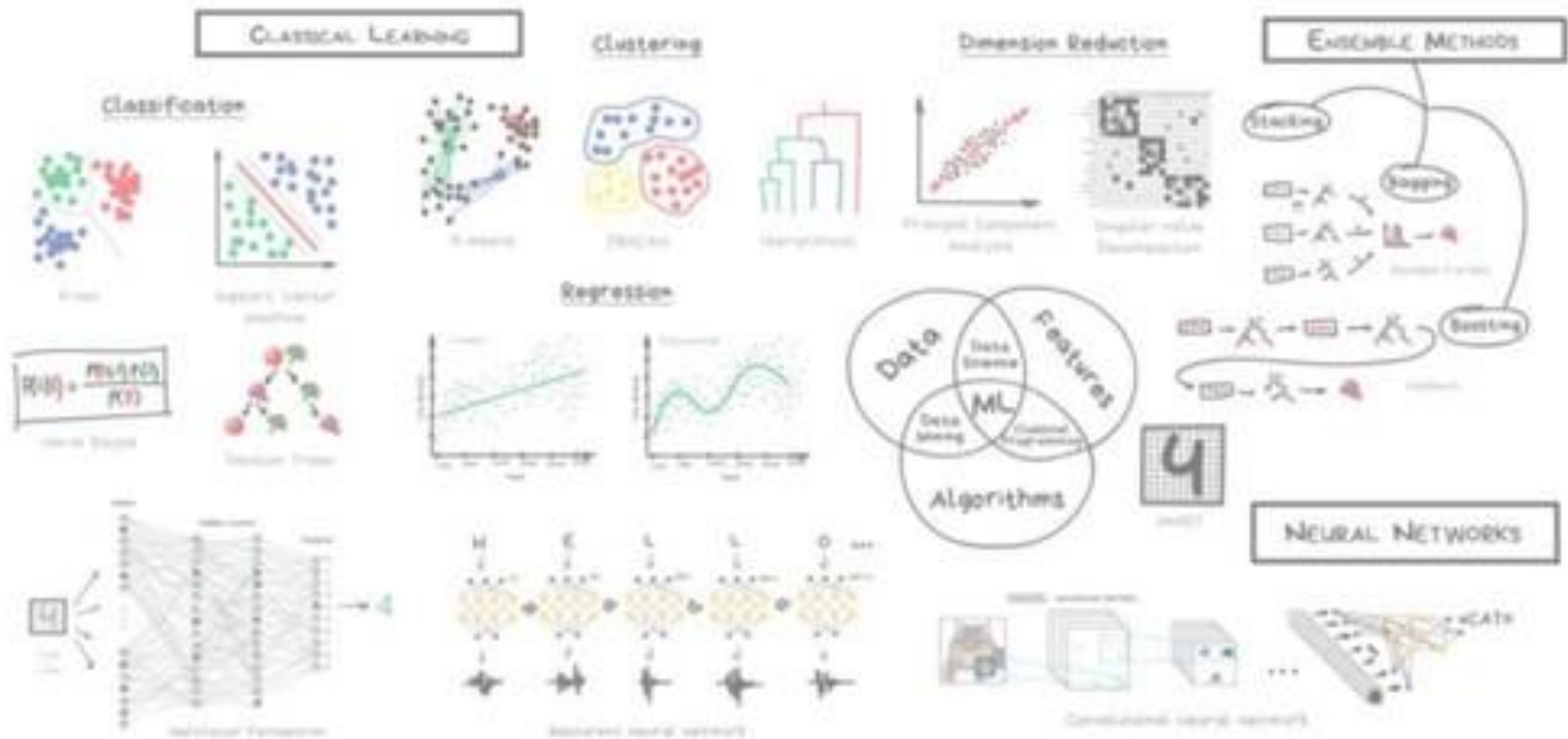


# Машинне навчання

## Класичне навчання. Навчання з вчителем



# Основні типи класичного навчання

## Классическое Обучение



# Класичне навчання

Перші алгоритми прийшли до нас із чистої статистики ще у 1950-х. Сьогодні на **класичних алгоритмах** тримається добра половина інтернету.

## Приклад

- Коли ви зустрічаєте блок «Рекомендовані статті» на сайті
- Банк блокує усі ваші гроші на картці після першої ж купівлі кави за кордоном

Це майже завжди справа рук одного із **класичних алгоритмів**.

Великі корпорації люблять вирішувати всі проблеми за допомогою **нейромереж**. Тому що зайві **2% точності** для них легко конвертуються в додаткові **2 мільярди прибутку**.

Коли завдання може бути вирішене **класичними методами**, дешевше реалізувати скільки-небудь корисну для бізнесу систему на них, а потім думати про покращення.

Якщо ви не вирішили завдання **класичними методами**, то не вирішити її на 2% краще вам не надто допоможе.

# Навчання з учителем

Класичне навчання поділяють на дві категорії:

- з учителем (Supervised Learning)
- без учителя (Unsupervised Learning)

У навчанні **з учителем** машина має якийсь вчитель, який говорить їй як правильно. Розповідає, що на цьому малюнку кішка, а на цьому собака. Тобто вчитель вже заздалегідь розділив усі дані на кішок та собак, а машина навчається на конкретних прикладах – **дані розмічені**

У навчанні **без вчителя**, машині просто вивалюють купу фотографій тварин на стіл і кажуть «розберись, хто тут на кого схожий» — **дані не розмічені**, машина не має вчителя, і вона намагається сама знайти будь-які закономірності.

Очевидно, що **з учителем** машина навчиться швидше та точніше, тому в **бойових** завданнях його використовують набагато частіше.

Ці завдання поділяються на два типи:

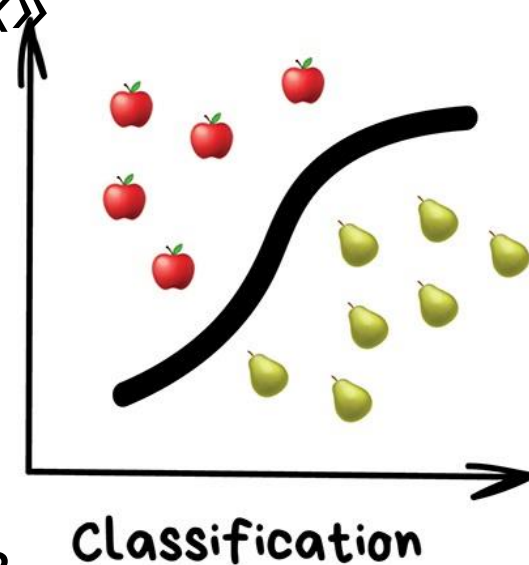
- **класифікація** - передбачення категорії об'єкта
- **регресія** - передбачення місця на числовій прямій

# Класифікація

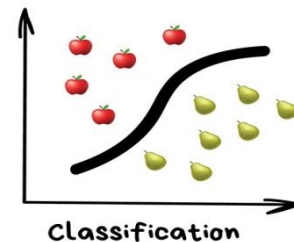
*«Розділяє об'єкти за заздалегідь відомій ознаці: шкарпетки по кольорах, документи по мовах, музику по жанрах»*

**Використовують для:**

- Спам-фільтри
- Визначення мови
- Аналіз тональності
- Пошук схожих документів
- Розпізнавання прописних букв та цифр
- Визначення підозрілих транзакцій



# Класифікація



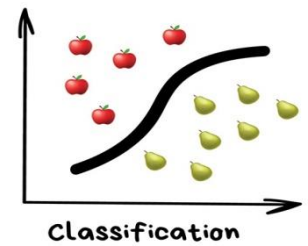
**Класифікація речей** - найпопулярніше завдання у всьому машинному навчанні. Машина в ній як дитина, яка вчиться розкладати іграшки: роботів в одну скриньку, танки в іншу. А якщо це робот-танк? Тоді час розплакатися та випасти у помилку.

Для класифікації завжди потрібен **вчитель** - **розмічені дані** з ознаками та категоріями, які машина вчитиметься визначати за цими ознаками.

Класифікувати можна будь-що:

- користувачів за інтересами - так роблять алгоритмічні стрічки,
- статті за мовою та тематикою — важливо для пошукових систем,
- музику за жанрами - плейлисти (наприклад, Apple music)
- листи у вашій поштовій скриньці.

# Методи (алгоритми) класифікації

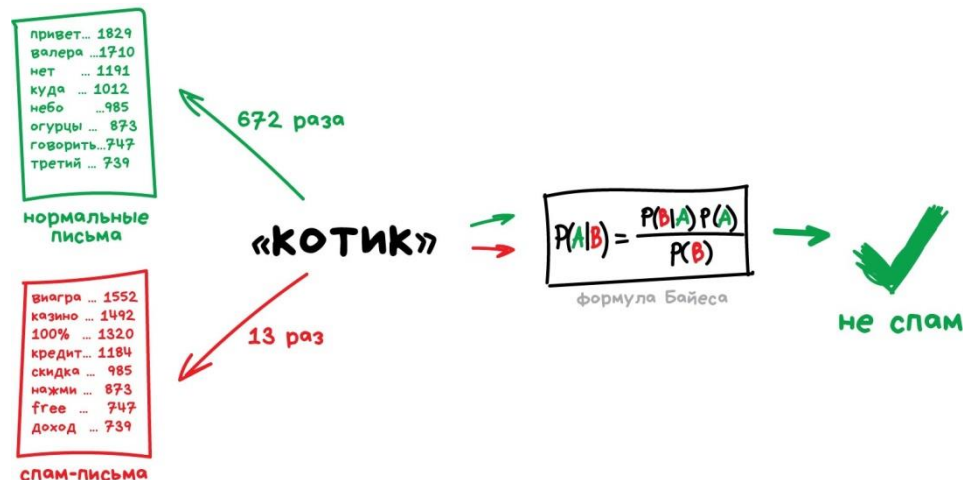


- Простий б́айєсівський класифікатор (*Naive Bayes classifiers*)
- Древа рішень (*Decisions Trees*)
- К найближчих сусідів (*K-NN*)
- Метод опорних векторів (*SVM, Support Vector Machine*)
- Логістичная регресія (*Logit Model*)
- .....

# Методи класифікації

## Простий байєсівський класифікатор

заснований на  
застосуванні  
теорему Байєса  
зі строгими  
припущеннями  
про незалежність.



Залежно від точної природи імовірнісної моделі, прості класифікатори Байєса можуть навчатися дуже ефективно. У багатьох практичних додатках для оцінки параметрів для простих байєсових моделей використовують метод максимальної правдоподібності



# Модель простого байєсівського класифікатора

Імовірнісна модель для класифікатора — умовна модель

$$p(C | f_1, \dots, f_n)$$

над залежною змінною класу  $C$  з малою кількістю результатів або (класів), залежить від декілької змінних (ознак)  $f_1, \dots, f_n$ .

Проблема полягає в тому, що коли кількість ознак дуже велика або коли одна ознака може приймати велику кількість значень, тоді будувати таку модель на імовірнісних таблицях стає неможливо.

Використовується теорема Байєса

$$p(C | f_1, \dots, f_n) = \frac{p(C)p(f_1, \dots, f_n | C)}{p(f_1, \dots, f_n)}$$

На практиці цікавий лише чисельник цього дробу, оскільки знаменник не залежить від залежної змінної  $C$  та значення ознак  $f_i$  дани, так що знаменник — константа.

# Модель простого байєсівського класифікатора

Чисельник є еквівалентним до сумісної ймовірності моделі

$$p(C | f_1, \dots, f_n)$$

яка може бути переписана наступним чином, з використанням повторного застосування визначення умовної ймовірності:

$$\begin{aligned} p(C | f_1, \dots, f_n) &\propto p(C) p(f_1, \dots, f_n | C) = \\ &= p(C) p(f_1 | C) p(f_2, \dots, f_n | C, f_1) = \\ &= p(C) p(f_1 | C) p(f_2 | C, f_1) p(f_3, \dots, f_n | C, f_1 f_2) = \\ &= p(C) p(f_1 | C) p(f_2 | C, f_1) \cdots p(f_n | C, f_1 f_2, f_3, \dots, f_{n-1}) \end{aligned}$$

Припущення умовної ймовірності: припустимо, що кожна ознака  $f_i$  умовно не залежить від будь-якої іншої ознаки  $f_j$  при  $j \neq i$ :

$$p(f_i | C, f_j) = p(f_i | C)$$

Таким чином, сумісна модель може бути виражена як:

$$p(C | f_1, \dots, f_n) \propto p(C) p(f_1 | C) p(f_2 | C) p(f_3 | C) \cdots p(f_n | C)$$

Із припущення про незалежність маємо:

$$p(C | f_1, \dots, f_n) = \frac{1}{p(f_1, \dots, f_n)} p(C) \prod_{i=1}^n p(f_i | C)$$

$Z = [p(f_1, \dots, f_n)]^{-1}$  — масштабний множник, що залежить лише від вектора ознак  $f_1, \dots, f_n$ , тобто є константа, якщо значення ознак відомі.

# Приклад простого байєсівського класифікатора

## Дано:

- Навчальна вибірка з однією ознакою «погодні умови»
- Цільова функція «Відбудеться гра чи ні»

## Визначити:

- Чи відбудеться гра («yes») за умови сонячної погоди («sunny»)

**Розв'язок:** Перетворимо набір даних в таблицю правдоподібності (likelihood table), шляхом розрахунку відповідних імовірностей.

Likelihood table				
Weather	No	Yes		
Overcast	0	4	=4/14	0.29
Rainy	3	2	=5/14	0.36
Sunny	2	3	=5/14	0.36
All	5	9		
	=5/14	=9/14		
	0.36	0.64		

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

# Приклад простого байєсівського класифікатора

Likelihood table				
Weather	No	Yes		
Overcast	0	4	=4/14	0.29
Rainy	3	2	=5/14	0.36
Sunny	2	3	=5/14	0.36
All	5	9		
	=5/14	=9/14		
	0.36	0.64		

- Априорна ймовірність даного значення ознаки

$$p(f_1 = \text{sunny}) = 0.36$$

- Априорна ймовірність даного класу

$$p(C = \text{Yes}) = 0.64$$

- Ймовірність даного значення ознаки при даному класі

$$p(f_1 = \text{sunny} \mid C = \text{yes}) = \frac{3}{9}$$

- Апостеріорна ймовірність даного класу  $C$  (тобто даного значення цільової змінної) при даному значенні ознаки  $f_1$

$$p(C \mid f_1) = \frac{p(C) p(f_1 \mid C)}{p(f_1)} = 0.6$$

# Методи класифікації

## Дерева прийняття рішень

Машина автоматично поділяє всі дані за питаннями, відповіді на які можуть бути «так» чи «ні». Питання можуть бути не зовсім адекватними з погляду людини, але машина вигадує їх так, щоб на кожному кроці розбиття було найточнішим.

Так виходить дерево питань.  
Чим вище рівень,  
тим більш загальне питання.



Дерева нашли свою нішу в областях із високою відповідальністю: діагностиці, медицині, фінансах.

# Модель дерева прийняття рішень

## Дано:

- Навчальна вибірка

## Задача

- Класифікувати новий випадок

Дерево прийняття рішень це дерево. На ньому є мітки:

- У вузлах, що не є листям: атрибути, за якими розрізняються випадки
- У листках: значення цільової функції
- На ребрах: значення атрибута, з якого виходе ребро

## Розв'язок:

- Спуститися по дереву від кореня до відповідного листка

# Приклад використання дерева прийняття рішень

Задача: Чи відбудеться гра в теніс

Навчальна  
вибірка:

9 yes / 5 no

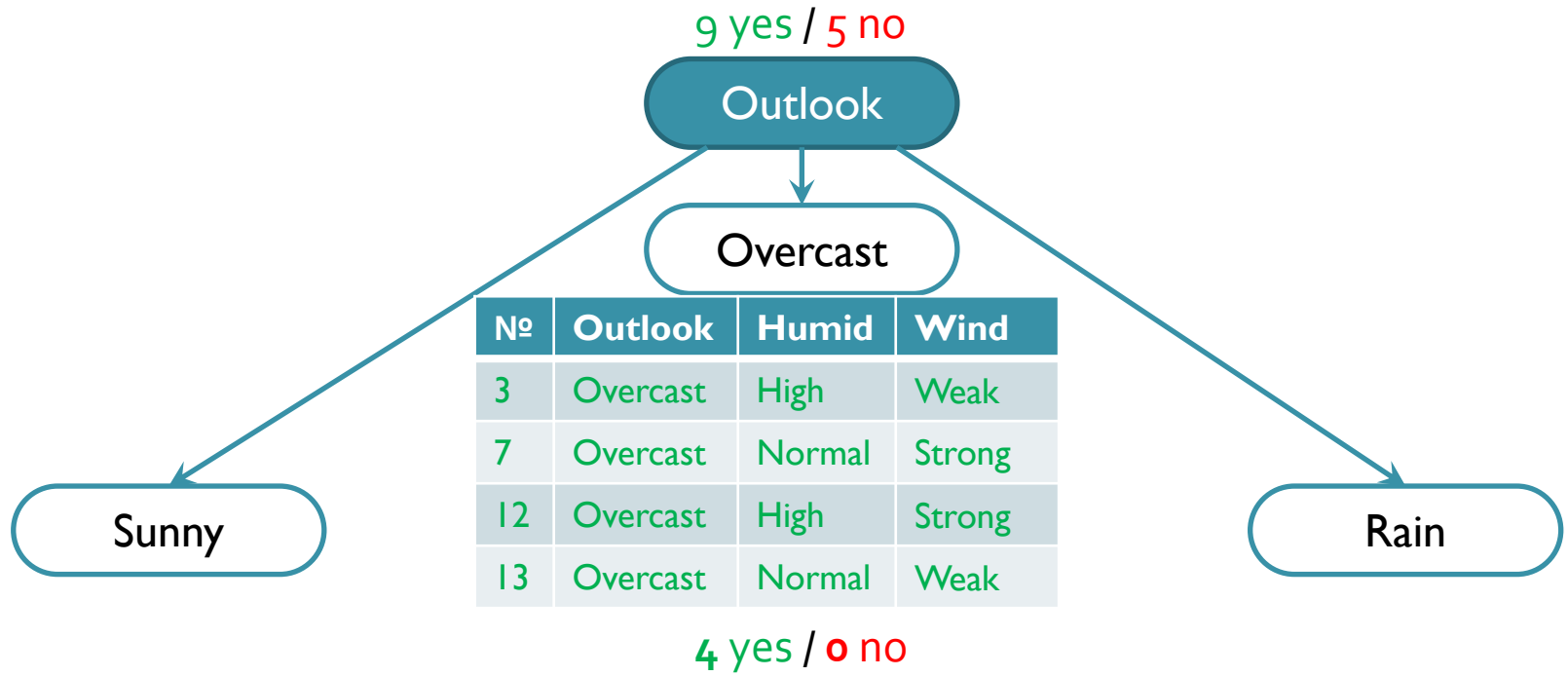
## АЛГОРИТМ

- Розмічаємо дані за значенням цільової функції (Yes/No) або (1/0)
- Поділяємо дані за однією з ознак
- Робимо перевірку на «чистоту»:
  - (Всі Yes або все No)
  - Так – зупиняємось
  - Ні – продовжуємо
- Визначаємо, якому підмножині належать нові дані

№	Outlook	Humidity	Wind	Play
1	Sunny	High	Weak	No
2	Sunny	High	Strong	No
3	Overcast	High	Weak	Yes
4	Rain	High	Weak	Yes
5	Rain	Normal	Weak	Yes
6	Rain	Normal	Strong	No
7	Overcast	Normal	Strong	Yes
8	Sunny	High	Weak	No
9	Sunny	Normal	Weak	Yes
10	Rain	Normal	Weak	Yes
11	Sunny	Normal	Strong	Yes
12	Overcast	High	Strong	Yes
13	Overcast	Normal	Weak	Yes
14	Rain	High	Strong	No
15	Rain	High	Weak	???

# Приклад використання дерева прийняття рішень

Задача: Чи відбудеться гра в теніс



№	Outlook	Humidity	Wind
1	Sunny	High	Weak
2	Sunny	High	Strong
8	Sunny	High	Weak
9	Sunny	Normal	Weak
11	Sunny	Normal	Strong

2 yes / 3 no

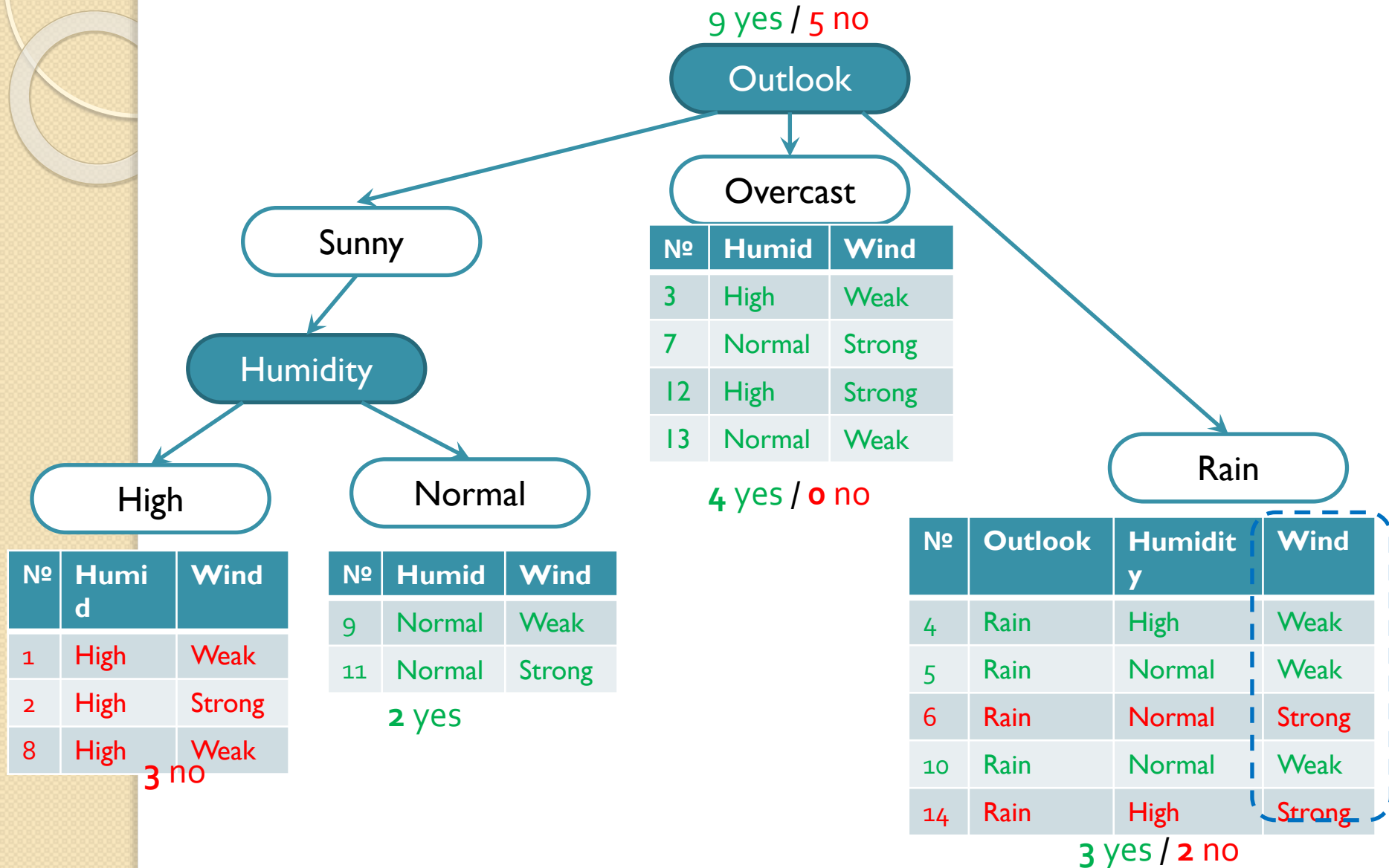
№	Outlook	Humidity	Wind
4	Rain	High	Weak
5	Rain	Normal	Weak
6	Rain	Normal	Strong
10	Rain	Normal	Weak
14	Rain	High	Strong

3 yes / 2 no



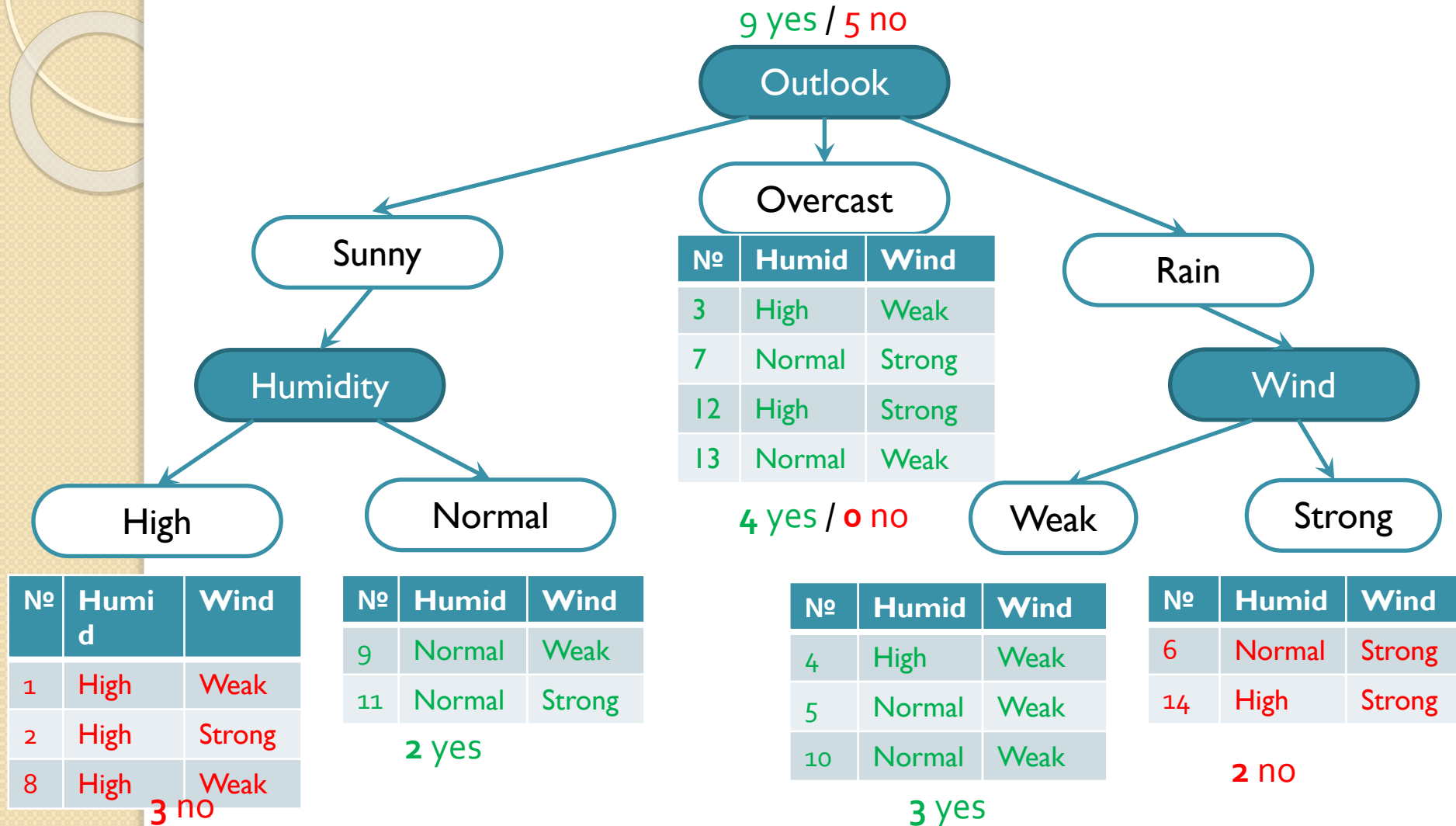
# Приклад використання дерева прийняття рішень

Задача: Чи відбудеться гра в теніс



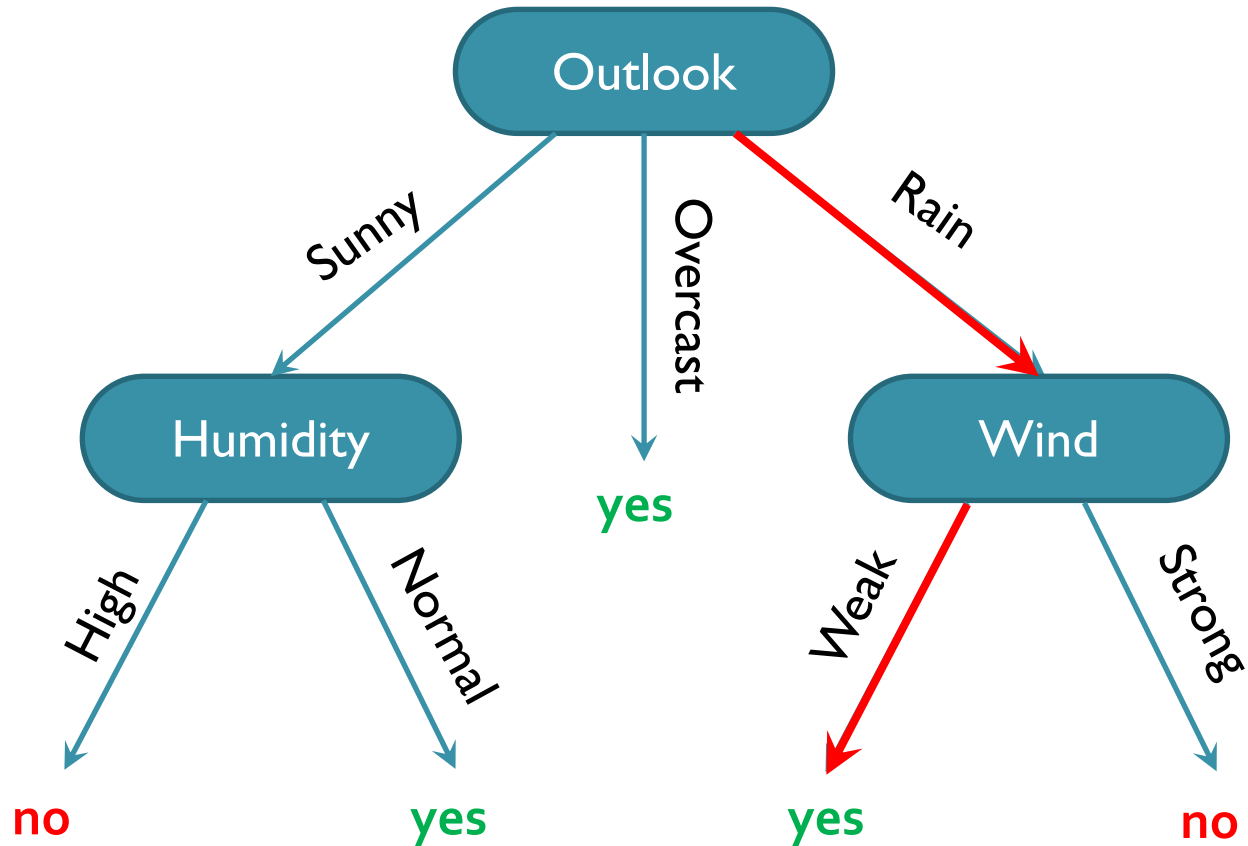
# Приклад використання дерева прийняття рішень

Задача: Чи відбудеться гра в теніс



# Приклад використання дерева прийняття рішень

Задача: Чи відбудеться гра в теніс



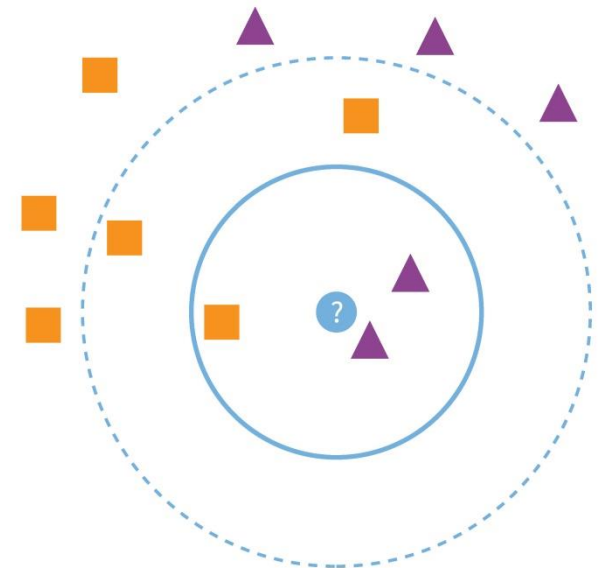
№	Outlook	Humidity	Wind	Play
15	Rain	High	Weak	Yes

# Методи класифікації

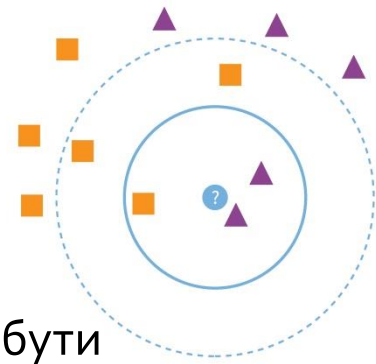
## Метод К найближчих сусідів (KNN)

Метод належить до класу непараметричних, тобто не вимагає припущень про те, з якого статистичного розподілу було сформовано **навчальну вибірку**. Отже, класифікаційні моделі, побудовані за допомогою методу KNN, також будуть непараметричними. Це означає, що структура моделі не визначається жорстко спочатку, а визначається даними.

Для кожного об'єкта визначаються  $k$  найближчих (у сенсі деякої метрики) попередньо класифікованих спостережень. Потім вибирається клас, якому належить більшість з  $k$  найближчих прикладів-сусідів, і до цього ж класу належить об'єкт, що класифікується.



# Модель класифікації методом $k$ найближчих сусідів (KNN)

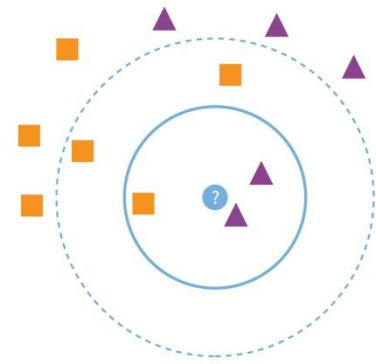


- У найпростішому випадку клас нового об'єкта може бути визначений простим вибором класу, що найчастіше зустрічається серед  $k$  прикладів. При цьому розраховується відстань від даного об'єкта до кожного об'єкта з навчальної вибірки за формулою

$$D_j = \sqrt{\sum_{i=1}^l (f_i - \hat{f}_i)^2}$$

- У звичайному випадку використовують **просте невважене голосування** (simple unweighted voting). При цьому передбачається, що всі приклади мають однакове право «голосу» незалежно від відстані до об'єкта, що класифікується.
- Однак, логічно припустити, що чим далі приклад розташований від об'єкта, що класифікується, в просторі ознак, тим нижче його значимість для визначення класу. Тому для покращення результатів класифікації вводять **зважування** прикладів залежно від їхньої віддаленості. У цьому випадку використовують **зважене голосування** (weighted voting).

# Модель класифікації методом к найближчих сусідів (KNN)

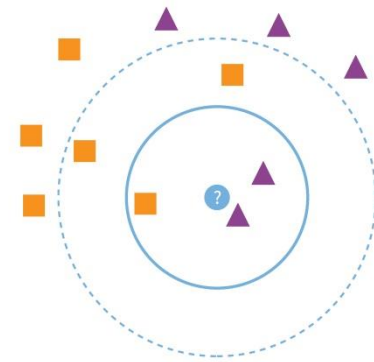


- В основі ідеї зваженого голосування лежить введення «штрафу» для класу, залежно від того, наскільки приклади, що відносяться до нього, віддалені від класифікованого об'єкта. Такий «штраф» представляється як сума величин, обернених квадрату відстаней від прикладу класу до класифікованого об'єкта (часто це значення називають **показником близькості**):

$$Q_j = \sum_{i=1}^{n_j} \frac{1}{D^2(f, a_{ij})}$$

- де  $D$  — оператор розрахунку відстані,  $f$  — вектор ознак об'єкта, що класифікується,  $a_{ij}$  —  $i$ -й приклад  $j$ -го класу. Таким чином, «перемагає» той клас, для якого величина  $Q_j$  буде найбільшою. При цьому також знижується імовірність того, що класи отримають однакову кількість голосів.

# Модель класифікації методом $k$ найближчих сусідів (KNN)



- Вибір параметра  $k$  є важливим для отримання коректних результатів класифікації.
- Якщо значення параметра мале, виникає ефект перенавчання, коли рішення щодо класифікації приймається з урахуванням малої кількості прикладів і має низьку значимість.
- Якщо встановити  $k=1$ , то алгоритм просто присвоюватиме будь-якому новому спостереженню мітку класу найближчого об'єкта.
- Використання невеликих значень  $k$  збільшує вплив шумів на результати класифікації, коли невеликі зміни даних приводять до великих змін у результатах класифікації. Але при цьому межі класів виявляються більш вираженими (клас при голосуванні перемагає з великим рахунком).
- Якщо значення параметра  $k$  занадто велике, у процесі класифікації бере участь багато об'єктів, які належать до різних класів. Така класифікація виявляється дуже грубою і погано відбиває локальні особливості набору даних. Тут зменшується зашумленість результатів класифікації, але знижується вираженість меж класів.
- У задачах бінарної класифікації буває доцільно вибрати  $k$  як непарне число, оскільки це дозволяє уникнути рівності «голосів» щодо класу нового спостереження.
- Таким чином, вибір параметра  $k$  є компромісом між точністю і узагальнюючою здатністю моделі.

# Приклад класифікації методом k найближчих сусідів (KNN)

Маємо набір даних щодо позичальників банку частина з яких допустили просрочення платежу.

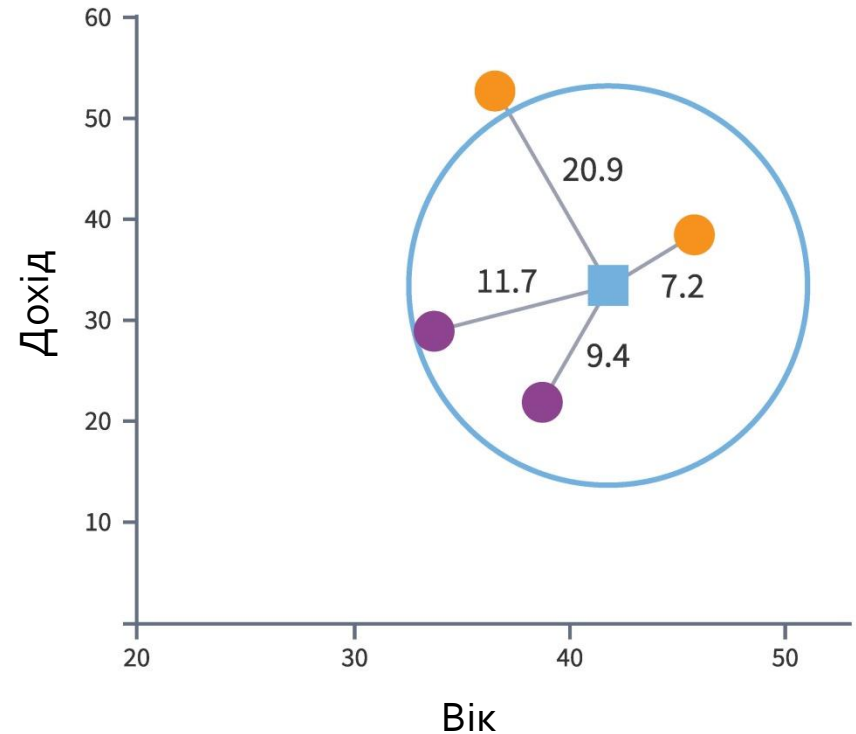
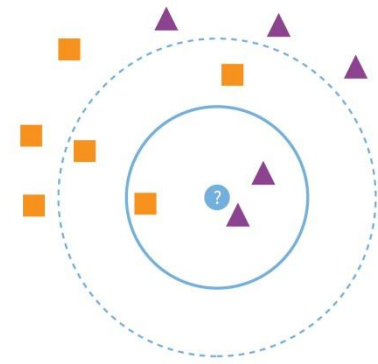
Ознаками є вік ( $f_1$ ) позичальника та його середньомісячний дохід ( $f_2$ ). Мітками класа в полі «Просрочено» будуть «Так» та «Ні».

$f_1$ Вік	$f_2$ Дохід	Y
46	41	No
36	54	No
34	29	Yes
38	23	Yes

Задача полягає в тому, щоб провести класифікацію нового позичальника для якого

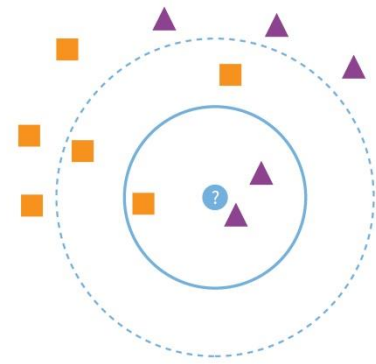
$$f_1 = 42 \text{ и } f_2 = 34$$

З метою оцінити можливість просрочення ним платежу.





# Приклад класифікації методом $k$ найближчих сусідів (KNN)



1. Задамо значення параметра  $k=3$
2. Розрахуємо відстань  $D$  між вектором ознак об'єкта, що класифікується, і векторами навчальних прикладів і встановимо для кожного прикладу його ранг
3. Виключимо з розгляду приклад, який при  $k=3$  не є сусідом і розглянемо класи, що залишилися.

$$\hat{f}_1 = 42 \text{ та } \hat{f}_2 = 34$$

$$D_j = \sqrt{\sum_{i=1}^l (f_i - \hat{f}_i)^2}$$

$f_1$ Вік	$f_2$ Дохід	Distance	Rang	Neighbor	Y
46	41	8.06	1	Yes	No
<del>36</del>	<del>54</del>	<del>20.9</del>	<del>4</del>	<del>No</del>	<del>No</del>
34	29	9.4	2	Yes	Yes
38	23	11.7	3	Yes	Yes

## Приклад класифікації методом k найближчих сусідів (KNN)

$f_1$ Вік	$f_2$ Доход	Distance	Rang	Y
46	41	8.06	1	No
34	29	9.4	2	Yes
38	23	11.7	3	Yes

Шляхом простого **незваженого голосування** визначаємо його клас «Так»

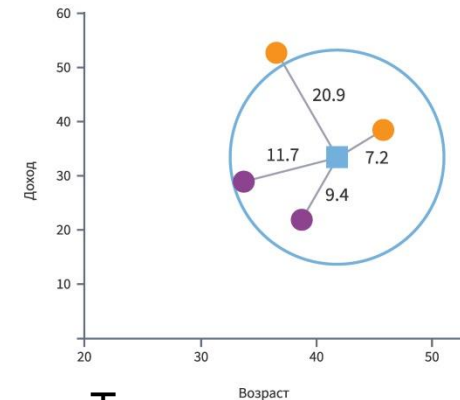
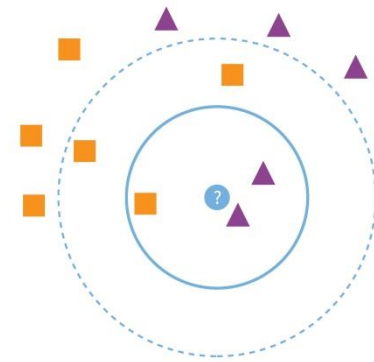
Показник близькості:

$$Q_j = \sum_{i=1}^{n_j} \frac{1}{D^2(f, a_{ij})}$$

$$Q_{No} = \frac{1}{65} \approx 0.015 \quad Q_{YES} = \frac{1}{98} + \frac{1}{137} \approx 0.018$$

Шляхом **зваженого голосування** визначаємо його клас «Так»

З роботи класифікатора робимо висновок, що позичальник із заданими характеристиками може допустити прострочення виплати кредиту.





Дякую за увагу