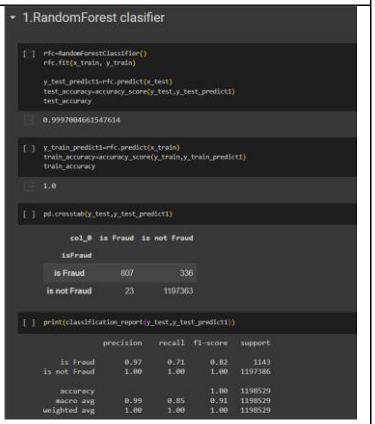
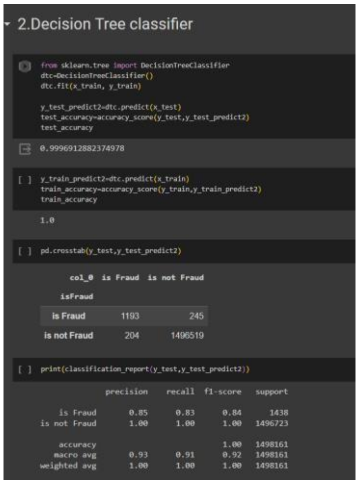
Project Development Phase
Model Performance Test

Date	18 February 2026
Team ID	LTVIP2026TMIDS48224
Project Name	Online Payments Fraud Detection using Machine Learning
Maximum Marks	10 Marks

Model Performance Testing:

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

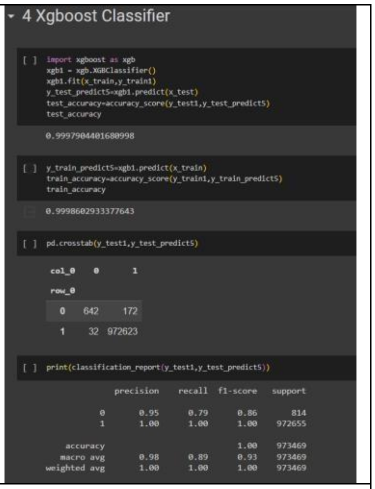
Sl.No.	Parameter	Values	Screenshot
1.	Metrics	**Classification Model:** Confusion Matrix – Accuray Score- Classification Report -	 

1.RandomForest clasifier

```
[ ] rfc=RandomForestClassifier()
    rfc.fit(x_train, y_train)

    y_test_predict1=rfc.predict(x_test)
    test_accuracy=accuracy_score(y_test,y_test_predict1)
    test_accuracy

    0.9997004661547614

[ ] y_train_predict1=rfc.predict(x_train)
    train_accuracy=accuracy_score(y_train,y_train_predict1)
    train_accuracy

    1.0

[ ] pd.crosstab(y_test,y_test_predict1)

          col_0  is Fraud  is not Fraud
        isFraud
       is Fraud       807           336
   is not Fraud        23       1197363

[ ] print(classification_report(y_test,y_test_predict1))

                  precision    recall  f1-score   support

      is Fraud       0.97      0.71      0.82      1143
  is not Fraud       1.00      1.00      1.00   1197386

      accuracy                           1.00   1198529
     macro avg       0.99      0.85      0.91   1198529
  weighted avg       1.00      1.00      1.00   1198529
```

2.Decision Tree classifier

```
    from sklearn.tree import DecisionTreeClassifier
    dtc=DecisionTreeClassifier()
    dtc.fit(x_train, y_train)

    y_test_predict2=dtc.predict(x_test)
    test_accuracy=accuracy_score(y_test,y_test_predict2)
    test_accuracy

    0.9996912882374978

[ ] y_train_predict2=dtc.predict(x_train)
    train_accuracy=accuracy_score(y_train,y_train_predict2)
    train_accuracy

    1.0

[ ] pd.crosstab(y_test,y_test_predict2)

          col_0  is Fraud  is not Fraud
        isFraud
       is Fraud      1193           245
   is not Fraud       204       1496519

[ ] print(classification_report(y_test,y_test_predict2))

                  precision    recall  f1-score   support

      is Fraud       0.85      0.83      0.84      1438
  is not Fraud       1.00      1.00      1.00   1496723

      accuracy                           1.00   1498161
     macro avg       0.93      0.91      0.92   1498161
  weighted avg       1.00      1.00      1.00   1498161
```

			
2.	Tune the Model	Hyperparameter Tuning -	The accuracy for the model is high without hyperparameter tunning and the type 2 error is also very low.

4 Xgboost Classifier

```
[ ] import xgboost as xgb
    xgb1 = xgb.XGBClassifier()
    xgb1.fit(x_train,y_train1)
    y_test_predict5=xgb1.predict(x_test)
    test_accuracy=accuracy_score(y_test1,y_test_predict5)
    test_accuracy

    0.9997904401680998
```

```
[ ] y_train_predict5=xgb1.predict(x_train)
    train_accuracy=accuracy_score(y_train1,y_train_predict5)
    train_accuracy

    0.9998602933377643
```

```
[ ] pd.crosstab(y_test1,y_test_predict5)
```

col_0	0	1
row_0		
0	642	172
1	32	972623

```
[ ] print(classification_report(y_test1,y_test_predict5))

              precision    recall  f1-score   support

           0       0.95      0.79      0.86       814
           1       1.00      1.00      1.00    972655

    accuracy                           1.00    973469
   macro avg       0.98      0.89      0.93    973469
weighted avg       1.00      1.00      1.00    973469
```