# 学生校园消费行为分析

## 数据说明

data1.csv、data2.csv对应学生ID表、消费记录表，它们的数据说明分别如表 1和表 2所示。

表 1 data1.csv字段说明

|  |  |
| --- | --- |
| **字段名** | **描述** |
| Index | 序号 |
| CardNo | 校园卡号。每位学生的校园卡号都唯一 |
| Sex | 性别。分为“男”和“女” |
| Major | 专业名称 |
| AccessCardNo | 门禁卡号。每位学生的门禁卡号都唯一 |

表 2 data2.csv字段说明

|  |  |
| --- | --- |
| **字段名** | **描述** |
| Index | 流水号。消费的流水号 |
| CardNo | 校园卡号。每位学生的校园卡号都唯一 |
| PeoNo | 校园卡编号。每位学生的校园卡编号都唯一 |
| Date | 消费时间 |
| Money | 消费金额。单位：元 |
| FundMoney | 存储金额。单位：元 |
| Surplus | 余额。单位：元 |
| CardCount | 消费次数。累计消费的次数 |
| Type | 消费类型 |
| TermSerNo | 消费项目的序列号 |
| conOperNo | 消费操作的编码 |

## 任务1 数据探索与预处理

1. 任务1.1 根据实际项目需求对数据进行必要的数据探索和数据预处理。
2. 任务1.2 对学生个人信息表和消费记录表进行关联。

### 数据探索

1. 统计data1的空值，如图 1所示。

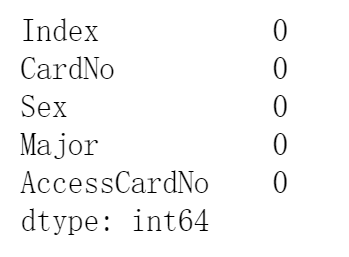


图 1 data1的空值

1. 对data2消费表的Money字段进行统计分析，如均值、方差等，如图 2所示。

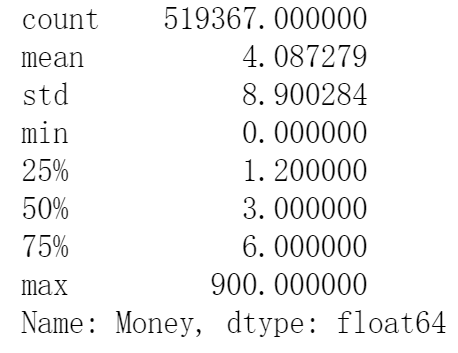


图 2 data2的Money字段的统计分析

根据对消费表的Money字段进行统计分析，最大值为900，在实际学校学生消费情况来看，不合理。

观察原data2表，如图 3所示，可以看到Money=900的Type类型为取款，这时再结合分析内容为“学生校园消费行为分析”，所以只要选择Type为消费的类型，就可以去除异常数据。



图 3 Money=900的数据

1. 统计data2消费表的空值情况，如图 4所示。

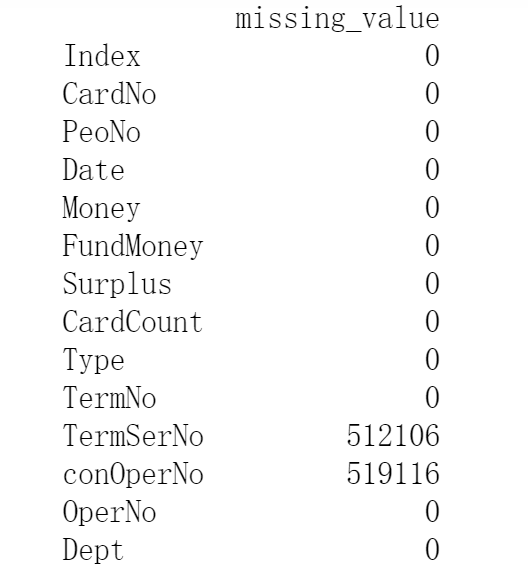


图 4 统计data2消费表的空值情况

根据图 4的统计情况，可以知道TermSerNo、conOperNo字段缺失过多，所以可以在预处理时，直接删除这些字段。

代码 1 数据探索

|  |
| --- |
| import pandas as pd  import os  os.chdir('F:\产品部\在线实习\数据及代码v2.1')  data1 = pd.read\_csv('data\data1.csv', encoding = 'gbk')  data2 = pd.read\_csv('data\data2.csv', encoding = 'gbk')  # ----------data1数据探索-----------  # data1查看描述性统计  print(data1.describe())  # 查看data1数据类型  print(data1.dtypes)  # 统计空值  print(data1.isnull().sum())  # ----------data2数据探索-----------  # data2查看描述性统计  data2\_des = data2['Money'].describe()  print(data2\_des)  # 统计空值  data2\_null = data2.isnull().sum()  data2\_null = pd.DataFrame(data2\_null,columns=['missing\_value'])  print(data2\_null)  print('data1数据形状',data1.shape)  print('data2数据形状',data2.shape) |

### 缺失值处理

通过对学生消费记录进行缺失值检查，发现存在缺失值较大的两个列，分别是TermSerNo列和conOperNo列，缺失值分别达到512106和519116，对于缺失值较大的列，在实际的数据分析中无意义，予以删除处理。

代码 2 缺失值处理

|  |
| --- |
| # 缺失值处理,筛选不要的列  data2 = data2.drop(['TermSerNo', 'conOperNo'], axis = 1)  data2 = data2.dropna() # 去空值 |

### 重复值处理

查看data1的重复值。因为每位学生的校园卡号和门禁卡号都唯一，所以要进行重复值检查。

代码 3 字段异常值处理

|  |
| --- |
| # 查看data1的CardNo有没有重复值  data1\_drop=data1.drop\_duplicates(['CardNo'])  print('CardNo去重前数据形状:',data1.shape)  print('CardNo去重后数据形状:',data1\_drop.shape)  # 查看data1的AccessCardNo有没有重复值  data1\_drop=data1.drop\_duplicates(['AccessCardNo'])  print('AccessCardNo去重前数据形状:',data1.shape)  print('AccessCardNo去重后数据形状:',data1\_drop.shape) |



图 5 去重情况

由图 5可知，CardNo没有重复值，AccessCardNo列有两条记录存在重复，但是我们是否要进行处理呢，还要根据实际进行下一步的分析。

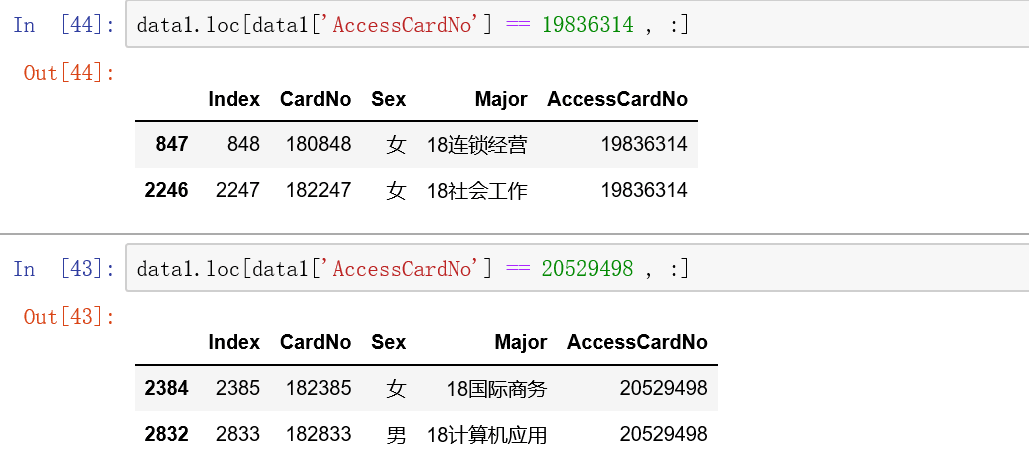


图 6 重复值

由图 6可知，虽然AccessCardNo列存在重复，但是CardNo是没有的，由于最终使用的是CardNo字段进行表关联，并没有使用AccessCardNo，且AccessCardNo对后续的任务没有影响，在这里可不做处理。

### 异常值处理

查看data2学生消费表的消费时间是否存在异常值，以及只筛选消费类型的记录。

代码 4 时间异常值处理

|  |
| --- |
| # 修改Date列为时间序列  data2['Date']=pd.to\_datetime(data2['Date'])  # 查看data2学生消费表的消费时间有没有异常值  data2['hour']=data2.Date.dt.hour #提取小时  print('最小时间',data2['hour'].min())  print('最大时间',data2['hour'].max())  # data2 消费时间异常，正常的学校饭堂并不会通宵营业，所以筛选5点以后的数据  data2=data2[data2['hour']>5]  # 筛选消费类型的记录  data2=data2[data2['Type'] =='消费'] |

提取小时出来，发现分布在0-23时之间，没有异常。由于该分析过程学校食堂的营业时间为6:00-24:00，因此对0:00-6:00之内的所有消费记录记为异常值并删除。

### 数据合并

对消费表录与学生表进行合并，合并结果如图 7所示。

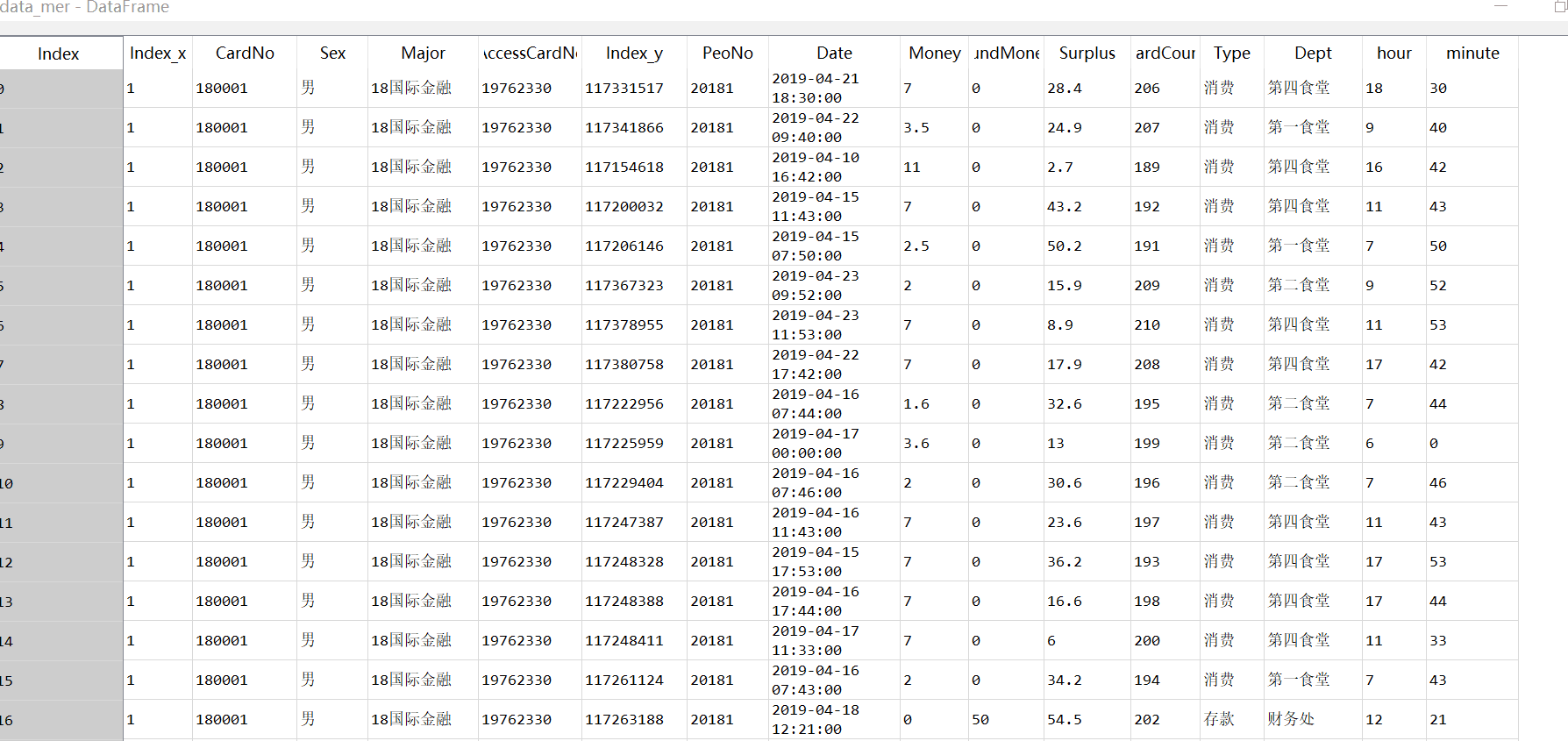


图 7 合并结果

代码 5 数据合并

|  |
| --- |
| # 对学生个人信息表和消费记录表进行关联  data\_mer = pd.merge(data1, data2, left\_on = 'CardNo', right\_on = 'CardNo', how = 'inner')  # 统计合并后数据  data\_out = data\_mer['Money'].describe().T  data\_out = data\_mer.isnull().sum(axis=0)  data\_out = pd.DataFrame(data\_out,columns=['missing\_value'])  print(data\_out)  print('合并后数据形状',data\_mer.shape)  # 保存数据  data\_mer.to\_csv('tmp/data\_mer.csv',index = False,encoding = 'gbk')  data2.to\_csv('tmp/data2.csv',index = False,encoding = 'gbk') |

这里使用pandas库中的merge函数，关联主键为CardN，关联方式为“inner”。

## 任务2 食堂就餐行为分析

1. 任务2.1 绘制各食堂就餐人次的占比饼图，分析学生早中晚餐的就餐地点是否有显著差别。
2. 任务2.2 通过食堂刷卡记录，分别绘制工作日和非工作日食堂就餐时间曲线图，分析食堂早中晚餐的就餐峰值。

### 学生早中晚餐就餐地点占比饼图

1. 查看数据，如图 8所示，可以看到Dept字段除了有食堂类型外，还有其他地点，如人文社科等。根据要求，绘制的是学生就餐地点饼图，所以只提取有关食堂的记录，考虑到教师食堂一般不对学生开放，故不纳入学生就餐餐厅范围内。



图 8 统计Dept字段

1. 因为学生打饭为饭菜分开，不能刷卡一次就认为就餐一次，所以需要引入就餐人次的定义：如果学生前后刷卡时间小于10分钟（600秒）的数据，那么就统计为一次就餐次数。

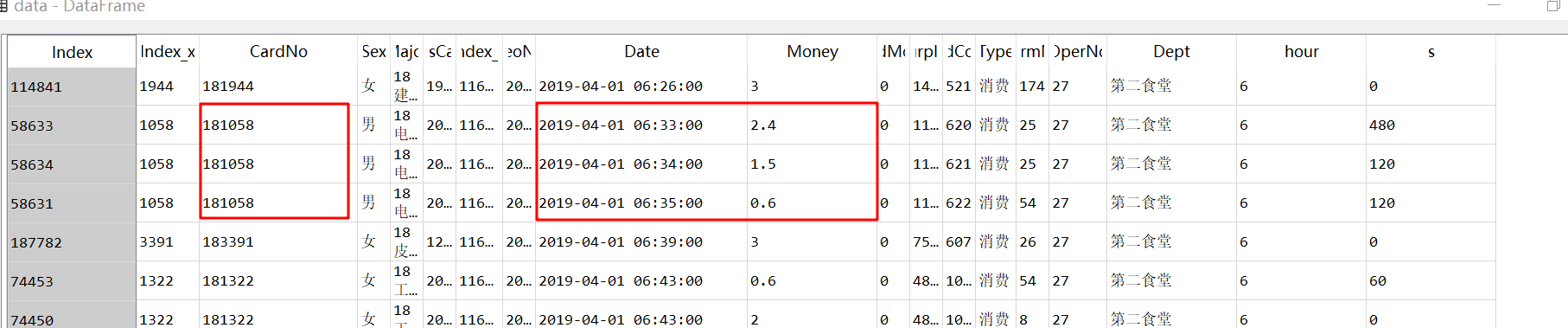


图 9 就餐人次

如图 9所示，卡号为181058的学生，在6点33、34、35分都有刷卡记录，看刷卡金额也可以知道，这是一次就餐行为，如果直接统计刷卡频次作为就餐频次，显然不合理。

处理学生消费记录表，合并同一地点同一刷卡时间的就餐记录。

实现过程如下：

* 添加一列标签列lab；
* 用shift()函数判断，如果前一个就餐地点不等于下一个就餐地点，那么记为标签为1；
* 对每个卡号分别计算前后刷卡时间差（单位：秒），在这个过程一定要对时间进行排序，计算的时间差才是前后刷卡时间差。如果时间差大于10分钟，那么记为标签为1；
* 用dcumsum()函数对标签列lab，进行累记，记为re\_index列；
* 对re\_index进行分组聚和，求一次就餐金额、刷卡时间和，并且进行去重；
* 最后直接统计每个卡号的长度即为就餐频次，消费金额为一次就餐金额。

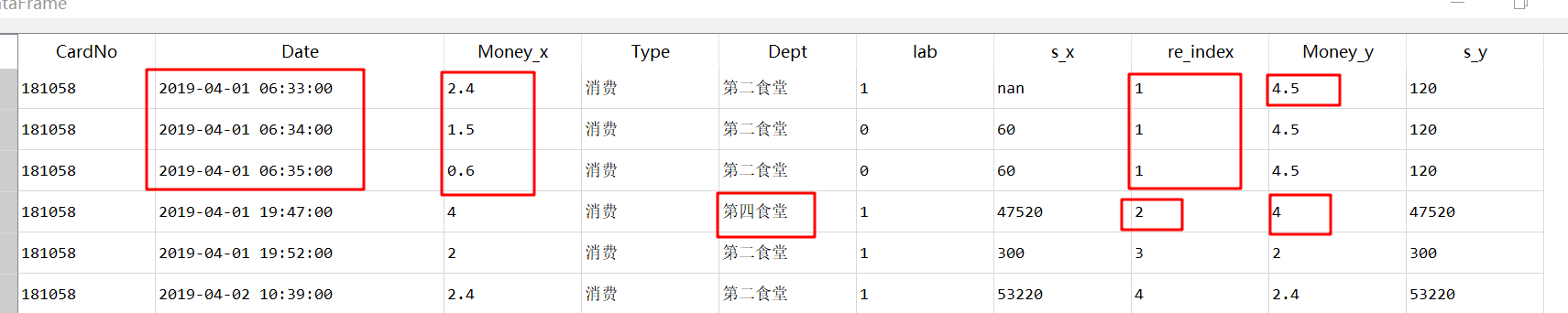


图 10 计算就餐人次

如图 10所示，卡号为181058的学生，在6点33、34、35分都有刷卡记录，通过计算在同一地点（第二食堂），就餐金额为4.5元，只要对re\_index进行去重，统计频数，就可得出就餐次数。

代码 6 筛选数据和统计就餐次数

|  |
| --- |
| # 任务2.1  # 筛选食堂的数据  import pandas as pd  from collections import Counter  import matplotlib.pyplot as plt  import warnings  warnings.filterwarnings("ignore")  import os  os.chdir('F:\产品部\在线实习\数据及代码v2.1')  data2= pd.read\_csv("tmp\data2.csv" ,encoding = 'gbk')  index=['第一食堂','第二食堂','第三食堂','第四食堂','第五食堂']  data = data2.loc[data2['Dept'].isin(index)]  data['Date'] = pd.to\_datetime(data['Date'], format='%Y-%m-%d %H:%M:%S')  # 处理单个学生消费记录表，合并同一地点同一刷卡时间的就餐记录。  # 实现过程：1、为了实现“前后刷卡”，所以对时间进行排序  # 2、添加一列标签列  # 3、用shift()函数判断，如果前一个就餐地点不等于下一个，那么记为标签为1  # 4、用diff()函数计算前后刷卡时间差  # 5、如果时间差大于10分钟，那么记为标签为1  # 6、用dcumsum()函数对标签列进行累记，记为re\_index列  # 7、对re\_index进行分组聚和，求一次就餐金额、刷卡时间和，并且进行去重  def deal\_data(data):  data=data.sort\_values(by='Date') # 对时间进行排序  data['lab']=0  data.loc[data['Dept']!=data['Dept'].shift(1),'lab']=1  data['s']=data['Date'].diff()  # 时间差转化为秒  data['s']=[s.total\_seconds() for s in data['s']]  # 默认第一次刷卡为一次就餐，赋值大于阈值即可(设置为10分钟=600秒)  data.loc[data['s']>600,'lab']=1  data['re\_index']=data['lab'].cumsum()  df1=data.groupby(['re\_index']).agg({'Money':sum,'s':sum}).reset\_index()  data\_new=pd.merge(data,df1,on='re\_index')  data\_new=data\_new.drop\_duplicates(['re\_index'])  del data\_new['s\_x'],data\_new['re\_index'],data\_new['Money\_x']  return data\_new  # 对所有学生进行分组，处理消费记录  data\_new=data.groupby(['CardNo']).apply(deal\_data) |

1. 根据如下分类，绘制学生早中晚餐就餐地点的占比饼图。

早餐：6点，7点，8点，9点；

午餐：11点，12点，13点；

晚餐：17点，18点，19点，20点。

早餐就餐次数如图 11所示。

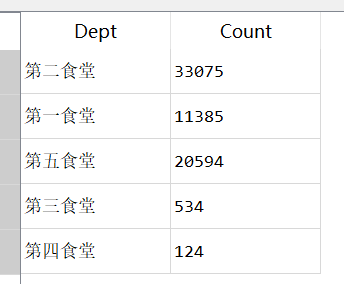


图 11 早餐就餐次数

午餐就餐次数如图 12所示。

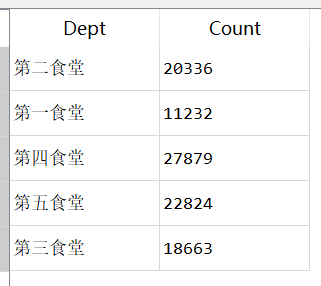


图 12 午餐就餐次数

晚餐就餐次数如图 13所示。

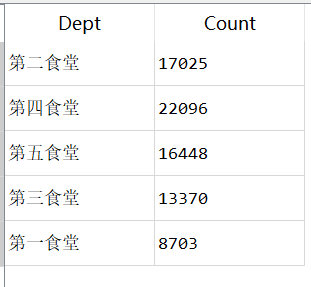


图 13 晚餐就餐次数

代码 7 划分早中晚并统计就餐次数

|  |
| --- |
| # 绘制各食堂就餐人次的占比饼图，分析学生早中晚餐的就餐地点是否有显著差别。  # 提取早中晚时间  data\_morning = data\_new.loc[(data\_new['hour'].apply(lambda x: x in [6, 7, 8, 9,])), :]  data\_noon = data\_new.loc[(data\_new['hour'].apply(lambda x: x in [11, 12, 13])), :]  data\_dinner = data\_new.loc[(data\_new['hour'].apply(lambda x: x in [17, 18, 19, 20])), :]  # 统计早中晚餐就餐次数  data\_dict1 = pd.DataFrame.from\_dict(Counter(data\_morning['Dept']), orient = 'index').reset\_index()  data\_dict2 = pd.DataFrame.from\_dict(Counter(data\_noon['Dept']), orient = 'index').reset\_index()  data\_dict3 = pd.DataFrame.from\_dict(Counter(data\_dinner['Dept']), orient = 'index').reset\_index()  data\_dict1.columns = ['Dept', 'Count']  data\_dict2.columns = ['Dept', 'Count']  data\_dict3.columns = ['Dept', 'Count'] |

早中晚餐就餐地点的占比饼图，如图 14：

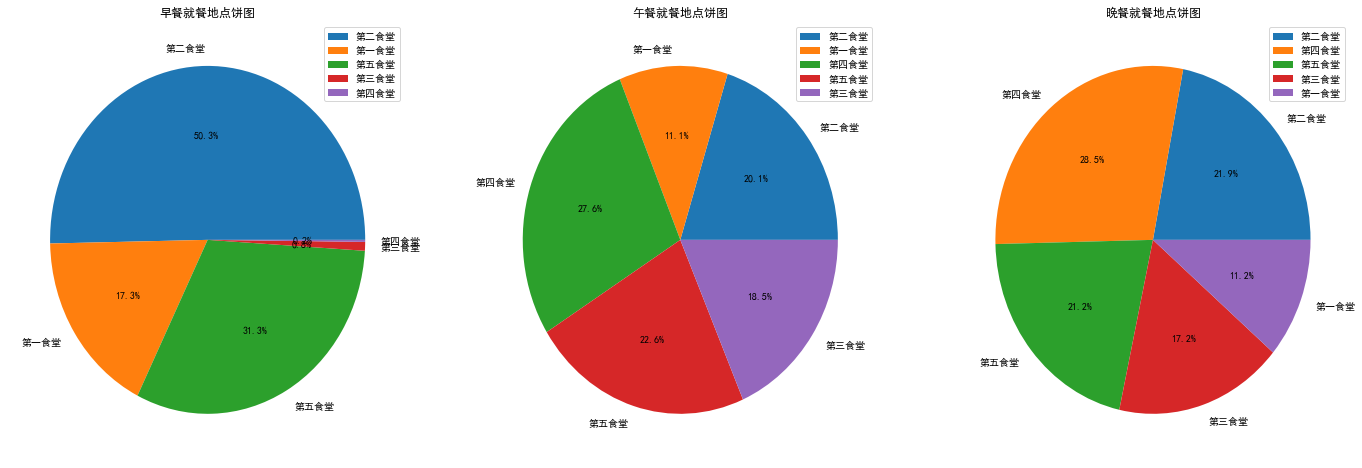


图 14 早中晚餐就餐地点的占比饼图

分析如下：

早餐：学生就餐地点第二食堂人数最多，其次为第五食堂等；

午餐：学生就餐地点第四食堂人数最多，其次为第五食堂、第二食堂、第三食堂等；

晚餐：学生就餐地点第四食堂人数最多，其次为第五食堂、第二食堂等。

对于时间段而言，第二、五食堂的早餐供应量可以增加，第三、四食堂的早餐供应量可以减少，午餐晚餐的各个食堂的就餐人数都相对均衡，可以不做调整。

代码 8 绘制占比饼图

|  |
| --- |
| plt.rcParams['font.sans-serif'] = 'SimHei' # 设置中文显示  p = plt.figure(figsize = (24, 8)) # 将画布设定为正方形，则绘制的饼图是正圆  ax1 = p.add\_subplot(1,3,1)  label1 = data\_dict1['Dept'] # 定义饼图的标签，标签是列表  values1 = data\_dict1['Count']  plt.pie(values1, autopct = '%1.1f%%', labels = label1) # 绘制饼图  plt.title('早餐就餐地点饼图') # 绘制标题  plt.legend(loc='upper right')  ax2 = p.add\_subplot(1,3,2)  label2 = data\_dict2['Dept'] # 定义饼图的标签，标签是列表  values2 = data\_dict2['Count']  plt.pie(values2, autopct = '%1.1f%%', labels = label2) # 绘制饼图  plt.title('午餐就餐地点饼图') # 绘制标题  plt.legend(loc='upper right')  ax3 = p.add\_subplot(1,3,3)  label3 = data\_dict3['Dept'] # 定义饼图的标签，标签是列表  values3 = data\_dict3['Count']  plt.pie(values3, autopct = '%1.1f%%', labels = label3) # 绘制饼图  plt.title('晚餐就餐地点饼图') # 绘制标题  plt.legend(loc='upper right')  plt.show() |

### 工作日和非工作日食堂就餐时间曲线图

1. 根据如下分类，绘制工作日和非工作日食堂就餐时间曲线图。

工作日：星期一、二、三、四、五，以及调休日（由于五一假期，所以04-28要上班）.

非工作日：星期六、日，节假日（清明节）。

实现过程如下（0表示工作日，1表示非工作日）：

* 对日期划分星期，并新增weekday字段，对星期1-5标记为0，星期6、7标记为1；

其中四月份还包含节假日，要另外处理；

* 节假日处理：提取天数，并新增day字段，判断如果day等于节假日时间，那么标记为1。

部分数据如图 15所示。

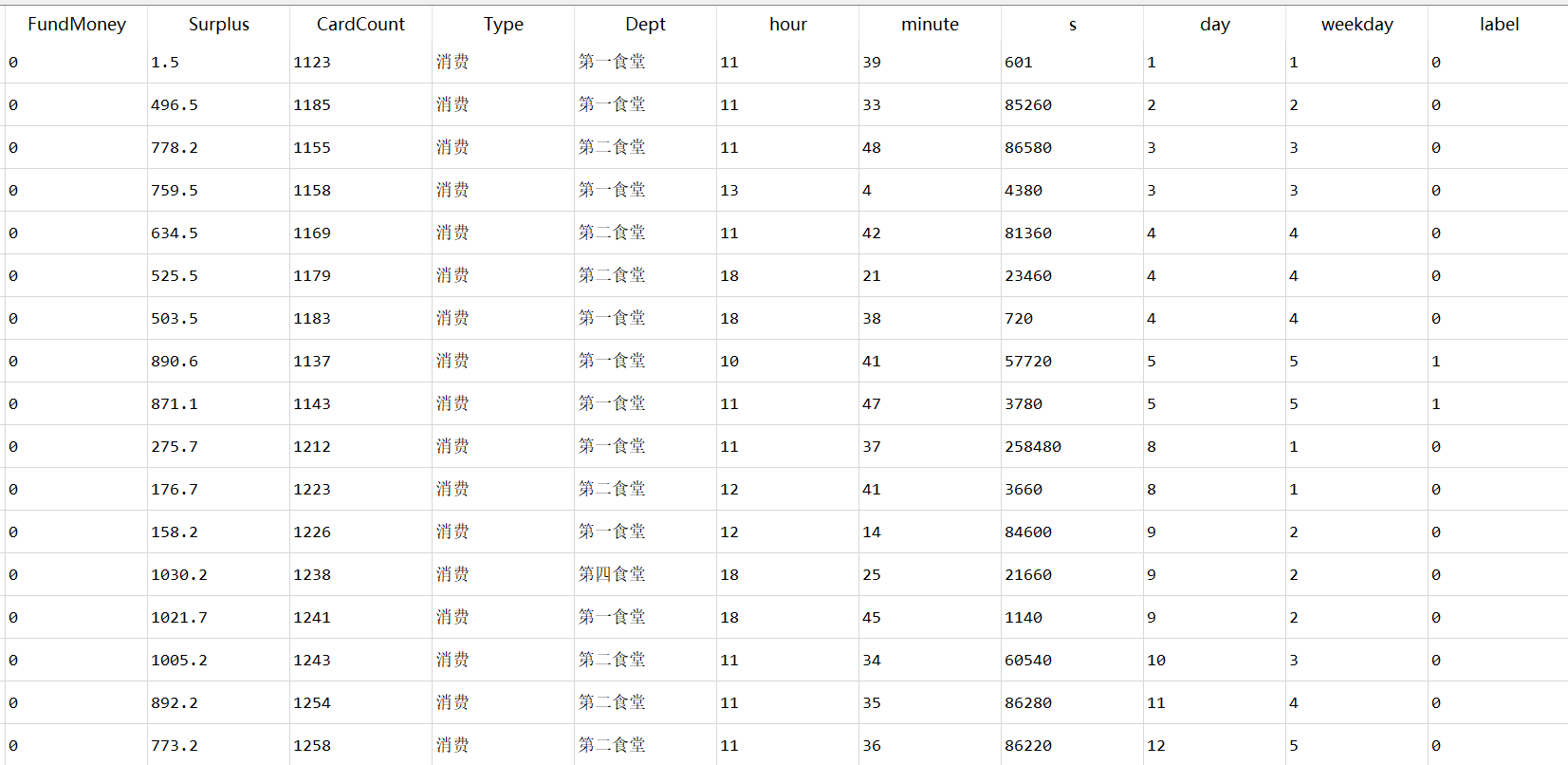


图 15 工作日与非工作日划分

工作日每小时食堂就餐频次如图 16所示。

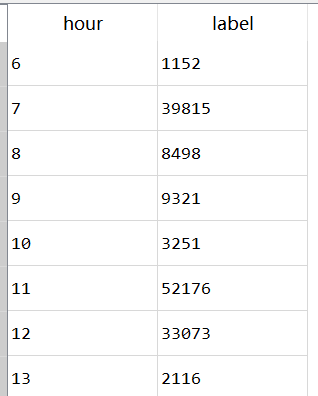


图 16 工作日每小时食堂就餐频次

非工作日每小时食堂就餐频次如图 17所示。

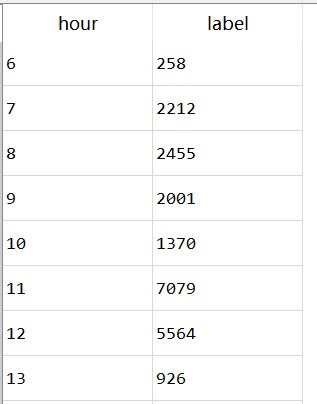


图 17 非工作日每小时食堂就餐频次

代码 9 划分工作与非工作日并统计就餐次数

|  |
| --- |
| # 任务2.2  # 通过食堂刷卡记录，分别绘制工作日和非工作日食堂就餐时间曲线图，分析食堂早中晚餐的就餐峰值。  # 划分工作日和非工作日，0表示工作日，1表示非工作日  data\_new['day'] = data\_new['Date'].apply(lambda x: x.day)  # 工作日与休息日  data\_new['weekday'] = data\_new['Date'].apply(lambda x: x.weekday()+1)  isrest = ((data\_new['weekday'] == 6) | (data\_new['weekday'] == 7))  data\_new['label'] = isrest\*1  # 节假日(5、6、7日为清明节放假，28日因五一假期，所以调休后04-28要上班)  holiday = [5, 6, 7]  for i in holiday:  data\_new.loc[data\_new['day']==i,'label']=1  data\_new.loc[data\_new['day']==28,'label']=0  data\_weekday = data\_new[data\_new['label']==0] # 提取工作日数据  data\_weekend = data\_new[data\_new['label']==1] # 提取非工作日数据  print(data\_weekday.shape)  print(data\_weekend.shape)  # 计算工作日和非工作日每小时食堂就餐频次  data\_gb1 = data\_weekday['label'].groupby(data\_weekday['hour']).count().reset\_index()  data\_gb2 = data\_weekend['label'].groupby(data\_weekend['hour']).count().reset\_index() |

1. 根据工作日和非工作日的label字段标签进行分组聚合，绘制折线图，如图 18所示。

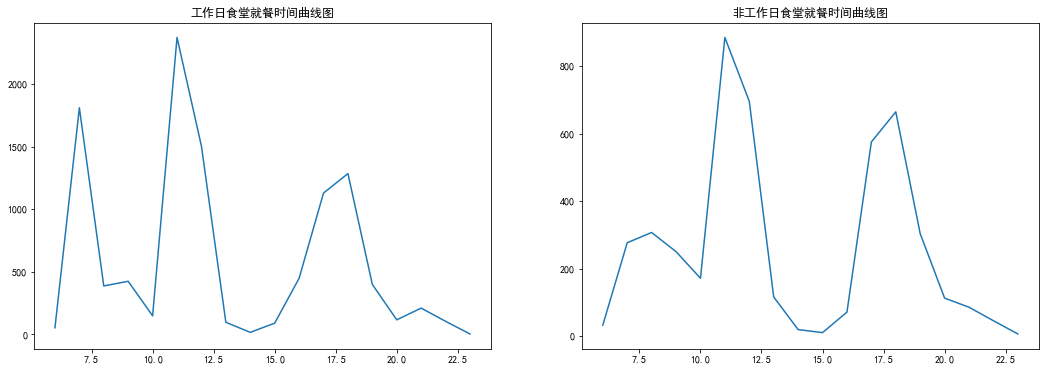


图 18 工作日和非工作日就餐时间曲线图

1. 对于就餐时间而言，工作日早中晚的供应量应该要增加，非工作日早餐可以考虑减少供应量以避免食物浪费。
2. 在工作日就餐高峰期基本都是学生放学后的就餐时间，所以人流大。针对这种情况，学校可以采用分流的方式，调整学生上下课时间，把不同课程的学生上下课时间错开。
3. 食堂可以在高峰期前一两个小时开始准备食物，以避免时间的浪费。
4. 整体来看，可以在周末仅开设第二食堂和第五食堂，在休息日就餐数量来看，这两个食堂足够去供应人们的就餐消费，而且同时还可以减少运营的成本。

代码 10 工作日和非工作日食堂就餐时间折线图

|  |
| --- |
| # 画工作日和非工作日食堂就餐时间折线图  plt.rcParams['font.sans-serif'] = 'SimHei' # 设置中文显示  p = plt.figure(figsize = (18, 6)) # 将画布设定为正方形，则绘制的饼图是正圆  ax1 = p.add\_subplot(1,2,1)  plt.plot(data\_gb1['hour'], data\_gb1['label'])  plt.title('工作日食堂就餐时间曲线图')  ax2 = p.add\_subplot(1,2,2)  plt.plot(data\_gb2['hour'], data\_gb2['label'])  plt.title('非工作日食堂就餐时间曲线图')  plt.show() # 显示图形  data.to\_csv('tmp/data2\_new.csv',index = False,encoding = 'gbk') |

## 任务3 学生消费行为分析

1. 任务3.1 根据学生的整体校园消费数据，计算本月人均刷卡频次和人均消费，分析不同专业间不同性别学生群体的消费特点。
2. 任务3.2 根据学生的整体校园消费行为，选择合适的特征，构建聚类模型，分析每一类学生群体的消费特点，为学校判定学生的经济状况提供参考意见。

### 计算本月刷卡频次和人均消费

1. 本月人均刷卡频次和人均消费如表 3所示。

表 3 本月人均刷卡频次和人均消费

|  |  |
| --- | --- |
| **人均刷卡频次** | **人均消费** |
| 58.521981939163496 | 246.217105513308 |

代码 11 本月人均刷卡频次和人均消费

|  |
| --- |
| import numpy as np  import pandas as pd  import matplotlib.pyplot as plt  # 任务3.1  import os  os.chdir('F:\\产品部\\数据及代码\\tmp')  data2= pd.read\_csv('data2.csv', encoding = 'gbk')  # 总消费次数  tic = len(data2)  # 总消费金额  price = data2['Money'].sum()  # 总人数  people = len(data2['CardNo'].unique())  # 本月人均刷卡频次  aver\_tic = tic/people  print('本月人均刷卡频次',aver\_tic)  # 人均消费金额  aver\_price = price/people  print('本月人均消费金额',aver\_price) |

1. 为了分析不同专业、不同性别的学生消费，所以还是使用合并后的数据，不同专业间不同性别学生消费情况，如图 19所示。

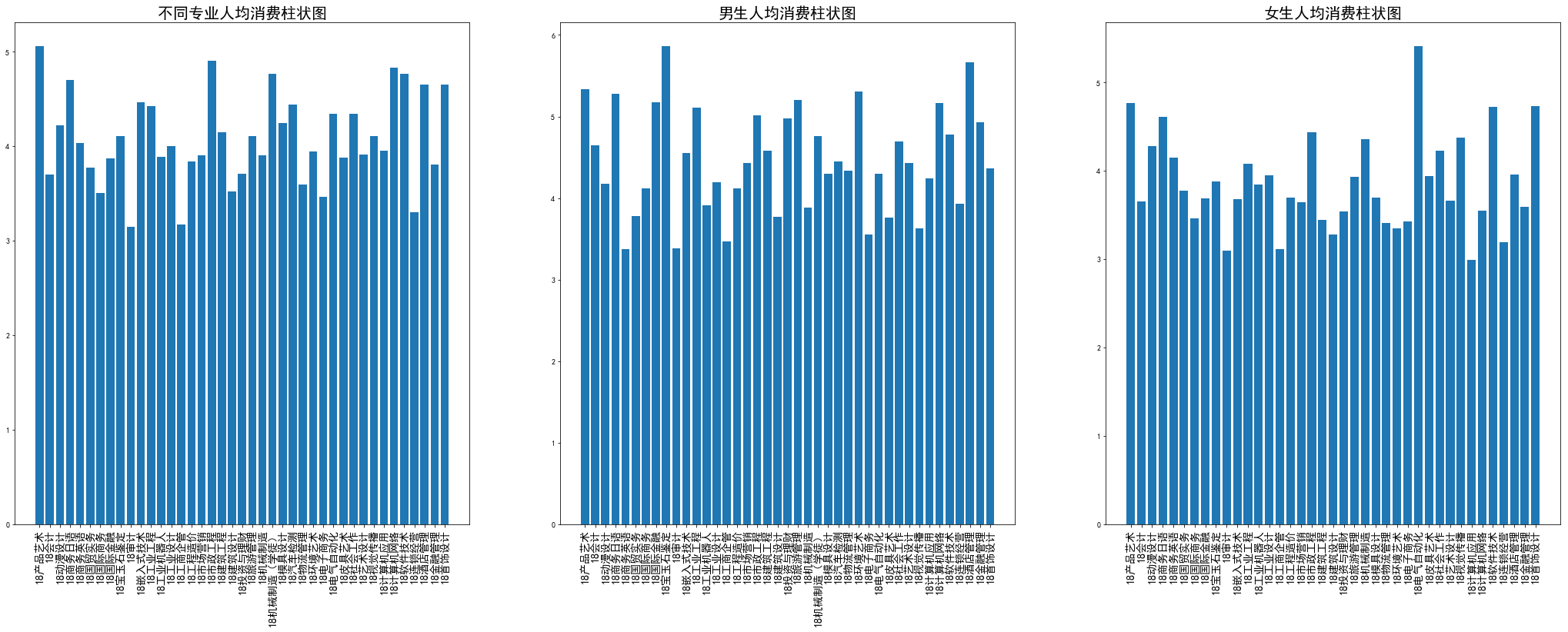


图 19 不同专业间不同性别学生消费情况

代码 12 绘制不同专业、不同性别人均消费柱状图

|  |
| --- |
| data\_mer= pd.read\_csv('../tmp/data\_mer.csv', encoding = 'gbk')  # 计算不同专业、不同性别人均消费  data\_gb1 = data\_mer['Money'].groupby(data\_mer['Major']).mean().reset\_index()  data\_gb2 = data\_mer['Money'].groupby([data\_mer['Sex'], data\_mer['Major']]).mean().reset\_index()  data\_boy = data\_gb2[data\_gb2['Sex'] == '男']  data\_girl = data\_gb2[data\_gb2['Sex'] == '女']  # 绘制不同专业、不同性别人均消费柱状图  plt.rcParams['font.sans-serif'] = 'SimHei' # 设置中文显示  p = plt.figure(figsize = (36,12)) # 将画布设定为正方形，则绘制的饼图是正圆  ax1 = p.add\_subplot(1,3,1)  plt.bar(data\_gb1['Major'], data\_gb1['Money'])  plt.title('不同专业人均消费柱状图', size=20) # 绘制标题  plt.xticks(rotation=90, size=14)  ax2 = p.add\_subplot(1,3,2)  plt.bar(data\_boy['Major'], data\_boy['Money'])  plt.title('男生人均消费柱状图', size=20) # 绘制标题  plt.xticks(rotation=90, size=14)  ax3 = p.add\_subplot(1,3,3)  plt.bar(data\_girl['Major'], data\_girl['Money'])  plt.title('女生人均消费柱状图', size=20) # 绘制标题  plt.xticks(rotation=90, size=14)  plt.show() |

1. 分析不同专业不同性别学生消费特点，我们可以选择三个专业，一共有42个专业，从经管类、工科、艺术类分别选取一个代表进行研究。分别选取了18国际金融、18计算机网络、18艺术设计三个专业，如表 4所示。

表 4 专业人均刷卡频次和人均消费

|  |  |  |  |
| --- | --- | --- | --- |
|  | **18国际金融** | **18计算机网络** | **18艺术设计** |
| 专业人均消费 | 3.87 | 4.83 | 3.91 |
| 专业男生人均消费 | 5.17 | 5.16 | 4.43 |
| 专业女生人均消费 | 3.69 | 3.55 | 3.66 |

为了分析不同专业不同性别的学生消费情况，所以可以绘制柱状图，进行直观的分析，如图 20所示。

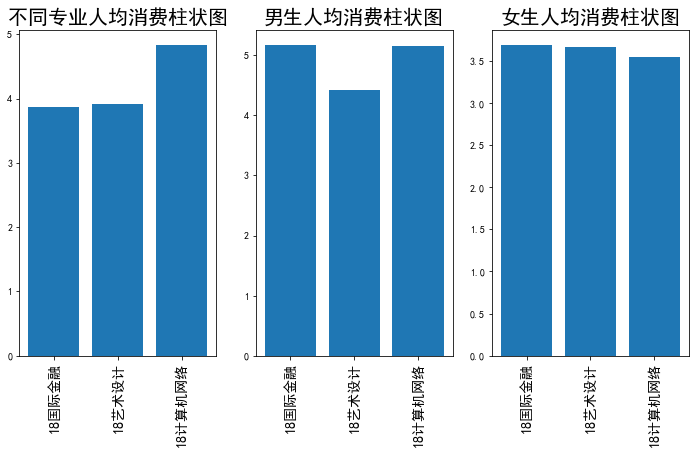


图 20 不同专业消费柱状图

消费特点如下：

1. 18计算机网络的专业人均消费额都比18艺术设计、18国际金融的高；
2. 总体来说，男生消费水平比女生高。

代码 13 选择三个专业进行分析

|  |
| --- |
| # 选择三个专业，分析不同专业不同性别学生消费特点，一共有4 2个专业，从经管类、工科、艺术类分别选取一个代表进行研究。  # 分别选取了18国际金融、18计算机网络、18艺术设计三个专业。  major = ['18国际金融','18计算机网络','18艺术设计']  data\_gb3 = data\_gb1.loc[data\_gb1['Major'].isin(major)]  data\_boy1 = data\_boy.loc[data\_boy['Major'].isin(major)]  data\_girl1 = data\_girl.loc[data\_girl['Major'].isin(major)]  # 绘制三个专业、不同性别人均消费柱状图  plt.rcParams['font.sans-serif'] = 'SimHei' # 设置中文显示  p = plt.figure(figsize = (12,6)) # 将画布设定为正方形，则绘制的饼图是正圆  ax1 = p.add\_subplot(1,3,1)  plt.bar(data\_gb3['Major'], data\_gb3['Money'])  plt.title('不同专业人均消费柱状图', size=20) # 绘制标题  plt.xticks(rotation=90, size=14)  ax2 = p.add\_subplot(1,3,2)  plt.bar(data\_boy1['Major'], data\_boy1['Money'])  plt.title('男生人均消费柱状图', size=20) # 绘制标题  plt.xticks(rotation=90, size=14)  ax3 = p.add\_subplot(1,3,3)  plt.bar(data\_girl1['Major'], data\_girl1['Money'])  plt.title('女生人均消费柱状图', size=20) # 绘制标题  plt.xticks(rotation=90, size=14)  plt.show() |

### K-Mean聚类

1. K-Means算法是一种常用的聚类算法，简单易行且适用于大中型数据量的数据聚类。本次分析将采用K-Means算法，根据学生的消费行为对学生进行分类，最终得到不同特征的学生群体，并分析不同学生群体存在的特征。构建指标求平均每餐消费额，需要合并前后刷卡时间小于10分钟的消费额，作为一次就餐金额。

根据学生的校园消费行为，构建的特征如表 5所示。

表 5 聚类特征

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **特征** | morning\_data | noon\_data | dinner\_data | month\_count |
| **含义** | 早餐平均每餐消费额 | 午餐平均每餐消费额 | 晚餐平均每餐消费额 | 月就餐次数 |

合并特征，如图 21所示。



图 21 聚类特征

代码 14构建特征

|  |
| --- |
| # 任务3.2  # 根据学生的整体校园消费行为，选择合适的特征，构建聚类模型  data\_new = pd.read\_csv("tmp\data\_new.csv" ,encoding = 'gbk')  #构建特征：1、早餐平均每餐消费额；2、午餐平均每餐消费额；3、晚餐平均每餐消费额  morning\_data = data\_new.loc[(data\_new['hour'].apply(lambda x: x in [6, 7, 8, 9,])), :]  noon\_data = data\_new.loc[(data\_new['hour'].apply(lambda x: x in [11, 12, 13])), :]  dinner\_data = data\_new.loc[(data\_new['hour'].apply(lambda x: x in [17, 18, 19, 20])), :]  morning\_data=morning\_data[['CardNo','Money\_y']].groupby(by='CardNo').mean().reset\_index()  noon\_data=noon\_data[['CardNo','Money\_y']].groupby(by='CardNo').mean().reset\_index()  dinner\_data=dinner\_data[['CardNo','Money\_y']].groupby(by='CardNo').mean().reset\_index()  # 4、月就餐次数  month\_count=data\_new[['CardNo','Money\_y']].groupby(by='CardNo').count().reset\_index()  # 合并特征  features = pd.merge(morning\_data,noon\_data,on='CardNo',how='inner')  features = pd.merge(dinner\_data,features,on='CardNo',how='inner')  features = pd.merge(month\_count,features,on='CardNo',how='inner')  features.columns=['CardNo','月就餐次数','晚餐平均每餐消费额','早餐平均每餐消费额','午餐平均每餐消费额'] # 修改列名 |

1. 根据特征数据做标准化处理后，做出轮廓系数走势图，使用轮廓系数法确定聚类的分类簇的数量，如图 22所示。

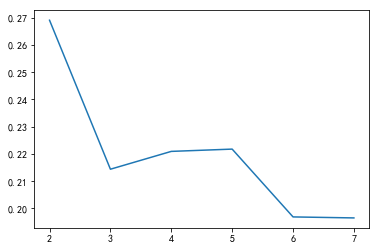


图 22 轮廓系数走势图

根据轮廓系数走势图可以看出，聚类数目为3时平均畸变程度最大，所以确定聚类的数目为3。

代码 15 轮廓系数走势图

|  |
| --- |
| from sklearn import preprocessing  from sklearn.cluster import KMeans  # 标准化处理  data\_prp = preprocessing.scale(features.iloc[:,1:])  # 轮廓系数法确定聚类数  from sklearn.metrics import silhouette\_score  silhouettteScore=[]  for i in range(2,8):  kmeans1=KMeans(n\_clusters=i,random\_state=123).fit(data\_prp )  score=silhouette\_score(data\_prp ,kmeans1.labels\_)  silhouettteScore.append(score)  plt.figure(figsize=(6,4))  plt.plot(range(2,8),silhouettteScore,linewidth=1.5,linestyle="-")  plt.show() |

1. 使用K-Means构建聚类模型。

代码 16 Kmeans模型

|  |
| --- |
| # 构建Kmeans模型  kmeans\_model = KMeans(n\_clusters= 3, max\_iter = 100)  kmeans\_model.fit(data\_prp)  fit\_label = kmeans\_model.labels\_  features['fit\_label']=fit\_label  center=kmeans\_model.cluster\_centers\_ |

1. 对聚类结果进行分群分析，绘制雷达图，如图 23所示。

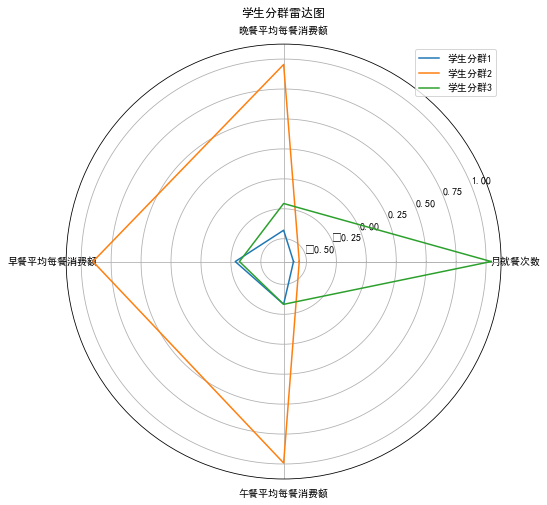


图 23 雷达图

代码 17 分群雷达图

|  |
| --- |
| # 分群雷达图  fig = plt.figure(figsize=(8,8))  plt.rcParams['font.sans-serif'] = 'SimHei' # 设置字体为SimHei显示中文  ax = fig.add\_subplot(111, polar=True) # polar参数  labels=['月就餐次数','晚餐平均每餐消费额','早餐平均每餐消费额','午餐平均每餐消费额']  angles = np.linspace(0, 2\*np.pi,4, endpoint=False)  angles = np.concatenate((angles, [angles[0]])) # 闭合  for i in range(3):  data= np.concatenate((center[i], [center[i][0]])) # 闭合  ax.plot(angles, data)  ax.set\_thetagrids(angles \* 180/np.pi, labels)  plt.legend(("学生分群1","学生分群2","学生分群3"),loc=1) # 图例  ax.set\_title("学生分群雷达图") # 绘制标题  ax.grid(True) # 显示网格  plt.show() |

### 聚类结果分析

学生群体的消费分析：

学生群体1：在食堂就餐少，所以消费额也相对少。

学生群体2：在早餐平均每餐消费额、午餐平均每餐消费额、晚餐平均每餐消费额特征上最大，月就餐次数少，说明该群体每餐消费额多，并且在食堂就餐次数少。

学生群体3：在早餐平均每餐消费额、午餐平均每餐消费额、晚餐平均每餐消费额特征上小，但是月就餐次数最大，说明该群体每餐消费额少，但是在食堂就餐次数多。

通过对学生群体的消费行为进行分析，发现学生群体3消费金额低说明该群体节俭，月就餐次数高说明除了在经常食堂消费且消费额少。由此可以判定，此类学生群体为贫困学生，对于此类群体的学生，助学金的评定可以大比例的向该类学生群体倾斜。