Capstone Project

Credit Card Default Prediction

Technical Documentation

By

Nidhi Pandey

AlmaBetter

Date: 29th June 2023.

Table of Contents

Abstract:

This technical document represents a rule-based regression predictive model for credit card default prediction. To predict whether the customer will default on their credit card payment next month, we can predict potential default accounts based on certain attributes. The idea is that the earlier the potential default accounts are detected, the lower the losses we will embrace. The dataset which was provided to us is 'default of credit card clients'. This dataset contains the amount of given credit, gender, education, marital status, age, history of past payment, amount of bill statement, amount of previous payment. While doing this project various models have been created. These various models are being analyzed and we tried to study various models to intuitively get the best performing model for our project.

An analysis with variable importance was carried to analyze the most significant variables for all the models developed with the given data sets considered. We are getting the best results from Random forest and XGBoost Classifier.

Problem Statement:

This project is aimed at predicting the case of customers' default payments in Taiwan. From the perspective of risk management, the result of predictive accuracy of the estimated probability of default will be more valuable than the binary result of classification - credible or not credible clients. We can use the K-S chart to evaluate which customers will default on their credit card payments.

Data Summary:

This research employed a binary variable, default payment (Yes = 1, No = 0), as the response variable. This study reviewed the literature and used the following 23variables as explanatory variables:

Attribute Information:

- X1: Amount of the given credit (NT dollar): it includes both the individual consumer credit and his/her family (supplementary) 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. We tracked the past monthly payment records (from April to September, 2005) as follows: X6 = the repayment status in September, 2005; X7 = the repayment status in August, 2005; . . .;X11 = the repayment status in April, 2005. The measurement scale for the repayment status is: -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 (NT dollar). X12 = amount of bill statement in September, 2005; X13 = amount of bill statement in August, 2005; . . .; X17 = amount of bill statement in April, 2005.
- X18-X23: Amount of previous payment (NT dollar). X18 = amount paid in September, 2005; X19 = amount paid in August, 2005;. . .;X23 = amount paid in April, 2005.

Steps involved:

1. Exploratory Data Analysis

After loading the dataset we compared our target variable that is the default payment next month with other independent variables. This process helped us figure out various aspects and relationships among the dependent and the independent variables. It gave us a better idea of which feature behaves in which manner compared to the dependent variable.

2. Null values Treatment

Our dataset didn’t have any null values to be treated.

3. Encoding of categorical columns

We used One Hot Encoding(converting to dummy variables) to produce binary integers of 0 and 1 to encode our categorical features because categorical features that are in string format cannot be understood by the machine and needs to be converted to the numerical format.

4. Feature Selection

In these steps, we used correlation, VIF analysis to check the results of each feature i.e which feature is more important compared to our model and which is of less importance.

5. Standardization of features

Our main motive through this step was to scale our data into a uniform format that would allow us to utilize the data in a better way while performing fitting and applying different algorithms to it.

The basic goal was to enforce a level of consistency or uniformity to certain practices or operations within the selected environment.

6. Fitting different models

For modeling, we tried various classification algorithms like:

- Logistic regression
- Decision Tree
- Random Forest regression
- Gradient boosted
- XGBoost regression

7. Tuning the hyperparameters for better accuracy

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

8. Features Explainability

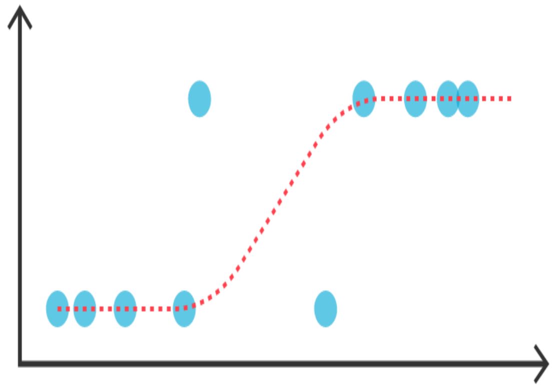
We have applied ROC AUC to all the models to determine the features that were most important while predicting an instance and also build a feature importance graph to find out which features were important and which were redundant in a model

Algorithms:

1. Logistic Regression:

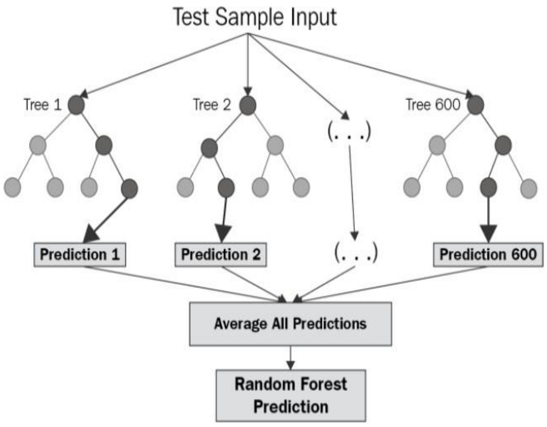
Logistic Regression is a process of modeling the probability of a discrete outcome given an input variable. The most common logistic regression models a binary outcome; something that can take two values such as true/false, yes/no, and so on. Multinomial logistic regression can model scenarios where there are more than two possible discrete outcomes. Logistic regression is a useful analysis method for classification problems, where you are trying to determine if a new sample fits best into a category. As aspects of cyber security are classification problems, such as attack detection, logistic regression is a useful analytic technique.It is used in statistical software to understand the relationship between the dependent variable and one or more independent variables by estimating probabilities using a logistic regression equation. This type of analysis can help you predict the likelihood of an event happening or a choice being made.

The optimization algorithm used is Gradient Descent. We also performed different types of regularization techniques to prevent overfitting in the model.



2. Random Forest Regression:

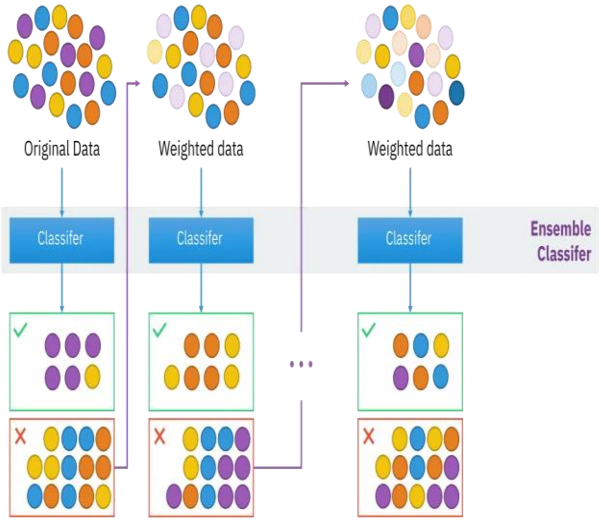
Random Forest is a bagging type of Decision Tree Algorithm that creates several decision trees from a randomly selected subset of the training set and n features, collects the values from these subsets, and then averages the final prediction out of all n number of decision trees



3. Gradient Boosting:

Gradient boosting is an approach where new models are created that predict the residuals or errors of prior models and then added together to make the final prediction. It is called gradient boosting because it uses a gradient descent algorithm to minimize the loss when adding new models.

This approach supports both regression and classification predictive modeling problems.



4. XGBoost:

XGBoost is an algorithm that has recently been dominating applied machine learning and Kaggle competitions for structured or tabular data.XGBoost is an implementation of gradient boosted decision trees designed for speed and performance.

The implementation of the algorithm was engineered for the efficiency of computing time and memory resources. A design goal was to make the best use of available resources to train the model. Some key algorithm implementation features include:

- **Sparse Aware:** implementation with automatic handling of missing data values.

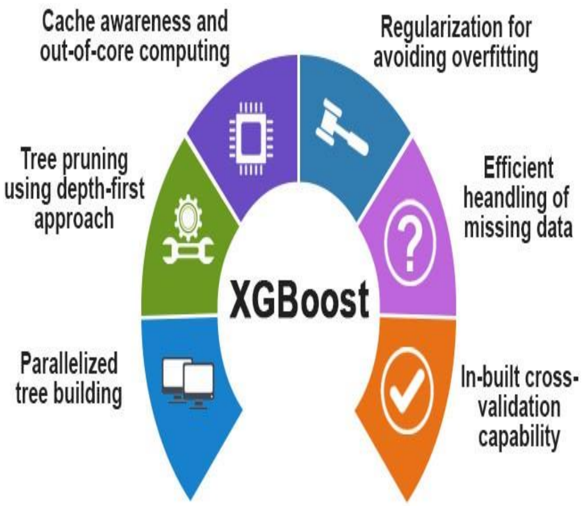
- **Block Structure:** to support the parallelization of tree construction.
- **Continued Training:** so that you can further boost an already fitted model on new data.

XGBoost is free open source software available for use under the permissive Apache-2 license.

Why Use XGBoost?

The two reasons to use XGBoost are also the two goals of the project:

1. Execution Speed.
2. Model Performance.


Cache awareness and out-of-core computing
Regularization for avoiding overfitting
Tree pruning using depth-first approach
Efficient heandling of missing data
XGBoost
Parallelized tree building
In-built cross-validation capability

Model performance:

The model can be evaluated by various metrics such as:

1. Mean square error

The MSE of an estimator measures the average of the squares of the errors—that is, the average squared difference between the estimated values and the actual value.

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(Y_i - \hat{Y}_i)^2.$$

2. Root mean square error

RMSE is just the root of MSE. It is directly interpretable in terms of measurement units, and so is a better measure of goodness of fit than a correlation coefficient. One can compare the RMSE to observed variation in measurements of a typical point

$$\mathrm{RMSD} = \sqrt{\frac{\sum_{i=1}^{N}(x_i - \hat{x}_i)^2}{N}}$$

3. R square:

R-Squared is a statistical measure of fit that indicates how much variation of a dependent variable is explained by the independent variable(s) in a regression model. It has one limitation that its value increases as the number of Parameters increase even if that parameter does not improve model

$$SSE = \sum_{i=1}^{m}(y_i - \hat{y}_i)^2$$

$$SST = \sum_{i=1}^{m}(y_i - \bar{y})^2$$

$$R^2 = 1 - \frac{SSE}{SST}$$

ŷ is the predicted value, y is the actual value and ȳ is the mean .

4. Adjusted R Square:

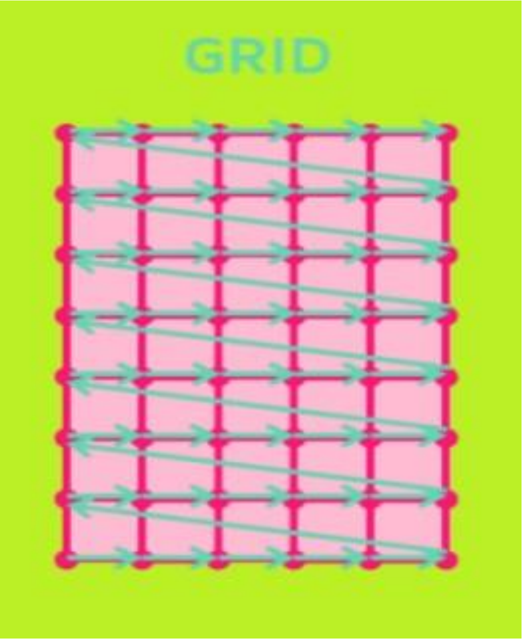
Adjusted R-squared is a modified version of R-squared that overcomes the problem of r2 and has been adjusted for the number of predictors in the model. The adjusted R-squared increases when the new term improves the model more than would be expected by chance. It decreases when a predictor improves the model by less than expected.

Hyperparameter tuning:

Hyperparameters are sets of information that are used to control the way of learning an algorithm. Their definitions impact parameters of the models, seen as a way of learning, change from the new hyperparameters. This set of values affects the performance, stability, and interpretation of a model. Each algorithm requires a specific hyperparameters grid that can be adjusted according to the business problem. Hyperparameters alter the way a model learns to trigger this training algorithm after parameters to generate outputs. We used Grid Search CV, Randomized Search CV, and Bayesian Optimization for hyperparameter tuning. This also results in cross-validation and in our case we divided the dataset into different folds. The best performance improvement among the three was by Bayesian Optimization.

1. Grid Search CV-Grid:

Search combines a selection of hyperparameters established by the scientist and runs through all of them to evaluate the model's performance. Its advantage is that it is a simple technique that will go through all the programmed combinations. The biggest disadvantage is that it traverses a specific region of the parameter space and cannot understand which movement or which region of the space is important to optimize the model.



Conclusion:

- Logistic Regression is the least accurate as compared to other models performed.
- After performing the various models we get the best accuracy from the Random forest and XGBoost classifier.
- XGBoost has the best precision and the recall balance.
- Higher recall can be achieved if low precision is acceptable.
- We can deploy the model and can serve as an aid to human decision.
- Model can be improved with more data and computational resources