

**Data Focussed Python**

Readme File

Universi

[**Group Z2-10**](https://canvas.cmu.edu/courses/31959/groups#)

Ashutosh Choudhari - archoudh

Dibyanshu Patnaik - dpatnaik

Jaineet Shah - jaineets

Namit Bansal - namitb

Niyati Mittal - niyatim

# Introduction

# We are using three data sources for the purpose of this project.

1. The Books Dataset has been extracted from amazon using Beautiful Soup.
2. The Youtube Videos Dataset has been extracted using the Youtube API.
3. The Jobs Dataset has been extracted from the Naukri.com website using Selenium.

The purpose of this project is to help the data science enthusiasts to be able to search for books, jobs and youtube videos related to but not limited to data science. The user can use our easy to use GUI to enter some search terms related to Data Science ( such as R Programming, Python) and can get recommendations on books, jobs and youtube videos related to the user search query. For the purpose of this project we are using pre extracted/ scraped data details of which are shared in the following sections.

# Instructions

## **Books Dataset**

A raw file called raw\_books.csv is also present in the directory to show the users the raw data that we had extracted which has 924 books. After data cleaning we had created the csv file final\_books\_dataset.csv which has 614 books along with the description for each book extracted again from amazon.

Please use the **final\_books\_dataset.csv present** in the directory to run this project. We will be using this csv file which has already been scraped from amazon.

Live Scraping is possible but it will take a lot of time to extract and create a dataset for 1000 books.

Issues with Live Web Scraping of Data from amazon :

1. The user has to search the type of books he/she wants to extract, and then extract all the information that is present on the result page that is displayed.
2. Now from the result page the user can get the link for each book which he/she can then visit to extract the description for each book. Now, imagine doing that for 1000 books, especially given the fact that amazon is not a government website and scraping data from it doesn’t seem legal.
3. Now if the user still wants to run the code to extract the dataset then he/she should be ready to face some challenges as the amazon tags are routinely updated and a workaround might be needed to extract the data.
4. The user can run the books\_semanticsearch.pywhich uses the already extracted .csv data file mentioned above (final\_books\_dataset.csv**)** to perform a semantic search on the books dataset based on the input query. (We are using a semantic search algorithm to retrieve the most relevant books based on the search term that the user enters)

Remember :

For the current implementation of the project please make use of the **final\_books\_dataset.csv** present in the directory.

## **Youtube Videos Dataset:**

Please use the **youtube\_output\_raw.xls, youtube\_output\_clean.xls and youtube\_output\_caption\_summary.xls** presentin the directory to run this project. These excel files contain the already scraped from **Youtube\_API\_Data\_Scraping.py** and processed summary data from **Summary\_Generator\_Youtube\_Captions.py**.

Live scraping from the Youtube API is possible but it is time consuming to scrape around 900 videos.

1. In order to run **Youtube\_API\_Data\_Scraping.py**, the user needs to authenticate himself as a test user in the google developer console of the admin and hence we have added [jostlund@andrew.cmu.edu](mailto:jostlund@andrew.cmu.edu) and [jkostlund@gmail.com](mailto:jkostlund@gmail.com) as test users. The credentials.json and token4.pickle file is included in the directory for the authentication process. After the authentication, the data scraping for about 900 videos takes about 20-30 mins.
2. The **Summary\_Generator\_Youtube\_Captions.py** generates the summary of the captions of the scraped youtube videos. It uses the **Bart hugging-face** model and therefore it takes around **6-7 hours** to generate the summary of all the videos' captions.

Hence in order to avoid the hassle, the user can run the **Youtube\_Api\_Data\_Processing.py** which uses the already extracted .xls data files mentioned above to perform a semantic search on the youtube videos dataset based on the input query.

***The main file is GUI.py. The user can directly obtain the GUI with all functionality by running GUI.py. Note that all the files must be present in the same directory.***

## **Jobs DataSet :**

# 

# 

# 

# 

# Requirements and Installation

1. **pip install PySimpleGUI**

Required for implementing the GUI.

1. **pip install pandas**
2. **pip install numpy**
3. **pip install selenium**
4. **pip install gensim**

Required for implementing the tfidf and lsi model for semantic search.

1. **pip install tensorflow**
2. **pip install wordcloud**
3. **pip install spacy**

Required for cleaning and tokenizing the text.

1. **pip install openpyxl**
2. **pip install transformers**

Required for downloading and running the hugging-face Bart model for summary generation.

1. **pip install sentence-transformers**

Includes dependencies for the Bart model.

1. **pip install protobuf==3.19.6**

It is used for compatibility of the Bart model.

1. **pip install youtube\_transcript\_api**
2. **pip install clean-text**
3. **pip install nltk (nltk.download())**
4. **pip install google-api-python-client**

Authentication with youtube API.

1. **pip install beautifulsoup4**
2. **pip install** <https://github.com/explosion/spacy-models/releases/download/en_core_web_sm-3.2.0/en_core_web_sm-3.2.0-py3-none-any.whl> ->

Required for downloading the en\_core\_web\_sm model for processing text.

1. **pip install requests**

1. **conda install -c huggingface transformers-> includes dependencies for the Bart model.**