MICRO CREDIT DEFAULTER MODEL

Submitted By:

Junaid Shaikh

ACKNOWLEDGMENT

I would like to thank Flip Robo Technologies for providing me with the opportunity to work on this project from which I have learned a lot.
Most of the concepts used to predict the Micro-Credit loan defaulters are learned from Data Trained Institute and below documentations.

Some of the reference sources are as follows:

- Medium.com
- StackOverflow

INTRODUCTION

BUSINESS PROBLEM FRAMING

A Microfinance Institution (MFI) is an organization that offers financial services to low-income populations. MFS becomes very useful when targeting especially the unbanked poor families living in remote areas with not much sources of income. The Microfinance services (MFS) provided by MFI are Group Loans, Agricultural Loans, Individual Business Loans and so on. Many microfinance institutions (MFI), experts and donors are supporting the idea of using mobile financial services (MFS) which they feel are more convenient and efficient, and cost saving, than the traditional high-touch model used since long for the purpose of delivering microfinance services. Though, the MFI industry is primarily focusing on low-income families and are very useful in such areas, the implementation of MFS has been uneven with both significant challenges and successes. Today, microfinance is widely accepted as a poverty-reduction tool, representing $70 billion in outstanding loans and a global outreach of 200 million clients. We are working with one such client that is in Telecom Industry. They are a fixed wireless telecommunications network provider. They have launched various products and have developed its business and organization based on the budget operator model, offering better products at Lower Prices to all value conscious customers through a strategy of disruptive innovation that focuses on the subscriber. They understand the importance of communication and how it affects a person's life, thus, focusing on providing their services and products to low-income families and poor customers that can help them in the need of hour. They are collaborating with an MFI to provide micro-credit on mobile balances to be paid back in 5 days. The Consumer is believed to be defaulter if he deviates from the path of paying back the loaned amount within the time duration of 5 days. For the loan amount of 5 (in Indonesian Rupiah), payback amount should be 6 (in Indonesian Rupiah), while, for the loan amount of 10 (in Indonesian Rupiah), the payback amount should be 12 (in Indonesian Rupiah). Using the historical data of the customer on their recharges, we will be predicting the defaulters with the help of Machine Learning models

Conceptual Background OF The Domain Problem

Telecom Industries understand the importance of communication and how it affects a person's life, thus, focusing on providing their services and products to low-income families and poor customers that can help them in the need of hour.

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

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

We must build a model which can be used to predict in terms of a probability for each loan transaction, whether the customer will be paying back the loaned amount within 5 days of insurance of loan. In this case, Label '1' indicates that the loan has been paid i.e., non-defaulter, while Label '0' indicates that the loan has not been paid i.e., defaulter.

Review OF Literature

In this case we will study different variables and how these independent variables are related with dependent variables and how this will help us to predict whether the customer will become defaulter or not using different machine learning model and thus selecting the final model that giving us best score.

Motivation OF The Problem Undertaken

The main objective behind doing this project is to make an understanding of the micro financial services that are widely accepted nowadays as a poverty reduction tool. They also focus primarily on low-income families and remote areas. Hope this analysis may help micro financial industries to deliver more offers and help more unbanked poor families

Analytical Problem Framing

Mathematical/ Analytical Modeling OF The Problem

In this project we have performed various mathematical and statistical analysis such as we checked description or statistical summary of the data using describe, checked correlation using corr and visualized it using heatmap.

The given dataset has 209593 rows and 36 rows. Label as the target column containing two classes Label '1' indicates that the loan has been paid i.e., non-defaulter, while Label '0' indicates that the loan has not been paid i.e., defaulter. Hence it is a binary classification problem and classification algorithms will be used while building the model. There are no null values in the dataset. It was observed that more than 90% zero values of some columns, if kept, then it will create high skewness in the model hence decided to drop those columns. To get better insight on the features different visualization tools have been used like distribution plot, bar plot and count plot. Outliers and skewness were detected in the dataset which were then reduced using percentile method and yeo-Johnson method respectively. I have used all the classification algorithms while building model then hyper-tunned the best model and saved the best model. At last, I have predicted the label using saved model.

Data Sources & Data Formats

The data was provided in csv (comma separated values) format.

The given dataset has 209593 rows and 36 rows. There are no null values in the dataset.

Dataset was imported using Panda's library and then transformed into data-frame.

Dataset

	label	msisdn	aon	daily_decr30	daily_decr90	rental30	rental90	last_rech_date_ma	last_rech_date_da	last_rech_amt_ma	cnt_ma_rech30
0	0	21408I70789	272.0	3055.050000	3065.150000	220.13	260.13	2.0	0.0	1539	2
1	1	76462I70374	712.0	12122.000000	12124.750000	3691.26	3691.26	20.0	0.0	5787	1
2	1	17943I70372	535.0	1398.000000	1398.000000	900.13	900.13	3.0	0.0	1539	1
3	1	55773I70781	241.0	21.228000	21.228000	159.42	159.42	41.0	0.0	947	0
4	1	03813I82730	947.0	150.619333	150.619333	1098.90	1098.90	4.0	0.0	2309	7
...	...	...	...	...	...	...	...	...	...	...	...
209588	1	22758I85348	404.0	151.872333	151.872333	1089.19	1089.19	1.0	0.0	4048	3
209589	1	95583I84455	1075.0	36.936000	36.936000	1728.36	1728.36	4.0	0.0	773	4
209590	1	28556I85350	1013.0	11843.111667	11904.350000	5861.83	8893.20	3.0	0.0	1539	5
209591	1	59712I82733	1732.0	12488.228333	12574.370000	411.83	984.58	2.0	38.0	773	5
209592	1	65061I85339	1581.0	4489.362000	4534.820000	483.92	631.20	13.0	0.0	7526	2

209593 rows × 36 columns

Features Information:

1. label: Flag indicating whether the user paid back the credit amount within 5 days of issuing the l oan {1: success, 0: failure}
2. msisdn: mobile number of users
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 Indonesi an Rupiah)
14. medianamnt_ma_rech30: Median of amount of recharges done in main account over last 30 da ys at user level (in Indonesian Rupiah)
15. medianmarechprebal30: Median of main account balance just before recharge in last 30 days a t 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 Indonasi an Rupiah)
19. medianamnt_ma_rech90: Median of amount of recharges done in main account over last 90 da ys at user level (in Indonasian Rupiah)
20. medianmarechprebal90: Median of main account balance just before recharge in last 90 days a t user level (in Indonasian 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

Current dataset is raw data. By proper Data Transformation methods, a lot of valuable insights can be gained.

Then statistical analysis was done by checking shape, value counts, info etc.….

Then while looking into the value counts, I found some columns with more than 90% data having same values, this creates skewness in the model and there are chances of getting model bias, so I have dropped those columns with more than 90% same values.

While checking for null values I found no null values in the dataset.

I have also dropped Unnamed:0, msisdn and pcircle column as I found they are useless.

Next as a part of feature extraction I converted the pdate column to pyear, pmonth and pday. Thinking that this data will help us more than pdate.

In some columns I found negative values which were unrealistic, so I have converted those negative values to positive using abs command.

I have also dropped columns like pyear, pdate, pday & last_rech_date_ma.

As well I have dropped all the data with amnt_loans90=0 as it gives the persons who have not taken any loans.

In this project we have performed various mathematical and statistical analysis such as description or statistical summary of the da

ta using describe, checked correlation using corr and visualized it using heatmap. Then we have used Z-Score to plot outliers and remove them.

```
df.describe()
```

	label	aon	daily_decr30	daily_decr90	rental30	rental90	last_rech_date_ma	last_rech_date_da	last_rech_amt_n
count	209593.000000	209593.000000	209593.000000	209593.000000	209593.000000	209593.000000	209593.000000	209593.000000	209593.000000
mean	0.875177	8112.343445	5381.402289	6082.515068	2692.581910	3483.406534	3755.847800	3712.202921	2064.452797
std	0.330519	75696.082531	9220.623400	10918.812767	4308.586781	5770.461279	53905.892230	53374.833430	2370.786034
min	0.000000	-48.000000	-93.012667	-93.012667	-23737.140000	-24720.580000	-29.000000	-29.000000	0.000000
25%	1.000000	246.000000	42.440000	42.692000	280.420000	300.260000	1.000000	0.000000	770.000000
50%	1.000000	527.000000	1469.175667	1500.000000	1083.570000	1334.000000	3.000000	0.000000	1539.000000
75%	1.000000	982.000000	7244.000000	7802.790000	3356.940000	4201.790000	7.000000	0.000000	2309.000000
max	1.000000	999860.755168	265926.000000	320630.000000	198926.110000	200148.110000	998650.377733	999171.809410	55000.000000

cnt_ma_rech30	fr_ma_rech30	sumamnt_ma_rech30	medianamnt_ma_rech30	medianmarechprebal30	cnt_ma_rech90	fr_ma_rech90	sumamnt_ma_rech9
209593.000000	209593.000000	209593.000000	209593.000000	209593.000000	209593.00000	209593.000000	209593.000000
3.978057	3737.355121	7704.501157	1812.817952	3851.927942	6.31543	7.716780	12396.218352
4.256090	53643.625172	10139.621714	2070.864620	54006.374433	7.19347	12.590251	16857.793882
0.000000	0.000000	0.000000	0.000000	-200.000000	0.00000	0.000000	0.000000
1.000000	0.000000	1540.000000	770.000000	11.000000	2.00000	0.000000	2317.000000
3.000000	2.000000	4628.000000	1539.000000	33.900000	4.00000	2.000000	7226.000000
5.000000	6.000000	10010.000000	1924.000000	83.000000	8.00000	8.000000	16000.000000
203.000000	999606.368132	810096.000000	55000.000000	999479.419319	336.00000	88.000000	953036.000000

Data Inputs-Logic-Output Relationship

Since I had all numerical columns, I have plotted dist. plot to see the distribution of each column data.

I have used box plot for each pair of categorical features that shows the relation between label and independent features. Also, we can observe whether the person pays back the loan within the date based on features.

In maximum features relation with target, I observed non-defaulter count is high compared to defaulters.

Hardware and Software Requirements and Tools Used

Hardware required:

- Processor — core i5 and above
- RAM — 8 GB or above
- SSD — 250GB or above

Software/s required: Anaconda

LIBRARIES:

The tools, libraries, and packages we used for accomplishing this project are pandas, NumPy, matplotlib, seaborn, SciPy, sklearns's, mlxtend, xgboost, joblib.

Through panda's library we loaded our csv file 'Data file' into data frame and performed data manipulation and analysis.

With the help of NumPy we worked with arrays.

With the help of matplotlib and seaborn we did plot various graphs and figures and done data visualization.

Train_test_split is a function in Sklearns's model selection for splitting data arrays into two subsets: for training data and for testing data. With this function, you don't need to divide the dataset manually. By default, Sklearns's Train_test_split will make random partitions for the two subsets.

With sklearns's StandardScaler package we scaled all the feature variables onto single scale. As these columns are different in scale, they are standardized to have common scale while building machine learning model. This is useful when you want to compare data that correspond to different units.

With sklearns's package we imported many regression models, we could obtain cross_val_score, which is an accuracy metric used to evaluate model, we could obtain best parameters of a model using GridsearchCV or RandomizedSearchCV, we could reduce skewness using power transform library of sklearns's.

Model/s Development and Evaluation

Identification of possible problem-solving approaches

For skewness removal I have used power transform method, for scaling down I have used standard scaling. Class imbalance is handled by over sampling.

In [77]:
```
#removing skewness by power transform
from sklearn.preprocessing import power_transform
x=power_transform(x,method='yeo-johnson')
```

Scaling

In [78]:
```
#applying standard scaling method on X parameters
from sklearn.preprocessing import StandardScaler
scalar=StandardScaler()
x=scalar.fit_transform(x)
```

Class imbalance removal

In [79]:
```
#applying over sampling on X and Y parameters
from imblearn.over_sampling import SMOTE
NR=SMOTE()
x_over,y_over=NR.fit_resample(x,y)
```

Apart from this multicollinearity refers to the collinearity between the features. Multicollinearity occurs when our model includes multiple factors that are correlated with each other's other than with label. It makes more difficult for the model predict and affects the accuracy. They are treated using PCA (principlecomponent analysis) method. This algorithm reduces the no. of columns by removing highly correlated feature columns.

Testing of Identified Approaches (Algorithms)

Since label was my target and it was a classification column with 0-defaulter and 1-non-defaulter, so this problem was Classification problem. And I have used all Classification algorithms to build my model. By looking into the difference of accuracy score and cross validation score I found RandomForestClassifier as a best model with least difference. Also, to get the best model we must run through multiple models and to avoid the confusion of overfitting we have go through cross validation.

Run & evaluate selected models

```
models=[LogisticRegression(),DecisionTreeClassifier(),KNeighborsClassifier(),RandomForestClassifier(),SVC(),RidgeClassifier(),
        BaggingClassifier(),GradientBoostingClassifier(),SGDClassifier(),
       LGBMClassifier(),XGBClassifier(),ExtraTreesClassifier(),AdaBoostClassifier(),
       QuadraticDiscriminantAnalysis(),CalibratedClassifierCV(),LinearSVC(),NuSVC(),
       LinearDiscriminantAnalysis(),RidgeClassifierCV(),GaussianNB(),BernoulliNB(),
        PassiveAggressiveClassifier(),Perceptron(),DummyClassifier()]
```

```
for i in models:
    x_train,x_test,y_train,y_test=train_test_split(principalComponents,y,random_state = 97,test_size=0.20,
                                               stratify=y)
    scores=cross_val_score(i,x_train,y_train,cv=5,scoring='roc_auc')
    score=np.mean(scores)
    i.fit(x_train,y_train)
    y_pred=i.predict(x_test)
    if roc_auc_score(y_test,y_pred)>score:
        diff=roc_auc_score(y_test,y_pred)-score
        print('roc bigger')
    else:
        diff=score-roc_auc_score(y_test,y_pred)
    print('*'*10)
    print(i)
    print('score',score)
    print('roc',roc_auc_score(y_test,y_pred))
    print('diff',diff)
```

Key Metrics for success in solving problem under consideration

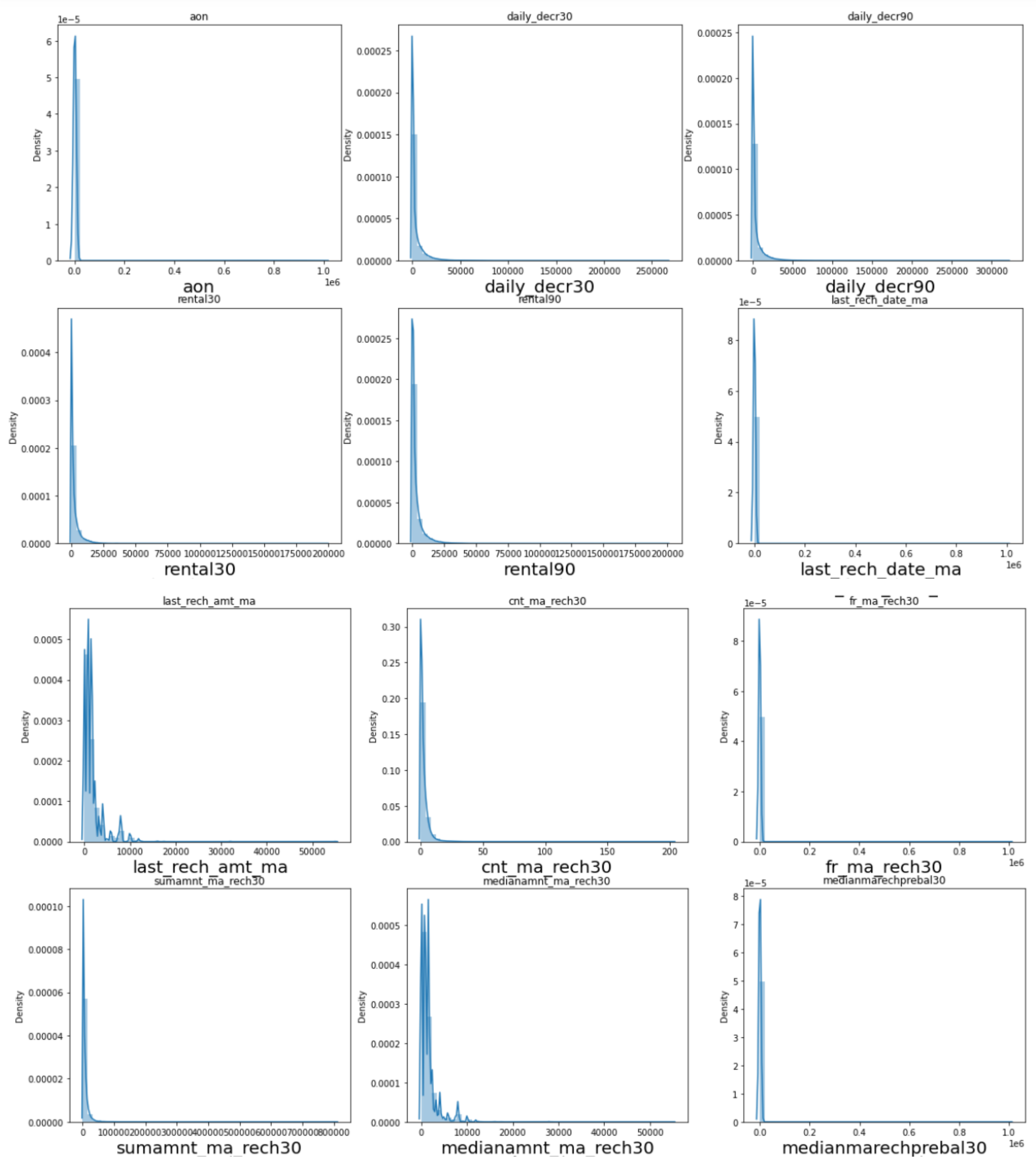
Following metrics were used to evaluate our model:

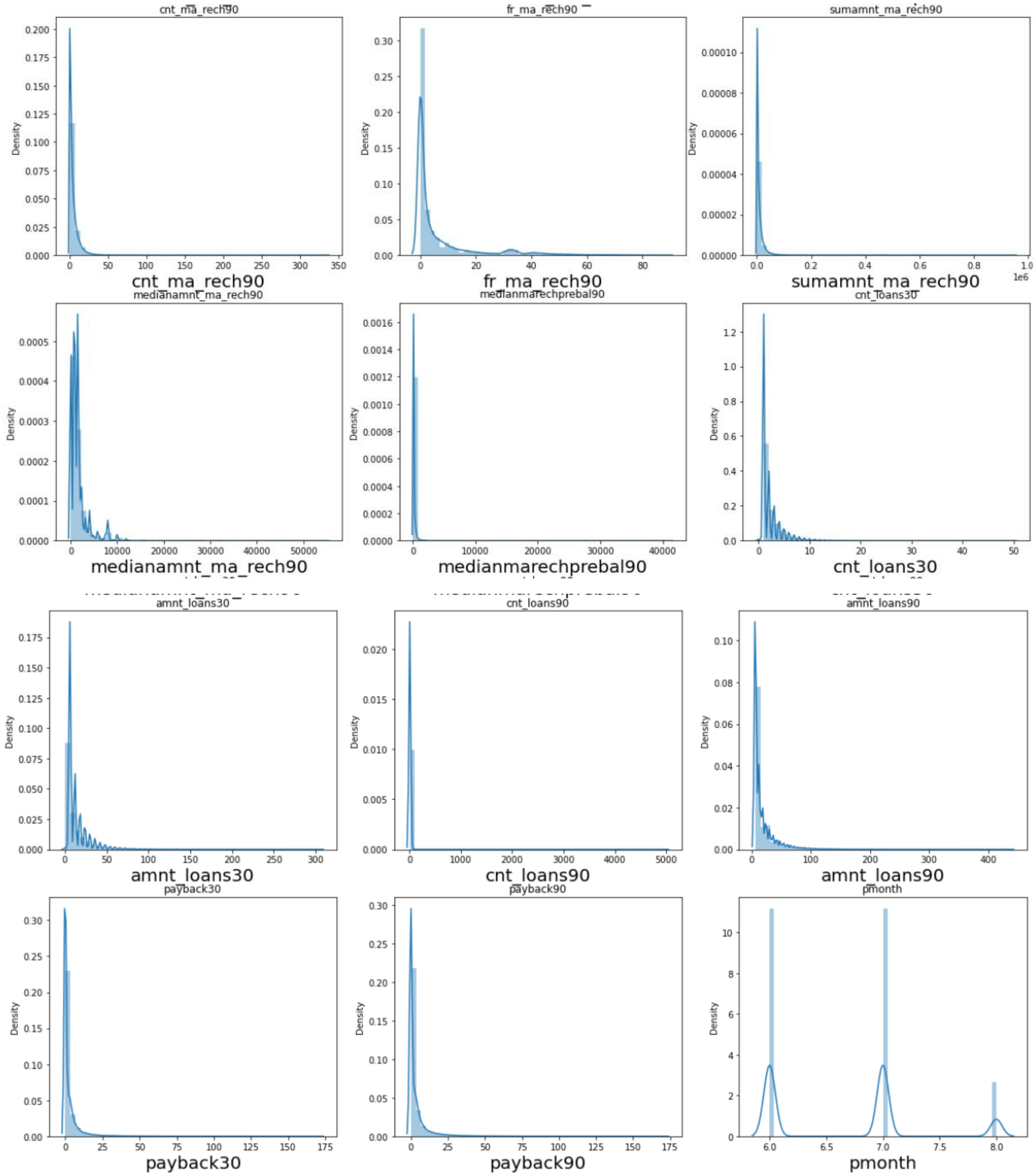
--- Cross Val Score
--- AUC ROC Score
--- Standard deviation error
--- F1 score
--- Confusion matrix
--- Classification report

Visualization

Univariate Analysis:

Distribution plot of all columns





Observations:

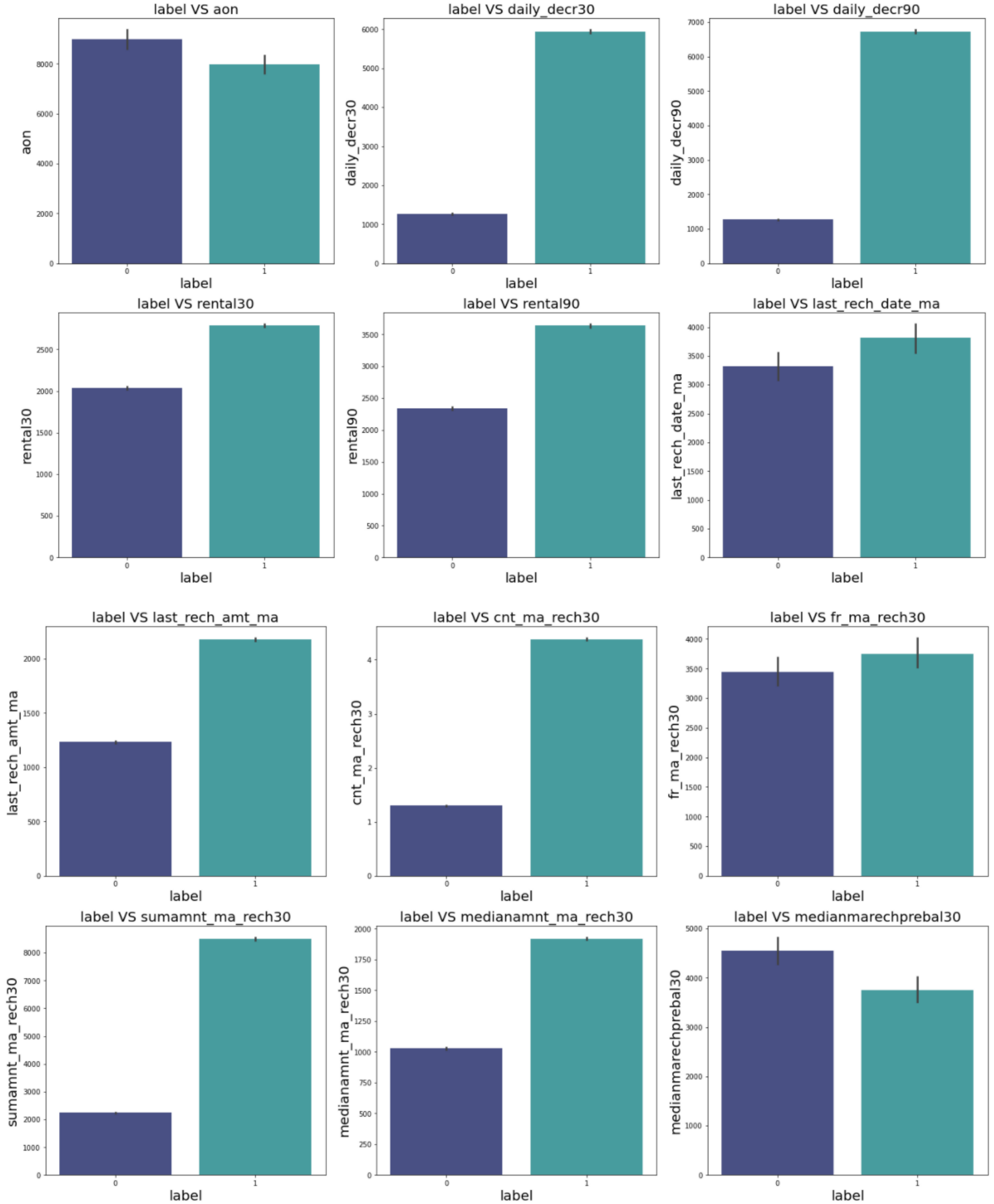
there is skewness in most of the columns, so we must treat them.

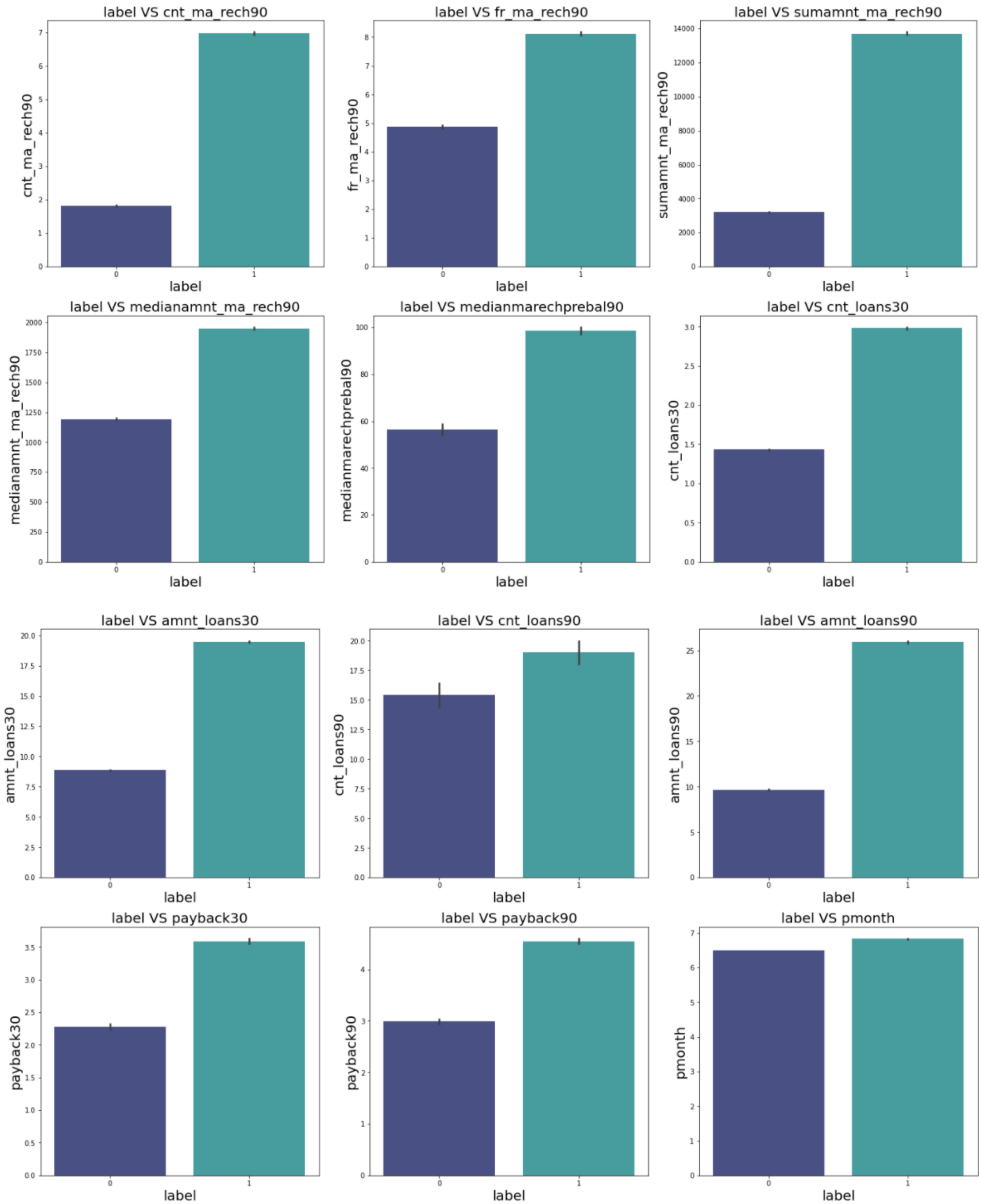
apart from pmonth rest all columns have datapoints in normal distribution (with some distortion also)

pmonth has bimodal plot distribution

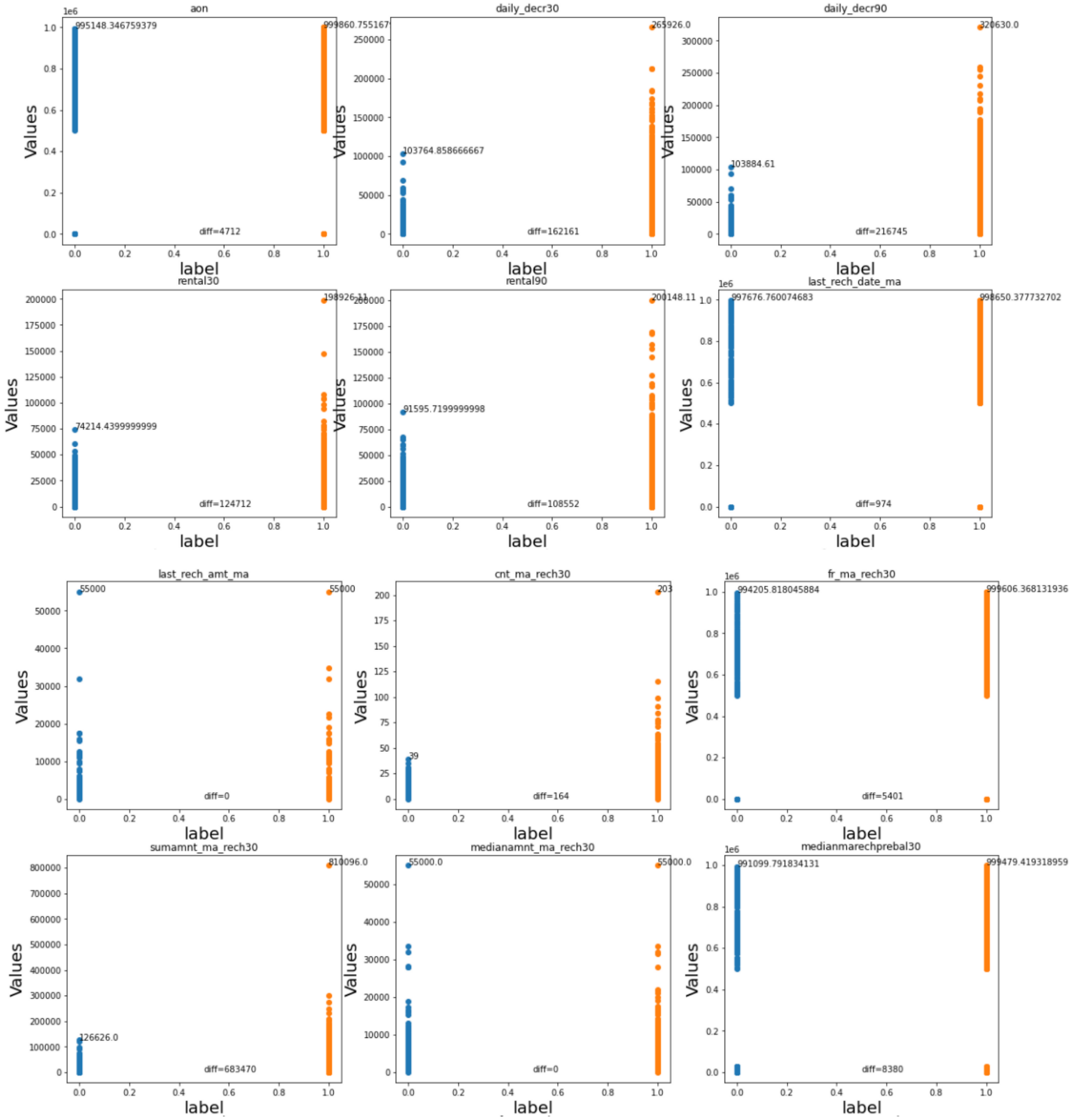
Bivariate analysis

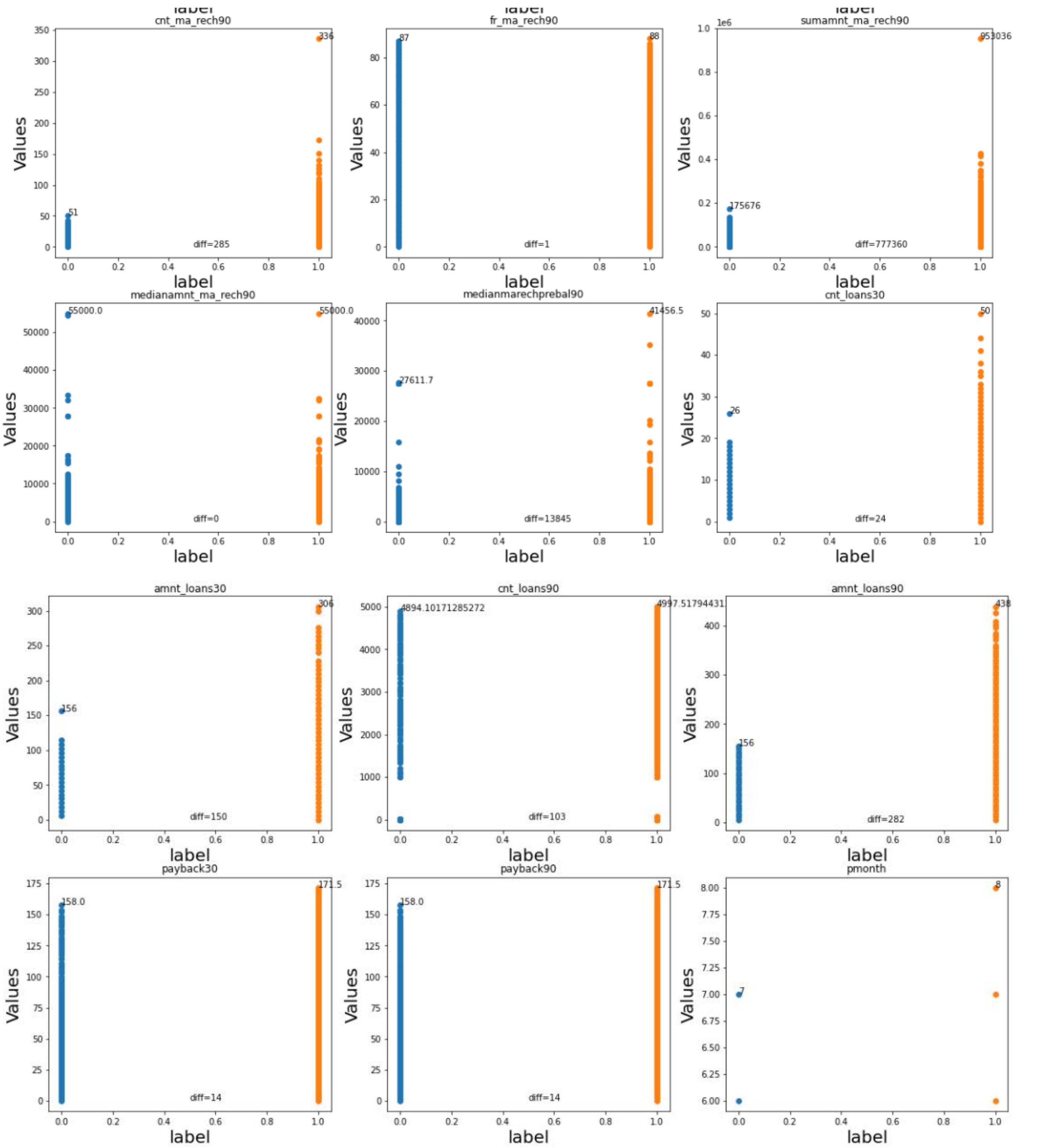
Bar Graph of every column




label VS cnt_ma_rech90
cnt_ma_rech90
label
label VS fr_ma_rech90
fr_ma_rech90
label
label VS sumamnt_ma_rech90
sumamnt_ma_rech90
label
label VS medianamnt_ma_rech90
medianamnt_ma_rech90
label
label VS medianmarechprebal90
medianmarechprebal90
label
label VS cnt_loans30
cnt_loans30
label
label VS amnt_loans30
amnt_loans30
label
label VS cnt_loans90
cnt_loans90
label
label VS amnt_loans90
amnt_loans90
label
label VS payback30
payback30
label
label VS payback90
payback90
label
label VS pmonth
pmonth
label

Scatter Plot of each column values w.r.t label column





Observation for all above graphs:

People with longer duration of network usage are maximum defaulters

People with higher Median of main account balance just before recharge in last 30 days at user level are maximum defaulters

Peps with high value of Daily amount spent from main account, averaged over last 30 days (daily_decr30) are maximum individuals who pay their loan.

Peps with high value of Daily amount spent from main account, averaged over last 90 days(daily_decr90) are maximum individuals who pay their loan.

Peps with high value of Average main account balance over last 30 days(rental30) are maximum individuals who pay their loan.

Peps with high value of Average main account balance over last 90 days(rental90) are maximum individuals who pay their loan.

Peps with high Number of days till last recharge of main account(last_rech_date_ma) are maximum individuals who pay their loan.

Peps with high value of Amount of last recharge of main account (last_rech_amt_ma) are maximum individuals who pay their loan.

Peps with high value of Number of times main account got recharged in last 30 days(cnt_ma_rech30) are maximum individuals who pay their loan.

Peps with high value of Frequency of main account recharged in last 30 days(fr_ma_rech30) are maximum individuals who pay their loan, and the count is high for defaulters comparatively non-defaulters are more in number.

Peps with high value of Total amount of recharge in main account over last 30 days (sumamnt_ma_rech30) are maximum individuals who pay their loan.

Peps with high value of Median of amount of recharges done in main account over last 30 days at user level (medianamnt_ma_rech30) are maximum individuals who pay their loan.

Peps with high value of Median of main account balance just before recharge in last 30 days at user level (medianmarechprebal30) are maximum individuals who pay their loan.

Peps with high value of Number of times main account got recharged in last 90 days(cnt_ma_rech90) are maximum individuals who pay their loan.

Peps with high value of Frequency of main account recharged in last 90 days(fr_ma_rech90) are maximum individuals who pay their loan.

Peps with high value of Total amount of recharge in main account over last 90 days (sumamnt_ma_rech90) are maximum individuals who pay their loan.

Peps with high value of Median of amount of recharges done in main account over last 90 days at user level (medianamnt_ma_rech90) are maximum individuals who pay their loan.

Peps with high value of Median of main account balance just before recharge in last 90 days at user level (medianmarechprebal90) are maximum individuals who pay their loan.

Peps with high value of Number of loans taken by user in last 30 days(cnt_loans30) are maximum individuals who pay their loan.

Peps with high value of Total amount of loans taken by user in last 30 days(amnt_loans30) are maximum individuals who pay their loan.

Peps with high value of maximum amount of loan taken by the user in last 30 days(maxamnt_loans30) are maximum individuals who pay their loan.

Peps with high value of Number of loans taken by user in last 90 days(cnt_loans90) are maximum individuals who pay their loan.

Peps with high value of Total amount of loans taken by user in last 90 days(amnt_loans90) are maximum individuals who pay their loan.

Peps with high value of maximum amount of loan taken by the user in last 90 days(maxamnt_loans90) are maximum individuals who pay their loan.

Peps with high value of Average payback time in days over last 30 days(payback30) are maximum individuals who pay their loan.

Peps with high value of Average payback time in days over last 90 days(payback90) are maximum individuals who pay their loan.

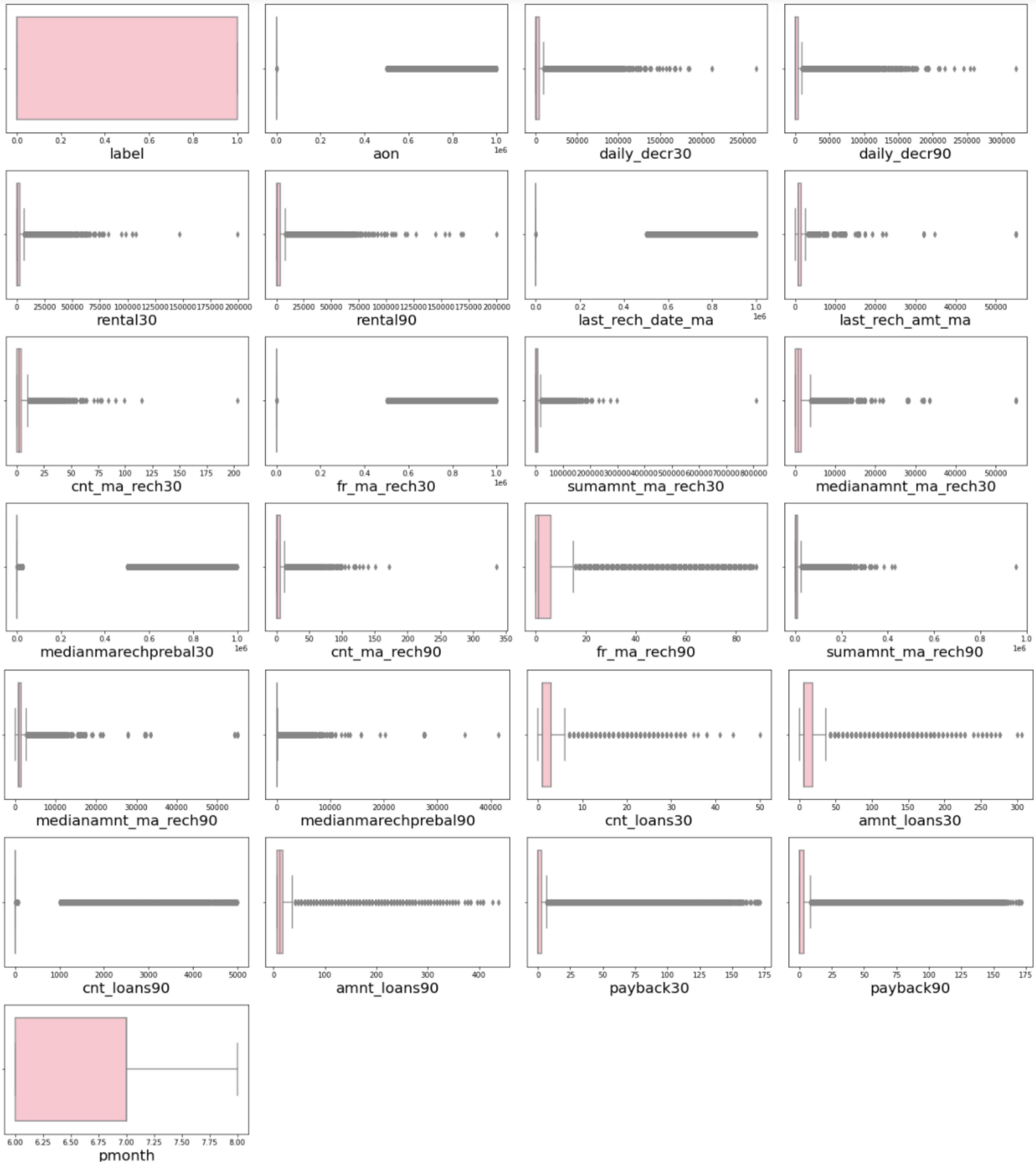
Peps having pmonth 8 have always paid back their loan

From bar graph we can also see there are outliers present

search them and remove them

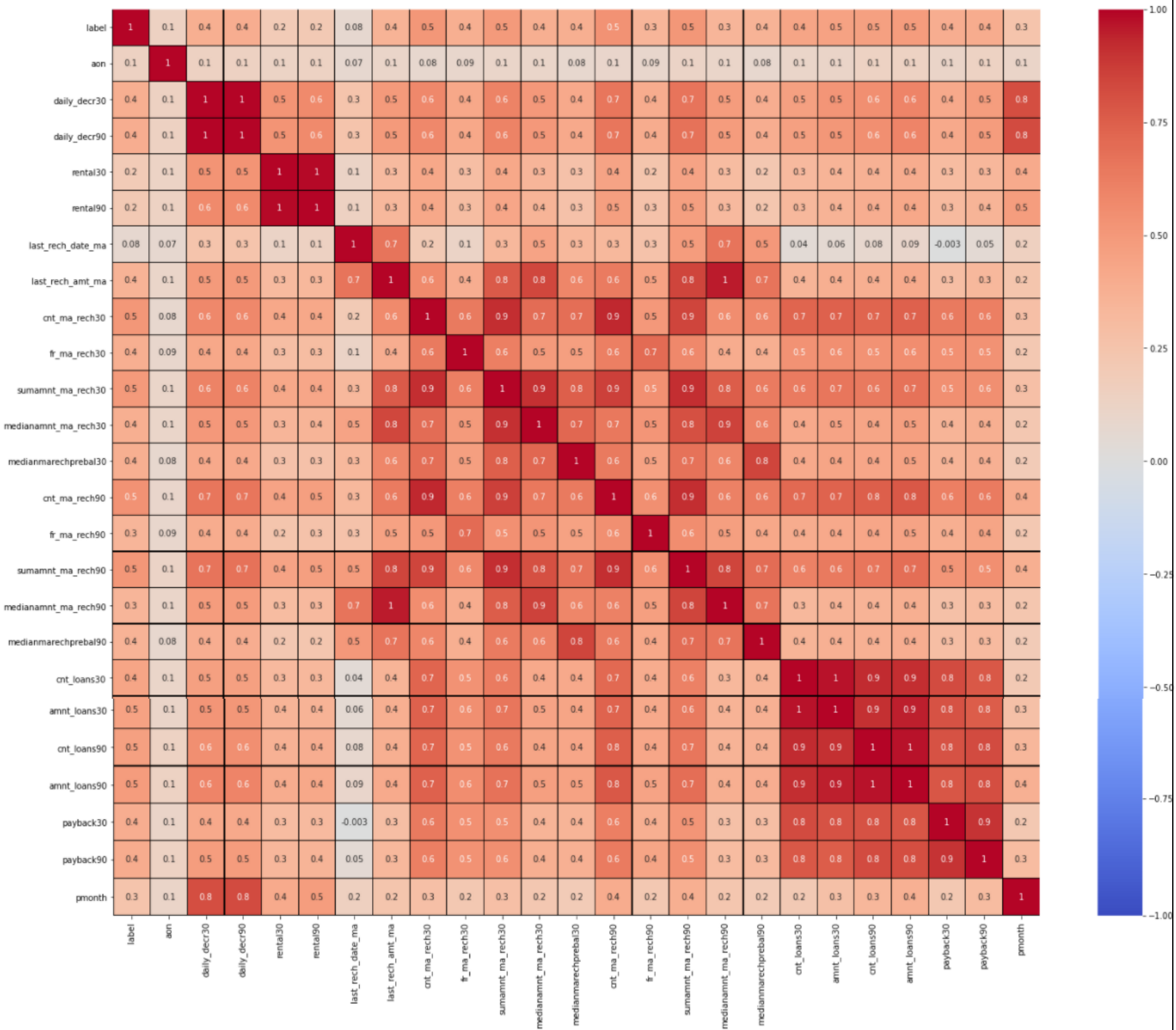
Check For Outliers

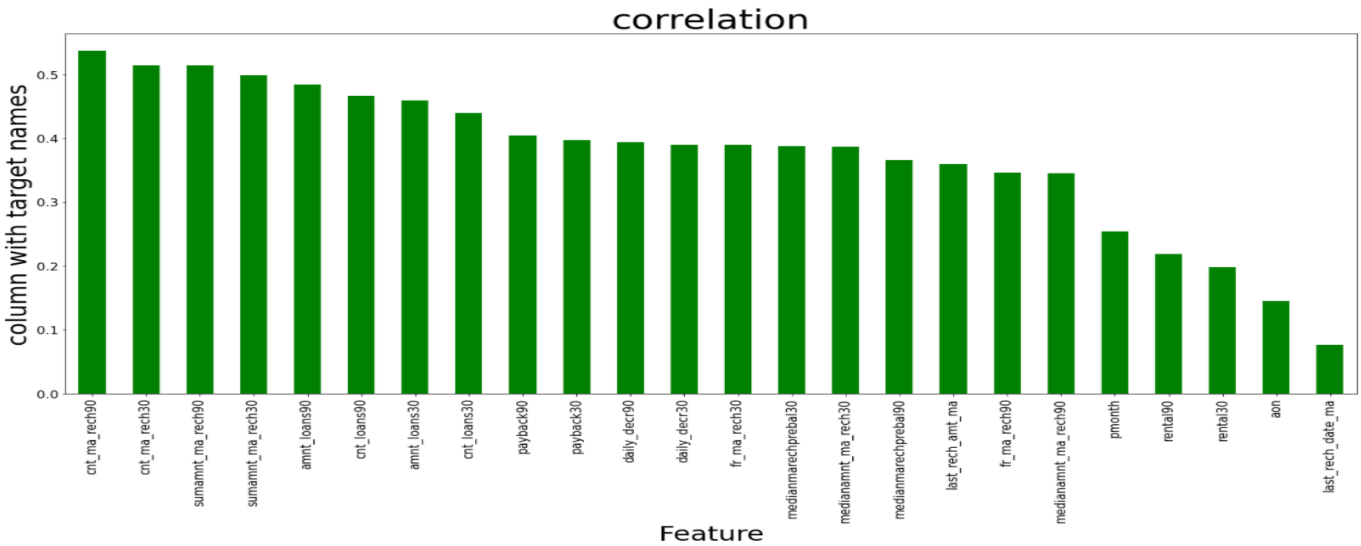
Visualize Boxplot of every column having datapoints of continuous nature



Outliers were detected in almost every column, remove them using percentile method

Check For Correlation





Observation:

Very few columns have high correlation with column label.

drop column last_rech_date_ma as it has extremely low correlation with the target variable label

Model Building

Perform Feature Scaling

Before we start model building, we need to perform feature scaling on all columns, to avoid biasing of data.

Also check for skewness in data and remove it.

```
aon                      1.022268
daily_decr30             2.398688
daily_decr90             2.529767
rental30                 2.196557
rental90                 2.268951
last_rech_date_ma        2.411731
last_rech_amt_ma         2.093678
cnt_ma_rech30            1.694185
fr_ma_rech30             1.992427
sumamnt_ma_rech30        1.996563
medianamnt_ma_rech30     2.296886
medianmarechprebal30     3.006313
cnt_ma_rech90            1.871836
fr_ma_rech90             2.235240
sumamnt_ma_rech90        2.128282
medianamnt_ma_rech90     2.255631
medianmarechprebal90     2.893772
cnt_loans30              2.033932
amnt_loans30             2.072804
cnt_loans90              2.372262
amnt_loans90             2.288744
payback30                3.037782
payback90                3.061069
dtype: float64
23
```

```
aon                     -0.053316
daily_decr30            -0.082770
daily_decr90            -0.075683
rental30                -0.119164
rental90                -0.116561
last_rech_date_ma        0.068162
last_rech_amt_ma        -0.401904
cnt_ma_rech30            0.052610
fr_ma_rech30             0.410220
sumamnt_ma_rech30       -0.407205
medianamnt_ma_rech30    -0.450275
medianmarechprebal30    -0.039734
cnt_ma_rech90            0.045461
fr_ma_rech90             0.341779
sumamnt_ma_rech90       -0.363397
medianamnt_ma_rech90    -0.411610
medianmarechprebal90    -0.065441
cnt_loans30              0.489167
amnt_loans30             0.403232
cnt_loans90              0.442523
amnt_loans90             0.360909
payback30                0.724178
payback90                0.582462
dtype: float64
2
```

Model Building

As we know, this is a classification problem we need to build a model using classification algorithm models.

First, we need to write a function which can find us best random state for train test split.

Then we shall iterate through all the models supporting classification algorithms to find the best models.

From above we get to know that the top 3 models are:

- ExtratreesRegressor
- LGMBRegressor
- XGBRegressor

Fine tune all these models and find their best parameters to use.

Next, find the best random state for train test split.

we obtain test accuracy of more than 98%.

CV score of this model is more than 99%.

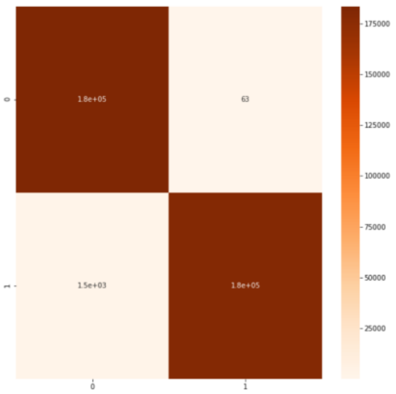
To analyze our model, we shall find the difference between actual and predicted value.

```
cross_val_score: 0.9977680339136483
roc 0.9956521425763813
diff 0.0021158913372669731
Confusion matrix
 [[183368     63]
 [  1515 179873]]
f1 score is :  0.9956327285206629
classification report
               precision    recall  f1-score   support

           0       0.99      1.00      1.00    183431
           1       1.00      0.99      1.00    181388

    accuracy                           1.00    364819
   macro avg       1.00      1.00      1.00    364819
weighted avg       1.00      1.00      1.00    364819

std: 0.00012514858015236357
```

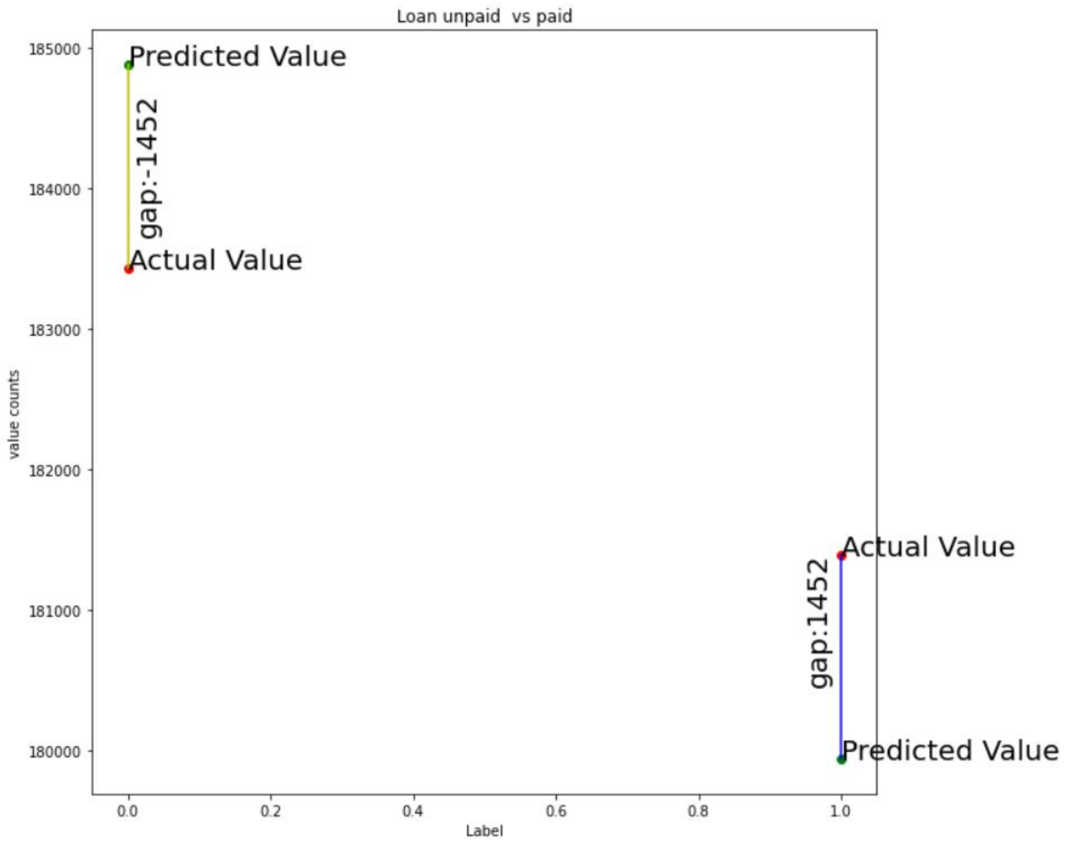
above metrics indicate that our model is performing at a very high accuracy

Finally our model has accuracy ranging from: 99% to 98%

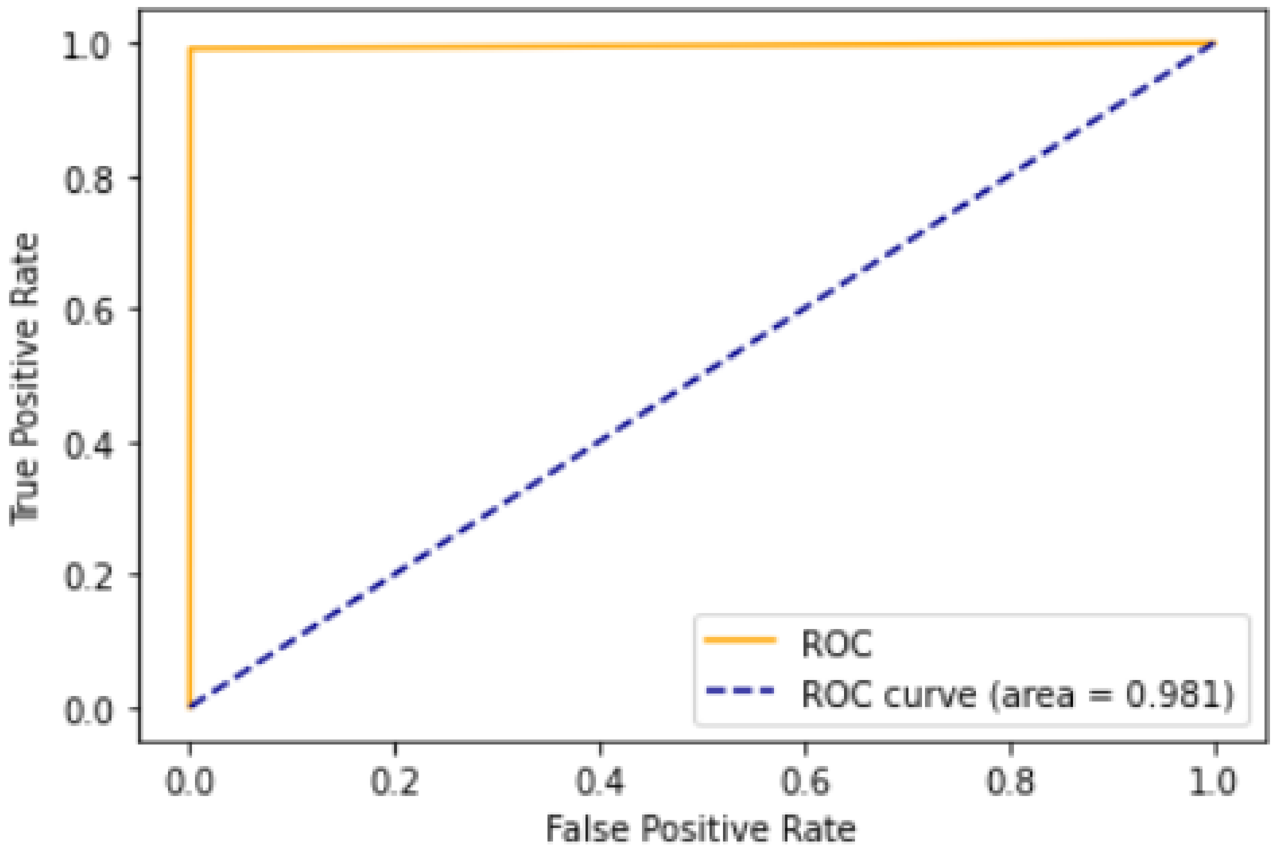
confusion matrix



Plot of Actual vs Predicted value



AUC ROC Curve



Interpretation of the Results

The dataset was very challenging to handle it had 37 features with 30days and 90days information of customers.

Firstly, the datasets were not having any null values.

But there was huge number of zero entries in maximum columns, so we must be careful while going through the statistical analysis of the datasets.

And proper plotting for proper type of features will help us to get better insight on the data. I found maximum numerical columns in the dataset, so I have chosen bar plot to see the relation between target and features.

I notice a huge amount of outliers and skewness in the data so we have choose proper methods to deal with the outliers and skewness. If we ignore these outliers and skewness, we may end up with a bad model which has less accuracy.

Then scaling dataset has a good impact like it will help the model not to get biased. Since we have not removed outliers and skewness completely from the dataset, so we must choose Normalization.

We must use multiple models while building model using dataset as to get the best model out of it.

And we must use multiple metrics like F1_score, precision, recall and accuracy score which will help us to decide the best model.

I found ExtraTrees Classifier as the best model with 99.9% accuracy score. Also, I have improved the accuracy of the best model by running hyper parameter tunning.

At last, I have predicted whether the loan is paid back or not using saved model. It was good!! that I was able to get the predictions near to actual values.

Conclusion

Key Features and conclusion of the study

In this project we have tried to predict whether the customer will pay the loan or not. The best accuracy score was achieved by fine-tuned ExtraTrees Classifier model.

LEARNING OUTCOMES OF THE STUDY IN RESPECT OF DATA SCIENCE

Through different powerful tools of visualization, we were able to analyze and interpret different hidden insights about the data.

Through data cleaning we were able to remove unnecessary columns and outliers from our dataset due to which our model would have suffered from overfitting or underfitting.

The data was improper scaled, so we scaled it to a single scale using sklearns's package StandardScaler.

The columns were skewed due to presence of outliers which we handled through percentile technique.

Model was then built having accuracy more than 90% using train dataset.

LIMITATIONS OF THIS WORK AND SCOPE FOR FUTURE WORK

Due to the presence of lot of outliers, we are unsure whether the model is going to perform well to a completely new dataset.

Due to a class imbalance, we had to rebalance the class 0. This might also have some effect while trying to predict the outcome with completely new data.

During data-collection, we could place limits on few continuous variables, where the customer could enter data within a limit because the variables like age on the network cannot be more than certain months