1A: For model assessment I decided to split the data into training and test before I passed it into my function. I also decided to pass the whichever model I wanted to use as a parameter. For example I could've passed in sklearn's Logistic Regression model as a parameter. For the assessment part I trained the model using the xTrain data and the yTrain data. Then I predict the model using the test info and using the accuracy found the number of mistakes the model made.
1B - E: See code q1.py

2 A: See Code
2 B: For each of the three datasets I ran large numbers for the epoch so I can find the epoch where the algorithm terminate due to zero mistakes. I determined the optimal epoch by creating a script that ran through a range of epochs and kept track of all the number of mistakes on the test data

Binary Word Map:
Terminates at epoch 10
Num of Mistakes: 40
Estimated Predictive Error: 40/1801 : 2.2%
Optimal Epoch: 6
15 Positive Words:
[monei,simpl,sight,charg,mai,you,deathtospam,temov,will,hour,click,market,guarante,profession]
15 Negative Words:
[set,wrote,on,but,seem,author,data,re,url,prefer,try,hardwar,copyright,data,were]

Count Word Map:
Terminates at epoch 160
Num of mistakes: 49
Estimated Predictive Error: 49/1801 : 2.7%
Optimal Epoch: 159
15 Positive Words:
[numberenumb,numberdnumb,order,monei,china,remov,face,numbera,model,size,island,numberc,will,ve,our]
15 Negative Words:
[numberp,but,date,re,file,subscript,wrote,write,url,shop,she,canon,bui,strip,subscrib]

TFIDF Word Map:
Terminates at epoch 105
Num of mistakes: 47
Estimated Predictive Error: 47/1801: 2.6%
Optimal Epoch: 44
15 Positive Words:
[our,method,bodi,tv,island,send,flash,deathtospam,china,guarante,clcik,remov,monei,size,below]

15 Negative Words:
[date,wrote,cnet,url,network,but,if,re,messag,set,said,spam,start,file,about]

3A-3B: See Code end of q1.py
When implementing Naive Bayes in sklearn the algorithm ranges in the number of mistakes similar to my own implementation of Naive Bayes but the variation of the number of mistakes is greater than my own implementation. For example in one run of the sklearn Naive Bayes the number of mistakes for Binary, Count, TDIF data was 40,54,87 respectively and in another run the mistakes jump to 50,56,95 respectively. The sklearn Naive Bayes outperforms my model sometimes and sometimes underperforms compared to my model. Sklearn Logistic Regression performs in a similar manner to the Naive Bayes but the results most of the times outperforms my model. For the Binary and Count data the number of mistakes usually ranges between 25 - 35 for both datasets but for the TDIF datasets the mistakes are still pretty high at around 80 - 90 mistakes