import tkinter as tk

from tkinter import messagebox

import matplotlib.pyplot as plt

def calculate\_freshwater\_influx():

try:

# Get input values for each parameter for the years 2018 to 2023

sea\_ice\_concentration = [float(entry\_sea\_ice\_concentration\_2018.get()), float(entry\_sea\_ice\_concentration\_2019.get()),

float(entry\_sea\_ice\_concentration\_2020.get()), float(entry\_sea\_ice\_concentration\_2021.get()),

float(entry\_sea\_ice\_concentration\_2022.get()), float(entry\_sea\_ice\_concentration\_2023.get())]

sea\_surface\_temperature = [float(entry\_sea\_surface\_temperature\_2018.get()), float(entry\_sea\_surface\_temperature\_2019.get()),

float(entry\_sea\_surface\_temperature\_2020.get()), float(entry\_sea\_surface\_temperature\_2021.get()),

float(entry\_sea\_surface\_temperature\_2022.get()), float(entry\_sea\_surface\_temperature\_2023.get())]

air\_temperature = [float(entry\_air\_temperature\_2018.get()), float(entry\_air\_temperature\_2019.get()),

float(entry\_air\_temperature\_2020.get()), float(entry\_air\_temperature\_2021.get()),

float(entry\_air\_temperature\_2022.get()), float(entry\_air\_temperature\_2023.get())]

wind\_speed = [float(entry\_wind\_speed\_2018.get()), float(entry\_wind\_speed\_2019.get()),

float(entry\_wind\_speed\_2020.get()), float(entry\_wind\_speed\_2021.get()),

float(entry\_wind\_speed\_2022.get()), float(entry\_wind\_speed\_2023.get())]

sea\_ice\_thickness = [float(entry\_sea\_ice\_thickness\_2018.get()), float(entry\_sea\_ice\_thickness\_2019.get()),

float(entry\_sea\_ice\_thickness\_2020.get()), float(entry\_sea\_ice\_thickness\_2021.get()),

float(entry\_sea\_ice\_thickness\_2022.get()), float(entry\_sea\_ice\_thickness\_2023.get())]

salinity = [float(entry\_salinity\_2018.get()), float(entry\_salinity\_2019.get()),

float(entry\_salinity\_2020.get()), float(entry\_salinity\_2021.get()),

float(entry\_salinity\_2022.get()), float(entry\_salinity\_2023.get())]

# Calculate Volume of Melted Sea Ice for each year

volume\_melted\_sea\_ice = [(conc / 100) \* thickness for conc, thickness in zip(sea\_ice\_concentration, sea\_ice\_thickness)]

# Get the reference salinity from the entry field

reference\_salinity = float(entry\_reference\_salinity.get())

# Calculate Freshwater Influx for each year

freshwater\_influx = [volume \* (sal - reference\_salinity) for volume, sal in zip(volume\_melted\_sea\_ice, salinity)]

# Display the results

result\_label.config(text="Predicted Freshwater Influx for the years 2018 to 2023:")

for year, influx in zip(range(2018, 2024), freshwater\_influx):

result\_label.config(text=result\_label.cget("text") + f"\n{year}: {influx:.2f} cubic kilometers")

except ValueError:

messagebox.showerror("Error", "Please enter valid numeric values for all parameters.")

def draw\_histogram():

# Get the salinity values for the years 2018 to 2023

salinity = [float(entry\_salinity\_2018.get()), float(entry\_salinity\_2019.get()),

float(entry\_salinity\_2020.get()), float(entry\_salinity\_2021.get()),

float(entry\_salinity\_2022.get()), float(entry\_salinity\_2023.get())]

# Create the histogram

plt.figure(figsize=(8, 6))

plt.hist(salinity, bins=10, edgecolor='black', color='skyblue')

plt.xlabel('Salinity (psu)')

plt.ylabel('Frequency')

plt.title('Distribution of Salinity for the years 2018 to 2023')

plt.grid(True)

plt.tight\_layout()

plt.show()

# Create the GUI

root = tk.Tk()

root.title("Data-Driven Emulator: Predict Freshwater Influx in the Arctic Ocean")

# Create entry fields for each parameter for the years 2018 to 2023

entry\_sea\_ice\_concentration\_2018 = tk.Entry(root, width=10)

entry\_sea\_ice\_concentration\_2019 = tk.Entry(root, width=10)

entry\_sea\_ice\_concentration\_2020 = tk.Entry(root, width=10)

entry\_sea\_ice\_concentration\_2021 = tk.Entry(root, width=10)

entry\_sea\_ice\_concentration\_2022 = tk.Entry(root, width=10)

entry\_sea\_ice\_concentration\_2023 = tk.Entry(root, width=10)

entry\_sea\_surface\_temperature\_2018 = tk.Entry(root, width=10)

entry\_sea\_surface\_temperature\_2019 = tk.Entry(root, width=10)

entry\_sea\_surface\_temperature\_2020 = tk.Entry(root, width=10)

entry\_sea\_surface\_temperature\_2021 = tk.Entry(root, width=10)

entry\_sea\_surface\_temperature\_2022 = tk.Entry(root, width=10)

entry\_sea\_surface\_temperature\_2023 = tk.Entry(root, width=10)

entry\_air\_temperature\_2018 = tk.Entry(root, width=10)

entry\_air\_temperature\_2019 = tk.Entry(root, width=10)

entry\_air\_temperature\_2020 = tk.Entry(root, width=10)

entry\_air\_temperature\_2021 = tk.Entry(root, width=10)

entry\_air\_temperature\_2022 = tk.Entry(root, width=10)

entry\_air\_temperature\_2023 = tk.Entry(root, width=10)

entry\_wind\_speed\_2018 = tk.Entry(root, width=10)

entry\_wind\_speed\_2019 = tk.Entry(root, width=10)

entry\_wind\_speed\_2020 = tk.Entry(root, width=10)

entry\_wind\_speed\_2021 = tk.Entry(root, width=10)

entry\_wind\_speed\_2022 = tk.Entry(root, width=10)

entry\_wind\_speed\_2023 = tk.Entry(root, width=10)

entry\_sea\_ice\_thickness\_2018 = tk.Entry(root, width=10)

entry\_sea\_ice\_thickness\_2019 = tk.Entry(root, width=10)

entry\_sea\_ice\_thickness\_2020 = tk.Entry(root, width=10)

entry\_sea\_ice\_thickness\_2021 = tk.Entry(root, width=10)

entry\_sea\_ice\_thickness\_2022 = tk.Entry(root, width=10)

entry\_sea\_ice\_thickness\_2023 = tk.Entry(root, width=10)

entry\_salinity\_2018 = tk.Entry(root, width=10)

entry\_salinity\_2019 = tk.Entry(root, width=10)

entry\_salinity\_2020 = tk.Entry(root, width=10)

entry\_salinity\_2021 = tk.Entry(root, width=10)

entry\_salinity\_2022 = tk.Entry(root, width=10)

entry\_salinity\_2023 = tk.Entry(root, width=10)

entry\_reference\_salinity = tk.Entry(root, width=10)

# Add labels for each parameter for the years 2018 to 2023

label\_sea\_ice\_concentration = tk.Label(root, text="Sea Ice Concentration (%)")

label\_sea\_surface\_temperature = tk.Label(root, text="Sea Surface Temperature (°C)")

label\_air\_temperature = tk.Label(root, text="Air Temperature (°C)")

label\_wind\_speed = tk.Label(root, text="Wind Speed (m/s)")

label\_sea\_ice\_thickness = tk.Label(root, text="Sea Ice Thickness (m)")

label\_salinity = tk.Label(root, text="Salinity (psu)")

label\_reference\_salinity = tk.Label(root, text="Reference Salinity (psu)")

# Create buttons for calculation and histogram

calculate\_button = tk.Button(root, text="Calculate Freshwater Influx", command=calculate\_freshwater\_influx)

histogram\_button = tk.Button(root, text="Draw Salinity Histogram", command=draw\_histogram)

# Create a label to display the result

result\_label = tk.Label(root, text="", font=("Helvetica", 14, "bold"))

# Layout the widgets using grid

label\_sea\_ice\_concentration.grid(row=0, column=0)

label\_sea\_surface\_temperature.grid(row=1, column=0)

label\_air\_temperature.grid(row=2, column=0)

label\_wind\_speed.grid(row=3, column=0)

label\_sea\_ice\_thickness.grid(row=4, column=0)

label\_salinity.grid(row=5, column=0)

entry\_sea\_ice\_concentration\_2018.grid(row=0, column=1)

entry\_sea\_ice\_concentration\_2019.grid(row=0, column=2)

entry\_sea\_ice\_concentration\_2020.grid(row=0, column=3)

entry\_sea\_ice\_concentration\_2021.grid(row=0, column=4)

entry\_sea\_ice\_concentration\_2022.grid(row=0, column=5)

entry\_sea\_ice\_concentration\_2023.grid(row=0, column=6)

entry\_sea\_surface\_temperature\_2018.grid(row=1, column=1)

entry\_sea\_surface\_temperature\_2019.grid(row=1, column=2)

entry\_sea\_surface\_temperature\_2020.grid(row=1, column=3)

entry\_sea\_surface\_temperature\_2021.grid(row=1, column=4)

entry\_sea\_surface\_temperature\_2022.grid(row=1, column=5)

entry\_sea\_surface\_temperature\_2023.grid(row=1, column=6)

entry\_air\_temperature\_2018.grid(row=2, column=1)

entry\_air\_temperature\_2019.grid(row=2, column=2)

entry\_air\_temperature\_2020.grid(row=2, column=3)

entry\_air\_temperature\_2021.grid(row=2, column=4)

entry\_air\_temperature\_2022.grid(row=2, column=5)

entry\_air\_temperature\_2023.grid(row=2, column=6)

entry\_wind\_speed\_2018.grid(row=3, column=1)

entry\_wind\_speed\_2019.grid(row=3, column=2)

entry\_wind\_speed\_2020.grid(row=3, column=3)

entry\_wind\_speed\_2021.grid(row=3, column=4)

entry\_wind\_speed\_2022.grid(row=3, column=5)

entry\_wind\_speed\_2023.grid(row=3, column=6)

entry\_sea\_ice\_thickness\_2018.grid(row=4, column=1)

entry\_sea\_ice\_thickness\_2019.grid(row=4, column=2)

entry\_sea\_ice\_thickness\_2020.grid(row=4, column=3)

entry\_sea\_ice\_thickness\_2021.grid(row=4, column=4)

entry\_sea\_ice\_thickness\_2022.grid(row=4, column=5)

entry\_sea\_ice\_thickness\_2023.grid(row=4, column=6)

entry\_salinity\_2018.grid(row=5, column=1)

entry\_salinity\_2019.grid(row=5, column=2)

entry\_salinity\_2020.grid(row=5, column=3)

entry\_salinity\_2021.grid(row=5, column=4)

entry\_salinity\_2022.grid(row=5, column=5)

entry\_salinity\_2023.grid(row=5, column=6)

label\_reference\_salinity.grid(row=6, column=0)

entry\_reference\_salinity.grid(row=6, column=1, columnspan=2)

calculate\_button.grid(row=7, column=0, columnspan=3, pady=10)

histogram\_button.grid(row=7, column=3, columnspan=4, pady=10)

result\_label.grid(row=8, column=0, columnspan=7)

root.mainloop()