Report Details:

The report consists of group and individual portions. The report should be single-spaced, 12pt font, in the following format:

- **Introduction** (written as a group- approx 1 page): Describe your research question(s) and discuss the project context (what makes it important, to whom is it important, etc)
- **Dataset** (written as a group- approx 1 page): Describe the raw dataset (where did you obtain it, how many rows, how many features, what are the features, etc)
- **Introduction to Approaches** (written as a group- approx 1 page): Describe each team member's role in the project, introducing the work that will be described in detail in the individual sections to follow
- **Individual Work** (between one and two pages per section per group member): Describe the work that you as an individual contributed to the project. Make sure to discuss your contributions in detail, noting (where applicable) how you applied concepts and techniques learned in class. Please include your name on each of your individual section pages.
- **Conclusions** (written as a group- at least a page): Synthesize your individual work and provide findings and conclusions. Please include a specific sub-section on **Real-World Implications** of your project.

Grading:

- 30%: Quality/content of group report sections (graded as a group)
- 40%: Quality/content of individual section (note- please don't copy-paste code into the report!)
- 30%: In-class Presentation (graded as a group)
    - Presentations will be graded on the quality with which they communicate the idea. If you haven't completed the entire project by the time you present, make sure to indicate how far you've gotten and what remains.

If you are an INFO254 student but there are members in your group that are not, you, and any other INFO254 students are responsible for the intro, dataset, and approaches section, and you are individually responsible for writing about the work that you did. You do not need to describe the work of any DATA144 group members at length, but you may need to summarize it in order for the write-up to be coherent.

Introduction

“Wine for Newbs” is a project started with a purpose in mind to provide a useful feature that many people can use on a daily basis. Wine is a drink that a lot of us encounter a lot of times - People drink wine during happy hours, during meals to accompany foods, and by itself just like how many people drink coffee. While a lot of us drink wines a lot, not too many of us actually know what are considered “good” wines, or even wines that suit our taste. Most people, especially those in their twenties, just end up picking whatever they find “nice” from the liquor store or grocery store based on the price or packaging, or ask servers or sommeliers for recommendations not even knowing what their preferences are. Wouldn’t it be nice to be able to pick a wine that you’d know how it should taste like, than having every bottle opening as a surprise all the time? “Wine for Newbs” is targeted to people who do not have much knowledge in wine or anyone interested in knowing more about wines. Our feature is a go-to search engine for those people to solidify their preferences in specific wine variety, brands, and gain general knowledge in how “good” the wine is (how the wine critics could perceive the wine).

The project answers following research questions:

1. Can I get wine recommendations based on features of specific wines that I liked in the past?
   ex) Can I get recommendation on wine that is dry, oaky, fruity, and has a tint of chocolate?
2. Can I get recommendations on wines that are similar to the specific one I like?
3. Is there an easy way to find a good wine based on a quick glance at few visible features on the bottle?

By answering these questions, our team is hoping that the users would be able to gain enough knowledge about wines that allows them to confidently select the wines that they know they like at any occasions and more fully enjoy the taste of wine than before they used our feature.

Dataset

We obtained the dataset from Kaggle wine review thread which was scraped from WineEnthusiast during the week of June 15th, 2017. The raw dataset contains 150 thousand rows of individual wine reviews conducted by wine testing professionals. Additionally, there are 14 features in the original dataset, including: country, description, designation, points, price, province, region_1, region_2, taster_name, taster_twitter_handle, title, variety, winery.

Out of the 14 raw features, only two of them (price and ratings) are quantitative. Both prices and ratings are quantitative continuous data points. While prices range from $4 to $135, ratings are in quantitative that ranges from 80 to 96 as a measurement wine quality. The rest of 12 features are in qualitative nominal format that are descriptive of the wine. If we have to break it down further. These features are describing 1). General property of the wine itself, 2) tasters’

information and their reviews of the wine, and 3). the physical information about the producer of the wine.

General property of the wine consists designation and variety of the wine. Designation describes the specific vineyard of the wine while variety describes the type of the wine such as Chardonnay, Pinot Noir, Sauvignon Blanc etc.) Tasters' information and their reviews of the wine are made of features such as taster_name, taster_twitter_handle, title, and description. Title and descriptions are text reviews made by professional tasters. Lastly, physical information about the wine producer or the winery consists of country, province, region_1, region_2, winery name. Region_1 and region_2 provide more detailed geographic location of the winery such as Sonoma, Dry Creek Valley, California. However, a large portion of region_2 data (61%) percent is missing.

Within these features, wine descriptions and wine variety are used highly in our analytics to understand the less popular wine and their properties, which will be covered in the approach section of this report.

Introduction to Approaches

Our team used various tools and approaches to answer our research questions. Below is a summary of each member's role and contribution in the project:

- Hongyang Zheng
    - Data Scraping using Google Maps geocoding to turn descriptive geolocation data into quantitative data
    - Geographical visualization for wine related insights.
- Grace Chung
    - Predicted whether or not the wines received ratings at or above 88 or below using logistic regression modeling.
    - Utilized feature engineering to combine and drop features useful to predict the accuracy of the model.
    - Normalized the dataset to scale from 0 to 1 to fairly weigh the feature
    - Split up the data to test/train and cross validated to avoid overfitting
    - Used logistic regression to predict the points on the test data
    - Used neural network to predict the points on the test data
- Thyne Boonmark
    - Word2Vec embedding of wine descriptions
    - Used word vectors to represent individual wines
    - Created recommendation function that would take in a list of words that described a wine and return the most similar wine in the dataset

    - Performed clustering to understand natural forming groups of wines based on text descriptions
- Daniel McAndrew
    - Doc2Vec embedding of wine descriptions
    - Used document vectors to understand similarly described wines
    - Aggregated wine varietals to create red/white features
    - Performed random forest regression to predict point score

Individual contributions

Data Scraping and Visualization (Hongyang Zheng)

As mentioned in the dataset section, there are a limited amount of quantitative data in the raw dataset. However, we have very descriptive geographic information on the winery and even the specific vineyard produced the wine. I transformed the descriptive data into quantitative location data measured in latitude and longitude and generated interactive regional and world view heat maps utilizing tools such as Google Maps geocoding API and folium package.
The first step was to clean up the original dataset to create individual geo query entry for geocoding. A series of string concatenation was performed and more detailed specific locations (such as region_2 were preferred over region_2) were utilized by using 'combine_first' function. Since there are no null value in country and winery name, I was able to generate a list of searchable addresses to pass along to google geocoding.
The second step utilized code snippet from github thread python_batch_geocode and modified to scrap longitude and latitude information from google maps. Google Maps geocoding API allows structured geo inquiry with an authoring API_key attached to the end of query url and returns a json file with extensive geo information including latitude and longitude. Additionally, the scraping tool was set up to run on 0.25 second interval to avoid denied access or network latency. The overall performance of the scraping tool was relatively robust with only ~2.5% null results.

Last step involves data visualize with folium package which enables interactive zoom in-n-out heat maps based on geographic locations and wine associated quantitative measurements. Below is an example of visualization in the Bay Area based on simply on count of observations. As expected, the Napa/ Sonoma area has the largest density of wineries. To take a step further, I specified the density heatmap to graph based on rating, price, and rating over price for all wines in the dataset. The visualization generated interesting insights on how to pick wine by geographic location if a new wine taster is trying to get a higher rated wine with the most reasonable cost.

Chart: Winery Density in the bay area

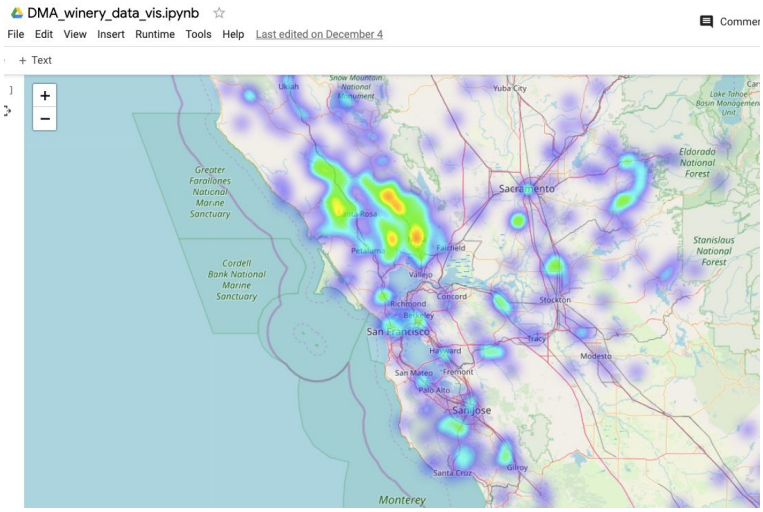
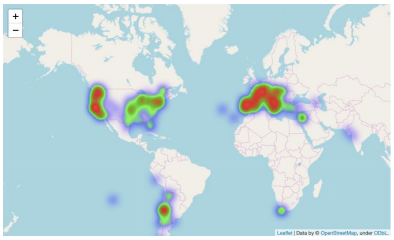
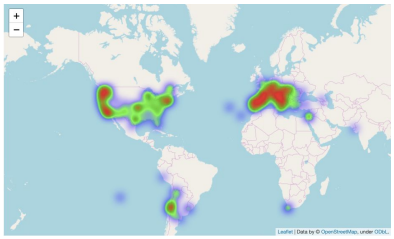
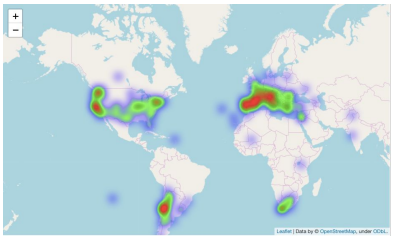


Chart 2: Winery density by rating, price, and rating over price.

Rating	Price	Rating Over Price
		

Regression Analysis (Grace Chung)

The logistic regression analysis was performed with a purpose to answer the following research question: Is there an easy way to find a good wine? While the definition of good wine is murky and subjective, here I defined “good” wine as the ones that received good ratings from wine critics on our dataset. The wine rating on our dataset shows a normal distribution ranging from 80 to 100, with its median at 88. Therefore, I have defined the criteria of good wine as the ones that received critic rating at or above 88, and explored ways to predict the “goodness” of the wine using the following features.

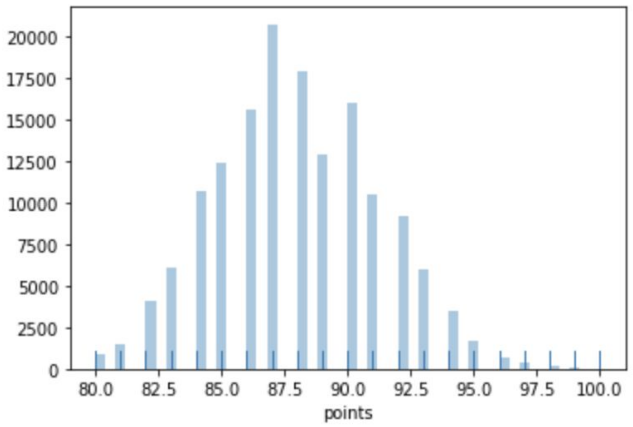


Figure 1. Histogram of distribution of points

- Feature 1. Words in Text Analysis
  This feature uses the description column of the dataset that wine critics wrote about the wine. I used wordcloud to figure out the most used words for wines that are at or above 88 points and below 88 points and created a list that clearly distinguishes the two. I have also manually looked through the data set to find frequently used words used for highly rated wines into the list. Then, I created a function called words_in_text that creates a matrix based on the list of words and assign 1 or 0 depending on whether the word was actually in the description or not. Here is the list of words that I predicted "good" wines have: ['opulent', 'fabulous', 'impressive', 'beautiful', 'umami', 'tremendous', 'exotic', 'complex', 'perfect', 'valuable', 'dazzle', 'silk', 'outstanding', 'magnificent', 'gorgeous', 'powerful', 'magnificent' , 'wonderful', 'exceptional', 'glorious', 'impeccable', 'enjoyable', 'elegant', 'mouthfeel', 'profound', 'seduce', 'seamless', 'concentrated']

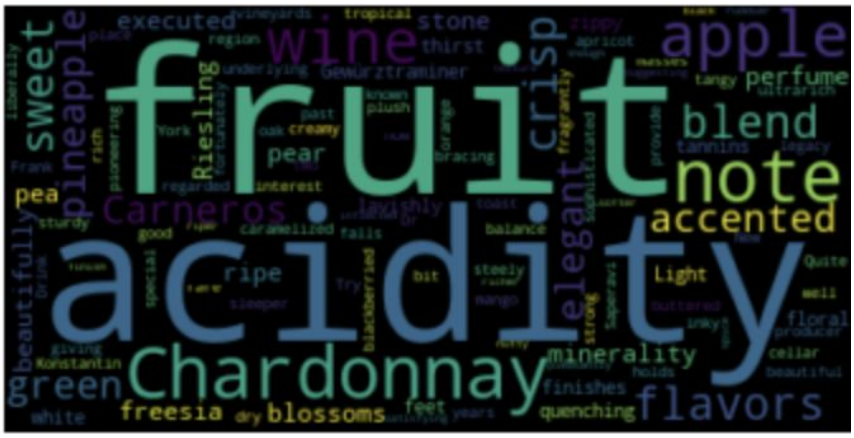


Figure 2. Wordcloud for dataset of wines at or above 88 points

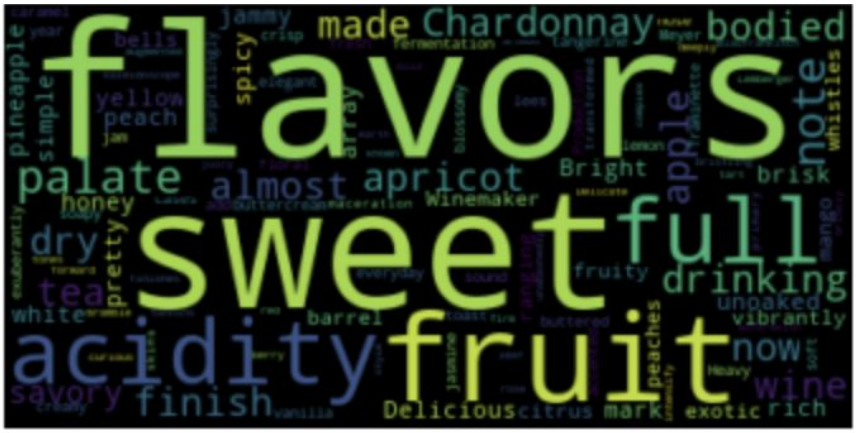


Figure 3. Wordcloud for dataset of wines below 88 points

- Feature 2. Points per Province (pts_province)
  Each wine data point has "province" column that indicates which province the product is from. I have aggregated the average of "points" grouped by "province"

- Feature 3. Average Points per Region 1 (pts_region_1)
  Each wine data point has "region_1" column that indicates which bigger region the product is from. I have aggregated the average of "points" grouped by "region_1"

- Feature 4. Average Points per Region 2 (pts_region_2)
  Each wine data point has "region_2" column that indicates which smaller region the product is from. I have aggregated the average of "points" grouped by "region_2"

- Feature 5. Average Points per Variety (pts_variety)
  Each wine data point has "variety" column that indicates the variety of the wine i.e. Chardonnay, Riesling, etc. I have aggregated the average of "points" grouped by "variety."

- Feature 6. Price of the Wine (price)
  Each wine data point has "price" column that indicates the price of the wine.

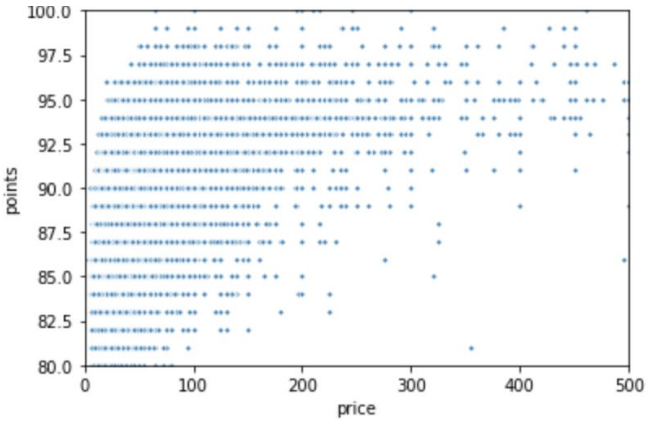


Figure 4. Scatterplot of price vs. points

After determining the features to consider using for the modeling, I have split up the dataset "winemag-data_first150k.csv" by train and test set. For trainning set, I have further split up the set train and validation set by 5 folds cross validation to reduce overfitting. Then, I decided on which features to include to my model. I have initially decided to include features 2 to 6 and drop feature 1. While feature 1 is a great feature that clearly is related to the points, I decided to drop this feature on my final model because I did not see much value of how users would use this in a daily setting. Since most users will pick and choose the wine solely on the information written on the packaging to determine which wines to pick, wine critic description would not be a deciding feature most of the times. For the 5 features used, I have normalized the data to have all features be scaled the same to 0 to 1. This was done since price feature showed big scale difference ranging from 4 to 2300 compared to other features.

After fitting the training data into the logistic regression model, the training accuracy came out to be 68.65%. I looked at the feature importance in the prediction.

```
Logistic Regression (L2) feature Importance:
pts_region_2: 0.0749
pts_province: -1.0214
pts_variety: -1.6913
pts_region_1: -4.7709
price: -30.2073
```

The results showed that pts_region_2 was barely any deciding factor for determining the model. I presume that this could be due to many missing values in that column. I decided to remove the feature from the model and ran again to achieve [0.67408629, 0.69613412, 0.7128985 , 0.6483144 , 0.66031513] training accuracies in the five validations, and 0.62485 test accuracy. I have attempted to do neural network modeling as well, achieving 0.67422 training accuracy and 0.65801 test accuracy. While the modeling has provided not-so-great accuracy, the analysis has provided insight that price is the most significant indicator in determining the goodness of the wine in terms of wine critic's point.

Word2vec Representations of Wines (Thyne Boonmark)

Since each description contained detailed information about the smell, taste, and quality of a wine; we were able to extract these features of wines through natural language processing. To do this, we generated corpus based on all of the wine descriptions, and represented each wine as the average of all the word vectors within its description. This method of representation allowed us to compare wines through cosine similarity of these vectors and also search for the most relevant wine given a few key words.

Generating Word Vectors

We made use of Google's gensim model for our natural language processing. Before creating our own word2vec representations of wine descriptions, we had to first process the data so we could create a higher quality model.

For preprocessing, we first removed any punctuation and stop words (such as common articles and conjunctions) from the descriptions and converted all text to lowercase using the nltk package's tokenizer. Originally we tried keeping all stop words in the descriptions and add them to the word vector model, but possibly due to our dataset not being large enough, most words ended up becoming highly correlated with common articles and this prevented us from finding meaningful similarities between different wines.

Once the wine descriptions were preprocessed, we could tokenize all of the descriptions and create a gensim word2vec model. For the vector representations of words, we decided to use 10 as the length of the vector. This helped simplify our vector data and reduce the size of our data, which maked dimensionality reduction work faster and prevented the code from running into memory errors.

Wine Recommendations Based on Description

We based our recommendation systems based solely on text data from wine descriptions. We decided not to include other features such a region and grape variety since we wanted to make the system easier for total wine novices to use. It would be difficult for people unfamiliar with wine to search get a good recommendation if we needed them to provide a text description, plus many other details that they may not even be familiar with.

For our wine recommendations, we had two main use cases in mind. First of all, if someone has no idea what wine to get but has some idea of what flavor and aroma qualities the wine they want should have, then they should be able to find a suitable wine that matches these qualities. Our word2vec representations of wines allows just that. Given a list of descriptive words; such as "dry", "acidic", "fruity", etc; we can convert these into their word vector representations using our model, take their sum, and use cosine similarity to compare the resulting word vector to each wine's vector representation to find the best match. In the example below, we search for a wine that is most associated with dry, cherry, and chocolate and our query

recommends us the Syrah Lawer wine. This wine is described to be a dry wine with chocolate and berry flavors and appears to be a pretty good match given the original query.

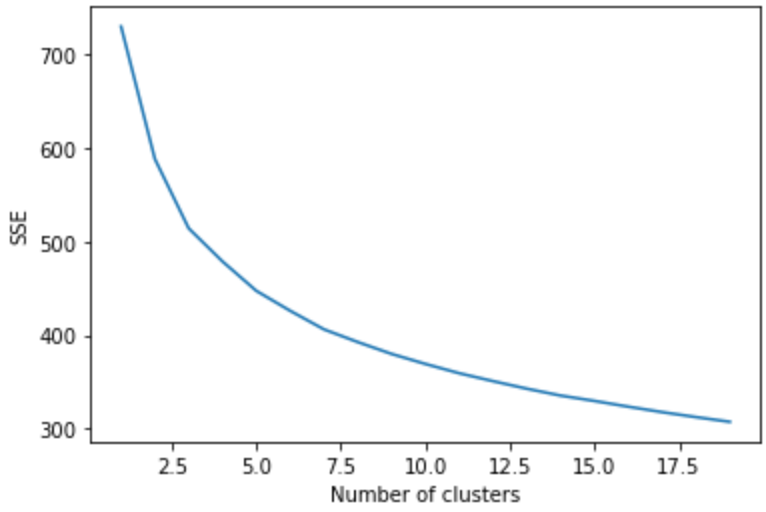
```
winerec = getSimilarWine(wineData, ["dry", "cherry", "chocolate"], aveVectors2, own_model)
```

```
Syrah
Lawer
Plenty of red currant, chocolate, blackberry jam, cola, pepper and cedar flavors in this dry, tannic wine. Drink up.
```

The second use case is for people who have a desired wine in mind but do not have access to it either due to costs or local availability. Given some description of a wine they have in mind, they can search through the dataset for the most similar other wines. We first take their wine description, preprocess it by removing punctuation and stopwords, then take the average of all of these words' vector representations. Similarly to the use case described earlier, we then take the cosine similarity of this wine with all the other wines in our dataset and return the wine with the largest cosine similarity. This will allow someone to find a suitable replacement for the wine that they originally had in mind.

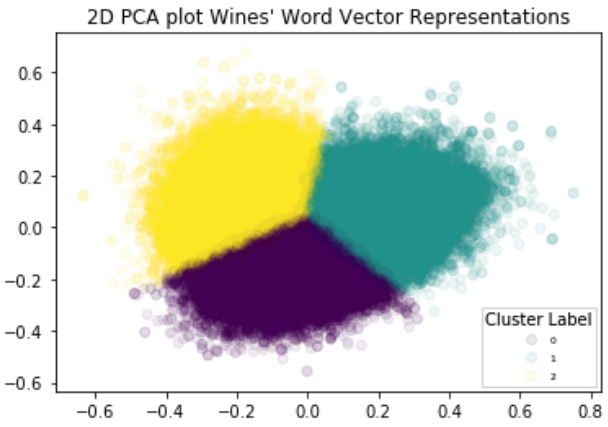
Wine Clustering

Besides providing users with individual wine recommendations, we were interested to see if there were naturally forming groups of wines based on their descriptions. Once we had the word vector representations of wines, we were able to cluster them based on these word vectors. For our clustering, we used the k-means algorithm provided by the scikitlearn package. Since we do not have extensive domain knowledge of wines, we did not have any preconceived ideas of how many clusters we should use. Instead we used the elbow method to decide the number of clusters we would split our wines into.

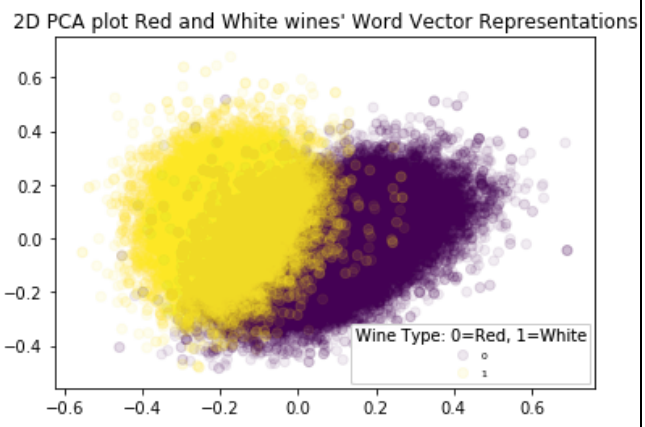


Looking at the SSE vs Number of clusters chart, we can see that there is a pretty gradual curve. However the bend at three clusters appears to have the greatest angle, so we decided to cluster our wine data into three groups.

Dimensionality Reduction

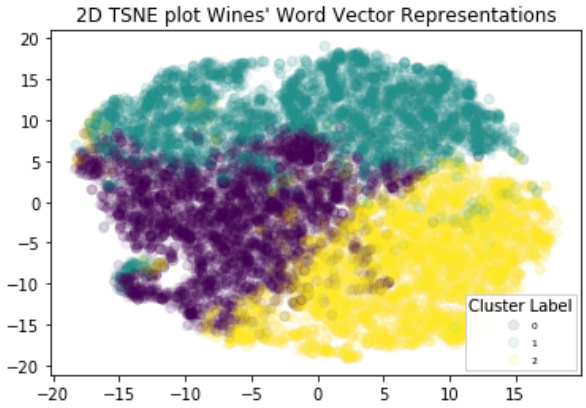


PCA Fig 1

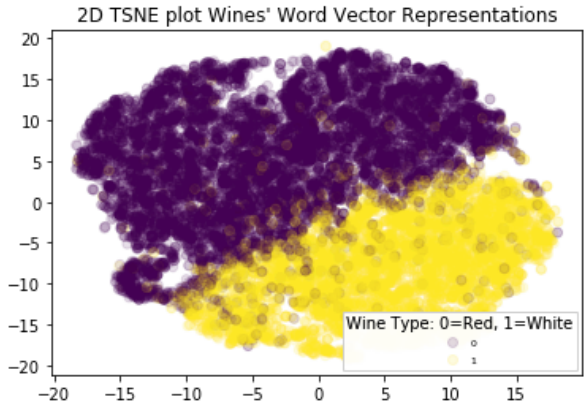


PCA Fig 2

After clustering our data into three groups, we can see that our groups are not that well separated. Looking at figure 1 above, there is a ton of large overlapping areas between the three clusters, the edges of the clusters are blurred and not that distinct from each other. When we instead color each wine by whether or not they are a red or white wine, we can see also see how the two wine groups are not perfectly separated. In figure 2 we can see the two distinct groups of wines (red being data points that are purple and white being data points that are yellow), but there is a significant overlap. This suggests that while text descriptions of a wine are able to differentiate between different varieties, more features may need to be included in order to really create distinct clusters of wine types.



TSNE Fig 1



TSNE Fig 2

Besides using PCA for dimensionality reduction, we also used TSNE to see if this separated the data any better. While we were unable to use TSNE on our whole dataset due to computer memory limitations, we instead ran it on a random sample of our dataset to see how well it differentiated our clusters. In the first figure above, we can see that the three cluster groups are not much better separated compared to when we conducted PCA. However, when we instead color data points by their wine type (red or white), we see much better separation compared to PCA.

Cluster Interpretation

Clusters	word	count
0	black-currant	2919
0	incomparable	1037
0	chêvre	874
0	couched	868
0	char-barrel	828

Clusters	word	count
1	supported	3817
1	matched	3719
1	deranges	2959
1	black-cherry	2587
1	primary	2331

Clusters	word	count
2	butter-cream	3052
2	lifted	2940
2	hinting	1378
2	citrus	1348
2	texture	1033

In order to get a better understanding of the clusters formed from our data, we looked through to see if we could find any common themes among the word vector representations of wines in each group. Since we represented each wine in our dataset as a single word vector, we converted these representations into the most similar word in our original text corpus using our gensim model and then counted the most common words in each cluster.

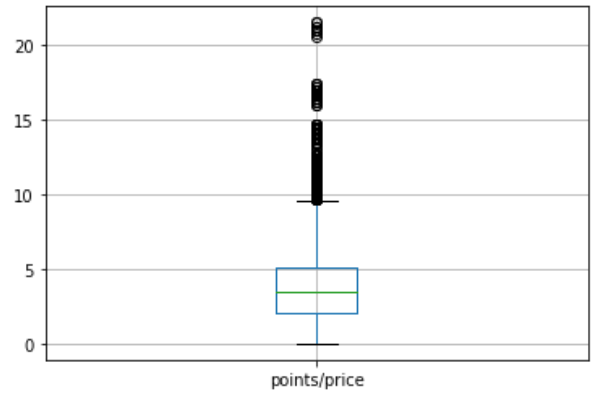
Since our group does not have much domain knowledge about wines, it is difficult for us to make sense of our findings here but at the very least we observe some differences across the three clusters. Cluster 0 has three terms related to taste and aroma of the wine: black-currant, chevre, and char-barrel. Wines in cluster 1 seem to be more associated with black cherries, and wines in cluster 2 appear to be reminiscent of butter cream, citrus, and have a lifted (not too acidic) taste to them. Because the three clusters have different most common descriptors, it seems that there are naturally forming groups of wines based on flavor and aroma profile.

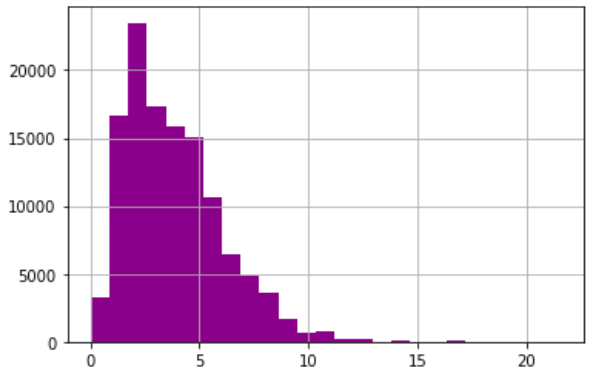
Daniel McAndrew:

Preprocessing and Feature Engineering

I created the features of "red" and "white" by grouping sets of varietals in the original data that belong to each of these aggregated varieties. These were useful for understanding the results of the dimensionality reduction and clustering of the word and document vectors.

I explored the data to understand some of the wine "value" as points per price, that Bobby focussed on more in depth. Below are some of the figures that I created while trying to understand the spread of points per price. The boxplot shows the median and interquartile range of the price per points, indicating that there are many high value outliers which have a notably high rating for a relatively low price. The same trend can be seen in the histogram, where the x axis is price/points and the y axis is frequency. These wines are the best deals and should be recommended. Bobby's more detailed analysis of this engineered feature showed that the value wines tended to correlate with geographic region.





Doc2Vec Embedding

I applied Doc2vec paragraph embeddings to the wine review descriptions. I used gensim's Doc2Vec, which is an extension of the Word2Vec method that we learned in class and is based on the results of the paper Quoc Le and Tomas Mikolov: "Distributed Representations of Sentences and Documents". Just as in the Word2Vec method, the paragraph embedding maps each word in the vocabulary to a unique vector in some vector space with a dimension defined as a hyperparameter of the model by the user, but each document also has a paragraph token which is trained to be a vector in this same vector space and can be thought of as an additional word. The paragraph token essentially acts as a memory that attempts to remember what is missing from the current context. For this reason, the model is sometimes referred to as the Distributed Memory Model of Paragraph Vectors (PV-DM). The figure below illustrates how the words and paragraph vectors are learned.

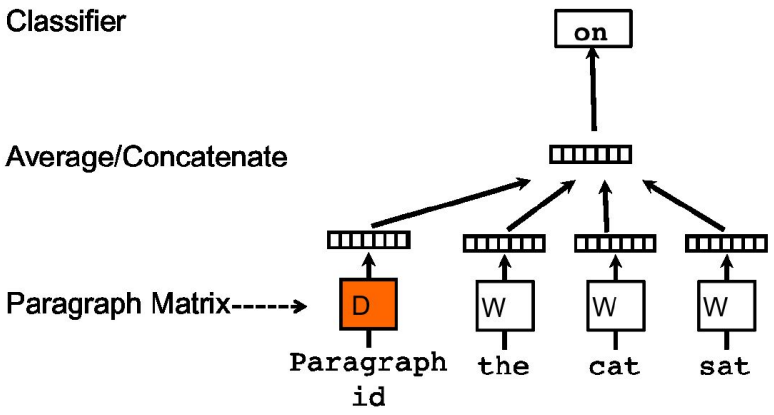


Figure taken from Quoc Le and Tomas Mikolov: "Distributed Representations of Sentences and Documents". Here W represents the word matrix where each word is a unique column of the matrix. Similarly D is the document or paragraph matrix and each document's paragraph token is a column vector of the matrix D. In this model, the concatenation or average of the paragraph token vector with a context of three words is used to predict the fourth word. The paragraph vector represents the missing information from the current context and can act as a memory of the topic of the paragraph.

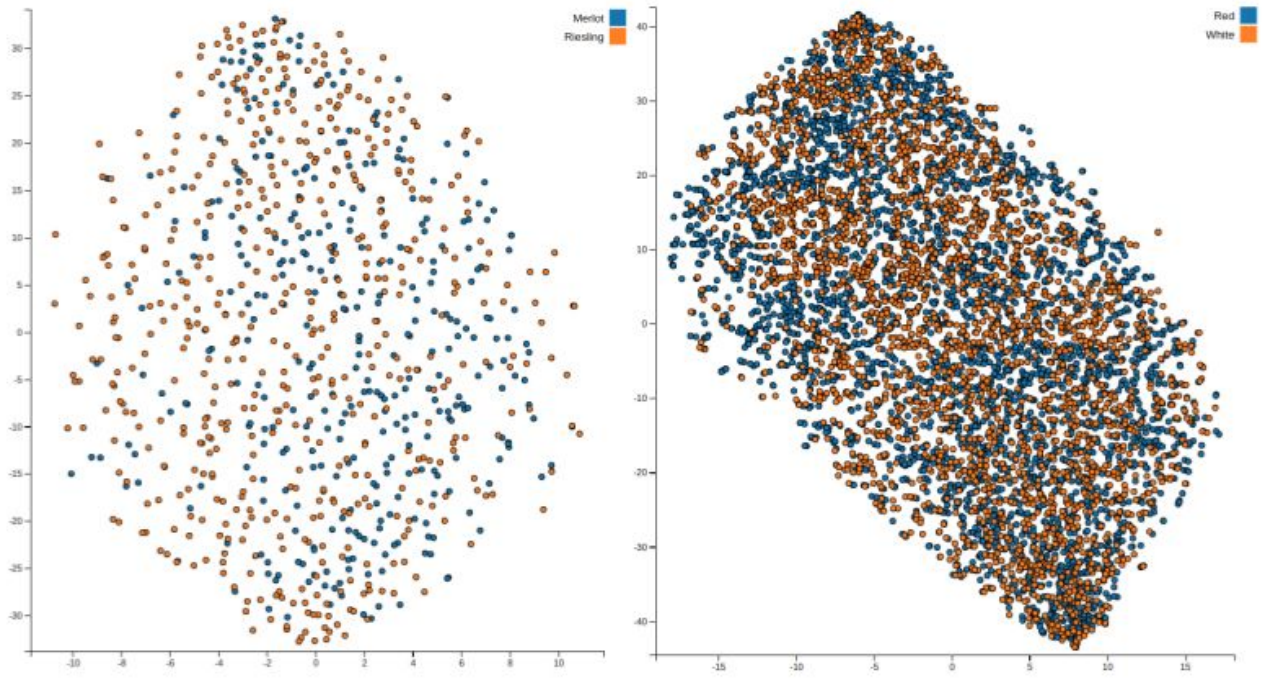
Description Similarity

Using the doc2vec embedding, for each wine, I created a list of most similar wines by description. This could act as a sort of recommendation system so that if a user knows that they like a certain wine, they know which wines have been subjectively described most similarly. An example of the most similar descriptions is provided below:

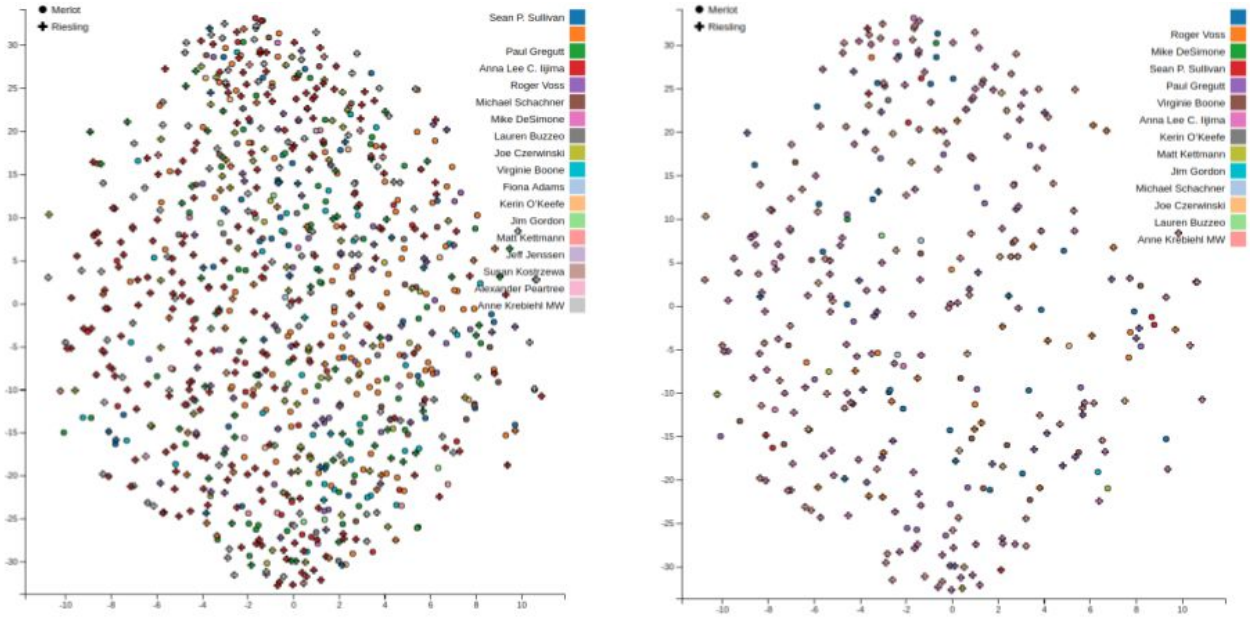
Wine Description	Most Similar Description	Second Most Similar Description
Aromas include tropical fruit, broom, brimstone and dried herb. The palate isn't overly expressive, offering unripened apple, citrus and dried sage alongside brisk acidity.	Apple and spice aromas come in front of a soft palate. Melony flavors are short on zest and finish abruptly.	This wine is buttery and toasty, with sourdough on the nose alongside a hint of clementine pith. The palate is a creamy blend of mango, pineapple, sourdough and sweet cream flavors, with a slightly nutty finish.
This is ripe and fruity, a wine that is smooth while still structured. Firm tannins are filled out with juicy red berry fruits and freshened with acidity. It's already drinkable, although it will certainly be better from 2016.	This Peyronie family cru bourgeois estate is one of the rare properties in Pauillac that is not a classified growth. That alone makes it a good value. The wine comes from vines close to Pichon Baron in the south of the appellation. This juicy black-currant flavored wine is still very young, full of its tannins and dark character. With its structure and acidity, the wine will age well. Drink from 2022.	A tight, tannic wine, showing considerable acidity as well as fresh red fruits. The wine is densely packed with dry tannins, then a fruity character develops, sparked by acidity.
Tart and snappy, the flavors of lime flesh and rind dominate. Some green pineapple pokes through, with crisp acidity underscoring the flavors. The wine was all stainless-steel fermented.	Lemon zest on nose and palate signal a very zippy, citrus-scented white with a rather neutral aromatic expression—as befits unoaked Chardonnay from a cool year. It is light bodied and cannot be beaten for freshness.	The wine is ripe with rounded black-plum fruits over the toasty flavor. It has structure certainly, although this is already integrated into the juicy character of this attractive wine. Drink now.

t-SNE Dimensionality Reduction

We reduced the dimensionality of the document vectors using t-SNE and noted that wines of specific varietals and reviews by specific tasters tended to be clustered together. Albeit in the embedded space, the wines with the same variety or taster were only clustered in small clusters that were not very well separated and did not include the entire set of wines that also had that variety or taster. Interestingly, the clustering for both variety and taster seemed strongest when the wines with lesser point values (below 90) were filtered out.



In the left plot, we can see that the t-SNE embedding of the document vectors illustrate some amount of clustering among specific varietals, but not as much as one might expect. The clustering tends to sometimes relate to specific flavor descriptions. For example, upon further inspection using the d3 visualization this was created with, one can see that all of the riesling wines the mention “blackberry” are tightly clustered. On the right we note that the t-SNE embedding of the document vector indicates some clustering based on the aggregated varietals “red” and ”white”, but the aggregated varietals are not separated very much from each other, rather they form clusters on a more granular level based off particular word descriptors.



The reviews are also somewhat clustered by taster in the 2D t-SNE embedded space. Here the points are colored by taster and shape by variety. This clustering makes sense because the tasters are likely to write in characteristic ways that are reflected in the document vectors of their reviews. This plot on the right is the same as the left embedding but with only the highly scoring wines included (below points = 90 are filtered out). Here we can see stronger grouping of the varietals reviewed by a specific taster, indicating that the reviewers tend to describe wines of a certain variety that they enjoy in unique ways. Although not dramatically so, as evident by the weakness of some of the clustering.

The clustering of varietals and aggregated varietals for the doc2vec model were not as cleanly separated as with the averaged word2vec model that Thyne worked on which shows an interesting way in which these seemingly different similar methods yielded different results.

Additional Regressions

I also ran a random forest regressor on the data. Based on Grace's work indicating that the price was the best feature for predicting point score, I used just price as a feature and achieved a mean absolute error of about 1.92 points on the validation data consisting of 20% of the dataset. Using just the document vector, I was able to get a mean absolute error of about 2.37 points and with points and document vectors together, we got a mean absolute error of 1.91 points (these values were averaged over a few tests). So the document vectors added some additional predictive power but not much. Compared to the standard deviation of 3.04 rating points, which can serve as a sort of baseline to compare the MAE results to, these predictions are reasonably good but there could be room for improvement, if other features were engineered.

Conclusion

Wines are often presented to likely consumers accompanied by various information about the wine, but this information can be somewhat confusing. There are many varieties of wine from different regions and it can be difficult for consumers with little experience to understand the difference between them and make an informed decision about what they should buy. We sought to demystify wine selection somewhat by mining wine reviews. One of the first questions that we wanted to answer was what features are most related to good quality wine.

Our results indicate that price is the most predictive feature in our dataset for determining wine rating, which we used as a proxy for quality. However, despite wine quality being linked to price, there are a number of high value wines which have notably high ratios of rating to price. Visualizing the geographic spread of the features in our dataset including score/price, we were able to identify regions which are more likely to produce high value wine.

We also looked at the text of the wine reviews to find patterns that we could leverage to better understand how certain wines may appeal to consumers with particular preferences. We vectorized the wine reviews and performed some sanity checks to make sure that the vector embedding made sense and noted the structures that they revealed. Unsurprisingly, wines of similar varieties were near one another in this vector space and reviews by particular tasters were also near each other. Notably, we found that the wines could be clustered into certain specific flavor profile groups. Using this vector encoding and clustering, given a particular wine, we were able to provide a list of wines that were most similarly described by tasters. Also, given a set of descriptive words like specific flavors, we were able to find wines that most matched this description.

An extension of our work could lead to the creation of a wine recommendation system to make the processes of understanding wine easier for new consumers. Leveraging our findings, we could design an interface that would allow users to enter specific flavors or wines that they have previously enjoyed and discover other similar wines. The system could also determine if the user's tastes fall within certain clusters of wines or professional taster preferences and then provide them with suggestions from that cluster. Similarly, we could develop a map navigation interface that allows the user to view the geographic distribution of different wines based on specific features and flavor profiles. It could then be used to discover wines that are similar to their preferences and are most easily available in their own area. Such an interactive tool could synthesize the insights we found in the wine review data and provide a useful resource for better enjoyment and easier navigation of the world of wine.