Prediction of Wine Quality Using Machine Learning

Anubhav Roy Bhattacharya and **Chuxuan (Sophie) Yang**
{anroybhattacharya,soyang}@davidson.edu
Davidson College
Davidson, NC 28035
U.S.A.

Abstract

This project focused on using machine learning to predict the quality of red and white wine using existing datasets. In this project, we tested five regression models: linear regression, stochastic gradient descent (SGD), ridge regression (using the L-2 norm), LASSO (using the L-1 norm), and SGDClassifier. After conducting a series of experiments using these models, we ranked their performances based on the results of wine quality prediction. We found that `SGDClassifier` performed the best, but among the models discussed in class, ridge regression performed the best. This was true for both red and white wine. The best models used a combination of grid search with cross-validation and second degree polynomials. We think this project can be used to support wine marketing and quality control efforts, and can help businesses better understand consumer needs.

1 Introduction

Wine is one of the most popular alcoholic beverages in the world, and the wine industry is a multi-billion dollar industry that boosts the economy of a number of US states. With the thousands of varieties of wine on the market currently, an important question arises: Which one should you buy? Wine experts can of course rate the quality of a wine, but the average consumer does not always have easy access to such expertise. In this paper, we attempt to predict the quality of a wine given a number of its physicochemical properties. A machine learning algorithm that can accurately predict the quality of a wine given some of its features could be instrumental in bringing the capability to evaluate the quality of wine to consumers that don't have any special training.

Our dataset consists of two files, `winequality-red.csv` and `winequality-white.csv` (Cortez and Reis 2009) , containing red wine and white wine data respectively. The red wine dataset includes information about 1599 wines, and the white wine dataset includes information about 4898 wines. The information included in both datasets is the same. A quality column holds information about the quality of wine as rated by wine experts with scores between 0(very bad) and 10(very excellent). This quality data is what we are interested in predicting, and will henceforth be referred to as the target variable. The datasets also include columns that detail information regarding the fixed acidity, volatile acidity, citric acid, residual sugar, chlorides, free sulfur dioxide, total sulfur dioxide, density, pH, sulphates, and alcohol content of each wine. These 11 variables make up the set of features that will be used to predict the target or the quality. The features are all in floating point format while the target is in integer format.

Next, we will provide a brief explanation of the models we used while solving this problem as well as an overview of the preprocessing that the data goes through. Following that, we will conclude by summarizing the performance of our models, and comparing and analyzing the results.

2 Background

Given that this is our first project for this class, we decided to stick mostly to models that had been mentioned in class. As such, we tried out the linear regression, stochastic gradient descent (SGD), ridge regression, and LASSO models from the Python `scikit-learn` library. We also tried a stochastic gradient descent classification model, `SGDClassifier`, from the same library as well.

- **Linear regression** - A straightforward model that focuses on finding the weights associated with the least cost in a linear combination of the features of the dataset.
- **SGD** - Similar to linear regression, except the entire set of training examples is not evaluated each time a decision about minimizing the cost function is made. Instead, a random training example is chosen, and the minimum point of the cost function is calculated using that randomly chosen training example.
- **Ridge regression** - A version of linear regression that employs an L-2 regularization strategy. Ridge regression regulates the complexity of the curve used to fit the data by employing a penalty for higher weights. The specific penalty used is called the L-2 penalty and is defined as

$$\sum_{j=1}^{n} \theta_j^2$$

.
- **LASSO** - Similar to ridge regression, except the regular-

ization penalty used is L-1, which is defined as

$$\sum_{j=1}^{n} |\theta_j|$$

.

- **SGDClassifier** - While this model was not talked about in class, it is similar to the regular SGD model in that gradient descent updates are made using single or batches of training examples instead of the whole dataset. However, a major difference is that `SGDClassifier` is used for classification problems. A classification problem is any problem where the target variable can only take on a finite set of discrete values. Since our target variable, quality, can only take on values from 0 to 10 (inclusive), this problem can also be considered as a classification problem. `SGDClassifier` also makes use of a regularization penalty.

Before any of these models could be run, however, it was necessary that the data went through some preprocessing. We used the `pandas` Python library to import the data from the files and store them with the appropriate column headers. We used one file to house all the functions needed to preprocess the data, which we called `winequality.py`. This file held the following functions: `read_file`, `select_features`, `make_poly`, `standardize`, `drop_outliers`, `split`, `get_XY`, and `get_preprocessed_dataset`.

1. The `read_file` function was used to read in the .csv file and store it in a DataFrame object.
2. Then, we split the file into the features and the quality, since we only wanted to apply some preprocessing functions to the features dataset.
3. The `select_features` function was used to trim down the dataset to only the features we wanted to train the models with. The way we decided which features to use will be explained shortly.
4. The `make_poly` function used `PolynomialFeatures` imported from the `scikit-learn` library. We used `PolynomialFeatures` to add extra columns to the dataset to simulate higher-order polynomials.
5. The `standardize` function was then used to standardize the dataset using StandardScaler from the `scikit-learn` library.
6. The standardized DataFrame had the quality variable added back in before we dropped the outliers from the dataset. The `drop_outliers` function dropped any outliers that weren't within 3 standard deviations of the mean of any column.
7. The `split` function split the dataset into a 60%-20%-20% split to represent the training, validation, and testing datasets.

The `get_XY` function accepted a DataFrame as input and returned the features and the target separately. Finally, `get_preprocessed_dataset` was a master function that called all of the other functions and returned the preprocessed dataset.

To begin the process of feature selection, we ran some statistical analyses on our datasets. First, we created graphs of quality scores vs. each feature to see if there were any outliers. A few of these graphs can be seen in Figures 1, 2, and 3. All feature data came from the original dataset, prior to standardization, which is why the graphs have different scales. A few of the graphs showed some data points that were considerably different from any others. For example, the citric acid value for entry 152 of the red wine dataset was 3.7 standard deviations above the mean for citric acid. Even a single outlier can have a significant effect on the relationship between two variables (Wilcox 2001). Keeping this in mind, we decided to drop all outliers from our datasets.

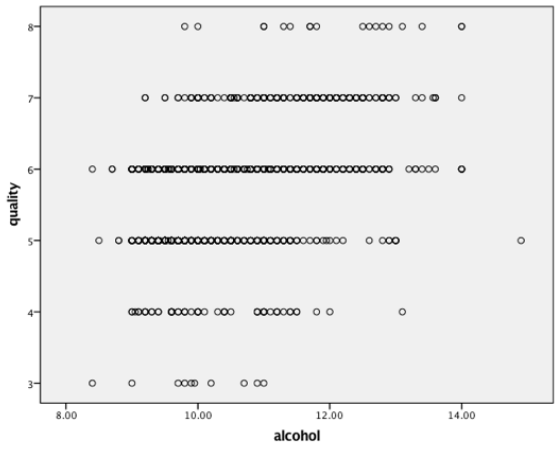

Figure 1: Correlation Between Alcohol and Quality for Red Wine

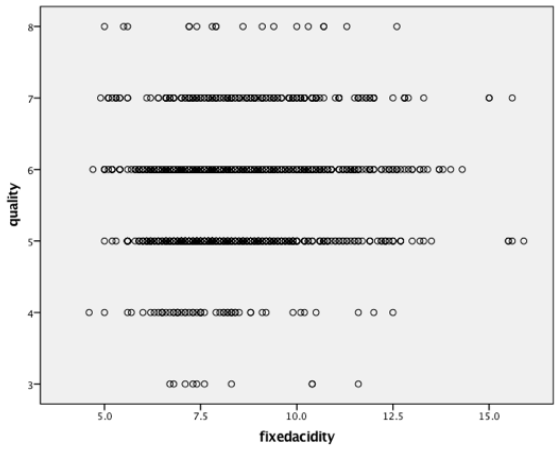

Figure 2: Correlation Between Fixed Acidity and Quality for Red Wine

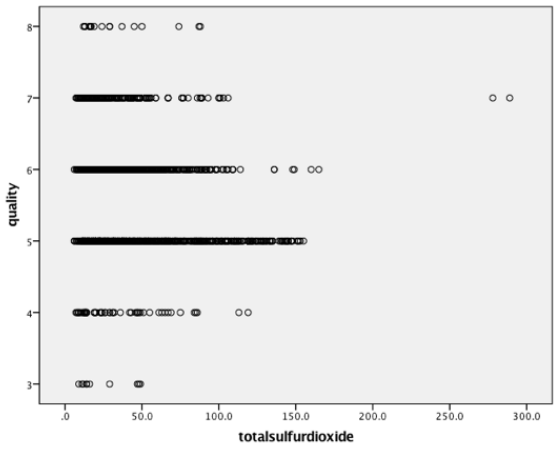

Figure 3: Correlation Between Total Sulfur Dioxide and Quality for Red Wine

Next, we computed a Pearson's correlation coefficient between each feature and quality for each dataset. Figures 4 and 5 contain the correlation coefficients we computed, ordered from the strongest to the weakest. We discovered that pH, free sulfur dioxide, and residual sugar showed the weakest correlations with quality for the red wine dataset. Sulphates, citric acid, and free sulfur dioxide showed the weakest correlations with quality for the white wine dataset. Given these results, we ran some experiments excluding some of these features that were weakly correlated with quality. A summary of our experiments and results is provided in the next two sections.

Feature	Correlation with Quality
Alcohol	0.476**
Volatile Acidity	−0.391**
Sulphates	0.251**
Citric Acid	0.226**
Total Sulfur Dioxide	−0.185**
Density	−0.175**
Chlorides	−0.129**
Fixed Acidity	0.124**
pH	−0.058*
Free Sulfur Dioxide	0.051*
Residual Sugar	0.014

Figure 4: Table of Feature Correlation with Quality for Red Wine.
** indicates the correlation is significant at the 0.01 level, while * indicates the correlation is significant at the 0.05 level.

Feature	Correlation with Quality
Alcohol	0.436**
Density	−0.307**
Chlorides	−0.210**
Volatile Acidity	−0.195**
Total Sulfur Dioxide	−0.175**
Fixed Acidity	−0.114**
pH	0.099**
Residual Sugar	−0.098**
Sulphates	0.054**
Citric acid	−0.009
Free Sulfur Dioxide	0.008

Figure 5: Table of Feature Correlation with Quality for White Wine
** indicates the correlation is significant at the 0.01 level, while * indicates the correlation is significant at the 0.05 level.

3 Experiments

The steps we followed while conducting our experiments are detailed below:

1. All of our models were housed in separate files. Each model imported their respective model files from the `scikit-learn` library, and the `get_preprocessed_dataset` and `get_XY` functions from `winequality.py`. Some models made use of other functions that will be detailed later in this section. All models were run using a `run_model` function that we wrote to fetch the preprocessed dataset, train the model, and report the R^2 score for the model using the model's `score` function. All models were run 50 times and the final R^2 scores reported in this paper are the average of 50 runs.
2. Initially, we ran all models using their default settings, and also using all the features in the dataset. The results we got served as our baseline results, so any time we modified our models, we could see if they did better or worse. After the initial step, we decided to add the `drop_outliers` function to the preprocessing step. Based on the graph of correlations between feature variables and quality scores, we observed a few outlier feature scores that were more than 3 standard deviations away from the mean. The performance scores for all models improved in a small but consistent manner after these outliers were dropped.
3. Next, we decided to tune hyperparameters to increase the performance of our models. We used `GridSearchCV` from the `scikit-learn` library to do this. `GridSearchCV` not only uses grid search to try different combinations of hyperparameters, but it also uses cross-validation when training and testing the models. For our SGD model, we varied **alpha**, a constant that multiplies the regularization term, over [0.00001, 0.00002, 0.00005, 0.0001]; we then varied **eta0**, the learning rate of the SGD model, over [0.001, 0.002, 0.005, 0.01]. For our ridge regression model, we varied **alpha** over [0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1.0, 2.0, 5.0, 10.0]. For the LASSO

model, we varied **alpha** over [0.01, 0.02, 0.05, 0.1, 0.2, 0.5, 1.0]. Finally, for the SGD classifier model, we varied **eta0** over [0.001, 0.002, 0.005, 0.01], and **alpha** over [0.00001, 0.00002, 0.00005, 0.0001]. `GridSearchCV` resulted in increased scores for all the models.

4. We were curious whether increasing the complexity of the model would improve the scores, so we decided to adopt the `PolynomialFeatures` function from the `scikit-learn` library. This feature generates new feature columns consisting of all polynomial combinations of the features with degree less than or equal to the specified degree. After trying to increase degrees to 2,3,and 4, we observed that a polynomial degree of 2 gave the best performance for all models, and scores started to deteriorate for models with degrees higher than 2.
5. Finally, we used feature selection/elimination in an effort to improve the performance of our models. Using the dataset, we generated a series of analyses using SPSSStatistics[1] on correlations between individual feature variables and quality scores. We then ranked correlations from highest to lowest for both red and white wine, and picked the top 8 features with the highest correlations as our default selected features. However, it turned out that feature selection didn't have a significant effect on model performance, so we decided to use all the features while training our models.

   Our results will be discussed in the next section.

4 Results

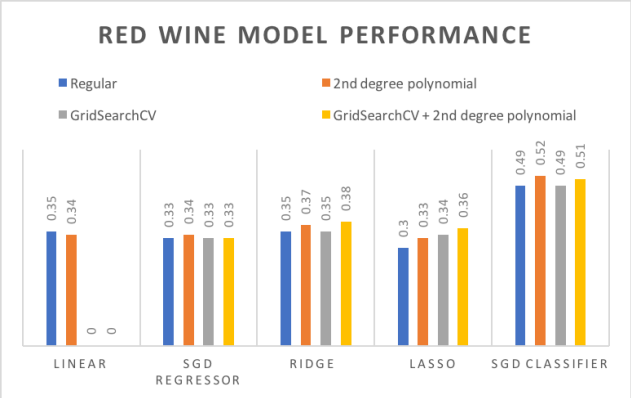

Figure 6: Visualization of red wine model performance at various experimental stages.

Our SGD Classifier model showed the best overall performance, for both red and white wines. However, since the SGD Classifier model wasn't explicitly discussed in class, we will focus more on the other models. Out of the other models, ridge regression proved to be the best model for both red and white wines. Running `GridSearchCV` improved the performance of all our models, except for SGDClassifier. Using a second degree polynomial, on the other hand, was better all across the board, and increased the performance of all our models. A summary of our results can be seen in Figures 6 and 7.

[1] A software we used to perform statistical analysis

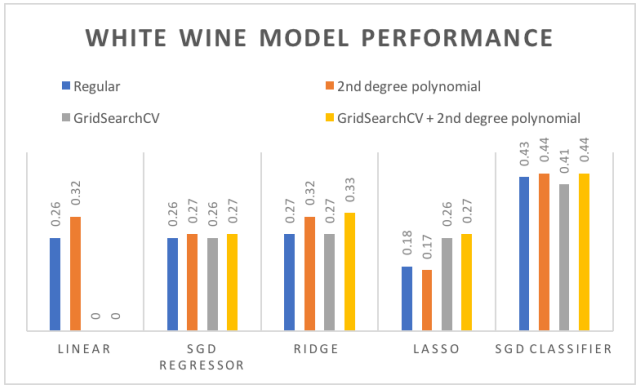

Figure 7: Visualization of white wine model performance at various experimental stages.

Also, a quick note about feature selection. The reason we ended up using all the features is that for all five models, there were no significant improvements after we eliminated the least correlated two features. In addition, when we tried to eliminate more features, we noticed decreases in performance scores for all models. Thus, we decided that feature selection was not going to result in a significant performance improvement, and decided to use all the features.

Another interesting trend from our results is that all the models performed better for red wine than white wine. This isn't really what we expected, since we thought the large number of training examples in the white wine dataset would lead to a better model. A reason for this sub-par performance may have been because of an inherent disconnect between the features and the target for the white wine dataset. However, the main take-away from our experiments is that while implementing grid search helps in almost all cases, using a second degree polynomial helps even more. In most cases, using a second degree polynomial alone was roughly equivalent to using both GridSearchCV and a second degree polynomial.

5 Conclusions

In this paper, we set out to build a model that could accurately predict the quality of wine from 11 of its physicochemical properties. After conceptualizing, building, and training our models, we discovered that `SGDClassifer` came away with the best results, for both red and white wine. From the models that we discussed in class, the ridge regression model proved to be the best, for both red and white wine again. Our most important discovery is that using second order polynomials improved the performance of our models by a fair amount. Perhaps using second order polynomials along with some other manipulation of

hyperparameters is required for this problem.

Although the performance of our models wasn't exactly groundbreaking, we think this is a valuable step towards the right direction. Future studies can perhaps look more deeply into feature selection as a way of improving the performance of the models. Also, we were limited by our knowledge to models we had covered in class. Perhaps other models would perform better than ours. So, even though some progress had been made, there's still time left before wine experts are replaced by computers.

6 Contributions

A.R.B. and C.Y. worked on the whole project together. All parts were completed with both partners present. A.R.B. and C.Y. both contributed to writing code for preprocessing the data and setting up each of the models. The contributions to the paper were also split equally, with both A.R.B. and C.Y. working on the figures and the individual sections. In the end, both authors proof-read the paper as well.

7 Acknowledgements

We would like to thank Dr. Ramanujan for his patient instructions at the beginning of our project, as well as our friends and classmates for their generous encouragement and support.

References

Cortez, P., C. A. A. F. M. T., and Reis, J. 2009. Modeling wine preferences by data mining from physicochemical properties. *Elsevier* 47(4):547–553.

Wilcox, R. R. 2001. Modern insights about pearson's correlation and least squares regression. *International Journal of Selection and Assessment* 9(2):195–205.