

Final Project Report: Statistical Learning (UNL STAT 983)

by Alison Kleffner, Sarah Aurit, Emily Robinson

April 12, 2021

1 Introduction

Successful marketing campaigns and productive selling strategies are directly linked to communication about key indicators of quality; hence, objective measurements of quality are essential. Within the wine industry, there are two types of quality assessment: physiochemical and sensory tests. Sensory tests require a human expert to assess the quality of wine based on visual, taste, and smell [Hu et al., 2016]. Hiring human experts to conduct sensory tests can take time and be expensive [Gupta, 2018]. In addition, taste is the least understood of all human senses [Cortez et al., 2009]. Unlike sensory tests, laboratory tests for measuring the physiochemical characteristics of wine such as acidity and alcohol content do not require a human expert. The relationship between physiochemical and sensory analysis is not well understood. Recently, research in the food industry has utilized statistical learning techniques to evaluate widely available characteristics of wine. This type of evaluation allows the automation of quality assessment processes by minimizing the need of human experts [Gupta, 2018]. These techniques also have the advantage of identifying important the phsiochemical characteristics that have an impact on the quality of wine as determined by a sensory test. In this report, we will investigate the accuracy of three classification techniques, and also address the challenges of classification with imbalanced data.

In this paper, we apply classification and resampling techniques to the “Wine Quality Data Set” found on the UCI Data Repository at <https://archive.ics.uci.edu/ml/datasets/Wine+Quality>. [UCI, Cortez et al., 2009]. This data set consists of white (4898 samples) and red (1599 samples) Portuguese “Vinho Verde” wine samples. Each wine sample was processed to obtain measures for 11 physiochemical variables: fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, and alcohol. A sensory test conducted by at least three sommeliers was used to determine a quality rating on an 11 point scale from 0 - very bad to 10 - very excellent. Following Hu et al. [2016], we separated the wine samples in to 3 classes: Low Quality (≤ 4), Normal ($5 - 7$), and High Quality (≥ 8). Basic descriptives are provided below; the Mann-Whitney and chi-square tests were used for comparisons of continuous and discrete variables, respectively.

The goal of this paper is train a model that would work well to classify wines into three categories, which are: poor quality, normal quality and high quality. It is desirable to classify wines using physicochemical properties since this does not involve human bias that would come into play with human tasters. We evaluated two classification techniques, eXtreme Boosting (XGBoost) and Random Forest. Given prior investigation of the white wine data showed the Random Forests technique performed well; we would like to test this method using both white and red wine. Previous papers on the wine data set have applied different versions of gradient boosting such as adaptive boost; we will apply XGBoost, an alternative gradient boosting technique. The quality categorization poses a challenge of working with imbalanced classes as there are many more normal quality wines than low or high quality wines (Figure 1). To address this, we will evaluate different resampling techniques in order to determine if resampling improves performance and to identify which resampling method is best. Accuracy of each classification model applied in conjunction with each resampling method was compared. Initially, we will first evaluated the classifier performance with no resampling. Since the data is very imbalanced, we hypothesized this to have low performance but provide a baseline. We will then apply a resampling method using SMOTE, which is an algorithm where the minority class (in this case low and high quality), will be oversampled. We hypothesize this method runs the risk of overfitting the mode. The final resampling method applied is a random under sampler, where majority class data are randomly removed from the data set. We hypothesize this risks losing valuable information in our model. Accuracy of each classification method and resampling technique was evaluated by the overall correct classification rate and each individual group (low quality, normal quality, and high quality) correct classification rate.

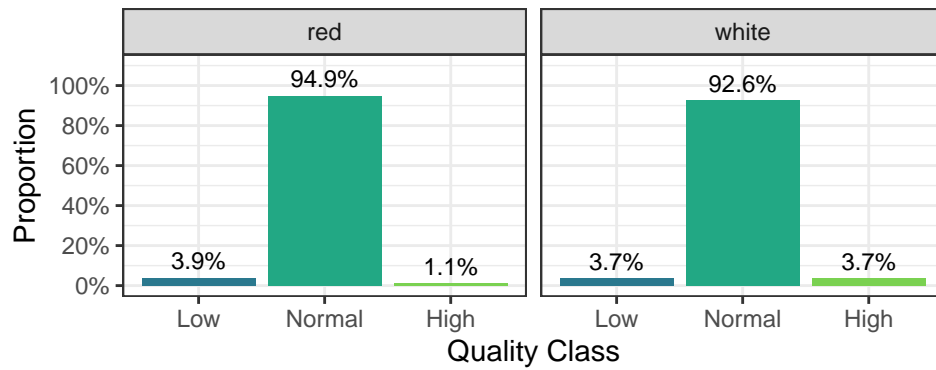


Figure 1: Wine quality class imbalance.

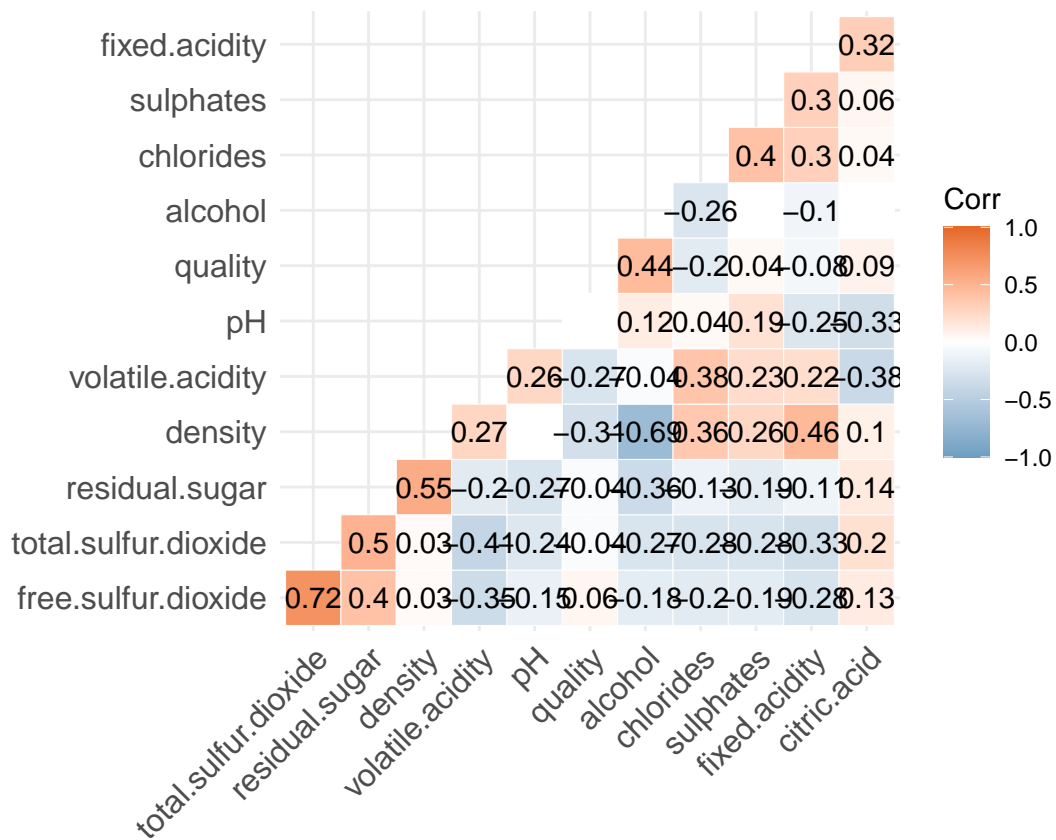


Figure 2: Correlation Plot

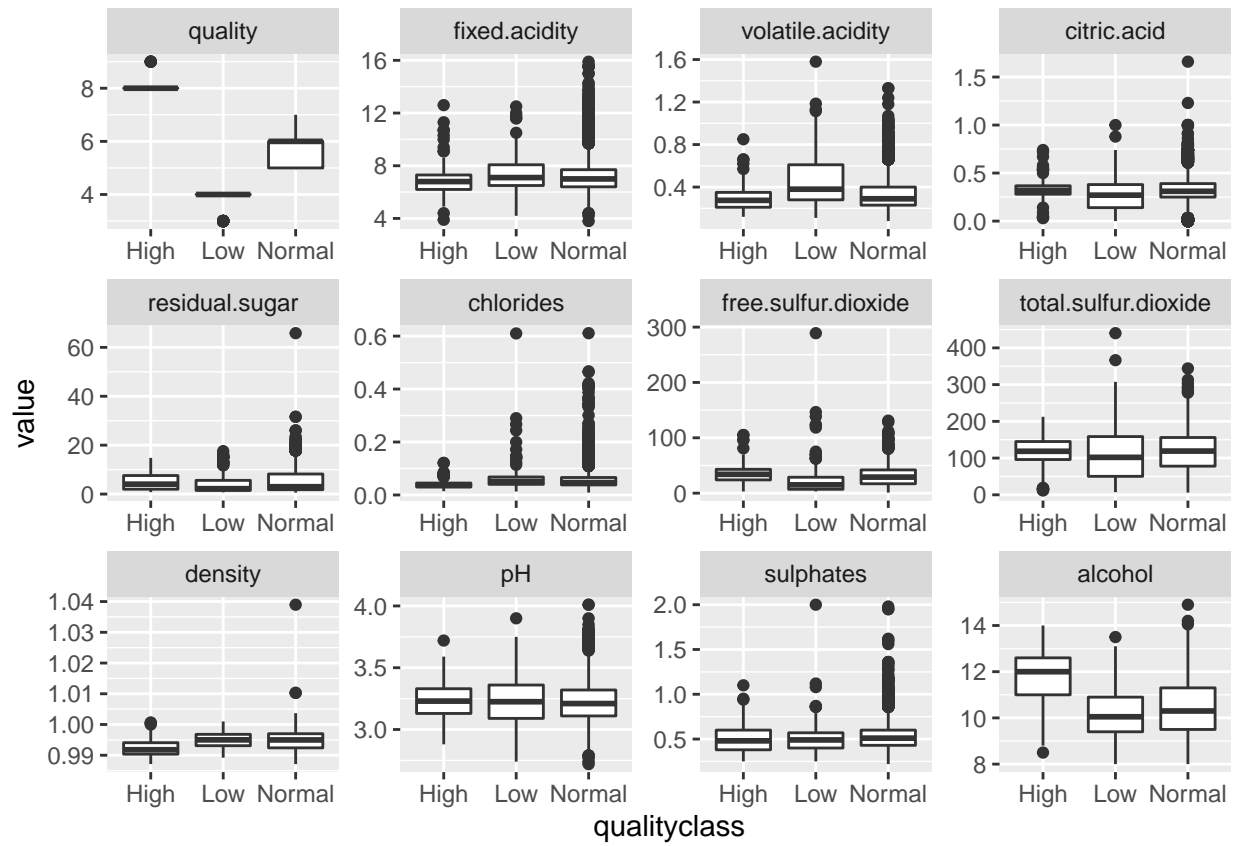


Figure 3: Distribution of the Predictor Variables

2 Methods

We applied two classification methods, XGBoost and Random Forest, on the original training set, the undersampled training set, and the oversampled training set. Accuracy of each classification method and resampling technique was evaluated by the overall correct classification rate and each individual group (low quality, normal quality, and high quality) correct classification rate. Monte Carlo Cross-Validation (MCMC) was used to select tuning parameters and evaluate model performance. The MCMC algorithm repeatedly splits the data into training and testing sets ($B = 50$) by randomly selecting the designated proportions (70% training; 30% testing) of the overall data set to each. The undersampling and oversampling techniques described below are then applied to the training set to obtain a final resampled training data set. For each split, the final resampled training data set is used to build the model and then accuracy is evaluated on the corresponding testing data set. The mean, 5th-quantile, and 95th-quantile of the correct classification rates are used to evaluate performance over the B splits.

2.1 Resampling

Before we are able to consider the different classification methods, the imbalanced issue in our data must be examined. This is because typical classifier algorithms assume a relatively balanced distribution, so with imbalanced data they tend to be biased towards the majority class [Yanminsun, 2011]. Classification rules for the minority classes tend to be undiscovered or ignored, so the minority class is misclassified more often than the majority class. If we were to look at our data set from a binary standpoint where “Low” and “High” were classified as rare, and “Normal” was classified as not rare, the majority class has 6053 observations, and the minority class has 444 observations, which is a ratio of about 13:1. So due to this imbalanced nature in the data, classifying algorithms will be biased towards classifying wines as Normal, and causing poor predictions for the Low and High classes. When creating the resampled data sets, the binary classification of “rare” and “not rare” is used due to the resampling algorithms needing a binary class.

A method on how to deal with the imbalanced issue in the data is through resampling the original data set either by oversampling the majority class, or undersampling the minority class. The goal with these methods is to create a data set that has close to a balanced class distribution, so the classifying algorithms will have better predictions, so it will predict to the minority class more accurately [Chawla, 2013]. The oversampling method involves replicating the minority class and creating synthetic data, or creating new instances using heuristics. This method tends to have an overfitting problem, where it’ll predict the minority class more than it should. The undersampling method removes instances from the majority class randomly or using some heuristics. Since data is being removed, potentially valuable information is not considered [Hu et al, 2016]. To see how valuable resampling is in our data set, we ran the classifying algorithms with just using the original imbalanced data set, randomly undersampling method, and SMOTE, which is an oversampling method.

The undersampling method that we considered is random undersampling. In this method, instances of the majority class are discarded at random until reaches balanced with the minority class [Chawla, 2013]. For example, say there are 1000 observations in the majority class, and 100 in the minority class, observations in the majority class will be randomly discarded until this class is also 100. The benefit of this method is that since the data frame is being reduced, it is less costly. However, potentially useful information is discarded, which may make the decision boundary between the majority and minority class less clear, causing a decrease in prediction performance [Chawla, 2013].

SMOTE, which stands for Synthetic Minority Over-Sampling Technique, is an oversampling technique which uses interpolation of the minority class to create synthetic data. The process begins by finding the k nearest neighbors of each observation of the minority class based on some distance measure. Then a point between the minority class observation and one of its nearest neighbors is randomly picked by first finding the difference between the observation and its nearest neighbor. This difference is then multiplied by a random number between 0 and 1. This is then added to the observation, which becomes the new synthetic data point that is then added to the data set [Chawla, 2002]. In Hu et al, the used $k=5$, so that is what we used to create our resampled data set [2016]. Oversampling tend to have an overfitting problem since now the minority class extends into the majority space, however this generally poses less of an issue with SMOTE [Luego, 2010].

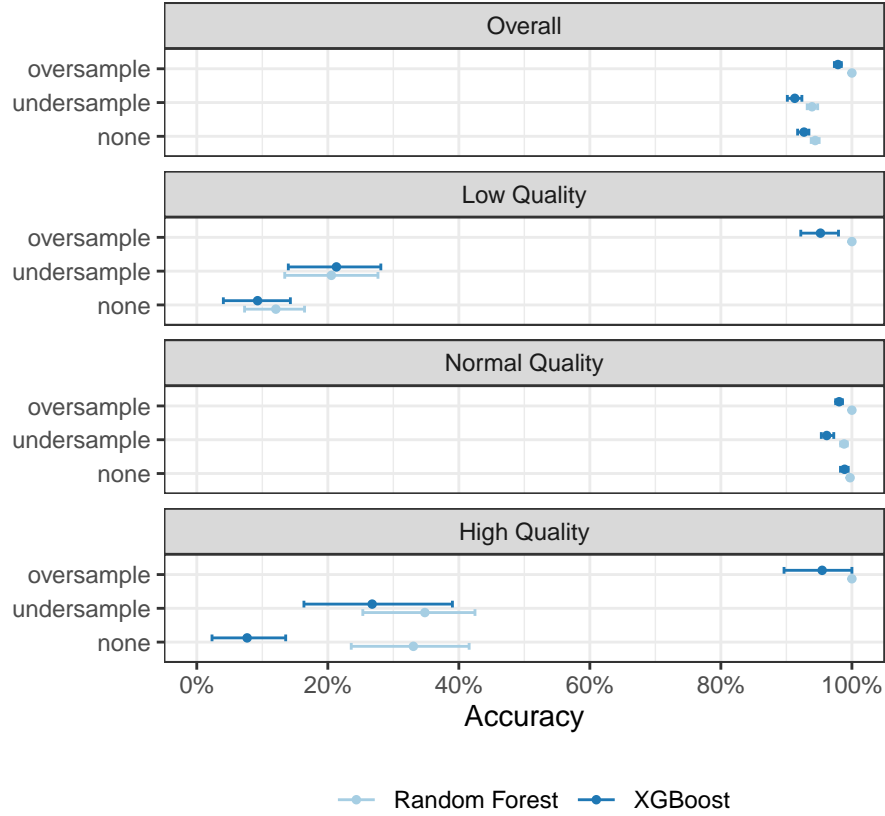


Figure 4: MCMC accuracy results for each classifier applied to each resampling technique.

2.2 Classification

The two classification methods selected are tree based ensemble methods. We apply eXtreme Gradient Boosting (XGBoost) using the `xgboost` function with the multiclass classification using the softmax objective function with three classes maximizing the multiclass misclassification error evaluation metric in the `xgboost` library [Chen et al., 2021]. This method combines the tree-based model approach by implementing recursive partitioning and the boosting algorithm which repeatedly optimizes classification methods on the training set. In repeated optimization, a weak classifier is fit on the original data set with each observation having equal weight, the weight is then calculated for the current model based on the error rate and observations are assigned new weights used to fit the next weak classifier. This process is repeated for a final boosted classifier given by the weighted sum of our weak classifiers. XGBoost allows for a variety of evaluation metrics providing a benefit over other boosting methods. Tuning parameters for maximum tree depth, step size shrinkage to prevent overfitting (η) were selected by conducting a hyperparameter grid search with MCMC.

3 Results

4 Conclusion

References

Uci machine learning repository: Wine quality data set. URL <https://archive.ics.uci.edu/ml/datasets/Wine+Quality>.

- T. Chen, T. He, M. Benesty, V. Khotilovich, Y. Tang, H. Cho, K. Chen, R. Mitchell, I. Cano, T. Zhou, M. Li, J. Xie, M. Lin, Y. Geng, and Y. Li. *xgboost: Extreme Gradient Boosting*, 2021. URL <https://CRAN.R-project.org/package=xgboost>. R package version 1.3.2.1.
- P. Cortez, J. Teixeira, A. Cerdeira, F. Almeida, T. Matos, and J. Reis. Using data mining for wine quality assessment. In *International Conference on Discovery Science*, pages 66–79. Springer, 2009.
- Y. Gupta. Selection of important features and predicting wine quality using machine learning techniques. *Procedia Computer Science*, 125:305–312, 2018.
- G. Hu, T. Xi, F. Mohammed, and H. Miao. Classification of wine quality with imbalanced data. In *2016 IEEE International Conference on Industrial Technology (ICIT)*, pages 1712–1217. IEEE, 2016.

A Code

Need to work on “Environment shaded undefined.”