

# Python 数据分析与数据挖掘（ Python for Data Analysis&Data Mining ）

## Chap 15 评论数据情感分析

---

### 内容：

- 问题背景
- 文本数据预处理
- 中文分词
- 主题模型
- 文本情感分析
- 应用：情感分析是非常重要的一类文本挖掘任务，可以应用到舆情分析、观点立场分析、公共安全、电商产品评论等；包含中文或英文的产品评论数据，例如电商数据、微博评论、豆瓣电影评论、推文评论、酒店评论、餐馆评论、金融股票情感预测、机器人情绪对话等。

### 实践：

- 获取数据：爬虫工具--八爪鱼采集器 (<http://www.bazhuayu.com/>)
- 自然语言处理基础：文本预处理、分词、停词过滤等
- 主题模型实现：LDA
- Gensim 词向量
- 自然语言处理的Python包：jieba、snownlp、gensim

### 安装Python库

- Gensim : pip install gensim # 词向量学习、主题模型
- jieba: pip install jieba # 中文分词
- snownlp: pip install snownlp # 分词，词性标注，情感分析

---

这节课是在前面数据分析和数据挖掘的基础上，针对中文文本类型的数据进行初步的自然语言处理和文本挖掘，并结合实际中文产品评论数据来构建评论情感分析系统。

本节课通过对电商平台上热水器产品的用户评论文本数据进行预处理、分析和挖掘，分析用户的情感倾向，挖掘产品的优点和缺点，提炼产品的卖点，帮助生产者改进产品，帮助潜在用户挑选和归纳产品优缺点。本节课虽然从某家电产品的评论数据分析，也可以扩展到其他实体产品、体验产品或者服务产品的情感分析（如社交评论、酒店餐饮、金融股票等）。本节课采用非常基本的自然语言处理和文本挖掘技术进行简单文本处理和分析，在实际应用中，还可以进行更深入的自然语言处理的研究和应用。

### 准备工作：导入库，配置环境等

In [ ]:

```
# -*- coding: utf-8 -*-
from __future__ import division
import os, sys

# 启动绘图
%matplotlib inline
import matplotlib.pyplot as plt

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

# 1. 问题背景

随着互联网和电子商务的快速发展，网上购物对人们消费模式产生巨大的影响。据《中国互联网发展趋势报告2016》统计，2005至2016这十年间，中国居民网上消费增长率明显高于收入增速。截至2016年6月，根据中国互联网络信息中心(CNNIC)发布的《第38次中国互联网络发展状况统计报告》，‘我国网络购物用户规模达到4.48亿，较2015年底增加3448万’，网络购物用户规模的半年增长率达到8.3%，其中手机端用户规模的半年增长率甚至接近20%。作为传统零售与信息消费相结合的产物，网络购物顺应了这个新型消费升级的发展趋势。目前我国的网络购物市场保持快速、稳健的增长趋势。

电商平台，如阿里巴巴(包含淘宝，天猫，聚划算，天猫国际，天猫超市，社交，旅游，餐饮，互联网金融，本地生活服务等)和京东等，涵盖了成千上万种类的产品，在没有接触实际产品的情况下选购产品，用户的评论信息具有很高的参考价值，潜在用户可以依据用户评论挑选合适的产品，店家和厂家也可以根据用户评论对产品和服务进行改进和提升。对成千上万条评价进行逐条浏览或者通过人工规则进行归纳都很费时费力，应用自然语言处理和文本挖掘的方法对产品评论进行自动挖掘，在很大程度上可以改善和提升用户体验。正是因为产品评论挖掘有着强大的现实应用意义和科学价值，评论情感分析(sentiment analysis, SA)也成为越来越多的研究者和工业界的兴趣焦点。

以下是京东、淘宝、豆瓣、点评等各领域不同产品的评论截图示例。

The image displays four screenshots illustrating user reviews across different platforms:

- JD.com (Top Left):** A search results page for "美的 热水器". It shows a grid of products with price tags like 4199.00 and 1499.00. To the right, two review snippets are shown for a "BS爆款遥控系列 60升" model. The first review (96% positive) says: "东西收到这么久，都忘了去好评，美的大品牌，值得信赖，东西整体来看，个人感觉还不错，没有出现什么问题，值得拥有！" The second review (4.8 rating) says: "安装师傅很给力，热水器也好用，感谢美的."
- Taobao (Bottom Left):** A search results page for "美的 热水器". It shows a grid of products with price tags like 1499.00 and 1299.00. To the right, a review snippet for a "美的 WIFI智控 爆款升级版" (4.8 rating) says: "态度不错(3491) 快递不错(3054) 质量好(2448) 款式漂亮(857)".
- Douban (Bottom Right):** A review for a "美的 F60-15WB(EY)" water heater. The review (4.7 rating) says: "性价比很高的的一款热水器，大小正合适，售后完美，及时有效不拖沓，态度可亲，安装师傅没看到，不太清楚，但线上的客服真的很不错。热水器没有用，会推荐给亲朋同事，抽油烟机，热水器都是美的的，准备空调也来这个。一开始比较急躁，态度强硬，对客服很是抱歉~".
- Dianping (Bottom Right):** A review for a hotel room. The review (4.7 rating) says: "以前预留的太阳能，所以只能接单路，买的时候看过了评价所以心里大概知道有安装的材料费，材料费140元可以接受安装师傅非常好，十分满意！比酒店便宜3.400块，昨天晚上订的今天".

## 豆瓣电影

电影、影人、影院、电视剧

影讯&购票 选电影 电视剧 排榜 分类 影评 2016年度榜单 2016观影报告

### 看过“摔跤吧！爸爸”的豆瓣成员

255663人参与评价 ······

★★★★★ 9.2
力荐

力荐	65.4%
推荐	29.1%
还行	4.9%
较差	0.4%
很差	0.2%

263837人看过
53707人想看

大琳姐 (上海)

2017-05-31 tags: 励志 成长 2017 ★★★★★

那个小谁 (青岛)

2017-05-31 ★★★★★

美味蟹堡堡 (北京)

2017-05-31

对于印度这个国家来说这样的爸爸确实非常难得 但这难道不是另一种压迫么。。不太尊重自己女儿的意愿 把自己的愿望嫁接在女儿身上 强制练习摔跤 剃光头 在意父亲的权威性对外来教学的抵抗心理 这也就是那个教练结果确实不好衬托出父亲的正确性 摔上这样的父亲是幸运也是可怜

TOTO (河源)

2017-05-31 ★★★★★

大众点评 HOT

上海  搜

首页 团购 找优惠 订座 同城活动 社区 热门影视

美食 火锅 人气聚餐 >

休闲娱乐 足疗 KTV >

结婚 旅游婚纱 婚纱摄影 >

丽人 美发 美甲美睫 >

酒店 经济型 五星级 >

亲子 亲子摄影 早教中心 >

周边游 展馆展览 温泉 >

运动健身 健身中心 >

购物 综合商场 服饰鞋包 >

家装 装修设计 设计案例 >

学习培训 外语 音乐 >

重返童年 寻宝领大奖 1 2 3

热门导航

分类: 小吃快餐 本帮江浙菜 火锅 西餐 面包甜点  
日本菜 粤菜 川菜 综合商场 婚宴酒店 全部分类  
自助餐 美容/SPA 足疗按摩 美发

商圈: 徐家汇 人民广场 淮海路 静安寺 南京西路  
八佰伴 五角场/大学区 中山公园 陆家嘴 打浦桥  
虹桥 南京东路 世纪公园 外滩 全部商区

地铁线: 1号线 2号线 4号线 10号线 全部线路

排行榜: 最佳餐厅 人气餐厅 口味最好餐厅 热门购物 热门休闲

餐厅订座 可提前28天预订座位，不用排队哦！

所有地区 2017-05-31 星期三 18:00 4人 我要订座

网友点评(267) 搜索评论

大家认为

- ★★★★★ 凹头客(11) 上家快(3) 林道贤(11) 服务热情(19) 环境优雅(6)
- ★★★★★ 性价比高(8) 深夜食堂(3) 午餐(39) 午餐(24) 跳蚤(7) 炸鸡(7)

停车场信息(5) 有图片(63) 全部星级

子茶抽子 VIP ★★★★★ 口味: 3 环境: 1 服务: 2

#外卖#刚想吃汉堡，刚刚巧遇吃货节，打折优惠力度很大，外卖点起来。原味板烧鸡腿堡套餐口感一直没有变过，入口肉汁浓郁，不过感觉小了一码，现在一个汉堡不吃饱了~麦辣鸡翅2对才打半折，打开包装只有一对辣翅，另一对居然莫名其妙变成了烤翅，不开心！麦辣源热热的口感最佳，满满的菠萝香味，汁水浓郁。

05-17 新于-17-05-18 07:55 麦当劳 赞(1) 回应 收藏 举报

rainx7 青青青青 VIP ★★★★★ 口味: 4 环境: 4 服务: 4  
刚刚装修好，生意很不错呢！  
离家比较近，比较方便。  
活动有很多，显得有些眼花缭乱。  
这次用的是优惠券，加上招商银行闪付25-10的活动，十五元钱买了两杯.....  
喜欢的点：薯条 新地

更多

鑫盛楼经典本帮川菜 人均 ¥67

吉利尼Kilin... 人均 ¥45

疯狂BOX 人均 ¥24

私家糖水铺 人均 ¥40

亲子屋和风饭... 人均 ¥54

一芳台湾水果茶 人均 ¥15

附近团购 附近商户

绝味鸭脖 绝味鸭脖 仅售26.5元！...

联豪食品 联豪食品 牛排+意面组合...

联豪食品 联豪食品 脆皮西生牛排1...

## 2. 挖掘目标

本节课对电商平台上的某品牌热水器评论进行文本挖掘建模，目标如下：

1. 分析某品牌热水器的用户情感倾向
2. 从评论文本中挖掘出该品牌热水器的优点与不足
3. 提炼不同品牌热水器的卖点

### 3. 分析方法

本节课针对电商平台上的某品牌热水器的消费者的中文文本评论数据，在对文本进行基本的预处理、中文分词、停用词过滤后，建立包括词向量表达、主题模型等多种文本挖掘模型，实现对文本评论数据的情感倾向性判断以及文本信息挖掘，以期望得到有价值的内容。

### 4. 分析过程

#### 4.1 评论数据的采集

电商平台上的产品评论数据的采集，可以采用爬虫工具--[八爪鱼采集器](http://www.bazhuayu.com/) (<http://www.bazhuayu.com/>)。

从电商平台上采集得到的原始数据保存在 data/huizong.csv 文件中，数据的样例内容如下：

	Id,已采,已发,电商平台,品牌,评论,时间,型号,PageUrl
1	1,TRUE,FALSE,京东,AO,挺好的，安装人员很负责 指的夸奖,2014/8/26 9:07, AO史密斯 (A.O.Smith) ET300J-60 电热水器 60升, <a href="http://s.club.jd.com/productpage/p-1008025-s-0-t-1-p-0.html?callback=CommentListNew.setData">http://s.club.jd.com/productpage/p-1008025-s-0-t-1-p-0.html?callback=CommentListNew.setData</a>
2	2,TRUE,FALSE,京东,AO,自己安装的，感觉蛮好。后续追加。,2014/10/17 14:24, AO史密斯 (A.O.Smith) ET300J-60 电热水器 60升, <a href="http://s.club.jd.com/productpage/p-1008025-s-0-t-1-p-1.html?callback=CommentListNew.setData">http://s.club.jd.com/productpage/p-1008025-s-0-t-1-p-1.html?callback=CommentListNew.setData</a>
3	3,TRUE,FALSE,京东,AO,\n\nAO史密斯 (A.O.Smith) ET300J-60 电热水器 60升 还没安装,2014/11/12 13:36, AO史密斯 (A.O.Smith) ET300J-60 电热水器 60升, <a href="http://s.club.jd.com/productpage/p-1008025-s-0-t-1-p-2.html?callback=CommentListNew.setData">http://s.club.jd.com/productpage/p-1008025-s-0-t-1-p-2.html?callback=CommentListNew.setData</a>
4	

#### 4.2 评论数据的抽取

从huizong原始数据中抽取出品牌为美的的产品评论数据。代码如下：

In [ ]:

```
#-*- coding: utf-8
import pandas as pd

inputfile = 'data/huizong.csv'
outputfile = 'data/meidi_jd.txt'
data = pd.read_csv(inputfile, encoding='utf-8')
data = data[[u'评论']][data[u'品牌'] == u'美的']
data.to_csv(outputfile, index=False, header=False, encoding='utf-8')
```

In [ ]:

```
print len(data)
print data.head()
print data.tail()
```

In [ ]:

```
print data.iloc[55430][0], '\n' # 全文是24个“一般”
print data.iloc[55370][0], '\n' # 全文是32个I
print data.iloc[55357][0], '\n' # 长篇评论的内容
print data.iloc[55584][0], '\n' # 许多感叹号
```

## 4.2 评论数据的预处理

- 评论数据有 55774 条
- 文本评论中有大量价值很低甚至没有价值含量的条目，会对分析造成很大的影响，得到的结果的质量也必然是存在问题的。

因此，需要在进行分析之前，进行文本预处理，包括：

- 文本去重：例如，系统默认的好评；统一用户购买多个产品的相同/相似评论；垃圾评论
- 压缩长的重复词：例如，好好好好好好，一般一般一般
- 删去短句：字数非常少的评论
- 中文分词：jieba，结巴提供分词、词性标注、未登录词识别，支持用户词典等

### 1 ) 原始评论数据去重处理

数据去重后文件在 data/meidi\_jd\_process\_1.txt，代码如下：

In [ ]:

```
#-*- coding: utf-8 -*-
import pandas as pd

inputfile = 'data/meidi_jd.txt' #评论文件
outputfile = 'data/meidi_jd_process_1.txt' #评论处理后保存路径
data = pd.read_csv(inputfile, encoding = 'utf-8', header = None)
l1 = len(data)
data = pd.DataFrame(data[0].unique())
l2 = len(data)
data.to_csv(outputfile, index = False, header = False, encoding = 'utf-8')
print u'原始数据 %s条评论，现在剩余 %s条评论' %(l1, l2)
print u'删除了 %s条评论。' %(l1 - l2)
```

### 2 ) 原始评论数据压缩去词

用户的文本评论数据质量参差不齐，没有意义的文本数据很多。经过去重之后，还有很多评论需要处理，例如：

- “非常好非常好非常好非常好非常好非常好”
- “一般一般一般一般一般一般一般一般”
- “哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈哈”
- “15分钟就出热水了，感觉还不错，但是安装费实在太贵太贵太贵太贵太贵太贵”
- “为什么为什么为什么安装费这么贵，毫无道理！”
- “安装师傅滔滔不绝的向我阐述这款热水器有多么多么好”

根据观察和分析，对于重复词，不能一概而论删除。

处理后文件在 data/meidi\_jd\_process\_2.txt

### 3 ) 原始评论数据删除短句

过少字数的评论是没有意义的评论，比如3个字，只能表达诸如“很不错”，“质量差”。

- 原本就过短的评论文本，如“很不错”，“质量差”
- 压缩去词后生成的过短的评论文本，原本存在连续重复的且无意义的长文本，如“好好好好好好好  
好”变成了“好”

结合特定的语料，可以确定4-8个字符为下限长度。

这里设定下限为7个字符，即经过前面两步预处理后得到的评论，如果长度小于等于4个字符，即删去该语料。这样，经过处理后的文件在 data/meidi\_jd\_process\_3.txt

### 4) 情感类标的机器标注

可以使用SnowNLP对评论进行类标标注，这样使用机器标注的办法也许不准确，但是减少了人工成本。

机器标注处理前的文件为 data/meidi\_jd\_process\_end.txt，采用SnowNLP对文件进行情感得分处理，并根据每个评论的得分将评论分为：正面评论和负面评论，分别为 data/meidi\_jd\_pos\_1.txt 和 data/meidi\_jd\_neg\_1.txt

## SnowNLP库

SnowNLP是一个Python写的类库，可以方便的处理中文文本内容，是受到了TextBlob的启发而写的，由于现在大部分的自然语言处理库基本都是针对英文的，于是写了一个方便处理中文的类库，并且和TextBlob不同的是，这里没有用NLTK，所有的算法都是自己实现的，并且自带了一些训练好的字典。

SnowNLP 可以实现情感分析，分词，词性标注等功能。

In [ ]:

```
from snownlp import SnowNLP

s = u'一般一般一般一般一般一般一般一般一般' # 0.5263
#s = u'希望你们可以给个说法，本人客观评价，随时等候你们联系' # 0.2876
#s = u'给公司宿舍买的，上门安装很快，快递也送的及时，不错的。给五分吧' # 0.8431
#s = u'不错不错的哦，第一次在京东买这些产品，感觉相当好' # 0.9900
#s = u'很满意，水方一晚上都还是热的早上还能再洗' # 0.1284
print s, SnowNLP(s).sentiments
```

In [ ]:

```
# -*- coding: utf-8 -*-
#-*-encoding=utf-8
import sys
reload(sys)
sys.setdefaultencoding('utf-8')

import time
import csv, codecs
from snownlp import SnowNLP

time = time.time()

inputfile = 'data/meidi_jd_process_end.txt' #评论文件
outputfile_pos = 'data/meidi_jd_pos_1.txt' #评论处理后保存路径
outputfile_neg = 'data/meidi_jd_neg_1.txt' #评论处理后保存路径

outf_pos = codecs.open(outputfile_pos, 'wb', encoding='utf-8')
outf_neg = codecs.open(outputfile_neg, 'wb', encoding='utf-8')

with codecs.open(inputfile, 'r', encoding='utf-8') as inf:
    for line in inf:
        if not line.strip(): # 跳过空行
            continue
        sentscore = round((SnowNLP(line.strip())).sentiments - 0.5)*20
#       print line, sentscore
        if sentscore > 0.0:
            outf_pos.write(str(sentscore)+'\t'+line.strip()+'\n')
        elif sentscore < 0.0:
            outf_neg.write(str(sentscore)+'\t'+line.strip()+'\n')

print time.time() - time
#outf_pos.close()
#outf_neg.close()
```

## 去除机器标注的情感类标

In [ ]:

```
#-*- coding: utf-8 -*-
import pandas as pd
import time

start = time.time()

#参数初始化
inputfile1 = 'data/meidi_jd_process_end_neg.txt'
inputfile2 = 'data/meidi_jd_process_end_pos.txt'
outputfile1 = 'data/meidi_jd_neg_2.txt'
outputfile2 = 'data/meidi_jd_pos_2.txt'

data1 = pd.read_csv(inputfile1, encoding = 'utf-8', header = None) #读入数据
data2 = pd.read_csv(inputfile2, encoding = 'utf-8', header = None)

data1 = pd.DataFrame(data1[0].str.replace('.*?\d+\t', '')) #用正则表达式修改数据
data2 = pd.DataFrame(data2[0].str.replace('.*?\d+\t', ''))

data1.to_csv(outputfile1, index = False, header = False, encoding = 'utf-8') #保存结果
data2.to_csv(outputfile2, index = False, header = False, encoding = 'utf-8')

l1 = len(data1)
l2 = len(data2)
print u'负面评论有 %s 条' %(l1)
print u'正面评论有 %s 条' %(l2)

print time.time() - start
```

## 5 ) 分词

对正面情感和负面情感两个文本都进行中文分词，保存为两个文本文档，并与停用词一起作为主题模型（LDA）的输入。分词处理的代码如下：

### jieba 中文分词

中文分词库，支持三种分词模式：

- 精确模式，试图将句子最精确地切开，适合文本分析；
- 全模式，把句子中所有的可以成词的词语都扫描出来，速度非常快，但是不能解决歧义；
- 搜索引擎模式，在精确模式的基础上，对长词再次切分，提高召回率，适合用于搜索引擎分词。

此外，支持繁体分词，支持自定义词典（添加自定义词典），关键词提取，词性标注，并行分词，Tokenize：返回词语在原文的起始位置，ChineseAnalyzer for Whoosh搜索引擎

In [ ]:

```
# -*- coding: utf-8 -*-
#encoding = utf-8
import jieba

s = u'这个东西真心很赞'
seg_words = jieba.cut(s, cut_all=True) # 全模式,
#seg_words = jieba.cut(s, cut_all=False) #默认是精确模式
#seg_words = jieba.cut_for_search(s) #搜索引擎模式
print(' '.join(seg_words))
```

In [ ]:

```
#encoding = utf-8
import jieba
import pandas as pd
import time

start = time.time()

inputfile1 = 'data/meidi_jd_pos.txt'
inputfile2 = 'data/meidi_jd_neg.txt'

outputfile1 = 'data/meidi_jd_pos_cut_1.txt'
outputfile2 = 'data/meidi_jd_neg_cut_1.txt'

data1 = pd.read_csv(inputfile1, encoding='utf-8', header=None)
data2 = pd.read_csv(inputfile2, encoding='utf-8', header=None)

myseg = lambda s: ' '.join(jieba.cut(s)) #自定义简单分词函数

data1 = data1[0].apply(myseg) # 通过广播形式分词, 加快速度
data2 = data2[0].apply(myseg) # 通过广播形式分词, 加快速度

data1.to_csv(outputfile1, index=False, header=False, encoding='utf-8') # 保存结果
data2.to_csv(outputfile2, index=False, header=False, encoding='utf-8') # 保存结果

print time.time() - start
```

## 4.3 LDA 主题模型

### Gensim库 : LDA

In [ ]:

```
#encoding = utf-8
import pandas as pd
import jieba
import time

start = time.time()

#参数初始化
negfile = 'data/meidi_jd_neg_cut.txt'
posfile = 'data/meidi_jd_pos_cut.txt'
stoplist = 'data/stoplist.txt'

neg = pd.read_csv(negfile, encoding='utf-8', header=None, engine='python') #读入数据
pos = pd.read_csv(posfile, encoding='utf-8', header=None, engine='python')
stop = pd.read_csv(stoplist, encoding='utf-8', header=None, engine='python', sep='tipdm')
#sep设置分割词，由于csv默认以半角逗号为分割词，而该词恰好在停用词表中，因此会导致读取出错
#所以解决办法是手动设置一个不存在的分割词，如tipdm。
stop = [' ', '']+list(stop[0]) #Pandas自动过滤了空格符，这里手动添加

neg[1]=neg[0].apply(lambda s: s.split(' ')) #定义一个分割函数，然后用apply广播
neg[2]=neg[1].apply(lambda x: [i for i in x if i not in stop]) #逐词判断是否停用词，思路同上
pos[1]=pos[0].apply(lambda s: s.split(' '))
pos[2]=pos[1].apply(lambda x: [i for i in x if i not in stop])

from gensim import corpora, models

#负面主题分析
neg_dict = corpora.Dictionary(neg[2]) #建立负面词典
neg_corpus = [neg_dict.doc2bow(i) for i in neg[2]] #建立语料库
neg_LDA = models.LdaModel(neg_corpus, num_topics = 3, id2word = neg_dict) #LDA模型训练
for i in range(3):
    neg_LDA.print_topic(i) #输出每个负面评价主题

#正面主题分析
pos_dict = corpora.Dictionary(pos[2]) # 建立正面词典
pos_corpus = [pos_dict.doc2bow(i) for i in pos[2]] # 建立语料库
pos_LDA = models.LdaModel(pos_corpus, num_topics = 3, id2word = pos_dict) #LDA模型训练
for i in range(3):
    pos_LDA.print_topic(i) # 输出每个正面评价主题

print time.time() - start
```

## 观察：

经过LDA主题分析后，评论文本被聚成3个主题，每个主题下面生成10个最有可能出现的词语以及相应的频率。

### 下表展示了正面评价文本中的潜在主题

主题1	主题2	主题3
很好	不错	安装
送货	的	了
快	东西	师傅
速度	价格	元
很快	感觉	没有
就是	还不错	自己
好	京东	美的
加热	美的	的
服务	很不错	售后
非常	值得	上门

- 主题1中的高频特征词反映：京东送货快；服务非常好；美的热水器加热速度快
- 主题2中的高频特征词，热门关注点主要是价格、东西和值得等，反映美的的热水器不错，价格合适值得购买等
- 主题3中的高频特征词的热门关注点是售后、师傅、上门和安装，反映京东的售后服务以及师傅上门安装等

### 下表展示了负面评价文本中的潜在主题

主题1	主题2	主题3
安装	就是	了
师傅	不错	的
美的	加热	东西
元	不知道	没有
送货	不过	京东
售后	有点	自己
服务	还可以	还是
不好	使用	但是
上门	速度	这个
好	吧	可以

- 主题1中的高频特征词热门关注点主要是安装、服务、元等，主要反映：美的热水器安装收费高、热水器售后服务不好等
- 主题2中的高频特征词是不过、有点、还可以等情感词

- 主题3中的高频特征词是没有，但是，自己等，主要反映：美的热水器可能不满足其需求，等

## 分析：

综合以上对主题和高频特征词可以看出，美的热水器的优势有：

- 价格实惠、性价比高、外观好看、热水器实用、使用起来方便、加热速度快、服务好

用户的抱怨点主要体现在：

- 美的热水器安装的费用贵，售后服务等

因此，用户购买的原因可以总结为：美的大品牌值得信赖，价格实惠，性价比高。

根据对京东平台上美的热水器的用户评价情况进行主题模型分析，我们对美的的建议可以有：

1. 保持热水器使用方便、价格实惠等优点的基础上，对热水器进行改进，从整体上提升热水器的质量
2. 提升安装人员及客服人员的整体素质，提高服务质量。安装费用收取明文细则，并进行公开透明，减少安装过程的乱收费问题。适度降低安装费用和材料费用，以此在大品牌的竞争中凸显优势。

In [ ]: