**What makes Wine Good or Bad?**

Jameson Buie, [jbuie@bellarmine.edu](mailto:jbuie@bellarmine.edu)

**ABSTRACT**

There are many large businesses whose primary function is production of alcoholic beverages. Kentucky is bourbon country, but I am using a dataset on wine instead to conduct my project. There are many qualities that make up wine and decide if it is good or bad. This dataset has about 1600 different measurements of wines and I will be using logistic regression modeling to do a predictive analysis on whether a wine will be good or bad given its qualities. I will be splitting the dataset into training and testing groups and seeing if the model can correctly predict if the wine is deemed either good or bad.

1. **INTRODUCTION**

The output variable in this dataset is wine “quality”. My goal is to create a predictive model that can determine if a certain wine is good or bad, given its qualities which are the input variables. I will be using logistic classification to predict a discrete class that the wine belongs to. The two classes are good or bad.

1. **BACKGROUND**

This dataset was created by the UC Irvine Machine Learning Repository. It includes samples from different wines in the north of Portugal. Its intended use was to practice machine learning and create a reliable model that can predict wine quality based on several physiochemical tests. Overall, the creators of the dataset wanted to create an organized, clean dataset to perform machine learning on that would result in an accurate predictive model.

1. **EXPLORATORY ANALYSIS**

This wine quality dataset contains 1599 samples with 12 columns of mostly integer data types, with the output value being an object type because it is a class.

**Table 1: Data Types**

|  |  |
| --- | --- |
| *Variable Name* | *Data Type* |
| fixed acidity | Float64 |
| volatile acidity | Float64 |
| citric acid | Float64 |
| residual sugar | Float64 |
| chlorides | Float64 |
| free sulfur dioxide | Float64 |
| total sulfur dioxide | Float64 |
| Density | Float64 |
| pH | Float64 |
| sulphates | Float64 |
| alcohol | Float64 |
| quality | object |

*A pie chart of quality counts distribution

Description automatically generated*

**A table with numbers and symbols

Description automatically generated with medium confidence**

**A graph of a graph of different types of substances

Description automatically generated with medium confidence**

**A graph of blue dots

Description automatically generated**

1. **METHODS**
   1. *Data Preparation*

All the independent/explanatory variables in this dataset were crucial in predicting the output variable which was wine quality. There was no need to drop any variables or not include them in the logistic regression model because they did not overpower one another. One thing that needed to be changed was the data type of the dependent/output variable. Wine quality was either “Good” or “Bad”, but this would have caused major problems in the logistic regression model. Since this was a discrete, categorical variable. I chose to assign the number 1 as the quality class for “Good” and 0 for the ones in the “Bad” quality class. This was very quick and efficient to do because I implemented a single line of code that would create a dictionary with either ‘Good : 1’ or ‘Bad : 0’ as the keys and values and then swapped them in the table. This changed the output variable to integer type and thus allowed me to properly create a logistic regression model.

* 1. *Experimental Design*

Table X: Experiment Parameters

|  |  |
| --- | --- |
| **Experiment Number** | **Parameters** |
| 1 | The first test used all independent variables as indicator values and the dependent variable as the response. The training test was split into 70% of the original data, and the remaining 30% was the test data. This resulted in a 75% accurate model according to the classification report. |
| 2 | The second test used all independent variables as indicator values and the dependent variable as the response. The training test was split into 80% of the original data, and the remaining 20% was the test data. This also resulted in a 75% accurate model according to the classification report. |
| 3 | The third logistic model was creating using only specific variables that R deemed as statistically significant as predictor values. The training set was split into 80% of the original data, and the remaining 20% was the test data. My reason for splitting the data this way is because I hypothesized that using only statistically significant explanatory variables would create a more efficient and accurate model. This resulted in a 76% accurate model, which is the most accurate of the three models that I created. |

* 1. *Tools Used*

The following tools were used for this analysis: Python v3.5.2 running the Anaconda 4.3.22 environment for Apple Macintosh computer was used for all analysis and implementation. In addition to base Python, the following libraries were also used: Pandas 0.18.1, Numpy 1.11.3, Matplotlib 1.5.3, Seaborn 0.7.1, SKLearn 0.18.1. I chose these tools because we were required to do a machine learning project in Python. I used the Pandas library because it allowed me to quickly access and manipulate data in the very large data frame that I used. Numpy was also necessary for creation of the data frame, as you cannot create 2D arrays without the utilization of Numpy. Both Matplotlib and Seaborn were used for data visualization purposes. They were both used in creating graphs that showed correlation, interaction, and distribution of the different variables in the dataset. SKLearn was the most important package that was used. It would have not been possible for me to create a Logistic Regression model in Python without me having this package installed. I was able to use their LogisticRegression tool to create the model, and also utilized their Confusion Matrix and Classification Report to test the accuracy of my model.

1. **RESULTS**
   1. *Classification Measures/ Accuracy measure*

|  |  |
| --- | --- |
| **Experiment Number** | **Accuracy Measurements (Classification Report)** |
| 1 – Using all independent as predictors. 80/20 Split |  |
| 2 – Using all independent as predictors. 70/30 Split | A number of numbers in a row  Description automatically generated with medium confidence |
| 3 – Using only statistically significant variables as predictors. 80/20 Split | A number of numbers in a row  Description automatically generated with medium confidence |

* 1. *Discussion of Results*

Experiment number 3 provided the best classification. This one was the best because it used less predictor variables and had a higher accuracy percentage than the other two according to the classification reports. I believe this one was my best model because I used only statistically significant variables as predictor values. My worst model was Experiment number 2. I believe this one was the worst because it only used 70% of the data from the dataset to train the model, while the other two tests used 80% of the data from the model. It had less information input and was not able to predict as effectively as the other two.

* 1. *Problems Encountered*

One problem that I encountered was that my response variable was not of the integer data type. When doing any type of linear modeling, whether that is classification or regression, you need to use integer typed variables when training and testing the model. It will not be possible to run the model if there are objects in the dataset. I was able to fix this by assigning a binary number to the classes that were object types in the dataset, and this made the model production much more effective and it ran smoothly.

* 1. *Limitations of Implementation*

One limitation that my models possess are that continuous, numerical variables are good for predicting a continuous response variable, but not determining what class an item with certain attributes belongs to. If the response variable (wine quality) was different from “Good” or “Bad”, maybe rating the wine on a scale of 1 to 10, a linear regression model would have been much more accurate than using the same predictor values and determining the class, not quality rating, of the wine. Overall, it is very difficult to decide of a wine is good or bad based on many qualities that are all very similar and do not have large ranges. This may be why the models were only around 70% accurate.

* 1. *Improvements/Future Work*

In the future, I would have hundreds of people test the wines that are in the dataset and have them rate the wine on a scale of 1 to 10. Then, I would take the means of these ratings and use them as the quality rating in the dataset. I would create a linear regression model instead of a logistic regression model and try to create a model that would predict the ratings/quality of a wine based on the attributes that it has.

1. **CONCLUSION**

Finish up with a paragraph or two of summarizing your problem, the results and your conclusions (good model, bad model, needs more work, etc.).

In conclusion, I created a somewhat useful model that can be used to predict whether wine is good or bad based on certain attributes that it has. There is not much correlation between many of the independent variables, as shown on the pair plots and the correlation heatmap. Overall, the model was not bad but not exceptional. I was not completely satisfied with the models averaging around 75% accuracy, but it is hard to create an extremely accurate classification model when the subjects are being predicted into only two classes. As said before, a linear regression model would probably be more accurate and response variables being ratings instead of classes would be better. Overall, I learned a lot about logistic modeling during this project and would like to find a dataset with more discrete independent variables, instead of all the independent variables being continuous.