Micro Credit Defaulter

Submitted by:

Rijul Kumar

ACKNOWLEDGMENT

Reference that i have used are:

- Data Trained Education online video
- Materials provided by Flip Robo
- Geeks for Geeks
- Stackoverflow

INTRODUCTION

- ## Business Problem Framing

    1. Microfinance Institution (MFI) is an organization that offers financial services to low income populations.
    2. We are working with one such client that is in Telecom Industry. They are a fixed wireless telecommunications network provider. They are collaborating with an MFI to provide micro-credit on mobile balances to be paid back in 5 days.
    3. We have to build a model which can be used 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.

- ## Conceptual Background of the Domain Problem

    The probability of repaying loan within 5 days of issuing the loan can be predicted with the help of some factors. Such as:

    - Average main account balance over last 90 days
    - Average payback time in days over last 90 days
    - Total amount of loans taken by user in last 90 days
    - Daily amount spent from main account, averaged over last 90 days (in Indonesian Rupiah)
    - And so on…

- ## Motivation for the Problem Undertaken

  - We have to build a model which can be used 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.

  - Then with help of this model telecom company will be able to focus on providing their services and products to low income families and poor customers efficiently.

Analytical Problem Framing

- Mathematical/ Analytical Modeling of the Problem
  - First of all i imported data from train and test csv files to DataFrames using pandas.
  - After that i used .describe() to know the statistical information (such as max, min value,etc) of train dataframe.
  - Then i used .shape to know shape of train and test dataframe.
  - Afterwards i used .dtypes to know data type of each column of train and test dataframe.
  - Next i moved on to data preprocessing part.

- Data Sources and their formats
  - Data file.csv provided by FlipRobo
  - Micro Credit Loan Use Case.docx provided by FlipRobo
  - Data_Description.xlsx provided by FlipRobo

- ## Data Preprocessing Done
    - First of all for data preprocessing i checked whether there is a NULL value or not in dataframe using heatmap as well as .isnull()
    - After that i used count plots from seaborn library to plot all categorical columns for visualisation.
    - Next i used Density plots from seaborn library to plot all continuous columns for visualisation.
    - Then i did encoded the dataframe using Ordinal Encoder.
    - After that i checked for correlations using heatmaps, correlation matrix and BAR plot.
    - Finally i confirmed high correlations using VIF and dropped highly correlated columns.
    - After that i removed skewness from continuous data using yeo-johnson algorithm.
    - Afterwards i checked for outliers.
    - Since removal of outliers was giving data loss of 18.56%, i did not remove outliers but opted to take tree based models during later part as tree algorithms are not affected by them.

- ## Data Inputs- Logic- Output Relationships
    - Data Input :

      These are basically the factors (such as Average main account balance over last 90 days, etc) which helps us in predicting whether a person will

repay the loan or not within 5 days of issuing the loan.

- Data Output :

  Our Target variable is label which is the flag indicating whether the user paid back the credit amount within 5 days of issuing the loan {1:success, 0:failure}.

- ## Hardware and Software Requirements and Tools Used

  - Hardware used:

    Laptop with intel core i5 7th gen

  - Software used:

    i. Jupyter notebook

    ii. Required python libraries such as numpy, pandas, seaborn, matplotlib, etc

    iii. Required libraries for model such as sklearn, etc

Model/s Development and Evaluation

- Identification of possible problem-solving approaches (methods)
    - After preprocessing data (checking NULL, encoding, removing high correlations, removing skewness and outliers) i separated columns into features and target.
    - As this is a regression problem so we tried 4 models - **LogisticRegression, DecisionTreeClassifier, RandomForestClassifier and Gaussian Naive Bayes.**
    - We checked for each model analytically (classification report and confusion matrix) as well as graphically (AUC-ROC curves).
    - Finally i used Ensemble Technique.

- Testing of Identified Approaches (Algorithms)

  As this is a regression problem so we tried following 4 models -
    - LogisticRegression
    - DecisionTreeClassifier
    - RandomForestClassifier
    - Gaussian Naive Bayes

- Run and Evaluate selected models
  I defined a function model and then tried 4 different models using it

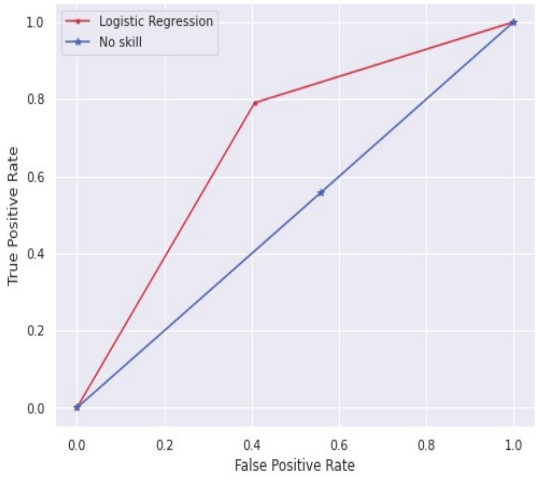
```
def model_selection(algorithm_instance,features_train,target_train,features_test,target_test):
    algorithm_instance.fit(features_train,target_train)
    model_1_pred_train = algorithm_instance.predict(features_train)
    model_1_pred_test = algorithm_instance.predict(features_test)
    print("Accuracy for the training model : ",accuracy_score(target_train,model_1_pred_train))
    print("Accuracy for the testing model : ",accuracy_score(target_test,model_1_pred_test))
    print("Confusion matrix for model : \n",confusion_matrix(target_test,model_1_pred_test))
    print("Classification Report for train data : \n",classification_report(target_train,model_1_pred_train))
    print("Classification Report for test data : \n",classification_report(target_test,model_1_pred_test))

    Train_accuracy = accuracy_score(target_train,model_1_pred_train)
    Test_accuracy = accuracy_score(target_test,model_1_pred_test)

    for j in range(2,4):
        cv_score = cross_val_score(algorithm_instance,feature_over,target_over,cv=j)
        cv_mean = cv_score.mean()
        print("At cross fold " + str(j) + " the cv score is " + str(cv_mean) + " and accuracy score for training is " + str(Train_accurac
        print("\n")

    #Plotting auc_roc curve
    plt.figure(figsize=(8,6))

    # calculate roc curves
    lr_fpr, lr_tpr, _ = roc_curve(target_test, model_1_pred_test)
    lr_fpr1, lr_tpr1, _ = roc_curve(target_train, pred_train)

    plt.plot(lr_fpr, lr_tpr, marker='.', label=algorithm_instance, color='r')
    plt.plot(lr_fpr1, lr_tpr1, marker='*', label='No skill', color='b')

    plt.xlabel('False Positive Rate')
    plt.ylabel('True Positive Rate')

    plt.legend()
    plt.show()
```

Result that i got for each model :

- LogisticRegression :
  At random state 8 the training accuracy is :
  0.6921008964560852
  At random state 8 the testing accuracy is :
  0.6910443896256117

  At cross fold 2 the cv score is 0.5317612617278431 and accuracy score for training is 0.49965756808602707 and accuracy score for testing is 0.6910443896256117

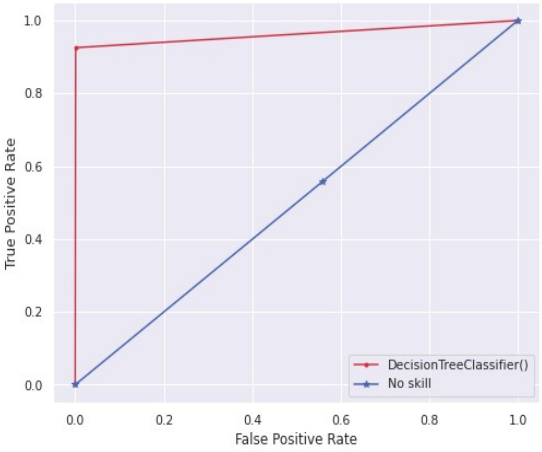
```
Confusion matrix for model :
 [[21902 15042]
 [ 7627 28802]]
Classification Report for train data :
              precision    recall  f1-score   support

           0       0.50      0.44      0.47    146487
           1       0.50      0.56      0.53    147002

    accuracy                           0.50    293489
   macro avg       0.50      0.50      0.50    293489
weighted avg       0.50      0.50      0.50    293489

Classification Report for test data :
              precision    recall  f1-score   support

           0       0.74      0.59      0.66     36944
           1       0.66      0.79      0.72     36429

    accuracy                           0.69     73373
   macro avg       0.70      0.69      0.69     73373
weighted avg       0.70      0.69      0.69     73373
```



- <u>DecisionTreeClassifier</u> :
  Accuracy for the training model : 1.0
  Accuracy for the testing model : 0.962452128167037

  At cross fold 2 the cv score is 0.4982527489900835 and accuracy score for training is 1.0 and accuracy score for testing is 0.962452128167037

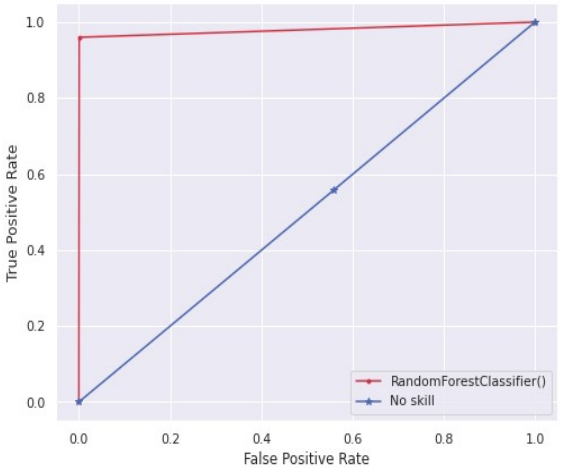
```
Confusion matrix for model :
 [[36905    39]
 [ 2716 33713]]
Classification Report for train data :
              precision    recall  f1-score   support

           0       1.00      1.00      1.00    146487
           1       1.00      1.00      1.00    147002

    accuracy                           1.00    293489
   macro avg       1.00      1.00      1.00    293489
weighted avg       1.00      1.00      1.00    293489

Classification Report for test data :
              precision    recall  f1-score   support

           0       0.93      1.00      0.96     36944
           1       1.00      0.93      0.96     36429

    accuracy                           0.96     73373
   macro avg       0.97      0.96      0.96     73373
weighted avg       0.96      0.96      0.96     73373
```



- RandomForestClassifier :
  Accuracy for the training model :
  0.999965927172739
  Accuracy for the testing model :
  0.979665544546359

  At cross fold 2 the cv score is 0.5288473594975768 and accuracy score for training is 0.999965927172739 and accuracy score for testing is 0.979665544546359

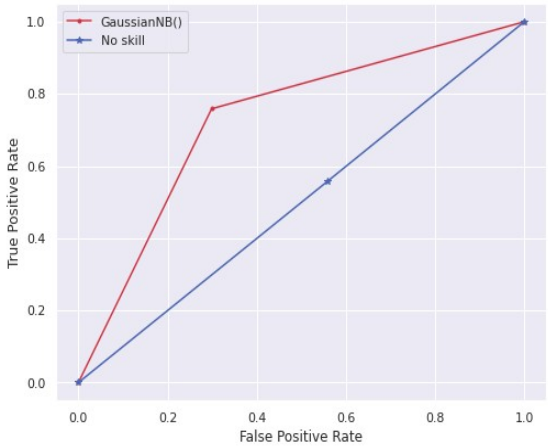
```
Confusion matrix for model :
 [[36911    33]
 [ 1459 34970]]
Classification Report for train data :
              precision    recall  f1-score   support

           0       1.00      1.00      1.00    146487
           1       1.00      1.00      1.00    147002

    accuracy                           1.00    293489
   macro avg       1.00      1.00      1.00    293489
weighted avg       1.00      1.00      1.00    293489

Classification Report for test data :
              precision    recall  f1-score   support

           0       0.96      1.00      0.98     36944
           1       1.00      0.96      0.98     36429

    accuracy                           0.98     73373
   macro avg       0.98      0.98      0.98     73373
weighted avg       0.98      0.98      0.98     73373
```



- GaussianNB :
  Accuracy for the training model :
  0.7295707845949934
  Accuracy for the testing model :
  0.7297643547353931

  At cross fold 2 the cv score is 0.5900174997683052 and accuracy score for training is 0.7295707845949934 and accuracy score for testing is 0.7297643547353931

```
Confusion matrix for model :
 [[25904 11040]
 [ 8788 27641]]
Classification Report for train data :
              precision    recall  f1-score   support

           0       0.74      0.70      0.72    146487
           1       0.72      0.76      0.74    147002

    accuracy                           0.73    293489
   macro avg       0.73      0.73      0.73    293489
weighted avg       0.73      0.73      0.73    293489

Classification Report for test data :
              precision    recall  f1-score   support

           0       0.75      0.70      0.72     36944
           1       0.71      0.76      0.74     36429

    accuracy                           0.73     73373
   macro avg       0.73      0.73      0.73     73373
weighted avg       0.73      0.73      0.73     73373
```



Finally i concluded that although RandomForestRegressor() gives best accuracy.

But DecisionTreeClassifier() gives approximately same accuracy as well as much higher model training speed.

Hence i took **DecisionTreeClassifier()** as main model

- ## Interpretation of the Results

  So the results which i got were:

  - Although RandomForestRegressor() is best model in terms of accuracy but DecisionTreeClassifier() is much better if accuracy as well as training speed is considered.
  - Above point can be seen through best fit curve as well as accuracy_score.
  - We got our final accuracy as 77.1% for target variable after hypertuning.

CONCLUSION

- Key Findings and Conclusions of the Study
    - From this study i learnt that sometimes loan repayment can be deduced on the basis of some factors which involves customers past records.
    - Many columns (such as daily_decr30 and daily_decr90) which contains record of last 30 days and last 90 days were highly correlated i.e. behaviour of customers remains almost same over a period of time.

- Learning Outcomes of the Study in respect of Data Science

  Some problems faced and their solution (using visualisation and algorithm) used were:

    - Removing outliers was leading to very high data loss (18.56%). So in place of removing it i used tree algorithms based model which is not sensitive to outliers.
    - Having large data and lack of high GPU resulted in some models giving each fold of output cross validation in a very long time. So to solve this problem i decreased number of folds of cross validations.

- Limitations of this work and Scope for Future Work

  Some limitations are :

  - There are many other factors (such as competition with other telecom company) which are not in the data which may play major role in predicting loan repayment within 5 days.
  - Unrelated factors such as family emergencies, etc also plays minor role in affecting our target variable.

  Scope for future work :

  - This can be made further accurate by taking more and more factors into account