CSC 570AG Machine Learning

# Final project – machine learning in politics

In this project, the machine learning algorithms Naïve Bayes and Decision tree (C5.0) have been implemented on a dataset. The dataset “house-votes.txt” includes votes for each of the U.S. House of Representatives Congressmen. The goal of the project is to predict whether the voter is a republican or a democrat based on their votes.

The tasks that are performed in this project are:

1. Preparing the data by imputing missing values
2. Implementation of Naïve Bayes, and Decision tree algorithms on the dataset and comparing the performance of the two algorithms
3. Implementation of 10-fold Cross Validation to estimate how well the algorithms perform
4. Perform Automated parameter tuning for both the models using “caret” package
5. Improve the performance of each algorithm by “bagging” (ensemble learning) and the caret package

Preparing the data

The dataset contained 435 observations of 17 variables. The dataset contained several missing values. To prepare the dataset for the implementation of the two algorithms, I imputed the missing values using a package called “mice” which stand for Multiple Imputation by Chained Equations. This package performs PMM (Predictive Mean Matching), which is a semi-parametric imputation approach. It helps in imputing missing values with plausible data values. These plausible values are drawn from a distribution specifically designed for each missing datapoint.

Implementation of Decision tree and Naïve Bayes

The dataset was split into train and test sets in the ratio of 80% to 20%.

On implementing the Decision tree algorithm on the dataset, an accuracy of 99% was displayed. Only one instance was incorrectly classified as democratic when it was republic.

The Naïve Bayes algorithm showed an accuracy of 91% with two instances classified as republican when they were democratic and seven instances classified as democratic when they were republican.

On analysis, it can be proved that the Decision tree algorithm performed better than the Naïve Bayes algorithm.

Implementing 10-fold Cross Validation

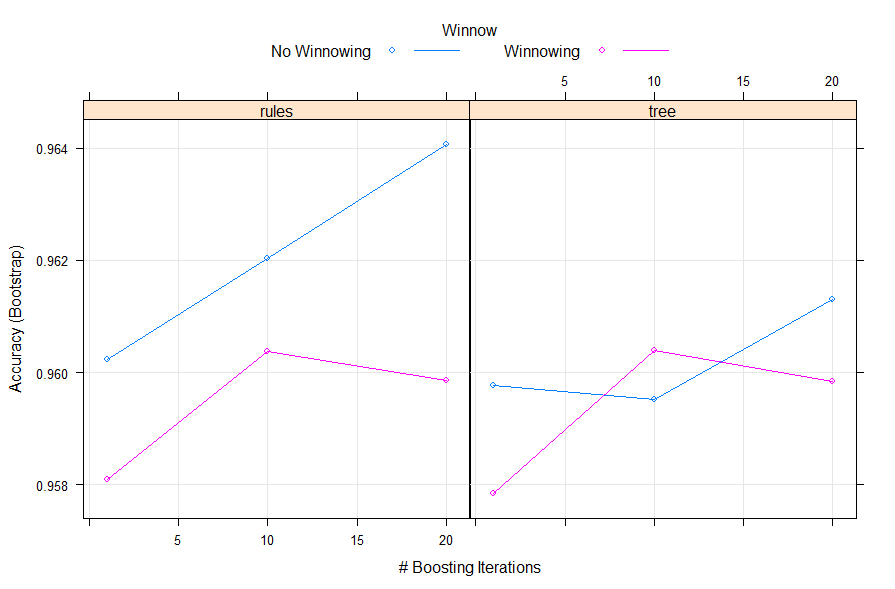
The “caret” package was used to implement 10-fold cross validation. The number of folds were set to 10. After implementation of Cross validation, the Decision tree algorithm displayed an accuracy of 90.8%. Whereas, the Naïve Bayes algorithm displayed an accuracy of 80.4%.

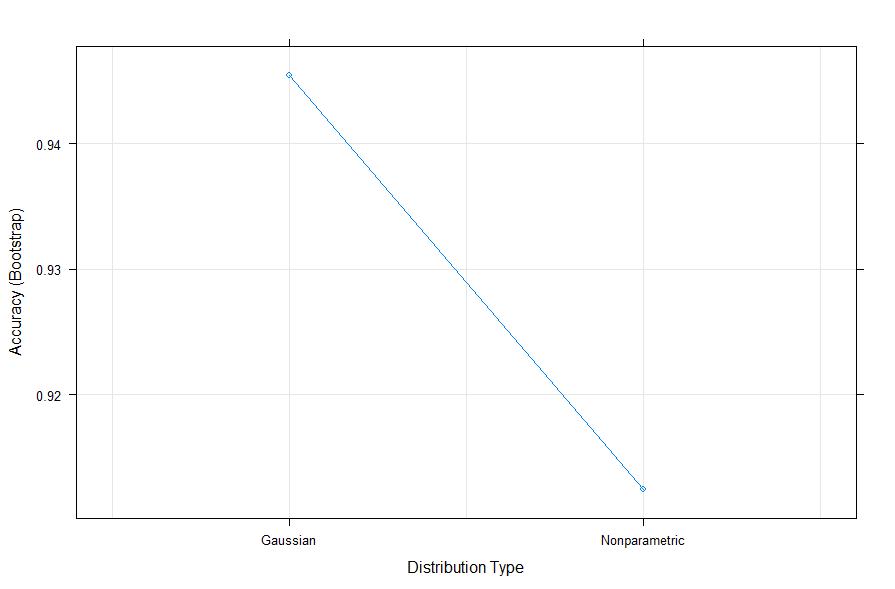
On analyzing the results, it can clearly be seen that the Decision Tree C 5.0 algorithm performs much better than Naïve Bayes algorithm.

Performing automated parameter tuning

To perform automated tuning, the train () function was used. The method parameter was set to “C5.0” and “nb” for Decision tree and Naïve Bayes algorithms respectively.

The automated parameter tuning for the decision tree algorithm displayed an accuracy of 96%, whereas the Naïve Bayes algorithm showed an output accuracy of 91.2% with kappa value of 82%. Below are the plots for both the models:





Ensemble learning (Bagging)

The method used for ensemble learning is the Bagging method.

When bagging method was implemented on decision tree C 5.0, the accuracy displayed is 96.5% with kappa value of 92.7%. When bagging method was applied to the Naïve Bayes algorithm, it returned an accuracy of 91.2% with kappa value of 81.5%.

Conclusion

On analyzing the results of Decision tree C (5.0) algorithm and Naïve Bayes algorithm on the dataset, it can be concluded that the Decision tree C(5.0) algorithm always performs better than the Naïve Bayes algorithm.