Summer Mentorship Program IITK 2021

By:Manas Mishra
Mentor:Shubhadeep Das,EY

Mortgage Loan Default Prediction

Objective: Build a model which gives a certain acceptable level of accuracy using optimum number of features so that it can be beneficial from a business perspective also.

Data Description

- The Dataset is mortgage loan level data collected from the portfolios underlying U.S. residential mortgage-backed securities (RMBS) securitization portfolios and provided by International Financial Research.

- There are 49999 instances in the dataset where each instance contains the status of loan and 24 features some of which are loan to value ratio,balance amount,house price index and more.

- The features are from both observation time and origination time of the loan.

Outline

- Dropping columns which are irrelevant
- Splitting of Data in Two parts one part for model building and hyperparameter tuning,Second part for evaluation of the model.
- Fitting simple logistic regression model to know the areas where we can improve upon.
- Feature engineering to help the model learn the pattern more significantly.
- Weight of evidence approach for Feature selection because we want the model to be as simple as possible with a certain level of accuracy.
- Gradient Boosting to enhance the performance of the model.

Studying about the data

Before starting to build the model we observed the data and studied about it variables.It was found out that two features loss given default and recovery amount were observed when the default happened and hence they did not help in predicting the probability of default hence we dropped them.

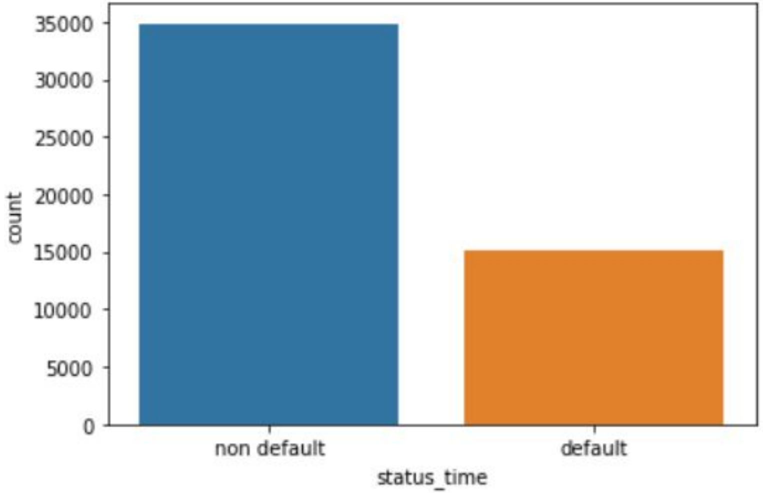
Missing values can also hamper our models hence they should also be dealt with smartly.In one of the features i.e. LTV it was found that some of the features were missing hence it was imputed them by its column mean.

Some highly correlated variables were also dropped.Data imbalance is also one of the things which we came across in this dataset we talk about it in the next slide.

Data Imbalance

- As we can clearly see from the figure on right hand side that the dataset has more number of non default cases than the default
- In this project we tried to deal with this problem so that the model does perform well for both the cases

-
  ```
  non default cases   34845

  default cases      15154
  ```



Splitting of Dataset

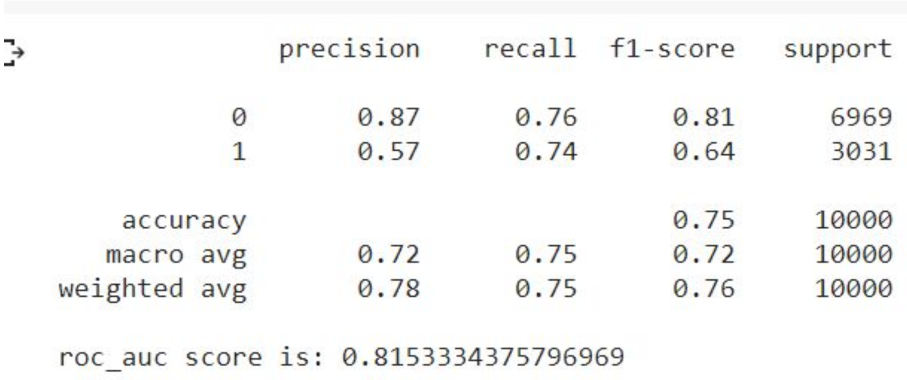
Data was partitioned in two parts:

Training and Testing part:39999 observation

Out of sample part:10000 observations

Vanilla Logistic model

- Firstly we train a simple logistic model using all the original features.
- We see that the recall score which is an important metric here is quite low,we will want it to increase it substantially
- The roc_auc score here is 81.4 which is not bad but the features are quite high.We intend to build a model which gives a higher roc auc score that too with lesser number of variable so that it is beneficial from a business perspective.



```
              precision    recall  f1-score   support

           0       0.87      0.76      0.81      6969
           1       0.57      0.74      0.64      3031

    accuracy                           0.75     10000
   macro avg       0.72      0.75      0.72     10000
weighted avg       0.78      0.75      0.76     10000

roc_auc score is: 0.8153334375796969
```

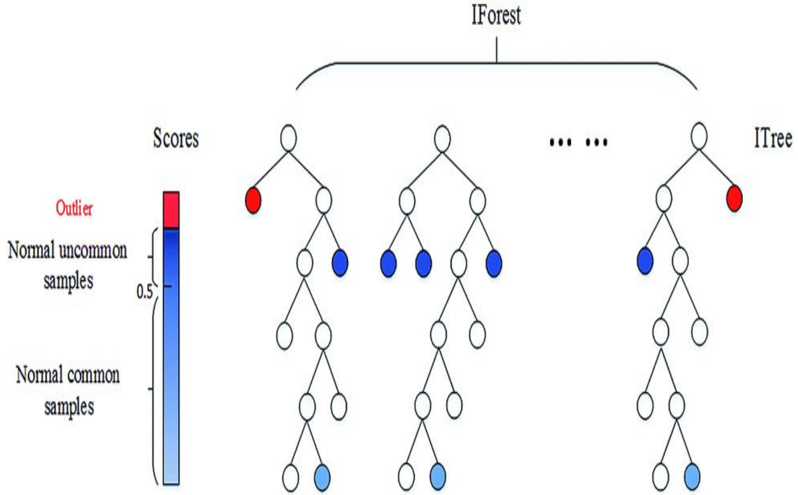
Feature Engineering

If we want our model to make better predictions then our machine should be able to learn more patterns which can be done by providing more information to the machine which is what feature creation helps in.Since our dataset has lesser number of default cases so we can treat the default cases as anomalies and make a new binary column assigning 0 and 1 to default and non default values respectively.This will help the machine in learning more patterns in the data.

Isolation forest

For feature creation described in the previous slide we use the Isolation forest algorithm.For example if we segregate cricket players based on number of runs scored then Sachin Tendulkar will stand alone and will be a outlier in a sense.This is what isolation forest does the outliers which are rare will be segregated first by the nature of decision tree works and hence they will have a shortest path from the root nodes which can then be identified as anomalies.

For detailed description:https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.IsolationForest.html



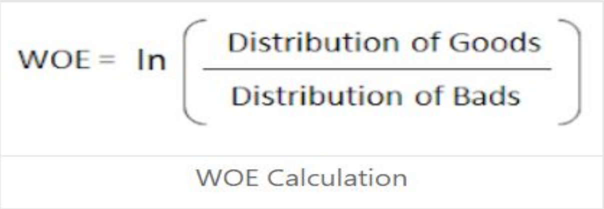
Feature Selection using weight of evidence

1)If there are too many variables then our model's performance might get affected due to noisy variables which due not contribute much hence we want to select only those feature which are best predictors for our case.

2)For this approach we have used weight of evidence and information value.One added advantage of using this approach was that we did not have to check for outliers in the data because of the way weight of evidence works.At the end we select six most important features from the given 24 in the original dataset.

Describing weight of evidence

The weight of evidence tells the predictive power of an independent variable in relation to the dependent variable. Since it evolved from credit scoring world, it is generally described as a measure of the separation of good and bad customers. **"Bad Customers"** refers to the customers who defaulted on a loan. and **"Good Customers"** refers to the customers who paid back loan.



$$\text{WOE} = \text{In}\left(\frac{\text{Distribution of Goods}}{\text{Distribution of Bads}}\right)$$

WOE Calculation

> ***Distribution of Goods -*** *% of Good Customers in a particular group*
> ***Distribution of Bads -*** *% of Bad Customers in a particular group*
> ***In -*** *Natural Log*

Selected Features

At the end we select six most important features from the given 24 features in the original dataset using the weight of evidence approach.Features having IV value greater than 0.2 are selected.

1)Loan to value ratio

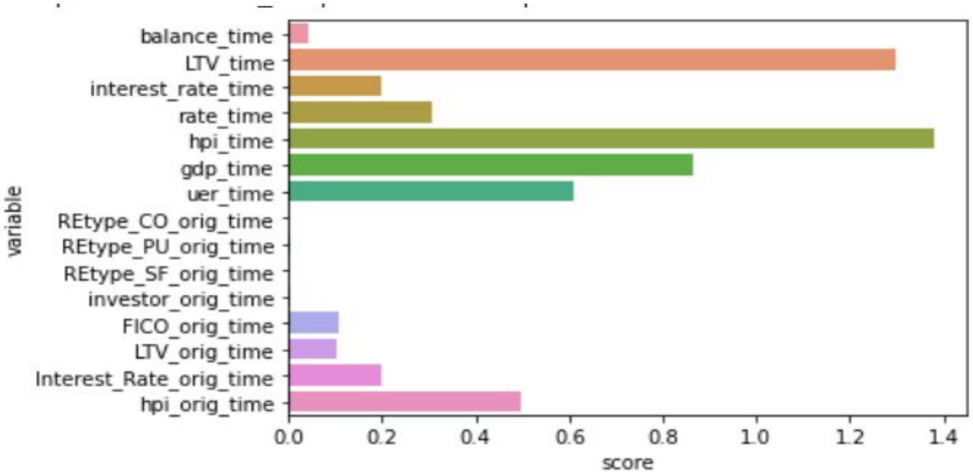
2)risk-free rate

3)interest rate

4)House price Index

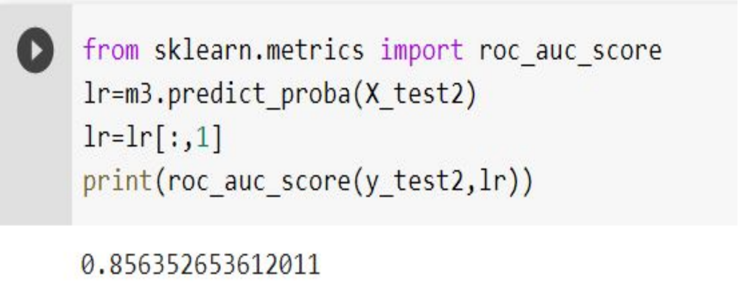
5)GDP

6)House price index at origination time



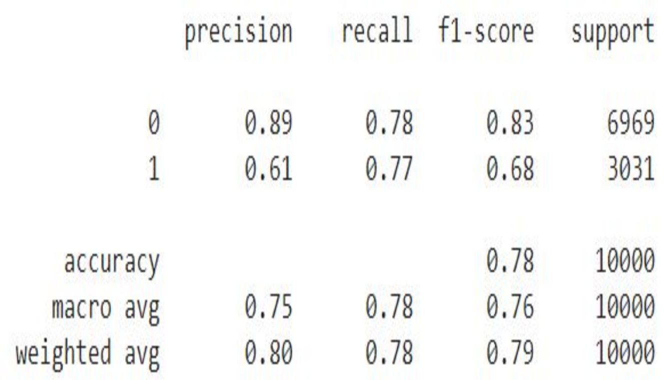
Logistic Model

Using the previously shown 6 features we build two models.If the requirements are of an explainable model and we want a simple deployment then Logistic regression with the selected features are used.It increases the roc-auc score by 4% and the recall score by 3% that too with lesser number of variables.



```
from sklearn.metrics import roc_auc_score
lr=m3.predict_proba(X_test2)
lr=lr[:,1]
print(roc_auc_score(y_test2,lr))
```

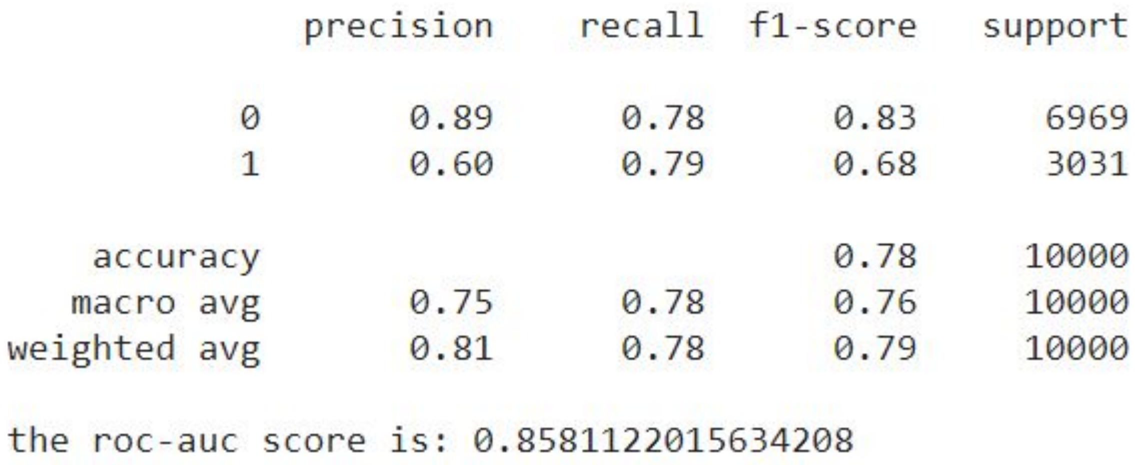
```
0.856352653612011
```



```
              precision    recall  f1-score   support

           0       0.89      0.78      0.83      6969
           1       0.61      0.77      0.68      3031

    accuracy                           0.78     10000
   macro avg       0.75      0.78      0.76     10000
weighted avg       0.80      0.78      0.79     10000
```

Machine Learning Model

Since the bias is an issue here instead of using one model we can use a group of model just like in medical diagnosis we not only consult more than one doctor in order to have a better prediction.In machine learning this approach is called ensemble learning.Here we have used Light Gradient Boosting(Light GBM) and from the figure below we can see that the performance has improved significantly.Our roc-auc score and recall score both have increased by 4% and 5% respectively that too with only 25% of the number of original features.

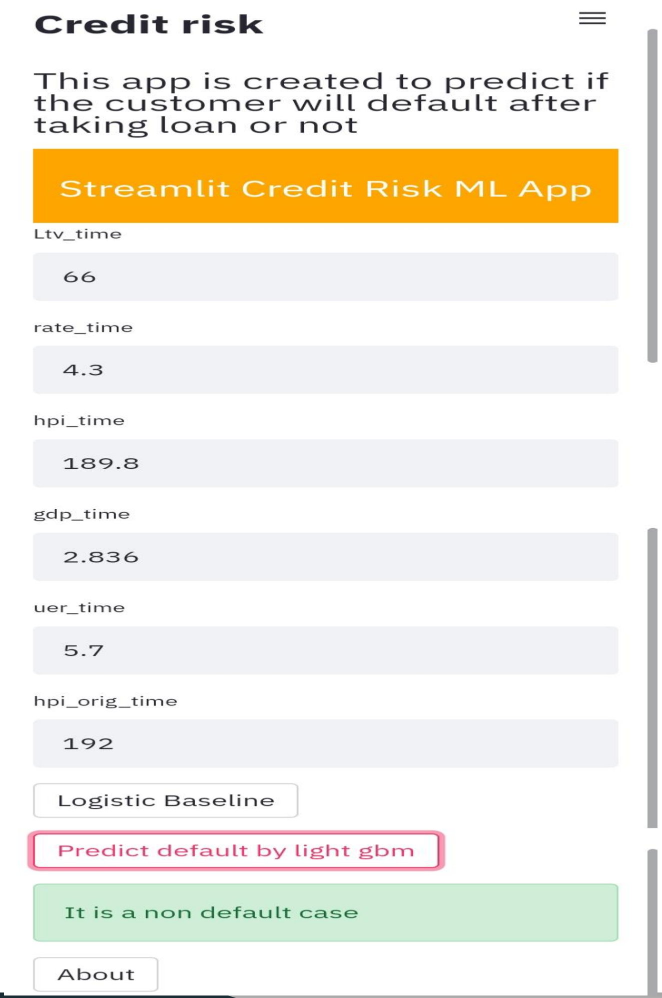


```
              precision    recall  f1-score   support

           0       0.89      0.78      0.83      6969
           1       0.60      0.79      0.68      3031

    accuracy                           0.78     10000
   macro avg       0.75      0.78      0.76     10000
weighted avg       0.81      0.78      0.79     10000

the roc-auc score is: 0.8581122015634208
```

App

After building the model an interactive app was created where the lender could just enter the details and find out if the customer will default or not.

Applink:https://credit-risk-new.herokuapp.com/



Credit risk

This app is created to predict if the customer will default after taking loan or not

Streamlit Credit Risk ML App

Ltv_time

66

rate_time

4.3

hpi_time

189.8

gdp_time

2.836

uer_time

5.7

hpi_orig_time

192

It is a non default case