# From Food to Wine

*CS221 Fall 2014: Homework [p-final]*
*SUNet IDs: [jmmeier]*
*Name: [Justin Meier]*

## 1. Introduction

The nuances of wine pairing are not something to be easily overlooked. As quoted by Gerald Asher in his novel *The Pleasures of Wine,* "it needs only a good bottle of wine for a roast chicken to be transformed into a banquet." The idea of having complementary wine and food is something that dates back centuries. Since its development, the pairings have taken on a more strict set of regulations, limiting exactly what wines pair well with certain foods. For example, it has typically been established that red meats pair best with red wines whereas fish pairs best with a white wine. Yet, when taking into consideration a grouping of foods that comprises a dinner, this task may not be so well defined.

Therefore, the overall goal of the following presented models will be to 1) see how universal the rules of food/wine pairing are and 2) analyze whether knowledge of a particular wine/food pairing can be accurately used to predict a wine for an entire meal. The two models presented to analyze these questions are a Naïve Bayes algorithm and a K-Means algorithm, both of which will be explained in detail below.

## 2. Current Research

There is not currently a large amount of research being done surrounding the meshing of artificial intelligence and food/wine pairing. However, while I was at the poster presentation, a man informed me that he and his colleagues were currently working on an algorithm very similar to mine and wanted to extend the application of it, using it as a way to help everyday people pair wine and food and also find the best places to buy said wine. After researching application that paired wine with food, I discovered that there are in fact a large number of companies attempting to make the art of wine pairing more accessible for people who might not be sommeliers. However, the applications gave no indication of any sort of algorithms that they used to find the best wine to pair with a dinner. Therefore, it seems as those there is a need to make food/wine pairing more accessible, but that this idea has not been fully approached from the lens of artificial intelligence and machine learning

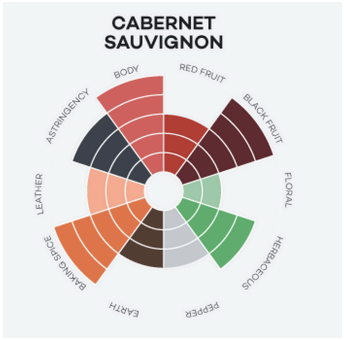
## 2. Datasets

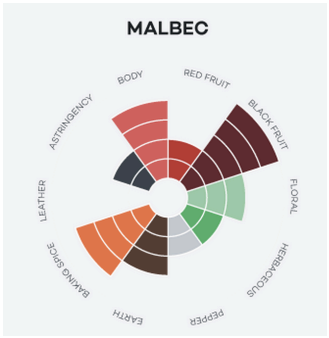
For the models presented, there are a total of three datasets, each pertaining to a separate area of wine classification. By running the models using multiple databases, I hope to demonstrate that each database itself follows a certain set of rules when pairing food and wine. That is, if the algorithm proved to be successful with a high accuracy rating, one could assume that all of the databases are likely were following some general trend surrounding the art of wine pairing.

The first dataset contains pairings of a specific type of food to a specific type of wine, the color of the wine, and a user inputted rating. The website itself was essentially a community forum for which users could input any type of food with any type of wine and then give the pairing a rating from 1 to 5 to demonstrate how well the food and wine matched. A general depiction of the site from which the data came is given below.

| Food | | Type | Varietal | Rating |
|---|---|---|---|---|
| Tomatoes Provencale | Paired with: | Rosé | Côtes de Provence, Rosé AOC | |
| Fried Green Tomatoes | Paired with: | White Wine | Sauvignon Blanc | |
| Fried Green Tomatoes | Paired with: | White Wine | Pinot Grigio | |
| Tomatoes | Paired with: | Red Wine | Barbera DOC | |
| Tomatoes | Paired with: | Red Wine | Chianti DOCG | |
| Tomatoes | Paired with: | Red Wine | Sangiovese | |

The second dataset involves the tastes of the wine. Each wine in the dataset is associated with a "flavor profile" consisting of 10 total flavors, ranging from herbaceous, to astringency, to earthy. The dataset itself contains a total of 32 wines, 16 of which were red and 16 of which were white. A general depiction of the site from which the data came is given below:




The third dataset was by far, the most difficult database to change into a usable format, and only after numerous emails to the company associated with the website was I able to obtain the data. The dataset itself acts as the data upon which I test my algorithms; the data contains a specific wine mapped to a dinner that is composed of a list of ingredients. I use the list of ingredients from this dataset as my input and then assess the accuracy of the output based on whether its predicted wine is the wine given by this dataset. A general depiction of the site from which the data came is given below.


In order to put the results presented later into a better context, below are some general facts regarding the amount of data obtained from the three datasets

*Total Pairings = 2521*

*Unique Foods = 477*
*Unique Wines = 131*
*Dinners = 495*
*Flavor Profiles = 32*

# 3. Features & Preprocessing

## *3.1 Raw Input Data*

The raw input features presented in the first dataset are 1) a specific type of food, 2) a specific type of wine, 3) the color of the wine, and 4) a general using rating.

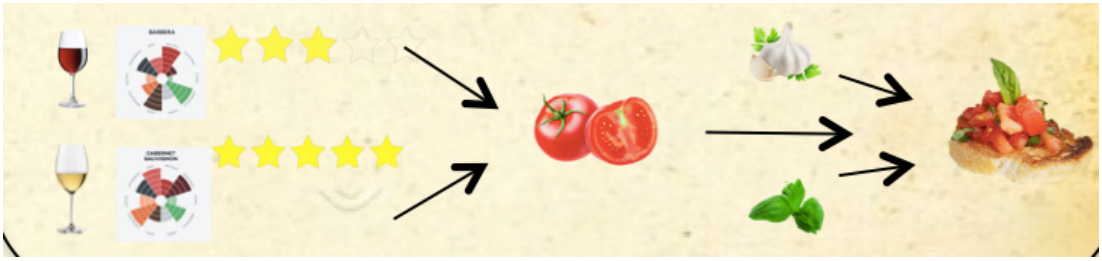
The raw input features associated with the second dataset are simply a specific wine matched to a flavor profile consisting of ten flavors: body, red fruit, black fruit, floral, herbaceous, pepper, earth, baking spice, leather, and astringency. Each of these flavors is associated with an integer value ranging from 0 to 5. Since only 32 wines are available for this dataset, it results in the K-Means algorithm having a narrower selection of wines to choose from, thereby differentiating its results from the results demonstrated by the Naïve Bayes algorithm that has to choose between a total of 131 wines. This will be discussed in a later section.

The raw input features for the third dataset are simply a specific type of wine matched with a list of ingredients. As stated, this dataset is used as the dataset upon which the accuracy of the model are tested.

## *3.2 Derived Features:*

The derived data involves integrating the raw input data presented in all three algorithms, specifically using the flavor profiles as a way to increasing the number of variables associated with each food/wine pairing and each dinner/wine pairing. The first step of this process involves calculating an average wine flavor profile for each specific food. This involves combining the information from the first and second dataset. The flavor profile for each food is obtained by 1) taking the wine flavor profile for each wine paired with the food, 2) multiplying the flavor profile by the user given rating, 3) adding together all the flavor profiles, and 4) dividing each flavor in the flavor profile by the sum of the ratings for the food. The process of multiplying each flavor profile by the rating allows pairings with a higher rating to have a more significant impact on the average flavor profile for the food.

A similar process was used to obtain the average flavor profile for every dinner. The average wine flavor profile for the dinner was found by averaging together all of the wine flavor profiles for each ingredient found within the dinner. And illustration and some equations have been provided below to aide in understanding how this transformation of data occurred.



$$FlavProf(food) = \frac{1}{sumOfRatings} \sum flavorProf(wine) * rating$$

$$FlavProf(dinner) = \frac{1}{numberOfIngredients} \sum flavorProf(food)$$

# 4. Models & Results

The two models used in this project are a Naïve Bayes algorithm and a K-Means algorithm. Each model is analyzed based on its accuracy within two scenarios: 1) predicting the correct color of wine to match with a dinner and 2) predicting the correct type of wine to match with a dinner.

## *4.1 Naïve Bayes:*

The Naïve Bayes algorithm predicts the probability of obtaining a given wine based on a general Naïve Bayes algorithm with multiple given features. The argument that maximizes the equation (whether it be red/white or a specific type of wine) is the top prediction for the color or type of wine.

$$P(x|f_1 \dots f_n) = \frac{P(x)\prod_{i=1}^{n} P(f_i|x)}{P(f_1 \dots f_n)}$$

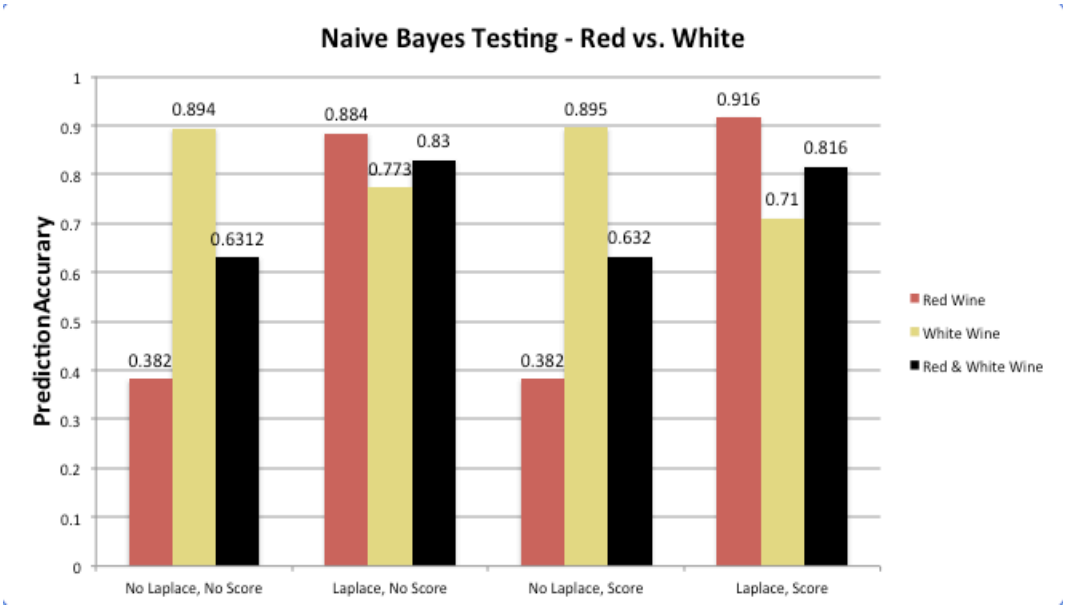
$$\arg\max P(x|f_1 \dots f_n) = \arg\max_x P(x)\prod_{i=1}^{n} P(f_i|x)$$

### *4.1.1 Red vs. White Prediction:*

For predicting the color of the wine, the algorithm uses a total of four distinct versions of the model, involving the inclusion or exclusion of Laplace Smoothing and something termed rank consideration. Let us first consider the implications of the latter.

When rank consideration is used in the model, the probabilities demonstrated in the Naïve Bayes algorithm were calculated based on the rating of the pairing rather than simply the occurrence of a wine and a food together. That is, wine and food pairings with higher ratings would receive a higher probability of occurring together. Thereby, wines with higher pairings for a given food would be the more likely predicted wine for said food. When rank consideration is not used, the probabilities for the Naïve Bayes algorithm are calculated simply based on the occurrence of a wine and a food together in a pairing.

Now let us examine how Laplace smoothing works. When there is no rank consideration, Laplace smoothing works normally. However, when rank consideration is in place, Laplace smoothing has to be slightly modified. Rather than having our smoothing parameter be 1, as often is the case, the smoothing parameter is recalculated to be the average ranking amongst all wines and foods. This value ends up being close to 3.8. The prediction accuracy of these four variants of the Naïve Bayes algorithm can be seen below.

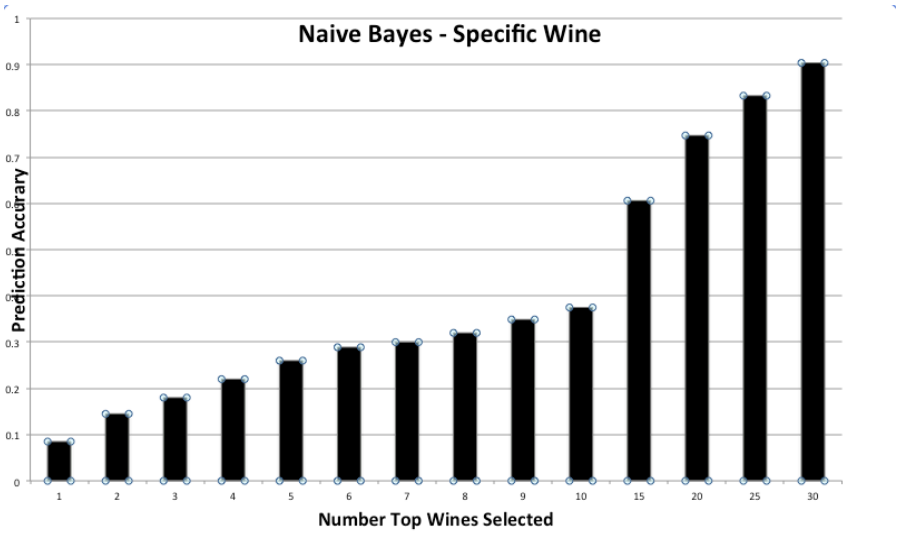



| *Red vs. White* | | | |
|---|---|---|---|
| | Red Pred. Accuracy | White Pred. Accuracy | Both Pred. Accuracy |
| No Laplace No Rank | 38.2% | 89.4% | 63.1% |
| Laplace No Rank | 88.4% | 77.3% | 83.0% |
| No Laplace Rank | 38.2% | 89.5% | 63.2% |
| Laplace Rank | 91.6% | 71.0% | 81.6% |

*4.1.2 Specific Wine Prediction:*

For predicting a specific wine, the Naïve Bayes model uses the variation with the highest accuracy when predicting the color of the wine. Therefore, it uses the Naïve Bayes with Laplace smoothing with no rank consideration.

Because of the large variety of wine choices, the algorithm keeps track of the top N number of predicted wines so as to best demonstrate its accuracy.

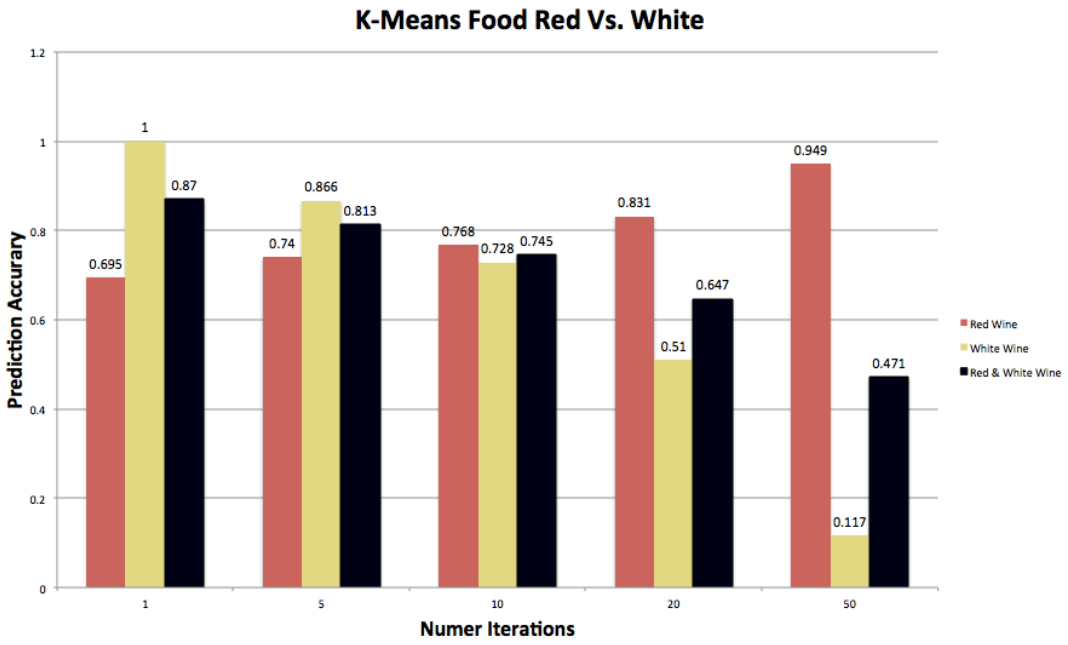


| *Specific Wine (with top N wines)* | |
|---|---|
| N = 1 | 8.5% |
| N = 3 | 18.0% |
| N = 5 | 26.0% |
| N = 10 | 37.5% |
| N = 20 | 74.7% |
| N = 30 | 90.4% |

## *4.2 K-Means*

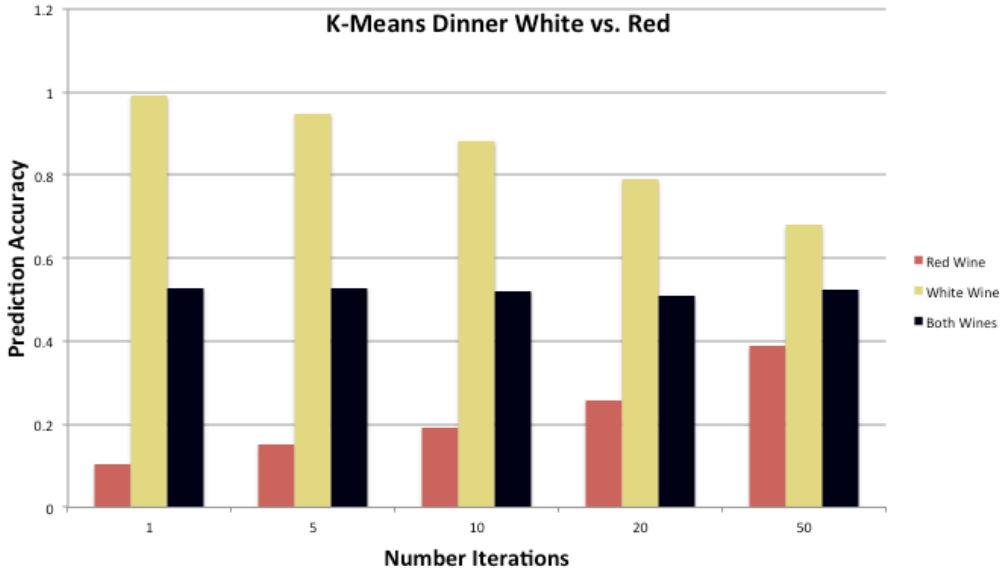
### *4.2.1 Red vs. White Wine Prediction*

For an initial test, I calculate the K-Means algorithm using the flavor profile of a specific food as a point and using the flavor profiles of the original wines in the second database as the centroids. Ultimately, the algorithm is unable to converge, despite numerous attempts to resolve the issue, including variable subset selection. Therefore, the data below plots the accuracy using numbers of iterations.




| **K-Means Using Food** | | | |
|---|---|---|---|
| Red vs. White from Specific Wine for Food | | | |
| | Red Pred. Accuracy | White Pred. Accuracy | Both Pred. Accuracy |
| Iters = 1 | 69.5% | 100% | 87.0% |
| Iters = 5 | 74.0% | 86.6% | 81.3% |
| Iters = 10 | 76.8% | 72.8% | 74.5% |
| Iters = 20 | 83.1% | 51.0% | 64.7% |
| Iters = 50 | 94.9% | 11.7% | 47.1% |

The second algorithm had the points represented as the flavor profiles of the dinner and the centroids as the flavor profiles of the wines in the original database. Once again unable to converge, I calculated the accuracy when predicting a wine for a dinner, using a number of iterations:



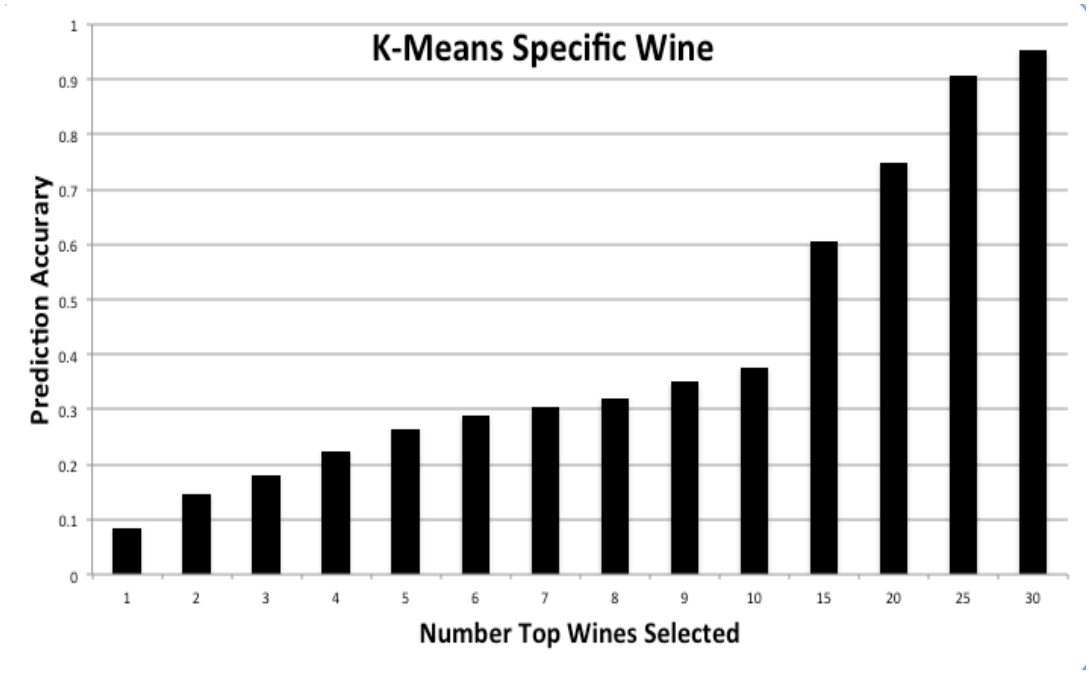

| K-Means Using Dinner | | | |
|---|---|---|---|
| Red vs. White from Specific Wine for Food | | | |
| | Red Pred. Accuracy | White Pred. Accuracy | Both Pred. Accuracy |
| Iters = 1 | 10.3% | 099.1% | 0.529 |
| Iters = 5 | 15.1% | 94.7% | 0.527 |
| Iters = 10 | 19.1% | 88.2% | 0.521 |
| Iters = 20 | 25.9% | 79.0% | 0.511 |
| Iters = 50 | 29.0% | 68.0% | 0.525 |

The third variation involves computing an average flavor profile for a red wine and for a white wine by averaging together all the red wine flavor profiles and all the white wine flavor profiles. Therefore, rather than having 32 centroids, there would only be 2 centroids, one representing the average red wine and the other representing the average white wine.

| Red vs. White Prediction Accuracy with Averaged Red/White Flavor Profiles | | |
|---|---|---|
| Red Pred. Accuracy | White Pred. Accuracy | Both Pred. Accuracy |
| 61.7% | 80.0% | 70.9% |

*4.2.2 Specific Wine Prediction:*

For finding a specific wine, I use the number of iterations that produce the highest accuracy when calculating the color of the wine. This occurred when there is only one iteration. Similar to the Naïve Bayes for predicting a specific wine, the algorithm keeps track of the top N predicted wines to determine accuracy. The top N wines are determined based on the top N centroids that have the smallest Euclidian distance from the point representing the dinner. The results are shown below:




| *Specific Wine (with top N wines)* | |
|---|---|
| N = 1 | 8.4% |
| N = 3 | 18.0% |
| N = 5 | 22.4% |
| N = 10 | 37.6% |
| N = 20 | 74.7% |
| N = 30 | 95.2% |

# 5. Discussion

## *5.1 Naïve Bayes*

Overall, the Naïve Bayes algorithm has a high prediction accuracy for both predicting the color of the wine and also for predicting a specific type of wine. When looking at how the accuracy varies with Laplace smoothing and with rank consideration, one notices that it is highest with Laplace smoothing and it seems that rank consideration has a negligible effect on the accuracy.

Looking at Laplace smoothing, the increase in accuracy makes sense. When analyzing the dataset, it seems that there are quite a few foods that are only paired with a few white wines and have no pairing at all with any type of red wine. However, these foods are often times integrated into dishes were a red wine would be appropriate. Lagrange smoothing fixes this inaccuracy by enabling wines that are not paired with any red wines to still have a possibility of being paired with a red wine (however low it may be).

For predicting a specific wine, the accuracy far surpasses my initial expectations. Given that we have a total of 131 possible wines, the probability of randomly selecting the correct wine is .76%. However, as shown in the graph, when predicting only the top wine, the model predicts the correct one with 8.0% accuracy, 10 fold better than simply guessing. As well, when keeping track of the top 30 wines, the algorithm has 90% accuracy, demonstrating that the model often recognizes the correct wine as a high contender, even if it isn't the top predicted wine.

### *5.2 K-Means*

The K-Means algorithm has less accuracy and also far less consistency, especially when predicting the color of the wine. One of the largest challenges of this project was trying o get the K-Means algorithm to converge. As well, a phenomenon occurred in which as the number of iterations increased, the accuracy of white wine prediction greatly decreased. This effect was reversed for the accuracy of red wine prediction. One theory behind this phenomenon is that there are a few dinners that were extremely weighted towards white wine. That is, some dinners contained only white-wine-associated ingredients such as fish, clam, mussels, lemon, lime, any number of vegetables, and olive oil. Essentially, a large number of seafood dishes fall under this category. However, there are a large amount of of white-wine-paired dishes and nearly all red-wine-paired dishes that contain both ingredients associated with red wine and ingredients associated with white wines. Therefore, at the beginning, the white wine prediction is high. However, as the number of iterations increases, the white wine centroids become more similar to the extreme white-wine-paired dishes. As this occurs, more dishes will be paired with a red wine because they will likely be closer to the red wine centroids as opposed to the white wine centroids, which have shifted to accommodate to the strongest wine-wine paired dishes. When analyzing what dishes are paired with what color wine using a larger number of iterations, this seems to be the case. Almost only seafood dishes with large white-wine-associated ingredients are predicted with a white wine.

For predicting a specific type of wine, the algorithm also does worse than Naïve Bayes. That is, although is has an accuracy of 8.4% when predicting the top predicted wine, it also only has 32 possible wines, as opposed to 131 wines for Naïve Bayes. However, it is still pretty accurate, as guessing the correct wine from 32 possible wines only occurs with 3.2% probability. Although this model has a slightly larger percentage than Naïve Bayes when keeping track of the top 30 wines, this can largely be attributed to this model simply having fewer wines to choose from.

## 6. Conclusion

Based on the results of the two algorithms, the naïve Bayes is more consistent and accurate at predicting the color as well as the type of wine. This could likely be due to the case that Naïve Bayes took a more simplistic and direct approach to problem whereas the K-Means model involved multiple components from each dataset. It is entirely possible that with the number of transformation used to obtain the data used n the K-Means algorithm, the general rules and regulations of wine pairing got lost in the data.

When considering the broad conclusions drawn from the results, it seems that the accuracy of the Naïve Bayes algorithm definitely supports the idea that is a strong and universal set of rules that dominate how wine and food pair together. Because we used separate databases to train and test our algorithm, the accuracy we received is actually quite impressive. Therefore, it would appear that the two databases used for Naïve Bayes had a similar protocol for wine pairing. From this, it also seems that there is a strong link between food/wine pairing and dinner/wine pairing. Namely, this demonstrates that the general rules of wine pairing can apply even when multiple foods are present.

In terms of the conclusions drawn from the K-Means algorithm, it seems that there may be a correlation between the flavor of a wine and how it pairs, but that this correlation is not quite concrete. The algorithm certainly did poorly predicting a red and a white wine, and only did mildly better predicting a specific type of wine. Therefore, it seems that perhaps the flavor of a wine itself may not always correlate to what foods it pairs best with.

# 7. Future

One thing the models do not take into account, that I believe would drastically increase the accuracy of the algorithm, is the percentage different foods might play on determining the wine for a given meal. That is, almost all meats are consistently paired with a red wine. However, many spices are paired with a white wine. When crafting a dish with both meat and spices, however, it may be more beneficial to consider the steak as having more impact on the wine selection than the spices. This could be done by considering the weight of each ingredient. but ultimately this was not something my third database provided.

# References

*What To Pair* [Online]. Available: http://www.whattopair.com

M. Puckette. (2014, Sept. 22). *Flavor Profiles of Wines (Infographic)* [Online] Available: http://winefolly.com/review/red-wine-flavor-profiles/

*Match My Wine* [Online]. Available: http://www.matchmywine.com