

西南财经大学

Southwestern University of Finance and Economics

课程论文

学年学期：2018-2019 第二学期

课程名称：Python 数据分析

论文题目：基于新浪微博评论的“996”情感分析

学生学号：41641034

学生姓名：郑涵昱

学 院：金融学院

年级专业：2016 级金融工程

评语：

得 分：

评阅教师签字：

年 月

一. 引言

“996”工作制是指许多互联网公司采用的工作制，指的是上午 9 点上班，晚上 9 点下班，一周工作 6 天。2019 年 3 月 27 日，一个名为 996.ICU 的项目在 GitHub 上传开。在这个项目指向的域名页面上，发起人这样写到：“什么是 996.ICU？工作 996，生病 ICU”。他将 996 工作制下最低 72 个工时与《劳动合同法》等条文对比，并呼吁“程序员生命为重（Developers’ lives matter）”。

在国内，“996”迅速成为一个热门话题，诸如马云、刘强东等互联网公司巨头竞相为“996”发声。但在 2019 年 4 月 11 日，人民日报针对“996 工作制”发表评论员文章《强制加班不应成为企业文化》；4 月 12 日，阿里巴巴通过其官方微信号上分享了马云有关 996 的一些观点，当天下午马云再度回应称“任何公司不应该，也不能强制员工 996”。

为了研究广大人民群众对于“996”工作制的态度，本文对[马云 2019 年 4 月 14 日发布的题为“再谈 996：理性讨论比结论更重要，周末愉快！”](#)的微博的评论进行情感分析。截止 2019 年 6 月 10 日，该微博已有 30954 次转发、96816 条评论以及 73974 个点赞。可以看出，该微博的关注度极高，一定程度上能够总体反映大众对“996”工作制的情感态度。

本文研究步骤包括数据挖掘、数据处理、情感分析、结论四步。数据挖掘部分主要解决从新浪微博爬取所需要的评论数据问题；数据处理部分主要包括处理缺失值、剔除重复值等；情感分析则是通过评论文本分析人们对于“996”的态度；结论部分包括结论可视化（如制作词云等）以及研究结果。整体框架如下：

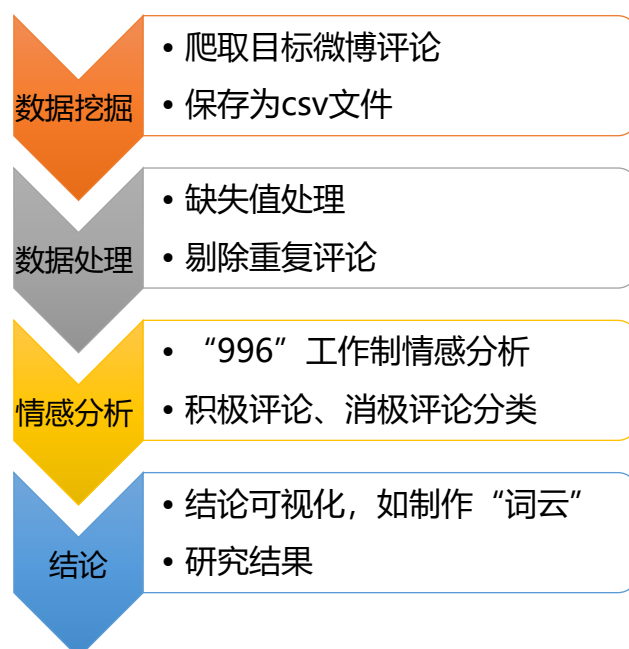


图 1 研究框架

二. 文献综述

文本情感分析：又称意见挖掘、倾向性分析等。简单而言,是对带有情感色彩的主观性文本进行分析、处理、归纳和推理的过程。互联网(如博客和论坛以及社会服务网络如大众点评)上产生了大量的用户参与的、对于诸如人物、事件、产品等有价值的评论信息。这些评论信息表达了人们的各种情感色彩和情感倾向性,如喜、怒、哀、乐和批评、赞扬等。基于此,潜在的用户就可以通过浏览这些主观色彩的评论来了解大众舆论对于某一事件或产品的看法。

赵妍妍等(2010)综合已有的研究成果,将情感分析归纳为 3 项层层递进的研究任务,即情感信息的抽取、情感信息的分类以及情感信息的检索与归纳。陈琳等(2017)根据微博网页的特点,通过模拟登录新浪微博,利用关键词匹配技术,匹配符合规定条件的微博,实时抓取微博中指定用户的微博正文等内容,并利用此方法雾霾相关的微博与时间关系。周中华等(2014)开发了一款支持并行的微博数据抓取工具,该工具可以实时抓取微博中指定用户的粉丝信息、微博正文等内容,并使用该工具对部分微博数据作了一个关于流感问题的分析。杜晓旭(2019)选择使用纯爬虫的方式,重点爬取移动端的新浪微博内容,以利用基于 Python 的新浪微博爬虫程序对关于“十九大”的微博数据进行抓取,通过对相关微博及微博发布时间进行分析。

国外方面, Alexander Pak etc. (2010)利用 Twitter 平台,并利用语料库,构建了一个情感分类器,它能够为一段文字确定其积极、消极和中性的情感,实验结果表明,其提出的方法具有较高的效率和较好的性能。Bai Xue etc. (2014)提出了一种基于语义指向点相似距离(SO-SD)模型的基于 Word2vec 工具的情感词典构建模型,然后我们使用情感词典来获得微博信息的情感倾向。

可以看出,国内外关于微博、推特等社交媒体内容的情感分析研究已经相当成熟,在此基础上,本文采用了特别参考了陈琳等(2017)的文章,通过模拟登录新浪微博爬取用户评论信息,并利用赵妍妍等(2010)提出的情感分析步骤进行情感分析。

三. 研究方法

3.1. 数据爬取

3.1.1. 爬取网址选取

新浪微博是中国最早兴起的自媒体平台,每个人都可以上面发表自己的观点。到现在微博已经成为了官方、明星等新闻发布的第一阵地。更为重要的是:微博不同于 QQ 空间、微信朋友圈,不需要对方加好友,也不需要关注对方,就可以看到对方的全部动态及个人信息。所以,微博变成了唯一可以爬的社交平台。

爬取微博评论有两个方向,一个是通过官方 OpenAPI 接口爬取,一个则是直接从网页爬取。但是官方 API 限制太多,非注册的开发人员只能获得少量信

息，因此本文采用第二个思路，通过网页爬取。需要注意的是，微博网页入口有三种，分别是微博 PC 端 (<http://weibo.com>)，微博手机端 (<http://m.weibo.cn>)，以及微博移动端(<http://weibo.cn>)。其中网页端是 Ajax 动态加载，爬取难度大，且限制多；手机端爬取容易，但是最多只能获取 100 页评论。综合爬取难度和爬取效果，本文最终选取微博移动端(<http://weibo.cn>)进行爬取。

3.1.2. 获取账号 Cookie

新浪微博网页端的评论必须登录后才能完全查看，因此想要爬取全部评论，也必须在代码中模拟登陆或者“伪装”登录。在程序设计中 Cookie 专门指的是一小段文本信息，服务器在接收到新浪微博用户客户端发送的请求之后，将会对这一新浪微博用户客户端的实际状态进行准确记录，并按照既定规则对微博账号进行加密处理。随后将加密后的微博账号及时存储在 Cookie 当中。此时客户端浏览器将会从 response 处获得其颁发的一个 Cookie，该 Cookie 在及时被客户端浏览器存储后，新浪网站在接收得到浏览器发送的请求时，浏览器将会连同 Cookie 与发送的请求网址一并提交至服务器。由服务器负责对 Cookie 进行检查，从而对客户端的实际状态进行准确辨别。

要获取账号 Cookie，首先登陆 m.weibo.cn，进入浏览器开发者模式，在 Network 选项中找到 m.weibo.cn，就可以在弹出的 Request Headers 中找到 Cookie。

在代码中，模拟登陆微博的代码如下：

```
def get_one_page(url):#请求函数：获取某一网页上的所有内容
    headers = {
        'User-agent' : 'your user-agent',
        'Host' : 'weibo.cn',
        'Accept' : 'application/json, text/plain, */*',
        'Accept-Language' : 'zh-CN,zh;q=0.9',
        'Accept-Encoding' : 'gzip, deflate, br',
        'Cookie' : 'your cookie',
        'DNT' : '1',
        'Connection' : 'keep-alive'
    }#请求头的书写，包括 User-agent,Cookie 等
    response = requests.get(url,headers = headers,verify=False)#利用
requests.get 命令获取网页 html
    if response.status_code == 200:#状态为 200 即为爬取成功
        return response.text#返回值为 html 文档，传入到解析函数当中
    return None
```

3.1.3. 分析网页 URL

观察网址：

<https://weibo.cn/comment/HpzmF8bpc?uid=2145291155&rl=0&page=2>，其中 uid 表示微博博主的 ID，每个微博用户都有独一无二的 ID，本文中即为马云的微博 ID；page=2 则表示这是评论的第二页。由此可以看出，所有评论页的 URL 格式都相同，区别只在于页码。目标微博共有 9682 页，为了保证选取评论的质量，本文选取第 500 页到 8500 页的评论。

在分析好 URL 格式后就可以使用 `requests.get` 命令获取目标网址 `html`。再使用 `html.text` 即可获得 `html` 文档，其中包含该页评论所有信息，包括评论内容、点赞数量、评论者 ID、发表时间等。根据本文研究需求，导出了评论内容和发表时间，代码如下：

```
def parse_one_page(html):#解析 html 并存入到文档 weibo.csv 中
    pattern_time = re.compile('<span class="ct">.*?&nbsp;',' re.S)
    pattern_comment = re.compile('<span class="ctt">.*?</a></span>',' re.S)
    items_time = re.findall(pattern_time,html)
    items_comment = re.findall(pattern_comment,html)
    data = pd.DataFrame(items_time, items_comment)
    data.to_csv('weibo.csv', header = False, mode='a+',encoding = "GB18030")
```

3.1.4. 设置循环爬取所有评论

完成以上步骤以后，就可以设置循环，对任意页码的微博评论进行爬取。本文为了避免最初和最近评论的影响，选取 500 页至 8500 页的评论进行分析。还需要注意每爬取一页评论后应该休息 1-3 秒，防止被系统误判为机器人而封锁 IP。代码如下：

```
for i in range(500,8500):
    url="https://weibo.cn/comment/HpzmF8bpc?uid=2145291155&rl=0&page="+str(i)
    html = get_one_page(url)
    parse_one_page(html)
    time.sleep(random.uniform(1,3))#每次循环后随机停顿 1-3 秒
```

3.2. 数据处理

3.2.1. 数据清洗

3.1. 部分爬取的微博数据有许多无关符号、标签等，如二级评论前会有“回复[a href=](#)”，表情符号前会有“<img alt=”，且每一条评论末尾有“”。可以使用正则表达式，利用 `re` 包中的 `re.sub` 函数即可去除不需要的内容。

接着用同样的方法去除标点符号，但是本文中并不去除感叹号与问号，原因是在情感分析部分，感叹号、问号前的情感词会有更高的权重，因此本部分只用去除感叹号、问号以外的其余标点符号。

3.2.2. 删除重复值、缺失值

评论中有很多重复值，可以使用 `pandas` 包中的 `drop_duplicates` 函数去除重复值。去除缺失值时，由于原评论数据为字符串类型，要先将空字符串替换为

np.nan，然后在使用 drop.na 函数去除缺失值。

3.3. 情感分析

3.3.1. 原理

情感分析就是分析一句话说得是很主观还是客观描述，分析这句话表达的是积极的情绪还是消极的情绪。本文主要采用了情感词典进行情感分析。

本部分以例句：“这手机的画面极好，操作也比较流畅。不过拍照真的太烂了！系统也不好。”为例阐述情感分析的原理

① 情感词

要分析一句话是积极的还是消极的，最简单最基础的方法就是找出句子里面的情感词，积极的情感词比如：赞，好，顺手，华丽等，消极情感词比如：差，烂，坏，坑爹等。出现一个积极词就+1，出现一个消极词就-1。

例句里面就有“好”，“流畅”两个积极情感词，“烂”一个消极情感词。那它的情感分值就是 $1 + 1 - 1 + 1 = 2$ 。很明显这个分值是不合理的，需要改进。

② 程度词

“好”，“流畅”和“烂”前面都有一个程度修饰词。“极好”就比“较好”或者“好”的情感更强，“太烂”也比“有点烂”情感强得多。所以需要在找到情感词后往前找一下有没有程度修饰，并给不同的程度一个权值。比如“极”，“无比”，“太”就要把情感分值乘4，“较”，“还算”就情感分值乘2，“只算”，“仅仅”这些就乘0.5了。那么这句话的情感分值就是： $4 \times 1 + 1 \times 2 - 1 \times 4 + 1 = 3$

③ 感叹号

可以发现太烂了后面有感叹号，叹号意味着情感强烈。因此发现叹号可以为情感值+2。那么这句话的情感分值就变成了： $4 \times 1 + 1 \times 2 - 1 \times 4 - 2 + 1 = 1$

④ 否定词

可以看出最后面那个“好”并不是表示“好”，因为前面还有一个“不”字。所以在找到情感词的时候，需要往前找否定词。比如“不”，“不能”这些词。而且还要数这些否定词出现的次数，如果是单数，情感分值就乘-1，但如果是偶数，那情感就没有反转，还是乘1。在这句话里面，可以看出“好”前面只有一个“不”，所以“好”的情感值应该反转，即乘负-1。

因此这句话的准确情感分值是： $4 \times 1 + 1 \times 2 - 1 \times 4 - 2 + 1 \times 1 = -1$

⑤ 积极和消极分开

再接下来，很明显就可以看出，这句话里面有褒有贬，不能用一个分值来表示它的情感倾向。而且这个权值的设置也会影响最终的情感分值，敏感度太高了。因此对这句话的最终的正确处理，是得出这句话的一个积极分值，一个消极分值（这样消极分值也是正数，无需使用负数了）。它们同时代表了这句话的情感倾向。所以这句评论应该是“积极分值：6，消极分值：7”

⑥ 以分句的情感为基础

再仔细一步，详细一点，一条评论的情感分值是由不同的分句加起来的，因此要得到一条评论的情感分值，就要先计算出评论中每个句子的情感分值。这条例子评论有四个分句，因此其结构如下（[积极分值，消极分值]）：[[4, 0], [2, 0], [0, 6], [0, 1]]

以上就是使用情感词典来进行情感分析的主要流程了，算法的设计也会按照这个思路来实现。

3.3.2. 算法设计

- 第一步：读取评论数据，对评论进行分句；
- 第二步：查找对分句的情感词，记录积极还是消极，以及位置；
- 第三步：在情感词前查找程度词，找到就停止搜寻。为程度词设权值，乘以情感值；
- 第四步：在情感词前查找否定词，找完所有否定词，若数量为奇数，乘以-1，若为偶数，乘以 1；
- 第五步：判断分句结尾是否有感叹号，有叹号则往前寻找情感词，有则相应的情感值+2；
- 第六步：计算完一条评论所有分句的情感值，用数组（list）记录起来；
- 第七步：计算并记录所有评论的情感值；
- 第八步：通过分句计算每条评论的积极情感均值，消极情感均值，积极情感方差，消极情感方差；

3.3.3. 情感分类与关键词排序

情感分析完成后，如果一条评论的积极情感值大于消极情感值，则将其归入积极评论（positive_comment），反之则归入消极评论（negative_comment）。由于有些评论字数很少，以至于没有情感词典中的词语，最终评分全为 0，这种评论很难进行分析，因此归为无用评论（no_use_comment）。接着使用 jieba 分词包括，使用哈工大停词表去除停词，再使用 jieba.analyse 包中的 jieba.analyse.extract_tags 函数，即可提取出现频率最高的关键词，在此基础上也可以制作词云。

四. 实验结果

4.1.数据爬取

数据爬取阶段的完整结果见附件 df.csv，本文简要展示前两行：

表格 1 微博评论数据爬取结果（部分）

	评论	时间
1	呵呵，理直气壮地违法，还剥削广大劳动人民还有理，利用诡辩论为自己辩解？什么样的国家里有什么样的剥削者！那么被剥削阶级怎么办呢？联合起来，反对剥削！ 举报 <a	04月23日11:17

	href="/attitude/HqW5LiuvM/update?object_type=comment&uid=3602934943&rl=1&st=d81aa1">赞[0]	
2	在血汗工厂里，一边用吸管吸着工人人们的血，一边高喊 你们是幸福的，要奋斗。 举报 赞[0]	04 月 23 日 11:10
.....		

64625 rows × 2 columns

4.2. 数据处理

数据处理阶段的完整结果见附件 [comments with punc.csv](#)，简要展示部分：

表格 2 数据处理后的评论数据（部分）

	评论	时间
1	呵呵理直气壮地违法还剥削广大劳动人民还有理利用诡辩论为自己辩解？什么样的国家里有什么样的剥削者！那么被剥削阶级怎么办呢？联合起来反对剥削！	04 月 23 日 11:17
2	在血汗工厂里一边用吸管吸着工人人们的血一边高喊你们是幸福的要奋斗	04 月 23 日 11:10
3	马老师需不需要助理我愿意直接在你手下工作 996 制跪求	04 月 23 日 11:10
4	我错了我不应该和你辩论加油你是最优秀的	04 月 23 日 11:09
5	劳动法自己翻	04 月 23 日 11:08
6	请不要当教育家了看一看共青团为您写的新闻稿	04 月 23 日 11:07

59765 rows × 2 columns

4.3. 情感分析

4.3.1. 情感评分

所有积极评论情感值见附件 [positive_comment.csv](#)，消极评论情感值见附件 [negative_comment.csv](#)。部分结果展示如下：

表格 3 积极评论情感值评分

	评论	积极评分	消极评分	积极均分	消极均分
1	在血汗工厂里一边用吸管吸着工人人们的血一边	5	0	0	0.5

	高喊你们是幸福的要奋斗				
2	马老师需不需要助理我愿意直接在你手下工作 996 制跪求	9	0	0	0.5
3	我错了我不应该和你辩论加油你是最优秀的	8	7	0.5	1.4
4	你说你会展示资本家的獠牙面目但你确实在展示	3	0	0	0.4
.....					

21176 rows × 5 columns

表格 4 消极评论情感值评分

	评论	积极评分	消极评分	积极均分	消极均分
1	呵呵理直气壮地违法还剥削广大劳动人民还有理利用诡辩论为自己辩解？什么样的国家里有什么样的剥削者！那么被剥削阶级怎么办呢？联合起来反对剥削！	36	177	4.8	1.6
2	请不要当教育家了看一看共青团为您写的新闻稿	0	11	0.9	0
3	支持马老师如果你有好爹你可以到处旅游老是媚外就是奴性的表现生存在这个社会之上本身就不公平你想获得更多你就去努力没毛病	18	105	2.8	1.6
4	棺材都买不起了直接海里随便扔就好了因为连坟墓都买不起	0	19	1.2	0
.....					

24287 rows × 5 columns

4.3.2. 关键词排序

表格 5 关键词筛选与排序

积极评论		消极评论		积极评论		消极评论	
关键词	频次	关键词	频次	关键词	频次	关键词	频次
996	4304	996	4574	国家	139	鸡汤	139
奋斗	1269	加班	884	劳动法	138	中国	136
工作	877	员工	700	别人	137	事业	133
员工	752	工作	665	知道	136	孩子	132
加班	733	马云	606	希望	133	回报	130
热爱	530	不是	582	家人	131	需要	128
马云	521	奋斗	536	有钱	130	洗脑	127
老板	380	资本家	475	不要	126	家庭	127
生活	377	没有	450	家庭	125	工作制	124
理想	362	老板	397	年轻人	124	普通人	121
不是	356	剥削	371	工作制	124	成功	119

努力	343	加班费	364	个人	123	家人	117
资本家	341	时间	300	问题	122	年轻人	116
时间	339	工资	296	科学家	121	知道	115
成功	298	生活	286	人生	120	很多	112
喜欢	295	公司	273	中国	118	这种	105
企业	287	企业	270	鸡汤	115	阿里巴巴	104
老师	272	阿里	264	强制	111	下班	103
事业	250	问题	259	打工	106	个人	99
工资	250	不要	258	追求	106	其实	97
付出	250	选择	246	孩子	105	梦想	96
公司	249	强制	241	赚钱	105	关键	96
法律	245	老师	233	不能	103	言论	95
快乐	242	现在	231	利益	101	要求	94
愿意	241	热爱	228	事情	101	反对	94
阿里	240	觉得	227	洗脑	101	讨论	93
加班费	234	劳动法	219	普通员工	99	自愿	92
很多	231	偷换概念	202	下班	99	每天	91
觉得	231	小时	196	大部分	99	可能	90
梦想	230	压榨	190	理解	96	打工	84
选择	230	违法	187	价值	96	人民	83
没有	228	不能	183	其实	96	劳动者	82
现在	207	努力	174	阿里巴巴	95	爸爸	82
小时	207	付出	172	大多数	95	快乐	82
7	198	法律	169	真正	93	普通员工	82
偷换概念	195	社会	159	压榨	93	大部分	80
社会	179	真的	158	言论	92	现实	80
每个	178	大家	154	找到	91	不想	80
企业家	172	喜欢	154	想要	89	不好	79
应该	170	理想	153	每天	89	支持	78
幸福	160	955	151	思考	88	科学家	78
普通人	158	愿意	149	讨论	87	997	77
上班	158	企业家	147	可能	87	淘宝	77
回报	156	别人	147	这种	85	出来	77
需要	153	不会	147	24	84	不了	76
自愿	149	上班	146	健康	82	事情	76
剥削	149	应该	145	不会	82	人生	75
真的	148	7	143	劳动者	81	辞职	74
955	146	每个	141	提升	80	大多数	73
大家	140	国家	140	当然	79	劳动	73

词云



图 2 消极评论关键词词云



图 3 积极评论关键词词云

五. 结论

从情感分析结果来看,有效评论 45463 条中,有 21176 被判定为积极评论,24287 条被判定为消极评论。因此总体而言,群众倾向于反对“996”。

从积极评论来看,出现较多频次的词是“奋斗”、“热爱”、“成功”、“理想”等积极性词语,且与目标微博正文表达思想一致,也就是说有很多人在一定程度上赞同马云有关“996”工作制的观点。

从消极评论来看,出现较多频次的词是“资本家”、“剥削”、“压榨”、“加班费”等词语,一定程度上反映了人们对“996”工作制的反对观点:这是资本家的剥削、压榨;“996”意味着奋斗是偷换概念。

总的来说,分析结果表明大部分人们对“996”工作制持反对态度,这也与主流舆论相符合,但是还是有一定数量的人支持马云的观点,认为“996”工作制“是福报”或“是奋斗”。

六. 参考文献

- [1] Agarwal, A., Xie, B., Vovsha, I., Rambow, O., & Passonneau, R. (2011). Sentiment analysis of twitter data. In Proceedings of the Workshop on Language in Social Media (LSM 2011) (pp. 30-38).
- [2] Alaei, A. R., Becken, S., & Stantic, B. (2019). Sentiment analysis in tourism: capitalizing on big data. *Journal of Travel Research*, 58(2), 175-191.
- [3] Kouloumpis, E., Wilson, T., & Moore, J. (2011, July). Twitter sentiment analysis: The good the bad and the omg!. In Fifth International AAAI conference on weblogs and social media.
- [4] Nasukawa, T., & Yi, J. (2003, October). Sentiment analysis: Capturing favorability using natural language processing. In Proceedings of the 2nd international conference on Knowledge capture (pp. 70-77). ACM.
- [5] Pak, A., & Paroubek, P. (2010, May). Twitter as a corpus for sentiment analysis and opinion mining. In *LREc* (Vol. 10, No. 2010, pp. 1320-1326).
- [6] Pang, B., & Lee, L. (2008). Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, 2(1-2), 1-135.
- [7] Taboada, M., Brooke, J., Tofiloski, M., Voll, K., & Stede, M. (2011). Lexicon-based methods for sentiment analysis. *Computational linguistics*, 37(2), 267-307.
- [8] Ye, Y., & Tan, X. (2019, May). Visualization of Sina Weibo Propagation and Sentiment Analysis. In 2019 International Conference on Computer, Network, Communication and Information Systems (CNCI 2019). Atlantis Press.
- [9] 王英杰.基于 Python 的微博数据爬虫程序设计研究[J].信息与电脑(理论版),2018(23):93-94.
- [10] 张紫琼,叶强,李一军.互联网商品评论情感分析研究综述[J].管理科学学报,2010,13(06):84-96.
- [11] 赵妍妍,秦兵,刘挺.文本情感分析[J].软件学报,2010,21(08):1834-1848.

[12]周中华,张惠然,谢江.基于 Python 的新浪微博数据爬虫[J].计算机应用,2014,34(11):3131-3134.

七. 课程总结

一学期前我因为对 Python 的好奇选修了《Python 数据分析》课程，一学期来收获良多，首先要非常感谢郑海超老师一学期的细心准备和辛勤教学。我将从课程收获、给之后选修此课程的同学的建议两方面总结这学期的学习。

收获方面，一方面，我学到了许多 Python 基础包，特别是 NumPy 和 Pandas，极大方便了处理超大表格，比 Excel 更快捷高效；其次，我从课堂学到了爬虫的基本方法，并在自己的学习探究中学会了更高难度的爬虫，如爬取微博等具有反爬虫的网站，这也是我课程论文的难点部分；我也接触到了基础的机器学习思想与算法，我认为这非常重要，因为人工智能是目前的潮流，金融学也急需这样的思想与方法，我需要更深入地学习，我非常希望能在这一方面进行深造。

最重要的，我认为并不是这些知识性的东西，而是在学习编程时，一定要“learn to learn”，也就是学会学习。因为编程内容太多，方法太灵活，不可能在课程中学习到所有东西，我反而从郑老师分享的书籍、学习网站中收获更多。比如从《Python Data Science Handbook》学习了 matplotlib 的基础应用，从《Hands On Machine Learning》中学到了 KNN、K-means 等分类算法。在完成自己的个人项目时，我也通过广泛查找学会了反反爬虫、制作词云等操作。因此我认为学习编程最重要的就是学会学习。

我有两条给之后选修此课程的同学建议，1. 一定要课前预习，老师都会把代码、PPT 提前发出来，如果不预习，上课时会有点不知所措，某种程度上，预习甚至比复习更重要；2. 一定要多查资料，不要遇到问题就去问老师问别人，GitHub，CSDN，Stack Overflow 都是重要的学习资源。

最后，再次感谢郑老师一学期的教学，祝老师身体健康，工作顺利！