“沐风杯”第三届数据科学寒假线上研习活动

研 习 报 告

题　　目	基于 Sklearn 线性回归模型的新型冠状病毒疫情分析
学　　校	合肥学院
院　　系	人工智能与大数据学院
专业班级	17 信息与计算机科学
姓　　名	张博
邮　　箱	1013764208@qq.com
联系电话	18297984520

2020 年 1 月

数统协会制

基于 sklearn 线性回归模型的新型冠状病毒疫情分析

张博

（合肥学院　安徽 合肥 230601）

摘要：本文对 2019 年新型冠状病毒（2019-nCoV）疫情进行数据分析。基于 scikitlearn 线性回归模型，拟合了 1 月 14 至 2 月 20 日，全国新型冠状病毒的累计确诊人数，对 2 月 21 至 2 月 29 日疫情进行预测。

关键字：新型冠状病毒；sklearn 线性回归；数据分析

Analysis of New Coronavirus Epidemic Situation Based on Sklearn Linear Regression Model

Zhang bo

(Hefei University　Anhui Hefei 230601)

Abstract: This article analyzes the data of the new coronavirus (2019-nCoV) outbreak in 2019. Based on the scikitlearn linear regression model, from January 14 to February 20, the cumulative number of confirmed new coronaviruses nationwide was fitted, and the epidemic situation from February 21 to February 29 was predicted.

Key words: new coronavirus; sklearn linear regression; data analysis

1 引言

2020 年武汉爆发新型冠状病毒，由于正值春运，病毒传播迅速，感染人数与日俱增。2020 年 1 月 23 日，武汉“封城”，随即世界卫生组织宣布，疫情构成全球突发公共卫生事件。据国家卫健委网站消息指，2 月 4 日 24 时，31 个省(自治区、直辖市)和新疆生产建设兵团报告新增确诊病例 3887 例(湖北省 3156 例)，新增重症病例 431 例(湖北省 377 例)，新增死亡病例 65 例(湖北省 65 例)，新增治愈出院病例 262 例(湖北省 125

例)，新增疑似病例 3971 例(湖北省 1957 例)。此外，现有疑似病例 23260 例。而目前累计追踪到密切接触者 252154 人，当日解除医学观察 18457 人，现有 185555 人正在接受医学观察。本文基于 sklearn 线性回归模型的基础上，对此后 10 日的疫情趋势予以预测。

2 相关方法概述

2.1 线性回归概述

数理统计中回归分析，用来确定两种或两种以上变量间相互依赖的定量关系的一种统计分析方法，其表达形式为$y = wx' + e$，其中只有一个自变量的情况称为简单回归，多个自变量的情况叫多元回归。

简单线性回归：只有一个特征值，预测值公式如下：

$$\hat{y}^{(i)} = ax^{(i)} + b$$

多元线性回归：具有 n 个特征值，预测值公式如下：

$$\hat{y}^{(i)} = \theta_0 + \theta_1 x_1^{(i)} + \theta_2 x_2^{(i)} + ... + \theta_n x_n^{(i)} = x_b \cdot \theta$$

$$x_b = \begin{vmatrix} 1 & x_1^{(1)} & x_2^{(1)} & \cdot & x_n^{(1)} \\ 1 & x_1^{(2)} & x_2^{(2)} & \cdot & x_n^{(2)} \\ \vdots & \vdots & \vdots & \cdot & \vdots \\ 1 & x_1^{(m)} & x_2^{(m)} & \cdot & x_n^{(m)} \end{vmatrix} \qquad \theta = \begin{pmatrix} \theta_0 \\ \theta_1 \\ \cdot \\ \theta_n \end{pmatrix}$$

多元线性回归方程演变成求θ，使得下列目标值(预测值 y 的误差平方和)最小：

$$\sum_{i=1}^{m} \left(y^{(i)} - \hat{y}^{(i)}\right)^2 = \sum_{i=1}^{m} \left(y^{(i)} - x_b \cdot \theta\right)^2$$

Sklearn-LinearRegression 中的 fit 方法就是通过训练集求取θ

2.2.1 线性回归分析的内容

回归分析是指通过提供变量之间的数学表达式来定量描述变量间相关关系的数学过程，这一数学表达式通常称为经验公式。我们不仅可以利用概率统计知识，对这个经验公式的有效性进行判定，同时还可以利用这个经验公式，根据自变量的取值预测因变量的取值。如果是多个因素作为自变量的时候，还可以通过因素分析，找出哪些自变量

对因变量的影响是显著的，哪些是不显著的。

线性回归假设因变量与自变量之间为线性关系，用一定的线性回归模型来拟合因变量和自变量的数据，并通过确定模型参数来得到回归方程。根据自变量的多少，线性回归可有不同的划分。当自变量只有一个时，称为一元线性回归，当自变量有多个时，称为多元线性回归。

2.2.2 回归分析一般步骤

第 1 步：确定回归方程中的因变量和自变量。

第 2 步：确定回归模型。

第 3 步：建立回归方程。

第 4 步：对回归方程进行各种检验。主要有拟合优度检验；回归方程的显著性检验；回归系数的显著性检验。

第 5 步：利用回归方程进行预测

2.2.3 线性回归分析的模型推导

设定线性模型：

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_i x_i$$

$$h_\theta(x) = \sum_{i=0}^{n} \theta_i x_i = \theta^T x$$

将训练数据代入上式设定模型中，可以通过模型预测得到最终值样本：

$$y^{(i)} = \theta^T x^{(i)} + \varepsilon^{(i)}$$

样本真实值y和模型训练预测的值之间是有误差ε，当训练样本的数据量很大时，可根据中心极限定律得到$\sum \varepsilon$满足(u, δ^2)高斯分布；由于方程有截距项，故使用可以$u = 0$；故满足$(0, \delta^2)$的高斯分布；

$$y^{(i)} = \theta^T x^{(i)} + \varepsilon^{(i)}$$

$$p(\epsilon^{(i)}) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(\epsilon^i)^2}{2\sigma^2}\right)$$

$$p(y^{(i)}|x^{(i)};\theta) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{(y^{(i)} - \theta^T x^{(i)})^2}{2\sigma^2}\right)$$

如上式可知，对于每一个样本 x，代入到$p(y|\ x;\theta)$都会得到一个 y 的概率；又因为设定样本是独立同分布的；对其求最大似然函数：

$$\begin{aligned} L(\theta) &= \prod_{i=1}^{m} p(y^{(i)}|x^{(i)};\theta) \\ &= \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{(y^{(i)}-\theta^T x^{(i)})^2}{2\sigma^2}\right) \end{aligned}$$

对其简化：

$$\begin{aligned} l(\theta) &= \log L(\theta) \\ &= \log \prod_{i=1}^{m} \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{(y^{(i)}-\theta^T x^{(i)})^2}{2\sigma^2}\right) \\ &= \sum_{i=1}^{m} \log \frac{1}{\sqrt{2\pi\sigma}} \exp\left(-\frac{(y^{(i)}-\theta^T x^{(i)})^2}{2\sigma^2}\right) \\ &= m\log\frac{1}{\sqrt{2\pi\sigma}} - \frac{1}{\sigma^2}\cdot\frac{1}{2}\sum_{i=1}^{m}(y^{(i)}-\theta^T x^{(i)})^2 \end{aligned}$$

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{m}(h_\theta(x^{(i)}) - y^{(i)})^2$$

以上就得到了回归的损失函数最小二乘法的公式，再对损失函数进行优化，使得损失函数最小化。

$$\begin{aligned} L(W) &= \frac{1}{2}(XW-y)^T(XW-y) \\ &= \frac{1}{2}[W^T X^T XW - W^T X^T y - y^T XW + y^T y] \\ &= \frac{1}{2}[W^T X^T XW - 2W^T y + y^T y] \end{aligned}$$

$$\begin{aligned} &\frac{\partial L(W)}{W} = 0 \\ &\frac{\partial L(W)}{W} = \frac{1}{2}[W^T X^T X - 2X^T y] = 0 \\ &X^T XW = X^T y \\ &W = (X^T X)^{-1} X^T y \end{aligned}$$

上式为按矩阵方法优化损失函数，但此方法具有一定的局限性，就是要可逆。还可以采取梯度下降算法。

3 数据说明与描述

3.1 数据来源

本文数据源于腾讯网新型冠状病毒肺炎疫情实时追踪，采取爬虫技术提取。 采集数据包括全国确诊累计病例数、全国疑似累计病例数、全国累计死亡数、全国累计治愈数、现确诊病例数、死亡率、治愈率，数据截至 2020 年 2 月 19 日。

seqence	date	confirm	suspect	dead	heal	nowConfir	nowSever	deadRate	healRate
1	2020.01.1	41	0	1	0	0		2.4	0
2	2020.01.1	41	0	1	0	0		2.4	0
3	2020.01.1	41	0	2	5	0		4.9	12.2
4	2020.01.1	45	0	2	8	0		4.4	17.8
5	2020.01.1	62	0	2	12	0		3.2	19.4
6	2020.01.1	198	0	3	17	0		1.5	8.6
7	2020.01.1	275	0	4	18	0		1.5	6.5
8	2020.01.2	291	54	6	25	291		2.1	8.6
9	2020.01.2	440	37	9	25	431		2	5.7
10	2020.01.2	574	393	17	25	557		3	4.4
11	2020.01.2	835	1072	25	34	776		3	4.1
12	2020.01.2	1297	1965	41	38	1218		3.2	2.9
13	2020.01.2	1985	2684	56	49	1880		2.8	2.5
14	2020.01.2	2761	5794	80	51	2630		2.9	1.8
15	2020.01.2	4535	6973	106	60	4369		2.3	1.3
16	2020.01.2	5997	9239	132	103	5762		2.2	1.7
17	2020.01.2	7736	12167	170	124	7442		2.2	1.6
18	2020.01.3	9720	15238	213	171	9336		2.2	1.8
19	2020.01.3	11821	17988	259	243	11319		2.2	2.1
20	2020.02.0	14411	19544	304	328	13779		2.1	2.3
21	2020.02.0	17238	21558	361	475	16402		2.1	2.8
22	2020.02.0	20471	23214	425	632	19414		2.1	3.1
23	2020.02.0	24363	23260	491	892	22980		2	3.7
24	2020.02.0	28060	24702	564	1153	26343		2	4.1
25	2020.02.0	31211	26359	637	1542	29032		2	4.9
26	2020.02.0	34598	27657	723	2052	31823		2.1	5.9
27	2020.02.0	37251	28942	812	2651	33788		2.2	7.1
28	2020.02.0	40235	23589	909	3283	36043		2.3	8.2
29	2020.02.1	42708	21675	1017	3998	37693		2.4	9.4
30	2020.02.1	44730	16067	1114	4742	38874		2.5	10.6
31	2020.02.1	59882	13435	1368	5915	52599		2.3	9.9
32	2020.02.1	63932	10109	1381	6728	55823		2.2	10.5
33	2020.02.1	66576	8969	1524	8101	56951		2.3	12.2
34	2020.02.1	68584	8228	1666	9425	57493		2.4	13.7
35	2020.02.1	70635	7264	1772	10853	58010		2.5	15.4
36	2020.02.1	72528	6242	1870	12561	58097		2.6	17.3
37	2020.02.1	74279	5248	2006	14387	57886		2.7	19.4
38	2020.02.1	74675	4922	2121	16168	56386		2.8	21.7

图 1 原始数据

3.2 数据描述

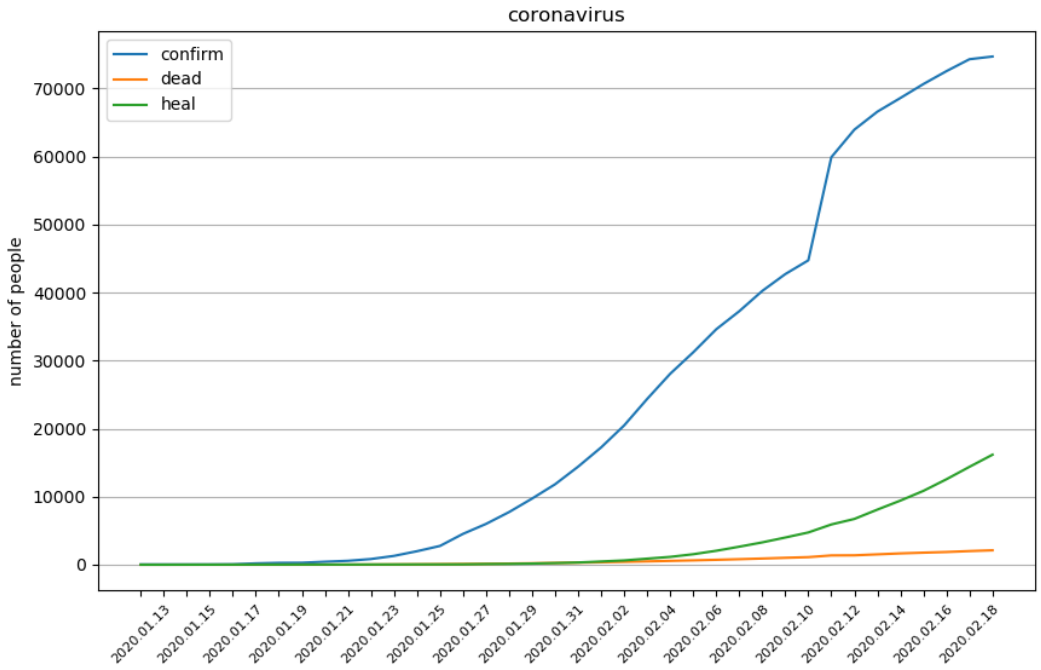

图 2 新型冠状病毒疫情数据线性图

使用 matplotlib 对数据可视化分析，绘制折线图

4 模型的建立

4.1 相关分析模型

4.1.实验结果

日期	确诊人数
2020.02.20	68960
2020.02.21	71235
2020.02.22	73510
2020.02.23	75784
2020.02.24	78059
2020.02.25	80333
2020.02.26	82608
2020.02.27	84882
2020.02.28	87157
2020.02.29	89432

本文以原始数据作为训练集通过 sklearn linear regression 模型，获取预测数据。

4.2 拟合检验

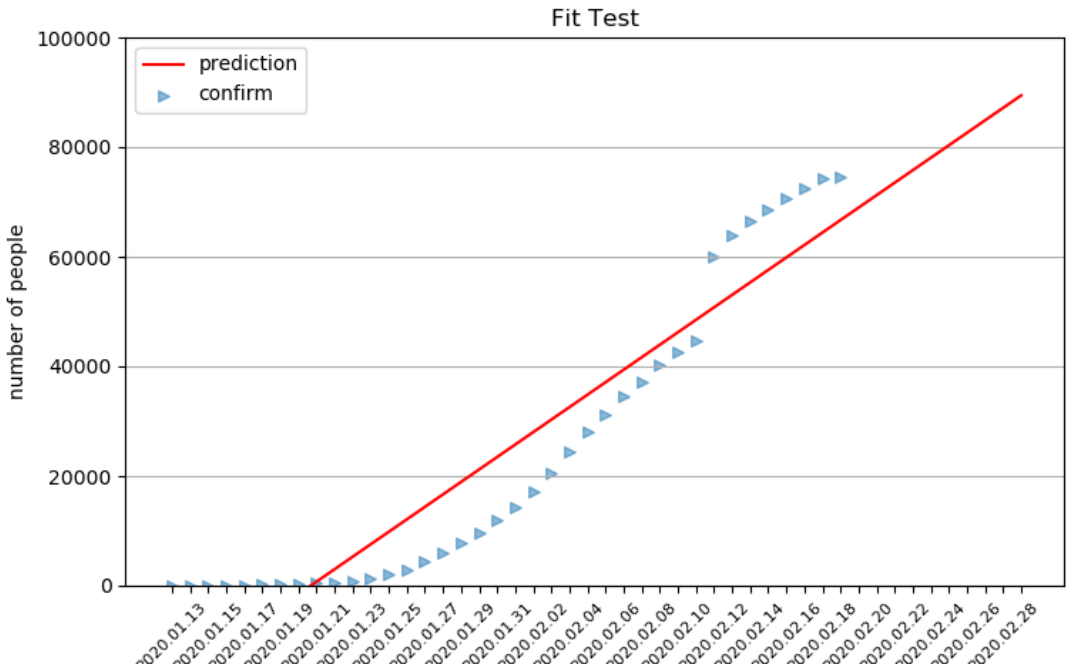

图 3 基于 sklearn 线性回归模型下的疫情预测折线

在图 3 中，1 月 13 日-2 月 19 日的拟合折线，其预测值与 2 月 19 日之前公布的确诊数据比较接近。虽然得到的是一次函数折线并存在一定的误差，但仍能为现阶段疫情发展提供参考。

5 结论与建议

本文借助 linear regression 模型，对新型冠状病毒疫情的累计确诊人数进行线性拟合。拟合结合疫情发展过程，拟合结果与实证数据基本吻合，能够为疫情的防控效果以及发展趋势提供参考。

近日，部分地区开始出现人群扎推现象，甚至有人已摘下口罩。虽然疫情得到了有效的控制，但拐点并没有到。现阶段仍处于疫情的增长期，应听从专家建议提高警惕，避免外出。

6 参考文献

[1]张琳. 新型冠状肺炎疫情传播的一般增长模型拟合与预测[J/OL]. 电子科技大学学报

[2] scikit-learn .api.

[3]孙荣恒．应用数理统计（第三版）．北京：科学出版社, 2014

[4]基于不确定环境下的线性回归分析模型[J]. 张天一，符一平，王志刚. 迷糊系统与数学. 2016（04）

[5] Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition. Aurélien Géron September 2019 .O'Reilly Media, Inc

[6]Python Machine Learning By ExampleYuxi (Hayden) Liu May 30, 2017

[7]Statistical Models:Theory and Practice. Cambridge University Press. David A. Freedman (2009)

[8] Linear Regression (Machine Learning)" . University of Pittsburgh.

7 附页（python 程序）

```
# —*— coding: utf-8 —*—
import requests
import json
import time
import pandas as pd

# 请求的 URL
url =
'https://view.inews.qq.com/g2/getOnsInfo?name=disease_h5&callback=&_=%d'

# 伪装请求头
headers = {
    'user-agent': 'Mozilla/5.0 (Windows NT 10.0; Win64; x64)
AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.130 Safari/537.36',
    'referer':
'https://news.qq.com/zt2020/page/feiyan.htm?from=timeline&isappinstalled=0'
}

# 抓取数据
r = requests.get(url % time.time(), headers=headers)

data = json.loads(r.text)
data = json.loads(data['data'])

lastUpdateTime = data['lastUpdateTime']
print('数据更新时间 ' + str(lastUpdateTime))

# part 2. 采集中国历史数据

print('采集中国历史数据...')

china_day_list = data['chinaDayList']

col_names_cd =  ['date','confirm','suspect','dead', 'heal', 'nowConfirm',
```

```
'nowSevere','deadRate','healRate']

my_df_cd = pd.DataFrame(columns = col_names_cd)

for day_item in china_day_list:
    date ='2020.'+day_item['date']
    confirm = day_item['confirm']
    suspect = day_item['suspect']
    dead = day_item['dead']
    heal = day_item['heal']
    nowConfirm = day_item['nowConfirm']
    nowSevere = day_item['nowSevere']
    deadRate = day_item['deadRate']
    healRate = day_item['healRate']

    # 向 df 添加数据
    data_dict = {'date': date,'confirm': confirm,'suspect': suspect,'dead':
dead, 'heal': heal, 'nowConfirm': nowConfirm,
                 'nowSevers':nowSevere,'deadRate':
deadRate,'healRate':healRate}
    my_df_cd.loc[len(my_df_cd)] = data_dict

my_df_cd.index += 1
my_df_cd.to_csv(r'./china_daily_status_{}.csv'.format(str(lastUpdateTime).s
plit()[0]), encoding='utf_8_sig', header='true')

print('Success')




from sklearn import linear_model
import pandas as pd
import matplotlib.pyplot as plt

# 读取数据
train = pd.read_csv('D:\PyCharm\沐风\ crawler\china_daily_status_2020-02-
21.csv')
test = pd.read_csv('D:\PyCharm\沐风\ crawler\predication03.csv')
```

```
test_confirm=pd.read_csv('D:\PyCharm\沐风\
crawler\predication03_confirm.csv')

seqence_train = train.loc[:,['seqence']]
sum_train= train.loc[:,['confirm']]

# linear regression
linear = linear_model.LinearRegression()

# 训练数据
linear.fit(seqence_train,sum_train)

# 预测
# test_seqence = test['seqence'].values.tolist()
predicted_sum = linear.predict(test)

# 输出
predict_sequnce_number = len(test)
print('predict for',predict_sequnce_number,'sequnce:')
for i in range(0,predict_sequnce_number):
    print('sequnce:',int(test.values[i]),'confirm:',int(predicted_sum[i]))

# 预测折线图
fig = plt.figure(figsize=[8.5, 5])
ax = fig.add_subplot(1,1,1)
# print(perdicted_sum)
date = pd.pivot_table(test_confirm,index=['date'],aggfunc='count')
# print(date.index)
ax.plot(date.index,predicted_sum,'-',label='prediction', alpha=1,c='r')
plt.xticks(date.index, rotation=45, fontsize=8)

ax.yaxis.grid(True)

# 散点图
date1 = pd.pivot_table(train,index=['date'],aggfunc='count')
ax.scatter(date1.index,train['confirm'],label='confirm',s=30,
alpha=0.5,marker='>')

# 隐藏
```

```
for label in ax.get_xticklabels()[1::2]:
    label.set_visible(False)

# 显示多标签
plt.legend()
plt.ylabel('number of people')

# title
ax.set_title('Fit Test')
plt.ylim(0,100000)
plt.show()
```