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 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

The two datasets are related to red and white variants of the Portuguese “Vinho Verde” wine. The inputs include object tests and the output is based on sensory data (median of at least three evaluations made by wine experts). Each expert graded the wine quality between 0 (very bad) and 10 (very excellent).

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

1 - Fixed acidity

2 - Volatile Acidity

3 – Citric Acid

4 – Residual sugar

5 – Chlorides

6 – Free sulfur Dioxide

7 – Density

9 – pH

10 – Sulphates

11 – Alcohol

Output variables (based on sensory data)

12 – Quality (score between 0 and 10)

There are no missing attributes values in both datasets.

There are 4898 white wine samples and 1599 red wine samples.

Here is the quality distribution of red wine and white wine in the datasets

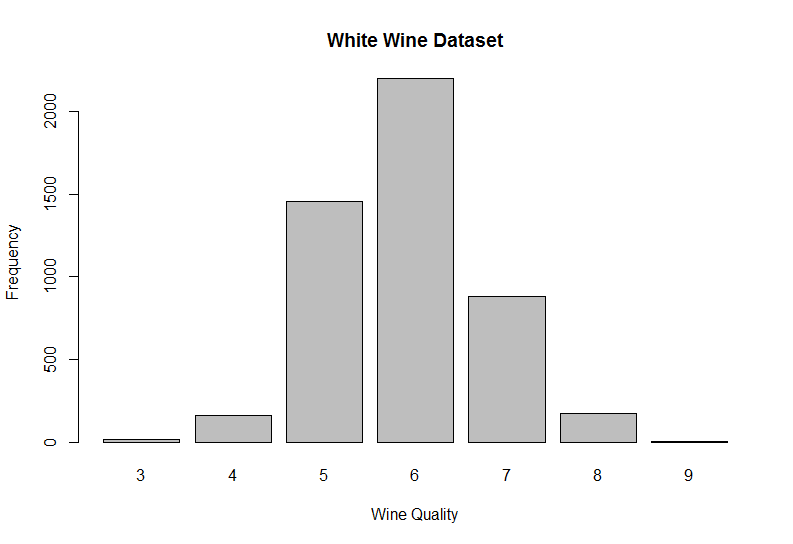


Figure 1

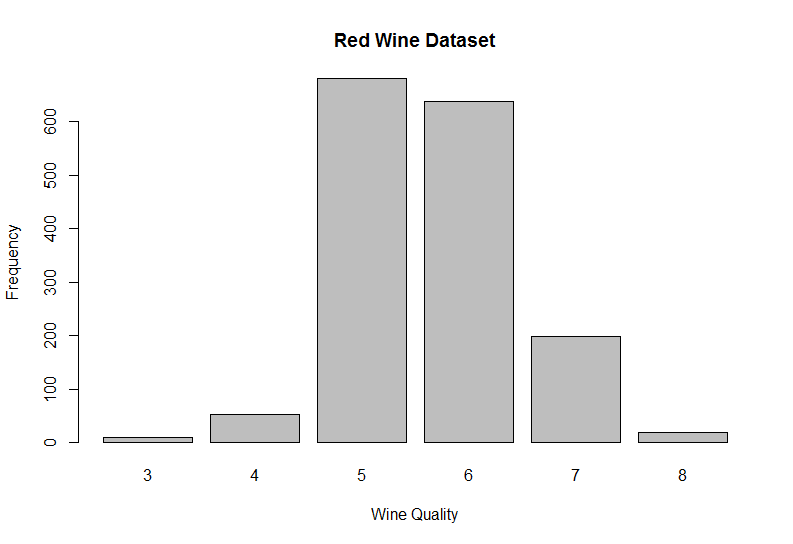


Figure 2

Both white and red wine datasets are divided into half for training and remaining half for testing with random selection from the sample and consistent seed to guarantee same test and training sets for every run.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Quality | White Wine Training | White Wine Test | Red Wine Training | Red Wind Test |
| 1 | 0 | 0 | 0 | 0 |
| 2 | 0 | 0 | 0 | 0 |
| 3 | 10 | 10 | 4 | 4 |
| 4 | 97 | 66 | 21 | 32 |
| 5 | 737 | 720 | 330 | 351 |
| 6 | 1084 | 1114 | 328 | 310 |
| 7 | 430 | 450 | 110 | 89 |
| 8 | 90 | 85 | 6 | 12 |
| 9 | 1 | 4 | 0 | 0 |

Table 1

Table 1 shows the wine quality distribution in four data sets. There are 4898 white wine samples and 1599 red wine samples. Since white has large of size of sample, the following analysis will primary be focusing on white wine dataset.

Since quality attribute values are in numeric and no quality falls between two integers and this is classification problem, the quality attribute values have been converted to characters to fit this goal of this assignment.

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

# Decision Tree

## 3.1 No pruning

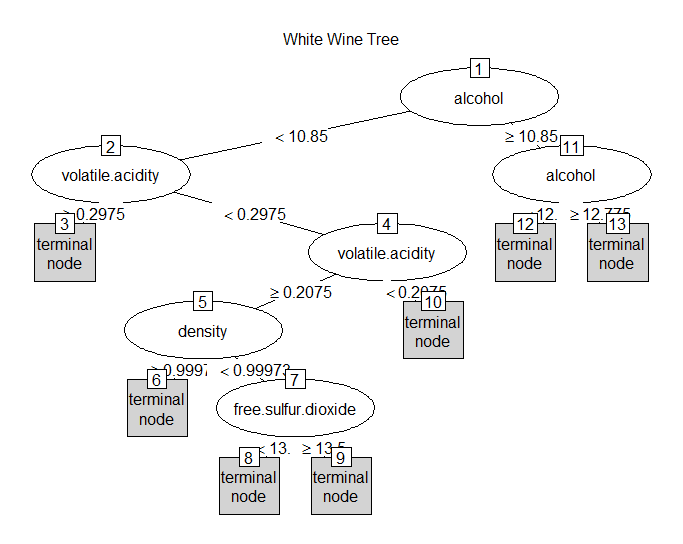


Figure 3

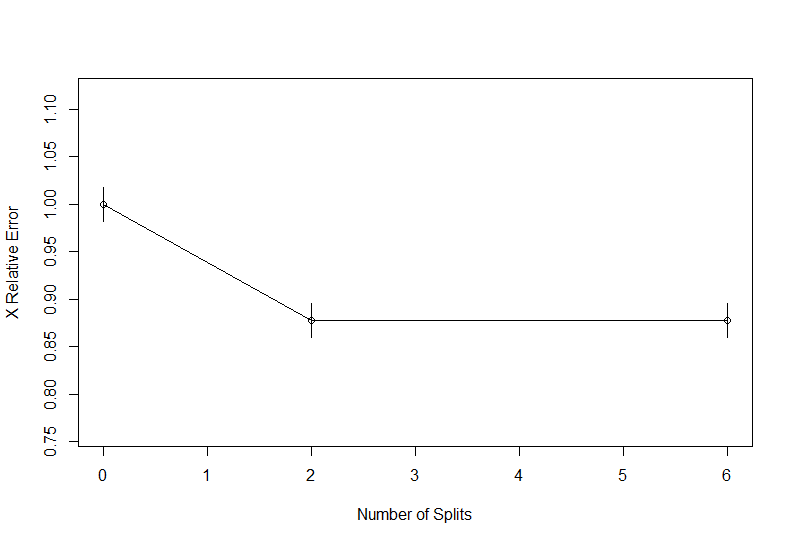


Figure 4

Figure 3 shows that there are six splits on decision tree using the white wine test data. Figure 4 indicates that the starting from split 2, the cross errors remains at just above 0.85.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Predicted Class | | | | | | |
|  |  | 3 | 4 | 5 | *6* | 7 | 8 | 9 |
| Actual Class | 3 | 0 | 0 | 3 | 7 | 0 | 0 | 0 |
| 4 | 0 | 0 | 28 | 38 | 0 | 0 | 0 |
| 5 | 0 | 0 | 377 | 343 | 0 | 0 | 0 |
| 6 | 0 | 0 | 220 | 894 | 0 | 0 | 0 |
| 7 | 0 | 0 | 14 | 436 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 85 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 4 | 0 | 0 | 0 |

Table 2

The probability of predicting correct classes on the white wine test dataset is 0.518. The calculation is based on 1/n where n is the number of test dataset sample and k is the index of prediction and actual classes. If prediction value equals to actual class, returns 1 otherwise 0.

## 3.2 Pruned Tree

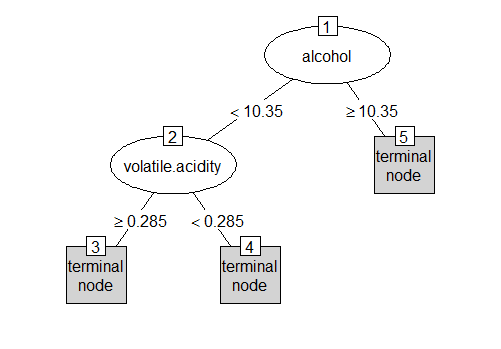


Figure 5

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Predicted Class | | | | | | |
|  |  | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| Actual Class | 3 | 0 | 0 | 2 | 8 | 0 | 0 | 0 |
| 4 | 0 | 0 | 21 | 45 | 0 | 0 | 0 |
| 5 | 0 | 0 | 268 | 452 | 0 | 0 | 0 |
| 6 | 0 | 0 | 142 | 972 | 0 | 0 | 0 |
| 7 | 0 | 0 | 10 | 440 | 0 | 0 | 0 |
| 8 | 0 | 0 | 0 | 85 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 4 | 0 | 0 | 0 |

Table 3

Figure 5 has pruned tree based on 2 splits and cross error indicated on Figure 4. The probability of predicting correct class on white wine test dataset using pruned is 0.506 which is slightly lower than non-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.

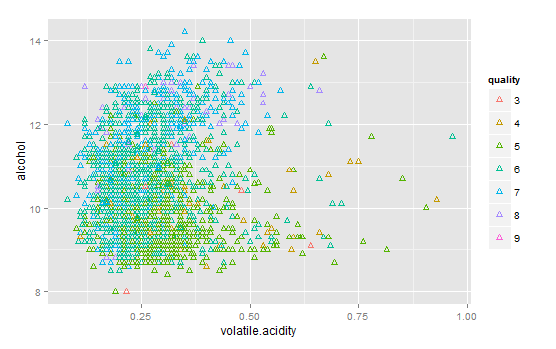


Figure 6

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

Remarks: Due to my computer hardware configuration and performance, the size of training and testing dataset has been scaled down to 1/25 of the size of white wine training and testing data used in above Decision Tree experiment and there are only three attributes (volatile acidity, density and alcohol) being used.

## One Layers and Two Activations

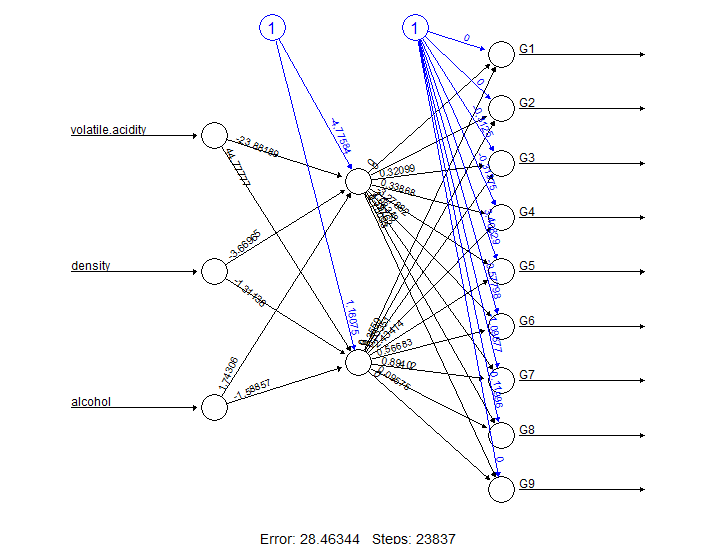


Figure 7

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Prediction Class | | | |
|  |  | 4 | 5 | 6 | 7 |
| Actual Class | 3 | 0 | 1 | 0 | 0 |
| 4 | 0 | 1 | 1 | 0 |
| 5 | 0 | 20 | 11 | 0 |
| 6 | 0 | 3 | 41 | 0 |
| 7 | 1 | 1 | 11 | 0 |
| 8 | 0 | 0 | 4 | 1 |
| 9 | 0 | 0 | 1 | 0 |

Table 5

The probability of predicting correct classes on white wine test dataset is 0.6288 based on about 100 training and test datasets. Even with smaller dataset, the accuracy is higher than Decision Tree that has a lot larger sample size.

|  |  |
| --- | --- |
| error | 28.46343968 |
| reached.threshold | 0.009778691 |
| steps | 23837 |

Table 6

In this single layer and two activation, the training process needed 23873 steps until all absolute partial derivatives of error function were smaller than the default threshold 0.01. Since there is only one layer and two activations, the error remains undesirable for solving real problem. More layers and more activation might improve the accurray.

## 2.2 Two Layers and Three Activations

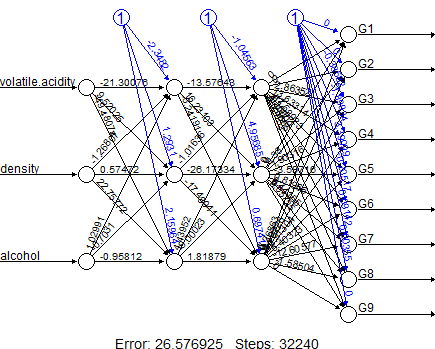


Figure 8

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  | Prediction Class | | |
|  |  | 4 | 5 | 6 |
| Actual Class | 3 | 0 | 1 | 0 |
| 4 | 0 | 1 | 1 |
| 5 | 1 | 19 | 11 |
| 6 | 1 | 3 | 40 |
| 7 | 2 | 1 | 10 |
| 8 | 0 | 1 | 4 |
| 9 | 0 | 0 | 1 |

Table 7

The probability of predicting correct classes on white wine testing dataset is 0.6088 based on about 100 training and test datasets and with two layers and three activations.

|  |  |
| --- | --- |
| error | 26.57693 |
| reached.threshold | 0.009687 |
| steps | 32240 |

Although the reach.threshold has reached lower than default 0.0.1 and the error is 26.57693 slightly lower than single layer error 28.46343968, the results with additional layer and activation did not improve the accuracy. It could be approaching over fitting and reached the lower minimal. But it is too hard to draw conclusion on these two runs with very small dataset and not enough attributes to try various scenarios.

# Support Vector Machine

In the SVM simulation, there are ~2500 white wine samples used for training data and ~2500 white wine samples used for testing data.

This test includes all eleven attributes described in section 2. SVM based on Gaussian Kernel is used to classify the sample into nine different qualities as discrete variables.

Here is the confusion matrix.

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Prediction Class | | | | | | |
|  |  | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| Actual Class | 3 | 0 | 0 | 5 | 5 | 0 | 0 | 0 |
| 4 | 0 | 3 | 37 | 26 | 0 | 0 | 0 |
| 5 | 0 | 4 | 424 | 290 | 2 | 0 | 0 |
| 6 | 0 | 0 | 225 | 828 | 61 | 0 | 0 |
| 7 | 0 | 0 | 13 | 321 | 116 | 0 | 0 |
| 8 | 0 | 0 | 0 | 61 | 24 | 0 | 0 |
| 9 | 0 | 0 | 0 | 1 | 3 | 0 | 0 |

Table 8

Using the same probability calculation as Decision Tree and Neural Network, the probability to predicting correct classes on white wine test data is 0.5598 which falls between Decision Tree and Neural Network.

But the training time is a lot shorter than NN and decision tree. Even with ~2500 sample data for each training and testing, the accuracy remains very close to NN and Decision Tree.

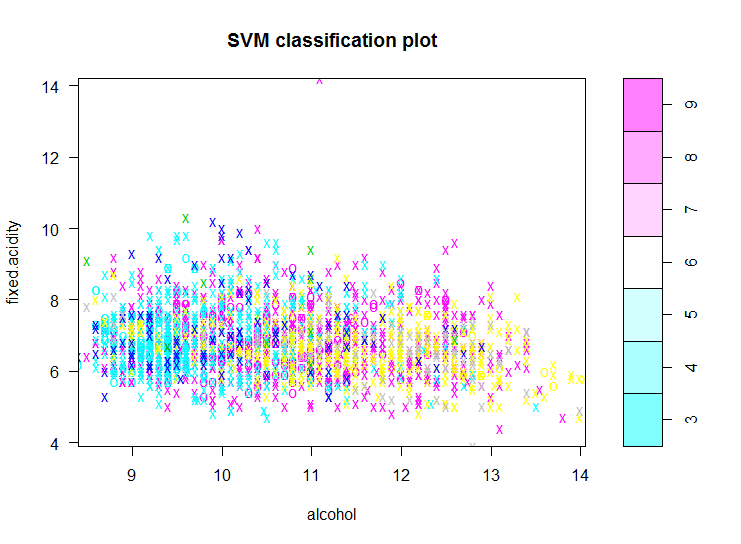


Figure 9

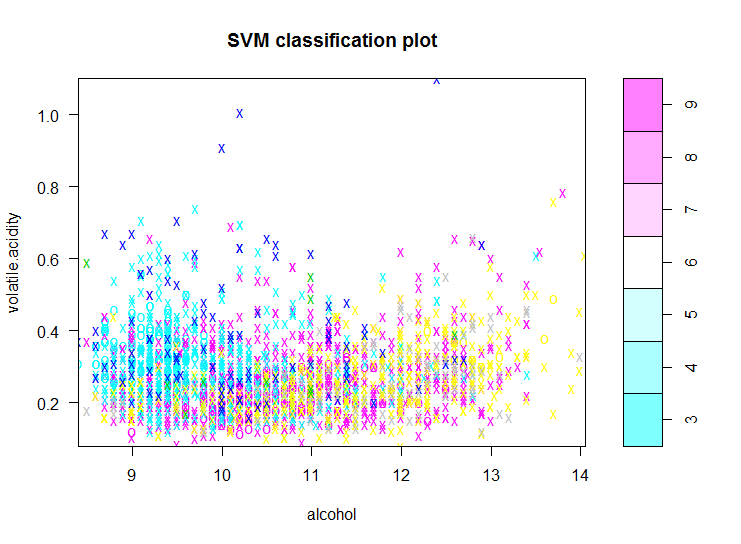


Figure 10

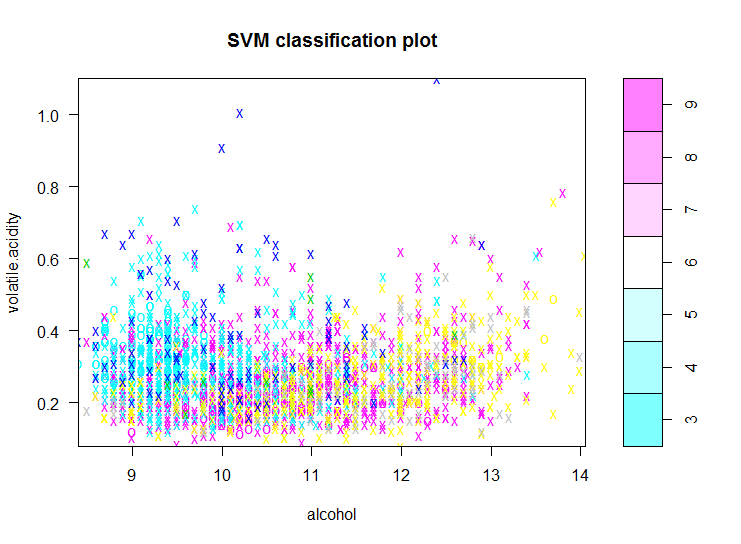


Figure 11

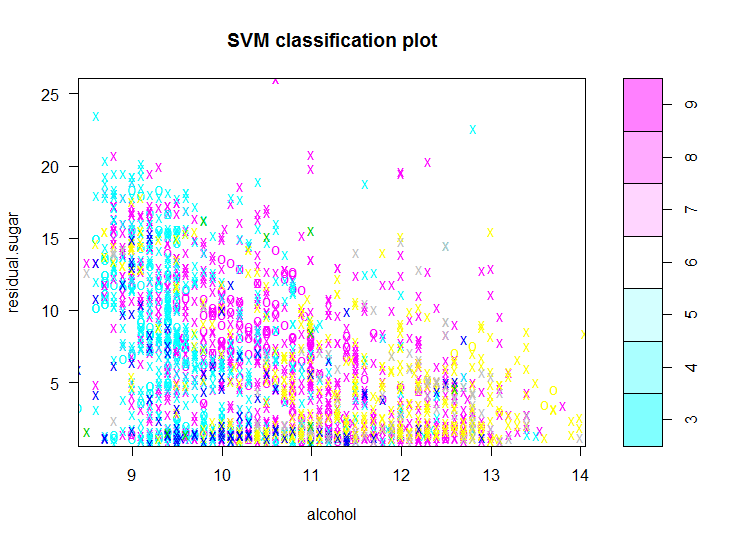


Figure 12

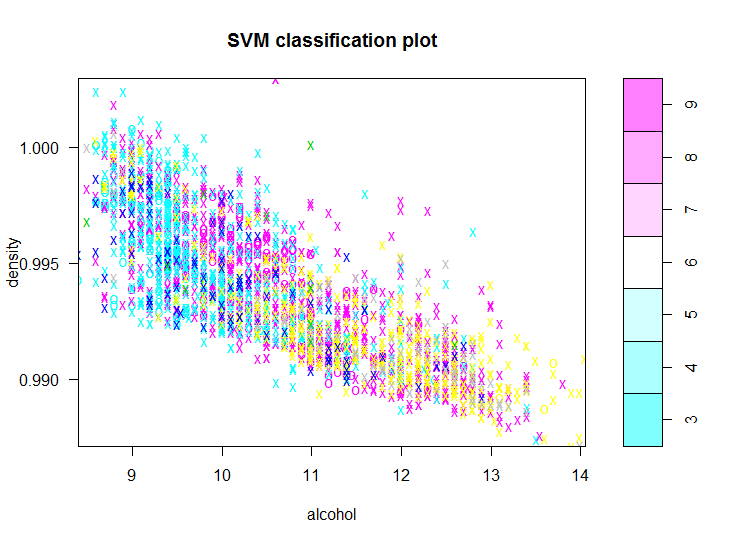
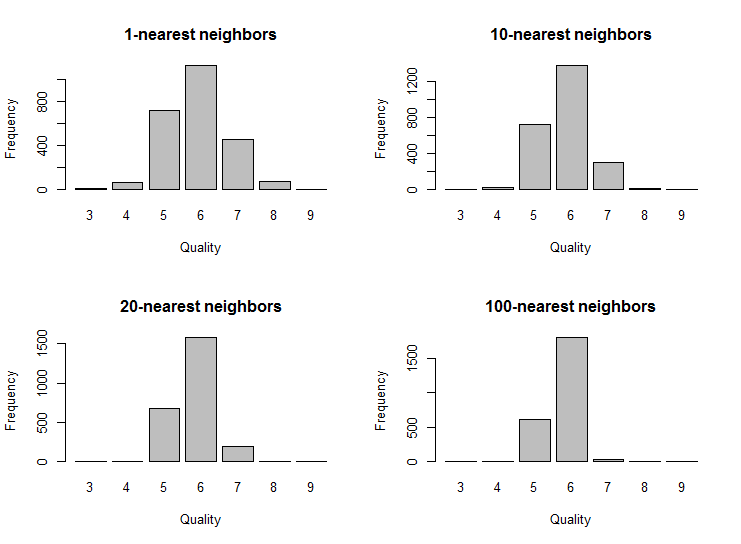


Figure 13

Figure 9 to 13 shows the classifications based on two variables (Alcohol vs other), lower grade wine (grade 3 to 5) are more populated in lower alcohol level. But higher end wine are not necessary only in higher alcohol level. In fact higher end wine are scattered across different alcohol level. Alcohol level plays a key factor to determine the quality.

# *k*-nearest neighbors

White wine training and testing datasets are used in this k-nearest neighbors. Different *k*s (1, 10, 20 and 100) were applied to different runs.



It is interesting to notice that some quality 3 and 8 can be found when 1-nearest neighbor. On the other hand, the result remains very close when k is 10, 20 or 100.

That is related to dataset where both ends of quality (3 and 9) are very rare in both training and testing datasets. Table 1 shows that there are few than 10 samples of quality 3 and 9 each.

As the k increases, quality 4, 7 and 8 become harder to be classified. Again, this reflects many quality 4, 7 and 8 are very close to quality 5 and 6. As soon as k is increased, it is very easy to be classified to be quality 5 or 6.

Chart 1

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Prediction Class | | | | | | |
|  |  | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| Actual Class | 3 | 1 | 2 | 3 | 4 | 0 | 0 | 0 |
| 4 | 3 | 10 | 19 | 27 | 6 | 1 | 0 |
| 5 | 3 | 23 | 389 | 237 | 61 | 7 | 0 |
| 6 | 0 | 19 | 246 | 659 | 167 | 23 | 0 |
| 7 | 1 | 7 | 56 | 159 | 210 | 17 | 0 |
| 8 | 0 | 0 | 7 | 38 | 12 | 28 | 0 |
| 9 | 0 | 0 | 0 | 3 | 1 | 0 | 0 |

Table 9 (k=1)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Prediction Class | | | | | | |
|  |  | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| Actual Class | 3 | 1 | 1 | 2 | 6 | 0 | 0 | 0 |
| 4 | 0 | 4 | 28 | 31 | 2 | 1 | 0 |
| 5 | 0 | 12 | 299 | 360 | 44 | 5 | 0 |
| 6 | 0 | 7 | 286 | 689 | 127 | 5 | 0 |
| 7 | 0 | 1 | 67 | 287 | 92 | 3 | 0 |
| 8 | 0 | 0 | 20 | 42 | 22 | 1 | 0 |
| 9 | 0 | 0 | 0 | 4 | 0 | 0 | 0 |

Table 10 (k=10)

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Prediction Class | | | | | | |
|  |  | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
| Actual Class | 3 | 0 | 0 | 2 | 8 | 0 | 0 | 0 |
| 4 | 0 | 0 | 20 | 46 | 0 | 0 | 0 |
| 5 | 0 | 0 | 280 | 435 | 5 | 0 | 0 |
| 6 | 0 | 0 | 259 | 842 | 13 | 0 | 0 |
| 7 | 0 | 0 | 51 | 393 | 6 | 0 | 0 |
| 8 | 0 | 0 | 8 | 77 | 0 | 0 | 0 |
| 9 | 0 | 0 | 0 | 4 | 0 | 0 | 0 |

Table 11 (k=100)

Using k=1, the confusion matrix indicated that it was able to predicting classify 3 at least one and actual quality 3 has been spread across in prediction class 4, 5 and 6. This also applies to actual class 8. Regarding actual class 9, it cannot be classified in any k values. In fact, actual 9 cannot be classified in any supervised learning.

Going back to the white wine training set, there is only one class 9 sample in the entire white wine training set.

As “k” increases, the prediction will be shifting to the middle populated class 5 and 6.

Chart 1 indicates that it has highest probability of predicting correct class when k = 1, it dropped significantly when k is between 1 and 10. And then the probability increase slightly when k is 20. From k=20 to k=100, there is no improvement at all.

Although k-nearest neighbor, this instance-based learning, has achieved similar accuracy as other learning algorithms, but it has zero time of training. On the other hand, if the actual classes are very close and a few outliners like this white wine data sets, it is not easy to generalize to improve the accuracy.