Capstone 2 Milestone Report:

For my second capstone project, I decided to use the Wine Reviews data set from Kaggle, which was sourced from WineEnthusiast. This data consisted of over 130K wine reviews, which include detail on the wine variety, region, points awarded, price, country, as well as a description. This data set was spanning many different wine varieties and regions, which gave a robust demographic of wines to compare. My goal for this capstone was to see if keywords in the wine descriptions had a correlation with the points awarded to a given wine. This would be of interest to a wine shop owner as they may want to stock wines with specific keywords to boost sales and customer satisfaction. There were also other comparisons I wanted to review, such as are points and price correlated? Also, what is the relationship to points and region? These types of questions could help wine sellers; for example, if there is a relationship between region and points of wine, they may want to stock their shelves with wines from specific regions.

The dataset from WineEnthusiast was fairly clean and formatted in neat columns as a CSV, which was a nice start to the project; however, there was some cleaning that I performed. I immediately dropped some columns of no-use to this project such as winery and designation. I took a look at the wine volumes across countries and saw that US led the charge with over 60,000 wines. Italy came in second with 20,000, so quite a discrepancy there. I also did some EDA on the top wine varieties per country and saw that USA's Pinot Noir, Cabernet Sauvignon, and Chardonnay led the group.

Next, I reviewed NANs for the data-set and saw that there were NAN values in the country, price, and province columns. Upon further exploration, I saw that the country and province column only had 5 NANs, so decided to drop these due to low volume. Price, on the other hand, had quite a few NANs (13,695). Contemplating how to treat these values, I took a look at the mean price and saw it was 33, which was pretty reasonable. I decided to fill these NAN values with the mean wine price per variety. After filling those values, I took a look at the price distribution and saw that the min was 4.00 and max was 2300. The mean was 33 and STD was 34, which is a pretty high STD. With the max being 2300, I knew that for any testing I did with price, I may need to drop outliers. I reviewed the NANs again and saw that there

were still a small amount missing a price. There were only 21, and these were for unique wine varieties that did not have another wine price to use for comparison. Since the volume was so low, I decided to drop these values as well.

After treating the NAN values, I wanted to take a deeper look at the wine varieties and saw that there were 619 wine varieties, and there were quite a few that were the only wines of their kind. Looking at the top wine varieties, those consisted of varieties that are household names, such as Cab Sauv, Chardonnay, and Pinot Noir. Knowing that is helpful because those are good varieties to look at specific words for.

From here, I began creating the bag of words from the NLTK word_tokenize functions. I decided to review the wine description column as a whole and review the most commonly used words. I split each description into words and saw that 'and' and 'a' were most commonly used. After removing stopwords, I was able to get a better picture of the top words (wine, aromas, ripe, blend, fruit). This had some useful descriptive words, but also wanted to remove some generic words such as wine, aromas, and blend. After removing those, I saw that the top words were ripe, fruit, nose, rich, dry. I created a word cloud to visualize data and saw that there were some descriptive words, such as sweet, dark, fruity, cherry, but there were also some words that were not helpful, such as pinot, cabernet, and shows.

I took a look at the top wine varieties again because I wanted to review specific words for top varieties. I saw that Chardonnay, Pinot Noir, and Cab Sauv were the top three, so decided to repeat the word tokenization/ word cloud steps on all three varieties for comparison. Top words for Chardonnay were chard, oak, apple, crips, oaky, acidity. Top words for Pinot Noir were dry, cherry, light, red, silky, raspberry. Top words for Cab Sauv were dry, cherry, tannic, black, blackberry. I was able to see that each variety seemed to have specific descriptive words, which was good to see. This would be of interest to a wine seller to see if there is a relationship.

After getting a basic understanding of the bag of words, I went back to do some pre-processing on the wine descriptions. I used WordNetLemmatizer as well as PorterStemmer to stem and lemmatize the words. This is useful because stemming breaks words down into the roots such as amusing, amusement into amus. Lemmatization groups similar words together such as good, better, best into good. This is useful to enhance our bag of words and show the true words of interest.

After running these through the wine descriptions, I did not notice much of a change between my initial bag of words created. One thing I did notice was that there were certain words that came up that seemed irrelevant, such as wine names like pinot, cabernet, chardonnay. I wanted to play around with different ways to review the bag of words and compile as I found a lot of different resources online.

I also wanted to create a column in my dataset that contained the cleaned descriptions so that this could be used for modeling. I used a lambda function on the wine descriptions column to create a new 'description_cleaned' column of stemmed words. This is useful because now I can use this column for the modeling section. I also created a boxplot of country and points to see what the distributions looked liked, and the US, Italy, and France had wines that reached as high as 100 points, but also as low as 80, so a fairly large range. I also created a scatter plot of price versus points and noted not that strong of a relationship between the two variables, but the price outliers seemed to be throwing the graph off. I created another scatter plot of wines priced 100 and less, and these had a more significant correlation.

I calculated the covariance of price and points, which showed the variables were related but not as significantly as I would have expected at 50.8. I performed a linregress on price and points to obtain some summary statistics such as p value and r value. I also calculated the pearson and spearman coefficient, pearson was 0.45 and spearman was 0.58, so these variables are related. Those coefficients are useful because they show how closely two variables are related, so we can see price and points are related. I performed a Poisson distribution of wine price and saw that the wine from data was 33.5 and predicted price was 33.4, so very similar. I also performed a Poisson distribution for wines with points greater than 90. The mean of data was 52.7 and predicted mean was 52.65 so also very similar. I charted a histogram and saw that the prices ranged from 40-65, with the majority around 50, which matched our data. The poisson distribution is a useful test because it shows how likely an event is to occur, so based on the results, we can see that the predicted values are very similar to the actual values.

I took a deeper look at the distribution of wine prices greater than 90 points, and saw the distribution was large with a STD of 52. Top 20 most expensive wines range from 800-2300, which seemed to be skewing the data.. I also performed a random normal distribution and saw that the range was between -100 and 200 for

price with the majority hovering around 50. I believe the negative values are coming from the large STD. I re-ran the test with the outliers removed (focused on prices 500 and below) and saw that tightened the range, with this being between -50 - 125. The STD is still fairly high at 34, which is accounting for the negative values

I also performed a couple of ANOVA tests looking at two variables to observe the statistical relationship between them. This was helpful to get an idea of relationships to test in the modeling phase. I first performed an ANOVA test of price and variety and saw that the r-squared value was 0.11, which is fairly low, so variables do not seem strongly related. I also compared wine points and provinces and saw that the r-squared value was again low but p value was 0.0, which signifies a relationship between those two variables. I also compared points and variety, and saw that these variables once again had a low r-squared value at .10, but still had a significant p value showing some type of significant relationship between these variables. Lastly, I wanted to compare the points and price, since these seemed to have a good correlation. I calculated the Pearson correlation coefficient as well as performed an OLS test between the two variables. The Pearson correlation came back fairly strong, with a 0.45. The r-squared also came back strong at 0.495 with a probability of 0.00. This shows a pretty strong relationship between the two variables, which will allow for more modeling.

For the modeling section, I had a lot of different tests I wanted to perform given the trends I had observed during the EDA of this project. I started out with running a CountVectorizer on the description column of the data-set along with using train, test, split. I wanted to compare the wine descriptions against the varieties as I noticed clear differences between the descriptions when I was creating the bag of words. After fitting and transforming the data, I used MultinomialNB as my classifier, since this is a useful classifier for words counts and trends. After running through the classifier, the score was 0.53, which is a pretty high score comparing those two features. Since I had created a cleaned column that consisted of stemmed words (description_cleaned), I re-ran this model using that data and saw an increase in my accuracy score, which was 0.5399. This showed that the pre-processing steps of stemming and lemmatization were useful as increased the accuracy of the model.

Since I had also seen a relationship between price and points, I decided to perform a linear regression between those two variables. I also scaled the data and performed pre-processing steps prior to fitting to the model. I used a successful code that I had used in my previous capstone to compare these variables and saw that the accuracy score was 0.1973, which was not very high but still somewhat significant. This data still contained the outliers, so I re-ran the model with the outliers removed (prices over 500) and saw that the accuracy score bumped up to 0.29598, which is a great improvement. Points and price do not have a very strong relationship, but there is a moderate correlation between these variables.

After looking at the Naives Bayes classifier, I saw that description was very correlated to variety, which made sense because I had seen similar trends with my bag of words. I decided to compare the wine descriptions against points and obtained an accuracy of .284, so somewhat significant relationship. I decided to run another model of points and description, but focusing solely on the top three varieties. This has a beneficial impact on the model, with an accuracy score of .32. I also explored the relationship between country and points, since countries like USA, Italy, and France had wines with the highest points; however, did not see a high accuracy returned on this model (0.139). This made sense since I had seen very large points ranges across the different countries, so not a consistent trend between those variables. I also took a look at the trend between points and province and also did not see a high score, returning 0.1489.

From there, I wanted to deepen my bag of words and create a TF-IDF model because there are common words that should be omitted from the bag of words, such a wine and pinot. Reading through some documentation, I had read that the TF-IDF model works by removing top words that are not unique. I created my corpus of words by splitting the cleaned wine descriptions that I had created during the EDA and assigned each word a token ID. I then fit the TF-IDF model on the corpus to obtain an array of tokens and score; however, this was causing issues on sorting because the tokens were contained in different dimensions and could not easily flatten and sort to obtain the highest rated words. I did some research and found that the TF-IDF vectorizer should meet my needs by returning the top words.

I used the TF-IDF vectorizer on my cleaned description column that I had created during the EDA. I then enumerated the data to get the feature name for

each token that was generated. This would allow me to identify the words with the associated token ID. Printing the matrix, I could see that the vectorizer was returning the word data in a cleaned list, which could then be sorted. I wrote a for loop that would sort the list by rank and then tag the feature name (or description word) to each token so that I could generate a list of highest ranked words that appeared in the description. Oddly enough, wine still came out as the number one word, yet I had thought this would be removed after running the data through the vectorizer. I did see an improvement on the words that came back using this model. For example, other top words were flavor, fruit, acid, finish, cherri, tannin, dri, aroma, ripe. The descriptive words improved by using this model in comparison to the bag of words I had created previously in the EDA section.

The last forms of modeling I wanted to test were comparisons with price. I was having trouble using this column in the MultiNomial NB, and after much trial error and research, finally determined that the errors were due to the data-type of the price column. The price column was actually data-type float64, which was throwing the model off. I changed the data-type to int64, and that solved all of the issues. Another option that I was going to try prior to discovering that was binning the prices into categories and then comparing them that way; however, that was not needed as the data-type was able to be changed.

After changing the data-type, I was able to use the train, test, split and Multinomial NB functions to compare the price and description cleaned. The accuracy score came back as .162, which indicated not a strong relationship. As there were price outliers that has been discovered during the EDA, I removed anything priced over 100 and re-ran the model. This marginally improved the model, with a score of 0.164. I had also seen that Chardonnay, Pinot Noir, and Cab Sauv were the top three wine varieties, so I decided to just look at those three types for the modeling. Re-running with specifically those varieties selected improved the model, returning an accuracy score of 0.185.

From there, I wanted to see if there were any significant trends between price and other factors, so I modeled the price against the country, returning an accuracy score of .07. I also modeled price against the province, which returned an accuracy score of 0.10, so not strongly related.

After modeling with Multinomial Naive Bayes, I was not satisfied with the accuracy scores I was getting, so turned to Random Forest Regressor to see if that would fit the data better; Random Forest is an ensemble method, which usually works better on modeling since it combines different types of models and essentially averages them. I also used the RF with a count vectorizer as I had done previously with the NB method. I first compared points with the description, and this returned an accuracy score of 0.701, which was a vast improvement from the NB model. I also modeled the price and description using RF and count vectorizer, and this returned an accuracy score of 0.302, which was still on the lower side.

I also wanted to try using the TF-IDF vectorizer with RF rather than the CountVectorizer. Re-running the RF models with TF-IDF returned an accuracy score of 0.707 for points and description, which was basically no improvement with the model. Price and description returned an accuracy score of 0.296, so similar comparison. Since the price and description were not responding well, I removed any outliers, which I deemed wines priced over 100. This returned an accuracy score of 0.547, which was a large improvement from the previous models. Looking at these accuracy scores, I feel confident in stating that there is a correlation between the description and points/price.

Since it is in the wine sellers best interest to boost sales and try to cater their stock to wines that people will enjoy, it seems that the points are an important variable to consider. During my modeling, I noted that points and price have a somewhat strong relationship with an accuracy score of 0.30. I also noted that points and description have a fairly strong relationship, with a top accuracy score of .71. That displays to me that points and description have a strong relationship, so wine sellers should look to select wines with the top words identified, as these will most likely result in higher pointed wines. Since there is also a relationship between points and description, it would be in the wine sellers' interest to select wines with those keywords that were identified in the TFIDF vectorizer(fruit, acid, cherry, tannin, spice, sweet). I would recommend that the wine sellers look to stock wines with higher points as they are somewhat correlated with higher prices and will thus boost sales. There also is a fairly strong relationship between price and descriptions, so by choosing wines with those keywords should result in higher priced wines.

I also demonstrated in my modeling that there is a strong relationship between wine variety and description, so I would also recommend the wine sellers to use the keywords identified in my bag of words for the top three wine varieties. The top words varied across the different wine types, which showed that there are different aspects of the top wines that should be catered to.

As next steps for the capstone, in order to build a better model to show the anticipated customer reaction to different wine points/ prices, actual wine reviews and ratings from consumers would be useful. From there, I would be able to create models that would be based on the actual customer review and can explore any trends and relationships between the wine varieties/ points/ price and the customer rating and review. This would ultimately help the wine sellers determine how to stock their stores to boost sales as it would be based on customer data.