FLIP ROBO

PROJECT REPORT

ON

MICRO CREDIT DEFAULTER PROJECT

SUBMITTED BY

V Samyukta

ACKNOWLEDGMENT

It gives me immense pleasure to deliver this report. Working on this project was a great learning experience that helped me attain in-depth knowledge on data analysis process.

Flip Robo Technologies (Bangalore) provided all of the necessary information and datasets, required for the completion of the project.

I express my gratitude to my SME, Ms. Khushboo Garg, for providing the dataset and directions for carrying out the project report procedure.

My heartfelt gratitude to DataTrained institute and FlipRobo company for providing me this internship opportunity. Last but not least to my sincere thanks to my family and all those who helped me directly or indirectly in completion this project.

CONTENTS

1.INTRODUCTION

- **Business Problem Framing:**

  Microfinance Institutions (MFIs) are businesses that provide financial services to low-income people. When addressing unbanked poor families living in rural places with few sources of income, MFS becomes quite effective. Group Loans, Agricultural Loans, Individual Business Loans, and other Microfinance Services (MFS) are some of the Microfinance Services (MFS) provided by MFI. Many microfinance institutions (MFI), experts, and donors support the idea of using mobile financial services (MFS), which they believe are more convenient, efficient, and cost-effective than the traditional high-touch model that has been used for many years to deliver microfinance services. Though the MFI industry is primarily focused on low-income families and is extremely beneficial in these areas, MFS implementation has been uneven, with significant challenges and successes. Microfinance is now widely accepted as a tool for poverty reduction, with $70 billion in outstanding loans and a global client base of 200 million people.
  We are now working with a client in the telecom industry. They are a provider of fixed wireless telecommunications networks. They've released a number of products and built their business and organization around the budget operator model, which entails providing better products at lower prices to all value conscious clients via a disruptive innovation strategy that focuses on the subscriber. They've teamed up with a microfinance institution to offer micro-credit on mobile balances that must be paid back in five days. If the Consumer deviates from the course of repaying the loaned amount within the time period of 5 days, he is considered a defaulter. The payback amount for a loan of 5 (in Indonesian Rupiah) should be 6 (in Indonesian Rupiah), whereas the payback amount for a loan of 10 (in Indonesian Rupiah) should be 12. (in Indonesian Rupiah). Build a model that can be used to predict if a client will pay back the lent amount within 5 days of loan insurance in terms of a probability for each loan transaction. Label '1' shows that the loan has been paid, indicating that it is a non-defaulter, whereas Label '0' indicates that the loan has not been paid, indicating that it is a defaulter.

- **Conceptual Background of the Domain Problem:**

  Micro Credit is a technology and service package given in collaboration with telecom providers at the final stage of product delivery.
  Some of the significant features of microfinance are as follows:

1. The borrowers are generally from low-income backgrounds
2. Loans availed under microfinance are usually of small amount, i.e., micro loans
3. The loan tenure is short
4. Microfinance loans do not require any collateral
5. These loans are usually repaid at higher frequencies
6. The purpose of most microfinance loans is income generation

- **Review of Literature**

  With the rapid growth of technology and increased competition, telecom firms are looking for ways to improve the quality of their service and, as a result, the health of their revenue. Miniature credit arrangement furnishes administrators and specialist organizations with the capacity to stretch out their support of their clients through a little, transient credit office. At the point when we go through the dataset given, we should look at deliberately every one of the characteristics given, arrange the clients between defaulters and non-defaulters, and lessen the possibility of deceitfulness in miniature credit advances by clients.

- **Motivation for the Problem Undertaken:**

  In this task, I need to construct an AI model which makes expectations to find deceitful clients in light of their past exercises which makes it simpler for specialist organizations and telecoms to give this office to their wholesalers, affiliates, and endorsers by limiting the credit risk for them. This takes the weight off the shoulders of the administrator who can zero in on working on the nature of administration being given to the clients.

2. ANALYTICAL PROBLEM FRAMING

- **Mathematical / Analytical Modelling of the Problem:**

  This dataset contains 209593 rows and 36 columns. By looking into the target column, I concluded that it a classification problem as my label column has categorical values so I have to use all classification algorithms while building the model. Also, I observed in some columns more than 90% of values are equal to 0. So, I decided to drop those columns. If I keep those columns as it is, it will create high skewness in the model. While checking the null values in

the datasets I found there are no null values in the dataset. To get better insight on the features I used plots like distribution plot, bar plot and heat plot with these plots I was able to understand the relation between the features in better manner. Also, I found outliers and skewness in the dataset so I removed outliers using percentile method and I removed skewness using Z-score method. I have used all the classification models while building model then turned the best model and saved the best model and predicted the defaulters.

- **Data Sources and their formats:**

  This data is given to FlipRobo from their client dataset. The information given in the dataset is in the CSV format. I'll be changing it into DataFrame to perform the essential procedures like choosing, erasing, adding, and renaming. There are 209593 rows and 37 columns in the dataset provided. All the attributes are numerical datatypes except ‘pCircle’ and ‘pdate’.

- **Data Pre-processing Done:**

  In Machine Learning, data pre-processing refers to the process of cleaning and organising raw data in order to make it appropriate for creating and training Machine Learning models. In other words, anytime data is received from various sources, it is collected in raw format, which makes analysis impossible. Data pre-processing is a crucial stage in Machine Learning since the quality of data and the relevant information that can be gleaned from it has a direct impact on our model's capacity to learn; consequently, we must pre-process our data before feeding it into our model. As a result, it is the first and most important stage in developing a machine learning model.

  I have used some following pre- processing steps:

  - Loading the training dataset as a dataframe.
  - Used pandas to set display I ensuring we do not see any truncated information.
  - Checked the number of rows and columns present in our training dataset.
  - Checked for missing data and the number of rows with null values.
  - Dropped all the unwanted columns and duplicate data present in our dataframe.
  - Separated categorical column names and numeric column names in separate list variables for ease in visualization.
  - Checked the unique values information in each column to get a gist for categorical data.
  - Used Pandas Profiling during the visualization phase along with dist plot, count plot, heatmap and the others.
  - Thoroughly checked for outliers and skewness information.

- With the help of heatmap, correlation bar graph was able to understand the Feature vs Label relativity and insights on multicollinearity amongst the feature columns.
- Separate feature and label data to ensure feature scaling is performed avoiding any kind of biasness.
- Checked for the best random state to be used on our classification Machine Learning model pertaining to the feature importance details.
- Finally created a classification model function along with evaluation metrics to pass through various model formats.

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

  There are 87.5% of non-defaulters and 12.5% of defaulter customers, the data is
  unbalanced, the target variable should be balanced before feeding the data to model. After the removal of outliers, there is not much skewness present in the data, as
  we can see all the values of skewness are less than 10. There is only 1 value 'maxamnt_loans30' with a higher value, we can proceed by dropping this attribute. The correlation graph shows the attributes are having positive and negative
  correlations. The attributes with values near zero are not contributing to the model prediction. We can drop them using t h e feature engineering technique.

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

  Hardware technology being Used:-
  - CPU: HP Pavilion
  - Chip: intel core13 8th Gen
  - RAM: 8 GB

  Software Technology being Used:-

  - Programming language: Python
  - Distribution: Anaconda Navigator
  - Browser based language shell: Jupyter Notebook

  Libraries/Packages Used:-

Pandas, NumPy, matplotlib, seaborn, scikit-learn and pandas_profiling.

3.DATA ANALYSIS AND VISUALIZATION

- **Identification of possible problem-solving approaches (methods):**

  - To tackle the problem, I employed both statistical and analytical methodologies, which mostly included data pre-processing and EDA to examine the connection of independent and dependent characteristics. In addition, before feeding the input data into the machine learning models, I made sure that it was cleaned and scaled.
  - We need to anticipate micro credit defaulters for this project, our target column is a categorical data. Which makes this project as classification challenge. I evaluated the prediction using a variety of classification methods. After a series of assessments, I determined that Extra Trees classifier is the best method for our final model since it has the best r2-score and the smallest difference in r2-score and CV-score of all the algorithms tested
  - I used K-Fold cross validation to gain high performance and accuracy. Then hyper parameter tweaked the final model.
  - Once I had my desired final model, I made sure to save it before loading the testing data.

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

  To feed the dataset to the model, the independent and dependent variables are to be split and the independent attributes are standardized using 'StandardScaler' library. Now the data obtained is clean and is having least multicollinearity, independent and balanced data. 'train_test_split' function in model selection is used for splitting data arrays into two subsets: for training data and for testing data. We train the model using the training set and then apply the model to the test set. I have chosen 5 classification machine learning algorithms which is suitable for the pre-processed and cleaned data to train and test the dataset.

- RandomForestClassifier
- ExtraTreeClassifier
- GradientBoostingClassifier

- SupportVectorClassifier

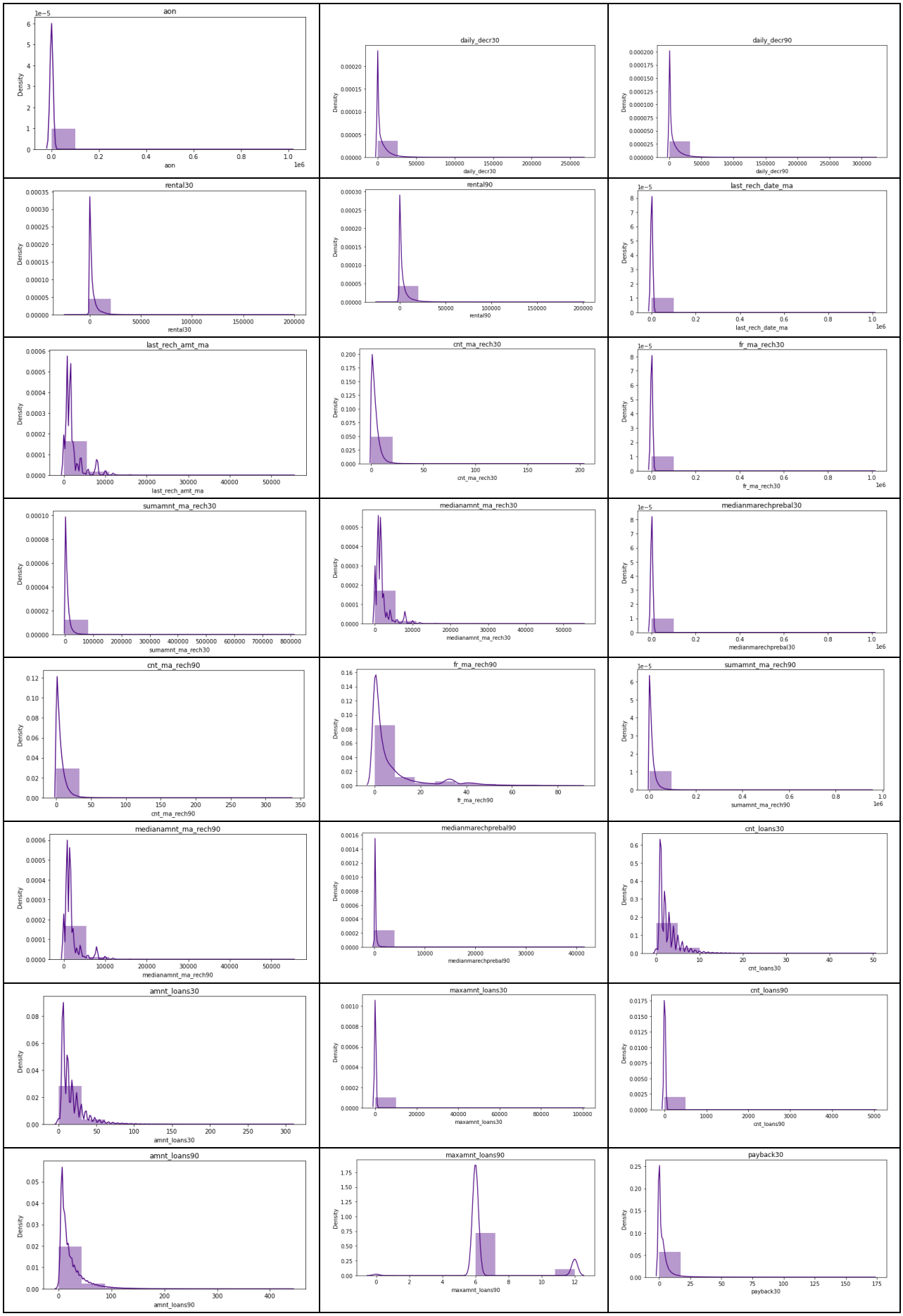
- **Key Metrics for success in solving problem under consideration:**

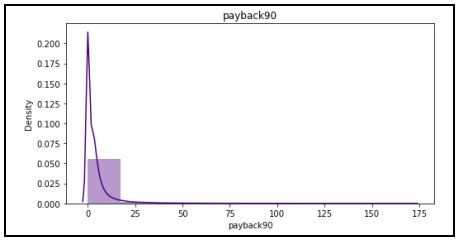
  A key/evaluation metric quantifies the performance of a predictive model This typically. Following are the metrics I have used to evaluate the model performance.
  1. Confusion Matrix: The confusion matrix provides a more insightful picture based on the counts of test records correctly and incorrectly predicted by the model, and what type
  of errors are being made. The confusion matrix is useful for measuring recall (also known as Sensitivity), Precision, Specificity, Accuracy, and, most importantly, the AUC-ROC Curve.
  2. Sensitivity: It measures how many observations out of all positive observations have we classified as positive. It tells us how many fraudulent transactions we recalled from all fraudulent transactions.
  3. Precision: It measures how many observations predicted as positive are in fact positive. Taking our fraud detection example, tells us what ratio of transactions is correctly classified as fraudulent.
  4. Accuracy: It measures how many observations, both positive and negative, were correctly classified.
  5. F1 Score: A good F1 score means that we have low false positives and low false negatives, so we're correctly identifying real threats, and we are not disturbed by false alarms.
  6. Cross-Validation Score: It is commonly used in applied machine learning to compare and select a model for a given predictive modelling problem because it is easy to understand, easy to implement, and results in skill estimates that generally have a lower bias than other methods.

- **Visualization:**

  I have used bar plots to see the relation of categorical feature and I have used 2 types of plots for numerical columns one is strip plot for ordinal features and other is reg plot for continuous features.

  - **<u>Visualization of numerical features with target:</u>**

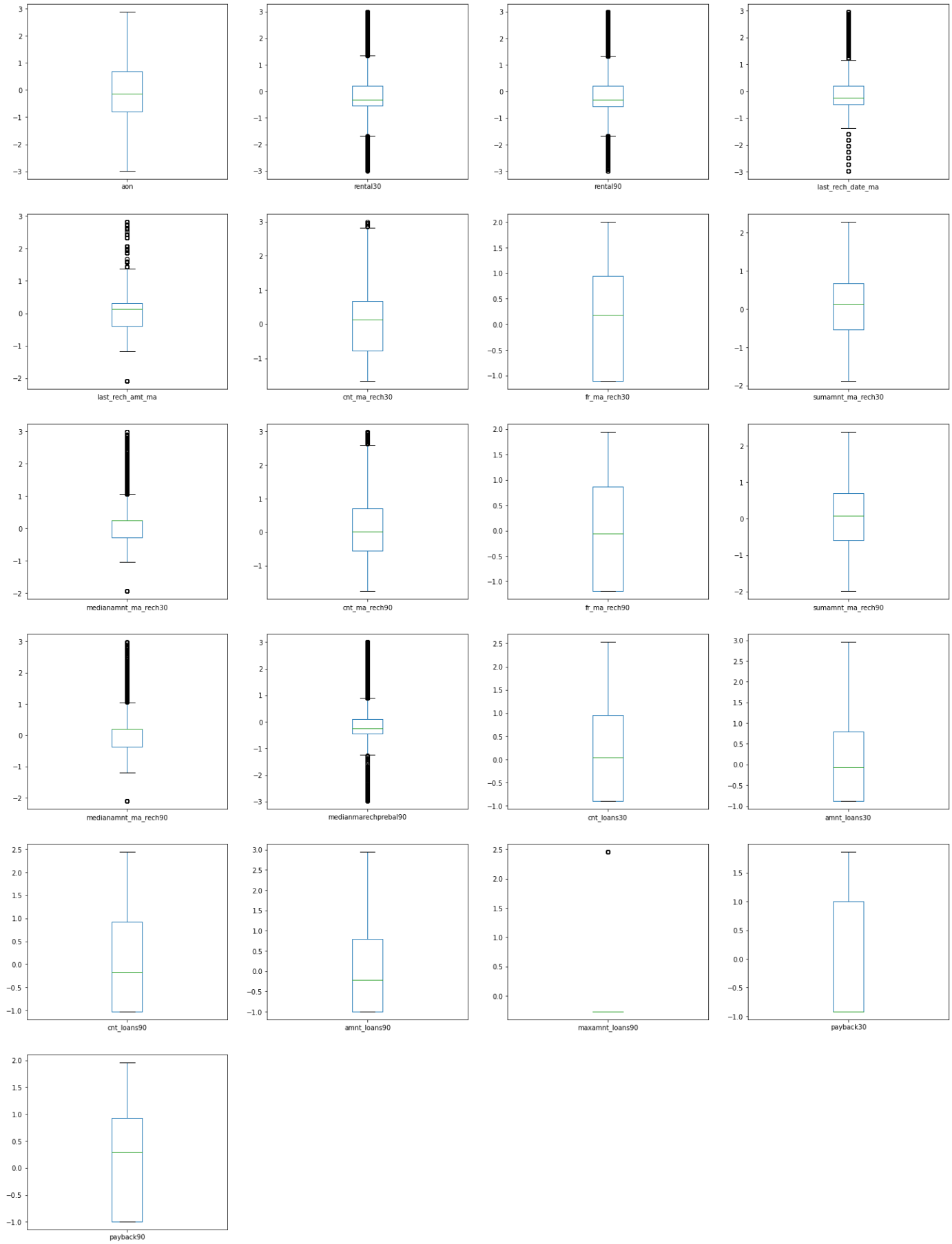

aon
daily_decr30
daily_decr90
rental30
rental90
last_rech_date_ma
last_rech_amt_ma
cnt_ma_rech30
fr_ma_rech30
sumamnt_ma_rech30
medianamnt_ma_rech30
medianmarechprebal30
cnt_ma_rech90
fr_ma_rech90
sumamnt_ma_rech90
medianamnt_ma_rech90
medianmarechprebal90
cnt_loans30
amnt_loans30
maxamnt_loans30
cnt_loans90
amnt_loans90
maxamnt_loans90
payback30
Density



Observations:

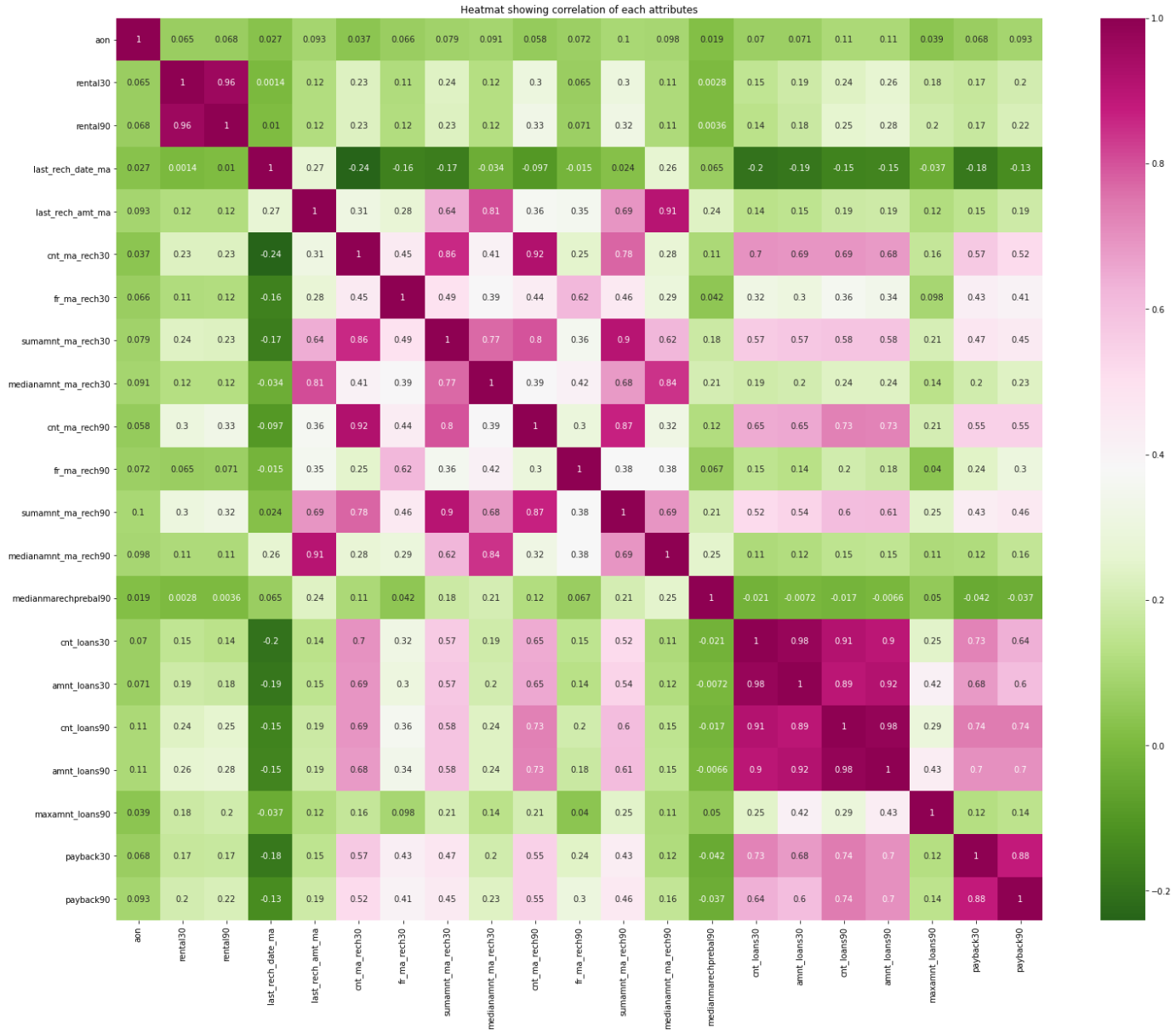
- Skewness is present in all the columns.
- Data in all the columns are widely spread.
- In most of the column’s data is concentrated at some particular values.

Boxplot



In most of the column's outliers are present.

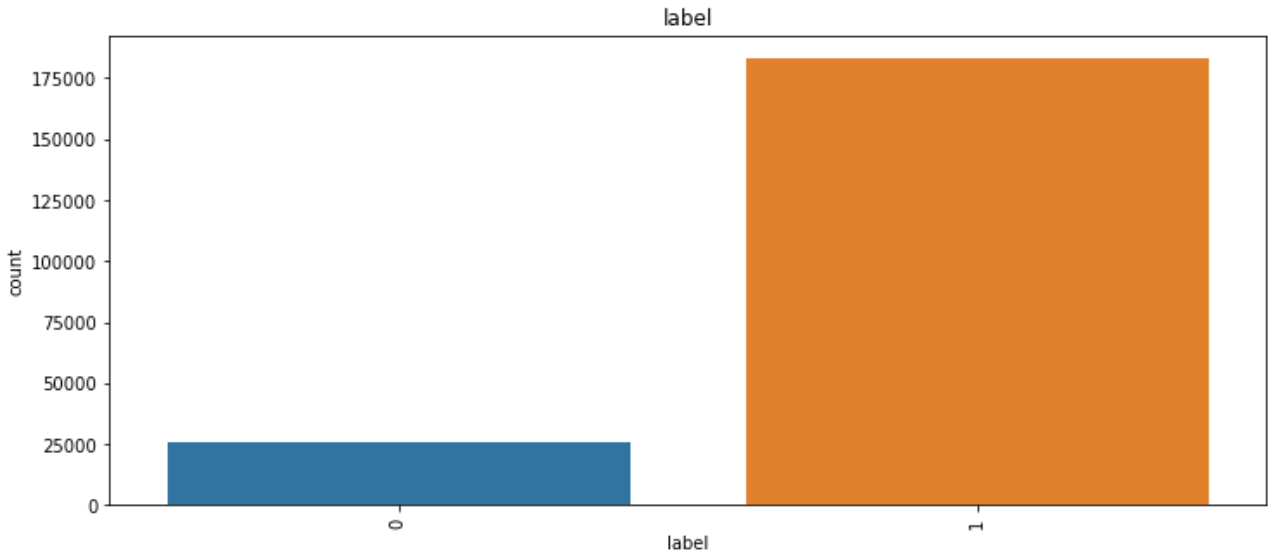
Heatmap



Observations:

- There are many multicollinearity issues in the dataset.
- amnt_loans90,cnt_loans90,amnt_loans30,sumamnt_ma_rech30 columns have high multicollinearity issues.
- We can understand that these values have less importance towards the model inference. Let us proceed by dropping these columns from the dataset.

2. Visualization of categorical features with target:



Observations:

There are 87.5% of non-defaulters and 12.5% of defaulter customers, the data is unbalanced we will use random oversampling technique to balance the target.

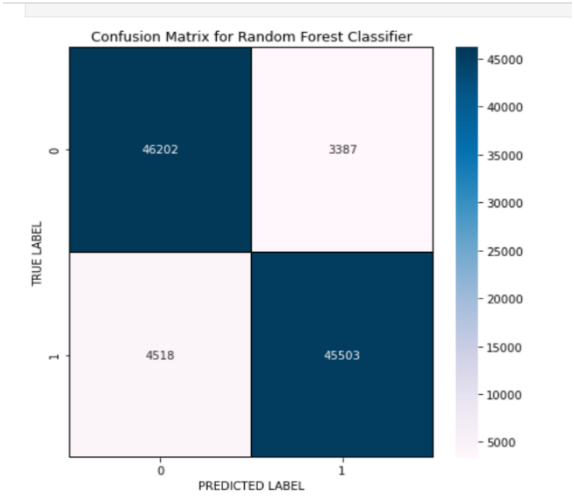
- **Run and Evaluate selected models**

1.Model building:

i) RandomForestClassifier:

```
RFC=RandomForestClassifier()
RFC.fit(X_train,y_train)
predrf=RFC.predict(X_test)
print('Accuracy Score:',accuracy_score(y_test, predrf))
print('Confusion Matrix:',confusion_matrix(y_test, predrf))
print(classification_report(y_test,predrf))
```

```
Accuracy Score: 0.9206404979419737
Confusion Matrix: [[46202  3387]
 [ 4518 45503]]
              precision    recall  f1-score   support

           0       0.91      0.93      0.92     49589
           1       0.93      0.91      0.92     50021

    accuracy                           0.92     99610
   macro avg       0.92      0.92      0.92     99610
weighted avg       0.92      0.92      0.92     99610
```

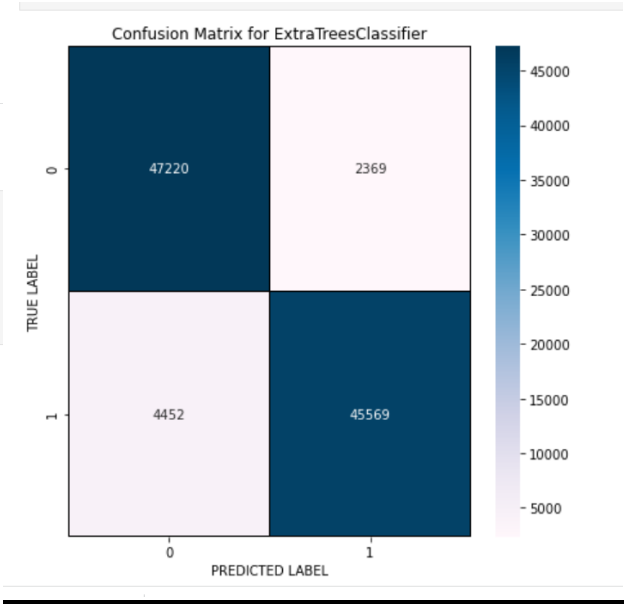


- RandomForestClassifier has given me 92.06% accuracy but still we have to look into multiple models.

ii) ExtraTreeClassifier:

```
ETC=ExtraTreesClassifier()
ETC.fit(X_train,y_train)
predet=ETC.predict(X_test)
print('Accuracy Score:',accuracy_score(y_test, predet))
print('Confusion Matrix:',confusion_matrix(y_test, predet))
print(classification_report(y_test,predet))
```

```
Accuracy Score: 0.9315229394639093
Confusion Matrix: [[47220  2369]
 [ 4452 45569]]
              precision    recall  f1-score   support

           0       0.91      0.95      0.93     49589
           1       0.95      0.91      0.93     50021

    accuracy                           0.93     99610
   macro avg       0.93      0.93      0.93     99610
weighted avg       0.93      0.93      0.93     99610
```

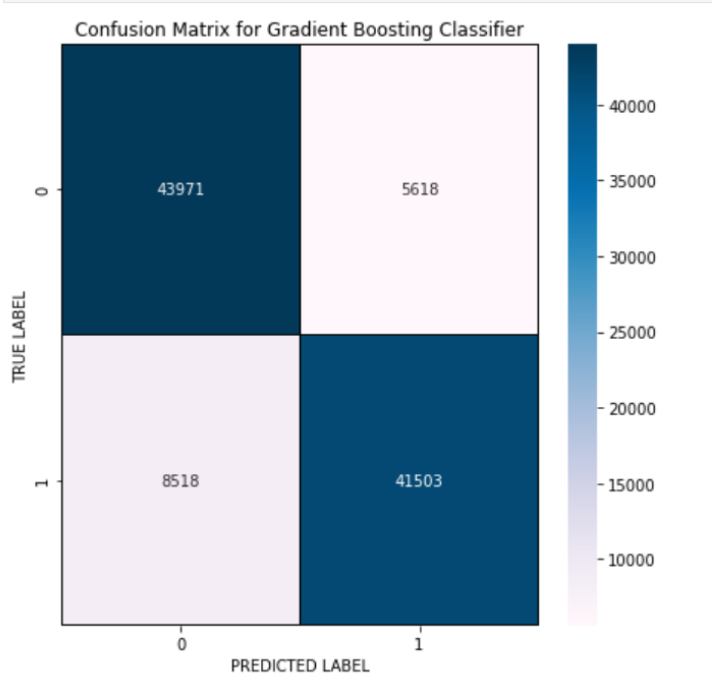


- ExtraTreeClassifier is giving me 93.15% accuracy.

iii) Gradient Boosting Classifier:

```
GBC=GradientBoostingClassifier()
GBC.fit(X_train,y_train)
predgb=GBC.predict(X_test)
print('Accuracy Score:',accuracy_score(y_test, predgb))
print('Confusion Matrix:',confusion_matrix(y_test, predgb))
print(classification_report(y_test,predgb))
```

```
Accuracy Score: 0.8580865374962353
Confusion Matrix: [[43971  5618]
 [ 8518 41503]]
              precision    recall  f1-score   support

           0       0.84      0.89      0.86     49589
           1       0.88      0.83      0.85     50021

    accuracy                           0.86     99610
   macro avg       0.86      0.86      0.86     99610
weighted avg       0.86      0.86      0.86     99610
```



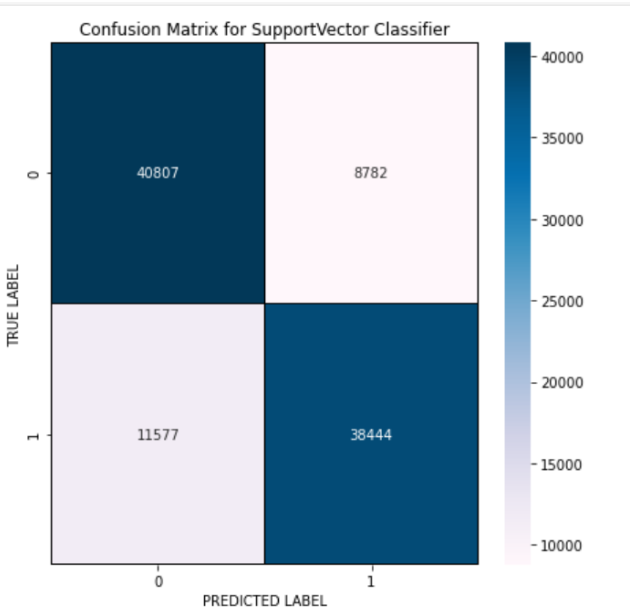
- GradientBoostingClassifier is giving me 85.8% accuracy.

iv) SupportVectorClassifier:

In [218…

```
SV=SVC()
SV.fit(X_train,y_train)
predsv=SV.predict(X_test)
print('Accuracy Score:',accuracy_score(y_test, predsv))
print('Confusion Matrix:',confusion_matrix(y_test, predsv))
print(classification_report(y_test,predsv))
```

```
Accuracy Score: 0.7956128902720611
Confusion Matrix: [[40807  8782]
 [11577 38444]]
              precision    recall  f1-score   support

           0       0.78      0.82      0.80     49589
           1       0.81      0.77      0.79     50021

    accuracy                           0.80     99610
   macro avg       0.80      0.80      0.80     99610
weighted avg       0.80      0.80      0.80     99610
```



- SupportVectorClassifier is giving me 75.56% accuracy.

  - By looking into the difference of model accuracy and cross validation score I found ExtraTreesRegressor as the best model.

2. Hyper Parameter Tunning:

26…

```
# Giving the parameters list for ETC model.
parameter = {'criterion':['gini','entropy'],
             'max_depth': [10,12,15,20,22],
             'n_estimators':[500,700,1000,1200],
             'max_features':['aoto','sqrt','log2'],
             'min_samples_split': [2]}
```

27…

```
GCV=GridSearchCV(ExtraTreesClassifier(),parameter,cv=5)
```

28…

```
GCV.fit(X_train,y_train)
```

28…

```
GridSearchCV(cv=5, estimator=ExtraTreesClassifier(),
             param_grid={'criterion': ['gini', 'entropy'],
                         'max_depth': [10, 12, 15, 20, 22],
                         'max_features': ['aoto', 'sqrt', 'log2'],
                         'min_samples_split': [2],
                         'n_estimators': [500, 700, 1000, 1200]})
```

29…

```
GCV.best_params_
```

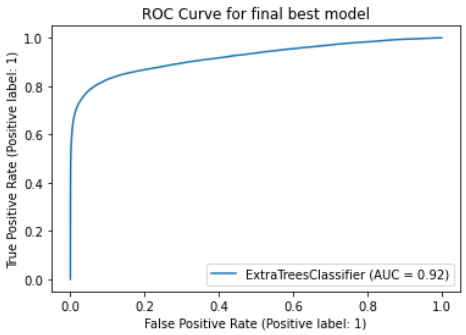
29…

```
{'criterion': 'gini',
 'max_depth': 22,
 'max_features': 'log2',
 'min_samples_split': 2,
 'n_estimators': 1000}
```

```
Final_mod=ExtraTreesClassifier(criterion='gini', max_depth=22,max_features='log2', min_samples_split=2, n_estimators=1000)
Final_mod.fit(X_train,y_train)
pred=Final_mod.predict(X_test)
acc=accuracy_score(y_test, pred)

print('Accuracy Score:',(accuracy_score(y_test,pred)*100))
print('Confusion matrix:',confusion_matrix(y_test,pred))
print(classification_report(y_test,pred))
```

```
Accuracy Score: 86.52042967573537
Confusion matrix: [[45224  4365]
 [ 9062 40959]]
              precision    recall  f1-score   support

           0       0.83      0.91      0.87     49589
           1       0.90      0.82      0.86     50021

    accuracy                           0.87     99610
   macro avg       0.87      0.87      0.86     99610
weighted avg       0.87      0.87      0.86     99610
```

```
from sklearn.metrics import accuracy_score,plot_roc_curve
#Ploting ROC curve for final best model
plot_roc_curve(Final_mod, X_test, y_test)
plt.title('ROC Curve for final best model')
plt.show()
```



- I have chosen all parameters of ExtraTreesClassifier, after tunning the model with best parameters I got 92% accuracy. Also, AUC and ROC values has increased the accuracy of the model.

3. Saving the model :

- I have saved my best model using .pkl as follows.

```
#Saving the model as .pkl file
import joblib
joblib.dump(Final_mod,"micro_credit_defaulters.pkl")
```

- **Interpretation of the Results:**

  We have trained several models above for the dataset we had prepared, and we got different results for the different algorithms. **Random Forest classifier** model gave us 92% accuracy and cross validation score of 96.99%. **ExtraTree classifier model** gave us accuracy and cv scores of 93.1% and 93.4 %. **Gradient Boosting classifier** has given us 85.9% and 85.3% of accuracy and cv score for the test dataset. **Support Vector classifier** has given us 75.5% and 79.5% of accuracy and cv score for the test dataset. The code prints out the number of false positives it detected and compares it with the actual values. This is used to calculate the accuracy score and precision of the algorithms. These results along with the classification report for each algorithm are given in the output as follows, where class 0 means the transaction was determined to be valid and 1 means it was determined as a fraud transaction. This result matched the class values to check for false positives.

4 CONCLUSION

- **Key Findings and Conclusions of the Study:**

  1) Around 28% users are highly defaulters with a mostly negative or null balance.

  2) Users with high equilibrium and a much lower number are defaulter.

  3) Nonstandard loans (i.e. 98 percent of the category) are paid to users who take up more loans as they pay back the loan within 5 days.

  4) 10% to 12% of users are defaulters in the Average and Low Balance categories.

  5) Non-defaulting users who have taken no loans.

  6) Around 97% users are taking large loans which fall into non-default categories.

  7) Defaulters include 40 percent of the users that do not have a single recharge in 3 months.

  8) Around 14 percent of users fall into the category of defaulting loans, on average.

9) The default is only 40% of users who do not reload in 90 days.

10) Users who recharge very high pay their loans on time. That is, 98% of them are non-defaulting ones.

11) defaulting is 34 percent of users who reload less.

12) Old and largely non default users are trusted

13) 17% of users receiving small loans are non-performing.

14) The new users constitute 32% of the users defaulting.

15) Of users who recharge and pay their loans on time, 99 percent are more in number, which is good news for the company than for any other category.

- **Learning Outcomes of the Study in respect of Data Science:**

1.Data Exploration and Cleaning, on data exploration, I found that the dataset was imbalanced for the target feature (87.5% for non-defaulters and 12.5% for Defaulters). Also, I found that the data had some very unrealistic values such as 999860 days which is not possible. Also, there were negative values for variables that must not have one (for example, frequency, amount of recharge, etc). All these unrealistic values were imputed which caused data to stabilize.
2. Feature Selection: there were 36 features, many of which I suspected were redundant because of the data duplication. It was imperative to select only the most significant of them to make ML models more efficient and cost-effective. The method used was 'Univariate Selection' using a chi-square test. I selected the top 20 features which were highly significant.
3. Data Visualization: On visualizing data, there were two important insights I gathered.
a. Imbalance of data
b. Distribution was not normal
4. Data Standardization: Since the data was not normal, all the features except the target variable which was dichotomous (Values ‘1’ and ‘0’).
5. Oversampling of Minority Class: The data was expensive, I did not want to lose out on data by under-sampling the majority class. Instead, I decided to
oversample the minority class using Random Oversampling.
6. Build Models: It was a supervised classification problem, I built 4 models

to
evaluate the performance of each of them:
a. Random Forest Classifier
b. ExtraTree Classifier
c. GradientBoosting Classifier
d. SupportVectorClassifier
The data was imbalanced, accuracy was not the correct performance metric. Instead, I focused on other metrics like precision, recall and ROC-AUC Curve.

- **Limitations of this work and Scope for Future Work:**

The data set consists of a large number of outliers which hinders the performance of machine learning models. Unless we solve the outlier problems, we are not reaching the best model accuracy. One can focus on the collection of real time customer-oriented data which can be useful for EDA. And more infercan be provided based on the analysis.

5 REFERENCE

fliprobo