CP421 Data Mining - Assignment 2
Due Date: Nov 6, 2023 at 11:59 PM

Assignment Submission Guidelines

1. **File Naming**: Ensure your assignment file is named in the following format: your network login followed by the assignment number. For instance, if the username is "barn4520" and you're submitting Assignment 1, the file should be named "barn4520_a01.ipynb".
2. **Assignment Format**: All assignments should be completed using Jupyter Notebook. Once you're done, ensure you run all the cells to verify their functionality, then save your work as a ".ipynb" file. For theoretical or conceptual queries, provide your responses within the notebook using markdown cells. Coding segments must be well-documented for clarity.
3. **Submission Platform**: All submissions must be made via the MyLearningSpace website. We do not accept assignments through email.
4. **Late Submissions**: If you submit your assignment within 24 hours post the deadline, your grade will be reduced by 50%. Unfortunately, we cannot accept submissions made beyond 24 hours from the deadline, and such submissions will receive a grade of 0.
5. **Plagiarism Policy**: At WLU, we take academic integrity seriously. All submitted code will undergo a plagiarism check. Engaging in plagiarism can have significant academic consequences.

Before You Start

This assignment may use the following Python packages. Feel free to have a look before you start.

1. **pandas**:
   - *Usage*: Data manipulation and analysis.
   - *Applied In*: Loading datasets from CSV files, data preprocessing, and handling data in tabular format.
2. **numpy**:
   - *Usage*: Numerical computations and handling arrays.
   - *Applied In*: Mathematical computations, array manipulations, and other numerical operations.
3. **sklearn**:
   - *Usage*: Machine learning library.
   - *Applied In*:
     - Text preprocessing: `TfidfVectorizer` for converting text to vectors.
     - Clustering algorithms: `KMeans`, `DBSCAN`, `AgglomerativeClustering`, and `GaussianMixture`.
     - Dimensionality reduction: `PCA` for visualizing clusters.
     - Nearest Neighbours: `NearestNeighbors` for estimating optimal *eps*.
4. **matplotlib**:
   - *Usage*: Plotting and visualization.
   - *Applied In*: Visualizing the results of clustering and other data visualizations.
5. **nltk**:

- *Usage*: Natural Language Toolkit for text processing.
- *Applied In*:
  - Stop words removal: `stopwords`.
  - Lemmatization: `WordNetLemmatizer`.

About the Data

In this assignment, you will undertake the task of clustering documents, specifically news articles. The dataset, sourced from BBC News, serves as a standard for machine learning studies. It encompasses 2225 articles collected from the BBC news website, reflecting stories spanning five themes from the years 2004-2005. The original dataset contained five class labels: business, entertainment, politics, sport, and tech. For the purpose of this assignment, we have excluded the class labels, retaining only the title and content of each article. For simplicity, you can combine each title and content together to form an article vector. The dataset can be downloaded from MyLS as 'bbc-news-data-modified.csv'.

Text Preprocessing

With the BBC news dataset at hand, the initial step involves preparing the textual data for subsequent clustering. This preparation is multi-faceted and encompasses:

1. **Tokenization** refers to the process of breaking down a piece of text into smaller units, commonly known as tokens. Typically, tokens are words, but they can also be phrases, sentences, or any other unit that makes sense for the specific analysis. Tokenization is essential because it helps convert the unstructured form of textual data into a form that can be utilized in various Natural Language Processing (NLP) tasks.
2. **Stop words removal** are commonly used words in any language which don't add much meaning to a sentence. Words like 'and', 'the', 'is', and 'in' are examples of stop words. In text mining and search engines, these words are eliminated from the text to expedite the processing.
3. **Lemmatization and stemming** involves converting a word to its base or dictionary form. For instance, 'running' becomes 'run', 'better' becomes 'good'.
4. **Vectorizerization** can be used to transform the text data into a matrix of TF-IDF features. This matrix serves as the input for the clustering algorithms, enabling them to cluster the documents based on the significance and occurrence of terms within them.

Clustering Implementation

Upon preprocessing the text, the next phase is to perform clustering. Given that the original data contains five classes, it's reasonable to expect five clusters during the clustering process. You're tasked with implementing four different clustering algorithms:

1. **K-means Clustering**: With the processed text, segregate the news articles into 5 distinct clusters using the K-means algorithm.
2. **DBSCAN Clustering**: Utilize the preprocessed text and implement the DBSCAN clustering algorithm.
3. **Gaussian Mixture Model (GMM) Clustering**: Allocate the news articles into 5 clusters leveraging the Gaussian Mixture Model.
4. **Agglomerative Clustering**: Administer Agglomerative Clustering on the processed text, aiming to form 5 coherent clusters.

Questions

As you build the clusters, consider addressing these questions:

1. For each clustering method, report the number of articles encapsulated in each cluster (10 points).
2. Illustrate the clusters derived from every clustering technique, employing PCA (Principal Component Analysis) condensed to 2 components. Plot these clusters after dimensionality reduction. (4 points).

3. In the DBSCAN algorithm, assuming $MinPts = 2$, implement the K-distance Graph to estimate the optimal eps. Plot the distance of every point to its $k - th$ nearest neighbor, where k = min_samples. The idea is to find an "elbow" in this plot, which can be a good estimate for eps. Additionally, quantify the count of detected noise points and compare the clustering result with the outcomes from other clustering methods. Comment on your observations. (3 points).

4. Utilize the Within Cluster Sum of Squares (WCSS) method to identify the ideal number of clusters k for the K-means algorithm. Evaluate potential cluster counts ranging from 1 to 10. For each prospective k, initialize the KMeans module with the current cluster count, leveraging the "k-means++" technique for centroid initialization, setting a cap of 300 iterations per individual run, opting for 10 initial centroid configurations, and ensuring reproducibility with a random state of 0. Illustrate the resulting WCSS values on a plot. The plot's "elbow" point, where the decline becomes pronounced, denotes the optimal k value.(3 points).