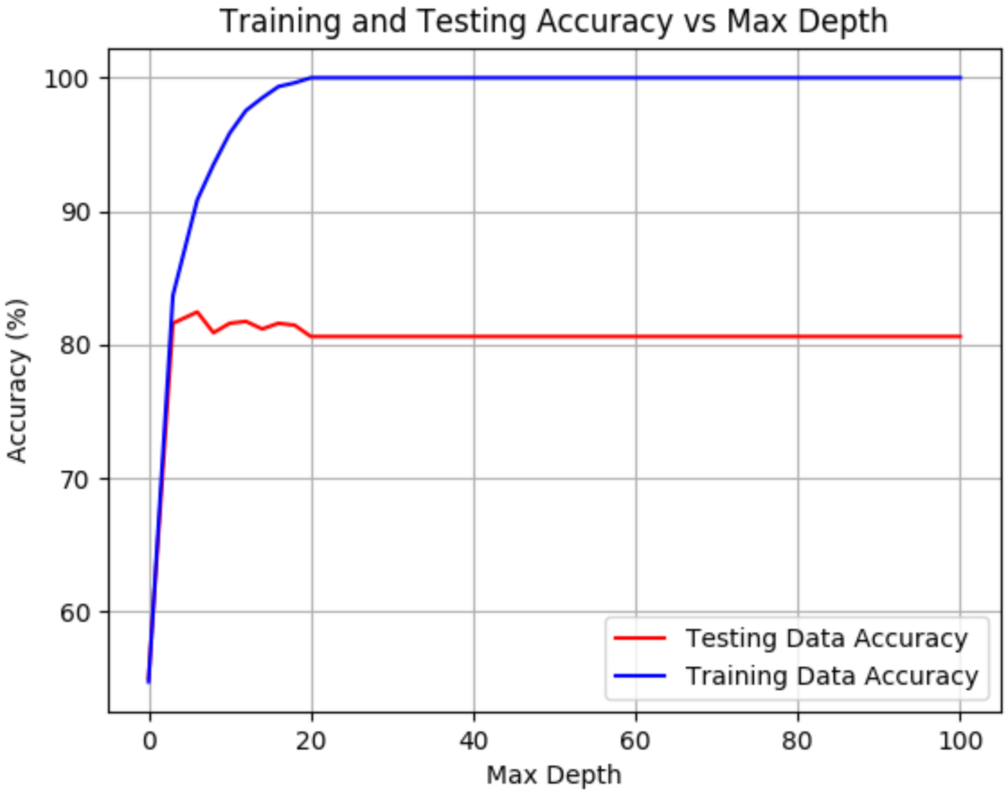
Decision Tree Learning

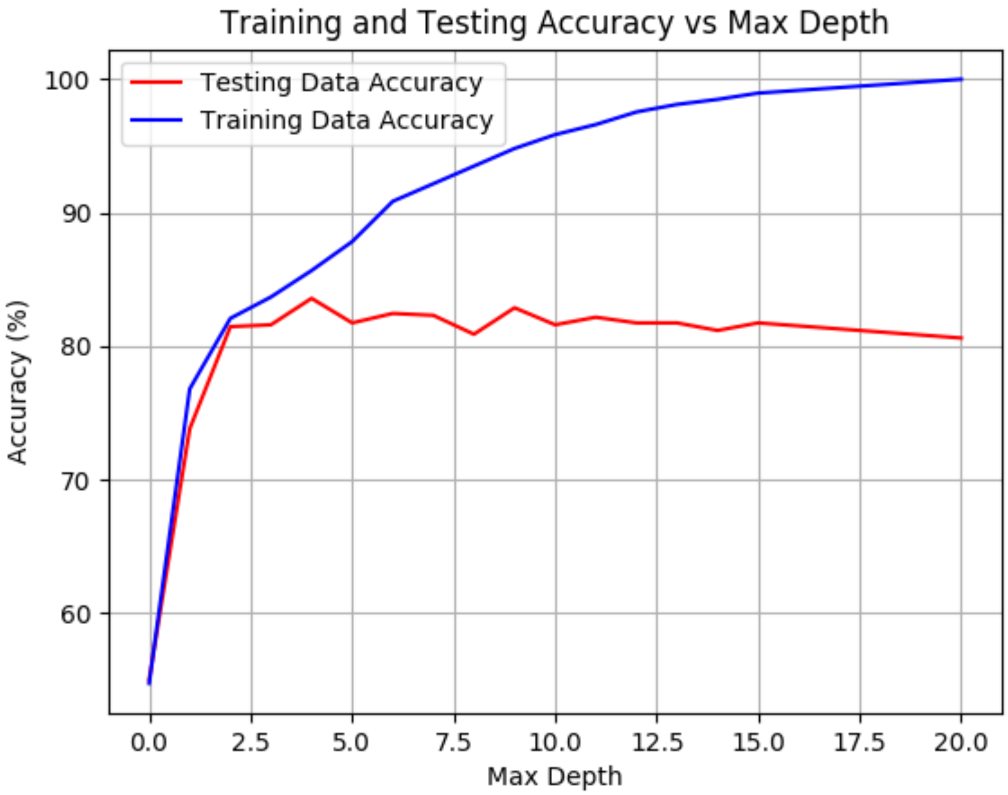
1. A printout of the code is attached. Testing and training accuracies of the implemented algorithm are in agreement with the accuracies of DTL algorithms from the scikit-learn library's , Example of an output:

```
Prediction:
Testing Data Accuracy:  80.6223479491
Training Data Accuracy:  100.0
-----------------------------------------

Prediction with scikit-learn
Testing Data Accuracy:  81.7538896747
Training Data Accuracy:  100.0
```

2. Below are two graphs showing training and testing accuracy as the maximum depth increases. The graphs have different resolution.





3. Overfitting occurs at the depth when the test set starts to decrease or remain steady, while the training set continues to grow. According to the graphs above, overfitting occurs after the maximum depth of 4. After the maximum depth of 4 the testing accuracy starts to remain constant (and then decrease), while training accuracy keeps increasing to 100%.

4. A printout of the decision tree is attached (decision_tree.txt).

5. According to the decision_tree.txt, the word features that were selected by the decision tree early are:
   Level 1: 484 (writes)
   Level 2: 211 (god) and (graphics)
   Level 3: 152 (that), 183 (use) and 2108 (image),
   Level 4: 73 (bible), 187 (wrote), 0 (archive), 152 (that)

These word features reduce entropy (uncertainty) the most, whereby making the decision tree smaller (and more effective). For example, the subtrees of *god-present* and *graphics-present* are very small, meaning that if these words are present in the document, the class can be determined more quickly. Looking at the word features at the first 4 levels, we can see that the words *god*, *graphics*, *image*, *bible* and *archive* can be easily associated with one of the classes (*atheism* and *graphics*). Other words, such as *writes*, *that*, *use*, and *wrote* cannot be easily associated with one of the classes, but based on the results they still reduce the entropy the most (at places where they are used).

Naïve Bayes Model

1. 1. A printout of the code is attached. Testing and training accuracies of the implemented algorithm are exactly equal to the accuracies of the Bernoulli Naïve Bayes algorithms from the scikit-learn library's

2. A printout listing 10 most discriminative word features:

```
   Most Discriminative Word Features:
        3142 (graphics)
        2 (atheism)
        16 (religion)
        425 (moral)
        570 (keith)
        767 (evidence)
        562 (atheists)
        211 (god)
        73 (bible)
        271 (christian)
```

   These are good discriminative features between two classes, because they can be easily associated with one of two classes. Interestingly, only one word feature (*graphics*) is related to the "graphics" class. Nine out of 10 most discriminative features are related to the "atheism" class.

3. Printout of training and testing accuracies:

```
      Prediction:
      Testing Data Accuracy:  88.9674681754
      Training Data Accuracy:  92.8369462771
      -----------------------------------------

      Prediction with scikit-learn
      Testing Data Accuracy:  88.9674681754
      Training Data Accuracy:  92.8369462771
```

4. The assumption that word features are independent given the class is only partially correct.

   Each of the classes (*atheism* and *graphics*) has its own common words. The words specific to a certain topic are more likely to be used together in the same document. For example, if there is a word "*god*" in a document, it is more likely that the words "*bible*", "*evidence*", "*religion*", "*atheism*", "*Christian*" are also in that document. However, if we are given the class (*atheism* or *graphics*) these words (attributes) become independent of each other.

At the same time, there are some cases, for which, even when the class is given, there are dependences between words. Some of the words are used together as one expression. For example, the words “most” and “of” are frequently used as an expression “most of”. The words “holy” and “bible” are frequently used as an expression “holy bible”.

5. If we want the Bayes Model to take into account dependences between words, we can increase conditional probability of the dependent words. For example if the word “most” occurs in the text, the probability (theta) of the word “of” to occur in the text should be increased. Similarly, if the word “holy” occurs in the text, the probability (theta) of the word “bible” to occur in the text should be increased

6. According to the results, Naïve Bayes model performs better on the testing set that the Decision tree, even though, as we know, the naïve Bayes classifier makes a strong (naïve) independence assumption. In the case of this specific problem, most of the highly discriminative features are “relatively” conditionally independent, given the class (label). Therefore, the naïve assumption doesn’t have a very strong effect on the performance of the algorithm. On the other hand, decision trees tend to overfit the training data more than other techniques.