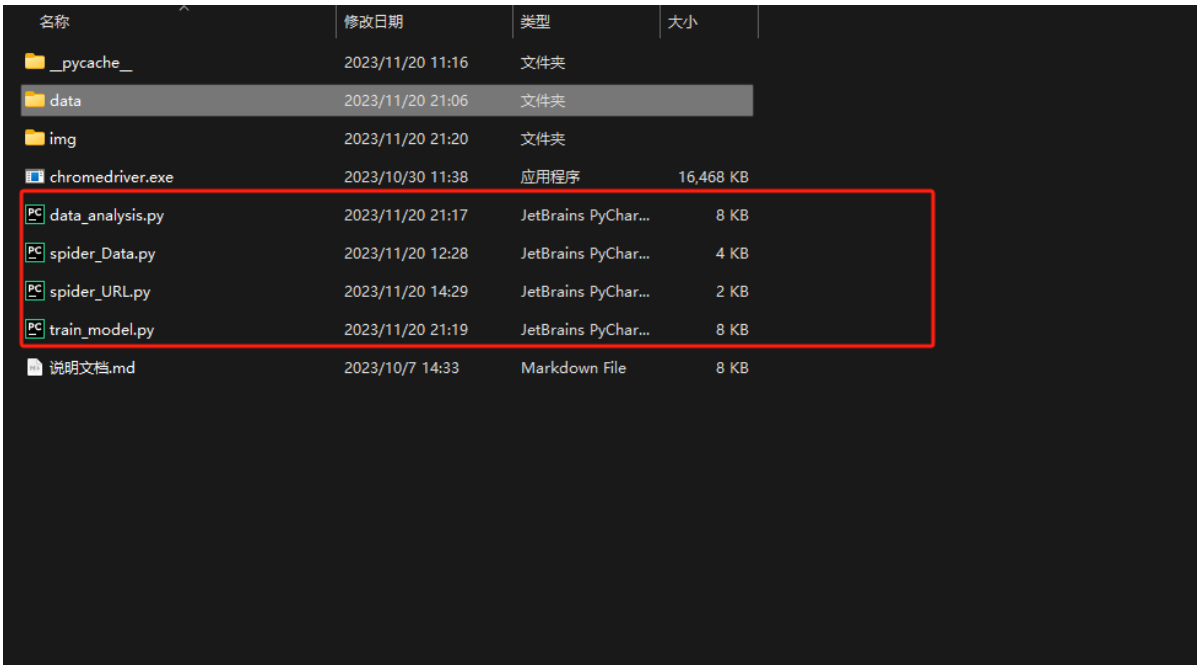
该项目为微博爬虫以及数据分析说明文档

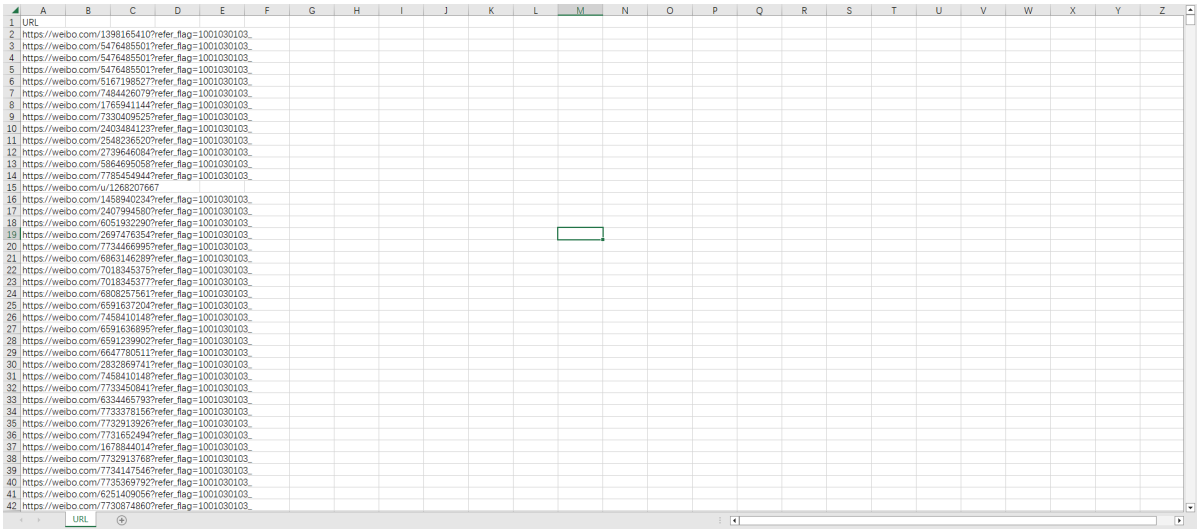
一共为4个代码组成:



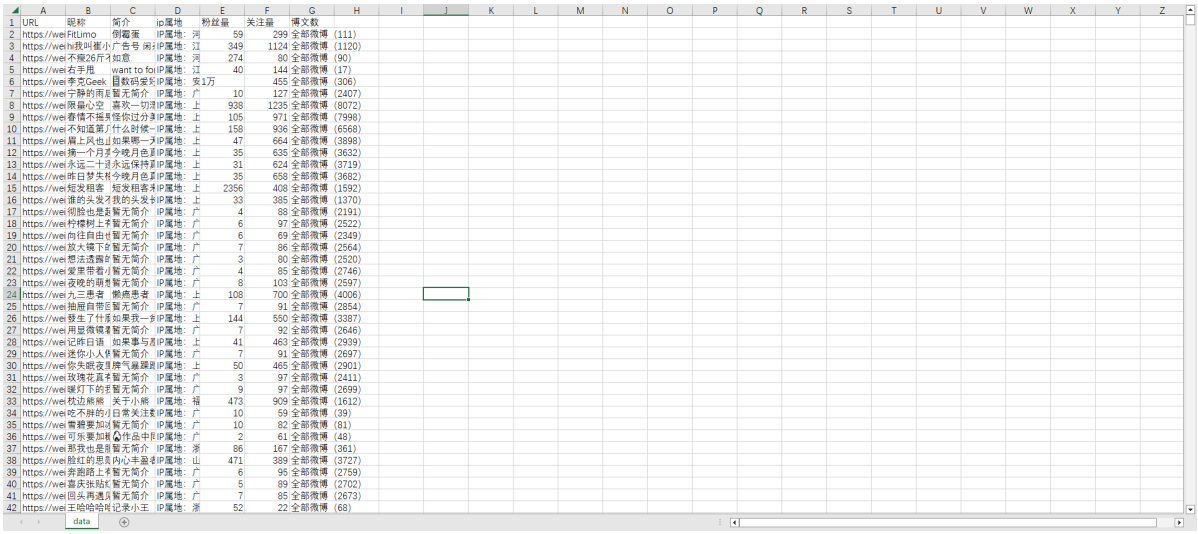
先来说说爬虫部分

爬虫部分分为spider_URL和spider_Data

spider_URL就是去爬取URL的，就是通过输入相关的热词，然后去爬取实时的用户对应的url



接着通过spider_Data去模拟登陆然后去爬取相关的用户信息数据



URL	昵称	简介	ip属地	粉丝量	关注量	博文数
https://wei	FitLimo	倒霉蛋	IP属地：河	59	299	全部微博（111）
https://wei	hi我叫崔小	广告号 闲	IP属地：江	349	1124	全部微博（1120）
https://wei	不瘦26斤才如意		IP属地：河	274	80	全部微博（90）
https://wei	右手甩	want to fo	IP属地：江	40	144	全部微博（17）
https://wei	李艾Geek	目权码爱好	IP属地：安	1万+	455	全部微博（306）
https://wei	宁静的雨点	暂无简介	IP属地：广	10	127	全部微博（2407）
https://wei	限量心空	喜欢一切浪	IP属地：上	938	1235	全部微博（8072）
https://wei	看情不揭易	怪你过分美	IP属地：上	105	971	全部微博（7998）
https://wei	不知道第几	什么时候一	IP属地：上	158	936	全部微博（6568）
https://wei	肩上风也止	如果哪一天	IP属地：上	47	664	全部微博（3898）
https://wei	傅一个月牙	今晚月色真	IP属地：上	35	635	全部微博（3632）
https://wei	永远二十岁	永远保持真	IP属地：上	31	624	全部微博（3719）
https://wei	昨日梦失村	今晚月色真	IP属地：上	35	658	全部微博（3682）
https://wei	短发租客	短发租客来	IP属地：上	2356	408	全部微博（1592）
https://wei	谁的头发才	我的头发长	IP属地：上	33	385	全部微博（1370）
https://wei	彻脸也是美	暂无简介	IP属地：广	4	88	全部微博（2191）
https://wei	柠檬树上不	暂无简介	IP属地：广	6	97	全部微博（2522）
https://wei	向往自由社	暂无简介	IP属地：广	6	69	全部微博（2349）
https://wei	放大镜下的	暂无简介	IP属地：广	7	86	全部微博（2564）
https://wei	想法透露的	暂无简介	IP属地：广	3	80	全部微博（2520）
https://wei	堂里带着小	暂无简介	IP属地：广	4	85	全部微博（2746）
https://wei	夜晚的明亮	暂无简介	IP属地：广	8	103	全部微博（2597）
https://wei	九三患者	躺癌患者	IP属地：上	108	700	全部微博（4006）
https://wei	地握自助反	暂无简介	IP属地：广	7	91	全部微博（2854）
https://wei	登生了什么	如果我一直	IP属地：上	144	550	全部微博（3387）
https://wei	用显微镜看	暂无简介	IP属地：广	7	92	全部微博（2646）
https://wei	记昨日语	如果事与愿	IP属地：上	41	463	全部微博（2939）
https://wei	迷你小人怪	暂无简介	IP属地：广	7	91	全部微博（2697）
https://wei	你失眠夜是	脾气暴躁是	IP属地：上	50	465	全部微博（2901）
https://wei	玫瑰花真不	暂无简介	IP属地：广	3	97	全部微博（2411）
https://wei	缓灯下的反	暂无简介	IP属地：广	9	97	全部微博（2699）
https://wei	枕边熊熊	关于小熊	IP属地：福	473	909	全部微博（1612）
https://wei	吃不胖的小	日常关注套	IP属地：广	10	59	全部微博（39）
https://wei	雪碧要加冰	暂无简介	IP属地：广	10	82	全部微博（81）
https://wei	可乐要加橘	作品中国	IP属地：广	2	61	全部微博（48）
https://wei	那夜也是刮	暂无简介	IP属地：浙	86	167	全部微博（361）
https://wei	脸红的思凡	内心丰富多	IP属地：山	471	389	全部微博（3727）
https://wei	奔跑路上不	暂无简介	IP属地：广	6	95	全部微博（2759）
https://wei	喜庆张贴红	暂无简介	IP属地：广	5	89	全部微博（2702）
https://wei	回头再遇见	暂无简介	IP属地：广	7	85	全部微博（2673）
https://wei	王哈哈哈哈	记录小王	IP属地：浙	52	22	全部微博（68）

data

接着我们爬取相关信息之后，我们开始数据清洗工作以及数据建模工作

也就是train_model，我们先把相关的数据特征收集下来的，无用的则删除掉，这里根据提供的数据集进行训练

接着我们再去做数据清洗工作，把数字的归为数字，文字的进行筛选，去掉无用词以及机械压缩

```
#ip处理
def ip_chuli1(x):
    x1 = str(x).split(": ")
    return x1[-1]

#ip处理
def ip_chuli2(x):
    provinces = ['安徽', '澳门', '北京', '重庆', '福建', '甘肃', '广东', '广西', '贵州', '海南', '河北', '黑龙江', '河南', '湖北', '湖南', '江苏', '江西',
                 '吉林', '辽宁', '内蒙古', '宁夏', '青海', '山东', '上海', '山西', '陕西', '四川', '台湾', '天津', '西藏', '香港', '新疆', '云南', '浙江']

    if x in provinces:
        #表示属于国内
        return 1
    else:
        #表示属于外国
        return 0


def jianjie_processing(x):
    x1 = str(x)
    if x1 == '暂无简介':
        #表示低活跃用户
        return 0
    else:
        #表示高活跃用户
        return 1

#数字处理
def number_data(x):
    try:
        x1 = str(x)
        number = re.findall(r'\d+', x1)
```

```
        return number[0]
    except:
        return 0

#粉丝处理
def fan_chuli(x):
    try:
        x1 = int(x)
        return x1
    except:
        if '万' in x:
            x1 = str(x).replace('万','')
            x1 = float(x1) * 10000
            return x1

#停用词函数
stop_words = []
with open("./data/stopwords_cn.txt", 'r', encoding='utf-8') as f:
    lines = f.readlines()
    for line in lines:
        stop_words.append(line.strip())


#去掉标点符号，以及机械压缩
def preprocess_word(word):
    word1 = str(word)
    word1 = re.sub(r'转发微博', '', word1)
    word1 = re.sub(r'#\w+#', '', word1)
    word1 = re.sub(r'【.*?】', '', word1)
    word1 = re.sub(r'@[\w]+', '', word1)
    word1 = re.sub(r'[a-zA-Z]', '', word1)
    word1 = re.sub(r'\.\d+', '', word1)
    return word1


def emjio_tihuan(x):
    x1 = str(x)
    x2 = re.sub('(\[.*?\])', "", x1)
    x3 = re.sub(r'@[\w\u2E80-\u9FFF]+:?|\[\w+\]', '', x2)
    x4 = re.sub(r'\n', '', x3)
    return x4


def is_all_chinese(strs):
    for _char in strs:
        if not '\u4e00' <= _char <= '\u9fa5':
            return False
    return True


# 定义机械压缩函数
def yasuo(st):
    for i in range(1, int(len(st) / 2) + 1):
        for j in range(len(st)):
            if st[j:j + i] == st[j + i:j + 2 * i]:
                k = j + i
```

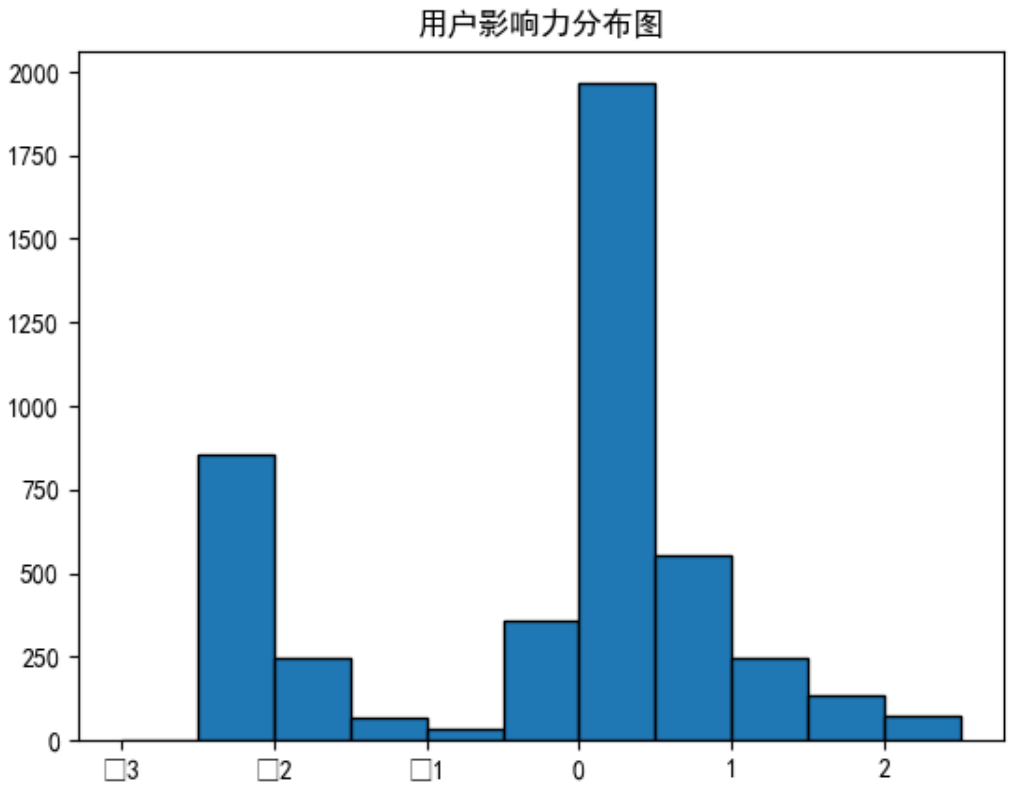
```
                while st[k:k + i] == st[k + i:k + 2 * i] and k < len(st):
                    k = k + i
                st = st[:j] + st[k:]
    return st


def get_cut_words(content_series):
    # 读入停用词表
    # 分词
    word_num = jieba.lcut(content_series, cut_all=False)

    # 条件筛选
    word_num_selected = [i for i in word_num if i not in stop_words and len(i) >=
2 and is_all_chinese(i) == True]

    return ' '.join(word_num_selected)


def null_paichu(x):
    x1 = str(x)
    if len(x1) != 0:
        return x1
    else:
        return np.NAN


data = pd.read_csv('./data/data.csv')
data['ip属地'] = data['ip属地'].apply(ip_chuli1)
data['属地分类'] = data['ip属地'].apply(ip_chuli2)
data['简介分类'] = data['简介'].apply(jianjie_processing)
data['博文数'] = data['博文数'].apply(number_data)
data['粉丝量'] = data['粉丝量'].apply(fan_chuli)
data['关注量'] = data['关注量'].apply(fan_chuli)
data['关注量'] = data['关注量'].astype('int')
data['粉丝量'] = data['粉丝量'].astype('int')
data['博文数'] = data['博文数'].astype('int')
data['属地分类'] = data['属地分类'].astype('int')
data['简介分类'] = data['简介分类'].astype('int')
```

我们对处理好的数据集进行规范化处理，也就是z-score

把这些数据进行相加，生成影响力，根据影响力来进行判断用户的类别

```
data1 = data[['属地分类','简介分类','粉丝量','关注量','博文数']]

#将数据进行[0,1]规范化
scaled_x = preprocessing.scale(data1)

list_scaler = []
for m in scaled_x:
    number1 = m[0] + m[1] + m[2] + m[3] + m[4]
    list_scaler.append(number1)

data['影响力'] = list_scaler
plt.rcParams['font.sans-serif'] = ['SimHei']
```

```
plt.hist(list(data['影响力']), bins=np.arange(-3, 3, 0.5), edgecolor="black")
plt.title('用户影响力分布图')
plt.savefig('./img/用户影响力分布图.png')
plt.show()
```



根据该图，我们分四个类别，分类规则如下：

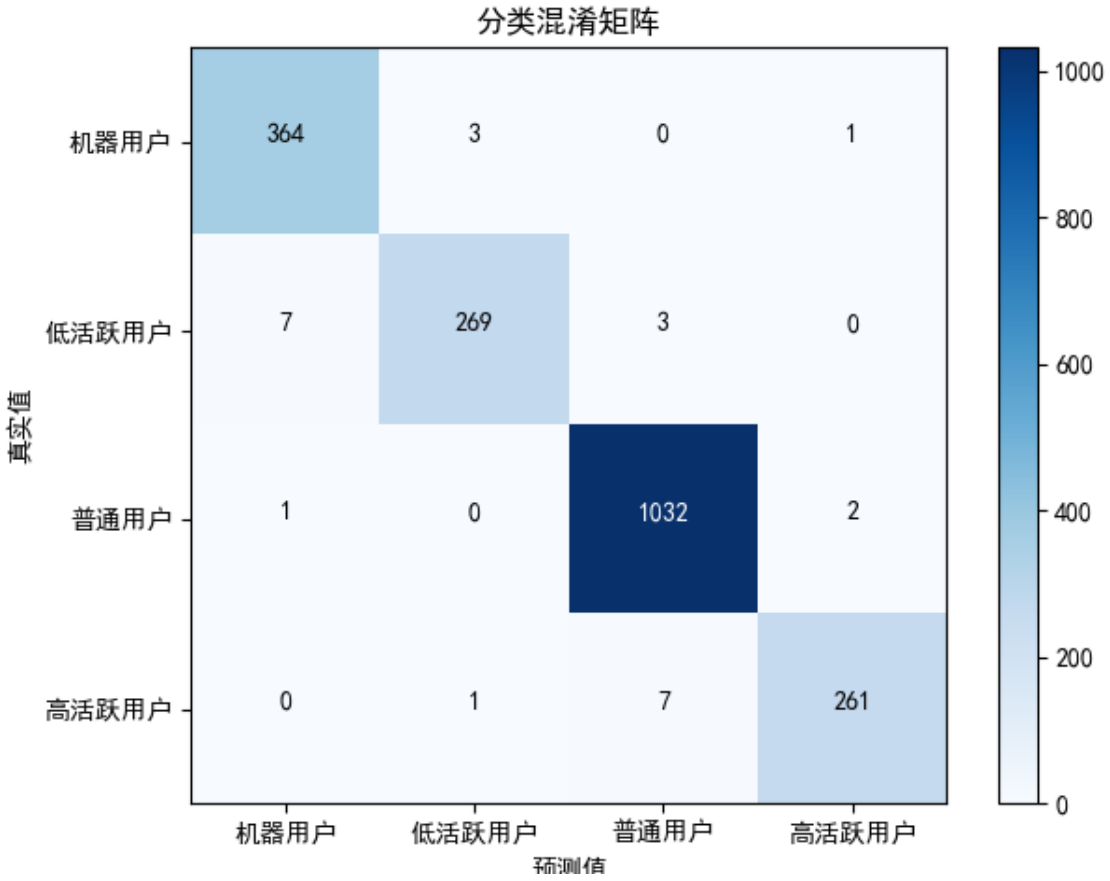
```
在影响力小于-2的的时候，归为机器用户，
当影响力大于-2小于0的时候，归为低活跃用户
当影响力大于0小于1的时候，归为普通用户
当影响力大于1的时候，归为高活跃用户

代码如下：

def class_type(x):
    x1 = float(x)
    if x1 <= -2:
        return 0
    elif -2 < x1 <= 0:
        return 1
    elif 0 < x1 <= 1:
        return 2
    else:
        return 3

data['用户分类'] = data['影响力'].apply(class_type)
```

接着 我们做好用户分类之后，我们来做随机森林分类学习任务，代码如下：

```
# 分割训练集和测试集
X_train, X_test, y_train, y_test = train_test_split(data1, list(data['用户分类']),
test_size=0.4, random_state=42)

# 利用网格搜索进行超参数调优
param_grid = {'n_estimators': [50, 100, 200],
              'max_features': ['sqrt', 'log2'],
              'max_depth' : [4, 5, 6, 7, 8],
              'criterion' :['gini', 'entropy']
             }
rfc = RandomForestClassifier(random_state=1)
CV_rfc = GridSearchCV(estimator=rfc, param_grid=param_grid, cv = 5)
CV_rfc.fit(X_train, y_train)

# 输出最佳参数
print(CV_rfc.best_params_)

# 使用最佳参数重建模型
rfc1=RandomForestClassifier(random_state=1, n_estimators=100,
max_features='sqrt', max_depth=8, criterion='entropy')
rfc1.fit(X_train, y_train)

# 进行预测
pred = rfc1.predict(X_test)

# 输出模型评估结果
print('Accuracy score: ', accuracy_score(y_test,pred))
print('Confusion Matrix: \n', confusion_matrix(y_test,pred))
```

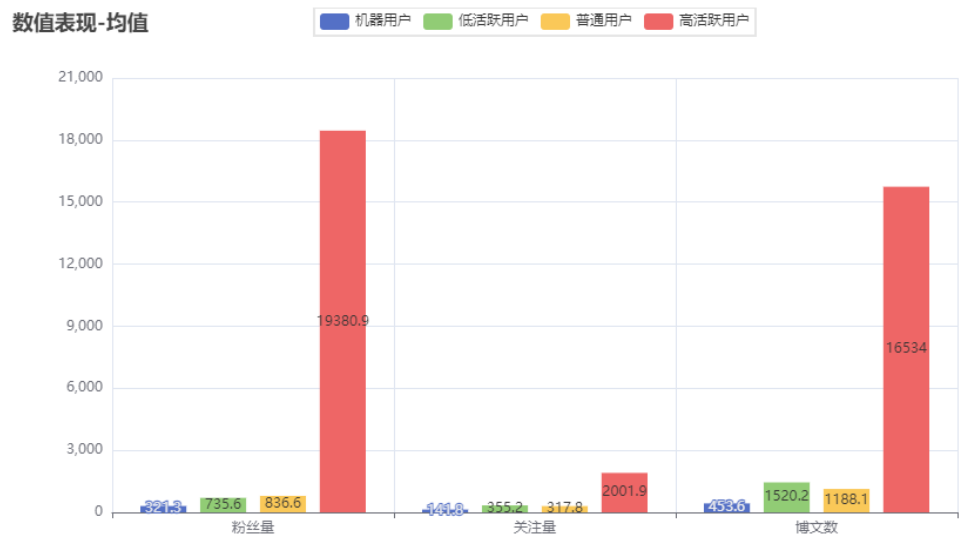
结果呈现：

```
              precision    recall  f1-score   support

           0       0.98      0.99      0.98       368
           1       0.99      0.96      0.97       279
           2       0.99      1.00      0.99      1035
           3       0.99      0.97      0.98       269

    accuracy                           0.99      1951
   macro avg       0.99      0.98      0.98      1951
weighted avg       0.99      0.99      0.99      1951
```

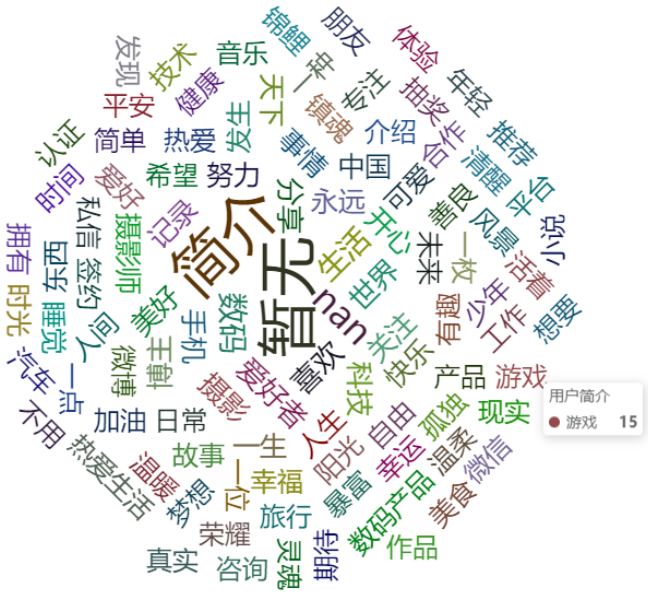


最后我们依次来介绍这些可视化的含义：

该图是代表不同用户，他们的粉丝量、关注量、博文量的平均数，是统计该类型的全部数值，然后求的平均数

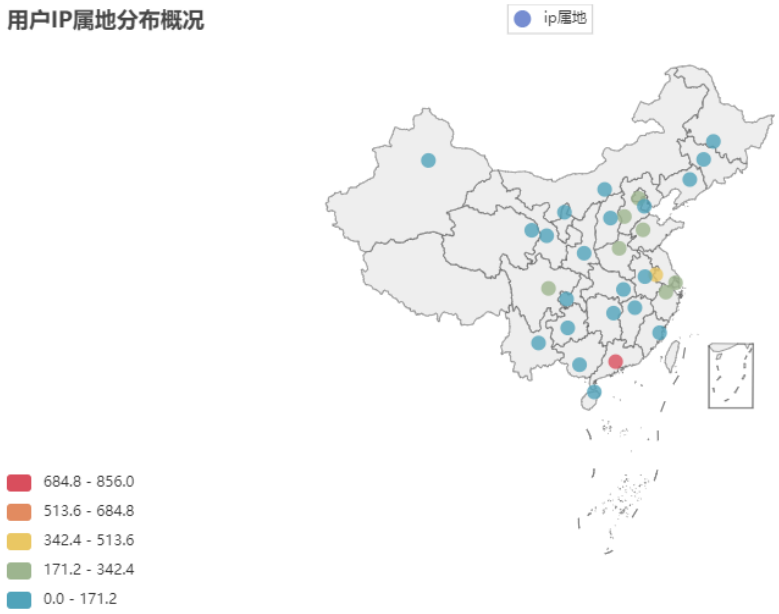


简介热词分析



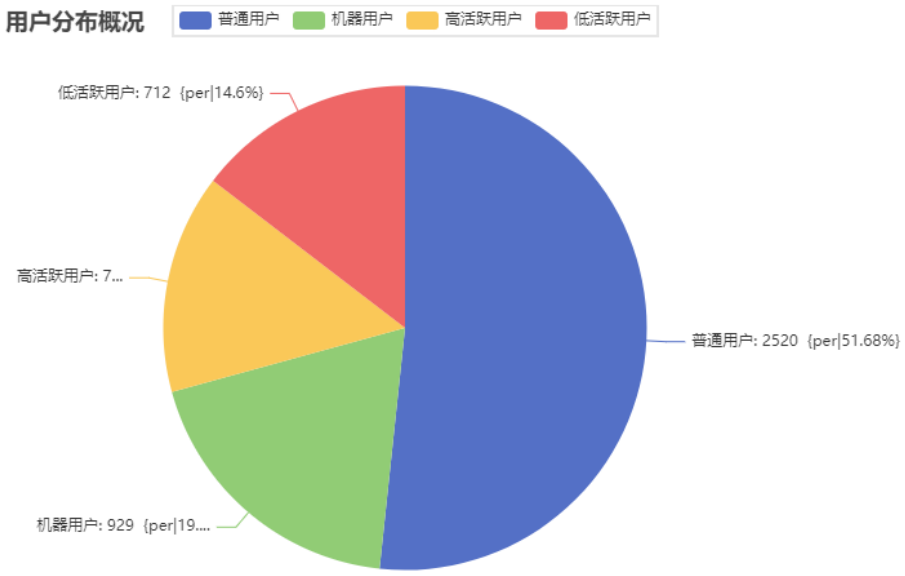
这个是基于用户简介里面做的词云图，其目的是看看用户平时主要用哪些词语来组成简历，显而易见，大多数用户都是暂无简介的

用户IP属地分布概况



中国地图，这里是不同用户，他们归属地的分布数量，地图，根据左下角的颜色，可以区分数量的大小，数量越大，则对应的颜色也会发生变化

饼图的话，就是不同用户的一个占比状况



以上便是整个项目的解析，如果有什么不懂的，可以在群里讨论