**NLP Statistics Analyzer – README**

*Kirthana Ramesh (kirthanaramesh@arizona.edu)*

**Overview**

This project comprises two main sections:

1. Scraping and analyzing health-related web pages using Python to gather NLP-related statistics.
2. A Java-based web application to analyze text input and compare it with pre-aggregated NLP statistics.

**According to Task 1 – Generation Data Science and ML Bullets 1 & 2: Python Web Scraping and NLP Analysis**

Development Environment

* IDE: Visual Studio
* Python Version:3.11

Python Libraries Used are

* requests: Used for making HTTP requests to web pages for scraping content.
* bs4 (BeautifulSoup): Utilized for parsing HTML and XML documents, extracting necessary data from web pages.
* spacy, Textblob, nltk: Utilized for NLP related tasks
* pandas: Employed for data manipulation and analysis, particularly for handling dataframes and exporting results to CSV files.

Process

* The script accesses the health-related website https://www.healthline.com/directory/topics and extracts href links from *`<a>`* tags with the class *'css-1hacg05'*.
* It then scrapes content from each linked page, focusing on data within *`<p>`* tags.
* NLP statistics calculated include the number of words, sentences, verbs, common nouns (singular and plural), proper nouns (singular and plural), and type-token ratio (TTR). TTR is a linguistic measure indicating vocabulary diversity.
* Results are stored in a list of maps, converted to a dataframe, and then exported as *'nlp\_statistics.csv'*.
* The average of these statistics is computed and saved in *'aggregated\_results.csv'*.

**According to Task 1 – Generation Data Science and ML Bullets 3 & 4: Java Web Application for NLP Analysis**

Project Configuration

* IDE: Spring Tool Suite 4
* Spring Boot Version: 3.2.2
* Java Version: JDK 17
* Type: Maven
* Packaging: Jar
* Dependency: Spring Web, Thymeleaf, Stanford-corenlp 4.2.0

Application Details

* The Spring Boot application runs on Tomcat at port 8080 (http).
* *index.html*: Accepts text input (file or textbox) and generates NLP statistics.
* *comparison.html*: Displays a comparison of newly generated NLP statistics with pre-aggregated results in a table format.
* The application reads 'aggregated\_results.csv' for comparison purposes. The File Path is given explicitly. You can change the file location in nlp-stats-app-2 => src/main/java => com.example.demo.service => NLPService.java In NLPService.java, change the file path present in the function readAggregatedResults()

Usage & Customization

* Download the project and import it into Spring Tool Suite then ensure the correct file path is set in `NLPService.java` for reading 'aggregated\_results.csv' and run the project as a Spring Boot Application. In your browser access http://localhost:8080/ Follow the instructions on the web page from there on.
* Input can be provided either as a text file or directly as a text.

**GitHub Repository Link:** [NLP Statistics Analyzer](https://github.com/my-link)