**G3.29**

**HỌ TÊN: Nguyễn Diệu Hương MSSV: 21110489 KÝ TÊN:**

**TRƯỜNG: HCMUTE**

**MÔN: Lập trình Python**

**NGÀY: 26/04/2023 (BUỔI HỌC SỐ 13)**

**Riêng buổi học Phòng máy ghi thêm, SỐ MÁY: PHÒNG MÁY:**

============

**Buổi 13\_... = Bài tập 3 Phần 2 = GĐ 3 EDA: Phân tích dữ liệu thăm dò**

**Giai đoạn 3 = Bước 7, 8, 9, 10**

# **MỘT SỐ LÝ THUYẾT**

**CÁC THƯ VIỆN CHÍNH**

from sklearn import preprocessing # Thư viện tiền xử lý DL (XL ngoại lệ: Isolated)

from sklearn.feature\_selection import SelectKBest, chi2 # Nạp hàm Thư viện hỗ trợ Mô hình phân tích dữ liệu thăm dò

**2.** **FULL CODES THAM KHẢO**

**(Chỉ là phần core: sv phát triển & hoàn thiện lên Form = Bài tập ĐAHP theo cá nhân + nên cá nhân hóa thông tin bài làm)**

# PHẦN BUỔI 4 = TIỀN XỬ LÝ

from tkinter import messagebox

import numpy as np

import pandas as pd

import tkinter as tk

from tkinter import ttk

from scipy import stats

from sklearn.preprocessing import StandardScaler

from sklearn.feature\_selection import SelectKBest, chi2, f\_classif

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error

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

master = tk.Tk()

master.geometry('1500x800')

master.title("29\_Nguyễn Diệu Hương")

master.resizable(tk.FALSE, tk.FALSE)

df0 = pd.read\_csv('./CO2\_Emissions.csv')

df = df0.copy()

def Output():

    df\_29 = pd.read\_csv('./CO2\_Emissions.csv')

    selected\_step = steps.get()

    if selected\_step == 'Xử lý cột':

        output\_text.delete('1.0', tk.END)

        df\_29 = df.drop(*columns*=['Country Name'],*axis*=1)

        output\_text.insert(tk.END, df\_29)

    elif selected\_step == 'Xử lý dòng':

        output\_text.delete('1.0', tk.END)

        df\_29  =  df.dropna(*how* = 'any')

        output\_text.insert(tk.END, df\_29)

    elif selected\_step == 'Xử lý cá biệt':

        output\_text.delete('1.0', tk.END)

        z = np.abs(stats.zscore(df.\_get\_numeric\_data()))

        df\_29 = df[(z > 0.0001 ).all(*axis*=1)]

        output\_text.insert(tk.END, "MA TRAN Z-SCORE:\n")

        output\_text.insert(tk.END, *str*(z) + "\n")

    elif selected\_step == 'Chuẩn hóa dữ liệu':

        output\_text.delete('1.0', tk.END)

        select\_cols=df\_29.columns.tolist()

        data\_selected=df\_29.loc[:,select\_cols]

        output\_text.insert(tk.END,"Các cột được chọn để chuẩn hóa dữ liệu: \n")

        output\_text.insert(tk.END,data\_selected)

        scaler = preprocessing.MinMaxScaler()

        scaler.fit(data\_selected)

        df\_29 = pd.DataFrame(scaler.transform(data\_selected), *index*=data\_selected.index, *columns*=data\_selected.columns)

        output\_text.insert(tk.END,"\n")

        output\_text.insert(tk.END,data\_selected.iloc[4:10])

        output\_text.insert(tk.END,"\n")

def FeatureSelection():

    selected = method.get()

    num\_k = *int*(k.get())

    if selected == 'Chi2':

        output\_text.delete('1.0', tk.END)

        a = df0.loc[:, df0.columns != 'Country Name']

        b = df0[['2018']]

        selector = SelectKBest(chi2, *k* = num\_k)

        selector.fit(a,b)

        a\_new = selector.transform(a)

        selected\_columns = a.columns[selector.get\_support(*indices* = True)]

        output\_text.insert('end', 'Các đặc trưng quan trọng: \n' )

        for col in selected\_columns:

            output\_text.insert('end', '-{}\n'.format(col))

        output\_text.insert(tk.END, *str*(a\_new))

        output\_text.insert(tk.END, "\n")

def B9(*selected\_colnames*):

    df\_29 = pd.read\_csv('./CO2\_Emissions.csv')

    df\_29 = df[:, :]

steps\_label = tk.Label(master, *text*='Select preprocessing steps:')

steps\_label.place( *x* = 1275, *y* = 50)

steps\_list = ['Xử lý cột', 'Xử lý dòng', 'Xử lý cá biệt', 'Thay thế giá trị', 'Chuẩn hóa dữ liệu']

steps = ttk.Combobox(master, *values*=steps\_list, *state*='readonly')

steps.place(*x* = 1275 ,*y* =80)

steps.current(0)

output\_button = tk.Button(master, *text*='Output data after preprocessing', *command*=Output, *bg*="black", *fg*="white")

output\_button.place(*x*= 1280 , *y* = 110)

output\_text = tk.Text(master, *bg*="#B9FFFC",  *height*=45, *width*=150)

output\_text.place(*x* = 5, *y* =20)

output\_text.tag\_configure("left", *justify*='left')

scrollbar = tk.Scrollbar(master, *command*= output\_text.xview)

scrollbar.pack(*side* = tk.RIGHT, *fill* = tk.X)

output\_text.config(*xscrollcommand*=scrollbar.set)

method\_label = tk.Label(master, *text*='Select method:')

method\_label.place( *x* = 1275, *y* = 150)

method\_list = ['Chi2' , 'f\_classif']

method = ttk.Combobox(master, *values*=method\_list, *state*='readonly')

method.place(*x* = 1275 ,*y* = 180)

method.current(0)

k\_label = tk.Label(master, *text* = "Choose the k: ")

k\_label.place (*x* = 1280, *y*= 210)

k = tk.Entry(master, *width*=10)

k.place(*x* = 1280, *y*= 240)

method\_button = tk.Button(master, *text*='Output data after feature selection',  *bg*="black", *fg*="white")

method\_button.place(*x*= 1280 , *y* = 270)

selected\_columns = pd.DataFrame(*columns*=['column'])

master.mainloop()