MICRO CREDIT DEFAULTER CASE PROJECT

Submitted by:

Anita Thapa

ACKNOWLEDGMENT

I would like to express my special thanks of gratitude to my SME Swati Ma’am for giving me an opportunity to do this project on topic Micro credit defaulter case study. I have a lot of research on this project by using various internet sites like towardsdatascience, medium and Wikipedia which helped me understand the main problem statement and also in finding the solutions.

INTRODUCTION

- ## Business Problem Framing

  The project is based on predicting the loan defaulters from the given dataset as the projects is based on finding the non defaulter customers who will pay back the loan provided by our client while partnering with Micro Finance Institute.

- ## Conceptual Background of the Domain Problem

  The main concept behind this project of Micro Credit defaulter case is to understand the Micro finance institute and its services. In the given dataset telecom industry is providing its users with loan facilities and based upon its users recharge and loan history it is predicting where the loan will be repaid by its customers within 5 days. For predicting the value we have the history of recharge done, balance maintained, loan taken and loan repaid by the customers in the span of 30 and 90 days.

- ## Review of Literature

  The research was done on the topic Micro Finance Institute in order to understand its functioning in providing financial services to the customers. Then the dataset given was properly studied to understand each and every column present so better prediction.

- ## Motivation for the Problem Undertaken

  Main motivation for doing this project was to understand the Micro Finance Institute and its services as they are treated as Poverty reduction tool and as we know Poverty is a huge problem in maximum all the countries. This project helped me analyse and predict the best model for a real time dataset for improving upon the knowledge.

Analytical Problem Framing

- Mathematical/ Analytical Modelling of the Problem

  The problem dataset was solved in the Jupyter notebook and there various libraries were used to read and analyse the dataset like Numpy, pandas, matplotlib, seaborn, scikit etc. The dataset was converted into dataframe using pandas library, then using plotting libraries like matplotlib and seaborn the data presented information in graphical way. The object datatype was encoded to numeric datatype as Machine Learning works on numeric datatype. Then to remove imbalance in the target label SMOTE library was used. Skewness, multicollinearity and outliers were also taken care of. And finally before modeling the dataset, it was scaled using Standard scaler and after that 5 Machine Learning Algorithm like Decision Tree, Random Forest etc were used to model the dataset. After selecting the best model, the final model was saved using pickle library.

- Data Sources and their formats

  What are the data sources, their origins, their formats and other details that you find necessary? They can be described here. Provide a proper data description. You can also add a snapshot of the data.

  The data was in the csv format which is comma separated file. Then it was converted into dataframe format in Jupyter notebook using panda library.

Imported Dataset

```
import pandas as pd
import numpy as np
import warnings
warnings.filterwarnings('ignore')
```

```
ds=pd.read_csv("Data file.csv")
creditdf=pd.DataFrame(ds)
```

Using the head () the above 5 data was visible like shown in the picture below.

```
creditdf.head()
```

	Unnamed: 0	label	msisdn	aon	daily_decr30	daily_decr90	rental30	rental90	last_rech_date_ma	last_rech_date_da	...	maxamnt_
0	1	0	21408I70789	272.0	3055.050000	3065.150000	220.13	260.13	2.0	0.0	...	
1	2	1	76462I70374	712.0	12122.000000	12124.750000	3691.26	3691.26	20.0	0.0	...	
2	3	1	17943I70372	535.0	1398.000000	1398.000000	900.13	900.13	3.0	0.0	...	
3	4	1	55773I70781	241.0	21.228000	21.228000	159.42	159.42	41.0	0.0	...	
4	5	1	03813I82730	947.0	150.619333	150.619333	1098.90	1098.90	4.0	0.0	...	

5 rows × 37 columns

The dataset contains 209593 rows and 37 columns, there are no null values present in the dataset. Data description is found out using describe() code.

```
creditdf.describe()
```

	msisdn	aon	daily_decr30	daily_decr90	rental30	rental90	last_rech_date_ma	last_rech_date_da
count	209593.000000	209593.000000	209593.000000	209593.000000	209593.000000	209593.000000	209593.000000	209593.000000
mean	93100.650179	8112.343445	5381.402289	6082.515068	2692.581910	3483.406534	3755.847800	3712.202921
std	53758.461427	75696.082531	9220.623400	10918.812767	4308.586781	5770.461279	53905.892230	53374.833430
min	0.000000	-48.000000	-93.012667	-93.012667	-23737.140000	-24720.580000	-29.000000	-29.000000
25%	46506.000000	246.000000	42.440000	42.692000	280.420000	300.260000	1.000000	0.000000
50%	93073.000000	527.000000	1469.175667	1500.000000	1083.570000	1334.000000	3.000000	0.000000
75%	139626.000000	982.000000	7244.000000	7802.790000	3356.940000	4201.790000	7.000000	0.000000
max	186242.000000	999860.755168	265926.000000	320630.000000	198926.110000	200148.110000	998650.377733	999171.809410

8 rows × 35 columns

From looking into data it is clear that it is a Classification problem and the target column is ‘label’.

- Data Pre-processing Done

Using isnull().sum() we found out that the dataset doesn’t have null values. Then using info() we got the information about the dataset that there are 209593 rows and 37 columns, and 34 columns are numeric datatype and the 3 columns are object datatype.

Unique values of each column was checked to search for nan values but there were no nan values also. Hence the data was then used for visualization process for further insights.

- Data Inputs- Logic- Output Relationships

  The data was in the csv file format and then after the data was converted into dataframe format having rows and columns for better presentation of the data.

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

  The columns like unnamed :0 was removed from dataset assuming it to be not related to dataset. Also later while performing skewness certain other columns like fr_da_rech90,medianmarechprebal30 and last_rech_date_da were removed as they were having high skewness value and also were negatively correlated with the target column.

- Hardware and Software Requirements and Tools Used

  Listing down the hardware and software requirements along with the tools, libraries and packages used. Describe all the software tools used along with a detailed description of tasks done with those tools.

  In the hardware only Laptop with 4gb RAM was used and in the software Jupyter notebook was used for the problem.

  Libraries used were:

  1. Numpy : It is a Numerical Python library used for numerical computation like arrays in Python.

  2. Panda: It was used for reading and converting csv file into dataframe.

  4. matplotlib: this library was used for plotting the graph in the EDA.

  5. Seaborn: This library is also used for visualization as it has helped to plot different types of plots like countplot, displot, boxplot and heatmap in the notebook.

6. Label encoder: It helped to encode the object datatype into numeric datatype as the Machine learning algorithm works on numeric datatype only.

7. imblearn and SMOTE: this library was used for class imbalance removal

8. VIF: this library was used for finding the value of multicollinearity in the dataset

9. sklearn.preprocessing and power transform: It was used to remove the skewness from the dataset.

10. Standard Scaler : It was used to scale the feature column of the dataset before model selection

11. sklearn: This library was used to import train_test_split, metrics like accuracy score, classification report and also importing Algorithms likes Logistic Regression, Decision Tree Classifier, SVC, Rnadom Forest Classifier and K-Neigbor Classifier.

12. pickle: This library was used to save the final model.

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

  The problem was to predict the whether the customer of the telecom industry will pay back the loan within 5 days or not. For this we firstly scaled the data after doing proper EDA. Then using logistic regression found out the accuracy of the training and testing data and after that used train and test split to predict the value of testing data. Imported metric library to find out the accuracy score of the training data. Also used GridsearchCV to find out the best parameters for each algorithm for better results. Also printed the AUC-ROC curve for each algorithm.

- Testing of Identified Approaches (Algorithms)

  Algorithms used for the training and testing are:

  1. Logistic Regression
  2. Decision Tree Classifier
  3. SVC
  4. Random Forest Classifier
  5. K-Neighbors Classifier

- Run and Evaluate selected models

  Out of all the 5 algorithm used the best algorithm used is Decision Tree Classifier as the accuracy score and cross validation score difference is less.

Decision Tree classifier

```
from sklearn.model_selection import GridSearchCV
from sklearn.tree import DecisionTreeClassifier

dtc=DecisionTreeClassifier()
grid_param={
    'criterion':['gini','entropy']
  }
gd_sr=GridSearchCV(estimator=dtc,
                   param_grid=grid_param,
                   scoring='accuracy',
                   cv=9)
gd_sr.fit(train_x,train_y)
best_parameters=gd_sr.best_params_
print(best_parameters)
best_result=gd_sr.best_score_
print(best_result)

print(round(best_result,2))
```

```
{'criterion': 'entropy'}
0.8863985658357513
0.89
```

```
#using best parameters for decision tree classifier
dtc= DecisionTreeClassifier(criterion='entropy')
dtc.fit(train_x,train_y)
dtc.score(train_x,train_y)
```

```
0.9999821081384115
```

```
#finding cross validation score

dtscore = cross_val_score(dtc,x,y,cv=9)
dts = dtscore.mean()
print('Cross Val Score:',dts*100)
```

```
Cross Val Score: 88.65515587564876
```

Cross val score is 88.65%

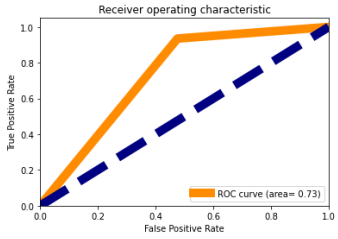
```
#Predicted value
pred = dtc.predict(test_x)
from sklearn.metrics import accuracy_score, confusion_matrix
accuracy=accuracy_score(test_y,pred)
confusion= confusion_matrix(test_y,pred)
print("Accuracy of the model is: ",accuracy)
print("Confusion Matrix: ", confusion)
```

```
Accuracy of the model is:  0.8814380114029438
Confusion Matrix:  [[ 2869  2366]
 [ 2604 34080]]
```

Accuracy score is 88.1%

```
from sklearn.metrics import roc_curve, auc
fpr, tpr, thresholds= roc_curve(pred,test_y)
roc_auc = auc(fpr,tpr)

plt.figure()
plt.plot(fpr, tpr, color='darkorange',lw=10, label='ROC curve (area= %0.2f)' % roc_auc)
plt.plot([0,1],[0,1], color='navy', lw=10,linestyle='--')
plt.xlim([0.0,1.0])
plt.ylim([0.0,1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver operating characteristic')
plt.legend(loc='lower right')
plt.show()
```



AUC ROC Curve id 73% which is good.

- Key Metrics for success in solving problem under consideration

  The main metric used was to find out the accuracy of the model which helped to determine the best model when compared with the cross

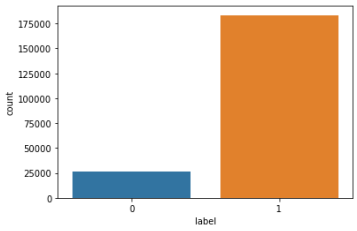
validation score. For model selection the algorithm having less difference between cross validation score and accuracy is considered to be the best model.

- Visualizations

Different plots were used in the problem like displot, countplot, scatter plot, heatmap and AUC ROC curve using matplotlib and seaborn library as shown in the images below.

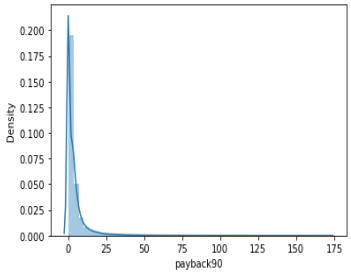
```
#plotting target column:'label'
ax=sns.countplot(x="label", data=creditdf)
print(creditdf["label"].value_counts())
```

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



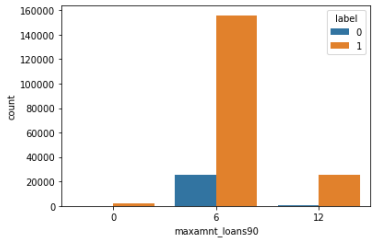

```
sns.distplot(creditdf['payback90'], kde=True)
```

```
<AxesSubplot:xlabel='payback90', ylabel='Density'>
```



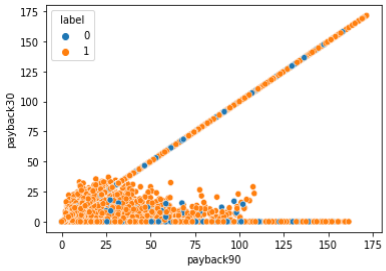

```
sns.countplot(x='maxamnt_loans90', hue='label', data = creditdf)
```

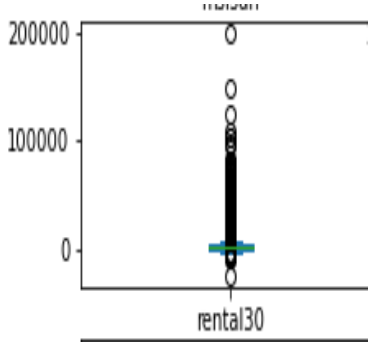
```
<AxesSubplot:xlabel='maxamnt_loans90', ylabel='count'>
```

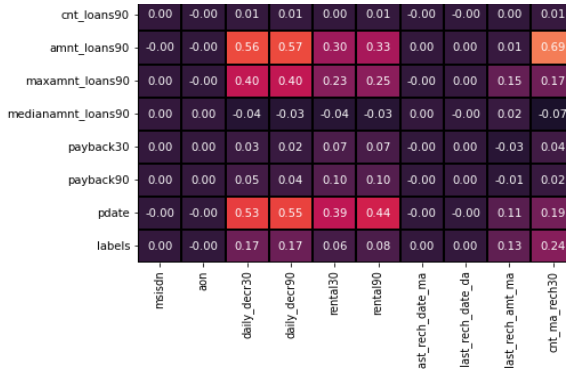


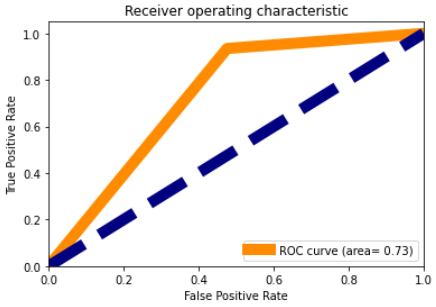

```
sns.scatterplot(x='payback90', y='payback30',hue='label', data = creditdf)
```

```
<AxesSubplot:xlabel='payback90', ylabel='payback30'>
```









- Interpretation of the Results

  From the visualizations, preprocessing and modeling of the data it was interpreted that maximum customers are the non defaulters of loan and they can be offered loan services in future by our client. The most correlated data with the target column were cnt_ma_rech30 and cnt_ma_rech_90 which shows the number of times the account got recharged in the span of 30 and 90 days. As per the model the data can be predicted with 88% accuracy.

CONCLUSION

- Key Findings and Conclusions of the Study

  Key findings in the whole problem were the class imbalance in the target value which was balanced using SMOTE technique. In the dataset a larger chunks of data is having outliers but they were removed as the data is precious so used the data judiciously. In the project the model is approximately able to predict the defaulters from the list of customers of the client.

- Learning Outcomes of the Study in respect of Data Science

  Using Visualization libraries like matplotlib and seaborn it was easy to find the correlation between the dataset columns and the plotting of each column along the target column helped to analyse the problem pretty well. As visualization helped to get better insight of the data like class imbalance, skewness present in the data.

- Limitations of this work and Scope for Future Work

  As the model accuracy is 88.14% so it can approximately predict the correct value as sometimes it might predict wrong values but chances are less. This work can really help the client to boost their collaboration with the MFI as by using this prediction the client will be able to approximately predict the customer to which they can offer loan and who will pay back loan positively.