FLIP ROBO

Micro Credit Loan Defaulter Prediction

Instant Loan
Intelligent User Profiling
Automatic Recovery
Business Continuity
Loan requested
Loan delivered
MicroCredit
Client Integration
Resellers & Subscribers
Seamless Micro Credit
Service Provider

Submitted by:

Dhrubajyoti Mandal

ACKNOWLEDGMENT

I would like to express my gratitude towards Flip Robo 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. Sapna Verma 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 payed i.e. Non- defaulter, while, Label ‘0’ indicates that the loan has not been payed 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

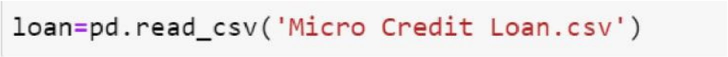
Micro Credit Loan is a value-added service designed on the premise of "what the consumer needs", provides ease of access of airtime stock credit to customers of a telecom operator when they run out of balance.

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.

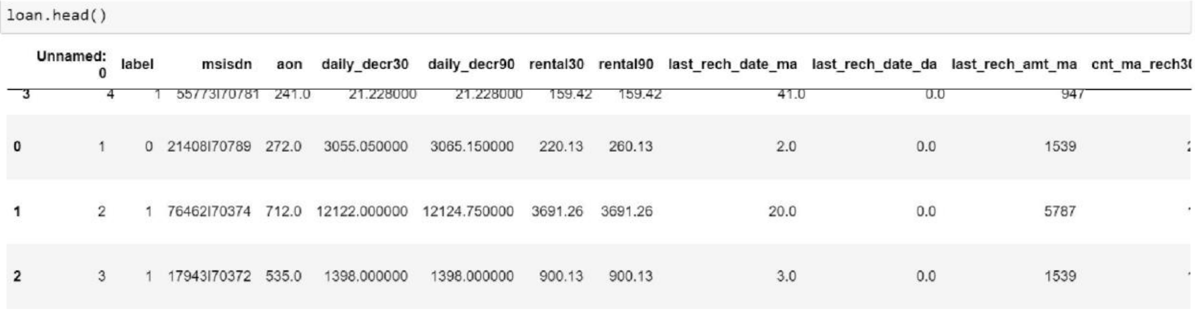
I completely agree that it is a very effective way of offering funds to the economically underprivileged sections of the society.

Analytical Problem Framing

- Dataset Representation:



```
loan=pd.read_csv('Micro Credit Loan.csv')
```



```
loan.head()
```

	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_rech3(
3	4	1	55773I70781	241.0	21.228000	21.228000	159.42	159.42	41.0	0.0	947	
0	1	0	21408I70789	272.0	3055.050000	3065.150000	220.13	260.13	2.0	0.0	1539	2
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	'

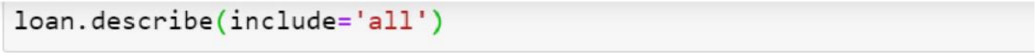
Observation:

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

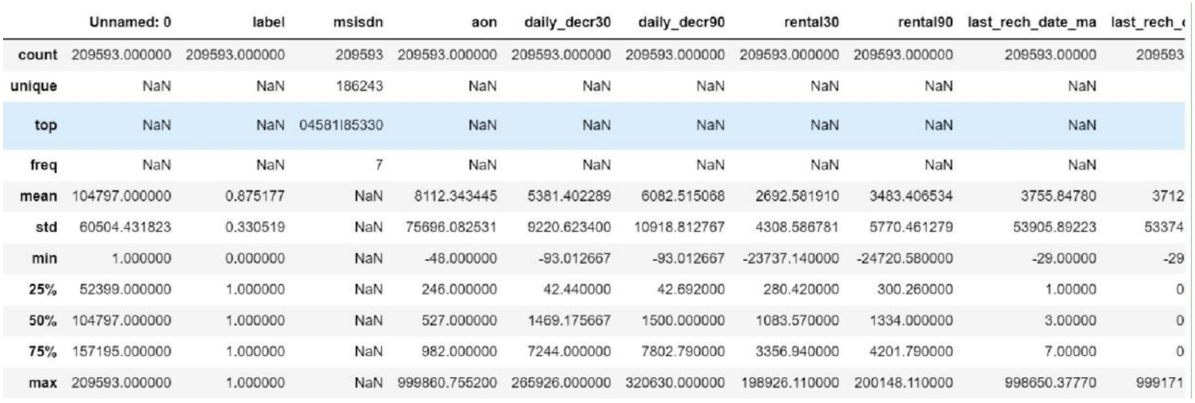
So clearly it is a classification problem.

- **Statistical Data:**



```
loan.describe(include='all')
```

Output:



	Unnamed: 0	label	msisdn	aon	daily_decr30	daily_decr90	rental30	rental90	last_rech_date_ma	last_rech_(
count	209593.000000	209593.000000	209593	209593.000000	209593.000000	209593.000000	209593.000000	209593.000000	209593.00000	209593
unique	NaN	NaN	186243	NaN	NaN	NaN	NaN	NaN	NaN	
top	NaN	NaN	04581I85330	NaN	NaN	NaN	NaN	NaN	NaN	
freq	NaN	NaN	7	NaN	NaN	NaN	NaN	NaN	NaN	
mean	104797.000000	0.875177	NaN	8112.343445	5381.402289	6082.515068	2692.581910	3483.406534	3755.84780	3712
std	60504.431823	0.330519	NaN	75696.082531	9220.623400	10918.812767	4308.586781	5770.461279	53905.89223	53374
min	1.000000	0.000000	NaN	-48.000000	-93.012667	-93.012667	-23737.140000	-24720.580000	-29.00000	-29
25%	52399.000000	1.000000	NaN	246.000000	42.440000	42.692000	280.420000	300.260000	1.00000	0
50%	104797.000000	1.000000	NaN	527.000000	1469.175667	1500.000000	1083.570000	1334.000000	3.00000	0
75%	157195.000000	1.000000	NaN	982.000000	7244.000000	7802.790000	3356.940000	4201.790000	7.00000	0
max	209593.000000	1.000000	NaN	999860.755200	265926.000000	320630.000000	198926.110000	200148.110000	998650.37770	999171

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

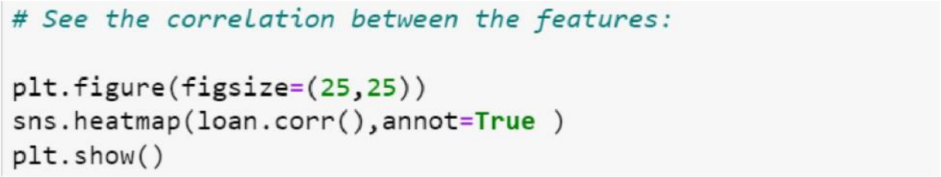
- Mobile numbers are integers type having an alphabet which we have removed.
- Age or number of days are never negative which we have treated.
- Extremely large positive values are data entered in either minutes or seconds instead of days.
- Amount spent cannot be negative hence that is being handled.
- Larger amounts in some feature are very natural as limitations in the data are not given.
- In some feature the values of 30 days cannot be more than 90 days value,hence it needs to be handled.
- 5 and 10 Indonasian Rupiah are the only loan amount options given for the consumer.
- There could be records with 0 as the loan amount as well.
- 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 Preprocessing Done

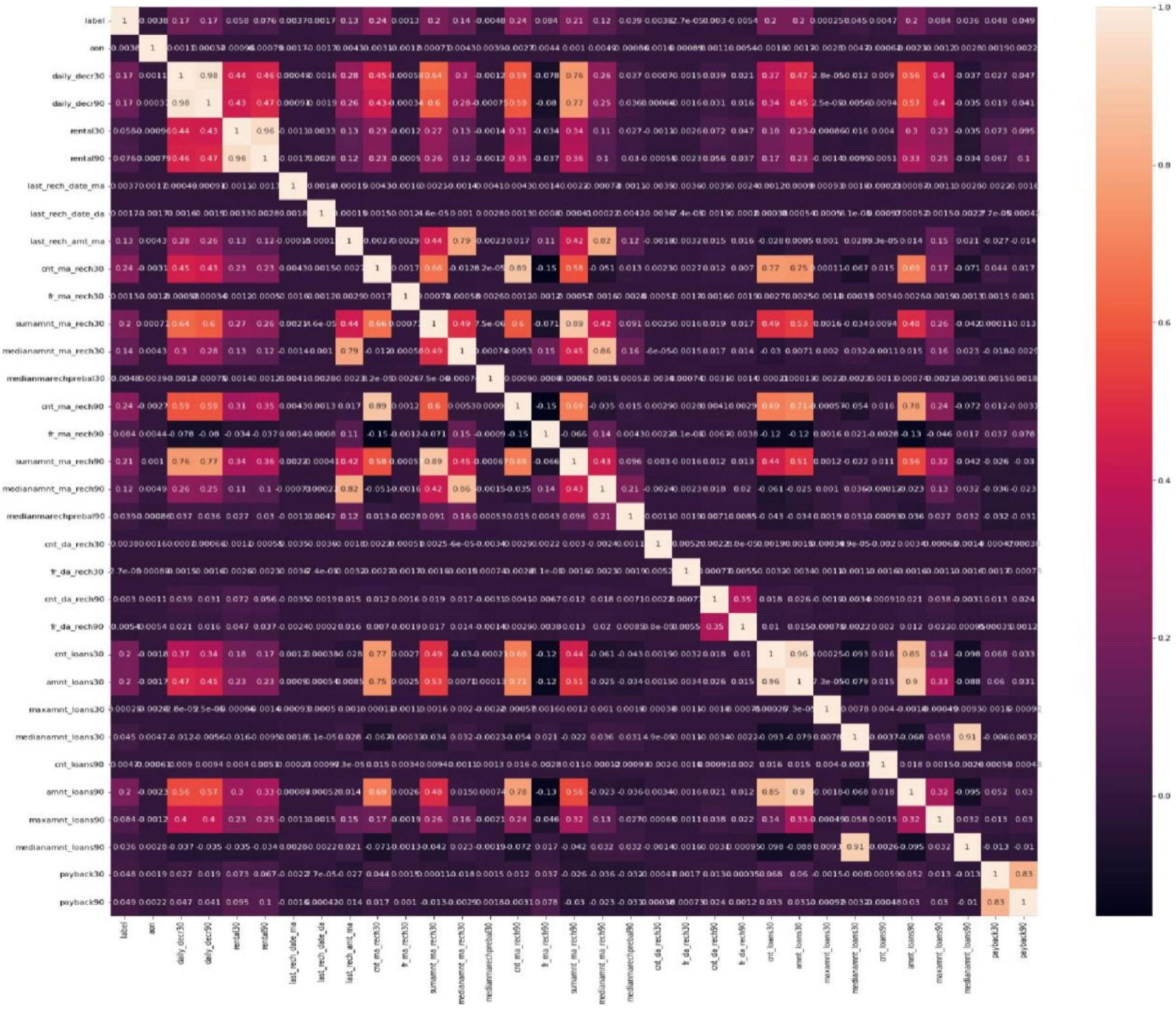
- From the above discription we see that almost our data is imbalanced, we have more counts of 1 than 0 in or target variable.
- Almost every independent variable is highly skewed to right and has outliers. aon shows negative value at minimum as age (in terms of days) cannot be negative, daily_decr30 and daily_decr90 also shows generative value at minimum that too needs to be corrected.
- last_rech_date_ma and last_rech_date_da - minimum value also here should be corrected.
- 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.
- I will replace the values which are more than (Q3 + 1.5(IQR)) with Q3 + 1.5(IQR) and in some columns where values are less than (Q1-1.5(IQR)) with Q1-1.5(IQR)
- 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 :



```
# See the correlation between the features:

plt.figure(figsize=(25,25))
sns.heatmap(loan.corr(),annot=True )
plt.show()
```



Observation:

1. The payback30 and payback90 show strong negative correlation with the target variable 'label'.
2. Rental30 and rental90 are hioghly correlated to daily_dec30 and daily_dec90.
3. amt_loan 30 and amt_loan 90 are strongly correlated.
4. cnt_amt30 and amt_loan30 are also strongly correlated.
5. medianamnt30 and medianamnt90 are highly correlated to each other.
6. Also some of the features such as amt_ma_rech30,amt_ma_rech90,cnt_ma_reach30 and cnt_ma_rech90 and cntda_reach30 and cnt_da_reach90 are mildly correlated.

- Assumption for the problem:

> So clearly it is a classification problem. We will be using both simple 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

- **Testing of Identified Approaches (Algorithms)**
- **Simple Techniques:**

> LogisticRegression()

> DecisionTreeClassifier()

> GaussianNB()

> SVC()

- **Ensemble Techniques:**

> AdaBoostClassifier()

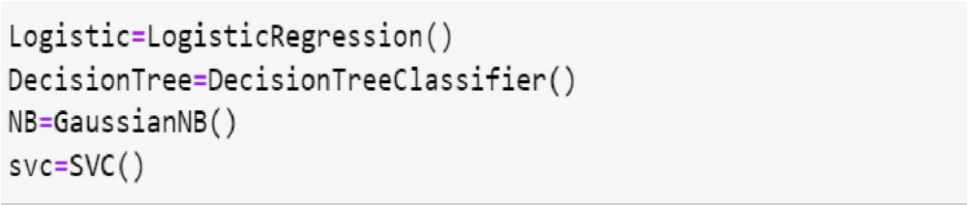
> GradientBoostingClassifier()

> RandomForestClassifier()

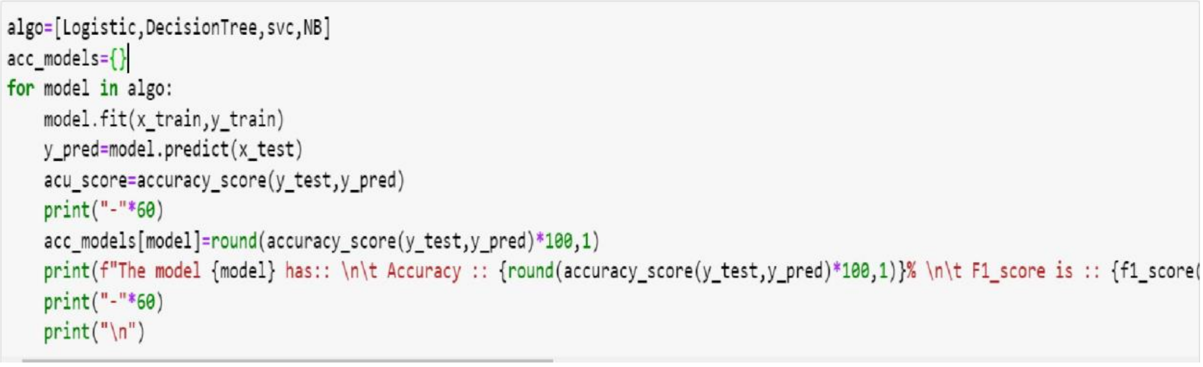
> XGBClassifier()

- Running the selected Models:

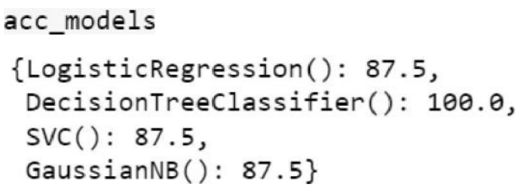
Simple Techniques:



```
Logistic=LogisticRegression()
DecisionTree=DecisionTreeClassifier()
NB=GaussianNB()
svc=SVC()
```

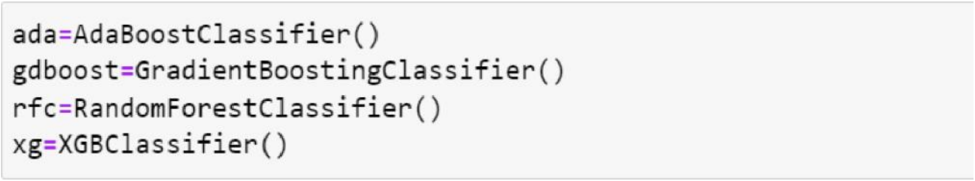


```
algo=[Logistic,DecisionTree,svc,NB]
acc_models={}
for model in algo:
    model.fit(x_train,y_train)
    y_pred=model.predict(x_test)
    acu_score=accuracy_score(y_test,y_pred)
    print("-"*60)
    acc_models[model]=round(accuracy_score(y_test,y_pred)*100,1)
    print(f"The model {model} has:: \n\t Accuracy :: {round(accuracy_score(y_test,y_pred)*100,1)}% \n\t F1_score is :: {f1_score(
    print("-"*60)
    print("\n")
```

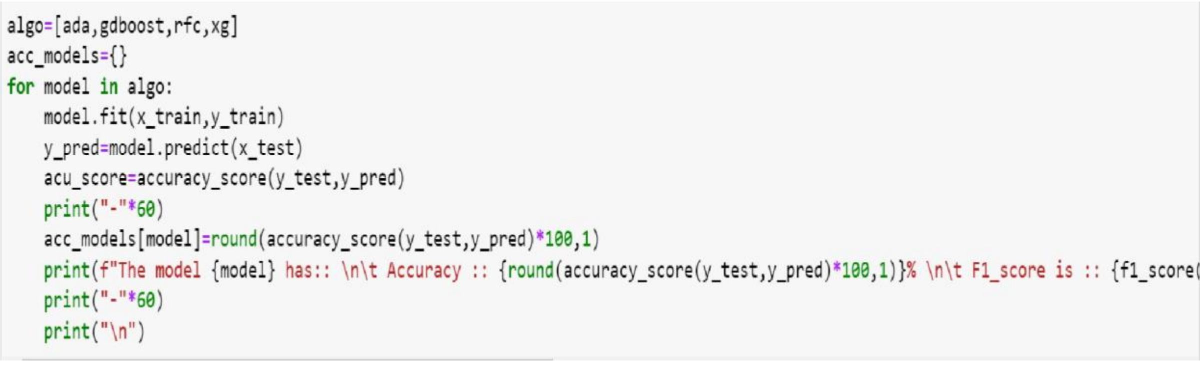


```
acc_models

 {LogisticRegression(): 87.5,
  DecisionTreeClassifier(): 100.0,
  SVC(): 87.5,
  GaussianNB(): 87.5}
```

Ensemble Techniques:



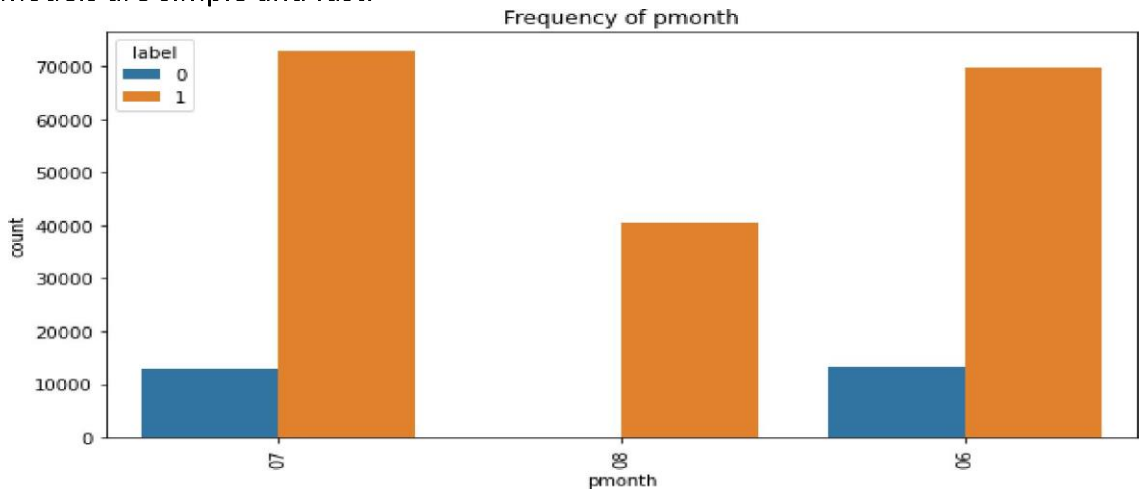
```
ada=AdaBoostClassifier()
gdboost=GradientBoostingClassifier()
rfc=RandomForestClassifier()
xg=XGBClassifier()
```



```
algo=[ada,gdboost,rfc,xg]
acc_models={}
for model in algo:
    model.fit(x_train,y_train)
    y_pred=model.predict(x_test)
    acu_score=accuracy_score(y_test,y_pred)
    print("-"*60)
    acc_models[model]=round(accuracy_score(y_test,y_pred)*100,1)
    print(f"The model {model} has:: \n\t Accuracy :: {round(accuracy_score(y_test,y_pred)*100,1)}% \n\t F1_score is :: {f1_score(
    print("-"*60)
    print("\n")
```

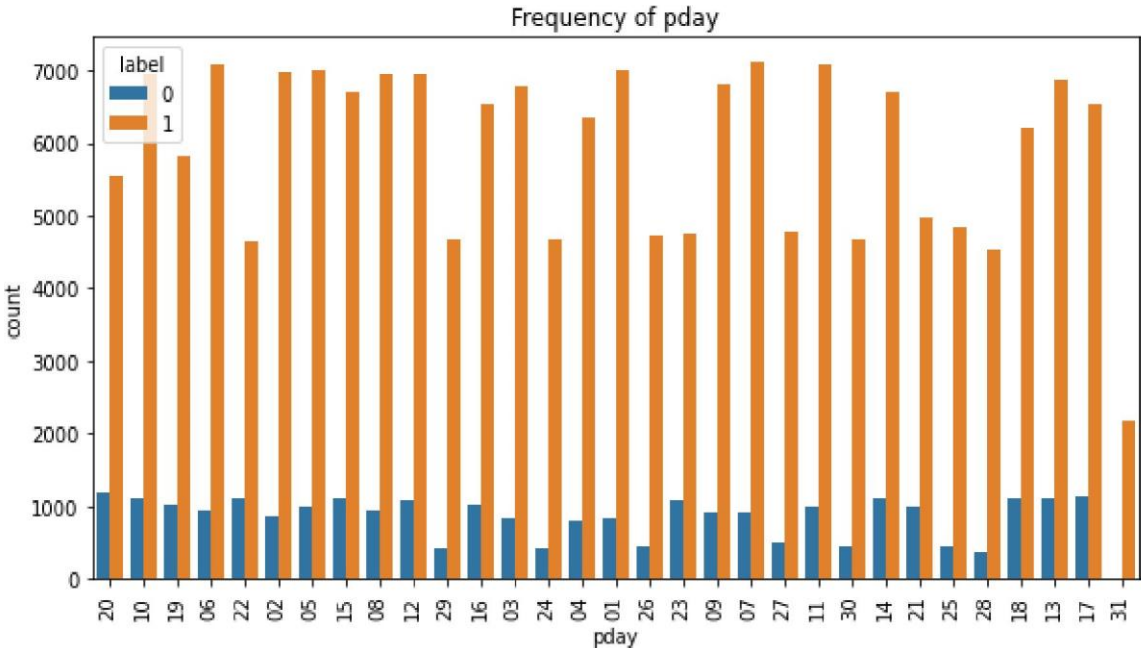
Observations:

1. All the models are showing good score. All the Ensemble techniques are performing better than simple models but seems to be overfitted.
2. We will be hypertuning the simple model for this problem since the simple models are also giving us very good results with low complexity. I am selecting LogisticRegression and DecisionTreeClassifier for hyper parameter tuning since these models are simple and fast. `



Observation:

The month of 6 and 7 have some defaulters. Month 8 doesn't have any defaulters.

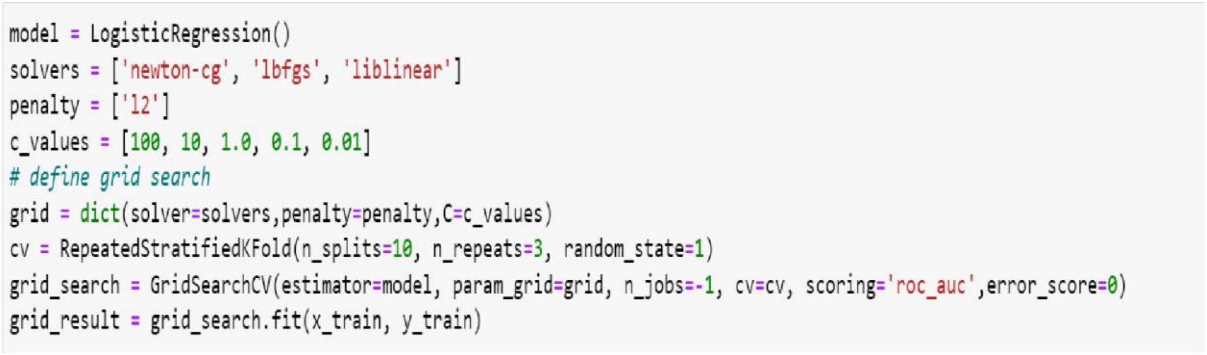


Observation:

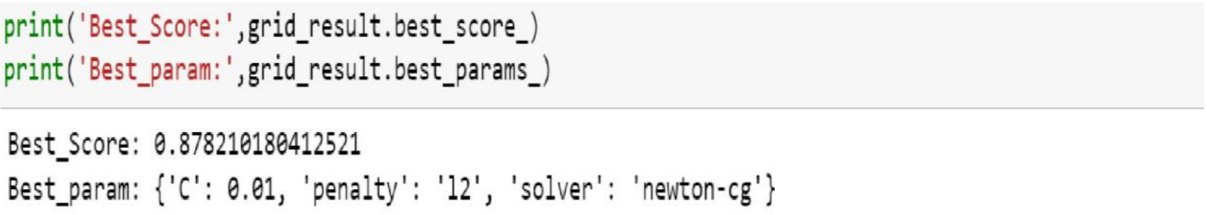
Almost on all the dates we have defaulters except on 31.

- Key Metrics for success in solving problem under consideration

  Hyper Tuning the Models Logistic Regression:

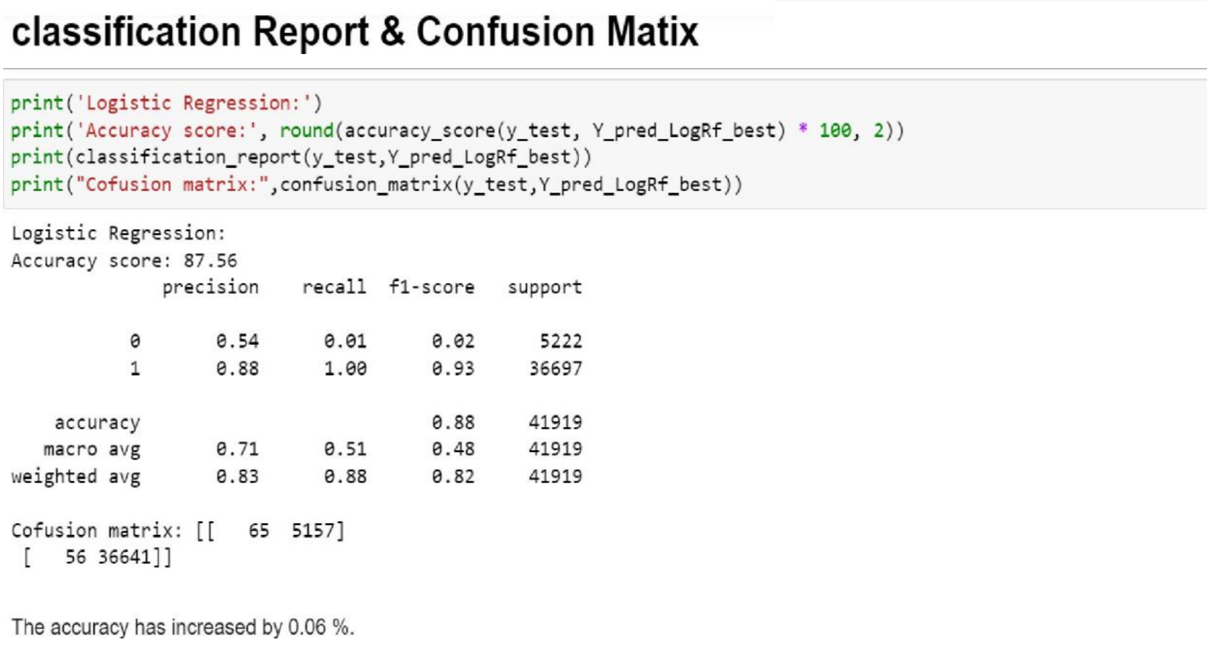


```
model = LogisticRegression()
solvers = ['newton-cg', 'lbfgs', 'liblinear']
penalty = ['l2']
c_values = [100, 10, 1.0, 0.1, 0.01]
# define grid search
grid = dict(solver=solvers,penalty=penalty,C=c_values)
cv = RepeatedStratifiedKFold(n_splits=10, n_repeats=3, random_state=1)
grid_search = GridSearchCV(estimator=model, param_grid=grid, n_jobs=-1, cv=cv, scoring='roc_auc',error_score=0)
grid_result = grid_search.fit(x_train, y_train)
```



```
print('Best_Score:',grid_result.best_score_)
print('Best_param:',grid_result.best_params_)
```

```
Best_Score: 0.878210180412521
Best_param: {'C': 0.01, 'penalty': 'l2', 'solver': 'newton-cg'}
```

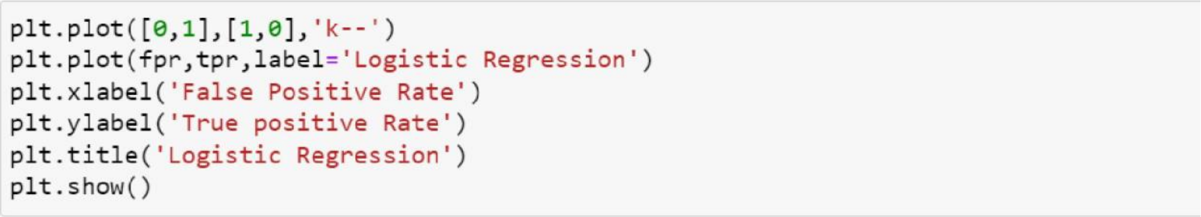


classification Report & Confusion Matix

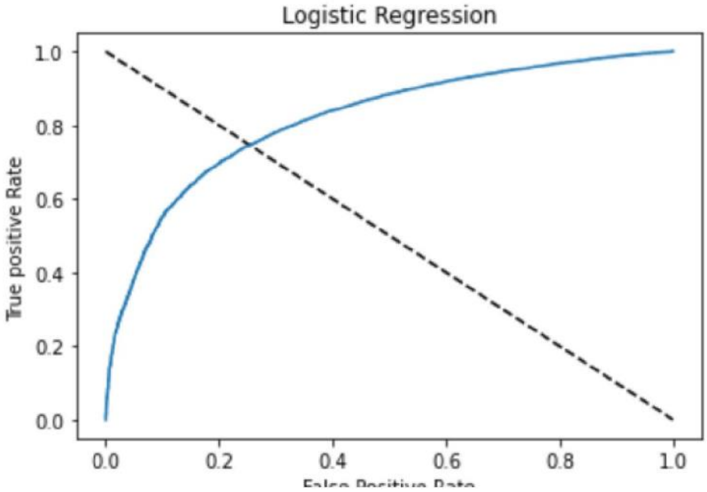
```
print('Logistic Regression:')
print('Accuracy score:', round(accuracy_score(y_test, Y_pred_LogRf_best) * 100, 2))
print(classification_report(y_test,Y_pred_LogRf_best))
print("Cofusion matrix:",confusion_matrix(y_test,Y_pred_LogRf_best))
```

```
Logistic Regression:
Accuracy score: 87.56
              precision    recall  f1-score   support

           0       0.54      0.01      0.02      5222
           1       0.88      1.00      0.93     36697

    accuracy                           0.88     41919
   macro avg       0.71      0.51      0.48     41919
weighted avg       0.83      0.88      0.82     41919

Cofusion matrix: [[   65  5157]
 [   56 36641]]
```

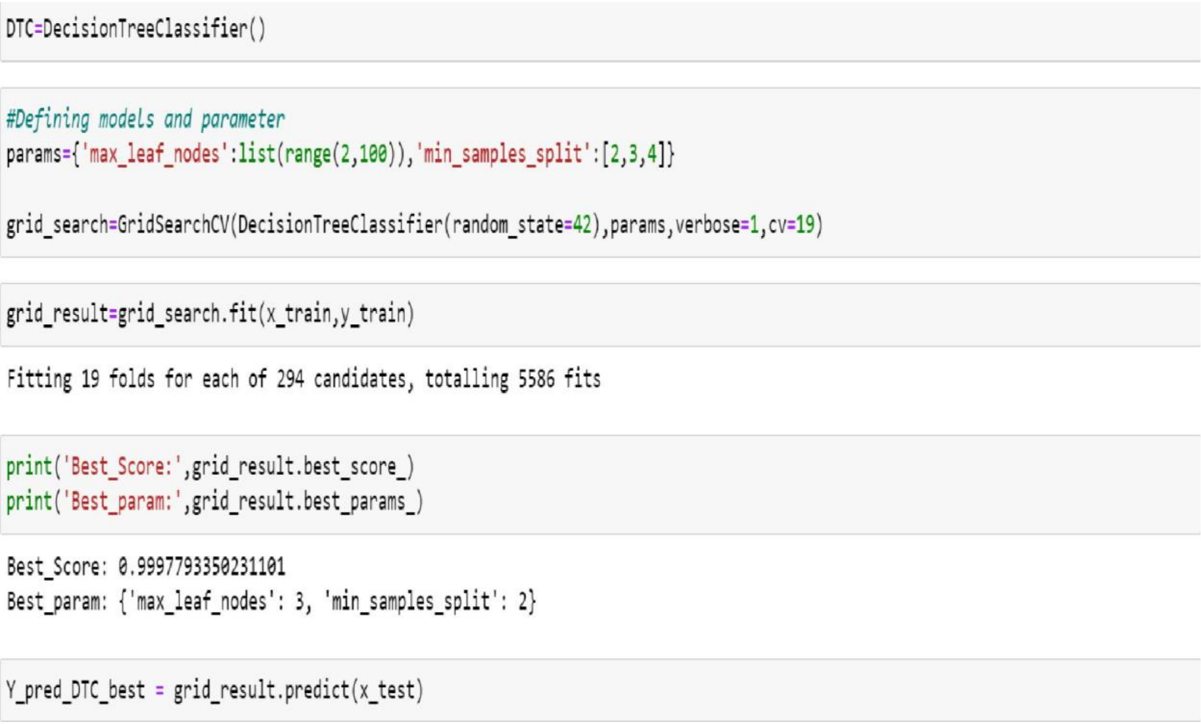
The accuracy has increased by 0.06 %.



```
plt.plot([0,1],[1,0],'k--')
plt.plot(fpr,tpr,label='Logistic Regression')
plt.xlabel('False Positive Rate')
plt.ylabel('True positive Rate')
plt.title('Logistic Regression')
plt.show()
```



Hypertune The Decision Tree Model:



```
DTC=DecisionTreeClassifier()
```

```
#Defining models and parameter
params={'max_leaf_nodes':list(range(2,100)),'min_samples_split':[2,3,4]}

grid_search=GridSearchCV(DecisionTreeClassifier(random_state=42),params,verbose=1,cv=19)
```

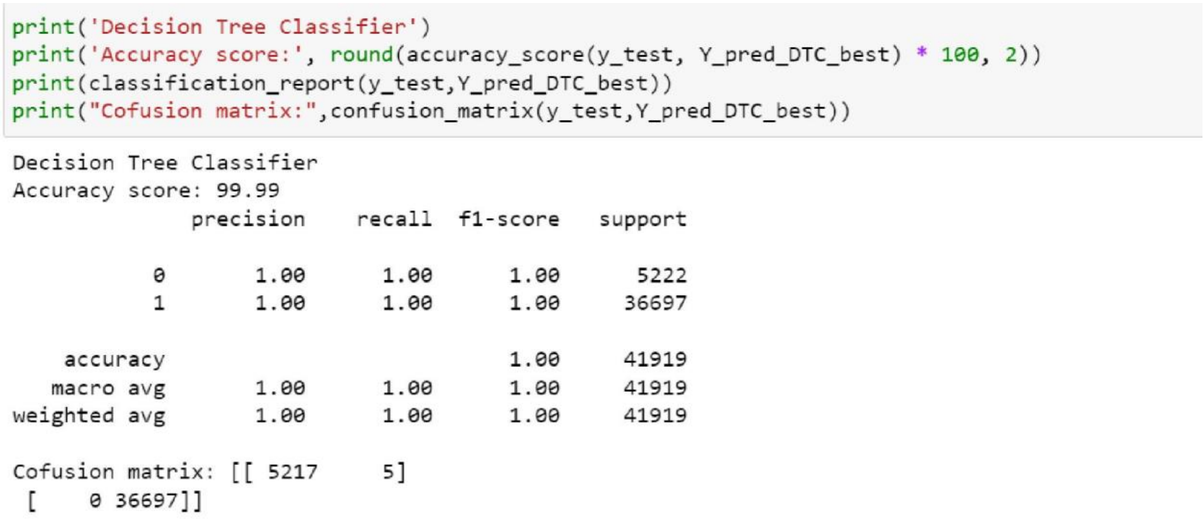
```
grid_result=grid_search.fit(x_train,y_train)
```

```
Fitting 19 folds for each of 294 candidates, totalling 5586 fits
```

```
print('Best_Score:',grid_result.best_score_)
print('Best_param:',grid_result.best_params_)
```

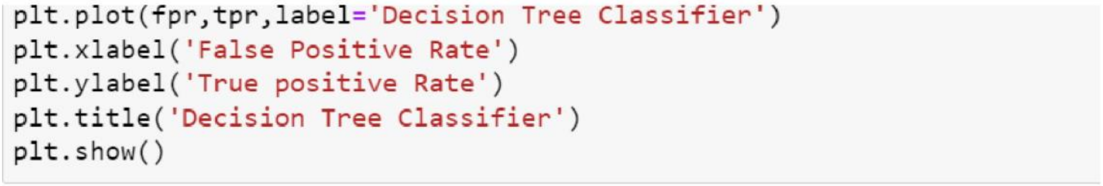
```
Best_Score: 0.9997793350231101
Best_param: {'max_leaf_nodes': 3, 'min_samples_split': 2}
```

```
Y_pred_DTC_best = grid_result.predict(x_test)
```

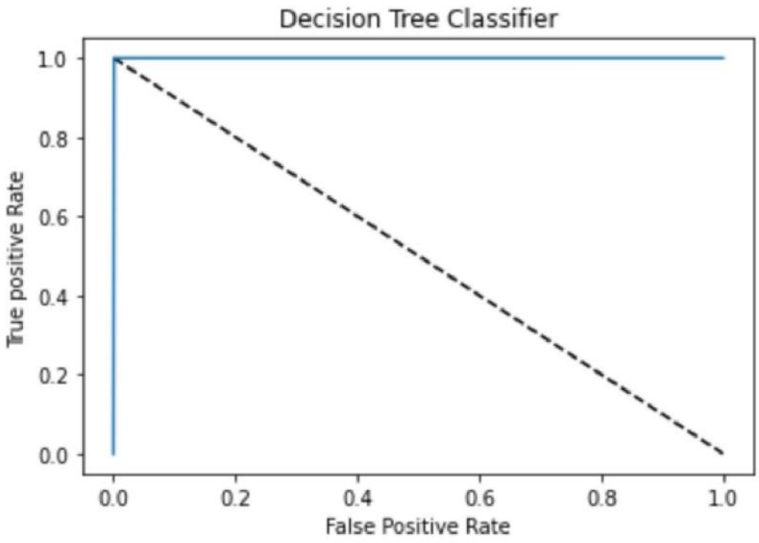


```
print('Decision Tree Classifier')
print('Accuracy score:', round(accuracy_score(y_test, Y_pred_DTC_best) * 100, 2))
print(classification_report(y_test,Y_pred_DTC_best))
print("Cofusion matrix:",confusion_matrix(y_test,Y_pred_DTC_best))
```

```
Decision Tree Classifier
Accuracy score: 99.99
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      5222
           1       1.00      1.00      1.00     36697

    accuracy                           1.00     41919
   macro avg       1.00      1.00      1.00     41919
weighted avg       1.00      1.00      1.00     41919

Cofusion matrix: [[ 5217     5]
 [    0 36697]]
```



```
plt.plot(fpr,tpr,label='Decision Tree Classifier')
plt.xlabel('False Positive Rate')
plt.ylabel('True positive Rate')
plt.title('Decision Tree Classifier')
plt.show()
```



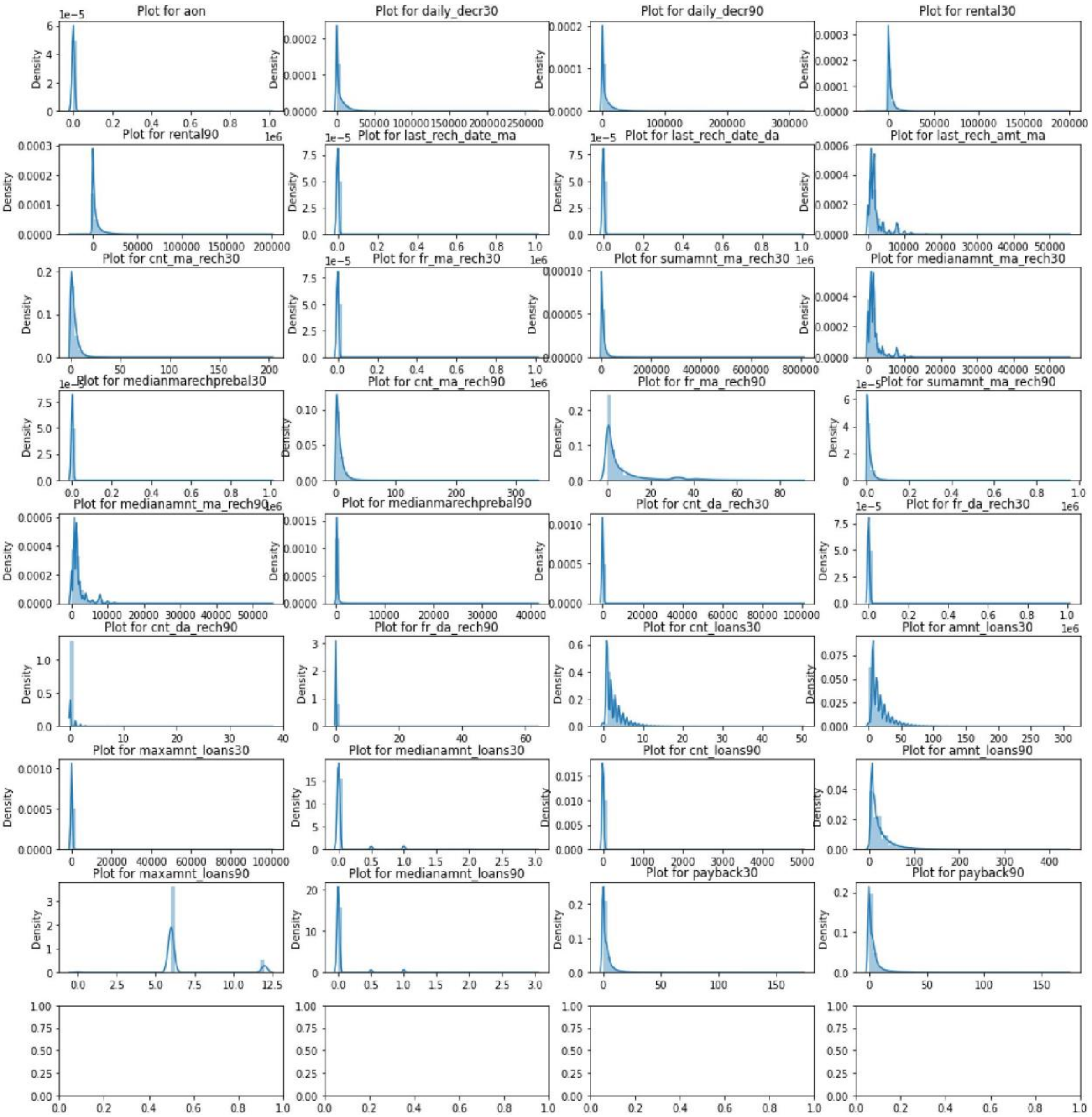
The accuracy has decreased fromm 100% to 99.99%.

- Mobile numbers are integers type having an alphabet which we have removed.
- Age or number of days are never negative which we have treated.
- Extremely large positive values are data entered in either minutes or seconds instead of days.
- Amount spent cannot be negative hence that is being handled.
- Larger amounts in some feature are very natural as limitations in the data are not given.
- In some feature the values of 30 days cannot be more than 90 days value,hence it needs to be handled.
- 5 and 10 Indonasian Rupiah are the only loan amount options given for the consumer.

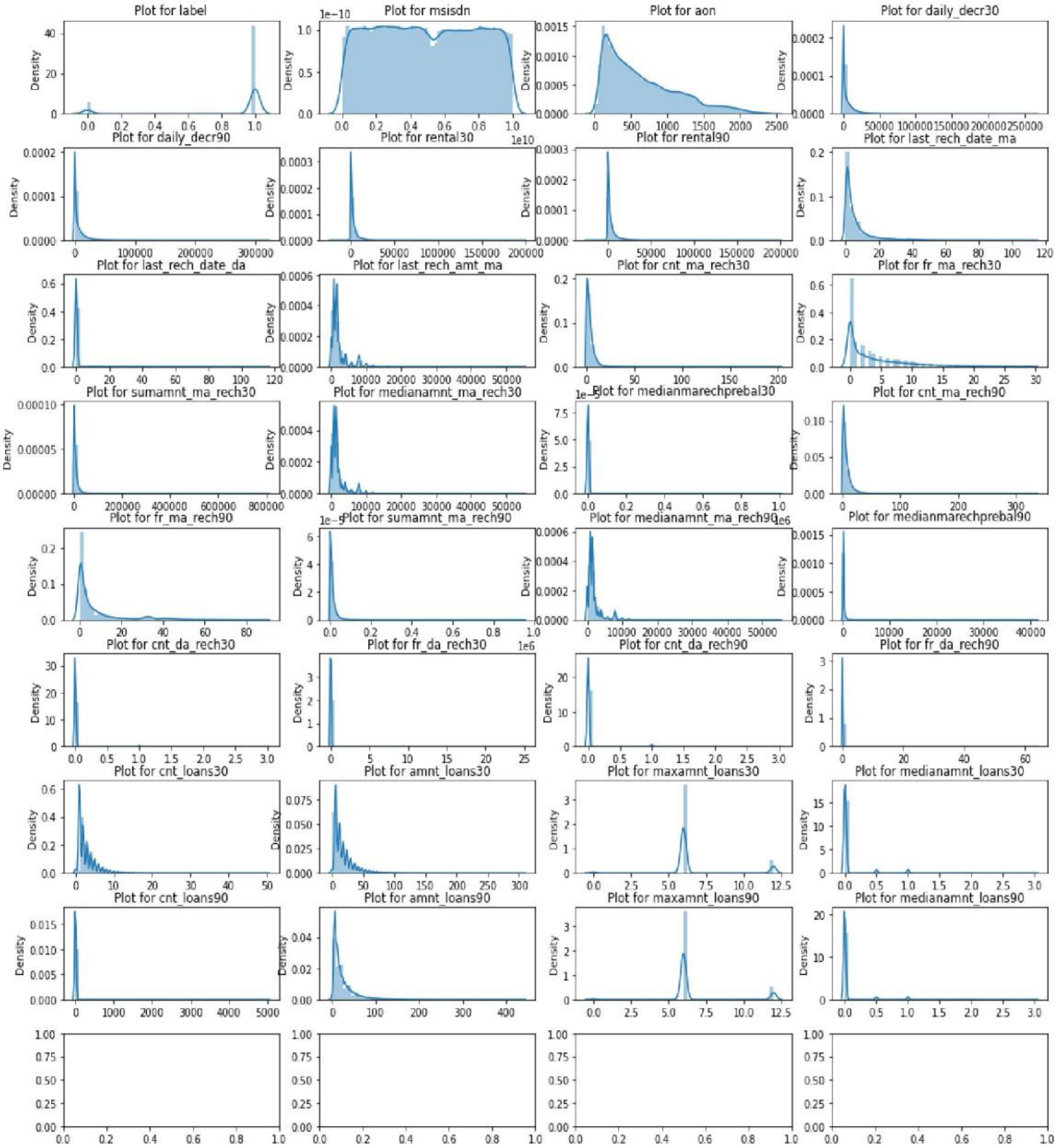
- There could be records with 0 as the loan amount as well.
- Retun amount can be 0,6 and 12 only.
- The customer will be a defaulter if they don't pay back the loan amount within 5 days of issuing the loan. So, the Average payback value will be less than or equal to 5 for records with label = 1 and Average payback value to be greater than 5 for records with label = 0.

- Visualizations(Original Data and Clean Data)

  Original Data:



Clean Data:



Observation:

- After cleaning the data we can see some improvements in the data distribution.
- The distributions still need to be transformed since some data are still skewed and need to be scaled.

- Interpretation of the Results

Some important inferences that is being made are:

Aon

- After cleaning the date, the 'aon' variable can distinguish between the labels 1 and 0.
- The tenure of defaulters on an average is lesser than the tenure of defaulters on the cellular network.

daily_decr30

- Most of the values are pretty close.
- The labels 0 and 1 are somewhat distinguished.
- The daily spent amount for 30 days for defaulters is lesser than non defaulters on an average

daily_decr90

- Most of the values are pretty close. The labels 0 and 1 are somewhat distinguished.
- The daily spent amount for 90 days for defaulters is slightly lesser than non defaulters on an average.

last_rech_date_ma

- The classes 0 and 1 are showing some differences.
- The last recharge date on the main account is slightly lesser for non-defaulters than defaulters.

last_rech_amt_ma

- The classes 0 and 1 in 'label' in distributions are showing some difference.
- The last recharged amount on an average for defaulters

fr_ma_rech30

- We can see that the distribution of labels 0 &1 and well distinguished in the fr_ma_rech30 after cleaning.Left is before cleaning and right is after cleaning.
- The frequency of recharging the main account done for 30 days by non defaulters is greater than defaulters.

medianamnt_ma_rech30

- The graph above shows that the Defaulters have done a slightly lesser amount of recharges to their main account than non-defaulters.

cnt_ma_rech90

- On an average, the number of recharges done by the defaulters over 90 days is slightly lesser than the number of recharges done by non-defaulters.

fr_ma_rech90

- The classes 0 and 1 show some distinctions.
- On an average, the frequency of recharges done by defaulters on the main account over a period of 90 days is lesser than that of non-defaulters

sumamnt_ma_rech90

- On an average, the total amount of recharge done by defaulters over 90 days is slightly lesser than done by non-defaulters

medianamnt_ma_rech90

- The label 0 and 1 distribution in medianamnt_ma_rech90 show some difference.
- On an average, the Median amount of recharge done by defaulters on the main account over 90 days is slightly lesser than that done by non-defaulters.

cnt_loan30

- The cnt_loans30 distribution shows some difference between labels 0 and 1
- On an average, the defaulters have taken a lesser number of loans over 30 days than non defaulters.

amnt_loans30

- The amnt_loans30 distribution shows good difference between labels 0 and 1.
- The total amount of loans taken by defaulters in the last 30 days is lesser than that of non-defaulters on an average.

maxamnt_loans30

- The distribution means are showing some distinction between labels 0 and 1 for maxamnt_loans30.
- The maximum amount taken as loan by defaulters is on an average lesser than the non-defaulters.

amnt_loans90

- The distributions between 0 and 1 labels show good difference in amnt_loans90.
- On an average, the total amount of loans taken by the defaulters is lesser than non-defaulters.

maxamnt_loans90

- On an average, the maximum amount of loans taken by defaulters over a period of 90 days is lesser than the maximum amount of loans taken by non-defaulters.

CONCLUSION

• Pay Back Observations

- After cleaning the payback variables show pretty good difference between labels 0 and 1.
- On an average, the time taken by defaulters to pay back the loan is greater than the time taken by non-defaulter to pay back the loan over a period of 30 or 90.
- Chosen **Decision Tree Classifier Algorithm** with 99.99% accuracy as my best model. - Choose the final model based on weighted ROC-AUC curve and confusion metrics.

• 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.