Predicting Wine Quality

Mackenzie Carey and **Laura Hiatt**
{macarey,lahiatt}@davidson.edu
Davidson College
Davidson, NC 28035
U.S.A.

Abstract

Wine quality is measured through sensory input. Given a set of scientific measurements of wine attributes, our experiment aims to see if through k-fold cross validation with ridge regression and standard lasso regression models we can estimate the quality value that experts would give to an unkown wine. Through our experiments, we discovered that our models yield results that are better than those produced from predicting the average wine quality score of all of the given samples for each wine in a given data set. These results are concluded from the results of measurements such as coefficients of determination, mean average errors and mean squared errors.

1 Introduction

Given a set of data points, regression in its simplest form attempts to determine a line of best fit which describes the relationship between an input and an output variable. However, a straight line may fail to accurately represent the data. A better representation may be through the form of a higher order polynomial, which may be found through different forms of regression. In order to achieve a more accurate model of the data, there are various approaches with which to analyze the mathematical representation of any data set.

In our experiment, we utilize a random search k-fold cross-validation implementation of stochastic average gradient descent to determine a relationship between 11 attributes of wine, as displayed in Figure 1, and a quality score. The quality is based on the sensory score determined by the average taste rating of at least three different wine experts. Scores range between one and ten. A score of one represents a lower quality wine, while a score of ten distinguishes the highest wine samples. The 11 attributes are ones of scientific measure.

The evaluation runs two separate but identical tests on the red and white wine selections of Portugese "Vinho Verde" wine. The tests were preformed on red and white wine separately to make sure that the data was not compromised. It is important to note that there is a large difference in the number of data points, with white wine have more than twice the amount of data. If the sample were run together, this would have skewed our results and not been an accurate assessment on wine quality. Previous experiments have been run on this same data set to form predictions about wine quality using data mining techniques. Three separate techniques were preformed and were compared with each other based on efficiency. The method deemed the best was support vector machines, which was shown to outperform a multivariable regression. (P. Cortez 2009).

In the remainder of this paper, we discuss our analysis of the data and representations given by our experiment.

Feature
fixed acidity
volatile acidity
citric acid
residual sugar
chlorides
free sulfur dioxide
total sulfur dioxide
density
pH
sulphates
alcohol

Figure 1: Objective attributes of wine.

2 Background

Stochastic gradient descent is a common algorithm in the field of machine learning and thus, there are multiple off-the-self regression solvers which utilize this approach available for use. Examples include software such as Matlab or, as used in this experiment, the scikit-learn library for Python.

The scikit-learn library provides variations of gradient descent along with a range of other methods such as ordinary least squares and logistic regression. The particular regression solver used in this experiment is a RandomizedSearchCV with a Ridge estimator object. In other words, we invoke a randomized search of hyper parameters, "free variables", while using a cross-validation scheme under the

constraints of ridge regression. Details of these particular features will be further discussed in the background.

One thing that our model shares with the other machine learning models is the use of a cost-function. Cost-functions vary from model to model, but they all penalize errors made in predictions. The predictions are made with a hypothesis function, $h(\theta)$, specific to the data under investigation. The value of any prediction can then be used in a cost function to determine the quality of the model. Only penalizing for error, the cost function is represented as:

$$J(\vec{\theta}) = \frac{1}{2m}[\sum_{i=1}^{m}(y^{(i)} - h(\theta))^2]$$

However, as representations add higher-order terms, the cost function may be augmented to penalize for complexity as well.

Regularization

Our initial model uses L_2 Regularization, otherwise known as ridge regression. Regularization inhibits models from becoming too complex. The complexity of an equation is determined by the order of the polynomials present and the size of coefficients; smaller coefficients reduce the impact of higher order polynomials on the curve and thus reduce drastic changes in the value x_2 based on changes in x_1.

The added concern for complexity calls for a change in the cost function. Under the experiments carried out in this project, the cost function becomes:

$$J(\vec{\theta}) = \frac{1}{2m}[\sum_{i=1}^{m}(y^{(i)} - h(x^{(i)})^2 + \alpha||\theta^2||]$$

where α represents the regard for complexity. A higher α value reduces complexity by driving down the weights attributed to higher older polynomial values (Pedregosa et al. 2011).

Why worry about complexity? The higher order a polynomial, the worse the function generalizes to new data points. A higher order polynomial may fit exceptionally well with test data, but poorly predict new values. This is called overfitting. On the other hand, there is a risk of underfitting data. When the model underfits data, the model poorly predicts values in all of the training, validation, and test sets. An example of overfitting and underfitting data sets can be seen in Fig 2. A good fit is usually determined to have both low complexity and low error.

A Variant

Another type of Regularization is known as L_1 Regularization. While L_2 Regularization squares the value of the θ value, L_1 takes the absolute value of θ and multiplies it by α. This is expressed by the following equation:

$$J(\vec{\theta}) = \frac{1}{2m}[\sum_{i=1}^{m}(y^{(i)} - h(x^{(i)})^2 + \alpha||\theta||]$$

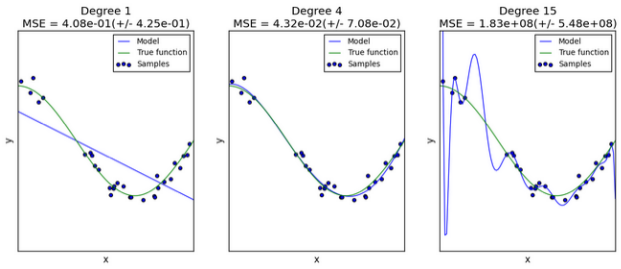

Figure 2: L: Underfitting, M: Neither underfitting nor overfitting, R: Overfitting. Image courtesy of `scikit-learn`.

However, gradient descent cannot be used with L_1 since all parameters would shrink to a small size. This type of regression is known as Lasso Regression. Another feature of Lasso Regression, is that it is a sparse model. In other words, some of the final weights given to certain attributes in the final model used to predict new values. Some of the values shrink to 0, meaning only some of the features used to predict the final x_2 value will be utilized. In our investigation, we compare R^2 scores from our primary results using the RandomizedSearchCV method and ridge regression model to those produced by a lasso regression model, Lasso. (Pedregosa et al. 2011)

Cross-Validation

Cross-validation splits a set of data into a number of distinct and separate groups, termed k-fold. Each of these groups is then cycled through to represent the validation set and test sets. The validation error across these different groupings of data is then averaged together to create the final validation error. Overall, cross-validation acts to better generalize the decision function since it is tested on more than one specific data set. The model is exposed to different combinations of data points and is thus less likely to overfit to one group of test data.

Grid-Search versus Randomized-Search

We will now discuss the process with which hyper parameters for the equation are chosen. Two common approaches are grid-search and randomized search. Grid-search is the most commonly used to optimize parameters (Pedregosa et al. 2011). Parameters are chosen by systematically exploring a range of possible combinations of all parameters. This ensures that a wide range of combinations of parameters are tested. Meanwhile, random search, the search type investigated in this project, randomly (as the name suggests) chooses the set of parameters. Using at least 60 random sets of hyper parameters (we use 100 in our experiments) we can with 95% confidence get reasonably close to the optimal hyper parameters.

Further discussions of all types of models offered by the scikit-learn can be read in further detail at (Pedregosa et al. 2011).

3 Data Preparation

Before we ran our experiments, we added features to the given data that were the results of squaring and cubing the values of of the 11 objective features for a total of 34 possible θ values including the addition of an x_2-intercept. Thus, our models could be polynomials up to the third degree.

Additionally, using built-in functions in the scikit-learn and numPy packages, our data was also normalized before the experiments were undertaken.

Data was separated into training and validation sets using 80% of the provided data points. A hold-out set containing 20% of the wine scores was used as a test set to report all errors and scores.

4 Experiments

Our experiment was designed to develop a model which could predict quality ratings of Portugese "Vinho Verde" wine based on the 11 features depicted in Fig 1. All the data points provided by (P. Cortez and Reis. 2009) were restricted to two categories: white wine and red wine. Regression tests were run on these two categories separately.

As stated earlier, we chose to investigate the model produced by scikit-learn's RandomizedSearchCV model using ridge regression. The Ridge model performed it's computational routines using Stochastic Average Gradient descent. The drawback of this form of gradient descent is that without similarly scaled data, fast convergence on a model is not guaranteed (Pedregosa et al. 2011). A common practice of feature scaling, or normalizing, is used to fix this problem. To combat this potential error, all feature data was normalized before testing began, as stated earlier.

The estimator used by our final model was, of course, a Ridge object. We specifically looked at the outcomes of different magnitudes of α ranging from 10^{-7} to 10^{3}. Rather than using the default 3-fold cross-validation, we implemented 10-fold cross-validation. A larger value of k, in k-fold cross-validation reduces the variability of the test data. Our reasoning was that while the model would have less data to learn on at a time, the model produced would be less susceptible to overfitting the test data.

On the other end of the spectrum, we also implemented a Lasso regression test. Lasso Regression does not utilize stochastic gradient descent. However, the model still requires an α value to multiply by the L1 score to penalize the cost function. The alphas tested for the Lasso model covered a much different range, requiring much lower α values. The lasso values ranged from 10^{-16} to 10^{0}.

The data reported are the averages computed for each set of parameter values over 500 trials.

5 Results

In order to conclude that our models actually learned, we needed to compare our results to a baseline. We produced three measures with which to make these comparisons: the coefficient of determination, mean absolute error, and mean squared error.

The coefficient of determination is also known as the R^2 score. This metric returns a value between -1.0 and 1.0. A score of 0.0 denotes an equivalent quality to that of a model which reports the expected value of the data set, regardless of the features, for every prediction (Pedregosa et al. 2011). A negative score is worse than a score of 0.0 and a positive score is better. The average scores for white wine and red wine were 5.880 and 5.664 respectively. From this score alone, we know that both of our models predicted the correct wine score as determined by professional wine tasters better than a model which only reports the average of a data set would.

With an α value of 0.0001, our randomized search cross-validation model which used ridge regression produced received an R^2 score of 0.2308. This validates that our model did indeed learn how to predict wine scores with some accuracy. A graphical representation of the relationship between the α values and the R^2 score for this method can be seen in Fig 3. Additionally, our lasso model performed better than the baseline R^2 score of 0.0 as well. The coefficient of determination for lasso regression was 0.2139 for an α of 10^{-16}, the smallest tested α. In general, smaller α values constructed more accurate models for our lasso regression model. However these gains were at the loss of consideration of certain features. The graph of α values for the lasso model versus the coefficient of determination can be seen in Fig **??**.

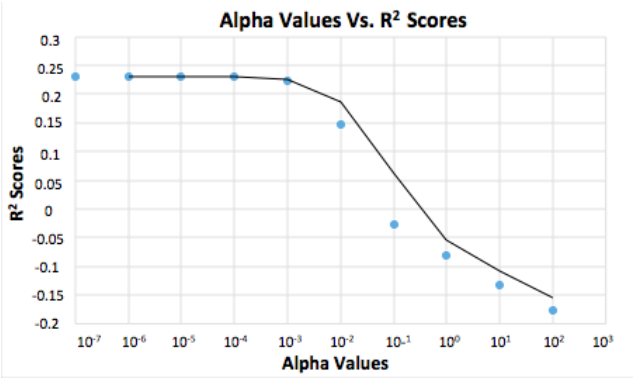

Figure 3: The graph that represents α versus R^2 score. The same score was received for both white and red wine.

For our primary model, the randomized search k-fold cross validation model, we report the mean absolute error and the mean square error. A comparison of these errors versus α values can be viewed in Fig 5. These scores also tell us that our model is doing better than the baseline. If all

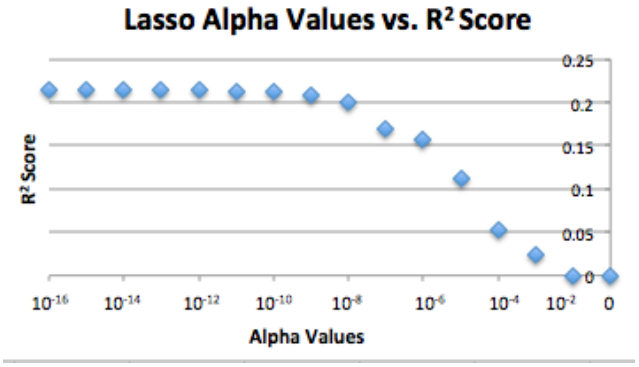

Figure 4: The correlation between α and R^2 score for lasso regression.

values in the data set were predicted to be the value of the average of the data, then the calculated mean and absolute squared errors would be higher than those found in our ridge regression model. The calculated errors can be viewed in Fig 6.

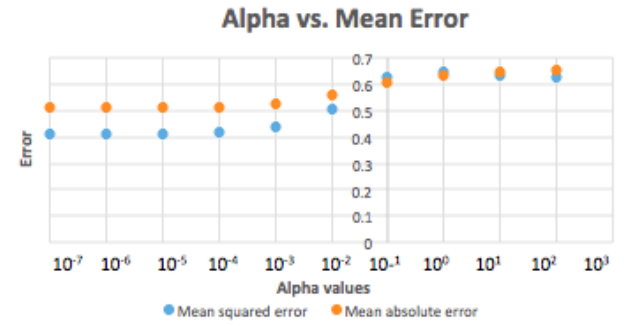

Figure 5: α versus mean square and absolute errors for ridge regression.

	Mean Squared Error	*Mean Average Error*
White	0.6008	0.5682
Red	0.6249	0.6521

Figure 6: Baseline error values for white and red wines.

At an α value of 0.001, our model produced a mean absolute error of 0.5135 and a mean squared error of 0.4148. These numbers are lower than those presented in Fig 6. We did not perform this set of experiments on the lasso regression, as it was a later addition to our experiments.

6 Conclusions

Our task was to take objective wine attributes of a certain wine and predict the score that wine experts would give based on its own features. These results validate our conclusions that our implementation of a k-fold cross validation model using ridge regression was successful. Our evaluation of mean errors and the coefficient of determination scores both implicate this conclusion. We still hold the same conclusions for our evaluation of lasso regression based on its coefficient of determination value. However, we do believe that the ridge regression model was more successful than the lasso regression model based on the R^2 scores produced.

7 Contributions

L.H. and M.C. paired programmed the scikit-learn python package. L.H. ran the ridge regression and lasso experiments and M.C. processed the csv files and created the graphs in the results. M.C. wrote the abstract and the introduction. L.H. wrote the background and experiments. M.C and L.H. wrote the data preparation section. M.C. wrote the results. Both proofread the entire document and added input to each section.

8 Acknowledgements

We would like to thank scikit-learn for their tutorials. We would also like to thank Dr. Ramanujan for helpful insight during office hours.

References

P. Cortez, A. Cerdeira, F. A. T. M., and Reis., J. 2009. Modeling wine preferences by data mining from physicochemical properties. *In Decision Support Systems* 47(4):547–553.

P. Cortez, J. Teixeira, A. C. F. A. T. M. J. R. 2009. Using data mining for wine quality assessment. *DBLP*.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; and Duchesnay, E. 2011. Skikit-learn: Machine learning in python. *JMLP* 12:2825–2830.