## Project Development Phase : Model Performance Test

Date	07 November 2022
Team ID	PNT2022TMID40045
Project Name Exploratory Analysis of RainFall Data in India for Agriculture.	
Maximum Marks	4 Marks

## **Model Performance Testing:**

SI. No.	Parameter	Values [Classification Model]	Screenshot
---------	-----------	-------------------------------	------------

	5083] [ 1467  4908]]	<pre>print(accuracy_score(y_test3, y_pred3)) print() print(classification_report(y_test3, y_pred3))</pre>		
		[[17634 5083] [ 1467 4908]]		
Accuracy Score- 0.7748521930427609		0.7748521930427609		
& Classification Report –	& Classification Report –	precision recall f1-score support		
	precision recall f1-score support	0 0.92 0.78 0.84 22717 1 0.49 0.77 0.60 6375		
	0 0.92 0.78 0.84 22717 1 0.49 0.77 0.60 6375	accuracy macro avg 0.71 0.77 0.72 29092 weighted avg 0.83 0.77 0.79 29092		
	accuracy 0.77 29092 macro avg 0.71 0.77 0.72 29092 weighted avg 0.83 0.77 0.79 29092	<pre>metrics.plot_roc_curve(lr, X_test3, y_test3) metrics.roc_auc_score(y_test3, y_pred3, average = None) 0.7730646082617578</pre>		
		True Positive Rate (Positive label: 1) 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.0 0.		
		0.0 LogisticRegression (AUC = 0		

2.		Gaussian Naïve Bayes	v pred1 = o
		Confusion Matrix — [[17179 5538] [ 1700 4675]]	<pre>y_pred1 = g print(confu print() print(accur print() print(class</pre>
		Accuracy Score- 0.7512030798845043	[[17179 55 [ 1700 46
		& Classification Report –	0.751203079
		precision recall f1-score support	
		0 0.91 0.76 0.83 22717 1 0.46 0.73 0.56 6375	
		accuracy 0.75 29092 macro avg 0.68 0.74 0.69 29092 weighted avg 0.81 0.75 0.77 29092	accurac macro av weighted av
			metrics.plo metrics.roc
	Metrics		0.744775571
	Wethos		1.0
			£
			8.0 Per
			Positive 0.6
			P.0 Rate
			True Positive Rate (Positive label: 1)
			0.0
			0.0

```
gnb.predict(X_test1)
fusion_matrix(y_test1, y_pred1))
uracy_score(y_test1, y_pred1))
ssification_report(y_test1, y_pred1))
5538]
4675]]
798845043
                    recall f1-score
     precision
                                           support
                                   0.83
0.56
           0.91
                       0.76
                                             22717
 1
           0.46
                       0.73
                                               6375
acy
avg
                                   0.75
                                             29092
           0.68
                       0.74
                                   0.69
                                             29092
           0.81
                       0.75
                                   0.77
                                             29092
avg
lot_roc_curve(gnb, X_test1, y_test1)
oc_auc_score(y_test1, y_pred1, average = None)
718918284
                                - GaussianNB (AUC = 0.82)
           0.2
                       0.4
                                   0.6
                                               0.8
                                                           1.0
```

False Positive Rate (Positive label: 1)

	Random Forest Classifier		y_pred3 = rfCla	assifier.pr	edict(X_t	est3)	
	Confusion Matrix – [[20552 2165]		<pre>print(confusion print()</pre>	n_matrix(y_	test3, y_	pred3))	
	[ 2470 3905]]		print(accuracy_ print()	_score(y_te	st3, y_pr	ed3))	
	Accuracy Score- 0.8406778495806407		print(classific	cation_repo	rt(y_test	3, y_pred3	))
	& Classification Report – recall f1-score support	precision	[[20552 2165] [ 2470 3905]]				
		20545	0.8406778495800	6407			
	0 0.89 0.90 0.90 1 0.64 0.61 0.63	22717 6375	į.	precision	recall	f1-score	support
	accuracy macro avg 0.77 0.76	0.84 29092 0.76 29092 weighted	0 1	0.89 0.64	0.90 0.61	0.90 0.63	22717 6375
	avg 0.84 0.84 0.84	29092	accuracy macro avg weighted avg	0.77 0.84	0.76 0.84	0.84 0.76 0.84	29092 29092 29092
			metrics.plot_rometrics.roc_aud	oc_curve(rf c_score(y_t	Classifie est1, y_p	r, X_test1 red1, aver	, y_test1) age = None)
Matria			0.684045456154	0247			
Metrics			1.0				
			<del>-</del>				
			8.0 <u>ap</u>	/			
			itive				
			8 0.6 d				
			ate				
			© 0.4				
			Positive Rate (Positive				
			0.2				
			Ĕ				
			0.0		— R	andomForest	Classifier (AUC = 0.
			0.0	0.2 False	0.4 Positive Ra	0.6 ate (Positive	0.8 label: 1)

4.	XGBoost Classifier		<pre>y_pred1 = xgbC.predict(X_test1) print(confusion_matrix(y_test1, y_pred1))</pre>		
	Confusion Matrix –		print()		
	[[21308 1409] [ 2925 3450]]		<pre>print(accuracy_score(y_test1, y_pred1)) print()</pre>		
	[ 2923 3430]]		<pre>print(classification_report(y_test1, y_pred1))</pre>		
	Accuracy Score- 0.8510243365873	378	[[21308 1409] [ 2925 3450]]		
	& Classification Report –	precision	0.851024336587378		
	recall f1-score support	-	precision recall f1-score support		
		.94 0.91 22717 61 6375	0 0.88 0.94 0.91 22717 1 0.71 0.54 0.61 6375		
	1 0.71 0.54 0.	01 6373	accuracy 0.85 29092		
	accuracy	0.85 29092	macro avg 0.79 0.74 0.76 29092 weighted avg 0.84 0.85 0.84 29092		
	macro avg 0.79 0.74 weighted avg 0.84 0	0.76 29092 .85 0.84 29092	weighted drg offer offer 25052		
			<pre>metrics.plot_roc_curve(xgbC, X_test1, y_test1) metrics.roc_auc_score(y_test1, y_pred1, average = None)</pre>		
			0.7395762178622384		
Metr	CS				
			1.0		
			True Positive Rate (Positive label: 1)		
			de 0.0		
			e / l		
			tis 0.6		
			<u>d</u>		
			tate /		
			ψ 0.4		
			£		
			å aa		
			9 0.2		
			F		
			0.0 XGBClassifier (AUC = 0.88)		
			0.0 0.2 0.4 0.6 0.8 1.0		
			False Positive Rate (Positive label: 1)		

5.		Confusion Matrix — [[21416 1301] [ 2891 3484]]  Accuracy Score- 0.855905403547367	<pre>y_pred1 = catBoostC.predict(X_test1) print(confusion_matrix(y_test1, y_pred1)) print() print(accuracy_score(y_test1, y_pred1)) print() print(classification_report(y_test1, y_pred1)) [[21416 1301] [ 2891 3484]]</pre>
		& Classification Report – precision	0.855905403547367
		recall f1-score support	precision recall f1-score support
		0     0.88     0.94     0.91     22717       1     0.73     0.55     0.62     6375	0 0.88 0.94 0.91 22717 1 0.73 0.55 0.62 6375
		accuracy 0.86 29092 macro avg 0.80 0.74 0.77 29092 weighted avg 0.85 0.86 0.85 29092	accuracy
			<pre>metrics.plot_roc_curve(catBoostC, X_test1, y_test1) metrics.roc_auc_score(y_test1, y_pred1, average = None)</pre>
	Metrics		0.744619958966551
			1.0 (F. ideal o.8  O.0  CatBoostClassifier (AUC = 0.88)  0.0  0.0  0.2  0.4  0.6  0.8  1.0  False Positive Rate (Positive label: 1)
6.	Hyperparameters Tuning	iterations = {1000, 2000, 3000,} eval_metric = 'AUC	<pre>catBoostC = CatBoostClassifier(iterations = 3000, eval_metric = 'AUC')</pre>