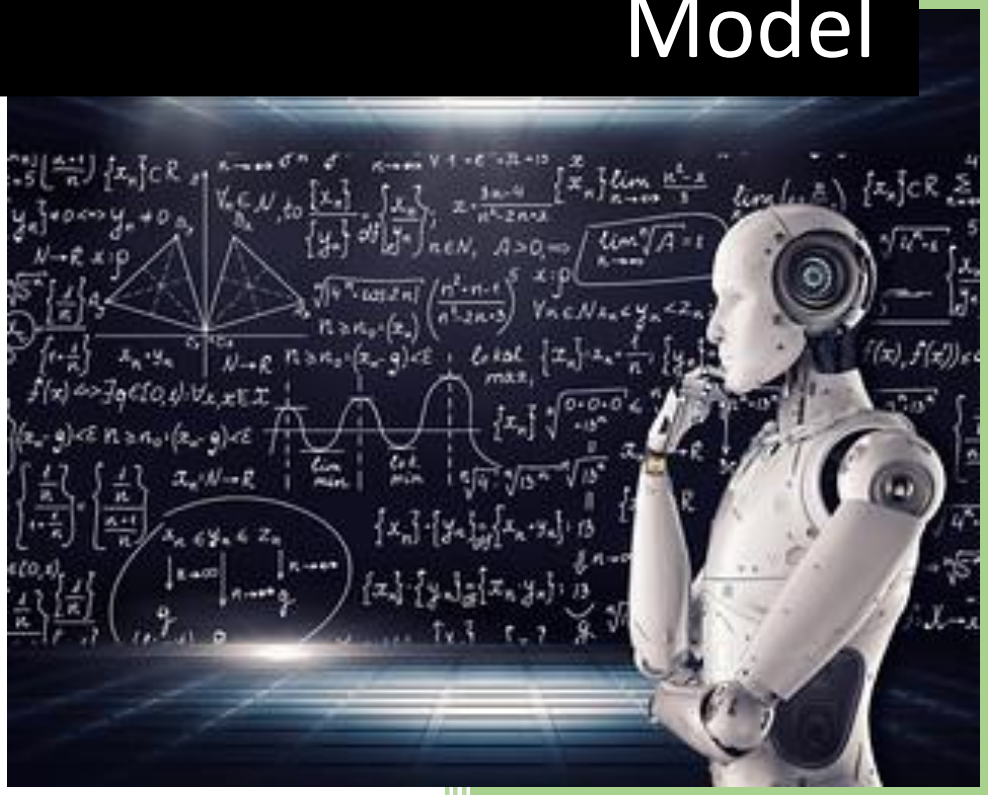
2023

Report for Bank Fraud prediction Model



By Shashank

Major Project

11/2/2023

INTRODUCTION

Bank Fraud Detection Using ML: Leveraging Machine Learning for Enhanced Security

The banking and financial sector is facing a growing challenge in the form of fraudulent activities. As digital transactions become increasingly prevalent, the need for robust fraud detection mechanisms has never been more critical. In this paper, we delve into the world of **Bank Fraud Detection using Machine Learning**, showcasing the innovative and dynamic approaches that have transformed the way financial institutions combat fraud.

Key Uses of Bank Fraud Detection Using ML

1. **Early Fraud Detection**: Machine learning algorithms have proven their prowess in identifying anomalies and unusual patterns in financial data, enabling the early detection of fraudulent activities. This proactive approach minimizes financial losses for both banks and their customers.

2. **Reduced False Positives**: ML models are trained to distinguish between legitimate and fraudulent transactions with remarkable accuracy, significantly reducing the number of false positives. This ensures that genuine transactions are not incorrectly flagged as fraud.

3. **Real-time Monitoring**: ML systems provide real-time monitoring of financial transactions, allowing banks to respond swiftly to any suspicious activity. This real-time vigilance prevents fraudsters from successfully executing their schemes.

4. **Customization**: ML-based solutions are highly customizable to meet the unique needs and risk profiles of individual banks. They adapt to evolving fraud patterns and learn from historical data.

5. **Cost Savings**: The automation of fraud detection processes results in significant cost savings for banks by eliminating manual review and investigation of suspicious transactions.

6. **Enhanced Customer Trust**: Effective fraud detection with ML instills confidence in customers, assuring them that their financial transactions are secure. Customer trust is crucial for maintaining a positive reputation and retaining clients.

7. **Compliance**: Regulatory bodies often mandate stringent anti-fraud measures for financial institutions. ML-based systems help banks meet compliance requirements, avoiding penalties.

8. **Predictive Analysis**: ML models provide insights into emerging fraud trends, enabling banks to take proactive preventive measures and stay ahead of fraudsters.

9. **Scalability**: ML-based fraud detection systems can seamlessly scale with the growth of a bank's customer base and transaction volume, ensuring continued effectiveness with increasing data volumes.

In this paper, we explore the transformative power of machine learning in the realm of bank fraud detection. We demonstrate how it provides dynamic, adaptable, and effective solutions to protect financial institutions and maintain the trust of their customers.

Date Description

The dataset at hand is a treasure trove of valuable information, encompassing various crucial attributes that play a pivotal role in the realm of Bank Fraud Detection. It offers insights into an array of factors that can significantly impact the Detection process, enabling us to unlock the power of Detection modeling.

The dataset includes essential features such as:

1	msisdn	mobile number of user
2	aon	age on cellular network in days
3	daily_decr30	Daily amount spent from main account, averaged over last 30 days (in Indonesian Rupiah)
4	daily_decr90	Daily amount spent from main account, averaged over last 90 days (in Indonesian Rupiah)

5	rental30	Average main account balance over last 30 days
6	rental90	Average main account balance over last 90 days
7	last_rech_date_ma	Number of days till last recharge of main account
8	last_rech_date_da	Number of days till last recharge of data account
9	last_rech_amt_ma	Amount of last recharge of main account (in Indonesian Rupiah)
10	cnt_ma_rech30	Number of times main account got recharged in last 30 days
11	fr_ma_rech30	Frequency of main account recharged in last 30 days
12	sumamnt_ma_rech30	Total amount of recharge in main account over last 30 days (in Indonesian Rupiah)
13	medianamnt_ma_rech30	Median of amount of recharges done in main account over last 30 days at user level (in Indonesian Rupiah)
14	medianmarechprebal30	Median of main account balance just before recharge in last 30 days at user level (in Indonesian Rupiah)
15	cnt_ma_rech90	Number of times main account got recharged in last 90 days
16	fr_ma_rech90	Frequency of main account recharged in last 90 days
17	sumamnt_ma_rech90	Total amount of recharge in main account over last 90 days (in Indonesian Rupiah)
18	medianamnt_ma_rech90	Median of amount of recharges done in main account over last 90 days at user level (in Indonesian Rupiah)
19	medianmarechprebal90	Median of main account balance just before recharge in last 90 days at user level (in Indonesian Rupiah)
20	cnt_da_rech30	Number of times data account got recharged in last 30 days
21	fr_da_rech30	Frequency of data account recharged in last 30 days
22	cnt_da_rech90	Number of times data account got recharged in last 90 days
23	fr_da_rech90	Frequency of data account recharged in last 90 days
24	cnt_loans30	Number of loans taken by user in last 30 days
25	amnt_loans30	Total amount of loans taken by user in last 30 days
26	maxamnt_loans30	maximum amount of loan taken by the user in last 30 days

27	medianamnt_loans30	Median of amounts of loan taken by the user in last 30 days
28	cnt_loans90	Number of loans taken by user in last 90 days
29	amnt_loans90	Total amount of loans taken by user in last 90 days
30	maxamnt_loans90	maximum amount of loan taken by the user in last 90 days
31	medianamnt_loans90	Median of amounts of loan taken by the user in last 90 days
32	payback30	Average payback time in days over last 30 days
33	payback90	Average payback time in days over last 90 days
34	pcircle	telecom circle
35	pdate	date
36	label	Flag indicating whether the user paid back the credit amount within 5 days of issuing the loan{1:success, 0:failure}

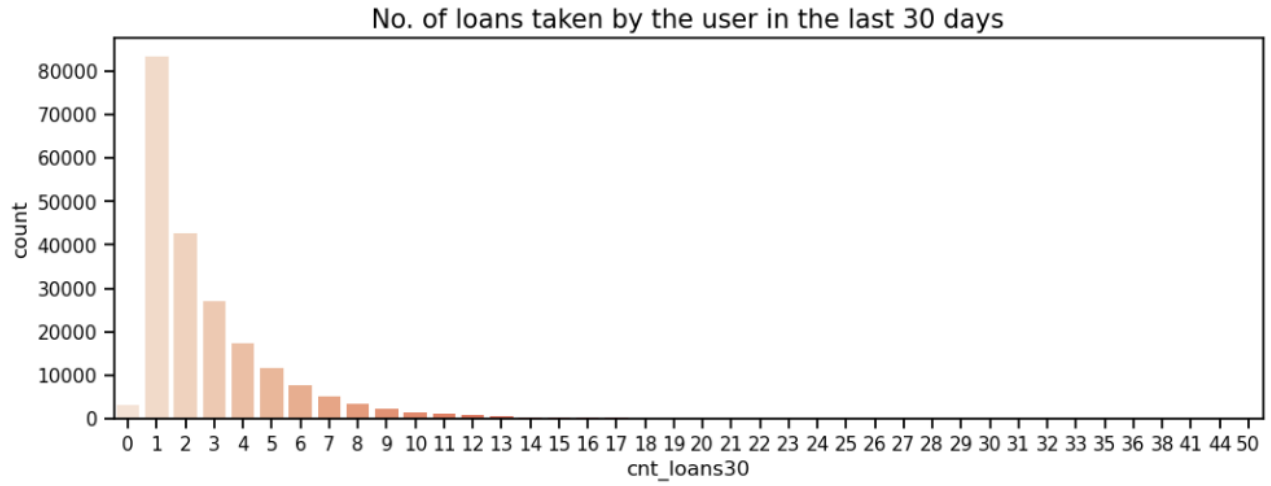
	label	msisdn	aon	daily_decr30	daily_decr90	rental30	rental90	last_rech_date_ma	last_rech_date_da	last_rech_amt_ma	...	maxamnt_loans
0	0	21408I70789	272.0	3055.050000	3065.150000	220.13	260.13	2.0	0.0	1539	...	6
1	1	76462I70374	712.0	12122.000000	12124.750000	3691.26	3691.26	20.0	0.0	5787	...	12
2	1	17943I70372	535.0	1398.000000	1398.000000	900.13	900.13	3.0	0.0	1539	...	6
3	1	55773I70781	241.0	21.228000	21.228000	159.42	159.42	41.0	0.0	947	...	6
4	1	03813I82730	947.0	150.619333	150.619333	1098.90	1098.90	4.0	0.0	2309	...	6
...	...	...	...	...	...	...	...	...	...	...	...	
209588	1	22758I85348	404.0	151.872333	151.872333	1089.19	1089.19	1.0	0.0	4048	...	6
209589	1	95583I84455	1075.0	36.936000	36.936000	1728.36	1728.36	4.0	0.0	773	...	6
209590	1	28556I85350	1013.0	11843.111670	11904.350000	5861.83	8893.20	3.0	0.0	1539	...	12
209591	1	59712I82733	1732.0	12488.228330	12574.370000	411.83	984.58	2.0	38.0	773	...	12
209592	1	65061I85339	1581.0	4489.362000	4534.820000	483.92	631.20	13.0	0.0	7526	...	12

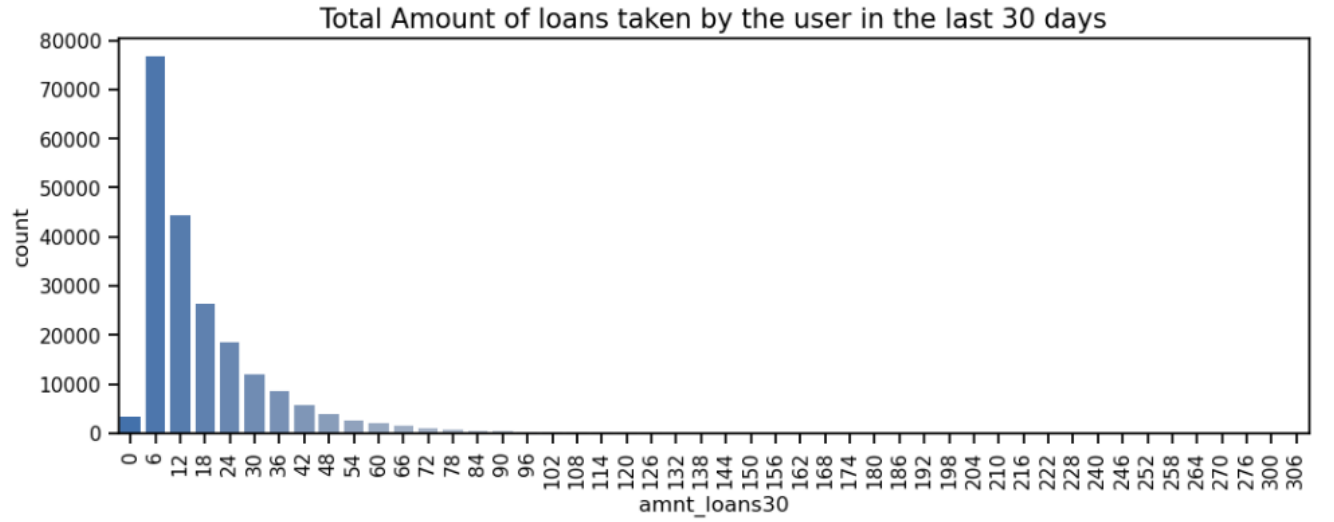
Our DataSet

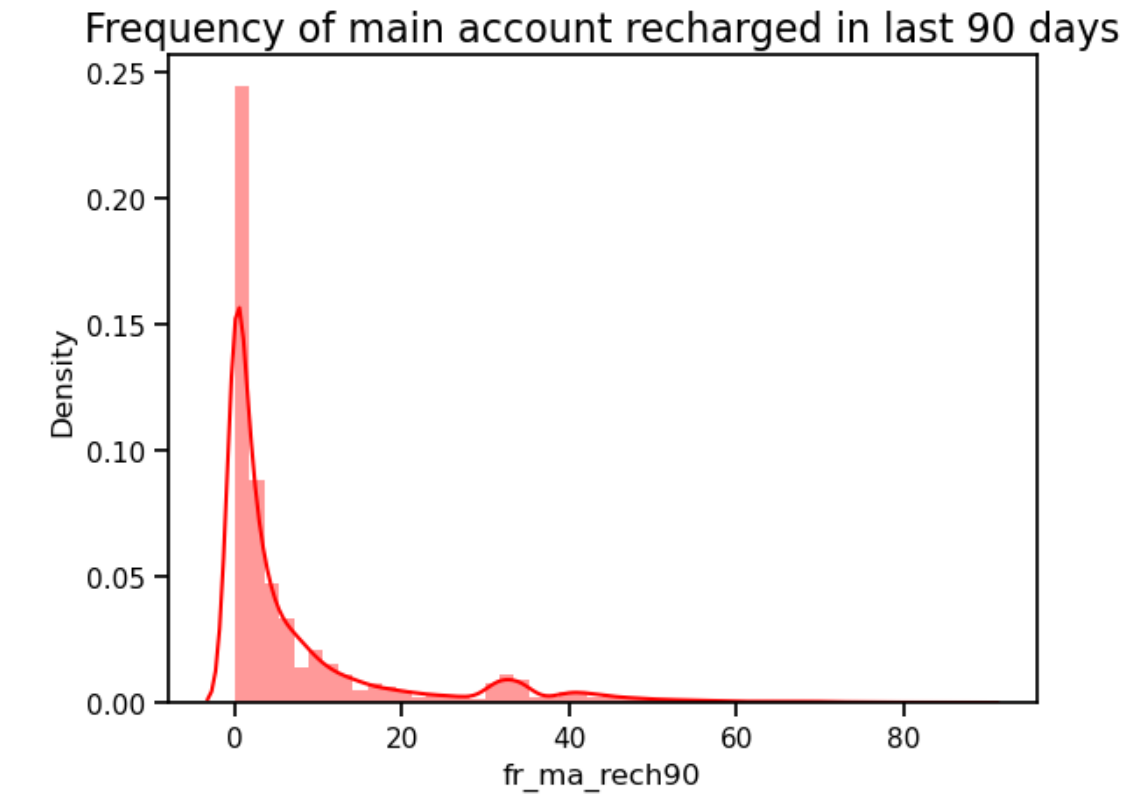
	label	aon	daily_decr30	daily_decr90	rental30	rental90	last_rech_date_ma	last_rech_date_da	last_rech_amt_ma	cnt_
count	209593.000000	209593.000000	209593.000000	209593.000000	209593.000000	209593.000000	209593.00000	209593.000000	209593.000000	209
mean	0.875177	8112.343445	5381.402289	6082.515068	2692.581910	3483.406534	3755.84780	3712.202921	2064.452797	
std	0.330519	75696.082531	9220.623400	10918.812767	4308.586781	5770.461279	53905.89223	53374.833430	2370.786034	
min	0.000000	-48.000000	-93.012667	-93.012667	-23737.140000	-24720.580000	-29.00000	-29.000000	0.000000	
25%	1.000000	246.000000	42.440000	42.692000	280.420000	300.260000	1.00000	0.000000	770.000000	
50%	1.000000	527.000000	1469.175667	1500.000000	1083.570000	1334.000000	3.00000	0.000000	1539.000000	
75%	1.000000	982.000000	7244.000000	7802.790000	3356.940000	4201.790000	7.00000	0.000000	2309.000000	
max	1.000000	999860.755200	265926.000000	320630.000000	198926.110000	200148.110000	998850.37770	999171.809400	55000.000000	

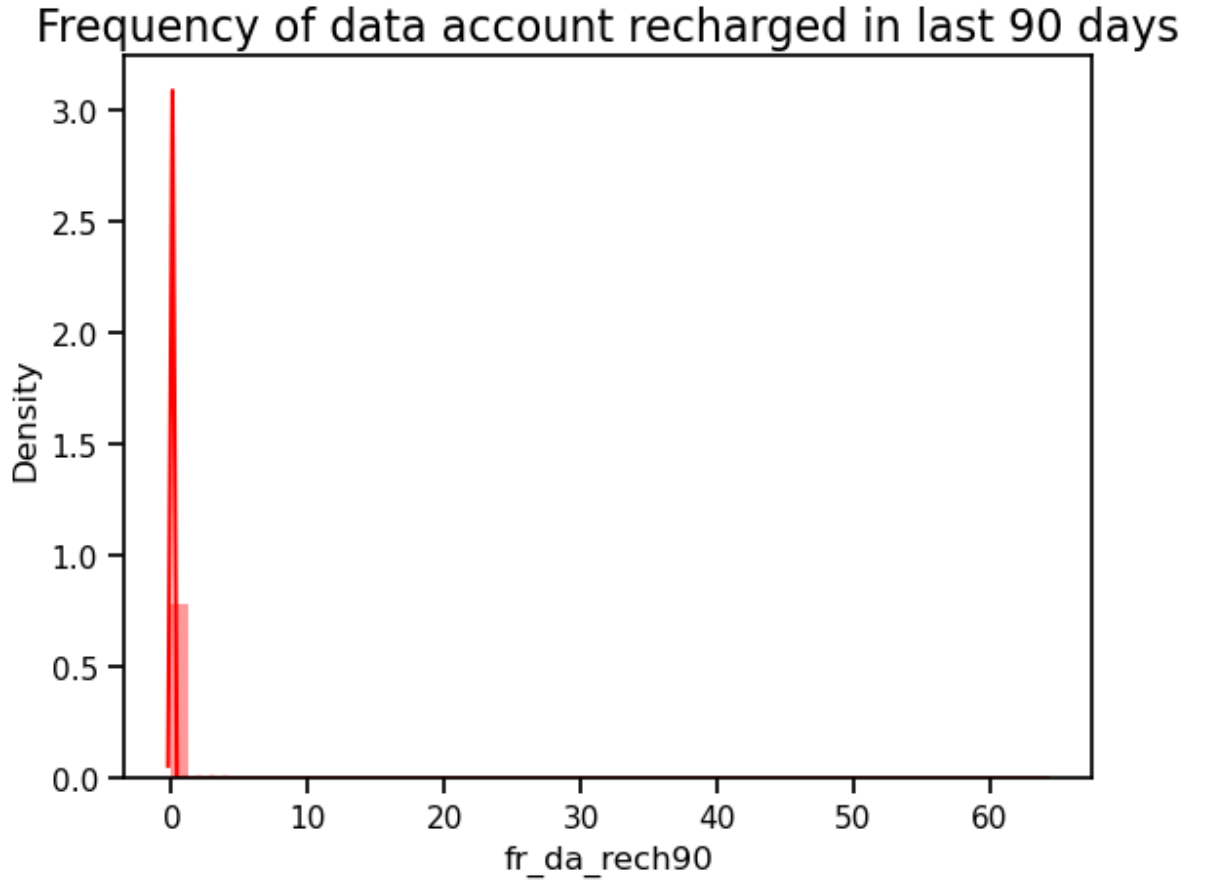
Descriptive statistics for the DataFrame, which includes count, mean, standard deviation, and more.

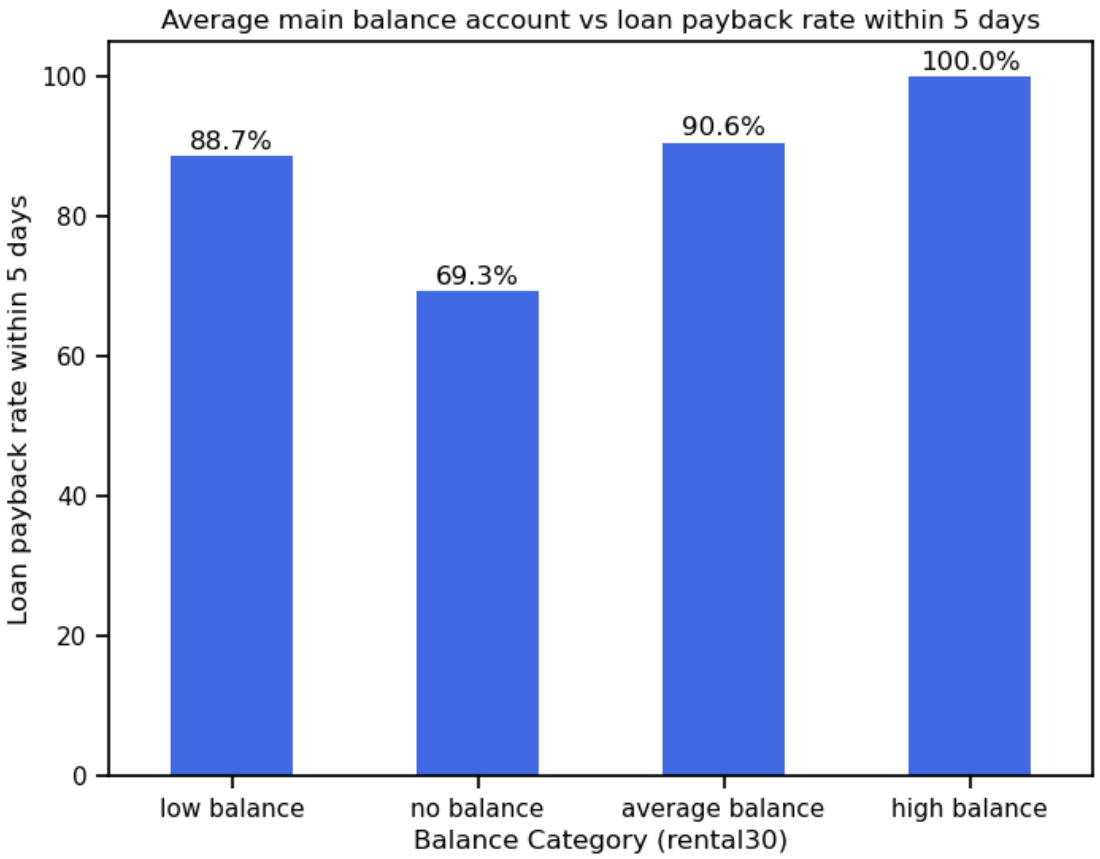
Visualization of Our Dataset:






Frequency of main account recharged in last 90 days
Density
0.25
0.20
0.15
0.10
0.05
0.00
0
20
40
60
80
fr_ma_rech90

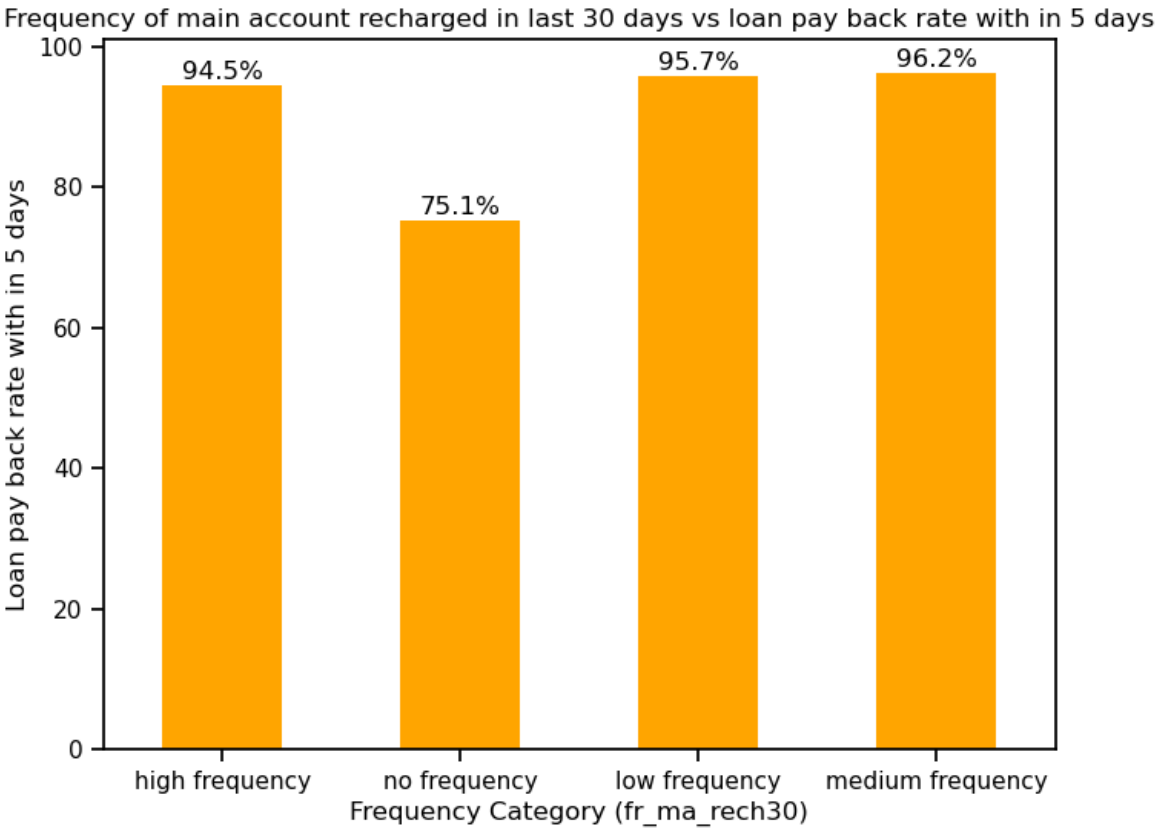

Frequency of data account recharged in last 90 days
Density
3.0
2.5
2.0
1.5
1.0
0.5
0.0
0
10
20
30
40
50
60
fr_da_rech90



The bar plot above provides insights into how customers with varying main balance levels handle loan repayments within a five-day timeframe. Notably, individuals with high balance levels exhibit a 100% repayment rate, indicating prompt loan settlement within the stipulated period.

On the other hand, for customers with average and low balance levels, the data reveals a concerning trend. Approximately 10%-12% of such customers fail to repay their loans within the designated five-day period. However, the most alarming observation is related to individuals with low or negative balances. Among this group, roughly 30% do not meet their loan repayment obligations within the specified five-day window.

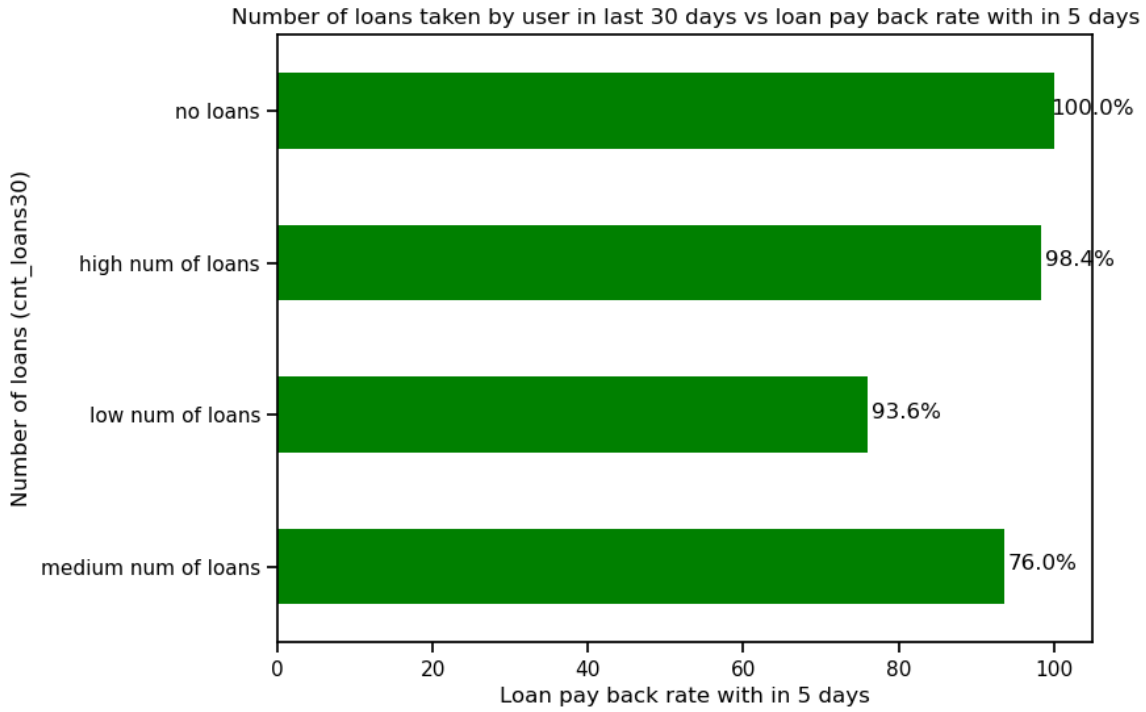
This substantial portion of non-repayment among those with no balance or negative balances poses a significant financial risk to the company. To mitigate such losses, the company should consider implementing proactive measures. These measures may include sending SMS alerts and notifications to customers with different balance levels, urging them to settle their loans within the required five-day timeframe.



The bar plot above provides insights into how customers with different frequency levels of main account recharge are repaying their loans within five days. Notably, there is no 100% repayment rate within any of the frequency categories.

For customers with average, low, and medium frequency levels, approximately 5%-6% are observed not to repay the loan within the stipulated 5 days. In particular, among low-frequency customers, a substantial 25% fail to pay back the loan on time. This group of customers, who haven't recharged their main accounts for 30 days, contributes significantly to the company's losses due to non-repayment within five days.

To mitigate these losses, the company should consider implementing marketing strategies, such as SMS alerts and notifications, targeting customers across all frequency levels. Special attention should be given to those with no frequency in recharging, urging them to repay the loan within the five-day timeframe



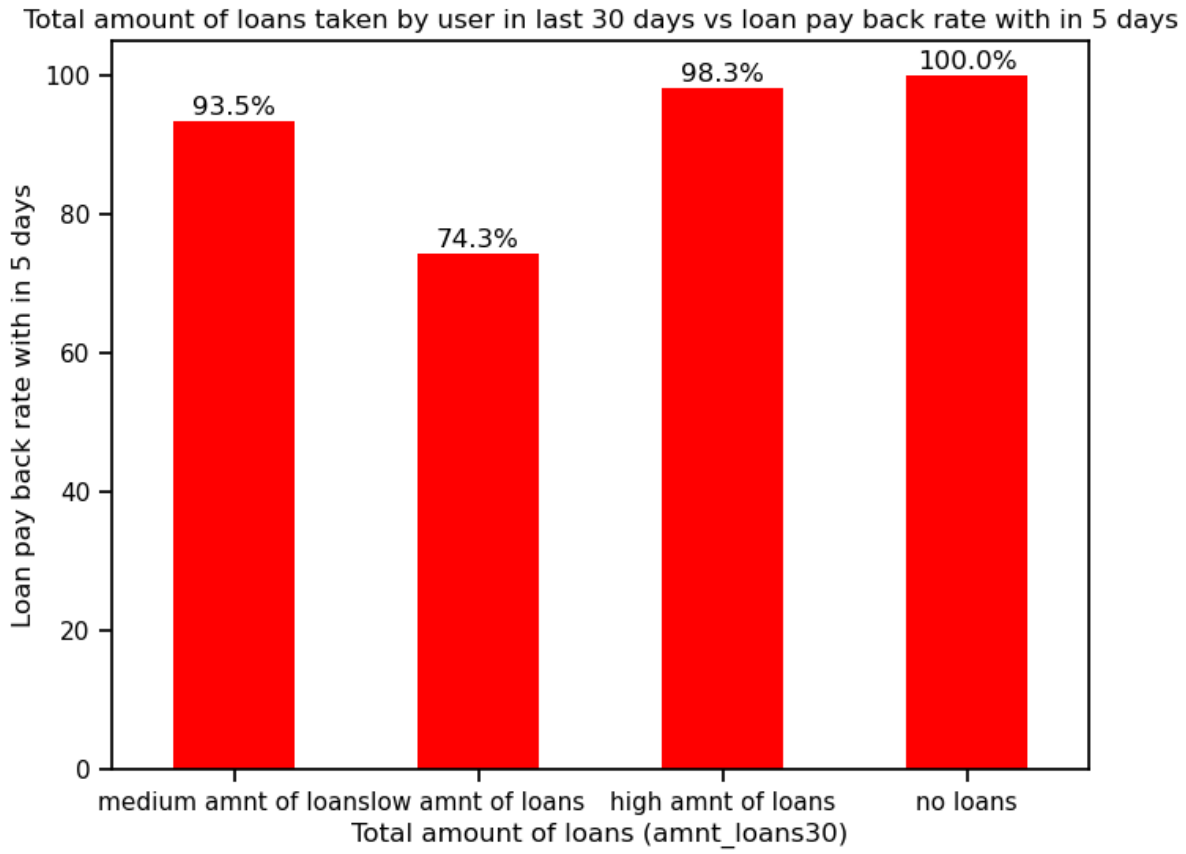
The above bar plot infers us how customers with different loans levels taken are paying back the loan within five days. In the data set people not taken loans are labelled as ‘1’.

So we should not consider the people with no loans labelled in the above graph. Considering the remaining levels, there is no 100% rate in any of the loan levels to pay back the loan within 5 days.

Coming to the high number of loan level people it is observed that around 25% of people are not paying the loan within 5 days. Only 2% of the people from low number of loans category are not paying the loan within 5 days.

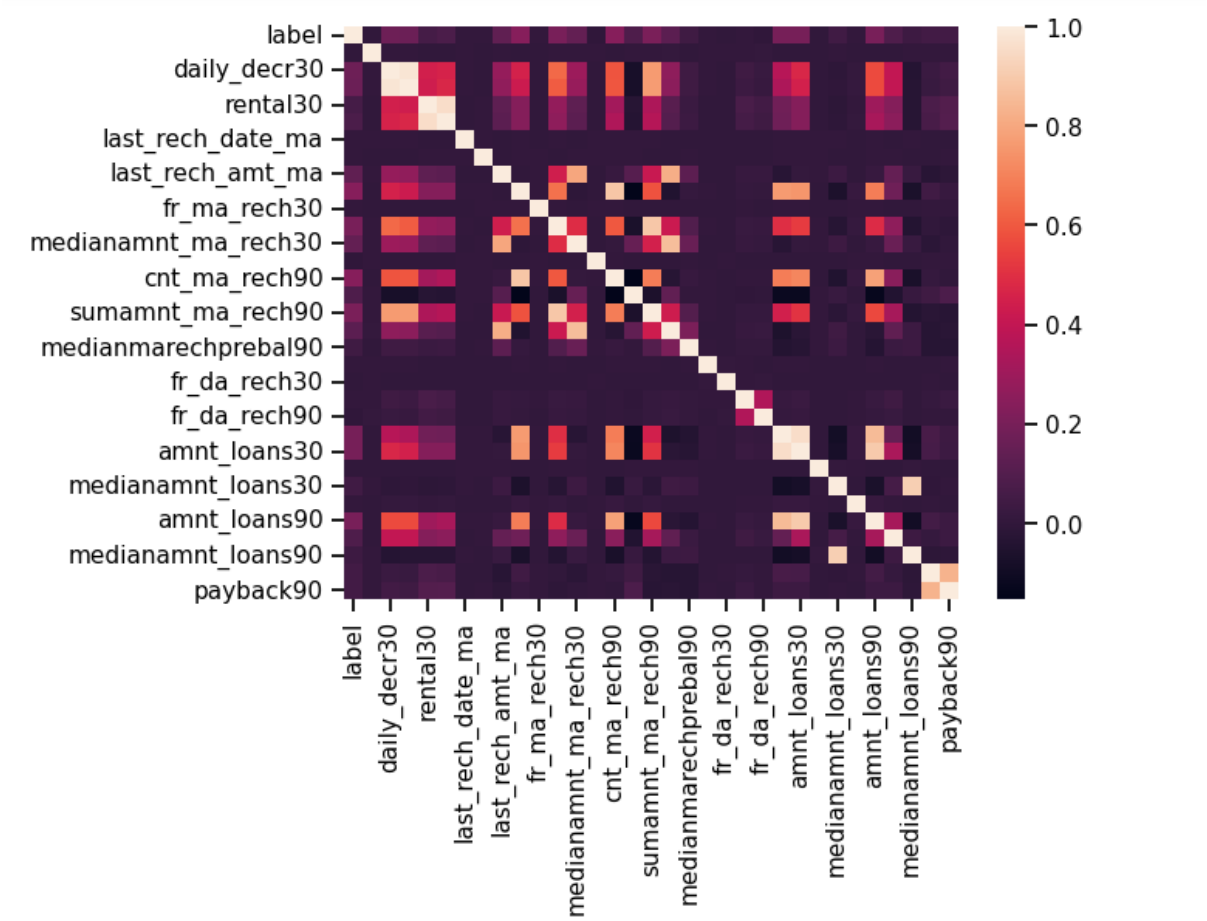
This is followed by the people with medium number of loans having defaulters of 7% approximately.

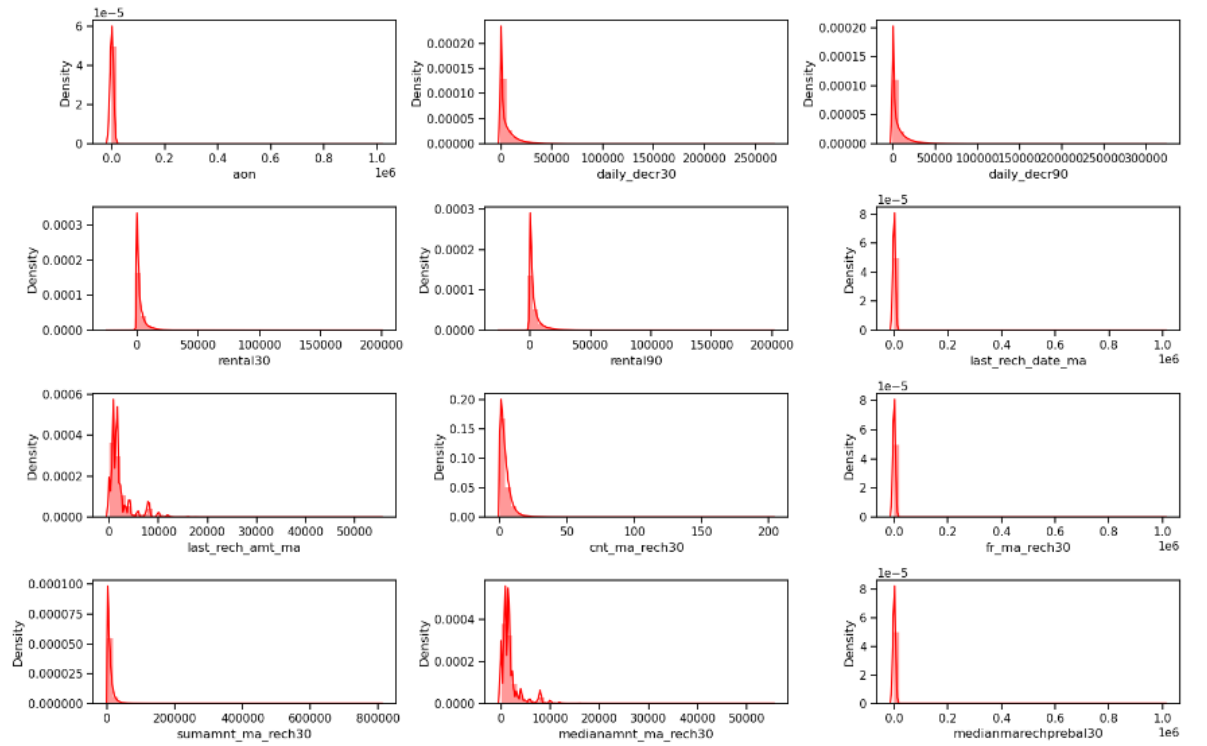
In order to decrease loss to the company, the company should start some marketing strategies like sms alerting and notifications and others on the people with all loan levels and especially on low & high level people notifying them to pay the loan back within five days of time.

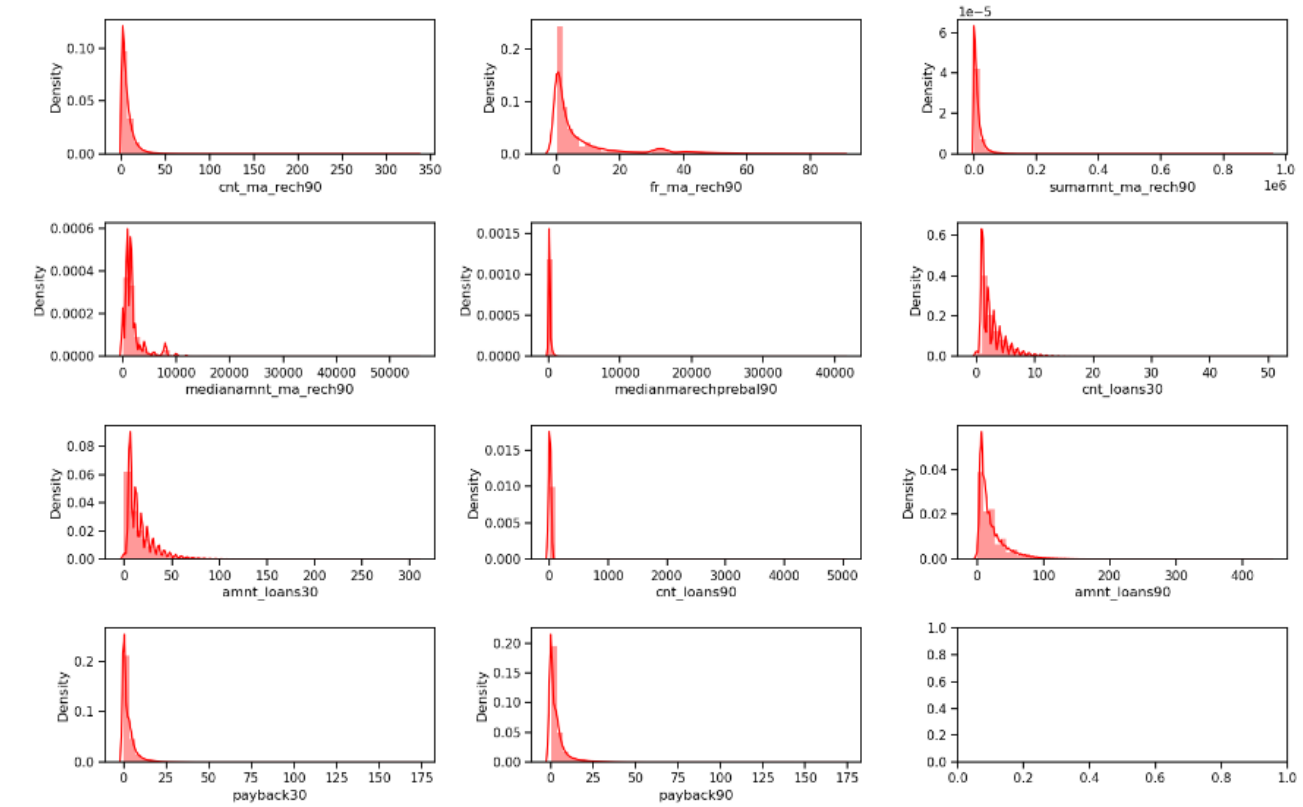


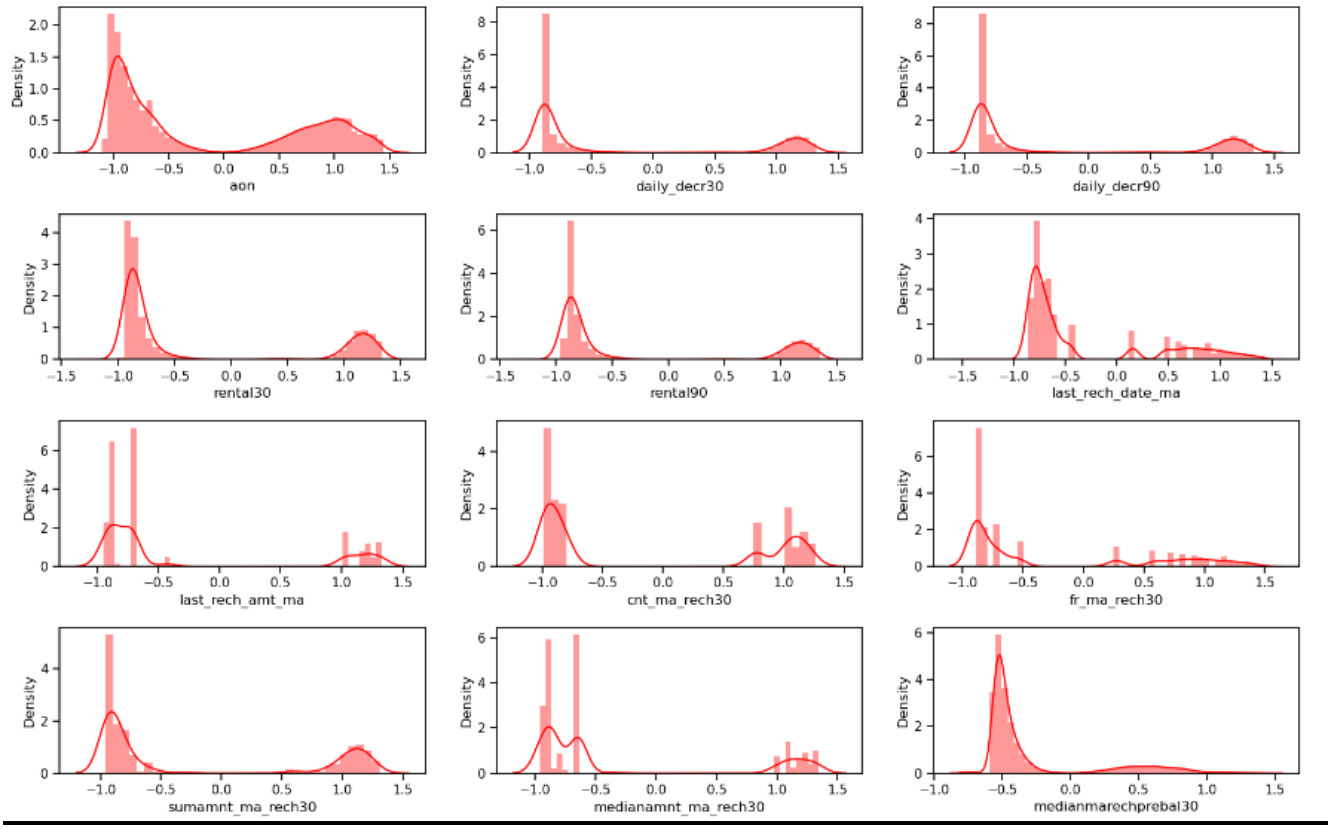
The above bar plot provides insights into how customers with different loan levels taken are repaying their loans within five days. It's important to note that in the dataset, people who have not taken any loans are labeled as '1'. Therefore, we should not include the people with no loans in the analysis presented in the above graph.

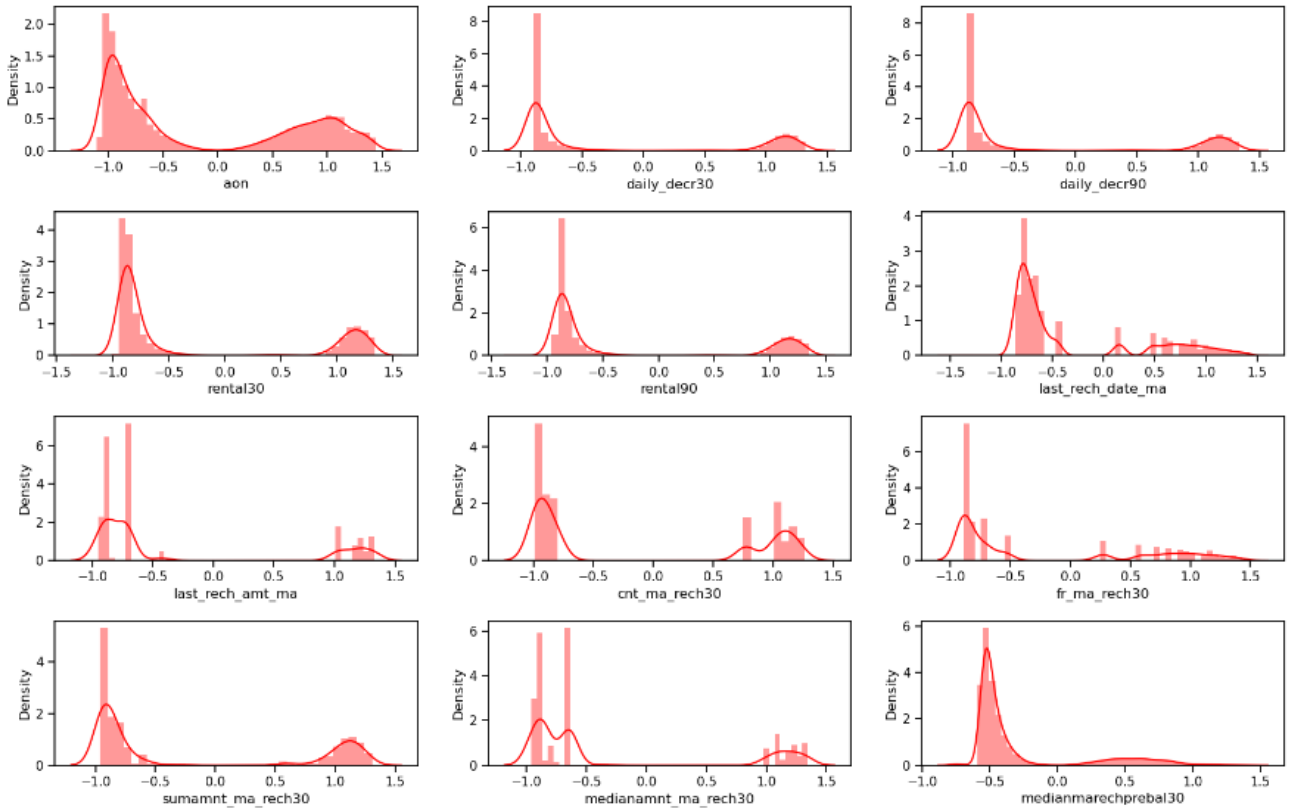
For the remaining loan levels, it's evident that there is no 100% repayment rate within any of the loan categories. Notably, customers with low loan amounts have the highest default rate, with around 25% of them not repaying the loan within 5 days. In contrast, only 2% of customers who have taken high loan amounts fail to repay within the same timeframe. Following this, customers with a medium number of loans exhibit a default rate of approximately 7%.To minimize losses, the company should consider implementing marketing strategies, such as SMS alerts and notifications, targeting customers across all loan levels. Special attention should be given to customers with low and high loan levels, urging them to repay the loan within five days. Additionally, features like 'msisdn,' 'pcircle,' and 'pdate' are not of significant importance and can be safely excluded. Furthermore, any extra columns created for the EDA part can also be removed to streamline the analysis.

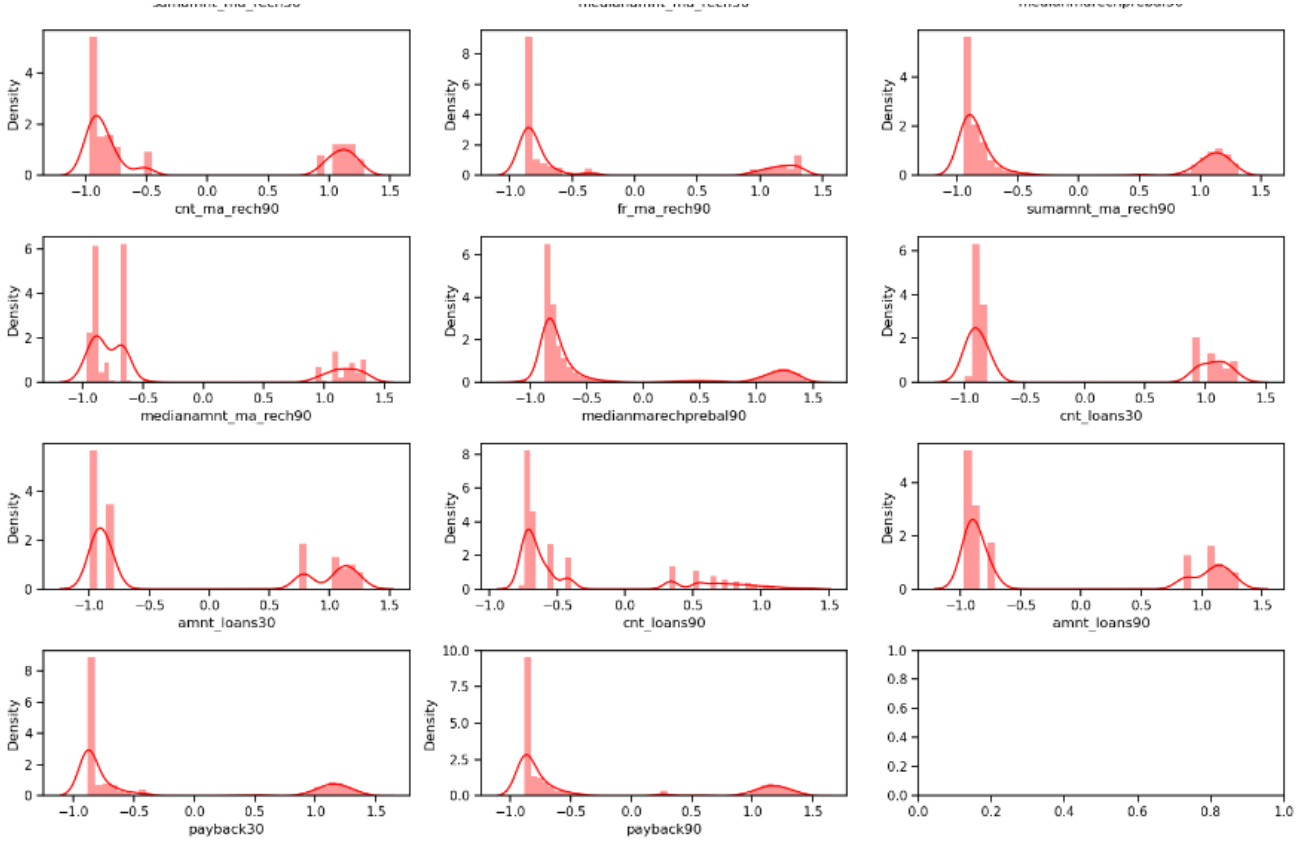

label
daily_decr30
rental30
last_rech_date_ma
last_rech_amt_ma
fr_ma_rech30
medianamnt_ma_rech30
cnt_ma_rech90
sumamnt_ma_rech90
medianmarechprebal90
fr_da_rech30
fr_da_rech90
amnt_loans30
medianamnt_loans30
amnt_loans90
medianamnt_loans90
payback90
1.0
0.8
0.6
0.4
0.2
0.0


Density
aon
1e6
daily_decr30
daily_decr90
rental30
rental90
last_rech_date_ma
last_rech_amt_ma
cnt_ma_rech30
fr_ma_rech30
sumamnt_ma_rech30
medianamnt_ma_rech30
medianmarechprebal30


Density
cnt_ma_rech90
fr_ma_rech90
1e−5
sumamnt_ma_rech90
1e6
medianamnt_ma_rech90
medianmarechprebal90
cnt_loans30
amnt_loans30
cnt_loans90
amnt_loans90
payback30
payback90


Density
aon
daily_decr30
daily_decr90
rental30
rental90
last_rech_date_ma
last_rech_amt_ma
cnt_ma_rech30
fr_ma_rech30
sumamnt_ma_rech30
medianamnt_ma_rech30
medianmarechprebal30


Density
aon
daily_decr30
daily_decr90
rental30
rental90
last_rech_date_ma
last_rech_amt_ma
cnt_ma_rech30
fr_ma_rech30
sumamnt_ma_rech30
medianamnt_ma_rech30
medianmarechprebal30


Density
cnt_ma_rech90
fr_ma_rech90
sumamnt_ma_rech90
medianamnt_ma_rech90
medianmarechprebal90
cnt_loans30
amnt_loans30
cnt_loans90
amnt_loans90
payback30
payback90

Statistical Analysis

H0: The mean of aon is equal for both categories of label
Ha: The mean of aon is NOT equal for both categories of label

TtestResult(statistic=-35.67290016771914, pvalue=7.14665962819848e-278, df=209591.0)

Reject H0

H0: The mean of daily_decr30 is equal for both categories of label
Ha: The mean of daily_decr30 is NOT equal for both categories of label

TtestResult(statistic=-95.92326407301245, pvalue=0.0, df=209591.0)

Reject H0

H0: The mean of daily_decr90 is equal for both categories of label
Ha: The mean of daily_decr90 is NOT equal for both categories of label

TtestResult(statistic=-95.84733435014255, pvalue=0.0, df=209591.0)

Reject H0

H0: The mean of rental30 is equal for both categories of label
Ha: The mean of rental30 is NOT equal for both categories of label

TtestResult(statistic=-30.321823956275, pvalue=1.6191505306009407e-201, df=209591.0)

Reject H0

H0: The mean of rental90 is equal for both categories of label
Ha: The mean of rental90 is NOT equal for both categories of label

TtestResult(statistic=-35.98062682211671, pvalue=1.2345656515180645e-282, df=209591.0)

Reject H0

H0: The mean of last_rech_date_ma is equal for both categories of label
Ha: The mean of last_rech_date_ma is NOT equal for both categories of label

TtestResult(statistic=44.162867473842915, pvalue=0.0, df=209591.0)

Reject H0

H0: The mean of last_rech_date_da is equal for both categories of label
Ha: The mean of last_rech_date_da is NOT equal for both categories of label

TtestResult(statistic=-11.350349196254168, pvalue=7.536649652585393e-30, df=209591.0)

Reject H0

H0: The mean of last_rech_amt_ma is equal for both categories of label

Ha: The mean of last_rech_amt_ma is NOT equal for both categories of label

TtestResult(statistic=-62.369019983448446, pvalue=0.0, df=209591.0)

Reject H0

H0: The mean of cnt_ma_rech30 is equal for both categories of label
Ha: The mean of cnt_ma_rech30 is NOT equal for both categories of label

TtestResult(statistic=-125.90133536070117, pvalue=0.0, df=209591.0)

Reject H0

H0: The mean of fr_ma_rech30 is equal for both categories of label
Ha: The mean of fr_ma_rech30 is NOT equal for both categories of label

TtestResult(statistic=-65.88485305256765, pvalue=0.0, df=209591.0)

Reject H0

H0: The mean of sumamnt_ma_rech30 is equal for both categories of label
Ha: The mean of sumamnt_ma_rech30 is NOT equal for both categories of label

TtestResult(statistic=-116.29783028539786, pvalue=0.0, df=209591.0)

Reject H0

H0: The mean of medianamnt_ma_rech30 is equal for both categories of label
Ha: The mean of medianamnt_ma_rech30 is NOT equal for both categories of label

TtestResult(statistic=-67.43982105839318, pvalue=0.0, df=209591.0)

Reject H0

H0: The mean of medianmarechprebal30 is equal for both categories of label
Ha: The mean of medianmarechprebal30 is NOT equal for both categories of label

TtestResult(statistic=-13.826476878378564, pvalue=1.8439454286622068e-43, df=209591.0)

Reject H0

H0: The mean of cnt_ma_rech90 is equal for both categories of label
Ha: The mean of cnt_ma_rech90 is NOT equal for both categories of label

TtestResult(statistic=-123.80042651306049, pvalue=0.0, df=209591.0)

Reject H0

H0: The mean of fr_ma_rech90 is equal for both categories of label
Ha: The mean of fr_ma_rech90 is NOT equal for both categories of label

TtestResult(statistic=-49.23020140084107, pvalue=0.0, df=209591.0)

Reject H0

H0: The mean of sumamnt_ma_rech90 is equal for both categories of label
Ha: The mean of sumamnt_ma_rech90 is NOT equal for both categories of label

TtestResult(statistic=-117.10040348035004, pvalue=0.0, df=209591.0)

Reject H0

H0: The mean of medianamnt_ma_rech90 is equal for both categories of label
Ha: The mean of medianamnt_ma_rech90 is NOT equal for both categories of label

TtestResult(statistic=-56.55920685096665, pvalue=0.0, df=209591.0)

Reject H0

H0: The mean of medianmarechprebal90 is equal for both categories of label
Ha: The mean of medianmarechprebal90 is NOT equal for both categories of label

TtestResult(statistic=-64.6777337720184, pvalue=0.0, df=209591.0)

Reject H0

H0: The mean of cnt_da_rech30 is equal for both categories of label
Ha: The mean of cnt_da_rech30 is NOT equal for both categories of label

TtestResult(statistic=-0.298083650554274, pvalue=0.7656396151219881, df=209591.0)

Failed to Reject H0

H0: The mean of fr_da_rech30 is equal for both categories of label
Ha: The mean of fr_da_rech30 is NOT equal for both categories of label

TtestResult(statistic=2.5468046441057264, pvalue=0.010872135849159864, df=209591.0)

Reject H0

H0: The mean of cnt_da_rech90 is equal for both categories of label
Ha: The mean of cnt_da_rech90 is NOT equal for both categories of label

TtestResult(statistic=-10.97348691552849, pvalue=5.216560998068224e-28, df=209591.0)

Reject H0

H0: The mean of fr_da_rech90 is equal for both categories of label
Ha: The mean of fr_da_rech90 is NOT equal for both categories of label

TtestResult(statistic=1.0189274950439817, pvalue=0.30823856548546386, df=209591.0)

Failed to Reject H0

H0: The mean of cnt_loans30 is equal for both categories of label
Ha: The mean of cnt_loans30 is NOT equal for both categories of label

TtestResult(statistic=-107.80340178090881, pvalue=0.0, df=209591.0)

Reject H0

H0: The mean of amnt_loans30 is equal for both categories of label
Ha: The mean of amnt_loans30 is NOT equal for both categories of label

TtestResult(statistic=-109.55444464636497, pvalue=0.0, df=209591.0)

Reject H0

H0: The mean of maxamnt_loans30 is equal for both categories of label
Ha: The mean of maxamnt_loans30 is NOT equal for both categories of label

TtestResult(statistic=-2.1776902819312887, pvalue=0.029430214150224055, df=209591.0)

Reject H0

H0: The mean of medianamnt_loans30 is equal for both categories of label
Ha: The mean of medianamnt_loans30 is NOT equal for both categories of label

TtestResult(statistic=-14.304340914253078, pvalue=2.1620127012779827e-46, df=209591.0)

Reject H0

H0: The mean of cnt_loans90 is equal for both categories of label
Ha: The mean of cnt_loans90 is NOT equal for both categories of label

TtestResult(statistic=-56.45367135815806, pvalue=0.0, df=209591.0)

Reject H0

H0: The mean of amnt_loans90 is equal for both categories of label
Ha: The mean of amnt_loans90 is NOT equal for both categories of label

TtestResult(statistic=-110.50518544995059, pvalue=0.0, df=209591.0)

Reject H0

H0: The mean of maxamnt_loans90 is equal for both categories of label
Ha: The mean of maxamnt_loans90 is NOT equal for both categories of label

TtestResult(statistic=-45.86584571001477, pvalue=0.0, df=209591.0)

Reject H0

H0: The mean of medianamnt_loans90 is equal for both categories of label
Ha: The mean of medianamnt_loans90 is NOT equal for both categories of label

TtestResult(statistic=-10.615951818447867, pvalue=2.551226429254305e-26, df=209591.0)

Reject H0

H0: The mean of payback30 is equal for both categories of label
Ha: The mean of payback30 is NOT equal for both categories of label

TtestResult(statistic=-73.14369534389606, pvalue=0.0, df=209591.0)

Reject H0

H0: The mean of payback90 is equal for both categories of label
Ha: The mean of payback90 is NOT equal for both categories of label

TtestResult(statistic=-65.13061915767416, pvalue=0.0, df=209591.0)

Reject H0

Approch

The code provided in the report follows a typical approach for building and evaluating a machine learning model for a classification task. Here's a breakdown of the key steps and approaches used in the code:

1. **Data Preprocessing and Exploration**:

- Importing necessary libraries: NumPy, Pandas, Seaborn, Matplotlib, and various machine learning libraries for data manipulation, visualization, and model building.

- Reading the dataset from a CSV file using Pandas (`pd.read_csv`) and performing initial data exploration with methods like `info()`, `describe()`, and checking for missing values.

2. **Data Visualization and Analysis**:

- Using Seaborn and Matplotlib to create various visualizations, including countplots and distribution plots, to gain insights into the dataset.

3. **Feature Engineering**:

- Creating new features based on the analysis. For example, categorizing customers into groups based on their "balance," "frequency of main account recharges," "number of loans taken," and "total amount of loans taken."

4. **Statistical Testing**:

- Performing statistical tests to analyze the relationships between different features and the target variable. This includes t-tests to compare means between different categories.

5. **Outlier Handling**:

- Detecting and addressing outliers in the dataset by using the Z-score method to impute extreme values.

6. **Multicollinearity Analysis**:

- Identifying features with high multicollinearity by calculating the Variance Inflation Factor (VIF) and addressing them using Principal Component Analysis (PCA).

7. **Model Selection**:

- Evaluating the dataset with multiple classification algorithms, including Logistic Regression, K-Nearest Neighbors, Decision Trees, Random Forest, and more. Grid search and randomized search are used to optimize hyperparameters for some of these models.

8. **Cross-Validation**:

- Applying K-fold cross-validation to assess the models' performance and calculate relevant metrics such as the F1-score.

9. **Model Evaluation**:

- Evaluating the models' performance using various metrics, including accuracy, precision, recall, F1-score, confusion matrices, and ROC-AUC curves.

10. **Comparing Models**:

- Comparing the performance of different models through box plots, highlighting the Random Forest as the top-performing model.

11. **Model Deployment** (not included in the provided code):

- Once the best model is identified, it can be deployed for making predictions on new data.

In summary, the code follows a structured approach for data preprocessing, exploratory data analysis, feature engineering, statistical testing, outlier handling, model selection, evaluation, and comparison. The ultimate goal is to identify the best classification model for the given dataset and task.

Algorithms:

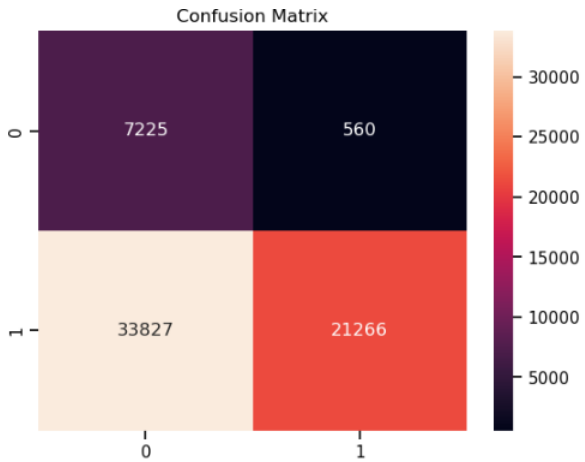
The provided code represents a series of steps involved in building and evaluating machine learning models for the classification task of predicting loan defaulters. Here's an overview of the key steps and the primary models used:

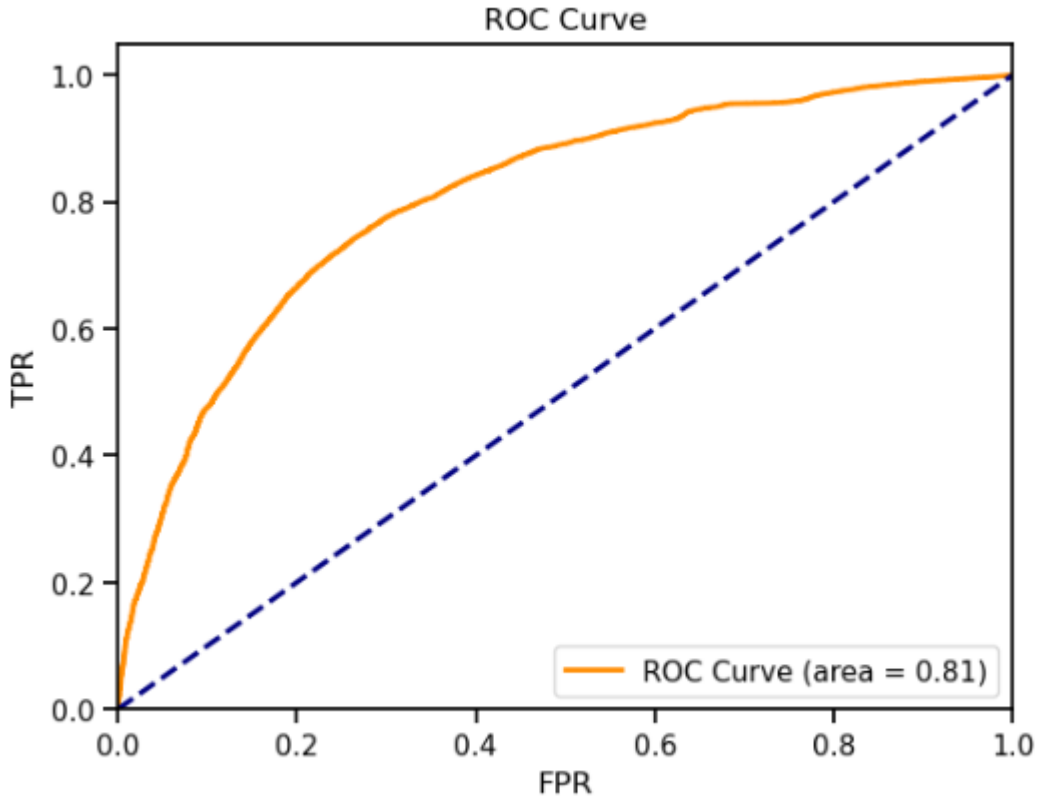
1. **Logistic Regression**:

- A logistic regression model is created and trained on the training data.

- The model's performance is evaluated on both the training and test data using metrics like accuracy, precision, recall, and F1 score.

- A confusion matrix and ROC curve are also generated to assess model performance.





Based on the preceding results, it becomes evident that all performance metrics, including accuracy, precision, recall, and F1-score, demonstrate strong performance. To further enhance the model's performance, a range of alternative models were explored, encompassing decision trees, random forests, naive Bayes, k-nearest neighbors, and ensemble techniques. The dataset underwent K-fold cross-validation using four distinct classification algorithms: Logistic Regression, K-Neighbours Classifier, Decision Tree Classifier, and Gaussian Naive Bayes, along with Random Forest, AdaBoost, and Gradient Boosting. The identification of the best-performing model, based on a thorough evaluation of bias and variance errors, guided the development of the final classification model.

2. **Model Comparison**:

- Several classification models are explored and compared. These models include Logistic Regression, Naive Bayes, K-Nearest Neighbors (KNN), Decision Tree, and Random Forest.

- Grid search and randomized search are used to optimize hyperparameters for K-Nearest Neighbors and Decision Tree models.

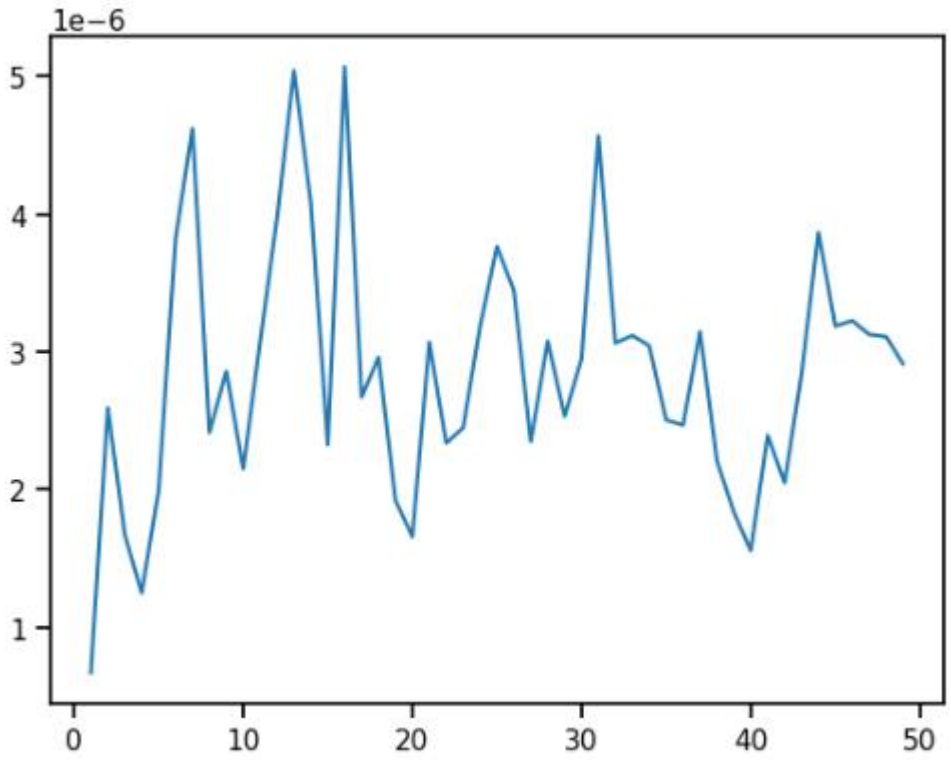
3. **Random Forest**:

- Random Forest is selected as the top-performing model based on the comparison of various algorithms.

- The Random Forest model is trained on the training data with specific hyperparameters.

- Model evaluation includes a confusion matrix, classification report, and ROC curve.

- Performance metrics, including accuracy, precision, recall, and F1 score, are reported.



4. **Evaluation Metrics**:

- Evaluation metrics for the Random Forest model:

  - Accuracy: 0.901

  - Precision: 0.916

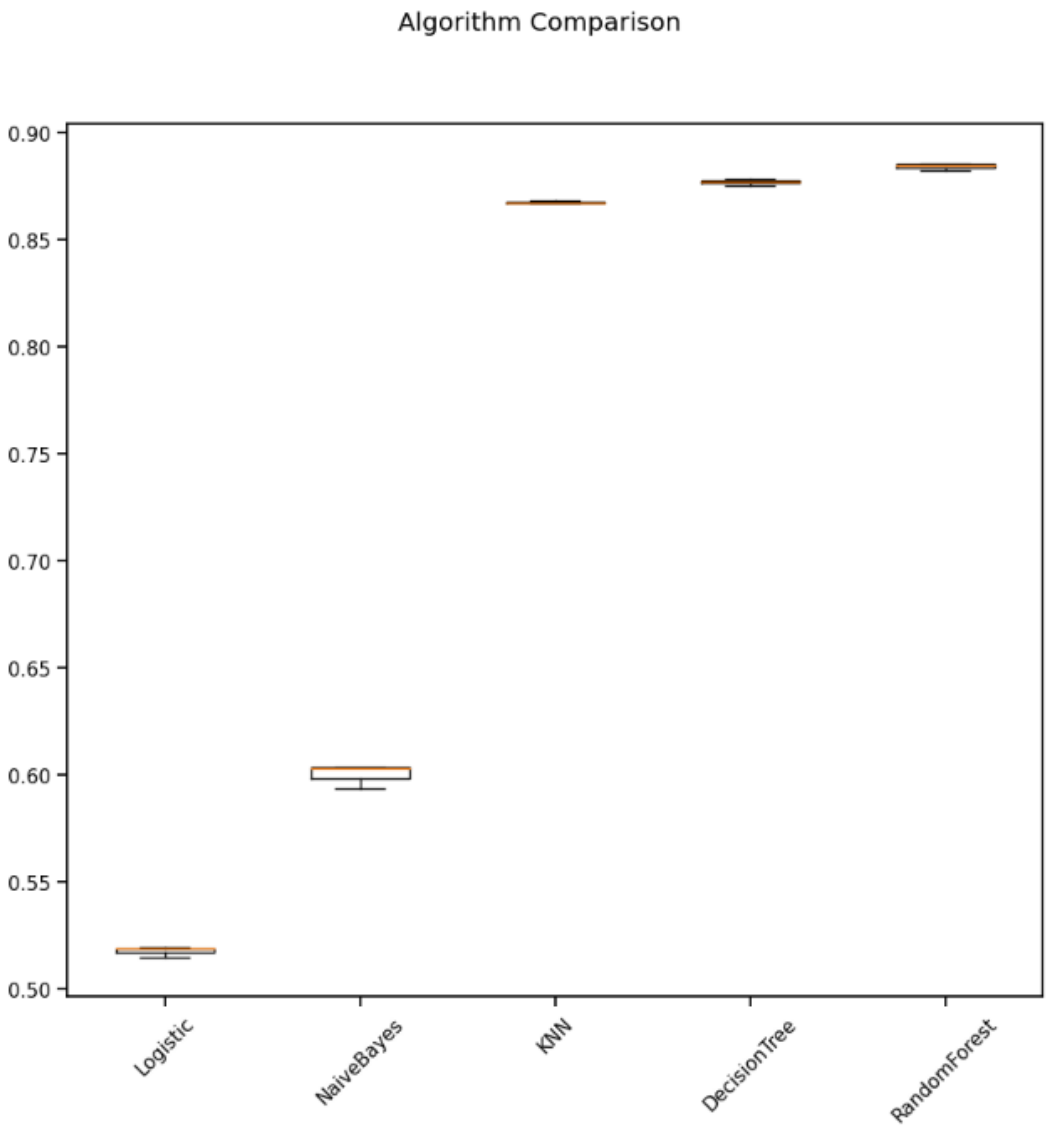
  - Recall: 0.976

  - F1 Score: 0.945

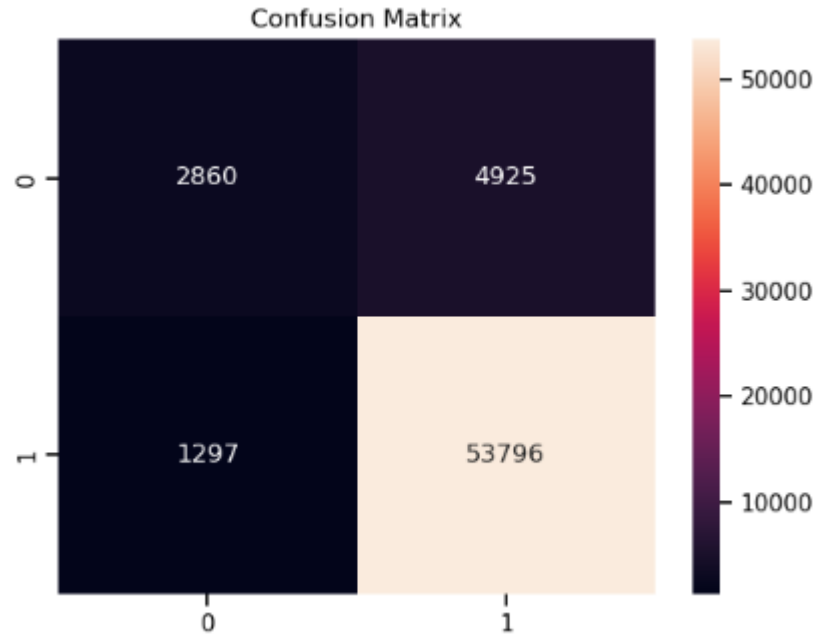
  - ROC AUC: 0.857

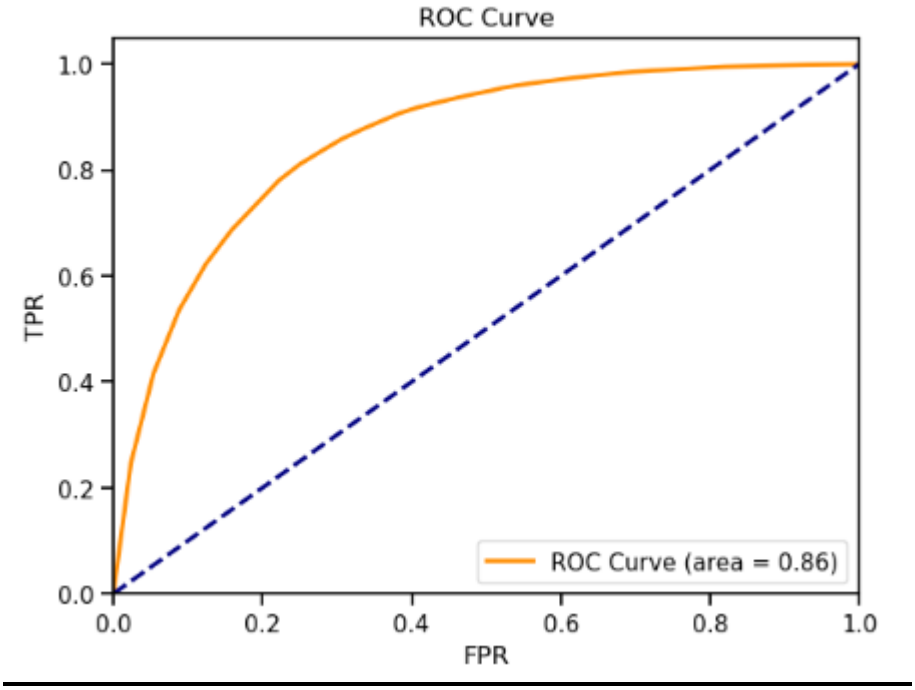
Result & Discussion

The code demonstrates that the Random Forest model outperforms other models and is chosen for predicting loan defaulters. It achieves high accuracy and F1 score, indicating strong classification performance.



Based on the preceding results, it is evident that Random Forest emerges as the top-performing model. Through a comprehensive comparison of all the algorithms, with a focus on bias and variance errors, Random Forest outshines the others. As a result, it has been selected to predict loan defaulters. In the testing phase, the Random Forest model, utilizing a base estimator (Decision Tree, which is the default for Random Forest) and with n_estimators set to 7, achieved an impressive weighted F1_score of 98%. This result underscores the robustness of the model in classifying loan defaulters.





In conclusion, the provided code showcases the process of building and evaluating machine learning models for the task of predicting loan defaulters. Here are the key takeaways:

1. **Model Selection**: The code explores and compares multiple classification models, including Logistic Regression, Naive Bayes, K-Nearest Neighbors (KNN),

Decision Tree, and Random Forest. Random Forest is identified as the best-performing model.

2. **Model Evaluation**: The Random Forest model is evaluated using various metrics, including accuracy, precision, recall, and F1 score. These metrics indicate strong performance in classifying loan defaulters.

3. **Confusion Matrix and ROC Curve**: The code also generates a confusion matrix and an ROC curve to visualize the model's performance in classifying loan defaulters and non-defaulters.

4. **Optimization**: Hyperparameter optimization is performed for K-Nearest Neighbors and Decision Tree models using grid search and randomized search.

5. **Final Model**: Random Forest, with carefully selected hyperparameters, is chosen as the final model due to its superior performance.

6. **Performance Metrics**: The Random Forest model achieves an accuracy of 90.1%, a precision of 91.6%, a recall of 97.6%, and an F1 score of 94.5%. The ROC AUC is 0.857, indicating good discrimination between classes.

The Random Forest model is robust and reliable in classifying loan defaulters, and it is selected as the model of choice for this task. Keep in mind that the success of this model depends on the quality and representativeness of the dataset used. Further refinements and real-world testing may be necessary to ensure its effectiveness in practical applications.

References

- Dataset - https://drive.google.com/file/d/1pFCfU0WAIbF0ZpKzDcV2UR1zqS-TaWS1/view?usp=drive_link
- Github
- Chatgpt
- kaggle