Project Development Phase
Model Performance Test

Date	18 November 2022
Team ID	PNT2022TMID44312
Project Name	Developing a Flight Delay Prediction Model using Machine Learning
Maximum Marks	10 Marks

Model Performance Testing:

S.No.	Parameter	Values	Screenshot
1.	Metrics	**Classification Model:** Confusion Matrix - , Accuray Score- & Classification Report -	***Classification Report***

```
print(classification_report(Y_test, Y_pred_log_test))
```

```
              precision    recall  f1-score   support

         0.0       0.96      0.94      0.95      1985
         1.0       0.60      0.73      0.66       262

    accuracy                           0.91      2247
   macro avg       0.78      0.83      0.81      2247
weighted avg       0.92      0.91      0.92      2247
```

Accuracy, Precision, Recall, F1 Score

```
acc_log = accuracy_score(Y_test, Y_pred_log_test)
prec_log, rec_log, f1_log, sup_log = precision_recall_fscore_support(Y_test, Y_pred_log_test)
print('Accuracy Score =', acc_log)
print('Precision =', prec_log[0])
print('Recall =', rec_log[0])
print('F1 Score =', f1_log[0])
```

```
Accuracy Score = 0.9127725856697819
Precision = 0.9632314862765406
Recall = 0.9370277078085643
F1 Score = 0.9499489274770173
```

Checking for Overfitting and Underfitting

```
log_train_acc = accuracy_score(Y_train, Y_pred_log_train)
log_test_acc = accuracy_score(Y_test, Y_pred_log_test)
print('Training Accuracy =', log_train_acc)
print('Testing Accuracy =', log_test_acc)
```

```
Training Accuracy = 0.9205253784505788
Testing Accuracy = 0.9127725856697819
```

			Confusion Matrix `pd.crosstab(Y_test.ravel(), Y_pred_log_test)` col_0 0.0 1.0 row_0 0.0 1860 125 1.0 71 191

2.	Tune the Model	Hyperparameter Tuning
		Validation Method -
			Tuning the Hyper Parameters of Logistic Regression
			`parameters = { 'solver':['newton-cg', 'lbfgs', 'liblinear'], 'C':[100, 10, 1.0, 0.1, 0.01], 'penalty':['l2']}`
			In [57]: `tuned_model = GridSearchCV(LogisticRegression(max_iter=800), param_grid=parameters, verbose=2)`
			`tuned_model.fit(X_train, Y_train.ravel())`
			Out[57]: `GridSearchCV(estimator=LogisticRegression(max_iter=800), param_grid={'C': [100, 10, 1.0, 0.1, 0.01], 'penalty': ['l2'], 'solver': ['newton-cg', 'lbfgs', 'liblinear']}, verbose=2)`
			Testing the Tuned Model
			`Y_pred_tun_train = tuned_model.predict(X_train)`
			`Y_pred_tun_test = tuned_model.predict(X_test)`
			`pd.DataFrame(Y_pred_tun_train).value_counts()`
			`0.0 7734`
			`1.0 1250`
			`dtype: int64`
			`pd.DataFrame(Y_pred_tun_test).value_counts()`
			`0.0 1922`
			`1.0 325`
			`dtype: int64`

Evaluating the Tuned Model using Metrics

Classification Report

```
print(classification_report(Y_test, Y_pred_tun_test))
```

```
              precision    recall  f1-score   support

         0.0       0.97      0.94      0.95      1985
         1.0       0.61      0.76      0.68       262

    accuracy                           0.92      2247
   macro avg       0.79      0.85      0.81      2247
weighted avg       0.93      0.92      0.92      2247
```

Accuracy, Precision, Recall, F1 Score

```
acc_tun = accuracy_score(Y_test, Y_pred_tun_test)
prec_tun, rec_tun, f1_tun, sup_tun = precision_recall_fscore_support(Y_test, Y_pred_tun_test)
print('Accuracy Score =', acc_tun)
print('Precision =', prec_tun[0])
print('Recall =', rec_tun[0])
print('F1 Score =', f1_tun[0])
```

```
Accuracy Score = 0.9158878504672897
Precision = 0.9672216441207075
Recall = 0.9365239294710328
F1 Score = 0.9516252879447147
```

Checking for Overfitting and Underfitting

```
tun_train_acc = accuracy_score(Y_train, Y_pred_tun_train)
tun_test_acc = accuracy_score(Y_test, Y_pred_tun_test)
print('Training Accuracy =', tun_train_acc)
print('Testing Accuracy =', tun_test_acc)
```

```
Training Accuracy = 0.9213045414069457
Testing Accuracy = 0.9158878504672897
```

Confusion Matrix

```
pd.crosstab(Y_test.ravel(), Y_pred_tun_test)
```

col_0	0.0	1.0
row_0		
0.0	1859	126
1.0	63	199