DSCI560 Lab2 Report

Group Code Submission Link:
https://github.com/shubhamdarekar/DSCI560---Shubham/tree/main/Lab%202

Part 1.
Team name: For Resume

Team details:
Saavani Vaidya 9385579920
Yuxuan Liu 4780355176
Shubham Darekar 1641138809

Part 2.
Saavani Vaidya
Financial Education Chatbot

- Regional Cost of Living Analysis:
  - https://www.kaggle.com/datasets/heidarmirhajisadati/regional-cost-of-living-analysis
  - This dataset provides insights into the cost of living and average monthly income across various countries and regions worldwide from 2000 to 2023.
  - Reasoning: This data can help the chatbot offer practical advice rooted in real world financial conditions. The chatbot can give users insights on how to adjust their finances based on regional expenses, such as rent, utilities, and daily needs, as well as help with budgeting.
- Credit Card Eligibility Data
  - https://www.kaggle.com/datasets/rohit265/credit-card-eligibility-data-determining-factors
  - This dataset provides a variety of attributes that can be used for analysis and modeling to understand the factors influencing credit card eligibility.
  - Reasoning: By understanding patterns from the dataset, the chatbot can offer suggestions to users on how to improve their creditworthiness, such as increasing income and paying off debts.

Course Textbook Chatbot

- Course to Textbook Mapping
  - https://www.kaggle.com/datasets/polartech/us-college-textbooks-and-courses-dataset
  - The datasets contain over 200,000 courses from 40+ American universities and maps them to their textbooks and information about them.
  - Reasoning: With the mapping and information on textbook content, chapters, and topics, the chatbot can assist students in quickly locating specific sections or topics within a textbook, identify overlap between courses that use the same textbook, and give better content summaries depending on the course.

Shubham Darekar
Medical Datasets:

- First Aid Help chatbot:
    - First Aid Intents Dataset, First Aid Recommendations Intents, Websites like:First Aid Instructions for 10 Medical Emergencies
    - First Aid Data is publicly available and can be helpful to chat with a bot in situations where quick help is required rather than skimming through the texts and available First Aid brochures
-

Itinerary planner for a Tourist Town:

- Websites like City of Sedona | Home and its outlinks as well as data from A Jam-Packed Sedona Itinerary! - My Perfect Itinerary
- As most of the tourist spots have documented important information, as well as there are websites which list down itineraries according to the availability of time, a chatbot can help in planning the day

Customer Service for Online stores:

- eCommerce Customer Service Satisfaction, Telecom Customers
- With the transcribed information from customer service calls, as well as the written communication, a chatbot can be trained and used to answer primary questions of the end user.

Yuxuan Liu
Programming Resources:

- Stack Overflow Data:
    - https://www.kaggle.com/datasets/stackoverflow/stackoverflow
    - This dataset includes real world programming questions, answers, and discussions.
    - Enables the chatbot to address common programming issues and provides detailed troubleshooting and debugging assistance.

FAQ Dataset:

- FAQ Dataset:
    - https://www.kaggle.com/datasets/umairnasir14/all-kaggle-questions-on-qoura-dataset
    - This dataset contains frequently asked questions across various domains, such as customer support, education, and finance. It includes question-answer pairs, which can be used to train the chatbot to handle routine inquiries.
    - Enables the chatbot to provide quick and accurate responses to common questions and improves user experience by automating repetitive task

Part 3.

We selected the travel itinerary planner idea from above to use in this part because we thought it would have more data available to collect for this assignment than the other topics.

Saavani
https://www.kaggle.com/datasets/fuarresvij/bali-popular-destination-for-tourist-2022
https://www.kaggle.com/datasets/vitaliymalcev/russian-touris-attractions
https://www.kaggle.com/datasets/faizadani/european-tour-destinations-dataset

Shubham
Tourist Chatbot for Hill Track Areas Bangladesh
Home - My Perfect Itinerary
Travel Dataset: Guide to India's Must See Places
Tour-itinerary.pdf
https://latourist.com/documents/LA_Tourist_Itinerary.pdf

Yuxuan
https://www.kaggle.com/datasets/vitaliymalcev/russian-touris-attractions
NZJYBG1.pdf
https://www.mytouragent.com/6D_PEK_-_XIA_PVT.pdf

Part 4.

Data Selection:

i. CSV or Excel

   https://www.kaggle.com/datasets/faizadani/european-tour-destinations-dataset

ii. ASCII Texts like Forum Postings and HTML

   Home - My Perfect Itinerary

iii. PDF and Word Documents that require conversion and OCR

   https://latourist.com/documents/LA_Tourist_Itinerary.pdf

We picked these out of the options in the previous part because they had the most diverse and descriptive data.

Chatbot Improvements:

1. Kayak Chatbot

One example of an existing tourism related chatbot is the Ask Kayak Chatbot. It helps users search for flights, hotels, and car rentals, provides pricing details and booking assistance, and can answer simple travel-related queries. Some limitations are that it has limited itinerary customization for activities beyond booking, it cannot plan multi-destination trips efficiently, and it is focused mainly on transactions instead of detailed travel planning. Our dataset focuses more on the activity and detailed planning side of tourism which will improve the performance in these aspects.

2. Copilot2trip Chatbot

Copilot2trip, like many AI travel assistants, faces challenges in emotional intelligence, complex query handling, contextual understanding, and creative problem-solving. However, by enhancing our dataset, we can significantly improve its performance. Incorporating a diverse range of real-world travel planning conversations will help the AI better understand and respond to users' emotional needs and complex requests. Expanding the database with detailed, up-to-date information on various travel destinations will improve contextual understanding, allowing Copilot2trip to provide more nuanced and personalized recommendations. Including case studies of complex travel issues and their resolutions will enhance the AI's problem-solving capabilities, enabling it to offer more tailored and creative solutions. These improvements will help Copilot2trip evolve from a basic information provider to a more empathetic, adaptable, and innovative travel planning assistant, capable of handling the diverse and often unpredictable needs of modern travelers.

3. Wanderboat Chatbot

Wanderboat faces limitations such as the lack of accommodation recommendations and flight booking support, as well as a dependency on existing datasets that might be outdated or inaccurate, leading to discrepancies between suggestions and real-world conditions. In contrast, our dataset offers rich contextual data with detailed multi-turn conversations and annotations, enabling the chatbot to maintain context throughout complex interactions. Additionally, our dataset includes domain-specific knowledge, such as examples of hotel bookings, visa requirements, and budget constraints, allowing the chatbot to address these gaps and provide more accurate and comprehensive travel planning assistance.

Script:

CSV

Code for CSV file:

```
 GNU nano 7.2                                                  data exploration.py *
import pandas as pd

#uploaded destinations.csv
#from https://www.kaggle.com/datasets/faizadani/european-tour-destinations-dataset?select=destinations.csv
#to Google Drive for easy access with new link
df = pd.read_csv("https://drive.google.com/uc?id=1yXmfBxtt1RGJK0HImvcv3ZlTz97RWUcD", encoding='latin1')

#display first few records
print(df.head())

#calculate size and dimension of datset
print("size : ", df.size)
print("dimension (row, col) : ", df.shape)

#identify missing data
print("missing values present: ", df.isnull().values.any())
print("missing values per col:")
print(df.isnull().sum())
print("total missing values: ", df.isnull().sum().sum())
```

This part of the data_exploration.py script reads the Kaggle dataset CSV file from the given Google Drive URL into a Pandas dataframe. It shows the first few records as well as the size and dimension of the dataset to give an overview of the structure. Then, it checks if there are any missing values in the dataset, counts the number of missing values in each column, and calculates the total number of missing values in the entire dataset.

After running the command “python3 data_exploration.py” we get these results for the CSV:

```
  Destination    Region  ...                          Cultural Significance                                        Description
0        Rome     Lazio  ...  The capital city, known for its historical lan...  A hub of ancient history and modern culture, w...
1    Florence   Tuscany  ...  A Renaissance city famous for its art, archite...  Home to world-class museums, including the Uff...
2      Venice    Veneto  ...  A unique city built on canals, known for its g...  An iconic city of water, renowned for romantic...
3       Milan  Lombardy  ...  A fashion capital known for its shopping, muse...  A modern city with an ancient soul, featuring ...
4      Naples  Campania  ...  A vibrant city known for its delicious food, h...  Famous for pizza, Pompeii, and proximity to th...

[5 rows x 16 columns]
size :  3344
dimension (row, col) :  (209, 16)
missing values present:  True
missing values per col:
Destination                     0
Region                          0
Country                         0
Category                        0
Latitude                        0
Longitude                       0
Approximate Annual Tourists     0
Currency                        0
Majority Religion               0
Famous Foods                    0
Language                        0
Best Time to Visit              0
Cost of Living                  0
Safety                          0
Cultural Significance           0
Description                    50
dtype: int64
total missing values:  50
```

HTML

Code for HTML:

```
 GNU nano 7.2                                                  data_exploration.py *
def initialize_driver():
    chrome_options = Options()
    chrome_options.add_argument("--headless")
    chrome_options.add_argument("--no-sandbox")
    chrome_options.add_argument("--disable-gpu")
    chrome_options.add_argument("--disable-dev-shm-usage")

    service = Service(executable_path="/usr/bin/chromedriver")
    driver = webdriver.Chrome(service=service, options=chrome_options)
    return driver

def scrape_cards():
    url = "https://myperfectitinerary.com/category/itineraries/"
    driver = initialize_driver()
    driver.get(url)

    soup = BeautifulSoup(driver.page_source, 'html.parser')
    driver.quit()

    data = []

    # Find all card elements
    cards = soup.find_all("a", class_="penci-image-holder penci-lazy")
    for card in cards:
        title = card["title"].strip() if "title" in card.attrs else "No Title"
        link = card["href"].strip() if "href" in card.attrs else "No Link"
        data.append({"Title": title, "Link": link})

    df = pd.DataFrame(data)
    csv_path = "itineraries_cards.csv"
    df.to_csv(csv_path, index=False, encoding='utf-8')
    print("Data has been saved to itineraries_cards.csv!")
    return csv_path
```

This section of the data_exploration.py script fetches data from the "Itineraries" section of My Perfect Itinerary by accessing the URL: https://myperfectitinerary.com/category/itineraries/. Using Selenium, the script loads the webpage, and BeautifulSoup processes its HTML content to locate and extract card elements. For each card, the script captures title and link.

The extracted data is saved into a CSV file for further analysis:

```
(venv) paracosmgracee@yuxuanliu:~/lab2$ python3 data_exploration.py
Data has been saved to itineraries_cards.csv!

--- First Few Records ---
                                    Title                                                Link
0  A Perfect Weekend in Scottsdale: 3 Jam-Packed ...  https://myperfectitinerary.com/weekend-in-scot...
1       Best Things to do in Scottsdale for Couples!  https://myperfectitinerary.com/best-things-to-...
2  23 UNIQUE Things to do in La Fortuna - Time fo...  https://myperfectitinerary.com/things-to-do-in...
3  10 Days in Costa Rica - Explore The Jungle & T...  https://myperfectitinerary.com/10-days-in-cost...
4  27 Things to do in Buena Park for the PERFECT ...  https://myperfectitinerary.com/things-to-do-in...

--- Dataset Size and Dimensions ---
Rows: 10, Columns: 2

--- Missing Data ---
Title    0
Link     0
dtype: int64
(venv) paracosmgracee@yuxuanliu:~/lab2$ cat itineraries_cards.csv
Title,Link
A Perfect Weekend in Scottsdale: 3 Jam-Packed Days!,https://myperfectitinerary.com/weekend-in-scottsdale/
Best Things to do in Scottsdale for Couples!,https://myperfectitinerary.com/best-things-to-do-in-scottsdale-for-couples/
23 UNIQUE Things to do in La Fortuna - Time for an adventure!,https://myperfectitinerary.com/things-to-do-in-la-fortuna/
10 Days in Costa Rica - Explore The Jungle & The Coast!,https://myperfectitinerary.com/10-days-in-costa-rica-explore-the-jungle-the-coast/
27 Things to do in Buena Park for the PERFECT Girls Weekend!,https://myperfectitinerary.com/things-to-do-in-buena-park/
Weekend in Vegas Itinerary - A Perfect 3 Days in Las Vegas!,https://myperfectitinerary.com/weekend-in-vegas-itinerary/
A Jam-Packed Capri Day Trip from the Amalfi Coast!,https://myperfectitinerary.com/capri-day-trip/
An EPIC 2 Week Italy Itinerary (With Amalfi Coast)!,https://myperfectitinerary.com/2-week-italy-itinerary-with-amalfi-coast/
A Romantic Lake Como Itinerary for Your Italian Summer Holiday!,https://myperfectitinerary.com/lake-como-itinerary/
Utah National Parks Road Trip - 14 EPIC DAYS!,https://myperfectitinerary.com/utah-national-parks-road-trip/
```

PDF:

Code for PDF:

```python
import pandas as pd
import pdfplumber
import requests

## Task: Extract structured data from a PDF file and save it to a CSV file
def get_raw_data_from_url(url,file_name):
    response = requests.get(url)
    file_name = "Lab 2/raw_data/" + file_name

    if response.status_code == 200:
        with open(file_name, "wb") as file:
            file.write(response.content)
        print(f"File downloaded successfully as {file_name}")
    else:
        print(f"Failed to download file. Status code: {response.status_code}")

def extract_text_from_pdf(pdf_file,text_file_name):
    extracted_text = ""
    with pdfplumber.open("Lab 2/raw_data/"+pdf_file) as pdf:
        for page in pdf.pages:
            extracted_text += page.extract_text()

    text_file = "Lab 2/raw_data/" + text_file_name
    with open(text_file, "w", encoding="utf-8") as file:
        file.write(extracted_text)
    print(f"Text extracted and saved to {text_file}")

def get_data_from_text(text_file):
    text_file = "Lab 2/raw_data/" + text_file

    with open(text_file, "r", encoding="utf-8") as file:
        extracted_text = file.read()
    data = []
    columns = ["Day", "Time", "Activity"]
    current_day = None

    for line in extracted_text.split("\n"):
        line = line.strip()
        if line.isupper() and "DAY" in line:
            current_day = line
        elif current_day and ":" in line:
            parts = line.split(maxsplit=1)
            if len(parts) == 2:
                time, activity = parts
```

This part of the script downloads the pdf file from the web using the request library and saves it in the raw_data folder. Then using the pdfplumber library, it extracts the text from the pdf files. Then using the text extracted, the code tries to extract day time and activity from the itinerary, and save it into a python list. Using this list it is saved in csv format.

```python
def get_data_from_pdf():
    # Defining the PDF file name and URL
    pdf_file = "LA_Tourist_Itinerary.pdf"
    url = "https://latourist.com/documents/LA_Tourist_Itinerary.pdf"

    # Downloading the raw PDF file from the URL
    get_raw_data_from_url(url, pdf_file)

    # Extracting text from the downloaded PDF file and save it to a text file
    extract_text_from_pdf(pdf_file, "LA_Tourist_Itinerary.txt")

    # Extracting structured data from the text file and create a DataFrame
    df = get_data_from_text("LA_Tourist_Itinerary.txt")

    # Saving the DataFrame to a CSV file
    save_to_csv(df, "Lab 2/processed_data/LA_Tourist_Itinerary.csv")

    # Printing the first few records of the DataFrame
    print("\nFirst few records:")
    print(df.head())

    # Printing the dimensions of the DataFrame
    print("\nDataset dimensions:")
    print(df.shape)

    # Checking for missing data in the DataFrame
    print("\nMissing data check:")
    print(df.isnull().sum())
```

Now these functions are called in the main data_exploration.py file.