Project Micro credit loan Prediction

Submitted By:

Aman Saxena

ACKNOWLEDGMENT

- https://scikit-learn.org/stable/ - For the libraries used in the project.
- Rest project is done by myself only.

INTRODUCTION

- Business Problem Framing

The dataset is of the Micro credit loan. 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). . 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. Build a model which can be used to predict in terms of a probability for each loan transaction, whether the customer will be paying back the loaned amount within 5 days of insurance of loan. In this case, Label '1' indicates that the loan has been payed i.e. Non- defaulter, while, Label '0' indicates that the loan has not been payed i.e. defaulter.

.

- Conceptual Background of the Domain Problem

For more understanding we can simply correlate it with the loan payback or not if the customer have cleared the loan or not.

- Motivation for the Problem Undertaken

In this project we have to build the model that will predict whether the customer have clear the loan or not using the independent variables in order to improve the selection of customers for the credit.

Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem
- First of all we load the dataset from the local. Then we check the shape of the dataset and then describe the dataset from that we have gather some information. Then we check the datatypes of the columns and found that some of the columns are of object datatype. After that we check for the columns which have 0 values but we did not remove them as these were the possible outcomes. Then we check the correlation of the columns with the target variable and remove the columns which have very less relation or no relation with the dataset. Then the visualization of the categorical and continuous columns were done seperately.By looking at the distribution of the continuous data we observed that there was lots of skewness in the data. .After that we check for the outliers using the boxplot we found lots of outliers in the columns so we remove some of the outliers using the z score technique as we remove more outliers so we loss most og the data .
- Since the categorical values were present so we have to encode them to make the prediction for that we have used ordinal encoder.
- Then the dataset is divided into dependent and independent variables. Then the data is trained on different models and Random forest classifier is giving the best accuracy and the other metrices are also good in that so we chose it our best model .Then

the hyperparameter tuning is done . Then ROC AUC curve is plotted and the area under curve is 93% which is good.

.

- Data Sources and their formats

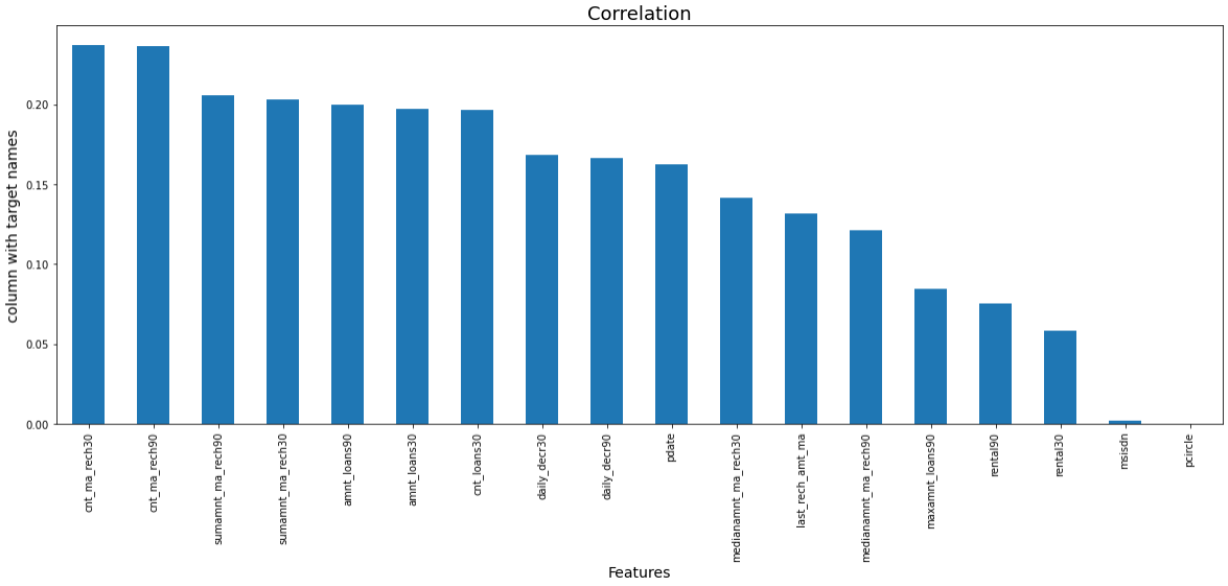
We load the dtatset and check their datatypes so we observe that all the columns are of object datatype and then we do the encoding of the dataset so that we can predict that which customer have cleared the load.

Data Preprocessing Done

- First of all we load the dataset from the local. Then we check the shape of the dataset and then describe the dataset from that we have gather some information. Then we check the datatypes of the columns and found that some of the columns are of object datatype. After that we check for the columns which have 0 values but we did not remove them as these were the possible outcomes. Then we check the correlation of the columns with the target variable and remove the columns which have very less relation or no relation with the dataset. Then the visualization of the categorical and continuous columns were done seperately.By looking at the distribution of the continuous data we observed that there was lots of skewness in the data. .After that we check for the outliers using the boxplot we found lots of outliers in the columns so we remove some of the outliers using the z score technique as we remove more outliers so we loss most og the data .

- Since the categorical values were present so we have to encode them to make the prediction for that we have used ordinal encoder.
- Then the dataset is divided into dependent and independent variables. Then the data is trained on different models and Random forest classifier is giving the best accuracy and the other metrices are also good in that so we chose it our best model .Then the hyperparameter tuning is done . Then ROC AUC curve is plotted and the area under curve is 93% which is good.

- Data Inputs- Logic- Output Relationships



In the above picture we can clearly see that what features are positively related with the target variable and what features are negatively related with the target variable.

- Hardware and Software Requirements and Tools Used

- import pandas as pd
- import numpy as np
- from sklearn.tree import DecisionTreeClassifier
- from sklearn.linear_model import LogisticRegression
- from sklearn.model_selection import train_test_split,GridSearchCV
- from statsmodels.stats.outliers_influence import variance_inflation_factor
- from sklearn.preprocessing import StandardScaler
- from sklearn.metrics import accuracy_score,confusion_matrix,roc_curve,roc_auc_score,classification_report
- import matplotlib.pyplot as plt
- import seaborn as sns
- import warnings
- warnings.filterwarnings('ignore')

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

Since the problem is of classification and we have to predict the binary output so the approaches which we can use are Logistic Regression, Decision tree classifier,Random forest classifier,SVC and all the given approaches were used in our project.

- Testing of Identified Approaches (Algorithms)
  - Logistic Regression
  - Decision tree classifier
  - Random forest Classifier
  - SVC

- Run and Evaluate selected models
  - Logistic Regression

```
LR=LogisticRegression()
LR.fit(x_train,y_train)
y_pred=LR.predict(x_test)
accuracy=accuracy_score(y_test,y_pred)
print(accuracy*100)
print(confusion_matrix(y_test,pred))
print(classification_report(y_test,pred))
```

```
88.33919986263736
[[  202  5919]
 [ 1230 39241]]
              precision    recall  f1-score   support

           0       0.14      0.03      0.05      6121
           1       0.87      0.97      0.92     40471

    accuracy                           0.85     46592
   macro avg       0.50      0.50      0.49     46592
weighted avg       0.77      0.85      0.80     46592
```

- Decision tree classifier

Decision Tree Classifier ¶

```
from sklearn.tree import DecisionTreeClassifier
DT=DecisionTreeClassifier()
DT.fit(x_train,y_train)
pred=DT.predict(x_test)
print('Accuracy ',accuracy_score(y_test,pred)*100)
print(confusion_matrix(y_test,pred))
print(classification_report(y_test,pred))
```

```
Accuracy  87.43131868131869
[[ 3431  2690]
 [ 3166 37305]]
              precision    recall  f1-score   support

           0       0.52      0.56      0.54      6121
           1       0.93      0.92      0.93     40471

    accuracy                           0.87     46592
   macro avg       0.73      0.74      0.73     46592
weighted avg       0.88      0.87      0.88     46592
```

- Random Forest Classifier

Random Forest Classifier

```
from sklearn.ensemble import RandomForestClassifier
RF=RandomForestClassifier()
RF.fit(x_train,y_train)
pred=RF.predict(x_test)
print('Accuracy ',accuracy_score(y_test,pred)*100)
print(confusion_matrix(y_test,pred))
print(classification_report(y_test,pred))
```

```
Accuracy  91.23025412087912
[[ 3240  2881]
 [ 1205 39266]]
              precision    recall  f1-score   support

           0       0.73      0.53      0.61      6121
           1       0.93      0.97      0.95     40471

    accuracy                           0.91     46592
   macro avg       0.83      0.75      0.78     46592
weighted avg       0.91      0.91      0.91     46592
```

➢ SVC

SVC

```
1 from sklearn.svm import SVC
2 svc=SVC()
3 svc.fit(x_train,y_train)
4 pred=svc.predict(x_test)
5 print('Accuracy ',accuracy_score(y_test,pred)*100)
6 print(confusion_matrix(y_test,pred))
7 print(classification_report(y_test,pred))
```

```
Accuracy  87.06859546703298
[[  153  5968]
 [   57 40414]]
              precision    recall  f1-score   support

           0       0.73      0.02      0.05      6121
           1       0.87      1.00      0.93     40471

    accuracy                           0.87     46592
   macro avg       0.80      0.51      0.49     46592
weighted avg       0.85      0.87      0.81     46592
```

- Key Metrics for success in solving problem under consideration

  accuracy_score → for calculating the accuracy
  confusion_matrix →to see how many are rightly predicted.
  Classification report → precision,recall etc.
  Cross Val Score → For cross validation

CONCLUSION

- First of all we load the dataset from the local. Then we check the shape of the dataset and then describe the dataset from that we have gather some information. Then we check the datatypes of the columns and found that some of the columns are of object datatype. After that we check for the columns which have 0 values but we did not remove them as these were the possible outcomes. Then we check the correlation of the columns with the target variable and remove the columns which have very less relation or no relation with the dataset. Then the visualization of the categorical and continuous columns were done seperately.By looking at the distribution of the continuous data we observed that there was lots of skewness in the data. .After that we check for the outliers using the boxplot we found lots of outliers in the columns so we remove some of the outliers using the z score technique as we remove more outliers so we loss most og the data .
- Since the categorical values were present so we have to encode them to make the prediction for that we have used ordinal encoder.
- Then the dataset is divided into dependent and independent variables. Then the data is trained on different models and Random forest classifier is giving the

best accuracy and the other metrices are also good in that so we chose it our best model .Then the hyperparameter tuning is done . Then ROC AUC curve is plotted and the area under curve is 93% which is good.