

Project Development Phase Model Performance Test

Date	19 November 2022
Team ID	PNT2022TMID39920
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 -	<p>Classification Report</p> <pre>print(classification_report(Y_test, Y_pred_log_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>0.0</td><td>0.96</td><td>0.94</td><td>0.95</td><td>1985</td></tr><tr><td>1.0</td><td>0.60</td><td>0.73</td><td>0.66</td><td>262</td></tr><tr><td>accuracy</td><td></td><td></td><td>0.91</td><td>2247</td></tr><tr><td>macro avg</td><td>0.78</td><td>0.83</td><td>0.81</td><td>2247</td></tr><tr><td>weighted avg</td><td>0.92</td><td>0.91</td><td>0.92</td><td>2247</td></tr></tbody></table> <p>Accuracy, Precision, Recall, F1 Score</p> <pre>: 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])</pre> <p>Accuracy Score = 0.9127725856697819 Precision = 0.9632314862765406 Recall = 0.9370277078085643 F1 Score = 0.9499489274770173</p> <p>Checking for Overfitting and Underfitting</p> <pre>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)</pre> <p>Training Accuracy = 0.9205253784505788 Testing Accuracy = 0.9127725856697819</p>		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
	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																													

			<p>Confusion Matrix</p> <pre>pd.crosstab(Y_test.ravel(), Y_pred_log_test)</pre> <table><tr><td>col_0</td><td>0.0</td><td>1.0</td></tr><tr><td>row_0</td><td></td><td></td></tr><tr><td>0.0</td><td>1860</td><td>125</td></tr><tr><td>1.0</td><td>71</td><td>191</td></tr></table>	col_0	0.0	1.0	row_0			0.0	1860	125	1.0	71	191
col_0	0.0	1.0													
row_0															
0.0	1860	125													
1.0	71	191													
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>In [57]: tuned_model = GridSearchCV(LogisticRegression(max_iter=800), param_grid=parameters, verbose=2) tuned_model.fit(X_train, Y_train.ravel())</pre> <pre>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)</pre> <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 7734 1.0 1250 dtype: int64</pre> <pre>pd.DataFrame(Y_pred_tun_test).value_counts()</pre> <pre>0.0 1922 1.0 325 dtype: int64</pre>												

			<div><div>Evaluating the Thned Model using Nletrics</div><div><div>Classification Report</div><div><pre>print(classification_report(Y_test, Y_red_tun_test))</pre><table><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr><tr><td>B.B</td><td>6.97</td><td>B.94</td><td>0.95</td><td>1985</td></tr><tr><td>1.0</td><td>0.61</td><td>B.76</td><td>0.68</td><td>262</td></tr><tr><td>accuracy</td><td></td><td></td><td>8.92</td><td>2247</td></tr><tr><td>macro avg</td><td>0.79</td><td>B.BS</td><td>0.81</td><td>2247</td></tr><tr><td>weighted avg</td><td>6.93</td><td>B.92</td><td>0.92</td><td>2247</td></tr></table><div>Accu/acy Precisn, recall, ri scoe</div><div><pre>print('Accuracy Score =', acc_tun) print('Precision =', prec_tun[8]) print('Recall =', rec_tun[0]) print('F1 Score =', f1_tun[e])</pre><div>Accuracy Score = 0.9158878504672897 fDecislr=0.9672264412%075 Recall = B.9365239294710328 F1 Score =0.951625287944710T</div></div><div><div>Checking for Overfitting and Underfitting</div><div><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><div>Training Accuracy = 0.9213B454*4B694S7 Test1ng Accuracy = B. 915B87B5B4672897</div></div><div><div>Confusion Matrix</div><div><pre>pd.crosstab(Y_test.ravel(), Ypred_tun_test)</pre><table><tr><th>col_0</th><th>0.0</th><th>1.0</th></tr><tr><td>D.0</td><td>1859</td><td>t26</td></tr><tr><td>1.0</td><td>63</td><td>199</td></tr></table></div></div></div></div></div></div>		precision	recall	f1-score	support	B.B	6.97	B.94	0.95	1985	1.0	0.61	B.76	0.68	262	accuracy			8.92	2247	macro avg	0.79	B.BS	0.81	2247	weighted avg	6.93	B.92	0.92	2247	col_0	0.0	1.0	D.0	1859	t26	1.0	63	199
	precision	recall	f1-score	support																																						
B.B	6.97	B.94	0.95	1985																																						
1.0	0.61	B.76	0.68	262																																						
accuracy			8.92	2247																																						
macro avg	0.79	B.BS	0.81	2247																																						
weighted avg	6.93	B.92	0.92	2247																																						
col_0	0.0	1.0																																								
D.0	1859	t26																																								
1.0	63	199																																								