航空公司客户价值分析

项目分析过程如下

1、了解航空公司现状与客户价值分析

一、航空公司现状：

（1）行业内竞争：

民航的竞争除了三大航空公司之间的竞争之外，还将加入新崛起的各类小型航空公司、民营航空公司，甚至国外航空巨头。航空产品生产过剩，产品同质化特征愈加明显，于是航空公司从价格、服务间的竞争逐渐转向对客户的竞争。

（2）行业外竞争：

随着高铁、动车等铁路运输的兴建，航空公司受到巨大冲击。

二、客户价值分析

（1）航空公司的数据分析：

目前航空公司已积累了大量的会员档案信息和其乘坐航班记录。以2014-03-31为结束时间，选取宽度为两年的时间段作为分析观测窗口，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，44个特征，总共62988条记录。数据如air_data.csv所示！

（2）航空公司的营销实施经验分析：

1、公司收入的80%来自顶端的20%的客户。

2、20%的客户其利润率100%。

3、90%以上的收入来自现有客户。

4、大部分的营销预算经常被用在非现有客户上。

5、5%至30%的客户在客户金字塔中具有升级潜力。

6、客户金字塔中客户升级2%，意味着销售收入增加10%，利润增加50%。

注意：这些经验也许并不完全准确，但是它揭示了新时代客户分化的趋势，也说明了对客户价值分析的迫切性和必要性。

2、预处理航空客户数据

（1）通过观察，航空公司客户原始数据存在少量的缺失值和异常值，需要清洗后才能用于分析。具体表现在：

1、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。

处理方法：丢弃票价为空的记录。

```
#首先，导入项目所需要的库
import pandas as pd #pandas库用于文件操作
from sklearn.preprocessing import StandardScaler #用于对数据的标准化
import matplotlib.pyplot as plt #用于绘制图像可视化
import numpy as np #用于对数据的运算
from sklearn.cluster import KMeans  #sklearn封装的KMeans算法库
```

```
def load_data():
    """
    加载数据
    :return: air_data
    """
    air_data = pd.read_csv('../dates/air_data.csv', encoding='ansi')

    return air_data
```

```
 #丢弃票价为空的记录。
 msk1 = pd.notnull(air_data['SUM_YR_1'])  # 如果有值，则为True,如果为空，则为False
 msk2 = pd.notnull(air_data['SUM_YR_2'])  # 如果有值，则为True,如果为空，则为False

 # 都为True ---置为True   只要有一个False --->False
 msk = msk1 & msk2

 # 筛选数据
airline_notnull = air_data.loc[msk,:]
print('删除缺失记录后数据的形状为：',airline_notnull.shape)
```

2、其他的数据可能是客户乘坐0折机票或者积分兑换造成。由于原始数据量大，这类数据所占比例较小，对于问题影响不大，因此对其进行丢弃处理。

处理方法：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。

```
#只保留票价非零的，或者平均折扣率不为0且总飞行公里数大于0的记录。
# b、丢弃票价为 0、平均折扣率不为 0、总飞行千米数大于 0 的记录。
# --保留对航空公司有价值的数据：  票价 > 0 ,同时折扣 > 0 同时   飞行里程 > 0
msk3 = air_data['SUM_YR_1'] > 0
msk4 = air_data['SUM_YR_2'] > 0
# 2种思考方式：  两个票价必须都大于0，票价才大于0 ；  只要有一个票价大于0 ,那么票价就大于0

# 折扣大于0
msk5 = air_data['avg_discount'] > 0

# 飞行里程 > 0
msk6 = air_data['SEG_KM_SUM'] > 0
msk = (msk3 | msk4) & msk5 & msk6
# 筛选数据
air_data = air_data.loc[msk, :]
print('删除异常记录后数据的形状为：',air_data.shape)
```

（2）构建航空客户价值分析的关键特征

1、**RFM**模型介绍：

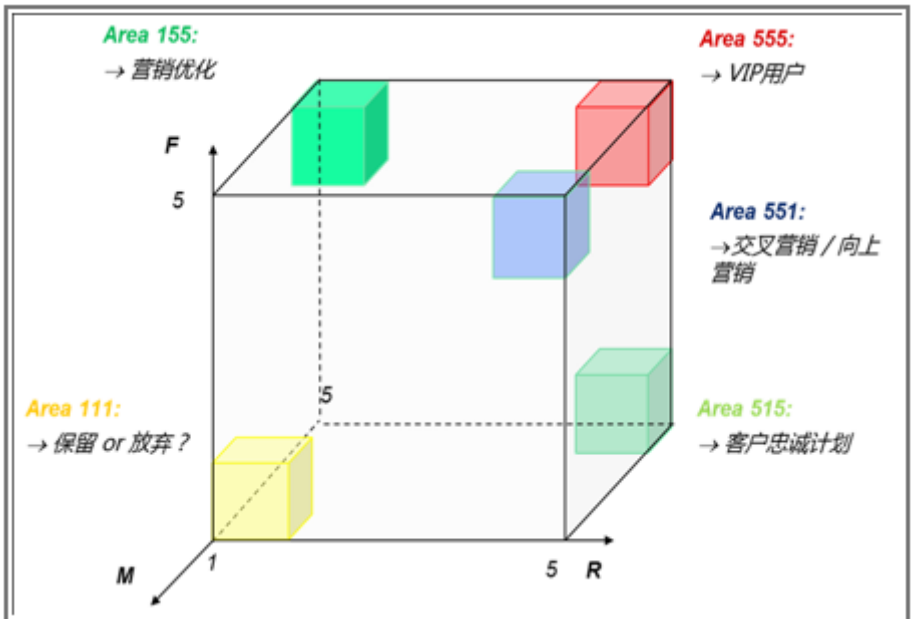
本项目的目标是客户价值分析，即通过航空公司客户数据识别不同价值的客户，识别客户价值应用最广泛的模型是RFM模型。

一、R（Recency）指的是最近一次消费时间与截止时间的间隔。通常情况下，最近一次消费时间与截止时间的间隔越短，对即时提供的商品或是服务也最有可能感兴趣。

二、F（Frequency）指顾客在某段时间内所消费的次数。可以说消费频率越高的顾客，也是满意度越高的顾客，其忠诚度也就越高，顾客价值也就越大。

三、M（Monetary）指顾客在某段时间内所消费的金额。消费金额越大的顾客，他们的消费能力自然也就越大，这就是所谓“20%的顾客贡献了80%的销售额”的二八法则。

2、**RFM**模型结果解读：



RFM模型包括三个特征，使用三维坐标系进行展示，如图所示。X轴表示Recency，Y轴表示Frequency，Z轴表示Monetary，每个轴一般会分成5级表示程度，1为最小，5为最大。

3、航空客户价值分析的**LRFMC**模型：

本项目选择客户在一定时间内累积的飞行里程M和客户在一定时间内乘坐舱位所对应的折扣系数的平均值C两个特征代替消费金额。此外，航空公司会员入会时间的长短在一定程度上能够影响客户价值，所以在模型中增加客户关系长度L，作为区分客户的另一特征。

本项目将客户关系长度L，消费时间间隔R，消费频率F，飞行里程M和折扣系数的平均值C作为航空公司识别客户价值的关键特征（如图所示），记为LRFMC模型。

模型	L	R	F	M	C
航空公司LRFMC模型	会员入会时间距观测窗口结束的月数	客户最近一次乘坐公司飞机距观测窗口结束的月数	客户在观测窗口内乘坐公司飞机的次数	客户在观测窗口内累计的飞行里程	客户在观测窗口内乘坐舱位所对应的折扣系数的平均值

```
# 选取需求特征
airline_selection = air_data[["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"])
L = L.astype("str").str.split().str[0]
L = L.astype("int")/30
## 合并特征
airline_features = pd.concat([L,
    airline_selection.iloc[:,2:]],axis = 1)
print('构建的LRFMC特征前5行为：\n',airline_features.head())
```

```
# 标准化后LRFMC五个特征
from sklearn.preprocessing import StandardScaler
data = StandardScaler().fit_transform(airline_features)
np.savez('../dates/airline_scale.npz',data) #保存处理好的数据
print('标准化后LRFMC五个特征为：\n',data[:5,:])
```

3、使用K-Means算法进行客户分群

```
airline_scale = np.load('../dates/airline_scale.npz')['arr_0']
k = 5 ## 确定聚类中心数
#构建模型
kmeans_model = KMeans(n_clusters = k,n_jobs=4,random_state=123)
fit_kmeans = kmeans_model.fit(airline_scale)   #模型训练
centers = kmeans_model.cluster_centers_
print("聚类中心",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)
```

4、数据可视化（利用雷达图，从不同的特征来描述数据对象）

```
def show_res(centers):
    """
    绘制雷达图，来展示结果
    :param centers: 各个类别的聚类中心
    :return: None
    """
    # 1、创建画布
    plt.figure()

    # 支持中文，支持负号：
    plt.rcParams['font.sans-serif'] = 'SimHei'
    plt.rcParams['axes.unicode_minus'] = False

    # 2、绘图及修饰
    # 绘制雷达图
    datalength = centers.shape[1]
    # 构建角度----从0-2π生成5个元素的等差数组
    angle = np.linspace(0, 2 * np.pi, datalength, endpoint=False)
    # 闭合角度
    angle = np.concatenate((angle, [angle[0]]), axis=0)
    print('angle:\n', angle)

    # 闭合数据
    centers = np.concatenate((centers, centers[:, 0:1]), axis=1)

    # 绘制雷达图
    for i in range(centers.shape[0]):
        plt.polar(angle, centers[i, :])
```

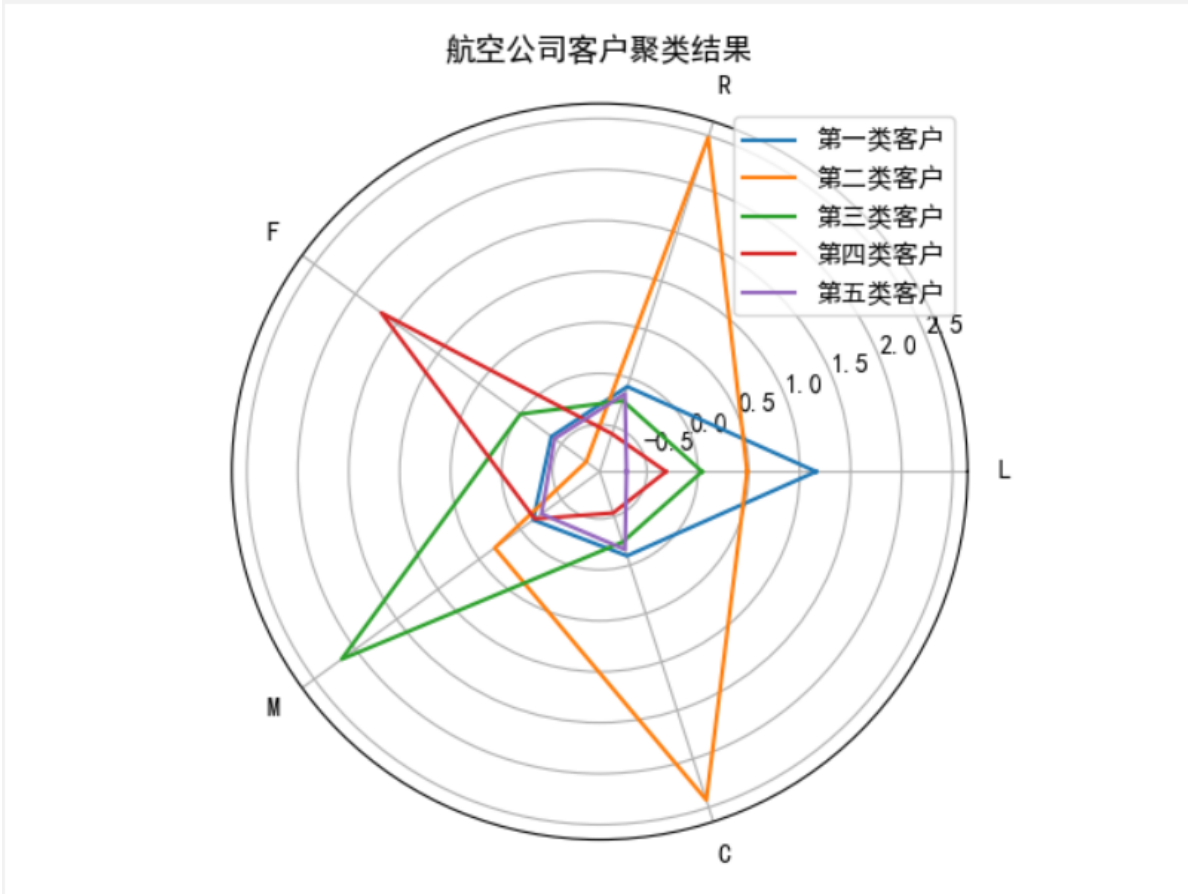
```
    # 添加标题
    plt.title('航空公司客户聚类结果')

    # 修改刻度
    plt.xticks(angle[:-1], ['L', 'R', 'F', 'M', 'C'])

    # 添加图例
    plt.legend(['第一类客户', '第二类客户', '第三类客户', '第四类客户', '第五类客户'],
loc=0)

    # 保存
    # plt.savefig('./data/航空公司客户聚类结果.png')
    # 3、保存及展示
    plt.show()
```

```
#数据可视化
show_res(centers)
```



预测结果所示：

```
最终每个类别的数目为：
4    24611
0    15730
3    12111
1     5337
2     4255
```

5、分析结果

6、完整代码

```
# -*- codeing = utf-8 -*-
# @Project :中共教育实训
# @Date :2021/7/13,18:59
# @Author :田智龙
# @File :答辩航空公司分析项目
# @Software:中共教育实训
# -*- coding: utf-8 -*-

import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.cluster import KMeans

def load_data():
    """
    加载数据
    :return: air_data
    """
    air_data = pd.read_csv('../dates/air_data.csv', encoding='ansi')
    return air_data

def dell_data(air_data):
    # 丢弃票价为空的记录。
    msk1 = pd.notnull(air_data['SUM_YR_1'])  # 如果有值，则为True,如果为空，则为False
    msk2 = pd.notnull(air_data['SUM_YR_2'])  # 如果有值，则为True,如果为空，则为False

    # 都为True ---置为True    只要有一个False --->False
    msk = msk1 & msk2

    # 筛选数据


    airline_notnull = air_data.loc[msk, :]
    print('删除缺失记录后数据的形状为：', airline_notnull.shape)
    # 只保留票价非零的，或者平均折扣率不为0且总飞行公里数大于0的记录。
    # b、丢弃票价为 0、平均折扣率不为 0、总飞行千米数大于 0 的记录。
    # --保留对航空公司有价值的数据：  票价 > 0 ,同时折扣 > 0 同时   飞行里程 > 0
    msk3 = air_data['SUM_YR_1'] > 0
    msk4 = air_data['SUM_YR_2'] > 0
    # 2种思考方式：  两个票价必须都大于0，票价才大于0 ；  只要有一个票价大于0 ,那么票价就大于0

    # 折扣大于0
    msk5 = air_data['avg_discount'] > 0

    # 飞行里程 > 0
    msk6 = air_data['SEG_KM_SUM'] > 0
    msk = (msk3 | msk4) & msk5 & msk6
    # 筛选数据
    air_data = air_data.loc[msk, :]
    print('删除异常记录后数据的形状为：', air_data.shape)
```

```
    # 选取需求特征
    airline_selection = air_data[["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"])
    L = L.astype("str").str.split().str[0]
    L = L.astype("int") / 30
    ## 合并特征
    airline_features = pd.concat([L,
                                  airline_selection.iloc[:, 2:]], axis=1)
    print('构建的LRFMC特征前5行为：\n', airline_features.head())

    # 标准化后LRFMC五个特征
    from sklearn.preprocessing import StandardScaler
    data = StandardScaler().fit_transform(airline_features)
    # np.savez('../dates/airline_scale.npz', data)  # 保存处理好的数据
    print('标准化后LRFMC五个特征为：\n', data[:5, :])

    return air_data

def show_res(centers):
    """
    绘制雷达图，来展示结果
    :param centers: 各个类别的聚类中心
    :return: None
    """
    # 1、创建画布
    plt.figure()

    # 支持中文，支持负号：
    plt.rcParams['font.sans-serif'] = 'SimHei'
    plt.rcParams['axes.unicode_minus'] = False

    # 2、绘图及修饰
    # 绘制雷达图
    datalength = centers.shape[1]
    # 构建角度----从0-2π生成5个元素的等差数组
    angle = np.linspace(0, 2 * np.pi, datalength, endpoint=False)
    # 闭合角度
    angle = np.concatenate((angle, [angle[0]]), axis=0)
    print('angle:\n', angle)

    # 闭合数据
    centers = np.concatenate((centers, centers[:, 0:1]), axis=1)

    # 绘制雷达图
    for i in range(centers.shape[0]):
        plt.polar(angle, centers[i, :])

    # 添加标题
    plt.title('航空公司客户聚类结果')

    # 修改刻度
    plt.xticks(angle[:-1], ['L', 'R', 'F', 'M', 'C'])

    # 添加图例
```

```
    plt.legend(['第一类客户', '第二类客户', '第三类客户', '第四类客户', '第五类客户'],
loc=0)

    # 保存
    # plt.savefig('./data/航空公司客户聚类结果.png')
    # 3、保存及展示
    plt.show()

def main():
    #加载数据
    air_data = load_data()
    #处理数据
    dell_data(air_data)
    k = 5  ## 确定聚类中心数
    # 构建模型
    kmeans_model = KMeans(n_clusters=k, n_jobs=4, random_state=123)
    fit_kmeans = kmeans_model.fit(air_data)  # 模型训练
    centers = kmeans_model.cluster_centers_
    print("聚类中心", 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)

    #数据可视化
    show_res(centers)

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