数据科学导论最终报告

实验环境简述

在本项目中采用的 python 版本为 Python 3.9.7，pandas 版本为 1.3.4。
在本项目中所用到的 python 库包括：（用到直接在以下添加）

1. seaborn version：0.11.2
2. numpy version：1.20.3
3. jieba version：0.42.1
4. wordcloud version：1.8.1
5. matplotlib version：3.3.4
6. pillow version：8.2.0

Python 日志分析算法简述

在本部分根据任务的先后顺序对每个部分的算法进行简述。

1. 搜索关键词统计

首先，调用 jieba 库对搜索词条进行分词，并根据已有的中文停用词表，去除分词后包含的停用词。最后再将所有的词汇总成一个列表，对列表中的元素进行归并处理，并按照频率从高到低排序，即可得到不同关键词的分布。取列表中前 10 个关键词得到 top 10 关键词。

注：在检查了分词结果后，于常见中文停用词表中额外加入了 http，www 等网页搜索相关词汇以及一些误切分内容，同时基于本数据集往 jieba 中添加了自己的词典。

2. Word Cloud 可视化

按照词频从高到低，对第一部分得到的关键词进行排序，并提取排名前 100 的词语和对应词频。

调用 wordcloud 库，根据 wordcloud 库中的参数设置，采用白云的图片作为词云背景图片，并设置字体参数，导入关键词。

3. 搜索点击次数统计

在此部分先对原有用户搜索记录进行操作，将用户 id 和搜索词条归并形成列表，再根据列表元素进行同类归并，统计每一个相同列表的个数。最后，再将每一个列表拆分成 id 和查询词，得到最终结果。

4. Order 分析

首先利用 numpy 中的 mean 函数计算日志数据中 order 列（用户点击的顺序号）的平均值，然后使用 dataframe 中的 value_counts 功能统计每个顺序号出现的频次并排序，取前 10 的顺序号保存至 click_order_top10.txt。最后利用 matplotlib 中的 pie 函数绘制排名前十位的 Order 和频数的数据分布饼状图。

5. 访问时间分析

首先，用 split 函数提取出“访问时间”列中的“小时”，并用其替代原“访问时间”列。其次，用 groupby 函数依据“访问时间”重组 dataframe。接着，得到重组后每一组对应的时间段和用户搜索次数，将时间段和用户搜索次数这两列合并为新的 dataframe，存入 search_time_analysis.txt。最后，画出对应的柱状图，在极大极小值点上标注数值，并存为 search_time_analysis.png

6．Bonus—用户平均点击量分析

首先，去掉“用户点击的 URL”列，用 groupby（）依据“用户 ID”分组。接着，计算每个用户 ID 对应的总点击数和总查询词个数，用总点击数/总查询词个数计算出每个用户 ID 在单个查询词上的平均点击数。其次，分别统计平均点击量=1，>=3，>=5，>=7，>=10 的用户数和用户占比，将这些数据合并为新的 dataframe，存入 rep_click.csv

7．Bonus—URL 点击搜索分析

URL 地址代表用户搜索点击的具体网址。即便是一样的搜索词，用户也可能选择点进不同的词条，即可能产生不一样的 URL。在此处对 URL 地址使用 groupby 函数，并计算不同 URL 的频数，对其分布进行可视化分析。

数据处理效率及运行结果分析

数据处理效率

尽管数据集规模较大，但通过调用库函数和降低循环复杂度，依然获得了不错的运行速度。

运行结果分析

1．关键词提取

这里提取了前 10 个频率最高的关键词

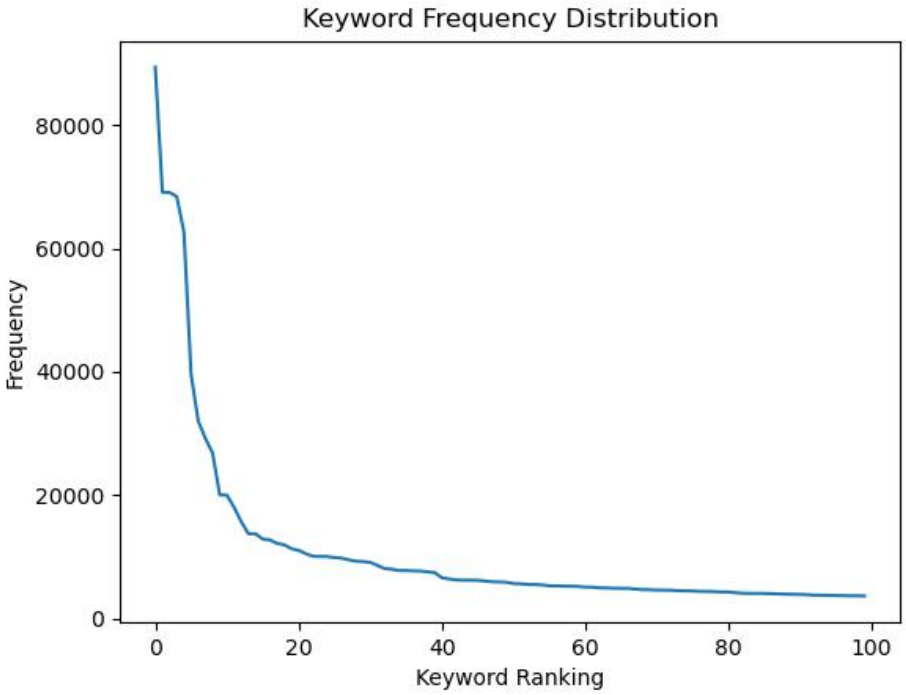
1	地震	89422
2	救灾物资	69092
3	哄抢	69084
4	汶川	68422
5	原因	62688
6	下载	39600
7	图片	32007
8	视频	29173
9	暗娼	26825
10	名单	20026

在此处我们统计了所有的关键词里面出现频率最高的十个单词。这里可以看到，排名最高的'地震'在此处频数约等于 9 万（这与 2008 年的汶川大地震背景相呼应），然而排名第十的词语频数很快就衰减到了 2 万左右。考虑所有单词的分布，我们可以对表格得到如下统计结果：

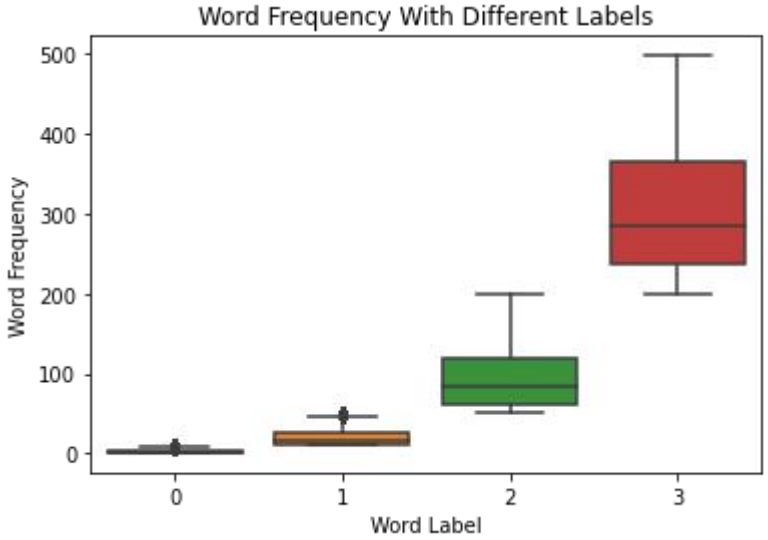
	Frequency
count	117339.000000
mean	33.018425
std	565.349828
min	1.000000
25%	1.000000
50%	3.000000
75%	10.000000
max	89422.000000

根据以上统计数据可以看到，词频的平均值为 33，远远小于 top10 关键词的频数，且有 25%的关键词词频为 1，有 75%的关键词词频小于 10，均属于低频数据。该现象满足 zipf 分布，属于幂律的表现形式之一，即 80%的搜索频数是由 20%的关键词所贡献的。若在后续想要完成对不同搜索词的特征分类，则应当重点关注低频词，因为相对而言，高频词在此处容易被

区分。



在此处继续对搜索关键词进行可视化分析，先对其进行粗略的标签分类：在此处仅考虑频率小于 500 的单词（由以上分析可知高频词汇已经不需要进行进一步的分析），并且按照频率在 1—10，10—50，50—200，200—500 分别赋予 3—0 四种标签。



2．云图

在之前得到的搜索词频数列表中，取频数最高的 100 个词，将其作为参数绘制云图，并采用白色背景与云朵形状。词云中较为明显的词条，有汶川、地震、物资等。结合该日志数据的年份“2008 年”，可以知道是汶川大地震影响了人们的搜索行为。例如‘物资’，‘哄抢’

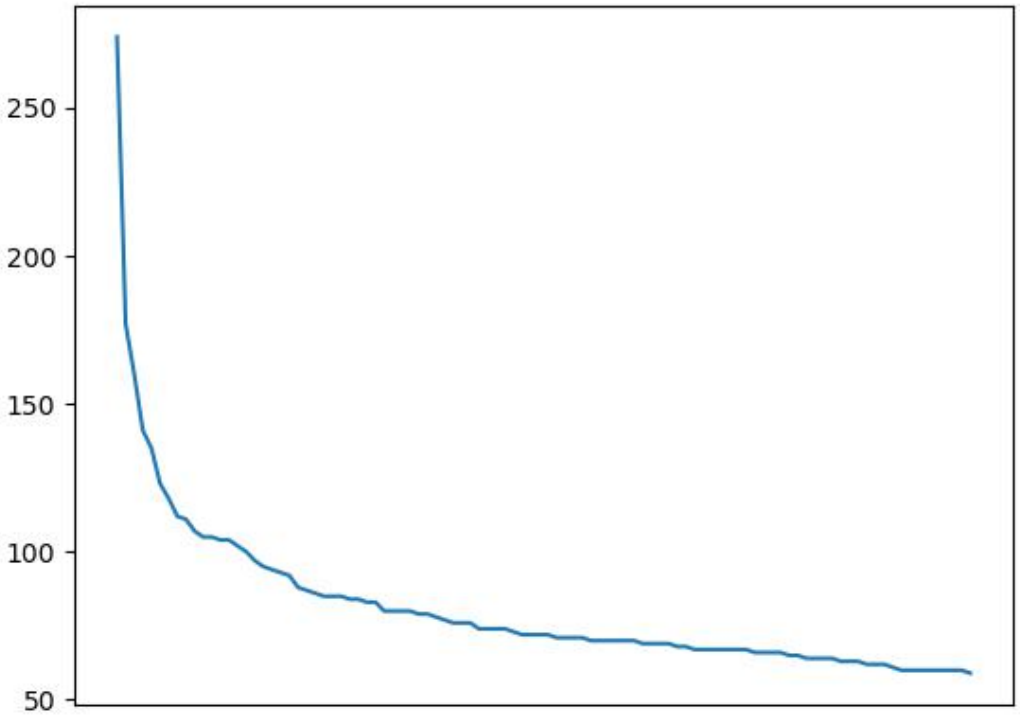
都和地震过后的救援工作有关，以及‘原因’这里推测用户都在搜索汶川大地震发生的原因。由此可见，搜索关键词较多受到高热度事件的影响，大多数搜索关键词都聚焦于年度热点事件。

3．查询词分析

1	7822241147182134	free+girls	274
2	9165829432475153	人妖摄影	177
3	49180486532951556	babes	160
4	19447614244798927	绳艺kb视频	141
5	7076435807359547	库娃+三围	135
6	900755558064074	liuhecai	123
7	1756178764125793	屋面种植土	118
8	0767168563136269	沈阳家电维修+亨达	112
9	5515612701706876	玄幻小说	111
10	48453866664474126	free+movie	107

以上为 top10 查询词。在此处排名最高的词条被一个用户查询了 274 次，而排名第十的只有 107，可见查询词条和关键词一样都有着相似的 zipf 分布的性质，即频数衰减较快，第十名搜索词只有第一名的一半。

绘制次数最高的前 100 项的频度分布图如图示。



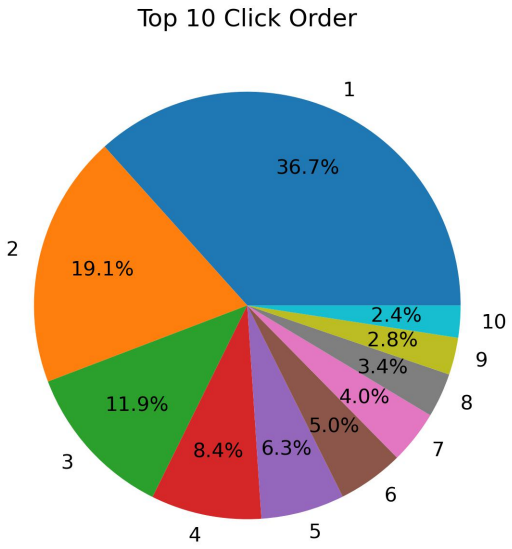
4．order 分析

用户点击的顺序号平均值为 7.715943730194448

频数排序前十的顺序号及其频数如下：

1	511806
2	267076
3	165821
4	116992
5	88327
6	69848
7	56197
8	46796
9	39280
10	33799

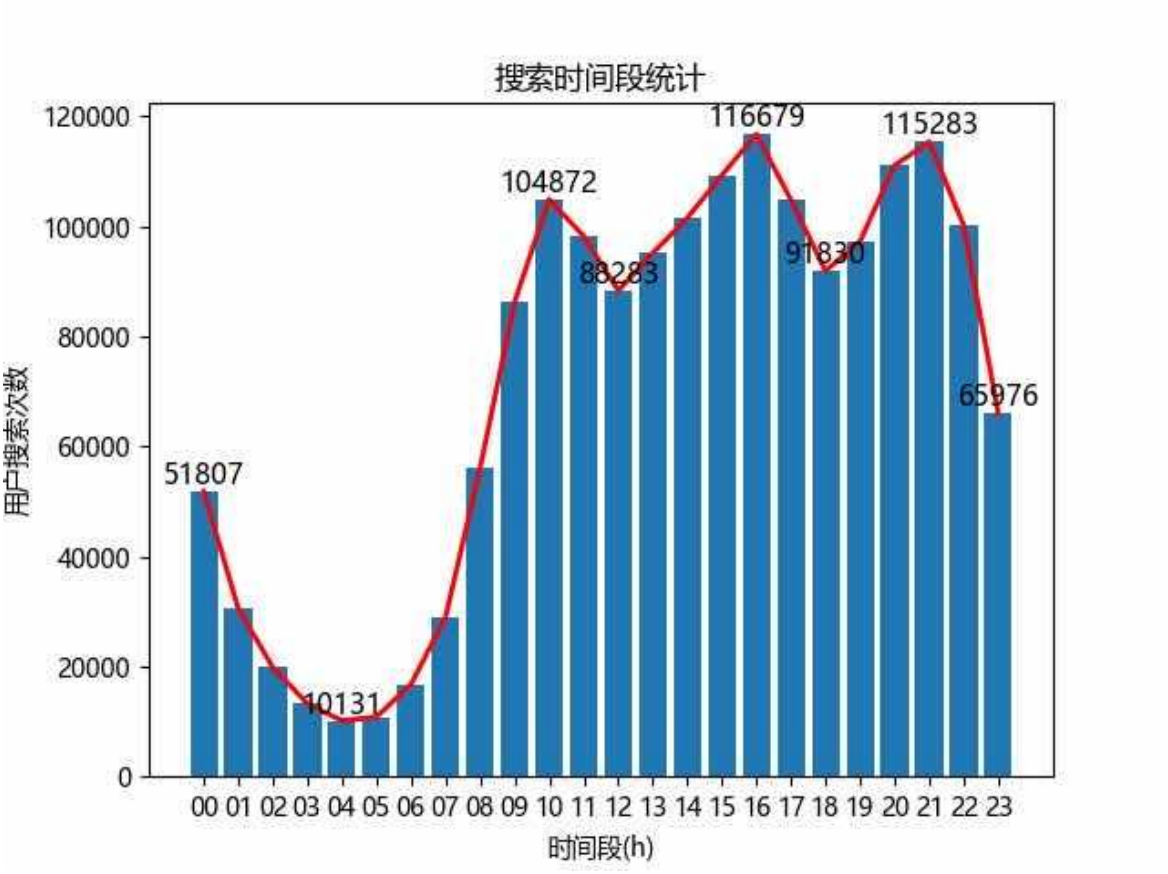
对其绘制数据分布饼状图如下：



通过饼状图可以直观看出，顺序越靠后的搜索结果用户点击的频率越低，点击第一个和第二个结果的次数占据了一半以上。这与大多数用户的使用习惯相一致，即用户通常会按顺序点击，直到得到了满意的结果。

5．**时间段分析**

时间段	用户搜索次数
00	51807
01	30498
02	19813
03	13239
04	10131
05	10838
06	16733
07	28936
08	56032
09	86227
10	104872
11	98135
12	88283
13	95095
14	101455
15	109255
16	116679
17	104756
18	91830
19	97247
20	111022
21	115283
22	100122
23	65976



由柱状图可以看出：

1. 用户搜索主要集中在早上八点至凌晨十二点，与大众作息时间相符
2. 搜索量的几个峰值出现在中午十一十二点，下午四五点以及晚上八九点，再结合极小值分析可以发现，饭点的用户搜索量会相对较低，而饭前饭后的用户搜索量则会相对较高

6．Bonus—用户平均点击量分析

单次搜索平均点击量	用户数	用户占比
=1	286453	0.5300984312803839
>=3	94691	0.17523136624985888
>=5	28273	0.05232087968214783
>=7	12059	0.02231590167605209
>=10	4720	0.00873464266613864

由表格可以看出，83%的用户在一个查询词结果中的平均点击量<3，即平均而言，大部分用户不会在一个搜索词的结果中较多次点击。这可能说明搜狗引擎得到的结果整体上与用户需求匹配度较高，也可能说明用户会及时更改搜索词以获取更多信息或是得到更贴切的结果。更进一步的分析还需要依赖于对用户搜索词相似度的研究（探究用户更换搜索词的行为）

7．Bonus—URL 查询分析

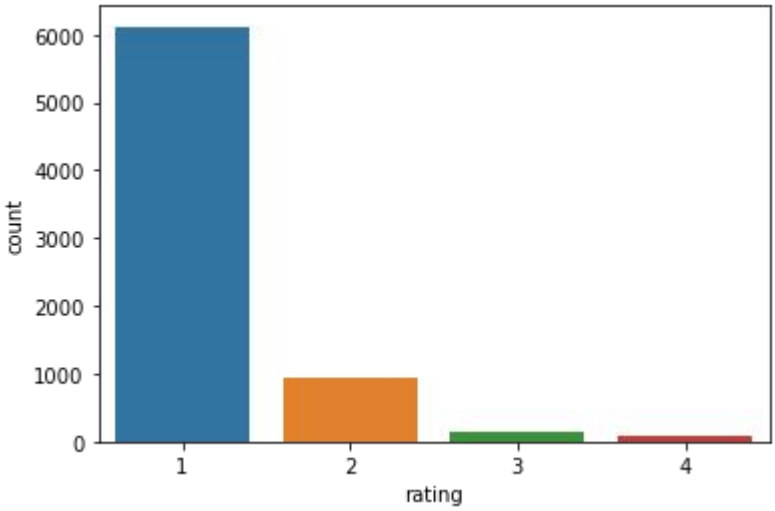
	Unnamed: 0	url	Frequency
0	0	news.21cn.com/social/daqian/2008/05/29/4777194...	17229
1	1	news.21cn.com/zhuanti/domestic/08dizhen/2008/0...	10955
2	2	pic.news.mop.com/gs/2008/0528/12985.shtml	9222
3	3	www.tudou.com/programs/view/2F3E6SGHFLA/	8229
4	4	bjyouth.ynet.com/view.jsp?oid=40472396	6994
5	5	www.17tech.com/news/20080531107270.shtml	5360
6	6	www.baidu.com/	3881
7	7	bbs.cdqss.com/thread-59453-1-1.html	2964
8	8	www.taihainet.com/news/military/jslwt/2007-06-...	2391
9	9	news.vnet.cn/photo/292_6.html	2290

以上是统计出现次数最多的 URL 地址，再对 URL 的频数分布进行统计，得到如下结果：

	Frequency
count	514760.000000
mean	2.037017
std	38.716364
min	1.000000
25%	1.000000
50%	1.000000
75%	1.000000
max	17229.000000

和前面关键词分布所不同的，由于即使搜索同一个关键词，同一个用户也不会反复点击同一个 URL，所以在此处大部分 URL 的出现频率都是 1 次。但是最高点击次数较多，达到接近 20000 次，这里推测可能是较为热门的帖子，例如关于汶川大地震的新闻报道等。

类似于前面对关键词进行分类的标准，在此处对 URL 依旧进行如上的标签分类，在此处高于 10 次的占少数。



程序运行及测试方式

程序运行方式

将每个子任务单独写进一个 python 源文件中，通过保存为文件的形式实现数据的互通（例如一个程序的输出结果作为另一个的输入）

测试方式：

由于数据集规模较大，不方便调试，我们又从官网上下载了规模更小的迷你版日志进行调试，这样可以获得更高的调试效率。待每个程序基本没有漏洞后，再换成最终的大规模数据集。通过迷你数据集上的测试效果以及运行时间，可以合理预测大样本上的程序运行时间，从而判断是否应该采用该种算法进行特征提取。

运行结论：

在此项目中仍采用 pandas 进行数据处理。在调试中我们发现，调用 pandas 包中的内置函数会比直接写循环有更高的运行效率。例如在任务 3 中，如果采用循环对每个用户的搜索词条进行采集，那么对前 10 万条数据处理需要花费 5 分钟，而相同调用 python 的 pandas 内置函数则需要花费约 20s。此外，将 for 循环改写为表达式形式也能很大地提升运行效率。因此在数据处理过程中，应当考虑到 python 的语言特性。

最后，由于 pandas 本身处理数据效率存在一定的上限，在面对百万量级的数据时仍需要一定的处理时间。如果采用 pyspark 分布式处理数据则可以在相同的时间内以更高的效率处理更多的数据。

组内成员分工（具体任务后的名字表示参与该项任务）

任务 1：金佳熠、肖桐

任务 2：金佳熠

任务 3：郭骏宇、金佳熠

任务 4：金佳熠

任务 5：　肖桐

Bonus1（单次搜索点击量分析）：肖桐

Bonus2（URL 点击搜索分析）：郭骏宇

报告撰写：郭骏宇、金佳熠、肖桐

PPT 制作：郭骏宇、金佳熠、肖桐