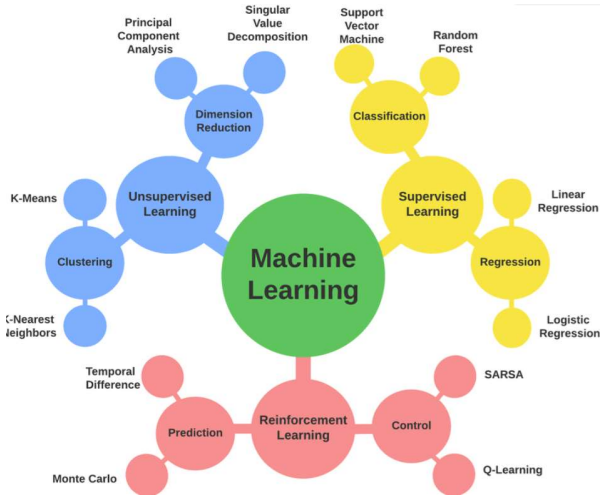
Các phương pháp học máy
Machine learning methods

4 TC: 2 LT – 2 TH

Giảng viên: **Tạ Hoàng Thắng**

tahoangthang@gmail.com

0975399307

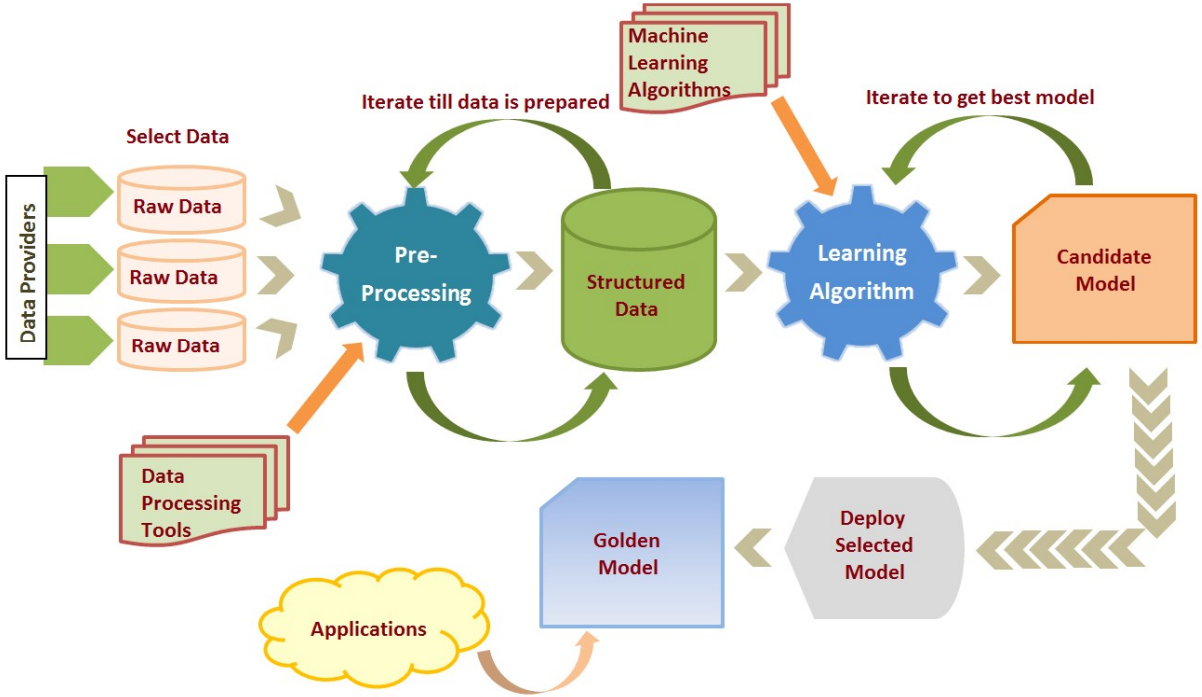


Nhập nguồn dữ liệu

Import data

There are several ways for importing data in machine learning using Python:

- Through URLs
- Integrated Libraries
- From Files



Nhập nguồn dữ liệu

Through URLs

- Use **requests** library (`pip install requests`)

```python
import requests

# Define the URL
url = 'https://www.example.com'

# Send a GET request to the URL
response = requests.get(url)

# Check if the request was successful
if response.status_code == 200:
    # Print the content of the response
    print(response.text)
else:
    print(f"Failed to retrieve data. Status code: {response.status_code}")
```

Nhập nguồn dữ liệu

Integrated Libraries

- Use **sklearn.datasets**
  - https://scikit-learn.org/stable/datasets.html

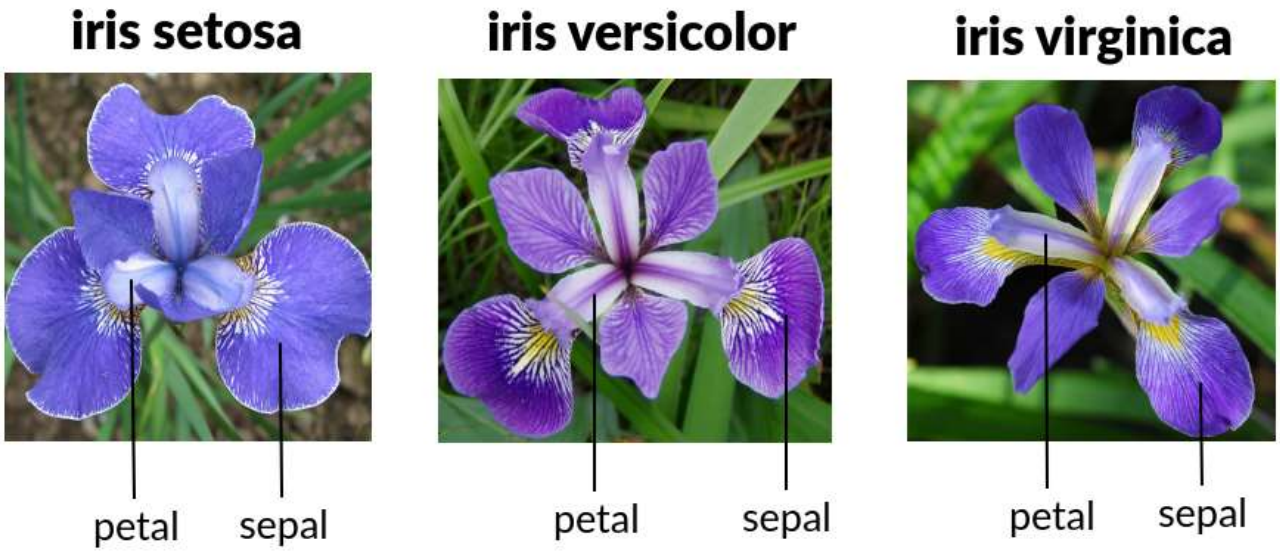
7.1. Toy datasets

7.2. Real world datasets

Nhập nguồn dữ liệu

Integrated Libraries

- Use **sklearn.datasets**
  - Load dataset **iris**



Nhập nguồn dữ liệu

Integrated Libraries: Use **sklearn.datasets:** Load dataset **iris**

```python
from sklearn.datasets import load_iris
import pandas as pd

# Load the Iris dataset
iris = load_iris()

# Create a DataFrame with features and labels
iris_df = pd.DataFrame(iris.data, columns=iris.feature_names)
iris_df['species'] = iris.target

# Map numeric labels to species names
species_map = {i: species for i, species in enumerate(iris.target_names)}
iris_df['species'] = iris_df['species'].map(species_map)

# Display the DataFrame
print("Iris dataset with labels:")
print(iris_df.head())

#Iris dataset with labels:
#   sepal length (cm)  sepal width (cm)  ...  petal width (cm)  species
# 0                5.1               3.5  ...               0.2   setosa
# 1                4.9               3.0  ...               0.2   setosa
# 2                4.7               3.2  ...               0.2   setosa
# 3                4.6               3.1  ...               0.2   setosa
# 4                5.0               3.6  ...               0.2   setosa
# [5 rows x 5 columns]
```

Nhập nguồn dữ liệu

Integrated Libraries

- Use **sklearn.datasets**
  - Load dataset **wine**

```python
from sklearn.datasets import load_wine
import pandas as pd

# Load the Wine dataset
wine = load_wine()

# Create a DataFrame for the feature data
wine_df = pd.DataFrame(wine.data, columns=wine.feature_names)

# Add the target labels to the DataFrame
wine_df['class'] = wine.target

# Map target labels to class names
wine_df['class'] = wine_df['class'].map({i: name for i, name in enumerate(wine.target_names)})

# Display the first few rows of the DataFrame
print(wine_df.head())

# alcohol  malic_acid   ash  alcalinity_of_ash  ...   hue  od280/od315_of_diluted_wines  proline    class
#0    14.23        1.71  2.43               15.6  ...  1.04                          3.92   1065.0  class_0
#1    13.20        1.78  2.14               11.2  ...  1.05                          3.40   1050.0  class_0
#2    13.16        2.36  2.67               18.6  ...  1.03                          3.17   1185.0  class_0
#3    14.37        1.95  2.50               16.8  ...  0.86                          3.45   1480.0  class_0
#4    13.24        2.59  2.87               21.0  ...  1.04                          2.93    735.0  class_0
```

Nhập nguồn dữ liệu

Integrated Libraries

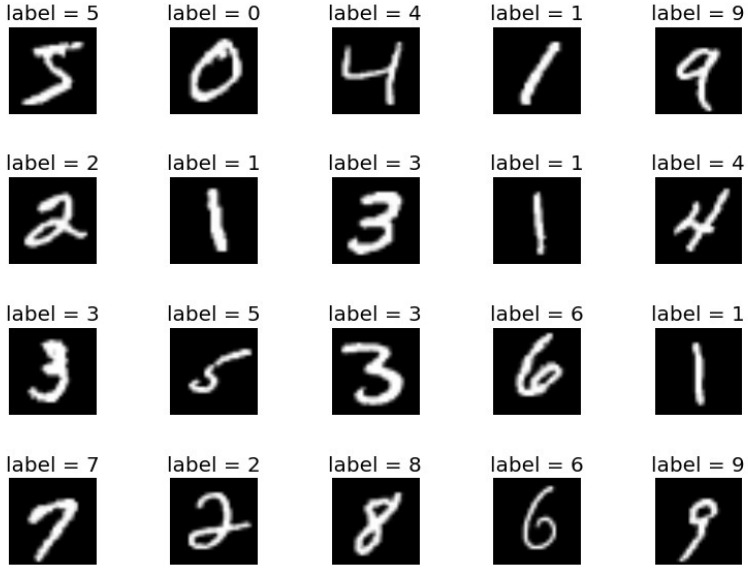
- Use **seaborn** *(pip install seaborn)*
  - Load **iris**

```python
import seaborn as sns

# Load the 'iris' dataset
iris = sns.load_dataset('iris')
print(iris.head())

#   sepal_length  sepal_width  petal_length  petal_width species
#0           5.1          3.5           1.4          0.2  setosa
#1           4.9          3.0           1.4          0.2  setosa
#2           4.7          3.2           1.3          0.2  setosa
#3           4.6          3.1           1.5          0.2  setosa
#4           5.0          3.6           1.4          0.2  setosa
```

Nhập nguồn dữ liệu

Integrated Libraries

- Use TensorFlow/PyTorch
  - Load **MNIST**



Nhập nguồn dữ liệu

Integrated Libraries

- **Use TensorFlow/PyTorch (pip install torch, pip install tensorflow)**
  - Load **MNIST**

```python
import torch
from torchvision import datasets, transforms

# Define a transform to normalize the data
transform = transforms.Compose([
    transforms.ToTensor(),  # Convert image to tensor
    transforms.Normalize((0.5,), (0.5,))  # Normalize with mean=0.5 and std=0.5
])

# Download and load the training data
train_dataset = datasets.MNIST(root='data', train=True, download=True, transform=transform)
train_loader = torch.utils.data.DataLoader(train_dataset, batch_size=64, shuffle=True)

# Download and load the test data
test_dataset = datasets.MNIST(root='data', train=False, download=True, transform=transform)
test_loader = torch.utils.data.DataLoader(test_dataset, batch_size=64, shuffle=False)
```

Nhập nguồn dữ liệu

Integrated Libraries

- Use **TensorFlow/PyTorch**
  - Load **MNIST**

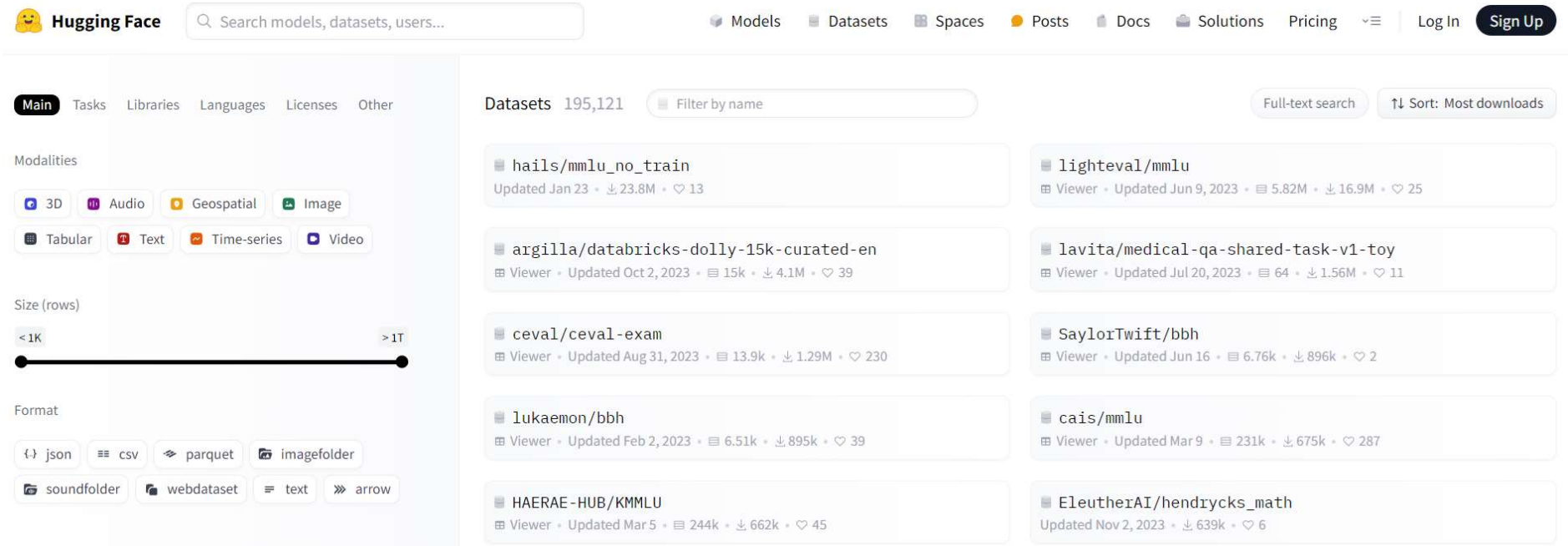
```python

# Check the size of the dataset
print(f'Training dataset size: {len(train_dataset)}')
print(f'Test dataset size: {len(test_dataset)}')

# Example: Accessing a single batch of data
data_iter = iter(train_loader)
images, labels = next(data_iter)

print(f'Batch of images shape: {images.shape}')
print(f'Batch of labels shape: {labels.shape}')
print(f'Example label: {labels[0].item()}')

```

```
Training dataset size: 60000
Test dataset size: 10000
Batch of images shape: torch.Size([64, 1, 28, 28])
Batch of labels shape: torch.Size([64])
Example label: 1
```

Nhập nguồn dữ liệu

Integrated Libraries

- Use **datasets** *(pip install datasets, https://huggingface.co/datasets)*



Nhập nguồn dữ liệu

Integrated Libraries

- Use **datasets** *(pip install datasets)*
  - *https://huggingface.co/datasets*
- Tìm hiểu **CIFAR10** và **dog-food** bằng *https://huggingface.co/datasets*
  - Số lượng dữ liệu ở tập train/test/validation?
  - Số nhãn (labels)?
  - Mô tả bí dụ?
  - Kích thước của 1 input?

Hugging Face

Nhập nguồn dữ liệu

Integrated Libraries

- Use **datasets**
  - Load **IDBM**

```python
from datasets import load_dataset

# Load the IMDB dataset
dataset = load_dataset('imdb')

# Display the dataset structure
print(dataset)

# Access the training and test splits
train_dataset = dataset['train']
test_dataset = dataset['test']

# Display the first few examples from the training set
print(train_dataset[0])

# Display the first few examples from the test set
print(test_dataset[0])
```

Nhập nguồn dữ liệu

Integrated Libraries

- Use **datasets**
  - Load **IDBM**

```
DatasetDict({
    train: Dataset({
        features: ['text', 'label'],
        num_rows: 25000
    })
    test: Dataset({
        features: ['text', 'label'],
        num_rows: 25000
    })
    unsupervised: Dataset({
        features: ['text', 'label'],
        num_rows: 50000
    })
})
```

```
{'text': 'I rented I AM CURIOUS-YELLOW
from my video store because of all the
controversy that surrounded it when it
was first released in 1967. I also heard
that at first it was seized by U.S.
customs if it ever tried to enter this
country, therefore being a fan of films
considered "controversial" I really had
to see this for myself..|.', 'label': 0}
```

Nhập nguồn dữ liệu

From Files

- **CSV: example.csv + pandas**

```
Name,Age,City
Alice,30,New York
Bob,25,Los Angeles
Charlie,35,Chicago
David,40,San Francisco
```

```python
import pandas as pd

# Load the CSV file into a DataFrame
df = pd.read_csv('example.csv')

# Display the DataFrame
print("DataFrame:")
print(df)
```

```
DataFrame:
      Name  Age           City
0    Alice   30       New York
1      Bob   25   Los Angeles
2  Charlie   35        Chicago
3    David   40  San Francisco
```

Nhập nguồn dữ liệu

From Files

- **CSV: example.csv + pandas**

```python
# View the first few rows
print("\nFirst few rows:")
print(df.head())

# View the last few rows
print("\nLast few rows:")
print(df.tail())

# Get DataFrame information
print("\nDataFrame info:")
print(df.info())
```

```
DataFrame info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4 entries, 0 to 3
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Name    4 non-null      object
 1   Age     4 non-null      int64
 2   City    4 non-null      object
dtypes: int64(1), object(2)
memory usage: 224.0+ bytes
None
```

Nhập nguồn dữ liệu

From Files

- **CSV: example.csv + pandas**

```python
# View the first few rows
print("\nFirst few rows:")
print(df.head())

# View the last few rows
print("\nLast few rows:")
print(df.tail())

# Get DataFrame information
print("\nDataFrame info:")
print(df.info())
```

```
DataFrame info:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4 entries, 0 to 3
Data columns (total 3 columns):
 #   Column  Non-Null Count  Dtype
---  ------  --------------  -----
 0   Name    4 non-null      object
 1   Age     4 non-null      int64
 2   City    4 non-null      object
dtypes: int64(1), object(2)
memory usage: 224.0+ bytes
None
```

Nhập nguồn dữ liệu

From Files

- **CSV: example.csv + csv**

```python
import csv

# Initialize an empty list to store the data
data = []

# Open and read the CSV file
with open('example.csv', 'r') as file:
    reader = csv.DictReader(file)  # Read CSV into a dictionary format
    for row in reader:
        data.append(row)

# Display the data
print("Data loaded as list of dictionaries:")
print(data)
```

```
E:\Sammi\ML>python test.py
Data loaded as list of dictionaries:
[{'Name': 'Alice', 'Age': '30', 'City': 'New York'}, {'Name': 'Bob', 'Age
': '25', 'City': 'Los Angeles'}, {'Name': 'Charlie', 'Age': '35', 'City':
 'Chicago'}, {'Name': 'David', 'Age': '40', 'City': 'San Francisco'}]
```

Nhập nguồn dữ liệu

From Files

- **TSV: example.tsv + pandas**

```
Name    Age    City
Alice   30     New York
Bob     25     Los Angeles
Charlie 35     Chicago
David   40     San Francisco
```

```python
import pandas as pd

# Load the CSV file into a DataFrame
df = pd.read_csv('example.tsv', delimiter='\t')
```

Nhập nguồn dữ liệu

From Files

- **TSV: example.tsv + csv**

```python
import csv

# Initialize an empty list to store the data
data = []

# Open and read the TSV file
with open('example.tsv', 'r') as file:
    reader = csv.DictReader(file, delimiter='\t')
    for row in reader:
        data.append(row)

# Display the data
print("Data loaded as list of dictionaries:")
print(data)
```

```
E:\Sammi\ML>python test.py
Data loaded as list of dictionaries:
[{'Name': 'Alice', 'Age': '30', 'City': 'New York'}, {'Name': 'Bob', 'Age
': '25', 'City': 'Los Angeles'}, {'Name': 'Charlie', 'Age': '35', 'City':
 'Chicago'}, {'Name': 'David', 'Age': '40', 'City': 'San Francisco'}]
```

Nhập nguồn dữ liệu

From Files

- **JSON: example.json + pandas**

```
[
    {"Name": "Alice", "Age": 30, "City": "New York"},
    {"Name": "Bob", "Age": 25, "City": "Los Angeles"},
    {"Name": "Charlie", "Age": 35, "City": "Chicago"},
    {"Name": "David", "Age": 40, "City": "San Francisco"}
]
```

```
import pandas as pd

# Load the JSON file into a DataFrame
df = pd.read_json('example.json')

# Display the DataFrame
print("DataFrame:")
print(df)
```

Nhập nguồn dữ liệu

From Files

- **JSON: example.json + json**

```
[
    {"Name": "Alice", "Age": 30, "City": "New York"},
    {"Name": "Bob", "Age": 25, "City": "Los Angeles"},
    {"Name": "Charlie", "Age": 35, "City": "Chicago"},
    {"Name": "David", "Age": 40, "City": "San Francisco"}
]
```

```python
import json

# Load the JSON file
with open('example.json', 'r') as file:
    data = json.load(file)

# Display the data
print("Loaded data:")
print(data)
```

Nhập nguồn dữ liệu

From Files

- **JSONL: example.json + pandas**

```
{"Name": "Alice", "Age": 30, "City": "New York"}
{"Name": "Bob", "Age": 25, "City": "Los Angeles"}
{"Name": "Charlie", "Age": 35, "City": "Chicago"}
{"Name": "David", "Age": 40, "City": "San Francisco"}
```

```
import pandas as pd

# Load the JSONL file into a DataFrame
df = pd.read_json('example.jsonl', lines=True)

# Display the DataFrame
print("DataFrame:")
print(df)
```

Nhập nguồn dữ liệu

From Files

- **JSONL: example.json + json**

```
{"Name": "Alice", "Age": 30, "City": "New York"}
{"Name": "Bob", "Age": 25, "City": "Los Angeles"}
{"Name": "Charlie", "Age": 35, "City": "Chicago"}
{"Name": "David", "Age": 40, "City": "San Francisco"}
```

```python
import json

# Initialize an empty list to store the data
data = []

# Open and read the JSONL file
with open('example.jsonl', 'r') as file:
    for line in file:
        # Parse each line as a JSON object and append it to the list
        data.append(json.loads(line))

# Display the loaded data
print("Loaded data:")
print(data)
```

Xuất nguồn dữ liệu

From Files

- **CSV/TSV/JSON/JSONL + pandas**

```python
import pandas as pd

# Sample DataFrame
df = pd.DataFrame({
    'Name': ['Alice', 'Bob', 'Charlie', 'David'],
    'Age': [30, 25, 35, 40],
    'City': ['New York', 'Los Angeles', 'Chicago', 'San Francisco']
})

# Write to CSV
df.to_csv('example.csv', index=False)

# Write to TSV
df.to_csv('example.tsv', sep='\t', index=False)

# Write to JSON
df.to_json('example.json', orient='records', lines=False)

# Write to JSON Lines (JSONL)
df.to_json('example.jsonl', orient='records', lines=True)

print("Data successfully written to files.")
```

Xuất nguồn dữ liệu

From Files

- **CSV/TSV/JSON/JSONL + csv & json**

```python
import csv

# Write data to a CSV or TSV file
with open('example.csv', 'w', newline='') as file:
    writer = csv.DictWriter(file, fieldnames=["Name", "Age", "City"])
    #writer = csv.DictWriter(file, fieldnames=["Name", "Age", "City"], delimiter='\t')
    writer.writeheader()
    writer.writerows(data)
```

```python
import json

# Write data to a JSON file
with open('example.json', 'w') as file:
    json.dump(data, file, indent=4)  # indent=4 for pretty-printing

# Write data to a JSON Lines (JSONL) file
with open('example.jsonl', 'w') as file:
    for record in data:
        json.dump(record, file)
        file.write('\n')  # Write a newline character after each JSON object
```

Thảo luận

CSV vs. TSV vs. JSON vs. JSONL

- Which one? Why?