FLIP ROBO

Micro-Credit Defaulter Model

Submitted by:

Utsav Rastogi

ACKNOWLEDGMENT

This project was made possible with the help of my supervisor Swati Mahaseth. Under her guidance, I was able to complete the project on time and ample learning possibilities this project provided me with.

This project gave me the first glimpse of how real-world datasets look like and what is actually need to be provided to the client in terms of solutions. This was my first-time experience with a dataset consisting of more than 80,000 samples, which also made me look for different approaches and research libraries meant for handling large datasets.

Data analysis and preprocessing was done keeping in mind how to make the code run efficiently and quickly with utilizing minimal memory. I had to refer to various articles on Medium, Analytics Vidhya, Kaggle and stack exchange to look for potential debugging solutions and visualization plots.

INTRODUCTION

Business Problem Framing

This project tries to shed some light on the different parameter that can result in defaulting on a micro loan. This can help MFIs to adopt a marketing strategy more conducive of including more non-defaulters and try to minimize factors which can eventually lead to defaulting and can contribute in revenue loss.

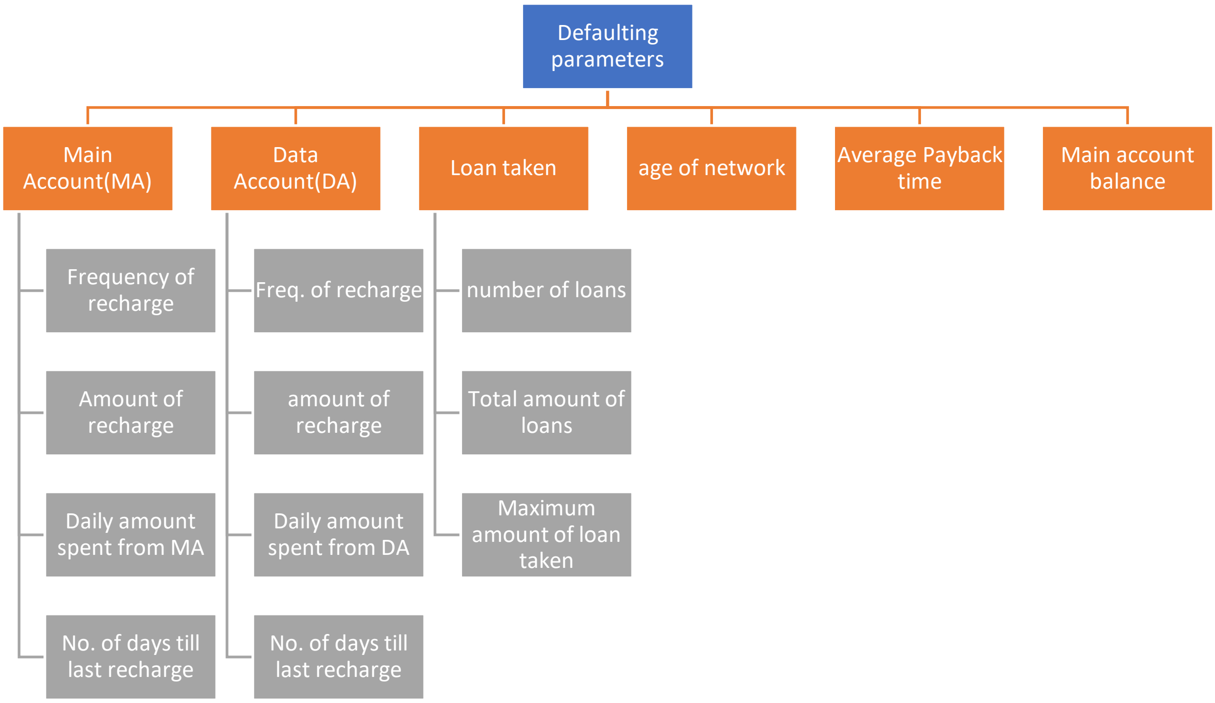
Conceptual Background of the Domain Problem

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.
A fixed wireless telecommunication network provider is 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.

Review of Literature

The chances of defaulting rest on these features, where the highest correlation is seen with these 5 columns:

1. Number of times main account got recharged in the last 90 days
2. Number of times main account got recharged in the last 30 days
3. Total amount of recharge in main account over last 90 days
4. Total amount of recharge in main account over last 30 days
5. Total amount of loans taken by user in last 90 days



Motivation for the Problem Undertaken

The objective behind this project was to provide the company an overview about how some parameters is affecting the defaulting rate more than the others and where to focus their energy, time and money to gain and retain those type of customers which showcase a history of non-defaulting type of behaviour.

This can assist the company in drafting their business approaches in such a way that can discourage defaulting by charging higher interest rates and can reward on time payers with small perks time to time.

The motivation specifically was to encourage telecom connectivity between low-income consumers, who can feel restrained due to their financial conditions. With the help of MFI’s, telecom companies can provide small credit so people can stay connected even during an emergency without loosing connectivity with their loved ones when it is needed the most.

Analytical Problem Framing

Mathematical/ Analytical Modeling of the Problem

Describe the mathematical, statistical and analytics modelling done during this project along with the proper justification.

Statistical analysis	• Zscore was used to determine the outliers in the data and • VIF was used to determine the multicollinearity between features and was paired with • Correlation helped understand the relation between features and target. • Skewness was also used to check the distribution and remove it in the features if it wasn't gaussian.
Visual analysis	Libraries used: Matplotlib and Seaborn
Algorithms	Different classification algorithms such as Gradient Boosting classifier & Decision Tree were used which are all provided by sklearn library.

Data Sources and their formats

What are the data sources, their origins, their formats and other details that you find necessary? They can be described here. Provide a proper data description. You can also add a snapshot of the data.

Data origin: Data is provided to Fliprobo by their client which is a Telecom company having a collaboration with a MFI.

Data Source: This dataset is only a part of the original dataset to help us interns understand the Machine Learning pipeline and how to relate it to a business problem.

- Dataset format: CSV

- Data Description:
- Dataset consists of 37 columns names:
  'unamed', 'label', 'msisdn', 'aon', 'daily_decr30', 'daily_decr90','rental30', 'rental90','last_rech_date_ma', 'last_rech_date_da', 'last_rech_amt_ma', 'cnt_ma_rech30','fr_ma_rech30','sumamnt_ma_rech30','medianamnt_ma_rech30','medianmarechprebal30', 'cnt_ma_rech90', 'fr_ma_rech90', 'sumamnt_ma_rech90','medianamnt_ma_rech90', 'medianmarechprebal90', 'cnt_da_rech30', 'fr_da_rech30', 'cnt_da_rech90', 'fr_da_rech90', 'cnt_loans30', 'amnt_loans30', 'maxamnt_loans30', 'medianamnt_loans30', 'cnt_loans90', 'amnt_loans90', 'maxamnt_loans90', 'medianamnt_loans90', 'payback30', 'payback90', 'pcircle', 'pdate'
- There were only 3 categorical features present, namely – pcircle, pdate and msisdn. Rest 34 columns were either int or float.
- We have to predict the label column using Supervised binary classification machine learning algorithm. Label means Flag indicating whether the user paid back the credit amount within 5 days of issuing the loan {1: success, 0: failure}
- There were no null values present in the dataset.
- Pdate was converted to a datetime object from string, which can be further used to extract day, month and year
- Target column is imbalanced, and can be balanced using balancing methods.

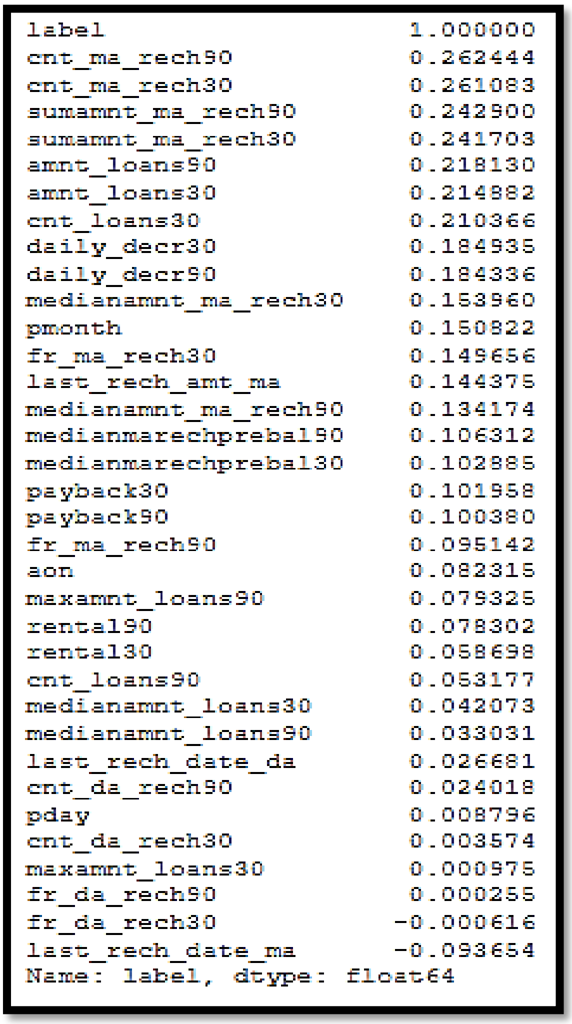
Data Pre-processing

Exploratory Data Analysis (EDA) was done to ensure that model will be fed with cleaned and meaningful data. This is usually the most time-consuming task in a Project.

- Performed an outlier check during EDA so as to hopefully reduce data and outliers, to make the process ahead quicker.
- Using Zscore, outliers were eliminated and we incurred a Data loss of 9%
- To expedite the process ahead, I also converted all the datatypes to 32 bits from 64 bits to reduce memory intake. It reduced from 60MB to 31 MB, almost 50% decrease.

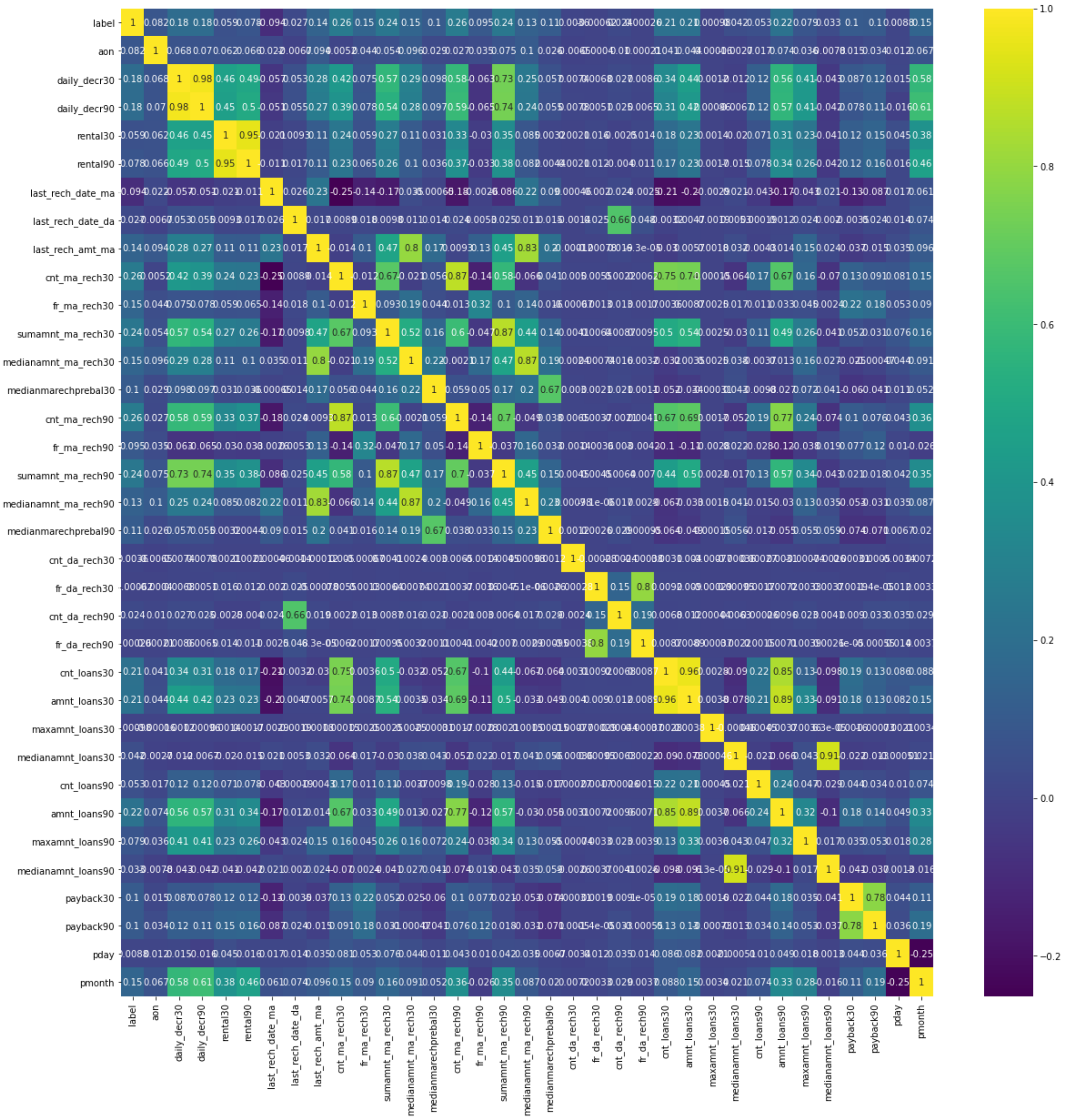
- We dropped Pdate (after converting it to datetime object, and extracting day, month & year), msisdn (consists of 90+% unique values), pcircle (only 1 unique value) and index features.
- We can also drop pyear (extracted from Pdate and is made up of only one value 2016)

Data Inputs- Logic- Output Relationships



```
label                   1.000000
cnt_ma_rech90           0.262444
cnt_ma_rech30           0.261083
sumamnt_ma_rech90       0.242900
sumamnt_ma_rech30       0.241703
amnt_loans90            0.218130
amnt_loans30            0.214882
cnt_loans30             0.210366
daily_decr30            0.184935
daily_decr90            0.184336
medianamnt_ma_rech30    0.153960
pmonth                  0.150822
fr_ma_rech30            0.149656
last_rech_amt_ma        0.144375
medianamnt_ma_rech90    0.134174
medianmarechprebal90    0.106312
medianmarechprebal30    0.102885
payback30               0.101958
payback90               0.100380
fr_ma_rech90            0.095142
aon                     0.082315
maxamnt_loans90         0.079325
rental90                0.078302
rental30                0.058698
cnt_loans90             0.053177
medianamnt_loans30      0.042073
medianamnt_loans90      0.033031
last_rech_date_da       0.026681
cnt_da_rech90           0.024018
pday                    0.008796
cnt_da_rech30           0.003574
maxamnt_loans30         0.000975
fr_da_rech90            0.000255
fr_da_rech30           -0.000616
last_rech_date_ma      -0.093654
Name: label, dtype: float64
```

- Almost all columns are positively related with the target column, “label”. We can see some columns have almost zero correlation with the targe, indicating we can drop them without impacting the prediction of the target in models in the further steps.



- In the heatmap above, we can see that there's collinearity between various features, and hence we'll use VIF to also determine which additional columns can be dropped which have VIF >12.
- "cnt_loans30","daily_decr90" are dropped, which reduces the vif to below 10 for the remaining columns.

- Hardware and Software Requirements and Tools Used

Library	Description
Pandas	Data Manipulation
sklearn	Machine learning algorithms
Seaborn, matplotlib	Visualisation

RandomizedSearchCV	Hyperparameter tuning
Pickle	Saving the model
Numpy and Stats.api	Array manipulation, Zscore and VIF

Software: Windows 11
IDE: Jupyter Notebook

Model/s Development and Evaluation

Identification of possible problem-solving approaches (methods)

The dataset was analyzed using the standard procedure where we indulge in statistical and graphical analysis of the dataset. Statistical methods like Zscore and VIF were used to clean data.

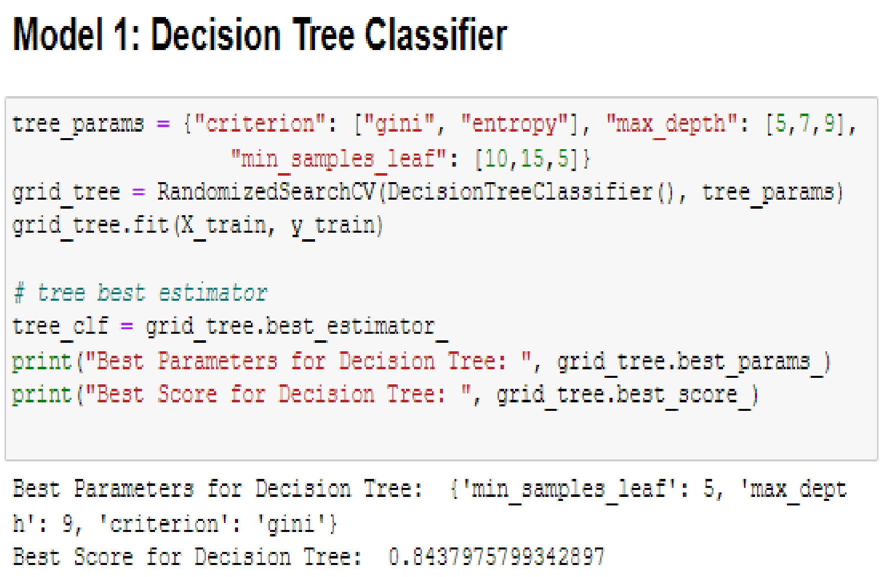
Graphical analysis indicated the class imbalance in the target variable. Hence, SMOTE was used to upsample the minority class, 1 and 0 were made equal with each having 1,24,790 samples.

Testing of Identified Approaches (Algorithms)

Algorithms used:

- Logistic Regression
- Decision Tree Classifier
- Random Forest Classifier
- Gradient Boosting Classifier

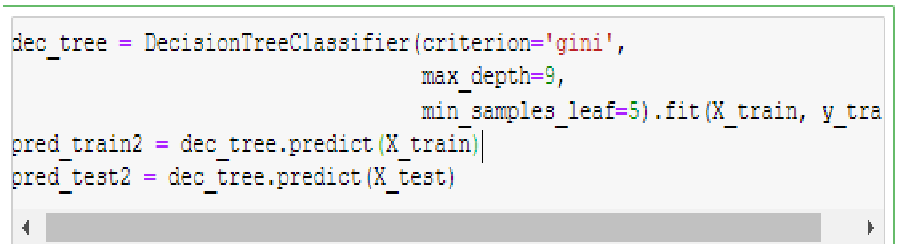
- Run and Evaluate selected models



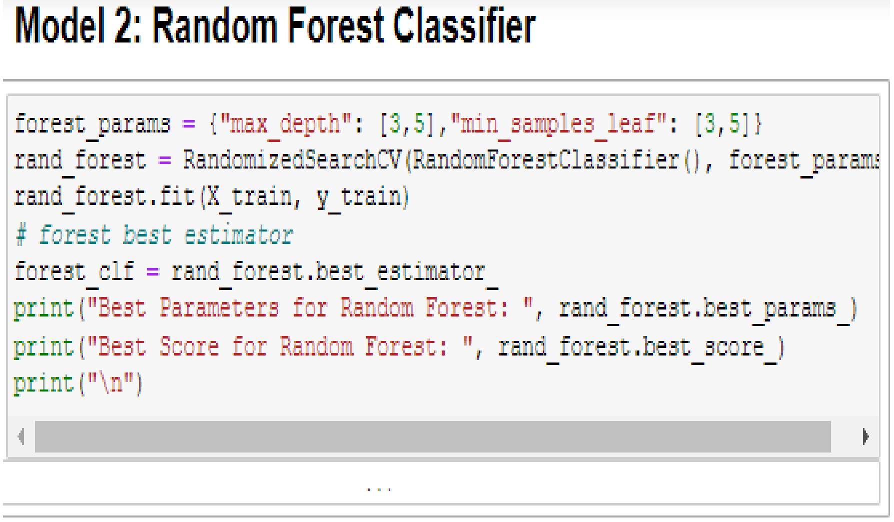
Model 1: Decision Tree Classifier

```
tree_params = {"criterion": ["gini", "entropy"], "max_depth": [5,7,9],
                  "min_samples_leaf": [10,15,5]}
grid_tree = RandomizedSearchCV(DecisionTreeClassifier(), tree_params)
grid_tree.fit(X_train, y_train)

# tree best estimator
tree_clf = grid_tree.best_estimator_
print("Best Parameters for Decision Tree: ", grid_tree.best_params_)
print("Best Score for Decision Tree: ", grid_tree.best_score_)
```

```
Best Parameters for Decision Tree:  {'min_samples_leaf': 5, 'max_dept
h': 9, 'criterion': 'gini'}
Best Score for Decision Tree:  0.8437975799342897
```



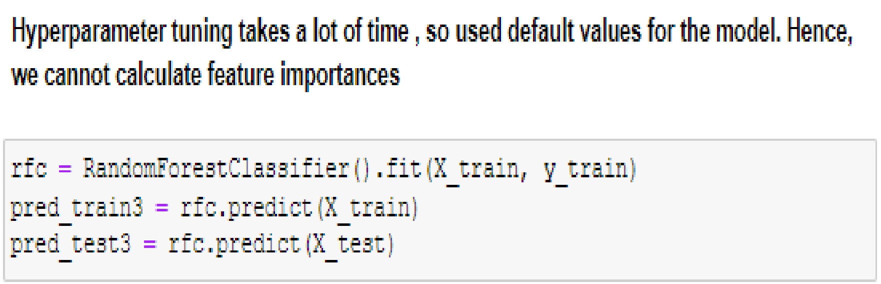
```
dec_tree = DecisionTreeClassifier(criterion='gini',
                                  max_depth=9,
                                  min_samples_leaf=5).fit(X_train, y_tra
pred_train2 = dec_tree.predict(X_train)
pred_test2 = dec_tree.predict(X_test)
```



Model 2: Random Forest Classifier

```
forest_params = {"max_depth": [3,5],"min_samples_leaf": [3,5]}
rand_forest = RandomizedSearchCV(RandomForestClassifier(), forest_param
rand_forest.fit(X_train, y_train)
# forest best estimator
forest_clf = rand_forest.best_estimator_
print("Best Parameters for Random Forest: ", rand_forest.best_params_)
print("Best Score for Random Forest: ", rand_forest.best_score_)
print("\n")
```

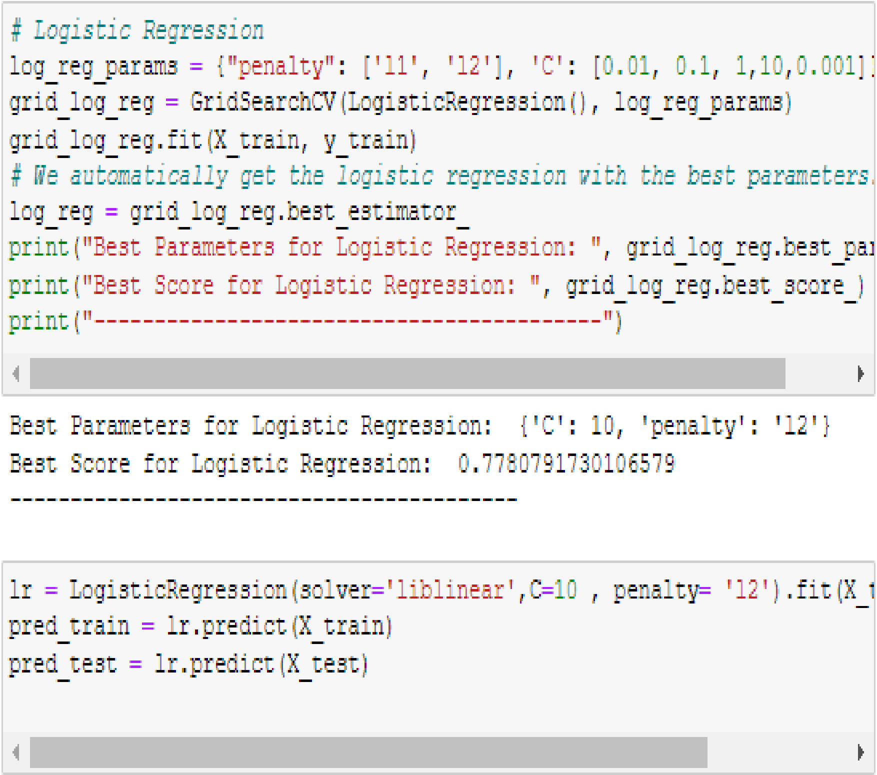
...



Hyperparameter tuning takes a lot of time , so used default values for the model. Hence, we cannot calculate feature importances

```
rfc = RandomForestClassifier().fit(X_train, y_train)
pred_train3 = rfc.predict(X_train)
pred_test3 = rfc.predict(X_test)
```

Model 3: Logistic Regression

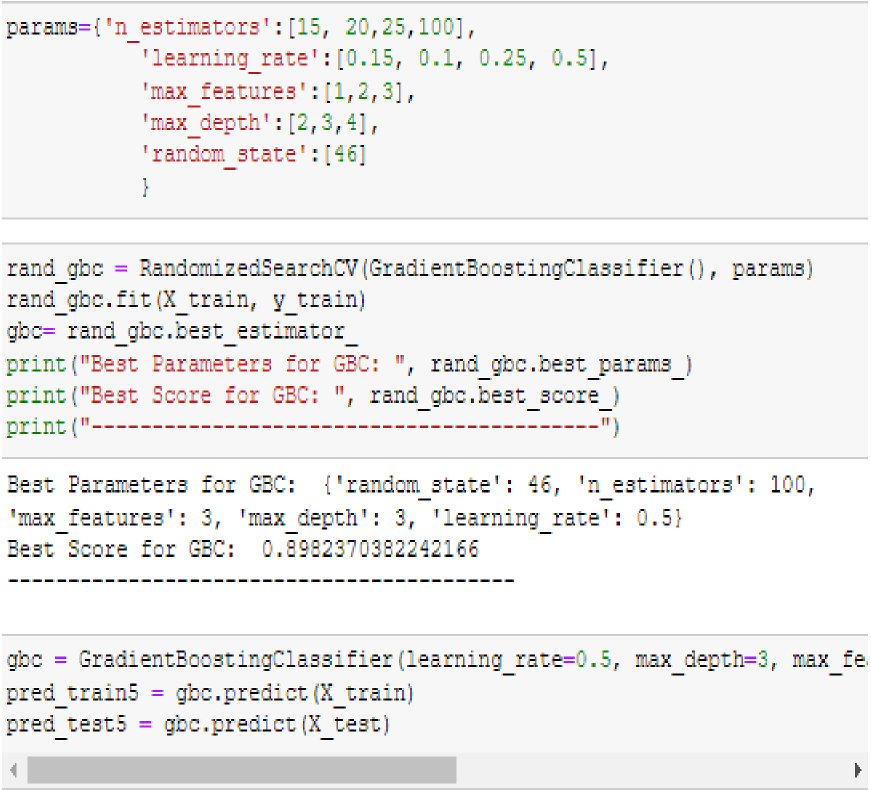


```
# Logistic Regression
log_reg_params = {"penalty": ['l1', 'l2'], 'C': [0.01, 0.1, 1,10,0.001]}
grid_log_reg = GridSearchCV(LogisticRegression(), log_reg_params)
grid_log_reg.fit(X_train, y_train)
# We automatically get the logistic regression with the best parameters.
log_reg = grid_log_reg.best_estimator_
print("Best Parameters for Logistic Regression: ", grid_log_reg.best_pa
print("Best Score for Logistic Regression: ", grid_log_reg.best_score_)
print("------------------------------------------")
```

```
Best Parameters for Logistic Regression:  {'C': 10, 'penalty': 'l2'}
Best Score for Logistic Regression:  0.7780791730106579
------------------------------------------
```

```
lr = LogisticRegression(solver='liblinear',C=10 , penalty= 'l2').fit(X_t
pred_train = lr.predict(X_train)
pred_test = lr.predict(X_test)
```

Model 4: Gradient Boosting Classifier



```
params={'n_estimators':[15, 20,25,100],
           'learning_rate':[0.15, 0.1, 0.25, 0.5],
           'max_features':[1,2,3],
           'max_depth':[2,3,4],
           'random_state':[46]
           }
```

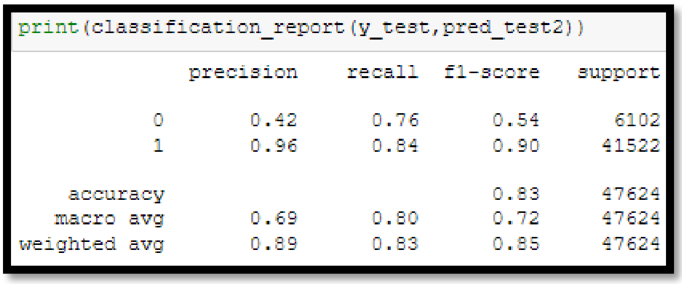
```
rand_gbc = RandomizedSearchCV(GradientBoostingClassifier(), params)
rand_gbc.fit(X_train, y_train)
gbc= rand_gbc.best_estimator_
print("Best Parameters for GBC: ", rand_gbc.best_params_)
print("Best Score for GBC: ", rand_gbc.best_score_)
print("------------------------------------------")
```

```
Best Parameters for GBC:  {'random_state': 46, 'n_estimators': 100,
'max_features': 3, 'max_depth': 3, 'learning_rate': 0.5}
Best Score for GBC:  0.8982370382242166
------------------------------------------
```

```
gbc = GradientBoostingClassifier(learning_rate=0.5, max_depth=3, max_fe
pred_train5 = gbc.predict(X_train)
pred_test5 = gbc.predict(X_test)
```

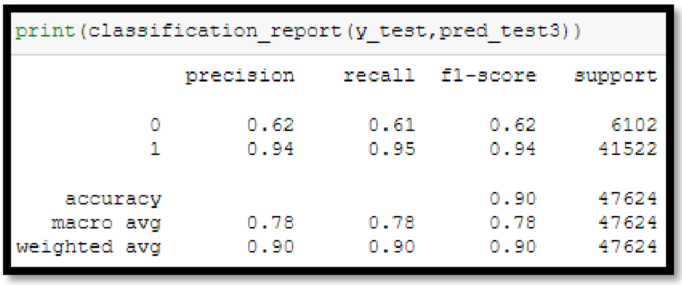
Classification report for the above models:

Model1: Decision Tree Classifier



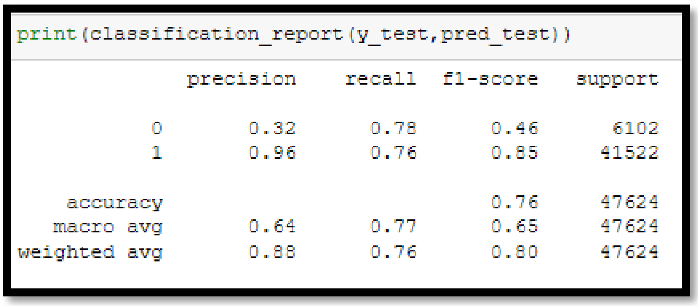
```
print(classification_report(y_test,pred_test2))
              precision    recall  f1-score   support

           0       0.42      0.76      0.54      6102
           1       0.96      0.84      0.90     41522

    accuracy                           0.83     47624
   macro avg       0.69      0.80      0.72     47624
weighted avg       0.89      0.83      0.85     47624
```

Model 2: Random Forest Classifier



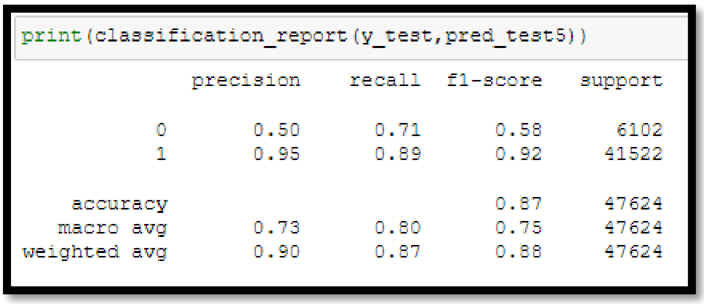
```
print(classification_report(y_test,pred_test3))
              precision    recall  f1-score   support

           0       0.62      0.61      0.62      6102
           1       0.94      0.95      0.94     41522

    accuracy                           0.90     47624
   macro avg       0.78      0.78      0.78     47624
weighted avg       0.90      0.90      0.90     47624
```

Model 3: Logistic Regression



```
print(classification_report(y_test,pred_test))
              precision    recall  f1-score   support

           0       0.32      0.78      0.46      6102
           1       0.96      0.76      0.85     41522

    accuracy                           0.76     47624
   macro avg       0.64      0.77      0.65     47624
weighted avg       0.88      0.76      0.80     47624
```

Model 4: Gradient Boosting Classifier



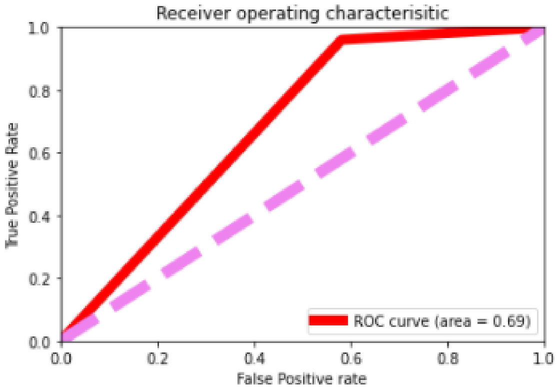
```
print(classification_report(y_test,pred_test5))
              precision    recall  f1-score   support

           0       0.50      0.71      0.58      6102
           1       0.95      0.89      0.92     41522

    accuracy                           0.87     47624
   macro avg       0.73      0.80      0.75     47624
weighted avg       0.90      0.87      0.88     47624
```

Cross Validation Score:

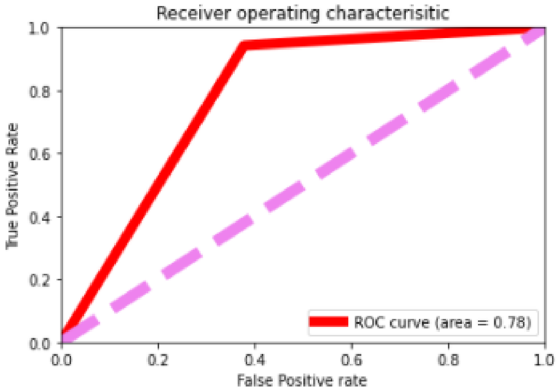
Model	Score
Decision Tree	91.22763732571812
Random Forest	91.44286494204603
Logistic Regression	87.85853771207795
Gradient Boosting Classifier	91.39666974634638

ROC Curve Area:

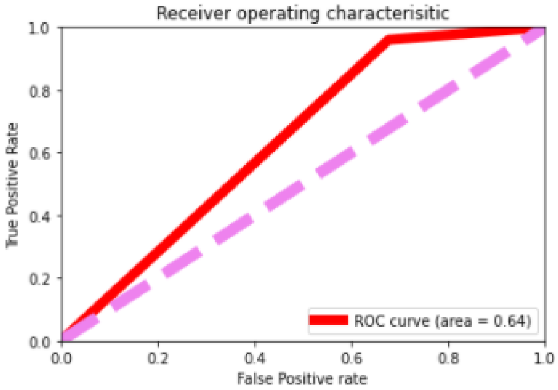
1.



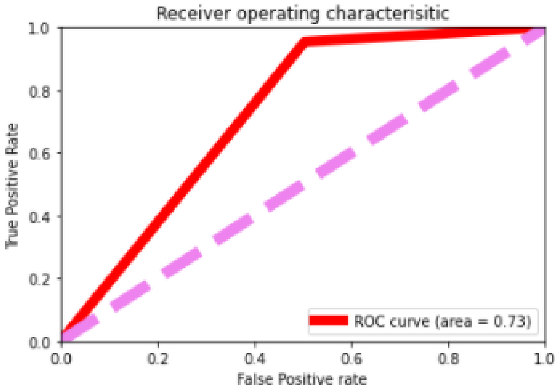
2.



3.



4.



The Area Under the Curve (AUC) is the estimate of the potential of any classifier to differentiate between classes and is used as a summary of the ROC curve statistics.

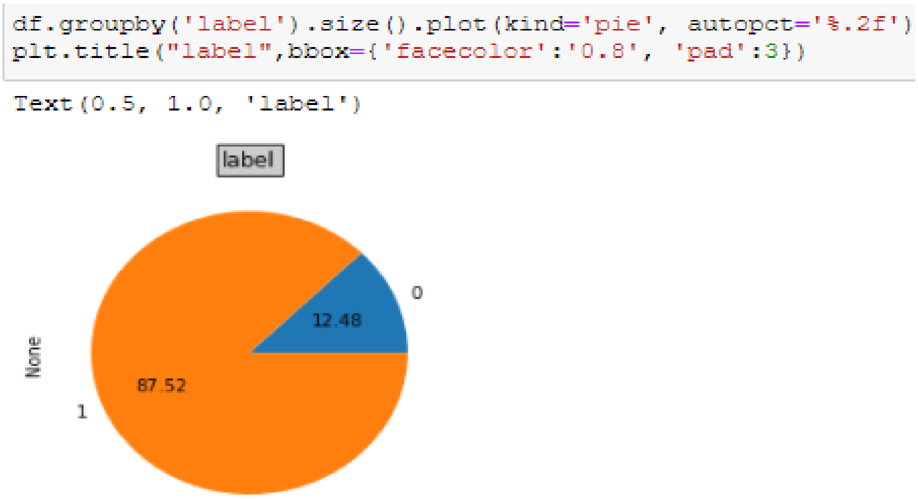
The greater the AUC, the better the performance of the model at differentiating between the binary classes.

Key Metrics for success in solving problem under consideration

- **Hyperparameter Tuning:**
  In the snapshots above, all the models are trained with their parameters tuned using Randomized Search CV, except Random Forest classifier, where it took too long to train the model on hyperparameters, so, model was trained on default values giving us the best accuracy and ROC curve area among all the models.
- The accuracy score, ROC AUC curve was used Accuracy is used when the True Positives and True negatives are more important while F1-score is used when the False Negatives and False Positives are imperative. However, we would be preferring the Accuracy score over F1 score because we are more concerned about the True Positive and True Negative.

Visualizations

Class imbalance:

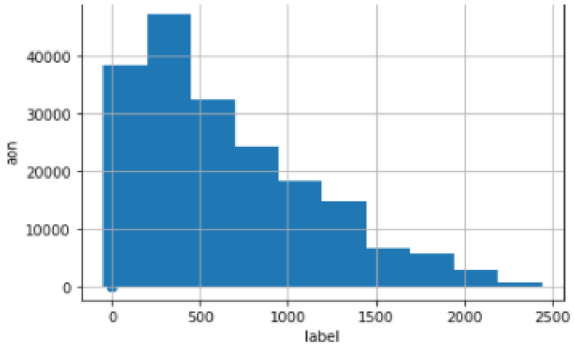


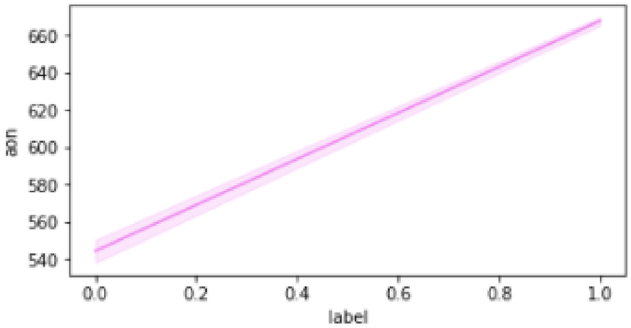
```
df.groupby('label').size().plot(kind='pie', autopct='%.2f')
plt.title("label",bbox={'facecolor':'0.8', 'pad':3})
```

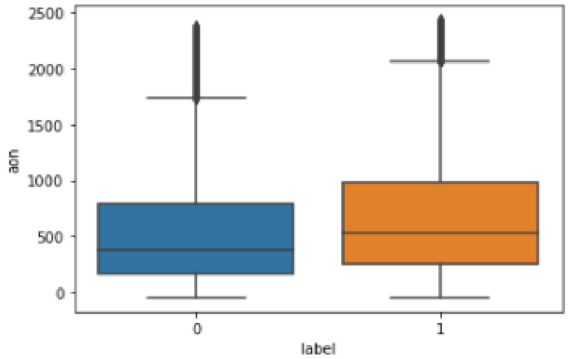
```
Text(0.5, 1.0, 'label')
```

More 1’s than 0’s in the dataset, target column is severely imbalanced

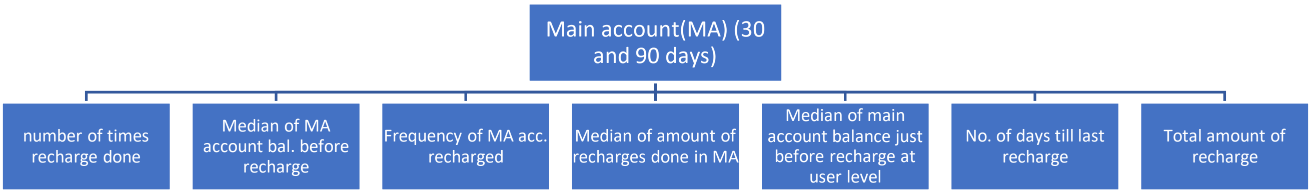
2. Age of network







- Greater the age, lesser default rates.
- Defaulting takes place when median age is around 480days, maximum is 1700 days.
- Majority of the customers are below 1000 days old.



Main account(MA) (30 and 90 days)

number of times recharge done

Median of MA account bal. before recharge

Frequency of MA acc. recharged

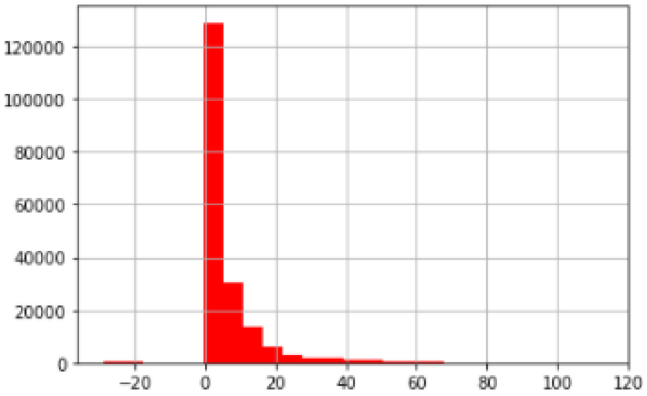
Median of amount of recharges done in MA

Median of main account balance just before recharge at user level

No. of days till last recharge

Total amount of recharge

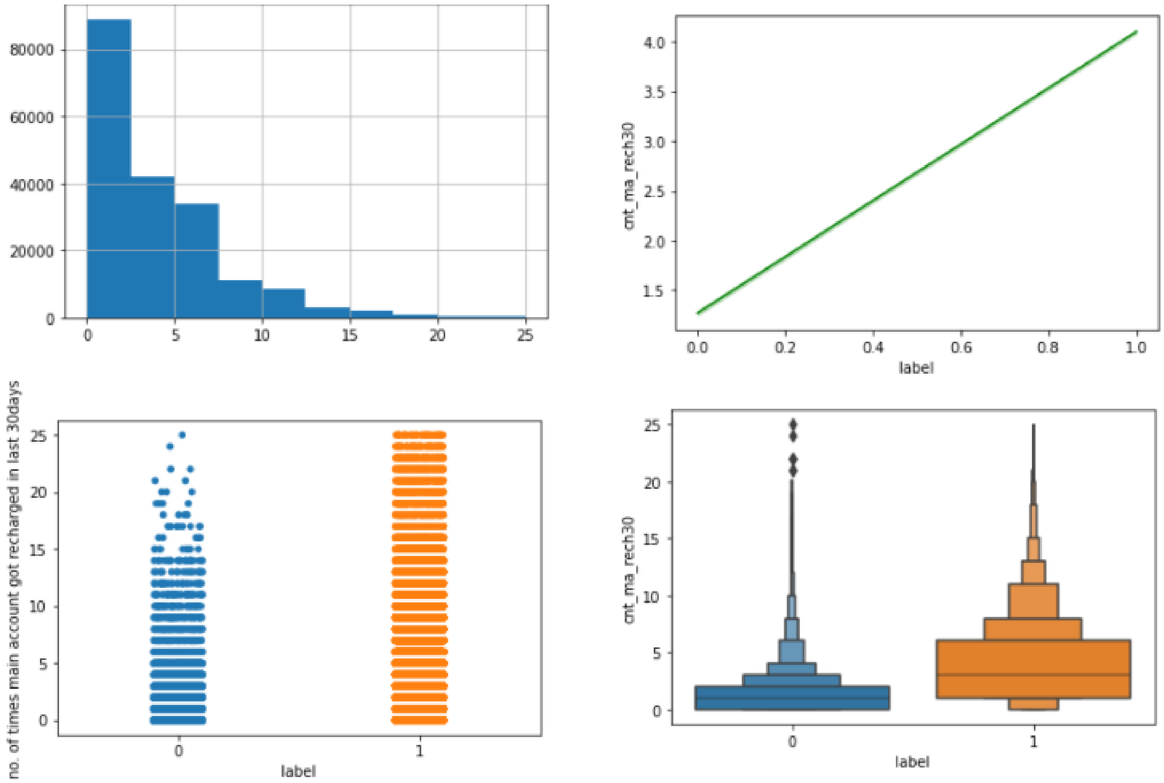
3. Number of days till last recharge of main account



Most of the main accounts were recharged 10 to 12 days before the sample was listed.

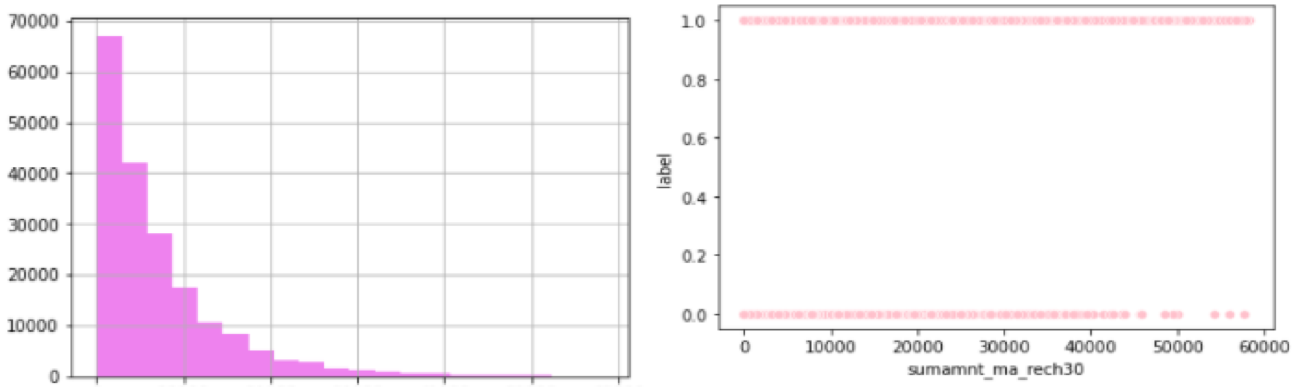
3.1. For the last 30 days:

3.1.1 Number of times main account got recharged



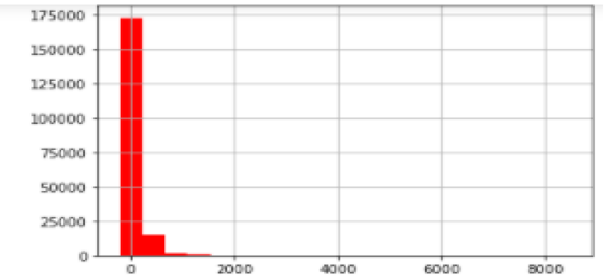
- Maximum number of recharges is between 1 and 4.
- Defaulting is less for people with number of recharges above 20 per month

3.1.2 Total amount of recharge in main account over last 30 days (in Indonesian Rupiah)

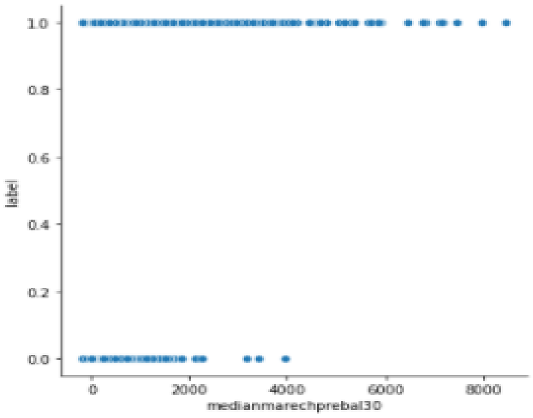


- Maximum recharges were of the amount ranging between 0 to 10k.
- Defaulting becomes less when the recharging amount crosses 45k IDR, meaning people with higher recharges were less likely to default on a micro loan

3.1.3 Median of main account balance just before recharge



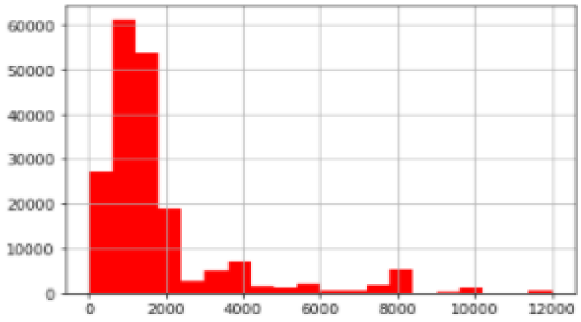
```
<seaborn.axisgrid.FacetGrid at 0x2088e0496a0>
```



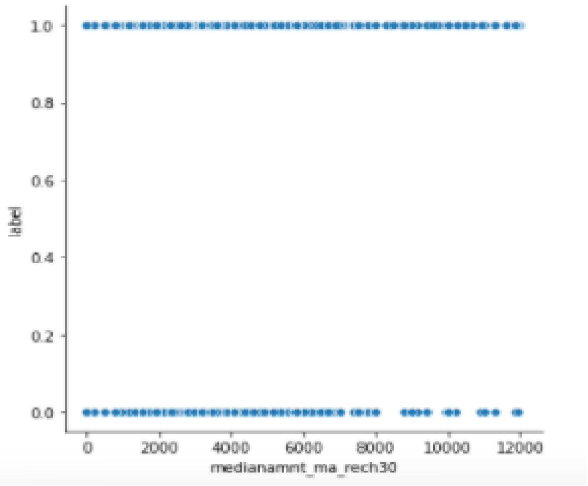
- Median of amount is between 0 and 100
- Lesser chances to default if the median before recharge was above 2500.

3.1.4 Median of amount of recharges done in main account at user level

- Median ranges between 500 to 2200 IDR
- Scatter plot tells us that the median of defaulters is constant till 8000 IDR after which it becomes inconsistent indicating chances of defaulting decreases.

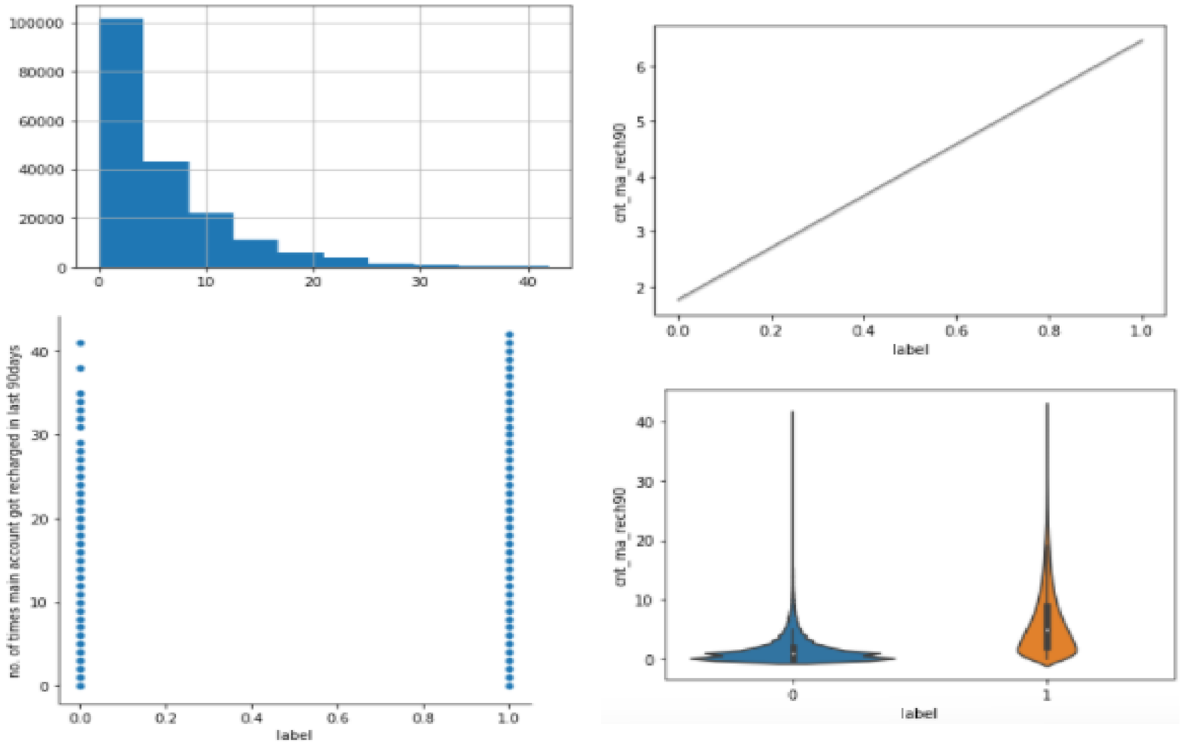


```
<seaborn.axisgrid.FacetGrid at 0x2088c4bb1c0>
```



3.2 Last 90 Dyas

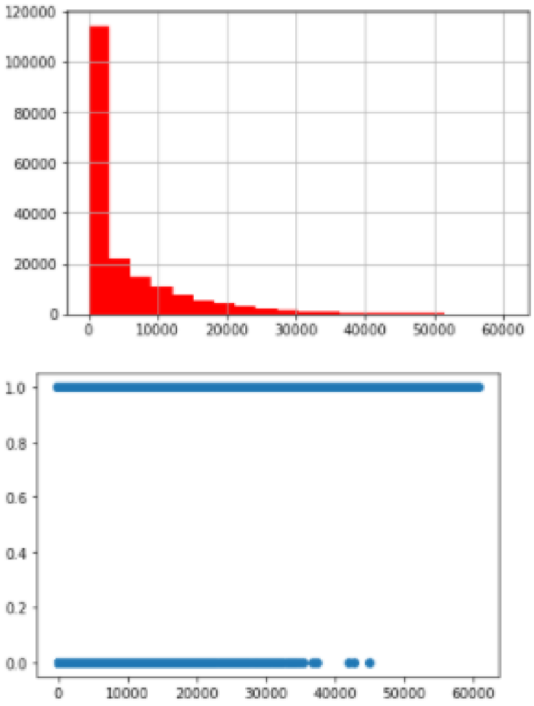
3.2.1 Number of times main account got recharged



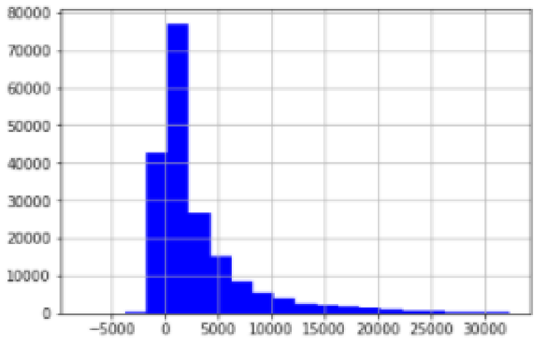
- Majority of the recharges were done below 5
- Defaulters were recharging their account on average only twice in 90 days.
- Defaulting rate decreases when recharging rate is above 30 times in 90 days

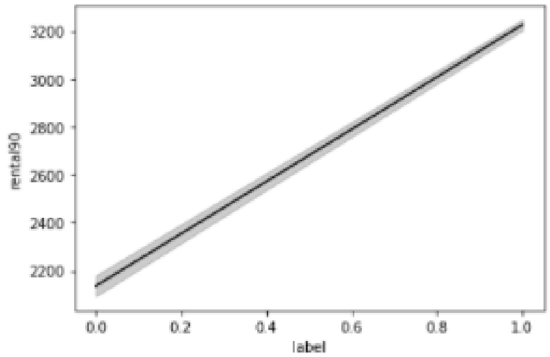
3.2.2. Daily amount spent from main account

- Majority spent between 0 to 2500 daily
- Defaulting reduces when daily amount spent exceeds 38000 IDR



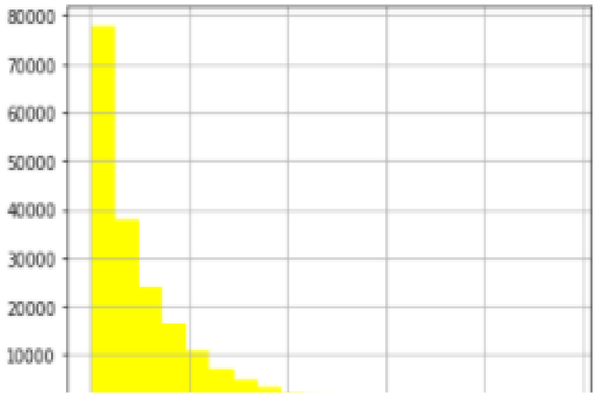
3.2.3 Average Main account balance

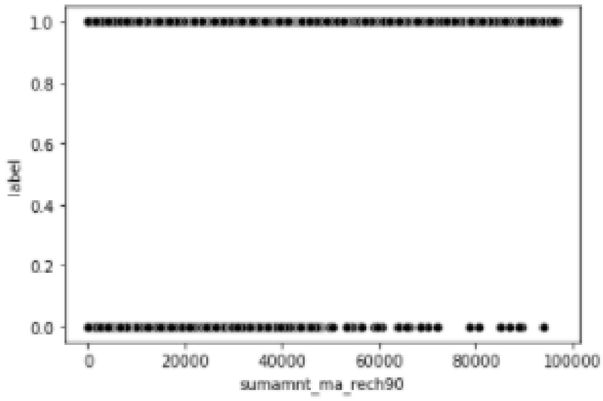




- Defaulters had an account balance below 2600 and some had negative balances as well.
- Non defaulters have average balance of at least 3000 in the last 90 days

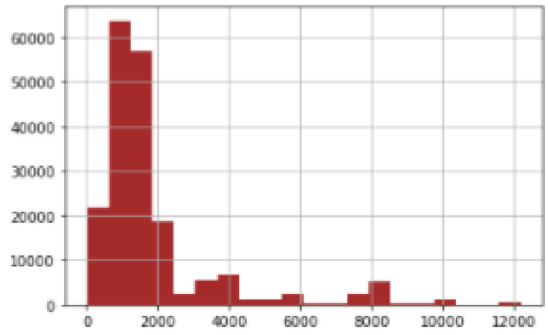
3.2.4 Total amount of recharge



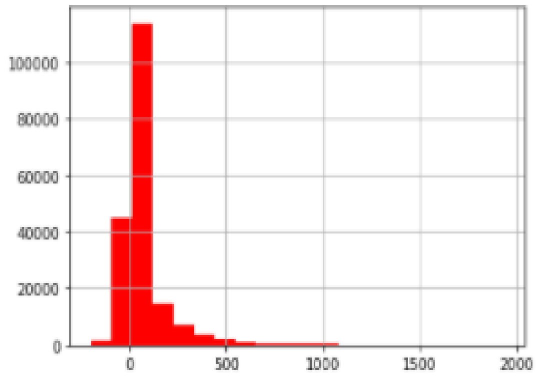


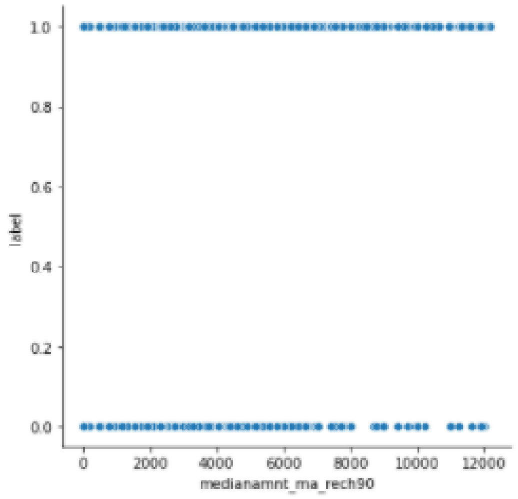
- Majority recharged their accounts with an amount close 10k IDR
- Defaulting diminishes after amount of 50000 IDR recharges done by customers in last 90 days

3.2.5 Median of amount of recharges done at user level and Median of main account balance just before recharge

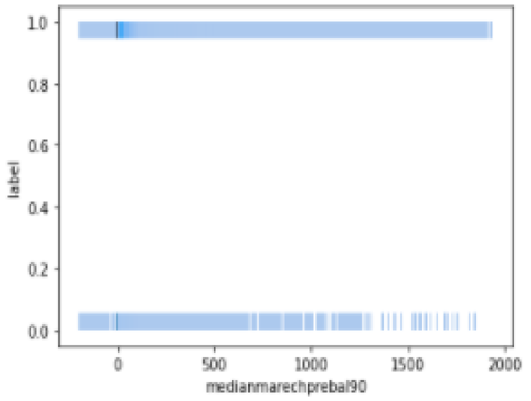


```
<seaborn.axisgrid.FacetGrid at 0x2088c7cb9d0>
```



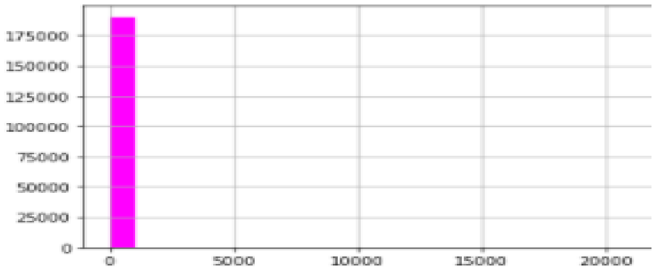


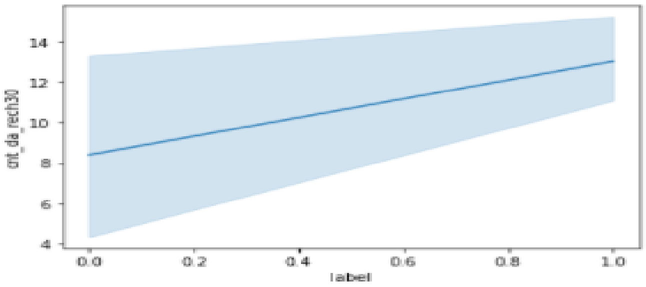
```
<AxesSubplot:xlabel='medianmarechprebal90', ylabel
```

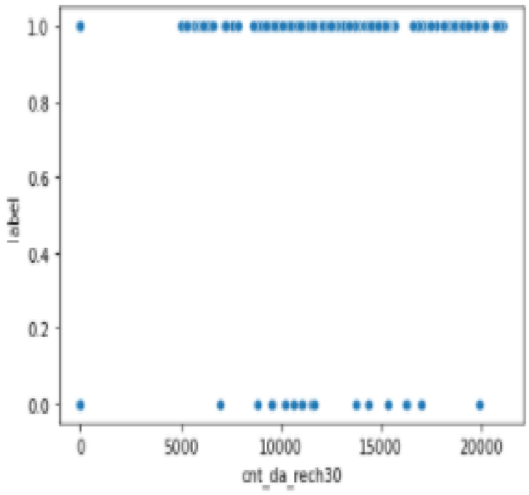


4. **Data Account**

4.1 Number of times data account got recharged in last 30 days

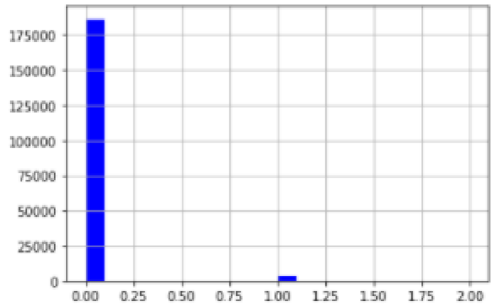


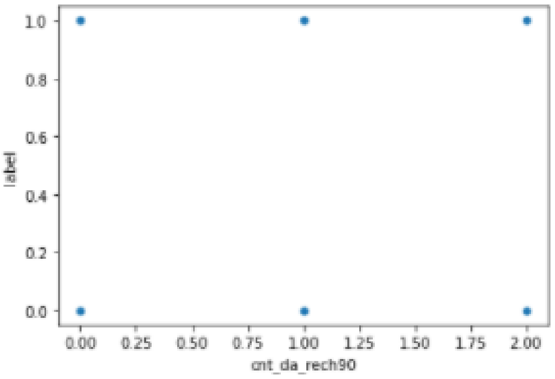


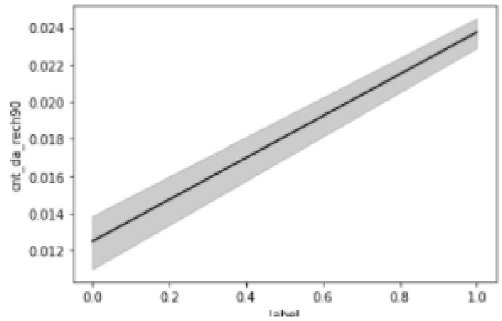


- Customers are defaulting between the recharge amount range of 7000 to 17,500 IDR
- Customers recharging below 8 times in a month are prone to defaulting

4.2 Number of times data account got recharged in last 90 days





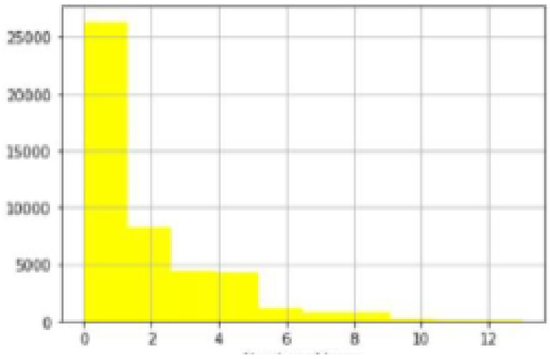


- Majority of the people had 0 recharges in the last 90 days
- No impact on the label.

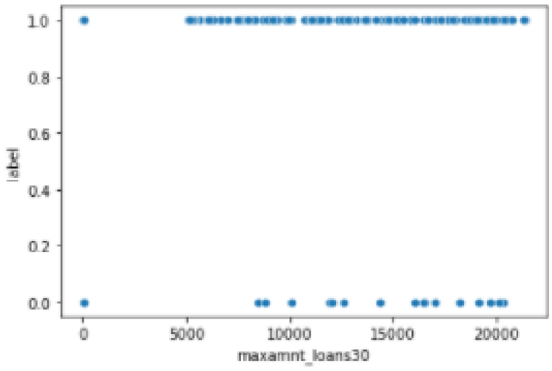
5. Loan

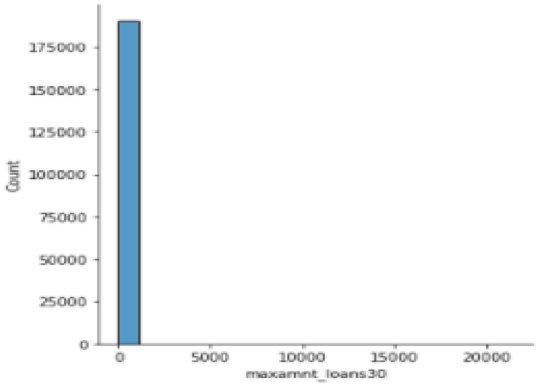
5.1 Number of loans taken by user in last 30 days

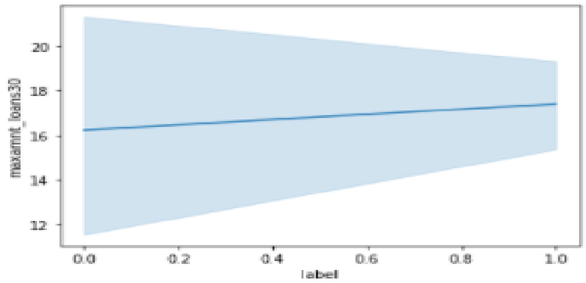
- Majority have taken at loans between 0 and 2



5.2 Maximum amount of loan taken in 30 days

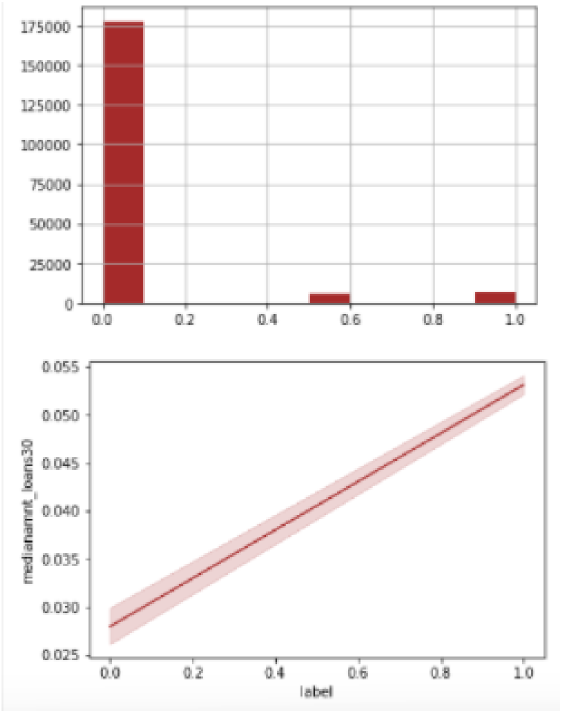






- Defaulting is high above when amount is greater than 16000 IDR
- Only two amounts in loan – 5 and 10 which is why the histogram is centred at 0.

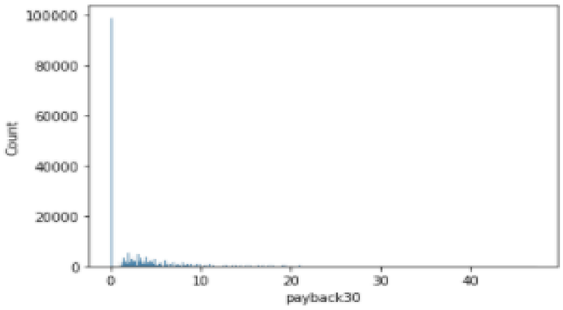
5.3 Median of amounts of loan taken by the user in last 30 days

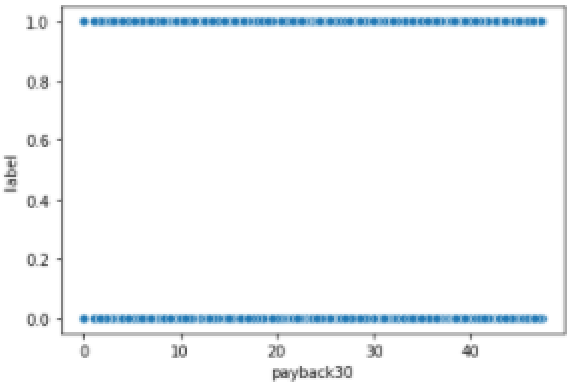


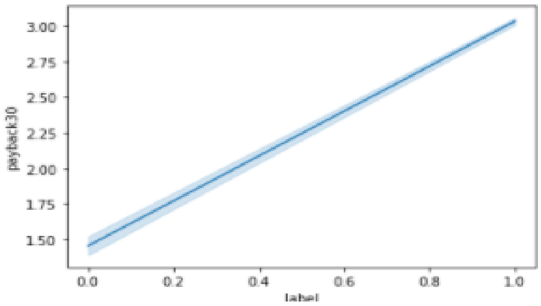
- Median is below 0.1
- Median is almost half for defaulters when compared to non-defaulters.

6. Payback Time

6.1 Average payback time in days over last 30 days

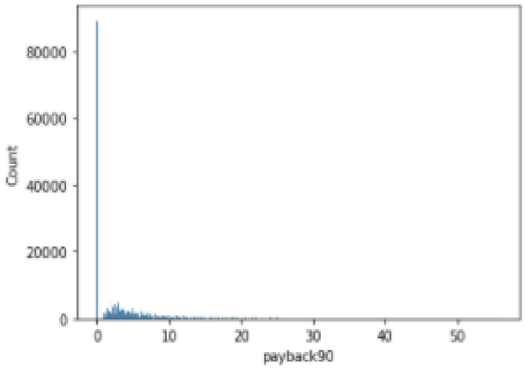


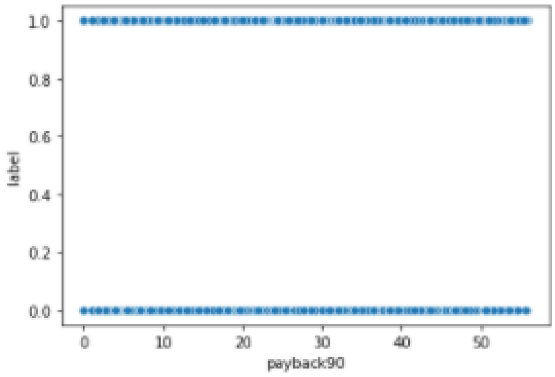


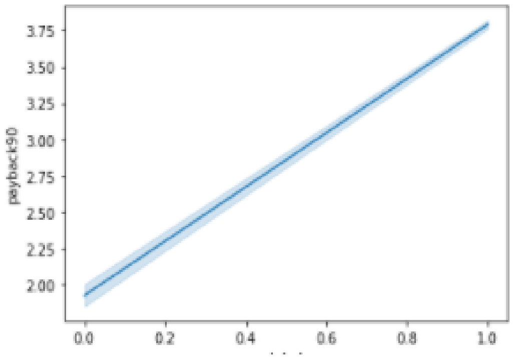


- No impact on label
- Mean Payback time is below 10 days
- Majority(75%) returned their loans before 1 day

6.2 Average payback time in days over last 90 days







- No impact on label
- Most defaulters have a payback time of 2 days for 90 days loan.

CONCLUSION

Key Findings and Conclusions of the Study

This project dealt with helping the company understand how and what kind of factors is leading to customers defaulting on their micro loans. This can help them perform a root cause analysis on how to mitigate such factors and retain non defaulting customers.

Deduction:

1. Greater the age of person being a customer, lesser the default rate.
2. Defaulting takes place when median age is around 480days, maximum is 1700 days.
3. Majority of the customers are below 1000 days old.
4. Most of the main accounts were recharged 10 to 12 days before the sample was listed.
5. Maximum number of recharges is between 1 and 4.
6. Defaulting is less for people with number of recharges above 20 per month
7. Maximum recharges were of the amount ranging between 0 to 10k.
8. Defaulting becomes less when the recharging amount crosses 45k IDR, meaning people with higher recharges were less likely to default on a micro loan
9. Median of amount is between 0 and 100
10. Lesser chances to default if the median before recharge was above 2500.
11. Median ranges between 500 to 2200 IDR
12. Scatter plot tells us that the median of defaulters is constant till 8000 IDR after which it becomes inconsistent indicating chances of defaulting decreases.

    Last 90 days (MA):
13. Majority of the recharges were done below 5
14. Defaulters were recharging their account on average only twice in 90 days.
15. Defaulting rate decreases when recharging rate is above 30 times in 90 days
16. Majority spent between 0 to 2500 daily
17. Defaulting reduces when daily amount spent exceeds 38000 IDR
18. Defaulters had an account balance below 2600 and some had negative balances as well.
19. Non defaulters have average balance of at least 3000 in the last 90 days
20. Majority recharged their accounts with an amount close 10k IDR

21. Defaulting diminishes after amount of 50000 IDR recharges done by customers in last 90 days

    Data Account (DA):

22. Customers are defaulting between the recharge amount range of 7000 to 17,500 IDR
23. Customers recharging below 8 times in a month are prone to defaulting
24. Majority of the people had 0 recharges in the last 90 days
25. No impact of number of recharges in 90 days on the label.

    Loan:

26. Majority have taken at loans between 0 and 2
27. Defaulting is high above when amount is greater than 16000 IDR
28. Only two amounts in loan – 5 and 10 which is why the histogram is centred at 0.
29. Median is below 0.1
30. Median is almost half for defaulters when compared to non-defaulters.

    Average Payback Time:

31. No impact of average payback time in 30 days on label
32. Mean Payback time is below 10 days
33. Majority (75%) returned their loans before 1 day
34. No impact of average payback time in 90 days on label
35. Most defaulters have a payback time of 2 days for 90 days loan.

Learning Outcomes of the Study in respect of Data Science

This dataset shed light on the power of visualization and EDA. As the data is largely imbalanced which makes it expectedly tough to perform exploratory data analysis and reach any fruitful conclusions. Also, the data description had a fair share of flaws in nomenclature which made it exceptionally difficult to make any sense of the data. Some features were redundant making the process lengthy for the analyst and people who were at the forefront of data collection. But it definitely proved to be a wonderful learning exercise and learnt about techniques that I will definitely use in later projects with large datasets having at least a million samples.

Limitations of this work and Scope for Future Work

This project is somewhat limited as the quality of data is not up to the mark and does not cover various facets such as less samples, data available for only one year, not much information about user information except a phone number.

With larger models' cloud and large-scale data libraries such as Pandas can be used to create models and further expand the scope of the project to include various social classes of the population. Features can be added to express its users in different ways and help the company form strategies to market themselves to those sect of customers