

Project Development Phase

Model

Performance Test

Date	15 FEBRUARY 2026
Team ID	LTVIP2026TMIDS79486
Project Name	Project – Online Fraud Detection System
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 -	<div> <p>1.RandomForest classifier</p> <pre>[] rf=RandomForestClassifier() rf.fit(x_train, y_train) y_test_predict1=rf.predict(x_test) test_accuracy=accuracy_score(y_test,y_test_predict1) test_accuracy 0.999788661547614</pre> <pre>[] y_train_predict1=rf.predict(x_train) train_accuracy=accuracy_score(y_train,y_train_predict1) train_accuracy 1.0</pre> <pre>[] pd.crosstab(y_test,y_test_predict1) col_0 is Fraud is not Fraud is Fraud 807 336 is not Fraud 23 1197363</pre> <pre>[] 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 1197366 accuracy 0.99 0.85 1.00 1198509 macro avg 0.99 0.85 0.93 1198509 weighted avg 1.00 1.00 1.00 1198509</pre> </div> <div> <p>2.Decision Tree classifier</p> <pre>[] 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</pre> <pre>[] y_train_predict2=dtc.predict(x_train) train_accuracy=accuracy_score(y_train,y_train_predict2) train_accuracy 1.0</pre> <pre>[] pd.crosstab(y_test,y_test_predict2) col_0 is Fraud is not Fraud is Fraud 1103 245 is not Fraud 204 1496519</pre> <pre>[] print(classification_report(y_test,y_test_predict2)) precision recall f1-score support is Fraud 0.85 0.83 0.84 1418 is not Fraud 1.00 1.00 1.00 1496723 accuracy 0.93 0.91 0.92 1498141 macro avg 0.93 0.91 0.92 1498141 weighted avg 1.00 1.00 1.00 1498141</pre> </div>

			<div><div>4 Xgboost Classifier</div><pre>[] import xgboost as xgb xgb1 = xgb.XGBClassifier() xgb1.fit(x_train,y_train) y_test_predicts=xgb1.predict(x_test) test_accuracy=accuracy_score(y_test,y_test_predicts) test_accuracy 0.9997904401680998 [] y_train_predicts=xgb1.predict(x_train) train_accuracy=accuracy_score(y_train,y_train_predicts) train_accuracy 0.9998602933777643 [] pd.crosstab(y_test,y_test_predicts) col_0 0 1 row_0 0 642 172 1 32 972623 [] print(classification_report(y_test,y_test_predicts)) precision recall f1-score support 0 0.95 0.79 0.86 814 1 1.00 1.00 1.00 972655 accuracy 0.98 0.89 0.93 973469 macro avg 0.98 0.89 0.93 weighted avg 1.00 1.00 1.00</pre></div>
2.	Tune the Model	Hyperparameter Tuning -	The accuracy for the model is high without hyperparameter tuning and the type 2 error is also very low.