综合实验设计说明书

（2018）

Python 结课论文

年　　级　2016

组　　号　第 1 组

组　　长　1610121144 马骏超

指导教师　许涛

完成时间　2018 年 12 月 24 日

目录

1. 数据集摘要

名称	BuddyMove Data Set Data Set
特征简介	Multivariate,Text,Real
记录数	249
分析目标	到南印度的旅行者目的地分布情况；南印度的旅游目的地最受欢迎的情况；南印度的各种旅游目的地的发展情况等。

2. 项目简介

（1）数据来源

这个数据集来自 holidayiq.com 的 249 位评论者发布的目的地评论，直到 2014 年 10 月。考虑了南印度各个目的地中的 6 个类别的评论，并且每个评论者（旅行者）的每个类别的评论数量都被捕获。

（2）属性

- 属性一：唯一用户 ID
- 属性二：体育场馆等的评论数量
- 属性三：宗教机构等的评论数量
- 属性四：自然景观等的评论数量
- 属性五：剧院展览等的评论数量
- 属性六：商场购物等的评论数量
- 属性七：公园野餐等的评论数量

（3）数据量

数据量：249

（4）基本统计特征

基本统计特征：多变量，文本，真实

（5）分析目标

1.到南印度的旅行者目的地分布情况

2.南印度的旅游目的地最受欢迎的情况

3.南印度的各种旅游目的地的发展情况

等

（6）分析手段

1.对于分析目标 1，将数据集中所有的六个场景的评论数逐类相加，将六个类别的数据总和用柱状图和饼状图表示，每个场景的总评论数就显示出来了。

2.对于分析目标 2，数据集中每个人都对上述六个场景进行了评论，我们取个人评论数量最多的那个场景加一分，遍历 249 份数据，六个场景的分数用柱状图和饼状图表示，最受欢迎的场景就显示出来了。

3.对于分析目标 3，将数据集中每个场景的每个人的评论数从头至尾取出，每个场景都做一个折线图（由于每个场景都有 249 份数据，所以每 10 条数据取一个平均值，最后 9 条取平均值，共取得 25 条数据），增长趋势一目了然。

（7）结论简述

1.去往南印度的旅行者对于“宗教机构”“自然景观”“剧院展览”“商场购物”和“公园野餐”的兴趣充分，而对“体育场馆”的兴趣很低。

2.去往南印度的旅行者最感兴趣的是当地的“自然景观”，最无感的是当地的“体育场馆”。

3.南印度的旅游业发展势头持续高涨，六处场地的留言数都在高速增长。

3．数据读取与处理

（1）数据处理环境

编程语言：Python3.7

编译器：PyCharm + numpy（插件） + pandas （插件）+ matplotlib（插件）

（2）数据处理过程

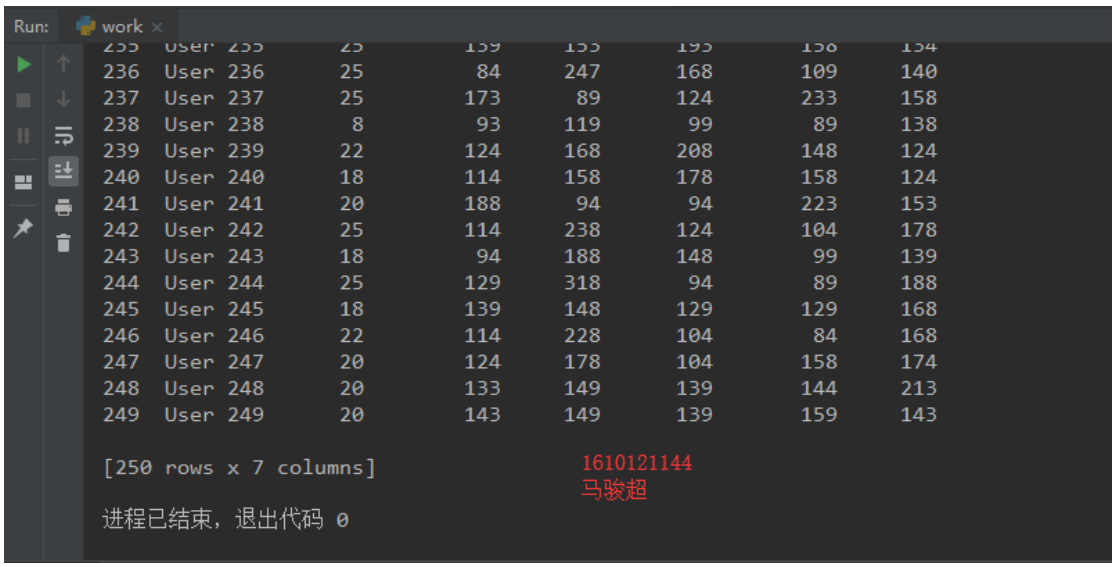
1.读取数据

```
'''
    Description: Python期末结课作业
    Create by JetBrains PyCharm
    Author:马骏超
    Time:2018/12/15 11:00                    1610121144
'''                                          马骏超
# -*- coding: utf-8 -*-

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

plt.rcParams['font.sans-serif'] = 'SimHei'
plt.rcParams['axes.unicode_minus'] = False

data = pd.read_csv('buddymove_holidayiq.csv', sep=',', header=None, encoding='utf-8')
print(data)

#print(data.shape)#数据集格式(250,7)
```

数据集文件为”buddymove_holidayiq.csv”

读出结果如下图

```
Run:  work
D:\JetbrainProjects\PycharmProjects\PythonWork\venv\Scripts\python.exe D:/Jetbrai
            0         1          2         3         4         5       6
0     User Id    Sports  Religious  Nature  Theatre  Shopping  Picnic
1      User 1         2         77      79        69        68      95
2      User 2         2         62      76        76        69      68
3      User 3         2         50      97        87        50      75
4      User 4         2         68      77        95        76      61   1610121144
5      User 5         2         98      54        59        95      86   马骏超
6      User 6         3         52     109        93        52      76
7      User 7         3         64      85        82        73      69
8      User 8         3         54     107        92        54      76
9      User 9         3         64     108        64        54      93
10    User 10         3         86      76        74        74     103
11    User 11         3        107      54        64       103      94
12    User 12         3        103      60        63       102      93
13    User 13         3         64      82        82        75      69
14    User 14         3         93      54        74       103      69
15    User 15         3         63      82        81        78      69
16    User 16         3         82      79        75        75      82
17    User 17         5         59     131       103        54      86
```



2.数据预处理

因为数据较少，数据集只有 250*7，所以通过手动查看数据即可，发现并无非法数据，此数据集的数据预处理结束。

若数据集中的数据量较多，可使用 data.isnull()函数判断数据是否为空来进行数据的预处理。

3.分类

通过对数据集的读取可以看出，此数据集本身就是分类数据集，数据通过 7 个属性“User ID”“Sports”“Religious”“Natur”“Theatre”“Shopp”和“Picnic”来分类存储。

4.聚类

只举一例（代码中多次使用）。

对于分析目标 1（到南印度的旅行者目的地分布情况），将上述后六个属性的各自 249 条数据相加，得到 6 条六个属性的总和数据，用柱状图和饼状图表示。

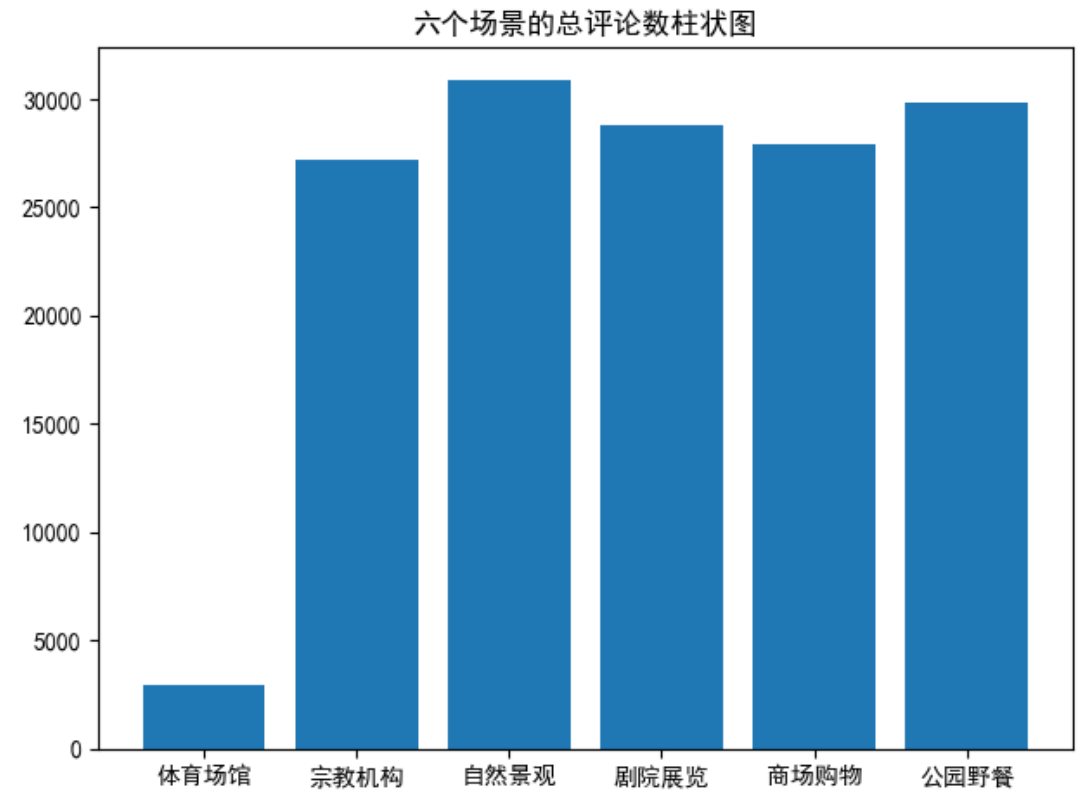
```python
##1.1 六个场景的总评论数柱状图

sports = religious = nature = theatre = shopping = picnic = 0

for i in range(1, 249):
    sports = sports + int(data[1][i])
    religious = religious + int(data[2][i])
    nature = nature + int(data[3][i])
    theatre = theatre + int(data[4][i])
    shopping = shopping + int(data[5][i])
    picnic = picnic + int(data[6][i])

#print(sports);print(religious);print(nature);print(theatre);print(shopping);print(picnic);

labels = ['体育场馆', '宗教机构', '自然景观', '剧院展览', '商场购物', '公园野餐']
newdata = [sports, religious, nature, theatre, shopping, picnic]
plt.title('六个场景的总评论数柱状图')
#plt.savefig('pic/六个场景的总评论数柱状图.png')
plt.bar(range(len(newdata)), newdata, tick_label=labels)
plt.show()
```

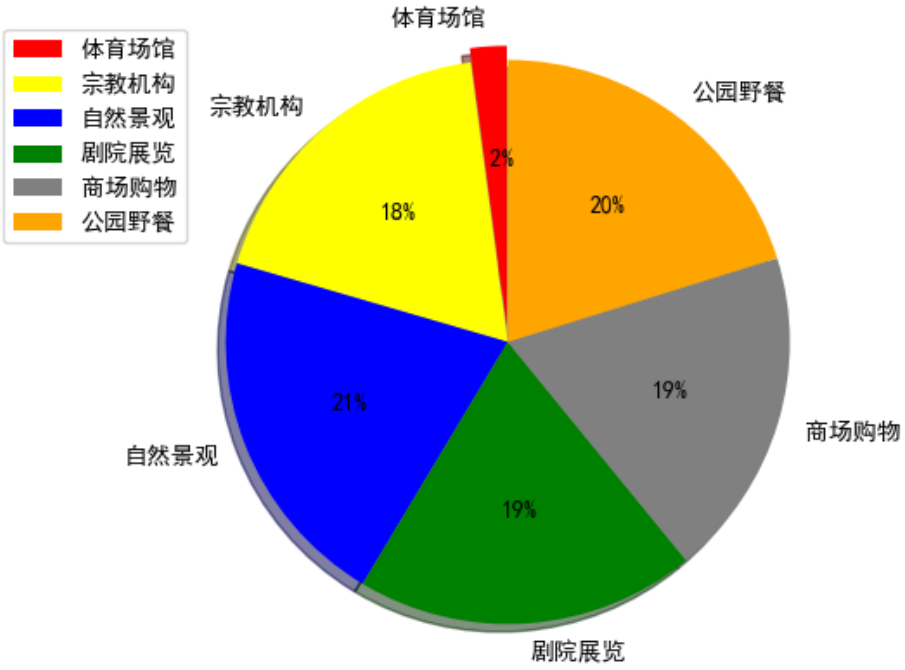
1610121144
马骏超

```python
##1.2 六个场景的总评论数扇形图

labels = labels = ['体育场馆', '宗教机构', '自然景观', '剧院展览', '商场购物', '公园野餐']
sizes = [2.1, 18.4, 20.9, 19.5, 18.9, 20.2]
colors = ['red', 'yellow', 'blue', 'green', 'gray', 'orange']
explode = (0.05, 0, 0, 0, 0, 0)
patches, l_text, p_text = plt.pie(sizes, explode=explode, labels=labels, colors=colors, labeldistance=1.1, autopct='%2.0f%%', shadow=True, startan
for t in l_text:
        t.set_size = 30
for t in p_text:
        t.set_size = 20
plt.axis('equal')
plt.legend(loc='upper left', bbox_to_anchor=(-0.1, 1))
plt.grid()
#plt.savefig('pic/六个场景的总评论数饼状图.png')
plt.show()
```

1610121144
马骏超

图像结果如下图



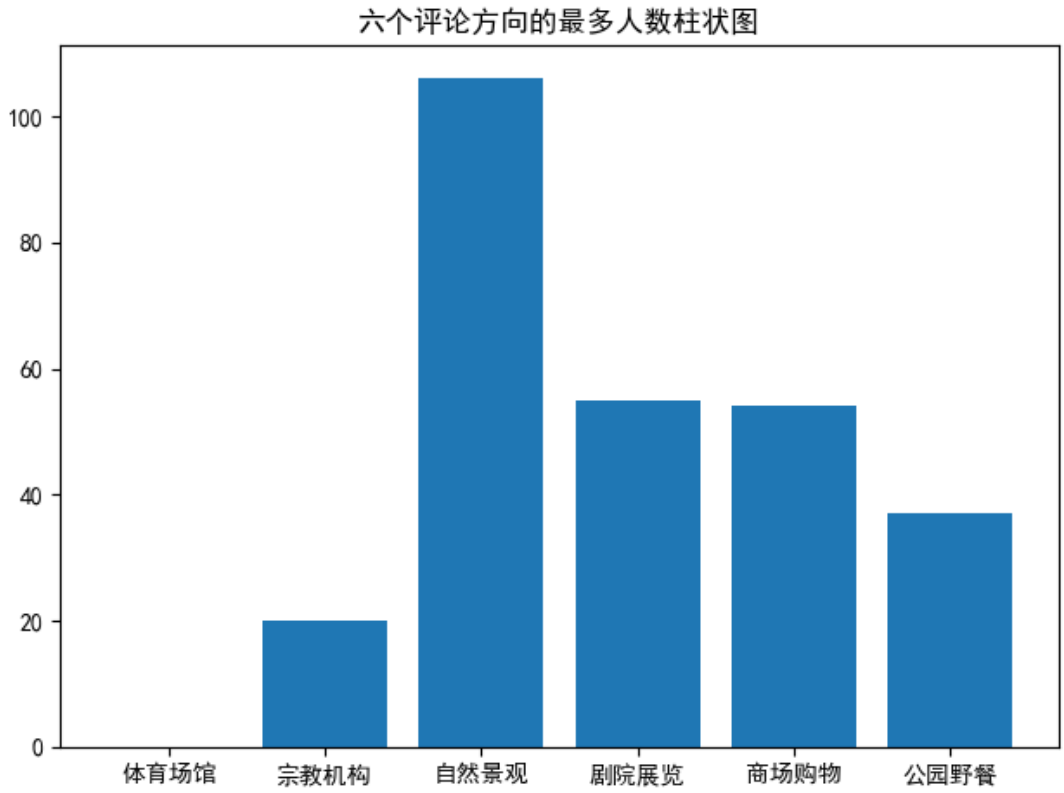


5.核心算法

算法一：六个变量对应六个属性，变量初始值为 0。六个属性中的评论数据，最大者该属性对应的变量+1，最终得到六个评论方向的最多人数柱状图。

```
##2.1 六个评论方向的最多人数柱状图

sports = religious = nature = theatre = shopping = picnic = 0
most = [0, 0, 0, 0, 0, 0]
for i in range(1, 250):
    for j in range(1, 7):
        most[j-1] = int(data[j][i])

    most.sort(reverse=True)
    #print(most)                          1610121144
                                          马骏超
    if most[0] == int(data[1][i]):
        sports = sports+1
    if most[0] == int(data[2][i]):
        religious = religious+1
    if most[0] == int(data[3][i]):
        nature = nature+1
    if most[0] == int(data[4][i]):
        theatre = theatre+1
    if most[0] == int(data[5][i]):
        shopping = shopping+1
    if most[0] == int(data[6][i]):
        picnic = picnic+1

#print(sports);print(religious);print(nature);print(theatre);print(shopping);print(picnic);

labels = ['体育场馆', '宗教机构', '自然景观', '剧院展览', '商场购物', '公园野餐']
newdata = [sports, religious, nature, theatre, shopping, picnic]
plt.title('六个评论方向的最多人数柱状图')
plt.savefig('pic/六个评论方向的最多人数柱状图.png')
plt.bar(range(len(newdata)), newdata, tick_label=labels)
plt.show()
```

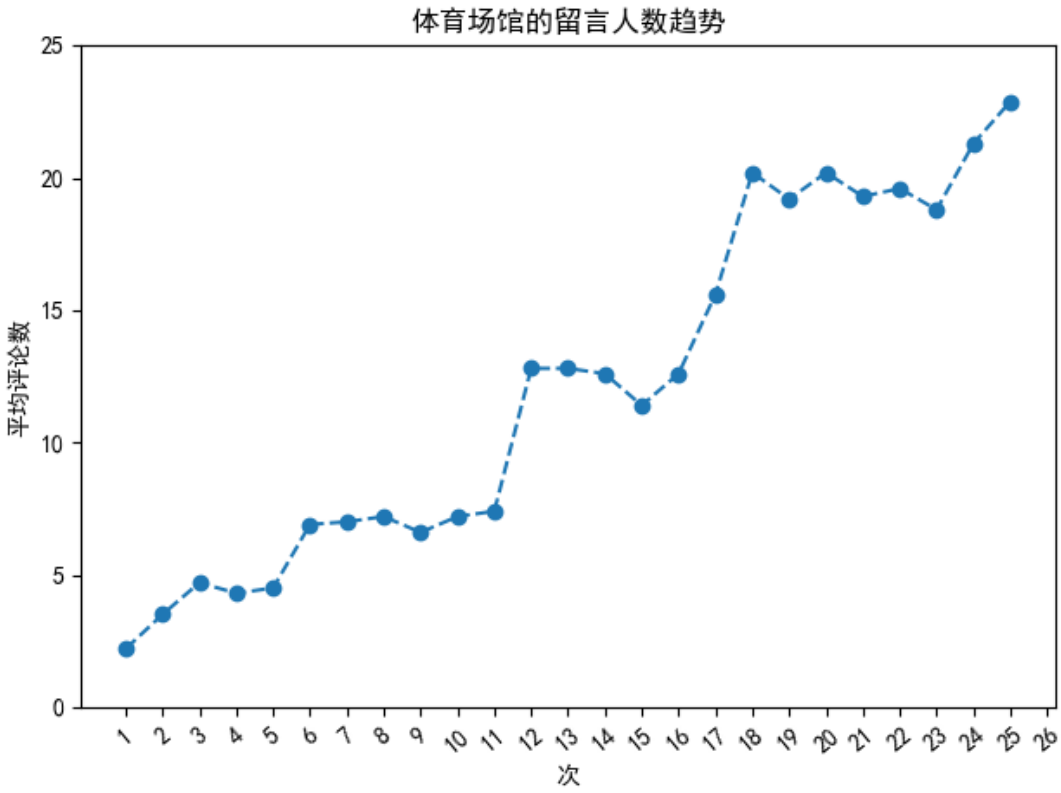
结果如下图



算法二：将数据集中每个场景的每个人的评论数从头至尾取出，每个场景都做一个折线图（由于每个场景都有 249 份数据，所以每 10 条数据取一个平均值，最后 9 条取平均值，共取得 25 条数据），绘制出 xxxx 的留言人数趋势折线图。

```python
#3.1.1 体育场馆的留言人数趋势

a = []
for i in range(0, 250):
    a.append(0)

for i in range(1, 250):
    a[i] = data[1][i]

#print(a)

                                    1610121144
m = 0                               马骏超
n = 0
b = []

for j in range(0, 250):
    m = m + int(a[j])
    n = n + 1
    if j == 249:
        b.append(m / 9)
        m = 0
        n = 0
        break
    if n == 10:
        b.append(m / 10)
        m = 0
        n = 0

#print(b)

pl = plt.figure()
ax = pl.add_subplot(1, 1, 1)
plt.plot(np.arange(1, 26), b[:], linestyle='--', marker='o')
plt.xlabel('次')
plt.ylabel('平均评论数')
plt.xticks(range(1, 27, 1), range(27), rotation=40)
plt.ylim(0, 25)
plt.title('体育场馆的留言人数趋势')
plt.savefig('pic/体育场馆的留言人数趋势.png')
plt.show()
```

结果如图



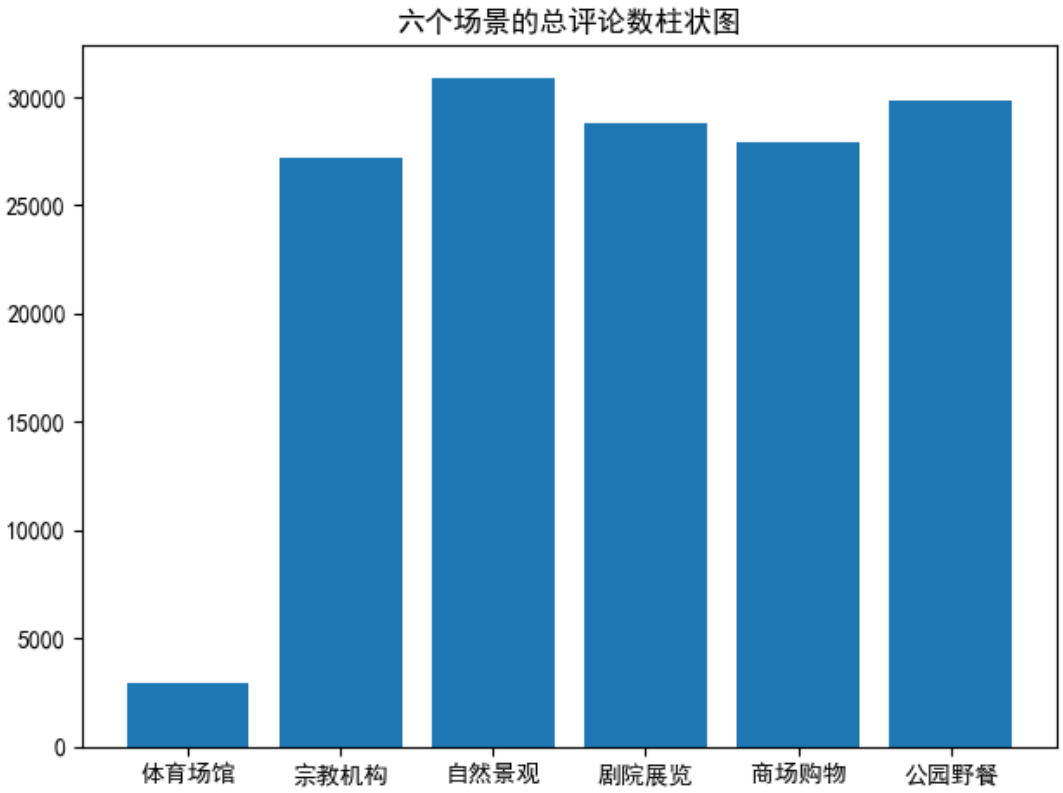
4. 分析结果

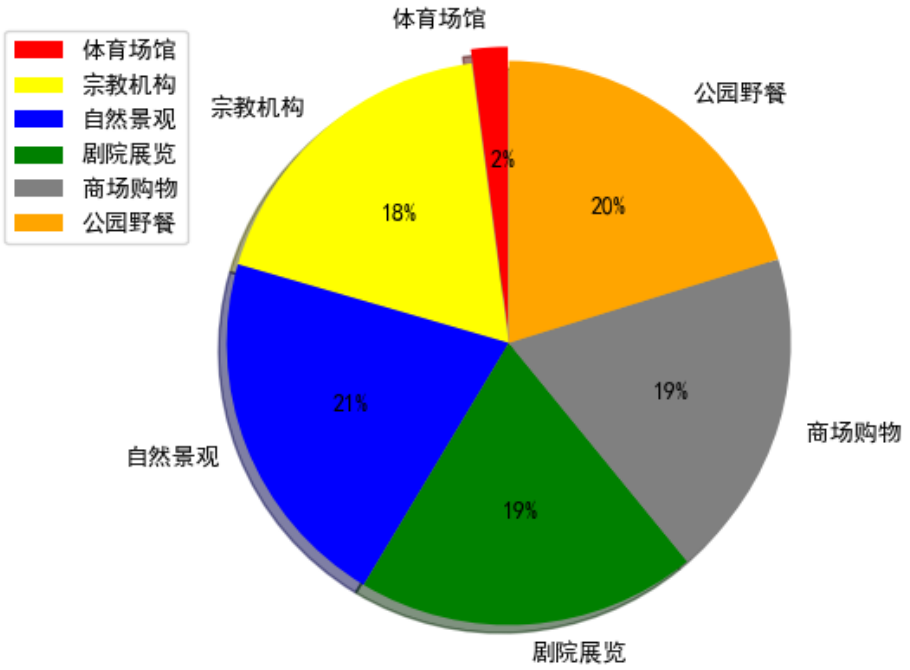
1.去往南印度的旅行者对于“宗教机构”“自然景观”“剧院展览”“商场购物”和“公园野餐”的兴趣充分，而对“体育场馆”的兴趣很低。

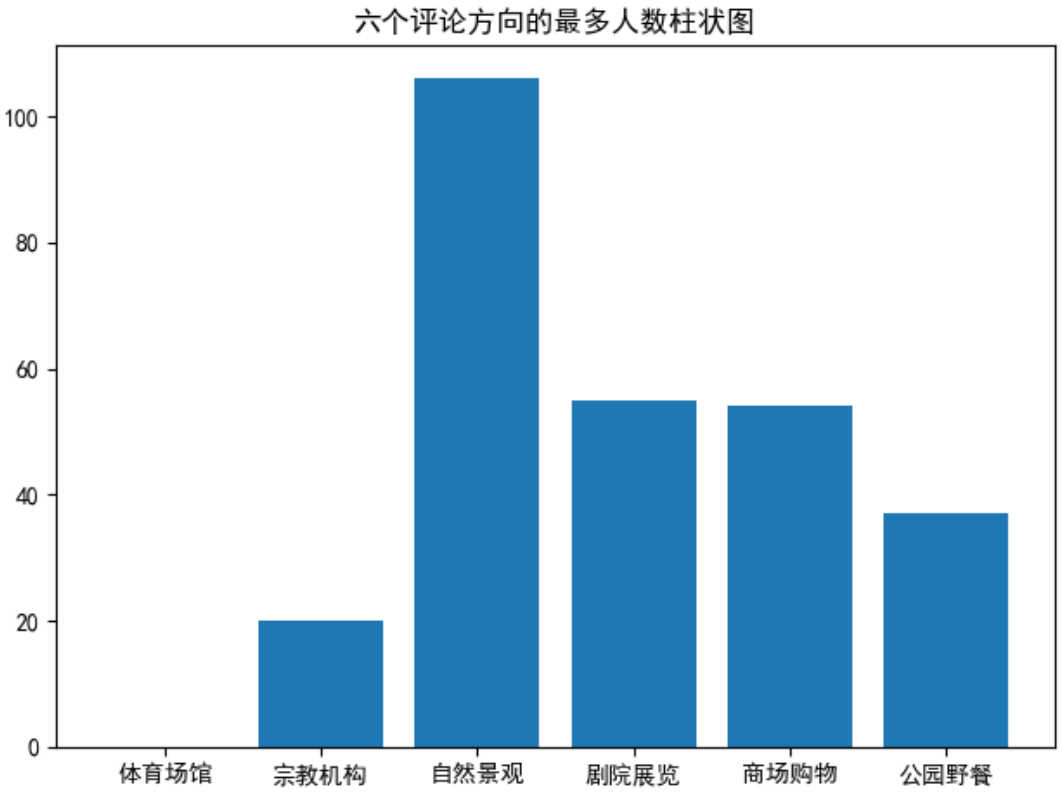
2.去往南印度的旅行者最感兴趣的是当地的“自然景观”，最无感的是当地的“体育场馆”。

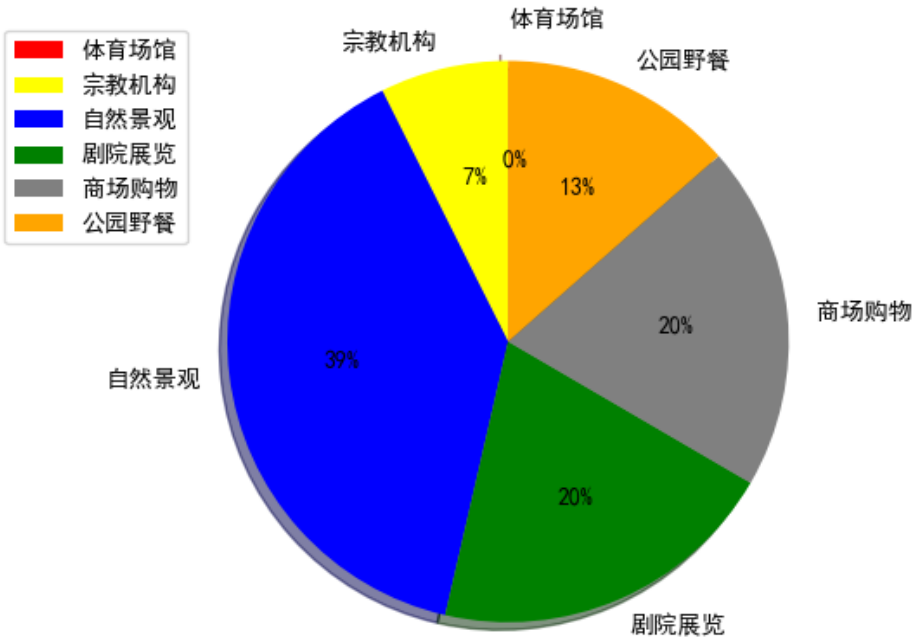
3.南印度的旅游业发展势头持续高涨，六处场地的留言数都在高速增长。

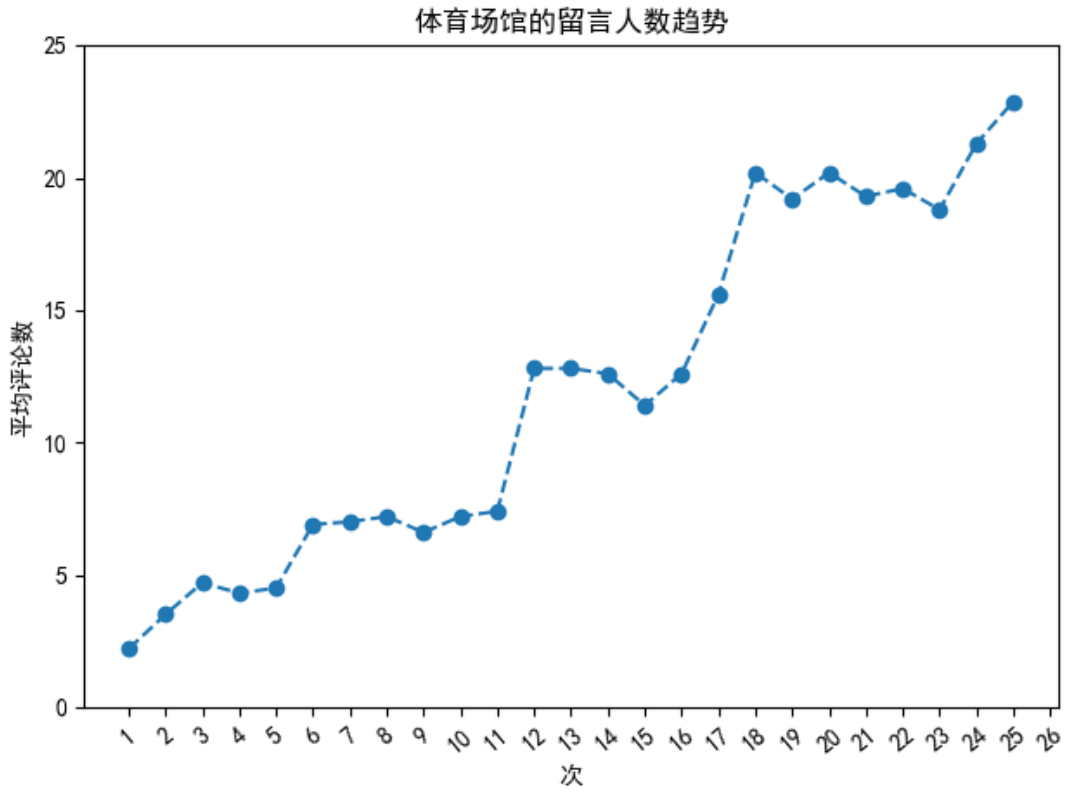
4.图示

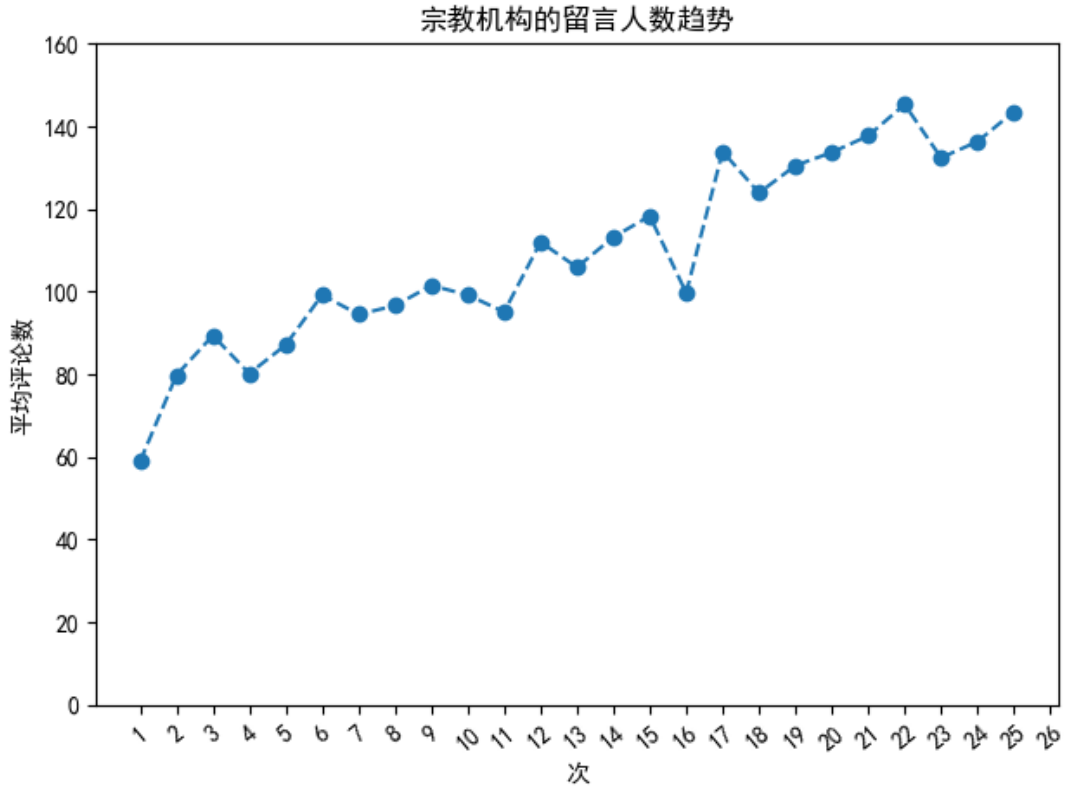

六个场景的总评论数柱状图
30000
25000
20000
15000
10000
5000
0
体育场馆
宗教机构
自然景观
剧院展览
商场购物
公园野餐

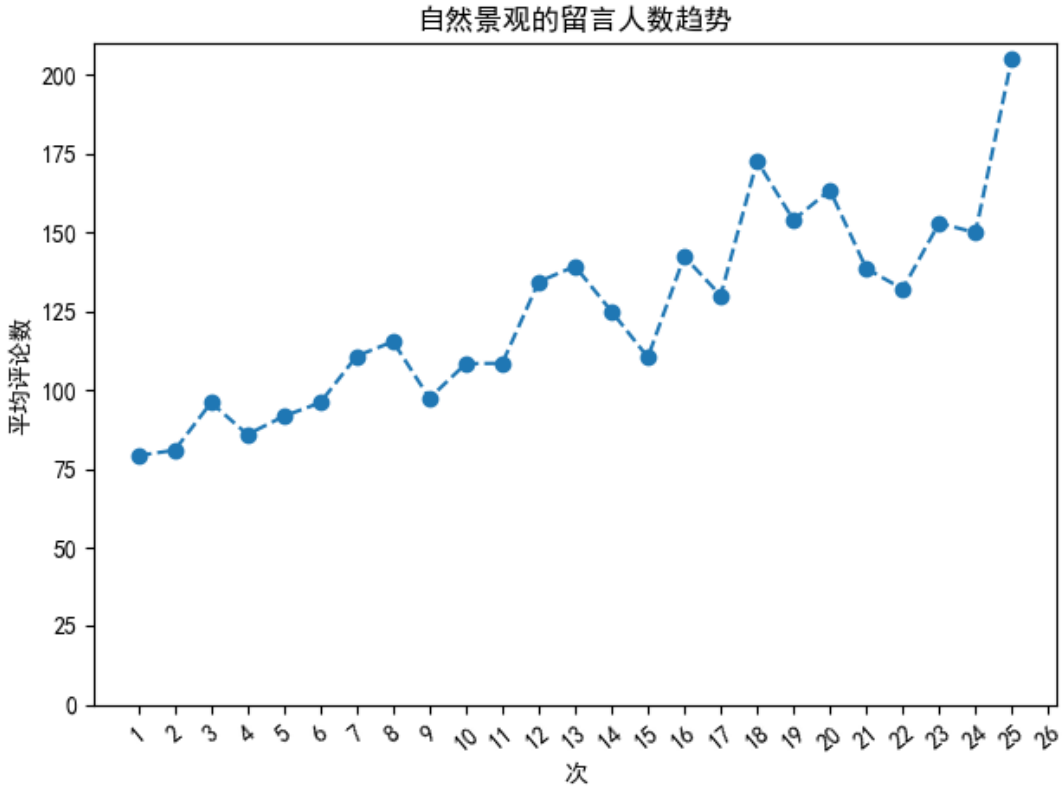

体育场馆
宗教机构
自然景观
剧院展览
商场购物
公园野餐
2%
18%
20%
21%
19%
19%

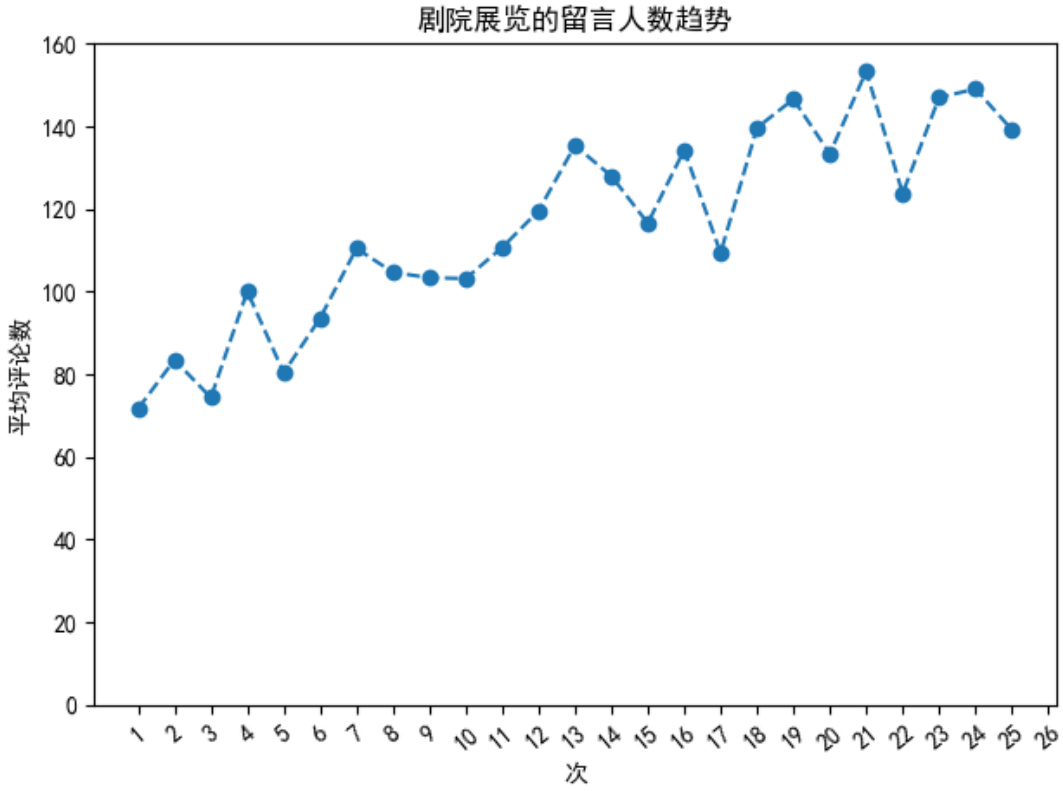

六个评论方向的最多人数柱状图
100
80
60
40
20
0
体育场馆
宗教机构
自然景观
剧院展览
商场购物
公园野餐

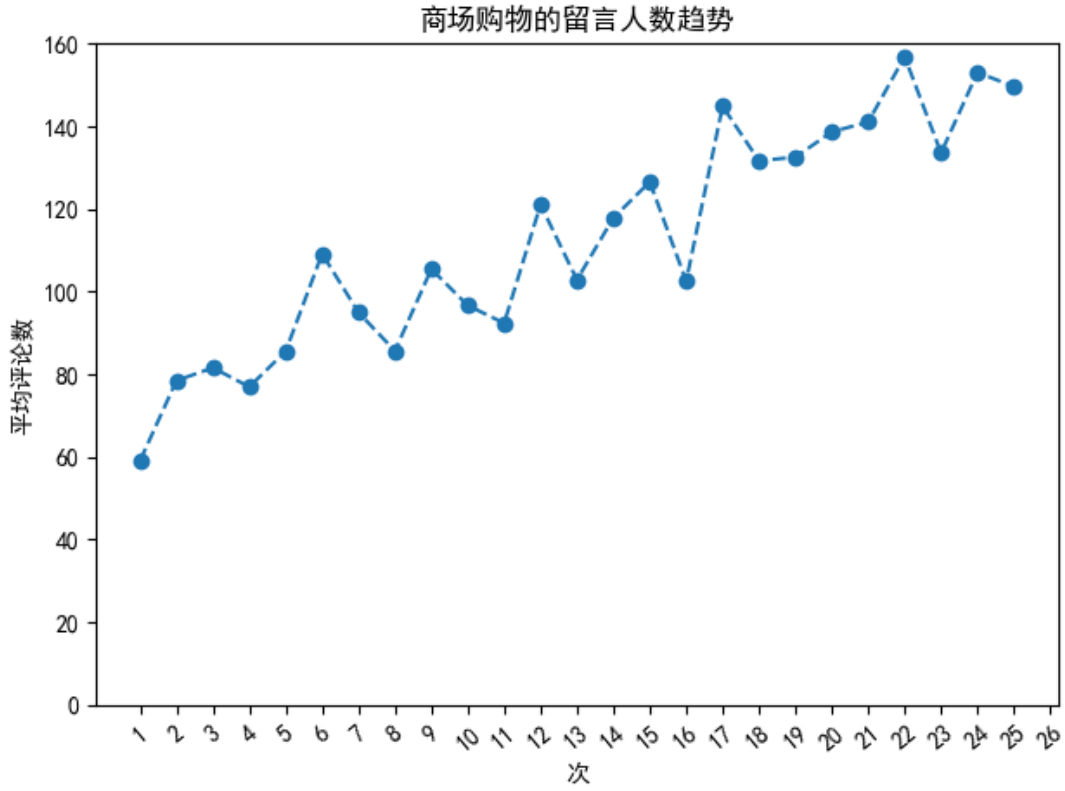

体育场馆
宗教机构
自然景观
剧院展览
商场购物
公园野餐
0%
7%
13%
20%
39%
20%

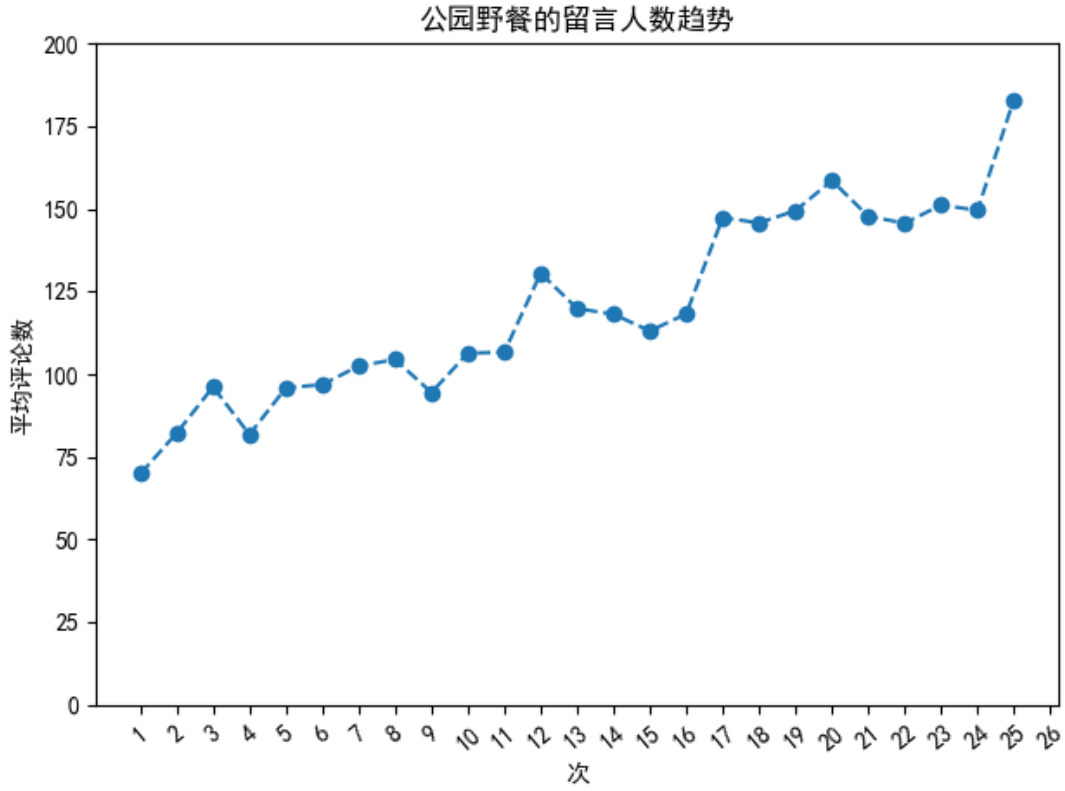

体育场馆的留言人数趋势
平均评论数
次


宗教机构的留言人数趋势
平均评论数
次


自然景观的留言人数趋势
平均评论数
0
25
50
75
100
125
150
175
200
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
次


剧院展览的留言人数趋势
平均评论数
0
20
40
60
80
100
120
140
160
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
次


商场购物的留言人数趋势
平均评论数
次


公园野餐的留言人数趋势
平均评论数
次

5．结论与展望

（1）总结

本次分析实践采用了柱状图和饼状图来描述六个场景评论的数量占比情况，并采用了折线图描述南印度地区的旅游发展情况。分析出了南印度地区的旅游热点地区和旅游发展情况等信息。

（2）结论

1.去往南印度的旅行者对于“宗教机构”“自然景观”“剧院展览”“商场购物”和“公园野餐”的兴趣充分，而对“体育场馆”的兴趣很低。

2.去往南印度的旅行者最感兴趣的是当地的“自然景观”，最无感的是当地的“体育场馆”。

3.南印度的旅游业发展势头持续高涨，六处场地的留言数都在高速增长。

（3）展望

1.此次使用的数据集属性较少，数据量较少，处理难度较小。以后的学习中可以增大难度，锻炼能力。

2.在对数据进行聚类、排序等操作的过程中，使用了 sort()等方法，明白了 Python 中很多情况下都是以 String 类型存放的数据，在进行排序、赋值、计算等操作时，需要转化为 Int 类型才可以进行操作。

3. 此次对数据集中数据的处理时，高级算法使用较少，算法使用较为单一，在遇到对数据的提取、处理等问题时，编写的算法的时间、空间复杂度过于复杂。以后的学习中可以多使用巧妙、高级算法对数据进行提取和分析，提高自己的思维和编程能力。