# Wine and NFL Predictions with Supervised Learning

Nirave Kadakia

## Overall Description of Classification

There are two interesting and distinct classification problems presented. One is NFL (National Football League) winners and losers, and the other is red wine quality.

The NFL is unique because it presents a classification that is at the heart of many football fans, gamblers, advertisers, and more. With various statistics gathered over the past year, machine learning methods are applied to predict who will win.

The same methods will be applied to red wine to determine red wine quality, an overall subjective score that describes the wine.

## Overall Description of Data

The NFL statistics gathered have 33 different types of data, ranging from whether who is the home team, the Over/Under (the bettor's average of how many total points), to yards gained via passing, rushing, etc. The list is exhaustive and provides a plethora of information. Two seasons of data are gathered, and for each week, the last 16 games (one season) of combined stats is used. This means that statistics are gathered between seasons, which may pose a problem as teams do tend to change year by year. However, the variance of team victories pales in comparison to having a limited sample size in the beginning of the year – thus a 16 game rolling average is used. To obtain this data and thorough description of the attributes: http://www.repole.com/sun4cast/data.html.

Statistics are gathered in a 16 game rolling average, and certain criteria are removed. Average punt yards are removed because it is insignificant, and not every team punts in every game. Team names are also removed as that is not a factor that should be used. In addition, the betting line is removed, as that is often times a reflection of the betting community's winning favorite. Incorporating that statistic would almost be considered "cheating", as the betting line probably uses complex formulas/statistics/algorithms to determine the winner.

Red Wine quality data, and complete description of features, is located https://archive.ics.uci.edu/ml/datasets/Wine+Quality. This contains 11 attributes based on physiochemical tests, ranging acidity, pH level, alcohol level and more. This is used to predict one attribute ranging from 3 to 8 describing the quality.

## Why is the Data Important?

Approximately 15 million will watch a football each week. In fact, the upcoming Super Bowl will be watched by almost half the nation. Determining winners has a real financial impact as well, as winning teams are worth millions more, and gambling on football is a mulit-billion dollar business.

The NFL statistics presents a unique way of training where past games impact the future. This, however, limits the ability to train. For example, when dividing the set into training and validation, the latter games must be used for validation. If validation was from the earlier weeks, then the training set would have stats from the validation set (because of the prior 16 week rolling average), and would be invalid. This also means that cross validation cannot be used.

NFL games will be a simple classification – win or lose. Ties are rare and will be ignored.

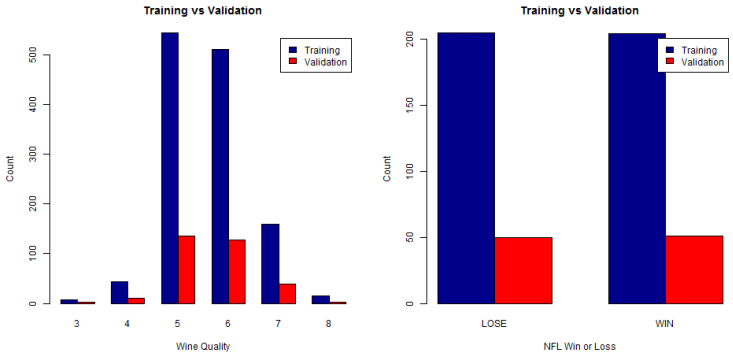
Red Wine quality is important because wine is expensive to procure and choosing the wrong wine can cost a lot of money to both an individual and wine merchants.

## Splitting of Data into Training and Validation

The NFL statistics are split into two. The training set and validation set. The training set is the 2013 first 13 weeks, and the validation are the last 5 weeks. Note that teams have two weeks off, so 16 games plus those 2 bye weeks equals 18 weeks. In addition, cross validation cannot be used, as cross validation will create eventually folds with validation sets containing statistics already represented in the training set's previous weeks.

Red Wine quality will be split, randomly, with 80% training, and 20% validation.

For both datasets, it is important that the training and validation classification be proportionally similar. Here is a graph of the validation and test to visual show that the proportions of wins/losses and wine ratings are equivalent.

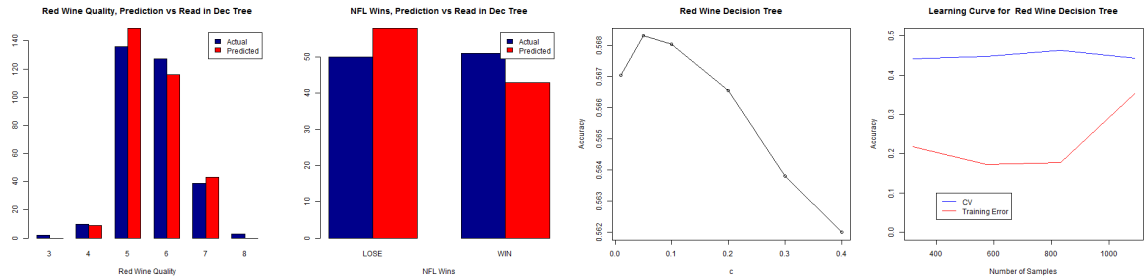



## Decision Tree

The NFL statistics uses a pruned data tree based. This is from R's RWeka library, particularly the function J48, which implements the C4.5 algorithm. This was used because it relies on information gain (important for large attribute sets the NFL dataset) and error pruning.

The decision tree is a good method to use because it is understandable and can eliminate many of the statistics that in the NFL training statistics that may not matter. Often times, the "Site" is a really important determining factor, and having that be on top of the tree means that anyone can instantly understand the ramifications of a team playing at home or away.

Being interpretable is wildly important for both datasets, as this allows the ability to control or learn from various outcomes. For example, with the NFL, the pruned tree states that Sack yards is quite important. While Sack yardage accounts for perhaps 1-2% of total yardage in the game, it says that Sacks are important. So, if a player who is known for sacks is injured, a better or coach can make appropriate adjustments. For wine – knowing the most important tests can help make a better wine. With this decision tree, alcohol content is the most important attribute. Vineyards, for example, can adjust alcohol by extending or reducing the fermentation time and can thus lead to better wine next season.




The first two graphs show the predictions. The predictions for both NFL and red wine predictions are on the low end, but still respectable at 60.37% and 58.99% respectively.

For all hyperparameter findings, cross-validation training is done with various parameters, and the parameter that yields the lowest cross-validation error is the hyperparameter chosen. In this case, cross validation was done using the confidence level hyperparameter. From the third graph above, a confidence level of 0.05 performs quite well, so that was the level used for the wine dataset to get the most accurate predictions.

It is important to note since this method requires cross-validation, only wine, and not NFL predictions, will use this method to find the best hyperparameter.

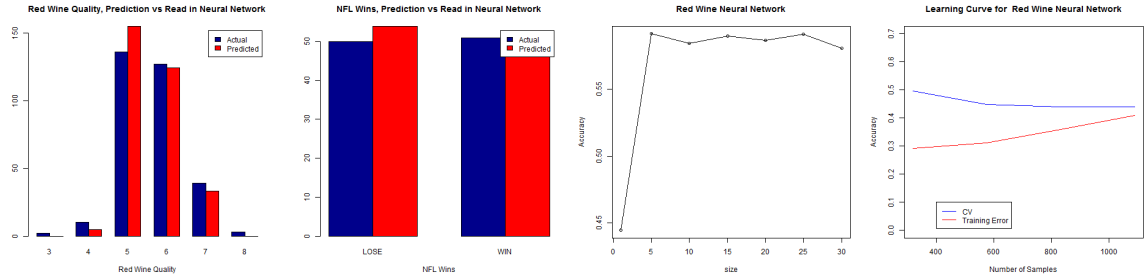
The fourth graph is a learning curve to determine the bias/variance for the Decision Tree for Red Wine (note, NFL cannot use cross-validation, so it is ignored). Though tough to see, it does seem like there could be high variance – even though cross validation error does not go down that much. This means that this learning algorithm could possibly benefit from more training observations.

In terms of size, the Decision Trees for wine consisted of 337 nodes and 169 leaves. The NFL dataset, despite being much larger, was reduced down dramatically to 46 leaves and 89 nodes. This proves that the NFL dataset is quite inundated with "useless" information that is most likely captured in other data (e.g. Passing yards is a component of Total Yards), while the wine data's attributes are quite sufficiently

separated. This makes sense as the wine dataset is a widely-referenced dataset from UCI, as the NFL dataset is an ad-hoc creation with room for improvement.

## Neural Network

Neural Networks is implemented with R's nnet library. Neural Networks provides the overall best accuracy for NFL predictions, but is the second worst for wine quality. This could be that Neural Networks, and the multitude of hidden layers, offers the ability to fit most of the data to the NFL dataset.




The first two graphs show the distribution of predictions. For both sets, the distribution seems fairly normal. However, with the wine score, there is a tendency to bunch up in the middle, leaving the outliers under-predicted.

For overall simplicity, backpropagation was the activation function chosen as it is the most popular. However, the number of hidden units was determined using cross-validation to determine this proper hyperparameter. The third graph shows that the number of number of hidden units for the wine dataset reaches the optimal value of around 5 before leveling out.

The fourth graph shows the Learning Curve. We see that the training error and the cross-validation error have converged, showing high bias, meaning that there is some underfitting going on, meaning that more data would not help. However, if there are more features that could be added, it could help. For example, an easily available feature, price of the wine, could be added and potentially improve accuracy.

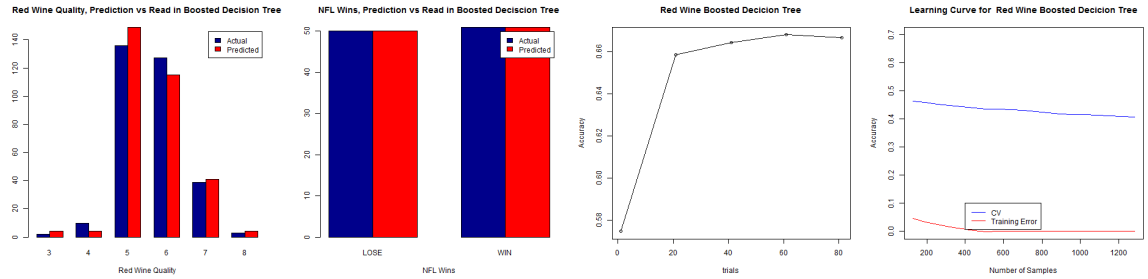
## Boosting

Boosting is done through the C5.0 algorithm via R's C50 library. C50 is the improved version of C45 from the Decision Tree algorithm from above, and includes boosting. Obviously, with boosting, it has improved the accuracy dramatically to 69.4% for the Wine dataset and 66.3%, making it the most accurate prediction method for wine, and the second most for the NFL. However, this comes at a cost of having a higher modeling and prediction time compared to its non-boosted counterpart, Decision Tree.

In terms of size, the NFL trees go through 61 boosting iterations and creates an average tree size of 150.7, thus slightly reducing the tree size compared to Decision Trees. The NFL dataset has an average

tree size of 32.1, meaning there is a drastic reduction in complexity. In other words, boosting and C5.0 has done a lot pruning and simplification for the NFL.

Boosting also boasts all the advantages of interpretability that the Decision Trees have as well. In fact, the important factors at the top of the tree for wine (e.g. alcohol, followed by total.sulphur.dioxide) are the same for both tree. The same top nodes for the NFL dataset are also the same.



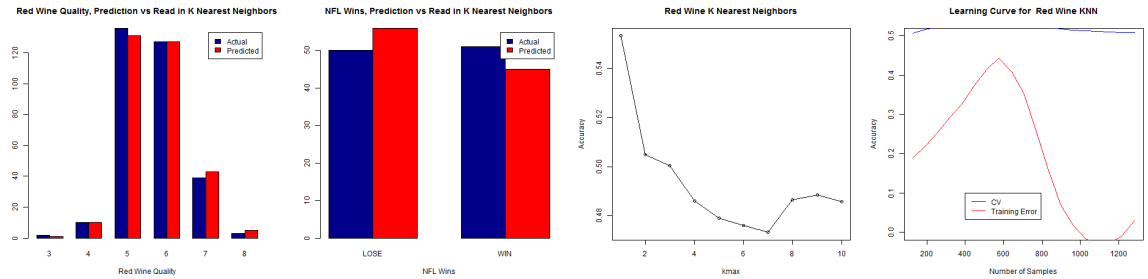

The first two graphs above show that predictions are pretty accurate, and show an even distribution of guesses, again proving it to be the of most accurate learning methods.

For the sake of simplicity, model type is set to a constant "tree", and winnow is set to False. The third graph is used to try the find the best hyperparameter of number of trials. After about 20 trials, the improvement levels out – any additional trials would not yield any significant positive trials. At about 80 trials it peaks.

The fourth graph shows the learning curve and demonstrates the there is a high gap between the cross-validation error and the training error. This is a sign of high variance or overfitting. This would mean that we would need to increase the training set, or reduce the number of features. However, this could prove difficult for wine, as the number of varietals and vineyards is limited around the world.

## K-Nearest Neighbor

K Nearest Neighbor is done via R's class library. K Nearest Neighbor is a lazy learner, meaning it takes little time to predict. However, the training time took a long time, which shows no real cause – this could be due to an inefficient library.




The predictions are somewhat accurate, with the distribution being fairly accurate as well.

The kernel was set to the default, Euclidean distance, so that cross-validation could help pinpoint only one hyperparameter – Kmax. Interestingly, the hyperparameter Kmax (the K in K Nearest Neighbors) that works best when was when it set to 1 for Red Wine dataset. This may mean that there are quite a lot observations in the training set – enough for each new real world observation to match an observation.

This is also reflected in the learning curve. It is tough to absolutely discern the high variance of data, as the cross-validation error is pretty consistent, and the training error does not follow a logarithmic path. However, from the observations in the third graph, and the fact that the peak is reached with just a few samples, the variance is most likely high, and additional observations would cause noise to be modeled – a class case of overfitting. More features would be helpful instead.

### Support Vector Machines

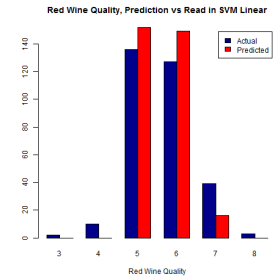
Running Support Vector machines is done through R's library: kernlab. Support Vector Machines are computational expensive, but with both datasets being small, and the power of computing quite powerful today, this was not a significant hurdle. The cost of running non-cross validated functions was not much higher than a Decision Tree.

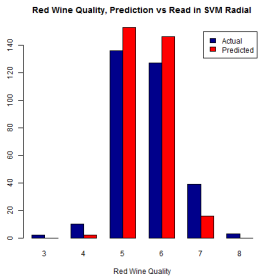
Unfortunately, especially with the NFL dataset, the amount of attributes leads to a black box solution, meaning that the results are not very interpretable. Fortunately, the accuracy was slightly higher than Decision Trees and KNN for both the NFL and the wine dataset, and the modeling time was lower than Boost. Thus, SVM is a middle of the pack learning method for both datasets.

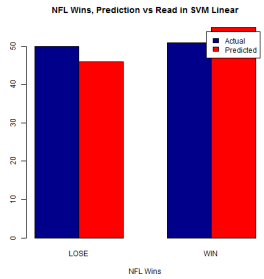
There were two kernels chosen – linear and radial. It is not certain that either dataset is linearly separable. Though with a large amount of attributes in the NFL dataset, it most likely is given the large amount of dimensions. This may be one reason why the accuracy in the linear case is higher than the radial kernel, despite the fact that the radial kernel has a much lower training error. The win e

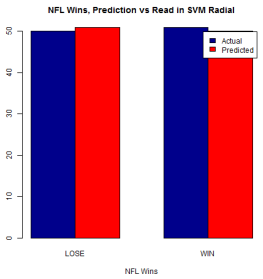
For the wine dataset, cross-validation was used with the built-in kernlab parameter. As explained earlier, the NFL dataset uses no cross-validation.

| | Wine - Linear Kernel | Wine - Radial Kernel | NFL – Linear Kernel | NFL – Radial Kernel |
|---|---|---|---|---|
| Training error | 32.22% | 31.51% | 23.23% | 12.22% |
| Accuracy against validation | 64.04% | 64.36% | 64.36 | 61.39% |

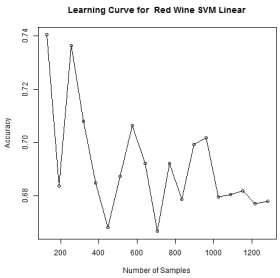


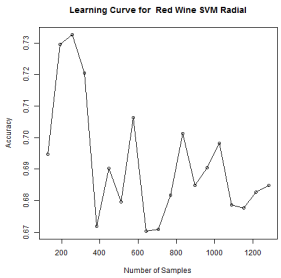



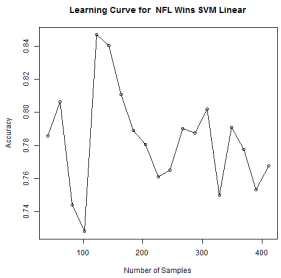



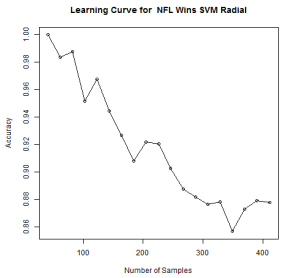




Accuracy is average, and the distributions seem pretty accurate. One important note is that the SVM do tend to cluster wine categories in the middle. This may be to the scores on the edges (3, 4, 7, 8) being outliers that are not reflected in the SVMs.



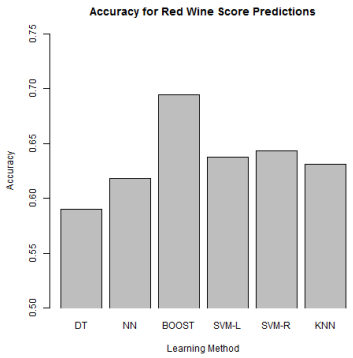



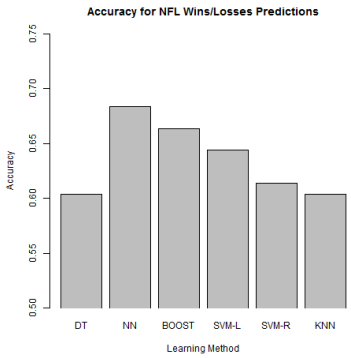




Above are the chart of accuracy, against training sets, against the number of samples. These curves show that it only requires very few observations to obtain good accuracy. This is most likely due to the fact that just a few attributes are needed to get great results. For example, with the NFL, the Site is usually one of the most important. Such things as Passing Yards may actually cloud the subject. On the surface, the more Passing Yards, the better. But in many cases, teams tend to rack up Passing Yards trying to catch up, as they are losing badly. So, this type of statistics may be made more prominent with SVMs.

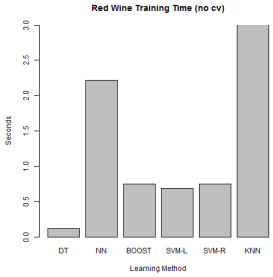
So, this makes SVMs are a good marker for limited data. For example, the red wines may have many observations, but more obscure wines like mead (honey wine) would not have as many, as SVM would make the most sense. For the NFL, the SVM would be ideal for early season betting, where there are only a few weeks of games available to learn from.

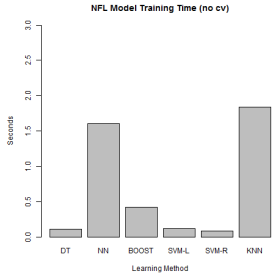
## Overall Results

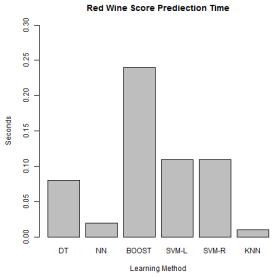


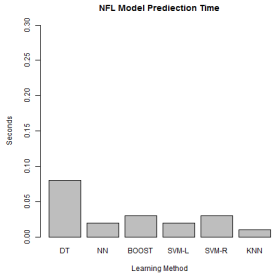




Accuracy for all machine learning algorithms is pretty consistent, with Boosting being the clear winner in the red wine dataset, and Neural Networks being the best in the NFL dataset, and non-boosted Decision Trees being the worst in both cases:



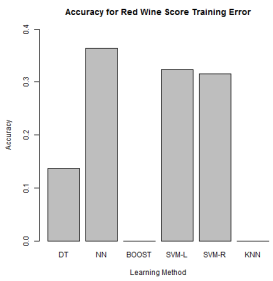



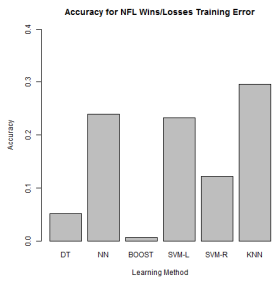




Training time is pretty small overall, and for both datasets, pose no issue. It is worth noting that the modeling times were used with non-cross validated training, to reduce the difference between the NFL (which does not use CV) and the Wine Dataset (which can use CV). The first two graphs above show the modeling time for each learning method. KNN is the slowest in both instances, but it is a bit puzzling since it is a lazy learner. This could be due to the fact that a different, more inefficient library was used. However, all the rest behave as expected, with Neural Networks being the most time consuming (primarily because of the numerous hidden layers), along with boosting due to the amount of trials.

The second graph shows the prediction time for the entire validation set. As shown, prediction time is really quick, thus not posing a problem for these datasets, as predicting NFL winners or opening a wine bottle is not very time sensitive. KNN is the quickest, and that could primarily be because of overfitting and relying on a K of 1 – thus just a quick look up. Decision Trees and Boost are on the higher size, which makes sense as predictions have to traverse a tree.



Finally, here are the graphs of the training errors. It is worth noting that the training errors for Decision Trees are really low in both cases, but the accuracy is not high – this is probably due to overfitting. Boost, has the lowest training error, and has some of the highest accuracy. This shows that Boost is working as advertised.

## Improvements

Improvements can be made to both datasets to improve performance. The NFL dataset is large, and the features are not independent. For example, total offensive yards is the addition of passing yards and rushing yards – so that particular feature could be removed. In addition, defensive statistics are often times the reverse of the offensive statistics – meaning that data is often duplicated in some way or another.

The reason why it is important to remove unnecessary features is to avoid the Curse of Dimensionality, where large numbers of features reduces the predictive power.

Often times, the variance is high for many of these learning methods – meaning that, for methods like boosting, extra observations in the training data would be helpful. This can be easy with the NFL, as more seasons can be looked at. For wine, this could be including white wines.

## Conclusion

All learning methods seem to work well, meaning that it is really the quality of the dataset that matters. While there are some trade-offs, perhaps the best learning method in terms of accuracy, prediction time, and training time would be Boosted Decision Trees.

## References:

Learning curve: http://stackoverflow.com/questions/20370827/plot-learning-curves-with-caret-package-and-r/40885119#40885119

Ng, Andrew, 10.6-AdviceForApplyingMachineLearning-LearningCurves- Machine Learning: https://www.youtube.com/watch?v=g4XluwGYPaA

Caret, Train model by Tags, https://topepo.github.io/caret/train-models-by-tag.html

Wikipedia, Support Vector Machines, https://en.wikipedia.org/wiki/Support_vector_machine

Wikipedia, Bias-variance tradeoff, https://en.wikipedia.org/wiki/Bias%E2%80%93variance_tradeoff