Submitted by-

KAUSHIK VEER

ACKNOWLEDGMENT

All thanks to flip robo technologies for providing me the opportunity to work on this project. I leaned a lot from this project.

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.

- Conceptual Background of the Domain Problem

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.

- Review of Literature

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).

In order to improve the selection of customers for the credit, the we have done some predictions that could help the client in further investment and improvement in selection of customers.

- Motivation for the Problem Undertaken
- In order to improve the selection of customers for the credit, the we have done some predictions that could help

the client in further investment and improvement in selection of customers.

We will 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 payed i.e. Non- defaulter, while, Label ‘0’ indicates that the loan has not been payed i.e. defaulter.

Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem

  There are no null values in the dataset.
  There may be some customers with no loan history.
  The dataset is imbalanced. Label '1' has approximately 87.5% records, while, label '0' has approximately 12.5% records.
  For some features, there may be values which might not be realistic. You may have to observe them and treat them with a suitable explanation.
  You might come across outliers in some features which you need to handle as per your understanding. Keep in mind that data is expensive and we cannot lose more than 7-8% of the data.
  Extensive EDA has been performed to gain relationships of important variable and label.
  Data contains numerical as well as categorical variable.
  We will build Machine Learning models and determine the optimal values of Hyper Parameters.

- Data Sources and their formats

  In the micro credit dataset there are 209593 rows and columns are 37

Following are the description of all the 37 variables :-

label	Flag indicating whether the user paid back the credit amount within 5 days of issuing the loan{1:success, 0:f
msisdn	mobile number of user
aon	age on cellular network in days
daily_decr30	Daily amount spent from main account, averaged over last 30 days (in Indonesian Rupiah)
daily_decr90	Daily amount spent from main account, averaged over last 90 days (in Indonesian Rupiah)
rental30	Average main account balance over last 30 days
rental90	Average main account balance over last 90 days
last_rech_date_ma	Number of days till last recharge of main account
last_rech_date_da	Number of days till last recharge of data account
last_rech_amt_ma	Amount of last recharge of main account (in Indonesian Rupiah)
cnt_ma_rech30	Number of times main account got recharged in last 30 days
fr_ma_rech30	Frequency of main account recharged in last 30 days
sumamnt_ma_rech30	Total amount of recharge in main account over last 30 days (in Indonesian Rupiah)
medianamnt_ma_rech3(	Median of amount of recharges done in main account over last 30 days at user level (in Indonesian Rupiah)
medianmarechprebal30	Median of main account balance just before recharge in last 30 days at user level (in Indonesian Rupiah)
cnt_ma_rech90	Number of times main account got recharged in last 90 days
fr_ma_rech90	Frequency of main account recharged in last 90 days
sumamnt_ma_rech90	Total amount of recharge in main account over last 90 days (in Indonasian Rupiah)
medianamnt_ma_rech9(	Median of amount of recharges done in main account over last 90 days at user level (in Indonasian Rupiah)
medianmarechprebal90	Median of main account balance just before recharge in last 90 days at user level (in Indonasian Rupiah)
cnt_da_rech30	Number of times data account got recharged in last 30 days
fr_da_rech30	Frequency of data account recharged in last 30 days
cnt_da_rech90	Number of times data account got recharged in last 90 days
fr_da_rech90	Frequency of data account recharged in last 90 days
cnt_loans30	Number of loans taken by user in last 30 days
amnt_loans30	Total amount of loans taken by user in last 30 days
maxamnt_loans30	maximum amount of loan taken by the user in last 30 days
medianamnt_loans30	Median of amounts of loan taken by the user in last 30 days
cnt_loans90	Number of loans taken by user in last 90 days
amnt_loans90	Total amount of loans taken by user in last 90 days
maxamnt_loans90	maximum amount of loan taken by the user in last 90 days
medianamnt_loans90	Median of amounts of loan taken by the user in last 90 days
payback30	Average payback time in days over last 30 days
payback90	Average payback time in days over last 90 days
pcircle	telecom circle
pdate	date

- Data Pre-processing Done

  misdn represents mobile numbers of users and that are unique all the time pcircle same output for all the columns so we can drop pcircle and msisdn as well as unnamed as it is of no use for model building

```
#dropping the columns which are of no use
df=df.drop(['pcircle','msisdn','Unnamed: 0'],axis=1)
```

- We converted pdate into numerical form by converting the data into day-month and year and dropping the column pdate as it was of no use then.

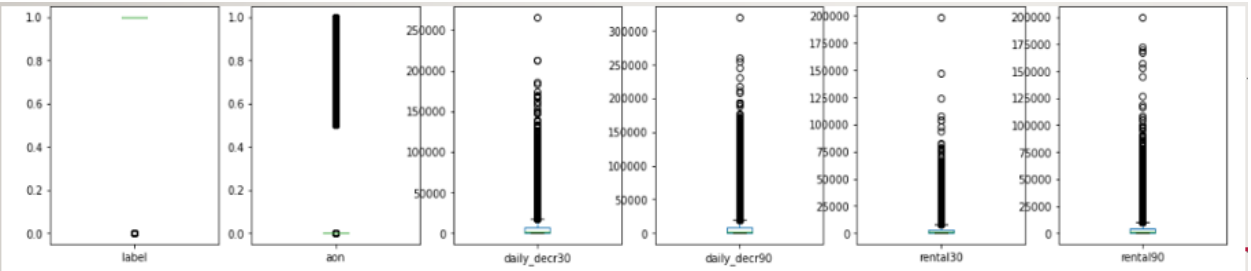
```
#convering the pdate from object type data to numerical and in date month and year format
df['pdate'] =  pd.to_datetime(df['pdate'])
df["pyear"]=pd.to_datetime(df.pdate, format="%d/%m/%Y").dt.year
df["pmonth"]=pd.to_datetime(df.pdate, format="%d/%m/%Y").dt.month
df["pday"]=pd.to_datetime(df.pdate, format="%d/%m/%Y").dt.day
```

- We checked value counts for all the variables and we found that pyear has only one data so we can drop the columns as we already have month and day to define the entry
- columns like fr_da_rech90, medianamnt_loans30, medianamnt_loans90,last_rech_date_da,cnt_da_rech30,fr_da_rech30,cnt_da_rech90,maxamnt_loans30, maxamnt_loans90 are showing biasing in their data as you can see in the value counts so we dropped them.

```
#dropping the above mentioned columns
df=df.drop(['last_rech_date_da','cnt_da_rech30','fr_da_rech30','cnt_da_rech90','fr_da_rech90','medianamnt_loans30',],axis=1)
df=df.drop(['medianamnt_loans90','pyear','maxamnt_loans90','maxamnt_loans30'],axis=1)
df
```

- We saw some negative values so we converted them into positive values using abs function.

We saw some outliers present in the data.



- Outliers are present in almost all the columns except for pmonth
- We used zscore method to remove the outliers but data loss was around 16% which I can't afford to lose.
- We used quantile method to remove the outliers and we were able to remove outiers to some extent with minimum data loss.

- <u>Hardware and Software Requirements and Tools Used</u>

Hardware specifications are :-

Ryzen 5

16gb ram

RTX 2070 super graphics

Software used :-

Operating system : windows 10

Jupyter notebook and anaconda navigator- for coding and using data analytics tools and libraries.

Libraries used are matplot and seaborn for visualization purpose and scipy and sklearn for building models and pre-processing of data.

g

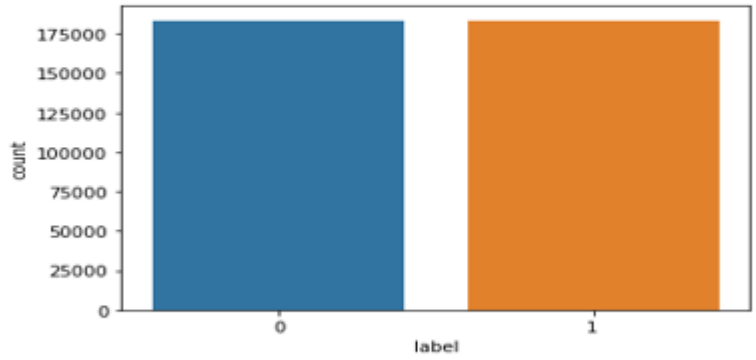
Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)
  There were no nan values present in the data.
  We found outliers and removed them with quantile method as we cannot proceed to model building without removing the outliers.
  We removed the skewness from non-categorical variables using power-transform method.
  We used standard scaler to scale the data.
  The data was not balanced so we balanced the data using SMOTE over sampling.

```
from imblearn.over_sampling import SMOTE
sm=SMOTE()
x,y=sm.fit_resample(dfx,dfy)
```

```
sns.countplot(y)
```

```
<AxesSubplot:xlabel='label', ylabel='count'>
```



the data is now balanced

- Testing of Identified Approaches (Algorithms)

- We split the data into into X and Y where x hold all the input variables after performing all the EDA and cleansing of data and Y holds the output variable which is a label.
- Our output variable is categorical nature so it's a classification problem for which we used classification models to predict values.
- We used train test split to split the data and perform machine learning tasks.

- Run and Evaluate selected models

- We used train_test_split to split the data into xtrain,xtest and ytrain,ytest with test size set to 0.20 which is 20% of the data set. It means that ytest and xtest will hold 20% of the random data which the model will use for testing purpose and predict the data for the xtest and check how accurate it is with ytest.
- Below is the snapshot of the code :-

Classification Models

```
[42]: lg=LogisticRegression()
      dtc=DecisionTreeClassifier()
      etc=ExtraTreesClassifier()
      sgdc=SGDClassifier()

      model=[lg,dtc,etc,sgdc]
```

```
[43]: #testing the models and checking their accuracy, cross_val_score as well as roc_auc-score
      xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.20,random_state=45)
      for m in model:
          m.fit(xtrain,ytrain)
          m.score(xtrain,ytrain)
          pred=m.predict(xtest)
          print('Accuracy score of ',m,'is :')
          print(accuracy_score(ytest,pred))
          score=cross_val_score(m,x,y,cv=5).mean()
          print('cross_val mean score :',score)
          print(confusion_matrix(ytest,pred))
          print(classification_report(ytest,pred))
          print('roc auc score :',roc_auc_score(ytest,pred))
          print('\n')
```

- We included cross val score and cv set to 5 and checking the mean score. Random state is set to 45, we will be checking the roc_auc score for accuracy results.
- Below is the image of the results that we got from the above code:-

```
Accuracy score of  LogisticRegression() is :
0.7867880555517697
cross_val mean score : 0.7846820404371593
[[29504  7235]
 [ 8409 28225]]
              precision    recall  f1-score   support

           0       0.78      0.80      0.79     36739
           1       0.80      0.77      0.78     36634

    accuracy                           0.79     73373
   macro avg       0.79      0.79      0.79     73373
weighted avg       0.79      0.79      0.79     73373

roc auc score : 0.7867647215399677


Accuracy score of  DecisionTreeClassifier() is :
0.8943889441620214
cross_val mean score : 0.8878707854261394
[[33191  3548]
 [ 4201 32433]]
              precision    recall  f1-score   support

           0       0.89      0.90      0.90     36739
           1       0.90      0.89      0.89     36634

    accuracy                           0.89     73373
   macro avg       0.89      0.89      0.89     73373
weighted avg       0.89      0.89      0.89     73373

roc auc score : 0.8943759919475415
```

```
Accuracy score of  ExtraTreesClassifier() is :
0.9409728374197593
cross_val mean score : 0.9392824611880473
[[34647  2092]
 [ 2239 34395]]
              precision    recall  f1-score   support

           0       0.94      0.94      0.94     36739
           1       0.94      0.94      0.94     36634

    accuracy                           0.94     73373
   macro avg       0.94      0.94      0.94     73373
weighted avg       0.94      0.94      0.94     73373

roc auc score : 0.9409698494904948


Accuracy score of  SGDClassifier() is :
0.7824404072342688
cross_val mean score : 0.7812965760041869
[[29445  7294]
 [ 8669 27965]]
              precision    recall  f1-score   support

           0       0.77      0.80      0.79     36739
           1       0.79      0.76      0.78     36634

    accuracy                           0.78     73373
   macro avg       0.78      0.78      0.78     73373
weighted avg       0.78      0.78      0.78     73373

roc auc score : 0.7824131440695911
```

What we observed from the results:-

- logistic regression gave us close to 79% accuracy
- decision tree gave us close to 90% accuracy
- extratreesclassifier is giving us a very high accuracy with close to 95% accuracy and it is the highest among others.

- Key Metrics for success in solving problem under consideration

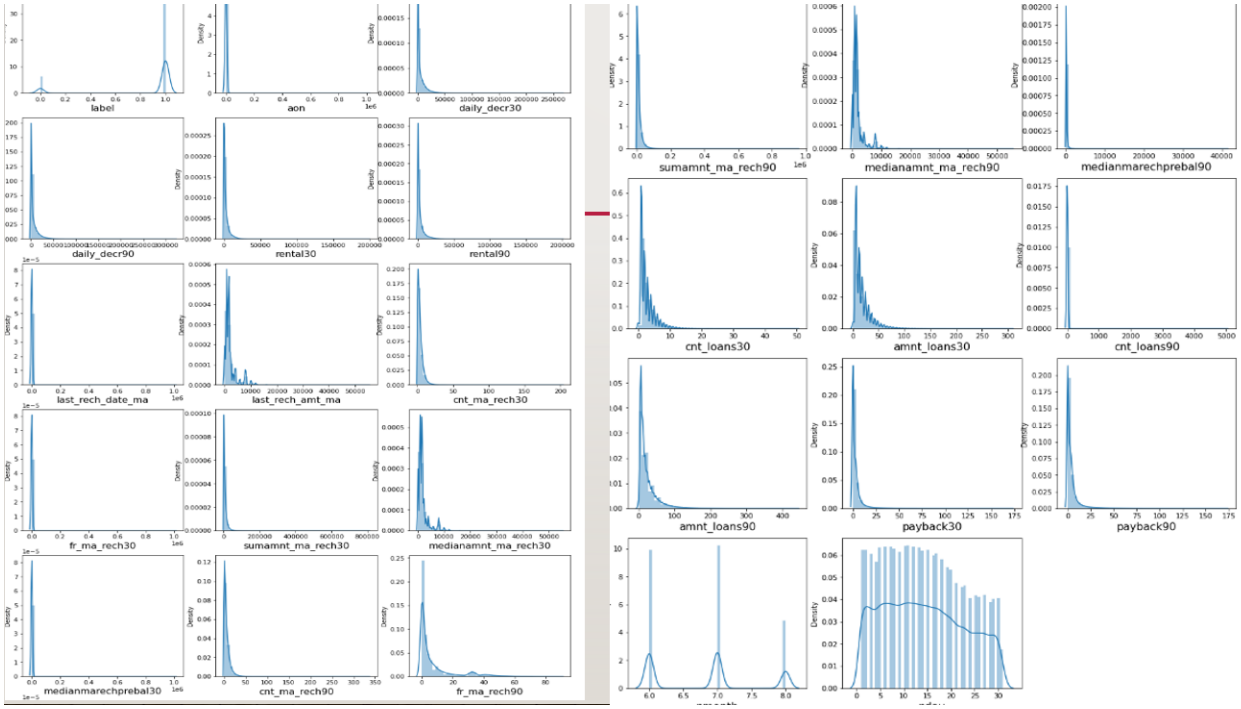
We used roc_auc_score for checking accuracy of the model.

We used cross val score and checking the mean score with sv set to 5 and checking the difference in the mean cross val score and roc-auc score to avoid other problems and choose the models with best accuracy and least difference.

We used hyper tuning to improve the model and when we the model was tested on the whole dataset the accuracy was 99%.

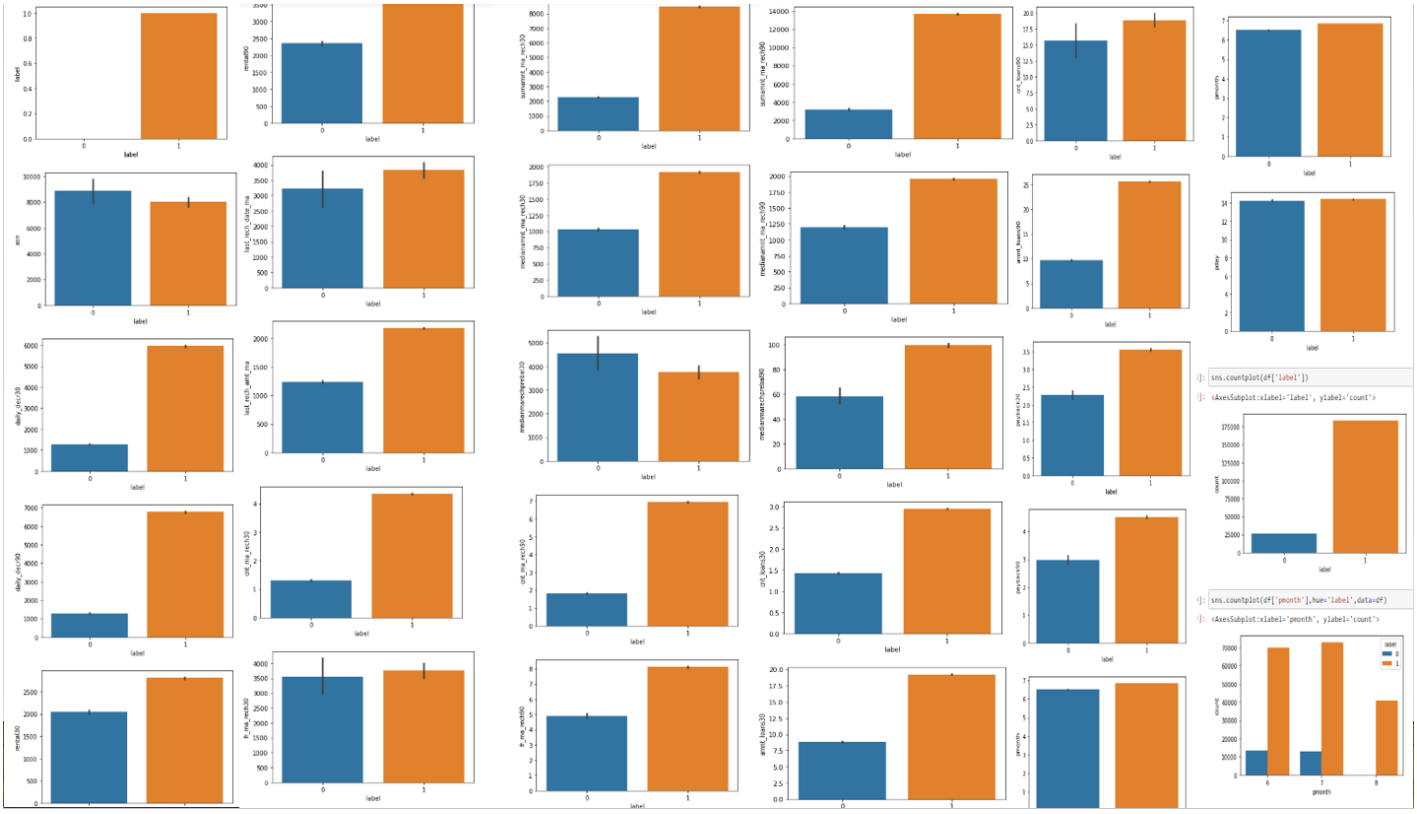
- Visualizations

  We used distribution plots to check the normal distribution in the variables.



- there is skewness in most of the columns so we have to treat them.
- pmonth and pday shows biomodal distribution.

We used barplots for all the variables with refrence to label to check how its affects our output variable and who are most likely to pay their loans.



Observation from the above graphs :-

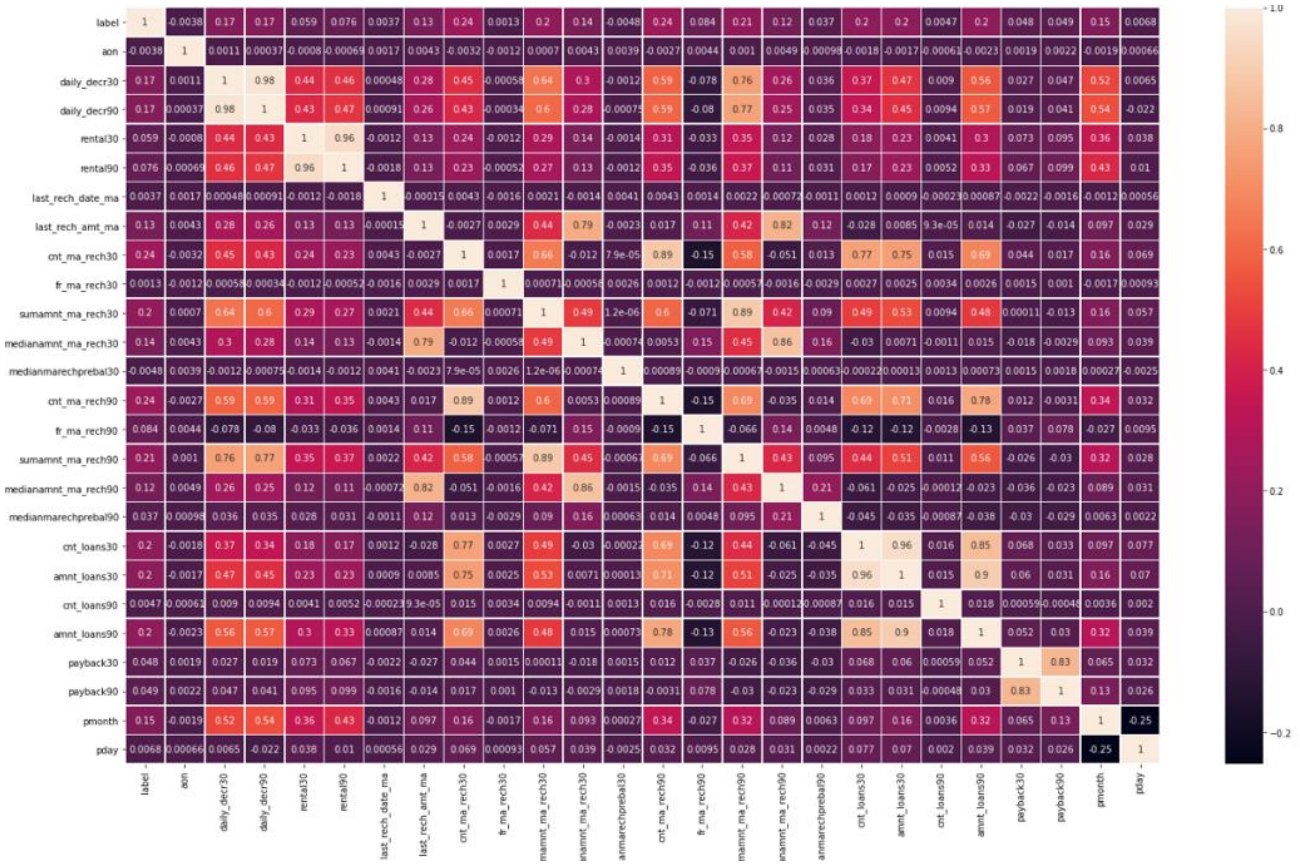
- People with longer duration of network usage are maximum defaulters
- People with higher Median of main account balance just before recharge in last 30 days at user level are maximum defaulters
- people with majority of Daily amount spent from main account, averaged over last 30 days (daily_decr30) are most likely to pay their loan.
- people with majority of Daily amount spent from main account, averaged over last 90 days(daily_decr90) are most likely to pay their loan.
- people with majority of Average main account balance over last 30 days(rental30) are most likely to pay their loan.

- people with majority of Average main account balance over last 90 days(rental90) are most likely to pay their loan.
- people with majority of days till last recharge of main account(last_rech_date_ma) are most likely to pay their loan.
- people with majority of Amount of last recharge of main account (last_rech_amt_ma) are most likely to pay their loan.
- people with majority of Number of times main account got recharged in last 30 days(cnt_ma_rech30) are most likely to pay their loan.
- people with majority of Frequency of main account recharged in last 30 days(fr_ma_rech30) are are most likely to pay their loan and also the count is high for defaulters comparitively Non-defaulters are more in number.
- people with majority of Total amount of recharge in main account over last 30 days (sumamnt_ma_rech30) are most likely to pay their loan.
- people with majority of Median of amount of recharges done in main account over last 30 days at user level (medianamnt_ma_rech30) are most likely to pay their loan.
- people with majority of Median of main account balance just before recharge in last 30 days at user level (medianmarechprebal30) are most likely to pay their loan.
- people with majority of Number of times main account got recharged in last 90 days(cnt_ma_rech90) are most likely to pay their loan.
- people with majority of Frequency of main account recharged in last 90 days(fr_ma_rech90) are most likely to pay their loan.

- people with majority of Total amount of recharge in main account over last 90 days (sumamnt_ma_rech90) are most likely to pay their loan.
- people with majority of Median of amount of recharges done in main account over last 90 days at user level (medianamnt_ma_rech90) are most likely to pay their loan.
- people with majority of Median of main account balance just before recharge in last 90 days at user level (medianmarechprebal90) are most likely to pay their loan.
- people with majority of Number of loans taken by user in last 30 days(cnt_loans30) are most likely to pay their loan.
- people with majority of Total amount of loans taken by user in last 30 days(amnt_loans30) are most likely to pay their loan.
- people with majority of maximum amount of loan taken by the user in last 30 days(maxamnt_loans30) are most likely to pay their loan.
- people with majority of Number of loans taken by user in last 90 days(cnt_loans90) are most likely to pay their loan.
- people with majority of Total amount of loans taken by user in last 90 days(amnt_loans90) are most likely to pay their loan.
- people with majority of maximum amount of loan taken by the user in last 90 days(maxamnt_loans90) are most likely to pay their loan.
- people with majority of Average payback time in days over last 30 days(payback30) are most likely to pay their loan.
- people with majority of Average payback time in days over last 90 days(payback90) are most likely to pay their loan.

- People having pmonth 8 have always payed back their loan

We used heatmap to check correlation between the variables



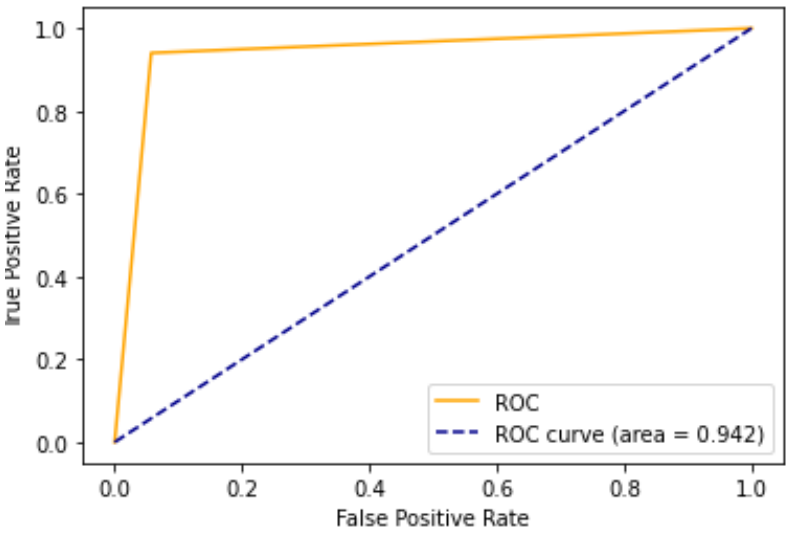
Observation from the above correlation heatmap

- pday, cnt_loans90, medianmarechprebal90, medianmarechprebal30, fr_ma_rech30, aon have very less correlation with the label
- cnt_ma_rech30, cnt_ma_rech90, sumamnt_ma_rech30 are some of the columns highest correlation with label
- we decide to drop pday and cnt_loans90, last_rech_date_ma as they have the lowest correlation with output variable

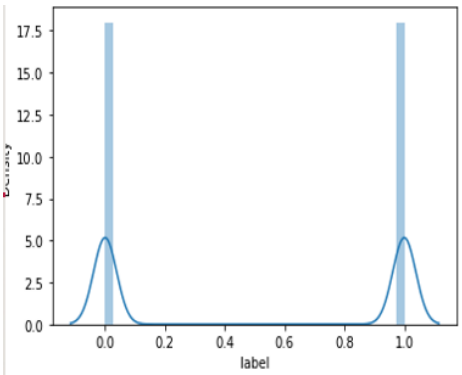
- Interpretation of the Results

- We saw that the data was not in its finest form we need to clear some nan values and outliers and clean the data.
- We also saw biasing in the data which we removed using power transform method.
- From visualization we found that cnt_ma_rech30 is highly correlated with label and least correlated is aon.

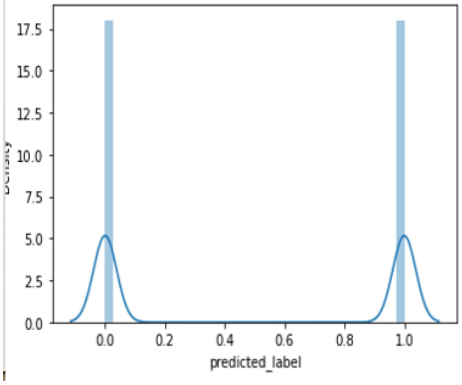
We found the best performing model after checking the metrics such as roc_auc score, cross val mean score which is Extra Trees classifier with the accuracy of 94% after hyper tuning.



After selecting the models we compared the model's with the predicted values and actual value.




```
AxesSubplot:xlabel='predicted_label', ylabel='Density'
```



We can see that the actual data and predicted data are very close to each other. This indicates that our model is giving good results with accuracy of 91%.

Conclusion

- Key Findings and Conclusions of the Study
- We had to perform analysis on the train data and clean the data and build a model which will predict the house prices for the test data.
- We did some visualization and pre-processing and after scaling the data we were able to build a model with close to 95% accuracy.
- We used the model to predict label value for the test data. Here is the image of the results:-

	predicted_label	label
0	0	0
1	1	1
2	1	1
3	1	1
4	1	1
...	...	...
366857	0	0
366858	0	0
366859	0	0
366860	0	0
366861	0	0

- Learning Outcomes of the Study in respect of Data Science
- We found the important features and how do they affect positively and negatively with label values of the micro credict loans and used them to predict the defaulters and non defaulters list using machine learning models.

- I learned that using data science it can help the business to make better strategies and focus on the areas which affects the output variables and make better decisions.

- Limitations of this work and Scope for Future Work
- We can see the data was not in its finest form with some more analysis we can still improve the accuracy of the model.
- The goal was to achieve 100% accuracy in predicting the house prices but we ended up with the accuracy of 94%.
- Due to lack of experience I was not able to reach my goal but with working on more such projects and gaining more experience will help me to grow and develop more valuable skills to reach my goal in future.