integrated profile       0
Skewness of the integrated profile              0
Mean of the DM-SNR curve                        0
Standard deviation of the DM-SNR curve          0
Excess kurtosis of the DM-SNR curve             0
Skewness of the DM-SNR curve                    0
target_class                                    0
dtype: int64
```

In [38]:
```
train_data.shape
```

Out[38]:
```
(14318, 10)
```

In [39]:
```
train_data.corr()
```

Out[39]:

	id	Mean of the integrated profile	Standard deviation of the integrated profile	Excess kurtosis of the integrated profile	Skewness of the integrated profile	Mean of the DM-SNR curve	Standard deviation of the DM-SNR curve	Excess kurtosis of the DM-SNR curve
id	1.000000	0.012082	-0.002086	-0.009448	-0.008541	-0.006347	-0.006732	0.0034
Mean of the integrated profile	0.012082	1.000000	0.552478	-0.873473	-0.738974	-0.287538	-0.297747	0.2272
Standard deviation of the integrated profile	-0.002086	0.552478	1.000000	-0.526525	-0.543117	0.012791	-0.045540	0.0255
Excess kurtosis of the integrated profile	-0.009448	-0.873473	-0.526525	1.000000	0.946269	0.404575	0.426082	-0.3362
Skewness of the integrated profile	-0.008541	-0.738974	-0.543117	0.946269	1.000000	0.406169	0.411906	-0.3263
Mean of the DM-SNR curve	-0.006347	-0.287538	0.012791	0.404575	0.406169	1.000000	0.797382	-0.6155
Standard deviation of the DM-SNR curve	-0.006732	-0.297747	-0.045540	0.426082	0.411906	0.797382	1.000000	-0.8085
Excess kurtosis of the DM-SNR curve	0.003443	0.227218	0.025596	-0.336287	-0.326339	-0.615579	-0.808571	1.0000
Skewness of the DM-SNR curve	0.001682	0.139540	0.024178	-0.211153	-0.202633	-0.352638	-0.572731	0.9232
target_class	-0.008594	-0.672304	-0.362363	0.791776	0.711313	0.395620	0.490554	-0.3902

Split data into X and y data

In [40]:
```
X = train_data.iloc[:, :-1]
y = train_data.iloc[:, -1]
```

Scale data

In [41]:
```
# Scale the data to be between -1 and 1
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
scaler.fit(X)
X = scaler.transform(X)
```

Split the data into training and testing datasets

In [42]:
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=1)
print(X_train.shape, X_test.shape, y_train.shape, y_test.shape)
```

```
(11454, 9) (2864, 9) (11454,) (2864,)
```

K-fold cross validation

In [43]:
```
#K=10
from sklearn.model_selection import cross_val_score
svc=SVC(kernel='linear',C=0.1)
svc.fit(X_train,y_train)
scores = cross_val_score(svc, X, y, cv=10, scoring='accuracy')
print(scores)
```

```
[0.97625698 0.9797486  0.9853352  0.97695531 0.97905028 0.98393855
 0.97835196 0.97136872 0.98043326 0.97554158]
```

Accuracy

In [44]:
```
from sklearn import metrics
test1 = svc.predict(X_test)
score = metrics.accuracy_score(y_test, test1)
```

In [45]:
```
print("The accuracy is: ",score)
```

```
The accuracy is:  0.977304469273743
```

Load test data

In [46]:
```
test_data  = pd.read_csv("C:\\Users\\mvamd\\Downloads\\test.csv")
test_data.head()
```

Out[46]:

	id	Mean of the integrated profile	Standard deviation of the integrated profile	Excess kurtosis of the integrated profile	Skewness of the integrated profile	Mean of the DM-SNR curve	Standard deviation of the DM-SNR curve	Excess kurtosis of the DM-SNR curve	Skewness of the DM-SNR curve
0	15	130.960938	51.285872	0.072242	-0.584550	1.940635	13.092250	10.481509	147.756658
1	28	23.703125	41.677774	4.112599	16.687125	12.272575	39.323386	3.710673	14.110997
2	30	113.882812	45.562907	0.064720	0.243603	2.101171	13.008511	10.184925	145.275542
3	35	101.882812	44.301633	0.584880	0.722995	2.645485	19.539765	9.101964	90.419878
4	48	96.335938	43.957060	0.302221	0.693381	1.807692	16.216959	11.322476	144.368633

In [47]:
```
test_data.isnull().sum()
```

Out[47]:
```
id                                              0
Mean of the integrated profile                  0
Standard deviation of the integrated profile    0
Excess kurtosis of the integrated profile       0
Skewness of the integrated profile              0
Mean of the DM-SNR curve                        0
Standard deviation of the DM-SNR curve          0
Excess kurtosis of the DM-SNR curve             0
Skewness of the DM-SNR curve                    0
dtype: int64
```

In [48]:
```
test_data.shape
```

Out[48]:
```
(3580, 9)
```

In [49]:
```
test_data.corr()
```

Out[49]:

	id	Mean of the integrated profile	Standard deviation of the integrated profile	Excess kurtosis of the integrated profile	Skewness of the integrated profile	Mean of the DM-SNR curve	Standard deviation of the DM-SNR curve	Excess kurtosis of the DM-SNR curve
id	1.000000	0.000770	-0.020027	0.007923	0.011200	-0.016730	-0.031661	0.009459
Mean of the integrated profile	0.000770	1.000000	0.525318	-0.875686	-0.738073	-0.346264	-0.345847	0.264170
Standard deviation of the integrated profile	-0.020027	0.525318	1.000000	-0.500620	-0.526275	-0.017839	-0.056450	0.045310
Excess kurtosis of the integrated profile	0.007923	-0.875686	-0.500620	1.000000	0.943515	0.455596	0.461459	-0.361977
Skewness of the integrated profile	0.011200	-0.738073	-0.526275	0.943515	1.000000	0.437031	0.428813	-0.339559
Mean of the DM-SNR curve	-0.016730	-0.346264	-0.017839	0.455596	0.437031	1.000000	0.793013	-0.617767
Standard deviation of the DM-SNR curve	-0.031661	-0.345847	-0.056450	0.461459	0.428813	0.793013	1.000000	-0.815074
Excess kurtosis of the DM-SNR curve	0.009459	0.264170	0.045310	-0.361977	-0.339559	-0.617767	-0.815074	1.000000
Skewness of the DM-SNR curve	-0.004250	0.163526	0.042577	-0.229186	-0.214465	-0.361872	-0.589813	0.926418

In [50]:
```
from sklearn import preprocessing
d=preprocessing.scale(test_data)
test_data=pd.DataFrame(d,columns=test_data.columns)
```

Perform predictions on test data

In [51]:
```
pred=svc.predict(test_data)
pred
```

Out[51]:
```
array([0, 1, 0, ..., 1, 0, 0], dtype=int64)
```

Create csv file of predictions

In [52]:
```
df=pd.DataFrame(zip(test_data['id'],pred), columns = ['id', 'target_class'])
df.to_csv('J003_Predictions.csv', index = False)
```