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

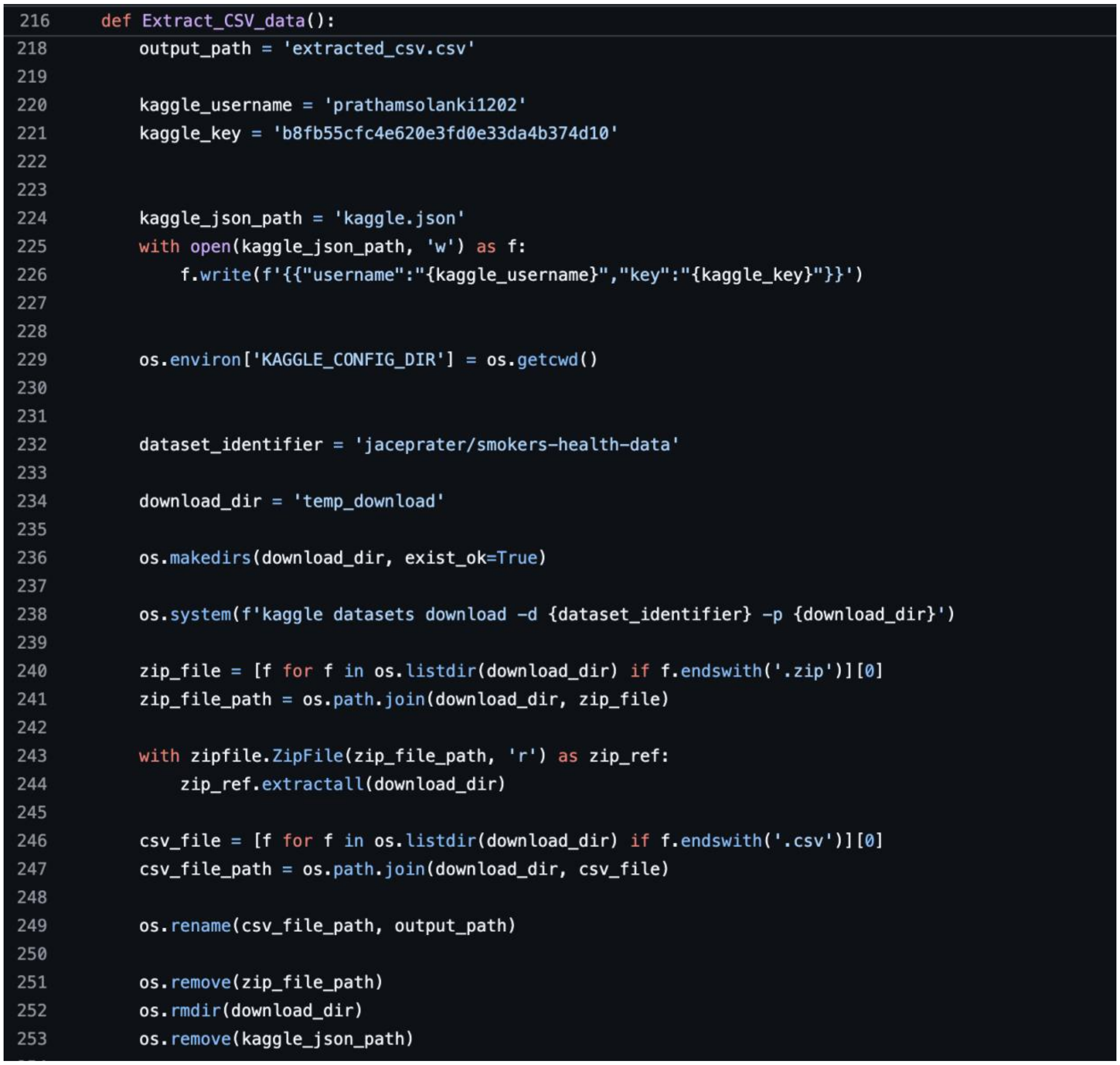
For our group project, we have selected three key domains: **NLP Forums**, **Health and Lifestyle**, and **Educational Course Materials**. These domains provide diverse and rich datasets that align with our objectives.

1. **NLP Forums**
   - **Dataset Source**: Hugging Face Discussions
   - **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
   - **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
   - **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.

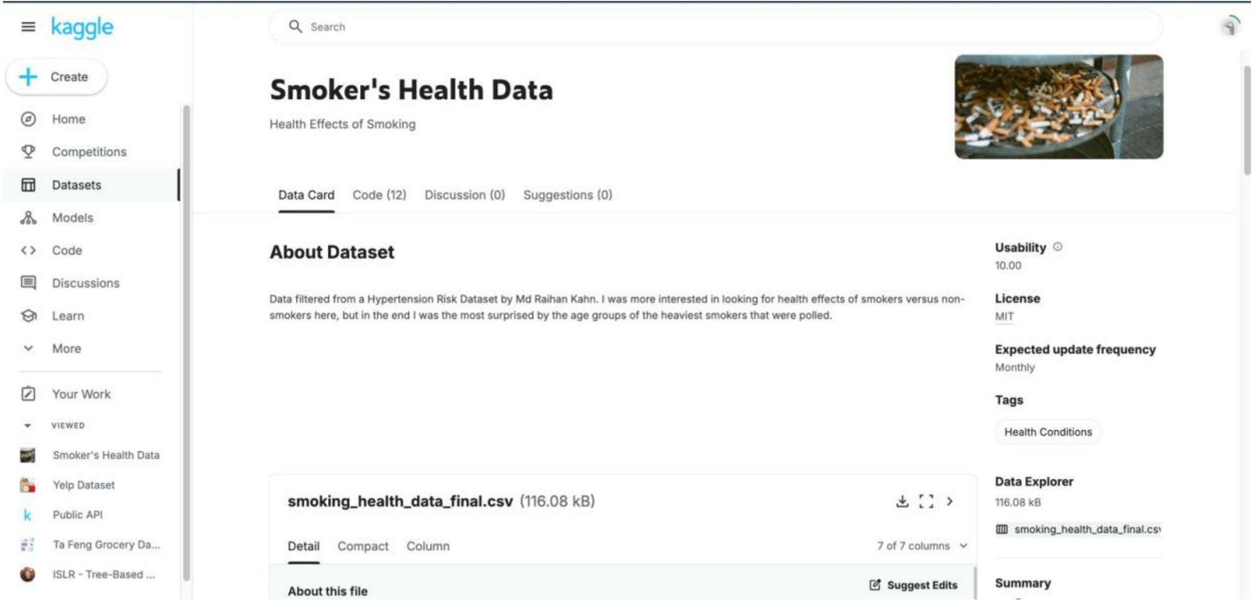
i) CSV or Excel

Snapshot of Code:



```
def Extract_CSV_data():
    output_path = 'extracted_csv.csv'

    kaggle_username = 'prathamsolanki1202'
    kaggle_key = 'b8fb55cfc4e620e3fd0e33da4b374d10'


    kaggle_json_path = 'kaggle.json'
    with open(kaggle_json_path, 'w') as f:
        f.write(f'{{"username":"{kaggle_username}","key":"{kaggle_key}"}}')


    os.environ['KAGGLE_CONFIG_DIR'] = os.getcwd()


    dataset_identifier = 'jaceprater/smokers-health-data'

    download_dir = 'temp_download'

    os.makedirs(download_dir, exist_ok=True)

    os.system(f'kaggle datasets download -d {dataset_identifier} -p {download_dir}')

    zip_file = [f for f in os.listdir(download_dir) if f.endswith('.zip')][0]
    zip_file_path = os.path.join(download_dir, zip_file)

    with zipfile.ZipFile(zip_file_path, 'r') as zip_ref:
        zip_ref.extractall(download_dir)

    csv_file = [f for f in os.listdir(download_dir) if f.endswith('.csv')][0]
    csv_file_path = os.path.join(download_dir, csv_file)

    os.rename(csv_file_path, output_path)

    os.remove(zip_file_path)
    os.rmdir(download_dir)
    os.remove(kaggle_json_path)
```

SOURCE PAGE



Smoker's Health Data

Health Effects of Smoking

About Dataset

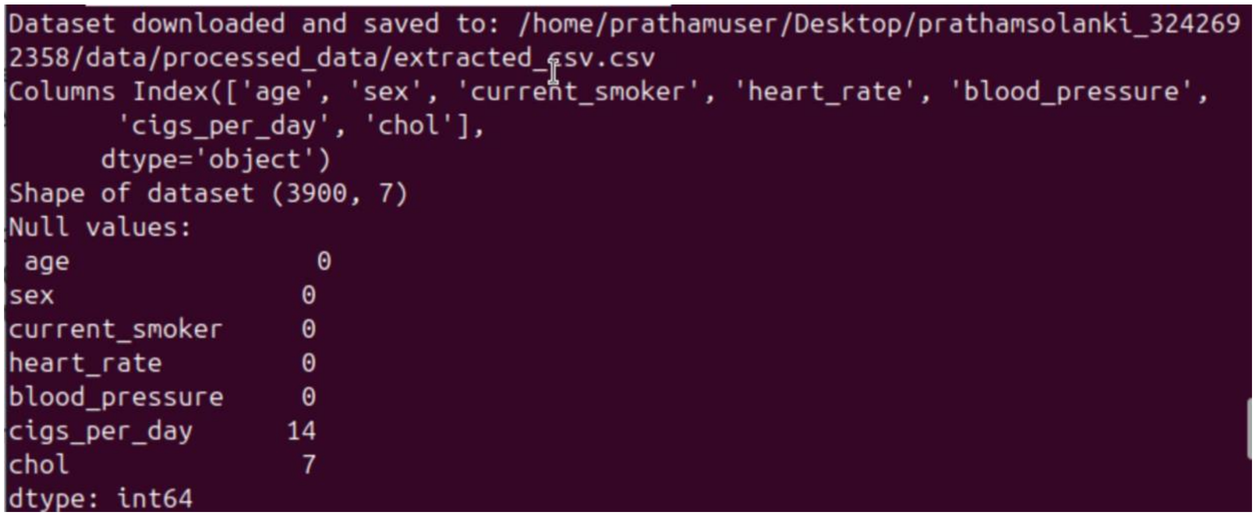
Data filtered from a Hypertension Risk Dataset by Md Raihan Kahn. I was more interested in looking for health effects of smokers versus non-smokers here, but in the end I was the most surprised by the age groups of the heaviest smokers that were polled.

smoking_health_data_final.csv (116.08 kB)

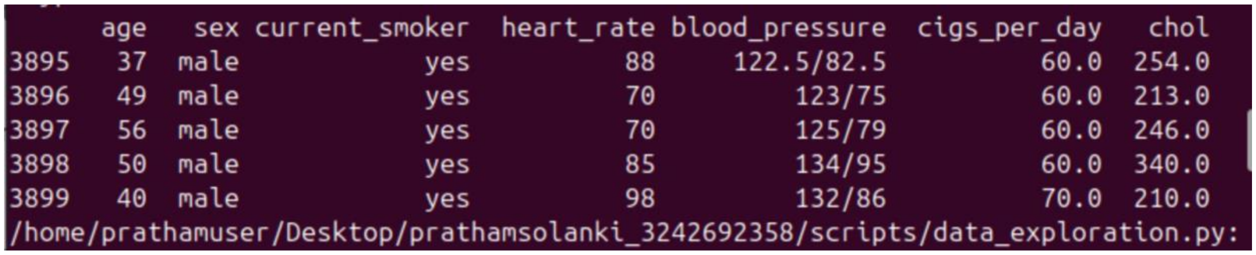
About this file

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:

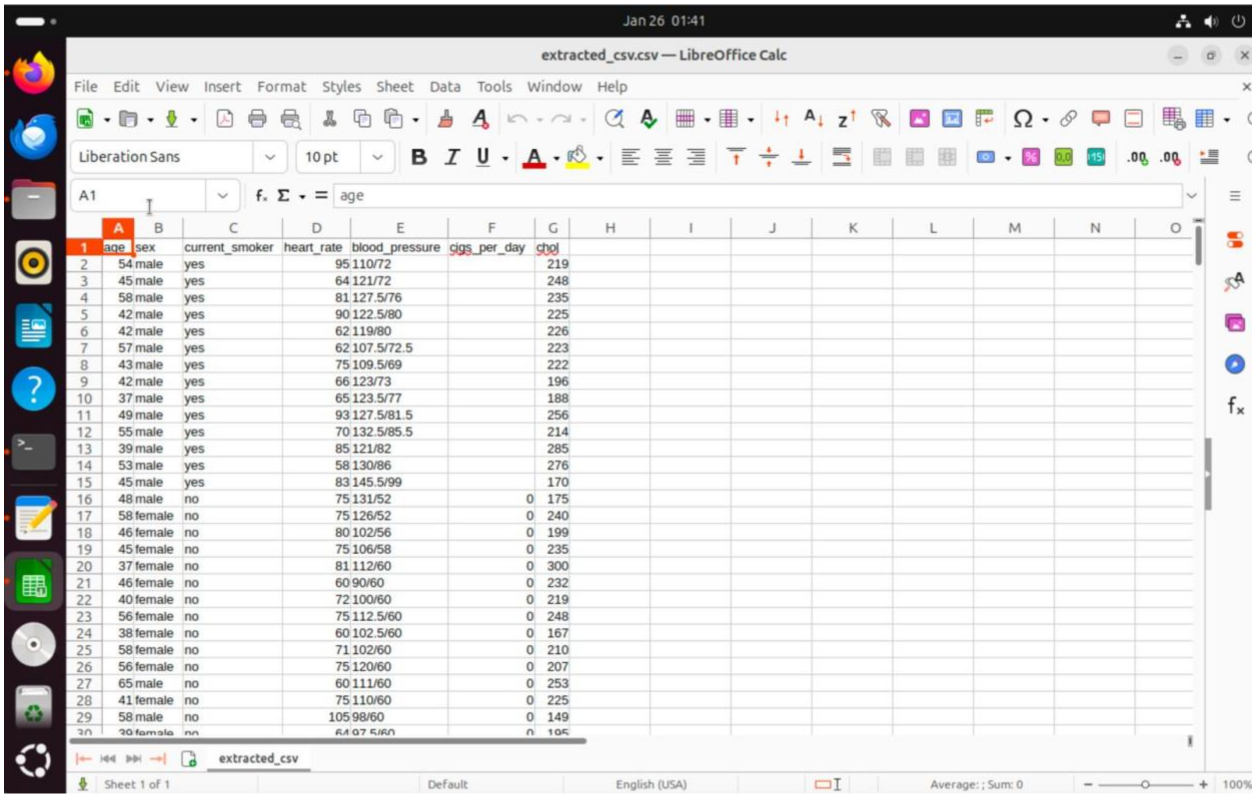


```
Dataset downloaded and saved to: /home/prathamuser/Desktop/prathamsolanki_324269
2358/data/processed_data/extracted_csv.csv
Columns Index(['age', 'sex', 'current_smoker', 'heart_rate', 'blood_pressure',
       'cigs_per_day', 'chol'],
      dtype='object')
Shape of dataset (3900, 7)
Null values:
 age                0
sex                0
current_smoker     0
heart_rate         0
blood_pressure     0
cigs_per_day      14
chol               7
dtype: int64
```



```
      age    sex current_smoker  heart_rate blood_pressure  cigs_per_day   chol
3895   37   male            yes          88     122.5/82.5          60.0  254.0
3896   49   male            yes          70         123/75          60.0  213.0
3897   56   male            yes          70         125/79          60.0  246.0
3898   50   male            yes          85         134/95          60.0  340.0
3899   40   male            yes          98         132/86          70.0  210.0
/home/prathamuser/Desktop/prathamsolanki_3242692358/scripts/data_exploration.py:
```

OUTPUT CSV FILE



Jan 26 01:41

extracted_csv.csv — LibreOffice Calc

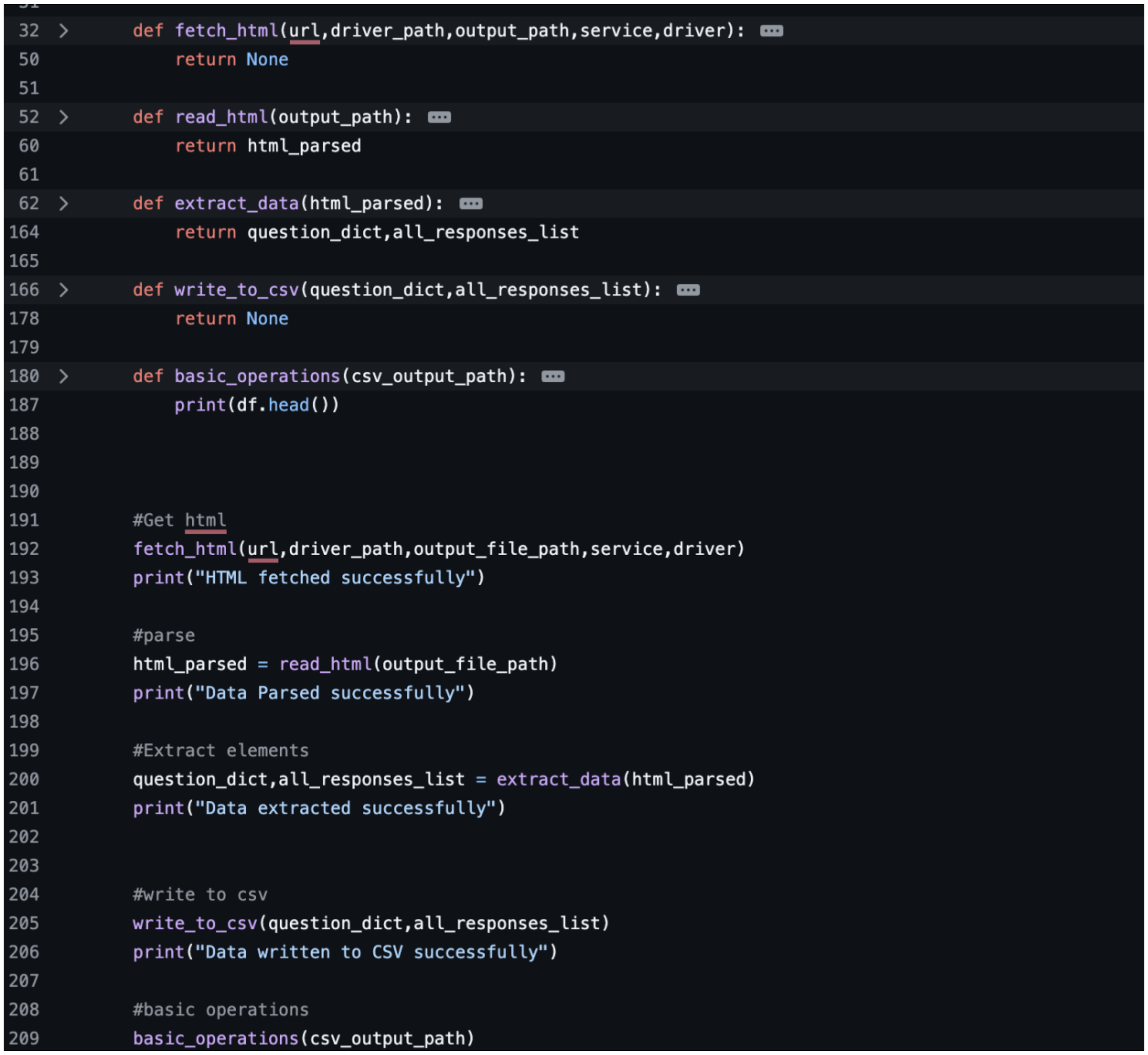
	A	B	C	D	E	F	G
1	age	sex	current_smoker	heart_rate	blood_pressure	cigs_per_day	chol
2	54	male	yes	95	110/72		219
3	45	male	yes	64	121/72		248
4	58	male	yes	81	127.5/76		235
5	42	male	yes	90	122.5/80		225
6	42	male	yes	62	119/80		226
7	57	male	yes	62	107.5/72.5		223
8	43	male	yes	75	109.5/69		222
9	42	male	yes	66	123/73		196
10	37	male	yes	65	123.5/77		188
11	49	male	yes	93	127.5/81.5		256
12	55	male	yes	70	132.5/85.5		214
13	39	male	yes	85	121/82		285
14	53	male	yes	58	130/86		276
15	45	male	yes	83	145.5/99		170
16	48	male	no	75	131/52	0	175
17	58	female	no	75	126/52	0	240
18	46	female	no	80	102/56	0	199
19	45	female	no	75	106/58	0	235
20	37	female	no	81	112/60	0	300
21	46	female	no	60	90/60	0	232
22	40	female	no	72	100/60	0	219
23	56	female	no	75	112.5/60	0	248
24	38	female	no	60	102.5/60	0	167
25	58	female	no	71	102/60	0	210
26	56	female	no	75	120/60	0	207
27	65	male	no	60	111/60	0	253
28	41	female	no	75	110/60	0	225
29	58	male	no	105	98/60	0	149
30	39	female	no	64	97.5/60	0	195

Sheet 1 of 1

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:



```
def fetch_html(url,driver_path,output_path,service,driver): ...
    return None

def read_html(output_path): ...
    return html_parsed

def extract_data(html_parsed): ...
    return question_dict,all_responses_list

def write_to_csv(question_dict,all_responses_list): ...
    return None

def basic_operations(csv_output_path): ...
    print(df.head())



#Get html
fetch_html(url,driver_path,output_file_path,service,driver)
print("HTML fetched successfully")

#parse
html_parsed = read_html(output_file_path)
print("Data Parsed successfully")

#Extract elements
question_dict,all_responses_list = extract_data(html_parsed)
print("Data extracted successfully")


#write to csv
write_to_csv(question_dict,all_responses_list)
print("Data written to CSV successfully")

#basic operations
basic_operations(csv_output_path)
```

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.

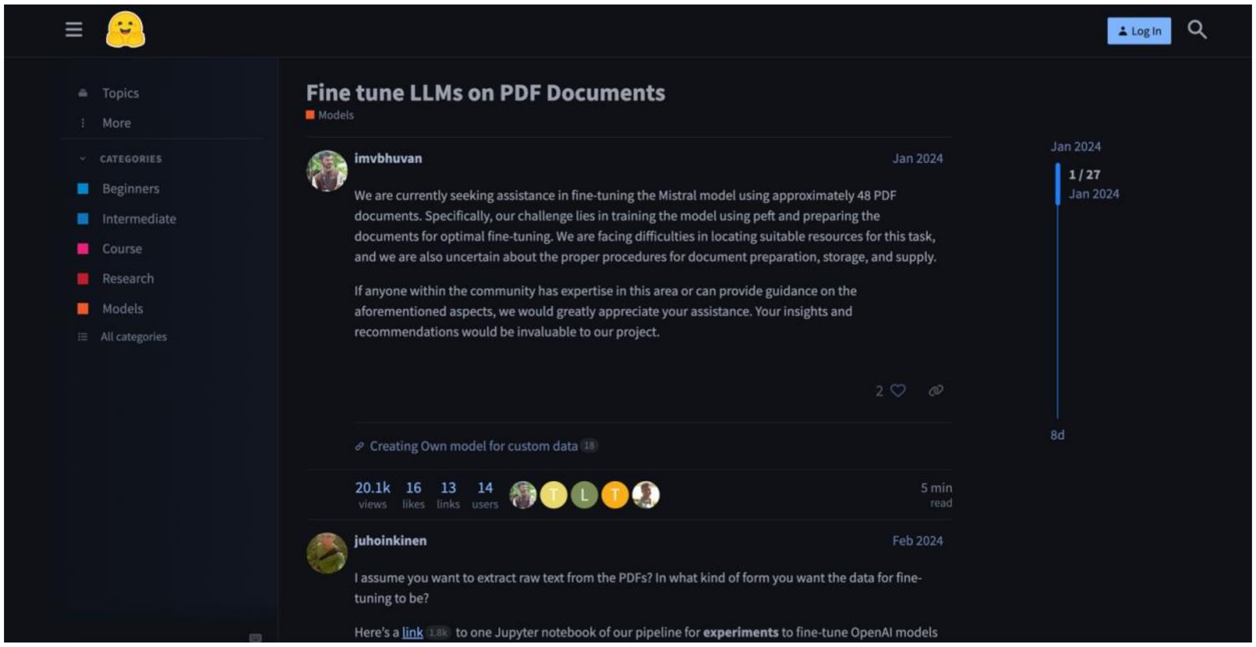
Next, we read the html_parsed.html file to extract data. Using Beautiful Soup, we extract the following:

- 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"



Fine tune LLMs on PDF Documents
Models

imvbhuvan Jan 2024

We are currently seeking assistance in fine-tuning the Mistral model using approximately 48 PDF documents. Specifically, our challenge lies in training the model using peft and preparing the documents for optimal fine-tuning. We are facing difficulties in locating suitable resources for this task, and we are also uncertain about the proper procedures for document preparation, storage, and supply.

If anyone within the community has expertise in this area or can provide guidance on the aforementioned aspects, we would greatly appreciate your assistance. Your insights and recommendations would be invaluable to our project.

2

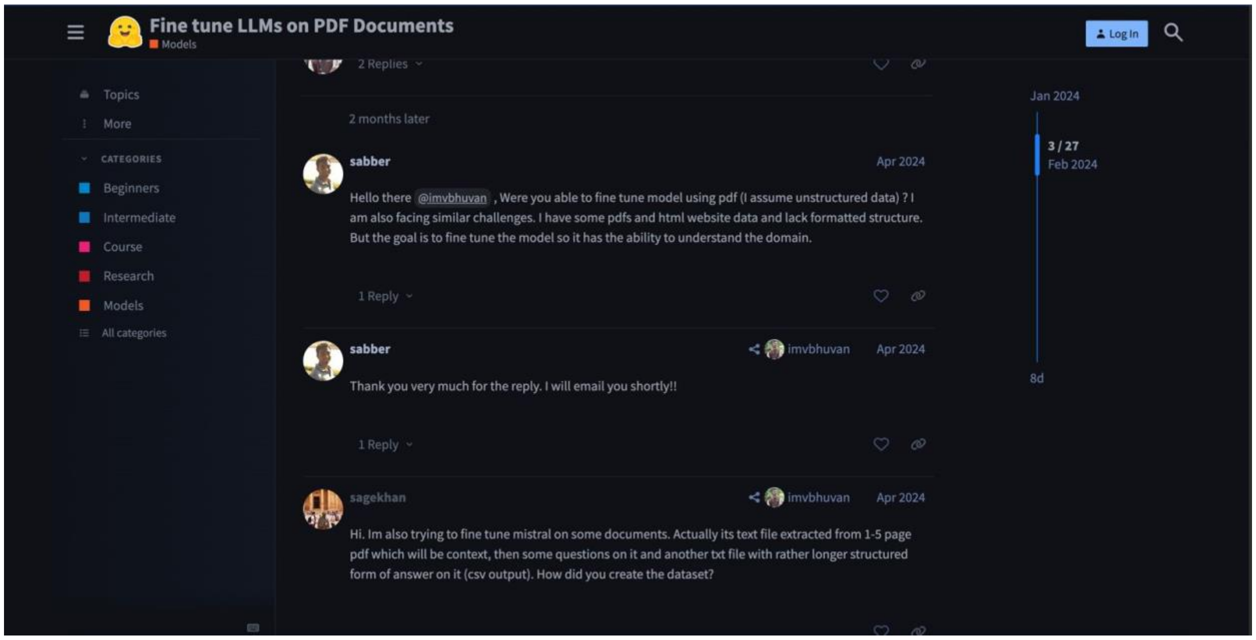
Creating Own model for custom data 18

20.1k views 16 likes 13 links 14 users 5 min read

juhoinkinen Feb 2024

I assume you want to extract raw text from the PDFs? In what kind of form you want the data for fine-tuning to be?

Here's a link 1.8k to one Jupyter notebook of our pipeline for **experiments** to fine-tune OpenAI models



Fine tune LLMs on PDF Documents
Models

2 Replies

2 months later

sabber Apr 2024

Hello there @imvbhuvan , Were you able to fine tune model using pdf (I assume unstructured data) ? I am also facing similar challenges. I have some pdfs and html website data and lack formatted structure. But the goal is to fine tune the model so it has the ability to understand the domain.

1 Reply

sabber imvbhuvan Apr 2024

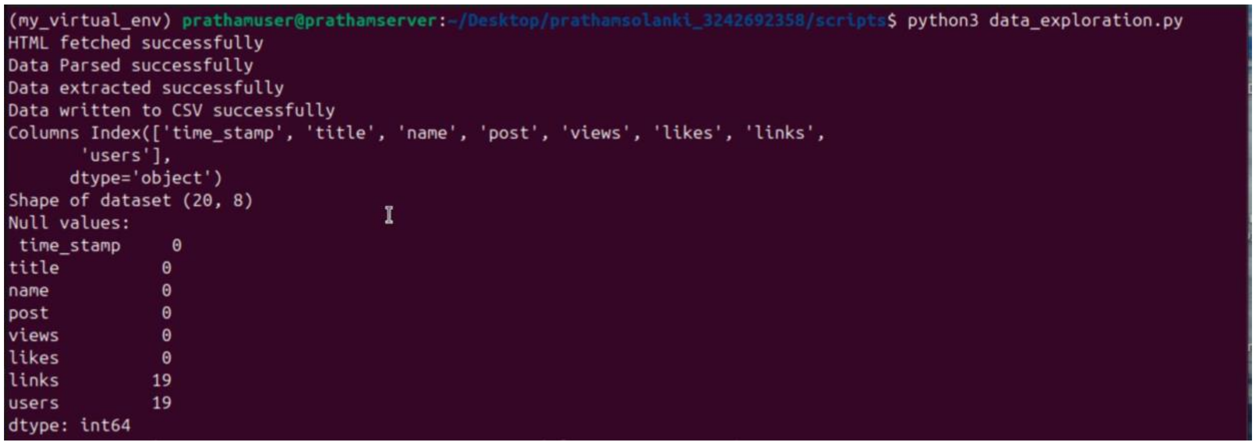
Thank you very much for the reply. I will email you shortly!!

1 Reply

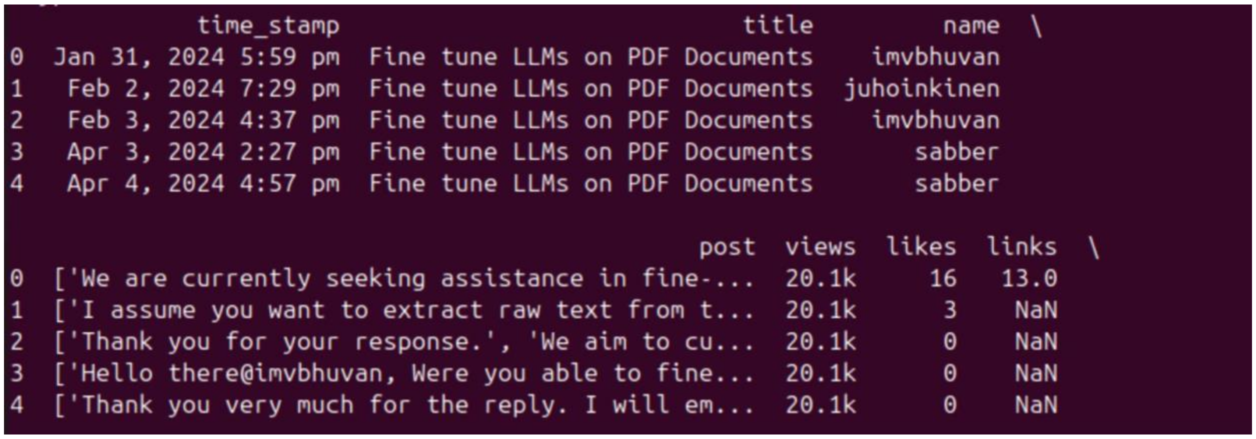
sagekhan imvbhuvan Apr 2024

Hi. Im also trying to fine tune mistral on some documents. Actually its text file extracted from 1-5 page pdf which will be context, then some questions on it and another txt file with rather longer structured form of answer on it (csv output). How did you create the dataset?

OUTPUT

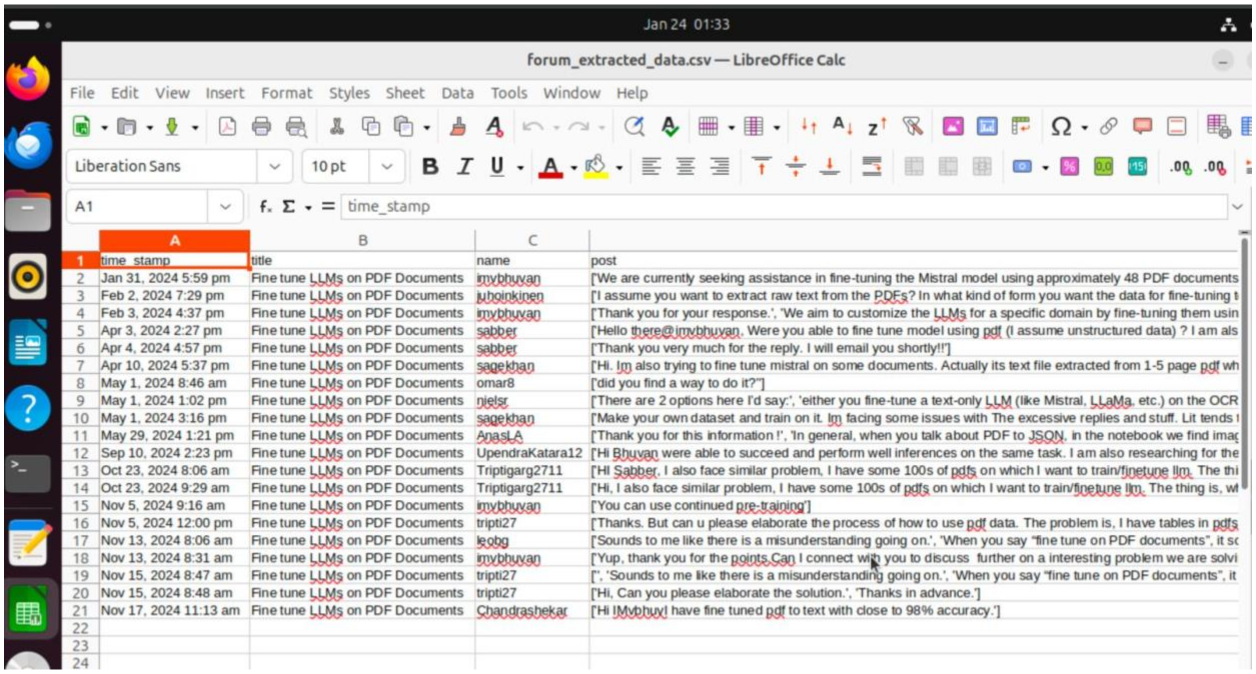


```
(my_virtual_env) prathamuser@prathamserver:~/Desktop/prathamsolanki_3242692358/scripts$ python3 data_exploration.py
HTML fetched successfully
Data Parsed successfully
Data extracted successfully
Data written to CSV successfully
Columns Index(['time_stamp', 'title', 'name', 'post', 'views', 'likes', 'links',
       'users'],
      dtype='object')
Shape of dataset (20, 8)
Null values:
 time_stamp     0
title         0
name          0
post          0
views         0
likes         0
links        19
users        19
dtype: int64
```



```
              time_stamp                                 title         name  \
0  Jan 31, 2024 5:59 pm  Fine tune LLMs on PDF Documents    imvbhuvan
1   Feb 2, 2024 7:29 pm  Fine tune LLMs on PDF Documents  juhoinkinen
2   Feb 3, 2024 4:37 pm  Fine tune LLMs on PDF Documents    imvbhuvan
3   Apr 3, 2024 2:27 pm  Fine tune LLMs on PDF Documents       sabber
4   Apr 4, 2024 4:57 pm  Fine tune LLMs on PDF Documents       sabber

                                               post  views  likes  links  \
0  ['We are currently seeking assistance in fine-...  20.1k     16   13.0
1  ['I assume you want to extract raw text from t...  20.1k      3    NaN
2  ['Thank you for your response.', 'We aim to cu...  20.1k      0    NaN
3  ['Hello there@imvbhuvan, Were you able to fine...  20.1k      0    NaN
4  ['Thank you very much for the reply. I will em...  20.1k      0    NaN
```

OUTPUT CSV FILE

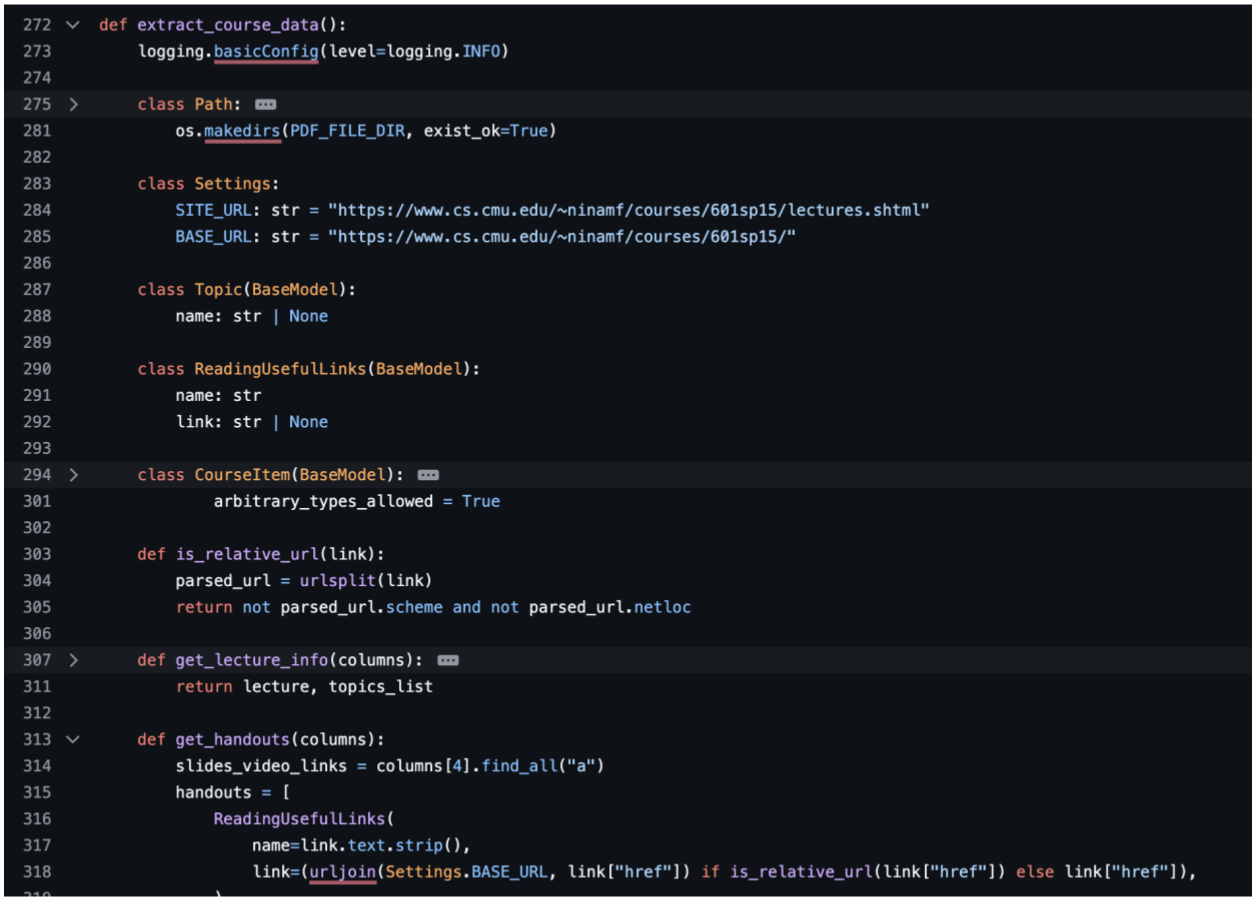


forum_extracted_data.csv — LibreOffice Calc

time_stamp	title	name	post
Jan 31, 2024 5:59 pm	Fine tune LLMs on PDF Documents	imvbhuvan	['We are currently seeking assistance in fine-tuning the Mistral model using approximately 48 PDF documents
Feb 2, 2024 7:29 pm	Fine tune LLMs on PDF Documents	juhoinkinen	['I assume you want to extract raw text from the PDFs? In what kind of form you want the data for fine-tuning t
Feb 3, 2024 4:37 pm	Fine tune LLMs on PDF Documents	imvbhuvan	['Thank you for your response.', 'We aim to customize the LLMs for a specific domain by fine-tuning them usin
Apr 3, 2024 2:27 pm	Fine tune LLMs on PDF Documents	sabber	['Hello there@imvbhuvan. Were you able to fine tune model using pdf (I assume unstructured data) ? I am als
Apr 4, 2024 4:57 pm	Fine tune LLMs on PDF Documents	sabber	['Thank you very much for the reply. I will email you shortly!!']
Apr 10, 2024 5:37 pm	Fine tune LLMs on PDF Documents	sagekhan	['Hi. Im also trying to fine tune mistral on some documents. Actually its text file extracted from 1-5 page pdf wh
May 1, 2024 8:46 am	Fine tune LLMs on PDF Documents	omar8	['did you find a way to do it?']
May 1, 2024 1:02 pm	Fine tune LLMs on PDF Documents	nielsr	['There are 2 options here I'd say:', 'either you fine-tune a text-only LLM (like Mistral, LLaMa, etc.) on the OCR
May 1, 2024 3:16 pm	Fine tune LLMs on PDF Documents	sagekhan	['Make your own dataset and train on it. Im facing some issues with The excessive replies and stuff. Lit tends t
May 29, 2024 1:21 pm	Fine tune LLMs on PDF Documents	AnasLA	['Thank you for this information !', 'In general, when you talk about PDF to JSON, in the notebook we find imag
Sep 10, 2024 2:23 pm	Fine tune LLMs on PDF Documents	UpendraKatara12	['Hi Bhuvan were able to succeed and perform well inferences on the same task. I am also researching for the
Oct 23, 2024 8:06 am	Fine tune LLMs on PDF Documents	Triptigarg2711	['HI Sabber, I also face similar problem, I have some 100s of pdfs on which I want to train/finetune llm. The thi
Oct 23, 2024 9:29 am	Fine tune LLMs on PDF Documents	Triptigarg2711	['Hi, I also face similar problem, I have some 100s of pdfs on which I want to train/finetune llm. The thing is, wh
Nov 5, 2024 9:16 am	Fine tune LLMs on PDF Documents	imvbhuvan	['You can use continued pre-training']
Nov 5, 2024 12:00 pm	Fine tune LLMs on PDF Documents	tripti27	['Thanks. But can u please elaborate the process of how to use pdf data. The problem is, I have tables in pdfs
Nov 13, 2024 8:06 am	Fine tune LLMs on PDF Documents	leobg	['Sounds to me like there is a misunderstanding going on.', 'When you say "fine tune on PDF documents", it sc
Nov 13, 2024 8:31 am	Fine tune LLMs on PDF Documents	imvbhuvan	['Yup, thank you for the points.Can I connect with you to discuss further on a interesting problem we are solvi
Nov 15, 2024 8:47 am	Fine tune LLMs on PDF Documents	tripti27	['', 'Sounds to me like there is a misunderstanding going on.', 'When you say "fine tune on PDF documents", it
Nov 15, 2024 8:48 am	Fine tune LLMs on PDF Documents	tripti27	['Hi, Can you please elaborate the solution.', 'Thanks in advance.']
Nov 17, 2024 11:13 am	Fine tune LLMs on PDF Documents	Chandrashekar	['Hi IMvbhuvI have fine tuned pdf to text with close to 98% accuracy.']

iii. PDF and Word Documents that require conversion and OCR

Code Snippet:



```
def extract_course_data():
    logging.basicConfig(level=logging.INFO)

    class Path: ...
        os.makedirs(PDF_FILE_DIR, exist_ok=True)

    class Settings:
        SITE_URL: str = "https://www.cs.cmu.edu/~ninamf/courses/601sp15/lectures.shtml"
        BASE_URL: str = "https://www.cs.cmu.edu/~ninamf/courses/601sp15/"

    class Topic(BaseModel):
        name: str | None

    class ReadingUsefulLinks(BaseModel):
        name: str
        link: str | None

    class CourseItem(BaseModel): ...
            arbitrary_types_allowed = True

    def is_relative_url(link):
        parsed_url = urlsplit(link)
        return not parsed_url.scheme and not parsed_url.netloc

    def get_lecture_info(columns): ...
        return lecture, topics_list

    def get_handouts(columns):
        slides_video_links = columns[4].find_all("a")
        handouts = [
            ReadingUsefulLinks(
                name=link.text.strip(),
                link=(urljoin(Settings.BASE_URL, link["href"]) if is_relative_url(link["href"]) else link["href"]),
```

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.

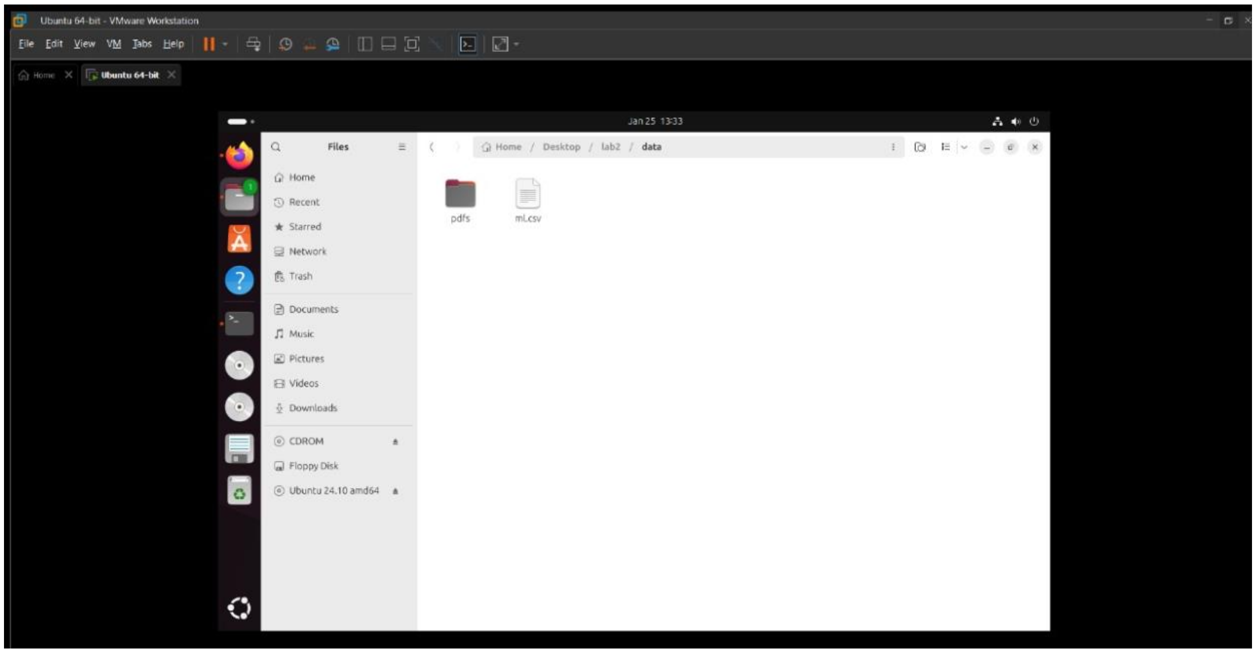
The script includes utility functions for managing the content of the webpage. First, it detects and resolves relative URLs, ensuring that all links are correctly connected. The scrape_course_page function fetches the HTML content from the designated webpage using requests and parses it with Beautiful 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.

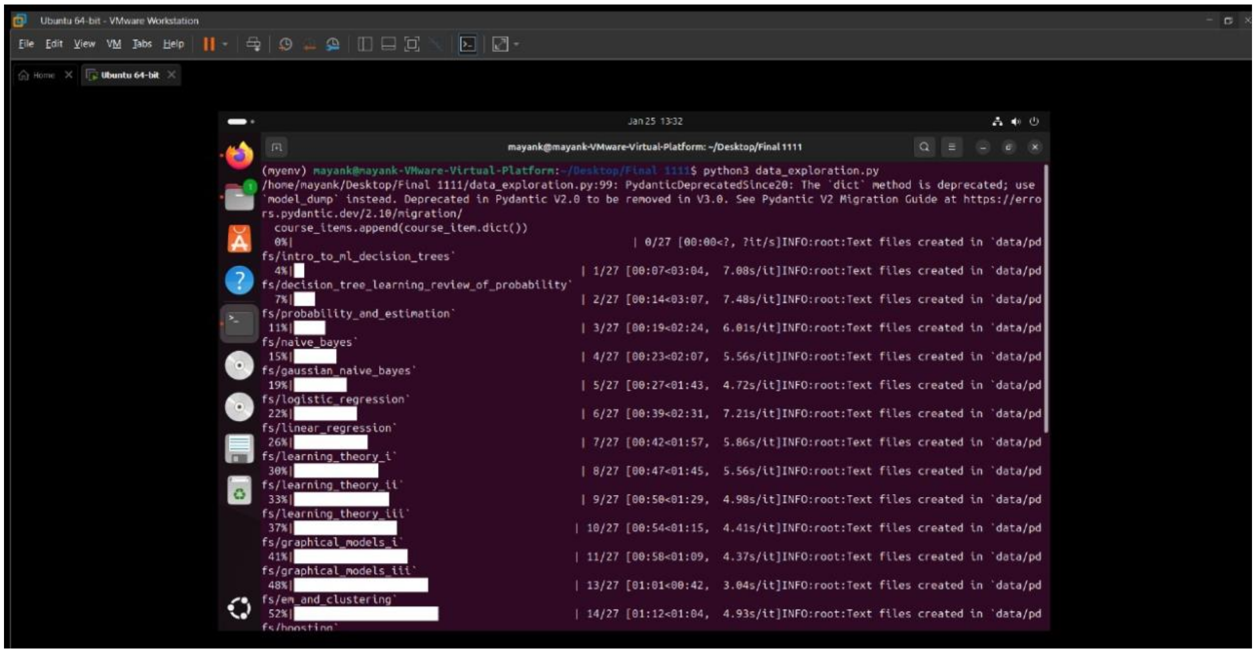
Once the lecture data is gathered, it is stored in a CSV file (ml.csv) for future utilization. The script further processes this data to handle lecture handouts, particularly PDFs. The read_pdf_from_url function retrieves PDFs from the given URLs, extracts their text with fitz, and saves the text as .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.

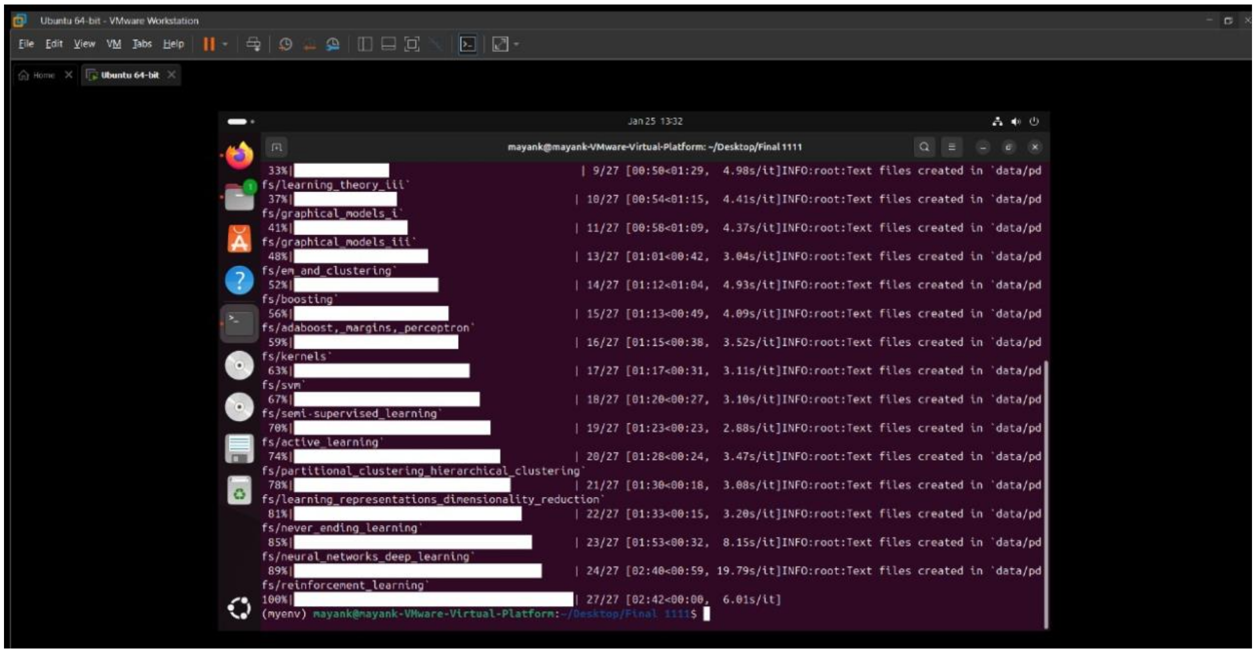
In summary, this code streamlines the process of scraping course information, formatting it as structured CSV files, and extracting text from lecture PDFs for further analysis. The approach guarantees clarity, maintainability, and usability, making it highly suitable for academic data management tasks.

OUTPUT:

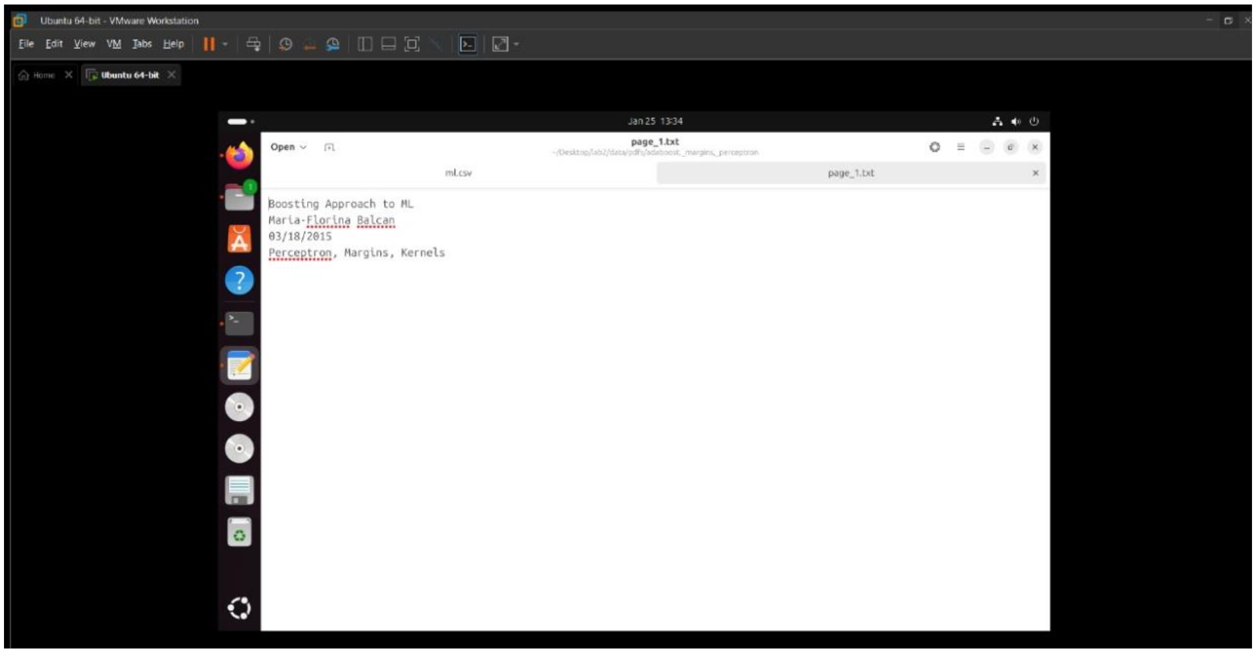



Ubuntu 64-bit - VMware Workstation
File Edit View VM Tabs Help
Home
Ubuntu 64-bit
Jan 25 13:32
mayank@mayank-VMware-Virtual-Platform: ~/Desktop/Final 1111

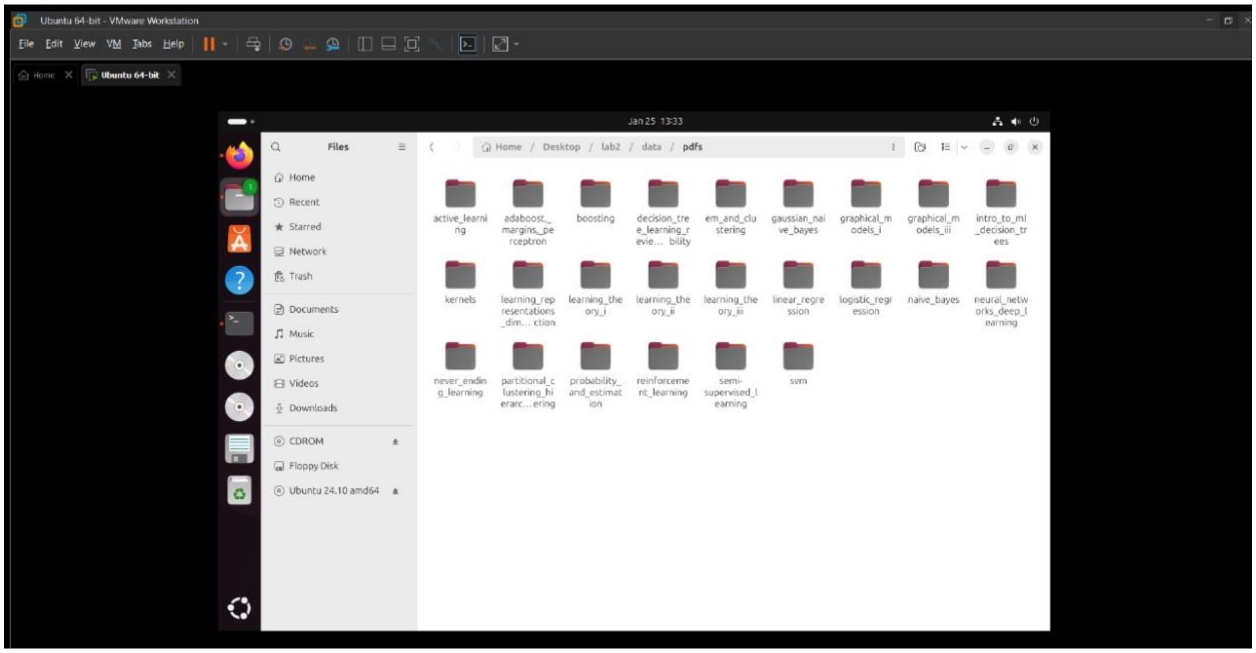
```
(myenv) mayank@mayank-VMware-Virtual-Platform:~/Desktop/Final 1111$ python3 data_exploration.py
/home/mayank/Desktop/Final 1111/data_exploration.py:99: PydanticDeprecatedSince20: The `dict` method is deprecated; use
`model_dump` instead. Deprecated in Pydantic V2.0 to be removed in V3.0. See Pydantic V2 Migration Guide at https://erro
rs.pydantic.dev/2.10/migration/
  course_items.append(course_item.dict())
  0%|                                          | 0/27 [00:00<?, ?it/s]INFO:root:Text files created in `data/pd
fs/intro_to_ml_decision_trees`
  4%|█                              | 1/27 [00:07<03:04,  7.08s/it]INFO:root:Text files created in `data/pd
fs/decision_tree_learning_review_of_probability`
  7%|██                             | 2/27 [00:14<03:07,  7.48s/it]INFO:root:Text files created in `data/pd
fs/probability_and_estimation`
 11%|███                            | 3/27 [00:19<02:24,  6.01s/it]INFO:root:Text files created in `data/pd
fs/naive_bayes`
 15%|████                           | 4/27 [00:23<02:07,  5.56s/it]INFO:root:Text files created in `data/pd
fs/gaussian_naive_bayes`
 19%|█████                          | 5/27 [00:27<01:43,  4.72s/it]INFO:root:Text files created in `data/pd
fs/logistic_regression`
 22%|██████                         | 6/27 [00:39<02:31,  7.21s/it]INFO:root:Text files created in `data/pd
fs/linear_regression`
 26%|███████                        | 7/27 [00:42<01:57,  5.86s/it]INFO:root:Text files created in `data/pd
fs/learning_theory_i`
 30%|████████                       | 8/27 [00:47<01:45,  5.56s/it]INFO:root:Text files created in `data/pd
fs/learning_theory_ii`
 33%|█████████                      | 9/27 [00:50<01:29,  4.98s/it]INFO:root:Text files created in `data/pd
fs/learning_theory_iii`
 37%|██████████                    | 10/27 [00:54<01:15,  4.41s/it]INFO:root:Text files created in `data/pd
fs/graphical_models_i`
 41%|███████████                   | 11/27 [00:58<01:09,  4.37s/it]INFO:root:Text files created in `data/pd
fs/graphical_models_iii`
 48%|█████████████                 | 13/27 [01:01<00:42,  3.04s/it]INFO:root:Text files created in `data/pd
fs/em_and_clustering`
 52%|██████████████                | 14/27 [01:12<01:04,  4.93s/it]INFO:root:Text files created in `data/pd
fs/boosting`
```

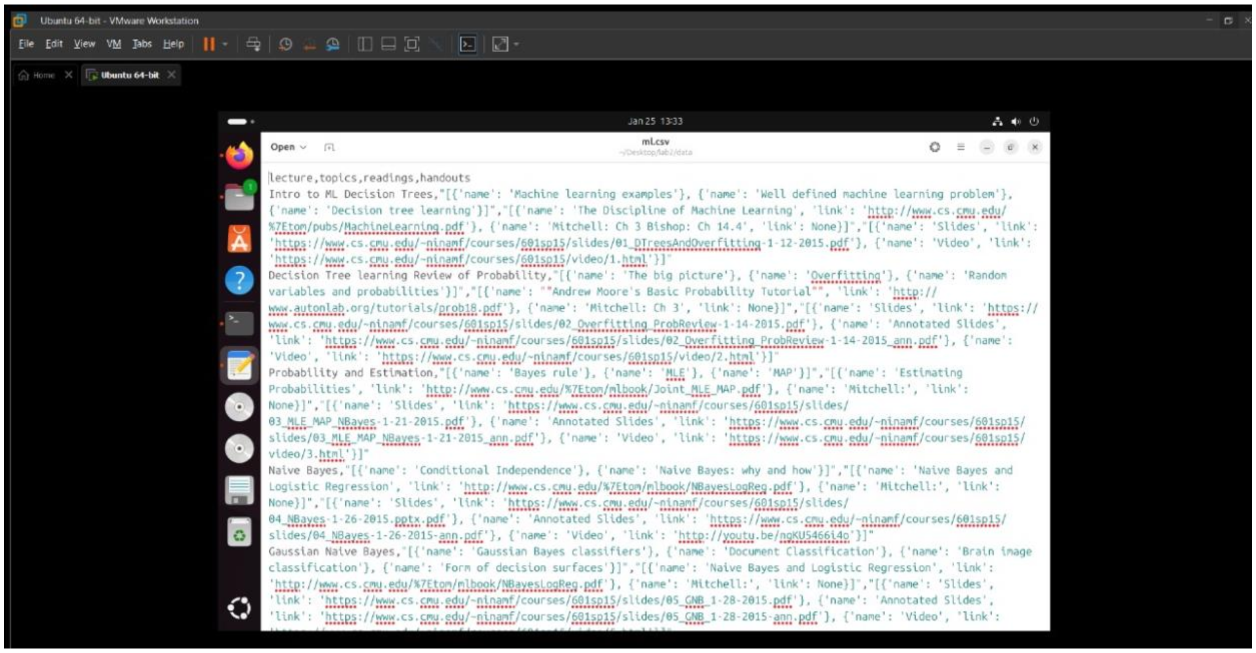

Ubuntu 64-bit - VMware Workstation
File Edit View VM Tabs Help
Home
Ubuntu 64-bit
Jan 25 13:32
mayank@mayank-VMware-Virtual-Platform: ~/Desktop/Final 1111

```
 33%|██████████          | 9/27 [00:50<01:29,  4.98s/it]INFO:root:Text files created in `data/pd
fs/learning_theory_iii`
 37%|███████████         | 10/27 [00:54<01:15,  4.41s/it]INFO:root:Text files created in `data/pd
fs/graphical_models_i`
 41%|████████████        | 11/27 [00:58<01:09,  4.37s/it]INFO:root:Text files created in `data/pd
fs/graphical_models_iii`
 48%|██████████████      | 13/27 [01:01<00:42,  3.04s/it]INFO:root:Text files created in `data/pd
fs/em_and_clustering`
 52%|███████████████     | 14/27 [01:12<01:04,  4.93s/it]INFO:root:Text files created in `data/pd
fs/boosting`
 56%|████████████████    | 15/27 [01:13<00:49,  4.09s/it]INFO:root:Text files created in `data/pd
fs/adaboost,_margins,_perceptron`
 59%|█████████████████   | 16/27 [01:15<00:38,  3.52s/it]INFO:root:Text files created in `data/pd
fs/kernels`
 63%|██████████████████  | 17/27 [01:17<00:31,  3.11s/it]INFO:root:Text files created in `data/pd
fs/svm`
 67%|███████████████████ | 18/27 [01:20<00:27,  3.10s/it]INFO:root:Text files created in `data/pd
fs/semi-supervised_learning`
 70%|████████████████████ | 19/27 [01:23<00:23,  2.88s/it]INFO:root:Text files created in `data/pd
fs/active_learning`
 74%|█████████████████████ | 20/27 [01:28<00:24,  3.47s/it]INFO:root:Text files created in `data/pd
fs/partitional_clustering_hierarchical_clustering`
 78%|██████████████████████ | 21/27 [01:30<00:18,  3.08s/it]INFO:root:Text files created in `data/pd
fs/learning_representations_dimensionality_reduction`
 81%|███████████████████████ | 22/27 [01:33<00:15,  3.20s/it]INFO:root:Text files created in `data/pd
fs/never_ending_learning`
 85%|████████████████████████ | 23/27 [01:53<00:32,  8.15s/it]INFO:root:Text files created in `data/pd
fs/neural_networks_deep_learning`
 89%|█████████████████████████ | 24/27 [02:40<00:59, 19.79s/it]INFO:root:Text files created in `data/pd
fs/reinforcement_learning`
100%|████████████████████████████| 27/27 [02:42<00:00,  6.01s/it]
(myenv) mayank@mayank-VMware-Virtual-Platform:~/Desktop/Final 1111$
```

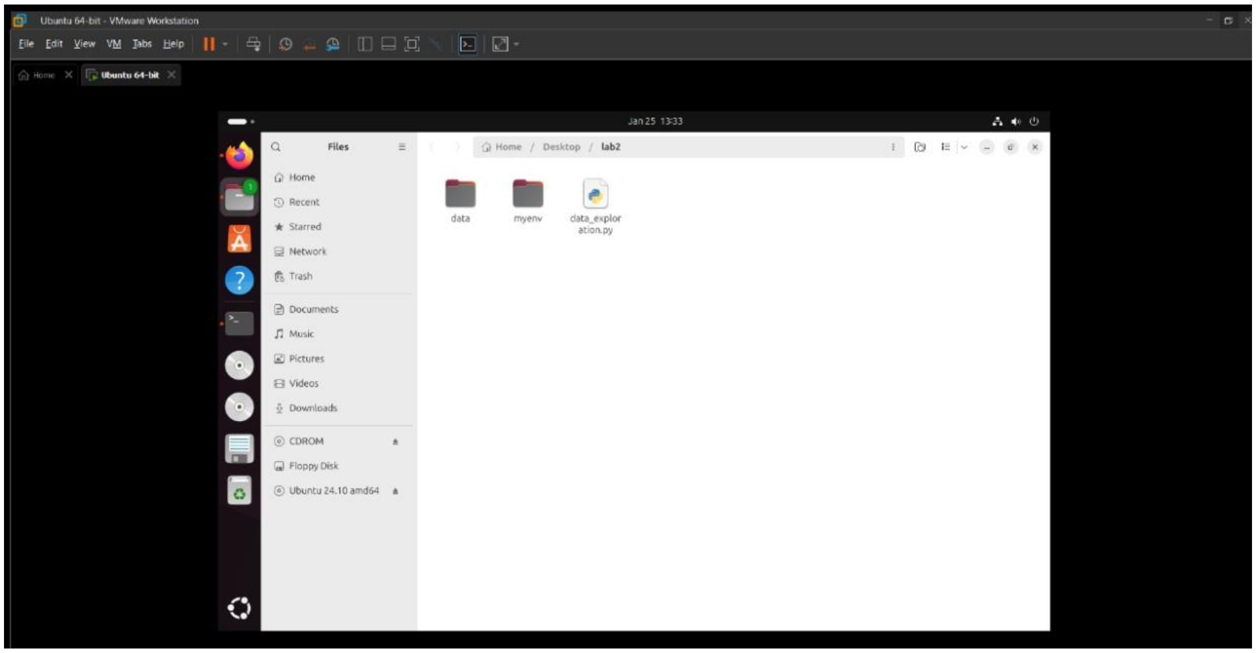

Ubuntu 64-bit - VMware Workstation
File Edit View VM Tabs Help
Home
Ubuntu 64-bit
Jan 25 13:34
Open
page_1.txt
~/Desktop/lab2/data/pdfs/adaboost,_margins,_perceptron
ml.csv
page_1.txt

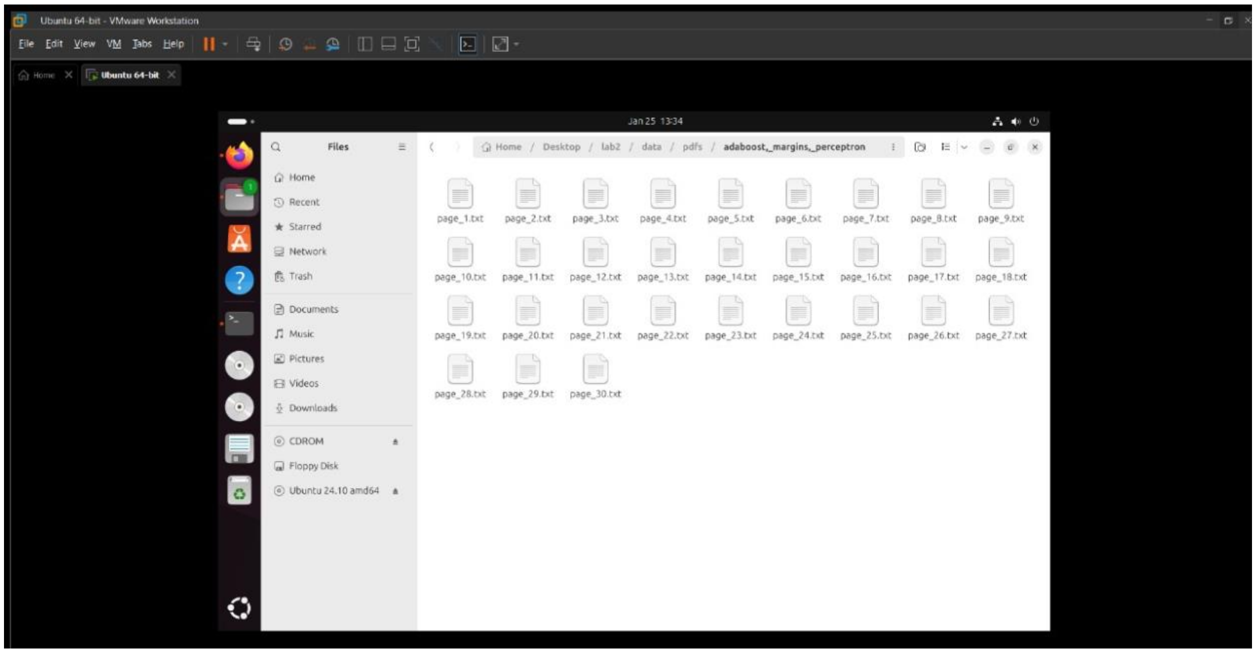
```
Boosting Approach to ML
Maria-Florina Balcan
03/18/2015
Perceptron, Margins, Kernels
```

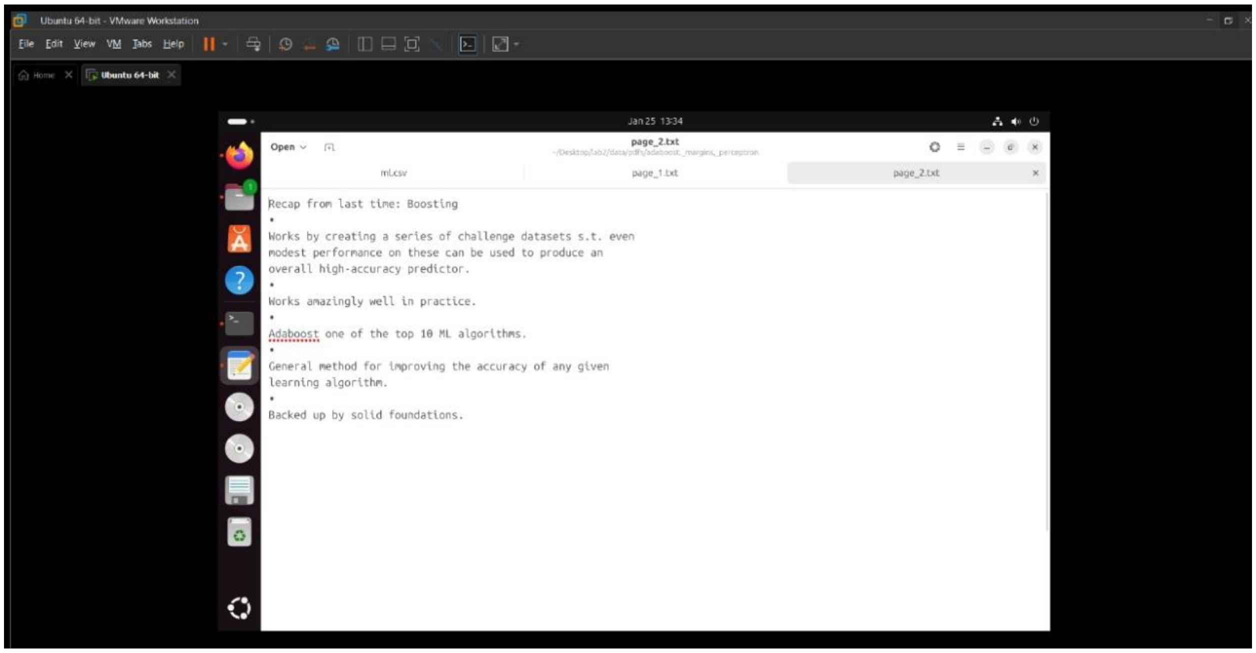




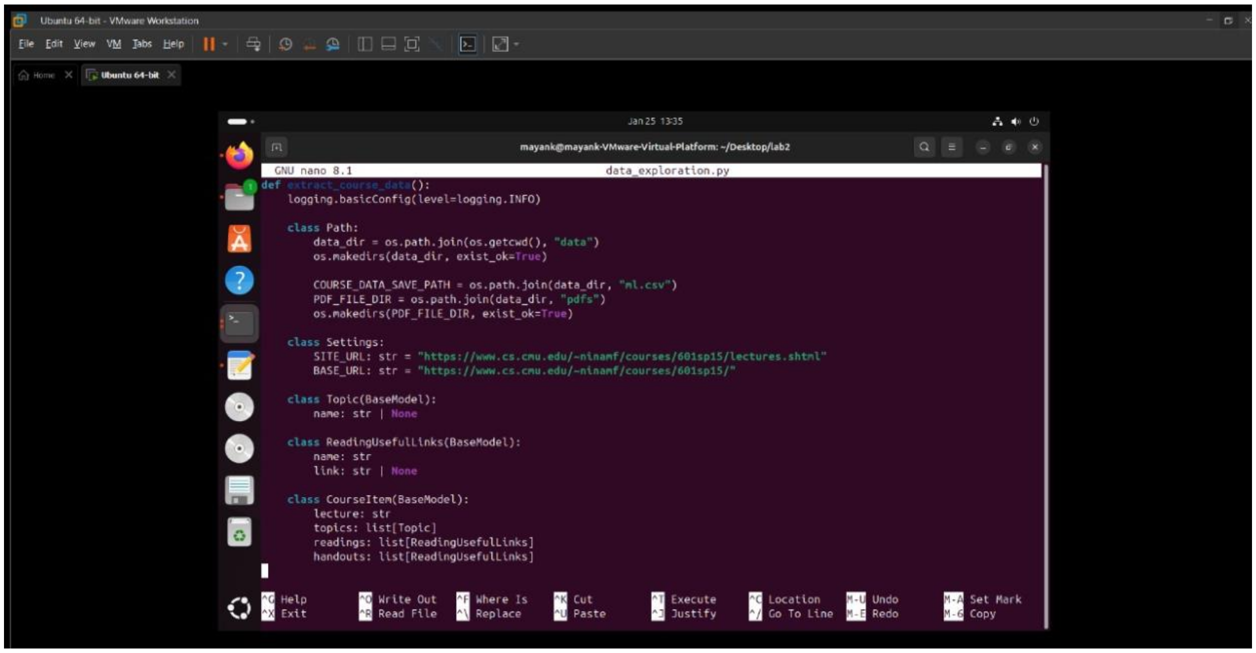
```
lecture,topics,readings,handouts
Intro to ML Decision Trees,"[{'name': 'Machine learning examples'}, {'name': 'Well defined machine learning problem'},
{'name': 'Decision tree learning'}]","[{'name': 'The Discipline of Machine Learning', 'link': 'http://www.cs.cmu.edu/
%7Etom/pubs/MachineLearning.pdf'}, {'name': 'Mitchell: Ch 3 Bishop: Ch 14.4', 'link': None}]","[{'name': 'Slides', 'link':
'https://www.cs.cmu.edu/~ninamf/courses/601sp15/slides/01_DTreesAndOverfitting-1-12-2015.pdf'}, {'name': 'Video', 'link':
'https://www.cs.cmu.edu/~ninamf/courses/601sp15/video/1.html'}]"
Decision Tree learning Review of Probability,"[{'name': 'The big picture'}, {'name': 'Overfitting'}, {'name': 'Random
variables and probabilities'}]","[{'name': ""Andrew Moore's Basic Probability Tutorial"", 'link': 'http://
www.autonlab.org/tutorials/prob18.pdf'}, {'name': 'Mitchell: Ch 3', 'link': None}]","[{'name': 'Slides', 'link': 'https://
www.cs.cmu.edu/~ninamf/courses/601sp15/slides/02_Overfitting_ProbReview-1-14-2015.pdf'}, {'name': 'Annotated Slides',
'link': 'https://www.cs.cmu.edu/~ninamf/courses/601sp15/slides/02_Overfitting_ProbReview-1-14-2015_ann.pdf'}, {'name':
'Video', 'link': 'https://www.cs.cmu.edu/~ninamf/courses/601sp15/video/2.html'}]"
Probability and Estimation,"[{'name': 'Bayes rule'}, {'name': 'MLE'}, {'name': 'MAP'}]","[{'name': 'Estimating
Probabilities', 'link': 'http://www.cs.cmu.edu/%7Etom/mlbook/Joint_MLE_MAP.pdf'}, {'name': 'Mitchell:', 'link':
None}]","[{'name': 'Slides', 'link': 'https://www.cs.cmu.edu/~ninamf/courses/601sp15/slides/
03_MLE_MAP_NBayes-1-21-2015.pdf'}, {'name': 'Annotated Slides', 'link': 'https://www.cs.cmu.edu/~ninamf/courses/601sp15/
slides/03_MLE_MAP_NBayes-1-21-2015_ann.pdf'}, {'name': 'Video', 'link': 'https://www.cs.cmu.edu/~ninamf/courses/601sp15/
video/3.html'}]"
Naive Bayes,"[{'name': 'Conditional Independence'}, {'name': 'Naive Bayes: why and how'}]","[{'name': 'Naive Bayes and
Logistic Regression', 'link': 'http://www.cs.cmu.edu/%7Etom/mlbook/NBayesLogReg.pdf'}, {'name': 'Mitchell:', 'link':
None}]","[{'name': 'Slides', 'link': 'https://www.cs.cmu.edu/~ninamf/courses/601sp15/slides/
04_NBayes-1-26-2015.pptx.pdf'}, {'name': 'Annotated Slides', 'link': 'https://www.cs.cmu.edu/~ninamf/courses/601sp15/
slides/04_NBayes-1-26-2015-ann.pdf'}, {'name': 'Video', 'link': 'http://youtu.be/ngKU5466i4o'}]"
Gaussian Naive Bayes,"[{'name': 'Gaussian Bayes classifiers'}, {'name': 'Document Classification'}, {'name': 'Brain image
classification'}, {'name': 'Form of decision surfaces'}]","[{'name': 'Naive Bayes and Logistic Regression', 'link':
'http://www.cs.cmu.edu/%7Etom/mlbook/NBayesLogReg.pdf'}, {'name': 'Mitchell:', 'link': None}]","[{'name': 'Slides',
'link': 'https://www.cs.cmu.edu/~ninamf/courses/601sp15/slides/05_GNB_1-28-2015.pdf'}, {'name': 'Annotated Slides',
'link': 'https://www.cs.cmu.edu/~ninamf/courses/601sp15/slides/05_GNB_1-28-2015-ann.pdf'}, {'name': 'Video', 'link':
```


Ubuntu 64-bit - VMware Workstation
File Edit View VM Tabs Help
Home
Ubuntu 64-bit
Jan 25 13:33
Files
Home / Desktop / lab2
Home
Recent
Starred
Network
Trash
Documents
Music
Pictures
Videos
Downloads
CDROM
Floppy Disk
Ubuntu 24.10 amd64
data
myenv
data_exploration.py


Ubuntu 64-bit - VMware Workstation
File Edit View VM Tabs Help
Home
Ubuntu 64-bit
Jan 25 13:34
Files
Home / Desktop / lab2 / data / pdfs / adaboost,_margins,_perceptron
Home
Recent
Starred
Network
Trash
Documents
Music
Pictures
Videos
Downloads
CDROM
Floppy Disk
Ubuntu 24.10 amd64
page_1.txt
page_2.txt
page_3.txt
page_4.txt
page_5.txt
page_6.txt
page_7.txt
page_8.txt
page_9.txt
page_10.txt
page_11.txt
page_12.txt
page_13.txt
page_14.txt
page_15.txt
page_16.txt
page_17.txt
page_18.txt
page_19.txt
page_20.txt
page_21.txt
page_22.txt
page_23.txt
page_24.txt
page_25.txt
page_26.txt
page_27.txt
page_28.txt
page_29.txt
page_30.txt


Ubuntu 64-bit - VMware Workstation
File Edit View VM Tabs Help
Home
Ubuntu 64-bit
Jan 25 13:34
Open
page_2.txt
~/Desktop/lab2/data/pdfs/adaboost,_margins,_perceptron
ml.csv
page_1.txt
page_2.txt

```
Recap from last time: Boosting
•
Works by creating a series of challenge datasets s.t. even
modest performance on these can be used to produce an
overall high-accuracy predictor.
•
Works amazingly well in practice.
•
Adaboost one of the top 10 ML algorithms.
•
General method for improving the accuracy of any given
learning algorithm.
•
Backed up by solid foundations.
```


Ubuntu 64-bit - VMware Workstation
File Edit View VM Tabs Help
Home
Ubuntu 64-bit
Jan 25 13:35
mayank@mayank-VMware-Virtual-Platform: ~/Desktop/lab2

```
 GNU nano 8.1                         data_exploration.py
def extract_course_data():
    logging.basicConfig(level=logging.INFO)

    class Path:
        data_dir = os.path.join(os.getcwd(), "data")
        os.makedirs(data_dir, exist_ok=True)

        COURSE_DATA_SAVE_PATH = os.path.join(data_dir, "ml.csv")
        PDF_FILE_DIR = os.path.join(data_dir, "pdfs")
        os.makedirs(PDF_FILE_DIR, exist_ok=True)

    class Settings:
        SITE_URL: str = "https://www.cs.cmu.edu/~ninamf/courses/601sp15/lectures.shtml"
        BASE_URL: str = "https://www.cs.cmu.edu/~ninamf/courses/601sp15/"

    class Topic(BaseModel):
        name: str | None

    class ReadingUsefulLinks(BaseModel):
        name: str
        link: str | None

    class CourseItem(BaseModel):
        lecture: str
        topics: list[Topic]
        readings: list[ReadingUsefulLinks]
        handouts: list[ReadingUsefulLinks]

^G Help      ^O Write Out  ^F Where Is  ^K Cut       ^T Execute   ^C Location   M-U Undo    M-A Set Mark
^X Exit      ^R Read File  ^\ Replace   ^U Paste     ^J Justify   ^/ Go To Line M-E Redo    M-6 Copy
```

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.