Predicting Wine Points using sentiment analysis

Yik Lun, Peter, Chan
University of California, San Diego
ylc015@ucsd.edu

Kevin Gu
University of California, San Diego
kegu@ucsd.edu

Stephen Yang
University of California, San Diego
sgyang@ucsd.edu

Abstract

There is immense research being done in Data mining on the field of sentiment analysis to predict the rating given to item. In this article, we will discuss the different methods like linear regression and logistic regression we used to predict the rating a reviewer will give to a wine. We will discuss the exploration of data, different approaches we took and results of these approaches.

1. Introduction To Data Set

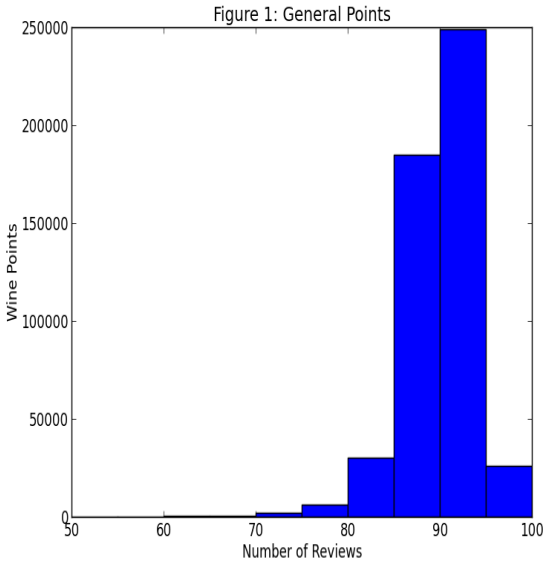
As a group we decided to use the wine reviews. There are a total of 2,025,995 wine reviews, 44,268 users, 485,179 wines, 5,957 users with more than 50 reviews, and a mean of 29 words for the review text. The reviews of each wine consist of the name, id, variant or category, year of the wine. Also, each review provided information on reviewer's user id, user name, time of the review, text about the wine, and points which is the rating of the wine. The points is out of 100. We wanted to predict the points of each review based on the other features of the review.

2. Exploratory Analysis

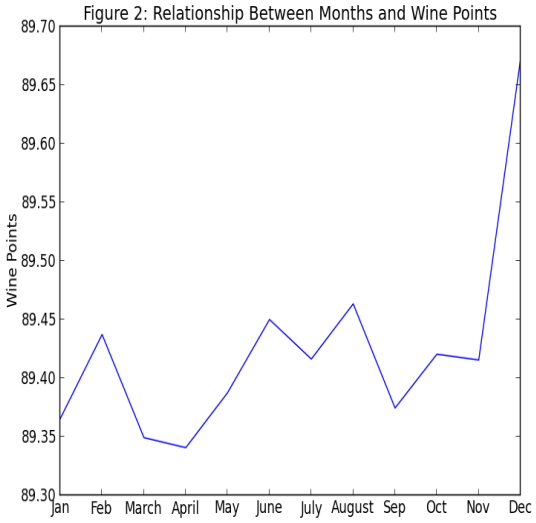
we decided to plot our data points to see the correlation between certain features and points. We decided to explore 500,000 out of the 2,025,995 reviews to view the relationship between the features and points.

2.a Wine Data Set

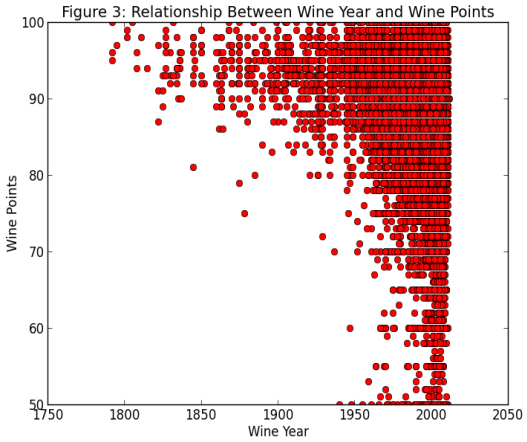
We began by plotting the ratings and number of reviews in a certain range of the ratings. We wanted to see the general points that reviewers gave for wines. We discovered that majority of the reviews points were within the range of 85 and 95 which seemed really high as shown in Figure 1.



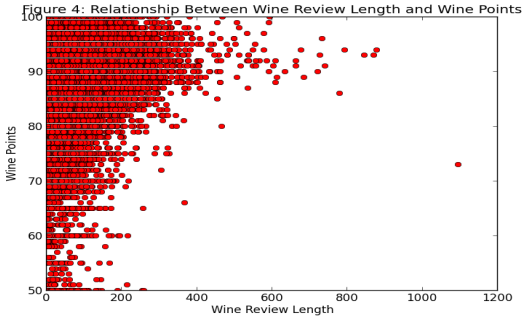
Next, we wanted to see the average points for a given month. The average points for each month are consistent and seem to only waver by 0.05 points as you can see in Figure 2. We decided that the month the review was given did not affect the point value given.



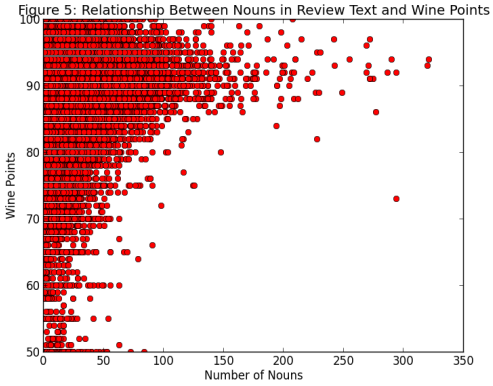
For our first feature, we wanted to explore the relationship between the year of the wine and the review points. We believed that the older wines had better quality and would receive higher points than new wines. This hypothesis seem to be true for older wines because in figure 3 most of their reviews are high, but there are a lot less reviews for older wines compared to the newer wines. For the newer wines, the points seem a lot more sporadic, so the year feature should not be used for newer wines.

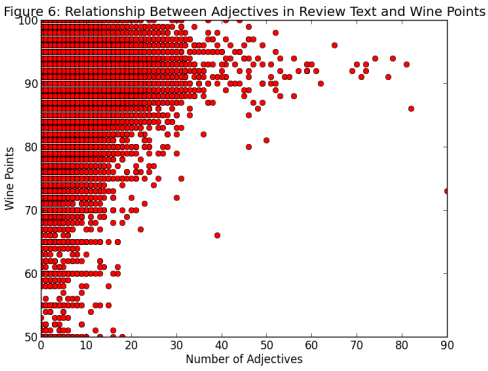


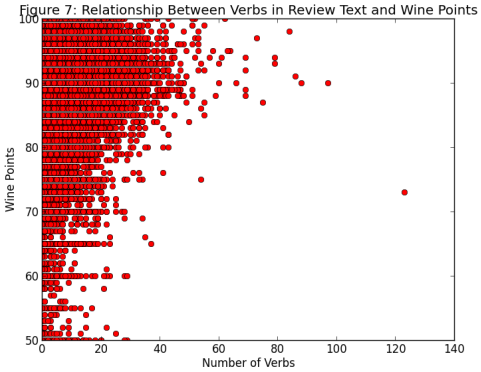
Our next feature, we wanted to see the relationship between the length of the review text and rating. We predicted that people who usually wrote more in their review gave higher points. Figure 4 shows that people who give higher points wrote longer review text, but the relationship between this feature and points seem to be sporadic, so review text length is not a good feature to be used to predict the review point value.



We then decided that we wanted to explore the specific word within a text. With an average of 29 words per review text, we believe that the review text feature would be a great way to predict the points of each review. We tried to find a correlation between the number nouns, adjectives, and verbs in a review text and the point value. The more use of these types of words correspond to a better rating, but it seems to have the same correlation as the review length. The longer the review length, the more nouns, adjectives, and verbs will be used.







3. Predictive Task

We predicted the rating, which is in points from 1 to 100, using features we grabbed from year and the review text. we can discard all the reviews without data on either rating, year, or review text. Then we can construct feature vectors and run different models on it. This dataset is suitable of using both linear regression and classification. We feel like linear regression would be a better model because as above in section 1, we noticed several slight linear relationships with the rating. However, classification would also work due to our huge feature vector. Though, the number of classes (theoretically 100 classes for a score of from 1 to 100) may prove to be difficult for any classifier to accurately predict the rating. For our baseline, we will be use linear regression.

Though we believe that linear regression should produce better results, we are not convinced until we run classification on it because according to Braun and Timpe [2], naive bayes with bag of words model gave excellent results. Therefore besides the linear models that we will use, we will also be using Gaussian Naive Bayes, Logistic Regression, and if possible, Regression Tree, K Nearest Neighbor, and Support Vector Machines.

We will be using the features year, length of review text, and the bag of words model for the first 1000 most popular words. Though, in Assignment 1, text mining proved to be extremely time intensive, so we might have to shorten that feature vector if it proves to take too much time. For data, we just take the first 50,000 reviews with points, year, and review text, and for our training set, and the second 50,000 reviews with points, year, and review text for our testing set. We will explain the difficulties that arises with the data extraction in later sections.

Lastly, running the models should be a relatively easy task, as most of them are present in the python library sklearn. We will be accessing the validity of our models using the Mean Squares Error(MSE) on both our training set and testing set, which each has 50,000 data.

4. Related Works

We are using the dataset from snap.standard.edu which is parse by professor Julian McAuley. In essence, we are trying to predict the rating of an item based on wine features, but mainly using the review text given by the reviewer. Many literatures have been written on the best methods in predicting the rating based, and most of them believe that using the review text is the best way to predict the rating of an item.

In McAuley, Leskovec, and Jurafsky's paper [3], they discussed their Pale Lager model which they used to predict the rating of beers from BeerAdvocate based on the review's text. They categorized certain words to sensory groups(feel, look, smell, taste, and overall) and particular ratings. With this model, they can accurately figure the negative and positive words that describe a beer and correlate a rating to it. We believe that our model does match Pale Lager model because we also found that certain words in a review can be used to predict the rating of an item.

In Que, Ifrim, and Weikum's paper [4] , they discuss that unigram and n-gram are not the best methods in predicting rating from review text because unigram may miss important expressions and n-gram usually aren't trained enough on these phrases. They argue that bag-of-opinions and use the data from Amazon book reviews. They use MPQA lexicon to find opinion roots, modifier, and negation word which are used to predict the rating.

In McAuley and Leskovec's paper [5] , they discuss how the experience of user can affect how they rate an item. They performed their models on reviews for beers, foods, movies, and wines They used a latent factor model with Community evolution at uniform intervals, Individual user evolution at uniform intervals, Community evolution at learned intervals, and Individual user evolution at learned intervals features. They concluded that user's tastes and preferences over time as they try more items. At the same time, there seems to be more consistent review rate from experts compared to newcomers.

5. Models and results

dataset: Our dataset is splitted into two portions; first half as training set to train our models and second half as testing set to test our models accuracy.

baseline model - sentiment analysis and linear regression

To produce better result, we have decided to use text mining as predictor. Our tools included N-gram, bag-of-words, and stemming. We spended a lot of time trying to convert string to unicode such that the stopwords library from nltk can take out unnecessary stopwords. Due to the unicode decoding, so of the string are corrupted as ascii encoding method was not able to decode non utf-8 type. The procedures are simple but nevertheless take a while to process the data. First, we take out all the stop words and punctuations because these words will not contribute much to our model. If we were to use all the unique words we filtered before, the feature will have many dimensions, which is not what we desired. So we limited the feature to the first 1000 most popular words. Using Ridge linear regression with lambda of 1.0. We obtain 10.250212 mse on our training set and 20.431772 on our testing set. After further tuning, we arrived at a reduced MSE and is now served as our baseline model as linear regression is believe to be the most naive approach.

We then ran every model we had on 50000 training data and 50000 testing data and made them into a table below. We immediately notice a few models, namely Decision Tree, Naive Bayes, and K Nearest Neighbors that we can throw out due to poor performance. Then we are left with linear regression and logistic regression, and upon compare we found that linear models work the better than logistic. Earlier we thought that classification would work reasonably well with the number of feature for each vector we have (100+), but apparently it isn't.

Model	training data MSE(50000)	testing data MSE(50000)
Baseline (linear regression)	10.250212	20.431772
Vanilla linear regression	9.731	13.6744
Logistic Regression	10	15
Lasso Regression(alpha=1)	12.5937	14.9801
Ridge Regression(alpha=100)	9.732	13.655
Decision Tree	0	27
SVM	beyond our capabilities	beyond our capabilities
Naive Bayes	246	284
K Nearest Neighbor	23	43
latent factor	15.24586	14.58906

Bag of words sample :

[unbeatable, 92, 90+, 100, perfection, unbelievable, awesome, sublime, magnificent…...]

Comparing to the results obtained by Rossi [1], 16.2886 , our best model performs 20 % better. We expected Naive Bayes to perform well but it turned out to be the worst model. The results obtained by Braun and Timpe [2] indicated that Naive Bayes with bag-of-words as feature performed the best. Nevertheless, Our ridge linear regression yields the best result. We suspect that since our dataset size are different from each other, the ridge regression was able to find a linear relationship between rating and text. Moreover, the processing might lead to lost of some valuable data as we were unable to solve the unicode problem. Our model might have a higher chance of overfitting than the Braun and Timpe's model. Consider the limited processing power that our laptop provided, 50,000 is the best we can do.

On the other hand, our feature has over a hundred of dimensions and is not suitable for advanced classifier models such as Decision Tree and SVM as it take a long period of time to finish processing. We have attempted to use PCA to reduce the dimensionality of our feature vector, but there was no obvious improvement; it means that many of the words that we chose contribute uniformly to the predictions.

Finally, we expected latent factor to be easily defeated by naive bayes with bag-of-words, the result is a surprise to us. Generally speaking, latent factor would not outperform Naive bayes because it does not use bag-of-words as feature. We concluded that user's ratings are highly dependent on his/her history of rating items.

6. Conclusion

In conclusion, our tests prove that this problem is really not a classification problem. With the number of classes it has (50+), it would be really hard to classify each combination of features into a specific class accurately. Though our best classifier - logistic regression - did predict reasonably well on the 50,000 testing datas with a MSE of 15, all our regressors predicted better MSEs; our ridge regressor predicted a MSE of 13.66. Also, the longer the length of the feature vectors in the bag of words model doesn't necessarily yield better results. As we can see in our baseline model, we ran a regression on 1000 most popular words, which produced a MSE of 20.4. However, we then limited our regression to 100 most popular words, and the linear regression produced a significantly better MSE of 13.7.

Although our models yield promising results, there are limitations to these models. For example, our dataset is limited to 50000, which is relatively small compare to other research that we referenced. The overfitting problem might be presented in our models. We tuned our models multiple times to compensate for lack of generalization, such as using different lambda for our linear regression models, and used PCA to reduce dimensionality of our feature vector.

When approaching similar problem in the future, we will expand our repertoire to more models, such as decision tree and TF-IDF models. These models require intense computation since it needs to iterate every item's text and its own libraries multiple time, making the algorithm extremely inefficient. We will put more effort on reducing dimensionality of our feature vector in hope of better results. The field of studying rating prediction has been evolving quickly in

the recent decades, the economic values of these highly accurate models are unfathomable. We hope that our future works will contribute to the ecommerce-ecosystem.

7. **reference**

[1] Rossi,Dominic, '*Predicting wine ratings using linear models',* http://cseweb.ucsd.edu/~jmcauley/cse255/projects/Dominic_Rossi.pdf

[2] Braun, Benjamin & Timpe, Robert, *Text based rating predictions from beer and wine reviews* http://cseweb.ucsd.edu/~jmcauley/cse255/projects/Benjamin_Braun_Robert%20Timpe.pdf

[3] Julian McAuley, Jure Leskovec, Dan Jurafsky, *Learning Attitudes and Attributes from Multi-Aspect Reviews,* 2012 IEEE 12th International Conference on Data Mining http://cs.stanford.edu/people/jure/pubs/beerrec-www13.pdf

[4] Lizhen Qu , Georgiana Ifrim , Gerhard Weikum, *The Bag-of-Opinions Method for Review Rating Prediction from Sparse Text Patterns* https://people.mpi-inf.mpg.de/~lqu/bagOfopinionColing10_camera-ready_final.pdf

[5] Julian McAuley, Jure Leskovec, *From Amateurs to Connoisseurs: Modeling the Evolution of User Expertise through Online Reviews.* http://cs.stanford.edu/people/jure/pubs/beerrec-www13.pdf