Predicting wine ratings using linear models

Dominic Rossi
A08435889
University of California, San Diego
dmrossi@ucsd.edu

ABSTRACT

In this paper I analyze reviews taken from a popular wine review website. I use linear regression to tune a model suited for predicting wine ratings. I compare a two models and select the best one with the best parameters using k fold cross validation. The results indicate that better predictions come from using more of the features, rather than less of them. This suggests that both the age of a wine as well as the varietal are important when predicting the rating of a particular wine.

1. INTRODUCTION

The era of big data has allowed for unprecedented analysis. Of the available data are various forms of online reviews. These massive online databases of product reviews enable researchers to find patterns in human behavior. Companies are interested in using these patterns to recommend the best products to their users. In order to achieve this goal, researchers use machine learning techniques on existing reviews in order to find the best products for users.

In this assignment, I examine data from wine reviews which were taken from *CellarTracker.com* [3]. From their homepage, "CellarTracker is the world's largest collection of wine reviews, tasting notes and personal stories from people who love wine." The dataset consists of over 2 million reviews spanning hundreds of varietals and thousands of users. Most of the reviews contain a score assigned to a particular wine, in the range of 50 to 100 points. I chose to predict this score based on certain features extracted from the dataset.

Traditionally, analysis of online reviews is used to recommend new products to users [1, 4, 2]. Rather than consider the task of recommending products to users, I chose to predict the ratings of wines based on certain features. In a way, it is similar recommender systems in that one could easily use it to predict which wines will be most highly rated. On the other hand, predicting the ratings of wines does not take into account any sort of user preference modeling.

One of the works that analyzed this dataset, [3], looked at how user ratings varied across users with more experience. Experience was gauged by the number of products reviewed. The users with more experience were considered experts and it was found that across several review datasets, experts tend to rate items more strongly than inexperienced users. Specifically, experts rated products that were below average lower than less experienced users. Similarly, experts rated above average products higher than less experienced users. These conclusions could be utilized by the owners of websites to fine tune product recommendation systems to users based on experience.

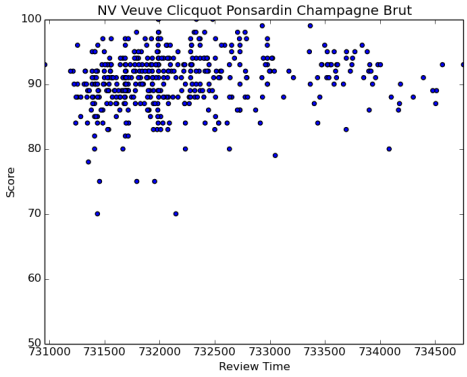
Since most analysis of user reviews is used to recommend products to new users, predicting rating is not an active area of research. As a result, state of the art methods for predicting wine ratings default to state of the art prediction techniques. Popular prediction techniques include linear regression, nonparametric regression, and Bayesian linear regression. Within linear regression, there are several models one could form. For this reason, as well as linear regression's well studied past, [5], I chose to focus on this method alone.

The rest of the paper is organized as follows: Section 2 describes the dataset and the relevant features for prediction. Section 3 describes the experiment performed on the dataset. Section 4 discusses the results of using the models for prediction. Section 5 summarizes the insights gained from the predictive task as they apply to this dataset and wine as a whole.

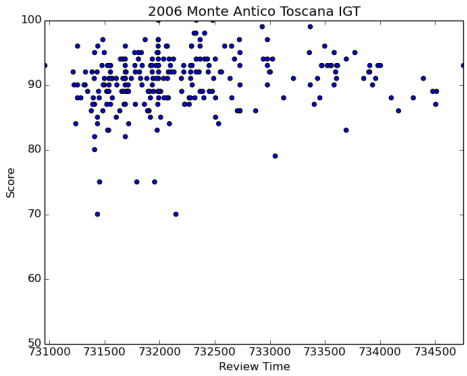
2. DATASET

The dataset used for prediction contains $2,025,995$ reviews taken from *CellarTracker.com*. Within the datset are fields for {wine name, points (score), wine varietal, user ID, username, review time, wine year, wine ID, and review text }. The number of unique reviewers in the dataset are $44,268$. There are $485,179$ unique wines spanning 830 varietals. The average wine score is 88.82 points on a scale from 50 to 100. Interestingly, only 77.48% of reviews have an associated score. The average review length, in words, is 36.0 words.

In order to avoid dealing with a natural language processing (NLP) problem, I chose to ignore the review texts altogether. Also, the name of a wine is probably rather useless when predicting the score. The exception to this might be a wine with an extremely off putting name, but I chose to ignore

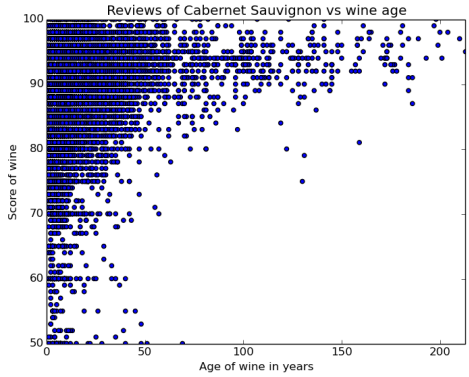


(a) Wine 1

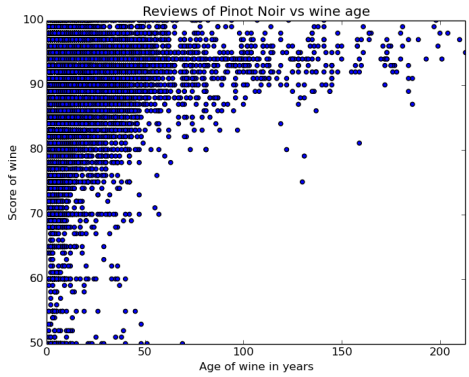


(b) Wine 2

Figure 1: Plots of two common wine reviews over time



(a) Varietal 1



(b) Varietal 2

Figure 2: Plots of two common wine varietals over the age of a wine

this possibility. Finally, since I am not modeling user preferences at all, I opted to discard any information related to users. After these decisions, the wine dataset can be thought of as only containing the following features: {points (score), wine varietal, review time, wine year, wine ID}.

Figure 1 shows plots of two of the most reviewed wines as a function of review time. It is hard to really extract any useful trends from this other than that it might slightly look like the wines tend to increase in rating over time. Using domain specific knowledge about wine, we expect that, over time, a wine's rating will tend to increase to a point and then decrease afterwards. It is difficult to see such a trend here though.

Figure 2 shows wines of a specific varietal plotted as a function of the wine age in years. Wine age is calculated simply by subtracting the review year from the wine year. This makes two assumptions: first that the reviewer writes the review during the same year in which they consumed the wine. This is probably a safe assumption. Second, this method assumes that all wines are released on January 1 of a given year. While this is certainly not the case, it is an assumption worth noting since there is no avoiding this issue, given the current dataset.

The plots in Figure 2 do seem to suggest that a wine will increase in review score over time. It does not appear however that the wine scores will ever decrease. This is likely a combination of two things: first users who purchase an old wine, likely at a higher cost, are more inclined through a buyer's guilt, to rate the wine more highly. Second, of the users who do not like the older wines, they too feel a sort of social pressure to rate older wines more highly. As a result of disliking a wine and heeding societal pressures, these users choose not to rate the wine. While neither of these speculations are proven, they serve as a possible explanation for the data shown in Figure 2. Regardless of the cause, it does appear that within a given varietal, wine ratings tend to increase over time to a certain point, after which they level out.

From the above analysis, I propose that a linear model, as a function of time and varietal, can be trained to predict the rating of various wines.

3. PROCEDURES

In the following subsections I outline the methods used in performing the experiments. This includes feature extraction, model selection, and practices utilized in performing the predictive task.

3.1 Preprocessing

One of the first thing needed in order to fit a parameter vector to the training data is a set of features. In Section 2

I alluded to the feature space comprising of the elements: {points (score), wine varietal, review time, wine year, wine ID}. In order to train linear regression for predicting wine scores, the first task is to convert any text-based features to numerical ones. I choose to convert wine varietal to a binary vector of length equal to the number of varietals. For each varietal, only one value in the feature vector is 1 while the rest are 0. Since there are over 800 varietals in the dataset, I opt to use only the top 10 varietals. Top 10 is defined as those varietals which contain the greatest number of reviews. By restricting the number of varietals I signifcantly reduce the dataset size and make the feature encoding of the varietal much more computationally feasible.

Of the remaining features, I choose to collapse review time and wine year into a single feature by taking the difference between the two. The result can be interpreted as the age of the wine in years. I choose not to do any further preprocessing on this feature, i.e. rescaling or mean subtraction. For the wine ID, I opt to discard it in the use of a linear model. Retaining this parameter would mean that a wine's score is at least partially dependent on its wine ID, which seems unlikely, especially considering the IDs are assigned by the website and not a third party.

The original dataset contains over two million reviews. I prune this by selecting only the reiews which are of wines in the top 10 varietals and contain all of the following fields: {points (score), wine varietal, review time, wine year}. There are also a handful of reviews whose wine age calculation result in a negative value. I opt to exclude these few reviews. The pruned dataset contains $984,337$ reviews.

3.2 Models

Linear regression assumes a predictor of the form

$$y = X\theta. \tag{1}$$

This can be solved for the parameter, θ, using the pseudoinverse of X:

$$\theta = (X^T X)^{-1} X^T y, \tag{2}$$

where $(X^T X)^{-1} X^T$ denotes the pseudoinverse of X. The problem with the above solution is that the value of θ may not generalize well to unseen data. To solve this problem, we can use a regularized linear regression:

$$\text{argmin}_\theta \frac{1}{N} ||y - X\theta||_2^2 + \lambda ||\theta||_2^2, \tag{3}$$

where λ is a regularization parameter that penalizes model complexity. By adding the additional term, we avoid creating a model whose performance, as a function of mean square error, increases as the dimensionality of θ increases. λ prevents overfitting on the training set by increasing the generalizability of the model to a test set. We use gradient descent to learn θ, since a closed form solution no longer exists:

$$\theta := \theta - \alpha f'(\theta), \tag{4}$$

where $f(\theta)$ is given by equation (3).

In order to examine the effect of wine varietal on predicting the score, I opt to create two models using linear regression. The first model contains both features: wine age and varietal. As a result, $X \in \mathbb{R}^{n \times d}$, where d is the dimension of the feature space and n is the number of examples in the dataset. For the first model,

$$d_1 = \underbrace{1}_{\text{intercept term}} + \underbrace{10}_{\text{varietal encoding}} + \underbrace{1}_{\text{age of wine in years}} = 12$$

and for the second model:

$$d_2 = \underbrace{1}_{\text{intercept term}} + \underbrace{1}_{\text{age of wine in years}} = 2.$$

3.3 Experiment

I first randomly shuffle the data and split the data into 90/10 training and test sets. The test set will be used for final evaluation and the training set will be used for hyperparameter tuning and model selection.

In order to select a value of λ, the regularization parameter, I perform k-fold cross validation on the training set and grid search over various values for λ. I set $k = 10$ and for each value of λ I train the data using $k - 1 = 9$ folds and evaluate using the reamining fold. For each value of λ, the model is trained k times and the result is the averaged mean squared error on each of the k validation sets. Proceeding in this way, I find that $\lambda = 0$ yields the best results, which implies equation (2) can be used. As a note, $\lambda = 0$ is likely partially due to the large size of the training set. Since the training set is so much larger than the validation set, the model is not required to generalize to a very large, new set. This means that the overfitting that might occur when using equation (2) diminishes as a result of the relative training and validation set sizes.

After finding the best λ for each model, I perform k-fold cross validation again in order to select the best model. Since training both models is quick, this turns out to be an irrelevant task. Instead it is more useful to train both models and compare their evaluations on the test set. Note though that the results from cross validation here will agree with the results on the test set.

In order to evaluate my linear models, I make use of both mean square error:

$$\text{MSE}(f) = \frac{1}{N} \sum_{i=1}^{n} (X_i \theta - y_i)^2, \tag{5}$$

and the coefficient of determination:

$$R^2 = 1 - \text{FVU}(f) = 1 - \frac{\text{MSE}(f)}{\text{var}(f)}, \tag{6}$$

where

$$\text{var}(f) = \frac{1}{N} \sum_{i=1}^{n} (\bar{y} - y_i)^2$$

$$\bar{y} = \frac{1}{N} \sum_{i=1}^{n} y_i.$$

The MSE gives a measure of how well the a model fits the data and the coefficient of determination is a measure of the proportion of variance explained by the model. As an additional note, it is possible for the R^2 value to be negative for a particular model. The interpretation here is that the model would be better off predicting the mean value rather

than the outcome of the prediction function. This occurs frequently during my search for the optimal hyperparameter λ, and is the reason the ideal value is likely not zero, but very small.

After finding the optimal value of $\lambda = 0$ using grid search and cross validation, I train both of my models using 90% of the data and test on the remaining 10%. Even though the k-fold cross validation results suggested that model 1 is to be preferred, I train both models and evaluate them on the test set.

4. RESULTS

The results of training both models is shown in Table 1.

	MSE	R^2
Model 1	16.2886	0.0364293
Model 2	16.4063	0.0294672
Mean	16.9044	0

Table 1: Results of training both models

As a reminder, model 1 corresponds to the feature matrix, X, containing both the varietal, the age of the wine and an intercept term. Model 2 consists of X containing only the age of the wine and an intercept term. From the table we can see that while model 1 has a lower MSE, model 2's R^2 value is lower. This illustrates the trade off between the two models: the better the fit of the model to the data, the worse the fit of the data to the model. The bottom row of Table 1 denotes the statistics of a mean-based predictor.

5. CONCLUSIONS

In this assignment I trained two different models for predicting the score of wine reviews. The model which made use of varietal information outperformed the other model which only took into account age of the wine. While performances look very close, it looks as though varietal is slightly informative when making wine review predictions. The results are not entirely conclusive due to the fact that the second model does better at explaining the variance of the data than does the first.

The nature of a linear predictor makes the model parameters quite easy to interpret. The model parameter associated with intercept essentially represents the average wine review. The model parameters associated with the wine varietals, if present, denote how many points should be added to or subtracted from a review based on its varietal. Finally, the model parameter associated with the age of the wine denotes how much the age of the wine should add to the overall rating of the wine.

If I had the opportunity to further explore the dataset, I would investigate the use of other features. One thing I explicitly avoided was dealing with the textual content of the reviews. By applying sentiment analysis to the reviews, I might be able to add another useful feature to the model and further lower the MSE. I would expect the text of the review to be the most important feature for predicting the score of a wine review.

While the results at first might seem disappointing, it is probably comforting, to wine connoisseurs anyway, that a wine cannot be judged on varietal and age alone. Furthermore, in addition to textual information, I expect that user information might further contribute to predictive powers of the model. By incorporating a user feature, the model could account for individual user biases when predicting wine ratings. Overall, the fact that wine age and varietal are poor predictors of a wine's rating is not a bad thing. This means that wines can be much better or worse than average, regardless of age and varietal. However, given age and varietal, one can train a linear predictor which outperforms a mean-based predictor.

6. REFERENCES

[1] A. Levi, O. Mokryn, C. Diot, and N. Taft. Finding a needle in a haystack of reviews: Cold start context-based hotel recommender system. In *Proceedings of the Sixth ACM Conference on Recommender Systems*, RecSys '12, pages 115–122, New York, NY, USA, 2012. ACM.

[2] G. Linden, B. Smith, and J. York. Amazon. com recommendations: Item-to-item collaborative filtering. *Internet Computing, IEEE*, 7(1):76–80, 2003.

[3] J. J. McAuley and J. Leskovec. From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. In *World Wide Web*, 2013.

[4] S. Rendle, C. Freudenthaler, Z. Gantner, and L. Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 452–461. AUAI Press, 2009.

[5] G. A. Seber and A. J. Lee. *Linear regression analysis*, volume 936. John Wiley & Sons, 2012.