MICRO CREDIT DEFAULTER PROJECT

Presented By:

Akshay Dinesh Shah

ACKNOWLEDGMENT

I would like to express my gratitude to my guide Srishti Maan (SME, Flip Robo) for giving me this opportunity. I would like to thank FlipRobo for giving me this opportunity to develop and accomplish this project. I would like to express my special thanks of gratitude to the sources Analytics Vidhya, GreeksofGreeks.

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. The Microfinance services (MFS) provided by MFI are Group Loans, Agricultural Loans, Individual Business Loans and so on. Many microfinance institutions (MFI), experts and donors are supporting the idea of using mobile financial services (MFS) which they feel are more convenient and efficient, and cost saving, than the traditional high- touch model used since long for the purpose of delivering microfinance services. Though, the MFI industry is primarily focusing on low income families and are very useful in such areas, the implementation of MFS has been uneven with both significant challenges and successes. Today, microfinance is widely accepted as a poverty-reduction tool, representing

$70 billion in outstanding loans and a global outreach of 200 million clients.

We are working with one such client that is in Telecom Industry. They are a fixed wireless telecommunications network provider. They have launched various products and have developed its business and organization based on the budget operator model, offering better products at Lower Prices to all value conscious customers through a strategy of disruptive innovation that focuses on the subscriber. They understand the importance of communication and how it affects a person's life, thus, focusing on providing their services and products to low income families and poor customers that can help them in the need of hour. They are collaborating with an MFI to provide micro-credit on mobile balances to be paid back in 5 days. The Consumer is believed to be defaulter if he deviates from the path of paying back the loaned amount within the time duration of 5 days. For the loan amount of 5 (in Indonesian Rupiah), payback amount should be 6 (in Indonesian Rupiah), while, for the loan amount of 10 (in Indonesian Rupiah), the payback amount should be 12 (in Indonesian Rupiah).

Background of the Domain Problem

The Micro Credit Defaulter Project is to predict the defaulter case if a customer will pay back the mobile balance credit within the due date or not? First of all we will understand the core concept of the use case and its motive.

In the Micro-Credit use case we have understood that a Micro Finance Institute (MFI) is collaborating with a Telecom industry to provide micro credit loans to those customers of that telecom industry who are actually low income customers and have very poor source of income.

Micro finance institutes targets the low income population specially the unbanked poor families living in remote areas. Because these micro finance institutions primarily focuses on low income population to provide them financial credits at very low rate, they are widely accepted as a poverty reduction tool, their global outreach is 200 million clients with $70 billion in total outstanding loans.

These micro finance institutes understood the importance of communication in modern era and thus wanted to improve the life of those poor low income population in terms of communication too, and hence they collaborated with a telecom industry who offers better products at lower price to all value.

The collaboration of these 2 organization come up with a concept of providing the micro – credits on mobile balance to be paid back in 5 days. For the loan amount of 5 Indonesian Rupiah the payback amount would be 6 Rupiah and the loan amount of 10 Rupiah the payback amount would be 12 Indonesian Rupiah. Now if a customer deviates from the path of paying back the loaned amount within the duration of 5 days will be consider as a defaulter.

Now one thing is sure in this business is if a customer defaults in paying back the loaned amount will not be eligible for further loan in future but the challenge is how to recognize if a customer is going to pay back the loaned amount or not? So that the organization could know it before & could prevent or reduce the default case in their business. To solve this we have to build a Machine learning model to predict the defaulter in order to improve the selection of the customers for the credit. The client had provided us the data from their database in which basic past history information of the services uses of the customer is available with the history of default case of the customer.

The data consist of 209590 customer's information of their telecom services uses. There are 36 features in the data which describes the various services uses of telecom services including the customer's mobile number & services of data usage call or other telecom facility usage. The data has some features like average daily balance, daily spent, last recharge, amount of last recharge, loan taken in last 30 or 90 days & many more such information which would be very useful in studying the pattern of customer's behavior and to analyze the default case.

Literature Review

The Micro Credit Defaulter Prediction Project is a machine learning project, where we have to research on a telecom industry data to build a machine learning model that can predict the defaulters who are not going to pay back the loaned amount. The data to build the machine learning model, comes from a telecom industry who collaborated with a Micro Financial Institute (MFI) to provide the mobile balance credit to their customers primarily the customers who are low income.

The research & Model building on this project has done in 7 important steps: -

1. Description of Data and Understanding.
2. Exploratory Data Analysis.
3. Data Pre-processing and PCA
4. Machine Learning Model Building & Metrics evaluation.
5. Cross – Validation of Machine Learning Model.
6. Hyper Parameter Tuning of Finalized model.
7. Model saving & conclusion.

1. Description of Data and Understanding

At very first step we have imported all the required libraries and packages in Python's Jupyter Notebook & we have imported the data set which was saved in our local system. After reading the data set we found that data consist of 209590 records of customers in 37 columns. We had a close look on the data set & deep understanding of each columns what it states. We found some columns are not useful in project and dropped them, we found no null values. We have sorted one object data type column.

2. Exploratory Data Analysis.

In EDA of the project first we took a close look on the target column Label understood the data distribution than we had a detailed Univariate Data Analysis of each column in detail, we found that data distribution in all the columns were not normal and they were heavily right skewed data having a lots of outliers & the removal of those outliers would lead to lots of data loss which we cannot afford, so we took necessary steps to resolve the problem in Pre-processing part. We also had a close look on the correlation of the features and target column in EDA part.

3. Data Pre-Processing.

In Pre-processing part we have tried to remove the skewness of the data with the help of Power_Transform yeo-johnson method, because we can't afford to remove all the outliers & in most of the cases we removed skewness significantly but not in all cases but it was good to move ahead. We rechecked the skewness by visualizing the transformed data again than we scaled the data with the help of MinMaxScaler but because we already had a transformation of the data during the skewness removal it did not do any changes in scaling part, we found best random state on Logistic Regression Model & Split the data into train & test data.

4. Model Building.

After completion of the Data Pre-processing part we had all the research on the data we took necessary steps to resolve the data problem according to our understanding of the data research and analysis & hence we got data ready for Model Building, here comes the machine learning model building part we decided to build 5 machine learning models on this project which were 1.) Logistic Regression 2.) K-Nearest Neighbours 3.) Decision Tree 4.) Random Forest 5.) Support Vector Machine. Wehave successfully build the models and evaluated the models on the basis of Accuracy Metrics. We found that all the models had given the accuracy percentage very close to each other from 88.34% to 92.28%. Random Forest model had the highest accuracy & Decision Tree had the lowest accuracy.

5. Cross-validation

After building successfully the Machine Learning Models with having a great accuracy it was the time to check if our models are victim of overfitting or underfitting. So to check this we have cross validated all of the models with scikit's inbuilt cross_val_score function with CV fold = 10. The result of cross validation was very impressive as we got a very minor differences between the original model accuracy score and cross validation score in all cases even we got greater cross validation score than the actual model accuracy score in Decision Tree Model's cross validation, which proved that our model was not an over fitted or under fitted model. Now we had to decide the best model for our micro-credit defaulter project & we had one model with highest accuracy Random Forest & one Model with least difference (difference in negative) between actual model accuracy and cross validation accuracy score, Decision Tree. Because we knew that none of our model is over fitted or under fitted model so why to choose decision tree having lowest accuracy score over Random Forest having highest accuracy score so to sort out this problem we did one more metrics evaluation of models by plotting the ROC Curve of all the models and we found that Random Forest model had the highest area under the curve hence we finalized the Random Forest Machine Learning Model as our Micro-Credit Defaulter Prediction Model.

6. Hyper Parameter Tuning

After finalizing the Random Forest Model we need to hyper tune the model to check if we could get the better accuracy then the default model or not so we set some useful parameters of Random Forest algorithm and trained it with Grid Search CV. This process was the most stressful process of the whole project I must say because this data set was very big data set having more 2 lacks record in one column and having 37 such column the overall count would be more than 8 million data & processing this much data in Random Forest Grid Search CV is not that easy to think the training of this model took approximately 42 hours in my local system and it was not able to train on Google Colab. Somehow we finished the Hyper tuning of the model, despite giving so much of time the accuracy was not up to the mark it had given even 1% lesser accuracy than the default model.

7. Model Saving & Conclusion.

After finishing the Hyper Parameter Tuning of the Random Forest model we decided to save the model with default parameters not with tuned parameters hence we saved the file with pickle in local system and also tested the saved model by loading it and predicting the test data with loaded model we concluded the result in data frame where we have shown the actual data and predicted data by loaded model. Here we finished our Micro-Credit Defaulter Prediction Project by building a Random Forest Model for it

.Motivation for the Problem Undertaken

Learning Data Science and to become Data scientist is the only motive to get hands on this project. Working on Micro-Credit Defaulter Project is a part of my Internship with FLIP ROBO TECHNOLOGY. This project is provided by the company to see my knowledge & Capability in the field of data science, hence to perform well in this project to show my capabilities is the motivation behind doing this project.

Mathematical/ Analytical Modelling of the Problem

To build a better machine learning model of predicting the defaulter case from the micro credit data set we encountered with several statistical modelling while doing data analysis & off course we needed mathematical modelling to build several machine learning models in this project.

Since the Micro credit data has the supervision of label column which were telling that if a user did default or not and we had to predict the default case using the same data hence we used Supervised Machine Learning modelling to build the 5 different models like: Logistic Regression which uses mathematics of probability in predicting the outcomes, knn algorithm which uses mathematics of Euclidean distance to know the distance between 2 data points, Decision Tree algorithm which uses different – different mathematical calculations in splitting the nodes etc.

We used Statistical modelling in the project to visualize the statistical data distribution in the column in data analysis, we used statistical modelling of correlations to know the inter relations between 2 features & and features vs target. We also used the statistical modelling in the metrics evaluation of machine learning models like confusion matrix to know the TP, TN, FP & FN, used confusion metrics to know the accuracy of the model precision, recall etc., ROC curve plotting to know the best model and things like that.

Data Sources and their formats

The Micro Credit Defaulter data is given to the FLIP ROBO TECHNOLOGY by their client of one of Telecom Industry which we mentioned earlier who collaborated with MFI. The data was originated by client's customer data base where all the data of their users is being recorded as per their services. Client had shared this data from their database directly to Flip Robo technology in order to predict the defaulter. The data was later saved and recorded in csv format by Flip Robo & provided to the interns to work upon. The Micro Credit Data consist of 209590 records of the users who are the subscriber of a particular Telecom industry. We have also been provided the data descriptions in a separate excel sheet where all the definition of 36 columns are described in detail to help in understand all the features what they are about.

```
df.head()
```

	Unnamed: 0	label	msisdn	aon	daily_decr30	daily_decr90	rental30	rental90	last_rech_date_ma	last_rech_date_da	...	maxamnt_loans30	medianam
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	

5 rows × 37 columns

Data Pre-processing

In order to process the data for machine learning models we have to perform some of data cleaning, not much data cleaning required in the data set as the data has no null values so we were sorted in terms of filling null values by feature engineering and all. But while data reading and understanding we found 1 unnamed column which was nothing but an index to the data set we found no use of that index as pandas provides us index for all the rows in data hence we dropped that, we also found one column with the name of pcircle in which only 1 unique value was present for all the rows and hence we assumed that it is going to same and neutral for all the rows & will not affect or play role in machine learning prediction hence we removed the column. We had a date column in the data set in YYYY-MM-DD format we separated year, month and date from the date column in 3 different column and dropped the original date column. We also dropped the msisdn column from the data set as it was the mobile number of the customer which was unique in every case and we also assumed that mobile number is not going to help or play any role no prediction hence we dropped this column as well.

After cleaning the data set we analyzed the pattern of data by data visualization and analysed that data distribution in each column were complicated had lots of skewness and outliers we assumed that outlier removal from the data set would lead to lots of data loss which we can't afford hence we performed the skewness removal by power_transfom yeo- johnson method in order to remove the skewness from the column and it did very well in most of the cases but not in all the cases. After that we scaled the data by MinMaxScaler but we found that because we already performed the power transformation in the data hence it did not changes in continuous features but changed some of categorical and date columns in the same category and the final step of data pre -processing we performed by splitting the data into train data & test data to send to machine learning model and test and to evaluate it.

Data Inputs- Logic- Output Relationships

The data inputs in the Micro-credit defaulter project are the data of a particular telecom industry's users or subscribers that what type of services user is subscribing for recharges of mobile balance users are doing, spending amounts in using service in a time frame of 30 & 90 days, maintaining the balance, how many credits a user opts for these type of data are present in the dataset. The format of data totally depends upon the type of column what it describing. In the data set there was 2 categorical column label & pcircle. Label was the column which was describing if a user had default in credits or not the format of the data was in integer in which 0 stands default and 1 stands for No default. & in pcircle was also a categorical data describing the circle of telecom and having only 1 category named with UPW and the data was in string format as it can be seen the category was in English language which can only be in string format. We had one more column in string format despite the data inside it was alphanumeric and this column was msisdn it was users contact number. Rest all of the data was either in integer format or in float format. The relationship between data input and its format was totally dependent what is column describing about like if there is a column stating something about money or time in days than its format was in integer format and there outputs were could be either in integer format of float both is possible like mean mauve of the whole data could be float of an integer data type data inputs but when the data type is in string format the output cannot be in integer or float format because we cannot apply mathematics in object data type that is why machine learning algorithms does not process the string format data inputs because there outputs cannot be defined which comes only after processing with some mathematics formulations and we need it to change in either integer data type of in float data type some time we have to encode the string data type which inputs are in string data type of any English like language and the encoding gives them shape of mathematical language which can be interpreted by the machines in data inputs to get outputs. We have done some statistical modelling of the data to understand the distribution of the data, since we used continuous data into it we had to use distribution plot in order to get the desire output. We could have used the count plot for the same but it wouldn't be readable so to understand the data pattern of a continuous data we must use distribution plot not count plot on the other we have used the visualization of some categorical data too with string data format and to see the data distribution in categorical data we had to use the count plot with string format in it as distribution plot will not perform the mathematical formulation it requires in order to produce the outputs hence we had to use the count plot, this is how the data inputs relations between its format affects the outputs.

Hardware and Software Requirements and Tools Used

The Micro-credit defaulter project is about to build a machine learning model that could predict the default case so that the company could decide to whom they should provide the loan amount and to whom they should not. Hence to build a machine learning model we used Python programming language to work upon this project. The whole project was performed on a Jupyter notebook which is available on Anaconda software. Some of the task was unable to perform in local system hence we also used Google Colab to work on this project.

We required a lot of libraries and it's packages to work upon this project. The first and very basic libraries to read the data and to perform all the work upon the data set we used Pandas library and to perform some mathematical operations on the data set we used Numpy library.

In EDA part of the project to perform data visualization we used Matplotlib library's Pyplot package & Seaborn library.

For Pre-processing, model building, model evaluations we used Scikit library's multiple packages & finally we used Pickle to save the model

Model/s Development and Evaluation

Identification of possible problem-solving approaches (methods)

During the statistical analyzation of the data distribution we found data was heavily skewed data and had lots of outliers and we solved it by applying power transformation technique of Scikit learn on the data and we were succeeded significantly in doing so. We also found some not useful column in the data set and we removed them with the help of pandas. We had sorted the Date column with the help of pandas. We had used ROC plot to evaluate the best model and its selection. To Hyper tune the model we used scikit learn’s Research method.

Testing of Identified Approaches (Algorithms)

As per the Micro Credit default use case demanded the prediction of the default case, we analysed the data and found that the problem is of Supervised Machine Learning Classification problem. Hence we decided to use the following algorithms to build the model for the use case:

- Logistic Regression.
- K-Nearest Neighbours Classification.
- Decision Tree Classifier.
- Random Forest Classifier.
- Select Vector Machine Classifier.

All the above mentioned algorithms have been used to train and test the data and we evaluated these models based evaluation metrics and found all of the above models have performed significantly well and have given the accuracy very close to each other, we further evaluated the models by cross validation and Roc plot to choose the best model from it.

Run and Evaluate selected models

Logistic Regression

After identifying the use case problem we identified that Logistic Regression would be one of important algorithm that can give us a good model having good accuracy in binary classification problems because Logistic Regression works very well in binary classification problem, hence we decided to use Logistic Regression for our project.

Logistic Regression

```
lr = LogisticRegression()
lr.fit(X_train,y_train)
```

```
LogisticRegression()
```

```
lr.score(X_train,y_train)
```

```
0.879206585493085
```

```
pred_lr = lr.predict(X_test)
```

```
print('Confusion Matrix for Logistic Regression Model is:\n\n',confusion_matrix(y_test,pred_lr),'\n\n')

print('Accuracy Score for Logistic Regression Model is :', accuracy_score(y_test,pred_lr),'\n\n')

print('Classification Report for the Logistic Regression :\n\n', classification_report(y_test,pred_lr),'\n\n')
```

```
Confusion Matrix for Logistic Regression Model is:

 [[  811  5614]
 [  433 45541]]


Accuracy Score for Logistic Regression Model is : 0.8845970342945476


Classification Report for the Logistic Regression :

              precision    recall  f1-score   support

           0       0.65      0.13      0.21      6425
           1       0.89      0.99      0.94     45974

    accuracy                           0.88     52399
   macro avg       0.77      0.56      0.57     52399
weighted avg       0.86      0.88      0.85     52399
```

In above snapshot of the Logistic Regression you can see that we have trained and tested the dataset in Logistic Regression algorithm and the algorithm have given a **88.45%** accuracy score but very poor recall & f1 score.

K-Nearest Neighbours Classifier

After seeing the result of Logistic Regression we decided to use Knn algorithm for our next model because Knn Model use to measure the distance between each data points using Euclidean distance and hence it gives better accuracy because it counts on each data points locations and predicts accordingly.

KNeighborsClassifier

```
knn = KNeighborsClassifier()
knn.fit(X_train,y_train)
```

```
KNeighborsClassifier()
```

```
knn.score(X_train,y_train)
```

```
0.9228278433019071
```

```
pred_knn = knn.predict(X_test)
```

```
print('Confusion Matrix for Logistic Regression Model is:\n\n',confusion_matrix(y_test,pred_knn),'\n\n')

print('Accuracy Score for Logistic Regression Model is :', accuracy_score(y_test,pred_knn),'\n\n')

print('Classification Report for the Logistic Regression :\n\n', classification_report(y_test,pred_knn),'\n\n')
```

```
Confusion Matrix for Logistic Regression Model is:

 [[ 2955  3470]
 [ 1778 44196]]


Accuracy Score for Logistic Regression Model is : 0.899845416897269


Classification Report for the Logistic Regression :

              precision    recall  f1-score   support

           0       0.62      0.46      0.53      6425
           1       0.93      0.96      0.94     45974

    accuracy                           0.90     52399
   macro avg       0.78      0.71      0.74     52399
weighted avg       0.89      0.90      0.89     52399
```

In above snapshot of the Knn model you can see that we have trained & tested the data set with Knn model and got the **89.98%** of accuracy which is greater than Logistic Regression and it recall and f1 score is also far better than the Logistic Regression.

Decision Tree Classifier

We also used the Decision Tree Classifier algorithm for our model because Decision Tree is a very good algorithm in terms of classification because it use to split the positive and negative data points into a tree by making multiple branches which is very beneficial in predicting the similar kind of data points by putting it into the branch it created.

DecisionTreeClassifier

```
dtc = DecisionTreeClassifier()
dtc.fit(X_train,y_train)
```

```
DecisionTreeClassifier()
```

```
dtc.score(X_train,y_train)
```

```
0.9999936384340369
```

```
pred_dtc = dtc.predict(X_test)
```

```
print('Confusion Matrix for Logistic Regression Model is:\n\n',confusion_matrix(y_test,pred_dtc),'\n\n')

print('Accuracy Score for Logistic Regression Model is :', accuracy_score(y_test,pred_dtc),'\n\n')

print('Classification Report for the Logistic Regression :\n\n', classification_report(y_test,pred_dtc),'\n\n')
```

```
Confusion Matrix for Logistic Regression Model is:

 [[ 3621  2804]
 [ 3281 42693]]


Accuracy Score for Logistic Regression Model is : 0.8838718296150689


Classification Report for the Logistic Regression :

              precision    recall  f1-score   support

           0       0.52      0.56      0.54      6425
           1       0.94      0.93      0.93     45974

    accuracy                           0.88     52399
   macro avg       0.73      0.75      0.74     52399
weighted avg       0.89      0.88      0.89     52399
```

In above snapshot you can see we have trained and tested the data in Decision tree algorithm and it has given **the 88.38%** of accuracy also I has given the precision of **0.52, recall of 0.56 & f1 score of 0.54**. Its precision is very less compare to Logistic Regression and Knn Model but recall and f1 score is greater than both of above models.

Random Forest Classifier

After applying the Decision Tree algorithm we got good accuracy and also better recall & precision so we decided to use Random Forest model because Random Forest algorithm uses modelling of multiple decision tree algorithm by taking samples of features and rows with replacement and it puts the test

data into each model and gives the output as most of the DT models gave similar kind of output hence it is very effective algorithm in terms of predicting the accurate result.

RandomForestClassifier

```
rf = RandomForestClassifier()
rf.fit(X_train,y_train)
```

```
RandomForestClassifier()
```

```
rf.score(X_train,y_train)
```

```
0.9999872768680739
```

```
pred_rf = rf.predict(X_test)
```

```
print('Confusion Matrix for Logistic Regression Model is:\n\n',confusion_matrix(y_test,pred_rf),'\n\n')

print('Accuracy Score for Logistic Regression Model is :', accuracy_score(y_test,pred_rf),'\n\n')

print('Classification Report for the Logistic Regression :\n\n', classification_report(y_test,pred_rf),'\n\n')
```

```
Confusion Matrix for Logistic Regression Model is:

 [[ 3345  3080]
 [  997 44977]]


Accuracy Score for Logistic Regression Model is : 0.9221931716254127


Classification Report for the Logistic Regression :

              precision    recall  f1-score   support

           0       0.77      0.52      0.62      6425
           1       0.94      0.98      0.96     45974

    accuracy                           0.92     52399
   macro avg       0.85      0.75      0.79     52399
weighted avg       0.92      0.92      0.92     52399
```

In above snapshot you can see that we have trained and tested the data set into the Random Forest algorithm and got the very good accuracy of **92.28%** above all the previous models we have built. Also it has very good precision, recall & f1 score above all previous model only recall was little less than Decision tree but even though the performance was very good as expected.

Support Vector Machine Classification.

We also had trained and tested the data into Support Vector Machine algorithm because it is a very good algorithm for both linearly separable data & non-linearly separable data, it creates a hyperplane with marginal distance between positive and negative data points passing through the support vectors it uses SVM kernels to convert low dimension data into high dimension data in non-linearly separable data and then it puts the similar kind of data into any side of the hyperplane and predicts accordingly so it's very effective algorithm to apply in classification algorithm.

Support Vector Machine

```
sv = SVC()
sv.fit(X_train,y_train)
```

```
SVC()
```

```
sv.score(X_train,y_train)
```

```
0.8870631194574856
```

```
pred_sv = sv.predict(X_test)
```

```
print('Confusion Matrix for Logistic Regression Model is:\n\n',confusion_matrix(y_test,pred_sv),'\n\n')


print('Accuracy Score for Logistic Regression Model is :', accuracy_score(y_test,pred_sv),'\n\n')


print('Classification Report for the Logistic Regression :\n\n', classification_report(y_test,pred_sv),'\n\n')
```

```
Confusion Matrix for Logistic Regression Model is:

 [[ 1695  4730]
 [  916 45058]]


Accuracy Score for Logistic Regression Model is : 0.892249852096414


Classification Report for the Logistic Regression :

              precision    recall  f1-score   support

           0       0.65      0.26      0.38      6425
           1       0.90      0.98      0.94     45974

    accuracy                           0.89     52399
   macro avg       0.78      0.62      0.66     52399
weighted avg       0.87      0.89      0.87     52399
```

In the above snapshot you can see that we had trained and tested the data into Support Vector Classification Model & it gave the accuracy of **89.29%**. Its precision score was very good but recall & f1 score was not much good.

Key Metrics for success in solving problem under consideration

To evaluate the Machine Learning algorithms we mainly used almost all the classification metrics evaluation in this project. Our focus was on mainly accuracy score of the model, precision, recall, f1 score and the Roc-Auc curve of the all the models. Mainly focus was on accuracy score and we compared accuracy score with the cross validation score and prioritize the minimum difference model as a best fit model and later we evaluated the models with ROC Curve and whichever algorithm had the maximum area under the curve we finalized that model for our project.

Cross validation

Cross validation

LogisticRegression:

```
print(cross_val_score(lr,scaled_X,y,cv=5).mean())
```

```
0.8807355220741394
```

KNeighborsClassifier

```
print(cross_val_score(knn,scaled_X,y,cv=5).mean())
```

```
0.897988014120339
```

RandomForestClassifier

```
print(cross_val_score(rf,scaled_X,y,cv=5).mean())
```

```
0.9212187462596262
```

DecisionTreeClassifier

```
print(cross_val_score(dtc,scaled_X,y,cv=5).mean())
```

```
0.8841469022614268
```

Support Vector Machine

```
print(cross_val_score(sv,scaled_X,y,cv=5).mean())
```

```
0.8887462922129348
```

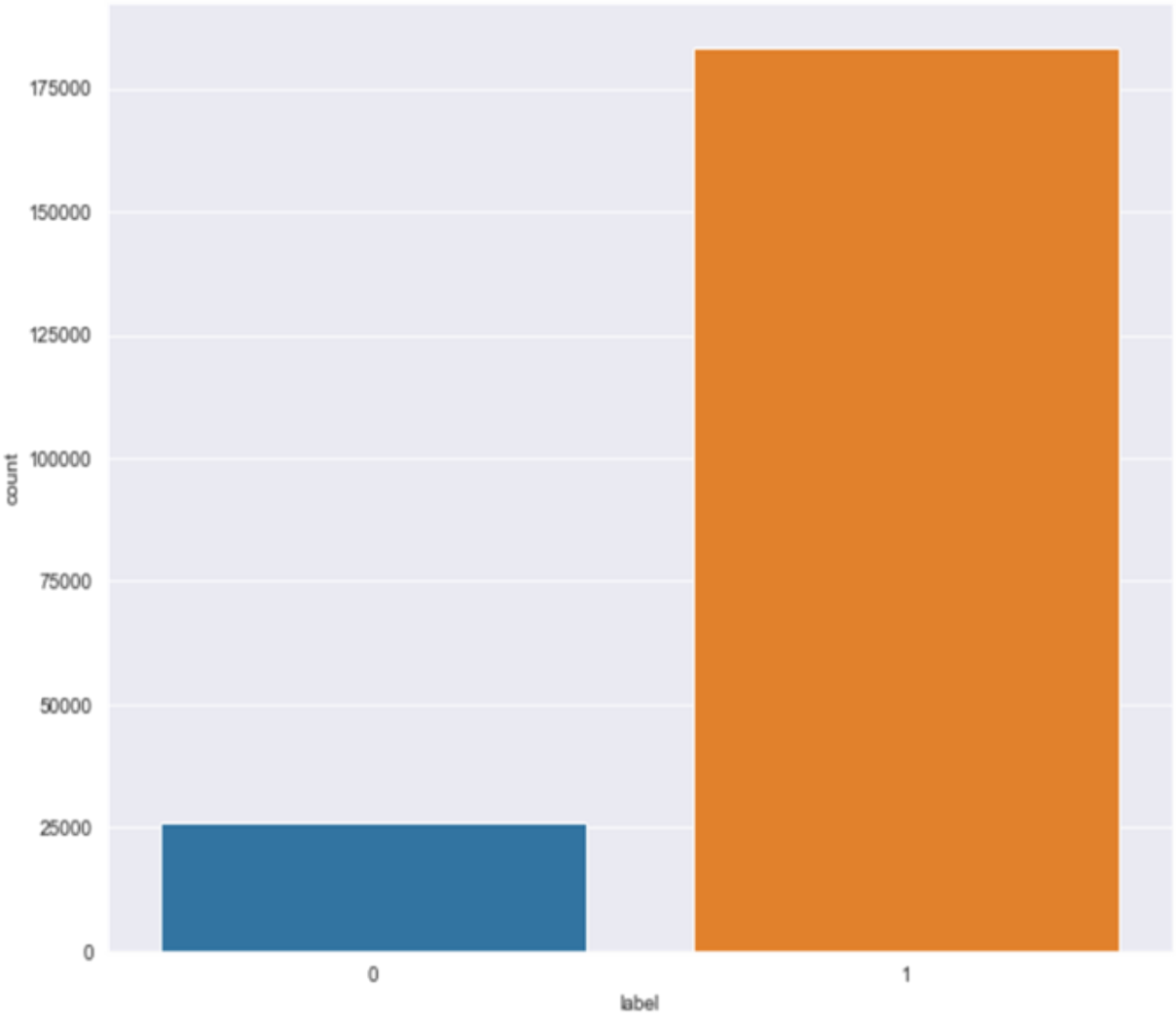
We can seen that Decision Tree model's cross validation accuracy has given greater accuracy than default model accuracy & After that **Random Forest Model** has very least difference between model's default accuracy & cross validation accuracy. So from here we have concluded that we have 2 best models for **Micro credit Defaulter.**

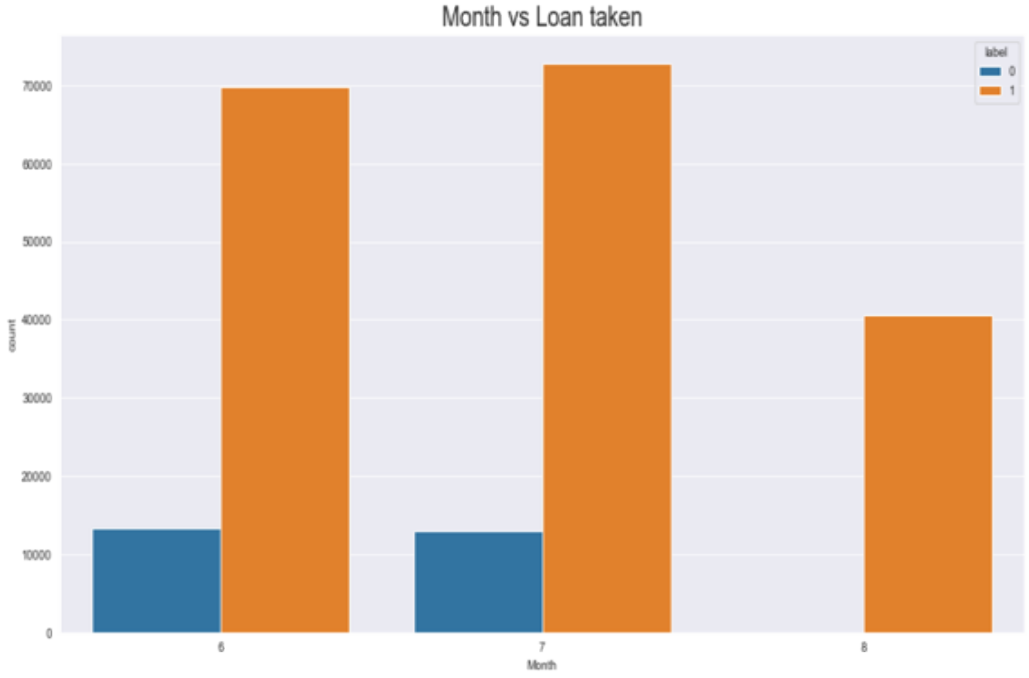
Models	Accuracy	Cross Validation Score
Logistic Regression	88.45%	88.07%
KNeighbours Classifier	89.98%	89.79%
Random Forest Classifier	92.21%	92.12%
Decision Tree Classifier	88.38%	88.41%
Support Vector Machine	89.22%	88.87%

Visualizations

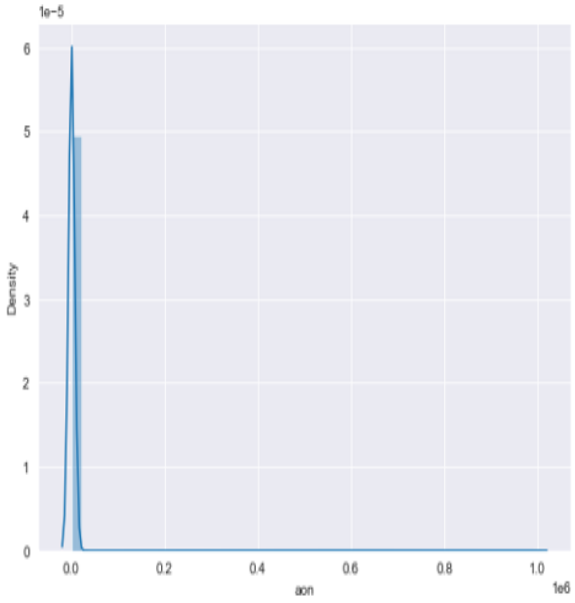
We used Matplotlib. Pyplot and Seaborn to visualize the & statistical analysis of the data from these 2 library's we mainly focused on data distribution in the columns to analyze the data and it's structure for further approaches and to know the data distribution in the columns we have 2 approaches to apply 1 is count plot which we use to visualize the categorical columns and 2nd is distribution plot which we use to visualize the continuous data since maximum of the columns were continuous data we plotted the distribution plot for all the continuous columns and we have observed that almost all the columns data distribution were highly messed up and highly skewed. Later we plot the boxplot to see the amount of outliers in the columns so that we can approach the appropriate method to resolve this problem.

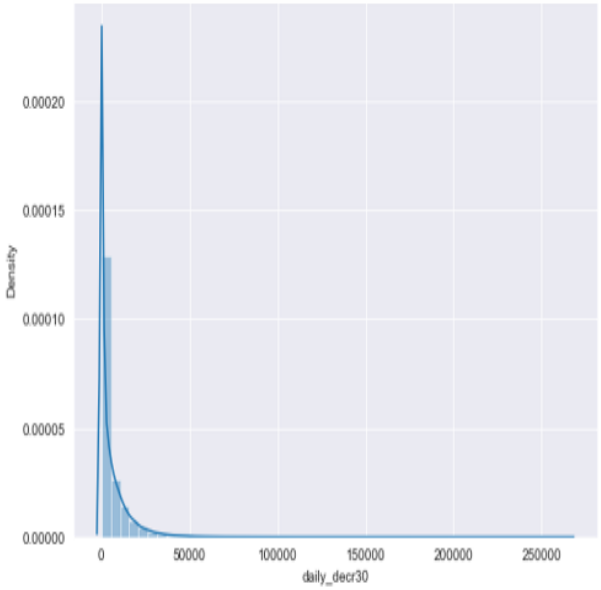
Analyzing the Target variable

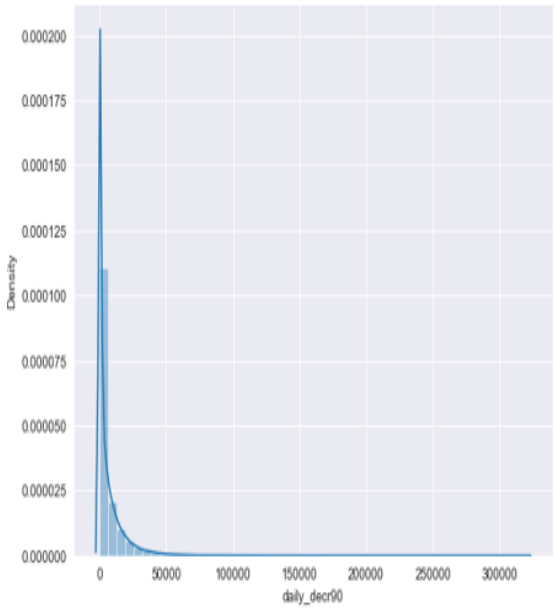


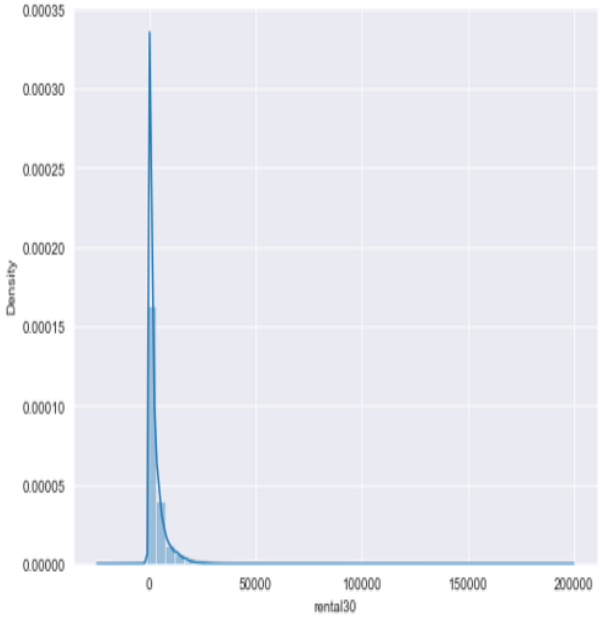


Data Distribution

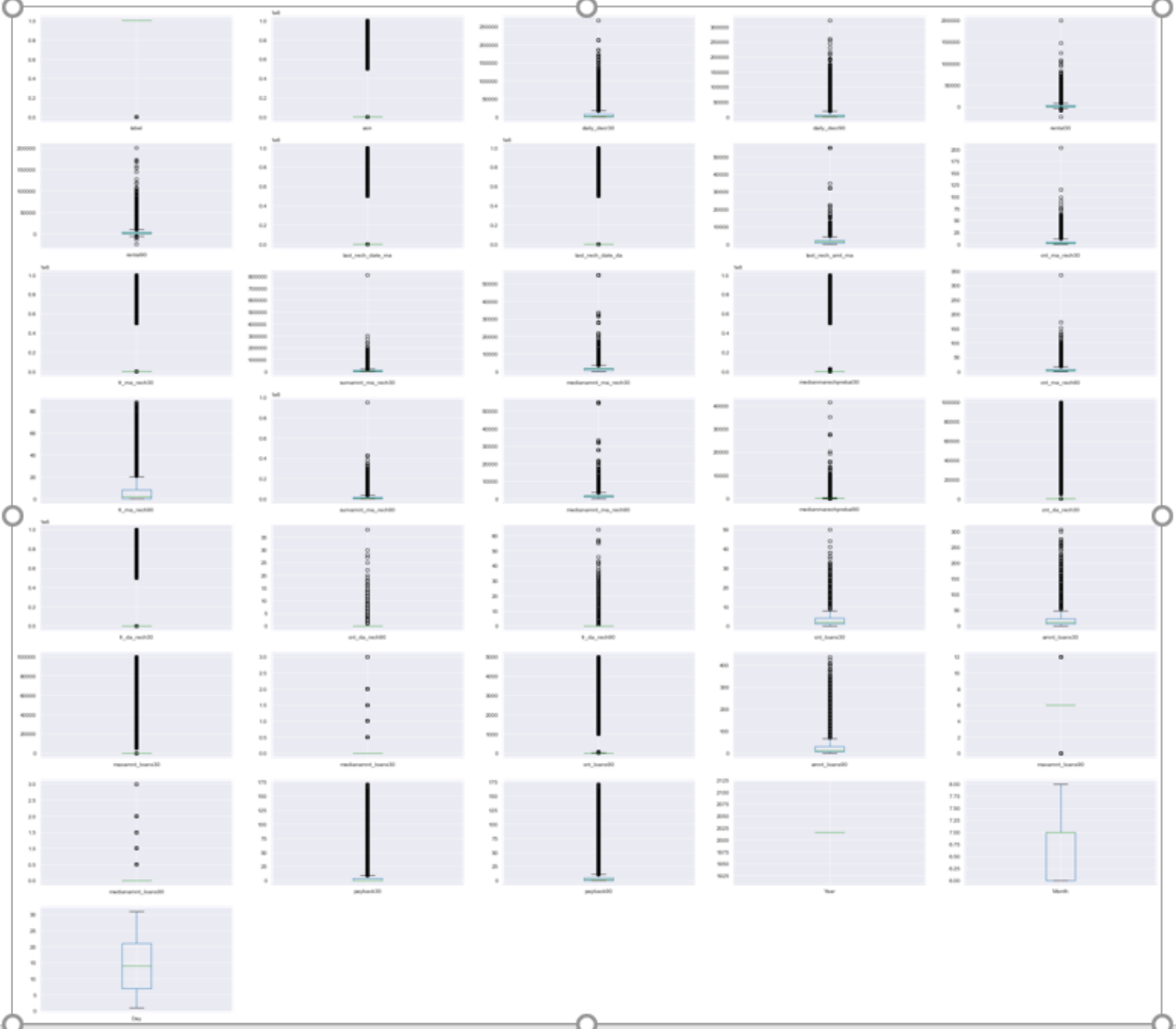




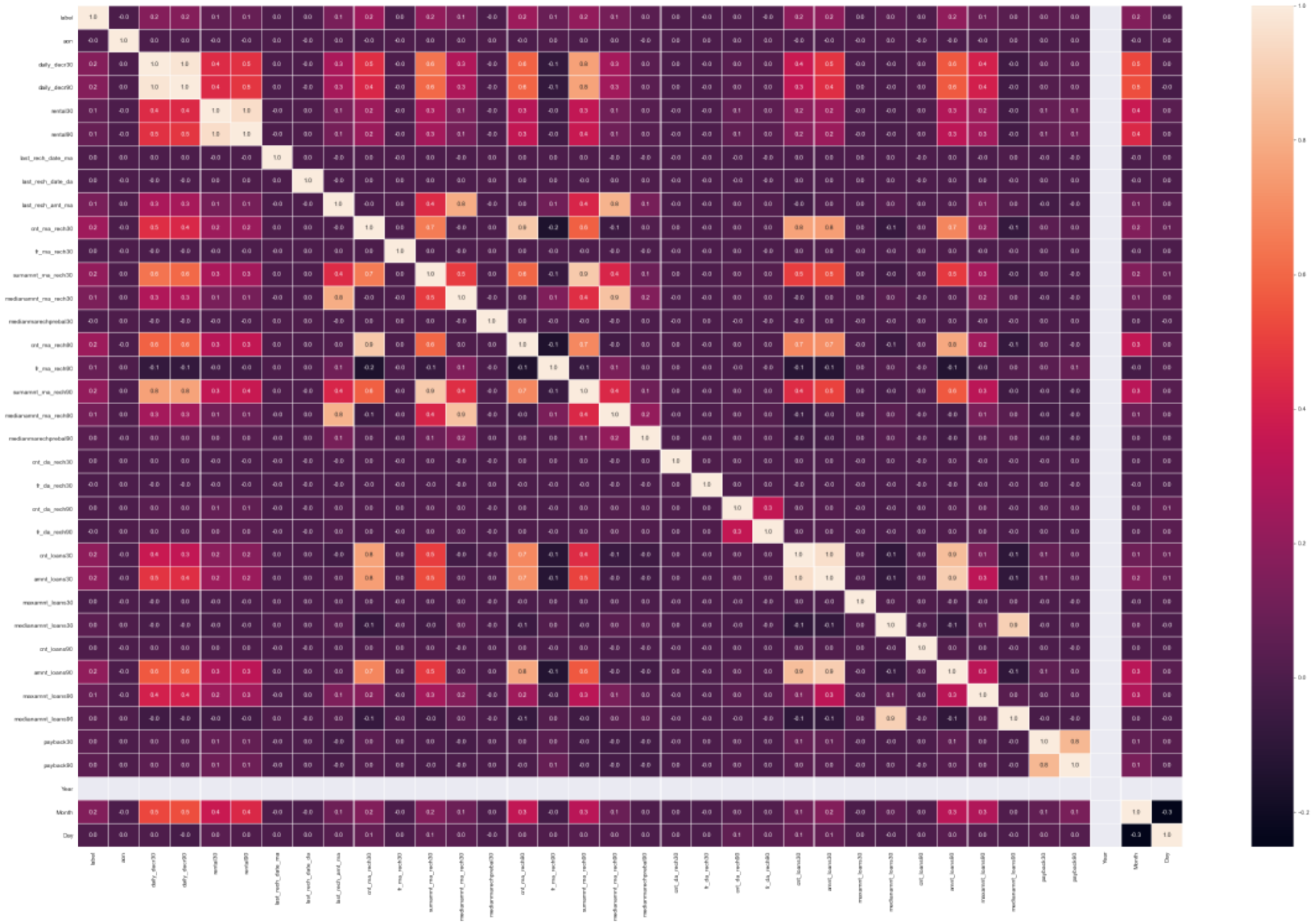




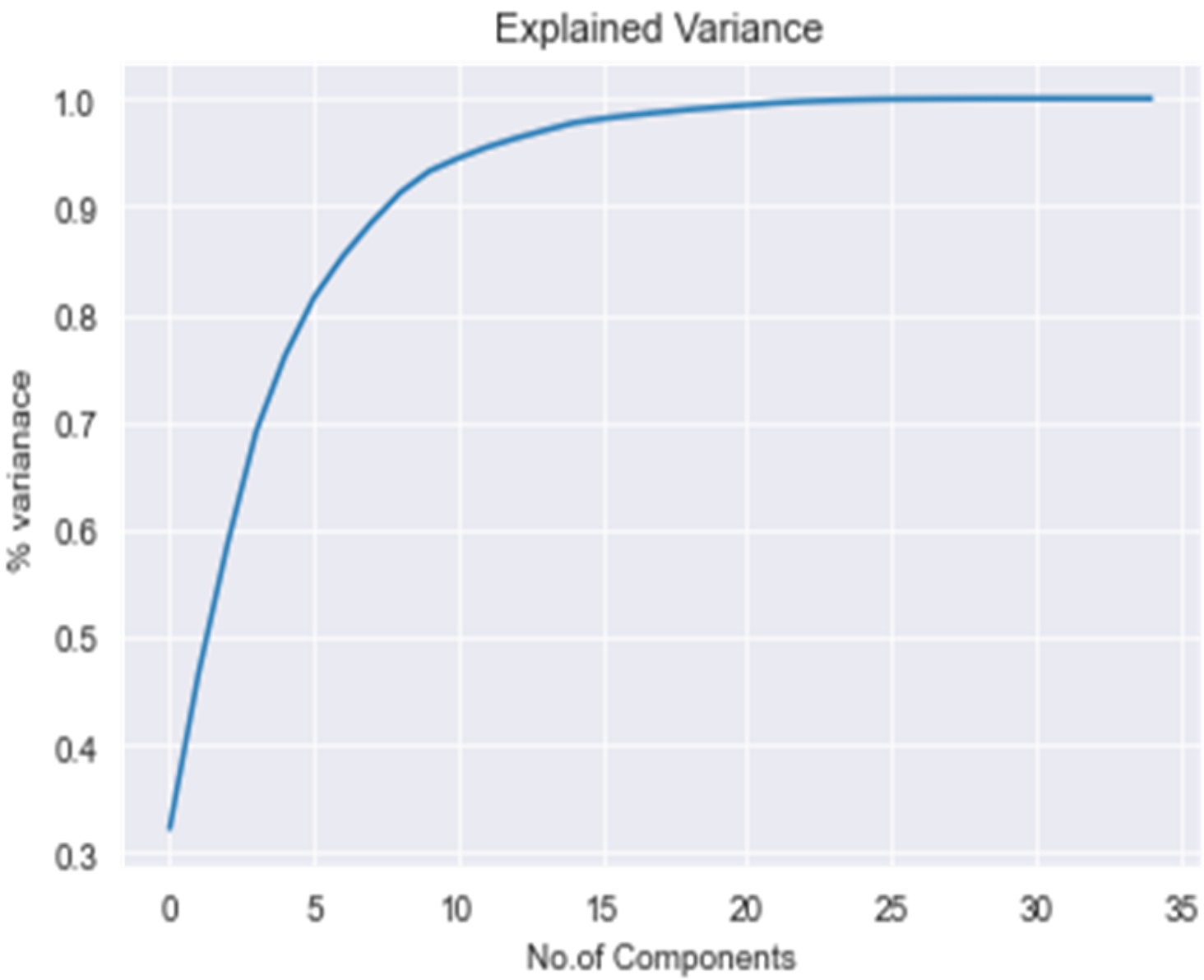
Outlier Detection



CORRELATION



Principle Component Analysis



Interpretation of the Results

From the Visualization of the data the results were interpreted that pcircle column has only 1 type of data so we interpreted no use of the neutral column, we interpreted that the Label column label was class imbalanced data, data distribution in all the continuous column were highly skewed and have lots of outliers.

In the pre-processing the skewness in the data set was detected and we interpreted that removing of outliers would lead to heavy data loss so we resolved it by the power transformation method, we used yeo-Johnson method because during the pre-processing the 0 was detected in most of the continuous column.

5 Algorithms were chosen based on the data analysis of the data to build the Machine Learning models based on the data structure and the nature of data distribution in the feature columns. After evaluating the models by metrics evaluators we interpreted all the model as very good hence we decided to choose the best model based on the minimum difference between model accuracy and the cross validation accuracy, but later we found a very much fluctuation in the precision, recall & f1 score so we decided to use Roc plot to find out the which model has maximum area under the curve and we got Random Forest Model as our best model.

Hyper Parameter Tuning

RandomForestClassifier

```
RFC = RandomForestClassifier()
```

```
parametrs = {'max_depth':[2,4,6,8],'max_features':['auto','sqrt'],'n_estimators':[10,20,30,40,50,60,70,80,90,100],'criterion' :[
```

```
rfc = GridSearchCV(RFC, parametrs, cv=5)
```

```
rfc.fit(X,y)
```

```
GridSearchCV(cv=5, estimator=RandomForestClassifier(),
             param_grid={'criterion': ['gini', 'entropy'],
                         'max_depth': [2, 4, 6, 8],
                         'max_features': ['auto', 'sqrt'],
                         'n_estimators': [10, 20, 30, 40, 50, 60, 70, 80, 90,
                                          100]})
```

```
rfc.best_params_
```

```
{'criterion': 'gini',
 'max_depth': 8,
 'max_features': 'sqrt',
 'n_estimators': 10}
```

```
model = RandomForestClassifier(n_estimators = 10, criterion = 'gini', max_depth = 8, max_features = 'sqrt')
model.fit(X_train,y_train)
pred_rf = model.predict(X_test)
print('Confusion Matrix for Logistic Regression Model is:\n\n',confusion_matrix(y_test,pred_rf),'\n\n')


print('Accuracy Score for Logistic Regression Model is :', accuracy_score(y_test,pred_rf),'\n\n')


print('Classification Report for the Logistic Regression :\n\n', classification_report(y_test,pred_rf),'\n\n')
```

```
Confusion Matrix for Logistic Regression Model is:

 [[ 2204  4221]
 [  408 45566]]


Accuracy Score for Logistic Regression Model is : 0.9116586194393023


Classification Report for the Logistic Regression :

              precision    recall  f1-score   support

           0       0.84      0.34      0.49      6425
           1       0.92      0.99      0.95     45974

    accuracy                           0.91     52399
   macro avg       0.88      0.67      0.72     52399
weighted avg       0.91      0.91      0.89     52399
```

CONCLUSION

Key Findings and Conclusions of the Study

From the whole problems of Micro Credit Defaulter data set the key finding were as follows:

1. Data set had 209590 records of telecom users with 36 types of the services related data were present in the data.
2. Dataset had 3 object data type columns, 12 integer data type columns & 21 float data type columns.
3. No missing values were in the data.
4. Mathematical description of the data was not so good, in almost all the columns the standard deviation was greater than the mean value.
5. Class data was imbalanced.
6. Data distribution was mostly heavily right skewed and had lots of outliers in it.
7. Random Forest model performed extremely well with the data set.
8. None of the model was over fit or under fit.

- From this dataset I get to know that each feature plays a very import role to understand the data. Data format plays a very important role in the visualization and Appling the models and algorithms.

- The power of visualization is helpful for the understanding of data into the graphical representation its help me to understand that what data is trying to say, Data cleaning is one of the most important steps to remove missing value or null value fill it by mean median or by mode or by 0.

- Various algorithms I used in this dataset and to get out best result and save that model. The best algorithm is Random Forest Classifier.

Interference from the key findings of the data set were taken as follows:

1. Some of the column were interfered as not useful.
2. Removal of the outliers were considered as data loss hence transformation of the data was preferred.
3. Based on EDA & the data structure 5 algorithms were selected to build a best machine learning model.
4. Based on metrics evaluation the best model was selected.

Learning Outcomes of the Study in respect of Data Science

In this project I learned most fundamentals of the machine learning like how a complex data set can be sorted out well, how visualization of the data makes you understand more and more about the data & the story of the data what it tells. I learned how to decide which column is not useful in terms of machine learning and how to treat them.

I learned about how different algorithms works differently on the same data set and gives result as per their capability. I found out that when the data distribution is complex and are not linearly separable in that case Random Forest & Support Vector Machine algorithms works best because their mathematical intuitions are very advanced.

I faced challenge in this project while Hyper parameter tuning of the models after the selection of best model. During the hyper parameter tuning of the Random Forest I provided very large parameters in GridSearchCV to select the best parameters in hope of getting even better results (accuracy score, recall, precision, f1 score, Area under the Roc curve) and we know that Hyper parameter tuning of random forest is complicated and time consuming from all the other model hence my model took more than 48 hours to produce the result. Hence I decided to minimize the parameters of Random Forest algorithm and trained the model but it doesn't gave me the better result than what comes with default parameters.

Limitations of this work and Scope for Future Work

The limitation of this work, what was done in this project is that we developed a financial institutions problem on the basis of accuracy score measuredly. The result which could be not up to the mark when it will be applied on real life scenario because false positive rate is high in this model and increase of false positive rate in a financial prediction could be not useful as we expected. To improve the result the industry should change in some of their services or management in order to get subscription of all type of services available with the industry so that data could be distributed in equal amount on each instance we have seen that a very large population opts for only one type of service and hence the 70 to 90 percent data even more distributes on a single point and on the other hand there are some high class people who goes up to higher level which makes the data messy and complicated which could be worse in predicting the desired result.

THANK YOU