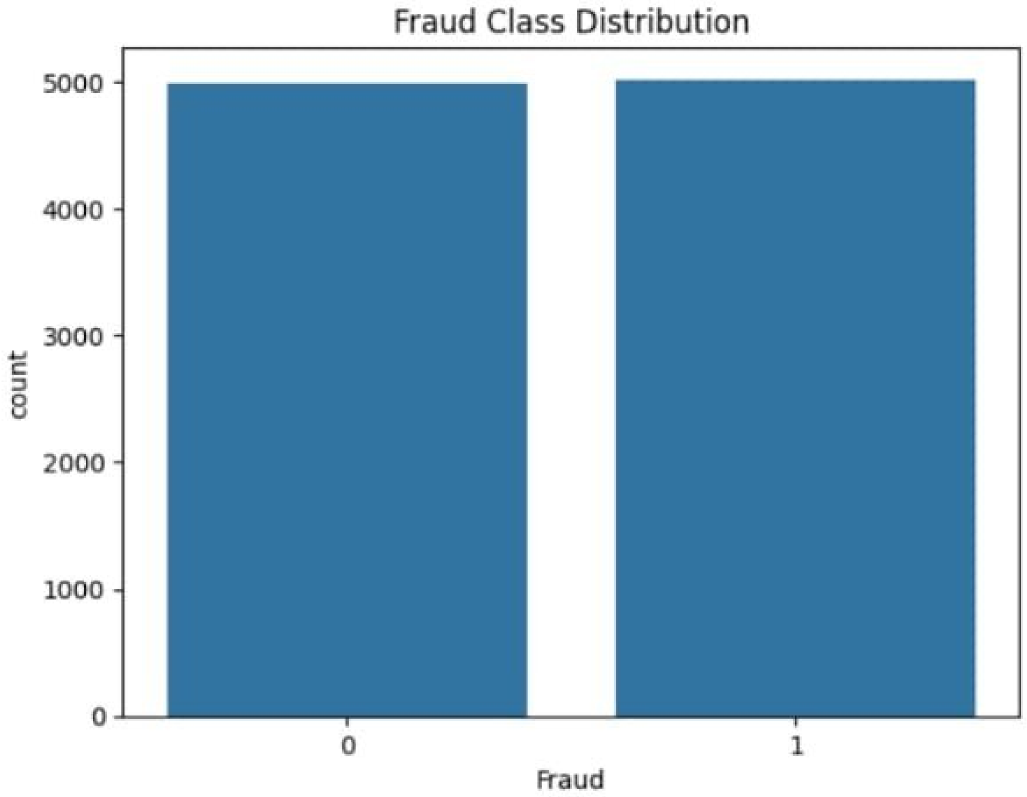
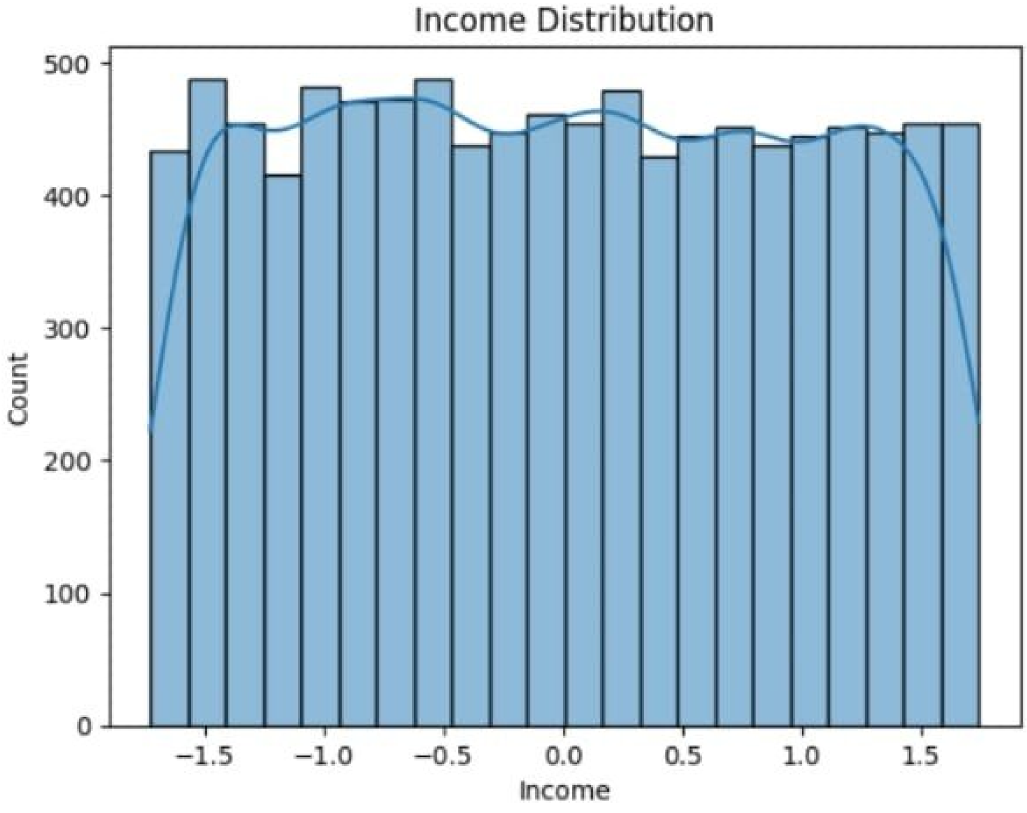
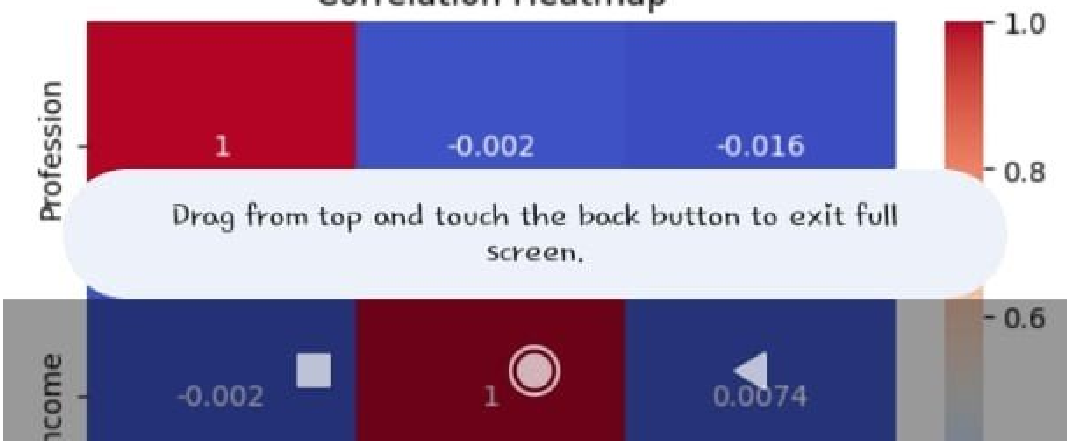
Fraud Class Distribution



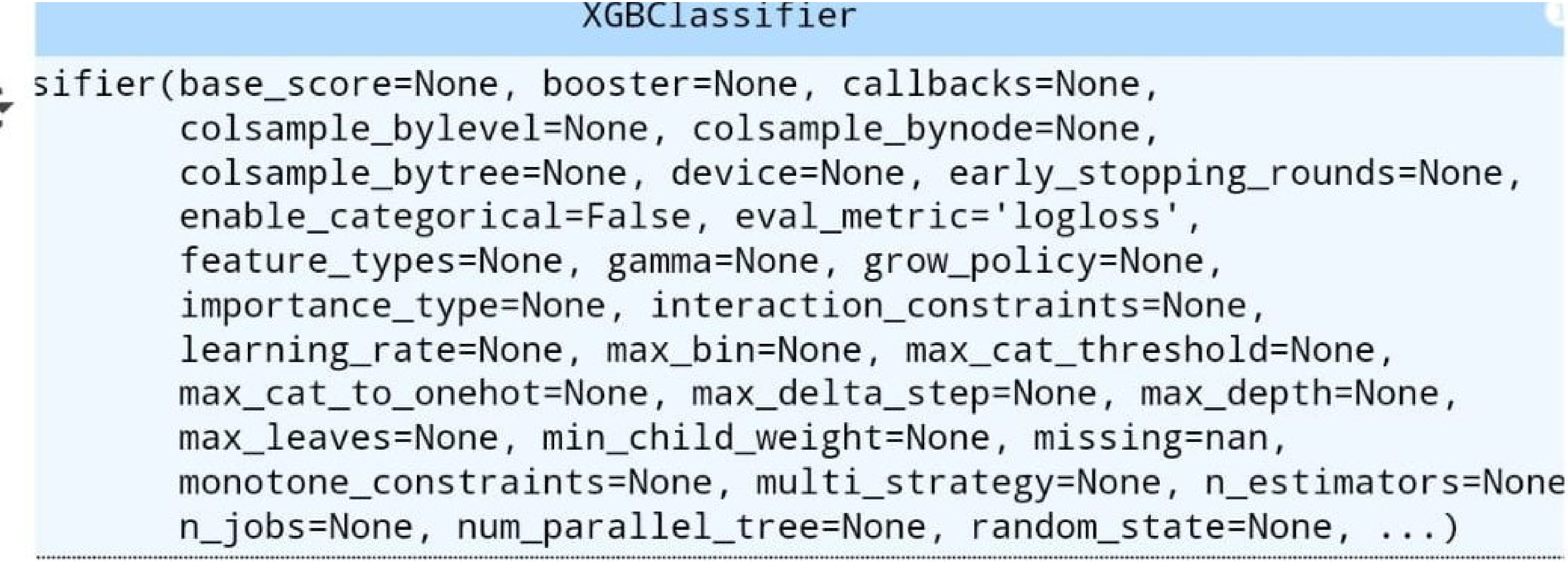
Income Distribution



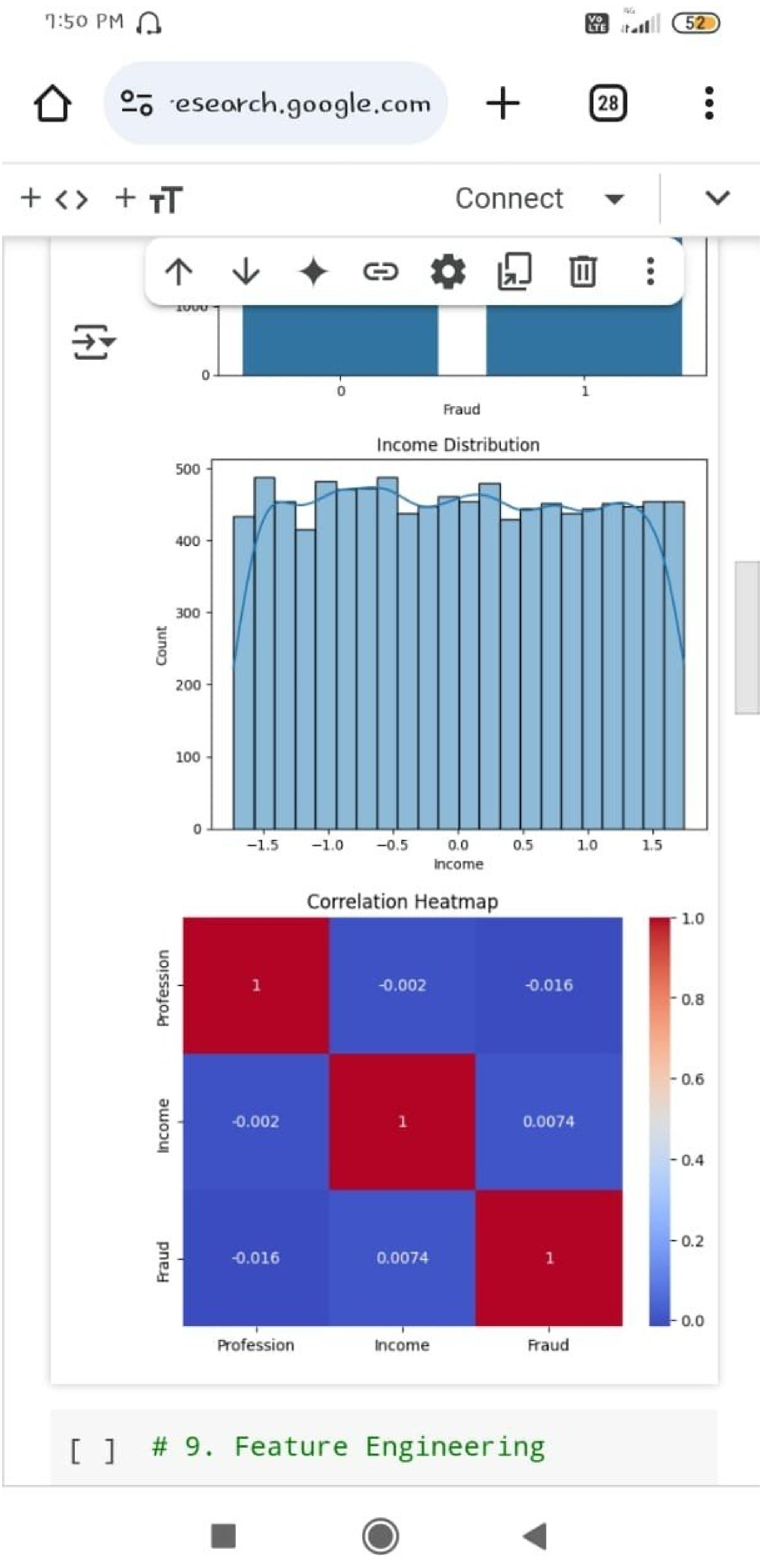
Correlation Heatmap



Drag from top and touch the back button to exit full screen.

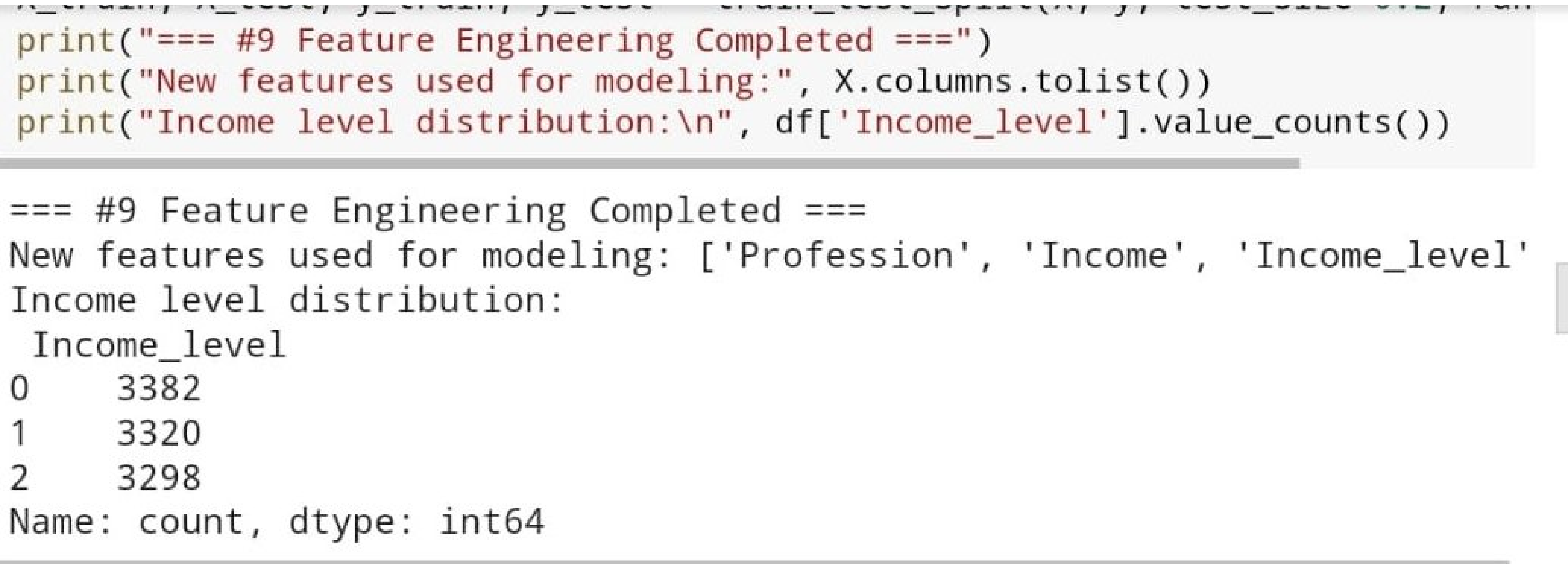


```
sifier(base_score=None, booster=None, callbacks=None,
        colsample_bylevel=None, colsample_bynode=None,
        colsample_bytree=None, device=None, early_stopping_rounds=None,
        enable_categorical=False, eval_metric='logloss',
        feature_types=None, gamma=None, grow_policy=None,
        importance_type=None, interaction_constraints=None,
        learning_rate=None, max_bin=None, max_cat_threshold=None,
        max_cat_to_onehot=None, max_delta_step=None, max_depth=None,
        max_leaves=None, min_child_weight=None, missing=nan,
        monotone_constraints=None, multi_strategy=None, n_estimators=None
        n_jobs=None, num_parallel_tree=None, random_state=None, ...)
```


7:50 PM
52
research.google.com
28
Connect
1000
0
0
1
Fraud
Income Distribution
500
400
300
200
100
0
Count
−1.5
−1.0
−0.5
0.0
0.5
1.0
1.5
Income
Correlation Heatmap
1.0
0.8
0.6
0.4
0.2
0.0

	Profession	Income	Fraud
Profession	1	-0.002	-0.016
Income	-0.002	1	0.0074
Fraud	-0.016	0.0074	1

```
[ ]  # 9. Feature Engineering
```

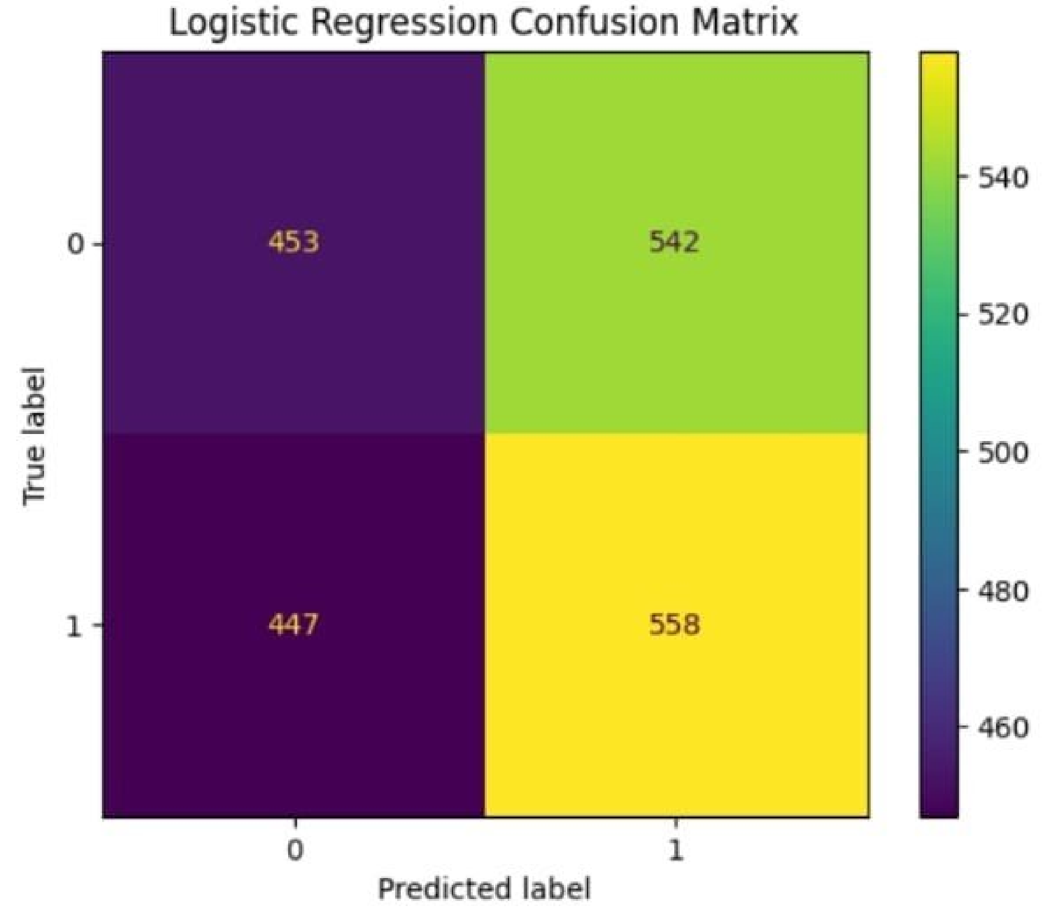


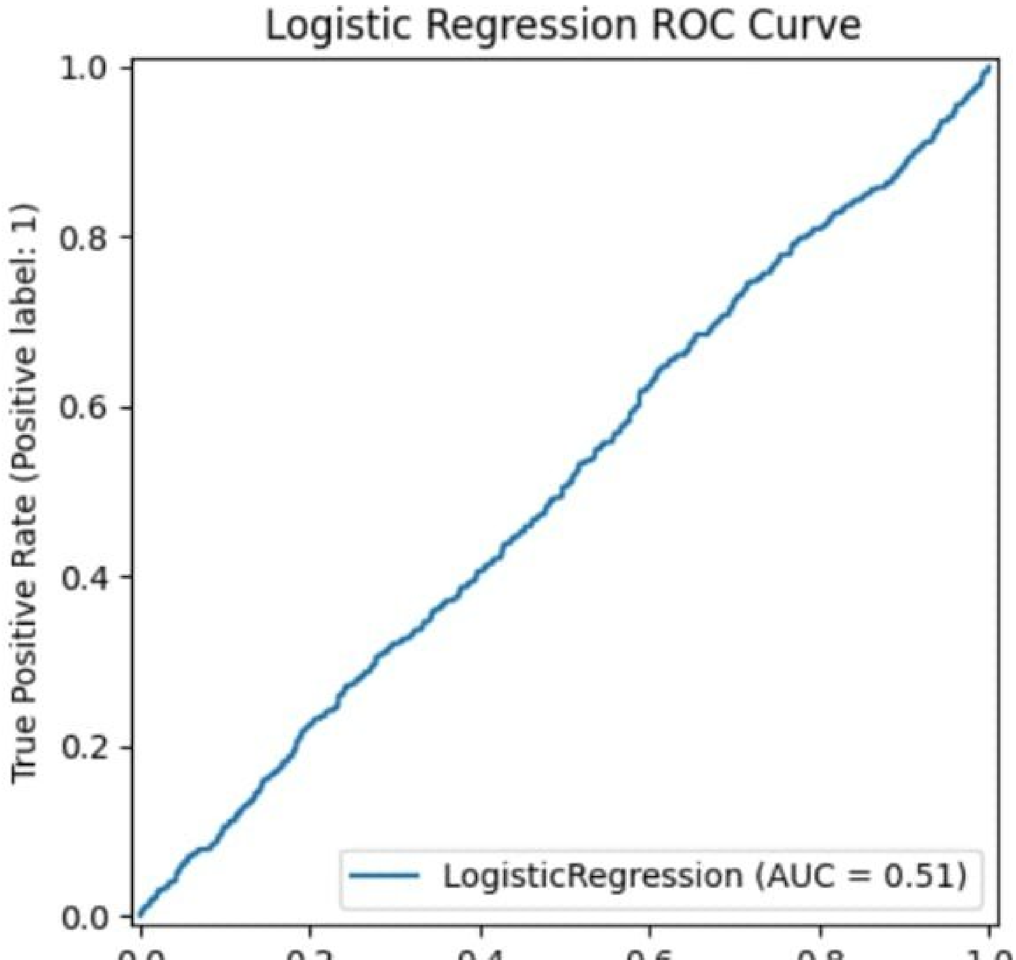
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, ran
print("=== #9 Feature Engineering Completed ===")
print("New features used for modeling:", X.columns.tolist())
print("Income level distribution:\n", df['Income_level'].value_counts())
```

```
=== #9 Feature Engineering Completed ===
New features used for modeling: ['Profession', 'Income', 'Income_level'
Income level distribution:
 Income_level
0    3382
1    3320
2    3298
Name: count, dtype: int64
```

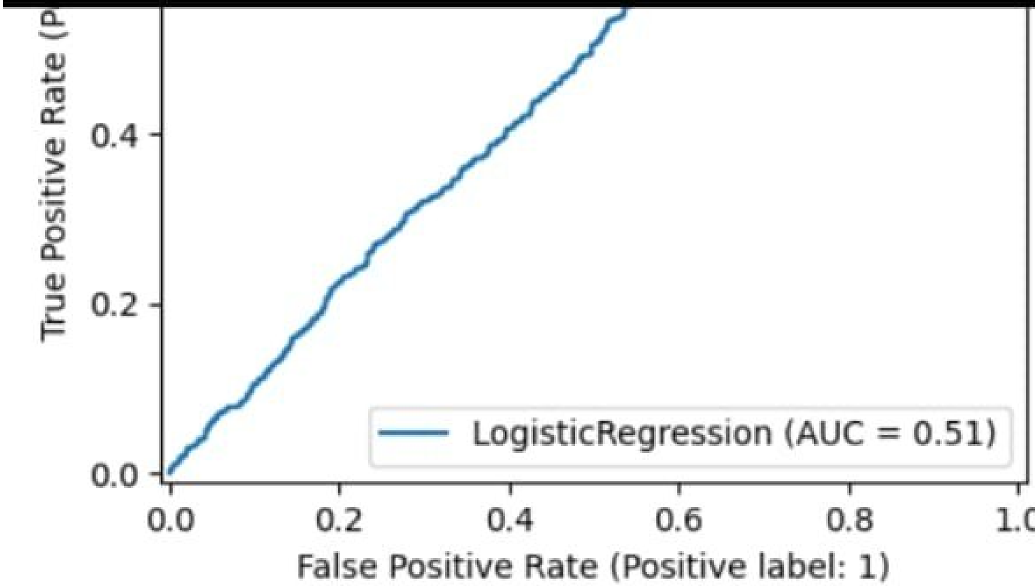
[redacted]

```
    accuracy                                    0.5
   macro avg       0.51        0.51        0.5
weighted avg       0.51        0.51        0.5

ROC AUC Score: 0.5052501312532813
```

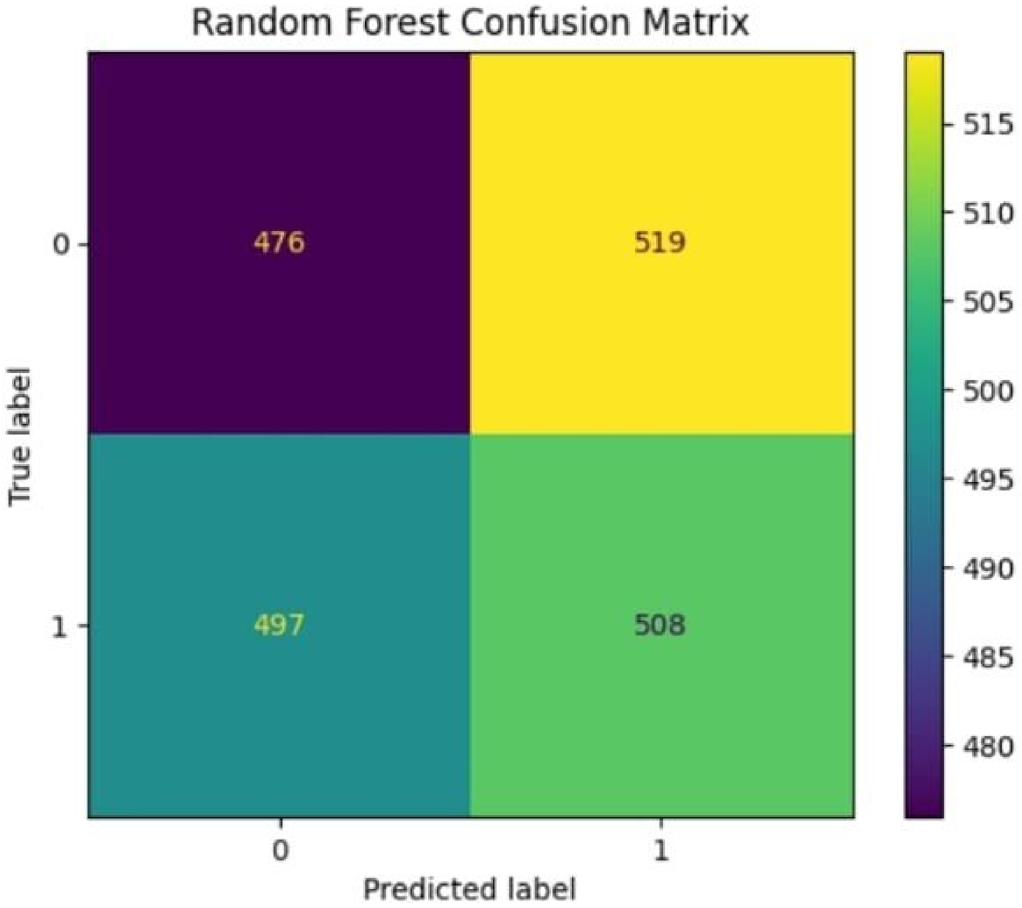




[redacted]

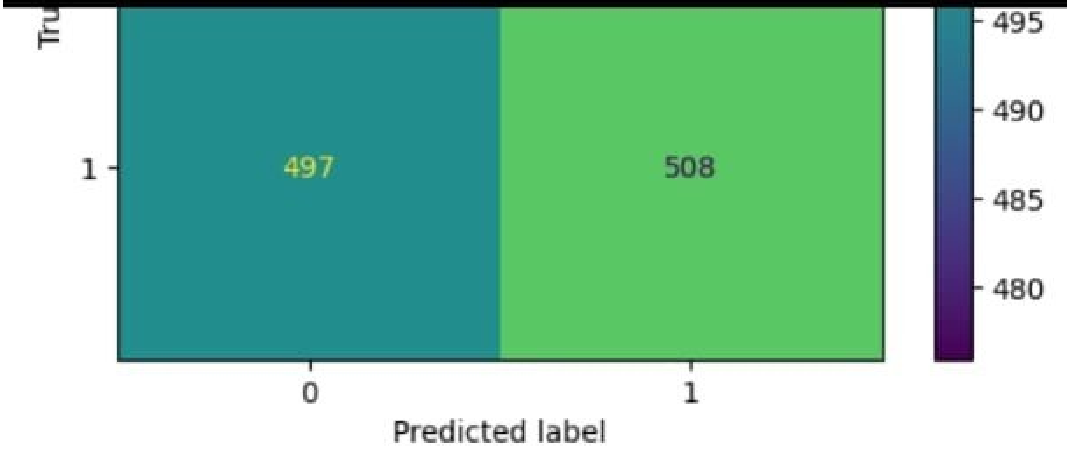


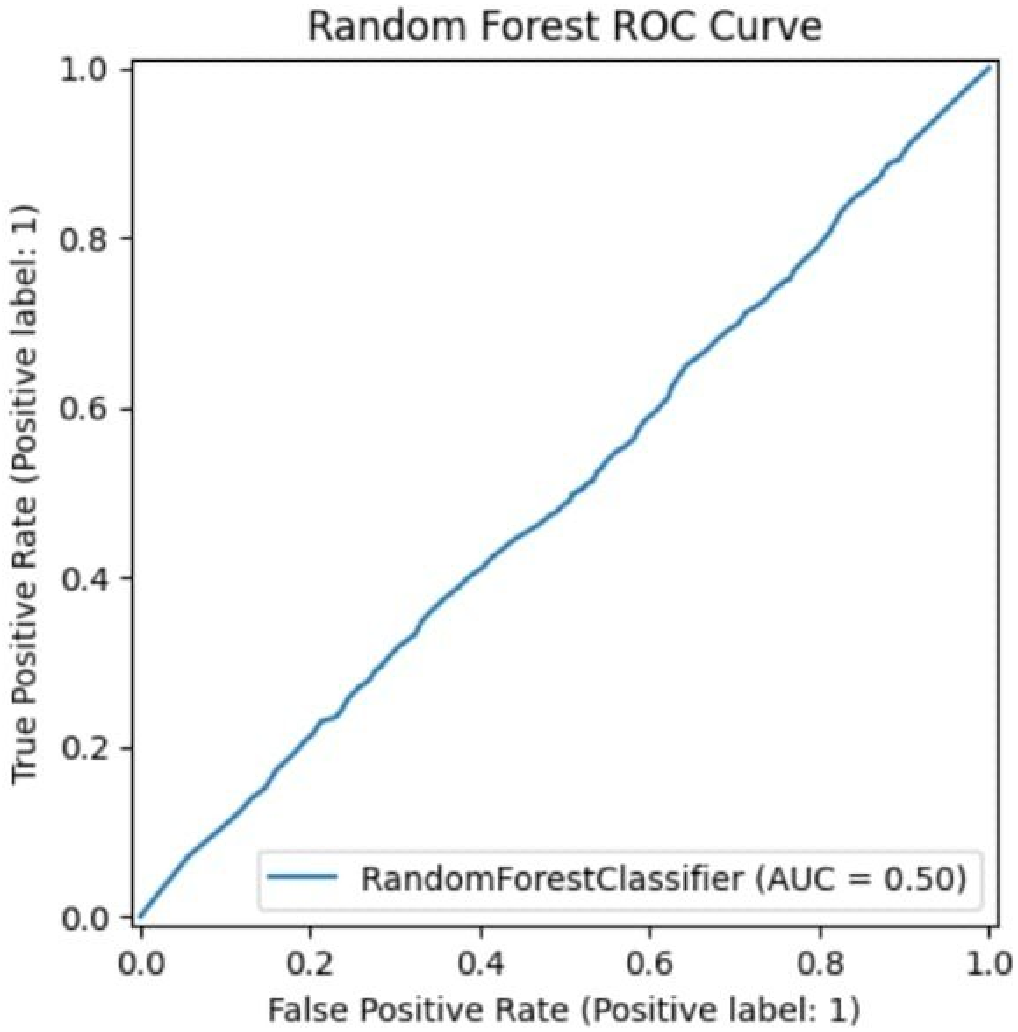
```
--- Random Forest ---
              precision    recall  f1-scor

           0       0.49      0.48      0.4
           1       0.49      0.51      0.5

    accuracy                           0.4
   macro avg       0.49      0.49      0.4
weighted avg       0.49      0.49      0.4

ROC AUC Score: 0.49193229830745766
```


Random Forest Confusion Matrix

Random Forest ROC Curve

[redacted]



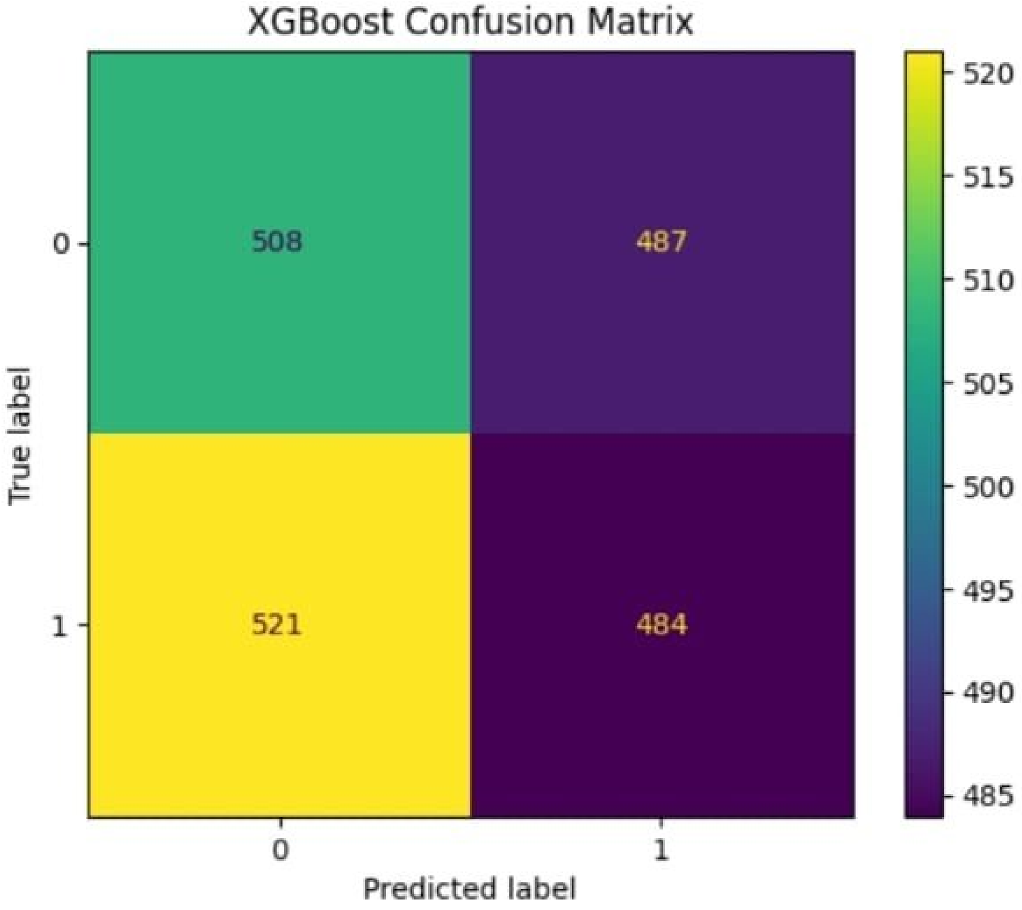


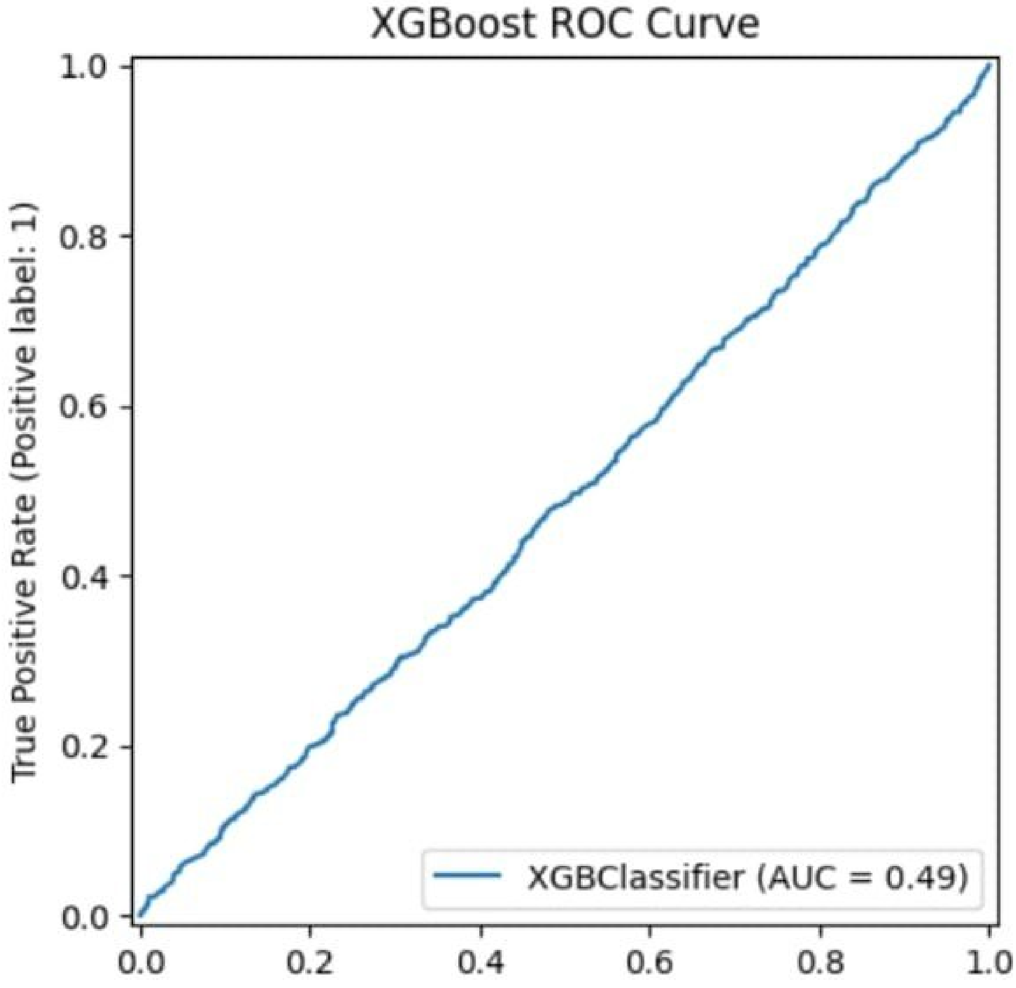
```
--- XGBoost ---
              precision    recall  f1-scor

           0       0.49      0.51      0.5
           1       0.50      0.48      0.4

    accuracy                           0.5
   macro avg       0.50      0.50      0.5
weighted avg       0.50      0.50      0.5

ROC AUC Score: 0.4960724018100452
```

XGBoost Confusion Matrix

[redacted]

```
   macro avg       0.50      0.50      0.5
weighted avg       0.50      0.50      0.5

ROC AUC Score: 0.4960724018100452
```




XGBoost ROC Curve
True Positive Rate (Positive label: 1)
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
XGBClassifier (AUC = 0.49)