FLIP ROBO

# Micro-Credit Defaulter Model

Submitted By:

**Komal Ghatvilkar**

## ACKNOWLEDGMENT

I Would Like To Thanks FlipRobo & Team To Support And Helped Me To Complete This Project Successfully. I Got So Much Relatable Information From Sites Like Google, Kaggle, GitHub Etc. For The Research.

I Am Happy By Getting The Guidance And Help From These And I Would Not Be Able To Complete This Project Without It.

As Per The Research For A Period Of Time From The 1980s To The Early 2000s, "Microloans" Or "Microfinance" Were All The Rage In International Development.

The Idea Was Simple Enough Was That By Giving A Very Small Loan To Someone Living In A Poor Country, You Could Help Them Expand A Small Business, Which Would Lift Their Family Out Of Poverty. When They Pay Back The Loan, The Money Can Be Cycled To More Borrowers, Getting More Families Out Of Poverty. Organizations Offering Microcredit To Poor Borrowers Many Living On Very Less Per Day Took Off In Those Decades. Investors And Donors Poured Money Into Microcredit, Hundreds Of Organizations Offered Loans. The Microcredit Movement Has Been Undeniably Successful In Opening Up Financial Services To Poor People Across Many Countries.

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 Of The Client That Is In Telecom Industry. 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, Payback Amount Should Be 6, While, For The Loan Amount Of 10, The Payback Amount Should Be 12. 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.

The Dataset Provided By The Client To Us Containing Approximately 210000. Data Has The Information About Every Individual's Expenditure And Other Information Of Recharge Done In Every Month And Every 3 Months. It Also Shows The Loan Amount Related Information Taken By The User.

I Have Tried To Do The Possible R&D On The Data And General Information About Microcredit Concept. With That Done The Changes In The Data Needed As Per My Knowledge For The Research. The Aim Of This Study Is To Investigate The Customers Who Are Really Needed This Opportunity So That It Will Help To Reduce The Poverty.

I Used My Laptop To Work In, Search Engines Like Google To Find Out The Information, Get Help With Different Sites Like GitHub, Kaggle To Learn About The Project More, Used Software MS-Office, Jupyter Notebook To Perform The Tasks In It.

We have done the following analysis of the dataset where we Imported necessary libraries so that we can work on datasets with the Jupyter notebook.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings('ignore')
```

Data contains 209593 entries each having 19 variables. After reading the dataset I done the necessary changes in dataset by doing the EDA.

I removed some unnecessary columns which was not needed to  keep in dataset. Those are :

```
df.drop({'Unnamed:0','msisdn','daily_decr30','rental30','pcircle','fr_ma_rech30','cnt_ma_re
ch30','pdate','sumamnt_ma_rech30','medianamnt_ma_rech30','medianmarechprebal30','cnt_da_rec
h30','fr_da_rech30','cnt_loans30','amnt_loans30','maxamnt_loans30','medianamnt_loans30','pa
yback30'},axis=1,inplace=True)
```

I checked the description of data with .info() method  & find no null values contains in dataset.

```
df.info()
```

After .describe() done found the statistical  description of data & with the correlation among all the columns checked the correlation and found most of the data is positively correlated with each other.

```
df.describe()
df.corr()
```

In data visualization done the following visualizations :
First used Correlation Matrix for showing the correlation between all columns with Heatmap.

From the output of correlation matrix, we can see that it is symmetrical i.e. the bottom left is same as the top right. It is also observed that most of the variables are positively correlated with each other.

```
# Correlation Matrix
corr_mat=df.corr()
# Size of the canvas
plt.figure(figsize=[30,30])
#Plot Correlation Matrix
sns.heatmap(corr_mat,annot=True) # annot represnts each value encoded in heatmap
plt.title('Correlation Matrix')
plt.show()
```

The result of the Correlation Matrix is on following GitHub link.

https://github.com/komalghatvilkar/Internship/blob/main/Micro-Credit%20Defaulter/Correlation%20Matrix.png

Second, used Density plot & Histogram both for all dataset for visualizing the data individually.
The visualization plot shows that each variable distributed differently and as we can see some data has categorical values, so used histogram also for better understanding to show the distribution.

```
df.plot(kind='density',subplots=True,layout=(5,10),sharex=False,fontsize=1,figsize=(30,20))
plt.show()

# plot histogram data vizualization
df.hist(bins=20,figsize=(20,20))
#plot showing
plt.show()
```

The result of the Density plot & Histogram is on following GitHub link.

https://github.com/komalghatvilkar/Internship/blob/main/Micro-Credit%20Defaulter/Density%20Plot.png

https://github.com/komalghatvilkar/Internship/blob/main/Micro-Credit%20Defaulter/Histogram.png

I split the data into x & y as it is categorical data & taken the output as the column "'label'" because our problem statement is to predict the defaulters of loan and then checked and removed the skewness of the data. After that checked the outliers if any, found many so tried to  remove the outliers. I used boxplot to check the outliers in dataset.

```
x=df.drop('label',axis=1) # Split the data
```

```
y=df['label']

x.skew().sort_values(ascending=False) # For descending

from sklearn.preprocessing import power_transform
x_new=power_transform(x)
x=pd.DataFrame(x_new,columns=x.columns)

x.boxplot(figsize=[20,15])
plt.subplots_adjust(bottom=0.25)
plt.show()

from scipy.stats import zscore
z=np.abs(zscore(x))
z.shape

threshold=3
print(np.where(z>3))

x_new=x[(z<3).all(axis=1)]
print(x.shape)
print(x_new.shape)

loss_percent=(209593-177740)/209593*100
print(loss_percent)

y_new=y[(z<3).all(axis=1)]
print(y.shape)
print(y_new.shape)

x_new.boxplot(figsize=[20,15])
plt.subplots_adjust(bottom=0.25)
plt.show()
```

The result of the Boxplot is on following GitHub link.

https://github.com/komalghatvilkar/Internship/blob/main/Micro-Credit%20Defaulter/Outliers.png

And then did the Train Test Split and find the best accuracy & random state.

```
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
from sklearn.metrics import confusion_matrix,classification_report
from sklearn.model_selection import train_test_split
maxAccu=0 # max accuracy
maxRS=0 # best random state value for which max accuracy is achieved
for i in range(0,200):

x_train,x_test,y_train,y_test=train_test_split(x_new,y_new,test_size=.20,random_state=i)
    LR=LogisticRegression()
    LR.fit(x_train,y_train) # Fitting the data will train the model
    predrf=LR.predict(x_test) # Predicting the data # Predicted traget variable
    acc=accuracy_score(y_test,predrf) # two target varaible # accuracy score
    if acc>maxAccu:
        maxAccu=acc
        maxRS=i
print("Best accuracy is ",maxAccu,"on Randon_state ",maxRS)
```

Then performed the model building and used best possible model building techniques for classification dataset to perform The tasks.

```
from sklearn.linear_model import LogisticRegression
LR=LogisticRegression()
LR.fit(x_train,y_train)
predlr=LR.predict(x_test)
print("Accuracy",accuracy_score(y_test,predlr)*100)
print(confusion_matrix(y_test,predlr))
print(classification_report(y_test,predlr))

from sklearn.tree import DecisionTreeClassifier
dt=DecisionTreeClassifier()
dt.fit(x_train,y_train)
preddt=dt.predict(x_test)
print("Accuracy",accuracy_score(y_test,preddt)*100)
print(confusion_matrix(y_test,preddt))
print(classification_report(y_test,preddt))

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

Next did Cross Validation with "cross_val_score" for the models used & it shows the output :-

```
from sklearn.model_selection import cross_val_score
scr=cross_val_score(LR,x_new,y_new,cv=5)
print("Cross Validation Score Of Logistic Regression Model :",scr.mean())
scr1=cross_val_score(LR,x_new,y_new,cv=5)
print("Cross Validation Score Of Decision Tree Model :",scr1.mean())
scr2=cross_val_score(rf,x_new,y_new,cv=5)
print("Cross Validation Score Of Random Forest Model :",scr2.mean())
#scr3=cross_val_score(svc,x_new,y_new,cv=5)
#print("Cross Validation Score Of SVC Model :",scr3.mean())
```

Then with the GridSearchCV checked hyper parameter tuning & showed the best accuracy of data.

```
from sklearn.model_selection import GridSearchCV

# Create parameters list to pass in GridSearchCV

parameters={'max_features':['auto','sqrt','log2'],'max_depth':[1,2,3,4,5],'criterion':['gin
i','entropy']}

GCV=GridSearchCV(RandomForestClassifier(),parameters,cv=2,scoring="accuracy")

GCV.fit(x_train,y_train) # fitting the data in model

GCV.best_params_ # Printing the best parameter found by GridSearchCV

GCV_pred=GCV.best_estimator_.predict(x_test) # predicting with best parameters

accuracy_score(y_test,GCV_pred) # checking final accuracy
```

Then with the ROC curve to see the performance measurement for the dataset at various threshold settings.

```
from sklearn.metrics import plot_roc_curve
plot_roc_curve(GCV.best_estimator_,x_test,y_test)
plt.title("ROC AUC Plot")
plt.show()
```

Finally came to the Conclusion as per the results found those are the model is showing the exact result for the dataset with 86% accuracy.

Please find the GitHub links for ROC-AUC curve to refer.

https://github.com/komalghatvilkar/Internship/blob/main/Micro-Credit%20Defaulter/ROC%20AUC%20Plot.png

Please find the GitHub links for Jupyter Notebook Solution of dataset to refer.

https://github.com/komalghatvilkar/Internship/blob/main/Micro-Credit%20Defaulter/Micro-Credit%20Defaulter%20Model.ipynb

I found that the data has many outliers and we can't remove all of them so i tried to push myself as much as possible to reduce it and give a cleaned data for process further. I observed that There are two features of same concept for e.g recharge done for 30 days and 90 days so I took 90 days as it will give no major changes if i take 90 days instead 30 days according to me.

There is a hope and trust with the analysis of this project that we can try to reduce the poverty and improve the standard of living of people with the helping hand of microfinance. Analysis will give the exact idea of the correctness of data provided with which microfinance providers can decide easily what to do next but the data provided to us should be true.

The results of the data analysis shows that the dataset is correct 86% & it is not totally perfect but we can take this data for use.

Thank you...!!