Assignment 2

Due: March 4

Submission Instructions

- **If you use Python, create a README file, with simple, clear instructions on how to compile and run your code.** *If the TA cannot run your program by following the instructions, you will receive 50% of the programing score.*
- **If you use Weka, submit screenshots to show your work.**
- **Zip all your files (code, README, written answers, etc.) in a zip file named** $\{firstname\}\_\{lastname\}\_CS5790\_HW1.zip$ **and upload it to Blackboard**

1. (40 points) Implement the KNN classifier.

   Note: *This question carries an additional 10 points bonus if you implement the model in Python. You can call existing Python functions for normalzation and Euclidean distance calculations. But you are not allowed to call the KNN model provided by the Python libraries.*

   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)?

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.

Table 1: Decision Tree Training Data

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

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**.

Table 2: Validation Data

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

3. (30 points) Build a Naive Bayes classifier for the given training data in Table 1 with the **add 1 smoothing** technique covered in the lecture slides. Use your model to classify the following new instances:

Instance	Education Level	Career	Years of Experience
1	High School	Service	Less than 3
2	College	Retail	Less than 3
3	Graduate	Service	3 to 10