电子商务数据分析

第2章 数据采集与预处理

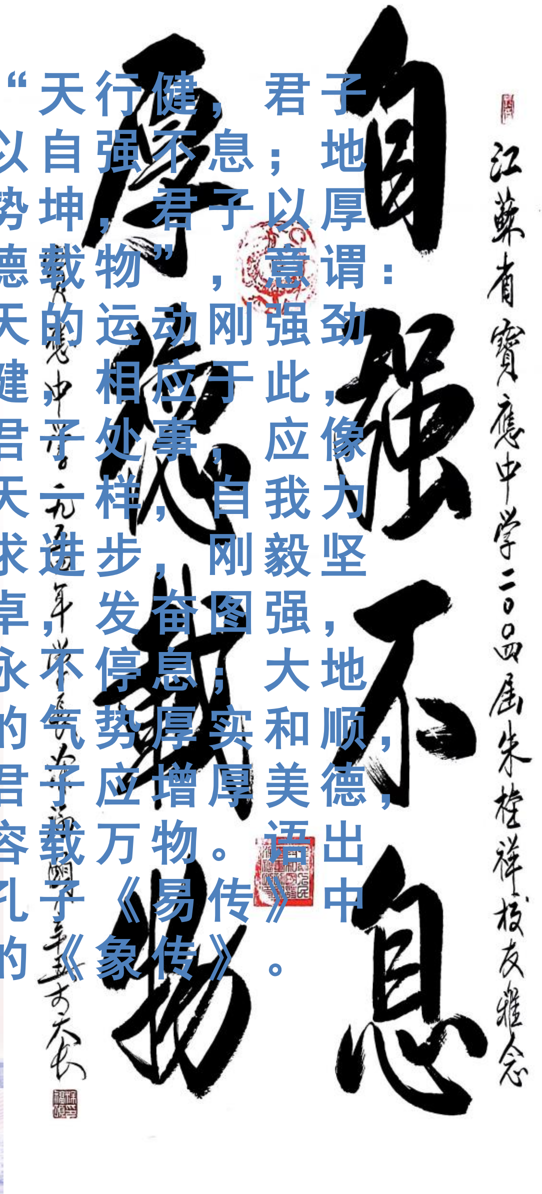
朱桂祥 (9120201070@nufe.edu.cn)
南京财经大学信息工程学院
江苏省电子商务重点实验室
电子商务信息处理国家级国际联合研究中心
电子商务交易技术国家地方联合工程实验室

https://github.com/zgx881205/Analysis-of-E-commerce-Data

南京财经大学 NANJING UNIVERSITY OF FINANCE & ECONOMICS

校训是什么?

清华校训：自强不息 厚德载物



“天行健，君子以自强不息；地势坤，君子以厚德载物”，意谓：天的运动刚强劲健，相应于此，君子处事，应像天一样，自我力求进步，刚毅坚卓，发奋图强，永不停息；大地的气势厚实和顺，君子应增厚美德，容载万物。语出孔子《易传》中的《象传》。

https://tv.cctv.com/2014/08/17/VIDE1408233723621465.shtml

数据采集与预处理

为什么要有数据采集和预处理？
(1)数据是大数据分析的原材料，但原始数据(raw data)往往不能直接使用。面临着不规范，不完整的问题。
(2)数据采集可以解决数据不完整的问题，而预处理可以解决不规范的问题。
(3)数据采集，预处理是数据分析的前期工作，并在整个大数据分析中起到非常重要的基础作用。如果数据出错了，后面的分析工作往往受重要影响。浮沙之上，难建高楼。
(4) 数据采集面临着6V中的规模性(Volume)，多样性(Variety)，高速性(Velocity)，价值高(Value)等诸多挑战。

预处理面临着大数据6V中的Veracity(真实性)的问题。

数据采集与预处理

数据从哪里来？

(1)系统的内部数据
系统内部的数据库，各种文档，图片，音频和视频。
系统内部的业务数据，人员数据，日程事务数据等。

(2)系统的外部数据
政府公开的数据，竞争对手的情报数据，社交网站的舆情数据，与业务相关的外部支撑数据，聘用新员工所需的人力资源数据等。

(3)大数据分析需要解决数据外部性的痛点。
只有内外互补，才可能解决问题。

数据采集与预处理

怎样采集数据？

(1)对于结构化的日常事务数据，在线表单，在线调查，线下问卷调查，存储在数据库。
(2)对于用户消费者行为数据，网络日志采集。
(3)互联网上面海量的公开数据，网络爬虫
(4)物联网上的海量数据。各种设备感知，存储。
(5)第三方数据库。

数据采集与预处理

怎样采集数据？

(1)对于结构化的日常事务数据，在线表单，在线调查，线下问卷调查，存储在数据库。一个大学校就业调查的例子[1]。

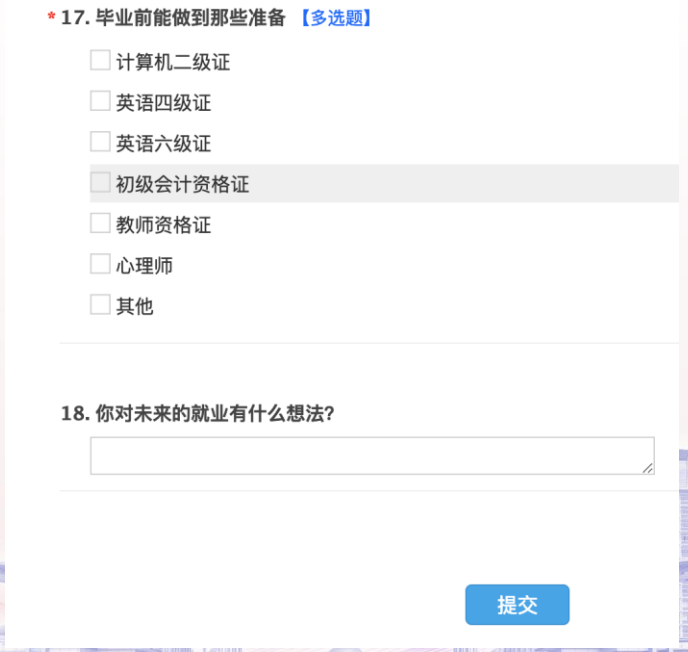
当代大学生就业意向的调查分析问卷

据了解，本次调查的目的在于分析、研究大学生就业发展中存在的问题及企业对大学生素质的要求，找出人才培养与人才需求的分歧，让大学生认清现状，有针对性地加强对自身的培养。

*1. 您的性别:
- 男
- 女

*2. 请问你是大几的学生
- 大一
- 大二
- 大三
- 大四
- 已毕业

*3. 大学的规划
- 很明确
- 一般般
- 没规划



*17. 毕业前能做到那些准备 【多选题】
- 计算机二级证
- 英语四级证
- 英语六级证
- 初级会计资格证
- 教师资格证
- 心理师
- 其他

18. 你对未来的就业有什么想法?

提交

https://www.wjx.cn/jq/35173829.aspx

数据采集与预处理

怎样采集数据？

(2)对于用户消费者行为数据，网络日志采集。

2.1 浏览器数据采集(BS模式(Brower Server))

通过html5,javascript用于收集用户通过上网泄漏的各种信息，包括地理位置，IP地址，照片，语音，浏览器版本等信息。结合大数据，可实现广告定向投放，用户追踪，用户行为分析，用户群体调研等一系列更人性化的服务。

2.2 户端数据采集(CS模式(Client Server))

Client/Server结构(C/S结构)是大家熟知的客户机和服务器结构。它是软件系统体系结构，通过它可以充分利用两端硬件环境的优势，将任务合理分配到Client端和Server端来实现，降低了系统的通讯开销。
目前大多数应用软件系统都是Client/Server形式的两层结构，由于现在的软件应用系统正在向分布式的Web应用发展，Web和Client/Server

数据采集与预处理

怎样采集数据？

(3)互联网上面海量的公开数据，网络爬虫 （八爪鱼[1]）

云采集

5000台云服务器，24*7高效稳定采集，结合API可无缝对接内部系统，定期同步爬取数据

智能防封

自动破解多种验证码 ，模拟真实用户访问，提供全球最大代理IP池，结合UA切换，不怕防采集

海量模板

内置400+网站数据爬虫模版，全面覆盖多个行业，只需简单设置，就可快速准确获取数据

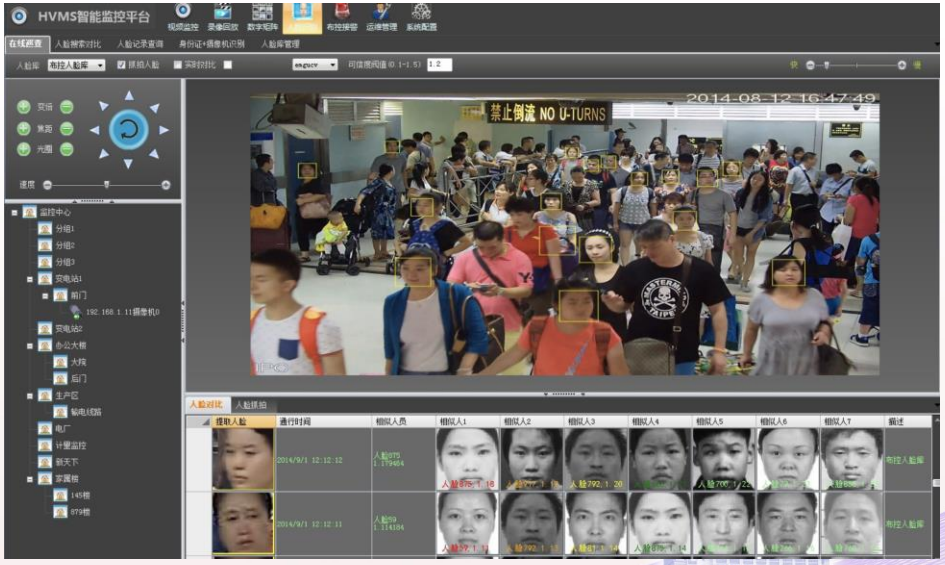
http://www.bazhuayu.com/

数据采集与预处理

怎样采集数据？

(4)物联网上的海量数据。各种感知，存储设备。

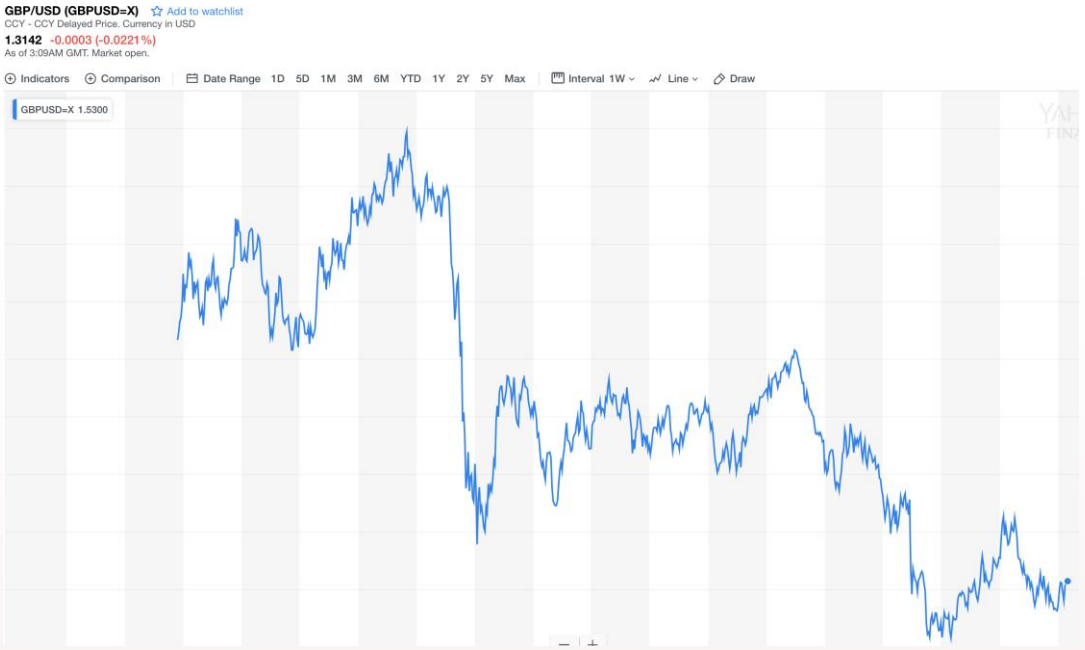
视频数据：高清防抖摄像头，获得关键信息(人脸，车牌等)。
语音数据：麦克风阵列，消除背景噪声。
传感器数据采集：智慧农业中的温度传感器，湿度传感器等。



数据采集与预处理

怎样采集数据？

(4)第三方数据库公开数据
金融数据：雅虎财经
(如右图，从2013年12月至今英镑和美元的汇率波动数据)



学术数据库：
知网，Elsevier，arXiv等。

数据采集与预处理

网络爬虫

为什么要使用网络爬虫？

(1)网络爬虫，搜索引擎背后的基础技术。百度和谷歌搜索显示的页面，都源自于网络爬虫每天不停的工作。
(2)网络爬虫可以一次下载大量网页。
(3)网络爬虫和各种网站开放的API有什么不同？
(4)多源异构的互联网开放数据,经过预处理，数据融合以后的有价值数据。

数据采集与预处理

网络爬虫

网络爬虫(Web Crawler)的定义：

（1）也叫网络蜘蛛(Web Spider)，利用HTTP 协议，根据超链接和Web 文档检索的方法遍历Web空间的程序，是一种“自动化浏览网络”的程序，或者说是一种网络机器人[1]。

（2）A Web crawler, sometimes called a spider or spiderbot and often shortened to crawler, is an Internet bot that systematically browses the World Wide Web, typically for the purpose of Web indexing[2].

[1]大数据分析，曹杰等编著
[2]维基百科:https://en.wikipedia.org/wiki/Web_crawler

数据采集与预处理

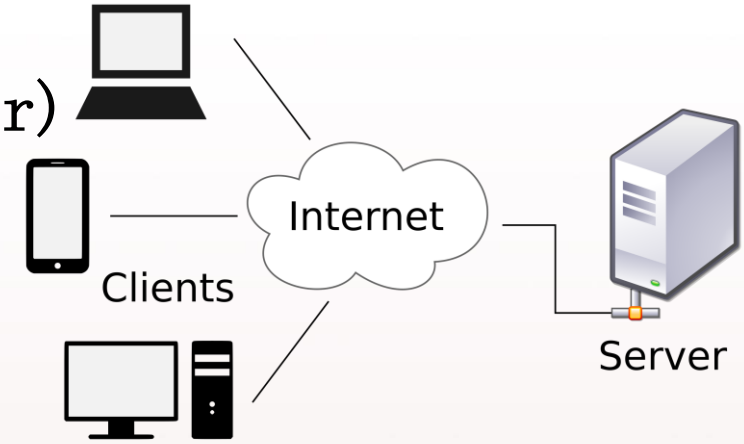
网络爬虫

网络爬虫的分类：
(1)全网爬虫:搜集整个互联网的网页(百度，谷歌，搜狗等)
(2)主题网络爬虫：特定需求的爬虫，比如八爪鱼
(3)增量式网络爬虫:不抓取重复的数据，保证新数据和旧数据的唯一性。
(4)深层网络爬虫：深层网络爬虫对应深层网络数据。
表层网络数据：网页显示的内容。
深层网络数据：藏在网页背后的数据库的内容，并没有完全通过网页展示出来。或者是特定用户才有权限看到的内容。

数据采集与预处理

网络爬虫

网络爬虫的预备基础知识：
客户端-服务器CS模式(Client - Server)



客户端向服务器发起请求(request)，
服务端向客户端给予响应(response)。

根据具体的业务不同，不同的请求
和响应规则由不同的协议来定义。
包括HTTP(Hypertext Transfer Protocol)
协议,HTTPS(Hypertext Transfer
Protocol Secure)等等。

数据采集与预处理

常用的网络爬虫工具：

- **Scrapy: Scrapy 是基于python的一个库，为了抓取网页数据、提取结构性数据而编写的应用框架，该框架是封装的，包含 request （异步调度和处理）、下载器（多线程的 Downloader）、解析器（selector）和 twisted（异步处理）等。对于网站的内容爬取，其速度非常快捷。优点：通过管道的方式存入数据库，灵活，可保存为多种形式。缺点：无法用它完成分布式爬取。**

- **PySpider：一个国人编写的强大的网络爬虫系统并带有强大的WebUI。采用Python语言编写，分布式架构，支持多种数据库后端，强大的WebUI支持脚本编辑器，任务监视器，项目管理器以及结果查看器。Python脚本控制，可以用任何你喜欢的html解析包。**

- **Nutch是为搜索引擎设计的爬虫，Nutch运行的一套流程里，有三分之二是为了搜索引擎而设计的。对精抽取没有太大的意义。也就是说，用Nutch做数据抽取，会浪费很多的时间在不必要的计算上。而且如果你试图通过对Nutch进行二次开发，来使得它适用于精抽取的业务，基本上就要破坏Nutch的框架，把Nutch改的面目全非。**

数据采集与预处理

网络爬虫的预备基础知识：
HTTP 协议

HTTP协议基于**CS**模式，是一种请求响应的协议。
通常请求由客户机上的**Web**浏览器发出，而服务器上的**Web**网站收到请求，给出响应。

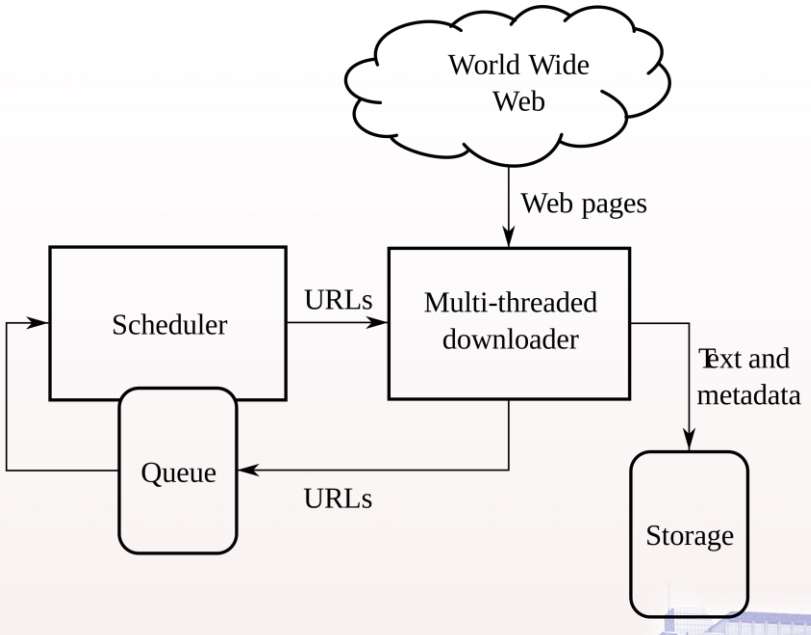
右边是一个访问**Wiki**百科的例子。包括**Request, Response Header, Response body**三个部分。

```
josh@blackbox:~$ telnet en.wikipedia.org 80
Trying 208.80.152.2...
Connected to rr.pmtpa.wikimedia.org.
Escape character is '^]'.
GET /wiki/Main_Page http/1.1                                                    Request
Host: en.wikipedia.org

HTTP/1.0 200 OK                                                                 Response headers
Date: Thu, 03 Jul 2008 11:12:06 GMT
Server: Apache
X-Powered-By: PHP/5.2.5
Cache-Control: private, s-maxage=0, max-age=0, must-revalidate
Content-Language: en
Vary: Accept-Encoding,Cookie
X-Vary-Options: Accept-Encoding;list-contains=gzip,Cookie;string-contains=enwikiToken;string-contains=enwikiLoggedOut;string-contains=enwiki_session;
string-contains=centralauth_Token;string-contains=centralauth_Session;string-contains=centralauth_LoggedOut
Last-Modified: Thu, 03 Jul 2008 10:44:34 GMT
Content-Length: 54218
Content-Type: text/html; charset=utf-8
X-Cache: HIT from sq39.wikimedia.org
X-Cache-Lookup: HIT from sq39.wikimedia.org:3128
Age: 3
X-Cache: HIT from sq38.wikimedia.org
X-Cache-Lookup: HIT from sq38.wikimedia.org:80
Via: 1.0 sq39.wikimedia.org:3128 (squid/2.6.STABLE18), 1.0 sq38.wikimedia.org:80 (squid/2.6.STABLE18)
Connection: close

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">   Response body
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en" dir="ltr">
        <head>
                <meta http-equiv="Content-Type" content="text/html; charset=utf-8" />
                                <meta name="keywords" content="Main Page,1778,1844,1863,1938,1980 Summer Olympics,2008,2008 Guizhou riot,2008 Jerusal
...
''' This content has been removed to save space
...
"Non-profit organization">nonprofit</a> <a href="http://en.wikipedia.org/wiki/Charitable_organization" title="Charitable organization">charity</a>.<b
r /></li>
                                <li id="privacy"><a href="http://wikimediafoundation.org/wiki/Privacy_policy" title="wikimedia:Privacy policy">Privac
y policy</a></li>
                                <li id="about"><a href="/wiki/Wikipedia:About" title="Wikipedia:About">About Wikipedia</a></li>
                                <li id="disclaimer"><a href="/wiki/Wikipedia:General_disclaimer" title="Wikipedia:General disclaimer">Disclaimers</a>
</li>
                        </ul>
                </div>
</div>

                <script type="text/javascript">if (window.runOnloadHook) runOnloadHook();</script>
<!-- Served by srv93 in 0.050 secs. --></body></html>
Connection closed by foreign host.
josh@blackbox:~$
```

数据采集与预处理

网络爬虫

网络爬虫的技术原理（如右图所示）：
(1)调度器(Scheduler)输入种子URL
(2)下载器（Downloader）下载相关页面。
(3)下载器从下载的页面中，由挖网页解析器抽取关联的URLs,放入调度器的队列Queue,等待下一轮处理。
(4)下载器把下载得到的页面进行存储。
(5)调度器从增量的URLs开始新一轮任，重复步骤(1)-(4)。
(6)整个过程迭代，直到队列中的URLs列表为空，停止下载。



数据采集与预处理

网络爬虫

网络爬虫实战的常用工具。
C/C++等语言一般用于百度等搜索引擎公司，用于设计通用的搜索引擎，但是由于实现比较复杂，不适合初学者。

Python相比较C/C++而言，具有简单易学，功能较全的特点。

Python的url和urllib：由URLs列表得到网页内容，
Re库：通过正则表达式解析下载网页中的URLs，并放到队列Queue中。

整个爬虫框架Scrapy = Scrach(抓取)+Python

数据采集与预处理

网络爬虫:一个Scrapy的例子：抓取某网站的内容(1).

```
import scrapy


class QuotesSpider(scrapy.Spider):
    name = 'quotes'
    start_urls = [
        'http://quotes.toscrape.com/tag/humor/',
    ]

    def parse(self, response):
        for quote in response.css('div.quote'):
            yield {
                'text': quote.css('span.text::text').get(),
                'author': quote.xpath('span/small/text()').get(),
            }

        next_page = response.css('li.next a::attr("href")').get()
        if next_page is not None:
            yield response.follow(next_page, self.parse)
```

```
scrapy runspider quotes_spider.py -o quotes.json
```

http://quotes.toscrape.com/tag/humor/

数据采集与预处理

网络爬虫:一个Scrapy的例子：抓取某网站的内容(2).

```
[{
    "author": "Jane Austen",
    "text": "\u201cThe person, be it gentleman or lady, who has not pleasure in a good novel,
},
{
    "author": "Groucho Marx",
    "text": "\u201cOutside of a dog, a book is man's best friend. Inside of a dog it's too dar
},
{
    "author": "Steve Martin",
    "text": "\u201cA day without sunshine is like, you know, night.\u201d"
},
...]
```

数据采集与预处理

```
# -*- encoding:utf-8 -*-
import pyodbc
import sys
import csv
import datetime
import time
import re
import os
reload(sys)
sys.setdefaultencoding('utf-8')
#conn=pymssql.connect(host='192.168.0.184',user='sa',password='pwd',database='ShcemDW')
starttime=datetime.datetime.now()
#conn=pymssql.connect(host='.',database='TUNIU-BI198',charset="utf8")
conn = pyodbc.connect('DRIVER={SQL Server};SERVER=localhost;PORT=1433;DATABASE=TUNIU-BI198',charset="utf8")
cur=conn.cursor()
#file1='ID CtiyName Type=0 1 3 19.txt'
#file1='ticket type=19 ID City.csv'
file1='final City jwd.txt'
file1='new.txt'
Dir=os.path.abspath('')
#print Dir+'/提取全球目的地经纬度/'+file1
#fin1=open(Dir+'/提取全球目的地经纬度/'+file1,'r')
fin1=open(file1,'r')
content=fin1.readlines()
for line in content:
    line=line.strip('\n')
    #line=line.encode('utf-8')
    list1=line.split(' ')
    CityName=list1[0].encode('utf-8')
    wd=list1[1]
    jd = list1[2]
    #print CityName,wd,jd
    #sql="insert into [Tuniu purchase prediction training data].[dbo].[Type=0 1 3 19 TypeID CityName] VALUES (" + str(TypeID) + ",'" + unicode(CityName)+ "')"
    sql = "insert into [Distinct City JWD made by 20180125]  VALUES ('" + unicode(CityName) + "','" + str(wd)+"','"+str(jd)+ "')"
    print sql
    cur.execute(sql)
fin1.close()
conn.commit()
cur.close()
conn.close()
#[Tuniu purchase prediction training data].[dbo].[Type=0 1 3 19 TypeID CityName]
```

数据采集与预处理

网络爬虫与反爬虫技术：

为什么要反爬虫？
(1)爬虫消耗了大量的服务器响应资源，使得正常的响应变慢。
(2)爬虫会盗用一部分网站不想公开的数据和信息。

反爬虫的主要技术有哪些？
(1)基于Headers反爬虫：浏览器访问服务器，在Headers中有User-Agent-Referer字段。爬虫可以模拟浏览器绕过此限制。
(2)基于用户行为反爬虫:同一用户短时间大量访问某网站。
爬虫应对策略:使用代理IP或降低访问频率。
(3)动态页面反爬虫:模拟AJAX请求,或是模拟浏览器发送动态请求。
(4)Cookie限制:Cookie检验。
(5)验证码限制：拖动某个图片，或输入某个字母进行手动验证。

数据采集与预处理

网络爬虫的法律与道德约束：
合理合法地获得网上的数据
(1)未经授权，不得擅自将有版权的数据公布，供人下载。
(2)不得擅自下载，或者暴力破解数据。
(3) 不得违规下载涉密数据。
(4) 遵循robots协议，那些页面能够被抓取，那些页面不能被抓取。

数据采集与预处理

电子商务数据的采集：

数据的来源及分类：
(1)电子商务数据平台的基础数据
(2)电商专业网站的研究数据
(3)基于电商媒体的数据
(4)基于电商评论的数据

电商平台的数据采集：
(1)HTML网页文本，图片-爬虫
(2)JSON或XML文本-API

电商平台数据采集的困难：
Volume(数据量大)
Variety(种类太多)：包括平台，研究数据，媒体数据等等。
Velocity(流式数据，高速性)
反爬虫技术，数据孤岛等等。

数据预处理

■ **Python Scrapy爬虫**

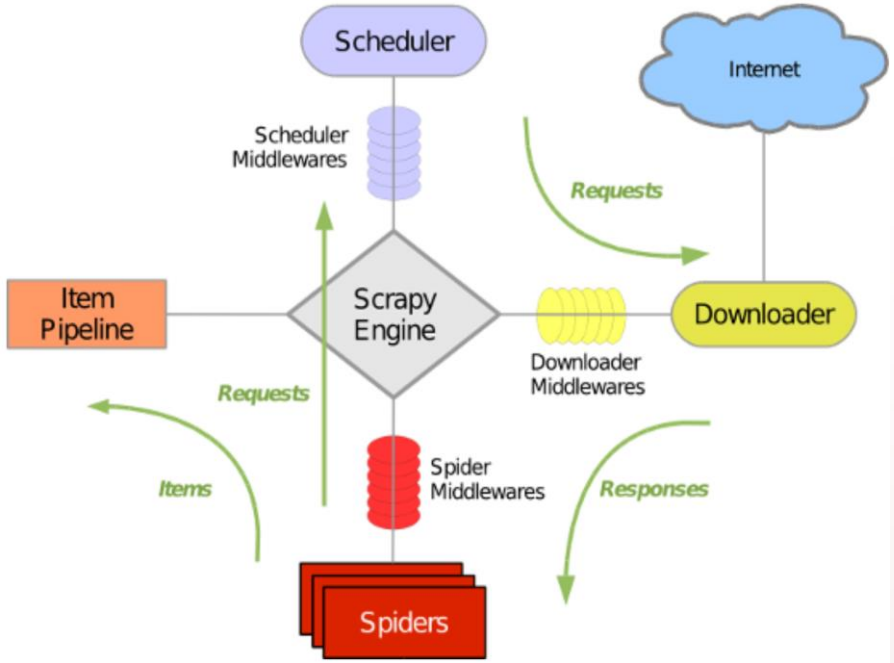
所需环境：

python

scrapy

xpath or BeautifulSoup

urllib.request



爬虫步骤：

1. 确定数据所在的地址（分析<网页性质>）

2.代码请求地址数据

3.解析数据（xpath、BeautifulSoup等）

4.数据保存(csv、xlsx、sql server和mongodb等数据库)

数据预处理

■ Python Scrapy案例（豆瓣Top250电影）

1.打开url并返回BeautifulSop对象：

```
# -*- coding:utf-8 -*-

from urllib.request import urlopen
from bs4 import BeautifulSoup
from collections import defaultdict
import pandas as pd
import time
import re


class DoubanMovieTop():
    def __init__(self):
        self.top_urls = ['https://movie.douban.com/top250?start={0}&filter='.format(x*25) for x in range(10)]
        self.data = defaultdict(list)
        self.columns = ['title', 'link', 'score', 'score_cnt', 'top_no', 'director', 'writers', 'actors', 'types',
                        'edit_location', 'language', 'dates', 'play_location', 'length', 'rating_per', 'betters',
                        'had_seen', 'want_see', 'tags', 'short_review', 'review', 'ask', 'discussion']
        self.df = None

    def get_bsobj(self, url):
        html = urlopen(url).read().decode('utf-8')
        bsobj = BeautifulSoup(html, 'lxml')
        return bsobj
```

数据预处理

■ Python Scrapy案例（豆瓣Top250电影）

2.解析并获取目标对象：

```
def get_info(self):
    for url in self.top_urls:
        bsobj = self.get_bsobj(url)
        main = bsobj.find('ol', {'class': 'grid_view'})

        # 标题及链接信息
        title_objs = main.findAll('div', {'class': 'hd'})
        titles = [i.find('span').text for i in title_objs]
        links = [i.find('a')['href'] for i in title_objs]

        # 评分信息
        score_objs = main.findAll('div', {'class': 'star'})
        scores = [i.find('span', {'class': 'rating_num'}).text for i in score_objs]
        score_cnts = [i.findAll('span')[-1].text for i in score_objs]

        for title, link, score, score_cnt in zip(titles, links, scores, score_cnts):
            self.data[title].extend([title, link, score, score_cnt])
            bsobj_more = self.get_bsobj(link)
            more_data = self.get_more_info(bsobj_more)
            self.data[title].extend(more_data)
            print(self.data[title])
            print(len(self.data))
            time.sleep(1)
```

数据预处理

■ Python Scrapy案例（豆瓣Top250电影）

3.保存为csv格式文件：

```
def dump_data(self):
    data = []
    for title, value in self.data.items():
        data.append(value)
    self.df = pd.DataFrame(data, columns=self.columns)
    self.df.to_csv('douban_top250.csv', index=False)
```

4. 运行执行函数：

```
if __name__ == '__main__':
    douban = DoubanMovieTop()
    douban.get_info()
    douban.dump_data()
```

数据预处理

■ Python Scrapy案例（豆瓣Top250电影）

5. csv保存的结果：

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	中文名	外文名	电影链接	图片链接	评分	评价人数	概评	概述									
2	肖申克的救	TheShaws	https://mov	https://img	9.7	2271492	希望让人自	导演:弗兰克·德拉邦特FrankDarabont主演:蒂姆·罗宾斯TimRobbins...1994美国犯罪剧情									
3	霸王别姬		https://mov	https://img	9.6	1686271	风华绝代。	导演:陈凯歌KaigeChen主演:张国荣LeslieCheung张丰毅FengyiZha...1993中国大陆中国香港剧情爱情同性									
4	阿甘正传	ForrestGu	https://mov	https://img	9.5	1710247	一部美国近	导演:罗伯特·泽米吉斯RobertZemeckis主演:汤姆·汉克斯TomHanks...1994美国剧情爱情									
5	这个杀手不	Léon	https://mov	https://img	9.4	1893186	怪蜀黍和小	导演:吕克·贝松LucBesson主演:让·雷诺JeanReno娜塔莉·波特曼...1994法国美国剧情动作犯罪									
6	泰坦尼克号	Titanic	https://mov	https://img	9.4	1669190	失去的才是	导演:詹姆斯·卡梅隆JamesCameron主演:莱昂纳多·迪卡普里奥Leonardo...1997美国剧情爱情灾难									
7	美丽人生	Lavitaèbell	https://mov	https://img	9.5	1056697	最美的谎言	导演:罗伯托·贝尼尼RobertoBenigni主演:罗伯托·贝尼尼RobertoBeni...1997意大利剧情喜剧爱情战争									
8	千与千寻	千と千尋の	https://mov	https://img	9.4	1787954	最好的宫崎	导演:宫崎骏HayaoMiyazaki主演:柊瑠美RumiHîragi入野自由Miy...2001日本剧情动画奇幻									
9	辛德勒的名	Schindler's	https://mov	https://img	9.5	873227	拯救一个人	导演:史蒂文·斯皮尔伯格StevenSpielberg主演:连姆·尼森LiamNeeson...1993美国剧情历史战争									
10	盗梦空间	Inception	https://mov	https://img	9.3	1658977	诺兰给了我	导演:克里斯托弗·诺兰ChristopherNolan主演:莱昂纳多·迪卡普里奥Le...2010美国英国剧情科幻悬疑冒险									
11	忠犬八公的	Hachi:ADo	https://mov	https://img	9.4	1135721	永远都不自	导演:莱塞·霍尔斯道姆LasseHallström主演:理查·基尔RichardGer...2009美国英国剧情									
12	星际穿越	Interstellar	https://mov	https://img	9.3	1331082	爱是一种力	导演:克里斯托弗·诺兰ChristopherNolan主演:马修·麦康纳MatthewMc...2014美国英国加拿大剧情科幻冒险									
13	海上钢琴师	Laleggend	https://mov	https://img	9.3	1349887	每个人都是	导演:朱塞佩·托纳多雷GiuseppeTornatore主演:蒂姆·罗斯TimRoth...1998意大利剧情音乐									
14	楚门的世界	TheTruma	https://mov	https://img	9.3	1244054	如果再也不	导演:彼得·威尔PeterWeir主演:金·凯瑞JimCarrey劳拉·琳妮Lau...1998美国剧情科幻									
15	三傻大闹宝	3Idiots	https://mov	https://img	9.2	1512164	英俊版憨豆	导演:拉库马·希拉尼RajkumarHirani主演:阿米尔·汗AamirKhan卡...2009印度剧情喜剧爱情歌舞									
16	机器人总动	WALL·E	https://mov	https://img	9.3	1066994	小瓦力，大	导演:安德鲁·斯坦顿AndrewStanton主演:本·贝尔特BenBurtt艾丽...2008美国科幻动画冒险									
17	放牛班的春	Leschorist	https://mov	https://img	9.3	1049666	天籁一般的	导演:克里斯托夫·巴拉蒂ChristopheBarratier主演:热拉尔·朱尼奥Gé...2004法国瑞士德国剧情音乐									
18	大话西游之	西遊記大結	https://mov	https://img	9.2	1216024	一生所爱。	导演:刘镇伟JeffreyLau主演:周星驰StephenChow吴孟达ManTatNg...1995中国香港中国大陆喜剧爱情奇幻古装									
19	疯狂动物城	Zootopia	https://mov	https://img	9.2	1471339	迪士尼给我	导演:拜伦·霍华德ByronHoward瑞奇·摩尔RichMoore主演:金妮弗·...2016美国喜剧动画冒险									
20	无间道	無間道	https://mov	https://img	9.2	1006599	香港电影史	导演:刘伟强麦兆辉主演:刘德华梁朝伟黄秋生2002中国香港剧情犯罪悬疑									
21	熔炉	도가니	https://mov	https://img	9.3	742870	我们一路看	导演:黄东赫Dong-hyukHwang主演:孔侑YooGong郑有美Yu-miJung...2011韩国剧情									
22	教父	TheGodfat	https://mov	https://img	9.3	742747	千万不要说	导演:弗朗西斯·福特·科波拉FrancisFordCoppola主演:马龙·白兰度M...1972美国剧情犯罪									
23	当幸福来敲	ThePursuit	https://mov	https://img	9.1	1216322	平民励志片	导演:加布里尔·穆奇诺GabrieleMuccino主演:威尔·史密斯WillSmith...2006美国剧情传记家庭									
24	龙猫	となりの	https://mov	https://img	9.2	1014045	人人心中者	导演:宫崎骏HayaoMiyazaki主演:日高法子NorikoHidaka坂本千夏Ch...1988日本动画奇幻冒险									

数据预处理

■ Python Jieba分词：

jieba是目前最好的 Python 中文分词组件，它主要有以下 3 种特性：
1.支持 3 种分词模式：精确模式、全模式、搜索引擎模式
2.支持繁体分词
3.支持自定义词典

1. 安装
pip install jieba

2. 导入 jieba
import jieba
import jieba.posseg as pseg #词性标注
import jieba.analyse as anls #关键词提取

数据预处理

■ Python Jieba分词

3.分词操作

```
import jieba
#全模式
seg_list = jieba.cut("他来到上海交通大学", cut_all=True)
print("【全模式】： " + "/ ".join(seg_list))
```

Out:

【全模式】： 他/ 来到/ 上海/ 上海交通大学/ 交通/ 大学

```
#精确模式
seg_list = jieba.cut("他来到上海交通大学", cut_all=False)
print("【精确模式】： " + "/ ".join(seg_list))
```

Out:

【精确模式】： 他/ 来到/ 上海交通大学

数据预处理

■ Python Jieba分词

4.去除停用词

1	$	32	一	76	与其
2	-	33	一些	77	与其说
3	(	34	一何	78	与否
4	)	35	一切	79	与此同时
5	(	36	一则	80	且
6	)	37	一方面	81	且不说
7	;	38	一旦	82	且说
8	/	39	一来	83	两者
9	+	40	一样	84	个
10	?	41	一般	85	个别
11		42	一转眼	86	临
12	“	43	万一	87	为
13	”	44	上	88	为了
14	、	45	上下	89	为什么
15	。	46	下	90	为何
16	《	47	不	91	为止
17	》	48	不仅	92	为此
18	段	49	不但	93	为着
19	粒	50	不光	94	乃
20	克	51	不单	95	乃至
21	盒	52	不只	96	乃至于
22	袋	53	不外乎	97	么
23	瓶	54	不如	98	之
24	月	55	不妨	99	之一
25	岁	56	不尽	100	之所以
26	罐	57	不尽然	101	之类
27	支	58	不得	102	乌乎
28	片	59	不怕	103	乎
29	mg	60	不惟	104	乘
30	ml	61	不成	105	也
31	g	62	不拘	106	也好
		63	不料	107	也罢
		64	不是	108	了
		65	不比	109	二来
		66	不然	110	于
		67	不特		
		68	不独		

```python
import numpy as np
import pandas as pd
import re,os,jieba
import csv
"""第一步：用正则表达式清洗数据，并去除停用词"""
df = pd.read_csv("item_itemid.csv")
df = df[['ID','商品']].dropna()#删除缺失数据
def stopwordslist():
    stopwords = [line.strip() for line in open('./cn_stopwords.txt',encoding='UTF-8').readlines()]
    return stopwords
def seg_depart(sentence):
    sentence_depart = jieba.cut(sentence.strip())
    stopwords = stopwordslist()
    outstr = ''
    for word in sentence_depart:
        if word not in stopwords:
            outstr += word
            outstr += "|"
    return outstr.strip('|')
#docs_text = df['商品']
def mian():
    outfilename = "fenci results.csv"
    f = open(outfilename, 'w',  newline='',encoding='utf-8-sig')
    csv_writer = csv.writer(f)
    csv_writer.writerow(['ID','Item','Item_fenci'])
    for IDX,Values in df.iterrows():
        ID=Values['ID']
        print('ID:',ID)
        Item=Values['商品']
        print('Item:',Item)
        fenci = re.sub('片片','片',str(Item))
        fenci = re.sub(r'[^\u4e00-\u9fa5]+','',fenci)
        fenci_seg = seg_depart(fenci.strip())
        print('分词后结果:',fenci_seg)
        csv_writer.writerow([str(ID),str(Item),str(fenci_seg)])
    f.close()
    print("删除停用词,并分词成功！")
mian()
```

数据预处理

■ Python Jieba分词

4. 分词结果

	ID	Item	Item_fenci
2	0	HiPP喜宝有机Combiotic较大婴儿配方奶粉2段	喜宝|有机|较大|婴儿|配方|奶粉
3	1	HiPP喜宝有机Combiotic婴儿配方奶粉1段	喜宝|有机|婴儿|配方|奶粉
4	2	HiPP喜宝有机Combiotic幼儿配方奶粉3段	喜宝|有机|幼儿|配方|奶粉
5	3	HiPP喜宝有机婴幼儿5种谷物粉200克/盒	喜宝|有机|婴幼儿|种|谷物|粉克盒
6	4	喜宝有机婴幼儿精选小米粉350g/盒	喜宝|有机|婴幼儿|精选|小米|粉盒
7	5	HiPP喜宝有机婴幼儿大米粉200克/盒	喜宝|有机|婴幼儿|米粉|克盒
8	6	德国HiPP喜宝益生元婴幼儿奶粉2+段	德国|喜宝益|生元|婴幼儿|奶粉
9	7	德国HiPP喜宝益生元婴幼儿奶粉2段	德国|喜宝益|生元|婴幼儿|奶粉
10	8	德国HiPP喜宝益生元婴幼儿乳奶粉1+段	德国|喜宝益|生元|婴幼儿|乳|奶粉
11	9	喜宝吸吸乐苹果草莓香蕉口味100g/袋	喜宝|吸吸|乐|苹果|草莓|香蕉|口味
12	10	喜宝吸吸乐梨香蕉猕猴桃口味100g/袋	喜宝|吸吸|乐梨|香蕉|猕猴桃|口味
13	11	喜宝吸吸乐苹果梨香蕉口味100g/袋	喜宝|吸吸|乐|苹果梨|香蕉|口味
14	12	喜宝吸吸乐苹果桃子莓果口味100g/袋	喜宝|吸吸|乐|苹果|桃子|莓果|口味
15	13	喜宝吸吸乐苹果芒果桃子口味100g/袋	喜宝|吸吸|乐|苹果|芒果|桃子|口味
16	14	喜宝吸吸乐梨苹果芒果百香果口味100g/袋	喜宝|吸吸|乐梨|苹果|芒果|百香果|口味
17	15	喜宝有机奶粉2段800克	喜宝|有机|奶粉|段克
18	16	德国HiPP喜宝有机奶粉3段_800g	德国|喜宝|有机|奶粉
19	17	Purtier月见草核苷酸胶囊60粒/瓶	月见草|核苷酸|胶囊|粒瓶
20	18	SoriaNatural森力亚天然铁源素口服液250ml/瓶	森力亚|天然|铁源素|口服液
21	19	SoriaNatural森力亚皇金肝口服液	森力亚|皇金肝|口服液
22	20	Swissekids儿童脑部发育益智DHA软胶囊30粒	儿童|脑部|发育|益智|软胶囊
23	21	SwisseUltiboost补铁片30片	补铁|片片
24	22	Swisse蔓越莓毛孔修复面膜70G	蔓越莓|毛孔|修复|面膜
25	23	Swisse黄瓜卸妆液300ml	黄瓜|卸妆液
26	24	Swisse摩洛哥坚果抗老化眼霜15ml	摩洛哥|坚果|抗老化|眼霜
27	25	胶原蛋白加透明质酸片	胶原蛋白|加|透明质|酸片
28	26	Swisse睡眠改善片100片	睡眠|改善|片片
29	27	Swisse深海鱼油软胶囊无腥味1500mg400粒新版	深海鱼|油|软胶囊|腥味|新版
30	28	Swisse维C泡腾片60片新	维|泡腾片|片新
31	29	Swisse钙+维生素D片150片	钙|维生素|片片
32	30	Swisse葡萄糖胺片180片	葡萄糖|胺|片片
33	31	Swisse婴幼儿DHA+EPA鱼油软胶囊新版60粒	婴幼儿|鱼油|软胶囊|新版
34	32	Swisse婴幼儿柠檬酸钙D软胶囊60粒	婴幼儿|柠檬酸|钙|软胶囊
35	33	Swisse奶蓟草肝脏排毒片120片	奶蓟草|肝脏|排毒|片片
36	34	Swisse儿童骨骼成长咀嚼片50片	儿童|骨骼|成长|咀嚼|片片
37	35	Swisse高强度蜂胶胶囊210粒	高强度|蜂胶|胶囊
38	36	Swisse胶原蛋白液500ml	胶原蛋白|液
39	37	Swisse清肺片90片	清肺|片片
40	38	Swisse高强度蔓越莓胶囊25000mg30粒	高强度|蔓越莓|胶囊

数据预处理

■ Python Jieba分词

5. LDA

LDA（Latent Dirichlet Allo
叶斯概率模型，包含词、主
为一篇文章的每个词都是通
一定概率选择某个词语”这
到词服从多项式分布。

层贝
门认
中以
主题

```
def LDA():
    allwords = []  # 所有的词汇
    train = []  # 所有样本
    f = open('./fenci results.csv', 'r', encoding='utf-8-sig')
    reader = csv.reader(f)
    i=1
    for row in reader:
        if i>1:
            fenci=row[2]
            wordslist=fenci.split('|')
            train.append(wordslist)
            for word in wordslist:
                allwords.append(word)
        i=i+1
    dictionary = corpora.Dictionary(train)#"""构建词频矩阵，训练LDA模型"""
    corpus = [dictionary.doc2bow(text) for text in train]
    tfidf = models.TfidfModel(corpus)  # 统计tfidf
    corpustfidf = tfidf[corpus]  # 得到每个文本的tfidf向量，稀疏矩阵
    K=6#主题数量
    lda = LdaModel(corpus=corpustfidf, id2word=dictionary,minimum_probability=pow(0.1,1000),
                   num_topics=K,alpha='auto',eta='auto',iterations=10000, gamma_threshold=0.0001, random_state=0)
    topic_list = lda.print_topics(num_topics=K, num_words=10)
    print("主题的单词分布为：\n")
```

LDA主题的单词分布为：

```
(0, '0.025*"HiPP" + 0.022*"有机" + 0.022*"喜宝" + 0.019*"婴幼儿" + 0.019*"奶粉" + 0.014*"德国" + 0.014*"Combiotic" + 0.013*"益生元" + 0.012*"配方" + 0.011*"维生素"')
(1, '0.014*"蓝色" + 0.013*"德国" + 0.011*"Bambinchen" + 0.011*"星球" + 0.011*"400g" + 0.011*"羊奶粉" + 0.010*"液" + 0.010*"面膜" + 0.010*"Swisse" + 0.009*"250ml"')
(2, '0.020*"Swisse" + 0.013*"胶囊" + 0.013*"维C" + 0.012*"维生素" + 0.012*"泡腾片" + 0.011*"纸尿裤" + 0.011*"铂金" + 0.011*"装" + 0.010*"清肺" + 0.010*"Huggies"')
(3, '0.020*"雅培" + 0.020*"奶粉" + 0.017*"幼儿" + 0.016*"配方" + 0.016*"心美力" + 0.013*"Ultiboost" + 0.011*"补铁" + 0.011*"婴儿" + 0.011*"喜力特" + 0.011*"有机"')
(4, '0.020*"乳蛋白" + 0.019*"部分" + 0.018*"水解" + 0.016*"港版" + 0.015*"惠氏" + 0.015*"BABYNES" + 0.015*"配方" + 0.014*"婴儿" + 0.012*"奶粉" + 0.011*"儿童"')
(5, '0.017*"配方" + 0.016*"吸吸乐" + 0.016*"100g" + 0.015*"奶粉" + 0.015*"口味" + 0.014*"婴儿" + 0.013*"部分" + 0.013*"水解" + 0.013*"咀嚼片" + 0.012*"复合"')
```

Blei D M, Ng A Y, Jordan M I. Latent dirichlet allocation[J]. Journal of machine Learning research, 2003, 3(Jan): 993-1022.

数据预处理

■ **Python Jieba分词**

4.分词后的关键词云图

词权重: TF-IDF, LDA

```
import genism
from gensim.models import LdaModel
from wordcloud import WordCloud
```

作业

电子商务数据分析课程作业1

1. 电子商务名词解释：B2B，B2C，C2B，C2C，O2O，并结合国内外知名电子商务平台案例进行说明。
2. 分析大数据6V的特点，如果是参考已有的文章或者网址，请在参考文献部分标出。
3. 常用的网络爬虫工具，以及各个工具的特点分析。