Python 数据分析与数据挖掘（Python for Data Analysis&Data Mining）

Chap 10 客户聚类

内容：

- 客户价值识别模型
- 客户聚类建模
- 客户价值分析和客户画像（刻画每类客户的特征）
- 模型应用
- 算法：K-means聚类算法，朴素贝叶斯算法的性能对比
- 应用领域：客户价值分析，客户聚类，客户画像，客户关系管理，用户粘性，客户流失分析，等

实践：

- 数据探索、缺失处理、数据变换
- 数据归一化（z-score）
- 聚类算法（scikit K-means）

实例：

- 实例1：航空公司客户价值分析（Python数据分析与挖掘实战 第七章）
- 实例2：客户流失建模和分析

这节课是在前面数据分析的基础上，对无监督数据进行分析和聚类建模（K-means算法）。本节课通过某航空公司的客户数据构建客户关系管理分析，对客户进行聚类分析，勾画每类客户的特征，从而进行客户的价值分析，才能进一步针对每类客户进行有效的客户关系管理。虽然本节课是以航空公司客户的数据来进行实践分析，也适用于不同行业的多种用户类型数据。

准备工作：导入库，配置环境等

In [1]:
```
from __future__ import division
import os, sys

# 启动绘图
%matplotlib inline
import matplotlib.pyplot as plt

import pandas as pd
import numpy as np
```

实例1：航空公司客户价值分析

1. 问题背景

信息时代，企业的营销焦点从产品中心转为客户中心，客户关系管理成为企业的核心问题。获取新客户的成本很高（销售、市场、广告和人员工资等），然而大多数新客户产生的利润不如老客户多。客户关系管理的关键问题是客户分类，从而区分低价值客户和高价值客户，针对不同价值的客户制定优化的个性化服务方案，采取不同营销策略，将有限的营销资源集中于高价值客户，实现企业利润最大化的目标。准确的客户分类是企业优化营销资源分配的重要依据，客户分类是客户关系管理中的关键问题之一。

激烈的市场竞争环境，面临着旅客流失、竞争力下降和航空资源未充分利用等经营危机，通过建立合理的客户价值评估模型，对客户进行分群，分析比较不同客户群的客户价值，并制定相应的营销策略，对不同的客户群提供个性化的客户服务。

2. 原始数据

目前公司已经积累了大量的会员档案信息和会员乘坐航班记录，经加工后得到的部分客户的数据信息的属性说明表包含下面的信息：

- 客户基本信息：
  - 会员卡号，入会时间，性别，年龄，第一次飞行日期，会员卡级别，工作地城市，工作地所在省份，工作地所在国家
- 乘机信息：
  - 观测窗口内的飞行次数，观测窗口内的结束时间，最后一次乘机时间至观测窗口结束时长，评价折扣率，观测窗口的票价收入，观测窗口的总飞行公里数，末次飞行日期，平均乘机时间间隔，最大乘机间隔
- 积分信息：
  - 积分兑换次数，总精英积分，促销积分，合作伙伴积分，总累计积分，非乘机的积分变动次数，总基本积分

客户数据的属性说明情况见：data/客户信息属性说明.xls

客户的数据信息表：data/air_data.csv， 数据样本如下：

表7-2　航空信息数据表

MEMB-ER_NO	FFP_ DATE	FIRST_FLIGI	GENDE	FFP_ TIER	WORK_ CIT Y	WORK_ PROVIN	WORK	AGE	LOAD_ TIME	FLIGHT _COUNT	BP_ SUM
289047040	2013/03/16	2013/04/28	男	6			US	56	2014/03/31	14	147 158
289053451	2012/06/26	2013/05/16	男	6	乌鲁木齐	新疆	CN	50	2014/03/31	65	112 582
289022508	2009/12/08	2010/02/05	男	5		北京	CN	34	2014/03/31	33	77 475
289004181	2009/12/10	2010/10/19	男	4	S.P.S	CORTES	HN	45	2014/03/31	6	76 027
289026513	2011/08/25	2011/08/25	男	6	乌鲁木齐	新疆	CN	47	2014/03/31	22	70 142
289027500	2012/09/26	2013/06/01	男	5	北京	北京	CN	36	2014/03/31	26	63 498
289058898	2010/12/27	2010/12/27	男	4	ARCADIA	CA	US	35	2014/03/31	5	62 810
289037374	2009/10/21	2009/10/21	男	4	广州	广东	CN	34	2014/03/31	4	60 484
289036013	2010/04/15	2013/06/02	女	6	广州	广东	CN	54	2014/03/31	25	59 357

3. 挖掘目标

根据这些数据，实现以下目标：

- 1） 对客户进行分类。
- 2） 对不同的客户类别进行特征分析，比较不同类客户的客户价值。
- 3） 对不同价值的客户类别提供个性化服务，制定相应的营销策略。

4. 分析方法和过程

识别客户价值应用最广泛的模型是 **RFM** 模型，它通过下面3个指标来进行客户细分，识别出高价值的客户：

- 最近消费时间间隔（Recency）R
- 消费频率（Frequency）F
- 消费金额（Monetary）M

这个模型是否适合航空公司的客户价值识别？

- 消费金额跟航线长短和舱位等级等多种因素有关，同样消费金额的不同旅客对于航空公司的价值是不同的。
- 考虑：一位购买长航线、低等级舱位的旅客 vs 一位购买短航线、高等级舱位的旅客，后者的价值可能更高

消费金额指标不适合，怎么解决不足？

1.) 一定时间内积累的飞行里程 M
2.) 一定时间内乘坐舱位所对应的折扣系数的平均值 C

使用上面两个指标 M 和 C 来代替消费金额 M

此外考虑客户的入会时间长短也在一定程度上能够影响客户价值，所以，再增加一个指标：

- 客户关系长度L

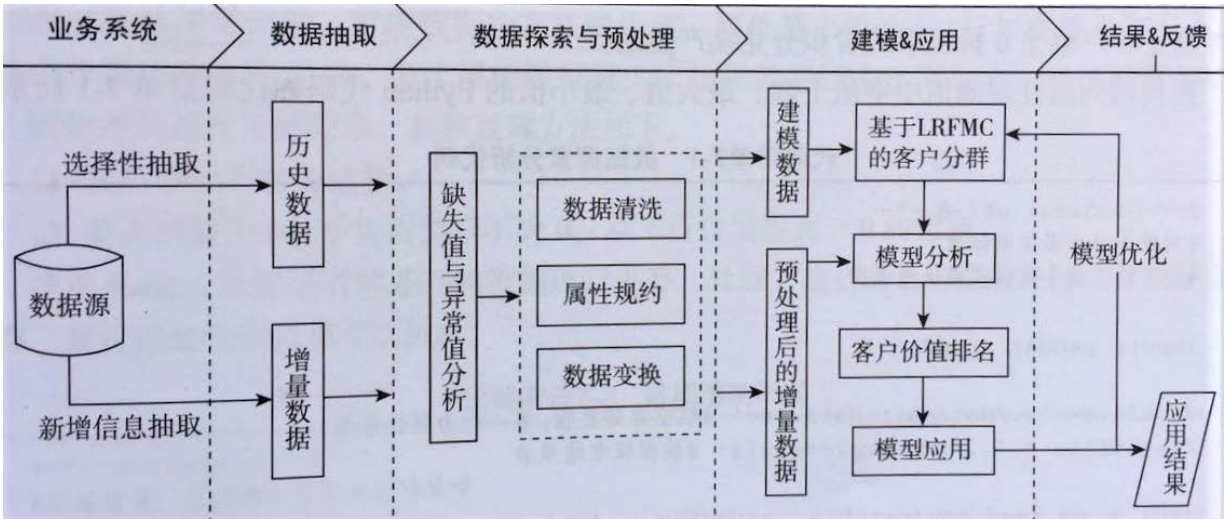
这样，对于本实例，识别客户价值的模型为 **LRFMC** 模型，包含下面5个指标：

1. 客户关系长度 L：会员入会时间距离观测窗口结束的月数
2. 消费时间间隔 R： 客户最近一次乘机距离观测窗口结束的月数
3. 消费频率 F： 客户在观测窗口内乘坐公司飞机的次数
4. 飞行里程 M： 客户在观测窗口内累计的飞行里程
5. 折扣系数的平均值 C： 客户在观测窗口内乘坐舱位所对应的折扣系数的平均值

虽然可以采用传统的数据分析的属性分箱方法对客户价值进行分析，但是细分的客户群太多，提高了针对性营销的成本。 属性分箱方法：根据属性的平均值进行划分，大于平均值表示为+，小于平均值的表示为-，然后针对客户的各个属性值的不同组合进行识别。例如：

1. （M+， R+， F+）： 重要价值客户
2. （M+， R+， F-）： 重要发展客户
3. （M+， R-， F+）： 重要保持客户
4. （M+， R-， F-）： 重要挽留客户
5. ...

本实例通过LRFMC模型的5个指标，采用K-means聚类算法，识别出最有价值客户。本实例进行航空客户价值分析的总体流程图如下：



本案例的航空客户价值的数据分析和挖掘主要包括以下步骤：

1. 从数据源中进行选择性抽取与新增数据抽取，分别形成历史数据和增量数据
2. 对步骤1)中形成的两个数据集进行数据探索分析与预处理，包括数据缺失值与异常值的探索分析，数据的属性规约、清洗和变换
3. 利用步骤2)中形成的已完成数据预处理的建模数据，基于客户直接分析的LRFMC模型进行客户分群，对各个客户群进行特征分析，识别出有价值的客户
4. 针对模型结果得到不同价值的客户，采用不同的营销手段，提供定制化的服务。

1. 数据抽取

以2014-03-31为结束时间，选取观测窗口为两年，抽取观测窗口内有乘机记录的所有客户的详细数据形成历史数据。对于后续新增的客户详细信息，以后续新增数据中最新的时间点作为结束时间，采用同样的方法进行抽取，形成增量数据。

从航空公司系统内的客户基本信息、乘机信息以及积分信息等详细数据中，根据末次飞行日期（LAST_FLIGHT_DATE），抽取2012-04-01至2014-03-31内所有乘客的详细数据，总共有 62988 条记录，包含了44个属性，如：会员卡号、入会时间、性别、年龄、会员卡级别、工作地城市、工作地所在省份、工作地所在国家、观测窗口结束时间、观测窗口乘机积分、飞行公里数、飞行次数、飞行时间、乘机时间间隔和平均折扣率等属性。

2. 数据探索分析

本案例中的数据探索分析对数据进行缺失值分析和异常值分析，分析数据的规律以及异常值。

通过对数据观察发现原始数据中存在票价为空值，票价最小值为0、折扣率最小值为0、总飞行公里数大于0的记录。

票价为空值的数据可能是客户不存在乘机记录造成，其他的数据可能是客户乘坐0折机票或者积分兑换产生的。

下面的代码查找每列属性观测值中空值个数、最大值、最小值。

In [2]:
```
#-*- coding: utf-8 -*-
#对数据进行基本的探索
# 返回缺失值个数以及最大最小值

import pandas as pd
```

```
datafile= 'data/air_data.csv' #航空原始数据,第一行为属性标签
resultfile = 'data/explore.xls' #数据探索结果表

#读取原始数据，指定UTF-8编码（需要用文本编辑器将数据装换为UTF-8编码）
data = pd.read_csv(datafile, encoding = 'utf-8')
data.head()
```

Out[2]:

	MEMBER_NO	FFP_DATE	FIRST_FLIGHT_DATE	GENDER	FFP_TIER	WORK_CITY	WO
0	54993	2006/11/02	2008/12/24	男	6	.	北京
1	28065	2007/02/19	2007/08/03	男	6	NaN	北京
2	55106	2007/02/01	2007/08/30	男	6	.	北京
3	21189	2008/08/22	2008/08/23	男	5	Los Angeles	CA
4	39546	2009/04/10	2009/04/15	男	6	贵阳	贵州

5 rows × 44 columns

In [3]:
```
#包括对数据的基本描述，percentiles参数是指定计算多少的分位数表（如1/4分位数、中位数等）；
#T是转置，转置后更方便查阅
explore = data.describe(percentiles = [0.25,0.75], include = 'all').T # 增加0.25和0.75，默认有50%
explore.head()
```

Out[3]:

	count	unique	top	freq	mean	std	min	25%
MEMBER_NO	62988	NaN	NaN	NaN	31494.5	18183.2	1	15747.8
FFP_DATE	62988	3068	2011/01/13	184	NaN	NaN	NaN	NaN
FIRST_FLIGHT_DATE	62988	3406	2013/02/16	96	NaN	NaN	NaN	NaN
GENDER	62985	2	男	48134	NaN	NaN	NaN	NaN
FFP_TIER	62988	NaN	NaN	NaN	4.10216	0.373856	4	4

In [4]:
```
#describe()函数中没有空值个数，需要手动计算空值数
explore['null'] = len(data) - explore[:]['count']
explore['null'].head()
```

Out[4]:
```
MEMBER_NO          0
FFP_DATE           0
FIRST_FLIGHT_DATE  0
GENDER             3
FFP_TIER           0
Name: null, dtype: object
```

In [5]:
```
explore = explore[['null', 'max', 'min']]
explore.columns = [u'空值数', u'最大值', u'最小值'] #表头重命名
'''这里只选取部分探索结果。
describe()函数自动计算的字段有count（非空值数）、unique（唯一值数）、
top（频数最高者）、freq（最高频数）、mean（平均值）、std（方差）、
min（最小值）、50%（中位数）、max（最大值）'''
# 前面增加了0.25和0.75四分位数
explore.head()
```

Out[5]:

	空值数	最大值	最小值

MEMBER_NO	0	62988	1
FFP_DATE	0	NaN	NaN
FIRST_FLIGHT_DATE	0	NaN	NaN
GENDER	3	NaN	NaN
FFP_TIER	0	6	4

In [6]:
```
explore.to_excel(resultfile) #导出结果
```

In [7]:
```
# 发现票价有空值，最小值为0
print explore.ix['SUM_YR_1'] # 第一年总票价
print explore.ix['SUM_YR_2'] # 第二年总票价
```

```
空值数      551
最大值    239560
最小值         0
Name: SUM_YR_1, dtype: object
空值数      138
最大值    234188
最小值         0
Name: SUM_YR_2, dtype: object
```

In [8]:
```
# 平均折扣率最小值为0
print explore.ix['avg_discount']
```

```
空值数      0
最大值    1.5
最小值      0
Name: avg_discount, dtype: object
```

In [9]:
```
# 总飞行公里数大于0
print explore.ix['SEG_KM_SUM']
```

```
空值数         0
最大值    580717
最小值       368
Name: SEG_KM_SUM, dtype: object
```

3. 数据预处理

数据预处理包括：数据清理、属性归一化和数据变换。

3-1. 数据清理：丢弃缺失值

前面数据的探索发现数据有缺失值，票价最小值为0，折扣率最小值为0，总飞行公里数大于0的记录。考虑这类数据所占的比例较小，对于问题影响不大，因此对其进行丢弃处理，具体办法如下：

- 丢弃票价为空的记录
- 丢弃票价为0、平均折扣率不为0、总飞行公里数大于0的记录

使用pandas对满足清洗条件的一行数据全部丢弃。

In [10]:
```
#-*- coding: utf-8 -*-
# 数据清洗，过滤掉不符合规则的数据

import pandas as pd
```

```
datafile= 'data/air_data.csv' #航空原始数据,第一行为属性标签
cleanedfile = 'data/air_data_cleaned.csv' #数据清洗后保存的文件

#读取原始数据，指定UTF-8编码（需要用文本编辑器将数据装换为UTF-8编码）
data = pd.read_csv(datafile,encoding='utf-8')
print u"原始数据条数：  ", len(data)

data = data[data['SUM_YR_1'].notnull() & data['SUM_YR_2'].notnull()] #两年的票价都为非空值才保留

#只保留票价非零的，或者平均折扣率与总飞行公里数同时为0的记录。
index1 = data['SUM_YR_1'] != 0
index2 = data['SUM_YR_2'] != 0
index3 = (data['SEG_KM_SUM'] == 0) & (data['avg_discount'] == 0) #该规则是“与”
data = data[index1 | index2 | index3] #该规则是“或”

data.to_csv(cleanedfile,encoding='utf-8') #导出结果
print u"清洗后数据条数：  ", len(data)
```

```
原始数据条数：  62988
清洗后数据条数：  62044
```

In [11]:
```
data.head()
```

Out[11]:

	MEMBER_NO	FFP_DATE	FIRST_FLIGHT_DATE	GENDER	FFP_TIER	WORK_CITY	WO
0	54993	2006/11/02	2008/12/24	男	6	.	北京
1	28065	2007/02/19	2007/08/03	男	6	NaN	北京
2	55106	2007/02/01	2007/08/30	男	6	.	北京
3	21189	2008/08/22	2008/08/23	男	5	Los Angeles	CA
4	39546	2009/04/10	2009/04/15	男	6	贵阳	贵州

5 rows × 44 columns

3-2. 数据属性选择

原始数据中属性太多，根据航空公司客户价值的 LRFMC 模型，删除与其不相关、弱相关或冗余的属性，例如，会员卡号、性别、工作地城市、工作地所在省份、工作地所在国家和年龄等属性，最终 选择与 LRFMC 指标相关的6个属性：

1. FFP_DATE - 办理会员卡的开始时间
2. LOAD_TIME - 观测窗口的结束时间 （2014/3/31）
3. FLIGHT_COUNT - 飞行次数（频次）
4. avg_discount - 平均折扣率
5. SEG_KM_SUM - 观测窗口总飞行公里数
6. LAST_TO_END - 最后一次乘机时间至观察窗口末端时长

In [12]:
```
#-*- coding: utf-8 -*-
#数据属性选择，过滤掉不符合规则的属性

import pandas as pd

datafile= 'data/air_data.csv' #航空原始数据,第一行为属性标签
cleanedfile = 'data/air_data_cleaned.csv' #数据清洗后保存的文件
```

```
#读取原始数据，指定UTF-8编码（需要用文本编辑器将数据装换为UTF-8编码）
data = pd.read_csv(datafile,encoding='utf-8')
print u"原始数据条数：  ", len(data)

data = data[data['SUM_YR_1'].notnull() & data['SUM_YR_2'].notnull()] #票价非空值才保留

#只保留票价非零的，或者平均折扣率与总飞行公里数同时为0的记录。
index1 = data['SUM_YR_1'] != 0
index2 = data['SUM_YR_2'] != 0
index3 = (data['SEG_KM_SUM'] == 0) & (data['avg_discount'] == 0) #该规则是“与”
data = data[index1 | index2 | index3] #该规则是“或”

# 原始数据中属性太多，选择与LRFMC指标相关的6个属性
data = data[['FFP_DATE','LAST_TO_END','FLIGHT_COUNT','SEG_KM_SUM','LOAD_TIME','avg_discount']]

data.to_csv(cleanedfile,encoding='utf-8',index=False) #导出结果
print u"清洗后数据条数：  ", len(data)
data.head()
```

```
原始数据条数：  62988
清洗后数据条数：  62044
```

Out[12]:

	FFP_DATE	LAST_TO_END	FLIGHT_COUNT	SEG_KM_SUM	LOAD_TIME	avg_discoun
0	2006/11/02	1	210	580717	2014/03/31	0.961639
1	2007/02/19	7	140	293678	2014/03/31	1.252314
2	2007/02/01	11	135	283712	2014/03/31	1.254676
3	2008/08/22	97	23	281336	2014/03/31	1.090870
4	2009/04/10	5	152	309928	2014/03/31	0.970658

3-3. 数据变换

数据变换将数据转换为 “适当的” 格式来适应挖掘任务和算法的要求。本实例中主要采用的数据变换方式包含：

1. 属性构造（利用已有的属性来提取五个指标特征）
2. 数据归一化（标准化）

首先，原始数据中并没有直接给出 LRFMC 模型的这5个指标，需要通过原始数据进行属性构造，以提取出这5个指标，具体计算方式如下：

1. L = 观测窗口的结束时间 - 入会时间（单位：月）， LOAD_TIME - FFP_DATE
2. R = 最后一次乘机时间到观测窗口的时长（单位：月）， LAST_TO_END
3. F = 观测窗口内的飞行次数（单位：次）， FLIGHT_COUNT
4. M = 观测窗口的总飞行公里数 （单位：公里）， SEG_KM_SUM
5. C = 观测窗口内乘坐舱位对应的折扣系数的平均值，即平均折扣率， avg_discount

提出这5个指标（特征）后，观察分析这5个特征，发现其数据的取值范围数据差异较大，如下图：

	L	R	F	M	C
min	11.992033	1.0	2.0	368.0	0.136017
25%	24.148340	29.0	3.0	4874.0	0.613085
50%	41.988542	105.0	7.0	10200.0	0.712162
75%	71.689357	260.0	15.0	21522.5	0.809293
max	112.922237	731.0	213.0	580717.0	1.500000

为了消除数量级数据带来的影响，需要对数据进行标准化处理（standardization，也叫归一化处理normalization）。

数据归一化处理的方法，见PPT

In [13]:
```
#-*- coding: utf-8 -*-
#数据属性选择，过滤掉不符合规则的属性

import pandas as pd
import numpy as np

datafile= 'data/air_data_cleaned.csv' #航空原始数据,第一行为属性标签
normdfile = 'data/air_data_normd.csv' #数据归一化后保存的文件

#读取原始数据，指定UTF-8编码（需要用文本编辑器将数据装换为UTF-8编码）
data = pd.read_csv(datafile,encoding='utf-8')

data[u'LOAD_TIME'] = pd.to_datetime(data[u'LOAD_TIME'])
data[u'FFP_DATE'] = pd.to_datetime(data[u'FFP_DATE'])
#将时间差数据转换为月为单位，默认是天days
data['L'] = (data[u'LOAD_TIME'] - data[u'FFP_DATE']) / np.timedelta64(1, 'M')
data['R'] = data['LAST_TO_END']
data['F'] = data['FLIGHT_COUNT']
data['M'] = data['SEG_KM_SUM']
data['C'] = data['avg_discount']

#原始数据中属性太多，选择与LRFMC指标相关的5个属性进行标准化处理
data = data[['L','R','F','M','C']]

#z-score标准化处理
data = (data - data.mean(axis = 0)) / (data.std(axis = 0))

data.columns = ['Z'+i for i in data.columns] #表头重命名
data.to_csv(normdfile,encoding='utf-8',index=False) #导出结果
data.head()
```

Out[13]:

	ZL	ZR	ZF	ZM	ZC
0	1.435707	-0.944948	14.034016	26.761154	1.295540
1	1.307152	-0.911894	9.073213	13.126864	2.868176
2	1.328381	-0.889859	8.718869	12.653481	2.880950
3	0.658476	-0.416098	0.781585	12.540622	1.994714
4	0.386032	-0.922912	9.923636	13.898736	1.344335

4. 模型构建

客户价值分析模型构建包括两个部分：

1. 根据航空公司客户价值分析LRFMC模型的5个指标，对客户进行聚类分群
2. 结合业务对每个客户群进行特征分析，分析每类客户群的价值，并进行排名

4-1. 客户聚类（K-means算法）

采用K-means聚类算法对客户数据进行客户分群，算法原理见PPT，K=聚类的类数，通常由用户自己指定该参数的值。

结合对本案例业务的理解与分析，确定客户的类别数量为5类客户群，因此采用K-means算法对客户数据，聚成5类。

In [14]:
```
#-*- coding: utf-8 -*-
#K-Means聚类算法

import pandas as pd
from sklearn.cluster import KMeans #导入K均值聚类算法

inputfile = 'data/air_data_normd.csv' #待聚类的数据文件
k = 5                  #需要进行的聚类类别数

#读取数据并进行聚类分析
data = pd.read_csv(inputfile) #读取数据

#调用k-means算法，进行聚类分析
kmodel = KMeans(n_clusters = k, n_jobs = 8) #n_jobs是并行数，一般等于CPU数较好
kmodel.fit(data) #训练模型
```

Out[14]:
```
KMeans(algorithm='auto', copy_x=True, init='k-means++', max_iter=300,
    n_clusters=5, n_init=10, n_jobs=8, precompute_distances='auto',
    random_state=None, tol=0.0001, verbose=0)
```

In [15]:
```
kmodel.cluster_centers_ #查看聚类中心
```

Out[15]:
```
array([[ 1.16066672, -0.37722119, -0.08691852, -0.09484404, -0.1559046 ],
       [ 0.05184279, -0.00266813, -0.22680311, -0.23125407,  2.19134701],
       [ 0.48332845, -0.79938326,  2.4832016 ,  2.42472391,  0.30863003],
       [-0.31367829,  1.68625847, -0.57401599, -0.53682019, -0.1733261 ],
       [-0.70020646, -0.41488827, -0.16114258, -0.16095751, -0.25513154]])
```

In [16]:
```
kmodel.labels_ #查看各样本对应的类别
```

Out[16]:
```
array([2, 2, 2, ..., 4, 3, 3])
```

查看聚类算法的结果

结果1如下图：

	ZL	ZR	ZF	ZM	ZC	聚类个数
Class						
0	-0.700357	-0.417193	-0.157817	-0.156808	-0.272233	24349
1	0.485141	-0.799940	2.483600	2.424669	0.314843	5338
2	-0.310240	1.692406	-0.574687	-0.536611	-0.188374	12000
3	1.163842	-0.378411	-0.085278	-0.092930	-0.159965	15653
4	-0.006685	0.009080	-0.251630	-0.261650	2.064710	4704

结果2如下图：

	ZL	ZR	ZF	ZM	ZC	聚类个数
Class						
0	1.160572	-0.377246	-0.086948	-0.094826	-0.155740	15740
1	-0.313508	1.686181	-0.573985	-0.536821	-0.172953	12125
2	0.052631	-0.002343	-0.226362	-0.230845	2.194638	4184
3	0.483328	-0.799383	2.483202	2.424724	0.308630	5336
4	-0.700165	-0.414931	-0.161205	-0.161035	-0.254707	24659

结果3如下图：

	ZL	ZR	ZF	ZM	ZC	聚类个数
Class						
0	-0.700453	-0.417101	-0.157887	-0.156862	-0.272342	24349
1	-0.006241	0.008583	-0.251298	-0.261383	2.064652	4704
2	1.163843	-0.378501	-0.085243	-0.092891	-0.160022	15652
3	0.485141	-0.799940	2.483600	2.424669	0.314843	5338
4	-0.310097	1.692360	-0.574681	-0.536618	-0.188052	12001

结果4如下图：

	ZL	ZR	ZF	ZM	ZC	聚类个数
Class						
0	0.052997	-0.002302	-0.226085	-0.230527	2.195419	4181
1	-0.700196	-0.414876	-0.161240	-0.161077	-0.254597	24661
2	-0.313508	1.686181	-0.573985	-0.536821	-0.172953	12125
3	0.483328	-0.799383	2.483202	2.424724	0.308630	5336
4	1.160548	-0.377266	-0.086983	-0.094862	-0.155659	15741

由于算法原理和算法的精度问题，重复实验得到的聚类类标和聚类中心也可能略有变化！

In [17]:
```
data['Class'] = kmodel.labels_
data.head()
```

Out[17]:

	ZL	ZR	ZF	ZM	ZC	Class
0	1.435707	-0.944948	14.034016	26.761154	1.295540	2
1	1.307152	-0.911894	9.073213	13.126864	2.868176	2
2	1.328381	-0.889859	8.718869	12.653481	2.880950	2
3	0.658476	-0.416098	0.781585	12.540622	1.994714	2
4	0.386032	-0.922912	9.923636	13.898736	1.344335	2

In [18]:
```
data.groupby(['Class']).count() # 每个cluster的统计个数
```

Out[18]:

	ZL	ZR	ZF	ZM	ZC
Class					
0	15740	15740	15740	15740	15740
1	4184	4184	4184	4184	4184
2	5336	5336	5336	5336	5336
3	12125	12125	12125	12125	12125
4	24659	24659	24659	24659	24659

In [19]:
```
#每个cluster的均值，即cluster的中心，kmodel.cluster_centers_
m = data.groupby(['Class']).mean()
c = data.groupby(['Class']).count()
m[u'聚类个数'] = c['ZL']
m
```

Out[19]:

	ZL	ZR	ZF	ZM	ZC	聚类个数
Class						
0	1.160572	-0.377246	-0.086948	-0.094826	-0.155740	15740
1	0.052631	-0.002343	-0.226362	-0.230845	2.194638	4184
2	0.483328	-0.799383	2.483202	2.424724	0.308630	5336
3	-0.313508	1.686181	-0.573985	-0.536821	-0.172953	12125
4	-0.700165	-0.414931	-0.161205	-0.161035	-0.254707	24659

In [20]:
```
# 由于算法的精度问题，重复实验得到的聚类中心也可能略有变化
kmodel.cluster_centers_[0]
```

Out[20]:
```
array([ 1.16066672, -0.37722119, -0.08691852, -0.09484404, -0.1559046 ])
```

In [21]:
```
#data.groupby(['Class']).describe()
```

4-2. 客户价值分析

针对聚类结果进行特征分析。结合前面定义的LRFMC的定义，分析数据，并刻画每类客户的特征，区分重要保持客户，重要发展客户，重要挽留客户，以及一般客户和低价值客户

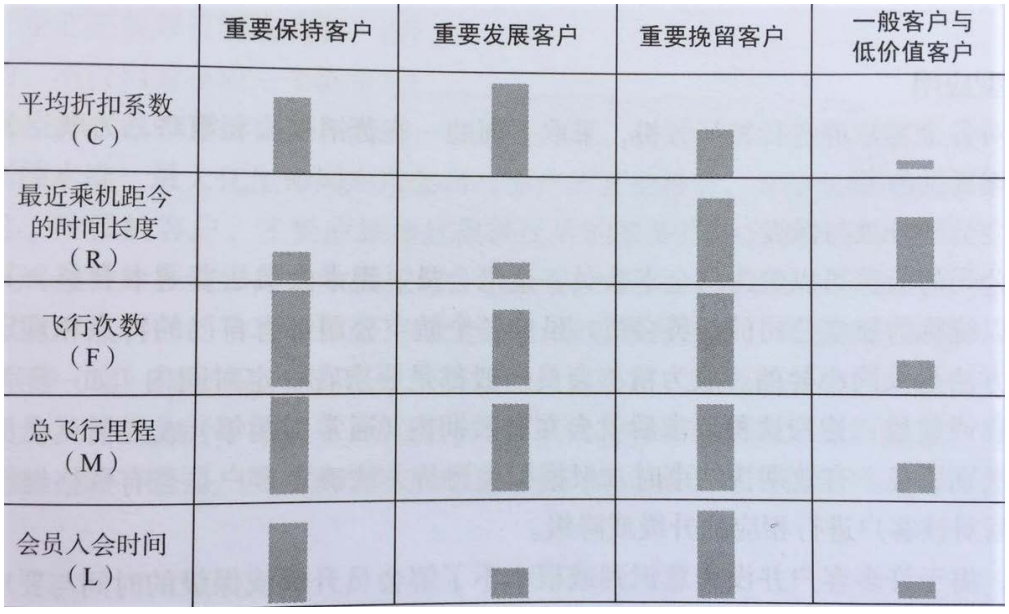
回顾5个指标：

1. L = 入会时间
2. R = 最近乘机时间
3. F = 飞行频次
4. M = 总飞行公里数
5. C = 机票平均折扣率

观察聚类结果：

1. 5336类客户群：在F（飞行频次）和M（飞行公里数）属性上最大，在R（最近乘机）属性上最小，平均折扣率C（航班的舱位等级）较高；说明，经常飞行，飞行公里数累计很大，飞行活跃，乘坐航班的舱位等级较高
2. 15740类客户群：在L（入会时间）属性上最大，即入会时间很久
3. 12125类客户群： 在R（最近乘机）属性上最大（好久没有乘机），在F（飞行频次）和M（飞行公里数）属性上最小（飞行不频繁，少出行）
4. 24659类客户群： 在L（入会时间）和C（平均折扣率）属性上最小，刚入会
5. 4148类客户群：在C（平均折扣率）属性上最大（所乘航班的舱位等级较高）

然后，结合业务分析，通过比较各个指标在群间的大小对某一个群的特征进行评价分析，可见每个客户群都有显著不同的表现特征，基于该特征描述，本案例定义了5个等级的客户类别：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。他们之间的区别见下图：



其中每类客户的特征如下：

1. 重要保持客户 ： 这类客户在F（飞行频次）和 M（飞行公里数）属性上最大，在R（最近乘机）属性上最小，平均折扣率C（航班的舱位等级）较高；说明 ==> 经常飞行，飞行公

里数累计很大，飞行活跃，乘坐航班的舱位等级较高。他们是航空公司的高价值客户，是最为理想的客户类型，对航空公司的贡献最大，所占比例却较小 （5K vs 62K）。因此，航空公司应该优先资源投放在他们身上，对他们进行差异化管理和一对一营销，提高这类客户的忠诚度与满意度，尽可能延长这类客户的高水平消费。

2. 重要发展客户 ： 这类客户在C（平均折扣率）属性上最大（一般所乘航班的舱位等级较高），最近乘机R属性较低，但是乘坐次数F和里程数M较低，入会时间L较短。他们是航空公司的潜在价值客户，虽然这类客户的当前价值并不是很高，但却有很大的发展潜力。航空公司要努力促使这类客户增加在本公司的乘机消费，通过客户价值提升，加强这类客户的满意度，提高他们转向竞争对手的转移成本，使他们逐渐成为本公司的忠诚客户。这类客户比例较少（4K vs 62K）
3. 重要挽留客户 ： 这类客户入会时间很久，所乘航班的折扣率C、乘坐次数F 或 里程M 较高，但是较长时间已经没有乘坐本公司的航班（R高）。他们的客户价值变化的不确定性很高，由于这些客户衰退的原因各不相同，所以掌握客户的最新信息、维持与客户的互动就尤为重要。航空公司应该根据这些客户的最近消费时间、消费次数的变化情况，推测客户消费的异动状况，并列出客户名单，对其重点联系，采取一定的营销手段，延长客户的生命周期。这类客户比例（15K vs 62K）
4. 一般与低价值客户 ： 这类客户所乘航班的平均折扣率C很低，较长时间没有乘坐过本公司航班R高，乘坐的次数F 或 里程M 较低，入会时长L短。他们是航空公司的一般用户与低价值客户，可能是在航空公司机票打折促销时，才会乘坐本公司航班。一般客户（24K vs 62K），低价值客户（12K vs 62K）

其中，重要发展客户、重要保持客户、重要挽留客户这三类重要客户分别可以归入客户生命周期管理的发展期、稳定期、衰退期三个阶段。

根据每种客户类型的特征，对各类客户群进行客户价值排名。针对不同类型的客户群提供不同的产品和服务，提升重要发展客户的价值、稳定和延长重要保持客户的高水平消费、防范重要挽留客户的流失并积极进行关系恢复。

本模型采用历史数据进行建模，随着时间的变化，分析数据的观测窗口也在变换。因此，对于新增客户详细信息，考虑业务的实际情况，该模型建议每一个月运行一次，对其新增客户信息通过聚类中心进行判断，同时对本次新增客户的特征进行分析。如果增量数据的实际情况与判断结果差异大，需要业务部门重点关注，查看变化大的原因以及确认模型的稳定性。如果模型稳定性变化大，需要重新训练模型进行调整。目前模型进行重新训练的时间没有统一标准，大部分情况都是根据经验来决定。经验建议：每隔半年训练一次模型比较合适。

4-3. 模型应用

根据对各个客户群进行特征分析，采取一些营销手段和策略，为航空公司的价值客户群管理提供参考。

（1） 会员的升级与保级

（2） 首次兑换

（3） 交叉营销

客户识别期和发展期为客户关系打下基石，但是这两个时期带来的客户关系是短暂的，不稳定的。企业要获得长期的利润，必须具有稳定的，高质量的客户。保持客户对于企业是至关重要的。（1）争取一个新客户的成本远远高于维持老客户的成本；（2）客户流失会造成公司收益的直接损失。

实例**2**：客户流失建模和分析

客户关系管理中客户流失问题对公司往往造成巨大的损害。客户流失对例如增长造成的负面影响非

常大，尤其客户与公司的关系越长久，对公司造成的利润就会越高。流失一个客户比获得一个新客户对公司的损失更大。如何改善流失问题，继而提高客户满意度、忠诚度是公司维护自身市场并面对激烈竞争的一件大事，客户流失分析将成为帮助公司开展持续改进活动的指南。

客户流失建模和分析怎么做？

1. 挖掘目标： 针对目前老客户进行分类预测
2. 客户定义：针对公司客户信息数据，进行老客户以及客户类型的定义
   A. 老客户：飞行次数大于6次的客户
   B. 流失客户： 第二年飞行次数是第一年飞行次数比例小于50%
   C. 准流失客户： 第二年飞行次数是第一年飞行次数比例在 [50%, 90%]内的客户
   D. 未流失客户： 第二年飞行次数是第一年飞行次数比例大于 [90%]的客户
3. 数据属性选择： 需要选取客户信息中的关键属性，如
   A. 会员卡级别、客户类型（流失、准流失、未流失）、平均乘机时间间隔、平均折扣率、积分兑换次数、非乘机积分综合、单位里程票价和单位里程积分等
4. 构建客户的流失模型
   A. 随机选取数据的80%做训练样本，剩余的20%作为测试样本
   B. 选择分类算法（使用过决策树，朴素贝叶斯，k-最近邻，支持向量机等分类算法）
   C. 运用模型预测未来客户的类别归属（未流失、准流失或已流失）

In [ ]: