



Основи Машинного Навчання (AI)

Ознайомтеся зі світом штучного інтелекту та машинного навчання. Вивчіть класифікацію, регресію, кластеризацію, теорему Баєса, та ключові метрики оцінки моделей. Опануйте алгоритми лінійної регресії та K-наближчих сусідів.

Курс поєднує теорію з практикою в Python, готуючи вас до вирішення реальних задач з машинного навчання.



by Oleksandr Denishchuk



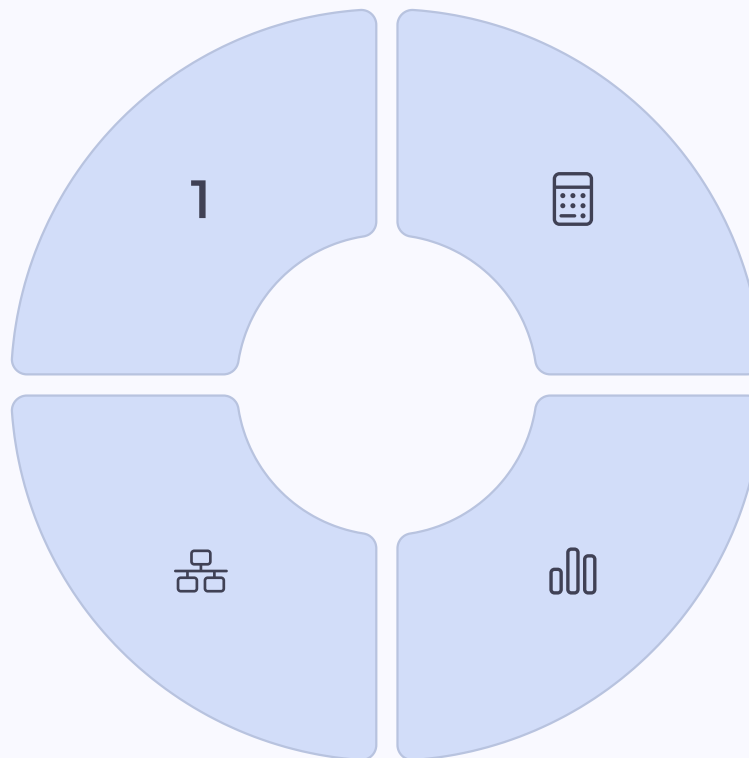
Розкриваємо Таємниці Машинного Навчання

Машинне навчання

Машинне навчання (ML) - це підрозділ штучного інтелекту, де комп'ютери навчаються з даних, виявляють закономірності і приймають рішення без прямого програмування.

Основні методи

Три підходи: навчання з учителем (класифікація, регресія), без учителя (кластеризація) та з підкріпленням (рішення в динамічних середовищах).



Теорема Баєса

Математична основа імовірнісних моделей ML. Оновлює прогнози на основі нових даних у класифікації та системах прийняття рішень.

Функції втрат

Вимірюють точність моделі. MSE для регресії. Їх мінімізація - ключове завдання.



Machine Learning (ML)?

Основні концепції ML

- Алгоритми навчаються на великих масивах даних.
- Системи самовдосконалюються без прямого програмування.
- Базується на математичних моделях та теоремі Баєса.
- Якість оцінюється через функції втрат.
- Використовує класифікацію та регресію для розв'язання задач.

Практичне застосування:

- Розпізнавання облич у безпеці та соцмережах 🧠
- Персоналізація контенту (Netflix, YouTube) 🎬
- Голосові помічники (Siri, Alexa) 🗣️
- Прогнозування поведінки споживачів 📊
- Діагностика за медичними знімками 🏥

Види задач Machine Learning

Класифікація

- **Класифікація** – віднесення об'єктів до заздалегідь визначених категорій
- 📧 Спам-фільтрація електронних листів
- 👤 Розпізнавання облич на фотографіях
- 🏥 Діагностика захворювань за симптомами

Регресія

- **Регресія** – прогнозування числових значень на основі даних
- 🌡️ Передбачення температури повітря
- 📈 Прогноз курсу валют
- 🏠 Оцінка вартості нерухомості за параметрами

Кластеризація

- **Кластеризація** – автоматичне групування схожих об'єктів
- 👥 Сегментація клієнтів за поведінкою
- 🔍 Пошук аномалій у банківських транзакціях
- 📊 Групування товарів за характеристиками

Кожен тип задач використовує специфічні алгоритми та функції втрат, що дозволяє оптимізувати моделі під конкретні потреби бізнесу чи наукових досліджень.

Таємна зброя: Теорема Баєса

Що це таке?

Проста формула: $P(A|B) = P(B|A) \times P(A) / P(B)$

Допомагає перерахувати ймовірність події A, коли ми спостерігаємо подію B

Основа для багатьох алгоритмів машинного навчання

Використовується від простих класифікаторів до складних моделей

Ключові поняття

Чутливість і специфічність

Показники точності розпізнавання позитивних і негативних випадків. Важливі для медичної діагностики

Приклад з медицини

Навіть якщо тест точний на 98%, але хвороба рідкісна (1% населення), позитивний результат означає хворобу лише у третині випадків

Задача про бібліотекаря

Показує, як легко помилитися, оцінюючи професію людини тільки за її рисами, без врахування того, скільки людей працює в різних професіях

Гра "Три двері"

Змінивши свій вибір дверей після підказки ведучого, ви збільшуєте шанси на виграш вдвічі

Як це працює в машинному навчанні:

Алгоритм починає з приблизного припущення і постійно уточнює його, отримуючи нові дані. Це схоже на те, як ми самі навчаємось, поєднуючи попередній досвід з новою інформацією.

Приклад: При фільтрації спаму система обчислює, наскільки ймовірно, що повідомлення зі словом "виграш" є спамом, базуючись на попередніх прикладах. Хоча метод і спрощений, він працює напрочуд добре.

Як модель розуміє, що вона помиляється?

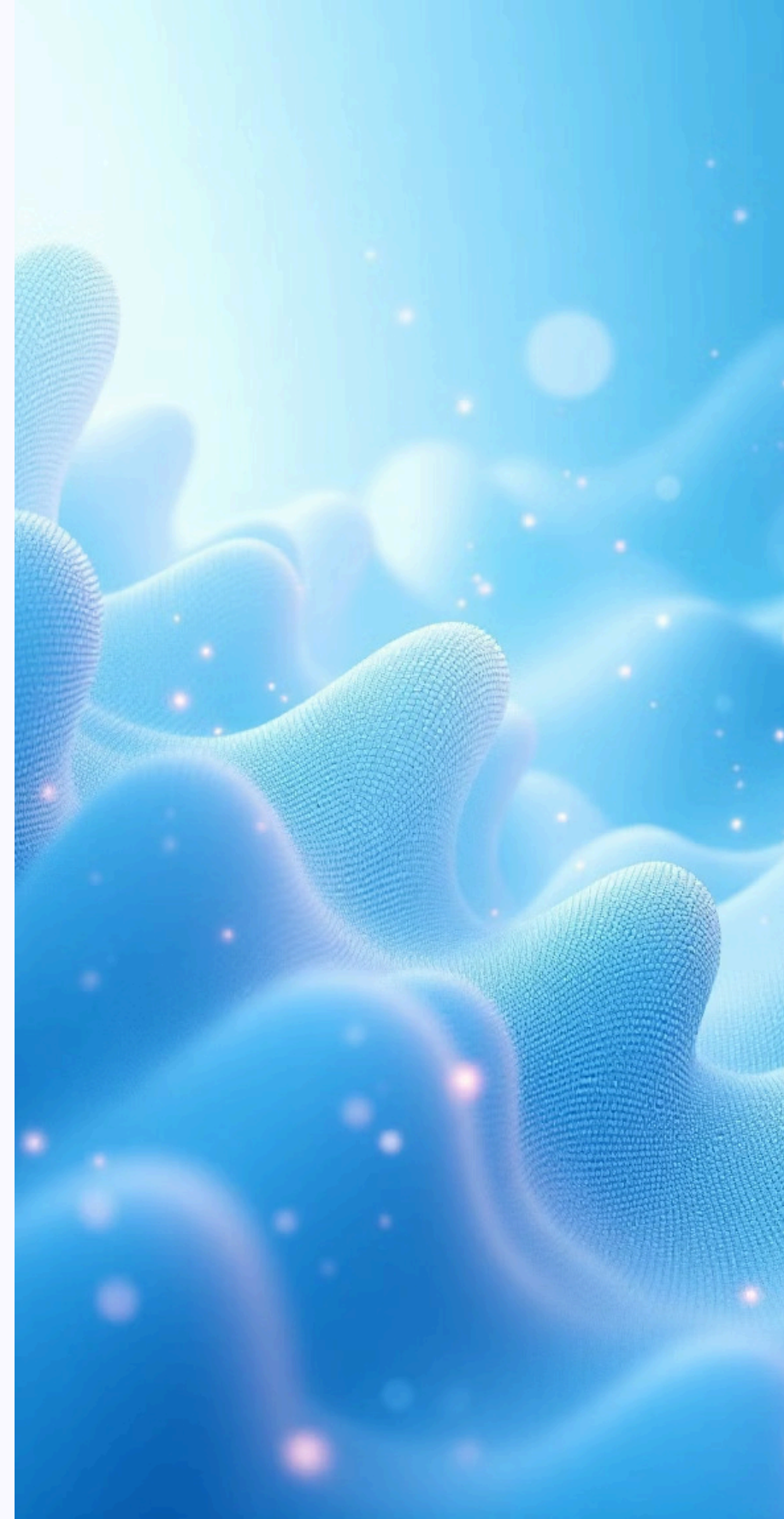
Функції втрат (Loss Functions) – математичні функції, що:

- 1** Кількісно оцінюють розбіжність між прогнозованими та фактичними значеннями
- 2** Спрямовують процес навчання через градієнтний спуск
- 3** Дозволяють моделі самостійно коригувати свої параметри

Найпоширеніші функції втрат:

- **MSE (середньоквадратична помилка)** – ідеальна для задач регресії, чутлива до викидів
- **MAE (середня абсолютна помилка)** – менш чутлива до викидів, підходить для зашумлених даних
- **Cross-Entropy** – оптимальна для задач класифікації, вимірює відстань між імовірнісними розподілами

Мета оптимізації: мінімізувати значення функції втрат через ітеративне налаштування вагових коефіцієнтів моделі, подібно до того, як ми використовуємо теорему Байеса для уточнення наших припущень з новими даними



Лінійна регресія

1 Призначення:

Метод Машинного Навчання для **прогнозування числового значення** (цільової змінної).

2 Як працює:

Шукає **лінійну залежність** між однією або кількома вхідними ознаками (факторами) та цільовим значенням. По суті, намагається знайти **пряму лінію** (або гіперплощину у багатовимірному просторі), яка **найкраще описує** зв'язок у даних.

3 Формула (загальний вигляд):

$$y = b_0 + b_1x_1 + b_2x_2 + \dots + b_nx_n$$

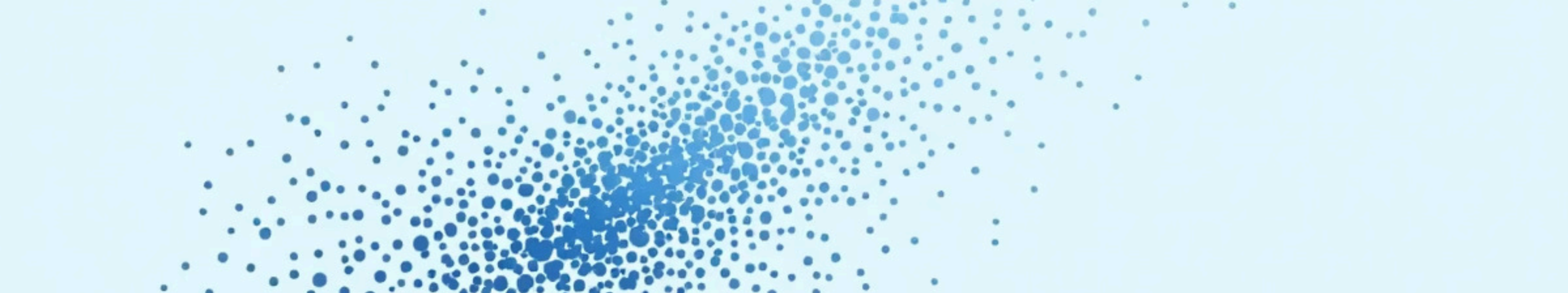
- y - передбачуване значення
- x_i - вхідні ознаки
- b_i - коефіцієнти (нахили, ваги), що вказують на вплив кожної ознаки
- b_0 - зсув (точка перетину з віссю Y)

4 Мета:

Знайти такі коефіцієнти (b_i), щоб **мінімізувати помилку** між передбаченими моделлю значеннями та фактичними значеннями в даних.

5 Приклади застосування:

Прогнозування ціни нерухомості, передбачення обсягів продажів, оцінка витрат.



К-найближчих сусідів (KNN): Класифікація на основі близькості

1

Призначення:

Простий та інтуїтивний алгоритм ML для **класифікації** нових об'єктів.

2

Основна ідея:

Новий об'єкт відноситься до класу, який є **найбільш поширеним серед його K найближчих "сусідів"** у просторі даних.

3

Що таке "K":

Кількість найближчих відомих точок даних, які ми розглядаємо для "голосування" за клас.

4

Що таке "найближчий":

Визначається за допомогою **функції відстані** (наприклад, Евклідова відстань) між об'єктами за їхніми ознаками.

5

Як відбувається класифікація:

Серед K найближчих сусідів підраховується, скільки належить до кожного класу. Новий об'єкт отримує клас, який набрав **найбільше голосів** (має більшість).

6

Особливість:

Немає фази "навчання" моделі як такої (алгоритм просто запам'ятовує дані). Прогноз робиться під час запиту, що може бути повільно на дуже великих даних.

7

Приклади:

Визначення типу клітини, класифікація рукописних цифр, рекомендації.

Світ ML — це не лише лінійна регресія!

Розглянемо ключові алгоритми машинного навчання та їх застосування:

1

■ Лінійна регресія: Основа прогнозування числових значень

Знаходить найкращу пряму лінію для прогнозування на основі даних. Застосовується для прогнозування цін на нерухомість, аналізу продажів та економічних трендів. Навчання відбувається шляхом поступового покращення моделі, щоб зменшити помилки в прогнозах.

2

■ Логістична регресія: Потужний інструмент для класифікації

Визначає ймовірність належності об'єкта до певного класу (наприклад, "так" чи "ні"). Використовується в медичній діагностиці, оцінці кредитоспроможності та маркетингу. Модель навчається розрізняти між класами і штрафується за помилкові прогнози.

3

■ Древа рішень: Інтуїтивний підхід до класифікації та регресії

Працює як серія питань "так/ні", щоб діяти до висновку. Наприклад: "Вік більше 30?" → "Дохід вище середнього?" тощо. Ідеально підходить для кредитного скорингу, медичної діагностики та систем рекомендації завдяки зрозумілості. Обмежує складність дерева для уникнення надмірного ускладнення.

4

■ Random Forest: Ансамблевий метод для підвищення точності

Створює багато різних дерев рішень (зазвичай 100-500) і комбінує їхні результати. Кожне дерево бачить лише частину даних і ознак. Застосовується в комп'ютерному зорі, біоінформатиці та фінансовому прогнозуванні, де показує вищу точність і стійкість до шуму порівняно з одиничними деревами.

5

■ К-найближчих сусідів (KNN): Класифікація на основі близькості

Визначає клас нового об'єкта за принципом "скажи мені, хто твої друзі, і я скажу, хто ти". Знаходить K найбільш схожих прикладів у навчальних даних і приймає рішення на основі їхніх класів. Ефективний для розпізнавання рукописного тексту, рекомендаційних систем і класифікації зображень при невеликих наборах даних.

6

■ Нейронні мережі: Біологічно натхненний підхід до глибокого навчання

Імітують роботу людського мозку, використовуючи шари з'єднаних "нейронів". Глибокі мережі мають спеціальні архітектури: CNN для зображень, RNN/LSTM для послідовностей (текст, мова) та Transformer для обробки природної мови. Навчаються шляхом коригування зв'язків між нейронами для зменшення помилок.

7

● K-Means: Алгоритм кластеризації для виявлення природних груп

Знаходить групи схожих об'єктів у даних. Процес: 1) вибір початкових центрів груп, 2) віднесення кожного об'єкта до найближчої групи, 3) оновлення центрів груп. Повторюється до стабілізації. Застосовується для сегментації клієнтів, спрощення зображень та виявлення аномалії. Оптимальну кількість груп визначають спеціальними методами.

Python – інструмент дослідника даних

📁 1. Дані — це таблиця з CSV-файлу або бази даних

Наприклад, файл квартири.csv:

```
площа,кімнати,район,ціна
50,2,A,80000
75,3,B,120000
60,2,C,95000
```

📦 2. Завантаження в Python:

```
import pandas as pd
df = pd.read_csv("квартири.csv")
```

df – це **DataFrame** (таблиця з бібліотеки pandas)

📊 3. Формування даних для ML:

✅ X — ознаки (тип: 2D масив / DataFrame):

```
X = df[["площа", "кімнати", "район"]]
```

→ Тип: pandas.DataFrame

→ Розмір: [кількість_рядків, кількість_ознак], напр. (3, 3)

🎯 y — ціль (тип: 1D масив / Series):

```
y = df["ціна"]
```

→ Тип: pandas.Series

→ Розмір: [кількість_рядків], напр. (3,)

🧠 Мета ML-моделі:

Навчитися передбачати y (ціну), аналізуючи X (площу, кімнати, район)



Навчаємо модель, але чесно 🎯✂️

📌 Навіщо ділити дані?

Ми хочемо, щоб модель **вміла передбачати нові дані**, а не просто “завчила” старі.

Тому ми ділимо дані на дві частини:

✅ Навчальні (Training set) – 80%

Модель вчиться: як ознаки X пов'язані з цілями y

❌ Але **не оцінюється** на них!

🔧 Тестові (Test set) – 20%

Тут ми **перевіряємо**, чи модель справді навчилась, а не просто запам'ятала

📦 Python-код:

```
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)
```

💡 Головна ідея:

Вчимо на одному, перевіряємо на іншому.

Це як готуватись до ЗНО: тренуєшся вдома, а справжній результат – на іспиті.



Ключові моменти, які ми вивчили

1 Функція втрат:
Математичний інструмент, що допомагає моделі "вчитися на помилках" та покращувати прогнози

2 Теорема Баєса:
Дозволяє визначати ймовірності подій на основі попередніх знань

3 Різні задачі ML:
Класифікація, прогнозування та кластеризація з практичним застосуванням

4 Лінійна регресія та KNN:
Алгоритми, які ми реалізували для моделювання даних

5 Багатовимірність не страшна:
За допомогою Python та правильних даних можна вирішувати складні задачі

