Project Development Phase
Model Performance Test

Date	16 November 2022
Team ID	PNT2022TMID26927
Project Name	Project – Web Phishing Detection
Maximum Marks	10 Marks

Model Performance Testing:

S.No.	Parameter	Values	Screenshot

1.	Metrics	**Classification Model:Randon Forest Classifier**
		Confusion Matrix - ,
		Accuray Score- &
		Classification Report -
			Confusion Matrix:
			`In [12]: confusion_matrix(y_pred_ranf, y_test)`
			`Out[12]: array([[ 961, 16], [ 53, 1181]], dtype=int64)`
			Accuracy Score:
			`In [100]:`
			`y_pred_ranf = ranf.predict(x_test)`
			`y_train_rf = ranf.predict(x_train)`
			`test_acc_ranf = accuracy_score(y_test,y_pred_ranf)*100`
			`acc_train_rf = accuracy_score(y_train,y_train_rf)*100`
			`print("Accuracy on training Data: ",acc_train_rf)`
			`print("Accuracy on test Data: ",test_acc_ranf)`
			Accuracy on training Data: 99.02758932609679
			Accuracy on test Data: 97.01492537313433
			Classification Report:
			`In [16]:`
			`from sklearn.metrics import classification_report`
			`print(classification_report(y_test,y_pred_ranf))`
			precision recall f1-score support
			-1 0.98 0.95 0.97 1014
			1 0.96 0.99 0.97 1197
			accuracy 0.97 2211
			macro avg 0.97 0.97 0.97 2211
			weighted avg 0.97 0.97 0.97 2211

2.	Tune the Model	Hyperparameter Tuning using GridSearchCV	**Hyperparameter (GridSearch CV)**
			`In [79]:`
			`# Number of trees in random forest`
			`n_estimators = [100, 150, 200]`
			`# Number of features to consider at every split`
			`max_features = ['auto', 'sqrt']`
			`# Maximum number of levels in tree`
			`max_depth = [20,30,40]`
			`# Minimum number of samples required to split a node`
			`min_samples_split = [2, 5]`
			`# Minimum number of samples required at each leaf node`
			`min_samples_leaf = [1, 2]`
			`# Method of selecting samples for training each tree`
			`bootstrap = [True, False]`
			`param_grid = {'n_estimators': n_estimators, 'max_features': max_features, 'max_depth': max_depth, 'min_samples_split': min_samples_split, 'min_samples_leaf': min_samples_leaf, 'bootstrap': bootstrap}`
			`print(param_grid)`
			`{'n_estimators': [100, 150, 200], 'max_features': ['auto', 'sqrt'], 'max_depth': [20, 30, 40], 'min_samples_split': [2, 5], 'min_samples_leaf': [1, 2], 'bootstrap': [True, False]}`
			`In [80]: hp_rf_model = RandomForestClassifier()`
			`hp_rf_model`
			`Out[80]: RandomForestClassifier()`
			`In [81]:`
			`from sklearn.model_selection import GridSearchCV`
			`rf_Grid = GridSearchCV(estimator = hp_rf_model, param_grid = param_grid, cv = 3, verbose=2, n_jobs = -1)`
			`In [82]: rf_Grid.fit(x_train, y_train)`
			`Fitting 3 folds for each of 144 candidates, totalling 432 fits`
			`Out[82]: GridSearchCV(cv=3, estimator=RandomForestClassifier(), n_jobs=-1, param_grid={'bootstrap': [True, False], 'max_depth': [20, 30, 40], 'max_features': ['auto', 'sqrt'], 'min_samples_leaf': [1, 2], 'min_samples_split': [2, 5], 'n_estimators': [100, 150, 200]}, verbose=2)`
			`In [83]: rf_Grid.best_params_`
			`Out[83]: {'bootstrap': False, 'max_depth': 30, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 150}`
			`In [98]:`
			`print (f'Train Accuracy - : {rf_Grid.score(x_train,y_train):.3f}')`
			`print (f'Test Accuracy - : {rf_Grid.score(x_test,y_test):.3f}')`
			Train Accuracy - : 0.990
			Test Accuracy - : 0.969