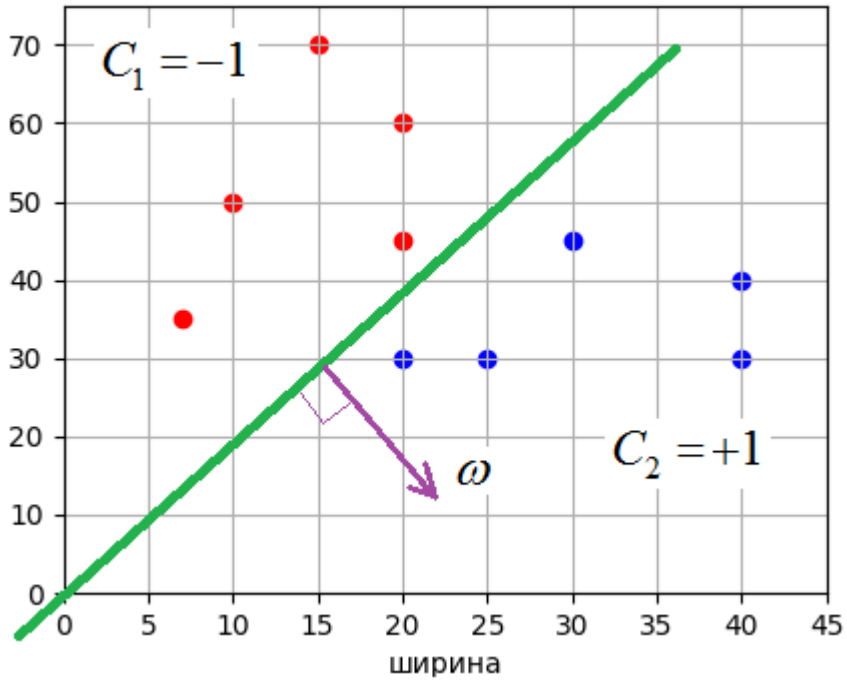
Заняття 2. Класичне машинне навчання

Задача 1.

Створити класифікатор, який би відрізняв гусениць від сонечок за двома результатами вимірювання: шириною і довжиною. Таким чином, наша навчальна вибірка буде складатися з наступних спостережень:

№	Ширина	Довжина	Жук
1	10	50	гусениця
2	20	30	сонечко
3	25	30	сонечко
4	20	60	гусениця
5	15	70	гусениця
6	40	40	сонечко
7	30	45	сонечко
8	20	45	гусениця
9	40	30	сонечко
10	7	35	гусениця

Візуально набір розмірів влаштований наступним чином:



Тут червоними точками відзначені гусениці, а синіми - сонечка. Причому гусениці матимуть цільове значення -1, а сонечка – +1. Також можна побачити, що навчальна вибірка

утворює лінійно відокремлювані класи, а лінія поділу може бути проведена через початок координат. Тому будемо шукати рівняння прямої у вигляді:

$$\omega_1 x_1 + \omega_2 x_2 = 0$$

Спростимо вираз, перепишемо його, наступним чином:

$$x_2 = -\frac{\omega_1}{\omega_2} x_1$$

Якщо прийняти $\omega_2 = -1$, тоді залишається лише один параметр, який треба вибрати:

$$x_2 = \omega_1 x_1$$

І загальний вектор:

$$\omega = \left[\omega_1, -1\right]^T$$

Власне, тут ω_1 - кутовий коєфіцієнт прямої і нам потрібно його знайти.

Приблизно в середині 1950-х років Френк Розенблатт вирішив аналогічну проблему. Критерій якості лінії поділу він сформулював як число неправильних класифікацій:

$$Q\left(a, X^l\right) = \sum_{i=1}^{l} \left[a\left(x_i\right) \neq y_i\right]$$

Тут квадратні дужки є індикатором помилки, вони переводять логічні значення True і False в значення 1 і 0 (позначення Айверсона):

$$\left[a\left(x_i\right) \neq y_i\right]$$

True → 1 *False* → 0

У цьому випадку, якщо відповідь вирішального правила $a(x)$ не відповідає цільовому показнику y, тоді значення True, тобто, 1. И, навпаки, якщо буде збіг, ми отримаємо False і значення 0. В результаті функціонал $Q\left(a, X^l\right)$ підрахує кількість помилок класифікації.

Однак при розв'язанні задач бінарної класифікації, коли цільові відповіді вибираються з множини $y \in \{-1, +1\}$, цей же показник якості можна розрахувати наступним чином:

$$Q\left(a, X^l\right) = \sum_{i=1}^{l} \left[y_i \cdot a\left(x_i\right) < 0\right]$$

Тут величина $y_i \cdot a(x_i)$ прийме від'ємні значення з помилковими класифікаціями і додатні значення з правильними. Дійсно, якщо знаки величин y_i, $a(x_i)$ збігаються, а це означає, що ми правильно віднесли зображення до потрібного класу і добуток буде позитивним. В іншому випадку знаки будуть іншими, і добуток буде менше нуля.

Такий добуток дуже часто використовується при розробці алгоритмів бінарної класифікації. Позначається великою літерою M:

$$M_i = a(x_i) \cdot y_i$$

і називається **відступом** (в перекладі з англійської. margin). Ця величина може показати не тільки ознаку правильної класифікації, але і те, як далеко зображення знаходиться від площини, що ділить.

$$a(x) = \langle \omega, x \rangle$$

і

$$M_i = \langle \omega, x \rangle \cdot y_i$$

Але поки що нам це не потрібно, і ми залишимо sign(). Тут головне пам'ятати, як розраховується відступ і що він означає.

Отже, сформульовано критерій якості вирішення задачі:

$$Q(a, X^l) = \sum_{i=1}^{l} [M_i < 0] \to \min$$

Як ми використовуємо його зараз, щоб знайти шанси ω_1? Очевидно, що математично мінімізувати цей функціонал неможливо, оскільки він є кусково неперервною недиференційовною функцією. Тому рішення буде алгоритмічним, подібним до того, яке запропонував Френк Розенблатт.

- **Вхідні дані:** Вибірка X^l, крок навчання η, Максимальна кількість ітерацій N
- **Вихід: вектор ваг** $\omega = [\omega_1, \omega_2]^T$

1) Ініціалізація $\omega = [0, -1]^T$

2) повторити N раз

3) По черзі вибираємо x_i, y_i з навчальної вибірки X^l

4) якщо $M_i = sign\left(\left\langle \omega, x_i \right\rangle\right) \cdot y_i < 0$, що

5) Скоригуємо ваги: $\omega_1 = \omega_1 + \eta \cdot y_i$

6) Обчислення показника якості $Q\left(a, X^l\right) = \sum_{i=1}^{l} \left[M_i < 0\right]$

7) якщо $Q\left(a, X^l\right) = 0$, тоді перериваємо цикл (рішення знайдено)

Задача 2.

Аналітичний розв'язок

Кожне спостереження представлено двома ознаками: шириною і довжиною. В нашому випадку вектор параметрів ω складається з трьох компонентів:

$$\omega = \left[\omega_0, \omega_1, \omega_2\right]^T$$

(додається зсув), так як рівняння розділової лінії записується у вигляді:

$$\omega_2 x_2 + \omega_1 x_1 + \omega_0 = 0$$

Тому додамо до наших спостережень ще одну ознаку – константу 1:

$$x = \left[x_1, x_2, 1\right]^T$$

Після виконання ми побачимо значення коефіцієнтів:

[0.05793234 -0.0346272 0.1912188]

Тобто тут

$$\omega_1 = 0.05793234;\ \omega_2 = -0.0346272;\ \omega_0 = 0.1912188$$

Кутовий коефіцієнт прямої:

$$k = -\frac{\omega_1}{\omega_2} \approx 1.67$$

Задача 3. Масштабування даних

Масштабування означає, що ви трансформуєте ваші дані так, щоб вони вписувалися в певний масштаб, наприклад, від 0 до 100 або від 0 до 1. Масштабування даних необхідне,

коли ви використовуєте методи, засновані на вимірюванні відстаней між точками даних, такі як метод опорних векторів (SVM) або метод найближчих сусідів (KNN). Для цих алгоритмів зміна на "1" в будь-якій числовій ознаці має однакову важливість.

Масштабуючи свої змінні, ви можете допомогти порівнювати різні змінні на рівних умовах. Щоб закріпити, як виглядає масштабування, давайте розглянемо вигаданий приклад.

Приклад:

Є набір даних: Висота і вага людей.
Якщо ви використовуєте методи, які враховують відстань між точками даних, важливо, щоб ці набори даних були масштабовані до одного діапазону значень. Це дозволить уникнути ситуації, коли одна ознака домінує над іншою через різний масштаб.
Вибір функції масштабування залежить від характеру ваших даних та типу алгоритму машинного навчання, який ви використовуєте. Ось кілька рекомендацій щодо вибору функції масштабування:

1. Для алгоритмів на основі відстані (KNN, SVM, нейронні мережі): Найчастіше використовують MinMaxScaler або StandardScaler.
2. Для алгоритмів, які припускають нормальний розподіл (лінійна регресія, логістична регресія, PCA): Використовують StandardScaler.
3. Для даних з викидами: Використовують RobustScaler. Цей метод менш чутливий до викидів.

Приклад: Фінансові дані, де можуть бути великі відхилення в значеннях.

Нормалізація даних.

Масштабування змінює лише діапазон ваших даних. Нормалізація ж є більш радикальною трансформацією. Мета нормалізації полягає в тому, щоб перетворити ваші спостереження так, щоб їх можна було описати за допомогою нормального розподілу.
Нормальний розподіл: також відомий як "дзвоноподібна крива", це специфічний статистичний розподіл, де приблизно однакова кількість спостережень падає вище та нижче середнього значення, середнє та медіанне значення збігаються, і більшість спостережень ближчі до середнього значення. Нормальний розподіл також відомий як Гауссів розподіл.
Загалом, ви будете нормалізувати свої дані, якщо плануєте використовувати методи машинного навчання або статистичні методи, які припускають, що ваші дані мають нормальний розподіл.
Задача 4

Є набір даних, що містить інформацію про пацієнтів у лікарні, включаючи їх вік, вагу і рівень цукру в крові. Ці дані потрібно нормалізувати, щоб використовувати в алгоритмах машинного навчання, таких як лінійна регресія або кластеризація. Нормалізація даних допомагає алгоритмам машинного навчання швидше сходитися та підвищити точність моделей.

Вік (роки)	Вага (кг)	Рівень цукру (мг/дл)
25	80	90
45	120	85
35	60	110
50	100	95
23	75	105

Задача 5. Завантажити датасет.

Зробити попередній аналіз даних. Датасет містить наступні ознаки:

Ознаки (X):

- вік: вік пацієнта (років)
- анемія: зниження еритроцитів або гемоглобіну (булеве значення)
- креатинінфосфокіназа (КФК): рівень ферменту КФК у крові (мкг/л)
- діабет: якщо у пацієнта діабет (булеве значення)
- фракція викиду: відсоток крові, що залишає серце при кожному скороченні (у відсотках)
- високий кров'яний тиск: якщо у пацієнта гіпертонія (булеве значення)
- тромбоцити: тромбоцити в крові (кілотромбоцити/мл)
- стать: жінка або чоловік (бінарний)
- сироватковий креатинін: рівень сироваткового креатиніну в крові (мг/дл)
- сироватковий натрій: рівень сироваткового натрію в крові (мекв/л)
- куріння: курить пацієнт чи ні (булеве значення)
- час: період спостереження (днів)
- DEATH_EVENT: якщо пацієнт помер протягом періоду спостереження (булеве значення)
  Побудувати модель лін. Регресії, а також з врахуванням L1 і L2. Зробити аналіз результатів.

  Результат:

- Linear Regression: 0.1807476509625888
- Lasso Model MSE: 0.13399318201079297
- Ridge Model MSE: 0.12696821193956367

Ці значення вказують на те, що модель Ridge має найменше MSE, тобто є кращою за інші моделі у прогнозуванні на цьому наборі даних. Модель Lasso також показує кращі результати, ніж проста лінійна регресія, що підтверджує ефективність регуляризації в порівнянні з класичним підходом.