Programming Assignment One Write-up

Programming assignment one consisted of writing a multinomial naive Bayes classifier which focused on classifying the sentiment of a given movie review. The following sections discuss the most relevant topics of the classifier: feature selection and test results.

For any piece of software it makes the most sense to pick the easiest path to a working solution. With that in mind, the feature selection process of this model was initially just the entire vocabulary. This selection process was both easy to code as well as relatively expedient in terms of overall model runtime. The results of this method of feature selection can be found in the next section. The other method of feature selection used in this model is done by computing the normalized mutual information of each word. Calculating the normalized mutual information allows the model to choose words as features such that they meet some minimum threshold of mutual information. The experimentally derived threshold for this model and the given data that produced the highest accuracy is 0.00008. That is, as long as a word's normalized mutual information was larger than 0.00008, it was considered a good feature. The threshold is a hyperparameter of the model and is thus subject to change given different data and is likely not the optimal choice.

Results for both the full vocabulary feature selection and normalized mutual information feature selection can be found in tables 1 - 1b and 2 -2c respectively. Two rows in the tables are particularly interesting; the magnitude of the features vector and the time rows. These rows show the greatest discrepancy between the two feature selection methods. You can see that the full vocabulary method had approximately ten times as many features as the normalized mutual information feature selection method. However, you can see the normalized mutual information method took approximately ten times as long to select the features, train, and test on the data as compared to the full vocabulary method. The discrepancy in time and features does lead to a noticeably better model when using the normalized mutual information feature selection method. From tables 1 and 2 the discrepancy in accuracy of the two models with the two feature selection methods is 7%. It can also be observed that the precision and recall scores across the classes for both feature selection methods are quite close. That is no matter the feature selection method, the precision of the model does not dominate the recall and vice versa. The real determination between these two feature selection methods and the results of their respective models comes down to whether you need a lower runtime and a decent accuracy or higher accuracy with an extra penalty on runtime.

Accuracy	0.765
|Features|	42841
Time to train/test in seconds	1.6554994583129883

Table 1: Overall Metrics for Full Vocabulary

Negative Reviews	
Precision	0.7596153846153846
Recall	0.7821782178217822

F1	0.7707317073170732

Table 1a: Negative Reviews Full Vocabulary Metrics

Positive Reviews	
Precision	0.7708333333333334
Recall	0.7474747474747475
F1	0.758974358974359

Table 1b: Positive Reviews Full Vocabulary Metrics

Accuracy	0.835
|Features|	4782
Time to train/test in seconds	13.178499937057495

Table 2: Overall Metrics for Normalized Mutual Information

Negative Reviews	
Precision	0.8617021276595744
Recall	0.801980198019802
F1	0.8307692307692307

Table 2a: Negative Reviews Normalized Mutual Information Metrics

Positive Reviews	
Precision	0.8113207547169812
Recall	0.8686868686868687
F1	0.8390243902439025

Table 2b: Positive Reviews Normalized Mutual Information Metrics