**数据挖爬取分析实战：利用python强大库**

1. **背景与挖掘目标**

当疫情发生后，及时掌握扩散的范围和带病人数，是制定防疫方案的重要前提。如果对发病人数和范围不及时掌握，那直接的后果就是病毒扩散扩散再扩散。今天的局面，我想其中一个主要原因就是年初对疫情的情况没有一个真实而直观的认识，导致对疫情发展的有所轻视而造成的。所以在疫情实时监控的方案中，除了数据的真实和实时性外，可视化也是一个非常重要的要素。而我们今天看到，各卫建委每天的疫情发布中，还是采用文字描述的形式，可见信息化水平有多低。

疫情防控不可松懈，利用数据挖掘做好重点区域人群密度预测，本实战案例基于词云图可视化的数据进行建模，对不同省份的人群密度进行预测，其目标如下：

（1）借助重点区域历史的人群密度，统计人流量指数和迁徙指数特征；

（2）建立模型预测重点区域未来的人群密度，掌握人员流动聚集的动向；

（3）针对人群密度较大的区域，做好紧急疫情防控工作。

**二、文章目录**

疫情数据爬取与可视化展示：

1. 疫情数据抓取

2. 初步分析

3. 数据处理

3.1 国内各省疫情数据提取

3.2 国际疫情数据提取

3.3 数据整合

4. 可视化展示

4.1 国内疫情态势可视化

4.2 国际疫情态势可视化

4.3 国内疫情方寸间

4.4 国际疫情方寸间

5.词云图

**三、分析方法与过程**

导入相关的库：

import time

import json

import requests

from datetime import datetime

import pandas as pd

import numpy as np

1. 疫情数据抓取

通过腾讯新闻公布的数据进行爬取

网址:https://news.qq.com/zt2020/page/feiyan.htm#/

分别对该网站爬取国内信息跟国外信息定义成函数以便后面，所以疫情信息都在这里，可以更直观的观看全球疫情

代码清单数据爬取：

def Domestic():

url = 'https://view.inews.qq.com/g2/getOnsInfo?name=disease\_h5'

reponse = requests.get(url=url).json()

data = json.loads(reponse['data'])

return data

def Oversea():

url = 'https://view.inews.qq.com/g2/getOnsInfo?name=disease\_foreign'

reponse = requests.get(url=url).json()

data = json.loads(reponse['data'])

return data

domestic = Domestic()

oversea = Oversea()

运行结果可以获得国内外信息的各个指标，以便于后面信息分析的提取与处理

2. 初步分析

提取各地区数据明细

# 提取各地区数据明细

areaTree = domestic['areaTree']

# 查看并分析具体数据

areaTree

提取国外地区数据明细

# 提取国外地区数据明细

foreignList = oversea['foreignList']

# 查看并分析具体数据

foreignList

就可以看到在json数据存储的结构了

3. 数据处理

3.1 国内各省疫情数据提取

通过对爬取到的html文件进行筛选，获取到省份信息，累计确诊，累计治愈，累计死亡，现有确诊的信息。然后对其建立一个数据框架可视化查看。

代码清单数据提取：

#国内数据

china\_data = areaTree[0]['children'] #列表里面嵌套字典，去除外面最后哦一层列表，提取国内的信息

china\_list = []

for a in range(len(china\_data)): #遍历国内的信息,下面分别提取省份信息，累计确诊，累计治愈，累计死亡，现有确诊

province = china\_data[a]['name']

confirm = china\_data[a]['total']['confirm']

heal = china\_data[a]['total']['heal']

dead = china\_data[a]['total']['dead']

nowConfirm = confirm - heal - dead

china\_dict = {}

china\_dict['province'] = province

china\_dict['nowConfirm'] = nowConfirm

china\_list.append(china\_dict)

china\_data = pd.DataFrame(china\_list) #将该省份信息，现有确诊写入成为一个数据框的形式显示

china\_data.to\_excel("国内疫情.xlsx", index=False) #存储为EXCEL文件

china\_data.head()

获取得到的结果为：



3.2 国际疫情数据提取

有了以上爬取国内信息代码的基础上，同理爬取国际筛选获取各种信息，并将其可视化成为列表

代码清单数据提取：

#国际数据

world\_data = foreignList

world\_list = []

for a in range(len(world\_data)): #遍历国内的信息,下面分别提取省份信息，累计确诊，累计治愈，累计死亡，现有确诊

# 提取数据

country = world\_data[a]['name']

nowConfirm = world\_data[a]['nowConfirm']

confirm = world\_data[a]['confirm']

dead = world\_data[a]['dead']

heal = world\_data[a]['heal']

# 存放数据

world\_dict = {}

world\_dict['country'] = country

world\_dict['nowConfirm'] = nowConfirm

world\_dict['confirm'] = confirm

world\_dict['dead'] = dead

world\_dict['heal'] = heal

world\_list.append(world\_dict)

world\_data = pd.DataFrame(world\_list)

world\_data.to\_excel("国外疫情.xlsx", index=False)

world\_data.head()

获取得到的结果为：

