**Analysis Report – COMP262 Assignment 1**

**Exercise 1: Web Scraping**

**Objective:**  
To extract information about the Artificial Intelligence program from Centennial College’s website, including:

* Website title
* Program highlights
* Program overview (first two paragraphs)

**Method:**

* Implemented using requests + BeautifulSoup for HTML parsing.
* Scraped page content and attempted to locate highlights and overview sections.
* Saved results into Parichit\_my\_future.csv.

**Challenges:**

* The Centennial website uses **JavaScript to load content dynamically**.
* Requests only capture static HTML → sidebar links (Visit Us, Donate, etc.) were retrieved, but main program details were missing.
* A Selenium-based scraper would be needed to fully load the dynamic content.

**Results:**

* Extracted website title successfully.
* Program highlights and overview not fully captured.
* CSV file created: **Parichit\_my\_future.csv**

A screen shot of a computer

Description automatically generated

**Exercise 2: Text Preprocessing & Data Augmentation**

**Objective:**  
To preprocess tweets and augment data using Word2Vec to improve the dataset size and diversity. Dataset used: Artificial\_Intelligence\_mini.csv (since first name begins with P).

**Preprocessing Steps:**

1. Converted text to lowercase.
2. Removed URLs (http...).
3. Removed mentions (@username).
4. Removed punctuation, numbers, and special characters.
5. Added a new clean\_tweet column.

**Data Augmentation:**

* Used **Google News Word2Vec model**.
* For each tweet:
  + Tokenized and removed stopwords.
  + Replaced up to 3 random words with semantically similar words from Word2Vec.
  + Created augmented tweets while keeping sentiment unchanged.
* Combined original and augmented tweets → dataset size doubled.

**Results:**

* Original dataset size: 4 tweets.
* After augmentation: 8 tweets.
* Output saved as: **parichit\_df\_after\_random\_insertion.csv**

A computer screen with white text

Description automatically generated

**Exercise 3: Lexicon-Based Sentiment Analysis**

**Objective:**  
To predict tweet sentiment using lexicons and compare against labeled data. Dataset used: Artifical\_inteligence\_data.csv.

**Preprocessing Steps:**

1. Removed URLs, mentions, punctuation, and numbers.
2. Converted text to lowercase.
3. Added a tweet\_len column (word count per tweet).

**Method:**

* Loaded **positive** and **negative lexicon files**.
* For each tweet:
  + Counted words appearing in each lexicon.
  + Calculated % positive and % negative relative to tweet length.
* Classification rules:
  + Positive if % positive > % negative
  + Negative if % negative > % positive
  + Neutral if equal

**Evaluation:**

* Compared predicted\_sentiment\_score vs. original sentiment.
* Metrics:
  + **Accuracy:** 0.459 (~46%)
  + **F1 Score:** 0.475 (~47%)

**Discussion:**

* Performance is low because lexicons cannot capture context, sarcasm, or slang.
* Tweets often contain complex language structures that rule-based methods fail to handle.

**Improvements Suggested:**

* Use **machine learning classifiers** (Naive Bayes, Logistic Regression).
* Apply **word embeddings** (Word2Vec, GloVe, BERT) for contextual understanding.
* Expand lexicons with Twitter-specific slang and emojis.

**Results File:**

* **parichit\_sentiment\_analysis.csv**

A screen shot of a computer

Description automatically generated