# **EMBEDDING ASSIGNMENT**

## **Problem Statement:**

This assignment aims to explore various text embedding techniques using a chosen dataset. By implementing different embedding methods such as **Bag of Words**, **TF-IDF**, **Word2Vec**, **GloVe**, and **FastText**, students will gain a deeper understanding of how to convert text into numerical representations. These embeddings will be used to analyze and extract meaningful insights from the dataset.

## **Guidelines:**

### **1. Foundational Knowledge:**

* Understand the principles of different embedding techniques and their applications.
* Familiarize yourself with the concepts of
  + vector space models
  + word embeddings and
  + the significance of context in text representation.
* Recognize the strengths and limitations of each embedding technique.

### **2. Embedding Techniques Implementation:**

* **Bag of Words (BoW):** Implement the BoW model to represent text as a sparse matrix of word counts.
* **TF-IDF:** Apply the TF-IDF model to weigh the importance of words in the text based on their frequency and inverse document frequency.
* **Word2Vec:** Train a Word2Vec model to capture the semantic relationships between words using the skip-gram or continuous bag of words (CBOW) approach.
* **GloVe:** Use pre-trained GloVe embeddings or train a GloVe model on the dataset to capture global word-word co-occurrence statistics.
* **FastText:** Implement FastText embeddings to consider subword information and improve the representation of rare words.

## **Step-by-Step Approach to Embedding Techniques:**

### **1. Setup and Data Preparation:**

* Import necessary libraries: pandas, numpy, nltk, sklearn, gensim, matplotlib, seaborn.
* Load the dataset: Choose a dataset for embedding generation.
* Preprocess the text data: Perform cleaning, tokenization, stop-word removal, and other preprocessing steps.

### **2. Bag of Words (BoW):**

* Create a BoW representation: Use CountVectorizer from sklearn to convert text into a matrix of word counts.
* Analyze the BoW matrix: Examine the sparsity and interpret the word frequencies.

### **3. TF-IDF:**

* Apply TF-IDF transformation: Use TfidfVectorizer to generate TF-IDF features from the text data.
* Interpret TF-IDF scores: Analyze how TF-IDF weighs different words in the dataset.

### **4. Word2Vec:**

* Train a Word2Vec model: Use Gensim's Word2Vec to train a model on the dataset.
* Extract word vectors: Obtain vector representations for words and visualize them.

### **5. GloVe:**

* Use pre-trained GloVe embeddings: Load GloVe vectors and map words in the dataset to their corresponding embeddings.
* Train a GloVe model (optional): Use the GloVe package to train embeddings if desired.

### **6. FastText:**

* Train a FastText model: Use Gensim’s FastText to train embeddings that consider subword information.
* Analyze FastText embeddings: Compare them with Word2Vec and GloVe embeddings.

## **Dataset for the Assignment:**

* Use any dataset and create embeddings.

## **Submission:**

* **Detailed Document:** A report
  + detailing each embedding technique
  + the process followed
  + and the results obtained.
* **Jupyter Notebook:** A notebook containing
  + all code
  + Explanations
  + and visualizations for the assignment.