import pandas as pd

from datetime import datetime, timedelta

import random

def generate\_date\_pairs(num\_samples):

    data = []

    base\_date = datetime(2024, 1, 1)

    date\_formats = ["%m-%d-%Y", "%d-%m-%Y", "%Y-%m-%d"]

    for \_ in range(num\_samples):

        start\_date = base\_date + timedelta(days=random.randint(1, 365))

        # Decide randomly whether to create a positive or negative example

        if random.random() > 0.5:  # Positive example

            days\_diff = random.randint(-4, 4)  # Difference within 4 days

        else:

            days\_diff = random.choice([random.randint(5, 30), random.randint(-30, -5)])  # Significant difference

        end\_date = start\_date + timedelta(days=days\_diff)

        format1, format2 = random.sample(date\_formats, 2)

        source = start\_date.strftime(format1)

        destination = end\_date.strftime(format2)

        label = 1 if abs(days\_diff) <= 4 else 0  # Label based on the day difference

        data.append([source, destination, label])

    return pd.DataFrame(data, columns=["source", "destination", "label"])

# Generate and save dataset

df = generate\_date\_pairs(1000)

df.to\_csv("date\_matching\_dataset\_Final.csv", index=False)

import pandas as pd

from datetime import datetime, timedelta

import random

def generate\_date\_pairs(num\_samples):

    data = []

    base\_date = datetime(2024, 1, 1)

    date\_formats = ["%m-%d-%Y", "%d-%m-%Y", "%Y-%m-%d"]

    for \_ in range(num\_samples):

        start\_date = base\_date + timedelta(days=random.randint(1, 365))

        # Decide randomly whether to create a positive or negative example

        if random.random() > 0.5:  # 50% chance

            # Positive example: difference within 4 days

            days\_diff = random.randint(-4, 4)

        else:

            # Negative example: difference more than 4 days

            days\_diff = random.choice([random.randint(5, 30), random.randint(-30, -5)])

        end\_date = start\_date + timedelta(days=days\_diff)

        # Randomly choose date formats

        format1, format2 = random.sample(date\_formats, 2)

        # Convert dates to strings with chosen formats

        source = start\_date.strftime(format1)

        destination = end\_date.strftime(format2)

        # Label: 1 if they are considered the same, 0 otherwise

        label = 1 if abs(days\_diff) <= 4 else 0

        data.append([source, destination, label])

    return data

# Generate balanced data

num\_samples = 2000  # Total samples

data = generate\_date\_pairs(num\_samples)

# Create DataFrame

df = pd.DataFrame(data, columns=["source", "destination", "label"])

# Save to CSV

df.to\_csv("date\_matching\_dataset.csv", index=False)