**Naive Bayes**

1)

2) In order to split the data into training data, testing data and validation data, the 'test\_train\_split' function was used to split the data into training and testing data(80/20 split, respectively). In order to pre-process the data regular expressions were used to remove punctuation from the headlines so that we could analyse the words accurately. We also used smoothing with an alpha value of 1, in order to avoid multiplying or dividing by a zero value.

3) For Naive Bayes we created several functions to help make predictions. The first being the get\_text function, which gathered all headlines with the same classifications (0 for not sarcastic and 1 for sarcastic headlines). We then implemented the count\_text function, which returns the counter for the amount of times a word occurs. The get\_y\_count function was implemented in order to obtain the number of items in each class(0 or 1). These functions were implemented in order to assist with making predictions and calculating probabilities, which are needed when implementing Naive Bayes.

The main functions involved in classifying the headlines were the make\_decison and make\_class\_preditctions. The former of which was used to classify each word in each headline as either sarcastic or not sarcastic, we also applied smoothing in this function in order to multiplying/dividing by a zero when performing calculations. The latter of the two functions made the final decision on the whole headline based on which there were more sarcastic or non-sarcastic words in the headline. We set our alpha value for smoothing to 1 as this yielded the greatest accuracy.

4) Our Naive Bayes algorithm yielded an accuracy of 0.859, the algorithm that we implemented predicted the headlines correctly 85% of the times. From our recall value of 0.86, we see that 86% of the headlines that were sarcastic were correctly classified as sarcastic. Our precision score was 0.82 which implies that 82% of all headlines that were predicted to be sarcastic, were in fact sarcastic. These high statistics tell us that a high percentage of the headlines were correctly classified. Our f1 score was 0.84 which implies that our Naive Bayes model is a good classifier. We recommend calculating probabilities that are needed for this function, such as standard deviation, mean, etc…for each class of words in separate functions as it allows calculations to be performed with a bit more ease and makes error tracing easier.