CS37300 Kaggle Competition Report

Aayla Secura / Yunhu Kim / 0323523302

December 11, 2022

1 Introduction

For this competition we had to predict whether a person would repay their loan.

2 Pre-processing performed

1. Following columns are dropped: ID, months since last delinq, and race.
ID was dropped because every row has one unique ID, race was removed to follow law, and months since last delinq was removed because 50 percent values of NA.

2. Changed employment length column from categorical to numerical

3. Filled empty categorical values with NA, and filled empty numerical value with average number

4. contain the most frequent values in each column. I contained at least 80 percent of the distribution of the each columns, then set every other values as Others.

5. check difference of distribution of training dataset and distribution of the test dataset using K-Mean Cluster.

6. create validation dataset that contains same distribution of the test dataset.

7. Scalar numerical columns.

8. one-hot encoding categorical columns

3 Knowledge Representation

Here you must describe the model you used. For instance, the data is $D = \{x_i, y_i\}_{i=1}^{n}$, were x_i are the features of the i-th customer and $y_i \in \{0, 1\}$ describes whether customer i repaid its loan.

The model representation is **Gradient Boosting Regressor**
First, I use K-Mean Clustering to get covariance shift weights.
Then I used GradientBoostingRegressor with base classifier Decision Tree max depth = 7 and 110 estimators.
Then I create Threshold based on AUROC Optimization. This will help classifying imbalanced dataset.
then by Using validation dataset that has same distribution of the test dataset and K-Cross Validation, I checked check best parameters.

model parameters: Weight of the each decision trees, and a set of node attributes for each trees.

Input: D, base-classifier: Decision Tree, n-estimators: 110.

Output of GradientBoostingRegressor: prediction of the yi based on xi

Output of model: prediction of the class of the yi based on the xi, yi and threshold. if yi ¿ threshold, then yi = 1. Else yi = 0.

Model Space: 110 times the model space of the decision tree and weights used to combine predictors
Model space of decision Tree: A set of trees with node attributes

4 Score Function

Loss function: Mean Squared Error
n = number of datapoints
y = true y value
y_i = true i th value of y value
$\hat{y}$ = predicted y value
$\hat{y_i}$ = predicted y$_i value$
$Formula = 1/n * \sum_{i=1}^{n}(y_i - \hat{y_i})^2$

Score function
$\omega(W)$ = penalty for model complexity
$LossFunction(D_train; W)$ is MSE
W is model parameter
$F = LossFunction(D_train; W) + \omega(W)$

5 Search Method

Here you describe your search method.

The search function (how are you finding good models?) E.g., for gradient boosted decision trees, please describe the gradient (gradient of which function?) and how it is used (e.g., gradient descent? or gradient ascent?) and boosting is used.

For the search function of the decision tree, I used gini entropy. Then for the search function of the gradient boost, search function is gradient descent of the weight of the decision trees.