**2020年安徽省大数据与人工智能应用初赛**

**学校名称： 皖西学院 队伍名称： zzw 队伍编号： wxxy3**

# 第四部分：可视化（15分）

大数据时代已经到来，各企业迫切希望从已经积累的数据中分析出有价值的东西，而用户行为和评价体系的分析尤为重要。利用大数据来分析商品评分与消费习惯，可以预测商品的发展的趋势，提高产品质量，同时提高用户满意度。用户在选择商品的时候也可以根据商品评分去衡量。

**数据： **

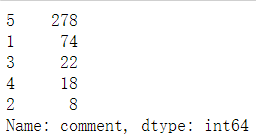
**某电商网站商品的评价：**共有400行、2列。这代表本训练集共有400条数据，每条数据有2类信息。包括：  
• Id => 用户ID  
• 评价等级 （1-5个等级，类似于1星-5星评价）

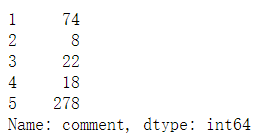
**要求用以上数据集合做可视化数据分析：**

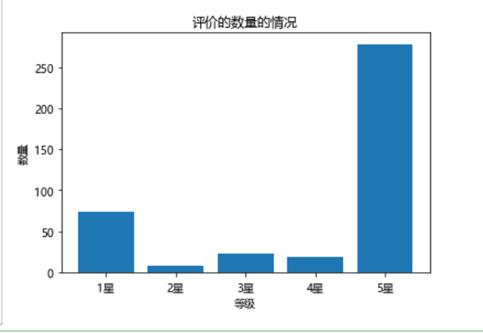
### 1利用柱状图显示各种评价的数量并保存结果（5分）

方法一：

#导入包  
import numpy as np  
import matplotlib.pyplot as plt  
import pandas as pd  
import matplotlib  
  
#解决中文乱码  
plt.rcParams['font.sans-serif'] = ['MicroSoft YaHei']  
plt.rcParams['axes.unicode\_minus'] = False  
  
#读取数据  
df = pd.read\_csv('goods.txt',names = ['id','comment'])  
  
#处理数据画图  
a = df['comment'].value\_counts() #巨强大，可以对其分类并计算出每一类的数量

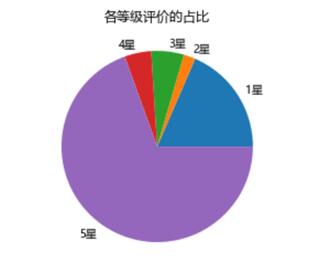
  
a = a.sort\_index()#巨强大，排序

  
plt.bar(range(1,6,1),a)  
plt.xticks(range(1,6,1),['1星','2星','3星','4星','5星'])  
plt.xlabel('等级')  
plt.ylabel('数量')  
plt.title('评价的数量的情况')



2通过饼图来展示各等级评价的占比（10分）

#导入包  
import numpy as np  
import matplotlib.pyplot as plt  
import pandas as pd  
import matplotlib  
  
#解决中文乱码  
plt.rcParams['font.sans-serif'] = ['MicroSoft YaHei']  
plt.rcParams['axes.unicode\_minus'] = False  
  
#读取数据  
df = pd.read\_csv('goods.txt',names = ['id','comment'])  
  
#处理数据画图  
a = df['comment'].value\_counts()  
a = a.sort\_index()  
plt.pie(a,labels = ['1星','2星','3星','4星','5星'])  
plt.title('各等级评价的占比')  
plt.show()



柱状图：

一：

#读数据并将各个列赋名字

df = pd.read\_csv("./goods.txt",names=["id","comment"])

#设置标签

plt.xlabel("等级")

plt.ylabel("数量")

plt.title("评价的数量的情况")

#读出列名为"comment"的值并排序

a = df["comment"].value\_counts()

a = a.sort\_index()

x = range(1,6)

\_xtick\_labels = ["{}星".format(i) for i in x]

plt.bar(x,a)

plt.xticks(x,\_xtick\_labels,rotation=45)

plt.show()

二：

#读数据并将各个列赋名字

df = pd.read\_csv("./goods.txt",names=["id","comment"])

#设置标签

plt.xlabel("等级")

plt.ylabel("数量")

plt.title("评价的数量的情况")

#读出列名为"comment"的值并排序

a = df["comment"].value\_counts()

a = a.sort\_index()

x = range(0,5) #如果是range(1,6)的话最后结果会整体向右移动一位，原因是尽管你这这里设置了1-6但是后面那个a.plot(kind=”bar”)将其还原为0-6

\_xtick\_labels = ["{}星".format(i+1) for i in x]

a.plot(kind="bar")

# plt.bar(x,a)

plt.xticks(x,\_xtick\_labels,rotation=45)

plt.show()