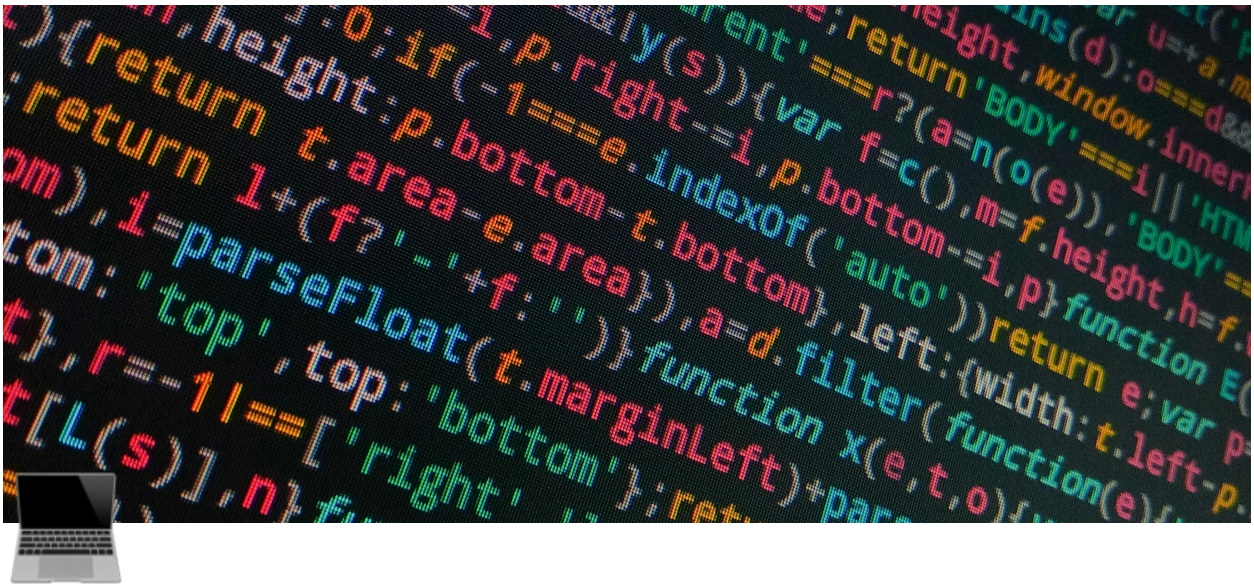

BDA Assignment #02

Created	@March 28, 2024 2:30 AM
Class	Big Data

Project Report: Text Mining and Analysis Pipeline

Date: March 31, 2024

Team Members:

- Muhammad Rayyan Mohsin 22i-2052
- Sufyan Nasr 22i-1898
- Haider Farooq 22i-1943

Overview:

The goal of our project is to analyze and extract insights from a substantial dataset containing textual information. We have undertaken a multi-phased approach that involves data preprocessing, TF-IDF vector generation, and the implementation of MapReduce tasks using Hadoop clusters.

Data Preprocessing:

In the initial phase, we performed data preprocessing to clean and standardize the textual data. This involved removing special characters, tokenizing sentences, removing stopwords, and lemmatizing words using the NLTK library. The preprocessed data was then saved into a cleaned dataset named "Cleaned_Dataset.csv".

TF-IDF Vector Generation:

Following data preprocessing, we developed code to generate TF-IDF values for each word in a document or sentence. This code calculates the TF-IDF values by first computing term frequency (TF) and inverse document frequency (IDF) for each word in the document. The TF-IDF vectors provide a numerical representation of the importance of words in the document relative to the entire corpus.

Normalization of TF-IDF Vectors:

After generating the TF-IDF values, we normalized the vectors to ensure that each document vector has a unit norm. This normalization step is crucial for ensuring that documents of varying lengths can be compared effectively in subsequent analyses. The normalization process was implemented using NumPy.

Hadoop Clusters on Microsoft Azure VMs:

One of the key challenges we encountered was handling the sheer volume of data. To address this challenge, we leveraged the scalability and parallel processing capabilities of Hadoop, deploying local clusters on Microsoft Azure

Virtual Machines (VMs). This strategic decision enabled us to distribute data processing tasks efficiently across multiple nodes, resulting in faster computations and efficient resource utilization.

MapReduce Implementation:

To scale our analysis to large datasets, we utilized the MapReduce paradigm within the Hadoop framework. We developed three sets of mapper and reducer files to handle various tasks:

1. **Mapper1.py / Reducer1.py:** Tokenizes input text and emits key-value pairs where the key is a word and the value is the count of occurrences of that word within the document.
2. **Mapper2.py / Reducer2.py:** Calculates the inverse document frequency (IDF) values for each term across the entire corpus.
3. **Mapper3.py / Reducer3.py:** Combines term frequency (TF) and IDF values to calculate TF-IDF scores for each term-document pair.

Conclusion:

In conclusion, our project has made significant progress in analyzing and extracting insights from textual data. Through meticulous data preprocessing, TF-IDF vector generation, and the implementation of MapReduce tasks using Hadoop clusters, we have laid a robust foundation for advanced analysis. Moving forward, we will continue to refine our methodologies, leverage the TF-IDF vectors for advanced analyses, and document our findings comprehensively. Our ultimate goal is to deliver valuable insights that contribute to our domain of interest and facilitate informed decision-making.