**Machine Learning Analysis of the Quality of Red and White Wines**

**Authors: Jordan Eisenman, Cameron Abernethy**

**Date: 2/23/2022**

Objective Description -

As a group, Cameron and Jordan were given two data sets concerning red wine and white wine. We sought to create models to predict a particular wine's quality, as rated by sommeliers, based on various factors – including but not limited to alcohol, acidity, and sugar content.

Results -

In the end, we implemented two different types of models, a linear and multivariate model. The following data was our final residual sum of squares for the top three predictors in red and white wines, respectively:

Linear Models:

Red Wine

* Alcohol: 0.792
* Volatile Acidity: 0.868
* Density: 0.944

White Wine

* Sulfates: 84
* Alcohol .89
* Volatile\_acidity: 0.77

Analysis -

The linear models were moderately effective in predicting a wine's quality. Citric acid was the strongest predictor of any features. We made multivariate models with the strongest predictors in the single-dimensional linear case, given these results. We have the following results:

Multivariable Models:

* Alcohol, Density: 0.232
* Alcohol, Volatile Acidity: 0.317
* Density, Volatile Acidity: 0.180

White Wine

* alcohol, volatile\_acidity 0.27
* sulfates, volatile\_acidity 0.08
* alcohol, sulfates 0.19

Analysis -

The multivariate regression was much more effective in predicting a wine's quality for red and white wines. As seen from the data, the most effective variables in predicting the quality of red wine were Density and Volatile Acidity. The most effective variables in predicting the quality of white wine were sulfates and volatile acidity. Based on these results, we proceeded to combine each of the most effective variables in hopes of achieving greater accuracy through multivariate regressions.

Data exploration -

We went through the long and arduous exploration of the data sets to achieve our objective. Keeping the data in mind, we built the requisite linear and polynomial regression models by hand and used the pre-built python packages for multivariate-linear regression. We then tested our intuition against the data – we tested the accuracy of each model, calculating the average of the residual sum of squares.

Method -

Further exploration of the data required building requisite models to test our hypotheses. Thus, we built our own models to test our hypotheses. We started by implementing gradient descent for a linear model. Our model acted erratically, but we resolved these issues, creating methods to normalize our data. We proceeded to implement gradient descent on our linear model. Building on our linear model, we then implemented our quadratic regression, with only a few changes to our original code. Finally, we saw an opportunity to generalize our stochastic descent method for any polynomial. The regression model worked as intended; however, it exhibited more erratic behavior. This problem was especially true when we tested the regression model on higher-ordered polynomials. For example, cubic functions became increasingly challenging for the computer. Not only did the regression take exponentially more time, but the line would also frequently change, as the stochastic gradient descent would output the wrong betas. Implementing our three-dimensional multivariate regression model, we used the statsmodels python package. Cameron also created some amazing three-dimensional

figures using matplotlib.

Pre-processing -

Building the models also required processing our data and fine-tuning our hyper-parameters. In pre-processing, we replaced the spaces in the column names with underscores. The data needed to be normalized before being put into our model. We created our own model that normalized the data in each column before placing it into our models. We found that tolerance and learning rate significantly affected our models concerning our hyper-parameters. In particular, a tolerance of .01 resulted in extremely quick but inaccurate models. In most cases, we used a tolerance of 0.0001, which yielded relatively quick and accurate models. Concerning the learning rate, we generally used 0.001. Similar to tolerance, this learning rate yielded quick and accurate models. To be sure, these parameters struggled with a cubic model, as it took fundamentally more computing power.

Ethics -

In terms of ethics, we understand the term "ethics" in two ways: (a) what is the morality of such a model? and (b) how should we act knowing what we know? We took issue with the conclusion that more alcoholic wines are generally better. Not only is this conclusion biased – these professional reviewers drink alcohol for a living – but it could also lead to harm. More alcoholic wine may harm consumers because higher amounts of alcohol have been associated with health issues in humans. Being aware of this bias, we should treat these models for what they are: gauges of how professional wine drinkers enjoy their wine. Meaning a sommelier's experience may differ from that of the average person, thus causing an average person to rate the quality of a particular wine differently from the sommelier.