FLIP ROBO

Micro Credit Loan Defaulter Project

Submitted by:

Arpan Pattanayak

ACKNOWLEDGMENT

I would like to express my sincere gratitude to my SME Mr. Shwetank Mishra from Flip Robo Technologies for letting me work on this project and providing me with all necessary information and dataset for the project. I would also like to thank my mentor of Data Trained academy for providing me sufficient knowledge so that I can complete the project.

I have utilized a few external resources that helped me to complete the project. I ensured that I learn from the samples and modify things according to my project requirement. All the external resources that were used in creating this project are listed below:

- https://www.google.com/
- https://www.youtube.com/
- https://scikit-learn.org/stable/user_guide.html
- https://github.com/
- https://www.analyticsvidhya.com/

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.

- Conceptual Background of the Domain Problem

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).

- Review of Literature

Microfinance refers to the financial services provided to low-income individuals or groups who are typically excluded from traditional banking. Most microfinance institutions focus on offering credit in the form of small working capital loans, sometimes called microloans or microcredit. However, many also provide insurance and money transfers, and regulated microfinance banks provide savings accounts.
Microfinance aims to improve financial services access for marginalized groups, especially women and the rural poor, to promote self-sufficiency.

Access to essential financial services can empower individuals economically and socially by creating self-reliance and economic sustainability in impoverished communities where salaried jobs are scarce. The benefits of microfinance include:

- Small loans enable entrepreneurs to start or expand micro, small and medium enterprises.
- Savings help families build assets to finance school fees, improve homes (e.g., install power or running water) and achieve goals.
- Insurance products can offset the cost of medical care.
- Money transfers and remittances allow families to easily send and receive money across borders.

Hundreds of millions of low-income people have benefited from microfinance since its inception, with about 140 million borrowers served by the industry worldwide annually.

- Motivation for the Problem Undertaken

Our main objective of doing this project is to build a model to predict whether the users are paying the loan within the due date or not. We are going to predict by using Machine Learning algorithms.
The sample data is provided to us from our client database. In order to improve the selection of customers for the credit, the client wants some predictions that could help them in further investment and improvement in selection of customers.

Analytical Problem Framing

- Mathematical/ Analytical Modelling of the Problem

We need to build a Machine Learning 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 the dataset, the Label '1' indicates that the loan has been paid i.e., non-defaulter, while Label '0' indicates that the loan has not been paid i.e., defaulter.
Clearly it is a binary classification problem where we need to use classification algorithms to predict the results. There were no null values in the dataset. There were some unwanted entries like more than 90% of zero values present in some of the columns which means these customers have no loan history so, I have dropped those columns. I found some negative values while summarizing the statistics of the dataset, I have converted them into positive. To get better insights on features I have used some plots like pie plot, count plot, bar plot, distribution plot, box plots etc. There were lots of skewness and outliers present in our dataset which need to be cleaned using appropriate techniques and balanced the data. At last, I have built many classification models to predict the defaulter level at the institution.

- Data Sources and their formats

The data which I received from the Flip Robo Technologies was in CSV (Comma Separated Values) format. In the dataset there were 209593 rows and 37 columns.

The data descriptions are as follow: -

Variable	***Definition***
label	Flag indicating whether the user paid back the credit amount within 5 days of issuing the loan{1:success, 0:failure}
msisdn	mobile number of user
aon	age on cellular network in days
daily_decr30	Daily amount spent from main account, averaged over last 30 days (in Indonesian Rupiah)
daily_decr90	Daily amount spent from main account, averaged over last 90 days (in Indonesian Rupiah)
rental30	Average main account balance over last 30 days
rental90	Average main account balance over last 90 days
last_rech_date_ma	Number of days till last recharge of main account
last_rech_date_da	Number of days till last recharge of data account
last_rech_amt_ma	Amount of last recharge of main account (in Indonesian Rupiah)
cnt_ma_rech30	Number of times main account got recharged in last

	30 days
fr_ma_rech30	Frequency of main account recharged in last 30 days
sumamnt_ma_rech30	Total amount of recharge in main account over last 30 days (in Indonesian Rupiah)
medianamnt_ma_rech30	Median of amount of recharges done in main account over last 30 days at user level (in Indonesian Rupiah)
medianmarechprebal30	Median of main account balance just before recharge in last 30 days at user level (in Indonesian Rupiah)
cnt_ma_rech90	Number of times main account got recharged in last 90 days
fr_ma_rech90	Frequency of main account recharged in last 90 days
sumamnt_ma_rech90	Total amount of recharge in main account over last 90 days (in Indonasian Rupiah)
medianamnt_ma_rech90	Median of amount of recharges done in main account over last 90 days at user level (in Indonasian Rupiah)
medianmarechprebal90	Median of main account balance just before recharge in last 90 days at user level (in Indonasian Rupiah)
cnt_da_rech30	Number of times data account got recharged in last 30 days
fr_da_rech30	Frequency of data account recharged in last 30 days
cnt_da_rech90	Number of times data

	account got recharged in last 90 days
fr_da_rech90	Frequency of data account recharged in last 90 days
cnt_loans30	Number of loans taken by user in last 30 days
amnt_loans30	Total amount of loans taken by user in last 30 days
maxamnt_loans30	maximum amount of loan taken by the user in last 30 days
medianamnt_loans30	Median of amounts of loan taken by the user in last 30 days
cnt_loans90	Number of loans taken by user in last 90 days
amnt_loans90	Total amount of loans taken by user in last 90 days
maxamnt_loans90	maximum amount of loan taken by the user in last 90 days
medianamnt_loans90	Median of amounts of loan taken by the user in last 90 days
payback30	Average payback time in days over last 30 days
payback90	Average payback time in days over last 90 days
pcircle	telecom circle
pdate	date

- Data Preprocessing Done

Data pre-processing is the process of converting raw data into a well-readable format to be used by Machine Learning model. Data pre-processing is an integral step in Machine Learning as the quality of data and

the useful information that can be derived from it directly affects the ability of our model to learn; therefore, it is extremely important that we pre-process our data before feeding it into our model. I have used following pre-processing steps:

- Importing necessary libraries and loading dataset as a data frame.
- Used pandas to set display maximum columns ensuring not to find any truncated information.
- Checked some statistical information like shape, number of unique values present, info, finding zero values etc.
- Checked for null values and did not find any null values.
- Dropped some unwanted columns like Unnamed:0, pcircle, msisdn as they are of no use for prediction.
- Dealt with zero values by verifying the percentage of zero values in each column and decided to discard the columns having more than 90% of zero values.
- Converted time variable “pdate” from object into datetime and extracted Day, Month and Year for better understanding. Checked value counts for each and dropped Year column as it contains unique value throughout the dataset.
- Checked unique values and value counts of target variable.
- Visualized each feature using seaborn and matplotlib libraries by plotting several categorical and numerical plots like pie plot, count plot, bar plot, distribution plot, box plots etc.
- Identified Outliers and removed the outliers using percentile method by setting data loss to 2% as using zscore and IQR method data loss was very high.
- Checked for skewness and removed skewness in numerical columns using power transformation method (yeo-johnson).
- Used Pearson’s correlation coefficient to check the correlation between label and features. 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 and feature scaling is performed using Standard Scalar method to avoid any kind of data biasness.

- Since the dataset was imbalanced. Label '1' had approximately 87.5% records, while label '0' had approximately 12.5% records. So, performed Oversampling method using SMOTE to balance the data.
- Checked for the best random state to be used on our Classification Machine Learning model pertaining to the feature importance details.
- Finally created classification model along with evaluation metrics.

Data Inputs- Logic- Output Relationships

The dataset consists of label and features. The features are independent, and label is dependent as the values of our independent variables changes as our label varies.

- Since we had only numerical columns so, I checked the distribution of skewness using dist plots.
- To analyse the relation between features and label I have used many plotting techniques where I found some of the columns having strong relation with label.
- The visualization helped me to understand that maximum distribution is for non-defaulter for all the features & maximum defaulter list are from people who have Average payback time in days over last 30 & 90 days, also frequency of recharge done in the main account since last 90 days. So, the features, which I have kept after dropping few are having relationship with the output.
- I have checked the correlation between the label and features using heat map and bar plot. Where I got the positive correlation between the label and features and there was not much relation.

State the set of assumptions (if any) related to the problem under consideration

The assumption part for me was relying strictly on the data provided to me. I have dropped the 2016 year from pdate columns because the data is from the year 2016, only the date and months are different. We separated months and days to different columns.

- Hardware and Software Requirements and Tools Used

Hardware Used:

1. Processor — core i5 and above
2. RAM — 8 GB or above
3. SSD — 250GB or above

Software Used:

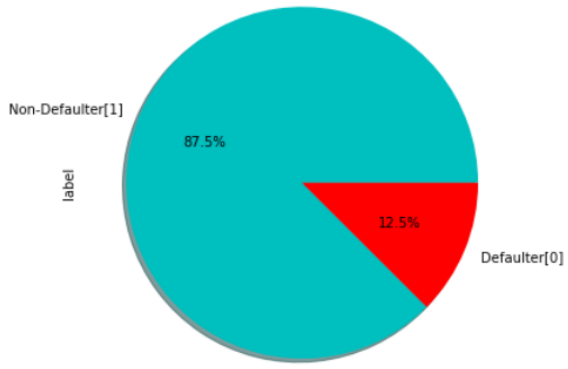
i. Programming language: Python
ii. Distribution: Anaconda Navigator
iii. Browser based language shell: Jupyter Notebook
Libraries/Packages Used:
Pandas, NumPy, matplotlib, seaborn, scikit-learn and pandas_profiling

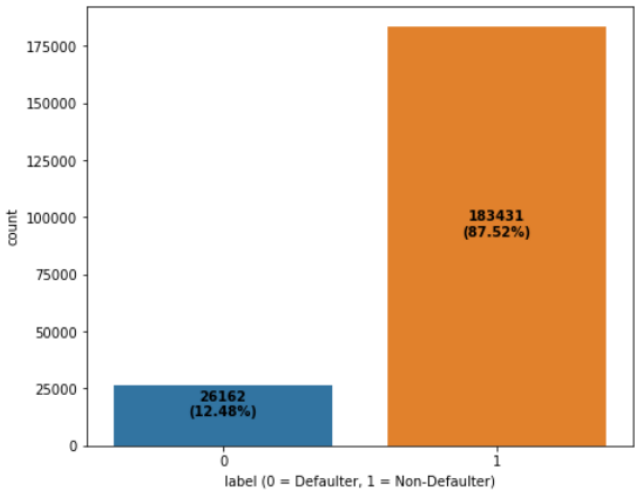
Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

I have used both statistical and analytical approaches to solve the problem which mainly includes the pre-processing of the data also used EDA techniques and heat map to check the correlation of independent and dependent features. Also, before building the model, I made sure that the input data is cleaned and scaled before it was fed into the machine learning models. The data mainly had class imbalancing issue which looks like below.

```
1    183431
0     26162
Name: label, dtype: int64
```





From the above we can see that the data set is highly imbalanced, so applied SMOTET method to balance the dataset.

For this project we need to predict whether the user paid back the credit loan amount within 5 days of issuing the loan. In this dataset, label is the target variable, which consists of two categories, defaulters, and non-defaulters. Which means our target column is categorical in nature, so this is a classification problem.

I have used many classification algorithms and got the prediction results. By doing various evaluations I have selected Gradient Boosting Classifier as best suitable algorithm to create our final model as it is giving least difference in accuracy score and cross validation score among all the algorithms used.

To get good performance and to check whether my model getting over-fitting and under-fitting I have made use of the K-Fold cross validation and then hyper parameter tuning on best model. Then I saved my final model and loaded the same for predictions.

- Testing of Identified Approaches (Algorithms)

Since label is my target variable, which is categorical in nature, from this I can conclude that it is a classification type problem hence I have used following classification algorithms

1. Decision Tree Classifier
2. Random Forest Classifier
3. Extra Trees Classifier
4. Gradient Boosting Classifier
5. Bagging Classifier
6. Extreme Gradient Boosting Classifier (XGB)

- Run and Evaluate selected models

I used a total of 6 classification Models after choosing the random state amongst 1-200 number. I have used Decision Tree Classifier to find best random state and the code is as below
The code for the models is listed below.

Random State:

```
maxAccu=0
maxRS=0
for i in range(1,200):
    x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=.30, random_state =i)
    DTC = DecisionTreeClassifier()
    DTC.fit(x_train, y_train)
    pred = DTC.predict(x_test)
    acc=accuracy_score(y_test, pred)
    if acc>maxAccu:
        maxAccu=acc
        maxRS=i
print("Best accuracy is ",maxAccu," on Random_state ",maxRS)
```

```
Best accuracy is  0.9166901389254855  on Random_state  23
```

Model Building:

Decision Tree Classifier

```
# Checking Accuracy and evaluation metrics for Decision Tree Classifier
DTC = DecisionTreeClassifier()

# Training the model
DTC.fit(x_train,y_train)

#Predicting y_test
predDTC = DTC.predict(x_test)

# Accuracy Score
DTC_score = accuracy_score(y_test, predDTC)*100
print("Accuracy Score:", DTC_score)

# ROC AUC Score
from sklearn.metrics import roc_auc_score
roc_auc_score = roc_auc_score(y_test,predDTC)*100
print("\nroc_auc_score:", roc_auc_score)

# Confusion Matrix
conf_matrix = confusion_matrix(y_test, predDTC)
print("\nConfusion Matrix:\n",conf_matrix)

# Classification Report
class_report = classification_report(y_test,predDTC)
print("\nClassification Report:\n", class_report)

# Cross Validation Score
cv_score = (cross_val_score(DTC, x, y, cv=5).mean())*100
print("Cross Validation Score:", cv_score)

# Result of accuracy minus cv scores
Result = DTC_score - cv_score
print("\nAccuracy Score - Cross Validation Score is", Result)
```

```
Accuracy Score: 91.58087934653231

roc_auc_score: 91.58206097674332

Confusion Matrix:
 [[50623  4297]
 [ 4969 50170]]

Classification Report:
               precision    recall  f1-score   support

           0       0.91      0.92      0.92     54920
           1       0.92      0.91      0.92     55139

    accuracy                           0.92    110059
   macro avg       0.92      0.92      0.92    110059
weighted avg       0.92      0.92      0.92    110059

Cross Validation Score: 90.94974575607687

Accuracy Score - Cross Validation Score is 0.6311335904554483
```

Random Forest Classifier

```
# Checking Accuracy and evaluation metrics for Random Forest Classifier
RFC = RandomForestClassifier()

RFC.fit(x_train,y_train)
predRFC = RFC.predict(x_test)

RFC_score = accuracy_score(y_test, predRFC)*100
print("Accuracy Score:", RFC_score)

from sklearn.metrics import roc_auc_score
roc_auc_score2 = roc_auc_score(y_test,predRFC)*100
print("\nroc_auc_score:", roc_auc_score2)

conf_matrix = confusion_matrix(y_test, predRFC)
print("\nConfusion Matrix:\n",conf_matrix)

class_report = classification_report(y_test,predRFC)
print("\nClassification Report:\n", class_report)

cv_score2 = (cross_val_score(RFC, x, y, cv=5).mean())*100
print("Cross Validation Score:", cv_score2)

# Result of accuracy minus cv scores
Result = RFC_score - cv_score2
print("\nAccuracy Score - Cross Validation Score is", Result)
```

```
Accuracy Score: 95.31342280049792

roc_auc_score: 95.31468429710587

Confusion Matrix:
 [[52695  2225]
 [ 2933 52206]]

Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.96      0.95     54920
           1       0.96      0.95      0.95     55139

    accuracy                           0.95    110059
   macro avg       0.95      0.95      0.95    110059
weighted avg       0.95      0.95      0.95    110059

Cross Validation Score: 95.03057026354098

Accuracy Score - Cross Validation Score is 0.2828525369569377
```

ExtraTrees Classifier

```
# Checking Accuracy and evaluation metrics for ExtraTrees Classifier
XT = ExtraTreesClassifier()

XT.fit(x_train,y_train)
predXT = XT.predict(x_test)

XT_score = accuracy_score(y_test, predXT)*100
print("Accuracy Score:", XT_score)

roc_auc_score3 = roc_auc_score(y_test,predXT)*100
print("\nroc_auc_score:", roc_auc_score3)

conf_matrix = confusion_matrix(y_test, predXT)
print("\nConfusion Matrix:\n",conf_matrix)

class_report = classification_report(y_test,predXT)
print("\nClassification Report:\n", class_report)

cv_score3 = (cross_val_score(XT, x, y, cv=5).mean())*100
print("Cross Validation Score:", cv_score3)

# Result of accuracy minus cv scores
Result = XT_score - cv_score3
print("\nAccuracy Score - Cross Validation Score is", Result)
```

```
Accuracy Score: 95.9712517831345

roc_auc_score: 95.97426962939863

Confusion Matrix:
 [[53542  1378]
 [ 3056 52083]]

Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.97      0.96     54920
           1       0.97      0.94      0.96     55139

    accuracy                           0.96    110059
   macro avg       0.96      0.96      0.96    110059
weighted avg       0.96      0.96      0.96    110059

Cross Validation Score: 96.36457531404268

Accuracy Score - Cross Validation Score is -0.3933235309081766
```

Gradient Boosting Classifier

```
# Checking Accuracy and evaluation metrics for GradientBoosting Classifier
GB = GradientBoostingClassifier()

GB.fit(x_train,y_train)
predGB = GB.predict(x_test)

GB_score = accuracy_score(y_test, predGB)*100
print("Accuracy Score:", GB_score)

roc_auc_score4 = roc_auc_score(y_test,predGB)*100
print("\nroc_auc_score:", roc_auc_score4)

conf_matrix = confusion_matrix(y_test, predGB)
print("\nConfusion Matrix:\n",conf_matrix)

class_report = classification_report(y_test,predGB)
print("\nClassification Report:\n", class_report)

cv_score4 = (cross_val_score(GB, x, y, cv=5).mean())*100
print("Cross Validation Score:", cv_score4)

# Result of accuracy minus cv scores
Result = GB_score - cv_score4
print("\nAccuracy Score - Cross Validation Score is", Result)
```

```
Accuracy Score: 89.91450040432859

roc_auc_score: 89.91752681269122

Confusion Matrix:
 [[50218  4702]
 [ 6398 48741]]

Classification Report:
              precision    recall  f1-score   support

           0       0.89      0.91      0.90     54920
           1       0.91      0.88      0.90     55139

    accuracy                           0.90    110059
   macro avg       0.90      0.90      0.90    110059
weighted avg       0.90      0.90      0.90    110059

Cross Validation Score: 89.67324027909426

Accuracy Score - Cross Validation Score is 0.2412601252343336
```

Bagging Classifier

```
# Checking Accuracy and evaluation metrics for Bagging Classifier
BC = BaggingClassifier()

BC.fit(x_train,y_train)
predBC = BC.predict(x_test)

BC_score = accuracy_score(y_test, predBC)*100
print("Accuracy Score:", BC_score)

roc_auc_score5 = roc_auc_score(y_test,predBC)*100
print("\nroc_auc_score:", roc_auc_score5)

conf_matrix = confusion_matrix(y_test, predBC)
print("\nConfusion Matrix:\n",conf_matrix)

class_report = classification_report(y_test,predBC)
print("\nClassification Report:\n", class_report)

cv_score5 = (cross_val_score(BC, x, y, cv=5).mean())*100
print("Cross Validation Score:", cv_score5)

# Result of accuracy minus cv scores
Result = BC_score - cv_score5
print("\nAccuracy Score - Cross Validation Score is", Result)
```

```
Accuracy Score: 94.14586721667469

roc_auc_score: 94.14872415666916

Confusion Matrix:
 [[52495  2425]
 [ 4018 51121]]

Classification Report:
              precision    recall  f1-score   support

           0       0.93      0.96      0.94     54920
           1       0.95      0.93      0.94     55139

    accuracy                           0.94    110059
   macro avg       0.94      0.94      0.94    110059
weighted avg       0.94      0.94      0.94    110059

Cross Validation Score: 93.71127315965356

Accuracy Score - Cross Validation Score is 0.43459405702112974
```

Extreme Gradient Boosting(XGB) Classifier

```
# Checking Accuracy and evaluation metrics for XGB Classifier
XGB = xgb(verbosity=0)

XGB.fit(x_train,y_train)
predXGB = XGB.predict(x_test)

XGB_score = accuracy_score(y_test, predXGB)*100
print("Accuracy Score:", XGB_score)

roc_auc_score6 = roc_auc_score(y_test,predXGB)*100
print("\nroc_auc_score:", roc_auc_score6)

conf_matrix = confusion_matrix(y_test, predXGB)
print("\nConfusion Matrix:\n",conf_matrix)
class_report = classification_report(y_test,predXGB)
print("\nClassification Report:\n", class_report)

cv_score6 = (cross_val_score(XGB, x, y, cv=5).mean())*100
print("Cross Validation Score:", cv_score6)

# Result of accuracy minus cv scores
Result = XGB_score - cv_score6
print("\nAccuracy Score - Cross Validation Score is", Result)
```

```
Accuracy Score: 95.06537402665843

roc_auc_score: 95.06360616703753

Confusion Matrix:
 [[51721  3199]
 [ 2232 52907]]

Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.94      0.95     54920
           1       0.94      0.96      0.95     55139

    accuracy                           0.95    110059
   macro avg       0.95      0.95      0.95    110059
weighted avg       0.95      0.95      0.95    110059

Cross Validation Score: 93.61234288941812

Accuracy Score - Cross Validation Score is 1.4530311372403162
```

- Key Metrics for success in solving problem under Consideration

The key metrics used here were Accuracy Score, Precision, Recall, F1 score, Cross Validation Score, Roc Auc Score and Confusion Matrix. We tried to find out the best parameters and to increase our scores by using Hyperparameter Tuning and used RandomizedSearchCV method.

- **Accuracy score** means how accurate our model is that is the degree of closeness of the measured value to a standard or true value. It is one metric for evaluating classification models. Accuracy is the ratio of number of correct predictions into number of predictions.
- **Precision** is the degree to which repeated measurements under the same conditions are unchanged. It is amount of information that is conveyed by a value. It refers to the data that is correctly classified by the classification algorithm.
- **Recall** is how many of the true positives were recalled (found). Recall refers to the percentage of data that is relevant to the class. In binary classification problem recall is calculated as below:

  Recall = Number of True Positives/ (Total number of True Positives + Total number of False Negatives)

- **F1 Score** is used to express the performance of the machine learning model (or classifier). It gives the combined information about the precision and recall of a model. This means a high F1-score indicates a

high value for both recall and precision.

- **Cross Validation Score** is a technique in which we train our model using the subset of the data-set and then evaluate using the complementary subset of the data-set. It is used to protect against overfitting in a predictive model, particularly in a case where the amount of data may be limited. In cross-validation, you make a fixed number of folds (or partitions) of the data, run the analysis on each fold, and then average the overall error estimate. It is used to estimate the performance of ML models.
- **Roc Auc Score:** The Receiver Operator Characteristic (ROC) curve is an evaluation metric for binary classification problems. It is a probability curve that plots the TPR against FPR at various threshold values.

  The Area Under Curve (AUC) is the measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve.

- **Confusion Matrix** is one of the evaluation metrics for machine learning classification problems, where a trained model is being evaluated for accuracy and other performance measures. And this matrix is called the confusion matrix since it results in an output that shows how the system is confused between the two classes.

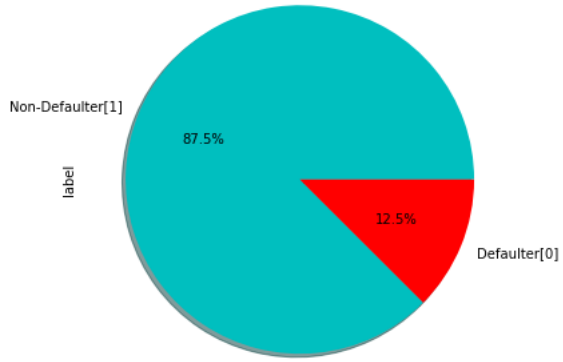
- **Hyperparameter Tuning:** There is a list of different machine learning models. They all are different in some way or the other, but what makes them different is nothing but input parameters for the model. These input parameters are named as Hyperparameters. These hyperparameters will define the architecture of the model, and the best part about these is that you get a choice to select these for your model. You must select from a specific list of hyperparameters for a given model as it varies from model to model.

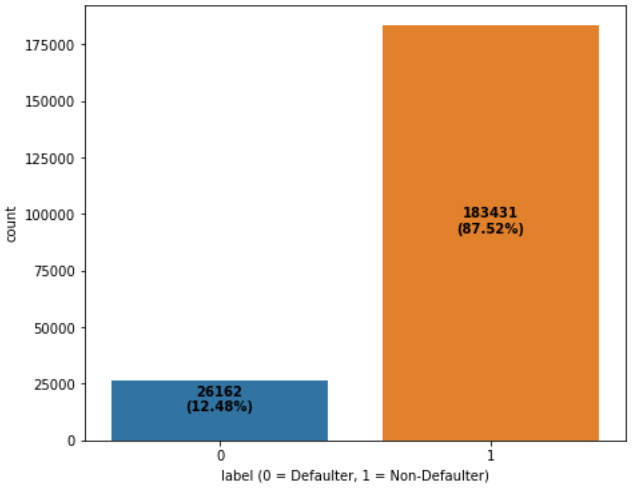
- Visualizations

I used pandas profiling to get the over viewed visualization on the pre-processed data. Pandas is an open-source Python module with which we can do an exploratory data analysis to get detailed description of the features and it helps in visualizing and understanding the distribution of each variable. I have analysed the data using both univariate and bivariate analysis. In univariate analysis I have used distribution plot, pie plot and count plot and in bivariate analysis I have used bar plots. These plots have given good pattern.

Univariate Analysis: Visualizing label whether the user paid back the credit amount within 5 days of issuing the loan or not.

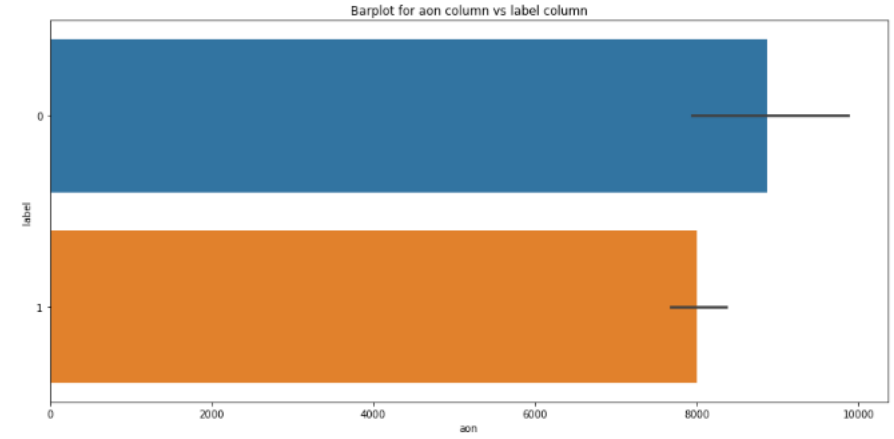
```
1    183431
0     26162
Name: label, dtype: int64
```

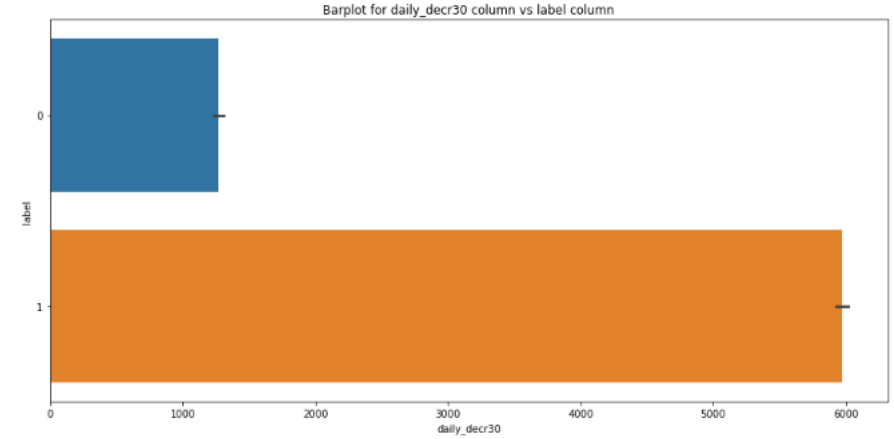


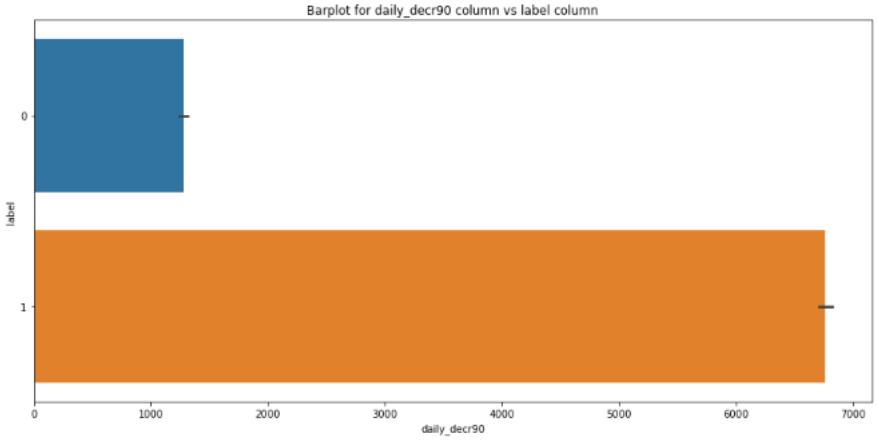


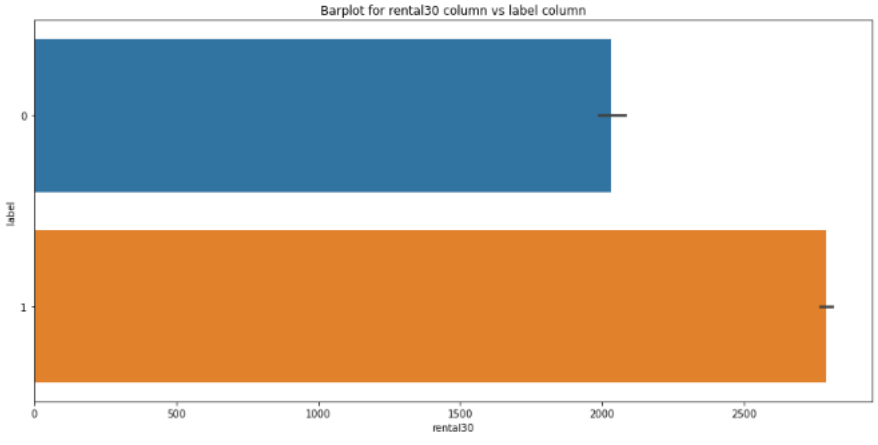
From the above plots we can observe around 87.5% of the loan has been paid by the user and only 12.5% of the loan failed to pay.
Since the data was not balanced, I have used SMOTE method to balance it that I already have been mentioned.

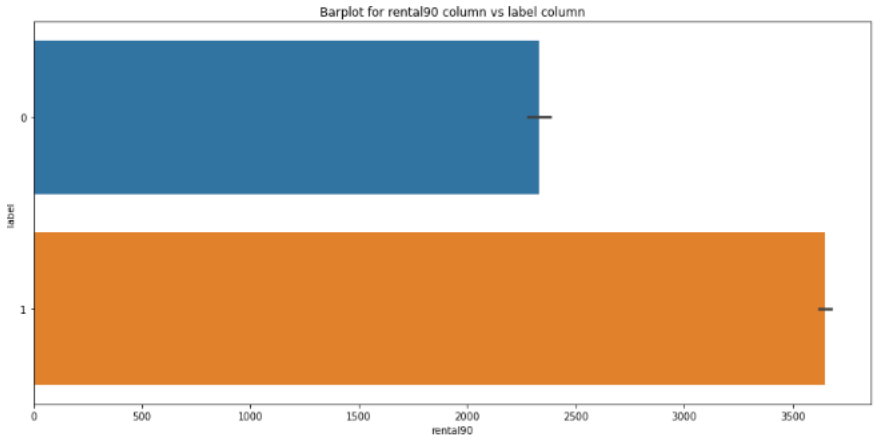
Bivariate Analysis: Comparing label with remaining features:











Observations:

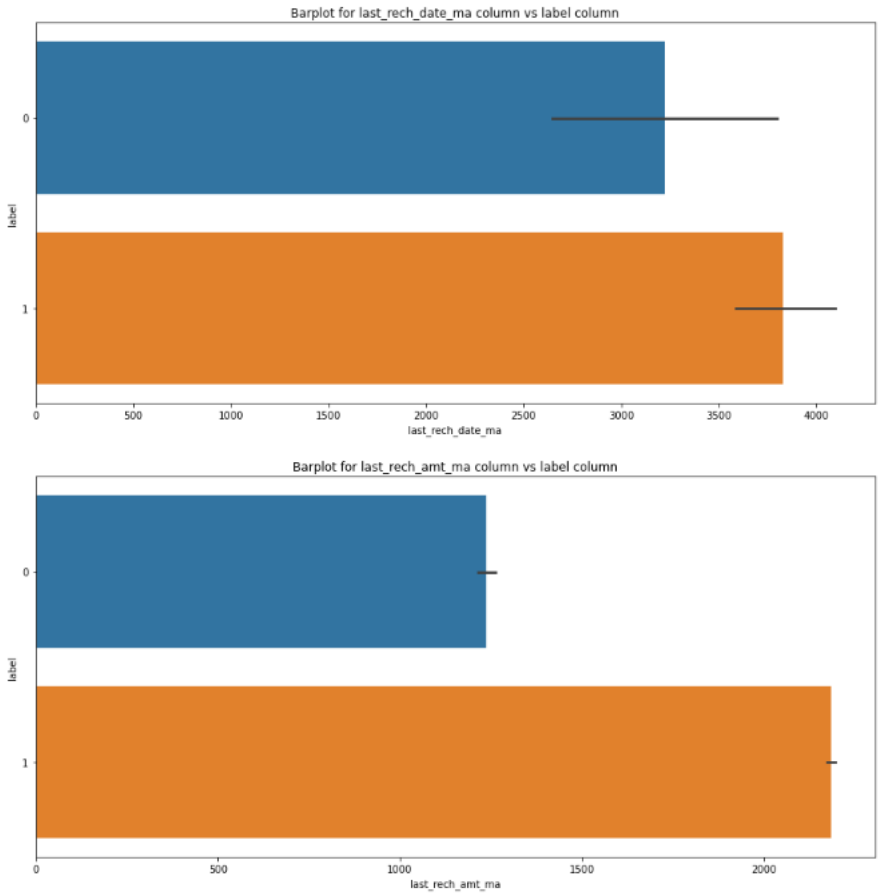
From the above bar plot we can observe that the defaulter rate is higher where the user age on cellular network in days is high.

Most of the users who have paid back the credit amount within 5 days of issuing loan, they have high rate of daily amount spent from the account over last 30 days and 90 days

The users who have spent daily amount from main account over last 30 days and 90 days have always paid back the loan amount within 5 days

Non defaulter users have average main account balance over last 30 days and 90 days

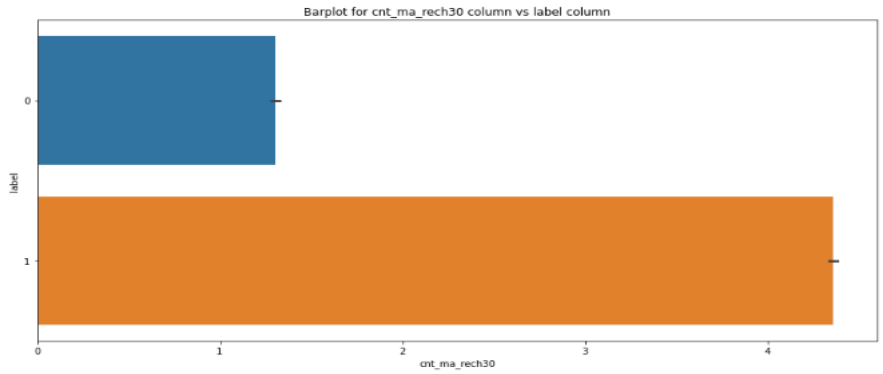
That means the users who have average main account balance always pays back the credit amounts within 5 days. And around 1% of the users either failed to payback the loan amount within the due date or they are not paying the loan.

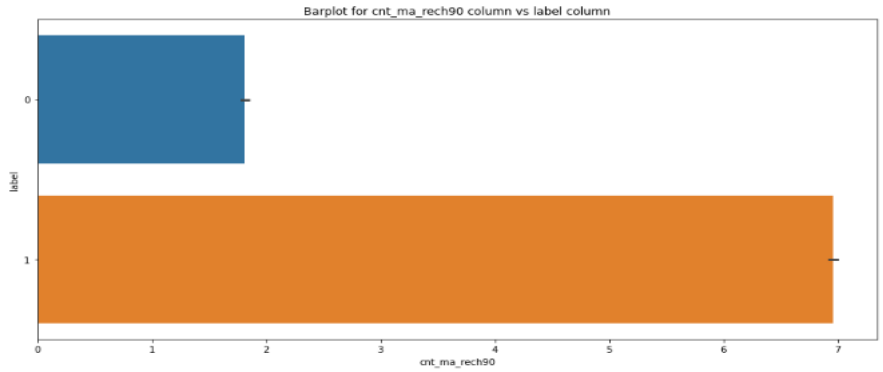


Observations:

The users who have recharged their main account on time are most likely to pay back their loan amount within 5 days. Also some of the users who have not paid back their loan within 5 days they also recharged their main account on time

Looking at above plot of last_rech_amt_ma, we can say that if the amount of last recharge of main account is around 2000 then a greater number of people will pay back the loan amount.

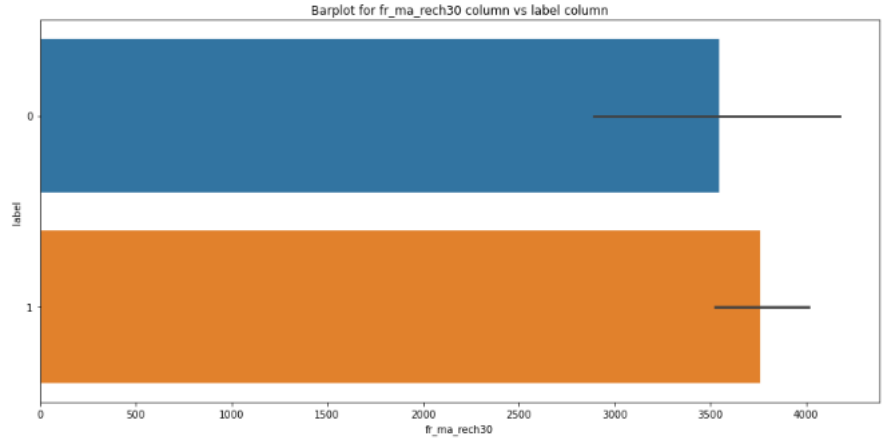


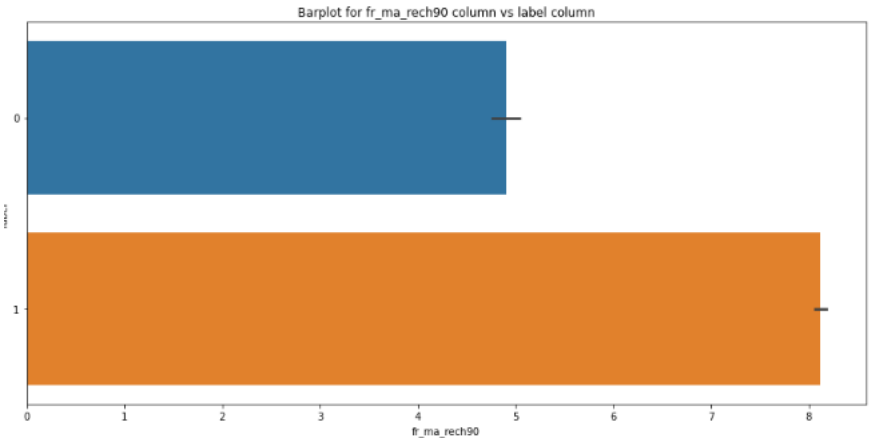


Observations:

The users who have paid back their loan within 5 days have got recharged their main account upto 7 times in last 90 days and the users who have not been paid loan within due date, they have got recharged their main account twice in last 90 days.

From both the plots we can say that the users who got recharged their main account maximum times, they are able to pay back their loan amount within 5 days compared to the users who got their main account recharged less than 2 times.



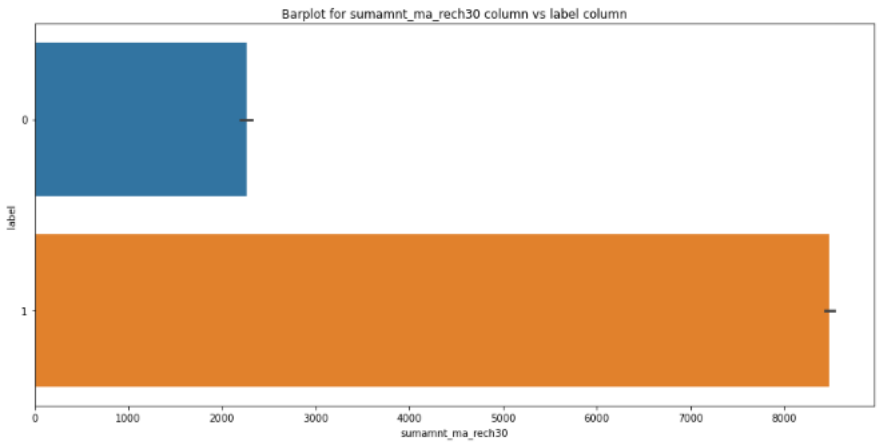


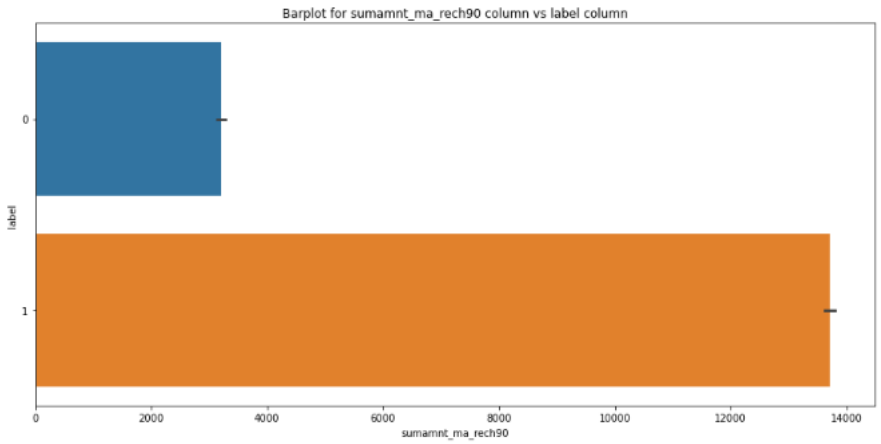
Observations:

The count of defaulters and non-defaulters is almost similar for the frequency of main account recharged in last 30 days. They didn't pay back the loan within 5 days. Which means there it is not contributing more for prediction

The frequency of main account recharged in last 90 days is increased for non-defaulters compared to defaulters.

From the frequency of main account recharged in last 30 days & 90 days we have seen the users with low frequency are causing huge losses, company should implement strategies to reduce that like send SMS alerts for notification.

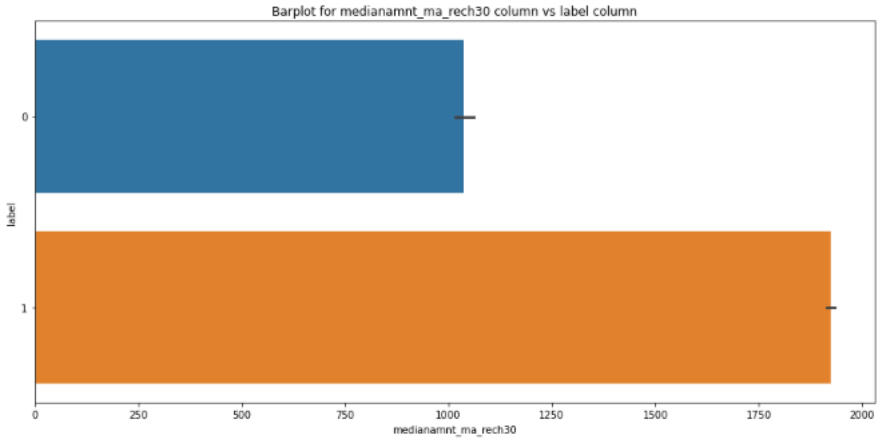


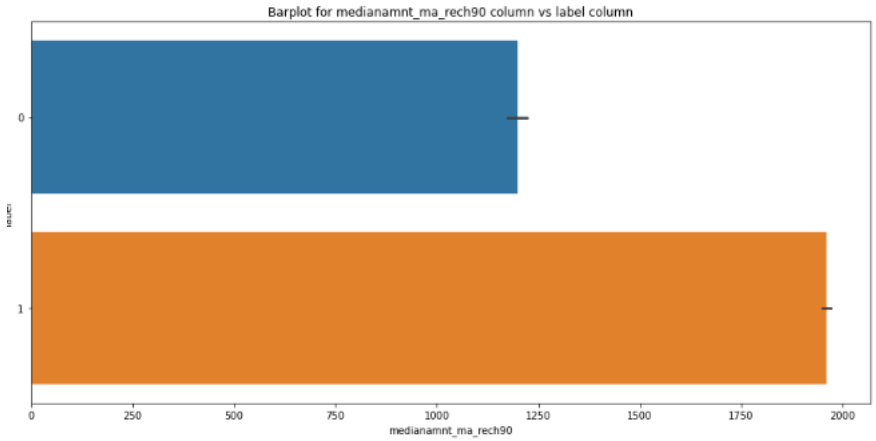


Observations:

The users who failed to pay back the loan within 5 days have less amount of recharge in their main account over last 30 days which is around 2000-2400 (in Indonesian Rupiah). And the users who paid back their loan within 5 days, they are recharging their main account more than 8000 (in Indonesian Rupiah) in last 30 days.

The users who have paid their loan amount within 5 days have the total amount of recharge in their main account around 13700 (Indonesian Rupiah) in last 90 days while the defaulters have their total amount of recharge around 3500 (Indonesian Rupiah) over last 90 days.

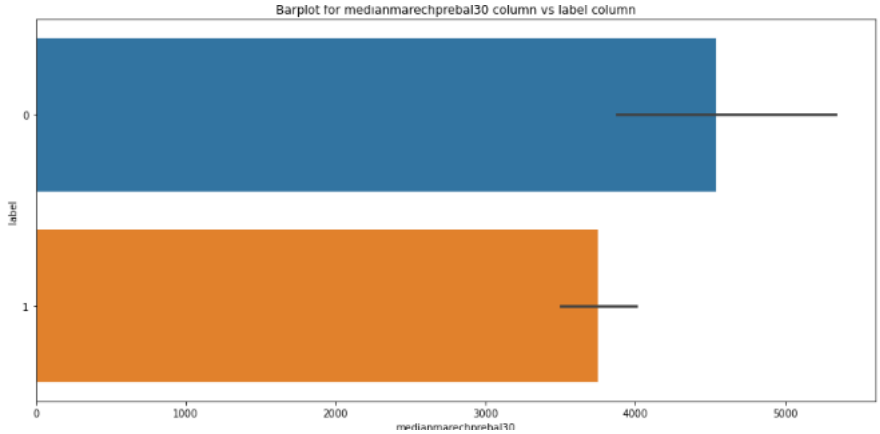


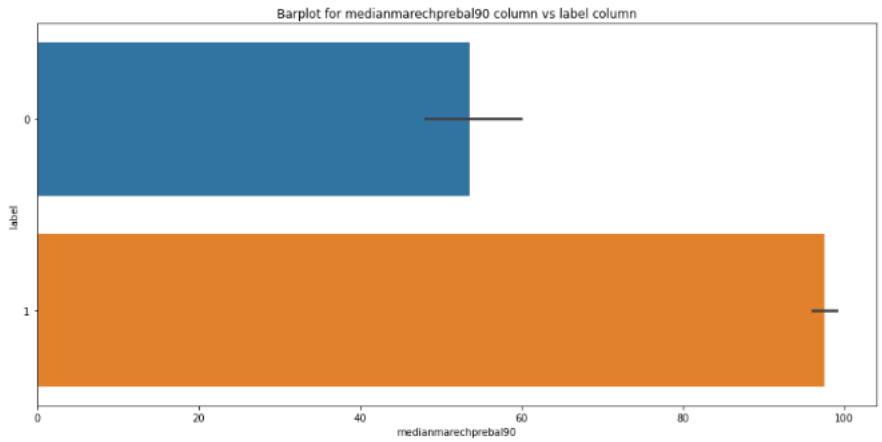


Observations:

The users who have done their median amount of recharge of around 2000 in main account over last 30 days have successfully paid their credit amount within 5 days of issuing loan while the users who have done amount recharge of around 1000 have failed to pay back the loan within due date.

Like 30 days data, here also the users who have done their median amount recharge of 1950 in their main account over last 90 days they have paid back their credit amount within 5 days while the users having their median amount around 1200 have not paid the loan within 5 days

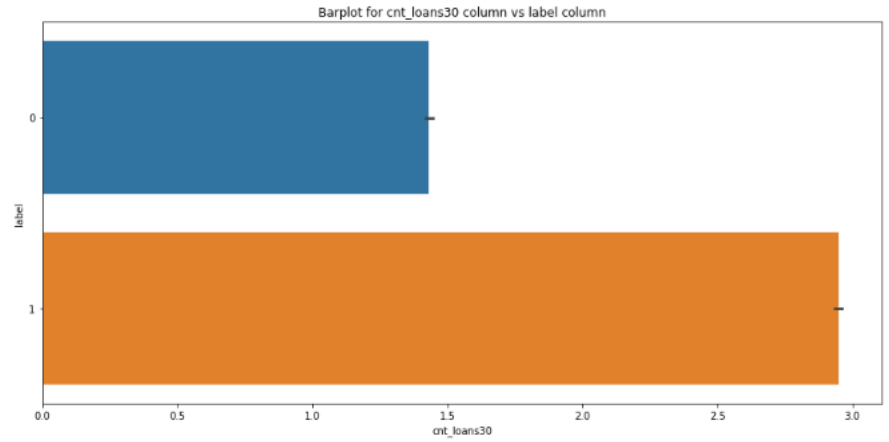


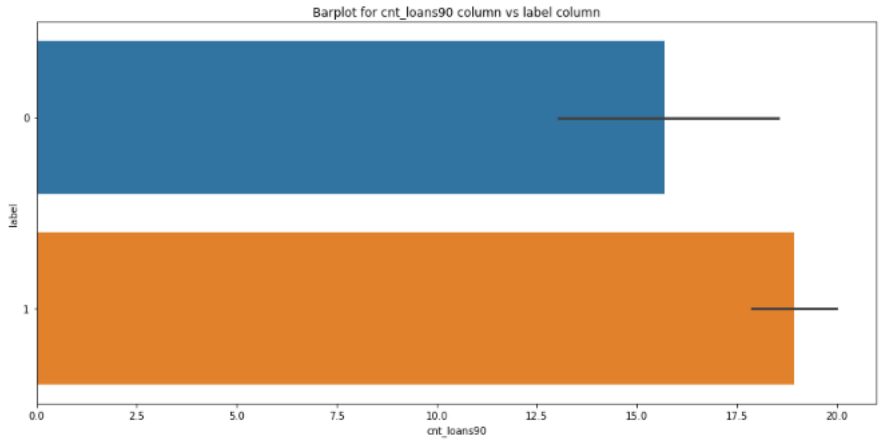


Observations:

In 30 days data, the median of main account balance for defaulters are around 4500 (Indonesian Rupiah) which is high compared to non-defaulters. Which means increasing median of main account balance just before recharge in last 30 days at user level, increasing the probability to being defaulter.

In last 90 days data, the median of main account balance for non-defaulters are around 100 (Indonesian Rupiah) which is high compared to defaulters. Which means increasing median of main account balance just before recharge in last 90 days at user level, increasing the probability of being non-defaulters.

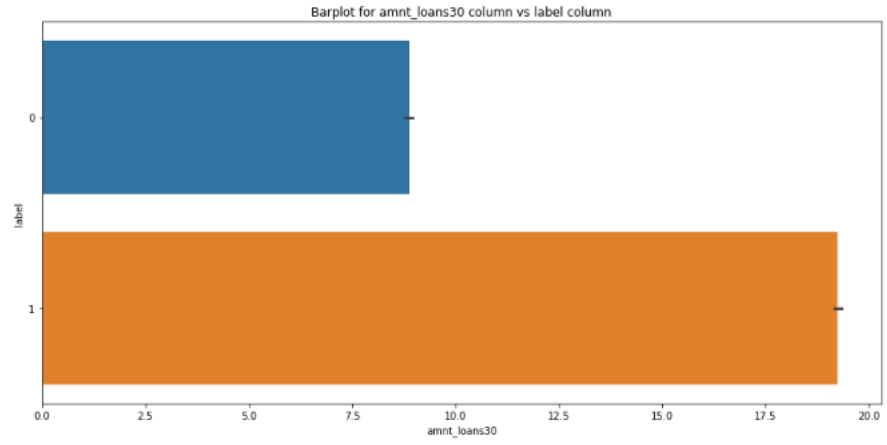


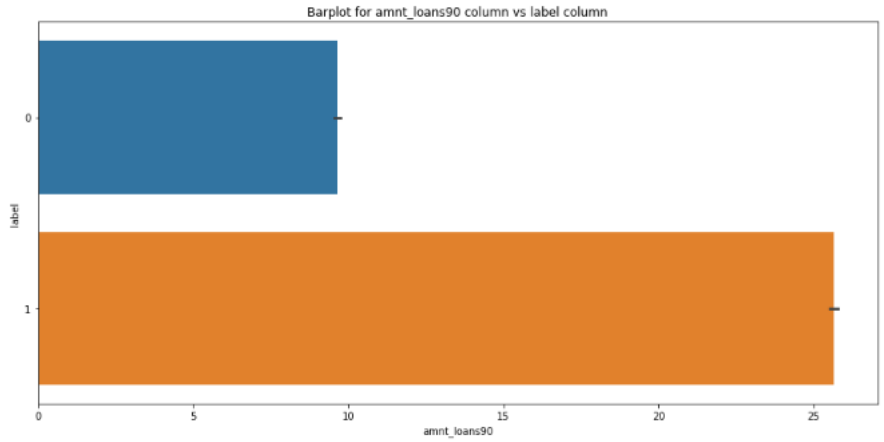


Observations:

Defaulters have taken 1 loan in last 30 days that is when a person takes loan amount for 1 time in last 30 days the chances of not paying back the credit amount are higher. And the users who have paid back the loan, they have taken maximum number of 3 loans in last 30 days data.

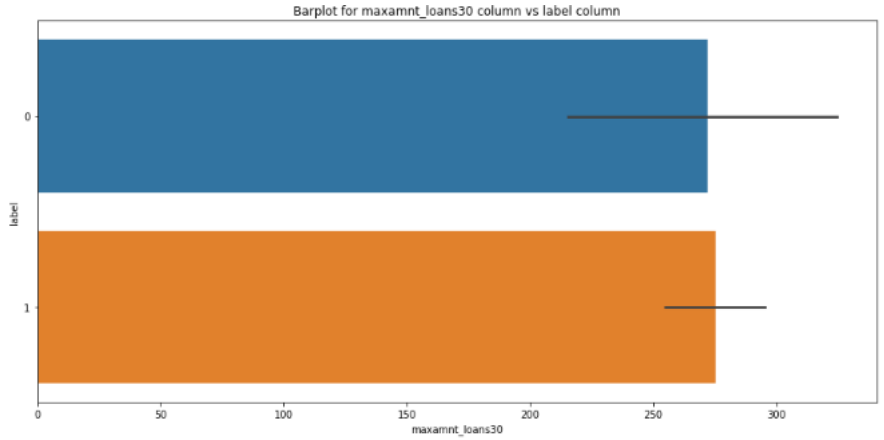
In 90 days data, the number of loans taken by the defaulters are highly increasing also increasing the probability to being defaulter. Also, the number of loans taken by non-defaulters being decreased in last 90 days when compared to 30 days data.

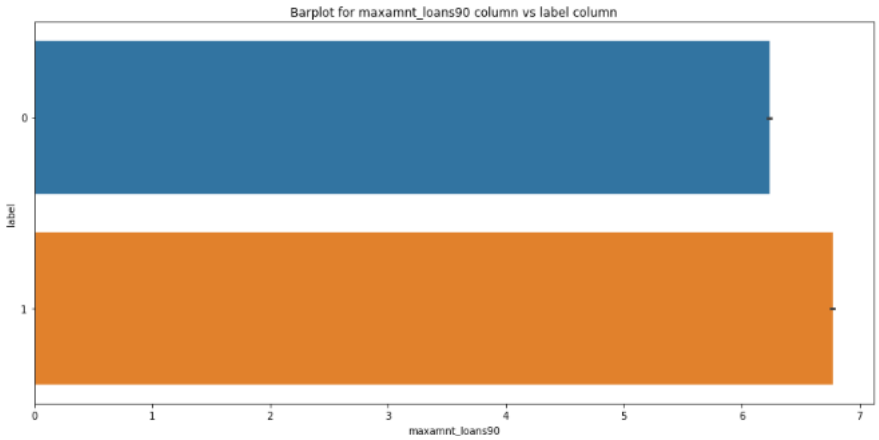




The total amount of loans taken by the defaulters in last 30 days are in the range of 7.5-10 while the non-defaulters have taken around 20 loans in last 30 days.

The total amount of loans taken by the defaulters in last 90 days are upto 10 and the non- defaulters have taken total amount of loans around 26 in last 90 days.

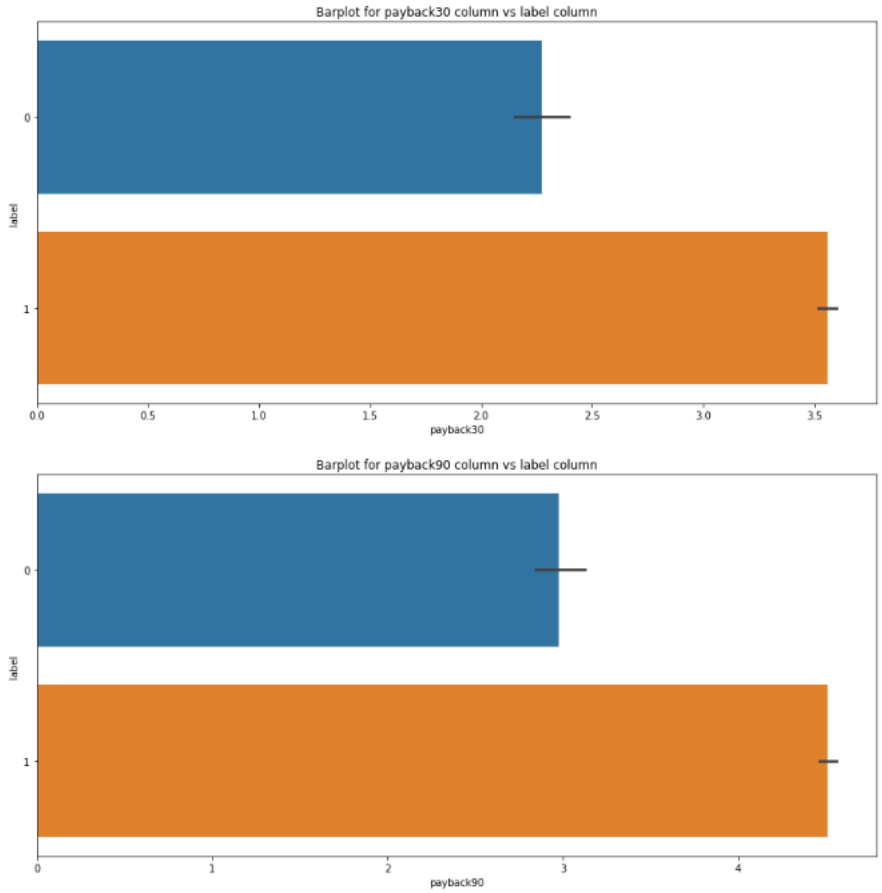




Observations:

The maximum amount of loan taken by the user in last 30 days and 90 days are almost same. The maximum amount of loan taken by the defaulters and non-defaulters are upto 6 and 7 respectively in last 30 and 90 days.

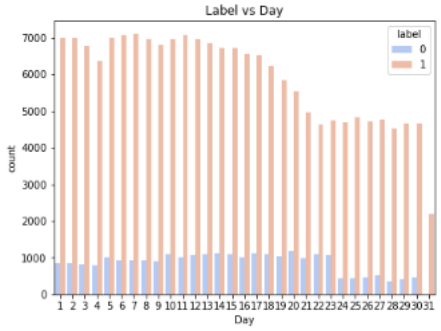
So from the plot we can say that whenever the user takes the maximum loan amount of 6, then only some users may not pay back the loan amount.

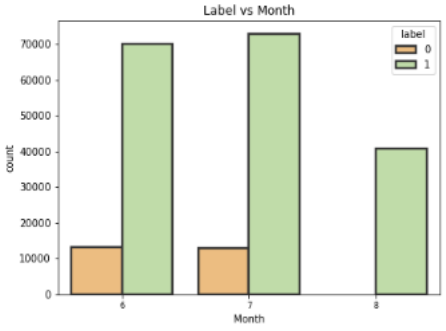


Observations:

The defaulters are paying back their loan in an average of 2-2.5 days and the non-defaulters are paying back their loan in an average of 3 days over last 30 days.

The defaulters in last 90 days, are paying back their loan in an average of 3 days and non-defaulters are paying back their loan in 4-5 days over last 90 days.





Observations:

The users who have taken loans in the month of august, they seem paying back their loan within 5 days.

- Interpretation of the Results

Visualizations: I have used distribution plot to visualize the numerical variables. Used bar plots to check the relation between label and the features. The heat map and bar plot helped me to understand the correlation between dependent and independent features. Also, heat map helped to detect the multicollinearity problem and feature importance. Detected outliers and skewness with the help of box plots and distribution plots respectively. And I found some of the features skewed to right. I got to know the count of each column using bar plots.
Pre-processing: The dataset should be cleaned and scaled to build the ML models to get good predictions. I have performed few processing steps which I have already mentioned in the pre-processing steps where all the important features are present in the dataset and ready for model building.
Model building: After cleaning and processing data, I performed train test split to build the model. I have built multiple classification models to get the accurate accuracy score, and evaluation metrics like precision, recall, confusion matrix, f1 score. I got Gradient Boosting Classifier as best model which gives 90% accuracy score. I checked the cross-validation score ensuring

there will be no overfitting. After tuning the best model Gradient Boosting Classifier, I got 95% accuracy score and also got increment in AUC-ROC curve. Finally, I saved my final model and got the good predictions results for defaulters.

CONCLUSION

- Key Findings and Conclusions of the Study

In this study, we have used multiple machine learning models to predict the house sale price. We have gone through the data analysis by performing feature engineering, finding the relation between features and label through visualizations. And got the important feature and we used these features to predict the price by building ML models. We have got good prediction results and after hyper parameter tuning, R2 score was nearly 89% also the errors decreased which means no over-fitting issue.

Findings:

From the whole study we found that the MFIs have provided loan to the user who have no recharge or balance in their account which needs to be stopped. Also, the frequency of main account recharged in last 30 days & 90 days we have seen the users with low frequency are causing huge losses, company should implement strategies to reduce like sending SMS alerts for notification. We found the defaulting rate is higher in old customers list. We found outliers and removed them and couldn't remove all the outliers since the data is expensive so, proceeded the data with remaining outliers. Further, removed skewness. Looking at the heat map, I could see there were few features which were correlated with each other, yet I haven't removed them based on their correlation thinking multicollinearity will not affect prediction. Other insight from this study is the impact of SMOTE on the model performance as well as how the number of variables included in the models.

- ## Learning Outcomes of the Study in respect of Data Science

While working on this project I learned more things about the housing market and how the machine learning models have helped to predict the price of house which indeed helps the sellers and buyers to understand the future price of the house. I found that the project was quite interesting as the dataset contains several types of data. I used several types of plotting to visualize the relation between target and features.

This graphical representation helped me to understand which features are important and how these features describe the sale price. Data cleaning was one of the important and crucial things in this project where I replaced all the null values with imputation methods and dealt with features having zero values and time variables.

Finally, our aim is achieved by predicting the house price for the test data, I hope this will be further helps for sellers and buyers to understand the house marketing. The machine learning models, and data analytic techniques will have an important role to play in this type of problems. It helps the customers to know the future price of the houses.

- ## Limitations of this work and Scope for Future Work

Limitation is it will only work for this particular use case and will need to be modified if tried to be utilized on a different scenario but on a similar scale. Scope is that we can use it in companies to find whether we should provide loan to a person or not and we can also make prediction about a person buying an expensive service on the basis of their personal details that we have in this dataset like number of times data account got recharged in last 30 days and daily amount spent from main account, averaged over last 30 days (in Indonesian Rupiah) so even a marketing company can also use this.

Thank You