FLIP ROBO

Predicting Micro Credit Loan Defaulters

Submitted by:

Manoj Kumar Saxena

ACKNOWLEDGMENT

I would like to express my gratitude towards FlipRobo Technologies for their kind co-operation and encouragement which help me in completion of this project.

I would like to express my special gratitude and thanks to industry persons and my mentor (Ms. Shristi Maan) for giving me such attention and time as and whenever required.

INTRODUCTION

Business Problem Statement

- 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.

Conceptual Background of the Domain Problem

- Micro Credit Loan is a value-added service designed on the premise of “what the consumer needs”, **provides ease of access of airtime stock credit to customers of a telecom operator when they run out of balance**.

- 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.

- 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 MFI

- 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.

- 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.

- ## Motivation for the Problem Undertaken

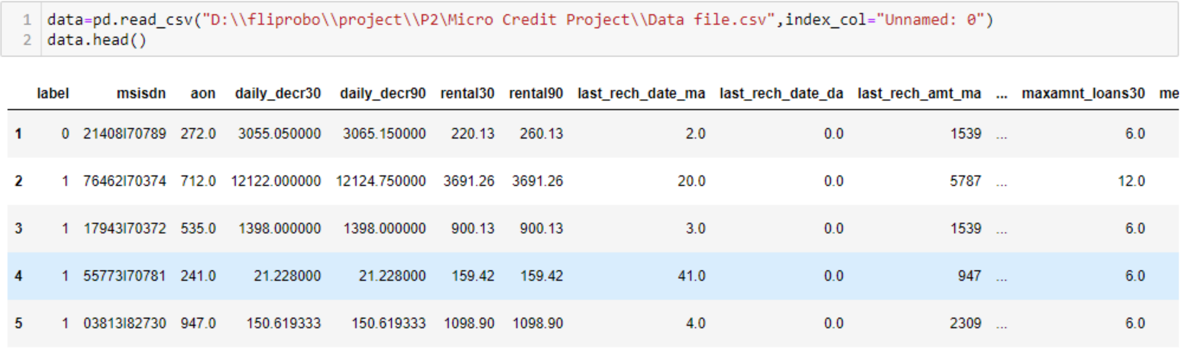
Micro Credit Loan is a value-added service designed on the premise of “what the consumer needs”, provides ease of access of airtime stock credit to customers of a telecom operator when they run out of balance.

Credit Loan is an interactive service that allows customers to take a sundry credit amount of any configurable denomination when they run out of their main account balance and are either far away from a recharging location or are short of money to immediately recharge their account. Micro Credit is an open and transparent service with a clear reporting structure.

I completely agree that it is a very effective way of offering funds to the economically underprivileged sections of the society.

Analytical Problem Framing

- ## Dataset Representation:



```
data=pd.read_csv("D:\\fliprobo\\project\\P2\Micro Credit Project\\Data file.csv",index_col="Unnamed: 0")
data.head()
```

	label	msisdn	aon	daily_decr30	daily_decr90	rental30	rental90	last_rech_date_ma	last_rech_date_da	last_rech_amt_ma	...	maxamnt_loans30	me
1	0	21408I70789	272.0	3055.050000	3065.150000	220.13	260.13	2.0	0.0	1539	...	6.0	
2	1	76462I70374	712.0	12122.000000	12124.750000	3691.26	3691.26	20.0	0.0	5787	...	12.0	
3	1	17943I70372	535.0	1398.000000	1398.000000	900.13	900.13	3.0	0.0	1539	...	6.0	
4	1	55773I70781	241.0	21.228000	21.228000	159.42	159.42	41.0	0.0	947	...	6.0	
5	1	03813I82730	947.0	150.619333	150.619333	1098.90	1098.90	4.0	0.0	2309	...	6.0	

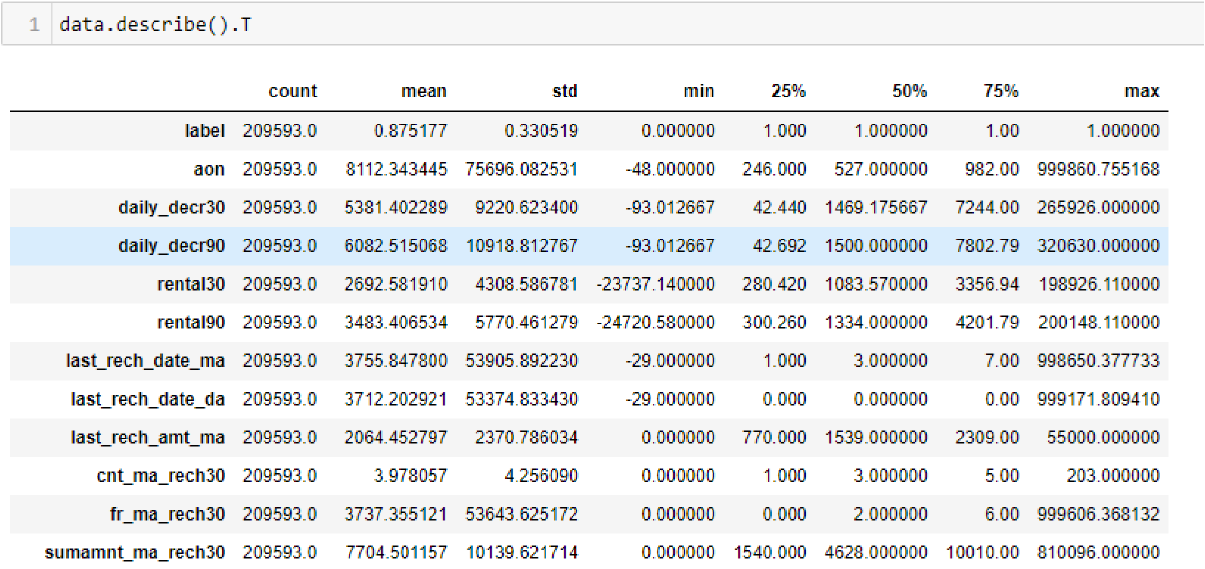
Observation:

Seeing the data we have to 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.

1. The data seems to be a combination of both numerical and categorical features.
2. Msisdn, pcircle, pdate are categorical and rest features are numerical in type.

So clearly it is a classification problem.

- **Statistical Data:**



```
data.describe().T
```

	count	mean	std	min	25%	50%	75%	max
label	209593.0	0.875177	0.330519	0.000000	1.000	1.000000	1.00	1.000000
aon	209593.0	8112.343445	75696.082531	-48.000000	246.000	527.000000	982.00	999860.755168
daily_decr30	209593.0	5381.402289	9220.623400	-93.012667	42.440	1469.175667	7244.00	265926.000000
daily_decr90	209593.0	6082.515068	10918.812767	-93.012667	42.692	1500.000000	7802.79	320630.000000
rental30	209593.0	2692.581910	4308.586781	-23737.140000	280.420	1083.570000	3356.94	198926.110000
rental90	209593.0	3483.406534	5770.461279	-24720.580000	300.260	1334.000000	4201.79	200148.110000
last_rech_date_ma	209593.0	3755.847800	53905.892230	-29.000000	1.000	3.000000	7.00	998650.377733
last_rech_date_da	209593.0	3712.202921	53374.833430	-29.000000	0.000	0.000000	0.00	999171.809410
last_rech_amt_ma	209593.0	2064.452797	2370.786034	0.000000	770.000	1539.000000	2309.00	55000.000000
cnt_ma_rech30	209593.0	3.978057	4.256090	0.000000	1.000	3.000000	5.00	203.000000
fr_ma_rech30	209593.0	3737.355121	53643.625172	0.000000	0.000	2.000000	6.00	999606.368132
sumamnt_ma_rech30	209593.0	7704.501157	10139.621714	0.000000	1540.000	4628.000000	10010.00	810096.000000

Observation:

1. There are some unnatural values in the dataset.
2. There are also some outliers in the dataset.
3. Label data is highly imbalanced.
4. We can also see some negative values in age column which is absolutely impossible.

Data Sources and their formats & inferences

- Mobile numbers are integers type having an alphabet which we have transformed using Label Encoder.
- Age or number of days are never negative which we have treated.
- Extremely large positive values are data entered in we have o remove such values that are greater than 18000.
- Amount spent cannot be negative hence that is been removed.
- Larger amounts in some features are may be because of the groups of loan taken .
- 5 and 10 Indonesian Rupiah are the only loan amount options given for the consumer.
- There could be records with 0 as the loan amount as well this is may be user doesn’t used their card much..
- Return amount can be 0,6 and 12 only.
- The customer will be a defaulter if they don't pay back the loan amount within 5 days of issuing the loan. So, the Average payback value will be less than or equal to 5

for records with label = 1 and Average payback value to be greater than 5 for records with label = 0.

-

- **Data Pre-processing Done**

- From the above description we see that almost our data is imbalanced, we have more counts of 1 than 0 in or target variable.

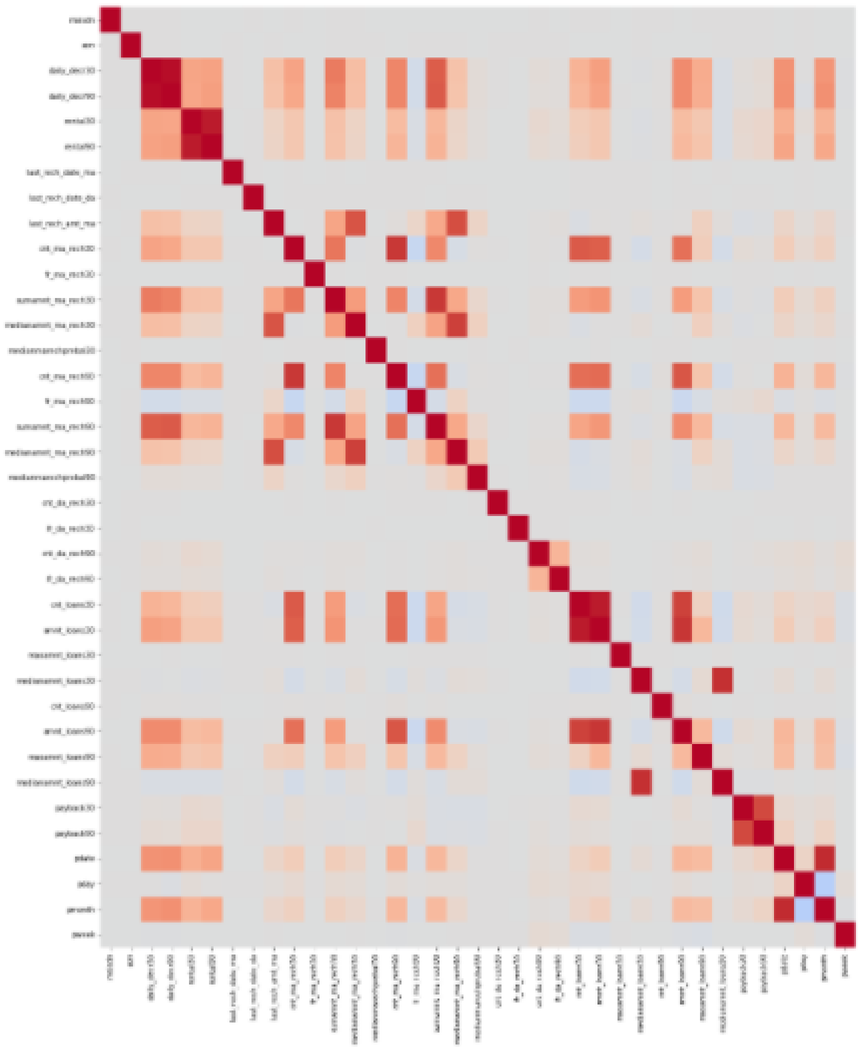
- Almost every independent variable is highly skewed to right and has outliers. aon shows negative value at minimum as age (in terms of days) cannot be negative, daily_decr30 and daily_decr90 also shows generative value at minimum that too needs to be removed.

- last_rech_date_ma and last_rech_date_da - minimum value also here should be removed.

- I will 1st convert aon, daily_decr30, daily_decr90, last_rech_date_ma and last_rech_date_da into absolute values to remove the negative value.

- But some of the positively skewed values are not treated as the limitation are not a constrain and not even mentioned, hence can left unhandled.

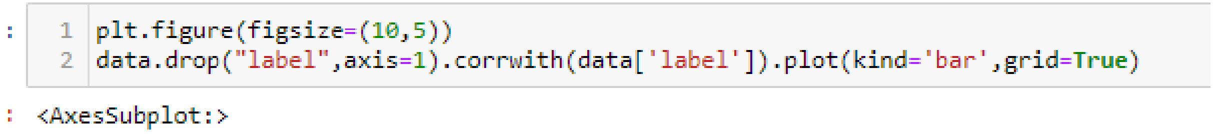
- **Data Inputs- Logic- Output Relationships**
- **Correlation:**



Observation:

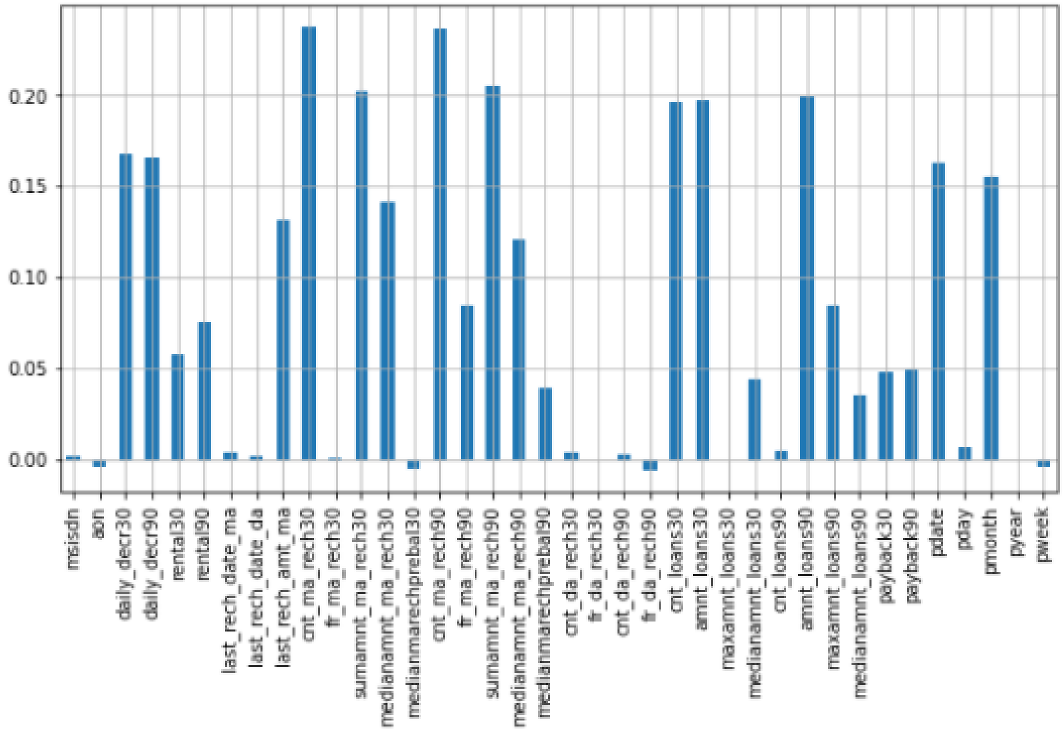
- daily_decr30----daily_decr90
- rental30------rantal90
- cnt_ma_rech30------cnt_ma_rech90
- sumamnt_ma_rech30-----sumamnt_ma_rech90
- cnt_loans30----amnt_loans30
- cnt_loans30---amnt_loans90
- amnt_loans30---amnt_loans90
- medianamnt_loans30-----medianamnt_loans90
- pdate---pmonth

Above these are features are co-linear with 90+% collinearity.



```
plt.figure(figsize=(10,5))
data.drop("label",axis=1).corrwith(data['label']).plot(kind='bar',grid=True)
```

```
<AxesSubplot:>
```



Observations:

- daily_decr30,
- last_rech_amt_ma,
- cnt_ma_rech30,
- sumamny_ma_rech30,
- medianamnt_ma_rech30,
- cnt_ma_rech90,
- sumamnt_ma_rech90,
- cnt_loans30,
- amnt_loan30,
- amnt_loan90,
- pdate and month

  Through these features labels prediction chances are high.

- ## Assumption for the problem:

➢ So clearly it is a classification problem. We will be using both simple algorithms, tree algorithms and ensemble algorithm to solve our problem.

- ## Hardware and Software Requirements and Tools Used

Software Used:

- Jupyter Notebook
- Ms-Paint
- MS-PowerPoint
- MS-Word

Hardware used:

- Laptop
- Good internet connectivity

Model/s Development and Evaluation

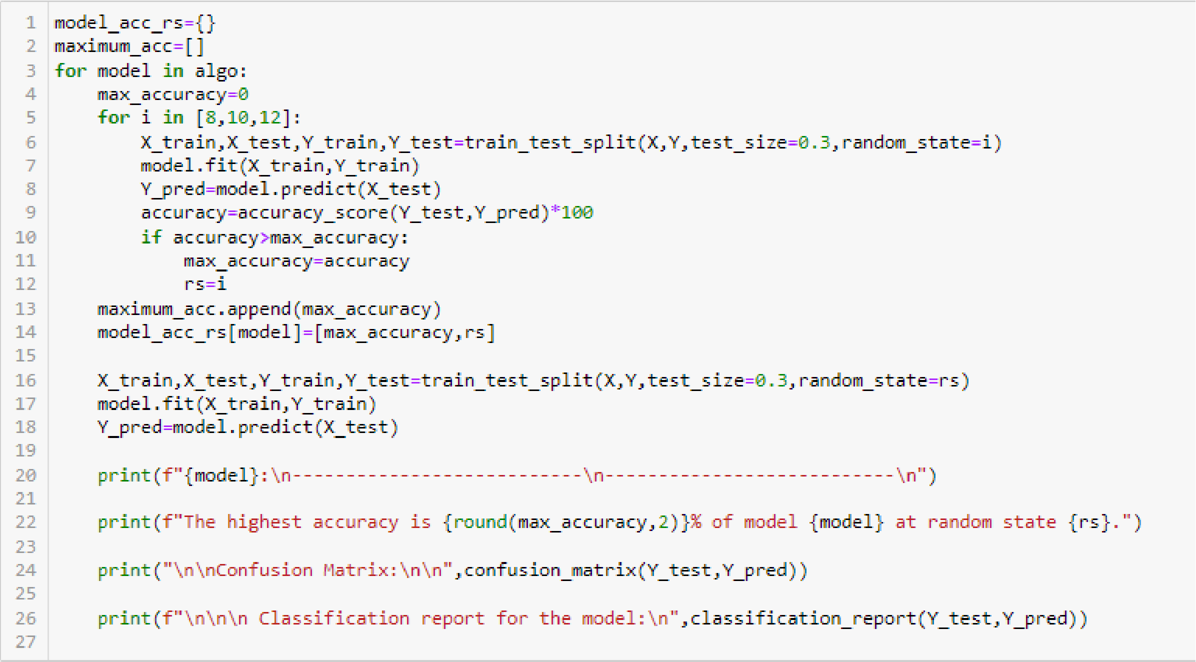
- Identification of possible problem-solving approaches (methods)
  a. Used Z-score techniques to remove outliers.
  b. Used Power Transformer with yeo-johnson method to reduce skewness.
  c. SelectKBest for feature selection where we selected top 24 features.
  d. Standardized the features using StandardScalar().
  e. Balanced the data set with upsampling 75%.

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

- LogisticRegression()
- DecisionTreeClassifier()
- RandomForestClassifier()
- AdaBoostClassifier()
- BaggingClassifier()

- Run and Evaluate selected models

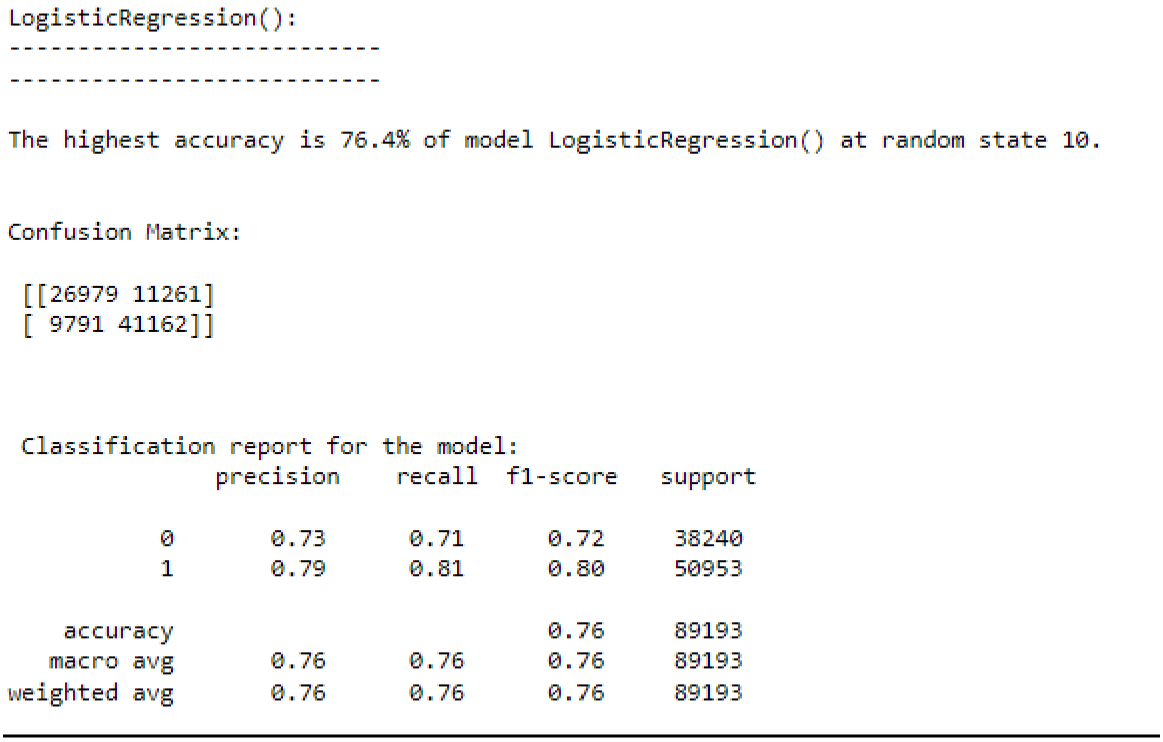
Code:



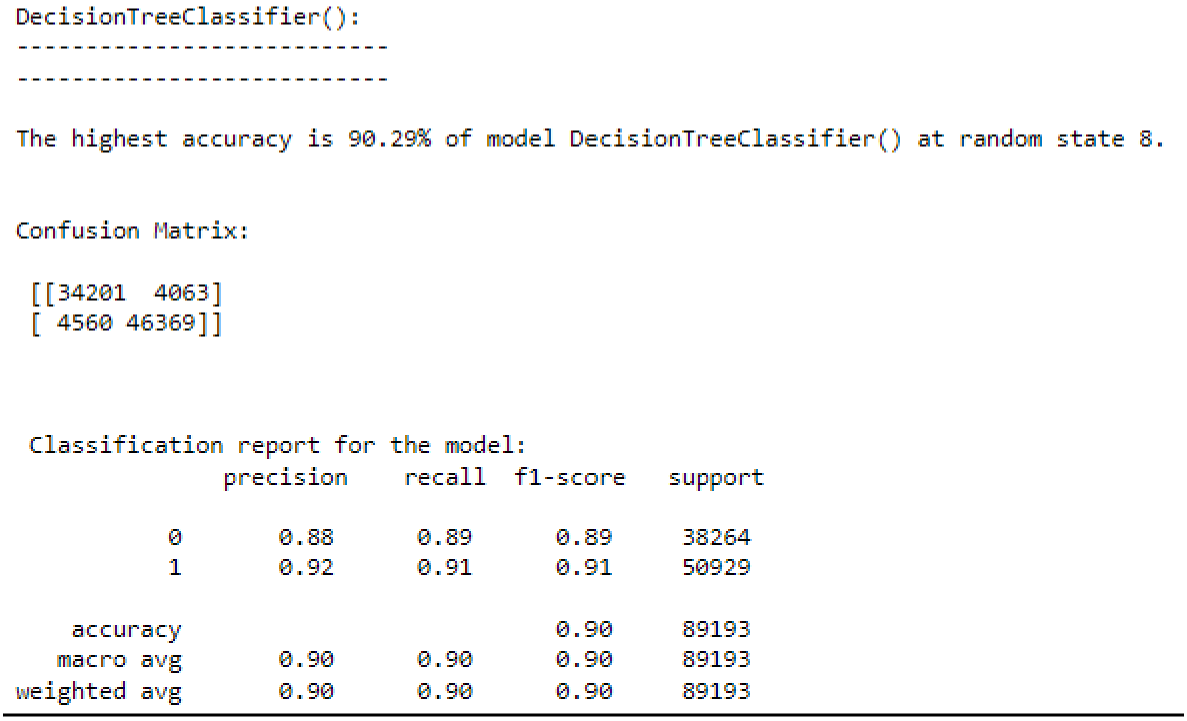
```python
model_acc_rs={}
maximum_acc=[]
for model in algo:
    max_accuracy=0
    for i in [8,10,12]:
        X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.3,random_state=i)
        model.fit(X_train,Y_train)
        Y_pred=model.predict(X_test)
        accuracy=accuracy_score(Y_test,Y_pred)*100
        if accuracy>max_accuracy:
            max_accuracy=accuracy
            rs=i
    maximum_acc.append(max_accuracy)
    model_acc_rs[model]=[max_accuracy,rs]

    X_train,X_test,Y_train,Y_test=train_test_split(X,Y,test_size=0.3,random_state=rs)
    model.fit(X_train,Y_train)
    Y_pred=model.predict(X_test)

    print(f"{model}:\n---------------------------\n---------------------------\n")

    print(f"The highest accuracy is {round(max_accuracy,2)}% of model {model} at random state {rs}.")

    print("\n\nConfusion Matrix:\n\n",confusion_matrix(Y_test,Y_pred))

    print(f"\n\n\n Classification report for the model:\n",classification_report(Y_test,Y_pred))

```

The above code shows the highest accuracy of each model at their best random state with the evaluation metrics like Confusion metrics and classification report that shows the precision, recall and F1-score for each model.

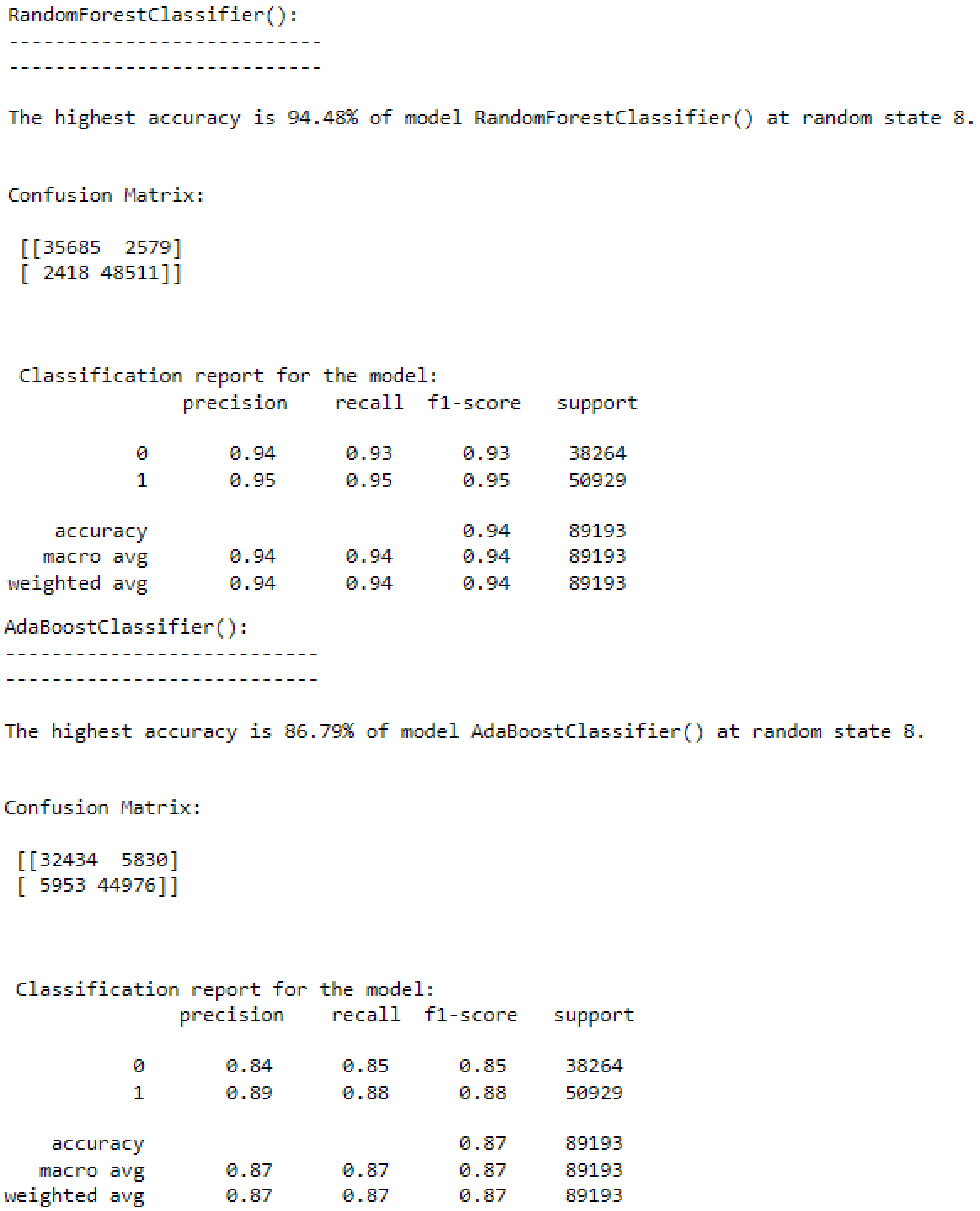
Output:



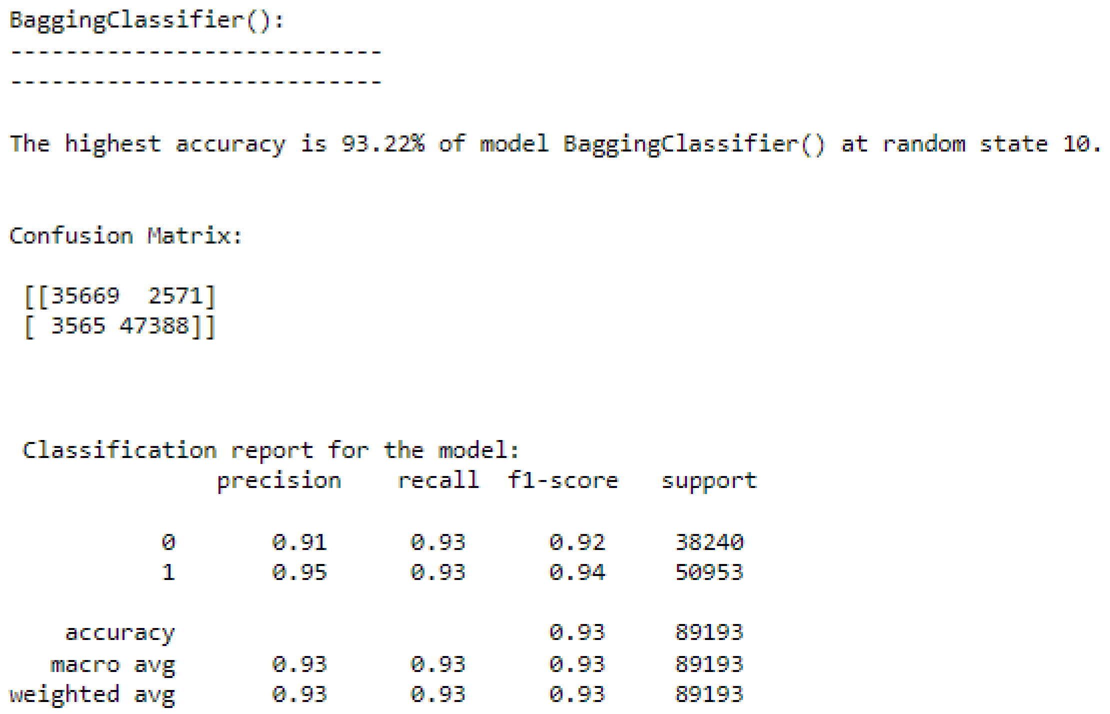
```
LogisticRegression():
---------------------------
---------------------------

The highest accuracy is 76.4% of model LogisticRegression() at random state 10.


Confusion Matrix:

 [[26979 11261]
 [ 9791 41162]]



 Classification report for the model:
               precision    recall  f1-score   support

           0       0.73      0.71      0.72     38240
           1       0.79      0.81      0.80     50953

    accuracy                           0.76     89193
   macro avg       0.76      0.76      0.76     89193
weighted avg       0.76      0.76      0.76     89193
```



```
DecisionTreeClassifier():
---------------------------
---------------------------

The highest accuracy is 90.29% of model DecisionTreeClassifier() at random state 8.


Confusion Matrix:

 [[34201  4063]
 [ 4560 46369]]



 Classification report for the model:
               precision    recall  f1-score   support

           0       0.88      0.89      0.89     38264
           1       0.92      0.91      0.91     50929

    accuracy                           0.90     89193
   macro avg       0.90      0.90      0.90     89193
weighted avg       0.90      0.90      0.90     89193
```



```
RandomForestClassifier():
---------------------------
---------------------------

The highest accuracy is 94.48% of model RandomForestClassifier() at random state 8.


Confusion Matrix:

 [[35685  2579]
 [ 2418 48511]]



 Classification report for the model:
               precision    recall  f1-score   support

           0       0.94      0.93      0.93     38264
           1       0.95      0.95      0.95     50929

    accuracy                           0.94     89193
   macro avg       0.94      0.94      0.94     89193
weighted avg       0.94      0.94      0.94     89193
AdaBoostClassifier():
---------------------------
---------------------------

The highest accuracy is 86.79% of model AdaBoostClassifier() at random state 8.


Confusion Matrix:

 [[32434  5830]
 [ 5953 44976]]



 Classification report for the model:
               precision    recall  f1-score   support

           0       0.84      0.85      0.85     38264
           1       0.89      0.88      0.88     50929

    accuracy                           0.87     89193
   macro avg       0.87      0.87      0.87     89193
weighted avg       0.87      0.87      0.87     89193
```

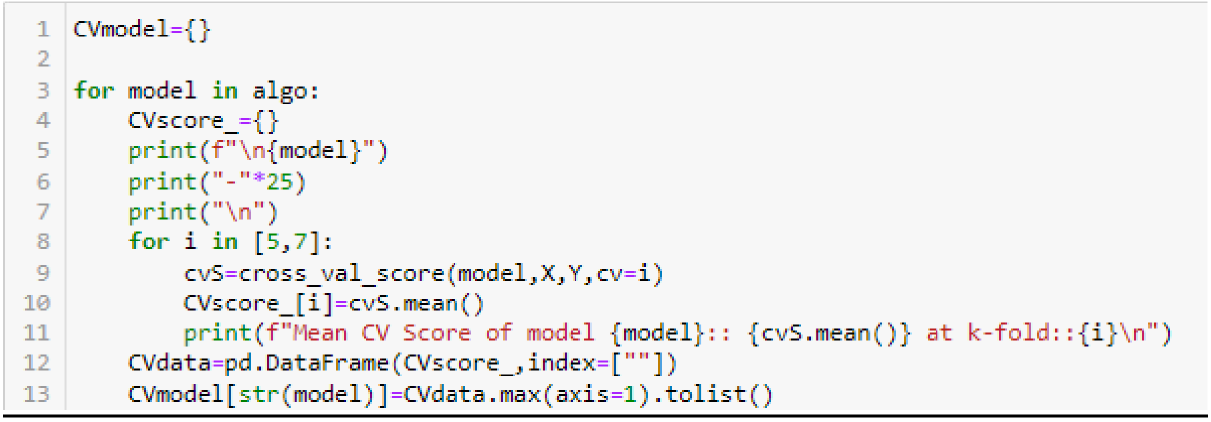


```
BaggingClassifier():
---------------------------
---------------------------

The highest accuracy is 93.22% of model BaggingClassifier() at random state 10.


Confusion Matrix:

 [[35669  2571]
 [ 3565 47388]]



 Classification report for the model:
               precision    recall  f1-score   support

           0       0.91      0.93      0.92     38240
           1       0.95      0.93      0.94     50953

    accuracy                           0.93     89193
   macro avg       0.93      0.93      0.93     89193
weighted avg       0.93      0.93      0.93     89193
```

- **<u>Key Metrics for success in solving problem under consideration</u>**

  Here we used CV Score for all models.

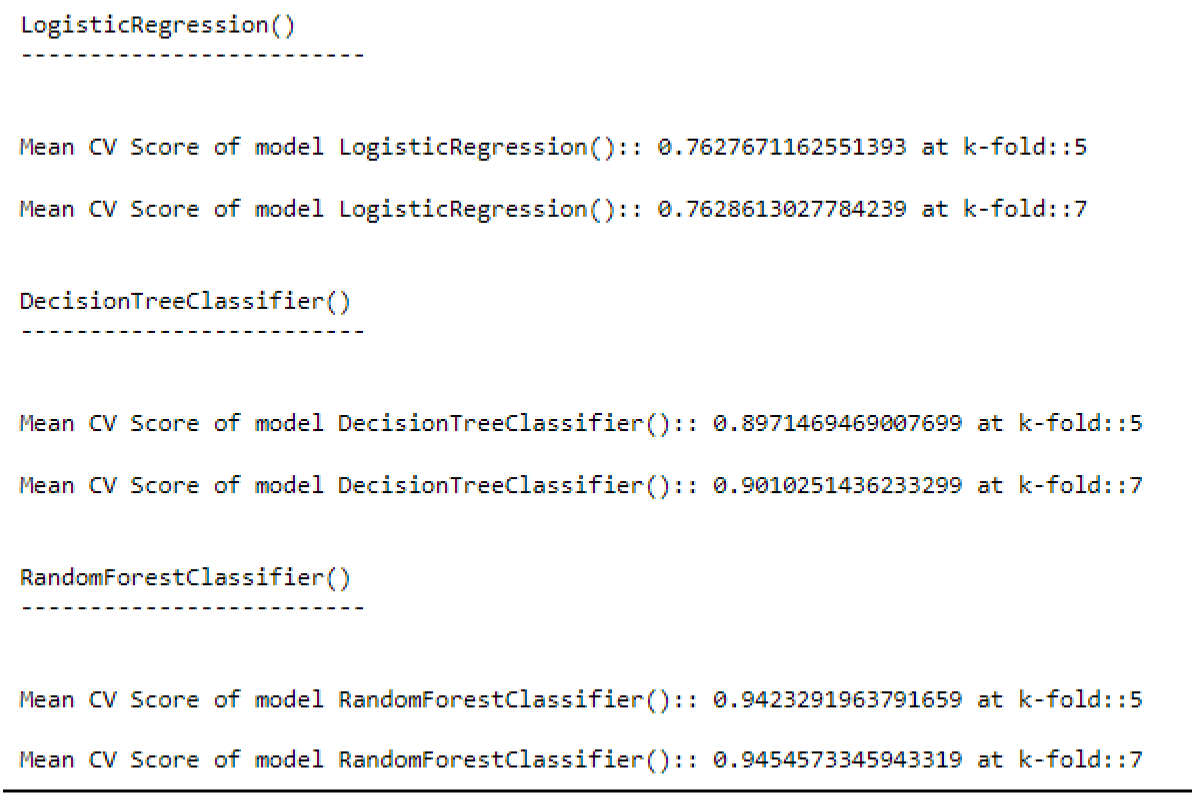
  <u>Code:</u>



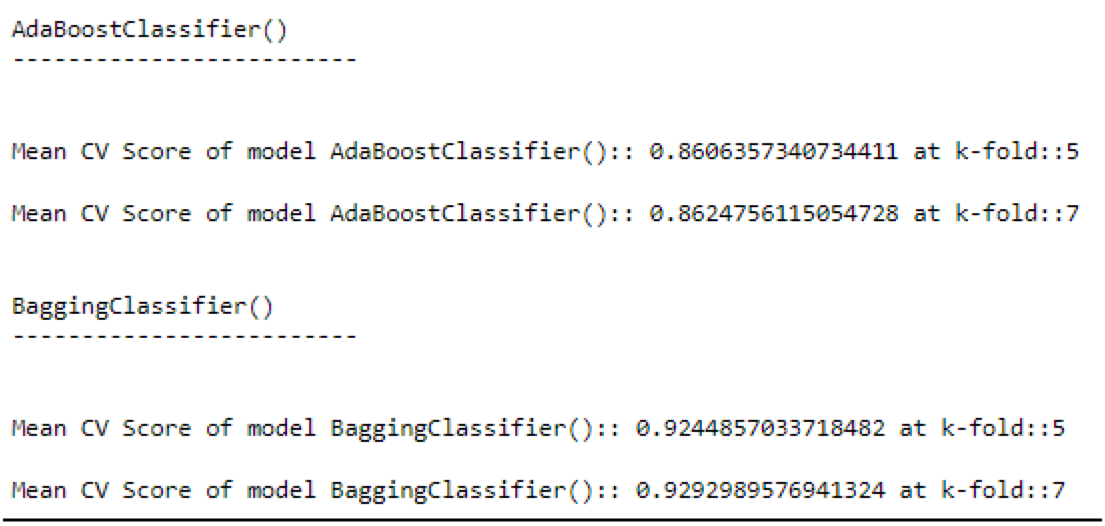
```python
CVmodel={}

for model in algo:
    CVscore_={}
    print(f"\n{model}")
    print("-"*25)
    print("\n")
    for i in [5,7]:
        cvS=cross_val_score(model,X,Y,cv=i)
        CVscore_[i]=cvS.mean()
        print(f"Mean CV Score of model {model}:: {cvS.mean()} at k-fold::{i}\n")
    CVdata=pd.DataFrame(CVscore_,index=[""])
    CVmodel[str(model)]=CVdata.max(axis=1).tolist()
```

  <u>Output:</u>

  Above code shows the CV score for all model that we used with k-fold 5& 7.

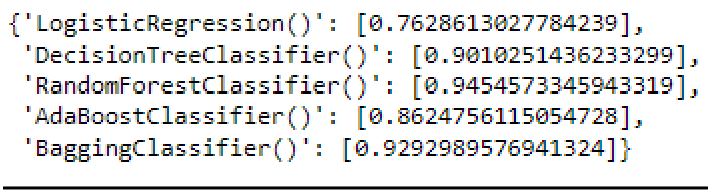


```
LogisticRegression()
-------------------------


Mean CV Score of model LogisticRegression():: 0.7627671162551393 at k-fold::5

Mean CV Score of model LogisticRegression():: 0.7628613027784239 at k-fold::7


DecisionTreeClassifier()
-------------------------


Mean CV Score of model DecisionTreeClassifier():: 0.8971469469007699 at k-fold::5

Mean CV Score of model DecisionTreeClassifier():: 0.9010251436233299 at k-fold::7


RandomForestClassifier()
-------------------------


Mean CV Score of model RandomForestClassifier():: 0.9423291963791659 at k-fold::5

Mean CV Score of model RandomForestClassifier():: 0.9454573345943319 at k-fold::7
```



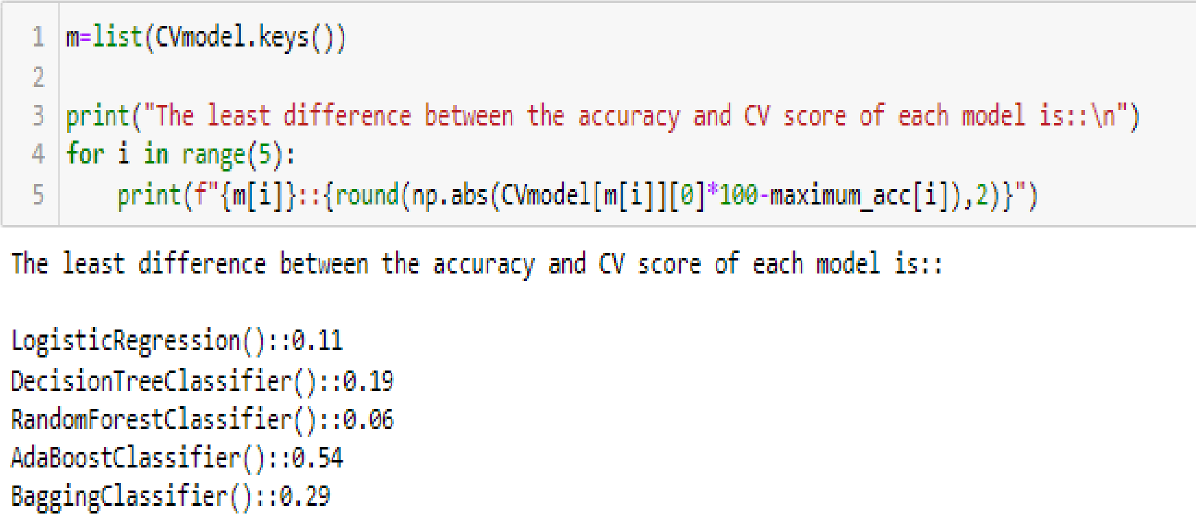
```
AdaBoostClassifier()
-------------------------


Mean CV Score of model AdaBoostClassifier():: 0.8606357340734411 at k-fold::5

Mean CV Score of model AdaBoostClassifier():: 0.8624756115054728 at k-fold::7


BaggingClassifier()
-------------------------


Mean CV Score of model BaggingClassifier():: 0.9244857033718482 at k-fold::5

Mean CV Score of model BaggingClassifier():: 0.9292989576941324 at k-fold::7
```

The below snapshot shows the highest CV Score of each model.



```
{'LogisticRegression()': [0.7628613027784239],
 'DecisionTreeClassifier()': [0.9010251436233299],
 'RandomForestClassifier()': [0.9454573345943319],
 'AdaBoostClassifier()': [0.8624756115054728],
 'BaggingClassifier()': [0.9292989576941324]}
```

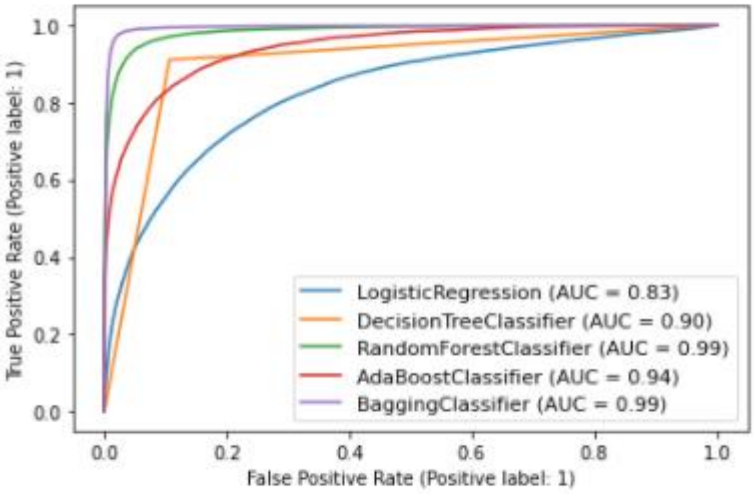
The below snapshot shows the Code for the least difference between the highest **accuracy** of each **model** with their best random state and highest **CV score.**



```
m=list(CVmodel.keys())

print("The least difference between the accuracy and CV score of each model is::\n")
for i in range(5):
    print(f"{m[i]}::{round(np.abs(CVmodel[m[i]][0]*100-maximum_acc[i]),2)}")
```

```
The least difference between the accuracy and CV score of each model is::

LogisticRegression()::0.11
DecisionTreeClassifier()::0.19
RandomForestClassifier()::0.06
AdaBoostClassifier()::0.54
BaggingClassifier()::0.29
```

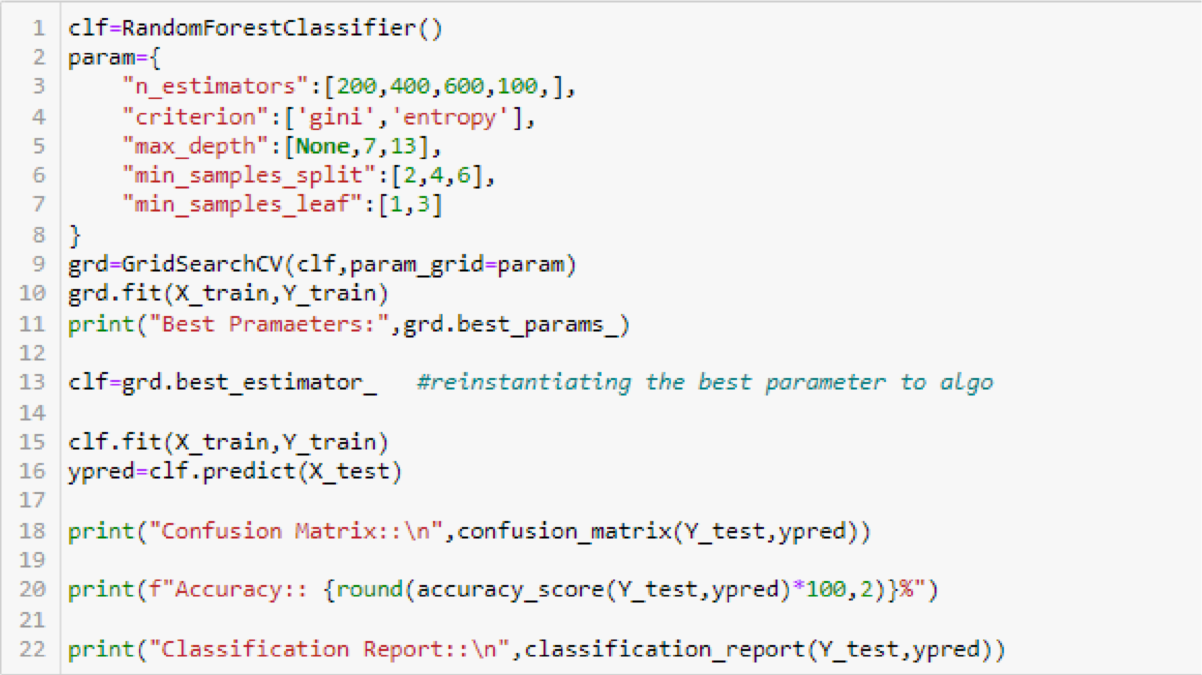
From above output we get Random Forest model with the least difference i.e., to be the finalized model.



Here we can see the bagging classifier and **Random Forest** has same AUC score. But using Random Forest helps to get the prediction faster than Bagging.

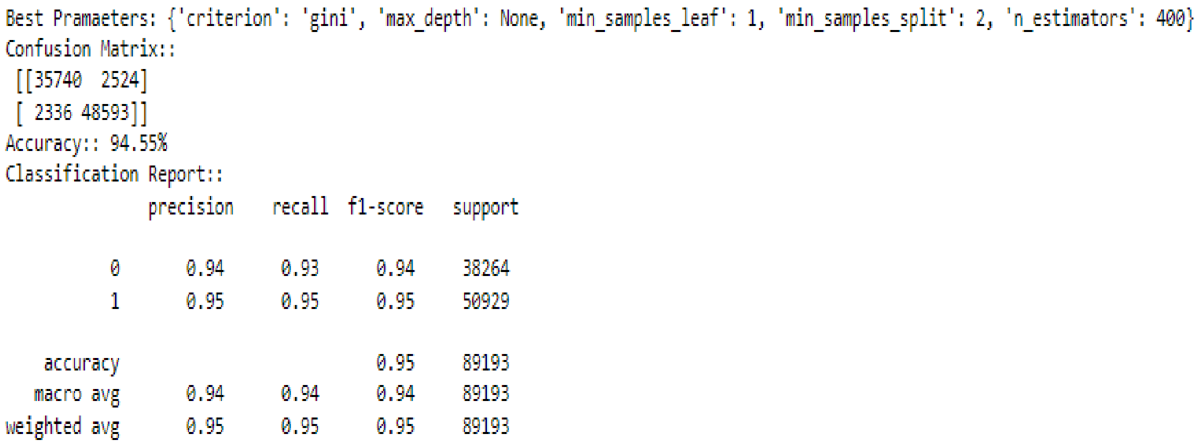
Hyper Parameter tuning

Code:



```
clf=RandomForestClassifier()
param={
    "n_estimators":[200,400,600,100,],
    "criterion":['gini','entropy'],
    "max_depth":[None,7,13],
    "min_samples_split":[2,4,6],
    "min_samples_leaf":[1,3]
}
grd=GridSearchCV(clf,param_grid=param)
grd.fit(X_train,Y_train)
print("Best Pramaeters:",grd.best_params_)

clf=grd.best_estimator_   #reinstantiating the best parameter to algo

clf.fit(X_train,Y_train)
ypred=clf.predict(X_test)

print("Confusion Matrix::\n",confusion_matrix(Y_test,ypred))

print(f"Accuracy:: {round(accuracy_score(Y_test,ypred)*100,2)}%")

print("Classification Report::\n",classification_report(Y_test,ypred))
```

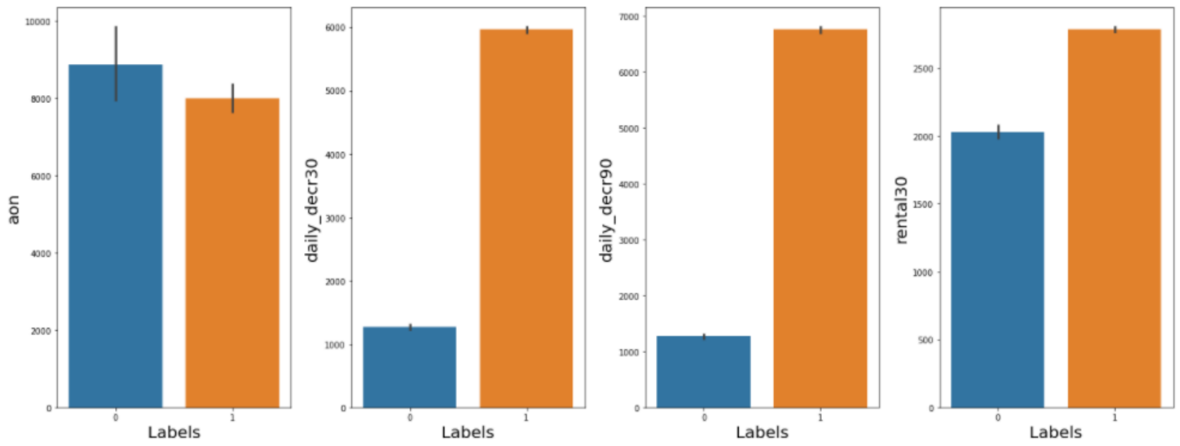
Output:



```
Best Pramaeters: {'criterion': 'gini', 'max_depth': None, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 400}
Confusion Matrix::
 [[35740  2524]
 [ 2336 48593]]
Accuracy:: 94.55%
Classification Report::
               precision    recall  f1-score   support

           0       0.94      0.93      0.94     38264
           1       0.95      0.95      0.95     50929

    accuracy                           0.95     89193
   macro avg       0.94      0.94      0.94     89193
weighted avg       0.95      0.95      0.95     89193
```

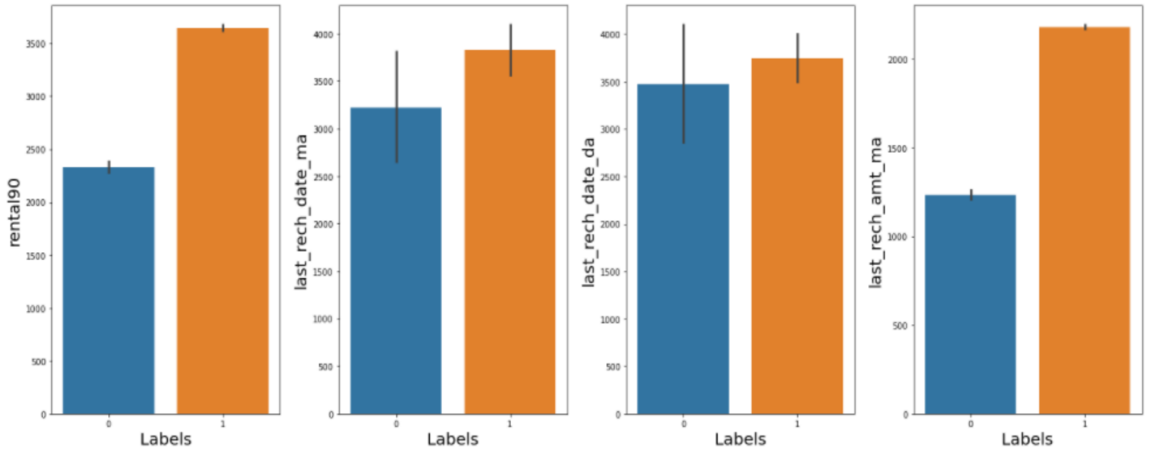
After Hyper tuned the finalized model we have slight increase in the accuracy.

- Visualizations



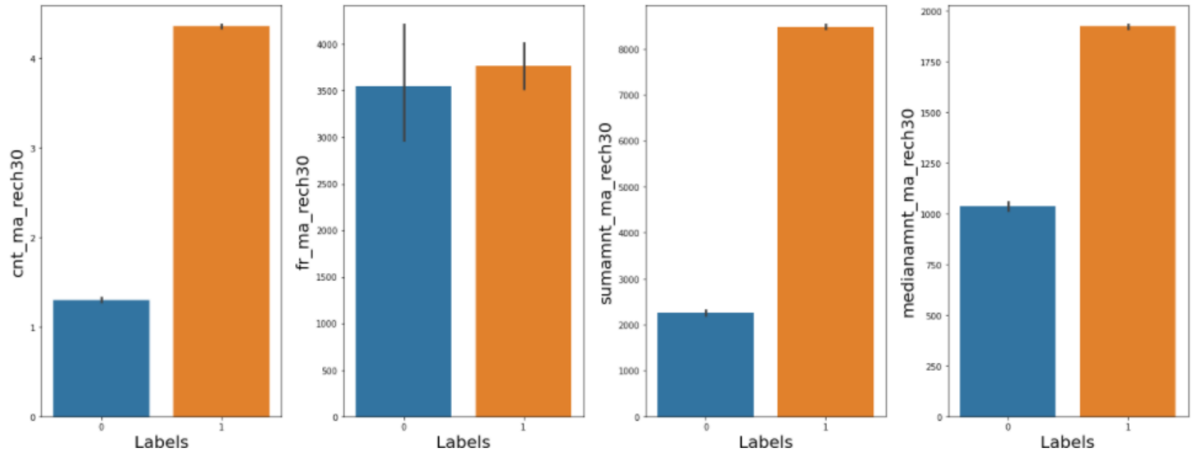
Observations:

- Maximum Users who paid bill having aon values has maximum 8000 age on its cellular networks.
- Daily amount spent from main account, averaged over last 30/90 days (in Indonesian Rupiah) who are maximum defaulter are in range of 1000 to 1200.
- Average main account balance over last 30 days is having maximum 2000 are the maximum no. of defaulters.



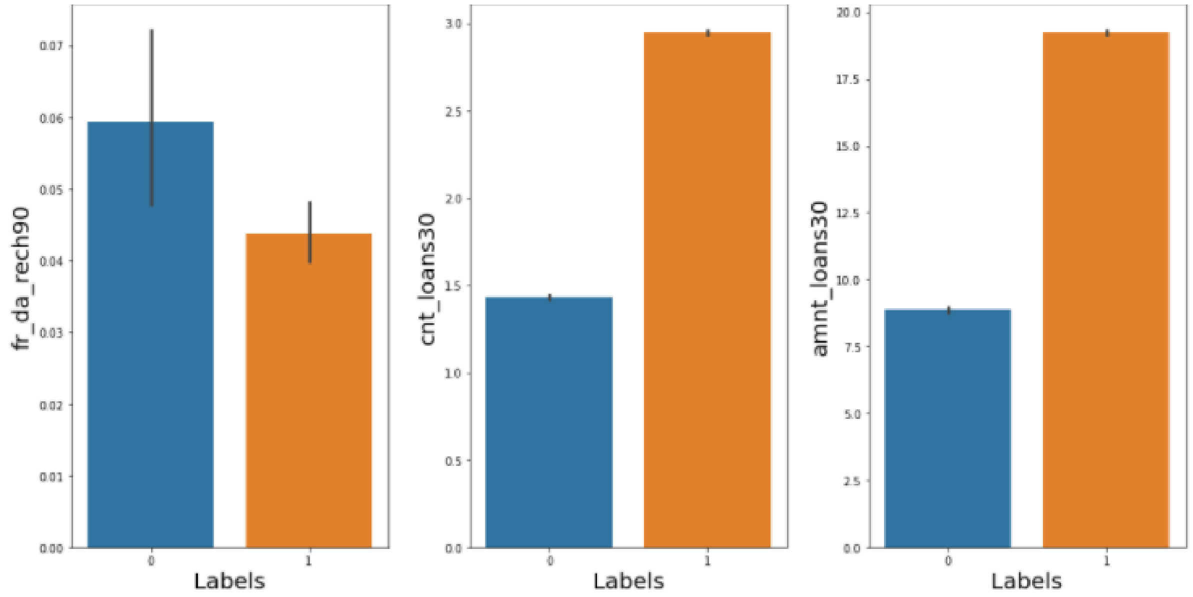
Observations:

- Over 90 days the maximum defaulter having maximum main account balance in range of 2000-2500.
- Number of days till last recharge of main account for defaulter having max in range off 3700 to 3800 while for data account we didn't get much info we plot more graphs for data account.
- Defaulter has done maximum last recharge for main account (in Indonesian Rupiah) is in range of 1200-1300 while other users done more than 2500.



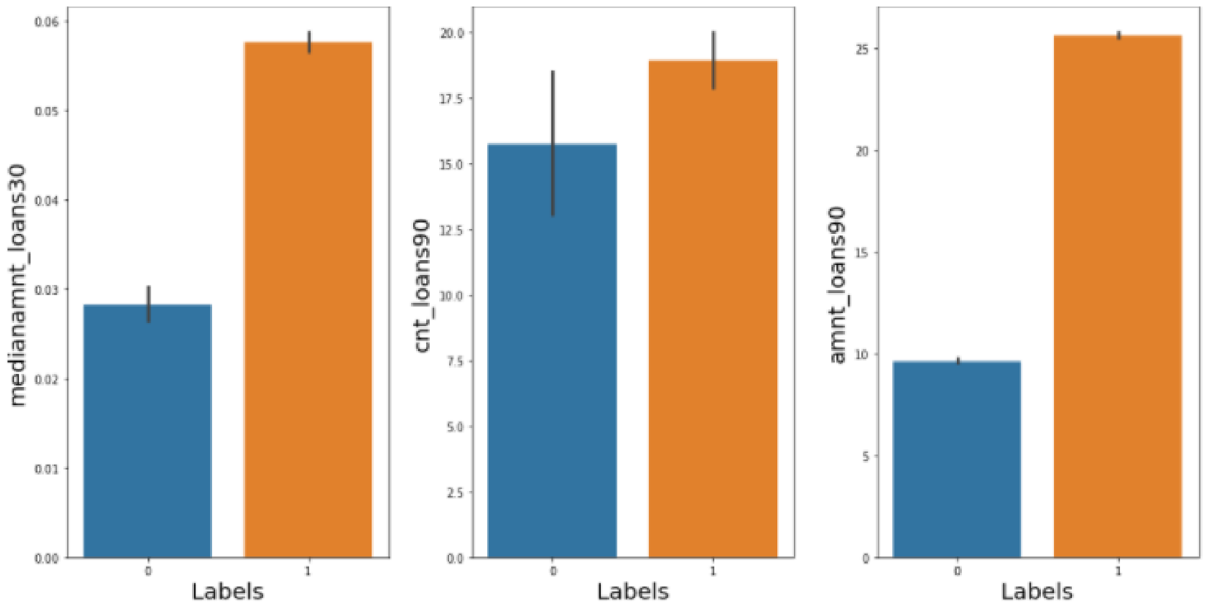
Observations:

- Defaulter has recharge maximum 1-2 times while others user recharges their credit for more than 4 times in a month.
- Defaulters has total amount of recharge in main account over last 30 days maximum in range 2200 -2300 while other users recharged their main account with summation more than 8000.
- Defaulters has median amount of recharge in main account over last 30 days maximum in range 100 -1050 while other users recharged their main account with summation more than 1900.
- Median of main account balance just before recharge in last 30 days at user level (in Indonesian Rupiah) for defaulter maximum was more than 5000 while others having approximately 4000.

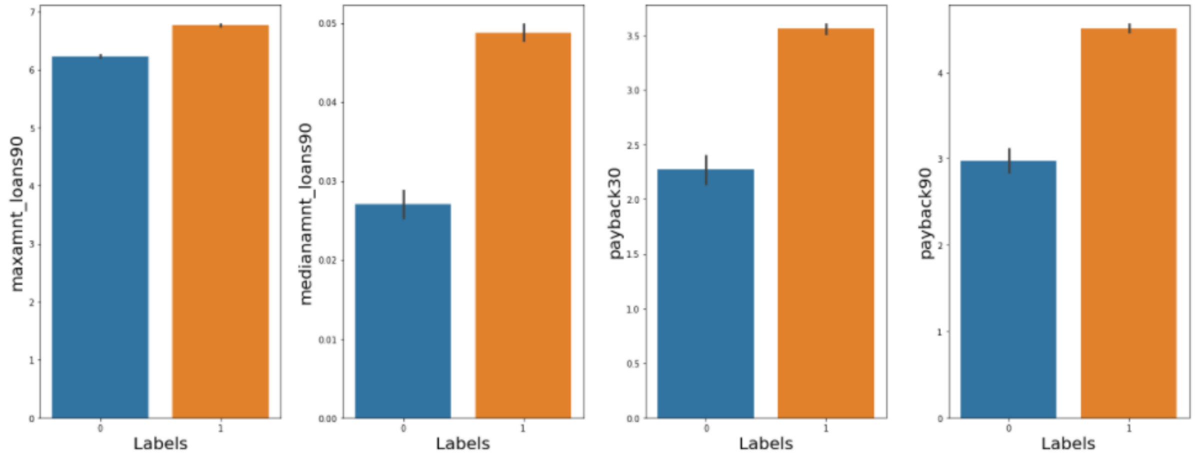


Observations:

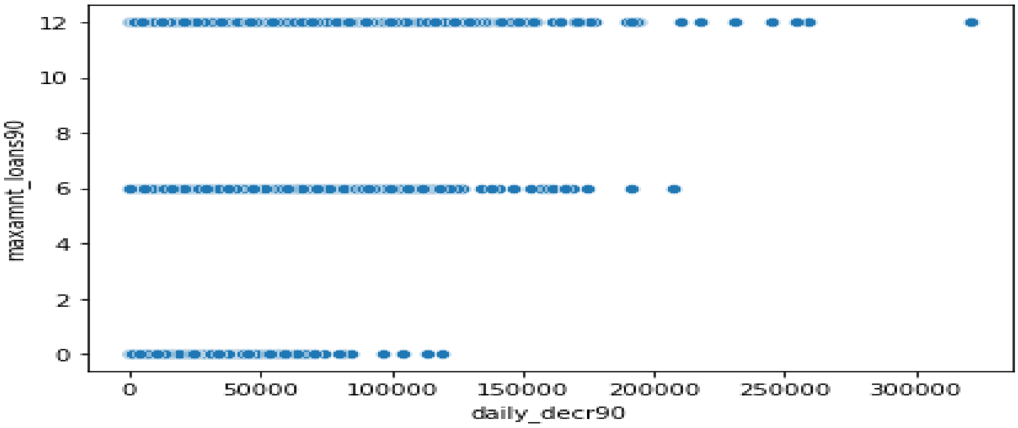
- Frequency for data recharge over 90 days for defaulter is more than 0.07 while other users has approximately 0.05
- Number of loans taken by defaulters in last 30 days was approximately 1.5 while other users has approximately 3.0
- Total amount of loans taken by defaulter in last 30 days was approx. 9 while other users has approx. 20.



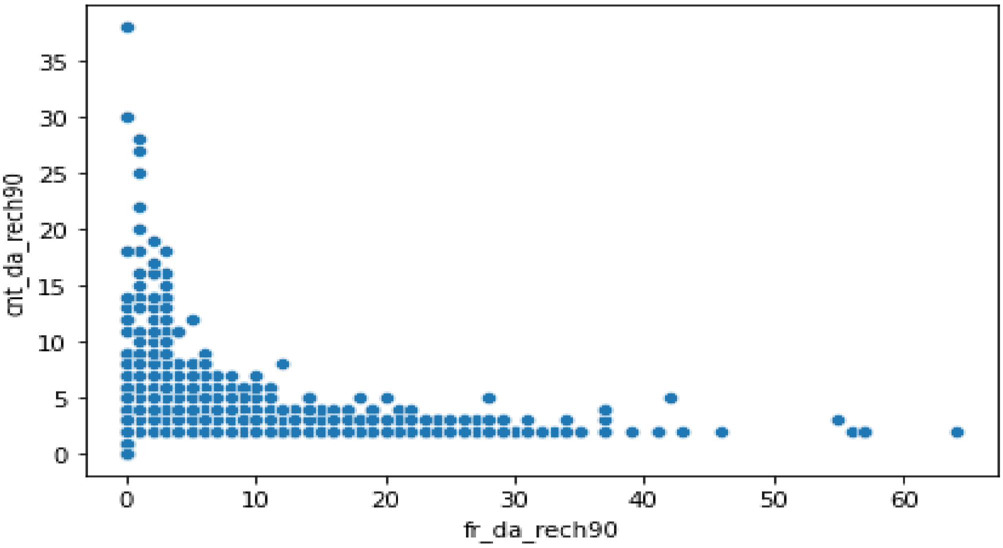
- Median of amounts of loan taken by the defaulters in last 30 days was approx. 0.03 while other users has approx. 0.06.
- Number of loans taken by defaulters in last 30 days has max approx. 17.5 while other users have max of approx. 20.0.
- Max total amount of loans taken by defaulters in last 30 days is of 10 while other users has taken more than 25.



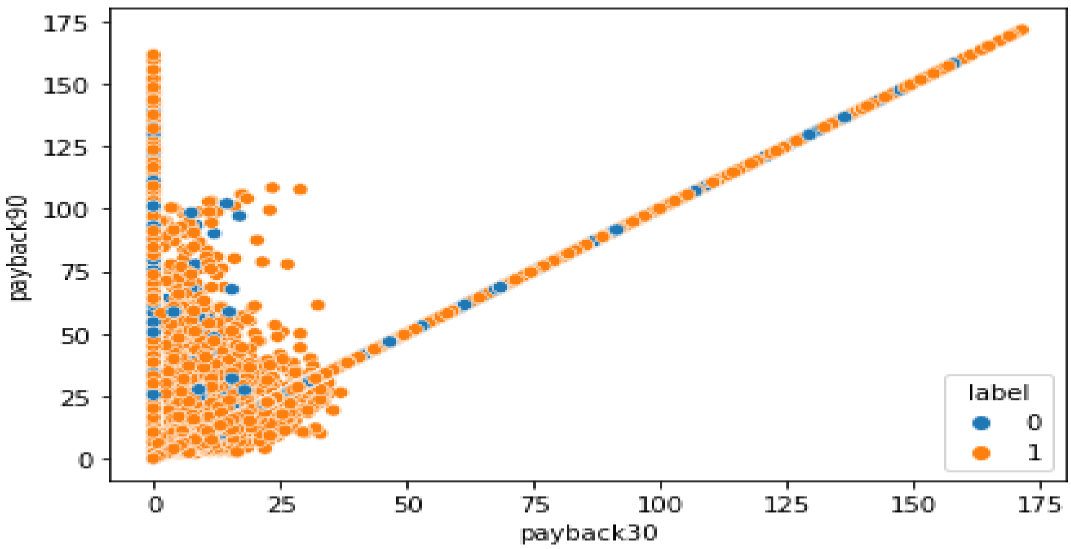
- Max amount of loan taken by the defaulters is approx. 6 while others has taken approx. 6.6. over 90 days
- Max median of amount taken by the defaulters is approx. 0.03 while other users has taken approx. 0.05 over 90 days.
- payback by the defaulter is approx. 2.5 while other user's payback more than 3 over 30 days
- payback by the defaulter is approx. 3 while other users payback more than 4 over 90 days



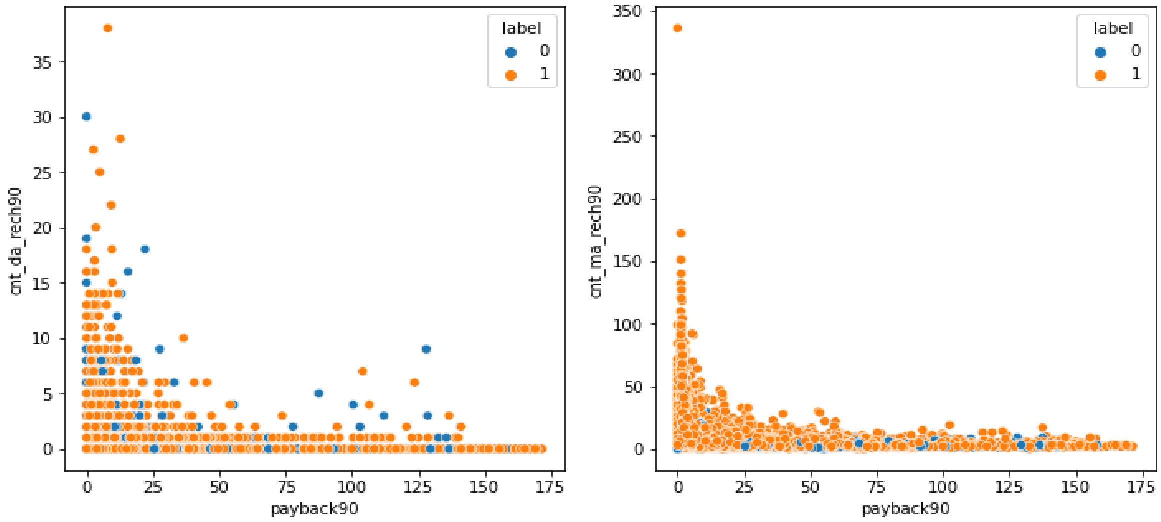
- Here we can see that somehow the max_amnt_loan_over 90 days and daily amount spent over 90 are linear dependent.



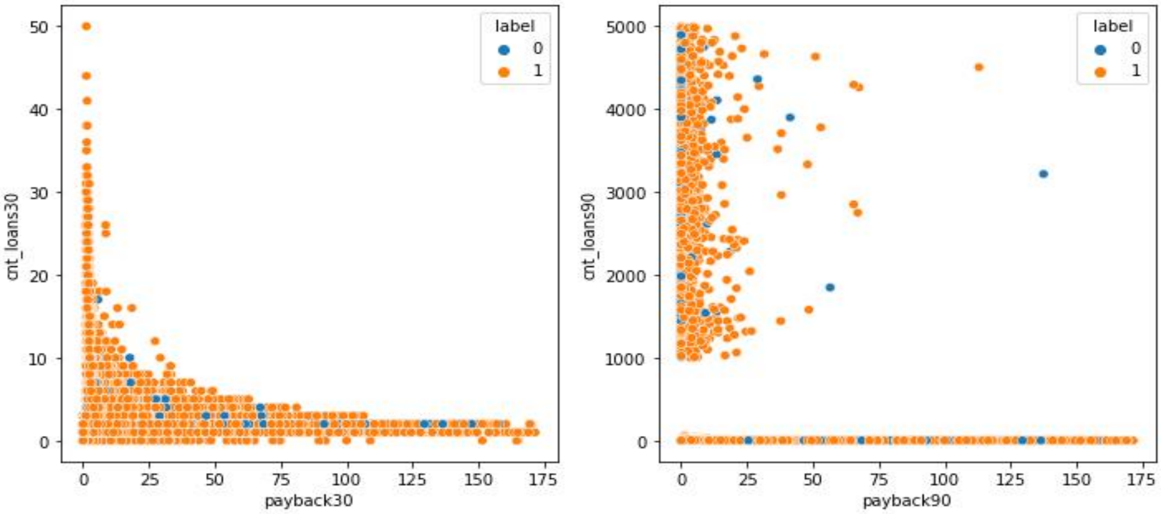
- Here we can see that Count of data recharge over 90 days increases w.r.t. decrease in frequency of data recharge over 90 days.



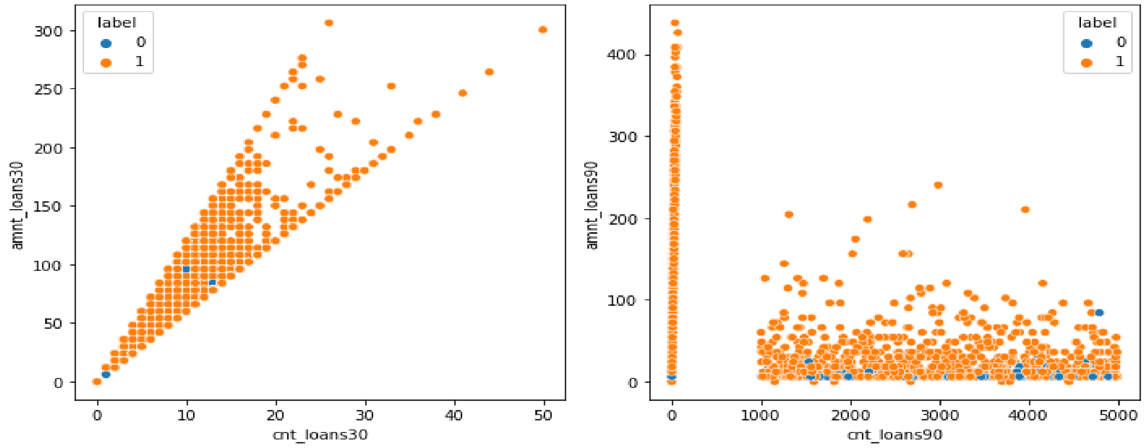
- From average payback over 30 days is co-linear with the payback90.



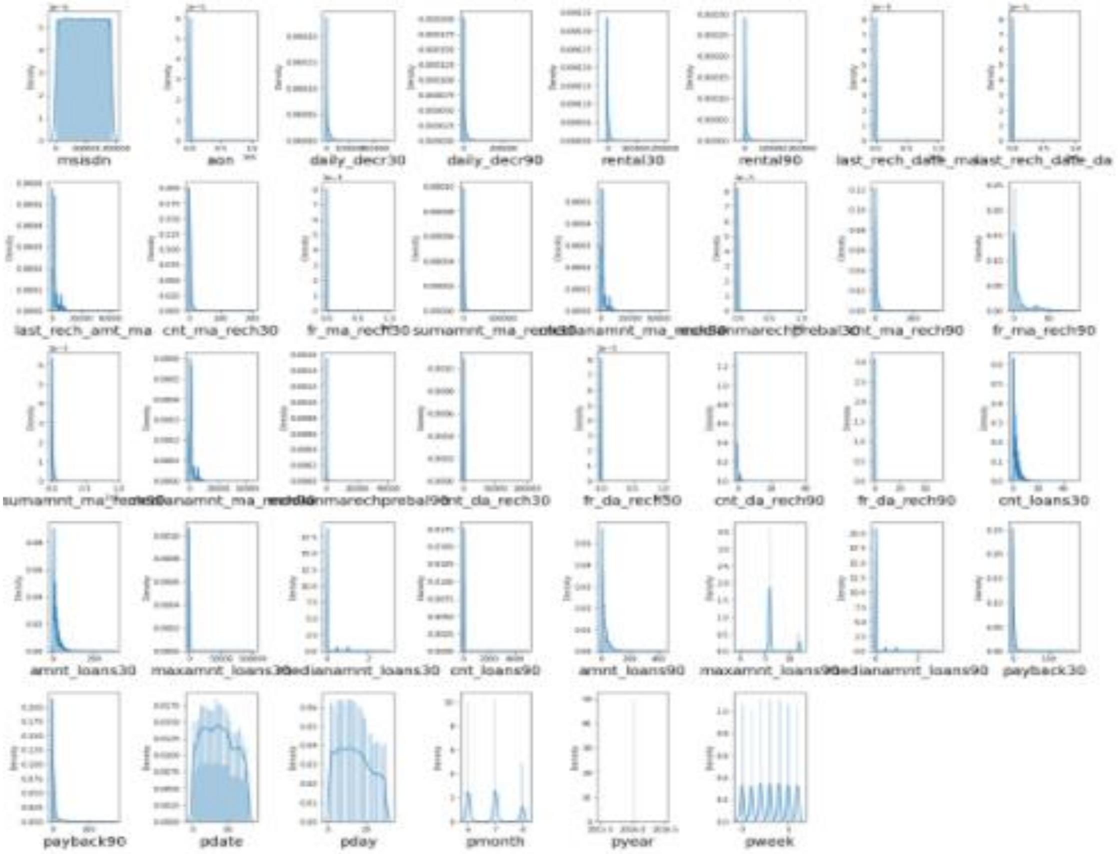
- There seems some relation that the user pay backs their loans who had recharge less no. of time for their data account as well as their main account too.



- Over 30 days the user highly pay back their loan who has taken less no. of loan.
- But over 90 days the user who has taken greater than 1000 times the loans has very rarely paybacks their loans this customer has benefits with the credit card but there seems to be defaulter too.
-

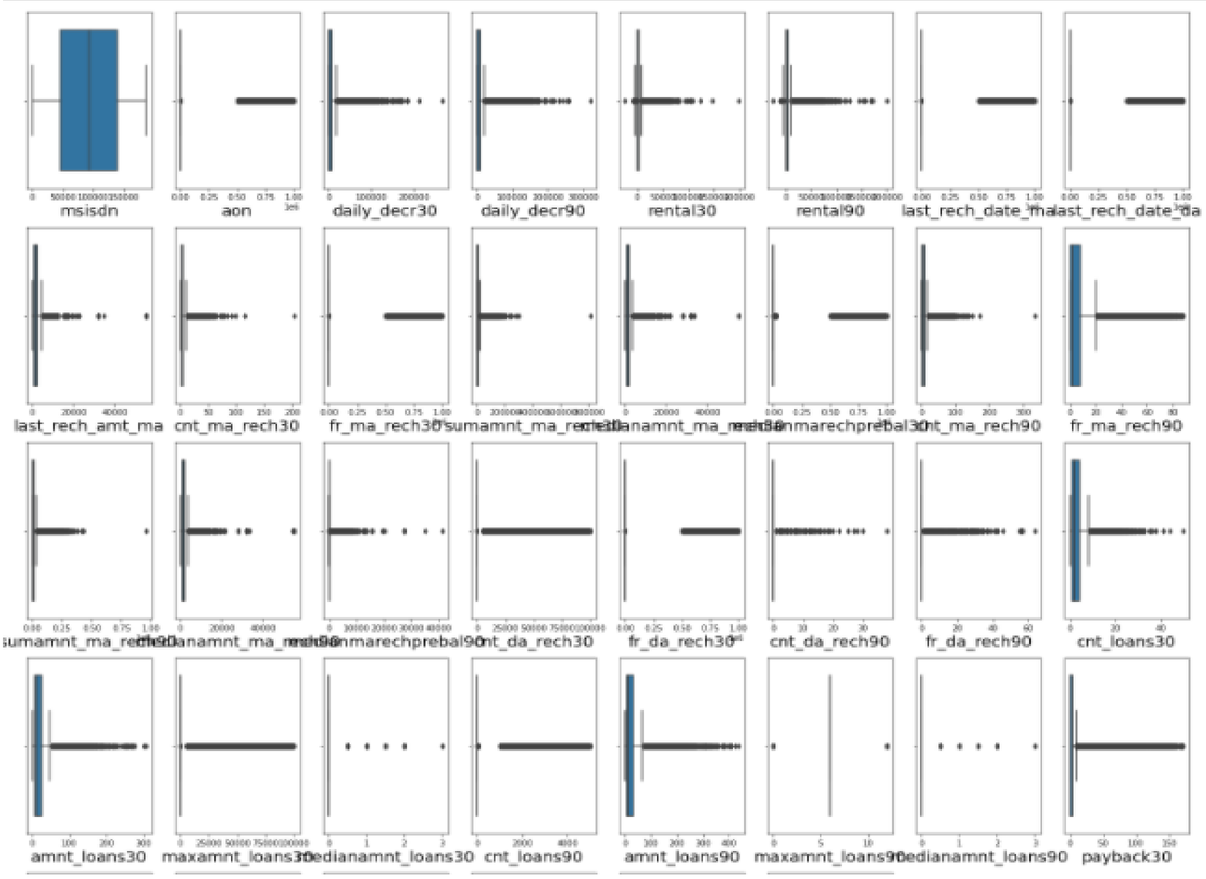


- Over 30 days the user has higher amount of loans with respect to increase in the no. of times the loan taken but there is no such relation seems for over 90 days the most of user has overcome the amount of loan taken.

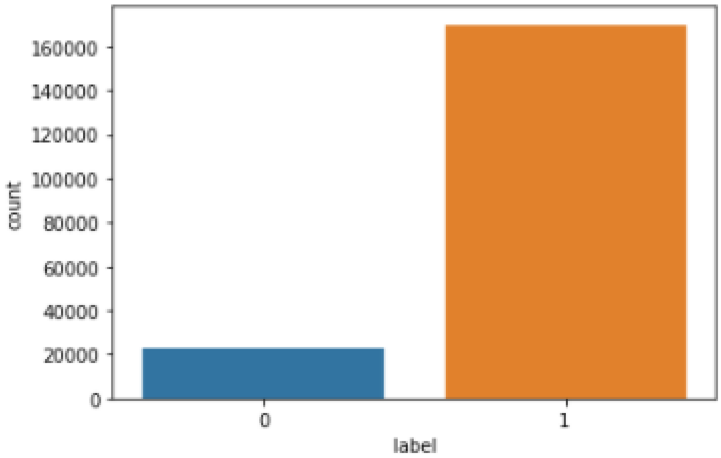


Observations:

All features are highly skewed except columns Msisdn, pdate, pday, pmonth, pyear, pweek.



Looks like all columns has outliers but 'daily_decr30', 'last_rech_amt_ma','cnt_ma_rech30','sumamnt_ma_rech30', 'medianamnt_ma_rech30' , 'medianamnt_ma_rech90','medianmarechprebal90' these columns has such very less no. records that are not usual/continuous. While others columns are seeming to be continuous.



Our target variable isn't balanced. We do **over sampling upto 75%** we incresed the no. of records.

- Interpretation of the Results

1. Pre-processing:

i. In this section we observed that the mobile no. of the users, pdate and pcircle are of object type while others are either of integer or float type.
ii. Pcircle contains only single value. So, this feature is to be removed.
iii. In the statistical description approx., all features have standard deviation greater than their mean. So, this will all be treated.
iv. No. of days and amount in description has minimum values in negative these also has to be removed.
v. No. of days that are greater than 18000 this means the users uses the credit card over 50 years. So, such records to be removed.
vi. There were some negative values for amount that also to be removed.
vii. There were some maximum values for amount & no. of transactions that seeming to be improper but it will be possible if the credit card is used by the group of users.

2. Visualizations

a. Over 30 days the user has higher amount of loans with respect to increase in the no. of times the loan taken but there is no such relation seems for over 90 days the most of user has overcome the amount of loan taken.
b. Over 30 days the user highly pay back their loan who has taken less no. of loan.
c. Here we can see that somehow the max_amnt_loan_over 90 days and daily amount spent over 90 are linear dependent.
d. Here we can see that Count of data recharge over 90 days increases w.r.t. decrease in frequency of data recharge over 90 days.
e. From average payback over 30 days is co-linear with the payback90.

f. There seems some relation that the user pay backs their loans who had recharge less no. of time for their data account as well as their main account too.

g. WE plot ROC curve to know the finalized model through which we know the finalized model.

3. Modelling

a. Before modelling we done feature selection.

b. We standardized the Features.

c. We did Upsampling up to 75% to balanced the dataset.

d. Used 5 algorithms: Logistic Regression, Decision Tree, Random Forest, AdaBoost, Bagging.

e. For every model we get highest accuracy at their best random State in (8, 10 & 12). With their evaluation metrices.

f. We Hyper tuned the model that has highest AUC curve and least difference between the best CV Score and accuracy of each model.

CONCLUSION

- Key Findings and Conclusions of the Study

➢ As we have real vales data. So, I didn't transform some records I just removed that are in-proper like negative values for amount, days, maximum values for these too.

➢ Remove such attributes that giving collinearity problems and has single vales in it.

➢ Reduce Outliers and skewness of the data.

➢ Chosen **Random Forest Classifier Algorithm** as the finalized model with the best AUC score and after hyper tuning the best accuracy i.e., 94.54%.

- Choose the final model based on weighted ROC-AUC curve and least difference between the best CV score and highest accuracy with their best random state..

Learning Outcomes of the Study in respect of Data Science

- Micro Credit solution provides operators and service providers with the ability to extend their service to their users through a small, short term credit facility.

Limitations of this work and Scope for Future Work

Micro credit loan facility is an emergency credit service, it allows a consumer to use the service by availing a loan that will be repaid within a given time. This is particularly useful in cases where subscribers or resellers of mobile network operators need airtime to make emergency calls or sell airtime respectively.

This loan can easily be recovered once the user recharges his prepaid account again. Because of its ability to improve operator revenues, enhance service delivery and ensure customer satisfaction, Micro Credit service proves to be a game changer for all stakeholders in the service delivery, distribution and consumption process.