

Project Development Phase Model Performance Test

Date	10 November 2022
Team ID	PNT2022TMID15570
Project Name	Project - Developing a Flight Delay Prediction Model using Machine Learning
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	<p>Regression Model: MAE - , MSE - , RMSE - , R2 score -</p> <p>Classification Model: Confusion Matrix - , Accuray Score- & Classification Report -</p>	<p>Classification Report</p> <pre>print(classification_report(Y_test, Y_pred_tun_test))</pre> <pre> precision recall f1-score support 0.0 0.96 0.95 0.96 1974 1.0 0.67 0.72 0.69 273 accuracy 0.92 2247 macro avg 0.81 0.84 0.82 2247 weighted avg 0.93 0.92 0.92 2247 </pre> <p>Accuracy, Precision, Recall, F1 Score</p> <pre>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])</pre> <p>Accuracy Score = 0.92118380623053 Precision = 0.9010456176312936 Recall = 0.9498480243161894 F1 Score = 0.9554140127388534</p> <p>Checking for Overfitting and Underfitting</p> <pre>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)</pre> <p>Training Accuracy = 0.919432285129118 Testing Accuracy = 0.922118380623053</p> <p>Confusion Matrix</p> <pre>pd.crosstab(Y_test.ravel(), Y_pred_tun_test)</pre> <pre> val\pred 0.0 1.0 0.0 1974 0 1.0 273 197 </pre>
2.	Tune the Model	Hyperparameter Tuning - Validation Method -	<p>Tuning the Hyper Parameters of Logistic Regression</p> <pre>parameters = {'solver': ['newton-cg', 'lbfgs', 'liblinear'], 'C': [100, 10, 1.0, 0.1, 0.01], 'penalty': ['l2']}</pre> <pre>tuned_model = GridSearchCV(LogisticRegression(max_iter=800), param_grid=parameters, verbose=2) tuned_model.fit(X_train, Y_train.ravel())</pre> <p>Fitting 5 folds for each of 15 candidates, totalling 75 fits [CV] B0C=100, penalty=l2, solver=newton-cg; total time= 0.4s</p> <p>Testing the Tuned Model</p> <pre>Y_pred_tun_train = tuned_model.predict(X_train) Y_pred_tun_test = tuned_model.predict(X_test)</pre> <pre>pd.DataFrame(Y_pred_tun_train).value_counts()</pre> <pre> 0.0 7772 1.0 1212 dtype: int64 </pre> <pre>pd.DataFrame(Y_pred_tun_test).value_counts()</pre> <pre> 0.0 1951 1.0 296 dtype: int64 </pre>

Evaluating the Tuned Model using Metrics

Classification Report

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

	precision	recall	f1-score	support
0.0	0.96	0.95	0.96	1974
1.0	0.67	0.72	0.69	273
accuracy			0.92	2247
macro avg	0.81	0.84	0.82	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 = 0.922118380623853
Precision = 0.9610456176319836
Recall = 0.9498480243161094
F1 Score = 0.9554140127388534

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 = 0.9294120805129118
Testing Accuracy = 0.922118380623853

There is no big variation in the training and testing accuracy. Therefore, the Tuned Logistic Regression model is not overfit or underfit.

Confusion Matrix

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

	0.0	1.0
0.0	1974	79
1.0	79	197