一、课题综述

1.1. 课题说明

2252699 王泓烨 手写实现 kmeans 聚类和相关优化，撰写实验报告

2252699 王鉴戈 实现数据预处理和数据降维，撰写并整理实验报告

2252078 朱亚琨 手写实现层次聚类和相关优化，撰写实验报告

2250409 罗尹泽 手写实现 DBSCAN 聚类和相关优化，撰写实验报告

1.2. 课题目标（示例）

本课题的目标是使用多种聚类分析方法对数据集中的客户进行聚类，实现客户个性分析。这有助于企业更好地了解客户，并使他们更容易根据不同类型客户的具体需求、行为和担忧修改产品。课题使用的数据集来自于数据分析与数据挖掘竞赛 Kaggle，包含某商店各种客户的购物行为和个人信息的记录。为实现课题的目标，我们首先进行了**数据清洗**，并使用**特征工程**和 **PCA 数据降维**方法对数据进一步处理。然后我们分别**手写**实现了 **kmeans**，**层次聚类**，和 **DBSCAN** 三种聚类方法进行聚类分析，并实现了**结果可视化**，最后对三种机器学习方法进行了**优化与比较分析**。

1.3. 课题数据集

本课题使用的数据集为来自 Kaggle 网站中的 **Customer Personality Analysis**（https://www.kaggle.com/datasets/imakash3011/customer-personality-analysis/data）数据集，该数据集共包含了 2240 条客户相关数据，包括客户的学历、收入、购买记录、消费金额等 29 个属性，用于项目对客户类别进行进一步的聚类划分。

二、实验报告

2.1. 数据准备

下载 kaggle 网站的 **Customer Personality Analysis** 的数据集，获得一个大小为 216KB 的.csv 格式的数据文件，共包括 2240 条客户数据，每条数据包括 29 个属性。

2.2. 数据预处理

2.2.1 数据清洗

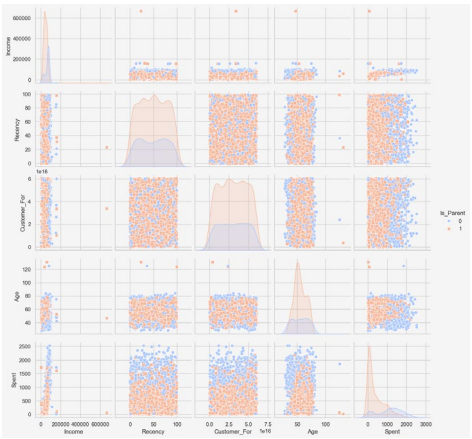
通过观察数据可以发现，原始数据中收入（Income）字段存在缺失值。Dt_Customer 字段表示客户加入数据库的日期，但尚未解析为日期时间格式（DateTime）。数据中存在一些分类特征（字段类型为 object），需要将这些特征编码为数值形式。

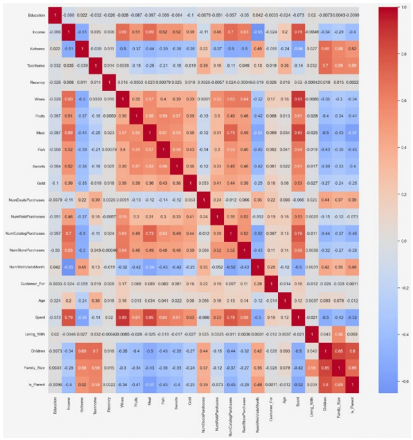
缺失数据较少，这里直接删除了存在缺失值的数据（共 24 条，约 1%）；对于日期类型的数据，将 string 类型的数据做格式转换，转换为日期格式，方便比较和计算。对于一些分类特征的属性，采用独热编码的方式对每一个 label 进行编码，转换为数值型的数据。

2.2.1 特征工程

通过详细阅读每个数据字段的意义，我们进行特征工程对当前数据集的特征进行优化和筛选。我们整合原有特征信息获得 6 个新增特征：客户年龄，总支出，生活状态，子女数量，家庭总人数，是否为父母；对原有的教育特征的分组进行简化，重命名一些奇怪特征名称，最后删除信息重复和冗余的特征。

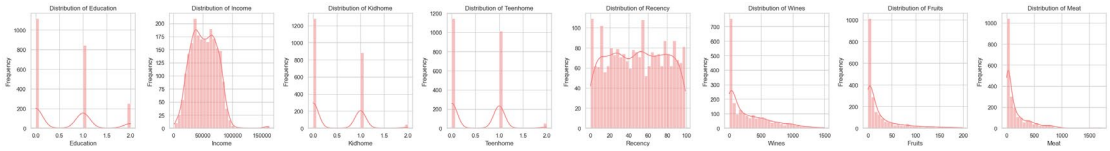
然后我们对特征中可能会出现异常的特征进行绘图观察，如左图：

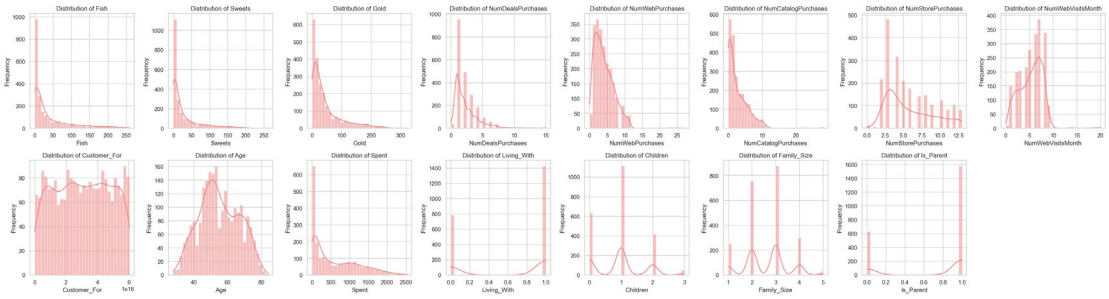




发现收入和年龄特征中确实存在一些异常值（如 131 岁的年龄），那么删除这些异常数据。然后我们绘制了相关性矩阵的热力图，观察了变量间的相似性，如右图，可以发现数据已经全部清洗干净，新的特征也全部包括在内。

进一步观察我们对所有清晰和特征工程后的特征的分布绘图进行了展示：

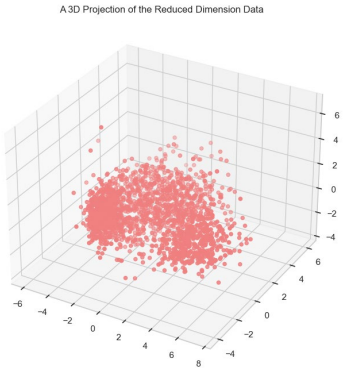




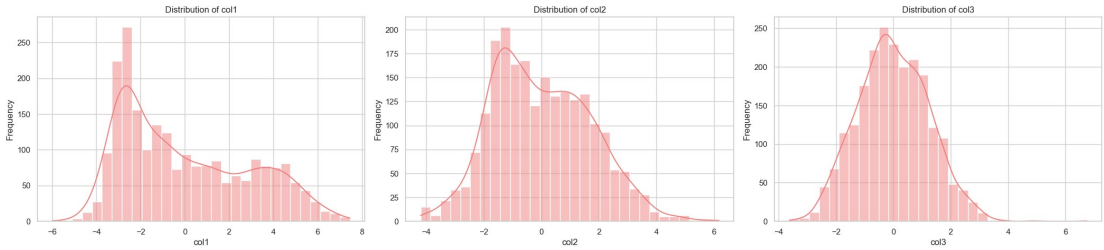
3.数据降维

我们使用主成分分析（PCA）降低数据集维度，可以提高可解释性，同时最大限度地减少信息损失。主成分分析（PCA）是一种常用的降维技术，通过线性变换将数据投影到一个新的坐标系中，新的坐标系由数据的主成分构成。PCA 的目标是将数据中的方差最大化，即找到最能解释数据变异性的方向，从而减少维度，同时保留最重要的特征。

降维前，数据包括 23 个特征，即 23 维的数据，我们使用 PCA 将其降维到 3 维，降维后数据点的三维分布如下：



这三个维度上的数据分布如下：



最后将清洗后和降维后的数据全部保存到.csv 文件中供后续使用。数据预处理详细过程在 code/DataPreprocess.ipynb 文件中。

2.3. 模型搭建

2.3.1 kmeans

K-Means 是一种基于划分的无监督聚类算法，旨在通过迭代优化将数据集

X={x1,x2,...,xn}划分为 k 个簇。K-means 的核心思想是最小化每个簇内的数据点到其簇中心的平方距离总和(SSE)，以提高簇内紧密性和簇间分离性。

对于给定的数据点 x，K-Means 的目标是找到最优的簇分配 C={C1,C2,…,Ck}和簇中心 u={u1,u2,…,un}，使得以下目标函数最小化：

$$J = \sum_{i=1}^{k} \sum_{x \in Ci} \|x - ui\|^2$$

实验中我们手写实现了 K-means 模型和 K-means++模型：

（1）K-means 模型：

- 假定特征是连续型数据，基于欧几里得距离度量数据点与簇中心的相似性。
- K-means 通过最小化簇内平方距离总和（SSE），以提高簇内点的相似性。

（2）kmeans++模型：

- K-means++ 是对 K-means 初始化方式的优化，通过更合理的中心选择，提高模型性能，减少收敛时间，避免陷入局部最优。
- K-means++ 在数据点分布不均或簇之间差异较大时更有效，适用于高维数据和复杂数据分布。

具体搭建过程详见 code\kmeans.ipynb

2.3.2 层次聚类

层次聚类是一种基于树状结构的无监督聚类算法，旨在通过构建一个嵌套的聚类树将数据集 $X = \{x_1, x_2, \dots, x_n\}$ 逐步聚合或拆分为不同的簇。层次聚类的核心思想是通过递归地合并相似的簇或拆分已有的簇来构建一个聚类层次结构，最终形成一个树状的聚类结构（称为树状图或树形图），从而揭示数据之间的关系。

在凝聚层次聚类（Agglomerative Hierarchical Clustering）中，每个数据点初始时被视为一个独立的簇，通过反复合并两个最近的簇，最终将所有的数据点聚合到一起。层次聚类的目标是通过最小化簇间的距离来找到一个最优的聚类结构。

对于给定的数据点 x，凝聚层次聚类的目标是找到最小化簇之间距离的方法 $C = \{C_1, C_2, \dots, C_k\}$ 其中每个簇包含一个或多个数据点，使得以下距离最小化：

$$J = \sum_{i=1}^{k} \sum_{x \in C_i} d(x, C_i)$$

其中 $d(x, C_i)$ 表示数据点 x 与簇 C_i 之间的距离，常用的距离度量方法包括欧氏距离、曼哈顿距离等。

AgglomerativeClustering 手写实现

在手写实现的凝聚层次聚类（AgglomerativeClustering_diy）中，核心步骤如下：

1. 初始化：每个数据点被视为一个独立的簇。

2. 计算距离矩阵：根据指定的距离度量（如欧氏距离或曼哈顿距离）计算簇之间的距离矩阵。

3. 合并最近的簇：选择距离最小的两个簇进行合并，更新簇标签和距离矩阵。

4. 重复合并：不断重复合并过程，直到达到目标簇数 k。

5. 更新簇中心：根据每次合并后的簇结构更新簇中心位置。

在实现中，AgglomerativeClustering_diy 类中提供了 fit_predict 方法，用于执行聚类过程，并返回最终每个数据点所属的簇标签。在每一轮合并中，目标是找到距离最小的簇对，将它们合并，从而逐步构建聚类树直至达到预定的簇数。

具体搭建过程详见 code\HierarchicalClustering.ipynb

2.3.3 DBSCAN

DBSCAN（Density-Based Spatial Clustering of Applications with Noise）是一种基于密度的聚类算法。该算法特别适用于发现任意形状的簇，并能够有效处理噪声数据。DBSCAN 无需预先指定簇的数量，通过密度的定义来自主确定簇的结构。

- **核心思想**

DBSCAN 基于以下几个概念构建聚类：

1） 邻域（Neighborhood, Eps-Neighbors）：以某个点为中心，半径为 (ϵ)（eps）的范围内的所有点组成该点的邻域。

2） 密度（Density）：在某点的邻域内的点数即为该点的密度。

3） 核心点（Core Point）：若某点的邻域内包含的点数（包括该点本身）大于等于指定的阈值 $(minPts)$，则该点被称为核心点。

4） 边界点（Border Point）：位于某个核心点的邻域内，但自身邻域的点数小于 $(minPts)$ 的点。

5） 噪声点（Noise Point）：既不是核心点也不是边界点的点，被视为离群点或噪声。

- **算法步骤**

DBSCAN 通过密度连接的方式将点聚类为簇，主要步骤如下：

1） 标记所有数据点：

2） 遍历所有数据点，计算其邻域半径 $(epsilon)$ 内的点数。

3） 分类核心点、边界点和噪声点：

将点分为核心点、边界点和噪声点。

4） 构建簇：从任意一个未被访问的核心点出发，将所有密度直达的点标记为同一簇。若核心点之间通过密度可达性连接，则合并为同一簇。

具体搭建过程详见 code\DBSCAN.ipynb

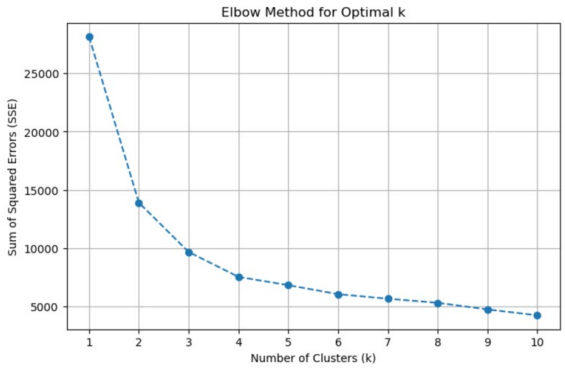
2.4. 模型训练测试

三种模型使用的数据： processed_data_pca.csv，大小为 2212×3。数据为经过 PCA 降维的特征数据，包含三列主成分。

2.4.1 kmeans

（1）数据预处理

簇数选择：通过计算不同簇数的簇内误差平方和绘制出肘部图，从而得到最佳簇数为 4.



（2）模型训练

分别使用 kmeans 模型和 kmeans++模型进行聚类，聚类过程包括以下步骤：

- **簇中心初始化：**使用 kmeans 或 kmeans++模型的方法初始簇中心。
- **分配：**对于每个数据点 x,根据其与各簇中心的距离，将其分配到最近的簇 Ci.
- **更新：**对每个簇 Ci，重新计算簇中心为簇内点的均值。
- **重复：**不断重复步骤 2 和 3，直到簇中心不再变化或达到最大迭代次数。

（3）模型结果分析

根绝聚类结果评估模型的表现。画出相似性矩阵，并分别计算轮廓系数和 Calinski-Harabasz 指数，并与库实现的聚类效果进行了对比。

2.4.2 层次聚类

（1）实现细节

使用手写实现的层次聚类（Agglomerative Hierarchical Clustering）算法，实验参数：

- n_clusters（簇的数量）：探索不同簇数（从 1 到 10）对聚类结果的影响，试图找到数据的最佳聚类数。
- linkage_method（链接方法）：采用 single（单连接）、complete（全连接）和 average（均值连接）三种链接方式，分析其对聚类效果和簇内紧凑性、簇间分离性的影响。
- metric（距离度量）：采用 manhattan（曼哈顿距离），探索距离度量对聚类质量的影响。

找到最佳的簇数和合适的链接方法，以获得高质量的聚类结果。并且分析簇数变化和链接方法变化对聚类效果的影响，评估手写实现的聚类算法与库函数实现的性能差异。

（2） 可视化

可视化聚类结果通过 3D 散点图展示不同簇的分布与关系，并通过相似矩阵热力图反映簇的紧凑性与分离性，从而帮助直观分析聚类效果。

（3）评价指标

评价指标通过 Silhouette Score（轮廓系数）、Calinski-Harabasz Score 和 SSE 评估聚类的紧凑性、分离性及聚类效果，帮助确定最佳的簇数和聚类方法。。

（4）与库函数实现的对比

通过对比手写实现与 Scikit-learn 的 AgglomerativeClustering，分析两者在聚类可视化、质量指标（Silhouette Score 和 Calinski-Harabasz Score）以及性能和精度上的差异。

2.4.2 DBSCAN

（1）算法

使用手写实现的 DBSCAN 算法，实验参数：

eps（邻域半径）：探索不同取值对聚类结果的影响。

min_samples（最小点数）：分析其对簇数量与噪声点分布的影响。

目标：找到最佳参数组合，分析参数变化对聚类效果的影响。

（2）可视化

绘制 3D 散点图 展示聚类结果。

使用 相似矩阵热力图 反映簇的紧凑性与分离性。

（3）指标评价

Silhouette Score：衡量簇的紧凑性和分离性。

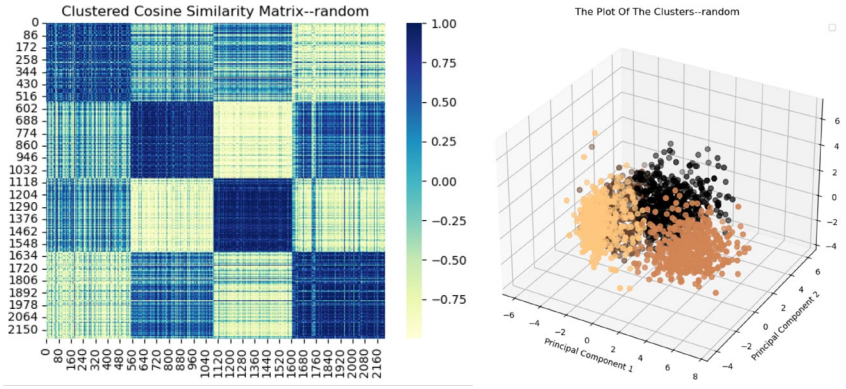
Calinski-Harabasz Score：评估簇内紧凑性与簇间分离性。

与库函数实现的聚类结果进行对比，说明手写实现的准确性。

2.5. 结果可视化

2.5.1 kmeans 模型

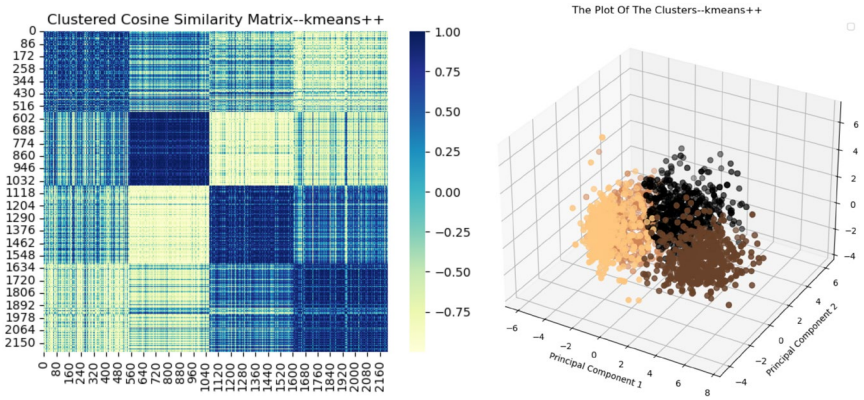
（1）kmeans 模型




```
random_Silhouette Coefficient: 0.3731871936044875
random_Calinski-Harabasz Index: 2005.4406878663453
```

从上述结果可知，聚类结果簇间分离度和簇内紧密度较好，轮廓系数较低可能是受到了数据分布的影响。

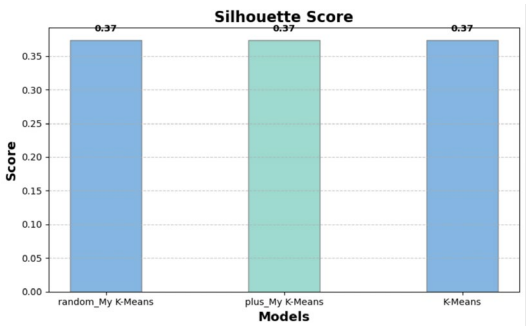
（2）kmeans++模型

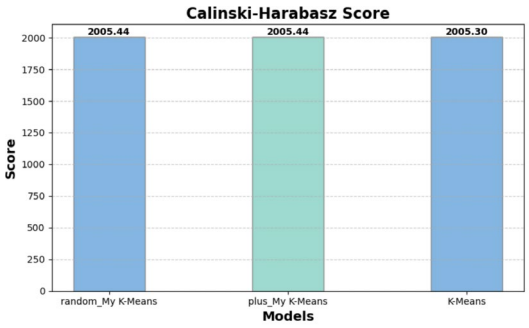



```
plus_random_Silhouette Coefficient: 0.3731457738398309
plus_Calinski-Harabasz Index: 2005.4396469509907
```

从上述结果可知，kmeans++的效果与 kmeans 相差不大，聚类结果簇间分离度和簇内紧密度较好，说明数据的分布较为规则，随机初始化的 K-Means 也有较高概率找到合理的初始中心。

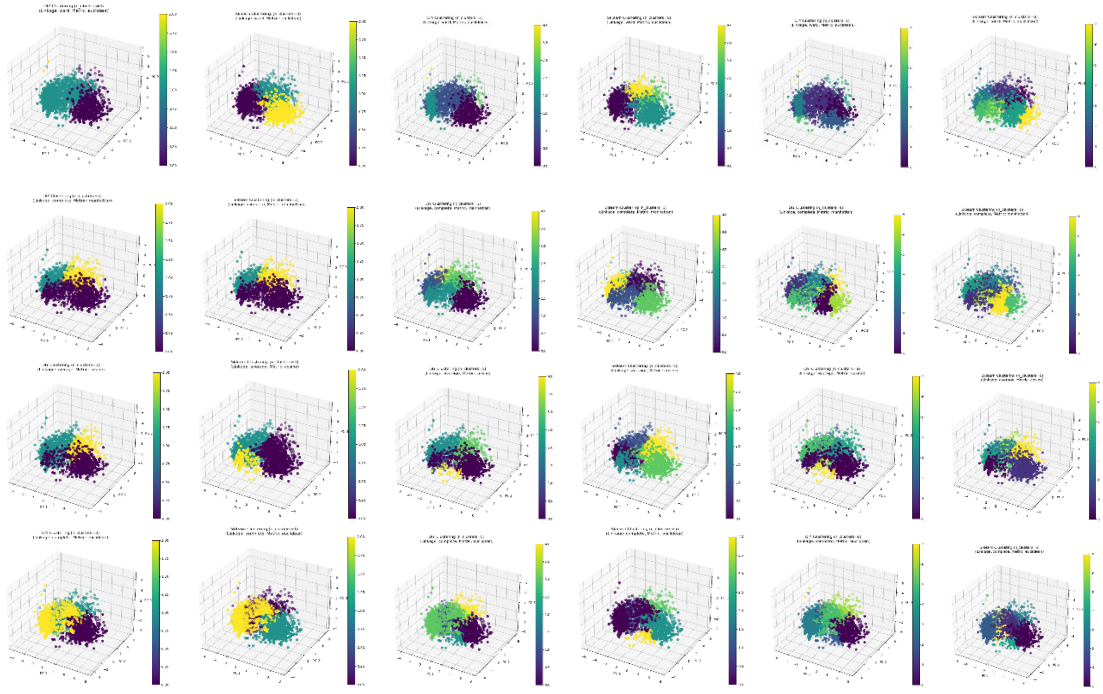
（3）与 sklearn 库中模型对比

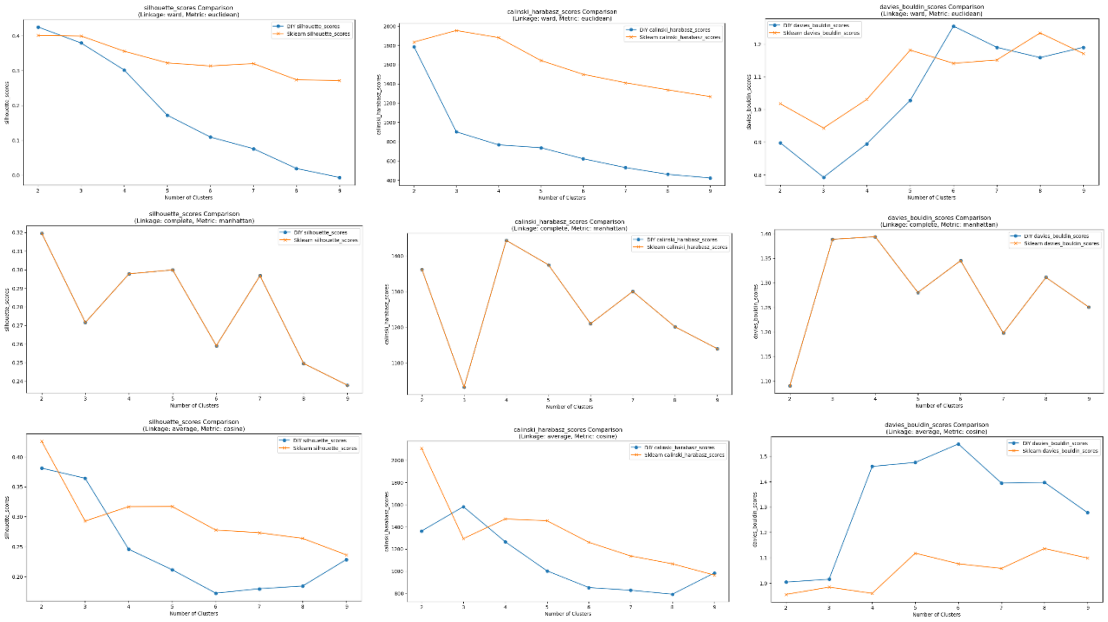


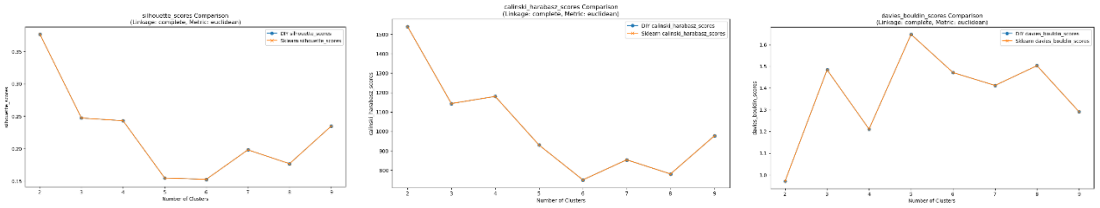


可以发现，手写的 kmeans 模型与相应的库中的模型实现了几乎一致的优秀表现。

2.5.2 层次聚类





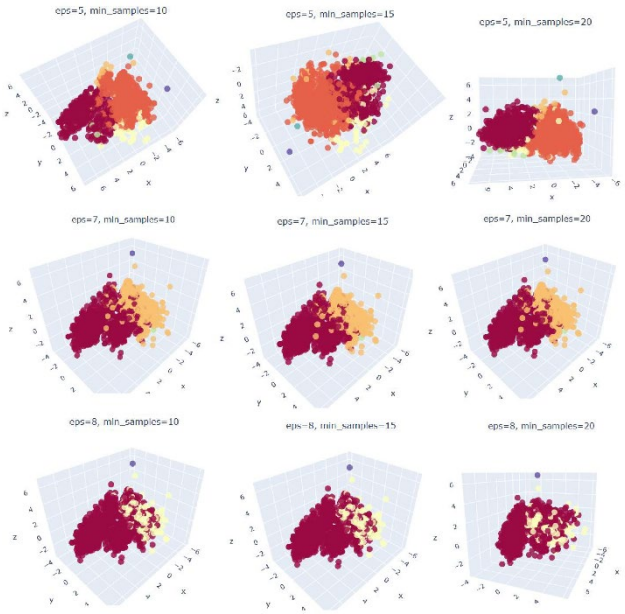


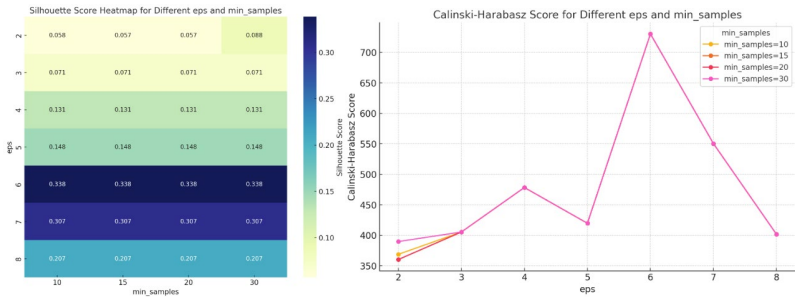
我们使用手写实现的层次聚类（Agglomerative Clustering）算法，分析了不同链接方法（Linkage Method）和相似度度量（Affinity Metric）对聚类效果的影响。实验选择了四种参数组合：Ward linkage 与 Euclidean metric、Complete linkage 与 Manhattan metric、Average linkage 与 Cosine metric、Complete linkage 与 Euclidean metric，聚类簇数从 2 到 10 不等。通过 3D 散点图和三个指标（Silhouette Score、Calinski-Harabasz Score、Davies-Bouldin Score）评估聚类结果。

可视化结果显示，在簇数为 3 时，手写实现和库函数都能清晰区分簇，簇内点聚集、簇间分离性良好；在簇数为 5 时，手写实现的聚类效果尤为突出。随着簇数增至 8，簇间分离性较弱，尤其是库函数实现，显示较大的簇重叠。

指标分析结果表明，手写实现的聚类在大多数情况下表现较好，特别是使用 Ward linkage 和 Euclidean metric 时，Silhouette Score 较高；使用 Complete linkage 和 Manhattan metric 时，Calinski-Harabasz Score 表现最佳，说明此组合下聚类效果较好；在 Davies-Bouldin Score 上，手写实现的得分普遍低于库函数，显示出更优的聚类质量。

2.5.3 DNSCAN





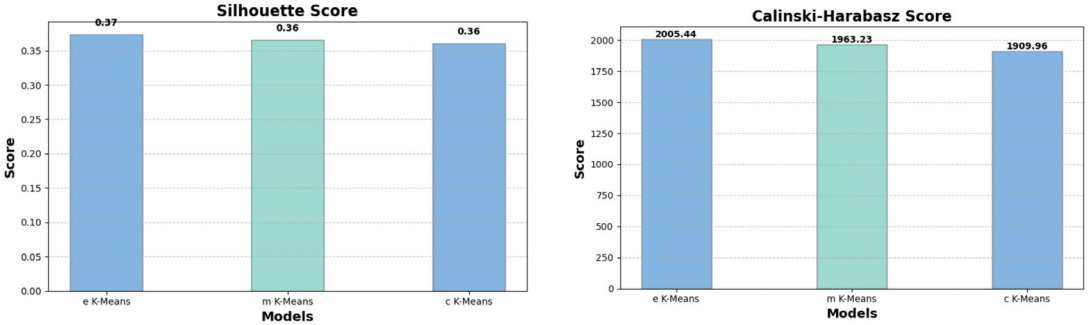
实验采用 DBSCAN 算法对数据集进行了系统的聚类实验，重点分析了邻域半径 eps 和最小点数 minPts 两个关键参数对聚类结果的影响。通过观察簇数量、噪声点数量、Silhouette Score 和 Calinski-Harabasz Score 四个指标，全面评估了不同参数组合下的聚类质量。实验结果表明，当 eps=2 时，簇数量较多，约为 43 个，噪声点数量较高，表明较小的邻域半径会将数据分割为多个小簇。随着 eps 的增大，簇数量逐渐减少，当 eps=8 时，簇数量降至 2 个。同时，噪声点数量随着 eps 的增大逐渐减少，当 eps>4 时，噪声点数量降为 0，表明较大的邻域半径可以包容更多的数据点。

在聚类质量方面，Silhouette Score 和 Calinski-Harabasz Score 综合反映了聚类的紧凑性和分离性。当 eps=6 且 minPts>10 时，Silhouette Score 达到最大值 0.338，Calinski-Harabasz Score 达到最大值 730.24，表明此时的聚类效果最优。较小的 eps 会导致簇内紧凑性不足，而较大的 eps 则会降低簇之间的分离性。此外，minPts 的变化对噪声点和簇数量有一定影响，但对聚类质量的总体趋势影响较小。

2.6. 分析和优化（要包含对两类模型的结果的比较讨论）

2.6.1 kmeans 模型分析和优化

根据上述聚类结果，尝试调整距离度量，以适应数据特点。分别采用欧几里得距离、曼哈顿距离和余弦距离来计算数据点到簇中心的距离。得到结果如下：

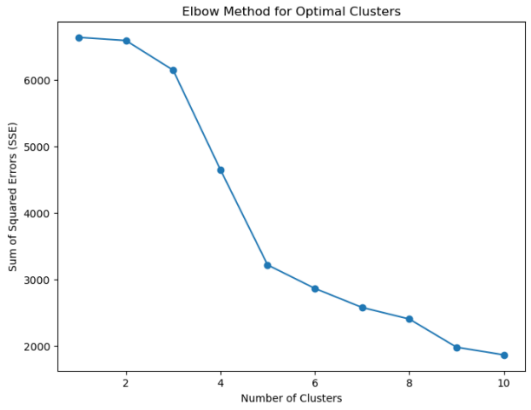


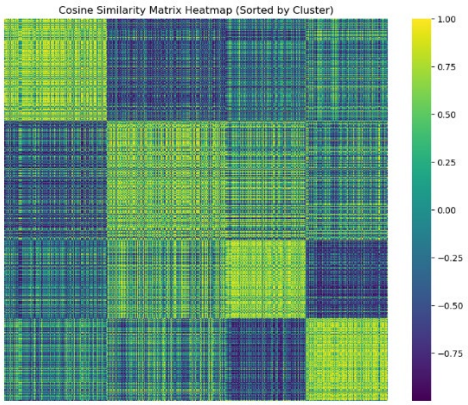
图中从左向右分别是欧几里得距离、曼哈顿距离和余弦距离，可以看到效果最好的是欧几里得距离。

2.6.2 层次聚类分析与优化

在本实验中，我们通过手写实现的层次聚类算法，选择了不同的簇数（n_clusters）并利用肘部法则评估聚类效果。肘部图显示，随着簇数增加，SSE 值的增长速度在 n_clusters=5 时开始加速，这表明簇间分离性下降。因此，n_clusters=5 被认为是最佳簇数。对于簇数较少（n_clusters<5）时，聚类效果较为稳定且紧凑，而超过 5 个簇后，聚类效果显著变差，表明簇的分离性减弱。

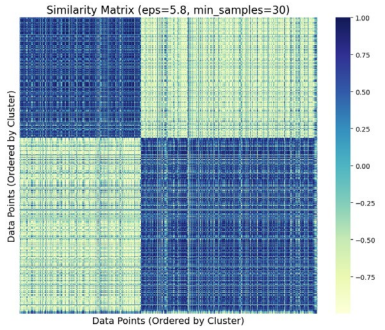
为进一步验证聚类效果，我们通过相似性矩阵热力图分析了聚类结果。在热力图中，每个簇内的点相似度较高，显示为紧密的黄色区域，而不同簇之间的相似度较低，呈现为深蓝色或绿色，表现出良好的分离性。通过这一分析方法，我们不仅直观地验证了最佳簇数的有效性，还证明了手写实现的层次聚类在数据集上的优良聚类效果。





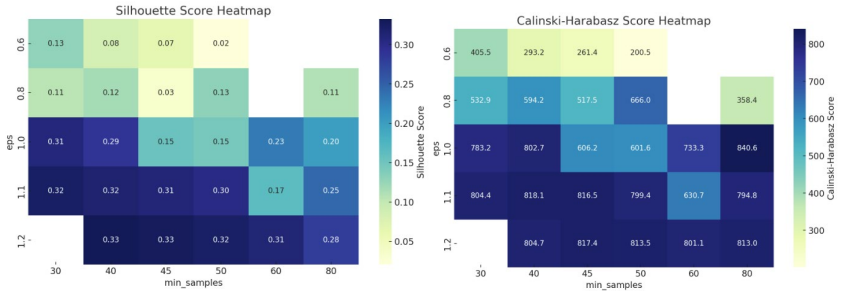
2.6.3 DBSCAN 分析与优化

在本实验中，我们通过绘制肘部图分析 DBSCAN 算法的最优参数组合，特别是邻域半径（eps）的选择。我们计算了不同 eps 值下的 SSE，并观察其变化趋势。实验结果表明，当 eps 小于 5.8 时，SSE 增长平缓，簇的紧凑性较高，但噪声点较多；当 eps 接近 5.8 时，SSE 的增长速度加快，表示簇间分离性逐渐降低；而当 eps 大于 5.8 时，SSE 迅速增加，聚类质量显著下降，簇间分离性差，多个小簇被合并成较大的簇。最终，我们选择了 eps=5.8 和 min_samples=30 作为最佳参数组合，通过热力图进一步展示了聚类结果，其中簇内数据点的相似度较高，簇间相似度低，表现出较强的分离性。此参数设置下，Silhouette Score 和 Calinski-Harabasz Score 均达到了较高的值，表明聚类效果既紧凑又有良好的分离性。




```
eps=5.8, min_samples=30
 - Number of Clusters: 4
 - Number of Noise Points: 0
 - Silhouette Score: 0.33997210574655257
 - Calinski-Harabasz Score: 753.3178443531787
```

此外，我们还将手写实现的 DBSCAN 与标准库函数的聚类结果进行了对比。尽管手写实现的 eps 值范围较大，主要是因为数据未进行归一化处理，而标准库推荐在标准化后的数据上计算，且使用了高效的邻域搜索算法（如 KD-Tree 或 Ball-Tree），但在调整参数后，手写实现和标准库在簇数量、噪声点数量、Silhouette Score 和 Calinski-Harabasz Score 等评价指标上的结果基本一致。通过这一对比，验证了手写实现的 DBSCAN 算法的有效性和可靠性，且两者在聚类结果上的一致性进一步证明了参数设置对聚类效果的重要影响。



3. 总结

在这次完成本次实验中，小组成员通过实现 K-means、层次聚类和 DBSCAN 等聚类算法，深入理解了这些算法的原理与实现过程，特别是在手写实现算法的过程中，掌握了优化方法，如 K-means++的初始化技巧。通过数据预处理，小组成员学会了如何处理缺失值、异常值，并进行特征工程和 PCA 降维，提升了数据质量和可解释性。在数据分析与可视化的过程中，我们提高了对数据分布和特征间关系的理解。这个课题让我们更深入地掌握了机器学习算法的应用和数据处理的实际操作，增强了我们分析与解决问题的能力，也为今后进行更复杂的数据分析任务打下了坚实的基础。