Assignment 2

Due: Oct.12

Submission Instructions

- **Your program must run on machines in Leon Lowenstein Bldg. 812**
- **Create a README file, with simple, clear instructions on how to compile and run your code**
- **Zip all your files (code, README, written answers, etc.) in a zip file named** $\{firstname\}\_\{lastname\}\_CS6930\_HW2.zip$ **and upload it to Blackboard**

1. (30 points) Implement the KNN classifier.

   Your implementation should accept two data files as input (both are posted with the assignment): a **spam_train.csv** file (**weka_spam_train.arff** for Weka users) and a **spam_test.csv** file (**weka_spam_test.arff** for Weka users). Both files contain examples of e-mail messages, with each example having a class label of either "1" (spam) or "0" (no-spam). Each example has 57 (numeric) features that characterize the message. Your classifier should examine each example in the **spam_test** set and classify it as one of the two classes. The classification will be based on an **unweighted** vote of its k nearest examples in the **spam_train** set. We will measure all distances using regular Euclidean distance:

$$d(x, y) = \sqrt{\sum_i (x_i - y_i)^2}$$

   (a) Report **test** accuracies when $k = 1, 5, 11, 21, 41, 61, 81, 101, 201, 401$ **without** normalizing the features.

   (b) Report **test** accuracies when $k = 1, 5, 11, 21, 41, 61, 81, 101, 201, 401$ **with z-score normalization** applied to the features.

   (c) In the **(b)** case, generate an output of KNN predicted labels for the first 50 instances (i.e. $t1$ - $t50$) when $k = 1, 5, 11, 21, 41, 61, 81, 101, 201, 401$ (in this order). For example, if $t5$ is classified as class 'spam" when $k = 1, 5, 11, 21, 41, 61$ and classified as class "no-spam" when $k = 81, 101, 201, 401$, then your output line for $t5$ should be:

   $t5$ **spam, spam, spam, spam, spam, spam, no, no, no, no**

   (d) What can you conclude by comparing the KNN performance in (a) and (b)?

   (e) Describe a method to select the optimal k for the KNN algorithm.

2. (30 points) Decision Tree

Table 1 below contains a small training set. Each line includes an individual's education, occupation choice, years of experience, and an indication of salary. Your task is to create a complete decision tree including the number of low's & high's , entropy at each step and the information gain for each feature examined at each node in the tree.

Instance	Education Level	Career	Years of Experience	Salary
1	High School	Management	Less than 3	Low
2	High School	Management	3 to 10	Low
3	College	Management	Less than 3	High
4	College	Service	More than 10	Low
5	High School	Service	3 to 10	Low
6	College	Service	3 to 10	High
7	College	Management	More than 10	High
8	College	Service	Less than 3	Low
9	High School	Management	More than 10	High
10	High School	Service	More than 10	Low

Table 1: Decision Tree Training Data

Please turn in a diagram similar to:

Top 6,4, .97
Education gain = <to be calculated>
1. High School 4,1, <to be calculated>
Experience gain = <to be calculated>
Etc.
Etc.

Prune the tree you obtained using the validation data given in Table 2. Show your work.

Instance	Education Level	Career	Years of Experience	Salary
1	High School	Management	More than 10	High
2	College	Management	Less than 3	Low
3	College	Service	3 to 10	Low

Table 2: Validation Data

3. (20 points) SVM using Weka

   For this exercise, we apply SVM with several different kernels and hyper-parameter choices to the **veh-prime.arff** file provided with the assignment. Import this file into Weka (free download from http://www.cs.waikato.ac.nz/ml/weka/) and then select the SMO classifier found under classifiers/function. Use 10 fold cross-validation. You can make kernel and hyper-parameter choices by clicking on "SMO ..." appearing next to Choose.

   You will make 5 runs of the algorithms. Select PolyKernel with exponent option 1, 2, and 4. Then select RBFKernel with gamma set to 0.01 and 1.0. For each run record the number of correctly and incorrectly classified instances. Explain why some of the choices do not work well.

4. (20 points) Kernels

   Assume that $x = (x_1, x_2)$ is a two dimensional vector and we have a function K defined as $K(x, z) = x_1 * z_1 + x_1 * e^{z_2} + z_1 * e^{x_2} + e^{x_2+z_2}$. Prove that K is a kernel.