Project Development Phase
Performance Test

Date	22 November 2022
Team ID	PNT2022TMID00067
Project Name	Statistical Machine Learning Approaches for Liver Disease Prediction
Maximum Marks	10 Marks

Model Performance Testing:

Project team shall fill the following information in model performance testing template.

S No	Parameter	Values	Screenshot
1.	Metrics	**Classification Model:** Confusion Matrix , Accuracy Score & Classification Report	**Logistic Regression**

Accuracy Of Logistic Regression

```
Accuracy_test_log=accuracy_score(y_test,pred_lr)
Accuracy_test_log
```

```
0.6929824561403509
```

```
report_test_log=classification_report(y_test, pred_lr)
print(report_test_log)
```

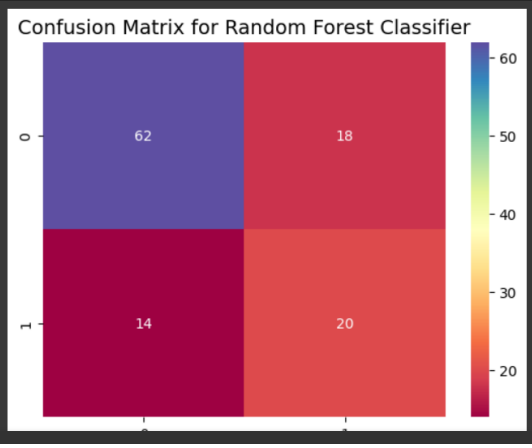
```
              precision    recall  f1-score   support

           1       0.94      0.60      0.73        80
           2       0.49      0.91      0.64        34

    accuracy                           0.69       114
   macro avg       0.72      0.76      0.69       114
weighted avg       0.81      0.69      0.70       114
```

```
[ ]  Accuracy_train_lr

     0.7060702875399361
```

```
[ ]  confusion_matrix(y_test,pred_lr)

     array([[48, 32],
            [ 3, 31]])
```

Random Forest Classifier

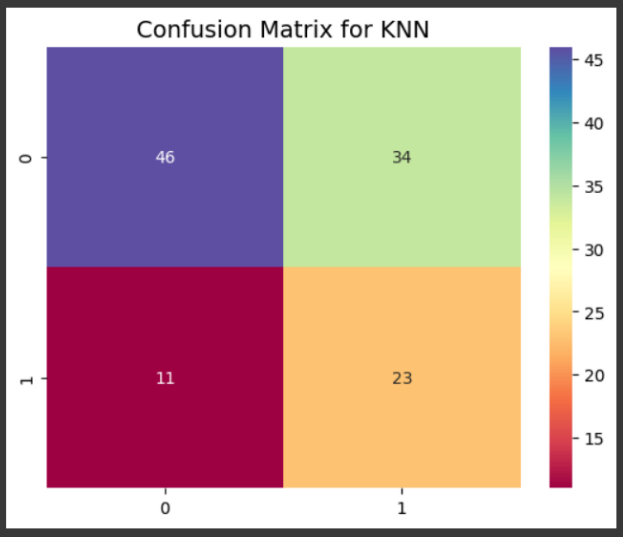
```
[ ] x_test_predict=rfc.predict(x_test)

[ ] RandomForestClassifierScore = rfc.score(x_train,y_train)
    print("Accuracy obtained by Random Forest Classifier model:",RandomForestClassifierScore*100)
    RandomForestClassifierScore = rfc.score(x_test, y_test)
    print("Accuracy obtained by Random Forest Classifier model:",RandomForestClassifierScore*100)

    Accuracy obtained by Random Forest Classifier model: 100.0
    Accuracy obtained by Random Forest Classifier model: 71.9298245614035

[ ] report=classification_report(y_train,x_train_predict)
    print(report)

                  precision    recall  f1-score   support

               1       1.00      1.00      1.00       313
               2       1.00      1.00      1.00       313

        accuracy                           1.00       626
       macro avg       1.00      1.00      1.00       626
    weighted avg       1.00      1.00      1.00       626

    y_pred_rfc = rfc.predict(x_test)
    cf_matrix = confusion_matrix(y_test, x_test_predict)
    sns.heatmap(cf_matrix, annot=True, cmap="Spectral")
    plt.title("Confusion Matrix for Random Forest Classifier", fontsize=14,y=1.03);
```



KNN

```
KNeighborsClassifierScore = KNN.score(x_test, y_test)
print("Accuracy obtained by K Neighbors Classifier model:",KNeighborsClassifierScore*100)

Accuracy obtained by K Neighbors Classifier model: 60.526315789473685

[ ] Accuracy_test_knn=accuracy_score(y_test,pred_KNN)

[ ] Accuracy_test_knn

    0.6052631578947368
```

```
cf_matrix = confusion_matrix(y_test, pred_KNN)
sns.heatmap(cf_matrix, annot=True, cmap="Spectral")
plt.title("Confusion Matrix for KNN", fontsize=14,y=1.03);
```



Decision Tree Classifier

```
[ ] Accuracy_test_DT=accuracy_score(y_test,x_test_pred_DT)
    Accuracy_test_DT
```

```
0.6754385964912281
```

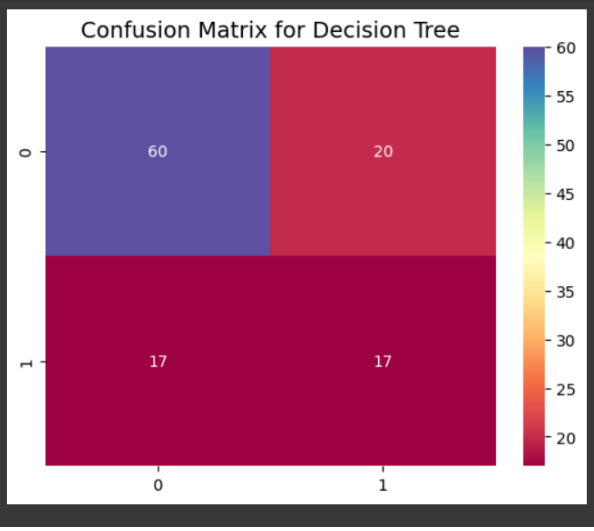
```
report_test_DT=classification_report(y_test, x_test_pred_DT)#
print(report_test_DT)
```

```
              precision    recall  f1-score   support

           1       0.78      0.75      0.76        80
           2       0.46      0.50      0.48        34

    accuracy                           0.68       114
   macro avg       0.62      0.62      0.62       114
weighted avg       0.68      0.68      0.68       114
```

```
[ ] print('Testing score for DT regression is',DT.score(x_test,y_test))
```

```
Testing score for DT regression is 0.6754385964912281
```

```
plt.title("Confusion Matrix for Decision Tree", fontsize=14,y=1.03);
```



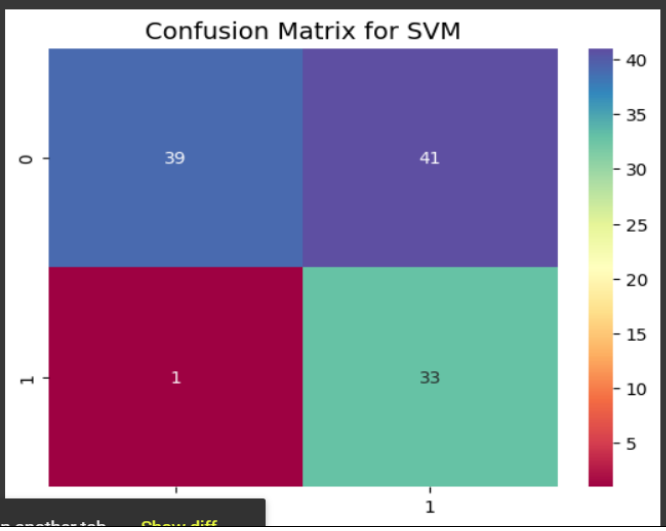
Support Vector Machine

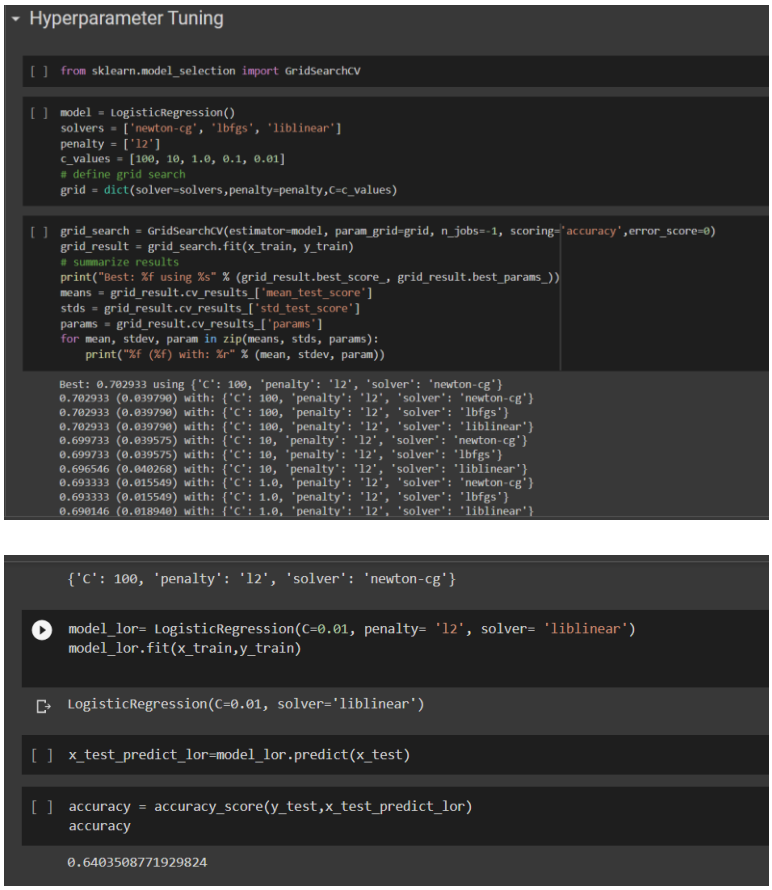
```
[75] x_test_predict=svclassifier.predict(x_test)
```

```
[76] report=classification_report(y_test,x_test_predict)
     print(report)
```

	precision	recall	f1-score	support
1	0.97	0.49	0.65	80
2	0.45	0.97	0.61	34
accuracy			0.63	114
macro avg	0.71	0.73	0.63	114
weighted avg	0.82	0.63	0.64	114

```
cf_matrix = confusion_matrix(y_test, x_test_predict)
sns.heatmap(cf_matrix, annot=True, cmap="Spectral")
plt.title("Confusion Matrix for SVM", fontsize=14,y=1.03);
```



			Conclusion Random Forest Classifier as the highest accuracy but it is overfitting the training data. Therefore, we used Logistic Regression model which gives second best accuracy.
2.	Tune the Model	Hyperparameter Tuning, Validation Method	**Grid Search CV For Logistic Regression**  **Grid Search CV For SVM**

```
[ ] parameters = {'kernel':['linear','rbf'],
                  'C':[0.1,0.5,1.0],
                  'gamma':['scale', 'auto']

    }
```

```
[ ] clf=GridSearchCV(SVC(),param_grid=parameters,verbose=2)
```

```
[ ] clf.fit(x_train,y_train)

    Fitting 5 folds for each of 12 candidates, totalling 60 fits
    [CV] END ..................C=0.1, gamma=scale, kernel=linear; total time=   0.0s
    [CV] END ..................C=0.1, gamma=scale, kernel=linear; total time=   0.0s
    [CV] END ..................C=0.1, gamma=scale, kernel=linear; total time=   0.0s
```

```
[ ] clf.best_score_

    0.6901079365079363
```

```
    clf.best_params_

    {'C': 1.0, 'gamma': 'scale', 'kernel': 'linear'}
```

```
[ ] model_svc=SVC(C=1,gamma='scale',kernel='linear')
    model_svc.fit(x_train,y_train)

    SVC(C=1, kernel='linear')
```

```
[ ] y_=model_svc.predict(x_test)
```

```
[ ] accuracy_score(y_test,y_)

    0.6578947368421053
```

```
    pd.crosstab(y_test,y_)
```

Dataset	col_0 1	2
1	42	38
2	1	33

```
[ ] print(classification_report(y_test,y_))

                  precision    recall  f1-score   support

               1       0.98      0.53      0.68        80
               2       0.46      0.97      0.63        34

        accuracy                           0.66       114
       macro avg       0.72      0.75      0.66       114
    weighted avg       0.82      0.66      0.67       114
```