**Python爬取疫情数据及分析报告**

**摘要**

2020年伊始，一场突如其来的新型冠状病毒引起的肺炎疫情肆虐，全球经济遭遇重创。美国资本市场十天内四度熔断，世界更是风声鹤唳。中国全民抗疫，采取了“封城”等措施，经济同样受到很大影响。如今，2020年已到最后一个月，中国疫情虽然取得阶段性胜利，但是全球疫情增长却愈演愈烈，而作为世界上最大的发达国家-美国的疫情数据则更是令人担忧。因此，我将结合实际对美国疫情数据进行分析和预测。本次使用python的request库从腾讯新闻上爬取了美国的疫情数据，保存为csv文件。并且借助matplotlib、numpy等第三方库对数据进行清洗以及可视化处理。最后，借助机器学习的回归模型对未来的疫情数据进行了一个简单的预测。

目录

**摘要**

1. **系统设计背景**
2. **疫情数据分析的重要性**
3. **系统设计目的**
4. **系统思想**
5. **爬取美国疫情数据**
6. **数据的处理与可视化**
7. **数据获取流程**
8. **寻找美国疫情数据**
9. **处理获取有用信息**
10. **保存csv文件**
11. **代码结构**
12. **可视化展示及分析**
13. **确诊人数折线图**
14. **死亡率和治愈率的折线图**
15. **确诊人数增长率与死亡率的散点图**
16. **预测死亡人数图（线性回归）**
17. **死亡人数预测图（KNN算法）**
18. **结论**
19. **感想与收获**

**一、系统设计背景**

**1.疫情数据分析的重要性**

这次新冠肺炎疫情，是新中国成立以来在我国发生的传播速度最快、感染范围最广、防控难度最大的一次重大突发公共卫生事件。实践证明，国家对于疫情形势的判断是准确的，各项工作部署是及时的。而这些离不开对疫情数据的分析和预测。如果没有专家对疫情数据进行解读，那么领导们又如何能下定决心去下达指示和政策呢？

1. **系统设计目的**

通过分析美国的疫情数据，来提醒大家全球疫情尚未结束，甚至可以说是愈演愈烈。我国的疫情虽然得到了有效控制，但是对沾染病毒的进口物品和境外输入病例的防控依然是压力巨大，并且时不时出现一些局部地区的疫情。因此，我通过python设计了一个爬取美国疫情数据的爬虫，并且对一些数据进行整理、可视化，然后借助机器学习的方法进行预测，同时结合实际进行合理的解读。

1. **系统思想**
2. **爬取美国疫情数据**

首先要处理疫情数据，我们必须要先获取数据，这里我选择用python对腾讯新闻上的美国疫情数据进行爬取。在得到疫情期间每一天的美国确诊人数、新增确诊病例数、治愈人数、死亡人数后，我们把它们保存为csv格式文件，同时计算了死亡率和治愈率，方便后面的分析。

1. **数据的处理和可视化**

得到疫情数据后，我们一边进行着数据的处理，一边进行着数据的可视化。这里我们首先制作了美国3月底到11月底总的确诊人数数据的折线图，并且得到了一张新的csv文件，里面包含每个月的疫情数据，以便于我们进行其他操作。

1. **数据获取流程**
2. 寻找美国疫情数据

首先，我们通过百度得知腾讯新闻上有国内和海外国家的疫情数据，网址是<https://news.qq.com/zt2020/page/feiyan.htm?isappinstalled=0#/global>

这里，我们可以查看海外国家的疫情数据。点击详情，我们就可以查看美国的疫情数据。



1. 处理获取有用信息

根据信息提示，我们找到了一个json文件，它包含了美国每天的疫情数据（确诊人数、新增病例数、治愈人数、死亡人数）。通过request.get()方法，请求网页，通过json.loads()方法获取这些数据，同时计算死亡率、治愈率，最后将这些疫情数据写入 美国疫情数据.csv文件中。网址：<https://api.inews.qq.com/newsqa/v1/automation/foreign/daily/list?country美国&>。



1. 保存csv文件

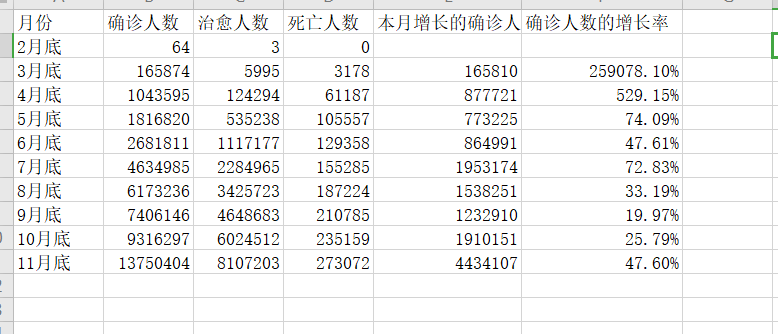
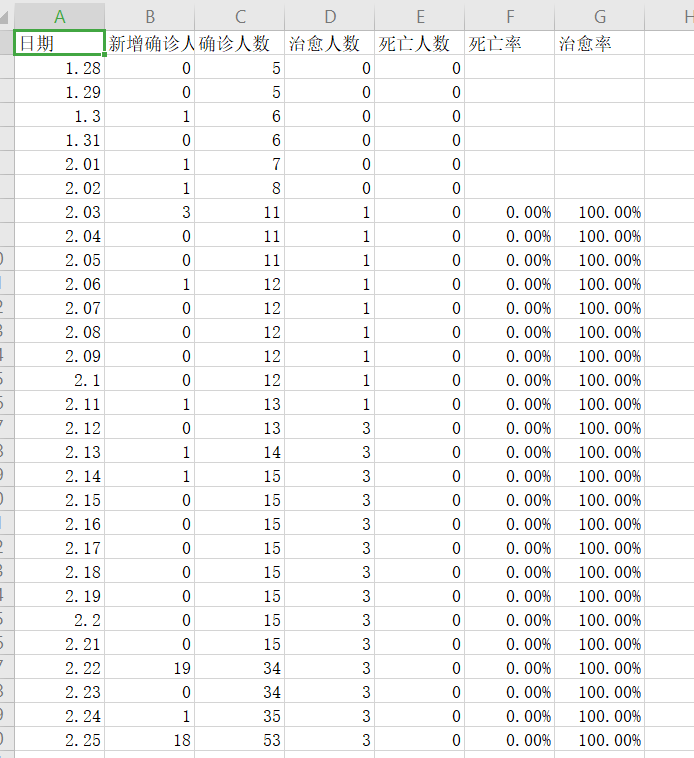
with open('美国疫情数据.csv','w',*newline*='') as fs:

        writer = csv.writer(fs)

        for row in alldata:

            writer.writerow(row)

通过import csv，我们可以轻松将包含疫情数据的列表写入csv文件。为了方便我们进一步解读疫情数据，我又制作了一个新的csv文件，它包含美国每个月底的疫情数据和一些新添加的统计结果。



1. **代码结构**

代码：

1.爬取美国疫情数据.py：爬取美国每天的疫情数据，同时将疫情数据保存到 美国疫情数据.csv文件。

*def* getHTMLText(*url*): 请求网页函数

*def* fillUnivList(*html*):处理爬取的疫情数据，写入文件

2.Process\_data.py: 对美国疫情数据进行加工处理，将处理后的疫情数据写入 处理后的疫情数据.csv文件，最后绘制了一张确诊人数的折线图。

with open("美国疫情数据.csv","r") as fIn:

    reader = csv.reader(fIn)*#创建阅读器*

    data = *list*(reader)*#存储csv数据*

    date = [row[0] for row in data]*#日期*

    confirm\_add = [row[1] for row in data]*#新增病例*

    confirm = [row[2] for row in data]*#确诊人数*

    heal = [row[3] for row in data]*#治愈人数*

    dead = [row[4] for row in data]*#死亡人数*

（1）读取美国疫情数据.csv

*#第一步，我们开始制作一个新的.csv文件，统计每个月的疫情数据[3月-11月]*

with open("处理后的疫情数据.csv","w",*newline*='') as fOut:

    myconfirm = []*#每个月底的确诊人数*

    myheal = []*#治愈人数*

    mydead = []*#死亡人数*

    Rate\_Increase = []*#增长率*

（2）统计每个月底的疫情数据，同时写入新文件

*#第二步，制作关于美国2月底到11月底的确诊人数的折线图*

（3）制作关于美国2月底到11月底的确诊人数的折线图

3.Analysis\_data.py：继续进行新的可视化，解读疫情数据，同时用多变量线性回归模型、KNN算法、决策树算法来预测疫情死亡人数。

(1)选择部分疫情数据，为后面的机器学习提供训练集和数据集

*#选择部分疫情数据，用作机器学习的训练集和测试集*

with open("美国疫情数据.csv","r") as fIn:

    reader = csv.reader(fIn)*#创建阅读器*

    data = *list*(reader)*#存储csv数据*

    date = [row[0] for row in data[40:]]*#日期*

    confirm\_add = [*int*(row[1]) for row in data[40:]]*#新增病例*

    confirm = [*int*(row[2]) for row in data[40:]]*#确诊人数*

    heal = [*int*(row[3]) for row in data[40:]]*#治愈人数*

    dead = [*int*(row[4]) for row in data[40:]]*#死亡人数*

    Rate\_Dead = [*float*(row[5].strip('%'))/100 for row in data[40:]]*#死亡率*

    Rate\_Heal = [*float*(row[6].strip('%'))/100  for row in data[40:]]*#治愈率*

（2）机器学习三种经典的回归算法

*#f = linear\_model.LinearRegression() #线性回归*

*#f = tree.DecisionTreeRegressor()#决策树回归*

f = neighbors.KNeighborsRegressor()*#KNN回归*

（3）用多种机器学习的回归模型预测疫情的死亡人数

*def* draw(*self*):*#绘制死亡人数预测图*

（4）绘制死亡率和治愈率折线图

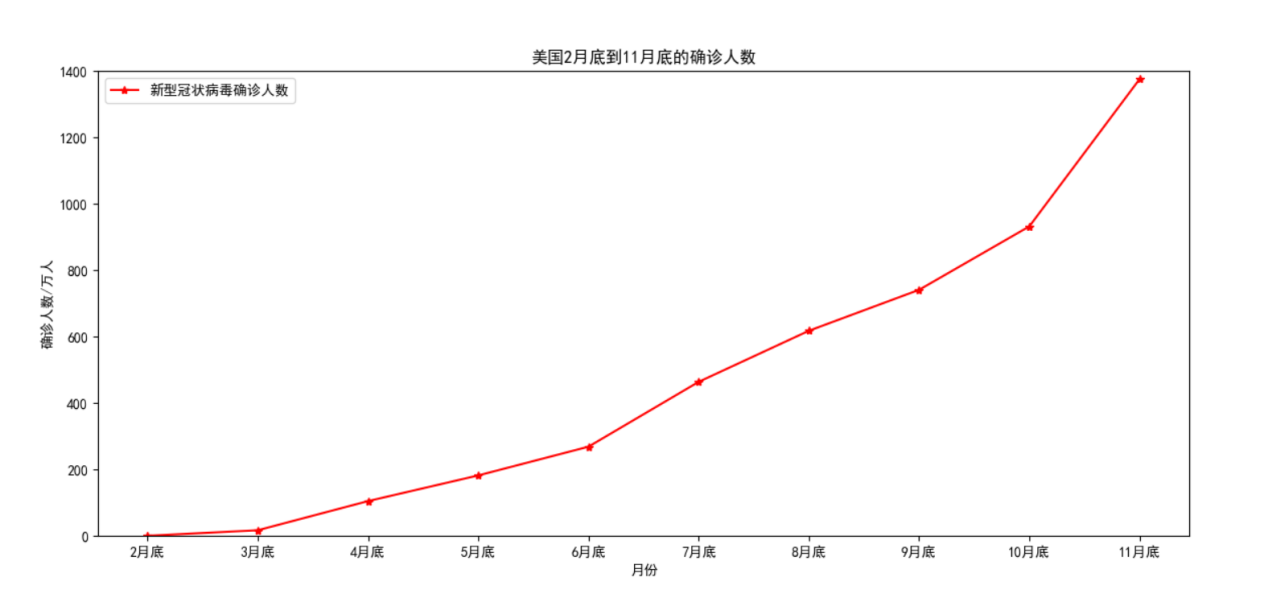
*def* draw\_2():*#绘制死亡率和治愈率折线图.png*

（5）绘制确诊人数增长率与死亡率的散点图

*def* draw\_3():*#绘制确诊人数增长率与死亡率的散点图.png*

1. **可视化展示及分析**
2. 关于美国2月底到11月底的确诊人数的折线图

采用matplotlib中的plt.plot()方法绘制确诊人数的变化曲线。



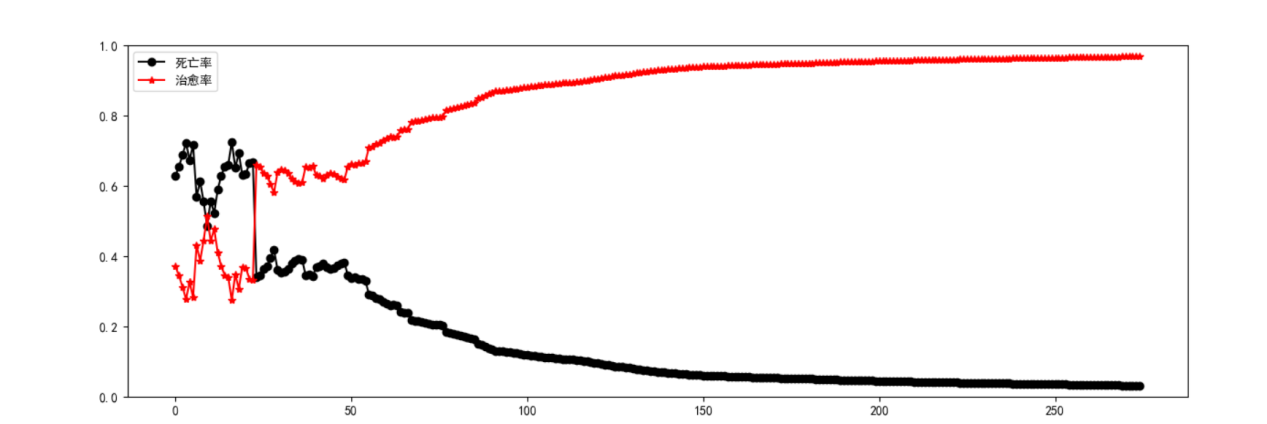
通过观察疫情数据，我们可以看出美国11月份的确诊人数明显加快。根据下面这张表，我们可以发现虽然11月的确诊人数的增长率并不是最高的。但是，美国11月时确诊病例已超1000w，基数太大，47.6%的增长率着实惊人。悲观的说，美国的疫情看起来已经失控。如果没有大规模的疫苗接种，疫情得到真正控制还有很漫长的时间。



1. 关于美国最近200多天的死亡率的折线图

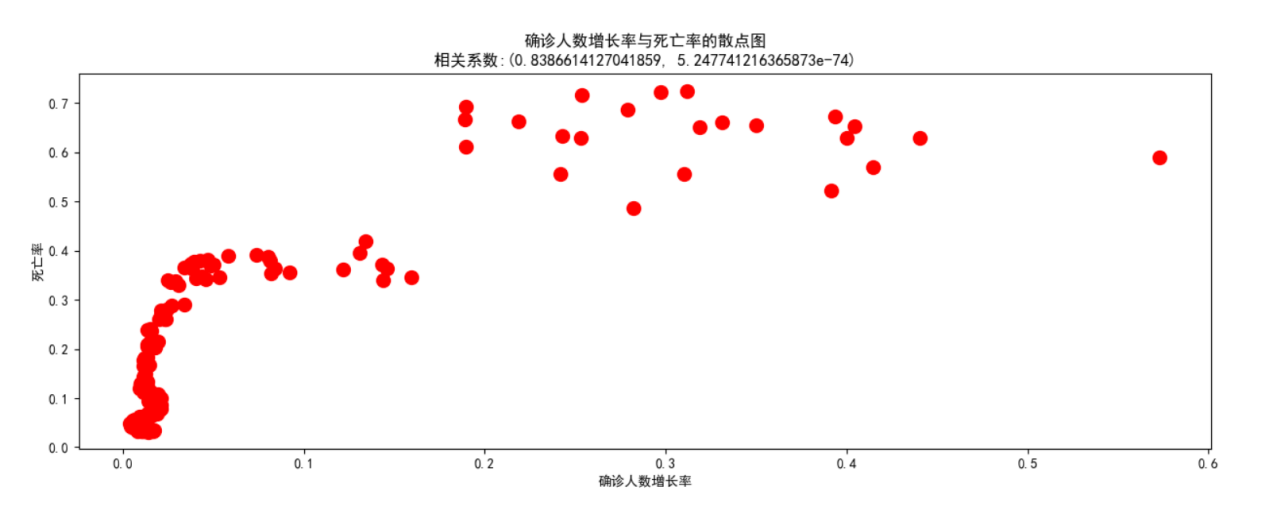
采用matplotlib中的plt.plot()方法绘制确诊人数的变化曲线。

通过观察下面这张图，我们可以看出在疫情初期，由于政府和医院准备不足、治疗新冠的经验不足、大量感染者没有被查出来等多个原因，病毒造成的死亡率居高不下。但是，美国的医疗系统也确实强大，在疫情疯狂蔓延的情况下，病毒造成的死亡率也最终得到抑制，并没有出现因为医疗击穿而造成大规模死亡的情况，而治愈率在低谷中不断提升，这也证明了新冠病毒的致死率没有像埃博拉、SARS病毒那么高，但是它的传播速度、耐热性、耐寒性、适应环境的能力都是特别强的，否则也不会出现如今这种在全球肆虐的情况。

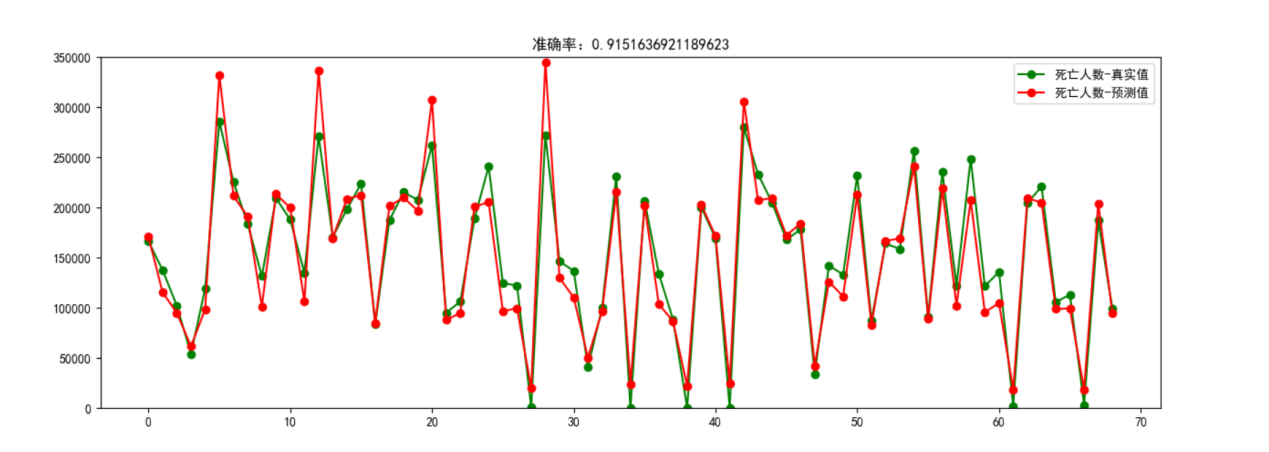


3 .关于美国新冠确诊人数增长率与死亡率的散点图

根据散点图和两者的相关系数，我们可以得出结论，它们呈正相关，虽然这个相关性不是特别强烈。但是，我们从实际出现，确诊人数增长越快，政府和医院的压力就会变大，各方面的准备就会显得不足，对感染者的追踪和治疗就会显得力不从心，这一定程度上导致死亡率的增加。



4 .利用机器学习的多变量线性回归模型来预测美国疫情死亡人数。



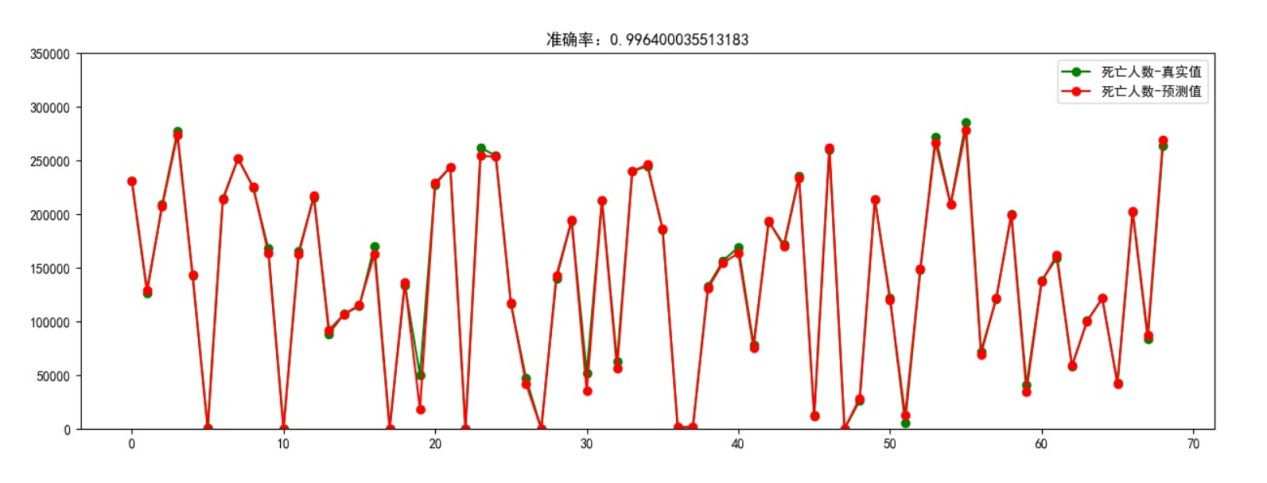
通过调用sklearn里的LinearRegression()来实现多变量线性回归。最后，实现对疫情死亡人数的预测，这里我们的训练集变量是（确诊人数、新增确诊人数、治愈人数）。经过一定次数的训练，我们的程序已经可以预测疫情死亡人数。

但是，我可以看到准确度其实并不是特别高（图中红点与绿点差距还是挺大）。

5.利用机器学习的KNN算法(或者用决策树算法)来预测美国疫情死亡人数。

为了解决上面的问题，我选择了sklearn里的KNN算法（K最近邻算法），

这次我们发现预测程序的准确率已经相当高了。然后我就试了下决策树算法，发现预测效果也很理想。



1. **结论**
2. 首先根据爬取的美国疫情数据，我们可以看出美国疫情最近又加速蔓延，这些跟美国最近的大选活动、抗议、聚集、政府抗疫松懈、在冬天病毒更加活跃等都有一定的关系。同时，这也提醒我们要坚持防控疫情，不能松懈，防止在国内出现疫情的第二次大流行。
3. 通过绘制这几百天的美国疫情的死亡率和治愈率的折线图，我们可以看出新冠病毒只要治疗及时，是有很大机率治好的，所以我们在科学防控疫情的同时，不需要太害怕，也不用过于紧张和悲观。
4. 通过绘制美国新冠确诊人数增长率与死亡率的散点图，我们发现两者有一定的相关性，这警示我们如果不认真防控疫情，导致疫情增长过快，政府和医院的压力就会增长，病毒造成的死亡率进而也会增加。
5. 最后，我借助机器学习的算法预测了美国疫情造成的死亡人数，并且通过一定数量的测试证明了它的预测效果很好。这提示我们这次蔓延全球的病毒虽然致死率没有那么高，但是它的传播速度、抗热性、抗寒性等特征都是极其强悍的，所以它造成的死亡人数最终将远远超过于我们之前见过的病毒。
6. **感想与收获**

通过这次的python大作业，我巩固了很多关于爬虫、可视化、机器学习等python课上提过的知识，并且对利用python爬取信息->可视化->案例分析这一过程有了实质性的认识。

在完成这次作业的过程中，我不仅对numpy、requests、matplotlib这些第三方库的用法有了更深的理解，还对折线图、散点图等一些可视化方法有了更清晰的认识，同时第一次尝试用机器学习课上的一些算法去完成预测数据的功能，这为我以后利用机器学习的算法去解决实际问题奠定了坚实的基础。

最后，我非常感谢自己能选上Python这门课，身为一名计算机专业的大学生，我认为Python应该是一门计算机专业的必修课，它不仅是一种简单、功能强大、编写代码效率很高的编程语言，还是我们每个人用来处理数据的有力工具。我以后一定会进一步学习Python。

道阻且难，行则将至，与君共勉。