zhuanlan.zhihu.com

Python数据分析——类别数据的转换

7-9 minutes

背景：最近在看《Python机器学习》这本书，想整理成笔记，供自己和小伙伴们学习。

这次的内容是数据预处理中的**类别数据的转换。**

什么是*类别数据*

什么是***类别数据***呢？***类别数据***是有分类特征的数据，相对应的是***数值数据***。比如说，在一个电影数据集中，电影类型特征列中就有一些***类别数据***（科幻、爱情、恐怖、乡村等等）。

以下用电影数据集为例说明：

	类型	地区	评星	适宜儿童	时长（min）
0	爱情	内地	2	是	126
1	恐怖	欧美	4	否	131
2	动作	日本	3	否	135
3	搞笑	港台	5	是	110

利用Pandas写的DataFrame数据框

标称特征和有序特征

类别数据特征又可分为**标称特征**和**有序特征。**

标称特征只代表类别，数据无序，如电影数据集中的类型、地区特征，爱情和动作是无法做比较的。

有序特征的数据是用于分类且有序的，如电影数据集中的评星，显然5高于4，3高于2，可以比较。

构造电影数据集

我这里用Python的pandas库构造了DataFrame数据框，pandas是非常有用的数据处理工具，各种逆天接口让你爽翻。下面把代码写下：

```
import pandas as pd
Movies = pd.DataFrame([
    ['爱情','内地',2,'是'],
```

```
    ['恐怖','欧美',4,'否'],
    ['动作','日本',3,'否'],
    ['搞笑','港台',5,'是']
                    ],
    columns=['类型','地区','评星','适宜儿童'])
Movies
```

运行后得到结果：

	类型	地区	评星	适宜儿童	时长（min）
0	爱情	内地	2	是	126
1	恐怖	欧美	4	否	131
2	动作	日本	3	否	135
3	搞笑	港台	5	是	110

可以看到，该数据集包含3个标称特征（类型、地区、适宜儿童），1个有序特征（评星），1个数值特征（时长）。

类标的编码（重点）

接下来进行到本篇笔记的重点，也就是类表的编码。

可以看到，类型、地区特征里数据都是字符串，虽然方便观看，但是机器学习库（算法运用）要求类标以整数形式进行编码。

这里用到3种方式进行类标编码：

1、字典映射

以‘适宜儿童’这一特征列为例，将‘是’映射为1，将‘否’映射为0。

```
dic = {'是':1,'否':0}
Movies['适宜儿童'] = Movies['适宜儿童'].map(dic)
Movies
```

执行命令后得到：

	类型	地区	评星	适宜儿童	时长（min）
0	爱情	内地	2	1	126
1	恐怖	欧美	4	0	131
2	动作	日本	3	0	135
3	搞笑	港台	5	1	110

‘适宜儿童’特征列数据变成1和0，对应是和否

假如我们有很多特征值，比如‘类型’特征，电影类型有几十种，总不能挨个写成字典映射，这样太累了。经济的做法是采用枚举方式对每个特征进行编码，因为标称特征无序，所以哪一类被编成哪一个整数不重要。

开始代码：

```
import numpy as np
dic = {label:idx for idx,label in enumerate(np.unique(Movies['类型']))}
dic
```

输出：dic = {'动作': 0, '恐怖': 1, '搞笑': 2, '爱情': 3}

```
Movies['类型'] = Movies['类型'].map(dic)
Movies
```

执行命令后得到：

	类型	地区	评星	适宜儿童	时长（min）
0	3	内地	2	1	126
1	1	欧美	4	0	131
2	0	日本	3	0	135
3	2	港台	5	1	110

2、使用scikit-learn库进行整数编码

对‘地区’特征列进行编码

先导入scikit-learn库中的LabelEncode类，该类可完美执行整数编码工作。

上代码：

```
from sklearn.preprocessing import LabelEncoder
le = LabelEncoder()
y = le.fit_transform(Movies['地区'])
y
```

输出：y = array([0, 2, 1, 3], dtype=int64)

y是一个numpy数组，四个数字分别对应内地、欧美、日本、港台

执行命令后得到：

	类型	地区	评星	适宜儿童	时长（min）
0	3	0	2	1	126
1	1	2	4	0	131
2	0	1	3	0	135
3	2	3	5	1	110

3、机器学习最中意的：独热编码

前面我们将地区分成四个数字，虽然地区没有顺序大小之分，但如果把数据扔到分类器里，分类器会默认3>2>1>0，这样四个地区便成了有序特征。

这不是我们要的目的，最优的操作是，能判别出非此即彼，某电影要么是欧美片要么不是欧美片，要么是内陆片要么不是内陆片。。。。对每种地区进行判断，只有两种结果，是和不是。

解决该问题的方法是独热编码技术。即创建一个虚拟特征，虚拟特征的每一列各代表标称数据的一个值。

把‘地区’这1列裂变成4列：

	内陆	日本	欧美	港台
0	1	0	0	0
1	0	0	1	0
2	0	1	0	0
3	0	0	0	1

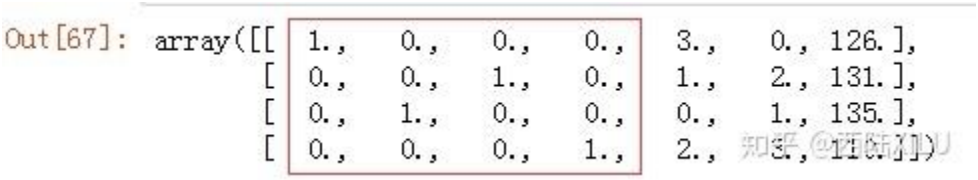
1代表该电影属于该地区，0代表不属于该地区。

这就是独热编码，这样表示有利于分类器的更好运算。

给出代码：

```
from sklearn.preprocessing import OneHotEncoder
ohe = OneHotEncoder(categorical_features=[2])
ohe.fit_transform(Movies.values).toarray()
```

输出：



前四列变成四个地区特征，0代表否，1代表是

还可以用pandas（神器）中的get_dummies方法实现独热编码技术，该方法只对字符串列进行转换，数值列保持不变。