# Machine Learning Assignment 1:

For this assignment we where tasked with creating a python program that used the Bayesian Classification to predict the Sentiment of a review score based on the text it reads. The data used in this assignment was an excel file that was read through the pandas import in python.

## Task 1

Task 1 is in function “count()”, revolved around how we read the data given in the excel file to the program. This was done using pandas and taking in the data within the excel file as a Dataframe. We then filtered the specific data , for example : No. of positive test reviews. This was tested using the excel file itself and using the count command to get the same specific split of data and comparing.

## Task 2

Task 2 is in “cleanWords()” and involved taking in all reviews within the training set and removing all non-alphanumeric characters, converting everything in lowercase and splitting each word. This is done so that we can count the words involved without having any unwanted characters interfere with the count number. The split helps us read the reviews as individual words. This was tested using print strings after each call to see if it was correctly converting the letters in the review

The second part used “countOccurences()” to count the words within a training set. The function started with having a minimum length parameter and a minimum occurrence parameter that was used to get a basic requirement for a word to registered if it has met both conditions. The function itself loops around all reviews and then loops around all split words within the reviews. If a word meets the conditions(minWordLenght,), its added to the list and incremented. If the word count exceeds the min occurrence, it is then printed and returned as a list of words that meet the requirements. This was tested again through printing variables through the different loops and seeing if it was reading, counting and appending as expected.

## Task 3

Task 3 involves the list of words we have gotten from the previous task and looping through all positive and negative reviews, reading their words and seeing if any of the list of words that was collected in the prior task match with words in the review. We then count for each of these words, the number of both positive and negative sentiments and returned the set. We then counted the number f positive reviews and negative reviews for each word, the functions “posCount()” and “negCount” implement the count. This was tested again through printing through the loops and

seeing if we are getting the right data. In particular seeing if the appending in the word list at the end of the function could be counted was tested quite a bit.

## Task 4

Task 4 takes the return from two functions ( the positive review count and the negative review count from task 3) to calculate the number of both positive and negative reviews but more importantly calculating the likelihood that a word within our accumulated list will appear in a positive and negative review with a smoothing factor of 1. We looped around the count of words that we got in the prior task while also getting the words themselves printing by using an index on the Counter previously used. The frequencies, positive and negative are then appended to a list and both are returned in “calFreq()”. Testing done in this task was mainly done through calculating a temporary smaller set and seeing if the logic used was working and then applying it to the data set, we have been using up until now.

## Task 5

This task, “maxLikelihood()” is used to predict if a new review was added to the test set, what would be the likelihood of it being either positive or negative. We pass in both our lists of frequencies and perform the equation of Bayesian using math function to access to log command. After looping through the new review and the frequencies of each list a sentiment is printed out and meant to represent the predicted sentiment for the new review. At the very end there is a line of code to get the accuracy of our prediction using the test and training sets. This was tested again with using a smaller set of data to get a better idea of how these functions should work, unfortunately this task was not completed fully.

## Task 6

Task 6, “crossValid()” is a cross validation of our prediction derived from previous task to see if what we have is actually consistent. This is done using Stratified K-Fold and looping through our potential predictions based off the one we have given the new review. A mean average of predictions as well as finding predictions that are likely to be wrong is the purpose of this function. This task was not completed fully because of the pervious function failed to calculate a prediction, but the theory needed is coded in this function.