实验 5-2 利用聚类发现同级红酒

建议课时：20 分钟

一、 实验目的

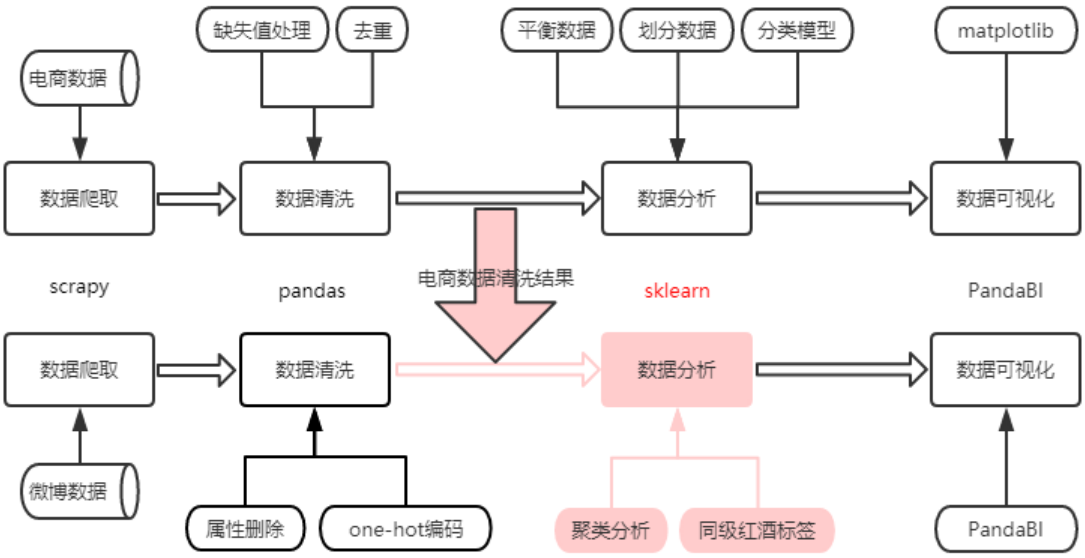
- 了解 pandas 的常用函数
- 了解聚类方法的处理流程

二、 实验环境

Python3 开发环境，第三方包有 pandas，sklearn

三、 实验步骤

本节处理的内容有：



在之前观察数据时，可以发现很多离散数据都含有 6-8 个不同数据取值，所以此处我们将选择聚类中的 k 值为 6，查看聚类效果。

1. 读取之前为微博数据分析准备的数据

```
import pandas as pd
winedf = pd.read_csv("./prepare_for_weibo.csv")
```

2. one-hot 编码

查看 winedf 的 dtypes：

```
winedf.dtypes
```

```
keyword                 object
price                  float64
产区                      object
原产地                     object
口感                      object
国产/进口                    object
特性                      object
甜度                      object
颜色                      object
冰酒/贵腐/甜酒                  int64
白葡萄酒                    int64
果味葡萄酒                   int64
桃红葡萄酒                   int64
```

由于 kmeans 接受数值型数据，所以需要采用 one-hot 编码对 object 类别的数据进行转换

```
resdf = pd.get_dummies(winedf) #非数值型都转化为one-hot
```

3. 聚类

```
from sklearn.cluster import KMeans

kmodel = KMeans(n_clusters = 6,n_jobs=5)
kmodel.fit(resdf)         #训练模型
rs = kmodel.predict(resdf) #类别索引
winedf['label'] = rs
```

代码使用很简单，定好聚类数目即可，其他参数可见官方文档，也可在 jupyter 编辑器中利用？来查询源码和用法。

4. 同级红酒发现

看一下每个类别中数目较多的红酒类别：

```
winedf.groupby("label").agg({'price':['min','max','count']})
```

	price		
	min	max	count
label			
0	3.32	23040.0	3114
1	9.83	3399.0	3206
2	4.83	50000.0	4264
3	16.00	8800.0	206
4	78000.00	79560.0	2
5	3499.00	29800.0	163

每个类中的价格最小值和最大值，可见分类界限没那么明确，造成这样的原因主要是两部分：一是聚类数目没有选择好，二是数据的准确度，之前的红酒价格是根据数据规范算出来的，但有可能提取的瓶数是错的，导致数据是不正确的，所以聚类上对品牌会造成影响。

但我们可以查看一下每个类中数目占比就多的红酒品牌：

```
classA = winedf[winedf.label==0].groupby(["keyword"]).size().sort_values(ascending=False).reset_index()
classA.columns=["keyword","A"]
classB = winedf[winedf.label==1].groupby(["keyword"]).size().sort_values(ascending=False).reset_index()
classB.columns=["keyword","B"]
classC = winedf[winedf.label==2].groupby(["keyword"]).size().sort_values(ascending=False).reset_index()
classC.columns=["keyword","C"]
classD = winedf[winedf.label==3].groupby(["keyword"]).size().sort_values(ascending=False).reset_index()
classD.columns=["keyword","D"]
classE = winedf[winedf.label==4].groupby(["keyword"]).size().sort_values(ascending=False).reset_index()
classE.columns=["keyword","E"]
classF = winedf[winedf.label==5].groupby(["keyword"]).size().sort_values(ascending=False).reset_index()
classF.columns=["keyword","F"]
```

```
classA[:10]
```

	keyword	A
0	长城/greatwall	606
1	张裕/changyu	219
2	通化/tonghua	180
3	王朝/dynasty	102
4	拉菲/lafite	98
5	黄尾袋鼠/yellow tail	97
6	杰卡斯/jacob's creek	97
7	罗曼尼康帝庄园	96
8	高斯达	89
9	奔富/penfolds	88

```
classB[:10]
```

	keyword	B
0	长城/greatwall	345
1	拉菲/lafite	333
2	奔富/penfolds	272
3	黄尾袋鼠/yellow tail	158
4	纷赋/wolfblass	158
5	干露/concha y toro	158
6	蒙特斯/montes	150
7	通化/tonghua	143
8	贝灵哲/beringer	102
9	杰卡斯/jacob's creek	101

```
classC[:10]
```

	keyword	C
0	罗曼尼康帝庄园	468
1	通化/tonghua	368
2	波尔多	334
3	长城/greatwall	179
4	黄尾袋鼠/yellow tail	155
5	蒙特斯/montes	150
6	路易拉菲/louis lafon	144
7	张裕/changyu	143
8	贺兰山	123
9	名庄靓年	87

从结果上看，同样的红酒品牌会在多个类中出现，可以由数量上判断该品牌属于哪个类别，由此整理得到：

A	'长城/greatwall ', '张裕/changyu ', '王朝/dynasty ', '高斯达', '莫高/mogao ', '智象/chilephant', '拉梦堡/lamengbao', '芙华/la fiole', '玛茜/rochemazet', '尼雅/niya ', '天鹅庄', '华东/huadong', '嘉伦多', '罗莎庄园', '蒙大菲/robert mondavi'　等等
B	'拉菲/lafite', '奔富/penfolds', '纷赋/wolfblass', '黄尾袋鼠/yellow tail', '干露/concha y toro ', '蒙特斯/montes ', '贝灵哲/beringer', '杰卡斯/jacob’s creek', '圣丽塔', '威赛帝斯', '卡思黛乐/castel', '璞立/beaulieu vineyard', '也买酒/yesmywine', '麦格根/mcguigan', '玛歌酒庄/chateau margaux', '加州乐事', '音符/awjs'　等等
C	'罗曼尼康帝庄园', '通化/tonghua', '波尔多', '路易拉菲/louis lafon ', '贺兰山', '名庄靓年', '拉蒙', '香奈/j.p.chenet ', '木桐', '丰收'　等等
D	'罗马假日', '君顶', '优尼特/riunite', '阿维娃/aviva', '蓝海之鲸/mr.sparkling'
F	'木桐古堡/ch. mouton rothschild', '拉图酒庄', '作品一号/opus one'

代码如下：

```
level = pd.DataFrame(list(set(winedf['keyword'])),columns=["keyword"])
level = pd.merge(level,classA,how='left',on=["keyword"])
level = pd.merge(level,classB,how='left',on=["keyword"])
level = pd.merge(level,classC,how='left',on=["keyword"])
level = pd.merge(level,classD,how='left',on=["keyword"])
level = pd.merge(level,classF,how='left',on=["keyword"])
level = level.fillna(0)

level['max_value']=level.max(axis=1)
level=level[level.max_value>0]
print(level.shape)
def appendlevel(sr):
    levels=list('ABCDF')
    for i in levels:
        if sr[i] == sr['max_value']:
            return i
level['level'] = level.apply(lambda x:appendlevel(x),axis=1)#每一行apply
```

```
level[level.level=='C'].sort_values("max_value",ascending=False)
```

	keyword	A	B	C	D	F	max_value	level
140	罗曼尼康帝庄园	96.0	75.0	468.0	14.0	44.0	468.0	C
20	通化/tonghua	180.0	143.0	368.0	29.0	0.0	368.0	C
142	波尔多	46.0	73.0	334.0	13.0	3.0	334.0	C
128	路易拉菲/louis lafon	63.0	32.0	144.0	0.0	0.0	144.0	C
134	贺兰山	1.0	0.0	123.0	7.0	0.0	123.0	C
78	名庄靓年	0.0	3.0	87.0	1.0	0.0	87.0	C
167	拉蒙	0.0	0.0	71.0	0.0	0.0	71.0	C
0	香奈/j.p.chenet	31.0	57.0	65.0	1.0	0.0	65.0	C
153	木桐	5.0	1.0	54.0	0.0	0.0	54.0	C
161	丰收	4.0	10.0	50.0	5.0	0.0	50.0	C

可以用同样的方法查看其他级别的红酒品牌类目。

5. 给微博数据加标签

把清洗微博数据的结果和同级红酒的处理结果做合并

```
result = pd.merge(df,level[["keyword","level"]],how="left",on="keyword")
```

结果如下：

	keyword	gender	age	country	region	level
0	卓林/zonin	m	9999	其他	其他	A
1	怡园酒庄/grace vineyard	f	9999	香港	其他	B
2	莱茵黑森	f	9999	广东	广州	B
3	长城/greatwall	m	34	上海	浦东新区	A
4	奥兰	f	9999	北京	其他	B
5	华东/huadong	f	9999	山东	青岛	A
6	黄尾袋鼠/yellow tail	m	9999	辽宁	大连	C
7	黄尾袋鼠/yellow tail	m	9999	上海	其他	C
8	奥兰	f	9999	广东	深圳	B
9	安徒生	m	9999	其他	其他	A
10	安徒生	m	9999	湖南	长沙	A
11	木桐嘉棣/mouton cadet	f	9999	北京	朝阳区	C
12	加州乐事	m	9999	其他	其他	C

保存为 level.csv 文件，下一节采用 pandaBI 的大屏展示显示相关数据