首都经济贸易大学
CAPITAL UNIVERSITY OF ECONOMICS AND BUSINESS

本科生实习（社会调查）报告

√认知实习　　□专业实习　　□毕业实习

学　　院　　信息学院

专　　业　　计算机科学与技术(互联网+)

班　　级　　计算机一班

学　　号　　32016070129

姓　　名　　李杰

指导教师　　覃爱明，王汀，申蔚

2018 年 07 月 20 日

实　习 时　间 社会调查	2018 年 07 月 02 日 至 2018 年 07 月 20 日		
实　习 单位名称 社会调查	首都经济贸易大学信息学院	学院实习基地	是： 否：√

**实　习
单位简介
社会调查**

首都经济贸易大学创建于 1956 年，是由原北京经济学院和原北京财贸学院于 1995 年 3 月合并、组建的北京市属重点大学。60 余年来，学校已发展为拥有经济学、管理学、法学、文学、理学和工学等六大学科，以经济学、管理学为重要特色和突出优势，各学科相互支撑、协调发展的现代化、多科性财经类大学。

首都经济贸易大学特大城市经济社会发展研究院为首批首都高端智库试点单位。学校拥有北京市哲学社会科学 CBD 发展研究基地、北京市经济社会发展政策研究基地、国家税收法律研究基地和城市群系统演化与可持续决策北京市重点实验室等市级研究机构，以及人口经济研究所、首都经济研究所等 30 个校级研究机构。学校主办的《经济与管理研究》是 CSSCI 来源期刊、全国中文核心期刊、中国人文社会科学核心期刊、RCCSE 核心期刊。《人口与经济》是我国最早创刊的人口学类期刊之一，是 CSSCI 来源期刊、全国中文核心期刊、中国人文社会科学核心期刊、国家社会科学基金资助的 200 个重要期刊之一。

学校与 33 个国家和地区的 132 所大学、研究机构、社会团体等有学术交流与合作往来。学校自 1986 年开始招收留学生，现已发展形成多层次、多科性的国际人才培养体系，学生类别包含博士研究生、硕士研究生、本科生、高级进修生、普通进修生、语言生和各类短期生等。学校于 2007 年开办全英文授课的硕士班，2011 年开办全英文授课的博士班。目前有来自 79 个国家的留学生在校就读，2017 年达 910 人。

学校将继续坚持“立足北京、服务首都、面向全国、走向世界”，秉承“崇德尚能，经世济民”的校训，以培养适应当代经济和社会发展需要、德智体全面发展、理论基础扎实、知识面较宽、富有创新精神和实践能力的高素质应用型人才为目标，朝着建设“现代化、国际化、多科性、有特色的国内一流、国际知名财经大学”的目标开拓奋进。

实　　习 社会调查 记录及所在单位评语				
实习记录 1.Python 基本语法（Python 的简介、安装、数据类型、数据结构、运算符、异常处理、函数、生成器与迭代器、面向对象-类、文件读写、Numpy & Pandas 常见语法、Matplotlib & Pandas 中的绘图函数） 2.Python 数据分析（Requests 库、Xpath、正则表达式、Scrapy 框架、Python 操作 MySQL、Scrapy 爬虫实战、数据探索及数据预处理常用函数） 3.Python 数据实战（RFM 模型、聚类分析原理、RFM 项目实战数据预处理、K-Means 聚类分析、分析实战、时间序列分析原理、数据探索及预处理、利用 Python 进行时间序列分析）				
出勤(天)	缺　　勤　(天)			
	事 假	病 假	旷 工	合　计
表现评语： 实习单位盖章： 负责人签字： 年　　月　　日				

实习（社会调查）报告

于 2018 年 7 月 2 日，接受学校安排的在校认知实习。学习时间为 3 周，主要学习当前最火的 Python 语言。要求基本了解掌握 Python 的基本语法，以及 Python 在数据方面的处理，分析，实战；还包括网络数据的爬虫。主讲老师在 Python 语言教学方面很有经验，主要以学生亲自动手实战为主，真正营造以学生为中心、老师为辅助的学习环境。并且在各位辅导老师孜孜不倦的教导下，达到培养我们学生自己主动学习，自主解决问题难题的目的。学习期间主要分为四个阶段。

一、入门阶段

1.1 Python 简介

老师开始主要介绍了 Python 语言在当下的广泛应用。Python 是一门相当高级的语言。其设计之初的定位是“优雅”、“明确”、“简单”，所以 Python 程序看上去总是简单易懂，并且对于同一件事的处理代码量相对于 C 语言及 Java 语言等等要少很多。在日常任务、做网站、做网络游戏的后台、网络应用、后台服务等方面 Python 已经很流行；特别是在当下的大数据处理及人工智能方面，Python 有很大的优势。在最近几年 Python 的前景排名及应用也一直在上升。如图-1-2 TIOBE 排行榜。

May 2018	May 2017	Change	Programming Language	Ratings	Change
1	1		Java	16.380%	+1.74%
2	2		C	14.000%	+7.00%
3	3		C++	7.668%	+2.92%
4	4		Python	5.192%	+1.64%
5	5		C#	4.402%	+0.95%
6	6		Visual Basic .NET	4.124%	+0.73%
7	9	^	PHP	3.321%	+0.63%
8	7	v	JavaScript	2.923%	-0.15%
9	-	^^	SQL	1.987%	+1.99%
10	11	^	Ruby	1.182%	-1.25%
11	14	^	R	1.180%	-1.01%
12	18	^^	Delphi/Object Pascal	1.012%	-1.03%
13	8	vv	Assembly language	0.998%	-1.86%
14	16	^	Go	0.970%	-1.11%
15	15		Objective-C	0.939%	-1.16%
16	17	^	MATLAB	0.929%	-1.13%
17	12	vv	Visual Basic	0.915%	-1.43%
18	10	vv	Perl	0.909%	-1.69%
19	13	vv	Swift	0.907%	-1.37%
20	31	^^	Scala	0.900%	+0.18%

图-1.近两年编程语言排名

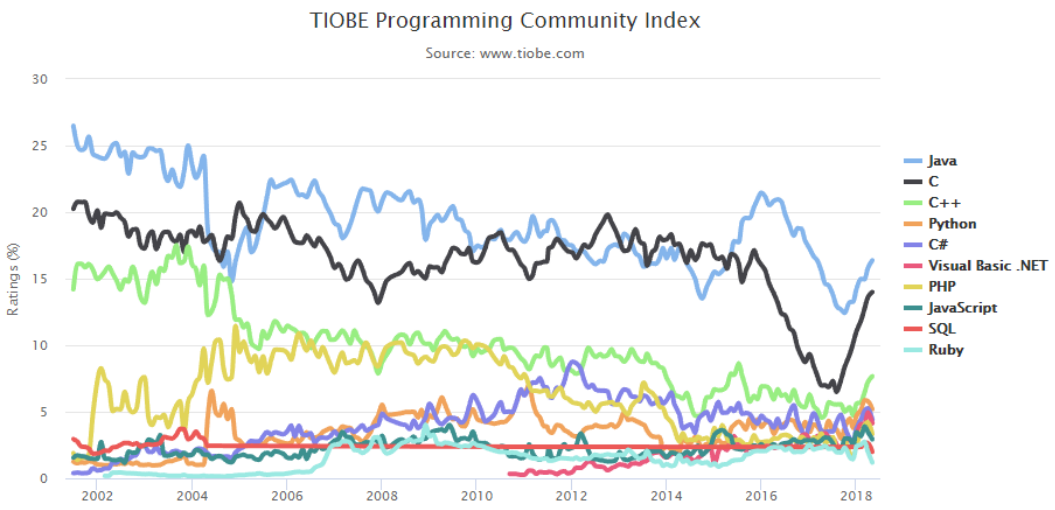

图-2. Python 应用趋势

1.2 安装及基本语法

主要学习对 Python3.6 的安装，基本语法，类-对象，Python 的一些基本类库等等的讲解。中间遇到了很多问题，在各位老师及同学相互之间的帮助下，几乎所有的同学都对 Python 有了新的了解。

然后是 Python 的数据类型，数据结构。我觉得数据结构是 Python 的基础，其组成了 Python 的基本骨架。尤为重要。主要内容及功能如表-1。

	列表	元组	字典	集合
简要概念	可变的序列，数据项不需要相同。	与列表类似，不同于元组的元素不能修改。	键必须是唯一的，但值则不必。	集合（set）是一个无序不重复元素的序列。
创建	方括号括起来，逗号分隔不同的数据项。	小括号中添加元素，逗号分隔不同的数据项。	每个键值对用冒号分割，每个对之间用逗号分割，整个字典包括在花括号中。	可以使用大括号 { } 或者 set() 函数创建集合。
举例	list1=['Google','A',1997] list2 = [1, 2, 'B' , 4,]	tup1=('Google','A',1997) tup2 = (1, 2, 'A', 4)	d={'A':'2','B':'3'}	basket={'apple','orange'}
访问	print(list1[0]) print(list2[1:3])	print(tup1[0]) print(tup2[1:3])	print(d['A'])	print(basket) 'orange' in basket
简要方法	list2.remove(1) list1.append(1998)	len(tup1) tuple(list1)	len(d) str(d)	s.add(x) s.remove(x)

表-1. 数据结构

二、数据处理

第二阶段主要是学习 Python 在网络数据爬虫及处理的提高。学习了几个很重要的库的用法及应用实战。

2.1 NumPy 库

这是 Python 语言的一个扩充程序库。支持大量高级的维度数组与矩阵运算，此外也针对数组运算提供大量的数学函数库，并且由于其在运算效率方面极好,是大量机器学习框架的基础库。其中数组与标量间的运算，基本的索引与切片，花式索引-fancy indexing 等等是我们需要掌握的。

2.2 Pandas 库

pandas 是基于 NumPy 的一种工具,该工具是为了解决数据分析任务而创建的。Pandas 纳入了大量类库和一些标准的数据模型，提供了高效地操作大型数据集所需的工具；而其中主要的两个数据结构：Series 和 DataFrame 是我们重点学习的对象。

2.2.1 Series 与 DataFrame

Series 是一个一维的类似数组的对象，它包含一个数组数据（任何 numpy 数据类型）和一个与数组关联的索引。为了方便理解，我们可以把 Series 看着是一个有序字典。其中索引是连续的，从 0 开始。而一个 DataFrame 表示一个表格，它包含一个经过排序的列表集。每一个列表都可以有不同的类型值（数字，字符串，布尔等等）。DataFrame 有行和列的索引；它可以被看作是一个 Series 的字典。Series 与 DataFrame 的简单创建如图-3-4。

```
#唯一值
from pandas import Series
o = Series([1,2,3,2,2,3])
o
```

```
0    1
1    2
2    3
3    2
4    2
5    3
dtype: int64
```

图-3. Series 创建

```
import pandas as pd
import numpy as np
from pandas import DataFrame
```

```
data = np.arange(16)
print(data)
```

```
[ 0  1  2  3  4  5  6  7  8  9 10 11 12 13 14 15]
```

```
data = data.reshape((4,4))
df = DataFrame(data=data,index=['a','b','c','d'],columns=['one','two','three','four'])
df
```

	one	two	three	four
a	0	1	2	3
b	4	5	6	7
c	8	9	10	11
d	12	13	14	15

图-4. DataFrame 创建

2.3 Requests 库与网络请求

紧接着还学习了 Requests 库请求网络，学习了一些简要的网络知识，如反爬虫策略中设置浏览器代理，IP 代理，模拟登录等等；结合之前的学习，不仅加深了印象，而且还学会合理运用、实战处理，并且也了解到异步加载的解决方法；还有 pandas 保存数据到 Excel，文件数据的读取及转换；在此之间学习了 xpath、正则表达式 re 爬取网页数据匹配网页字符串。正则表达式是一个很好的工具，不仅在 Python 中，在其他编程语言中也很重要并且有广泛的运用。在 Python 中，xpath 及 re 正则表达式的简要用法如表-2 所示。

2.4 绘图及数据处理

第二阶段的最后主要还学习了 Matplotlib 及 Pandas 中的绘图函数，基于 DataFrame 的数据处理及对数据的导入与保存；还了解 Scrapy 库的一些简要原理，并且学习根据数据绘制折线图，柱形图，饼图，散点图等等。还做了一些在绘图上的提升，数据探索及数据预处理常用函数分析等等。

	Xpath		正则表达式	
导入包	From lxml import etree import requests		import re import requests	
基本语法	/	从根节点选取	search() match()	从文本中查询匹配
	//	从匹配选择的当前节点选择文档,不考虑位置	.	匹配除换行符以外的任何字符
	@	选取属性	*	匹配 0/1/多次字符
	*	匹配任何元素节点	?	匹配 0/1 次字符
	nodename	选取此节点所有子节点。	()	模式匹配--即想要的内容
	node()	匹配任何类型的节点	strip()	去除空格
示例	1.xpath('//span[@class="el"]/a/text()') 2.xpath('//div[@class="el"]/a/@title')		1.p1 = "t.*d" 2.positionpat='title="(.*)"href=".*"'	

表-2 xpath 与 re

三、爬取网页数据

本阶段学习 Python 操作 MySQL 数据库、Scrapy 爬虫实战。

3.1 Scrapy 框架

Scrapy 是一个为了爬取网站数据，提取结构性数据而编写的应用框架。可以应用在包括数据挖掘，信息处理或存储历史数据等一系列的程序中。此阶段相当于一个小的总结，结合之前所学的知识，结合 MySQL 以一个实战爬虫讲解为主。此处将以爬取网易云音乐其中歌单的所有信息并对数据作简要分析为主。

3.2 爬取网易云歌单

爬取的信息主要为 15 页的歌单的创建者，歌单描述，收藏量，转发量，评论量，标签信息及图片地址，信息主要来自两个完全不同的网页，异步加载数据并实时跳转到详细页，然后对爬取到的信息数据以标签为主分析其相关性。歌单首页信息及详细信息如图-5-6 所示。

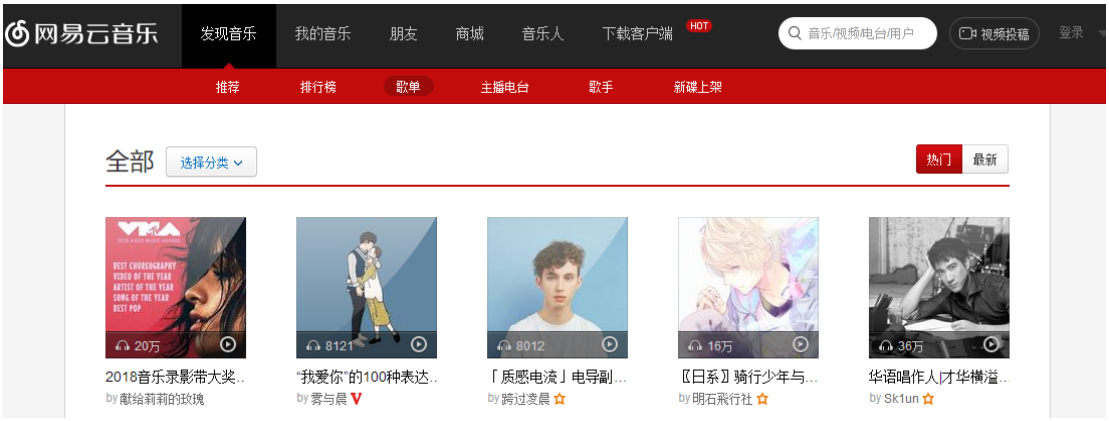

图-5 网易云歌单首页

图-6 网易云歌单详细页

3.2.1 创建 Item 及 Spider

在创建 Scrapy 项目之后定义 Item。当非结构性的数据源提取结构性数据，例如网页，Scrapy 提供 Item 类来满足这样的需求。Item 对象是种简单的容器，保存了爬取到得数据。其提供了类似于词典的 API 以及用于声明可用字段的简单语法。Item 使用简单的 class 定义语法以及 Field() 对象来声明。

然后创建一个 Spider。Spider 是用户编写用于从单个网站(或者一些网站)爬取数据的类。其包含了一个用于下载的初始 URL，如何跟进网页中的链接以及如何分析页面中的内容，提取生成 item 的方法。在 Spider 中初始 URL 即网易云歌单首页，新建 item 对象，在 parse()方法里运用 xpath 及 re 正则表达式匹配上述想要的信息。

因为网易云音乐实时更新并且爬取的是两个完全不同网页，所以此处运用到异步加载及循环获取每个歌单详细链接地址再异步加载获取信息的知识，同时也学习了 Python 字符串与其数据结构之间转换的知识。

3.2.2 Scrapy 的中间框架

为了运用到反爬虫策略浏览器代理知识，需要用到 Middlewares 类组件。Middlewares 中间件是介入到 Scrapy 的 spider 处理机制的钩子框架，可以添加代码来处理发送给 Spiders 的 response 及 spider 产生的 item 和 request。所以在项目下的 Middlewares 里定义浏览器代理，并在其方法 process_request()里面随机调用浏览器。

为了将数据导出或保存到 MySQ 数据库，又用到 Pipeline 类组件。当 Item 在 Spider 中被收集之后，它将会被传递到 Item Pipeline，一些组件会按照要求执行对 Item 的处理。所以此时我在 Pipeline 类组件里定义连接 MySQL 数据库并创建 MySQL 的数据库，遍历 Item 将每条数据插入到数据库并导出为.csv 文件。此处运用到 Python 操作 MySQL 数据库，Scrapy 将数据导出的知识。

最后为了使 Middlewares 中间件及 Pipeline 类组件被调用，需要在项目下的 setting 里将 DOWNLOADER_MIDDLEWARES 及 ITEM_PIPELINES 注解去掉。用 scrapy crawl name 运行程序。如图-7 为爬取的前 10 条数据。

id	createBy	descs	addNum	transNum	commNum	tag	imgPath
1	夕阳下的盛宴	印象主义文艺思潮视域下音乐美学思想	764	6	13	古典器乐经典	http://p1.r
2	礁池	人间惆怅•曾有青春年少时	667	6	20	华语摇滚感动	http://p1.r
3	Spyromun_	Sweet Time·在耳朵里融化的电音糖	499	9	5	欧美电子清新	http://p1.r
4	凉予	人生这么短，请你把时间浪费在我身上。	766	6	21	华语流行浪漫	http://p1.r
5	EricCheng-	斯人若彩虹 遇上方知有	844	5	19	欧美清新治愈	http://p1.r
6	YoakeNatsu	僕の夏日集/// 蝉声轻掠 风拂浓荫 是夏天啊	299	9	6	轻音乐日语	http://p1.r
7	辞雀	『古风』催泪剪辑常用BGM	58885	7	3284	华语流行古风	http://p1.r
8	风若雨寒	「日系」冷门良曲|在时光里等你牵我手	8296	3	47	日语流行	http://p1.r
9	云音乐VIP	有风吹过的地方就有他的声音回荡	4287	8	146	华语流行经典	http://p1.r
10	鹿白川	论如何优雅的做个不动声色的大人	93906	4	952	欧美安静治愈	http://p1.r

图-7 网易云信息

3.3 数据简要分析

对爬取到的 672 条数据进行简单分析处理。首先基于标签首个关键字作 describe()处理，得出如图-8 所示。

	收藏量	转发量	评论量
count	672.000000	672.000000	672.000000
mean	20990.325893	4.366071	259.602679
std	33082.507362	2.824722	700.709263
min	98.000000	0.000000	1.000000
25%	2607.250000	2.000000	34.000000
50%	9496.500000	4.000000	90.000000
75%	24037.000000	7.000000	263.500000
max	394330.000000	9.000000	10289.000000

图-8

标签	收藏量	转发量	评论量	标签2	收藏量2	转发量2	评论量2
华语	7040336	902	104552	快乐	18012	1	200
欧美	2572639	600	23415	安静	17596	7	128
日语	1308391	305	11081	无	16572	5	320
电子	641522	170	6120	旅行	11468	9	637
流行	584551	178	9571	小语	9408	30	268
轻音	558206	167	3755	后摇	9074	10	271
影视	301084	51	1610	夜晚	7576	17	136
韩语	180332	149	3037	R&	6759	5	144
治愈	146336	3	2608	钢琴	6476	7	69
古典	124680	33	741	运动	4842	20	160
民谣	109066	56	656	雷鬼	4698	10	65
兴奋	89085	2	404	放松	4111	3	21
摇滚	78523	39	979	孤独	4040	9	64
粤语	58380	24	461	校园	3508	1	49
游戏	45670	18	1384	浪漫	3491	26	162
世界	38324	8	590	驾车	1814	0	60
古风	30725	5	165	舞曲	1774	0	8
说唱	23567	28	139	英伦	792	7	8
清新	21441	18	187	感动	571	0	21
另类	19740	10	189	AC	319	1	18

图-9 关键字排名

再根据标签以收藏量，转发量，评价量为关键字作降序处理，得到如图-9 所示。可得出以华语为关键字的音乐是最受欢迎的，其次欧美，电子，影视类的音乐也很受欢迎；相对于舞曲，英伦类的音乐欢迎度较小。然后在 672 条数据里以标签出现次数作简要处理，得出前 10 如图-10，所占百分比如图-11。由图也可得出华语受欢迎的结论。

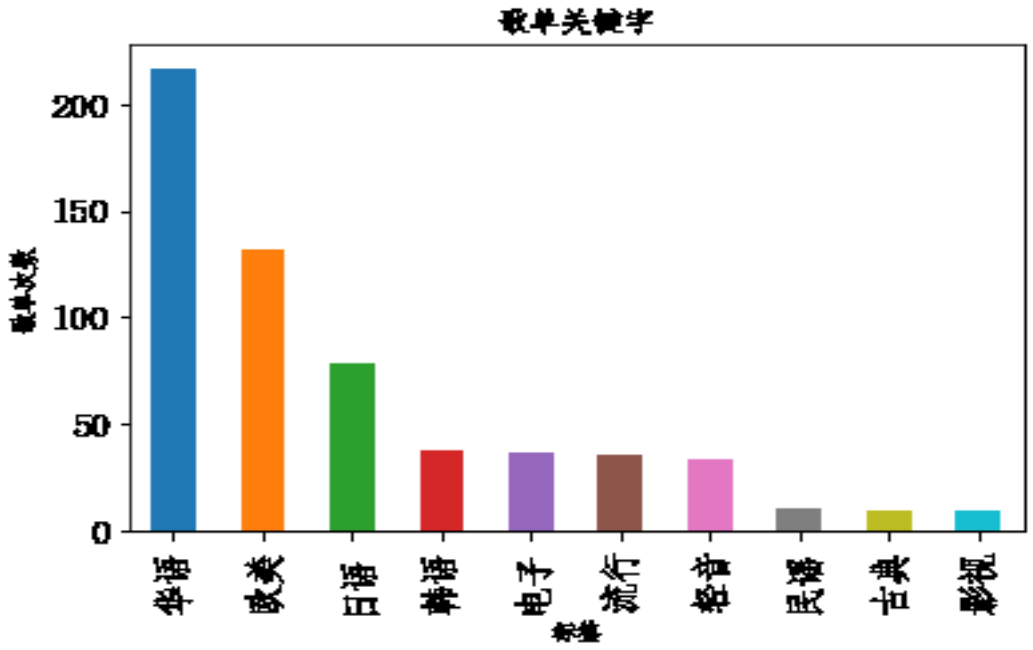

图-10 关键字前 10

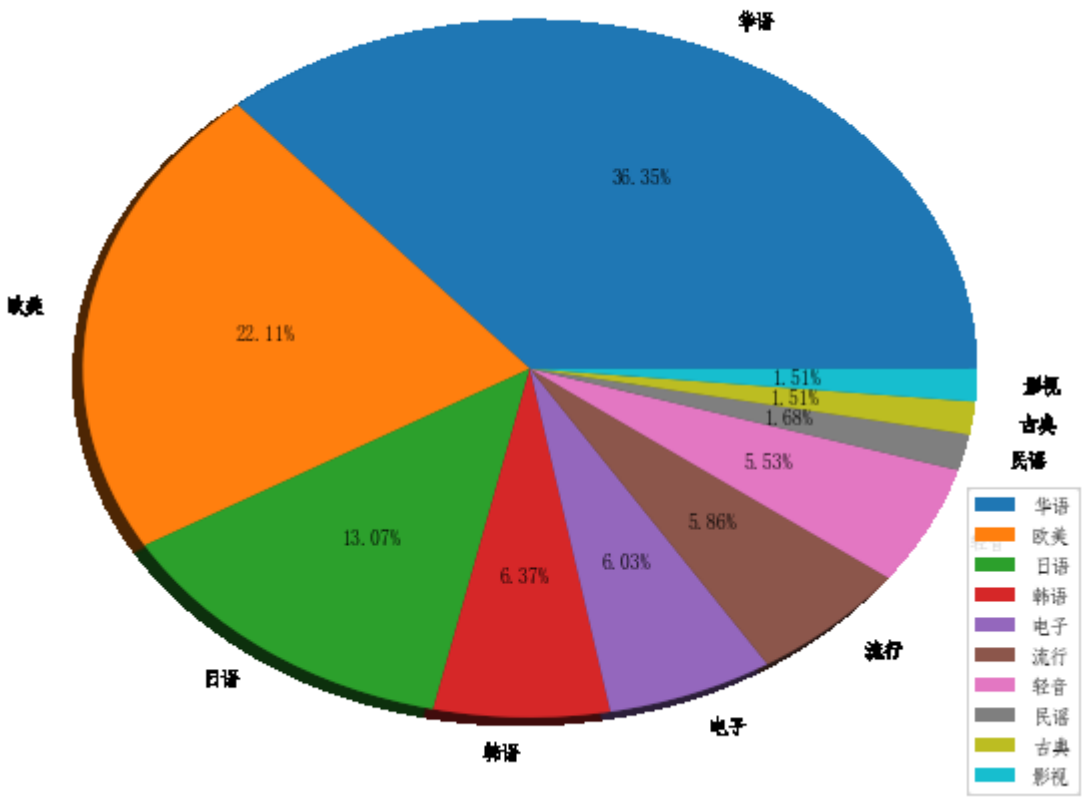

图-11 前 10 比重

3.4 系统聚类分析

系统聚类是将每个样品分成若干类的方法，其基本思想先将各个样品各看成一类，然后规定类与类之间的距离，选择距离最小的一对合并成新的一类，计算新类与其他类之间的距离，再将距离最近的两类合并，这样每次减少一类，直至所有的样品合为一类为止。

根据标签关键词，依次以收藏数，转发数，评论数为排序关键字倒序选出前 20 名。然后对前 20 数据作规格化处理，选取 K 值，再用 K-Means 算法进行简要的系统聚类处理，如图-12 所示。K-Means 算法是最简单的聚类算法之一，但是运用十分广泛。K-Means 一般在数据分析前期使用，选取适当的 K，将数据分类后，然后分类研究不同聚类下数据的特点。由聚类图可见较受欢迎的关键字仍然最终归类。可以得出与前面相似是结论。

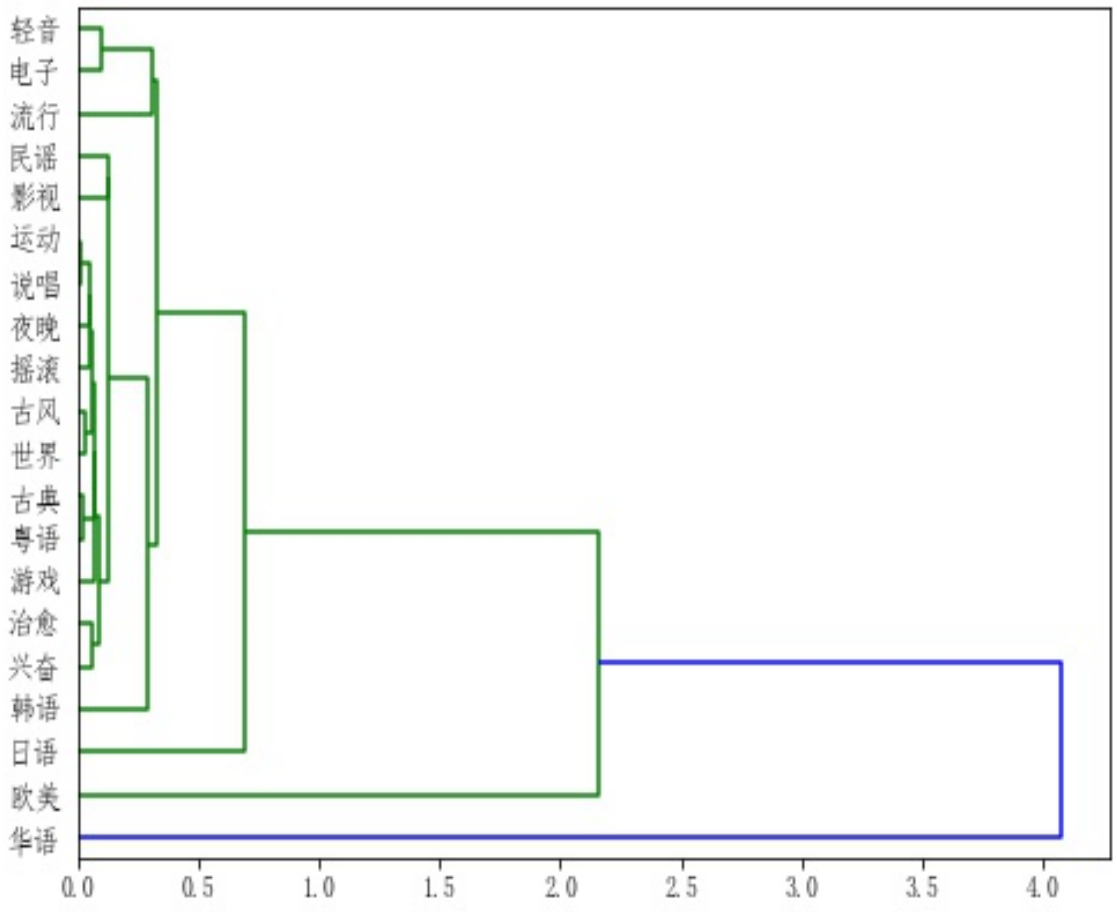

图-12 K-Means 聚类

四、自主学习-拓展 WordCloud

本阶段运用前面的网易云歌单标签关键字作扩展学习处理，主要学习 Python 的 Jieba 中文分词库及 WordCloud 词云图库。Jieba 中文分词库是将目标文本按要求取词义分隔后，试图将句子最精确地切开，把句子中所有的可以成词的词语都扫描出来，适合作文本分析。WordCloud 词云库是按词出现频率可以根据图片背景生成一张可视图，其是频率的一种体现。按标签关键字的最终词图的结果如图-12-13。

图-12 词云图

图-13 词云图

总结：

经过为期三周的辛苦学习，在各位老师的辛勤教学下，初步入门优美的 Python 语言，了解到其简易的代码及强大的功能；例如在网络数据爬取方面几行代码即能提取出想要的任何内容，在数据处理分析方面有强大的类库支持，还有各种绘图程序的搭配；Xpath 与正则表达式的结合，文本字符串的强大处理等等。正如 Python 的定位是“优雅”、“明确”、“简单”。

学习是一个持之以恒的过程，这三周仅仅是另一个开始，这三周不仅培养了自主学习的品德，还形成了一种以发现问题，寻找问题，解决问题的良好习惯。Python 语言在数据方面的学习还有很长的路要走，在其他方面也需要继续努力。

感谢各位老师的孜孜不倦，感谢各位同学的耐心帮助。

最后，望自己更勉励更进步。

<table>
<tr><td colspan="5">教师指导过程及评价意见</td></tr>
<tr><td colspan="5">指导过程简介：

对实习报告的评价意见：

教师签字：　　年　　月　　日</td></tr>
<tr><td rowspan="3">成绩评定</td><td rowspan="2">指导教师建议成绩</td><td>表现成绩
50%</td><td>报告成绩
50%</td><td>总成绩（五等级制：优秀、良好、中等、及格、不及格）</td></tr>
<tr><td></td><td></td><td></td></tr>
<tr><td colspan="4">院系（部）意见

教学副院长签字：

学院盖章：　　年　　月　　日</td></tr>
</table>

附录项目主要代码：

Scrapy—Item 类

```
# -*- coding: utf-8 -*-
import scrapy
class Music163Item(scrapy.Item):

    createBy = scrapy.Field()          # 创建者
    desc = scrapy.Field()              # 描述
    addNum = scrapy.Field()            # 收藏量
    transNum = scrapy.Field()          # 转发数
    commNum = scrapy.Field()           # 评论数
    tag = scrapy.Field()               # 标签
    imgPath = scrapy.Field()           # 图片路径
```

Scrapy--Spider 类

```
# -*- coding: utf-8 -*-
import scrapy
from Music163.items import Music163Item
from scrapy.http import Request
import requests
from lxml import etree
import re


class MusicSpider(scrapy.Spider):
    name = 'music'
    allowed_domains = ['163.com']
    # 开始爬虫地址
    start_urls = ['https://music.163.com/discover/playlist/?limit=35&offset=0']

    def parse(self, response):
        # 创建 Item
        item = Music163Item()
        # data = response.body.decode("utf-8")
        # xpath 匹配 创建者，简述，图片地址
        createBy = response.xpath('//a[@class="nm nm-icn f-thide
s-fc3"]/text()').extract()
        desc = response.xpath('//p[@class="dec"]/a[@class="tit f-thide
s-fc0"]/@title').extract()
```

```
            imgpath = response.xpath('//ul[@class="m-cvrlst f-cb"]/li/div[@class="u-cover u-cover-1"]/img/@src').extract()
            # 获取另一个页面的 url
            href = response.xpath('//ul[@class="m-cvrlst f-cb"]/li//p[@class="dec"]/a/@href').extract()

            item["createBy"] = createBy
            item["desc"] = desc
            item["imgPath"] = imgpath
            # 创建临时列表
            addNumAll = []
            transNumAll = []
            commNumAll = []
            tagNumAll = []
            # 对歌单详细页遍历
            for i in range(len(href)):
                url = 'https://music.163.com' + href[i]
                newres = requests.get(url)
                newres.encoding = 'utf-8'
                # root = etree.HTML(newres.text)
                # addNum = root.xpath('//a[@class="u-btni u-btni-fav "]/i/text()')
                # print(newres.text)

                # 匹配收藏数
                addpat = 'data-count="(.*)"\n.*\nclass="u-btni u-btni-fav "'
                addNum = re.findall(addpat, newres.text, re.S)

                # 匹配转发数
                transpat = 'class="u-btni u-btni-share ".*(\d{1,5})'
                transNum = re.findall(transpat, newres.text)

                # 匹配评论数
                commpat = '<span id="cnt_comment_count">(.*)</span>'
                commNum = re.findall(commpat, newres.text)

                # 匹配标签
                tagpat = '<a class="u-tag" href=".*"><i>(.*)</i></a>'
                tag = re.findall(tagpat, newres.text)
                # 将标签转换为字符串
                tagAll = ''.join(tag)

                addNumAll += addNum
                transNumAll += transNum
                commNumAll += commNum
```

```
            # 此处为 append()
            tagNumAll.append(tagAll)

        item["addNum"] = addNumAll
        item["transNum"] = transNumAll
        item["commNum"] = commNumAll
        item["tag"] = tagNumAll
        yield item

        for i in range(10, 16):
            url = "https://music.163.com/discover/playlist/?limit=35&offset=" +
str(i * 35)
            yield Request(url, self.parse)
```

Scrapy--Piplines 类

```
import pymysql
from pandas import DataFrame
import pandas as pd


class MokePipeline(object):

    # 构造函数--定义数据库及 MySQL
    def __init__(self):
        self.keInfoAll = DataFrame()
        self.conn = pymysql.connect(
            host="127.0.0.1",
            user='root',
            passwd='123456',
            db='python',
            charset='utf8'
        )

    def process_item(self, item, spider):

        # c 导出到.csv
        keInfo =
DataFrame([item["name"],item["rank"],item["users"],item["desc"],item["imgpath"]]).
T
        # 设置列名
        keInfo.columns = ["课程名","等级","访问量","课程描述","图片地址"]
        self.keInfoAll = pd.concat([self.keInfoAll, keInfo])
```

```
            self.keInfoAll.to_csv("keInfo.csv", encoding="gbk")

            # 存储数据到 mysql
            for i in range(len(item["name"])):
                name = item["name"][i]
                rank = item["rank"][i]
                users = item["users"][i]
                descs = item["desc"][i]
                imgpath = item["imgpath"][i]
                # 定义游标
                cursor = self.conn.cursor()
                sql = "insert into moke(name,rank,users,descs,imgpath) values
('"+name+"','"+rank+"','"+users+"','"+descs+\
                      "','"+imgpath+"') "
                # 执行 mysql 命令
                cursor.execute(sql)
                self.conn.commit()
            return item

        # 关闭 mysql 连接
        def close_spider(self,spider):
            self.conn.close()
```

数据分析简要代码：

```
import matplotlib.pyplot as plot
ax = df_10.plot(kind = "bar",fontsize=15)
ax.set(xlabel="标签" ,ylabel="歌单次数",title="歌单关键字")
plot.show()

import matplotlib.pyplot as plot
ax = df_10.plot(kind = "pie",fontsize=15)
ax.set(xlabel="标签" ,ylabel="歌单次数",title="歌单关键字")
plot.show()

import matplotlib.pyplot as plot
ax = df_10.plot(kind = "pie",fontsize=15)
ax.set(xlabel="标签" ,ylabel="歌单次数",title="歌单关键字")
plot.show()
```

词云图代码：

```
import jieba
```

```
import matplotlib.pyplot as plot
from wordcloud import WordCloud,STOPWORDS,ImageColorGenerator
from scipy.misc import imread

word =''.join(tag)
text = ''
text = ' '.join(jieba.cut(word))
#print(text)
back_img = plot.imread('hg.png')
alice_color = imread(r'hg.png')
#alice_color = imread(r'22.png')

wc = WordCloud(
        width=1000,
        height=860,
        margin=2,
        background_color='#F5FFFA',     # 设置背景颜色
        mask=alice_color,                # 设置背景图片
        font_path='C:\Windows\Fonts\STZHONGS.TTF',  # 若是有中文的话，这句代码必须添加，不然会出现方框，不出现汉字
        max_words=2000,                  # 设置最大现实的字数
        stopwords=STOPWORDS,             # 设置停用词
        max_font_size=200,               # 设置字体最大值
        min_font_size=1,
        random_state=42,                 # 设置有多少种随机生成状态，即有多少种配色方案
        scale=3
)
wc.generate_from_text(text)

#print('开始加载文本')
#改变字体颜色
img_colors = ImageColorGenerator(alice_color)
#字体颜色为背景图片的颜色
#wc.recolor(color_func=img_colors)
# 显示词云图
plot.imshow(wc)
# 是否显示 x 轴、y 轴下标
plot.axis('off')
plot.show()
wc.to_file('test.png')
```