

MICRO CREDIT DEFAULTER MODEL

Submitted by:

SHAHBAZ ALAM

**ACKNOWLEDGMENT**

I would specially like to thank DataTrained team for teaching me Data Science & Machine Learning. I would also like to thank FlipRobo team for this opportunity to work in this insightful project of classifying Micro Credit Defaulters and their kind support during the internship.

Secondly, I would like to thank the online coding communities like stackoverflow.com which are always guiding all the learners.

**INTRODUCTION**

* Business Problem Framing

Our Client (a telecom operator) in collaboration with a Microfinance Institute (MFI) provides loans of value 6 and 12 (Indonesian Rupiah) to the network users. These loans are risky for our client as the loan is being provided to low income populations. Therefore it is imperative for this business to classify all potential defaulters in order to minimize business risk and avoid losses.

* Conceptual Background of the Domain Problem

Mobile financial services (MFS) are a very lucrative business as the returns are high but there is considerable risk of default involved. In our specific application, the telecom company in collaboration with a Microfinance Institute (MFI) provides loans of amount 5 and 10 (Indonesian Rupiah) for a very short period and the payback amount is 6 and 12 (Indonesian Rupiah) respectively which corresponds to a high interest rate of 20% in a very short period (usually 5 days). While the return is high, there is considerable risk of default involved, because the loan is being provided to low income populations.

Therefore it is necessary to classify all the defaulters to minimize business risk and avoid losses. The sample data is provided to us from our client database to classify defaulters which would help them in further investment and improvement in selection of customers.

We will use machine learning classification algorithms to predict the defaulters based on the sample data provided by the client.

* Review of Literature

We will begin our project with the sample dataset which contains loan default status along with associated features. We will look at all the features with following goals in mind:

1. Relevance of the feature
2. Distribution of the feature
3. Cleaning the feature
4. Visualization of the feature
5. Visualization of the feature as per loan default status for data analysis

After having gone through all the features and cleaning the dataset, we will move on to machine learning classification modelling:

1. Pre-processing the dataset for models
2. Testing multiple algorithms with multiple evaluation metrics
3. Select evaluation metric as per our specific business application
4. Hyper-parameter tuning using GridSearchCV for the best model parameter settings

* Motivation for the Problem Undertaken

The objective of the project is to prepare a model based on the sample dataset that classifies all loan defaulters and help our client in further investment and improvement in selection of customers.

**Analytical Problem Framing**

* Mathematical/ Analytical Modeling of the Problem

The distribution of the all features was studied and inferences are drawn for each feature as per user behaviour. The boxplot of all the features were studied as per default status which helped us in drawing inferences about the tendencies of defaulters. Multiple classification algorithms were tested for the sample data with multiple evaluating metrics to observe their performance. Finally hyper parameter tuning was performed for the best model with GridSearchCV.

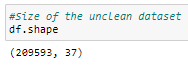
* Data Sources and their formats

The dataset is being provided by the client from their database, it was collected from June 2016 to August 2016 by the telecom operator. The dataset has following features:

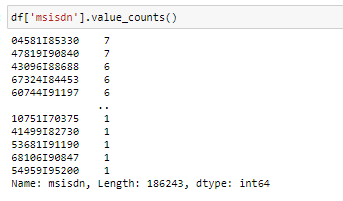
|  |  |  |
| --- | --- | --- |
| **S. No** | **Variable** | **Definition** |
| 1 | label | Flag indicating whether the user paid back the credit amount within 5 days of issuing the loan{1:success, 0:failure} |
| 2 | msisdn | mobile number of user |
| 3 | aon | age on cellular network in days |
| 4 | daily\_decr30 | Daily amount spent from main account, averaged over last 30 days (in Indonesian Rupiah) |
| 5 | daily\_decr90 | Daily amount spent from main account, averaged over last 90 days (in Indonesian Rupiah) |
| 6 | rental30 | Average main account balance over last 30 days |
| 7 | rental90 | Average main account balance over last 90 days |
| 8 | last\_rech\_date\_ma | Number of days till last recharge of main account |
| 9 | last\_rech\_date\_da | Number of days till last recharge of data account |
| 10 | last\_rech\_amt\_ma | Amount of last recharge of main account (in Indonesian Rupiah) |
| 11 | cnt\_ma\_rech30 | Number of times main account got recharged in last 30 days |
| 12 | fr\_ma\_rech30 | Frequency of main account recharged in last 30 days |
| 13 | sumamnt\_ma\_rech30 | Total amount of recharge in main account over last 30 days (in Indonesian Rupiah) |
| 14 | medianamnt\_ma\_rech30 | Median of amount of recharges done in main account over last 30 days at user level (in Indonesian Rupiah) |
| 15 | medianmarechprebal30 | Median of main account balance just before recharge in last 30 days at user level (in Indonesian Rupiah) |
| 16 | cnt\_ma\_rech90 | Number of times main account got recharged in last 90 days |
| 17 | fr\_ma\_rech90 | Frequency of main account recharged in last 90 days |
| 18 | sumamnt\_ma\_rech90 | Total amount of recharge in main account over last 90 days (in Indonesian Rupiah) |
| 19 | medianamnt\_ma\_rech90 | Median of amount of recharges done in main account over last 90 days at user level (in Indonesian Rupiah) |
| 20 | medianmarechprebal90 | Median of main account balance just before recharge in last 90 days at user level (in Indonesian Rupiah) |
| 21 | cnt\_da\_rech30 | Number of times data account got recharged in last 30 days |
| 22 | fr\_da\_rech30 | Frequency of data account recharged in last 30 days |
| 23 | cnt\_da\_rech90 | Number of times data account got recharged in last 90 days |
| 24 | fr\_da\_rech90 | Frequency of data account recharged in last 90 days |
| 25 | cnt\_loans30 | Number of loans taken by user in last 30 days |
| 26 | amnt\_loans30 | Total amount of loans taken by user in last 30 days |
| 27 | maxamnt\_loans30 | maximum amount of loan taken by the user in last 30 days |
| 28 | medianamnt\_loans30 | Median of amounts of loan taken by the user in last 30 days |
| 29 | cnt\_loans90 | Number of loans taken by user in last 90 days |
| 30 | amnt\_loans90 | Total amount of loans taken by user in last 90 days |
| 31 | maxamnt\_loans90 | maximum amount of loan taken by the user in last 90 days |
| 32 | medianamnt\_loans90 | Median of amounts of loan taken by the user in last 90 days |
| 33 | payback30 | Average payback time in days over last 30 days |
| 34 | payback90 | Average payback time in days over last 90 days |
| 35 | pcircle | telecom circle |
| 36 | pdate | date |

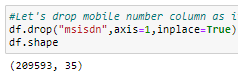
* Data Pre-processing Done

We have to clean the dataset as it contains many garbage values and much skewed data which needs to be cleaned for model but we also have a constraint that we do not lose too much data as data is valuable. The standard process is to remove all data points which have a z score greater than 3 but if we do that we will lose a significant fraction of data. So, we’ll use visualization and consider the number of data points lost in removing the outliers.

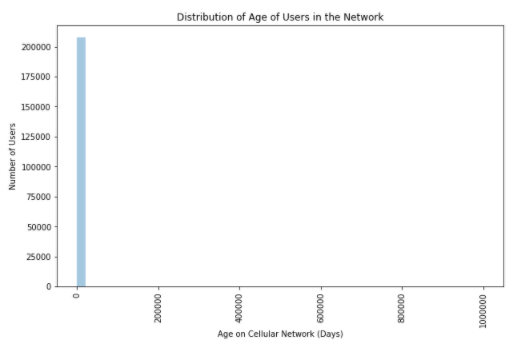
We’ll track the size of the dataset after cleaning each feature and in the end calculate the % of data cases lost after cleaning.

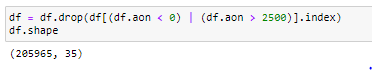
Let’s pre-process all the features:

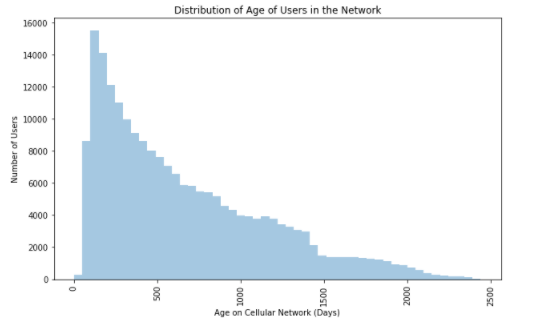
1. Label (Default status): This is the output variable which has the result whether loan was defaulted (0) or not (1). This features doesn’t require any pre-processing.
2. Msisdn (Phone number):

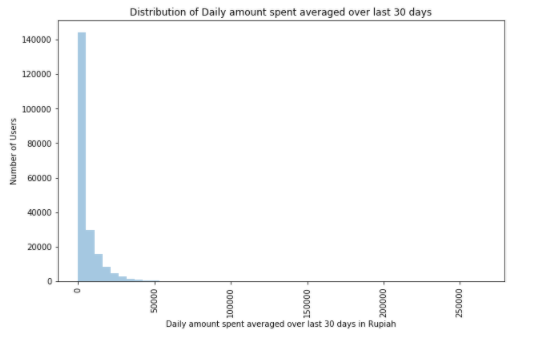
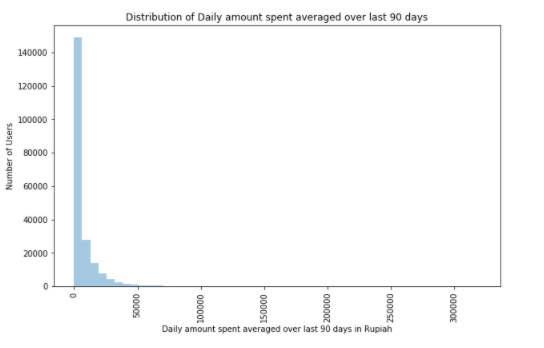
We have a total of 186243 unique mobile numbers or users in the dataset, also no user has more than 7 cases of loan. We will drop the mobile number column for the model as it has no relationship with loan default.

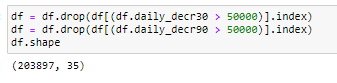
1. Aon (Age on Cellular Network (Days)):

Plot of uncleansed data:

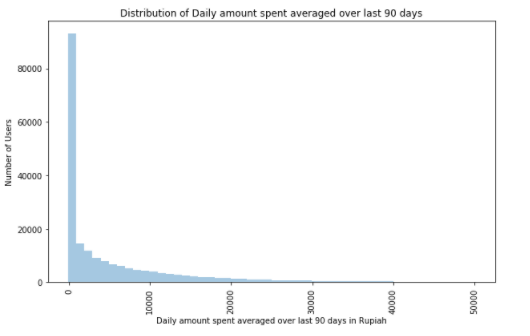
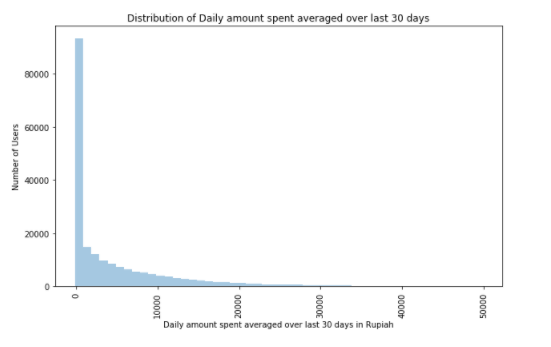
There are many garbage values like very high values and negative values which are not practical (Days), so we will drop all such cases from the dataset:

After cleaning the data:

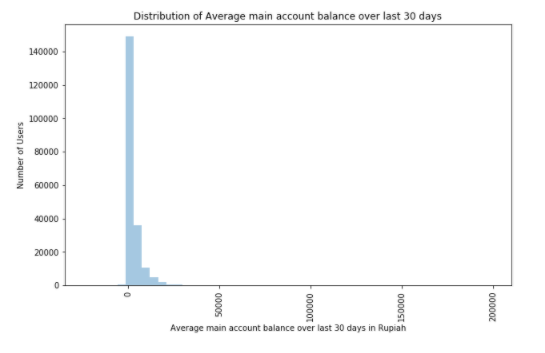
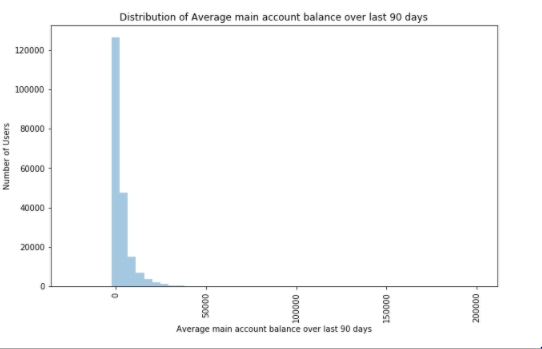
1. daily\_decr30 (daily amount spent averaged over last 30 days) and
2. daily\_decr90 (daily amount spent averaged over last 90 days):

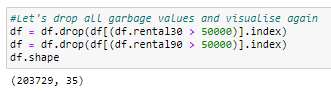
There are many outlier values present in the Daily amount spent averaged over last 30/90 days column i.e. extremely high values, so let's remove them:

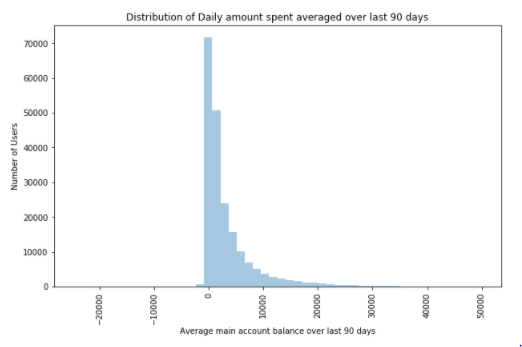
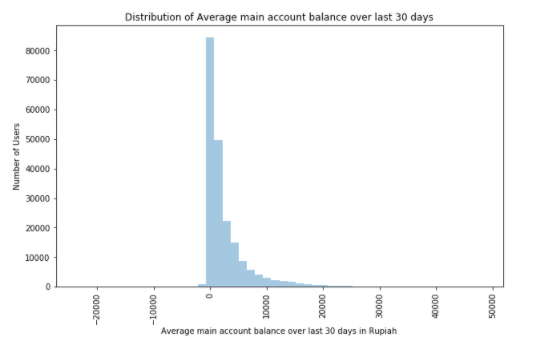
After removing outliers:

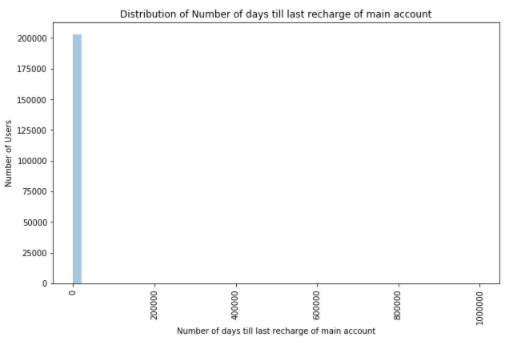
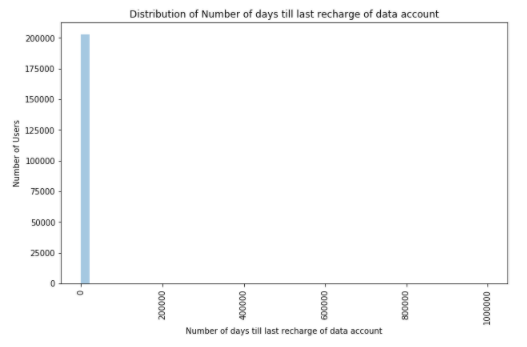


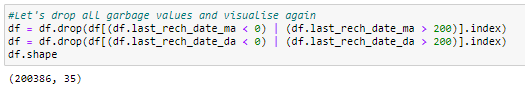
1. rental30 (Average main account balance over last 30 days) and
2. rental90 (Average main account balance over last 90 days):

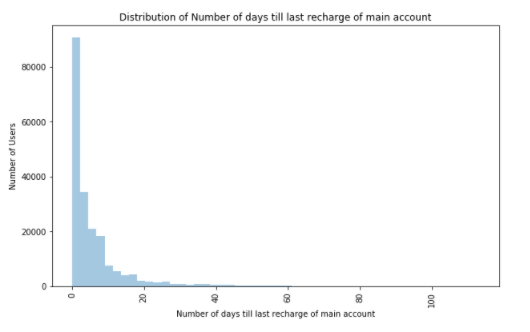


Very high values of Average main account balance is skewing the data let's remove all values over 50000:

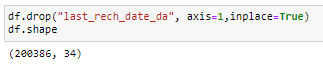
After cleaning the data:

1. last\_rech\_date\_ma (Number of days till last recharge of main account) and
2. last\_rech\_date\_da (Number of days till last recharge of data account):

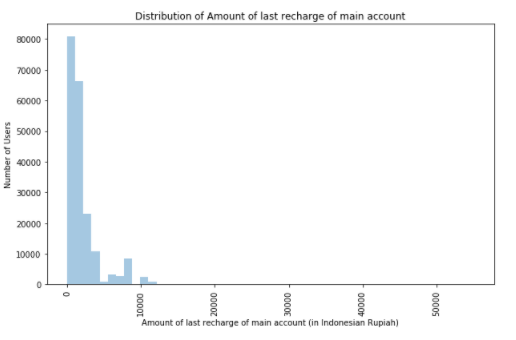
Number of days till last recharge of main/data account columns contain a lot of garbage values i.e. extremely high float numbers and negative numbers which are not practical so let's drop them

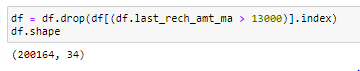
After cleaning the data:

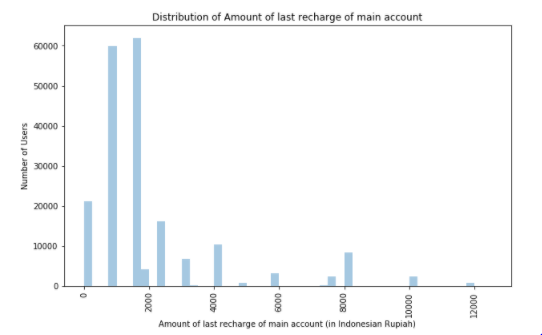


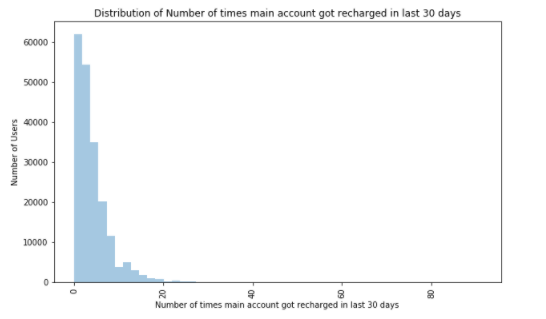
After cleaning the Number of days till last recharge of main/data account, the Number of days till last recharge of data account does not provide any inferences as most of the data points have zero as value, so let's drop it

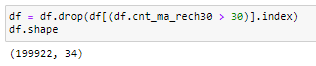
1. last\_rech\_amt\_ma (Amount of last recharge of main account (in Indonesian Rupiah)):

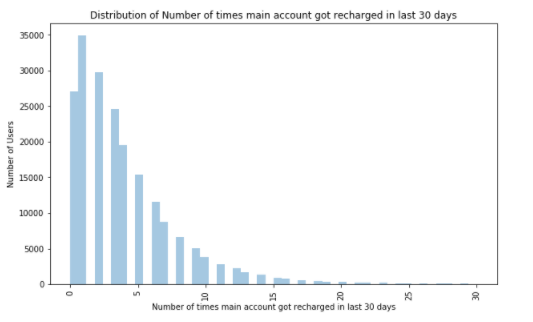
Plot of uncleansed data:

Amount of last recharge of main account (in Indonesian Rupiah) contains many outliers, let's remove all the values over 13000 Rupiah

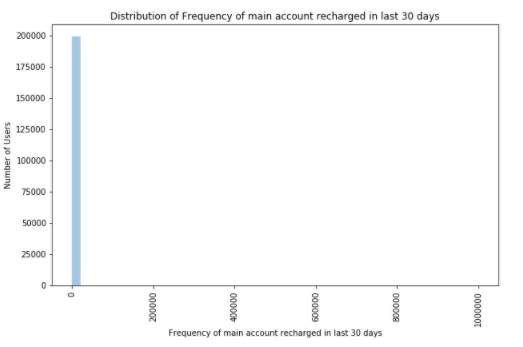
After cleaning the data:

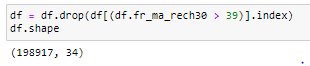
1. cnt\_ma\_rech30 (Number of times main account got recharged in last 30 days):

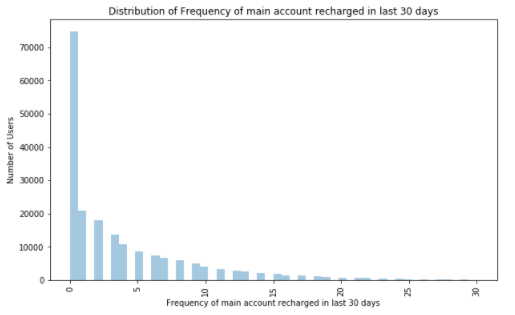
The distribution of Number of times main account got recharged in last 30 days is much skewed, let's remove all values over 30

After cleaning the data:

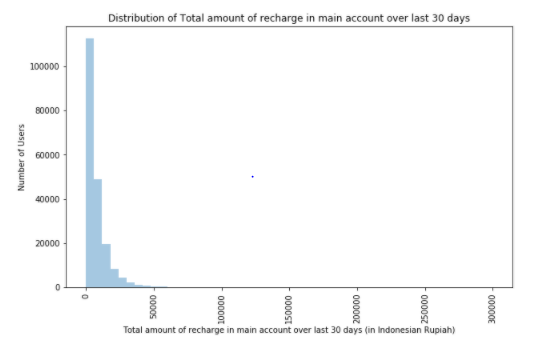
1. fr\_ma\_rech30 (Frequency of main account recharged in last 30 days):

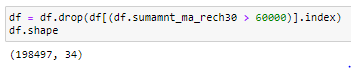
Plot of uncleansed data:

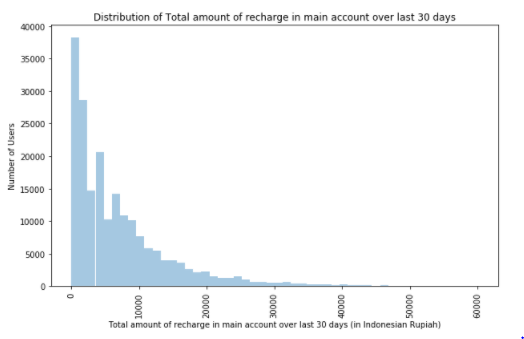
The Frequency of main account recharged in last 30 days has a lot of garbage values i.e. extremely high float numbers, so let's remove them

After cleaning the data:

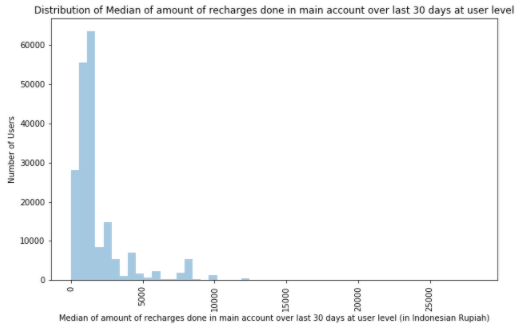
1. sumamnt\_ma\_rech30 (Total amount of recharge in main account over last 30 days (in Indonesian Rupiah)):

Plot of uncleansed data:

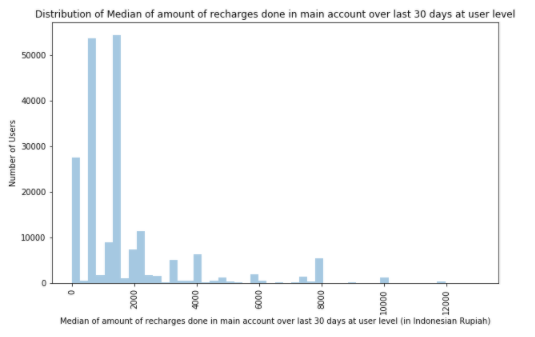
The Total amount of recharge in main account over last 30 days column is much skewed so let's remove the outliers:

After cleaning the data:

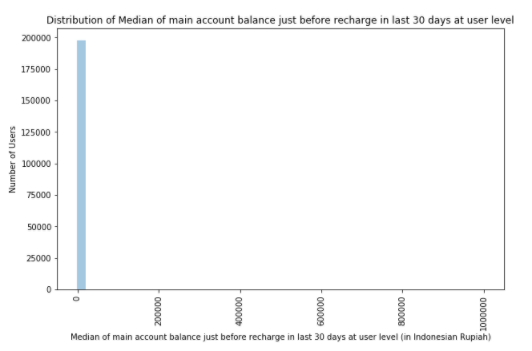
1. medianamnt\_ma\_rech30 (Median of amount of recharges done in main account over last 30 days):

Plot of uncleansed data:

The Median of amount of recharges done in main account over last 30 days at user level column contains a lot of higher values which are skewing the data, so let's drop all entries over 13000.

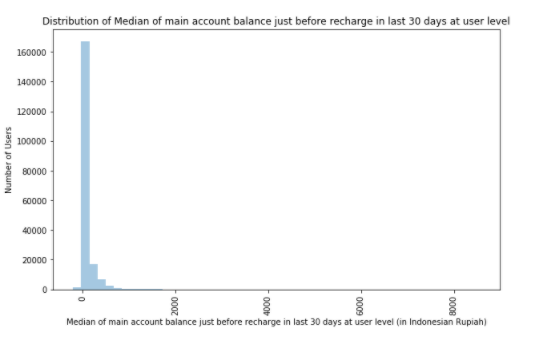
After cleaning the data:

1. medianmarechprebal30 (Median of main account balance just before recharge in last 30 days):

Plot of uncleansed data:

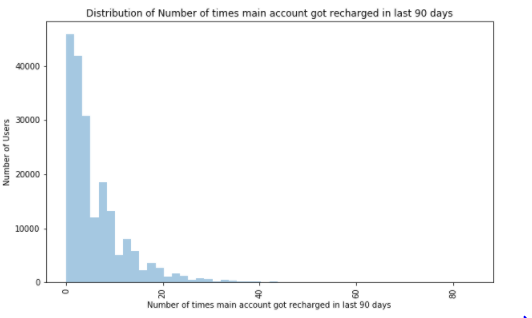
The Median of main account balance just before recharge in last 30 days at user level column contains a lot of higher values which are skewing the data, so let's drop all entries over 10000



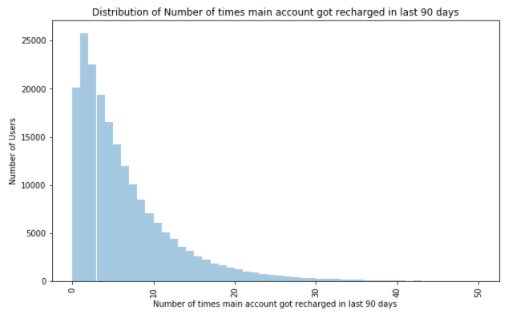
After cleaning the data:

The above plot shows that a very high majority of data points are zero and this feature does not reveal anything, so let's drop it

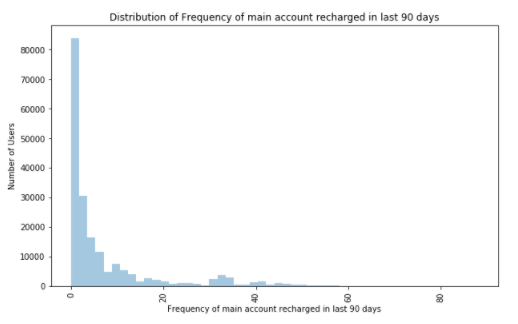
1. cnt\_ma\_rech90 (Number of times main account got recharged in last 90 days):

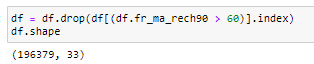
Plot of uncleansed data:

Number of times main account got recharged in last 90 days column contains high outlier values, so let's drop all values over 50:

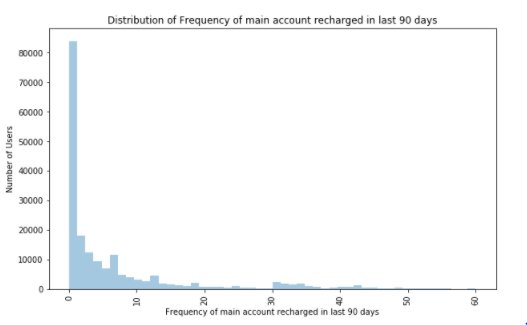
After cleaning the data:

1. fr\_ma\_rech90 (Frequency of main account recharged in last 90 days):

Plot of uncleansed data:

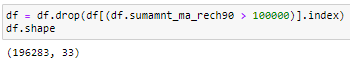
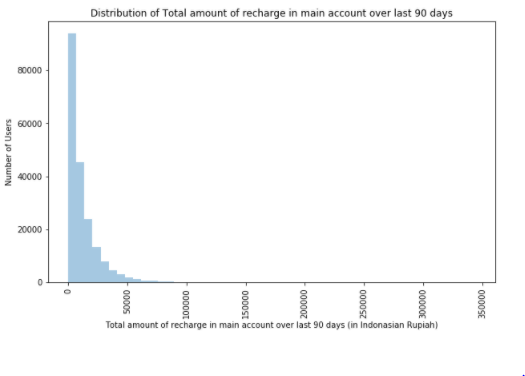
Frequency of main account recharged in last 90 days column contains a lot of high outlier values, so let's drop all values over 60:

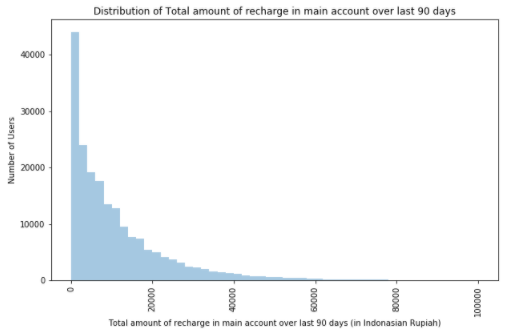
After cleaning the data:



1. sumamnt\_ma\_rech90 (Total amount of recharge in main account over last 90 days):

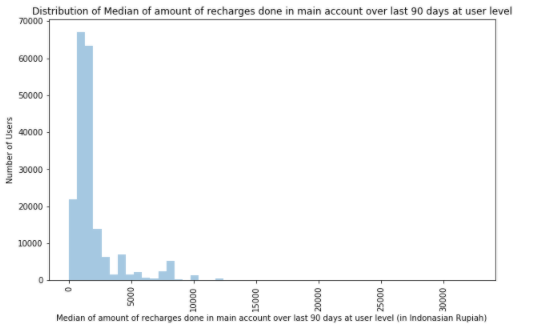
Plot of uncleansed data:

Total amount of recharge in main account over last 90 days column contains a lot of garbage values i.e. extremely high numbers, so let's drop them:

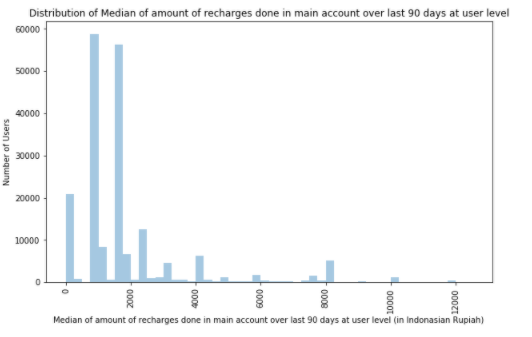
After cleaning the data:

1. medianamnt\_ma\_rech90 (Median of amount of recharges done in main account over last 90 days):

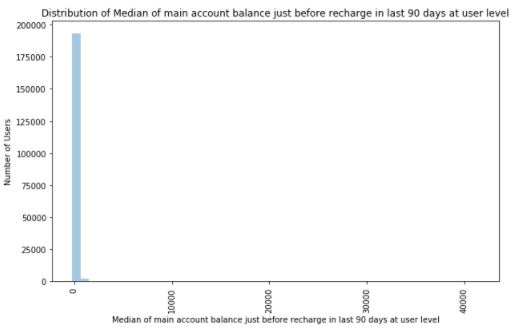
Plot of uncleansed data:



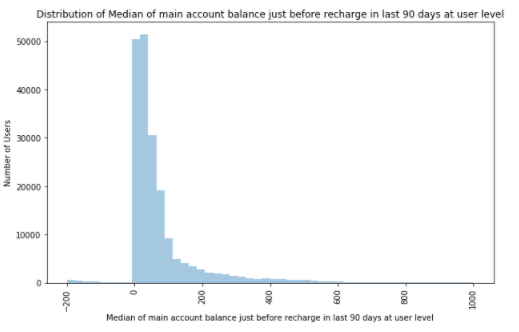
Median of amount of recharges done in main account over last 90 days at user level column contains a lot of garbage values i.e. very high numbers, so let's drop all values over 12600:

After cleaning the data:

1. medianmarechprebal90 (Median of main account balance just before recharge in last 90 days):

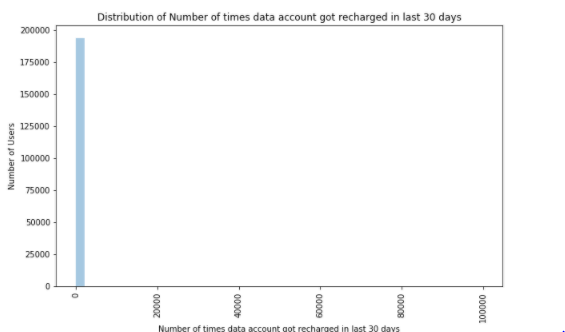
Plot of uncleansed data:

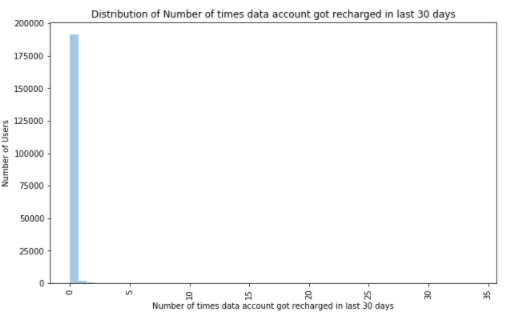
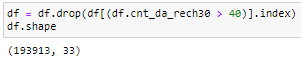
Median of main account balance just before recharge in last 90 days at user level column contains a lot of garbage values i.e. extremely high numbers, so let's drop them:

After cleaning the data:

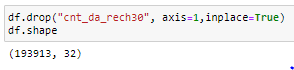
1. cnt\_da\_rech30 (Number of times data account got recharged in last 30 days):

Plot of uncleansed data:



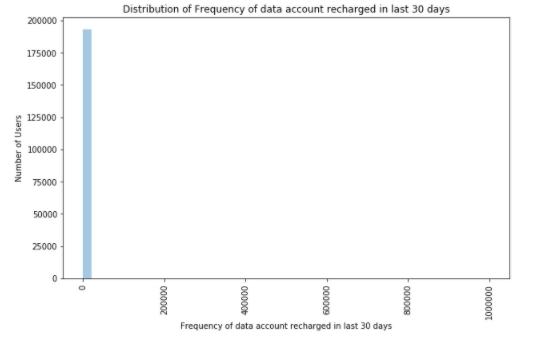
Number of times data account got recharged in last 30 days column contains a lot of garbage values i.e. extremely high numbers, so let's drop them

After cleaning the data:

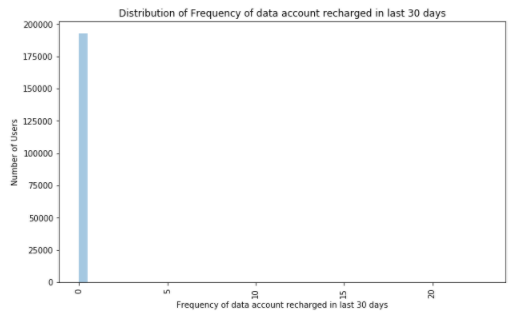
The above plot shows that a very high majority of data points are zero and this feature does not reveal anything, so let's drop it

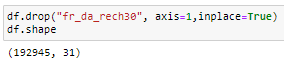
1. fr\_da\_rech30 (Frequency of data account recharged in last 30 days):

Plot of uncleansed data:

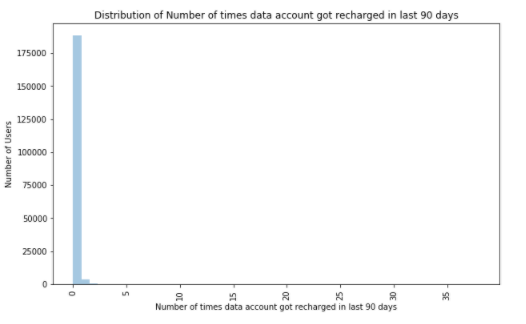


Removing the outliers:

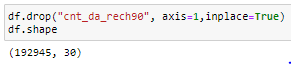
After cleaning the data:

The above plot shows that a very high majority of data points are zero and this feature does not reveal anything, so let's drop it.

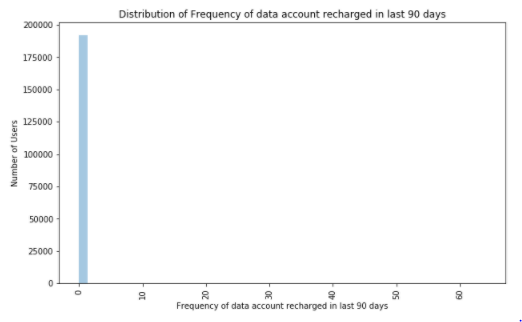
1. cnt\_da\_rech90 (Number of times data account got recharged in last 90 days):

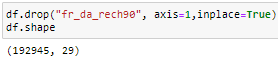
Plot of uncleansed data:

The above plot shows that a very high majority of data points are zero and this feature does not reveal anything, so let's drop it

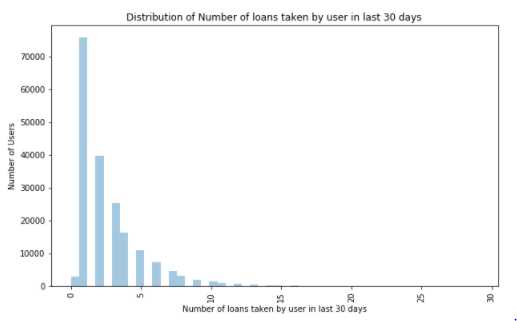


1. fr\_da\_rech90 (Frequency of data account recharged in last 90 days):

Plot of uncleansed data:

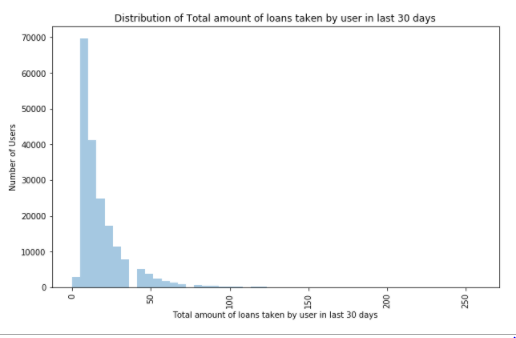
The above plot shows that a very high majority of data points are zero and this feature does not reveal anything, so let's drop it

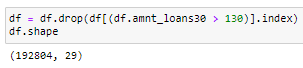
* 1. 30 (Number of loans taken by user in last 30 days):

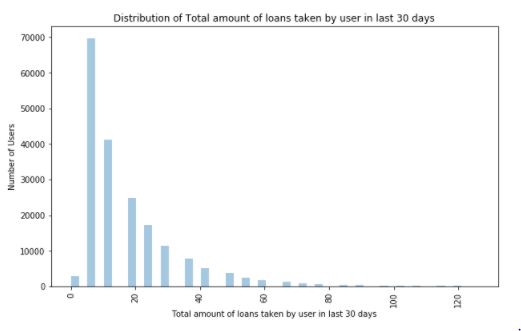
Plot of uncleansed data:

We could drop all cases with cnt\_loans30 >20 but in order to minimize data loss let’s keep this feature as it is.

1. amnt\_loans30 (Total amount of loans taken by user in last 30 days):

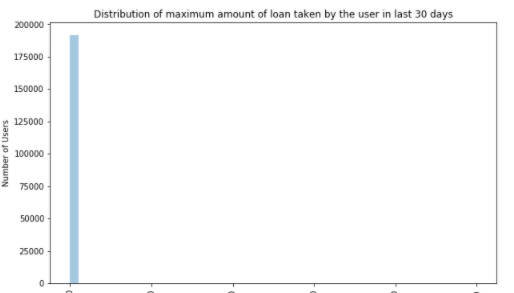
Plot of uncleansed data:

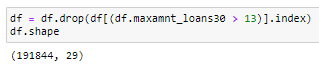
The above plot shows the presence of outliers so let's remove values over 130:

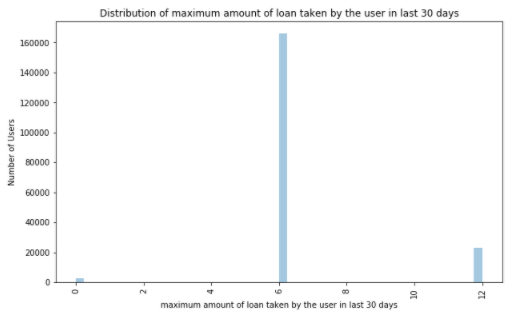
After cleaning the data:

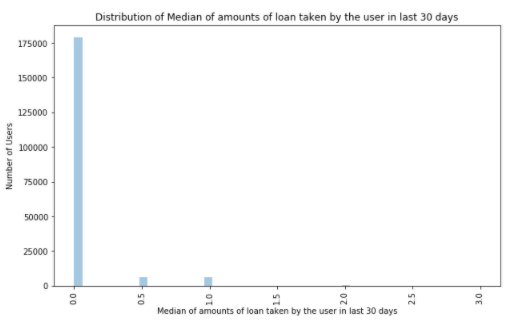
1. maxamnt\_loans30 (maximum amount of loan taken by the user in last 30 days):

Plot of uncleansed data:



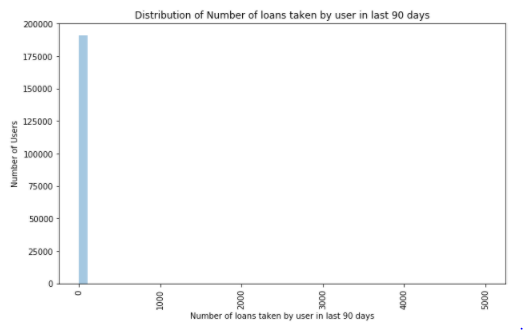
Maximum amount of loan taken by the user in last 30 days column contain a lot of garbage values i.e. extremely high numbers, so let's drop them:

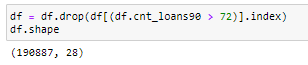
After cleaning the data:

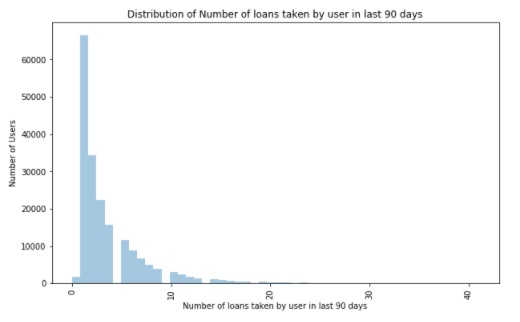
1. medianamnt\_loans30 (Median of amounts of loan taken by the user in last 30 days):

The above plot shows that a very high majority of data points are zero and this feature does not reveal anything, so let's drop it

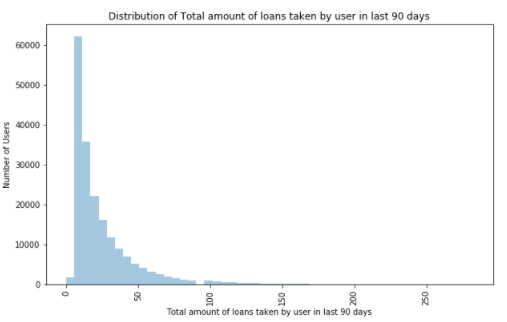
1.  cnt\_loans90 (Number of loans taken by user in last 90 days):

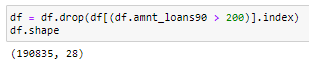
Plot of uncleansed data:

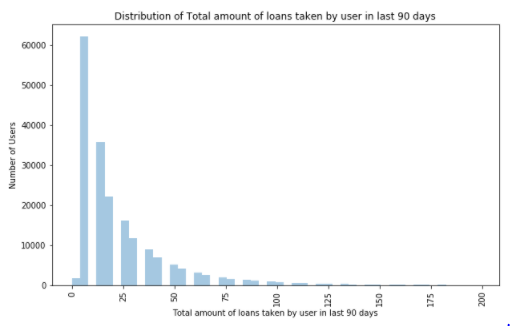
Number of loans taken by user in last 90 days column contains a lot of garbage values i.e. extremely high numbers, so let's drop them

After cleaning the data:

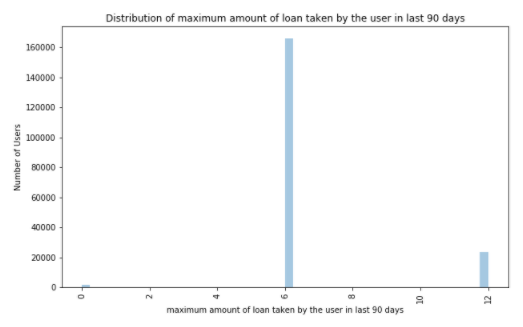
1. amnt\_loans90 (Total amount of loans taken by user in last 90 days):

Plot of uncleansed data:

Total amount of loans taken by user in last 90 days column contains a lot of high outlier values, so let's drop them

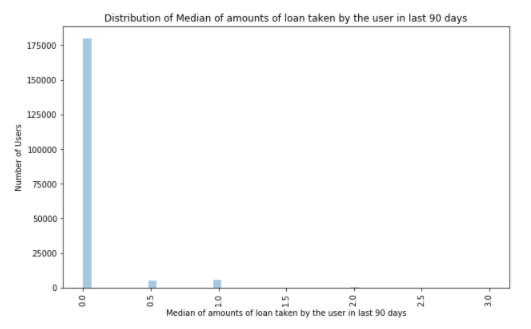
After cleaning the data:

1. maxamnt\_loans90 (maximum amount of loan taken by the user in last 90 days):

Plot of uncleansed data:

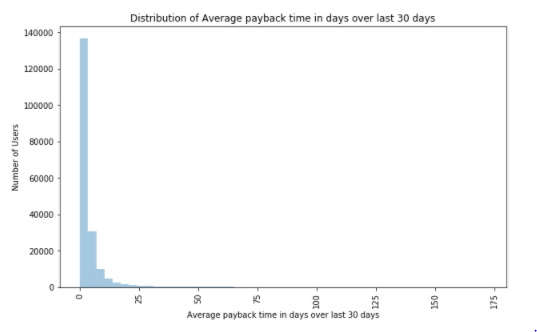
This feature does not need any cleaning.

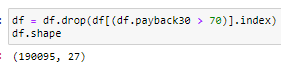
1. medianamnt\_loans90 (Median of amounts of loan taken by the user in last 90 days):

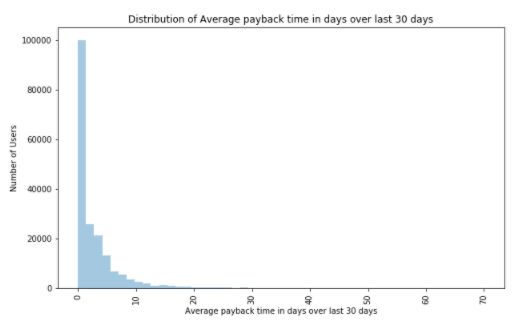
Plot of uncleansed data:

The above plot shows that a very high majority of data points are zero and this feature does not reveal anything, so let's drop it

1. payback30 (Average payback time in days over last 30 days):

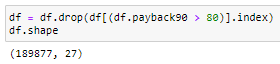
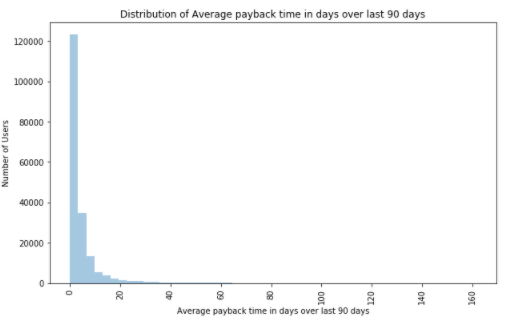
Plot of uncleansed data:

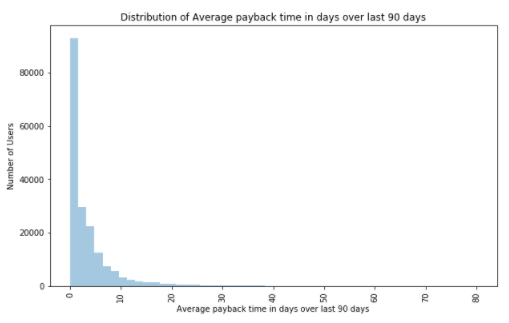
Average payback time in days over last 30 days column contains columns contain a lot of high outlier values, so let's drop all values over 70:

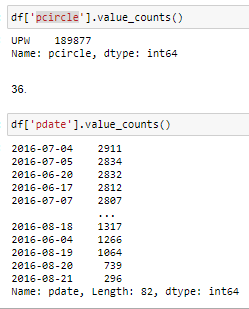
After cleaning the data:

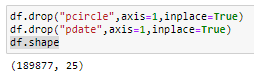
1. payback90 (Average payback time in days over last 90 days):

Plot of uncleansed data:

Average payback time in days over last 90 days column contains a lot of high outlier values, so let's drop them:

After cleaning the data:

1. pcircle (telecom circle) and
2. pdate (date):

The columns pcircle (telecom circle) and pdate (date) do not offer any analytical info so let's drop them

We have cleaned all the features and the dataset is ready for analytics and modelling. Let’s look at the % of data lost during cleaning:

% cases lost = (209593-189877)\*100/209593

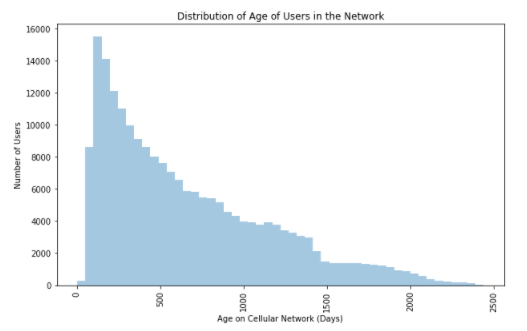
= 9.40

* Data Inputs- Logic- Output Relationships & Visualizations

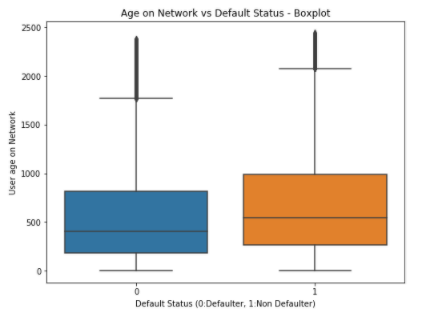
Let’s look at the relevance of all the features in the dataset after data cleaning and their relationship with the output variable i.e. Loan Default status:

Let’s look at the relevance of all the features in the dataset after data cleaning and their relationship with the output variable i.e. Loan Default status:

3. Aon (Age on Cellular Network (Days)):

Tracking the age of users is very important for telecom operators as users may be classified as per their age in the network. The distribution of age is as follows:

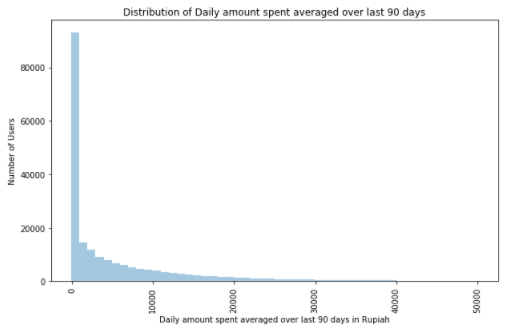
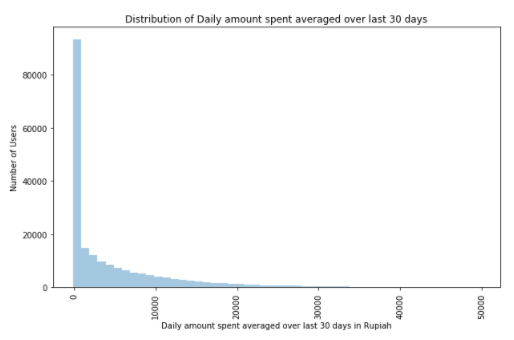
The above plot is highly skewed to the right, which means that most of the users are relatively new to the network and very few users have stayed with the network for a long time, this feature has been selected because age on the cellular network of credit defaulters is expected to be different from those of non-defaulters. Let’s look at the box plot of age on network as per loan default status:

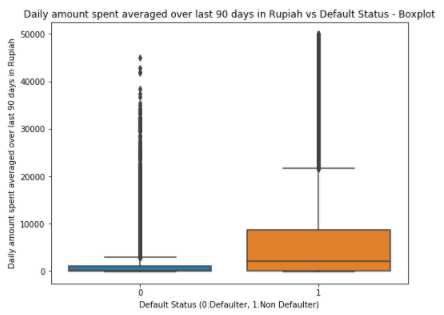
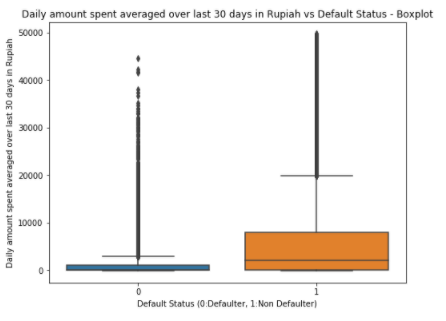


From the above boxplot, the defaulters have relatively lower median age on the network, this implies that long time users are generally more credible.

4. daily\_decr30 (daily amount spent averaged over last 30 days) and

5. daily\_decr90 (daily amount spent averaged over last 90 days): Average daily amount spent is a measure of how much business a user is giving to the telecom network and customers may also be classified based on Average daily amount spent. Let look at the daily amount spent averaged over last 30 and 90 days after cleaning:

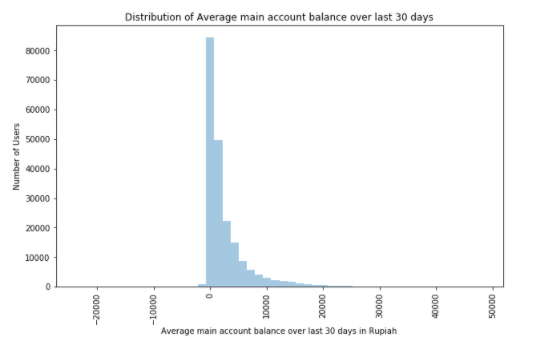
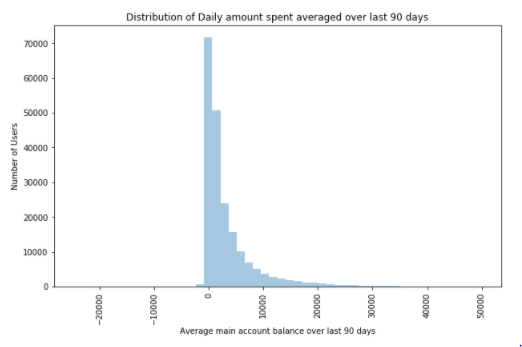
The above plots shows that majority of the users spent lesser amounts or very few users spent very higher amounts. Average daily amount spent of a defaulter is expected to be different from that of a non-defaulter as it reveals a user’s commitment to the network. Let’s look at the box plots of Average daily amount spent averaged over last 30/90 days as per loan default status:



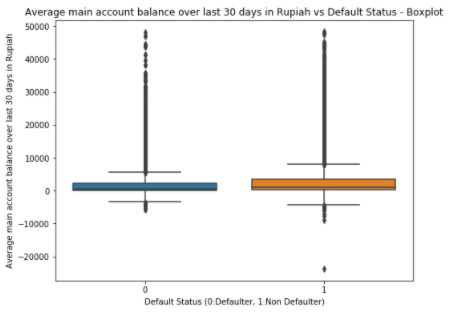
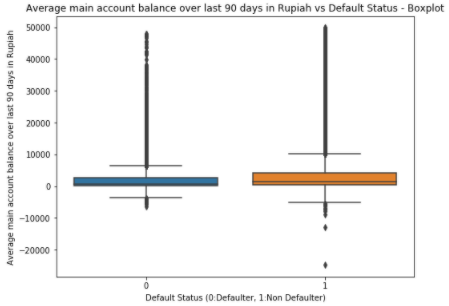
From the above plots, the users who defaulted have relatively lower Daily amount spent averaged over last 30 or 90 days, this means that people with higher average Daily amount spent are generally more credible.

6. rental30 (Average main account balance over last 30 days) and

7. rental90 (Average main account balance over last 90 days):

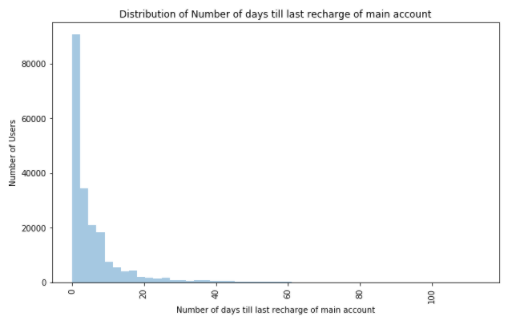
Average main account balance indicates of how much money on an average is the user keeping with the operator at all times. Users may be classified based on this. Let's look at the Average main account balance over last 30/90 days after cleaning:

Most of the users have low Average main account balance or very few users have higher balance. Let's look at Average main account balance over last 30/90 days as per default status:

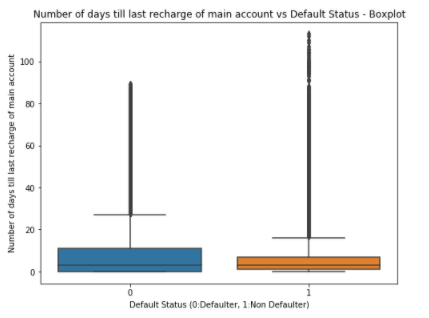


From the above plots, Average main account balance over last 30/90 days is relatively lower for the defaulters, this implies that people with higher average main account balance are generally more credible.

8. last\_rech\_date\_ma (Number of days till last recharge of main account)

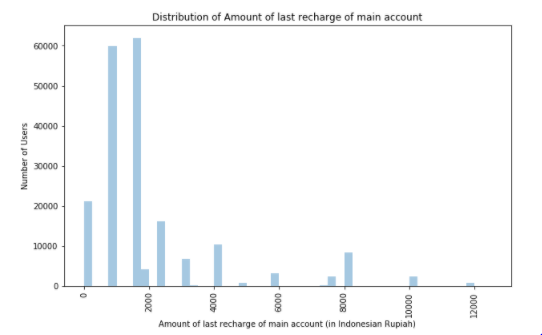
Let's look at the Number of days till last recharge of main account after cleaning:

The above plot shows that most of the users recharge frequently and very few users recharge after long periods.

Let's look at the Number of days till last recharge of main account as per default status:

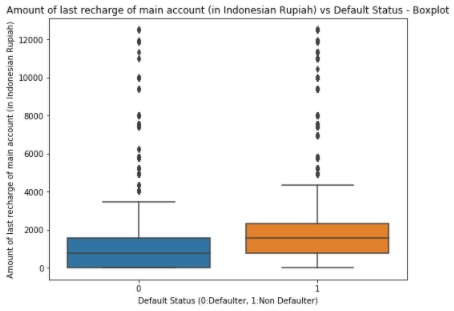
The defaulters have relatively higher periods without recharge, this implies that people who recharge frequently are generally more credible.

10. last\_rech\_amt\_ma (Amount of last recharge of main account (in Indonesian Rupiah)):

Let's visualize the Amount of last recharge of main account after cleaning:

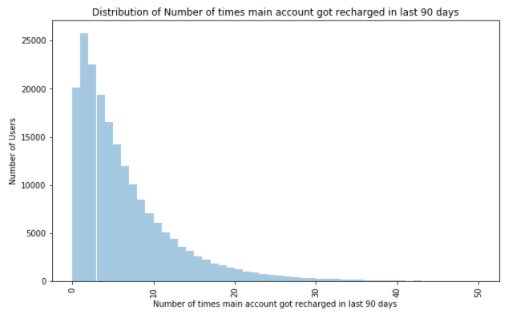
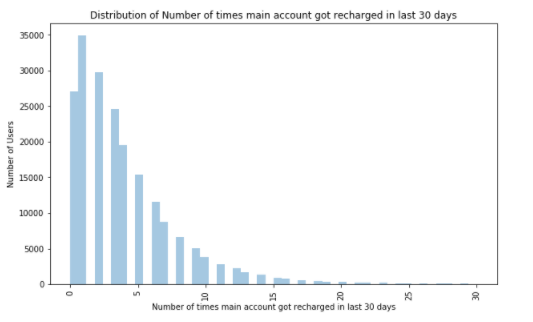
Most of the users have relatively lower amounts of last recharge of main account.

Let's look at the Amount of last recharge of main account as per default status:

The defaulters have relatively lower values for Amount of last recharge of main account, it also implies that users who recharge higher amounts are generally more credible.

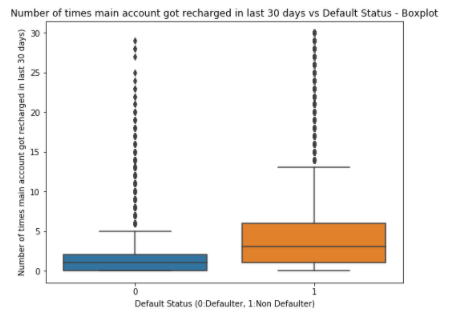
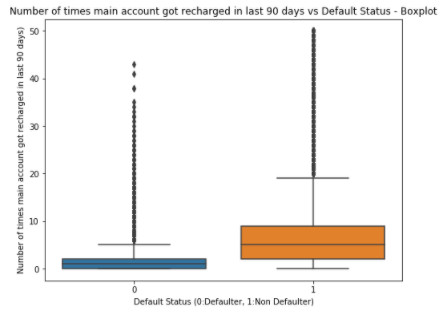
11. cnt\_ma\_rech30 (Number of times main account got recharged in last 30 days):

16. cnt\_ma\_rech90 (Number of times main account got recharged in last 90 days):

Let's look at the Number of times main account got recharged in last 30/90 days after cleaning

The above distributions are highly skewed to the right which means that very few users recharged higher number of times over the last 30/90 days.

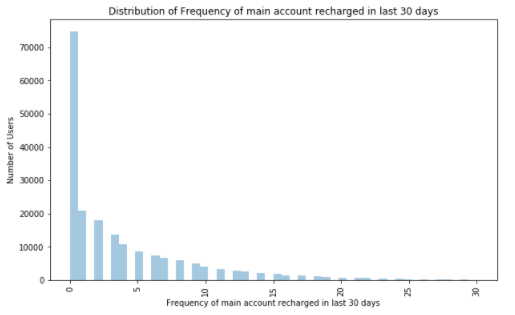
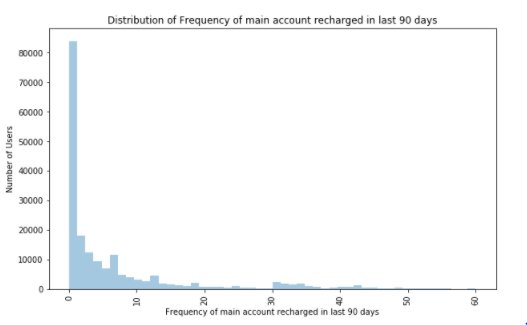
Let's look at the Number of times main account got recharged in last 30/90 days as per default status:



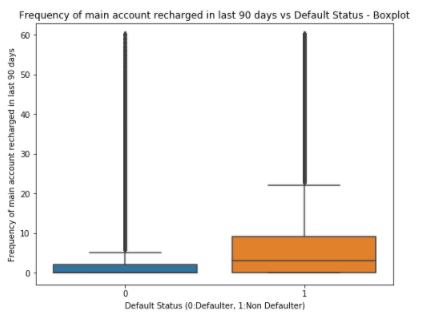
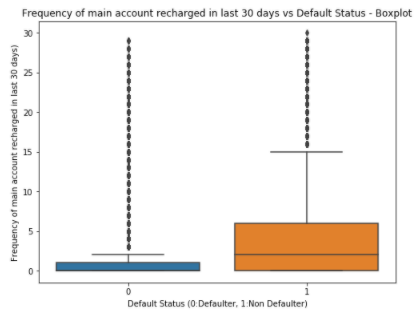
From the above boxplot, clearly the defaulters have relatively very less Number of times main account got recharged in last 30/90 days this implies that people who recharge higher number of times over a period are generally more credible.

12. fr\_ma\_rech30 (Frequency of main account recharged in last 30 days):

17.fr\_ma\_rech90 (Frequency of main account recharged in last 90 days):

Let's look at the distribution of Frequency of main account recharged in last 30/90 days:

Most of the users' frequency of recharge over the last 30/90 days are relatively lower.

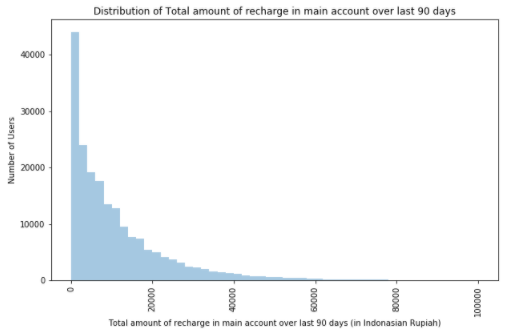
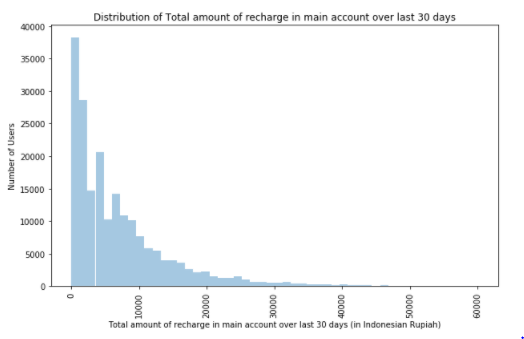
Let's look at the Frequency of main account recharged in last 30/90 days as per default status:

From the above boxplot, clearly the defaulters have relatively very less Frequency of main account recharged in last 90 days this implies that people who recharge very frequently over a period are generally more credible.

13.sumamnt\_ma\_rech30 (Total amount of recharge in main account over last 30 days (in Indonesian Rupiah)):

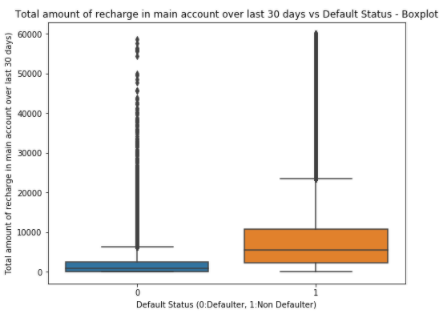
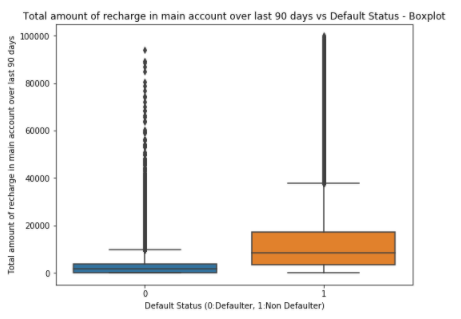
18.sumamnt\_ma\_rech30 (Total amount of recharge in main account over last 30 days (in Indonesian Rupiah)):

Let's look at the Total amount of recharge in main account over last 30/90 days after cleaning data:



From the above plot, majority of the users have low values of Total amount of recharge in main account over last 30/90 days.

Let's look at the Total amount of recharge in main account over last 30 days as per default status:

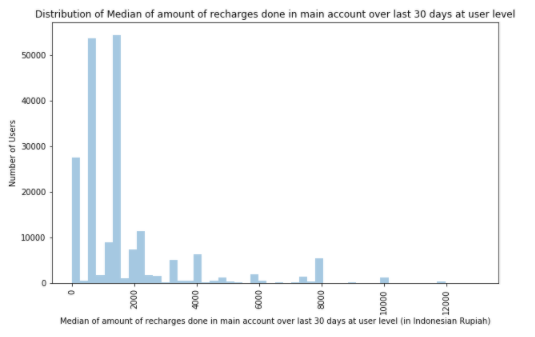
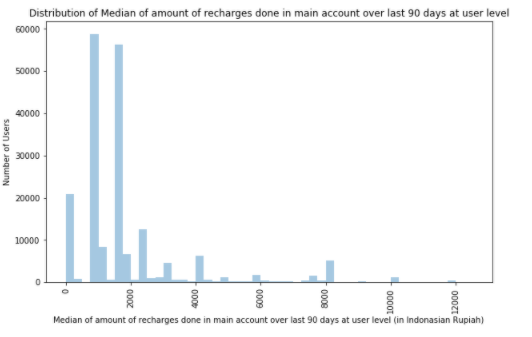


From the above plots, Total amount of recharge in main account over last 90 days is relatively very low for the defaulters, this implies that people with higher Total amount of recharge in main account over a period are generally more credible

14. medianamnt\_ma\_rech30 (Median of amount of recharges done in main account over last 30 days):

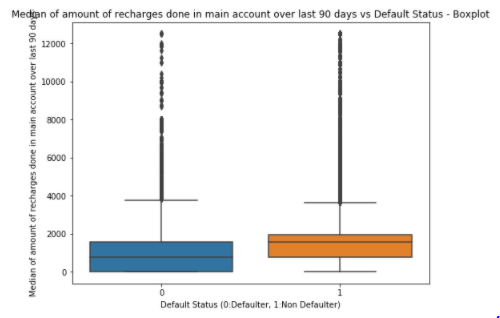
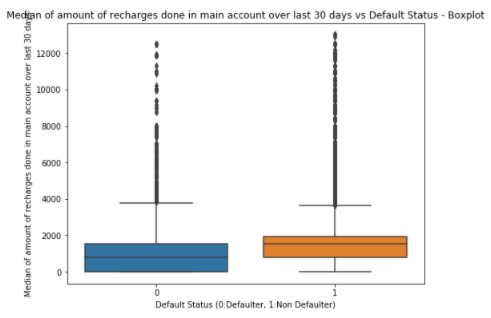
19. medianamnt\_ma\_rech90 (Median of amount of recharges done in main account over last 90 days):

Let's look at the Median of amount of recharges done in main account over last 30/90 days after cleaning:



The Median of amount of recharges done in main account over last 30 days at user level has very few higher values, or majority of the median recharge amounts are low.

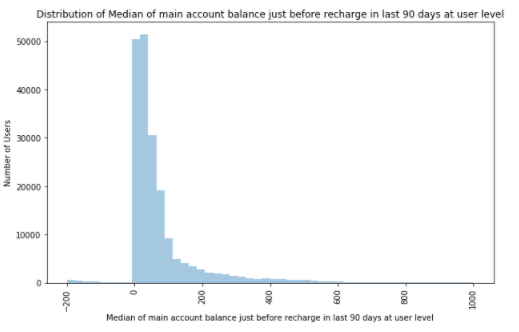
Let's look at the Median of amount of recharges done in main account over last 30/90 days at user level as per default status:



From the above plots, Median of amount of recharges done in main account over last 30/90 days is relatively lower for the defaulters, this

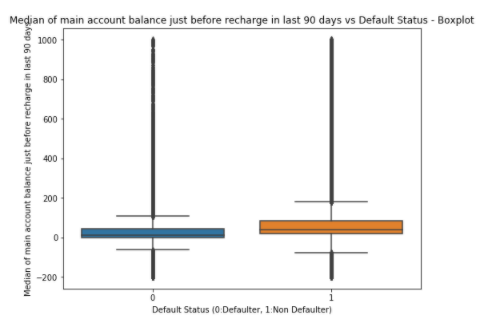
implies that people with higher Median of amount of recharges are generally more credible.

20. medianmarechprebal90 (Median of main account balance just before recharge in last 90 days):

Let's look at the Median of main account balance just before recharge in last 30 days at user level after cleaning the data:

From the above plot, majority of the users have low values for Median of main account balance just before recharge in last 90 days at user level.

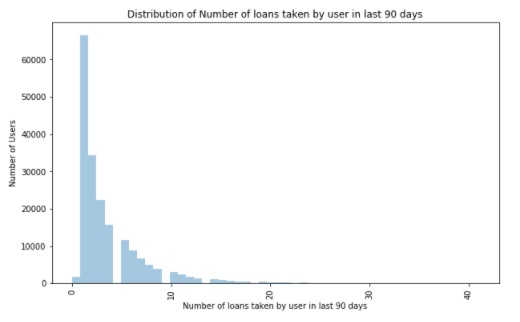
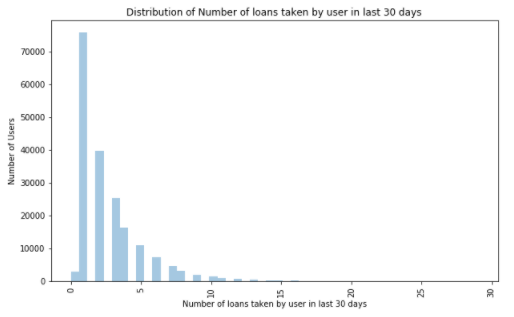
Let's look at the Median of main account balance just before recharge in last 90 days at user level as per default status



From the above plots, Median of main account balance just before recharge in last 90 days is relatively lower for the defaulters, this implies that people with higher Median of main account balance just before recharge are generally more credible.

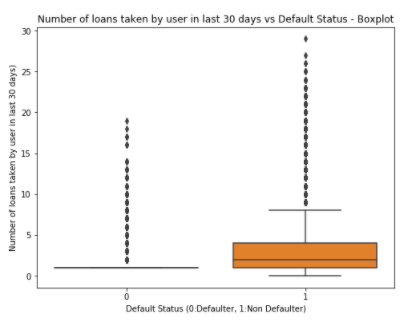
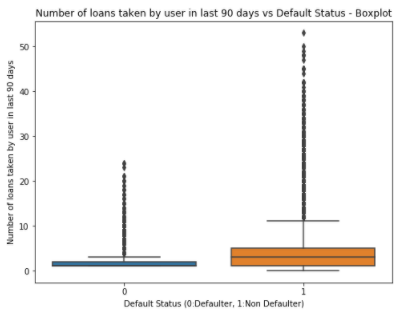
25. cnt\_loans30 (Number of loans taken by user in last 30 days) &

29. cnt\_loans90 (Number of loans taken by user in last 90 days):

Let's look at the Number of loans taken by user in last 30 days

Most of the users have lower Number of loans taken in last 30/90 days.

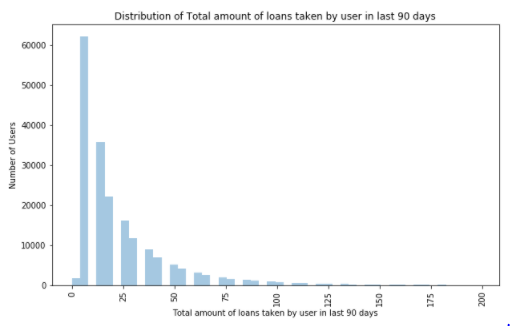
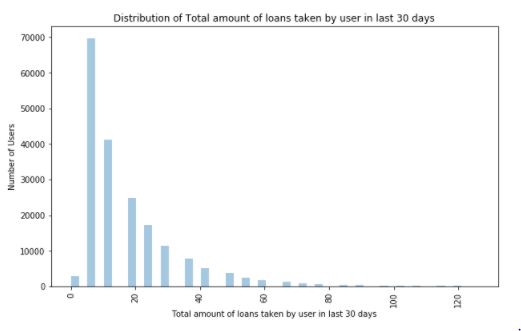
Let's look at the Number of loans taken by user in last 30/90 days as per default status:



From the above boxplot, the defaulters have taken relatively very less number of loans over the last 30/90 days, this also implies that people who take higher number of loans are generally more credible.

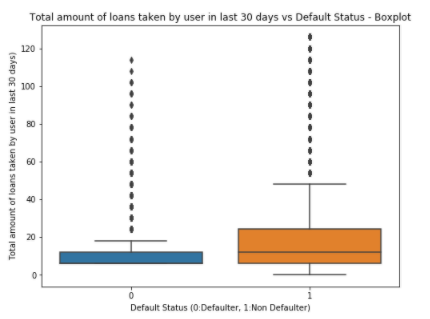
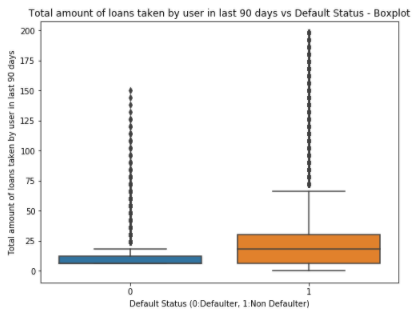
26.amnt\_loans30 (Total amount of loans taken by user in last 30 days) &

30.amnt\_loans90 (Total amount of loans taken by user in last 90 days):

Let's look at the Total amount of loans taken by user in last 30/90 days:

From the above plot, majority of the users took lower Total amount of loans over the last 30/90 days.

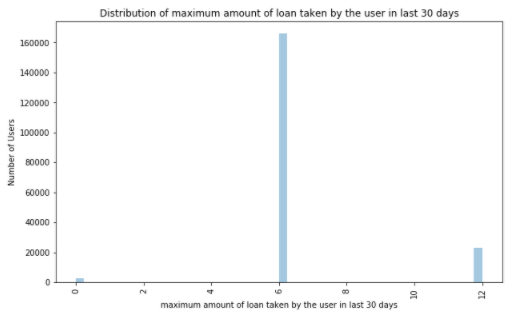
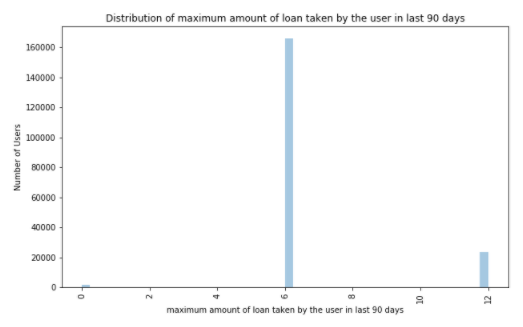
Let's look at the Total amount of loans taken by user in last 30/90 days as per default status



From the above plot, Total amount of loans taken by user in last 90 days is relatively lower for the defaulters, this implies that people with higher Total amount of loans over 30/90 days are generally more credible.

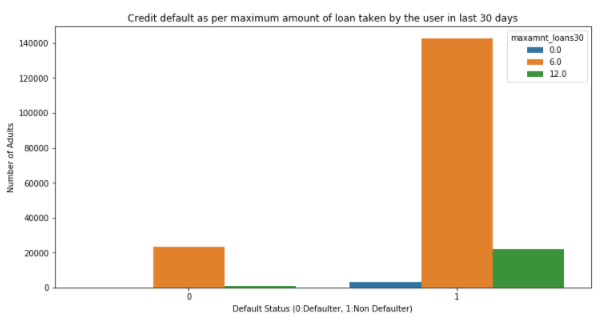
27. maxamnt\_loans30 (maximum amount of loan taken by the user in last 30 days) &

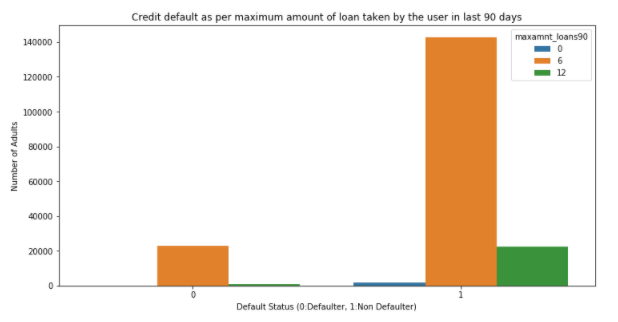
31. maxamnt\_loans90 (maximum amount of loan taken by the user in last 90 days):

Let's look at the maximum amount of loan taken by the user in last 30/90 days after cleaning the data:

Loans of values 6 & 12 Indonesian Rupiah were granted to users, also most users took loans of value 6 Rupiah.

Let's look at the maximum amount of loan taken by the user in last 30/90 days as per default status

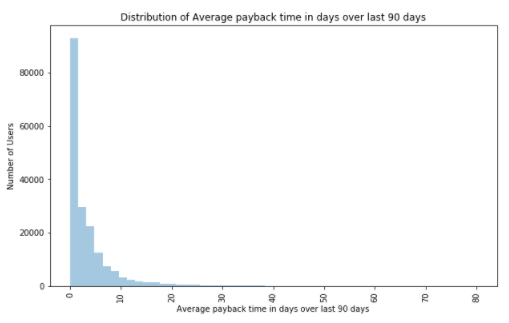
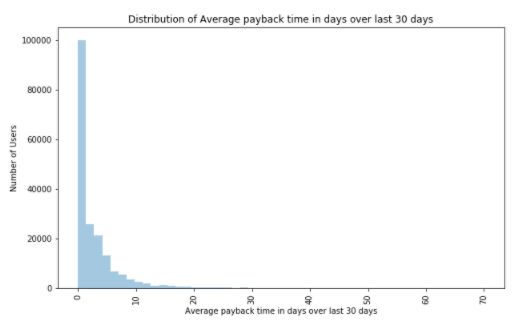




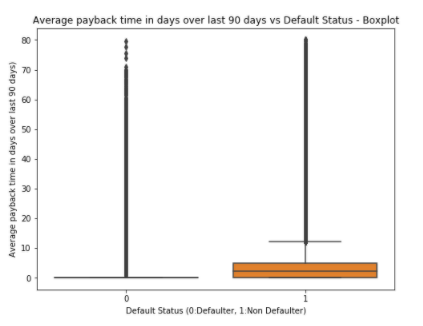
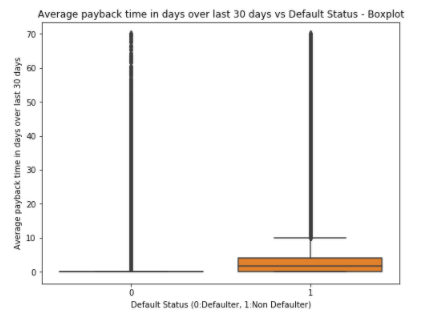
From the above plot, 6 Rupiah loans were most commonly defaulted

33. payback30 (Average payback time in days over last 30 days) &

34.payback90 (Average payback time in days over last 90 days):

Let look at the Average payback time in days over last 30/90 days

Majority of the users have very less payback time or very few users have higher payback time.

Let's look at the Average payback time in days over last 30/90 days as per default status:

From the above plots, Average payback time in days over last 30 days is relatively lower for the defaulters, which is surprising. Perhaps the median is zero for defaulters because of a lot of zero entries.

* State the set of assumptions (if any) related to the problem under consideration

1. The dataset was cleaned to minimise loss of data.
2. In this business application our goals is to classify all the defaulters correctly so recall score would be our choice of evaluation metric

* Hardware and Software Requirements and Tools Used

Hardware Used:

(4th Gen Ci5/1.8GHz /4GB/ 1TB/ Win8.1/ 2GB Graph)

Software Used: Python 3 (Jupyter Notebook)

**Model/s Development and Evaluation**

* Identification of possible problem-solving approaches (methods)

The dataset was cleaned with the goal to remove outliers as well as to minimise the data lost. Many features which had very high skewness and were not clearly bifurcating the defaulters were dropped.

The input data set was standard scaled before the modelling.

Multiple classification algorithms were tested along with multiple evaluation metrics.

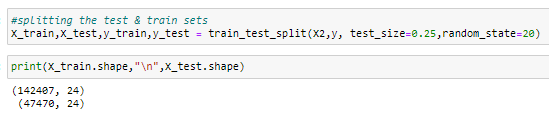
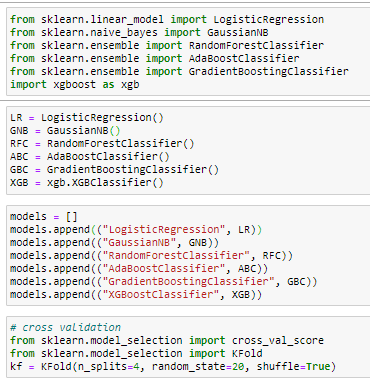
The most effective evaluation metric for this particular application is recall. As our goal is to identify all the defaulters.

* Testing of Identified Approaches (Algorithms)

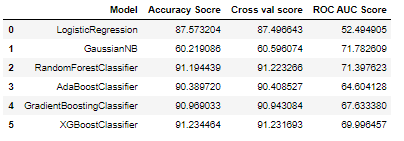
The following algorithms were used for the classification of defaulters:

1. LogisticRegression
2. GaussianNB
3. RandomForestClassifier
4. AdaBoostClassifier
5. GradientBoostingClassifier
6. XGBClassifier

* Run and Evaluate selected models

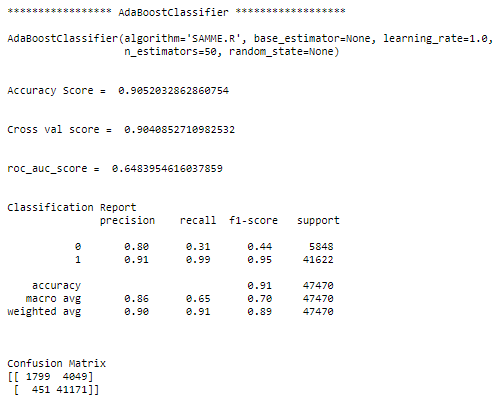
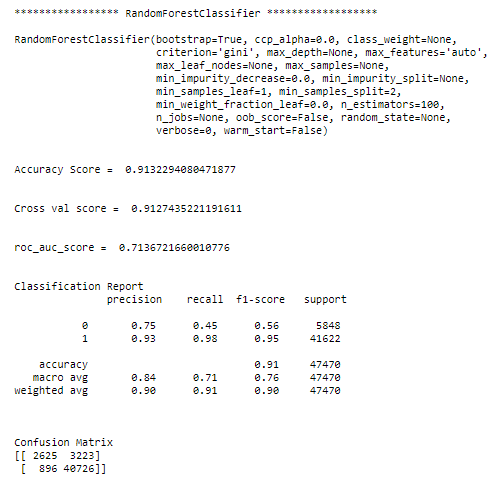
Code for all the classification models:

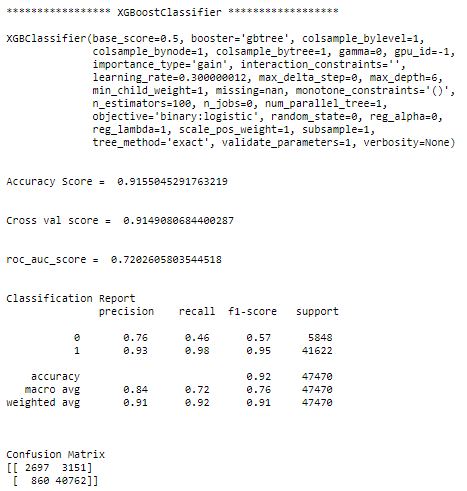
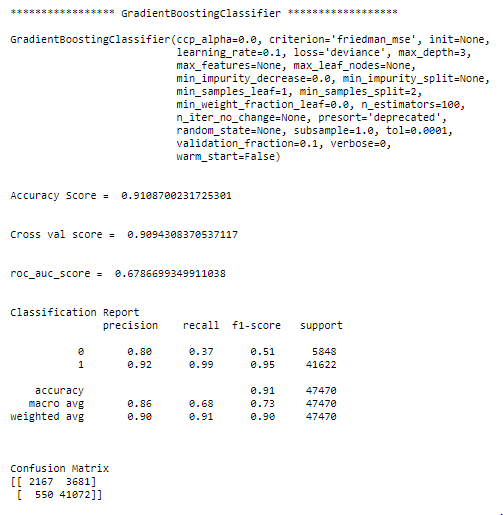


The summary of evaluation metrics is as follows:

It might appear from the accuracy and cross validation scores that Ensemble methods are performing better but if we look at the confusion matrix and recall scores, we observe that these models are not good at classifying the defaulters:

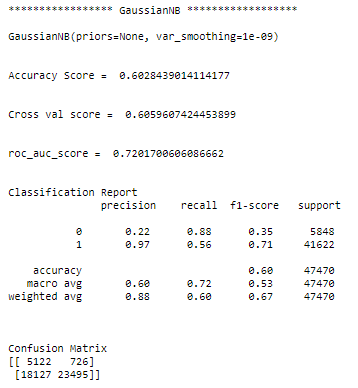
Let’s look at other evaluation parameters for the ensemble methods:

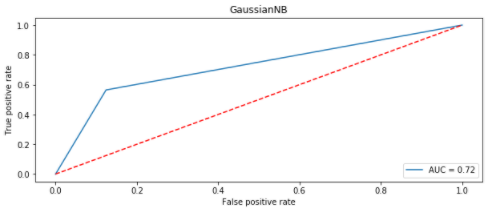
 

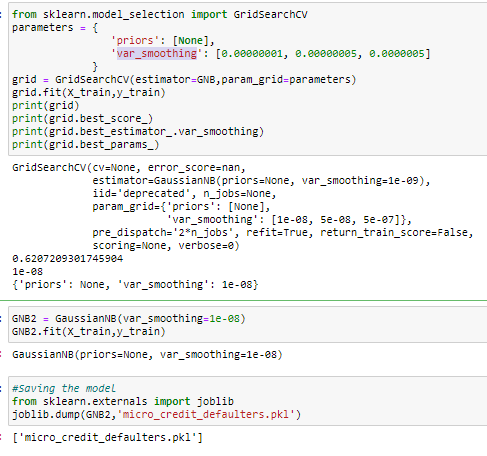


We are getting better accuracy scores and cross validation scores for ensemble methods because these models classify non-defaulters very well and the dataset is imbalanced i.e. there are a very high number of non-defaulters compared to defaulters. But their inefficiency in classifying defaulters can be observed from the classification matrices and low recall scores.

In this project we are mainly interested in detecting the defaulters and the GaussianNB Model performs far better than any other model in classifying defaulters. The GaussianNB Model Classifies 5122 out of 5848 total defaulters in the test set. While the precision score which measures the accuracy of classifying the defaulters correctly out of the total defaulters classified is low (0.22) for GaussianNB Model (Precision is highest for GradientBoostingClassifier: 0.80), the Recall Score which measures the accuracy of classifying all the defaulters is highest (0.88) for GaussianNB Model and no other model gives a better recall score. The GaussianNB Model also has the highest precision score (0.97) out of all models for non-defaulters. The Area under ROC Curve (AUC) Score is also highest for GaussianNB indicating better performance than other models. In our business application, we need to correctly classify all the possible defaulters and GaussianNB Model does that better than all but there is a trade-off that correctly identifying all defaulters comes with a cost of incorrectly classifying many non-defaulters which is why the accuracy & cross validation scores are low. The fact that the dataset is highly skewed (only 12.5% of defaulters), also contributes to the low accuracy score.





Parameter setting using GridSearchCV: Though GaussianNB does not have much parameters to set we can still perform GridSearchCV with var\_smoothing as follows:

* Key Metrics for success in solving problem under consideration

The key metrics used in solving the problem were recall and confusion matrix as we are interested in classifying all the defaulters correctly in this specific business application.

* Interpretation of the Results

Standard scaled data gives better performance than the unscaled data. Also as the dataset is imbalanced with too many non-defaulters, usual evaluation metrics like accuracy score and cross validation score fail to determine the best algorithm as the problem is to classify all the defaulters correctly. Recall score is the choice of evaluation metric for this business application.

While ensemble methods are providing accuracy & cross validation scores as high as 91% their recall scores are very low. The best recall score obtained is 88% by using the GaussianNB Classifier which means it classified 88% of the total defaulters correctly, it also has the highest precision score of 97% out of all the models for classifying non-defaulters which means out of all the classified non-defaulters 97% were correctly classified. But such high recall score for defaulters and high precision score for non-defaulters comes with a lot of non-defaulters being classified as defaulters which is why the accuracy and cross validation scores are lower for the GaussianNB Model. The fact that the dataset is highly skewed (only 12.5% of defaulters), also contributes to the low accuracy score.

**CONCLUSION**

* Key Findings and Conclusions of the Study

From the data analysis we found that the following types of users are generally more likely to default:

1. Short-time users
2. Users with lesser average daily amount spent
3. Users with lesser average main account balance
4. Users who recharge less frequently or lesser number of times over 30/90 days
5. Users who recharge lesser amounts
6. Users who recharge lower total amounts over 30/90 days
7. Users with lower median of amount of recharges over 30/90 days
8. Users with lower median of main account balance just before recharge
9. Users who take loans less frequently or lower number of loans taken over 30/90 days
10. Users who take lower total amounts as loans over 30/90 days

Also 6 Rupiah loans were defaulted more frequently when compared to 12 Rupiah loans.

Accuracy or cross validation score as high as 91% can be obtained but since the data is imbalanced the recall score for defaulters is low which is why models cannot be selected based on accuracy or cross validation score.

In this specific business application we are interested in classifying all the defaulters correctly hence our choice of evaluation metric is recall score.

Ensemble algorithms like RandomForestClassifier & GradientBoostingClassifier give almost 91% accuracy/cross validation score but their recall scores are low. The highest recall score obtained was 88% using GaussianNB algorithm, also this model has the highest precision score of 97% out of all the models for classifying non-defaulters classifier Which means that the GaussianNB classified 88% of the total defaulters correctly, also out of all the classified non-defaulters 97% were correctly classified. But the model also wrongly classifies many non-defaulters as defaulters which is why we get lower accuracy and cross validation scores. The fact that the dataset is highly skewed (only 12.5% of defaulters), also contributes to the low accuracy score. So from the business perspective it allows us to safeguard against defaulters as well as an improved selection of customers for loans. But this safeguard comes at a cost of classifying many non-defaulters as defaulters. But if there’s a considerable fraction of defaulter in the dataset or the dataset is less imbalanced than no model performs better than GaussianNB model in selecting customers correctly (high precision score) and in rejecting all potential defaulters (high recall score).

* Learning Outcomes of the Study in respect of Data Science

From the visualizations of all the features in the cleaned dataset, we were able to characterise the tentative behaviour of defaulters. Looking at the Box plot as per default status provides us a very good idea about the tendencies of defaulters for each continuous features.

We could’ve cleaned the dataset as per usual practice of discarding all values having z>3 but the amount of data lost is very high in this case. Also, even after performing this much of cleaning the performance of the models is not better.

As the primary goal of this business application is to identify all the defaulters, we took recall and confusion matrix as the evaluation metrics and based on that GaussianNB provides the best performance. Also, high recall score for defaulters and high precision score for non-defaulters comes with a cost of having low accuracy & cross validation score which is a limitation in this business application.

* Limitations of this work and Scope for Future Work

The primary goals of the model is to classify all defaulters correctly and we are classifying the defaulters with recall score as high as 88% but it comes with a cost of low accuracy & cross validation scores as the model misclassifies many non-defaulters as defaulters. The fact that the dataset is highly skewed (only 12.5% of defaulters), also contributes to the low accuracy score. But perhaps in a country where the default fraction is higher and the dataset is less imbalanced we can observe better accuracy and cross validation scores with the GaussianNB model as the model has highest recall score for defaulters and highest precision score for non-defaulters. So if the default fraction is higher in the region, it becomes even more important to select customers correctly (high precision score) and reject all potential defaulters (high recall score) in that case no model performs better than GaussianNB.