The University Of Sheffield.

Assignment 1: Sentiment analysis

Name: Daotan Liu

Student ID: 200206596

Step 2.2

The formula for calculating the Performance Matrix is as follows:

$$Accuracy = \frac{correctly\, classfied\, texts}{texts}$$

$$Precision\, Pos = \frac{texts\, correctly\, classified\, as\, positive}{texts\, classified\, as\, positive}$$

$$Recall\, Pos = \frac{texts\, correctly\, classified\, as\, positive}{positive\, texts}$$

$$F - measure\, Pos = \frac{2 * Precision\, Pos * Recall\, Pos}{Precision\, Pos + Recall\, Poss}$$

The function is as follows:

```
def performances(correct,total,correctpos,totalpospred,totalpos,correctneg,totalnegpred,totalneg,dataName):
    accuracy = correct/total
    precision_pos = correctpos/totalpospred
    recall_pos = correctpos/totalpos
    precision_neg = correctneg/totalnegpred
    recall_neg = correctneg/totalneg
    F1_pos = (2*precision_pos*recall_pos)/(precision_pos+recall_pos)
    F1_neg =(2*precision_neg * recall_neg)/(precision_neg + recall_neg)

    print('----------------------------------------')
    print(' The performance of NaiveBayes on the %s'%dataName)
    print('(1) Accuracy : %0.2f'%accuracy)
    print('(2) Precision(positive): %0.2f, Recall(positive): %0.2f'%(precision_pos,recall_pos))
    print('(3) Precision(negative): %0.2f, Recall(negative): %0.2f'%(precision_neg,recall_neg))
    print('(4) F1 score(positive): %0.2f, F1 score(negative): %0.2f'%(F1_pos,F1_neg))
```

The variable:

correct = correctly classified texts

total = texts

correctpos = texts correctly classified as positive

totalpos = texts classified as positive

correctpos = texts correctly classified as positive

totalpos = positive texts

Negative Performance Matrix is similar to the above.

Step 2.3

```
 The performance of NaiveBayes on the Films  (Test Data, Naive Bayes)
(1) Accuracy : 0.79
(2) Precision(positive): 0.79, Recall(positive): 0.77
(3) Precision(negative): 0.78, Recall(negative): 0.80
(4) F1 score(positive): 0.78, F1 score(negative): 0.79
```

Step 3.1

```
 The performance of NaiveBayes on the Films (Train Data, Naive Bayes)
(1) Accuracy : 0.89
(2) Precision(positive): 0.90, Recall(positive): 0.88
(3) Precision(negative): 0.88, Recall(negative): 0.90
(4) F1 score(positive): 0.89, F1 score(negative): 0.89
----------------------------------------
 The performance of NaiveBayes on the Nokia   (All Data,  Naive Bayes)
(1) Accuracy : 0.61
(2) Precision(positive): 0.78, Recall(positive): 0.61
(3) Precision(negative): 0.40, Recall(negative): 0.60
(4) F1 score(positive): 0.68, F1 score(negative): 0.48
```

Step 3.2

The code above has run Naive Bayes on three different datasets and get three results. By comparing the classified results of test data and training data. It is easy to find the result (Accuracy, Precision and F1) of the train data is significantly better than that on the test data. This is mainly because in the training of Bayesian classifier, all parameters (Prior, Likelihood and Normalisation) are calculated from the train dataset. Therefore, when it is used for classification, the result of the training data is better than that of the test data. However, the result of the test set is closer to the real performance of the classifier.

The reason why the performance of using Naive Bayes classifier on Nokia Product Reviews is lower than that on Test Data. It is mainly because the pWord, pWordPos and pWordNeg are based on reviews of movies. Since features in film reviews vary, words, sentiments and opinions are different. As a result, the classification effect of the classifier trained by film Review data on Nokia Product Reviews is not good.Another reason is that data of Nokia Product Reviews may be small and unbalanced.

Step 4.1

The top 50 negative:

```
NEGATIVE:
['stupid', 'badly', 'mediocre', 'routine', 'generic', 'unfunny', 'lame', 'boring', 'poorly',
'mindless', 'pointless', 'stale', 'annoying', 'shoot', 'tiresome', 'bore', 'unless', 'meandering',
'offensive', 'dreary', 'disguise', 'junk', 'inept', 'animal', 'amateurish', 'horrible', 'banal',
'harvard', 'wasted', 'lifeless', 'uninspired', 'dull', 'supposed', 'stealing', 'apparently', 'chan',
'pathetic', 'product', 'seagal', 'lousy', 'pinocchio', 'thin', 'ill', 'total', 'waste', 'cable',
'pacing', 'missed', 'pile', 'incoherent']
```

The top 50 positive:

```
POSITIVE:
['flaws', 'frailty', 'gradually', 'richly', 'subversive', 'unexpected', 'flawed', 'warm', 'powerful',
'aspects', 'heartbreaking', 'timely', 'delightful', 'assured', 'iranian', 'vivid', 'record',
"world's", 'captures', 'martha', 'detailed', 'tour', 'realistic', 'captivating', 'lively',
'jealousy', 'nicely', 'heartwarming', 'wry', 'touching', 'tender', 'playful', 'polished', 'respect',
'vividly', 'chilling', 'wonderfully', 'provides', 'gem', 'quietly', 'thoughtful', 'wonderful',
'extraordinary', 'refreshingly', 'mesmerizing', 'affecting', 'riveting', 'refreshing', 'inventive',
'engrossing']
```

Step 4.2

The code for this part is embedded in the main script.

```
head_num = 0
tail_num = 0
for word in head:
    if word in sentimentDictionary.keys():
        head_num +=1

for word in tail:
    if word in sentimentDictionary.keys():
        tail_num +=1

print('There are %d most useful negative words and %d most
      useful positive words in the sentiment dictionary.'%(head_num,tail_num))
```

There are 27 most useful negative words and 30 most useful positive words in the sentiment dictionary.

Since there are top 50 negative words and top 50 positive words, only 54% of the Top 50 negative words and 60% of negative words are in the sentiment dictionary. Hence, these selected words are not good sentiment terms.

Step 5.1

The code has been built in testDictionary(). The performance of rule-based system:

```
----------------------------------------
 The performance on the Films (Train Data, Rule-Based)
(1) Accuracy : 0.54
(2) Precision(positive): 0.54, Recall(positive): 0.60
(3) Precision(negative): 0.55, Recall(negative): 0.49
(4) F1 score(positive): 0.57, F1 score(negative): 0.52
----------------------------------------
 The performance on the Films  (Test Data, Rule-Based)
(1) Accuracy : 0.55
(2) Precision(positive): 0.53, Recall(positive): 0.59
(3) Precision(negative): 0.56, Recall(negative): 0.50
(4) F1 score(positive): 0.56, F1 score(negative): 0.53
----------------------------------------
 The performance on the Nokia   (All Data, Rule-Based)
(1) Accuracy : 0.68
(2) Precision(positive): 0.79, Recall(positive): 0.73
(3) Precision(negative): 0.47, Recall(negative): 0.56
(4) F1 score(positive): 0.76, F1 score(negative): 0.51
```

Step 5.2

Through comparison, it can be found that the performance of using rule-based classifier is not as good as that of Naive Bayes. In the rule-based classifier, it simply adds up the number of negative and positive words and without context, in many cases, this operation is not accurate enough or even results in the opposite classification. In addition, the rule-based classifier in this classifier does not adopt the gradable system. The value of all the words in the dictionary is either 1 or -1, which cannot capture the opinion holder's attitude accurately. Moreover, the classifier Requires a lexicon of emotional words, which should be fairly comprehensive, covering new words. However, the advantage of a rule-based classifier is that it requires no training and is highly efficient. Because the rule-based classifier does not need training, its performance of Train data and Test data are similar. The performance on Nokia review is higher than that on Films, indicating that sentiment Dictionary's collection of words related to film review is not comprehensive as regards the collection of words related to phones.

Step 5.3

The code is named ***improved_testDictionary()*** in the **Sentiment.py**

The result of improved rule-based system:

```
----------------------------------------
 The performance of NaiveBayes on the Films (Train Data, Rule-Based)
(1) Accuracy : 0.54
(2) Precision(positive): 0.54, Recall(positive): 0.63
(3) Precision(negative): 0.55, Recall(negative): 0.45
(4) F1 score(positive): 0.58, F1 score(negative): 0.49
----------------------------------------
 The performance of NaiveBayes on the Films  (Test Data, Rule-Based)
(1) Accuracy : 0.55
(2) Precision(positive): 0.53, Recall(positive): 0.63
(3) Precision(negative): 0.57, Recall(negative): 0.47
(4) F1 score(positive): 0.57, F1 score(negative): 0.51
----------------------------------------
 The performance of NaiveBayes on the Nokia   (All Data, Rule-Based)
(1) Accuracy : 0.69
(2) Precision(positive): 0.79, Recall(positive): 0.77
(3) Precision(negative): 0.49, Recall(negative): 0.51
(4) F1 score(positive): 0.78, F1 score(negative): 0.50
```

The improved code introduces weight and adopts five rules: Negation rule, Intensifier rule, diminisher rule, Exclamation rule and Capitaization rule. Among the rules, the Negation rule reverses the emotion of the word. Intensifier rule, Exclamation rule and Capitaization rule strengthen the sentiment and the diminisher rule diminishes the sentiment.

The comparison of the prediction results of the three data sets by using two functions shows that the accuracy of the prediction of the improved function is improved on the whole. The positive sentences have a higher recall rate, while the negative ones have a lower recall rate, and the values of the three F1(positive) have been improved to varying degrees. This means that the improved algorithm improves the accuracy of Positive words, which means that it lowers the threshold for judging a sentence as positive, which means that negative loses some of its accuracy.

Step 6.3

The function prints out a very large number of errors, which can be divided into two categories:

The first one is the form of *ERROR (pos classed as neg 0.24)* which means the sentence is positive, but it was classified as negative. The second one is the form of *ERROR (neg classed as pos 0.63)* which means the sentence is negative, but it was classified as positive.

Step 6.4

The main reason for these errors is that the function simply uses the most basic naive Bayes, and in practice the naive Bayes assumption is almost never true. Using naive Bayes two without considering the context, this algorithm is too simple. In real-world contexts, opinion holders occasionally use irony, for example: *ERROR (neg classed as pos 0.94): the armenian genocide deserves a more engaged and honest treatment.* In this sentence, the words "*a more engaged and honest*" seem to be positive, but the overall meaning of the sentence reveals the film is not enough engaged and honest on armenian genocide. Some fail to take into account the negative words, which tend to make the sentence mean the opposite of what it means, for example: *"ERROR (pos classed as neg 0.12): may take its sweet time to get wherever it's going, but if you have the patience for it, you won't feel like it's wasted yours."* In this sentence, if the function only focusses on the word 'waste', it will be easy to judge it as negative, but combined with 'won't', it is positive sentiment. The third case is because the sentence has a twist, for example: *"ERROR (neg classed as pos 0.99): though it was made with careful attention to detail and is well-acted by james spader and maggie gyllenhaal, i felt disrespected."* In this sentence, the positive vocabulary is redundant with the negative vocabulary, but because this sentence is a transition sentence, it generally expresses a negative attitude. In addition, a few of the comments are not in English, so misclassification occur.