大数据研究科技报告

# 课题背景及意义 (高树韬)

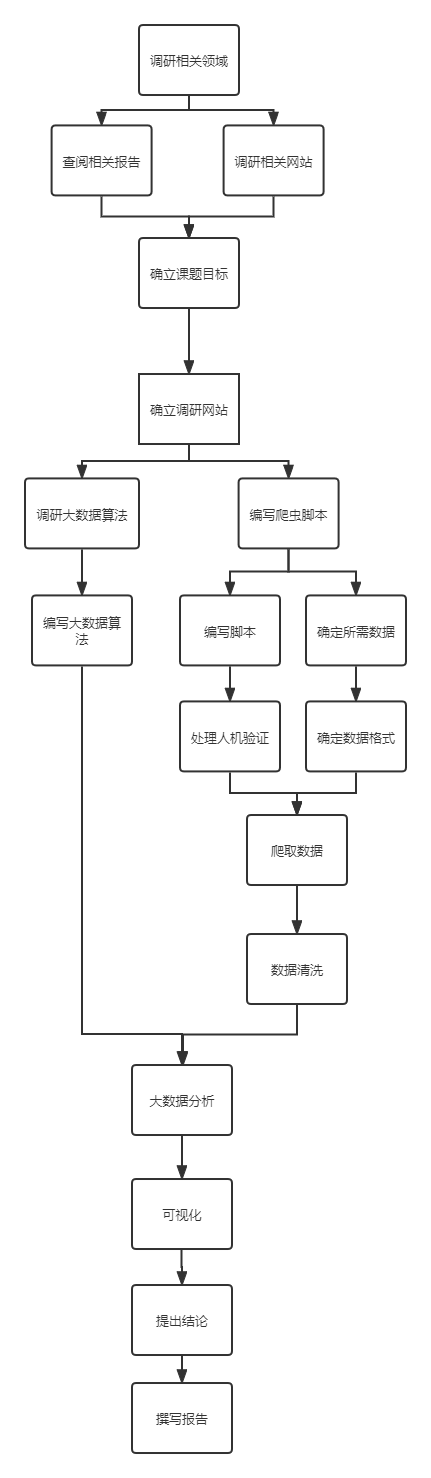
当下大龄剩男剩女数量日益增加， “催婚”和“婚姻”成为社会关注和网上热议的热门话题。近五年来25-39岁的单身人口数量逐年递增，适婚年龄段人口的性别比也在持续攀升，或将有超3000万青年男性成为剩男。这意味着一面“恐催婚”，一面“结婚难”已然成为当下年轻人面临的现状。

据珍爱网网络调问卷查调查显示，现在的年轻人其实并不是绝对的不婚主义者，反而多数人是期待爱情降临的，但是，这并不代表着年轻人愿意随意恋爱和结婚。尽管婚恋难是不争的事实，但多数青年仍然愿意等待而不是将就，将有无感觉、是否有感情和今后的婚姻质量放在了重要位置，择偶更看对方的能力，宁缺毋滥是多数青年男女的共识。

现在的年轻人婚恋更重视内在的匹配度、心灵的契合度以及性格与认知的契合度，这是婚姻品质的基本保证，但客观上也无疑增加了婚恋的成本和成功的难度，年轻人结婚难的问题已经迫在眉睫。

所以我们决定利用大数据对目前主流婚恋网站上的数据进行清洗汇总并以可视化的方式展现出来，以便得出一些输入的理论和结论来解释目前的问题，并为未来提出更加具体的婚恋问题解决方案提供一个较为基础的数据支撑。

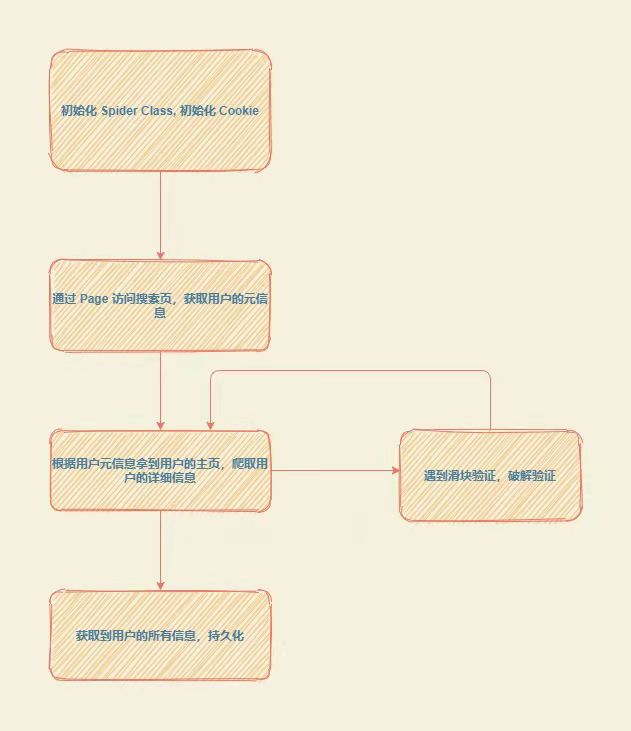
# 技术路线(高树韬)



# 数据采集(齐呈祥)

我们目前在多个相亲网站上爬取会员信息以调研国内相亲的大数据，在这里我们以世纪佳缘的爬虫为例，来介绍我们的解决方案与技术手段。

首先我们的爬虫程序，分别使用了单线程和异步来进行爬取(由于python的 GIL (全局解释器锁) 机制，我们没有采取多线程的技术方案)。单线程主要是为了更关注于爬虫的逻辑的简单实现，而异步则是在单线程的基础上做的进一步优化。



### 3.1 爬虫流程

在爬取世纪佳缘网站的时候，我们首先进入搜索页可以看到个人信息，打开Chrome开发者进行了简单的分析，可以发现可以通过一个 POST请求来获取用户的信息，于是我们可以通过模拟规定的payload的格式向对应的API发送请求即可获得对应的用户信息，这里我们需要提供一些搜索相关的信息以及对应的page\_id来爬取信息。当获得用户的信息后，我们发现拿到的其实并不是用户的所有信息，要获取到用户的所有信息，我们需要去用户的个人主页去获取，见过简单的分析，我们发现只要能拿到用户的realUid就可以构造个人主页的URL。

当访问用户的个人主页时，我们可以使用GET方法拿到用户主页的html格式的信息，然后通过BeautfulSoup和 re库的帮助我们可以很轻易地解析出用户全部个人信息。

### 3.2 反反爬虫

然而，网站对于爬虫实际对于爬虫具有一定的反爬虫限制措施。

#### 3.2.1 滑块验证码

首先，网站对于用户个人主页的访问频率有一定限制，当我们连续访问 10 个个人主页时，网站将会弹出极验验证码让用户识别(即滑块验证)，倘若我们无法通过验证则无法再继续爬取。我们一开始尝试过减慢访问频率和使用 IP 代理池都没有成功。最后我们决定暴力破解滑块验证。

和点触验证码相比，极验验证码其实相对更简单，因为破解点触验证码需要使用深度学习进行训练才可以有较高的识别率，而破解极验验证码只需要获取到拼接前的图片和拼接后的图片，之后进行逐像素的比对即可获取验证码需要滑动的距离。于是我们分析了一下网页的 html 的源代码，发现拼接前和拼接后的图像都被嵌在 Canvas中，于是我们可以使用 senlenium+ ChromeDriver 模拟用户进行操作，首先我们通过执行一段 JavaScript 程序拿到 Canvas的图片，然后逐像素比对两张图片的不同，并获取到需要滑动的距离，当我们获取到滑动距离后，我们就可以模拟用户滑动到对应的位置来破解滑块验证。

然而网站事实上对于滑块速度有一定要求，如果你是直接将滑块从开始一直滑到最终位置，一点像素偏差都没有，网站会迅速识别出你是爬虫。于是我们尝试模拟人类的滑动轨迹，我们尝试了具有一定加速度的滑动以及随机暂停的滑动还有一些较为平滑地模拟运动曲线数学公式的公式(这些定义在 track.py里面)，最终我们可以以一种较高的效率破解滑块验证。于是我们在爬取的时候检测到验证码出现的时候进行滑块验证的破解，在破解后继续进行爬取。

#### 3.2.2 模拟用户登录

由于对于网站个人主页的某些信息需要登录后才能查看，因此我们必须模拟出用户登录的 "假象"，从而才能获取到用户的全部信息，因此我们登录后可以从 Chrome 检查者中拿到我们的 Cookie 之后使用 requests 库的 RequestsCookieJar 属性将其解析成 requests 能够识别的格式，最后每次访问都将 cookie 带上即可模拟用户的登录。

### 3.3 异步

在 python 提供了 asyncio 这个库实现异步编程，我们可以使用 async 和 await 来做异步。事实上 asyncio 提供的异步为无栈协程，所谓无栈协程即是在用户态实现的轻量级无栈的线程，它使用状态机来实现线程的功能，并由一个 Runtime 来实行调度，以充分利用机器的性能，例如，当我们进行网络请求的时候会产生阻塞，因此我们可以把网络请求，标记为 async，而在调用网络请求时使用 await 来执行异步调用，当网络请求阻塞时，他会立刻调度到其他函数，当网络请求成功后，它将会收到网络请求返回的消息并继续处理，这样就不会在网卡阻塞时占用 CPU 的时间，从而充分利用了机器的性能。其中调度的部分全部是由语言内部的 Runtime 来做，async 和 await 是一个语法糖，在运行时翻译成 Runtime 的函数。

在我们的实现中，我们实现了使用异步来爬取、解析、持久化等操作，通过在标记为 async 的函数中使用 asyncio.create\_task() 来创建异步任务，并对其执行 await 操作即可使其异步运行。由于 async 操作具有传染性，所有调用 async 的函数都必须是异步函数，因此当我们调用顶层函数时，需要使用 asyncio.run() 来执行。

### 3.4 CI/CD

为了更快的爬取到数据，我们利用了 Github Action 的功能白嫖 github 服务器，每当我们向仓库 push 或者 Pull Request 的时候都会对只能的网站进行爬取与记录，最终将其写入文件并 push 到仓库中。

# 数据清洗（康玥）

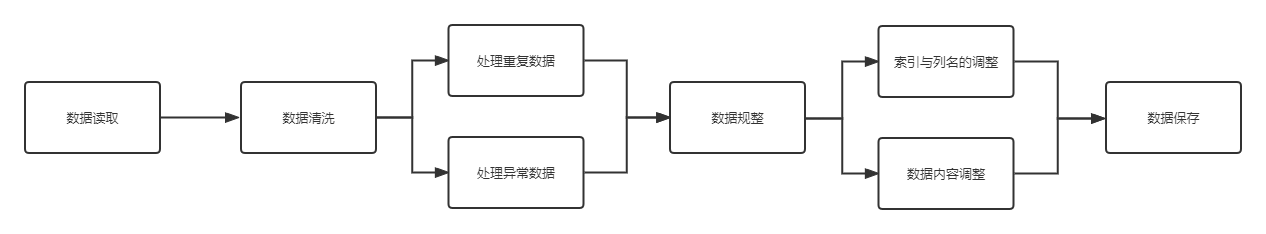
### 4.1 数据清洗

针对所爬取的数据源，利用Python结合Pandas来进行数据清洗的工作。

#### 4.1.1 理想数据

1. 数据源中所有数据均不带索引并有对应列名；
2. 数据源中所有数据均无重复与冗余；
3. 数据源中所有数据均无异常。

#### 4.1.2 理想处理过程



#### 4.1.3 具体流程

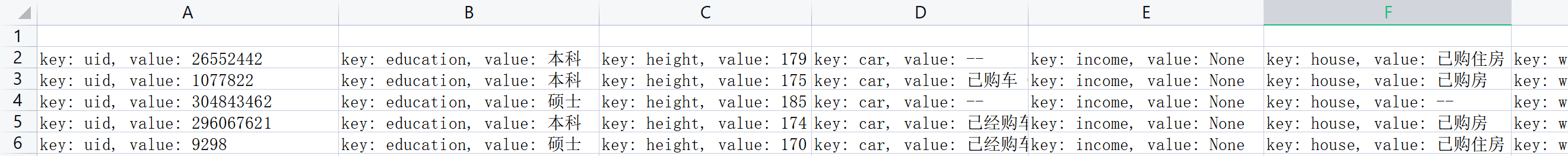
数据读取：pandas默认编码是utf-8，以数据源.csv为例，数据源中包含中文，指定encoding='utf\_8\_sig'；sep:指定分割符形式，CSV默认逗号分割，可以忽略这个参数。

读取代码：data = pd.read\_csv('世纪佳缘-男.csv')

##### 1）数据清洗

1. 检查数据

由于我们的数据源较为庞大，所以需要先检查一下我们刚刚读入数据的基本结构，Pandas 提供了 head() 方法打印输出前五行数据。目的是让我们对读入的数据有一个大致的了解。



通过记录发现，现有数据存在的问题有：缺失列名；无用字母与字符的重复；无用项的干扰；某些数据异常；数据缺失值等问题。

1. 处理重复数据

在数据清洗时，发现20万条数据，删除重复项后，只有几千条，推测是搜索的排名规则问题，导致很多人大量重复地出现。在处理重复数据时首先用duplicated()方法进行逻辑判断，确定是否有重复值。再用duplicates（subset,keep,inplace)方法对某几列下面的重复行删除：

subset:以哪几列作为基准列，判断是否重复，如果不写则默认所有列都要重复才算；

keep: 保留哪一个，fist-保留首次出现的，last-保留最后出现的，False-重复的一个都不保留，默认为first；

inplace: 是否进行替换，最好选择False，保留原始数据，默认也是False；

1. 处理异常数据

缺失数据是最常见的问题之一。产生这个问题可能的原因有：从来没有填正确过；数据不可用；计算错误。无论什么原因，只要有空白值得存在，就会引起后续的数据分析的错误。我们采用去掉/删除缺失率高的行这种方法来处理缺失的数据。

先通过isnull函数看一下是否有空值，结果是有空值的地方显示为True，没有的显示为False；再通过isnull().any()直接看每一列是否有空值，这个是只要这一列有1个空值，结果就是True;如果想具体看哪几行有空值，可以再用data.isnull().values==True来定位

假设我们想删除任何有缺失值的行。这种操作太据侵略性，但是我们可以根据我们的需要进行扩展。

删除任何包含 NA 值的行比较容易操作：

data.dropna()

当然，也可以删除一整行的值都为 NA：

data.dropna(how=**'all'**)

可以增加一些限制，在一行中有多少非空值的数据是可以保留下来的（在下面的例子中，行数据中至少要有 5 个非空值）

data.drop(thresh=5)

也可以上面的操作应用到列上。仅仅需要在代码上使用 axis=1 参数。这个意思就是操作列而不是行。（我们已经在行的例子中使用了 axis=0，因为如果我们不传参数 axis，默认是axis=0。）

##### 2）数据规整

##### （1）索引与列名的调整

最终的数据可能是有计算机生成的，那么，列名有可能也是计算机按照一定计算规律生成的。这些列名对计算机没有什么，但是对于人来说可能就不够友好，这时候，我们就需要重命名成对人友好的列名，代码如下：

data, rename(columns={**"uid"**, **"education"**, **"height"**, **"car"**, **"income"**, **"house"**, **"weight"**, **"constellation"**, **"nationality"**,  
 **"zodiac"**, **"blood"**, **"criterias\_age"**, **"criterias\_height"**, **"criterias\_nationality"**,  
 **"criterias\_education"**, **"criterias\_marriage"**,  
 **"criterias\_location"**, **"nickname"**, **"sex"**, **"marriage"**, **"work\_location"**, **"image"**, **"randTag"**,  
 **"randListTag"**, **"shortnote"**})

像上面这样，我们就完成了n个列的重命名。

经过商议决定我们不用数据源的索引，因此在结果中删除索引以便进行后续操作。

##### （2）数据内容规整

刚拿到的数据可能都需要进行一些必要的变换，比如去除掉我们所不需要的列，以及所需信息中的冗余值，还有一些内容的简易数字替换（性别男统一为1，性别女统一为0）等。

以下为去除所需信息中的冗余值：

data[**'shortnote'**] = data[**'shortnote'**].str.replace(**'key: shortnote, value: '**, **''**)  
data[**'shortnote'**] = data[**'shortnote'**].str.replace(**'?'**, **''**, regex=**False**)

值得注意的一点是，我们的数据中含有一些复杂的数据冗余需要利用到正则表达式进行相应的替换与补充，具体如下：

# 正则表达式：\s.

**def** clear\_characters(text):  
 **return** re.sub(**'\W'**, **''**, text)  
 data[**'criterias\_age'**] = data[**'criterias\_age'**].astype(str)  
 data[**'criterias\_height'**] = data[**'criterias\_height'**].astype(str)

但在去掉冗余后发现一些身高和体重范围之间的短符号”-“也被剔除，需要重新加入

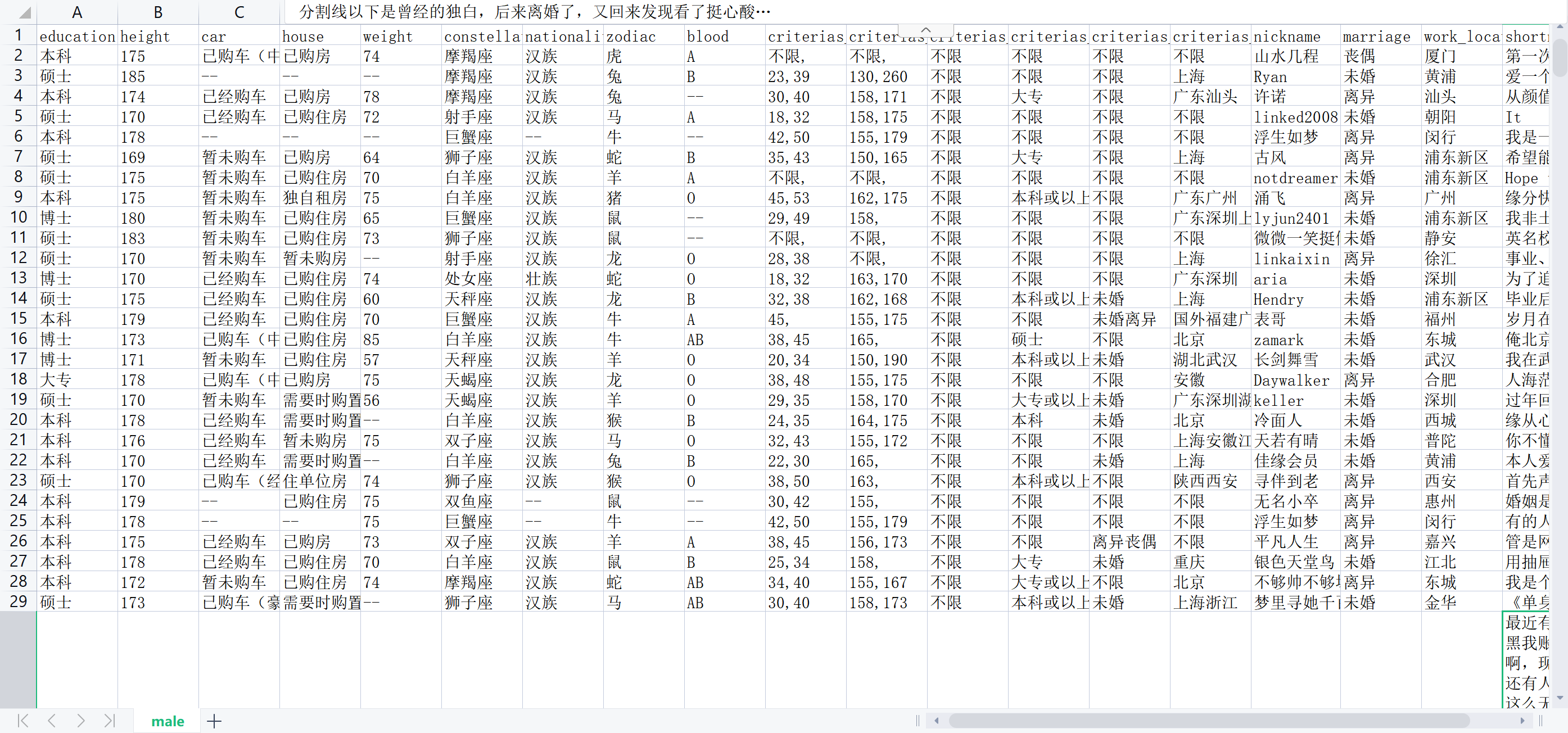
data[**'criterias\_age'**] = data[**'criterias\_age'**].str[0:2] + **','** + data[**'criterias\_age'**].str[2:]  
data[**'criterias\_height'**] = data[**'criterias\_height'**].str[0:3] + **','** + data[**'criterias\_height'**].str[3:]

### 4.2 数据保存

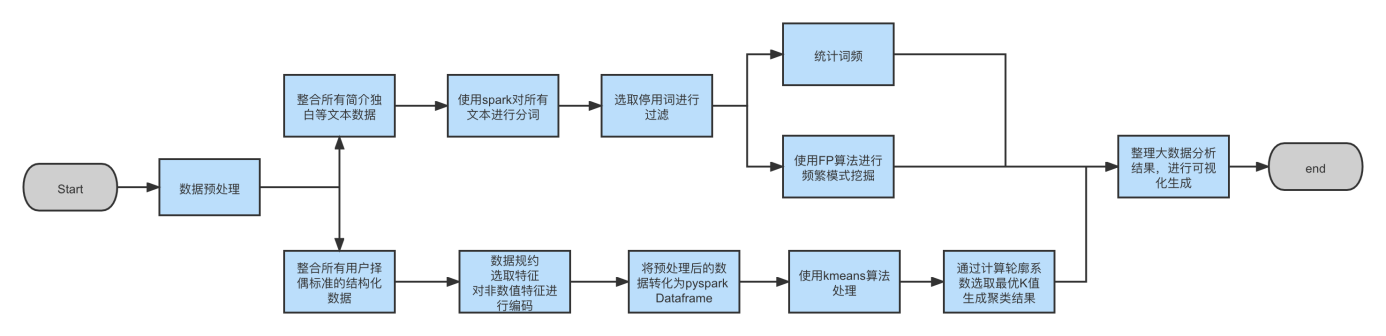
我们完成数据清洗之后，一般会把结果再以 csv 的格式保存下来，以便后续其他程序的处理。同样，使用Pandas：

data.to\_csv(‘cleanfile.csv’ encoding=’utf-8’)eight**'].str[3:]**

### 4.3 结果一览



# 数据加工与大数据分析算法（张轶群负责）



### 5.1 词频计算

在该课程设计中，我们选择了spark作为大数据算法的框架。并使用pyspark进行实际代码编写和生成应用。从相亲网站上爬取的所有数据在经过预处理后都上传至三虚拟机构成伪集群的HDFS。

jieba分词 统计各种词出现的频率

我们通过爬虫收集到的各种相亲网站的数据中，都有“个人简介”和“内心独白”这种类型的数据。从类型格式的角度上分析，这些数据都是一段相亲网站用户想说的话。长度和内容没有限制，有关于自己各种方面的介绍，也有对于理想另一半的一些期望。总之，这类数据比较灵活多样，属于非结构化的文本数据，是我们所获取到数据中占比较大的部分。对数据进行初步审查和清洗后，可以发现这类数据中确蕴含了很多有价值的信息。相比于其他格式固定的个人信息，例如身高、年龄、择偶要求身高等，纯文本可以包含的信息极其丰富，无论是涉及领域还是内容都大大扩展。如何通过各式大数据分析算法挖掘出其中的价值，是我们的主要任务和挑战。

回到我们的目标和数据本身，我们希望从收集到的数据中分析出当今相亲人士的特征、择偶心理和择偶要求，从而进一步推出结论。个人简介这一类数据有对上述需求的直接体现。但是每个人各不相同的内心独白无法让人非常直观地了解到这些文字主要想表达的内容。所以我们需要对这类文字进行处理。在思考过后，一个最简单却十分有效的spark应用——WordCount成为了理想的选择。词频计算是我们学习hadoop和spark第一个接触到的应用，在该场景下，如果对每个人的内心独白进行分词处理后计算词频，可以得到在大家的文字描述中频繁出现的词语，其中的很多词语直观而精炼地表达了用户的情感和态度等内容。而出现的次数则显示了哪些内容是大家普遍注重的。

分词我们使用了python的jieba库。将处理好的数据上传至HDFS后就可以编写spark应用，读入RDD数据后进行包括分词和展平flatten的一系列map变换后，最终降序得到词语和其出现的次数。

在该算法还有一个至关重要的部分就是选取停用词并过滤。我们希望最终得到的词语都具有一定的现实意义，反映用户想表达的核心内容。例如：“真诚”表达了相亲者注重的个人品格，“家庭”表达了相亲者注重家庭，再或是“健身游泳”传达了兴趣爱好。而其他无法单独表达含义的词语就需要我们过滤掉。除了常见的中文自然语言处理中的常用停用词外，我们还需要根据我们的需要来不断筛选和过滤。最终将选取的所有停用词上传至HDFS，与数据一并读入然后进行过滤处理。

词频计算的最终结果可以进行可视化生成词云，进一步直观地展示给我们当今相亲者的注重点、特征、态度等信息。

### 5.2 频繁模式挖掘

频繁模式挖掘是数据挖掘的重点内容，也是我们课上详细讲解的内容。在很多场景下，我们都希望挖掘出事物之间的联系。在本次课程设计中，经过对个人简介这类数据分析和词频计算后，可以对其进一步使用频繁模式挖掘算法。从我们数据处理的目的和现实意义上讲，在得到各种具有表达性词语的词频统计后，很自然地想得到哪些词语经常一起出现。单独的词频结果无法反映信息之间的相关性。在相亲用户中，人们的需求和特征各式各样，仅仅是最频繁的需求和特征无法提供有力的材料来帮助我们分析和推断论证。我们想知道哪些内容是经常同时出现的，例如，“简单”和“真诚”同时出现表示用户希望对方性格简单而真诚。“责任心”和“性格”同时出现表示用户重点关注对方是有具有责任心以及性格特征。除了这些语义的搭配外，多个词一起出现还能够相互加强含义，使得信息表达更精确。例如“共度”和“余生”。单独的“共度”表达有些不明确，“共度余生”则非常明确。当然，这类问题也可以通过分词算法的改进来实现。但因为文本数据的丰富，有时各种搭配都会出现，进行频繁模式挖掘可以帮助我们更好地理解数据

频繁模式挖掘的处理与分词有些差别。不能flatten词语，单位数据项是每个人的内心独白，而每个内心独白进行分词处理，最终得到一个string array。我们的目的就是挖掘出哪些词语经常在同一句话中出现。在pyspark中，将分词处理后的数据转换为DataFrame格式。选取的挖掘算法为FP-growth频繁模式树算法，其相对于Apriori算法效率更高。由于文本数据的庞大，各种词语的数量极大，在经过多次尝试后选取了支持度为0.005，置信度为0.005。最终根据frequent和confidence降序输出结果。

### 5.3 聚类算法

在对所获取的所有数据进行分析并结合我们的研究目的后，我们决定以择偶标准作为特征尝试使用机器学习中的聚类算法来分析研究。择偶标准有年龄、身高、学历和婚姻状况等多种类型。我们期望得到当今相亲人士需求的大体分布。与对客户信息进行聚类分析类似，对相亲用户的择偶标准使用聚类可以帮助我们进一步窥探到相亲市场的状况，以便于下一步的分析和论证。

聚类算法选取了经典的K-means算法，在使用算法前，要对数据进行统一的预处理。原始数据的身高和年龄标准一般以区间形式给出。但由于有些用户只填写了下限，需要对数据进一步规范处理。最终我们选择将身高年龄这类数据使用下限和上限两个特征来表示。缺少上限的数据以所有数据的最大值代替。学历和婚姻状况则是一种分类属性。数据仍然有格式混乱和异常的情况出现。最终我们决定以“下限”来表示这一特征，即某人的要求为本科，则本科以及本科以上，包括硕士、博士都可以接受。使用顺序编码对该特征进行编码。此处在考虑使用独热编码还是顺序编码时，因为我们认为作为下限，需求为大专和需求为博士之间的差距相比于需求为本科和需求为博士的差距要更大一些。所以最终选择按照学历从低到高的顺序进行编码，这样使用聚类算法在欧氏空间上计算距离时，能够按照我们设想的方式进行。婚姻状况和学历类似。我们以标准从低到高进行编码，例如只接受未婚则比可接受未婚、离异要求更高。

数据经过预处理后，上传只HDFS并以pyspark dataframe的形式读入。在做聚类前对每一类特征进行StandardScaler标准化处理，使得数据规范为均值0，方差1的分布。kmeans算法的K值需要手动设置，由于数据量较大，对于聚类中心个数难以估计。我们选择根据Silhouette Coefficient轮廓系数选取K值，对于所有数据，尝试k在2-100范围内使用聚类算法，然后对结果计算轮廓系数，最终选择轮廓系数最大的最优k值进行聚类。

# 可视化(高树韬 康玥)

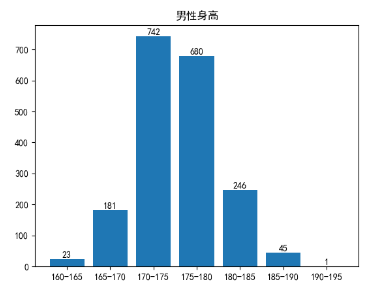
### 6.1 技术栈选取

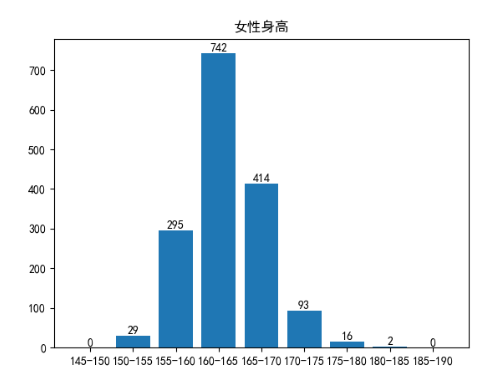
针对我们清洗获得的多列数据，以及大数据算法加工过后的多元数据，我们选取了很多可视化展示的方法。基于python 平台 我们选择了pandas作为我们数据处理的依赖，matplotlib生成简单图片，利用sklearn进行了降维处理方便可视化展示。对于词频分析，我们选取了wordcloud这一库进行词云的生成。而对于频繁模式挖掘的数据，matplot已经无法满足我们的需求，我们转而选取基于web的echarts进行可视化展示，并且生成了web页面。

### 6.2 直接数据可视化展示

本节中，我们对一些大家较为关心，以及更符合我们研究方向的单列单维度数据进行了可视化的展示，我们可以从中获取到一些大家关心的关于相亲网站上用户数据的简单统计。

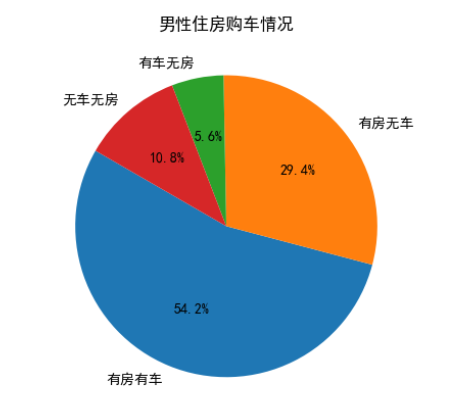
#### 6.2.1 男女身高统计

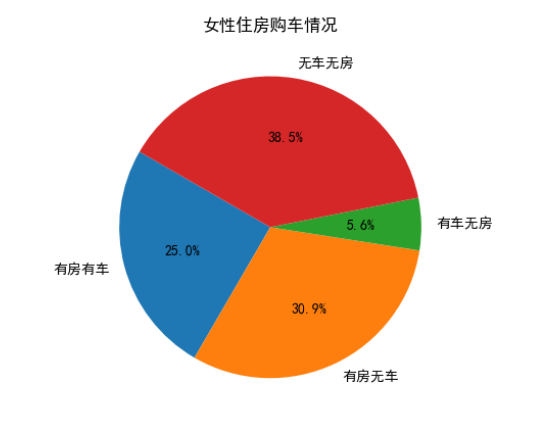




**结论：**针对这两句图中对比，我们可以得知男性身高大多处于170-180的区间，而女性身高则处于160-170的区间。

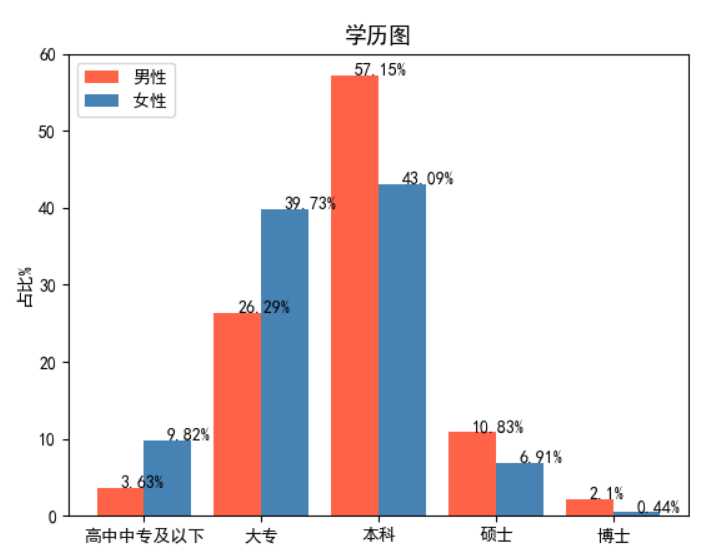
#### 6.2.2 男女财产统计





**结论：**通过图片我们可以看出在对于不动产的准备和购车情况上，男性略微好于女性。

#### 6.2.3 男女学历统计



**结论：**对于男女学历分布来说，男性学历普遍高于女性

### 6.3 大数据分析算法结果

#### 6.3.1 基于词频计算的词云

世纪佳缘网和珍爱网用户可以写短评，内容可以是自我介绍，也可以抒写对对方的期许。通过对短评分词，分别对不同婚姻状况的用户做了词云图，如下：

女性个人宣言词云



男性个人宣言词云



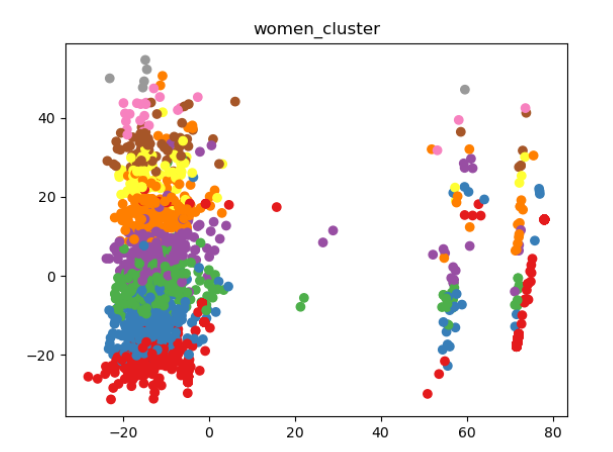
不管是男性女性，他们的情感诉求中：“生活”、“喜欢”、“善良”都是非常重要的几点。

男性的生活诉求要比女性更重一些，并且“真诚”也占了相当大的一部分比重，女性相对而言更感性一些，“性格”、“朋友”，“幸福”等成为关键词。

都说婚姻是走向爱情的“坟墓”，希望每个人都心甘情愿找到属于自己的那片归土，相濡以沫，落叶归根。

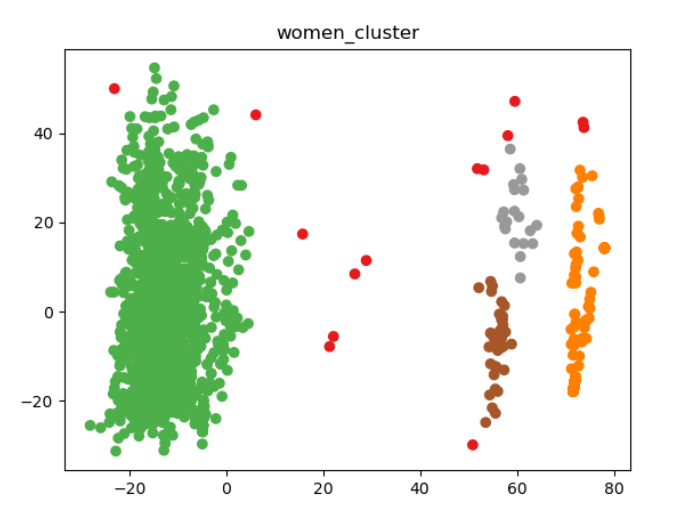
#### 6.3.2 kmeans聚类结果降维展示

我们将大数据分析后的聚类结果使用PCA降维后进行染色进行展示,基于最大轮廓系数的类别为78类



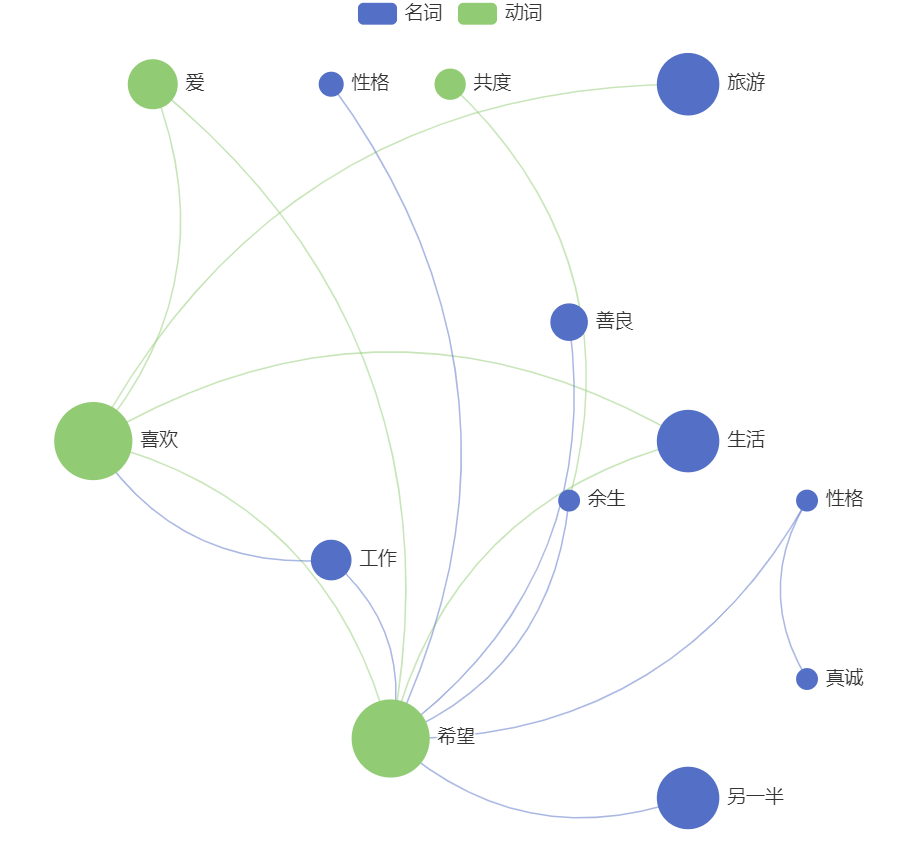
#### 6.3.3 DBSCAN聚类结果展示

本图为基于PCA+DBSCAN的聚类展示，一共有4类和-1类



#### 6.3.4 频繁项集

我们使用echarts对频繁项集中出现次数较为频繁的词汇进行了可视化的展示，其中我们选取了几个出现次数最多的词汇进行了建模，由图中我们可以看出，名词远远多于动词出现，但是喜欢希望是两种类别中出现次数最多的词汇，而希望作为动词来说和其他的名词联系最为紧密和多。



# 结果论证(张轶群 高树韬 康玥)

从相亲网站上搜集的用户资料可以帮助我们分析当今单身人士的特征和心理诉求，有助于社会各界做出努力来促进有需求人士的匹配和恋爱。当然，仅仅婚恋网站上的数据主要面向渴望恋爱和渴望婚姻的单身人士。类似于“幸存者偏差”，无法展现出不愿婚恋人士的心理因素。

### 7.1 结论一

**婚恋网站用户都表达了对爱情的渴望，需求最多方面是性格品质和生活品质方面**

从相亲网站用户的个人介绍和内心独白的词频分析中，我们可以明显看到无论男女都有着强烈的想找到另一半的期待和渴望，对爱情的向往。这也是词频最大的内容。接下来就是对相亲对象性格和未来生活品质的一些诉求。例如“真诚”、“简单”、“温柔”等美好品质，还有“快乐”、“运动”、“旅游”等兴趣爱好和生活方式方面的要求。当然，也有“收入”、“工作稳定”等现实层面的需求。男女之间的差异总体来说并不大，只在某些成分的比重上有所区别。

### 7.2 结论二

**婚恋网站用户在个人条件上往往有一定的趋同性**

在数据处理与可视化中，我们对女性的数据进行了聚类，通过不同的处理方式我们获得两种聚类结果。我们在2000的女性样本中成功使用kmeans和dbscan聚类出23类和4类，由此我们可以获得一个大致的结论，对于女性样本来说，其往往存在一个特定的模式使得他们选择在婚恋网站上进行公开择偶。再次举一个例子，对于一些重视个人发展，事业心较强或者学历较高的女性，往往在20-30岁之间将精力放在自我提升上，她们往往拥有较好的学历和条件，这也与我们的统计结果相符合，其他类似的情况也出现在各种各样的群体上，但是总体而言，婚恋网站用户有一定的趋同性。

### 7.3 结论三

**不同性别的学历分布情况及其对感情的诉求不尽相同**

结合kmeans聚类算法和学历分布统计图可以得出：假如某人的学历为本科，则另一半的学历应为本科以及本科以上，包括硕士、博士都可以接受。统计图中显示男女性都是本科占比最高，其次是大专。男性在本科以上学历人数要多于女性，女性在本科以下人数多于男性，男性的受教育水平要高一些。57.15%的男性本科生中有近87%都需求另一半不能低于本人学历，而学历为大专对另一半的受教育程度需求和学历为博士对另一半的受教育程度需求之间的差距相比于学历为本科和学历为博士的差距要更大一些。总之，自身受教育程度越高对另一半的教育水平要求就会相对越高。再根据筛选出的高学历人群的个人简介分析原因可能有：保持后代基因优良；思想观念个人状态相近导致沟通不会产生过大的差异；父母的要求与限制等。

总而言之，爱情从来不是两个人的事，也不是一朝一暮的事。互联网时代，婚恋网站通过资源+技术匹配，帮助2亿单身男女高效寻找另一半成为可能，然而在爱情这场长跑中，能够走到终点，仍需社会各方和个人共同的努力。