FLIP ROBO

MICRO CREDIT DEFAULTER PROJECT

Submitted by:

Raj

ACKNOWLEDGMENT

1) Rural Micro Credit Assessment using Machine Learning in a Peruvian microfinance institution - Henry Ivan Condori-Alejoa , Miguel Romilio Aceituno-Rojoa , Guina Sotomayor Alzamoraa,
2) Why Tree-Based Models are Preferred in Credit Risk Modeling? - https://analyticsindiamag.com/why-tree-based-models-are-preferred-in-credit-risk- modeling/

INTRODUCTION

- Business Problem Framing

  Micro Credit and Finance is that giving small loans to the people with low income, poor families etc. for their well-being and growth. Similarly it is applied by the telecom industries to provide some talk time balance and to be repaid by the customer in 5 days. But there is a huge risk that the customers are not paying the credit which is a great loss to company.

- 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.
- 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).

- ## Review of Literature

From the Literature survey, we have seen how important a Micro Credit and Finance works for the betterment of backward and poor people, but there are challenges faced by the MFI's that they are not getting paid back and facing huge losses.

So they should be careful and consider necessary background checks, whether the loan taker can pay back the debt or not by checking their account history, transaction details and predicting they can pay or not or how much amount at what interest rate can be given.

This is rigorous and time taking process with lot of procedures, complexities and verifications though their accuracy rate in that is around 70% only in estimating.

We are going to make this process simpler, reduce time and costs involved by implementing a Machine Learning model to understand the behaviour and predict. Different models re tried and find the better model that give us better accuracy in predicting to decide whether the credit should be given or not. We can see the model giving more accurate results than in the traditional way of deciding.

- ## Motivation for the Problem Undertaken

To make the service available to the more needful and poor people by the MFI's. More such investors and companies to come forward for keeping such business models and extend support. To make that possible. In such motive and this inspiration made me take one of this kind of project in Telecom Industry to give credit when needful to user.

Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem

  I had tried different Mathematical classification models on the data, i.e.

Model	Accuracy Score	F1 Score	CV Score	Difference
XGBoost classifier	0.927	0.93	0.921	0.08
Random Forest classifier	0.939	0.94	0.941	0.001
Gradient Boosting classifier	0.864	0.86	0.858	0.002
Ada Boost classifier	0.806	0.81	0.805	0.005

We have observed Random Forest classifier model is giving the best results with least difference between cv score and f1 score. So this is our best mathematical model to work with.

- Data Sources and their formats

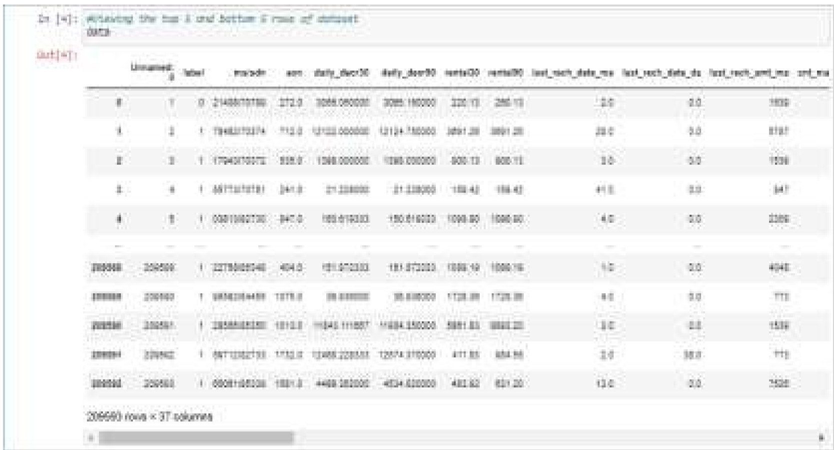
  Data Source: Sample Data given by the

  client Data type: .csv file

  Data Description: The complete data given is the records of 3 months details of a one particular year and single network operator.

  Year: 2016, Months: 6,7 & 8 , Operator: UPW

Data shape: 209593 rows x 37 columns



```
#Viewing the top 5 and bottom 5 rows of dataset
data
```

	Unnamed: 0	label	msisdn	aon	daily_decr30	daily_decr90	rental30	rental90	last_rech_date_ma	last_rech_date_da	last_rech_amt_ma	cnt_ma
0	1	0	21408I70789	272.0	3055.050000	3065.150000	220.13	260.13	2.0	0.0	1539	
1	2	1	76462I70374	712.0	12122.000000	12124.750000	3691.26	3691.26	20.0	0.0	5787	
2	3	1	17943I70372	535.0	1398.000000	1398.000000	900.13	900.13	3.0	0.0	1539	
3	4	1	55773I70781	241.0	21.228000	21.228000	159.42	159.42	41.0	0.0	947	
4	5	1	03813I82730	947.0	150.619333	150.619333	1098.90	1098.90	4.0	0.0	2309	
...	...	...	...	...	...	...	...	...	...	...	...	
209588	209589	1	22758I85348	404.0	151.872333	151.872333	1089.19	1089.19	1.0	0.0	4048	
209589	209590	1	95583I84455	1075.0	36.936000	36.936000	1728.36	1728.36	4.0	0.0	773	
209590	209591	1	28556I85350	1013.0	11843.111667	11904.350000	5861.83	8893.20	3.0	0.0	1539	
209591	209592	1	59712I82733	1732.0	12488.228333	12574.370000	411.83	984.58	2.0	38.0	773	
209592	209593	1	65061I85339	1581.0	4489.362000	4534.820000	483.92	631.20	13.0	0.0	7526	

209593 rows × 37 columns

Data Types:

```
data.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 209593 entries, 0 to 209592
Data columns (total 37 columns):
 #   Column                Non-Null Count   Dtype  
---  ------                --------------   -----  
 0   Unnamed: 0            209593 non-null  int64  
 1   label                 209593 non-null  int64  
 2   msisdn                209593 non-null  object 
 3   aon                   209593 non-null  float64
 4   daily_decr30          209593 non-null  float64
 5   daily_decr90          209593 non-null  float64
 6   rental30              209593 non-null  float64
 7   rental90              209593 non-null  float64
 8   last_rech_date_ma     209593 non-null  float64
 9   last_rech_date_da     209593 non-null  float64
 10  last_rech_amt_ma      209593 non-null  int64  
 11  cnt_ma_rech30         209593 non-null  int64  
 12  fr_ma_rech30          209593 non-null  float64
 13  sumamnt_ma_rech30     209593 non-null  float64
 14  medianamnt_ma_rech30  209593 non-null  float64
 15  medianmarechprebal30  209593 non-null  float64
 16  cnt_ma_rech90         209593 non-null  int64  
 17  fr_ma_rech90          209593 non-null  int64  
 18  sumamnt_ma_rech90     209593 non-null  int64  
 19  medianamnt_ma_rech90  209593 non-null  float64
 20  medianmarechprebal90  209593 non-null  float64
 21  cnt_da_rech30         209593 non-null  float64
 22  fr_da_rech30          209593 non-null  float64
 23  cnt_da_rech90         209593 non-null  int64  
 24  fr_da_rech90          209593 non-null  int64  
 25  cnt_loans30           209593 non-null  int64  
 26  amnt_loans30          209593 non-null  int64  
 27  maxamnt_loans30       209593 non-null  float64
 28  medianamnt_loans30    209593 non-null  float64
 29  cnt_loans90           209593 non-null  float64
 30  amnt_loans90          209593 non-null  int64  
 31  maxamnt_loans90       209593 non-null  int64  
 32  medianamnt_loans90    209593 non-null  float64
 33  payback30             209593 non-null  float64
 34  payback90             209593 non-null  float64
 35  pcircle               209593 non-null  object 
 36  pdate                 209593 non-null  object 
dtypes: float64(21), int64(13), object(3)
memory usage: 59.2+ MB
```

- Data Preprocessing Done
  1) Verified the observed Data Types to the Machine displayed data type.
  2) Removed the unnecessary columns.
  3) Checked for null values.
  4) Removed the columns that contain only one value.
  5) Found the skewness
  6) Removed the outliers to reduce skewness.
  7) Since the outliers are more and huge data loss, threshold is set to <6, where we removed the outliers in acceptable range of loss.
  8) Then we removed the skewness in all columns.
  9) Found the correlation and VIF to find the multicollinearity and remove those columns with high multicollinearity.
  10) We separated the Features and Target into x and y.
  11) Applied scaling on features.
  12) Then finally we made the dataset balanced before sending data to make model.

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

  Assumptions:

In general by any financial organisation while giving a credit they will check their previous transactions, their loans, income etc. for the past 3 months records minimum to check and finalize whether to give credit or not, if given how much can be given and at what interest rate.

So we do consider the same and drop all the columns with data of 30 days.

- Hardware and Software Requirements and Tools Used

  Hardware: 8GB Ram, i5 processor,

  Windows10 Software used:

  Anaconda

  Framework Jupyter

  Notebook Python:

  3.8.3

  xgboost==1.5.0 , ML model
  statsmodels==0.11.1
  seaborn==0.11.0 To plot
  graphs
  scipy==1.5.0 To perform statistical operations, find
  outliers. scikit-learn==1.0 To use pre built machine
  learning models pandas==1.0.5 To clean, manipulate,
  organise data, make
  dataframe
  numpy==1.18.5
  matplotlib==3.2.2 To plot
  graphs
  joblib==0.16.0 To save the final model, that can be used
  for deployment

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

  The approach I identified is, since the target is 0 or 1, binary. I considered the problem as Classification model.

  The dataset is so huge, so we considered the bagging/ boosting methods of machine learning models.

So from Ensemble module, we imported the algorithms and checked the model.

- Testing of Identified Approaches (Algorithms)

The models thus made are evaluated using

1) Accuracy score
2) Confusion matrix
3) Classification report
4) Cross validation

   Each model is checked for the above metrics, and done cross validation to determine the best model, and improve the model further using tuning.

- Run and Evaluate selected models Models :

```
from xgboost import XGBClassifier
from sklearn.ensemble import RandomForestClassifier, GradientBoostingClassifier, AdaBoostClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score,classification_report,confusion_matrix
from sklearn.model_selection import cross_val_score
models=[XGBClassifier(),RandomForestClassifier(),GradientBoostingClassifier(), AdaBoostClassifier()]
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.3,random_state=47)
for m in models:
    m.fit(x_train,y_train)
    predm=m.predict(x_test)
    ac=accuracy_score(y_test,predm)
    cm=confusion_matrix(y_test,predm)
    cr=classification_report(y_test,predm)
    cvscore=cross_val_score(m,x,y,cv=5)
    print(f'Metrics of {m}:\n')
    print('accuracy score:',ac)
    print('confusion matrix:\n',cm)
    print('classification report:\n',cr)
    print('Mean cv score:',cvscore.mean())
    print('\n\n')
```

Results:

XGBoost

```
Metrics of XGBClassifier(base_score=0.5, booster='gbtree', colsample_bylevel=1,
              colsample_bynode=1, colsample_bytree=1, enable_categorical=False,
              gamma=0, gpu_id=-1, importance_type=None,
              interaction_constraints='', learning_rate=0.300000012,
              max_delta_step=0, max_depth=6, min_child_weight=1, missing=nan,
              monotone_constraints='()', n_estimators=100, n_jobs=4,
              num_parallel_tree=1, predictor='auto', random_state=0,
              reg_alpha=0, reg_lambda=1, scale_pos_weight=1, subsample=1,
              tree_method='exact', validate_parameters=1, verbosity=None):

accuracy score: 0.9267290124003107
confusion matrix:
 [[46432  4312]
 [ 3139 47808]]
classification report:
               precision    recall  f1-score   support

         0.0       0.94      0.92      0.93     50744
         1.0       0.92      0.94      0.93     50947

    accuracy                           0.93    101691
   macro avg       0.93      0.93      0.93    101691
weighted avg       0.93      0.93      0.93    101691

Mean cv score: 0.9206655456235066
```

Random Forest Classifier:

```
Metrics of RandomForestClassifier():

accuracy score: 0.938814644363808
confusion matrix:
 [[47650  3094]
 [ 3128 47819]]
classification report:
               precision    recall  f1-score   support

         0.0       0.94      0.94      0.94     50744
         1.0       0.94      0.94      0.94     50947

    accuracy                           0.94    101691
   macro avg       0.94      0.94      0.94    101691
weighted avg       0.94      0.94      0.94    101691

Mean cv score: 0.9407263179632416
```

Gradient Boosting Classifier:

```
Metrics of GradientBoostingClassifier():

accuracy score: 0.8639407617193262
confusion matrix:
 [[44638  6106]
 [ 7730 43217]]
classification report:
              precision    recall  f1-score   support

         0.0       0.85      0.88      0.87     50744
         1.0       0.88      0.85      0.86     50947

    accuracy                           0.86    101691
   macro avg       0.86      0.86      0.86    101691
weighted avg       0.86      0.86      0.86    101691

Mean cv score: 0.8580405345605806
```

Ada Boost:

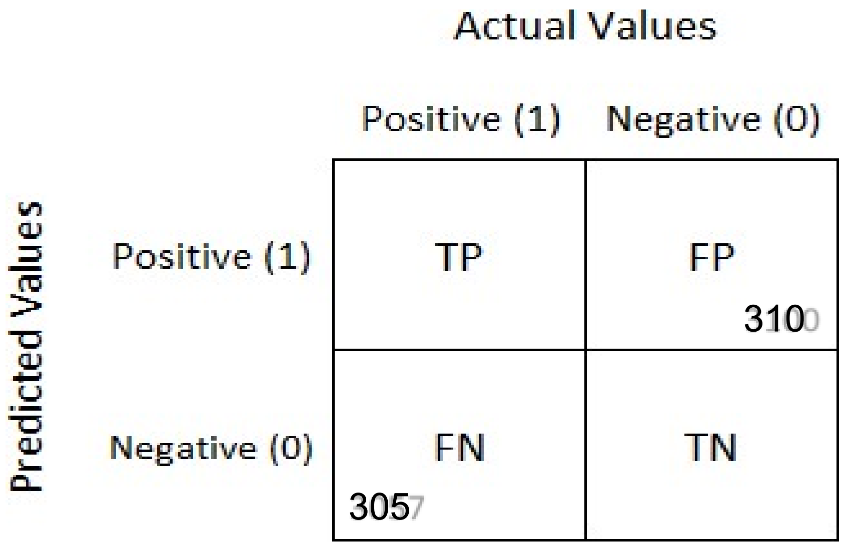
```
Metrics of AdaBoostClassifier():

accuracy score: 0.8061283692755504
confusion matrix:
 [[41570  9174]
 [10541 40406]]
classification report:
              precision    recall  f1-score   support

         0.0       0.80      0.82      0.81     50744
         1.0       0.81      0.79      0.80     50947

    accuracy                           0.81    101691
   macro avg       0.81      0.81      0.81    101691
weighted avg       0.81      0.81      0.81    101691

Mean cv score: 0.8051715491046405
```

- Key Metrics for success in solving problem under consideration

  First Key metric:

  Least difference between the F1 score and Mean CV score, gives us the confidence to decide which the best model is, basing the fit of the model.

Confusion Matrix report:



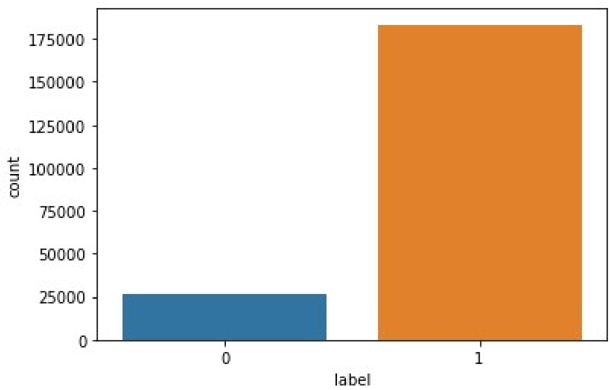
From confusion matrix we seen the best model considered the less FN.
It’s ok to cross verify, check or not give the credit to 3% of customers. We can improve the service.
But if we give to FN 3057 customers, we should face huge loss. Out of all the models the model we considered best is giving the less FN.

- Visualizations

Our data has 26162 records of unpaid. Remaining all paid customers details.

```
sns.countplot(df['label'])
print(df['label'].value_counts())
```

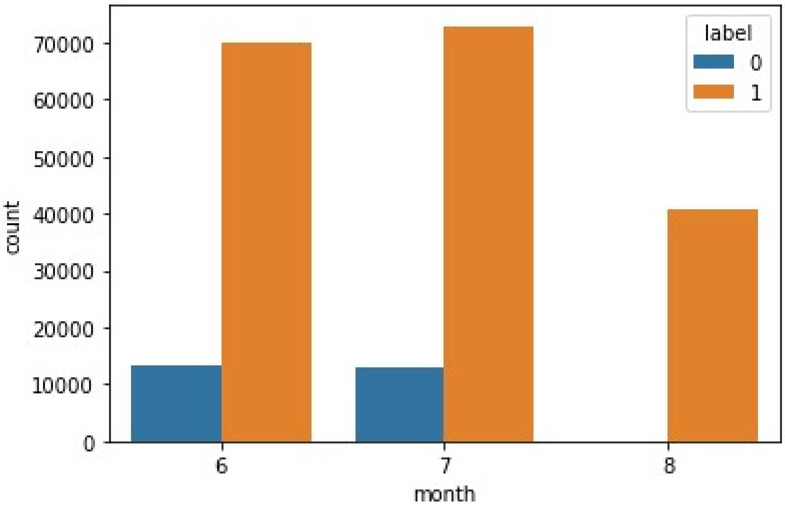
```
1    183431
0     26162
Name: label, dtype: int64
```

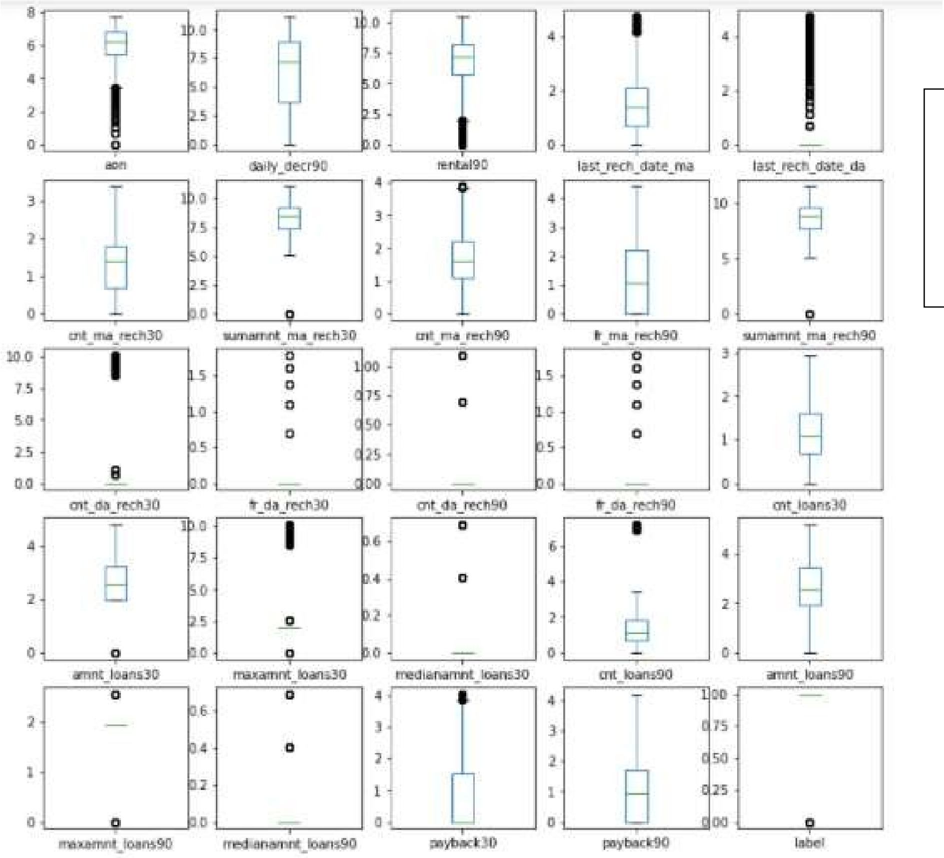


Most people not paid credit amount in 7th month.

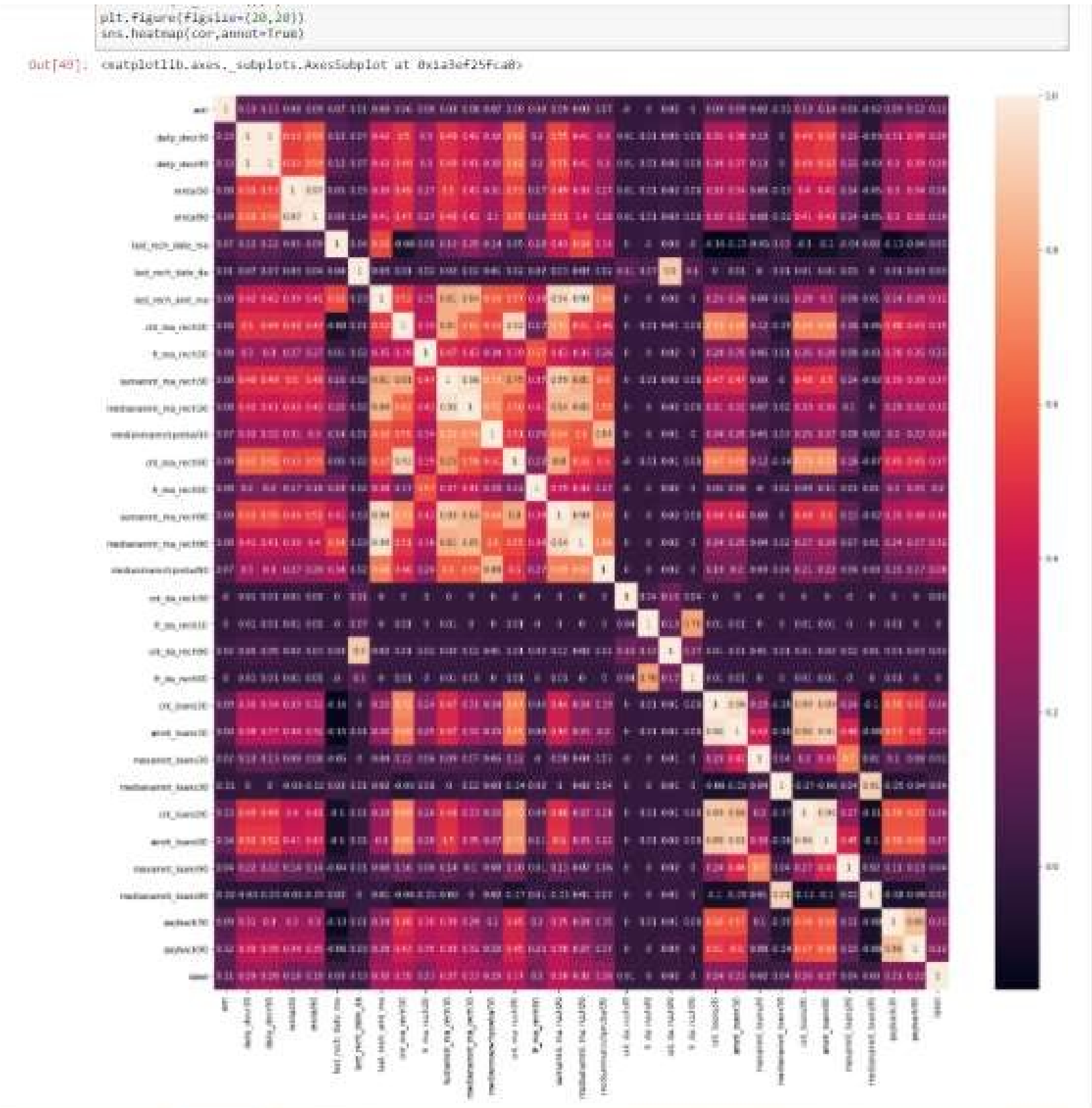
```
sns.countplot(df['month'],hue=df['label'])
print(df['month'].value_counts())
```

```
7    85765
6    83154
8    40674
Name: month, dtype: int64
```





We observed that all columns have outliers.



From this heat map we see which columns are highly correlated and how much is their effect in determining the credit payment will be done or not.

- Interpretation of the Results

From Heatmap of correlation, we can say:

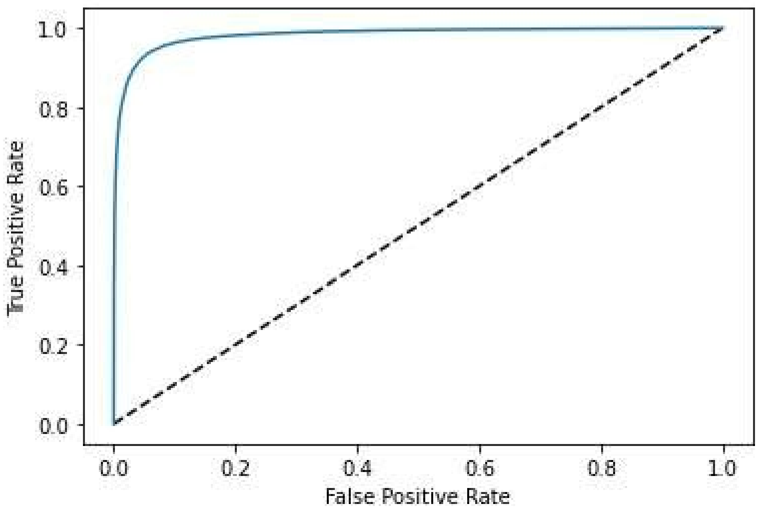
Target is highly correlated to the columns –

sum of amount recharged into main account in last 90 days.

Rental90, daily_decr90 . These 3 columns are mostly positively correlated to target. With the increase of values in this column, more probability that the customer pays back.

Frda_rech30 & Frda_rech90 has no relation to the target variable.

Finally: The confusion matrix report, f1 score are most considered to check the accuracy of model and AUC-ROC curve.




```
Score: 0.9396497000987022
```

CONCLUSION

- Key Findings and Conclusions of the Study
  1) More number of customers not paid in 7th month of 6 & 7 but almost equal.
  2) We found that users with high amount usage and recharge done to main account in last 90 days paid back. Based on this before giving credit these factors, customer usage should be verified, this can make huge impact in deciding to credit or not also reduce loss.

- Learning Outcomes of the Study in respect of Data Science

Visualisation made it simple to understand trends, find the underlying insights of data.

In finding the skewness, and outliers also, visualisation plays a key role.

It also made easier to understand the correlation between the columns and with target variable.

Also to determine the fit of the model whether it is over fitted or under fitted we check AUC-ROC curve.

Best Algorithm:

For this kind of classification problems the best models can be Tree based.

Since the data is huge we used ensemble techniques to make the work faster.

Challenges Faced:

When first observed for brief of statistics about the given data, there are negative values. That can't be possible since all the data we have are days, amount etc.. , this problem is overcome by assuming those values entered might be manual mistake and turned all the values into positive.

The other major problem, there are lot of columns, high multicollinearity between them. So difficult in deciding which column to be dropped and which to include. Overcame this bit by considering the correlation and High multicollinearity columns, which column is having less correlation with target, I dropped those columns.

Though there are still man columns left, considering them as data loss if I removed them too.

- Limitations of this work and Scope for Future Work

The model is limited to 91% accuracy only, where there are still some errors occurring in prediction who not pays credit.

I cannot perform tuning operation on the model to increase it's accuracy due to Hardware problem. This model can be tuned for getting better results by implementing with high configuration hardware systems.

Perform Hyper parameter tuning on the Random Forest classifier model.

We can still try dropping few existing columns and make the model with different algorithms.