

Different Ways to Load a Dataset in Python

In Python, datasets can be loaded using various methods depending on the format and the libraries. Below are some common ways to load datasets in Python.

1 Loading a CSV File

CSV (Comma-Separated Values) is a common format for datasets. Here are two common methods:

1.1 Using Pandas

```
import pandas as pd

# Load CSV file into a DataFrame
data = pd.read_csv('dataset.csv')
print(data.head())
```

1.2 Using NumPy

```
import numpy as np

# Load CSV file into a NumPy array
data = np.genfromtxt('dataset.csv', delimiter=',', skip_header=1)
print(data[:5])
```

2 Loading an Excel File

2.1 Using Pandas

```
import pandas as pd

# Load Excel file into a DataFrame
data = pd.read_excel('dataset.xlsx', sheet_name='Sheet1')
```

```
print(data.head())
```

3 Loading a JSON File

3.1 Using Pandas

```
import pandas as pd

# Load JSON file into a DataFrame
data = pd.read_json('dataset.json')
print(data.head())
```

3.2 Using Python's built-in json module

```
import json

# Load JSON data from a file
with open('dataset.json', 'r') as f:
    data = json.load(f)
print(data)
```

4 Loading a Dataset from SQL Databases

4.1 Using SQLite with Pandas

```
import sqlite3
import pandas as pd

# Connect to the database
conn = sqlite3.connect('database.db')

# Load data from a SQL query into a DataFrame
data = pd.read_sql_query("SELECT * FROM tablename", conn)
print(data.head())
```

4.2 Using SQLAlchemy

```
from sqlalchemy import create_engine
import pandas as pd
```

```
# Create engine and load data
engine = create_engine('sqlite:///database.db')
data = pd.read_sql("SELECT * FROM tablename", engine)
print(data.head())
```

5 Loading a Dataset from a URL

```
import pandas as pd

# Load CSV from URL
url = 'https://example.com/dataset.csv'
data = pd.read_csv(url)
print(data.head())
```

6 Loading Datasets from Python Libraries

6.1 Using Scikit-learn

```
from sklearn.datasets import load_iris

# Load the Iris dataset
data = load_iris()
print(data.data[:5]) # Feature values
print(data.target[:5]) # Labels
```

6.2 Using Seaborn

```
import seaborn as sns

# Load a built-in dataset
data = sns.load_dataset('titanic')
print(data.head())
```

7 Loading a Text File

```
# Load text data from a file
with open('data.txt', 'r') as file:
    text_data = file.readlines()
print(text_data[:5])
```

8 Loading Data from APIs

```
import requests

# Fetch data from an API
response = requests.get('https://api.example.com/data')
data = response.json() # Assuming the response is in JSON format
print(data)
```

9 Loading Image Datasets

9.1 Using OpenCV

```
import cv2

# Load image data
image = cv2.imread('image.jpg')
print(image.shape)
```

9.2 Using PIL (Pillow)

```
from PIL import Image

# Load image data
image = Image.open('image.jpg')
image.show()
```

10 Loading Custom Datasets for Deep Learning

10.1 Using PyTorch

```
import torch
from torchvision import datasets, transforms

# Define transformation and load dataset
transform = transforms.Compose([transforms.ToTensor()])
dataset = datasets.ImageFolder(root='data_folder', transform=
    ↳ transform)
dataloader = torch.utils.data.DataLoader(dataset, batch_size=32,
    ↳ shuffle=True)
```

```
# Iterate through the dataset
for images, labels in dataloader:
    print(images.shape, labels)
```

10.2 Using TensorFlow

```
import tensorflow as tf

# Load dataset
dataset = tf.keras.preprocessing.image_dataset_from_directory(
    ↪ data_folder, batch_size=32)
for images, labels in dataset:
    print(images.shape, labels)
```