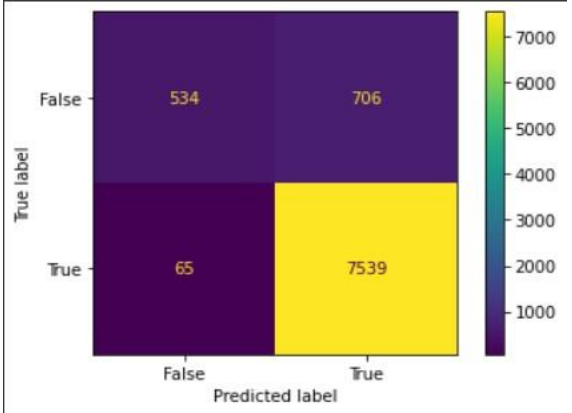
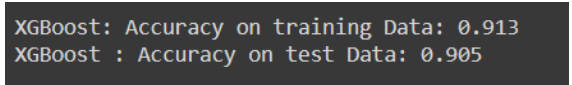


Project Development Phase Model Performance Test

Date	18 November 2022
Team ID	PNT2022TMID27489
Project Name	Project - Web phishing detection
Maximum Marks	10 Marks

S.no	Parameter	Values	Screenshot
1.	Metrics	Classification Model	
		Confusion Matrix <pre> confusion_matrix = metrics.confusion_matrix(y_train,y_train_ xgb) cm_display = metrics.ConfusionMatrixDisplay(confusion _matrix = confusion_matrix, display_labels = [False, True]) cm_display.plot() plt.show() </pre>	
		Accuracy Score <pre> acc_train_xgb = accuracy_score(y_train,y_train_xgb) acc_test_xgb = accuracy_score(y_test,y_test_xgb) print("XGBoost: Accuracy on training Data: {:.3f}".format(acc_train_xgb)) print("XGBoost : Accuracy on test Data: {:.3f}".format(acc_test_xgb)) </pre>	

		Classification Report <pre>classification_report=metrics.classification_report(y_train,y_train_xgb) print(classification_report)</pre>	<pre> precision recall f1-score support -1 0.89 0.43 0.58 1240 1 0.91 0.99 0.95 7604 accuracy 0.91 8844 macro avg 0.90 0.71 0.77 8844 weighted avg 0.91 0.91 0.90 8844</pre>
2.	Tune the model	Hyperparameter tuning <pre>from xgboost import XGBClassifier from sklearn.model_selection import GridSearchCV estimator = XGBClassifier(objective= 'binary:logistic', nthread=4, seed=42) parameters = { 'max_depth': range(2, 10, 1), 'n_estimators': range(60, 220, 40), 'learning_rate': [0.1, 0.01, 0.05] } grid_search = GridSearchCV(estimator=estimator, param_grid=parameters, scoring = 'roc_auc', n_jobs = 10, cv = 10, verbose=True) grid_search.fit(X_train, y_train) grid_search.best_estimator_</pre>	<pre>Fitting 10 folds for each of 96 candidates, totalling 960 fits GridSearchCV(cv=10, estimator=XGBClassifier(nthread=4, seed=42), n_jobs=10, param_grid={'learning_rate': [0.1, 0.01, 0.05], 'max_depth': range(2, 10), 'n_estimators': range(60, 220, 40)}, scoring='roc_auc', verbose=True)</pre> <pre>XGBClassifier(max_depth=9, n_estimators=140, nthread=4, seed=42)</pre>
		Validation method- k-Fold Cross Validation <pre>from sklearn.model_selection import KFold from sklearn.model_selection import cross_val_score kfold = KFold(n_splits=10, random_state=7,shuffle=True) results = cross_val_score(xgb, X_train, y_train, cv=kfold) print("Accuracy: %.2f%% (%.2f%%)" % (results.mean()*100, results.std()*100))</pre>	<pre>Accuracy: 90.39% (0.79%)</pre>