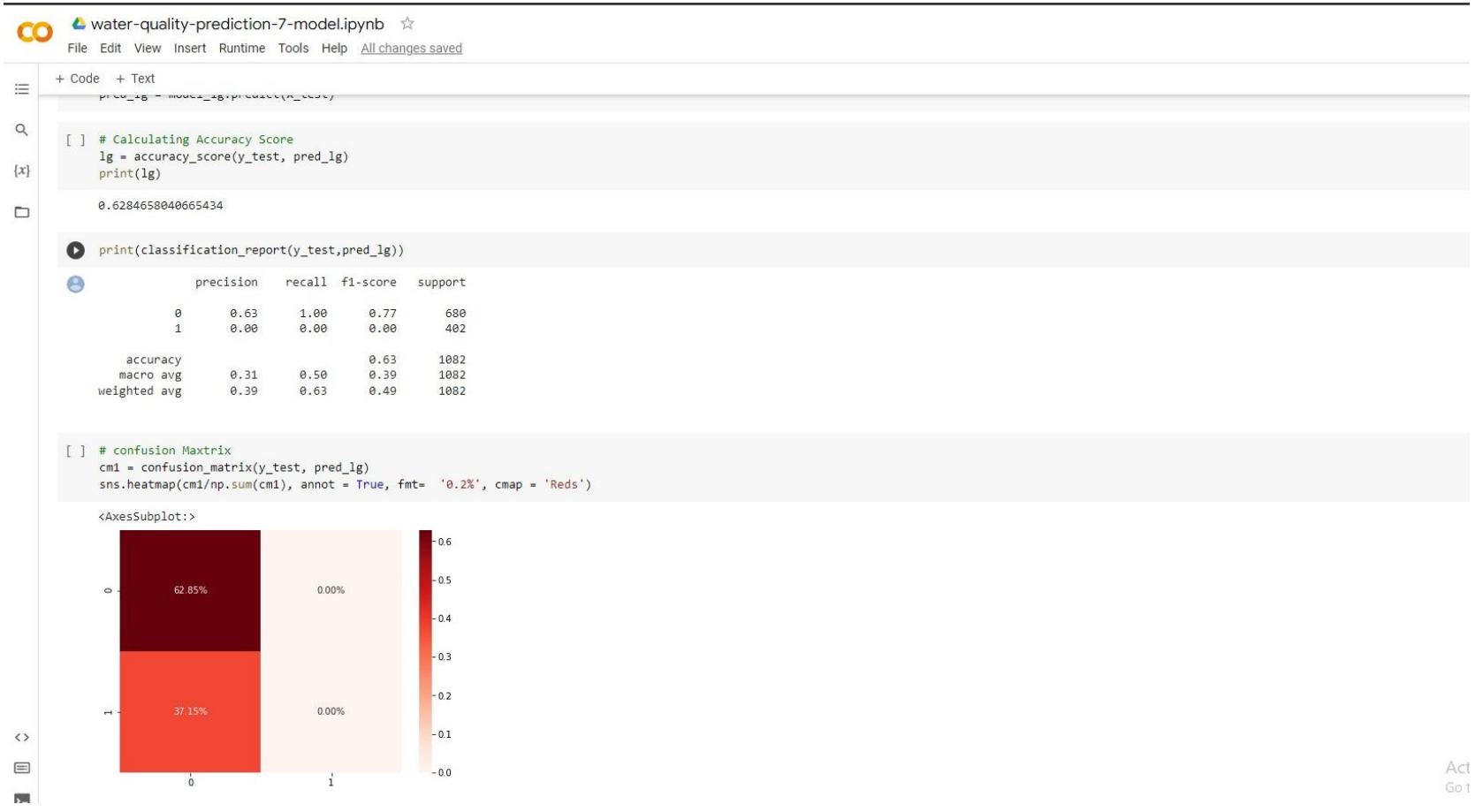
SPRINT 2
PROJECT DELIVERABLES (Model building & Evaluation

Team ID	PNT2022TMID28413
Project Name	Efficient Water Quality Analysis & Prediction using Machine Learning

LOGISTIC REGRESSION EVALUATION:



DECISION TREE EVALUATION:

water-quality-prediction-7-model.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
dt = accuracy_score(y_test, pred_dt)
print(dt)
```

```
0.6451016635859519
```

```
print(classification_report(y_test,pred_dt))
```

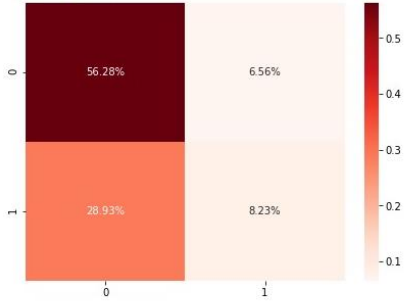
```
              precision    recall  f1-score   support

           0       0.66      0.90      0.76       680
           1       0.56      0.22      0.32       402

    accuracy                           0.65      1082
   macro avg       0.61      0.56      0.54      1082
weighted avg       0.62      0.65      0.60      1082
```

```
# confusion Maxtrix
cm2 = confusion_matrix(y_test, pred_dt)
sns.heatmap(cm2/np.sum(cm2), annot = True, fmt=  '0.2%', cmap = 'Reds')
```

```
<AxesSubplot:>
```



Using Random Forest

RANDOM FOREST EVALUATION:

water-quality-prediction-7-model.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
[ ] # Calculating Accuracy Score
    rf = accuracy_score(y_test, pred_rf)
    print(rf)
```

```
0.6284658040665434
```

```
[ ] print(classification_report(y_test,pred_rf))
```

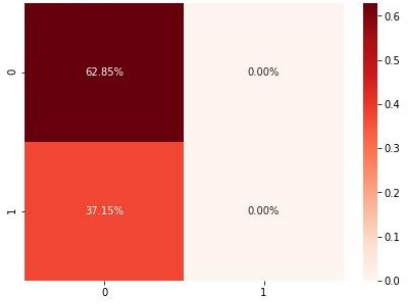
```
              precision    recall  f1-score   support

           0       0.63      1.00      0.77       680
           1       0.00      0.00      0.00       402

    accuracy                           0.63      1082
   macro avg       0.31      0.50      0.39      1082
weighted avg       0.39      0.63      0.49      1082
```

```
# confusion Maxtrix
cm3 = confusion_matrix(y_test, pred_rf)
sns.heatmap(cm3/np.sum(cm3), annot = True, fmt=  '0.2%', cmap = 'Reds')
```

```
<AxesSubplot:>
```



+ Code + Text

XGBOOST CLASSIFIER EVALUATION:

water-quality-prediction-7-model.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
pred_xgb = model_xgb.predict(X_test)
```

```
[ ] # Calculating Accuracy Score
    xgb = accuracy_score(y_test, pred_xgb)
    print(xgb)
```

```
0.6709796672828097
```

```
[ ] print(classification_report(y_test,pred_xgb))
```

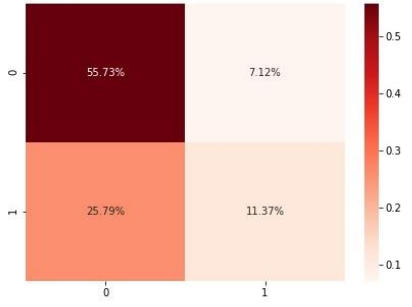
```
              precision    recall  f1-score   support

           0       0.68      0.89      0.77       680
           1       0.61      0.31      0.41       402

    accuracy                           0.67      1082
   macro avg       0.65      0.60      0.59      1082
weighted avg       0.66      0.67      0.64      1082
```

```
# confusion Maxtrix
cm4 = confusion_matrix(y_test, pred_xgb)
sns.heatmap(cm4/np.sum(cm4), annot = True, fmt=  '0.2%', cmap = 'Reds')
```

```
<AxesSubplot:>
```



KNN EVALUATION:

water-quality-prediction-7-model.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
[ ] # Calculating Accuracy Score
    kn = accuracy_score(y_test, pred_kn)
    print(kn)
```

```
0.6534195933456562
```

```
[ ] print(classification_report(y_test,pred_kn))
```

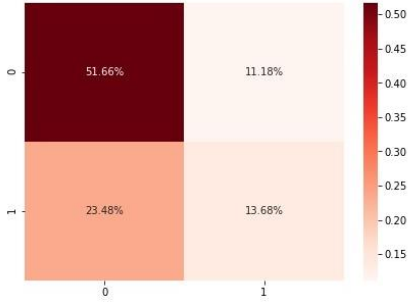
```
              precision    recall  f1-score   support

           0       0.69      0.82      0.75       680
           1       0.55      0.37      0.44       402

    accuracy                           0.65      1082
   macro avg       0.62      0.60      0.59      1082
weighted avg       0.64      0.65      0.63      1082
```

```
# confusion Maxtrix
cm5 = confusion_matrix(y_test, pred_kn)
sns.heatmap(cm5/np.sum(cm5), annot = True, fmt=  '0.2%', cmap = 'Reds')
```

```
<AxesSubplot:>
```



SVM EVALUATION:

water-quality-prediction-7-model.ipynb

File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
pred_svm = model_svm.predict(X_test)
```

```
# Calculating Accuracy Score
sv = accuracy_score(y_test, pred_svm)
print(sv)
```

```
0.6885397412199631
```

```
print(classification_report(y_test,pred_kn))
```

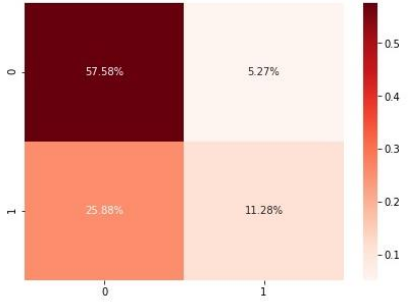
```
              precision    recall  f1-score   support

           0       0.69      0.82      0.75       680
           1       0.55      0.37      0.44       402

    accuracy                           0.65      1082
   macro avg       0.62      0.60      0.59      1082
weighted avg       0.64      0.65      0.63      1082
```

```
# confusion Maxtrix
cm6 = confusion_matrix(y_test, pred_svm)
sns.heatmap(cm6/np.sum(cm6), annot = True, fmt=  '0.2%', cmap = 'Reds')
```

```
<AxesSubplot:>
```



COMPARISSON:

water-quality-prediction-7-model.ipynb

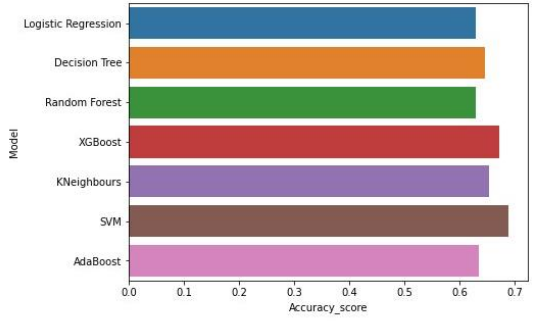
File Edit View Insert Runtime Tools Help All changes saved

+ Code + Text

```
models = pd.DataFrame({
    'Model':['Logistic Regression', 'Decision Tree', 'Random Forest', 'XGBoost', 'KNeighbours', 'SVM', 'AdaBoost'],
    'Accuracy_score' :[lg, dt, rf, xgb, kn, sv, ada]
})
models
sns.barplot(x='Accuracy_score', y='Model', data=models)

models.sort_values(by='Accuracy_score', ascending=False)
```

	Model	Accuracy_score
5	SVM	0.688540
3	XGBoost	0.670980
4	KNeighbours	0.653420
1	Decision Tree	0.645102
6	AdaBoost	0.634011
0	Logistic Regression	0.628466
2	Random Forest	0.628466



+ Code + Text

Conclusion :- Here SVM classifier has achieved highest accuracy.

CONCLUSION:

Out of all we found that *SVM Classifier* has the highest accuracy among all, so we are going to use *SVM Classifier* to train our Water Prediction machine learning model.