**import** pandas **as** pd  
**import** re  
*#读取数据*data=pd.read\_csv(**'comment.csv'**,encoding=**'utf-8'**)  
data.品牌.value\_counts()*#查看每个品牌的数据量*data=data[data.品牌==**'AO'**].评论  
  
  
*#数据预处理  
#换行符号删除*data=data.apply(**lambda** x:re.sub(**r'\\n'**,**''**,str(x)))*#raw表示原生的意思  
#&hellip;表情符号删除:以&符号开始,以;结束*data=data.apply(**lambda** x:re.sub(**'&[a-z]{1,10};'**,**''**,str(x)))  
*#热水器型号删除*data=data.apply(**lambda** x:re.sub(**'AO史密斯.+升'**,**''**,str(x)))  
  
  
  
*##分词***import** jieba *##使用jieba模块对中文语句进行分词*data\_cut=data.apply(**lambda** x:jieba.lcut(x))  
  
*##去除停用词*stop=pd.read\_csv(**'stoplist.txt'**,header=**None**,  
 encoding=**'utf-8'**,sep=**'song'**)  
stop.drop\_duplicates(inplace=**True**)  
stop=[**' '**,**'\u3000'**]+list(stop[0])  
  
*#确保停用词表里面不包含否定词以及程度词*degree=pd.read\_csv(**'degree.csv'**,engine=**'python'**,encoding=**'utf-8'**)  
degree.columns=[**'term'**]  
  
degree[**'score'**]=degree.index  
degree.index=list(range(0,len(degree)))  
no=pd.read\_csv(**'not.csv'**,engine=**'python'**,encoding=**'utf-8'**)  
no[**'score'**]=-999  
dic=pd.concat([degree,no],axis=0)  
**for** i **in** dic.term:  
 **if** i **in** stop:  
 stop.remove(i)  
data\_after=data\_cut.apply(**lambda** x:[i **for** i **in** x **if** i **not in** stop])  
  
  
*#词频***def** cipin(x,n=10):  
 temp=[**' '**.join(i) **for** i **in** x]  
 temp1=**' '**.join(temp)  
 temp2=pd.Series(temp1.split())  
 num=temp2.value\_counts()  
 **return** (num[num>n])  
cut\_num=cipin(data\_after)  
num=cipin(data\_after)  
  
  
**import** cv2  
**from** wordcloud **import** WordCloud  
**import** matplotlib.pyplot **as** plt  
plt.rcParams[**'font.sans-serif'**] = **'SimHei'** *#设置字体为SimHei显示中文*plt.rcParams[**'axes.unicode\_minus'**] = **False**back\_pic = cv2.imread(**"aixin.jpg"**) *# 设置背景图片*wc = WordCloud(font\_path=**'simkai.ttf'**, *# 设置字体* background\_color=**"white"**, *# 背景颜色* max\_words=2000, *# 词云显示的最大词数* mask=back\_pic, *# 设置背景图片* max\_font\_size=200, *# 字体最大值* random\_state=42)  
gar\_wordcloud = wc.fit\_words(num[:400]) *# cut\_num是由频数构成的Series的形式,且单词作为索引*plt.figure(figsize=(16, 8))  
plt.imshow(gar\_wordcloud)  
plt.title(**'京东评价关键词'**)  
plt.axis(**'off'**)  
plt.show()  
  
  
  
*#情感倾向分析  
#1.1计算情感词表  
#导入情感词表*feel=pd.read\_table(**'BosonNLP\_sentiment\_score.txt'**,sep=**' '**,  
 header=**None**,engine=**'python'**,encoding=**'utf-8'**)  
feel.columns=[**'word1'**,**'score1'**]  
*#评分  
#分词后的结果,转为Series的格式*temp=**' '**.join([**' '**.join(x) **for** x **in** data\_after])  
new\_data=pd.Series(temp.split())  
len(new\_data)  
*#n\_data=[]#如果没有把全角符号的空格去除掉的话,也可以正常运行的  
#for i in comment\_after:  
 #n\_data.extend(i)  
#Index索引改为对应的评论位置*ID=[]  
**for** i **in** data\_after.index:  
 ID.extend([i]\*len(data\_after[i]))  
new\_data.index=ID  
*#len(new\_data)*new\_data=pd.DataFrame(new\_data)  
new\_data.columns=[**'word'**]  
*#len(new\_data)  
##把否定词表,评分表,以及分词结果合并*temp=pd.merge(new\_data,feel,how=**'left'**,left\_on=**'word'**,right\_on=**'word1'**)  
ndata=pd.merge(temp,dic,how=**'left'**,left\_on=**'word'**,  
 right\_on=**'term'**)  
ndata.index=ID  
*#len(ndata)***del** ndata[**'word1'**],ndata[**'term'**]  
*#找出包含否定词,程度副词的Index  
#len(ndata[ndata['score'].isnull()==False])*index\_dic=list(set(ndata[ndata[**'score'**].isnull()==**False**].index))  
*#len(index\_dic)  
#找出不包含否定词,程度副词的Index*index\_nor =list(set([i **for** i **in** ndata.index **if** i **not in** index\_dic]))  
*#对不含否定词,程度词的评论进行评分求和*new\_score = pd.DataFrame(index=list(set(ndata.index)))  
new\_score[**'score'**] = 0 *## 储存我们情感总分***for** i **in** index\_nor:  
 new\_score.loc[i,**'score'**] = ndata.loc[0,**'score1'**].sum()  
  
*# 对含有否定词、程度词的评论进行情感分值的调整***for** i **in** index\_dic:  
 temp = ndata.loc[i]  
 **if** len(temp.word) > 1:  
 temp.index = range(len(temp))  
 *# 否定词 -999  
 #1 双重否定表肯定；2 否定词在句末；3 否定词后面是词语，词语情感反转* a = [x **for** x **in** temp.index **if** temp.loc[x,**'score'**] == -999]  
 **for** k **in** a:  
 **if** k == len(temp) - 1: *#2 否定词在句末* temp.loc[k,**'score1'**] = 0  
 **elif** temp.loc[k+1,**'score'**] == -999: *#1 双重否定表肯定；* temp.loc[k,**'score1'**] = 0  
 temp.loc[k+1,**'score1'**] = 0  
 a.remove(k+1)  
 **else**:  
 temp.loc[k,**'score1'**] = 0 *#3 否定词后面是词语，词语情感反转* temp.loc[k+1,**'score1'**] = (-1)\*temp.loc[k+1,**'score1'**]  
 *# 程度词-50、-200、-400%10==0;程度词在句末；不在句末* b = [i **for** i **in** range(len(temp)) **if** temp.loc[i,**'score'**]%10 == 0]  
 **for** k **in** b:  
 **if** k == len(temp) -1:  
 temp.loc[k,**'score1'**] = 0  
 **else**:  
 temp.loc[k+1, **'score1'**] = temp.loc[k+1,**'score1'**]\*temp.loc[k, **'score'**]/(-100)  
 temp.loc[k, **'score1'**] = 0  
 new\_score.loc[i,**'score'**] = temp.score1.sum()  
  
  
new\_score[**'score'**].sort\_values(ascending=**False**)  
data\_after[6057]  
*#print(data['评论'][6057])  
# 根据new\_score的分数将评论划分为正面评论及负面评论*index\_pos = [i **for** i **in** new\_score.index **if** new\_score.loc[0,**'score'**] > 0]  
pos = data\_after[index\_pos] *##正面评论*index\_neg = [i **for** i **in** new\_score.index **if** new\_score.loc[i,**'score'**] < 0]  
neg = data\_after[index\_neg] *##负面评论  
  
  
##根据正面评论绘制词云  
##根据正面评论绘制词云*num=[]  
[num.extend(i) **for** i **in** pos]  
num=pd.Series(num).value\_counts()  
**import** cv2  
**from** wordcloud **import** WordCloud  
**import** matplotlib.pyplot **as** plt  
plt.rcParams[**'font.sans-serif'**] = **'SimHei'** *#设置字体为SimHei显示中文*plt.rcParams[**'axes.unicode\_minus'**] = **False**back\_pic = cv2.imread(**"aixin.jpg"**) *# 设置背景图片*wc = WordCloud(font\_path=**'simkai.ttf'**, *# 设置字体* background\_color=**"white"**, *# 背景颜色* max\_words=2000, *# 词云显示的最大词数* mask=back\_pic, *# 设置背景图片* max\_font\_size=200, *# 字体最大值* random\_state=42)  
gar\_wordcloud = wc.fit\_words(num[:400]) *# cut\_num是由频数构成的Series的形式,且单词作为索引*plt.figure(figsize=(16, 8))  
plt.imshow(gar\_wordcloud)  
plt.title(**'京东正面评价关键词'**)  
plt.axis(**'off'**)  
plt.show()  
  
  
*##根据负面评论绘制词云  
##根据负面评论绘制词云*num=[]  
[num.extend(i) **for** i **in** neg]  
num=pd.Series(num).value\_counts()  
**import** cv2  
**from** wordcloud **import** WordCloud  
**import** matplotlib.pyplot **as** plt  
plt.rcParams[**'font.sans-serif'**] = **'SimHei'** *#设置字体为SimHei显示中文*plt.rcParams[**'axes.unicode\_minus'**] = **False**back\_pic = cv2.imread(**"aixin.jpg"**) *# 设置背景图片*wc = WordCloud(font\_path=**'simkai.ttf'**, *# 设置字体* background\_color=**"white"**, *# 背景颜色* max\_words=2000, *# 词云显示的最大词数* mask=back\_pic, *# 设置背景图片* max\_font\_size=200, *# 字体最大值* random\_state=42)  
gar\_wordcloud = wc.fit\_words(num[:400]) *# cut\_num是由频数构成的Series的形式,且单词作为索引*plt.figure(figsize=(16, 8))  
plt.imshow(gar\_wordcloud)  
plt.title(**'京东负面评价关键词'**)  
plt.axis(**'off'**)  
plt.show()