Assignment 2: Sentiment Classification
CSPE73: Natural Language Processing

Due Date: 30/10/2022 **Max Marks: 10**

You will be using the movie review dataset, where these review snippets are taken from Rotten Tomatoes. You are going perform positive/negative binary sentiment classification of sentences, with neutral sentences discarded from the dataset. The data files given to you contain each sentiment (reviews) example per line. Each line consisting of a label (0 or 1) followed by a tab, followed by the sentence, which has been tokenized but not lowercased. The data has been split into a train, development (dev), and blind test set. On the blind test set, you do not see the labels and only the
sentences are given to you.

Feature extraction

First, you will need a way of mapping from sentences (lists of strings) to feature vectors, a process called feature extraction. A unigram feature vector will be a sparse vector with length equal to the vocabulary size. There is no single right way to define unigram features. For example, do you want to throw out low-count words? Do you want to lowercase? Do you want to discard stopwords? Do you want the value in the feature vector to be 0/1 for absence or presence of a word, or reflect its count in the given sentence?
(Features in Unigram=great Vs Bigram=great|movie.)
Feature extraction iterate through all training points and pre-extract features so you know how many there are in advance .

Suggestion: Feature vectors

Since there are a large number of possible features, it is always preferable to represent feature vectors sparsely. That is, if you are using unigram features with a 10,000 word vocabulary, you should not be instantiating a 10,000-dimensional vector for each example, as this is very inefficient. Instead, you want to maintain a list of only the nonzero features and their counts. Since the features are strings here, you should may each feature to an integer.

Q1) Implement Naïve Bayesian Classifier.

a) Consider unigram features
   i. Count Versus binary representation of features
   ii. Add weights to features i.e. add tf-idf
b) Consider bigram features
   i. Count Versus binary representation of features
   ii. Add weights to features i.e. add tf-idf

Q2) Implement Logistic Regression

a) Consider unigram features
   i. Count Versus binary representation of features

ii. Add weights to features i.e. add tf-idf

b) Consider bigram features

i. Count Versus binary representation of features

ii. Add weights to features i.e. add tf-idf

Deliverables

1. Code in one folder

2. A short report discussing the results with respect to different feature representations and classifiers and your observations. (describing the purpose or functionality of each code file). Also Include the results from training and test data in the report.