使用Python快速制作一个疫情数据分析应用

原创 Sumner 数析万变 昨天

Ⅰ. 说在前面的话

首先祝贺大家新春快乐。

现在本该是一个阖家团圆的幸福节假日，但是，近期的新型肺炎导致今年春节尤其严肃。在此向奋战在一线的医护工作者以及在全国各地辛苦执勤的各部门工作人员说一声“辛苦了”！同时也提醒大家“少出门，戴口罩，勤洗手”，让我们一起击退这来势汹汹的病毒疫情。

我的老家也在湖北。但是在临回家前几个小时决定不回去了，改在外地一个人封闭式地过年。除了跟亲朋好友们通过网络“云过年”之外，也想做点有意义的事情。看到不少公号都及时地提供了新型肺疫情的实时动态，联想最近用来做机器学习快速应用的Streamlit框架，就想着也做一个疫情跟进的可视化小工具。

请你打开浏览器，输入以下链接：

http://www.saritrend.com/

就可以直接看到相应界面（网站目前处于备案状态，预计两到三天后可以访问）。

应用的全部源代码，我都会上传到 Github 上，访问以下地址即可获取：

https://github.com/xSumner/SARI_Trend

Ⅱ. 数据获取

首先，要获取详细且准确的疫情数据。

公众号大部分由个人或企业运营，因此无法判断其实时性和准确性。一开始，我想从人民日报的微博数据获取，后来发现卫健委上有实时公布的疫情防控动态，这些数据是从各地方的卫健委上报而来，应该可以作为最精准的官方数据。以下为1月25日公布的最新数据：

同时再以人民日报与丁香园公布的数据一起，整理成两份csv格式的数据，一份是国内的的疫情数据，一份是世界的疫情数据。使用Pandas分别读取两份数据：

```
import pandas as pd

# 读取国内疫情数据
df_china = pd.read_csv("china_20200125.csv", encoding='utf-8', sep=",")
# 读取世界疫情数据
df_world = pd.read_csv("world_20200125.csv", encoding='utf-8', sep=",")

# 查看数据前5行
df_china.head()
```

	地区	确诊	死亡	治愈
0	湖北	730	39.0	32.0
1	广东	78	NaN	2.0
2	浙江	62	NaN	1.0
3	重庆	57	NaN	NaN

	地区	确诊	死亡	治愈
4	湖南	43	NaN	NaN

```
df_world.head()
```

	国家	确诊	死亡	治愈
0	China	1377	41.0	38.0
1	Tailand	4	NaN	2.0
2	Japan	2	NaN	1.0
3	Korea	2	NaN	NaN
4	United States	2	NaN	NaN

Ⅲ. 可视化展示

Streamlit支持altair及ploty等可视化框架，但其地图类图表接口在国内较难使用，因此这里直接使用pyecharts制作疫情热力图后再进行页面展示。

ⅰ. pyecharts的安装

直接使用pip安装，然后安装对应的地图拓展包：

```
$ pip install echarts
$ pip install echarts-countries-pypkg
$ pip install echarts-china-provinces-pypkg
$ pip install echarts-china-cities-pypkg
$ pip install echarts-china-counties-pypkg
$ pip install echarts-china-misc-pypkg
```

```
$ pip installecharts-united-kingdom-pypkg
```

这里使用的版本是 pyecharts-1.6.2，比这个高的版本应该都兼容，但注意不要使用低于1.0的版本（两个大版本差异较大）。

Pyecharts中的主要地图来源于charts模块中的 Map，因此先导入以及其他必要的模块：

```
# 使用jupyter lab时要提前声明
from pyecharts.globals import CurrentConfig, NotebookType
CurrentConfig.NOTEBOOK_TYPE = NotebookType.JUPYTER_LAB

from pyecharts.charts import Map, Geo
from pyecharts import options as opts
```

ii. 绘制世界各国疫情地图

```
# 将dataframe中的数据转化为list
zone_n= df_world['国家'].tolist()
value_n= df_world['确诊'].tolist()

# 定义地图可视化函数
defmap_world(nations, values):
    m = (
        Map()
        .add("",[list(z) for z inzip(nations, values)], "world")
        .set_series_opts(label_opts=opts.LabelOpts(is_show=False))
        .set_global_opts(
            title_opts=opts.TitleOpts(title="疫情地图（全球）"),
            visualmap_opts=opts.VisualMapOpts(max_=1500),
        )
    )
    return m

# 实例化地图函数
n = map_world(nations=zone_n, values=value_n)
#n.render(path="世界地图.html")
```

iii. 绘制国内各地疫情地图

```
# 将dataframe中的数据转化为list
zone_c= df_china['地区'].tolist()
value_c= df_china['确诊'].tolist()

# 定义地图可视化函数
defmap_visualmap(nations, values):
    c = (
        Map()
        .add("",[list(z) for z inzip(nations, values)], "china")
        .set_global_opts(
            title_opts=opts.TitleOpts(title="疫情地图（全国）"),
            visualmap_opts=opts.VisualMapOpts(max_=800, is_piecewise=True, pie
                {"min": 100, "color": "Maroon"},
                {"min": 10, "max": 100, "color": "Crimson"},
                {"min": 1, "max": 10, "color": "DarkSalmon"}
            ]),

        )
    )
    return c

# 实例化地图函数
```

```
c = map_visualmap(nations=zone_c, values=value_c)
#c.render(path="中国地图.html")
```


ⅳ. 将可视化结果保存为图片

pyecharts 提供了 selenium, phantomjs和pyppeteer 三种方式将html文件渲染成图片，具体可以参考Pyecharts的官方文档。

这里我偷懒直接在浏览器中打开然后另存为图片，将可视化结果保存为**png格式的图片**。

Ⅳ. 制作Web APP

Streamlit是一个使用纯Python就可以快速搭建网络应用的工具，即使完全没有web开发基础也可以直接上手。项目迭代速度较快，目前更新到Version 0.53，其使用方法可见其接口文档。

这里我只简单介绍代码的逻辑，以及使用到的控件作用。

ⅰ. 安装Streamlit

在线安装可以直接使用

```
pip install streamlit
```

进行安装，由于其依赖的包较多，离线安装较为麻烦，建议先在线安装成功后再将依赖包全部导入安装。

ii. 代码解析

首先引入streamlit

```
import streamlit as st
```

然后加入侧边栏，在侧边栏中我们可以通过下拉框从三个选项中进行选择：

1. 查看疫情地图；
2. 寻找全国发热门诊联系方式；
3. 获取重要咨询以及辨别网络谣言。

Streamlit中的侧边栏控件为sidebar，下拉选择框为selectbox，代码如下：

```
option= st.sidebar.selectbox("请选择想要查看的内容：", ("疫情地图", "全国发热门诊联系
```

侧边栏中的下拉框的效果如图：

从代码中可以看出，下拉框的选择结果会传递给option这个变量，通过判断option的值就可以设置页面主要区域内容的显示逻辑。

当选择疫情地图的时候直接显示绘制好的疫情地图。

Streamlit中显示图片需要调用PIL库，保证已经安装 6.2 及以上版本，使用pip安装的方法为：

```
pip install pillow
```

然后使用以下代码在页面中显示疫情地图：

```
from PIL import Image

if option =="疫情地图":
    image_c = Image.open('china_20200125.png')
    image_w = Image.open('world_20200125.png')
    st.image(image_c, use_column_width=True)
    st.markdown('''---''')    # 添加一条分割线
    st.image(image_w, use_column_width=True)
```

至此得到一个Web APP的大致雏形：

由于疫情每天都是动态变化的，所以引入时间选择，根据选择的日期来查看当天的疫情地图。

疫情地图只能看到大致的分布，想要看到各个地区具体的数字还可以将dataframe直接显示出来，代码如下：

```
importdatetime
if option =="疫情地图":
    d = st.date_input("请选择想要查看的日期：", datetime.date(2020, 1, 25))
    d = d.strftime('%Y%m%d')
    try:
        image_c = Image.open('china_%s.png'%d)
        image_w = Image.open('world_%s.png'%d)
        # 读取国内疫情数据
        df_china = pd.read_csv("china_%s.csv"%d, encoding='utf-8', sep=",",inc
        # 读取世界疫情数据
        df_world = pd.read_csv("world_%s.csv"%d, encoding='utf-8', sep=",",inc
        # 显示疫情地图以及疫情数据
        st.image(image_c, use_column_width=True)
        st.table(df_china.style.highlight_max(axis=0))
        st.markdown('''---''')
        st.image(image_w, use_column_width=True)
        st.table(df_world.style.highlight_max(axis=0))
    except:
        st.write("抱歉，没有当天的数据！")
```

完成“疫情地图”部分就可以进行“全国发热门诊联系方式”相关内容的展示。

全国各个地市有多家发热门诊，使用嵌套字典的数据结构可以进行较为方便的查询。这部分数据在网络上都是非机构化的数据，而且有些地市放出来的只是Excel截图，因此我在后续会逐步转化为结构化数据并放在Github项目文档中。

在界面添加两个下拉框进行选择检索，注意下拉框中选项为元组结构：

```
if option =="全国发热门诊联系方式":
    province = st.selectbox("请选择省份：", province_tuple)
    city = st.selectbox("请选择城市", city_tuple)
    county = st.selectbox("请选择区县", county_tuple)
    st.write(hospitals[province][city][county])
```

这样就会将选择区域内的发热门诊及联系方式显示出来，由于数据是以JSON的形式存储在pickle中，所以显示的也是JSON结构。当然如果想要对数据的显示进一步处理为其他

样式也可以。

效果显示如下图：

在最后一个栏目“资讯与辟谣”中，我想要发布一些文字为主的内容，Streamlit里面的markdown模块可以很好的支持文本类数据。

授人以鱼不如授人以渔，这里我就以数据分析的思路对一些谣言进行谣言进行特征提取，方便大家在官方辟谣之前对谣言进行一定的鉴别。当然如果积累的样本多了，还可以使用机器学习的方法对谣言进行自动识别。

先看一下丁香园提供的“谣言排行榜”：

NO.1	熏醋可以杀死新型冠状病毒？
NO.2	抗流感药物可以预防新型冠状病毒
NO.3	有疫情的地方才需要戴口罩
NO.4	出门需要佩戴护目镜
NO.5	开空调暖气能杀死病毒

NO.1	熏醋可以杀死新型冠状病毒？
NO.6	益生菌、乳铁蛋白可以增加抵抗力预防病毒
NO.7	奥司他韦、板蓝根等可以用来预防新型肺炎
NO.8	病毒只传给大人不传染给孩子，婴幼儿很安全
NO.9	开空调暖气能杀死病毒
NO.10	盐水漱口能预防冠状病毒
NO.11	饮用高度酒能够抵抗冠状病毒
NO.12	只要不吃野味、海鲜，就不会被感染
NO.13	服用 VC 可以预防新型冠状病毒
NO.14	病毒感染的都是老年人，年轻人没事

与此同时腾讯较真上实时辟谣在1月26号主要是有关“封城”、“采访”等民众比较关心的实时问题，可见谣言也是与时俱进的，时刻把握住了大家关注的焦点。

因为没有来得及逐条分析所有的谣言，我这边只是凭经验描述一下谣言的一些特点：

1. 除了官媒之外，民众获取信息最快的渠道是“微博”、“微信公众号”以及“微信群聊”，这也是谣言最主要的扩散方式；
2. 散布谣言者的主要动机是“引流”，“爆假料”和“煽情绪”**是最好的引流方式**；
3. 引流成功之后，流量变现的方式目前主要是“吸粉”、“广告位升值”、“带货”以及“打赏”等。

可通过五个维度比较谣言和非谣言的一些差异：

类别	谣言	非谣言
来源	财经、生活、情感、时政等微博帐号或微信公众号	专业医生、医疗机构和政府机关
载体	文字截图、朋友圈或微博截图、群聊截图或短视频	正常排版的文字
标题	擅用惊叹号以及惊悚的文字以吸引眼球	专业中性的词汇
内容	内容有明显的情感倾向	以数据描述为主，较少评论
帐号特征	以前较少相关内容	与历史发布的内容有时间延续性

这里暂时先不做量化指标，只是将以上文字以markdown形式发布在页面上，相应代码为：

```
if option =="资讯及辟谣":
    st.markdown(rumor_content)
```

iii. 运行APP

如果想要运行我们做好的WEB APP，在配置好环境后直接运行主程序即可，运行命令为：

```
streamlit run SARI.py
```

如果是在本地运行，那么运行成功后会如下图显示：

然后通过http://localhost:8501/ 访问即可，streamlit默认的端口是8501：

如果是通过远程云服务器运行，那么运行后会显示对应的访问URL：

注意这里我将访问端口从默认的8501改为了80，具体修改方式可以查询streamlit官方文档，这里就不赘述了。

Ⅴ. 总结

在这篇文章中我描述了一个简单数据分析的WEB APP制作流程，其中包括数据的采集、可视化分析以及使用streamlit快速搭建网络服务。本文略过了数据清洗部分，也没有引入复杂的机器学习算法，但基本覆盖了主要流程，且代码100%使用Python, 简单好上手。

目前网上已经有不少跟进疫情的服务，腾讯官方也上线了辟谣和最新资讯功能，具体可搜索“腾讯较真”。这里并不是推荐大家重复造轮子。只是今年春节大家在被SARI“封锁”在家中的时候，可以学习一下已经非常流行的pyecharts和即将变为流行的streamlit。

最后还是提醒大家注意防护，也请勿听信谣言，**相信国家和政府能够带领大家共渡难关**。

以上所有的代码和数据都会逐步上传到文章开头提到的项目地址中，如有兴趣请及时更新。