# **Assignment 4 Report**

Methodology: First I preprocessed the train and test data. First, I used re and substituted out all digits and basic punctuation with empty characters. Then I used nltk tokenize feature to split each document into tokens. Then I used nltk stop words data set to remove all the words that are tokenized that existed in the stop words. I performed this preprocessing on the test and train DataFrame and inserted it into the DataFrame. Then to perform TF-IDF vectorization I used sklearn vectorizer. I labeled the training data as X\_train and fit the training data to the vectorizer, which gave me a matrix of all floating-point numbers. Then using this training data I performed LogisticRegression using the training data and ran the score method to calculate accuracy.

Results + Discussion: The accuracy that I got when scoring the model on the test data was around 70 percent. I think this is pretty good for a very multi-label regression problem. I calculated the precision, recall, and f-1 score for each class and the first and the last classes had the worst results.

Part 2:

**Random Partition Initialization:**

One algorithm that is for k-means clustering is the random partition initialization method. In this method, each data point is randomly assigned to a cluster, and you take the average of each cluster to yield the initial points. The pros of this method are that it is very simple to implement, it is efficient, since the initialization is at random, it is done faster. The cons of this method are that since it is done at random, clustered structures will not be accurate, and the clustering solutions will not be the most optimized.

**K-means++ algorithm**

Another algorithm that is also used for the k-means clustering algorithm is the k-means++ algorithm. This algorithm finds cluster centers that minimize the intra-class variance. The pros for this algorithm are that it is better than random initialization is that the convergence speeds are higher, since it gathers a larger exploration of the data space, the clusters are better represented. It also generally requires fewer iterations. The cons for this algorithm are that it is more computationally expensive, and it does not work as well for weird data shapes/clusters because of the limitations in capturing the boundaries of the clusters.