IE 582

PROJECT

Imbalanced Classification Problem

Instructor: Mustafa Gökçe Baydoğan, Ph.D., Professor

Abdulsamed Kağıt

Kadir Ahmet Aksu

İlker Zeybek

February 15, 2021

TABLE OF CONTENTS

1. INTRODUCTION

In statistics, classification is the problem of identifying to which of a set of categories a new observation belongs, on the basis of a training set of data containing observations whose class label is known. Classification is considered as an instance of supervised learning.[1] The aim of this project is to build a classification model for the given data based on the methods discussed in the IE 582 course.

Various classification approaches have been used to find best possible model that represents the true underlying relationship between the input and output variables. Firstly, the existing class imbalance in the dataset was ignored, and Random Forest, Gradient Boosting, and Penalized Regression models were built with 3 times 10-fold-cross validation. To be able to tune the hyperparameters of Penalized Regression, Random Forest, and Gradient Boosting, grid search method is used. During the training of these models, different preprocessing methods are used. These preprocessing methods include excluding near zero variance and zero variance features, centering the features according to mean, and scaling in order to avoid biases towards features with wider ranges in distance calculations. Secondly, since dataset used in this project has built-in class-imbalance, same models stated above are applied with the calculated weights of the classes. Best performing model for this dataset was Stochastic Gradient Boosting model with assigned class weights. Tuned hyperparameters of this model and details of the model implementation will be discussed in later chapters.

2. RELATED LITERATURE

Classification is a significant pattern recognition task. A variety of classification algorithms, such as decision tree, backpropagation neural network, Bayesian network, nearest neighbor, support vector machines, and associative classification, have been well developed and successfully applied to many application domains. However, for most classification algorithms that assume a reasonably balanced distribution, datasets that possess class-imbalance have encountered a serious challenge. The imbalance is characterized as having many more instances than others of certain classes. Classification algorithms that predict small classes tend to be uncommon, undiscovered or overlooked as rare instances occur infrequently; thus, test samples belonging to small classes are misclassified more often than those belonging to the dominant classes. The proper classification of the samples in small groups also has a higher importance in such applications than in the opposite case.[2]

In many areas of great significance to the data mining community, the class imbalance issue is a common one. Here are few examples that illustrate these cases[3]:

- **Fraud Detection.** Frauds such as credit card frauds are an important for many businesses. In these businesses, fraudulent transactions have smaller volume compared to legitimate transactions.
- **Medical Diagnosis.** In the medical datasets, disease class usually has much less instances compared to non-disease class.

3. SOLUTION METHODS AND RESULTS

In this project, we are given a dataset, which is collected for classification purpose. In the training dataset, there are 2074 instances with 60 independent variables and 1 dependent variable, which is class label of a particular instance. The test dataset contains 2073 observations with the same feature setting. The aim of the project is to build a classifier that labels individual instances successfully with high accuracy. Various classification algorithms learned to identify classes of the instances and they are trained on the given training dataset. Upcoming sub-chapters will discuss how these algorithms are implemented.

3.1 Random Forest

Random Forest is an ensemble learning method for classification and regression problems that operate by constructing a multitude of decision trees at training time. Random forests are frequently used as "blackbox" models in businesses, as they generate reasonable predictions across a wide range of data.[4]

Firstly, as stated before, we did not consider the class imbalance present in the data set. To be able to find best parameter setting for the Random Forest algorithm, we have used 3 times 10-fold cross-validation with a grid search method. The grid search of the parameters is listed below:

- mtry: 2, 5, 7, 10, 15, 30, and 60.
- Split rule: Gini impurity and extra trees.
- Minimum node size: 1, 3, 5, and 10.

After performing the cross-validation and grid search, best parameter setting for Random Forest algorithm without class weights is found to be:

- mtry = 5.
- Split rule = Gini impurity.
- Minimum node size = 5.

The ROC metric for the best parameter setting is found to be 0.8836179 as a result of 3 times 10-fold cross-validation. Sensitivity and specificity ratios are 0.9505893 and

0.5009020, respectively. Low specificity value indicates that this classifier is misclassifying the class that has fewer observations.

Secondly, class-imbalance present in the dataset has been included in the training of a Random Forest algorithm as class-weights. Best parameters for this model are searched with 3 repeats 10-fold cross-validation and grid search. The grid search of the parameters is same with the previous model training phase. As a result of the grid search, tuned parameters of the class-weighted Random Forest algorithm are:

- mtry = 7.
- Split rule = Gini impurity.
- Minimum node size = 5.

The ROC metric for the tuned algorithm is 0.8829239, which is slightly worse than the previous approach. Sensitivity and specificity ratios are 0.8621863 and 0.7045752, respectively. Our loss in sensitivity is relatively smaller than the gain in the specificity ratio, while ROC metric is slightly lower than the previous approach. Assigning weights to the classes has significantly improved the predictions of the class with fewer observations.

3.2 Gradient Boosting

Gradient Boosting is a machine learning method for classification and regression problems, which produces a predictive model with ensemble of weak prediction models in an iterative fashion. [5]

Firstly, a Gradient Boosting model fitted to the training data without considering the class-imbalance prevalent in the dataset. Grid search and 10-fold cross-validation with 3 repeats is used to find best possible parameter setting for this model. The grid search parameters for the Gradient Boosting model can be seen below:

- Interaction depth: 1, 3, and 5.
- Number of trees: (1:50)*50
- Learning rate: 0.01 and 0.001
- Minimum observations in node: 10.

As a result of the grid search with cross-validation, best parameter setting for this model is:

- Interaction depth = 5.
- Number of trees = 600.
- Learning rate = 0.01.
- Minimum observations in node = 10.

The ROC metric for the tuned Stochastic Gradient Boosting algorithm is 0.8883755, which is better than the previous Random Forest approach. Sensitivity and specificity ratios are 0.9380287 and 0.5525751634, respectively. This classifier is slightly better than the Random Forest approach without class-weights. However, like it is observed before in Random Forest approach without class-weights, specificity ratio is very low. This is an indicator of a bad classifier from the standpoint of class with fewer instances.

Secondly, class-weights are used during the training phase of the Stochastic Gradient Boosting models. In this part, we have used several preprocessing approaches for the training. There are 4 different models that trained with specific preprocessing methods. These methods can be seen below:

- Excluding zero variance features.
- Excluding zero variance features and applying principle component analysis transformation.
- Excluding near zero variance features.
- Model without any preprocessing.

Same cross-validation and grid search setting were used to build all of these models. Best performing model was the model without any preprocessing. Best subset of the parameters for this model can be seen below:

- Interaction depth = 3.
- Number of trees = 650.
- Learning rate = 0.01.
- Minimum observations in node = 10.

The ROC metric for this model is 0.8873173, which is slightly worse than the previous Gradient Boosting approach. However, sensitivity and specificity of this model is 0.7995767 and 0.8099869, respectively. This means that this classifier performing better in terms of specificity. The gain in the specificity is huge compared to loss in sensitivity. Therefore, we can conclude that this final model fits better to this dataset, considering that it has a certain class-imbalance built in it.

3.3 Penalized Regression

In statistics, lasso (least absolute shrinkage and selection operator) is a regression analysis method that performs both feature selection and regularization in order to boost the prediction accuracy and interpretability of the resulting regression model.[6] Ridge regression is another regularization method in which all parameters are regularized equally. It is useful in avoiding the problem of multicollinearity in regression models.[7]

Firstly, like we did in the other algorithms, we ignored the class-imbalance of the dataset. We have conducted 10-fold cross-validation with 3 repeats. Grid search method is used to identify the alpha parameter, which is type of regularization, and lambda parameter. Grid search scope can be seen below:

- Alpha: 0, 1.
- Lambda: sequence of 10 equally spaced real numbers between [0.0001, 0.1] and sequence of 10 equally spaced real numbers between [0.15, 5].

On top of this grid search, 4 different preprocessing combinations were performed. These combinations are:

- Centering the features according to the mean and scaling them.
- Excluding near zero variance features, centering the features according to the mean, and scaling them.
- Excluding near zero variance features, centering the features according to the mean, scaling, and applying principle component analysis transformation to them.
- Excluding near zero variance features, centering the features according to the mean, scaling, and applying spatial sign transformation in order to avoid outlier problems.

As a result of these 4 different models, best parameter and preprocessing combination are:

- Centering the features according to the mean and scaling them.
- Alpha = 1.
- Lambda = 0.112.

This alpha value indicates that lasso regularization is used for the regression model. The ROC metric for this model is 0.8814882. Sensitivity and specificity ratios are 0.9412080 and 0.523830065, respectively. This classifier is worst among the three models stated until now.

Secondly, class-weights are implemented into the training phase of the model. Same grid search parameters with same preprocessing combinations are applied to find best tuned model. Best parameters and combination of preprocessing steps can be seen below:

- Centering the features according to the mean and scaling them.
- Alpha = 1.
- Lambda = 0.112.

Alpha being equal to 1 shows that lasso regularization is chosen by the grid search as a best parameter. The ROC metric for the class-weighted model is 0.8802013. This is slightly worse than non-weighted penalized regression model. Sensitivity and specificity ratios are 0.7569805 and 0.8283922, respectively. The same trade-off between sensitivity and specificity appears in this model too. Introducing class-weights is improved the distinction of class with fewer observations.

4. CONCLUSIONS AND FUTURE WORK

As a result, best classifier algorithm for this dataset is class-weighted Stochastic Gradient Boosting without any preprocessing. Tuned parameters of the algorithm are stated below:

- Interaction depth = 3.
- Number of trees = 650.
- Learning rate = 0.01.
- Minimum observations in node = 10.

With this parameter setting, performance metrics of the classifier are:

- ROC = 0.8873173.
- Sensitivity = 0.7995767.
- Specificity = 0.8099869.

In the class-imbalanced datasets, considering the class-weights as an input at the training phase is important. When we introduced the class-weights into algorithms, we have to face with a trade-off between ROC metrics of the models. However, the change in the ROC metric is very small, thus negligible. On top of that, another trade-off was between sensitivity and specificity ratios of the classifiers. Since class-weighted algorithms had so much better specificity with a dispensable loss in sensitivity, which means better classification of class with fewer observations, we selected the class-weighted version of the Stochastic Gradient Boosting model.

To have a better predictive model, newer versions of the Gradient Boosting methods can be used. XGBoost, LightGBM, and CatBoost are some examples of different versions of Gradient Boosting. Main reason to use these libraries is for fast and efficient implementation of scalable models. However, due to some changes in the algorithm, it is seen that some of them performs better than the others.[8] Another way to improve the classification is to use a different and more complex models, such as neural networks. This topic is beyond the scope of our course.

5. REFERENCES

1. en.wikipedia.org. 2021. Statistical classification. [online] Available at: https://en.wikipedia.org/wiki/Statistical_classification [Accessed 11 February 2021].

2. SUN, Y., WONG, A. K. C., & KAMEL, M. S. (2009). *CLASSIFICATION OF IMBALANCED DATA: A REVIEW. International Journal of Pattern Recognition and Artificial Intelligence, 23(04), 687.* doi:10.1142/s0218001409007326

3. SUN, Y., WONG, A. K. C., & KAMEL, M. S. (2009). *CLASSIFICATION OF IMBALANCED DATA: A REVIEW. International Journal of Pattern Recognition and Artificial Intelligence, 23(04), 688–689.* doi:10.1142/s0218001409007326

4. en.wikipedia.org. 2021. Random forest. [online] Available at: https://en.wikipedia.org/wiki/Random_forest [Accessed 11 February 2021].

5. en.wikipedia.org. 2021. Gradient boosting. [online] Available at: https://en.wikipedia.org/wiki/Gradient_boosting [Accessed 11 February 2021].

6. en.wikipedia.org. 2021. Lasso (statistics). [online] Available at: https://en.wikipedia.org/wiki/Lasso_(statistics) [Accessed 11 February 2021].

7. en.wikipedia.org. 2021. Lasso (statistics). [online] Available at: https://en.wikipedia.org/wiki/Tikhonov_regularization [Accessed 11 February 2021].

8. lightgbm.readthedocs.io. 2021. Experiments. [online] Available at: https://lightgbm.readthedocs.io/en/latest/Experiments.html [Accessed 11 February 2021]