2024.03.21 AD

Calculate information gain

Consider data:

```
Data 0.243 0.245 0.437 0.481 0.608 0.666
Category C0 C0 C1 C1 C1 C0
```

Calculate j to make:

$$\begin{aligned} j &= \max\left(\text{GAIN}_{\text{split}}\right) \\ &= \max\left(\text{Entropy}(p) - \left(\sum_{i=1}^{k} \frac{n_i}{n} \text{Entropy}(p_i)\right)\right) \end{aligned}$$

Taking the spilit as `C0 | C0 C1 C1 C1 C0`, we get:

$$\sum_{i=1}^{k} \frac{n_i}{n} \text{Entropy}(p_i) = 0 + 0.4 * 1.322 + 0.6 * 0.737 = 0.9710$$

`C0 C0 | C1 C1 C1 C0`:

$$\sum_{i=1}^{k} \frac{n_i}{n} \text{Entropy}(p_i) = 0 + 0.75 * (1.585 - 2) + 0.5 = 0.8115$$

`C0 C0 C1 | C1 C1 C0`:

$$\sum_{i=1}^{k} \frac{n_i}{n} \text{Entropy}(p_i) = (0.66 * 0.585 + 0.33 * 1.585) * 2 = 1.826$$

`C0 C0 C1 C1 | C1 C0`:

$$\sum_{i=1}^{k} \frac{n_i}{n} \text{Entropy}(p_i) = (0.5 * 2 + 0.5 * 2) * 2 = 4$$

`C0 C0 C1 C1 C1 | C0`:

Same as `C0 | C0 C1 C1 C1 C0`.

Picking the split as `C0 C0 | C1 C1 C1 C0`, we get the maximum information gain.

Discretization

(1):

$$\begin{aligned} &P(C1) = \frac{1}{2} \quad P(C2) = \frac{1}{2} \\ &\text{Entropy} \\ &= -P(C1)\log_2 P(C1) - P(C2)\log_2 P(C2) \\ &= -\frac{1}{2}\log_2\frac{1}{2} - \frac{1}{2}\log_2\frac{1}{2} = 1 \end{aligned}$$

(2):

$$\begin{aligned} &P(C1) = \frac{1}{4} \quad P(C2) = \frac{1}{4} \quad P(C3) = \frac{1}{4} \quad P(C4) = \frac{1}{4} \\ &\text{Entropy} = 2 \end{aligned}$$

Exerciese 3

```python
from math import log2

def calc_dcg(rel_list: list[float], k: int) -> float:
    dcg = 0
    for i in range(k):
        dcg += (2 ** rel_list[i] - 1) / (log2(i + 2))
    return dcg


def calc_idcg(rel_list: list[float]) -> float:
    rel_list = sorted(rel_list, reverse=True)
    return calc_dcg(rel_list, len(rel_list))


def calc_ndcg(rel_list: list[float], k: int) -> float:
    dcg = calc_dcg(rel_list, k)
    idcg = calc_idcg(rel_list)
    return dcg / idcg


if __name__ == "__main__":
    A = [3, 3, 0, 2, 2, 1]
    B = [3, 3, 2, 0, 2, 1]

    print("A: {}, NDCG(A): {}".format(A, calc_ndcg(A, len(A))))
    print("B: {}, NDCG(B): {}".format(B, calc_ndcg(B, len(B))))
```

Running the code, we get:

```
$ python main.py
A: [3, 3, 0, 2, 2, 1], NDCG(A): 0.9746435566818092
B: [3, 3, 2, 0, 2, 1], NDCG(B): 0.9888925979887605
```