## 1.1 Метод $k$-найближчих сусідів (K-Nearest Neighbors)

Типовим представником методів класифікації є метод $k$-найближчих сусідів *k-nearest neighbors algorithm* (KNN). Він використовує просту логіку: об'єкт відноситься до того ж класу, що більшість його найближчих сусідів.

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

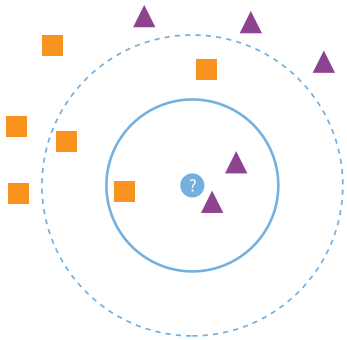
Оскільки ознаки, на основі яких проводиться класифікація, можуть мати різну фізичну природу і, відповідно, діапазони значень, для покращення результатів класифікації буде корисно виконати нормалізацію навчальних даних.

### 1.1.1 Алгоритм

Нехай є набір даних, що складається з $n$ спостережень $X_i (i = 1, \ldots, n)$ для кожного з яких заданий клас $C_j (j = 1, \ldots, m)$. Тоді на його основі може бути сформована навчальна множина, всі приклади якої являють собою пари $X_i, C_j$.

Алгоритм KNN можна розділити на дві прості фази: навчання та класифікації. Під час навчання алгоритм просто запам'ятовує вектори ознак спостережень та його мітки класів (тобто приклади). Також задається параметр алгоритму $k$, який задає число "сусідів", які будуть використовуватися при класифікації.

На фазі класифікації пред'являється новий об'єкт, котрого мітка класу не задана. Для нього 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 відноситься об'єкт, що класифікується. Пояснимо роботу алгоритму за допомогою рисунка.




Робота алгоритму KNN

Кружком представлений об'єкт, який потрібно класифікувати, віднісши до одного з двох класів "трикутники" та "квадрати". Якщо вибрати $k = 3$, то з трьох найближчих об'єктів виявляться "трикутниками" і один "квадратом". Відтак новому об'єкту буде надано клас "трикутник". Якщо поставити $k = 5$, то з п'яти "сусідів" два будуть "трикутники" і три "квадрати", в результаті об'єкт, що класифікується, буде розпізнаний як "квадрат".

### 1.1.2 Визначення класу нового об'єкта

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

$$D_j = \sqrt{\sum_{i=1}^{n}(x_i - a_i)^2}, \tag{1.1}$$

де $x_i$ – $i$-та ознака об'єкта, що класифікується $a_i$ – ознака класифікованих об'єктів.

Однак на практиці це не завжди вдале рішення, наприклад, якщо частота появи для двох або більше класів виявляється однаковою. І тут використовують деяку функцію, з допомогою якої визначається клас, звану функцією поєднання (combination function).

У звичайному випадку використовують так зване *просте невиважене голосування (simple unweighted voting)*. При цьому передбачається, що всі $k$ прикладів мають однакове право "голосу" незалежно від відстань 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. І тут використовують *зважене голосування (weighted voting)*.

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 $j$-го класу до класифікованого об'єкта (часто це значення називають *показником близькості*):

$$Q_j = \sum_{i=1}^{n_j} \frac{1}{D^2(x, a_{ij})}, \tag{1.2}$$

де $D$ – оператор обчислення відстані за формулою (1.1), $x$ – вектор ознак об'єкта, що класифікується, $a_{ij}$ – $i$-й приклад $j$-го класу. Таким чином, "перемагає" той клас, для якого величина $Q_j$ виявиться найбільшою. При цьому також знижується ймовірність того, що класи отримають однакову кількість голосів.

### 1.1.3 Вибір значення параметра $k$

Вибір параметра $k$ є важливим отримання коректних результатів класифікації. Якщо значення параметра мало, виникає ефект перенавчання , коли рішення щодо класифікації приймається з урахуванням малого числа прикладів і має низьку значущість. Це схоже на перенавчання у деревах рішень, коли в них багато правил, що належать до невеликої кількості прикладів. Якщо встановити $k = 1$, то алгоритм буде просто надавати будь-якому новому спостереженню мітку класу найближчого об'єкта.

Крім цього, слід враховувати, що використання невеликих значень $k$ збільшує вплив шумів на результати класифікації, коли невеликі зміни даних призводять до великих змін у результатах класифікації. Але при цьому межі класів виявляються більш вираженими (клас при голосуванні перемагає з великим рахунком).

Навпаки, якщо значення параметра занадто велике, то процесі класифікації бере участь багато об'єктів, які стосуються різних класів. Така класифікація виявляється дуже грубою і погано відбиває

локальні особливості набору даних. Таким чином, вибір параметра $k$ є компромісом між точністю та узагальнюючою здатністю моделі.

При великих значеннях параметра $k$ зменшується шумування 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.

### 1.1.4 Значущість ознак

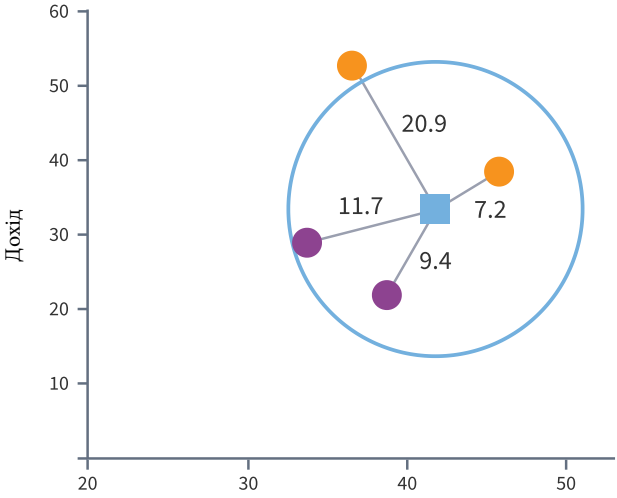
Для більшості методів класифікації існує проблема пов'язана з різною значущістю ознак з погляду визначення класу об'єктів. Врахування фактора значущості ознак в алгоритмі може дозволити підвищити точність класифікації. Для цього на основі суб'єктивної або деякої формальної оцінки можна задати рівень значущості ознаки, висловивши його за допомогою числового коефіцієнта (позначимо його $s$ від англ significance — значимість), який враховується при обчисленні відстані між прикладами та об'єктом, що класифікується:

$$D_E = \sqrt{\sum_{i=1}^{n} s_i(x_i - a_i)^2}, \tag{1.3}$$

де $s_i (i = 1, \ldots, p)$ – коефіцієнт значущості для $i$-ї ознаки, $p$ - кількість ознак вихідного набору даних. Такий прийом називається *розтягуванням осей* і він дозволяє збільшити або зменшити внесок ознаки у обчислення відстані від прикладу до об'єкта, що класифікується. Якщо $s_i > 1$, то завдяки відповідній ознакі відстань між прикладом об'єктом, що класифікується, зростає і внесок у визначення класу падає, а якщо $0 < s_i < 1$, то навпаки.

### 1.1.5 Чисельний приклад

Розглянемо простий чисельний приклад роботи алгоритму KNN, проілюстрований на рисунку 2.2. Нехай є набір даних про позичальників банку частина з яких допустили прострочення платежу (таблиця). Ознаками є вік та середньомісячний дохід. Мітками класу в полі "Прострочено" будуть "Так" та "Ні".




| $X_1$ – вік | $X_2$ – дохід | Прострочено |
|---|---|---|
| 46 | 40 | Ні |
| 36 | 54 | Ні |
| 34 | 29 | Так |
| 38 | 23 | Так |

Робота алгоритму KNN

На рисунку помаранчевими кружками представлені об'єкти класу "Ні", а фіолетовими класу "Так". Синім квадратом відображається об'єкт, що класифікується (новий позичальник).

Завдання полягає в тому, щоб виконати класифікацію нового позичальника для якого $a_1 = 42$ і $a_2 = 34$ з метою оцінити можливість прострочення ним платежів.

1. Задамо значення параметра $k = 3$

2. Розрахуємо відстань між вектором ознак об'єкта, що класифікується, і векторами навчальних прикладів за формулою (1.1) та встановимо для кожного прикладу його ранг.

| $X_1$ – вік | $X_2$ – дохід | Відстань | Ранг | Сусід | Прострочено |
|---|---|---|---|---|---|
| 46 | 40 | 7.2 | 1 | Так | Ні |
| 36 | 54 | 20.9 | 4 | Ні | Ні |
| 34 | 29 | 9.4 | 2 | Так | Так |
| 38 | 23 | 11.7 | 3 | Так | Так |

3. Виключимо з розгляду приклад, який при $k = 3$ не є сусідом і розглянемо класи, що залишилися.

| $X_1$ – вік | $X_2$ – дохід | Відстань | Ранг | Сусід | Прострочено |
|---|---|---|---|---|---|
| 46 | 40 | 7.2 | 1 | Так | Ні |
| 34 | 29 | 9.4 | 2 | Так | Так |
| 38 | 23 | 11.7 | 3 | Так | Так |

Таким чином, з трьох найближчих сусідів (на рисунку розташовані всередині кола) об'єкта, що класифікується, два мають клас "Так", а один – "Ні". Отже, шляхом простого невиваженого голосування визначаємо його клас як "Так".

Для проведення зваженого голосування розрахуємо показник близькості за формулою (1.2):

$$Q_{NO} = \frac{1}{65} \approx 0.015 \qquad Q_{YES} = \frac{1}{98} + \frac{1}{137} \approx 0.018.$$

Шляхом зваженого голосування визначаємо клас об'єкта, що класифікується, як "Так".

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

### 1.1.6 Області застосування алгоритму

Алгоритм KNN може застосовуватися практично у всіх завданнях класифікації, особливо у випадках, коли оцінити параметри ймовірнісного розподілу даних складно чи неможливо. Найбільш типовими програмами алгоритму KNN є:

- класифікація клієнтів (наприклад, за рівнем лояльності );

- медицина – класифікація пацієнтів за медичними показниками;
- маркетинг – класифікація товарів за рівнем популярності і т.д.

### 1.1.7 Переваги та недоліки алгоритму

На закінчення відзначимо переваги та недоліки алгоритму KNN. До переваг алгоритму можна віднести.

- стійкість до викидів і аномальних значень, оскільки ймовірність попадання записів, що містять їх, до k-найближчих сусідів мала. Якщо ж це сталося, то вплив на голосування (особливо зважене) також, швидше за все, буде незначним, а отже, малим буде і вплив на результати класифікації;
- програмна реалізація алгоритму доволі проста;
- результати роботи алгоритму легко піддаються інтерпретації. Логіка роботи алгоритму зрозуміла експертам у різних галузях.

До недоліків алгоритму KNN можна віднести:

- даний метод не створює будь-яких моделей, які узагальнюють попередній досвід, а інтерес можуть становити й самі правила класифікації;
- при класифікації об'єкта використовуються всі доступні дані, тому метод KNN є досить витратним у плані, особливо у разі великих обсягів даних;
- висока трудомісткість через необхідність обчислення відстаней до всіх прикладів;
- підвищені вимоги до репрезентативності вихідних даних.

### 1.1.8 Постановка задачі

Провести класифікацію даних з використанням методу дерева рішень. Етапи розв'язання

1. Імпортувати вибірку для проведення навчання
2. Розділити всю вибірку на дві частини (навчальну та тестову)
3. Побудувати алгоритм класифікації
   (а) Зафіксувати значення параметра $k$
   (б) Встановити значення ваг кожної з ознак рівним 1
   (в) Для кожного об'єкта з тестової вибірки визначити до якого класу з навчальної вибірки він належить
4. Порахувати кількість помилок класифікації шляхом порівняння результатів класифікації з відомими значеннями класів для кожного об'єкта з тестової вибірки
5. Провести оптимізацію параметра $k$ та ваг кожної ознаки
6. Порівняти результати з knn з sklearn
7. Оформити результати у вигляді звіту.