KHAI THÁC DỮ LIỆU

Nguyễn Xuân Việt Đức - 22280012

Bài tập lý thuyết - Lần 1

Cài đặt và thực thi mục 1 trên máy tính

1. Làm sạch và tiền xử lý dữ liệu:

- Quá trình bắt đầu từ việc đọc file csv, set index, kiểm tra dataframe có giá trị null hay không
- Nếu có giá trị null bên trong thì thực hiện fill bằng giá trị mean của cột đó
- Kiểm tra trùng lặp và xóa trùng lặp
- Xử lý thuộc tính categorical
- Feature engineering thuộc tính Date
- Sử dụng các phương pháp để chuẩn hóa dữ liệu bao gồm: $RobustScaler()$, $StandardScaler()$, $MinMaxScaler()$.
- Thực hiện rời rạc hóa cột numeric theo chiều rộng và chiều sâu.

2. Bài tập vận dụng:

Phân Tích Bộ Dữ Liệu Rối Loạn Nhịp Tim: Bộ dữ liệu này chứa các phép đo điện tâm đồ (ECG) được sử dụng để xác định và phân loại các loại rối loạn nhịp tim khác nhau. Nó bao gồm 452 hồ sơ bệnh nhân với 279 thuộc tính khác nhau được lấy từ các kết quả đo ECG và thông tin cơ bản về bệnh nhân.

Tổng Quan Về Bộ Dữ Liệu:

- Số lượng mẫu: 452
- Số lượng thuộc tính: 279
- Giá trị bị thiếu: Nhiều giá trị bị thiếu, được đánh dấu bằng '?'
- Phân bố lớp: Mất cân bằng nghiêm trọng, với hơn một nửa số mẫu thuộc nhóm "Bình thường"

Lưu Ý Quan Trọng:

- Bộ dữ liệu này nhằm phân biệt giữa sự có mặt và không có mặt của rối loạn nhịp tim, đồng thời phân loại nó vào một trong 16 nhóm.
- Tiêu chuẩn vàng để phân loại là chẩn đoán của bác sĩ tim mạch, mà bài toán học máy này cố gắng mô phỏng.
- Một số lớp (11, 12, 13) không có dữ liệu, do đó thực tế đây là bài toán phân loại 13 lớp.

Làm tiếp những chổ chưa hoàn chỉnh ở mục 2:

- Đọc dữ liệu arrhythmia.data bằng pandas
- Kiểm tra tính trùng lặp và thực hiện xóa bỏ
- Đếm số lượng question mark bên trong dataframe thì thấy rằng những kí tự này chỉ xuất hiện ở các cột thứ 10 đến cột thứ 14 trong dataframe, cột thứ 13 có số lượng null khá lớn khoảng hơn 80 phần trăm còn các cột khác chiếm rất nhỏ, nên em quyết định xóa bỏ cột 13 và fill các cột khác bằng giá trị mean.
- Kiểm tra thuộc tính categorical thì trong dataset này không có thuộc tính nào categorical.
- Thực hiện chuẩn hóa tương tự bài 1:

```
Data shape: (452, 279)
Features shape: (452, 278)
Target shape: (452,)
Number of numeric features: 278
Number of categorical features: 0

NaN values after StandardScaler normalization: 0
NaN values after MinMaxScaler normalization: 0
NaN values after RobustScaler normalization: 0

Original Data (first 5 rows, first 5 columns):
    0  1    2   3    4
0  75  0  190  80   91
1  56  1  165  64   81
2  54  0  172  95  138
3  55  0  175  94  100
4  75  0  190  80   88
```

```
StandardScaler Normalized (first 5 rows, first 5 columns):
          0         1         2         3         4
0  1.734439 -1.107520  0.641327  0.713814  0.135505
1  0.579312  0.902918 -0.031998 -0.251644 -0.516072
2  0.457720 -1.107520  0.156533  1.618932  3.197915
3  0.518516 -1.107520  0.237332  1.558590  0.721924
4  1.734439 -1.107520  0.641327  0.713814 -0.059968

MinMaxScaler Normalized (first 5 rows, first 5 columns):
          0    1         2         3         4
0  0.903614  0.0  0.125926  0.435294  0.270677
1  0.674699  1.0  0.088889  0.341176  0.195489
2  0.650602  0.0  0.099259  0.523529  0.624060
3  0.662651  0.0  0.103704  0.517647  0.338346
4  0.903614  0.0  0.125926  0.435294  0.248120

RobustScaler Normalized (first 5 rows, first 5 columns):
          0    1    2     3         4
0  1.272727 -1.0  2.6  0.60  0.357143
1  0.409091  0.0  0.1 -0.20 -0.357143
2  0.318182 -1.0  0.8  1.35  3.714286
3  0.363636 -1.0  1.1  1.30  1.000000
4  1.272727 -1.0  2.6  0.60  0.142857
```

- Thực hiện quá trình rời rạc hóa tương tự bài 1:

```
Discretization Summary:
Total columns processed: 278
Equi-width bins: All 278 columns successfully discretized with 10 bins
Equi-depth bins: 278 columns successfully discretized with 10 bins
Equi-depth bins: 0 columns could not use 10 bins (kept original values)

Original data (first 5 rows, first 5 numeric columns):
    0  1    2   3    4
0  75  0  190  80   91
1  56  1  165  64   81
2  54  0  172  95  138
3  55  0  175  94  100
4  75  0  190  80   88
```

```
Equi-width discretization (first 5 rows, first 5 discretized columns):
   equi-width_0   equi-width_1      equi-width_2   equi-width_3  \
0  (74.7, 83.0]  (-0.001, 0.1]    (172.5, 240.0]   (74.0, 91.0]
1  (49.8, 58.1]     (0.9, 1.0]  (104.325, 172.5]   (57.0, 74.0]
2  (49.8, 58.1]  (-0.001, 0.1]  (104.325, 172.5]  (91.0, 108.0]
3  (49.8, 58.1]  (-0.001, 0.1]    (172.5, 240.0]  (91.0, 108.0]
4  (74.7, 83.0]  (-0.001, 0.1]    (172.5, 240.0]   (74.0, 91.0]

     equi-width_4
0    (81.6, 94.9]
1    (68.3, 81.6]
2  (134.8, 148.1]
3    (94.9, 108.2]
4    (81.6, 94.9]

Equi-depth discretization (first 5 rows, first 5 discretized columns):
   equi-depth_0   equi-depth_1    equi-depth_2   equi-depth_3    equi-depth_4
0  (67.0, 83.0]  (-0.001, 1.0]  (175.0, 780.0]   (75.0, 80.0]    (90.0, 92.0]
1  (51.0, 56.0]  (-0.001, 1.0]  (164.0, 165.0]   (60.0, 65.0]    (78.2, 81.0]
2  (51.0, 56.0]  (-0.001, 1.0]  (171.0, 175.0]  (86.0, 176.0]  (102.9, 188.0]
3  (51.0, 56.0]  (-0.001, 1.0]  (171.0, 175.0]  (86.0, 176.0]   (96.0, 102.9]
4  (67.0, 83.0]  (-0.001, 1.0]  (175.0, 780.0]   (75.0, 80.0]    (86.0, 90.0]
```

3. PCA

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.

Phân Tích Bộ Dữ Liệu MUSK "Clean2"

Mô Tả Các Thuộc Tính:

- Bộ dữ liệu này chứa thông tin về cấu trúc phân tử với 166 thuộc tính mô tả các đặc điểm khác nhau của hình dạng phân tử (cấu dạng).

Bộ dữ liệu này là một bài toán "nhiều thể hiện" (multiple instance problem), nghĩa là:

- Mỗi phân tử có thể tồn tại ở nhiều cấu dạng (hình dạng khác nhau) do liên kết quay được
- Bộ dữ liệu chứa 6.598 cấu dạng nhưng chỉ thuộc về 102 phân tử
- Một phân tử được phân loại là musk nếu BẤT KỲ cấu dạng nào của nó có mùi xạ hương

- Một phân tử được phân loại là non-musk nếu KHÔNG CÓ cấu dạng nào của nó có mùi xạ hương

Phân Bố Lớp:

- Phân tử có mùi xạ hương: 39 (lớp 1)
- Phân tử không có mùi xạ hương: 63 (lớp 0)

Các bước thực hiện:

- Thực hiện kiểm tra thống kê mô tả để biết về dữ liệu
- Kiểm tra trùng lặp thì dữ liệu có 17 dòng bị trùng, thực hiện xóa các dòng trùng lặp.
- kiểm tra null thì không có
- Thực hiện chuẩn hóa dữ liệu bằng $StandardScaler()$:

```
X_std = (X - X.mean(axis = 0))/X.std(axis = 0, ddof = 1)
X_std.head()
```

	f1	f2	f3	f4	f5	f6	f7	f8	f9	f10	...	f157	f158	f159	f16
0	-0.240992	0.121391	0.191289	-0.851808	-0.206286	0.380478	0.452903	-1.769797	0.720549	0.687638	...	-0.394881	-1.144042	1.000815	0.27860
1	-0.335589	-0.759763	-1.059200	0.279005	-0.206286	-0.301199	0.617523	-1.880732	0.434665	-0.767787	...	-0.287492	1.165690	0.578577	1.13040
2	-0.240992	-0.825850	-1.059200	0.353564	-0.206286	0.677936	0.617523	-1.847452	0.434665	0.313386	...	-0.323289	0.469988	-0.609946	1.20259
3	-0.335589	-0.759763	-1.059200	0.279005	-0.206286	-0.313593	0.617523	-1.869639	0.434665	-0.753926	...	-0.299425	1.156414	0.562938	1.13040
4	-0.335589	-0.759763	-1.059200	0.279005	-0.206286	-0.313593	0.617523	-1.869639	0.434665	-0.753926	...	-0.299425	1.156414	0.562938	1.13040

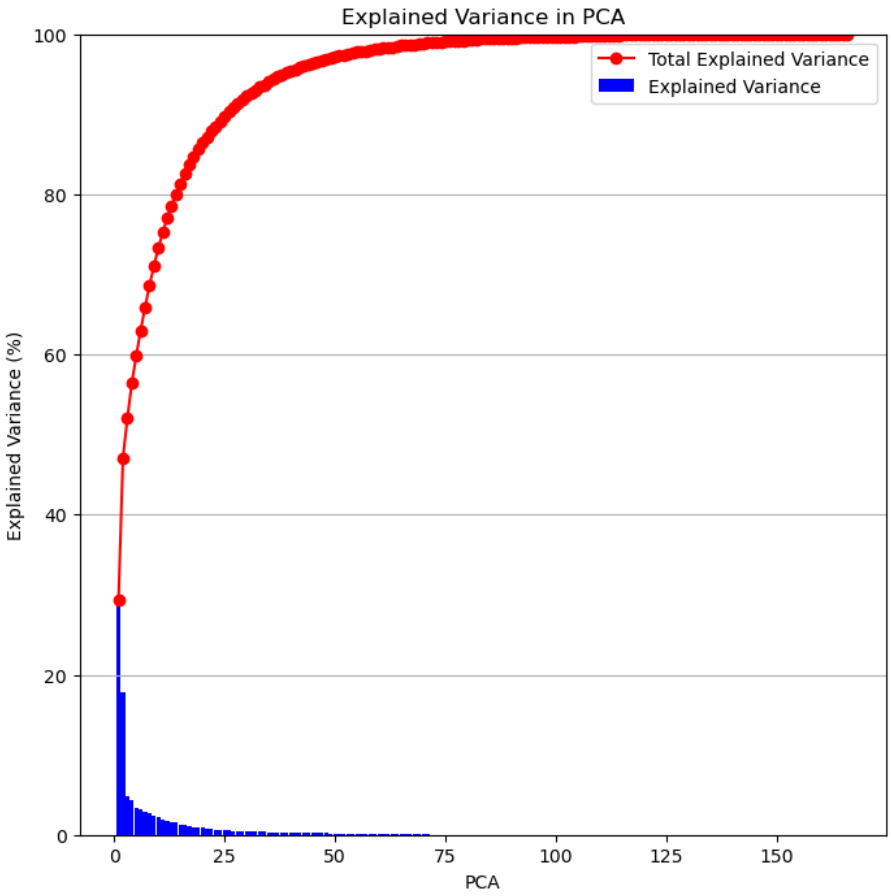
.

- Bắt đầu quá trình PCA:

```
# Fitting Our Data to a PCA Model
from sklearn.decomposition import PCA
pca = PCA()
pca.fit(X_std)
```

```
PCA
PCA()
```

- Phương sai giải thích trong PCA



Biểu đồ trên thể hiện phương sai giải thích của PCA:

- Đường màu đỏ (Total Explained Variance) cho thấy phương sai tích lũy tăng dần và đạt gần 100 phần trăm sau một số lượng thành phần chính nhất định. Điều này gợi ý rằng chỉ một số ít thành phần đầu tiên có thể giữ lại phần lớn thông tin của dữ liệu.
- Cột màu xanh (Explained Variance) đại diện cho lượng phương sai mà mỗi thành phần chính đóng góp. Ta thấy rằng các thành phần đầu tiên có phương sai cao, nhưng giảm dần về sau.
- Nhận xét: Để giảm chiều dữ liệu mà vẫn giữ lại hầu hết thông tin, ta có thể chọn số thành phần chính tại điểm "elbow" (gấp khúc) trên đường đỏ, nơi phương sai tích lũy bắt đầu bão hòa.