员工流失数据分析建模与预测案例

通过本案例涉及到数据描述性统计，数据缺失值处理，数据异常值处理等常见的数据预处理方法。涉及到数据探索分析，单样本 T 检验，交叉验证，单因素与多因素与目标属性间的关联分析，皮尔逊相关系数分析，预测建模分析等，涵盖了 Python 语言中的 Numpy，pandas，matplotlib，seaborn 库，以及第三方库 sklearn 机器学习库。通过本案例学生将学会常见的数据分析流程以及聚类分析方法的应用以及可视化分析。

本案例适合学生在学完《大数据分析与挖掘》或相关课程后的配套实践，也可以作为学生在学完大数据专业课程后的综合实践，适合应用型本科以及高职院校大数据，应用统计专业学生学习。

难易程度: ★★★　　复杂程序: ★★★★

1 案例简介

本节介绍案例目的，适用对象，时间安排，预备知识，硬件要求，软件工具，数据集，案例任务,案例背景，基本思路等内容。

1.1 案例目的

通过实践本次案例，学生可以达到以下目的:

- 数据导入与查看
- 学会应用 Numpy 库以及常见的操作方法
- 学会应用 pandas 库以及常见的操作方法
- 学会通过 matplotlib 库绘制折线图等常见可视化图形
- 学会通过可视化图形进行可视化分析
- 学会数据建模原理及方法的应用
- 学会数据挖掘原理及常见的分类方法（决策树算法）
- 学会通过 sklearn 库应用到聚类分析
- 熟悉缺失值数据处理，异常值数据处理，重复值数据处理等常见的数据预处理方法
- 学会探索性分析数据的要素以及交叉验证。

- 学会假设检验，会使用 T 检验证明假设。
- 熟悉单因素影响分析
- 熟悉多因素影响分析
- 熟悉皮尔逊相关系数分析
- 学会特征重要性分析
- 建立决策树模型

1.2适用对象

- 高校（高职）教师
- 高校（高职）学生
- 大数据学习者（有一定的大数据分析基础）
- 数据挖掘分析者

1.3时间安排

本案例适合学生在学完《大数据分析与挖掘》或相关课程后的配套的“大作业”实践，也可以作为学生在学完大数据专业课程后的综合实践案例之一，建议放在第 6 或第 7 学期，也可以作为学生寒暑假期大数据实习基础案例。学时 16 学时（2 天）。

1.4 预备知识

需要学生熟悉 Python 语言，基本语法，常见语义等；特别是常见库中的 Numpy，pandas，matplotlib 的方法应用；熟练掌握 Jupyter notbook 的操作方法；熟悉如何安装第三方库，熟悉第三方库(sklearn 机器学习库)方法应用；掌握一定的数据挖掘方法，特别是聚类分析方法的原理与应用，掌握聚类分析的常见算法；决策树原理与算法应用，了解常见的数据预处理的时机与方法；熟悉常见的数据建模方法，有一定的数据建模应用分析能力；特别强调应用统计相关知识，包括但不限于 T 检验，皮尔逊相关系数分析，交叉验证分析；需要了解一定的人工智能基础知识。

1.5 硬件要求

本案例单机要完成，无需要联网也可以操作（如果将数据集部署至私有云，需要局

域网访问数据集)。单机上比较流畅完成本案例，建议计算机硬盘配备 500G 以上硬盘，4G 以上内存。

1.6 软件工具

Anaconda3（64-bit）+Jupyter notbook

Python 3.7

1.7 数据集

该数据集包含 14999 个样本以及 10 个特征，通过现有员工已经是否离职的数据，建立模型预测有可能离职的员工.

1.8 案例任务

本案例需要完成以下实验任务

- 安装 Anaconda3（64-bit）+Jupyter notbook
- 安装 Python 3.7
- 安装第三方库 sklearn
- 数据准备及认识数据
- 实现数据加载
- 实现缺失值分析处理
- 实现异常值分析处理
- 实现描述统计值处理
- 实现热力图分析属性交叉验证
- 实现 T 检验分析
- 实现单因素影响分析
- 实现多因素影响分析
- 实现聚类分群处理
- 实现特征重要性程度分析
- 实现皮尔逊相关系数分析
- 实现决策树分类器及结果分析

1.9案例背景

员工流失（减员）是组织机构的一项主要成本， 在许多组织机构中，预测人事变动是人力资源（HR）的核心需求。这也是每一家企业都要面对的问题，优秀人才的离职会对公司的运营产生巨大的负面影响。一家大型企业希望能搞清楚为什么企业内的一些优秀员工会选择离职，此外，还希望能利用收集的员工数据建立一个预测模型以便提前预测哪些优秀员工有离职倾向。

通过员工离职情况数据来分析数据集内哪些特征变量会对员工离职造成影响，以及其背后可能的原因，并利用逻辑回归和决策树算法构建预测模型。员工对公司的满意度、岗位类别、以及优秀人才能否得到满意的薪酬和晋升空间，都是影响离职与否的主要因素。

1.10 案例目标

●了解什么因素对员工流动率影响最大，并创建一个能够预测某个员工是否会离开公司的模型。

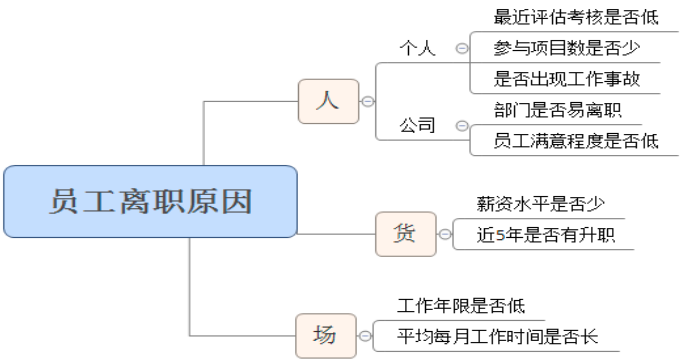
●员工对公司满意度平均水平如何？员工的最新考核情况又是如何？员工所参加项目数是怎样？员工平均每月工作时长以及平均工作年限分别是多少？

●当前离职率是多少？工作事故发生率？过去 5 年升职率？薪资水平又如何？共有多少种岗位？

●是否离职和其他 9 个特征的关系如何（相关分析）？

●根据现有数据，是否可以对某个员工是否离职进行预测（建模）？

1.11 基本思路



上图根据数据集的数据字段信息提炼员工离职原因思维导图的相关逻辑。根据相关逻辑图，在预处理完数据的基础上。

●第一主要做相关描述性统计分析，导出离职率等相关信息；

●第二通过假设检验分析，进行 T 检验证明离职人群与未离职人群间的差异性分析。

●第三针对不同的因素进行影响分析，可视化直观的方式得出不同因素对离职的影响程度。

●第四针对不同因素间做相关性分析，热力图得出不同因素间的影响程度，并做交叉检验。

●最后针对特征分析，分析出不同特征的重要程度，通过决策树分类器进行特征重要程度可视化分析。

2.实验步骤

2.1 实验步骤概述

步骤零：实验环境准备

步骤一：数据准备

步骤二：数据预处理

步骤三：数据探索性分析

步骤四：数据预测性分析

下面给出每个步骤所需要的具体实验内容。

步骤零：（1）安装 Anaconda3（64-bit）+Jupyter notbook

（2）安装 Python 3.7

（3）安装第三方库

【备注】：本手册默认环境已经预制，如果需要安装请具体参考软件安装手册。

步骤一：（1）数据准备与解读

（2）新建 notbook 文件

（3）数据加载

（4）导入与查看数据

（5）进行描述性统计

（6）进行数据类型查看

步骤二：（1）缺失值处理

(2) 重命名列字段

（3）异常值处理

（4）重复值处理

步骤三：（1）离职率分析

（2）交叉验证平均因素值

（3）单样本 t 检验

（4）t 检验分位数

（5）不同因素对离职影响分析

步骤四：（1）kmeans 员工离职聚类分析

（2）皮尔逊相关系数分析

（3）策树分类模型分析特征重要程度

2.2 步骤一：数据准备

2.2.1 数据解读

数据集名称：HR_comma_sep.csv

数据来源：数据集源自于 kaggle 平台用户分享，基于证书 CC0: Public Domain 发布。

【备注】：*参考* https://www.kaggle.com/jiangzuo/hr-comma-sep

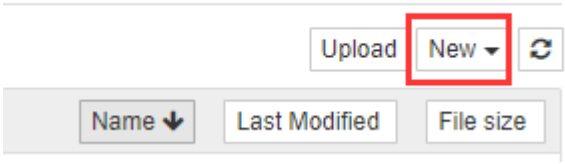
数据描述：汇集了 14999 条员工的信息数据统计，包括主原因 left(0 或 1)，企业因素如（部门 sales)、员工行为相关信息（项目数、平均每月工作时长、工作年限、是否升值，工资水平等)，以及工作相关因素（员工满意程度、最新绩效评估、是否出现工作事故)，总计 10 个列字段，字段描述信息见下表 1。

表 1：员工离职表字段含义

数据项	数据含义
satisfaction_level	满意度
last_evaluation	员工评价
number_project	项目统计
average_montly_hours	月平均工作时间
time_spend_company	年工作时间
Work_accident	工伤
left	离职
promotion_last_5years	晋升
sales	部门
salary	工资

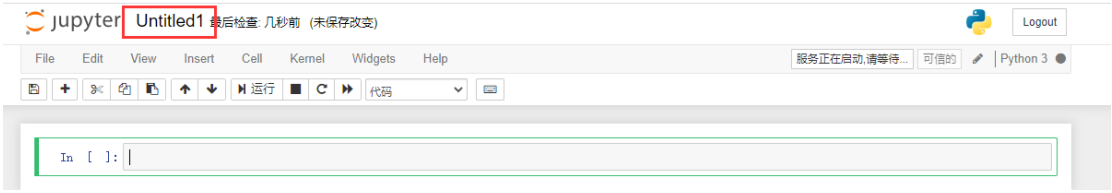
2.2.2 新建 notbook 文件

（1）在 Jupyter notbook 界面，选择左上角“new-Python3”（如下图）。

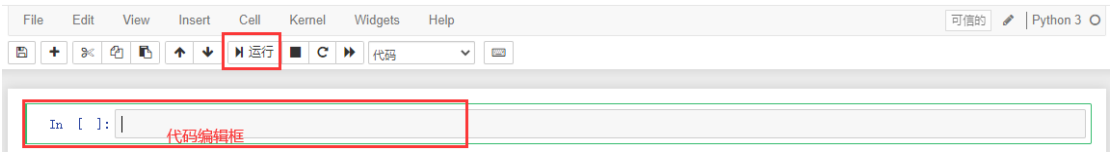


（2）创建完成后如下图，点击“Untitled1”重命名文件名为“员工流失数据分析建模与预测“。

”

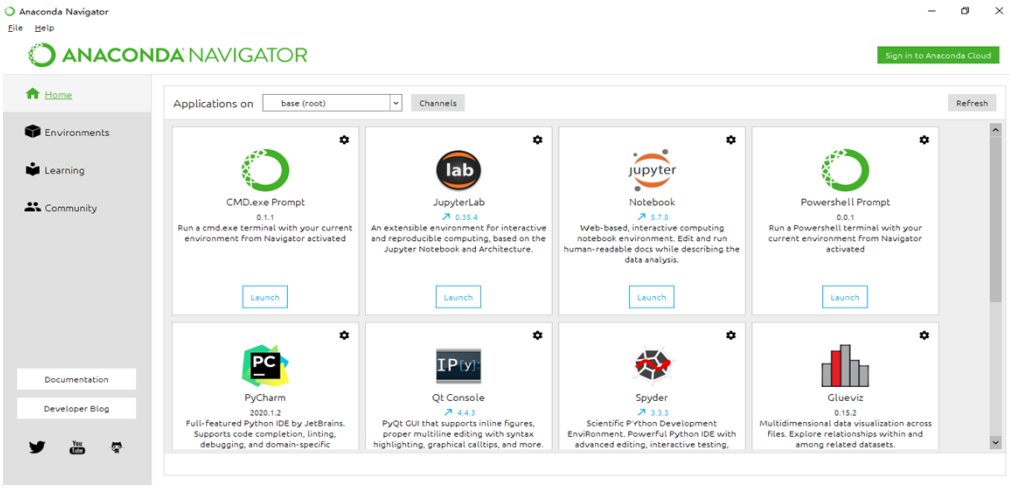


（3）接下来就可以编写代码，编写代码框如下图，编写完成后按“运行”按钮运行。

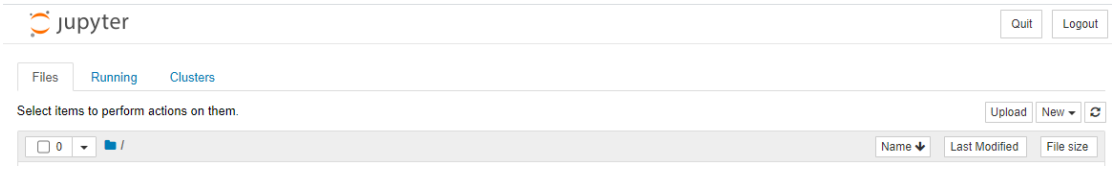


2.2.3 数据加载

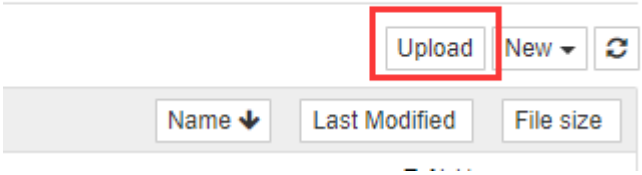
（1）：打开软件 Anaconda3，结果如下



（2）：选择 Jupyter notbook 点击“Launch”进入 Jupyter 的 Home 界面。结果如下



（3）：点击界面右上角“upload”，进入上传数据集界面。



（4）：选择 HR_comma_sep.csv，确定后点击上传，结果如下。

（5）：数据加载完成

【备注】：数据加载不一定需要，建议数据集不大加载数据，这样每次调用数据集的时候，只需要写其相对路径即可，当然如不加载数据，读取数据时就需要写数据保存的绝对路径；如果数据集较大（一般 5G 以上）建议保存当地服务器，为了减少错误，这样使用绝对路径较好。

2.2.4 导入与查看数据

（1）导入数据操作和可视化表示所需的模块

代码如下：

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt   #提供一个类似 matlab 的绘图框架。
import matplotlib as mpt
import seaborn as sns
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')   #过滤警告信息
```

读取数据集存储到名为“df”的 dataframe 中

代码如下：

```
df= pd.read_csv('HR_comma_sep.csv', index_col=None)   #读取数据集
df.head()
```

结果如下表 2

表 2：数据集（HR_comma_sep.csv）

	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company	Work_accident	left	promotion_last_5years	sales	salary
0	0.38	0.53	2	157	3	0	1	0	sales	low
1	0.80	0.86	5	262	6	0	1	0	sales	medium
2	0.11	0.88	7	272	4	0	1	0	sales	medium
3	0.72	0.87	5	223	5	0	1	0	sales	low
4	0.37	0.52	2	159	3	0	1	0	sales	low

2.2.5 进行描述性统计

这里主要针对每列做常规统计分析，分别有平均值，方差，最小值 ，最大值，总和，中位数等。代码如下：

```
df.describe()   #描述性统计
```

结果如下表 3：

表 3;描述性统计表

	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company	Work_accident	left	promotion_last_5years
count	14999.000000	14999.000000	14999.000000	14999.000000	14999.000000	14999.000000	14999.000000	14999.000000
mean	0.612834	0.716102	3.803054	201.050337	3.498233	0.144610	0.238083	0.021268
std	0.248631	0.171169	1.232592	49.943099	1.460136	0.351719	0.425924	0.144281
min	0.090000	0.360000	2.000000	96.000000	2.000000	0.000000	0.000000	0.000000
25%	0.440000	0.560000	3.000000	156.000000	3.000000	0.000000	0.000000	0.000000
50%	0.640000	0.720000	4.000000	200.000000	3.000000	0.000000	0.000000	0.000000
75%	0.820000	0.870000	5.000000	245.000000	4.000000	0.000000	0.000000	0.000000
max	1.000000	1.000000	7.000000	310.000000	10.000000	1.000000	1.000000	1.000000

从以上统计结果可以看出

●员工对公司满意度平均水平如何？员工的最新考核情况又是如何？员工所参加项目数是怎样？员工平均每月工作时长以及平均工作年限分别是多少？

●员工对公司的满意度：范围 0.09~1，中位数 0.640，均值 0.613，总体来说员工对公司较满意。

●最新考核评估：范围 0.36~1，中位数 0.720，均值 0.716，员工考核平均水平中等偏上。

●项目数：范围 2~7 个，中位数 4，均值 3.8，平均参加项目数为 4 个。

●平均每月工作时长：范围 96~310 小时，中位数 200，均值 201。

●工作年限：范围 2~10 年，中位数 3，均值 3.5。

●当前离职率是多少？工作事故发生率？过去 5 年升职率？薪资水平又如何？共有多少种岗位？

●当前离职率为 23.81%。

●工作事故发生率 14.46%。

●过去 5 年升职率 2.13%。

●薪资水平共有 3 个等级，最多的是低等，多达 7316 人。员工岗位有 10 种，其中最多的是销售，多达 4140 人。

2.2.6 进行数据类型查看

代码如下：

```
df.info()    #前两个特征为 64 位浮点型，后两个为字符型，其余为 64 位整型，且均无缺失值。
```

结果如下：

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 14999 entries, 0 to 14998
Data columns (total 10 columns):
satisfaction_level       14999 non-null float64
last_evaluation          14999 non-null float64
number_project           14999 non-null int64
average_montly_hours     14999 non-null int64
time_spend_company       14999 non-null int64
Work_accident            14999 non-null int64
left                     14999 non-null int64
promotion_last_5years    14999 non-null int64
sales                    14999 non-null object
salary                   14999 non-null object
dtypes: float64(2), int64(6), object(2)
memory usage: 1.1+ MB
```

2.3 步骤二：数据预处理

2.3.1 缺失值处理

检查数据集中是否有遗漏值

代码如下:

```
df.isnull().any()  #查看是否有遗漏值，True 代表有缺失值，False 代表没有。
```

结果如下:

```
satisfaction_level       False
last_evaluation          False
number_project           False
average_montly_hours     False
time_spend_company       False
Work_accident            False
left                     False
promotion_last_5years    False
sales                    False
salary                   False
dtype: bool
```

发现所有列字段没有缺失值。

2.3.2 重命名列字段

重新命名列名，以便获得更好的可读性

代码如下：

```
df = df.rename(columns={'satisfaction_level': '满意度',
                        'last_evaluation': '评价',
                        'number_project': '项目统计',
                        'average_montly_hours': '月平均工作时间',
                        'time_spend_company': '年工作时间',
                        'Work_accident': '工伤',
                        'promotion_last_5years': '晋升',
                        'sales' : '部门',
                        'left' : '离职',
                        'salary' : '工资'
                        })
# 将离职那一列移到最前面
front = df['离职']
df.drop(labels=['离职'], axis=1,inplace = True)
df.insert(0, '离职', front)
df.head()
```

结果如下表 4：

表 4：列重命名后数据集

	离职	满意度	评价	项目统计	月平均工作时间	年工作时间	工伤	晋升	部门	工资
0	1	0.38	0.53	2	157	3	0	0	sales	low
1	1	0.80	0.86	5	262	6	0	0	sales	medium
2	1	0.11	0.88	7	272	4	0	0	sales	medium
3	1	0.72	0.87	5	223	5	0	0	sales	low
4	1	0.37	0.52	2	159	3	0	0	sales	low

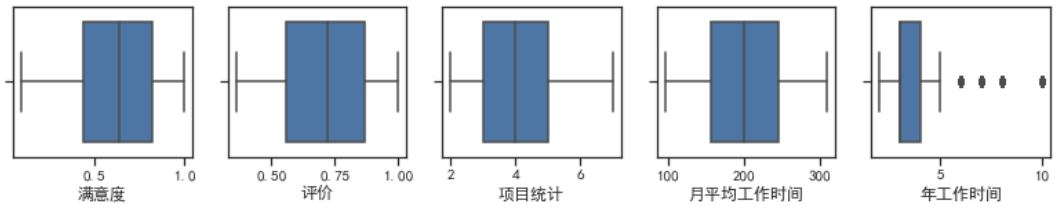
2.3.3 异常值处理

利用箱线图查看异常值

代码如下：

```
fig, ax = plt.subplots(1,5, figsize=(12, 2))   #产生 1*5 的箱线图
sns.boxplot(x=df.columns[1], data=df, ax=ax[0])   #绘制第 1 列（满意度）箱线图
sns.boxplot(x=df.columns[2], data=df, ax=ax[1])   #绘制第 2 列（评价）箱线图
sns.boxplot(x=df.columns[3], data=df, ax=ax[2])   #绘制第 3 列（项目统计）箱线图
sns.boxplot(x=df.columns[4], data=df, ax=ax[3])   #绘制第 4 列（月平均工作）箱线图
sns.boxplot(x=df.columns[5], data=df, ax=ax[4])   #绘制第 5 列（年工作时间）箱线图
```

结果如下图



除了工作年限外，其他均无异常值。大部分工作年限都在五年以下，"新鲜血液"较多，侧面反映年轻人较多。

2.3.4 重复值处理

数据不存在要哪个特征是唯一值的，所以不需要进行重复值处理。

2.4 步骤三：数据探索性分析

2.4.1 离职率分析

代码如下：

```
离职_rate = df.离职.value_counts() / len(df)   #计算离职率与未离职率
离职_rate
```

结果如下：

```
0    0.761917
1    0.238083
Name: 离职, dtype: float64
```

说明该公司的人员离职率约为 24%

似乎有 76%的员工留下了，24%的员工离开了。当执行交叉验证时，保持这个周转率是很重要的。

2.4.2 交叉验证平均因素值

①代码如下:

```
离职_Summary = df.groupby('离职')  #交叉验证平均因素值
离职_Summary.mean()
```

结果如下表 5:

表 5：分类统计各列平均值

离职	满意度	评价	项目统计	月平均工作时间	年工作时间	工伤	晋升
0	0.666810	0.715473	3.786664	199.060203	3.380032	0.175009	0.026251
1	0.440098	0.718113	3.855503	207.419210	3.876505	0.047326	0.005321

员工满意度平均为 0.61（可通过代码”df.满意度.mean()”获取），离职人员较未离职人员明显低于员工满意度平均值。

②代码将进行相关矩阵与热力图绘制，反映出列间相关性分析

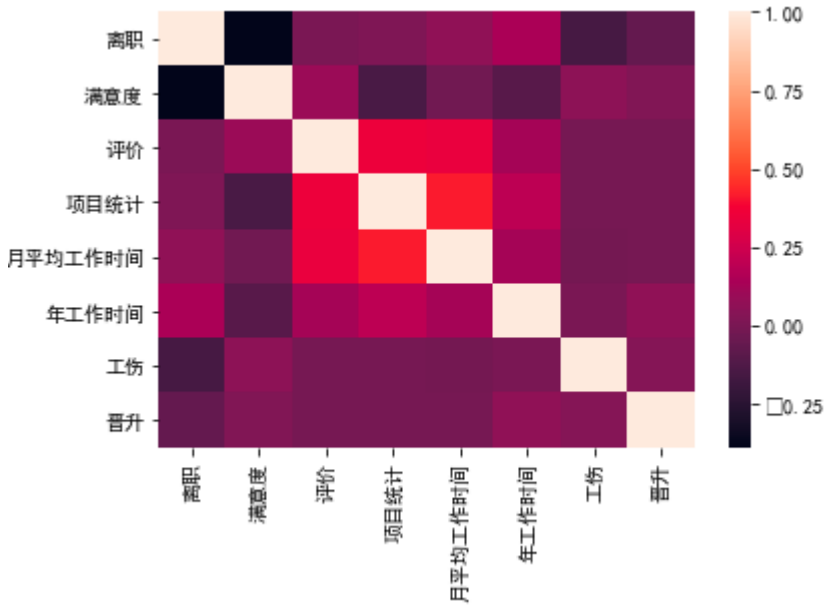
代码如下:

```
corr = df.corr()  #因素间相关性分析
corr = (corr)
sns.heatmap(corr,xticklabels=corr.columns.values,yticklabels=corr.columns.values) #绘制热力图
corr
```

结果如下表 6:

表 6：列间相关性表及热力图

	离职	满意度	评价	项目统计	月平均工作时间	年工作时间	工伤	晋升
离职	1.000000	-0.388375	0.006567	0.023787	0.071287	0.144822	-0.154622	-0.061788
满意度	-0.388375	1.000000	0.105021	-0.142970	-0.020048	-0.100866	0.058697	0.025605
评价	0.006567	0.105021	1.000000	0.349333	0.339742	0.131591	-0.007104	-0.008684
项目统计	0.023787	-0.142970	0.349333	1.000000	0.417211	0.196786	-0.004741	-0.006064
月平均工作时间	0.071287	-0.020048	0.339742	0.417211	1.000000	0.127755	-0.010143	-0.003544
年工作时间	0.144822	-0.100866	0.131591	0.196786	0.127755	1.000000	0.002120	0.067433
工伤	-0.154622	0.058697	-0.007104	-0.004741	-0.010143	0.002120	1.000000	0.039245
晋升	-0.061788	0.025605	-0.008684	-0.006064	-0.003544	0.067433	0.039245	1.000000



【备 注】：实验过程中文字部分可能乱码无法显示，但是不影响结果，可以使用浏览器“兼容模式“刷新加载后查看。

从热图中可以看出，项目统计、月平均工时和员工评价之间存在正(+)相关关系。这可能意味着花更多时间做更多项目的员工会得到更高的评价。

对于负(-)关系，离职与满意度高度相关。我的假设是，当人们不太满意的时候，他们往往会离开公司更多。

2.4.3 单样本 t 检验（测量满意度）

①假设检验

单样本 t 检验检验样本均值是否与总体均值不同。由于满意度与离职率的因变量相关性最高，我们来测试一下，有离职的员工的平均满意度是否与没有离职的员工不同。

假设检验:有离职的员工和没有离职的员工之间的满意度水平是否存在显著差异?

●零假设(H0: pTS = pES)零假设是，完成离职的员工与未离职的员工满意度没有差异。

●替代假设:(HA: pTS != pES)替代假设是，完成离职的员工和未离职的员工在满意度上存在差异。

代码如下:

```
emp_population = df['满意度'][df['离职'] == 0].mean()      #计算未离职平均满意度值
emp_离职_satisfaction = df[df['离职']==1]['满意度'].mean()  #计算离职平均满意度值
print( '未离职员工平均满意度是: ' + str(emp_population))
print( '离职员工平均满意度是: ' + str(emp_离职_satisfaction) )
```

实验结果如下:

```
未离职员工平均满意度是: 0.666809590479516
离职员工平均满意度是: 0.44009801176140917
```

②执行 t 检验

我们在 95%置信水平上进行 t 检验，看看它是否正确地拒绝了零假设， 我们可以使用 stats.ttest_1samp()函数

代码如下:

```
import scipy.stats as stats
stats.ttest_1samp(a= df[df['离职']==1]['满意度'], # t 检验，是否正确的拒绝零假设
                  popmean = emp_population)
```

结果如下:

```
Ttest_1sampResult(statistic=-51.3303486754725, pvalue=0.0)
```

t 检验结果

- 试验结果表明，试验统计量“t”= -51.33。这个检验统计量告诉我们样本均值偏离零假设的程度。如果 t 统计量位于与置信水平和自由度对应的 t 分布分位数之外，我们拒绝零

假设。我们可以用 stats.t.ppf()来检查分位数:

2.4.4 t 检验分位数

代码如下:

```
degree_freedom = len(df[df['离职']==1])
LQ = stats.t.ppf(0.025,degree_freedom)  # 左四分之一分位区间
RQ = stats.t.ppf(0.975,degree_freedom)  # 右四分之一分位区间
print ('t 分布左四分之一分位区间为: ' + str(LQ))
print ('t 分布右四分之一分位数范围为: ' + str(RQ))
```

结果如下:

```
t分布左四分之一分位区间为：-1.9606285215955626
t分布右四分之一分位数范围为：1.9606285215955621
```

●单样本 t 检验总结

t 检验= -51.33 | p 值= 0.000_ |拒绝零假设

●拒绝零假设，因为:

①t 值在分位数之外

②p 值低于 5%的置信水平

2.4.5 不同因素对离职影响分析

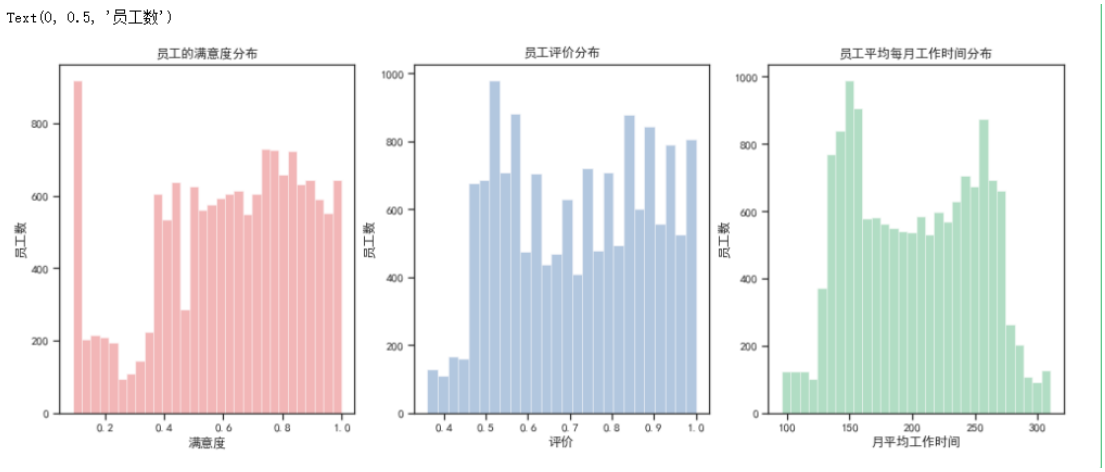
（1）不同因素下分类下员工数量分布

本案例统计出相对比较重要的指标下员工数量分布，分别是员工满意度分布，员工评价分布，员工月平均工作时长分布

代码如下：

```
f, axes = plt.subplots(ncols=3, figsize=(15, 6))
# Graph Employee Satisfaction
sns.distplot(df.满意度, kde=False, color="r", ax=axes[0]).set_title('员工的满意度分布')
 #员工满意度的员工数量分布
axes[0].set_ylabel('员工数')
sns.distplot(df.评价, kde=False, color="b", ax=axes[1]).set_title('员工评价分布')
 #员工评价的员工数量分布
axes[1].set_ylabel('员工数')
sns.distplot(df.月平均工作时间, kde=False, color="g", ax=axes[2]).set_title('员工平均每月工作时间分布')
#月平均工作时间员工数量分布
axes[2].set_ylabel('员工数')
```

结果如下图：



结果组图反映：

●满意度——低满意度和高满意度的员工会有一个巨大的峰值

●评价-低评价(小于 0.6)和高评价(大于 0.8)的员工存在双峰

●平均工作时间——还有另一种双峰分布，员工的平均月工作时间(少于 150 小时&超过 250 小时)有高低之分

●评估图和月平均小时数图都有相似的分布

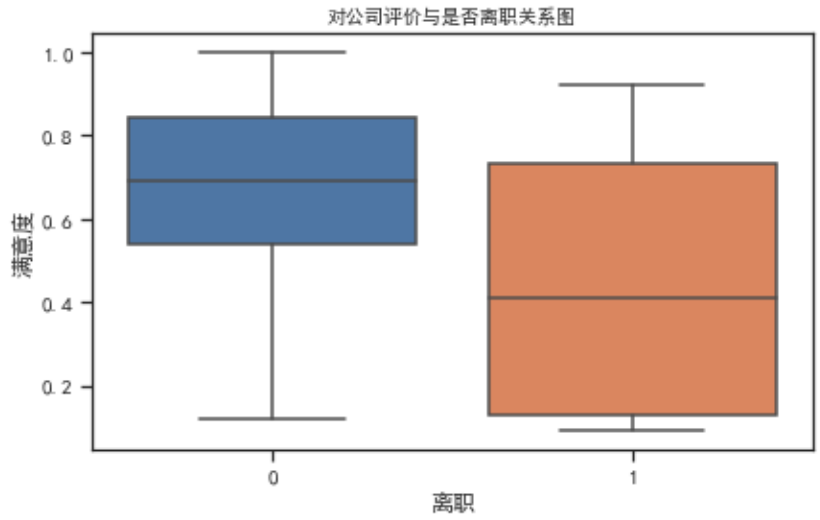
●平均月工作时间较低的员工得到的评估较低，反之亦然。回顾一下相关矩阵，评估和平均工作时间之间的高相关性确实支持这个发现

（2）满意度-离职

代码如下：

```
sns.set(style="ticks", color_codes=True,font="simhei")
sns.boxplot(x=df['离职'],y=df['满意度'], data=df)  #绘制箱线图
plt.title('对公司评价与是否离职关系图',fontsize=10)
plt.show()
```

结果如下图



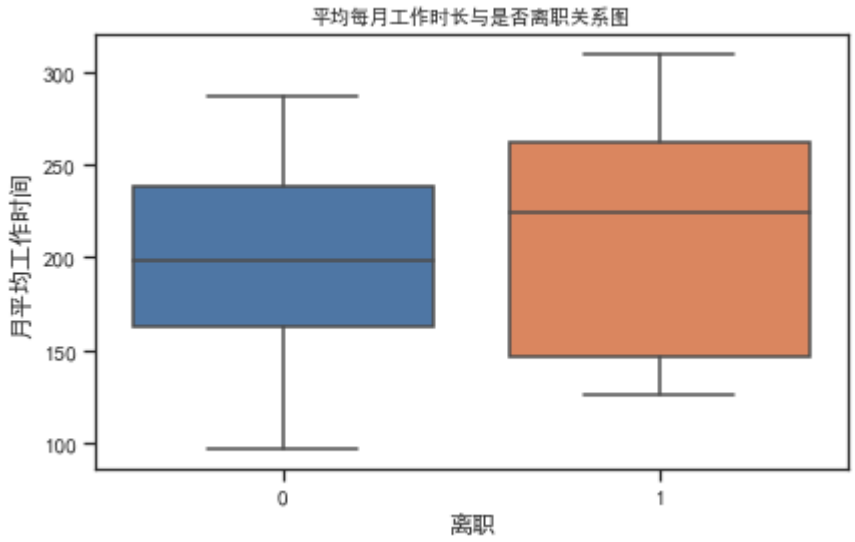
图中反映就中位数而言，离职人员对公司满意度相对较低，且离职人员对公司满意度整体波动较大．另外离职人员中没有满意度为 1 的评价，大部分都在 0.4 左右，最低在 0.1 左右

（3）月工作时长-离职

代码如下：

```
sns.set(style="ticks", color_codes=True,font="simhei")
sns.boxplot(x=df['离职'],y=df['月平均工作时间'], data=df)
plt.title('平均每月工作时长与是否离职关系图',fontsize=10)
plt.show()
```

结果如下图



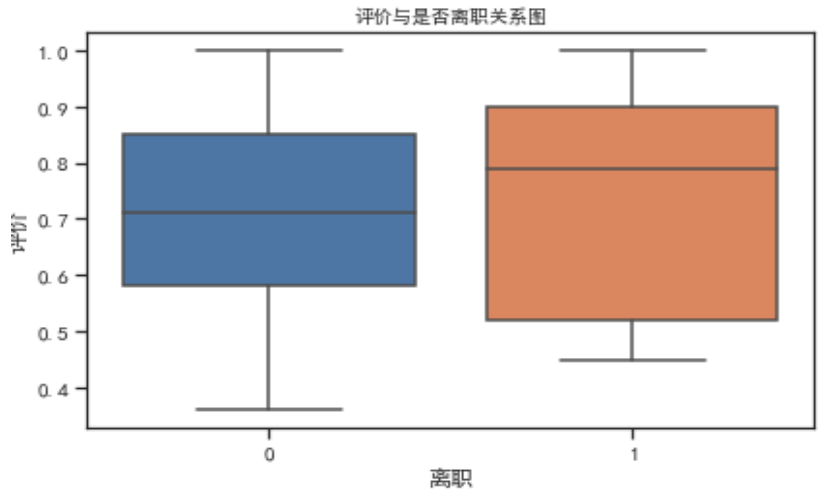
中位数而言，离职人员工作时长比较多，且离职人员工作时长整体波动较大．另外离职人员最高工作时长比在职人员高，最低工作时长比其低。

（4）评价-离职

代码如下:

```
sns.set(style="ticks", color_codes=True,font="simhei")
sns.boxplot(x=df['离职'],y=df['评价'], data=df)    #绘制评价-离职箱线图
plt.title('评价与是否离职关系图',fontsize=10)
plt.show()
```

结果如下图



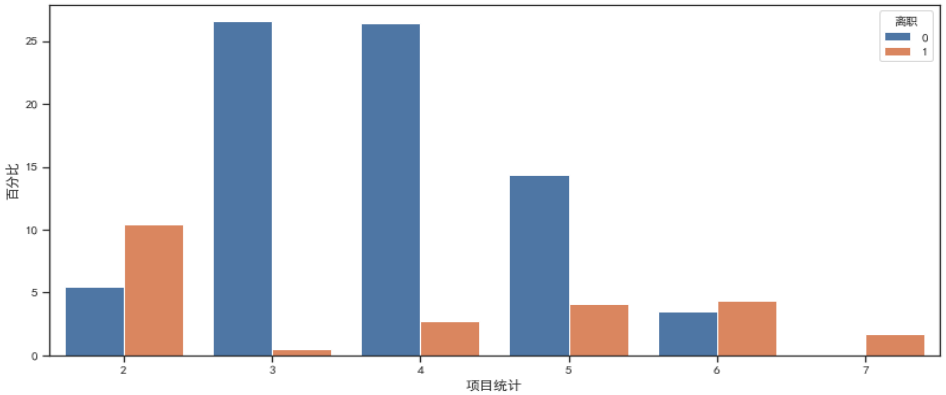
就中位数而言，离职人员考核分析比较高，且离职人员工作时长整体波动较大．另外离职人员最高考核分数比在职人员高，最低考核分数比其低。

（5）项目数-离职

代码如下：

```
plt.figure(figsize=(12,6))
#绘制项目数-离职柱状图
ax = sns.barplot(x="项目统计", y="项目统计", hue="离职", data=df, estimator=lambda x:
len(x) / len(df) * 100)
ax.set(ylabel="百分比")
```

结果如下图



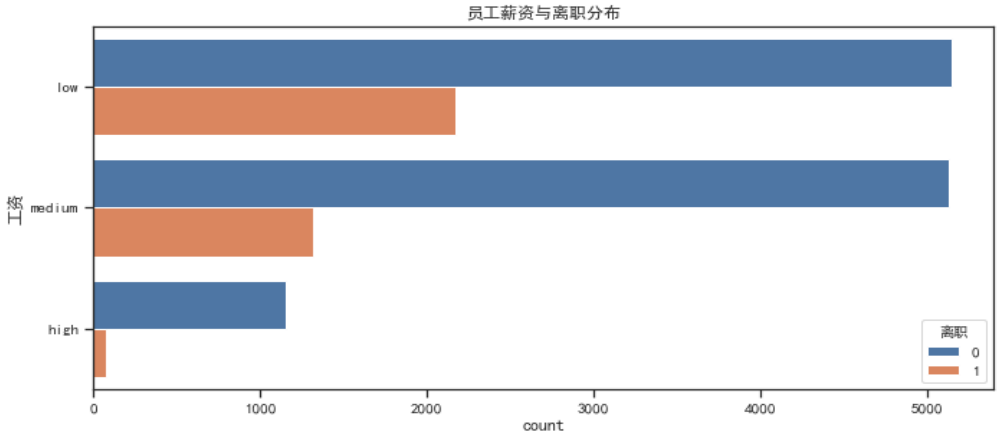
离职率随着项目数的增多而增大,但 2 个项目数是特例，可能是项目数为 2 的这部分人工作能力不被认可，离职人数就多了。

（6）工资-离职

代码如下：

```
f, ax = plt.subplots(figsize=(10, 5))
sns.countplot(y="工资", hue='离职', data=df).set_title('员工薪资与离职分布')  #绘制工资-离职柱状图
```

结果如下图



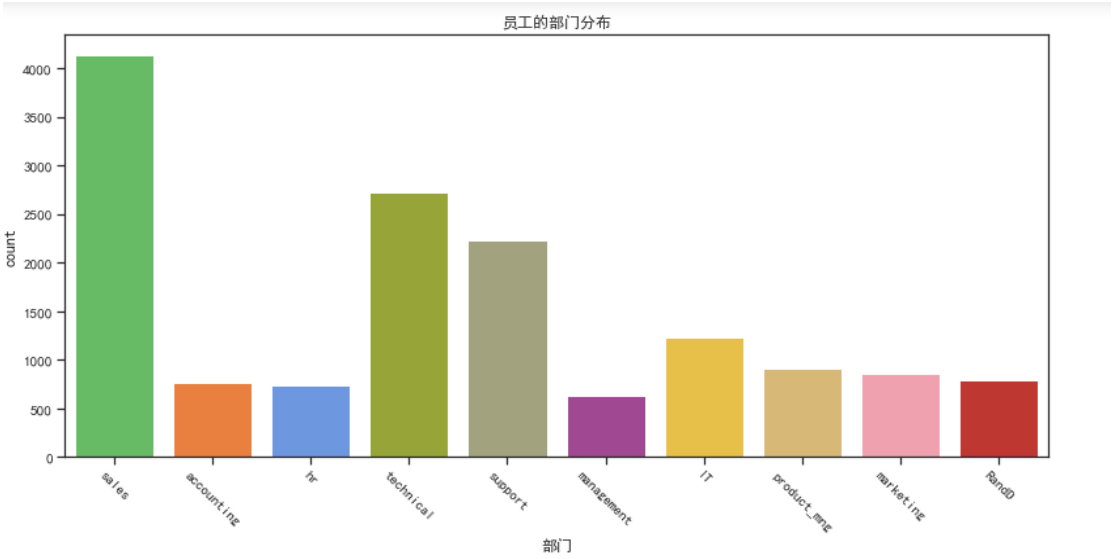
薪资水平为 low 级别的离职率最高，high 的离职率最低。两者成正比关系。

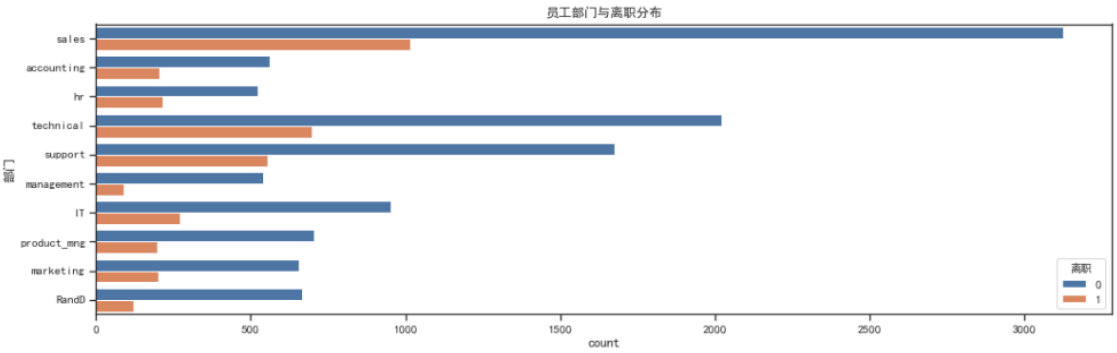
（7）部门-离职

代码如下：

```
plt.figure(figsize=(12,6))
color_types = ['#78C850','#F08030','#6890F0','#A8B820','#A8A878','#A040A0','#F8D030','#E0C068','#EE99AC','#C03028','#F85888','#B8A038','#705898','#98D8D8','#7038F8']  # 调色板
sns.countplot(x='部门', data=df, palette=color_types).set_title('员工的部门分布');
plt.xticks(rotation=-45)  #员工部门分布表
f, ax = plt.subplots(figsize=(15, 5))
sns.countplot(y="部门", hue='离职', data=df).set_title('员工部门与离职分布'); #不同部门离职员工分析
```

结果如下图：





销售、技术和支持部门是员工离职最高的 3 个部门

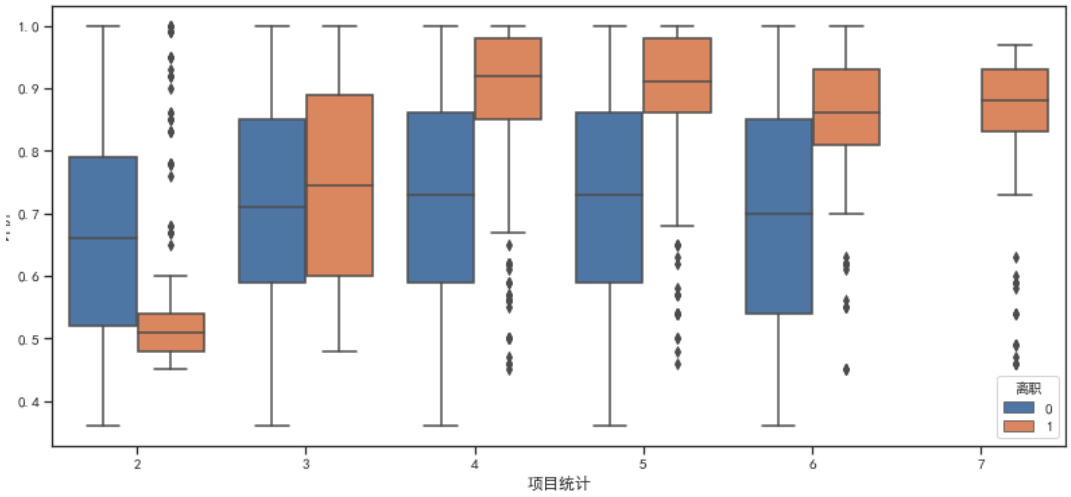
管理部门的离职最少

（8）项目数量-评价

代码如下：

```
import seaborn as sns
plt.figure(figsize=(12,6))
sns.boxplot(x="项目统计", y="评价", hue="离职", data=df) #不同项目数据，不同评价离职分布
```

结果如下图



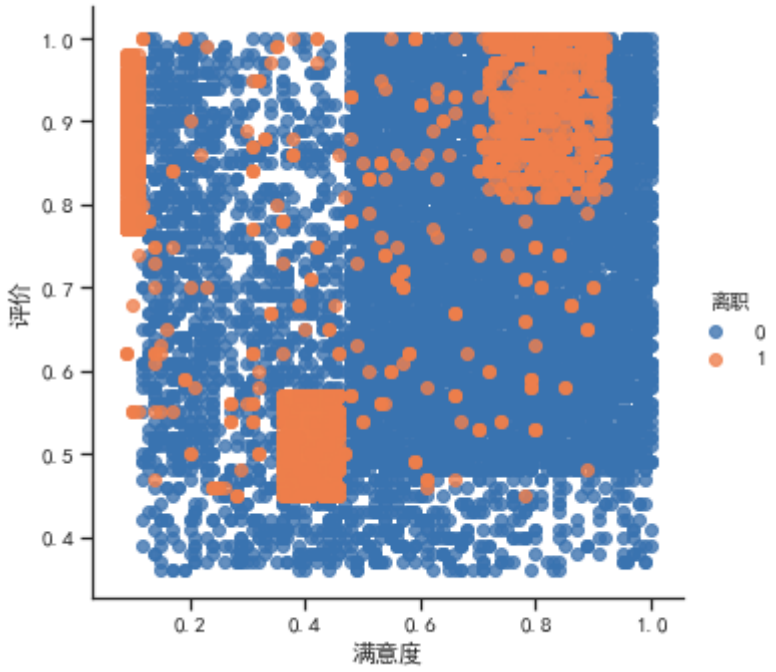
对于在离职小组中完成更多项目的员工的评价有所增加。但是，对于没有离职的团队来说，尽管项目数量有所增加，但这里的员工的评价得分是一致的。

(9)满意度-评价

代码如下：

```
plt.figure(figsize=(12,6))
sns.lmplot(x='满意度', y='评价', data=df,fit_reg=False,hue='离职')    # 绘制回归图形
```

结果如下图



离开公司的员工有 3 个不同的群体:

第 1 组(勤奋且悲伤的员工):满意度低于 0.2，评价大于 0.75。这可能是一个很好的迹象，表明离开公司的员工都是好员工，但对自己的工作感到很糟糕。

第二组(不良员工和悲伤员工):满意度在 0.35~0.45 之间，评价低于 0.58。这可能被看作是员工的评价很差，在工作中感觉很差。

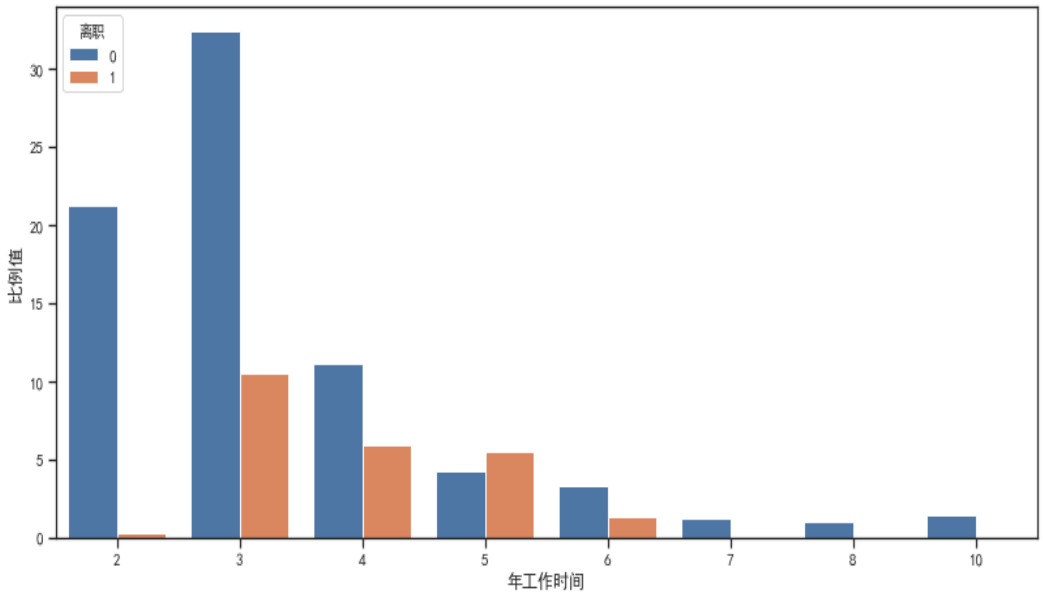
第三组(勤奋并快乐的员工):满意度在 0.7~1.0 之间，评价大于 0.8。这可能意味着这个集群中的员工是“理想的”。他们热爱自己的工作，并因他们的表现受到高度评价。

（10）年工作时间-离职

代码如下:

```
plt.figure(figsize=(12,6))
ax = sns.barplot(x="年工作时间", y="年工作时间", hue="离职", data=df, estimator=lambda x: len(x) / len(df) * 100) #工作年限离职比例
```

结果如下图



超过一半的 5 年的员工离开了公司。

5 年以上的员工应该被高度关注。

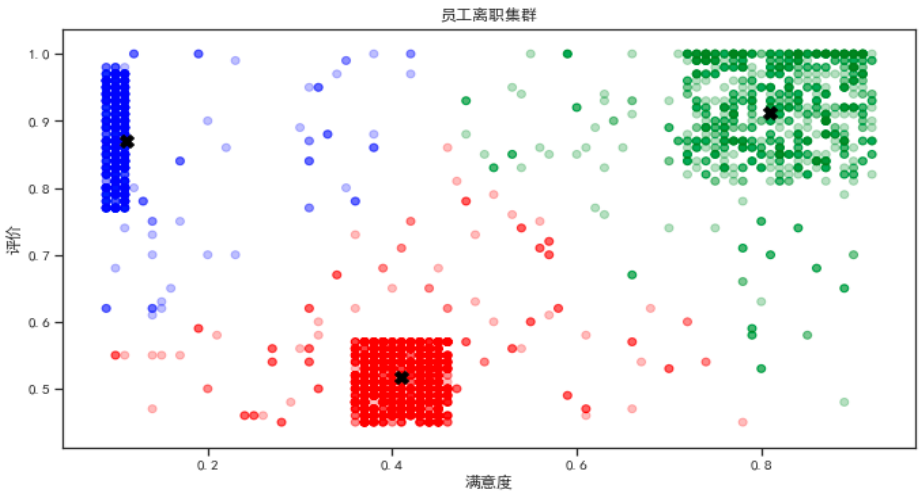
2.5 步骤四：数据预测性分析

2.5.1 kmeans 员工离职聚类分析

代码如下:

```
# Import KMeans Model
from sklearn.cluster import KMeans   #导入相关的包
#  绘制并创建 3 个员工流失集群
kmeans = KMeans(n_clusters=3,random_state=2)
kmeans.fit(df[df.离职==1][["满意度","评价"]])
kmeans_colors = ['green' if c == 0 else 'blue' if c == 2 else 'red' for c in kmeans.labels_]
fig = plt.figure(figsize=(10, 6))
plt.scatter(x="满意度",y="评价", data=df[df.离职==1],
                    alpha=0.25,color = kmeans_colors)
plt.xlabel("满意度")
plt.ylabel("评价")
plt.scatter(x=kmeans.cluster_centers_[:,0],y=kmeans.cluster_centers_[:,1],color="black",m
arker="X",s=100)
plt.title("员工离职集群")
plt.show()
```

结果如下图



结果如下:

集群 1(蓝色):工作努力、情绪低落的员工

集群 2(红色):坏的和悲伤的员工

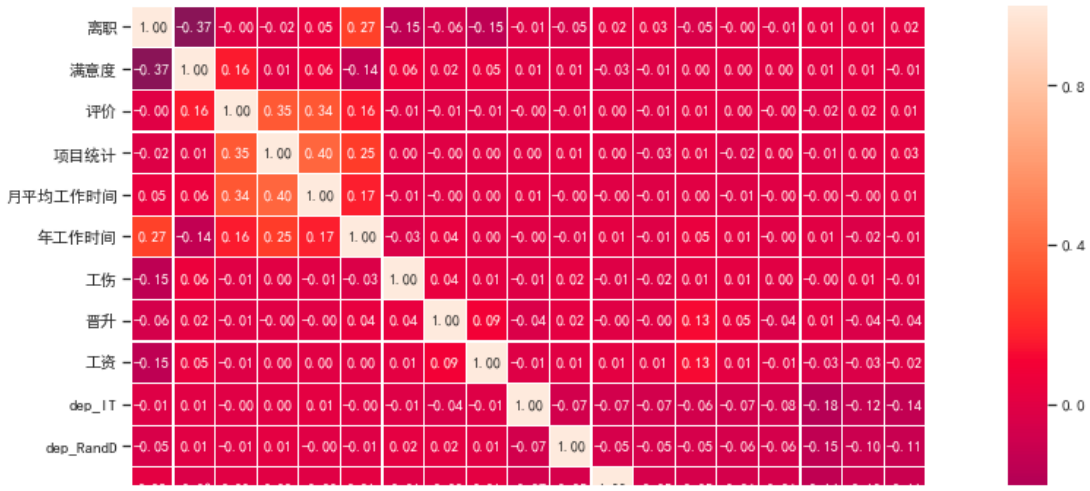
集群 3(绿色):工作勤奋，快乐

2.5.2 皮尔逊相关系数分析

代码如下:

```
df['工资'] = df.工资.map({"low": 0, "medium": 1, "high": 2}) #定序，薪资水平含有顺序意义，将其字符型转化为数值型。
df_one_hot = pd.get_dummies(df, prefix="dep") #定类，岗位是定类型变量，对其进行 one-hot 编码，这里直接利用 pandas 的 get_dummies 方法。
df_one_hot.shape
#逻辑回归模型能够适应连续型变量，因此可以不用进行离散化处理，
#又由于多个特征之间差异差异较大会造成梯度下降算法收敛速度变慢，故进行归一化处理
hours = df_one_hot['月平均工作时间']    #定义处理的连续型数据
df_one_hot['月平均工作时间'] = df_one_hot.月平均工作时间.apply(lambda x: (x-hours.min()) / (hours.max()-hours.min()))   #连续型数据处理
#使用使用皮尔逊相关系数分析
correlation = df_one_hot.corr(method = "spearman")
plt.figure(figsize=(18, 10))
#绘制热力图
sns.heatmap(correlation, linewidths=0.2, vmax=1, vmin=-1, linecolor='w',fmt='.2f',
                    annot=True,annot_kws={'size':10},square=True)
```

部分结果如下图



以上图形在 2.4.2 的基础上进一步分析出不同部门与其它因素之间的关联分析，其中“+”为正相关，“-”为负相关。

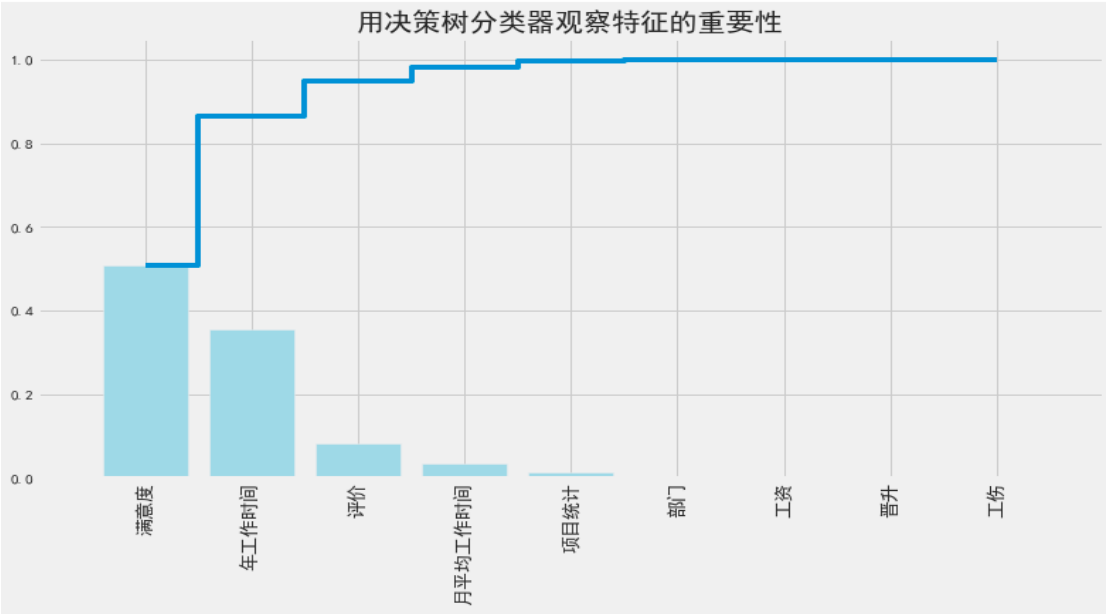
2.5.3 策树分类模型分析特征重要程度

代码如下：

```
from sklearn import tree
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
plt.style.use('fivethirtyeight')
plt.rcParams['figure.figsize'] = (12,6)
# 将这些变量转换为类别变量
df["部门"] = df["部门"].astype('category').cat.codes
df["工资"] = df["工资"].astype('category').cat.codes
# 建立训练集和测试集
target_name = '离职'
X = df.drop('离职', axis=1)
y=df[target_name]
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.15, random_state=123,
stratify=y)
```

```
dtree = tree.DecisionTreeClassifier(
        #max_depth=3,
        class_weight="balanced",
        min_weight_fraction_leaf=0.01
        )
dtree = dtree.fit(X_train,y_train)
importances = dtree.feature_importances_
feat_names = df.drop(['离职'],axis=1).columns
indices = np.argsort(importances)[::-1]
plt.figure(figsize=(12,6))
plt.title("用决策树分类器观察特征的重要性")
plt.bar(range(len(indices)), importances[indices], color='lightblue',  align="center")
plt.step(range(len(indices)), np.cumsum(importances[indices]), where='mid',
label='Cumulative')
plt.xticks(range(len(indices)), feat_names[indices], rotation='vertical',fontsize=14)
plt.xlim([-1, len(indices)])
plt.show()
```

结果如下图



通过使用决策树分类器，可以对用于预测的特征进行排序。前三名是员工满意度，工作年份和评价。这对创建逻辑回归模型很有帮助，因为当我们使用较少的特性时，理解模型中包含的内容会更容易理解。

3 结果分析

3.1 该公司的员工离职现状

（1）当前离职率为 23.81%;

（2）离职人员对公司满意度普遍较低，在 0.4 左右，应该反思具体哪里做的不对，有则改之无则加勉。

（3）.离职人员的考核成绩相对较高，说明离职人员多数为优秀人才；优秀人才承受了更大的压力，绷太紧导致离职，应该加快人才的培养，不会出现断层情况，一旦有离职立马有人接上。

（4）.项目数范围为 2~7 个，其中参加 7 个项目的离职率最高，其次是 2 个的; 7 个的工作能力较强，在其他企业有更好的发展, 2 个的可能是在该公司中工作能力不被认可；项目数的安排应该控制在一个合理的范围，既能让公司盈利，又能让员工实现价值，达到双赢的目的。

（5）.离职人员的平均每月工作时长较长。

（6）.离职人员的工作年限集中在 3 到 6 年。

（7）.5 年内未升职的离职率较高。

（8）.hr 岗位的离职率最高，目前企业普遍存在"留人难，招人难"，这可能是导致该岗位的离职率高的主要原因。

（9）.低等薪资水平的离职率最高。

由于企业培养人才是需要大量的成本，为了防止人才再次流失，因此应当注重解决人才的流失问题，也就是留人，另外如果在招人时注意某些问题，也能在一定程度上减少人才流失. 因此，这里可将对策分为两种，一种是留人对策，一种是招人对策

3.2 留人对策

（1）.建立良好的薪酬制度，至少不得低于市场平均水平

(2).建立明朗的晋升机制，公平公开公正晋升

(3).完善奖惩机制，能者多劳，也应多得

(4).实现福利多样化，增加员工对企业的忠诚度

(5).重视企业文化建设，树立共同的价值观，鼓励员工在建设过程中提意见

(6).改善办公环境以及营造良好的工作氛围，及时纾解员工心理问题

(7).鼓励员工自我提升，设立学习奖

3.3 招人对策

(1).明确企业招聘需求，员工的能力应当与岗位需求相匹配，而不是为指标而招

(2).与应聘者坦诚相见，不得为了完成指标哄骗应聘者

(3).招聘期间给予的相关承诺必须实现

(4).欢迎优秀流失人才回归