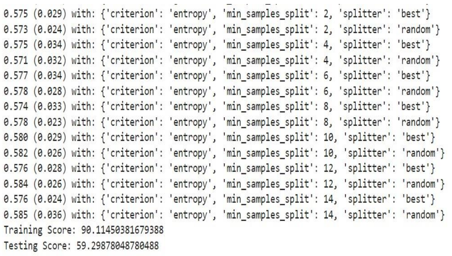
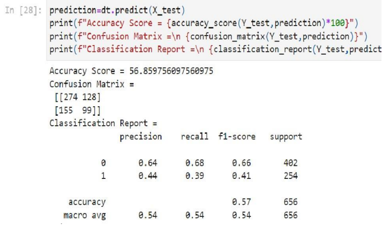
Model Performance Test

Date	21 / 11 / 2022
Team ID	PNT2022TMID06211
Project Name	Efficient Water Quality Analysis and prediction using Machine learning
Maximum Marks	10 Marks

Model Performance Testing:

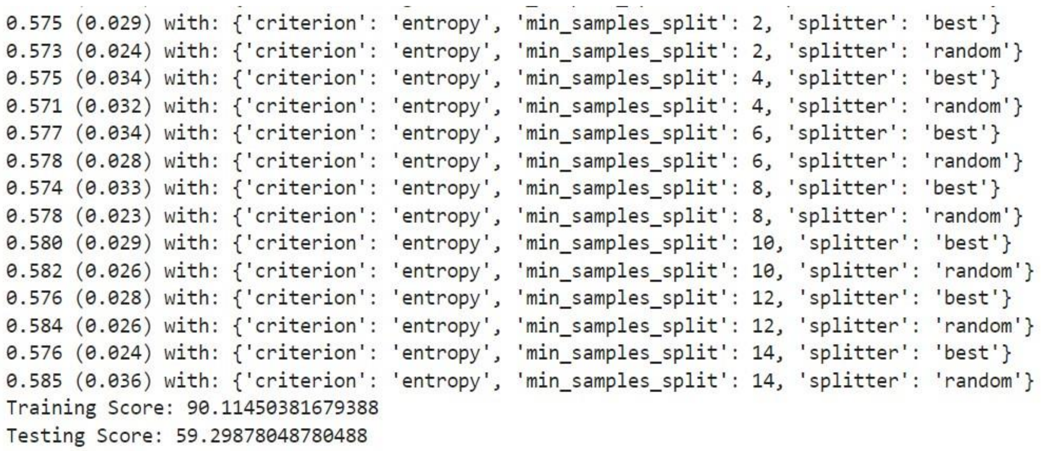
Project team shallfill the following information in model performance testing template.

S.No.	Parameter	Values	Screenshot
1.	Model Summary	Total params: 3279 rows,16 columns Trainable params: 3279 rows,16 columns Non-trainable params: 0	
2.	Accuracy	TrainingAccuracy – 12.68 ValidationAccuracy – 13.07	

```
0.575 (0.029) with: {'criterion': 'entropy', 'min_samples_split': 2, 'splitter': 'best'}
0.573 (0.024) with: {'criterion': 'entropy', 'min_samples_split': 2, 'splitter': 'random'}
0.575 (0.034) with: {'criterion': 'entropy', 'min_samples_split': 4, 'splitter': 'best'}
0.571 (0.032) with: {'criterion': 'entropy', 'min_samples_split': 4, 'splitter': 'random'}
0.577 (0.034) with: {'criterion': 'entropy', 'min_samples_split': 6, 'splitter': 'best'}
0.578 (0.028) with: {'criterion': 'entropy', 'min_samples_split': 6, 'splitter': 'random'}
0.574 (0.033) with: {'criterion': 'entropy', 'min_samples_split': 8, 'splitter': 'best'}
0.578 (0.023) with: {'criterion': 'entropy', 'min_samples_split': 8, 'splitter': 'random'}
0.580 (0.029) with: {'criterion': 'entropy', 'min_samples_split': 10, 'splitter': 'best'}
0.582 (0.026) with: {'criterion': 'entropy', 'min_samples_split': 10, 'splitter': 'random'}
0.576 (0.028) with: {'criterion': 'entropy', 'min_samples_split': 12, 'splitter': 'best'}
0.584 (0.026) with: {'criterion': 'entropy', 'min_samples_split': 12, 'splitter': 'random'}
0.576 (0.024) with: {'criterion': 'entropy', 'min_samples_split': 14, 'splitter': 'best'}
0.585 (0.036) with: {'criterion': 'entropy', 'min_samples_split': 14, 'splitter': 'random'}
Training Score: 90.11450381679388
Testing Score: 59.29878048780488
```

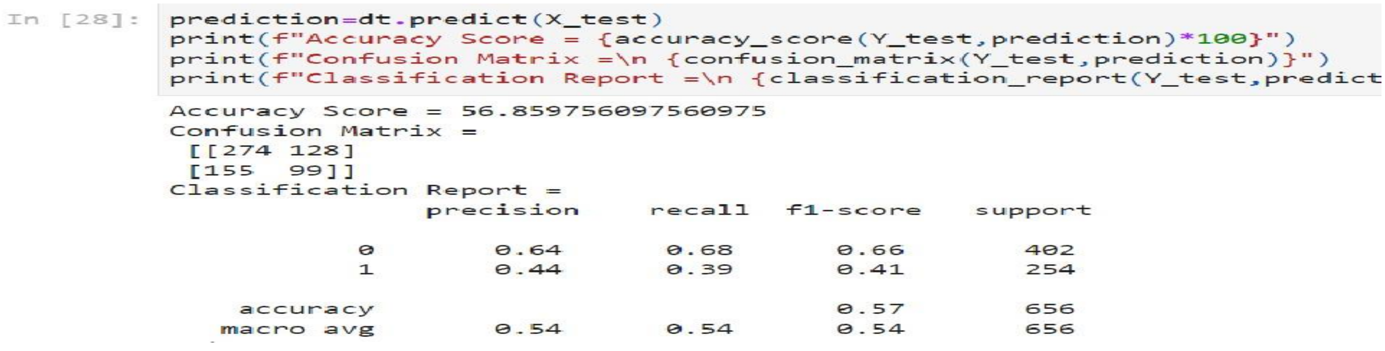
```
In [28]: prediction=dt.predict(X_test)
         print(f"Accuracy Score = {accuracy_score(Y_test,prediction)*100}")
         print(f"Confusion Matrix =\n {confusion_matrix(Y_test,prediction)}")
         print(f"Classification Report =\n {classification_report(Y_test,predict

         Accuracy Score = 56.859756097560975
         Confusion Matrix =
          [[274 128]
          [155  99]]
         Classification Report =
                       precision    recall  f1-score   support

                    0       0.64      0.68      0.66       402
                    1       0.44      0.39      0.41       254

             accuracy                           0.57       656
            macro avg       0.54      0.54      0.54       656
```

ModelSummary



```
0.575 (0.029) with: {'criterion': 'entropy', 'min_samples_split': 2, 'splitter': 'best'}
0.573 (0.024) with: {'criterion': 'entropy', 'min_samples_split': 2, 'splitter': 'random'}
0.575 (0.034) with: {'criterion': 'entropy', 'min_samples_split': 4, 'splitter': 'best'}
0.571 (0.032) with: {'criterion': 'entropy', 'min_samples_split': 4, 'splitter': 'random'}
0.577 (0.034) with: {'criterion': 'entropy', 'min_samples_split': 6, 'splitter': 'best'}
0.578 (0.028) with: {'criterion': 'entropy', 'min_samples_split': 6, 'splitter': 'random'}
0.574 (0.033) with: {'criterion': 'entropy', 'min_samples_split': 8, 'splitter': 'best'}
0.578 (0.023) with: {'criterion': 'entropy', 'min_samples_split': 8, 'splitter': 'random'}
0.580 (0.029) with: {'criterion': 'entropy', 'min_samples_split': 10, 'splitter': 'best'}
0.582 (0.026) with: {'criterion': 'entropy', 'min_samples_split': 10, 'splitter': 'random'}
0.576 (0.028) with: {'criterion': 'entropy', 'min_samples_split': 12, 'splitter': 'best'}
0.584 (0.026) with: {'criterion': 'entropy', 'min_samples_split': 12, 'splitter': 'random'}
0.576 (0.024) with: {'criterion': 'entropy', 'min_samples_split': 14, 'splitter': 'best'}
0.585 (0.036) with: {'criterion': 'entropy', 'min_samples_split': 14, 'splitter': 'random'}
Training Score: 90.11450381679388
Testing Score: 59.29878048780488
```

Accuracy



```
In [28]: prediction=dt.predict(X_test)
         print(f"Accuracy Score = {accuracy_score(Y_test,prediction)*100}")
         print(f"Confusion Matrix =\n {confusion_matrix(Y_test,prediction)}")
         print(f"Classification Report =\n {classification_report(Y_test,predict
```

```
Accuracy Score = 56.859756097560975
Confusion Matrix =
 [[274 128]
 [155  99]]
Classification Report =
               precision    recall  f1-score   support

           0       0.64      0.68      0.66       402
           1       0.44      0.39      0.41       254

    accuracy                           0.57       656
   macro avg       0.54      0.54      0.54       656
```