一、概述

本文主要讲述机器学习中的一些标记编码方式，分为onehot-encoding、Label-encoding，这两类编码方式是在进后对数据进行编码的主要方式。主要参考《Python机器学习经典实例》，以及各类博客。

二、内容

1. 独热编码One-Hot-Encoding

概述：

又称一位有效编码，其方法是使用N位状态寄存器来对特征的N个状态进行编码，每个状态都由他独立的寄存器位，并且在任意时候，其中只有一位有效。即有多少个状态就有多少bit，而且只有一个bit为1，其他全为0的一种码制。

作用：

对离散型的分类型数据进行数字化，比如将文本分类属性的性别进行数字化的独热编码。

1）解决了分类器不好处理属性数据的问题，

2）在一定程度上起到了扩充特征的作用

为什么要用独热编码

独热编码（哑变量 dummy variable）是因为大部分算法是基于向量空间中的度量来进行计算的，为了使非偏序关系的变量取值不具有偏序性，并且到圆点是等距的。

使用one-hot编码，将离散特征的取值扩展到了欧式空间，离散特征的某个取值就对应欧式空间的某个点，特征之间的距离计算更加合理。

离散特征进行one-hot编码后，编码后的特征，其实每一维度的特征都可以看做是连续的特征。就可以跟对连续型特征的归一化方法一样，对每一维特征进行归一化。比如归一化到[-1,1]或归一化到均值为0,方差为1。

独热编码优缺点

- 优点：

能够处理非连续型数值特征。在一定程度上也扩充了特征。比如性别本身是一个特征，经过one hot编码以后，就变成了男或女两个特征。

它的值只有0和1，不同的类型存储在垂直的空间。

- 缺点：

当类别的数量很多时，特征空间会变得非常大，稀疏矩阵会很稀，占内存空间大。在这种情况下，一般可以用PCA来减少维度。而且 one hot encoding+PCA 这种组合在实际中也非常有用

什么情况下 (不) 使用独热编码

使用：独热编码用来解决类别型数据的离散值问题

不用：将离散型特征进行one-hot编码的作用，是为了让距离计算更合理，但如果特征是离散的，并且不用one-hot编码就可以很合理的计算出距离，那么就没必要进行one-hot编码（计算距离的合理性方面）

有些基于树的算法在处理变量时，并不是基于向量空间度量，数值只是个类别符号，即没有偏序关系，所以不用进行独热编码。

Tree Model不太需要one-hot编码： 对于决策树来说，one-hot的本质是增加树的深度。

解释：

对于离散数据 {sex：{male， female，other}}， 如果单纯使用{1，2，0}进行编码（即标签编码），在模型训练中不同的值可能会使同一特征在样本中的权重发生变化。

采用独热编码，有3个分类值，需要3个bit位表示该特征值，对应bit位为1其他为0对应原特征值，得到的独热编码为 {100, 010, 001}分别表示{male, female, other}

实例：

如在代码中给出的输入类别： [[0,1,2,3],[3,4,1,2],[5,4,2,1],[0,4,1,16]]

在第一组向量特征中有[0,3,5]三个特征，在第二组向量特征中有[1，4]两个特征，在第三组向量特征中有[2，1]两个特征，在第四组向量特征中有[3，2，1，16]四个特征，

在第一组向量特征中有三个不同特征，所以我们用三位二进制[0，0，0]表示特征

特征值：	表达式：
0	[1,0,0]
3	[0,1,0]
5	[0,0,1]

在第二组向量特征中有两个不同特征，所以我们用两位二进制[0，0]表示特征

特征值：	表达式：
1	[1,0]
4	[0,1]

以此类推

我们将将对[[0,4,2,16]]进行编码，首位0对应[1,0,0]，4对应[0,1].....

[[0,4,2,16]]onehot编码结果为：[[1. 0. 0. 0. 1. 0. 1. 0. 0. 0. 1.]]

2. 标签编码

概述：

将离散型变量转换成连续的数值型变量，即对不连续的数字或者文本进行编号。

对于不同的特征，其编码表不同且相互独立；编码和解码都要使用对应特征的编码表。

实例：

比较好理解具体看代码。

三、心得体会

本章的编码标记方式同样为机器学习的基础，不同的数据类型需要用不同的编码形式，这个在之后写项目的时候要慢慢体会。

参考博客：https://blog.csdn.net/tcsjk/article/details/82774376

https://www.cnblogs.com/zhoukui/p/9159909.html