

**Python数据处理编程**

**实验2：航空公司客户价值分析**

**实验报告**

|  |  |
| --- | --- |
| **姓名** | **董道珠** |
| **学号** | **8210210121** |
| **班级** | **计科2102** |
| **专业** | **计算机科学与技术** |
| **年级** | **2021届** |
| **时间** | **2023/5/30** |

**中南大学计算机学院**

**2023**

目录

[一、实验目标与基本要求 3](#_Toc136386839)

[实验目标 3](#_Toc136386840)

[基本要求 3](#_Toc136386841)

[二、主要知识点、重点与难点 3](#_Toc136386842)

[主要知识点 3](#_Toc136386843)

[重点 3](#_Toc136386844)

[难点 4](#_Toc136386845)

[三、实验过程设计 4](#_Toc136386846)

[实验理论 4](#_Toc136386847)

[实验教学过程 4](#_Toc136386848)

[四、主要实现过程 5](#_Toc136386849)

[1 RFM模型基本原理 5](#_Toc136386850)

[2 LRFMC模型 6](#_Toc136386851)

[3 K-means算法 7](#_Toc136386852)

[4 数据的预处理 9](#_Toc136386853)

[5 数据处理 11](#_Toc136386854)

[K-means聚合 12](#_Toc136386855)

[画雷达图 14](#_Toc136386856)

[聚类评价 16](#_Toc136386857)

[6 结果分析 20](#_Toc136386858)

# 一、实验目标与基本要求

## 实验目标

重点结合航空公司客户价值分析的案例介绍K-Means聚类算法在客户价值分析中的应用。此外，介绍基于RFM客户价值分析模型的不足，使用K-Means算法构建航空客户价值分析LRFMC模型，详细的描述数据分析的整个过程。

## 基本要求

1. 了解RFM模型的基本原理。
2. 掌握K-Means算法的基本原理与使用方法。
3. 比较不同类别客户的客户价值，制定相应的营销策略。

# 二、主要知识点、重点与难点

## 主要知识点

1. RFM模型的基本原理。
2. K-Means算法的基本原理与使用方法。
3. 比较不同类别客户的客户价值，制定相应的营销策略。

## 重点

1. 航空客户价值分析的步骤和流程。
2. RFM模型的基本原理。
3. K-Means算法的基本原理与使用方法。
4. 比较不同类别客户的客户价值。

## 难点

1. RFM模型的基本原理。
2. KMeans算法的基本原理与使用方法。

# 三、实验过程设计

## 实验理论

1. 分析航空公司现状。
2. 认识客户价值分析。
3. 熟悉航空客户价值分析的步骤与流程。
4. 处理缺失值与异常值。
5. 构建爱你航空客户价值分析关键特征。
6. 标准化LRFMC 5个特征。
7. 了解K-Means聚类算法。
8. 分析聚类结果。
9. 模型应用。

## 实验教学过程

1. 处理数据缺失值与异常值。
2. 构建航空客户价值分析的关键特征。
3. 标准化LRFMC 5个特征。
4. 构建K-Means聚类模型。
5. 评价K-Means聚类模型。

# 四、主要实现过程

## 1 RFM模型基本原理

RFM模型是衡量客户价值和客户创造利益能力的重要工具和手段，在众多的客户关系管理 (CRM)的分析模式中，RFM模型是被广泛提到的。该机械模型通过一个客户的近期购买行为、购买的总体频率以及花了多少钱3项指标来描述该客户的价值状况。

其中三项指标分别为最近一次消费（Recency）、消费频率（Frequency）、消费金额（Monetary）。以上三个指标可以按照维度再细分，这样就可以细分出多类用户，根据每种用户的不同特点精准营销可以大大提高产品的经济效益。

比如重要价值客户（111）：最近消费时间近、消费频次和金额都高，我们将其设置为VIP。重要保持客户（011）：最近消费时间远，但消费频次和金额都很高，说明这是一个一段时间没来的忠诚客户，我们需要主动与其保持联系。等等类似的分类情况，有助于我们在运营企业时做出更好的决策。

尽管RFM模型有其优势，但也存在一些不足之处：

缺乏顾客生命周期视角：RFM模型主要关注最近的购买行为、频率和货币价值，但它并未考虑顾客的生命周期。一个顾客的长期价值和忠诚度不仅仅取决于他们最近的交易，还受到他们的购买历史、持续时间和变化模式的影响。因此，RFM模型无法提供对顾客在不同生命周期阶段的行为和价值的详细洞察。

忽略个性化需求：RFM模型以统一的方式对待所有顾客，没有考虑到顾客之间的个体差异和个性化需求。每个顾客都有独特的购买偏好、兴趣和行为特征，但RFM模型无法在个体层面上提供个性化的洞察，这对于制定个性化的营销策略和提供更好的顾客体验是不够的。

缺乏预测能力：RFM模型主要是对历史数据进行分析，根据顾客过去的行为来评估其价值和忠诚度。然而，对于企业来说，更重要的是能够预测未来的行为和价值，以便采取相应的措施。RFM模型本身并不具备预测能力，需要与其他模型或技术相结合才能实现更准确的预测和决策支持。

因此，我们在本实验中选用LRFMC模型。

## 2 LRFMC模型

LRFMC常用于航空公司对客户价值分析，正好契合我们本实验。LRFMC模型将客户聚类为重要保持客户，重要发展客户，重要挽留客户，一般客户，低价值客户，从而针对每种类别的客户制定对应的价格和服务。

首先，明确目标是客户价值识别，应用最广泛的模型是三个指标(消费时间间隔(Recency),消费频率(Frequency),消费金额(Monetary))以上指标简称RFM模型，作用是识别高价值的客户。消费金额，一般表示一段时间内，消费的总额。但是，**因为航空票价收到距离和舱位等级的影响，同样金额对航空公司价值不同。**因此，需要修改指标。选定变量，舱位因素=舱位所对应的折扣系数的平均值=C，距离因素=一定时间内积累的飞行里程=M，再考虑到，航空公司的会员系统，用户的入会时间长短能在一定程度上影响客户价值，所以增加指标L=入会时间长度=客户关系长度。因此，总共确定了五个指标，消费时间间隔R，客户关系长度L，消费频率F，飞行里程M和折扣系数的平均值C。以上指标，作为航空公司识别客户价值指标，记为LRFMC模型。

我们将在下面的代码中，将这几种数据提取出来并用K-mean算法将用户划分类别。

## 3 K-means算法

作为数据挖掘的一个重要研究课题，聚类分析技术越来越受到人们的关注。聚类就是将物理或抽象对象的集合分成多个相似的数据子集，同一个子集内的对象之间具有较高的相似度，而不同子集内的对象差别较大。K-means算法作为一种基于划分的动态聚类算法,它以误差平方和SSE作为聚类准则函数，具有简单有效、收敛速度较快、便于处理大型数据集等优点，从而获得了广泛的应用。

其一般步骤为：

1.选择K个初始聚类中心点：首先从数据集中随机选择K个对象作为初始聚类中心点。这些中心点可以是随机选择的，也可以使用其他启发式方法。

2.分配数据点到最近的聚类中心：对于数据集中的每个对象，根据其与各个聚类中心的距离，将其分配给最近的聚类中心。

3.更新聚类中心：计算每个聚类的平均值（或其他统计量），并将其作为新的聚类中心。

4.重复步骤2和3，直到满足停止条件：重复执行步骤2和3，不断更新聚类中心，并重新分配数据点，直到满足某个停止条件。常见的停止条件可以是达到最大迭代次数、聚类中心的变化量低于阈值，或者达到预先定义的误差限制。

5.输出最终的聚类结果：最终结果是K个聚类，每个聚类包含一组相似的数据对象

在本实验中，我们选用K-means算法对处理过的用户数据进行操作，可以将其分为不同类别的用户类型，方便我们后续对不同的用户做出不同的营销策略。

## 4 数据的预处理

首先，在我们的数据表中，总有一些数据不符合规范，比如空数据或者异常数据。如果保留这些数据势必会对我们的结果产生影响，因此在正式处理数据前，我们必须要进行预处理。

**缺失数据处理**

数据表有些关键信息，比如票价，如果为空的话，表明该用户未在本航空公司消费过，所以不适合将其加入到需要数据处理的集合中来，需要我们将其剔除。

airline\_data = pd.read\_csv("../data/air\_data.csv",  
 encoding="gb18030") *# 导入航空数据*print('原始数据的形状为：', airline\_data.shape)  
  
*# 去除票价为空的记录*exp1 = airline\_data["SUM\_YR\_1"].notnull()  
  
exp2 = airline\_data["SUM\_YR\_2"].notnull()  
  
exp = exp1 & exp2  
  
print('exp的形状是：', exp.shape)  
*# 选择所有为1的行，：表示选择所有列*airline\_notnull = airline\_data.loc[exp, :]  
  
print('删除缺失记录后数据的形状为：', airline\_notnull.shape)

在这段代码中首先读取了数据，然后选择第一年和第二年票价都不为空的数据。

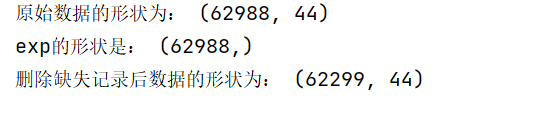


图4.4-1 删除空数据前后

根据运行结果我们可以看到，通过这个操作剔除了689行缺失数据。

**异常数据处理**

表格中往往有一些数据不符合常理，比如公里数为零和票价为零。那么将这样的数据加入到数据处理集中也不合适，同样也需要剔除。

index1 = airline\_notnull['SUM\_YR\_1'] != 0  
index2 = airline\_notnull['SUM\_YR\_2'] != 0  
index3 = (airline\_notnull['SEG\_KM\_SUM'] > 0) & \  
 (airline\_notnull['avg\_discount'] != 0)  
airline = airline\_notnull[(index1 | index2) & index3]  
  
print('删除异常记录后数据的形状为：', airline.shape)

在这段代码中，首先创建不同条件下的Series，然后按照我们的条件将索引值做运算，最后选出符合条件的数据集。

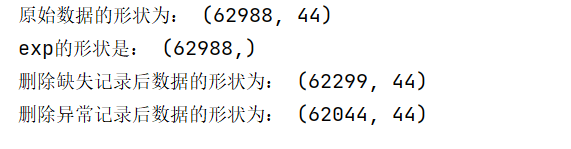


图4.4-2 删除异常数据前后

通过打印结果我们可以看到，共剔除了异常记录255条。

## 5 数据处理

由于我们本实验采用LRFMC模型来对该航空公司客户的数据进行分析，因此我们要构建符合LRFMC模型处理的数据集。

LRFMC包含了消费时间间隔R，客户关系长度L，消费频率F，飞行里程M和折扣系数的平均值C五个指标。在数据表中对应关系如下：

C对应avg\_discount，M对应SEG\_KM\_SUM，F对应FLIGHT\_COUNT，R对应LAST\_TO\_END，L对应FFF\_DATE和LOAD\_TIME的差值。

除L外，其他数据均可由表中直接得到。我们要由FFF\_DATE和LOAD\_TIME两个表项中的数据经过处理得到L。

*# 选取需求特征*airline\_selection = airline[["FFP\_DATE", "LOAD\_TIME",  
 "FLIGHT\_COUNT", "LAST\_TO\_END",  
 "avg\_discount", "SEG\_KM\_SUM"]]  
*# 构建L特征*L = pd.to\_datetime(airline\_selection["LOAD\_TIME"]) - \  
 pd.to\_datetime(airline\_selection["FFP\_DATE"])  
*# 转换为字符串类型*L = L.astype("str").str.split().str[0]  
*# 转换为月份*L = L.astype("int") / 30

在这段代码中，我们先将两个日期相减得到相差的天数，然后取出数字部分转化为Int类型后除以30得到月份。

现在所有的表项数据都有了，我们将其组装并打印。

*# 合并特征*airline\_features = pd.concat([L, airline\_selection.iloc[:, 2:]], axis=1) *# 第一个：选择所有的行切片，第二个2：为从2开始 axis为水平方向*print('构建的LRFMC特征前5行为：\n', airline\_features.head())  
*# 列索引（表项）变为字符串类型*airline\_features.columns = airline\_features.columns.astype(str)  
print(airline\_features)

组装好后我们将其进行标准化处理。标准化是一种常见的数据预处理方法，用于将数据转换为均值为0，标准差为1的标准正态分布。

from sklearn.preprocessing import StandardScaler  
  
data = StandardScaler().fit\_transform(airline\_features)  
  
np.savez('../tmp/airline\_scale.npz', data)  
  
print('标准化后LRFMC五个特征为：\n', data[:5, :]) *# 选择所有的列，切出前五行*

标准化结果：

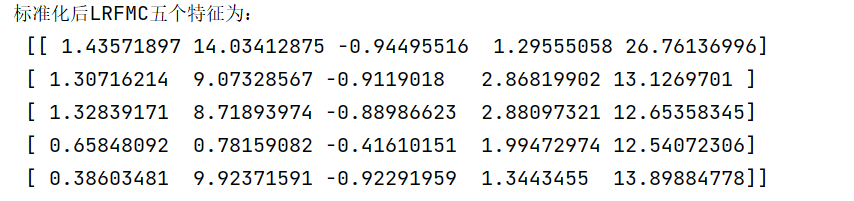


图4.5-1 标准化数据

### K-means聚合

我们已经拥有了标准化后的数据，我们将其用作K-means聚合的数据。

import numpy as np  
import pandas as pd  
  
from sklearn.cluster import KMeans *# 导入kmeans算法*airline\_scale = np.load('../tmp/airline\_scale.npz')['arr\_0']  
  
k = 5 *# 确定聚类中心数  
# 构建模型*kmeans\_model = KMeans(n\_clusters=k, random\_state=123, n\_init='auto')  
*# 运行次数，保证不会出现偶然  
# 确定不同聚合中心，选择最好的*fit\_kmeans = kmeans\_model.fit(airline\_scale) *# 模型训练*print(kmeans\_model.cluster\_centers\_) *# 查看聚类中心*print(kmeans\_model.labels\_) *# 查看样本的类别标签  
  
# 统计不同类别样本的数目*r1 = pd.Series(kmeans\_model.labels\_).value\_counts()  
  
print('最终每个类别的数目为：\n', r1)

在这段代码中，我们确定了聚类中心数为5，表示我们将数据分为5类。确定random\_state为123，表示我们多次进行运行，以免出现偶然情况。n\_init为auto，n\_init表示选择不同聚合中心的次数，并从中选出最合适的。运行后结果如下：

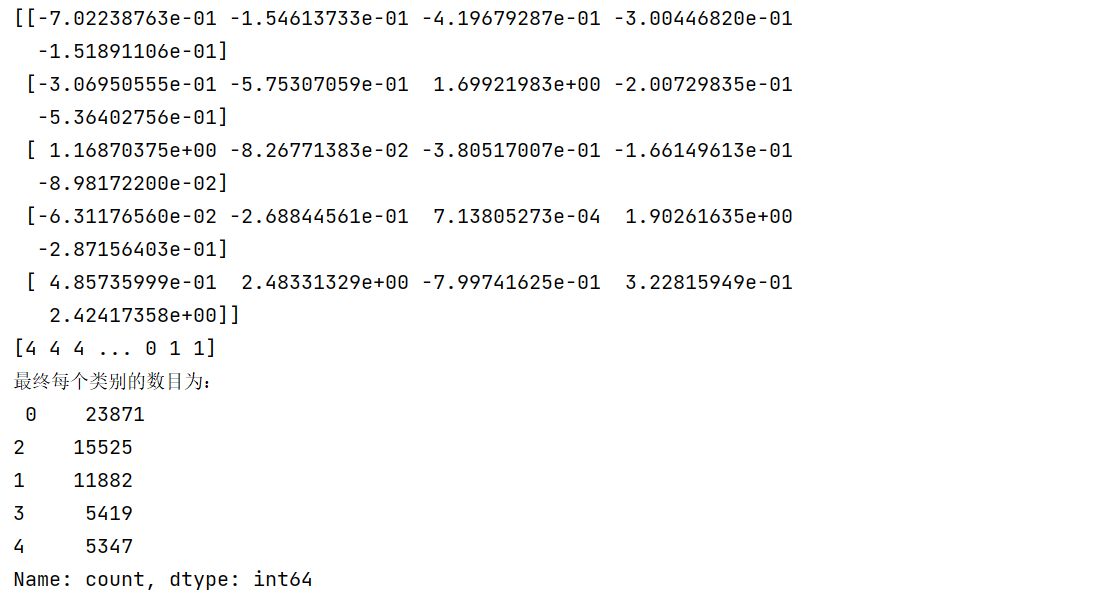


图4.5-2 聚合数据

从数据中我们可以看到聚合后，将数据分为了五类，其中每一类的数量分别为23871，15525，11882，5419，5347.

### 画雷达图

从上面的聚合中我们仅仅能得到每种类别中的数据量，却不能直观的看到每种类别的特点。雷达图是一种适合显示多个维度数据的图表类型，因此在具有多个参考因素的聚类分析中，使用雷达图可以更清晰地展示不同簇在各个维度上的特征差异。接下来我们就画出雷达图。

*# 画出雷达图*import matplotlib.pyplot as plt  
  
*# 聚类中心*cluster\_centers = kmeans\_model.cluster\_centers\_  
  
*# 创建雷达图*fig, ax = plt.subplots(figsize=(8, 6), subplot\_kw={'projection': 'polar'})  
  
*# 数据维度（列）数量*num\_dimensions = len(cluster\_centers[0])  
  
*# 分割一个圆形空间，并设置角度*angles = np.linspace(0, 2 \* np.pi, num\_dimensions, endpoint=False).tolist()  
angles += angles[:1] *# 闭合图形  
  
# 绘制雷达图*for i, center in enumerate(cluster\_centers):  
 values = np.concatenate((center, [center[0]])) *# 闭合图形* ax.plot(angles, values, label=f'Cluster {i+1}')  
 ax.fill(angles, values, alpha=0.25)  
  
*# 添加轴标签*ax.set\_xticks(angles[:-1])  
ax.set\_xticklabels(['Length', 'Frequency', 'Recency', 'Channel', 'Monetary Value'])  
  
*# 添加标题和图例*ax.set\_title('Radar Chart of Clusters')  
ax.legend()  
  
*# 显示雷达图*plt.show()

在这段代码中，首先获得聚类中心，然后创建图和子图，并确定图的映射方式为极坐标。然后根据聚类中心，获得雷达图的维度。根据维度，我们将整个圆划分为等值部分，并设置为首位连接。根据每个聚合中心的坐标和角度，依次沿每条边连线，为每个聚类中心设置标识label，画完边线后填充。最后为整幅图添加标签和标题并显示。

以下为效果图：

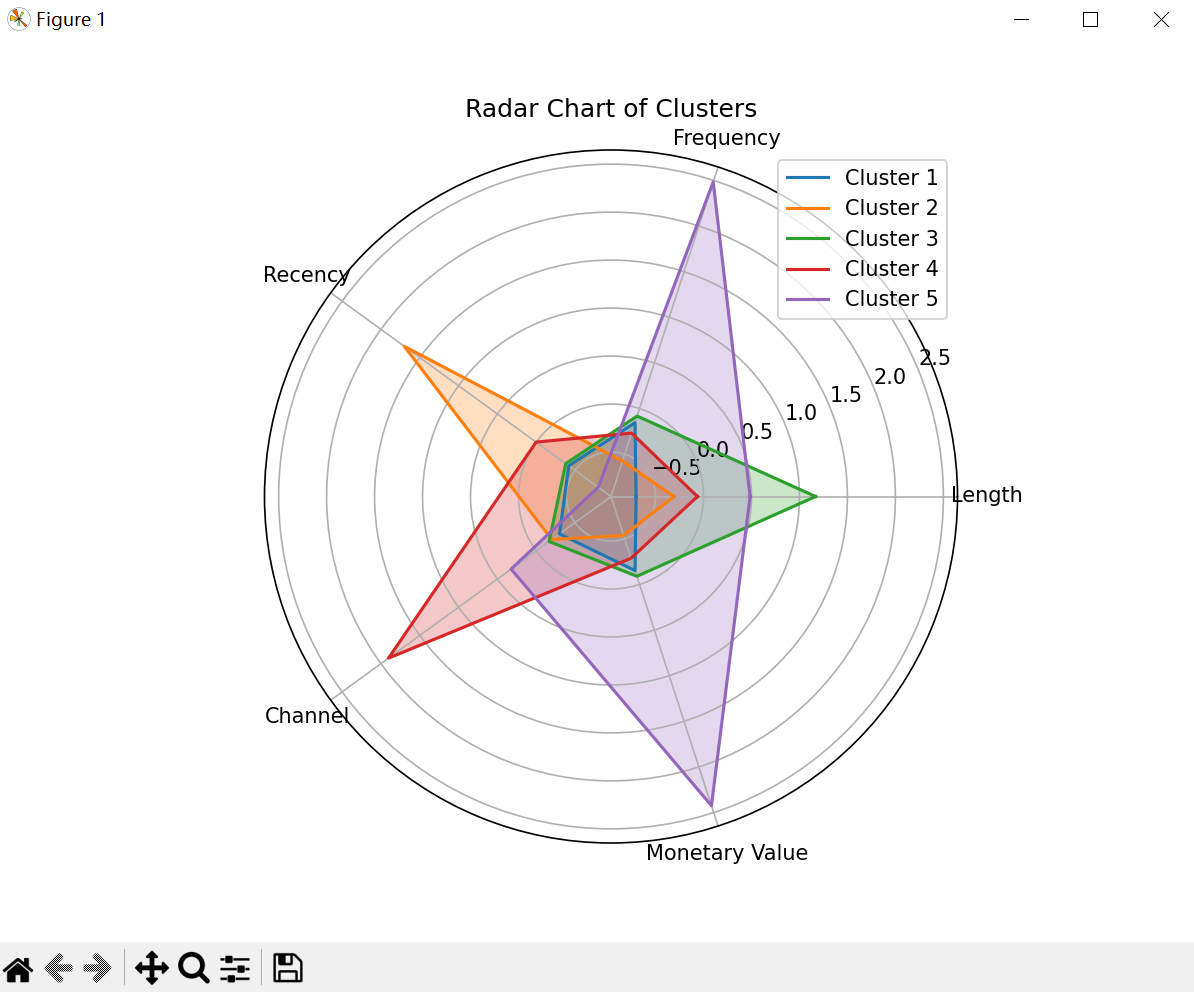


图4.5-3 k为5时的雷达图

### 聚类评价

在上述的聚类中，我们将类别分为了5类。当然这只是一种可能的分类方式，但是这种聚类方式效果如何我们并不知道。因此，我们接下来要对我们的聚类进行评价。

在老师给出的示例中，共提到了三种评价模式：calinski\_harabaz，silhouette\_score和FMI。前两种都是基于处理数据和聚合后的标签来进行分析评价，最后一种方式基于当前分类与聚合分类的相似形来进行分析评价。因此我们选择前两种方式来进行分析。

**Calinski\_harabaz：**

from sklearn.metrics import calinski\_harabasz\_score  
airline\_scale = np.load('../tmp/airline\_scale.npz')['arr\_0']

kmeans = KMeans(n\_clusters = 5,random\_state=123,n\_init="auto").fit(airline\_scale)  
score = calinski\_harabasz\_score(airline\_scale,kmeans.labels\_)  
print('数据聚%d类calinski\_harabaz指数为：%f'%(5,score))

运行结果：



图4.5-4 calinski\_harabaz评价k为5时的聚类结果

**Silhouette\_score：**

airline\_scale = np.load('../tmp/airline\_scale.npz')['arr\_0']  
silhouettteScore = []kmeans = KMeans(n\_clusters=5, random\_state=123, n\_init="auto").fit(airline\_scale)  
score = silhouette\_score(airline\_scale, kmeans.labels\_)  
silhouettteScore.append(score)  
print('数据聚%d类Silhouette\_score指数为：%f' % (5, score))

运行结果：



图4.5-5 Silhoutte\_score评价k为5时的聚类结果

仅仅有评价指标还不够，因为我们不知道其他情况下的指标为多少！因此我打算遍历一些可能的点数，从中选择合适的情况。

**Calinski\_harabaz：**

from sklearn.metrics import calinski\_harabasz\_score  
  
airline\_scale = np.load('../tmp/airline\_scale.npz')['arr\_0']  
for i in range(2,15):  
 *#构建并训练模型* kmeans = KMeans(n\_clusters = i,random\_state=123,n\_init="auto").fit(airline\_scale)  
 score = calinski\_harabasz\_score(airline\_scale,kmeans.labels\_)  
 print('数据聚%d类calinski\_harabaz指数为：%f'%(i,score))

运行结果：

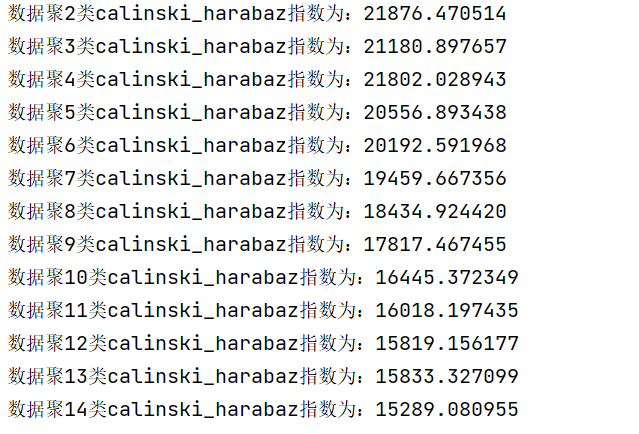


图4.5-6 不同k情况下calinski\_harabaz聚类结果评价

**Silhouette\_score：**

from sklearn.metrics import silhouette\_score  
import matplotlib.pyplot as plt  
  
airline\_scale = np.load('../tmp/airline\_scale.npz')['arr\_0']  
silhouettteScore = []  
for i in range(2, 15):  
 *#构建并训练模型* kmeans = KMeans(n\_clusters=i, random\_state=123, n\_init="auto").fit(airline\_scale)  
 score = silhouette\_score(airline\_scale, kmeans.labels\_)  
 silhouettteScore.append(score)  
 print('数据聚%d类Silhouette\_score指数为：%f' % (i, score))

运行结果：

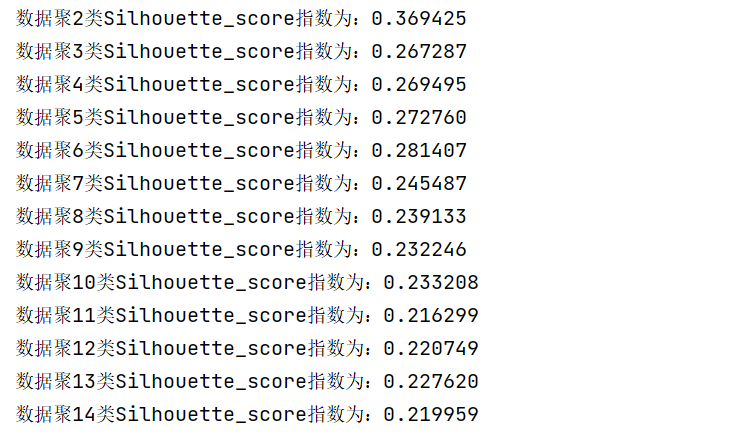


图4.5-7 不同k情况下Silhouette\_score聚类结果评价

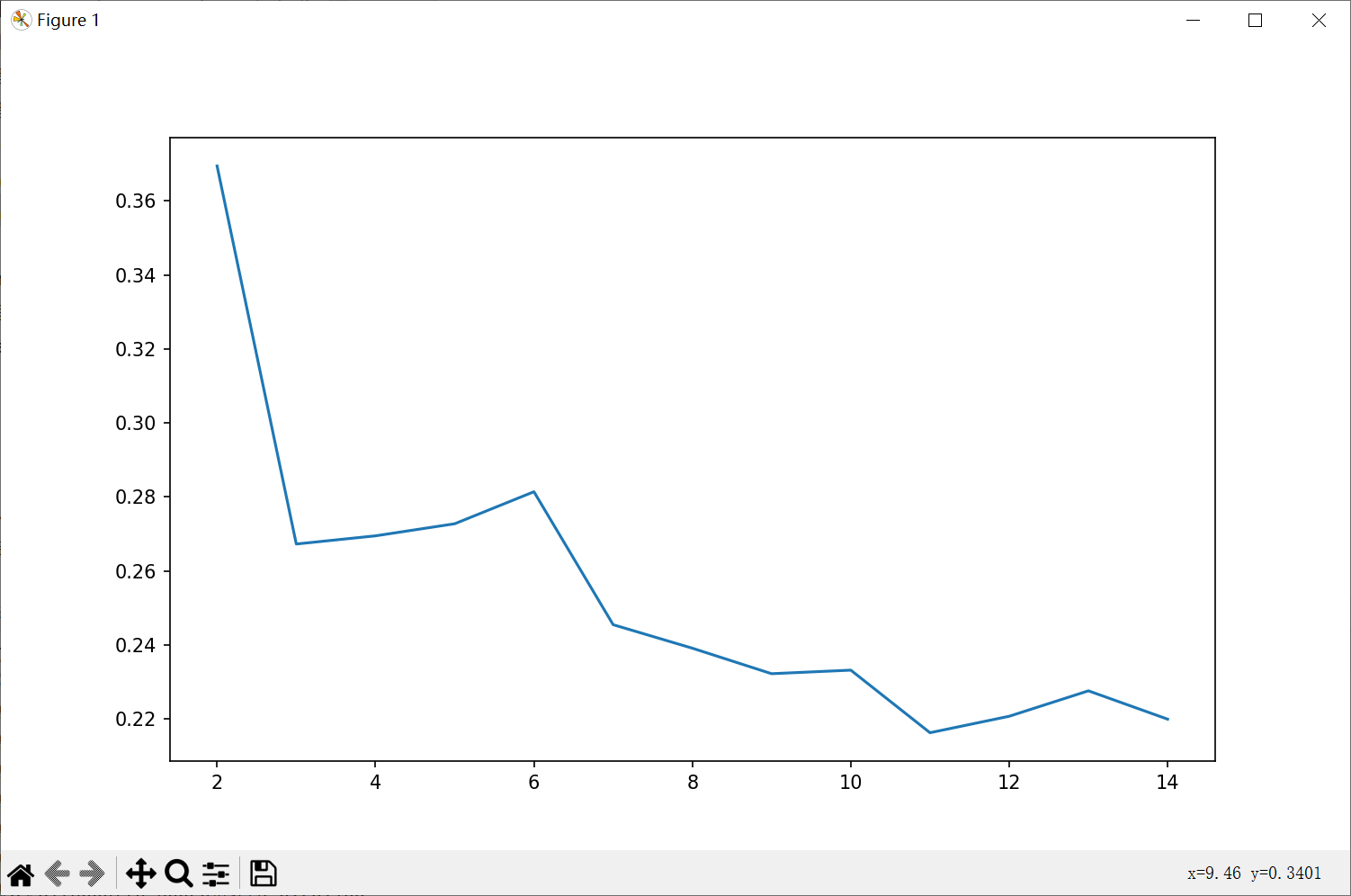


图4.5-8 不同k情况下Silhouette\_score聚类结果评价折线图

从两种评测方法中我们看到，当k为2时，聚合效果是最好的，因此我们重新用2进行聚合，并画出雷达图。

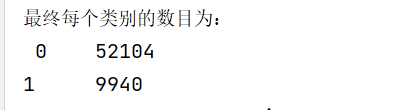


图4.5-9 k为2时的聚类结果

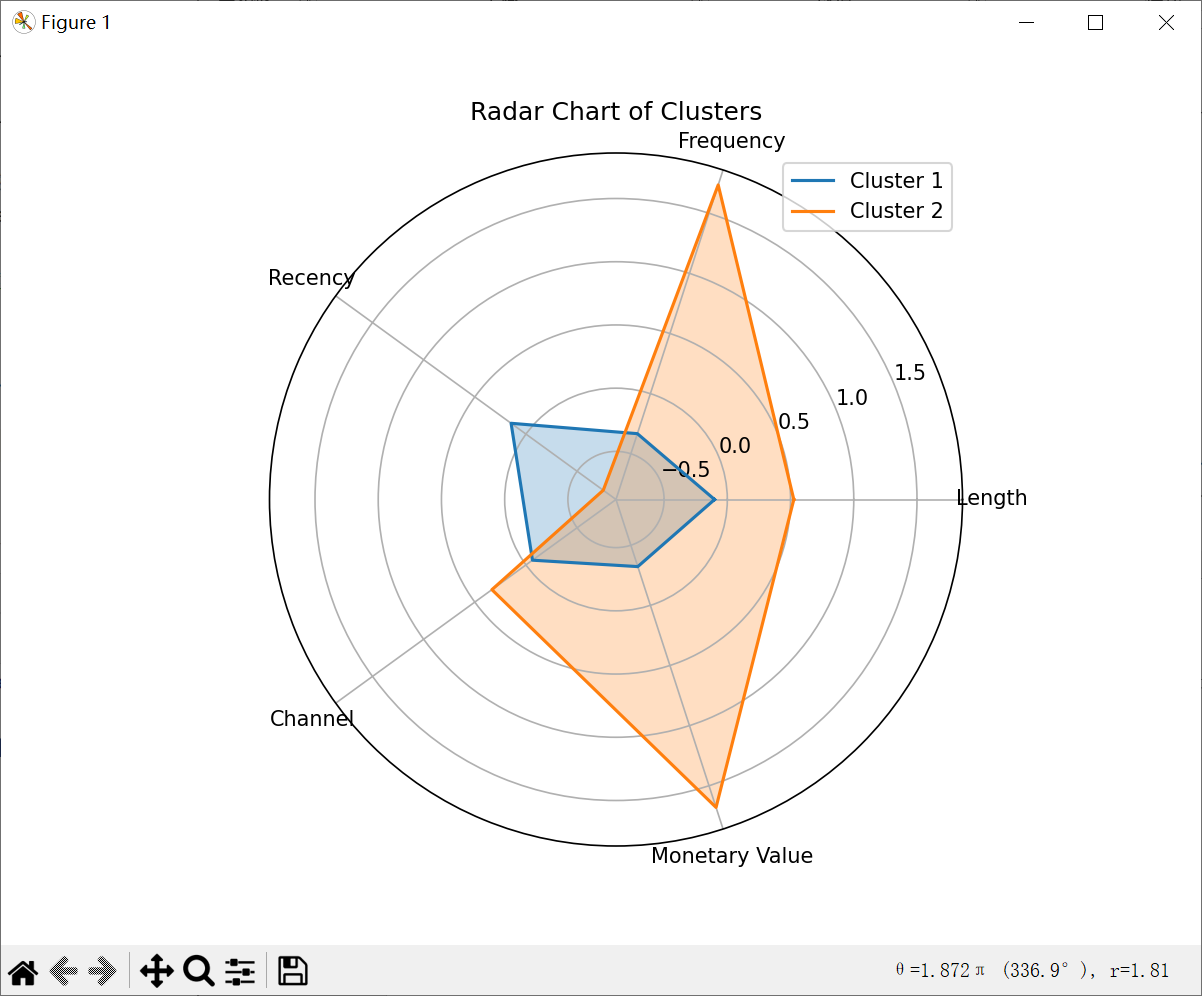


图4.5-10 k为2时的聚类结果雷达图

## 6 结果分析

K=5时，从雷达图中可以看到，有五类客户，分别为：

路程远频率高的高消费客户。对于这类客户我们要提高他们的用户体验，增强他们的满意度，将这些客户留在本航空公司内。

最近消费近但其余指标并不突出的客户。这类客户可能是最近刚注册本航空公司的用户，对于这类客户，我们可以提供注册新用户领取福利的政策。

折扣高但其余指标不突出的客户。这类客户可能是喜欢趁打折旅行的客户。对于这类客户，我们可以在一定程度上推出一些折扣，来刺激他们消费。

注册时间长，但其余指标不突出的客户。这类客户可能已经不再使用本公司产品，因此我们可以推出老用户回归福利来刺激他们重新回到本公司。

所有指标都不突出的客户。这类客户可能是经济情况比较拮据，也许我们可以推出满减的活动，既可以促进他们消费，又可以提高公司收入。

K=2时，从雷达图中可以看到，两类客户分别为：

频率高路程远且折扣入会时间久的客户。对于这类客户我们需要尽可能提高他们的体验感，将其留在本航空公司。

任何方面都不突出的普通客户。对于这类客户，我们可以推出一些福利活动来促进它们参与到本航空公司中。