**Step 1: Import Necessary Libraries**

First, import the necessary libraries: Requests for making HTTP requests, BeautifulSoup for parsing HTML content (if you don't already have it installed from the previous step), and CSV for saving the data.

import requests

from bs4 import BeautifulSoup

import csv

### Step 2: Define the Base URL and CSV Headers

Set the base URL for the dataset listings and define the headers for the CSV file where the scraped data will be saved.

def scrape\_uci\_datasets():

base\_url = "https://archive.ics.uci.edu/datasets"

headers = [

"Dataset Name", "Donated Date", "Description",

"Dataset Characteristics", "Subject Area", "Associated Tasks",

"Feature Type", "Instances", "Features"

]

data = []

### Step 3: Create a Function to Scrape Dataset Details

Define a function scrape\_dataset\_details that takes the URL of an individual dataset page, retrieves the HTML content, parses it using BeautifulSoup, and extracts relevant information.

def scrape\_dataset\_details(dataset\_url):

response = requests.get(dataset\_url)

soup = BeautifulSoup(response.text, 'html.parser')

dataset\_name = soup.find(

'h1', class\_='text-3xl font-semibold text-primary-content')

dataset\_name = dataset\_name.text.strip() if dataset\_name else "N/A"

donated\_date = soup.find('h2', class\_='text-sm text-primary-content')

donated\_date = donated\_date.text.strip().replace(

'Donated on ', '') if donated\_date else "N/A"

description = soup.find('p', class\_='svelte-17wf9gp')

description = description.text.strip() if description else "N/A"

details = soup.find\_all('div', class\_='col-span-4')

dataset\_characteristics = details[0].find('p').text.strip() if len(

details) > 0 else "N/A"

subject\_area = details[1].find('p').text.strip() if len(

details) > 1 else "N/A"

associated\_tasks = details[2].find('p').text.strip() if len(

details) > 2 else "N/A"

feature\_type = details[3].find('p').text.strip() if len(

details) > 3 else "N/A"

instances = details[4].find('p').text.strip() if len(

details) > 4 else "N/A"

features = details[5].find('p').text.strip() if len(

details) > 5 else "N/A"

return [

dataset\_name, donated\_date, description, dataset\_characteristics,

subject\_area, associated\_tasks, feature\_type, instances, features

]

### Step 4: Create a Function to Scrape Dataset Listings

Define a function scrape\_datasets that takes the URL of a page listing multiple datasets, retrieves the HTML content, and finds all dataset links. For each link, it calls scrape\_dataset\_details to get detailed information.

def scrape\_datasets(page\_url):

response = requests.get(page\_url)

soup = BeautifulSoup(response.text, 'html.parser')

dataset\_list = soup.find\_all(

'a', class\_='link-hover link text-xl font-semibold')

if not dataset\_list:

print("No dataset links found")

return

for dataset in dataset\_list:

dataset\_link = "https://archive.ics.uci.edu" + dataset['href']

print(f"Scraping details for {dataset.text.strip()}...")

dataset\_details = scrape\_dataset\_details(dataset\_link)

data.append(dataset\_details)

### Step 5: Loop Through Pages Using Pagination Parameters

Implement a loop to navigate through the pages using pagination parameters. The loop continues until no new data is added, indicating that all pages have been scraped.

skip = 0

take = 10

while True:

page\_url = f"https://archive.ics.uci.edu/datasets?skip={skip}&take={take}&sort=desc&orderBy=NumHits&search="

print(f"Scraping page: {page\_url}")

initial\_data\_count = len(data)

scrape\_datasets(page\_url)

if len(

data

) == initial\_data\_count:

break

skip += take

### Step 6: Save the Scraped Data to a CSV File

After scraping all the data, save it to a CSV file.

with open('uci\_datasets.csv', 'w', newline='', encoding='utf-8') as file:

writer = csv.writer(file)

writer.writerow(headers)

writer.writerows(data)

print("Scraping complete. Data saved to 'uci\_datasets.csv'.")

### Step 7: Run the Scraping Function

Finally, call the scrape\_uci\_datasets function to start the scraping process.

scrape\_uci\_datasets()

## **Full Code:**

import requests

from bs4 import BeautifulSoup

import csv

def scrape\_uci\_datasets():

base\_url = "https://archive.ics.uci.edu/datasets"

headers = [

"Dataset Name", "Donated Date", "Description",

"Dataset Characteristics", "Subject Area", "Associated Tasks",

"Feature Type", "Instances", "Features"

]

data = []

def scrape\_dataset\_details(dataset\_url):

response = requests.get(dataset\_url)

soup = BeautifulSoup(response.text, 'html.parser')

dataset\_name = soup.find(

'h1', class\_='text-3xl font-semibold text-primary-content')

dataset\_name = dataset\_name.text.strip() if dataset\_name else "N/A"

donated\_date = soup.find('h2', class\_='text-sm text-primary-content')

donated\_date = donated\_date.text.strip().replace(

'Donated on ', '') if donated\_date else "N/A"

description = soup.find('p', class\_='svelte-17wf9gp')

description = description.text.strip() if description else "N/A

details = soup.find\_all('div', class\_='col-span-4')

dataset\_characteristics = details[0].find('p').text.strip() if len(

details) > 0 else "N/A"

subject\_area = details[1].find('p').text.strip() if len(

details) > 1 else "N/A"

associated\_tasks = details[2].find('p').text.strip() if len(

details) > 2 else "N/A"

feature\_type = details[3].find('p').text.strip() if len(

details) > 3 else "N/A"

instances = details[4].find('p').text.strip() if len(

details) > 4 else "N/A"

features = details[5].find('p').text.strip() if len(

details) > 5 else "N/A"

return [

dataset\_name, donated\_date, description, dataset\_characteristics,

subject\_area, associated\_tasks, feature\_type, instances, features

]

def scrape\_datasets(page\_url):

response = requests.get(page\_url)

soup = BeautifulSoup(response.text, 'html.parser')

dataset\_list = soup.find\_all(

'a', class\_='link-hover link text-xl font-semibold')

if not dataset\_list:

print("No dataset links found")

return

for dataset in dataset\_list:

dataset\_link = "https://archive.ics.uci.edu" + dataset['href']

print(f"Scraping details for {dataset.text.strip()}...")

dataset\_details = scrape\_dataset\_details(dataset\_link)

data.append(dataset\_details)

skip = 0

take = 10

while True:

page\_url = f"https://archive.ics.uci.edu/datasets?skip={skip}&take={take}&sort=desc&orderBy=NumHits&search="

print(f"Scraping page: {page\_url}")

initial\_data\_count = len(data)

scrape\_datasets(page\_url)

if len(

data

) == initial\_data\_count:

break

skip += take

with open('uci\_datasets.csv', 'w', newline='', encoding='utf-8') as file:

writer = csv.writer(file)

writer.writerow(headers)

writer.writerows(data)

print("Scraping complete. Data saved to 'uci\_datasets.csv'.")

scrape\_uci\_datasets()