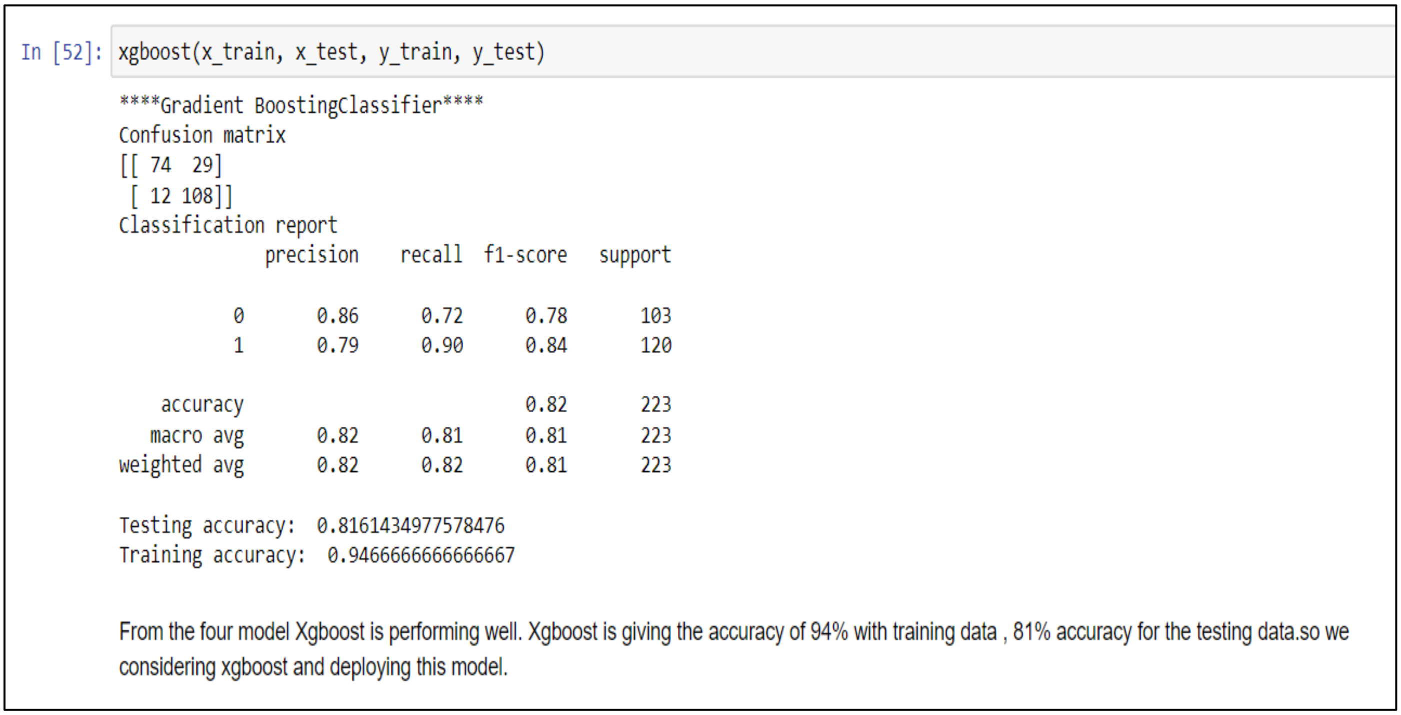
Project Development Phase
Model Performance Test

Date	25 November 2022
Team ID	PNT2022TMID36001
Project Name	Smart Lender - Applicant Credibility Prediction for Loan Approval
Maximum Marks	10 Marks

Model Performance Testing:

In our project we used XG-Boost model for prediction.

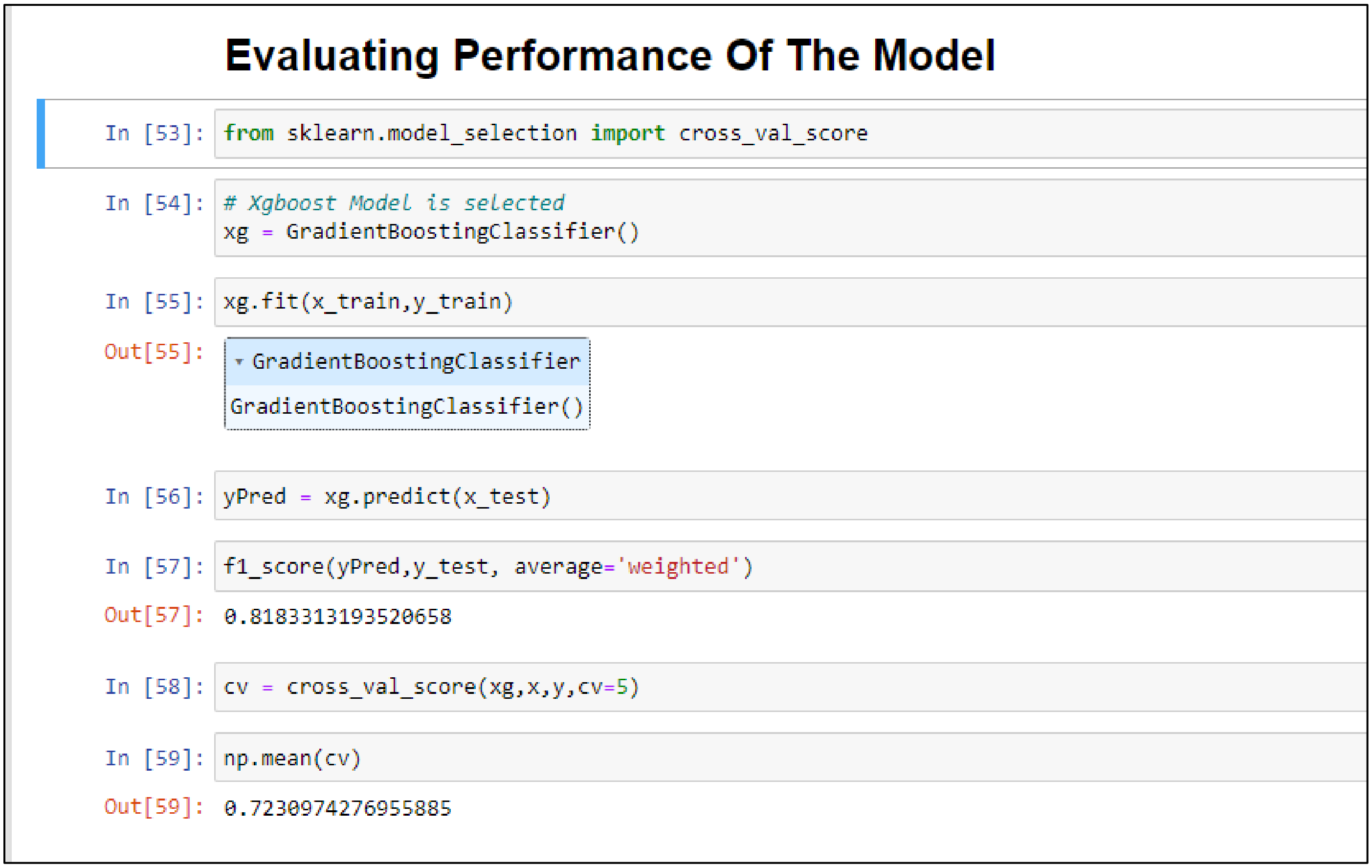
S.No.	Parameter	Values	Screenshot
1.	Metrics	**Classification Model:** Confusion Matrix - , Accuray Score- & Classification Report -	Fig 1
2.	Tune the Model	Hyperparameter Tuning Validation Method	Fig 2



```
In [52]: xgboost(x_train, x_test, y_train, y_test)

         ****Gradient BoostingClassifier****
         Confusion matrix
         [[ 74  29]
          [ 12 108]]
         Classification report
                       precision    recall  f1-score   support

                    0       0.86      0.72      0.78       103
                    1       0.79      0.90      0.84       120

             accuracy                           0.82       223
            macro avg       0.82      0.81      0.81       223
         weighted avg       0.82      0.82      0.81       223

         Testing accuracy:  0.8161434977578476
         Training accuracy:  0.9466666666666667
```

From the four model Xgboost is performing well. Xgboost is giving the accuracy of 94% with training data , 81% accuracy for the testing data.so we considering xgboost and deploying this model.

Fig 1 - Metrics



Evaluating Performance Of The Model

```
In [53]: from sklearn.model_selection import cross_val_score

In [54]: # Xgboost Model is selected
         xg = GradientBoostingClassifier()

In [55]: xg.fit(x_train,y_train)

Out[55]: ▾ GradientBoostingClassifier
         GradientBoostingClassifier()

In [56]: yPred = xg.predict(x_test)

In [57]: f1_score(yPred,y_test, average='weighted')

Out[57]: 0.8183313193520658

In [58]: cv = cross_val_score(xg,x,y,cv=5)

In [59]: np.mean(cv)

Out[59]: 0.7230974276955885
```

Fig 2 - Tune the Model