Domain background

Supervised learning is the machine learning task of inferring a function from labeled training data. The training data consist of a set of training examples. In supervised learning, each example is a pair consisting of an input object (typically a vector) and a desired output value (also called the supervisory signal). A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples. An optimal scenario will allow for the algorithm to correctly determine the class labels for unseen instances. This requires the learning algorithm to generalize from the training data to unseen situations in a "reasonable" way.

Nowadays, machine learning has been used vastly in insurance companies in order to recommend decent contracts to their clients based on the past information of them. For example, buying a new car and its insurance need a precise evaluation on both side client and insurance company to make a decent contract. Nothing ruins the thrill of buying a brand new car more quickly than seeing your new insurance bill. The sting's even more painful when you know you're a good driver. It doesn't seem fair that you have to pay so much if you've been cautious on the road for years.

Therefore, having a robust predictive model is a key factor for successful companies.

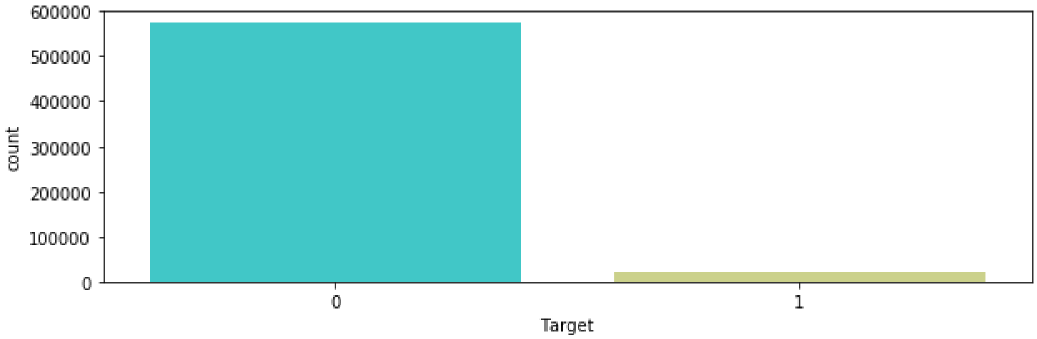
Problem statement

Porto Seguro, one of Brazil's largest auto and homeowner insurance companies. Inaccuracies in car insurance company's claim predictions raise the cost of insurance for good drivers and reduce the price for bad ones. The goal of this project is to build a model that predicts the probability that a driver will initiate an auto insurance claim in the next year. An accurate prediction will allow them to further tailor their prices, and hopefully make auto insurance coverage more accessible to more drivers.

Thus, I will predict the probability that an auto insurance policy holder files a claim. The dataset is avaible on Kaggle dataset.

Dataset and inputs

The dataset contains 2 .csv files with information necessary to make a prediction. In the train and test data, features that belong to similar groupings are tagged as such in the feature names (e.g., ind, reg, car, calc). In addition, feature names include the postfix bin to indicate binary features and cat to indicate categorical features. Features without these designations are either continuous or ordinal. Values of -1 indicate that the feature was missing from the observation. In this challenge the dataset target distribution is imbalance. The target has 573.518 one and 21.694k zero. Therefore we also have to deal with imbalance dataset.



train.csv contains the training data, where each row corresponds to a policy holder, and the target columns signifies that a claim was filed. The shape of train is: (595212, 59) and memory usage: 267.9 MB.

```
id            595212 non-null int64
target          595212 non-null int64
ps_ind_01       595212 non-null int64
ps_ind_02_cat    595212 non-null int64
ps_ind_03       595212 non-null int64
ps_ind_04_cat    595212 non-null int64
ps_ind_05_cat    595212 non-null int64
ps_ind_06_bin    595212 non-null int64
ps_ind_07_bin    595212 non-null int64
ps_ind_08_bin    595212 non-null int64
ps_ind_09_bin    595212 non-null int64
ps_ind_10_bin    595212 non-null int64
ps_ind_11_bin    595212 non-null int64
ps_ind_12_bin    595212 non-null int64
ps_ind_13_bin    595212 non-null int64
ps_ind_14       595212 non-null int64
ps_ind_15       595212 non-null int64
ps_ind_16_bin    595212 non-null int64
ps_ind_17_bin    595212 non-null int64
ps_ind_18_bin    595212 non-null int64
```

```
ps_reg_01       595212 non-null float64
ps_reg_02       595212 non-null float64
ps_reg_03       595212 non-null float64
ps_car_01_cat    595212 non-null int64
ps_car_02_cat    595212 non-null int64
ps_car_03_cat    595212 non-null int64
ps_car_04_cat    595212 non-null int64
ps_car_05_cat    595212 non-null int64
ps_car_06_cat    595212 non-null int64
ps_car_07_cat    595212 non-null int64
ps_car_08_cat    595212 non-null int64
ps_car_09_cat    595212 non-null int64
ps_car_10_cat    595212 non-null int64
ps_car_11_cat    595212 non-null int64
ps_car_11       595212 non-null int64
ps_car_12       595212 non-null float64
ps_car_13       595212 non-null float64
ps_car_14       595212 non-null float64
ps_car_15       595212 non-null float64
ps_calc_01      595212 non-null float64
ps_calc_02      595212 non-null float64
ps_calc_03      595212 non-null float64
ps_calc_04      595212 non-null int64
ps_calc_05      595212 non-null int64
ps_calc_06      595212 non-null int64
ps_calc_07      595212 non-null int64
ps_calc_08      595212 non-null int64
ps_calc_09      595212 non-null int64
ps_calc_10      595212 non-null int64
ps_calc_11      595212 non-null int64
```

```
ps_calc_12       595212 non-null int64
ps_calc_13       595212 non-null int64
ps_calc_14       595212 non-null int64
ps_calc_15_bin    595212 non-null int64
ps_calc_16_bin    595212 non-null int64
ps_calc_17_bin    595212 non-null int64
ps_calc_18_bin    595212 non-null int64
ps_calc_19_bin    595212 non-null int64
ps_calc_20_bin    595212 non-null int64
```

test.csv contains the test data. The shape of dataframe is: (892816, 58) and memory usage: 395.1 MB.

Solution statement

Machine learning techniques have been applied effectively in the insurance companies over the years. In this project, I will build a pipeline for doing end-to-end process of data analysis, data cleaning, features engineering, modeling optimization and save the final model automatically. Besides, different machine learning methods will be used to train our predictive model and to classify the client of Porto Seguro company.

In this project for modeling, I will stack several layers of predictions to have a robust model.

Level 1: I will stack Xgboost, LightGBM, Random Forest, Extra Tree, Logistic Regression, etc.

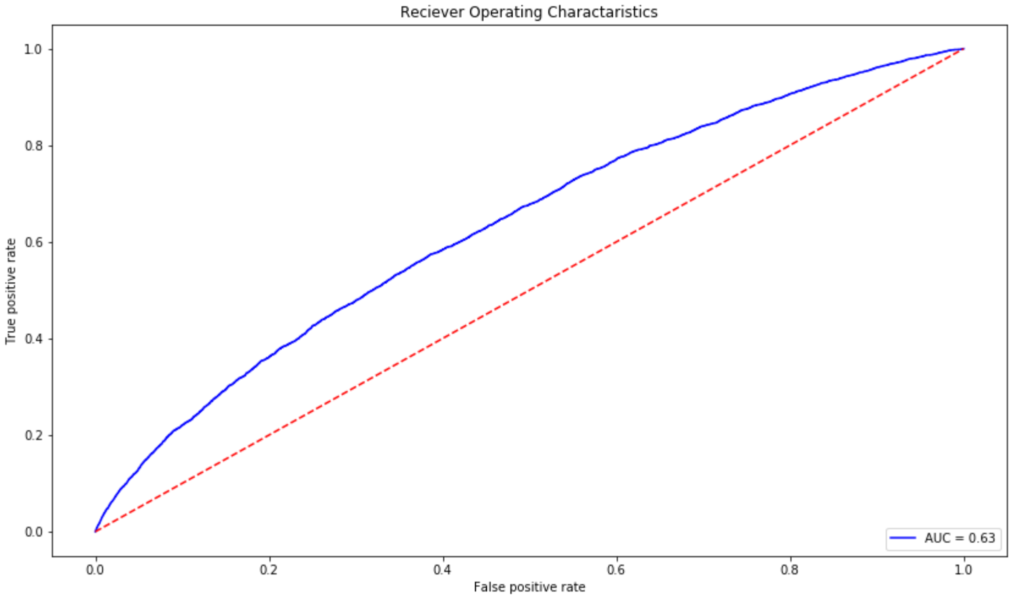
Level 2: I will use L1, Xgboost, LightGBM, Random Forest, Logistic Regression, etc.

Level 3: I will stack Xgboost, Logistic Regression.

Finally, I will take average of our output. However, I might change some models during the calculation based on the final result to have stronger model.

Additionally, I am going to use different type of machine learnings, non-tree based model such as logistic regression, tree-based model such as random forest and ensemble model such as Xgboost. The preparation data for non-tree and tree-based models are different. Therefore, I have to do different preprocessing data analysis for each model. Besides, each model has to optimized by its gridding method. Furthermore, If I can find a way to drop some of the results in the different levels of model, we will have a stronger model which is look like neural network.

Benchmark model

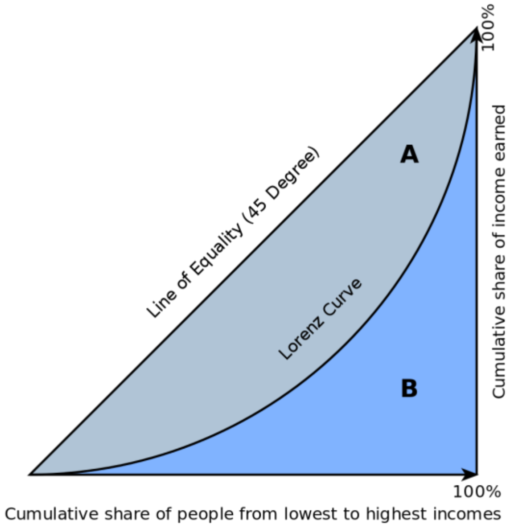


In this section, I show the result of the a logistic regression model which is applied on this problem. Our metric is Gini which is related to Gini = 2AUC – 1. The plot shows AUC = 0.63. I am going to improve our prediction result.

Evaluation metric

Our problem is binary classification. For binary classifiers, this option reports the AUC (area under curve) and Gini coefficient evaluation metrics. Both of these evaluation metrics are calculated together for each binary model. The values of the metrics are reported in a table in the analysis output browser.

The AUC evaluation metric is calculated as the area under an ROC (receiver operator characteristic) curve, and is a scalar representation of the expected performance of a classifier. The AUC is always between 0 and 1, with a higher number representing a better classifier. A diagonal ROC curve between the coordinates (0,0) and (1,1) represents a random classifier, and has an AUC of 0.5. Therefore, a realistic classifier will not have and AUC of less than 0.5.


100%
Line of Equality (45 Degree)
A
Lorenz Curve
B
Cumulative share of income earned
100%
Cumulative share of people from lowest to highest incomes

The Gini coefficient evaluation metric is sometimes used as an alternative evaluation metric to the AUC, and the two measures are closely related. The Gini coefficient is calculated as twice the area between the ROC curve and the diagonal, or as Gini = 2AUC - 1. The Gini coefficient is always between 0 and 1, with a higher number representing a better classifier. The Gini coefficient is negative in the unlikely event that the ROC curve is below the diagonal.

The Gini coefficient measures the inequality among values of a frequency distribution (for example, levels of income). A Gini coefficient of zero expresses perfect equality, where all values are the same (for example, where everyone has the same income). A Gini coefficient of 1 (or 100%) expresses maximal inequality among values (e.g., for a large number of people, where only one person has all the income or consumption, and all others have none, the Gini coefficient will be very nearly one). However, a value greater than one may occur if some persons represent negative contribution to the total (for example, having negative income or wealth).

The metric used for this Kaggle competition is the Normalized Gini Coefficient. During scoring, observations are sorted from the largest to the smallest predictions. Predictions are only used for ordering observations; therefore, the relative magnitude of the predictions are not used during scoring. The scoring algorithm then compares the cumulative proportion of positive class observations to a theoretical uniform proportion.

Project design

1. Data analysis: processing of inspecting, cleansing, transforming, and modeling data with the goal of discovering useful information, suggesting conclusions, and supporting decision-making.
2. Data cleaning: detecting and correcting (or removing) corrupt or inaccurate records from dataset and refers to identifying incomplete, incorrect, inaccurate or irrelevant parts of the data and then replacing, modifying, or deleting the dirty or coarse data. For example, dealing with Nan values, outliers, categorical features, ordinal feature, and ... .
3. Feature engineering: transforming raw data into features that better represent the underlying problem to the predictive models, resulting in improved model accuracy on unseen data.
4. Modeling: Stacking different machine learning methods to build a predictive model.
5. Optimization: I will optimize my model to have better prediction.
6. Finalization: save the model and preparing for the final usage.