Bài tập Thực hành môn Khai phá Dữ liệu

Họ và tên: Huỳnh Nguyễn Thế Dân

MSSV: 21110256

Lớp: 21TTH1

Cài đặt và thực thi mục 1 trên máy tính

```
import pandas as pd
import numpy as np
from sklearn.impute import SimpleImputer
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import KBinsDiscretizer

# Đọc dữ liệu từ tệp arrhythmia.data
df = pd.read_csv("arrhythmia\\arrhythmia.data", delimiter=",",
header=None)

# Hiển thị DataFrame
display(df)

# Kiểm tra giá trị null
print(df.isnull().sum())

# Điền giá trị NaN bằng giá trị trung bình
df["Age"].fillna(df["Age"].mean(), inplace=True)
df["Salary"].fillna(df["Salary"].mean(), inplace=True)

# Kiểm tra giá trị trùng lặp
print(df.duplicated().sum())

# Loại bỏ các giá trị trùng lặp
df = df.drop_duplicates()

# Hiển thị DataFrame sau khi loại bỏ giá trị trùng lặp
display(df)

# Chuyển các giá trị hạng mục thành 0/1
df = pd.get_dummies(df, columns=["Gender", "Department"])

# Hiển thị DataFrame sau khi chuyển đổi
display(df)

# Xử lý dữ liệu ngày tháng

## Chuyển cột ngày tháng thành đối tượng datetime
df["Date of Joining"] = pd.to_datetime(df["Date of Joining"])
```

```
## Trích xuất tháng và ngày trong tuần từ cột ngày tháng
df["month"] = df["Date of Joining"].dt.month
df["day_of_week"] = df["Date of Joining"].dt.day_name()

## Loại bỏ cột gốc "Date"
df = df.drop("Date of Joining", axis=1)

# Hiển thị DataFrame sau khi xử lý dữ liệu ngày tháng
display(df)

# Xử lý các giá trị ngoại lai, chuẩn hóa và tỉ lệ dữ liệu

df1 = df.drop(["First Name", "Last Name", "day_of_week"], axis=1)
array = df1.values

### Sử dụng RobustScaler() để loại bỏ các giá trị ngoại lai
scaler = preprocessing.RobustScaler()
robust_df = scaler.fit_transform(array)
robust_df = pd.DataFrame(robust_df)

### Chuẩn hóa theo Z-score
scaler = preprocessing.StandardScaler()
standard = scaler.fit_transform(array)
standard_df = pd.DataFrame(standard, index=df.index)

print("Dữ liệu chuẩn hóa theo Z-score: \n", standard_df)

### Tỉ lệ dữ liệu theo phương pháp Minmax
scaler = preprocessing.MinMaxScaler()
minmax = scaler.fit_transform(array)
minmax_df = pd.DataFrame(minmax, index=df.index)

print("Dữ liệu tỉ lệ: \n", minmax_df)

# 10 phạm vi equi-width với cột đầu tiên của standard_df
df2 = standard_df.copy()
df2["equi-width_column0"] = pd.cut(x=df2[0], bins=10)
print("Phân loại cột đầu tiên thành 10 phạm vi equi-width: \n", df2)

# 10 phạm vi equi-depth với cột đầu tiên của standard_df
df3 = standard_df.copy()
df3["equi-depth_column0"] = pd.qcut(x=df3[0], q=10)
print("Phân loại cột đầu tiên thành 10 phạm vi equi-depth: \n", df3)
```

Đoạn mã trên thực hiện một loạt các thao tác xử lý dữ liệu, bao gồm điền giá trị trống, loại bỏ giá trị trùng lặp, chuyển đổi giá trị hạng mục thành biến giả (dummy variables), xử lý dữ liệu ngày tháng, xử lý các giá trị ngoại lai, chuẩn hóa và tỉ lệ dữ liệu, và phân loại dữ liệu.

Làm tiếp những chỗ chưa hoàn chỉnh ở mục 2

```
import pandas as pd
import numpy as np
from sklearn.impute import SimpleImputer
```

```
from sklearn.preprocessing import StandardScaler
from sklearn.preprocessing import KBinsDiscretizer

# Đọc dữ liệu từ tệp arrhythmia.data
df = pd.read_csv("arrhythmia\\arrhythmia.data", delimiter=",",
header=None)

# Hiển thị DataFrame
display(df)

# Kiểm tra giá trị null
print(df.isnull().sum())

# Kiểm tra giá trị trùng lặp cho từng cột
for column in df.columns:
    print(df[column].duplicated().sum())

# Đếm số giá trị NaN
print(df.isnull().sum().sum())

# Chuyển đổi từ kiểu dữ liệu chuỗi sang kiểu dữ liệu số thực
df = df.astype(float)

# Điền giá trị NaN bằng giá trị trung bình của từng cột
df.fillna(df.mean(), inplace=True)

print(df)

# Xử lý các giá trị ngoại lai, chuẩn hóa và tỉ lệ dữ liệu
array = df.values

### Sử dụng RobustScaler() để loại bỏ các giá trị ngoại lai
scaler = preprocessing.RobustScaler()
robust_df = scaler.fit_transform(array)
robust_df = pd.DataFrame(robust_df)

### Chuẩn hóa theo Z-score
scaler = preprocessing.StandardScaler()
standard = scaler.fit_transform(array)
standard_df = pd.DataFrame(standard, index=df.index)
print("Dữ liệu chuẩn hóa theo Z-score: \n", standard_df)

### Tỉ lệ dữ liệu theo phương pháp Minmax
scaler = preprocessing.MinMaxScaler()
minmax = scaler.fit_transform(array)
minmax_df = pd.DataFrame(minmax, index=df.index)
print("Dữ liệu tỉ lệ: \n", minmax_df)

# Tạo một DataFrame với 10 phạm vi equi-width cho mỗi cột của
standard_df
df_ranges1 = standard_df.copy()
for column in standard_df.columns:
    df_ranges1[column] = pd.cut(df_ranges1[column], bins=10)

display(df_ranges1)
```

Đoạn mã trên thực hiện một loạt các thao tác xử lý dữ liệu, bao gồm kiểm tra giá trị null, kiểm tra giá trị trùng lặp, chuyển đổi kiểu dữ liệu, xử lý giá trị null bằng cách điền giá trị trung bình, xử lý các giá trị ngoại lai, chuẩn hóa và tỉ lệ dữ liệu, và tạo DataFrame mới với các phạm vi equi-width cho dữ liệu.

Sử dụng mục 1 để làm sạch dữ liệu và tiền xử lý dữ liệu, và sử dụng PCA trong thư viện sklearn để làm mục 3.

```
# Load data from data.csv to DataFrame Pandas
import pandas as pd
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
import numpy as np
import matplotlib.pyplot as plt

# Đọc dữ liệu từ tệp clean1.data
df = pd.read_csv("musk+version+1\\clean1.data\\clean1.data",
delimiter=",", header=None)

# Hiển thị DataFrame
display(df)

# Tạo một bản sao của DataFrame cho quá trình tiền xử lý
preprocessing_df = df.copy()

# Loại bỏ cột 0 và cột 1
preprocessing_df.drop(columns=[0], inplace=True)
preprocessing_df.drop(columns=[1], inplace=True)

# Hiển thị DataFrame sau khi loại bỏ cột
preprocessing_df

# Kiểm tra giá trị null
print(preprocessing_df.isnull().sum())

# Kiểm tra giá trị trùng lặp
print(preprocessing_df.duplicated().sum())

# Chuyển DataFrame thành mảng numpy
preprocessing_array = preprocessing_df.values

print(preprocessing_array)

### Chuẩn hóa theo Z-score
scaler = preprocessing.StandardScaler()
standard = scaler.fit_transform(preprocessing_array)
standard_df = pd.DataFrame(standard,index = df.index)
print("Dữ liệu chuẩn hóa theo Z-score: \n", standard_df)

# Tính ma trận hiệp phương sai
cov_matrix = standard_df.cov()
```

```
# Tính giá trị riêng và vector riêng
eigenvalues, eigenvectors = np.linalg.eig(cov_matrix)

# Vẽ biểu đồ Scree
plt.plot(range(1,168),eigenvalues)
plt.title("Scree plot")
plt.show()

# Sử dụng PCA để giảm chiều dữ liệu xuống còn 50 thành phần
pca = PCA(n_components=50)
tranformed_data = pca.fit_transform(standard_df)

# Tạo DataFrame mới từ dữ liệu sau khi giảm chiều
df_PCA = pd.DataFrame(tranformed_data)

# Hiển thị DataFrame mới
display(df_PCA)
```

Đoạn mã trên thực hiện các thao tác như sau:

1. Đọc dữ liệu từ tệp `clean1.data` và hiển thị DataFrame.
2. Tiền xử lý dữ liệu bằng cách loại bỏ các cột không cần thiết.
3. Kiểm tra giá trị null và giá trị trùng lặp.
4. Chuẩn hóa dữ liệu theo Z-score.
5. Tính toán ma trận hiệp phương sai và các giá trị riêng, sau đó vẽ biểu đồ Scree.
6. Sử dụng PCA để giảm chiều dữ liệu xuống còn 50 thành phần và hiển thị DataFrame mới sau khi giảm chiều.