Programming Assignment 1

Six individual feature selection methods are explored, including: **all**, **threshold**, **length**, **count ratio**, **remove**, and the combinations of each. A description for each method, their results, and additional comments are shown below.

ALL

Description: use the entire unique vocabulary of the training set as features. This will serve as a baseline to improve upon for other feature selection approaches.

```
Finished Training: 0.565 s
----------------------------------------
Classes: neg: 0, pos: 1

Confusion Matrix:
    0  1
  -------
0 | 78 23
1 | 27 72

Metrics:
            0     1     mean
          --------------------
Precision | 0.743 0.758 0.750
Recall    | 0.772 0.727 0.750
F1        | 0.757 0.742 0.750

Overall Accuracy: 75.000 %
----------------------------------------
Finished Training + Evaluation: 0.685 s
```

MANUAL

Description: a total of 39 features were predetermined and used as features. This includes the words “friend”, “never”, “recommend”, “no”, “yes”, “!”, “sad”, “happy”, and others that came to mind. These were personally chosen as words that I believe could potentially have influence over a positive or negative review.

```
Finished Training: 0.554 s
----------------------------------------
Classes: neg: 0, pos: 1

Confusion Matrix:
    0  1
  -------
0 | 96 5
1 | 87 12
```

```
Metrics:
            0     1     mean
          ------------------
Precision | 0.525 0.706 0.615
Recall    | 0.950 0.121 0.536
F1        | 0.676 0.207 0.573

Overall Accuracy: 54.000 %
---------------------------------------
Finished Training + Evaluation: 0.629 s
```

Comments: the manual selection of features performs poorly with an accuracy barely better than a coin flip. However, the negative classes in the test set are classified very accurately a recall around one. In fact, the degradation of performance comes from the the ability to correctly classify the positive samples. Removing the "!" as a feature improves performance for positive but lowers the performance for negative. Therefore, I speculate that more features surrounding positive expressions could be useful.

THRESHOLD

Description: using the entire vocabulary from all classes, the maximum and mean count of words are collected. It turns out the data is right skewed with a minimum of 1, mean of 26, standard deviation of 590, and maximum of 61974. Using these values, and by trial and error, the words that occur more than 60% of the mean and less than 1% of the maximum are considered as features. The idea is to remove words that occur infrequently or are on the tail of maximum occurrences (e.g., "is", "the", etc.), and to use the words that occur near the mean as features.

```
Finished Training: 0.445 s
---------------------------------------
Classes: neg: 0, pos: 1

Confusion Matrix:
    0  1
  -------
0 | 83 18
1 | 16 83

Metrics:
            0     1     mean
          ------------------
Precision | 0.838 0.822 0.830
Recall    | 0.822 0.838 0.830
F1        | 0.830 0.830 0.830

Overall Accuracy: 83.000 %
---------------------------------------
Finished Training + Evaluation: 0.523 s
```

LENGTH

Description: simply use as features the words in the vocabulary that include more characters than a specified length. By trial and error, a length greater than or equal to four performs best.

Effectively, this method removes all the one, two, and three character words such as “I”, “am”, “and”, etc., and uses only those that are longer.

```
Finished Training: 0.502 s
---------------------------------------
Classes: neg: 0, pos: 1

Confusion Matrix:
    0  1
  -------
0 | 83 18
1 | 27 72

Metrics:
            0     1     mean
          -------------------
Precision | 0.755 0.800 0.777
Recall    | 0.822 0.727 0.775
F1        | 0.787 0.762 0.776

Overall Accuracy: 77.500 %
---------------------------------------
Finished Training + Evaluation: 0.625 s
```

RATIO

Description: the ratio employed within is a slight alteration of the likelihood ratio used for NLTK that uses raw word counts instead of likelihood probabilities. Here, for all words the count ratio of a given word for two classes `i,j` is computed using the unequal cartesian product of classes. Thereafter, words that appear more often in one class over another with a ratio greater than a threshold are considered. By trial and error the value of 1.80 as a threshold works bestOnly the unique set of words are used as features, which is really only needed when there are more than two classes.

```
Finished Training: 0.474 s
---------------------------------------
Classes: neg: 0, pos: 1

Confusion Matrix:
    0  1
  -------
0 | 67 34
1 | 13 86

Metrics:
            0     1     mean
          -------------------
Precision | 0.838 0.717 0.777
Recall    | 0.663 0.869 0.766
F1        | 0.740 0.785 0.772

Overall Accuracy: 76.500 %
---------------------------------------
Finished Training + Evaluation: 0.572 s
```

Comments: I do not believe this is as effective as using the likelihood of a word for a given class, but it offers a little simpler approach of using the raw word counts. Unfortunately, using a threshold greater than two lowers performance, which is an unexpected observation. I also used the top *n* ratios as features, but had worse performance over the threshold method.

LENGTH+THRESHOLD

Description: a combination of applying the *length* and the *threshold* feature selection method. The actual features are the intersect of words returned from these two methods.

```
Finished Training: 0.501 s
----------------------------------------
Classes: neg: 0, pos: 1

Confusion Matrix:
    0  1
  -------
0 | 80 21
1 | 16 83

Metrics:
            0     1     mean
          -------------------
Precision | 0.833 0.798 0.816
Recall    | 0.792 0.838 0.815
F1        | 0.812 0.818 0.815

Overall Accuracy: 81.500 %
----------------------------------------
Finished Training + Evaluation: 0.580 s
```

REMOVE

Description: the removal method of feature selection attempts to remove as many features as possible without using a dictionary of nonessential words. As such, the threshold and removal methods are both performed and the intersection of words returned from these two methods are considered the base set. Thereafter, words that contain an apostrophe, double dash, or any numbers are then removed from the base set. While the removal of a word with an apostrophe removes contractions (e.g. “that’s”, “it’s”, etc.), which may be helpful, it also removes words that indicate possessions (e.g. “student’s”, “director’s”, etc.) which may not be helpful for classifying positive or negative reviews. Additionally, dashes and numbers do not seem important, and hence, are removed. The removal of words under these conditions take away 142 features from the base set and is used as the final set of selected features.

```
Finished Training: 0.550 s
----------------------------------------
Classes: neg: 0, pos: 1

Confusion Matrix:
    0  1
  -------
0 | 78 23
1 | 15 84
```

```
Metrics:
              0     1     mean
           ------------------
Precision | 0.839 0.785 0.812
Recall    | 0.772 0.848 0.810
F1        | 0.804 0.816 0.811

Overall Accuracy: 81.000 %
--------------------------------------
Finished Training + Evaluation: 0.620 s
```

Comments: Interestingly, removing additional words from the base set performs worse than the length+threshold method, but removal using the length as the base set improves performance.

Summary

	Accuracy (%)	F1
ALL	75.000	0.750
MANUAL	54.000	0.573
THRESHOLD	**83.000**	**0.830**
LENGTH	77.500	0.776
RATIO	76.500	0.772
LENGTH+THRESHOLD	81.500	0.815
REMOVE	81.000	0.811

Note: mean F1 score is that described in the lecture notes.