

本次项目一共分为两个步骤

1、情感分析

先从情感分析开始说起

情感分析，我们首先是要进行数据清洗工作的

首先清洗一些无中文的无效词，例如标点符号，表情包，网站这种

主要代码是在这一块地方

```
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

def is_all_chinese(strs):
    for _char in strs:
        if not '\\u4e00' <= _char <= '\\u9fa5':
            return False
    return True
```

接着再把处理好的文本内容，用jieba库进行分词处理

分好的之后，我们调用百度开源的情感模型库，paddlehub

对应的文档介绍：https://www.paddlepaddle.org.cn/hubdetail?name=senta_bilstm&en_category=SentimentAnalysis

通过这个库去给文本进行打分处理

#	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	车型	品牌	评论用户	评论时间	评论目的	购车目的	空间评分	驾驶感受	油耗评分	外观评分	内饰评分	性价比评分	配置评分	全文	操控评分	舒适性评分	智能辅助	智能化评分	价格政策	分期	信息价值	emotion.type				
2	小型SUV	本田XR-V	青春那些事	#####	4.71	自驾游	5	5	5	5	5	4	5	43210km行驶里程7.0L百公里油耗14.59万	7.0L百公里油耗14.59万	7.0L百公里油耗14.59万	7.0L百公里油耗14.59万	7.0L百公里油耗14.59万	7.0L百公里油耗14.59万	7.0L百公里油耗14.59万	0.9988	非负				
3	小型SUV	本田XR-V	蓝色之海	#####	4.57	自驾游	4	5	5	5	5	3	5	54780km行驶里程7.8L百公里油耗15.29万	7.8L百公里油耗15.29万	7.8L百公里油耗15.29万	7.8L百公里油耗15.29万	7.8L百公里油耗15.29万	7.8L百公里油耗15.29万	7.8L百公里油耗15.29万	0.9981	非负				
4	小型SUV	本田XR-V	虚假的纯洁	#####	4.86	购物	5	5	5	5	4	5	5	51829km行驶里程14.59万	14.59万	14.59万	14.59万	14.59万	14.59万	14.59万	0.9924	非负				
5	小型SUV	本田XR-V	上宾易珍	#####	4.71	自驾游	4	5	5	5	5	4	5	53558km行驶里程7.7L百公里油耗15.29万	7.7L百公里油耗15.29万	7.7L百公里油耗15.29万	7.7L百公里油耗15.29万	7.7L百公里油耗15.29万	7.7L百公里油耗15.29万	7.7L百公里油耗15.29万	0.9968	非负				
6	小型SUV	本田XR-V	静等桃花开	#####	4.86	跑长途	5	4	5	5	5	5	5	55893km行驶里程7.5L百公里油耗14.59万	7.5L百公里油耗14.59万	7.5L百公里油耗14.59万	7.5L百公里油耗14.59万	7.5L百公里油耗14.59万	7.5L百公里油耗14.59万	7.5L百公里油耗14.59万	0.9988	非负				
7	小型SUV	本田XR-V	康大福	#####	4.71	约会	5	5	5	4	4	5	5	52015km行驶里程7.5L百公里油耗14.59万	7.5L百公里油耗14.59万	7.5L百公里油耗14.59万	7.5L百公里油耗14.59万	7.5L百公里油耗14.59万	7.5L百公里油耗14.59万	7.5L百公里油耗14.59万	0.9988	非负				
8	小型SUV	本田XR-V	碧翠岚笑	2023/2/9	4.86	跑长途	5	5	5	5	5	4	5	54312km行驶里程8.0L百公里油耗14.59万	8.0L百公里油耗14.59万	8.0L百公里油耗14.59万	8.0L百公里油耗14.59万	8.0L百公里油耗14.59万	8.0L百公里油耗14.59万	8.0L百公里油耗14.59万	0.9988	非负				
9	小型SUV	本田XR-V	樱村王	#####	4.86	跑长途	5	5	5	5	5	5	5	43100km行驶里程8.1L百公里油耗14.59万	8.1L百公里油耗14.59万	8.1L百公里油耗14.59万	8.1L百公里油耗14.59万	8.1L百公里油耗14.59万	8.1L百公里油耗14.59万	8.1L百公里油耗14.59万	0.9988	非负				
10	小型SUV	本田XR-V	清逸之风	#####	4.86	自驾游	5	5	5	5	5	5	5	51215km行驶里程7.9L百公里油耗14.59万	7.9L百公里油耗14.59万	7.9L百公里油耗14.59万	7.9L百公里油耗14.59万	7.9L百公里油耗14.59万	7.9L百公里油耗14.59万	7.9L百公里油耗14.59万	0.9951	非负				
11	小型SUV	本田XR-V	有问题就修	2023/1/7	4.71	上下班	5	4	5	5	5	5	5	52913km行驶里程15.29万	15.29万	15.29万	15.29万	15.29万	15.29万	15.29万	0.9978	非负				
12	小型SUV	本田XR-V	澳阳XV	#####	4.86	购物	4	5	5	5	5	5	5	52251km行驶里程14.59万	14.59万	14.59万	14.59万	14.59万	14.59万	14.59万	0.9971	非负				
13	小型SUV	本田XR-V	谁是平头	#####	4.86	上下班	5	4	5	5	5	5	5	52125km行驶里程14.59万	14.59万	14.59万	14.59万	14.59万	14.59万	14.59万	0.998	非负				
14	小型SUV	本田XR-V	粉红的海兰	#####	4.86	购物	5	5	5	5	4	5	5	53351km行驶里程13.29万	13.29万	13.29万	13.29万	13.29万	13.29万	13.29万	0.9958	非负				
15	小型SUV	本田XR-V	这个世界才	#####	4.86	购物	5	5	5	5	5	5	5	43082km行驶里程15.29万	15.29万	15.29万	15.29万	15.29万	15.29万	15.29万	0.9901	非负				
16	小型SUV	本田XR-V	金鼎合金	#####	4.86	购物	5	5	5	4	5	5	5	53341km行驶里程14.59万	14.59万	14.59万	14.59万	14.59万	14.59万	14.59万	0.9978	非负				
17	小型SUV	本田XR-V	宇航员之家	#####	4.86	购物	5	5	4	5	5	5	5	53300km行驶里程7.8L百公里油耗15.29万	7.8L百公里油耗15.29万	7.8L百公里油耗15.29万	7.8L百公里油耗15.29万	7.8L百公里油耗15.29万	7.8L百公里油耗15.29万	7.8L百公里油耗15.29万	0.9971	非负				
18	小型SUV	本田XR-V	两盏小灯	#####	4.86	购物	5	5	5	5	5	5	5	41291km行驶里程14.59万	14.59万	14.59万	14.59万	14.59万	14.59万	14.59万	0.9988	非负				
19	小型SUV	本田XR-V	风里的烟圈	#####	4.86	购物	4	5	5	5	5	5	5	53281km行驶里程7.1L百公里油耗14.59万	7.1L百公里油耗14.59万	7.1L百公里油耗14.59万	7.1L百公里油耗14.59万	7.1L百公里油耗14.59万	7.1L百公里油耗14.59万	7.1L百公里油耗14.59万	0.9954	非负				
20	小型SUV	本田XR-V	购车后返	#####	4.86	购物	5	5	5	4	5	5	5	51550km行驶里程7.2L百公里油耗14.59万	7.2L百公里油耗14.59万	7.2L百公里油耗14.59万	7.2L百公里油耗14.59万	7.2L百公里油耗14.59万	7.2L百公里油耗14.59万	7.2L百公里油耗14.59万	0.9994	非负				
21	小型SUV	本田XR-V	天津车友S	#####	4.75	自驾游	5	4	5	5	5	4	5	800km行驶		5						0.9971	非负			
22	小型SUV	本田XR-V	羊了个羊	#####	4.71	自驾游	5	4	5	5	4	5	5	55266km行驶里程7.2L百公里油耗15.89万	7.2L百公里油耗15.89万	7.2L百公里油耗15.89万	7.2L百公里油耗15.89万	7.2L百公里油耗15.89万	7.2L百公里油耗15.89万	7.2L百公里油耗15.89万	0.9961	非负				
23	小型SUV	本田XR-V	夜空繁星	#####	3.98	上下班	2	5	4	5	5	5	5	11296km行驶里程9.3L百公里油耗12.50万	9.3L百公里油耗12.50万	9.3L百公里油耗12.50万	9.3L百公里油耗12.50万	9.3L百公里油耗12.50万	9.3L百公里油耗12.50万	9.3L百公里油耗12.50万	0.9931	非负				
24	小型SUV	本田XR-V	真真车友	#####	4.71	接送小孩	5	4	5	5	4	5	5	53250km行驶里程7.5L百公里油耗15.25万	7.5L百公里油耗15.25万	7.5L百公里油耗15.25万	7.5L百公里油耗15.25万	7.5L百公里油耗15.25万	7.5L百公里油耗15.25万	7.5L百公里油耗15.25万	0.9968	非负				
25	小型SUV	本田XR-V	Onan146	#####	4.71	商务差旅	5	5	4	5	4	5	5	53200km行驶里程6.6L百公里油耗11.44万	6.6L百公里油耗11.44万	6.6L百公里油耗11.44万	6.6L百公里油耗11.44万	6.6L百公里油耗11.44万	6.6L百公里油耗11.44万	6.6L百公里油耗11.44万	0.9988	非负				
26	小型SUV	本田XR-V	nmqzhd	#####	4.86	商务差旅	5	4	5	5	5	5	5	51900km行驶里程9.3L百公里油耗13.20万	9.3L百公里油耗13.20万	9.3L百公里油耗13.20万	9.3L百公里油耗13.20万	9.3L百公里油耗13.20万	9.3L百公里油耗13.20万	9.3L百公里油耗13.20万	0.9988	非负				
27	小型SUV	本田XR-V	即聊车友	#####	4.71	约会	5	5	3	5	5	5	5	511267km行驶里程9.1L百公里油耗14.88万	9.1L百公里油耗14.88万	9.1L百公里油耗14.88万	9.1L百公里油耗14.88万	9.1L百公里油耗14.88万	9.1L百公里油耗14.88万	9.1L百公里油耗14.88万	0.9988	非负				
28	小型SUV	本田XR-V	应和239	#####	4.57	自驾游	4	4	5	5	5	5	5	46700km行驶里程7.0L百公里油耗16.00万	7.0L百公里油耗16.00万	7.0L百公里油耗16.00万	7.0L百公里油耗16.00万	7.0L百公里油耗16.00万	7.0L百公里油耗16.00万	7.0L百公里油耗16.00万	0.9711	非负				
29	小型SUV	本田XR-V	那台车卖	#####	4.71	约会	5	4	5	5	5	4	5	51711km行驶里程7.9L百公里油耗13.00万	7.9L百公里油耗13.00万	7.9L百公里油耗13.00万	7.9L百公里油耗13.00万	7.9L百公里油耗13.00万	7.9L百公里油耗13.00万	7.9L百公里油耗13.00万	0.9961	非负				
30	小型SUV	本田XR-V	欠你的幸	#####	4.88	自驾游	5	5	5	5	5	5	5	9702km行		5						0.9982	非负			
31	小型SUV	本田XR-V	蓝颜知己	#####	4.88	自驾游	5	5	5	5	4	5	5	12415km行		5						0.9988	非负			
32	小型SUV	本田XR-V	生途b2	2022/4/6	4.88	自驾游	5	5	5	5	5	5	5	15421km行		5						0.9958	非负			
33	小型SUV	本田XR-V	PingShan	2022/4/6	4.75	跑长途	5	5	5	5	4	5	5	10000km行		5						0.9958	非负			
34	小型SUV	本田XR-V	有人打哈	#####	4.13	上下班	5	4	5	5	1	4	5	5368km行		5						0.0028	负面			
35	小型SUV	本田XR-V	超人强0o	#####	4.25	自驾游	4	4	5	5	4	4	5	9256km行		4						0.9977	非负			
36	小型SUV	本田XR-V	小开升052	#####	4.38	自驾游	5	5	5	5	1	5	5	4120km行		5						0.998	非负			
37	小型SUV	本田XR-V	yanghuale	#####	4.88	商务差旅	5	5	5	5	4	5	5	3300km行		5						0.9988	非负			
38	小型SUV	本田XR-V	可卿卿	2021/9/4	4.38	上下班	5	5	4	5	5	3	4	1100km行		5						0.9974	非负			
39	小型SUV	本田XR-V	ULUNRV	2021/8/4	4.88	商务差旅	5	5	5	5	4	5	5	5000km行		5						0.998	非负			
40	小型SUV	本田XR-V	KelinYui	#####	4.88	商务差旅	5	5	5	5	5	5	5	1268km行		5						0.9988	非负			
41	小型SUV	本田XR-V	Nq4455	#####	4.75	自驾游	5	5	5	5	3	5	5	5000km行		5						0.9228	非负			
42	小型SUV	本田XR-V	商务车友	2021/6/5	4.75	拉货	5	5	5	5	4	5	5	13500km行		5						0.9966	非负			
data 请自行输入																										

这些就是对应的情感分值了

然后再写一个函数，去对这些分值进行分类任务，分值小于等于0.35就是判断为负面，其他的就是非负

```
def main1(x):  
    x1 = float(x)  
    if x1 ≤ 0.35:  
        return '负面'  
    else:  
        return '非负'  
data['emotion_type'] = data['情感分值'].apply(main1)
```

上面的步骤处理好之后，我们回归到词本身的分析

先去查看词云图，代码在这一块地方

```
data = pd.read_csv('./data/data_情感分析.csv')  
text1 = []  
for x in data['分词']:  
    x = str(x).split(" ")  
    for j in x:  
        text1.append(j)  
text2 = ' '.join(text1)  
stylecloud.gen_stylecloud(text=text2, max_words=150,  
                           collocations=False,  
                           font_path='simhei.ttf',  
                           icon_name='fas fa-globe',  
                           size=500,  
                           # palette='matplotlib.Inferno_9',  
                           output_name='./data/词云图.png')  
Image(filename='./data/词云图.png')
```

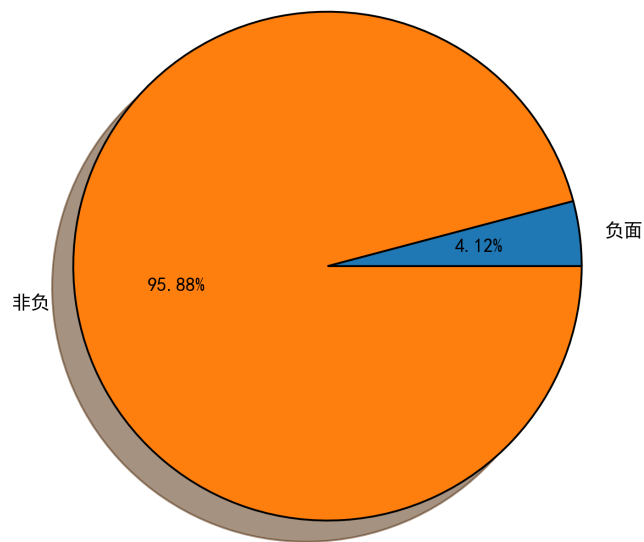
效果图展示



接着我们再去看看情感占比的多少

代码在这一块

```
def emotion_score():
    data = pd.read_csv('./data/data-情感分析.csv')
    new_df = data['emotion_type'].value_counts()
    new_df.sort_index(inplace=True)
    x_data = [x for x in new_df.index]
    y_data = [y for y in new_df.values]
    plt.figure(figsize=(9, 6), dpi=300)
    plt.rcParams['font.sans-serif'] = ['SimHei']
    plt.pie(y_data, labels=x_data, shadow=True, startangle=0, autopct='%1.2f%%',
            wedgeprops={'edgecolor': 'black'})
    plt.savefig('./data/情感分布.png')
```



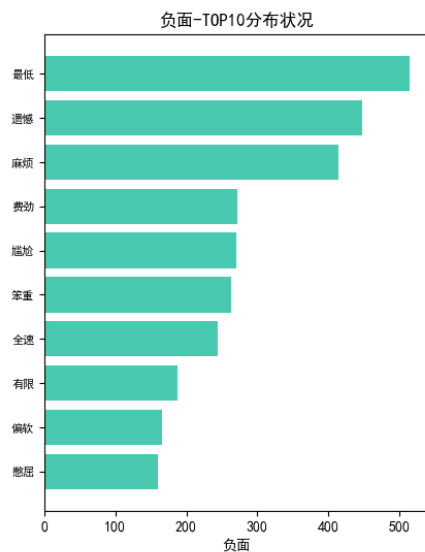
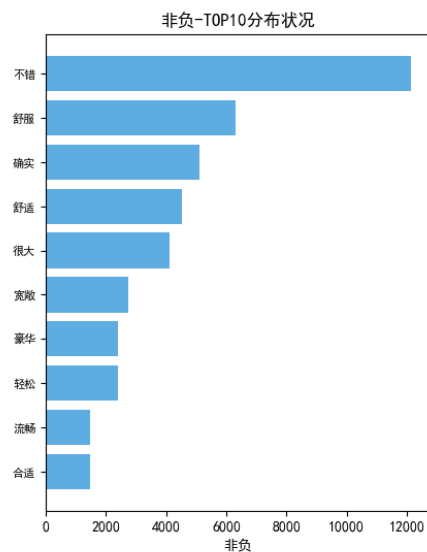
观察了整体的情况之后，我们再去情感词汇处理，主要代码在这里，和上面的区别在于是划分形容词的，就只要形容词，通过形容词去看看大致的情感方向

```
def emotion_word():  
    data = pd.read_csv('./data/data_情感分析.csv')  
    list1 = []  
    for n in data['分词']:  
        n = str(n).split(" ")  
        for i in n:  
            list1.append(i)  
    counts = {}  
    for l in list1:  
        res = posseg.cut(l)  
        for word, flag in res:  
            if flag == 'Ag' or flag == 'a' or flag == 'ad' or flag == 'an':  
                if len(word) >= 2:  
                    counts[word] = counts.get(word, 0) + 1
```

处理好之后

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	word	counts	情感分	emotion_type																						
2	不错	12142	0.9475	非负																						
3	舒服	6301	0.957	非负																						
4	确实	5092	0.9207	非负																						
5	舒适	4512	0.9504	非负																						
6	很大	4128	0.8233	非负																						
7	宽敞	2726	0.9588	非负																						
8	豪华	2414	0.8384	非负																						
9	轻松	2407	0.9532	非负																						
10	流畅	1479	0.9679	非负																						
11	合适	1475	0.8739	非负																						
12	简单	1453	0.498	非负																						
13	完美	1357	0.79	非负																						
14	平均	1122	0.7409	非负																						
15	平常	1099	0.5022	非负																						
16	精致	1080	0.9439	非负																						
17	便宜	1067	0.7884	非负																						
18	舒适	1059	0.9305	非负																						
19	漂亮	1053	0.9658	非负																						
20	优秀	1026	0.9726	非负																						
21	充足	1000	0.9033	非负																						
22	稳定	974	0.9425	非负																						
23	年轻	945	0.7794	非负																						
24	充满	940	0.8744	非负																						
25	清晰	927	0.9393	非负																						
26	好看	919	0.9803	非负																						
27	稳重	851	0.9677	非负																						
28	强劲	806	0.9363	非负																						
29	平稳	759	0.9742	非负																						
30	简洁	685	0.933	非负																						
31	很棒	685	0.9928	非负																						
32	惊喜	680	0.9785	非负																						
33	扎实	659	0.9766	非负																						
34	友好	589	0.7687	非负																						
35	柔软	585	0.9399	非负																						
36	最低	515	0.329	负面																						
37	沉稳	511	0.9483	非负																						
38	很小	504	0.3974	负面																						
39	充沛	488	0.9695	非负																						
40	遗憾	447	0.3018	负面																						
41	软硬	435	0.898	非负																						
42	宽大	428	0.9192	非负																						

再去把它们非负和负面的排名前10的情感词汇用可视化的方式呈现



2、LDA主题建模

LDA参考文献: <https://zhuanlan.zhihu.com/p/75222819>

<https://zhuanlan.zhihu.com/p/76636216>

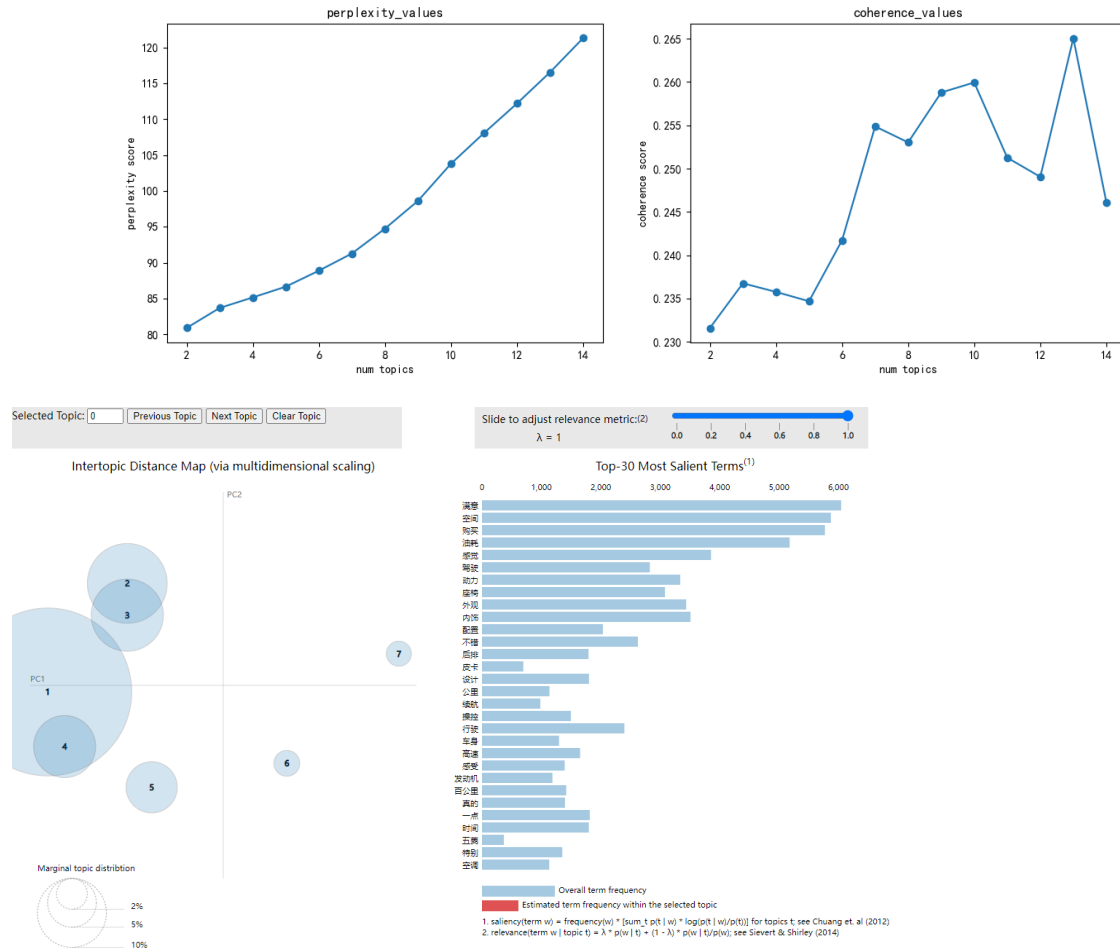
这里我们的步骤和上面和一样

首先我们还是先分词，把无效词给去除掉

接着我们开始构造主题数，寻找最优主题数

LDA模型中，困惑度 (perplexity) 和一致性 (coherence) 是两个常用的指标来判断主题数的优劣。其中，困惑度是一种常用的评价主题模型好坏的指标，它是对于给定的测试集，模型越好，困惑度越低。而一致性则是通过计算每个主题下词语的相似度来衡量主题质量的指标。在确定最优主题数时，可以通过比较不同主题数下的困惑度和一致性来选择最优的主题数

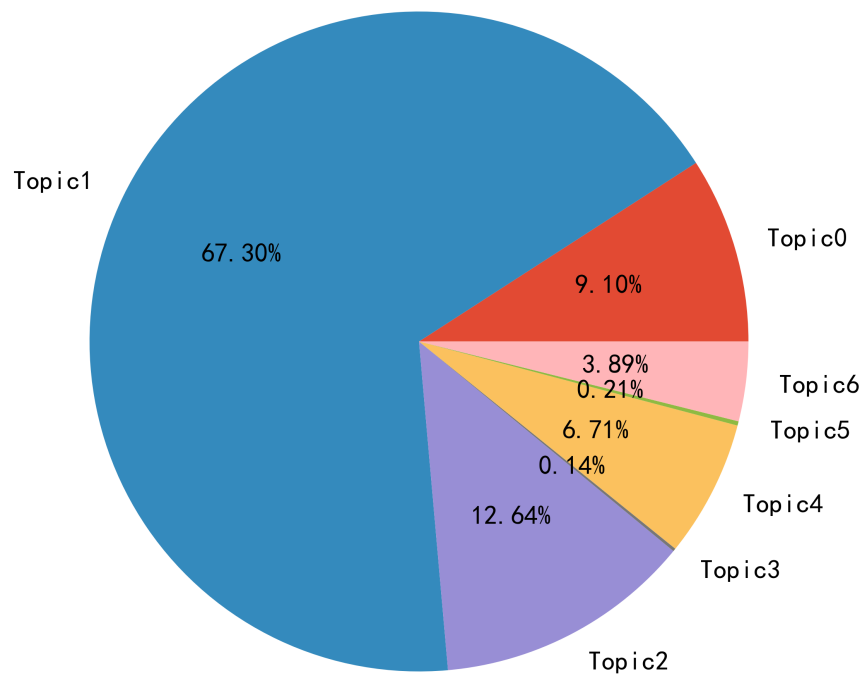
这里在困惑度保持上升的趋势下，我们则选择一致性来作为判断标准，选择它的峰值，也就是13或者7作为主题建模，在对比过后，选择7的主题优于选择13的主题，因此，我们的主题数选择7



接着我们再去判断主题的发布情况，这是主要的代码

```
#绘制主题强度饼图
def plt_pie():
    plt.style.use('ggplot')
    plt.rcParams['font.sans-serif'] = ['SimHei']
    plt.figure(dpi=500)
    df = pd.read_csv('./data/特征词.csv')
    x_data = list(df['所属主题'])
    y_data = list(df['文章数量'])
    plt.pie(y_data, labels=x_data, startangle=0, autopct='%1.2f%%')
    plt.title('主题强度')
    plt.tight_layout()
    plt.savefig('./data/主题强度.png')
```

主题强度



看的出来，主题的分布情况其实也和气泡的大小有关，气泡越大，那么所占的比重也就越大

然后，我们去把主题的所对应的主题词找出来，主要代码如下

```
c2 = c1.findall(ci)
list_word.append(c2)
c3 = ','.join(c2)

c4 = re.compile(".*?(\\d+).?")
c5 = c4.findall(ci)
for c in c5[::1]:
    if c != "0":
        gailv = str(0) + '.' + str(c)
        list_gailv.append(gailv)
list_gailv1.append(list_gailv)
list_gailv = []
zt = "Topic" + str(w[0])
topic.append(zt)
quanzhong.append(c3)

#把上面权重的统计算好之后，进行保存为csv文件
df2 = pd.DataFrame()
for j,k,l in zip(topic,list_gailv1,list_word):
    df2['{}-主题词'.format(j)] = l
    df2['{}-权重'.format(j)] = k
df2.to_csv('./data/主题词分布表.csv', encoding='utf-8-sig', index=False)
```

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
1	Topic0-主	Topic0-权	Topic1-主	Topic1-权	Topic2-主	Topic2-权	Topic3-主	Topic3-权	Topic4-主	Topic4-权	Topic5-主	Topic5-权	Topic6-主	Topic6-权	量											
2	满意	0.031	满意	0.035	满意	0.03	小蚂蚁	0.016	满意	0.032	满意	0.019	油耗	0.032												
3	购买	0.03	空间	0.034	空间	0.03	空间	0.016	空间	0.031	购买	0.018	满意	0.027												
4	空间	0.029	购买	0.033	购买	0.027	满意	0.016	油耗	0.03	空间	0.018	购买	0.024												
5	感觉	0.02	油耗	0.033	座椅	0.021	购买	0.015	购买	0.029	油耗	0.015	空间	0.024												
6	油耗	0.019	感觉	0.022	公里	0.017	水果	0.014	感觉	0.023	距离	0.013	感觉	0.017												
7	外观	0.016	动力	0.022	感觉	0.016	提高	0.01	皮卡	0.02	动力	0.013	动力	0.017												
8	座椅	0.015	内饰	0.021	内饰	0.016	耐用	0.009	外观	0.019	驾驶	0.013	驾驶	0.016												
9	内饰	0.015	外观	0.021	续航	0.015	油耗	0.009	动力	0.018	腰部	0.011	座椅	0.016												
10	后排	0.014	座椅	0.017	油耗	0.013	配置	0.009	驾驶	0.018	座椅	0.011	内饰	0.016												
11	不错	0.013	不错	0.016	外观	0.013	感觉	0.009	内饰	0.017	外观	0.01	后排	0.015												
12	驾驶	0.012	驾驶	0.016	助力	0.012	麦卡尤	0.009	不错	0.014	感觉	0.01	外观	0.013												
13	时间	0.011	行驶	0.015	时间	0.012	座椅	0.008	配置	0.014	内饰	0.01	行驶	0.012												
14	喜欢	0.011	配置	0.012	驾驶	0.012	驾驶	0.008	车身	0.012	设计	0.009	发动机	0.011												
15	续航	0.011	一点	0.011	空调	0.012	外观	0.008	设计	0.012	驾驶员	0.008	不错	0.01												
16	一点	0.01	设计	0.011	行驶	0.011	公里	0.007	座椅	0.011	不错	0.008	配置	0.01												
17	公里	0.009	性价比	0.011	五菱	0.011	方向	0.006	行驶	0.011	五菱	0.008	操控	0.01												
18	动力	0.009	时间	0.01	高速	0.011	内饰	0.006	版本	0.011	提供	0.008	百公里	0.01												
19	后排	0.009	车子	0.01	动力	0.009	后门	0.006	感受	0.011	车身	0.007	高速	0.009												
20	配置	0.008	后排	0.01	真的	0.009	块钱	0.006	操控	0.01	一排	0.007	感受	0.009												
21	舒服	0.008	舒适性	0.01	充电	0.009	续航	0.006	后排	0.01	钓鱼	0.006	前排	0.007												
22																										
23																										
24																										
25																										
26																										
27																										
28																										
29																										
30																										
31																										
32																										
33																										
34																										
35																										
36																										
37																										
38																										
39																										
40																										
41																										
42																										

这是其表现形式

错误集展示

```
LDA建模 x
File "C:\Users\y_yuhaozeng\Desktop\数据分析\LDA建模.py", line 210, in plt_pie
df = pd.read_csv('特征词.csv')
File "D:\Users\y_yuhaozeng\Anaconda3\lib\site-packages\pandas\util\decorators.py", line 311, in wrapper
return func(*args, **kwargs)
File "D:\Users\y_yuhaozeng\Anaconda3\lib\site-packages\pandas\io\parsers\readers.py", line 678, in read_csv
return _read(filepath_or_buffer, kwds)
File "D:\Users\y_yuhaozeng\Anaconda3\lib\site-packages\pandas\io\parsers\readers.py", line 575, in _read
parser = TextFileReader(filepath_or_buffer, **kwds)
File "D:\Users\y_yuhaozeng\Anaconda3\lib\site-packages\pandas\io\parsers\readers.py", line 932, in __init__
self._engine = self._make_engine(f, self._engine)
File "D:\Users\y_yuhaozeng\Anaconda3\lib\site-packages\pandas\io\parsers\readers.py", line 1216, in _make_engine
self._handles = get_handle( # type: ignore[call-overload]
File "D:\Users\y_yuhaozeng\Anaconda3\lib\site-packages\pandas\io\common.py", line 786, in get_handle
handle = open(
FileNotFoundError: [Errno 2] No such file or directory: '特征词.csv'
```

这里在运行的时候，文件的位置填写错误导致，代码无法读取

这一篇空白是因为Ida出现问题，要外网才能进行访问，对此的解决策略就是去网上下载对应css和js文件来解决


```
<link rel="stylesheet" type="text/css" href="https://cdn.rawgit.com/bmabey/pyLDavis/files/ldavis.v1.0.0.css">

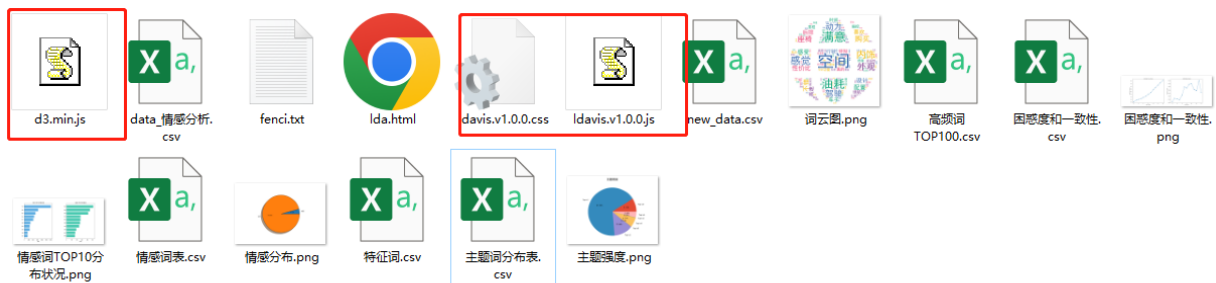
<div id="ldavis_el503214374660580809067974442"></div>
<script type="text/javascript">

var ldavis_el503214374660580809067974442_data = {"mdsDat": {"x": [-0.09271561703022393, -0.033407871927735276, -0.03337490451051042, -0.08012709024023142, -0.01519

function LDavis_load_lib(url, callback){
  var s = document.createElement('script');
  s.src = url;
  s.async = true;
  s.onreadystatechange = s.onload = callback;
  s.onerror = function(){console.warn("failed to load library " + url)};
  document.getElementsByTagName("head")[0].appendChild(s);
}

if(typeof(LDavis) !== "undefined"){
  // already loaded: just create the visualization
}
```

然后去到lda.html这个文件，修改源码，把读取的路径进行修改，其对应的文件



```
情感分析 x
↑
↓
Traceback (most recent call last):
  File "C:\Users\v_yuhaozeng\Desktop\数据分析\情感分析.py", line 220, in <module>
    snownlp_fx()
  File "C:\Users\v_yuhaozeng\Desktop\数据分析\情感分析.py", line 66, in snownlp_fx
    data['全文'] = data['全文'].apply(emjio_tihuan)
  File "D:\Users\v_yuhaozeng\Anaconda3\lib\site-packages\pandas\core\series.py", line 958, in __getitem__
    return self._get_value(key)
  File "D:\Users\v_yuhaozeng\Anaconda3\lib\site-packages\pandas\core\series.py", line 1069, in _get_value
    loc = self.index.get_loc(label)
  File "D:\Users\v_yuhaozeng\Anaconda3\lib\site-packages\pandas\core\indexes\base.py", line 3631, in get_loc
    raise KeyError(key) from err
KeyError: '全文'
```

这里是代码书写错误的问题，找到错误源之后，重新编写代码即可，

这里也是，这两个地方都是代码书写问题，修改命名规则即可

```
情感分析
Loading model cost 0.581 seconds.
[2023-04-03 11:14:07,515] [ DEBUG] __init__.py:164 - Loading model cost 0.581 seconds.
Prefix dict has been built successfully.
[2023-04-03 11:14:07,515] [ DEBUG] __init__.py:166 - Prefix dict has been built successfully.
[2023-04-03 11:16:10,892] [ WARNING] - The _initialize method in HubModule will soon be deprecated, you can use the __init__() to handle the initialization of the object
W0403 11:16:11.909073 2332 analysis_predictor.cc:2166] Deprecated. Please use CreatePredictor instead.
[2023-04-03 11:21:34,875] [ WARNING] - The _initialize method in HubModule will soon be deprecated, you can use the __init__() to handle the initialization of the object
Traceback (most recent call last):
  File "C:\Users\v_yuhaozeng\Desktop\数据分析\情感分析.py", line 223, in <module>
    emotion_word()
  File "C:\Users\v_yuhaozeng\Desktop\数据分析\情感分析.py", line 186, in emotion_word
    z_data = list(new_df.index)
NameError: name 'new_df' is not defined
```