### Comprehensive Outline for Assignment 4: Cluster Analysis, Text Mining, and ANN

#### ****General Information****

* **Due Date**: Friday, 11:59 PM
* **Points**: 100
* **Submission**: File upload
* **Datasets**:
  1. IMDB dataset: imdb\_dataset.csv (located in Files -> Labs -> Data).
  2. Admission dataset for ANN: previously used in Assignment 2.
  3. Text dataset (provided as a Python list).

### ****Part 1: Cluster Analysis****

1. **Objective**: Partition the IMDB dataset into logical groupings using clustering algorithms.
2. **Steps**:
   * **K-Means Clustering**:
     + Plot the **SSE vs. Number of Clusters** using the **Elbow Method** to determine the optimal number of clusters.
     + Apply K-Means with the chosen number of clusters.
   * **Hierarchical Clustering**:
     + Perform hierarchical clustering using the following linkage methods:
       - **Single Link**
       - **Complete Link**
       - **Group Average Link**
     + Visualize the hierarchy using a **dendrogram**.
3. **Notes**:
   * Remove labels for clustering (unsupervised learning).
   * Use TF-IDF for feature extraction:
     + Extract unique words across all documents.
     + Create count vectors for each document.
     + Compute TF-IDF vectors using the inverse document frequency formula.
     + Normalize the data to a [0, 1] interval for clustering.
     + Use normalized numerical vectors for distance computation in K-Means.
   * Reference Articles:
     + K-Means Clustering: Daniel Foley’s Medium article.
     + Text Feature Extraction: scikit-learn’s [documentation](https://scikit-learn.org/stable/modules/feature_extraction.html#text-feature-extraction).

### ****Part 2: Text Mining****

1. **Objective**: Analyze textual data to uncover key concepts and trends, and transform text into numeric representations.
2. **Steps**:
   * Create the following representations for the provided text dataset:
     + **Count Vector**: Calculate the frequency of each word across documents.
     + **TF-IDF Vector**: Transform the text using Term Frequency-Inverse Document Frequency.
   * Display and explain the results of both vectors, particularly the purpose and utility of TF-IDF in machine learning.

### ****Part 3: Artificial Neural Networks (ANN)****

1. **Objective**: Implement ANN for classification using the **admissions dataset**.
2. **Steps**:
   * Preprocess the data:
     + Normalize numerical variables to a [0, 1] scale if using the **logistic activation function**.
     + Scale data to a [-1, 1] range for the **tanh activation function**.
     + Encode categorical variables:
       - Use dummy variables for nominal categories.
       - Map ordinal categories to fractions in [0, 1] based on their order.
     + Apply a log transformation to highly skewed numerical variables before normalization.
   * Modify the code from **Tutorial 6** to work with the admissions dataset.
     + Hint: Encode the target attribute to binary values.
3. **ANN Implementation**:
   * Include:
     + Input layer
     + Hidden layer(s)
     + Output layer
   * Train the model using appropriate activation functions.

### ****Resources****

1. IMDB dataset: [Kaggle IMDB dataset description](https://www.kaggle.com/iarunava/imdb-movie-reviews-dataset).
2. Python code for clustering and TF-IDF (referenced in instructions and linked articles).
3. ANN preprocessing steps: From "Data Analytics for Business Analytics" by Galit Shmueli.
4. Tutorial 6 code for ANN implementation.

This structure ensures a logical progression through the tasks and leverages detailed guidelines from both documents.