Overview and Assignment Goals:

****

The objectives of this assignment are as follows:

- Implement the Nearest Neighbor classification algorithm
- Handle text data (Yelp reviews)
- Design and engineer features from text data.
- Choose the best model i.e., parameters of the Nearest Neighbor algorithm, features and similarity functions

****

Detailed Description:

****

A practical application in e-commerce applications is to infer sentiment (or polarity) from free-form review text submitted for a range of products.
For the purpose of this assignment, you have to implement a k-Nearest Neighbor Classifier to predict the sentiment for 18,000 reviews for various products provided in the test file (new_test.csv).

Positive sentiment is represented by a review rating of +1 and negative sentiment is represented by a review rating of -1.

In the test file, you are only provided the reviews but no ground truth rating which will be used for comparing your predictions.

Training data consists of 18000 reviews and exists in the file new_train.csv. Each row begins with the sentiment score followed by the text of the review.

For evaluation purposes (Leaderboard Ranking) we will use the simple accuracy metric comparing the predictions submitted by you on the test set with the ground truth. Some things to note:

The public leaderboard shows results for 50% of randomly chosen test instances only. This is standard practice in data mining challenges to avoid gaming of the system and prevent overfitting.

The private leaderboard will be released after the deadline evaluates all the entries in the test set. In a 24-hour cycle, you are allowed to submit a prediction file 10 times only.

format.csv shows an example file containing 18000 rows alternating with +1 and -1.

Your final submission should be similar to format.csv with same number of rows i.e., 18000 but the sentiment prediction should be generated by your developed model.

Rules:

- This is an individual assignment. Discussion of broad level strategies is allowed but any copying of prediction files and source code will result in an honor code violation.
- You can use a programming language of your choice for this assignment, but Python is highly recommended (see tutorial posted on Blackboard).
- While you can use libraries and templates for dealing with text data you are required to implement your own nearest neighbor classifier.

Deliverables:

- Valid submissions to the miner2.vsnet.gmu.edu website
- Gradescope submission of report and source code:
  Create a folder called HW1_LastName, and put all the source code there.

  Submit (on Gradescope) a 3-page, single-spaced report in PDF format describing details regarding the steps you followed for developing the classifier for predicting the product review sentiments.

  Be sure to include the following in the report:

  1. Your identifier as registered on the miner website.

  2. Rank and accuracy score for your submission (at the time of writing the report).

  3. A detailed and clear description of your approach, and how and why you chose the parameters, features, distance/similarity measure, etc.

  4. Any graphs or tables illustrating key experiments you did in the process of choosing your final model.

Also submit an archive of your HW1_LastName folder (.zip or .tar.gz) via Gradescope.

Grading:

Grading for the Assignment will be based on your **implementation (40%)**, **report (30%)** and **ranking results (30%)**.

Files:

- *Train Data:* Download File (/files/uploaded_files/1662145253_9967718_new_train.csv)
- *Test Data:* Download File (/files/uploaded_files/1662145254_0395114_new_test.csv)
- *Format File:* Download File (/files/uploaded_files/1662145254_0812266_format.csv)