

Project Development Phase

Model Performance Test

Date	18 th February 2026
Team ID	LTVIP2026TMIDS82036
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 – Accuray Score – Classification Report –	<div style="display: flex; justify-content: space-between;"> <div style="width: 45%;"> <pre> • 1.RandomForest classifier [] rfc=RandomForestClassifier() rfc.fit(x_train, y_train) y_test_predict=rfc.predict(x_test) test_accuracy=accuracy_score(y_test,y_test_predict) test_accuracy [] 0.99996061517614 </pre> <pre> [] y_train_predict=rfc.predict(x_train) train_accuracy=accuracy_score(y_train,y_train_predict) train_accuracy [] 1.0 </pre> <pre> [] pd.crosstab(y_test,y_test_predict) col_0 is Fraud is not Fraud is Fraud 807 356 is not Fraud 23 1197363 [] print(classification_report(y_test,y_test_predict)) precision recall f1-score support is Fraud 0.97 0.71 0.82 1143 is not Fraud 1.00 1.00 1.00 1198529 accuracy 0.99 0.81 0.91 1198529 macro avg 0.99 0.81 0.91 1198529 weighted avg 0.99 0.81 0.91 1198529 </pre> </div> <div style="width: 45%;"> <pre> • 2.Decision Tree classifier [] from sklearn.tree import DecisionTreeClassifier dtc=DecisionTreeClassifier() dtc.fit(x_train, y_train) y_test_predict=dtc.predict(x_test) test_accuracy=accuracy_score(y_test,y_test_predict) 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 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 1428 is not Fraud 1.00 1.00 1.00 1496723 accuracy 0.93 0.91 0.92 1498161 macro avg 0.93 0.91 0.92 1498161 weighted avg 0.93 0.91 0.92 1498161 </pre> </div> </div>

			<pre>4 Xgboost Classifier</pre> <pre>[] import xgboost as xgb xgb1 = xgboostClassifier() xgb1.fit(x_train,y_train) y_test_predict=xgb1.predict(x_test) test_accuracy=accuracy_score(y_test,y_test_predict) test_accuracy</pre> <pre>0.999790448168998</pre> <pre>[] y_train_predict=xgb1.predict(x_train) train_accuracy=accuracy_score(y_train,y_train_predict) train_accuracy</pre> <pre>0.9998602933377643</pre> <pre>[] pd.crosstab(y_test,y_test_predict)</pre> <pre>col_0 0 1 row_0 0 642 172 1 32 972623</pre> <pre>[] print(classification_report(y_test,y_test_predict))</pre> <table border="1"> <thead> <tr> <th></th> <th>precision</th> <th>recall</th> <th>f1-score</th> <th>support</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0.95</td> <td>0.79</td> <td>0.86</td> <td>814</td> </tr> <tr> <td>1</td> <td>1.00</td> <td>1.00</td> <td>1.00</td> <td>972655</td> </tr> <tr> <td>accuracy</td> <td></td> <td></td> <td></td> <td>973469</td> </tr> <tr> <td>macro avg</td> <td>0.98</td> <td>0.89</td> <td>0.93</td> <td>973469</td> </tr> <tr> <td>weighted avg</td> <td>1.00</td> <td>1.00</td> <td>1.00</td> <td>973469</td> </tr> </tbody> </table>		precision	recall	f1-score	support	0	0.95	0.79	0.86	814	1	1.00	1.00	1.00	972655	accuracy				973469	macro avg	0.98	0.89	0.93	973469	weighted avg	1.00	1.00	1.00	973469
	precision	recall	f1-score	support																													
0	0.95	0.79	0.86	814																													
1	1.00	1.00	1.00	972655																													
accuracy				973469																													
macro avg	0.98	0.89	0.93	973469																													
weighted avg	1.00	1.00	1.00	973469																													
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.																														