JADAVPUR UNIVERSITY BE IT 4th year

Name - JATIN SINGH CHUG

Roll number - 001811001074

ML LAB ASSIGNMENT 2

IMPORTING HEADERS

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import scikitplot as skplt
```

PRE-PROCESSING DATA

```
def preprocess(X,y,te_size,label=False,scale=False,pca=False):

    if label:
        from sklearn.preprocessing import LabelEncoder
        y = LabelEncoder().fit_transform(y)

    from sklearn.model_selection import train_test_split
    X_tr,X_te,y_tr,y_te = train_test_split(X,y,test_size=te_size)

    if scale:
        from sklearn.preprocessing import StandardScaler
        sc = StandardScaler()
        X_tr = sc.fit_transform(X_tr)
        X_te = sc.transform(X_te)

    if pca:
        from sklearn.decomposition import PCA
        pca = PCA(n_components='mle')
        X_tr = pca.fit_transform(X_tr)
        X_te = pca.transform(X_te)

    return X_tr,X_te,y_tr,y_te
```

TESTER FUNCTION

```
def tester(classi,X_t,y_t,y_p):

    from sklearn.metrics import classification_report,confusion_matrix,accuracy_score,plot_confusion_matrix
    print("Confusion Matrix")
    print(confusion_matrix(y_t,y_p))

    print('------------------------------------')
    print('------------------------------------')
    print('Preformance Evaluation:')
    print(classification_report(y_t,y_p))

    print('-----------------------------------')
    print('-----------------------------------')

    print('Accuracy Score:')
    print(accuracy_score(y_t,y_p))

    plot_confusion_matrix(classi,X_t,y_t)
    plt.title('Heat map for confusion matrix')
    plt.show()

    y_p_proba = classifier.predict_proba(X_t)

    skplt.metrics.plot_roc(y_t,y_p_proba)
    plt.show()
```

WORKING ON WINE DATASET

IMPORTING WINE DATASET

```
df1 = pd.read_csv('wine.data', header=None)
X = df1.iloc[:,1:]
y = df1.iloc[:,0]
```

WITHOUT PARAMETER TUNING

FOR 70:30 SPLIT

```
X_train,X_test,y_train,y_test = preprocess(X,y,0.3,scale=True,pca=True)
```

FOR 60:40 SPLIT

```
X_train,X_test,y_train,y_test = preprocess(X,y,0.4,scale=True,pca=True)
```

FOR 50:50 SPLIT

```
X_train,X_test,y_train,y_test = preprocess(X,y,0.5,scale=True,pca=True)
```

FOR 40:60 SPLIT

```
X_train,X_test,y_train,y_test = preprocess(X,y,0.6,scale=True,pca=True)
```

FOR 30:70 SPLIT

```
X_train,X_test,y_train,y_test = preprocess(X,y,0.7,scale=True,pca=True)
```

TEST-TRAIN CODE

```
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier

#SVM linear model

classifier = SVC(kernel='linear', probability=True)
classifier.fit(X_train,y_train)

y_pred = classifier.predict(X_test)

print('SVC Linear:')
tester(classifier,X_test,y_test,y_pred)

#SVM polynomial model degree 2
classifier = SVC(kernel='poly', degree=2, probability=True)
classifier.fit(X_train,y_train)

y_pred = classifier.predict(X_test)

print('SVC Polynomial degree 2:')
tester(classifier,X_test,y_test,y_pred)

#SVM polynomial model degree 3
classifier = SVC(kernel='poly', degree=3, probability=True)
classifier.fit(X_train,y_train)

y_pred = classifier.predict(X_test)

print('SVC Polynomial degree 3:')
tester(classifier,X_test,y_test,y_pred)

#SVM gaussian model
classifier = SVC(kernel='rbf', probability=True)
classifier.fit(X_train,y_train)

y_pred = classifier.predict(X_test)

print('SVC Gaussian:')
tester(classifier,X_test,y_test,y_pred)

#SVM sigmoid model
classifier = SVC(kernel='sigmoid', probability=True)
classifier.fit(X_train,y_train)

y_pred = classifier.predict(X_test)

print('SVC Sigmoid:')
tester(classifier,X_test,y_test,y_pred)

#mlp model
from sklearn.neural_network import MLPClassifier
classifier = MLPClassifier(max_iter=500)
classifier.fit(X_train,y_train)

y_pred = classifier.predict(X_test)

print('MLP:')
tester(classifier,X_test,y_test,y_pred)

#random forest model
classifier=RandomForestClassifier()
classifier.fit(X_train,y_train)

y_pred = classifier.predict(X_test)
```

```
print('Random Forest Classifier:')
tester(classifier,X_test,y_test,y_pred)
```

RUN PREPROCESSING CODE -> TESTER CODE-> SPLIT (with desired split ratio) -> TEST-TRAIN CODE

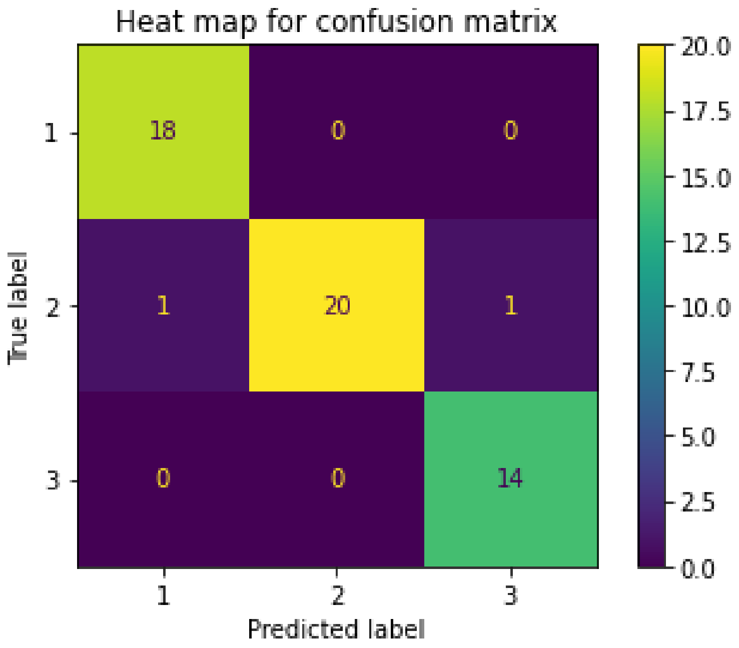
OUTPUT FOR 70:30 SPLIT

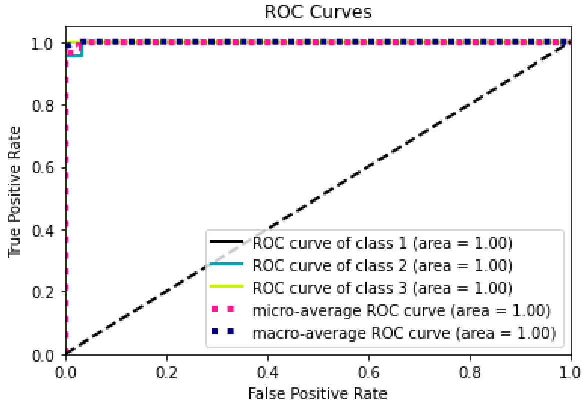
```
SVC Linear:
Confusion Matrix
[[18  0  0]
 [ 1 20  1]
 [ 0  0 14]]
-----------------------------------
-----------------------------------
Preformance Evaluation:
              precision    recall  f1-score   support

           1       0.95      1.00      0.97        18
           2       1.00      0.91      0.95        22
           3       0.93      1.00      0.97        14

    accuracy                           0.96        54
   macro avg       0.96      0.97      0.96        54
weighted avg       0.97      0.96      0.96        54

-----------------------------------
-----------------------------------
Accuracy Score:
0.9629629629629629
```



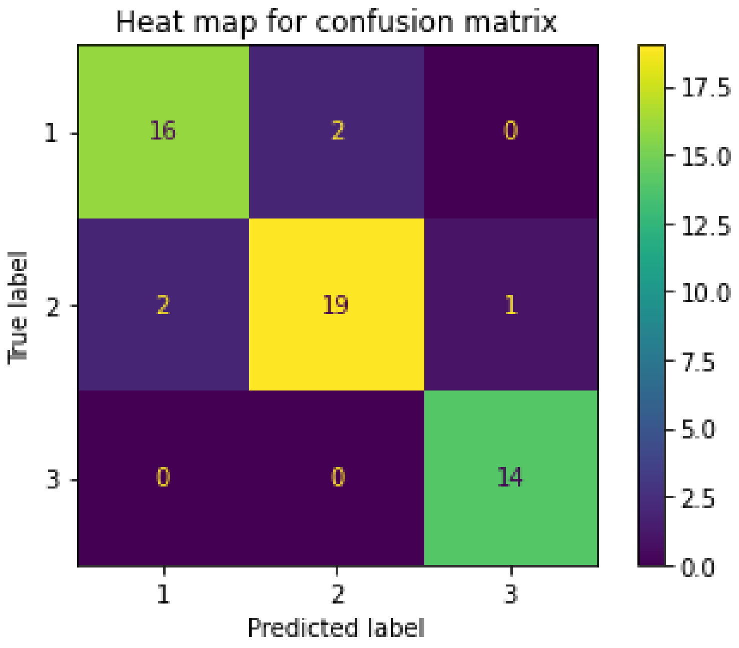


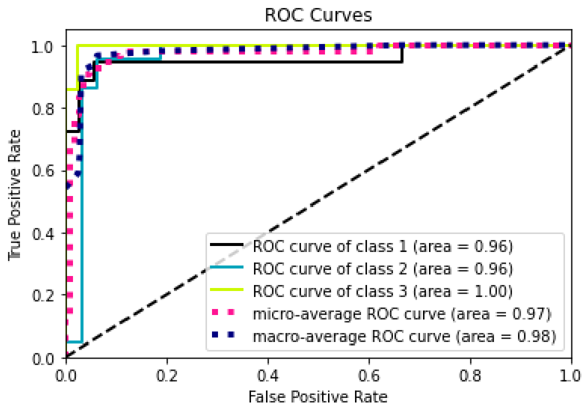
```
SVC Polynomial degree 2:
Confusion Matrix
[[16  2  0]
 [ 2 19  1]
 [ 0  0 14]]
-----------------------------------
-----------------------------------
Preformance Evaluation:
              precision    recall  f1-score   support

           1       0.89      0.89      0.89        18
           2       0.90      0.86      0.88        22
           3       0.93      1.00      0.97        14

    accuracy                           0.91        54
   macro avg       0.91      0.92      0.91        54
weighted avg       0.91      0.91      0.91        54

-----------------------------------
-----------------------------------
Accuracy Score:0.9074074074074074
```



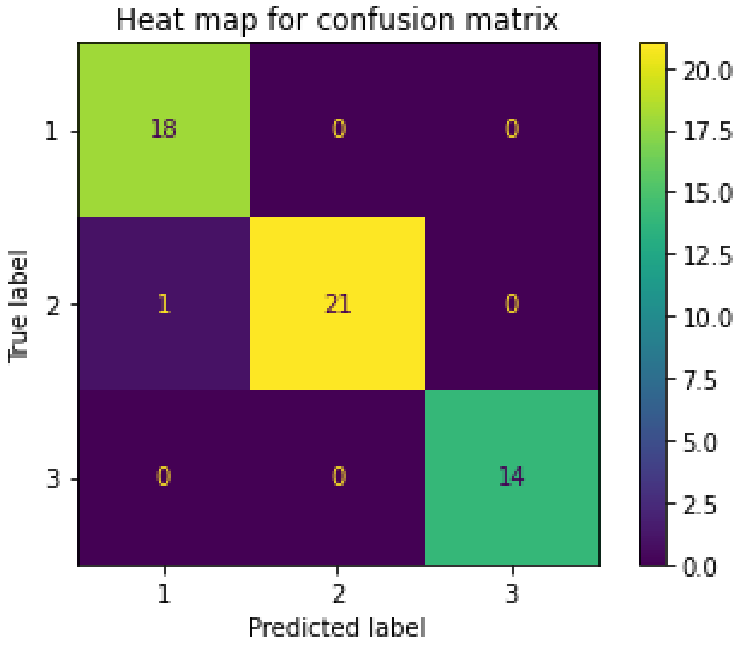


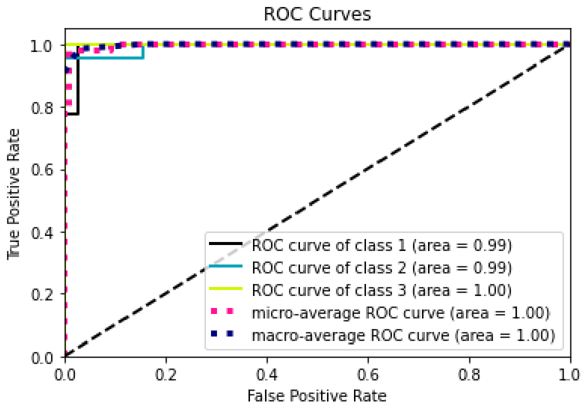
```
SVC Polynomial degree 3:
Confusion Matrix
[[18  0  0]
 [ 1 21  0]
 [ 0  0 14]]
-----------------------------------
-----------------------------------
Preformance Evaluation:
              precision    recall  f1-score   support

           1       0.95      1.00      0.97        18
           2       1.00      0.95      0.98        22
           3       1.00      1.00      1.00        14

    accuracy                           0.98        54
   macro avg       0.98      0.98      0.98        54
weighted avg       0.98      0.98      0.98        54

-----------------------------------
-----------------------------------
Accuracy Score:
0.9814814814814815
```



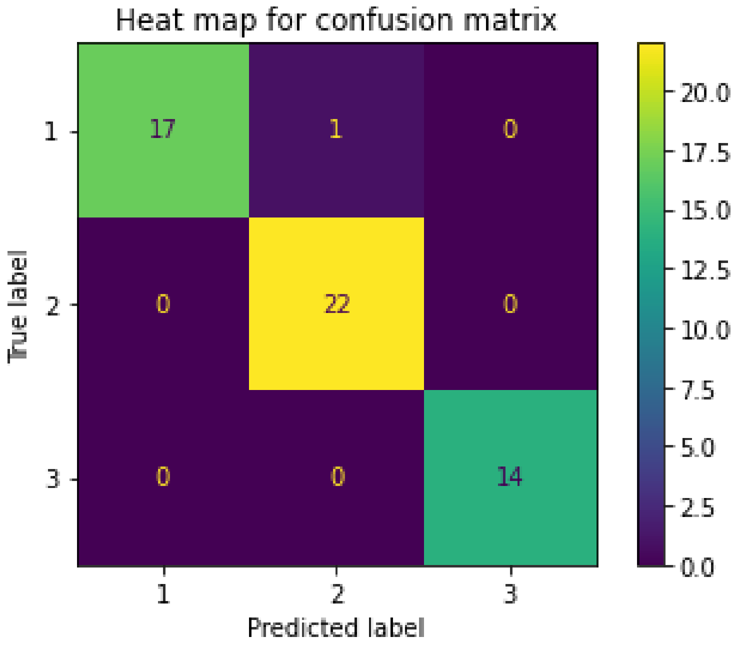


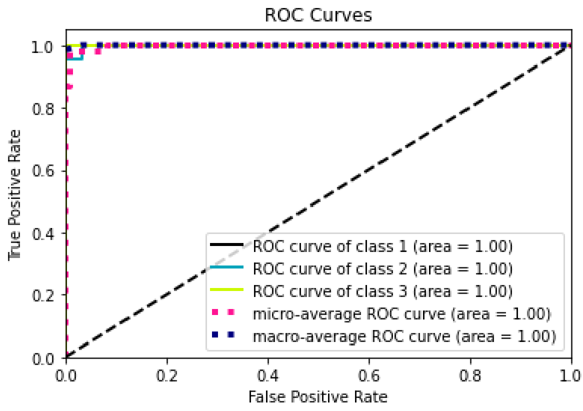
```
SVC Gaussian:
Confusion Matrix
[[17  1  0]
 [ 0 22  0]
 [ 0  0 14]]
-----------------------------------
-----------------------------------
Preformance Evaluation:
              precision    recall  f1-score   support

           1       1.00      0.94      0.97        18
           2       0.96      1.00      0.98        22
           3       1.00      1.00      1.00        14

    accuracy                           0.98        54
   macro avg       0.99      0.98      0.98        54
weighted avg       0.98      0.98      0.98        54

-----------------------------------
-----------------------------------
Accuracy Score:
0.9814814814814815
```



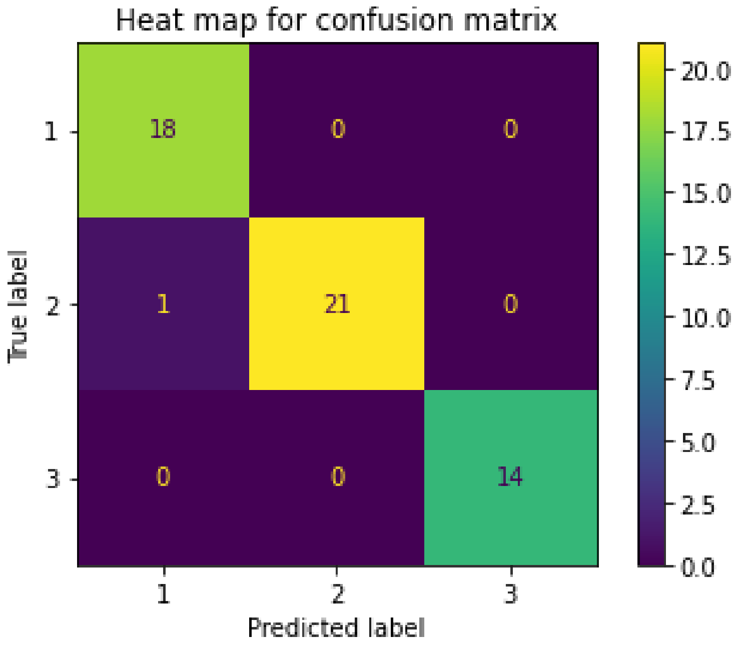


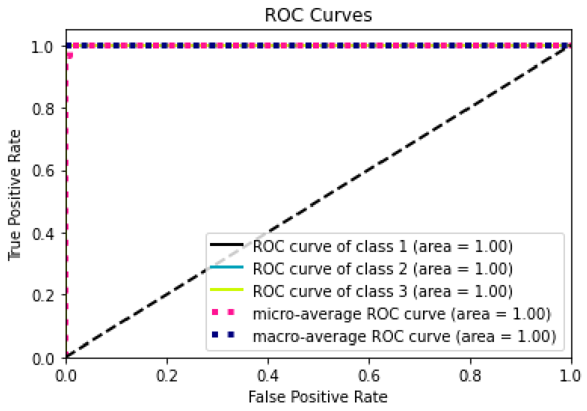
```
SVC Sigmoid:
Confusion Matrix
[[18  0  0]
 [ 1 21  0]
 [ 0  0 14]]
-----------------------------------
-----------------------------------
Preformance Evaluation:
              precision    recall  f1-score   support

           1       0.95      1.00      0.97        18
           2       1.00      0.95      0.98        22
           3       1.00      1.00      1.00        14

    accuracy                           0.98        54
   macro avg       0.98      0.98      0.98        54
weighted avg       0.98      0.98      0.98        54

-----------------------------------
-----------------------------------
Accuracy Score:
0.9814814814814815
```



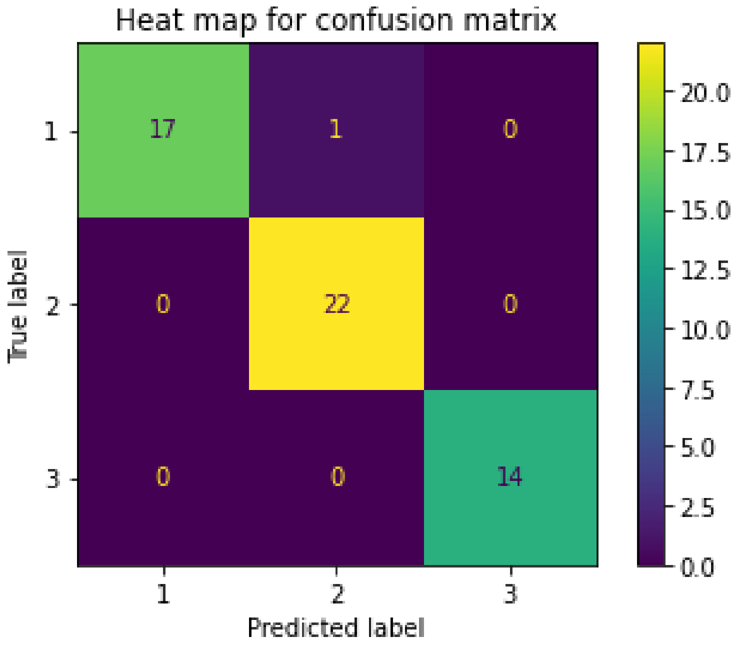


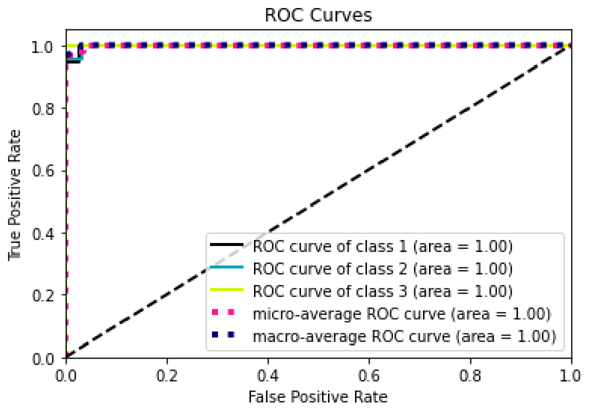
```
MLP:
Confusion Matrix
[[17  1  0]
 [ 0 22  0]
 [ 0  0 14]]
-----------------------------------
-----------------------------------
Preformance Evaluation:
              precision    recall  f1-score   support

           1       1.00      0.94      0.97        18
           2       0.96      1.00      0.98        22
           3       1.00      1.00      1.00        14

    accuracy                           0.98        54
   macro avg       0.99      0.98      0.98        54
weighted avg       0.98      0.98      0.98        54

-----------------------------------
-----------------------------------
Accuracy Score:
0.9814814814814815
```



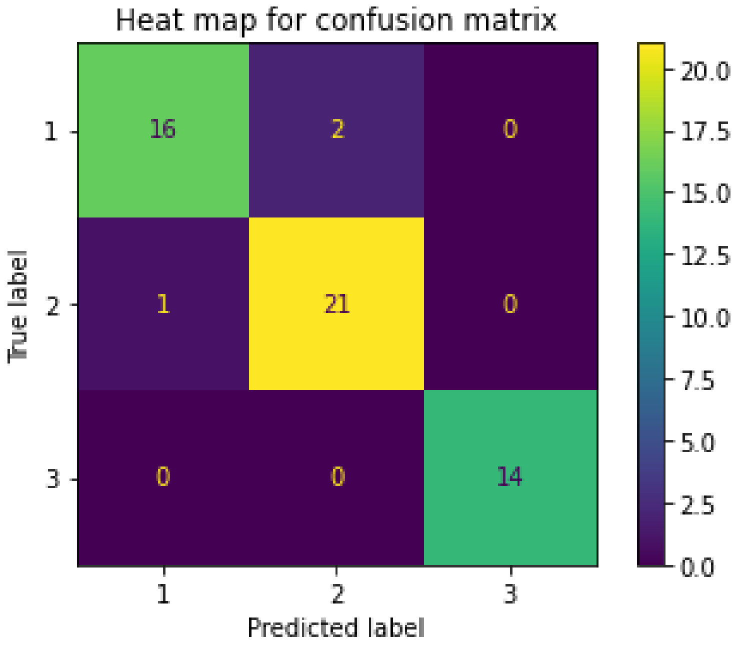


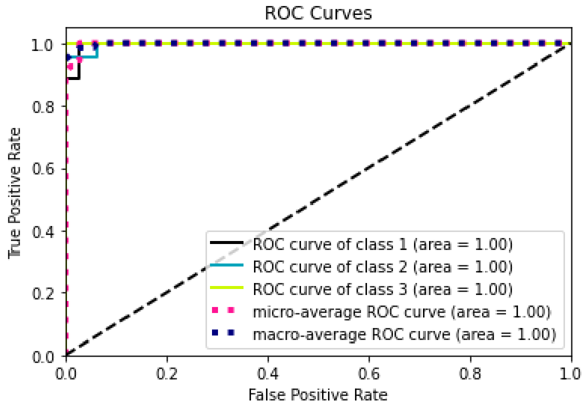
```
Random Forest Classifier:
Confusion Matrix
[[16  2  0]
 [ 1 21  0]
 [ 0  0 14]]
-----------------------------------
-----------------------------------
Preformance Evaluation:
              precision    recall  f1-score   support

           1       0.94      0.89      0.91        18
           2       0.91      0.95      0.93        22
           3       1.00      1.00      1.00        14

    accuracy                           0.94        54
   macro avg       0.95      0.95      0.95        54
weighted avg       0.94      0.94      0.94        54

-----------------------------------
-----------------------------------
Accuracy Score:
0.9444444444444444
```





SIMILARLY BELOW IS A TABLE WITH ALL THE DIFFERENT SPLITS EXECUTED AND THE BEST ACCURACY IS HIGHLIGHTED IN **BOLD.**

SPLIT	SVM LINEAR	SVM POLYNOMIAL DEGREE 2	SVM POLYNOMIAL DEGREE 3	SVM GAUSSIAN	SVM SIGMOID	MLP	RANDOM FOREST CLASSIFIER
70:30	0.962	0.907	**0.981**	**0.981**	**0.981**	**0.981**	0.944
60:40	0.930	0.819	0.888	**0.972**	**0.972**	0.958	0.930
50:50	0.966	0.808	0.932	0.966	0.955	**0.977**	0.966
40:60	**0.962**	0.887	0.897	**0.962**	**0.962**	0.943	0.953
30:70	0.960	0.856	0.936	0.984	**0.992**	0.984	0.976

OBSERVATIONS:

1. For 70:30 split, **SVM Polynomial classifier, SVM Gaussian classifier, SVM Sigmoid** and **MLP** classifier provides the best accuracy of **0.981.**
2. For 60:40 split, **SVM Gaussian** classifier and **SVM Sigmoid** provides the best accuracy **of 0.972.**
3. For 50:50 split, **MLP** classifier provides the best accuracy of **0.977.**
4. For 40:60 split, **SVM Linear classifier, SVM Gaussian** classifier and **SVM Sigmoid** provides the best accuracy of **0.962.**
5. For 30:70 split, **SVM Sigmo**id provides the best accuracy of **0.992.**
6. Out of all the models**, The Sigmoid** Classifier provides the maximum accuracy of all of the above at **0.992.**