

Tyler Angert
CS 378 HW2 Writeup

Decision Tree

Accuracy: 98.002%

Best splits (by attribute): 5 -> 19 -> 16 -> 8 -> 7 -> 4

```
----> Splitting on: 5
  Leaf a prediction: {'e': 1.0}
  Leaf c prediction: {'p': 1.0}
  Leaf f prediction: {'p': 1.0}
  Leaf m prediction: {'p': 1.0}
  Leaf l prediction: {'e': 1.0}
----> Splitting on: 19
  Leaf b prediction: {'e': 1.0}
  Leaf h prediction: {'e': 1.0}
  Leaf k prediction: {'e': 1.0}
  Leaf o prediction: {'e': 1.0}
  Leaf n prediction: {'e': 1.0}
  Leaf r prediction: {'p': 1.0}
----> Splitting on: 16
  Leaf y prediction: {'p': 1.0}
----> Splitting on: 8
  Leaf b prediction: {'e': 1.0}
----> Splitting on: 7
  Leaf c prediction: {'p': 1.0}
----> Splitting on: 4
  Leaf t prediction: {'p': 1.0}
  Leaf f prediction: {'e': 1.0}
  Leaf y prediction: {'e': 1.0}
  Leaf p prediction: {'p': 1.0}
  Leaf s prediction: {'p': 1.0}
  Leaf y prediction: {'p': 1.0}

Accuracy: 98.002853067%
```

The decision tree was pretty complicated to implement from scratch. I am a pretty visual learner and enjoy OOP a lot because of this, so I created the tree with LeafNode, DecisionNode, and Split objects to conceptually make its construction easier. I am not sure if this is considered slow or not, but my version takes about 4.5 seconds to build the tree. I predict that I probably did a bunch of superfluous calculations but I couldn't catch them in my code as of now. However this was really interesting to implement and I definitely understand

the process a lot better, especially the significance of why we use entropy and information gain.

Naive Bayes:

Accuracy: 98.857%

This algorithm was significantly easier to make and test compared to the decision tree. At first, I was able to write this all as one script in my main method without really having to think too much about data structures and optimization. Then, I modularized it but still didn't need an external objects other than dictionaries and lists. It is very fast and I enjoy how statistically clear the process is, given that everything is just based on conditional probability.

Implementing both of these algorithms was a huge lesson in understanding code modularity and really thinking about the structure of algorithms.