

景区实时关注度模型

一、模型概述

本模型聚焦于分析 X 市各景区的舆情热度，旨在通过抓取和分析相关舆情信息，筛选出关注度、热度较高的景区，并以直观的可视化形式展示出来，同时提供具体的被搜索、引用次数等详情，为旅游管理部门、景区运营者以及游客等实时掌握景区在公众舆论中的受关注情况、及时调整策略和做出合理决策提供有力的数据支撑。

二、数据来源与整理

搜索引擎数据：借助主流搜索引擎（如百度、搜狗等）提供的公开数据接口或者相关搜索指数工具，获取包含“X 市景区名称”等关键词的搜索记录，这些记录能够反映大众在不同时段对各景区的主动搜索情况，是了解景区关注度的重要基础数据源之一，从中可以提取出各景区被搜索的频次信息。

社交媒体平台数据：诸如微博、微信公众号、抖音、小红书等社交媒体平台上，用户发布的与 X 市景区相关的话题、动态、评论以及点赞、转发等互动信息，包含了大量关于景区游玩体验、特色亮点、活动举办等方面的内容，通过网络爬虫技术（在遵循平台规则和合法合规的前提下）收集这些文本数据，能够从社交传播角度统计各景区被提及、引用的次数，丰富景区关注度的数据来源，了解大众的讨论热度。

旅游专业网站与论坛数据：像马蜂窝、携程旅行网、去哪儿网等专业旅游网站以及各类旅游论坛中，游客分享的游记、攻略、问答等板块都涉及对 X 市景区的描述和评价，从中提取相关信息，可进一步补充完善反映景区受关注程度的数据，比如某个景区在攻略中被频繁推荐，在问答里被多次询问等情况，都体现了其在游客群体中的热度，使数据更具专业性和针对性。

数据整理流程：首先，将从上述多渠道收集到的海量文本数据汇总到统一的数据仓库中。接着，运用自然语言处理技术中的文本清洗工具，去除文本中的标点符

号、停用词（如“的”“是”“在”等对语义分析无实质帮助的常用词）以及重复、无关的信息内容，对文本进行规范化处理。然后，通过词法分析、词性标注等手段，将文本分解为一个个有意义的词语，并依据景区名称词库（事先构建好，包含 X 市所有景区的标准名称及常见别称等）进行筛选，提取出与各景区相关的有效文本信息，为后续的频次统计和热度分析做好准备。

三、核心算法与分析逻辑

景区关注度频次统计：运用数据分析软件（如 Python 的数据分析库 pandas 结合 collections 库中的 Counter 类等），对整理好的与各景区相关的文本数据进行遍历统计，计算每个景区在所有文本数据中被搜索、引用的总次数，形成“景区名称 - 频次”的对应关系数据，以此量化各景区受关注的程度，频次越高意味着该景区在舆情中越受大众关注，热度也就越高。

热度排序与筛选（可选，用于确定展示的景区范围）：为了聚焦呈现关注度较高的景区，可按照频次对所有景区进行降序排序，然后根据实际需求选取排名靠前的若干景区（例如前 10 名或前 20 名等）作为关注度较高的景区集合，后续重点对这些景区进行可视化展示和分析，当然也可以设定一个频次阈值，超过该阈值的景区纳入展示范围，具体方式可根据实际应用场景灵活调整。

四、模型输出与可视化

词云图或气泡图展示（整体热度展示）：

词云图：利用专业的可视化工具（如 Python 的 wordcloud 库或者在线词云生成平台等），根据景区的关注度频次情况，生成以 X 市景区为主题的词云图。在词云图中，将各个景区名称以不同的字体大小、颜色等形式呈现，景区被搜索、引用的次数越多，其在词云图中显示的字体越大、颜色越突出（可根据设计自行设定突出显示的方式），整体形成直观且形象的词云效果，让人一眼就能看出哪些景区是当前舆情关注度较高的热点景区。

气泡图：也可采用气泡图的形式，以景区名称为横坐标，关注度频次为纵坐标，

同时用气泡的大小来表示频次高低，不同颜色的气泡可以区分不同类型的景区（如自然景观类、人文历史类等，若有分类需求），通过气泡的大小和位置直观展示各景区的关注度对比情况，同样便于快速查看热度较高的景区。

交互功能实现：借助前端可视化开发技术（如 JavaScript 结合 Echarts 可视化库等），为可视化图表（词云图或气泡图）添加交互功能。当鼠标指针悬停在某个景区名称上时，在图表上方或旁边弹出信息框，清晰展示该景区的具体名称以及被搜索、引用的次数等详细信息，方便用户进一步了解详情，深入洞察各景区背后所反映的舆情情况。

五、模拟数据示例与可视化展示

假设通过收集和整理数据后，得到以下部分 X 市景区及对应的被搜索、引用次数模拟数据（仅为示例示意）：

景区名称	被搜索、引用次数
X 市 A 景区	500
X 市 B 景区	400
X 市 C 景区	350
X 市 D 景区	300
X 市 E 景区	250

对应的可视化展示如下：



以下是使用 Python 的 wordcloud 库生成满足需求的词云图的示例代码:

```
python

from wordcloud import WordCloud
import matplotlib.pyplot as plt

# 景区名称和对应的权重, 权重越大字体越大、颜色越突出
text = "X 市 A 景区 X 市 B 景区 X 市 C 景区 X 市 D 景区 X 市 E 景区"
weights = [5, 4, 3, 2, 1]

# 创建 WordCloud 对象
wc = WordCloud(background_color="white", width=800, height=400,
max_words=len(text.split(" ")),
relative_scaling=0.5).generate_from_text(text)

# 设置颜色函数, 使颜色依次变淡
def color_func(word, font_size, position, orientation,
random_state=None, **kwargs):
    colors = ['#FF0000', '#FF7F00', '#FFFF00', '#00FF00', '#0000FF']
    index = text.split(" ").index(word)
    return colors[index]

# 重新着色词云
wc.recolor(color_func=color_func)

# 显示词云图
plt.figure(figsize=(10, 6))
plt.imshow(wc, interpolation='bilinear')
plt.axis("off")
plt.show()
```

在上述代码中：

首先定义了景区名称的文本 `text` 和对应的权重 `weights`, 权重的数值大小决定了字体的相对大小。

然后创建 `WordCloud` 对象, 设置了背景颜色、尺寸、最大单词数等参数, 并根据文本生成词云。

接着定义了 `color_func` 函数来根据单词在文本中的位置返回不同的颜色, 从而实现颜色依次变淡的效果。

最后对词云进行重新着色, 并使用 `matplotlib` 显示词云图, 关闭坐标轴。

运行代码后, 将弹出一个窗口显示生成的词云图, 其中 “X 市 A 景区” 字体最大且颜色最突出, 其他景区名称按照要求依次排列。你可以根据实际需求调整颜色、字体等参数。

以下是使用 Python 的 `pandas` 和 `plotly.express` 库来生成气泡图的示例代码, 该代码可以根据上传的数据文件/`mnt/31.xlsx` 生成所需的气泡图:

一、代码实现

python

```
import pandas as pd
import plotly.express as px

def generate_bubble_chart(file_path):
    # 读取 Excel 文件
    df = pd.read_excel(file_path)

    # 生成气泡图
    fig = px.scatter(df, x='景区名称', y='被搜索、引用次数', size='被
    搜索、引用次数', color='景区类型',
                      labels={'被搜索、引用次数': '搜索/引用次数', '景
                      区名称': '景区名称'}
```

```
区类型' : '景区类型'},

            title='景区搜索/引用次数气泡图',
            template='plotly_white')

# 设置图表的布局
fig.update_layout(
    xaxis_title='景区名称',
    yaxis_title='被搜索、引用次数',
    xaxis_tickangle=-45,
    legend=dict(x=1, y=1),
    width=1200,
    height=600
)

# 显示图表
fig.show()
```

```
# 调用函数并传入文件路径
file_path = '/mnt/31.xlsx'
generate_bubble_chart(file_path)
```

二、代码解释

数据读取：

`pd.read_excel(file_path)` 使用 `pandas` 的 `read_excel` 函数读取指定路径的 Excel 文件。假设文件中包含景区名称、被搜索、引用次数和景区类型等列。

气泡图生成：

`px.scatter` 函数用于生成散点图（气泡图）。`x` 参数指定横坐标为景区名称，`y` 参数指定纵坐标为被搜索、引用次数，`size` 参数指定气泡的大小根据被搜索、引用次数来确定，`color` 参数指定根据景区类型来区分气泡的颜色。

`labels` 参数用于设置坐标轴和图例的标签。

`title` 参数设置图表的标题, `template` 参数设置图表的模板为 `plotly_white`(白色背景)。

图表布局设置:

`fig.update_layout` 方法用于更新图表的布局。`xaxis_title` 和 `yaxis_title` 分别设置横坐标和纵坐标的标题, `xaxis_tickangle=-45` 设置横坐标标签的旋转角度为 -45 度, 使其更易于阅读。`legend` 参数设置图例的位置在右上角, `width` 和 `height` 参数分别设置图表的宽度和高度。

显示图表:

`fig.show()` 方法用于显示生成的气泡图。

三、注意事项

数据文件路径:

确保 `file_path` 指向正确的 Excel 文件路径。如果文件路径不同, 需要相应地修改代码中的路径。

数据列名:

代码中假设数据文件中包含景区名称、被搜索、引用次数和景区类型列, 你需要根据实际的数据列名修改代码中的相关部分。

图表定制:

可以根据实际需求进一步定制图表的样式、颜色、标签等, 例如修改图表的颜色映射、添加数据标签等。可以参考 `plotly.express` 的官方文档获取更多的图表定制选项。

这样就可以根据上传的数据生成一个气泡图, 直观地展示景区名称、被搜索 / 引用次数和景区类型之间的关系。

(此处应插入一个简单的词云图示例, 图中 “X 市 A 景区” 字体最大、颜色最突出, “X 市 B 景区” 次之, 以此类推, 各景区名称按照被搜索、引用次数情况在词云图中呈现不同大小和颜色, 鼠标悬停在某个景区名称上有信息框展示景区名称及次数示例图; 或者插入一个气泡图示例, 横坐标为景区名称, 纵坐标

为被搜索、引用次数，气泡大小体现次数多少，不同颜色区分景区类型（可假设分类），鼠标悬停有交互效果示例图）

六、应用场景与价值

景区运营与宣传调整：各景区运营者通过模型实时掌握本景区以及其他景区在舆情中的关注度情况，可针对性地调整运营策略和宣传方案。如果发现自身景区关注度较低，可分析原因，比如是否宣传不到位、产品缺乏特色等，进而加大宣传推广力度、优化景区产品和服务，提升景区的吸引力；对于关注度高的景区，则要继续保持优势，挖掘更多亮点进行宣传，同时做好应对高热度带来的压力（如游客接待量增加等）的准备。

旅游管理决策支持：X 市旅游管理部门依据模型展示的景区实时关注度情况，能够合理调配旅游资源，例如对于热度高的景区，协调交通部门增加运力、通知周边配套服务企业做好充足准备等；还可以根据舆情关注的焦点和趋势，制定旅游发展规划、指导景区建设，推动全市旅游产业的均衡发展，提升整体旅游形象和竞争力。

游客出行参考：游客在计划游览行程时，参考该模型呈现的景区实时关注度信息，能够了解当前哪些景区是热门话题，知晓大众对各景区的大致评价和喜好程度，从而更合理地选择适合自己的景区前往游览，避免盲目跟风或错过一些有特色但关注度稍低的景区，提高旅游体验。