CSC570AG Machine Learning
Homework 2

Applying k-Nearest Neighbors to predict MPG
(due by the end of the day on Sunday, March 5th)

1. Download the dataset Auto.csv.
2. Explore the overall structure of the dataset using str(). Describe it one paragraph.
3. Convert the attribute horsepower from character to integer.
4. The horsepower attribute has some missing values. Remove the observations with missing values, i.e., delete the rows with missing values from the data frame.
5. Explore the data in order to investigate the association between mpg and the other features. Which of the other features seem most likely to be useful in predicting mpg (scatterplots may be useful tools to answer this question). Describe your findings.
6. Create a new attribute mpg1 that contains 1 if mpg is strictly greater than its median, and 0 if mpg is equal or less than its median.
7. Decide which attributes you are going to use to predict mpg1. Remove all remaining attributes, including mpg.
8. Set the seed of the random number generator to a fixed integer, say 1, so that you can reproduce your work:
   > set.seed(1)
9. Normalize the attribute values
10. Randomize the order of the rows in the dataset
11. Split the data into a training set and a test set. Use a test set of 100 rows.
12. Perform kNN on the training data, with several values of *K*, in order to predict mpg1. What test errors do you obtain? Which value of *K* seems to perform the best on this data set?

Commands you might need, which are not on the lecture slides:

as.intereger():	to coerce a string to integer
sapply(<vector>,<function>):	to apply a function to each element of a vector and return a vector of results
sample(nrow(<data frame)):	to create a random sample from 1 to the number of rows in <data frame>

Your submission must consist of two text files:
- a text file, description.txt, no longer than a page: Your answers to 2, 5, 7, and 12
- a text file, script.txt, with history of your session. Save the session into script.txt using:

> savehistory("script.txt")

It is important that your script.txt is clean, i.e., it does not contain any “drafts” or “debugging”. You can manually edit script.txt in order to remove any unnecessary commands, such as trials and errors. The order of the commands must follow the order stated above (1:12).

Let me know if you have questions.