Micro Credit Defaulter

Submitted by:
VAIBHAV GARG

ACKNOWLEDGMENT

This includes mentioning of all the references, research papers, data sources, professionals and other resources that helped you and guided you in completion of the project.

INTRODUCTION

• Business Problem Framing

MFI(Micro Finance Institute) is an organization that offers financial services(Loans) to low income populations. Here, we need to Predict if the customer is Defaulter i:e unable to pay back the Loan before time (5 days) or non-Defaulter i:e able to pay back within time duration of (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.

• Conceptual Background of the Domain Problem

Describe the domain related concepts that you think will be useful for better understanding of the project.
Here, MSI is providing loans to the low income population and on the basis of different details of a customer we need to find out if the customer can pay back the loan Amount within the given time or not.

• Review of Literature

This is a comprehensive summary of the research done on the topic. The review should enumerate, describe, summarize, evaluate and clarify the research done.
We researched on different details of the customer like

- From how long customer is on the certain mobile network.
- Daily amount spent from main account, averaged over last 30 days (in Indonesian Rupiah)
- Average Main Account Balance over last 90 days
- Number of Days till last recharge of Main Account
- Amount of the last recharge of the Main Account.
- Frequency of the Main Account Recharged in last 30 days
- Number of Loans Taken by the users in last 30 days
- Total Amount of Loans taken by the user in last 30 days

- Maximum Amount of loan taken by the user in last 30 days.

And many other parameters as well on the basis of which we need to predict if the individual can pay the given loan on time or Not.

- Motivation for the Problem Undertaken

Describe your objective behind to make this project, this domain and what is the motivation behind.

Our Main Objective to make this project or Machine Learning model is to predict if the loan has to be giving to the particular individual or not on the basis of there past track record and from this ML model we can save huge amount of money as it dramatically reduce the possibility that MSI approve the Loan for someone who is not going to pay within Time frame given.

Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem

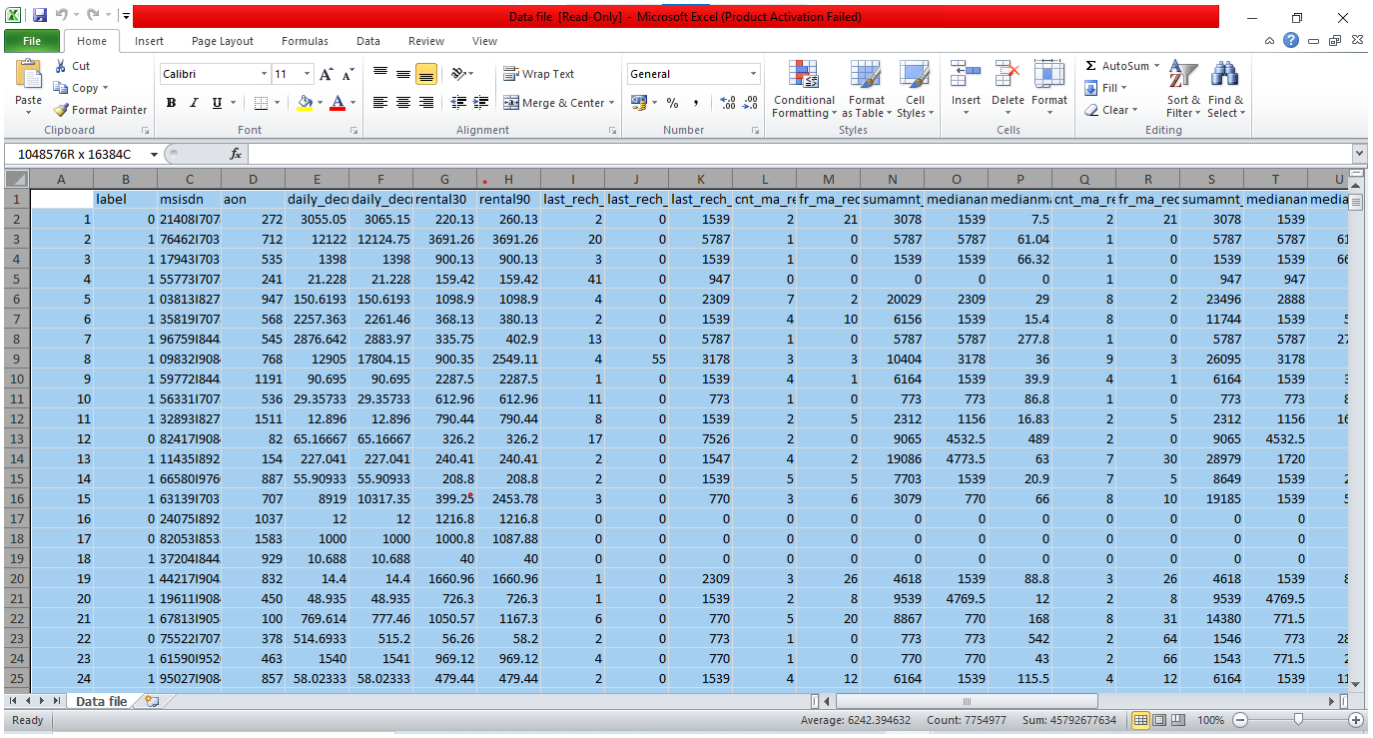
Describe the mathematical, statistical and analytics modelling done during this project along with the proper justification.

- 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.

We Got the Data from MSI so that on the basis of that given data we can build our ML model. The data we got are in Excel format and we converted the data in CSV format.

Here is the Screen Shot of the Data we have for this Project:



- Data Preprocessing Done

What were the steps followed for the cleaning of the data? What were the assumptions done and what were the next actions steps over that?
Firstly, We checked in this project that if there is any NaN value present in this Dataset.

Then, we have to figure out which is going to be our Target variable and our Independent variables(x) after this we can remove some columns as they are not going to put any effect on our target variable and some of them are negatively correlated and put inverse effect on the End result. We can check for OUTLIERS and on the basis of that we have to remove certain rows where z value is greater then 3.

- Data Inputs- Logic- Output Relationships

Describe the relationship behind the data input, its format, the logic in between and the output. Describe how the input affects the output.
IN this project we have to analyse that if the Individual can payback the loan within the time duration of 5 days of issuing of the loan(0: Unable to payback & 1: Successfully payback the loan on time) and to analyse this we have different details from the users and on the basis of those details

we need to apply Machine Learning modelling to predict the if the individual can payback the load within Time or not.
We have the Data in Csv format and the attributes of the data like

- Daily amount spent from main account, averaged over last 90 days (in Indonesian Rupiah)
- Average main account balance over last 90 days
- Number of times main account got recharged in last 30 days
- Total amount of recharge in main account over last 30 days (in Indonesian Rupiah)
- Frequency of main account recharged in last 90 days
- Etc

Logic behind the input data is that we need to see how actively an individual is on his/her network and on the basis of that we can predict the success and the Failure ratio.

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

I haven't taken any assumptions in this project.

• 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.

The Hardware System Requirements:

→ Intel i5 Processor
→8 GB Ram
→2GB Graphics Card

Software requirements:

→Python Anaconda(version 3) Installed.

Libraries and Packages Used:

#NUMPY – Used for Numerical operations
#PANDAS – Use for Data Analysis and with the help of Pandas we can import our data file in Jupyter Notebook

#SKLEARN (SCIKIT-LEARN)- This is the Indispensible part of the Python and without this we can't work without Machine learning projects
#SEABORN –It is a library for making statistical graphics in Python, It provides a variety of visualization patterns.
#MATPLOTLIB- It is a plotting library for the Python programming language
#SCIPY – It Stands for Scientific Python. SciPy is a library that uses NumPy for more mathematical functions

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

Describe the approaches you followed, both statistical and analytical, for solving of this problem.

We start checking that if there is any NaN values present in the dataset and then will check for the outliers and work accordingly and then we have to train our model according to our target variable and then will find Accuracy score, confusion_matrix and classification_report.

- Testing of Identified Approaches (Algorithms)

Listing down all the algorithms used for the training and testing.

- SVM(Support Vector Machine)
- KNN(KNearestNeighbors)
- Decision Tree Classifier
- Random Forest Classifier
- Ada Boost Classifier
- Gradient Boost Classifier

- Run and Evaluate selected models

Describe all the algorithms used along with the snapshot of their code and what were the results observed over different evaluation metrics.

If we talk about the Algorithms we have used in this projects:

- Support Vector Machine:

In [36]:
```
svm=SVC()
svm.fit(x_train,y_train)
svm.score(x_train,y_train)
predsvm=svm.predict(x_test)
print('Accuracy Score =',accuracy_score(y_test,predsvm))
print(confusion_matrix(y_test,predsvm))
print(classification_report(y_test,predsvm))
```

```
Accuracy Score = 0.8772869022869023
[[    0  4722]
 [    0 33758]]
              precision    recall  f1-score   support

           0       0.00      0.00      0.00      4722
           1       0.88      1.00      0.93     33758

    accuracy                           0.88     38480
   macro avg       0.44      0.50      0.47     38480
weighted avg       0.77      0.88      0.82     38480
```

- **K NearestNeighbors:**

In [37]:
```
knn=KNeighborsClassifier()
knn.fit(x_train,y_train)
knn.score(x_train,y_train)
predknn=knn.predict(x_test)
print('Accuracy-Score=',accuracy_score(y_test,predknn))
print(confusion_matrix(y_test,predknn))
print(classification_report(y_test,predknn))
```

```
Accuracy-Score= 0.898024948024948
[[ 2014  2708]
 [ 1216 32542]]
              precision    recall  f1-score   support

           0       0.62      0.43      0.51      4722
           1       0.92      0.96      0.94     33758

    accuracy                           0.90     38480
   macro avg       0.77      0.70      0.72     38480
weighted avg       0.89      0.90      0.89     38480
```

- **Decision Tree Classifier:**

In [38]:
```
dtc=DecisionTreeClassifier()
dtc.fit(x_train,y_train)
dtc.score(x_train,y_train)
preddtc=dtc.predict(x_test)
print('Accuracy Score',accuracy_score(y_test,preddtc))
print(confusion_matrix(y_test,preddtc))
print(classification_report(y_test,preddtc))
```

```
Accuracy Score 0.870010395010395
[[ 2408  2314]
 [ 2688 31070]]
              precision    recall  f1-score   support

           0       0.47      0.51      0.49      4722
           1       0.93      0.92      0.93     33758

    accuracy                           0.87     38480
   macro avg       0.70      0.72      0.71     38480
weighted avg       0.87      0.87      0.87     38480
```

- **Random Forest Classifier:**

In [40]:
```
rfc=RandomForestClassifier(n_estimators=200,random_state=66)
rfc.fit(x_train,y_train)
rfc.score(x_train,y_train)
predrfc=rfc.predict(x_test)
print('Accuracy Score',accuracy_score(y_test,predrfc))
print(confusion_matrix(y_test,predrfc))
print(classification_report(y_test,predrfc))
```

```
Accuracy Score 0.9141112266112266
[[ 2162  2560]
 [  745 33013]]
              precision    recall  f1-score   support

           0       0.74      0.46      0.57      4722
           1       0.93      0.98      0.95     33758

    accuracy                           0.91     38480
   macro avg       0.84      0.72      0.76     38480
weighted avg       0.91      0.91      0.91     38480
```

- **Ada Boost Classifier:**

In [41]:
```
adc=AdaBoostClassifier(n_estimators=200,random_state=66)
adc.fit(x_train,y_train)
adc.score(x_train,y_train)
predadc=adc.predict(x_test)
print('Accuracy Score',accuracy_score(y_test,predadc))
print(confusion_matrix(y_test,predadc))
print(classification_report(y_test,predadc))
```

```
Accuracy Score 0.9085239085239085
[[ 1585  3137]
 [  383 33375]]
              precision    recall  f1-score   support

           0       0.81      0.34      0.47      4722
           1       0.91      0.99      0.95     33758

    accuracy                           0.91     38480
   macro avg       0.86      0.66      0.71     38480
weighted avg       0.90      0.91      0.89     38480
```

- **Gradient Boost Classifier:**

In [42]:
```
gbc=GradientBoostingClassifier()
gbc.fit(x_train,y_train)
gbc.score(x_train,y_train)
predgbc=gbc.predict(x_test)
print('Accuracy Score',accuracy_score(y_test,predgbc))
print(confusion_matrix(y_test,predgbc))
print(classification_report(y_test,predgbc))
```

```
Accuracy Score 0.9119282744282744
[[ 1754  2968]
 [  421 33337]]
              precision    recall  f1-score   support

           0       0.81      0.37      0.51      4722
           1       0.92      0.99      0.95     33758

    accuracy                           0.91     38480
   macro avg       0.86      0.68      0.73     38480
weighted avg       0.90      0.91      0.90     38480
```

- Key Metrics for success in solving problem under consideration

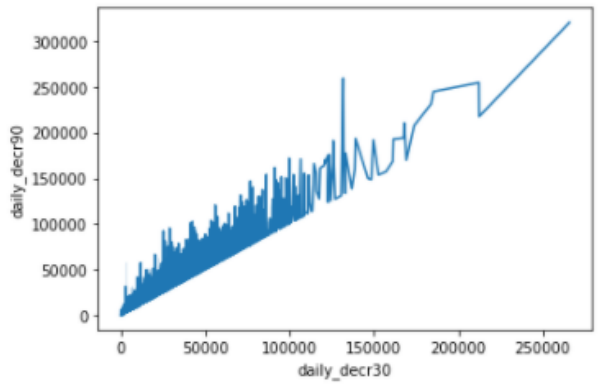
What were the key metrics used along with justification for using it? You may also include statistical metrics used if any.

Metrics used in this project:
ACCURACY SCORE
CONFUSION MATRIX
CLASSIFICATION REPORT

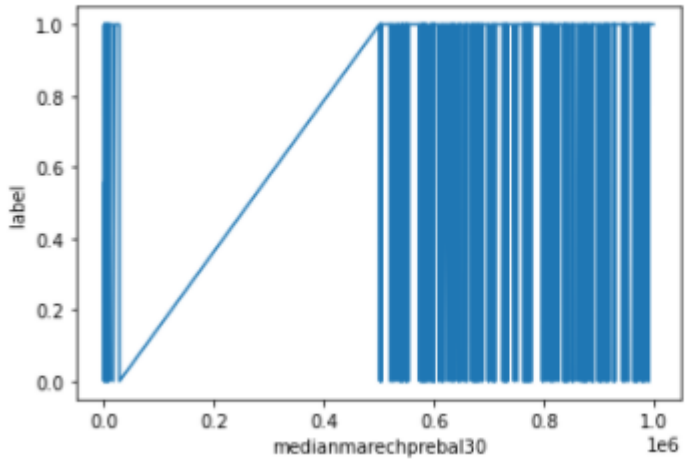
- Visualizations

Mention all the plots made along with their pictures and what were the inferences and observations obtained from those. Describe them in detail.
If different platforms were used, mention that as well.
Here in this project we have used different Seaborn and Matplot libraries for Lineplot, Barcharts, Heatmaps,

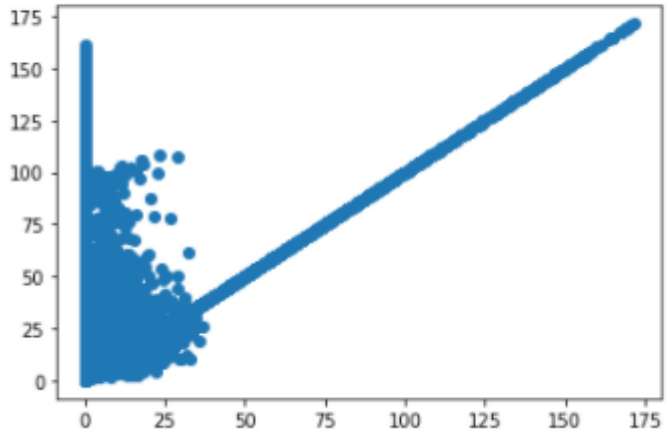
```
In [12]:
1 sns.lineplot(df['daily_decr30'],df['daily_decr90'])
2 plt.show()
```



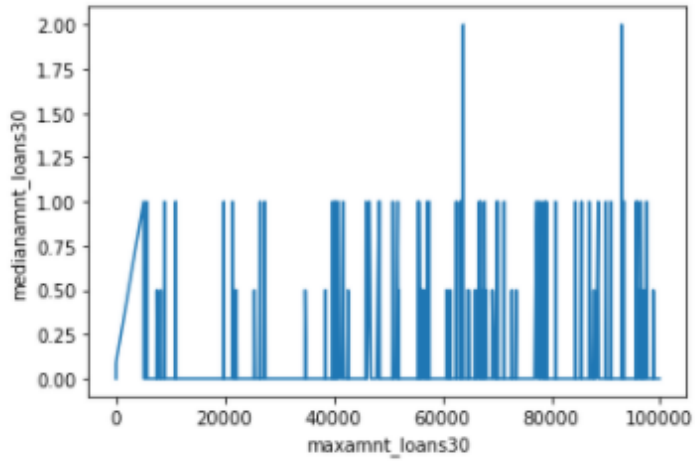
```
sns.lineplot(df['medianmarechprebal30'],df['label'])
plt.show()
```



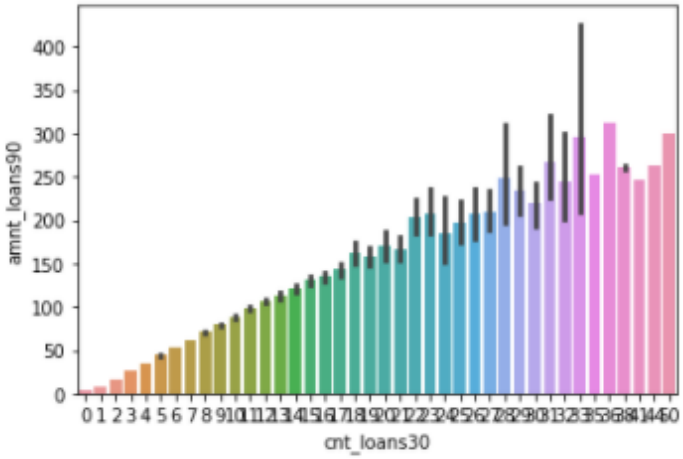

```
plt.scatter(df['payback30'],df['payback90'])
plt.show()
```



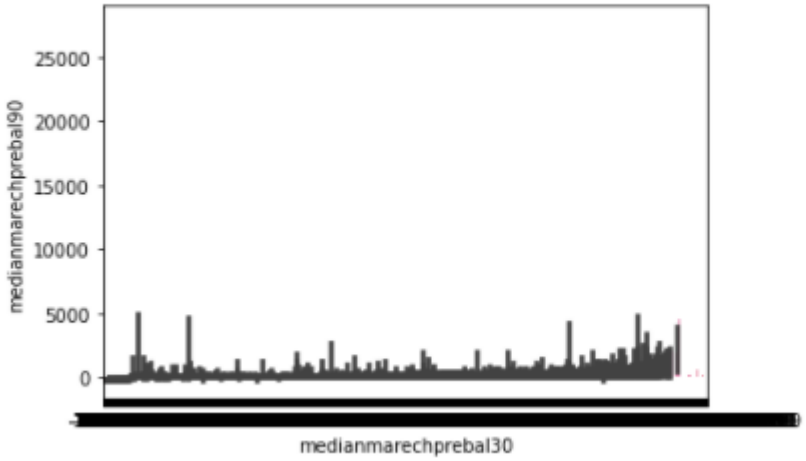
```
sns.lineplot(df['maxamnt_loans30'],df['medianamnt_loans30'])
plt.show()
```



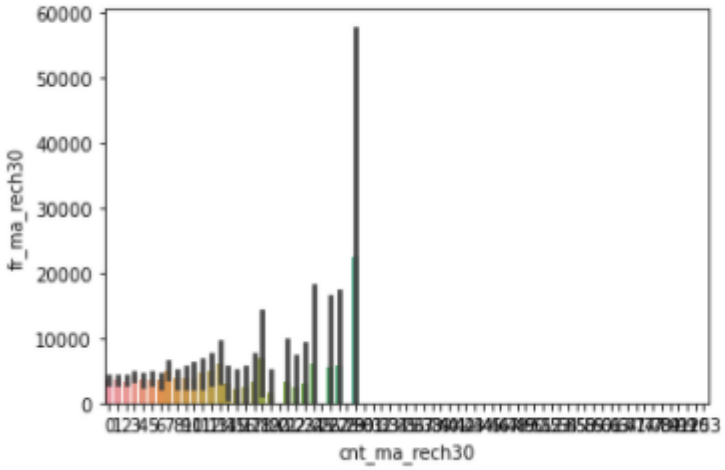
```
sns.barplot(df['cnt_loans30'],df['amnt_loans90'])
plt.show()
```




```
sns.barplot(df['medianmarechprebal30'],df['medianmarechprebal90'])
plt.show()
```




```
sns.barplot(df['cnt_ma_rech30'],df['fr_ma_rech30'])
plt.show()
```



- Interpretation of the Results

Give a summary of what results were interpreted from the visualizations, preprocessing and modelling.

After Appling Data Visualisation in this Data we can clearly see that most of the Attributes are directly proportional to each other.
For eg: 'Number of times data account got recharged in last 30 days and Number of times data Account got recharged in last 90 days'.
From the Data we have from the client and after applying our ML Algorithms we are getting a Decent result and on the basis of hat we can predict in more than 90% of the cases that if MSI need to give a Loan to that particular individual or not and from this project MSI are going to save their huge amount of money which was loaned and never returned back.

CONCLUSION

- Key Findings and Conclusions of the Study

Describe the key findings, inferences, observations from the whole problem.

In this Project we can see that there are some attributes which are not going to put any effect on our end result like mobile numbers of the Users, Telecom Circle, Date and Age on Cellular network.
After removing these Attributes we start working on the data and after applying visualisation we can see that "Daily Amount Spent from the Main Account in last 30 days is directly proportional to the Daily Amount spent in last 90 days"
When we are applying Bar-Graph on "Num of Loans taken in last 30 days and Total Amount of Loans taken in last 90 days" we can clearly see that when the Number of loans are increasing Total Amount of Loans are also Increasing.
After Applying Training and testing and machine learning algorithms we are getting Accuracy Score of around 87% and this Score goes upto 91% in Ensemble techniques like Random Forest and Gradient Boosting.

- Learning Outcomes of the Study in respect of Data Science

List down your learning's obtained about the power of visualization, data cleaning and various algorithms used. You can describe which algorithm works best in which situation and what challenges you faced while working on this project and how did you overcome that.

Using visualization allows users to better absorb the data and see new paths, Data Visualisation allows us to use our data Graphically, in the form of Bars, charts etc.

We use Different Algorithm in this project like,
KNN(KNeighborsClassifier), SVC(Support Vector Classifier), Decision Tree Classifier and some Boosting Algorithms as well like **Random Forest Classifier, Ada Boost Classifier and Gradient Boosting Classifier**.
Out of these algorithms we have use **RANDOM FOREST CLASSIFIER** works best and giving the best Accuracy Score out of all other algorithms we have used.
There is no such challenges i have faced in this project but when we are dealing with Outliers we are losing around 20% of the data and we cannot afford to lose this much of data as Data is Very expensive so I have changed the z value to 5 in place of 3(standard value) so that only 8-10% of data are deleted as Outliers.

- Limitations of this work and Scope for Future Work

What are the limitations of this solution provided, the future scope? What all steps/techniques can be followed to further extend this study and improve the results.

If we talk about the Limitation of this project/dataset, so, when we are applying the zscore value by putting z=3(which is a standard value round the globe) we are losing more than 20% of the data and we cannot afford to lose data as data is Expensive. According to me this is the Limitation which I think can help us to improve the results.