

7. 编码方式

由于机器学习算法都是在矩阵上执行线性代数计算的，所以参加计算的特征必须是数值型的。

对于非数值型的特征需要进行编码处理。

1. 标签编码：

特点：

- 解决了分类编码的问题，可以自由定义量化数值
- 数值本身没有任何含义，仅是标识，可解释性差。

适用范围： \Rightarrow 离散数据之间本身有排序，^{大小}高低

- 对于有序类型的数据，使用标签编码更好。因为具有排序逻辑

- 对数值大小不敏感的模型（如树模型），建议使用标签编码。

2. 独热编码：

采用 N 位状态寄存器对 N 个可能的取值进行编码，每个状态都由独热的寄存器来表示，并且任意时刻 R 有一位有效。

特点：

- 解决了分类器不好处理分类变量的问题，同时也可以扩展特征

- 编码后的属性是稀疏的，存在大量零元素。

- 当类别非常多的时候，特征空间会很大，容易导致维度灾难。

~~适用范围~~ 适用范围：对数值大小敏感的模型，必须使用独热编码。

notebook

3) 目标编码 (target encoding)

使用 target 值的均值作为 category 变量的替换值。

State	Score		State	Score
Ca	0.4	Avg → Ca: 0.45 Ne: 0.15 Tex: 0.85	Ca → 0.45	0.4
Ne	0.1		Ne → 0.15	0.1
Tex	0.9		Tex → 0.85	0.9
Ne	0.2		Ne → 0.15	0.2
Ca	0.5		Ca → 0.45	0.5
Tex	0.8		Tex → 0.85	0.8

缺点: 1) 依赖于 y 值

2) 对于 category 值, 全都来自训练集, 因此更容易出现 overfitting

4) 留一编码 (Leave-One-Out encoding)

改进了目标编码:

使用除了 ~~该行~~ 对应行 target 值外的其他同类 target 来计算。

State	Score		State	Score
Ca	0.4	→ Avg. Exc: 0.5 →	0.5	0.4
Ne	0.1	→ Avg. Exc: 0.2 →	0.2	0.1
Tex	0.9	→ Avg. Exc: 0.8 →	0.8	0.9
Ne	0.2	→ Avg. Exc: 0.1 →	0.1	0.2
Ca	0.5	→ Avg. Exc: 0.4 →	0.4	0.5
Tex	0.8	→ Avg. Exc: 0.9 →	0.9	0.8