# Knowledge Discovery and Data Mining

## Stevens Institute of Technology

**Group Project**

Credit Card Fraud Detection

**By**

| | | |
|---|---|---|
| Hem UrmeshkumarPatel | – | 20011907 |
| Bijalben Utpal Patel | – | 20009241 |
| Kishan Patel | – | 20009039 |
| Poorvi Rajendra Raut | – | 20009560 |

# Agenda

01 Project Definition

02 Dataset Overview

03 Project Walkthrough

04 Machine Learning Models

05 Results

01

# Project Definition

## Credit Card Fraud Detection

The project definition is as simple as that whether a credit card transaction is a normal transaction done by user or fraud transaction. But to predict or classify fraud transaction it takes a lot of considerations.

It is a serious threat for both financial institutions and individuals. Both parties may suffer large financial losses.

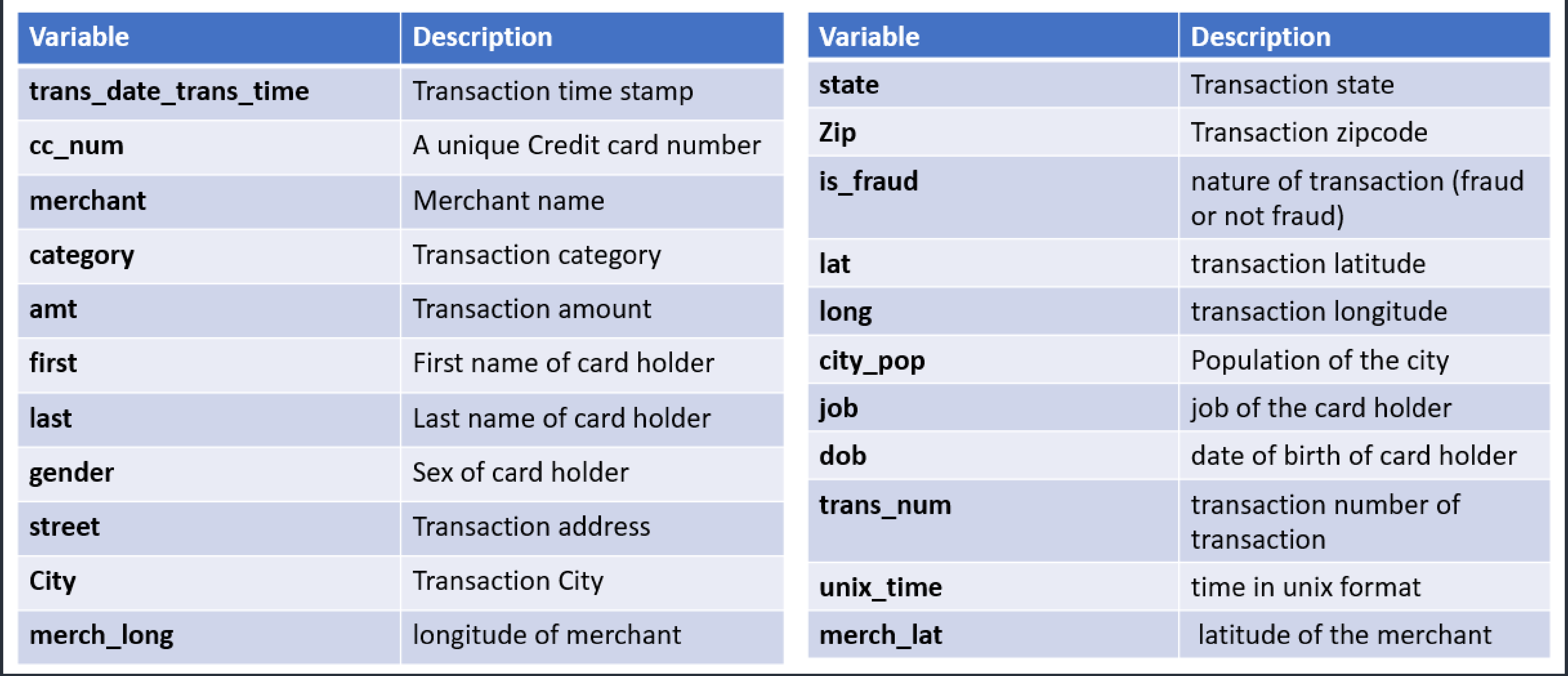
The ultimate goal is to create a system for detecting credit card fraud that can precisely identify fraudulent transactions.

02

# Dataset Overview

- Dataset : Kaggle2
- A dataset is provided containing all transactions' details, which holds a lot of important features to identify whether it is a fraud or not.
- Every row in the dataset, represents details of a single transaction and every column represents a valuable attribute.
- This dataset is already been divided into training and testing sets, but we have merged it together and now entire dataset contains 1,852,394 rows and 22 columns.

# 02 Dataset Overview



| Variable | Description |
|---|---|
| trans_date_trans_time | Transaction time stamp |
| cc_num | A unique Credit card number |
| merchant | Merchant name |
| category | Transaction category |
| amt | Transaction amount |
| first | First name of card holder |
| last | Last name of card holder |
| gender | Sex of card holder |
| street | Transaction address |
| City | Transaction City |
| merch_long | longitude of merchant |

| Variable | Description |
|---|---|
| state | Transaction state |
| Zip | Transaction zipcode |
| is_fraud | nature of transaction (fraud or not fraud) |
| lat | transaction latitude |
| long | transaction longitude |
| city_pop | Population of the city |
| job | job of the card holder |
| dob | date of birth of card holder |
| trans_num | transaction number of transaction |
| unix_time | time in unix format |
| merch_lat | latitude of the merchant |

03

# Project Walkthrough

01 Pre-Processing Data

02 Exploratory Data Analysis

03 Feature Encoding

04 Balancing Data

03

# Project Walkthrough

01

## Pre-processing Data

- In dataset we have DOB column which is of object type that cannot be incorporated directly into our model, so will derive age from the same.

- Similarly, we have derived hour, day and month-year form trans_date_trans_time column, because we cannot use date-time object to implement any Machine Learning model.

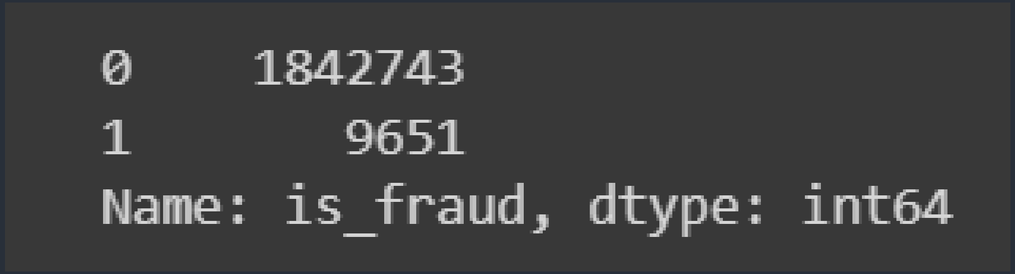
- After checking missing values, Scaling has been done on the entire dataset, too.

# 03 Project Walkthrough

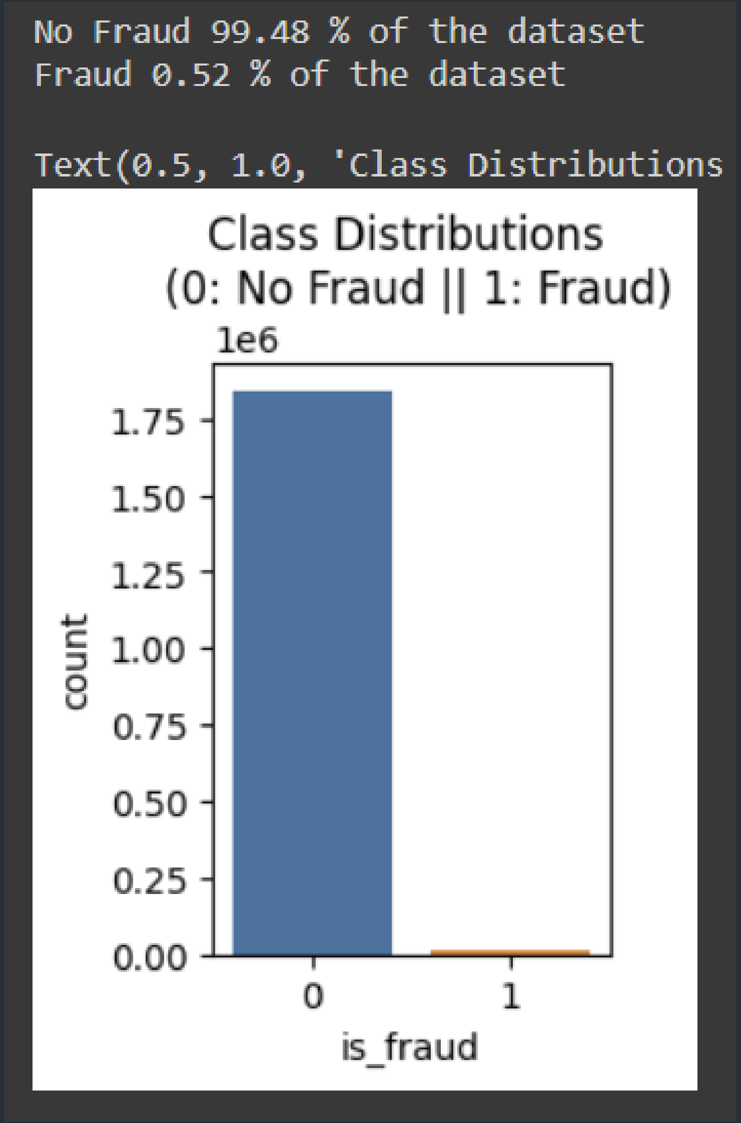
## 02 Exploratory Data Analysis

- Starting EDA from target variable, which is is_fraud column according to our dataset.

- Here, we can see that dataset is highly skewed, which needs to be fixed. Otherwise while implementing ML models it will have biasness for high volume data.



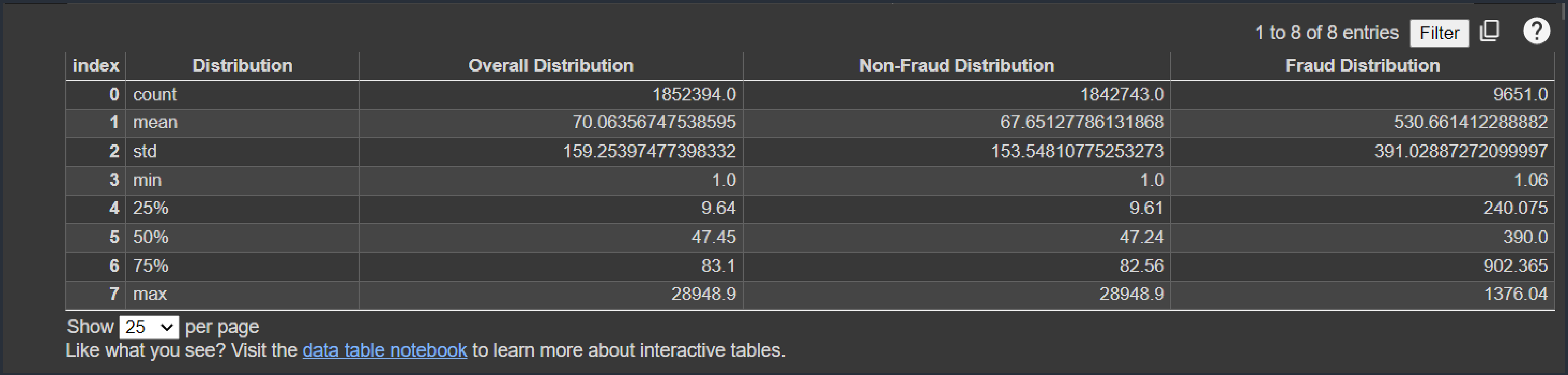
```
0    1842743
1       9651
Name: is_fraud, dtype: int64
```



```
No Fraud 99.48 % of the dataset
Fraud 0.52 % of the dataset

Text(0.5, 1.0, 'Class Distributions
```

# 03 Project Walkthrough

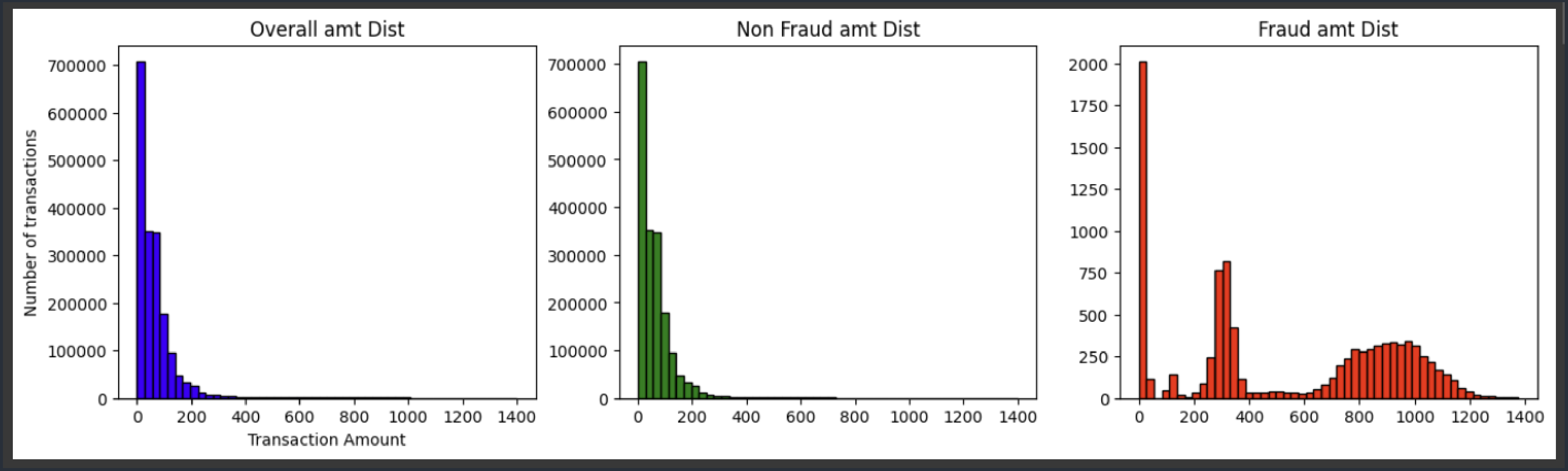


1 to 8 of 8 entries Filter

| index | Distribution | Overall Distribution | Non-Fraud Distribution | Fraud Distribution |
|---|---|---|---|---|
| 0 | count | 1852394.0 | 1842743.0 | 9651.0 |
| 1 | mean | 70.06356747538595 | 67.65127786131868 | 530.661412288882 |
| 2 | std | 159.25397477398332 | 153.54810775253273 | 391.02887272099997 |
| 3 | min | 1.0 | 1.0 | 1.06 |
| 4 | 25% | 9.64 | 9.61 | 240.075 |
| 5 | 50% | 47.45 | 47.24 | 390.0 |
| 6 | 75% | 83.1 | 82.56 | 902.365 |
| 7 | max | 28948.9 | 28948.9 | 1376.04 |

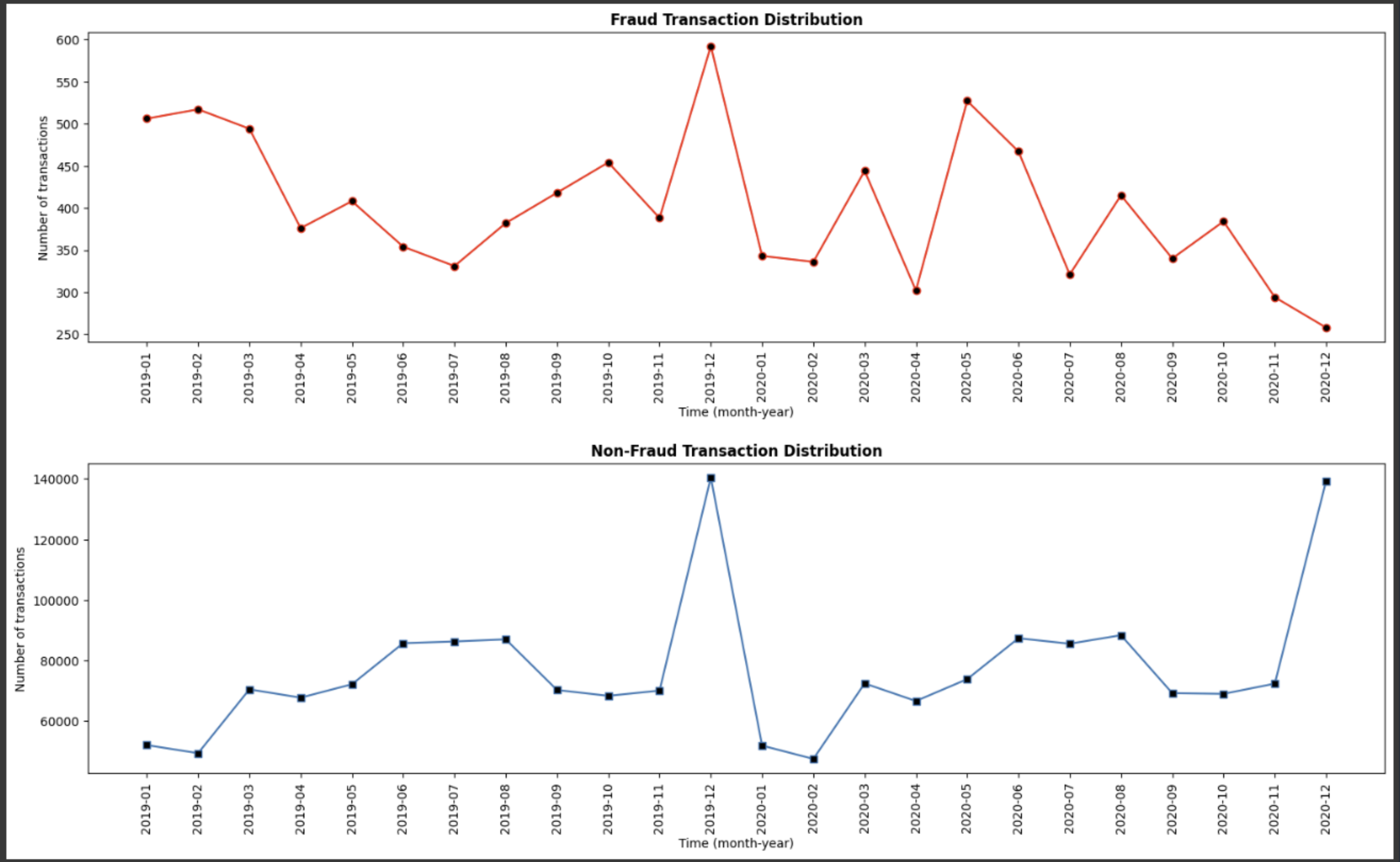
Show 25 per page

Like what you see? Visit the data table notebook to learn more about interactive tables.



03

# Project Walkthrough

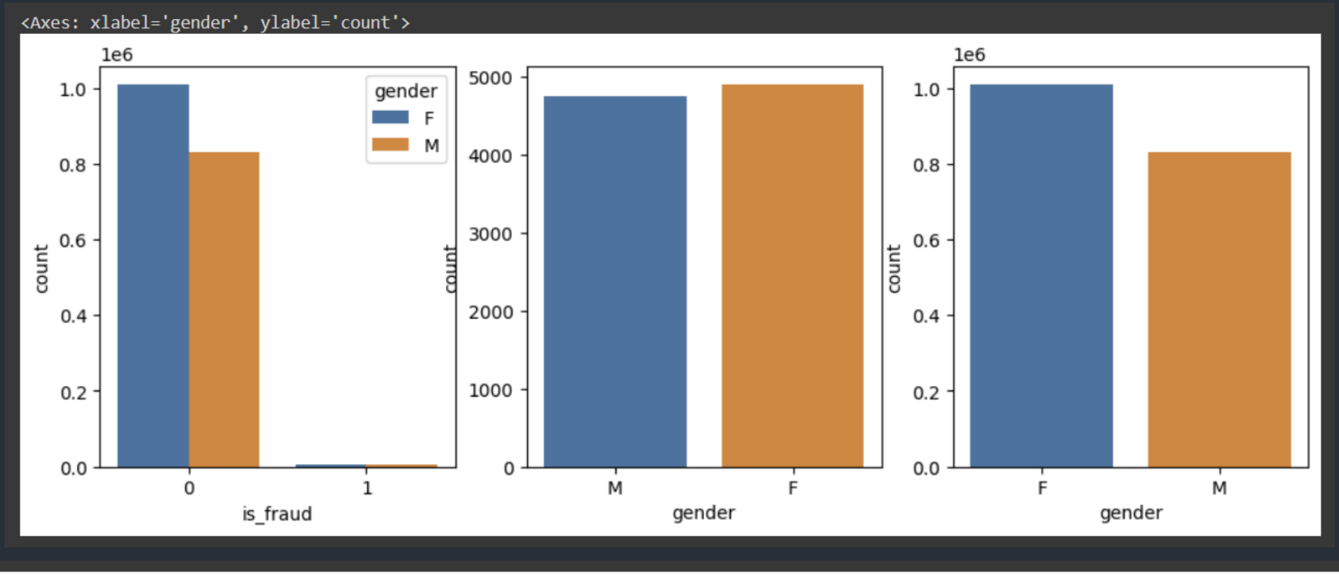


03

# Project Walkthrough

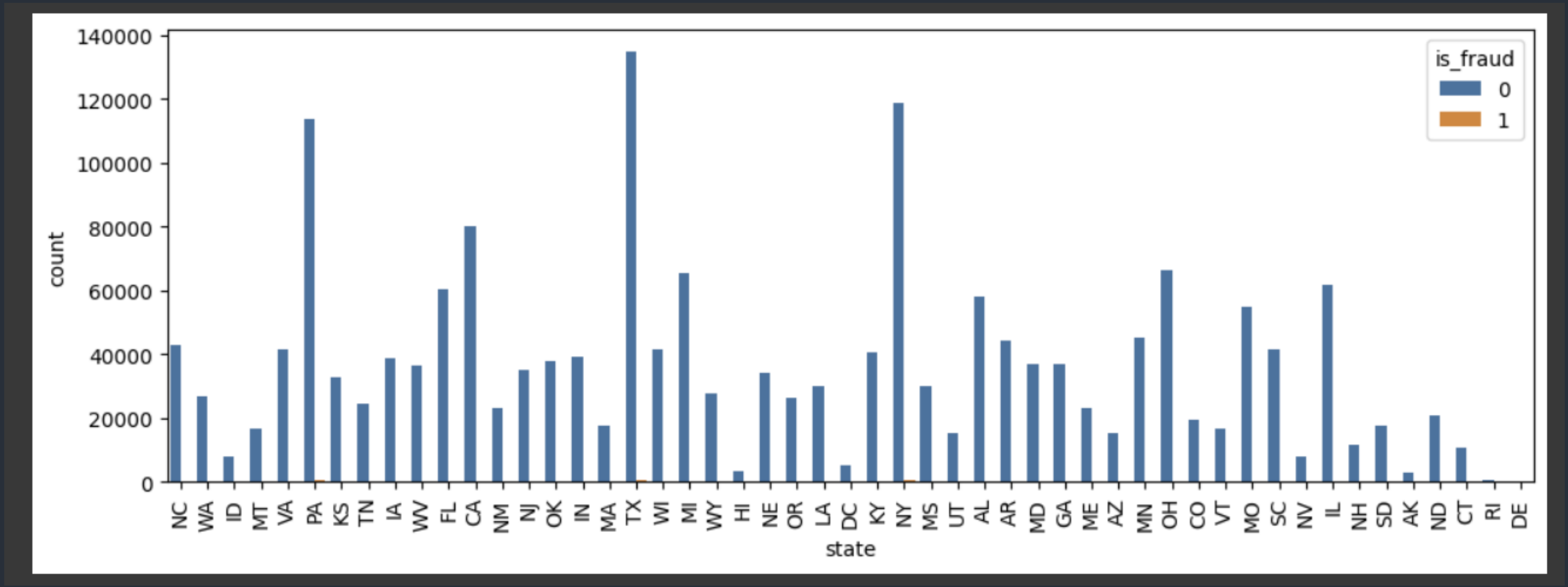
## EDA

- Target variable vs Gender



```
<Axes: xlabel='gender', ylabel='count'>
```

- Target variable vs State



*similarly, we have explored all the features.

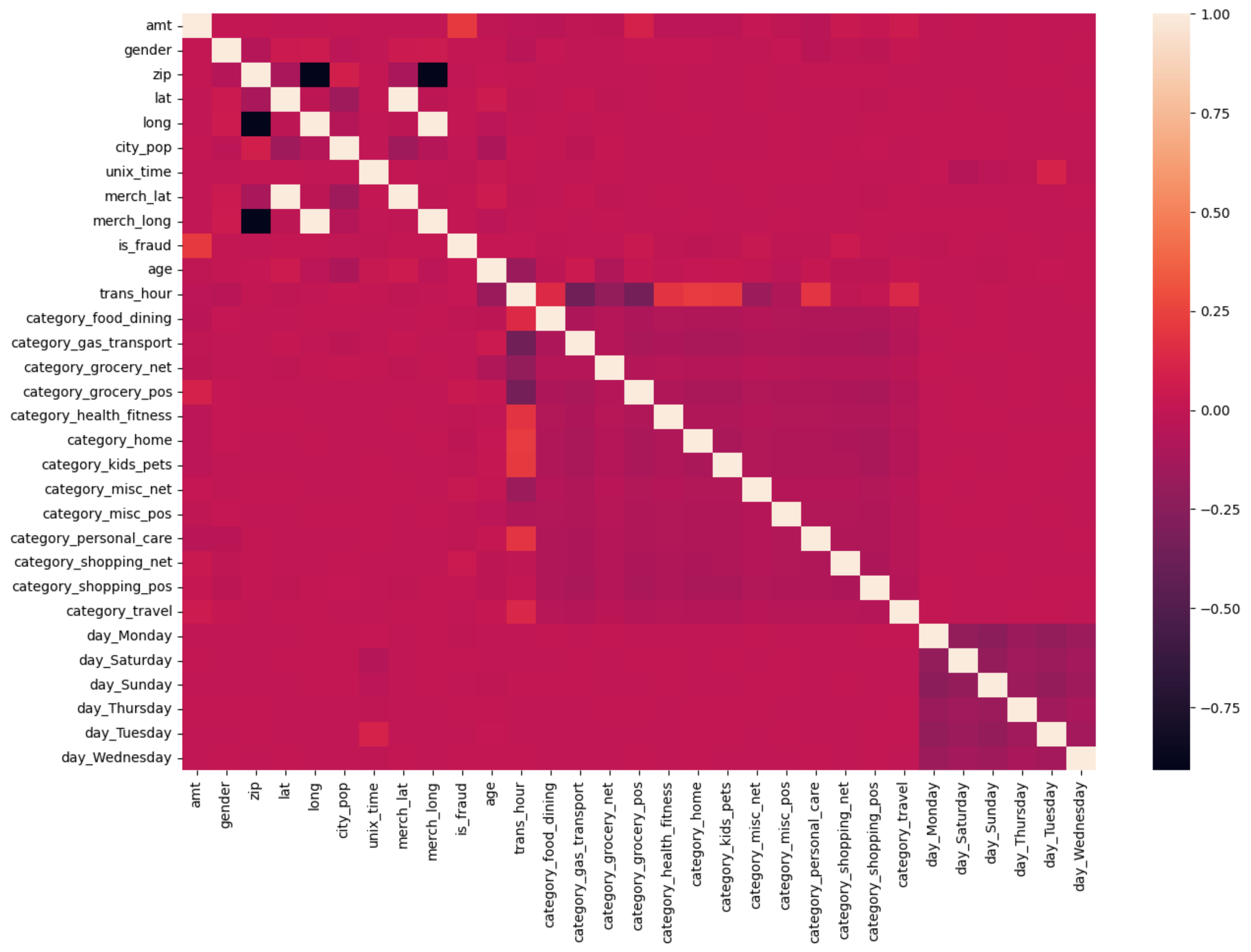
03

# Project Walkthrough

## 03 Feature Encoding and Correlation

- One hot encoding
- Direct Mapping

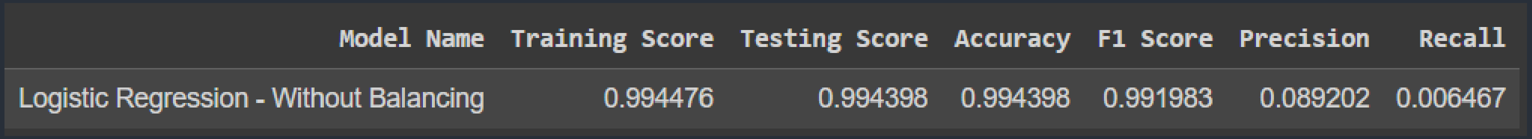
*Here, we have dropped some of the unnecessary and encoded columns.



# 03 Project Walkthrough

## 04 Balancing Data

- We have implemented one Logistic Regression model without balancing the skewness in the data and we got following result.



| Model Name | Training Score | Testing Score | Accuracy | F1 Score | Precision | Recall |
|---|---|---|---|---|---|---|
| Logistic Regression - Without Balancing | 0.994476 | 0.994398 | 0.994398 | 0.991983 | 0.089202 | 0.006467 |

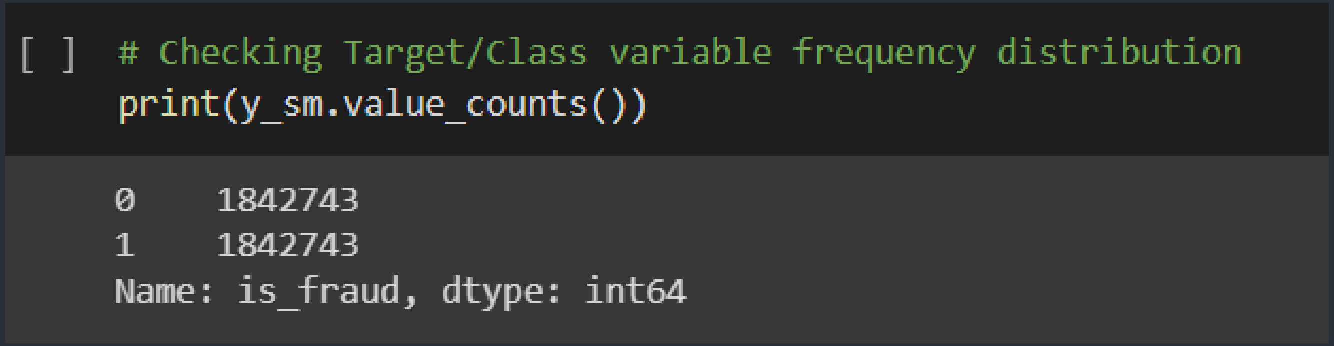
- We got least precision and recall, which shows less precision and completeness to predict true positive of the model.

# 03 Project Walkthrough

## 04 Balancing Data

- We have tried three different sampling method, which are Random Under Sampling, Random Over Sampling and SMOTE method.

- And finalized SMOTE (Synthetic Minority Oversampling Technique) method to balance our dataset.



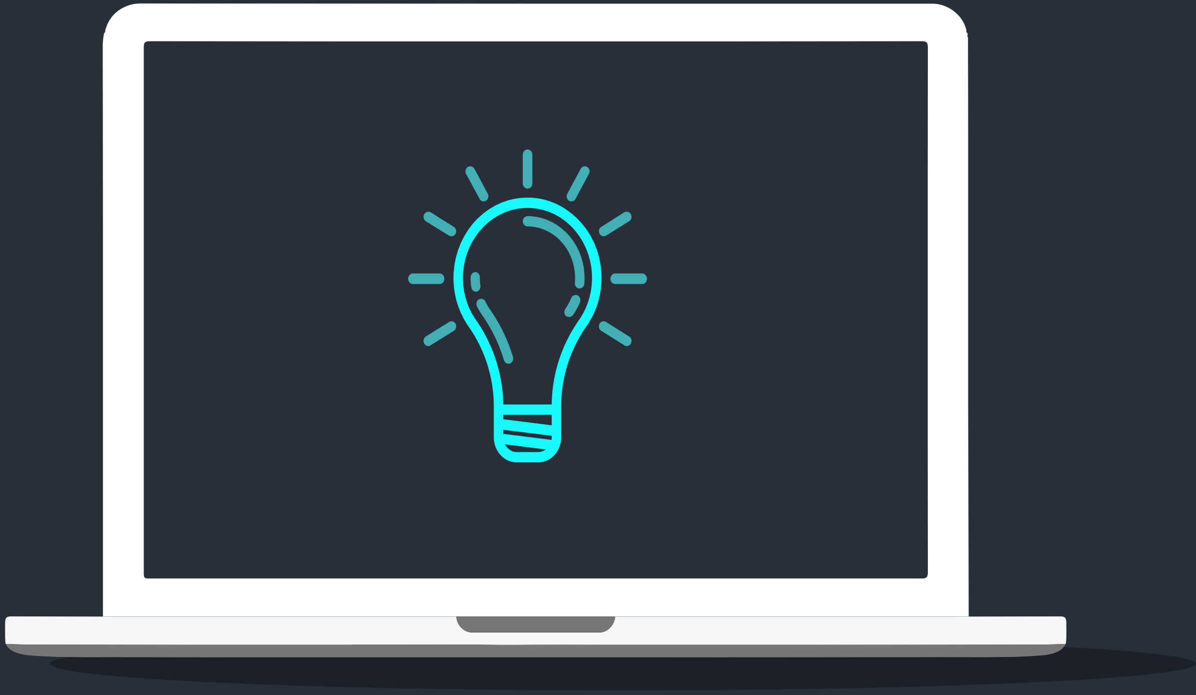
```
[ ]  # Checking Target/Class variable frequency distribution
     print(y_sm.value_counts())

     0    1842743
     1    1842743
     Name: is_fraud, dtype: int64
```

# 04 Machine Learning Models

- We have implemented following

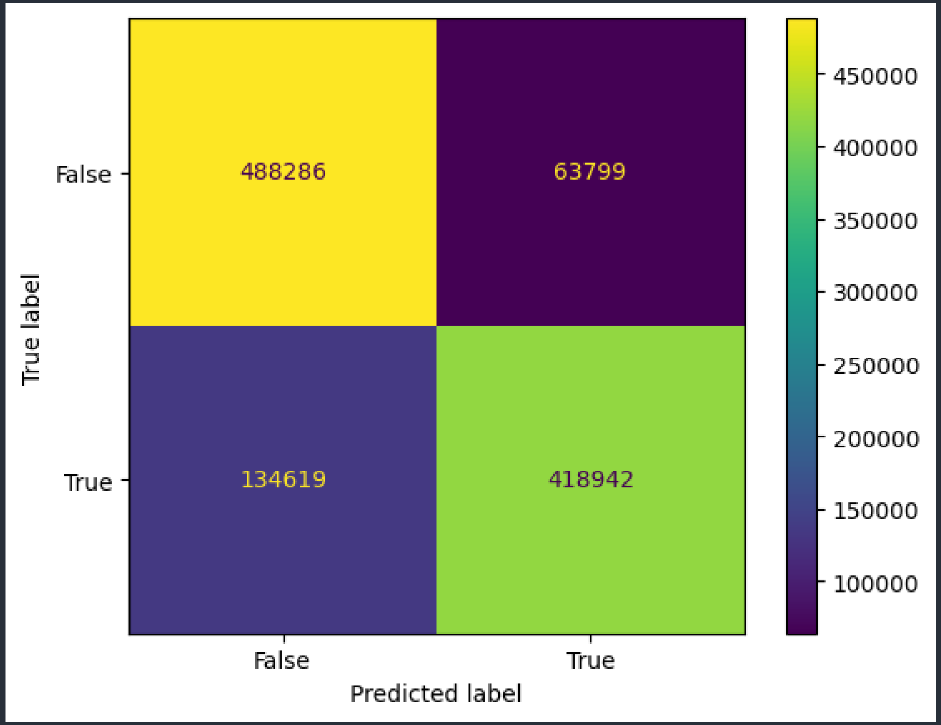
1. Logistic Regression
2. Support Vector Machine
3. K- Nearest Neighbor
4. Naïve Bayes
5. Decision Tree
6. Random Forest Classifier
7. AdaBoost Classifier
8. XGBoost Classifier
9. MLP Classifier

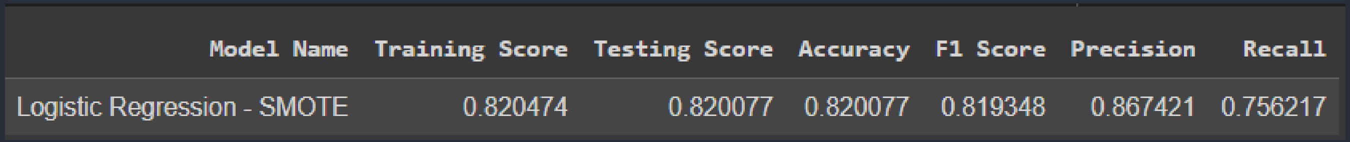


# 04 Machine Learning Models

## Logistic Regression

- It is a process of modeling the probability of a discrete outcome given an input variable.
- The most common logistic regression models a binary outcome; something that can take two values such as true/false, yes/no, and so on. That is why it is useful to classify our dataset into two categories.
- It is one of the useful analysis methods for classification problems.





| Model Name | Training Score | Testing Score | Accuracy | F1 Score | Precision | Recall |
|---|---|---|---|---|---|---|
| Logistic Regression - SMOTE | 0.820474 | 0.820077 | 0.820077 | 0.819348 | 0.867421 | 0.756217 |

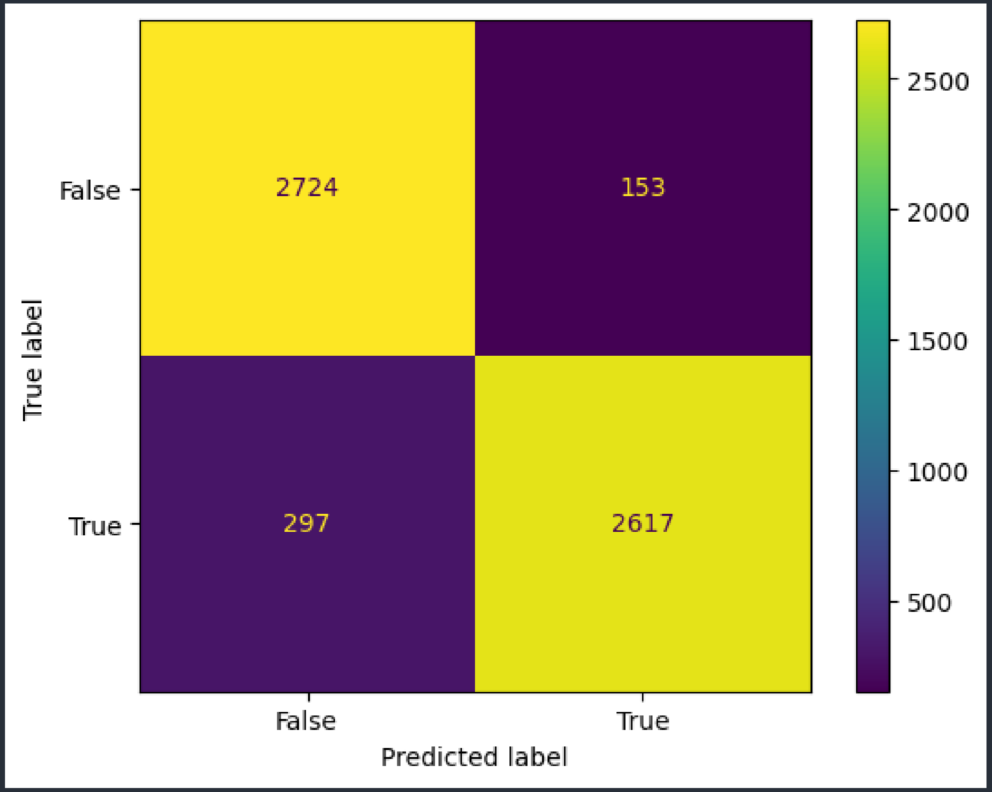
# 04 Machine Learning Models

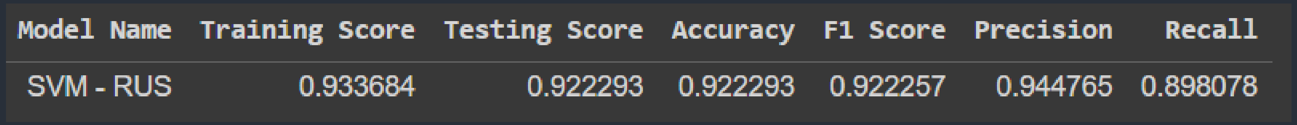
## Support Vector Machine

- It is a linear model for classification and regression problems. Here we have used SVC (Support Vector Classifier).

- Working of SVM algorithm: it creates a line or a hyperplane which separates the data into classes.

- Here, we have used data derived from random under sampling method because SVM is very slow fitting huge amount of data.





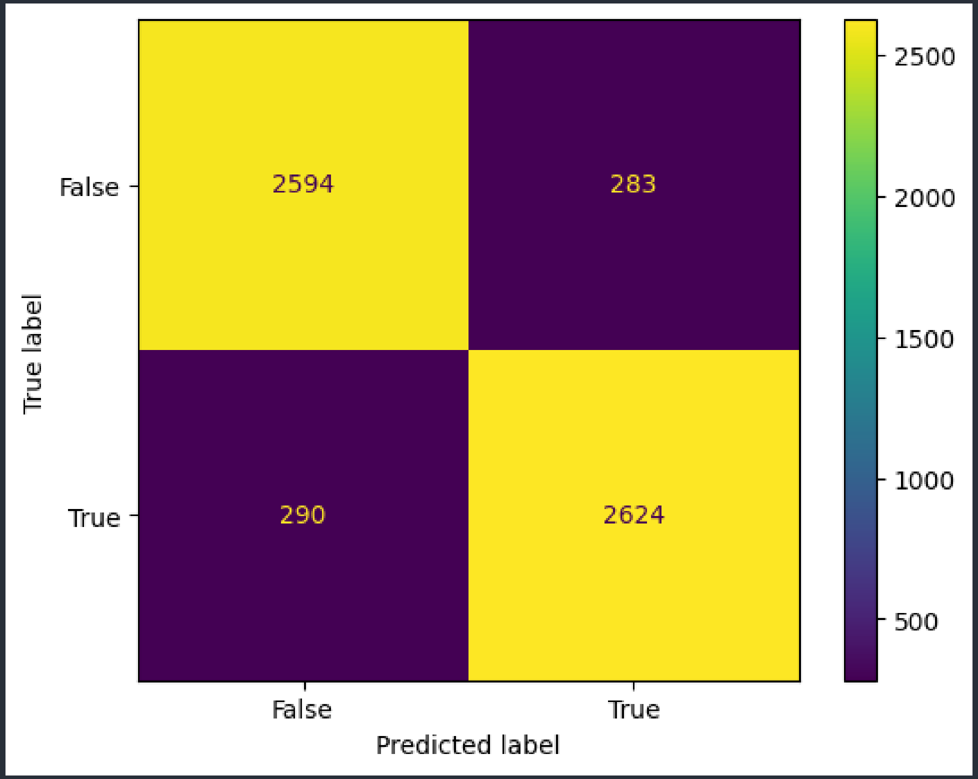
| Model Name | Training Score | Testing Score | Accuracy | F1 Score | Precision | Recall |
|---|---|---|---|---|---|---|
| SVM - RUS | 0.933684 | 0.922293 | 0.922293 | 0.922257 | 0.944765 | 0.898078 |

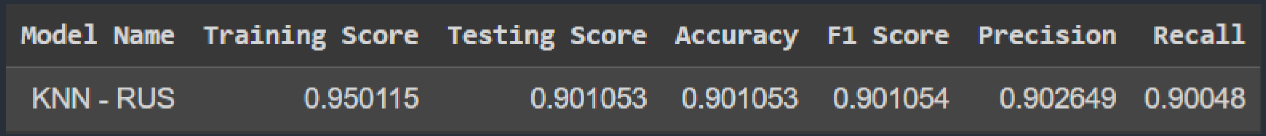
# 04 Machine Learning Models

## K- Nearest Neighbor

- This algorithm is a non-parametric, supervised learning classifier, which uses proximity to make classifications or predictions about the grouping of an individual data point.

- While it can be used for either regression or classification problems, it is typically used as a classification algorithm, working off the assumption that similar points can be found near one another.





| Model Name | Training Score | Testing Score | Accuracy | F1 Score | Precision | Recall |
|---|---|---|---|---|---|---|
| KNN - RUS | 0.950115 | 0.901053 | 0.901053 | 0.901054 | 0.902649 | 0.90048 |

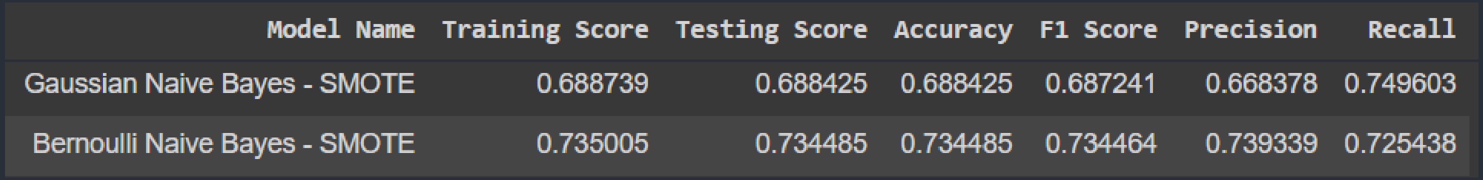
# 04 Machine Learning Models

## Naïve Bayes

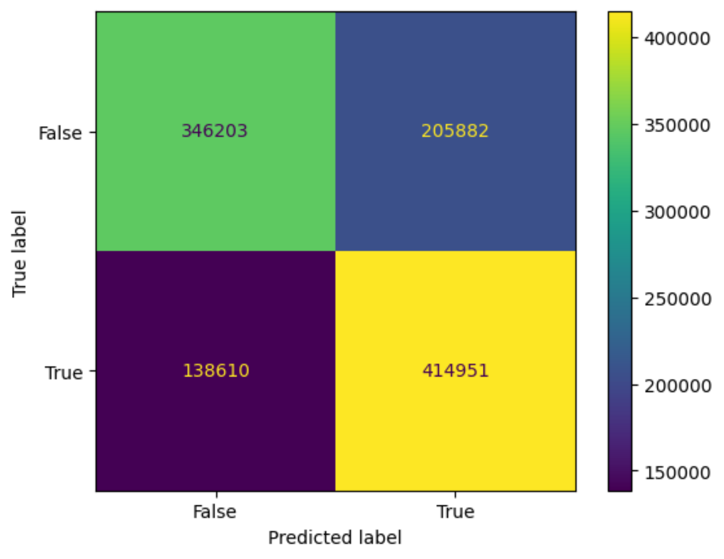
- It is one of the probabilistic machine learning models utilized for classification tasks, which uses Bayes theorem as the foundation.

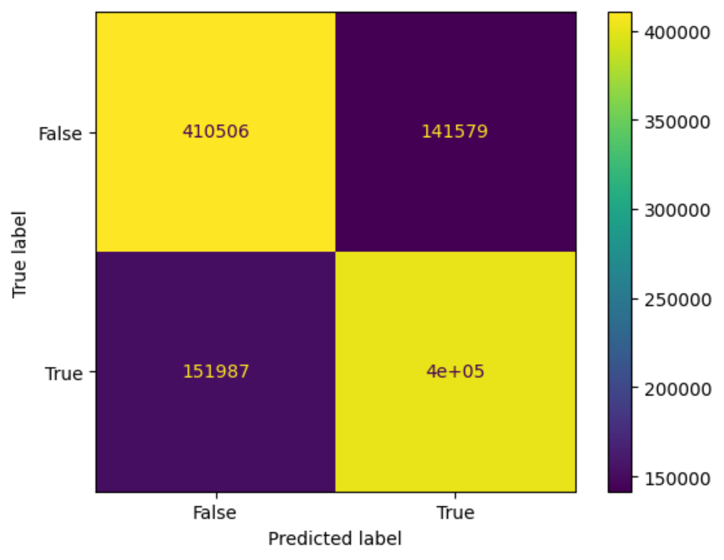
- When B has already happened, we may use the Bayes theorem to calculate the likelihood that A will also occur. Here, A is the hypothesis and B is the supporting evidence.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$



| Model Name | Training Score | Testing Score | Accuracy | F1 Score | Precision | Recall |
|---|---|---|---|---|---|---|
| Gaussian Naive Bayes - SMOTE | 0.688739 | 0.688425 | 0.688425 | 0.687241 | 0.668378 | 0.749603 |
| Bernoulli Naive Bayes - SMOTE | 0.735005 | 0.734485 | 0.734485 | 0.734464 | 0.739339 | 0.725438 |



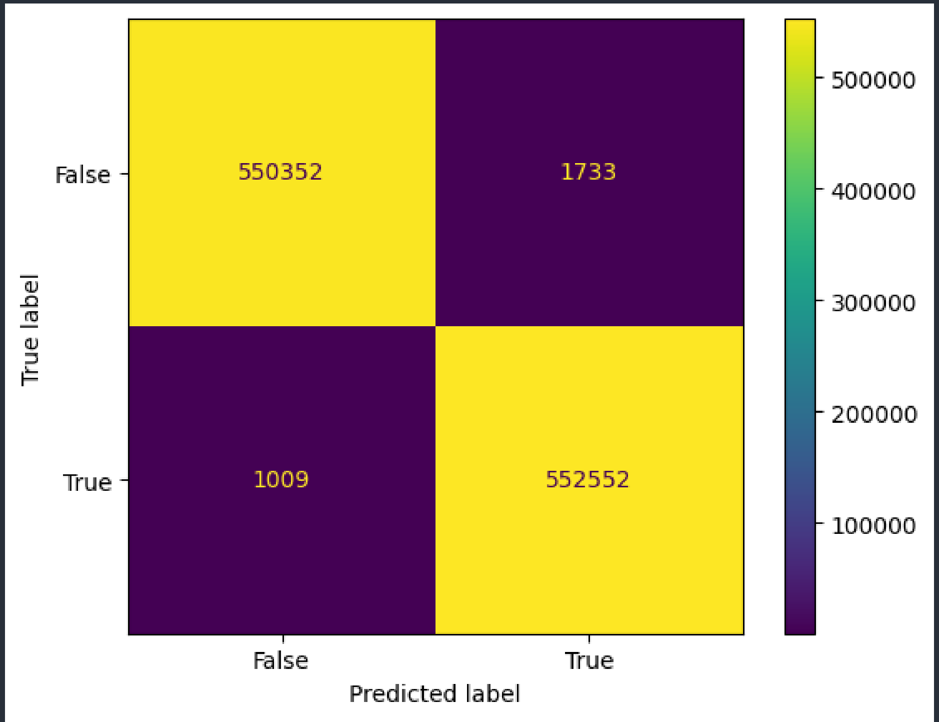


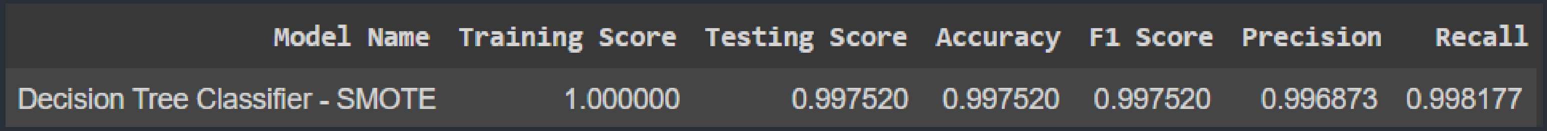
# 04 Machine Learning Models

## Decision Tree

- It is a technique used for predictive analysis in the fields of statistics, data mining, and machine learning.

- The CART algorithm is a type of classification algorithm that is required to build a decision tree on the basis of Gini's impurity index. One of the basic machine learning algorithms and provides a wide variety of use cases.





| Model Name | Training Score | Testing Score | Accuracy | F1 Score | Precision | Recall |
|---|---|---|---|---|---|---|
| Decision Tree Classifier - SMOTE | 1.000000 | 0.997520 | 0.997520 | 0.997520 | 0.996873 | 0.998177 |

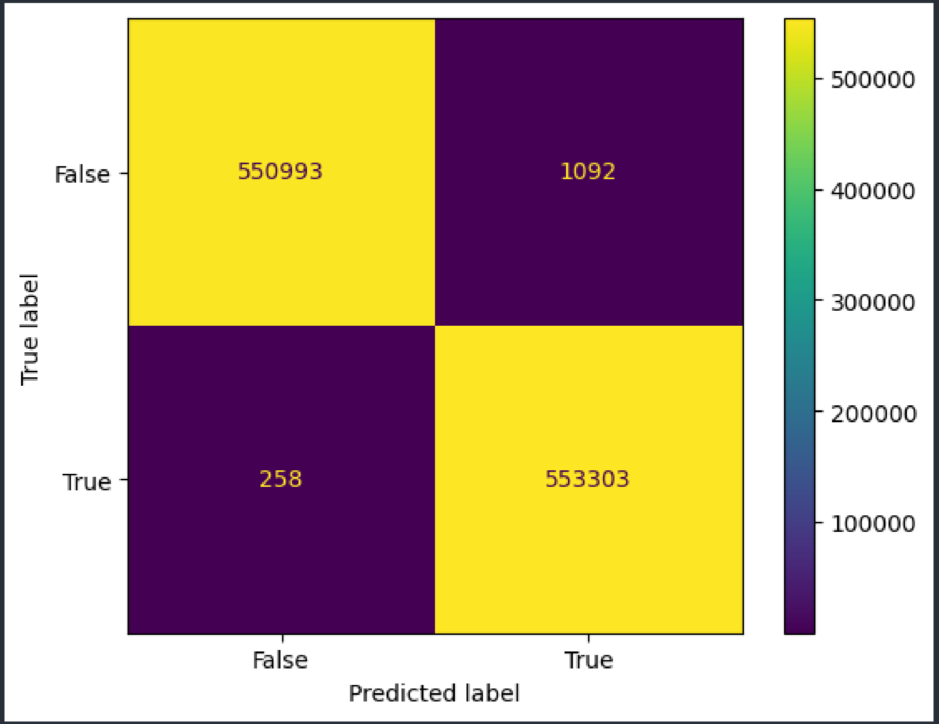
# 04 Machine Learning Models

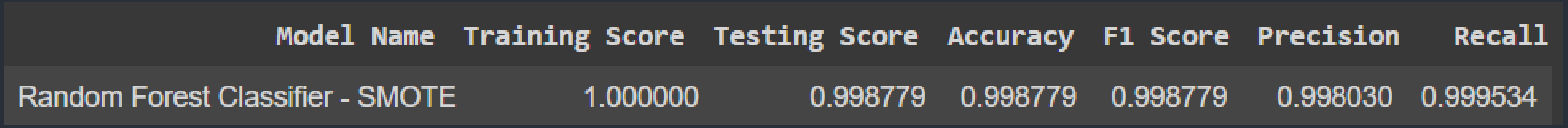
## Random Forest

- It is an ensemble technique capable of performing both regression and classification tasks with the use of multiple decision trees.

- The basic idea behind this is to combine multiple decision trees in determining the final output rather than relying on individual decision trees.

- Here, we randomly perform row sampling and feature sampling from the dataset forming sample datasets for every model.





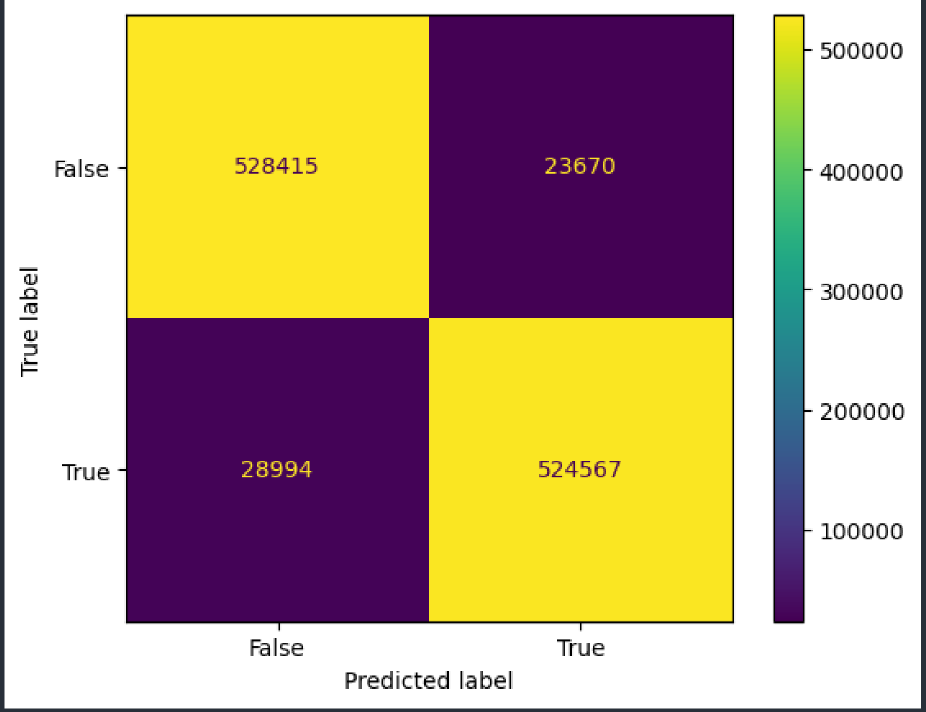
| Model Name | Training Score | Testing Score | Accuracy | F1 Score | Precision | Recall |
|---|---|---|---|---|---|---|
| Random Forest Classifier - SMOTE | 1.000000 | 0.998779 | 0.998779 | 0.998779 | 0.998030 | 0.999534 |

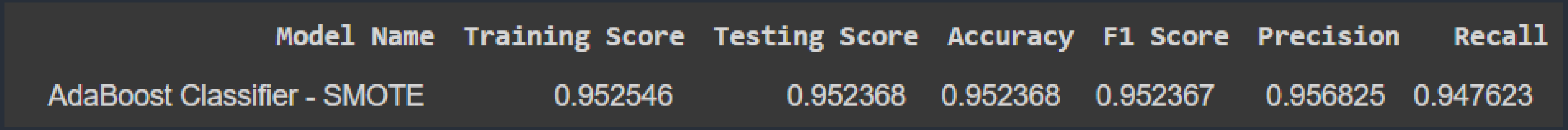
# 04 Machine Learning Models

## AdaBoost Classifier

- Adaptive Boosting algorithm is a boosting technique used as an Ensemble Method in ML. It is called Adaptive Boosting as the weights are re-assigned to each instance, with higher weights assigned to incorrectly classified instances.

- Boosting is used to reduce bias as well as variance for supervised learning. It works on the principle of learners growing sequentially. Except for the first, each subsequent learner is grown from previously grown learners.



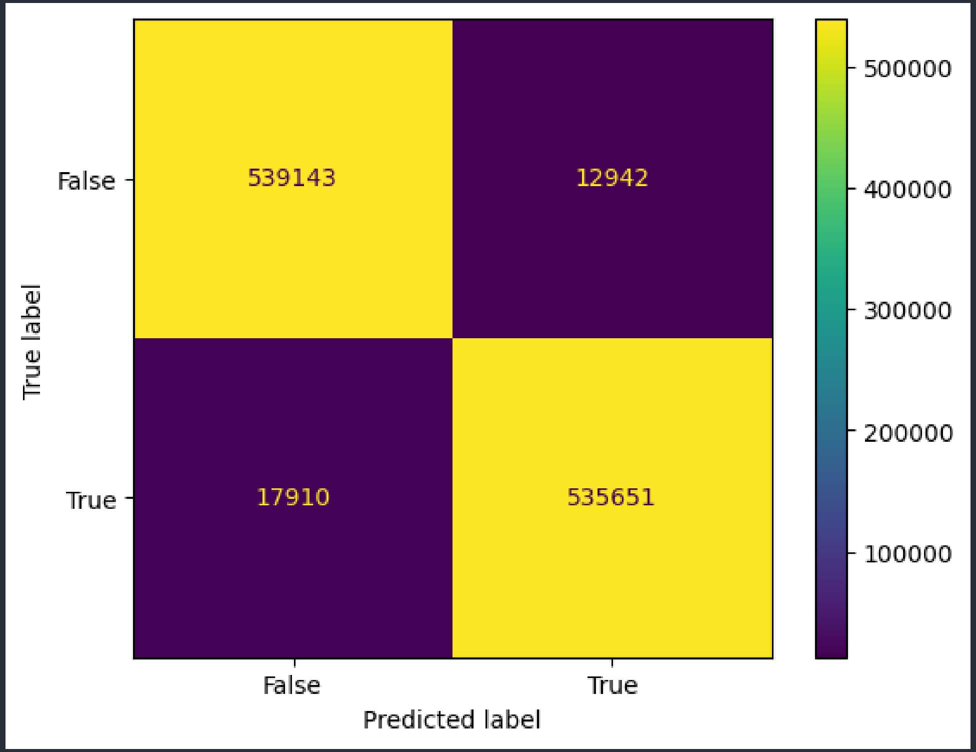


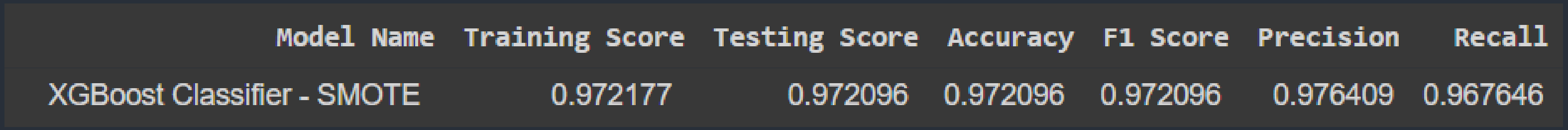
| Model Name | Training Score | Testing Score | Accuracy | F1 Score | Precision | Recall |
|---|---|---|---|---|---|---|
| AdaBoost Classifier - SMOTE | 0.952546 | 0.952368 | 0.952368 | 0.952367 | 0.956825 | 0.947623 |

# 04 Machine Learning Models

## XGBoost Classifier

- XGBoost stands for Extreme Gradient Boosting, which is a scalable, distributed gradient-boosted decision tree (GBDT) machine learning library.
- It provides parallel tree boosting and is the leading machine learning library for regression, classification, and ranking problems.
- It uses sequentially-built shallow decision trees to provide accurate results and a highly-scalable training method that avoids overfitting.



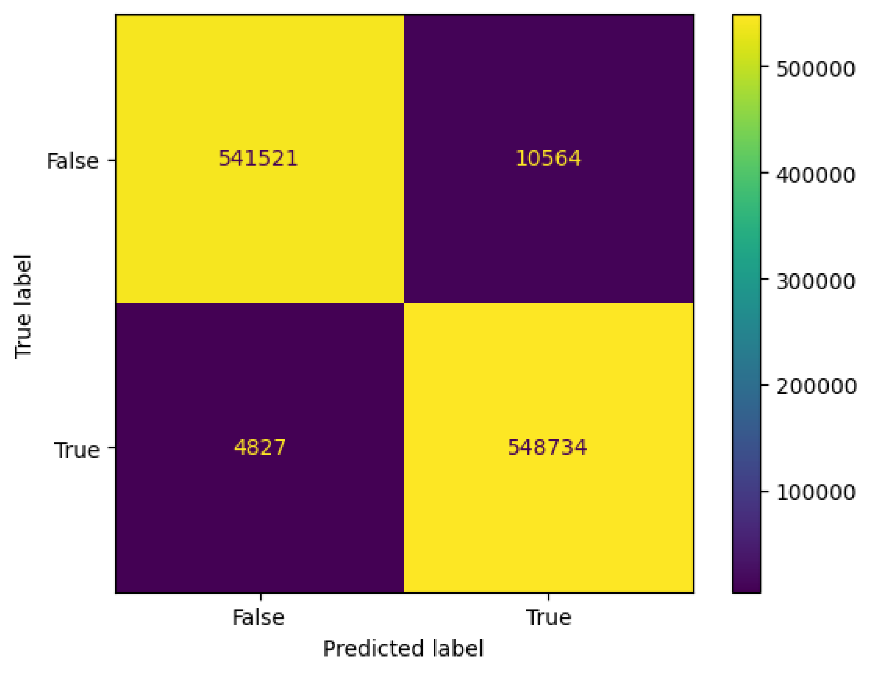


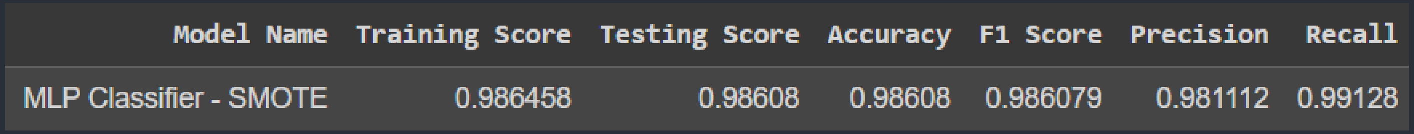
| Model Name | Training Score | Testing Score | Accuracy | F1 Score | Precision | Recall |
|---|---|---|---|---|---|---|
| XGBoost Classifier - SMOTE | 0.972177 | 0.972096 | 0.972096 | 0.972096 | 0.976409 | 0.967646 |

# 04 Machine Learning Models

## MLP Classifier

- The Multilayer Perceptron
- It  a feedforward artificial neural network model that maps input data sets to a set of appropriate outputs.
- An MLP consists of multiple layers and each layer is fully connected to the following one. Which means in this algorithm, data moves from the input to the output through layers in one (forward) direction.

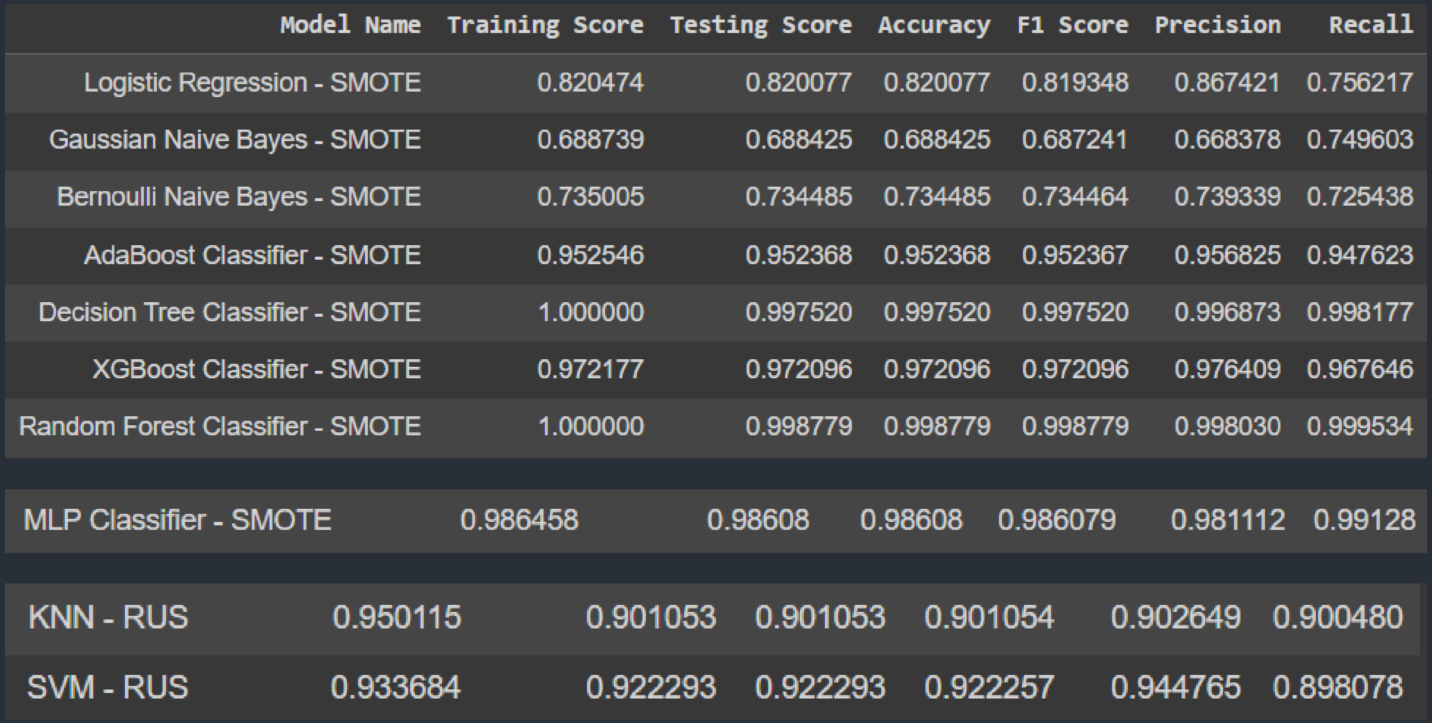




| Model Name | Training Score | Testing Score | Accuracy | F1 Score | Precision | Recall |
|---|---|---|---|---|---|---|
| MLP Classifier - SMOTE | 0.986458 | 0.98608 | 0.98608 | 0.986079 | 0.981112 | 0.99128 |

# Conclusion

Here, every model is being evaluated using many factors such as accuracy, precision, recall, F1 score etc. Looking at the figures it can be seen that Decision Tree and Random Forest lies in the top most efficient models for our data and definition.



| Model Name | Training Score | Testing Score | Accuracy | F1 Score | Precision | Recall |
|---|---|---|---|---|---|---|
| Logistic Regression - SMOTE | 0.820474 | 0.820077 | 0.820077 | 0.819348 | 0.867421 | 0.756217 |
| Gaussian Naive Bayes - SMOTE | 0.688739 | 0.688425 | 0.688425 | 0.687241 | 0.668378 | 0.749603 |
| Bernoulli Naive Bayes - SMOTE | 0.735005 | 0.734485 | 0.734485 | 0.734464 | 0.739339 | 0.725438 |
| AdaBoost Classifier - SMOTE | 0.952546 | 0.952368 | 0.952368 | 0.952367 | 0.956825 | 0.947623 |
| Decision Tree Classifier - SMOTE | 1.000000 | 0.997520 | 0.997520 | 0.997520 | 0.996873 | 0.998177 |
| XGBoost Classifier - SMOTE | 0.972177 | 0.972096 | 0.972096 | 0.972096 | 0.976409 | 0.967646 |
| Random Forest Classifier - SMOTE | 1.000000 | 0.998779 | 0.998779 | 0.998779 | 0.998030 | 0.999534 |
| MLP Classifier - SMOTE | 0.986458 | 0.98608 | 0.98608 | 0.986079 | 0.981112 | 0.99128 |
| KNN - RUS | 0.950115 | 0.901053 | 0.901053 | 0.901054 | 0.902649 | 0.900480 |
| SVM - RUS | 0.933684 | 0.922293 | 0.922293 | 0.922257 | 0.944765 | 0.898078 |

# Learning & Future Work

- Looking at the data veracity and its volume, which model/ml algorithm is to select is a very crucial decision. Moreover, cleaning and processing data according to the algorithm is more important and effort taking task then implementing a Machine Learning Model.
- Working on this project, we gain knowledge and some chief insights of the definition: credit card fraud detection.
- This project can be incorporated into online business like e-commerce, online banking where chances of fraudulent transaction is very high. And in the further extension we can try to fit more complex data within our model and to predict it accurately.

THANK YOU