**FINAL PROJECT**

BANK FRAUD DETECTION

Dhinesh Aaditya R C

Joby J

Karthikeyan J

Nishandhine U

Varshini A S

# **INTRODUCTION**

Bank fraud is a criminal act that involves stealing money or property from a bank through fraudulent means. Detecting and preventing bank fraud is crucial for financial institutions as it helps to maintain the trust of their customers, protect their assets, and ensure the stability of the financial system.

Bank fraud detection involves the use of various methods and technologies to identify fraudulent activities that are often difficult to detect through manual processes. Some common types of bank fraud include identity theft, credit card fraud, check fraud, and account takeover fraud.To detect bank fraud, financial institutions use various fraud detection systems that rely on artificial intelligence, machine learning, data analytics, and other advanced technologies. These systems analyze large volumes of data and detect patterns that may indicate fraudulent activities. They also use rules-based systems that flag suspicious transactions based on predefined criteria.

Bank fraud detection systems typically use a combination of both supervised and unsupervised learning techniques. Supervised learning involves training the system on labeled data, while unsupervised learning involves identifying patterns in unlabeled data. These techniques are used to identify anomalies, predict future fraudulent activities, and take preventive measures.

Overall, bank fraud detection is an essential aspect of the banking industry that helps to protect customers, prevent financial losses, and maintain the integrity of the financial system.

# **DATA DESCRIPTION**

The data for this project is used to evaluate the performance of the model.

The dataset contains 190001 observations and 36 variables,The variables in the dataset are:

| 1. **label** - Flag indicating whether the user paid back the credit amount within 5 days of issuing the loan {1:success, 0:failure} 2. **Msisdn** - mobile number of user 3. **Aon** - age on cellular network in days 4. **Daily\_decr30** - Daily amount spent from main account, averaged over last 30 days(in indonesian rupiah) 5. **Daily\_decr90** - Daily amount spent from main account, averaged over last 90 days(in indonesian rupiah) 6. **Rental30** - Average main account balance over last 30 days 7. **Rental90** - Average main account balance over last 90 days 8. **Last\_rech\_date\_ma** - Number of days till last recharge of main account 9. **Last\_rech\_date\_da** - Number of days till last recharge of data account 10. **Last\_rech\_amt\_ma** - Amount of last recharge of main account(in indonesian rupiah) 11. **Cnt\_ma\_rech30** - Number of times main account got recharged in last 30 days 12. **Fr\_ma\_rech30** - Frequency of main account recharged in last 30 days 13. **Sumamnt\_ma\_rech30** - Total amount of recharge in main account over last 30 days(in indonesian rupiah) 14. **Medianamnt\_ma\_rech30** - Median of amount of recharges done in main account over last 30 days at user level(in indonesian rupiah) 15. **Medianmarechprebal30** - Median of main account balance just before recharge in last 30 days at user level(in indonesian rupiah) 16. **Cnt\_ma\_rech90** - Number of times main account got recharged in last 90 days 17. **Fr\_ma\_rech90** - Frequency of main account recharged in last 90 days 18. **Sumamnt\_ma\_rech90** - Total amount of recharge in main account over last 90 days(in indonesian rupiah) 19. **Medianamnt\_ma\_rech90** - Median of amount of recharges done in main account over last 90 days at user level(in indonesian rupiah) 20. **Medianmarechprebal90** - Median of main account balance just before recharge in last 90 days at user level(in indonesian rupiah) 21. **Cnt\_da\_rech30** - Number of times data account got recharged in last 30 days 22. **fr\_da \_rech30** - Frequency of data account recharged in last 30 days 23. **Cnt\_da\_rech90** - Number of times data account got recharged in last 90 days 24. **fr\_da \_rech90** - Frequency of data account recharged in last 90 days 25. **Cnt\_loans30** - Number of loans taken by user in last 30 days 26. **Amnt\_loans30** - Total amount of loans taken by user in last 30 days 27. **Maxamnt\_loans30** - maximum amount of loan taken by the user in last 30 days 28. **Medianamnt\_loans30** - Median of amounts of loan taken by the user in last 30 days 29. **Cnt\_loans90** - Number of loans taken by user in last 90 days 30. **Amnt\_loans90** - Total amount of loans taken by user in last 90 days 31. **Maxamnt\_loans90** - maximum amount of loan taken by the user in last 90 days 32. **Medianamnt\_loans90** - Median of amounts of loan taken by the user in last 90 days 33. **Payback30** - Average payback time in days over last 30 days 34. **Payback90** - Average payback time in days over last 90 days 35. **pcircle** - telecom circle 36. **Pdate** - date   It is important to note that the data may contain missing values, outliers, or other anomalies that may need to be addressed during the data preprocessing stage. Additionally, feature engineering techniques such as scaling or normalization may be used to improve the performance of the machine learning algorithms. |
| --- |
|  |

# 

# **APPROACH**

1. We imported the dataset and created a profile report with the help of pandas-profiling.
2. We checked the dataset for missing and NA values. As there are no NA values, we further went on with the program .
3. After that with the help of the LabelEncoder we normalized the data in the dataset.
4. By splitting the training dataset into 80 and 20 randomly, We applied many algorithms to find which algorithm performs better.
5. With the help of KNeighborsClassifier, AdaBoostClassifier, RandomForestClassifier, GaussianNB we trained and predicted Y- label which indicated whether the user paid back the amount
6. And we calculated the accuracy score for the listed methods
7. By our observation RandomForestClassifier had the highest accuracy score of 91.77%
8. Then we merged RandomForestClassifier’s predicted Y to a new dataset “trained\_model”

# **VISUALIZATION**

Data visualization plays a crucial role in machine learning, as it helps in understanding the data and identifying patterns and trends. Data visualization is defined as a graphical representation that contains information and data, making it easier to understand and interpret the data.With the help of data visualization, we can quickly analyze how the data looks and what kind of correlation is held by the attributes of data .

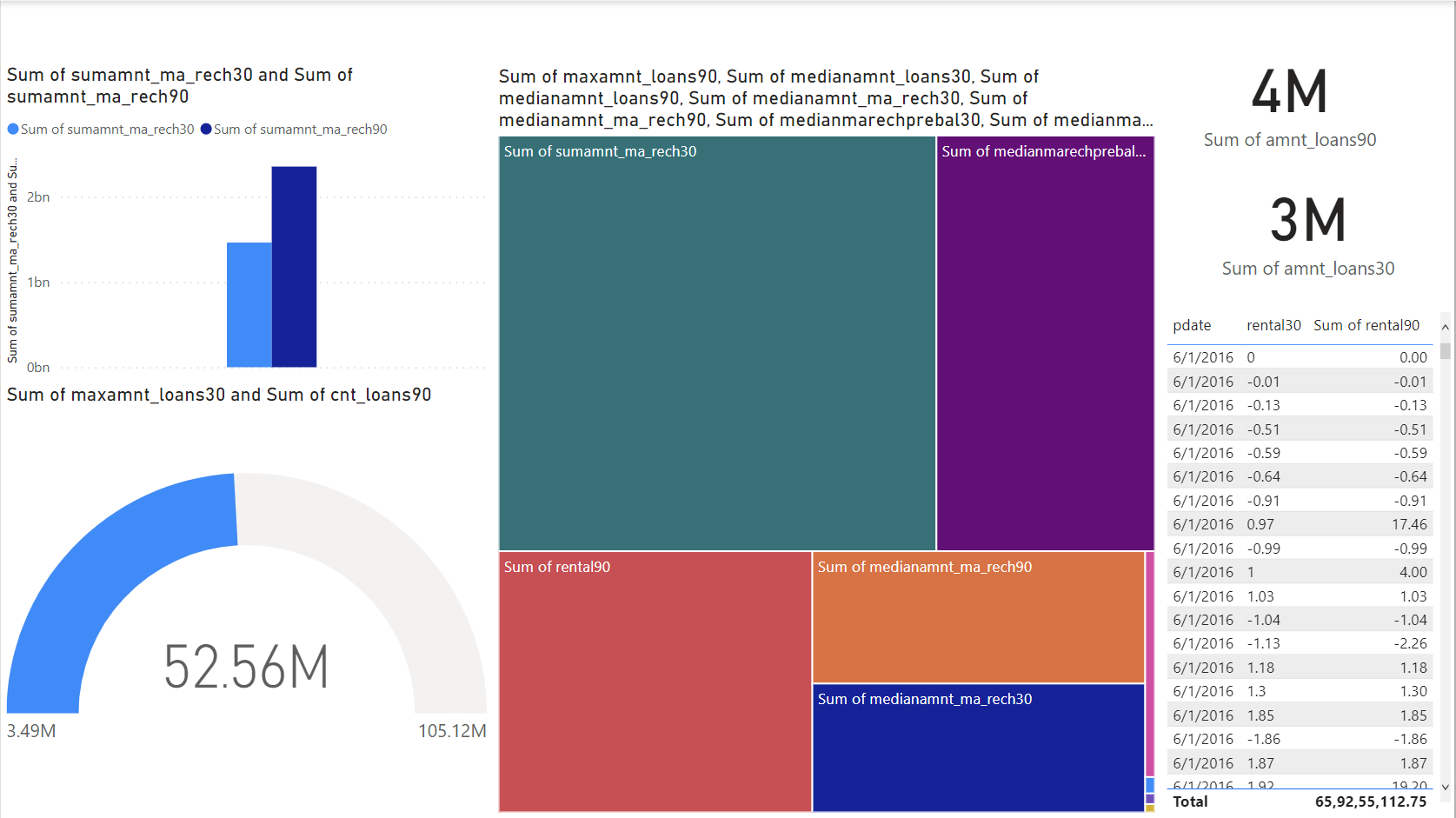
1. Scatter plots provide a representation of where each point in the entire dataset is present with respect to any two or three features.

2. By using visual elements like charts and graphs, machine learning practitioners can quickly and efficiently understand the underlying data and identify patterns that are not easily noticeable by simply looking at the raw data.

3. Identifying patterns and trends is one of the primary goals of machine learning, and data visualization is an essential tool to achieve this. Data visualization helps in pattern recognition, which is the process of identifying trends in the given pattern .

4. By analyzing the data through visualizations, machine learning practitioners can identify emerging trends and new opportunities that might not be visible otherwise.

5. Data visualizations written in languages like Python can help identify trends, patterns, and correlations that may have otherwise gone unnoticed



# 

# **ALGORITHMS**

We used the following algorithms in our project

* KNeighborsClassifier
* AdaBoost Classifier
* Random Forest Classifier
* Naive Bayes

## K Neighbors Classifier

K-Nearest Neighbors (KNN) is a simple yet powerful machine learning algorithm used for classification and regression tasks. The algorithm works by finding the k closest data points in the training set to a given test data point, and then using the labels of those neighbors to predict the label of the test point.

In classification, the most common label among the k neighbors is chosen as the predicted label for the test point. In regression, the algorithm predicts the average of the labels of the k nearest neighbors.KNN is a non-parametric algorithm, meaning it doesn't make any assumptions about the underlying distribution of the data. It's also a lazy algorithm, meaning it doesn't do any training or computation until it's given a new test data point to predict.

KNN has some advantages, such as its simplicity and ability to work with both numerical and categorical data. However, it can also be sensitive to outliers and noisy data, and it may not perform well on high-dimensional data.Overall, KNN is a useful algorithm in machine learning, and its effectiveness depends on the specific problem and dataset being used

## AdaBoost Classifier

AdaBoostClassifier is a machine learning algorithm used for classification tasks. It is an ensemble method that combines multiple weak classifiers into a single strong classifier.

The algorithm works by iteratively training a series of weak classifiers on the same dataset, with each subsequent classifier giving more weight to the misclassified data points from the previous classifier. The final classifier is a weighted combination of all the weak classifiers.AdaBoostClassifier can be used with any base classifier, such as decision trees, SVMs, or logistic regression. It is particularly effective with decision trees as the base classifier, creating a method known as AdaBoost Decision Trees (AdaBoostDT).

The algorithm has several advantages, including its ability to handle high-dimensional data and its tendency to avoid overfitting. It can also work well with imbalanced datasets.

However, AdaBoostClassifier can be sensitive to noisy data and outliers, and it may take longer to train than other algorithms.

Overall, AdaBoostClassifier is a powerful algorithm in machine learning and can be useful for a wide range of classification tasks. Its effectiveness depends on the specific problem and dataset being used.

## Random Forest Classifier

RandomForestClassifier is a machine learning algorithm used for classification tasks. It is an ensemble method that combines multiple decision trees into a single model.

The algorithm works by randomly selecting a subset of the features and a subset of the data samples, and then creating a decision tree on that subset. This process is repeated to create multiple decision trees. The final prediction is made by aggregating the predictions of all the individual decision trees.

RandomForestClassifier has several advantages, including its ability to handle high-dimensional data, its resistance to overfitting, and its robustness to noise and outliers. It can also work well with imbalanced datasets. The algorithm has a few hyperparameters that can be tuned for optimal performance, such as the number of trees and the maximum depth of each tree.

Overall, RandomForestClassifier is a powerful algorithm in machine learning and can be useful for a wide range of classification tasks. Its effectiveness depends on the specific problem and dataset being used, and it may require some hyperparameter tuning for optimal performance.

## Naive Bayes

GaussianNB is a machine learning algorithm used for classification tasks. It is a type of Naive Bayes algorithm that assumes that the features are normally distributed and that they are independent of each other.

The algorithm works by calculating the probability of each class for a given set of input features. It uses Bayes' theorem to calculate the conditional probability of each class given the input features, and then chooses the class with the highest probability as the predicted class.

GaussianNB has several advantages, including its simplicity, its ability to work with both numerical and categorical data, and its ability to handle high-dimensional data. It is also fast and requires minimal training data.

However, the assumption of independent features may not hold true for some datasets, which can lead to inaccurate predictions. It may also struggle with imbalanced datasets.

Overall, GaussianNB is a useful algorithm in machine learning and can be a good starting point for classification tasks. Its effectiveness depends on the specific problem and dataset being used, and it may require some preprocessing and feature engineering for optimal performance

# 

# 

# 

# **EVALUATION**

By using **accuracy\_score** and **f1\_score** from the sklearn library we draw the following evaluation. Hence we got the performance of each algorithm.

* Using KNN Algorithm :
  + Accuracy\_score: 88.81%
  + F1\_score : 93.84%
* Using AdaBoostClassifier Algorithm :
  + Accuracy\_score: 90.88%
  + F1\_score : 95.00%
* Using RandomForestClassifier Algorithm :
  + Accuracy\_score: 91.77%
  + F1\_score : 95.45%
* Using Naive Bayes Algorithm :
  + Accuracy\_score: 59.75%
  + F1\_score : 70.94%

From the above evaluation we conclude that **RandomForestClassifier** Algorithm have the highest accuracy score and stands out to be the most effective algorithm out of other algorithms

# **RESULT & DISCUSSION**

We created a machine learning model to identify fraud after analyzing the financial transaction data. We undersampled in an effort to improve the findings of the logistic regression, but a large amount of the data was left out, so the results remained the same. Through cross-validation, we verified that the models are not overfit. And then we executed the program with different algorithms. We can draw the conclusion that in this labeled dataset, fraud detection in financial transactions is effective, and the best algorithm for this purpose is Random Forest.

# **CONCLUSION**

People are constantly looking for methods, tools, or techniques that will make it easier for them to complete a job successfully. Algorithms used in machine learning are created in such a way that they attempt to learn on their own using prior knowledge. The algorithms can react and respond to circumstances for which they are not expressly programmed after learning from prior experience. Therefore, machine learning algorithms are very helpful in the discovery of fraud. It attempts to find obscure trends that aid in uncovering previously undetected fraud. Additionally, compared to conventional rule-based methods, its computation is quick

# 

# **FUTURE WORK**

There are several challenges that can arise when using machine learning for loan status prediction. Some of these challenges include:

1. **Bias in the data:** Machine learning models can be biased if the training data used to build the model is biased. For example, if the training data contains more data on one particular demographic group, then the model may be biased towards that group, leading to incorrect predictions for other groups.
2. **Overfitting:** Overfitting occurs when a model is too complex and fits too closely to the training data, resulting in poor performance on new data. This can be avoided by using techniques such as cross-validation and regularization.
3. **Missing or incorrect data:** Missing or incorrect data can lead to inaccurate predictions. It is important to ensure that the data used to train the model is complete and accurate.
4. **Interpretability:** Some machine learning algorithms, such as neural networks, can be difficult to interpret. This can be a problem when trying to understand why a particular loan application was approved or rejected.
5. **Changes in the underlying data:** Loan data can change over time due to changes in regulations, economic conditions, or borrower behavior. Machine learning models may need to be updated or retrained periodically to account for these changes.

It is important to be aware of these challenges when using machine learning for loan status prediction and to take steps to address them. This includes careful selection of the training data, use of appropriate machine learning algorithms, and ongoing monitoring and evaluation of model performance.

# **REFERENCE**

* <https://binariks.com/blog/financial-fraud-detection-machine-learning/>
* <https://github.com/awslabs/fraud-detection-using-machine-learning>
* <https://github.com/topics/fraud-detection>
* <https://scikit-learn.org/stable/>