Exploring Various Techniques in Surprised Learning

Kam Wing Chung (Kenneth)

Department of Computer Science

Georgia Tech University

Kchung42@gatech.edu

# Purpose

The purpose of this assignment is to explore supervised learning techniques using decision trees, neural networks, boosting, support vector machines and k-nearest neighbor algorithms. All of above supervising learning techniques will be applied on Wine Quality dataset found on (<http://archive.ics.uci.edu/ml/datasets/Wine+Quality>). All of above supervising learning techniques will be using R and R supervising learning packages. R code, dataset and README.txt can be found in the same package as this document.

# Wine Quality Dataset and Goal

Original dataset contains white and red wine dataset in separate files. WineColor column is added to each dataset and Red is assigned to red wine and White is assigned to white wine new WineColor column.

The two individual datasets are combined together to form one dataset.

Input variables (based on physiochemical tests and in real values)

1 - Fixed acidity (Numeric)

2 - Volatile Acidity (Numeric)

3 – Citric Acid (Numeric)

4 – Residual sugar (Numeric)

5 – Chlorides (Numeric)

6 – Free sulfur Dioxide (Numeric)

7 – Density (Numeric)

9 – pH (Numeric)

10 – Sulphates (Numeric)

11 – Alcohol (Numeric)

12 – WineColor(Numeric: 0=red,1=white)

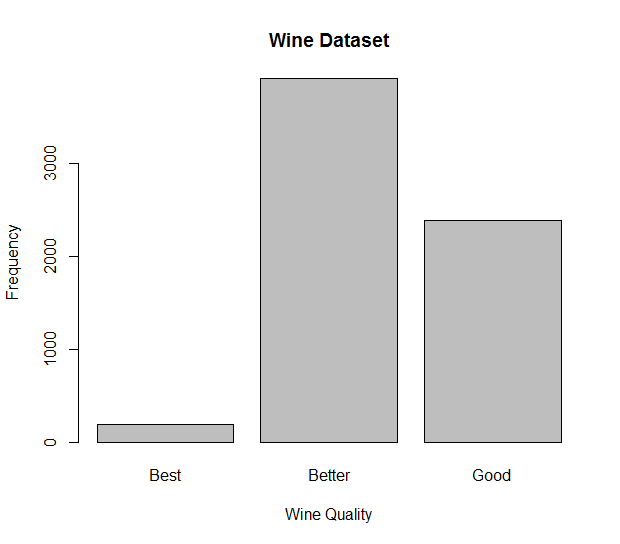
Output variables (based on sensory data)

13 – Quality (1 to 9)

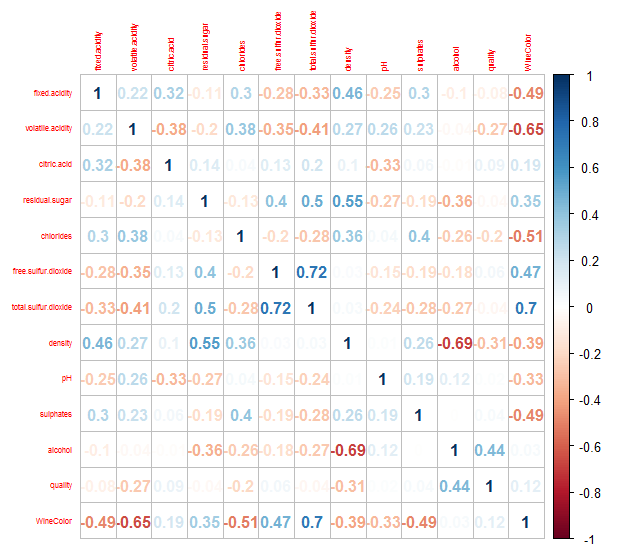
14– Quality3 (Good, Better and Best)

The combined wine quality dataset (WineDataset) will be used in the following supervised learning algorithm to predict wine quality. The original quality real numbers are transformed to Good (<6), Better (6 and 7) and Best (8 and 9).

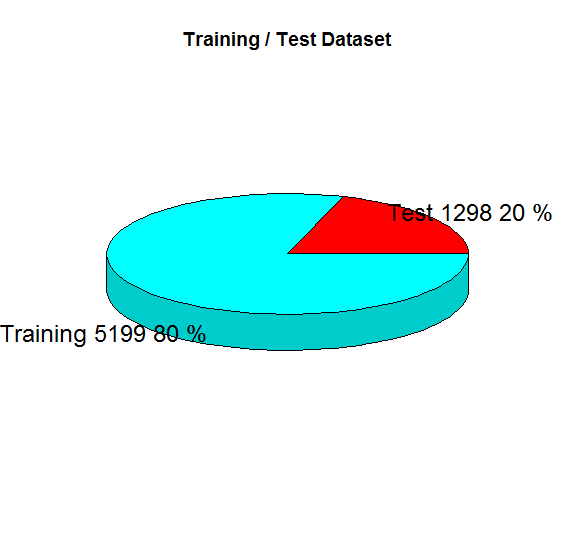
Here is the quality distribution based on three-class classification.



Linear regression suggested that density and WineColor variables which are highly to be a dependent variables predicted by the other variables. Density and WineColor variable will not be considered. The following correlation matrix has given overview about the relationship between independent variables.



The WineDataset is divided into 80% (size=5199) for training and 20% (1298) for testing.



There is a pattern to determine quality purely based on the following the variables or subjective taste preference from a group of wine experts.

There are no missing attributes values in both datasets.

Figure

|  |  |  |
| --- | --- | --- |
|  | Condition Positive | Condition Negative |
| Outcome Positive | True Positive | False Positive |
| Outcome Negative | False Negative | True Negative |

Contingency Table

Figure

The goal on this dataset is to classify the wine quality based on the attributes.

# Decision Tree

Representation of Decision Tree

Top down and greedy algorithm inducting decision tree ID3

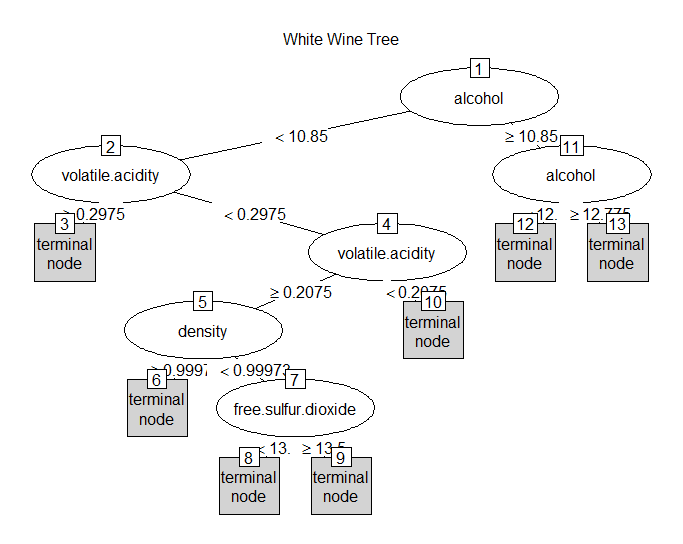
Expressive of Decision Tree

Bias of ID3

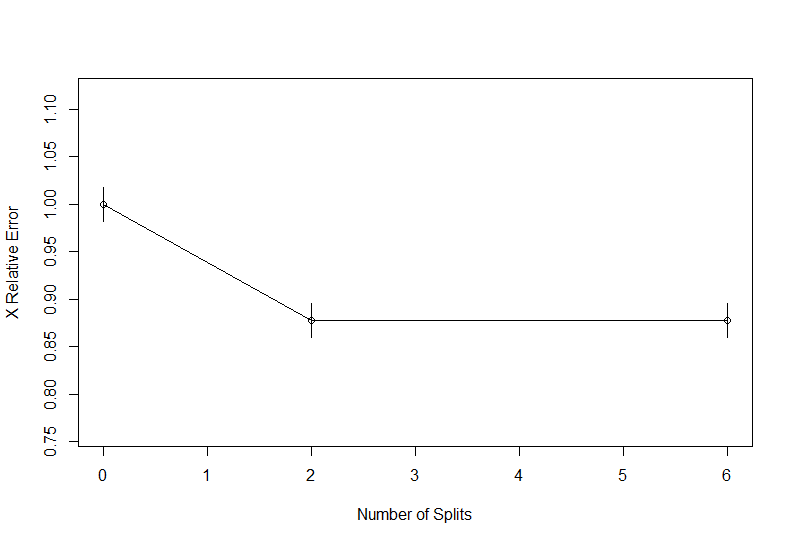
Best Attributes (Gain(S,A)

Deal of with Over fitting

## 3.1 No pruning

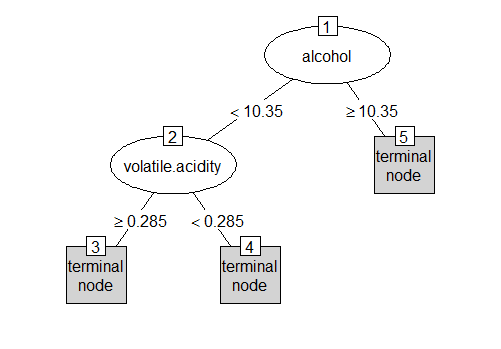


Figure



Figure

## 3.2 Pruned Tree



## 3.3 Decision Tree Conclusion

|  |  |  |
| --- | --- | --- |
|  | No Prune | Pruned |
| Probability correct prediction | 0.518 | 0.506 |
| Classes capable of recognizing | 5 and 6 | 5 and 6 |
| Classes incapable of recognizing | 3,4,7 and 8 | 3,4,7 and 8 |
| Number of Splits | 6 | 2 |
| Variables actually used in tree construction | Alcohol, volatile. Acidity, Density, Free Sulfur dioxide | Alcohol and volatile.acidity |
| Stop at Cross Error | ~0.85 | Slightly lower than ~0.85 |

Table 4

The probability of correct prediction has slight dropped in the pruned tree, essentially it has no significant impact in in both training and test dataset just under 2500 samples.

Both trees are only capable of predicting class 5 and 6; and incapable of predicting class 3, 4, 7, 8 and 9.

In the pruned tree, there are two out of eleven variables that are actually used to construct the decision tree.

Figure

In the figure 6, class 5 wine is populated around low volatile acidity and low alcohol. Class 6 wine is populated around also low volatile and high alcohol. Very small number of class 3 and 8 scattered across. There might be slight greater number of class 4, 7 and 8 that are populated inside the clusters of class 5 and 6.

# Neural Network

Cross Valuation

Linear and polynomial regression

Perceptron – threshold unit

Back propagation/Gradient Descent

Preference/Restriction bias of neural networks

## Configuration and Cross Valuation

Two major parameters for neural network for this implementation is the maximum iteration and k-fold valuation. Iteration is set to 1000 to archive some global minimum and k-fold is set to 10.

Here is the model and cross valuation summary.

|  |  |
| --- | --- |
| Sample Size | 5199 |
| Predictors | 10 |
| Classes | Best, Better, Good |
| Feature Scaling | To [0, 1] |
| Cross-validated | 10-fold |
| Summary of Sample | 4680, 4679, 4679, 4679, 4680, 4679 |

Feature scaling helps to converge faster when the larger range of real values in different variables.

Overall accuracy can be used, but this may be problematic when classes are not balanced.

Kappa statistic takes into account the expected error rate:

Kappa statists = Observed Accuracy – Expected Accuracy / 1 – Expected Accuracy.

Kappa statistics are higher when codes are equiprobable. The effect of bias is greater when Kappa is small than when it is large.

Sensitivity: Given that a result is truly an event, what is the probability that the model will predict an event results.

Sensitivity = # True Positive / # Test outcome positive

Specificity: Given that a result is truly not an event, what is the probability that the mode will predict a negative results?

Specificity = # true Negative / # Test outcome Negative

Unconditional probabilities are positive-predictive values and negative-predictive values,.

Accuracy # True Positive + True Negative / # Population

Data Transformation with range (caret preprocess)

K-Fold Cross-Validation

We leave out the first block of data and fit a model.

This model is used to predict the held=out block

We continue this process until we predict all K held-out blocks

Boostrapping. This procedures also has low variance but non-zero bias when compared to k-fold CV (createResample)

K-fold CV has smaller variance than k-fold CV, it is likely to be biase.

Resample the training samples allows to know when we are making poor choices for the values of these parameters. Resampling method try to inject variations in the system to approximate the model’s performance on future samples.

Resampling will give us honest estimates of future performance.

Repeated training.

Determine K based on highest cross-validated accuracy

Tree can be indexed by their maximum depth and the classical CART methodology uses a cost-complexity (CP) to be determine best tree depth.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| size | decay | Accuracy | decay  (MLP) | Accuracy (MPL) |
| 1 | 0 | 0.659 | Not Avail | 0.712 |
| 1 | 0.0001 | 0.712 | Not Avail |  |
| 1 | 0.1 | 0.712 | Not Avail |  |
| 3 | 0 | 0.679 | Not Avail | 0.721 |
| 3 | 0.0001 | 0.729 | Not Avail |  |
| 3 | 0.1 | 0.723 | Not Avail |  |
| 5 | 0 | 0.673 | Not Avail | 0.726 |
| 5 | 0.0001 | 0.738 | Not Avail |  |
| 5 | 0.1 | 0.731 | Not Avail |  |

The size of hidden activation nodes is 5, and the decay is zero (close to zero) is the final model to be used in the prediction.

## 2.2 Prediction Results and Analysis

The confusion matrix us shown as fellow:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | Prediction Class | | |
|  |  | Best | Better | Good |
| Actual Class | Best | 0 | 38 | 1 |
| Better | 0 | 654 | 129 |
| Good | 0 | 169 | 307 |

The overall statistics is shown as follow:

|  |  |
| --- | --- |
| Accuracy | 0..7404 |
| 95% CI | (0.7156, 0.764) |
| No Information Rate | 0.6633 |
| P-Value [Acc > NIR] | 1.188e-09 |
| Kappa | 0.455 |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Best | Better | Good |
| Sensitivity | NA | 0.6493 | 0.75168 |
| Specificity | 0.96995 | 0.7517 | 0.6832 |
| Pos Pred Value | NA | 0.9527 | 0.23529 |
| Neg Pred Val | NA | 0.2175 | 0.95499 |
| Prevalence | 0 | 0.8852 | 0.11479 |
| Detection Rate | 0 | 0.5747 | 0.08629 |
| Detection Prevalence | 0.03005 | 0.6032 | 0.36672 |
| Balanced Accuracy | NA | 0.7005 | 0.71744 |

|  |  |
| --- | --- |
| Accuracy | 0..7473 |
| 95% CI | (0.7227, 0.7707) |
| No Information Rate | 0.617 |
| P-Value [Acc > NIR] | < 2.2e-16 |
| Kappa | 0.4828 |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Best | Better | Good |
| Sensitivity | NA | 0. 7872 | 0. 6845 |
| Specificity | 0.96995 | 0. 6865 | 0. 8350 |
| Pos Pred Value | NA | 0. 7982 | 0. 8066 |
| Neg Pred Val | NA | 0. 6718 | 0. 3883 |
| Prevalence | 0 | 0. 6117 | 0.11479 |
| Detection Rate | 0 | 0. 4815 | 0. 2658 |
| Detection Prevalence | 0.03005 | 0.6032 | 0.36672 |
| Balanced Accuracy | NA | 0. 7368 | 0. 7598 |

Compare to k-nearest neighbors, there is slightly increase in accuracy but this requires high cost of computation spending on training neural network.

# Support Vector Machine

Maximize Margins

Kernel Tricks

Optimization problem for finding max margins

Support Vectors

Points that are within the margins (or on its boundary) are the support vectors. The prediction function only uses the support vectors.

Tuning the cost and kernel parameters. If the cost parameter is large, there is a significant penalty for having samples within the margin

## Configuration and Cross Valuation

Here is the model and cross valuation summary.

|  |  |
| --- | --- |
| Sample Size | 5199 |
| Predictors | 10 |
| Classes | Best, Better, Good |
| Feature Scaling | To [0, 1] |
| Cross-validated | 10-fold |
| Summary of Sample | 4680, 4679, 4679, 4679, 4680, 4679 |

The configuration and cross validation remains the same as Neural Network. SVM Linear and SVM Radial are examined.

|  |  |  |
| --- | --- | --- |
| C | Accuracy (Radial) | Accuracy (Linear) |
| 0.25 | 0.733 |  |
| 0.5 | 0.739 |  |
| 1 | 0.739 | 0.716 |

For SVM Radial, tuning parameter 'sigma' was held constant at a value of 0.09516869274. Accuracy was used to select the optimal model using the largest value. The final values used for the model were sigma = 0.09516869274 and C = 0.5.

In Linear, there is one value of tuning parameter 'C' which was held constant at a value of 1. The accuracy is 0.716 which is a bit lower than Radial. This reflects the dataset has fairly linear relationship quality and the input variables.

## Prediction Results and Analysis

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | SVM Radial Prediction Class | | |
|  |  | Best | Better | Good |
| Actual Class | Best | 0 | 38 | 1 |
| Better | 0 | 667 | 116 |
| Good | 0 | 159 | 317 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | SVM Linear Prediction Class | | |
|  |  | Best | Better | Good |
| Actual Class | Best | 0 | 38 | 1 |
| Better | 0 | 664 | 119 |
| Good | 0 | 199 | 277 |

The overall statistics is shown as follow:

|  |  |  |
| --- | --- | --- |
|  | SVM Radial | SVM Linear |
| Accuracy | 0.7580077 | 0.7249615 |
| 95% CI | 0.7306636, 0.7781912) | (0.6997931, 0.7491068) |
| No Information Rate | 0.6656394 | 0.6941448 |
| P-Value [Acc > NIR] | 2.527E-13 | 0.008222765 |
|  |  |  |
|  |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| SVM Radial | Best | Better | Good |
| Sensitivity | NA | 0.7679 | 0.7245 |
| Specificity | 0.96995 | 0.7269 | 0.8118 |
| Pos Pred Value | NA | 0.8493 | 0.6576 |
| Neg Pred Val | NA | 0.6097 | 0.8552 |
| Prevalence | 0 | 0.6672 | 0.3328 |
| Detection Rate | 0 | 0.5123 | 0.2411 |
| Detection Prevalence | 0.03005 | 0.6032 | 0.3667 |
| Balanced Accuracy | NA | 0.7474 | 0.7682 |

|  |  |  |  |
| --- | --- | --- | --- |
| SVM Linear | Best | Better | Good |
| Sensitivity | NA | 0.7366 | 0.6915 |
| Specificity | 0.96995 | 0.694 | 0.779 |
| Pos Pred Value | NA | 0.8429 | 0.584 |
| Neg Pred Val | NA | 0.5417 | 0.8491 |
| Prevalence | 0 | 0.6903 | 0.3097 |
| Detection Rate | 0 | 0.5085 | 0.2142 |
| Detection Prevalence | 0.03005 | 0.6032 | 0.3667 |
| Balanced Accuracy | NA | 0.7153 | 0.7353 |

# *k*-nearest neighbors

Instance based learning

Lazy and eager learnings

Similarity function (distance function)

Averaging

Locally weighted regression

## 4.1 Configuration and Cross Validation

**More features you include, more data you need** (The curses functionality)

|  |  |
| --- | --- |
| Sample Size | 5199 |
| Predictors | 10 |
| Classes | Best, Better, Good |
| Feature Scaling | To [0, 1] |
| Cross-validated | 10-fold |
| k-fold selection | 1,2,3,5,10,20,30,50 |

K=1 is selected because of having highest accuracy.

|  |  |
| --- | --- |
| k | Accuracy |
| 1 | 0.754 |
| 2 | 0.701 |
| 3 | 0.715 |
| 5 | 0.722 |
| 10 | 0.718 |
| 20 | 0.713 |
| 30 | 0.711 |
| 50 | 0.72 |

As the k increases, the accuracy remains pretty flat until k reaches 50.

## 4.2 Prediction Results and Analysis

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | Prediction Class (k=1) | | |
|  |  | Best | Better | Good |
| Actual Class | Best | 21 | 17 | 1 |
| Better | 24 | 643 | 116 |
| Good | 1 | 152 | 323 |

|  |  |
| --- | --- |
| Accuracy | 0.7604 |
| 95% CI | (0.7362, 0.7834) |
| No Information Rate | 0.6256 |
| P-Value [Acc > NIR] | <2e-16 |
| Kappa | 0.5182 |

|  |  |  |  |
| --- | --- | --- | --- |
|  | Best | Better | Good |
| Sensitivity | 0.45652 | 0.7919 | 0.7341 |
| Specificity | 0.98562 | 0.7119 | 0.8217 |
| Pos Pred Value | 0.53846 | 0.8212 | 0.6786 |
| Neg Pred Val | 0.98014 | 0.6718 | 0.8577 |
| Prevalence | 0.03544 | 0.6256 | 0.339 |
| Detection Rate | 0.01618 | 0.4954 | 0.2488 |
| Detection Prevalence | 0.03005 | 0.6032 | 0.3667 |
| Balanced Accuracy | 0.72107 | 0.7519 | 0.7779 |

# Boosting

Ensembles are good

Bagging are good

Combine simple classifiers -> Complex

Boosting is really good

Weaker Learners

Boosting avoid over fitting

After the first tree is created, weight are determined and subsequent iterations create weighted trees of about the same size as the first. The final prediction is a simple average