CSE 5243 Assignment 1 Problem 2

Report of Assumptions, Steps, and Results - Colin Farrell

Step 1.

When parsing through the files, the words were transformed into lowercase, punctuation was removed, and the “stop words” were also removed. Words were also stemmed (e.g., “likes” became like) using a built-in Python feature.

All three files were parsed and loaded into the D matrix with 3000 rows for each sentence. The top 2048 words were used (the code detailing “max features”), meaning D shows the frequency of these words in each sentence, making the dimension, or shape, of the matrix 3000x2048.

Step 2.

Sentences were randomly split into three sets. The randomizer in the code could have been set to any number, but I used 1234.

I used the suggested breakdown of sets, putting 60% into training, and 20% were used for the validation and test sets. Further testing confirmed the split, showing the shapes were 1800x2048, 600x2048, and 600x2048 for training, validation, and test, respectively.

Step 3.

The feature importance measure I used was TF-IDF. The training set was the only set used for the feature and the feature vector was pruned to the top 512 words using the score.. This was confirmed by comparing the original training set size of 1800x2048 to now, after using TF-IDF, being 1800x512.

Step 4.

Here, I implemented the k-nearest neighbors (KNN) classifier and used the built-in implementations of Naive Bayes and SVM. All three classifiers were implemented using both the original matrix and the new, reduced TF-IDF one. The implementations were supported by the setup of the feature vectors so no additional cleanup of code was needed.

Step 5.

Here are the results using the original features and the TF-IDF features.

Original, “Bag-of-Words” approach (BOW)

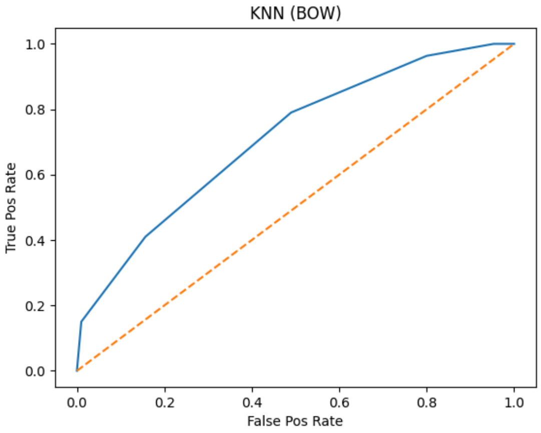
	Accuracy	Precision	Recall	Specificity	AUROC	Time
KNN	0.65	0.617	0.79	0.51	0.711	0.0
Bayes	0.793	0.782	0.813	0.773	0.867	0.002
SVM	0.76	0.758	0.763	0.757	0.844	0.005

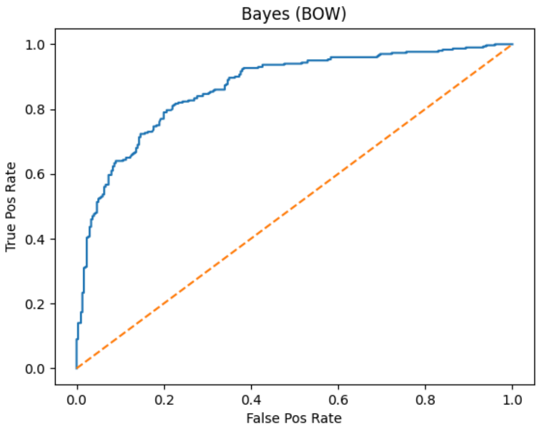
TF-IDF approach

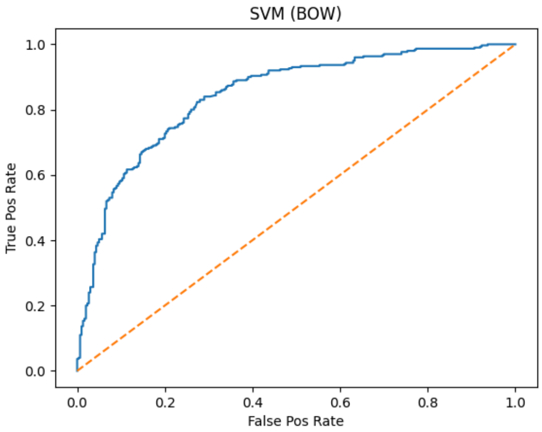
	Accuracy	Precision	Recall	Specificity	AUROC	Time
KNN	0.495	0.489	0.213	0.777	0.49	0.0
Bayes	0.522	0.526	0.44	0.603	0.519	0.002
SVM	0.525	0.525	0.53	0.52	0.525	0.004

Below, you will be able to find ROC curves for all 6 classifications.

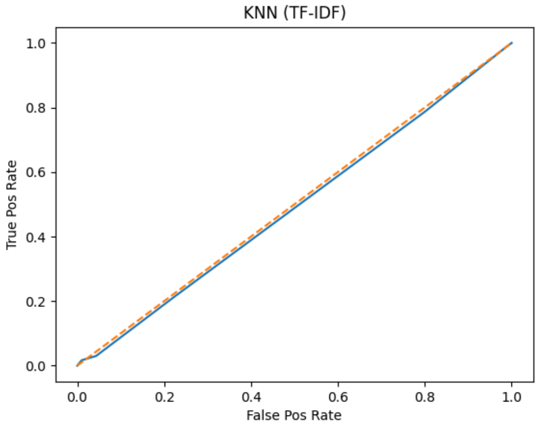
BOW Curves:

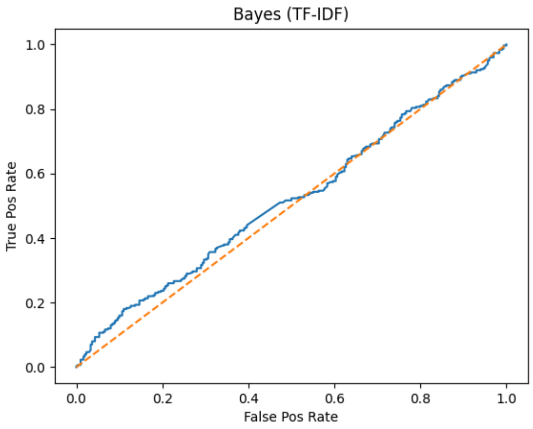


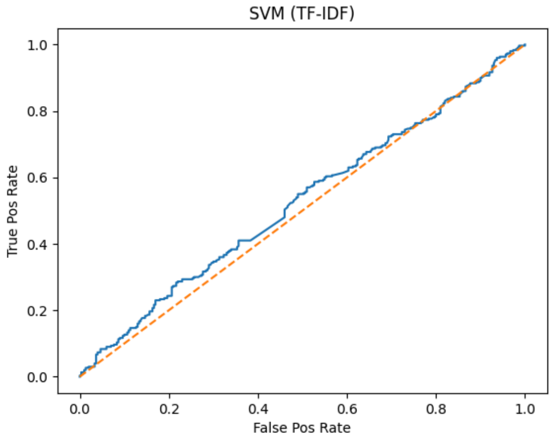




TF-IDF Curves:







Conclusion

My testing and above results show that the original approach with all of the words, and not the TF-IDF that selected the top 512, is an overall better feature class.

The data in the table is overall stronger for the original features compared to the TF-IDF features. The ROC curved for the original Bag-of-Words approach bows out more, showing a better classifier than the TF-IDF curves, which hug the diagonal and display more randomness in classifying. Also, the performance time was generally the same for both, meaning that using one over the other to improve time isn’t a factor considered.

Overall, my data shows that using a feature selection to narrow down the amount of sentences we have will not improve things, and is overall ineffective.