from sklearn.ensemble import RandomForestClassifier

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score

import pandas as pd

import pandas as pd

import random

# Generate synthetic data

data = {

"Openness": [random.randint(1, 5) for \_ in range(100)],

"Conscientiousness": [random.randint(1, 5) for \_ in range(100)],

"Extraversion": [random.randint(1, 5) for \_ in range(100)],

"Agreeableness": [random.randint(1, 5) for \_ in range(100)],

"Neuroticism": [random.randint(1, 5) for \_ in range(100)],

}

# Map personality traits to color palettes

def get\_color\_palette(openness, conscientiousness, extraversion, agreeableness, neuroticism):

if extraversion > 3:

return ["Red", "Orange"]

elif openness > 4:

return ["Purple", "Magenta"]

elif neuroticism > 3:

return ["Gray", "Black"]

elif agreeableness > 4:

return ["Yellow", "Pink"]

else:

return ["Blue", "Green"]

data["Color\_Palette"] = [

get\_color\_palette(row["Openness"], row["Conscientiousness"], row["Extraversion"], row["Agreeableness"], row["Neuroticism"])

for \_, row in pd.DataFrame(data).iterrows()

]

# Save to CSV

df = pd.DataFrame(data)

df.to\_csv("personality\_color\_data.csv", index=False)

print("Dataset saved as 'personality\_color\_data.csv'")

# Load the dataset

data = pd.read\_csv("personality\_color\_data.csv")

X = data.drop("Color\_Palette", axis=1) # Changed 'color\_palette' to 'Color\_Palette'

y = data["Color\_Palette"] # Changed 'color\_palette' to 'Color\_Palette'

# Train-test split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Train the model

model = RandomForestClassifier()

model.fit(X\_train, y\_train)

# Evaluate the model

y\_pred = model.predict(X\_test)

print(f"Accuracy: {accuracy\_score(y\_test, y\_pred)}")

import streamlit as st

import numpy as np

# Pre-defined questions

questions = [

"I enjoy large social gatherings.",

"I often try new and unusual activities.",

"I am detail-oriented and organized.",

"I often feel anxious or stressed.",

"I trust people easily."

]

# User responses

responses = []

st.title("Personality-Based Color Palette Recommender")

st.write("Answer the questions below to get a personalized color palette recommendation!")

# Display questions

for i, question in enumerate(questions):

response = st.radio(question, ["Strongly Disagree", "Disagree", "Neutral", "Agree", "Strongly Agree"], key=f"q{i}")

responses.append(response)

# Map responses to numerical values

response\_mapping = {

"Strongly Disagree": 1,

"Disagree": 2,

"Neutral": 3,

"Agree": 4,

"Strongly Agree": 5

}

numerical\_responses = [response\_mapping[r] for r in responses]

# Predict personality and recommend a color palette

if st.button("Get Color Palette"):

predicted\_palette = model.predict([numerical\_responses])

st.write(f"Your recommended color palette is: {predicted\_palette[0]}")

personality\_to\_palette = {

"Extraverted": ["Red", "Orange", "Yellow"],

"Introverted": ["Blue", "Green", "Gray"],

"Creative": ["Purple", "Magenta", "Turquoise"],

"Calm": ["Beige", "Pastel Green", "Soft Pink"]

}