FLIP ROBO

MicroCreditProject

Prediction

Submitted By

HariharaSudhan

Acknowledgement

I would like to express my deepest gratitude to my SME (Subject Matter Expert) Shwetank Mishra as well as Flip Robo Technologies who gave me the opportunity to do this project on Surprise Housing Price Prediction, which also helped me in doing lots of research wherein I came to know about so many new things.

Also, I have utilized a few external resources that helped me to complete the 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. It is widely accepted as a poverty-reduction tool, representing $70 billion in outstanding loans and a global outreach of 200 million clients. The client wants some predictions that could help them in further investment and improvement in selection of customers for the credit.

Conceptual BackGround of Domain Problem

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. 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). This problem contains data of customers who is defaulter / Non – defaulters and has the main account and data account recharge and total amount of sum amount and its frequency. So, we need to predict for each loan transaction, whether the customer will be paying back the loaned amount within 5 days of insurance of loan.

Review of Literature

We are going to work on the dataset provided for building a model to predict the nature of future customers –

a. Uploading the dataset
b. Exploratory analysis of features
c. Data -preprocessing, removing unnecessary features
d. Visualisaton of features, features along with target feature
e. Data cleaning by removing outliers or missing values treatment
f. Model pre-processing and model training
g. Tesing multiple algorithms with multiple evaluation metrics
h. Using the evaluation metrics best suited for our problem
i. Hyper-parameter tuning using RandomisedSearchCV for the best model parameter
j. Model prediction and saving the file in csv format
k. Saving the final model

Motivation for the Problem Undertaken

This project was provided to me by FlipRobo Technology as a part of internship program . This dataset helped me a lot for understanding the the real time problems with data we are going to face. It helps me in developing the skills needed for the future work.

Further diving into the dataset and understanding the current situation of my country due to covid-19 , I was highly motivated to take up this

project. Seeing the current covid situation, many people are facing health emergency .In this time they need to call ambulance,police, doctors ,relatives etc but due to lack of sufficient funds they may face calling problem. But this project is based on the solution of this problem ,as some Microfinance Institution are providing small credit loans to the poor familiers in remote areas.

The objective of this project is to prepare a model which can signify the difference in defaulters and non-defaulters and helps our client in further investment and selection of customers.

Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem 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. In the provided dataset, our target variable "label" is a categorical with two categories: " defaulter " and " Non- defaulter ". Therefore, we will be handling this modelling problem as classification.

- Data Sources and their formats

We can see that our target variable has more non-defaulters (paying loan on time) than defaulters (not paying loan on time)

Out[2]:

	Unnamed: 0	label	msisdn	aon	daily_decr30	daily_decr90	rental30	rental90	last_rech_date_ma	last_rech_date_da	...	maxamnt_loans30	mec
0	1	0	21408I70789	272.0	3055.050000	3065.150000	220.13	260.13	2.0	0.0	...	6.0	
1	2	1	76462I70374	712.0	12122.000000	12124.750000	3691.26	3691.26	20.0	0.0	...	12.0	
2	3	1	17943I70372	535.0	1398.000000	1398.000000	900.13	900.13	3.0	0.0	...	6.0	
3	4	1	55773I70781	241.0	21.228000	21.228000	159.42	159.42	41.0	0.0	...	6.0	
4	5	1	03813I82730	947.0	150.619333	150.619333	1098.90	1098.90	4.0	0.0	...	6.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	
209588	209589	1	22758I85348	404.0	151.872333	151.872333	1089.19	1089.19	1.0	0.0	...	6.0	
209589	209590	1	95583I84455	1075.0	36.936000	36.936000	1728.36	1728.36	4.0	0.0	...	6.0	
209590	209591	1	28556I85350	1013.0	11843.111667	11904.350000	5861.83	8893.20	3.0	0.0	...	12.0	
209591	209592	1	59712I82733	1732.0	12488.228333	12574.370000	411.83	984.58	2.0	38.0	...	12.0	
209592	209593	1	65061I85339	1581.0	4489.362000	4534.820000	483.92	631.20	13.0	0.0	...	12.0	

209593 rows × 37 columns

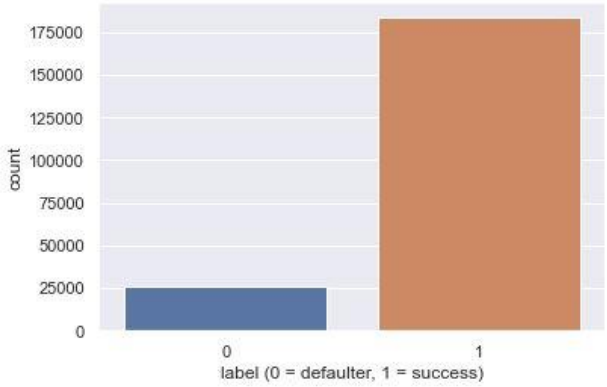
Data Formats

)ut[7]:

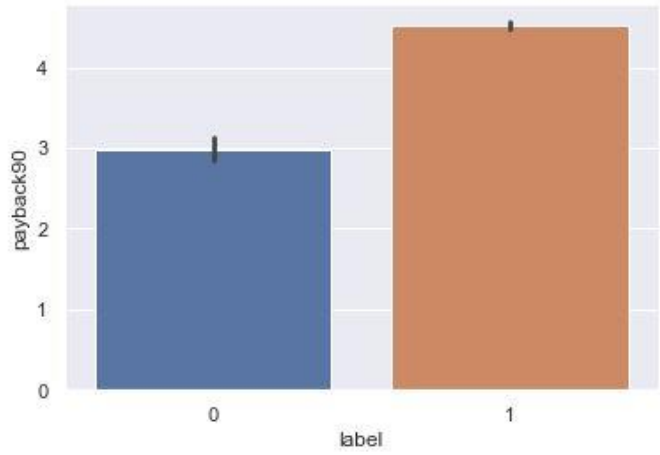
```
Unnamed: 0              int64
label                   int64
msisdn                 object
aon                   float64
daily_decr30          float64
daily_decr90          float64
rental30              float64
rental90              float64
last_rech_date_ma     float64
last_rech_date_da     float64
last_rech_amt_ma        int64
cnt_ma_rech30           int64
fr_ma_rech30          float64
sumamnt_ma_rech30     float64
medianamnt_ma_rech30  float64
medianmarechprebal30  float64
cnt_ma_rech90           int64
fr_ma_rech90            int64
sumamnt_ma_rech90       int64
medianamnt_ma_rech90  float64
medianmarechprebal90  float64
cnt_da_rech30         float64
fr_da_rech30          float64
cnt_da_rech90           int64
fr_da_rech90            int64
cnt_loans30             int64
amnt_loans30            int64
maxamnt_loans30       float64
medianamnt_loans30    float64
cnt_loans90           float64
```

EDA and Visualisation

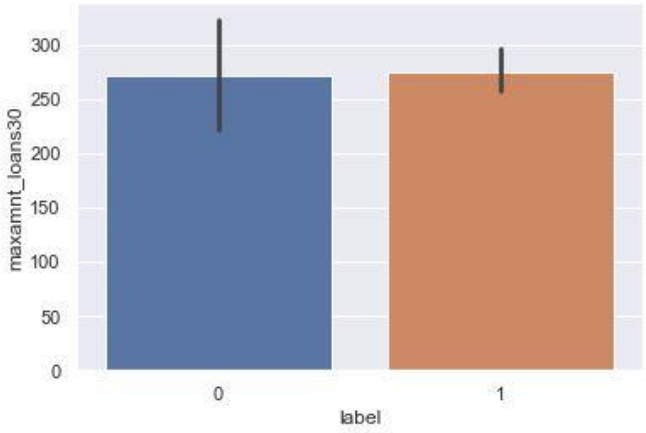
In [17]:
```
#Label
sns.set_theme()
sns.countplot(df['label'])
plt.xlabel('label (0 = defaulter, 1 = success)')
plt.show()
```



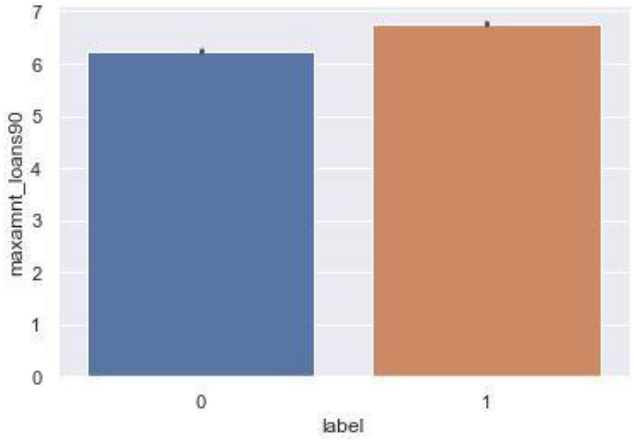
In [18]:
```
sns.set_theme()
sns.barplot(x = df['label'], y = df['payback90'])
plt.show()
```

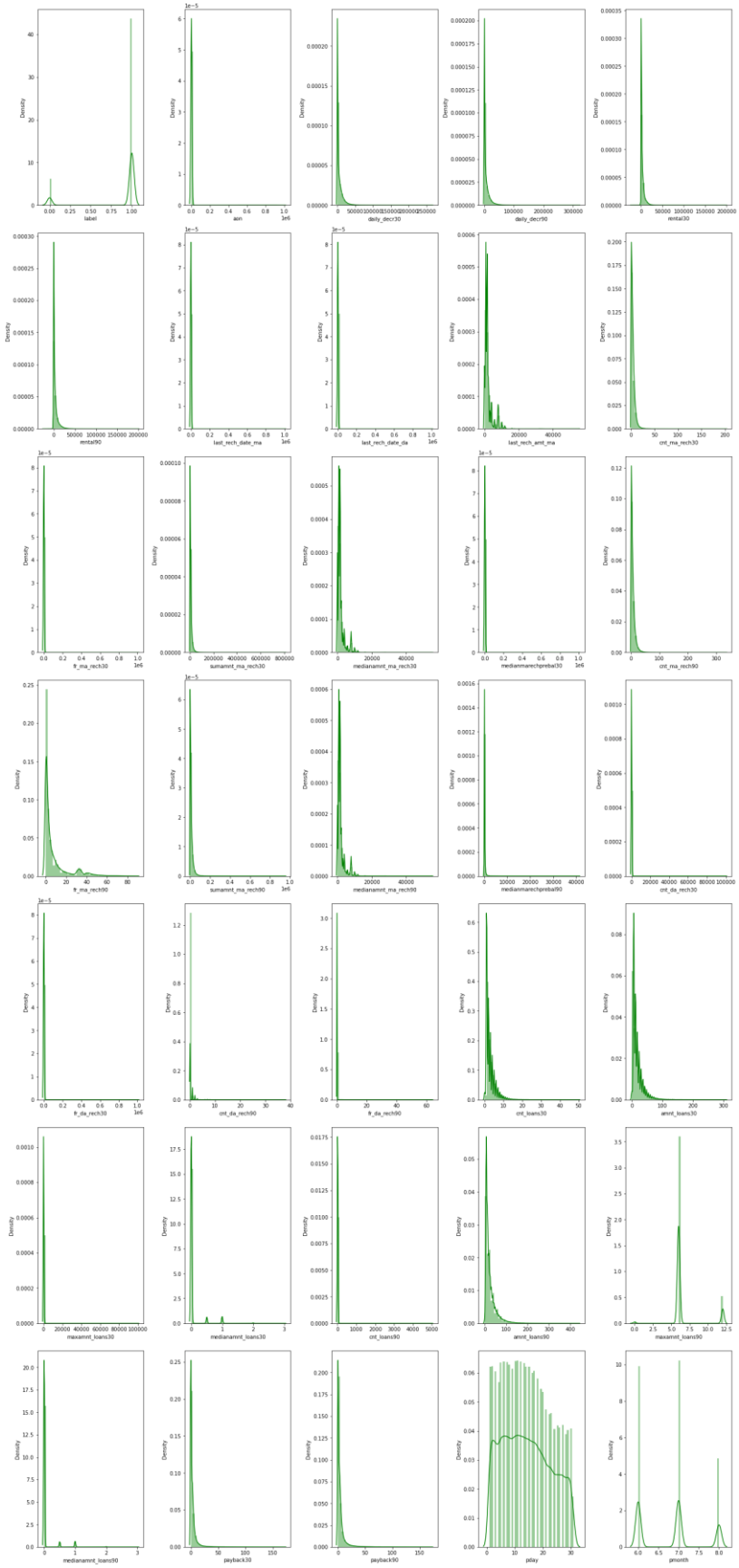


In [20]:
```
sns.set_theme()
sns.barplot(x = df['label'], y = df['maxamnt_loans30'])
plt.show()
```

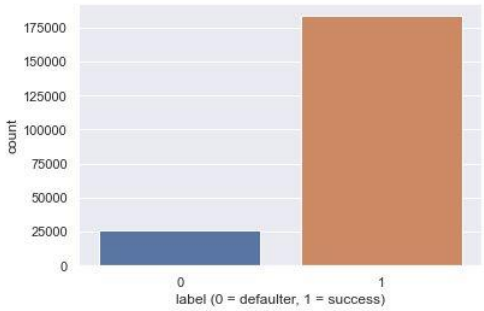


In [21]:
```
sns.set_theme()
sns.barplot(x = df['label'], y = df['maxamnt_loans90'])
plt.show()
```



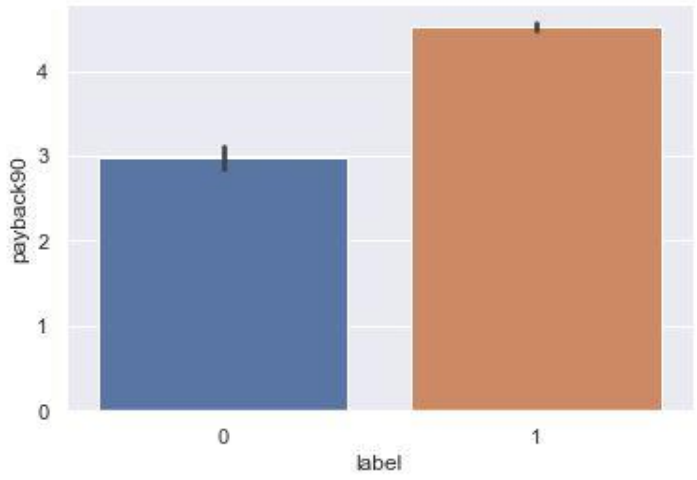

label
aon
daily_decr30
daily_decr90
rental30
rental90
last_rech_date_ma
last_rech_date_da
last_rech_amt_ma
cnt_ma_rech30
fr_ma_rech30
sumamnt_ma_rech30
medianamnt_ma_rech30
medianmarechprebal30
cnt_ma_rech90
fr_ma_rech90
sumamnt_ma_rech90
medianamnt_ma_rech90
medianmarechprebal90
cnt_da_rech30
fr_da_rech30
cnt_da_rech90
fr_da_rech90
cnt_loans30
amnt_loans30
maxamnt_loans30
medianamnt_loans30
cnt_loans90
amnt_loans90
maxamnt_loans90
medianamnt_loans90
payback30
payback90
pday
prioinfo

In [17]:
```
#Label
sns.set_theme()
sns.countplot(df['label'])
plt.xlabel('label (0 = defaulter, 1 = success)')
plt.show()
```



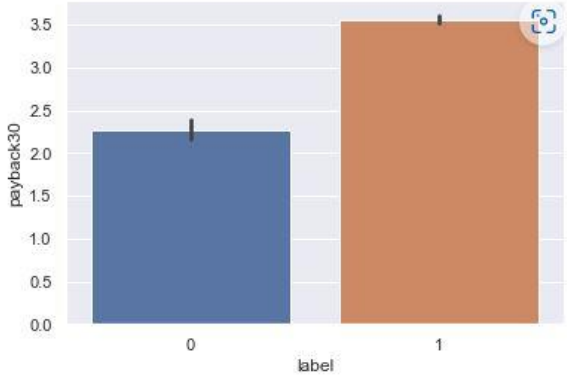
Most of the customers (non-defaulters) are paying back their loan by 3-5 days,

In [18]:
```
sns.set_theme()
sns.barplot(x = df['label'], y = df['payback90'])
plt.show()
```



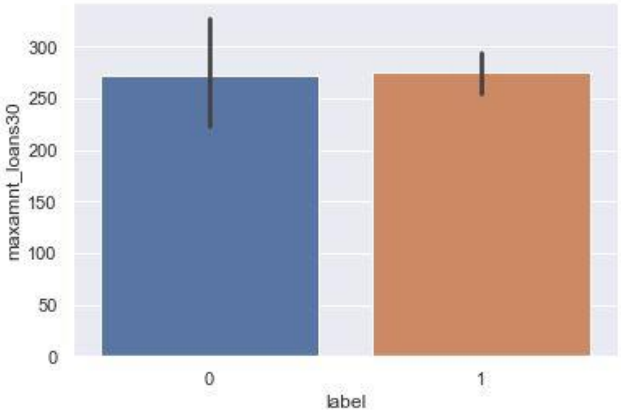

```
pay back time of potential defaulter in 90 days is 3 days.
pay back time of repayer in 90 days is greater than 4 days3
```

In [19]:
```
sns.set_theme()
sns.barplot(x = df['label'], y = df['payback30'])
plt.show()
```

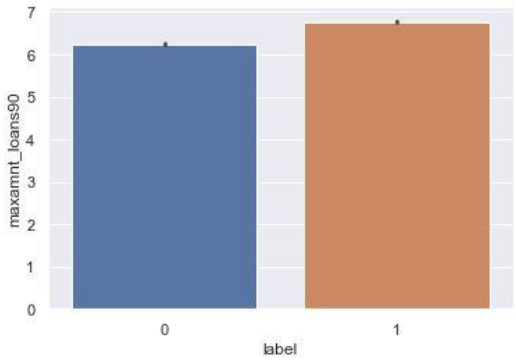


Maximum amount of loans is taken by defaulters <30 days and there are 2 options 5rs and 10 rs which customer needs to payback as 6rs and 12rs.

```
In [20]: sns.set_theme()
         sns.barplot(x = df['label'], y = df['maxamnt_loans30'])
         plt.show()
```




```
In [21]: sns.set_theme()
         sns.barplot(x = df['label'], y = df['maxamnt_loans90'])
         plt.show()
```



- **Data Preprocessing Done**

Replacing some of the 0 values to mean, median as it is having 0 values more and customer who got loan has to payback In 30 days and 90 days and frequency of main account and data account recharged and count of data account and main account of recharged.

If account got recharged and customers needs to payback the loan within their 30days and 90days. Also, we have outliers as well and tried applying Z-score method, we are losing >10% data, So I am removing skewness by using power transform method.

- **Data Inputs- Logic- Output Relationships**

Our target variable is label which indicates the customer is a defaulter who is not paying back or non-defaulter who is paying back the loan with some features like how often they are recharging, and daily amount spend by customer from main account and average main account balance and number of loans taken by user and maximum amount of loan taken by user in last 30 days or 90 days.

- **Hardware and Software Requirements and Tools Used**

1.Pandas is open-source library tool which provides high performance data analysis tool by its powerful data structures. It helps to shorten the procedure of handling the data with extensive set of features. 2.Numpy is most used package for scientific computing for multi-dimensional array of objects. 3.Other than this, as a pre-processing steps, I imported standard scaler for scaling the data. 4.I imported f1 score, classification report, confusion matrix, roc curve in terms of metrics to calculate the model score.

Models Development and Evaluation

- Testing of Identified Approaches (Algorithms)

I have used Decision tree algorithm, Random Forest, Ada Boost and Gradient Boost algorithm to calculate the score of the model.

- Run and Evaluate selected models DecisionTree model which has score – 88.10% and CV score – 88.25%.

DecisionTree Classifier

In [56]:
```
DT = DecisionTreeClassifier()
DT.fit(x_train,y_train)
pred_DT = DT.predict(x_test)
print(accuracy_score(y_test, pred_DT))
print(confusion_matrix(y_test, pred_DT))
print(classification_report(y_test,pred_DT))
```

```
0.8814862879062577
[[ 3624  2946]
 [ 3264 42565]]
              precision    recall  f1-score   support

           0       0.53      0.55      0.54      6570
           1       0.94      0.93      0.93     45829

    accuracy                           0.88     52399
   macro avg       0.73      0.74      0.74     52399
weighted avg       0.88      0.88      0.88     52399
```

RandomForest Classifier

Run and Evaluate selected models RandomForest model which has score – 91.86% and CV score – 91.98%.

In [60]:
```
RAN=RandomForestClassifier()
RAN.fit(x_train, y_train)
pred_RAN=RAN.predict(x_test)
print(accuracy_score(y_test, pred_RAN))
print(confusion_matrix(y_test, pred_RAN))
print(classification_report(y_test,pred_RAN))
```

```
0.9183190518903033
[[ 3377  3193]
 [ 1087 44742]]
              precision    recall  f1-score   support

           0       0.76      0.51      0.61      6570
           1       0.93      0.98      0.95     45829

    accuracy                           0.92     52399
   macro avg       0.84      0.75      0.78     52399
weighted avg       0.91      0.92      0.91     52399
```

LogisticRegression

Logistic Regression has scored 87.57% and cv is 50%

In [63]:
```
from sklearn.linear_model import LogisticRegression
```

In [64]:
```
log = LogisticRegression()
log.fit(x_train,y_train)
pred_log = log.predict(x_test)
print(accuracy_score(y_test, pred_log))
print(confusion_matrix(y_test, pred_log))
print(classification_report(y_test,pred_log))
```

```
0.8757228191377698
[[  127  6443]
 [   69 45760]]
              precision    recall  f1-score   support

           0       0.65      0.02      0.04      6570
           1       0.88      1.00      0.93     45829

    accuracy                           0.88     52399
   macro avg       0.76      0.51      0.49     52399
weighted avg       0.85      0.88      0.82     52399
```

In [65]:
```
print(cross_val_score(log,x,Y,cv=5).mean())
```

```
0.8763556008443762
```

In [66]:
```
roc_auc_score(y_test,pred_log)
```

Out[66]:
```
0.5089123461502528
```

Ada Boost Classifier

In [69]:
```
from sklearn.ensemble import AdaBoostClassifier
```

In [70]:
```
ADA=AdaBoostClassifier()
ADA.fit(x_train, y_train)
pred_ADA=ADA.predict(x_test)
print(accuracy_score(y_test, pred_ADA))
print(confusion_matrix(y_test, pred_ADA))
print(classification_report(y_test,pred_ADA))
```

```
0.9080325960419092
[[ 2443  4127]
 [  692 45137]]
              precision    recall  f1-score   support

           0       0.78      0.37      0.50      6570
           1       0.92      0.98      0.95     45829

    accuracy                           0.91     52399
   macro avg       0.85      0.68      0.73     52399
weighted avg       0.90      0.91      0.89     52399
```

In [71]:
```
print(cross_val_score(ADA,x,Y,cv=5).mean())
```

```
0.909295637751941
```

In [72]:
```
roc_auc_score(y_test,pred_ADA)
```

Out[72]:
```
0.6783710476503997
```

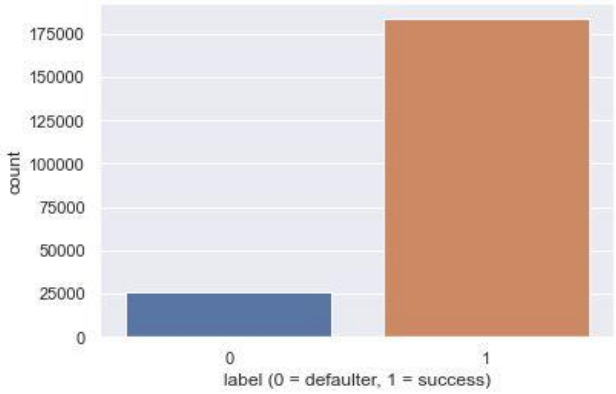
- **Key Metrics for success in solving problem under consideration**

Used F1 Score for calculating the accuracy score as the target variables classes are im balanced and accuracy score metric won't give correct results as it may take classes with more count. Classification report will display the overview of accuracy, precision, recall, f1 score , support and weighted average. Confusion matrix for calculating true positive and true negative.

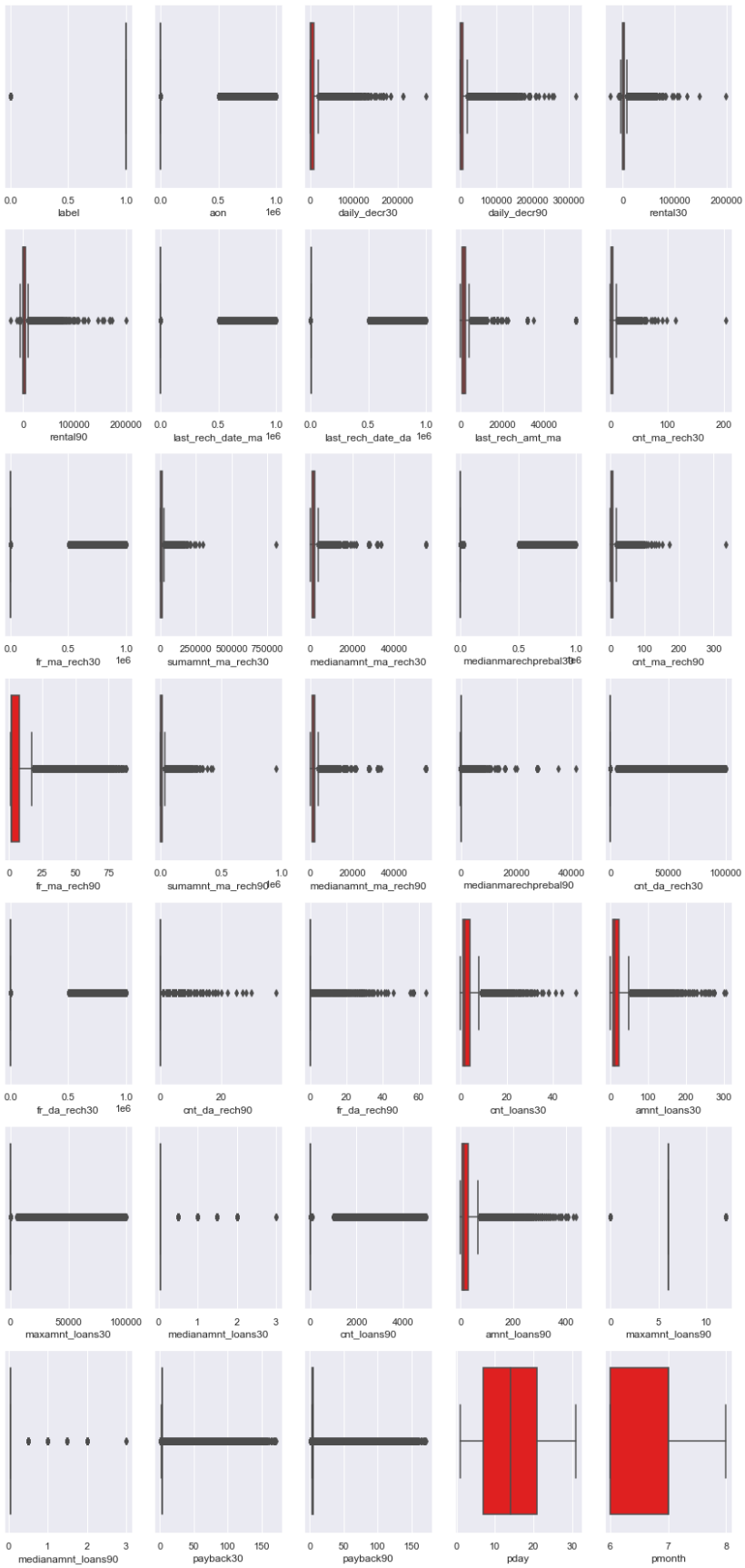
- **Visualizations**

Target variable plot where it shows the classes are im-balanced.

```
In [17]: #Label
         sns.set_theme()
         sns.countplot(df['label'])
         plt.xlabel('label (0 = defaulter, 1 = success)')
         plt.show()
```

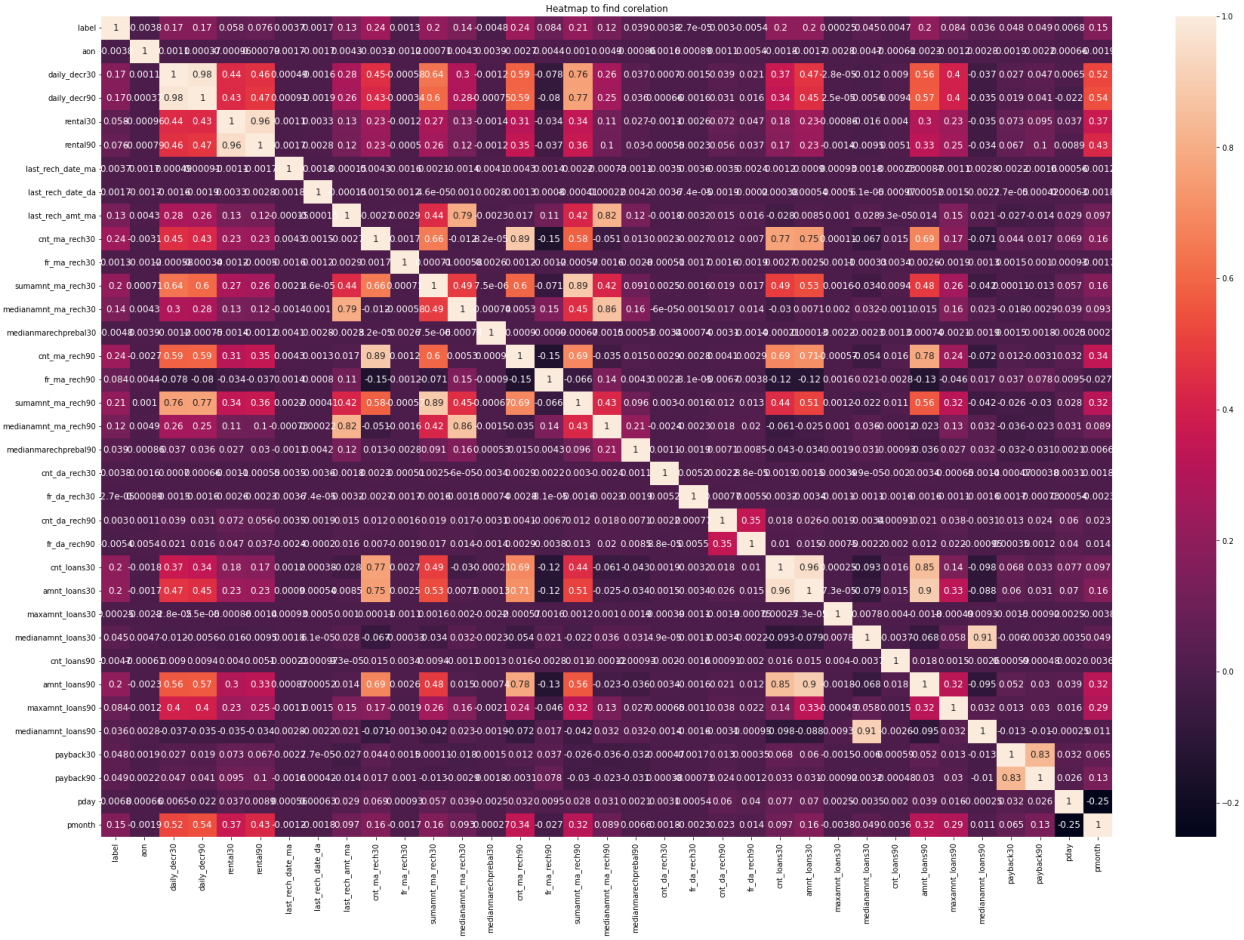


BoxPlot


label
aon
daily_decr30
daily_decr90
rental30
rental90
last_rech_date_ma
last_rech_date_da
last_rech_amt_ma
cnt_ma_rech30
fr_ma_rech30
sumamnt_ma_rech30
medianamnt_ma_rech30
medianmarechprebal30
cnt_ma_rech90
fr_ma_rech90
sumamnt_ma_rech90
medianamnt_ma_rech90
medianmarechprebal90
cnt_da_rech30
fr_da_rech30
cnt_da_rech90
fr_da_rech90
cnt_loans30
amnt_loans30
maxamnt_loans30
medianamnt_loans30
cnt_loans90
amnt_loans90
maxamnt_loans90
medianamnt_loans90
payback30
payback90
pday
pmonth

Interpretation of the Results

Correlation matrix after dropping less importance features and high skewed data,



Final model accuracy

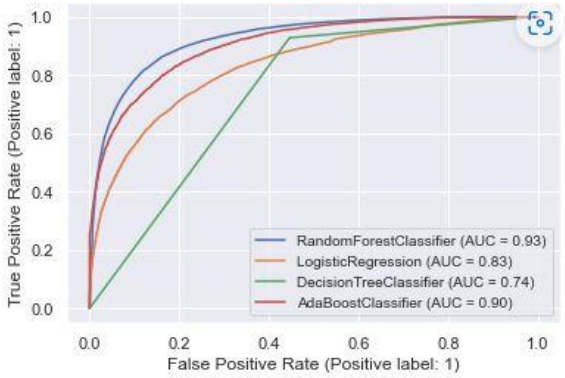
Random Forest Score – 91.86%

ROC curve of Final Model is

ROC AUC curves

In [74]:
```
from sklearn.metrics import plot_roc_curve
```

In [75]:
```
disp = plot_roc_curve(RAN,x_test,y_test)
mod = [log,DT,ADA]
for i in mod:
    plot_roc_curve(i,x_test,y_test, ax=disp.ax_)
plt.legend(prop={'size':10}, loc = 'lower right')
plt.show()
```



HyperParameter Tuning: GridSearchCV

In [81]:
```
,6],'min_samples_leaf':[2,5,10],'min_samples_split':[1,2,5],'criterion':['gini','entropy'],'max_features':["auto","sqrt","log2"]}
```

In [82]:
```
GridCV = GridSearchCV(RandomForestClassifier(),parameter,cv=5,n_jobs = -1,verbose = 1)
```

In [83]:
```
GridCV.fit(x_train,y_train)
```

```
Fitting 5 folds for each of 324 candidates, totalling 1620 fits
```

Out[83]:
```
GridSearchCV(cv=5, estimator=RandomForestClassifier(), n_jobs=-1,
             param_grid={'criterion': ['gini', 'entropy'], 'max_depth': [1, 6],
                         'max_features': ['auto', 'sqrt', 'log2'],
                         'min_samples_leaf': [2, 5, 10],
                         'min_samples_split': [1, 2, 5],
                         'n_estimators': [5, 10, 30]},
             verbose=1)
```

In [84]:
```
GridCV.best_params_
```

Out[84]:
```
{'criterion': 'gini',
 'max_depth': 6,
 'max_features': 'log2',
 'min_samples_leaf': 10,
 'min_samples_split': 5,
 'n_estimators': 5}
```

In [86]:
```
Best_mod2 = RandomForestClassifier(n_estimators = 30,criterion = 'entropy', max_depth= 6, max_features = 'sqrt',min_samples_leaf
Best_mod2.fit(x_train,y_train)
rfpred = Best_mod2.predict(x_test)
acc = accuracy_score(y_test,rfpred)
print(acc*100)
```

```
90.17347659306475
```

In [87]:
```
conf_matrx = confusion_matrix(y_test,rfpred)
conf_matrx
```

Out[87]:
```
array([[ 1622,  4948],
       [  201, 45628]], dtype=int64)
```

CONCLUSION

- Key Findings and Conclusions of the Study We can tell that target variable is im balanced and need to balance that and data loss is more actually and need to handle that as well as we can’t lose >8% of data. Dealing with huge dataset has taken a lot of time for running each algorithm and hyper parameter has taken more time to train the data and it was a nice experience that I have learnt so many things by worked on this project.