import pandas as pd

import os

# Specify the folder path and output file path on your desktop

desktop\_path = os.path.expanduser("~/Desktop")

folder\_path = os.path.join(desktop\_path, 'output')

output\_file = os.path.join(desktop\_path, 'consolidated\_output.csv')

# List all CSV files in the folder

csv\_files = [file for file in os.listdir(folder\_path) if file.endswith('.csv')]

# Create an empty list to hold DataFrames

data\_frames = []

# Iterate through the CSV files and read them into DataFrames

for csv\_file in csv\_files:

    file\_path = os.path.join(folder\_path, csv\_file)

    data = pd.read\_csv(file\_path)

    data\_frames.append(data)

# Concatenate the list of DataFrames into a single DataFrame

consolidated\_data = pd.concat(data\_frames, ignore\_index=True)

# Save the consolidated data to a single CSV file on your desktop

consolidated\_data.to\_csv(output\_file, index=False)

print(f'Consolidated data saved to {output\_file}')

import pandas as pd

import os

# Specify the file path and output folder on your desktop

desktop\_path = os.path.expanduser("~/Desktop")

file\_path = os.path.join(desktop\_path, 'sample.csv')

output\_folder = os.path.join(desktop\_path, 'output')

chunksize = 100  # Number of rows per chunk

# Create the output folder if it doesn't exist

if not os.path.exists(output\_folder):

    os.makedirs(output\_folder)

# Create an iterator to read the CSV file in chunks

chunk\_iterator = pd.read\_csv(file\_path, chunksize=chunksize)

# Initialize a variable to start chunk numbering from 1

chunk\_number = 1

# Process each chunk one at a time and save to the output folder

for chunk in chunk\_iterator:

    # Define the output file path for the chunk

    output\_file = os.path.join(output\_folder, f'chunk\_{chunk\_number}.csv')

    # Save the chunk as a separate CSV file in the output folder

    chunk.to\_csv(output\_file, index=False)

    print(f'Saved chunk {chunk\_number} to {output\_file}')

    # Increment the chunk number

    chunk\_number += 1