Lab week 7: Classifiers

1. Decision Tree for heart attack

The following table contains training examples that help predict whether a patient is likely to have a heart attack.

PATIENT ID	CHEST PAIN?	MALE?	SMOKES?	EXERCISES?	HEART ATTACK?
1.	yes	yes	no	yes	yes
2.	yes	yes	yes	no	yes
3.	no	no	yes	no	yes
4.	no	yes	no	yes	no
5.	yes	no	yes	yes	yes
6.	no	yes	yes	yes	no

A. Use information theory to construct a minimal decision tree that predicts whether or not a patient is likely to have a heart attack. SHOW EACH STEP OF THE COMPUTATION.

x	1	¼	2/3	3/4	1/3	1/2
$\log_2$_(x)	0	-2	-0.6	-0.4	-1.6	-1

Split Round #1

Entropy: -(4/6) * log2(4/6) - (2/6) * log2(2/6) = 0.918

a. Chest Pain

Entropy “Yes” : -3/3 * log2(3/3) = 0
Entropy “No” : -1/3 * log2(1/3) - 2/3 * log2(2/3) = 0.918
Entropy: (3/6 * 0) + (3/6 * 0.918) = 0.459
Gain: 0.918 - 0.459 = 0.459
Gain Ratio: 0.459 / (-3/6 * log2(3/6) - 3/6 * log2(3/6)) = **0.459**

b. Male:

Entropy “Yes” : -2/4 * log2(2/4) - 2/4 * log2(2/4) = 1
Entropy “No” : -2/2 * log2(2/2) = 0
Entropy: (4/6 * 1) + (2/6 * 0) = 0.667
Gain: 0.918 - 0.667 = 0.251
Gain Ratio: 0.251 / (-4/6 * log2(4/6) - 2/6 * log2(2/6)) = **0.273**

c. Smokes:

Entropy “Yes” : -3/4 * log2(3/4) - 1/4 * log2(1/4) = 0.811
Entropy “No” : -1/2 * log2(1/2) - 1/2 * log2(1/2) = 1
Entropy: (4/6 * 0.811) + (2/6 * 1) = 0.874
Gain: 0.918 - 0.874 = 0.044
Gain Ratio: 0.044 / (-4/6 * log2(4/6) - 2/6 * log2(2/6)) = **0.0479**

d. Exercises:
   Entropy “Yes” : -2/4 * log2(2/4) - 2/4 * log2(2/4) = 1
   Entropy “No” : -2/2 * log2(2/2) = 0
   Entropy: (4/6 *1) + (2/6 * 0) = 0.667
   Gain: 0.918 - 0.667 = 0.251
   Gain Ratio: 0.251 / (-4/6 * log2(4/6) - 2/6 * log2(2/6)) = **0.273**

Round #1 - Split on ‘Chest Pain’ - Yes, Branch No

Split Round #2
Entropy: -(1/3) * log2(1/3) - (2/3) * log2(2/3) = 0.918

a. Male:
   Entropy “Yes” : -2/2 * log2(2/2) = 0
   Entropy “No” : -1/1 * log2(1/1) = 0
   Entropy: (2/3 * 1) + (1/3 * 0) = 0
   Gain: 0.918 - 0 = 0.918
   Gain Ratio: 0.918 / (-2/3 * log2(2/3) - 1/3 * log2(1/3)) = **0.99**

b. Smokes:
   Entropy “Yes” : -1/2 * log2(1/2) -1/2 * log2(1/2) = 1
   Entropy “No” : -1/1 * log2(1/1) = 0
   Entropy: (2/3 * 1) + (1/3 * 0) = 0.667
   Gain: 0.918 - 0.667 = 0.251
   Gain Ratio: 0.251 / (-2/3 * log2(2/3) - 1/3 * log2(1/3)) = **0.27**

c. Exercises:
   Entropy “Yes” : -2/2 * log2(2/2) = 0
   Entropy “No” : -1/1 * log2(1/1) = 0
   Entropy: (2/3 * 1) + (1/3 * 0) = 0
   Gain: 0.918 - 0 = 0.918
   Gain Ratio: 0.918 / (-2/3 * log2(2/3) - 1/3 * log2(1/3)) = **0.99**

Round #2 - Split on ‘Male’ No, Branch Yes

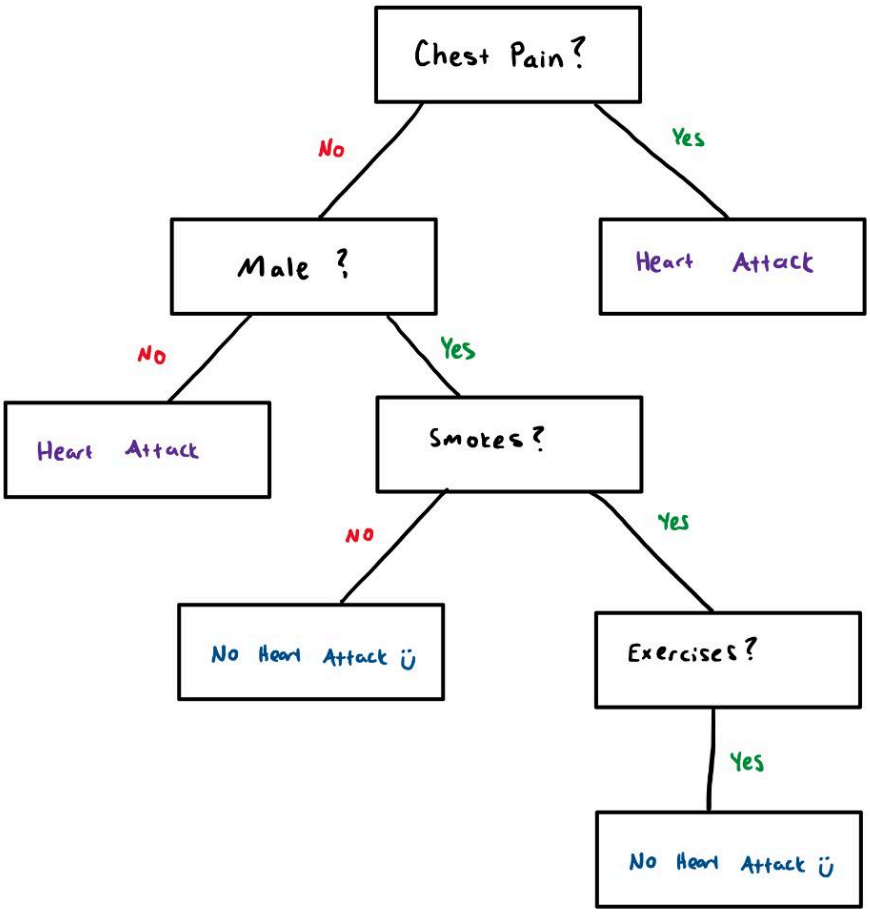
Split Round #3
Entropy: - (2/2) * log2(2/2) = 0

a. Smokes:
   Entropy “Yes” : -1/1 * log2(1/1) = 0
   Entropy “No” : -1/1 * log2(1/1) = 0
   Entropy: (1/2 * 0) + (1/2 * 0) = 0
   Gain: 0 - 0 = 0
   Gain Ratio: 0 / (-1/2 * log2(1/2) - 1/2 * log2(1/2)) = **0**

b. Exercises:

Entropy “Yes” : -2/2 * log2(2/2) = 0
Entropy “No” : 0
Entropy: (0) + (2/2 * 0) = 0
Gain: 0 - 0 = 0
Gain Ratio: 0 / (-2/2 * log2(2/2) - 0) = **0**

Round #3 - Split on ‘Smokes’ Yes, Branch No



B. Translate your decision tree into a collection of decision rules

ChestPain = Yes → Heart Attack
((ChestPain = No) ^ (Male = Yes)) → Heart Attack
((ChestPain = No) ^ (Male = No) ^ (Smokes = No)) → No Heart Attack
((ChestPain = No) ^ (Male = No) ^ (Smokes = Yes)) → No Heart Attack
((ChestPain = No) ^ (Male = No) ^ (Smokes = Yes) ^ (Exercises = Yes) → No Heart Attack

2. Classifier Playground https://www.ccom.ucsd.edu/~cdeotte/programs/classify.html

From the site

This page demonstrates basic classifiers: k nearest neighbors, decision tree, and linear classifiers. The ***linear classifiers*** *shown here are naive Bayes, logistic regression, and support vector machines.*

- *Naive Bayes is part of a larger family of Bayes classifiers which include linear discriminant analysis (LDA) and quadratic discriminant analysis (QDA).*
- *Logistic regression is part of a larger family called generalized linear models.*

Linear classifiers base their decision on a linear combination of the features. Any linear model can be converted into a more flexible model such as quadratic by engineering new features from the old. For example, given x_1 x_2, create $x_3 = x_1$^2, $x_4 = x_2$^2, $x_5 = x_1x_2$. Then you have a quadratic model. None-the-less, this is still a linear combination of features.

It can be shown that naive Bayes, logistic regression, and support vector machines are all similar mathematically. They all make their decision based on a linear combination of features. For example, given x_1 x_2, they predict y is blue if and only if $w_0+w_1x_1+w_2x_2 > 0$. They vary only in how they calculate the weights: w_0 ,w_1, w_2.

More advanced classifiers are created by combining multiple instances of these basic classifiers. Examples are neural networks, random forests, bagging, boosting, and stacking. Advanced classifiers create decision boundaries flexible enough to model any pattern.

The site comes preloaded with three examples. Your task is to apply the different classifiers and determine their behavior on the different data sets. In addition, you can add more data points to the examples by left-click adds a red point and right-click adding a blue point. See the directions on the site.

Go to the Classifier Playground site: https://www.ccom.ucsd.edu/~cdeotte/programs/classify.html
Read it carefully. Click on the Examples button. There are three examples available to you or you can make your own from scratch using left and right mouse clicks to place red and blue points in the data set. You can also modify the given data sets by adding points in the same way.

Task 1: To familiarize yourself with the data sets, run KNN classify on the test data sets with different values for k. For each data set run KNN for increasing values of k.

A. What do you observe for each of the three example data sets? For each data set what is the impact of increasing k?

   Example Data #1
   - k = 1 : Split the colors down the middle as well as separated the outliers on each side; classified the outliers (ex: blue dot in red classification) into their correct color group (blue instead of red)
   - k = 3 : Ignored the outliers in each dataset, focusing solely on the middle split
   - k = 5 : Continued to ignore the outliers in each dataset, focusing solely on the middle split (this time closer to the red points on the split)
   - Observations & Increasing Impact:

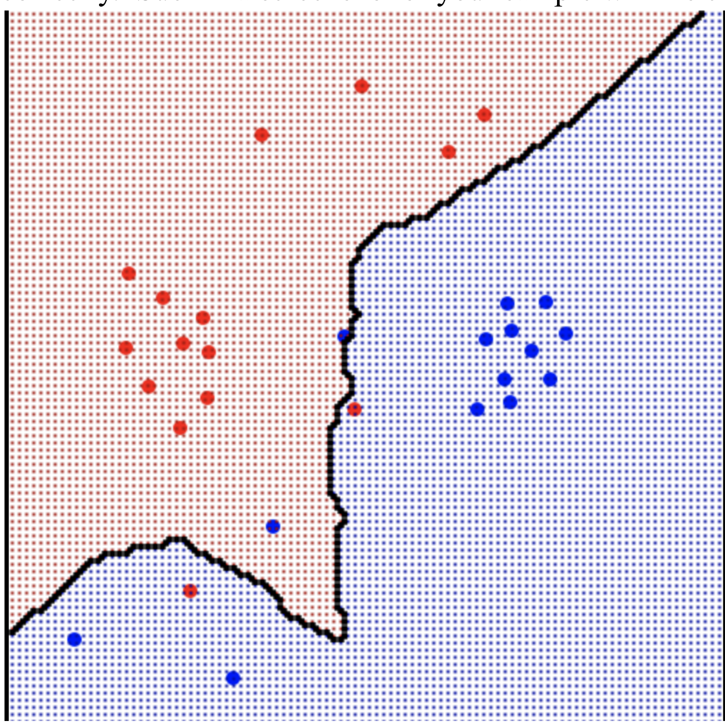
   Example Data #2
   - k = 1 : Split the colors accurately, drawing a circle around the red and having the surrounding / rest of the data be classified as blue

- k = 3 : Tightened in on the red points, making the classification area of red smaller
- k = 5 : Similar result to the k above; tightening red in another way
- Observations & Increasing Impact:
-

Example Data #3

- k = 1 : Split the points accurately into their color sections; curving like a x^3 graph
- k = 2 : Tightened the curve to be closer to the points that overlap vertically
- k = 3 : Incorrectly classified the points which overlap by tightening the curve to be a vertical line; treating the overlapping points as outliers
- Observations & Increasing Impact:

B. Develop an example that requires k to be at least 4 to get a classifier that classifies almost all of the data correctly. Submit a screenshot of your sample with its classification.



Task 2: Run Linear SVM classify on each example data set using different values for C.

A. What is the minimum value for C that does a good job of classification for each example data set?
B. Interpret these results in light of the description of C given in the Examples window when you click on the Examples button.

Example Data #1

- minimum value: c =5 because a line splits down the middle of the screen separating both sides, including the red points that lie in the middle of the two sets and ignoring the outliers
- interpretation: with a graph designated to overfitting / designed to have only a few outliers, it does a decent job of splitting down the middle just choosing to ignore the outliers

Example Data #2

- minimum value: c = 7 because a line splits the data as best it can since the data curves, only ignoring a few outliers (6 blue, 4 red) which are relevant even misclassification
- interpretation: with a graph designated to underfitting/ designed with points overlapping vertically, it does a bad job classifying the data since it is a vertically based line formula. there becomes a certain point where you just have to call it good enough

Example Data #3

- minimum value: c = 3.2 because it puts a line that exactly splits the data into their sections half and half, not having any outliers
- interpretation: with a graph designed similarly to y = x^3, I agree with the statement that a higher c value will prioritize separation as the vertical line split curves accordingly the higher the c value

Task 3: Compare Linear Logistic Regression, Naïve Bayes, Tree classify, Linear SVM classify.

A. For each of the three data sets, try to draw a conclusion and justify it. For example, do any of the methods seem particularly good or bad for a given type of data set.

Example Data #1 - Tree classification is the only out of the three that correctly classifies the outliers of the data

Example Data #2 - Tree classification is able to correctly classify each section, hugging close to the borderline points. Naive Bayes also does a decent job due to its quadratic curved behavior like the given graph / layout of the ponts.

Example Data #3 - Linear Logistic and Tree classification are able to accurately split the data evenly

B. For the decision trees for example 1 and example 2, which two nodes are best to prune. Interpret the results.

For the decision tree for Example #1, the best two nodes to prune are A. "N = 9: R = 8, B = 1" and B. "N = 1: R = 1, B = 10" . Removing these normalizes the split by ignoring the outliers, causing it to do a split directly down the middle like the other classification methods.

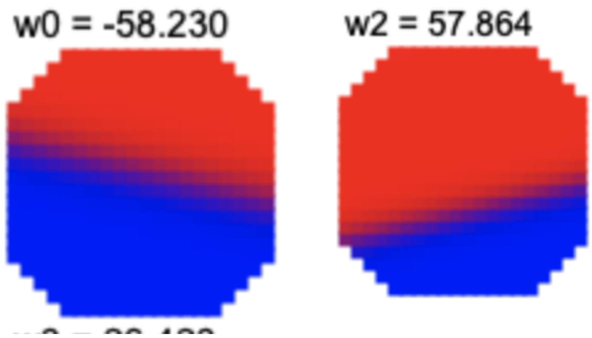
For the decision tree for Example #2 there are no two nodes to prune, the nodes available to prune will just cause the tree to incorrectly classify blue points as red (and vice versa) due to the points' curved shape.

Task 4: At the very bottom of the page, it reads ***Click here to play with a neural network***. Go to that link.

A. Work though **example 1** using values 1, 2, 3, 4 for H. This is just for fun. Notice that the Output decision is their sum represents the output from each of the nodes marked ***Hidden***

Will do! (I don't think part A is asking a specific question / needs us to provide an answer?)

B. For **example 2**, which of the Hidden nodes is largely responsible for the three line segments that separate the red and blue dots in the middle of the data set?

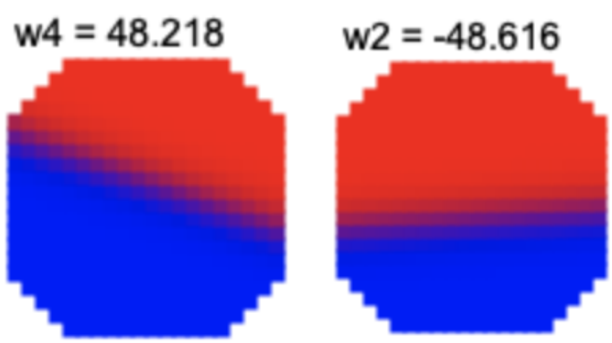


It seems that the w0 and w2 hidden nodes are largely responsible for the three line segments that separate the red and blue dots down the middle of the data. This is because the layout of the data is better split across the x axis and the second input is in charge of creating a horizontal boundary (representing the split of red and blue points that lie on the borderline of red and blue points).

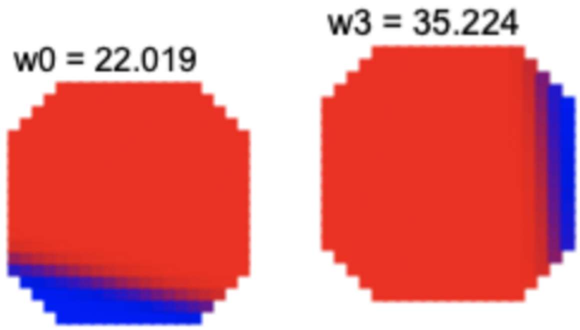
This example was done with H = 5

C. For **example 3**, which of the Hidden nodes is largely responsible for the triangle enclosing the blue points and which Hidden nodes are responsible for determining the rectangle that separates the red dots from the two blue dot areas?

It seems that the w4 and w2 are the hidden nodes largely responsible for the triangle enclosing the blue dots. This is because the layout of the data is better split across the x axis where the blue lies within the center and the red lies within the outside.



It seems that the w0 and w3 are the hidden nodes largely responsible for determining the rectangle that separates the red dots from the two blue dot areas. This is because the layout of the data is better split across the x axis where the red lies within the center.



This example was done with H = 5

Submit your writeup to Canvas with your response to each task clearly labeled.