This report was written as part of the requirements for SENG 474 / CSC578D: Data Mining, offered at the University of Victoria, taught by Dr. Alona Fyshe.

Final Report

SENG474: Wine Analysis

Noah Spriggs, Murray Dunne, Chris Life, Greg Richardson, Haoyan Xu

I. Problem Description

"Is wine tasting pseudoscience?" If you search Google for a question like this, most likely you will get positive answers from the online articles Google finds. Most of these articles list arguments based on the inconsistency and biases of studies conducted on wine tasting. However, some of the studies they reference may be twisted towards their arguments.

In article "Wine tasting is bullshit. Here is why.", by Robble Gonzalez [1], a tasting study by Frédéric Brochet [2] was referenced. In the study, 54 people were requested to taste a glass of white wine and a glass of identical white wine except it was colored into red with tasteless dye. None of the 54 people were able to tell the red color wine was actually white wine. In the online article, Robble Gonzalez referred the 54 people as "expert wine critics", and used it to argue that expert wine critics were not able to tell the difference between red wine and white wine by taste. The same experiment was described in details in another paper [3] of Frédéric Brochet, where the 54 test subjects were described as undergraduate students, who might not have much experience in wine tasting.

Online articles are different from academic research papers. They cannot be fully trusted, however there are still many valid research papers showing strong evidence that wine tasting is pseudoscience. If data mining algorithms can be applied to make predictions with lower errors than random prediction or dummy classifier, some consistency could be added to wine tasting.

It is difficult to determine an empirical relation between the subjective quality of a wine and its chemical composition. Wine makers want to know what they can do to their processes to optimize the quality of their wine. While attempts have been made to build classifiers for wine from chemical data, not all algorithms have been tested.

In addition to testing new algorithms and variations of algorithms for prediction purposes, we are also interested in analyzing the data itself. For example, do some ingredients have a stronger impact on the perceived wine quality that others? Should winemakers be focusing more on certain ingredients than others?

II. Related Work

Cortez et al. [4] produced the dataset we are using. They have used it to train a three different classifiers using, an artificial neural network, a support vector machine, and multiple regression. They concluded that a support vector machine was best suited to classifying this data, and provide the input importances for each attribute. They did not use a Naive Bayes classifier.

Bednarova et al. [5] examined a dataset of 131 Slovenian red wines based on chemical content to predict quality (among other attributes). Their data set has several attributes in common with ours, including volatile and non-volatile acidity, density, pH, free sulfur dioxide, and sugars. Several artificial neural networks were trained on the data and tasked with predicting the sensorial quality of a wine based on its chemical attributes much like Cortez et al.

Both of these studies neglect to test classification algorithms on their wine data. Our approach is to fully investigate and analyze classification algorithms (for reasons discussed in Approach), and compare them with regressors. Additionally, we want to investigate whether certain ingredients have a greater impact on perceived wine quality than others as this information would be useful to wine makers.

III. Data Description.

The dataset [6] contains chemical descriptions of 6499 Portuguese "Vinho Verde" wines. There are 4899 entries for white wine, and 1600 entries for red. For each wine, the datasets includes the following attributes:

1. Fixed acidity
2. Volatile acidity
3. Citric acid
4. Residual sugar
5. Chlorides
6. Free sulfur dioxide
7. Total sulfur dioxide
8. Density
9. pH
10. Sulphates
11. Alcohol
12. Quality

The source of the data is UCI Machine Learning Repository [6], where the dataset is provided by Paulo Cortez, from the University of Minho.

To aid our design decisions and analysis in the project, we performed a distribution analysis on each of the white wine attributes. This information is important as it will determine whether or not certain algorithms are appropriate for the dataset.

The graphs below show the frequency distribution for each attribute.

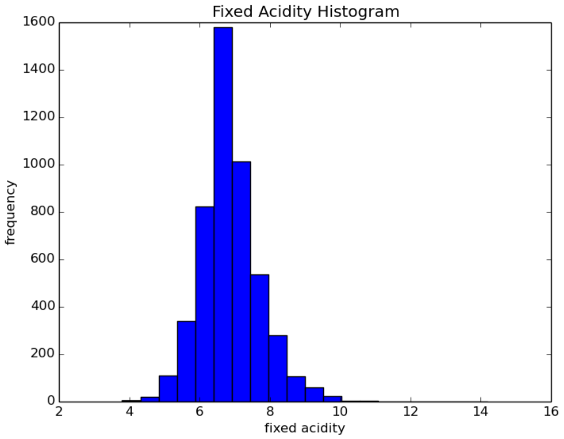


Figure 1.1 - Fixed acidity histogram

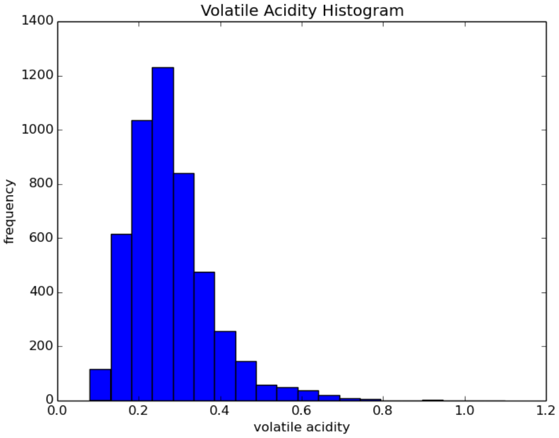


Figure 1.2 - Volatile acidity histogram

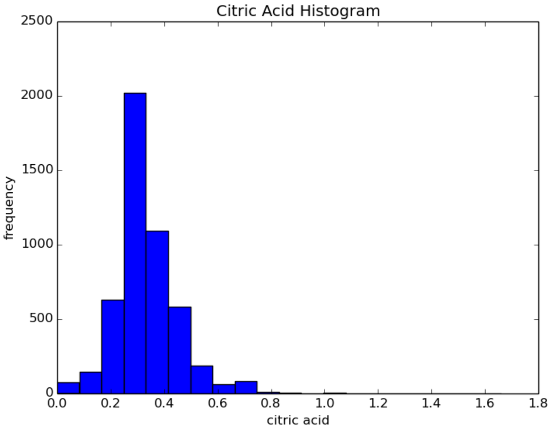


Figure 1.3 - Citric acidity histogram

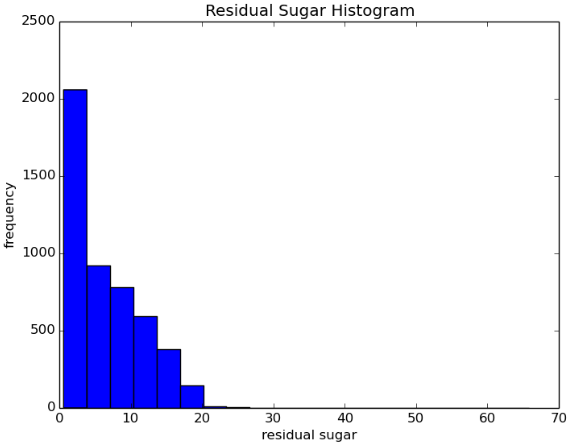


Figure 1.4 - Residual sugar histogram

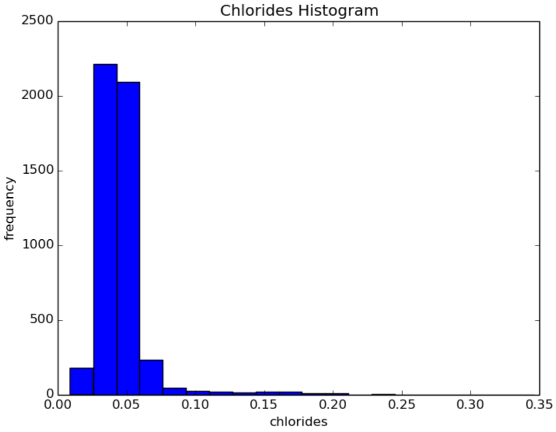


Figure 1.5 - Chlorides histogram

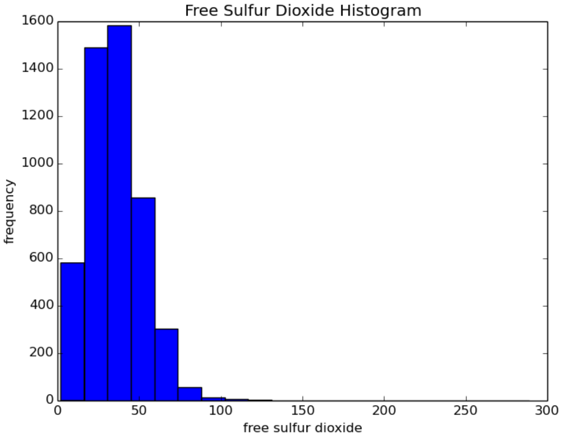


Figure 1.6 - Free sulfur dioxide histogram

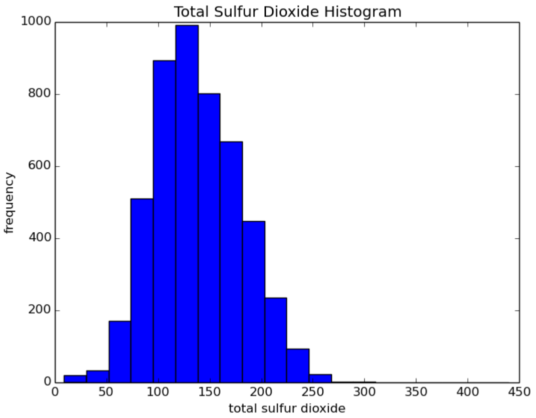


Figure 1.7 - Total sulfur dioxide histogram

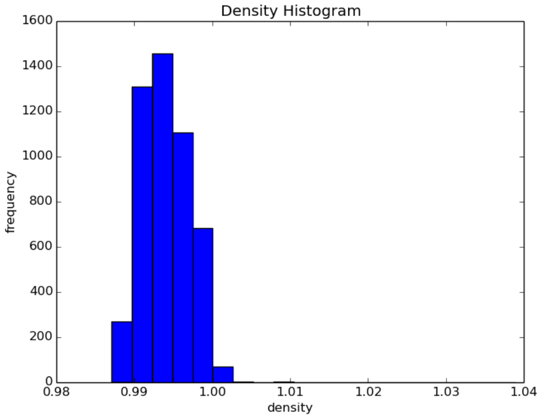


Figure 1.8 - Density histogram

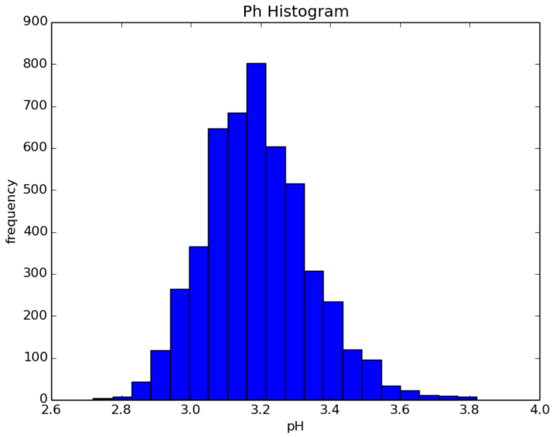


Figure 1.9 - pH histogram

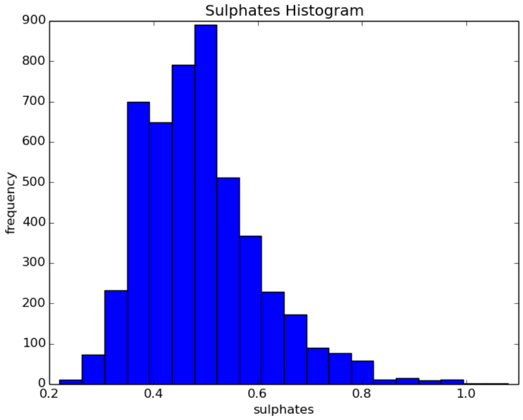


Figure 1.10 - Sulfates histogram

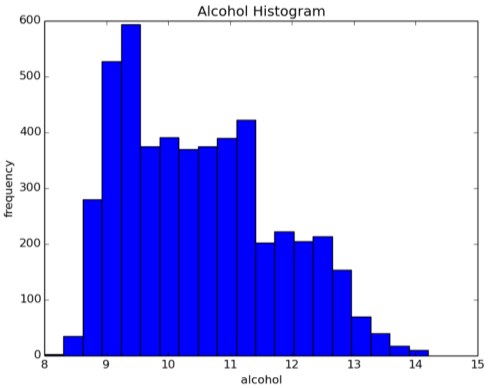


Figure 1.11 - Alcohol histogram

Most attributes follow approximately a normal distribution. Exceptions include residual sugar, sulfates, and alcohol which contain some outliers. More discussion on these distributions will be made in the Approach section.

IV. Approach

A. *Validation Metric*

The effectiveness of a data model can be hindered by the introduction of various types of error. Two of the most common are over fitting and bias introduced from having too large of a training set and too small of a test set. Over fitting occurs when the model is allowed to model too closely the training set, and as such is affected by disproportionately large error caused by deviations between the training set and the test set. Having a training set that consists of too much of the data and leaves a small test set reduces model effectiveness because the model is given an unfair amount of 'preparation' and the test set it is evaluated on is too small to give an accurate representation of effectiveness. These negative effects were mitigated in this research through the use of 10-fold cross validation in the model validation phase for each of these models. 10-fold cross validation works by creating different models from ten different training and test sets drawn from the data, and then averaging the resulting model. This reduces potential error and bias by training, testing, and creating ten different models, so any bias one model may experience is mitigated by averaging it with other models.

It order to evaluate the effectiveness of each different model in a way that allows for comparison of the models it is necessary to have one metric for success which is equally effective at judging every type of algorithm used. In this research the Root Mean Squared Error, or RMSE, was used as this unifying comparative tool. RMSE is a measure of the comparison between predicted and actual results, and is calculated by taking the square root of the sum of all deviations between predicted and actual results divided by the number of results, as shown in the below formula.

$$RMSE = \sqrt{\sum (Predicted - Actual)^2 / SetSize}$$

This is a useful measure for both regression and classification based predictive models because it can be calculated based on the confusion matrix produced by every predictive model, and is independent of the method used to create that confusion matrix.

B. *Regression*

The first and most straightforward model for wine quality prediction is linear regression. While the taster's quality rating is discrete, the values are ordered and can be treated as a continuous real number for the purposes of regression. Predicting a non-integer rating for a wine gives a reasonable approximation of an expert rating, and can be rounded if necessary.

Linear regression, without regularization, is the simplest model outlined herein. We used scikit-learn's ordinary least squares linear regression [7] as a basis point for comparison. We also added L2 and L1 regularization with Ridge and Elastic Net regression respectively.

Figure 2 - Naïve Bayes Results

Regressor	***White Wine RMSE***	***Red Wine RMSE***
Linear	0.7588	0.6597
Ridge(L2)	0.7617	0.6588
Elastic Net (L2 and L1)	0.7720	0.6643

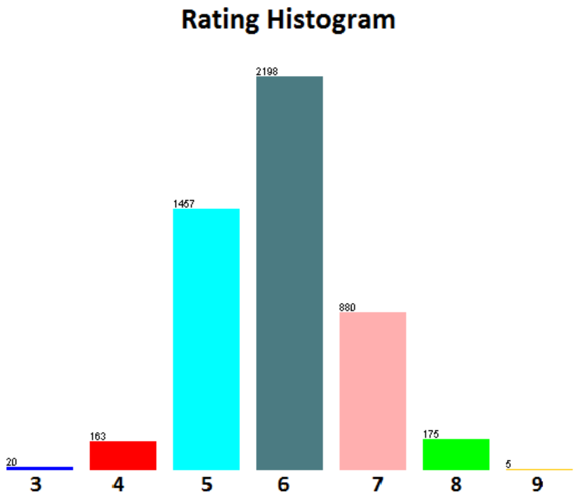
As shown in the table above, the RMSE when using a regularization term is slightly worse than pure least squares regression. This is because regularization disincentivizes larger coefficients to reduce over fitting. As the difference between regularized and non-regularized regression is so small, we can conclude that linear regression has almost no over fitting for this dataset.

C. *Classification*

As mentioned previously, regression is the most common approach for predicting continuous data. As a team we were interested to see if classification was possible on our dataset, what classification algorithms we could use, and how accurate they would be.

To understand if classification is appropriate, we have to look at the data being predicted: the wine quality rating. Over the 4898 items in our white wine dataset, each wine rating was found to be one of seven values in the range 3-9. The following histogram shows the distribution of these ratings.

Figure 3 - White wine rating histogram



Since the data we are trying to predict is exactly one of these seven values, we could consider each value a class. The relatively small number of discrete classes being predicted justifies the use of classification at a basic level and makes further investigation desirable. It should be noted that classification algorithms will not keep track of order or direction; that is, an algorithm will have no way to know that a rating of 9 is higher than a rating of 8. This may have an effect on accuracy.

Four types of classification algorithms were tested: Naive Bayes, SVM, PRISM, and Decision Trees. The reasons for experimenting with each of these algorithms and their prediction approaches will be discussed below.

1) *Naïve Bayes*

Naive Bayes is a classifier that predicts classes based on probabilities. It is based on Bayes' Theorem which describes the probability of an event based on evidence (or conditions) related to that event. In our case, we predict one of the seven ratings based on the ingredients of the wine.

It is important to consider the strengths and common uses for Naive Bayes before applying it to our dataset. Naive Bayes is most commonly used in text classification - that is, deciding whether a document belongs to one category or another. Most common applications include spam filtering and article classification (eg. sports vs. politics).

Since we are trying to predict continuous rating values using continuous attributes, Naive Bayes does not, at least from a traditional sense, make a lot of sense to use with our dataset. With that said, we were still curious to see the results from Naive Bayes. Perhaps our dataset had unforeseen conditions that help Naive Bayes make its prediction. If nothing else, this would provide a useful metric for comparison with other algorithms.

The three most common types of Naive Bayes are Gaussian, Multinomial, and Bernoulli.

In order to use Gaussian Naive Bayes, the attributes must have a normal distribution as Gaussian Naive Bayes uses the Gaussian assumption to calculate probabilities. As mentioned in Data Description, most of our attributes do follow a normal distribution. Some attributes, such as alcohol, do not appear entirely normal, but perhaps with a larger dataset they would.

To use Multinomial Naive Bayes, attributes would have to be placed in discrete bins. For example, you could say that any wine with alcohol percentages that fall within a certain range will be placed in one bin, and would be repeated for as many bins as desired. Multinomial Naive Bayes would then use frequencies of wines falling into a certain bin to calculate probability.

Bernoulli Naive Bayes expects attributes to be binary. This variation of Naive Bayes does not make a lot of sense for our dataset. To make this work, we would have to set some numeric threshold on each attribute that would consider it true or false. This is not ideal, as we would lose a lot of potentially useful information.

Our approach for Naive Bayes was to test Gaussian, and based on its results, decide whether multinomial was worth pursuing.

The table below shows our results for Gaussian Naive Bayes.

Figure 4 - Naïve Bayes Results

Classifier	***White Wine RMSE***	***Red Wine RMSE***
Gaussian NB	0.9455	0.7706

It is apparent that Gaussian Naive Bayes performs quite poorly. The RMSE for white wine was worse than our dummy classifier, and although red white performed better than the dummy classifier, it was by a very small margin.

Given this poor RMSE, we decided to not to pursue Naive Bayes any further. Naive Bayes is, as mentioned, suited more for classification that involves text rather than classification that involves continuous numeric data.

2) SVM

Support vector machine is known for being effective in multidimensional spaces [8]. With 12 attributes in this dataset, SVM has the potential to be investigated in this study. A SVM itself is a binary classifier, in order to make multi-class classification, one-vs-one approach and one-vs-all approach are used in this study

The One-vs-All classification method is to make one classifier for each class against all the other classes [9]. This method requires 10 classifiers in the case of this project. In example of classifier for class 1 (quality = 1), classifier returns positive on class 1, and it returns negative on all other classes. The class which has the highest return value of its own classifier will be returned as the result. In terms of tied results, the most common class will be taken as the result.

$$f(x) \ = \ arg \ MAX \, f_i(x)$$

All-vs-All classification method is to make one classifier for each pair of classes among all classes [9]. This method requires 10*9 classifier in this case. Each classifier is trained on data of two classes instead of all data. Each classifier is simply a binary classifier between 2 classes. The sum of positive return values of each class will be evaluated. The class with highest sum of positive return values will be the predicted result.

$$f(x) \ = \ arg \ MAX \ (\sum_i f_{ij}(x))$$

Even though it seems AVA has significantly more classifiers than OVA, the training space of each classifiers in AVA is significantly smaller than what is in OVA. In fact instead of 90 classifiers, 45 classifiers are enough since one classifiers is used for two classes. For example, the classifier return positive on class 1 and negative on class 2 is eventually the same as the classifier return positive on class 2 and negative on class 1. With some small twitch on the code, 45 classifiers can do the job of 90 classifiers. As a result, AVA has 4.5 times many classifiers of OVA with 2/10 times of the training time of each classifier. Assuming all classes are evenly distributed, the training time of AVA is actually 0.9 times of the training time of OVO. It is sufficient to conclude the total training time of two methods are approximately the same.

A python script was created to use the default SVM with AVA and OVA implementations on the wine quality data. The following is what the script does:

1. Load wine CSV file
2. Convert text data into a numpy compatible matrix
3. Split the raw data into two variables: data (X) and target (y)
4. Split the X and y data into training and target sets (90% training, 10% test)
5. For each AVA and OVA implementation, fit to the training data and predict on the test data
6. Evaluate the results of each implementation

SKlearn library was used in the script. Note AVA and OVA were called ovo (one-vs-one) and ovr (one-vs-rest). The result is shown in the Table below.

FIGURE 5 - DISTRIBUTION OF WHITE WINE ATTRIBUTES AND RATINGS IN THE TEST DATA

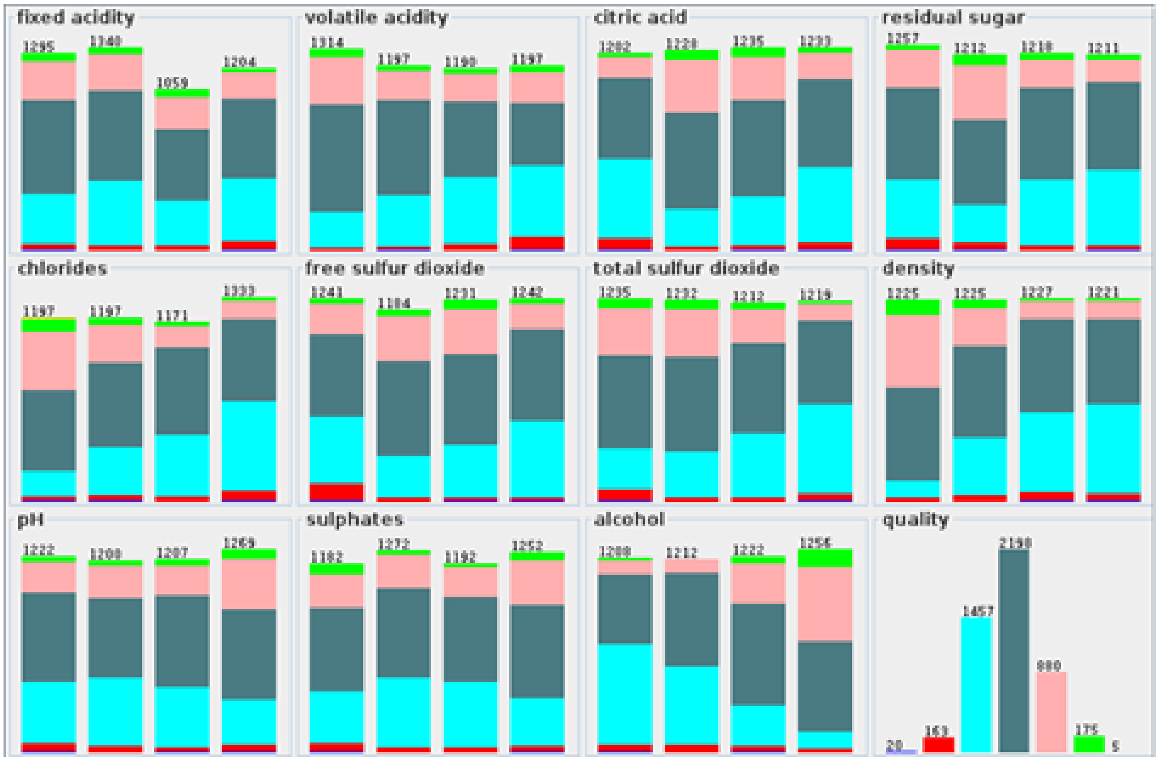


FIGURE 6 - SVM RESULTS

Algorithm	*Dataset*	*Number mislabeled*	*Total*	*RMSE*
SVM-AVA	Red Wine	70	159	0.772971
	White Wine	215	489	0.837515
SVM-OVA	Red Wine	70	159	0.772971
	White Wine	215	489	0.837515

The result shows that both AVA and OVA implementations of SVM gives the save result on the same data.

With experimenting on different train/test set generated by different random variables, AVA and OVA approach always shows the exact same result. Suspecting this was an implementation error, a different implementation method using sklearn.multiclass library [10] and LinearSVC was implemented. However the result of AVA and OVA approach are still the same. The cause of this result will be looked into future work.

3) Prism Algorithm

The PRISM algorithm is a rule based classifier. This algorithm splits a dataset into smaller increasingly homogenous sets by developing rules based on the dataset's attributes in order to synthesize a 'tree' which can be used to predict which class a new datum should be assigned based on the classes of its attributes. In the context of wine quality analysis, the values of the wine's attributes will be used to form rules which can be used to predict the quality of the wine.

The PRISM algorithm is designed to be implemented on data where each attribute is part of class. This introduced an added layer of complexity when it came to the wine dataset, because each wine attribute value fell on a continuous spectrum. In order to overcome this, the attributes were broken into classes, where the boundaries of class were set in order to have four classes per attribute with each class having an equal number of data points represented. The reason that this was done was so that no bias was introduced by increasing the likelihood that new wines would show up in a particular class. The quality ratings for each wine were not broken into bounded classes, they were converted directly from numeric to nominal values. This was done because there were only seven possible wine ratings expressed, and to create bounds would not result in a specific wine rating being predicted, just a range of ratings. Show in Figure 5 is the distribution which illustrates the boundaries used for each attribute white wine.

From here, even before the PRISM algorithm is run on the data it is possible to use the colour coding in the above Figure to draw basic conclusions about the effects of certain attributes on the wine quality. For example, looking at the alcohol content in both red and white wine it is clearly seen that as alcohol content increases the proportion of highly rated wines in each bucket markedly increases.

Using the PRISM algorithm to model the wine dataset, shown classified on the above distribution, the root mean squared error was found to be 0.7983 for white wine and 0.7226 for red wine. This result compared favourably to the other models used, being more accurate than Naïve Bayes, the dummy average predictor, Decision Trees, and SVM, but was less accurate than Linear Regression and Random Decision Forests.

The PRISM implementation resulted in a large number of rules generated; 663 rules were generated for red wine, and 2198 rules were generated for white wine. This compares to the entire dataset having a size of 4898 bottles for white wine and 1599 bottle for red, meaning that for white wine a rule was generated for every 2.2 bottles and for red wine a rule was generated for every 2.4 bottles. The comparatively large number of rules for each set suggests that over fitting is taking place or that the model may be suffering from other Type III errors, so 10-fold cross validation was used to mitigate this risk. This means that ten different sets of training and test data from the dataset were used to generate models, and these models were then averaged to create the final PRISM model.

4) Decision Trees

We initially used Decision Trees to classify the data based on the idea that, like regression, they would give us a good representation of which attributes were most important. Additionally, once built they offer a visual representation of the classifier. On the other hand, Decision Trees come with the difficulty of balancing generalization with over fitting. In general, these issues are solved by some combination of pre- and post-pruning. However, because this is not supported in scikit learn, we used a naive pruning method consisting of generating and evaluating trees of different heights and using the one that gave the best result. Some experimentation lead to the conclusion that for our data a tree depth of 4 gave the best results.

In order to increase the detail of our classifier (a tree of depth 4 is clearly not all encompassing for a dataset of over 5000), we switch to Random Forests. Using 4 attributes per tree, randomly selected with replacement, 100 decision trees were generated and predictions were made by averaging the results of each tree. This algorithm gave us the best results out of all the other classifiers.

FIGURE 7 - DECISION TREES RESULTS

Algorithm	***White Wine RMSE***	***Red Wine RMSE***
Decision Trees	0.8249	0.7577
Random Forests	0.6430	0.6322

V. RELEVANCE

An important question to winemakers is: what attributes of wine affect the quality? This is important at both extremes; the winemaker can optimize the most effective attributes and improve variety in the least effective attributes without compromising quality ratings.

Figure 9 is a heat map of the RMSE of linear regression run on every two attribute subset of the eleven attributes. Darker values indicate a lower RMSE, lighter values indicate higher RMSE after 10-fold cross-validation.

FIGURE 8

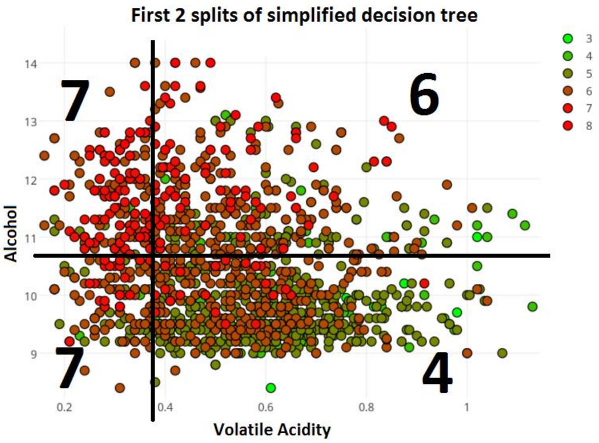
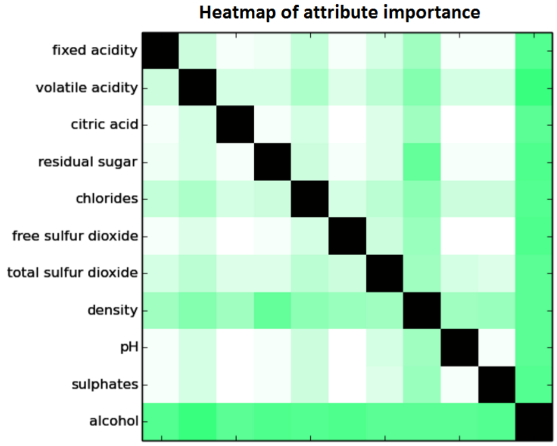


FIGURE 9



A lower RMSE indicates that linear regression was better able to find a line that matches the data, implying that those attributes are better predictors of wine quality. The most relevant attributes were alcohol, volatile acidity, density, and chlorides. The least relevant attributes were pH, citric acid, free sulphur dioxide, and sulphates. The most relevant combination was alcohol and volatile acidity.

Figure 8 shows a distribution of red wines by alcohol and volatile acidity. There is a clear trend where more highly rated wines have more alcohol and a lower volatile acidity. The large axis on the above graph indicates the top two decision boundaries of a pruned decision tree trained on all eleven attributes. The most differentiating choices were the alcohol and volatile acidity, with the predicted qualities much higher in the high-alcohol/low-volatile acidity quadrant.

An important caveat of the above conclusions is that our dataset does not include pricing or sales numbers for wine. While we can recommend winemakers focus on high alcohol and low volatile acidity to produce high quality ratings, these measurements do not predict the financial success of a particular wine.

VI. Combined Results

In this section we will discuss some complications that occurred when producing our results followed by our final, combined results.

A. *Complications*

When the results were initially compared, we found that the RMSE of PRISM was less than half of our second best classifier. Initially we interpreted this as being due to PRISM being a very good fit for our data. However, after further contemplation about the size of the increase in accuracy, and noting that PRISM was the only algorithm run in Weka, we decided to investigate further.

Figure 10 – Initial Results

	NB	***Decision Trees***	***SVM***	***PRISM***
White Wine	0.9455	0.8249	0.8375	0.3193
Red Wine	0.7706	0.7577	0.7730	0.3369

Upon running the other classifiers in Weka we discovered they all returned an RMSE of around 0.3. At first we assumed that Weka was doing things under the hood that made the algorithms perform much better. However after running ZeroR and receiving an RMSE of around 0.35 we were able to confirm that something was off with the RMSE calculation in Weka. The issue was that in order to run most of the classification algorithms, Weka required the class attribute to be in nominal form, and therefore returns an RMSE with any wrong value having an error of 1.

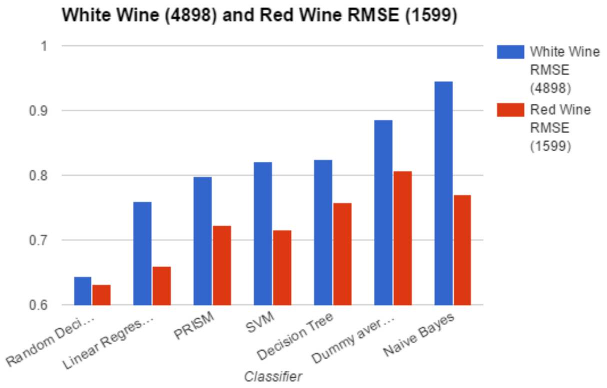
Nominal means the values an attribute can take are separate, named values with no relation to each other. On the other hand, numeric values are real numbers within a range. When doing the classification manually, we were treating the Quality as nominal during classification, but numeric when taking the RMSE. This approach makes more sense, because the classification algorithms require the data in nominal form to work, but predictions that are "closer" to the correct value should be considered to have less error.

With this information we were able to manually calculate the RMSE of each algorithm using the confusion matrix in Weka in order to make sure each RMSE was acquired using the same method.

B. *Final Results*

Figure 11 shows the final results of each algorithm for white and red wine with proper RMSE values.

Figure 11



VII. Conclusion

Looking back at the motivation of this research, namely attempting to mine a dataset composed of many red and white wines in order to try and develop a model to predict the quality of a wine based on the chemical composition of that wine, and by extension taking a look at the consistency of the field of wine tasting, it is possible to draw several conclusions from the results of the various models constructed.

Before considering any results which may be proposed, the limitations of the scope of this evaluation must be understood. First, it is important to note that the data set consists of wine metrics drawn from a selection of 6499 Portuguese "Vinho Verde" wines, of which 4899 bottles are white wine, and 1600 bottles are red wine, so any conclusions drawn are only applicable to this particular variety of wine. Also, because wine tasting is a qualitative, and to a degree subjective, judgment by a selection of professional sommeliers, the models developed in this research will be biased toward modeling the tastes of the particular sommeliers who generated the data, which may differ from the ratings which would have been given by different sommeliers.

The dummy average predictor model, which predicted the average rating for every wine had a RMSE of 0.8855 for white wine and 0.8073 for red wine. This established a lower bound of model effectiveness and, yielding a correct rating so rarely, indicated an extremely poor base predictive standard. Except for the Naïve Bayes classifier, applying the models discussed in the previous sections, all yielded increased accuracy over the dummy average predictor, with Random Decision Forests modeling the data most accurately with a RMSE of 0.6430 for white wine and 0.6322 for red wine.

Having all but one model predict, with a greater or lesser accuracy, more effectively than the dummy classifier would seem to suggest that while it is impossible to determine from this research whether or not wine tasting is an actual science, the results of these particular wine tastings were at least to a certain degree consistent and able to be modeled using a number of data mining techniques. This conclusion is further supported by the various data visualizations in the preceding sections, where the links between certain attributes and wine quality are strongly evident even to the untrained eye.

In conclusion, this research would seem to suggest that Random Decision Forests, with the lowest RMSE, is the best model to predict wine quality. This was a surprise as we had expected regression to give better accuracy given the continuous numeric data. We suspect this may be due to Random Decision Forests making discrete decisions similar to how a real wine taster would evaluate wine quality.

Additionally, wines with a low volatile acidity, low amounts of chlorides, high total sulfur dioxide, low density, and high alcohol content, are more likely to be of a higher quality as shown in the preceding visualizations, with alcohol content having the most marked association. This result would seem to line up with anecdotal evidence that a beverage with a high alcohol content quickly begins to taste better and be easier to drink the more one drinks it.

VIII. Future Work

In order to further increase the accuracy of our classifiers it is clear that either the data or the algorithms must be tweaked. We would recommend feature engineering, taking advantage of the potential relation between wine qualities, or applying boosting algorithms on the already most accurate methods.

The features included in our data set give a good high dimensional representation of each wine product, but it is possible that there are correlations within the features that are not immediately visible. For example, according to [11], in general pH is a "quantitative assessment" of fixed acidity. This implies that fixed acidity and pH should be highly correlated and perhaps could be merged to reduce the number of attributes, simplifying the problem.

Because we know that Random Forests was the best classifier, with regression tailing directly behind it, we estimate that Gradient Boosted Decision Trees has high potential to perform even better due to the large reliance on Random Forests, using regression to boost the performance [12]. By taking our weak learners and weighting them one could conceivably achieve better results.

References

[1] R. Gonzalez (2013, May, 8) "Wine tasting is bullshit. Here is why."[Online] Available: http://io9.gizmodo.com/wine-tasting-is-bullshit-heres-why-496098276

[2] F. Brochet, "Chemical Object Representation in the Field of Consciousness" Avilable: http://web.archive.org/web/20070928231853/http://www.academie-amorim.com/us/laureat_2001/brochet.pdf

[3] G. Morrot, F. Brochet, D. Dubourdieu, "The Color of Odors" August 2001 Avilable: https://web.stanford.edu/class/linguist62n/morrot01colorofodors.pdf

[4] P. Cortez, A. Cerdeira, F, Almeida, T. Matos, and J. Reis, "Modeling wine preferences by data mining from physicochemical properties," Decision Support Systems, vol. 47, iss. 4, pp. 547-553, Nov. 2009.

[5] A. Bednarova, D. Brodnjak-Voncina, R. Kranvogl, and T. Jug, "Prediction of wine sensoral quality by routinely measured chemical properties," Nova Biotechnologica et Chimica, vol. 13, iss. 2, pp. 182-196, Feb. 2015.

[6] P. Cortez. (2010, Oct 2). Wine Quality Data Set [Online]. Available: http://archive.ics.uci.edu/ml/datasets/Wine+Quality

[7] scikit-learn developers. (2014). Generalized Linear Models - scikit-learn 0.17.1 documentation [Online]. Available: http://scikit-learn.org/stable/modules/linear_model.html

[8] scikit-learn developers. (2014). Support Vector Machines - scikit-learn 0.17.1 documentation [Online]. Available:http://scikit-learn.org/stable/modules/svm.html

[9] R.Rifkin. (Spring, 2008) "Multiclass Classification" Statistical Learning Theory and Application class at MIT. Available: http://www.mit.edu/~9.520/spring08/Classes/multiclass.pdf

[10] scikit-learn developers. (2014). Multiclass and multilabel algorithms - scikit-learn 0.17.1 documentation [Online]. Available: http://scikit-learn.org/stable/modules/multiclass.html

[11] Calwineries. (2016). Acidity [Online]. Available: http://www.calwineries.com/learn/wine-chemistry/acidity

[12] T. Srivastava. (2015, Sept 11). Learn Gradient Bootsing Algorithm for better predictions [Online]. Available: http://www.analyticsvidhya.com/blog/2015/09/complete-guide-boosting-methods/

Task Breakdown

Name	***Task***
Noah	• Decision Tree and random forest design, implementation, analysis • Result visualization • Document formatting
Murray	• Regression and decision tree design, implementation, analysis • Relevance analysis
Chris	• PRISM design, implementation, analysis • Data visualization
Greg	• Naïve Bayes design, implementation, analysis • Attribute distribution analysis
Haoyan	• SVM design, implementation, analysis • Background research