CS 6350 Sample Midterm

1. How would you train a decision tree using the ID3 algorithm if some attributes are missing? (You might get asked to step through this procedure for a small data set like the Tennis data in the lecture.)

   - There are multiple ways to do this, but the most common is to end the branch at that location and choose the most common label at that point of the decision tree.

2. Show that the following dataset is linearly separable by providing a linear threshold unit that correctly classifies the example

x_1	x_2	x_3	y
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0

   - This is in effect just solving a multivariable problem to find a weight vector which can be considered a threshold line. doing so shoes all values except $w_3 = 1$.

3. How would you avoid overfitting when you use the decision tree algorithm? Why might shorter decision trees be more robust to noise in the training data?

   - You can use back greedy back propagation though the "held-out-set" aproach which checks the tree's performance on the creation of a new level against a held out subset of the training data. If the performance begins to decrease after some level then the tree is kept pruned to that level. SHorter trees are more robust because they keep expressivity and avoid over training.

4. (An exercise question from the class lecture) Suppose you want to build a nearest neighbors classifier to predict whether a beverage is a coffee or tea using two features: The volume of the liquid (in milliliters) and the caffeine content (in grams). You collect the following data

Volume (mL)	**Caffeine (g)**	**Label**
238	0.026	Tea
100	0.011	Tea
120	0.040	Coffee
237	0.095	Coffee

   What is the label for a test point with Volume = 120, Caffeine = 0.013? Why might this be incorrect? How would you fix this problem?

   - The "shortest distance" would classify this data as *Coffee*, as the distance would only be 0.027, however this is more than likely wrong because of the different scales of the measurements. A way to fix this would be to normalize the attributes in each feature to correct for this. By doing this, the best guess would that it should be *Tea*. This could also be done by looking at the ratio between the volume and caffeine, henceforth dubbed the *caffeine density*, but I believe that the caffeine density is not the answer to this question :-(CS people are dumb **;..(**
   - you can also dived the cafeine to volume which is in effect density and coffee will have density nearer to each other and likewise tea.

5. (An exercise question from the class lectures) What will happen when you choose K to the number of training examples for a K-nearest neighbor classifier?

   - This breaks the trianing data into K-sets, where $K-1$ subsets of the data are trained on and the one that is left out is tested over. Each subset of training data is tested, and the accuracy is done by averaging over each of the subset values. This is done so that the hyperparameters can be tested without revealing the training data and causing over fitting.

6. For each function below, state whether it can be written as a linear threshold unit in terms of the variables specified. If it can be written as one, write the linear threshold unit that is equivalent to the function. If not, suggest a transformation of the underlying space so that the function is linear in the new space.

(a) $\neg x_1$
- $x \leq 1$

(b) $x_1 \vee \neg x_2$
- $1 + -1$ gives threshold ≥ 0

(c) $(x_1 \vee \neg x_2) \wedge (\neg x_1 \vee x_3)$
- min for positive result in each disjunction group is 1. The conjunstion of the two gives 1+1. threshold ≥ 2

7. Show that the Halving algorithm for the finite concept space C will not make more than $\log |C|$ mistakes. Apply this to get a limit on the number of mistakes the algorithm will make for the class of k-conjunctions of n Boolean variables.
   - Solution goes here

8. State with an explanation whether the following are true and false:
   (a) The mistake bound model assumes that training and test examples are drawn from the same fixed, but unknown distribution
   - False this is the assumption of batch learning. Mistake bound has no assumption about the distribution.

   (b) The Perceptron mistake bound theorem guarentees that the algorithm will find a linear separator for *any* dataset
   - False, the theorem states *given* a linearly seperable data set you will converge to a linear seperater in R^2/γ^2 mistakes

   (c) Online online learning, batch learning does not seek to minimize the number of mistakes that the learner makes
   - batch learning does not, online does. batch seeks to find a hypothesis that has a low probability of making a mistake, not a low bound to the amount of mistakes it can make.

9. Prove the Perceptron mistake bound
   - Solution goes here

10. How many mistakes will the Perceptron algorithm make for disjunctions with n attributes? To answer this, you will first have to identify what R and γ are for this concept class
    - Solution goes here

11. Prove the Winnow mistake bound
    - Solution goes here

12. You are given a binary classificaiton dataset where the examples are 100,000 dimensional Boolean vectors. You suspect that the true classifier could not be a function of more than 100 features. Given this information, would you prefer using the Perceptron or Winnow algorithm for learning? Why?
    - Solution goes here

13. You wish you learn a hidden concept f using m training examples that are drawn from a distribution D. If the training set is called S and the hypothesis that your learning generates is h, write expressions for the training and generalization errors.
    - Solution goes here

14. Suppose our learning problem has n binary features. What is the size of the hypothesis space consisting of all decision trees over this space?
    - 2^{2^n}