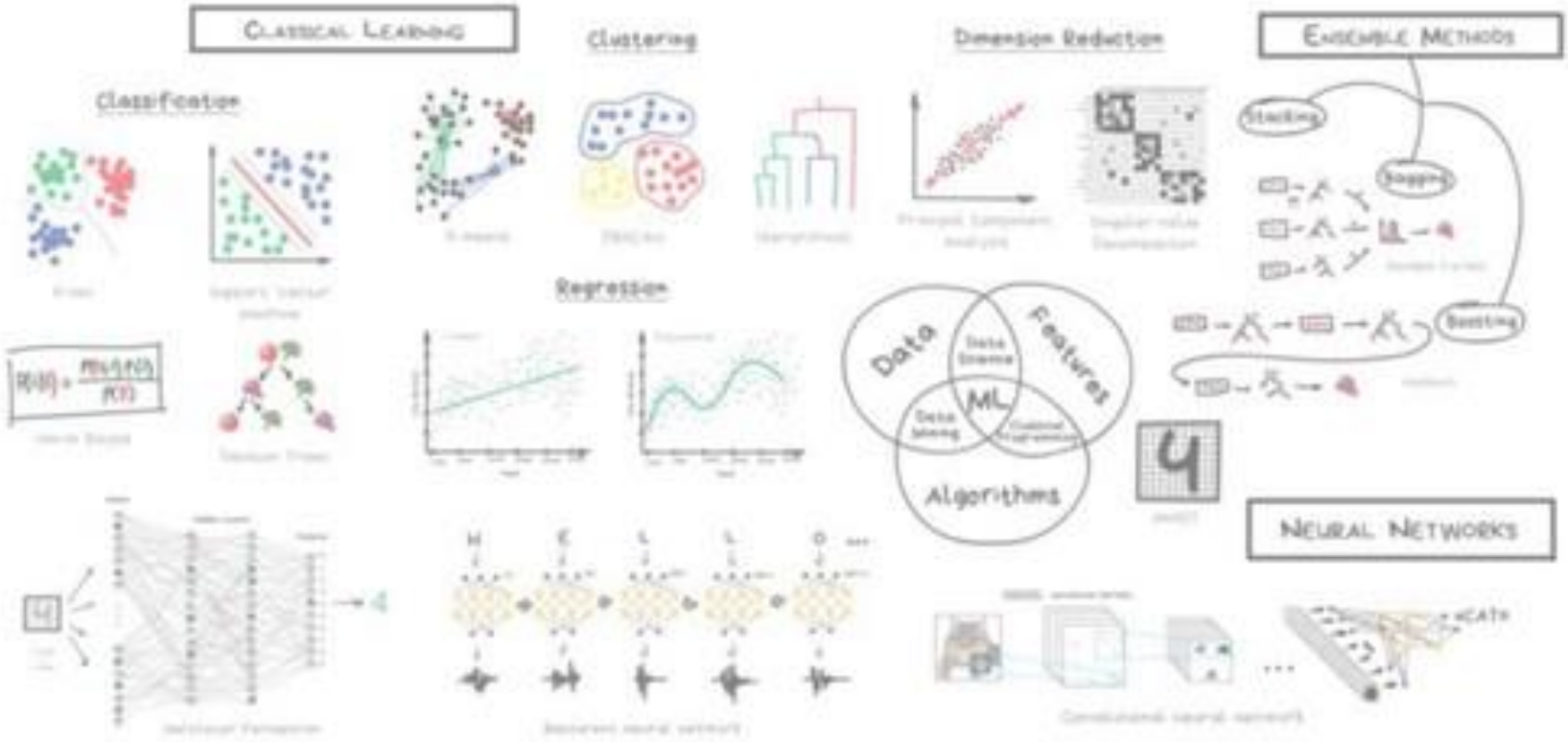
Машинне навчання



Лекція № 2

Основні поняття

Приклад: Уявіть, що ви – програміст, який працює в лікарні, і керівництво дає вам завдання написати програму, котра виявлятиме рак шкіри. Наприклад, щоб пацієнт міг сфотографувати нову родимку або іншу плямку на телефон, і додаток визначив би, чи бігти до лікаря. Проблема в тому, що ви нічого не тямите в онкології. Лікарі з клініки перерахували вам типові тривожні ознаки, але це не допомогло. Як це закодувати:

При цьому необхідно брати до уваги ще десятки інших характеристик десятків інших різновидів доброякісних і злоякісних пухлин. І щоб додатково ускладнити ситуацію, керівництво вам натякає: їм не подобається точність діагностики навіть досвідчених лікарів (20% помилок при простому огляді), і вони очікують, що ваша програма працюватиме значно краще.

Єдина позитивна сторона у вашій ситуації – те, що в архівах зберігається кілька тисяч фото як здорової, так і хворої шкіри.

Основні поняття

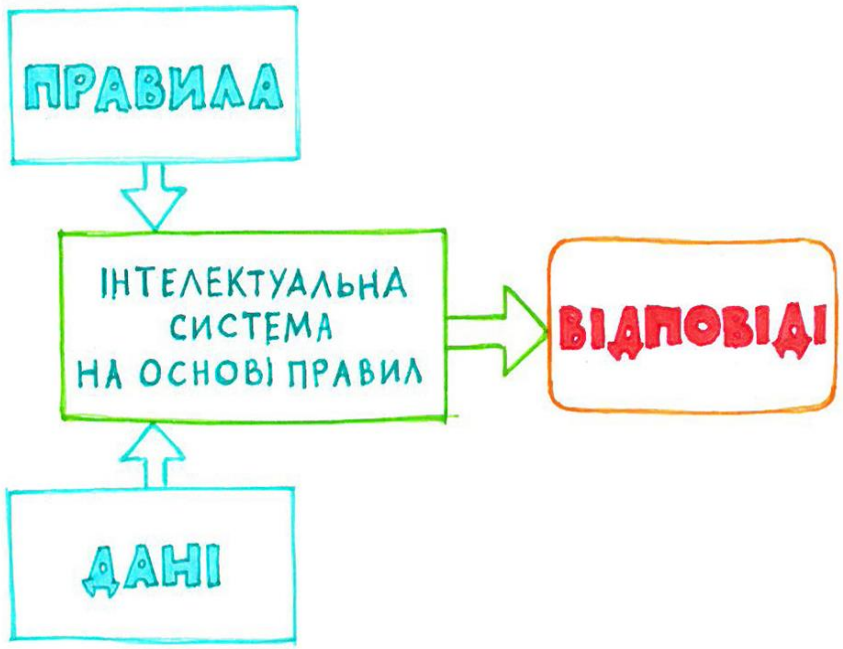
Та як написати і потім перетворити на код алгоритм визначення діагнозу, який працюватиме не з анкетою чи бланком, а безпосередньо із зображенням? Звичайним способом (мільйон if … else) – ніяк. Але це і непотрібно. Система з машинним навчанням (як то наприклад глибинна штучна нейронна мережа) навчиться відрізняти хвору шкіру від здорової, користуючись наявними архівними даними.

Що ж залишається робити людині-розробнику? Попивати мартіні під пальмою? Не все так просто. Нейромережі допомагають вирішити проблеми, що видаються невирішуваними, але не зменшують кількості роботи для програміста. Просто замість розробки алгоритму вирішення проблеми вам доведеться (кілька разів):

1. Провести попередній аналіз даних
2. Підготувати дані для введення в нейромережу
3. Підібрати (написати) метод навчання і аналізу (архітектуру нейромережі)
4. Провести навчання алгоритму на наявних даних
5. Провести перевірку алгоритму в реальному світі

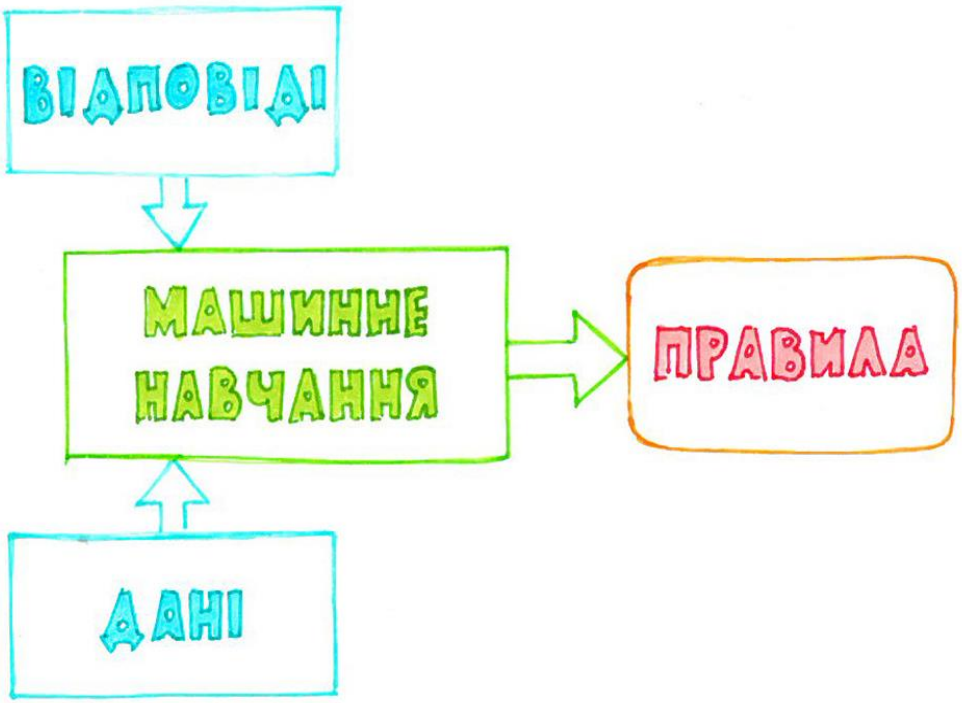
Основні поняття

Інтелектуальні системи без здатності до навчання ухвалюють рішення на основі даних, які до них надходять, і правил, написаних людиною-програмістом. Наприклад, калькулятор може додати 2 до 2, тому що хтось запрограмував для нього правила додавання. Для цього комп'ютеру не довелося вивчати тисячі прикладів додавання, програміст просто знав, як воно працює, і переклав мовою, зрозумілою комп'ютеру. Це прекрасно працює, коли сам програміст знає правила, які хоче запрограмувати.



Основні поняття

На жаль, в реальному житті це не завжди так. Як у прикладі з раком шкіри, програміст не завжди знає, за якими саме правилами повинна працювати програма. І тут нам на допомогу приходять системи з машинним навчанням на основі прикладів. Головна їх властивість – це здатність знаходити правила, використовуючи наявні приклади.



Основні поняття

Штучний інтелект — назва всієї області знань, як біологія чи хімія.

Машинне навчання — розділ штучного інтелекту.

Нейронні мережі (нейромережі) — один з видів машиного навчання. Популярний, проте існують й інші.

Глибоке навчання — архітектура нейромереж, один з підходів до їх побудови та навчання.

Основні поняття

Мета машиного навчання — передбачити результат за вхідними даними.
Чим різноманітніші вхідні дані, тим простіше машині знайти закономірності і тим точніше результат.

Машина може	Машина не може
Передбачати	Створювати нове
Запам’ятовувати	Різко порозумнішати
Відтворювати	Вийти за рамки задачі
Обирати краще	Вбити всіх людей

Машинне навчання

- У матеріалі Forbes, який з'явився у лютому 2018 року, **машинне навчання** розмістили на другій позиції в рейтингу найвпливовіших технологій найближчого майбутнього.

- CEO Google Сундар Пічаї (Sundar Pichai) каже, що штучний інтелект і, зокрема, **машинне навчання**, відіграють центральну роль у стратегії розвитку компанії.

- Більшість напрямів бізнесу Amazon об'єднує штучний інтелект: від алгоритмів рекомендацій до автоматизованих роботів, які керують складами.

- Очільник дослідницької групи прикладного машинного навчання Facebook Жоаквін Квінанеро Кандела стверджує, що сьогодні Facebook не може існувати без машинного навчання.

Три складові навчання

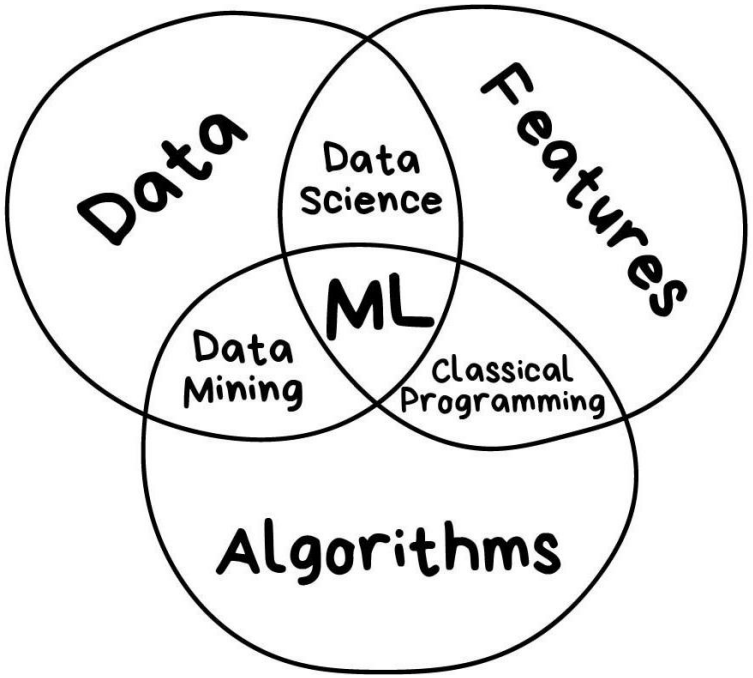
- Дані

Хочемо визначати спам — потрібні приклади спам-листів, передбачати курс акцій — потрібна історія цін, дізнатися про інтереси користувача — потрібні його лайки чи пости.

Даних потрібно якнайбільше !!!

- Ознаки (features) — властивості, характеристики,

Ознаки — ними можуть бути: пробіг автомобіля, стать користувача, ціна акцій, лічильник частоти появи слова в тексті, ...



- Алгоритм

Одну задачу можна вирішити різними методами приблизно завжди. Від вибору методу залежить точність, швидкість роботи та розмір готової моделі. Проте є один нюанс: якщо дані «погані», тонавіть найкращий алгоритм не допоможе у розв'язанні задачі.

!!! Не зациклюватися на відсотках, краще зібрати більше даних.

Попередній аналіз даних

Цей крок чи не найважливіший у циклі створення системи з машинним навчанням (нейромережі). Якщо вона не видає очікуваних результатів, до аналізу даних доводиться повертатися знову і знову.

На цьому етапі слід переконатися, що даних достатньо і вони не тенденційні. Наприклад, необхідно переконатися, що наявні дані адекватно відображають різні етнічні групи і гендери. Це досить складне завдання. Часто це неможливо зробити, не розпочавши створення самої системи з машинним навчанням.

Нерідко на цьому етапі все і завершується. Гірше, якщо це відбувається на десятій ітерації роботи Вашого алгоритму. Ще гірше – якщо після запуску продукту.

У такий спосіб Amazon одного разу створив нейромережу для відсіву претендентів на роботу, яка страждала від сексизму (насправді страждали, звісно, жінки, а не нейромережа).

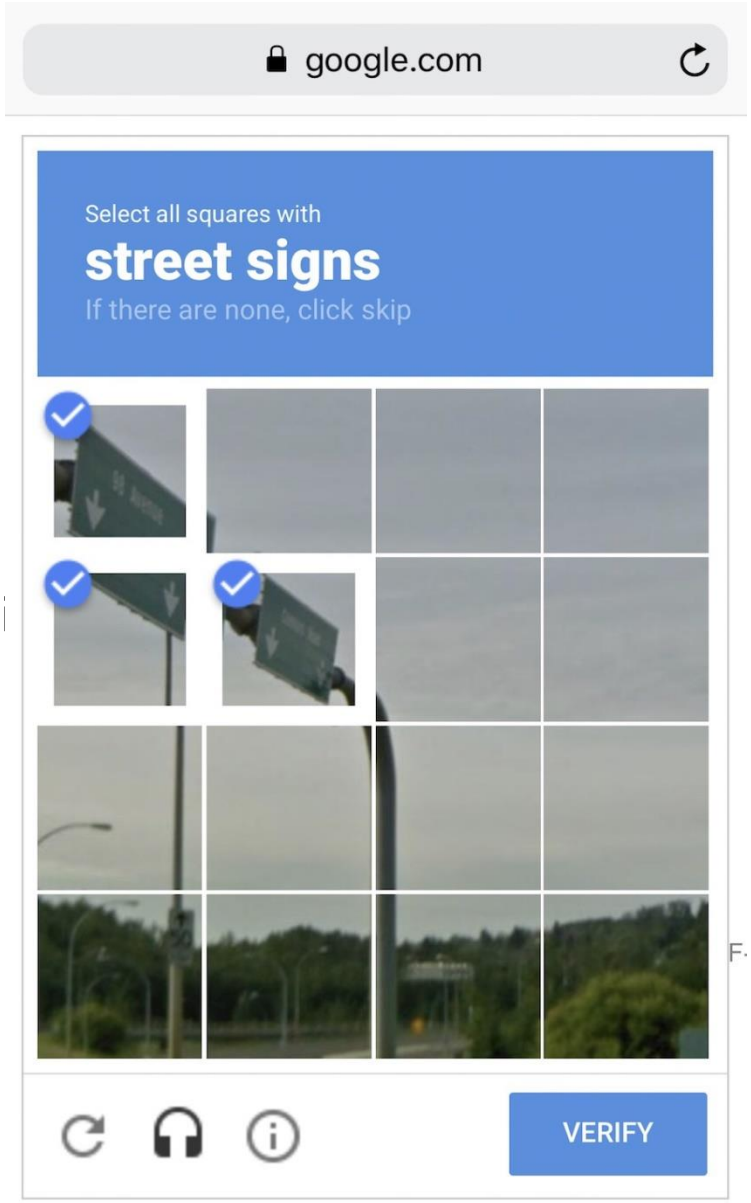
Попередній аналіз даних

Схожих прикладів безліч. У 2015 році Google потрапив у скандал через додаток Google Photos, який одночасно надає хмарне сховище для фотографій і групує зображення, щоб їх було легше знайти. Для цього Google Photos використовує технології машинного навчання та комп'ютерного зору, й інколи вони роблять прикрі помилки. На один з таких випадків натрапив програміст із Брукліна Джекі Алціне: Google Photos класифікував фотографію його чорношкірих друзів як світлину з горилами – він зробив знімок екрана і опублікував його у Twitter.

Дані

Дані збирають як можуть:
вручну - виходить довше, менше, проте без помилок;
автоматично - просто завантажуємо машині все, що знайшлося, і віримо у краще;
використання користувачів "ReCaptcha" (знайти на фотографії всі дорожні знаки).
За хорошими наборами даних (датасетами) йде велике полювання. Великі компанії, буває, розкривають свої алгоритми, але датасети вкрай рідко.

Доступні датасети
https://www.kaggle.com/datasets



Ознаки

Машина має знати, на що їй конкретно дивитися.

Добре, коли дані просто лежать у табличках — назви колонок і є ознаки. А якщо у нас сто гігабайт картинок із котами?

Коли ознак багато, модель працює повільно та неефективно.

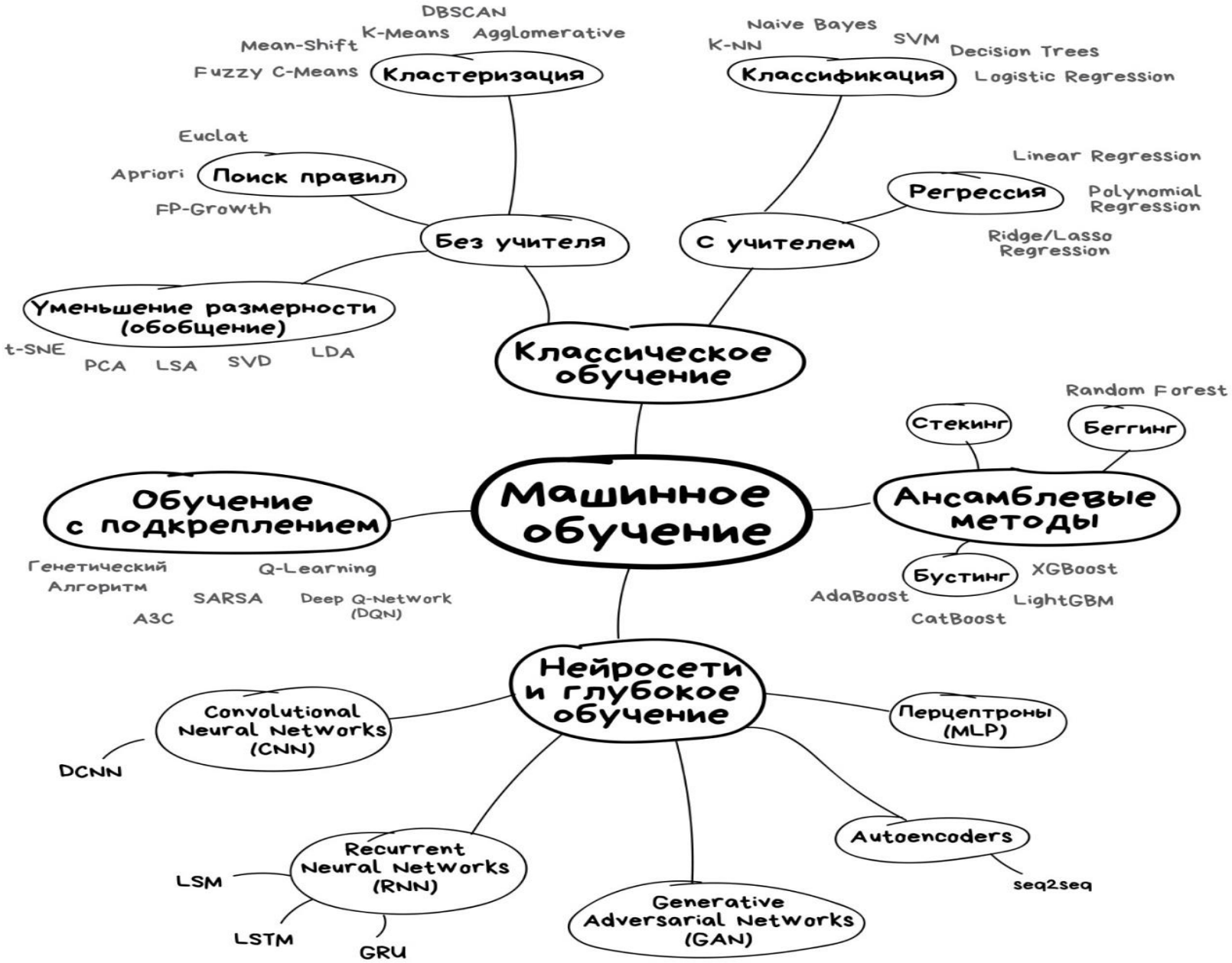
Найчастіше відбір правильних ознак займає більше часу, ніж решта навчання.

Але бувають і зворотні ситуації, коли ми відбираємо серед усіх лише «правильні» на наш погляд ознаки та вносимо до моделі суб'єктивність — вона починає дико брехати.

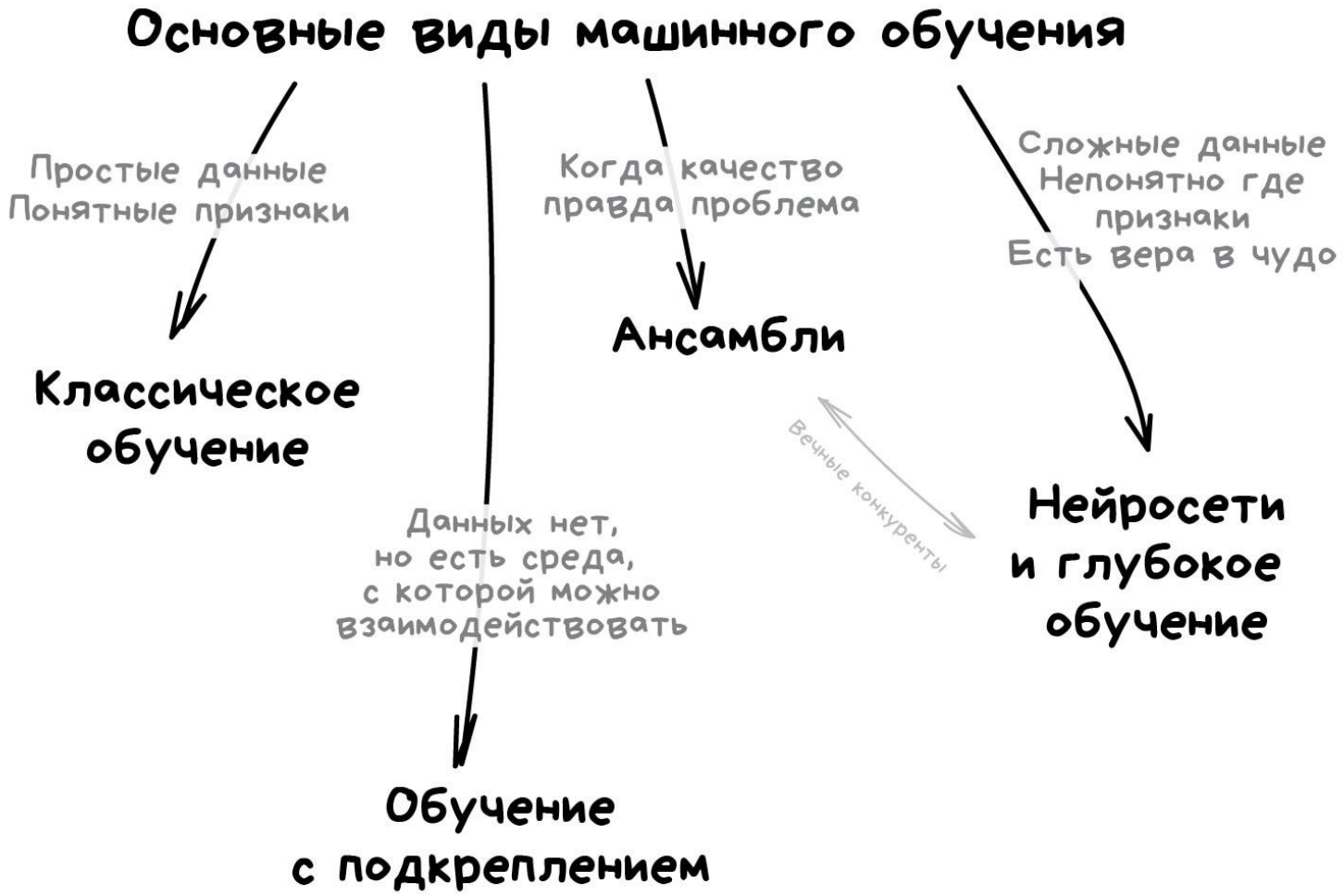
Алгоритми (методи)

Вирішувати завдання можна за допомогою різних методів. Усі методи діляться на чотири основних групи.

- Класичне навчання
  - З вчителем
    - Класифікація
    - Регресія
  - Без вчителя
    - Кластеризація
    - Зменшення розмірності
    - Пошук асоціативних правил
- Навчання з підкріпленням
- Ансамблеві методи
- Нейромережі
- Глибоке навчання


Машинное обучение
Классическое обучение
Без учителя
С учителем
Кластеризация
DBSCAN
K-Means
Agglomerative
Mean-Shift
Fuzzy C-Means
Классификация
Naive Bayes
K-NN
SVM
Decision Trees
Logistic Regression
Поиск правил
Euclat
Apriori
FP-Growth
Регрессия
Linear Regression
Polynomial Regression
Ridge/Lasso Regression
Уменьшение размерности (обобщение)
t-SNE
PCA
LSA
SVD
LDA
Обучение с подкреплением
Генетический Алгоритм
A3C
SARSA
Q-Learning
Deep Q-Network (DQN)
Ансамблевые методы
Стекинг
Беггинг
Random Forest
Бустинг
XGBoost
AdaBoost
LightGBM
CatBoost
Нейросети и глубокое обучение
Convolutional Neural Networks (CNN)
DCNN
Перцептроны (MLP)
Recurrent Neural Networks (RNN)
LSM
LSTM
GRU
Generative Adversarial Networks (GAN)
Autoencoders
seq2seq

Основні види машиного навчання



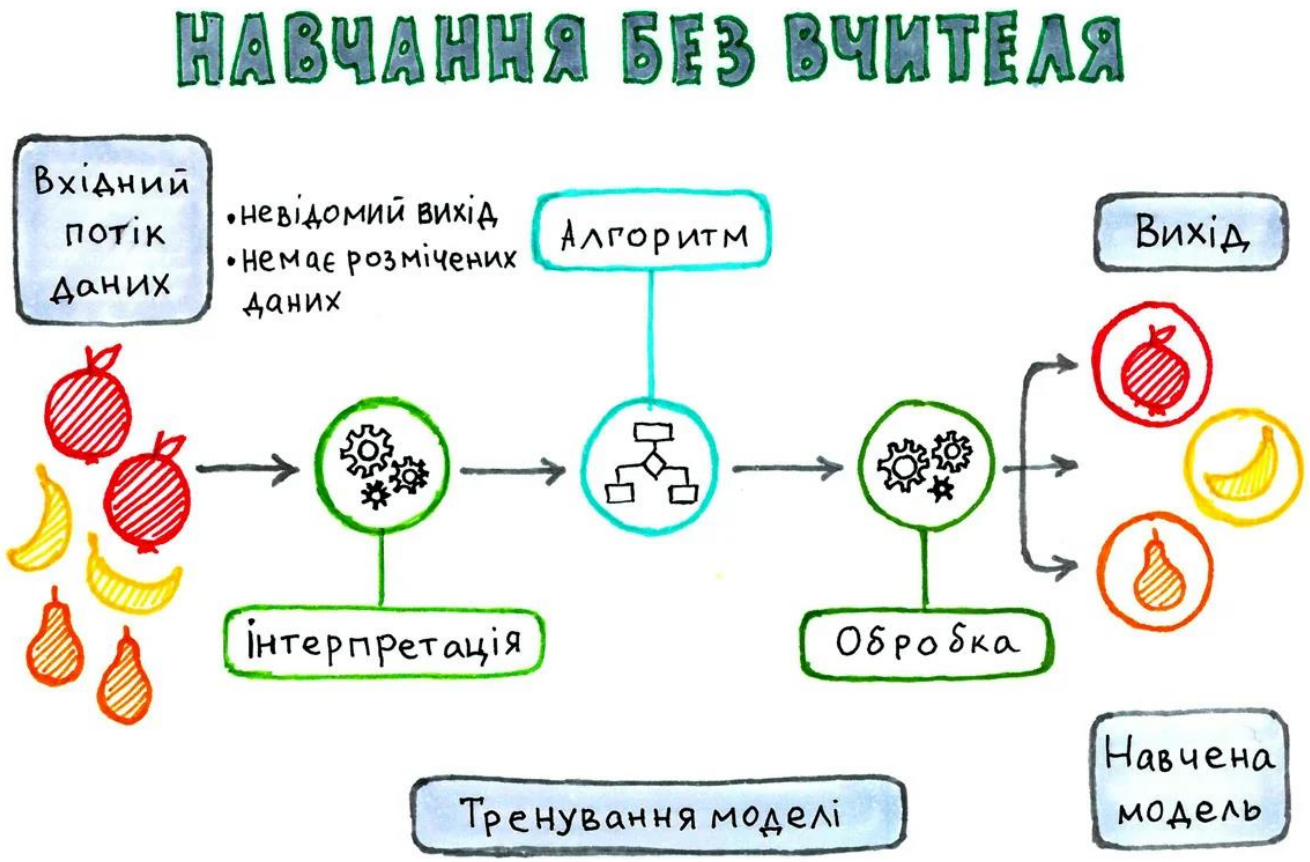
Навчання з учителем (Supervised Learning)

У цьому випадку машина має «наставника» – учителя, який говорить їй, як правильно вчинити. Вчитель заздалегідь відмічає всі потрібні дані, щоб машина навчалася на конкретних прикладах. Так він показує, що на цій світлині автомобіль, на іншій – велосипед.

Вчителем не завжди буває програміст, який стоїть над машиною і контролює кожну її дію. У термінах машинного навчання вчитель – це саме втручання людини в процес обробки інформації. З учителем машина вчиться набагато краще й швидше, тому для вирішення практичних завдань такі алгоритми використовують частіше. До алгоритмів навчання з учителем належать такі типи задач, як регресія і класифікація.

Навчання без вчителя (Unsupervised Learning)

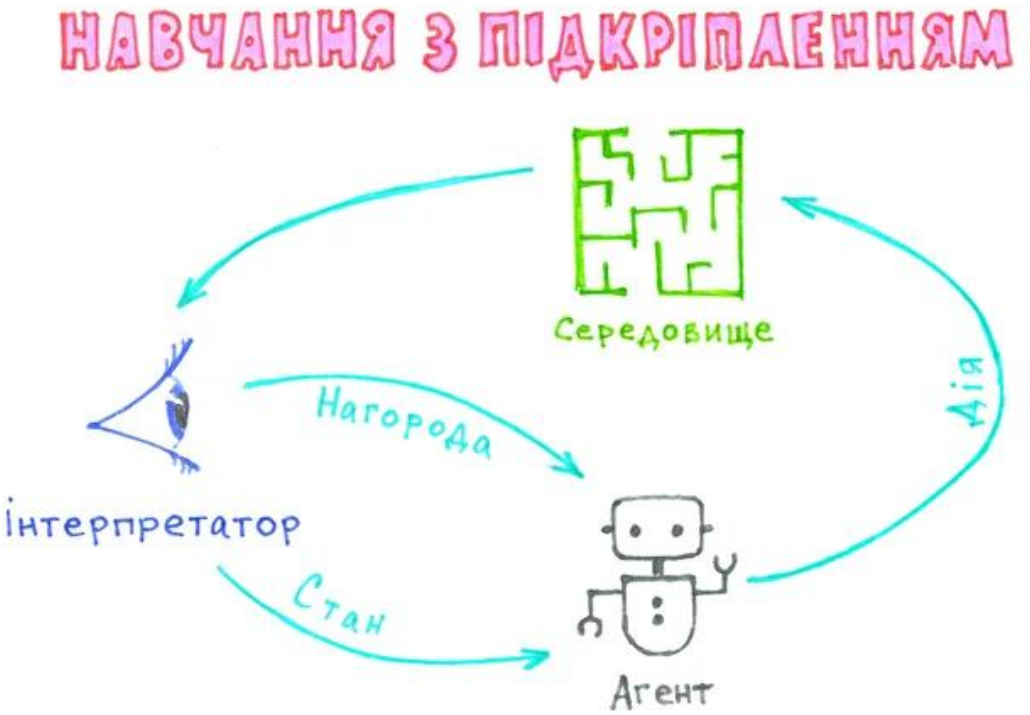
У навчанні без учителя машині просто вивалюють купу фотографій на стіл і кажуть: «розберися, що тут до чого». Дані в такому разі не розмічені, і машина сама намагається знайти закономірності.



На практиці такі алгоритми використовують рідше, зазвичай як методи аналізу та підготовки даних, а не як основний алгоритм, що вирішує конкретні завдання за допомогою цих даних. У випадках, коли розмітка даних неможлива, вдаються до методів навчання без учителя

Навчання з підкріпленням (Reinforcement Learning)

Навчання з підкріпленням менше схоже на попередні види, бо нагадує швидше той штучний інтелект, яким нас намагаються вразити у фантастичних фільмах. Такі алгоритми використовують не там, де потрібно проаналізувати дані, а там, де потрібно вижити в реальному середовищі.



Середовищем може бути що завгодно: як реальний світ, так і симуляція, і навіть комп'ютерна гра. Наприклад, існують роботи, які навчилися грати в Dota2 просто поринувши в середовище, або автопілот Tesla, який у симуляції вчиться не збивати пішоходів.

Навчання з підкріпленням (Reinforcement Learning)

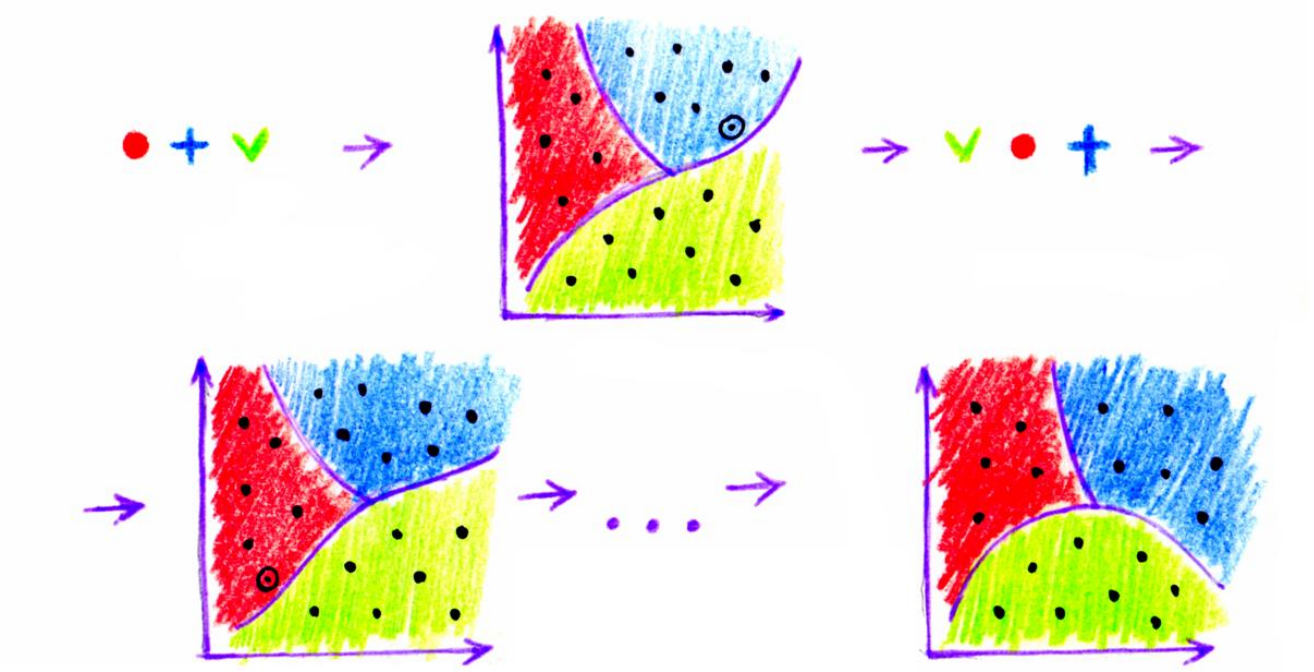
Знання про довкілля таким роботам корисні, але знати весь світ їм не обов'язково. Завдання таких машин – не розрахувати всі ходи, а мінімізувати помилки або максимізувати вигоду. Навчання з підкріпленням дуже схоже на реальне навчання людей – машину карають за помилки і заохочують за правильні вчинки.

Пам'ятаєте алгоритм AlphaGo, створений кілька років тому, який обіграв найкращих гравців в світі в го? Число комбінацій у грі перевищує кількість атомів у Всесвіті, тому всі комбінації машині запам'ятати неможливо. AlphaGo просто обирала найкращий вихід із ситуації, що склалася, і зробила це краще за людину.

Успіхи машин у комп'ютерних іграх, спорті або технології автопілоту виглядають досить ефектно. Та насправді алгоритмів навчання з підкріпленням не так багато, цей напрям лише починає розвиватися. Саме тому за ними, безсумнівно, майбутнє.

Ансамблеві методи

Основна ідея – використання паралельно чи послідовно декількох стандартних методів навчання для отримання більш якісного результату роботи алгоритму



Нейронні мережі

Постановка задачі навчання

- Розділити дані на дві частини:
  - навчальна вибірка (більша частина)
  - тестувальна вибірка
- Навчити машину за існуючою базою даних (навчальною вибіркою) приймати необхідне рішення – побудувати алгоритм прийняття рішень
- Перевірити побудований алгоритм на тестувальній вибірці

Приклад постановки задачі

Необхідно зробити прогноз чи зіграє гравець у теніс при певних погодних умовах?

Дані (ознаковий опис):
Існують дані (N записів) щодо попередніх ігор гравця при певних погодних умовах.
Ознаки:

1. Погодні умови (сонячно, похмуро, дощ)
2. Вологість (нормальна, висока)
3. Вітер (слабкий, сильний)

Результат – відбулась чи не відбулась гра

Задача - навчити машину за існуючою базою даних (навчальною вибіркою) приймати необхідне рішення – побудувати алгоритм прийняття рішень

Приклад постановки задачі

Чи зіграє гравець у теніс
при певних погодних умовах?

Дані: 14 записів (об'єктів)
з трьома признаками

№	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 постановки задачі

Чи зіграє гравець у теніс
при певних погодних умовах?

Дані: 14 записів (об'єктів)
з трьома признаками

1. Розділяємо всі дані
на навчальну
та тестувальну вибірки
2. Обираємо алгоритм навчання
(метод) та навчаємо машину
на навчальній вибірці
3. Перевіряємо алгоритм на
тестувальній вибірці –
оптимізуємо параметри
алгоритму
4. Прогнозуємо результат для
нових вхідних даних

№	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	???

Дякую за увагу