

Machine Learning Methods to Predict Wine Quality

Joseph Despres

April 13, 2022

Contents

Introduction	1
Problem Description	1
Data	1
Methodology	2
Results	4
Conclusions	4
References	5
Appendix	5

Introduction

Anyone looking to purchase a bottle of wine will be facing a search space so large that only possible for a computer to search through. Wine varies significantly from region to region. Therefore, this is an ideal problem for machine learning to handle. In particular, this project aims to specify a function that maps the chemical properties of various wines to the judge's quality scores. This report begins with a problem description followed by a description of the data. I use the Sklearn package (Scikit-learn) in the Python Programming Language (Van Rossum & Drake, 2009). After that, I devote several paragraphs to comparing different feature scaling strategies against logistic regression because there is a significant difference in runtime and performance. After that, I discuss the modeling techniques, elastic net regression, XGBoost, and Support Vector Machines in an ensemble to reach a training set prediction accuracy of 69

Problem Description

This project aims to discover the relationship between wine quality and chemical properties. The motivation for this project is that one purchasing wine seriously must consider different regions, grapes, growing climate, seasons, storage time, and substantial variations in quality and price. This is a problem well suited to Machine Learning because of the vast and complex search space. The benefits of an accurate predictive wine model are that one could select excellent wine at a low price, this lowers the risk of purchasing low-quality wine, or even determining undervalued wine at auctions.

Data

To determine the quality score based on the distinct chemical properties of different wines, we will use a dataset hosted by the University of California Irvine's Machine Learning repository (Lichman, 2013). These data are from a study conducted by the Portuguese

government studying a data mining approach to predicting human wine taste (Cortez, Cerdeira, Almeida, Matos, & Reis, 2009). Using support vector machines, they achieve an out-of-sample accuracy of 33

This is a structured, curated, tabular dataset conveniently in first normal form. The outcome we wish to predict is the quality as rated by a professional wine judge. One thing that was not obvious from the UCI repository is the fact that there are duplicated rows, which means multiple judges rated the same wine as the same quality. Judges are anonymous in these data, therefore there are no features indicating which judge is scoring. I will come back to this point after discussing model specifications. There are 11 additional features, which are listed in Table 1.

Table 1: Descriptive Statistics

Feature	Mean	Std
Fixed Acidity	7.215	1.296
Volatile Acidity	0.340	0.165
Citric Acid	0.319	0.145
Residual Sugar	5.443	4.758
Chlorides	0.056	0.035
Free Sulfur Dioxide	30.525	17.749
Total Sulfur Dioxide	115.745	56.522
Density	0.995	0.003
Ph	3.219	0.161
Sulphates	0.531	0.149
Alcohol	10.492	1.193
Quality	5.818	0.873
Is Red	0.246	0.431

Note: 6497 observations

the target is wine quality as scored by a professional judge. Wine quality is an ordinal categorical variable. The class assignments are significantly unbalanced. However, the nature of this problem is that very few wines will be of very high quality and many will be average. See Figure 2 for a plot of the class distributions. Note that only 5 of 6,497 of these wines are rated 9/10. This may

necessitate some re-sampling methods because the most valuable model will be able to detect high-quality wine. We have no examples of a 10, a 1, or a 2. At this point, I will split the data and stash 30% of the data to reserve for testing and I will not touch those data until it is time to report a final result.

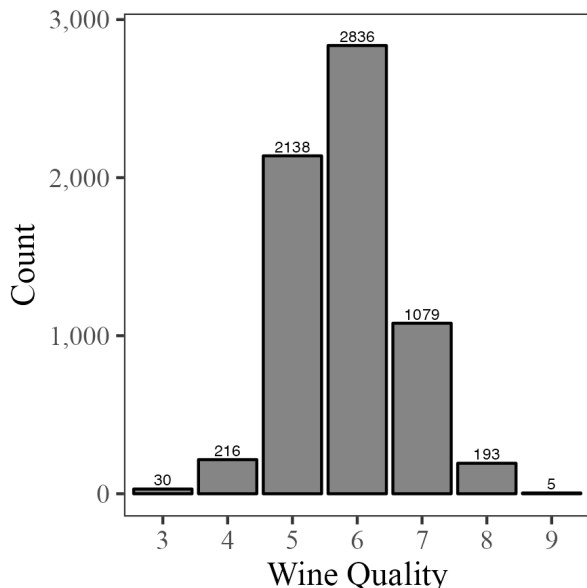


Figure 1: Comparing Resampled Prediction Accuracy by Transformation

Methodology

After some difficulties with fitting initial models and performance plateaus associated with the Min-max Scaling method. I decided to test and document the results of using several different scaling strategies. I decided to run 2,500 different partitions of training and validation data with a 50% split from the training data. To investigate the effects of different scaling schemes. There was a significant difference in results as well as runtime. This is due to the design matrices having difficulties inverting. The runtimes will differ based on the model, software implementation, hardware, and datasets so I don't

list them.

This paper will consider two performance metrics. The first accuracy is measured as the percentage of correct predictions. Note, this is an unbalanced assignment problem, accuracy should not be compared to the number of classes but rather the frequency of the most frequent class. The most frequent class is a quality score of 6. Which occurs in about 43% of the observations. This metric is intuitive but does not capture our true intentions. I will also use a balance between precision and recall known as AUC. AUC is simply the area under the ROC curve.

Table 2: Demo table

Scaling Method	Mean	Std	Median
No Transformation	0.590	0.024	0.592
Dropped Outliers	0.645	0.026	0.641
Min Max Scale	0.690	0.018	0.691
Rescaled to Uniform	0.707	0.015	0.707
Standard Scale	0.726	0.017	0.725
Yao Johnson Transform	0.733	0.015	0.732

The Yao-Johnson outperforms the Standard scaling methods, however, not by much and is a far more complicated transformation process. It is a Box-Cox transformation method modified to take values less than 0. The Box-Cox Method is for linearizing vectors by some exponent that will convert the variable to a linear scale. I do not see any indication that wine's chemical properties are anything other than linear in practice. My reasoning for this is that one does not add an exponentially increasing or decreasing amount of sugar for instance. Although it has slightly better accuracy and lower standard deviation, both are desirable properties, the increase in performance did not justify the additional complexity and a deviation from what I felt is reasonable. Therefore, I will use the standard scaler.

After studying the features and exploratory plotting, we are now ready to specify the models. The plan is to use an ensemble of models to predict wine quality. The idea is that a collection of visiting

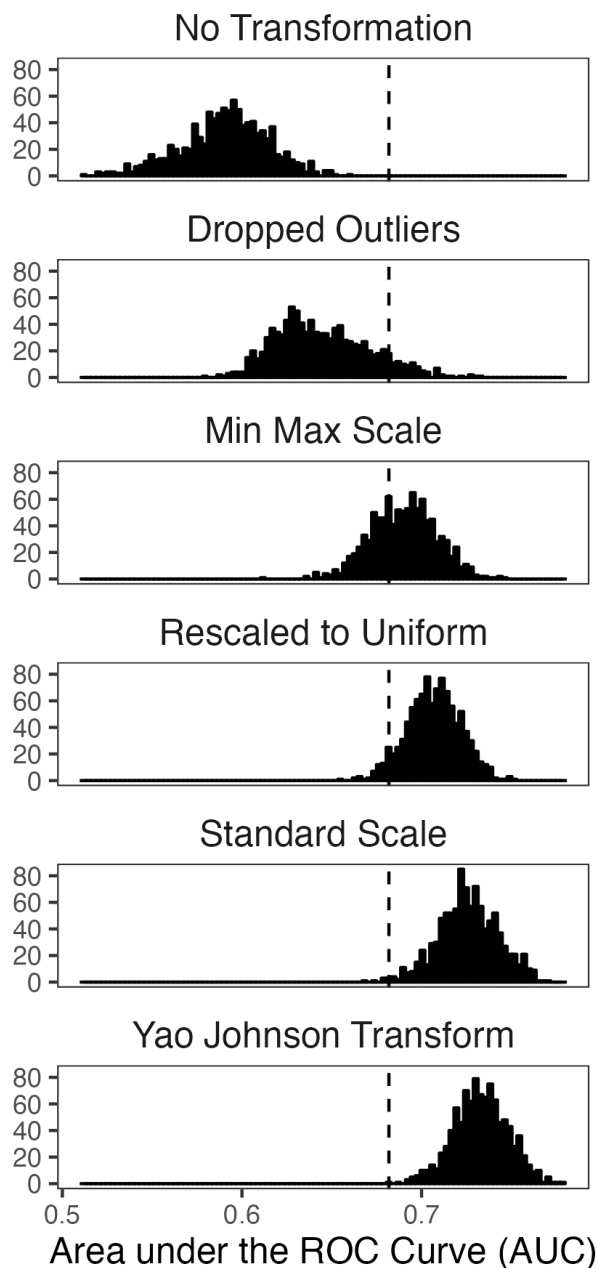


Figure 2: Comparing Resampled Prediction Accuracy by Transformation

classifiers will be more accurate than a single one. The reasoning is that one model could be wrong in many ways however it would be correct in only one. Add a collection together and they will be collected right more than wrong.

The first model is the multinomial logistic regression used during the feature scaling trials. This model is based on estimating the cumulative log odds of belonging to a given category based on the features. I add some regularization parameters L^1 and L^2 norms, commonly referred to as elastic net regression. A grid-search over a five-fold cross-validation shows that an optimal $L^1 = 0.0315$ and $L^2 = 0.013$. This achieves a training set accuracy of 54.53% and an AUC of 0.775. These are not impressive results, so we will fit a Support Vector Machines.

A support vector machine model aims to find the maximum margin hyperplane between features that separate points. I use the radial basis function kernel and tune only the regularization parameter squared L^2 norm. After cross-validating I find the best regularization parameter to be 6.58. This reaches an in-sample accuracy of 60% and an AUC of 0.801. A slight improvement.

After that, I use XGBoost (Chen & Guestrin, 2016) an efficient implementation of gradient boosted trees. There are many parameters to search through, however, I will only tune max tree depth, column sample by tree, and learning rate. After tuning, I found the column sample to be 50% to be the best a learning rate of about 0.034 and a max depth of 13. This yields an astonishing sample accuracy of 99.5% and an AUC of 0.998. This is a substantial improvement over the past two models. From here, I will wrap all three in an ensemble and test the accuracy. Given these metrics, it may be better to not even ensemble and just use XGBoost.

Then I save these models and wrap them into one ensemble model. Then fit that to the training data and compare different approaches. I want to re-engineer some features and report some results.

Results

The XGBoost model wildly outperforms the linear I found it has an accuracy of the final testing set. The ensemble. The model predictions are in Figure. This shows that the model could be used to

Conclusions

The main takeaway of this study

References

- Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785–794). New York, NY, USA: ACM. Retrieved from <http://doi.acm.org/10.1145/2939672.2939785> doi: 10.1145/2939672.2939785
- Cortez, P., Cerdeira, A., Almeida, F., Matos, T., & Reis, J. (2009). Modeling wine preferences by data mining from physicochemical properties. *Decision Support Systems*, 47(4), 547-553. Retrieved from <https://www.sciencedirect.com/science/article/pii/S0167923609001377> (Smart Business Networks: Concepts and Empirical Evidence) doi: <https://doi.org/10.1016/j.dss.2009.05.016>
- Lichman, M. (2013). *UCI machine learning repository*. Retrieved from <http://archive.ics.uci.edu/ml>
- Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... others (2011). Scikit-learn: Machine learning in python. *Journal of machine learning research*, 12(Oct), 2825–2830.
- Van Rossum, G., & Drake, F. L. (2009). *Python 3 reference manual*. Scotts Valley, CA: CreateSpace.

Appendix

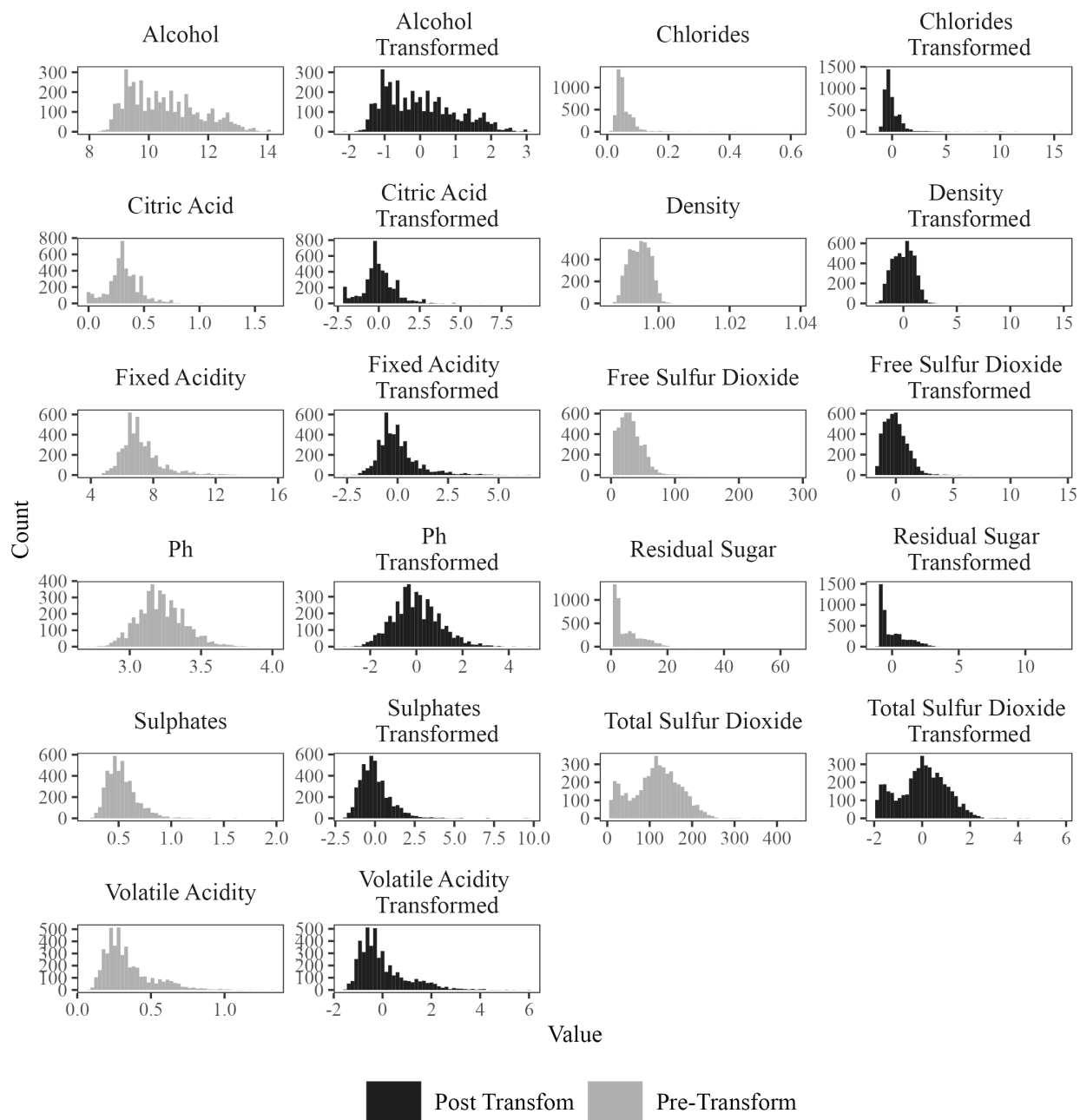


Figure 3: Comparing Re-sampled Prediction Accuracy by Transformation