1. Implemented in the attached code file

2. We uses Naive Bayes Classifiers for spam email filtering based on its Naive Bayes Theory:

a. Naive Bayes Theory

- Naive Bayes Theory assumes that consider having a set of n distinct words $\{x_1, x_2, ..., x_n\}$, We are interested in computing the probability in which an email is spam given that it contains the above set of words:
  $P(S \mid x_1, x_2, ..., x_n)$
- With Bayes' Theorem, the above probability is equivalent to:
  $\frac{P(x_1, ..., x_n \mid S)\, P(S)}{P(x_1, ..., x_n)}$, where
  $P(x_1, ..., x_n) = P(x_1, ..., x_n \mid S)\, P(S) + P(x_1, ... x_n \mid H)\, P(H)$
  ** S is the event of spam email while H is the event of non-spam (ham) email **
- Ignoring the denominator, we have:
  $P(x_1, ..., x_n \mid S)\, P(S) = P(x_1, ..., x_n, S)$
  Then, use the chain rule:

  $$P(x_1, ..., x_n, S) = P(x_1 \mid x_2, ..., x_n, S)\, P(x_2 \mid x_3, ..., x_n, S) ... P(x_{n-1} \mid x_n, S)\, P(x_n \mid S)\, P(S)$$

- Since the above formula contains so many conditions that it's generally not possible to calculate its probability accurately. Therefore, Naive Bayes makes the assumption that the words in an email are conditionally independent of each other, given that we already know the email is whether spam or ham. In other words, it naively assumes independence where it might not actually exists. With the assumption, we can rewrite chain rule as followed:

  $$P(x_1, ..., x_n, S) \approx P(x_1 \mid S)\, P(x_2 \mid S) ... P(x_{n-1} \mid S)\, P(x_n \mid S)\, P(S)$$
  $$= P(S) \prod_{i=1}^{n} P(x_i \mid S)$$

  By the similar argument for ham emails, we have:

  $$P(x_1, ..., x_n, H) \approx P(H) \prod_{i=1}^{n} P(x_i \mid H)$$

  Combine them altogether:

$$P(S \mid x_1, x_2, ..., x_n) \approx \frac{P(S)\prod_{i=1}^{n} P(x_i \mid S)}{P(S)\prod_{i=1}^{n} P(x_i \mid S) + P(H)\prod_{i=1}^{n} P(x_i \mid H)} \quad \textbf{(*)}$$

The above equation will be the estimate of the probability of an email being spam when it contains some words in the above distinct set as long as we have the prior probability of an email being spam, and the probabilities of seeing a particular word in either a spam or ham email

b. From the theory, we now can construct a general and powerful spam filter using Naive Bayes Classifiers:
   i. Data Collecting: Collect a dataset with labeled examples of both spam and non-spam (ham) emails
   ii. Preprocessing: Go over the labelled spam email and for a certain word *w* in the entire training set, count the number of spam emails contain *w*
   iii. Training and smoothing the dataset by computing:

   $$P\ (w \mid S)\ =\ \frac{|spam\ emails\ containing\ w| + 1}{|spam\ emails| + 2}$$

   Then, do the same for ham emails
   Note: Adding 1 and 2 respectively for numerator and denominator is the technique used to handle the problem of zero probabilities in the Naive Bayes classifier. In the context of spam filtering, zero probabilities can occur when a word appears in the test data but not in the training data. Without smoothing, the Naive Bayes classifier would assign a probability of zero to such events, which can cause issues when calculating the posterior probabilities

   iv. Classifying: Compute the probability of spam and ham emails in the dataset

   $$P\ (S)\ =\ \frac{|spam\ emails|}{|spam\ emails| + |ham\ emails|}$$

   $$P\ (H)\ =\ \frac{|ham\ \ emails|}{|spam\ emails| + |ham\ emails|}$$

   v. Evaluating: Given a set of unlabelled test emails, iterate over each:
      - Create a set $\{x_1, x_2, ..., x_n\}$ of the distinct words in the email. Ignore unseen words
      - Compute the above (*) formula
      - If $P(S \mid x_1, x_2, ..., x_n) > 0.5$, output "spam", else output "ham"

4.

- The use of Bayes' theorem is appropriate to calculate the probability of having the disease given positive test result. The formula for Bayes' Theorem is:

$P(A\mid B) = \frac{P(B\mid A)\,P(A)}{P(B)}$, where

- P(A|B) is the probability of having the disease given a positive test result.
- P(B|A) is the probability of a positive test result given having the disease.
- P(A) is the prior probability of having the disease.
- P(B) is the probability of a positive test result.

- The above terms have the following information:
  - P (B | A) = 0.99 (accuracy of the test)
  - P(A) = $\frac{1}{10000}$ (prevalence of the disease)
  - P(B) = $P(B\mid A)\,P(A) + P(B\mid \neg A)\,P(\neg A)$, in which:
    - $P(B\mid \neg A) = 1 - 0.99 = 0.01$
    - $P(\neg A) = 1 - \frac{1}{10000}$

Hence, P(A | B) = $\frac{0.99 \cdot \frac{1}{10000}}{0.99 \cdot \frac{1}{10000} + 0.01 \cdot (1 - \frac{1}{10000})} \approx 0.0098$

5.

- The algorithm has been used to schedule observations for the Hubble Space Telescope, reducing the time taken to schedule a week of observations from 3 weeks to around 10 minutes is **Min-Conflicts Heuristic Algorithms**
- A well-known CSP solver we described in class is **Google OR-Tools**