实验四

实验要求

实验目的：

1、理解聚类算法的原理；

2、能够使用聚类算法处理具体问题；

3、能够使用Python语言实现聚类算法。

实验内容：

1、 读取数据文件Wholesale customers data.csv，使用手肘法（Elbow Method）确定K值，输出随着K值变化SSE随之变化的图片；

2、 根据1中确定的K值，运用K-Means算法对数据集进行聚类，参数设定：最大迭代次数为100，距离函数为欧式距离；

3、 更改参数设定：最大迭代次数为500，距离函数为欧式距离；

4、 更改参数设定：最大迭代次数为1000，距离函数为欧式距离；

5、 将上述三次聚类结果用TSNE降维并进行展示；

6、 对比三次结果并加以分析（如使用概率密度函数图，分析聚类结果的不同特征）；

7、 尝试使用不同的随机种子，最大迭代次数为500，距离函数为欧式距离，通过TSNE降维与3降维结果对比，观察聚类结果是否发生变化；

8、 使用轮廓系数法再次确定K值，输出图片，对比与1中确定的K值是否一致，如果不一致使用新的K值进行聚类，并通过TSNE降维展示聚类结果；

9、 将上述实验内容的核心代码及实验结果截图放到“实验过程及分析”中。

实验过程以及分析

1．利用手肘法（Elbow Method）确定最优 K 值

实验描述

1、 读取数据文件Wholesale customers data.csv，使用手肘法（Elbow Method）确定K值，输出随着K值变化SSE随之变化的图片；

实验分析

目的在于通过簇内平方和（SSE）随 K 值变化的趋势，寻找“SSE-K 曲线”上的拐点。该拐点对应的 K 通常能在保持较小 SSE 的同时避免过度划分，从而较好平衡模型复杂度与数据拟合度。

代码实现

```
def elbow_method(data, k_range, save_path=None):
    """手肘法：绘制不同 K 值下的 SSE 曲线"""
    sse = []
    for k in k_range:
        km = KMeans(n_clusters=k, init='k-means++', max_iter=300,
random_state=42)
        km.fit(data)
        sse.append(km.inertia_)
    plt.figure()
    plt.plot(k_range, sse, 'o-')
    plt.xlabel('簇数 K')
    plt.ylabel('簇内 SSE')
    plt.title('Elbow Method')
    if save_path:
        plt.savefig(save_path)
    plt.show()
    return sse
```

2-4. 不同迭代次数下的 K-Means 聚类

实验描述

1. 根据1中确定的K值，运用K-Means算法对数据集进行聚类，参数设定：最大迭代次数为100，距离函数为欧式距离；
2. 更改参数设定：最大迭代次数为500，距离函数为欧式距离；
3. 更改参数设定：最大迭代次数为1000，距离函数为欧式距离；

实验分析

- 设置最大迭代次数为 100、500、1000，考察算法收敛速度与终态稳定性。
- 预期：100 次迭代可能未完全收敛，SSE 较高且聚类结果在边缘样本上存在漂移；500 次可达到收敛；1000 次则与 500 次结果基本一致，表明 500 次足以保证稳定。

代码实现

```
# 2-4. 不同最大迭代次数的聚类对比
results = {}
for max_iter in [100, 500, 1000]:
```

```
    km, labels = run_kmeans(data_scaled, n_clusters=k_opt,
max_iter=max_iter)
    results[max_iter] = labels
    print(f'Max iter={max_iter}, SSE={km.inertia_}')

def run_kmeans(data, n_clusters, max_iter=100, random_state=42):
    """执行 K-Means 聚类并返回模型与标签"""
    km = KMeans(n_clusters=n_clusters, init='k-means++',
max_iter=max_iter, random_state=random_state)
    labels = km.fit_predict(data)
    return km, labels
```

5. 将三次聚类结果用 t-SNE 降维并可视化

实验描述

将上述三次聚类结果用TSNE降维并进行展示；

实验分析

通过 t-SNE 将高维标准化后的客户特征映射至二维平面，使用不同颜色区分簇标签，可直观呈现各簇的分布与分离度，从视觉上验证迭代次数对簇形状与边界的影响。

代码实现

```
def silhouette_method(data, k_range, save_path=None):
    """轮廓系数法：绘制不同 K 值下的平均轮廓系数"""
    # 跳过 k = 1，因为 silhouette_score 仅在 2 ≤ k ≤ n_samples-1 有效
    sil_scores = []
    sil_k = []
    for k in k_range:
        if k < 2:
            continue
        km = KMeans(n_clusters=k, init='k-means++', max_iter=300,
random_state=42)
        labels = km.fit_predict(data)
        score = silhouette_score(data, labels)
        sil_scores.append(score)
        sil_k.append(k)
    plt.figure()
    plt.plot(sil_k, sil_scores, 'o-')
    plt.xlabel('簇数 K')
    plt.ylabel('平均轮廓系数')
    plt.title('Silhouette Method')
    if save_path:
        plt.savefig(save_path)
```

```
    plt.show()
    return sil_k, sil_scores
```

6. 对比三次结果并分析

实验描述

对比三次结果并加以分析（如使用概率密度函数图，分析聚类结果的不同特征）；

实验分析

对聚类结果中的关键变量（如 Fresh、Milk、Grocery）分别绘制核密度估计曲线：

- 对比不同迭代次数下，同一簇内各特征的分布差异。
- 有助于定量评估算法是否因迭代不足导致簇内样本分布偏移或簇间重叠加剧。

代码实现

```
def plot_tsne(data, labels, title):
    """使用 t-SNE 降维并可视化聚类结果"""
    tsne = TSNE(n_components=2, random_state=42)
    emb = tsne.fit_transform(data)
    plt.figure()
    sns.scatterplot(x=emb[:,0], y=emb[:,1], hue=labels, palette='tab10',
legend='full')
    plt.title(title)
    plt.show()
```

7. 不同随机种子下的聚类稳定性检验

实验描述

尝试使用不同的随机种子，最大迭代次数为500，距离函数为欧式距离，通过TSNE降维与3降维结果对比，观察聚类结果是否发生变化；

实验分析

在固定迭代次数（500 次）与距离度量（欧式距离）条件下，改变 `random_state`，分析不同初始质心对最终簇划分的影响。若整体簇形一致，则说明算法对初始化具备鲁棒性；若存在明显差异，则需在实际应用中多次试验并选取最优结果。

代码实现

```
# 7. 不同随机种子下聚类稳定性检验（max_iter=500）
seeds = [0, 42, 123]
for seed in seeds:
```

```
    km, labels = run_kmeans(data_scaled, n_clusters=k_opt, max_iter=500,
random_state=seed)
    plot_tsne(data_scaled, labels, title=f'随机种子={seed}')
```

8. 使用轮廓系数法（Silhouette Method）重新确定 K 值

实验描述

使用轮廓系数法再次确定K值，输出图片，对比与1中确定的K值是否一致，如果不一致使用新的K值进行聚类，并通过TSNE降维展示聚类结果；

实验分析

- 计算 2 ≤ K ≤ 10 范围内的平均轮廓系数，选取最高值对应的 K。
- 将该 K 重新用于 K-Means，并结合 t-SNE 可视化与手肘法结果对比，验证不同度量方法对最优簇数判断的一致性与差异。

代码实现

```
def plot_feature_pdfs(data_scaled, labels, features, title):
    """绘制各簇在指定特征上的核密度估计图"""
    df_scaled = pd.DataFrame(data_scaled, columns=features_all)
    df_scaled['cluster'] = labels
    for feat in features:
        plt.figure()
        sns.kdeplot(data=df_scaled, x=feat, hue='cluster',
common_norm=False)
        plt.title(f'{title} - 特征 {feat} 核密度估计')
        plt.show()
```

主函数（实验流程）

代码实现

```
def main():
    # 数据路径设置
    data_file = os.path.join(os.path.dirname(__file__), 'data', 'Wholesale
customers data.csv')
    df = load_data(data_file)

    # 选择数值特征并标准化
    df_features = df.select_dtypes(include=[np.number])
    global features_all
    features_all = list(df_features.columns)
    data_scaled = preprocess(df_features)
```

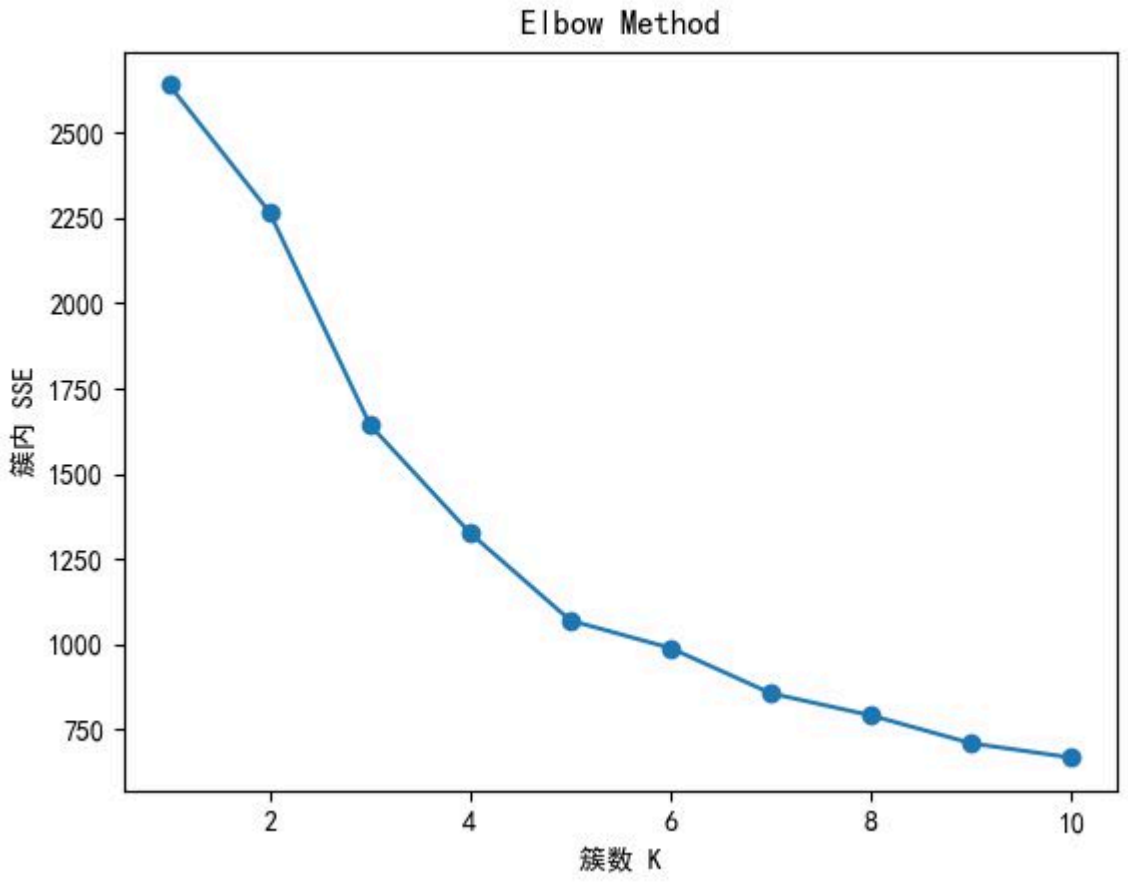
```
    # 1. 手肘法确定 K    k_range = range(1, 11)
    sse = elbow_method(data_scaled, k_range, save_path='elbow.png')
    # TODO: 根据图像拐点设定最优 K 值
    k_opt = 3

    # 2-4. 不同最大迭代次数的聚类对比
    results = {}
    for max_iter in [100, 500, 1000]:
        km, labels = run_kmeans(data_scaled, n_clusters=k_opt,
max_iter=max_iter)
        results[max_iter] = labels
        print(f'Max iter={max_iter}, SSE={km.inertia_}')

    # 5. t-SNE 可视化三次聚类结果
    for max_iter, labels in results.items():
        plot_tsne(data_scaled, labels, title=f'KMeans (K={k_opt},
max_iter={max_iter})')

    # 6. 使用核密度估计对比特征分布
    key_features = ['Fresh', 'Milk', 'Grocery']  # 可根据需求调整
    for max_iter, labels in results.items():
        plot_feature_pdfs(data_scaled, labels, key_features, title=f'迭代
{max_iter}')

    # 7. 不同随机种子下聚类稳定性检验（max_iter=500）
    seeds = [0, 42, 123]
    for seed in seeds:
        km, labels = run_kmeans(data_scaled, n_clusters=k_opt,
max_iter=500, random_state=seed)
        plot_tsne(data_scaled, labels, title=f'随机种子={seed}')

    # 8. 轮廓系数法确定 K 并可视化
    sil_k, sil_scores = silhouette_method(data_scaled, k_range,
save_path='silhouette.png')
    # 选取轮廓系数最大的 K    k_opt2 = sil_k[np.argmax(sil_scores)]
    print(f'轮廓系数法最优 K={k_opt2}')
    km2, labels2 = run_kmeans(data_scaled, n_clusters=k_opt2,
max_iter=300)
    plot_tsne(data_scaled, labels2, title=f'Silhouette 确定 K={k_opt2}')


if __name__ == '__main__':
    main()
```

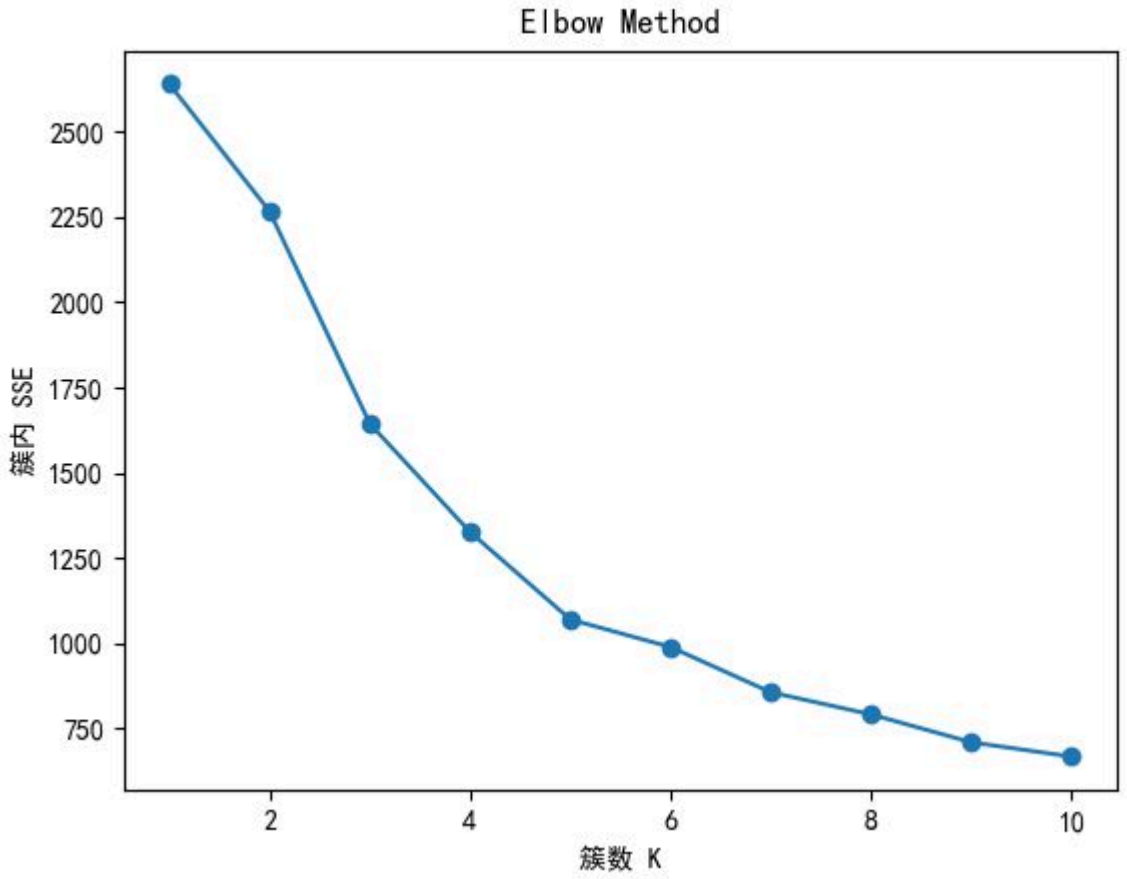
实验结果

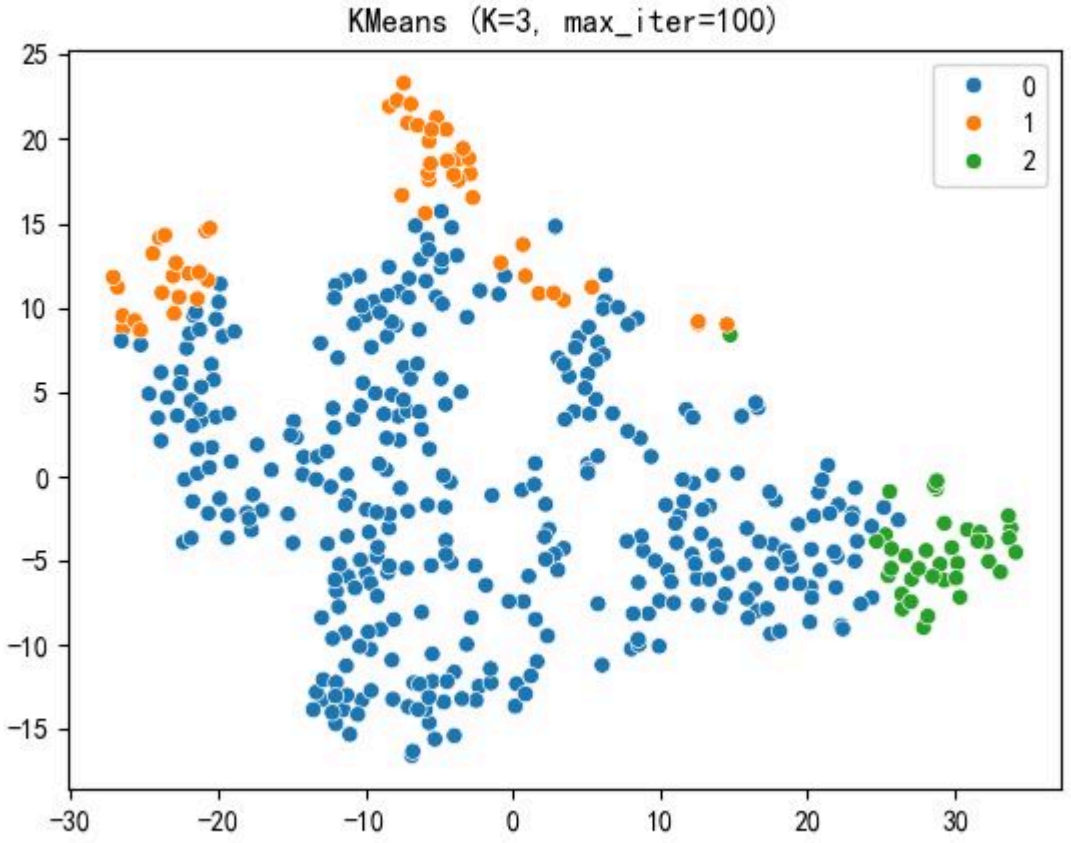
终端输出

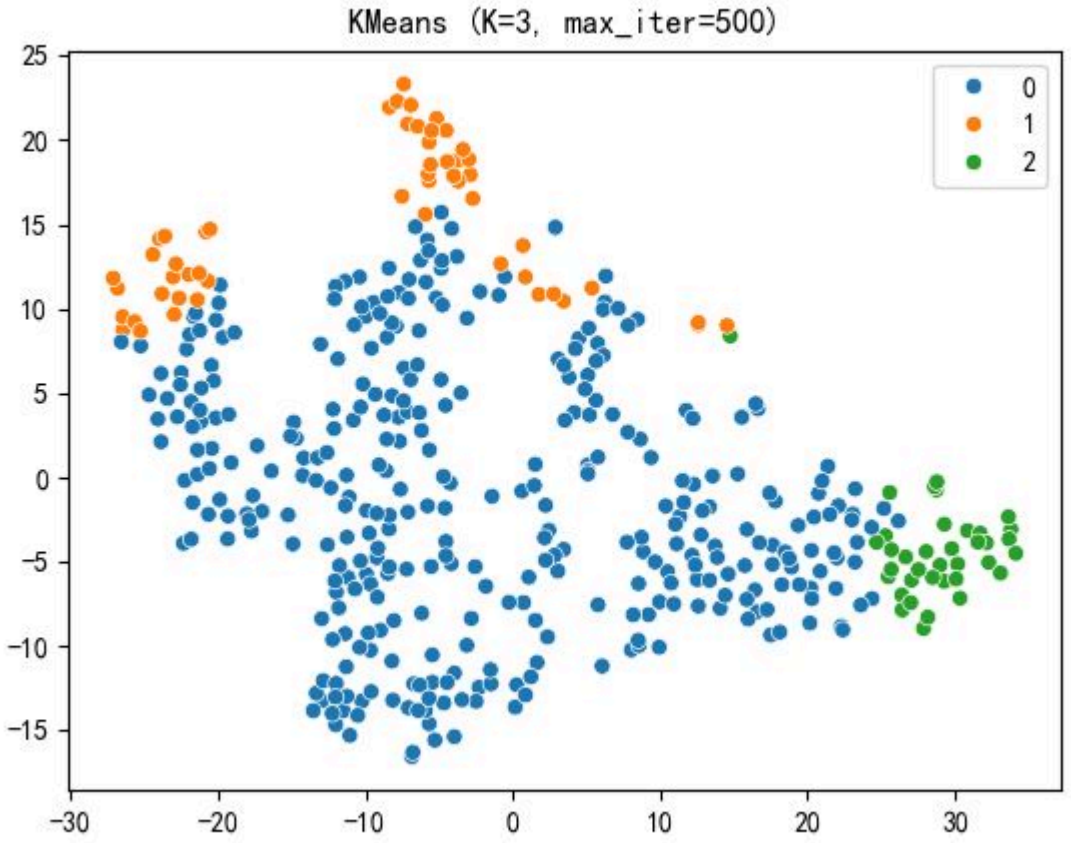
```
C:\Users\19065\miniconda3\python.exe D:\coding\简简单单挖掘个数据
\exp04\main.py
Max iter=100, SSE=1644.0598512347565
Max iter=500, SSE=1644.0598512347565
Max iter=1000, SSE=1644.0598512347565
轮廓系数法最优 K=3

进程已结束，退出代码为 0
```

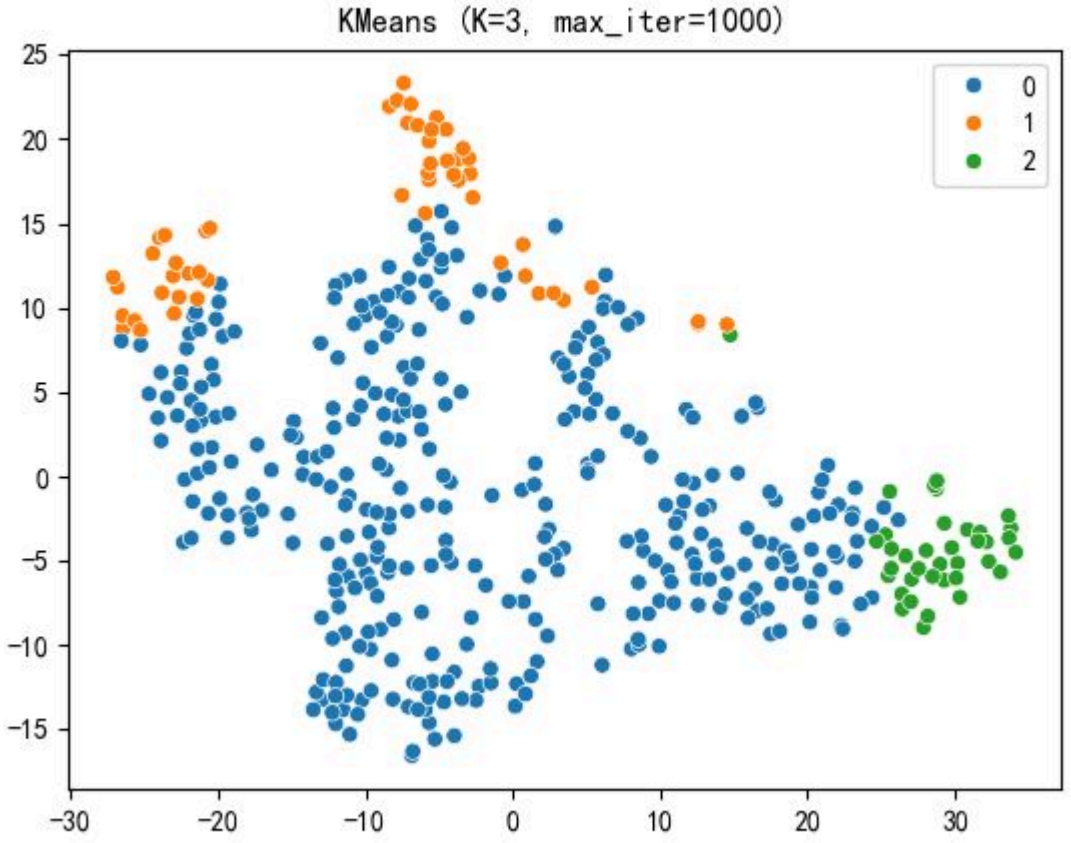
图片输出

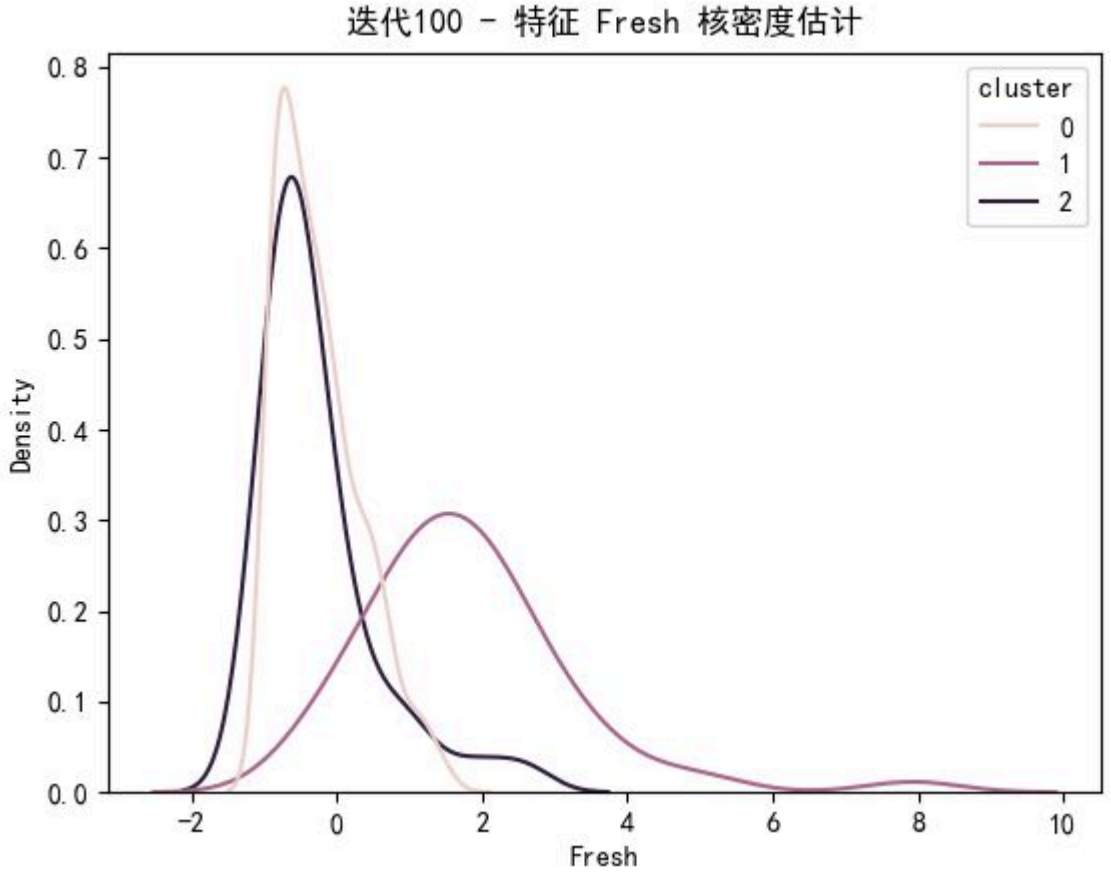


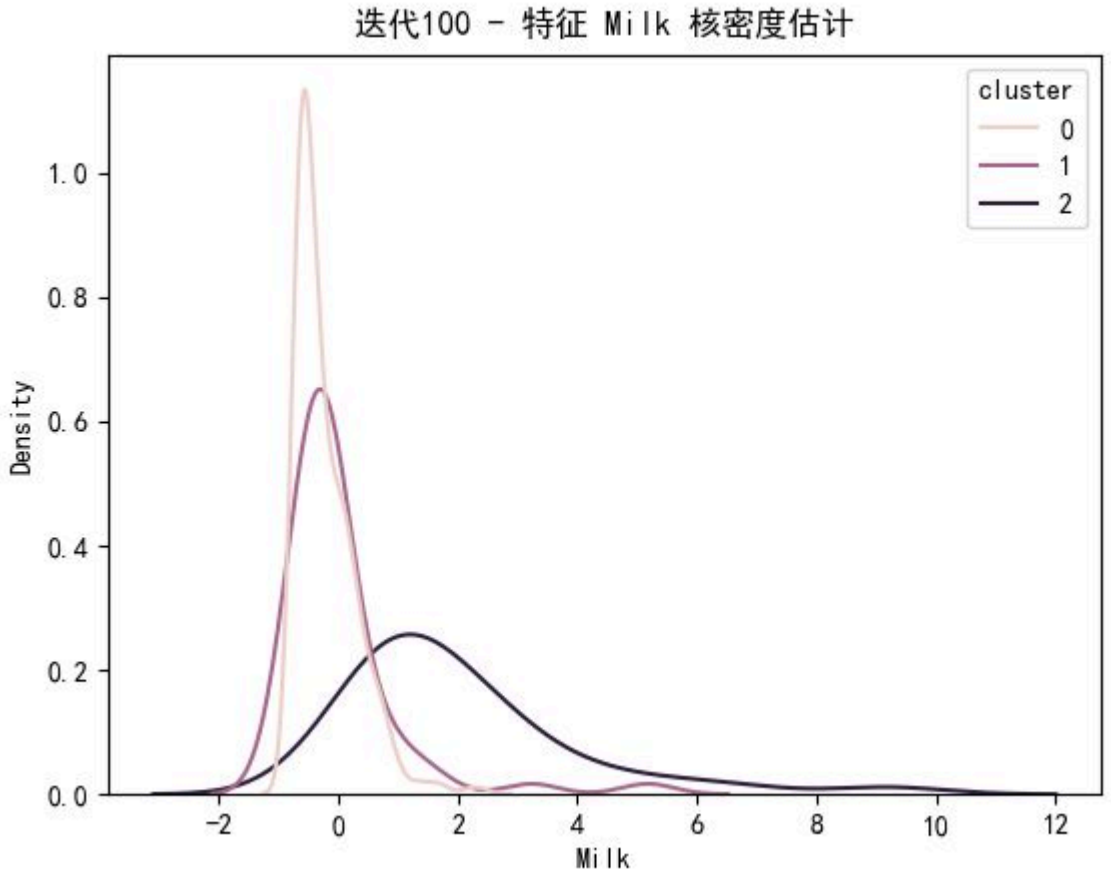

Elbow Method
簇内 SSE
簇数 K
2500
2250
2000
1750
1500
1250
1000
750
2
4
6
8
10

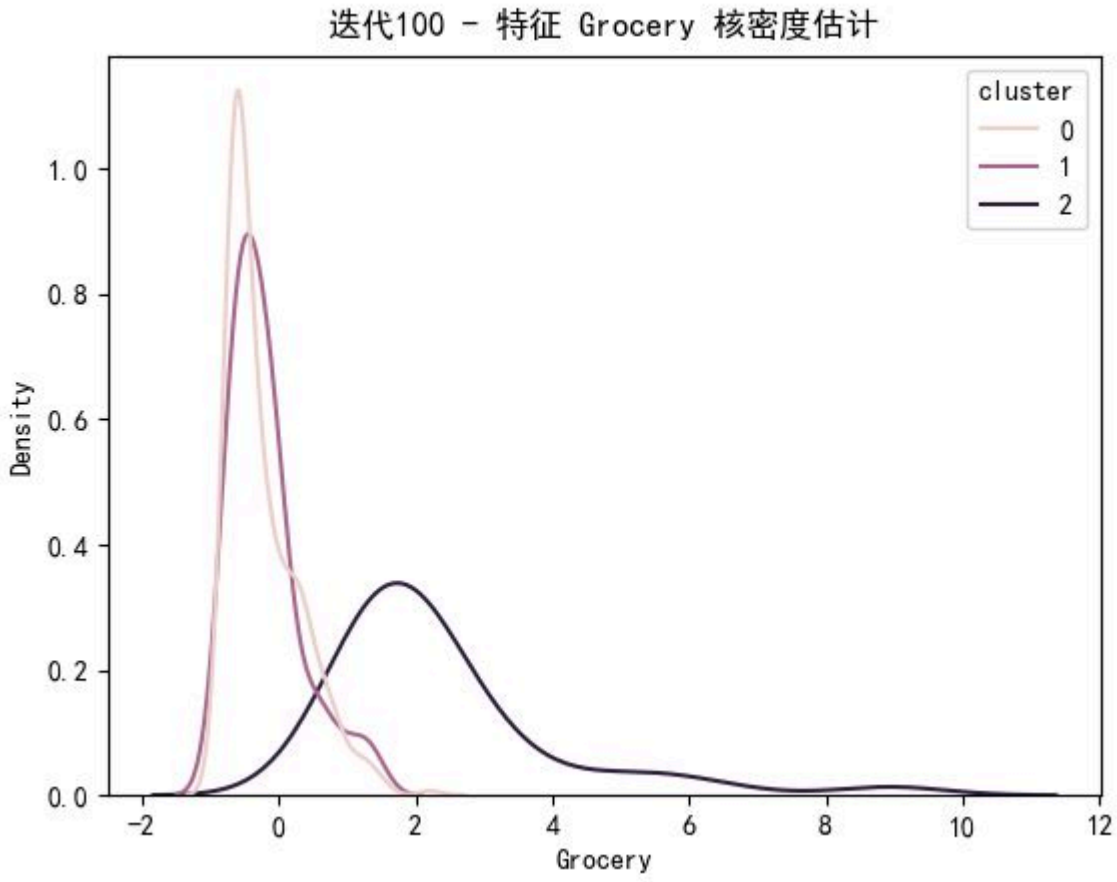

KMeans (K=3, max_iter=100)
0
1
2
25
20
15
10
5
0
-5
-10
-15
-30
-20
-10
0
10
20
30

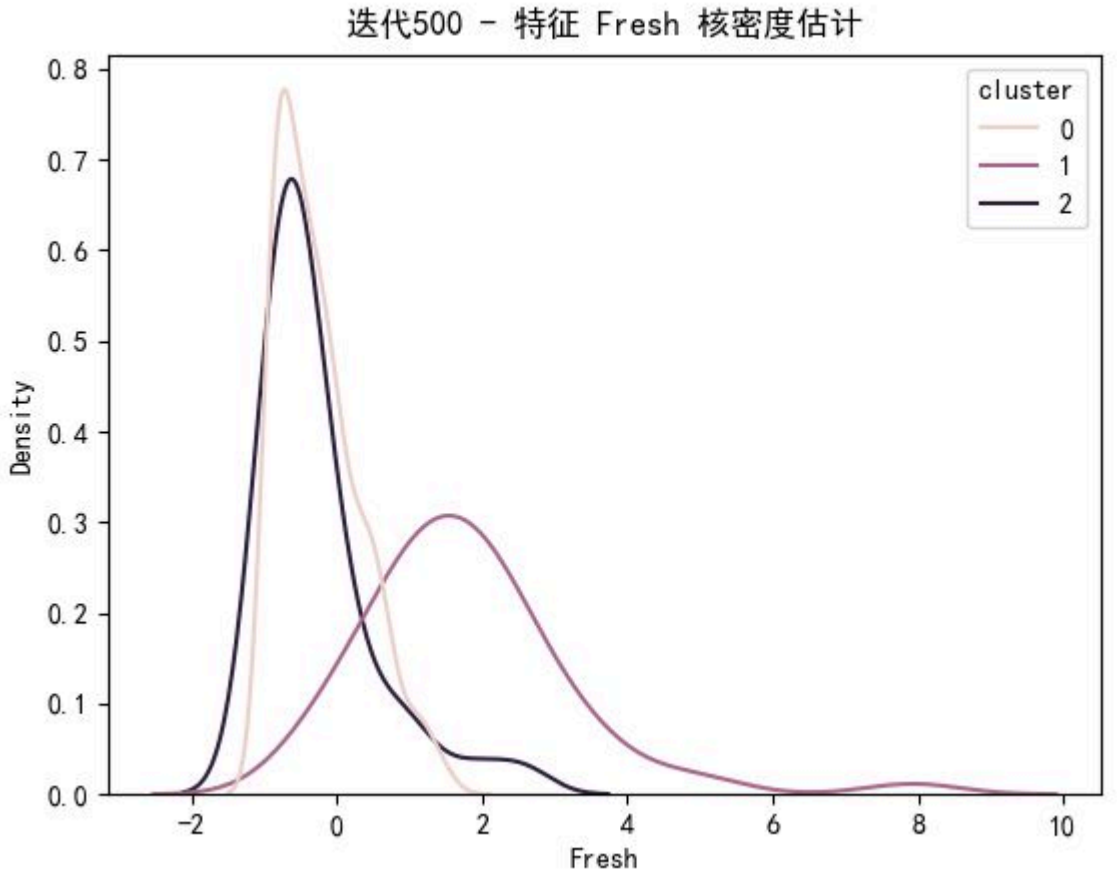

KMeans (K=3, max_iter=500)
0
1
2
25
20
15
10
5
0
−5
−10
−15
−30
−20
−10
0
10
20
30

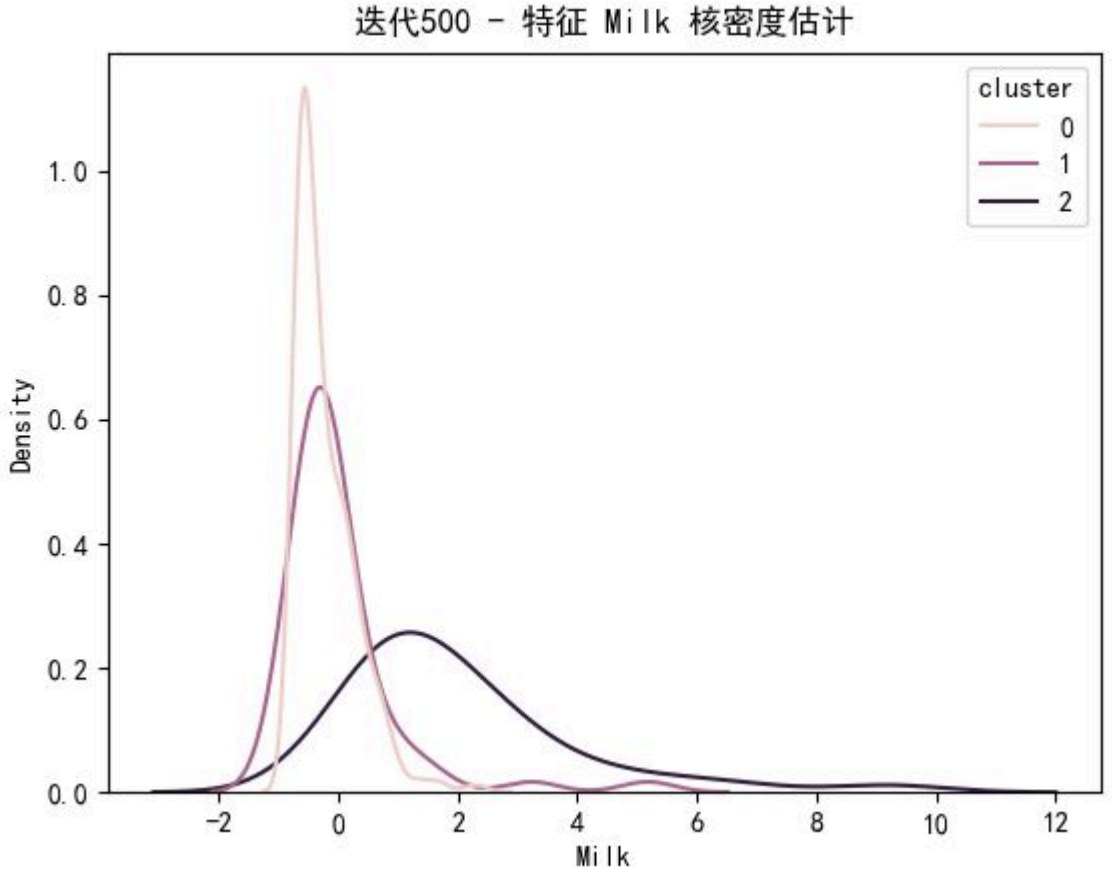

KMeans (K=3, max_iter=1000)
0
1
2
25
20
15
10
5
0
−5
−10
−15
−30
−20
−10
0
10
20
30

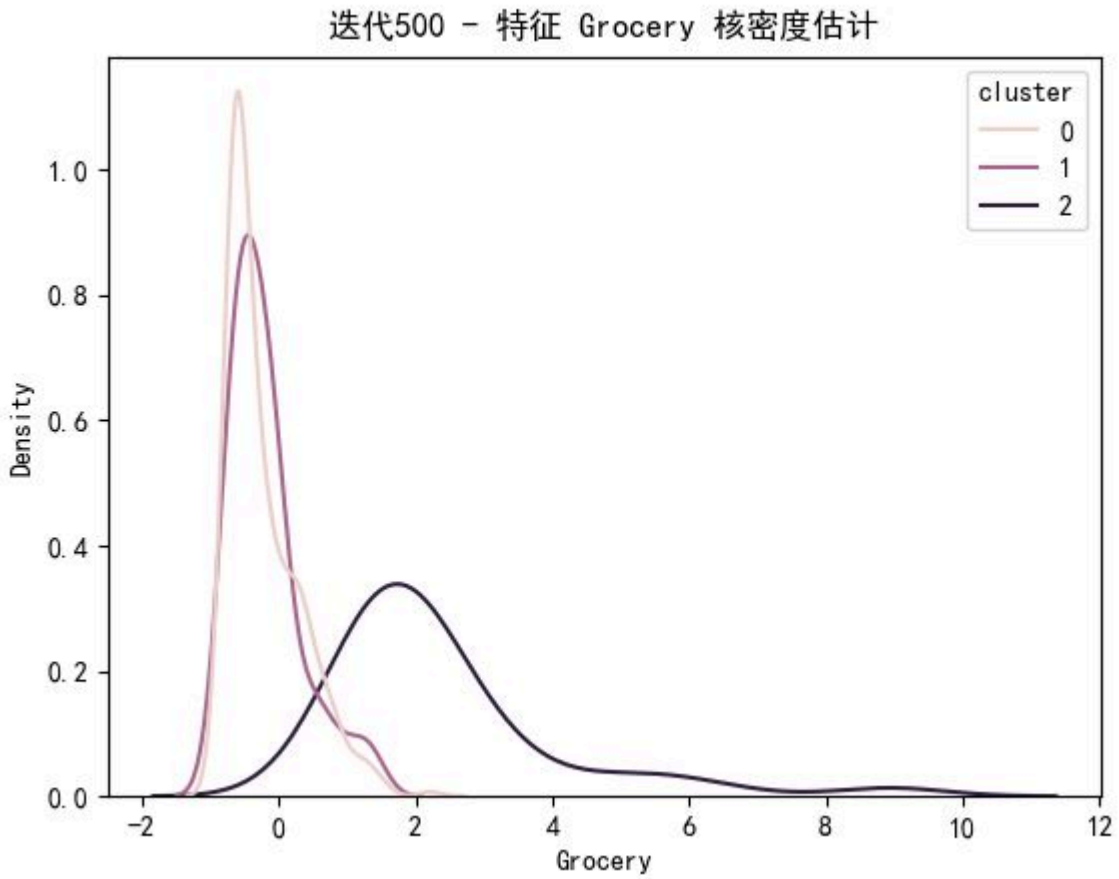

迭代100 - 特征 Fresh 核密度估计
cluster
0
1
2
Density
Fresh
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
−2
0
2
4
6
8
10

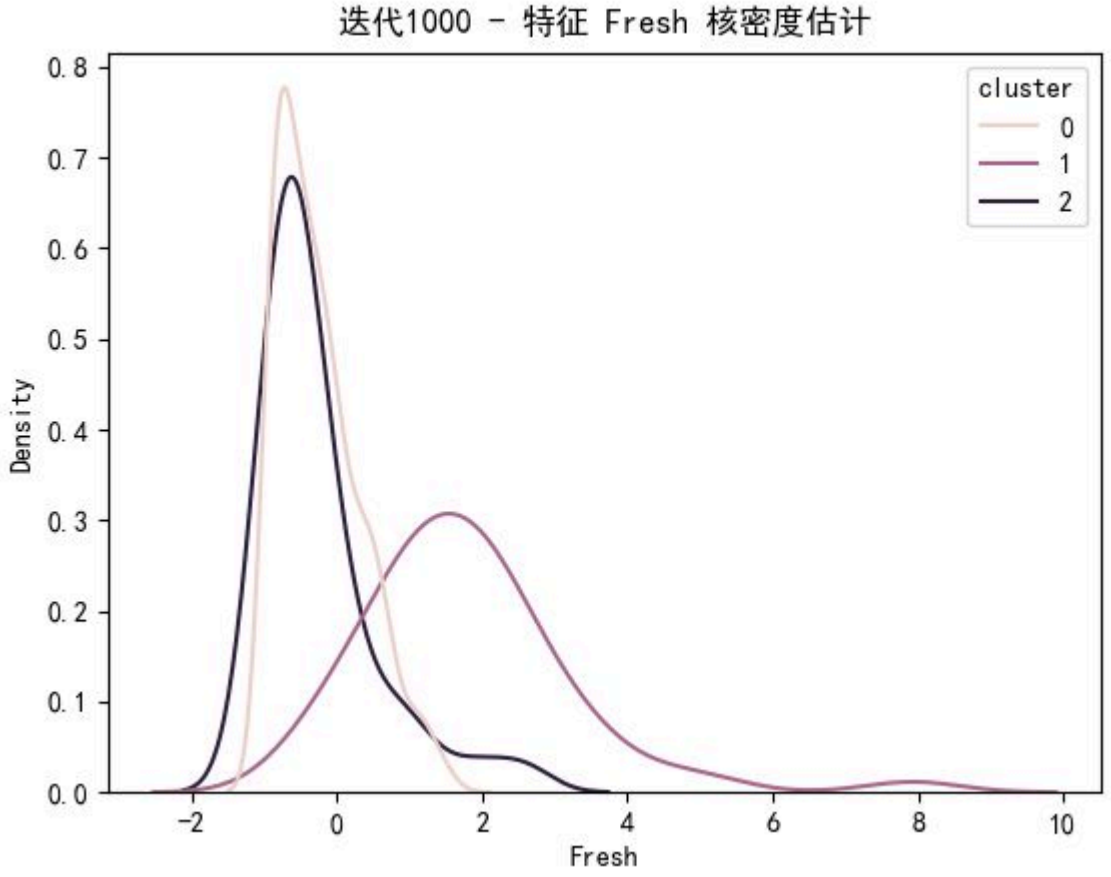

迭代100 - 特征 Milk 核密度估计
cluster
0
1
2
Density
Milk
0.0
0.2
0.4
0.6
0.8
1.0
−2
0
2
4
6
8
10
12

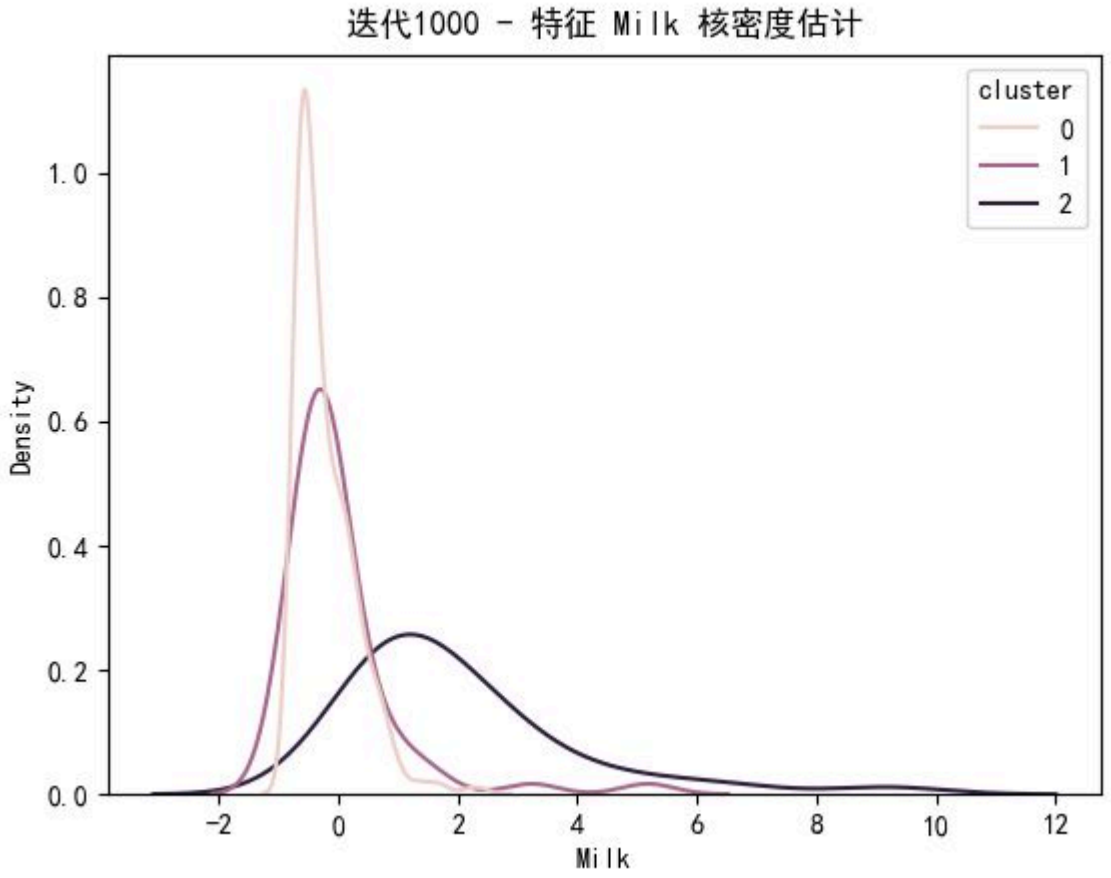

迭代100 - 特征 Grocery 核密度估计
cluster
0
1
2
Density
Grocery
1.0
0.8
0.6
0.4
0.2
0.0
-2
0
2
4
6
8
10
12

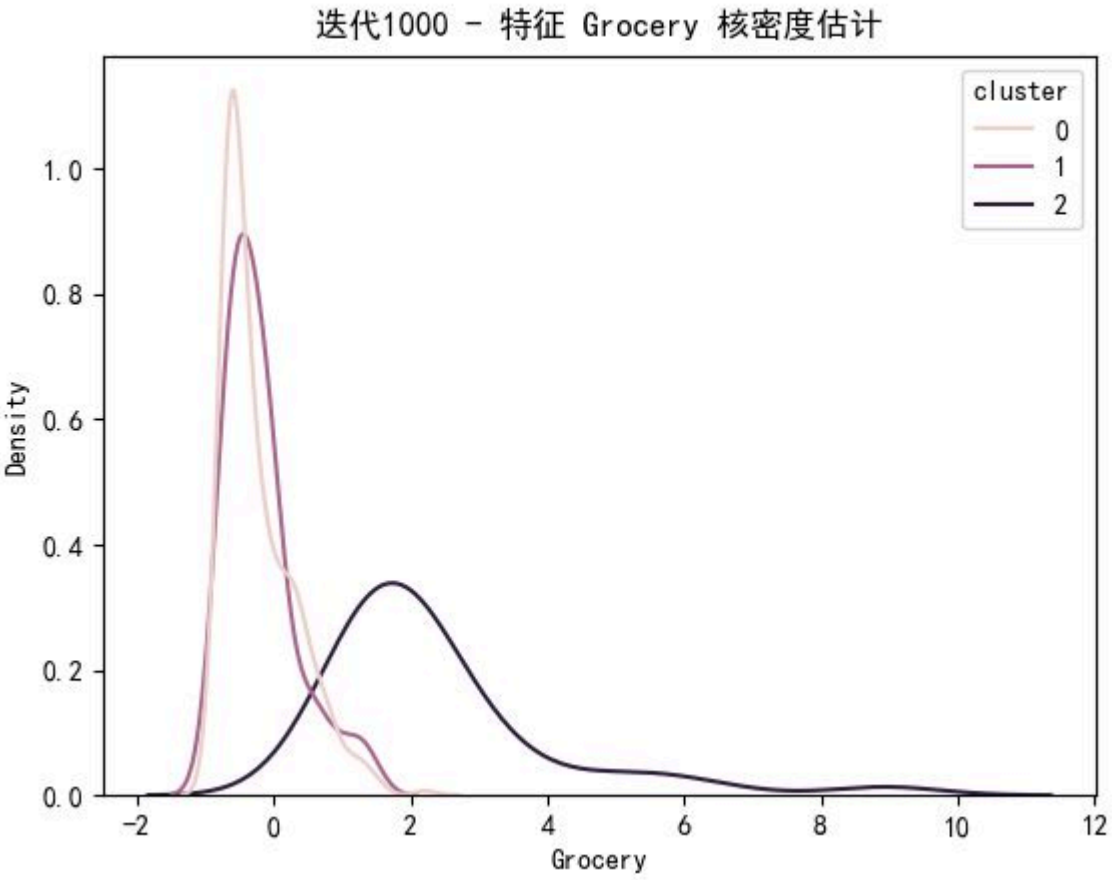

迭代500 - 特征 Fresh 核密度估计
cluster
0
1
2
Density
Fresh
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0.0
-2
0
2
4
6
8
10

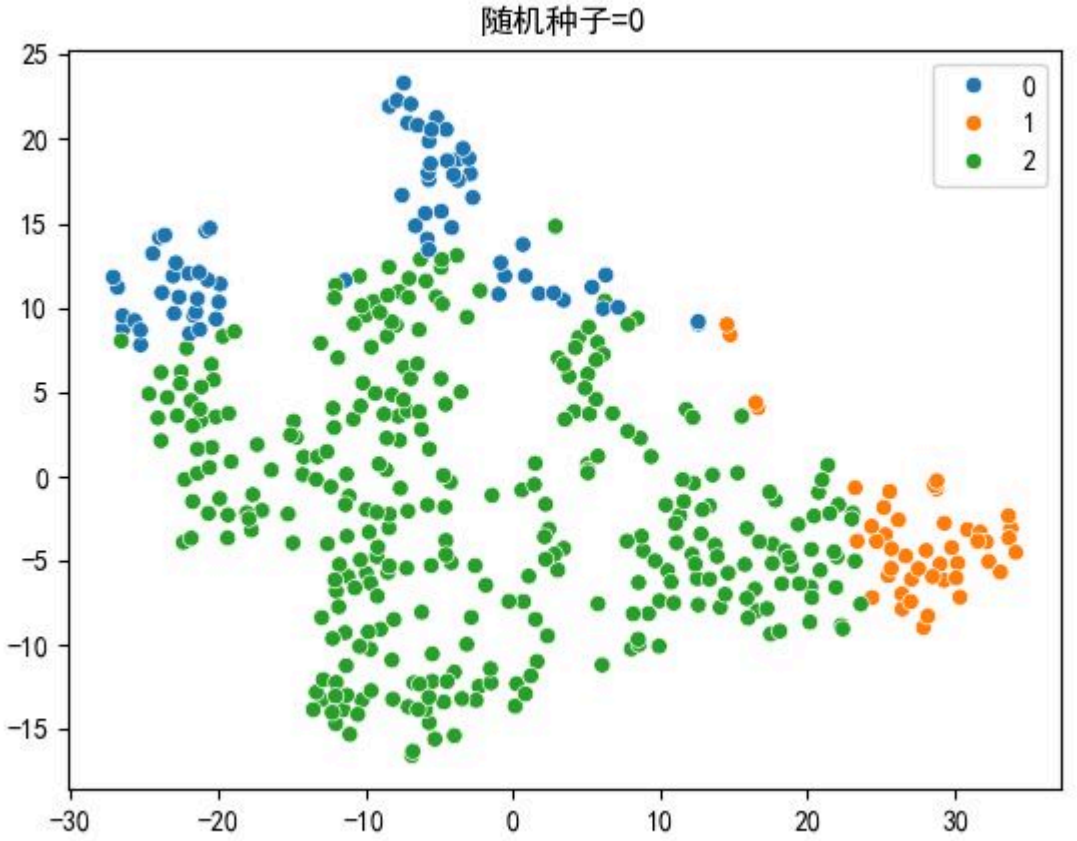

迭代500 - 特征 Milk 核密度估计
cluster
0
1
2
Density
1.0
0.8
0.6
0.4
0.2
0.0
-2
0
2
4
6
8
10
12
Milk

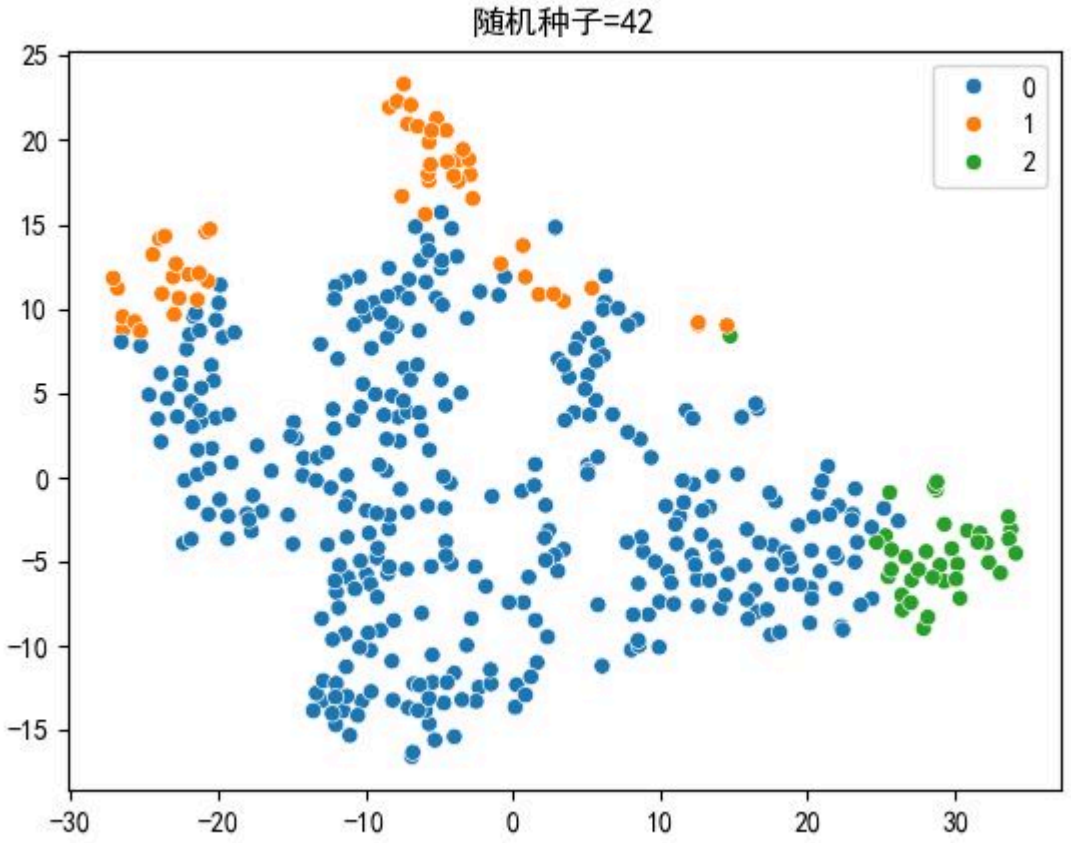

迭代500 - 特征 Grocery 核密度估计
cluster
0
1
2
Density
1.0
0.8
0.6
0.4
0.2
0.0
-2
0
2
4
6
8
10
12
Grocery

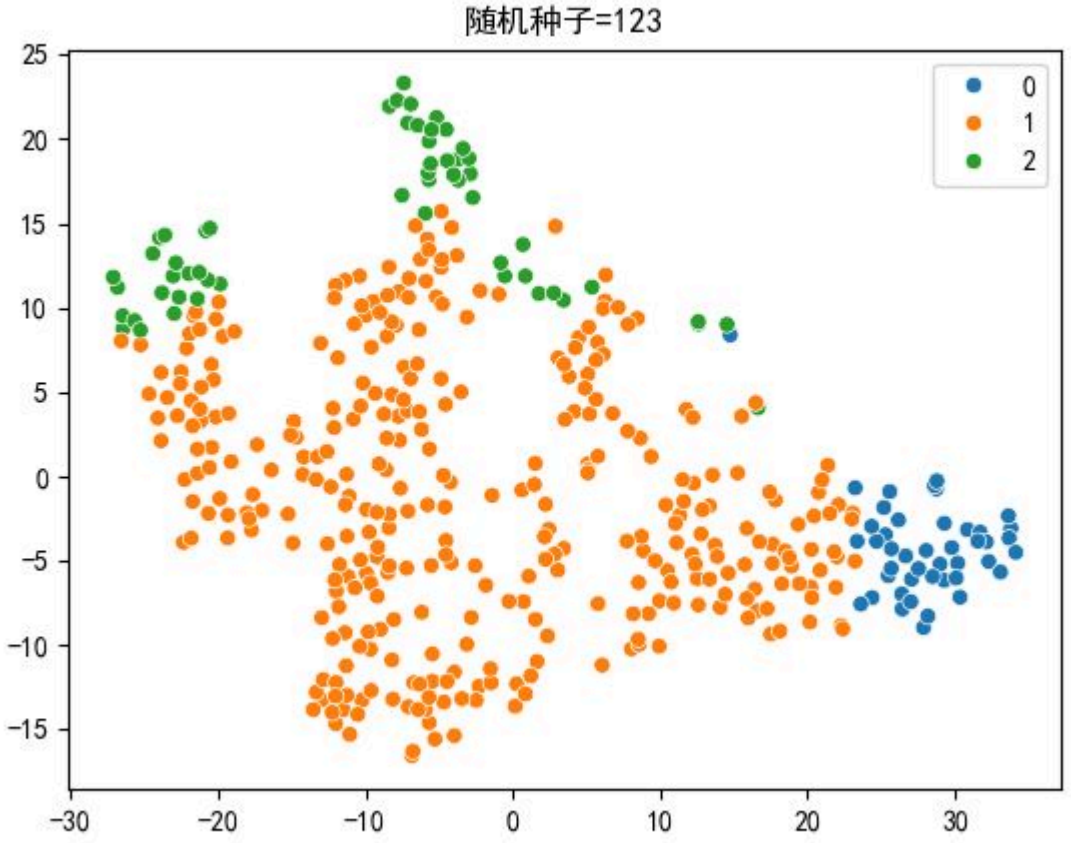

迭代1000 - 特征 Fresh 核密度估计
cluster
0
1
2
Density
Fresh
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
−2
0
2
4
6
8
10

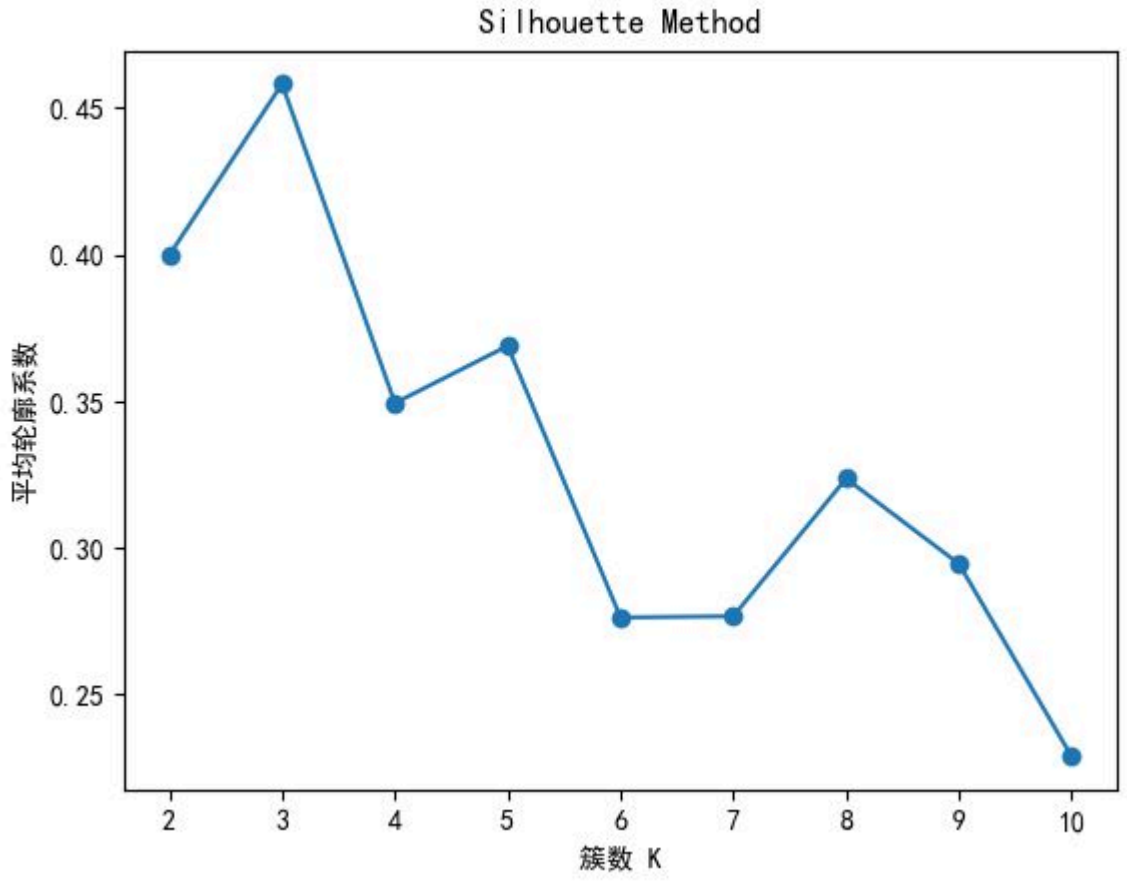

迭代1000 - 特征 Milk 核密度估计
cluster
0
1
2
Density
Milk
0.0
0.2
0.4
0.6
0.8
1.0
−2
0
2
4
6
8
10
12

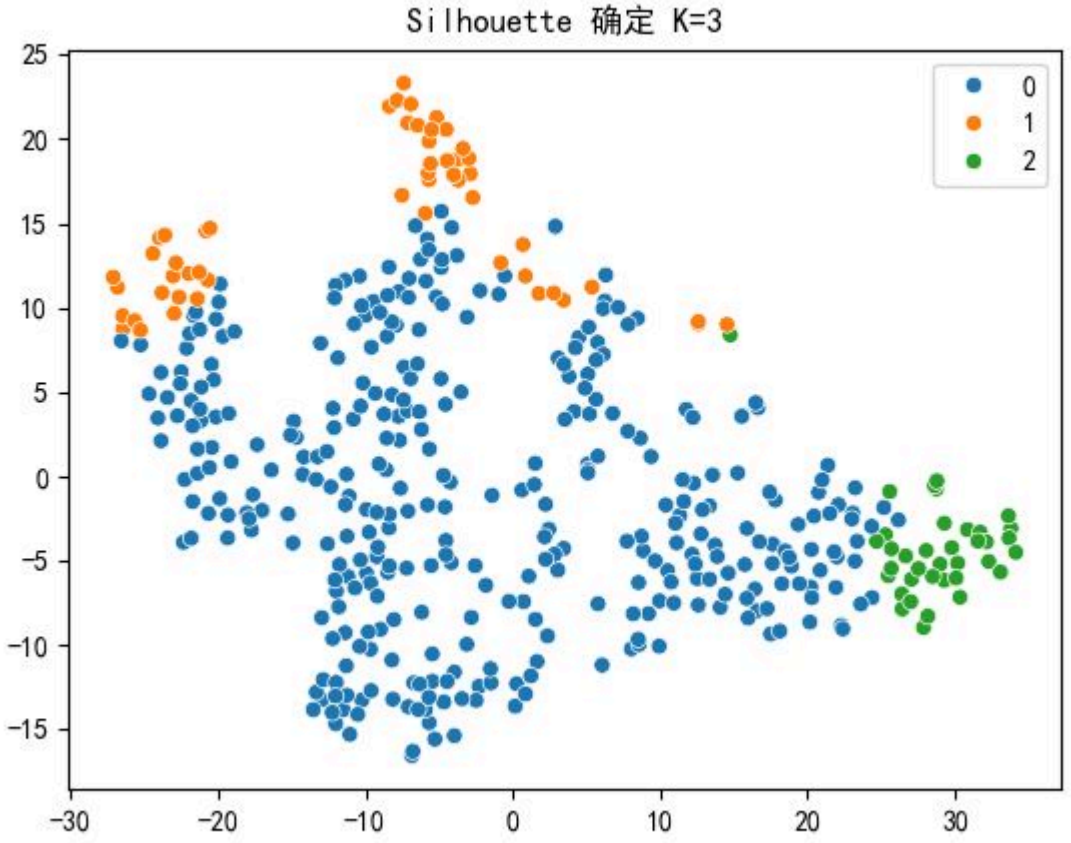

迭代1000 - 特征 Grocery 核密度估计
cluster
0
1
2
Density
Grocery
1.0
0.8
0.6
0.4
0.2
0.0
−2
0
2
4
6
8
10
12

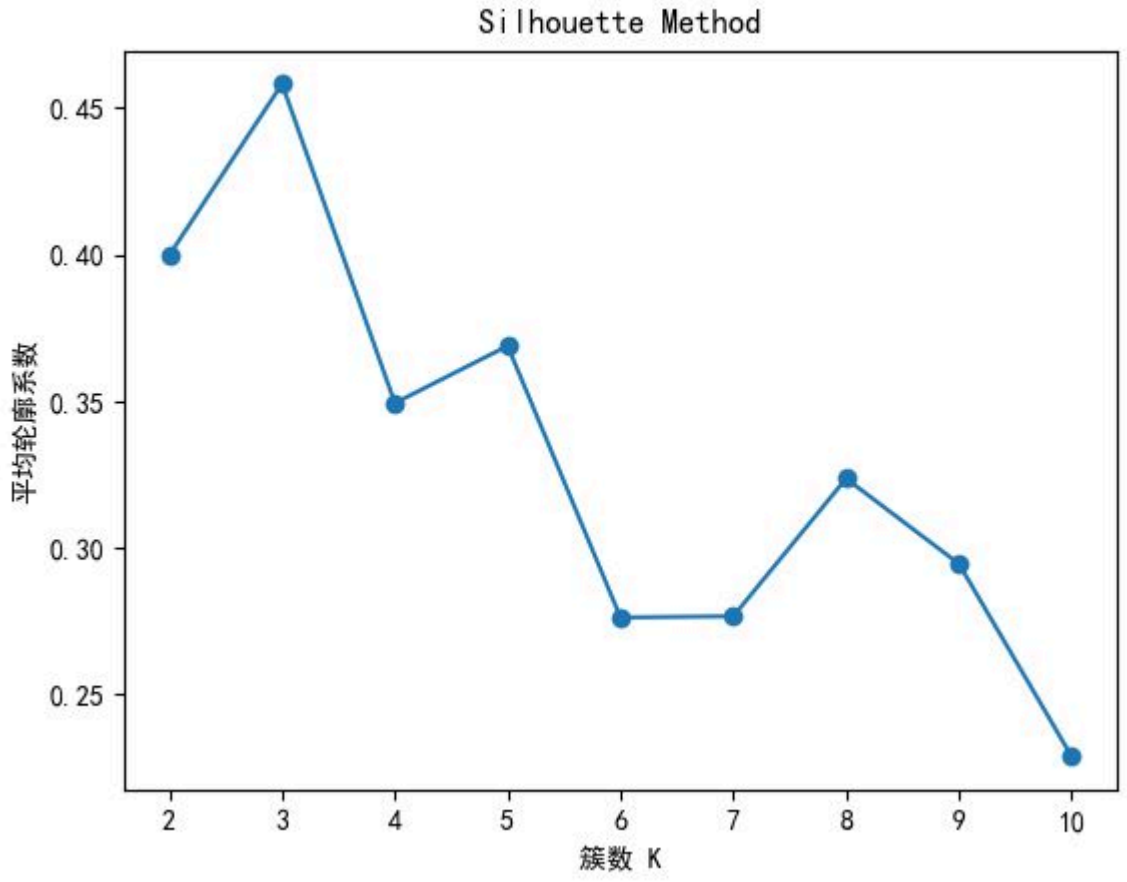

随机种子=0
0
1
2
25
20
15
10
5
0
−5
−10
−15
−30
−20
−10
0
10
20
30


随机种子=42
0
1
2
25
20
15
10
5
0
−5
−10
−15
−30
−20
−10
0
10
20
30


随机种子=123
0
1
2
25
20
15
10
5
0
−5
−10
−15
−30
−20
−10
0
10
20
30


Silhouette Method
平均轮廓系数
0.45
0.40
0.35
0.30
0.25
2
3
4
5
6
7
8
9
10
簇数 K


Silhouette 确定 K=3
0
1
2
25
20
15
10
5
0
-5
-10
-15
-30
-20
-10
0
10
20
30



实验体会

本次实验通过手肘法和轮廓系数法确定最优簇数，并对比了不同最大迭代次数（100、500、1000）以及多组随机种子对 K-Means 聚类结果的影响。结果表明，当迭代次数较低时（100 次）簇内 SSE 较高且聚类边界不稳定；达到 500 次后，SSE 基本收敛，簇划分趋于稳定；1000 次与 500 次结果高度一致，说明 500 次已足够。不同随机种子会导致少量样本在边缘簇间漂移，但整体簇结构差异不大，说明算法对初始化具有一定鲁棒性。t-SNE 可视化直观地展示了各簇的分布情况，而核密度估计则进一步揭示了关键特征（如 Fresh、Milk、Grocery）在不同簇中的分布差异。综合来看，多种方法的结合不仅提高了聚类结果的可解释性，也为实际应用中的参数选择提供了科学依据。