Лекція 10-11
Практичні рекомендації до лінійних моделей. Підбір параметрів моделі

§46 Масштабування ознак

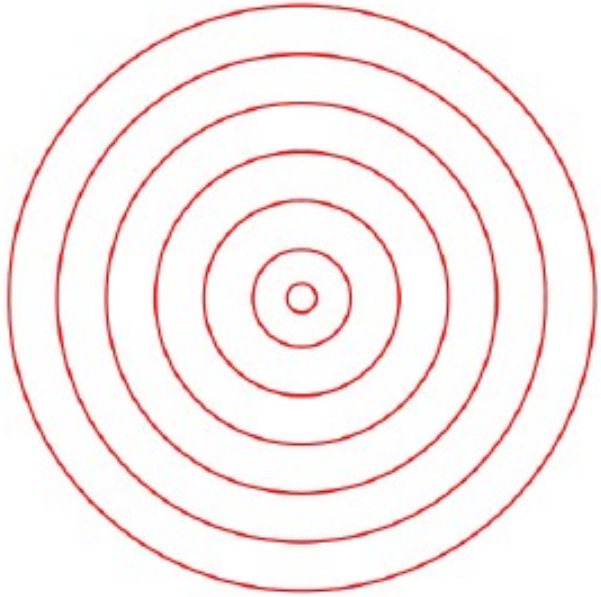
Приклад: необхідність масштабування

Розглянемо приклад: нехай необхідно знайти мінімум:

$$w_1^2 + w_2^2 \to \min_w$$

Відповідь очевидна — це точка (0, 0).

Під час пошуку цього мінімуму методом градієнтного спуску з початковою точкою ***w* = (1,1)**, вектор антиградієнта буде мати координати **(-2,-2)**.

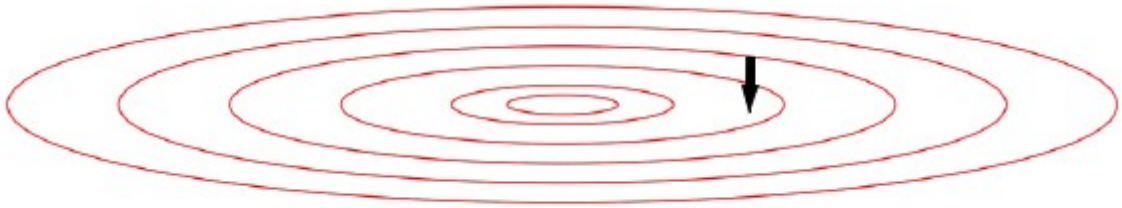


Це вектор проходить через точку мінімуму, а значить при правильно підібраному розмірі кроку, уже на першому кроці градієнтного спуска можна потрапити строго в точку мінімуму.

Змінимо функцію:

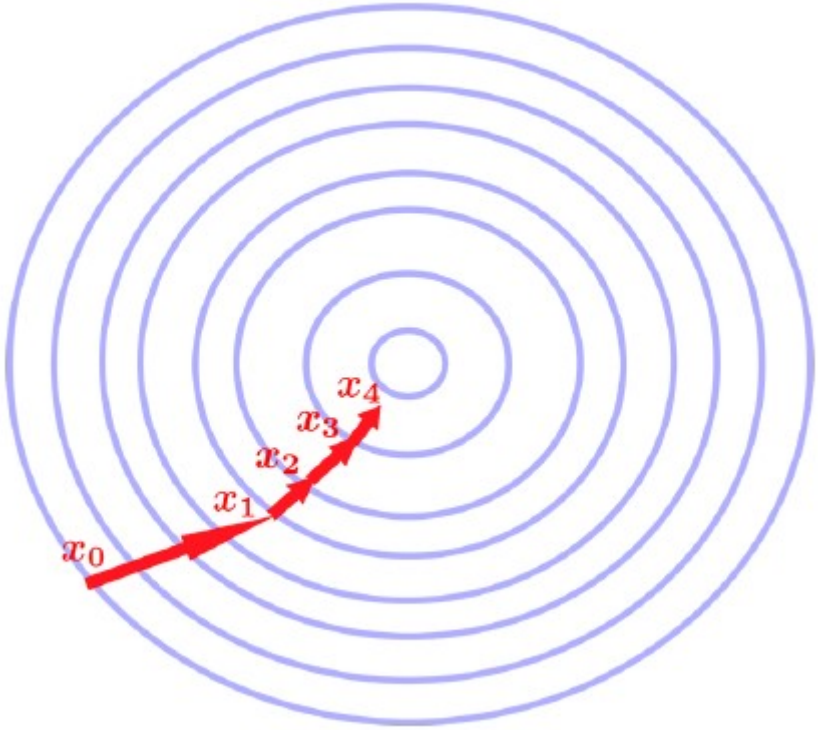
$$w_1^2 + 100\, w_2^2 \to \min_w$$

У цьому випадку лінії рівня являють собою еліпси, сильно витягнуті уздовж осі *x*.

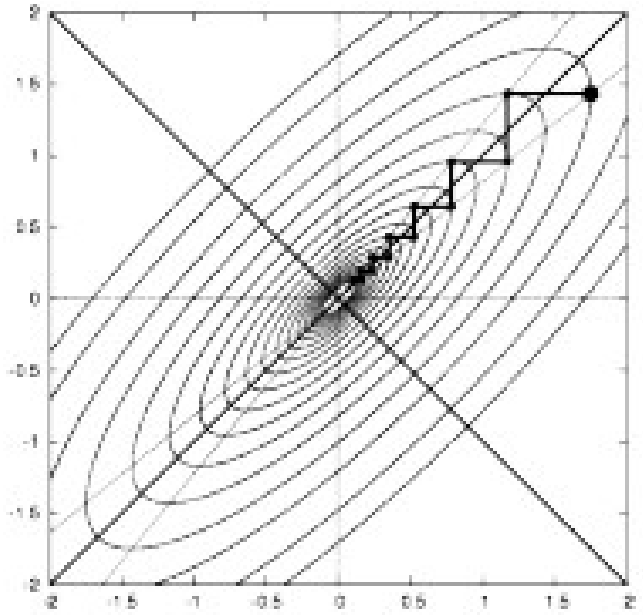


Якщо запустити градієнтний спуск із точки ***w* = (1, 1)**, вектор антиградієнта в цій точці буде мати координати **(-2, -200)**. Вектор буде дивитися майже строго вниз і проходити повз точку мінімуму функції. Більше того, існує шанс на наступній ітерації піти ще далі від мінімуму.

Якщо провести маштабування, то маємо такий результат



Без масштабування



Висновок:

- градієнтний спуск працює добре, якщо лінії рівня функції схожі на кола. У цьому випадку, звідки б ви не почали, вектор градієнта буде завжди дивитися убік мінімуму функції.
- якщо ж лінії рівня схожі на еліпси, напрямок антиградієнта буде слабко збігатися з напрямком убік мінімуму функції. Збіжність буде повільна, і більше того, існує ризик розбіжності ітеративного процесу.

Масштабування вибірки

Приклад: потрібно передбачити, чи буде виданий грант за заявкою. Заявка характеризується **двома ознаками** - скільки вже **успішних заявок** було в цього заявника й **рік народження заявника**.
Масштаби цих ознак суттєво різні: успіших заявок – одиниці, рік народження – тисячі.
Розглянемо **два способи масштабування.**

Перший спосіб називається стандартизацією.

Обчислимо середні значення ознак і стандартні відхилення:

$$\mu_j = \frac{1}{\ell}\sum_{i=1}^{\ell} x_i^j \qquad \sigma_j = \sqrt{\frac{1}{\ell}\sum_{i=1}^{\ell}(x_i^j - \mu_j)^2}$$

Щоб виконати ***стандартизацію ознаки***

$$x_i^j := \frac{x_i^j - \mu_j}{\sigma_j}$$

Другий підхід називається «масштабування на відрізок [0, 1]».

Обчислимо максимальне й мінімальне значення кожної ознаки:

$$m_j = \min(x_1^j, ..., x_\ell^j)$$

$$M_j = \max(x_1^j, ..., x_\ell^j)$$

Значення кожної ознаки на конкретному об'єкті перетворюється:

$$x_i^j := \frac{x_i^j - m_j}{M_j - m_j}$$

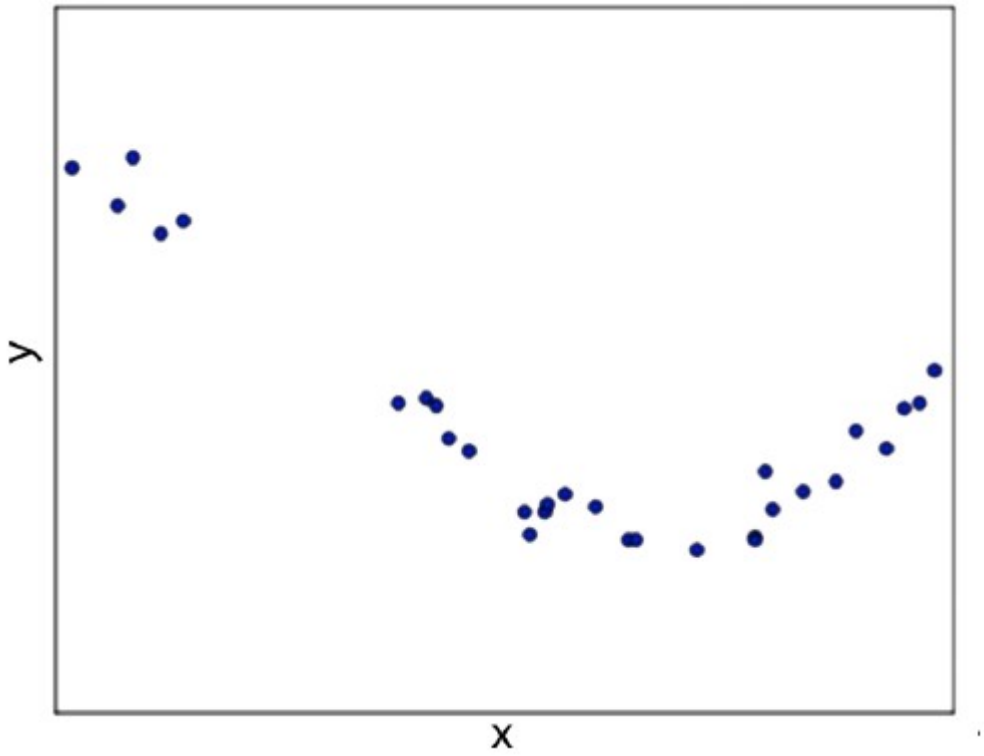
Тоді мінімальне значення ознаки відображається в нуль, а максимальні — в 1, тобто ознаки масштабуються на відрізок [0, 1].

§47 Нелінійні залежності

Розглянемо спрямовуючі простори, які дозволяють відновлювати нелінійні залежність за допомогою лінійних моделей. Спочатку приклад.

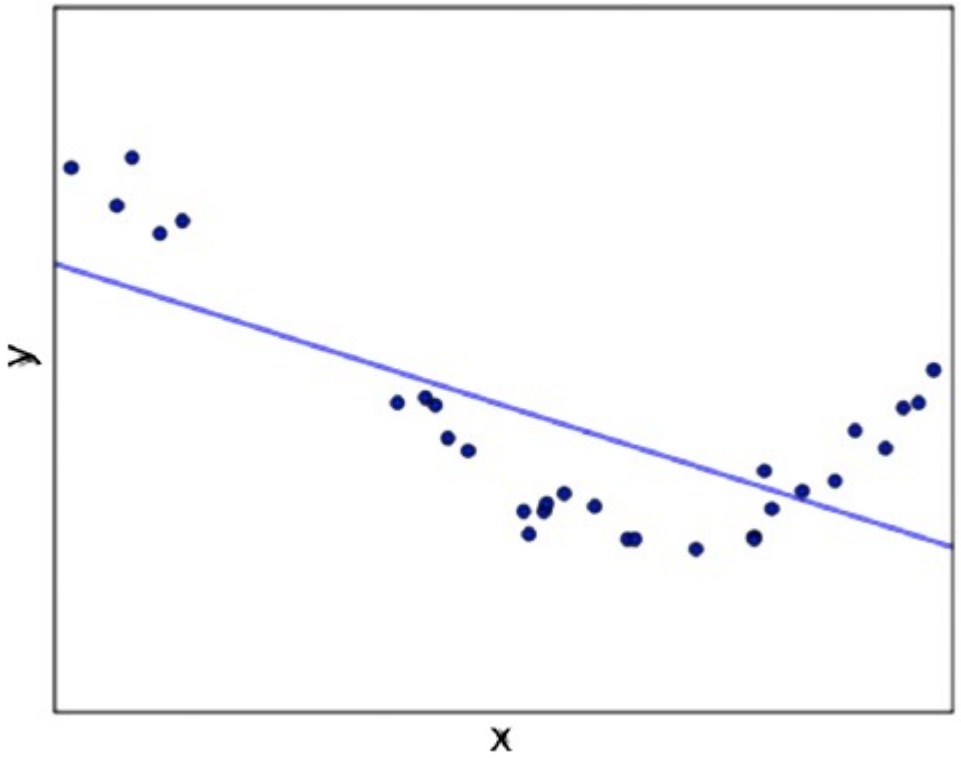
Нелінійна задача регресії

Нехай вирішується задача регресії з одною ознакою, відкладеною вздовж осі *x*. За цією ознакою потрібно відновити цільову змінну *y*.



Лінійна модель

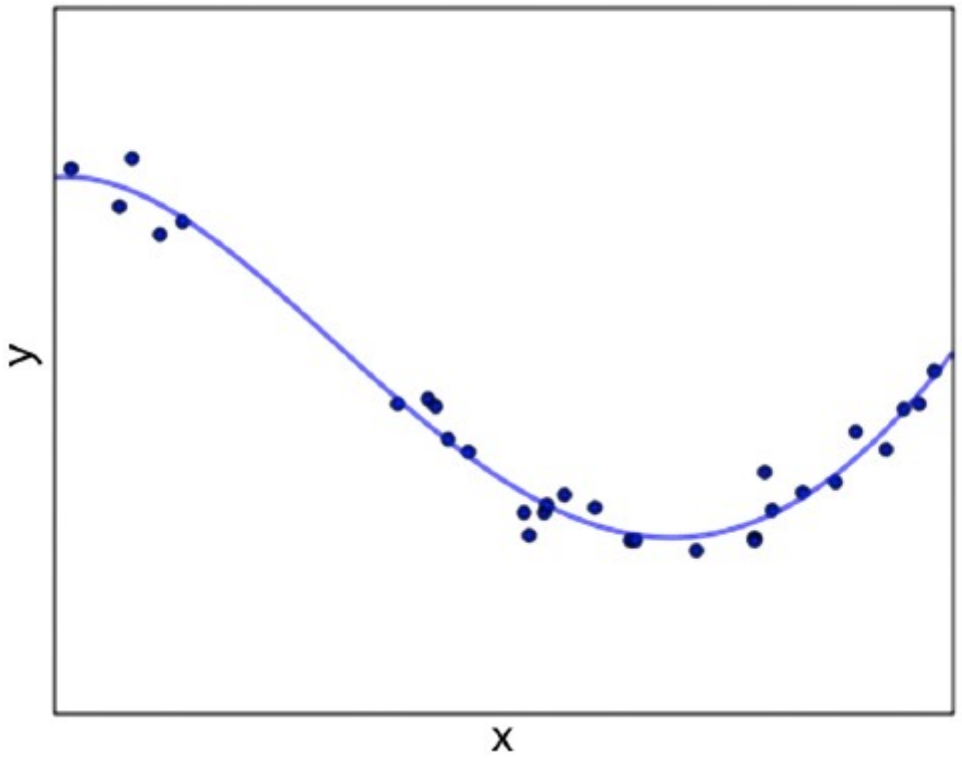
$$a(x) = w_0 + w_1 x$$



Якість є незадовільною.

Нелінійна модель

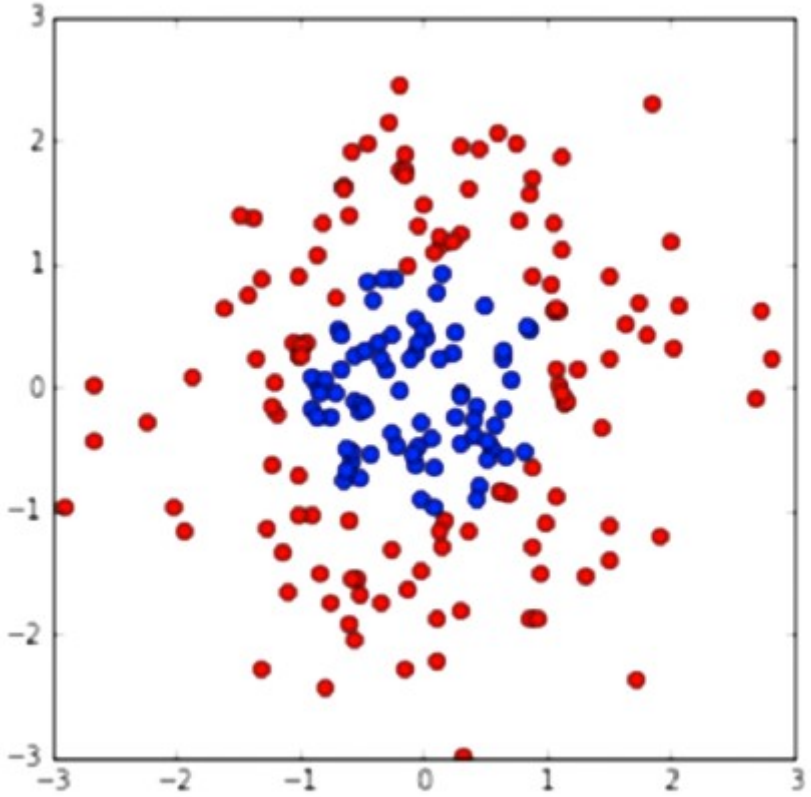
$$a(x) = w_0 + w_1 x + w_2 x^2 + w_3 x^3 + w_4 x^4$$



Фактично був зроблений **перехід до нового ознакового простору** із чотирьох ознак (x, x^2, x^3, x^4)замість одного, у якому була побудована лінійна модель. А на вихідному просторі ця модель уже нелінійна й відмінно описує дані.

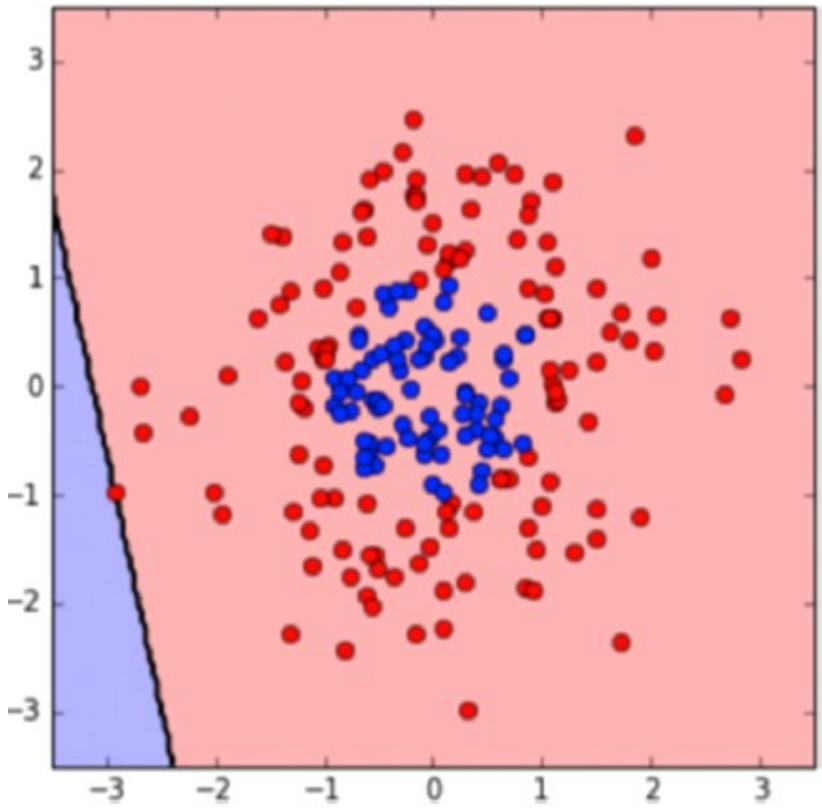
Задача класифікації з нелінійною границею

Інший приклад – вирішується задача класифікації із двома ознаками



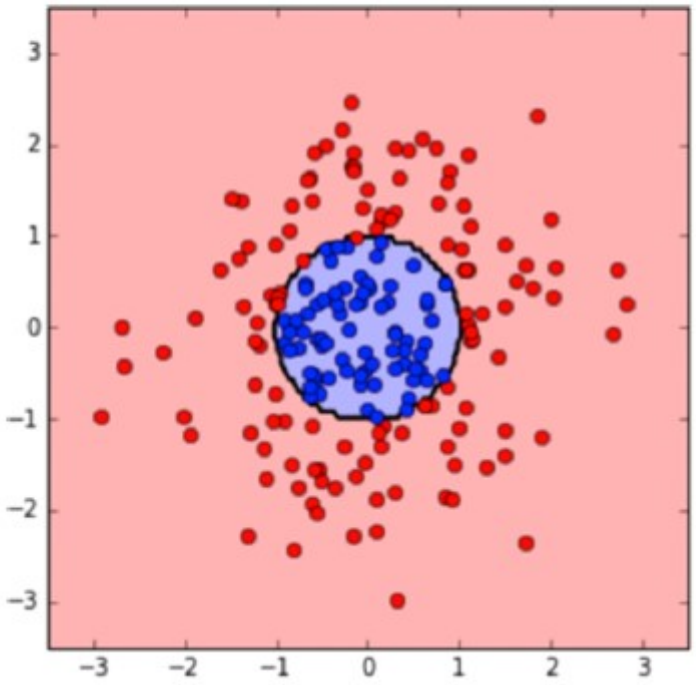
Бачимо, що **роздільна поверхня тут не є лінійною**.

Лінійний класифікатор:



Тобто **всі об'єкти будуть віднесені до червоного класу**: це краще, що він може виконати, але зрозуміло, що це ні на що не годиться.

$$a(x) = \text{sign}(w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1 x_2 + w_4 x_1^2 + w_5 x_2^2)$$



Тут також у новому просторі $(x_1, x_2, x_1 x_2, {x_1}^2, {x_2}^2)$ була побудована лінійна модель, що є **нелінійною у вихідному ознаковому просторі**.

Спрямляючий простір

Спрямляючим простором ознак називається такий простір, у якому задача добре вирішується лінійною моделлю. Спрямляючий простір може, у тому числі, бути побудовано:

- Через додавання квадратичних ознак, тобто коли до вихідних ознак додаються їх квадрати й попарні добутки:

$$(x_1, ..., x_d) \to (x_1, ..., x_d, x_1^2, ..., x_d^2, x_1 x_2, ..., x_{d-1} x_d)$$

  Число ознак збільшується на порядок.

- Через додавання поліноміальних ознак:

$$(x_1, ..., x_d) \to (x_1, ..., x_d, ..., x_i x_j, ..., x_i x_j x_k, ...)$$

  Цей спосіб варто використовувати тільки, коли об'єктів досить багато, тобто ризик перенавчання мінімальний.

- Через логарифмування (або будь-які інші нелінійні функції):

$$x_i \to \ln(x_i + 1)$$

$$x_i \to \ln(|x_i| + 1)$$

Розглянемо приклад про вартість книг в інтернет-магазині.

У цьому випадку **ознакою буде вартість книги**, значення якого для більшості книг складе **кілька сотень гривень**.

Але існують і зустрічаються досить часто **дуже дорогі книги**, так що розподіл цієї ознаки буде мати «**важкий правий хвіст**».

Відомо, що лінійні моделі **погано застосовні в такому випадку** й працюють набагато краще, якщо розподіл ознак близько до нормального. **Щоб розподіл ознаки «із хвостом» виконати більше близьким до нормального, потрібно прологарифмувати ці ознаки**.

Інколи такі ознаки просто видаляють.

§48 Робота з категоріальними ознаками

Приклади категоріальних ознак:

- Місто
- Колір
- Тарифний план
- Марка автомобіля
- і так далі...

Особливість категоріальних ознак полягає в тому, що **це елементи деякого неупорядкованої множини, і не можна говорити, що якесь значення більше або менше іншого.**

В лінійних моделях потрібно брати значення ознаки, множити на вагу, а потім складати з іншими числами, і цю операцію не можна робити зі значеннями категоріальних ознак.

Категоріальні ознаки потрібно спочатку перетворити, щоб їх можна було використовувати в лінійних моделях.

Бінарне кодування

- Категоріальна ознака $f_j(x)$
- *n* можливих значень
- пронумероємо ці значення: $c_1, c_2, ..., c_n$
- введемо *n* нових бінарних ознак: $b_1(x), b_2(x), ..., b_n(x)$
- $b_i(x) = [f_j(x) = c_i]$

У результаті одна категоріальна ознака заміняється *n* бінарними ознаками.

Бінарне кодування (приклад)

Ознака приймати три значення: {синій, зелений, червоний}.

Дано три об'єкти x_1, x_2, x_3, на яких значення категоріальної ознаки:

$f_j(x_1)$ = синій, $f_j(x_2)$ = червоний, $f_j(x_3)$ = синій.

Оскільки категоріальна ознака приймає 3 значення, буде потрібно 3 бінарних ознаки, щоб його закодувати. У результаті виходить наступна матриця (кожному об'єкту відповідає свій рядок, а кожному стовпцю - своє значення ознаки):

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$

Бінарне кодування: нові значення

Часто виникає наступна проблема. При кодування тестової вибірки може зустрітися об'єкт, на якому **категоріальна ознака приймає нове (*n* + 1)-е значення**, що до цього не зустрічалося в навчальній вибірці.

У цьому випадку **логічним підходом буде не додавати нову ознаку** (оскільки це складно реалізується), а **просто прирівняти до нуля всі існуючі ознаки**. Дійсно, за змістом бінарних ознак, чи приймає категоріальну ознаку значення c_i, $i \in \{1,2,...,n\}$, кожний з них у такому випадку повинен дорівнювати нулю.

§49 Незбалансовані дані

Незбалансована вибірка

Задача називається незбалансованою, якщо об'єктів одного класу **істотно менше, ніж об'єктів інших класів (менше 10%)**.

Приклади:

- Передбачення різких стрибків курсу долара (якщо визначення різкого стрибка має на увазі сильні зміни, то прикладів таких змін за всю історію – одиниці)
- Медична діагностика (хворих, як правило, набагато менше, ніж здорових)
- Виявлення шахрайських транзакцій (яких істотно менше, ніж звичайних транзакцій)
- Класифікація текстів і так далі.

Проблема, пов'язана з незбалансованими вибірками:

- класифікатори мінімізують число неправильних відповідей і ніяк не враховують ціни помилок.
- ціна похибки на кожному класі не є однаковою
- може виникнути ситуація, коли **вигідніше віднести всі об'єкти до більшого класу**, не намагаючись якось виділити об'єкти маленького класу.

Інакше кажучи, при роботі з незбалансованими вибірками класифікатори виходять дуже погані з погляду точності або повноти.

Розглянемо декілька способів розв’язання цієї проблеми

Undersampling

undersampling — його основна ідея полягає в тому, що **частина об'єктів з великого класу викидаються з вибірки**.

У цьому випадку необхідно викинути більшу частину 1-го класу й половину об'єктів 3-го класи. Розміри класів приблизно зрівняються.

Oversampling

oversampling – протилежний попередній: у цьому випадку об'єкти маленьких класів дублюються, щоб вирівняти співвідношення класів.

5 разів необхідно продублювати об'єкти 2-го класу, а також продублювати випадкову половину 3-го класи.

Стратифікація

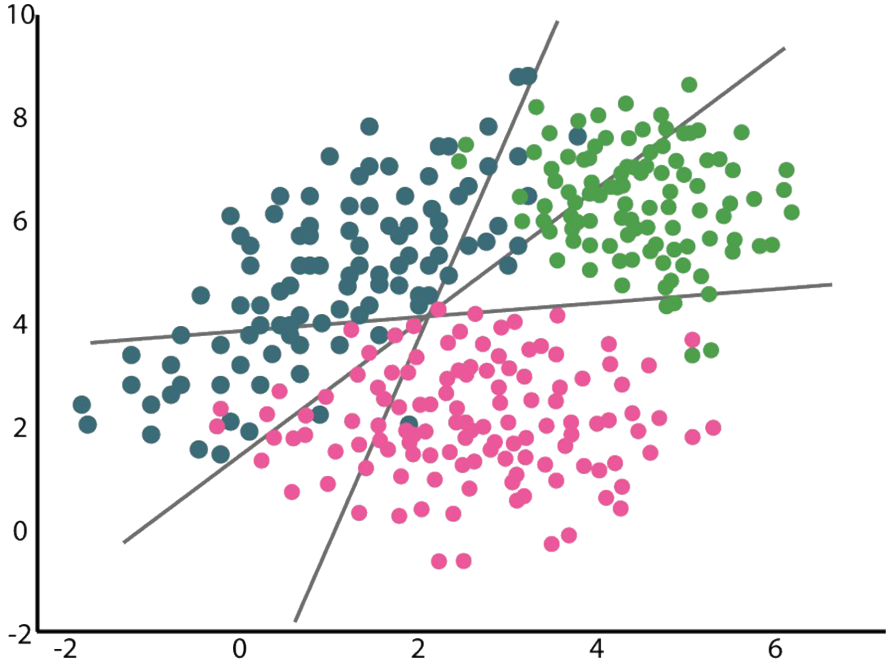
Проблема, наприклад, крос-валідації: якщо вибірка незбалансована, може вийти ситуація, що **в деякі блоки об'єкти якогось класу не потраплять взагалі.** При навчанні на цьому блоці виходить класифікатор, що ніколи не бачив один із класів.

Необхідно використовувати **стратифікацію**, тобто робити так, щоб **розподіл класів у кожному блоці приблизно збігалося з розподілом класів у вихідній вибірці**.

- Крос-валідація розбіває вибірку на блоки
- Потрібно розбивати так, щоб у кожному блоці зберігалось співвідношення класів

§50 Багатокласова класифікація

Як вирішувати задачі багато класової класифікації $Y = \{1,2,...,K\}$ за допомогою лінійних моделей?



У випадку бінарної класифікації підхід був простій — потрібно було знайти такий вектор ваг *w*, що вираз

$$a(x) = \text{sing} < w, x >$$

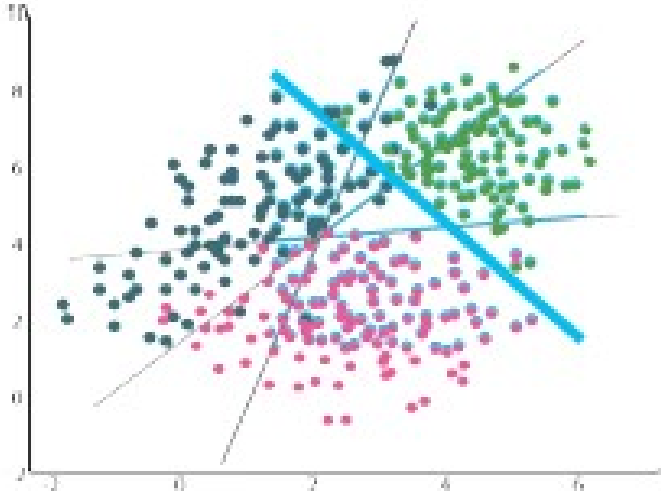
визначало б, якому класу цей об'єкт відноситься.

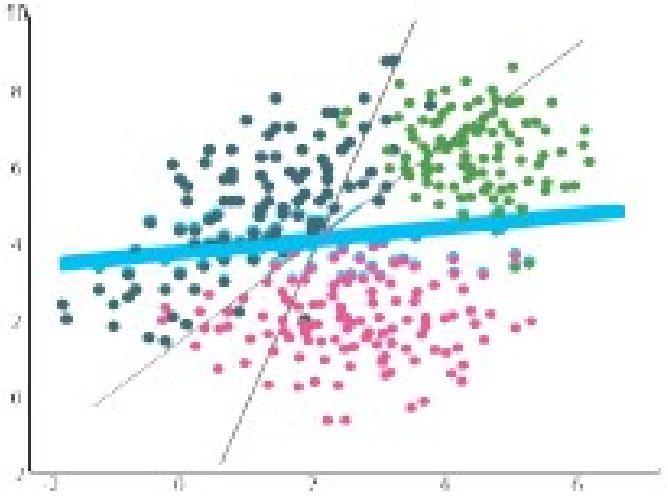
ONE-VS-ALL

Підхід називається «**один проти всіх**».

Для кожного класу буде будуватися свій бінарний класифікатор.

Задачею цього класифікатора буде відділення даного класу від всіх інших.





- K задач бінарної класифікації.
- для *k*-ої задачі створюємо специфічну вибірку (об'єкти x_i залишаються такими ж, а відповіді стають бінарними):

$$\boldsymbol{X = (x_i, [y_i = k])_{i=1}^{\ell}}$$

- побудований класифікатор, що відокремлює *k*-ий клас від всіх інших:

$$\boldsymbol{a_k(x) = \mathrm{sign}\langle \mathrm{w}_k, x\rangle}$$

- для багатокласового класифікатора можна використовувати наступний алгоритм:

$$\boldsymbol{a(x) = \underset{k\in\{1,\ldots,K\}}{\mathrm{argmax}}\ \langle \mathrm{w}_k, x\rangle}$$

буде повертати той клас *k*, для якого впевненість відповідного класифікатора (тобто значення відповідного скалярного добутку) найбільше.

Матриця помилок

Для аналізу того, наскільки добре працює багатокласовий класифікатор, зручно використовувати матрицю помилок.

	$y=1$	$y=2$	$\cdots$	$y=K$
$a(x)=1$	q_{11}	q_{12}	$\cdots$	q_{1K}
$a(x)=2$	q_{21}	q_{22}	$\cdots$	q_{2K}
$\cdots$	$\cdots$	$\cdots$	$\cdots$	$\cdots$
$a(x)=K$	q_{K1}	q_{K2}	$\cdots$	q_{KK}

Ця матриця дозволяє зрозуміти, які класи переплутуються найчастіше.
Також можна вимірювати вже відомі метрики якості, наприклад:

$$\mathbf{accuracy}(a, X) = \frac{1}{\ell}\sum_{i=1}^{\ell}[a(x_i) = y_i]$$

Можна знайти точність і повноту

- для задачі відділення певного класу від всіх інших класів;
- точність і повнота можуть бути усереднені, щоб одержати агреговані оцінки;
- можна знайти усереднену *F* -міру

§51 Підбір параметрів по сітці. Sklearn.grid_search

Див. JNotebook «3_1sklearn_grid_search.ipynb»

§52 Приклад:Sklearn_case_part1. Bike Sharing Demand

Див. JNotebook «3_2sklearn_case_part1.ipynb»

§53 Приклад:Sklearn_case_part2. Bike Sharing Demand

Див. JNotebook «3_2sklearn_case_part1.ipynb»

§54 Комп’ютерний проект 5: Попередня обробка даних і логістична регресія для завдання бінарної класифікації (КП05_Прізвище_Preprocessing_LR.ipynb)