1. Explanation of the Perceptron algorithm for the binary classification case

Pseudo code of training algorithm:

```
01:  W[1], W[2], W[3],…W[d] = 0
02:  bias = 0
03:  for iter = 1 … MaxIter do
04:      for i = 1…d do
05:          activation += W[i] * X[i]
06:      end for
07:      activation += bias
08:      if y * activation <=0 then
09:          for i = 1…d do
10:              W[i] += y * X[i]
11:          end for
12:          bias += y
13:      end if
14:  end for
15:  return W[1], W[2],…W[d], bias
```

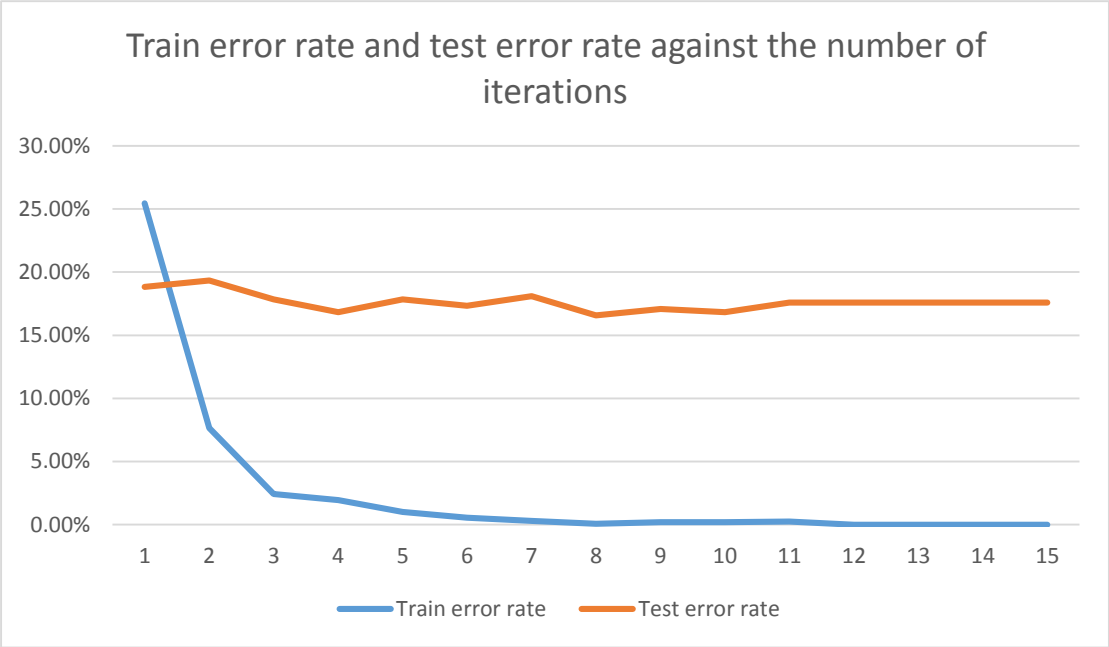
In line 1 and 2, we initialize the bias and the list of weights by 0. Each element in the weight list represents the weight of one feature. Each input (vector Xi) is a review. An element in Xi equals 0 means the feature represented by that element does not exist in the review, while equals 1 means it does. From line 4 to 7, we simply multiply each input (feature) by a weight and check whether this weighted sum (activation) is greater than a threshold (line 7). In this binary classification case, y is the pre-judged value of a review (1 means positive, -1 means negative). If the sign of y is different from the sign of activation (line 8: if y * activation <=0), then it means the computer makes a wrong classification. Learn only if we make a mistake when we classify using the current weight vector, so we modify the value of bias and weights(line 9-13).

Pseudo code of testing algorithm:

```
01:  for iter = 1 … MaxIter do
02:      for i = 1…d do
03:          activation += W[i] * X[i]
04:      end for
05:      activation += bias
06:      if y * activation <=0 then
07:          errors +=1
08:      end if
09:  return errors
```

In testing algorithm, we only check if computer make a right or wrong classification, without changing its weights and bias.

4. Plot the train error rate and test error rate against the number of iterations



In my experiment, the train error rate drops down rapidly (from 25.44% to 2.44%) in the first 3 iteration, then keeps decreasing and reach 0 at 12th iteration. Before 12th iteration, the test error rate fluctuates between 17% and 20%. As the train error rate reaches 0, it levels off at 17.59%. Therefore, 12 would be the best number of iteration of training data.