Project Development Phase
Model Performance Test

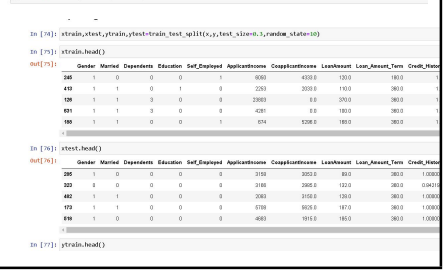
Date	10 November 2022
Team ID	PNT2022TMID39687
Project Name	Smart Lender - Applicant Credibility Prediction for Loan Approval
Maximum Marks	10 Marks

Model Performance Testing:

Project team shall fill the following information in model performance testing template.

S. No.	Parameter	Values	Screenshot

1.	Metrics	**Classification Model:**
		
		Confusion Matrix -
		
		Col_0 / Loan status: 0 → 52, 0; 1 → 16, 117
		
		Accuracy Score-
		Random Forest Model Testing Accuracy
		0.9135135135135135
		Random Forest Model Training Accuracy
		0.9137529137529138
		
		Classification Report -
		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
			`pd.crosstab(ytest,ypredR)`
			col_0 0 1
			Loan_Status
			0 52 0
			1 16 117
			
			`print("Random Forest Model Testing Accuracy")`
			`print(accuracy_score(ytest,ypredR))`
			`print("Random Forest Model Training Accuracy")`
			`print(accuracy_score(ytrain,ypred2R))`
			Random Forest Model Testing Accuracy
			0.9135135135135135
			Random Forest Model Training Accuracy
			0.9137529137529138
			
			`print(classification_report(ypredR,ytest))`
			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 tunning 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
			`Rmodel=RandomForestClassifier(n_estimators=5000,max_depth=80,max_features='log2')`
			`Rmodel.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.
			`ypredR=Rmodel.predict(xtest)`
			`ypred2R=Rmodel.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	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History
245	1	0	0	0	1	6080	4330.0	120.0	180.0	1
413	1	1	0	1	0	2283	3559.0	110.0	360.0	1
126	1	1	3	0	0	23803	0.0	370.0	360.0	1
631	1	1	3	0	0	4281	0.0	100.0	360.0	1
160	1	1	0	0	1	674	5296.0	168.0	360.0	1

In [76]: xtest.head()

Out[76]:

	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanAmount	Loan_Amount_Term	Credit_History
295	1	0	0	0	0	3159	3053.0	89.0	360.0	1.000000
323	0	0	0	0	0	3166	2065.0	122.0	360.0	0.842199
482	1	1	0	0	0	2083	3150.0	128.0	360.0	1.000000
173	1	1	0	0	0	5708	5625.0	187.0	360.0	1.000000
819	1	0	0	0	0	4583	1915.0	185.0	360.0	1.000000

In [77]: ytrain.head()