Assignment 4: Text Classification

1. Download “amazon_review_300.csv”. This dataset has three columns: label, title, and text. In this assignment, we build classifiers using label and text columns. The classifiers are used to classify text into one of the two labels (i.e. 1 or 2).

2. Experiment 1: **Compare the performance of classifiers with/without lemmatization**
   Write a block of code to create two **MutinomialNB** classifiers with 5-folder cross-validation using tf-idf matrixes generated by two different approaches:
   a) Use **TfidfVectorizer** from sklearn package with **stop words removed** option to generate tf-idf matrix.
   b) Use your solution to Assignment 3(b) to create a tf-idf matrix with **lemmatization option set to True.**

   Compare the performance of each classifier using the **average macro precision and recall** over the 5 folders, and write your analysis in a document.

3. Experiment 2: **Tune parameters using grid search**
   Write a block of code to tune the classifier you created in Step 2(a) using grid search. The grid search is to find best values for the following parameters:
   - **stop_words**: [None,"english"]
   - **min_df**: [1,2,3]
   - **alpha**: [0.5,1.0,1.5,2.0]

   With the best parameter returned from grid search, use them to train a classifier with 5-folder cross-validation, and report the average macro performance metrics over each fold. Compare the performance with the classifier in Step 2(a) and write your conclusion in the document

4. Experiment 3: **How many samples are enough? Show the impact of sample size on classifier performance**
   Download “**amazon_review_large.csv**” which contains 20,000 reviews. Starting with 300 samples, in each round you build a classifier with 300 more samples. i.e. in round 1, you use samples from 0:300, and in round 2, you use samples from 0:600, …, until you use all samples. In each round, do the following:
   a. create tf-idf matrix using **TfidfVectorizer** with stop words removed
   b. train the classifier using **linearSVC** model with 10-fold cross validation
   c. collect the macro precision/recall/f1 metrics for each fold and take the average of the 10 fold

   Draw a line chart show the relationship between sample size and each average performance metric (reference code provided below). Write your analysis on how sample size affects classification performance in the document

Submission guidelines:

Submit a python file that contains script for all 3 experiments, but clearly separate your code for each experiment. Also, have a print statement in your code to print out the performance metrics of each experiment.

```
import numpy as np
import matplotlib.pyplot as plt

# metrics is a list of list, e.g. [[300, 0.7, 0.7,0.7],
# [600, 0.78, 0.73,0.74], …]
results=np.array(metrics)
plt.plot(results[:,0], results[:,1], '-', label='precision')
plt.plot(results[:,0], results[:,2], '-', label='recall')
plt.plot(results[:,0], results[:,3], '-', label='f1')
plt.title('Impact of sample size on classification performance')
plt.ylabel('performance')
plt.xlabel('sample size')
plt.legend()
plt.show()
```