

**Homework 5**

<Your Name Here>

**Team:** Sabesan, Daniel, Saurabh, Gustavo, Huong, Jinua, Steve

**1) In homework 3 problem 4 ridge regression was performed on the wine quality data set : winequality-red.csv. Now use k-nearest neighbors to classify this data. Use cross validation to choose the best value for k. Round the results from the ridge regression to the nearest integer to form the classification with ridge regression. Using the best values for k (nearest neighbor) and lambda (ridge regression), compare and contrast the results of these two classification techniques.**

For RR, the minimum test error was “0.55” for a lambda value of “25.1” (see Homework 3 problem 4 for details). For KNN minimum error is “0.34” for “k=1” which is much lower than the value obtained with RR. It appears that KNN has done a much better job at classifying the data.

The following corresponds to k=1

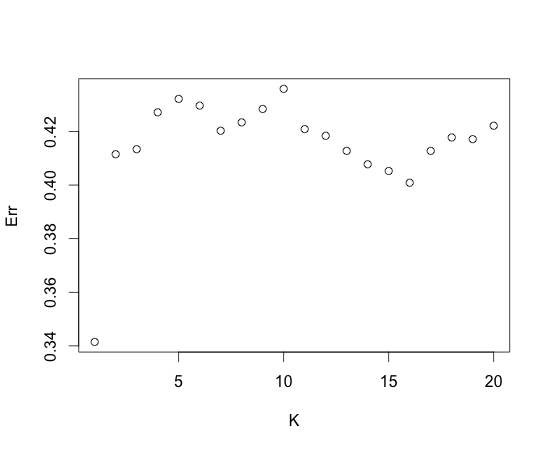
> which( Err == min(Err))

[1] 1

Err[ which( Err == min(Err)) ]

# [1] 0.3414634

Below the error is plotted for the various values of “K”. A value of “K=1” seems to be much lower than higher values of “K”.



**2) Use k-nearest neighbors to classify the Iris data set. Compare the k-nearest neighbor results with the results obtained in class using the Naive Bayes Classifier.**

Using the k-nearest neighbor algorithm, for k=12, I was able to get the minimum error of 0.03333. Naïve Bayes classifier produced an error of 0.04. Hence, there was a reduction of 1% in the error.

Err

###> Err

[1] 0.05333333 0.05333333 0.05333333 0.06000000 0.05333333 0.04000000

[7] 0.04000000 0.05333333 0.04666667 0.04000000 0.04666667 0.03333333

[13] 0.03333333 0.03333333 0.03333333 0.03333333 0.03333333 0.04000000

[19] 0.04666667 0.04000000 0.05333333 0.05333333 0.04666667 0.04666667

[25] 0.04666667 0.04666667 0.05333333 0.04000000 0.05333333 0.05333333

[31] 0.05333333 0.06000000 0.05333333 0.06000000 0.05333333 0.05333333

[37] 0.08666667 0.09333333 0.10000000 0.10666667 0.11333333 0.11333333

[43] 0.10666667 0.10666667 0.10666667 0.10666667 0.11333333 0.12000000

[49] 0.12666667 0.13333333

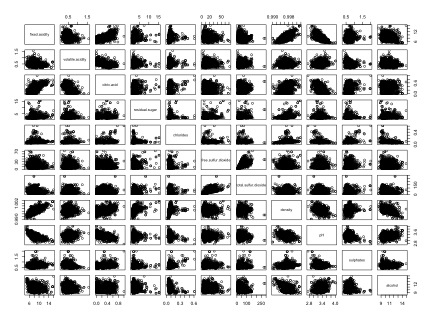
**3) Classify the wine quality data using Naive Bayes. Compare the results with the two methods described in problem 1 of this homework set. Think about why one of the methods used works better than the others.**

Naïve Bayes results in an error of around 44%. This is much higher than the knn approach used in Problem 1. A likely reason for this poor performance is the assumption of independence in attributes, as used by the Naïve Bayes classifier. We can see this by plotting the correlations between variables (as shown in the figure below). One can clearly see, for instance, that acidity and density are not independent.

Err <- 1 - sum(qualityHat$class == wdata$quality)/length(wdata$quality)

Err

# [1] 0.4396498



**4) Classify the sonar data using Naive Bayes. Compare the results with the methods used in class and with the last homework set. Give reasons for any discrepancies between the results for these methods. (Either in class or in homework, the following methods have been used on this data set: Trees, Linear Regression, Ridge Regression, an Ensemble Method, and now Naive Bayes.)**

Using Naïve Bayes, an error of “0.27” was obtained. Below is a list of some of the other methods and the error rates obtained:

* Naïve Bayes: 0.27 (as mentioned above)
* Random Forest: 0.14
* Trees: 0.21 (rpart & RR to combine trees)
* Lin. Reg. w/ CV: 0.25 (hw3, used “lm” with CV).
* Lin. Reg. w/ CV & Ensemble: 0.18 (hw3, “lm” and ensemble).
* Ridge Regression: 0.19 (from lmvsridgeSonarData.R)

Random forests and Linear regression, and Ridge Regression, produced the best results. This is likely due to these being better at handling outliers or noise. The poorest performers were LR w/ Cross Validation (non-ensemble) and Naïve Bayes. These may not be as resilient to noise in the data.

**5) Run the code in the file KfirstNearestNeighbor.R Does KNN create a better model if the data is first scaled and normalized? What should be chosen as the best value for k and why? Now use KNN with cross validation on the mixtureSimData.data. What is the best value for k for this data?**

1. Yes KNN does create a better model when data is first scaled and normalized.

With scaling and normalization:

> Err

[1] 0.1250000 0.1442308 0.1346154 0.1730769 0.1778846 0.1923077 0.1923077 0.2211538 0.2067308 0.2115385

[11] 0.2403846 0.2451923 0.2740385 0.2740385 0.2740385 0.2932692 0.2836538 0.2788462 0.2884615 0.2932692

Without scaling and normalization:

> Err

[1] 0.1730769 0.2115385 0.1875000 0.2067308 0.1730769 0.1971154 0.2307692 0.2451923 0.2644231 0.2932692

[11] 0.3221154 0.3317308 0.3413462 0.3413462 0.3317308 0.3413462 0.3413462 0.3221154 0.3269231 0.3076923

Better model with scaling and normalization.

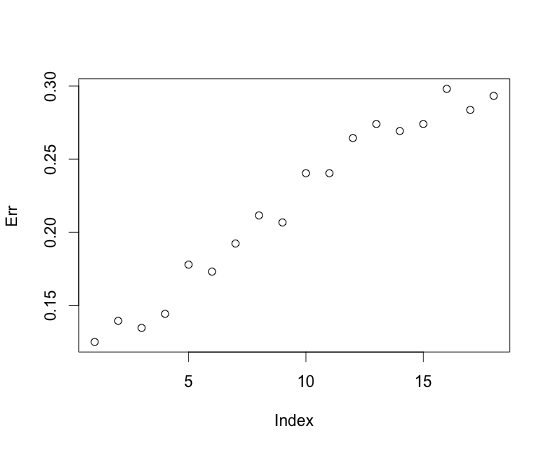
1. Ran the cross validation with multiple k’s i.e. nearest neighbors. After about k=18 the errors flattens off.

Number of k values: [1] 18

> Err

[1] 0.1250000 0.1394231 0.1346154 0.1442308 0.1778846 0.1730769 0.1923077 0.2115385 0.2067308 0.2403846

[11] 0.2403846 0.2644231 0.2740385 0.2692308 0.2740385 0.2980769 0.2836538 0.2932692

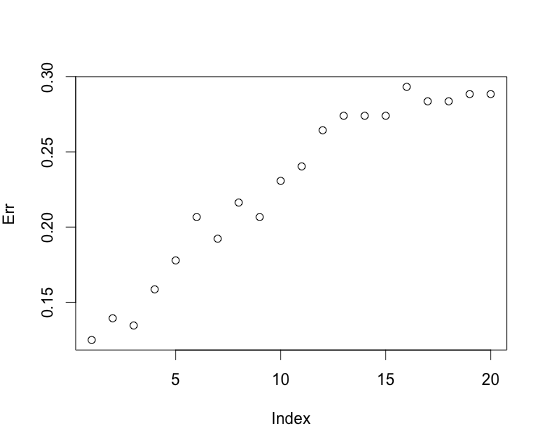


Number of k values: [1] 20

> Err

[1] 0.1250000 0.1394231 0.1346154 0.1586538 0.1778846 0.2067308 0.1923077 0.2163462 0.2067308 0.2307692

[11] 0.2403846 0.2644231 0.2740385 0.2740385 0.2740385 0.2932692 0.2836538 0.2836538 0.2884615 0.2884615



1. mixtureSimData has ONLY one column. The classifier column is missing and so unable to do any training followed by test
2. Unable to find the best value for k for this data since the classifier column is missing.