电商产品评论数据情感分析

项目背景

随着网上购物越来越流行，人们对于网上购物的需求越来越高，这让京东、淘宝等电商平台得到了很大的发展机遇。但是，这种需求也推动了更多的电商平台的崛起，引发了激烈的竞争。在这种电商平台激烈竞争的大背景下，除了提高产品质量、压低商品价格外，了解更多消费者的心声对于电商平台来说越来越有必要了，其中非常重要的就是对消费者的文本评论数据进行内在信息的数据挖掘分析。

项目需求

对京东某一热水器进行文本挖掘分析，目标如下。

- 分析热水器的用户情感倾向。
- 从评论文本中挖掘出该品牌热水器的优点与不足。
- 提炼不同品牌热水器的卖点。

项目简介

本次建模针对京东商城上“美的”品牌热水器的消费者的文本评论数据，在对文本进行基本的机器预处理、中文分词、停用词过滤后，通过建立包括栈式自编码深度学习、语义网络与`LDA`主题模型等多种数据挖掘模型，实现对文本评论数据的倾向性判断以及所隐藏的信息的挖掘并分析，以期望得到有价值的内在内容。

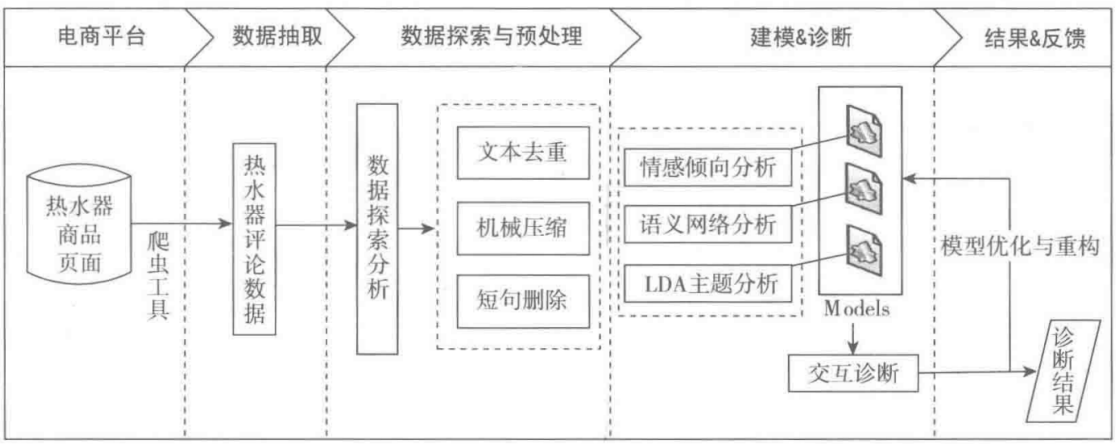
主要步骤如下。

1）利用`Scrapy-Redis`分布式爬虫框架，对京东商城进行热水器评论的数据采集。

2）但对获取的数据进行基本的处理操作，包括数据预处理、中文分词、停用词过滤等操作。

3）文本评论数据经过处理后，运用多种手段对评论数据进行多方面分析。

4）从对应结果的分析中获取文本评论数据中有价值的内容。



项目实现步骤

评论抽取代码

将品牌为“海尔”的一列评论抽取，另存为文本文件。代码如下：

```
# 电商产品评论数据情感分析/load_data.py
import pandas as pd

def csv2txt():
    '''
    提取评论并保存为txt
    :return:
    '''
    inputfile = 'data/huizong.csv'  # 评论汇总文件
    outputfile = 'data/output/meidi_jd.txt'  # 评论提取后保存路径
    data = pd.read_csv(inputfile, encoding='utf-8')
    data = data[[u'评论']][data[u'品牌'] == u'海尔']
    data.to_csv(outputfile, index=False, header=False, encoding='utf-8')

if __name__ == '__main__':
    csv2txt()
```

评论预处理

文本去重

文本去重的原因:

- 一些电商平台为了避免一些客户长时间不评论， 会设置一道程序， 如果用户超过规定的时间内没有评论， 会自动产生好评。 这类数据没有任何分析价值且大量重复， 必须删除。
- 同一个人购买多种热水器产品， 为了省事， 采用同样或者相近的评论。
- 人类的本质是复读机， 如果出现不同人评论之间完全重复， 一般也是毫无意义的， 直接删除。

如何实现去重?

- 去除一些自动好评的数据，重复的评论等没有价值的数据。大多数文本去重是基于文本之间的相似度，包括编辑距离去重，`simhash`算法去重等，这些会使得我们去除一些相近的表达，造成错删。下面代码采用比较删除法，尽量保留有用的评论。代码如下

```
# 电商产品评论数据情感分析/preprocessing_data.py
"""
数据预处理
"""
import pandas as pd
import jieba

def cleanSame():
    '''
    文本去重
    :return:
```

```
    '''
    inputfile = 'data/output/meidi_jd.txt'  # 评论文件
    outputfile = 'data/output/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)
    # 去重后返回时array，需要DataFrame化
    data = pd.DataFrame(data[0].unique())
    l2 = len(data)
    data.to_csv(outputfile, index=False, header=False, encoding='utf-8')
    print(u'删除了%s条评论。' % (l1 - l2))

if __name__ == '__main__':
    cleanSame()
```

- 运行结果

```
C:\Users\Administrator\AppData\Local\Programs\Python\Python37\python.exe C:/Users/Administrator/PycharmProjects/DataAnalyse/电商产品评论数据情感分析/preprocessing_data.py
删除了160条评论。
```

文本分词

文本分词，即将连续的字序列按照一定的规范重新组合成词序列的过程。分词的结果对后续算法有着很大的影响，本文采用`jieba`分词对文档中的评论数据进行中文分词。

LDA主题模型

模型介绍

模型实现

将文本一份为二，分为正面评价和负面评价两个文本分析。此处用`COSTCM6`中的情感分析做机器分类，因此得到的数据中要删除评分前缀，统一编码后再删除评分。代码如下:

```
# 电商产品评论数据情感分析/preprocessing_data.py

def cleanPrefix():
    '''
    删除前缀评分
    :return:
    '''
    # 参数初始化
    inputfile1 = 'data/meidi_jd_process_end_负面情感结果.txt'
    inputfile2 = 'data/meidi_jd_process_end_正面情感结果.txt'
    outputfile1 = 'data/output/meidi_jd_neg.txt'
    outputfile2 = 'data/output/meidi_jd_pos.txt'

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

    # 用正则表达式修改数据
```

```
    data1 = pd.DataFrame(data1[0].str.replace('.*?\d+?\\t ', ''))
    data2 = pd.DataFrame(data2[0].str.replace('.*?\d+?\\t ', ''))

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

if __name__ == '__main__':
    csv2txt()
```

- 源文件样式

```
4	还好 安装费有点贵
11	商品已经收到 打开包装检查一下外观完美；还没有安装使用 用后再评论
11	东西不错 租房子用的 足够了
8	很好 今天安装好了 非常满意
4	可以把 能用就好 出租的
16	安装材料费简直就是变相收费 几节pc管加几个弯头就要两百多
6	后期有一百多的管道费 插电一天要3 6度电\n
13	安装完成 还未使用感觉外观不错
16	东西挺好 样子挺漂亮 加热速度快
8	还不错的热水器 给爸妈买的 用起来很方便
```

- 处理后的文件样式

```
还好 安装费有点贵
商品已经收到 打开包装检查一下外观完美；还没有安装使用 用后再评论
东西不错 租房子用的 足够了
很好 今天安装好了 非常满意
可以把 能用就好 出租的
安装材料费简直就是变相收费 几节pc管加几个弯头就要两百多
后期有一百多的管道费 插电一天要3 6度电\n
安装完成 还未使用感觉外观不错
东西挺好 样子挺漂亮 加热速度快
还不错的热水器 给爸妈买的 用起来很方便
安装师傅服务态度差了点 京东服务还是不错的
便宜安装师傅好
```

去掉了前面的标签

对分类后的文档进行分词，代码如下：

```
# 电商产品评论数据情感分析/preprocessing_data.py

def cut():
    '''
    分词
    :return:
    '''
    # 参数初始化
    inputfile1 = 'data/output/meidi_jd_neg.txt'
    inputfile2 = 'data/output/meidi_jd_pos.txt'
```

```
    outputfile1 = 'data/output/meidi_jd_neg_cut.txt'
    outputfile2 = 'data/output/meidi_jd_pos_cut.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)

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

    # 通过“广播”形式分词，加快速度。
    # 将函数应用到由各列或行形成的一维数组上。DataFrame的apply方法可以实现此功能
    data1 = data1[0].apply(mycut)
    data2 = data2[0].apply(mycut)

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

if __name__ == '__main__':
    cut()
```

```
还好  安装费 有点 贵
商品 已经 收到  打开 包装 检查一下 外观 完美 ；  还 没有 安装 使用  用后 再 评论
东西 不错  租房子 用 的  足够 了
很 好  今天 安装 好 了  非常 满意
可以 把  能用 就 好  出租 的
安装 材料费 简直 就是 变相 收费  几节 pc 管加 几个 弯头 就要 两百多
后期 有 一百多 的 管道 费  插电 一天 要 3  6 度电 \ n
安装 完成  还 未 使用 感觉 外观 不错
东西 挺 好  样子 挺 漂亮  加热 速度 快
还 不错 的 热水器  给 爸妈 买 的  用 起来 很 方便
安装 师傅 服务态度 差 了 点  京东 服务 还是 不错 的
便宜 安装 师傅 好
```

实现了分词

分词之后建立LDA模型：

通过LDA主题分析后，评论被聚成3个主题，每个主题下生成10个最有可能出现的词语以及相应的概率。结果如下

根据对美的热水器好评的3个主题特征词提取，分析如下：

- 主题1中的高频特证词为很好，送货快，加热，速度，很快，服务和非常等，反映了京东送货快，服务好，美的热水器加热快。
- 主题2的高频特证词为价格、东西和值得，主要反映了热水器不错价格合适值得购买。

- 主题3的高频词为售后、师傅、上门和安装，反映了京东的售后服务以及师傅上门安装等。

差评的3个潜在主题中，可以看出主题1主要是安装、服务、元等，即反应了美的热水器安装收费高，热水器售后服务不好等；主题2是不过、有点、还可以等情感词；主题3是没有、但是、自己等，反映了热水器自己安装等。