MAN206

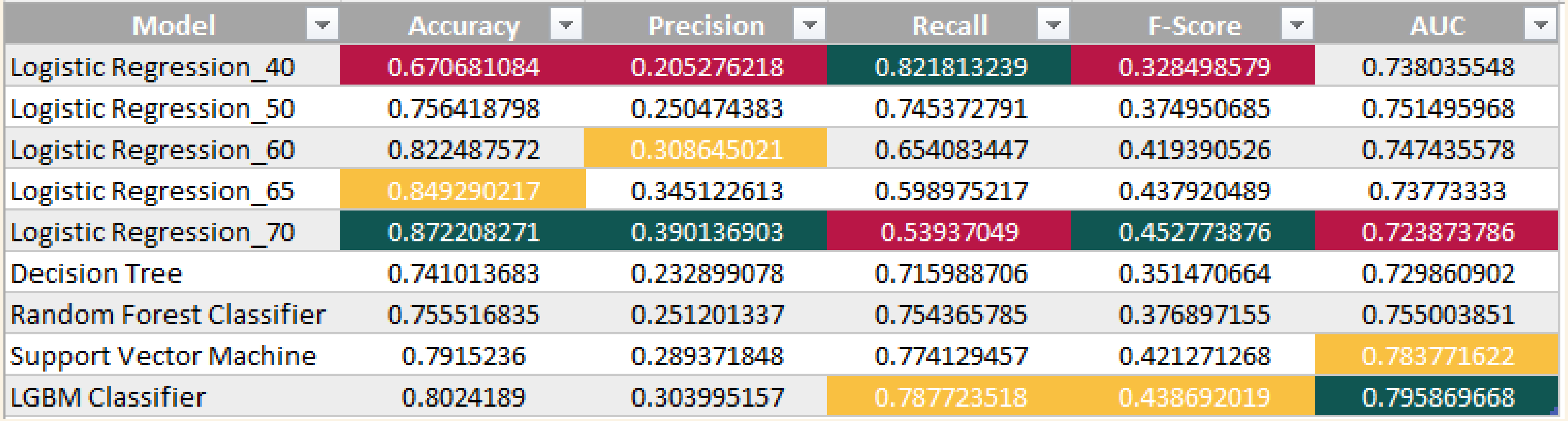
classification

RAIN PREDICTION

LEORANELE GRACE ESTRADA

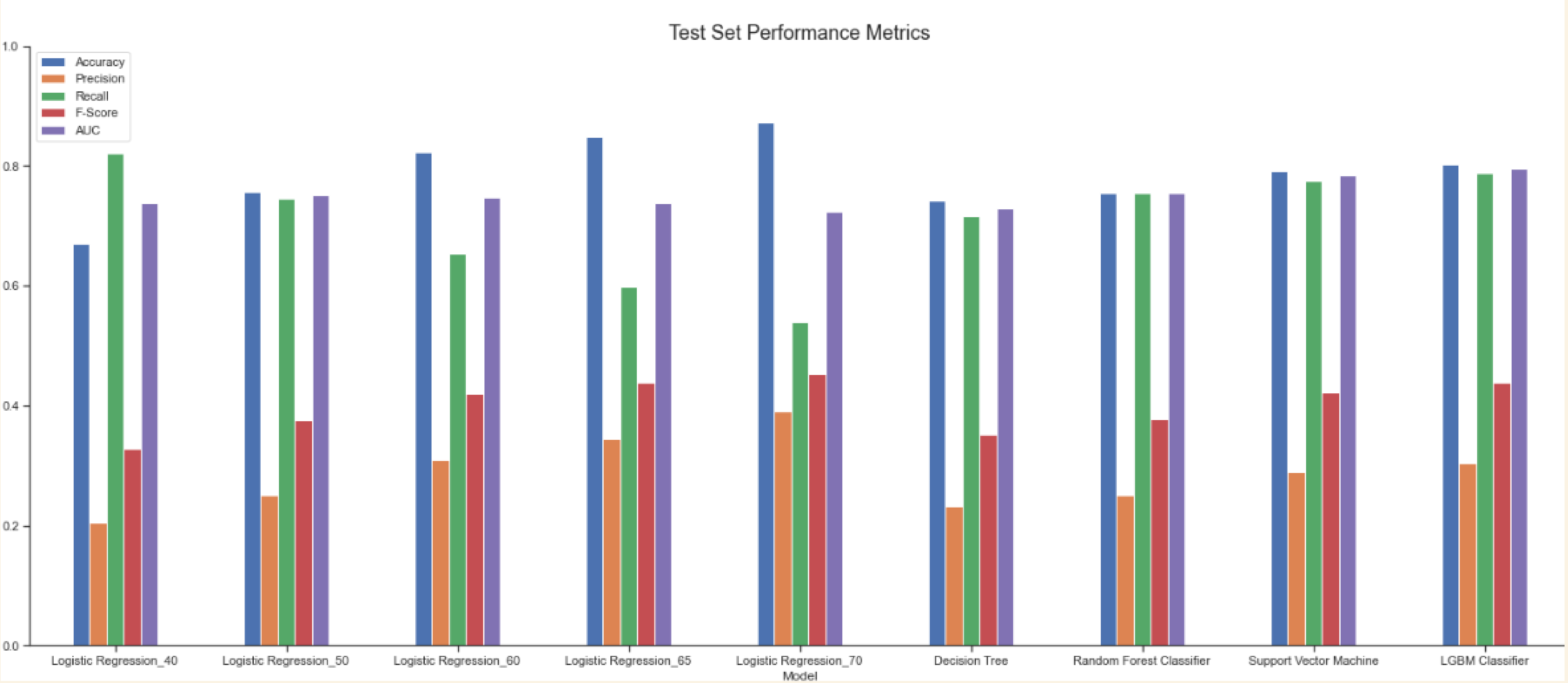
Predictive Analytics and Machine Learning

PERFORMANCE METRICS

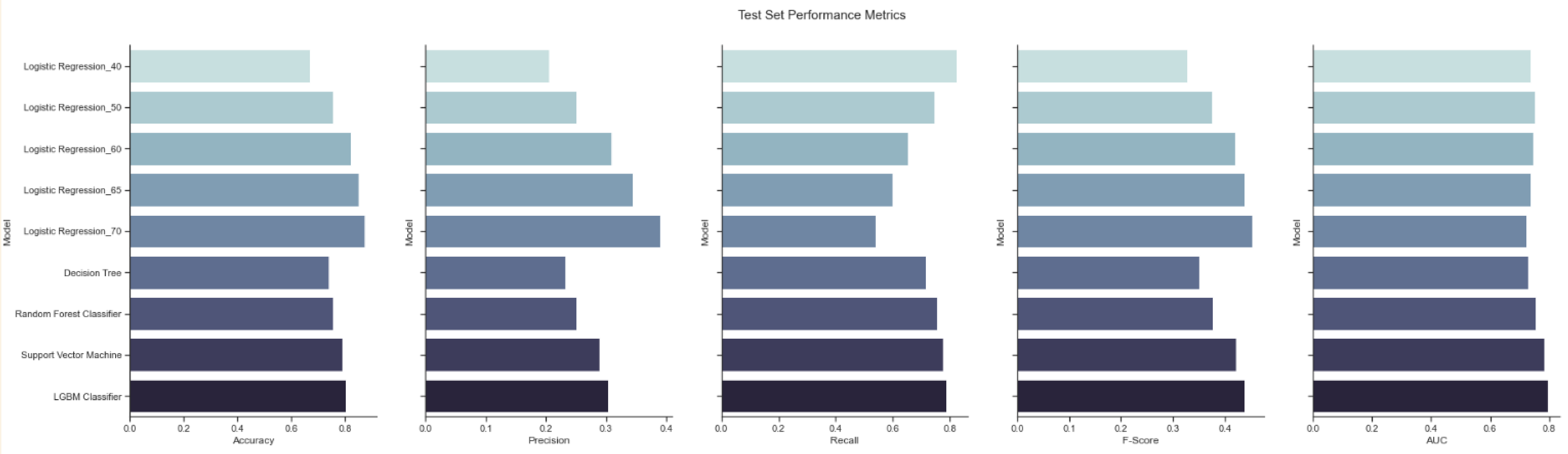


Model	Accuracy	Precision	Recall	F-Score	AUC
Logistic Regression_40	0.670681084	0.205276218	0.821813239	0.328498579	0.738035548
Logistic Regression_50	0.756418798	0.250474383	0.745372791	0.374950685	0.751495968
Logistic Regression_60	0.822487572	0.308645021	0.654083447	0.419390526	0.747435578
Logistic Regression_65	0.849290217	0.345122613	0.598975217	0.437920489	0.73773333
Logistic Regression_70	0.872208271	0.390136903	0.53937049	0.452773876	0.723873786
Decision Tree	0.741013683	0.232899078	0.715988706	0.351470664	0.729860902
Random Forest Classifier	0.755516835	0.251201337	0.754365785	0.376897155	0.755003851
Support Vector Machine	0.7915236	0.289371848	0.774129457	0.421271268	0.783771622
LGBM Classifier	0.8024189	0.303995157	0.787723518	0.438692019	0.795869668

PERFORMANCE METRICS

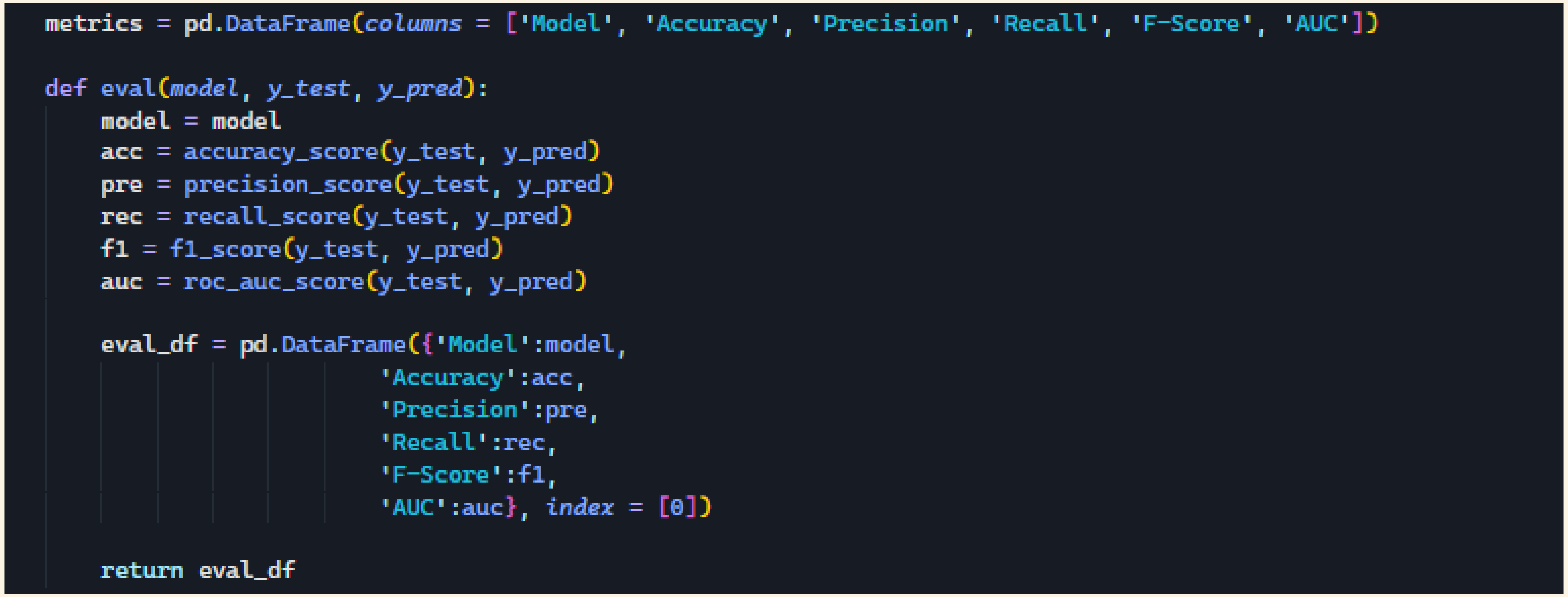


PERFORMANCE METRICS



custom code for

PERFORMANCE EVALUATION



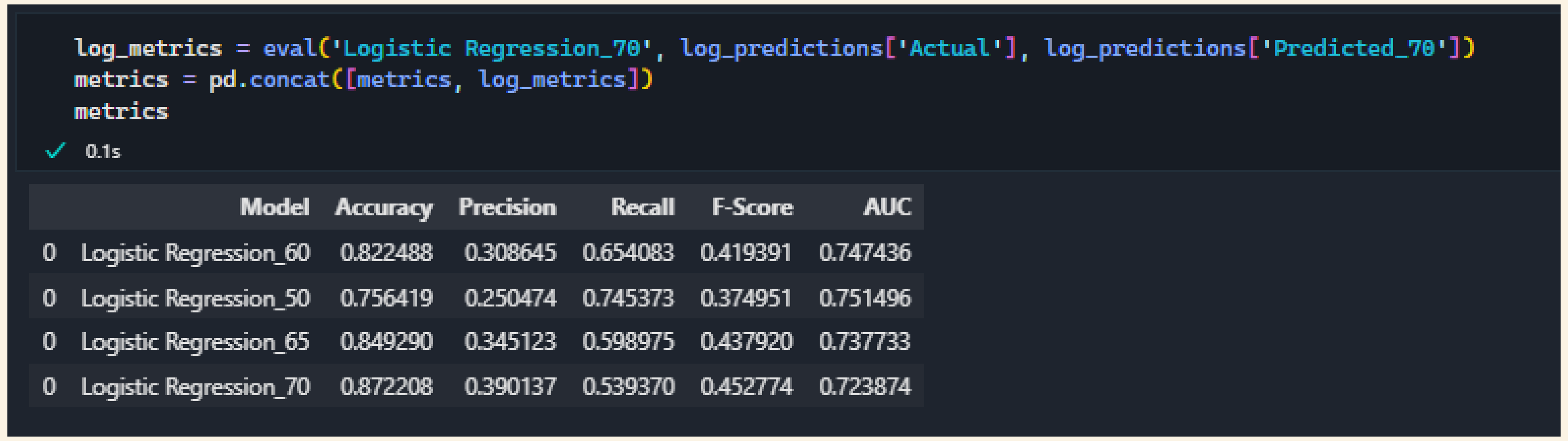
```python
metrics = pd.DataFrame(columns = ['Model', 'Accuracy', 'Precision', 'Recall', 'F-Score', 'AUC'])

def eval(model, y_test, y_pred):
    model = model
    acc = accuracy_score(y_test, y_pred)
    pre = precision_score(y_test, y_pred)
    rec = recall_score(y_test, y_pred)
    f1 = f1_score(y_test, y_pred)
    auc = roc_auc_score(y_test, y_pred)

    eval_df = pd.DataFrame({'Model':model,
                        'Accuracy':acc,
                        'Precision':pre,
                        'Recall':rec,
                        'F-Score':f1,
                        'AUC':auc}, index = [0])

    return eval_df
```

I created a custom code to evaluate the performance of the model based on Accuracy, Precision, Recall, F1 Score and AUC Score.

This function also returns it as a DataFrame to be concatenated with the metrics of the other models.

output of
PERFORMANCE EVALUATION

LOGISTIC REGRESSION

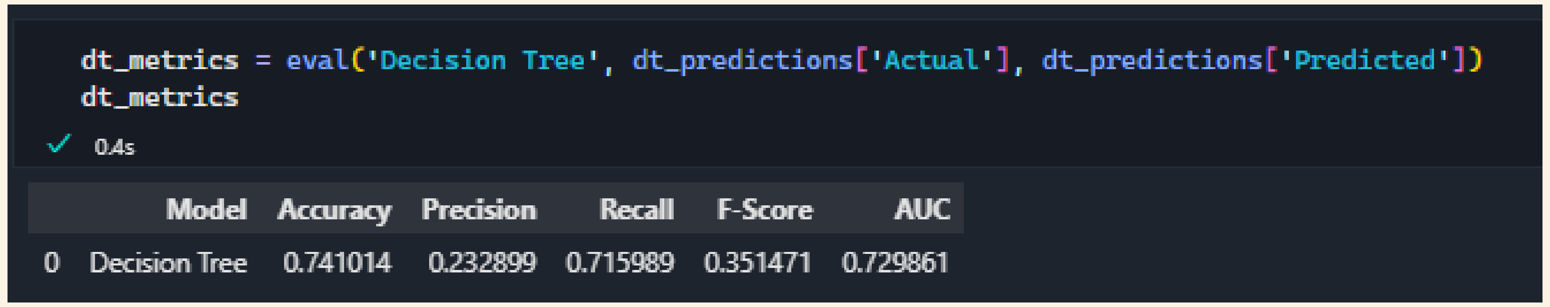


```
log_metrics = eval('Logistic Regression_70', log_predictions['Actual'], log_predictions['Predicted_70'])
metrics = pd.concat([metrics, log_metrics])
metrics
```

✓ 0.1s

	Model	Accuracy	Precision	Recall	F-Score	AUC
0	Logistic Regression_60	0.822488	0.308645	0.654083	0.419391	0.747436
0	Logistic Regression_50	0.756419	0.250474	0.745373	0.374951	0.751496
0	Logistic Regression_65	0.849290	0.345123	0.598975	0.437920	0.737733
0	Logistic Regression_70	0.872208	0.390137	0.539370	0.452774	0.723874

DECISION TREE



```
dt_metrics = eval('Decision Tree', dt_predictions['Actual'], dt_predictions['Predicted'])
dt_metrics
```

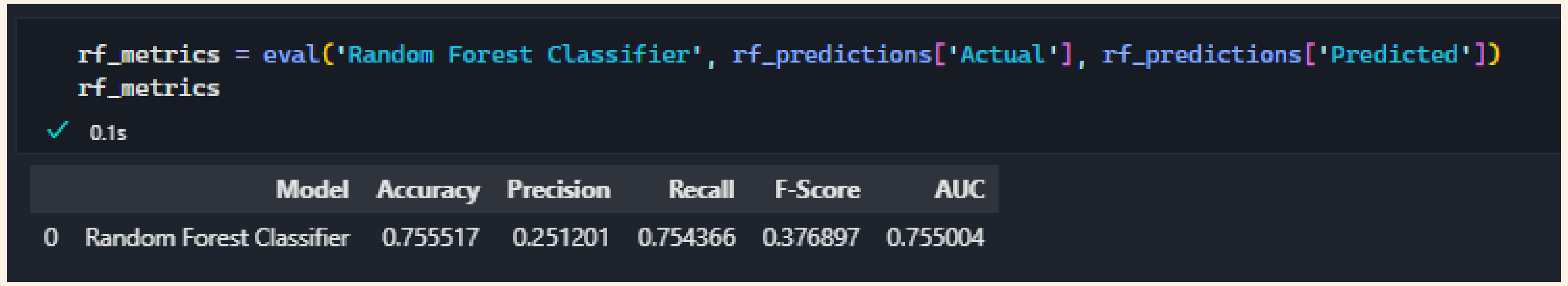
✓ 0.4s

	Model	Accuracy	Precision	Recall	F-Score	AUC
0	Decision Tree	0.741014	0.232899	0.715989	0.351471	0.729861

output of

PERFORMANCE EVALUATION

RANDOM FOREST

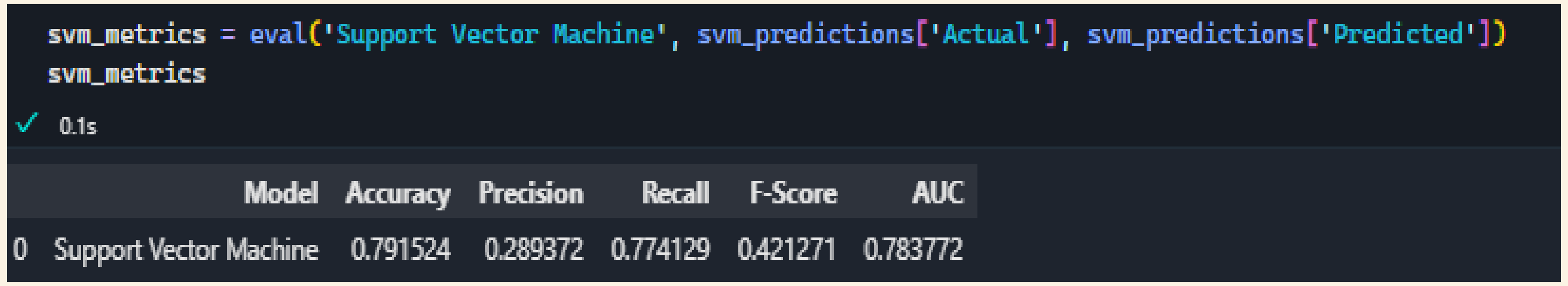


```
rf_metrics = eval('Random Forest Classifier', rf_predictions['Actual'], rf_predictions['Predicted'])
rf_metrics
```

✓ 0.1s

```
                      Model  Accuracy  Precision    Recall   F-Score       AUC
0  Random Forest Classifier  0.755517   0.251201  0.754366  0.376897  0.755004
```

SUPPORT VECTOR MACHINE



```
svm_metrics = eval('Support Vector Machine', svm_predictions['Actual'], svm_predictions['Predicted'])
svm_metrics
```

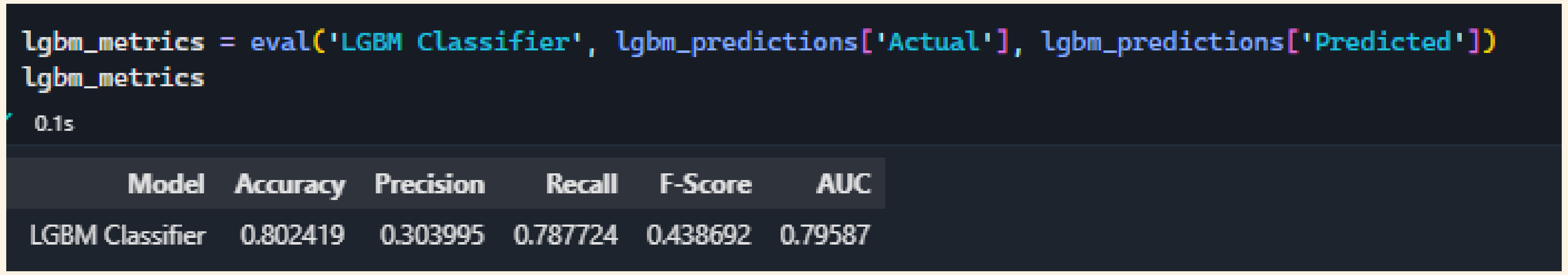
✓ 0.1s

```
                    Model  Accuracy  Precision    Recall   F-Score       AUC
0  Support Vector Machine  0.791524   0.289372  0.774129  0.421271  0.783772
```

output of

PERFORMANCE EVALUATION

LGBM CLASSIFICATION



```
lgbm_metrics = eval('LGBM Classifier', lgbm_predictions['Actual'], lgbm_predictions['Predicted'])
lgbm_metrics
```

```
✓ 0.1s
```

```
           Model  Accuracy  Precision    Recall   F-Score      AUC
 LGBM Classifier  0.802419   0.303995  0.787724  0.438692  0.79587
```

rain prediction

BEST MODEL

CONTEXT

For this project I assumed that the objective for predicting if it will rain tomorrow is to be able to make decisions whether the next day would be a good day to perform outdoor activities.

For this objective, I want to minimize the number of false negatives because it might be more dangerous/risky if one went out to do some outdoor activities because the prediction is that it will not rain but in actual it rained.

With that, I focused on the model with the balance of high Accuracy and Recall.

rain prediction

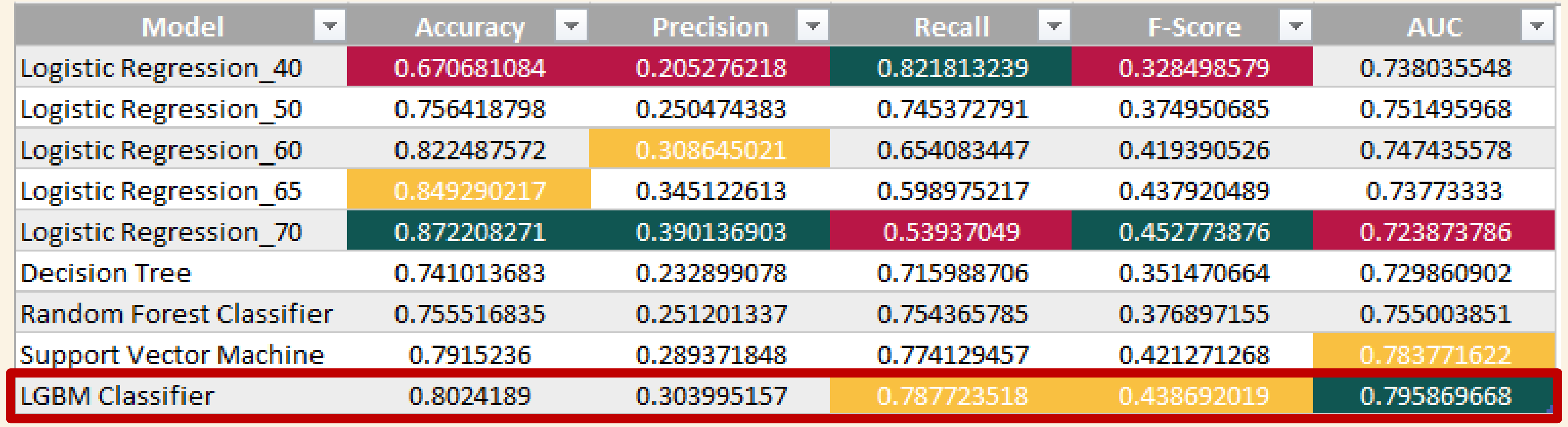
BEST MODEL

EXPLANATION / REASONING

✗ **Logistic Regression with a probability threshold of .70** gained the highest accuracy value of 85% but it also has the lowest Recall value, this is not a good model for the objective.

✗ **Logistic Regression with a probability threshold of .40** has the highest recall, yet it also has the lowest Accuracy, this is also not the model for the objective.

✓ The **LGBM Classifier** on the other hand, has the highest AUC, second to the highest F-Score and Recall, and Accuracy of 80%. It clearly has the balance of Accuracy and Recall. Therefore, **this is the best model** for the set objective.

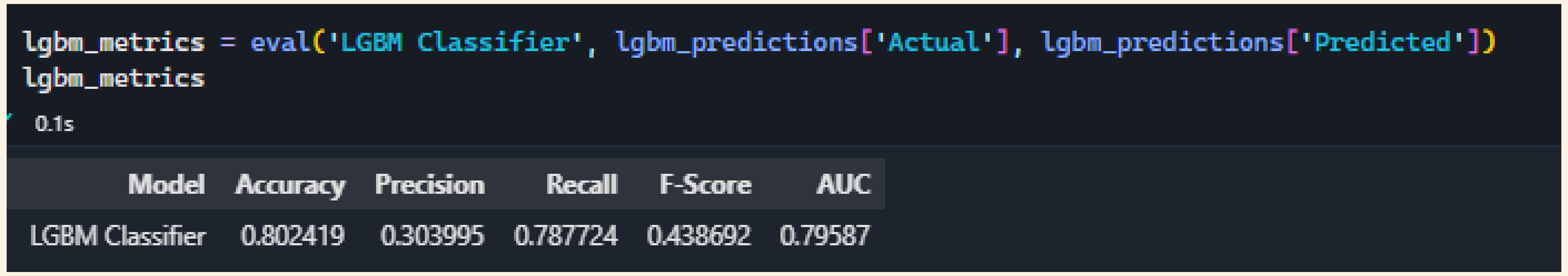


Model	Accuracy	Precision	Recall	F-Score	AUC
Logistic Regression_40	0.670681084	0.205276218	0.821813239	0.328498579	0.738035548
Logistic Regression_50	0.756418798	0.250474383	0.745372791	0.374950685	0.751495968
Logistic Regression_60	0.822487572	0.308645021	0.654083447	0.419390526	0.747435578
Logistic Regression_65	0.849290217	0.345122613	0.598975217	0.437920489	0.73773333
Logistic Regression_70	0.872208271	0.390136903	0.53937049	0.452773876	0.723873786
Decision Tree	0.741013683	0.232899078	0.715988706	0.351470664	0.729860902
Random Forest Classifier	0.755516835	0.251201337	0.754365785	0.376897155	0.755003851
Support Vector Machine	0.7915236	0.289371848	0.774129457	0.421271268	0.783771622
LGBM Classifier	0.8024189	0.303995157	0.787723518	0.438692019	0.795869668

rain prediction

BEST MODEL

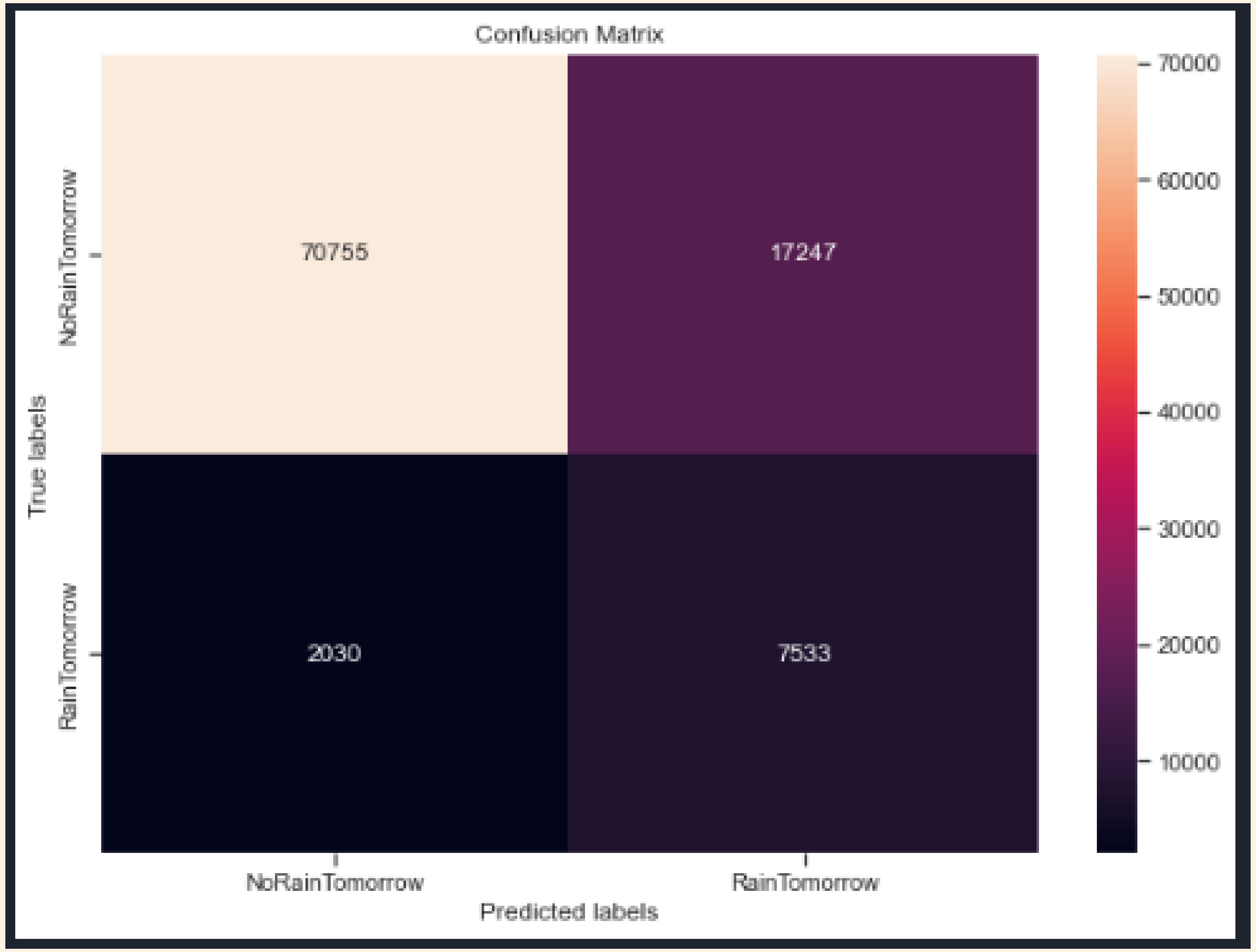
LGBM CLASSIFICATION MODEL



```
lgbm_metrics = eval('LGBM Classifier', lgbm_predictions['Actual'], lgbm_predictions['Predicted'])
lgbm_metrics
```

0.1s

Model	Accuracy	Precision	Recall	F-Score	AUC
LGBM Classifier	0.802419	0.303995	0.787724	0.438692	0.79587


Confusion Matrix
True labels
NoRainTomorrow
RainTomorrow
NoRainTomorrow
RainTomorrow
Predicted labels