MICRO CREDIT LOAN PROJECT

Submitted by:

YASHNA SHAH

ACKNOWLEDGMENT

I would like to express my deepest gratitude to my SME (Subject Matter Expert) Sapna Verma as well as Flip Robo Technologies who gave me the opportunity to do this project on Micro Credit Loan , which also helped me in doing lots of research wherein I came to know about so many new things especially the data collection part.

Also, I have utilized a few external resources that helped me to complete the project. I ensured that I learn from the samples and modify things according to my project requirement. All the external resources that were used in creating this project are listed below:

1) https://github.com/

2) https://www.kaggle.com/

3) https://medium.com/

4) https://towardsdatascience.com/

5) https://www.analyticsvidhya.com/

INTRODUCTION

- ## Business Problem Framing

  There are many poor families living in remote areas with not much sources of income, Since the communication via Telephone is important and how it affects a person's life, this project provides their services to low income families and poor customers that can help them in the need of hour by providing Micro credit loan.

- ## Conceptual Background of the Domain Problem

  A Microfinance Institution (MFI) is an organization that offers financial services to low income populations. MFS becomes very useful when targeting especially the unbanked poor families living in remote areas with not much sources of income. The Microfinance services (MFS) provided by MFI are Group Loans, Agricultural Loans, Individual Business Loans and so on.

- ## Review of Literature

- Many microfinance institutions (MFI), experts and donors are supporting the idea of using mobile financial services (MFS) which they feel are more convenient and efficient, and cost saving, than the traditional high-touch model used since long for the purpose of delivering microfinance services. Though, the MFI industry is primarily focusing on low income families and are very useful in such areas, the implementation of MFS has been uneven with both significant challenges and successes.
- Today, microfinance is widely accepted as a poverty-reduction tool, representing $70 billion in outstanding loans and a global outreach of 200 million clients.

- ## Motivation for the Problem Undertaken

  Motivation behind this project is that mainly focuses on poor families living in remote areas with not much sources of income, Since the communication via Telephone is important and how it affects a person's life, this project providesthe services to low income families and poor customers that can help them in the need of hour by providing Micro credit loan to telephone number

Analytical Problem Framing

- ## Mathematical/ Analytical Modeling of the Problem

Mathematical Summary: Dimensions of Dataset: There are 36 columns and 209593 rows in this dataset Null Values: There are no null values in this dataset Skewness: Skewness is present in almost every columns

Statistical Summary: Standard deviation is very high in most of the columns. There is lot of difference between mean and 50th percentile, which means data is skewed There is lot of difference between 75th percentile and max, which means there are outliers

- ## Data Sources and their formats

Data Sources: The sample data is provided to us from our client database. 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. In this Project, Label '1' indicates that the loan has been paid i.e. Non- defaulter, while, Label '0' indicates that the loan has not been paid i.e. defaulter.

Data Formats: Phone number and date columns are object datatype which not use-full. All the other columns are either int or float DataTypes.

- ## Data Preprocessing Done

1. Importing the necessary dependencies and libraries.
2. Reading the CSV file and converted into data frame.
3. Checking the data dimensions for the original dataset.
4. There are no null values in our dataset.
5. We have an imbalanced dataset, which we solved using SMOTE.
6. Checking the summary of the dataset.
7. Checking unique values.
8. Checking all the categorical columns in the dataset and converted the same into numerical using LabelEncoder.
9. Scaled the dataset using StandardScaler method.
10. Checking for multi collinearity using VIF.
11. Performed PCA
12. Performed Feature Importance using ExtraTrees Regression.

- ## Data Inputs- Logic- Output Relationships

Correlation between the target variable and input is very low. 0.23 is highest correlation for 'cnt_ma_reach30' with target variable.

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

  Here, we have an imbalanced dataset.

  87.5% of data is class ‘1’

  12.5% of data is class ‘0

- Hardware and Software Requirements and Tools Used

- Hardware technology being used.
- RAM : 16 GB
- CPU : 11th Gen Intel(R) Core(TM) i5-1135G7 @ 2.40GHz 2.42 GHz
- GPU : intel Iris Graphics
- Software technology being used.
- Programming language : Python
- Distribution : Anaconda Navigator
- Browser based language shell : Jupyter Notebook
- Libraries/Packages specifically being used.
- Pandas, NumPy, matplotlib, seaborn, scikit-learn

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)

The dataset contains more than 2 lakh data with no null values present. The dataset is imbalanced. Label 1 has 87.5% of data whereas label 0 has around 12.5%. I found lot of outliers and skewness present in the dataset. The outliers were removed by using Z score method where the loss of data was around 21% which is too high. The skewness was reduced using square root method. There were certain columns which were least important with our target variable, hence those were dropped.

- Testing of Identified Approaches (Algorithms)

All the regression machine learning algorithms used are:

- Logistic Regression Model
- Decision Tree Model
- AdaBoost Model
- SVM Model
- Random Forest Classifier Model

- Run and Evaluate selected models

1. Logistic Regression

```
Training accuracy : 1.0
Testing accuracy : 1.0
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     44003
           1       1.00      1.00      1.00     44044

    accuracy                           1.00     88047
   macro avg       1.00      1.00      1.00     88047
weighted avg       1.00      1.00      1.00     88047

[[44003     0]
 [    0 44044]]
```

2. Decision Tree Classifier

```
# USING DECISION TREE
from sklearn.tree import DecisionTreeClassifier


model = DecisionTreeClassifier()
model.fit(x_train, y_train)
y_pred = model.predict(x_test)

# calculating the training and testing accuracies
print("Training accuracy :", model.score(x_train, y_train))
print("Testing accuracy :", model.score(x_test, y_test))
```

```
Training accuracy : 1.0
Testing accuracy : 1.0
```

```
# classification report
print(classification_report(y_test, y_pred))

# confusion matrix
print(confusion_matrix(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     44003
           1       1.00      1.00      1.00     44044

    accuracy                           1.00     88047
   macro avg       1.00      1.00      1.00     88047
weighted avg       1.00      1.00      1.00     88047

[[44003     0]
 [    0 44044]]
```

3. Random Forest Classifier

```python
# Random Forest Classifier
from sklearn.ensemble import RandomForestClassifier


model = RandomForestClassifier(n_estimators = 200)
model.fit(x_train, y_train)
y_pred = model.predict(x_test)

# calculating the training and testing accuracies
print("Training accuracy :", model.score(x_train, y_train))
print("Testing accuracy :", model.score(x_test, y_test))
```

```
Training accuracy : 1.0
Testing accuracy : 1.0
```

```python
# classification report
print(classification_report(y_test, y_pred))

# confusion matrix
print(confusion_matrix(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     44003
           1       1.00      1.00      1.00     44044

    accuracy                           1.00     88047
   macro avg       1.00      1.00      1.00     88047
weighted avg       1.00      1.00      1.00     88047

[[44003     0]
 [    0 44044]]
```

4. Support Vector Classifier

```
from sklearn.svm import SVC

# creating the model
model = SVC()

# feeding the training set into the model
model.fit(x_train, y_train)

# predicting the results for the test set
y_pred = model.predict(x_test)

# calculating the training and testing accuracies
print("Training accuracy :", model.score(x_train, y_train))
print("Testing accuracy :", model.score(x_test, y_test))
```

```
Training accuracy : 0.9999964133923928
Testing accuracy : 0.9999772848592229
```

```
# classification report
print(classification_report(y_test, y_pred))

# confusion matrix
print(confusion_matrix(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     44003
           1       1.00      1.00      1.00     44044

    accuracy                           1.00     88047
   macro avg       1.00      1.00      1.00     88047
weighted avg       1.00      1.00      1.00     88047

[[44001     2]
 [    0 44044]]
```

5. AdaBoost Classifier

```
# Ada Boost Classifier
from sklearn.ensemble import AdaBoostRegressor
model = AdaBoostRegressor()



# feeding the training set into the model
model.fit(x_train, y_train)

# predicting the results for the test set
y_pred = model.predict(x_test)

# calculating the training and testing accuracies
print("Training accuracy :", model.score(x_train, y_train))
print("Testing accuracy :", model.score(x_test, y_test))
```

```
Training accuracy : 1.0
Testing accuracy : 1.0
```

```
# classification report
print(classification_report(y_test, y_pred))

# confusion matrix
print(confusion_matrix(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     44003
           1       1.00      1.00      1.00     44044

    accuracy                           1.00     88047
   macro avg       1.00      1.00      1.00     88047
weighted avg       1.00      1.00      1.00     88047

[[44003     0]
 [    0 44044]]
```

- # Visualizations

The plots used to visualize the data are

1. Histplot
2. Countplot
3. Scatterplot

- # Observations

The countplot is used to represent Label 0 and 1. From the visualizations we can see that we have an imbalanced distribution of data in Label 1 and 0. The number of data in Label 1 and 0 can also be seen. The histplot represents the distribution of data, whereas , Scatterplot, on the other hand, represents the relationship between the features with the Target variable.

- # Interpretation of the Results

From the given dataset it is clear that most of the customers are willing and inclined to repay the loan, as 87.5% of the customers repaid the loan and only 12.5% of the customers are defaulter.

CONCLUSION

- Key Findings and Conclusions of the Study

Most of the customers have the intentions of repaying the loan. There are very few numbers of people , who are not inclined to repay.

- Learning Outcomes of the Study in respect of Data Science

The dataset was full of outliers , skewness and unbalanced data which was the biggest challenge to overcome. Data Cleaning was important for accurate predictions. I have used Logistic Regression, Decision Tree, Random Forest Classifier, Support Vector Classifier, AdaBoost Classifier. Among these algorithms I would go with Random Forest Classifier. As the dataset was imbalanced, the other algorithms may overfit and may give inaccurate predictions whereas Random Forest can control overfitting and give best predictions.