Programming Assignment 3: Multinomial Logistic Regression

Instructions:

- The aim of this assignment is to give you an initial hands-on regarding real-life machine learning application.
- Use separate training and testing data as discussed in class.
- You can only use Python programming language and Jupyter Notebook.
- You can only use **numpy**, **matplotlib** and are not allowed to use **NLTK, scikit-learn or any other machine learning toolkit**.
- **Submit your code as one notebook file (.ipynb) on LMS. The name of file should be your roll number.**
- Deadline to submit this assignment is: **Thursday 2nd April, 2020 11:55 p.m.**

Problem:

The purpose of this assignment is to get you familiar with multinomial sentiment classification. By the end of this assignment you will have your very own "Sentiment Analyzer". You are given with Twitter US Airline Sentiment Dataset that contains around 14,640 tweets about airlines labelled as positive, negative and neutral. Your task is to train a Multinomial Logistic Regression classifier on this dataset.

Dataset Splitting:

Instead of a usual random split, you will split the dataset in a stratified fashion. Stratified splitting ensure that the train and test sets have approximately the same percentage of samples of each target class as the complete set. For example, in an 80-20 stratified split 80% samples of each class will be in train set and 20% in test set.

Implement stratified split and do the 80-20 train-test split of the provided dataset.

Dataset Preprocessing:

You'll represent a tweet as a bag-of-words, that is, an unordered set of words with their position ignored, keeping only their frequency in the tweet. For example, consider the below documents:

D1 = John likes to watch movies. Mary likes movies too.
D2 = Mary also likes to watch football games.

The bag-of-words representation (ignoring case and punctuation) for the above documents are:

Vocabulary	D1	D2
john	1	0
likes	2	1
to	1	1
watch	1	1
movies	2	0
mary	1	1
too	1	0
also	0	1
football	0	1
games	0	1

Similarly, represent all the tweets in the provided dataset as bag-of-words. Please note that in your case the vocabulary might be in thousands, so you can use text cleaning techniques such as ignore case, punctuation and frequent (stop) words like “a”, ”an”, “the” etc. to reduce the size of vocabulary.

Implementation:

Implement Multinomial Logistic Regression keeping in view all the discussions from the class lectures. Feel free to read Chapter 5 (Section 5.6) of Speech and Language Processing book to get in-depth insight of Multinomial Logistic Regression classifier. Specifically, you’ll need to implement the following:

- Dataset Splitting function
- Dataset Preprocessing function
- Softmax function
- Cross-entropy loss function
- Mini-batch Gradient Descent with batch size of 32 samples
- Prediction function that predict whether the tweet is positive, negative or neutral using learned multinomial logistic regression
- Evaluation report

Use the procedural programming style and comment your code thoroughly (just like programming assignment 1).

Evaluation Report:

You are required to provide a confusion matrix with values obtained by running your Multinomial Logistic Regression classifier on test set. Also report micro and macro average (Precision, Recall, Accuracy, and F1) scores.