Capstone Project

Ítalo Silva

Data Scientist Nanodegree

March 21th, 2022

Definition

Project Overview

Credit is trust. A trust that the lender will reimburse the borrower in the future, most of the time charging fees. In fact, the word "credit" derives from the Latin word "credere" which means believe or trust. With these statements, it becomes clear that the act of providing credit involves a question: is the borrower (a person or a company) trustable to repay the funds that the lender will provide?

In the past, the decision to provide credit are made individually analyzing the 3 C's of credit: "Character" (the history of the borrower in fulfilling its obligations), "Capital" (the assets of the borrower that could be used as guarantee or not) and "Capacity" (whether the income of the borrower is enough to repay). However, nowadays most of these decisions are automated using models that provides credit scores.

In this project, I created a model that provides the probability of default (PD) of a dataset from Small Business Administration (SBA). SBA is an institution from US funded in 1953 with objectives of promoting and assisting small enterprises in US credit market. SBA acts like insurance provider by taking some of the risk guaranteeing a portion of the loan. (Li, Mickel e Taylor 2018)

The article named ""Should This Loan be Approved or Denied?": A Large Dataset with Class Assignment Guidelines" was the base for this project and associated datasets.

Problem Statement

The goal of this project is to create a model which output is the PD of small business companies to decide if we should approve or not a loan. The tasks involved are the following:

1. Business and data understanding (reading the article and metadata)
2. Exploratory Data Analysis
3. Data Preprocessing and Feature Selection
4. Create a Machine Learning Pipeline
5. Hyper parameter Tuning
6. Evaluate the model
7. Test an alternative algorithm (and repeat steps 4-6)
8. Compare the results

Metrics

In general, credit lenders are more interested to know the PD from an operation than a simple binary classification (will default or not). It occurs because with large homogeneous groups with similar PD we can accept that some contracts will not be paid in full and to support that lost we charge a greater fee from the entire group. Is that fact that differentiate why some customers are charged more than others. With that in mind, a great approach to compare and measure PD models is using Receiving Operating Curve (ROC curve) and calculating the Area Under the Curve (AUC).

ROC curve consist in a chart with the False Positive Rate (FPR) in the x-axis and the True Positive Rate (TPR) in the y-axis. Below we can see how to calculate these values:

$$FPR = \frac{FP}{FP + TN}$$

$$TPR = \frac{TP}{TP + FN}$$

Where the components of these equations are parts of the confusion matrix illustrated bellow.

		Predicted condition	
	Total population = P + N	Positive (PP)	Negative (PN)
Actual condition	Positive (P)	True positive (TP), hit	False negative (FN), type II error, miss, underestimation
	Negative (N)	False positive (FP), type I error, false alarm, overestimation	True negative (TN), correct rejection

Figure 1 - Confusion Matrix (Wikipedia 2022)

We calculate the points to create the curve using different thresholds, ranging from 0 to 1, to separate into binary classification. In the image below, we can see that a model within the blue line is better than green or yellow one.

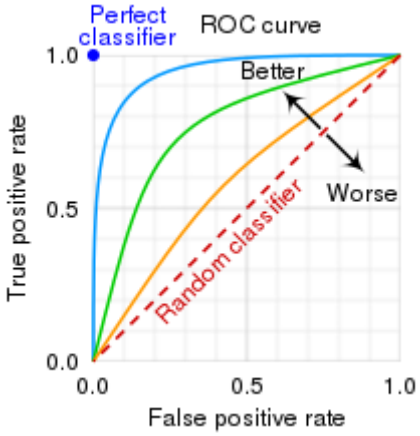

Figure 2 - ROC Curve (Wikipedia 2022)

Calculating the integral of the ROC curve is what give us the AUC score where greater values is better (the perfect classifier has AUC equals 1).

Analysis

Data Exploration

This project uses "SBA Case" dataset, a subset of "National SBA" dataset that includes data from 1987 to 2014 with 899,164 observations. The smaller one is a sample containing only loans from California State and has 2,102 observations. The data contains 35 columns described below. (Li, Mickel e Taylor 2018)

Variable name	Data type	Description of variable
LoanNr_ChkDgt	Text	Identifier – Primary key
Name	Text	Borrower name
City	Text	Borrower city
State	Text	Borrower state
Zip	Text	Borrower zip code
Bank	Text	Bank name
BankState	Text	Bank state
NAICS	Text	North American industry classification system code
ApprovalDate	Date/Time	Date SBA commitment issued
ApprovalFY	Text	Fiscal year of commitment
Term	Number	Loan term in months
NoEmp	Number	Number of business employees
NewExist	Text	1 = Existing business, 2 = New business
CreateJob	Number	Number of jobs created
RetainedJob	Number	Number of jobs retained
FranchiseCode	Text	Franchise code, (00000 or 00001) = No franchise
UrbanRural	Text	1 = Urban, 2 = rural, 0 = undefined
RevLineCr	Text	Revolving line of credit: Y = Yes, N = No
LowDoc	Text	LowDoc Loan Program: Y = Yes, N = No
ChgOffDate	Date/Time	The date when a loan is declared to be in default
DisbursementDate	Date/Time	Disbursement date
DisbursementGross	Currency	Amount disbursed
BalanceGross	Currency	Gross amount outstanding
MIS_Status	Text	Loan status charged off = CHGOFF, Paid in full = PIF
ChgOffPrinGr	Currency	Charged-off amount
GrAppv	Currency	Gross amount of loan approved by bank
SBA_Appv	Currency	SBA's guaranteed amount of approved loan

Figure 3 - 27 Original Variables of Dataset (Li, Mickel e Taylor 2018)

Variable name	Data type	Description of variable
New	Number	=1 if NewExist=2 (New Business), =0 if NewExist=1 (Existing Business)
Portion	Number	Proportion of gross amount guaranteed by SBA
RealEstate	Number	=1 if loan is backed by real estate, =0 otherwise
Recession	Number	=1 if loan is active during Great Recession, =0 otherwise
Selected	Number	=1 if the data are selected as training data to build model for assignment, =0 if the data are selected as testing data to validate model
Default	Number	=1 if MIS_Status=CHGOFF, =0 if MIS_Status=PIF
daysterm	Number	Extra variable generated when creating "Recession" in Section 4.1.6
xx	Number	Extra variable generated when creating "Recession" in Section 4.1.6

Figure 4 - 8 Additional Variables (Li, Mickel e Taylor 2018)

Exploratory Visualization

The dataset contains data from a wide number of years. It is important to note that the dataset has data from Financial Crisis of 2008. Let see how it influenced the default rate. In the chart below, we

can see that the deals closed since 2003 has an influence of Financial Crisis when the default rate starts to increase.

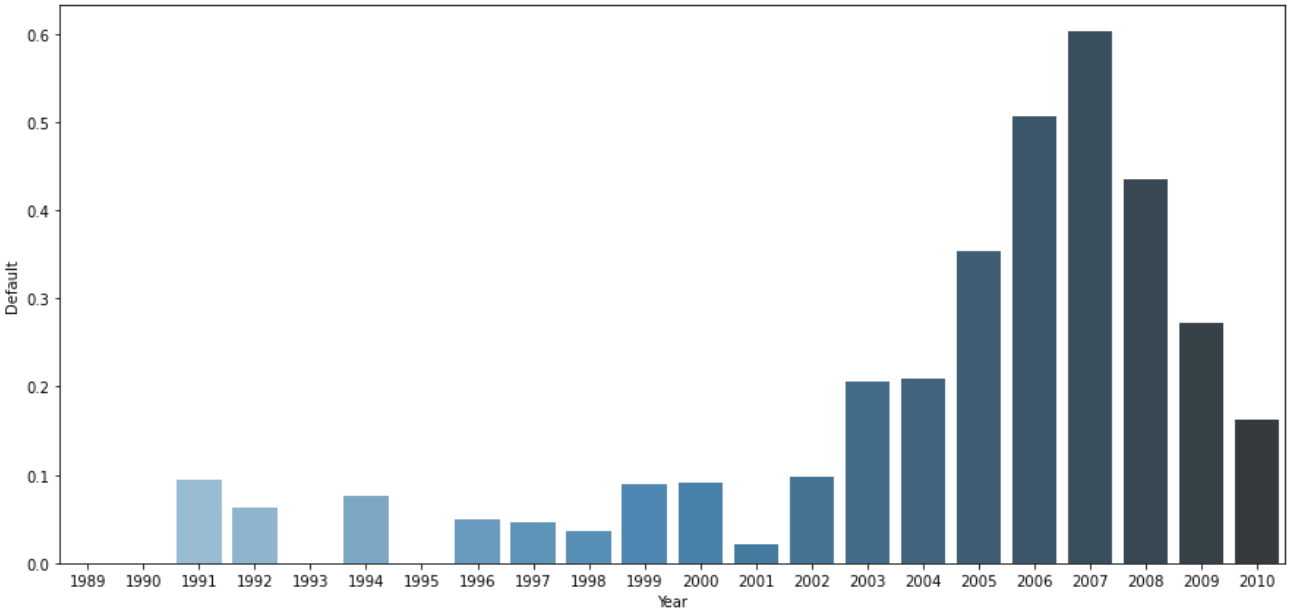

Figure 5 - Default Rate over the Years

Algorithms and Techniques

This project uses two classifiers algorithms that outputs the probability for each class in the problem.

The first algorithm used is **Logistic Regression**. It is a traditional algorithm, simple and effective. The algorithm has some issues with nonlinear data, sensible to outliers and correlated features. Nowadays we have some alternatives that can easily beat this one but we will evaluate that to compare with a more robust algorithm.

As a robust algorithm, I choose **LightGBM**. It is a tree-based algorithm, has many advantages of XGBoost and uses gradient boosting framework. The algorithm is known as faster to training, low memory usage with high accuracy.

Furthermore, I used some other techniques that involves the preprocessing and hyper parameter tuning, listed below:

- Box-cox Transformation: helps to normalize non-normal variables
- K-bins Discretization: for variables where box-cox didn't work I used K-bins with "quantile" parameter to split the variable into ordinal similar sample size
- One-hot Encoder: transforms categorical data into dummy variables
- Standard Scaler: standardize all variables
- Grid Search CV: hyper parameter tuning with k-fold cross-validation

Benchmark

As a benchmark, I found in the article that this project is based on, that the most parsimonious model, using three explanatory variables has an AUC score over than **0.75**. I expect that we can easily beat that score using Logistic Regression and even more with LightGBM.

Methodology

Data Preprocessing

During the EDA I found some variables that need to be dropped before modeling. Here I will list the motivation of the dropped ones:

Table 1 - Dropped variables

Variable	Motivation
Name	Categorical variable with too much distinct values that could make the ratio features/samples too high
Zip	
City	
Bank	
ApprovalFY	Date-related, dropped because macro-economy is a cycle and is dangerous to infer something using the date without more information
ApprovalDate	
DisbursementDate	
MIS_Status	Related with Default variable, so we keep only the Default
ChgOffPrinGr	
ChgOffDate	
State	Constant variable
BalanceGross	
daysterm	Described as extra variable
xx	
LoanNr_ChkDgt	ID
NAICS	ID for Industry Classification, the important info is the first two digits that are equal for the entire dataset
Selected	Suggestion of split train-test, I will create my own

Looking to numeric variables, I found some variables with outliers. To detect outliers I use 3 standard deviation from the mean (above and below). Below we can see the percent of outliers:

```
DisbursementGross    0.028109
GrAppv               0.027632
SBA_Appv             0.022392
CreateJob            0.014769
RetainedJob          0.011434
NoEmp                0.010005
LoanNr_ChkDgt        0.000000
Term                 0.000000
Portion              0.000000
Default              0.000000
```

Therefore, I plotted the distribution of those variables to see how it is distributed. Most of them are skewed to the right.

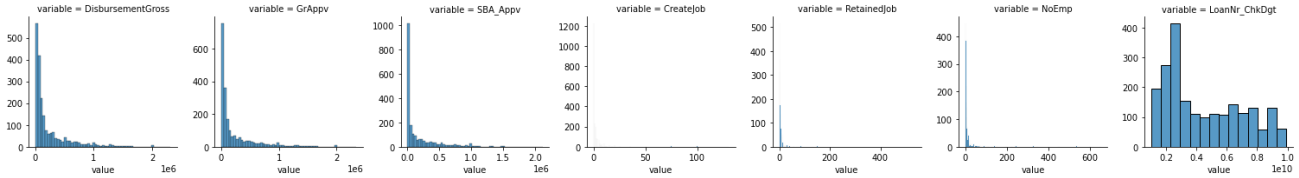

Figure 6 - Distribution before Transformation

To treat this I use two techniques, box-cox transformation and for that ones where the first didn't work I use k-bins discretization. After that, I checked for outliers now and none are found, the result is the one below.

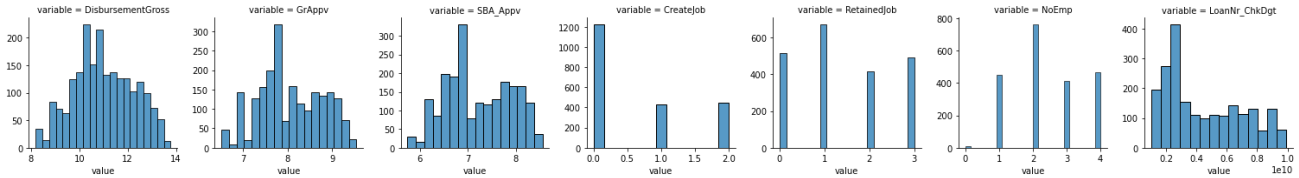

Figure 7 - Distribution after Transformation

For categorical features, I used one-hot encoder dropping the first one and treating missing as a feature itself. The goal here is to encode only the variables that result in a ratio feature per sample below 20 (arbitrary value to keep more samples than features and avoid overfitting). Variables that could compromise that premise were dropped.

Implementation

The implementation step was made in a new Jupyter Notebook after EDA. It follow these steps:

1. Data load and initial data treatment (manual transformations and dropping)
2. Train-test splitting and encapsulating all the transformers
3. Pipeline creation joining transformers, standardize scaler and classifier

Refinement

For refinement, we follow these steps:

1. Create a set for each parameter we want to test
2. Create Grid Search and set AUC as target metric
3. Fit the model

For Logistic Regression we test the following parameters:

- Penalty: None, L1, L2 or ElasticNet
- C (inverse of penalization): start with vast values and then I was funneling
- Solver: all possible

For LightGBM we test the following:

- Boosting Type: all possible
- Max Depth: from 3 to 12
- Number of Leaves: uses 2 ^ (Max Depth)

Results

Model Evaluation and Validation

To see the performance of the models I plotted the ROC curves using the test data.

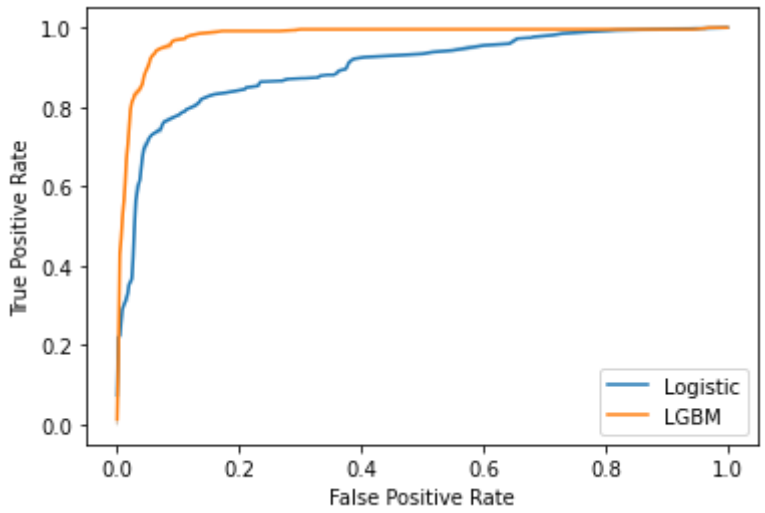

Figure 8 - ROC curve of Test Set

It's easy to see that the performance of LightGBM is superior than Logistic Regression. Let's compare the models with benchmark and see how the model performed over train and test data.

Table 2 - Model Performances

	Benchmark	Logistic Regression	LightGBM
Train	0.75	0.90	1.00
Test		0.90	0.98

The model has similar performances over train and test data indicating that overfitting is probably not a problem. The Logistic Regression surpasses the benchmark with considerably gain of performance and LightGBM had an outstanding performance with a score very close to one.

Conclusion

Reflection

The results obtained are good. We start with a benchmark specification of 0.75 for AUC score. The Logistic Regression model outperformed that value and by itself led us to good results.

We can see how relatively new algorithms can easily beat traditional ones. LightGBM was first released in 2016 and had an outstanding performance on the case above.

However, as an employee of a bank, it's clear that the case has a lot of simplifications and, in reality, calculating PD is not too simple. In reality, some variables could not be present when estimating PD like the flag marking the recession, most of the time when recession comes, I could stop issuing new loans, but I still have the older ones. Moreover, here we picked only the loans that have been approved, and a good part of the effort of credit modelling are in the denied part (generally called as Inference of

Denied – that is a model by itself, it is trying to discover how a denied loan could perform if it was approved).

Improvement

Some improvements that could be made are listed bellow:

- As mentioned in the reflection, it is important to know (and drop) which variables are future information, I mean, information that comes after the loan is approved. The right thing to do is considering only the variables that are present in the moment of approval.
- One topic that was not approached is Inference of Denied, and sometimes it is the most important part for credit expansion
- Other important topic is how the train-test split is made. Here I used a random approach, but in reality the correct to be made is split using dates. That way I can know how my model perform if I had implanted in some specific date. At the start of the project I tried to do this, but much of the contracts were issued close to the recession, and so if I had splitted in time I could face two problems:
    - If I train my model before contracts starts to be influenced by recession (around 2003 – most contracts had a term of 5+ years) I will have little data to train my model and the performance will probably be worst (lots of False Negatives)
    - If I trains my model picking part of recession, then my model will be a conservative one and probably have a lots of False Positive in the later years

As the article from the case used a random approach too, I decided to keep that way.

References

Li, Min, Amy Mickel, and Stanley Taylor. ""Should This Loan be Approved or Denied?": A Large Dataset with Class Assignment Guidelines." *Journal of Statistics Education* 26, no. 1 (2018): 55-66.

Wikipedia. 2022. https://en.wikipedia.org/wiki/Receiver_operating_characteristic (accessed March 18, 2022).