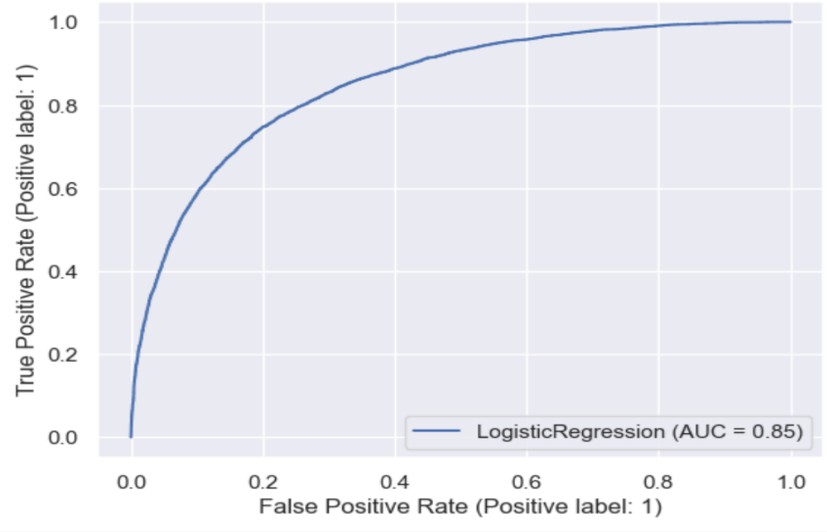
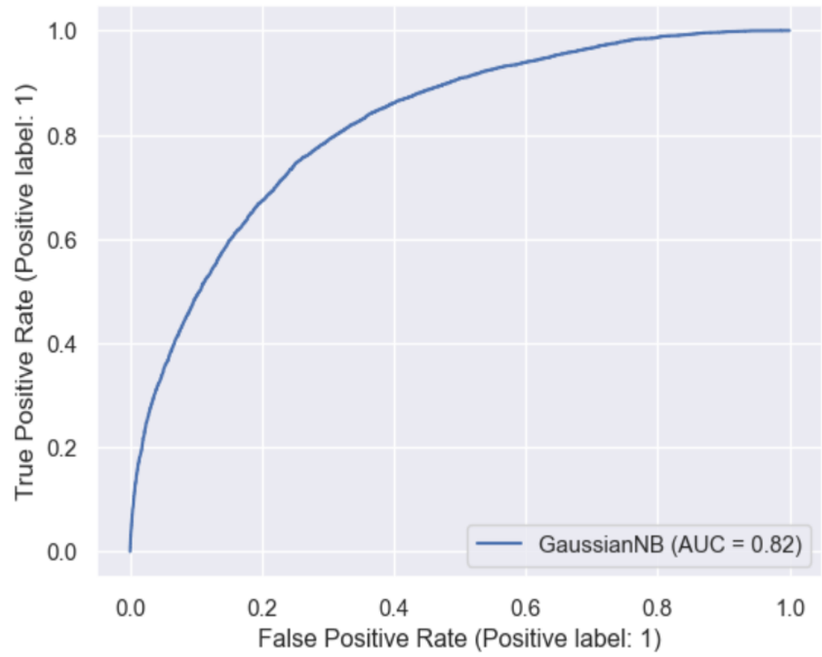
Project Development Phase : Model Performance Test

Date	07 November 2022
Team ID	PNT2022TMID12561
Project Name	Exploratory Analysis of RainFall Data in India for Agriculture.
Maximum Marks	4 Marks

Model Performance Testing:

Sl. No.	Parameter	Values [Classification Model]	Screenshot
1.	Metrics	*Logistic Regression* Confusion Matrix – `[[17634 5083]` ` [ 1467 4908]]` Accuracy Score- `0.7748521930427609` & Classification Report – `precision recall f1-score support` `0 0.92 0.78 0.84 22717` `1 0.49 0.77 0.60 6375` `accuracy 0.77 29092` `macro avg 0.71 0.77 0.72 29092` `weighted avg 0.83 0.77 0.79 29092`	`y_pred3 = lr.predict(X_test3)` `print(confusion_matrix(y_test3, y_pred3))` `print()` `print(accuracy_score(y_test3, y_pred3))` `print()` `print(classification_report(y_test3, y_pred3))` `[[17634 5083]` ` [ 1467 4908]]` `0.7748521930427609` `precision recall f1-score support` `0 0.92 0.78 0.84 22717` `1 0.49 0.77 0.60 6375` `accuracy 0.77 29092` `macro avg 0.71 0.77 0.72 29092` `weighted avg 0.83 0.77 0.79 29092` `metrics.plot_roc_curve(lr, X_test3, y_test3)` `metrics.roc_auc_score(y_test3, y_pred3, average = None)` `0.7730646082617578` 

2.	Metrics	*Gaussian Naïve Bayes* Confusion Matrix – `[[17179  5538]` ` [ 1700  4675]]` Accuracy Score- `0.7512030798845043` & Classification Report – `              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`	``` y_pred1 = gnb.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)) ``` `[[17179  5538]` ` [ 1700  4675]]` `0.7512030798845043` `              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` ``` metrics.plot_roc_curve(gnb, X_test1, y_test1) metrics.roc_auc_score(y_test1, y_pred1, average = None) ``` `0.7447755718918284` 

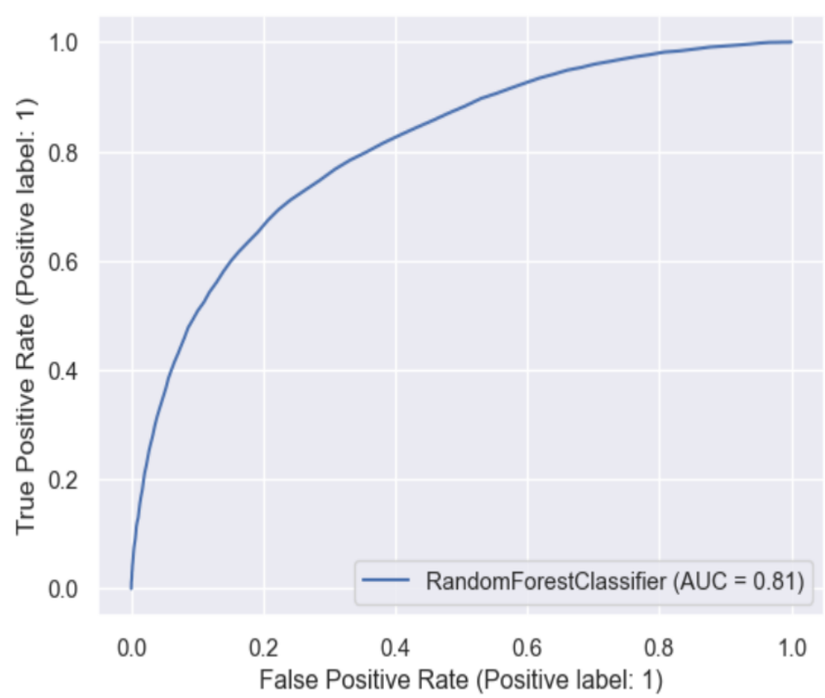
3.	Metrics	Random Forest Classifier Confusion Matrix – [[20552 2165] [2470 3905]] Accuracy Score- 0.8406778495806407 & Classification Report – precision recall f1-score support 0 0.89 0.90 0.90 22717 1 0.64 0.61 0.63 6375 accuracy 0.84 29092 macro avg 0.77 0.76 0.76 29092 weighted avg 0.84 0.84 0.84 29092	

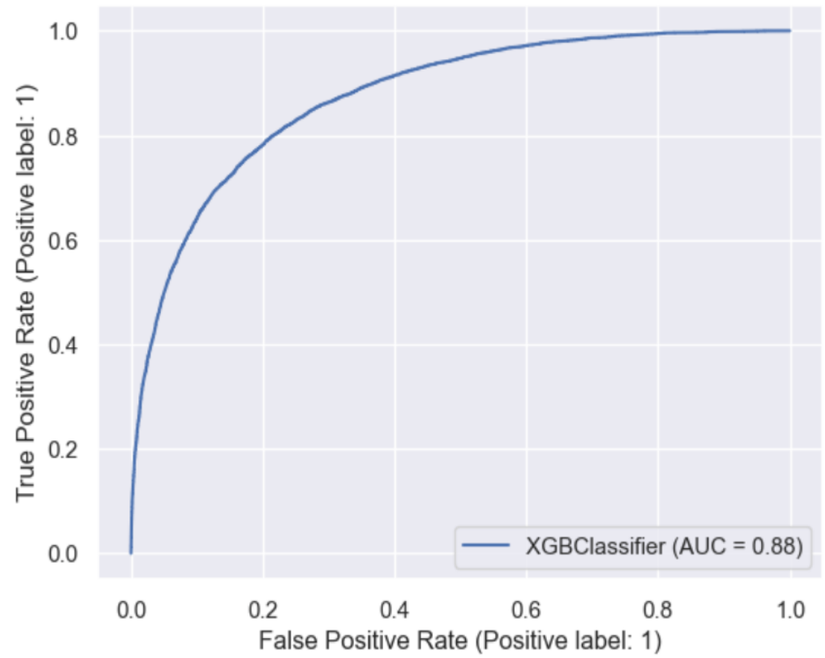
```
y_pred3 = rfClassifier.predict(X_test3)
print(confusion_matrix(y_test3, y_pred3))
print()
print(accuracy_score(y_test3, y_pred3))
print()
print(classification_report(y_test3, y_pred3))
```

```
[[20552  2165]
 [ 2470  3905]]

0.8406778495806407

              precision    recall  f1-score   support

           0       0.89      0.90      0.90     22717
           1       0.64      0.61      0.63      6375

    accuracy                           0.84     29092
   macro avg       0.77      0.76      0.76     29092
weighted avg       0.84      0.84      0.84     29092
```

```
metrics.plot_roc_curve(rfClassifier, X_test1, y_test1)
metrics.roc_auc_score(y_test1, y_pred1, average = None)
```

```
0.6840454561540247
```



4.	Metrics	XGBoost Classifier Confusion Matrix – `[[21308  1409]` ` [ 2925  3450]]` Accuracy Score- `0.851024336587378` & Classification Report – `precision  recall  f1-score  support` `0  0.88  0.94  0.91  22717` `1  0.71  0.54  0.61  6375` `accuracy  0.85  29092` `macro avg  0.79  0.74  0.76  29092` `weighted avg  0.84  0.85  0.84  29092`	`y_pred1 = xgbC.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))` `[[21308  1409]` ` [ 2925  3450]]` `0.851024336587378` `precision  recall  f1-score  support` `0  0.88  0.94  0.91  22717` `1  0.71  0.54  0.61  6375` `accuracy  0.85  29092` `macro avg  0.79  0.74  0.76  29092` `weighted avg  0.84  0.85  0.84  29092` `metrics.plot_roc_curve(xgbC, X_test1, y_test1)` `metrics.roc_auc_score(y_test1, y_pred1, average = None)` `0.7395762178622384` 

<table>
<tr>
<td>5.</td>
<td>Metrics</td>
<td>Cat Boost Classifier

Confusion Matrix –

<pre>[[21416 1301]
 [2891 3484]]</pre>
Accuracy Score- 0.855905403547367

& Classification Report –
<pre>   precision   recall f1-score   support

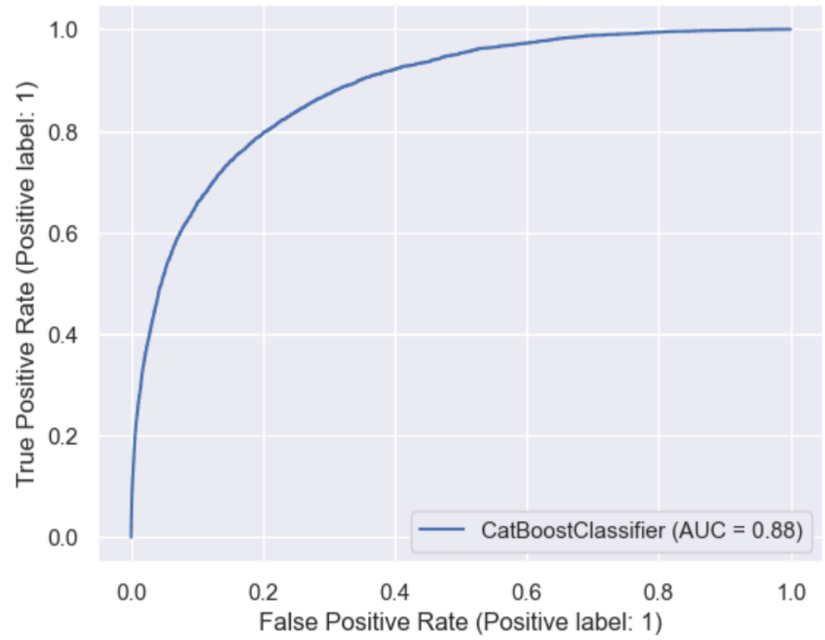
           0   0.88   0.94   0.91   22717
           1   0.73   0.55   0.62   6375

    accuracy   0.86   29092
   macro avg   0.80   0.74   0.77   29092
weighted avg   0.85   0.86   0.85   29092</pre>
</td>
<td>
<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))</pre>
<pre>[[21416 1301]
 [2891 3484]]

0.855905403547367

              precision   recall f1-score   support

           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</pre>
<pre>metrics.plot_roc_curve(catBoostC, X_test1, y_test1)
metrics.roc_auc_score(y_test1, y_pred1, average = None)</pre>
<pre>0.744619958966551</pre>


</td>
</tr>
<tr>
<td>6.</td>
<td>Hyperparameters Tuning</td>
<td>iterations = {1000, 2000, 3000,}
eval_metric = 'AUC</td>
<td>
<pre>catBoostC = CatBoostClassifier(iterations = 3000, eval_metric = 'AUC')</pre>
</td>
</tr>
</table>