Micro Credit Defaulter Project

Submitted by:

Kshitij Chawla

ACKNOWLEDGMENT

In this project different libraries and methods are used that are available in python which helped in completion of the project:

https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

http://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.PowerTransformer.html

http://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html

http://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html

http://scikit-learn.org/stable/modules/model_evaluation.html

http://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html

http://scikit-learn.org/stable/auto_examples/classification/plot_classifier_comparison.html

https://seaborn.pydata.org/generated/seaborn.countplot.html

https://www.analyticsvidhya.com/blog/2020/06/auc-roc-curve-machine-learning/

https://www.analyticsvidhya.com/blog/2020/10/how-to-choose-evaluation-metrics-for-classification-model/

https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/

INTRODUCTION

- Business Problem Framing

  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.

  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 industry 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.

  Today, microfinance is widely accepted as a poverty-reduction tool, representing $70 billion in outstanding loans and a global outreach of 200 million clients.

  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.

  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.

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).

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

The project will require knowledge and practice in building Graphs /plots and analysing them to get the relationship between dataset, Knowledge of Different Learning Models to build and predict the required output. Basic Data science concepts to increase the quality of the dataset and Python Knowledge (Coding Language) which will be used to solve the complete Micro Credit Defaulter project. Understanding of calculating F2 score, accuracy, skewness and basic mathematics/statistical approaches will help to build an accurate model for this project.

- ## Review of Literature

Microcredit is an extremely small loan given to those who lack a steady source of income, collateral, or any credit history. It aims to support and kickstart entrepreneurs who are unable to obtain the financial backing needed to start a small business or capitalize on an idea.

It is also more common in underdeveloped countries, as it is aimed to support people of a lower socioeconomic background. Individuals who receive a microcredit loan may be illiterate; thus, they are unable to apply for conventional loans due to the paperwork involved.

Microcredit is also part of microfinance, a line of finance that aims to help people of a lower socioeconomic background through catered financial services, which include savings accounts and loans.

It is said to be originated in 1983 by the Grameen Bank in Bangladesh, with the idea coming from economist Muhammad Yunus. More recently, it's been used as a tool to hopefully decrease the increasing wealth gap

Though the term microcredit is relatively new as it was invented in 1983, the concept is to provide financial help to those of a lower socioeconomic background. It is said that lending to people of lower socioeconomic background goes as far back as the 1700s in Ireland.

However, a new vision on the delivery of microcredit was introduced from the 1970s to the 1980s, and Muhammad Yunus was a key player in shaping the vision. He decided to open Grameen Bank in 1983 and realize his vision. Grameen Bank was able to receive funding and created a microcredit model.

One of the first examples of microcredit originated from a group of women who created bamboo stools in Bangladesh. The women were earning a minimal profit of $0.02 on each stool due to the repayment of suppliers.

Muhammad thought that if the women were provided with a source of credit to draw from to fulfil payments to suppliers, the women could make it out of poverty. The women were loaned $27 and were able to sustain the business and pay the loan off.

- Motivation for the Problem Undertaken

  I wanted to solve the real-life problem using the Technical skills gathered during the course of being a Data Analyst and improving the skill set.

Analytical Problem Framing

- Mathematical/ Analytical Modelling of the Problem----

  Classification Models->

  Classification is a process in which an algorithm is used to analyse an existing data set of known points. The understanding achieved through that analysis is then leveraged as a means of appropriately classifying the data. Classification is a form of machine learning that can be particularly helpful in analysing very large, complex sets of data to help make more accurate predictions.

  “Classification models are a form of supervised machine learning which is often used when the analyst needs to understand how they got to a certain point,”

  Some of the most common classification models include decision trees, random forests, nearest neighbour, and Naive Bayes.

  Decision Tree –

  It is a decision-making tool that uses a flowchart-like tree structure or is a model of decisions and all of their possible results, including outcomes, input costs and utility.

  Decision-tree algorithm falls under the category of supervised learning algorithms. It works for both continuous as well as categorical output variables.

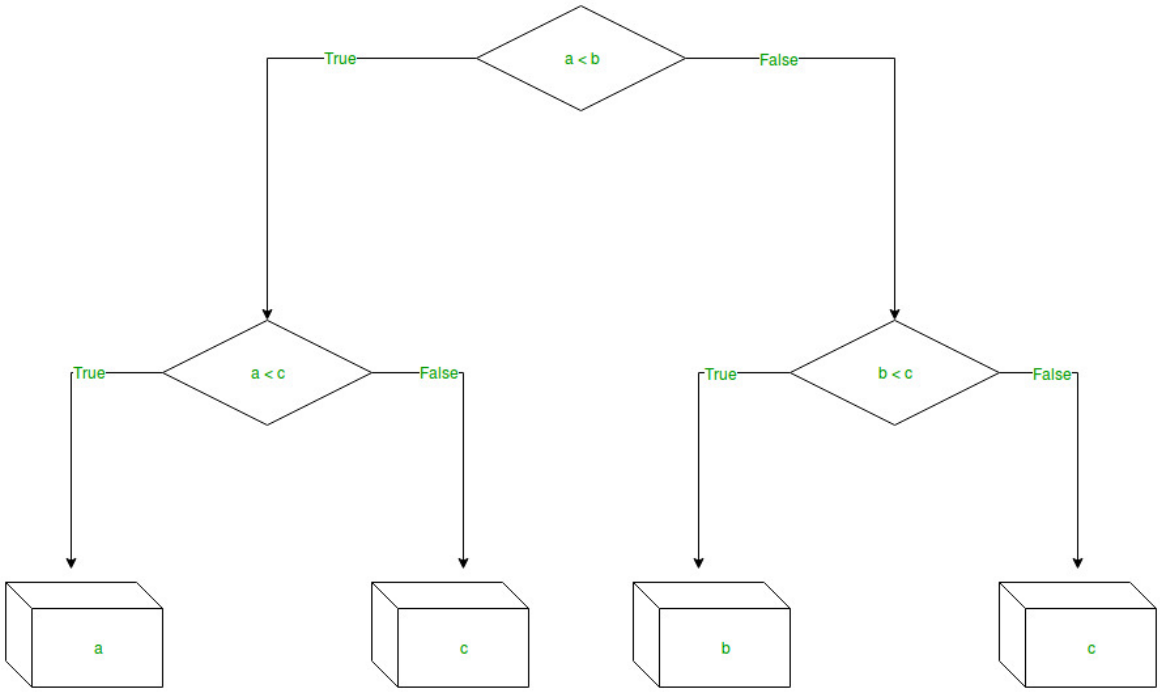
  The branches/edges represent the result of the node and the nodes have either:

  Conditions [Decision Nodes]

  Result [End Nodes]

  The branches/edges represent the truth/falsity of the statement and takes makes a decision based on that in the example below which shows a decision tree that evaluates the smallest of three

numbers:



Random Forest –

Random forest is a supervised learning algorithm which is used for both classification as well as regression. But however, it is mainly used for classification problems. As we know that a forest is made up of trees and more trees means more robust forest. Similarly, random forest algorithm creates decision trees on data samples and then gets the prediction from each of them and finally selects the best solution by means of voting. It is an ensemble method which is better than a single decision tree because it reduces the over-fitting by averaging the result.

Naive Bayes –

Naive Bayes classifiers are a collection of classification algorithms based on Bayes' Theorem. It is not a single algorithm but a family of algorithms where all of them share a common principle, i.e. every pair of features being classified is independent of each other.

Logistic Regression –

Logistic regression is a supervised learning classification algorithm used to predict the probability of a target variable. The nature of target or dependent variable is dichotomous, which means there would be only two possible classes.

In simple words, the dependent variable is binary in nature having data coded as either 1 (stands for success/yes) or 0 (stands for failure/no).

Mathematically, a logistic regression model predicts P(Y=1) as a function of X. It is one of the simplest ML algorithms that can be used for various classification problems such as spam detection, Diabetes prediction, cancer detection etc.

SVM –

Supervised Machine Learning Algorithm used for classification and/or regression. It is more preferred for classification but is sometimes very useful for regression as well. Basically, SVM finds a hyper-plane that creates a boundary between the types of data. In 2-dimensional space, this hyper-plane is nothing but a line.

We used different Plots/ graphs to perform EDA on the dataset->

1) Box Plot**:** It is a type of chart that depicts a group of numerical data through their quartiles. It is a simple way to visualize the shape of our data. It makes comparing characteristics of data between categories very easy.
2) Count Plot: IT is kind of like a histogram or a bar graph for some categorical area. It simply shows the number of occurrences of an item based on a certain type of category
3) Heat Map: It contains values representing various shades of the same colour for each value to be plotted. Usually the darker shades of the chart represent higher values than the lighter shade. For a very different value a completely different colour can also be used.

4) Scatter Plot: A scatter plot is a diagram where each value in the data set is represented by a dot. The Matplotlib module has a method for drawing scatter plots

- Data Sources and their formats

Below are the fields present in our dataset with the information what these fields describe

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 types of the fields:

Below is the information of all the attributes with their respective datatypes:

Column name	datatype
label	int64
msisdn	object
aon	float64
daily_decr30	float64
daily_decr90	float64
rental30	float64
rental90	float64
last_rech_date_ma	float64
last_rech_date_da	float64
last_rech_amt_ma	int64
cnt_ma_rech30	int64
fr_ma_rech30	float64
sumamnt_ma_rech30	float64
medianamnt_ma_rech30	float64
medianmarechprebal30	float64
cnt_ma_rech90	int64
fr_ma_rech90	int64
sumamnt_ma_rech90	int64
medianamnt_ma_rech90	float64
medianmarechprebal90	float64
cnt_da_rech30	float64
fr_da_rech30	float64
cnt_da_rech90	int64
fr_da_rech90	int64
cnt_loans30	int64
amnt_loans30	int64
maxamnt_loans30	float64
medianamnt_loans30	float64
cnt_loans90	float64
amnt_loans90	int64
maxamnt_loans90	int64
medianamnt_loans90	float64
payback30	float64
payback90	float64
pcircle	object
pdate	object

- Data Pre-processing Done

1) First we checked the data set dimensions

```
df['pcircle'].value_counts()
```

```
UPW    209593
Name: pcircle, dtype: int64
```

2) Then we checked whether there is any repeating data available

```
duplicate = df.duplicated()
print(duplicate.sum())
df[duplicate]
```

```
0
```

3) We checked the outliers using the Box Plot and replaced the outliers with more appropriate values. Removal of outliers can also be done but taking the Data Loss percentage into consideration It is better to replace the outliers

4) We checked the skewness for the dataset:

Column name	skewness
label	-2.270254
aon	10.392949
daily_decr30	3.946230
daily_decr90	4.252565
rental30	4.521929
rental90	4.437681
last_rech_date_ma	14.790974
last_rech_date_da	14.814857
last_rech_amt_ma	3.781149
cnt_ma_rech30	3.283842
fr_ma_rech30	14.772833
sumamnt_ma_rech30	6.386787
medianamnt_ma_rech30	3.512324
medianmarechprebal30	14.779875
cnt_ma_rech90	3.425254
fr_ma_rech90	2.285423
sumamnt_ma_rech90	4.897950
medianamnt_ma_rech90	3.752706
medianmarechprebal90	44.880503
cnt_da_rech30	17.818364
fr_da_rech30	14.776430
cnt_da_rech90	27.267278
fr_da_rech90	28.988083
cnt_loans30	2.713421
amnt_loans30	2.975719
maxamnt_loans30	17.658052
medianamnt_loans30	4.551043
cnt_loans90	16.594408
amnt_loans90	3.150006
maxamnt_loans90	1.678304
medianamnt_loans90	4.895720
payback30	8.310695
payback90	6.899951

As we can see that in attributes, we have both positive and negative skewness which will be required to be removed before handling the dataset. Hence Removed the Skewness from the dataset.

5) The object type datatypes were required to be converted into int/float type datatype:
    a) Converted ‘pdate’ column into 3 different columns providing date/month/year in int format
    b) Removed ‘pcircle’ column from the dataset as the column had same value in all the rows.
6) We checked the dataset and found the data is not scaled which will affect the model accuracy, so we performed the Scaling using MinMax Scaler.

- Hardware and Software Requirements and Tools Used

1) Software: Jupyter Notebook - To code and build the project in python
2) Libraries:
    a) numpy - To perform basic math operations
    b) pandas - To perform basic File operations
    c) Matplotlib - To plot Different Graphs/ Plots
    d) Seaborn - Advance library to enhance the quality of graphs/plots
    e) warnings - To ignore the unwanted warnings raised while interpreting the code
    f) sklearn - To build the Prediction models
    g) imblearn - To balance our dataset distribution

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

  We used different approaches from checking the dataset quality to building the model. We checked the null values and repeated rows in the dataset. For checking the Outliers, we used Box Plot and to remove the outliers we used IQR method. Then we moved to next step of checking data distribution and skewness. To scale the data, we used MinMax Scaler method and to remove the skewness we first checked the log and square root method but skewness of the dataset was not getting removed from it so we performed the Power transform to remove skewness. After that we solved the issue of unbalanced target data using SMOTE technique where it creates the sample data to fill the gap between the data. We started building different models and checked their Accuracy, Re-call, F1 score and selected the best suited model to perform Hyper tuning on. We got Random Forest Algo with the best result and after performing Hyper tuning we finalized the model.

- Testing of Identified Approaches (Algorithms)

  1) Logistic Regression
  2) Decision Tree
  3) Support Vector Machine
  4) Naïve Bayes
  5) Random Forest

- Run and Evaluate selected models

-------Logistic Regression-------

In [29]:
```
LogReg = LogisticRegression(max_iter = 500)
LogReg.fit(x_train_res,y_train_res)
pred = LogReg.predict(x_test)
acc = classification_report(y_test, pred)
print(acc)
```

```
              precision    recall  f1-score   support

           0       0.32      0.78      0.45      7924
           1       0.96      0.76      0.85     54954

    accuracy                           0.76     62878
   macro avg       0.64      0.77      0.65     62878
weighted avg       0.88      0.76      0.80     62878
```

-------DecisionTreeClassifier-------

In [30]:
```
DTC = DecisionTreeClassifier()
DTC.fit(x_train_res,y_train_res)
pred = DTC.predict(x_test)
acc = classification_report(y_test,pred)
print(acc)
```

```
              precision    recall  f1-score   support

           0       0.48      0.61      0.54      7924
           1       0.94      0.91      0.92     54954

    accuracy                           0.87     62878
   macro avg       0.71      0.76      0.73     62878
weighted avg       0.88      0.87      0.87     62878
```

------- Random Forest Classifier -------

In [31]:
```
RFC = RandomForestClassifier()
RFC.fit(x_train_res,y_train_res)
pred = RFC.predict(x_test)
acc = classification_report(y_test,pred)
print(acc)
```

```
              precision    recall  f1-score   support

           0       0.65      0.62      0.63      7924
           1       0.95      0.95      0.95     54954

    accuracy                           0.91     62878
   macro avg       0.80      0.79      0.79     62878
weighted avg       0.91      0.91      0.91     62878
```

------- GaussianNB -------

In [32]:
```
from sklearn.naive_bayes import GaussianNB
gnb = GaussianNB()
gnb.fit(x_train_res, y_train_res)

# making predictions on the testing set
pred = gnb.predict(x_test)
acc = classification_report(y_test,pred)
print(acc)
```

```
              precision    recall  f1-score   support

           0       0.28      0.79      0.42      7924
           1       0.96      0.71      0.82     54954

    accuracy                           0.72     62878
   macro avg       0.62      0.75      0.62     62878
weighted avg       0.87      0.72      0.77     62878
```

------- SVC -------

```
In [33]: from sklearn.svm import LinearSVC

         clf = LinearSVC(random_state=0, tol=1e-5)

         clf.fit(x_train_res, y_train_res.ravel())
```

```
Out[33]: LinearSVC(random_state=0, tol=1e-05)
```

```
In [34]: pred = clf.predict(x_test)
         acc = classification_report(y_test,pred)
         print(acc)
```

```
              precision    recall  f1-score   support

           0       0.41      0.61      0.49      7924
           1       0.94      0.87      0.90     54954

    accuracy                           0.84     62878
   macro avg       0.67      0.74      0.70     62878
weighted avg       0.87      0.84      0.85     62878
```

- Key Metrics for success in solving problem under consideration

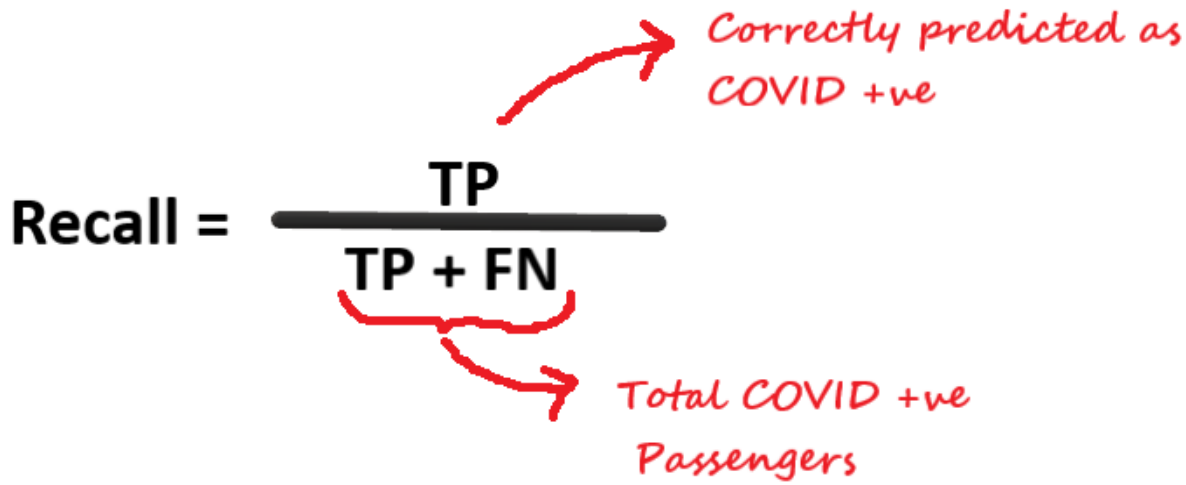
1) Accuracy:
   We want our model to focus on True positive and True Negative. Accuracy is one metric which gives the fraction of predictions our model got right. Formally, accuracy has the following definition:

   Accuracy = Number of correct predictions / Total number of predictions.

$$\text{Accuracy} = \frac{TP + TN}{TP + FP + FN + TP}$$

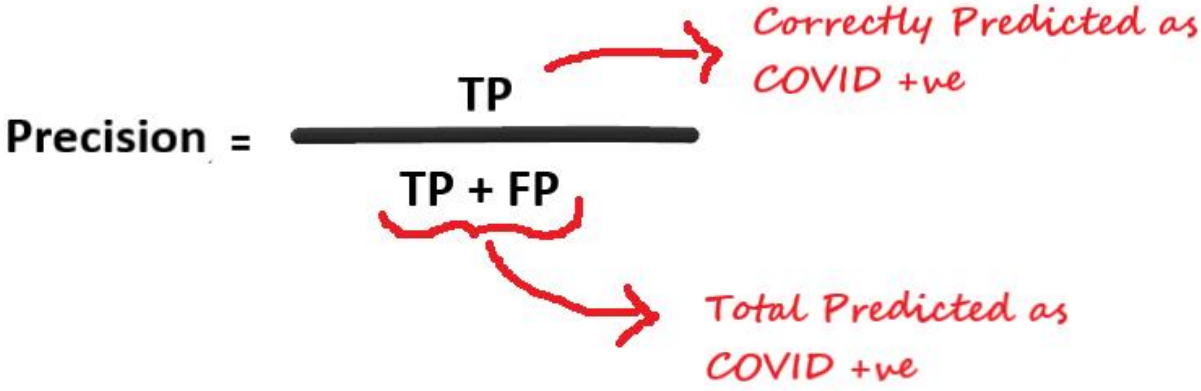
2) Recall (Sensitivity or True positive rate):

Recall gives the fraction you correctly identified as positive out of all positives.



3) Precision:

Precision gives the fraction of correctly identified as positive out of all predicted as positives.



4) F1 Score:
   It is defined as the harmonic mean of the model's precision and recall.

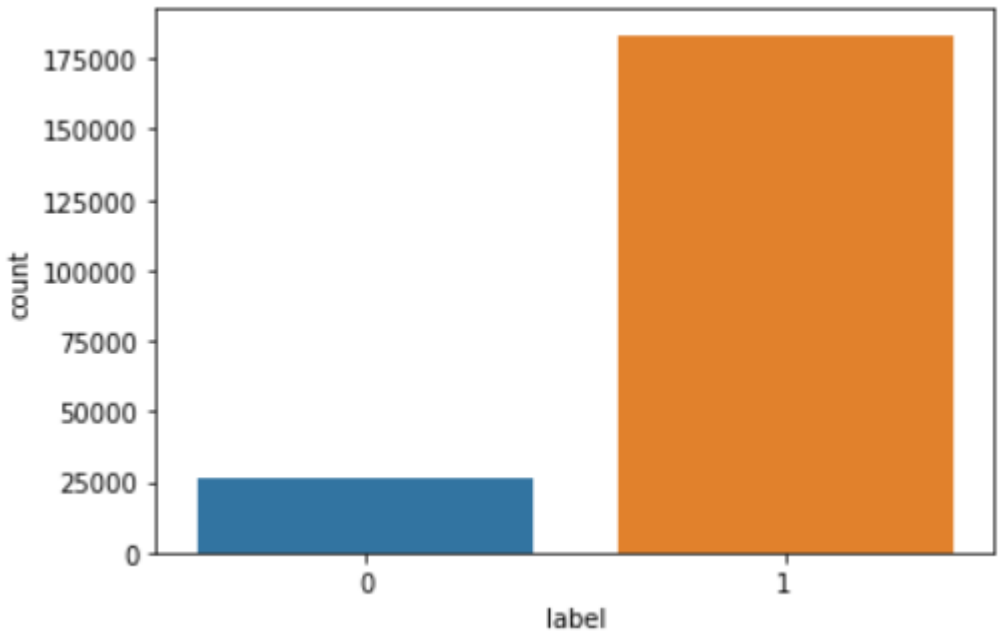
$$\text{F1 Score} = 2 * \frac{\text{Precision} * \text{Recall}}{\text{Precision} + \text{Recall}}$$

5) ROC/AUC Curve:
   The receiver operator characteristic is another common tool used for evaluation. It plots out the sensitivity and specificity for every possible decision rule cut off between 0 and 1 for a model. For classification problems with probability outputs, a threshold can convert probability outputs to classifications
   A) False Positive Rate: Fraction of negative instances that are incorrectly classified as positive.
   B) True Positive Rate: Fraction of positive instances that are correctly predicted as positive.

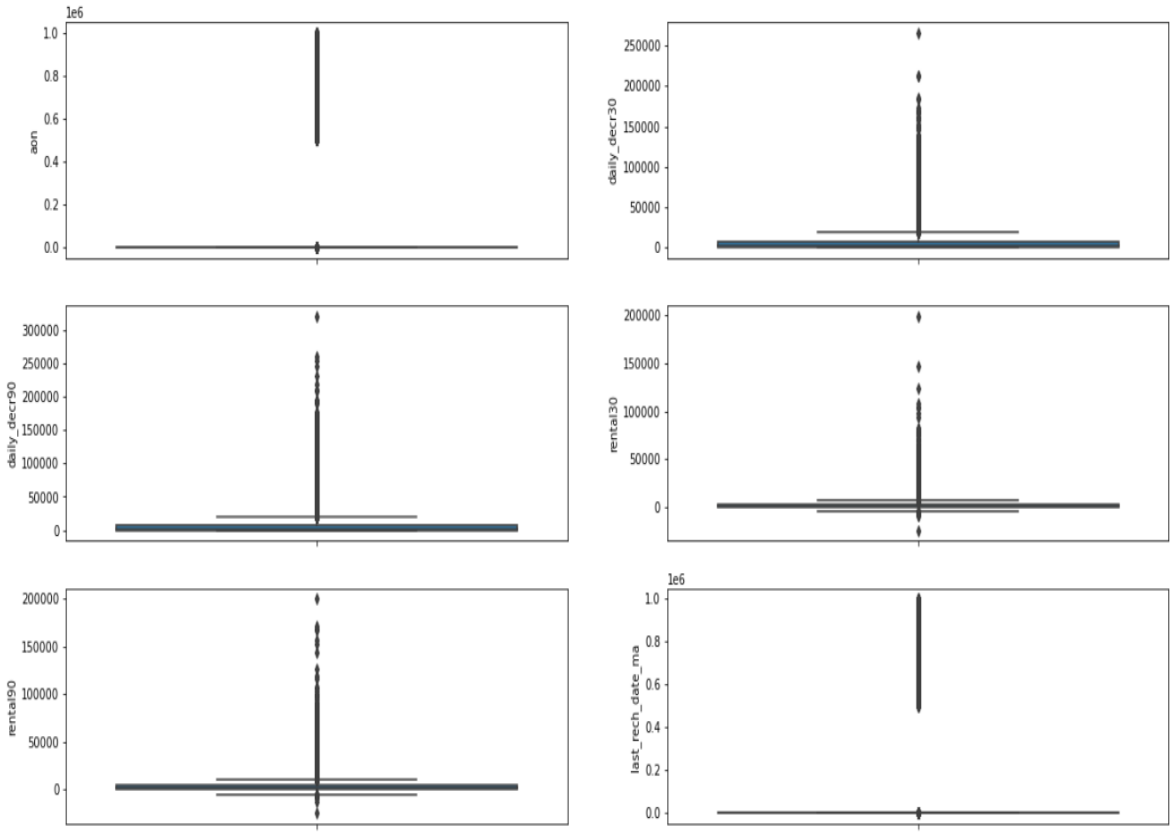
- Visualizations

  Data unbalance ->

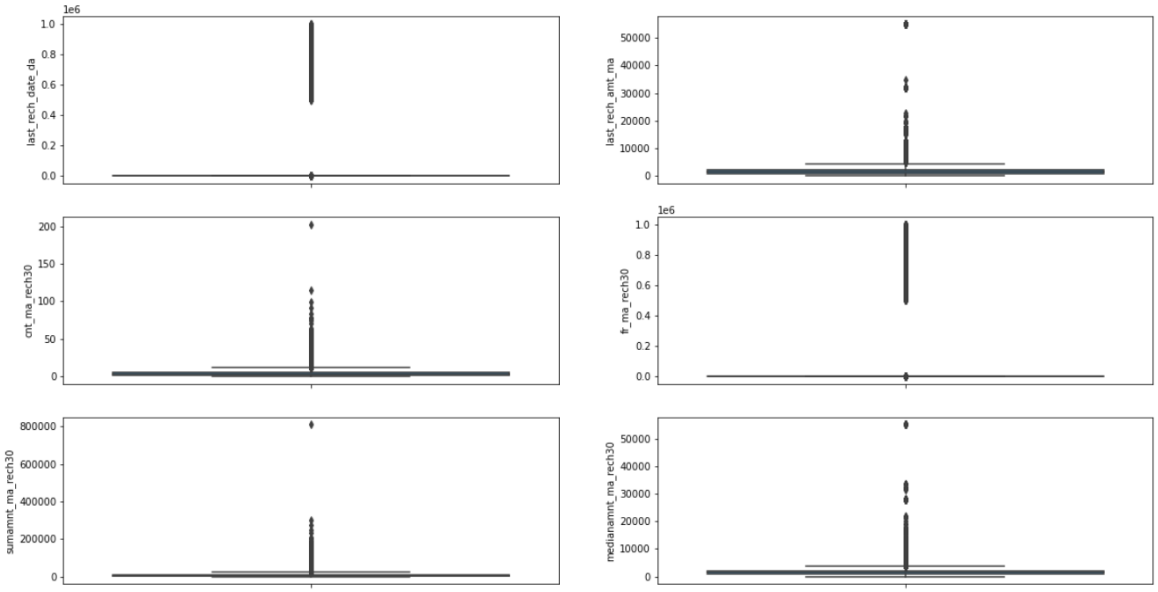
  Below graph shows that the target0 column 'label' is highly unbalanced as in label column we have 25000 -> 0 values and 175000 -> 1 values. Before training the model we will be required to resolved this unbalance issue.



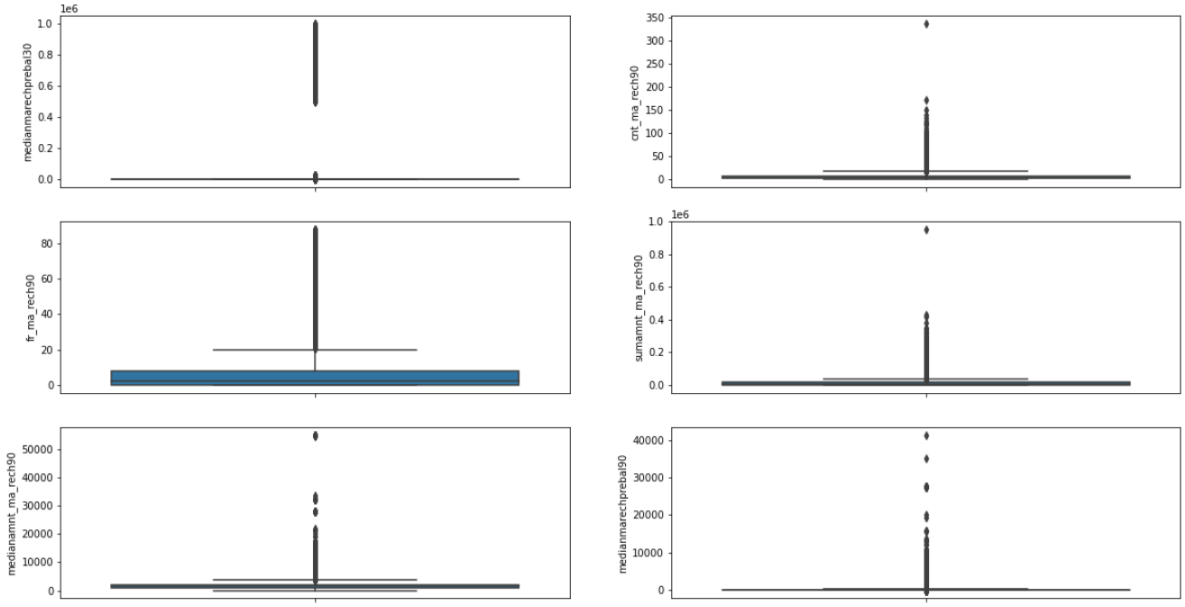
Checking Outliers ->



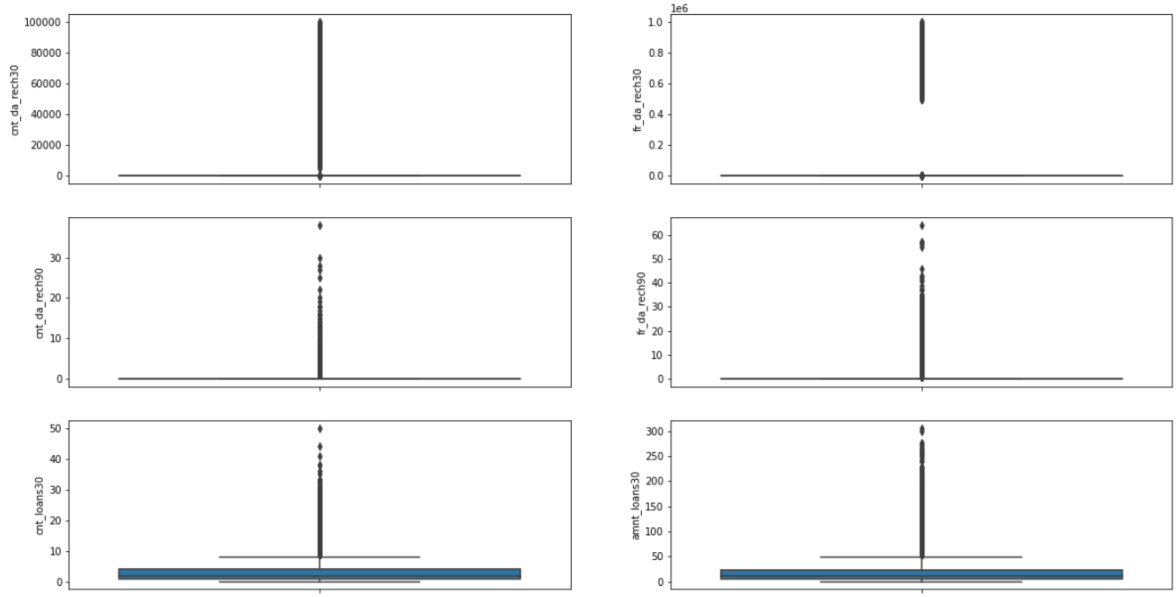
The columns aon, daily_decr30, daily_decr90, rental30, rental90, last_rech_date_ma have large numbers of outliers present in them. Before training the model we will be required to remove the outliers to increase the model accuracy.



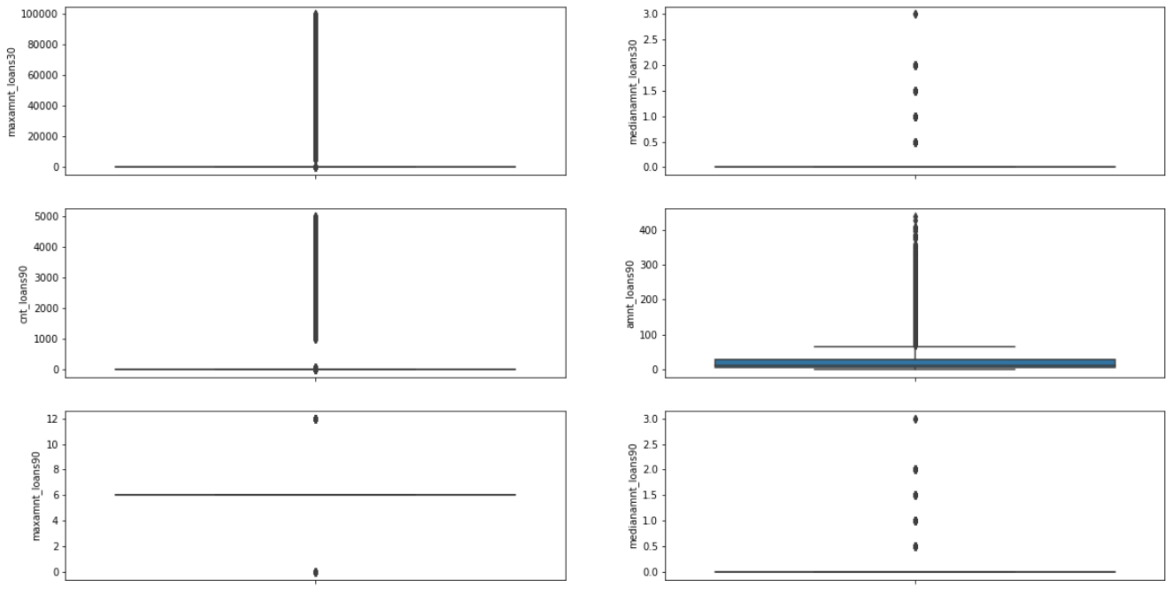
The columns last_rech_date_da, last_rech_amt_ma, cnt_ma_rech30, tr_ma_rech30, sumamnt_ma_rech30, medianamnt_ma_rech30 have large numbers of outliers present in them. Before training the model we will be required to remove the outliers to increase the model accuracy.



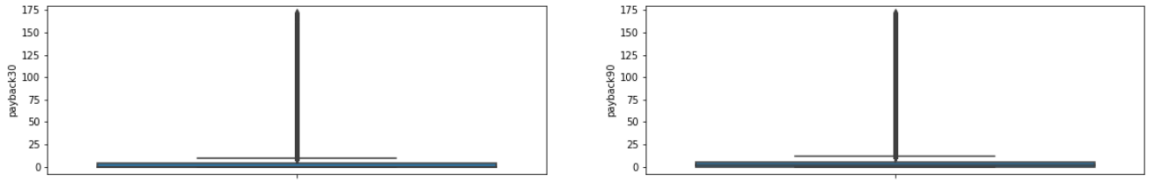
The columns medianmarechprebal30, cnt_ma_rech90, fr_ma_rech90, sumamnt_ma_rech90, medianamnt_ma_rech90, medianmarechprebal90'have large numbers of outliers present in them. Before training the model, we will be required to remove the outliers to increase the model accuracy.



The columns cnt_da_rech30, fr_da_rech30, cnt_da_rech90, fr_da_rech90, cnt_loans30, amnt_loans30 large numbers of outliers present in them. Before training the model, we will be required to remove the outliers to increase the model accuracy.

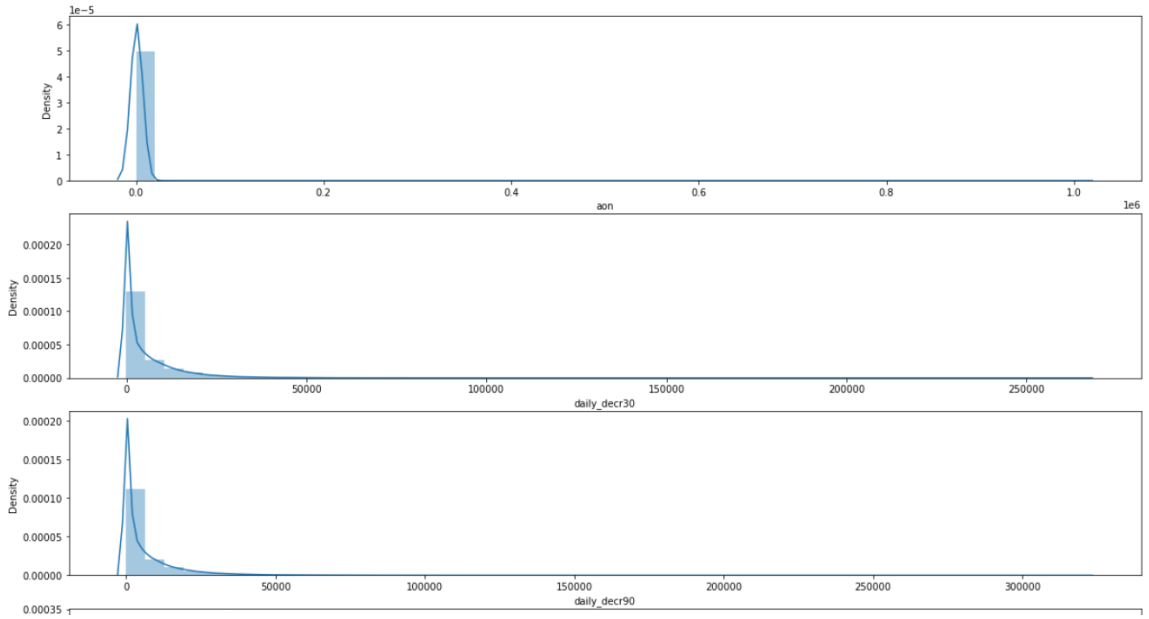


The columns maxamnt_loans30, medianamnt_loans30, cnt_loans90, amnt_loans90, maxamnt_loans90, medianamnt_loans90 large numbers of outliers present in them. Before training the model, we will be required to remove the outliers to increase the model accuracy.

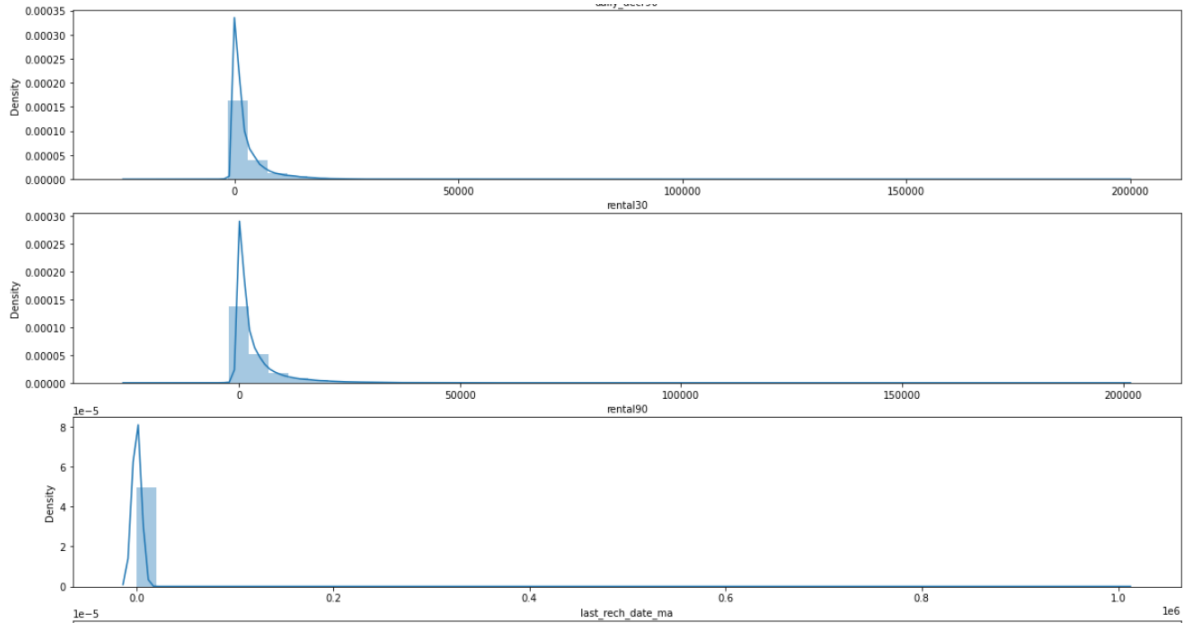


The columns payback30, payback90 large numbers of outliers present in them. Before training the model, we will be required to remove the outliers to increase the model accuracy.

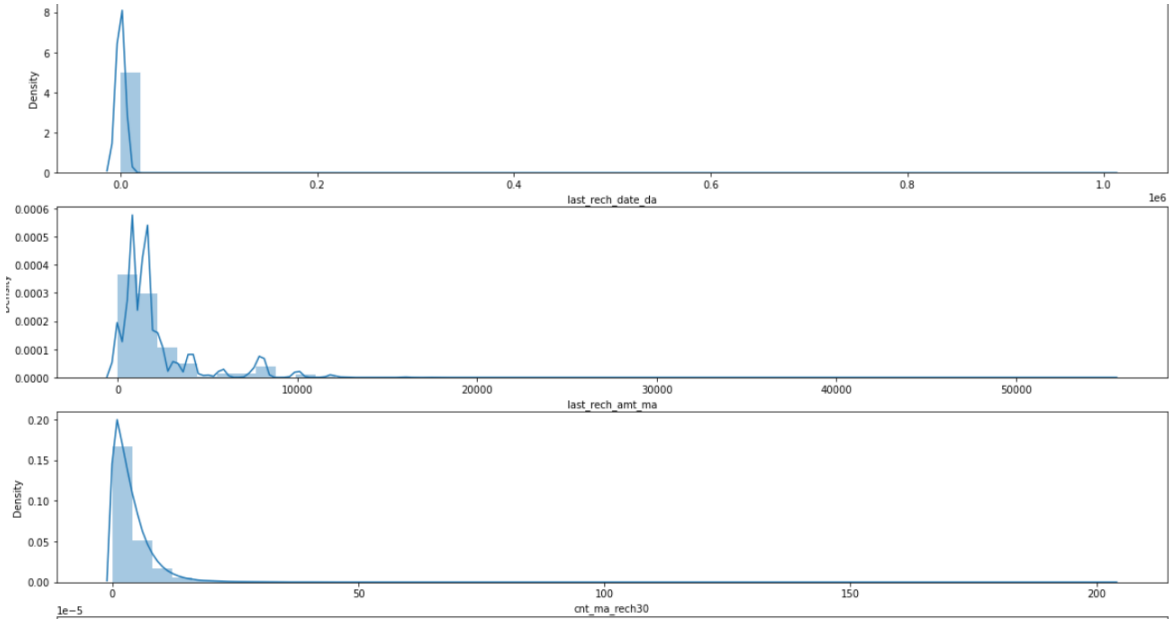
Checking whether the data is normally distributed->



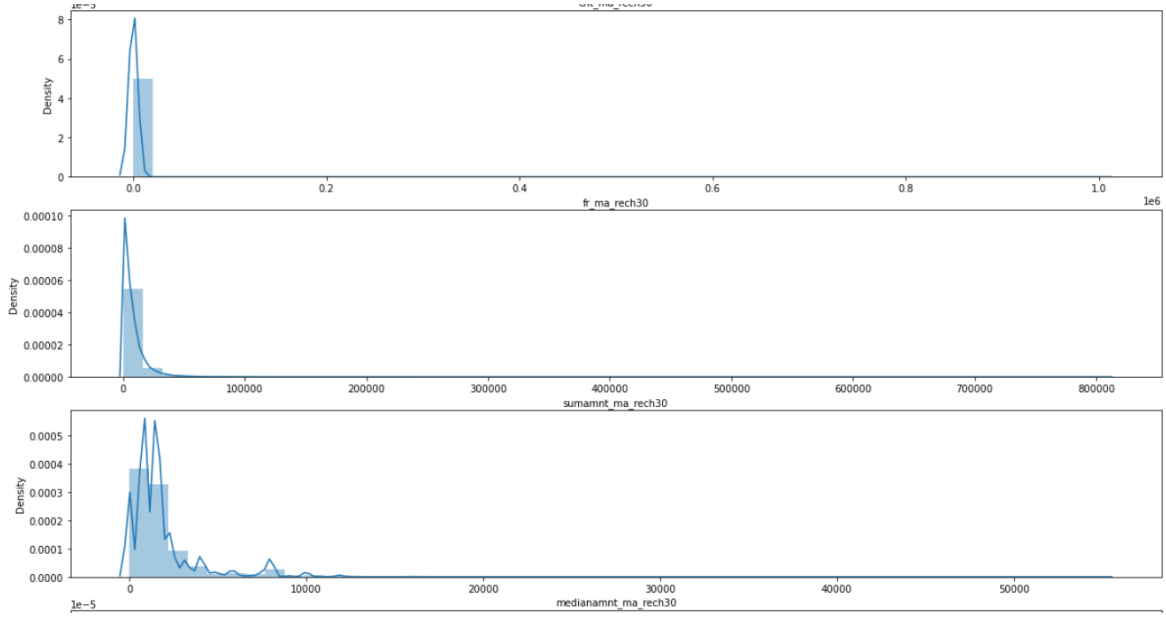
The 'aon', 'daily_decr30', 'daily_decr90', columns are normally distributed.



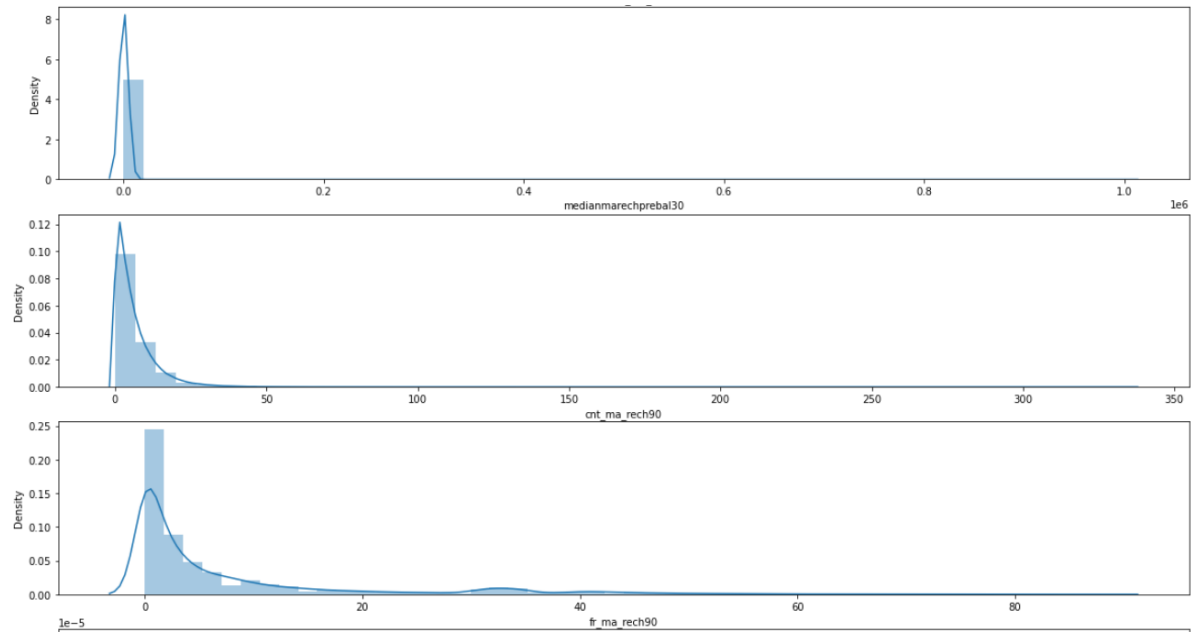
The columns 'rental30', 'rental90', 'last_rech_date_ma' is normally distributed



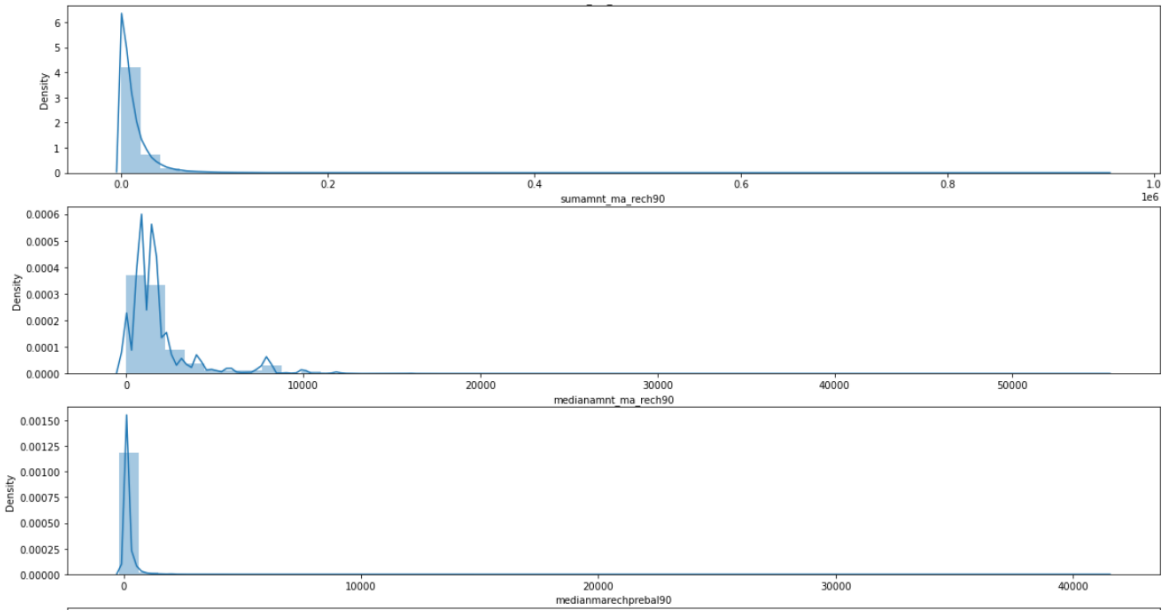
The column 'last_rech_date_da', 'cnt_ma_rech30' is normally distributed whereas 'last_rech_amt_ma', is not normally distributed which we would be required to correct before building the model to increase the model accuracy and precision.



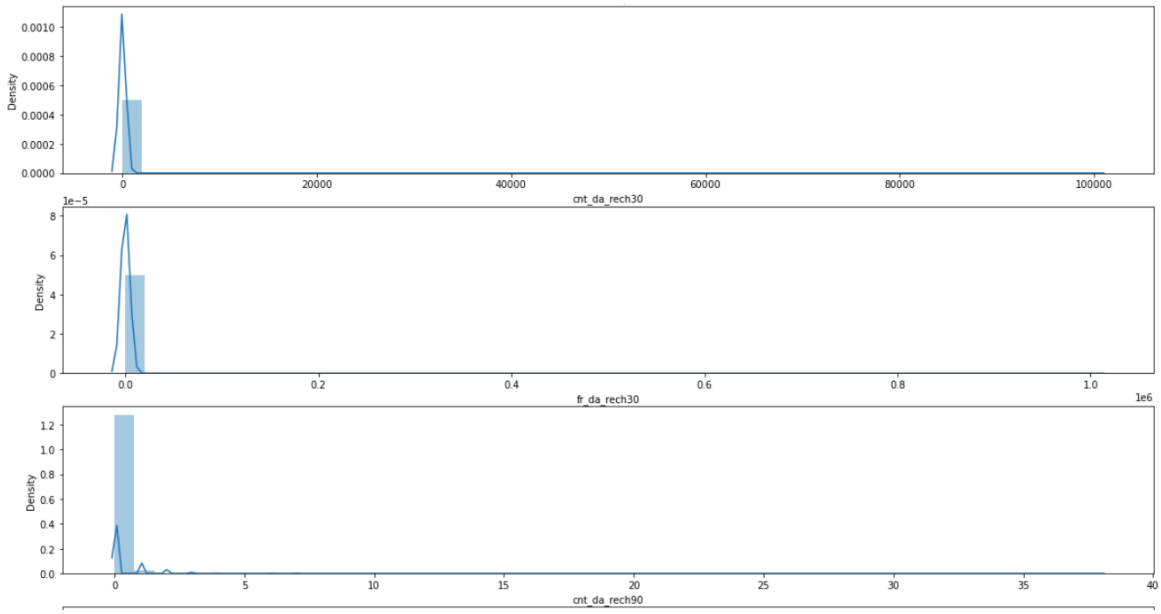
The column 'fr_ma_rech30', 'sumamnt_ma_rech30', is normally distributed whereas 'medianamnt_ma_rech30', is not normally distributed which we would be required to correct before building the model to increase the model accuracy and precision.



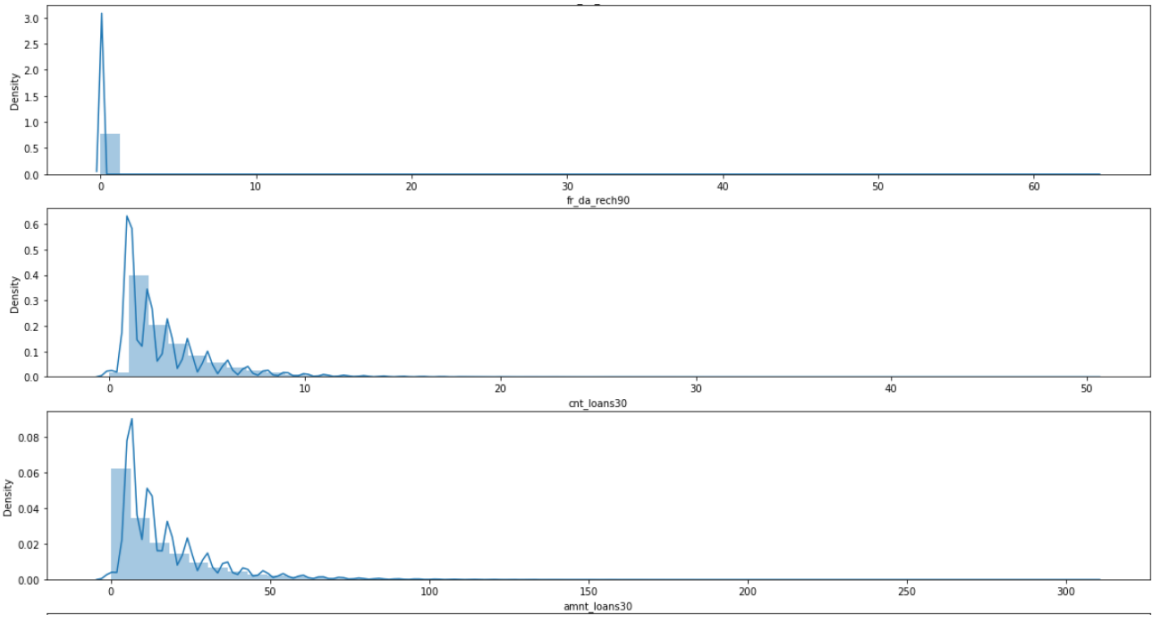
The column 'medianmarechprebal30', 'cnt_ma_rech90', 'fr_ma_rech90' are normally distributed



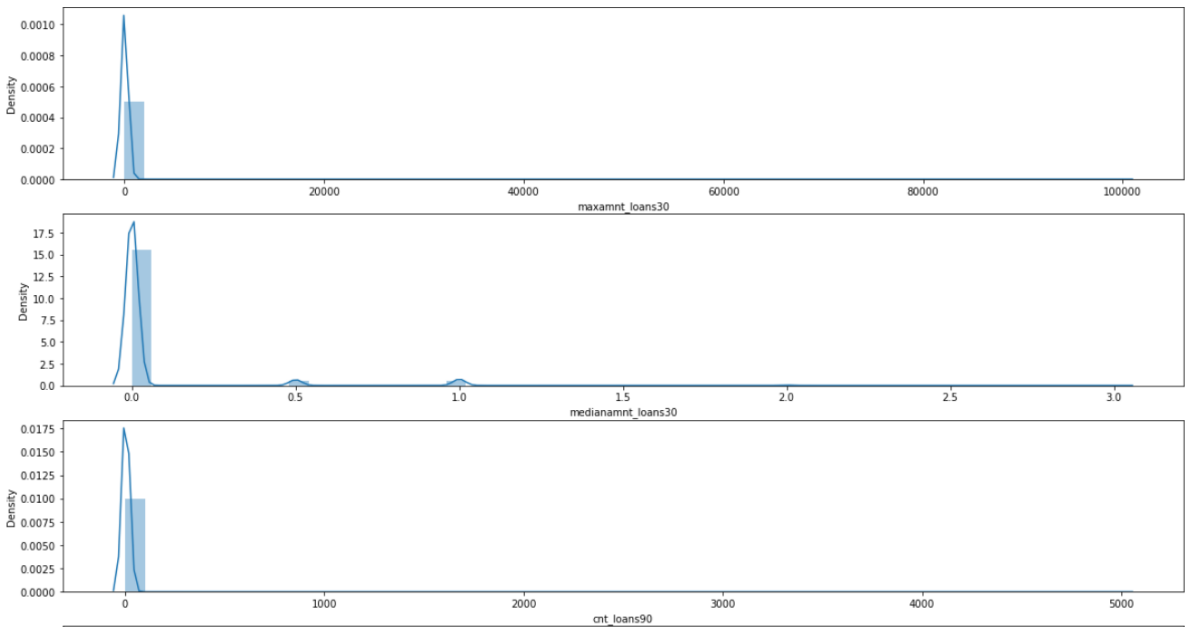
The 'sumamnt_ma_rech90', 'medianmarechprebal90' is normally distributed whereas 'medianamnt_ma_rech90' is not normally distributed which we would be required to correct before building the model to increase the model accuracy and precision.



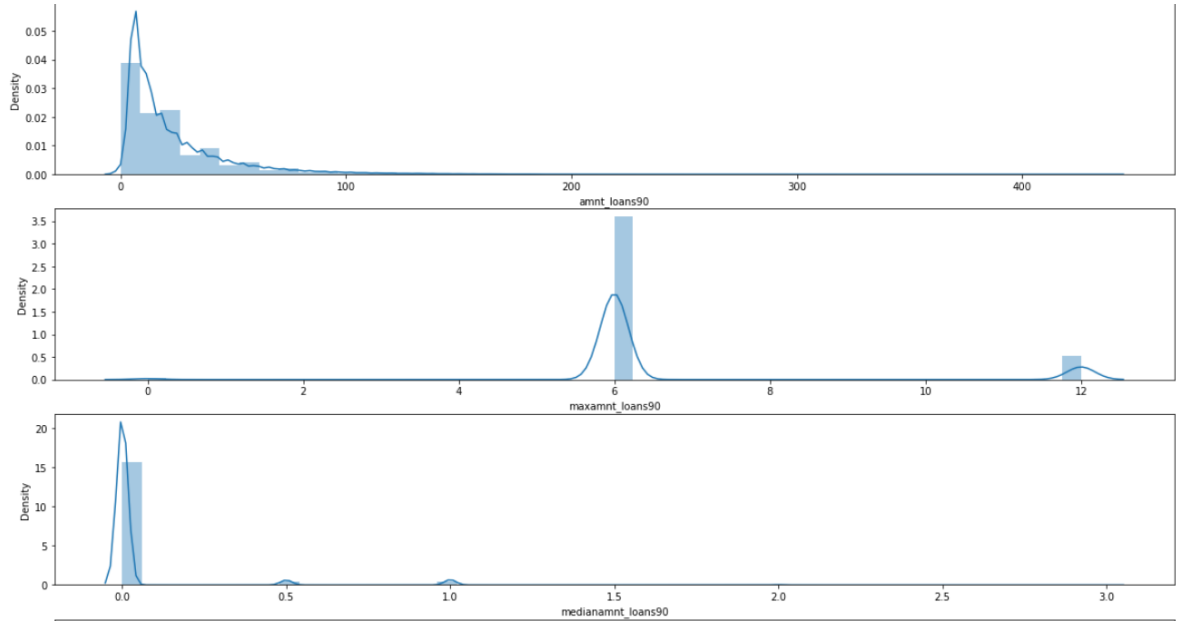
would be required to correct before building the model to increase the model accuracy and precision.



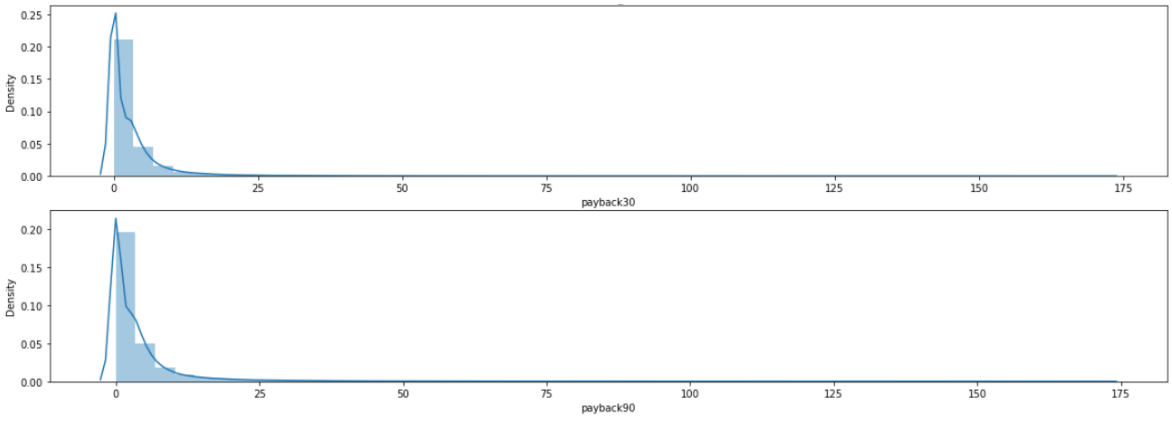
The 'fr_da_rech90', normally distributed whereas ‘cnt_loans30', 'amnt_loans30' is not normally distributed which we would be required to correct before building the model to increase the model accuracy and precision.



The 'maxamnt_loans30', 'medianamnt_loans30', 'cnt_loans90', are normally distributed.



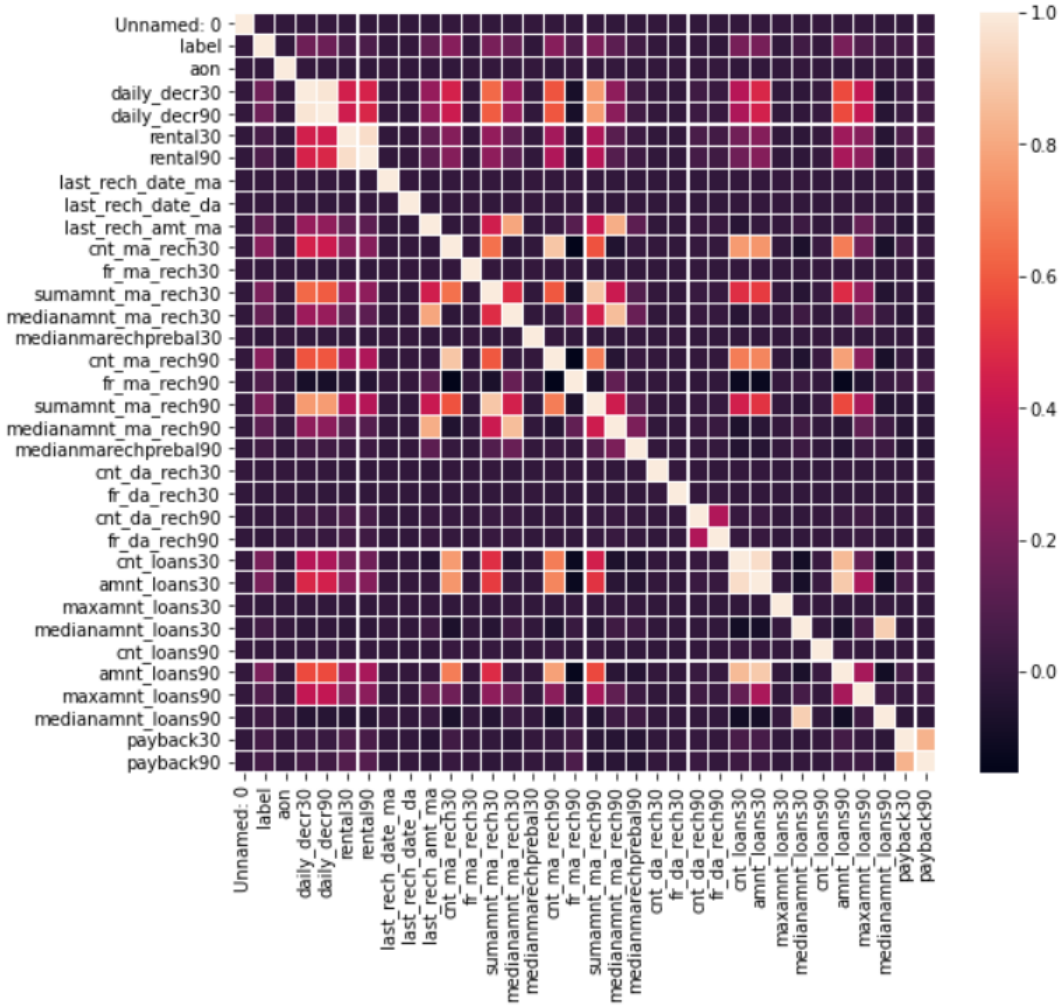
The column 'medianamnt_loans90' is normally distributed whereas 'amnt_loans90', 'maxamnt_loans90', is not normally distributed which we would be required to correct before building the model to increase the model accuracy and precision.



The column 'payback30',

'payback90' are normally distributed

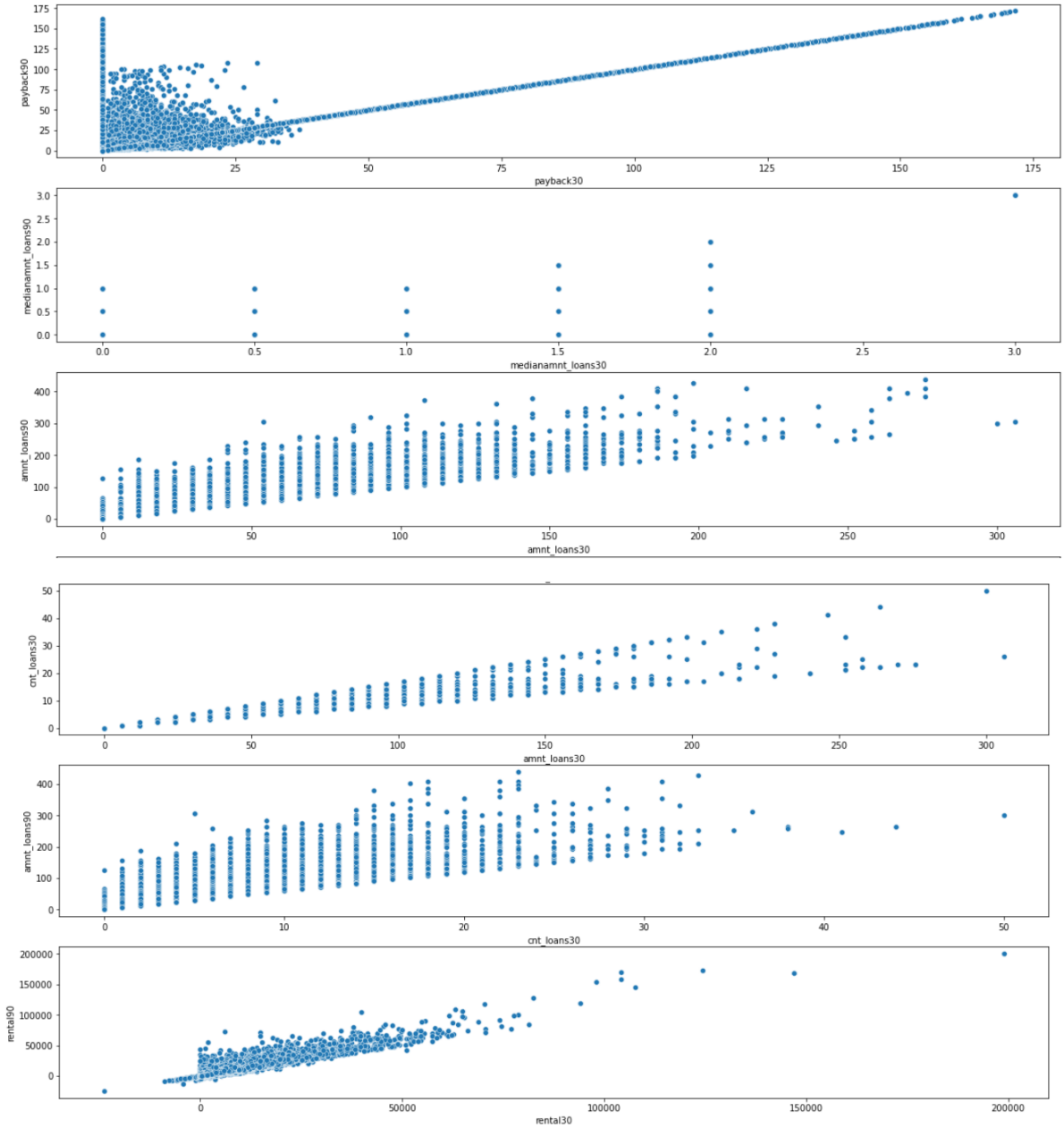
Finding correlation between columns using HEAT MAP->

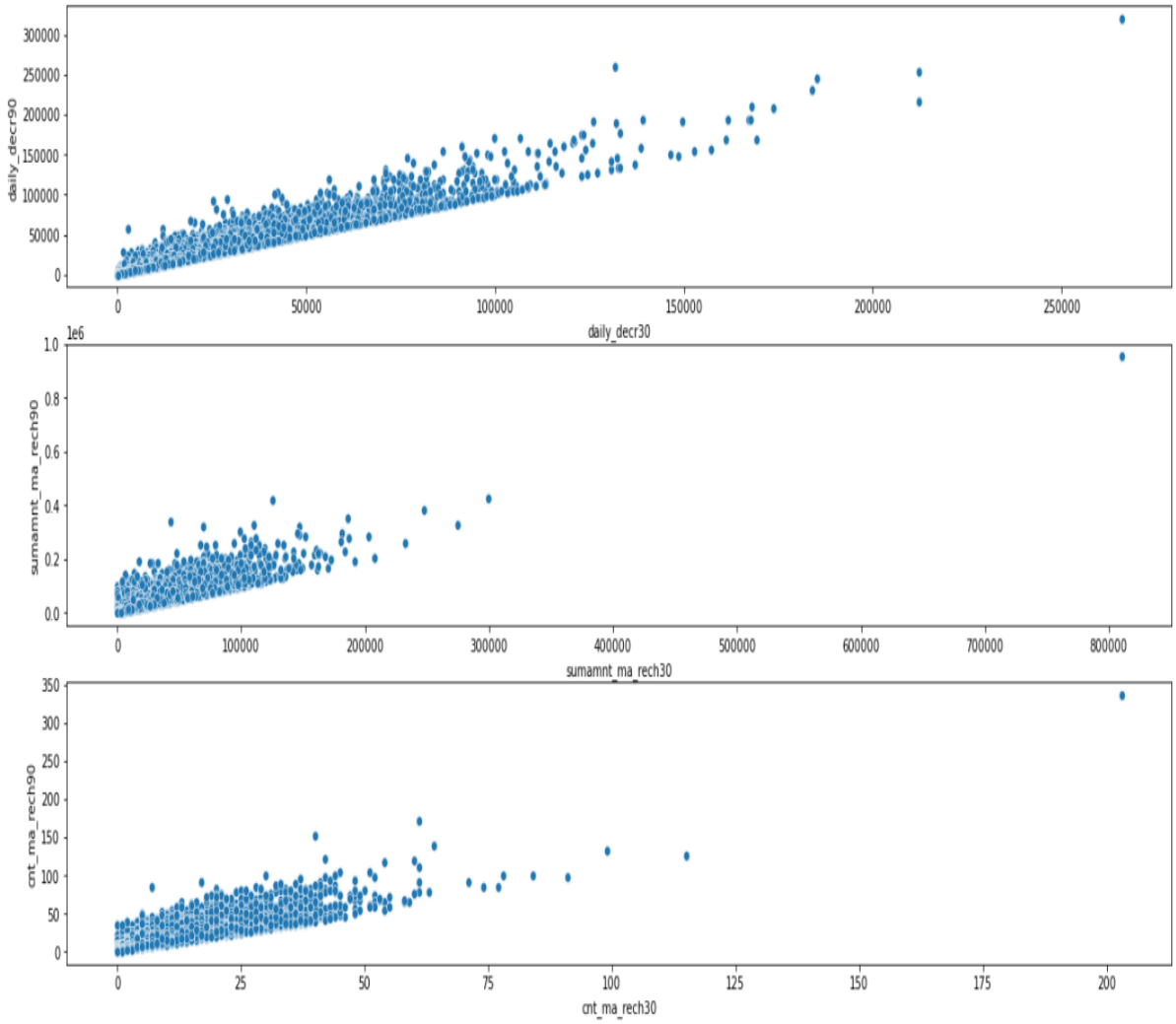


We can observe ->

a) amnt_loans30 is highly corelated with cnt_loans30
b) rental30 is highly correlated with rental 90
c) daily_decr30 is highly correlated with daily_decr90
d) Other columns are loosely correlated with each other.

Plotting the correlation between columns->





From above graphs we can predict that Linear Relationship is present between the attributes

cols1=['payback30','medianamnt_loans30','amnt_loans30','amnt_loans30','cnt_loans30','rental30','daily_decr30','sumamnt_ma_rech30','cnt_ma_rech30']

cols2=['payback90','medianamnt_loans90','amnt_loans90','cnt_loans30','amnt_loans90','rental90','daily_decr90','sumamnt_ma_rech90','cnt_ma_rech90']

Respectively

- Interpretation of the Results

  Results:

  1) No null values or repeated rows are present in dataset
  2) Target Variable is highly unbalanced
  3) Dataset have outliers in all variables
  4) Dataset is not normalized
  5) Dataset is highly skewed
  6) amnt_loans30 is highly corelated with cnt_loans30
  7) rental30 is highly correlated with rental 90
  8) daily_decr30 is highly correlated with daily_decr90
  9) Random Forest Algorithm is best suited for the current dataset

CONCLUSION

- ## Key Findings and Conclusions of the Study

  We found that to predict the Micro Credit Defaulter using Data Science the best way after performing Data Cleaning is to use Random Forest Algorithm it provides 91% accuracy which is better than other Classification algorithms.

- ## Learning Outcomes of the Study in respect of Data Science

  In data science, there are various steps involved during Data analysis and cleaning. With the help of various Visualization tools like plots, Graphs we were able to perform the actions and observe different things. Like for finding the outliers we used Box Plot visualization, for finding the skewness and normalization we used Count Plot visualization, for finding skewness we visualized the skewness using Heat Map for the clear picture of how the variables are co-related to each other in the dataset. We used AUC ROC curve to check which model best fits the prediction for the dataset.

  In the data set the target variable was not balanced. So, the challenge was to balance the variable, one way is to remove the extra data from the target variable but as the data was costly so the loss of data cannot be afforded. so, we did some research and found the method where we can create the Sample data and balance the target variable. we used the SMOTE technique to balance the data.

- ## Limitations of this work and Scope for Future Work

  Data was unbalanced if data was balanced more accurate and clear picture of the output -> result is dependent on the data

  Neural network classifier which are still unexplored & can be taken for future consideration