**Data Science Professional Practicum (DSCI 560)**

**Laboratory Assignment 2**

**Team Name:** Guardians of the Algorithm

**Team members:**

Jaival Chintankumar Upadhyay – 8278442205

Pratham Solanki - 3242692358

Mayank Patil - 9101437684

For our group project, we have selected three key domains: **NLP Forums**, **Health and Lifestyle**, and **Educational Course Materials**. These domains provide diverse and rich datasets that align with our objectives.

1. **NLP Forums**
   * **Dataset Source**: [Hugging Face Discussions](https://discuss.huggingface.co/t/fine-tune-llms-on-pdf-documents/71374)
   * **Description**: Contains threads on fine-tuning large language models with PDF documents.
   * **Sample Excerpt**: *Screenshots attached below*
2. **Health and Lifestyle**
   * **Dataset Source**: [Smokers Health Data on Kaggle](https://www.kaggle.com/datasets/jaceprater/smokers-health-data)
   * **Description**: Includes data on individuals' smoking habits and various health indicators.
   * **Sample Excerpt**: *Screenshots attached below*
3. **Educational Course Materials**
   * **Dataset Source**: [CMU Course Lectures](https://www.cs.cmu.edu/~ninamf/courses/601sp15/lectures.shtml)
   * **Description**: Provides information on machine learning course lectures, topics, and resources.
   * **Sample Excerpt**: *Screenshots attached below*

**Reasoning Behind Topic Choice**: These domains offer a comprehensive mix of technical discussions, real-world health data, and academic resources. This combination allows us to explore machine learning applications, analyze health-related patterns, and understand educational methodologies, providing a well-rounded foundation for our project.

i) CSV or Excel

**Snapshot of Code:**

**A screen shot of a computer program

AI-generated content may be incorrect.**

**SOURCE PAGEA screenshot of a computer

AI-generated content may be incorrect.**

**Description:** The Extract\_CSV\_data function automates the process of downloading the "Smokers Health Data" dataset from Kaggle by using provided API credentials. It begins by creating a kaggle.json file with the necessary authentication details and sets the environment for the Kaggle API. The function then downloads the specified dataset into a temporary directory, extracts the CSV file from the downloaded ZIP archive, and renames it to extracted\_csv.csv for easier access. After successfully saving the CSV file, the function cleans up by removing the temporary files and directories. Finally, it performs basic data operations by loading the CSV into a pandas DataFrame and printing out key information such as column names, dataset shape, null values, and the last few rows, providing an initial overview of the dataset for further analysis.

**OUTPUT:**

**A computer screen with white text

AI-generated content may be incorrect.**

**A screen shot of a computer

AI-generated content may be incorrect.**

**OUTPUT CSV FILEA screenshot of a computer

AI-generated content may be incorrect.**

In the above images, we see that the data has successfully been extracted and is stored on our desktop

ii) ASCII Texts like Forum Postings and HTML

**Snapshot of Code:**

A screenshot of a computer program

AI-generated content may be incorrect.

**Description:**

First, we are fetching the HTML source code using Selenium from the specified URL: https://discuss.huggingface.co/t/fine-tune-llms-on-pdf-documents/71374. This allows us to gather the complete HTML structure of the webpage for further processing.

After fetching and parsing, the HTML content is saved into a file named html\_parsed.html for record-keeping and easier access.

Next, we read the html\_parsed.html file to extract data. Using Beautiful Soup, we extract the following:

* Title of the forum post.
* Page statistics, such as the number of 'views', 'likes', 'links', and 'users'.
* For each post in the forum:
  + time\_stamp
  + name/author of the post
  + post content
  + likes received on the post

All the extracted data is then written into a structured CSV file for further use and analysis.

Finally, we read the generated CSV file using Pandas and perform some basic operations, such as inspecting the data, print the column names, shape, null value counts, and the first few rows of the dataset.

**Source Website:** "https://discuss.huggingface.co/t/fine-tune-llms-on-pdf-documents/71374"

A screenshot of a computer

AI-generated content may be incorrect.

A screenshot of a computer

AI-generated content may be incorrect.

**OUTPUT**

A computer screen with text on it

AI-generated content may be incorrect.A computer screen with text

AI-generated content may be incorrect.

**OUTPUT CSV FILE**

A screenshot of a computer

AI-generated content may be incorrect.A close up of a text

AI-generated content may be incorrect.

iii. PDF and Word Documents that require conversion and OCR

**Code Snippet: A screen shot of a computer program

AI-generated content may be incorrect.**

**Description:** The code automates the collection and processing of course-related data from the webpage located at https://www.cs.cmu.edu/~ninamf/courses/601sp15/lectures.shtml. The operation begins with setting up the environment and creating essential classes to manage data and file paths. The Path class ensures that directories are created correctly for storing the extracted data and PDFs, while the Settings class defines constants like the base URL and the target page for scraping. By leveraging pydantic, a data structure is established for topics, readings, handouts, and lectures to maintain consistency and clarity.

The script includes utility functions for managing the content of the webpage. First, it detects and resolves relative URLs, ensuring that all links are correctly connected. The scrape\_course\_page function fetches the HTML content from the designated webpage using requests and parses it with Beautiful Soup. It collects relevant information from the schedule table, including lecture titles, topics, readings, and handouts, while disregarding unnecessary rows. For each lecture, structured data is created using the CourseItem class, guaranteeing that all fields are well-organized.

Once the lecture data is gathered, it is stored in a CSV file (ml.csv) for future utilization. The script further processes this data to handle lecture handouts, particularly PDFs. The read\_pdf\_from\_url function retrieves PDFs from the given URLs, extracts their text with fitz, and saves the text as .txt files, one for each page. These files are organized in a structured directory hierarchy based on lecture names for easy navigation.

The final step involves reviewing the saved data to process all PDFs labeled as "Slides" in the handouts. A progress bar from tqdm provides visual updates throughout this process, enhancing the user experience. Error handling is incorporated into the script to ensure resilience against unexpected problems, such as network disruptions or missing elements on the webpage.

In summary, this code streamlines the process of scraping course information, formatting it as structured CSV files, and extracting text from lecture PDFs for further analysis. The approach guarantees clarity, maintainability, and usability, making it highly suitable for academic data management tasks.

**OUTPUT:**

**A screenshot of a computer

AI-generated content may be incorrect. A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.A screenshot of a computer

AI-generated content may be incorrect.**

**A screenshot of a computer

AI-generated content may be incorrect.**

**Question:** In the report, describe what the script does (conversion tasks and tools to keep only the relevant data) to create a clean single dataset. While there are a lot of attempts to build realistic chatbots, most people would rather speak to a real person because their capabilities are very limited. Describe what might be missing in these existing chatbots. Discuss how your dataset might improve the overall performance and correctness.

**Answer:** Many chatbots still miss the mark, making people prefer talking to real humans. They often struggle with complex questions, use outdated information, and give generic answers. Our project fixes these issues by using recent, specialized data focused on Machine Learning applications instead of general knowledge. By integrating up-to-date discussions from NLP forums, detailed health data, and educational materials, our chatbot can better understand and accurately respond to technical queries. This makes our chatbot more reliable and personalized, offering users more relevant and trustworthy interactions in machine learning and healthcare.