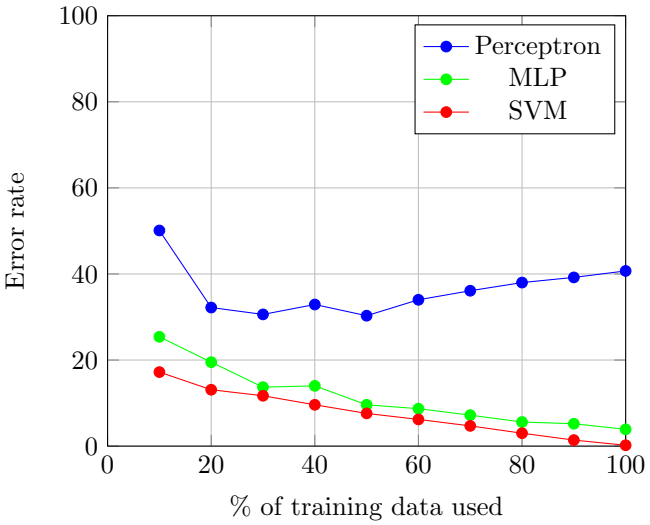
Assignment 2

Kyle Reagle and Yigit Gungor

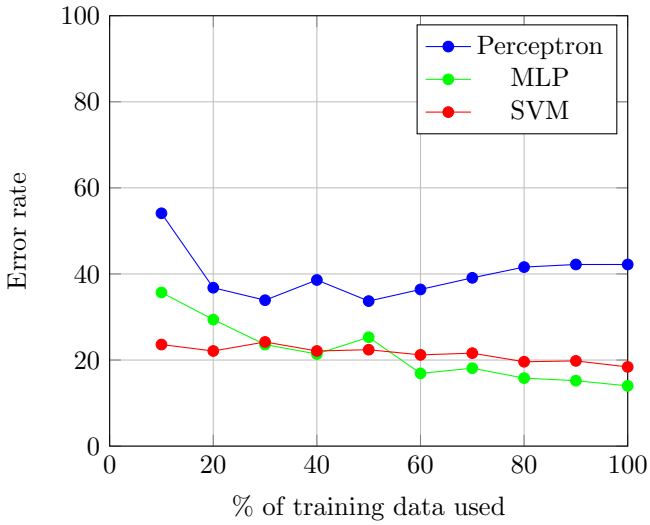
October 23, 2017

Question 1:

Predictions performed on the complete training data set



Predictions performed on the complete testing data set



Based on the graphs, it's obvious that the perceptron performed the worst of all the algorithms, especially since it was the slowest. Initially, the perceptron's prediction rate improved, but slowly got worse as more training data was used. The MLP and the SVM all improved their prediction rate as more training data was given. Since the perception and SVM were relatively easy to program, there wasn't much fine tuning to be done, and there didn't seem to be any failure conditions. The MLP was very inaccurate with high learning rates (alpha) around 1, so the learning rate had to be set around 0.0001 for the best results. The MLP also seemed to perform the best with 50 hidden nodes. Anything higher or lower than 50 seemed to decrease the prediction performance. All the algorithms had lower error rates as the number of iterations increased, but it obviously slowed down run time. For example, the perceptron got to around 15% error rate at 25 iterations, but that required around half an hour of run time.

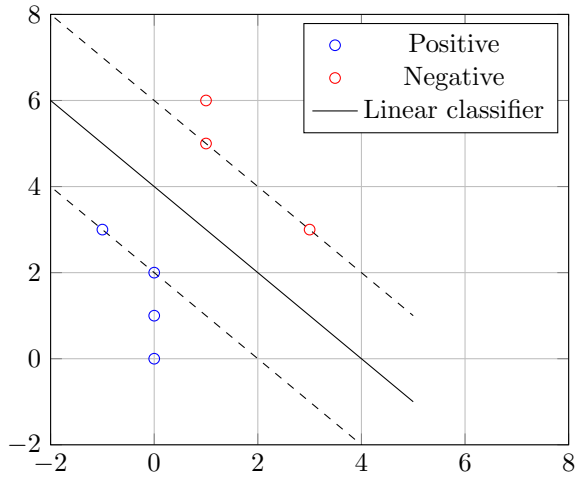
Question 2:

(a) The provided tree correctly categorizes the provided examples.
Below is a table showing the output of the tree corresponding to the attributes given in the table.

No.	Table	Tree
1	P	P
2	P	P
3	P	P
4	P	P
5	P	P
6	P	P
7	N	N
8	N	N
9	N	N
10	N	N
11	N	N
12	N	N

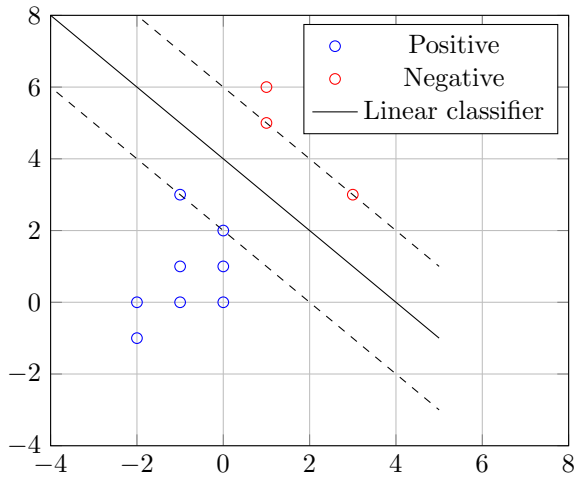
Question 3:

(a)



(b) $w = -1, b = 4$

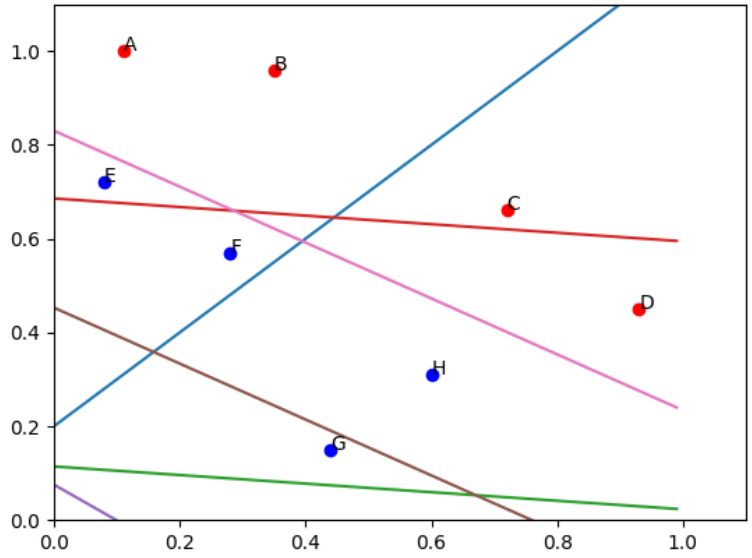
(c) The linear SVM is still $h(x) = -x + 4$



Question 4:

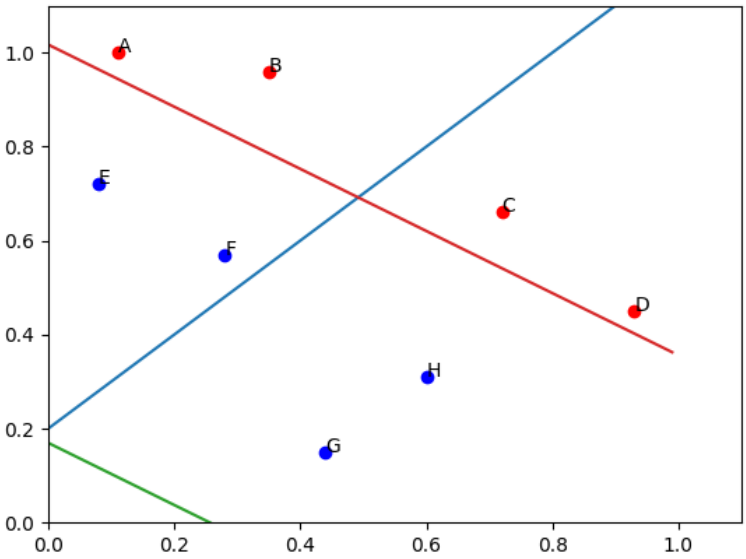
(a) Below are two examples of output from a program we wrote to classify the data points.

Figure 1: Final linear separator is the pink line



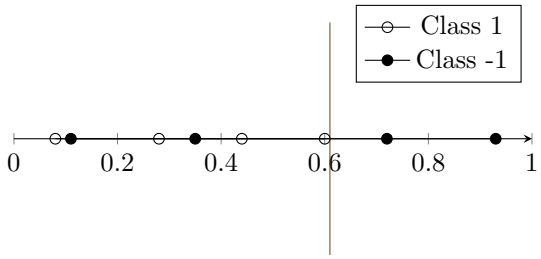

```
Iteration 1
Missclassified points: ['C', 'D', 'E', 'F']
New weight vector: [-0.8, -0.43999999999999995, -2.3200000000000003]
Iteration 2
Missclassified points: ['E', 'F', 'G', 'H']
New weight vector: [0.19999999999999996, -0.15999999999999992, -1.7500000000000004]
Iteration 3
Missclassified points: ['E', 'F', 'G', 'H']
New weight vector: [1.2, -0.15999999999999992, -1.7500000000000004]
Iteration 4
Missclassified points: ['D', 'E']
New weight vector: [0.19999999999999996, -2.02, -2.6500000000000004]
Iteration 5
Missclassified points: ['E', 'F', 'G', 'H']
New weight vector: [1.2, -1.58, -2.6500000000000004]
Iteration 6
Missclassified points: ['E', 'F', 'H']
New weight vector: [2.2, -1.58, -2.6500000000000004]
```

Figure 2: Final linear separator is the red line




```
Iteration 1
Missclassified points: ['C', 'D', 'E', 'F']
New weight vector: [-0.8, -0.8600000000000001, -1.9]
Iteration 2
Missclassified points: ['E', 'F', 'G', 'H']
New weight vector: [0.19999999999999996, -0.7800000000000001, -1.18]
Iteration 3
Missclassified points: ['E', 'F', 'G', 'H']
New weight vector: [1.2, -0.7800000000000001, -1.18]
```

(c) The plot below shows the line that misclassifies the least number of data points, which are the two black dots to the left of the linear separator. This line can be represented with a weight ratio of $-w_0/w_1 = 0.6$



$MSE = ((0.6 - 0.11)^2 + (0.6 - 0.35)^2)/8 = 0.038$