项目说明

该项目一共分三个需求：

1、首先按照两个时间阶段去分析

首先进行数据清洗，通过数据处理这个py文件进行数据清洗

按照以下步骤进行清洗：

1. 去掉标点符号，去掉表情包（通过re.sub函数）
2. 根据jieba分词对清洗好的文本进行分词处理，在分词的过程中，去判断分词是否出现在停用词的文本里面，分词的长度是否大于等于2，以及该分词是否为中文，都符合了，就可以把分词保留下来
3. 接着根据pandas中的dropna去掉空值，最后把空值那一行给去掉
4. 接着根据发布地这一列去判断账号的性质是否为官号，是官号就把那一整行取代
5. 处理好的文本数据，前半年数据还剩154657 后半年数据还是117084
6. 最后根据处理好的数据，通过snownlp进行情感分类，通过snownlp中的sentiments去计算每个文本的情感分值，去判断博文中哪个是属于正面情感，哪个是负面情感

在处理好上面的数据之后，我们开始做基于TF-IDF的词频统计，词云可视化和基于lda的主题挖掘模型

代码在整体内容分析.py这个文件里面，

这里先说说tf-idf怎么去做的，这里是首先把数据进行划分，区分哪些数据上半年的，哪些数据是下半年的，接着再去把上半年的数据进行情感分类，单独对上半年的正向情感进行tf-idf分析，其他的同理，最后得到上半年，正向情感，负面情感的tf-idf的词云可视化，接着对下半年一样这么处理

接着对于可视化的话的方面，这边是基于tf-idf计算出来的值进行可视化的，采用的是pyecharts中的WordCloud方法

去根据tf-idf里面词的权重Top100进行可视化展示，在前面都处理好之后，我们单独去计算高频词，高频词不同于tf-idf，高频词是单独去统计每个词出现的频率情况，这里用CSV表保存下来

回过头我们来说说lda的做法，LDA我们同样划分了4个部分，分别是：

前半年的正面情感文本，负面情感文本

后半年的正面情感文本，负面情感文本

然后我们根据得到的文本去查看前面处理好的数据，通过corpora.Dictionary和dictionary.doc2bow构建我们想要的字典模式，

然后去根据这些字典去用困惑度和一致性进行计算，来选取主题最优数

相关的可视化和数据都在需求一的文件夹里面，请你自己去自行查阅，名字：后半年*负面情感*困惑度和一致性.png、后半年*负面情感*da.html 一共有4个这样的文件

在选取最优主题数之后，我们对这些数据进行建模和可视化，最后得到不同部分对应的主题建模，同样也是放在了需要一的文档里面，因为数量太多，这里就不过多展示，因为我们这边主题图谱是基于每个主题的特征词去做的，所以我们在做完上面的建模后，根据print_topics函数去打印每个主题的模块，从而来获取每个主题在不同部分的特征词，在获取完特征词之后，我们这边还需要去统计每个节点之间的边的权重，这时候我们就通过共现网络.py这个代码去计算每个节点之间的边的权重，这个是计算的公式，最后计算好的文本在需求一的文件夹里面，具体名称：后半年*负面情感*entity.csv，有四个这样的文件

```
    while line:
        line = line.replace("\n", "")  # 过滤换行
        line = line.strip('\n')  # 过滤换行
        nums = line.split(' ')

        # 循环遍历关键词所在位置 设置word_vector计数
        i = 0
        j = 0
        while i < len(nums):  # ABCD共现 AB AC AD BC BD CD加1
            j = i + 1
            w1 = nums[i]  # 第一个单词
            while j < len(nums):
                w2 = nums[j]  # 第二个单词
                # 从word数组中找到单词对应的下标
                k = 0
                n1 = 0
                while k < len(word):
                    if w1 == word[k]:
                        n1 = k
                        break
                    k = k + 1
                # 寻找第二个关键字位置
                k = 0
                n2 = 0
                while k < len(word):
                    if w2 == word[k]:
                        n2 = k
                        break
                    k = k + 1
                # 重点: 词频矩阵赋值 只计算上三角
                if n1 <= n2:
                    word_vector[n1][n2] = word_vector[n1][n2] + 1
                else:
                    word_vector[n2][n1] = word_vector[n2][n1] + 1
                # print n1, n2, w1, w2
                j = j + 1
            i = i + 1
```

然后根据这些获取的数值内容，通过gephi来实现语义网络，生成的图也是在需要一文档里面，名字为：后半年_负面情感.svg，有四个这样的文件

2、按照各个地区去分析，他们的情感分析，和TF-IDF

这里的步骤，首先我们根据前面处理好的文本，通过各个地区情感表现以及词云图.py这个代码来实现，我们先确定每个省的名称：

```
provinces = ['安徽', '澳门', '北京', '重庆', '福建', '甘肃', '广东', '广西', '贵州',
'海南', '河北', '黑龙江', '河南', '湖北', '湖南', '江苏', '江西',
                  '吉林', '辽宁', '内蒙古', '宁夏', '青海', '山东', '上海', '山西', '陕
西', '四川', '台湾', '天津', '西藏', '香港', '新疆', '云南', '浙江']
```

接着去判断IP属地是否存在这个列表里面，如果没有说明，这个地区并不是我们想要的，这时候我们就把这些无效数据给剔除掉，剩下的数据就是符合我们想要分析的内容，

然后在确定好文本之后，我们依次去定位各个省的文本内容，然后基于这些文本内容，重新去做tf-idf每个词的权重计算，具体代码如下：

```
    for i in df['分词']:
        corpus.append(i.strip())

        # 将文本中的词语转换为词频矩阵 矩阵元素a[i][j] 表示j词在i类文本下的词频
    vectorizer = CountVectorizer()

    # 该类会统计每个词语的tf-idf权值
    transformer = TfidfTransformer()

    # 第一个fit_transform是计算tf-idf 第二个fit_transform是将文本转为词频矩阵
    tfidf = transformer.fit_transform(vectorizer.fit_transform(corpus))
    # 获取词袋模型中的所有词语
    word = vectorizer.get_feature_names_out()

    # 将tf-idf矩阵抽取出来 元素w[i][j]表示j词在i类文本中的tf-idf权重
    weight = tfidf.toarray()
```

在获取好每个词的tf-idf之后，我们还是和前面一样，采用的是pyecharts中的WordCloud方法进行词云可视化，并且去划分每个省的情感分析

去查看在各个省，正面情感的词云图表现情况，负面情感词云图的表现情况，这些全部内容就放在了需求二的文档里面，因为内容较多，这里也就不一一展示了

另外，如果某个省，缺失正面情感的分析或者负面情感的分析，说明因为该文本的内容缺失，导致无法分析，毕竟每个省的文本数据都不同

3、按照两个阶段去划分3个地区，然后去查看在不同的阶段，他们的情感、LDA、TF-IDF算法如何

这里的分析思路和上面的大同小异

这里无非就是把地区划分为中部，西部，东部，分析的代码名为：中西东区域表现.py

```
中部
central_area = ['河南','湖北','湖南','江西','安徽']
西部
west_area = ['内蒙古','新疆','宁夏','甘肃','青海','陕西','四川','重庆','贵州','云南','西藏']
东部
east_area = ['北京','天津','河北','山东','江苏','上海','浙江','福建','广东','海南','广西','香港','澳门','台湾']
```

这里是根据IP属地把地区划分为三个部分，然后对这三个部分进行一一分析

分析他们的tf-idf的词云图表现情况，分正负两面

以及根据不同地区的文本内容进行LDA建模，并且区分正负，做法和上面一致，只是分析思路有所不同，

以上便是一个简单的文档说明，该说明只是展示思路分析过程，内容之处还需要你个人补充