Implementing Various Classification Algorithms

by:

Darius Kharazi

October 23, 2017

The Ohio State University

1. Introductory Program Output Description

The provided program and analysis is able to output useful information that is extracted from the “wine” data. Once the sample is read, exploratory analysis and data modification is performed on each observation. Additionally, multiple classifiers are implemented in order to find a model or classifier that “best fits” the given data. In the early stages of fitting various models, the data is separated and cross-validated into training and testing data. Once the data is cross validated for each classifier, further analysis is performed, such as analyzing each classifier’s corresponding confusion matrix after cross-validation. Lastly, the classifiers are all compared to one another by their relative predictive values, such as true positive rates, true negative rates, and accuracies. In the end, a particular classifier will be preferred through statistical analysis and testing.

2. Exploratory Data Analysis

According to the pairwise plots between the continuous variables in the dataset, it seems like there aren't many outstanding variables that are correlated with quality or class. Some variables, such as "sulph," "tot_sulf_d" and "free_sulf_d," seem to share some level of multicollinearity. According to the summary of the "quality" variable, a large amount of the observations seems to be given a quality of 6.0, since the median is 6.0, mean is 5.64, and the 3rd quartile is 6.0. The majority of the observations seems to have a quality of around 6.0, since both the first and third quartiles are very close to the median/mean. However, the histogram of wine quality seems to be slightly skewed left, indicating the mean is on the right of the peak value (Figure 1). Additionally, there doesn't seem to be many wines with an alcohol content smaller than 9, but the majority of wines have an alcohol content between 9 and 10, according to the histogram, indicating a huge jump of alcohol content (Figure 2). According to the summary of wines with the highest alcohol contents and their corresponding box plots, wines with the highest alcohol content in the sample seem to have higher overall qualities compared to the general sample. Furthermore, the histogram relating to wine containing an extremely high alcohol content is left skewed, indicating a higher overall quality compared to the general sample’s quality (Figure 3). However, the differing sample sizes between the wines containing a higher alcohol content and the entire sample should be noted. The majority of wines have low levels of sulphates, which can be seen in the right-skewed histogram relating to the sample's sulphate levels (Figure 4). With additional research, this may seem obvious, since low levels of sulphates are used in wines to maintain freshness, but large amounts of sulphates tend to reduce the quality of wine. The majority of wines have low levels of total sulfur dioxide and free sulfur dioxide, since each variable's corresponding histograms are right-skewed, which validates our previous claim about the sulphates in wine

(Figure 5; Figure 6). This also could indicate some levels of multicollinearity between the three variables, which would validate our claim mentioned earlier, as well.

First, linear regression was used in the analysis, since the goal is to predict the overall quality, or the "class" variable. In an attempt to find the most statistically significant variables, we fit the full model and observe each variable's p-values, which are calculated from z-tests. From the coefficient summary, it seems clear that there are potential predictors to drop. The variables "fx_acidity," "citric_acid," "resid_sugar," and "density" should be dropped, since their p-values are relatively large, meaning they are marginally rejected from the model. On the other hand, the variables "vol_acidity," "chlorides," "free_sulf_d," "tot_sulf_d," "pH," "sulph," and "alcohol" should be included in a reduced model, since the p-values are extremely small. Also, it is important to note that the AIC score is 3162, which will be useful for any model comparison made in the future. The R-squared value is fairly low, which indicates that our model does not have much predictive power. Additionally, the residual plots are good, since the residuals are randomly distributed along the x-axis. However, the q-q plot demonstrates some skewness/ departure along the left tail, which challenges the normality assumption.

Since there are only 11 variables, an exhaustive variable search should be performed, while excluding the "quality" variable, since it is essentially the same variable as our response. From the exhaustive search, it seems that the "alcohol" variable has the most explanatory power, followed by the "vol_acidity" variable. The remaining variables compete with each other, but are not as strong as "vol_acidity" and "alcohol." Additionally, it seems that the best model according to the largest R-Squared value is model 15, which contains "fx_acidity," "vol_acidity," "chlorides," "pH," "sulph," and "alcohol." Furthermore, it seems that the best model according to the smallest BIC values is model 12, which contains "vol_acidity," "chlorides," "pH," "sulph," and "alcohol." Clearly, there are overlapping variables, but, more importantly, the "alcohol" and "vol_acidity" variables appear in both models. It's important to note, for future model comparisons, that the AIC values for both models are around 3184.

Next, we fit a logistic regression model, since we are wanting to predict the "class" variable, which has been binarized in the data transformation section of our analysis. In an attempt to find the most statistically significant variables, the full model is fit, and each variable's p-values are calculated from z-tests and observed. From the coefficient summary, it is clear that there are potential predictors to drop. The variables "fx_acidity," "resid_sugar," "density," and "pH" should be dropped, since their p-values are relatively large, meaning they are marginally rejected from the model. On the other hand, the variables "vol_acidity," "citric_acid," "chlorides," "free_sulf_d," "tot_sulf_d," "sulph," and "alcohol" should be included

in a reduced model, since the p-values are extremely small. The residual plots seem to perform fairly well, since the residuals are randomly distributed along the x-axis between -2 and 2. After comparing the AIC scores between the full model and the reduced model mentioned above, it seems that they both have an AIC score of 1679. The AIC score of the reduced model does not improve. However, since the model is simpler than the full model, the reduced model is preferred. Lastly, it seems that we will most likely need to deal with multicollinearity between particularly similar variables, such as "free_sulf_d" and "tot_sulf_d" after analysis of the coefficient summary and the previously conducted exploratory analysis. However, we should prefer an approach involving logistic regression predicting "class," rather than an approach involving linear regression predicting "quality," since the logistic regression model has a smaller AIC score.

3. Data Transformation

For modeling our data using logistic regression, the "class"" variable was slightly modified. The "high" values were assigned to the value "1," and the "low" values were assigned to "0," in order to properly fit a logistic regression model on the data. Outliers were not treated, after further examination. Outliers remained in the analysis, since they contained different values, and contributed some level of predictive power to our analysis. For example, many of the outliers contained high levels of alcohol or extremely low levels of alcohol. Since alcohol content seems to have some level of correlation with wine quality, this sometimes caused certain wines with high levels of alcohol content to be considered outliers. Additionally, observations with missing values were deleted, since our sample contained nearly 1,600 observations, and only 2 observations contained any missing values. Throughout the model selection approaches, two variables seemed to be included in every model: "alcohol" and "vol_acidity." However, since the full model had a very small R-Squared value, and since the data does not contain an overwhelmingly large amount of parameters, the full model will be primarily considered in further model development.

4. Model Development

In R, the Decision Tree classifier is fairly simple to implement when using the "rpart" package. The Decision Tree classifier in this package seems to be widely used and accepted amongst statisticians that use R, as well. It seems that the model only results in an accuracy of 77% on the training dataset, before even performing cross validation. A single iteration results in a True Positive Rate equal to 81%, a True Negative Rate equal to 73%, a False Positive Rate equal to 27% , and a False Negative rate equal to 18%. For validation purposes, we used the "caret" package's Decision Tree classifier to verify our findings, along with

performing a 10-fold cross validation, as well. This approach resulted in a somewhat different accuracy measurement: an accuracy of 73%, a True Positive Rate equal to 85%, and a True Negative Rate equal to 59%. Therefore, the initial Decision Tree classifier seemed to have been overfitting the data, since the results from cross-validation were somewhat worse, except for the True Positive Rate, which could be useful for those who are seeking a high True Positive Rate.

In R, the Rule-based classifier is fairly simple to implement when using the "C5.0" package. The Rule-based classifier in this package seems to be used and accepted amongst statisticians that use R, as well. The package seems to contain the "C5.0" rule-based classification function that fits classification tree models and rule-based models using Quilan's C5.0 algorithm. Seemingly, the model results in an accuracy of 86% on the training dataset, before even performing cross validation. A single iteration results in a True Positive Rate equal to 87%, a True Negative Rate equal to 85%, a False Positive Rate equal to 15% , and a False Negative rate equal to 13%. For validation purposes, we used the "caret" package's Rule-based classifier to verify our findings, along with performing a 10-fold cross validation, as well. This approach resulted in different predictive measurements: an accuracy of 78%, a True Positive Rate equal to 77%, and a True Negative Rate equal to 78%. Therefore, the initial Rule-based classifier seemed to have been overfitting the data, since the results from cross-validation were somewhat worse.

In R, the Naive Bayes algorithm is easy to implement when using the "e1071" package. The Naive Bayes algorithm in this package seems to be widely used and accepted amongst statisticians that use R, as well. Although the implementation is simple, the model only results in an accuracy of 73% on the training dataset, before even performing cross validation. A single iteration results in a True Positive Rate equal to 75%, a True Negative Rate equal to 72%, a False Positive Rate equal to 25% , and a False Negative rate equal to 25%. For validation purposes, we used the "caret" package's Naive Bayes algorithm to verify our findings, and to perform a 10-fold cross validation. This approach resulted in a similar accuracy measurement: a True Positive Rate equal to 74% and a True Negative Rate equal to 73%.

In R, the Neural Network classifier is somewhat difficult to implement, but is considerably easier to use with the "neuralnet" package. In this package, the Neural Network classifier seems to be widely used and accepted amongst statisticians that use R, as well. The model results in an MSE of 0.18, and an accuracy of 75% on the training dataset, before even performing cross validation. A single iteration results in a True Positive Rate equal to 77%, a True Negative Rate equal to 72%, a False Positive Rate equal to 28% , and a False Negative rate equal to 23%. For validation purposes, we used the "caret" package's Artifical Neural Network algorithm to verify our findings, and to perform a 10-fold cross

validation. This approach resulted in a slightly worse accuracy measurement: an average accuracy equal to 70%.

In R, the Support Vector Machine classifier is easy to implement when using the "e1071" package. In this package, the Support Vector Machine function seems to be widely used and accepted amongst statisticians that use R, as well. The model results in an accuracy of 80% on the training dataset, before even performing cross validation. A single iteration results in a True Positive Rate equal to 82%, a True Negative Rate equal to 77%, a False Positive Rate equal to 23% , and a False Negative rate equal to 18%. For validation purposes, we used the "caret" package's implementation of a Support Vector Machine classifier to verify our findings, and to perform a 10-fold cross validation. This approach resulted in a similar accuracy measurement: an accuracy equal to 69%. This implies that our model fit during the preliminary analysis on the testing data contained a high level of overfitting.

In R, the Random Forest function is fairly simple to implement when using the "randomForest" package. In this package, the Random Forest algorithm seems to be widely used and accepted amongst statisticians that use R, as well. Before performing cross-validation, the model resulted in an accuracy of 99% on the training dataset, which certainly indicates a high level of overfitting. The single iteration results in a True Positive Rate equal to 100%, a True Negative Rate equal to 98%, a False Positive Rate equal to 2% , and a False Negative rate equal to 0%. For validation purposes, we used the "caret" package's implementation of the Random Forest function to verify our findings, and to perform a 10-fold cross validation in order to avoid overfitting. This approach resulted in a very different accuracy measurement, but the best accuracy measurement in comparison to the other classifiers. The average accuracy equaled to 83%.

5. Model Evaluation

In order to avoid overfitting throughout any analysis, cross-validation was key in iterating over testing and training data to provide the most correct predictive measurements. After cross-validation, the Decision Tree classifier had an accuracy measurement equal to 73%. This accuracy measurement had potential for improvement, especially when comparing the measurement to other predictive measurements from different classifiers. For example, the Rule-based classifier had an accuracy of 78%. Although the Rule-based classifier had a better accuracy measurement, the majority of other tested classifiers had a worse accuracy measurement in comparison to the Decision Tree classifier, after performing cross-validation. The Naive Bayes classifier produced an accuracy of 73%, the Neural Network classifier had an accuracy equal to 70%, and the Support Vector Machine had the worst accuracy measurement: an accuracy of 69%. However, the classifier with the best accuracy measurement was the Random Forest classifier, which had

an accuracy of 83% after cross-validation. Therefore, the Random Forest classifier should generally be used for predictive purposes, since it seems to possess the greatest predictive power, generally. Although these classifiers generally did not produce very good accuracy measurements, some classifier could be preferred over others, and even the Random Forest classifier, depending on the goal during analysis. For example, someone could reasonably prefer the Decision Tree classifier if the goal is to predict the highest True Positive Rate, since the Decision Tree classifier supports the greatest True Positive Rate. Essentially, the preferred classifier can be modified or switched for another, depending on the overarching goal throughout the analysis.

6. References

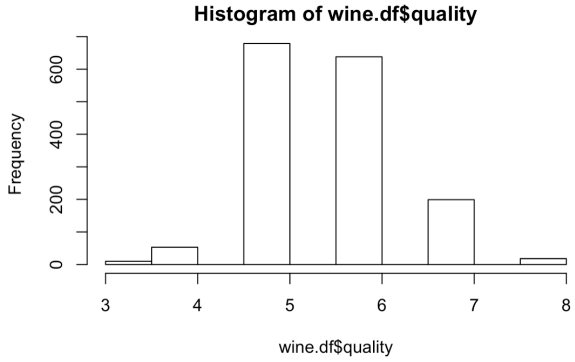

Figure 1: Distribution of wine quality

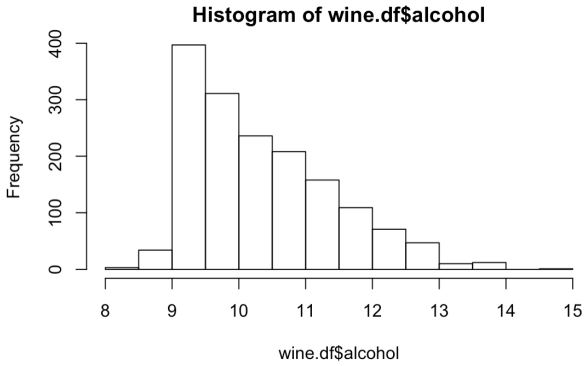

Figure 2: Distribution of alcohol content

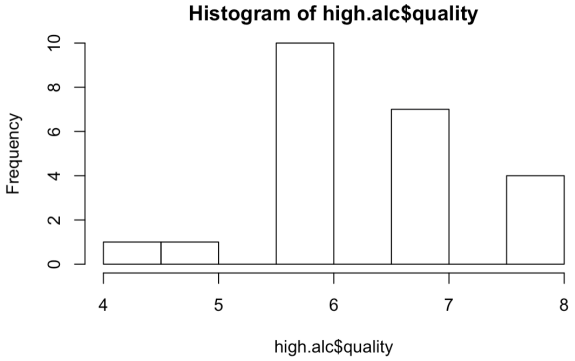

Figure 3: Distribution of quality of wine containing

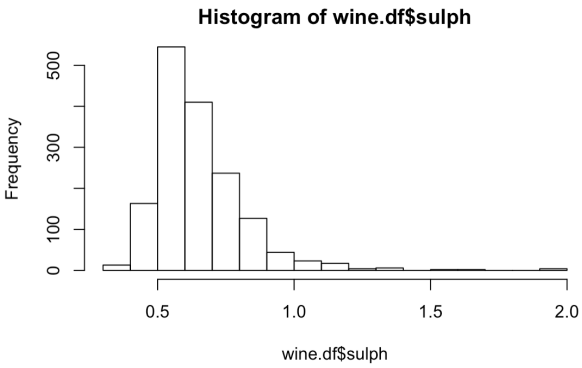

Figure 4: Distribution of sulphate content

Figure 5: Distribution of total sulphur dioxide

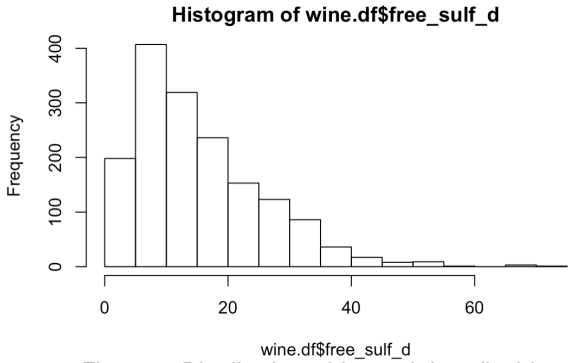

Figure 6: Distribution of free sulphur dioxide