

# **ОСНОВИ СИСТЕМ ШТУЧНОГО ІНТЕЛЕКТУ, НЕЙРОННИХ МЕРЕЖ та ГЛИБОКОГО НАВЧАННЯ**

## **Модуль 2. Навчання з вчителем**

### **Лекція 2.6.**

**Огляд методів класифікації. Бінарна  
класифікація. Оцінка якості класифікатора.**

# Класичний AI / Класичний ML

## Машинне навчання

```
graph TD; A[Машинне навчання] --> B[Навчання з вчителем]; A --> C[Навчання без вчителя]; A --> D[Навчання з підкріпленням]; B --> B1[Регресія]; B --> B2[Класифікація]; C --> C1[Кластерізація]; C --> C2[Зменшення розмірності]; D --> D1[Вживання у незнайомій обстановці];
```

The diagram illustrates the classification of Machine Learning into three main categories: Supervised Learning, Unsupervised Learning, and Reinforcement Learning. Supervised Learning is further divided into Regression and Classification, with Classification highlighted by a red dashed border. Unsupervised Learning is divided into Clustering and Dimensionality Reduction. Reinforcement Learning is associated with application in unfamiliar environments.

Навчання з  
вчителем

Регресія

Класифікація

Навчання без  
вчителя

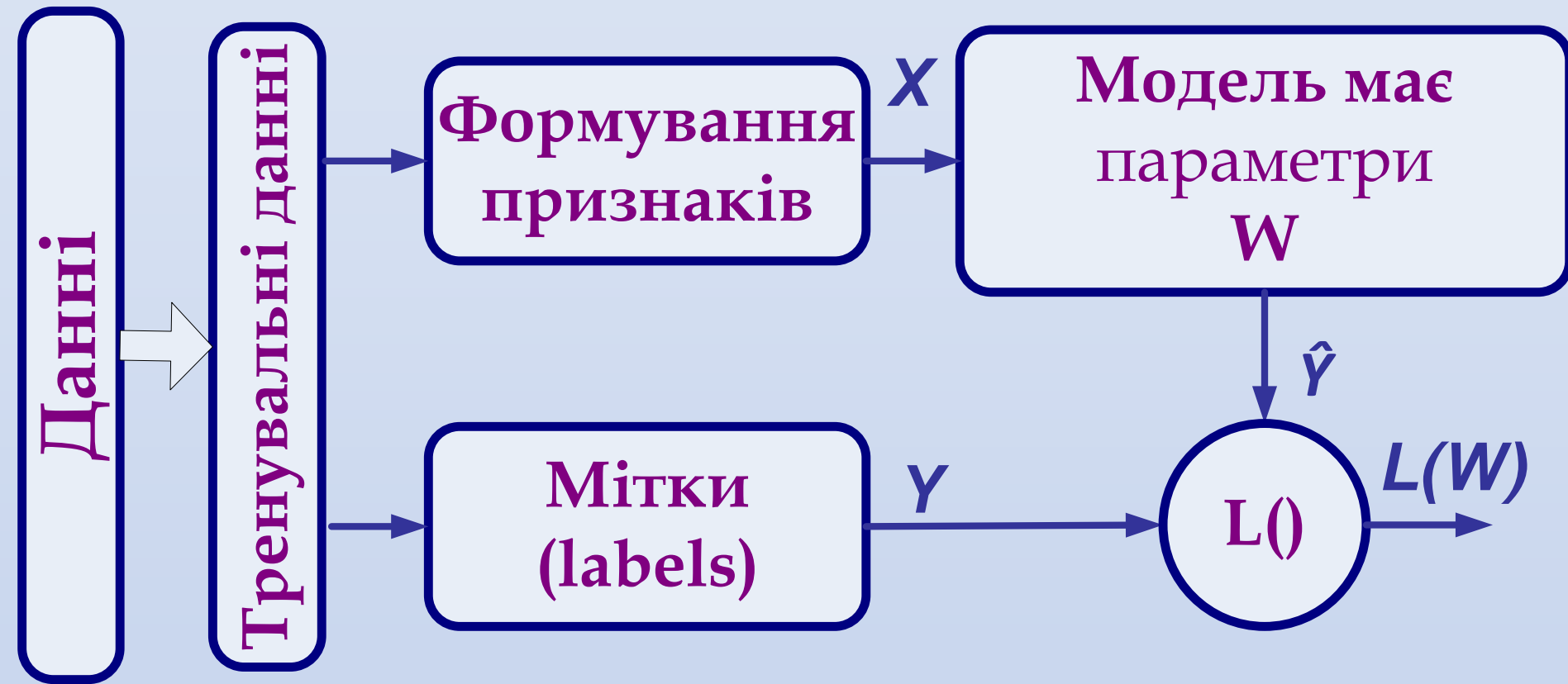
Кластерізація

Зменшення  
розмірності

Навчання з  
підкріпленням

Вживання у  
незнайомій  
обстановці

# Загальний процес ML з вчителем



# Класифікація

## Формально:

Маємо множину  $\mathbb{O}$  об'єктів  $o^{(j)}$ ,  $j = 1, 2, \dots, M$

Кожен об'єкт  $o^{(i)}$  має сукупність характеристик - ознак  $x_i^{(j)}$ ,  $i = 1, 2, \dots, N$  з множини  $\mathbb{X}$ .

Маємо множину  $\mathbb{C}$  класів  $c^{(k)}$ ,  $k = 2, \dots, K$

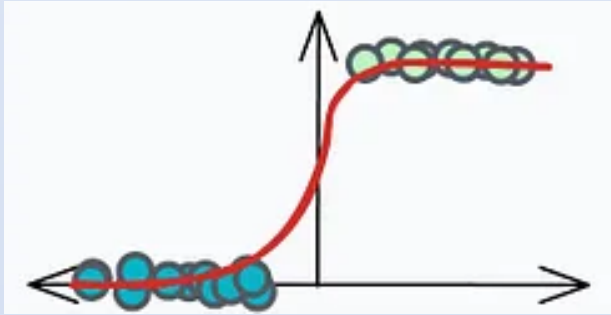
Існує невідома залежність (правило)  $\mathbb{F}$ , яка на підставі пар  $\langle o^{(j)}, c^{(k)} \rangle$  визначає, чи належить об'єкт  $o^{(j)}$  до класу  $c^{(k)}$ .

**Завдання:** знайти правило  $\check{\mathbb{F}}$ , максимально наближене до  $\mathbb{F}$ . Тобто, знайти вирішальне правило, що дозволяє класифікувати довільний об'єкт  $o$  за його ознаками.

# Методи Класифікації

- Регресійні методи, логістична регресія
- Метод k-найближчих сусідів (KNN)
- Метод опорних векторів (SVM)
- Наївний Байєс (ймовірнісний класифікатор)
- Дерева рішень
- ....

# Загальні методи класифікації



## Логістична регресія Logistic Regression

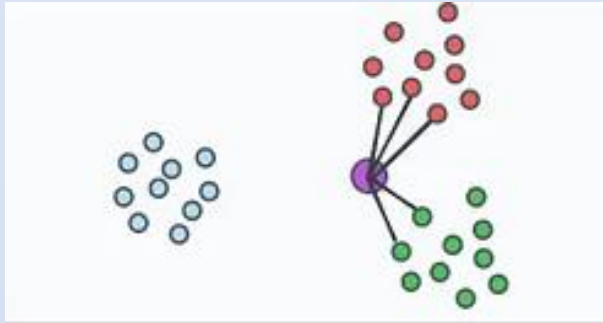
### Плюси:

- Проста інтерпретація моделі.
- Ефективна для задач із двома класами.
- Низька обчислювальна складність.

### Мінуси:

- Не підходить для багатокласових завдань.
- Чутлива до мультиколінеарності.
- Не може обробляти нелінійні залежності.

# Загальні методи класифікації



Метод k-найближчих сусідів  
**k-Nearest Neighbors (kNN)**

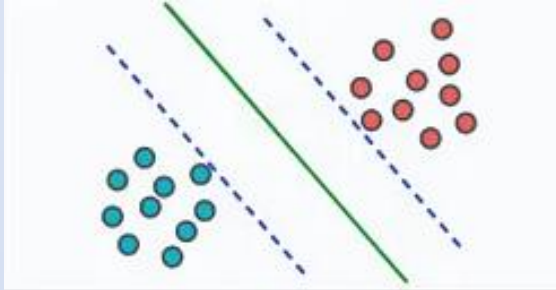
Плюси:

- Проста реалізація та інтерпретація.
- Добре працює з шумовими даними.
- Не вимагає навчання моделі.

Мінуси:

- Висока обчислювальна складність для великих наборів даних.
- Чутлива до вибору метрики відстані та значення k.
- Не може обробляти нелінійні залежності.

# Загальні методи класифікації



## Метод опорних векторів Support Vector Machines (SVM)

### Плюси:

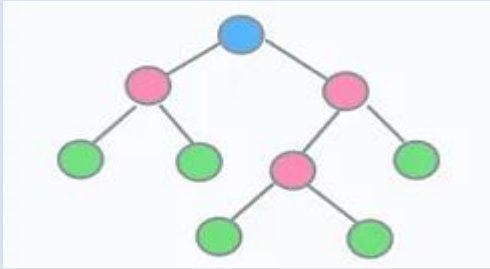
- Добре працює з високорозмірними даними.
- Ефективний для завдань з невеликою кількістю навчальних прикладів.
- Може обробляти нелінійні залежності.

### Мінуси:

- Чутливий до викидів.
- Вибір параметрів моделі може бути складним.
- Не завжди дає оптимальне рішення.



# Загальні методи класифікації



Дерева рішень

**Decision Tree Classifier**

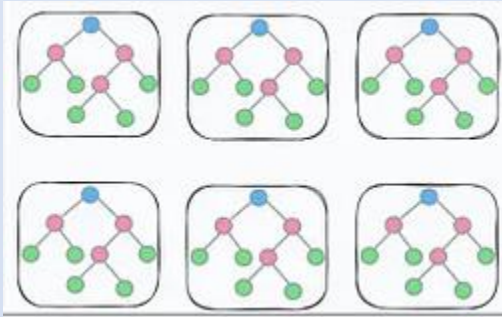
Плюси:

- Легко інтерпретувати правила.
- Добре працюють з категоріальними та числовими даними.
- Можуть обробляти нелінійні залежності.

Мінуси:

- Можуть бути перенавчені.
- Чутливі до шуму даних.
- Не завжди дають оптимальне рішення.

# Загальні методи класифікації



## Випадковий ліс Random Forest Classifier

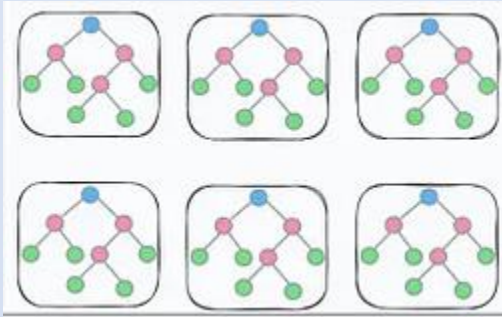
### Плюси:

- Стійкий до перенавчання.
- Добре працює з шумовими даними.
- Може обробляти завдання з безліччю класів.

### Мінуси:

- Інтерпретація моделі може бути складною.
- Висока обчислювальна складність для великих наборів даних.

# Загальні методи класифікації



Наївний баєсів класифікатор  
**Naive Bayes Classifier**

Ймовірнісний класифікатор

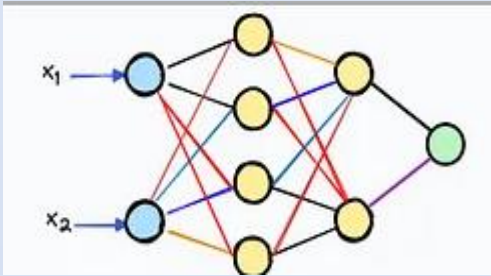
Плюси:

- Легко реалізувати та інтерпретувати.
- Може швидко навчатися та працювати з великими наборами даних.
- Стійкий до шуму даних.
- Може добре працювати з невеликими наборами даних.

Мінуси:

- Неможливість обробки нелінійних залежностей.
- Обмежена багатокласовість.

# Загальні методи класифікації



## Нейронні мережі Neural Networks

### Плюси:

- Можуть обробляти завдання з безліччю класів.
- Добре працюють із нелінійними залежностями.
- Можуть навчатись на великих обсягах даних.

### Мінуси:

- Інтерпретація моделі може бути складною.
- Висока обчислювальна складність.
- Потрібен великий обсяг навчальних даних.

# Бінарна класифікації

Маємо множину  $\mathbb{O}$  об'єктів  $o^{(j)}$ ,  $j = 1, 2, \dots, M$ .

Маємо два класи  $c^{(0)}$ ,  $c^{(1)}$ ,  $K = 2$ .

Припустимо, що кожен об'єкт  $o^{(j)}$  має тільки одну ознаку  $x^{(j)} \in \mathbb{R}$ , множини  $\mathbb{X}$ .

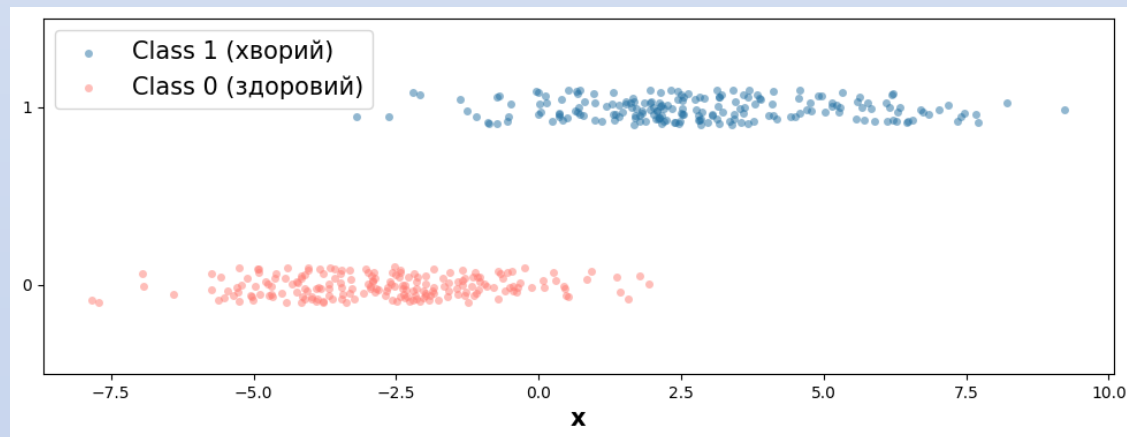
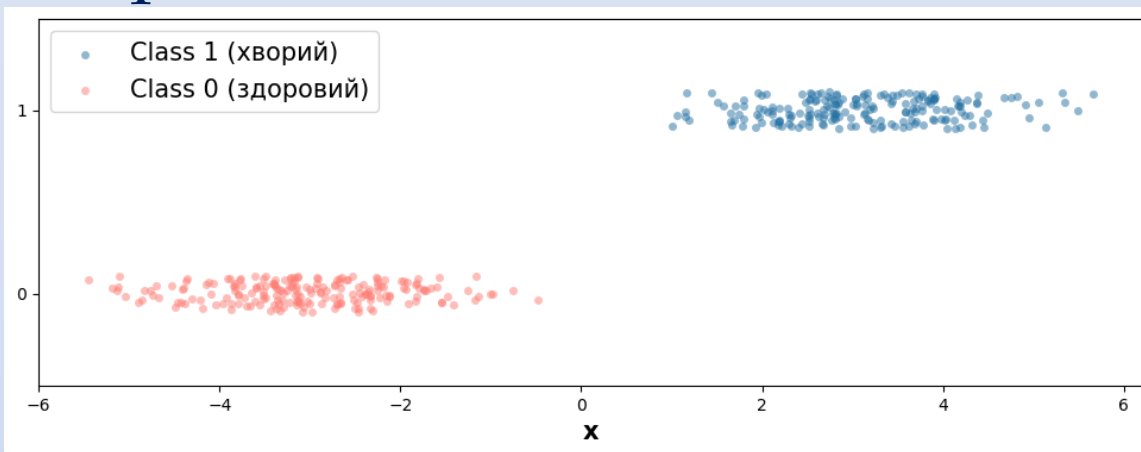
**Приклад. Діагностика.**

Є  $M$  пацієнтів  $o^{(j)}$ ,  $j = 1, 2, \dots, M$ . Стан пацієнта характеризується кількістю (типізацією) специфічних Т-клітин (дійсно число). Пацієнти або здорові  $c^{(0)}$ , або хворі  $c^{(1)}$ . Необхідно побудувати модель, що по значенню  $x^{(v)}$  класифікує пацієнта  $o^{(v)}$  як хворого, або здорового

Тобто знайти  $F(\mathbf{x})$ , натреновану на множені  $\mathbb{O}$ , що повертає 0 або 1 з дякою (мінімальною) похибкою.

# Бінарна класифікації

Проблема.

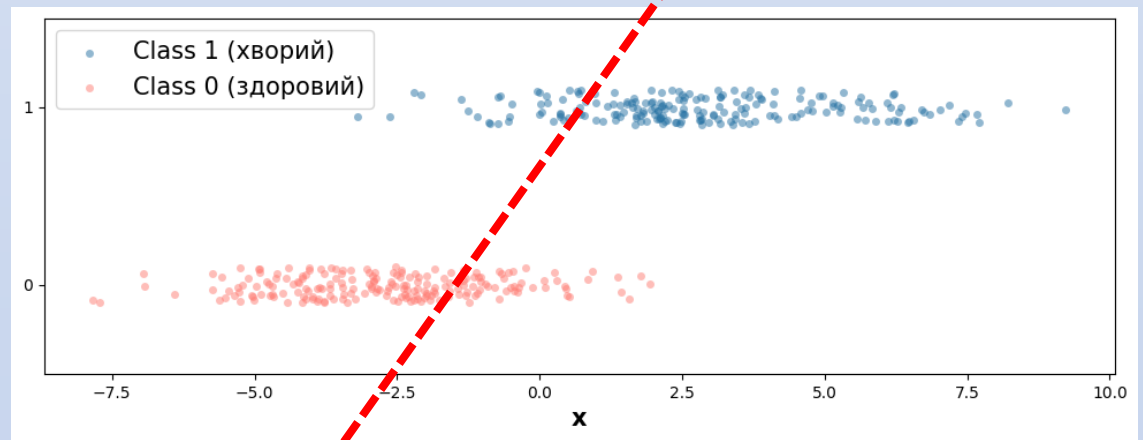
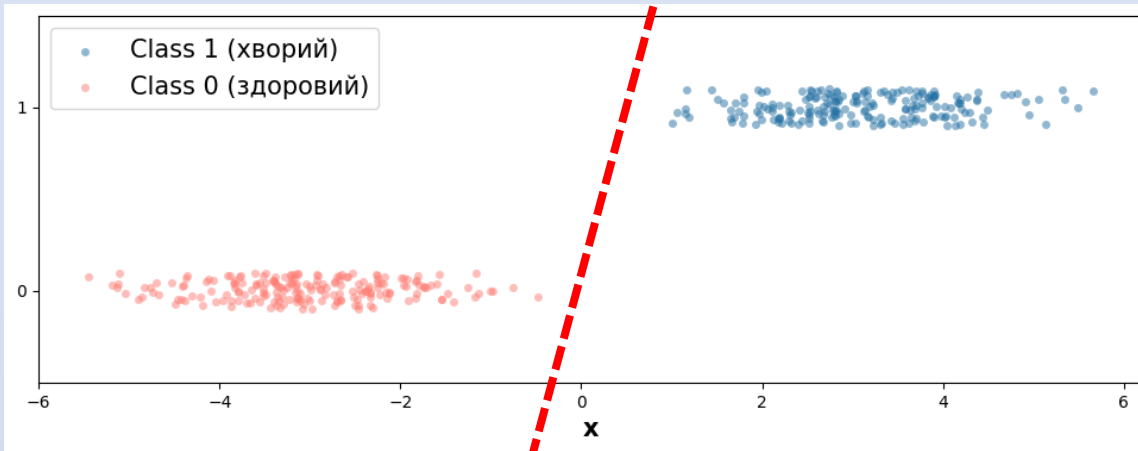


Як розділити класи?

# Бінарна класифікації

Проблема.

Лінійна регресія (погано)

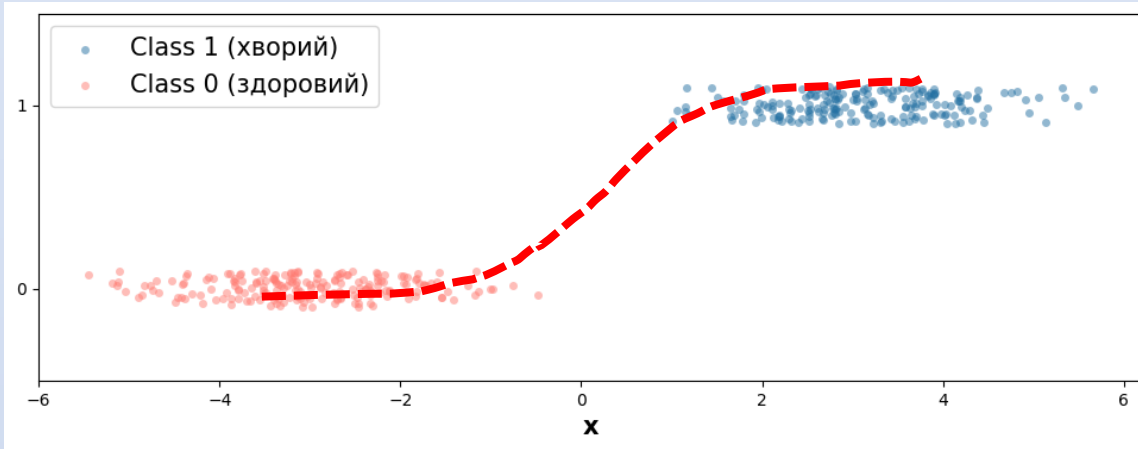


$$y = f(x) = w_1 * x + w_0$$

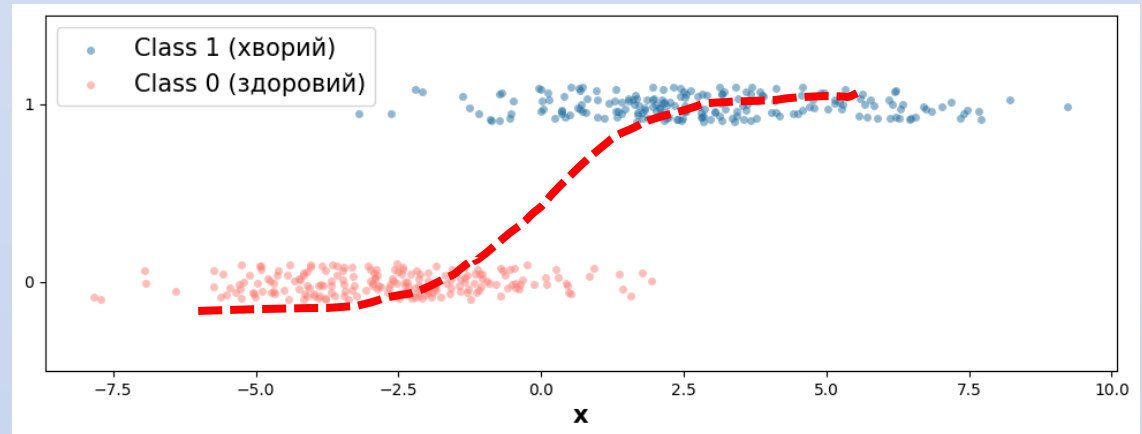
# Бінарна класифікації

Проблема.

Логістична регресія



$$y = f(z) = f(w_1 * x + w_0)$$



Змінна  $y$  - ймовірність того, конкретний пацієнт хворий або здоровий.



# Бінарна класифікації

Логістична регресія спочатку обчислює шанси події, що відбувається для різних рівнів кожної незалежної змінної, а потім бере логарифм, щоб створити безперервний критерій як перетворену версію залежної змінної.

Означмо  $p$  – ймовірність успіху (клас 1), то математично шанси з погляду ймовірності дорівнюють

$$\frac{p}{1-p}.$$

Логарифм шансів - це логіт ймовірності:

$$\log(p) = \log \frac{p}{1-p} = w_1 * x + w_0$$

$$\frac{p}{1-p} = e^{w_1 * x + w_0}$$

$$p = \frac{e^{(w_1 * x + w_0)}}{1 + e^{(w_1 * x + w_0)}}$$

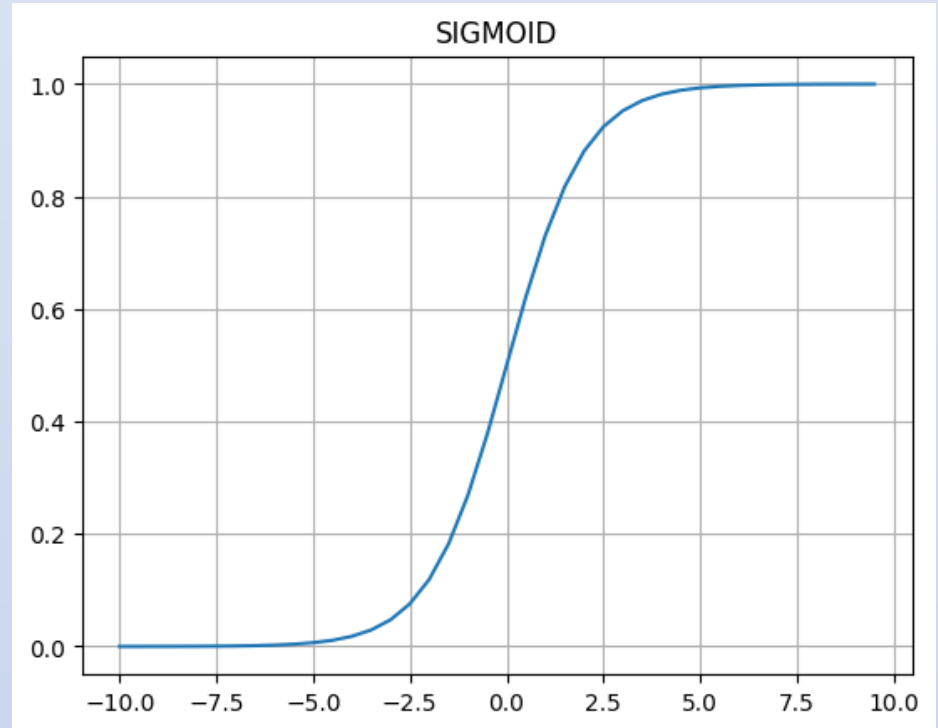
# Бінарна класифікації

Логістична функція, логістична крива,  
Логіт-перетворення

$$F(z) = \frac{1}{1+e^{-z}} = \frac{e^z}{1+e^z}$$
$$\frac{1}{1 + e^{-(w_1 * x + w_0)}}$$

$$F(z) \in (0, 1) \forall z$$

За допомогою логістичної  
регресії можна оцінити  
ймовірність того, що подія  
настане.



# Бінарна класифікації

Пошук параметрів моделі виконується на основі методу **максимальної правдоподібності**.

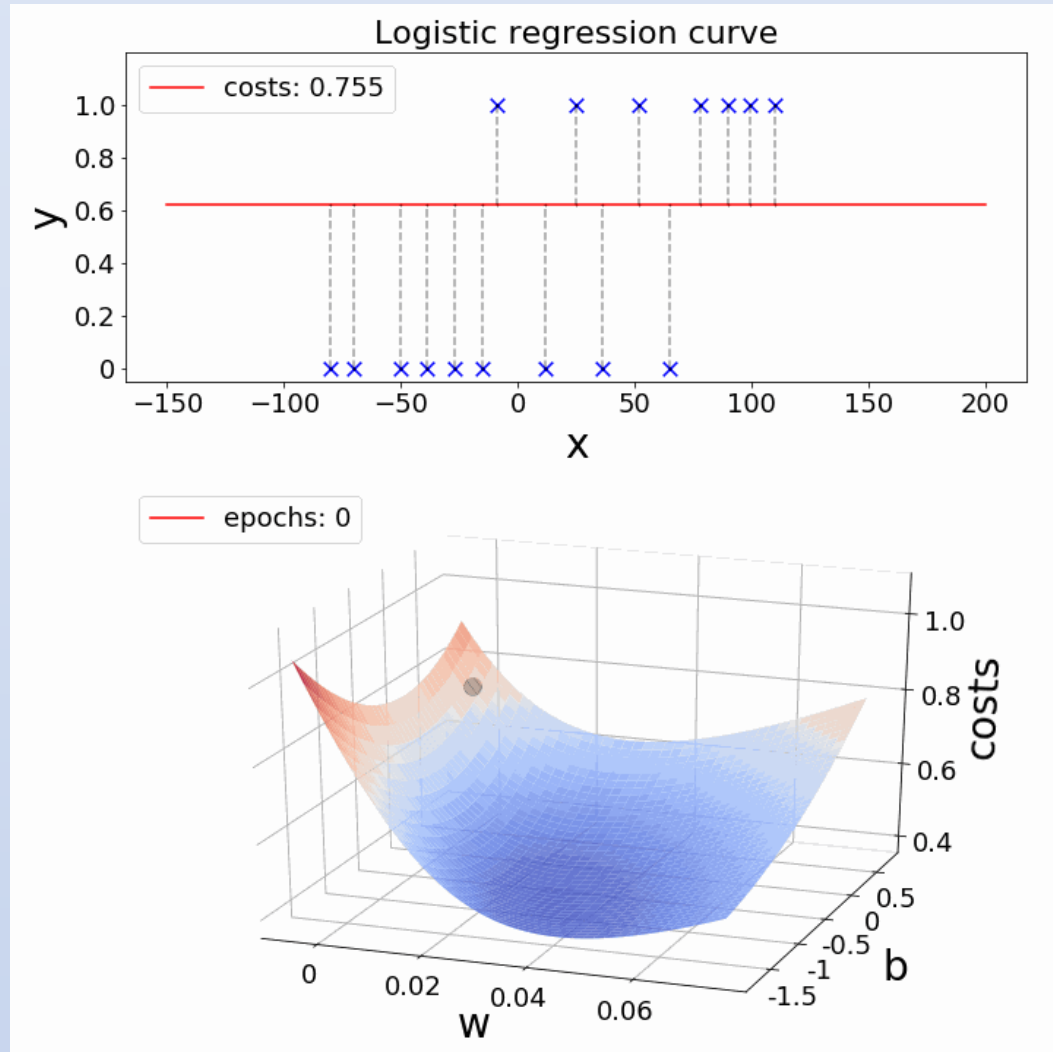
Обчислюється функція правдоподібності (*likelihood function, L*), що виражає щільність ймовірності спільної появи результатів вибірки.

Позначимо через  $p_i$  ймовірність появи одиниці для  $o^{(i)}$  :  $p_i = Prob(y_i = 1)$ . Ця ймовірність залежить від  $x_i, w_0, w_1$ . Для  $i \in c^{(0)}, y_i = 0$ ,  
Для  $i \in c^{(0)}, y_i = 0; i \in c^{(1)}, y_i = 1$ ; Тоді  $\log L$ :

$$\begin{aligned} L^*(W) &= \frac{1}{M} \left[ \sum_{i \in C0} \log p_i(x_i, W) + \sum_{i \in C1} \log(1 - p_i(x_i, W)) \right] \\ &= \frac{1}{M} \sum_{i=1}^M [y_i \log p_i(x_i, W) + (1 - y_i) \log(1 - p_i(x_i, W))] \end{aligned}$$

# Бінарна класифікації

## Візуалізація



# Бінарна класифікації

На відміну від лінійної регресії з нормально розподіленими залишками, неможливо знайти вираз у замкнутій формі для значень коефіцієнтів, що максимізують функцію правдоподібності (зведення до СЛАУ).

Замість цього слід використовувати ітераційний процеси, наприклад, градієнтні методи.

У деяких випадках модель може не досягти збіжності.

# Якість бінарної класифікації

1. **Матриця плутанини:** таблиця, яка використовується для опису продуктивності моделі класифікації.
2. **Крива ROC:** Графік, який ілюструє здатність моделі правильно прогнозувати позитивний клас на різних порогових рівнях, надаючи уявлення про баланс між чутливістю та специфічністю.
3. **AUC:** означає «Площа під кривою ROC», цей показник кількісно визначає загальну здатність моделі розрізняти класи, причому вищі значення вказують на кращу продуктивність.

# Якість бінарної класифікації

Матриця плутанини  
(confusion matrix)

|              |          | PREDICTED VALUE      |                      |
|--------------|----------|----------------------|----------------------|
|              |          | Positive             | Negative             |
| ACTUAL VALUE | Positive | TP<br>TRUE POSITIVE  | FN<br>FALSE NEGATIVE |
|              | Negative | FP<br>FALSE POSITIVE | TN<br>TRUE NEGATIVE  |

**TP** – кількість **вірно** класифікованих позитивно

**FP** – кількість **невірно** класифікованих позитивно  
(похибка 1-го роду)

**FN** – кількість **невірно** класифікованих негативно  
(похибка 2-го роду)

**TN** – кількість **вірно** класифікованих негативно

# Якість бінарної класифікації

Відносні значення

|  |            |                            |
|--|------------|----------------------------|
| Відносно<br>кількості<br>реально<br>позитивних | чутливість | $TPR = \frac{TP}{TP + FN}$ |
|  | невлучання | $FNR = \frac{FN}{TP + FN}$ |

|  |               |                            |
|--|---------------|----------------------------|
| Відносно<br>кількості<br>реально<br>негативних | специфічність | $TNR = \frac{TN}{TN + FP}$ |
|  |               | $FPR = \frac{FP}{TN + FP}$ |



# Якість бінарної класифікації

Точність:

$$\text{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$$

Значущість, влучність:

$$\text{Precision} = \frac{TP}{TP+FP}$$

Повнота, чутливість:

$$\text{Recall} = \frac{TP}{TP+FN}$$

Діагностичне відношення шансів

$$F_1 = \frac{2TP}{2TP+FP+FN}$$

# Якість бінарної класифікації

Приклад. Маємо тестовий датасет 100 пацієнтів

| Ідеальна CM |             | PREDICTED VALUE |               |
|-------------|-------------|-----------------|---------------|
|             |             | Positive = 70   | Negative = 30 |
| A           | Positive 70 | TP = 70         | FN = 0        |
| V           | Negative 30 | FP = 0          | TN = 30       |

| Реальна CM |             | PREDICTED VALUE |               |
|------------|-------------|-----------------|---------------|
|            |             | Positive = 60   | Negative = 40 |
| A          | Positive 70 | TP = 50         | FN = 20       |
| V          | Negative 30 | FP = 10         | TN = 20       |

***100% → Accuracy → 70%***

***100% → Precision = 83%***

***100% → Recall = 71%***

***100% → F1 = 76%***

# Контрольні запитання

- **Надайте загальну постановку задачі бінарної класифікації.**
- **Пояснить сутність логістичної регресії.**
- **Пояснить процес пошуку параметрів моделі логістичної регресії.**
- **Опишіть матрицю плутанини та її використання для оцінки якості вирішення задачі бінарної класифікації.**

## Рекомендована ЛІТЕРАТУРА

- **Глибинне навчання:** Навчальний посібник / Уклад.: В.В. Литвин, Р.М. Пелещак, В.А. Висоцька В.А. – Львів: Видавництво Львівської політехніки, 2021. – 264 с.
- Тимощук П. В., Лобур М. В. **Principles of Artificial Neural Networks and Their Applications: Принципи штучних нейронних мереж та їх застосування:** Навчальний посібник. – Львів : Видавництво Львівської політехніки, 2020. – 292 с.
- Morales M. **Grokking Deep Reinforcement Learning.** – Manning, 2020. – 907 с.
- Trask Andrew W. **Grokking Deep Learning.** – Manning, 2019. – 336 с.

# Корисні та цікави посилання

- **Машинне навчання**

[https://uk.wikipedia.org/wiki/машинне\\_навчання](https://uk.wikipedia.org/wiki/машинне_навчання)

- **Львівська політехніка**

<http://www.mmf.lnu.edu.ua/ar/1739>

<http://www.mmf.lnu.edu.ua/ar/1743>

**The END**

**Модуль 2. Лекція 06.**