Capstone 2 Final Report

Problem Statement

People often buy the same kind of wine over and over again. If you don't know what a wine's points value means or how origin can affect taste, you may not be as willing to try different kinds of wine. I want to build a recommendation tool that will recommend different wines that are similar to the given wine. The data will come from the following dataset:

https://www.kaggle.com/zynicide/wine-reviews

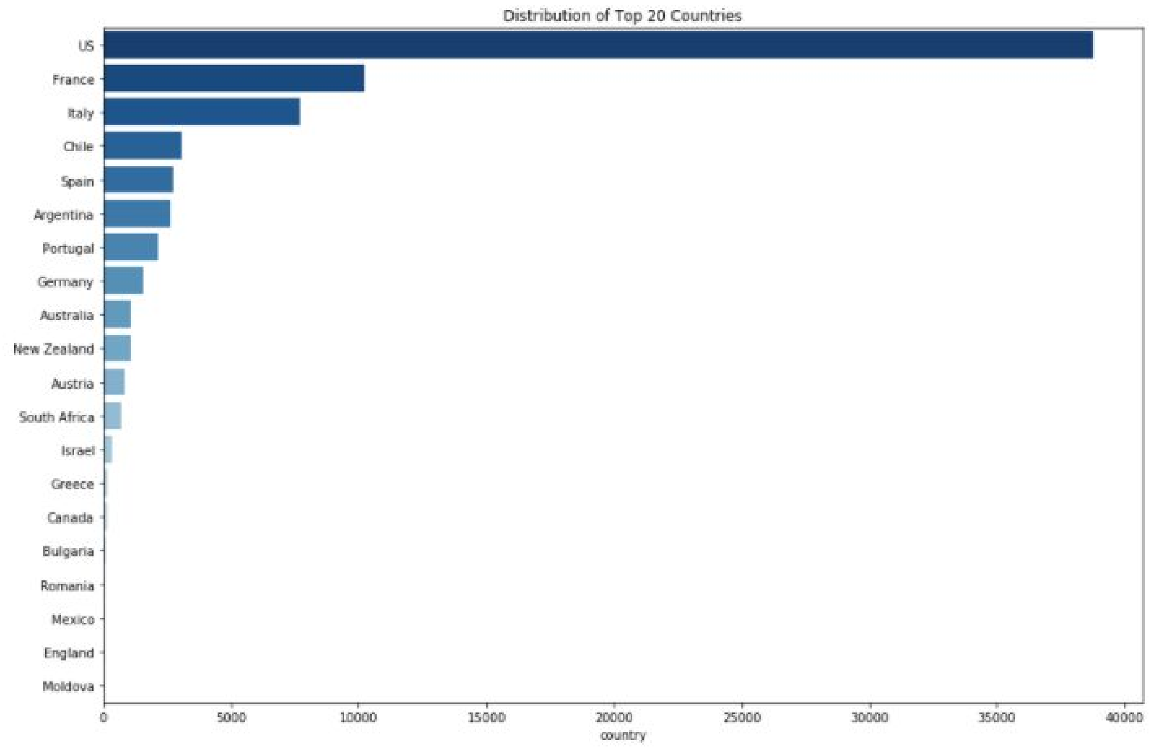
This dataset contains various attributes about a given wine, such as title, price, points, country of origin, the vineyard where the grapes were made, and a brief description of the wine from a sommelier. I will look for trends in the given data in order to provide accurate recommendations for users looking to branch out and try different types of wine.

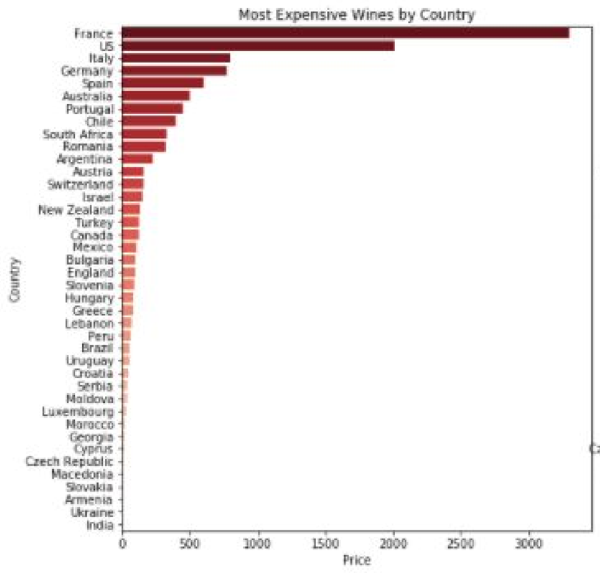
Data Wrangling

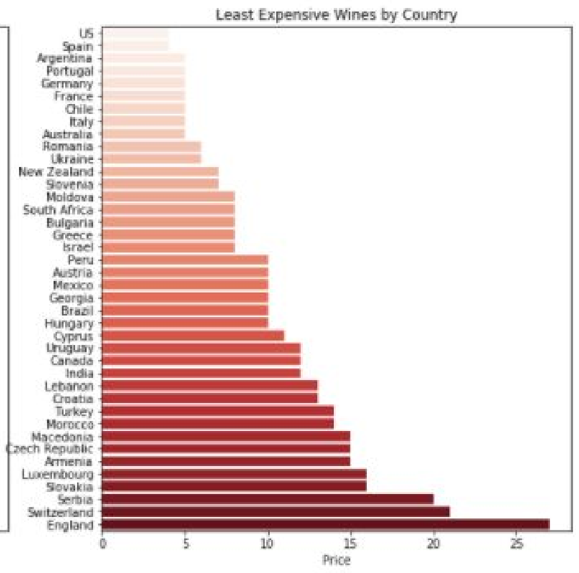
The wine data for this project is contained in two csv files, the first containing 150,000 rows and the second containing 130,000 rows. I first read in each csv file as a pandas dataframe and concatenated the two dataframes to make one large dataframe containing all of the data, which will make the analysis easier. Looking at the number of null values in each column, I removed the designation, region 1 and 2, taster name, and taster's twitter handle columns. Then, because of the duplicates as well as the null values in the title column, I decided to drop the duplicates in this column because I needed it to build the recommendation system. Other columns, such as country, province, and price, had a relatively small number of null values and contained information I was planning on using for my analysis, so I just dropped the rows containing null values in these columns rather than dropping the columns altogether. There were a large number of varieties in the dataset as well, so I decided to just focus on the top 20 for my analysis.

Exploratory Analysis

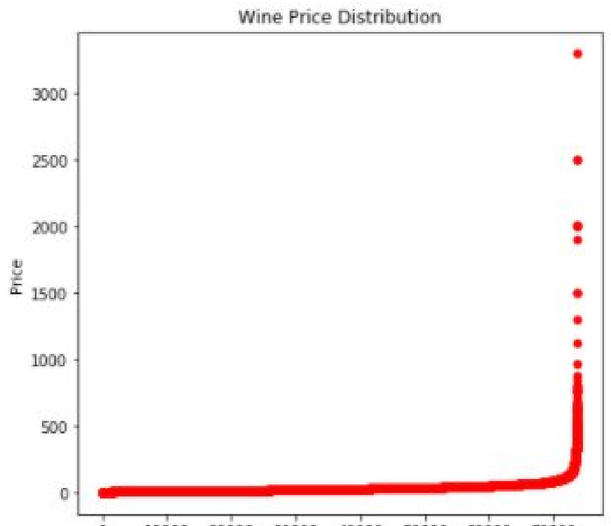
Out of the 50 countries represented in the dataset, the majority of the wine is produced by the US, followed by France and Italy. There is also a similar group of countries atop the list of most expensive wine sold and the least expensive wine sold, leading me to believe there is a great range of wines sold in countries like the US, France, Portugal, Argentina, and Italy. I noticed there was one country named “US-France”, which is likely a wine that is produced in the US and sold in France or vice-versa.

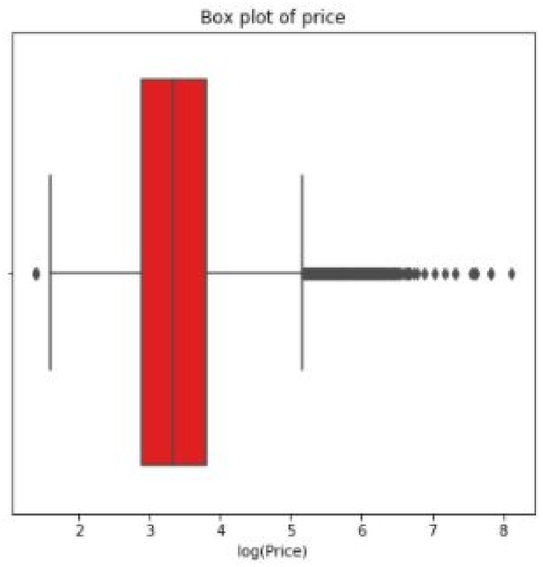




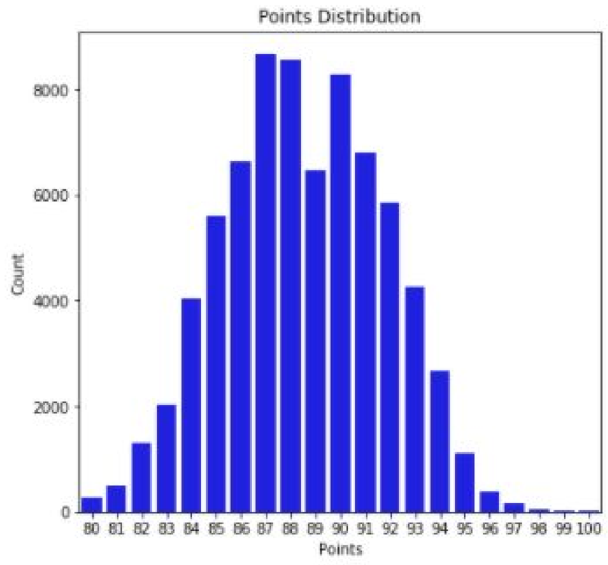


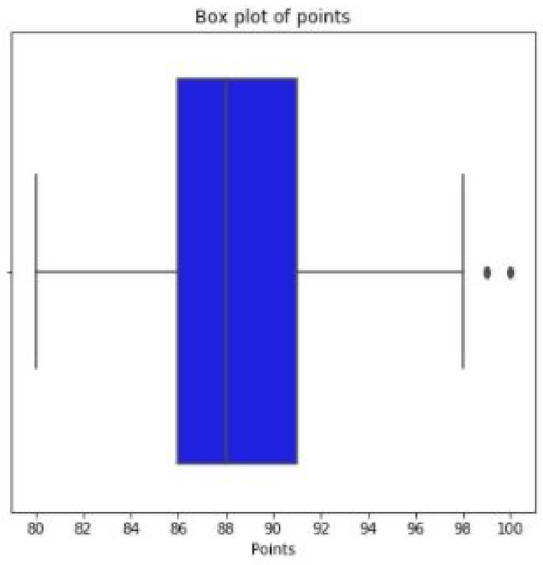
Looking at the distribution of wine prices, there appears to be some upper outliers, so I wrote a simple function to output the number of outliers in a given column. In the price column, there are no lower outliers and 751 higher outliers out of 73,691 total observations. This only represents 1.0296% of the data. Also, none of these outliers appear to be from erroneous data, so I decided to leave them in. I also decided to use a logarithmic scale because the range in prices was high.



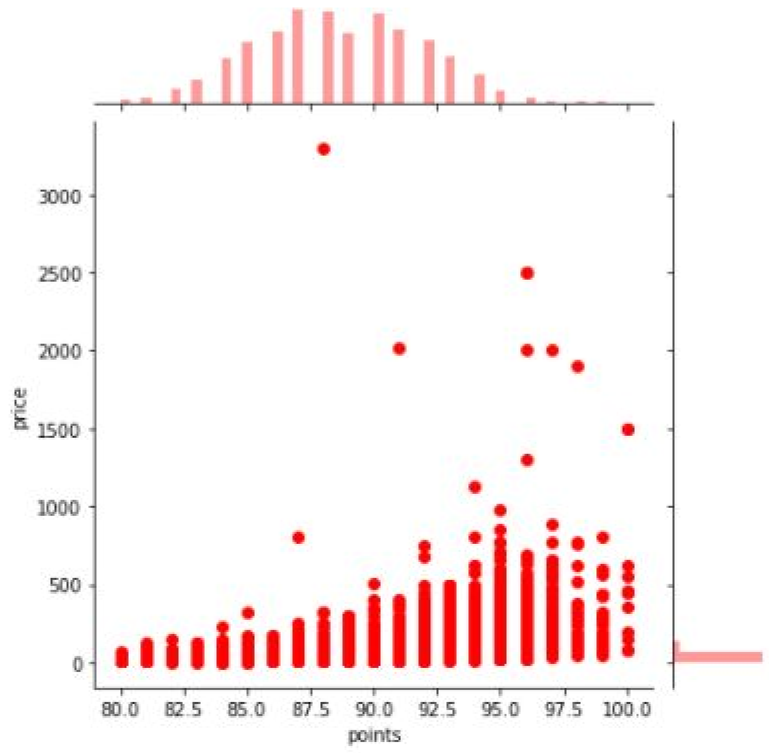


The points distribution appears to be normally distributed with a mean of about 88. There are no lower outliers and only 35 upper outliers, representing 0.0475% of total observations. It appears that any wine given a rating of 98 or higher was deemed an outlier in this dataset. For the same reason as the price outliers, I will not exclude them from the analysis.

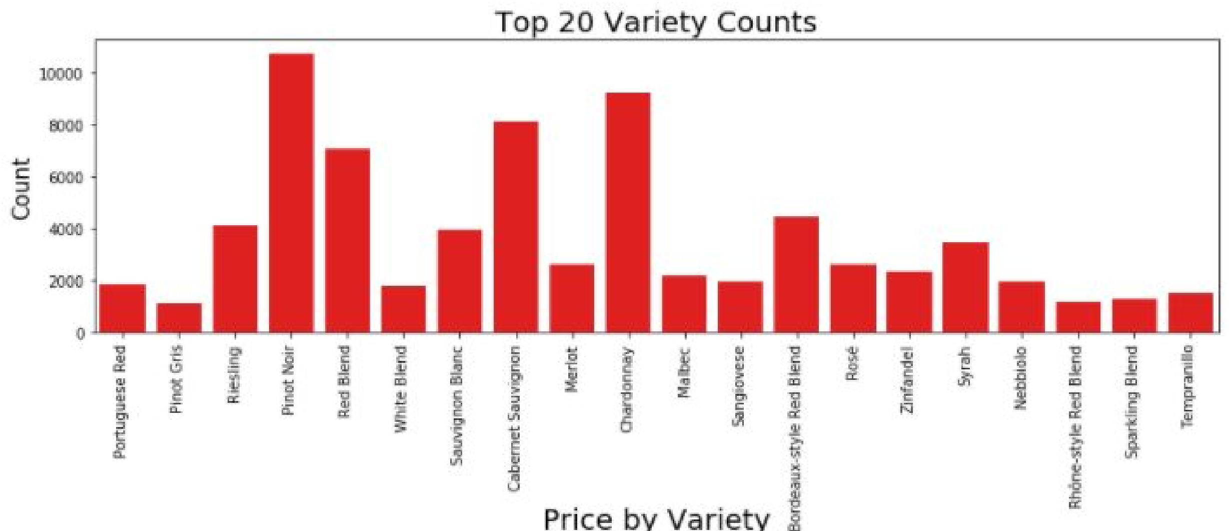


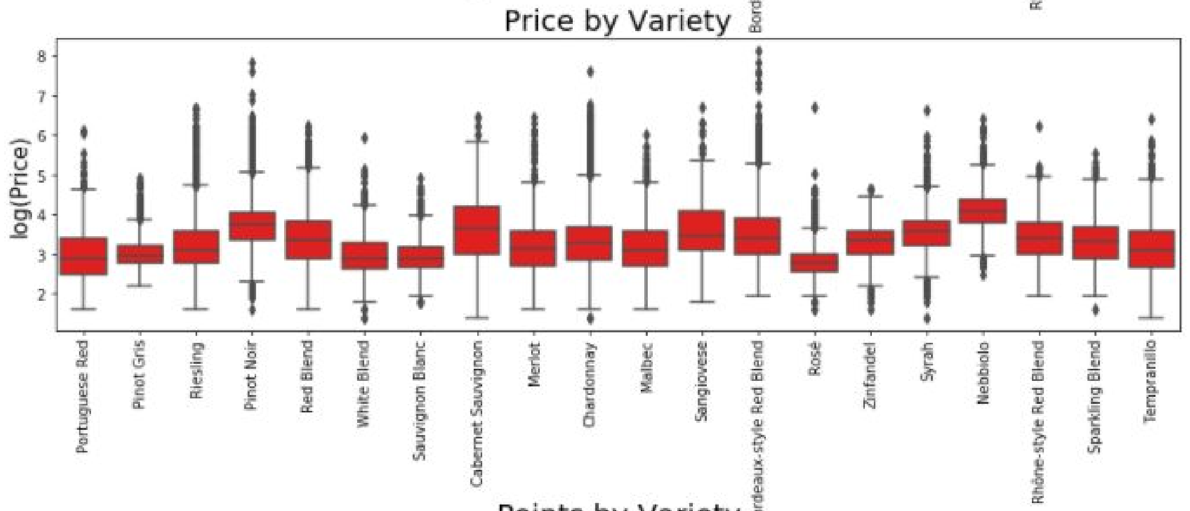


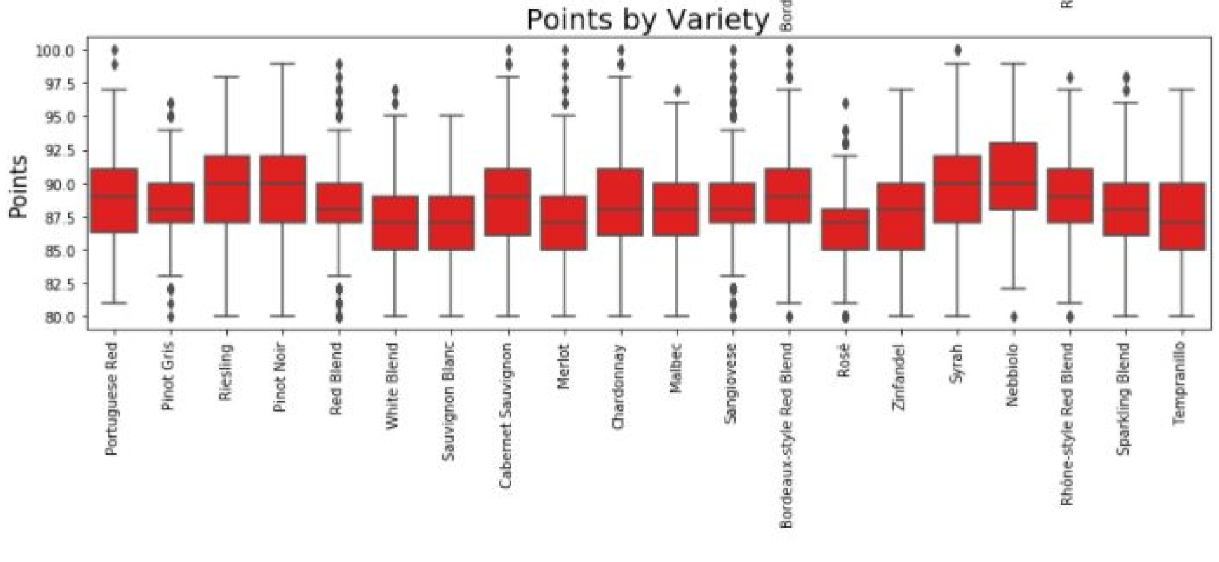
I also wanted to see if there was a correlation between price and points, so I made a joint plot with the points data on the x-axis and the price data on the y-axis. Points and price are generally positively correlated, though it appears there are some relatively affordable wines that are highly rated. Most interestly, the highest priced wine in the dataset at $3,300 was only given a rating of 88, which is around the mean of the points data. It lends credence to the possibility that not all expensive wine is worth the price.



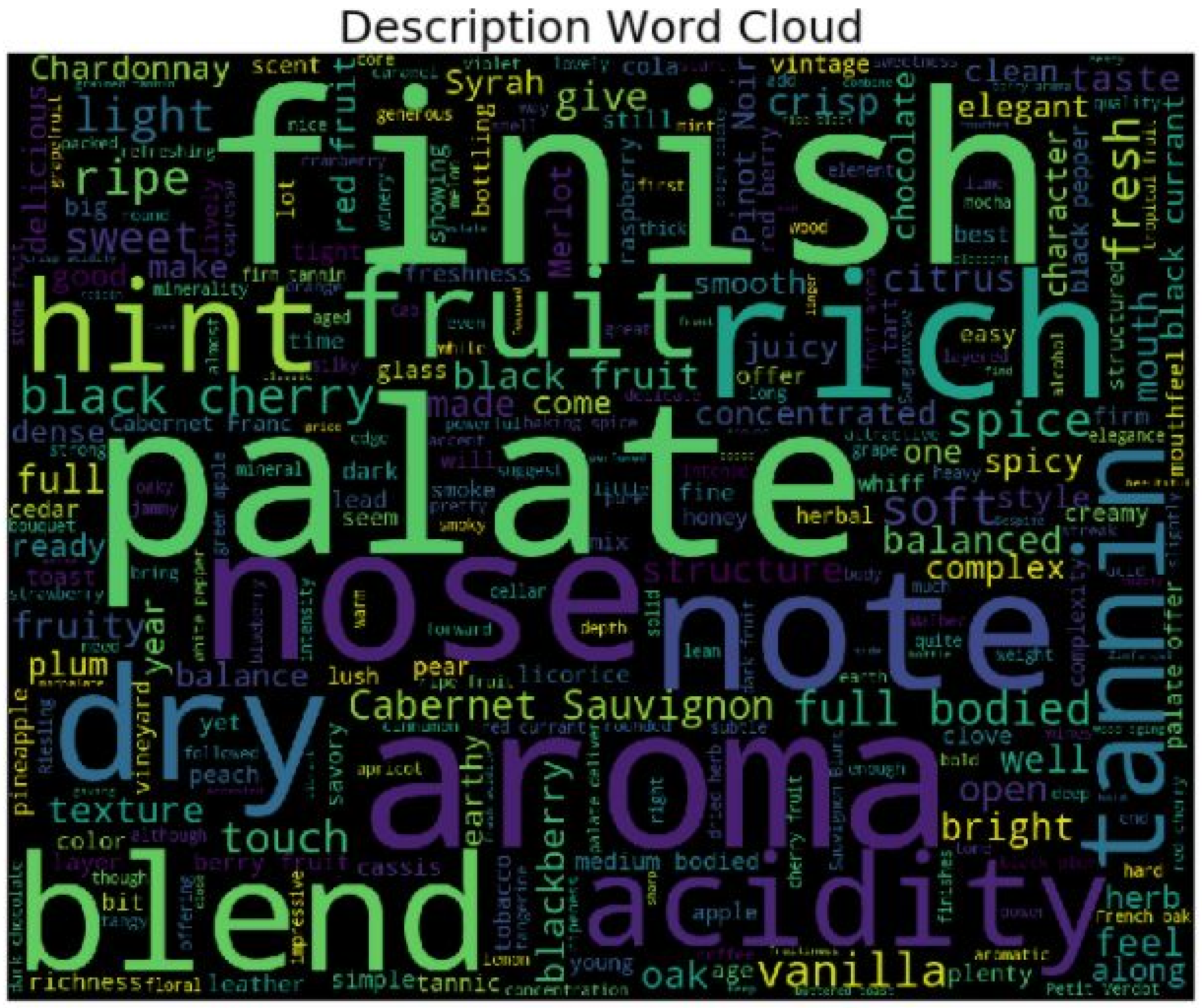
Without doing any machine learning, I thought the easiest way to make wine recommendations in practice was by recommending the same or similar variety, so I wanted to see a breakdown of the top 20 varieties in the dataset, along with boxplots of the price and points grouped by variety. The top 5 varieties in the dataset are pinot noir, chardonnay, cabernet sauvignon, red blends, and Bordeaux-style red blends. Once again, I used a logarithmic scale for the prices in each variety since there the range in prices is high, even when grouped by variety. The 5 most expensive varieties on average are nebbiolo, cabernet sauvignon, Bordeaux-style red blends, pinot noir, and sangiovese. The 5 highest-rated varieties on average are nebbiolo, riesling, pinot noir, syrah, and Rhone-style red blends.


Top 20 Variety Counts
Count
0
2000
4000
6000
8000
10000
Portuguese Red
Pinot Gris
Riesling
Pinot Noir
Red Blend
White Blend
Sauvignon Blanc
Cabernet Sauvignon
Merlot
Chardonnay
Malbec
Sangiovese
Bordeaux-style Red Blend
Rosé
Zinfandel
Syrah
Nebbiolo
Rhône-style Red Blend
Sparkling Blend
Tempranillo

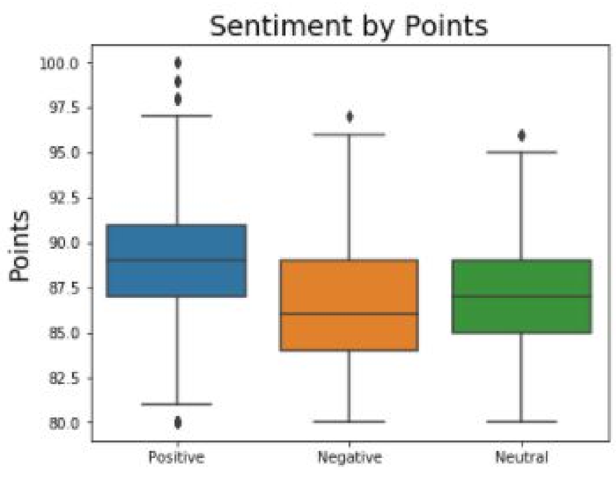

Price by Variety
log(Price)
2
3
4
5
6
7
8
Portuguese Red
Pinot Gris
Riesling
Pinot Noir
Red Blend
White Blend
Sauvignon Blanc
Cabernet Sauvignon
Merlot
Chardonnay
Malbec
Sangiovese
Bordeaux-style Red Blend
Rosé
Zinfandel
Syrah
Nebbiolo
Rhône-style Red Blend
Sparkling Blend
Tempranillo

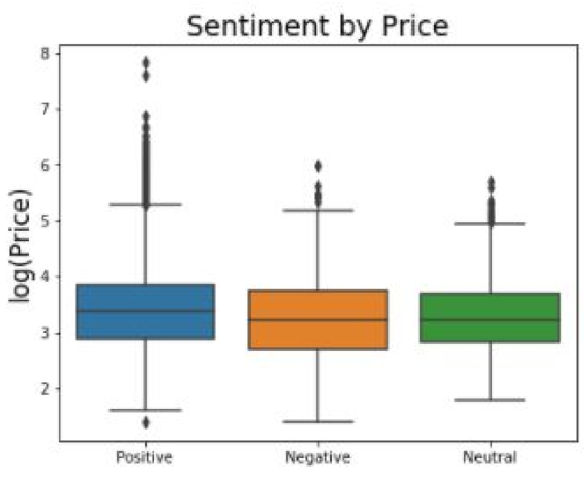

Points by Variety
Points
80.0
82.5
85.0
87.5
90.0
92.5
95.0
97.5
100.0
Portuguese Red
Pinot Gris
Riesling
Pinot Noir
Red Blend
White Blend
Sauvignon Blanc
Cabernet Sauvignon
Merlot
Chardonnay
Malbec
Sangiovese
Bordeaux-style Red Blend
Rosé
Zinfandel
Syrah
Nebbiolo
Rhône-style Red Blend
Sparkling Blend
Tempranillo

I also thought it would be interesting to make a word cloud of the sommelier reviews to see which words tended to be used more to describe wines. The top words look to be “finish”, “palate”, “nose”, “aroma”, and “rich”.



I also performed a sentiment analysis on the sommelier review using SentimentIntensityAnalyzer from nltk.sentiment.vader package. Using a random sample of 20,000 wines from the dataset, I obtained the polarity score from the Sentiment Intensity Analyzer on each wine and populated a new dataframe with an indicator column that read “Positive”, “Negative”, or “Neutral”, depending on if the polarity score was positive, negative, or 0.

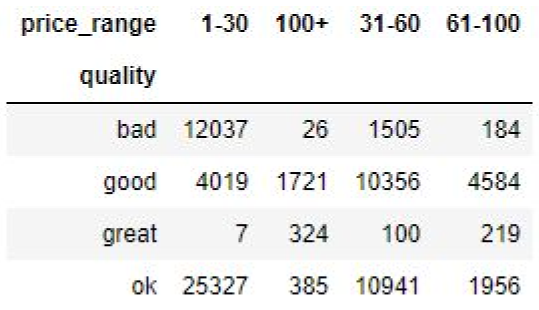




There appears to be a more positive sentiment among the higher rated wines and a more negative sentiment among lower rated wines, though there is less variation on sentiment when grouping by price.

Inferential Statistics

In order to test for statistical independence between the price and points columns, I first wrote functions to assign categories to each wine based on it's points rating, as well as categories for price range. This allowed me to summarize the data in a contingency table shown below.



price_range	1-30	100+	31-60	61-100
quality				
bad	12037	26	1505	184
good	4019	1721	10356	4584
great	7	324	100	219
ok	25327	385	10941	1956

Then, I performed a chi-squared test with the scipy.stats package. The test statistic was very large at 25,185.857, which was much larger than the critical value of 16.919. The p-value was also less than the chosen alpha, so I rejected the null hypothesis that these two variables are independent.

Machine Learning

The first test I ran is a logistic regression to predict the wine quality category. Using the scikit-learn package, I first used the CountVectorizer to create a vector representation of the sommelier description. Then, I created a training set and a test set with train_test_split, with the features of the vectors as the X, and the wine quality category as the y. After fitting the logistic regression model to the training set, I obtained the accuracy score of the test set, which was decent at about 71%. To test the model, I obtained probability estimates of the first wine in the dataset, which was in the "ok" quality category. The model predicted with 73% probability that this wine would be in the "ok" category.

I also ran this same model, but to instead predict the grape variety. Using the same steps, this logistic regression model yielded an accuracy score of about 68%, slightly lower than the accuracy score when predicting quality. The probability estimates on the first wine in the dataset, which is a Portuguese Red variety, was about 48% for Portuguese Red and 35% for Bordeaux-style Red Blend, so while this is a less effective prediction than wine quality, it is still accurate on this particular wine.

Recommendation Tools

The first recommendation tool I built utilized the TfidfVectorizer and linear_kernel tool from the scikit-learn package. I created a TfidfVectorizer object and fitted it to the description column of the training set. I only trained on 5% of the data in this case due to the amount of memory required to train on more. I then used the linear_kernel tool to create a matrix containing the cosine similarities of each description. I populated an empty dictionary with the wine ID number as a key, and a list of similar wines based on the cosine similarities of the sommelier descriptions. Then, I wrote simple functions to extract the wine ID number from the dataset, as well as give a list of recommendations of different wines based on the given ID number.

The second recommendation tool I built utilized the K Nearest Neighbors algorithm. I dropped all of the columns from the wine dataset besides the province, variety, points, and price. I then made a pivot table from the dataframe with the variety as the index, the province as the columns, and the points and price as the values. I then made a compressed sparse row matrix with the scipy.sparse package from the wine pivot table, and fit a nearest neighbors model to the matrix. Using 10 as my number of neighbors, I chose a random wine from the pivot table and output a list of recommended varieties based on the variety of the randomly chosen wine.