**Fraud Detection**

**Ali Bakhshesh**

**Data Analysis**

This dataset includes information of financial transactions. The target shows if a transaction is fraud or not. Our task in this project is that detect the fraud transactions using classifications model.

The dataset includes following features:

* trans\_date\_trans\_time
* cc\_num
* merchant
* category
* amt
* first
* last
* gender
* street
* city
* state
* zip
* lat
* long
* city\_pop
* job
* dob
* trans\_num
* unix\_time
* merch\_lat
* merch\_long
* is\_fraud

**Data Preprocessing**

1. **Drop some columns:**

In this step we eliminated some columns which we thought that are not necessary to predict if a transaction is fraud or not.

Thes columns are:

First – last – trans\_date\_trans\_time – dob – trans\_num

Also, we omitted some features which were highly correlated or which have the same meaning as other features these features includes:

Merch\_lat – merch\_long – city – state – street

1. **Data Encoding:**

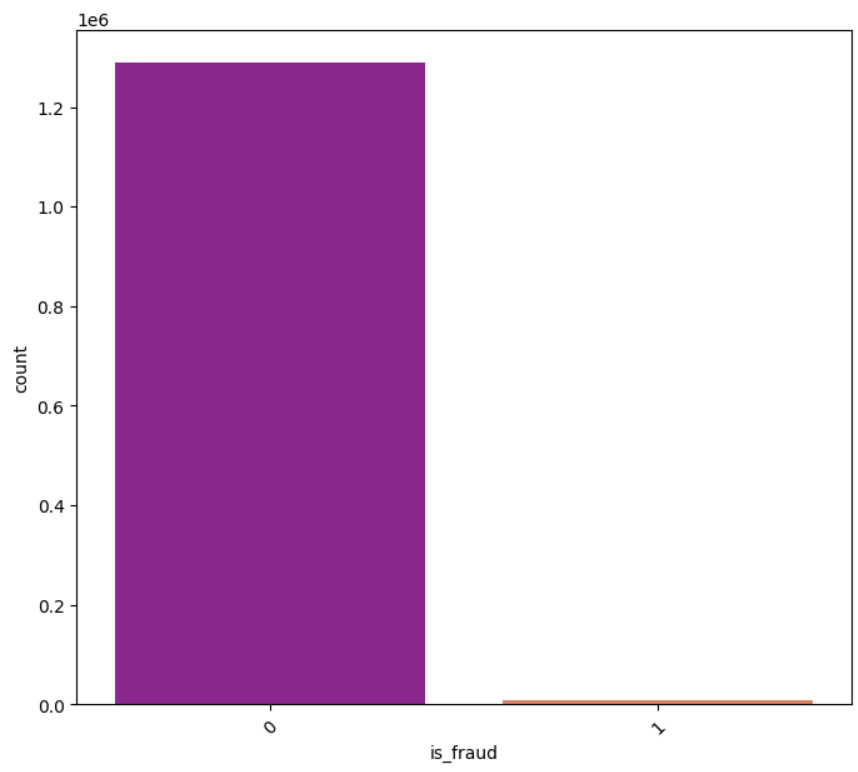
In this step we have some categorical and some string data that we should change them so that we can give them as input to the model. For *gender* it is obvious that we should use *OneHotEncoder.*

But for other features consisting of *job, category* and *merchant* we’ve used count encoding.

1. **Data Balancing:**

This is the most important step in our preprocessing.

A blue circle with black text

Description automatically generatedAccording to the following plots the target of this dataset is imbalanced.

So we must use some techniques to balance data. We have used “sklearn” OverSampler and UnderSampler simultaneously.

A blue and orange circle with text

Description automatically generatedAt the end we have 300000 data record with this portion:

A purple and orange bars

Description automatically generated

1. **Data Scaling:**

At the end of preprocessing step we have scaled train and test data.

**Model Selection**

In this step we should train six kinds of classification models on our data, and we will report the results for each model.

These six models are:

Logistic Regression – SVM – Decision Tree – Random Forest – KNN – Naive Bayes

In the following we will report results for each model on the train and test data.

**Logistic Regression**

Train data:

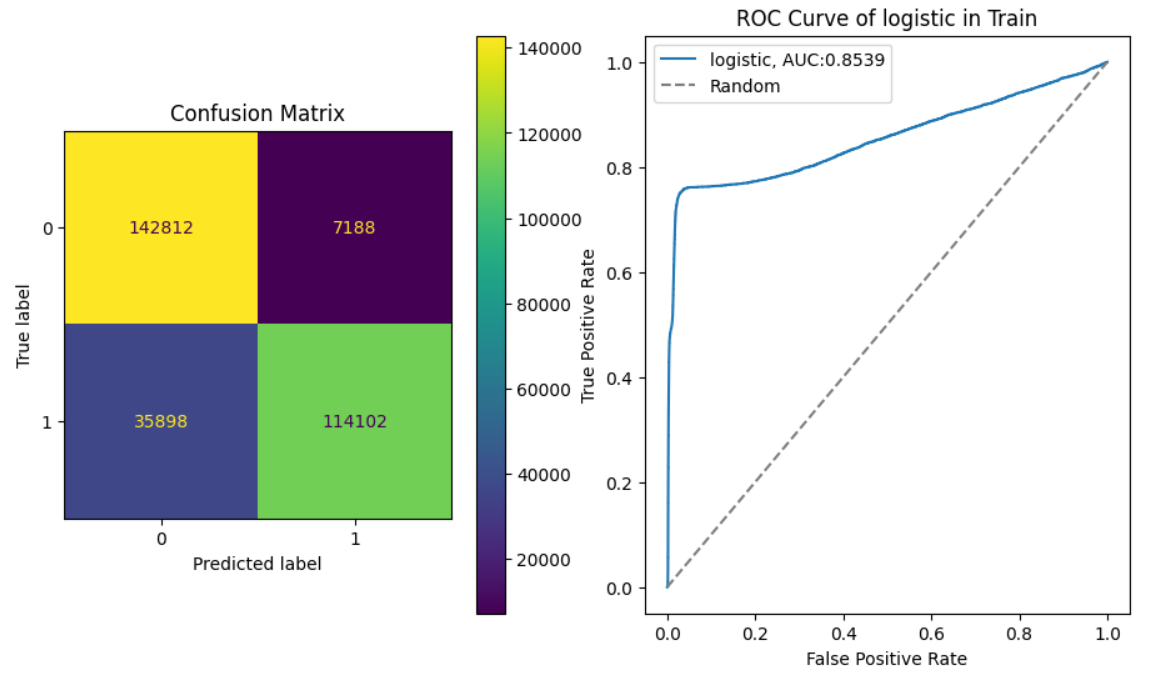
Accuracy Score: 0.8563

F1 Score: 0.8411

Recall Score: 0.7606

Precision Score: 0.9407

ROC AUC: 0.8538



Test data:

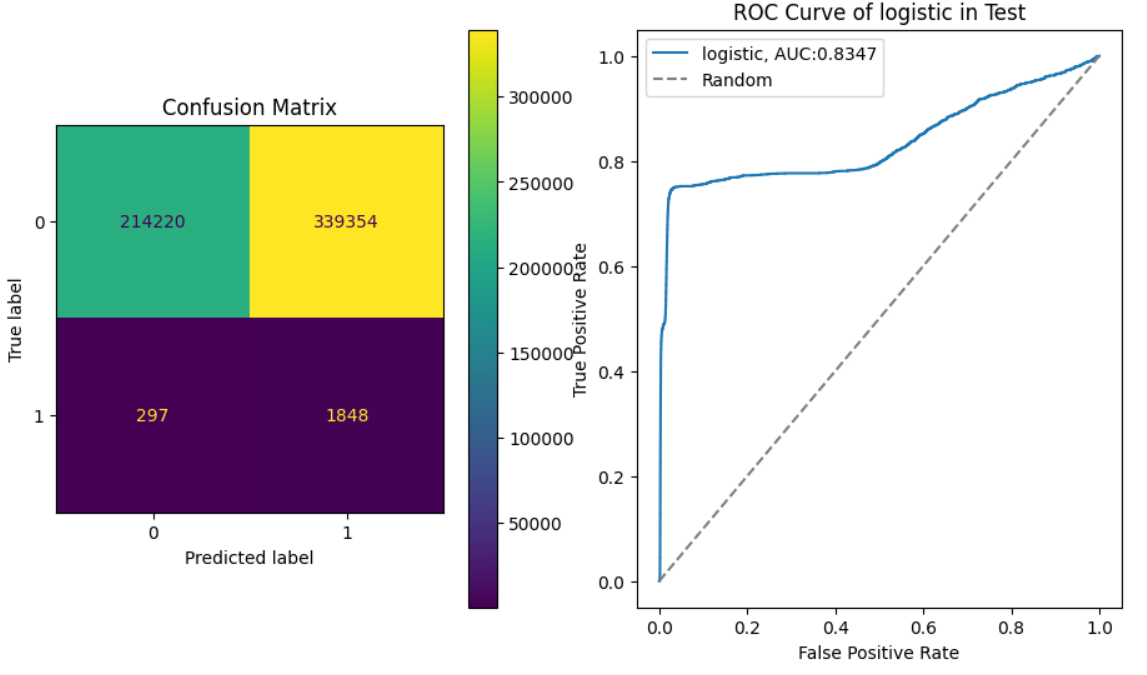
Accuracy Score: 0.3888

F1 Score: 0.01076

Recall Score: 0.8615

Precision Score: 0.0054

ROC AUC: 0.8347



**SVM**

Train data:

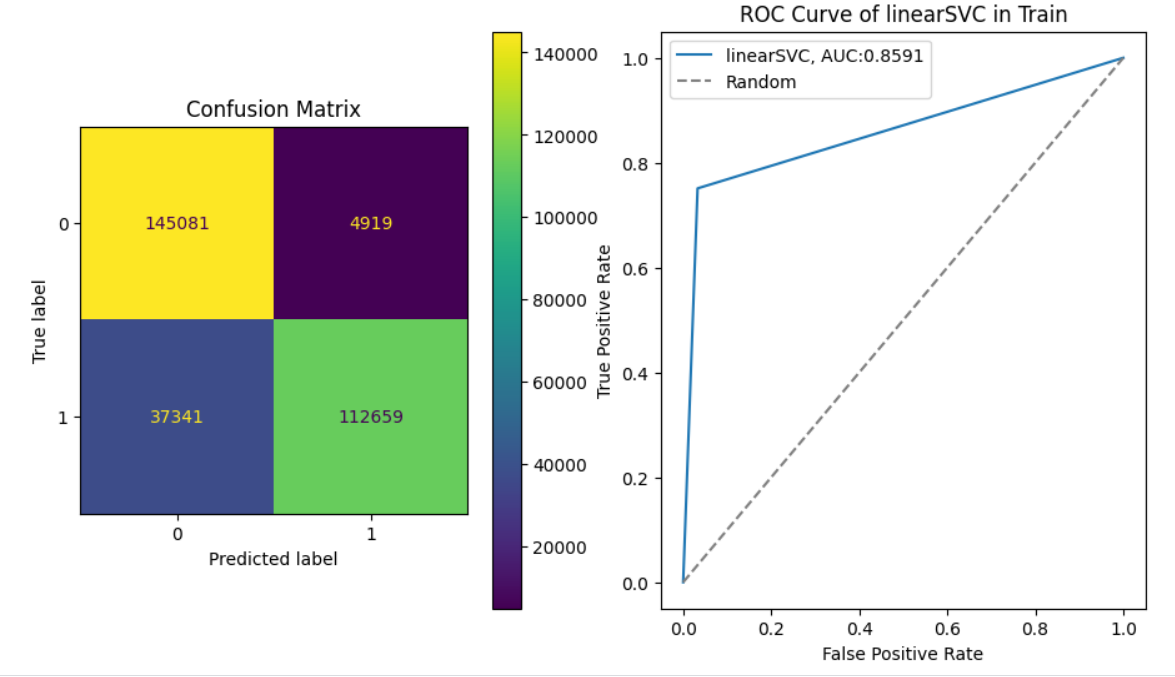
Accuracy Score: 0.85913

F1 Score: 0.8420

Recall Score: 0.75106

Precision Score: 0.9581

ROC AUC: 0.8591



Test data:

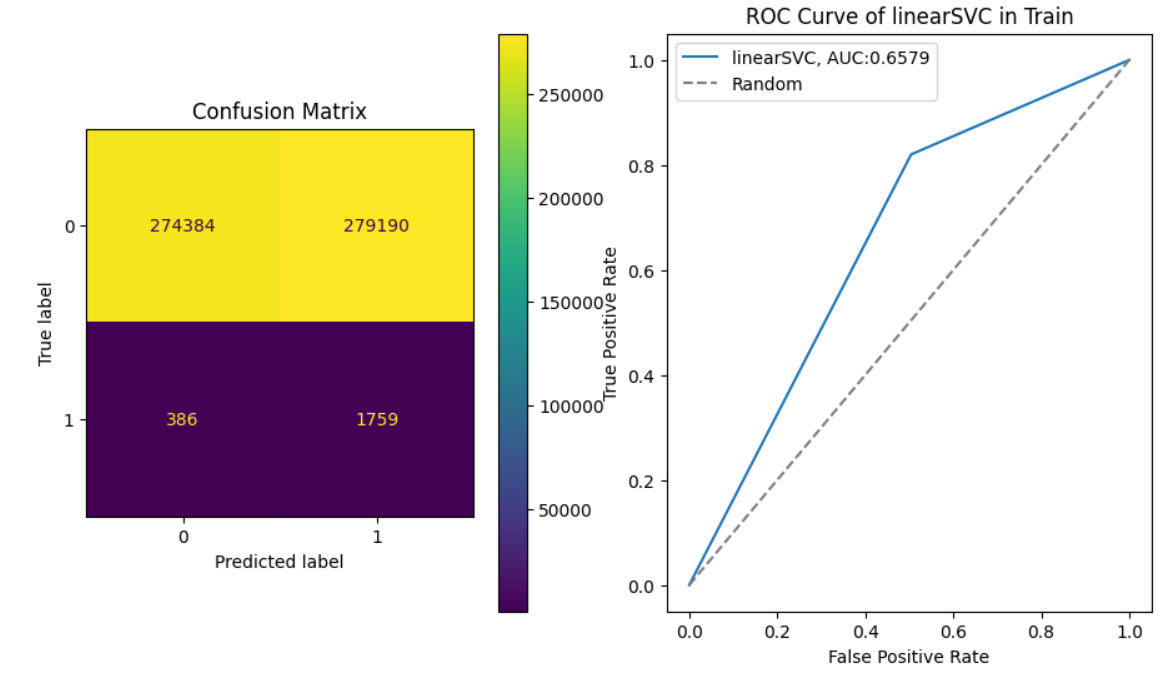
Accuracy Score: 0.4969

F1 Score: 0.012

Recall Score: 0.8200

Precision Score: 0.0062

ROC AUC: 0.6578



**Decision Tree**

Train data:

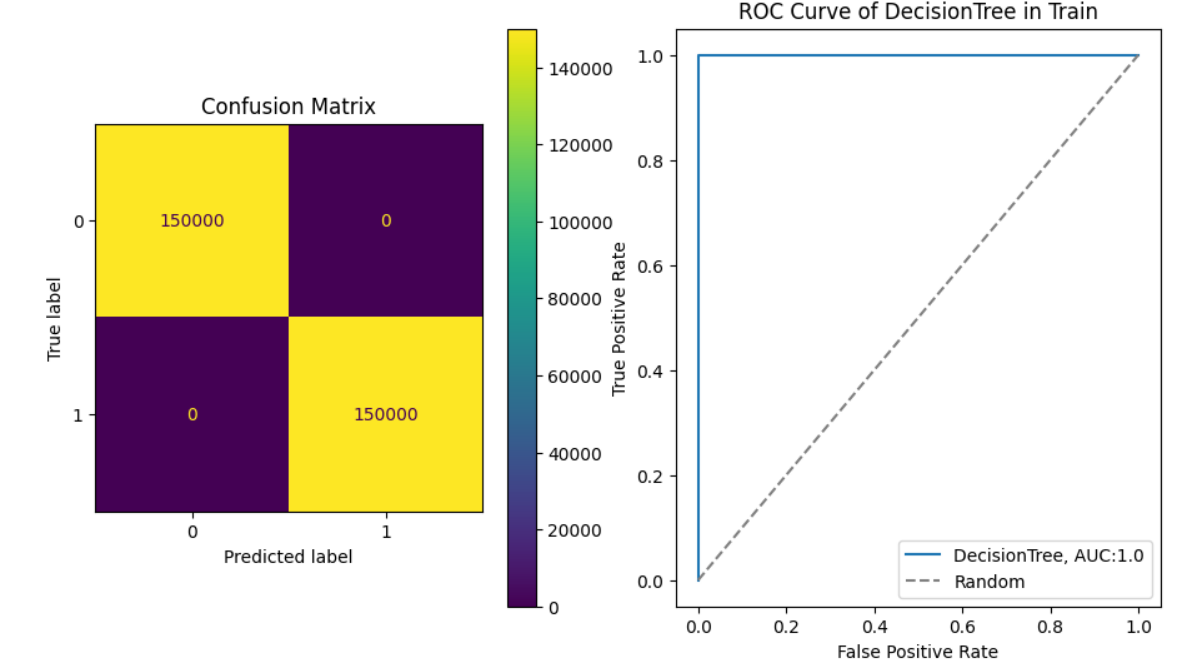
Accuracy Score: 1.0

F1 Score: 1.0

Recall Score: 1.0

Precision Score: 1.0

ROC AUC: 1.0



Test data:

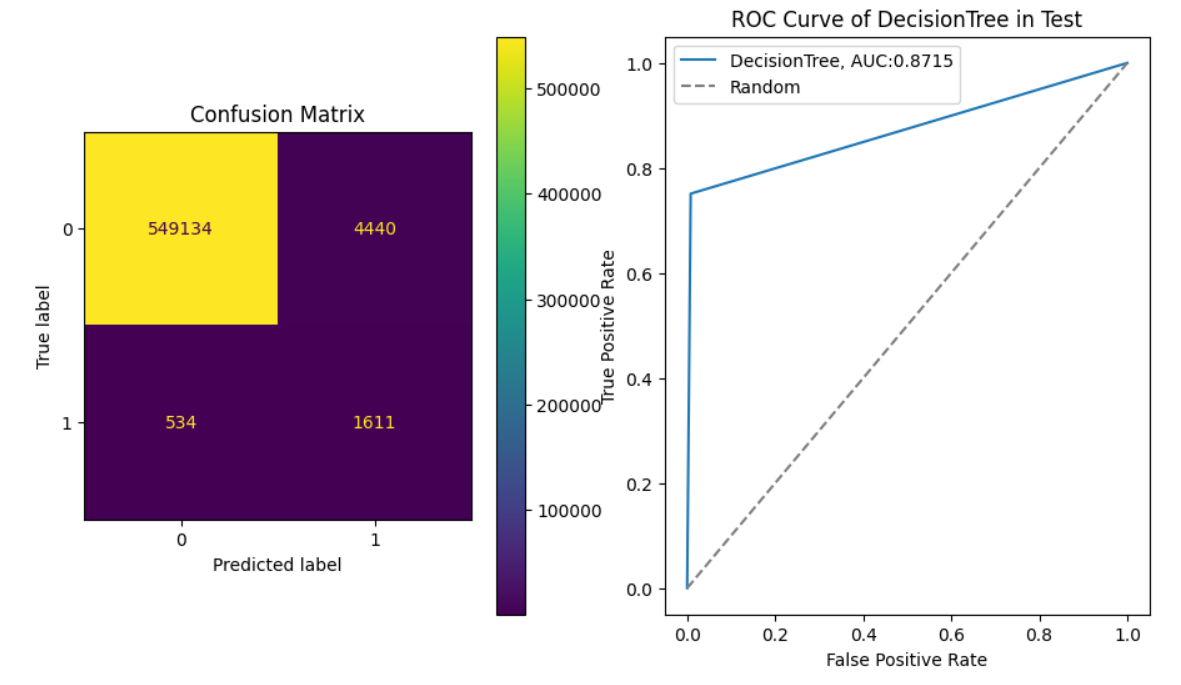
Accuracy Score: 0.9910

F1 Score: 0.3931

Recall Score: 0.7510

Precision Score: 0.2662

ROC AUC: 0.8715



**Random Forest**

Train data:

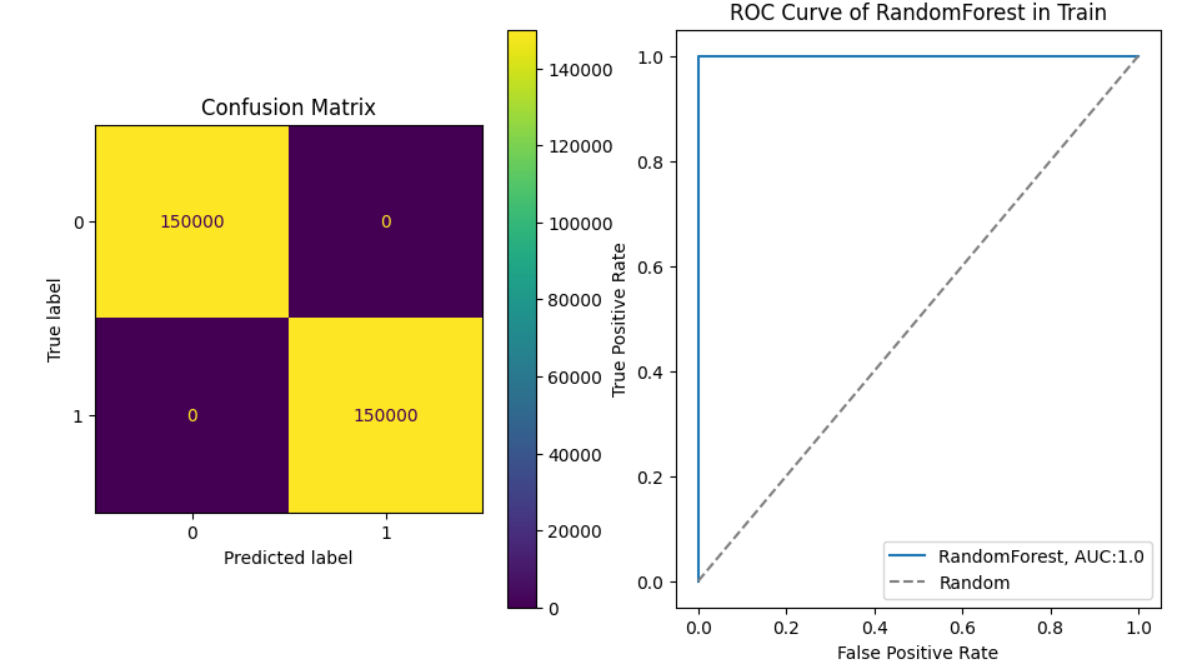
Accuracy Score: 1.0

F1 Score: 1.0

Recall Score: 1.0

Precision Score: 1.0

ROC AUC: 1.0



Test data:

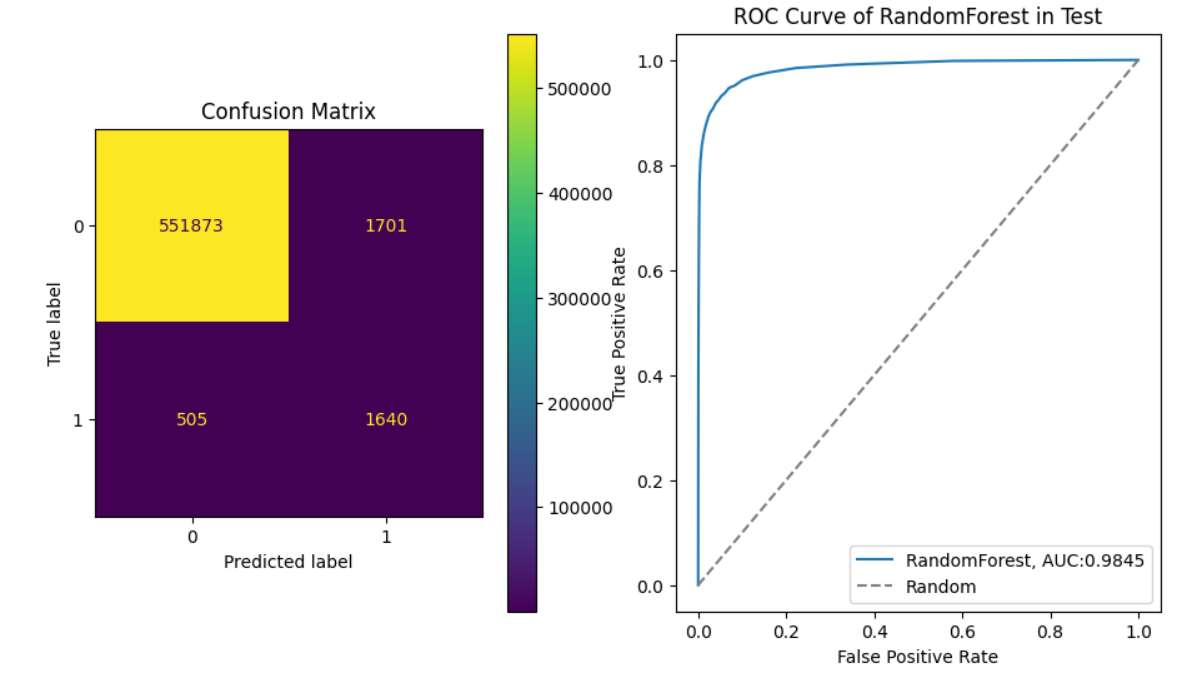
Accuracy Score: 0.9960

F1 Score: 0.5978

Recall Score: 0.7645

Precision Score: 0.4908

ROC AUC: 0.9845



**KNN**

Train data:

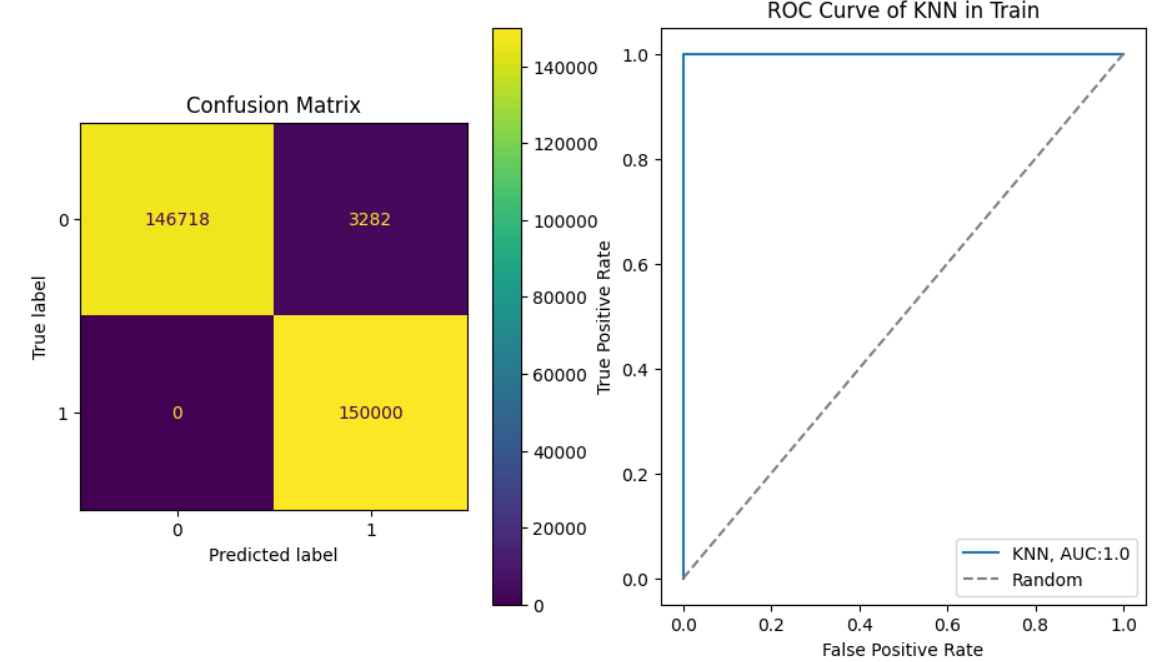
Accuracy Score: 0.98906

F1 Score: 0.9891783884305696

Recall Score: 1.0

Precision Score: 0.9785884839707206

ROC AUC: 1.0



Test data:

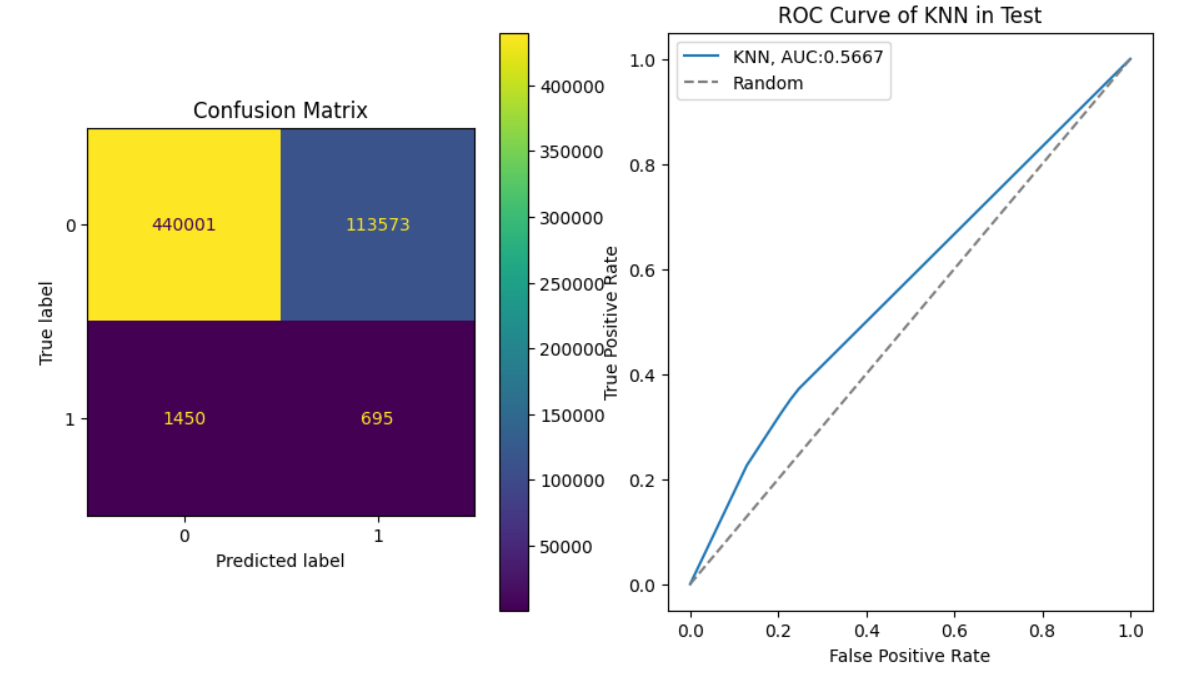
Accuracy Score: 0.7930194936649637

F1 Score: 0.011940247223248263

Recall Score: 0.32400932400932403

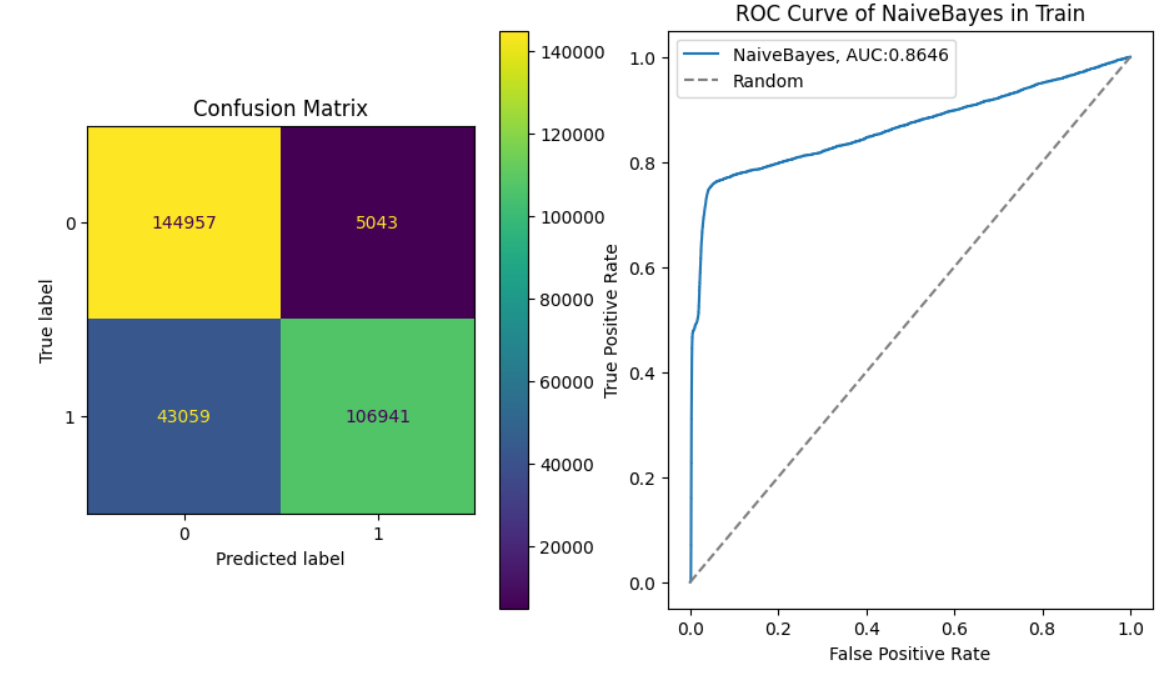
Precision Score: 0.006082192739874681

ROC AUC: 0.566735887128644



**Naive Bayes**

Train data:

Accuracy Score: 0.83966

F1 Score: 0.8163933675338951

Recall Score: 0.71294

Precision Score: 0.9549667809687098

ROC AUC: 0.8645951737333334

Test data:

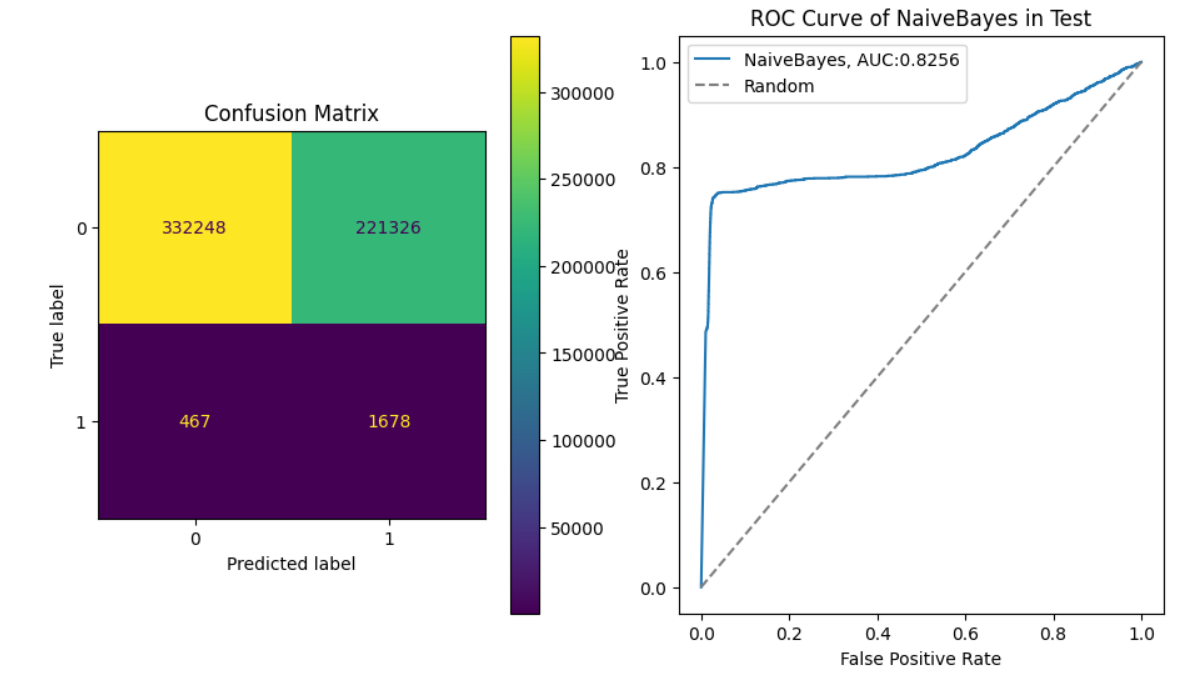
Accuracy Score: 0.6008900181566583

F1 Score: 0.014905684679923073

Recall Score: 0.7822843822843822

Precision Score: 0.007524528708005238

ROC AUC: 0.8256198523579219



It is obvious that *Decision Tree* and *Random Forest* has the most accuracy and f1. So the results for these models are the best and *Logistic Regression* has the worst performance.