Q1 实验报告

1854116 朱明志

目录

1 数据集描述

对住院患者进行高血糖的状况研究具有重要意义，因为它有助于控制与健康问题有关的发病率和死亡率。在这个数据集中，我将分析与再入院相关的因素以及与糖尿病患者相关的其他属性。本数据集代表了美国 130 家医院 10 年（1999-2008 年）的临床护理数据。它有超过 50 个代表病人和医院状态的特征。为满足以下标准的情况提供信息：

- 这是住院病人的信息（入院）
- 这是一个糖尿病的遭遇，也就是说，在这个过程中，任何类型的糖尿病都被作为诊断输入系统。
- 住院时间至少 1 天，最多 14 天
- 在就诊中进行了实验室测试
- 在就诊中服用了药物
- 数据包含诸如患者编号、种族、性别、年龄、入院类型、住院时间、入院医师的医学专业、进行实验室检查的次数、HbA1c 测试结果、诊断、药物数量、糖尿病药物、门诊患者数量、住院患者、住院前一年的急诊就诊以及其他的许多属性。

在这个作业中，我的目标是对这个输入数据集进行预处理、分析、可视化和无监督学习。在细节上，我计划使用聚类分析，以便根据特征将患者分组。我首先从数据预处理开始，通过探索数据、管理缺失值、处理接近零的方差变量和转换变量。然后，我使用 k-means 算法和层次聚类进行聚类分析。

糖尿病数据集可通过以下链接在 UCI 机器学习库网站上获得：

https://archive.ics.uci.edu/ml/datasets/diabetes+130-us+hospitals+for+years+1999-2008

2 数据清洗与数据预处理

原始数据集包含 101766 个观测值和 50 个特征。在数据预处理部分中，我的目标是通过做一些初步的统计分析、变量转换、NA 值处理等来熟悉数据集并做好准备。

2.1 缺失值处理

数据集有许多属性，其中有“？”作为数据属性的值。显然，这是一个缺失值。因此用 Nan 替换“？”，以便我更好地处理它们。经检查，在 50 个特征中，我只发现了 6 个属性中含有缺失值。下表列出了这 6 项属性中每项属性丢失数据的数量。我可以看到，对于一些变量，如权重，缺失值的数量是非常高的。

属性名	缺失值数量/数据总条数
race	2273/101766
weight	98569/101766
payer_code	40256/101766
medical_specialty	49949/101766
diag_1	21/101766
diag_2	358/101766
diag_3	1423/101766

为了处理这些缺失值，我考虑设置了一个阈值 40000，如果一个属性的缺失值数量大于 40000，即约有 40%以上的数据缺失了，我就将这个属性列从数据集中删除。对于其他属性列的缺失值，如果一行中存在某个或多个属性中含有缺失值，我仅删除这一条数据而不是删除这个属性列。

缺失值的处理结果保存在 csvoutput 子文件夹下的 output1.csv 文件中。

2.2 近零方差属性

接近零方差（NZV）指的是一些属性在整个数据集中每一条数据几乎取唯一值的情况。这类属性不仅有时是非信息性的，还可能破坏一些我可能想要使用的数据挖掘方法。因此，删除这些无用的属性列是一个好想法，特别是在处理具有大量变量的数据集时。在本数据集中，有几个因子变量的方差接近于零。保留它们以后会产生相当数量的虚拟变量，并增加计算的复杂性和对资源的需求。因此，我删除了所有 NZV 属性。默认情况下，如果样本中唯一值的百分比小于 10%，那么 R 语言中的 caret 包中的 nearZeroVar()函数将变量视为接近零的方差。在 python 中，我使用 pandas 的 value_count()函数，如果一个属性列出现最多次数的取值数量大于 90000，即约占样本数量的 90%以上，我便选择删除这一列属性。通过在数据集上执行此操作，我删除了 22 个接近零方差的属性。下表展示是方差接近零的属性：

number_emergency	max_glu_serum	repaglinide
nateglinide	chlorpropamide	glimepiride
acetohexamide	glyburide	tolbutamide
pioglitazone	rosiglitazone	acarbose
miglitol	troglitazone	tolazamide
examide	citoglipton	glyburide-metformin
glipizide-metformin	glimepiride-pioglitazone	metformin-rosiglitazone
metformin-pioglitazone		

2.3 多次接触的患者

数据集包含多个具有相同患者编号（patient_nbr）的行。目前尚不清楚医生和病人之间的多次接触是否是独立的。患者的多次就诊可能有关联的风险，因此会引入偏见，即患者的某些遭遇可能是相关的。为了消除这种风险，我决定只保留一次在医院时间最长的接触，假设在医院的时间最长是再入院的特征，并且在训练数据中有足够的差异。完成后，我删除了 patient_nbr 属性和 encounter_id 属性。因此，数据集减少到 23 个属性和 68468 个观测值。

前三小节的处理结果保存在 csvoutput 子文件夹下的 output2.csv 文件中。

2.4 属性值转换

一些分类属性包含太多的类别，这是三个属性的情况：diag_1、diag_2 和 diag_3，每个属性都有大约 700 个级别。这将需要大约 700 个转换后的属性值，并且计算需要的管理成本很高。为了整合这三个变量的级别，我遵循了关于这个数据集的原始报告中表 2 中的规则：https://www.hindawi.com/journals/bmri/2014/781670/，然后将级别降低到 9 个类别。除了这些属性外，我还考虑了其他一些非数值属性。性别属性的水平从 3（男性、女性和未知）降低到 2（男性和女性，未知有且仅有一个已经被删除）。我将年龄段从 10 级合并到 5 级，入院来源变量从 25 级合并到 5 级（转诊、转院、出生、未知和其他）。由于我的目标是使用聚类分析，这些算法通常需要数值来计算距离等度量。为了允许在这些算法中使用分类属性，一种解决方法是使用一个热编码将这些分类属性转换成数值，这正是我在这个项目中采用的解决方案。

下表是一些分类属性值对应转化的数值属性值：（diag_x 属性对应关系见链接）

Caucasian	0
Hispanic	1
AfricanAmerican	-1
Other	2
Asian	-2
Female	-1
Male	1
[0-10)、[10-20)、[20-30)、[30-40)	-2
[50-60)	-1
[60-70)	0
[70-80)	1
[80-90)、[90-100)	2
>8	1

>7	2
None	0
Norm	-1
No	0
Up	-1
Down	-2
Steady	1
Ch	1
Yes	1
>30	1
<30	-1

下表是 admission_source_id 整合后归为 5 类的对应关系：

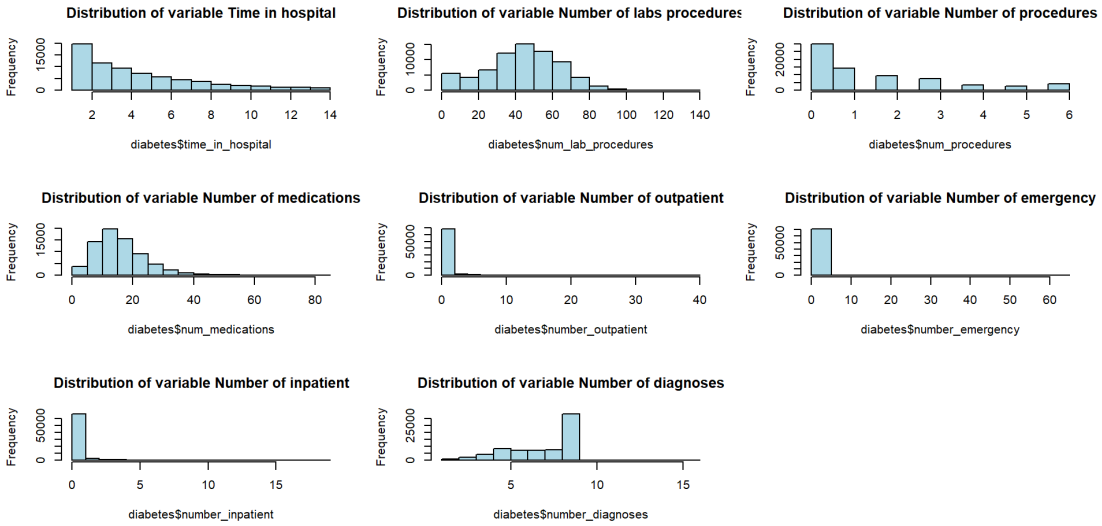
admission_source_id	description						
1	Physician Referral			1	referral	1 2 3	
2	Clinic Referral			2	transfer	4 5 6 10 18 19 22 25 26	
3	HMO Referral			3	birth	12 13 14 23 24	
4	Transfer from a hospital			4	unknown	9 15 17 20 21	
5	Transfer from a Skilled Nursing Facility (SN			5	other	7 8 11	
6	Transfer from another health care facility						
7	Emergency Room						
8	Court/Law Enforcement						
9	Not Available						
10	Transfer from critial access hospital						
11	Normal Delivery						
12	Premature Delivery						
13	Sick Baby						
14	Extramural Birth						
15	Not Available						
17	NULL						
18	Transfer From Another Home Health Agency						
19	Readmission to Same Home Health Agency						
20	Not Mapped						
21	Unknown/Invalid						
22	Transfer from hospital inpt/same fac reslt in a sep claim						
23	Born inside this hospital						
24	Born outside this hospital						
25	Transfer from Ambulatory Surgery Center						
26	Transfer from Hospice						

本小节的处理结果保存在 csvoutput 子文件夹下的 output3.csv 文件中。

2.5 离群值

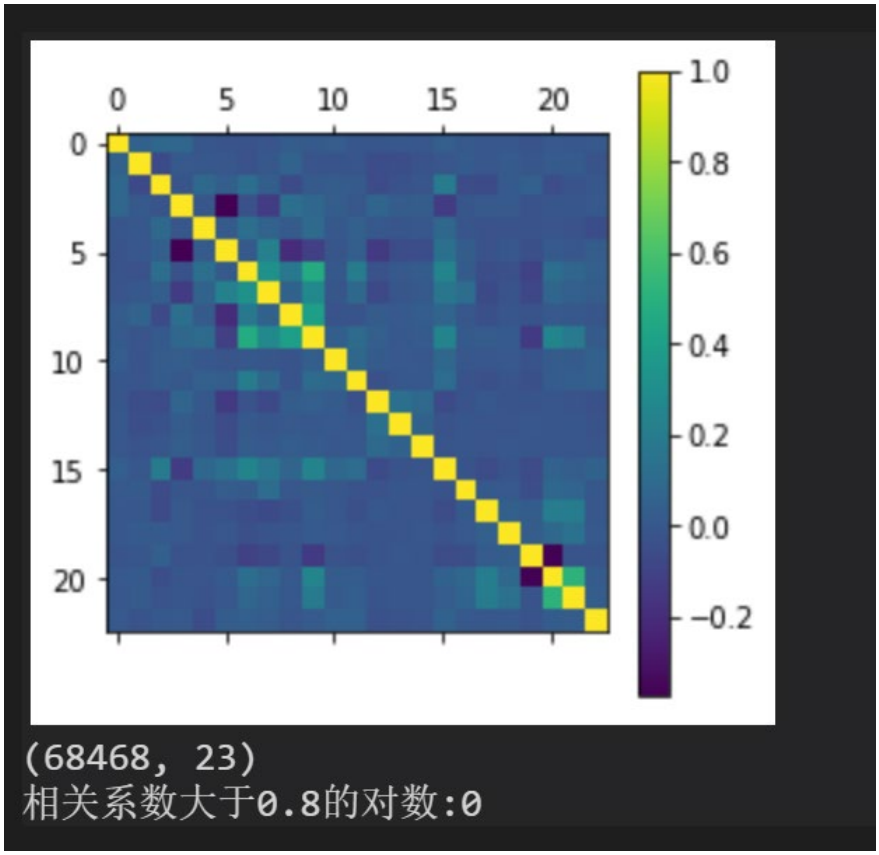
对于原本就是数值的属性而言（对于上一步转换的分类属性，在转换时就对离群值进行了合并处理），通过绘制数值属性来观察数据集，我发现具有极值的属性分布在少数观测值中。这些带有异常值的属性是 number_outpatient、number_inpatient 和 number_emergency。然而，这些变量的均值非常接近于零，去除一些异常值将导致所有汇总统计数据归零，这在后续处理中会造成一些

计算问题。因此，少数异常值保持原样。



2.6 相关系数

另一个需要探索的数值变量是它们之间的相关性。下面的热力图以绝对值显示了这些相关性，当颜色从深紫色变化到黄色时，相关性的绝对值从 0 增长到 1。统计了两两属性之间的相关系数大于 0.8 的组数为 0 组：一般来说，这些属性没有很强的相关性。



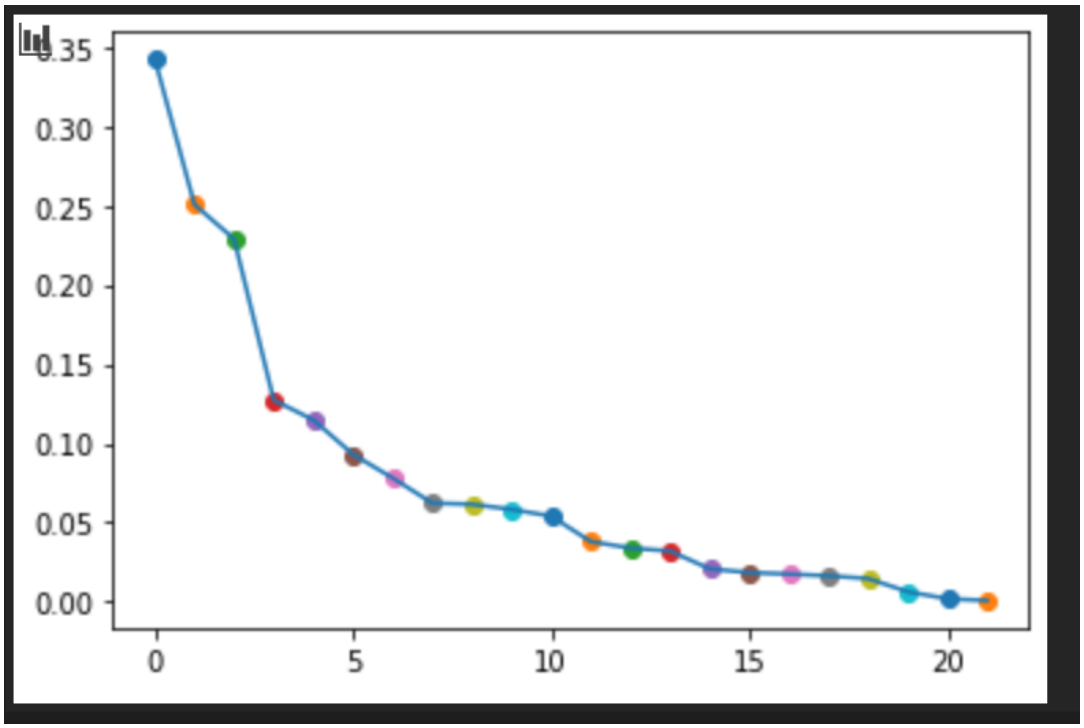
3 无监督学习

3.1 距离选择

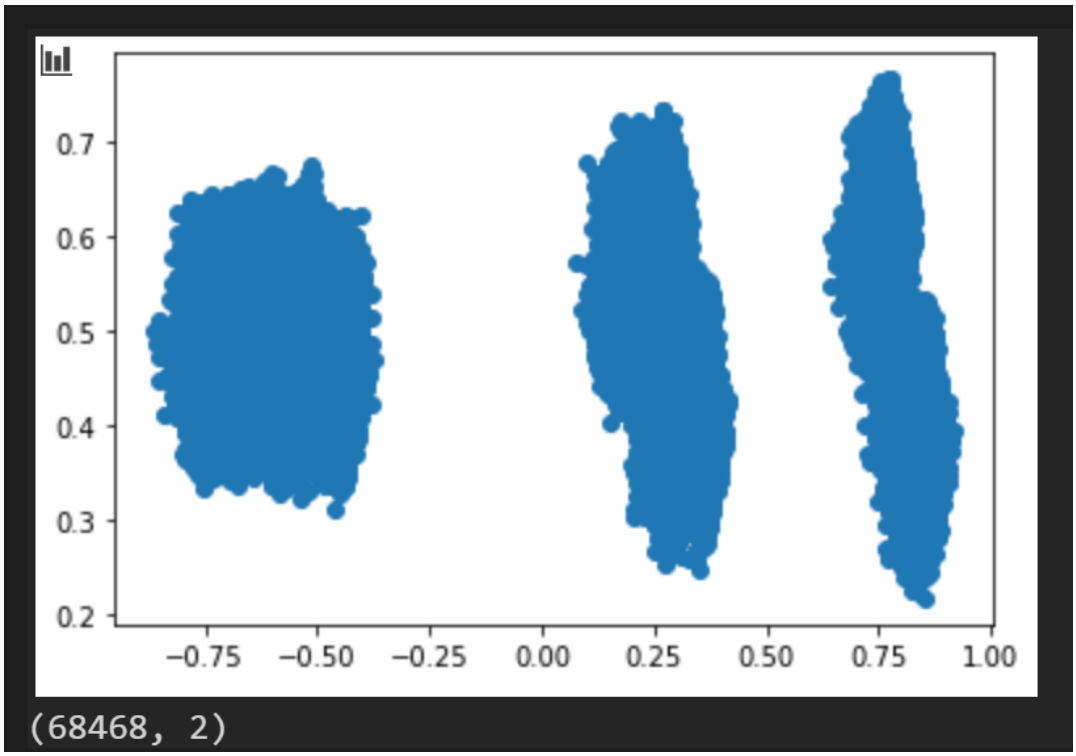
由上述的数据预处理可知，我将数据中所有分类值都替换为了数值。很显然的，使用欧式距离是一个简单好用的选择，但是有些属性本身就是数值，这样导致有些列数据分布跨度很大。因此，在欧式距离的基础上，我对数据使用正态标准化，对标准化后的数据进行欧式距离计算、PCA 降维，聚类等等操作。

3.2 PCA 分析

这一部分首先对数值属性进行主成分分析，以便将可能相关属性的观测值集转化为线性不相关属性的值集。这样做的目的是减少这些变量的数量，同时保持它们带来的信息量大致相同。经计算，第一分量累积方差的比例约为 20%。从下图中，我看到前两个主要成分累积方差比例约为 35%，因此可以对主组件做的一件事是，使用前两个组件来考虑到再入院属性来探索数据。



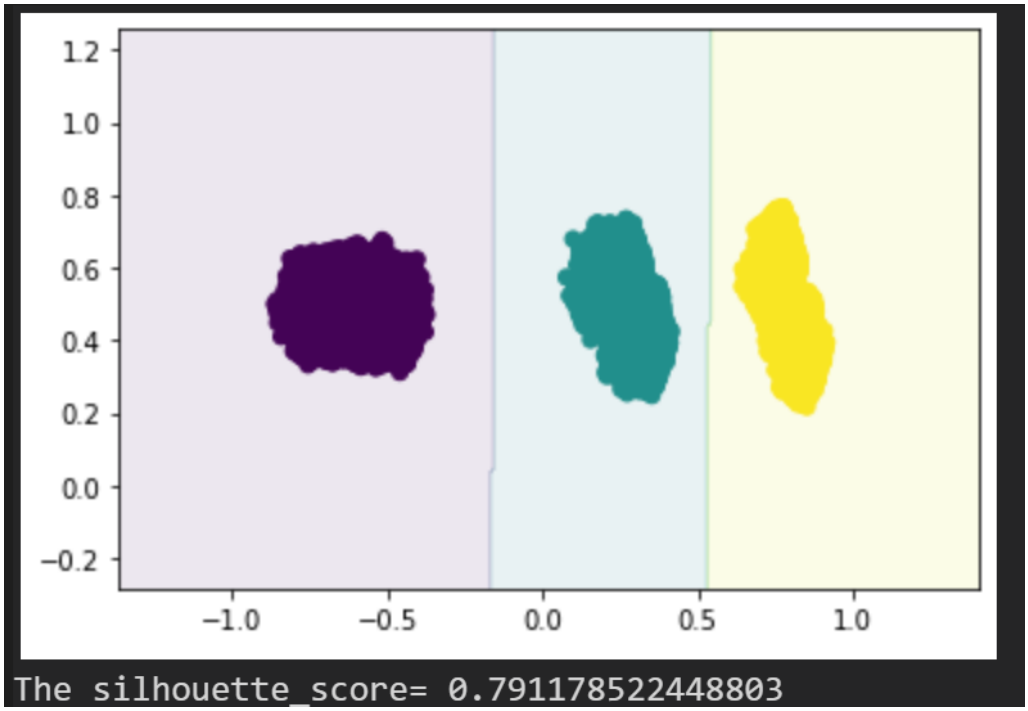
本段下面的图显示了数据集在前两个主成分上的投影。似乎有三个主要的星团：一个较大的在左边，两个形状相似的较小的在右边，下图中，这些簇在第一个主成分的方向上被分开，这意味着通过查看该成分的负载，可以确定当朝它的方向移动时哪些因素发生了变化。通过分析载荷，我可以洞察哪些分类特征在第一个主成分的方向上是变化的。



3.3 K-MEANS 聚类

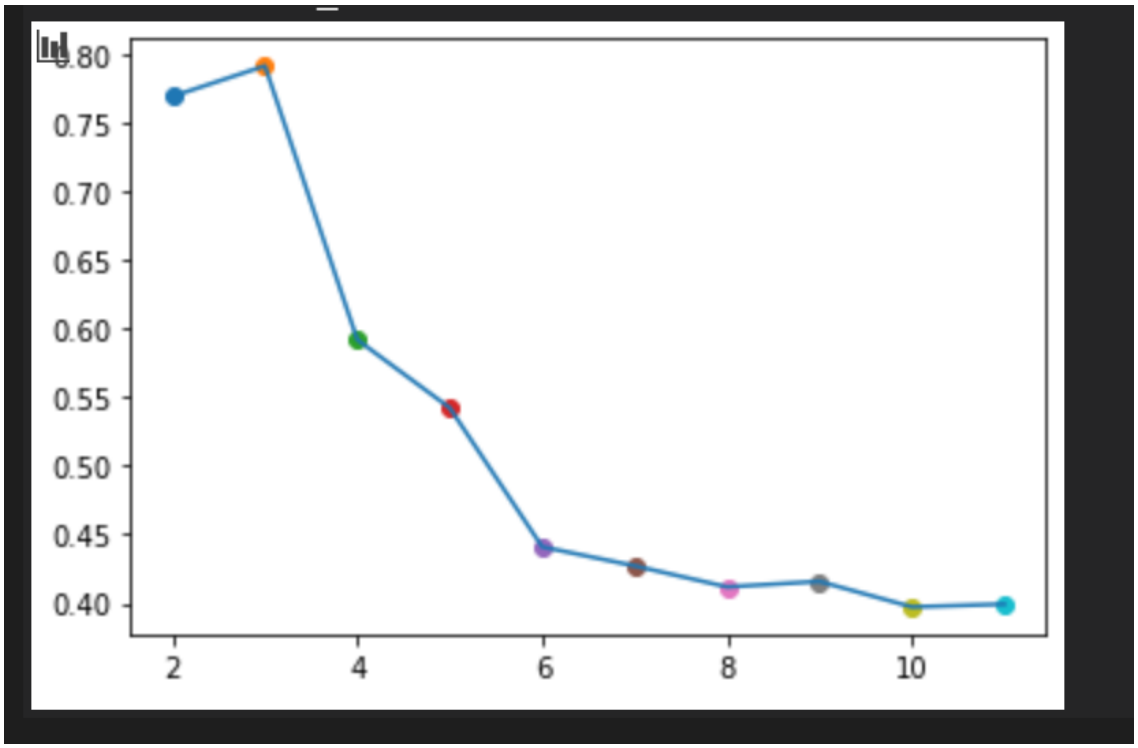
3.3.1 聚类结果

由 PCA 降维得到的结果，可以看出数据被明显被分为 3 团，因此 K 中心数显然是设置为 3。首先，我使用 k-means 聚类法将数据集分割成上面确定的聚类。通过在 python 中运行该算法，我有三个划分得很好的集群，现在可以查询它们相对于原始数据集的意义。（见第四部分）



3.3.2 聚类评价

可以看到，使用轮廓系数评价该聚类结果，SC 的值为 0.791，可以说是一个十分成功的聚类结果了。接下来对 K 中心取值从 2 到 11 遍历，纵轴为相应的轮廓系数，得到下图结果：



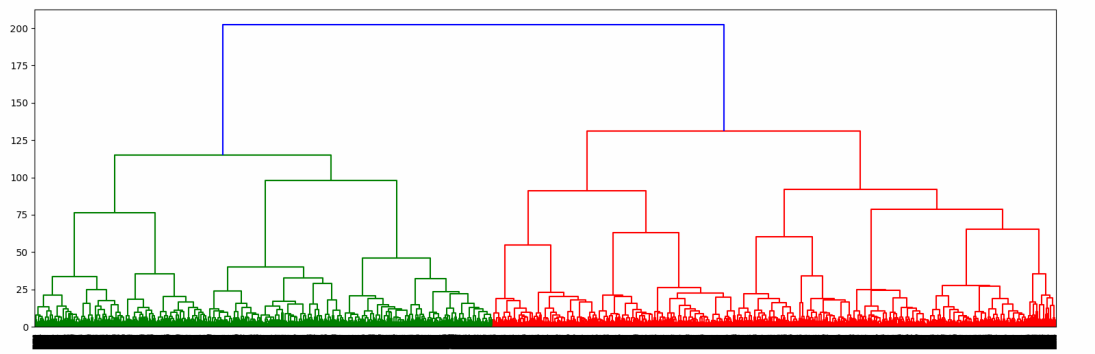
可见类的个数设置为 3 个时，K-means 算法有最优的表现。

3.4 层次聚类

3.4.1 树状图分析

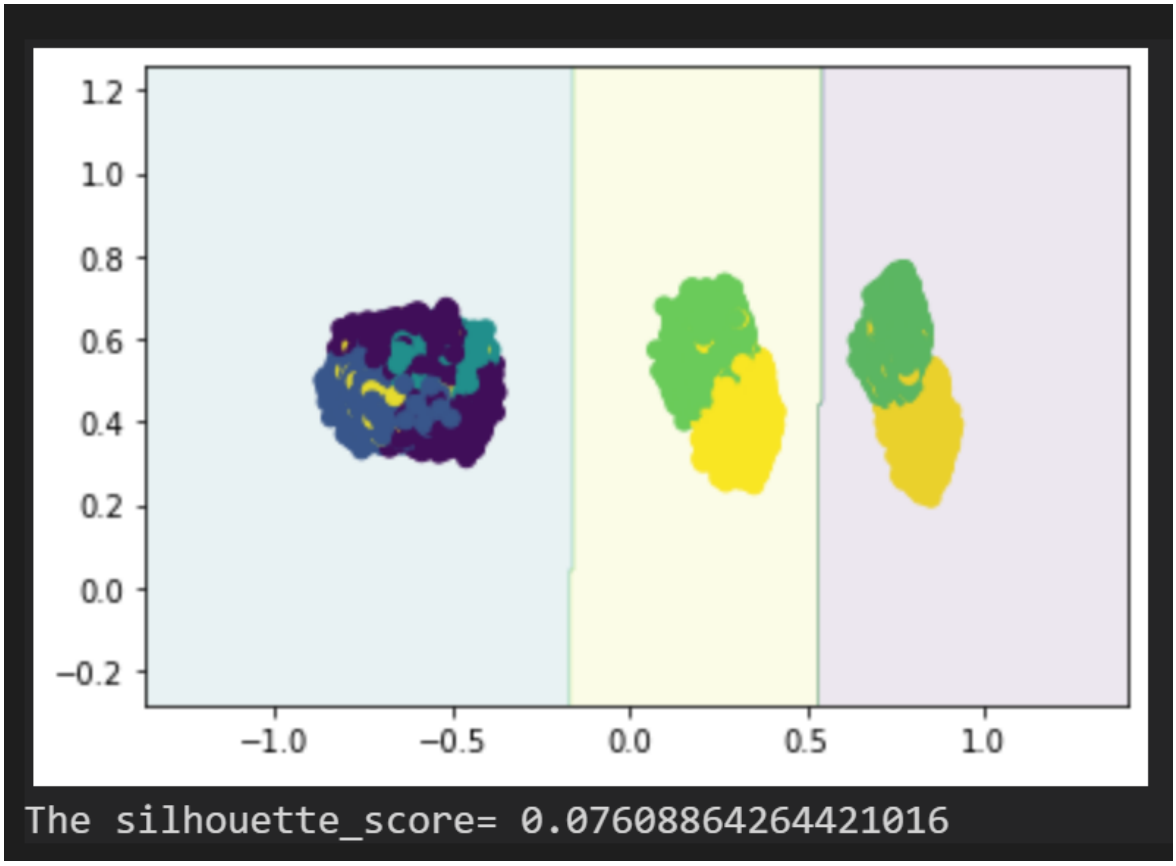
在这一部分中，我通过使用这种时间层次聚类来处理数据集，这种聚类与 K-means 一样，将具有相似特征的数据点分组在一起，我使用的仍然是“euclidean”方法，即欧氏距离。但是与 K-means 不同的是我没有对原数据进行 PCA 降维而是直接对 23 个属性进行层次聚类。我用 python 执行代码来计算树状图，得到下图。

因为数据太多，原图见 picture 子文件夹下 hier.jpg 文件

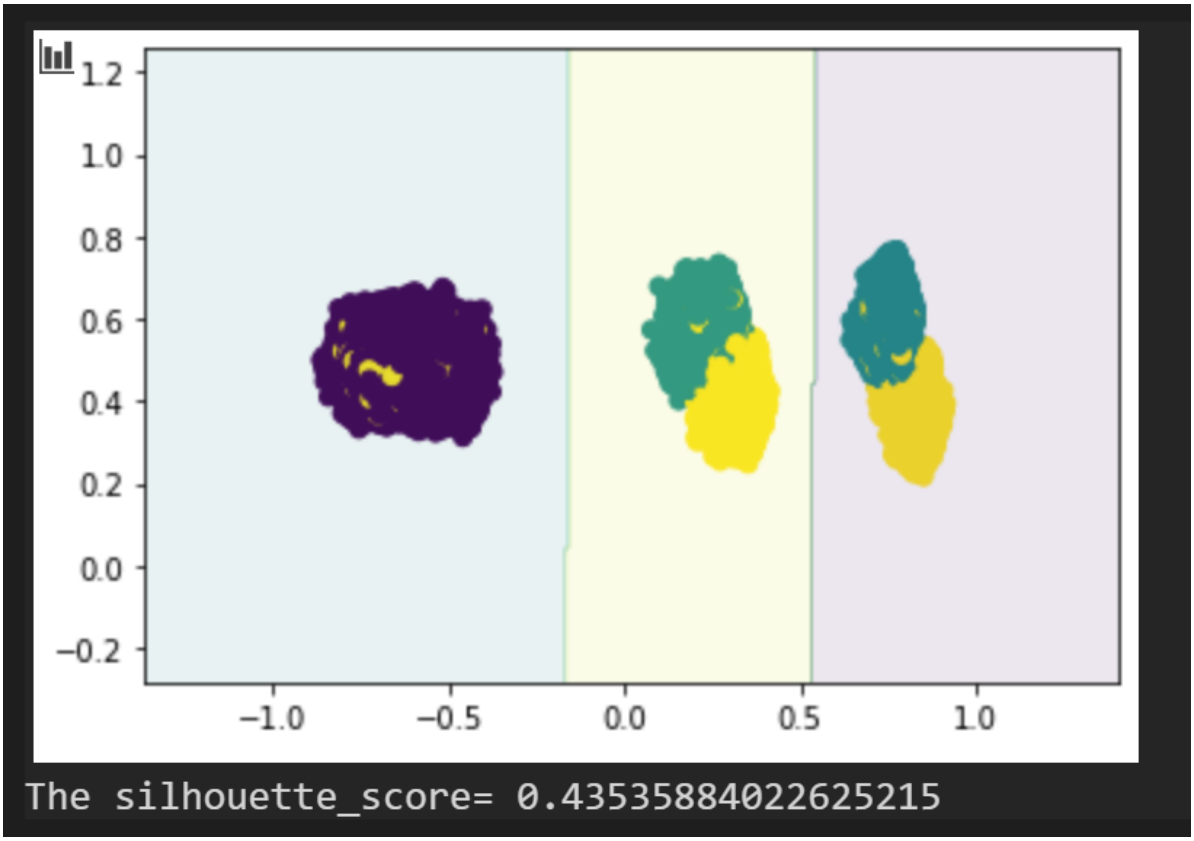


3.4.2 距离阈值切割分类

在构建树状图时，我决定在距离 95 处切割这棵树，得到 5 个聚类。下面，我根据前两个主成分绘制数据集，并根据它们所属的簇对点进行着色。我看到，与 K-均值聚类的结果不同，结果表明 5 个聚类在前两个主成分上没有很好的分离。我只看到浅绿色组的患者主要是分布于第二主成分的上侧。可以说，这类患者经常更换药物，通常被诊断为呼吸系统和糖尿病问题。



由 K-means 部分的结论可知，3 聚类在前两个主成分上有最佳的聚类区分表现。但是层次聚类与 K-means 不同之处在于不能直接设置中心点的个数。经过实验，我选择在距离 130 时重新切割聚类树，得到 3 个聚类。下面，我再次根据前两个主成分绘制数据集，并根据它们所属的簇对点进行着色。



可见对于层次聚类，距离阈值设置为 130 是一个合理的数据，即类的个数设置为 3 个类，但是这个聚类结果并没有 K-means 好。

4 实验结果讨论

这个作业是一个很好的学习机会。对于清理过程，我必须制定一个策略，使用 Python 清理和探索这个包含那么多变量的大数据集。对于建模部分，因为这些方法的计算成本非常高，使用 Python 来更好地利用内存也是一个好选择。两个聚类得到的聚类标签保存在 result1.csv 文件中

4.1 聚类算法开销比较

4.1.1 时间维度

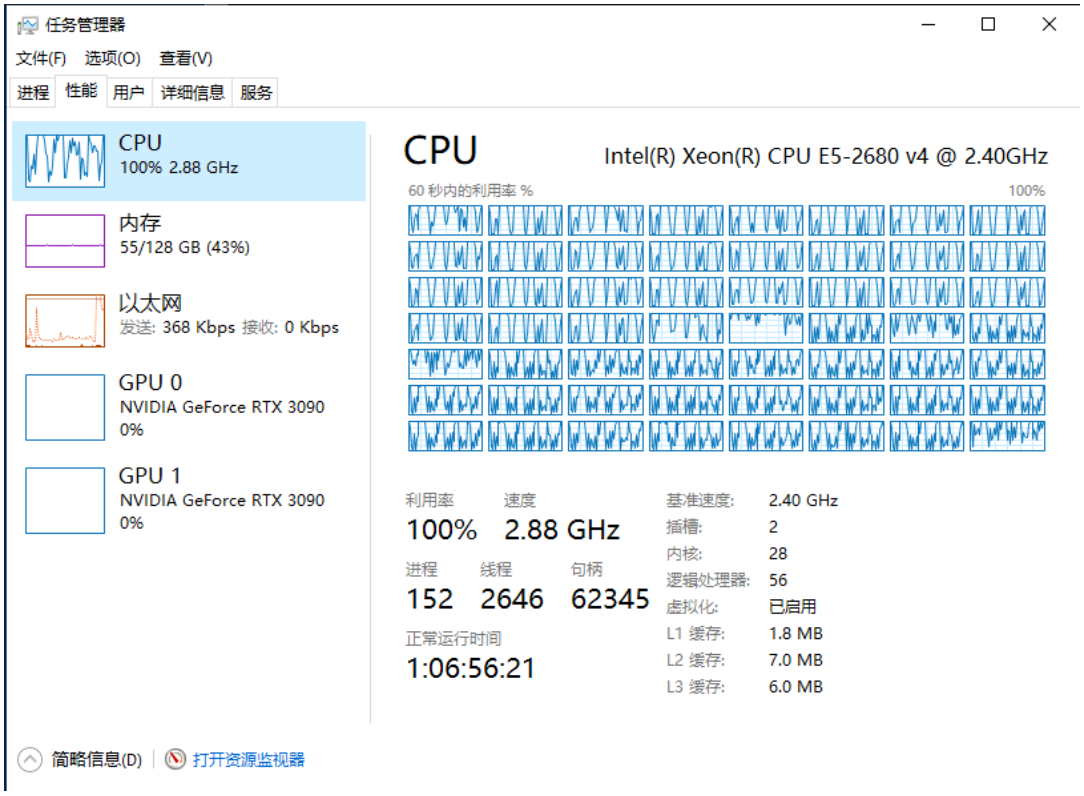
很明显的对比是，K-means 在时间复杂度上要优秀很多，即使不是 PCA 降维的原始 23 维数据，K-means 在我自己的笔记本电脑上跑一个聚类结果，也不会超过 5 分钟。标签的生成十分迅速，甚至可以支持我跑一个遍历 2-11 个 k 中心的循环，生成在不同数量 k 中心下的轮廓系数从而评价聚类结果。

与之对比的是，层次聚类的时间开销就大的多，考虑到层次聚类是一个网络的拓扑生成，这样的结果也不意外的。我使用了一个服务器来跑层次聚类，在有 28 核 56 线程 cpu 的电脑上，一轮层次聚类也要 20min，我也无法对距离进行遍

历找一个最佳的距离阈值，只能依赖 K-means 的结果实验出一个相对最佳的距离阈值。

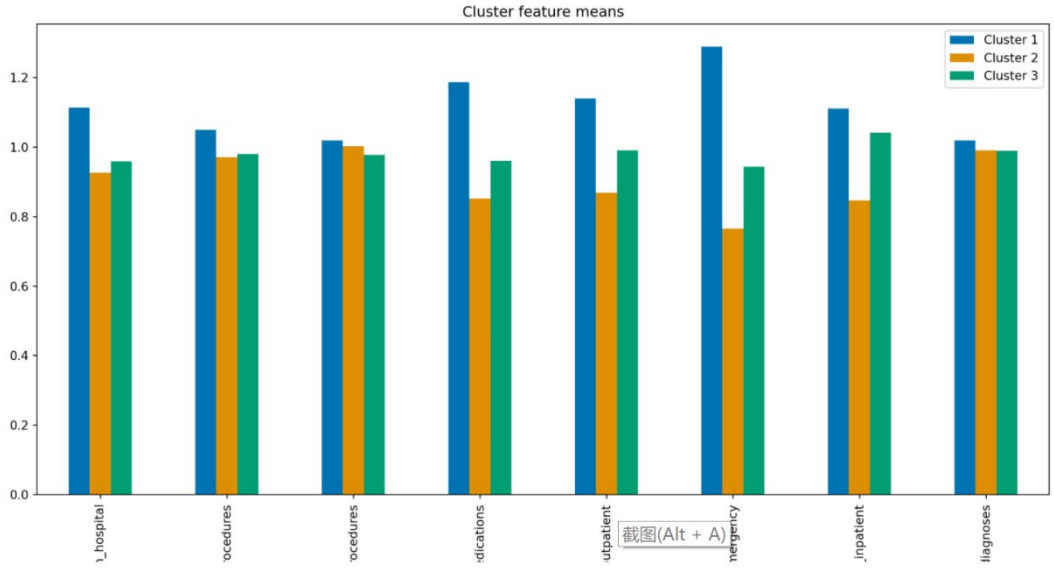
4.1.2 空间维度

空间的开销对比是更加明显的，K-means 对内存的消耗是几乎感受不到的，与之对比，层次聚类需要生成距离矩阵，68000*68000*float64，一个 64 位浮点数占用 8 个字节的空间，内存开销惊人。我自己 16g 内存的电脑根本跑不起来，看到有些同学采用采样的办法，正好我有一个大内存的服务器，就放在服务器上跑了跑，内存开销如下：



4.2 聚类算法结果分析

为了理解这些聚类，我将观测值的聚类标签作为列添加到数据框中，以便基于每个聚类分析变量结果的结果。我从数值变量开始，计算它们在每个簇中的平均值。我对这些特征进行了标准化，以便以类似的方式比较不同量级的变量。我在下图中展示了我的发现。

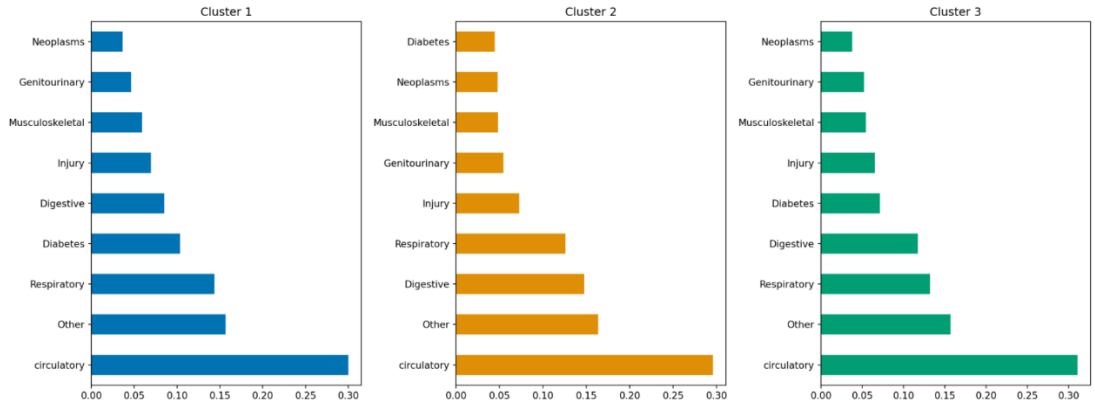


从上图中，可以得出一些观察结果：

- 第一组患者住院时间延长$\frac{1}{3}$到$\frac{1}{2}$天
- 第一组患者的实验室检查程序比第二组或第三组患者多 5%，平均多使用 15%到 25%的药物。
- 第一组中的患者有更多遭遇（住院、急诊和门诊）的记录。

在主成分部分，当我用分类变量来表示第一主成分中负荷的意义时，可以说较低的分数表明患者没有真正改变药物或没有使用任何糖尿病药物；相反，较高的分数表明患者改变了药物，至少使用了一种药物，倾向于使用更多的药物，或正在使用胰岛素/二甲双胍。因此，可以说第一组的患者倾向于使用药物，第二组的患者倾向于不使用药物，第三组的患者倾向于介于两者之间。

数据集中一个有趣的变量是诊断类型，我列出了每一个聚类中各个诊断类型出现的频率，它们的分布如下：



可以注意到：

- 所有集群的循环系统诊断都排在首位
- 服用较少/无糖尿病药物的患者因消化、呼吸和损伤问题再次入院的频率更高。服用多种糖尿病药物的患者因呼吸系统和糖尿病问题入院的频率更高。

诊断结果的分布表明，服用与糖尿病相关的药物较多的患者比服用较少或根本不服用的患者的糖尿病疾病更为突出。对这些患者来说，消化问题更为常见。

通过选择 K=3，使用 K-means 聚类，我已经看到数据聚类成三个不同的组，这可以很好地解释特征值的差异。其中一组代表服用大量药物的患者，这些患者通常被诊断为糖尿病和呼吸系统疾病。一组没有改变或不服用很多药物，这些患者通常被诊断为消化问题。而最后一类则介于前两者之间，具有一定的特殊性。

我还通过在生成 3 个聚类的层次上切割树来使用层次聚类。我发现这 3 个聚类在前两个主成分上并没有很好的分离，而且 K-means 聚类比层次聚类有更好的洞察力。我的解决方案的一个限制是，我只使用前两个主成分作为聚类的平均值，并解释聚类之间的差异。进一步的研究可以通过增加第三主成分和第四主成分来解决这个问题。