

## Project Development Phase Model Performance Test

Date	17 November 2022
Team ID	PNT2022TMID32310
Project Name	Project – Web Phishing Detection
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	<div><div>Regression Model: K-Nearest Neighbors</div><div>Classification Model:<ul style="list-style-type: none"><li>Confusion Matrix - <div><div>[[ 928 50]</div><div>[ 86 1147]]</div></div></li><li>Accuracy Score - 93.848937</li><li>Recall Score - 93.025142</li><li>ROC Score - 93.956334</li><li>Classification Report:<div><div>precision - 0.94</div><div>recall - 0.94</div><div>f1 - score - 0.94</div><div>support - 2211</div></div></li></ul></div></div>	<div><div>• Model building</div><div><pre>In [82]: from sklearn.neighbors import KNeighborsClassifier error = [] for i in range(1, 30):     knn = KNeighborsClassifier(n_neighbors=i)     knn.fit(x_train,y_train)     pred_i = knn.predict(y_test)     error.append(np.mean(pred_i != y_test))  In [83]: knn=KNeighborsClassifier(n_neighbors=1) model_2= knn.fit(x_train,y_train)  In [84]: knn_predict=model_2.predict(y_test)</pre></div><div><div>• Check the metrics of the model</div><div><pre>In [85]: from sklearn.metrics import accuracy_score,recall_score,roc_auc_score,confusion_matrix print("Accuracy Score : %f" %(accuracy_score(knn_predict,y_test)/100)) print("Recall Score : %f" %(recall_score(knn_predict,y_test)/100)) print("ROC Score : %f" %(roc_auc_score(knn_predict,y_test)/100)) print(confusion_matrix(knn_predict,y_test))  Accuracy Score : 93.848937 Recall Score : 93.025142 ROC Score : 93.956334  [[ 928  50]  [  86 1147]]</pre></div></div><div><div>• Classification Report</div><div><pre>In [86]: print(classification_report(knn_predict,y_test))</pre><table><thead><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr></thead><tbody><tr><td>-1</td><td>0.92</td><td>0.95</td><td>0.93</td><td>978</td></tr><tr><td>1</td><td>0.90</td><td>0.93</td><td>0.94</td><td>1233</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.94</td><td>2211</td></tr><tr><td>macro avg</td><td>0.94</td><td>0.94</td><td>0.94</td><td>2211</td></tr><tr><td>weighted avg</td><td>0.94</td><td>0.94</td><td>0.94</td><td>2211</td></tr></tbody></table></div></div></div>		precision	recall	f1-score	support	-1	0.92	0.95	0.93	978	1	0.90	0.93	0.94	1233	accuracy			0.94	2211	macro avg	0.94	0.94	0.94	2211	weighted avg	0.94	0.94	0.94	2211
	precision	recall	f1-score	support																													
-1	0.92	0.95	0.93	978																													
1	0.90	0.93	0.94	1233																													
accuracy			0.94	2211																													
macro avg	0.94	0.94	0.94	2211																													
weighted avg	0.94	0.94	0.94	2211																													

2.	Tune the Model	<div>Hyperparameter Tuning - (GridSearchCV)</div> <div>clf.best_score_ - 0.9315919723544503</div> <div>Validation Method - GridSearchCV (estimator=SVC())</div>	<div><pre>In [87]: from sklearn.svm import SVC model = SVC()  In [88]: model.fit(x_train,y_train)  Out[88]: SVC()</pre></div> <div><div>• Hyperparameter Tuning</div><div>GridsearchCV</div><pre>In [90]: from sklearn.model_selection import GridSearchCV  In [91]: parameters = {'kernel': ['linear', 'rbf'],                       'C': [0.1, 1, 10, 50, 100],                       'gamma': ['scale', 'auto']}  In [92]: clf=GridSearchCV(SVC(),param_grid=parameters,verbose=2)  In [93]: clf.fit(x_train,y_train)  Fitting 5 folds for each of 12 candidates, totalling 60 fits [CV] BMO .....C=0.1, gammascale, kernel=linear; total time= 0.8s [CV] BMO .....C=0.1, gammascale, kernel=linear; total time= 0.7s [CV] BMO .....C=0.1, gammascale, kernel=linear; total time= 0.7s</pre></div>
----	----------------	---	---