MICRO CREDIT DEFAULTER PROJECT

Submitted by:

ANJAY KRISHNA T U

INTRODUCTION

- ## Business Problem Framing

  Micro Finance Institution are providing Mobile financial services to low-income families and poor customers that can help them in the need of hour . The micro-credit providing to the customers mobile balances need to be paid back in 5 days, some customers are failing to do that. This project is to predict the customer will be paying the loan amount within 5 days or not. This will be very useful for the client in selecting the customers and future investment.

- ## Motivation for the Problem Undertaken

  Communication has important role in a person's life, The client providing services to poor customers so that may help them in many ways. If we can find non-default customers and provide services to them it will benefits both the client and customers.

Analytical Problem Framing

- Data Sources and their formats

  In the given CSV file, which has the details about the customers like mobile number, age on cellular network, daily amount spent from main account, average main account balance, last recharge date, loan amount and etc. The dataset has total 209593 rows and 37 columns. The data types of the columns are integer, float and object.

- Data Preprocessing Done

  The dataset has no null values, but some of the values were negative by using the absolute function converted those negative values to positive. The column 'Unnamed: 0' is not need so removed by dropping from the dataset. A large number of outliers are present in the dataset, using the z-score method the outliers are removed.

- Data Inputs- Logic- Output Relationships

  The label in the given dataset is the output and all others are the input. The output variable/label is highly positive correlated to cnt_ma_rech30,cnt_ma_rech90,sumamnt_ma_rech30,sumamnt_ma_rech90,cnt_loans30,amnt_loans30,amnt_loans90 and negative correlated to aon,medianmarechprebal30,fr_da_rech30, fr_da_rech90.

- Hardware and Software Requirements and Tools Used

  Jupyter Notebook – For programming

  Numpy – For mathematical calculations

  Pandas , Seaborn, Matplotlib – For visualization

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

  From the analysis it is clear that only few input variables are correlated to the output variables. So, if we focusing on the correlated input variables and removing the unwanted data from the dataset will improve the model.

- Testing of Identified Approaches (Algorithms)

  Here 4 Algorithms are used for training and testing.

  1.Logistic Regression

  2.Decision Tree Classifier

  3.Random Forest Classifier

  4.Gradient Boosting Classifier

- Run and Evaluate selected models

  1.Logistic Regression

  Logistic regression is a supervised machine learning classification algorithm used to predict probability of a target. Only two classes will be there either 0 or 1.

```
lg=LogisticRegression(C=1.5,multi_class='auto',penalty='l1',solver='saga')
lg.fit(x_train,y_train)
predlg=lg.predict(x_test)
print('Accuracy score :',accuracy_score(y_test,predlg))
print('Confusion matrix',confusion_matrix(y_test,predlg))
print('Classification report :',classification_report(y_test,predlg))
```

```
Accuracy score : 0.7673363095238095
Confusion matrix [[5362 1402]
 [1725 4951]]
Classification report :              precision    recall  f1-score   support

           0       0.76      0.79      0.77      6764
           1       0.78      0.74      0.76      6676

    accuracy                           0.77     13440
   macro avg       0.77      0.77      0.77     13440
weighted avg       0.77      0.77      0.77     13440
```

The Accuracy score of logistic regression model is 76.73%. This model predicts that out of 6764 True positive cases 5362 are True positive and out of 6676 True negative cases 4951 are True negative.

2.Decision Tree Classifier

Decision tree classifier is a supervised machine learning algorithm for both classification and regression problem. But more accurate in classification problem.

```
dtc=DecisionTreeClassifier(criterion='entropy',max_depth=11,max_features='sqrt',
                           min_samples_split=10,splitter='best')
dtc.fit(x_train,y_train)
preddtc=dtc.predict(x_test)
print('Accuracy score :',accuracy_score(y_test,preddtc))
print('Confusion matrix :',confusion_matrix(y_test,preddtc))
print('Classification report :',classification_report(y_test,preddtc))
```

```
Accuracy score : 0.7898065476190477
Confusion matrix : [[5540 1224]
 [1601 5075]]
Classification report :              precision    recall  f1-score   support

           0       0.78      0.82      0.80      6764
           1       0.81      0.76      0.78      6676

    accuracy                           0.79     13440
   macro avg       0.79      0.79      0.79     13440
weighted avg       0.79      0.79      0.79     13440
```

The Accuracy score of Decision tree model is 78.98%. This model predicts that 5540 True positive out of 6764 cases and 5075 True negative out of 6676 cases.

3.Random Forest Classifier

Random forest classifier is an ensemble method for both classification and regression problem. This classifier contains number of decision trees and average of their output is taken for the improved accuracy.

```
rfc=RandomForestClassifier(bootstrap=False,criterion='gini',
                           max_features='sqrt',min_samples_split=8)
rfc.fit(x_train,y_train)
predrfc=rfc.predict(x_test)
print('Accuracy score :',accuracy_score(y_test,predrfc))
print('Confusion matrix :',confusion_matrix(y_test,predrfc))
print('Classification report :',classification_report(y_test,predrfc))
```

```
Accuracy score : 0.8499255952380952
Confusion matrix : [[5576 1188]
 [ 829 5847]]
Classification report :              precision    recall  f1-score   support

           0       0.87      0.82      0.85      6764
           1       0.83      0.88      0.85      6676

    accuracy                           0.85     13440
   macro avg       0.85      0.85      0.85     13440
weighted avg       0.85      0.85      0.85     13440
```

The Accuracy score of random forest classifier is 84.99%. This model predicts that 5576 True positive cases out of 6764 cases and 5847 True negative cases out of 6676 cases.

4.Gradient Boosting Classifier

Gradient boosting classifier is also ensemble method, that works by combining several weak models to create new strong predictive model.

```
gb=GradientBoostingClassifier(learning_rate=0.1,loss='exponential',
                              max_depth=7,max_features='auto',subsample=1.0)
gb.fit(x_train,y_train)
predgb=gb.predict(x_test)
print('Accuracy score :',accuracy_score(y_test,predgb))
print('Confusion matrix :',confusion_matrix(y_test,predgb))
print('Classification report :',classification_report(y_test,predgb))
```

```
Accuracy score : 0.8591517857142857
Confusion matrix : [[5788  976]
 [ 917 5759]]
Classification report :              precision    recall  f1-score   support

           0       0.86      0.86      0.86      6764
           1       0.86      0.86      0.86      6676

    accuracy                           0.86     13440
   macro avg       0.86      0.86      0.86     13440
weighted avg       0.86      0.86      0.86     13440
```

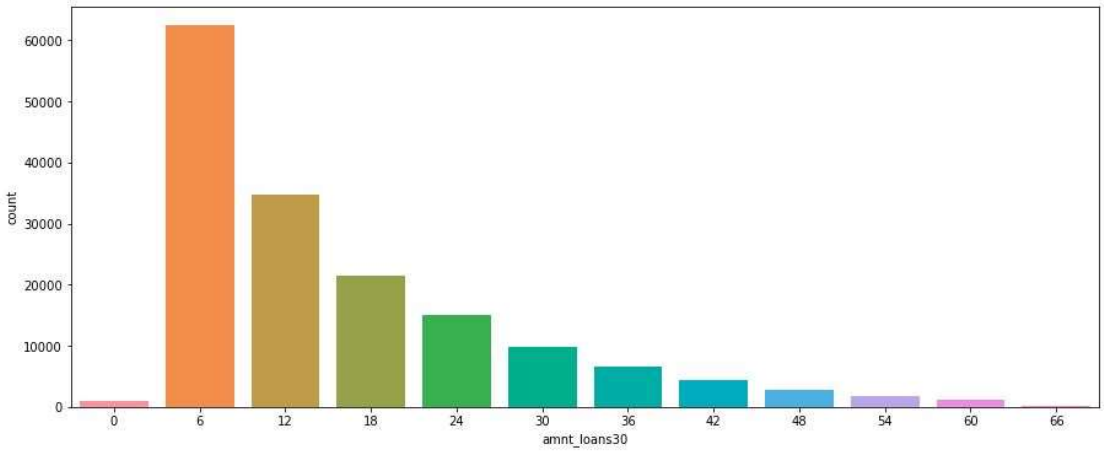
The Accuracy score of gradient boosting classifier is 85.91%. This model predicts that 5788 True positive cases out of 6764 cases and 5759 True negative cases out of 6676 cases.

- Visualizations

The count plot shows that the amount of loans taken by users in last 30 days are mostly 6,12,18 and 24 .

```
plt.figure(figsize=(15,6))
sns.countplot(df['amnt_loans30'])
```

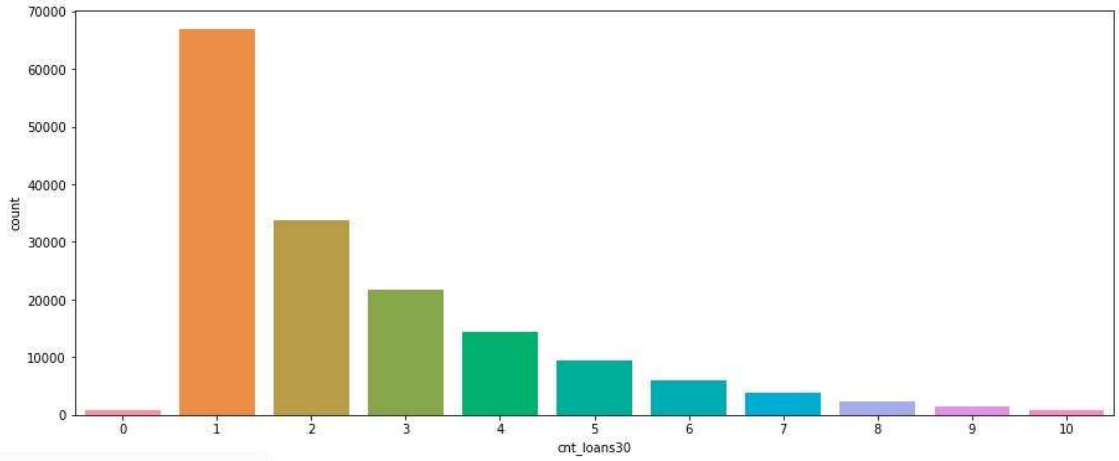
```
<AxesSubplot:xlabel='amnt_loans30', ylabel='count'>
```



The number of loans taken by user is last 30 days mostly 1,2,3 and 4.

```
plt.figure(figsize=(15,6))
sns.countplot(df['cnt_loans30'])
```

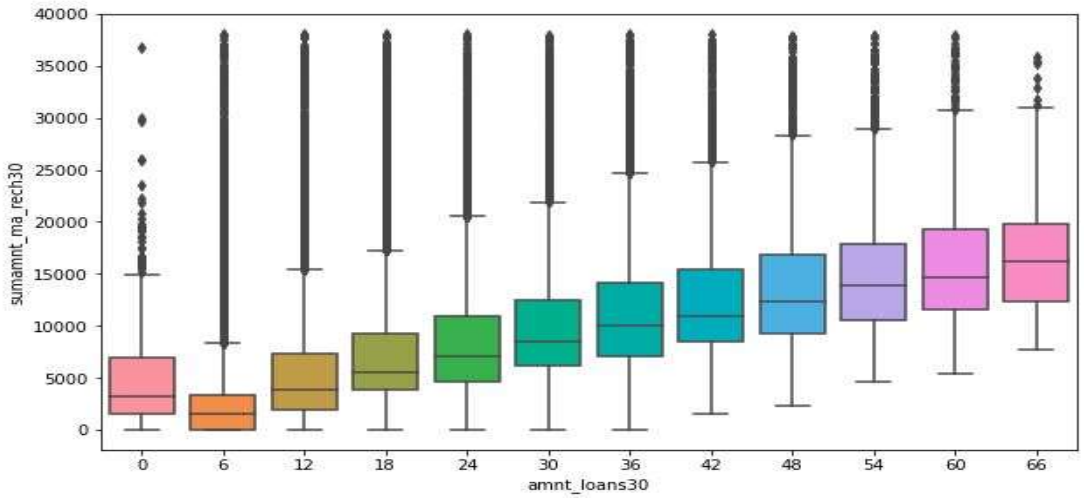
```
<AxesSubplot:xlabel='cnt_loans30', ylabel='count'>
```



The Boxplot shows users who took loan amount 6 has a total recharge amount in 30 days ranging from 0 to 8000. The loan amount 30 is ranging from 0 to 22500 of the total recharge amounts. The loan amount 66 is ranging from 6000 to 31000 of the total recharge amounts.

```
1 plt.figure(figsize=(10,6))
2 sns.boxplot(y='sumamnt_ma_rech30',x='amnt_loans30',data=df)
```

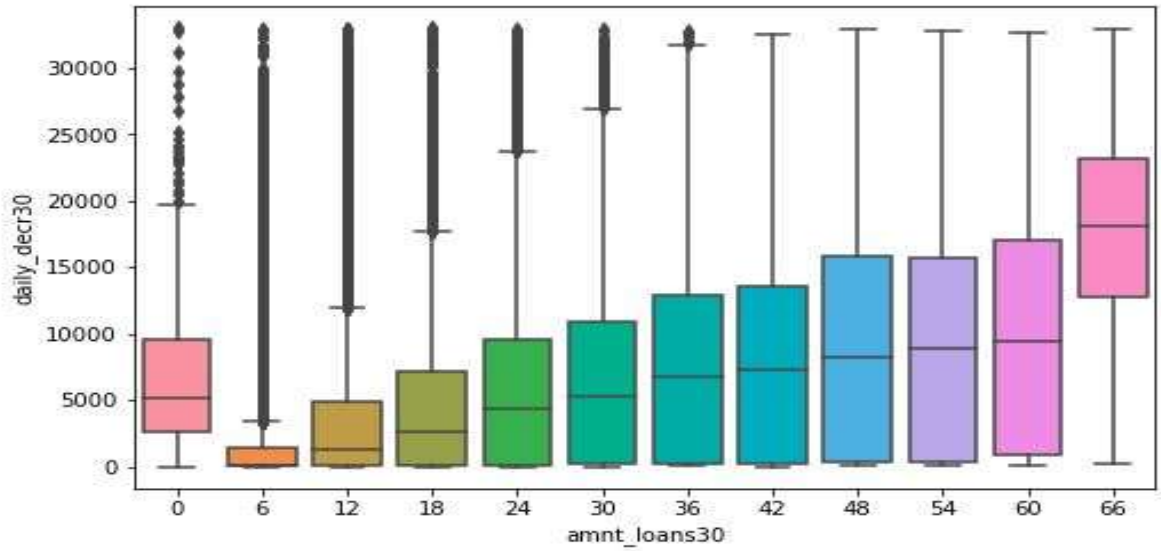
```
<AxesSubplot:xlabel='amnt_loans30', ylabel='sumamnt_ma_rech30'>
```



For the loan amount 6, daily amount spent is ranging from 0 to 4000. For the loan amount 18, daily amount spent is ranging from 0 to 17500. For the loan amount 42,48,54,60 and 66, daily amount spent is ranging from 0 to 40000.

```
1 plt.figure(figsize=(8,5))
2 sns.boxplot(y='daily_decr30',x='amnt_loans30',data=df)
```

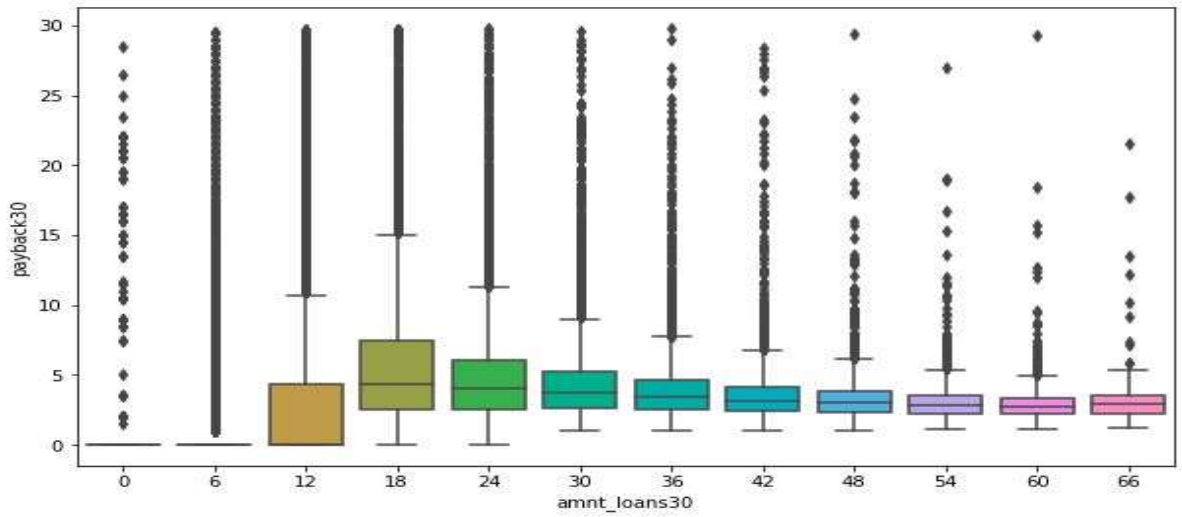
```
<AxesSubplot:xlabel='amnt_loans30', ylabel='daily_decr30'>
```



For the loan amount 12 the payback30 ranging from 0 to 11 days. For the loan amount 18 the payback30 ranging from 0 to 15 days. For the loan amount 60 the payback30 ranging from 0 to 5 days.

```
plt.figure(figsize=(10,6))
sns.boxplot(x='amnt_loans30',y='payback30',data=df)
```

```
<AxesSubplot:xlabel='amnt_loans30', ylabel='payback30'>
```



- Interpretation of the Results

From the visualizations is shows that the customers taking loan amount 12,18,24 and 30 are taking more than 5 days to payback compared to other loan amount taking users.

CONCLUSION

- Key Findings and Conclusions of the Study

  The output variable label is depending on the input variables like number of times recharged, total amount of recharge and number loans taken. These input variables play an important role in predicting the customers who are going to payback the loan amount within the time period. The customers who are taking loan amount of 12,18,24,30 is taking more than 5 days to payback.

- Learning Outcomes of the Study in respect of Data Science

  For predicting whether the customer will be paying back the loaned amount within 5 days or not, four machine learning algorithms are used i.e., Logistic Regression, Decision tree, Random Forest and Gradient boosting.

  The Gradient boosting model is the best model which gives the accuracy score of 85.91% and cross validation score 85.83%.