说明文档

1、先说说数据处理的问题

这边根据提供的文档，先把全部文档整理到一起，进行分析

主要是用情感分析.py这个文件

然后做数据预处理的步骤，

加入停用词，处理标签符号，表情包，判断是否为中文，以及采用jieba做分词处理

```
# 导入停用词列表
stop_words = []
with open("stopwords_cn.txt", 'r', encoding='utf-8') as f:
    lines = f.readlines()
    for line in lines:
        stop_words.append(line.strip())
```

```
#去掉标点符号，以及机械压缩
def preprocess_word(word):
    word1 = str(word)
    # word1 = re.sub(r'转发微博', '', word1)
    word1 = re.sub(r'#\w+#', '', word1)
    word1 = re.sub(r'【.*?】', '', word1)
    word1 = re.sub(r'@[\w]+', '', word1)
    word1 = re.sub(r'[a-zA-Z]', '', word1)
    word1 = re.sub(r'\.\d+', '', word1)
    return word1
```

```
def emjio_tihuan(x):
    x1 = str(x)
    x2 = re.sub('(\[.*?\])', "", x1)
    x3 = re.sub(r'@[\w\u2E80-\u9FFF]+:?/\[\w+\]', '', x2)
    x4 = re.sub(r'\n', '', x3)
    return x4
```

```
# 判断是否为中文
def is_all_chinese(strs):
    for _char in strs:
        if not '\u4e00' <= _char <= '\u9fa5':
            return False
    return True
```

```python
def get_cut_words(content_series):
    try:
        # 对文本进行分词和词性标注
        words = pseg.cut(content_series)
        # 保存名词和形容词的列表
        nouns_and_adjs = []
        # 逐一检查每个词语的词性，并将名词和形容词保存到列表中
        for word, flag in words:
            if word not in stop_words and len(word) >= 2 and is_all_chinese(word) == True:
                # 如果是名词或形容词，就将其保存到列表中
                nouns_and_adjs.append(word)
        if len(nouns_and_adjs) != 0:
            return ' '.join(nouns_and_adjs)
        else:
            return np.NAN
    except:
        return np.NAN
```

```python
# 去掉重复行以及空值
df1 = pd.read_excel('产品评论-东南亚_终.xlsx',sheet_name='Sheet1')
df1['地区分布'] = '东南亚'
df2 = pd.read_excel('产品评论-欧美_终.xlsx',sheet_name='Sheet1')
df2['地区分布'] = '欧美'
df3 = pd.read_excel('产品评论-日韩_终.xlsx',sheet_name='Sheet1')
df3['地区分布'] = '日韩'
df = pd.concat([df1,df2,df3],axis=0)

df['评论文本'] = df['评论文本'].apply(preprocess_word)
df['评论文本'] = df['评论文本'].apply(emjio_tihuan)
df.dropna(subset=['评论文本'], axis=0,inplace=True)
df['评论分词'] = df['评论文本'].apply(get_cut_words)
new_df = df.dropna(subset=['评论分词'], axis=0)
```

根据这些都处理好之后，就可以得到一堆的分词数据，数据文件在数据.csv这个文件夹里面

	S	T	U	V	W	X	Y	Z	AA	AB	AC	AD	AE	AF	AG	AH	AI	AJ	AK
1	评论用户名	评论等级	评论星级	评论点赞数	评论ip属地	评论出行类	评论文本	评论其他信	评论专项打	评论时间	评论供应商	评论供应商	图片列表	图片数量	年份	计数	地区分布	评论分词	情感分析
2	2 金書羽	铂金贵宾	5	2	浙江	情侣夫妻	第一次体验	热销酒店名	行程安排(5	########			https://vid	10	2023	1	东南亚	第一次 体验	积极态度
3	2 葛钰	铂金贵宾	5	1	辽宁	情侣夫妻	导游服务月	热销酒店名	行程安排(5	########			https://vid	10	2023	1	东南亚	导游 服务月	积极态度
4	2 匿名用户	铂金贵宾	5	0	甘肃	情侣夫妻	蜜月之行选	轻奢搭配3	行程安排(5	########			https://dim	8	2023	1	东南亚	蜜月 选择	积极态度
5	2 M36****72	铂金贵宾	5	0	江苏	情侣夫妻	选择巴厘岛	轻奢搭配3	行程安排(5	########			https://dim	8	2023	1	东南亚	选择 巴厘岛	积极态度
6	2 _M13****4	黄金贵宾	4.5	2	四川	情侣夫妻	这一趟巴厘	人气搭配3	行程安排(4	########			https://vid	10	2023	1	东南亚	一趟 巴厘岛	积极态度
7	2 M34****86	铂金贵宾	5	0	浙江	情侣夫妻	这是一篇对	明星同款3	行程安排(5	########			https://dim	5	2023	1	东南亚	一篇 导游	积极态度
8	2 M51****66	铂金贵宾	5	1	广东	情侣夫妻	很不错的体	独栋别墅+	行程安排(5	########			https://vid	10	2023	1	东南亚	不错 体验	积极态度
9	2 M23****11	铂金贵宾	5	3	四川	情侣夫妻	这次巴厘岛	明星同款3	行程安排(5	########	尊敬的游客	########	https://dim	6	2023	1	东南亚	巴厘岛 愉快	消极态度
10	2 M52****03	铂金贵宾	5	4	上海	情侣夫妻	很棒的蜜月	明星款AYA	行程安排(5	########			https://dim	9	2023	1	东南亚	很棒 蜜月	积极态度
11	2 M38****28	铂金贵宾	5	0	上海	其他出游	这次巴厘岛	明星款AYA	行程安排(5	########			https://dim	9	2023	1	东南亚	巴厘岛 行程	积极态度
12	2 小怪兽194	铂金贵宾	5	1	贵州	情侣夫妻	行程总体上	别墅+海边	行程安排(5	########			https://dim	9	2023	1	东南亚	行程 总体	积极态度
13	2 _WeCh****	钻石贵宾	5	1	广东	情侣夫妻	导游非常棒	人气搭配3	行程安排(5	########	尊敬的游客	########	https://dim	5	2023	1	东南亚	导游 中文	积极态度
14	2 M25****51	铂金贵宾	4	2	广东	单独旅行	1、服务：	别墅+海边	行程安排(5	########			https://dim	8	2023	1	东南亚	服务 整体	积极态度
15	2 257****26		4.5	0	北京	家庭亲子	导游非常专	别墅+海边	行程安排(5	########			https://dim	3	2023	1	东南亚	导游 专业	积极态度
16	2 E03****81	铂金贵宾	5	1	天津	情侣夫妻	本次的旅行	明星款AYA	行程安排(5	########			https://dim	6	2023	1	东南亚	本次 旅行	积极态度
17	2 M24****72	钻石贵宾	5	0	湖北	朋友出游	疫情后第一	明星同款3	行程安排(5	########	尊敬的游客	########		0	2023	1	东南亚	疫情 第一次	积极态度
18	2 _WeCh****	铂金贵宾	2.5	6	浙江	朋友出游	我只能说很	明星款AYA	行程安排(2	########	亲爱的携客	########		0	2023	1	东南亚	只能 说能	积极态度
19	2 爸爸耳	铂金贵宾	5	0	山东	情侣夫妻	行程很舒服	明星同款3	行程安排(5	########				0	2024	1	东南亚	行程 舒服	积极态度
20	2 E20****95	铂金贵宾	5	0	四川	情侣夫妻	行程规划比	2晚升级丽	行程安排(5	########				0	2023	1	东南亚	行程 规划	积极态度
21	2 Kenny27	铂金贵宾	4.5	0	上海	家庭亲子	主打一个自	2晚升级金	行程安排(5	########				0	2023	1	东南亚	主打 自由	积极态度
22	2 _WeCh****	铂金贵宾	5	0	河北	情侣夫妻	这次旅行体	热销酒店名	行程安排(5	########				0	2023	1	东南亚	旅行 体验	积极态度
23	2 M46****39	铂金贵宾	5	0	湖北	家庭亲子	带老父亲一	独栋别墅+	行程安排(5	########				0	2023	1	东南亚	老父亲 第一	消极态度
24	2 M29****57	铂金贵宾	5	0	上海	情侣夫妻	行程安排不	人气搭配3	行程安排(5	########				0	2023	1	东南亚	行程 安排	积极态度
25	2 M59****07	铂金贵宾	5	1	江苏	情侣夫妻	导游很棒，	人气搭配3	行程安排(5	########				0	2023	1	东南亚	导游 很棒	积极态度
26	2 moli-ruby	钻石贵宾	5	0	北京	家庭亲子	导游和司机	独栋别墅+	行程安排(5	########				0	2023	1	东南亚	导游 司机	积极态度
27	2 patrick80	钻石贵宾	5	5		家庭亲子	非常棒的一	别墅+海边	行程安排(5	########	尊敬的游客	########	https://dim	10	2020		1 东南亚	出行 亲子	积极态度

数据

接着根据这些分词数据采用snownlp情感分析库去做情感分析

```python
def analyze_sentiment(text):
    s = SnowNLP(text)
    sentiment = s.sentiments
    if sentiment > 0.5:
        return '积极态度'
    elif sentiment < 0.5:
        return '消极态度'
    else:
        return '中立态度'
```

把情感得分大于0.5为积极，小于0.5为消极，其余为中立

数据处理，以及情感分析就做好了

接着来说说，数据分析这一块

数据分析是基于数据.csv这个文件来进行分析的，代码在数据分析.py这个文件里面

情感分析的代码如下：

```python
def emotion_analysis():
    df = pd.read_csv('数据.csv')
    new_df = df['情感分析'].value_counts()
    # 计算每行的占比

    plt.style.use('ggplot')
    plt.rcParams['font.sans-serif'] = ['SimHei']
    plt.figure(figsize=(9, 6), dpi=500)

    x_data = [str(x) for x in new_df.index]
    y_data = [int(x) for x in new_df.values]
    z_data = []
    for y in y_data:
        proportions = y / sum(y_data)
        proportions = str(float(round(proportions,4)) * 100) + "%"
        z_data.append(proportions)
    plt.pie(y_data, labels=x_data, startangle=0, autopct='%1.2f%%')
    plt.title('情感分析')
    # 添加图例
```

先统计好情感分析这一列，每个类型出现的频次，接着用matplotlib去进行饼图可视化处理，再去计算他们每个类型的占比

最后把这些相关的图以及数据都保存下来，图片和数据都有相对的文件名

接着对于地图可视化，这边采用的是pyecharts来进行可视化

先把一些空缺值进行删除，然后再去统计每个地区出现的频次，再去判断属于中国的地域，统计好之后，按照从大到小的顺序进行排序

```
def map_analyze():
    df = pd.read_csv('数据.csv')
    df = df.dropna(subset=['评论ip属地'],axis=0)
    new_df1 = df['评论ip属地'].value_counts()
    d1 = {}
    for x,y in zip(new_df1.index,new_df1.values):
        d1[x] = y


    ls = list(d1.items())
    ls.sort(key=lambda x:x[1],reverse=True)

    data = []
    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']
    x_data = []
    y_data = []
```

根据上面处理好的数据做可视化对比

```
c = (
    Geo()
        .add_schema(maptype="china")
        .add("中国地图", data)
        .set_series_opts(label_opts=opts.LabelOpts(is_show=False))
        .set_global_opts(
        title_opts=opts.TitleOpts(title="地域分析"),
        visualmap_opts=opts.VisualMapOpts(is_piecewise=True,max_=int(y_data[0])),
    )
        .render("地域分析.html")
)
```

对于统计供应商的回复时间，相对来说简单一点，用to_datetime进行时间数据处理，然后再进行相减即可，就可以知道回复的时间间隔了

```
def process_time():
    df = pd.read_csv('数据.csv')
    df = df.dropna(subset=['评论供应商回复时间'], axis=0)
    df['评论时间'] = pd.to_datetime(df['评论时间'])
    df['评论供应商回复时间'] = pd.to_datetime(df['评论供应商回复时间'])


    # 计算时间差
    df['time_difference(单位：天)'] =  df['评论供应商回复时间'] - df['评论时间']


    df.to_excel('时间相差数据.xlsx',index=False)
```

对于统计三大区域的三线表格

首先先进行地区划分，把地区划分为东南亚 日韩 欧美地区，接着去统计产品名称出现的频次，可以得知每个地区产品的数量多少

再根据产品的名称去做去重，来得知每个产品的平均票价是多少，以及总共的出现人数是多少，和产品满意度的平均值

把这些的做好之后，根据前面的数据，去统计不同地区的好评和差评的数量，接着去统计它们各种的占比，最后把它们的数据进行保留，用csv文件去存储三大地区的数据

```
def myd_data(area):
    df = pd.read_csv('数据.csv')
    df1 = df[df['地区分布'] == area]
    new_df = df1['产品名称'].value_counts()
    number_name = len(new_df)
    df2 = df1.drop_duplicates(subset=['产品名称'], keep='first')
    mean_proice =int(df2['票价'].mean())
    people_number = df2['出游人数'].sum()
    sum_myd = round(df2['产品满意度'].mean(),2)
    new_df2 = df1['情感分析'].value_counts()
    new_df2 = new_df2.sort_index()
    x_data = [str(x) for x in new_df2.index]
    y_data = [int(x) for x in new_df2.values]
    z_data = []
    for y in y_data:
        proportions = y / sum(y_data)
        proportions = str(float(round(proportions, 4)) * 100) + "%"
        z_data.append(proportions)
```

而出现类型的思路和上面一样，也是先区分地区，区分好地区，再去统计不同类型的数量，和计算它们的占比，最后把数据保留

评分的做法和上面一样，区别在于做可视化处理，具体可视化代码如下：

```
plt.style.use('ggplot')
plt.rcParams['font.sans-serif'] = ['SimHei']
plt.figure(figsize=(12, 9), dpi=500)
# 创建画布和子图
fig, ax1 = plt.subplots()

# 绘制折线图
ax1.plot(x_data, y_data, 'b-')
ax1.set_ylabel('数量',color='b')

# 创建第二个 y 轴
ax2 = ax1.twinx()

# 绘制柱状图
ax2.bar(x_data, y_data, alpha=0.5, color='r')
ax2.set_ylabel(' ',color='r')
ax2.set_xlabel('评分')
# 设置 x 轴和标题
plt.xticks(x_data, [f'{i}' for i in x_data], rotation=45)
plt.title('{}-评分趋势'.format(area))
```

关于LDA

这边的做法是首先先区分地区，把三大地区给区分，然后再根据每个地区，把它们的正面和负面的评论区分开

接着根据它们的分词数据来做LDA建模

建模的过程就是在LDA这个文件里面，怎么建模，已经单独用LDA.py编写了。

至于他的困惑度和一致性，解释如下：

困惑度（perplexity）和一致性（coherence）是选择最优LDA主题模型时常用的指标，通常使用困惑度和一致性来评估主题模型中隐含狄利克雷分布(Dirichlet Distribution)的质量和准确度。

困惑度主要用于评估主题模型对未标注语料库的拟合能力，即在该模型下，这些隐含主题与实际背景密切相关的程度。困惑度越小，表示模型对新语料库的拟合效果越好，有更好的预测效果。

一致性主要用于评估主题模型中被提取的主题的质量和稳定性，主要从主题词语（Topic Term）出现的频率和相关性来评估主题的一致性。一致性越高，表示这些词语在主题下存在更密切的相关性，反之则表明主题下的词汇较为松散，难以对主题进行概括。

使用困惑度、一致性的目的是为了选择最优LDA主题模型，从而获得更好的聚类效果和更高的可解释性。通过困惑度和一致性评估，可以比较不同的主题分布参数等模型参数，并选择效果最优的模型。

在实践中，对于LDA主题模型的选择，我们通常会使用不同的主题数，计算困惑度和一致性，并选择困惑度最小、一致性最高的主题数作为最佳模型参数，以获得更好的聚类效果和模型拟合度。

因此，通过困惑度、一致性指标的综合评估，可以帮助我们选择最佳的LDA主题模型，提高聚类效果和模型的可解释性。

一般而言，最佳主题数是由困惑度最小，一致性最高所对应的主题数确定的。根据提供的数据，可以看到困惑度在不同主题数下的变化趋势，以及随着主题数的增加，一致性的变化趋势。

通常情况下，困惑度会随着主题数的增加而减少，一致性则会随着主题数的增加先升高再下降。综合考虑困惑度和一致性指标，选择主题数使得困惑度最小、一致性最高，可以得到最优模型。

这边的主题最优都是通过编写代码，让模型自动选择最优的主题