Technical Documentation

Credit card Default Prediction

Prabin Deb Nath

AlmaBetter, Bangalore.

Abstract:

Financial threats are displaying a trend about the credit risk of commercial banks as the incredible improvement in the financial industry has arisen. In this way, one of the biggest threats faces by commercial banks is the risk prediction of credit clients. In classification problems, an imbalanced dataset is also crucial to improve the performance of the model because most of the cases lied in one class, and only a few examples are in other categories. Traditional statistical approaches are not suitable to deal with imbalanced data. Data level resampling techniques are employed to overcome the problem of data imbalance. Various under sampling and oversampling methods are used to resolve the issue of class imbalance. Different machine learning models are also employed to obtain efficient results. The split method is utilized to validate the results in which data has been split into training and test sets.

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:

The data set contains the account holder information (Age, Marital Status, Gender, history of payments, etc), and whether customer will default his next month payment or not

Attribute Information:

- X1 - Amount of credit (includes individual as well as family 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 to X11 - History of past payments from April to September
- X12 to X17 - Amount of bill statement from April to September
- X18 to X23 - Amount of previous payment from April to September
- Y - Default payment next month

Steps involved:

1. Exploratory Data Analysis

After loading the dataset we compared our target variable that is "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 data set didn't have any null values to be treated.

3. Outliers treatment

We used Isolation Forest Algorithm to identify and treat outliers. Isolation forest is a tree-based algorithm that is very effective for both outlier and novelty detection in high-dimensional data.

4. 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.

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 Classification
- XGBoost Classification
- CatBoost Classification

7. Tuning the hyperparameters for better recall

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 SHAP on the XGBoost model 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

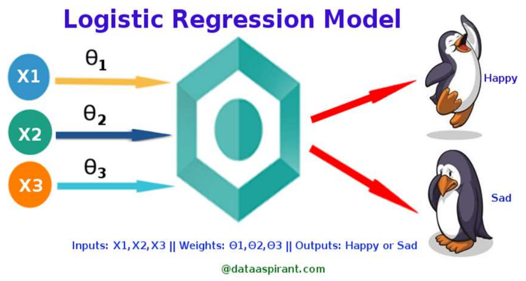
Algorithms:

1. Logistic Regression:

Logistic regression was used in the biological sciences in the early twentieth century. It was then used in many social science applications. Logistic Regression is used when the dependent variable(target) is categorical.

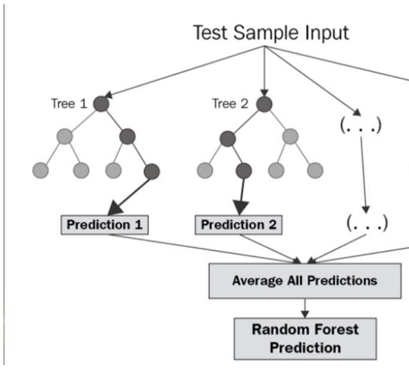
For example,

- To predict whether an email is spam (1) or (0)
- Whether the tumor is malignant (1) or not (0)


Logistic Regression Model
X1
X2
X3
θ1
θ2
θ3
Happy
Sad
Inputs: X1,X2,X3 || Weights: Θ1,Θ2,Θ3 || Outputs: Happy or Sad
@dataaspirant.com

2. Random Forest Classification:

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. XGBoost Classification:

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.

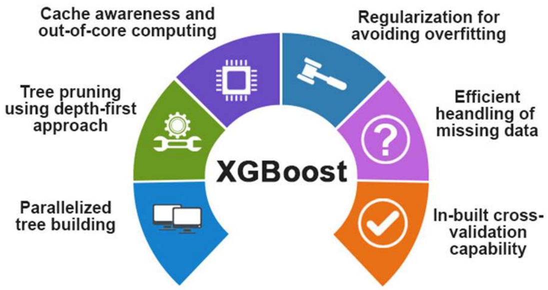
- **Block Structure** to support the parallelization of tree construction.
- **Continued Training** so that you can further boost an already fitted model on new data.

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

Why Use XGBoost?

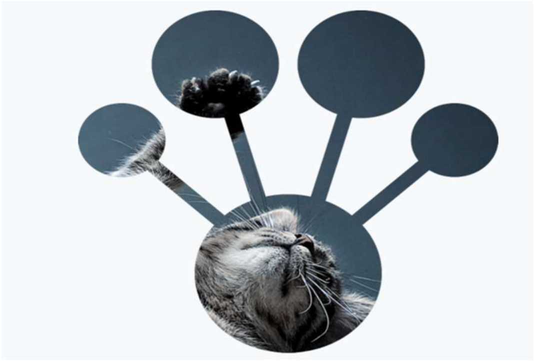
The two reasons to use XGBoost are also the two goals of the project:

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

4. CatBoost Classification:

CatBoost is a recently open-sourced machine learning algorithm from Yandex. It can easily integrate with deep learning frameworks like Google's TensorFlow and Apple's Core ML. It can work with diverse data types to help solve a wide range of problems that businesses face today. To top it up, it provides best-in-class accuracy.



It is especially powerful in two ways:

- It yields state-of-the-art results without extensive data training typically required by other machine learning methods, and
- Provides powerful out-of-the-box support for the more descriptive data formats that accompany many business problems.

"CatBoost" name comes from two words "Category" and "Boosting".

Model performance:

The model can be evaluated by various metrics such as:

1. Accuracy

Accuracy is the most intuitive performance measure and it is simply a ratio of correctly predicted observation to the total observations. One may think that, if we have high accuracy then our model is best. Yes, accuracy is a great measure but only when you have symmetric datasets where values of false positive and false negatives are almost the same. Therefore, you have to look at other parameters to evaluate the performance of your model. For our model, we have got 0.803 which means our model is approx. 80% accurate.

Accuracy = TP+TN/TP+FP+FN+TN

2. Precision

Precision is the ratio of correctly predicted positive observations to the total predicted positive observations.

Precision = TP/TP+FP

3. Recall

Recall (Sensitivity) - Recall is the ratio of correctly predicted positive observations to all observations in the actual class.

Recall = TP/TP+FN

4. F1-Score

F1 Score is the weighted average of Precision and Recall. Therefore, this score takes both false positives and false negatives into account. Intuitively it is not as easy to understand as accuracy, but F1 is usually more useful than accuracy, especially if you have an uneven class distribution. Accuracy works best if false positives and false negatives have similar costs. If the cost of false positives and false negatives are very different, it's better to look at both Precision and Recall.

F1 Score = 2*(Recall * Precision) / (Recall + Precision)

5. AUC-ROC

The Receiver Operator Characteristic (ROC) curve is an evaluation metric for binary

classification problems. It is a probability curve that plots the TPR against FPR at various threshold values and essentially separates the 'signal' from the 'noise'. The Area Under the Curve (AUC) is the measure of the ability of a classifier to distinguish between classes and is used as a summary of the ROC curve**.**

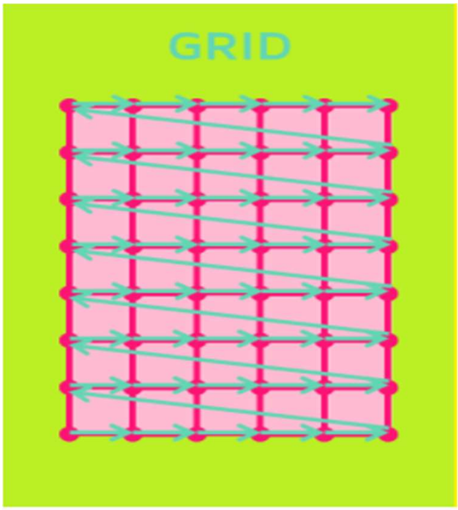
Hyperparameter tuning:

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

That's it! We reached the end of our exercise.

Starting with loading the data so far, we have done EDA, null values treatment, encoding of categorical columns, feature selection, and then model building.

The default rate is higher for males, increases as the education increases, also increase as the age of a person increases. i.e., clients whose age over 60 was higher than mid-age and young people.

In all of these models, our recall revolves in the range of 75 to 82% with the best fit model as XGBoost.