Mason Holcombe
2/13/2026

Homework 2 Writeup

For Homework 2, I decided to continue with the positive & negative movie review dataset provided. Much of the code I had written was to pre-process this data and determine frequencies of words per each class. The main `tokenize` function and movie review pre-processing has two parts:

1) Split sentence into individual words while removing single letter words & punctuation words
2) Remove individual punctuation from each word

After doing this for every review (both positive & negative), the final output of the pre-processing is a vocabulary, as well as lists of words from positive & negative review and the associated frequencies of the words in each class. A series of stop words have been removed from the frequency tables to avoid noise in the posterior calculation.

Step 1

To perform step 1, I began by concatenating the positive & negative reviews and randomly shuffling them. In total, there were 10,662 reviews. Every review has an associated label to indicate the sentiment. 0=Negative and 1=Positive. This dictionary of reviews was then split into three labeled datasets: Train (7,463), Dev (1,599), Test (1,600). Each of these datasets have ~50% split of positive & negative reviews.

Step 2

I implemented the Naive Bayes classifier by defining a class and constructing individual functions for each step along the way. The class takes a training dataset as a parameter and derives the frequency tables for both labels seen in the dataset. After constructing these tables, there are a series of functions which calculate the associated prior & conditional probability distributions. The first function, `calculate_priors`, takes in the training reviews and estimates the prior distribution of the class. There are an even number of each class in the total dataset, but the shuffling and selection for training might have not split 50/50. Regardless, this prior distribution will always be ~50%. The next function, `calculate_conditional_probabilities`, takes in a word and a label and returns the corresponding conditional probability of the word given that label. This is also the point in which Laplace smoothing was included. I did experiment with many values of alpha, but alpha=1 showed the best generalization accuracy. In both of these functions, I performed a logarithmic transformation on the probability. This had no impact on the performance of my classifier, but improved interpretability and avoided floating-point underflow. In the next function `estimate_label_posterior`, I made sure to sum over the log-probabilities of the review. In the final model building function, `naive_bayes_prediction`, I estimate the negative & positive sentiment posterior and argmax to predict the final label.

Final train with train_dataset accuracy: **94.6%**
Final test with dev_dataset accuracy: **75.0%**

Step 3

As the Naive Bayes Classifier takes training dataset as parameter, feed union of train & dev datasets as training, and evaluate on test dataset:

Final train with train_&_dev_dataset accuracy: **93.9%**
Final test with test_dataset accuracy: **77.9%**

Step 4

Very Confident (Large difference in pos/neg posterior)

warm in its loving yet unforgivingly inconsistent depiction of everyday people , relaxed in its perfect quiet pace and proud in its message . i loved this film .	Actual Label: 1 Predicted Label: 1
the redeeming feature of chan's films has always been the action , but the stunts in the tuxedo seem tired and , what's worse , routine .	Actual Label: 0 Predicted Label: 0
it's just too bad the screenwriters eventually shoot themselves in the feet with cop flick cliches like an oily arms dealer , squad car pile-ups and the requisite screaming captain .	Actual Label: 0 Predicted Label: 0

Very Uncertain (Small absolute difference in pos/neg posterior)

topkapi this is not .	Actual Label: 0 Predicted Label: 0
freundlich's made [crudup] a suburban architect , and a cipher .	Actual Label: 0 Predicted Label: 0
earnest but earthbound . . . a slow , soggy , soporific , visually dank crime melodrama/character study that would be more at home on the small screen but for its stellar cast .	Actual Label: 0 Predicted Label: 1

The classifier seems to be very confident in long reviews that clearly describe the sentiment throughout. For example: using warm, perfect, proud, loved. Similarly with negative sentiment reviews. Key words such as “worse” or “bad” clearly drive the classifier to predict negative. Alternatively, reviews which are short or use words that don’t describe much sentiment (suburban, architect, cipher) are not certain in the final predictions. The classifier predicted near exact neg/pos posteriors. The last example shows a review which had both negative & positive parts. The model described this particular example as positive when actually it is a negative review.

Step 5

Looking into the conditional probabilities of each class these words had the highest posterior in each class (after removing filler & stop words):

Negative Sentiment

1. bad
2. never
3. doesnt

Positive Sentiment

1. good
2. best
3. love

These words generally describe the overall sentiment of the review and provide a large contribution to the overall posteriors.