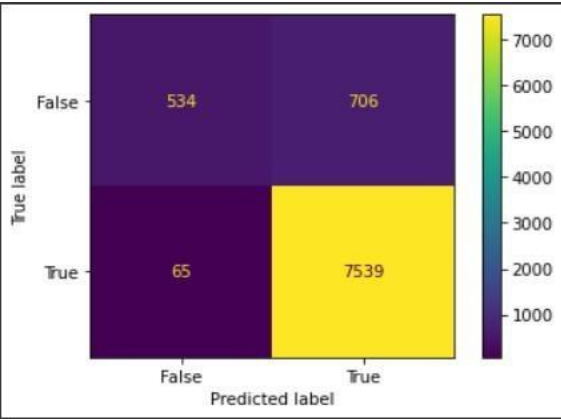
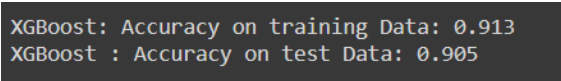


## Project Development Phase Model Performance Test

Date	18 November 2022
Team ID	PNT2022TMID23787
Project Name	Project - Web phishing detection
Maximum Marks	10 Marks

S.no	Parameter	Values	Screenshot
1.	Metrics	<b>Classification Model</b>	
		<b>Confusion Matrix</b>  <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>	
		<b>Accuracy Score</b>  <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>	

		<b>Classification Report</b>  classification_report=metrics.classification_report(y_train,y_train_xgb) print(classification_report)	<table><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr><tr><td>-1</td><td>0.89</td><td>0.43</td><td>0.58</td><td>1240</td></tr><tr><td>1</td><td>0.91</td><td>0.99</td><td>0.95</td><td>7604</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.91</td><td>8844</td></tr><tr><td>macro avg</td><td>0.90</td><td>0.71</td><td>0.77</td><td>8844</td></tr><tr><td>weighted avg</td><td>0.91</td><td>0.91</td><td>0.90</td><td>8844</td></tr></table>		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
	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																													
2.	Tune the model	<b>Hyperparameter tuning</b>  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_	<div>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)</div> <div>XGBClassifier(max_depth=9, n_estimators=140, nthread=4, seed=42)</div>																														
		<b>Validation method- k-Fold Cross Validation</b>  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))	<div>Accuracy: 90.39% (0.79%)</div>																														