Tên: Cao Trần Hoàng Vũ MSSV; 2274802011015

A paper with text on it

Description automatically generated

A black and white text with white text

Description automatically generated

**Câu 1**

import pandas as pd

file\_path = r'C:\Users\Admin\OneDrive\Nhap mon du lieu sau\dulieuxettuyendaihoc.csv'

df = pd.read\_csv(file\_path)

print("First few rows of the dataframe:")

print(df.head())

dt\_columns = df.select\_dtypes(include=["object", "category"]).columns

dl\_columns = set(df.columns) - set(dt\_columns)

dt\_df = df.drop(columns=dt\_columns)

print("First few rows of the numeric dataframe:")

print(dt\_df.head())\

**Kết quả:**

A screenshot of a computer

Description automatically generated

**Câu 2**

import pandas as pd

file\_path = r'C:\Users\Admin\OneDrive\Nhap mon du lieu sau\dulieuxettuyendaihoc.csv'

df = pd.read\_csv(file\_path)

print("First few rows of the dataframe:")

print(df.head())

dt\_columns = df.select\_dtypes(include=["object", "category"]).columns

dl\_columns = set(df.columns) - set(dt\_columns)

dt\_df = df.drop(columns=dt\_columns)

thang\_do = {

    'Cột': ['Giới tính', 'Mức độ hài lòng', 'Nhiệt độ', 'Chiều cao', 'Thu nhập'],

    'Thang đo': ['Danh mục (Nominal)', 'Thứ tự (Ordinal)', 'Khoảng (Interval)', 'Tỷ lệ (Ratio)', 'Tỷ lệ (Ratio)']

}

thang\_do\_df = pd.DataFrame(thang\_do)

print(thang\_do\_df)

**Kết quả:**

A screenshot of a computer

Description automatically generated

**Câu 3**

import pandas as pd

# Đường dẫn tới file CSV

file\_path = r'C:\Users\Admin\OneDrive\Nhap mon du lieu sau\dulieuxettuyendaihoc.csv'

# Đọc file CSV

df = pd.read\_csv(file\_path)

# Hiển thị 10 dòng đầu tiên của dataframe

print("First 10 rows of the dataframe:")

print(df.head(10))

# Hiển thị 10 dòng cuối cùng của dataframe

print("Last 10 rows of the dataframe:")

print(df.tail(10))

**Kết quả:**

A screenshot of a computer

Description automatically generated

**Câu 4**

import pandas as pd

file\_path = r'C:\Users\Admin\OneDrive\Nhap mon du lieu sau\dulieuxettuyendaihoc.csv'

df = pd.read\_csv(file\_path)

print("First few rows of the dataframe:")

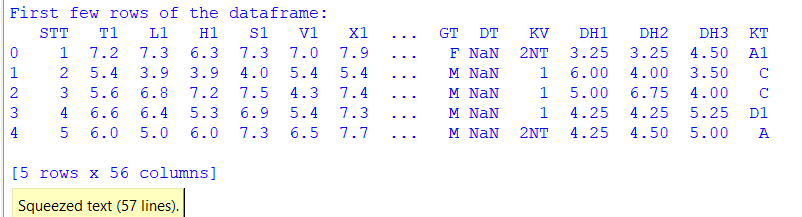
print(df.head())

dt\_columns = df.select\_dtypes(include=["object", "category"]).columns

dl\_columns = set(df.columns) - set(dt\_columns)

dt\_df = df.drop(columns=dt\_columns)

**Kết quả**



**Câu 5**

import pandas as pd

file\_path = r'C:\Users\Admin\OneDrive\Nhap mon du lieu sau\dulieuxettuyendaihoc.csv'

df = pd.read\_csv(file\_path)

print(df.info()) # Kiểm tra kiểu dữ liệu, số lượng giá trị null, v.v.

print(df.describe()) # Thống kê các số liệu mô tả (mean, std, min, max, ...)

numeric\_cols = df.select\_dtypes(include=['number']).columns

t1\_mean = df['T1'].mean()

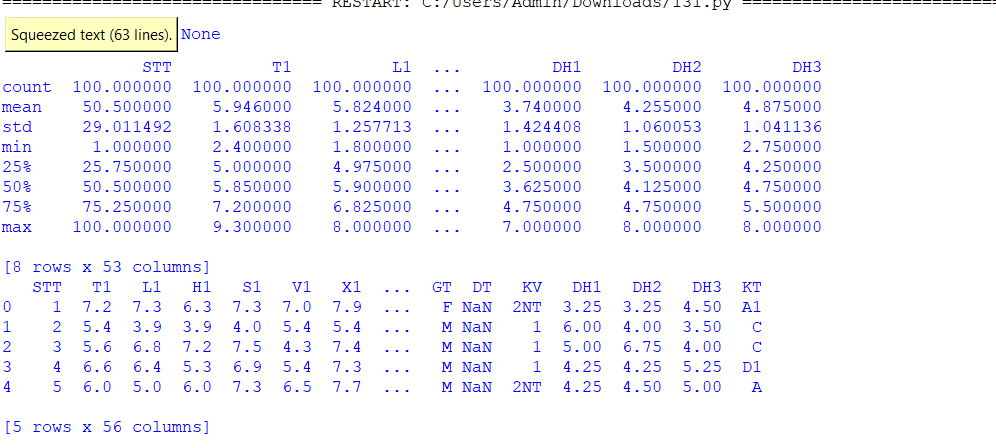
df['T1'] = df['T1'].fillna(t1\_mean)

print(df.head())

# Lưu DataFrame vào file CSV mới (nếu cần)

df.to\_csv('dulieuxettuyendaihoc\_cleaned.csv', index=False)

**Kết quả:**



**Câu 6**

import pandas as pd

file\_path = r'C:\Users\Admin\OneDrive\Nhap mon du lieu sau\dulieuxettuyendaihoc.csv'

df = pd.read\_csv(file\_path)

df['TBM1'] = (df['T1'] \* 2 + df['L1'] + df['H1'] + df['S1'] + df['V1'] + df['X1'] + df['D1'] + df['N1']) / 10

df['TBM2'] = (df['T2'] \* 2 + df['L2'] + df['H2'] + df['S2'] + df['V2'] + df['X2'] + df['D2'] + df['N2']) / 10

df['TBM3'] = (df['T6'] \* 2 + df['L6'] + df['H6'] + df['S6'] + df['V6'] + df['X6'] + df['D6'] + df['N6']) / 10

print(df.head)

**Kết quả:**

A screenshot of a computer

Description automatically generated

**Câu 7**

import pandas as pd

file\_path = r'C:\Users\Admin\OneDrive\Nhap mon du lieu sau\dulieuxettuyendaihoc.csv'

df = pd.read\_csv(file\_path)

# Tính toán điểm trung bình môn cho các năm lớp

df['TBM1'] = (df['T1'] \* 2 + df['L1'] + df['H1'] + df['S1'] + df['V1'] + df['X1'] + df['D1'] + df['N1']) / 10

df['TBM2'] = (df['T2'] \* 2 + df['L2'] + df['H2'] + df['S2'] + df['V2'] + df['X2'] + df['D2'] + df['N2']) / 10

df['TBM3'] = (df['T6'] \* 2 + df['L6'] + df['H6'] + df['S6'] + df['V6'] + df['X6'] + df['D6'] + df['N6']) / 10

def xep\_loai(diem):

    if diem < 5.0:

        return 'Y'

    elif 5.0 <= diem < 6.5:

        return 'TB'

    elif 6.5 <= diem < 8.0:

        return 'K'

    elif 8.0 <= diem < 9.0:

        return 'G'

    else:

        return 'XS'

# Áp dụng hàm cho từng cột TBM để tạo cột xếp loại tương ứng

df['XL1'] = df['TBM1'].apply(xep\_loai)

df['XL2'] = df['TBM2'].apply(xep\_loai)

df['XL3'] = df['TBM3'].apply(xep\_loai)

print(df.head())

**Kết quả:**

A screenshot of a computer

Description automatically generated

**Câu 8**

import pandas as pd

file\_path = r'C:\Users\Admin\OneDrive\Nhap mon du lieu sau\dulieuxettuyendaihoc.csv'

df = pd.read\_csv(file\_path)

df['TBM1'] = (df['T1'] \* 2 + df['L1'] + df['H1'] + df['S1'] + df['V1'] + df['X1'] + df['D1'] + df['N1']) / 10

df['TBM2'] = (df['T2'] \* 2 + df['L2'] + df['H2'] + df['S2'] + df['V2'] + df['X2'] + df['D2'] + df['N2']) / 10

df['TBM3'] = (df['T6'] \* 2 + df['L6'] + df['H6'] + df['S6'] + df['V6'] + df['X6'] + df['D6'] + df['N6']) / 10

def xep\_loai(diem):

if diem < 5.0:

return 'Y'

elif 5.0 <= diem < 6.5:

return 'TB'

elif 6.5 <= diem < 8.0:

return 'K'

elif 8.0 <= diem < 9.0:

return 'G'

else:

return 'XS'

df['XL1'] = df['TBM1'].apply(xep\_loai)

df['XL2'] = df['TBM2'].apply(xep\_loai)

df['XL3'] = df['TBM3'].apply(xep\_loai)

def min\_max\_normalization(x, old\_min, old\_max, new\_min, new\_max):

return ((x - old\_min) / (old\_max - old\_min)) \* (new\_max - new\_min) + new\_min

min\_TBM1 = df['TBM1'].min()

max\_TBM1 = df['TBM1'].max()

min\_TBM2 = df['TBM2'].min()

max\_TBM2 = df['TBM2'].max()

min\_TBM3 = df['TBM3'].min()

max\_TBM3 = df['TBM3'].max()

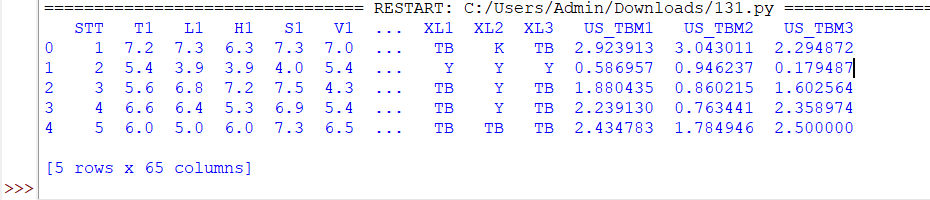
df['US\_TBM1'] = df['TBM1'].apply(lambda x: min\_max\_normalization(x, min\_TBM1, max\_TBM1, 0, 4))

df['US\_TBM2'] = df['TBM2'].apply(lambda x: min\_max\_normalization(x, min\_TBM2, max\_TBM2, 0, 4))

df['US\_TBM3'] = df['TBM3'].apply(lambda x: min\_max\_normalization(x, min\_TBM3, max\_TBM3, 0, 4))

print(df.head())

**Kết quả:**



**Câu 9**

#cau9

import pandas as pd

file\_path = r'C:\Users\Admin\OneDrive\Nhap mon du lieu sau\dulieuxettuyendaihoc.csv'

df = pd.read\_csv(file\_path)

df['TBM1'] = (df['T1'] \* 2 + df['L1'] + df['H1'] + df['S1'] + df['V1'] + df['X1'] + df['D1'] + df['N1']) / 10

df['TBM2'] = (df['T2'] \* 2 + df['L2'] + df['H2'] + df['S2'] + df['V2'] + df['X2'] + df['D2'] + df['N2']) / 10

df['TBM3'] = (df['T6'] \* 2 + df['L6'] + df['H6'] + df['S6'] + df['V6'] + df['X6'] + df['D6'] + df['N6']) / 10

def min\_max\_normalization(x, old\_min, old\_max, new\_min, new\_max):

return ((x - old\_min) / (old\_max - old\_min)) \* (new\_max - new\_min) + new\_min

min\_TBM1 = df['TBM1'].min()

max\_TBM1 = df['TBM1'].max()

min\_TBM2 = df['TBM2'].min()

max\_TBM2 = df['TBM2'].max()

min\_TBM3 = df['TBM3'].min()

max\_TBM3 = df['TBM3'].max()

df['US\_TBM1'] = df['TBM1'].apply(lambda x: min\_max\_normalization(x, min\_TBM1, max\_TBM1, 0, 4))

df['US\_TBM2'] = df['TBM2'].apply(lambda x: min\_max\_normalization(x, min\_TBM2, max\_TBM2, 0, 4))

df['US\_TBM3'] = df['TBM3'].apply(lambda x: min\_max\_normalization(x, min\_TBM3, max\_TBM3, 0, 4))

print(df.head())

**Kết quả:**

A screenshot of a computer

Description automatically generated

**Câu 10**

#cau10

import pandas as pd

file\_path = r'C:\Users\Admin\OneDrive\Nhap mon du lieu sau\dulieuxettuyendaihoc.csv'

df = pd.read\_csv(file\_path)

print("First few rows of the dataframe:")

print(df.head())

dt\_columns = df.select\_dtypes(include=["object", "category"]).columns

dl\_columns = set(df.columns) - set(dt\_columns)

dt\_df = df.drop(columns=dt\_columns)

data = {

    'KT': ['A', 'B', 'C', 'A1', 'D'],

    'DH1': [8.0, 5.0, 6.0, 7.0, 4.5],

    'DH2': [7.0, 6.0, 5.5, 6.5, 5.0],

    'DH3': [8.0, 7.0, 6.0, 5.5, 4.0]

}

df = pd.DataFrame(data)

# Hàm xác định kết quả xét tuyển

def calculate\_kqxt(row):

    if row['KT'] in ['A', 'A1']:  # Khối A và A1

        avg = (row['DH1'] \* 2 + row['DH2'] + row['DH3']) / 4

    elif row['KT'] == 'B':  # Khối B

        avg = (row['DH1'] + row['DH2'] \* 2 + row['DH3']) / 4

    else:  # Các khối khác

        avg = (row['DH1'] + row['DH2'] + row['DH3']) / 3

    return 1 if avg >= 5.0 else 0

df['KQXT'] = df.apply(calculate\_kqxt, axis=1)

print("Dữ liệu sau khi thêm biến KQXT:")

print(df)

**Kết quả:**

