This is a closed book, closed notes exam. Switch off your cell phone and do not communicate with anyone other than an exam proctor.

Start writing when instructed. Stop writing when your time is up.

Remember that your work is graded on the quality of your writing and explanation as well as the validity of the mathematics.

(1) (5 Points) Your goal is to build a classifier that will help filter spam emails from your inbox. Suppose you select 100 emails from your inbox uniformly at random (i.e. each email in your inbox is equally likely to be selected). For each of these 100 emails, you read the email, decide whether it is spam, and assign it a binary label accordingly. You take the first 60 labeled examples and make this the training set. The next 20 you make the validation set, and the final 20 you make the test set.

You decide to build a $K$-NN classifier using edit distance as your distance function. For $K = 1$, you get a train error of 0.0% and a validation error of 11.2%. For $K = 2$ you get a train error of 8.5% and a validation error of 9.4%. If you had to pick between these two classifiers, which should you expect to achieve a lower error on the test set? Why?

Solution: You should expect the classifier with the lower validation error to achieve a lower test error. Because the system has not been trained on the validation set, the validation error is a better predictor of future performance. Since the test set has never been trained on or evaluated on in this experiment, it has the same expected error as future emails which you haven’t seen yet.

(2) (5 Points) Now, someone gives you access to a huge database of emails with spam labels so that you can train and evaluate on as many emails as you like. You’re worried about how long the computation will take if you use large training and validation sets. If you’re using a 1-NN classifier, and you assume edit distance takes constant time to compute, how long will it take your system (asymptotically) to compute validation error for a validation set of $m$ examples while using a training set of $n$ examples? Why? [Use big-O notation in terms of $n$ and $m$. Assume your system is implemented efficiently.]

Solution: For each validation example, you must enumerate all $n$ training examples in order to find the one with lowest distance. (Note: Sorting the data by distance would be slower and is unnecessary. Computing the min directly is only linear time.) Since there are $m$ validation examples to make predictions on, the asymptotic runtime is $O(n \cdot m)$. 
