CS 373 - Group 4 Project Report

Cade Henschen, Denton Paul, Claudia Duncan

Data Handling

Initial Data

The dataset we analyzed was actually a part of a study conducted at the University of California Irvine - Center for Machine Learning and Intelligent Systems. While their research consisted of ensemble, feature reduction, and precision measurements along with data mining - our study simply generalized the given 11 features into 2 standard machine learning algorithms with 2 cross validation processes for each algorithm to tune hyperparameters. We used the red wine dataset from this study and proceeded to transform the data to match our classification algorithms - as described in the next heading. The initial dataset consisted of:

Input variables (based on physicochemical tests)

dm^3 refers to a cubic decimeter - which is 1 liter

1. **Fixed acidity (tartaric acid - g / dm^3):**
   most acids involved with wine are fixed or nonvolatile (do not evaporate readily)

2. **Volatile acidity (acetic acid - g / dm^3):**
   the amount of acetic acid in wine, which at too high of levels can lead to an unpleasant, vinegar taste

3. **Citric acid (g / dm^3):**
   found in small quantities, citric acid can add 'freshness' and flavor to wines

4. **Residual sugar (g / dm^3):**
   the amount of sugar remaining after fermentation stops, it's rare to find wines with less than 1 gram/liter and wines with greater than 45 grams/liter are considered sweet

5. **Chlorides (sodium chloride - g / dm^3):**
   the amount of salt in the wine

6. **Free sulfur dioxide (mg / dm^3):**
   the free form of SO2 exists in equilibrium between molecular SO2 (as a dissolved gas) and bisulfite ion; it prevents microbial growth and the oxidation of wine

7. **Total sulfur dioxide (mg / dm^3):**
   amount of free and bound forms of S02; in low concentrations, SO2 is mostly undetectable in wine, but at free SO2 concentrations over 50 ppm, SO2 becomes evident in the nose and taste of wine

8. **Density (g / cm^3):**
   the density of water is close to that of water depending on the percent alcohol and sugar content

9. **pH:**
   measure of how acidic/basic water is. The range goes from 0 - 14, with 7 being neutral. pHs of less than 7 indicate acidity, whereas a pH of greater than 7 indicates a base.

10. **Sulphates (potassium sulphate - g / dm^3):**
    a wine additive which can contribute to sulfur dioxide gas (S02) levels, which acts as an antimicrobial and antioxidant

11. **Alcohol (% by volume):**
    the percent alcohol content of the wine

Output variable (based on sensory data)

1. **Quality score:**
   the quality of the wine defined by a score between 0 and 10, with 10 being the highest rating

Data Transformation

In order to make classification problems easier, we converted the 12th label - the quality column - to a binary state. To do so, we converted any value >= 7 to 1 (representing ‘good quality’ wine) and anything below to 0 (to indicate ‘low quality wine’). We choose 7 as an arbitrary number, but does have cultural backing that any wine with a rating higher than 7 in taste is generally recommended where anything below is second rate compared to something higher than 7.

SVM

We attacked this problem with 2 different types of classification separators. SVM for linear separation and KNN for clustering. SVM was tuned with 10-fold and bootstrapping cross validation to mixed results.

Looking at the ten-fold validation, the first few hyperparameters of remarkably low C values resulted in a high variance and rather imprecise classifiers. Moving into the range of 1.0 and 5.0 for our C value, so saw a smaller variance, albeit still not incredibly small. Taking on larger values like 10.0 for C resulted in similar graphs but seemed unnecessary seeing as the margin is attempting to be massive without any real progress being made in classification without overfitting. The best decisions through these hyperparameter decisions were 1 or 5, ending with a range of 77-79% accuracy.

Looking at the bootstrapping validation, lower C values proved again to create a poor classifier, gaining way too much variance at the expense of any real form of bias (even being worse at classifying than chance on a few occasions). Moving into the 1 or 5 values for C show real and more consistent classifiers and while 10 is used, we run into an issue of overfitting as the variance is reduced smaller at the expense of bias becoming more prominent. Especially with the differences in classification being 2% between 1 and 10, banking on the side of a balance between variance and bias allows this algorithm to generalize in a safe and reliable way, clocking in at 82-83% using bootstrapping.

So why the difference between bootstrapping and ten-fold? Well ten-fold splits the data into basically a 90/10 split using a lot more data to train than test. Bootstrapping can actually use less testing data than ten-fold since bootstrapping runs the probability of repeating elements in the testing data, which are ultimately subtracted from the original set to get the training data. Testing data sizes can be fairly small, resulting in 'really good' classifiers because they may only test on a dozen points instead of 160. This isn't always true, but because bootstrapping isn't consistent with its testing and training data set sizes, variability will occur.

So how does SVM hold up against judging wine? Decently well - leaning more towards the percentages generated by the 1 and 5 C values of ten-fold, we see about a 77-80% accuracy from cross validation. While it is not our job to fit a classifier to data, it is our job to judge and test and explore the differences in methods and see which work best, which don't and why that is. SVM works reasonably well, but because it works as a decision boundary and less in clustering, SVM discounts certain aspects like having stronger features that may have been exposed through feature selection like PCA. SVM does help us understand the linear separation present in the data - which seems to be at least somewhat present, meaning several factors are needed to create a 'quality' wine.

<u>Final accuracies:</u>

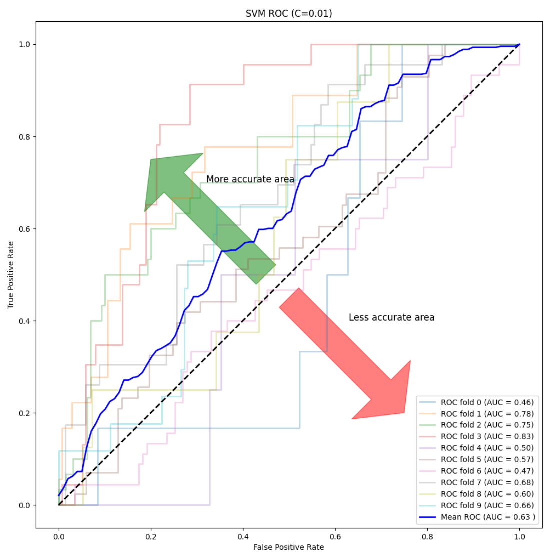
Tenfold:

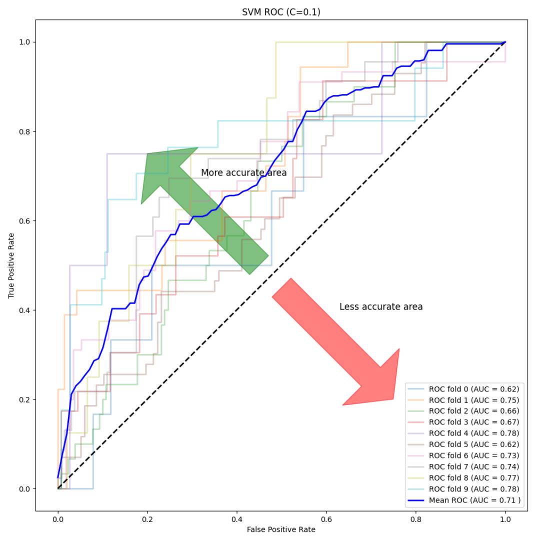
Error: 13.5%	Accuracy: 86.5%

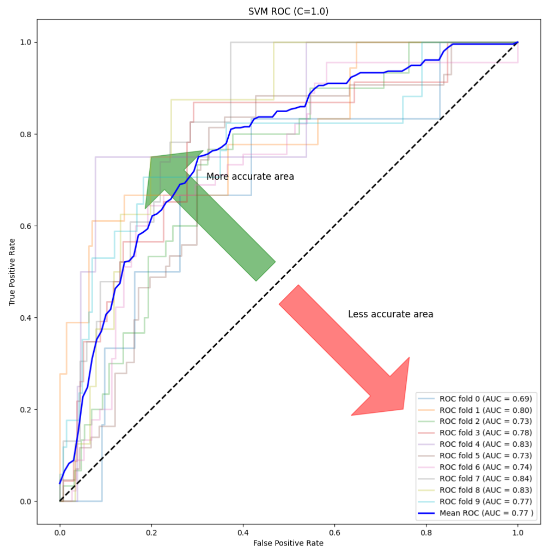
Bootstrapping:

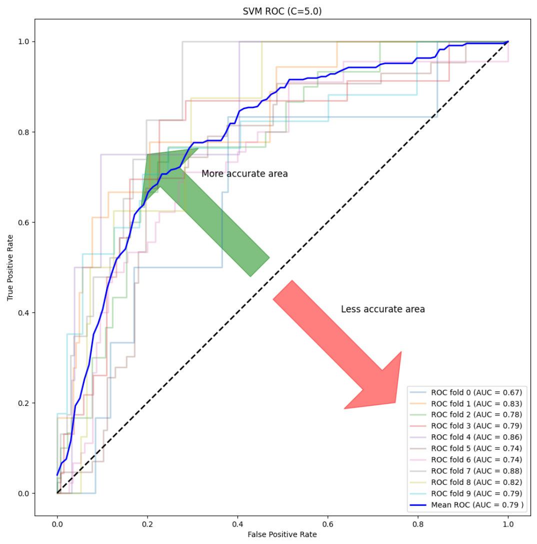
Error: 13.4%	Accuracy: 86.6%

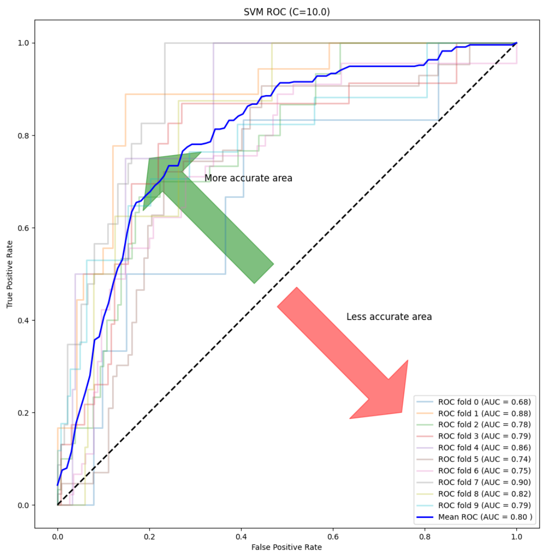
Ten-fold


SVM ROC (C=0.01)
More accurate area
Less accurate area
True Positive Rate
False Positive Rate
ROC fold 0 (AUC = 0.46)
ROC fold 1 (AUC = 0.78)
ROC fold 2 (AUC = 0.75)
ROC fold 3 (AUC = 0.83)
ROC fold 4 (AUC = 0.50)
ROC fold 5 (AUC = 0.57)
ROC fold 6 (AUC = 0.47)
ROC fold 7 (AUC = 0.68)
ROC fold 8 (AUC = 0.60)
ROC fold 9 (AUC = 0.66)
Mean ROC (AUC = 0.63)

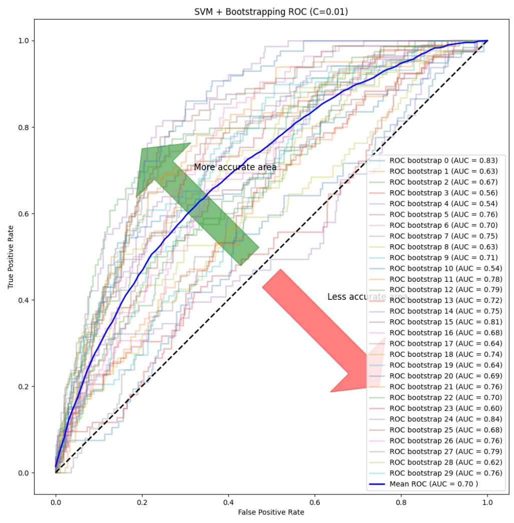

SVM ROC (C=0.1)
More accurate area
Less accurate area
True Positive Rate
False Positive Rate
ROC fold 0 (AUC = 0.62)
ROC fold 1 (AUC = 0.75)
ROC fold 2 (AUC = 0.66)
ROC fold 3 (AUC = 0.67)
ROC fold 4 (AUC = 0.78)
ROC fold 5 (AUC = 0.62)
ROC fold 6 (AUC = 0.73)
ROC fold 7 (AUC = 0.74)
ROC fold 8 (AUC = 0.77)
ROC fold 9 (AUC = 0.78)
Mean ROC (AUC = 0.71)

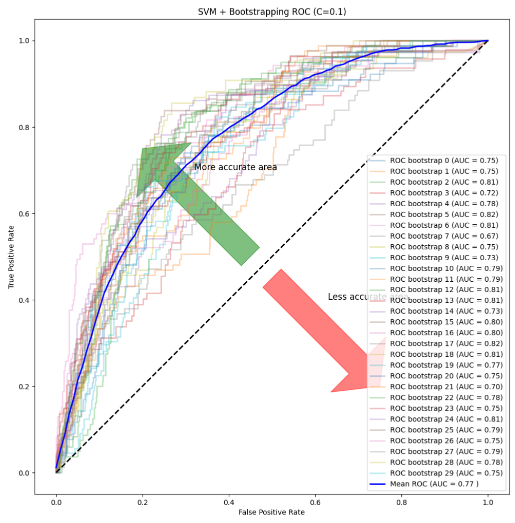

SVM ROC (C=1.0)
More accurate area
Less accurate area
True Positive Rate
False Positive Rate
ROC fold 0 (AUC = 0.69)
ROC fold 1 (AUC = 0.80)
ROC fold 2 (AUC = 0.73)
ROC fold 3 (AUC = 0.78)
ROC fold 4 (AUC = 0.83)
ROC fold 5 (AUC = 0.73)
ROC fold 6 (AUC = 0.74)
ROC fold 7 (AUC = 0.84)
ROC fold 8 (AUC = 0.83)
ROC fold 9 (AUC = 0.77)
Mean ROC (AUC = 0.77)

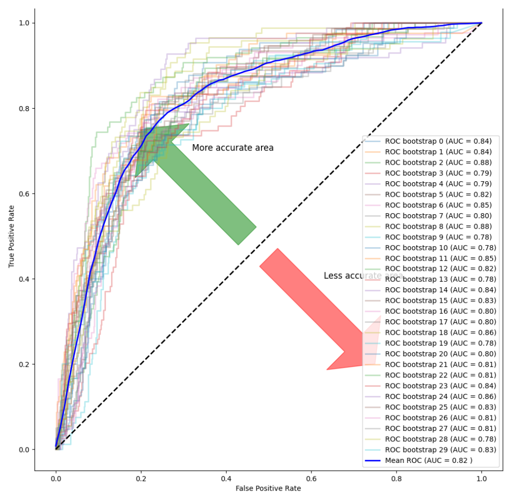

SVM ROC (C=5.0)
More accurate area
Less accurate area
True Positive Rate
False Positive Rate
ROC fold 0 (AUC = 0.67)
ROC fold 1 (AUC = 0.83)
ROC fold 2 (AUC = 0.78)
ROC fold 3 (AUC = 0.79)
ROC fold 4 (AUC = 0.86)
ROC fold 5 (AUC = 0.74)
ROC fold 6 (AUC = 0.74)
ROC fold 7 (AUC = 0.88)
ROC fold 8 (AUC = 0.82)
ROC fold 9 (AUC = 0.79)
Mean ROC (AUC = 0.79)

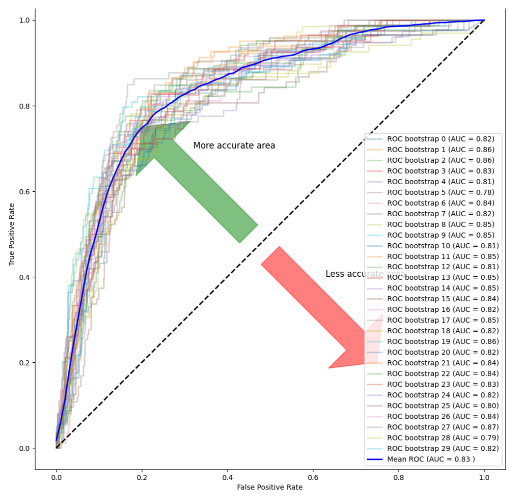

SVM ROC (C=10.0)
More accurate area
Less accurate area
True Positive Rate
False Positive Rate
ROC fold 0 (AUC = 0.68)
ROC fold 1 (AUC = 0.88)
ROC fold 2 (AUC = 0.78)
ROC fold 3 (AUC = 0.79)
ROC fold 4 (AUC = 0.86)
ROC fold 5 (AUC = 0.74)
ROC fold 6 (AUC = 0.75)
ROC fold 7 (AUC = 0.90)
ROC fold 8 (AUC = 0.82)
ROC fold 9 (AUC = 0.79)
Mean ROC (AUC = 0.80)

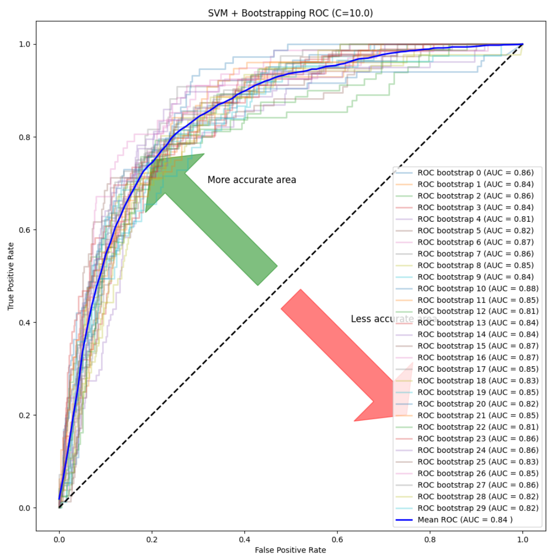
Bootstrapping











KNN

We then used KNN to check if clustering of data occurred. KNN was tuned with 10-fold and bootstrapping cross validation to mixed results - which will be elaborated on later.

Looking at the ten-fold validation, KNN shows pretty consistent variance and bias in each iteration, with incremental bias increase without a whole lot of variance decreasing. While K = 11 was the most “accurate”, outreaching to 11 nearest neighbors in 11D space is arguably a bit of a stretch, even with our 1600 samples. Variance, in fact, seems to grow a little as K increases. It seems like the best choices for KNN are the 5 and 7, which balance the accuracy (69-70%) with a notable amount of variance, while also not overfitting or overgeneralizing the point by clinging to 11 nearby points.

Looking at the bootstrapping validation, a similar phenomenon to SVM occurs, where accuracy ‘gets better’ at the expense of possibly fewer samples to test. Variance and bias seem to remain pretty consistent amongst each K hyperparameter, but the issue of finding 11 nearest neighbors in 11D space is going to become more difficult. A cluster will not be that dense that 11 neighbors can be entirely justified when 3 or 5 or 7 would do the same effect. Choosing a K value needs to both benefit from having good bias but also considering realism and appropriate variance in the process. Settling 3, 5 or 7 are all fair and adequate choices since bias and variance do not differ largely from these hyperparameters.

So how does KNN hold up against judging wine? Strong in certain areas, yet weaker in others. Like mentioned in the SVM section, machine learning is less about getting 100% accuracy, but rather exploring the pros, cons, strengths and weaknesses of each method and judging it against a dataset to gain insight. KNN is best known as a majority clustering classifier, that being it judges the new point not off of a decision boundary, but by the neighbors surrounding it. This allows many “quality” wines to have different ways of being “quality”. For example, a wine might have low sugar, but high in certain acids and sulfates that make a better wine, where maybe low sugar is a disadvantage against another set of attributes like lower pH levels and alcohol percentages. Clustering allows patterns to form and identify those patterns more soundly, although losing decision boundary testing in the process. KNN isn’t as much an algorithm of decision as it is an algorithm of majority, so there is no black and white divider and linear separability is no longer a question being asked unlike SVM.

<u>Final accuracies:</u>

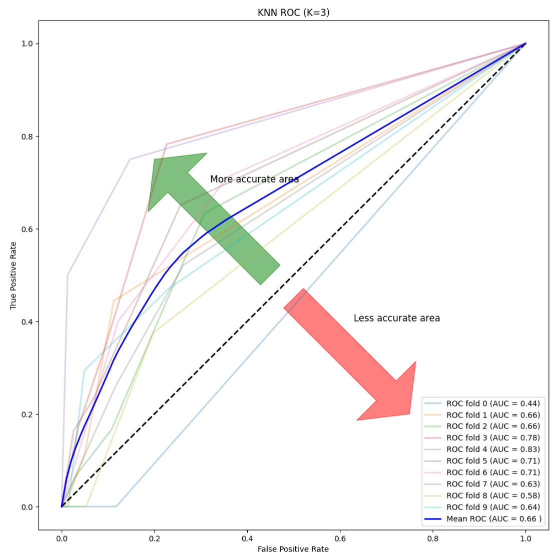
Tenfold:

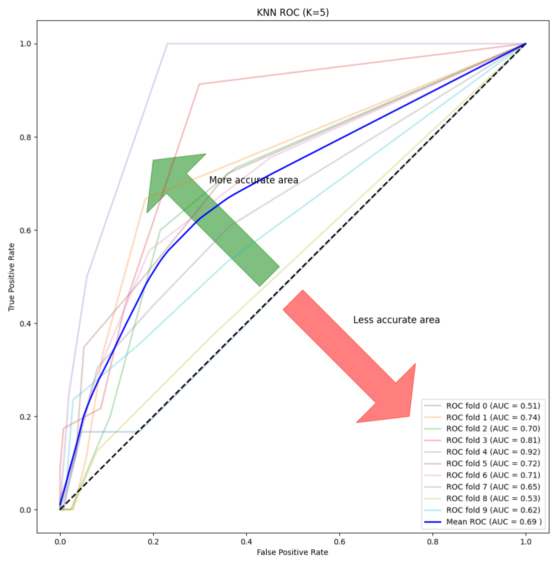
Error: 14.1%	Accuracy: 85.8%

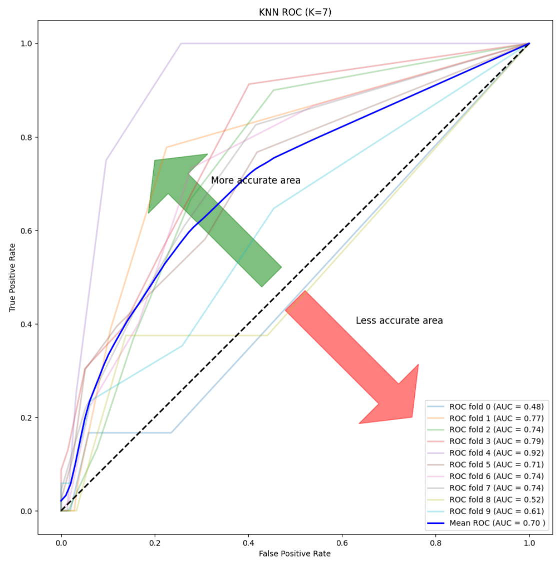
Bootstrapping:

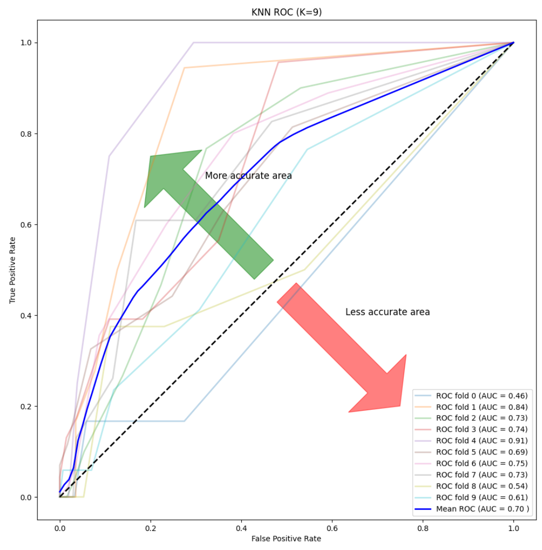
Error: 14.4%	Accuracy: 85.6%

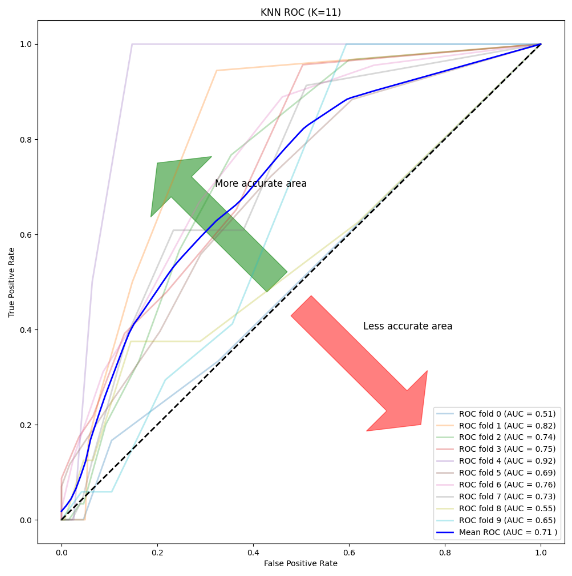
Ten-fold


KNN ROC (K=3)
True Positive Rate
False Positive Rate
More accurate area
Less accurate area
ROC fold 0 (AUC = 0.44)
ROC fold 1 (AUC = 0.66)
ROC fold 2 (AUC = 0.66)
ROC fold 3 (AUC = 0.78)
ROC fold 4 (AUC = 0.83)
ROC fold 5 (AUC = 0.71)
ROC fold 6 (AUC = 0.71)
ROC fold 7 (AUC = 0.63)
ROC fold 8 (AUC = 0.58)
ROC fold 9 (AUC = 0.64)
Mean ROC (AUC = 0.66)

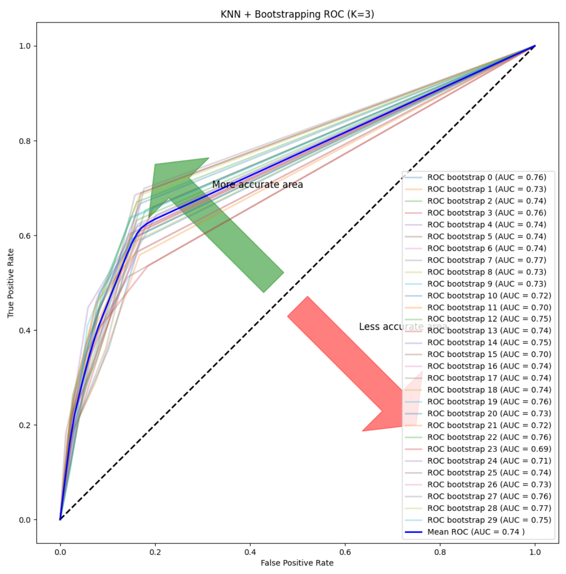

KNN ROC (K=5)
True Positive Rate
False Positive Rate
More accurate area
Less accurate area
ROC fold 0 (AUC = 0.51)
ROC fold 1 (AUC = 0.74)
ROC fold 2 (AUC = 0.70)
ROC fold 3 (AUC = 0.81)
ROC fold 4 (AUC = 0.92)
ROC fold 5 (AUC = 0.72)
ROC fold 6 (AUC = 0.71)
ROC fold 7 (AUC = 0.65)
ROC fold 8 (AUC = 0.53)
ROC fold 9 (AUC = 0.62)
Mean ROC (AUC = 0.69)

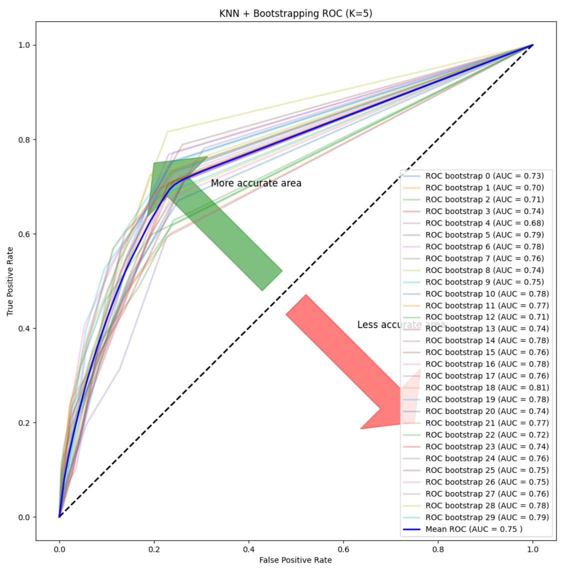

KNN ROC (K=7)
True Positive Rate
False Positive Rate
More accurate area
Less accurate area
ROC fold 0 (AUC = 0.48)
ROC fold 1 (AUC = 0.77)
ROC fold 2 (AUC = 0.74)
ROC fold 3 (AUC = 0.79)
ROC fold 4 (AUC = 0.92)
ROC fold 5 (AUC = 0.71)
ROC fold 6 (AUC = 0.74)
ROC fold 7 (AUC = 0.74)
ROC fold 8 (AUC = 0.52)
ROC fold 9 (AUC = 0.61)
Mean ROC (AUC = 0.70)

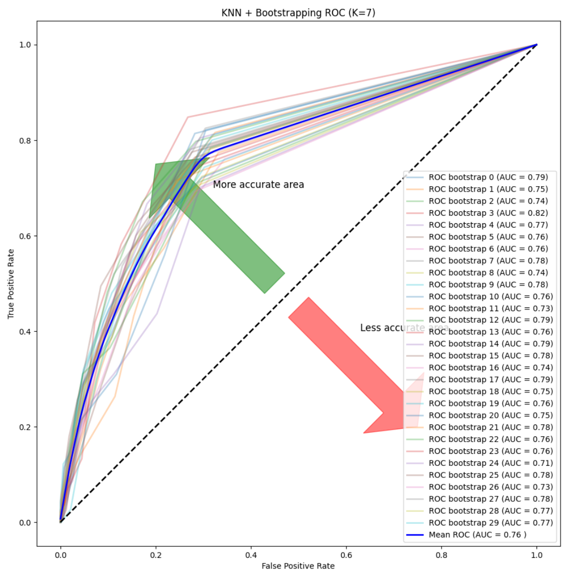

KNN ROC (K=9)
True Positive Rate
False Positive Rate
More accurate area
Less accurate area
ROC fold 0 (AUC = 0.46)
ROC fold 1 (AUC = 0.84)
ROC fold 2 (AUC = 0.73)
ROC fold 3 (AUC = 0.74)
ROC fold 4 (AUC = 0.91)
ROC fold 5 (AUC = 0.69)
ROC fold 6 (AUC = 0.75)
ROC fold 7 (AUC = 0.73)
ROC fold 8 (AUC = 0.54)
ROC fold 9 (AUC = 0.61)
Mean ROC (AUC = 0.70)

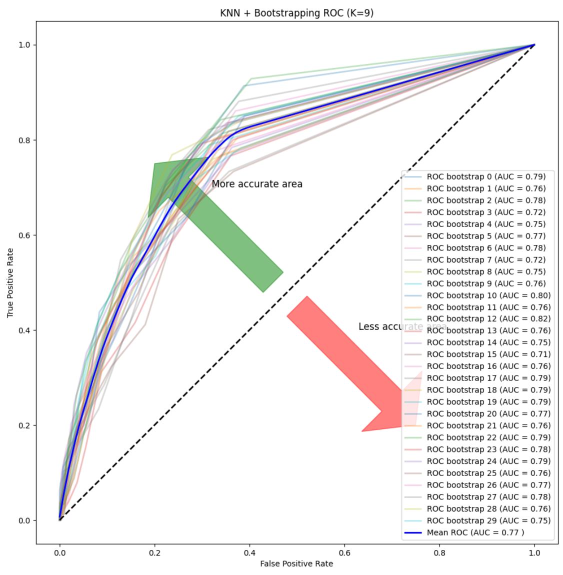

KNN ROC (K=11)
True Positive Rate
False Positive Rate
More accurate area
Less accurate area
ROC fold 0 (AUC = 0.51)
ROC fold 1 (AUC = 0.82)
ROC fold 2 (AUC = 0.74)
ROC fold 3 (AUC = 0.75)
ROC fold 4 (AUC = 0.92)
ROC fold 5 (AUC = 0.69)
ROC fold 6 (AUC = 0.76)
ROC fold 7 (AUC = 0.73)
ROC fold 8 (AUC = 0.55)
ROC fold 9 (AUC = 0.65)
Mean ROC (AUC = 0.71)

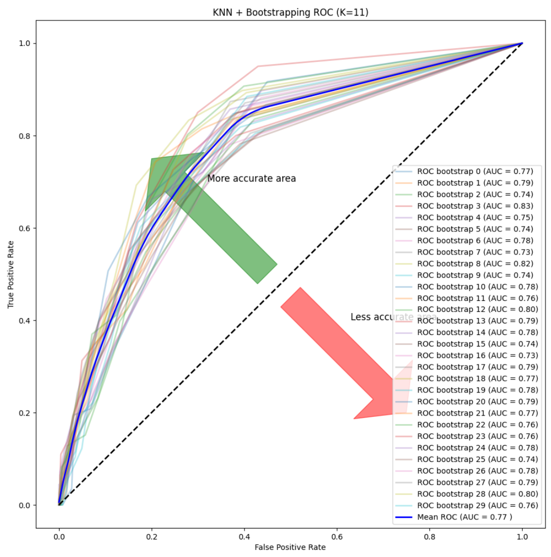
Bootstrapping











Conclusions

Looking at the results of our algorithms, we have to consider why some of them had very low accuracy (63%, barely above random guessing) while others seemed really good (84%) - both are suspicious in isolation. The 63% curve in SVM 10-fold with 0.01 C value is a garbage classifier because it hardly penalizes misses on the single line classifier - a miss penalty of 0.01 will never create a substantial classifier because correction is weak. But you might be asking: “but isn’t the 84% correlated with a high C value SVM classifier in bootstrap? How are they so wildly different?” As commented in the SVM section, bootstrapping doesn’t supply a consistent testing data size, so who’s to say that testing data wasn’t 20 items? This accuracy seems fantastic at face value - but deeper digging and comparison against other C values, algorithms, and validation techniques shows us the dangers of cherry picking what “looks good” over what’s accurate and true to the data.

Collected over all 4 different selections of cross validation and classifiers, on top of the 5 different values for each of those resulted in a ton of different data representation in the ROC graphs and accuracy in subsets. Discussed in the KNN and SVM sections, finding the balance between variance and bias is important - too much variance results in a weak classifier, too much bias results in an overfitting and poorly generalized classifier. Visualizing and assessing are what help strike the balance. We can see a solid balance in the classifiers between the 74-78% range - where KNN doesn’t over extend into 11 neighbors and SVM isn’t course correcting the entire algorithm when one mistake occurs. Additionally we avoid losing the classification strength by avoiding miniscule changes like C values of 0.01. We can settle on SVM values of 1.0 and 5.0 and KNN values of 7 to create accurate, yet generalizing predictors. Looking at the accuracy of subsets, we honestly prefer the training with bootstrap as it not only occurred more (30 iterations), but almost faced random data each time instead of subsets of the dataset individually and in isolation. Random data formed from bootstrapping aids in seeing if generalization against more randomized testing data provides advantageous results.

But anyone can read a graph - what does this mean for judging red wine? What makes it quality? Based on the consistent responses from KNN, we can conclude that while there’s clustering occurring, it doesn’t generalize well to increase accuracy. Increasing K marginally increases accuracy at the expense of increased bias. If we look at SVM, we’ll begin to notice the increase in accuracy as C increases, and by a considerable margin too. Under our naive and simplified experimentations in machine learning with this dataset - it seems like this data has a more ‘linear’ division than a clustering division, that SVM and linear classifiers like regression may have been the better focus area. In another reality of more time and resources, we would have loved to explore feature reduction and used data mining to find the most influential features of red wine to minimize the noise and give more accurate predictors with fewer features. Another interesting exploration of this experiment is to attempt to predict not just if it is quality or not, but how quality a wine may be based on its features. In the end, quality red wines seem to have similarities, as they tend to be grouped together in more of a division than several groups.