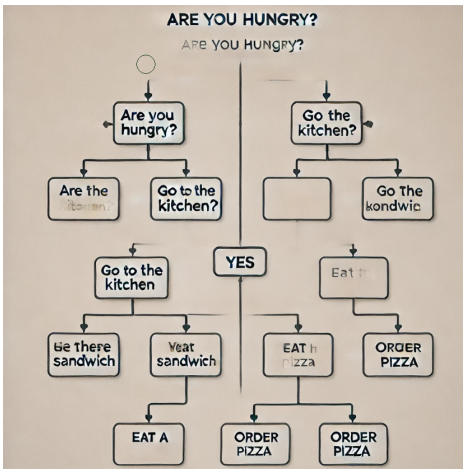
Дерево решений

Дерево решений - классика, которая хоть и простая, но часто применяется в реальной жизни. Это алгоритм машинного обучения, который разбивает объекты на группы по их признакам, но самое интересное, признаки он определяет сам. Мы можем лишь настроить количество групп, или глубину анализа данных.



Нам нужно создать дерево, которое само ставит правильные if в каждую вершину для самого оптимального разбиения объектов на группы. Первая идея как это сделать: для 1-й вершины перебирать всевозможные варианты, для каждого варианта для 1-й вершины перебрать всевозможные варианты для 2-й вершины.

Кто знаком с программированием, то это получается for внутри которого for внутри которого for ... и тд.

Чтобы настолько не нагружать компьютер, есть способ как это оптимизировать с помощью умных слов "вероятности, количество информации и энтропия".

Формула Хартли.

Задача 0: Сколько минимально ячеек нужно, чтобы зашифровать через двоичный код (используя 0 и 1) алфовит из 5 букв ABCDE?

Попытаемся сделать это с помощью 2 ячеек:

0	0	A
0	1	B
1	0	C
1	1	D

Не хватило для буквы E, значит нужно 3 ячейки return

0	0	0	A
0	0	1	B
0	1	0	C
0	1	1	D
1	0	0	E

Задача 1: Сколько минимально ячеек нужно, чтобы зашифровать через двоичный код (используя 0 и 1) алфовит из n букв

Решение: Если у нас есть x ячеек, то с помощью них мы можем зашифровать 2^x символов. Получается равенство $2^x = n$. Отсюда можно найти x.

Ответ: $\log_2 n$.

Формула Хартли используется для вычисления количества информации, необходимого для выбора одного события из множества равновероятных событий. Она записывается следующим образом:

$$I = \log_b N$$

где:

- N — количество возможных равновероятных событий,
- b — основание логарифма (обычно $b = 2$ для измерения в битах, или $b = 10$ для десятичных логарифмов),
- I — количество информации, измеряемое в битах (если $b = 2$).

Интуитивное объяснение

Представьте, что вам нужно выбрать одно событие из множества N равновероятных вариантов (нужно выбрать один символ из N символов). Чем больше N, тем больше неопределённость. Формула Хартли показывает, сколько информации потребуется, чтобы устранить эту неопределённость (сколько ячеек нужно чтобы через 0 и 1 определить этот символ).

Пример

Рассмотрим ситуацию, где $N = 8$ (например, выбор одного из 8 объектов). Тогда количество информации:

$$I = \log_2 8 = 3 \text{ бита.}$$

Это означает, что для определения одного из 8 событий потребуется 3 "да/нет"вопроса (0 или 1, по одному биту на вопрос).

Если $N = 16$, то:

$$I = \log_2 16 = 4 \text{ бита.}$$

В этом случае потребуется 4 бита информации.

Применения

Формула Хартли применяется в:

- Теории информации: для оценки количества информации в равновероятной системе.
- Кодировании данных: для вычисления минимального количества битов, необходимого для кодирования N событий.
- Комбинаторике: для анализа количества способов выбора или упорядочивания объектов.

Связь с энтропией

Если события равновероятны, энтропия Шеннона совпадает с формулой Хартли. Для событий с разными вероятностями энтропия Шеннона используется как более общий случай.

Формула Шеннона.

Формула Шеннона используется для измерения количества информации (энтропии), связанной с вероятностным распределением событий.

В физике энтропия описывает степень беспорядка системы: Чем выше энтропия, тем больше хаоса и меньше упорядоченности.

Задача 2 Мы подъехали к светофору в случайный момент времени, оказалось красный горит 50 секунд, зеленый 30 секунд и желтый 20 секунд. Определим количество информации красного цвета. ~~Если говорить детским языком, то насколько каждая секунда красного цвета влияет на дорожное движение.~~

Сначала применим формулу Хартли, чтобы определить количество информации одной конкретной секунды красного цвета. Она появляется с вероятностью p(A) = 0.5. Значит информация, которую нужно зашифровать $\frac{1}{p(A)}$

Количество ячеек, которые нужны чтобы столько информации зашифровать $\log_2 \frac{1}{p(A)}$

Из всех красных секунд, вероятность, что именно эту красную секунду нужно зашифровать p(A). Получается энтропия одной красной секунды

$$\text{p(A)} \ \log_2 \frac{1}{p(A)}$$

Энтропия всего светофора $H(X)$ это сумма таких выражений:

$$H(X) = -(p(A)\log_2 p(A) + p(B)\log_2 p(B) + p(C)\log_2 p(C)).$$

Подставляем значения:

$$H(X) = -(0.5 \cdot \log_2 0.5 + 0.3 \cdot \log_2 0.3 + 0.2 \cdot \log_2 0.2).$$

Вычисляем:

$$H(X) \approx 1.485 \text{ бита}.$$

Общий вид формулы Шеннона

$$H(X) = -\sum_{i=1}^{n} p_i \log_b p_i$$

где:

- $H(X)$ — энтропия случайной величины X,
- p_i — вероятность i-го события ($p_i > 0$, и $\sum p_i = 1$),
- n — количество возможных исходов,
- b — основание логарифма (обычно $b = 2$ для измерения в битах, или $b = e$ для измерения в натах).

Интуитивное объяснение энтропии

Энтропия показывает, насколько "неопределённа"информация. Рассмотрим два случая:

- Если все события равновероятны ($p_i = 1/n$), энтропия максимальна, так как неопределённость высока.
- Если одно из событий имеет вероятность 1 ($p_1 = 1$, а остальные $p_i = 0$), то энтропия равна 0, так как неопределённости нет.

Дерево решений.

В самой первой странице была идея перебором создать дерево решений. Но там получалось for внутри которого for внутри которого for ... и тд. Чтобы настолько не нагружать компьютер, перебор будем делать гораздо быстрее:

1. Подберем условие if для первой вершины нашего дерева: для точек на плоскости перебираем всевозможные горизантальные и вертикальные прямые, и для каждой проведенной прямой считаем общую энтропию разбиения.

   Чтобы посчитать общую энтропию распределения на группы воспользуемся следующим методом: процент шариков из левой группы (относительно шариков обоих групп) умножим на энтропию левой группы. Аналогично с правой группой. Суммируем два этих числа и получаем энтропию распределения.

   Та прямая, что получает самую маленькую энтропию и есть нужная для нас прямая. Т.к. она содержит самое маленькое количество информации, или если говорить проще: эта прямая создает меньше всего хаоса, больше всего порядка.

2. Теперь подбираем if для второй вершины нашего дерева. Но самое классное то, что подбор второго if не находится внутри подбора первого if. Это for которые стоят параллельно один после другого, а не один внутри другого.

3. Продолжаем подбор для всех остальных вершин до тех пор, пока

   - не встретим ограничение по количеству вершин (обычно задается в условии или определяется таким образом, чтобы не было overfitting и underfitting)

   - распределения тестовых данных по группам будет с энтропией 0.

   - или если новое разбиение уменьшает энтропию менее чем на 0.01, то можно остановить