FLIP ROBO

Micro Credit Defaulter Project

Submitted by:

Vikash Kumar Singh

ACKNOWLEDGMENT

In this project, I took references from our DataTrained's video of Shankargouda Tegginmani sir and some websites which are- https://www.youtube.com https://www.kaggle.com , https://www.github.com , https://stackoverflow.com and https://scikit-learn.org in completion of this project. The data source is provided by Flip Robo company.

INTRODUCTION

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

- They understand the importance of communication and how it affects a person’s life, thus, focusing on providing their services and products to low-income families and poor customers that can help them in the need of hour.
- They are collaborating with an MFI to provide micro-credit on mobile balances to be paid back in 5 days. The Consumer is believed to be defaulter if he deviates from the path of paying back the loaned amount within the time duration of 5 days. For the loan amount of 5 (in Indonesian Rupiah), payback amount should be 6 (in Indonesian Rupiah), while, for the loan amount of 10 (in Indonesian Rupiah), the payback amount should be 12 (in Indonesian Rupiah).
- The sample data is provided to us from our client database. It is hereby given to you for this exercise. In order to improve the selection of customers for the credit, the client wants some predictions that could help them in further investment and improvement in selection of customers.

- Conceptual Background of the Domain Problem

1. In preparation of any Machine Learning Model, we have to have conceptual knowledge of cleansing, scaling, outlier and many more method to perform the EDA analysis
2. The knowledge of Data Visualization is also required to observe the skewness, outlier, correlation and multicollinearity of the given dataset and you are also required to have knowledge to reduce the following observation.

3. Use standard techniques (PCA, Scaling, encoding, underfitting/overfitting, AUC-ROC curve, cross-validation, grid search, ensemble techniques) wherever applicable.

- ## Review of Literature

  The literature study gives an overview of this project and the EDA methods, the feature engineering methods that have been used in this study. As well as about evaluation metrics to measure the performance of the algorithms.

- ## Motivation for the Problem Undertaken

  Presently, I am doing an internship with FlipRobo company and they have given us this project to build a Machine Learning Model to predict whether the customer is a defaulter or not i.e., he is paying his loaned amount on time or not. This project gives me an opportunity to recall the teaching of Datatrained's mentor and to build multiple classification model with better accuracy. This will help me in upgrading my skills and knowledge.

Analytical Problem Framing

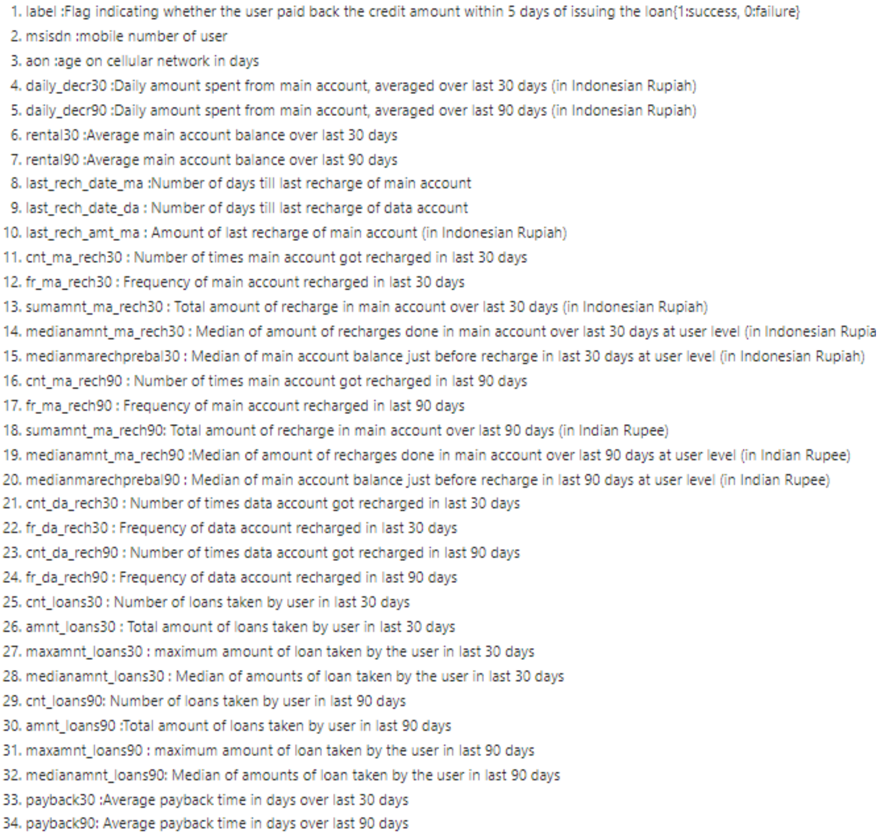
- ## Mathematical/ Analytical Modeling of the Problem

  In the given dataset, our target is in categorical form. 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.

  so, the model will be classification model (supervised learning) in which, we will design a predictive model to analyze the relation between features and target variables (Label ‘1’ and Label ‘0’).

- ## Data Sources and their formats

  Flip Robo has provided us the required data set for the model building in which it contains ‘209593’ rows and ‘37’ columns, out of which ‘3’ columns are of object datatype and rest are is of numerical datatypes.



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 Rupia
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 Indian Rupee)
19. medianamnt_ma_rech90 :Median of amount of recharges done in main account over last 90 days at user level (in Indian Rupee)
20. medianmarechprebal90 : Median of main account balance just before recharge in last 90 days at user level (in Indian Rupee)
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

- ## Data Pre-processing Done

In pre-processing, we have done the following steps:

1.) Imported some important libraries in our jupyter notebook.
2.) Imported the given dataset
3.) Check the details of each feature from the given dataset.
4.) Cleanse the dataset i.e., check for null values, duplicate value, shape of dataset etc.
5.) Drop the unnecessary features.
6.) Checked the skewness
7.) Transform the dataset with square root transformation
8.) Outlier detection
9.) Checked for correlation
10.) Visualization
11.) Separation of features and label
12.) Scaling of the data
13.) Train test split.

- ## Data Inputs- Logic- Output Relationships

As we know that any loan will get approved by bank or finance institution is depend upon many things and the most important among all is the income of the customer. To get customer's past, they check for the credit score of the customer. In credit score is determined by whether the customer is paying his loaned amount on time or not, how much loan, he has already taken, how many loan he has taken etc. These same is also applying in this model. We will thoroughly analyse each feature to get logic behind the label and features i.e., input- output relationships.

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

My only presumption was that the finance institution will give only on the basis of customer past record of other loan and income of the customer.

Hardware and Software Requirements and Tools Used

- **Hardware tools:**

1. Windows laptop
2. i5 processor
3. 4GB ram
4. 250 GB SSD card

- **Software tools:**

1. windows 10
2. Anaconda Navigator
3. Jupyter Notebook
4. Python

- **Libraries and packages:**

1. Pandas
2. NumPy
3. SciPy
4. Seaborn
5. Mat plot
6. Sklearn

Model/s Development and Evaluation

- **Identification of possible problem-solving approaches (methods)**

  From the domain knowledge and my own understanding, I have followed these steps:

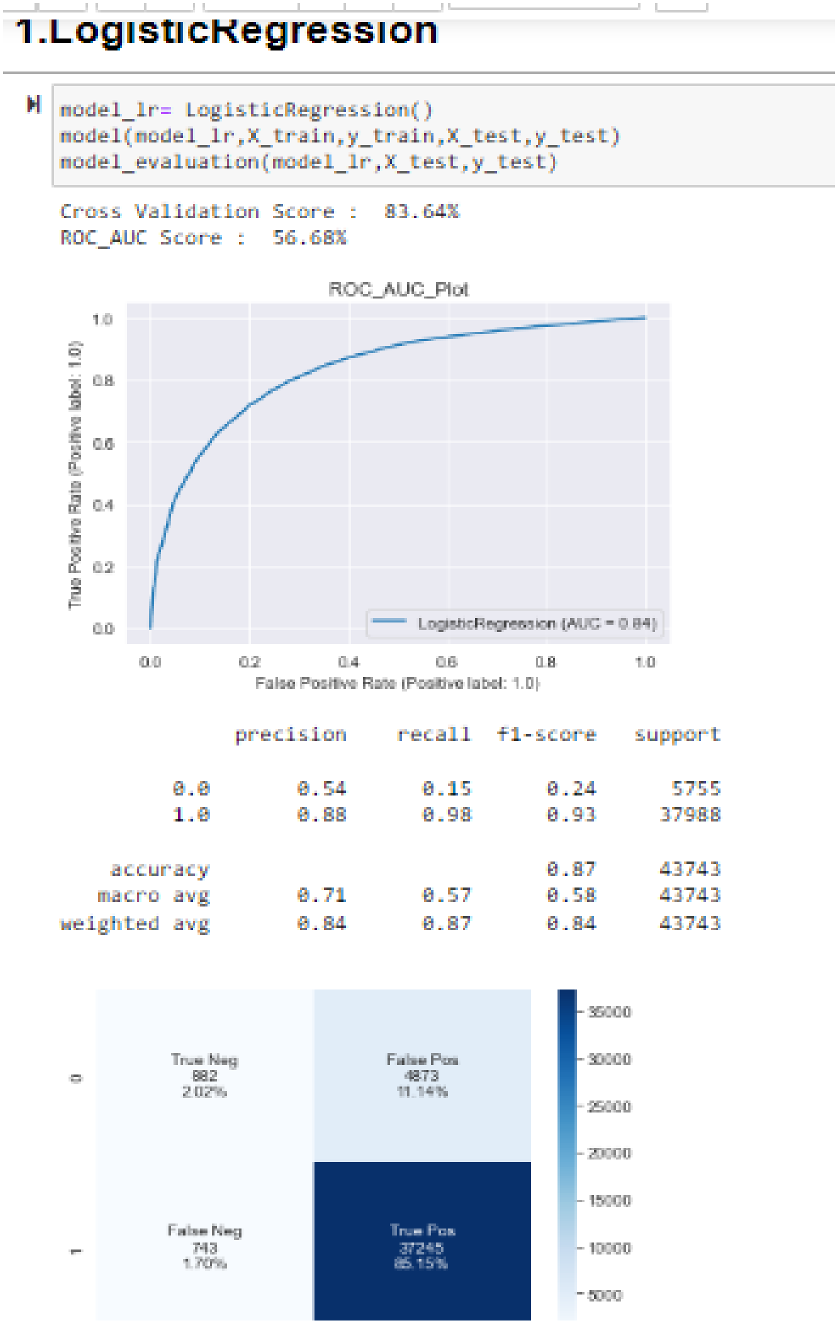
  1. Importing the given test and train dataset
  2. Checking null values in the dataset
  3. Imputation of given dataset
  4. Transformation of given dataset
  5. Standardization of given dataset
  6. Splitting the train and test dataset

- **Testing of Identified Approaches (Algorithms)**

  1. Logistic Regression
  2. Random Forest classifier
  3. Decision Tree classifier
  4. Support Vector classifier
  5. KNeighbors classifier
  6. Extra Tree classifier
  7. GaussianNB
  8. Dummy classifier
  9. AdaBoost classifier
  10. Gradient Boosting classifier

- **Run and evaluate selected models**

1. Logistic regression



1.LogisticRegression

```
model_lr= LogisticRegression()
model(model_lr,X_train,y_train,X_test,y_test)
model_evaluation(model_lr,X_test,y_test)
```

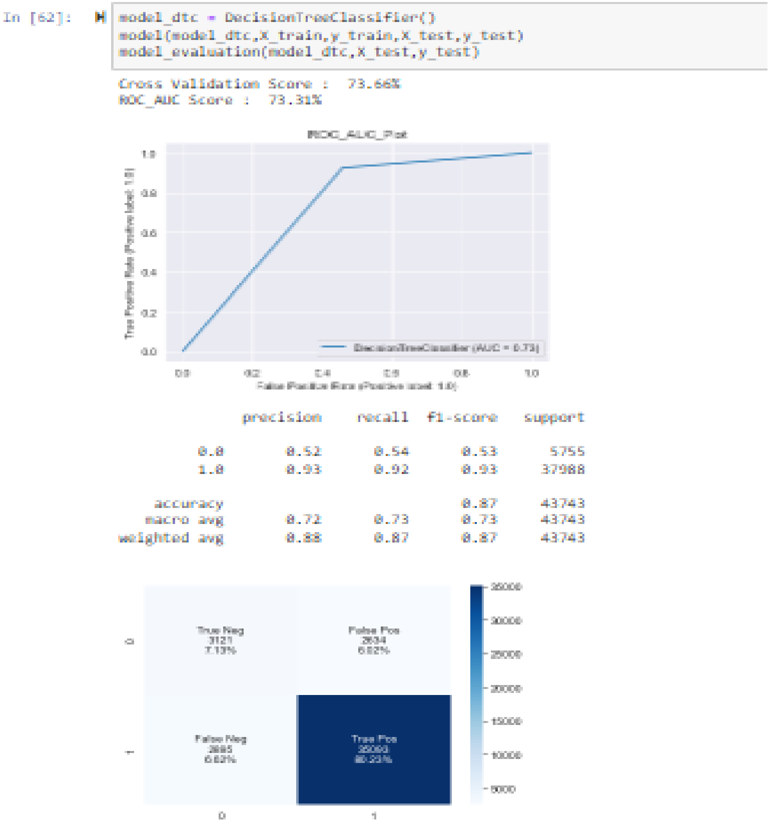
```
Cross Validation Score :  83.64%
ROC_AUC Score :  56.68%
```

ROC_AUC_Plot
True Positive Rate (Positive label: 1.0)
1.0
0.8
0.6
0.4
0.2
0.0
LogisticRegression (AUC = 0.84)
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate (Positive label: 1.0)

```
              precision    recall  f1-score   support

         0.0       0.54      0.15      0.24      5755
         1.0       0.88      0.98      0.93     37988

    accuracy                           0.87     43743
   macro avg       0.71      0.57      0.58     43743
weighted avg       0.84      0.87      0.84     43743
```

It has an accuracy of 87%, cross validation score of 83.64% and Roc auc score of 56.68%. It has also mentioned the score of precision, recall and f1.

2. Decision tree classifier



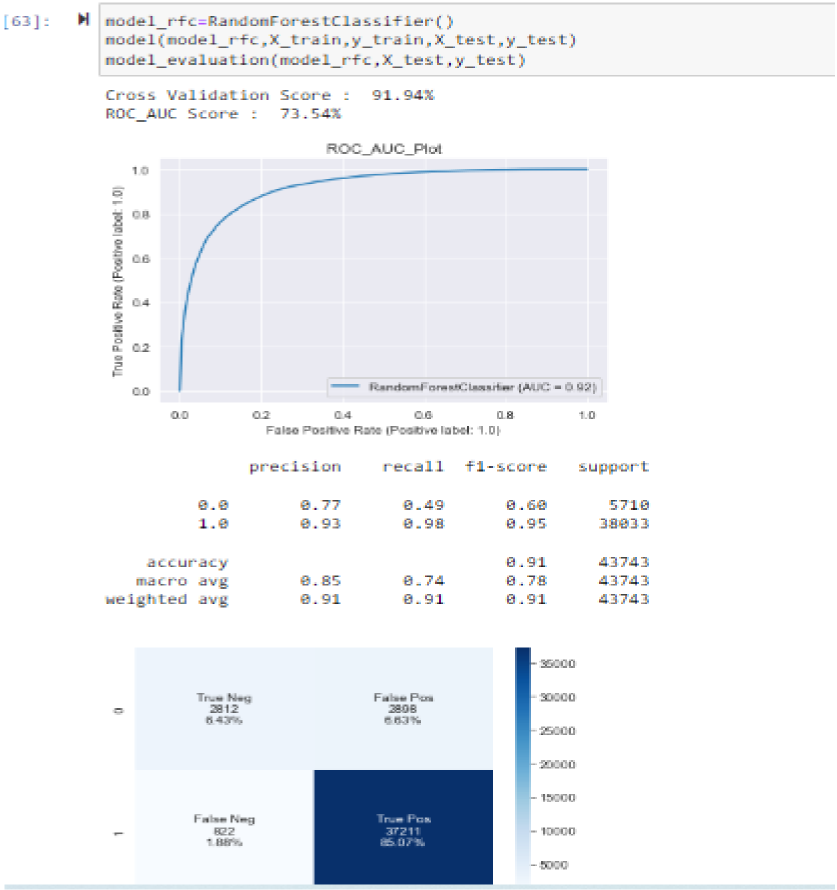
```
In [62]:  model_dtc = DecisionTreeClassifier()
          model(model_dtc,X_train,y_train,X_test,y_test)
          model_evaluation(model_dtc,X_test,y_test)
```

```
Cross Validation Score :  73.66%
ROC_AUC Score :  73.31%
```

ROC_AUC_Plot
1.0
0.8
0.6
0.4
0.2
0.0
True Positive Rate (Positive label: 1.0)
DecisionTreeClassifier (AUC = 0.73)
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate (Positive label: 1.0)

```
              precision    recall  f1-score   support

         0.0       0.52      0.54      0.53      5755
         1.0       0.93      0.92      0.93     37988

    accuracy                           0.87     43743
   macro avg       0.72      0.73      0.73     43743
weighted avg       0.88      0.87      0.87     43743
```

3. Random forest classifier



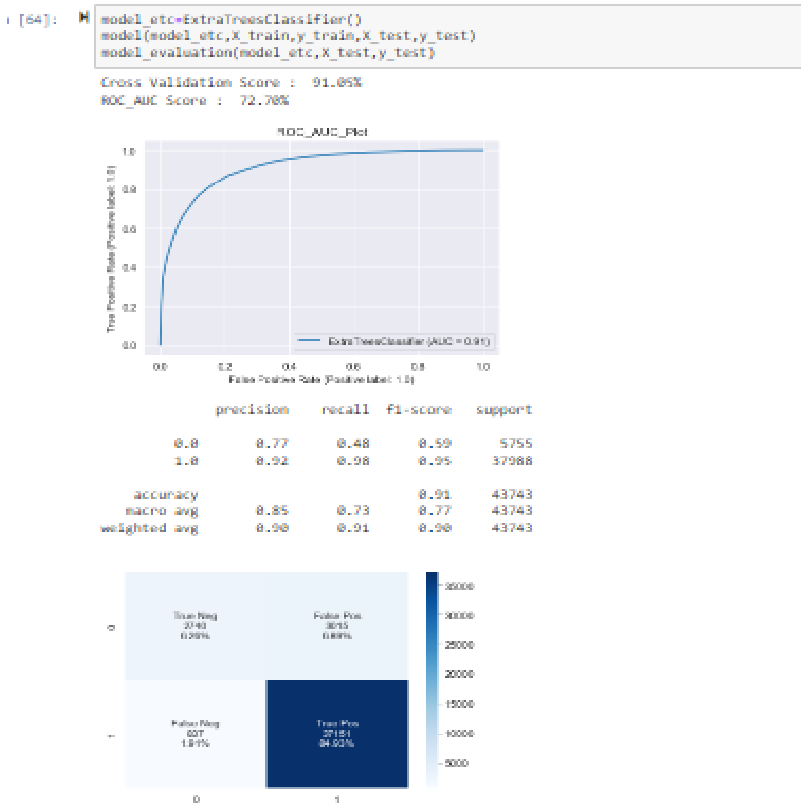
```
[63]:  model_rfc=RandomForestClassifier()
       model(model_rfc,X_train,y_train,X_test,y_test)
       model_evaluation(model_rfc,X_test,y_test)
```

```
Cross Validation Score :  91.94%
ROC_AUC Score :  73.54%
```

ROC_AUC_Plot
1.0
0.8
0.6
0.4
0.2
0.0
True Positive Rate (Positive label: 1.0)
RandomForestClassifier (AUC = 0.92)
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate (Positive label: 1.0)

```
              precision    recall  f1-score   support

         0.0       0.77      0.49      0.60      5710
         1.0       0.93      0.98      0.95     38033

    accuracy                           0.91     43743
   macro avg       0.85      0.74      0.78     43743
weighted avg       0.91      0.91      0.91     43743
```

4. Extra tree classifier



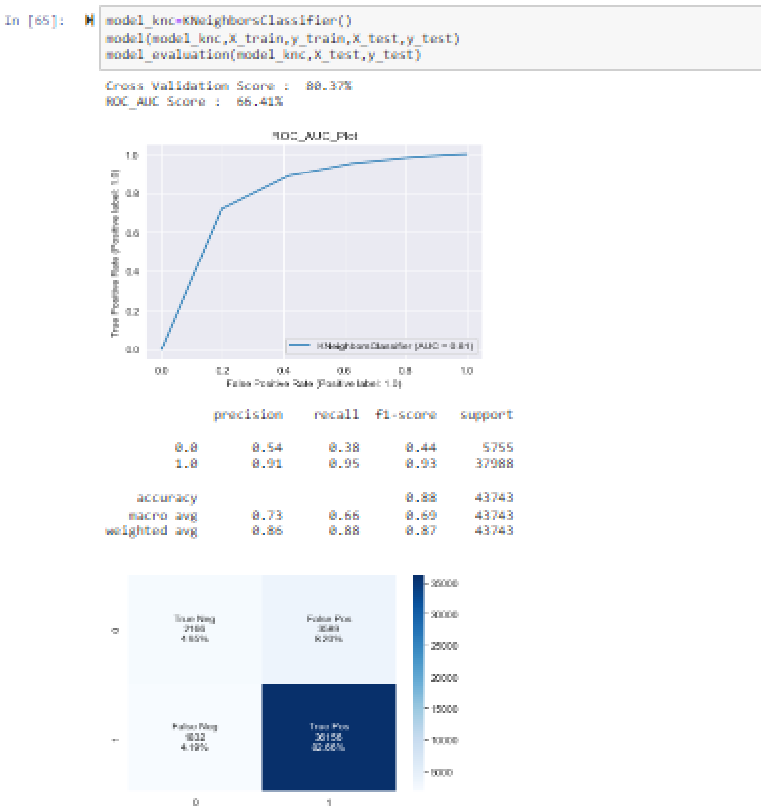
```
In [64]:  model_etc=ExtraTreesClassifier()
          model(model_etc,X_train,y_train,X_test,y_test)
          model_evaluation(model_etc,X_test,y_test)
```

```
Cross Validation Score :  91.05%
ROC_AUC Score :  72.70%
```

ROC_AUC_Plot
True Positive Rate (Positive label: 1.0)
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate (Positive label: 1.0)
ExtraTreesClassifier (AUC = 0.91)

```
              precision    recall  f1-score   support

         0.0       0.77      0.48      0.59      5755
         1.0       0.92      0.98      0.95     37988

    accuracy                           0.91     43743
   macro avg       0.85      0.73      0.77     43743
weighted avg       0.90      0.91      0.90     43743
```

5. KNeighbors classifier



```
In [65]:  model_knc=KNeighborsClassifier()
          model(model_knc,X_train,y_train,X_test,y_test)
          model_evaluation(model_knc,X_test,y_test)
```

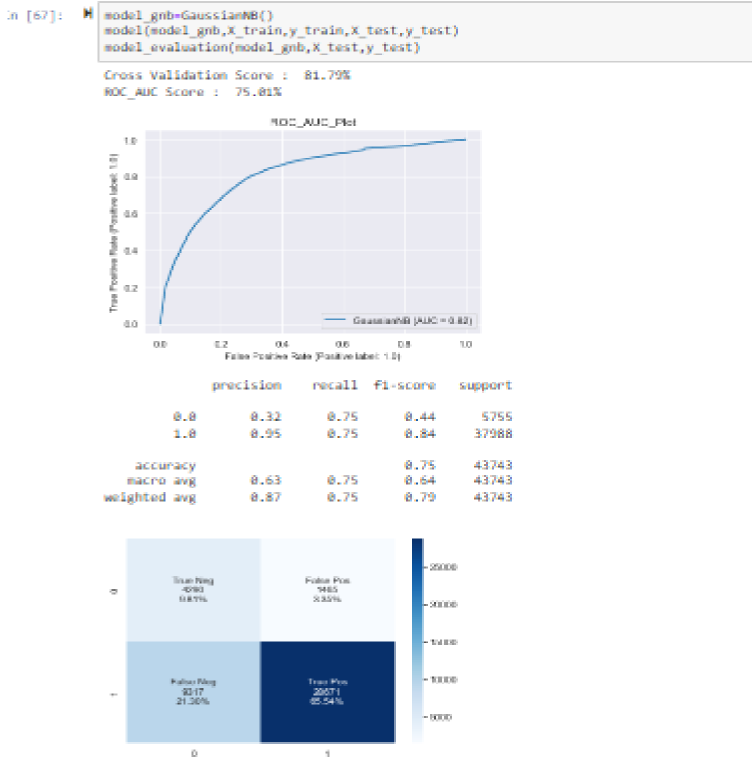
```
Cross Validation Score :  80.37%
ROC_AUC Score :  66.41%
```

ROC_AUC_Plot
True Positive Rate (Positive label: 1.0)
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate (Positive label: 1.0)
KNeighborsClassifier (AUC = 0.81)

```
              precision    recall  f1-score   support

         0.0       0.54      0.38      0.44      5755
         1.0       0.91      0.95      0.93     37988

    accuracy                           0.88     43743
   macro avg       0.73      0.66      0.69     43743
weighted avg       0.86      0.88      0.87     43743
```

6. SVC

7. GaussianNB



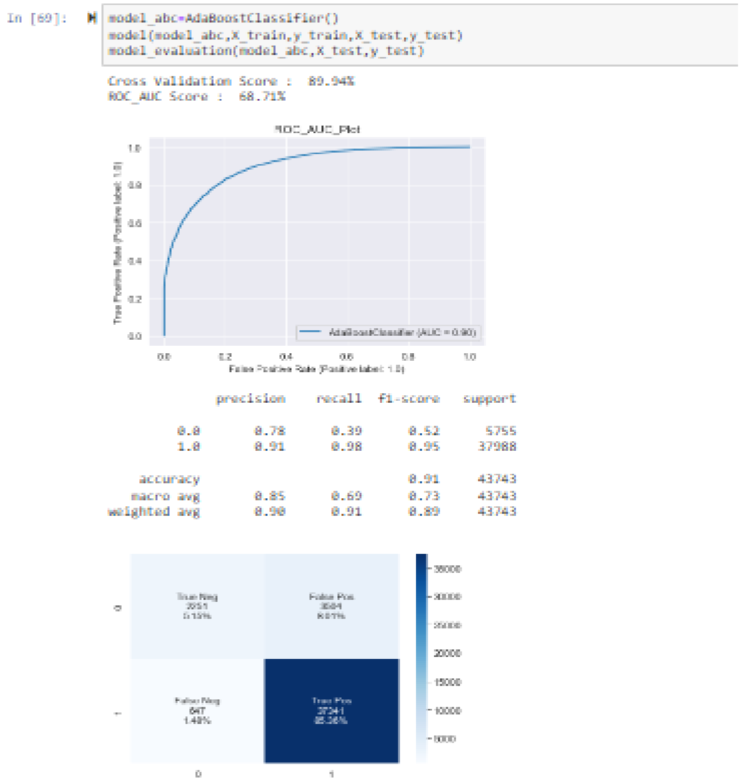
```
In [67]:  model_gnb=GaussianNB()
          model(model_gnb,X_train,y_train,X_test,y_test)
          model_evaluation(model_gnb,X_test,y_test)
```

```
Cross Validation Score :  81.79%
ROC_AUC Score :  75.01%
```

ROC_AUC_Plot
True Positive Rate (Positive label: 1.0)
1.0
0.8
0.6
0.4
0.2
0.0
GaussianNB (AUC = 0.82)
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate (Positive label: 1.0)

```
              precision    recall  f1-score   support

         0.0       0.32      0.75      0.44      5755
         1.0       0.95      0.75      0.84     37988

    accuracy                           0.75     43743
   macro avg       0.63      0.75      0.64     43743
weighted avg       0.87      0.75      0.79     43743
```

8. AdaBoosting classifier



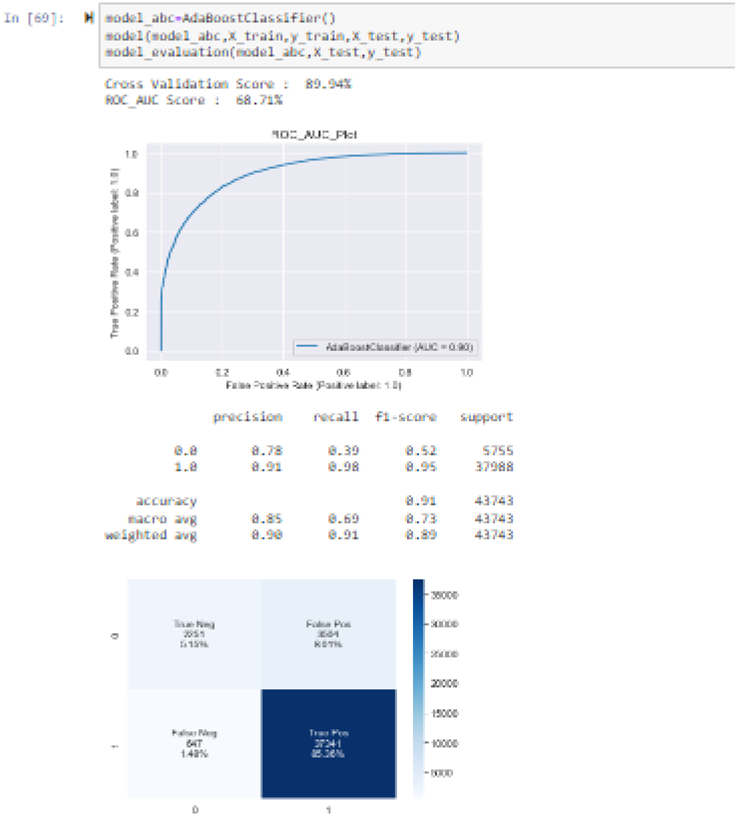
```
In [69]:  model_abc=AdaBoostClassifier()
          model(model_abc,X_train,y_train,X_test,y_test)
          model_evaluation(model_abc,X_test,y_test)
```

```
Cross Validation Score :  89.94%
ROC_AUC Score :  68.71%
```

ROC_AUC_Plot
True Positive Rate (Positive label: 1.0)
1.0
0.8
0.6
0.4
0.2
0.0
AdaBoostClassifier (AUC = 0.90)
0.0
0.2
0.4
0.6
0.8
1.0
False Positive Rate (Positive label: 1.0)

```
              precision    recall  f1-score   support

         0.0       0.78      0.39      0.52      5755
         1.0       0.91      0.98      0.95     37988

    accuracy                           0.91     43743
   macro avg       0.85      0.69      0.73     43743
weighted avg       0.90      0.91      0.89     43743
```

9. GradientBoosting classifier



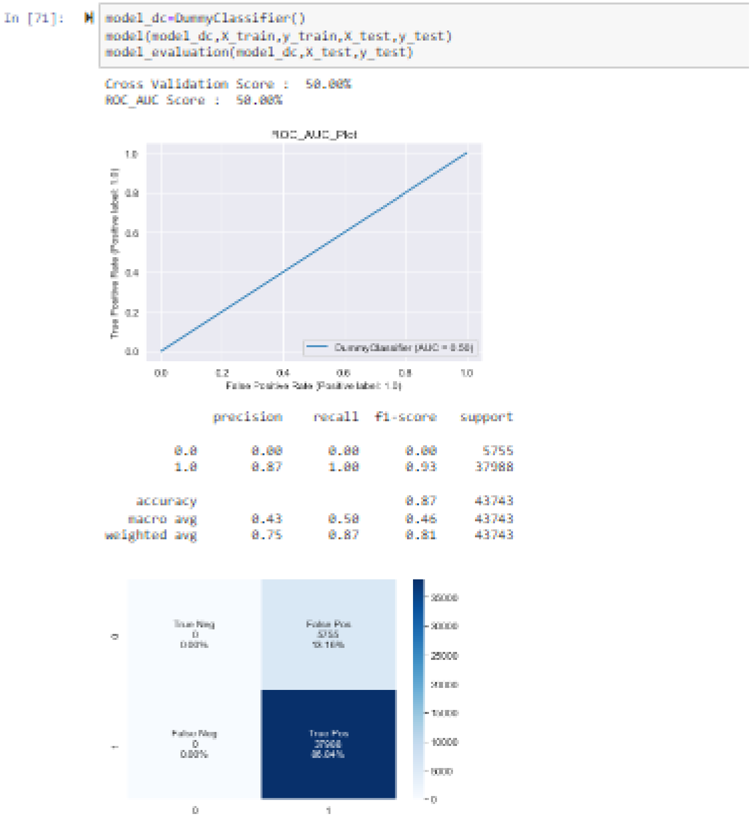
```
In [69]:  model_abc=AdaBoostClassifier()
          model(model_abc,X_train,y_train,X_test,y_test)
          model_evaluation(model_abc,X_test,y_test)
```

```
Cross Validation Score :  89.94%
ROC_AUC Score :  68.71%
```

ROC_AUC_Plot
True Positive Rate (Positive label: 1.0)
False Positive Rate (Positive label: 1.0)
0.0
0.2
0.4
0.6
0.8
1.0
AdaBoostClassifier (AUC = 0.90)

```
              precision    recall  f1-score   support

         0.0       0.78      0.39      0.52      5755
         1.0       0.91      0.98      0.95     37988

    accuracy                           0.91     43743
   macro avg       0.85      0.69      0.73     43743
weighted avg       0.90      0.91      0.89     43743
```

10. Dummy classifier



```
In [71]:  model_dc=DummyClassifier()
          model(model_dc,X_train,y_train,X_test,y_test)
          model_evaluation(model_dc,X_test,y_test)
```

```
Cross Validation Score :  50.00%
ROC_AUC Score :  50.00%
```

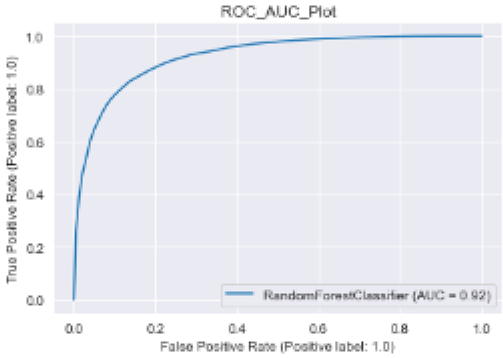
ROC_AUC_Plot
True Positive Rate (Positive label: 1.0)
False Positive Rate (Positive label: 1.0)
0.0
0.2
0.4
0.6
0.8
1.0
DummyClassifier (AUC = 0.50)

```
              precision    recall  f1-score   support

         0.0       0.00      0.00      0.00      5755
         1.0       0.87      1.00      0.93     37988

    accuracy                           0.87     43743
   macro avg       0.43      0.50      0.46     43743
weighted avg       0.75      0.87      0.81     43743
```

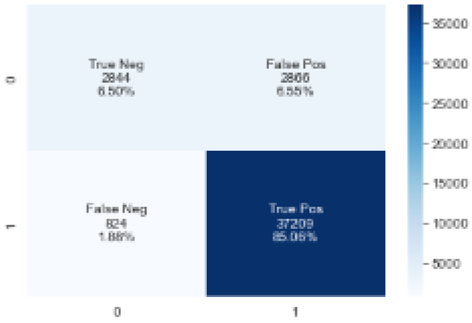
We have built different models of classification, in which we are evaluating them with cross validation score, rocauc score and classification report which includes precision, f1, recall score.Among all, Random forest classification model has highest accuracy and lowest difference between its accuracy and cross validation score. For further improvement in the accuracy, we have done hyperparameter tuning for this this model and it improved the accuracy.

After hyperparameter tuning.

```
Cross Validation Score :  92.29%
ROC_AUC Score :  73.82%
```



```
              precision    recall  f1-score   support

         0.0       0.78      0.50      0.61      5710
         1.0       0.93      0.98      0.95     38033

    accuracy                           0.92     43743
   macro avg       0.85      0.74      0.78     43743
weighted avg       0.91      0.92      0.91     43743
```



- ## Key Metrics for success in solving problem under consideration

a. Cross validation score

b. Roc auc score

c. Precision score

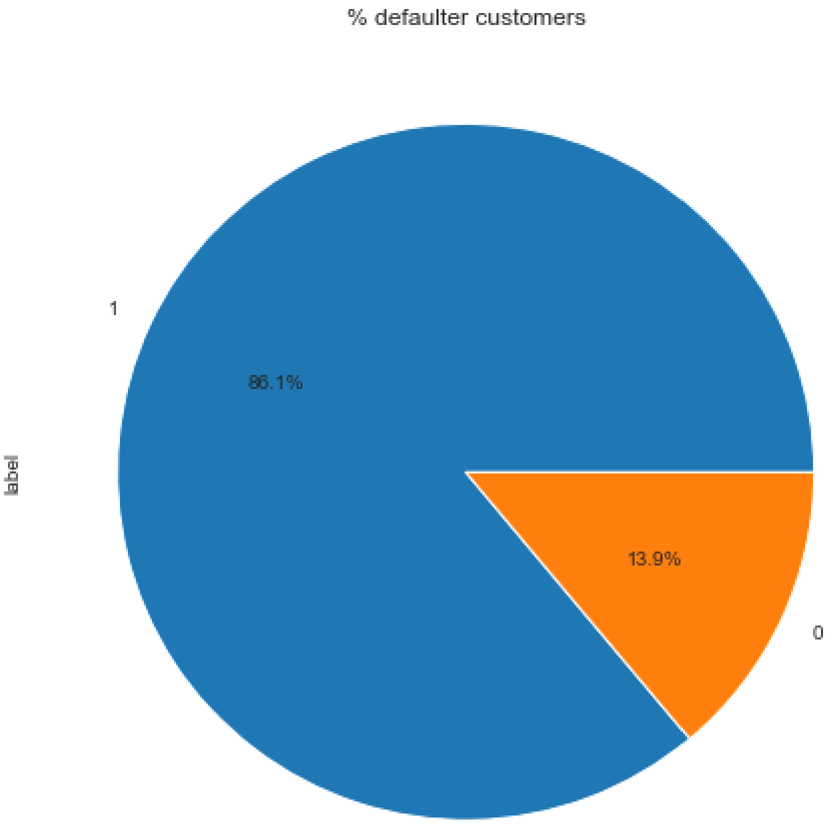
d. F1 score

e. Recall score

f. Accuracy

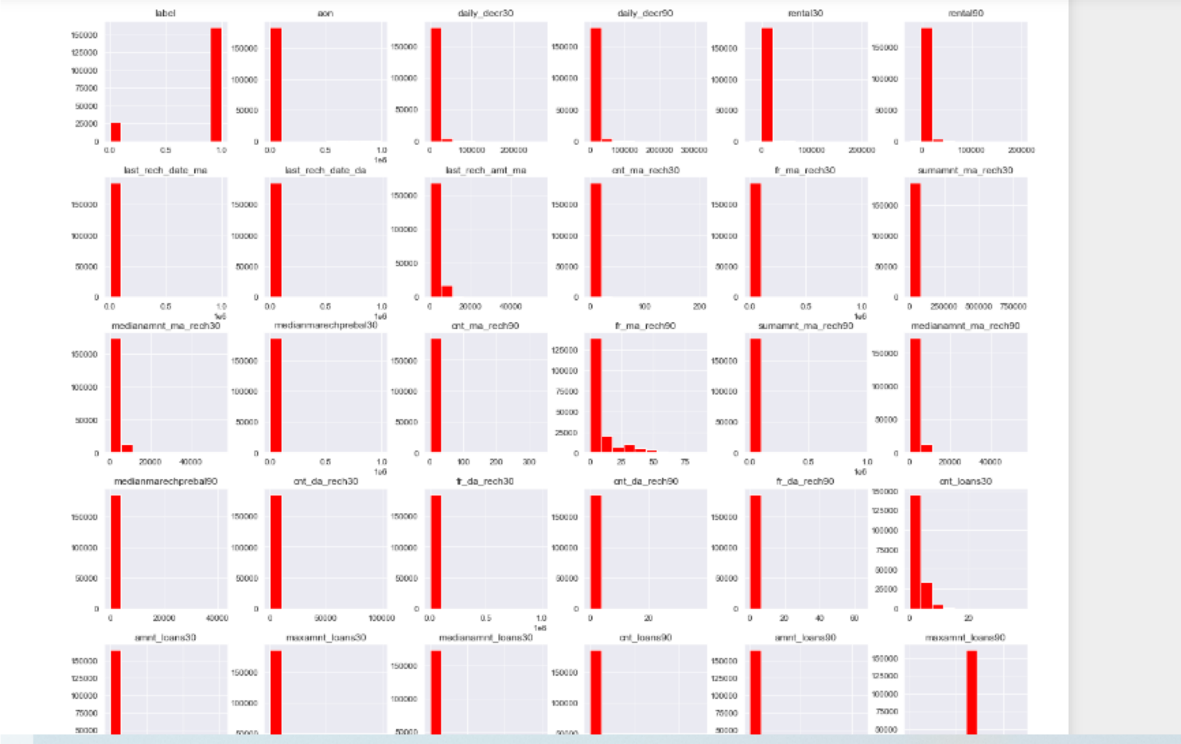
- **Visualizations**

1. Pie plot of 'label' column.

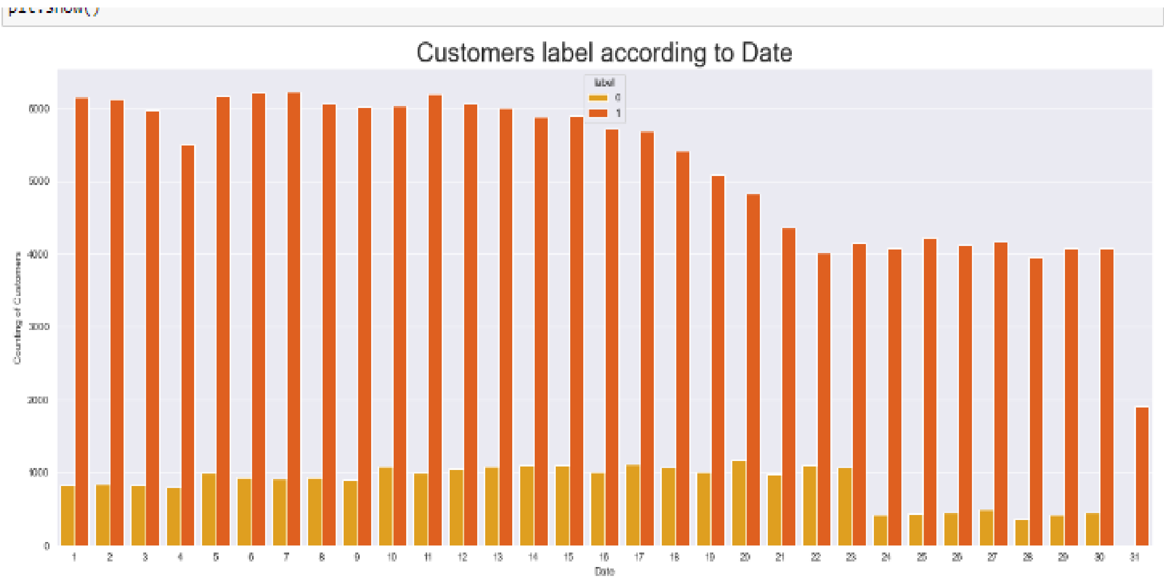


After seeing the label column which is also our target feature for this dataset it is clearly shown that 86.1% of data is label 1 and only 13.9% of data is label 0 so our dataset is imbalanced. So before making the ML model first we have to do sampling to get rid of imbalance dataset

2. Histplot

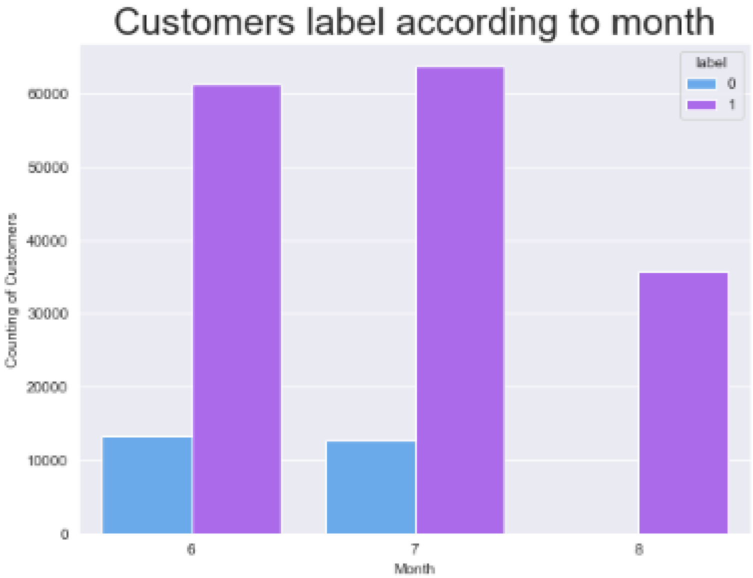


3. Customer label in respet to date

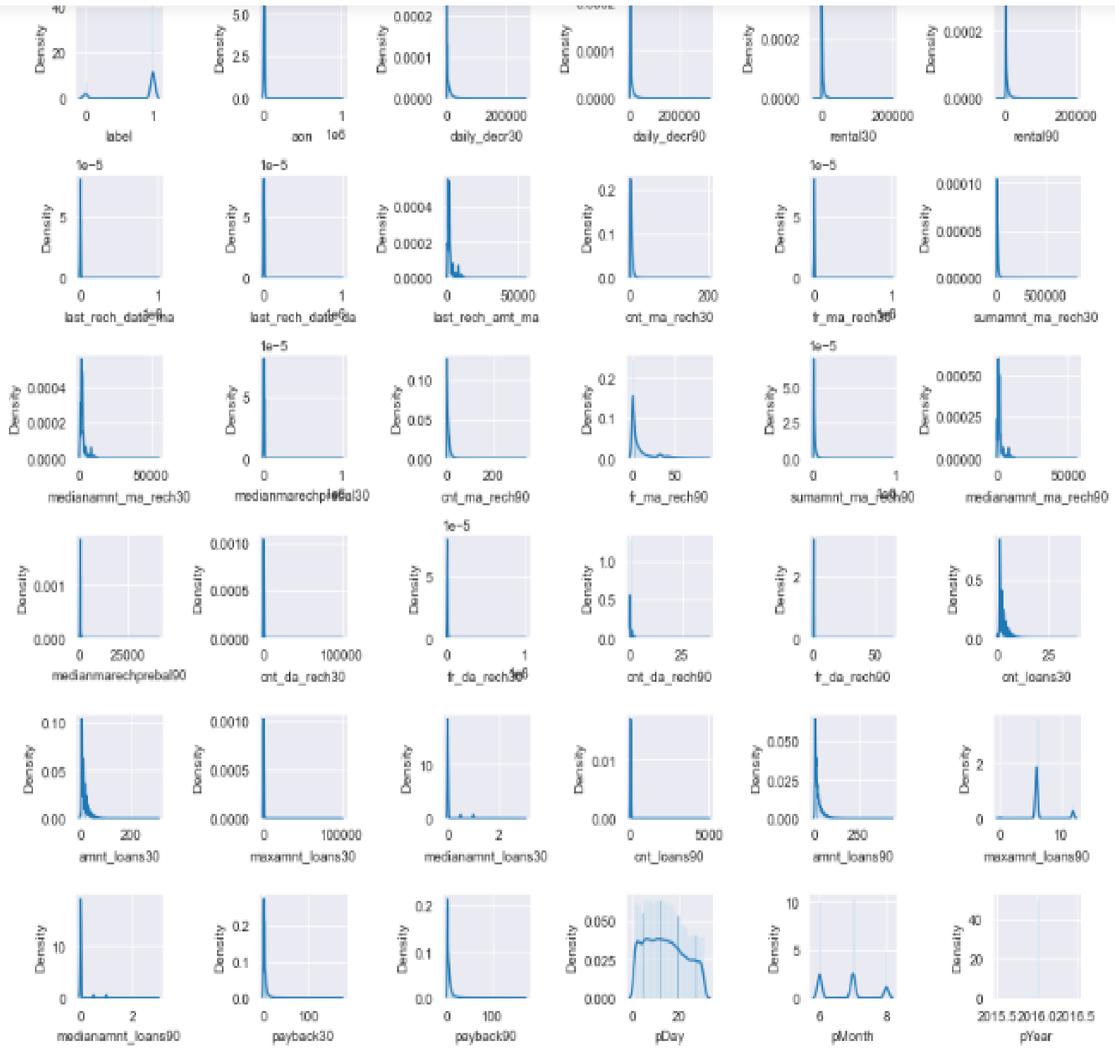


```
plt.show()
```

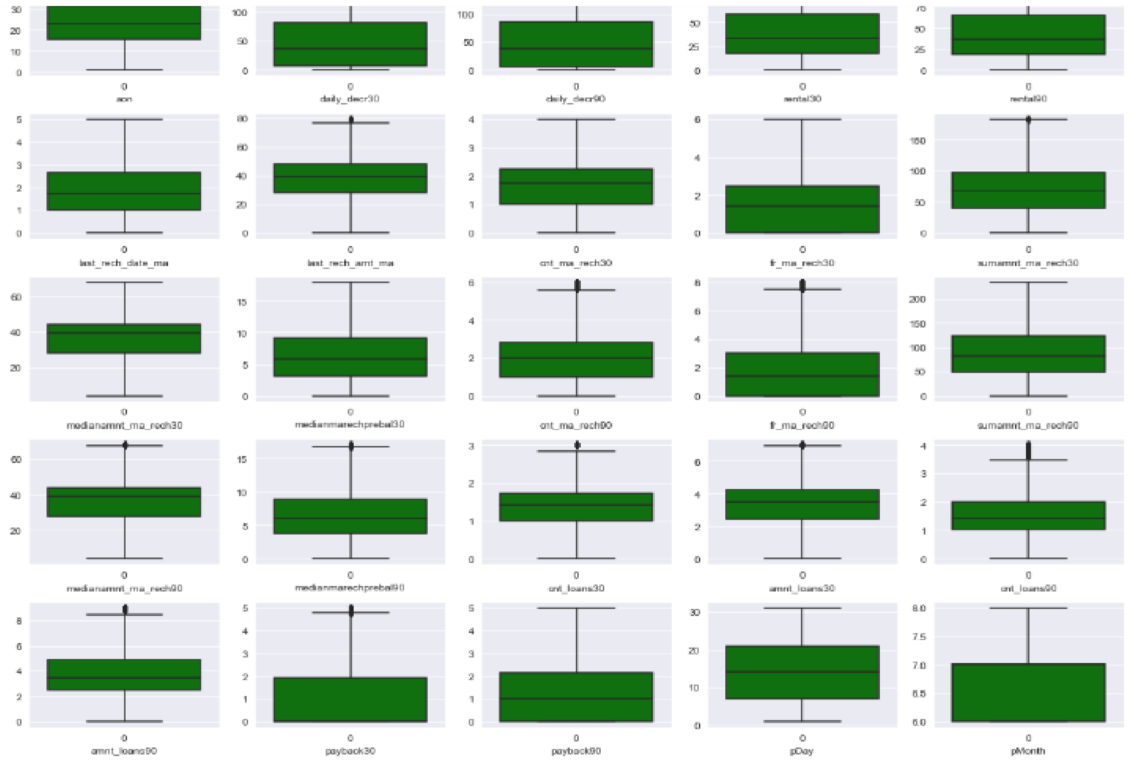
4. Customer label in according to month



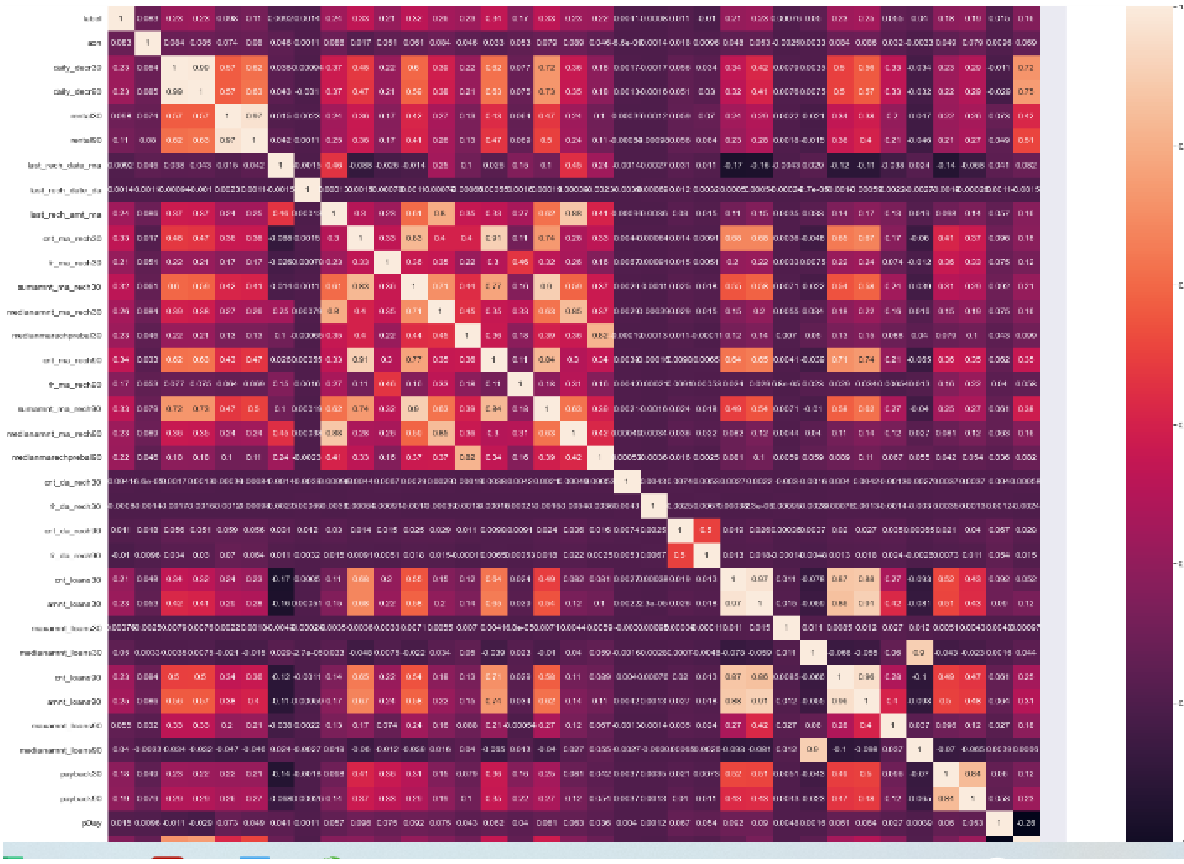
5. Scatterplot of features



6. Box plot to detect outlier



7. Heatmap



- ### Interpretation of the Results

  After Visualization, we have observed the skewness, outlier and multicollinearity of the data, and have tried to reduce it by square root transformation technique and interquartile range and respectively.

  After doing 10 different modelling, we analyzed that the random forest classifier model is best among all.

CONCLUSION

- ### Key Findings and Conclusions of the Study

  The aim of this project was to determine whether the customer will be paying back the loaned amount. In this project 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. In this project, to improve the selection of customers for the credit, the client needs some predictions that could help them in further investment and improvement in selection of customers. we have discovered many algorithms and application of machine learning techniques with the objective to predict the customer is a defaulter or not. We have first cleaning and exploring of the input data. We have performed Logistic regression, Random Forest classifier, Decision Tree classifier, SVC, KNeighbours classifier, Dummy classifier, GaussianNB, Adaboost classifier, XGB classifier and Gradient boosting classifier and hyperparameter tuning as we understand that parameterization can drive the significant result in the performance.

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

i. From different visualization, we have learned about the skewness, outlier and multicollinearity of the dataset and it helped in making better model.

ii. Data cleansing- we learned, how to fill or drop the null values data for the dataset by performing multiple statistical operation.

iii. Machine learning algorithm- we have performed 10 multiple regression model to predict the sale price of house, out of all Gradient boosting regressor have performed best among the others. We have also the Hyper Para meter tuning to the improve of model accuracy.

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

Limitations of this project were that we haven't cover all classification algorithms in our Data Science course, instead of it, they have focused on the basic algorithm. In Hyper Para Meter tuning, parameter for different model is very difficult to choose.