Due data: 4/19/2020, end of the day. **Please submit the following 2 files via Canvas:**

1) For question 1 – 3, please submit in a word file or a PDF file;
2) For question 4, please submit a **.ipynb** file (Python jupiter notebook file).

Question 1 (10 points):

Consider again the example application of Bayes rule in Section 6.2.1 of Tom Mitchell's textbook (or slide page 6 of Lecture 6-2). Suppose the doctor decides to order a second laboratory test for the same patient, and suppose the second test returns a positive result as well. What are the posterior probabilities of *cancer* and $\neg$*cancer* following these two tests? Assume that the two tests are independent.

Question 2 (5 points):

Consider a learned hypothesis, *h*, for some Boolean concept. When *h* is tested on a set of 100 examples, it classifies 80 correctly. What is the 95% confidence interval for the true error rate for $Error_D(h)$?

Question 3 (15 points):

Consider a two-layer feedforward ANN with two inputs a and b, one hidden unit c, and one output unit d. This network has five weights $(w_{ca}, w_{cb}, w_{c0}, w_{dc}, w_{d0})$, where w_{x0} represents the threshold weight for unit x. Initialize these weights to the values $(.1, .1, .1, .1, .1)$, then give their values after each of the first two training iterations of the BACKPROPAGATION algorithm. Assume learning rate $\eta = .3$, momentum $\alpha = 0.9$, incremental weight updates, and the following training examples:

a	b	d
1	0	1
0	1	0

Question 4 – Programming (40 points):

In this programming problem, you will get familiar with building a neural network using backpropagation. You are supposed to implement the following steps:

Step 1: use our "titanic" dataset in homework #3, and split data in the same way you did in homework #3 as training and test sets;

Step 2: fit a neural network using independent variables 'pclass + sex + age + sibsp' and dependent variable 'survived'. Omit all NA examples. Use 2 hidden layers and set the activation functions for both the hidden and output layer to be the **sigmoid** function. Set "solver" parameter as either **SGD** (stochastic gradient descend) or **Adam** (similar to SGD but optimized performance with mini batches). You can adjust parameter "alpha" for regularization (to control overfitting) and other parameters such as "learning rate" and "momentum" as needed.

Step 3: check the performance of the model: in-sample and out-of- sample accuracy, defined as

in-sample percent survivors correctly predicted (on training set)
in-sample percent fatalities correctly predicted (on training set)
out-of-sample percent survivors correctly predicted (on test set)
out-of-sample percent fatalities correctly predicted (on test set)

Please try two different network structures (i.e., number of neurons at each hidden layer) and show their respective accuracy.

Step 4: compare the in-sample and out-of-sample accuracy (as defined in step 3) with the pruned decision tree obtained in homework #3. (You can either use a table or a figure to compare the accuracy of the two learning algorithms)

Note: There are two options to implement the neural network:

Option 1: use scikit-learn library;

Here is the tutorial: http://scikit-learn.org/stable/modules/neural_networks_supervised.html

Option 2 (bonus: 5 points): implement backpropagation yourself; in your implementation, you better set the following:
(1) the initial weights to be uniformly between [-0.1, +0.1]
(2) the number of iterations to be at least 5000 or even more

You can choose either option for this homework. You will get 5 bonus points if you choose option 2. No matter what you choose, make sure you know how to update the weights.