import os

from kaggle.api.kaggle\_api\_extended import KaggleApi

import zipfile

# Step 1: Authenticate the Kaggle API (this should be done already)

api = KaggleApi()

api.authenticate()

# Step 2: Download the dataset

dataset = "hbchaitanyabharadwaj/audio-dataset-with-10-indian-languages"

download\_path = "dataset.zip"

api.dataset\_download\_files(dataset, path=download\_path, unzip=False)

# Step 3: Unzip the downloaded dataset

with zipfile.ZipFile(download\_path, 'r') as zip\_ref:

    zip\_ref.extractall("audio\_dataset")

print("Dataset downloaded and extracted successfully.")

import os

import zipfile

from kaggle.api.kaggle\_api\_extended import KaggleApi

# Step 1: Setup Kaggle API and download dataset

def download\_dataset():

    api = KaggleApi()

    api.authenticate()  # Authenticate using the kaggle.json credentials

    # Dataset URL: https://www.kaggle.com/datasets/hbchaitanyabharadwaj/audio-dataset-with-10-indian-languages

    dataset\_name = "hbchaitanyabharadwaj/audio-dataset-with-10-indian-languages"

    download\_dir = "indian\_language\_audio\_data"

    # Download the dataset

    api.dataset\_download\_files(dataset\_name, path=download\_dir, unzip=True)

# Step 2: Extract files (if zip is present)

def extract\_files(zip\_path):

    try:

        if zipfile.is\_zipfile(zip\_path):

            with zipfile.ZipFile(zip\_path, 'r') as zip\_ref:

                zip\_ref.extractall("extracted\_dataset")

            print("Files extracted successfully.")

        else:

            print("Not a valid zip file.")

    except Exception as e:

        print(f"Error extracting file: {e}")

# Step 3: List audio files and transcripts (assuming the dataset contains .wav and .txt files)

def list\_audio\_files():

    dataset\_path = "extracted\_dataset/audio\_dataset"  # Update this path based on your dataset structure

    try:

        audio\_files = [f for f in os.listdir(dataset\_path) if f.endswith(".wav")]

        transcript\_files = [f for f in os.listdir(dataset\_path) if f.endswith(".txt")]

        print(f"Audio files found: {len(audio\_files)}")

        print(f"Transcript files found: {len(transcript\_files)}")

        return audio\_files, transcript\_files

    except FileNotFoundError as e:

        print(f"Error: {e}")

        return [], []

# Run the steps

download\_dataset()

# After download and extraction, list files

audio\_files, transcript\_files = list\_audio\_files()

import os

import zipfile

from kaggle.api.kaggle\_api\_extended import KaggleApi

# Step 1: Setup Kaggle API and download dataset

def download\_dataset():

    api = KaggleApi()

    api.authenticate()  # Authenticate using the kaggle.json credentials

    # Dataset URL: https://www.kaggle.com/datasets/hbchaitanyabharadwaj/audio-dataset-with-10-indian-languages

    dataset\_name = "hbchaitanyabharadwaj/audio-dataset-with-10-indian-languages"

    download\_dir = "indian\_language\_audio\_data"

    # Download the dataset

    api.dataset\_download\_files(dataset\_name, path=download\_dir, unzip=True)

# Step 2: Extract files (if zip is present)

def extract\_files(zip\_path):

    try:

        if zipfile.is\_zipfile(zip\_path):

            with zipfile.ZipFile(zip\_path, 'r') as zip\_ref:

                zip\_ref.extractall("extracted\_dataset")

            print("Files extracted successfully.")

        else:

            print("Not a valid zip file.")

    except Exception as e:

        print(f"Error extracting file: {e}")

# Step 3: List audio files and transcripts (assuming the dataset contains .wav and .txt files)

def list\_audio\_files():

    dataset\_path = "extracted\_dataset/audio\_dataset"  # Update this path based on your dataset structure

    try:

        audio\_files = [f for f in os.listdir(dataset\_path) if f.endswith(".wav")]

        transcript\_files = [f for f in os.listdir(dataset\_path) if f.endswith(".txt")]

        print(f"Audio files found: {len(audio\_files)}")

        print(f"Transcript files found: {len(transcript\_files)}")

        return audio\_files, transcript\_files

    except FileNotFoundError as e:

        print(f"Error: {e}")

        return [], []

# Run the steps

download\_dataset()

# After download and extraction, list files

audio\_files, transcript\_files = list\_audio\_files()

from kaggle.api.kaggle\_api\_extended import KaggleApi

# Authenticate and verify the setup

api = KaggleApi()

api.authenticate()

mport shutil

# Move the kaggle.json to the right directory

shutil.move("kaggle.json", "/root/.kaggle/kaggle.json")