Using Dimensionality Reduction and KMeans Clustering to Find Trends in Wine Quality

Ethan Black, Campbell Moco, Grant Liu, Sam Schwartz

Introduction & Motivation

For our project, we decided to use a combination of KMeans Clustering and PCA dimensionality reduction to see how Wine Features(ie Density, pH, acidity) relate to the overall quality of the Wine as graded by professional wine tasters. We chose this topic for two reasons. Firstly, one of our groupmates recently read an article[1] about the gullibility of winetasters, and secondly, we found the topic interesting. Even though the project itself doesn't solve a worldwide problem, we did think that it could be important. Wine tasters often decide the success or failure of a vineyard, and it is possible that decision is based more on the subjective experience of the taster than any quantitative property of the wine.

Related Work

To prepare for our project, we found a blog post by Andrew Mourcos[2], where he details a similar procedure to ours using sklearn. We used this article as a framework for how we would implement our algorithms. Once we finished clustering the data, we expanded on Mourcos' original ideas by performing a more in depth analysis of what the cluster assignments might correspond to (i.e, quality, type).

Methods

Data Preparation

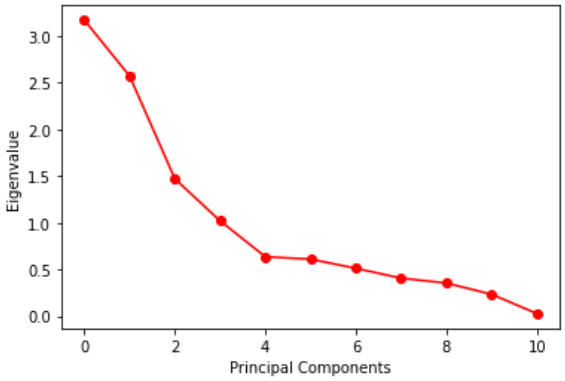
To prepare our data for use, we first loaded it into a numpy array using the built in "getfromtxt" method, using "," as our delimiter since our file was a csv. From here we dropped the header, Wine type (ie. red, white), and quality of the wine from our dataset since a standard PCA implementation is designed for continuous variables. This should also be fine because wine type should have no effect on quality, as a good red wine should be different from a good white wine. After this, we cleared all the data points with missing values because our algorithm was not designed to handle NaN values, and they might distort our data unpredictably. We then loaded the semi-cleaned data into a pandas DataFrame and removed all outliers using z scores from scipy's stats library. Finally, we created a separate numpy array for wine quality and wine type for use in our analysis.

---

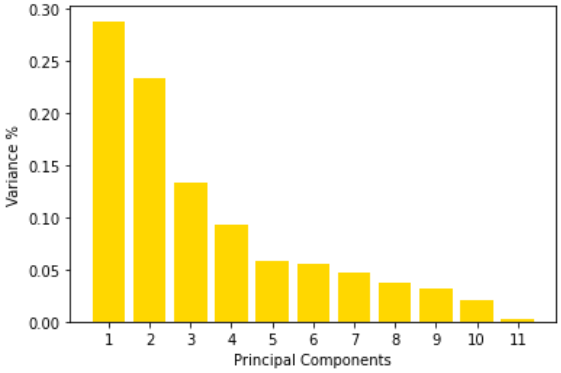
[1] https://gizmodo.com/wine-tasting-is-bullshit-heres-why-496098276
[2] https://andrewmourcos.github.io/blog/2019/06/06/PCA.html

PCA

For our PCA Dimensionality reduction, the first step we took was to standardize the data by finding the mean and standard deviation of each parameter, subtracting the mean value from every datapoint and dividing by the standard deviation. At first, there was some confusion as to if we needed to “normalize” versus “standardize” our data. As long as our data is centered[3], PCA may be able to approximate the data at a lower dimension. Using our normal standardization technique was able to center our data, so we went with standardization rather than normalization. From here, we found the eigenvectors and eigenvalues of the standardized data, and used them to create a covariance matrix. We sorted our eigenvalues and their corresponding eigenvectors by largest value, using them to create a scree plot of the 11 Major Principal Components. From there we calculated the variance of each principal component[4] across the data to better see how much of the variance each principal component captured. We projected the data using the first five principal components, which captured about 80% of the variance, and proceeded to use our K Means clustering algorithm on the projected data.



Scree Plot

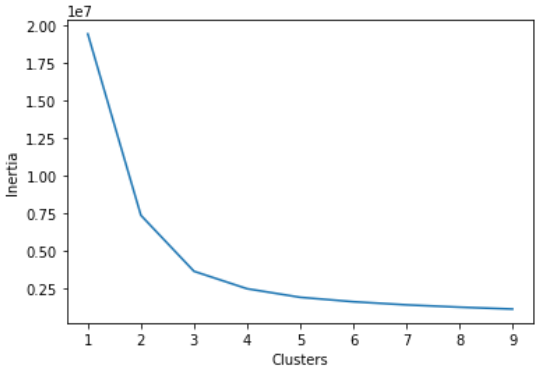


Variance Plot

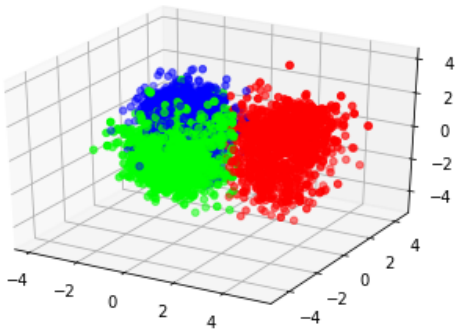
KMeans Clustering

After doing PCA to reduce the dimensionality of our data, we used k-means clustering to pull out useful information. We determined the optimal number of clusters to use with an inertia plot, cutting off the number at the elbow point[5]. We found that three clusters was the optimal K, and plotted the resultant clusters in both 2D and 3D. We also decided to plot every combination of two principal components from our five dimensional reduction to see how the visualization of our data would differ depending on which principal components were chosen. This gives us a rough idea of what each cluster would look like in a 5D space.

[3] https://stats.stackexchange.com/questions/385775/normalizing-vs-scaling-before-pca
[4] https://towardsdatascience.com/principal-component-analysis-pca-from-scratch-in-python-7f3e2a540c51
[5] https://medium.com/analytics-vidhya/choosing-the-best-k-value-for-k-means-clustering-d8b4616f8b86



Inertia Plot



3D Visualization using top 3 principal components

Quality Analysis

Given our k-means cluster assignments, we could attempt to parse out whether the algorithm is finding any human readable patterns in the data. The most likely candidate is quality, given its presence in the dataset as the only qualitative measure. Because clusters are discrete, we needed to partition the quality column into three possible quality categories, defined as mostly equal slices of possible quality values. The quality values are discrete ratings in the range of 4-8, so they were partitioned into high quality (8 and 7), mid quality (6), and low quality (5 and 4). There are six configurations in which the qualities can be arranged. If there is some correlation between the cluster assignments and the quality of each datapoint, one of the configurations should align with the cluster assignments. To check, the mean between the Rnk output of our KMeans and the quality vector, which had the same structure, were compared for each quality configuration, and compared against a randomly generated clustering. The random mean was around 55%, with the highest quality configuration being around 58%. We plotted that quality “clustering” and found no visually distinct clusters.

Discussion

Limitations

Unfortunately, we were unable to reduce our dimensionality to a three or two-dimensional space without losing significantly more variance. Five principal components represented around 80% of the variance of our data, while three principal components represented about 66% of the variance of our data. As a result, our cluster scatterplots are not fully accurate visualizations of the data.

Extensions

We could further expand upon our project by using the alternative method of first clustering our data and then utilizing PCA to visualize our clusters in a lower dimension. Another way we could expand on our project is by testing other forms of dimensionality reduction, such as t-SNE or

PPCA, by using different clustering techniques like the tree based BIRCH clustering or the density based DBSCAN algorithm.

Contributions

Ethan Black:
- Part of PCA analysis
- Quality analysis
- Parts of Data Preparation
- PC Variance visualization

Campbell Moco:
- Parts of Data Preparation
- Parts of KMeans
- Parts of Writeup
- Parts of Slideshow & video

Grant Liu:
- Proposed topic of project and dataset
- Scree and Inertia plot
- Part of PCA analysis
- Parts of video slideshow

Sam Schwartz:
- Part of PCA analysis
- K-means output visualization
- Removal of outliers in original data
- Standardization vs. normalization research

Code

https://github.com/locomoco1313/COGS118BProject2

References

1. Dataset: https://www.kaggle.com/rajyellow46/wine-quality
2. Variance visualization code and similar project: https://andrewmourcos.github.io/blog/2019/06/06/PCA.html
3. Inertia plot: https://medium.com/analytics-vidhya/choosing-the-best-k-value-for-k-means-clustering-d8b4616f8b86
4. Normalization vs. Standardization: https://stats.stackexchange.com/questions/385775/normalizing-vs-scaling-before-pca
5. calcSqDistances, determineRnk, and recalcMus were taken from Homework 3
6. Initial plotCurrent was TA provided code, as well as runKMeans, and eigsort functions