# Assignment 1: Data Preprocessing and Sentiment Analysis

Name: Jaspreet Kaur

Course: COMP 262

Submission Date: 06 Feb 2025

## Exercise #1: Web Scraping

### Overview

The objective of this exercise was to scrape data from the Centennial College Artificial Intelligence program webpage, extract relevant details using automated web scraping tools, and save them into a text file.

### Steps Taken

* Utilized Python with Selenium for web scraping and BeautifulSoup for parsing the HTML content.
* Extracted details including:
  + Webpage title
  + Program highlights
  + First two meaningful paragraphs from the program overview
* Data was saved into a text file named jaspreet\_my\_future.txt.

### Challenges and Solutions

* Initially faced issues with dynamically loaded content which prevented data extraction using simple HTTP requests.
* **Solution:** Implemented Selenium to control a headless browser, which allowed for the dynamic content to be loaded and accessed.

### Results

* Successfully extracted the webpage title, program highlights, and the initial sections of the program overview.
* Saved the extracted data into a text file, ensuring that all information was captured despite the dynamic nature of the webpage content.

Screenshot:

A screenshot of a computer

Description automatically generated

A screenshot of a computer

Description automatically generated

Output file: 

## Exercise #2: Text Preprocessing and Data Augmentation

### Overview

This exercise involved cleaning tweet data, applying preprocessing steps, and performing data augmentation using word embeddings (word2vec).

### Steps Taken

1. Loaded dataset (covid19\_mini.csv).

2. Dropped the user column.

3. Applied the following cleaning techniques:  
 - Removed URLs, mentions, hashtags, numbers, and extra spaces.  
 - Converted text to lowercase.  
 - Tokenized text and removed stopwords.

4. Additional preprocessing steps:  
 - Removed special characters and punctuation.  
 - Applied stemming using NLTK’s PorterStemmer.  
 - Handled contractions (e.g., 'don't' → 'do not').

### Challenges and Solutions

• word2vec vocabulary issues: Some words did not exist in the model.  
 - Solution: Ensured replacements were selected only from existing words in word2vec.

### Results

• Dataset size doubled, containing both original and augmented tweets.

Screenshot:

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

Final output file : 

## Exercise #3: Sentiment Analysis

### Overview

The goal was to analyze sentiment in a dataset using a lexicon-based approach. Since my name starts from ‘J’ so I will use covide19\_mini.csv.

### Steps Taken

1. Loaded the dataset and dropped the user column.

2. Performed text cleaning using regular expressions.

3. Added a tweet\_len column to store tweet lengths.

4. Used predefined positive and negative word lists.

5. Computed sentiment scores and normalized them by dividing by tweet length.

6. Applied classification rules to determine sentiment.

### Results and Observations

• Accuracy was 70%, showing moderate performance.  
• F1-score was 0.61, indicating room for improvement.

Screenshot:

A screenshot of a computer

Description automatically generated

A screenshot of a computer program

Description automatically generated

A screenshot of a computer program

Description automatically generated

## Conclusions & Improvements

✔ Web scraping requires handling dynamic content using JSON parsing or Selenium.

✔ Preprocessing steps improve data quality, but more advanced NLP techniques can be used.

✔ Data augmentation using word embeddings increases dataset diversity.

✔ Lexicon-based sentiment analysis has limitations—struggles with sarcasm, negations, and short tweets.