PROJECT REPORT
ON
MICRO-CREDIT DEFAULTER

Submitted by

MANISHA

Introduction

Business Problem Framing:

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.

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.

Conceptual Background of the Domain Problem:

We are working with one such client that is in Telecom Industry. They are a fixed wireless telecommunications network provider. They have launched various products and have developed its business and organization based on the budget operator model, offering better products at Lower Prices to all value conscious customers through a strategy of disruptive innovation that focuses on the subscriber.

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.

Review of Literature:

In the project we first import the excel file and then check the size of the data. Check the null values and data type, if there is object data type so convert into float type because analysis worked on numerical data. Then visualise the categorical data and continuous data. Describe the data where we get all statistical information, then find correlation. Check the outliers and skewness on the continuous data type. Fit the classification model and select the best model in all of them. Then tuning is applied to get optimize score.

Motivation for the Problem Undertaken:

The objective behind the project is 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.

Analytical Problem Framing

Mathematical/ Analytical Modeling of the Problem:

We use Statistical techniques and analytics modeling in our projects, such as:

- describe() : use to calculate the statistical values that are mean, standard deviation, quantile deviation, minimum and maximum values.
- corr(): use to calculate the relation between feature variable with the target variable.
- Multicolinearity between independent and dependent variable.
- Check the outliers by plotting boxplot
- skew(): use to check whether the skewness is present in the continuous data or not.

Data Source and their formats:

The data set of the Micro-Credit Defaulter as show in the fig:

```
data=pd.read_excel('Data file.xlsx')
data.head()
```

	Unnamed: 0	label	msisdn	aon	daily_decr30	daily_decr90	rental30	rental90	last_rech_date_ma	last_rech_date_da	...	maxamnt_loans30	medianam
0	1	0	21408170789	272.0	3055.050000	3065.150000	220.13	260.13	2.0	0.0	...	6.0	
1	2	1	76462170374	712.0	12122.000000	12124.750000	3691.26	3691.26	20.0	0.0	...	12.0	
2	3	1	17943170372	535.0	1398.000000	1398.000000	900.13	900.13	3.0	0.0	...	6.0	
3	4	1	55773170781	241.0	21.228000	21.228000	159.42	159.42	41.0	0.0	...	6.0	
4	5	1	03813182730	947.0	150.619333	150.619333	1098.90	1098.90	4.0	0.0	...	6.0	

5 rows × 37 columns

There are 209593 rows and 37 columns. ‘label’ column is our target variable and others are feature variable. There are 35 columns of numerical data and 2 columns of object type.

Data Pre-processing

There are no null values in the dataset. Apply feature engineering technique datetime to split the date into day, month and year. Convert object data into numerical data.

Data inputs-Logic-Output Relationships

There are 36 columns in which 35 are independent variable and one dependent variable called ‘label’. On checking the correlation we see that 60% data is highly correlated with the target variable. Left of the 40% data is low correlated and in which two columns have zero correlation.

Hardware and Software Requirements and Tools used

Hardware:

- Memory 16GB minimum
- Hard Drive SSD is preferred 500GB
- Processor intel i5 minimum
- Operating system Windows 10

Software:

- Jupyter notebook (Python)

Libraries:

pandas (used to create the data and read the data)

numpy (used with the mathematical function)

seaborn (used to create a different types of graphs)

matplotlib (used to plot the graph)

zscore (used to remove outliers)

GridSearchCV (give the best parameters for tuning the model)

accuracy_score (used to calculate accuracy score for train and test)

classification_report (to display precision, f1 score)

confusion_matrix (form the

roc_curve (to plot a curve for predicted values)

roc_auc_score (calculate area under the curve)

Model/s Development and Evaluation

Identification of possible problem:

We approach to both statistical and analytical problem

- ❖ Plot a bar graph for nominal data and distribution graph for continuous data
- ❖ describe () use to calculate mean, standard deviation, minimum, maximum and quantile deviation
- ❖ corr() used to calculate the correlation of input variable with the output variable.
- ❖ skew() used to calculate the skewness of the data
- ❖ zscore used to remove the outliers

Testing of Identified Approaches:

Machine Learning Models

Here we work on the classification problem so the models are:

- Logistic Regression
- K Neighbors Classification
- Random Forest Classification
- Decision Tree Classification

Run and Evaluate selected models:

- Logistic Regression

```
lr=LogisticRegression()
x_train,x_test,y_train,y_test = train_test_split(x_scaled,y, test_size=0.20,random_state=14)
```

```
lr.fit(x_train,y_train)
y_pred = lr.predict(x_test)
accuracy = accuracy_score(y_test,y_pred)*100
print("accuracy score:",accuracy)
```

```
accuracy score: 75.70833648822594
```

```
cm= confusion_matrix(y_test,y_pred)
print(cm)
```

```
[[22105  7039]
 [ 9002 27889]]
```

```
clr=classification_report(y_test,y_pred)
print(clr)
```

```
              precision    recall  f1-score   support

           0       0.71      0.76      0.73     29144
           1       0.80      0.76      0.78     36891

    accuracy                           0.76     66035
   macro avg       0.75      0.76      0.76     66035
weighted avg       0.76      0.76      0.76     66035
```

The accuracy score of the logistic regression is 76. The precision is 71, recall is 76 and f1-score is 73 for defaulter whereas the precision is 80, recall is 76 and the f1-score is 78.

- K Neighbors Classification

```
knn= KNeighborsClassifier()
x_train,x_test,y_train,y_test = train_test_split(x_scaled, y, test_size=0.20,random_state=97)
```

```
knn.fit(x_train,y_train)
y_pred = knn.predict(x_test)
accuracy = accuracy_score(y_test,y_pred)*100
print("accuracy score:",accuracy)
```

```
accuracy score: 87.45816612402514
```

```
cm= confusion_matrix(y_test,y_pred)
print(cm)
```

```
[[28660   778]
 [ 7504 29093]]
```

```
clr=classification_report(y_test,y_pred)
print(clr)
```

```
              precision    recall  f1-score   support

           0       0.79      0.97      0.87     29438
           1       0.97      0.79      0.88     36597

    accuracy                           0.87     66035
   macro avg       0.88      0.88      0.87     66035
weighted avg       0.89      0.87      0.87     66035
```

The accuracy score of the K Neighbors classification is 87. The precision is 79, recall is 97 and f1-score is 87 for defaulter whereas the precision is 97, recall is 79 and the f1-score is 88.

- Random Forest Classification

```
rfc=RandomForestClassifier()
x_train,x_test,y_train,y_test = train_test_split(x_scaled,y, test_size=0.20,random_state=57)
```

```
rfc.fit(x_train,y_train)
y_pred = rfc.predict(x_test)
accuracy = accuracy_score(y_test,y_pred)*100
print("accuracy score:",accuracy)
```

```
accuracy score: 95.01173620049973
```

```
cm= confusion_matrix(y_test,y_pred)
print(cm)
```

```
[[27638  1749]
 [ 1545 35103]]
```

```
clr=classification_report(y_test,y_pred)
print(clr)
```

```
              precision    recall  f1-score   support

           0       0.95      0.94      0.94     29387
           1       0.95      0.96      0.96     36648

    accuracy                           0.95     66035
   macro avg       0.95      0.95      0.95     66035
weighted avg       0.95      0.95      0.95     66035
```

The accuracy score of the Random Forest classification is 95. The precision is 95, recall is 94 and f1-score is 94 for defaulter whereas the precision is 95, recall is 96 and the f1-score is 96.

- Decision Tree Classification

```
clf=DecisionTreeClassifier()
x_train,x_test,y_train,y_test = train_test_split(x_scaled,y, test_size=0.20,random_state=30)
```

```
clf.fit(x_train,y_train)
y_pred = clf.predict(x_test)
accuracy = accuracy_score(y_test,y_pred)*100
print("accuracy score:",accuracy)
```

```
accuracy score: 90.33088513666995
```

```
cm= confusion_matrix(y_test,y_pred)
print(cm)
```

```
[[26261  2989]
 [ 3396 33389]]
```

```
clr=classification_report(y_test,y_pred)
print(clr)
```

```
              precision    recall  f1-score   support

           0       0.89      0.90      0.89     29250
           1       0.92      0.91      0.91     36785

    accuracy                           0.90     66035
   macro avg       0.90      0.90      0.90     66035
weighted avg       0.90      0.90      0.90     66035
```

The accuracy score of the Decision Tree classification is 90. The precision is 89, recall is 90 and f1-score is 89 for defaulter whereas the precision is 92, recall is 91 and the f1-score is 91.

The Random forest classification gives better accuracy score and the total of True Negative and False Negative in the confusion matrix is also less in the same model. So, we tune the Random forest to optimize the score and we get an accuracy score is 95. There is no increment in the score.

Visualisation:

On visualising the continuous data we see the data is right skewed and the target variable is imbalance. So, this we can balance by the SMOTE technique. pcircle and pyear have unique value in the columns by this it will not effect on our prediction.

Interpretation of the Results:

On our analysis basis we conclude that the 95% of the customer will be paying back the loaned amount within 5 days of insurance of loan and 5% of customers are defaulter.

Conclusion

Key Findings and Conclusions of the Study:

On study the problem we get to know the MFI provides loan to the low income person. They mortgage something while giving a loan to the person. So, we want to analyse and predict the customers who paying back the loan amount to MFI.

Learning Outcomes of the Study in respect of Data Science:

Here we use feature engineering technique on date and drop Unnamed: 0 column from dataset. Visualise the continuous data and here we see the data is skewed and have some columns with un relevant values. The target variable 'label' is imbalanced so we balanced it by SMOTE. In analysis we do describe the statistical values, correlation, outliers and skewness. Check multicolinearity and drop one of the higher correlated value. Fit some classification models and find the better one i.e. Random Forest Model and then tune the parameters. Calculate accuracy score, confusion metrics, classification report, roc curve and roc_auc score.

Limitations of this work and Scope for Future Work:

Limitations:

- Higher interest rate in comparison of banks.
- Lack of enough awareness of financial services in the economy.

Scope:

- To empower people and make them self reliant.