Credit Card Default Prediction

Harish Kollana

Data science trainees,

AlmaBetter, Bangalore.

Abstract:

The Taiwan Credit card issuer issues credit limits to the customer and in that there will be defaulters and non-defaulters. Based on the limit the issuer provided, Age, Education, Gender and other features the limit is provided.

We were provided with one such already classified label in our data set containing 30,000 observations with 25 columns.

Our experiments can help the issuer have a better understanding of their current and potential customers, which would inform their future strategy, including their planning of offering targeted credit products to their customers.

Keywords: Default, Gradient Boosting, Random Forest, Classifiers

1. Problem Statement

A Taiwan-based credit card issuer wants to better predict the likelihood of default for its customers, as well as identify the key drivers that determine this likelihood. This would inform the issuer's decisions on who to give a credit card to and what credit limit to provide.

X1: Amount of the given credit

X2: Gender (1 = male; 2 = female).

X3: Education (1 = graduate school; 2 = university; 3 = high school; 4 = others).

X4: Marital status (1 = married; 2 = single; 3 = others).

X5: Age (year).

X6 - X11: History of past payment. (-1 = pay duly; 1 = payment delay for one month; 2 = payment delay for two months; . . .; 8 = payment delay for eight months; 9 = payment delay for nine months and above.)

X12-X17: Amount of bill statement

X18-X23: Amount of previous payment

default.payment.next.month: default payment (1=yes,0=no

2. Introduction

Our goal here is to create a predictive model that identifies applicants who are relatively risky for a Credit card Default.

3. IsDefaulted:

Amount of the given credit (NT dollar): it includes both the individual consumer credit and his/her family (supplementary) credit.

6. Steps involved:

- ### Explorative Data ANALYSIS:

  After loading the dataset, we performed this method by comparing our target variable that is IsDefaulted with other independent variables. This process helped us figuring out various aspects and relationships among the target and the independent variables. It gave us a better idea of which feature behaves in which manner compared to the target variable.

- ### Feature Engineering:

  In Feature Engineering we will convert the Categories inside the column as values into model fit data using eval and other define functions.

- ### Fitting different models:

  For modelling we tried various Regression algorithms like:

  1. Logistic Regression
  2. Random Forest Classifier
  3. XGB Classifier

- ### Hyper Parameter Tuning:

  Tuning the hyperparameters of respective algorithms is necessary for getting better accuracy and to avoid overfitting in case of tree-based models like Random Forest Classifier and XGBoost classifier.

7. Algorithms:

- ### Linear Regression:

  Logistic regression is another powerful supervised ML algorithm used for binary classification problems (when target is categorical). The best way to think about logistic regression is that it is a linear regression but for classification problems. Logistic regression essentially uses a logistic function defined below to model a binary output variable (Tolles &

Meurer, 2016). The primary difference between linear regression and logistic regression is that logistic regression's range is bounded between 0 and 1. In addition, as opposed to linear regression, logistic regression does not require a linear relationship between inputs and output variables. This is due to applying a nonlinear log transformation to the odds ratio (will be defined shortly).

- ## Random Forest Classifier:

Random forest classifiers fall under the broad umbrella of ensemble-based learning methods [30]. They are simple to implement, fast in operation, and have proven to be extremely successful in a variety of domains [31,32]. The key principle underlying the random forest approach comprises the construction of many “simple” decision trees in the training stage and the majority vote (mode) across them in the classification stage. Among other benefits, this voting strategy has the effect of correcting for the undesirable property of decision trees to overfit training data [33]. In the training stage, random forests apply the general technique known as bagging to individual trees in the ensemble. Bagging repeatedly selects a random sample with replacement from the training set and fits trees to these samples. Each tree is grown without any pruning. The number of trees in the ensemble is a free parameter which is readily learned automatically using the so-called out-of-bag error [29].

Much like in the case of naïve Bayes– and k-nearest neighbor–based algorithms, random forests are popular in part due to their simplicity on the one hand, and generally good performance on the other. However, unlike the former two approaches, random forests exhibit a degree of unpredictability as regards the structure of the final trained model. This is an inherent consequence of the stochastic nature of tree building. As we will explore in more detail shortly, one of the key reasons why this characteristic of random forests can be a problem in regulatory reasons—clinical adoption often demands a high degree of repeatability not only in terms of the ultimate performance of an algorithm but also in terms of the mechanics as to how a specific decision is made.

- ## XGBOOST Classifier:

XGBoost is an ensemble learning method. Sometimes, it may not be sufficient to rely upon the results of just one machine learning model. Ensemble learning offers a systematic solution to combine the predictive power of multiple learners. The resultant is a single model which gives the aggregated output from several models.

The models that form the ensemble, also known as base learners, could be either from the same learning algorithm or different learning algorithms. Bagging and boosting are two widely used ensemble learners. Though these two techniques can be used with several statistical models, the most predominant usage has been with decision trees.

6.1 Model Performance:

The F-score, also called the F1-score, is a measure of a model's accuracy on a dataset. It is used to evaluate binary classification systems, which classify examples into 'positive' or 'negative'.

The F-score is a way of combining the precision and recall of the model, and it is defined as the harmonic mean of the model's precision and recall.

The F-score is commonly used for evaluating information retrieval systems such as search engines, and also for many kinds of machine learning models, in particular in natural language processing.

It is possible to adjust the F-score to give more importance to precision over recall, or vice-versa. Common adjusted F-scores are the F0.5-score and the F2-score, as well as the standard F1-score.

F-score Formula

The formula for the standard F1-score is the harmonic mean of the precision and recall. A perfect model has an F-score of 1.

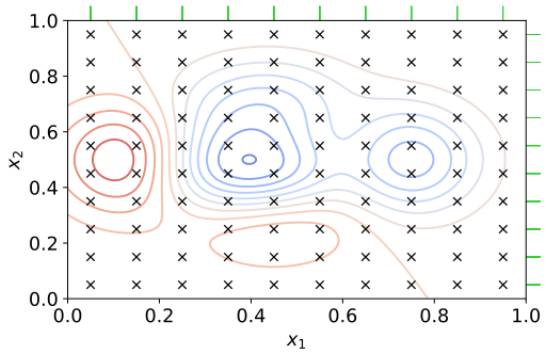
$$F_1 = \frac{2}{\frac{1}{\text{recall}} \times \frac{1}{\text{precision}}} = 2 \times \frac{\text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$$
$$= \frac{\text{tp}}{\text{tp} + \frac{1}{2}(\text{fp} + \text{fn})}$$

6.2. Hyper Parameters:

- ## Randomized Search Cv:

Random Search, the hyperparameters are chosen at random within a range of values that it can assume. The advantage of this method is that there is a greater

chance of finding regions of the cost minimization space with more suitable hyperparameters, since the choice for each iteration is random. The disadvantage of this method is that the combination of hyperparameters is beyond the scientist's control.



- ## Grid Search Cv:

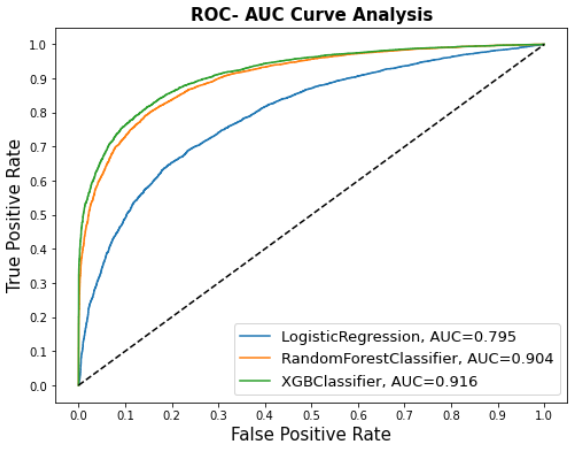
One can try the Manual Search method, by using the hit and trial process and can find the best hyperparameters which would take huge time to build a single model.

For this reason, methods like Random Search, GridSearch were introduced. Here, we will discuss how Grid Seach is performed and how it is executed with cross-validation in GridSearchCV.

Grid Search uses a different combination of all the specified hyperparameters and their values and calculates the performance for each combination and selects the best value for the hyperparameters. This makes the processing time-consuming and expensive based on the number of hyperparameters involved.

7. ROC-AUC Curve:

AUC - ROC curve is a performance measurement for the classification problems at various threshold settings. ROC is a probability curve and AUC represents the degree or measure of separability. It tells how much the model is capable of distinguishing between classes. Higher the AUC, the better the model is at predicting 0 classes as 0 and 1 classes as 1. By analogy, the Higher the AUC, the better the model is at distinguishing between customers with the defaulted and non defaulted.



Defining terms used in AUC and ROC Curve.

TPR (True Positive Rate) / Recall /Sensitivity

$$\text{TPR /Recall / Sensitivity} = \frac{TP}{TP + FN}$$

Specificity

$$\text{Specificity} = \frac{TN}{TN + FP}$$

FPR

$$\text{FPR} = 1 - \text{Specificity}$$

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

8. Conclusion:

We came to end stage by successfully building a model to predict whether the customer will default his / her payment

We have performed feature engineering, feature selection, hyperparameter tuning to prevent overfitting and for decreasing error.

The recall is the measure of our model correctly identifying True Positives. Thus, for all the Customers who actually default, recall tells us how many we correctly identified as is default.

As we had considered recall, XGBoost is our best model as we can see roc aoc curve is maximum.

References-

1. Analytics Vindhya
2. GeeksforGeeks
3. Medium