Wine Classification – Application Stream

ST494 Final Project

Myisha Chaudhry (200591740)

Kayleigh Habib (200370580)

Abigail Lee (200469770)

ST494 – Statistical Learning

Devan Becker

April 12th, 2023

Table of Contents

Executive Summary

Currently, wine quality is determined based on sensory tests performed by wine experts, which could result in subjective opinions. The purpose of this analysis is to use statistical learning methods to provide wine producers, connoisseurs, sommeliers, distributors, and restaurants with a less subjective way to determine the quality of wines they are interested in or wish to sell to their customers. This study is focused on the Vinho Verde Wine data retrieved from the University of California, Irvine (UCI) machine learning repository and included 6497 records. Exploratory data analysis (EDA) was conducted, using correlation plots, bar plots, group boxplots, as well as principal component analysis (PCA).

The study implemented four classification methods, creating a Logistic Regression model, KNN model, Random Forest Model, and Support Vector Machine (SVM) model. The results of the analysis indicated that each of the models explored provided more than 70% accuracy in predicting the wine quality. The Random Forest model provided the highest predictive ability and using the model we determined that alcohol content was the most important feature in predicting wine quality.

Further analysis could be performed using additional features to include other wine regions as well as climate. Also, other classification models can be applied, or an ensemble of models in order to improve predictive performance.

1.0 Introduction

1.1 Purpose

Our target audience includes wine producers, connoisseurs, sommeliers, distributors, and restaurants who need to know the quality of a particular wine, given certain attributes. The purpose of this analysis is to provide these parties with an objective means of predicting wine quality so that they could maintain a good standing in the wine community.

1.2 Constraints

Due to timing constraints, we are only able to analyze a limited number of models. With more time, we would have been able to dive deeper into applying various other classification and feature engineering techniques.

The scope of this analysis will be limited to the *Vinho Verde* wine which is exclusive to the north of Portugal and does not consider wine from outside this region. Since *Vinho Verde* wine is predominantly white wine, the data does not provide an equal representation of white and red wine (as seen in the data set of 4898 records, versus 1599 records). For this reason, we are restricted to analyzing more white than red wine, which could lead to our data being trained to understand the qualities and features of white wine better than red wine, therefore leading to a bias.

2.0 Data

2.1 Data Collection

The data was retrieved from the University of California, Irvine (UCI) machine learning repository and includes information on both red and white wine types from the northern Portugal region.

2.1.1 Data Description

The data was provided in 2 separate datasets: one for red and one for white wines. As they contained the same columns, we combined them into a single dataset, and added a new column for “type”. The combined data frame contained 6497 rows and 13 columns. For further analysis of each variable in the data, see Appendix A. There were no missing data fields.

2.2 Exploratory Data Analysis

We created dummy variables for the type of wine to help with later analysis (such as a correlation matrix as shown in Figure B3) as a numeric value for the type was needed. The target “quality” field consisted of integers ranging from 3 to 9. Further analysis of this field indicated that it had a median of 6 and a 3rd quartile of 6. Thus, we decided to create groupings for this variable, with the first grouping of “low” consisting of qualities between 3 to 5, and the grouping of “high” consisting of values from 6 to 9. The quality type of “low” contained 2,384 records, the type “high” contained 4,113 records. The distribution of low and high quality wines are shown in Figure B1.

We created group boxplots to compare the distributions of the features by wine quality. The boxplots show that high quality wine has a higher mean alcohol level than that of low quality wine. The other features such as pH, fixed acidity, and sulphate seem to have fairly even distributions and means.

As part of our EDA, we used a ggpairs plot to create scatterplots showing the relationships between pairs of features in our dataset. This indicated that wine quality is dependent on some combinations of the features. A correlation matrix was produced to more closely look at whether there was any collinearity between features. These charts are shown in Figures B2 and B3.

We applied principal component analysis (PCA) to explore whether we could further reduce the dimensionality of the data, or to look for any outliers. Based on the results of the PCA, we determined that we could use the first 7 components which would explain over 90% of variation in the data. The results of the PCA are shown in Appendix C. Ultimately, there was no significant improvement once PCA was applied, thus it was not used for the remainder of the analysis.

2.3 Feature Selection

Using the correlation plot, we found that Free Sulfur Dioxide and Total Sulfur Dioxide had a high positive correlation of 0.72, so removing one of these variables was possible. Free Sulfur Dioxide was then removed from the dataset as it was less correlated with the target, compared to Total Sulfur Dioxide.

2.4 Feature Scaling

As the features all have different measurements units and ranges, it is good practice to scale the fields as this will improve the performance of the statistical models. For each numeric feature, scaling involved subtracting the mean value of the feature and dividing by its standard deviation.

3.0 Methodology

3.1 Training and Testing Data

The data was split into a training set and a test set, using a random 80/20 split. Thus, 80% of the data (5,142 records) was used for training the models, while 20% of the data (1,355 records) was used for evaluating model performance. A seed of 0 was used for this split and to ensure reproducibility of results.

3.2 Models Used

3.2.1 Logistic Regression

The first method we used was a Logistic Regression. A logistic regression is used for classifying binary outcomes based on the probability that a record belongs to a specific class. If this probability is less than 0.5, the outcome is assigned to class 0 ("low"), otherwise it is assigned to class 1 ("high").

3.2.2 K-Nearest Neighbours

The second method we used was the K-Nearest Neighbours (KNN) method. KNN is used to classify individual data points into categories based on proximity to other data points. Although this method is easy to use and requires a few hyperparameters, KNN does not do well when dealing with high dimensional data and is at a high risk of overfitting.

3.2.3 Random Forest

For the third method, we implemented a Random Forest model. Random Forest combines the output of multiple decision trees to reach a single result which is useful for classification problems. Random Forest adds additional randomness to the model while growing the tree. It searches for the best feature among a random subset of features instead of searching for the most important feature.

3.2.4 Support Vector Machine

The last model we implemented was a Support Vector Machines (SVM). SVMs work by mapping data to a high-dimensional feature space in order to categorize data. There is then a separator found between the categories identified, and this is drawn as a hyperplane. An SVM function can use various kernels such as linear, polynomial, RBF and sigmoid.

3.3 Model Optimization

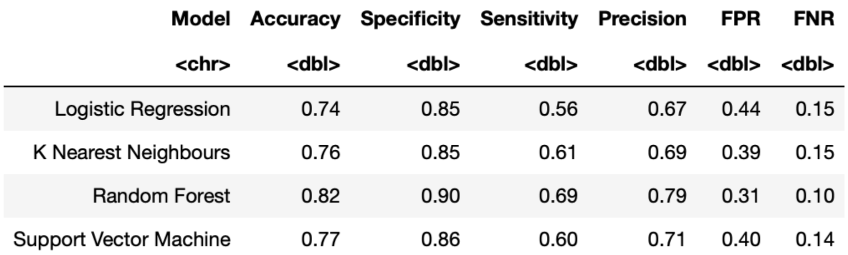
For each of the models, we used the caret package in R to first determine the optimal hyperparameters that would produce the highest accuracy. For this process we applied cross validation as well as a parameter grid to be used in tuning the model. The cross validation allowed the model to be tuned on random samples of the train data in order to reduce the likelihood of overfitting the model. Once the optimal parameters were identified, we created a final version for each model using the full training dataset. These final model versions were then used for comparing predictions on the test data. The final tuning parameters are shown in Appendix D.

4.0 Results

4.1 Confusion Matrix Metrics

To compare the various models, a confusion matrix was created for each one and the related metrics (Accuracy, Specificity, Sensitivity, Precision, False Positive Rate (FPR) and False Negative Rate (FNR)) were determined. All these metrics were calculated on the test data. Depending on the application of these models, different measurement metrics would be useful. For example, if the goal is to minimize the misclassification of low quality wines, then the appropriate metric would be minimizing the FPR.

Table 1 summarizes the results of the confusion matrix metrics, assuming a threshold of 0.5 for assigning predictions to the respective classes.



Model	Accuracy	Specificity	Sensitivity	Precision	FPR	FNR
<chr>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>	<dbl>
Logistic Regression	0.74	0.85	0.56	0.67	0.44	0.15
K Nearest Neighbours	0.76	0.85	0.61	0.69	0.39	0.15
Random Forest	0.82	0.90	0.69	0.79	0.31	0.10
Support Vector Machine	0.77	0.86	0.60	0.71	0.40	0.14

Table 1: Summary of Confusion Matrix Metrics

This indicates that the model with the highest test accuracy is the Random Forest while the model with the lowest test accuracy is Logistic Regression.

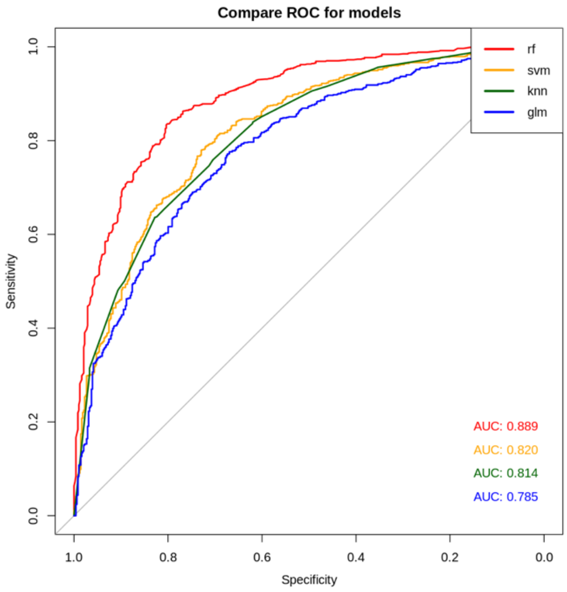
The False Positive Rate represents a low quality wine that is misclassified as high quality, while the False Negative Rate, represents a high quality wine being misclassified as low quality. In our analysis, the model with the highest FPR is Logistic Regression while the model with the lowest FPR is Random Forest. In addition, the model with the highest FNR is KNN, while the Random Forest model had the lowest FNR.

Specificity and Sensitivity can also be used to evaluate how well a model is performing. Specificity is the number of True Negatives (TN) which represents the number of low quality wines that are correctly classified, while sensitivity is the number of True Positives (TP) which represents the number of high quality wines that are correctly classified. From Table 1 , the Random Forest model has the highest Specificity while the KNN model has the lowest. The model with the highest Sensitivity is also Random Forest, while the Logistic Regression has the lowest Sensitivity.

4.2 ROC Curve and AUC

The Receiver Operating Characteristic (ROC) curve was constructed to measure the performance of each classification method implemented. The ROC curve is a probability curve measuring the performance at different thresholds. It tells how much the model is capable of distinguishing between classes where the area under the curve (AUC) represents the degree of separability. Figure 1 shows the ROC curve and AUC for each of the models.

Figure 1: ROC Curve and AUC Metrics



The Random Forest model again outperforms the other models on this metric, with an AUC of 0.889, while the Logistic Regression had the lowest AUC of 0.785.

4.3 Variable Importance

Using the Random Forest model, we looked at the variable importance, which indicated that alcohol had the most impact on predicting wine quality, while type (“red” or “white” wine) had the least significance. The result of the variable importance plot is shown in Appendix E.

5.0 Conclusion

Our analysis indicated that statistical methods can be applied to predict the quality of wines as “high” versus “low” quality based on their chemical content. This type of analysis could be very useful to both producers and distributors of wine, who currently rely on the subjective opinions of wine tasters. With the given data, we determined that a Random Forest model outperformed the other classification models analyzed both in terms of accuracy and AUC. This is likely due to the fact that tree-based models tend to be more effective than other classification models when data is imbalanced, as they tend to penalize misclassifications in the minority class.

To improve accuracy, several approaches can be further explored:

1. Obtain additional data fields, such as information of climate, year etc. that may be relevant to the quality of the grapes used in making wines.
2. Explore additional classification techniques such as neural networks or use an ensemble of methods.

6.0 Appendices

Appendix A: Data Description

Features	Definition
Fixed Acidity (g/dm^3)	Amount of tartaric acid in wine.
Volatile Acidity (g/dm^3)	Amount of acetic acid in wine. Too high of levels can lead to an unpleasant taste.
Citric Acid (g/dm^3)	In small quantities, citric acid can add ‘freshness’ and ‘flavour’ to wines.
Residual Sugar (g/dm^3)	Amount of sugar remaining after fermentation stops. Wines with greater than 45 grams/litre are considered sweet.
Chlorides (g/dm^3)	Amount of sodium chloride in wines impacts how salty the taste is.
Free Sulfur Dioxide (mg/dm^3)	Free form SO2 exists in equilibrium between molecular SO2 (as a dissolved gas) and bisulfite ion; it prevents microbial growth and the oxidation of wine.
Total Sulfur Dioxide (mg/dm^3)	Amount of free and bound forms of S02. It becomes evident in the smell and taste of wine.
Density (g/cm^3)	Determined by the concentration of alcohol, sugar, glycerol, and other dissolved solids.
pH	Describes how acidic or basic a wine is on a scale from 0 (very acidic) to 14 (very basic). Most wines are between 3 and 4 on the pH scale.
Sulphates (g/dm^3)	Wine additive which can contribute to sulfur dioxide gas (S02) levels, which acts as an antimicrobial and antioxidant
Alcohol (volume %)	The percent content of alcohol in the wine
Quality	Based on sensory data: (score between 0 and 10)
Type	Type of wine: “Red” or “White”

Table A1: Description of Data Attributes

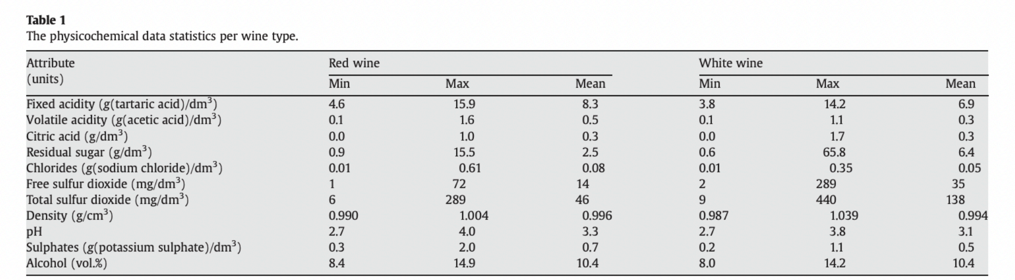


Table 1
The physicochemical data statistics per wine type.

Attribute (units)	Red wine			White wine		
	Min	Max	Mean	Min	Max	Mean
Fixed acidity (g(tartaric acid)/dm^3)	4.6	15.9	8.3	3.8	14.2	6.9
Volatile acidity (g(acetic acid)/dm^3)	0.1	1.6	0.5	0.1	1.1	0.3
Citric acid (g/dm^3)	0.0	1.0	0.3	0.0	1.7	0.3
Residual sugar (g/dm^3)	0.9	15.5	2.5	0.6	65.8	6.4
Chlorides (g(sodium chloride)/dm^3)	0.01	0.61	0.08	0.01	0.35	0.05
Free sulfur dioxide (mg/dm^3)	1	72	14	2	289	35
Total sulfur dioxide (mg/dm^3)	6	289	46	9	440	138
Density (g/cm^3)	0.990	1.004	0.996	0.987	1.039	0.994
pH	2.7	4.0	3.3	2.7	3.8	3.1
Sulphates (g(potassium sulphate)/dm^3)	0.3	2.0	0.7	0.2	1.1	0.5
Alcohol (vol.%)	8.4	14.9	10.4	8.0	14.2	10.4

Source: *https://journals-scholarsportal-info.libproxy.wlu.ca/pdf/01679236/v47i0004/547_mwpbdmfpp.xml_en*
Table A2: Summary Statistics for Data Attribute

Appendix B: Exploratory Data Analysis

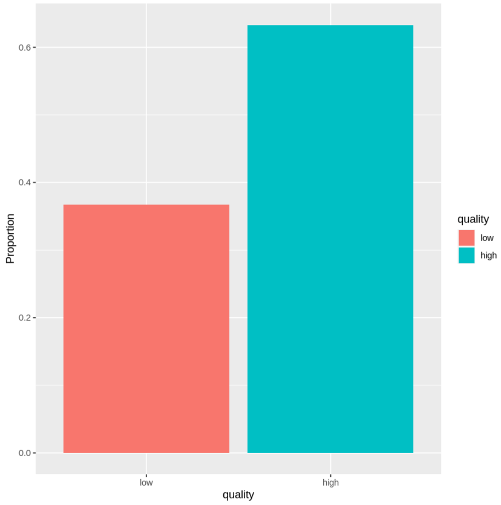


Figure B1: Distribution of Target Variable

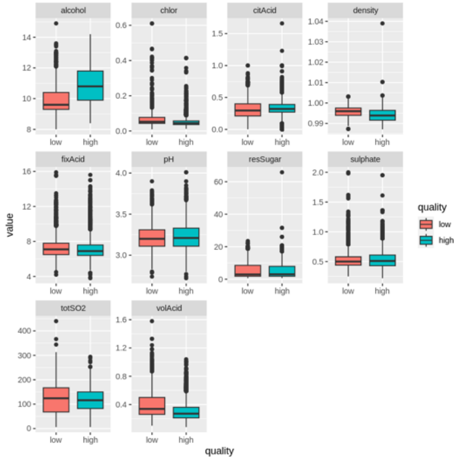


Figure B2: Distribution of Features by Target Variable

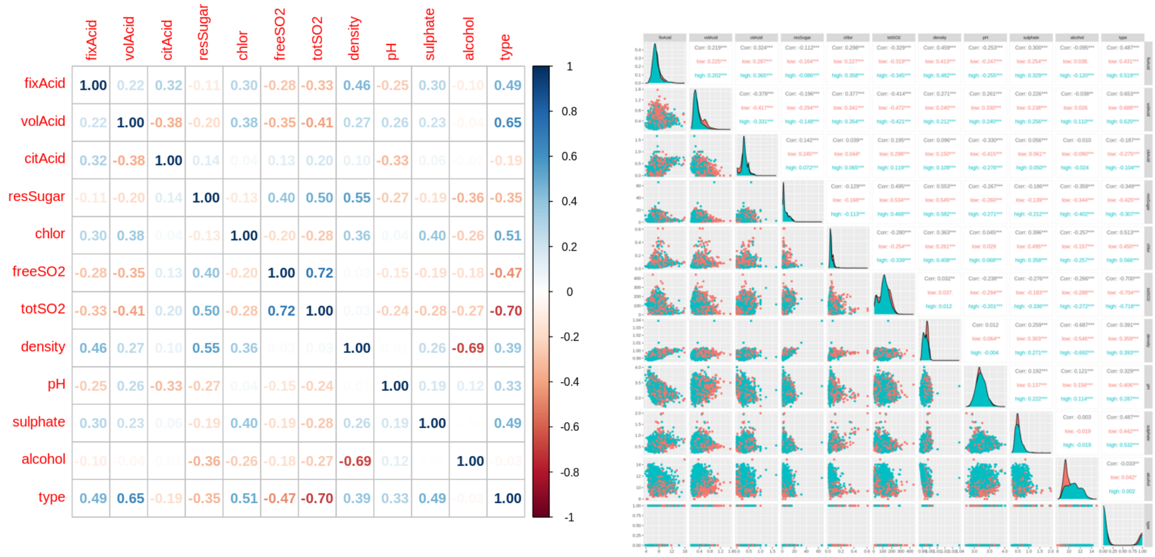


Figure B3: Correlation and Pair Plot of Features by Target Variable

Appendix C: Principal Component Analysis

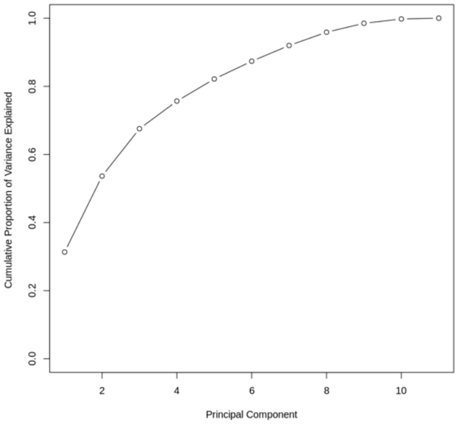


Figure C1: Cumulative Graph for PCA Components

Appendix D: Model Parameters

Model	Parameter Grid	Final Parameters
Logistic Regression	--	--
K Nearest Neighbours	K = seq(5,31,by = 2)	K = 9
Random Forest	mtry = seq(1,11,by=1)	mtry = 3

Support Vector Machines	Kernel = c(linear, radial)
	Cost = c(0.1,1,10)
	Gamma = c(0.5, 1, 2)
	Sigma = c(0.1,1)
		Kernel = radial
		Cost = 1
		Gamma = 1
		Sigma = 0.1

Table D1: Parameters used in Models

Appendix E: Random Forest Features

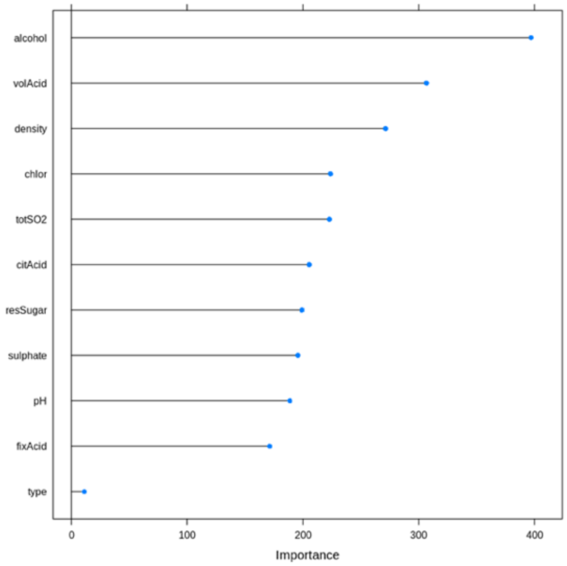


Figure E1: Important Features

References

Narkhede, Sarang. “Understanding AUC - Roc Curve.” *Medium*, Towards Data Science, 15 June 2021, https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5.

Nguyen, Leah. “Cofounding Variables Exploration with Wine Quality Dataset.” *Medium*, Medium, 29 Mar. 2022, https://medium.com/@ndleah/cofounding-variables-exploration-with-wine-quality-dataset-e05c7ceda6d9.

“Random Forest: A Complete Guide for Machine Learning.” *Built In*, https://builtin.com/data-science/random-forest-algorithm.

Tuning an SVM Model, https://www.ibm.com/docs/en/spss-modeler/saas?topic=models-tuning-svm-model#svm_perfimp.

UCI Machine Learning Repository: Wine Quality Data Set, https://archive.ics.uci.edu/ml/datasets/wine+quality.

“What Is Logistic Regression?” *IBM*, https://www.ibm.com/topics/logistic-regression.

“What Is the K-Nearest Neighbors Algorithm?” *IBM*, https://www.ibm.com/topics/knn #:~: text=The%20k%2Dnearest%20neighbors%20algorithm%2C%20also%20known%20as%20KNN%20or,of%20an%20individual%20data%20point.

“What Is Random Forest?” *IBM*, https://www.ibm.com/topics/random-forest.

Delineation of Work

<table>
<tr><td colspan="2">The project was split evenly (1/3) among all group members, with collaboration at each step. A further break down of the overall structure is as follows:</td></tr>
<tr><td>Myisha Chaudhry</td><td>Data Description, Data Manipulation, K-Nearest Neighbours, SVM, Report</td></tr>
<tr><td>Kayleigh Habib</td><td>EDA, Linear Regression, ROC Curve and AUC, Report</td></tr>
<tr><td>Abigail Lee</td><td>Data Description, PCA, Random Forest, Metrics Calculations, Report</td></tr>
</table>