

# 用python对微信好友进行分析

## 前言

随着现代人的“手机依赖症”越来越深入骨髓，微信一定是大家每天离不开的一款app。通讯，社交，支付.....

而我们的python一如既往地强大。这篇文章就会手把手地拆解如何利用python来爬取微信好友的数据并对它们进行分析。



人生苦短，我用Python！

本文中使用到的python第三方模块主要有：

- **itchat** 获取微信好友数据
- **pyecharts** 绘图
- **snownlp** 文本情感分析
- **jieba** 分词
- **wordcloud** 绘制词云
- **百度人脸api** 人脸识别

话不多说，立刻开干！

## 获取数据

俗话说得好，巧妇难为无米之炊。因此首先我们需要爬取到微信好友的数据，这里使用的是 **itchat** 模块。**itchat** 是一个开源的微信个人号接口。安装 **itchat** 十分简单，老样子，在终端使用以下命令即可。

```
pip install itchat
```

网页爬虫可能会写长长的代码，然而使用以下三行代码，即可爬取到微信好友数据，so easy。

```
import itchat
itchat.auto_login(hotReload = True)
friends = itchat.get_friends(update = True)
```

通过 `itchat.get_friends` 方法返回来完整的好友列表，其中每个好友为一个字典。先来看看它返回的数据：

```
[<User: {'MemberList': <ContactList: []>, 'UserName': '@d4f8019e6b326acedd6d7dadb30594ee', 'City': '', 'DisplayName': '', 'PYQuanPin': '', 'RemarkPYInitial': '', 'Province': '', 'KeyWord': '', 'RemarkName': '', 'PYInitial': '', 'EncryChatRoomId': '', 'Alias': '', 'Signature': '你的孤独是一座花园', 'NickName': '爽', 'RemarkPYQuanPin': '', 'HeadImgUrl': '/cgi-bin/mmwebwx-bin/webwxgeticon?seq=1156905341&username=@d4f8019e6b326acedd6d7dadb30594ee&skey=@crypt_d773b9a7_72fbedf97afafb84d18ffa60509e8e51', 'UniFriend': 0, 'Sex': 2, 'AppAccountFlag': 0, 'VerifyFlag': 0, 'ChatRoomId': 0, 'HideInputBarFlag': 0, 'AttrStatus': 0, 'SnsFlag': 49, 'MemberCount': 0, 'OwnerUin': 0, 'ContactFlag': 0, 'Uin': 2535459940, 'StarFriend': 0, 'Statues': 0, 'WebWxPluginSwitch': 0, 'HeadImgFlag': 1}>, <User: {'MemberList': <ContactList: []>, 'Uin': 0, 'UserName':
```

这样的数据形式实在是有点辣眼睛，我们再使用大名鼎鼎的 `pandas` 将所需要好友信息转化为数据框的格式存储起来。

长舒一口气，瞬间顺眼了很多。

	nickname	sex	province	city	signature	headimgurl
0	豚	1	四川	成都	大家好我是性感的贝贝	/cgi-bin/mmwebwx-bin/webwxgeticon?seq=68920416...
1	饭王	1	四川	成都		/cgi-bin/mmwebwx-bin/webwxgeticon?seq=65984225...
2	Raneelee••	2	四川	巴中	something maybe special	/cgi-bin/mmwebwx-bin/webwxgeticon?seq=581292&u...
3	Lamm	2	香港		Be mentally rich	/cgi-bin/mmwebwx-bin/webwxgeticon?seq=64133660...
4	李岑思	0				/cgi-bin/mmwebwx-bin/webwxgeticon?seq=63241721...
5	电路图	0				/cgi-bin/mmwebwx-bin/webwxgeticon?seq=64684384...
6	Bridget-C	2	四川	绵阳	缙仙扶醉跨残虹	/cgi-bin/mmwebwx-bin/webwxgeticon?seq=581411&u...
7	冯二	1	四川	成都	当时明月在，曾照彩云归！	/cgi-bin/mmwebwx-bin/webwxgeticon?seq=62268962...
8	7788	1	北京	朝阳	革命尚未完成 同志仍需努力	/cgi-bin/mmwebwx-bin/webwxgeticon?seq=64407438...
9	我被名字气死了	1	四川	成都	爱你，只要你在身边都是幸福，你是天赐之物，珍惜你，尽我一生！	/cgi-bin/mmwebwx-bin/webwxgeticon?seq=64405863...

## 好友地理位置分析

拿到数据后，我们对微信好友的地理位置进行分析。

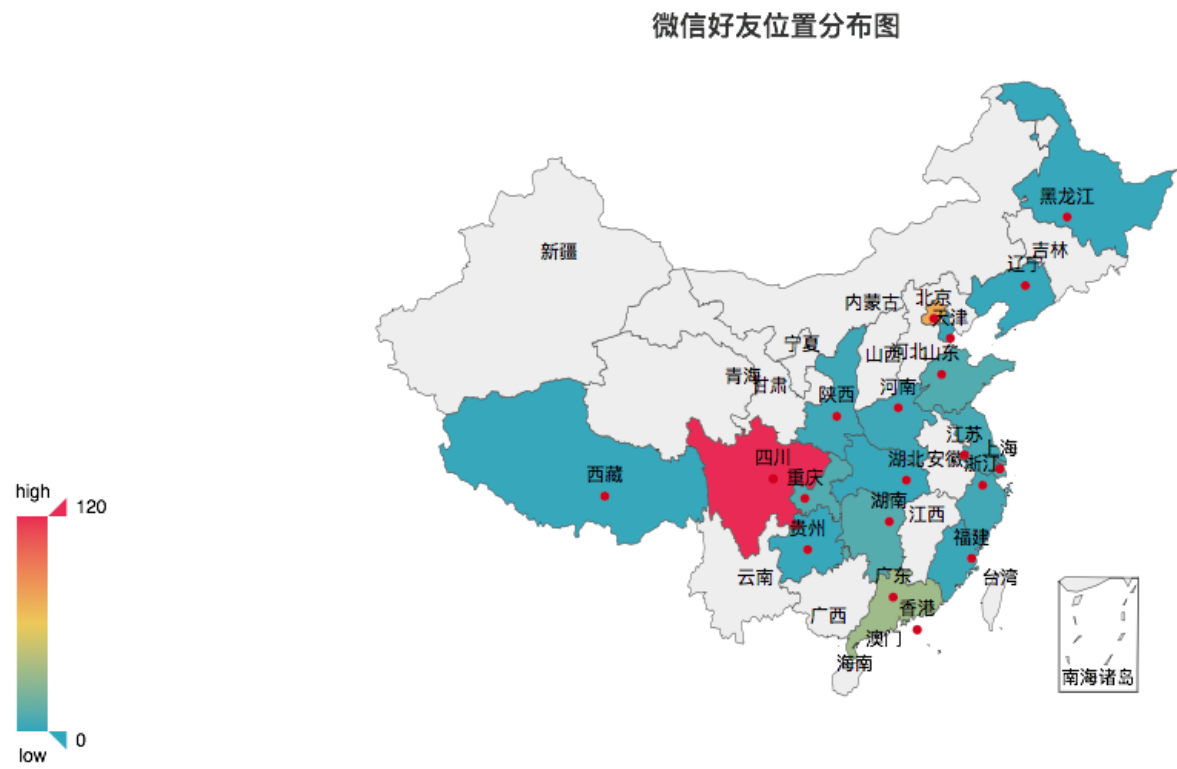
为了更直观地展示数据，我们进行了如下的可视化。这里我使用的是 `pyecharts` 模块。

`pyecharts` 的功能强大，而且作出的图十分美观，实在是可视化必备之良品。

```
from pyecharts import Map
provinces_count = data.groupby('province', as_index=True)['province'].count().sort_values()
attr = list(map(lambda x: x if x != '' else '未知', list(provinces_count.index)))#未填写地址的改为未知
value = list(provinces_count)
map_1 = Map("微信好友位置分布图",title_pos="center",width=1000, height=500)
map_1.add('', attr, value, is_label_show=True, is_visualmap=True, visual_text_color='#000',visual_range=[0,120])
```

```
map_1.render(path='your own path/微信好友分布.html')
```

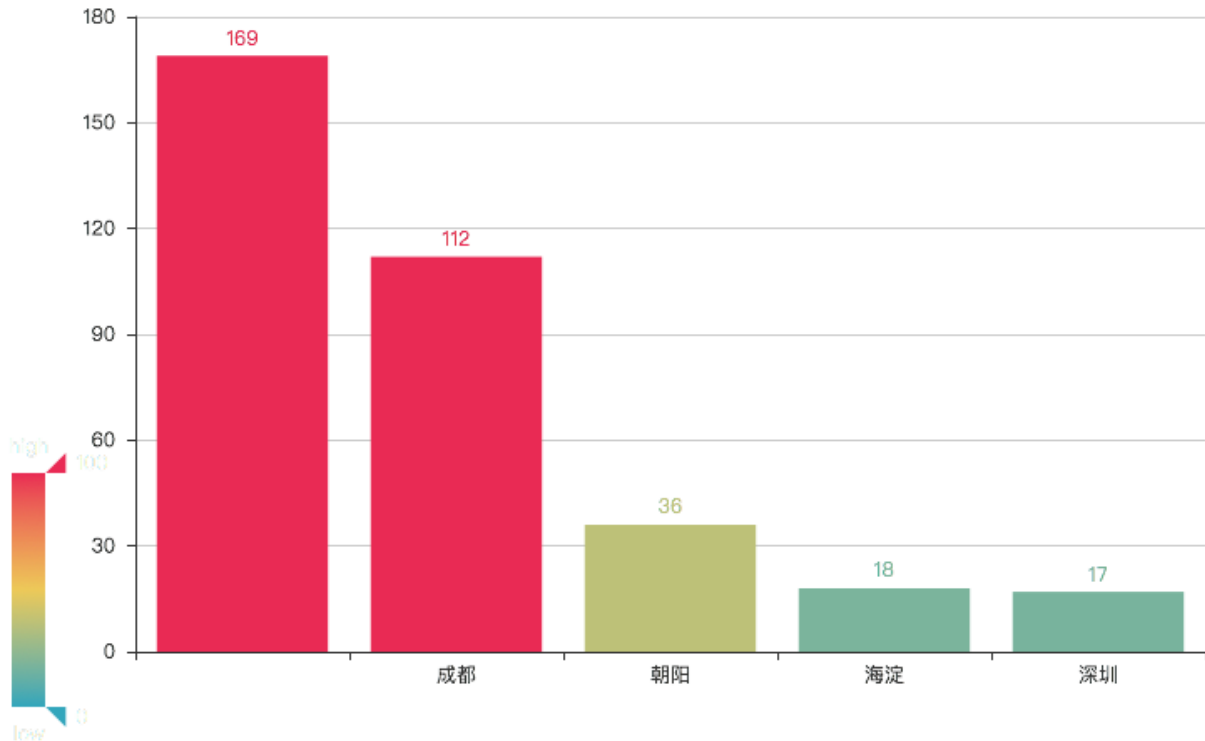
颜色越趋近于红色的地方说明好友越多。可以看到我的好友主要集中在四川，北京，广东这三个省份。这正是小编出生，上学，工作的几个省份，完全契合了我的生活轨迹。



再来统计一下我的微信好友主要都分布在哪些城市。

```
from pyecharts import Bar
from collections import Counter
city_5=Counter(city).most_common(5)
bar = Bar('好友所在城市TOP5', '', title_pos='center', width=800, height=500)
attr, value = bar.cast(city_5)
bar.add('', attr, value, is_visualmap=True, visual_text_color='#fff', is_more_utils=True, is_label_show=True)
bar
```

## 好友所在城市TOP5



emmmm..... 排在第一位的是空白，也就是说有很多好友并没有标明自己所在的城市。当然也许每个人的微信里都有几个位于“安道尔”，“阿布扎比”的好友，毕竟城市是用户随便填的而不是微信定位来的。

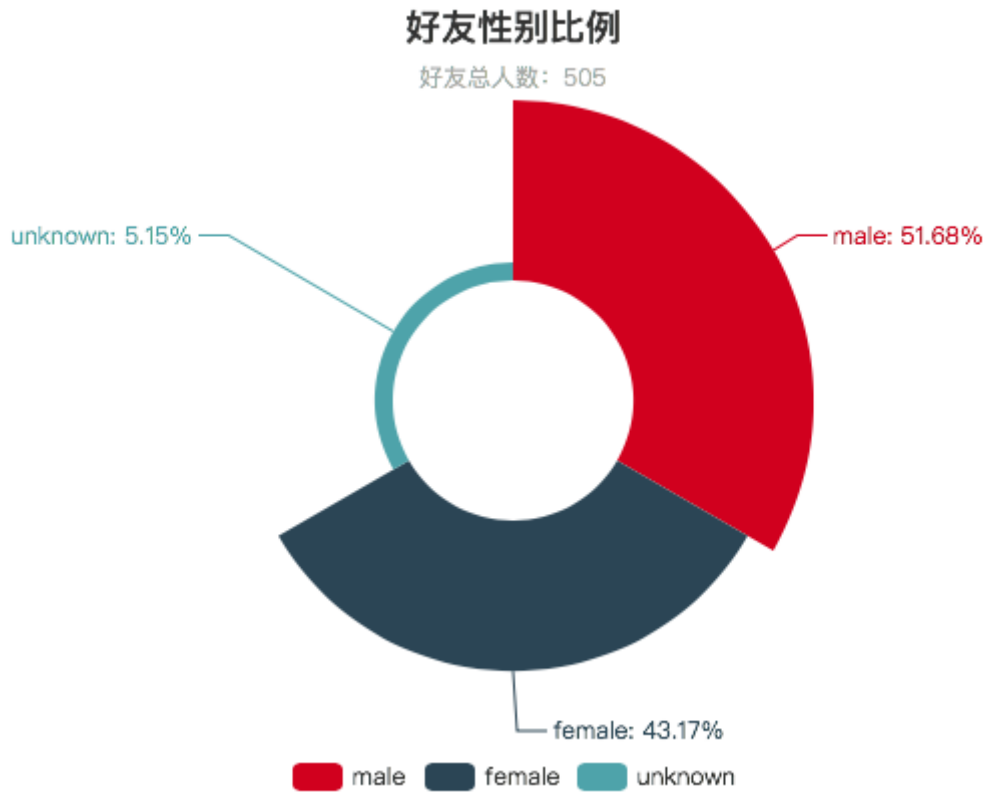
排在后面的分别为成都，朝阳，海淀，深圳。与上面的地图也可以对得上号。但是微信为何将朝阳和海淀定义为城市而不是地区。我不由得陷入沉思。

## 好友性别分析

对好友的性别进行分析，来看看我的微信好友男女比例为多少。同样地这里使用到了 `pyecharts` 中的饼图。

```
from pyecharts import Pie
sexs=list(map(lambda x:x['Sex'],friends[1:])) #提取好友性别
value = [sexs.count(1), sexs.count(2), sexs.count(0)]#对性别进行计数
sex_attr=['male','female','unknown']
pie = Pie('好友性别比例', '好友总人数: %d' % len(sex), title_pos='center')
pie.add(' ', sex_attr, value, radius=[30, 75], rosetype='area', is_label_show=True, is_legend_show=True, legend_top='bottom')
pie
```

如下图所示，小编的男性好友为51.68%，女性好友43.17%，还有少部分没有填写性别的，基本上算是一半一半，可以说是十分均衡了。



## 好友签名分析

好友签名是我们获取到的数据中最为丰富的文本信息。

毕竟，我们大部分人都会通过签名来装个文艺，提升一下逼格，比如“缟仙扶醉跨残虹”

或者来个搞笑的，比如说“大家好，我是性感的贝贝”。

当然也少不了秀恩爱的，比如“爱你，只在你在身边都是幸福”

对好友签名进行分析，这里主要使用了 `jieba`，`snownlp`，`wordcloud` 等python包。

首先，对签名文本进行清洗，因为签名文本中含有emoji表情以及其他一些不能识别的字符，因此先去掉emoji，span等字符。然后，使用 `jieba` 对个性签名进行分词，使用 `snownlp` 进行词性的情感分析。

```
signatures = ''
emotions = []
for friend in friends:
    signature = friend['Signature']
    if signature != None:
        signature = signature.strip().replace("span", "").replace("class",
        "").replace("emoji", "") #去除无关数据
        signature = re.sub(r'1f(\d.+)', "", signature)

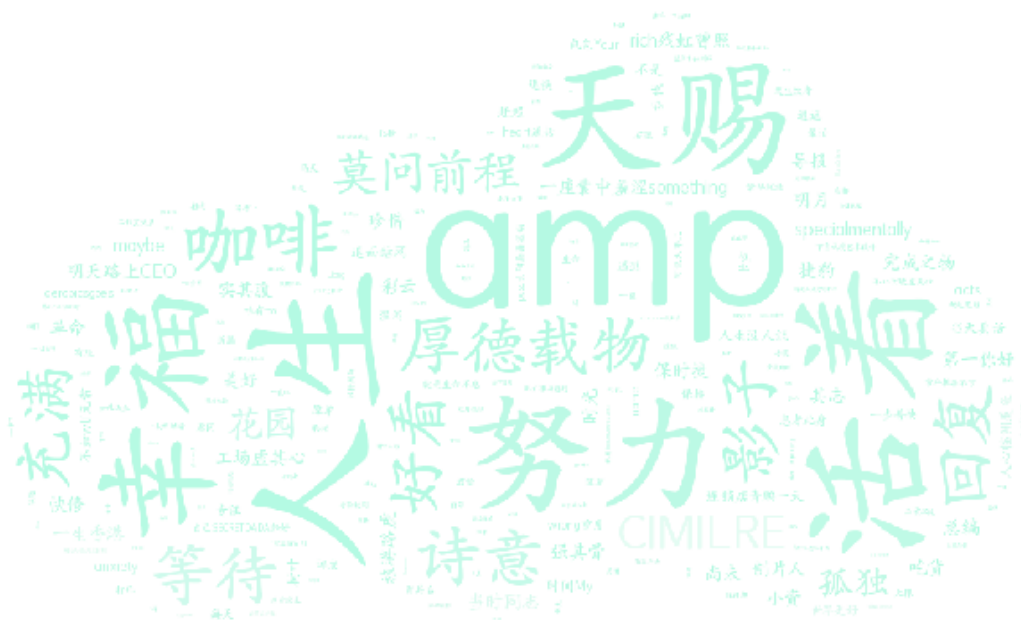
    if len(signature) > 0:
        nlp = snownlp.SnowNLP(signature)
        emotions.append(nlp.sentiments)
        signatures += " ".join(jieba.analyse.extract_tags(signature, 5)) #
```

关键字提取

然后开始绘制词云

```
back_coloring = np.array(Image.open("1.jpg"))#图片可替换你自己的图片
word_cloud2 = WordCloud(font_path = 'zt.ttf',
                          background_color = 'white',
                          max_words = 1200,
                          mask = back_coloring,
                          margin = 15)
word_cloud2.generate(signatures)
image_colors = ImageColorGenerator(back_coloring)
plt.figure(figsize=(6,5),dpi=160)
plt.imshow(word_cloud2.recolor(color_func=image_colors))
plt.axis("off")
plt.show()
word_cloud2.to_file("signatures.jpg")
```

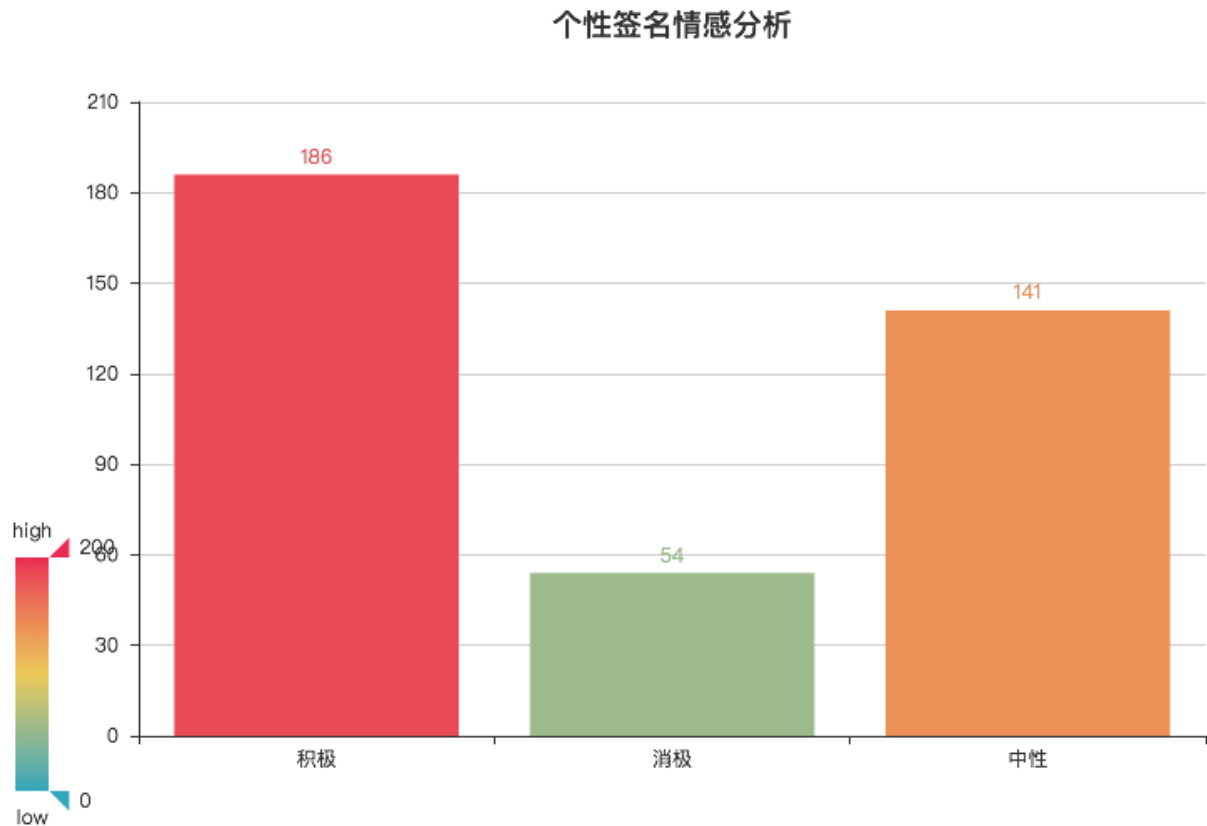
可以看到，词云中比较明显的包括“努力”，“人生”，“幸福”等等。正可谓是物以类聚人以群分，正能量的我也拥有正能量的朋友（大误）。



再看看 snownlp 分析好友签名中的情感倾向。代码如下：

```
count_positive = len(list(filter(lambda x:x>0.66,emotions)))#大于0.66为积极
count_negative = len(list(filter(lambda x:x<0.33,emotions)))#小于0.33为消极
count_neutral = len(list(filter(lambda x:x>=0.33 and x <= 0.66,emotions)))
value=[count_positive,count_negative,count_neutral]
att_attr=['积极','消极','中性']
bar=Bar('个性签名情感分析',title_pos='center',width=800,height=500)
bar.add(' ', att_attr, value, visual_range=[0, 200], is_visualmap=True, is_label_show=True)
bar
```

其中积极和中性的占据了大部分，消极的只占据了一部分，这个分析结果跟词云给我们的印象也是吻合的。



## 好友头像分析

`itchat` 返回的数据中包括好友头像的url，我们可以通过url将好友头像下载下来，再进行分析。

使用如下代码可以下载所有的好友头像保存到本地。

```
basePath = os.path.abspath('.')
baseFolder = basePath + '\\HeadImages\\'
if(os.path.exists(baseFolder) == False):
    os.makedirs(baseFolder)
image_tags = ''
for index in range(1,2):
    friend = friends[index]
    imgFile = baseFolder + '\\Image%s.jpg' % str(index)
    imgData = itchat.get_head_img(userName = friend['UserName'])
    if(os.path.exists(imgFile) == False):
        with open(imgFile,'wb') as file:
            file.write(imgData)
```

下载完所有的好友头像后，我们就能够对头像图片进行分析了。

由于这种图像分析涉及到了人脸识别等内容。为了方便起见，我们这里直接使用造好的轮子——百度人脸识别。

首先需要安装百度人脸识别sdk，直接输入以下命令即可：

```
pip install baidu-aip
```

然后，需要在百度应用里申请开发者账号并创建应用，拿到APP\_ID, API\_KEY以及SECRET\_KEY。

接下来就可以在python中调用百度人脸识别的服务了。

为了隐私起见，小编在这里就不用微信好友头像来做演示了，而是使用一张明星合影来展示，原图如下：



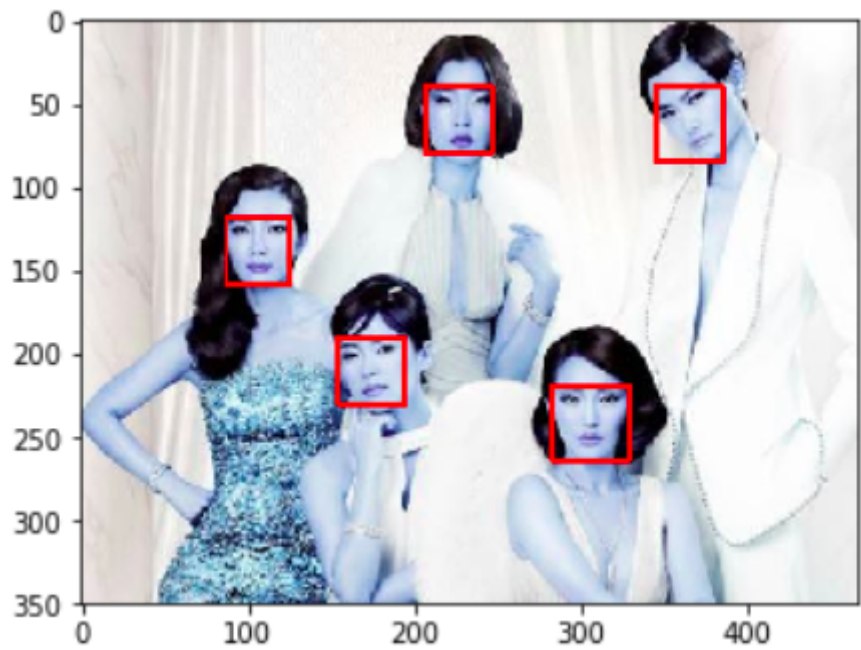
百度人脸识别服务可以识别出图中的人脸，并返回相应的人脸信息。代码如下：

```
from aip import AipFace
import base64
APP_ID = "your app_id"
API_KEY = "your api_key"
SECRET_KEY = "your secret_key"
client = AipFace(APP_ID, API_KEY, SECRET_KEY) #初始化aipface对象
filepath = "your own filepath"
with open(filepath, "rb") as fp:
    base64_data = base64.b64encode(fp.read())
image = str(base64_data, 'utf-8')
imageType = "BASE64"
options = {}
options["face_field"] = "age,gender,beauty,expression"
options["max_face_num"] = 10
options["face_type"] = "LIVE"
result = client.detect(image, imageType, options)
```



通过以上代码，我们就能够识别出图片中的人脸位置，还可以根据 `option` 选项，可以选择识别返回人脸的年龄，性别，颜值，表情，人种等方面的信息。

我们先来根据返回的人脸位置信息，用`opencv`模块作图标识出人脸位置：



返回的信息中，即 `result` 中的信息，小编选取了年龄，性别，颜值，和表情这四个维度。其中，年龄和颜值都是确定的值，表情和性别均为一个值以及对应的概率。

	age	beauty	expression	face_probability	gender	location
0	24	73.86	{'type': 'none', 'probability': 1}	1	{'type': 'female', 'probability': 1}	{'left': 282.64, 'top': 219.16, 'width': 46, 'h...
1	23	69.61	{'type': 'none', 'probability': 1}	1	{'type': 'female', 'probability': 1}	{'left': 206.05, 'top': 39.85, 'width': 44, 'h...
2	25	53.35	{'type': 'none', 'probability': 1}	1	{'type': 'female', 'probability': 0.94}	{'left': 344.09, 'top': 53.01, 'width': 46, 'h...
3	23	72.97	{'type': 'none', 'probability': 1}	1	{'type': 'female', 'probability': 1}	{'left': 154.37, 'top': 195.27, 'width': 46, 'h...
4	24	75.58	{'type': 'none', 'probability': 1}	1	{'type': 'female', 'probability': 1}	{'left': 86.75, 'top': 119.04, 'width': 43, 'h...

虽然粉丝经常为idol们的颜值高低battle到天荒地老，也有人说“各花入各眼”，不过可以看到，在五位女明星中，百度人脸识别对李冰冰的颜值作出了最高评分。

另外`expression`的判定上，所有人的表情均为`none`，并且概率都为1，果然是时尚杂志都偏爱面无表情的高冷范儿么。



## 最后

用python对微信好友进行数据分析就告一段落。事实上 `itchat` 有趣的应用还不只这一些，还可以利用它做聊天机器人，文件的定时发送等等。

同样的，文中所使用的其他包都有强大应用。希望这篇文章能给大家一些小小的启发，比芯～