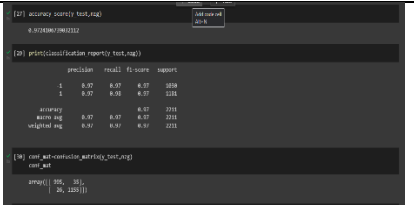
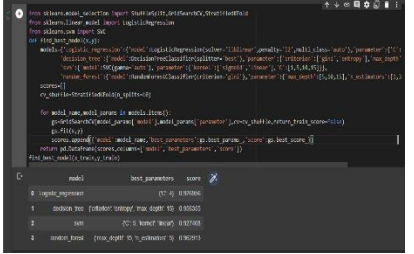


## Project Development Phase Model Performance Test

Date	10 November 2022
Team ID	PNT2022TMID8575
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	<b>Classification Model:</b> Confusion Matrix - , Accuracy Score- & Classification Report -	 <pre> [21]: accuracy_report = test_report         print(accuracy_report)          # Confusion Matrix         [22]: print(classification_report(y_test, y_pred))                precision    recall  f1-score   support           0.         0.97      0.97      0.97       1000          1.         0.99      0.98      0.99        111     accuracy: 0.97      0.97      0.97       2111    macro avg: 0.97      0.97      0.97       2111    weighted avg: 0.97      0.97      0.97       2111  [23]: conf_mat = confusion_matrix(y_test, y_pred)         conf_mat  array([[ 970,  30],        [  10,  11]]) </pre>																				
2.	Tune the Model	Hyperparameter Tuning - Validation Method -	 <pre> from sklearn.metrics import grid_search, cross_val_score, cross_val_predict from sklearn.linear_model import LogisticRegression from sklearn.datasets import load_iris from sklearn.model_selection import GridSearchCV  # Load the dataset iris = load_iris() X = iris.data y = iris.target  # Define the model model = LogisticRegression()  # Define the parameter grid param_grid = {'C': [0.1, 1, 10, 100, 1000],               'max_depth': [None, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50]}  # Create the GridSearchCV object grid_search = GridSearchCV(model, param_grid, cv=5, n_jobs=-1)  # Fit the model grid_search.fit(X, y)  # Print the best parameters and score print("Best parameters found: %s" % grid_search.best_params_) print("Best score: %s" % grid_search.best_score_)  # Print the cross-validated scores print("Cross-validated scores: %s" % cross_val_score(model, X, y, cv=5)) </pre> <table border="1"> <thead> <tr> <th>id</th> <th>model</th> <th>test parameters</th> <th>score</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>LogisticRegression</td> <td>C: 1000</td> <td>0.970000</td> </tr> <tr> <td>1</td> <td>Decision Tree</td> <td>max_depth: 10</td> <td>0.950000</td> </tr> <tr> <td>2</td> <td>SVM</td> <td>C: 1000</td> <td>0.970000</td> </tr> <tr> <td>3</td> <td>Random Forest</td> <td>max_depth: 10, n_estimators: 100</td> <td>0.960000</td> </tr> </tbody> </table>	id	model	test parameters	score	0	LogisticRegression	C: 1000	0.970000	1	Decision Tree	max_depth: 10	0.950000	2	SVM	C: 1000	0.970000	3	Random Forest	max_depth: 10, n_estimators: 100	0.960000
id	model	test parameters	score																				
0	LogisticRegression	C: 1000	0.970000																				
1	Decision Tree	max_depth: 10	0.950000																				
2	SVM	C: 1000	0.970000																				
3	Random Forest	max_depth: 10, n_estimators: 100	0.960000																				