实验 2-1 爬取电商平台上各类红酒的属性信息

建议课时：30 分钟

一、 实验目的

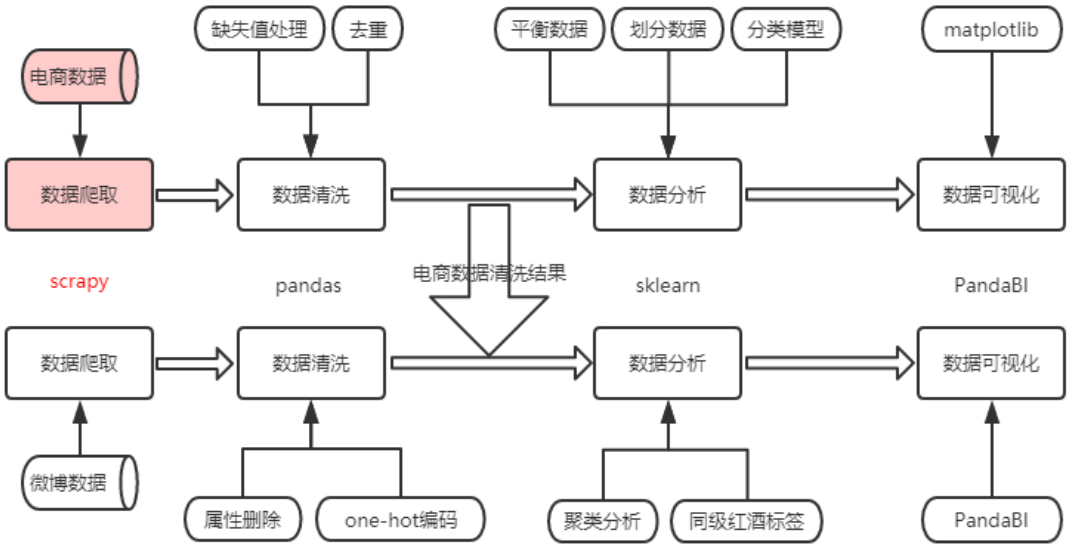
- 了解 scrapy 的常用函数
- 了解爬虫编写的流程

二、 实验环境

Python3 开发环境，第三方包有 scrapy，re

三、 实验步骤

本节处理的内容有：



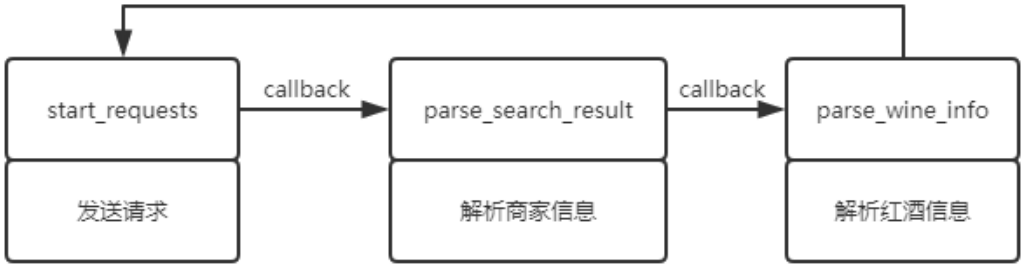
在某电商平台上的红酒描述信息如下：

商品介绍 | 规格与包装 | 售后保障 | 商品评价(54万+) | 本店好评商品

主体	产区	其它
	年份	以产品包装为准
	酒精度	12.5%vol
	保质期	3650天
	存储方法	避免阳光直射，于5℃~25℃干燥通风处卧放或倒放为宜。
	类别	红葡萄酒
	葡萄品种	其它
	甜度	干型
	口感	清新
	颜色	宝石红
	原产地	中国
	特性	普通餐酒
	国产/进口	国产
规格参数	容量	750ml
	包装	瓶装
	产品重量（kg）	7.98kg

包装清单　长城特酿3年解百纳干红葡萄酒 整箱装 750ml*6瓶 *1箱

我们需要通过编写 python 脚本进行爬取，下面选取核心代码讲解爬取京东红酒属性数据的逻辑流程。



1. 搜索关键词

根据红酒品牌列表加红酒字样作为搜索关键词，实现代码如下：

```
# 京东搜索链接（手机端）
jd_search_url = 'https://so.m.jd.com/ware/search._m2wq_list?keyword={kw}&page={page}&pagesize=10'
```

```
def start_requests(self):
    page = 1
    with open('./wine_project/data/红酒品牌.csv', 'r', encoding='utf-8') as f:
        csv_reader = csv.reader(f)
        for zh_name, en_name in csv_reader:
            print('++++++++++crawling:{}-{}'.format(zh_name, en_name))
            if zh_name.strip():
                wine_name = zh_name.strip() + ' 红酒'
                yield Request(jd_search_url.format(kw=wine_name, page=page), headers=self.headers,
                              meta={'kw': wine_name, 'page': page}, callback=self.parse_search_result)
            if en_name.strip():
                wine_name = en_name.strip() + ' 红酒'
                yield Request(jd_search_url.format(kw=wine_name, page=page), headers=self.headers,
                              meta={'kw': wine_name, 'page': page}, callback=self.parse_search_result)
```

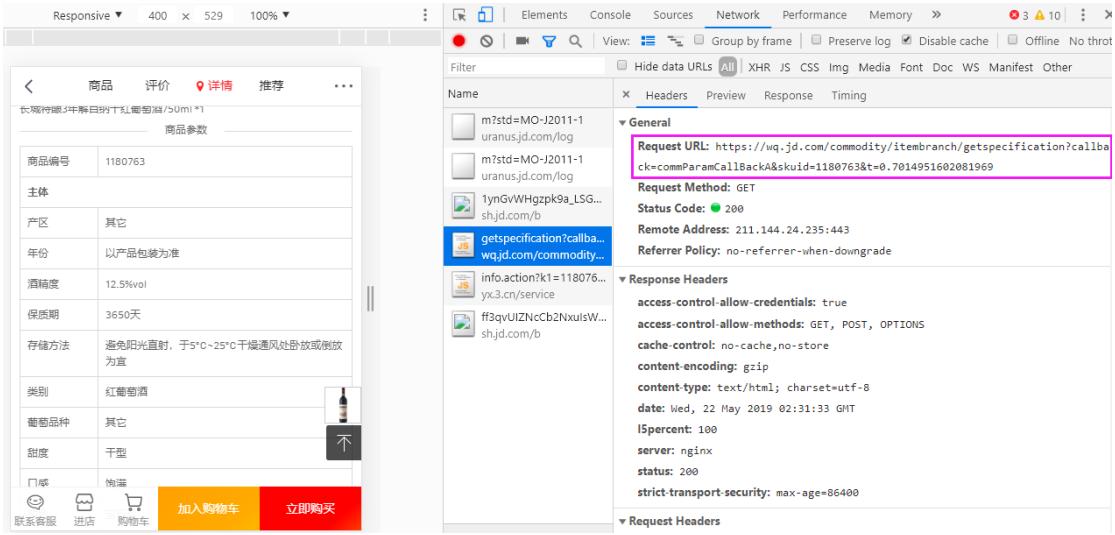
一个红酒品牌有中文和英文两种表达方式，代码中两种名称都要作为搜索关键词进行检索。

通过 scrapy 的 Request 创建京东数据获取对象，并且通过 yield 构造一个生成器函数，将获取的数据按需返回，防止因数据太多造成内存溢出等问题。

爬取的数据会通过 callback 函数回调，我们这里指定 parse_search_result 函数，该函数的任务是解析爬取的数据。

2. 获取红酒访问链接

通过浏览器的开发者模式，找到商品详情请求的 api，界面搜索结果如下：




```
# 京东红酒参数信息链接
jd_wine_info_url = 'https://wq.jd.com/commodity/itembranch/getspecification?callback=commParamCallBackA&skuid={skuid}&t=0.8241453845232118'
```

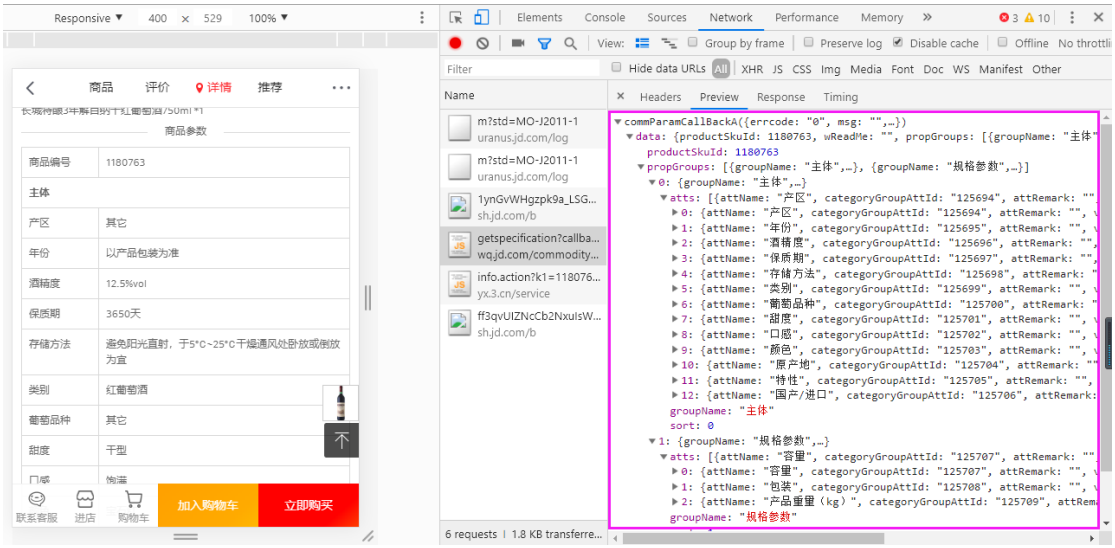
在 parse_search_result 函数的最后

```
yield Request(jd_wine_info_url.format(skuid=ret['sku_id']), headers=self.headers,
                                      meta=ret, callback=self.parse_wine_info)
```

回调了 parse_wine_info 函数解析红酒信息。

3. 明确解析字段

找到红酒各属性字段在返回值中的位置，层层解析得到相应字段，实现代码如下：



解析商家信息：

```
if 'data' in json_data and 'searchm' in json_data['data'] and json_data['data']['searchm']['Paragraph']:
    for item in json_data['data']['searchm']['Paragraph']:
        has_next_page = True
        ret = {}
        ret['name'] = item['Content']['warename']
        ret['sku_id'] = item['wareid']
        ret['id'] = item['wareid']
        ret['price'] = item['dredisprice']
        ret['shop_name'] = item['shop_name']
        ret['shop_id'] = item['shop_id']
        ret['url'] = 'https://item.jd.com/{}.html'.format(item['wareid'])
```

解析红酒信息：

```
def parse_wine_info(self, response):
    """解析红酒属性信息"""
    ret = response.meta
    matcher = wine_info_ptn.findall(response.text)
    if not matcher:
        print('**********get wine info error')
        return
    json_data = json.loads(matcher[0])

    # 红酒属性信息，这里直接将属性的中文作为key，方便理解！！
    prop_dict = {}
    for prop_group in json_data['data']['propGroups']:
        for attr in prop_group['atts']:
            prop_dict[attr['attName']] = '|'.join(attr['vals'])
    ret['prop'] = prop_dict
    yield ret
```

四、 实验结果

示例如下：


```
{
    "name": "长城（GreatWall）红酒 精选级赤霞珠干红葡萄酒 整箱装 750ml*6 瓶",
    "sku_id": "1088839",
    "price": "209.00",
    "shop_name": "长城葡萄酒京东自营旗舰店",
    "shop_id": "1000004719",
    "url": "https://item.jd.com/1088839.html",
    "keyword": "长城 红酒",
    "prop": {
        "产区": "其它",
        "年份": "以产品包装为准",
        "酒精度": "12.5%vol",
        "保质期": "3650 天",
        "存储方法": "避免阳光直射，于 5℃~25℃干燥通风处卧放或倒放为宜",
        "类别": "红葡萄酒",
        "葡萄品种": "赤霞珠（Cabernet Sauvignon）",
        "甜度": "干型",
        "口感": "饱满",
        "颜色": "宝石红",
        "原产地": "中国",
        "特性": "普通餐酒",
        "国产/进口": "国产",
        "容量": "750ml",
        "包装": "整箱",
        "产品重量（kg）": "7.99kg"
    }
}
```

最终的爬取结果在 电商红酒.csv 文件中。