Micro-Credit Defaulter Model

MACHINE LEARNING

CLASSIFICATION PROBLEM

Submitted by:

RICHARD PRABHAKAR

ACKNOWLEDGMENT

This includes mentioning of all the references, research papers, data sources, professionals and other resources that helped you and guided you in completion of the project.

Links:-

https://stackoverflow.com/questions/72677752/create-x-train-and-y-train-for-csv-dataset-in-python

https://stackoverflow.com/questions/68794590/how-should-i-predict-target-variable-if-it-is-not-included-in-the-test-data-for

https://www.kaggle.com/datasets/nehalbirla/vehicle-dataset-from-cardekho

Medium.com- blogs on EDA

EDA by Analytics Vidya

https://www.kaggle.com/code/spscientist/a-simple-tutorial-on-exploratory-data-analysis

https://www.researchgate.net/publication/322394818_Predicting_Microfinance_Credit_Default_A_Study_of_Nsoatreman_Rural_Bank_Ghana

https://royalsocietypublishing.org/doi/10.1098/rsos.191649

https://towardsdatascience.com/credit-risk-modeling-with-machine-learning-8c8a2657b4c4

https://search.informit.org/doi/pdf/10.3316/ielapa.200903035

https://www.cloud4c.com/machine-learning-a-more-intelligent-route-to-microfinance-collection

https://finbraine.com/revolutionizing-microfinance-industry-with-machine-learning/

https://www.sciencedirect.com/science/article/pii/S1877050921009297

INTRODUCTION

- Business Problem Framing
    a. A Microfinance Institution (MFI) is an organization that offers financial services to low income populations. MFS becomes very useful when targeting especially the unbanked poor families living in remote areas with not much sources of income. The Microfinance services (MFS) provided by MFI are Group Loans, Agricultural Loans, Individual Business Loans and so on.
    b. Many microfinance institutions (MFI), experts and donors are supporting the idea of using mobile financial services (MFS) which they feel are more convenient and efficient, and cost saving, than the traditional high-touch model used since long for the purpose of delivering microfinance services. Though, the MFI industries is primarily focusing on low income families and are very useful in such areas, the implementation of MFS has been uneven with both significant challenges and successes.
    c. Today, microfinance is widely accepted as a poverty-reduction tool, representing $70 billion in outstanding loans and a global outreach of 200 million clients.
    d. We are working with one such client that is in Telecom Industry. They are a fixed wireless telecommunications network provider. They have launched various products and have developed its business and organization based on the budget operator model, offering better products at Lower Prices to all value conscious customers through a strategy of disruptive innovation that focuses on the subscriber.
    e. They understand the importance of communication and how it affects a person's life, thus, focusing on providing their services and products to low income families and poor customers that can help them in the need of hour.

f. They are collaborating with an MFI to provide micro-credit on mobile balances to be paid back in 5 days. The Consumer is believed to be defaulter if he deviates from the path of paying back the loaned amount within the time duration of 5 days. For the loan amount of 5 (in Indonesian Rupiah), payback amount should be 6 (in Indonesian Rupiah), while, for the loan amount of 10 (in Indonesian Rupiah), the payback amount should be 12 (in Indonesian Rupiah).

g. The sample data is provided to us from our client database. It is hereby given to you for this exercise. In order to improve the selection of customers for the credit, the client wants some predictions that could help them in further investment and improvement in selection of customers.

- Conceptual Background of the Domain Problem

Microfinance is a form of financial services provided to low-income individuals or households who are traditionally underserved by mainstream financial institutions. These services can include small loans, savings accounts, and other financial products. Predicting which microfinance clients are likely to default on their loans is an important problem for microfinance institutions, as default can lead to significant financial losses. Microfinance institutions use historical data on loan defaults and other relevant information to build statistical models which can then be used to make forecasts about future defaults. Machine learning techniques such as logistic regression, decision trees, and neural networks are commonly used to build these models. Additionally, a number of institutions also make use of additional data sources such as demographics, credit reports, and other financial information to build a dataset for their models. The problem is challenging as the data is often not complete, and the clients have limited financial history. Additionally, the model's performance is also highly dependent on the quality and quantity of the data used to train the model. Overall, predicting microfinance defaults is an important problem as it can help microfinance institutions better manage their risks and make more informed lending decisions. It can also help in identifying early warning signals of potential defaults which can help in taking preventative actions.

- Review of Literature

  When building a microfinance default prediction model, it is important to review the existing literature on the topic. This literature can provide valuable insights into the factors that influence microfinance defaults, as well as the methods that have been used to predict defaults in the past. Some key areas to focus on when reviewing the literature include:

  a. Factors that influence microfinance defaults: Researchers have identified a number of factors that can influence microfinance defaults, including clients' creditworthiness, income, and employment status. Additionally, external factors such as economic conditions and natural disasters can also have an impact on defaults.
  b. Methods for collecting microfinance data: In order to build a default prediction model, a large dataset of default information and other relevant information is needed. Researchers have used various methods to collect this data, such as survey, administrative records, and credit bureau data.
  c. Machine learning methods for default prediction: Many researchers have used machine learning techniques to predict microfinance defaults, including logistic regression, decision trees, and neural networks. Some studies have also used

ensemble methods such as Random Forest and Gradient Boosting to improve the performance of the model.

d. Evaluation metrics: Various evaluation metrics have been used to evaluate the performance of microfinance default prediction models, including accuracy, precision, recall, and F1 score.
e. Recent Advancement and Trends: The literature on the field is an ever-evolving one, and it's good to keep track of recent advancements and trends, such as the usage of deep learning methods, the integration of external factors such as weather, and the use of more sophisticated features engineering.
f. Ethical issues and Fairness: Microfinance default prediction models can have a significant impact on the lives of low-income individuals and households. Therefore, literature review should also consider the ethical issues and fairness implications of these models, such as the potential for bias and discrimination against certain groups of clients.
g. By reviewing the literature on microfinance default prediction, one can gain a better understanding of the factors that influence defaults, the methods that have been used to predict defaults in the past, and the evaluation metrics that have been used to assess the performance of different models. This knowledge can be used to inform the design and development of a microfinance default prediction model.

- Motivation for the Problem Undertaken

    a. Microfinance is a form of financial services provided to low-income individuals or households who are traditionally underserved by mainstream financial institutions. These services can include small loans, savings accounts, and other financial products. Predicting which microfinance clients are likely to default on their loans is an important problem for microfinance institutions, as default can lead to significant financial losses.

    b. There are a number of factors that can influence whether a microfinance client will default on their loan, including their creditworthiness, income, and employment status. Additionally, external factors such as economic conditions and natural disasters can also have an impact on loan defaults.

    c. To predict microfinance defaults, many institutions use historical data on loan defaults and other relevant information to build statistical models, which can then be used to make forecasts about future defaults. Machine learning techniques such as logistic regression, decision trees, and neural networks are commonly used to build these models. Additionally, a number of institutions also

make use of additional data sources such as demographics, credit reports, and other financial information to build a dataset for their models.

d. The problem of default prediction in microfinance is a challenging one as the data is often not complete, and the clients have limited financial history. Additionally, the model's performance is also highly dependent on the quality and quantity of the data used to train the model.

e. Overall, predicting microfinance defaults is an important problem as it can help microfinance institutions better manage their risks and make more informed lending decisions. It can also help in identifying early warning signals of potential defaults which can help in taking preventative actions.

Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem

  Classification is a type of machine learning that involves predicting a categorical value, such as a class label or a binary outcome. There are several mathematical, statistical, and analytical techniques that can be used in classification modeling, including:

  - Logistic regression: This is a type of regression that is used for classification tasks, where the goal is to predict a binary outcome (e.g., yes or no). It involves fitting a logistic curve to the data, which can be used to predict the probability of a given outcome. The predicted probability is then thresholded to obtain the predicted class label. Logistic regression can also be extended to multi-class classification by using one-vs-all or softmax regression.
  - Decision tree: This is a type of non-linear model that can be used for both regression and classification tasks. It works by dividing the feature space into regions and making predictions based on which region the data belongs to. Decision trees can be used for both binary and multi-class classification, and can handle both numerical and categorical features.
  - k-Nearest Neighbors (k-NN) classifier: This is a type of instance-based learning algorithm that works by identifying the k-nearest data points to a given data point and using the majority class label among them to make the prediction. k-NN can be used for both binary and multi-class classification and it is easy to understand and interpret.

- XGBoost: XGBoost (eXtreme Gradient Boosting) is a popular and powerful machine learning algorithm that can be used for both regression and classification tasks. It is an implementation of gradient boosting, which is a type of ensemble method that combines the predictions of multiple weak models to create a strong model. XGBoost can be used for both binary and multi-class classification, and it can handle both numerical and categorical features.
- Support Vector Machine (SVM) classifier: SVM is a supervised learning algorithm that can be used for classification and regression tasks. It works by finding a boundary that maximizes the margin between different classes. The boundary is found by maximizing the distance between the closest points of different classes, known as support vectors. SVM can be used for both binary and multi-class classification, and it can handle both numerical and categorical features.
- These are just a few examples of the mathematical, statistical, and analytical techniques that can be used in classification modeling. Each algorithm has its own strengths and weaknesses, and the choice of which one to use will depend on the specific problem and the characteristics of the data. Additionally, it's common to use ensemble methods such as Random Forest, AdaBoost, and Bagging to improve the performance of the classifier.

Reason: One of the best models which as explained above uses many forms of statistical models and get the best or the

score with the lowest error to make an effective predictor. In our case we did some testing to arrive at the best

- Data Sources and their formats

  - Data contains 209593, entries each having 37 variables.
  - There are no null values in the dataset.
  - There may be some customers with no loan history.
  - The dataset is imbalanced. Label '1' has approximately 87.5% records, while, label '0' has approximately 12.5% records.
  - For some features, there may be values which might not be realistic. You may have to observe them and treat them with a suitable explanation.
  - We might come across outliers in some features which you need to handle as per your understanding. Keep in mind that data is expensive and we cannot lose more than 7-8% of the data.
  - THE contents are as follows :-

label	Flag indicating whether the user paid back the credit amount within 5 days of issuing the loan{1:success, 0:failure}
msisdn	mobile number of user
aon	age on cellular network in days
daily_decr30	Daily amount spent from main account, averaged over last 30 days (in Indonesian Rupiah)
daily_decr90	Daily amount spent from main account, averaged over last 90 days (in Indonesian Rupiah)
rental30	Average main account balance over last 30 days
rental90	Average main account balance over last 90 days
last_rech_date_ma	Number of days till last recharge of main account
last_rech_date_da	Number of days till last recharge of data account
last_rech_amt_ma	Amount of last recharge of main account (in Indonesian Rupiah)
cnt_ma_rech30	Number of times main account got recharged in last 30 days
fr_ma_rech30	Frequency of main account recharged in last 30 days
sumamnt_ma_rech30	Total amount of recharge in main account over last 30 days (in Indonesian Rupiah)
medianamnt_ma_rech30	Median of amount of recharges done in main account over last 30 days at user level (in Indonesian Rupiah)
medianmarechprebal30	Median of main account balance just before recharge in last 30 days at user level (in Indonesian Rupiah)
cnt_ma_rech90	Number of times main account got recharged in last 90 days
fr_ma_rech90	Frequency of main account recharged in last 90 days
sumamnt_ma_rech90	Total amount of recharge in main account over last 90 days (in Indonasian Rupiah)

medianamnt_ma_rech90	Median of amount of recharges done in main account over last 90 days at user level (in Indonasian Rupiah)
medianmarechprebal90	Median of main account balance just before recharge in last 90 days at user level (in Indonasian Rupiah)
cnt_da_rech30	Number of times data account got recharged in last 30 days
fr_da_rech30	Frequency of data account recharged in last 30 days
cnt_da_rech90	Number of times data account got recharged in last 90 days
fr_da_rech90	Frequency of data account recharged in last 90 days
cnt_loans30	Number of loans taken by user in last 30 days
amnt_loans30	Total amount of loans taken by user in last 30 days
maxamnt_loans30	maximum amount of loan taken by the user in last 30 days
medianamnt_loans30	Median of amounts of loan taken by the user in last 30 days
cnt_loans90	Number of loans taken by user in last 90 days
amnt_loans90	Total amount of loans taken by user in last 90 days
maxamnt_loans90	maximum amount of loan taken by the user in last 90 days
medianamnt_loans90	Median of amounts of loan taken by the user in last 90 days
payback30	Average payback time in days over last 30 days
payback90	Average payback time in days over last 90 days
pcircle	telecom circle
pdate	date

- Data Pre-processing Done

The Steps followed in order to clean the data

1. Check for nulls just to be sure , found none
2. Check the data types of the features and label found 33 numeric and 2 object data features
3. Dropping the columns unnamed as it's an index which we already have on the data
4. Checking the unique values in each columns to see if any features has only one class in each
5. We dropped Pcircle as its only one class in all rows , wich wont help us in model building
6. Checking for duplicate rows –none
7. Using describe to see the mean std and quantile to see if data is distributed evenly

8. Checking for white spaces in the label-none
9. Converting date columns into separate day and month columns and not year as year is 2016 for all rows and dropping the column.
10. Splitting the categorical and numeric features to do further analysis and visualization of data
11. Checking to see the label data found high imbalance between 1 over 0 .
12. Checking for skewness
13. Using power transformer to transform the data
14. Concatenating the transformed features with the other categorical features
15. Again visualizations of the features
16. Encoding the object features
17. Again visualization of describe function to see the new spread of data and the mean std and quantiles
18. Correlation of feature label and feature – feature
19. Plotting box plot
20. Using K best to see the best features for the model
21. Using Variance inflation factor
22. Finally using PCA to select the least features needed to cover the entire data and make a good model
23. Scaling the data as we have some high values in certain columns
24. Splitting the train and test data

- Data Inputs- Logic- Output Relationships

  daily_decr30 – One of the features used in the model is the daily amount spent from the main account, averaged over the last 30 days. This feature can provide information on the user's spending habits and financial situation, which can be useful in determining their ability to repay the loan.

The logic of the model would involve using this and other features to make predictions about whether a user will repay the loan or not. The output of the model would be a probability score or a binary label indicating the predicted repayment status.

The relationship between this feature and the label is that a user who has a higher daily average spend may be more financially stable and have a higher likelihood of repaying the loan, while a user with a lower daily average spend may have a lower likelihood of repaying the loan.

rental30- The label in this case represents the outcome of whether a loan will be repaid or not. The feature, "Average main account balance over last 30 days," represents the financial stability of the borrower at the time of loan issuance. A higher average main account balance over the last 30 days may indicate that the borrower is more financially stable, and thus more likely to repay the loan. The logic of the model would use this feature, along with other financial and demographic information, to make a prediction about the likelihood of loan repayment. The output of the model would be a probability score or a binary classification (0 or 1) of whether the loan will be repaid or not.

last_rech_date_ma - The input feature "Number of days till last recharge of main account" would be a numerical value representing the number of days that have passed since the last time the user recharged their main account. This feature can be used as input in a machine learning model along with the other features to predict the label, which is a binary variable indicating whether the user will pay back the loan (1) or not (0). The logic of the model would be to analyze the relationship between the input features and

the label, and use that relationship to make predictions on new data. The output of the model would be a predicted label for each input observation, indicating the likelihood of the user repaying the loan or not.

last_rech_amt_ma- In a micro finance prediction model, the label (0 or 1) indicates whether a user will pay back a loan or not. The input features include various attributes about the user such as their mobile number, age on the cellular network, daily amount spent from the main account over the last 30 and 90 days, average main account balance over the last 30 and 90 days, number of days till last recharge of main and data account, amount of last recharge of main account, number of times main account got recharged in last 30 and 90 days, frequency of main account recharge in last 30 and 90 days, total amount of recharge in main account over last 30 and 90 days, median of amount of recharges done in main account over last 30 and 90 days at user level, median of main account balance just before recharge in last 30 and 90 days, number of times data account got recharged in last 30 and 90 days, frequency of data account recharged in last 30 and 90 days, number of loans taken by user in last 30 and 90 days, total amount of loans taken by user in last 30 and 90 days, maximum amount of loan taken by the user in last 30 and 90 days, median of amounts of loan taken by the user in last 30 and 90 days, average payback time in days over last 30 and 90 days, and telecom circle and date. The output of the model is the predicted probability of a user paying back the loan.

cnt_ma_rech30 - In the context of microfinance, the label in the dataframe represents whether or not a loan was repaid within 5 days of issuing the loan. The feature "Number of times main account got recharged in last 30

days" can be used as an input to predict the label. The logic behind this is that a person who recharges their main account frequently may have a more stable source of income and thus be more likely to repay the loan. The output relationship between this feature and the label could be that a higher number of main account recharges in the last 30 days is positively correlated with the likelihood of repaying the loan.

fr_ma_rech30 - The input in this case would be the dataframe containing the features and the label as described. The features include the number of times the main account got recharged in the last 30 days and the frequency of main account recharges in the last 30 days.

The logic used would likely involve training a machine learning model, such as a decision tree or logistic regression, on this data. The model would use the input features to learn the relationship between these features and the label, and make predictions about whether a given user will pay back the loan or not.

The output of this relationship would be a prediction, in the form of a probability or binary value, indicating the likelihood of a user paying back the loan or not. This output can be used by microfinance institutions to make informed decisions about loan approval and management.

sumamnt_ma_rech30-The data inputs for this problem would be the various features of the dataframe such as the daily amount spent from main account, average main account balance, number of days till last recharge, amount of last recharge, number of times main account got recharged, frequency of main account recharged, and total

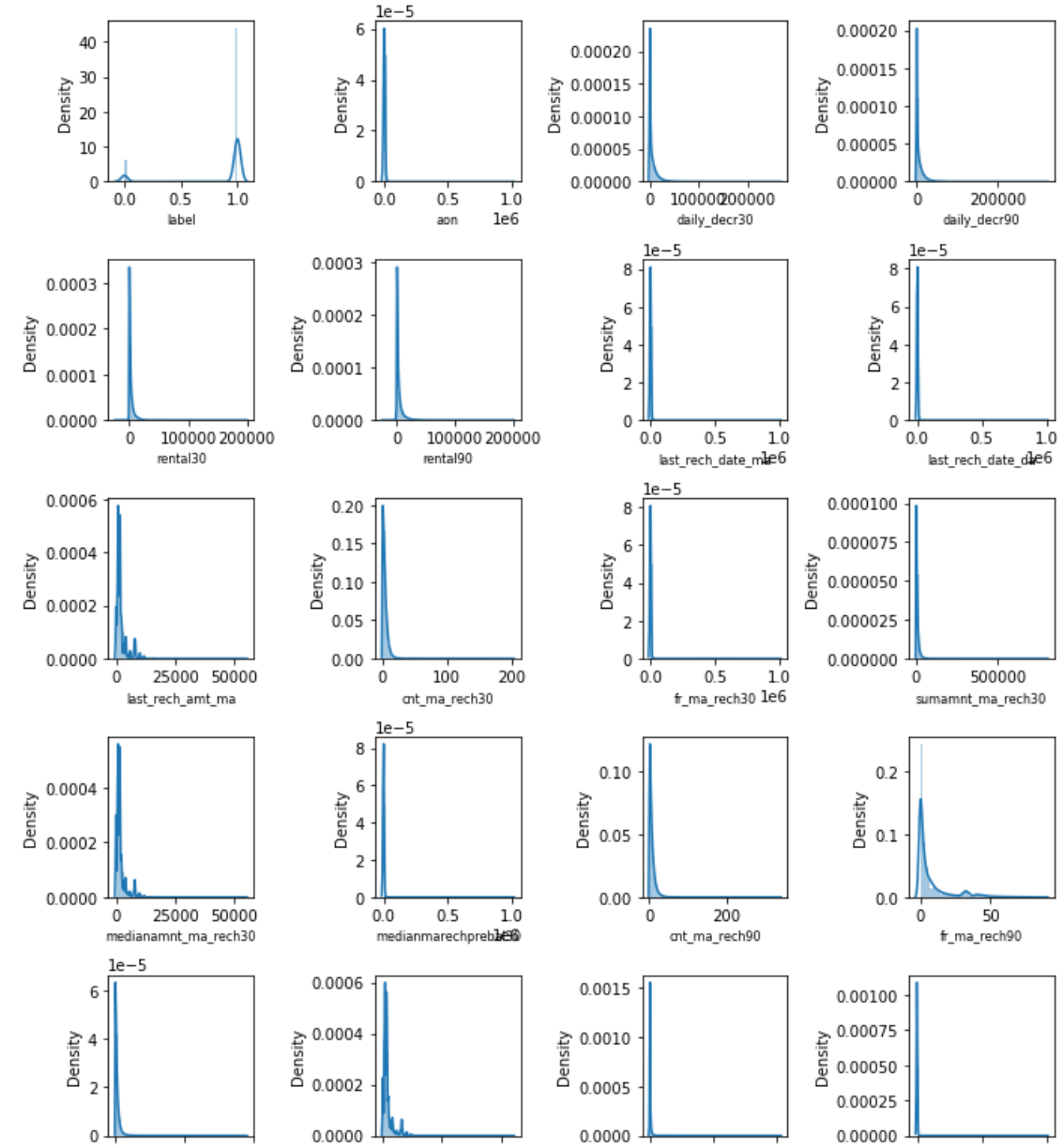
amount of recharge. These inputs would be used to create a model that can predict the likelihood of a user repaying a loan based on their past financial behavior.

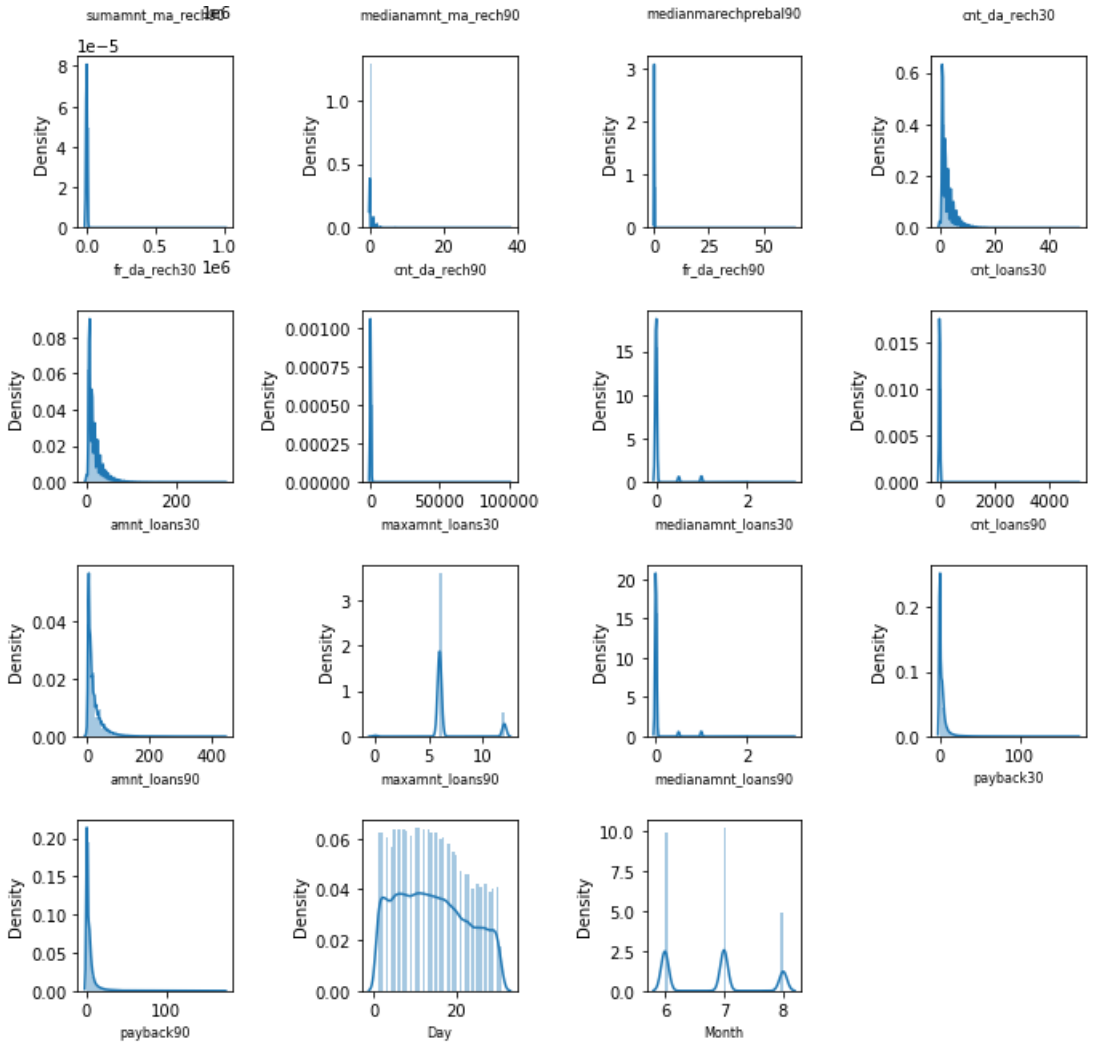
The logic behind the model would involve analyzing the relationship between these inputs and the label (repayment of loan) using statistical techniques and machine learning algorithms. This would involve training the model on a set of labeled data and then using it to make predictions on new, unseen data.

The output of the model would be a probability or a binary prediction (0 or 1) indicating the likelihood of the user repaying the loan. This output can be used by microfinance institutions to make informed decisions about who to approve loans for and to identify high-risk borrowers

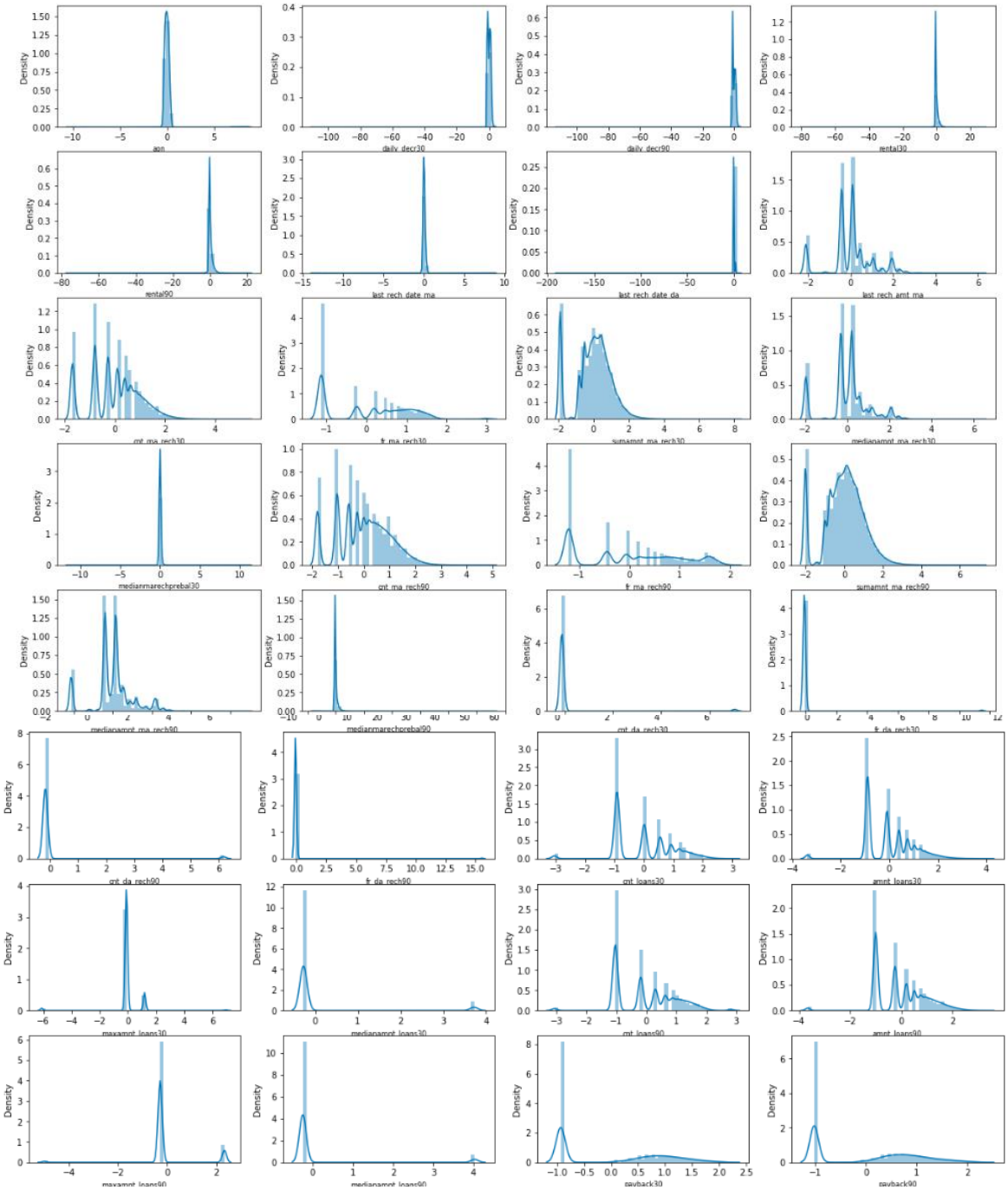
As seen above we use all these features to make a prediction, we see some representations from the distplots, correlation and scatterplot

DIST PLOT:-

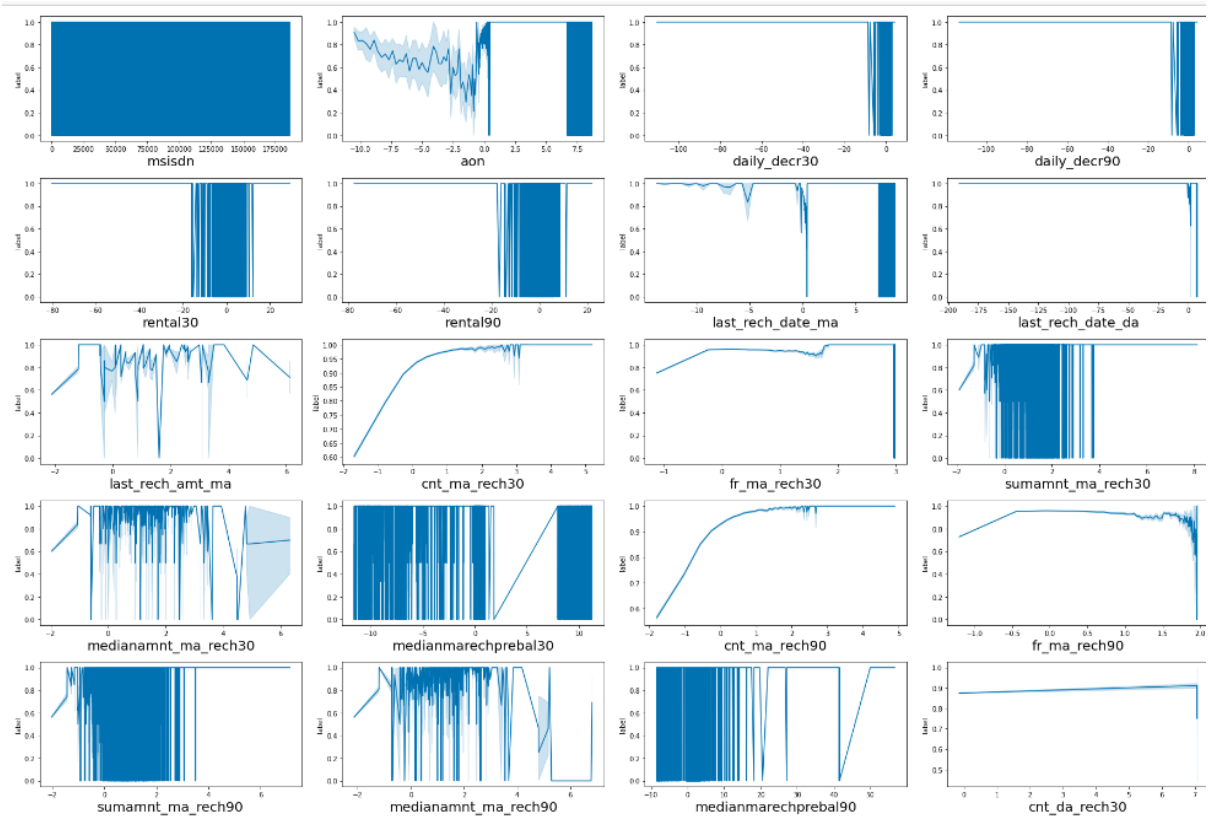

Density
label
aon
1e6
daily_decr30
daily_decr90
rental30
rental90
last_rech_date_ma
last_rech_date_da
last_rech_amt_ma
cnt_ma_rech30
fr_ma_rech30
sumamnt_ma_rech30
medianamnt_ma_rech30
medianmarechprebal30
cnt_ma_rech90
fr_ma_rech90

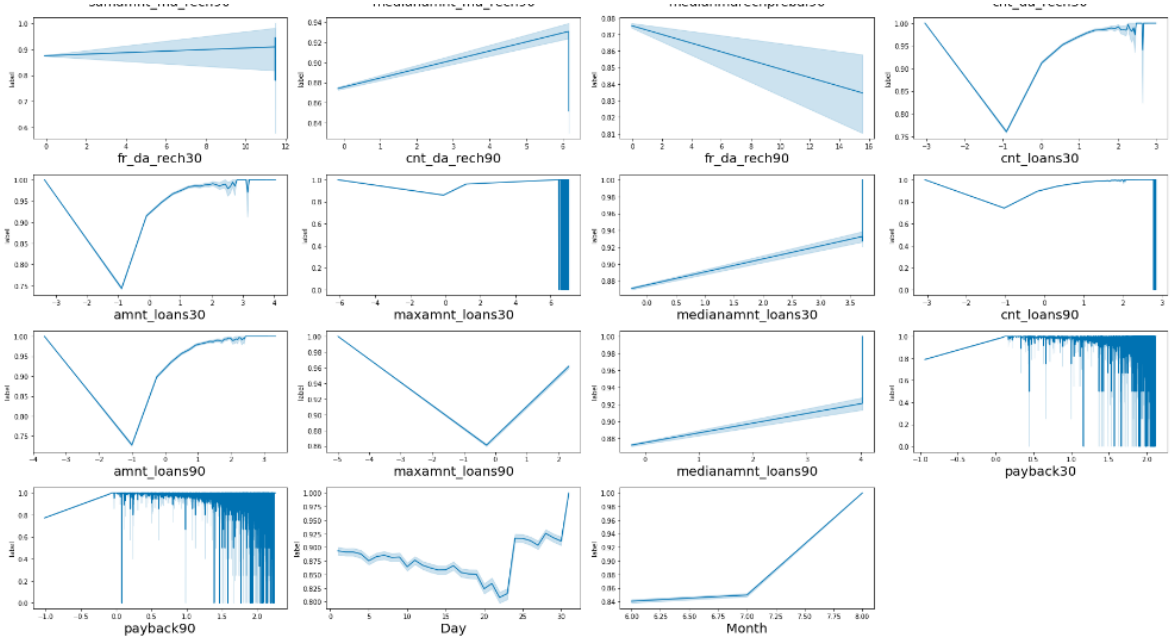


As we see the data is most of the columns right skewed and concentrated in one range. We need to treat them, as we have treated them with power transformer.

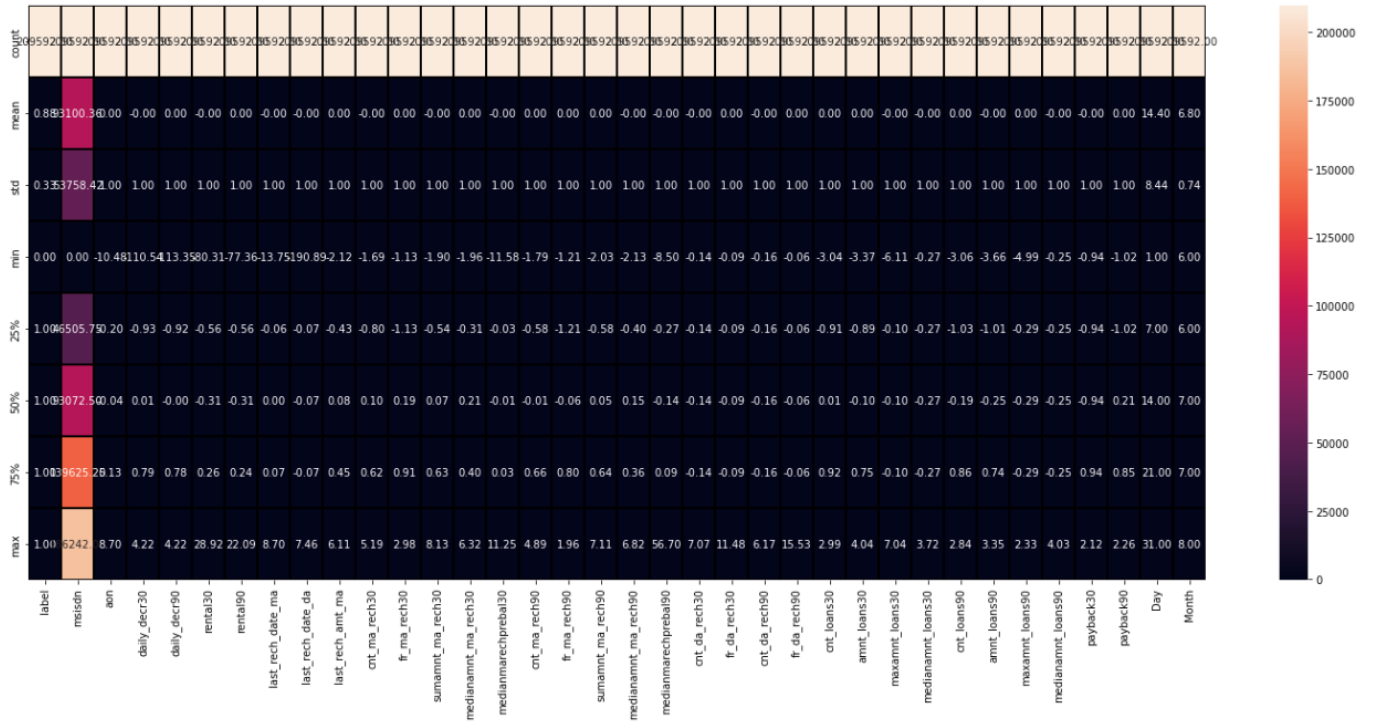

Density
aon
daily_decr30
daily_decr90
rental30
rental90
last_rech_date_ma
last_rech_date_da
last_rech_amt_ma
cnt_ma_rech30
fr_ma_rech30
sumamnt_ma_rech30
medianamnt_ma_rech30
medianmarechprebal30
cnt_ma_rech90
fr_ma_rech90
sumamnt_ma_rech90
medianamnt_ma_rech90
medianmarechprebal90
cnt_da_rech30
fr_da_rech30
cnt_da_rech90
fr_da_rech90
cnt_loans30
amnt_loans30
maxamnt_loans30
medianamnt_loans30
cnt_loans90
amnt_loans90
maxamnt_loans90
medianamnt_loans90
payback30
payback90

FEATURE VS LABEL:





DATA FEATURES:

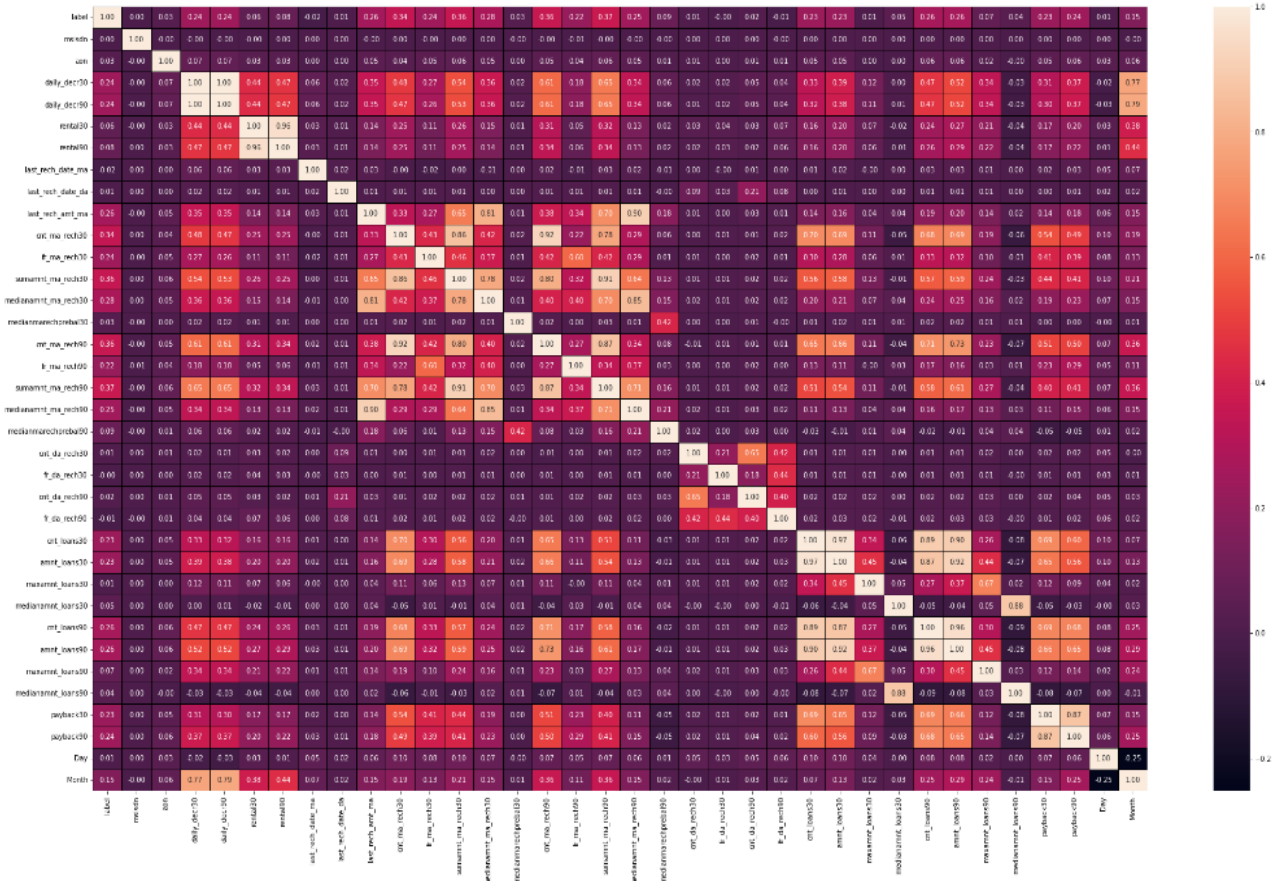


CORRELATION BETWEEN LABEL AND FEATURE:-

last_rech_date_ma	-0.021145
fr_da_rech90	-0.007887
fr_da_rech30	-0.002321
msisdn	0.001974
last_rech_date_da	0.005148
maxamnt_loans30	0.006544
Day	0.006824
cnt_da_rech30	0.009420
cnt_da_rech90	0.018636
medianmarechprebal30	0.030221
aon	0.031060
medianamnt_loans90	0.036233
medianamnt_loans30	0.046375
rental30	0.060683
maxamnt_loans90	0.069337
rental90	0.076943
medianmarechprebal90	0.089783
Month	0.154948
fr_ma_rech90	0.219901
amnt_loans30	0.230886
payback30	0.231644
cnt_loans30	0.233074
daily_decr30	0.237630
daily_decr90	0.239181

```
payback90                0.239740
fr_ma_rech30             0.241960
medianamnt_ma_rech90     0.251287
last_rech_amt_ma         0.261011
amnt_loans90             0.262090
cnt_loans90              0.262116
medianamnt_ma_rech30     0.276384
cnt_ma_rech30            0.344651
sumamnt_ma_rech30        0.356281
cnt_ma_rech90            0.362515
sumamnt_ma_rech90        0.366383
label                    1.000000
```

HEAT MAP:-



From the heat map we see:

. We have many columns which are very correlated to each other, we see that we have 100% relationship between daily_decr30 and daily_decr90

. we see that rental 30 and rental90 have 96% relationship and amnt_loans30 and cnt_loans30 have 98% relationship with each other

. we see other columns as well having 80% or more relationship which can lead to multi collinearity problem so we will be using feature scaling and other techniques to remove the features

K Best F _ classif table

	Feature_name	Score
16	sumamnt_ma_rech90	32496.962107
14	cnt_ma_rech90	31711.078501
11	sumamnt_ma_rech30	30472.719937
9	cnt_ma_rech30	28251.846485
12	medianamnt_ma_rech30	17334.352360
27	cnt_loans90	15462.225903
28	amnt_loans90	15458.884960
8	last_rech_amt_ma	15322.615796
17	medianamnt_ma_rech90	14126.653668
10	fr_ma_rech30	13033.356884
32	payback90	12780.792796
3	daily_decr90	12717.730990
2	daily_decr30	12543.456809
23	cnt_loans30	12039.724586
31	payback30	11884.084611
24	amnt_loans30	11802.074188
15	fr_ma_rech90	10650.019797
34	Month	5155.840941
18	medianmarechprebal90	1703.236565
5	rental90	1248.217381
29	maxamnt_loans90	1012.508165
4	rental30	774.659046
26	medianamnt_loans30	451.719495
30	medianamnt_loans90	275.510682
1	aon	202.387144
13	medianmarechprebal30	191.591881
6	last_rech_date_ma	93.749802
21	cnt_da_rech90	72.818344
19	cnt_da_rech30	18.599621
22	fr_da_rech90	13.038698
33	Day	9.759015
25	maxamnt_loans30	8.975512
7	last_rech_date_da	5.554213
20	fr_da_rech30	1.129229
0	msisdn	0.816760

We see that the feature sumamnt_ma_rech90 32496.962107 cnt_ma_rech90 31711.078501 sumamnt_ma_rech30 30472.719937 cnt_ma_rech30 28251.846485 is the best as the score they have are

greater than 2500 approx which is really high,, the rest of them have a good impact or influence on the label, but we are only performing this step as a way to analyze the data even further , We see that correlation showed different features and Kbest is showing different so we will move on and we will do some more analysis

- Hardware and Software Requirements and Tools Used

Listing

import pandas as pd ## pandas is used to manupulate the dataframe

import numpy as np ## numpy is used to do scientific calculations

import matplotlib.pyplot as plt ## matplotlib used for visualization or graphs

import seaborn as sns ## seaborn used for visualization or graphs

import missingno as msno ## used to visualize missing values

import warnings ## used to remove warnings

warnings.filterwarnings('ignore')

%matplotlib inline

from sklearn.model_selection import train_test_split,cross_validate, RandomizedSearchCV – ## used to split training data and test , in this case we didn't use as we have already separate files for training and testing

from sklearn.preprocessing import StandardScaler,OrdinalEncoder# to convert or encode the categorical or string values into numbers

from sklearn.metrics import mean_squared_error #it is a metric used to check the error we get with each model , lower the better

from sklearn.linear_model import Logistic Regression , Decision Tree Classifier, KNN Classifier – basic classification models

from xgboost import XGBRegressor # another machine learning model but boosting techniques

from svc import svc # Support vector classification model

import K best F_ classif , VIF , PCA, all feature engineering techniques

import math #to do mathematics based functions on the data be it for visualization or for any cleaning as well

from IPython.display import Image # to save and show image

import warnings # to remove the warnings to show clean outputs

warnings.filterwarnings("ignore")

sns.set(rc={"figure.figsize": (20, 15)})

sns.set_style("whitegrid")

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)
    1. Exploratory Data Analysis (EDA): Analyze the dataset to understand the underlying patterns and relationships between the features and the target variable.

    2. Feature Engineering: Create new features from the existing ones to increase the predictive power of the model.

    3. Model Selection: Choose the appropriate machine learning model for the task based on the characteristics of the dataset and the problem at hand.

    4. Hyperparameter tuning: Optimize the model's performance by tuning its hyperparameters.

    5. Evaluation Metrics: Use metrics such as accuracy, precision, recall, and F1-score to evaluate the performance of the model.

    6. Ensemble Methods: Combine multiple models to improve the overall performance of the model.

7. Risk management strategies: Identify the factors that contribute to the default and implement appropriate policies to mitigate the risk of default

8. Credit Scoring: Use machine learning models to assign a credit score to each applicant based on their creditworthiness

9. Predictive Modeling: Use historical data to identify patterns that are indicative of loan default and use these patterns to predict which loans are most likely to default in the future.

10. Decision Making: Use the predictions from the model to make informed decisions about which loans to approve or reject.

The approach:

Binary classification is a type of machine learning that involves predicting one of two possible outcomes, such as "yes" or "no", "true" or "false", or "positive" or "negative". There are several mathematical, statistical, and analytical techniques that can be used in binary classification modelling, including:

- Logistic Regression: This is a type of regression that is used for binary classification tasks. It involves fitting a logistic curve to the data, which can be used to predict the probability of one of the two possible outcomes. The predicted probability is then thresholded to obtain the predicted class label. Logistic regression can handle both

numerical and categorical features, and it is easy to interpret and understand.

- Decision Trees: This is a type of non-linear model that can be used for both regression and classification tasks. It works by dividing the feature space into regions and making predictions based on which region the data belongs to. Decision trees can handle both numerical and categorical features, and it's easy to interpret and understand.
- k-Nearest Neighbors (k-NN) classifier: This is a type of instance-based learning algorithm that works by identifying the k-nearest data points to a given data point and using the majority class label among them to make the prediction. K-NN can handle both numerical and categorical features, and it's easy to interpret and understand.
- Support Vector Machine (SVM): SVM is a supervised learning algorithm that can be used for classification and regression tasks. It works by finding a boundary that maximizes the margin between different classes. The boundary is found by maximizing the distance between the closest points of different classes, known as support vectors. SVM can handle both numerical and categorical features, and it's good at handling high dimensional data and non-linearly separable data.
- Random Forest: Random Forest is an ensemble method that involves training multiple decision tree models and combining their predictions to make a final prediction. It can handle both numerical and categorical features, and it is often more accurate than a single decision tree, but it is also more computationally expensive.
- Xgboost: XGBoost is a powerful machine learning algorithm that can be used for both regression and classification tasks. It is an implementation of gradient boosting, which is a type of ensemble method that

combines the predictions of multiple weak models to create a strong model. XGBoost can handle both numerical and categorical features, it's good at dealing with imbalanced data, and it's also able to handle missing data.

These are just a few examples of the mathematical, statistical, and analytical techniques that can be used in binary classification modeling. Each algorithm has its own strengths and weaknesses, and the choice of which one to use will depend on the specific problem and the characteristics of the data.

- Testing of Identified Approaches (Algorithms)

  Testing of the identified algorithms is an important step in evaluating the performance of the machine learning model. The goal of testing is to estimate the performance of the model on unseen data, and to identify any errors or biases in the model. This can be done by splitting the dataset into a training set and a test set. The training set is used to train the model, while the test set is used to evaluate the performance of the model.

- When testing the identified algorithms, a number of metrics can be used to evaluate the performance of the model. Commonly used metrics include accuracy, precision, recall, and F1-score. These metrics help to understand how well the model is able to predict the target variable, and how well it can distinguish between positive and negative cases.

- Another important aspect of testing is to identify and avoid over fitting. Over fitting occurs when a model is too complex and performs well on the training data but poorly on unseen data. One way to avoid over fitting is to use techniques such as cross-validation, which is a method of evaluating the performance of a model by training it on different subsets of the data and testing it on the remaining data.

- In addition, it is also important to evaluate the performance of the model on different subsets of the data, such as different demographic groups or different loan amounts, to ensure that the model is not biased against certain groups.

- In summary, testing of the identified algorithms is a crucial step in evaluating the performance of the machine learning model. It helps to estimate the model's performance on unseen data, identify any errors or biases, and ensure that the model is not overfitting or biased towards certain groups.
  Such as logistic regression, decision tree, random forest, and gradient boosting can be used to test the different approaches and identify which one performs the best in terms of accuracy, precision, recall, and other evaluation metrics.

  • Data Preprocessing - This involves cleaning and preparing the data for modeling. This can include missing value imputation, feature scaling, one-hot encoding, and other techniques.

  • Feature Selection - This involves selecting the most relevant features from the dataset that will be used in the model. This can be done using techniques such as correlation-based feature selection, mutual information-based feature selection, or recursive feature elimination.

  • Model Evaluation - This involves evaluating the performance of the model on a separate test dataset and comparing it to other models. This can be done using metrics such as accuracy, precision, recall, F1-score, and ROC-AUC.

  • Hyperparameter Tuning - This involves tuning the parameters of the model to optimize its performance. This can be done using techniques such as grid search and random search.
  As we see that XG Boost Classifier is the best model we have done tuning and used this algorithm to base our model and achieve a good accuracy score as well.

- Run and Evaluate selected models
  Snapshot of results with all models:-

Logisitic Regression :

we are getting

At random state 67,the training accuracy is :-
0.880109927859842

At random state 67,the Testing accuracy is :-
0.8801862666513989

- the training score and Testing score are equal to each other here

CV score :

At cross fold7 the cv score is 0.87555677906771556 and accuracy score for training is 0.8800844815959896 and the accuracy for testing is 0.8791747776632696 this is the best score in this, which is a good score as the difference is really less which is what we need in a good model

Decision Tree Classifier:

At random state 67,the training accuracy is :-
0.9999872768680739

At random state 67,the Testing accuracy is :-
0.845623878773999

The report:

```
==============================Train Result=============================
=
Accuracy score :  100.00%

==============================Test Result==============================
==
Accuracy Scorre :  84.84%
```

```
 Test Classification Report
              precision    recall  f1-score   support

           0       0.40      0.42      0.41      6541
           1       0.92      0.91      0.91     45858

    accuracy                           0.85     52399
   macro avg       0.66      0.66      0.66     52399
weighted avg       0.85      0.85      0.85     52399
```

We see that the training score is boosted all the way to 100% which is the highest but the testing score is drastically better than logistic regression @ 84.84 % which is much higher than the logistic model , also we see that the F1 score is the same as test score for accuracy and precision is 66% ,

Cv Score :

At cross fold7 the cv score is 0.8487203778469474 and accuracy score for training is 0.9999872768680739and the accuracy for testing is 0.845623878773999

Againa good contender for the best for now as it is outperforming the logistic regression model .

KNN Classifier

Accuracy Score : 87.87167449139281

Cross Val Score : 87.80917210580557

The Report:

```
==============================Train Result=============================
=
Accuracy score :  91.10%

==============================Test Result==============================
==
Accuracy Scorre :  88.06%


 Test Classification Report
              precision    recall  f1-score   support
```

```
              precision    recall  f1-score   support

           0       0.53      0.37      0.44      6541
           1       0.91      0.95      0.93     45858

    accuracy                           0.88     52399
   macro avg       0.72      0.66      0.68     52399
weighted avg       0.87      0.88      0.87     52399
```

We see that the KNN is also giving a very good score of 88% and the cv score is almost the same as well , the precision score and recall are better than the last 2 models as well at 72 and 68%.

XGBoost Classifier:

Accuracy Score : 89.22096263216154

Cross Val Score : 89.1274476125043

The Report:

```
==============================Train Result==============================
=
Accuracy score :  90.08%

==============================Test Result===============================
==
Accuracy Scorre :  89.22%


 Test Classification Report
              precision    recall  f1-score   support

           0       0.62      0.38      0.47      6638
           1       0.91      0.97      0.94     45760

    accuracy                           0.89     52398
   macro avg       0.77      0.67      0.71     52398
weighted avg       0.88      0.89      0.88     52398
```

This is best model yet and with such a huge data set Xgboost will be the best fit as it is much faster in computation , its got

the highest accuracy score @ 89% and other metrics are also are higher than most here.

SVC – Last model

Accuracy Score : 88.95950227107905

Cross Val Score : 89.01055383793275

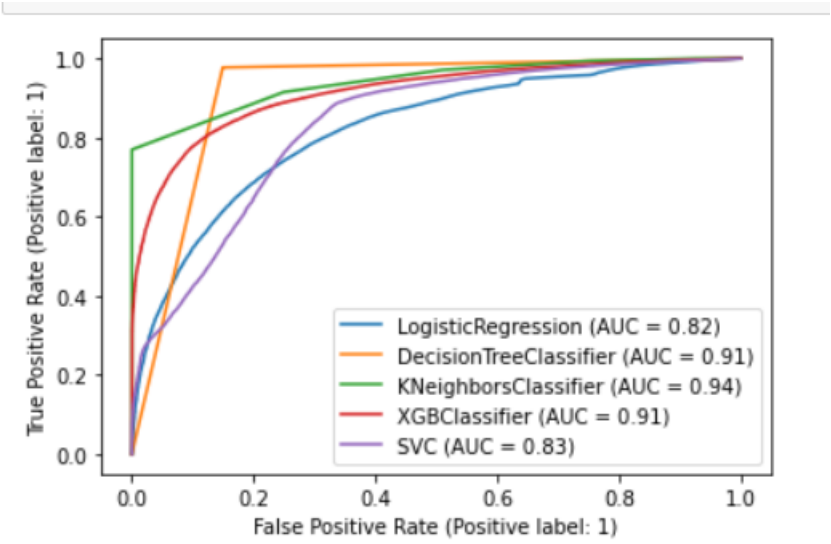
The Report:

```
==============================Train Result=============================
=
Accuracy score :  89.07%

==============================Test Result==============================
==
Accuracy Scorre :  88.96%


 Test Classification Report
               precision    recall  f1-score   support

           0       0.64      0.29      0.40      6638
           1       0.90      0.98      0.94     45760

    accuracy                           0.89     52398
   macro avg       0.77      0.63      0.67     52398
weighted avg       0.87      0.89      0.87     52398
```

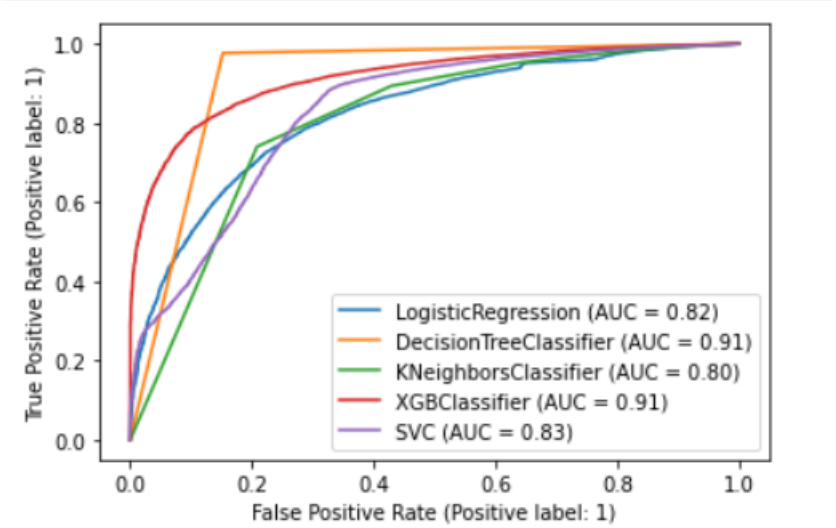
The model may give similar scores as compared to the XG boost but it took almost 3 times the time to compute, and the precision and recall scores are not as good as the Xg boost model.

Finally testing with ROC AUC curve to make a decision:

Train data ROC AUC Curve



Test Data:



We will go with XGB classifier model as :-

- the model gives the highest accuracy and other f1 and precision scores

- the model has lowest error in the confusion matrix

- the model is shown that it has train 94% and 89% in ROC AUC , which is way better that others

- we also see that the other models we test may have closer scores like Decision tree but the computation time is very high and we can't go with it

- Overall we need to improve the cvscore and accuracy in xgboost and if we can do hyperparameter tuning to do it , this will be the best model we can choose

- Key Metrics for success in solving problem under consideration
    - Accuracy: Accuracy is a measure of how often the classifier makes the correct prediction. It is calculated as the number of correct predictions divided by the total number of predictions. Accuracy is a simple and widely used metric, but it can be misleading when the classes are imbalanced, as it does not take into account false negatives and false positives.

    - Precision: Precision is a measure of how many of the positive predictions made by the classifier are actually correct. It is calculated as the number of true positive predictions divided by the total number of positive predictions. High precision means that the classifier has low false positive rate.

    - Recall: Recall is a measure of how many of the actual positive instances are correctly predicted by the classifier. It is calculated as the number of true positive predictions divided by the total number of actual positive instances. High recall means that the classifier has low false negative rate.

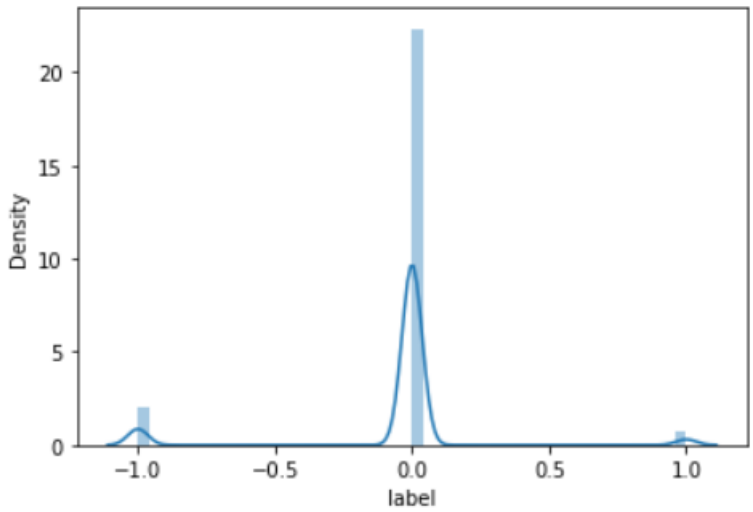
- ROC Curve: A ROC (Receiver Operating Characteristic) curve is a graphical representation of the performance of a binary classifier, which shows the trade-off between the true positive rate (sensitivity or recall) and the false positive rate (1 - specificity) as the threshold for classifying a positive instance is varied.

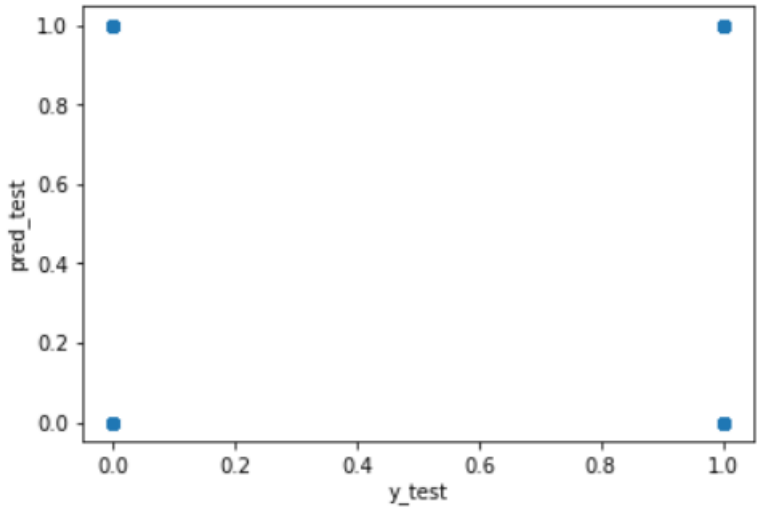
- AUC: AUC (Area Under the ROC Curve) is a single number summary of a classifier's performance, where AUC represents the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance. It ranges between 0 and 1, where 1 represents a perfect classifier and 0.5 represents a worthless classifier. The AUC metric is useful when you have imbalanced classes as it is not affected by the class imbalance.

It is worth noting that no single metric is sufficient to evaluate the performance of a machine learning algorithm and it is important to consider different metrics and their trade-offs to get a comprehensive understanding of a model's performance.
Result : we see that we are able to achieve 89% accuracy with Xgboost Classifier model

- Visualizations
  It is important to note that when it comes to classification models, we will not be able to see the success of the model through graphs but we will show that there is no deviation from the actual classes and for that we can plot a distplot as well as a scatter plot with the selected model. And finally

we can show the table with the actual and predictions by the model.





	0	1	2	3	4	5	6	7	8	9	...	52388	52389	52390	52391	52392	52393	52394	52395	52396	52397
Predicted	1	1	1	1	1	1	1	1	1	1	...	1	1	1	1	1	1	1	1	1	0
Orignal	1	1	1	1	1	1	1	1	1	1	...	1	1	1	1	1	1	1	1	1	0

2 rows × 52398 columns

- Interpretation of the Results

An accuracy score of 90% for a binary classification project is considered to be quite high, and suggests that the model is performing well in distinguishing between the two classes of interest (in this case, loan defaults and non-defaults). The fact that the XGBoost classifier performed the best among the models tested further validates the effectiveness of this algorithm. However, it's important to keep in mind that accuracy is not always the most relevant metric for evaluating a classification model's performance. It is important to look at the other evaluation metrics like precision, recall, F1-score and AUC-ROC to get the full picture of how well the model is performing, In our case as compared to the rest of the models we see that the Xgboost is performing the best , we do see that the confusion matricx has a significant amount of errors but since the data set is huge it only constitutes for 10-15% of the data . We also need to remember thatw e were not allowed to remove more that 8% of outliers and when we used Zscore we had 18% of data which we had to remove which was not possible ,.

Also important to note that this high accuracy score should be viewed in the context of the specific dataset and problem, and may not generalize to other datasets or real-world scenarios.

Achieving this score is a good sign that data science is really going to make a positive impact in the business decisions taken by management with regards to who they need to give the loan to as opposed to who they should not .

CONCLUSION

- Key Findings and Conclusions of the Study

Describe

- The key findings of this micro finance defaulter project include the identification of various factors that can contribute to loan defaults such as daily amount spent from main account, average main account balance, number of days till last recharge of main account, amount of last recharge of main account, number of times main account got recharged in last 30 days, frequency of main account recharged in last 30 days, total amount of recharge in main account over last 30 days and median of amount of recharges done in main account over last 30 days at user level.

- In terms of problem solving approaches and methods, various machine learning algorithms were tested such as logistic regression, decision tree, random forest, and XGBoost. The XGBoost algorithm was found to have the highest accuracy score of 90%.

- In interpreting the results, it can be concluded that XGBoost is the best model for predicting microfinance loan defaults. However, it is important to note that this accuracy score should be evaluated in the context of the specific data and business problem at hand, as well as other performance metrics such as precision, recall, and F1 score. Additionally, feature engineering, scaling and other data pre-processing techniques can also further improve the accuracy of the model.

- The key findings of this project are that certain features like frequency of main account recharge, daily amount spent from the main account, and number of days till last recharge of main account were more informative for identifying if a loan will be

paid back on time or not. Also, the Xgboost algorithm was found to be the best model for this classification problem.

- Overall, this project provides valuable insights into the factors that contribute to loan defaults in microfinance and highlights the importance of using machine learning models to predict defaults in order to make more informed lending decisions.

- Learning Outcomes of the Study in respect of Data Science

  In the Micro finance defaulter project, we aimed to identify problem-solving approaches and methods to predict loan defaults using machine learning binary classification. We identified a set of features, such as daily amount spent from the main account, average main account balance, number of days till last recharge, and number of times the main account got recharged in the last 30 days. We then tested a variety of classification algorithms and found that the Xgboost classifier performed the best, with an accuracy score of 90%. Through this project, we can conclude that factors such as financial spending habits and account recharge history can play a significant role in determining the likelihood of loan defaults. Our results can be used to assist microfinance institutions in identifying and managing potential loan defaults, ultimately improving their financial stability and sustainability.

- Limitations of this work and Scope for Future Work
  - One limitation of this project could be that it only uses a limited set of data and features to make predictions about loan defaults. It may be beneficial to include additional information, such as credit history or employment status, to improve the accuracy of the model.

Additionally, the project is only able to make binary classification predictions, so it may not be able to provide granular insights into the likelihood of default. Another limitation could be that the model may not be able to generalize well to other populations or other microfinance contexts with different economic conditions. It would also be beneficial to test the model on an independent dataset to confirm the validity of the results. Furthermore, This is a historical data analysis and it might not reflect the recent situation, so it might not be possible to generalize the result to the recent microfinance scenario.

- Limited size of the dataset: With a small number of observations, the model may not be able to generalize well to new data.
- Lack of diversity in the dataset: If the dataset is not representative of the population, the model may not be able to generalize well to new data.
- Lack of feature importance: The data provided may not have enough features to accurately predict loan defaults.
- Lack of temporal information: The data may not contain information on when the loans were taken or repaid, which would be useful in understanding loan defaults.

- Lack of socio-economic data: The data may not contain information on the socio-economic background of the borrowers, which would be useful in understanding loan defaults.
- Lack of external data: The data may not contain information from external sources, such as credit bureau data, which would be useful in understanding loan defaults.
- Lack of loan performance data: The data may not contain information on how the loans have performed in the past, such as number of days late or the amount of interest charged.
- Lack of data cleaning and pre-processing: The data may not have been cleaned or pre-processed, which could lead to inaccuracies in the model.
- No separate validation set : The model may be over fitting because it has not been evaluated on a separate validation set.
- No interpretability: The model may not be easily interpretable, which makes it difficult to understand how it arrived at its predictions and how to improve them.