## *Pseudocode*

1. Read the input data
2. Perform preprocessing (tokenizing, removing stopwords, converting words to numerical vectors, creating IDF models)
3. Split the data into training and testing sets
4. Calculate prior probabilities
5. Create two dictionaries to represent the sums and counts of the features
   1. For every feature index, add that feature’s value to the sum and increment the count
6. Add/combine sums and counts of features from different samples/documents in the data
   1. Add values from the sums and counts of different samples
7. Calculate likelihoods
8. Create predictions for each sample/document in the data
   1. Create log probabilities for each class label
   2. For every feature, if it’s > 0, then first multiply the log probability of that specific feature with the value of that feature and add it to the class label’s log probability
   3. Return prediction for the class label that has the highest probability
9. Make predictions
10. Calculate evaluation metrics (accuracy, recall, precision, F1 score, and the confusion matrix)
11. Display evaluation metrics

*\*More detail about how things are done/calculated can be seen in the comments in the code\**

## *Summary of Results*

Above are the results I got with my Naive Bayes implementation. 