Finding What Makes a Perfect Pour

A Dual Approach to Identifying What Makes a High Quality Wine Through Text Analysis & Feature Evaluation

Abstract

Wine, with its rich history and complex nature, that is evidently reflected in its evaluation by esteemed sommeliers[1], serves as a fascinating subject for natural language processing. This project will dive into the descriptions provided by wine experts, seeing if the complex nature of their prose holds useful information on what makes a superb wine. On a corpus of over 130,000 reviews, we'll leverage machine learning to tackle two key tasks–one focused on using review text for sentiment analysis, and another to classify wine on the same labels used in task one, but with other features aside from review text. By using both text and non-text based features, we'll evaluate whether there are latent factors hidden within the taste, odor, and essence of wine that concrete features like variety and price are unable to capture.

1. About the Dataset

The data we'll be working with for this project was scraped from *WineEnthusiast* during the week of November 22nd, 2017 and can be found on Kaggle here. With roughly 130,000 entries and 14 columns that contain the following features:

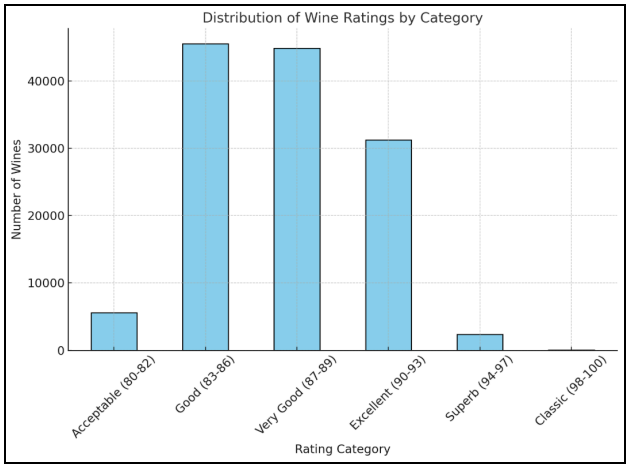
id	ID entry for each wine
country	Country of origin for each wine
description	Review description from taster
designation	Origin of grape - Which vineyard of winery
points	# of points the WineEnthusiast taster rated the wine
price	Cost for a bottle of that wine
province	Province or State of Origin
region_1	Winegrowing area
region_2	More specific region within winegrowing area
taster_name	Name of Reviewer
taster_twitter	Twitter handle of Reviewer
title	Title of the review, name of the wine
variety	Type of grapes used to make the wine
winery	Winery that made the wine

[1] A sommelier, or wine steward, is a trained and knowledgeable wine professional, normally working in fine restaurants, who specializes in all aspects of wine service as well as wine and food pairing.

Some important context in regards to the feature which will serve as our main target variable that we seek to classify (**points**) is that it has been assessed on the following scale according to *WineEnthusiant's* own website. With ratings below 80 deemed unacceptable and will thus not show up in the dataset, the remaining scale is shown as:

98-100	**Classic** The Pinnacle of quality
94-97	**Superb** A great achievement
90-93	**Excellent** Highly recommended
87-89	**Very Good** Often good value; well recommended
83-86	**Good** Suitable for everyday consumption, often good value
80-82	**Acceptable** Can be employed

However, we will be adjusting the scale for our specific project as we will be performing sentiment analysis–binning scores of (90-100) and (80-89) as 'High Quality' and 'Standard' respectively–to more broadly assess a quality wine. Doing such will also limit imbalanced data issues and will better allow us to more holistically evaluate wine that excels in its field. In terms of the distribution of the dataset, we see the top 5 most common countries of origin being the US, France, Italy, Spain, and Portugal respectively, with the US alone making up roughly 54,000 of the wines reviewed. With the US being the predominant country of origin, it is no surprise that the most common province within the dataset is thus none other than California (with Washington in second). Other notable facts include the most common wine varieties being Pinot Noir and Chardonnay, each making up around 10% of the dataset respectively. Depicted below is the distribution of ratings, binned on the categories we've mentioned above (before our adjustments):



As we can see, without our adjustments - classification models may achieve a high overall *accuracy*, yet may lack precision in flagging '**Classic**' or '**Superb**' wines due to their lack of availability in the dataset when compared to other rating categories. Diving a little deeper into other notable statistics about our data, more specifically average rating scores by designation, variety, winery, and region_1. After grouping the '**points**' by country and finding their average value, we see the top 3 countries with the highest average are England, India, and Austria (being the only countries to have an average point value of over 90) – while the bottom 3 countries were Ukraine, Egypt, and Peru (all hovering around the 83-84 mark). However, it is important to note that these statistics aren't fully representative of what countries have the best wine, as averages may be brought down or skewed by the # of entries and outliers (less reputable wineries). Additionally, the prestige of a particular wine's origin is typically evaluated based on region (which we will see later on). In consideration of the aforementioned details, when grouping by variety, winery, and finally region, we'll impose a 5 entry minimum to the following evaluations. First, we'll look at varieties:

```
Top Wineries by Average Points:
 winery
Salon                    96.800000
Tenuta dell'Ornellaia    96.700000
Cardinale                96.000000
```

Before moving onto winery averages:

```
Top Varieties by Average Points:
 variety
Bual                94.142857
Muscadelle          92.500000
Loin de l'Oeil      91.333333
```

And finally, regional differences:

```
Top Regions by Average Points:
 region_1
Bolgheri Sassicaia       96.625000
Montrachet               96.000000
Chevalier-Montrachet     95.428571
```

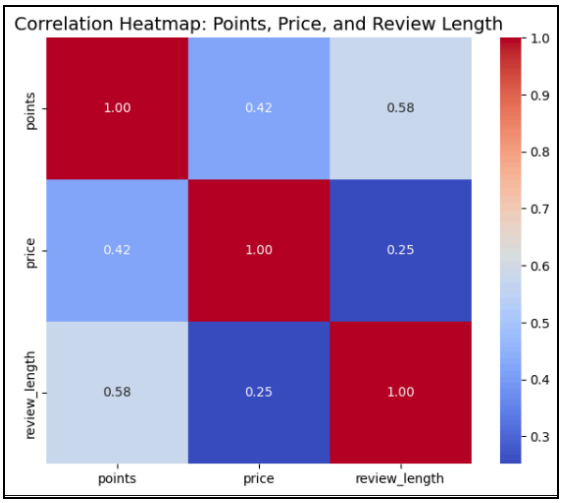
1.1 Data Quality

Meanwhile, when assessing data quality, some features contain many missing/NaN[2] values such as **designation** (29% missing), **price** (7%), **region_1** (16%), **region_2** (61%), **taster_name** (20%), and **taster_twitter** (24%). Due to the sheer volume of missing values in **designation**, and **regions**, they will be excluded from our non-text based model. Meanwhile, we will impute median values for **price**, which will be used as a feature in our non-text classification model. We will dive deeper into the exact feature details down below. Moving back to data quality, no duplicate entries were found within the dataset.

1.2 Exploring Possible Features For Non-Text-Based Classification

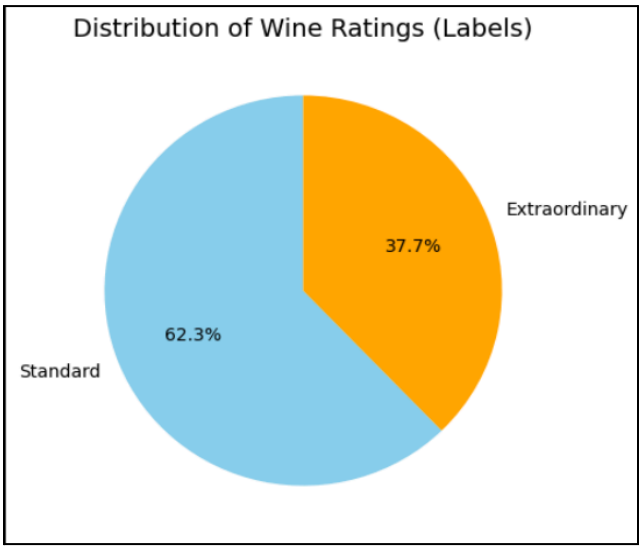
For now, we'll seek to identify any possible predictors of rating that we'll evaluate later on by creating a correlation heatmap. The table below will include point, price, and a newly engineered feature **review_length**, and their respective correlations. The following graph reveals a mild correlation between points given for a review, and the corresponding wine's price and the length of its review. This will be taken into consideration when determining features for our model.

[2] NaN, which stands for "Not a Number", is a numeric data type that represents an undefined value or a value that cannot be represented



1.3 Feature Engineering 'Label'

On the other hand, to prepare our data to be classified down the line, we'll create a new column '**label**', which will bin the '**points**' feature as described above (into Extraordinary or Standard). After having done that, we can see the new distribution of the target variable in the pie chart below:



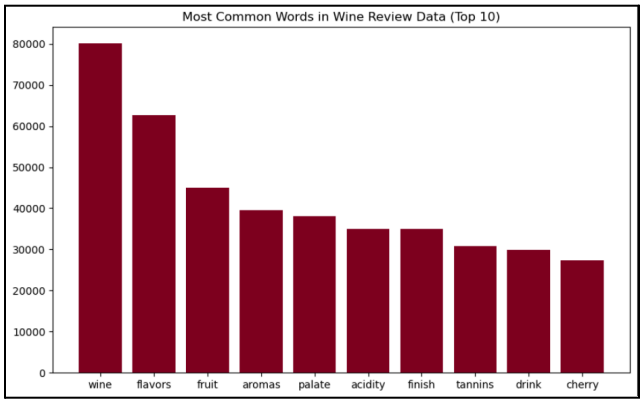
While it isn't split approximately in half, the dataset doesn't face the same imbalance issue as before (with classes like superb making up less than .01% of the dataset), and stratification of the predictor variable when creating train/test splits should provide a representative training sample for our classification models.

2. Data Preparation & Modeling

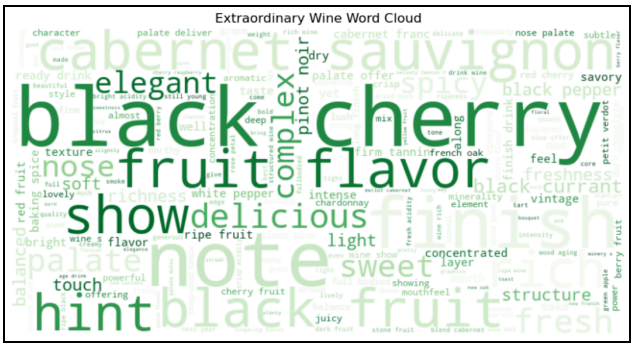
As mentioned before, we seek to accomplish two key tasks in this project–one task being to perform sentiment analysis and train a model to classify a wine rating (as Extraordinary or Standard) using review text data, and another to use non-text features to classify on the same labels. We'll weigh the two tasks (and their respective models), their pros and cons, and evaluate their performance later using the same metrics.

2.1 Sentiment Analysis

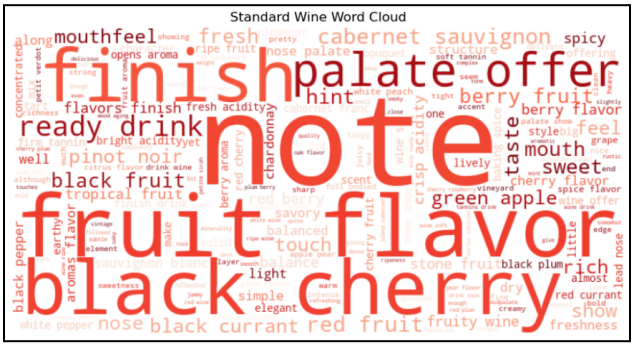
For our first task, we'll need to parse through the review text data, turning them into vectors for which we can use for our classification models. Before delving into building a bag-of-words feature vector and/or using TF-IDF scores, we'll need to tokenize the words in '**description**' and clean up the feature by removing capitalization, punctuation, and stopwords as these may provide limited information for our classifier models. We will be using the Natural Language Toolkit (nltk) library, and their respective **stopwords,** and **word_tokenize** functions to complete the aforementioned tasks. After doing such, we've created a chart of the most commonly used words in the wine review data:



However, it may be far more useful to actually separate these counts by their label before charting. In the format of a word cloud below, which depicts and scales common words within a review, the frequency of words within a wine classified as extraordinary is depicted below:



With a word cloud of wines classified as standard as:



Now, for our first task we'll use a bag-of-words vector of the 10,000 most common words (out of a total of ~43,000 unique words within the corpus) to train a logistic regression model. We'll use a classification report, from the Sklearn library, in order to evaluate test accuracy, precision, recall, and F1-score. This will be our baseline model before we employ a vector of TF-IDF values for the 10,000 most common words and compare their performance on the previously mentioned metrics.

2.2 Using Non-Text Features

For our second predictive task we will be training the same logistic regression model, but instead on the following features: **price**, **review_length**, and a one-hot encoding of the wine **variety**.

3. Model Evaluation

For both of our classification tasks, we did a 70/30 train-test split, stratifying on '**label**' to ensure that a representative sample is taken for training.

3.1 Logistic Regression & B.O.W

For our first task, we fit the B.O.W vector of the 10,000 most common words in the corpus to a logistic regression model with a max_iter set to 1000. The initial run through provided us with the following classification report:

```
Classification Report:
                precision    recall  f1-score

Extraordinary       0.80      0.76      0.78
     Standard       0.86      0.88      0.87

     accuracy                           0.84
```

To tune the model, we used Sklearn's GridSearchCV function to evaluate different C values in our tuning grid. These values included 0.1, 0.2, 0.5, 0.8, 1, 2, and 3–which through 5-fold cross validation on the training set, determined that 0.5 would be the best parameter value for C. However, optimizing the C parameter saw limited improvements in the model with accuracy scores rising less than 1%. Notably, the model did not run into any scalability issues as it was able to train and test on the entire dataset within 5 minutes without the need to be scaled down.

3.2 Support Vector Machine & B.O.W

We also fit a SVM with the same bag-of-words vector, setting the kernel as linear. We also tuned the model the same way as earlier, this time only on two parameters of C: 0.01 and 0.1 (as computational cost was a huge issue). The chosen regularization parameter was 0.1, producing the following classification report.

```
Classification Report for SVM:
                precision    recall  f1-score

Extraordinary       0.80      0.75      0.78
     Standard       0.86      0.89      0.87

     accuracy                           0.84
```

While the performance was nearly identical to that of the logistic regression model, the SVM struggled immensely with the size of the dataset–taking roughly 4 hours to train and test. For that reason, it was in the interest of time that we limited the available tuning parameters. However, the lack in performance difference may be due to the selected kernel–as perhaps using polynomial or radial

functions (rather than linear) may have provided a more notable difference.

3.3 TF-IDF

As the performance of both models were nearly identical, we adjusted the feature that we trained these models on by creating a vector of TF-IDF scores of the 10,000 most common words. We then fit this new vector to a logistic regression model to achieve the following results:

```
Classification Report (TF-IDF Top 10,000 Words):
              precision    recall  f1-score   

Extraordinary       0.79      0.73      0.76
     Standard       0.84      0.88      0.86

     accuracy                           0.83
```

Surprisingly, the model actually performed worse, with a slight 1-2% reduction in every metric. We would evaluate on different numbers of common words as well (i.e. Top 1000), which would all post similar (worse) results.

3.4 Logistic Regression & Non-Text Features

In this next section, we'll evaluate a logistic regression model's classification ability on the three features mentioned in section 2 (**price**, **review_length**, and **variety**). After fitting and testing the model, it was able to achieve the following scores:

```
Classification Report:
              precision    recall  f1-score

Extraordinary       0.76      0.62      0.68
     Standard       0.79      0.88      0.83

     accuracy                           0.78
```

When compared to our text-based classifiers, the model performed notably worse, with drops in precision, recall, and f1-score across both labels. This drop in performance aligns with what we sought to conclude earlier–that there is something intangible about high quality wines that cannot be simply captured through more concrete features.

3.5 Limitations Discussion

However, there are notable limitations within the model features which we have fit. Starting with our text-based classifier, while we explored both bag-of-words and TF-IDF scores, our exploration was limited to unigram feature vectors. With wine descriptions being a host of complex adjectives, perhaps a bigram model may have better captured the intricate nature of wine reviews. As for our non-text based classifier, a common issue was the inability to one-hot encode the sheer number of different designations, wineries, and regions (as well as complete omission of taster). There were data quality issues as well as we did not use the wine's designation as nearly a third of this feature would be missing. Both region_1 and region_2 were omitted features within our model as well due to similar data quality issues, as well as difficulty with one-hot encoding that many feature vectors. Wineries had similar problems as well. This may have definitely hurt the model as there are prestigious regions and wineries that are quite well-known for producing high quality wine. Perhaps a simpler one-hot encoding of whether a wine originated from a top 5 winery, origin, or region may have seen improvements in recall scores for the extraordinary label (which our non-text based model struggled with doing).

3.6 Results Discussion

So what do the results suggest about indicators of wine quality? First, it seems that the experts are able to describe and capture the quality of wine within their descriptions. With our text-based models being better able to find 10% more of the **extraordinary** wine within the dataset, it suggests that predicting on a wine's price and variety is not enough to dictate quality. Rather it is some latent information that, when uncovered and made tangible within a sommelier's description, is able to describe a superb wine. So what is that latent information–by examining the most "impactful" coefficients within our B.O.W logistic regression model we can see key descriptors that set a wine above its peers.

Top Words [Extraordinary]:

	Word	Coefficient
7510	sample	-2.818020
3788	gorgeous	-1.878711
8593	superb	-1.761339
648	beautiful	-1.758704
8876	terrific	-1.742931
3045	exquisite	-1.723068
7190	resist	-1.689537
5018	longterm	-1.684403
2256	dazzling	-1.674711
2521	distinguished	-1.597608

Meanwhile, for the words that were most indicative of a Standard wine, we see words like “simple”, “value priced”, “generic”, and “budget”.

4. Assessing Similar Literature

Similar studies have been done as evidenced by an article called *Using Neural Network Models for Wine Review Classification*, where authors employed the following neural networks: CNN, BiLSTM, and BERT, to classify wines according to different rating classes. Using a large collection of wine reviews from *Wine Spectator*, they found that BERT (a framework developed by Google) proved to perform the best. Their study also did a two-class classification using the same range as within this project, and was able to achieve an accuracy of 89.12%. They did, however, also perform four-class classification where scores hovered around 80% for all three models.

Rating Categorization	*Model*	*Accuracy*	*Precision*
Two-Class	CNN	0.8802	0.8708
	BiLSTM	0.8869	0.8759
	BERT	0.8912	0.8797
Four-Class	CNN	0.7935	0.7767
	BiLSTM	0.8011	0.7563
	BERT	0.8157	0.7801

The study arrived at a similar conclusion that our report seeks to prove, which is that wine reviews and ratings contain latent sensory information about wines that consumers aren’t able to obtain from the objective features measured about them. Another paper published by the Cambridge University Press as well backs this up. In the article *Uncovering the language of Wine Experts*, researchers sought to find out whether or not wine experts were actually able to convey important information in their review and descriptions. They would then build a classifier model (using a SVM) that tried to determine the color, variety, and origin of a given wine based on a wine review. With strong performances as shown:

Task	Number of reviews	F-score	Baseline
Color	68,224	95.4	65.8
Grape variety	48,760	57.1	14.0
Origin	72,925	61.5	56.0

The report was able to conclude that reviewers were able to distinguish between the vast differences in wines effectively and consistently. Another piece of literature worth mentioning as well explores wine quality using features not evaluated in either of the above reports. Features like citric acid, pH, fixed acidity, residual sugar, and alcohol were used as predictors of wine quality. This approach leaned into the chemistry side of wine that reports like ours did not consider. A future discussion that weighs how these selected features align with certain descriptors in review text may be an interesting topic to explore.

4.1 References

1. Katumullage D, Yang C, Barth J, Cao J. Using Neural Network Models for Wine Review Classification. *Journal of Wine Economics*. 2022;17(1):27-41. doi:10.1017/jwe.2022.2
2. Croijmans I, Hendrickx I, Lefever E, Majid A, Van Den Bosch A. Uncovering the language of wine experts. Natural Language Engineering. 2020;26(5):511-530. doi:10.1017/S1351324919000500
3. Yogesh Gupta, Selection of important features and predicting wine quality using machine learning techniques, Procedia Computer Science, 125, 2018, 305-312, https://doi.org/10.1016/j.procs.2017.12.041.