

Project Development Phase Model Performance Test

Date	10 November 2022
Team ID	PNT2022TMID45822
Project Name	Smart Lender - Applicant Credibility Prediction for Loan Approval
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>Classification Model:</div> <div>Confusion Matrix -</div> <table><tr><th>Col_0 \ Loan status</th><th>0</th><th>1</th></tr><tr><th>0</th><td>52</td><td>0</td></tr><tr><th>1</th><td>16</td><td>117</td></tr></table> <div>Accuracy Score-</div> <div>Random Forest Model Testing Accuracy</div> <div>0.9135135135135135</div> <div>Random Forest Model Training Accuracy</div> <div>0.9137529137529138</div> <div>Classification Report -</div> <table><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr><tr><td>0</td><td>1.00</td><td>0.76</td><td>0.87</td><td>68</td></tr><tr><td>1</td><td>0.88</td><td>1.00</td><td>0.94</td><td>117</td></tr></table> <table><tr><td>accuracy</td><td></td><td></td><td>0.91</td><td>185</td></tr><tr><td>macro avg</td><td>0.94</td><td>0.88</td><td>0.90</td><td>185</td></tr><tr><td>weighted avg</td><td>0.92</td><td>0.91</td><td>0.91</td><td>185</td></tr></table>	Col_0 \ Loan status	0	1	0	52	0	1	16	117		precision	recall	f1-score	support	0	1.00	0.76	0.87	68	1	0.88	1.00	0.94	117	accuracy			0.91	185	macro avg	0.94	0.88	0.90	185	weighted avg	0.92	0.91	0.91	185	<div>pd.crosstab(ytest,ypredR)</div> <table><tr><th>col_0</th><th>0</th><th>1</th></tr><tr><th>Loan_Status</th><th></th><th></th></tr><tr><th>0</th><td>52</td><td>0</td></tr><tr><th>1</th><td>16</td><td>117</td></tr></table> <div>print("Random Forest Model Testing Accuracy")</div> <div>print(accuracy_score(ytest,ypredR))</div> <div>print("Random Forest Model Training Accuracy")</div> <div>print(accuracy_score(ytrain,ypred2R))</div> <div>Random Forest Model Testing Accuracy</div> <div>0.9135135135135135</div> <div>Random Forest Model Training Accuracy</div> <div>0.9137529137529138</div> <div>print(classification_report(ypredR,ytest))</div> <table><tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr><tr><td>0</td><td>1.00</td><td>0.76</td><td>0.87</td><td>68</td></tr><tr><td>1</td><td>0.88</td><td>1.00</td><td>0.94</td><td>117</td></tr></table> <table><tr><td>accuracy</td><td></td><td></td><td>0.91</td><td>185</td></tr><tr><td>macro avg</td><td>0.94</td><td>0.88</td><td>0.90</td><td>185</td></tr><tr><td>weighted avg</td><td>0.92</td><td>0.91</td><td>0.91</td><td>185</td></tr></table>	col_0	0	1	Loan_Status			0	52	0	1	16	117		precision	recall	f1-score	support	0	1.00	0.76	0.87	68	1	0.88	1.00	0.94	117	accuracy			0.91	185	macro avg	0.94	0.88	0.90	185	weighted avg	0.92	0.91	0.91	185
Col_0 \ Loan status	0	1																																																																																		
0	52	0																																																																																		
1	16	117																																																																																		
	precision	recall	f1-score	support																																																																																
0	1.00	0.76	0.87	68																																																																																
1	0.88	1.00	0.94	117																																																																																
accuracy			0.91	185																																																																																
macro avg	0.94	0.88	0.90	185																																																																																
weighted avg	0.92	0.91	0.91	185																																																																																
col_0	0	1																																																																																		
Loan_Status																																																																																				
0	52	0																																																																																		
1	16	117																																																																																		
	precision	recall	f1-score	support																																																																																
0	1.00	0.76	0.87	68																																																																																
1	0.88	1.00	0.94	117																																																																																
accuracy			0.91	185																																																																																
macro avg	0.94	0.88	0.90	185																																																																																
weighted avg	0.92	0.91	0.91	185																																																																																

2. Tune the Model

Hyperparameter Tuning - No tuning is performed as we have got 91% accuracy

Parameters used-

n_estimators=5000,max_depth=80,max_features='log2'

Validation Method - In-sample validation

Random Forest Model

```
model=RandomForestClassifier(n_estimators=5000,max_depth=80,max_features='log2')
model.fit(x_res,y_res)

RandomForestClassifier(max_depth=80, max_features='log2', n_estimators=5000)

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

ypred=model.predict(xtest)

ypred2=model.predict(xtrain)
```

```
In [74]: xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.3,random_state=10)
In [75]: xtrain.head()
Out[75]:
```

	Gender	Marital	Dependent	Education	Self_Employed	ApplicantIncome	CreditScore	LoanAmount	Loan_Amount_Term	Credit_His
246	1	0	0	0	1	6500	4220.0	120.0	360.0	
410	1	1	0	1	0	2283	2020.0	110.0	360.0	
128	1	1	1	0	0	22803	0.0	210.0	360.0	
521	1	1	0	0	0	4281	0.0	100.0	360.0	
169	1	1	0	0	1	874	5280.0	160.0	360.0	

```
> In [76]: xtest.head()
Out[76]:
```

	Gender	Marital	Dependent	Education	Self_Employed	ApplicantIncome	CreditScore	LoanAmount	Loan_Amount_Term	Credit_His
396	1	0	0	0	0	2181	2880.0	80.0	360.0	1.00
333	0	0	0	0	0	2180	2880.0	121.0	360.0	0.00
482	1	1	0	0	0	22803	0.0	210.0	360.0	1.00
173	1	1	0	0	0	4278	0.0	100.0	360.0	1.00
876	1	0	0	0	0	4683	1670.0	160.0	360.0	1.00

```
> In [77]: ytrain.head()
Out[77]:
```