Лекція 12
Дерева ухвалення рішень

§55 Дерева ухвалення рішень

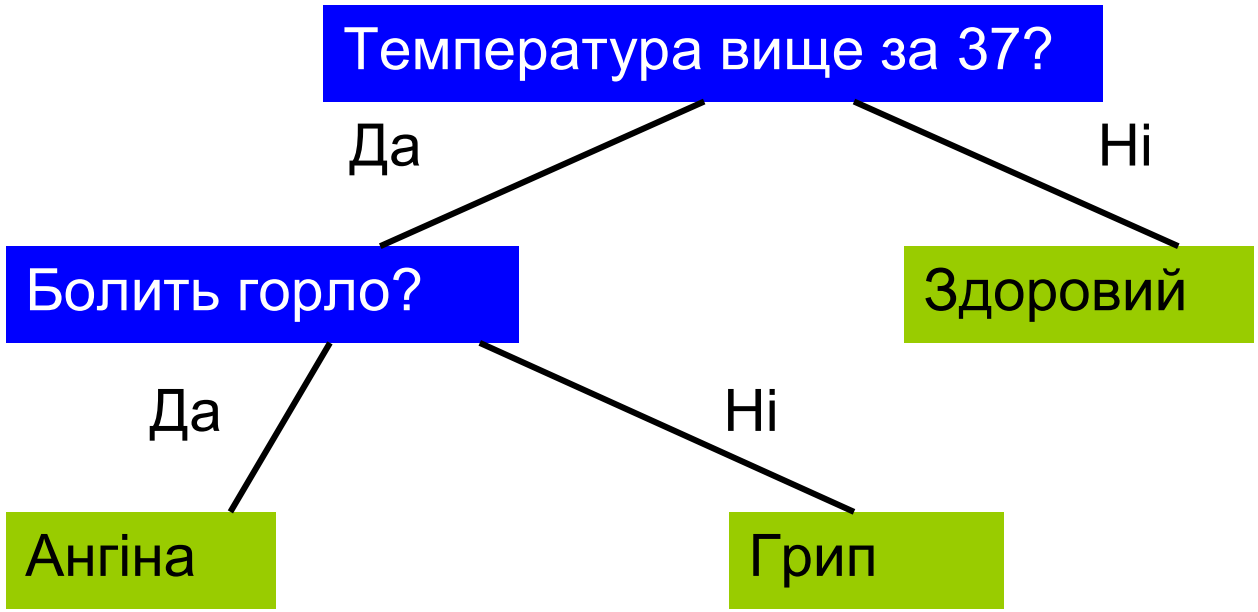
Лінійні моделі (огляд)

Особливостей лінійних моделей:

- Лінійні моделі **швидко вчаться**. У випадку із середньоквадратичною помилкою для вектора ваг навіть є аналітичне рішення. Також легко застосовувати для лінійних моделей градієнтний спуск.
- При цьому лінійні моделі можуть **відновлювати тільки прості залежності** через обмежену кількість параметрів (ступенів вільності).
- У той же час лінійні моделі можна використовувати для відновлення нелінійних залежностей за рахунок переходу до **спрямляючого простору**, що є досить складною операцією.

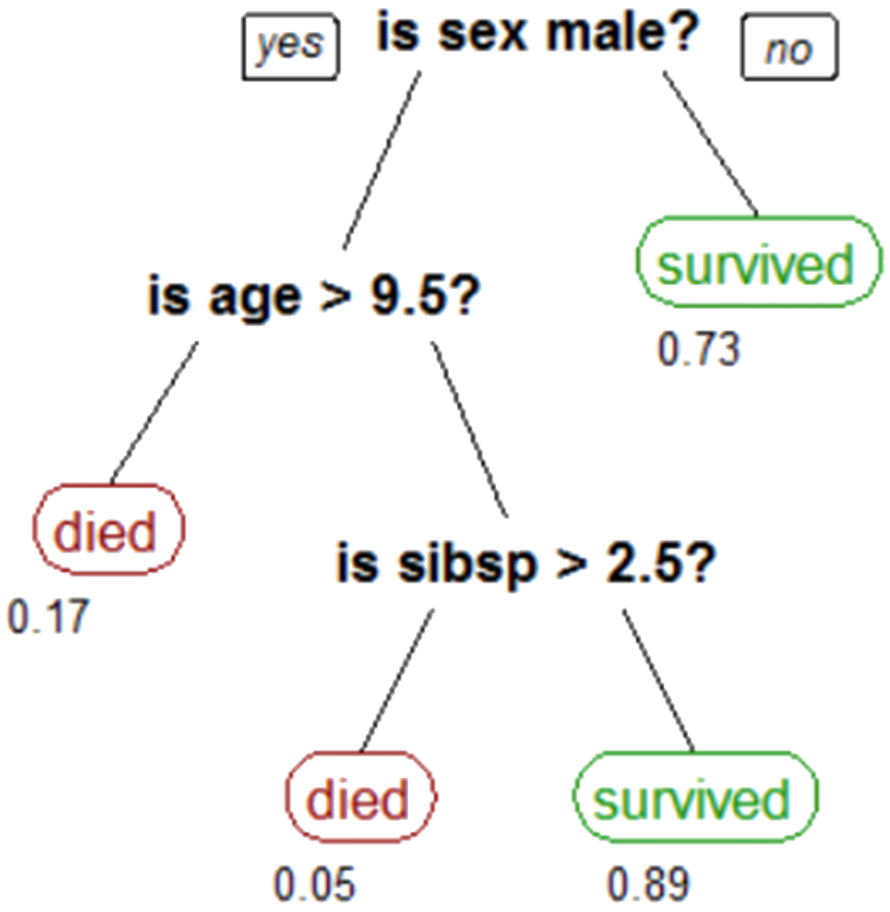
Дерева ухвалення рішень (приклад 1)

Приклад: Необхідно провести медичну діагностику. Лікар, що проводить цю діагностику, знає тільки 2 захворювання — ангіна й грип.



Дерева ухвалення рішень (приклад 2)

Інший приклад — виживе або не виживе той або інший пасажир Титаніка:



Дерева рішень

- Дерово рішень не обов'язково повинне бути бінарним
- У кожній **внутрішній вершині** записана умова
- В кожному **листку** написано прогноз

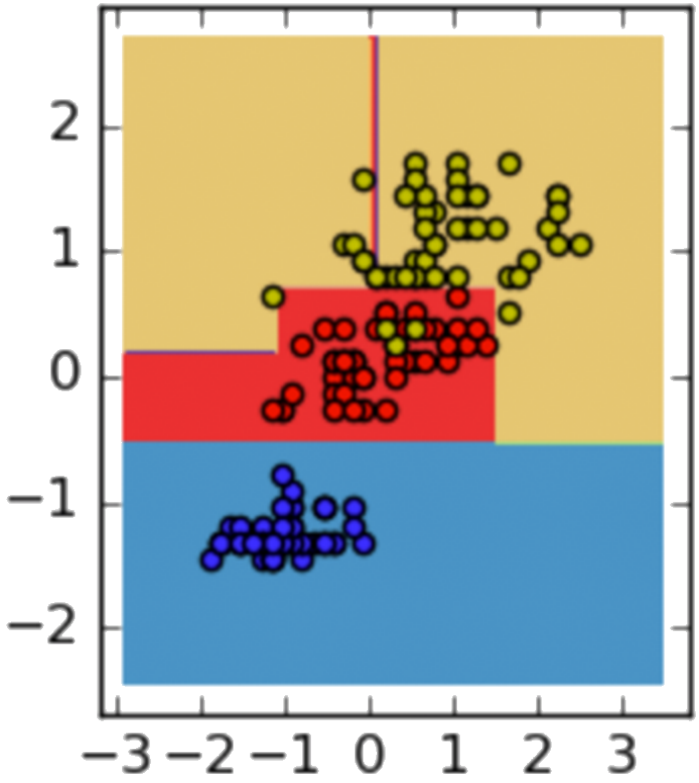
  - Найбільш поширineй варіант умови $[x^j \leq t]$

Прогноз в листку:

- дійсним числом, якщо вирішується **задача регресії**
- клас, або розподіл імовірностей класів, якщо вирішується **задача класифікації**

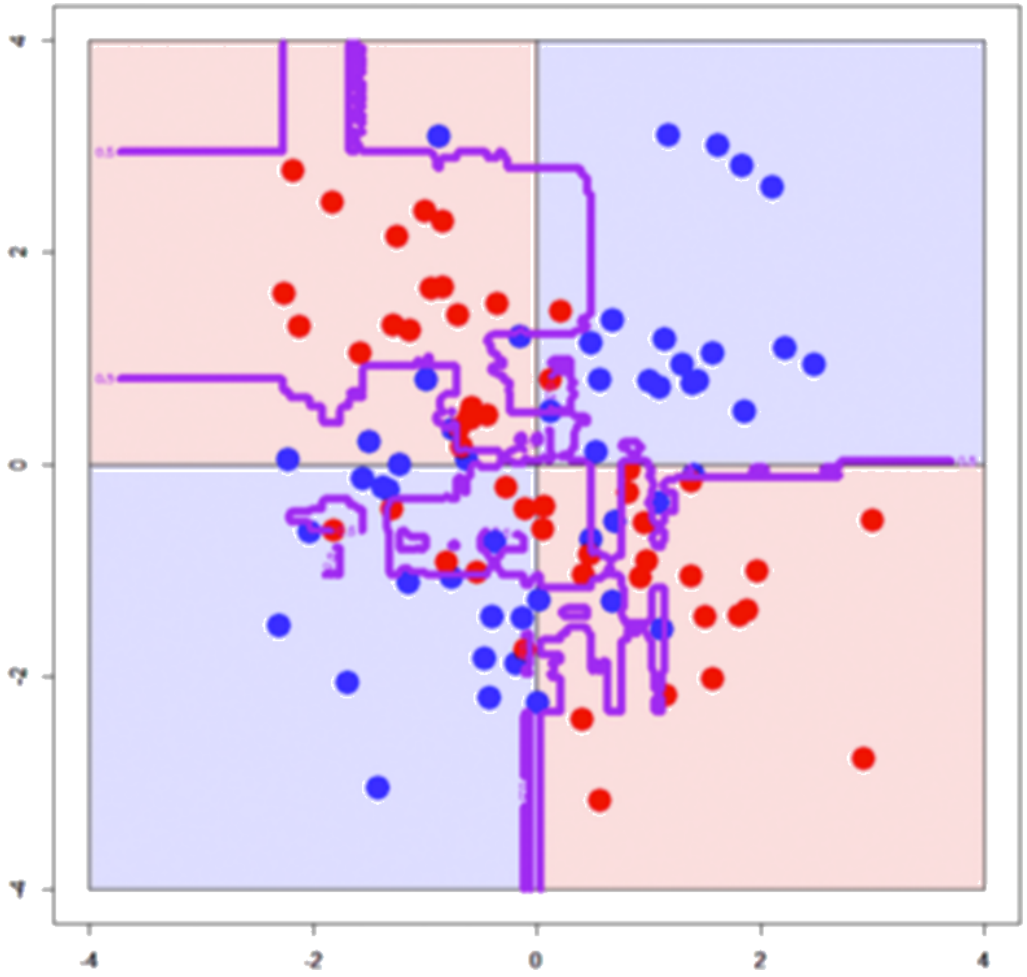
Дерева рішень в задачі класифікації

Розглянемо задачу класифікації із двома ознаками й трьома класами.



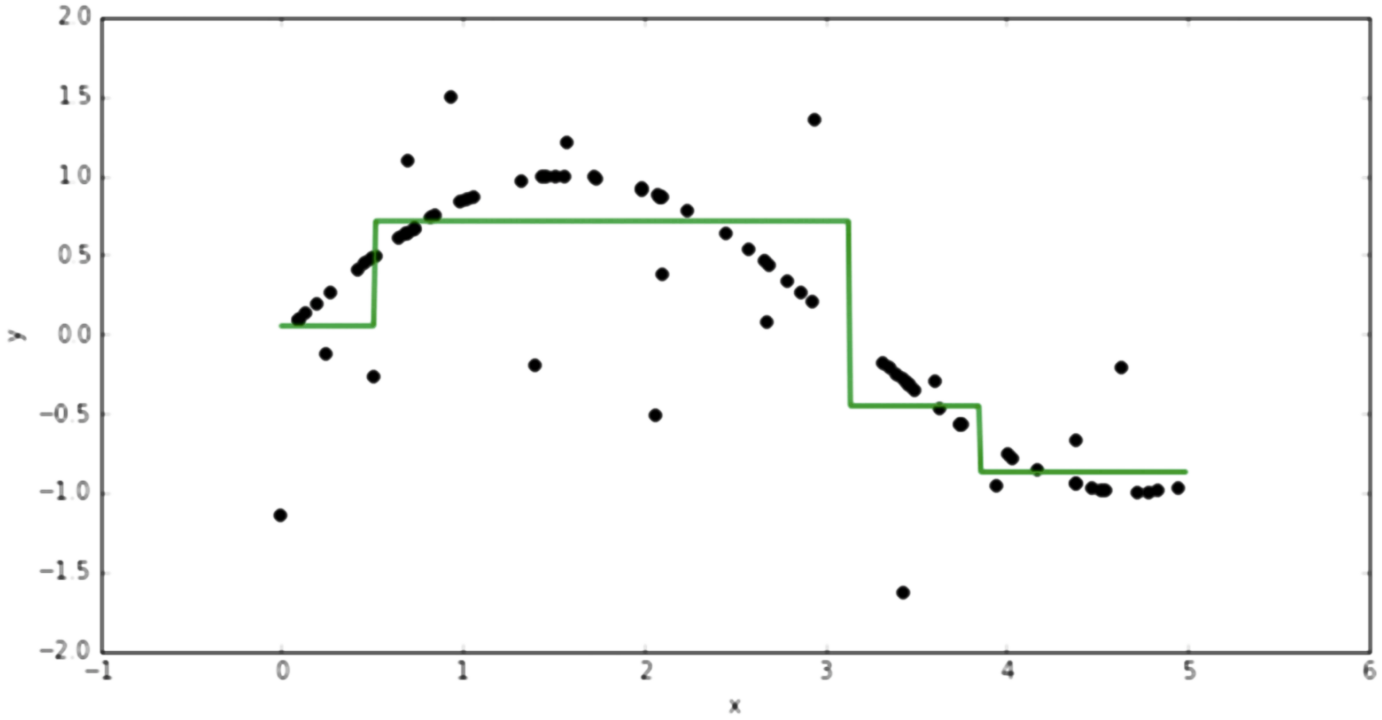
Поділяюча поверхня кожного класу **кусочно-стала**, і при цьому кожна сторона поверхні паралельна осі координат, тому що кожна умова порівнює значення однієї ознаки з порогом.

Дерево рішень може **перенавчитися**: його можна зробити настільки глибоким, що кожний листок вирішального дерева буде відповідати рівно одному об'єкту навчальної вибірки.

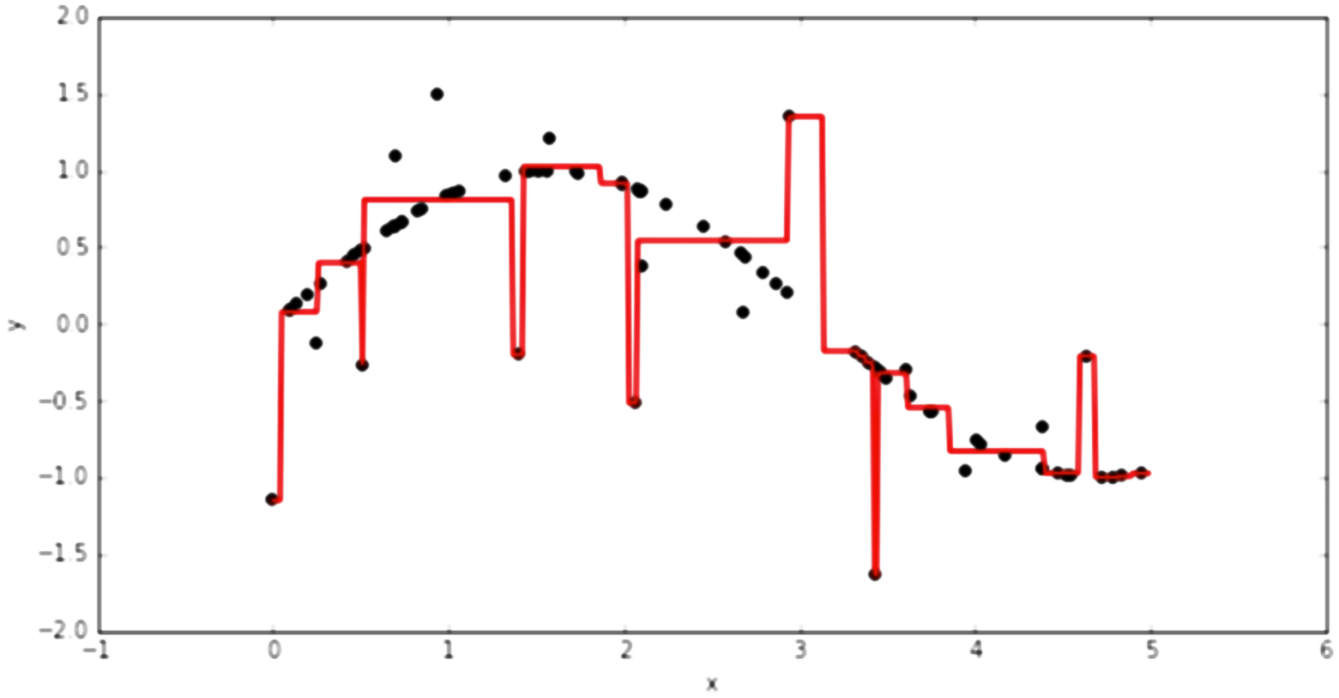


Дерева рішень в задачі регресії

Вирішуємо задачу регресії з однією ознакою (не дуже глибоке дерево)



При збільшенні глибини дерева функція, роздільна функція має такий вигляд:



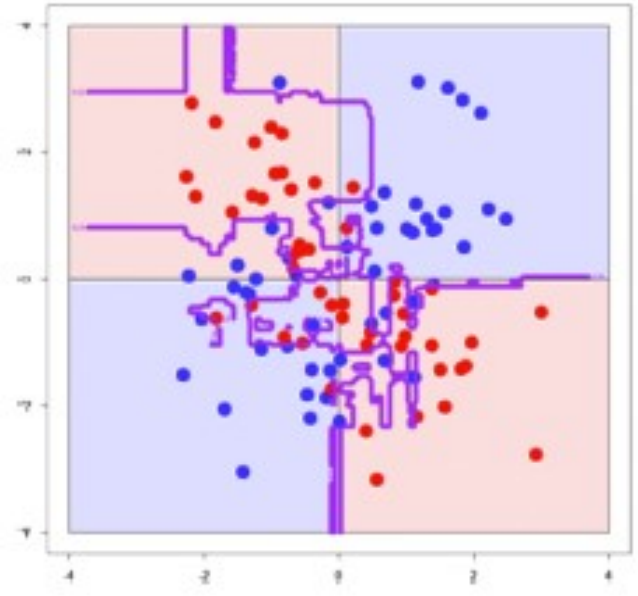
Підсумок

- Дерева рішень послідовно перевіряють прості умови
- Інтерпретовані
- Дозволяють відновлювати нелінейні залежності
- Легко перенавчаються

§56 Навчання дерев ухвалення рішень

Перенавчання дерев

Дерева рішень дуже легко перенавчаються. Можна побудувати дерево, у якого кожний листок буде відповідати одному об'єкту навчальної вибірки.



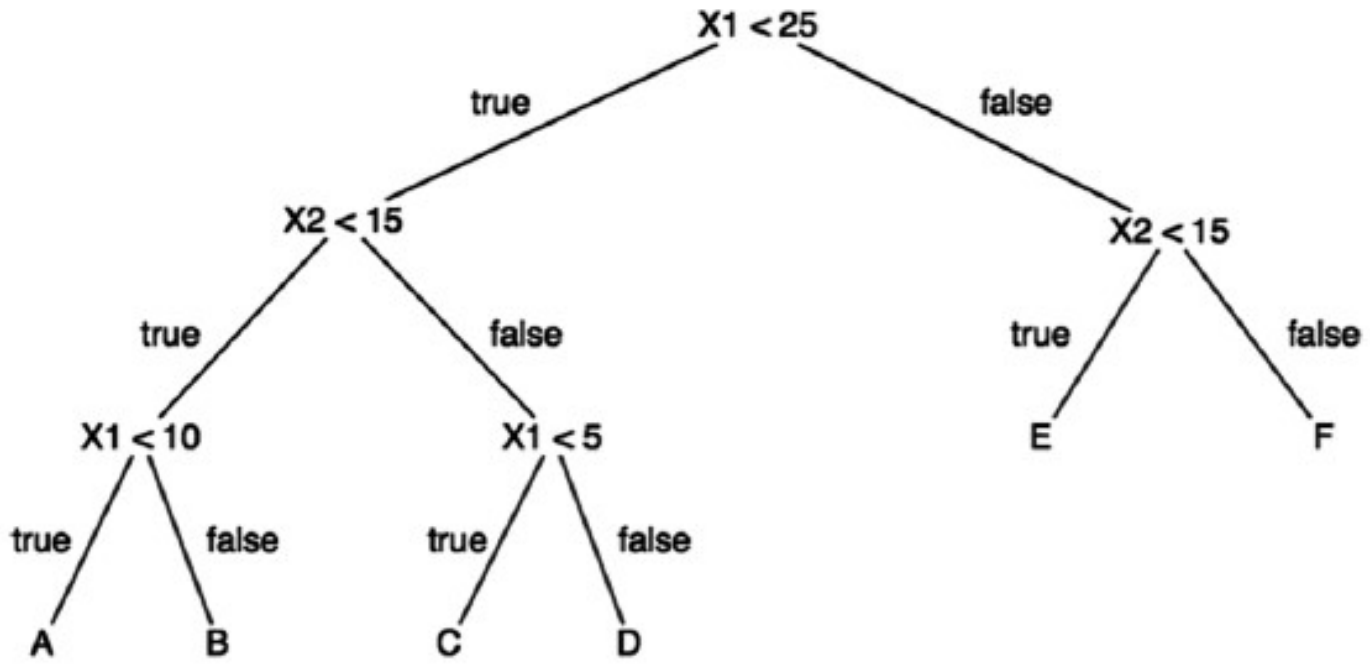
- Дерево може досягти нульової помилки на будь-якій вибірці
- Боротьба з перенавчанням: мінімальне дерево серед всіх з нульовою помилкою
- NP-повна (nondeterministic polynomial-time) задача

У теорії складності **обчислень задача є NP-повною**, коли:

- це задача, для якої правильність кожного рішення може бути перевірена за швидко поліноміальний час, а алгоритм пошуку може знайти рішення, **перебравши всі можливі рішення**.
- цю задачу можна використовувати для моделювання **іншої задачі NP-повної задачі**

Назва «NP-complete» є скороченням від «**nondeterministic polynomial-time complete**». У цій назві «недетермінований» означає **недетерміновану машину Тюрінга**. Поліноміальний час відноситься до часу, який вважається «швидким» для детермінованого алгоритму для перевірки одного рішення або для недетермінованої машини Тьюринга для виконання всього пошуку. «Повний» відноситься до властивості здатності симулювати все в тому самому класі складності.

Жадібний спосіб побудови



Спосіб побудови дерева рішень: спочатку вибирається корінь, що розбиває вибірку на дві. потім розбивається кожна з підвибірок й так далі. Дерево гілкується доти, поки цього не буде досить. На кожному етапі **порог вибирають так, щоб на цьому етапі розв'язку досякався максимум ефектиності (жадібний)**.

Як розбити вершину на дві?

Нехай у вершину *m* потрапило множину X_m об'єктів з навчальної вибірки

- Мінімізувати даний критерій похибки $Q(X_m, j, t)$ умови $[x^j \le t]$
- Шукаємо найкращі j, t перебором (*j* – номер ознаки, *t* – поріг)

$$Q(X_m, j, t) \to \min_{j,t}$$

Після того, як параметри були обрані

- Множина об'єктів з навчальної вибірки розбивається на дві множини

$$X_\ell = \{x \in X_m | [x^j \le t]\}$$
$$X_r = \{x \in X_m | [x^j > t]\}$$

- Цю процедуру можна продовжити для кожної з дочірніх вершин

Такий процес рано або пізно повинен зупинитися

Умови (критерій) зупинки:

- Якщо у вершину потрапив тільки один об'єкт навчальної вибірки
- Всі об'єкти належать одному класу (у задачах класифікації).
- Глибина дерева досягла певного значення.

Прогноз, якщо якась вершина була оголошена листком дерева:

- У задачі регресії, якщо функціонал — середньоквадратична помилка:

$$a_m = \frac{1}{|X_m|} \sum_{i \in X_m} y_i$$

- У задачі класифікації оптимально повертати той клас, що найбільш популярний серед об'єктів в X_m:

$$a_m = \underset{y \in \mathbb{Y}}{\operatorname{argmax}} \sum_{i \in X_m} [y_i = y]$$

- Якщо потрібно вказати ймовірності класів, їх можна вказати як частку об'єктів різних класів в X_m:

$$a_{mk} = \frac{1}{|X_m|} \sum_{i \in X_m} [y_i = k]$$

§57 Критерії інформативності

Як можна вибирати оптимальну розбивку при побудові дерева рішень?

Вибір критерію помилки

Нехай у вершину *m* потрапило множину X_m об'єктів з навчальної вибірки

- Мінімізуємо даний критерій похибки $Q(X_m, j, t)$ умови $[x^j \le t]$
- Шукаємо найкращі j, t перебором

$$Q(X_m, j, t) \to \min_{j,t}$$

- Множину об'єктів з навчальної вибірки розбиваємо на дві виходячи найкращих параметрів

$$X_\ell = \{x \in X_m | [x^j \le t]\}$$
$$X_r = \{x \in X_m | [x^j > t]\}$$

Критерій помилки:

$$Q(X_m, j, t) = \frac{|X_\ell|}{|X_m|} H(X_\ell) + \frac{|X_r|}{|X_m|} H(X_r)$$

Разброс ответов в левом листе

$$Q(X_m, j, t) = \frac{|X_\ell|}{|X_m|} H(X_\ell) + \frac{|X_r|}{|X_m|} H(X_r)$$

Доля объектов в листьях

Функція $H(X)$ — критерій інформативності: його **значення повинне бути тим менше, ніж менше розкид відповідей в *X***.

Задача регресії

У випадку регресії розкид відповідей — це дисперсія, тому:

$$H(X) = \frac{1}{|X|} \sum_{i \in X} (y_i - \bar{y}(X))^2$$

,

де

$$\bar{y}(X) = \frac{1}{|X|} \sum_{i \in X} y_i$$

Критерій інформативності Джини

Задача класифікації

Доля об’єктів класу k у вибірці X:

$$p_k = \frac{1}{|X|} \sum_{i \in X} [y_i = k]$$

Критерій інформативності Джини формулюється так

$$H(X) = \sum_{k=1}^{K} p_k (1 - p_k)$$

Його оптимум (мінімум) досягається тільки в тому випадку, коли всі об'єкти в X відносяться до одного класу.

$$p_1 = 1, p_2 = \ldots = p_k = 0, \text{ то } H(X) = 0$$

Одна з **інтерпретацій критерія Джини —** це ймовірність помилки випадкового класифікатора.

p_k – ймовірність отримати клас *k*.

Ентропійний критерій інформативності

Ще один критерій інформативності – ентропійний критерій:

$$H(X) = -\sum_{k=1}^{K} p_k \ln p_k$$

Пориймається, що $0 \cdot \ln(0) = 0$.

Ентропійний критерій має цікавий **фізичний зміст**: показує, наскільки розподіл класів в *X* відрізняється від виродженого.

Ентропія у **випадку виродженого розподілу дорівнює** 0: такий розподіл характеризується **мінімальним можливим ступенем хаосом**.

$$p_1 = 1, p_2 = \ldots = p_k = 0, \text{ то } H(X) = 0$$

Навпроти, **рівномірний розподіл найбільш хаотичний**, і йому відповідає максимальна ентропія.

§58 Критерій зупинки й стрижка дерев

Критерій зупинки

- Розбивати вершину, чи робити її листком?
- Які є способи боротьби з перенавчанням?

Критерій зупинки використовується, щоб прийняти рішення: **розбивати вершину далі або зробити листок**.

- Усі об’єкти відносяться до одного класу
- У вершину попало $\leq n$ об’єктів
- Коли $n = 1$ – маємо максимально перенавчене дерево
- n повинно бути таким, що побудувати надійний прогноз
- Рекомендація: $n = 5$
- Обмеження на глибину дерева

Стрижка дерев

- Будується дерево рішень максимальної складності й глибини, доти, поки в кожній вершині не виявиться по 1 об'єкту навчальної вибірки.
- Після цього починається «стрижка», тобто видалення листів у цьому дереві за певним критерієм.
- Наприклад, можна стригти доти, поки поліпшується якість деякої відкладеної вибірки.
- Існує думка, і це підкріплено багатьма експериментами, що стрижка працює набагато краще, ніж прості критерії

- Стрижка - дуже ресурсномістка процедура,
- Має сенс використовувати тільки для одного дерева
- Дерева на сьогоднішній день майже не використовуються, вони бувають потрібні тільки для побудови композиції й об'єднання великої кількості дерев в один алгоритм (**в ліси дерев**). У випадку з композиціями такі складні підходи до боротьби з перенавчанням уже не потрібні, тому що існують досить прості критерії зупинки.

§59 Дерева ухвалення рішень й категоріальні ознаки

- Раніше використовували наступна умова у вершині кожного дерева:

$$[x^j \leq t]$$

Можна використовувати тільки для дійсних або бінарних ознак.

- **Що робити, коли ознаки категоріальні?**

N-арні дерева

Потрібно будувати *n*-арні дерева, тобто такі дерева, що з кожної вершини можуть виходити по *n* ребер.

- Для x^j — категоріальної ознака, що вона може приймати значення $\{c_1, \ldots, c_n\}$
- Розбиваємо вершину на n дочірних вершин
- В j дочірню вершину попадають об'єкти з $x^j = c_i$.

- Розбиваємо X_m на n частин: $X_1, X_2, .., X_n$
- Критерій помилки такої розбивки будується за аналогією з випадком бінарного дерева:

$$Q(X_m, j) = \sum_{i=1}^{n} \frac{|X_i|}{|X_m|} H(X_i) \to \min_j$$

- Вибираємо із усіх розбиттів таке, у якому наменшою функція похибки $Q(X_m, j, t)$ або $Q(X_m, j,)$

- Часто категоріальні ознаки мають велике число n
- Маємо високу ймовірність перенавчання
- Гарно працює для великих вибірок

Бінарні дерева з розбивкою множини значень

Інший підхід дозволяє **не переходити до *n*-арних дерев і продовжувати працювати з бінарними деревами**.

- Нехай необхідно зробити розбивку вершини *m* за категоріальною ознакою x^j $(C = \{c_1, ..., c_n\})$.
- Розбиваємо множину значень категоріальної ознаки на дві непересічні підмножини:

$$C = C_1 \cup C_2, \qquad C_1 \cap C_2 = \varnothing.$$

- Після того, як така розбивка побудована, умова в даній вершині буде виглядати просто:

$$[x^j \in C_1]$$

Ця умова перевіряє, у яке з підмножин попадає значення ознаки в цей момент на даному об'єкті.

Як саме потрібно розбивати множину *C*?

Повна кількість можливих розбивок множини на дві підмножини — 2^n. Є спосіб, що дозволяє уникнути повного перебору й, більше того, працювати з категоріальною ознакою як з дійсною.

- Для цього можливі значення категоріальної ознаки $c_{(1)},...,c_{(n)}$ і заміняються на ***натуральні числа 1,...,n***.
- Після цього ***з цією ознакою можна вже працювати як з дійсною, а значення порога t буде визначати поділ множини C на дві підмножини***.

- Сортувати значення категоріальної ознаки у випадку задачі бінарної класифікації потрібно за таким принципом:

$$\frac{\sum_{i\in X_m}[x_i^j = c_{(1)}][y_i = +1]}{\sum_{i\in X_m}[x_i^j = c_{(1)}]} \leq \ \ldots \ \leq \frac{\sum_{i\in X_m}[x_i^j = c_{(n)}][y_i = +1]}{\sum_{i\in X_m}[x_i^j = c_{(n)}]}$$

Фактично, значення категоріальної ознаки ***сортуються за зростанням частки об'єктів*** +1 класу серед об'єктів вибірки X_m з відповідним значенням цієї ознаки.

- Сортування для задачі регресії сортування відбувається схожим чином, але обчислюється не частка об'єктів додатнього класу, а середня відповідь за усіма об'єктами, у яких значення категоріальної ознаки дорівнює *c*:

$$\frac{\sum_{i \in X_m} [x_i^j = c_{(1)}] y_i}{\sum_{i \in X_m} [x_i^j = c_{(1)}]} \leq \ \ldots \ \leq \frac{\sum_{i \in X_m} [x_i^j = c_{(n)}] y_i}{\sum_{i \in X_m} [x_i^j = c_{(n)}]}.$$

Головна особливість такого підходу полягає в тому, що отриманий результат ***повністю еквівалентний результату, який можна було б одержати в результаті повного перебору***. ***Ця умова працює для критерію Джини, MSE і ентропійного критерію***.

§60 Дерева ухвалення рішень в sklearn

Див. JNotebook «4_1_sklearn_decision_trees.ipynb»