Train In Data

Smoothing

Feature-engine

Encoding logic

The two probabilities are then "blended" using a weighting factor that is a function of the sample size:

$$\text{Value} = \lambda \times \text{posterior} + (1 - \lambda) \times \text{prior}$$

Encoding logic

$$\lambda = \frac{n \times t^2}{\sigma^2 + n \times t^2}$$

σ^2: variance within category

t^2: variance of entire sample

n: observations per category

Encoding logic

$$\lambda = \frac{n \times t^2}{\sigma^2 + n \times t^2}$$

σ^2: variance within category

t^2: variance of entire sample

n: observations per category

This implementation considers both sample size and variance

Encoding logic: less observations

$$\lambda = \frac{n \times t^2}{\sigma^2 + n \times t^2}$$

n	s2	t2
10	5	10
5	5	10
1	5	10

n x t2	lambda
100.00	0.95
50.00	0.91
10.00	0.67

Encoding logic: less variability

$$\lambda = \frac{n \times t^2}{\sigma^2 + n \times t^2}$$

n	s2	t2
10	10	10
10	5	10
10	1	10

n x t2	lambda
100.00	0.91
100.00	0.95
100.00	0.99

Encoding logic

The two probabilities are then "blended" using a weighting factor that is a function of the sample size:

$$\text{Value} = \lambda \times \text{posterior} + (1-\lambda) \times \text{prior}$$

- The more observations, the more we trust the posterior
- The less variability in the category, the more we trust the posterior

Smoothing in Feature-engine

If 0 ➔ just target mean encoder as explained previously

If **auto** ➔ λ as discussed here. It is calculated automatically, because it is a product of the variances and the sample size

If int ➔ then λ = ni / (ni+smoothing)

Where ni is the number of observations per category and smoothing is the arbitrary value entered by the user.

THANK YOU

www.trainindata.com