实验 5-1 清洗微博数据

建议课时：30 分钟

一、 实验目的

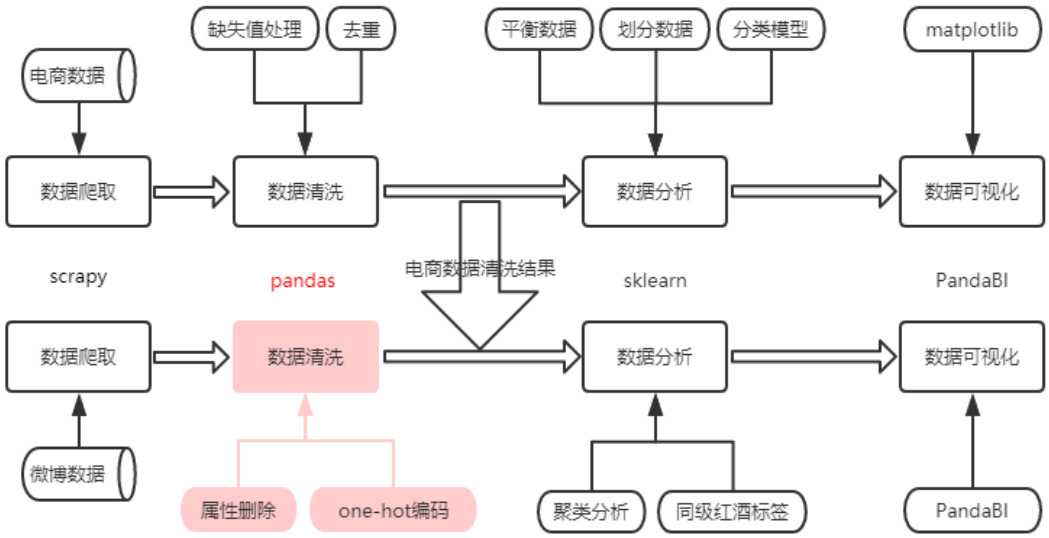
- 了解 pandas 的常用函数
- 了解数据清洗的基础方法

二、 实验环境

Python3 开发环境，第三方包有 pandas

三、 实验步骤

本节处理的内容有：



1. 读取数据

读取数据的方式和第二章的读取数据一致，都是采用 json.loads() 函数读成字典类型。

代码如下：

```
import csv
import json
csvfile = open('微博红酒.csv', 'r')
reader = csv.reader(csvfile)#读取到的数据是将每行数据当做列表返回的
rows = []#用来存储解析后的没条数据
for row in reader:
    row_str = ",".join(row)#row为list类型需转为str，该数据变为字典型字符串
    row_dict = json.loads(row_str)

    #将每行数据中嵌套字典拆开存储到列表中
    newdict = {}
    for k in row_dict:
        if type(row_dict[k]) == str:#将键值对赋给新字典
            newdict[k] = row_dict[k]
        elif type(row_dict[k]) == dict:#若存在嵌套字典，将该字典中的key和value作为属性和属性值
            newdict.update(row_dict[k])
    rows.append(newdict)
```

可以得到 24611 条数据，示例如下：


```
{'id': '10099536052',
 'url': 'https://weibo.com/1823411197/ghE0Li',
 'keyword': '莱茵黑森 红酒',
 'post_time': '2019-05-15 14:25:15',
 'post_text': '今晚有红衫鱼  蚝仔烙 粉丝带子，如果把这支红酒换成加拿大冰酒或者99年莱茵黑森白葡就完美啦哈哈哈哈超有满足感的一餐，
来吧 ',
 'post_source': '彩信',
 'description': '还是很想你……dbx',
 'gender': 'f',
 'name': '舞夜eva',
 '昵称': '舞夜eva',
 '所在地': '广东 广州',
 '性别': '女',
 '生日': '11月4日',
 '博客': '',
 '个性域名': 'https://weibo.com/evasdream',
 '简介': '还是很想你……dbx',
 '注册时间': '2010-09-25',
 '公司': '中外|职位：设计部',
 '标签': '小蝎子|设计师|化妆师|大胃王|小肉丸'}
```


处理搜索关键词，即 keyword 字段，处理红酒品牌的中英文表示，代码同上一章节类似，如下所示：

```
# 处理红酒名称（统一规范中文/英文的格式）
#去keyword中"红酒"字符
df['keyword']=df['keyword'].str.split('红酒').str[0]

#整理红酒品牌
brand = pd.read_csv("./data/wine_brand.csv",header=None,encoding="utf-8")
brands =[]
for k1,k2 in zip(list(brand[0]),list(brand[1])):
    if pd.isnull(k2):
        brands.append(k1)
    else:
        brands.append(k1+"/"+k2)

# 品牌替换
def modify_keywords(w, lists):
    for b in lists:
        if w.strip() in b:
            return b
    return w.strip()


df['keyword'] = df['keyword'].apply(modify_keywords, lists =brands)
df['keyword']=df['keyword'].str.lower()#转化为小写

print(df["keyword"].head(10))
```

接下来我们整理想要的属性，包含"keyword","post_time","所在地","性别","生日","gender"这六种属性，保存临时处理文件，用以观察。

2. 处理性别

用 df.count 函数查看各维度的非缺失值，结果如下：

```
keyword      24611
post_time    24611
所在地          18814
性别           18814
生日           12250
gender       24611
dtype: int64
```

可见 gender 字段比性别可行度高，使用 df.groupby 验证 gender 中是否存在异常值

```
df.groupby("gender").size().sort_values(ascending=False)
```

```
gender
m    12577
f    12034
dtype: int64
```

即对性别的处理，保留 gender 字段即可。

3. 处理年龄

据观察，出生字段有人写的是日期，有人写的是星座，有人写的是年月日

因而采用提取年份前面的数字再与当前年份相减得到年龄值。

```
# 处理年龄
# df.count()
def age(x):
    index = str(x).find('年')
    if index == -1 :
        return 9999
    else:
        res = 2019-int(str(x)[:index])
        # 超过100岁，视为不合理
        if res>100:
            return 9999
        else:
            return 2019-int(str(x)[:index])

df['age'] = df['生日'].map(age)
df.drop("生日",axis=1,inplace=True)
```

结果如下所示：

```
df.head(20)
```

	keyword	post_time	所在地	gender	age
0	卓林/zonin	2019-05-15 14:29:29	NaN	m	9999
1	怡园酒庄/grace vineyard	2019-05-15 14:25:30	香港 其他	f	9999
2	莱茵黑森	2019-05-15 14:25:15	广东 广州	f	9999
3	长城/greatwall	2019-05-15 14:24:20	上海 浦东新区	m	34
4	奥兰	2019-05-15 14:15:26	北京	f	9999
5	华东/huadong	2019-05-15 14:27:43	山东 青岛	f	9999
6	黄尾袋鼠/yellow tail	2019-05-15 14:25:05	辽宁 大连	m	9999
7	黄尾袋鼠/yellow tail	2019-05-15 14:25:05	上海	m	9999
8	奥兰	2019-05-15 14:15:26	广东 深圳	f	9999
9	安徒生	2019-05-15 14:27:42	NaN	m	9999
10	安徒生	2019-05-15 14:27:42	湖南 长沙	m	9999

4. 处理地区

据观察，所在地的取值有三种，分别是其他，大地区，大地区加小地区

处理方法：

- 替代 缺失值 为 其他
- 延申出两维数据，分别表示为大地区和小地区，没有小地区则表示为其他

代码如下：

```
# 处理地区
df['所在地'].fillna("其他",inplace=True)

def place(x):
    words = str(x).split(" ")
    if len(words) > 1:
        return words[0].strip(), words[1].strip()
    else:
        return words[0],"其他"

df["所在地"] = df["所在地"].map(place)
df['country'] = df["所在地"].apply(lambda x : x[0])
df['region'] = df['所在地'].apply(lambda x : x[1])
df.drop("所在地",axis = 1,inplace = True)
```

结果如下所示：

```
df.head(10)
```

	keyword	post_time	gender	age	country	region
0	卓林/zonin	2019-05-15 14:29:29	m	9999	其他	其他
1	怡园酒庄/grace vineyard	2019-05-15 14:25:30	f	9999	香港	其他
2	莱茵黑森	2019-05-15 14:25:15	f	9999	广东	广州
3	长城/greatwall	2019-05-15 14:24:20	m	34	上海	浦东新区
4	奥兰	2019-05-15 14:15:26	f	9999	北京	其他
5	华东/huadong	2019-05-15 14:27:43	f	9999	山东	青岛
6	黄尾袋鼠/yellow tail	2019-05-15 14:25:05	m	9999	辽宁	大连
7	黄尾袋鼠/yellow tail	2019-05-15 14:25:05	m	9999	上海	其他
8	奥兰	2019-05-15 14:15:26	f	9999	广东	深圳
9	安徒生	2019-05-15 14:27:42	m	9999	其他	其他

5. 处理发布时间

发布时间在一定程度上可以反映不同时间段对红酒品牌的关注度，可能在某一营销活动或文章之后，关注度有所提升，则可以认为这种营销活动和文章是有一定积极因素的。

观察 post_time 字段，发现均为 2019 年 5 月 15 日发布，由于数据量的局限性，此字段予以删除。学生们可以利用爬虫脚本爬取更多数据量，在此维度上做一些统计方法以辅助营销决策。

综上，微博的数据处理完毕，5.2 节将结合第三章的处理结果进行同级红酒的发现。