Question 1 (Observations): I see that a lot of algorithms used in data mining are popular because they are simple and efficient, yet given that there are many available implementations of both simple and complex algorithms, why aren't the complex ones more popular since there might not be any implementation overhead on the part of the user? The AdaBoost Algorithm was the most interesting to me. Reading about the facial detection work brought up a paper by Viola and Jones [1], and they used a technique called cascading, which I had no idea about, so I looked it up, and it is a technique which divides the window into smaller (24x24) windows, and if no facial features are found within a window, it is thrown out, hence the space of windows "cascades" down to a few windows that have facial features. Additionally, the AdaBoost algorithm supposedly does not exhibit much overfitting, yet the reason for this is not certain, and I wonder if it has something to do with the fact that weak learners are used. Would the boosting of strong learners result in overfitting, or would the result just be extremely high time complexity? How does the SVM algorithm decide to project into higher dimensions? Supposedly, the SVM hyperplane can be expressed in infinitely many dimensions, but is the possible number of upwards projections dependent on the length of inputs? I really appreciate how the authors took the time to include extensions of the base algorithms provided in the article. It shows how a simple idea can be generalized through both data and algorithm manipulation. EM was the most confusing to me, especially with all the different mathematical symbols. I didn't really understand the algorithm. I found the PageRank model to be pretty interesting sue to it's visible real-world success (Google). I found a python implementation on GitHub [2], which helped me understand how the ranking works within the algorithm.

1. https://towardsdatascience.com/face-detection-with-haar-cascade-727f68dafd08#:~:text=So%20what%20is%20Haar%20Cascade,Simple%20Features%E2%80%9D%20published%20in%202001
2. https://github.com/timothyasp/PageRank

2.1: Why is the Naive Bayes algorithm called **naive**? What is the meaning of the word Bayes?

- It assumes that all features of the data set are independent of each other.

- Bayes refers to a statistician named Bayes who is credited with Bayes' Theorem, which enables the use of probability to predict classifications.

2.2: List the algorithm(s) discussed in this article that generate decision trees. Which of these algorithms was published/released first?

- CART (1984) and C4.5 (1993) both produce decision trees.

- CART was released first in 1984.

2.3: Briefly explain the meaning of k in the kNN algorithm. Could k be set to 1? What would be the prediction for the 7NN algorithm with the majority vote where the 7 neighbors belong to the following classes: class1, class1, class3, class2, class1, class2, class3?

- K is the number of nearest neighbors that should be used to classify new data.

-Yes, k could be 1, but using only 1 neighbor leads to a model that follows the training data, which results in overfitting, and so testing any new data yields higher error rates.

- Since 3 of 7 of the closest neighbors are class1, then class1 is the majority, so the model would classify the new datum as part of class1.

2.4: Which five of the nine algorithms would be the most suitable choices to be used as the “weak learners” by the AdaBoost algorithm?

- the 5 algorithms that can be used as weak learners are kNN, C4.5, Naïve Bayes, CART, and SVM