Predicting wine quality from reviews

Luis A. Gonzalez-Arraga, Pavla Fuxova

1 Introduction

Our goal is to create an algorithm to predict the quality of a wine based on text reviews written by sommerliers. A example of a sommelier's description of a high-quality wine is:

”A knockout effort, this features perfectly ripened fruit flavors of berries, plums and cherries adorned with baking spices. There's a suggestion of candied citrus rind as well, but it's the superlush fruit that rocks on through a lengthy finish.” On the other hand, an example of a negative review is:

”Smells oaky and spicy, but it's almost shockingly lean in fruit. Tastes like oak-flavored alcohol, acids and tannins.” Our approach will be twofold: on the one hand we will try a series of classification models to assign wines to quality categories (high-low) and on the other hand we will create regression models to treat the quality as a continuous variable. Classification systems based on text often use as predictors the tf-idf of a list of frequent words in the vocabulary (bag-of-words model). We will start with a standard bag-of-words model and will later try to improve the performance of our classification by different models.

2 Description of the data

We will work using the dataset ”Wine Reviews” which can be downloaded from kaggle https://www.kaggle.com/zynicide/wine-reviews/data. The dataset contains ten columns, namely: country,description, designation, points, price, province, region1, region2, variety and winery . The columns ”points” (wine rating from 80 to 100 points) and ”price” (in US dollars) contain numerical data, and all others contain string data. The column ”description” contains a sommeliers review of a particular wine product. The dataset contains originally 150930 rows of data. After removal of duplicate rows the dataset is reduced to 97581 rows.

3 Methodology

We begin with a brief exploratory analysis of our dataset wich can be found in the notebook ranking varieties.ipynb. In the top panel of Fig.(1), we show a scatterplot of the countries in the dataset in a plane consisting of the average price of wines produced in that country vs the average rating score for the wines produced in it. England appears as an outlier, due to the high average quality and price of its wines, but this is easily explained when we see that only 9 wine reviews come from that country. In the bottom panel of Fig.(1) we show the number of reviews for the 15 most frequent wine-varieties, we see that only a handful of varieties show up more than 4000 times in the dataset (Pinot-Noir, Chardonnay,Cabernet Sauvignon and Red Blend). More than 40,000 of the reviews correspond to wines produced in the United States, more than the second and third countries combined (Italy: 14435 and France: 14235 wines respectively)

The type of vocabulary that is used to describe the flavors in wines is heavily dependent on the grape's color, see Fig.(2). Sommeliers often report tastes of dark and red fruits (such as cherries, raspberries, blueberries, etc) when describing red wines and whereas white wines contain flavors such as lime,lemon, orange, apricot and peaches. Since our main goal is to build an algorithm to predict a wines quality based on the text of the description, it is highly convenient to split the dataset into white and red varieties of wine prior to applying a predictive algorithm, in order to optimize the performance of our model. We begin by filtering our dataset so that it only contains wine varieties for which there are at least 10 rows of data, and removing punctuation symbols and transforming all words in the description column to lowercase letters (so that TextBlob doesn't interpret "Berry" and "berry." as different words, for example). Then for each wine variety, we count the frequency of words associated with red and white wines. The red wine related words we will consider are : cherry,berry, raspberry, blueberry,blackberry and red. The white wine related terms we will use are: lemon, lime, peach, apricot, pear, apple, nectarine, orange, pineapple and wine. Based on the frequency of these words, we will assign to each variety a redness and whiteness score (included as new columns in the dataframe), finally we will determine the color of the grape based on which of these scores is greater. As a side note, we use the same technique to rank wine

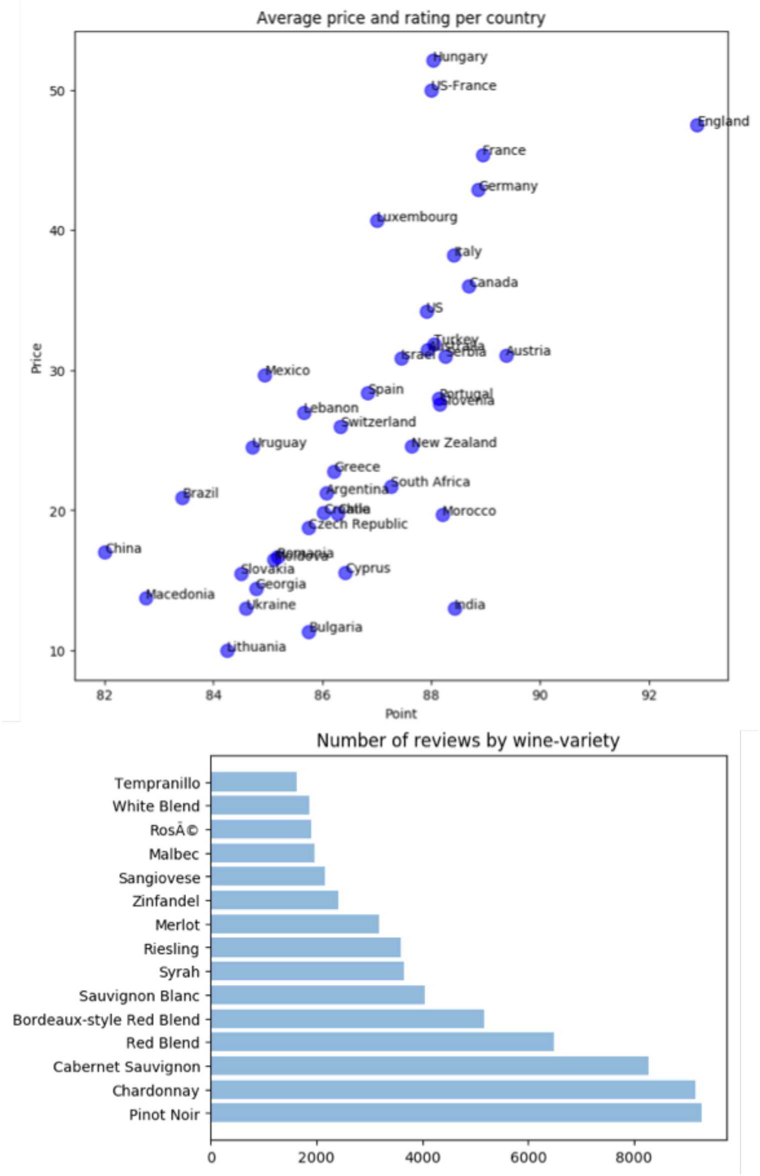

Figure 1: The top panel shows a scatter-plot of the average price and average rating of each wine variety. The bottom panel shows the number of reviews for the 15 most common wine varieties

varieties by important quality characteristics such as sweetness, tannin and acidity in the notebook ”ranking varieties”.

Once we have identified the grape color for each of the items in the dataset we begin to work on two models for classifier systems to identify

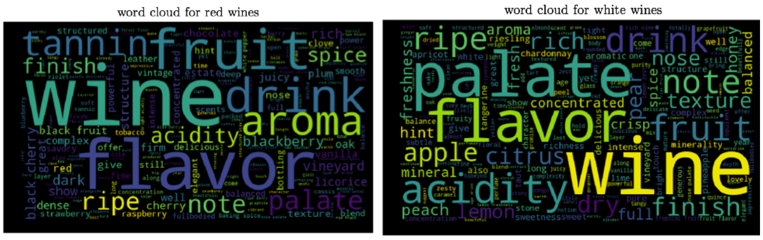

Figure 2: Wordclouds for red and white wine reviews

wines of poor (between 80 and 84 points) and high quality (93-100 points). We will try three different classification models, firstly a **bag-of-words** classifier: within this model we first define a vocabulary of words from the the description with medium frequency, secondly each wine description is transformed into an array in which each component is the tf-idf (term frequency-inverse document frequency) of a specific word of the vocabulary, and finally, we apply classification algorithms (Naive Bayes, and Logistic Regression) using the tf-idf arrays for each description. This, we develop explicitly in the notebook ”bag of words classifier.ipynb”. We used 50% of the data as a training dataset (in all further models we will use the same partition for train-test sets). The performance of this algorithm is greatly increased when the bag-of-words and the classifier is performed only on one type of wine-color, the area under the ROC curve for the logistic regression classifier goes from 0.878 (color-neutral bag of words) to 0.90 (if the classification task is carried only on red wines).The multivariate logistic classifier gives the best performance in this problem, it is based on the logistic equation:

$$P(x_1, x_2, ...x_n) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + ...\beta_n x_n)}} \tag{1}$$

where P is the probability of a wine being high quality and $x_1, x_2, ...x_n$ are the predictors (in this model the predictors are the tf-idf of each word in the vocabulary for example). The classifier first assigns a probability P for a particular wine description being high quality, if $P > 0.5$ the algorithm classifies the wine as high quality, else it classifies it as low-quality.

We find that an important weakness in the bag-of-words model, is that the vocabulary terms used for classifying purposes are simply chosen because of their frequency in the documents, but may very well not be correlated with wine-quality at all. Our second classifier, which we name **word-count of informative words** model attempts to overcome this difficulty

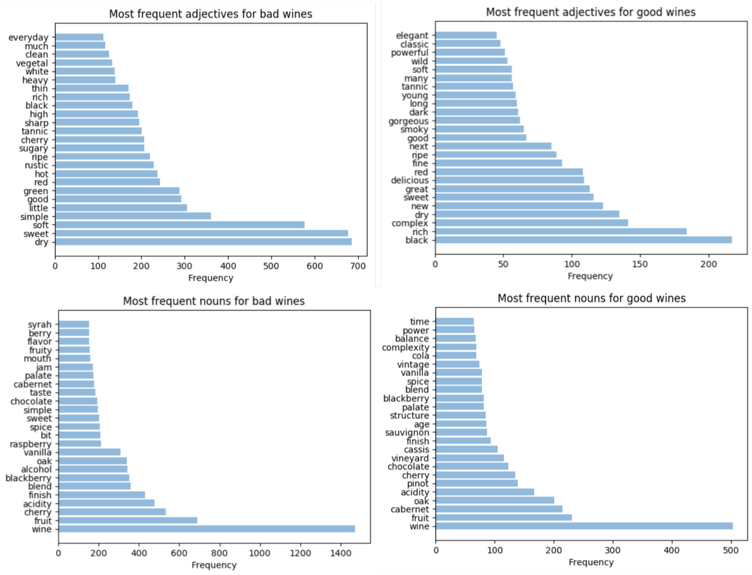

Figure 3: Most frequent adjectives and nouns for each quality category

by using only vocabulary words that are highly correlated with wine quality. We begin by using the part-of-speech function (pos-tagger) from the natural language tool-kit library from python (nltk) in order to obtain the most common adjectives and nouns for the two categories of wine, see Fig.(3). This allows us to get a sense of which frequent terms may be highly informative when assessing a wine's quality. We identify the highly informative words by plotting their frequency vs the wine rating and we select the ones that we find to be most heavily skewed towards one quality type, some examples are shown in Fig.(4). The code to select the highly informative words is shown in the notebook ”pos tagger informative words.ipynb”, whereas the the classification based in the word counts is in the notebook ”word count classifier.ipynb ”.We try three different classifying algorithms : a multivariate logistic regression classifier, K-nearest-neighbor and a Random Forest Classifier. Within this model based in the frequencies of informative words, the k-nearest neighbor algorithm poses a difficulty, given that this algorithm is based in the calculation of Euclidean distances that are unhelpful for highly dimensional spaces (link to wikipedia). Therefore, before applying the k-NN algorithm, we performed a Principal Component Analysis, thereby reducing 26 dimensions to just 10 principal components that together explain 80% of the variance.

The bag-of-words and the word-count classifiers are somewhat difficult

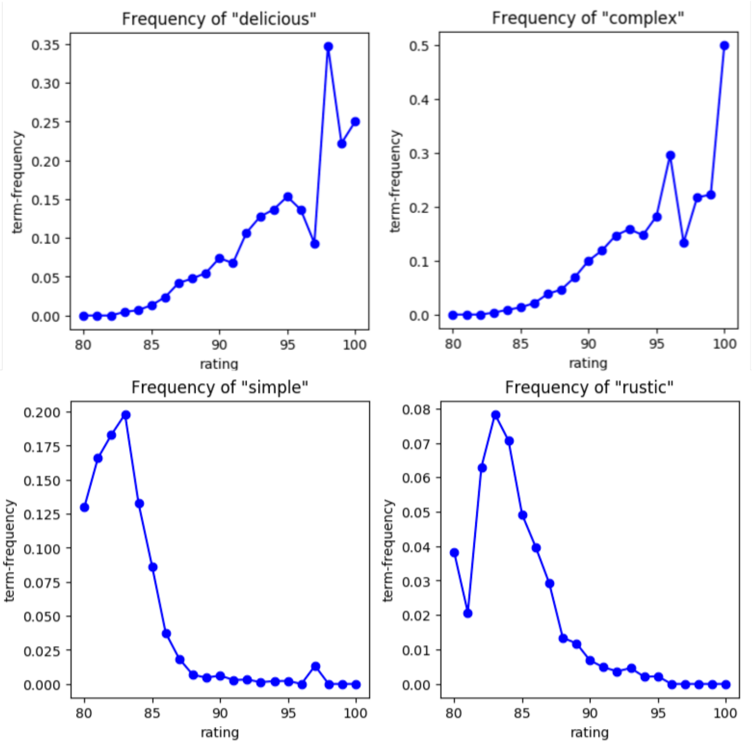

Figure 4: Frequency vs rating for a few of highly informative words. These are examples of words we will use for our word-count of informative words model

to apply on large datasets, given that both use a large number of predictors. It is therefore highly convenient to find a simpler method of classification, hence we try a**review length-polarity model** based on the length of the review (measured as the total number of words in the review) and the polarity of the description as extracted from the sentiment analysis function from the TextBlob library. This is shown in notebook ”length polarity classifier.ipynb”.Both predictors show a moderate positive correlation with the wine’s rating. For the sake of comparison, we will also perform an unsupervised K-means clustering based on the same variables.

We also develop an algorithm to predict the specific rating score of each wine description via a linear regression using as predictors,the review length and review polarity as well as the frequencies of the informative words. We use the root mean square error and the variance score as metrics to estimate accuracy of our model, this can be found at the end of the notebook ”word count classifier.ipynb”.

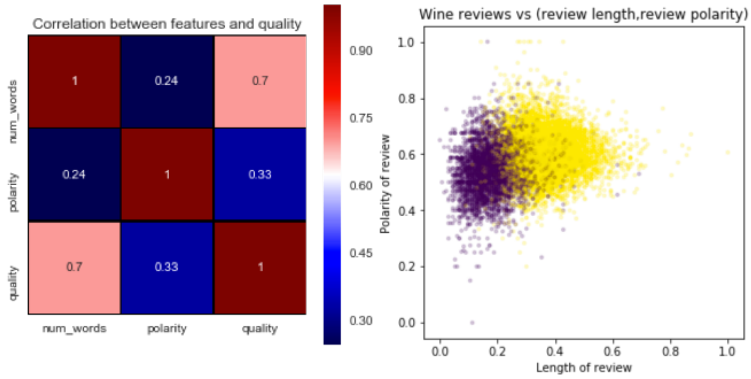

Figure 5: Left panel: a matrix showing the correlations between wine quality and review length and polarity. Right panel: a scatterplot of data points in a review-polarity vs review-length plane. Blue (yellow) data points belong to the low-quality (high-quality) category.

4 Summary of results

We found the word-count model based on highly informative words (area under ROC = 0.926) performs better than the bag-of-words model (area under ROC curve=0.90). We have also found that in both cases its best to perform the classifying task using a logistic regression classifier.

For the review length-polarity classifying model, the Logistic Regression classifier gave the best performance (measured as the highest area below the ROC curve) of the three algorithms we used, therefore here we only discuss the results of this algorithm. The top panel in Fig. (5) shows the confusion matrix. In order to compare the results of our classifier with those of unsupervised learning, we performed K-means clustering on the same datapoints; we use the elbow method and the silhouette score and observe that the data can be best partitioned in two clusters, which we found to mostly coincide with the low-quality and high-quality wines see Fig. (6). We have found the review length-polarity model to be most appropriate method of classification for this dataset based on its low-dimensionality, its independence on the grape color, and its high efficiency (best area under ROC curve for logistic regression: 0.973) which outperforms both the bag-of-words and word-count models.

Our linear regression to predict wine quality as a continuous variable uses as predictors the review length and polarity as well as the word-counts of all the highly informative words. The average wine-score is 88.13 and

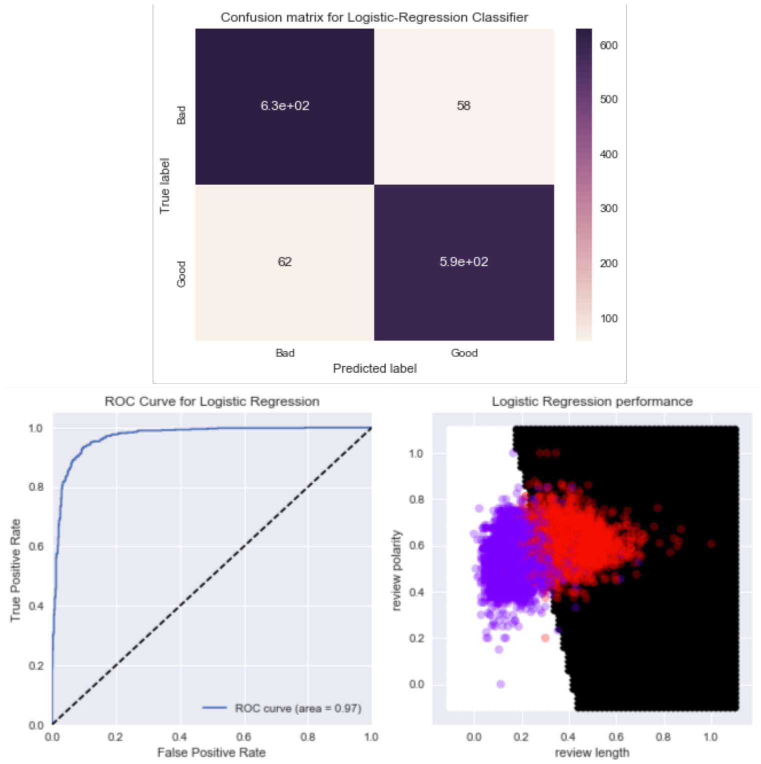

Figure 6: Performance parameters for logistic regression classifier based on review length and review polarity. The bottom right panel compares the actual datapoints (purple points are low-quality wines, red points are high-quality wines) with the predicted label, given by the background color (white for low-quality, black for high-quality)

standard deviation of the distribution of wine-scores is 3.294. The root-mean-square-error of the linear regression is 2.51, and the variance is $R^2 = 0.41$

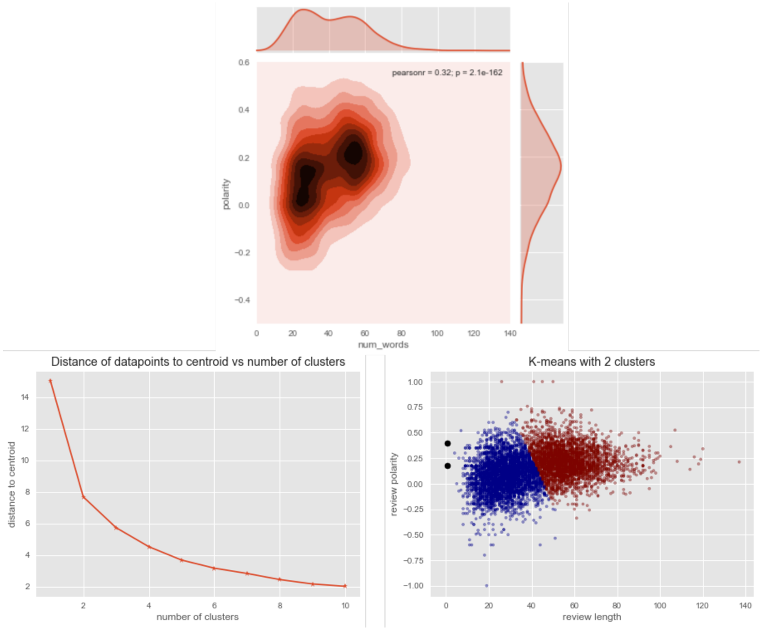

Figure 7: The top panel shows the distribution of datapoints in the plane of number of words and sentiment polarity. The bottom left panel shows the average distance of datapoints to cluster centroid as a function of the number of partitions, an ”elbow” point occurs for K=2. The bottom right panel shows the datapoints colored according to their cluster label, we can see that the unsupervised learning