CS4300 - Project Proposal

Andrea Smith

October 2, 2019

Contents

List of Tables

List of Figures

1 Dataset

The dataset for this project applies a rating to wine quality based on physico-chemical inputs, such as acidity and alcohol content. By applying a regression analysis to the dataset, we can determine if the quality of the wine has a relationship with its physical and chemical properties. The data all relates to a specific type of wine, but without the specifics of the type of grape used, the brand, the price, etc.

The dataset can be found here:
https://www.kaggle.com/uciml/red-wine-quality-cortez-et-al-2009/version/1

2 Output Data Visualization

The output data is a numeric rating of the wine quality. Within this datset, the output ranges from 3 to 8 and is represented as integer values. The lower values represent poor quality wine, while higher values represent excellent quality. Normal or average wines will measure in the 5 or 6 rating. The ratings in this dataset are not uniformly distributed, as average and normal wines are over-represented within the dataset. However, the distribution is within the standard bell-curve that would be realistic.

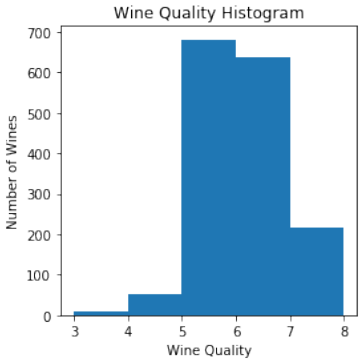

Figure 1: Wine Quality Distribution

3 Input Data Visualization

From a visual representation, there is no distinct relationship between a single input variable and the output variable.

Table 1: Input Feature Statistics

	Min	Max	Median	Mean
Fixed Acidity	4.60	15.90	7.90	8.32
Volatile Acidity	0.12	1.58	0.52	0.53
Citric Acid	0.00	1.00	0.26	0.27
Residual Sugar	0.90	15.50	2.20	2.54
Chlorides	0.01	0.61	0.08	0.09
Free Sulfur Dioxide	1.00	72.00	14.00	15.87
Total Sulfur Dioxide	6.00	289.00	38.00	46.47
Density	0.99	1.00	1.00	1.00
pH	2.74	4.01	3.31	3.31
Sulphates	0.33	2.00	0.62	0.66
Alcohol	8.40	14.90	10.20	10.42

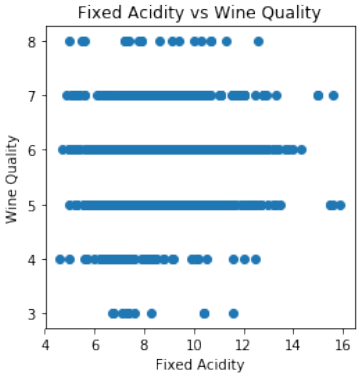

Figure 2: Fixed Acidity vs Wine Quality

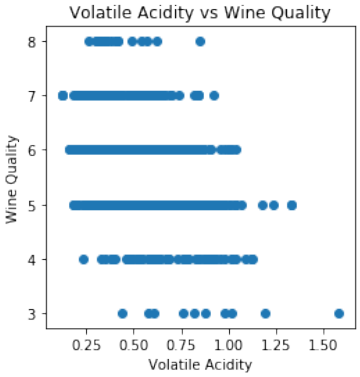

Figure 3: Volatile Acidity vs Wine Quality

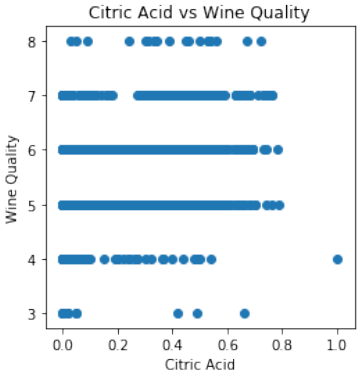

Figure 4: Citric Acid vs Wine Quality

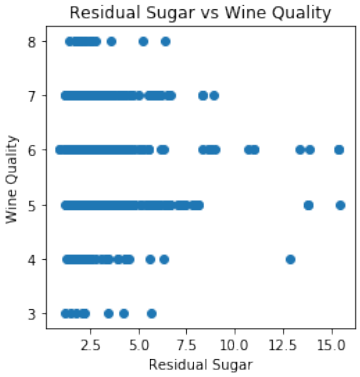

Figure 5: Residual Sugar vs Wine Quality

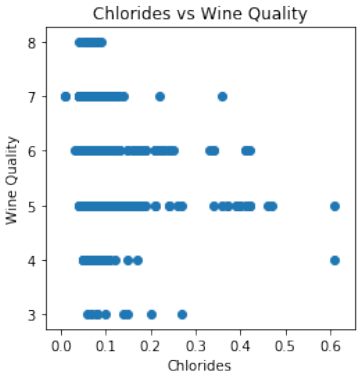

Figure 6: Chlorides vs Wine Quality

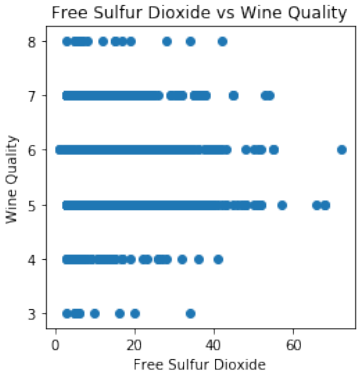

Figure 7: Free Sulfur Dioxide vs Wine Quality

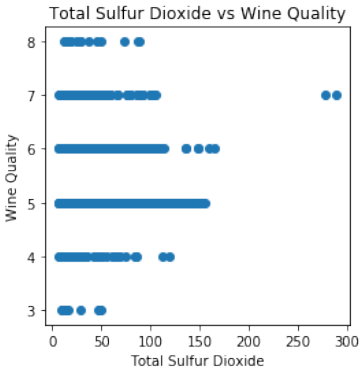

Figure 8: Total Sulfur Dioxide vs Wine Quality

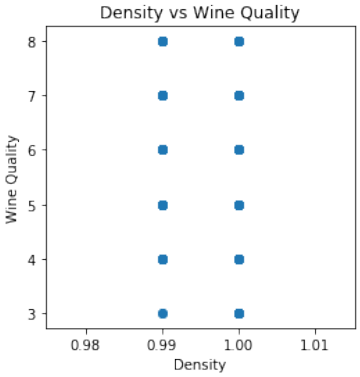

Figure 9: Density vs Wine Quality

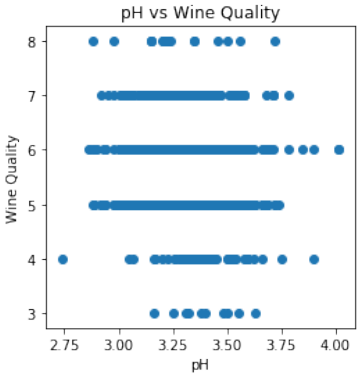

Figure 10: Ph. vs Wine Quality

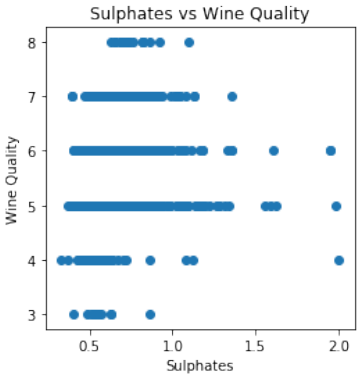

Figure 11: Sulphates vs Wine Quality

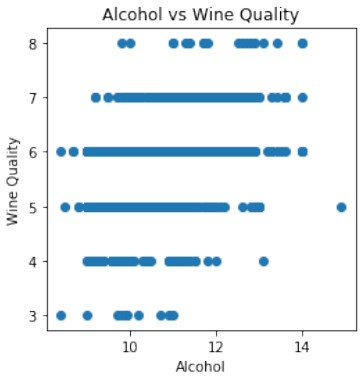

Figure 12: Alcohol vs Wine Quality

4 Data Analysis

4.1 Data Balancing

In order to attempt to balance the data, more wines of both low and high need to exist in the data. While we could remove data entries of the more 'average' wines, the dataset is not large enough to support the operation. Rather, we will generate more entries of both low and high quality wines. Generation is done after splitting the training and validation sets, and then only in the training set. The algorithm used for this is a nearest neighbor algorithm and is provided by the Python imbleam.SMOTE package.

4.2 Normalization

In order to normalize the data, each of the input variables will be normalized based on their range, known as Min-Max Normalization. For this method, the following equation can be employed on each input variable:

$$x^{'} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

This method will bring all data into the range [0, 1], so that the coefficients are representative of the potential of each input variable to change the output. If this method is insufficient, then traditional standardization can be used instead:

$$x^{'} = \frac{x - \overline{x}}{\sigma}$$

4.3 Data Splitting

The data consists of 1600 entries to be split across the training data, validation data, and test data. 800 entries will be used for training, 400 entries will be used for validation, and the remainder will be used for testing. After splitting, more entries will be generated in the training set. This pushes much of the data into the training phase, but due to data generation to help with the imbalance, we keep a keep a significant chunk separate for verification and testing.

4.4 Neural Networks

We will start with a feed-forward network based on multivariate logistic regression. Our loss function will be mean squared error, which will also be used to help evaluate the test set results. The network will contain an input node for each input feature and a singular output node for the wine quality. We will start with a single hidden layer in the model.

4.5 Model Evaluation and Improvement

The test set will use mean squared error, a precision metric, a recall metric, and a F_1 score to evaluate how well the model represents the data and it's ability

to accurately predict the outputs. A low mean squared error between predicted and actual outputs of the dataset implies that the model is an accurate representation of the dataset. A low precision score indicates a lot of false positives and a low recall score indicates a lot of false negatives. The F_1 score is a weighted average of the precision and recall score, such that a high value indicates model accuracy.

To improve the model, we can change the data normalization process to a Z-score (standardization) method. This method is more complex than the aforementioned Min-Max method, but is also more widely used in machine learning. Additionally, we can move more of the data into the training and validation sets, and a smaller proportion in the test sets. Lastly, we can increase the number of layers in the neural network.

5 Peer Review

5.1 Feedback Received: Christopher Cruzen

> The Dataset section provides a clear description of the goal of the project, the dataset source, and states that regression will be the fit method employed. Great introduction!

There is no action to address.

> The Output and Input sections do a good job of covering each of the model's features with helpful visualizations. It may also be worth briefly discussing the distribution of each input variable independent of wine quality here. For example, it appears that wines rich in sulfur dioxide are under-represented in this sample.

The distribution of each input feature is displayed in the table.

> Discussion of both data normalization and splitting is specific and clear.

There is no action to address.

> The Neural Networks section may benefit from some discussion of the structure of the model. For example, how many input and output nodes will be used? Will there be hidden layers? If so, how many and will these be adjusted with testing? If a more specific discussion of model structure is introduced, a simple layout visualization could also benefit the reader.

Information about nodes and layers was added to the section.

> Model evaluation and improvement techniques are described with good detail.

There is no action to address.

5.2 Feedback Received: Brett Schlereth

> The only thing I would suggest is possibly considering a logistic based regression instead of linear. It may increase the complexity of the algorithms, but I think it will more than likely generate better results.

The model was changed to logistic regression.

5.3 Feedback Received: Bikash Shrestha

> Need some balancing for the output labels because it is imbalanced. No. of wines for rating 3 is far less than no. of wines for rating 5. Provide a method of how you are going to balance the labels.

The section for data balancing was added and the model evaluation metric was changed.

> Kudos! for showing relationships between input features and output.

There is no action to address.

> Why you are planning to use min-max normalization? Better to provide some more insights.

Included information about why a min-max normalization and an alternative method to improve the model.

> Data Splitting (my suggestions): Go with 70% training, 20% validation and 10% testing. (If you want to keep this for future improvement, then it's fine to go with your current split method)

The data splitting was adjusted slightly and discussed more.

5.4 Feedback Given

Feedback sent to Amy Seidel on September 30th, 2019.
Feedback sent to Allison Chan on September 30th, 2019.