小鱼吻水

生命不息 折腾不止

scikit-learn 中 OneHotEncoder 解析

概要

在 sklearn 包中，OneHotEncoder 函数非常实用，它可以实现将分类特征的每个元素转化为一个可以用来计值。本篇详细讲解该函数的用法，也可以参考官网 sklearn.preprocessing.OneHotEncoder。

解析

该函数在 `sklearn.preprocessing` 类中，格式为：

```
OneHotEncoder(n_values='auto', categorical_features='all', dtype=<class 'numpy.float64'>,
sparse=True, handle_unknown='error')
```

为了方便理解，我们先看下面一个例子：

```
# -*- coding: utf-8 -*-

from sklearn.preprocessing import OneHotEncoder

enc = OneHotEncoder()
enc.fit([[0, 0, 3],
         [1, 1, 0],
         [0, 2, 1],
         [1, 0, 2]])

ans = enc.transform([[0, 1, 3]]).toarray()  # 如果不加 toarray() 的话，输出的是稀疏的存储格式，即索引加值的形
可以通过参数指定 sparse = False 来达到同样的效果
print(ans) # 输出 [[ 1.  0.  0.  1.  0.  0.  0.  0.  1.]]
```

下面解释输出结果的意思。对于输入数组，这依旧是把每一行当作一个样本，每一列当作一个特征，

- 我们先来看第一个特征，即第一列 $[0,1,0,1]$，也就是说它有两个取值 0 或者 1，那么 one-hot 就会使用表示这个特征，$[1,0]$ 表示 0，$[0,1]$ 表示 1，在上例输出结果中的前两位 $[1,0...]$ 也就是表示该特征为 0
- 第二个特征，第二列 $[0,1,2,0]$，它有三种值，那么 one-hot 就会使用三位来表示这个特征，$[1,0,0]$ 表示 $[0,1,0]$ 表示 1，$[0,0,1]$ 表示 2，在上例输出结果中的第三位到第六位 $[...0,1,0,0...]$ 也就是表示该特征为
- 第二个特征，第三列 $[3,0,1,2]$，它有四种值，那么 one-hot 就会使用四位来表示这个特征，$[1,0,0,0]$ 表 $[0,1,0,0]$ 表示 1，$[0,0,1,0]$ 表示 2，$[0,0,0,1]$ 表示 3，在上例输出结果中的最后四位 $[...0,0,0,1]$ 也就 该特征为 3

好了，到此相信我们已经很明白它的意思了。值得注意的是，虽然训练样本中的数值仅仅代表类别，但是也必用数值格式的数据，如果使用字符串格式的数据会报错。

下面解释一下函数中参数的意思，

- `n_values='auto'`，表示每个特征使用几维的数值由数据集自动推断，即几种类别就使用几位来表示。可以自己指定，看下面这个例子：

```
# -*- coding: utf-8 -*-

from sklearn.preprocessing import OneHotEncoder

enc = OneHotEncoder(n_values = [2, 3, 4])
enc.fit([[0, 0, 3],
         [1, 1, 0]])

ans = enc.transform([[0, 2, 3]]).toarray()
print(ans) # 输出 [[ 1.  0.  0.  0.  1.  0.  0.  0.  1.]]
```

注意到训练样本中第二个特征列没有类别 2，但是结果中依然将类别 2 给编码了出来，这就是自己指定维数的了（我们使用 3 位来表示第二个特征，自然包括了类别 2），第三列特征同样如此。这也告诫我们，如果训练

中有丢失的分类特征值，我们就必须显示地设置参数 `n_values` 了，这样防止编码出错。

- `categorical_features = 'all'`，这个参数指定了对哪些特征进行编码，默认对所有类别都进行编码。自己指定选择哪些特征，通过索引或者 bool 值来指定，看下例：

```
# -*- coding: utf-8 -*-

from sklearn.preprocessing import  OneHotEncoder

enc = OneHotEncoder(categorical_features = [0,2]) # 等价于 [True, False, True]
enc.fit([[0, 0, 3],
         [1, 1, 0],
         [0, 2, 1],
         [1, 0, 2]])

ans = enc.transform([[0, 2, 3]]).toarray()
print(ans) # 输出 [[ 1.  0.  0.  0.  0.  1.  2.]]
```

输出结果中前两位 $[1, 0]$ 表示 0，中间四位 $[0, 0, 0, 1]$ 表示对第三个特征 3 编码，第二个特征 2 没有进行编码
放在最后一位。

- `dtype=<class 'numpy.float64'>` 表示编码数值格式，默认是浮点型。
- `sparse=True` 表示编码的格式，默认为 True，即为稀疏的格式，指定 False 则就不用 toarray() 了
- `handle_unknown='error'`，其值可以指定为 "error" 或者 "ignore"，即如果碰到未知的类别，是返回一
还是忽略它。

方法 `transform(X)` 就是对 X 进行编码了。在实际应用中，我们更常用方法 `fit_transform()`，也就是-
位，看下例：

```
# -*- coding: utf-8 -*-

from sklearn.preprocessing import  OneHotEncoder

enc = OneHotEncoder(sparse = False)
ans = enc.fit_transform([[0, 0, 3],
                         [1, 1, 0],
                         [0, 2, 1],
                         [1, 0, 2]])

print(ans) # 输出 [[ 1.  0.  1. ...,  0.  0.  1.]
           #       [ 0.  1.  0. ...,  0.  0.  0.]
           #       [ 1.  0.  0. ...,  1.  0.  0.]
           #       [ 0.  1.  1. ...,  0.  1.  0.]]
```

分类: 机器学习

标签: OneHotEncoder, sklearn

小鱼吻水
关注 - 0
粉丝 - 4
+加关注

« 上一篇：TensorFlow 内置重要函数解析

posted @ 2018-06-09 16:28 小鱼吻水 阅读(1136) 评论(1) 编辑

评论列表

#1楼 2018-09-12 22:32 gongel

谢谢你