Світ штучного інтелекту

В цій темі

- Постановка задачі в машинному навчанні і приклад
- Які методи існують в машинному навчанні
- Які задачі вирішують з машинним навчанням
- Як пов’язані Data Science/ Artificial Intelligence/ Machine Learning
- Процес розв’язку Data Science задачі
- Інструменти, необхідні для роботи з DS задачею

Постановка задачі в машинному навчанні

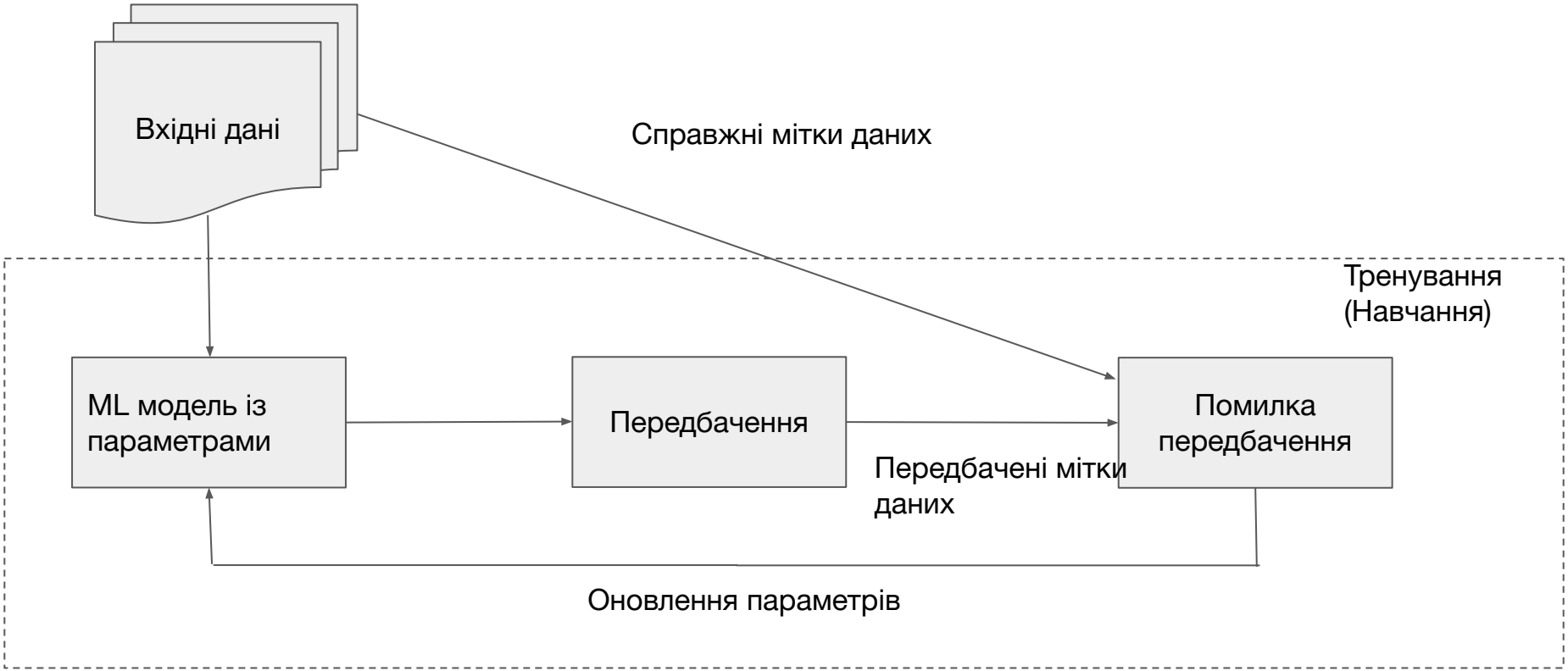
Традиційне програмування

Машинне навчання

Традиційне програмування

Машинне навчання

Що таке “навчання”



Формальне формулювання задачі машинного навчання (з вчителем)

X — множина об'єктів (вхідні дані)

Y — множина відповідей

f : X → Y — невідома залежність (target function)

Дано:

{x1, . . . , xℓ} ⊂ X — навчальна вибірка (training sample)

yi = y(xi), i = 1, . . . , ℓ — відомі відповіді

Знайти:

h : X → Y — алгоритм, вирішальну функцію (decision function), що наближає **y** на всій множині **X**.

Курс машинного навчання - це конкретизація:

- як задаються об'єкти і якими можуть бути відповіді
- як будується функція **h**
- що означає "**h** наближає y на всьому **X**" - метрики оцінки якості алгоритмів

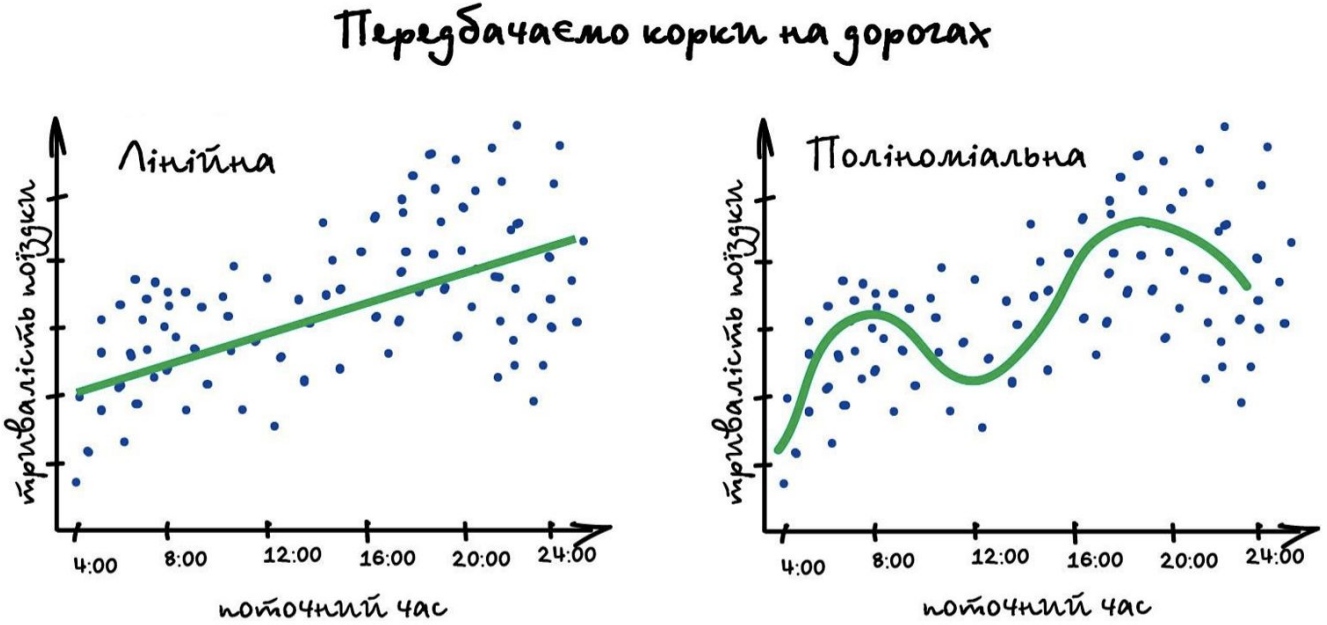
Приклад задачі машинного навчання з вчителем

Припустимо, ми хочемо спрогнозувати ціну будинку (Housing Price) за такими показниками як Floor area, School Disctict, Orientation.

За допомогою моделі ми можемо знайти прогноз (оцінку цільового значення).

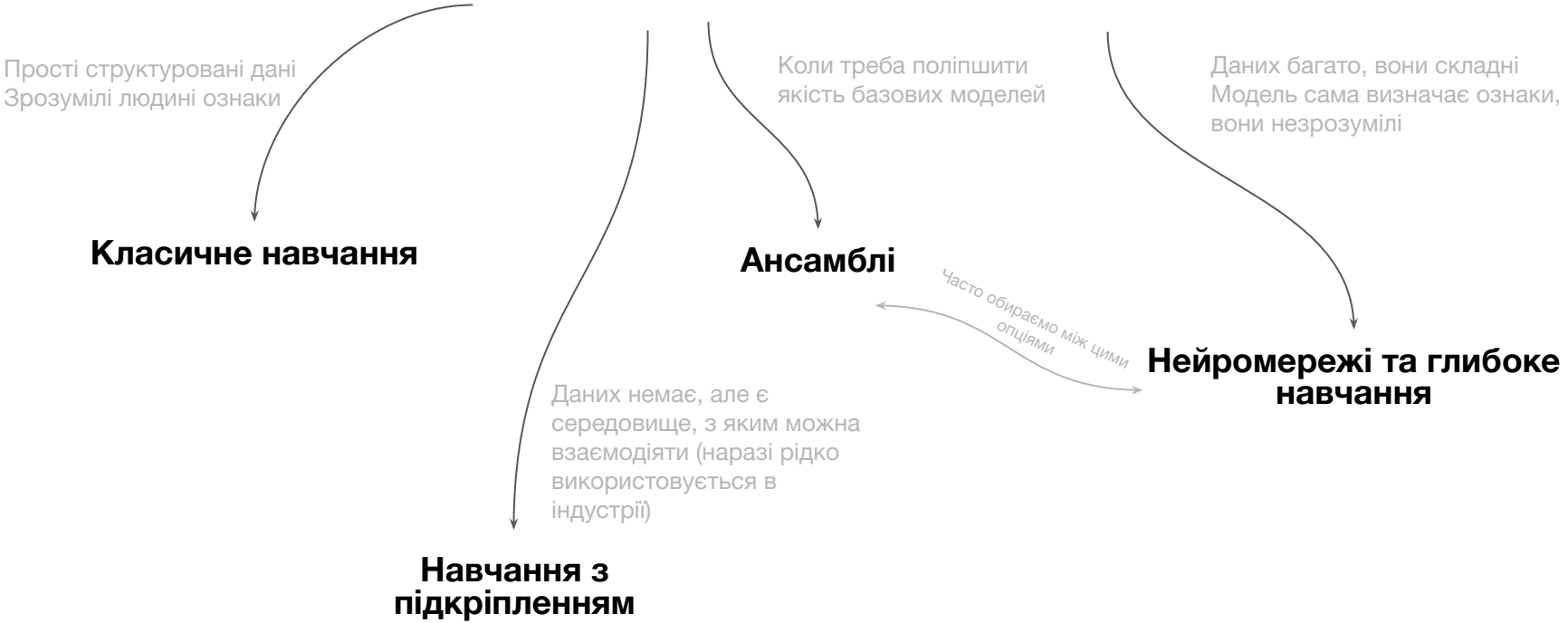
		Feature 1	Feature 2	Feature 3	Label
	Serial number	Floor area	School district	Orientation	Housing price
Training set	1	100	8	South	1000
	2	120	9	Southwest	1300
	3	60	6	North	700
	4	80	9	Southeast	1100
Test set	5	95	3	South	850

При цьому модель — це завжди певне наближення! Адже ми не знаємо точну формулу.



Навігація у світі методів машинного навчання

Основні методи машинного навчання



Класичне навчання

Дані заздалегідь категоризовані чи чисельні

Дані ніяк не розмічені

З вчителем

Передбачити категорію

Передбачити значення

Класифікація

“Розклади шкарпетки за кольором”

Регресія

“Розклади краватки за довжиною”

Без вчителя

Розділити за схожістю

Знайти залежності

Виявити послідовності

Кластеризація

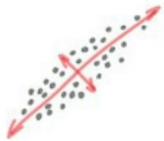
“Розклади схожі речі за купками”

Асоціація

“Знайди які речі я ношу разом”

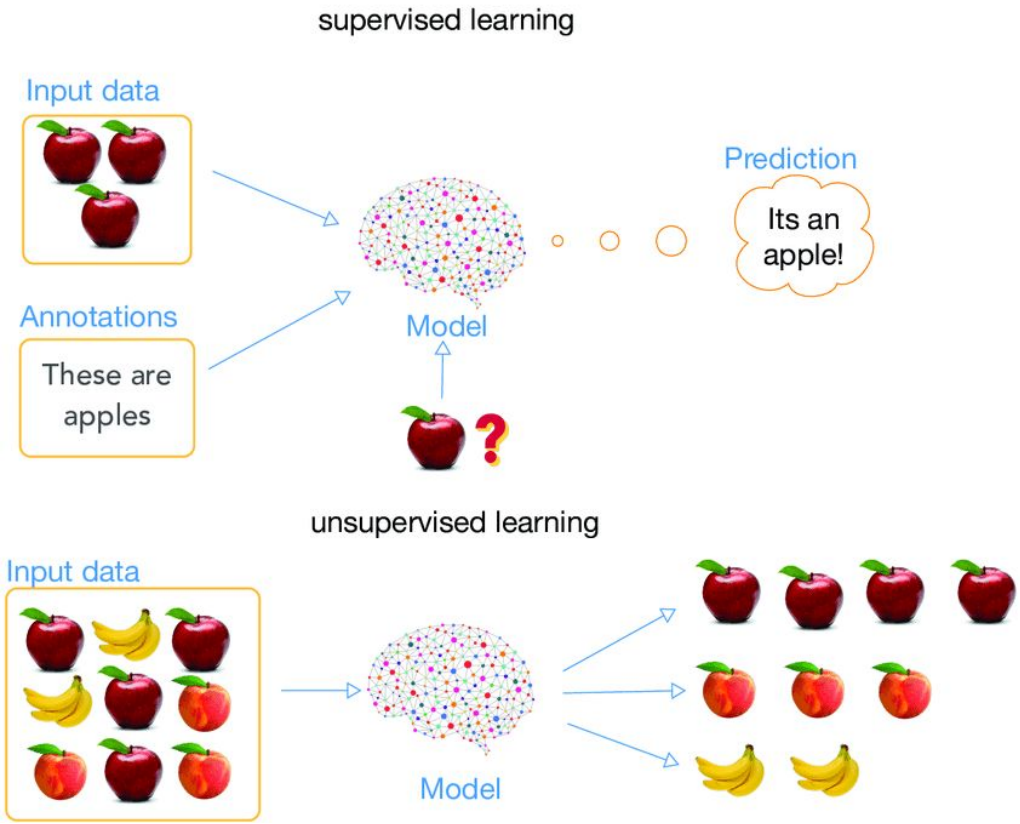
Зменшення розмірності (узагальнення)

“Збери з речей найкращі вбрання”



Більше: [тут]

Навчання з вчителем і без



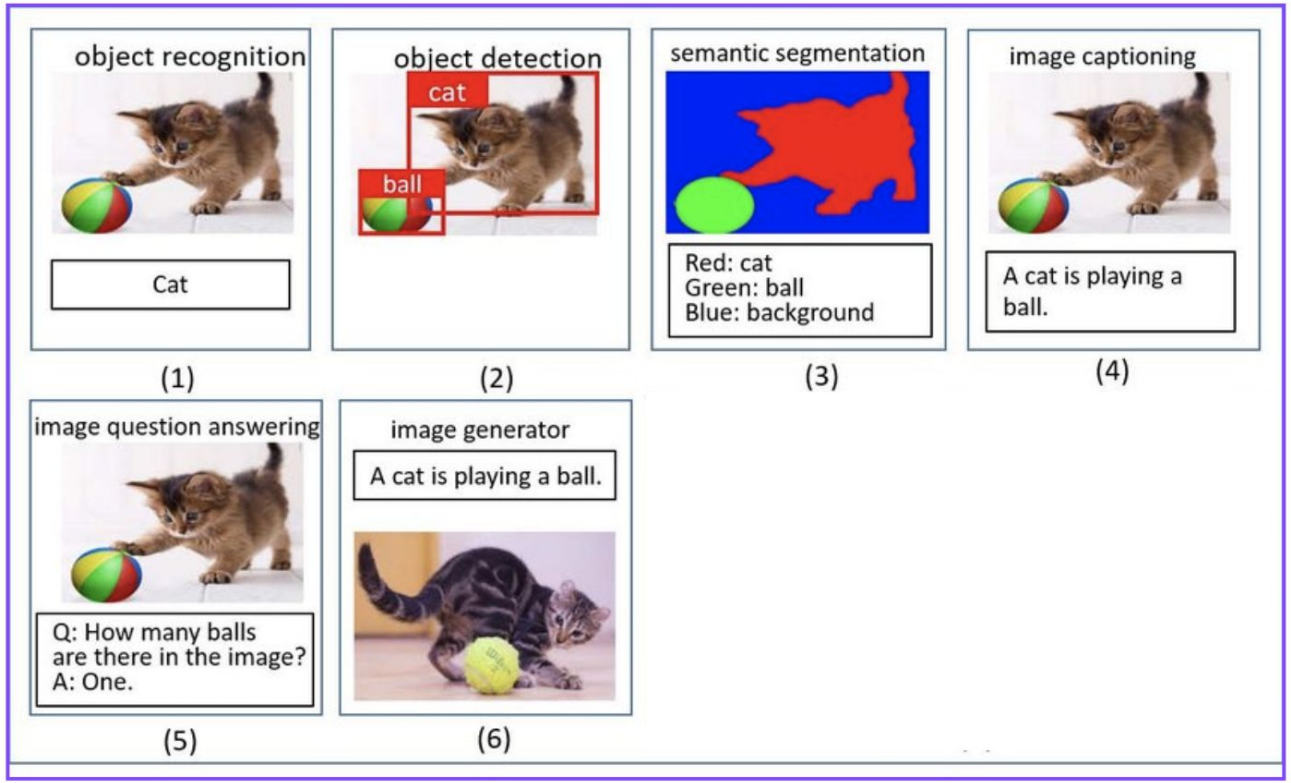
Які задачі вирішують з машинним навчанням

“Класичний” Data Science

- Прогнозування попиту та управління запасами
- Клієнтська аналітика
- Створення персоналізованих рекомендацій (Netflix, Amazon)
- Виявлення шахрайства
- Оптимізація логістичних операцій
- Автоматичний скейлинг потужностей для обслуговування більшої кількості користувачів

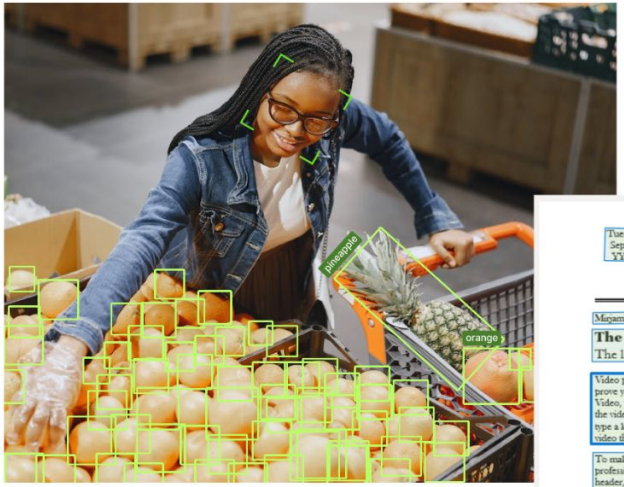
і багато інших задач!

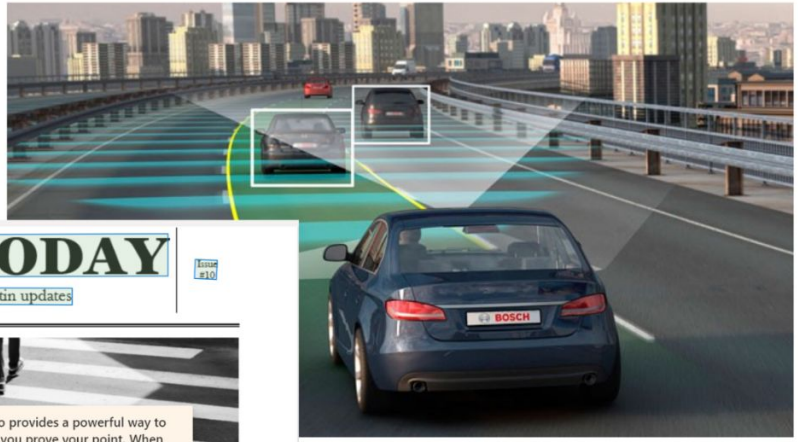
Комп’ютерний зір (Computer vision)



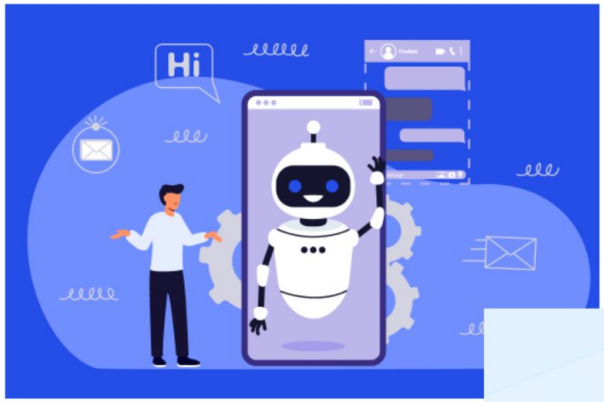
Комп’ютерний зір (Computer vision)

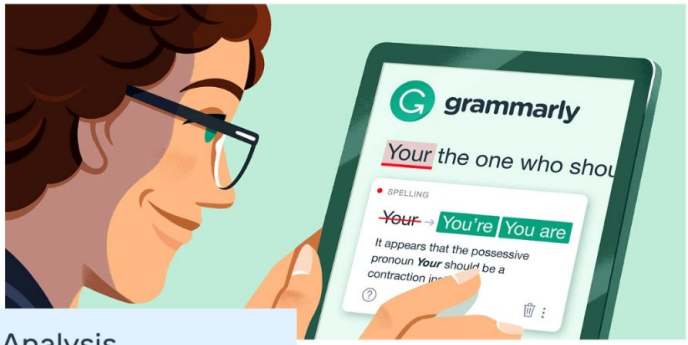
Приклади використання в бізнесі.

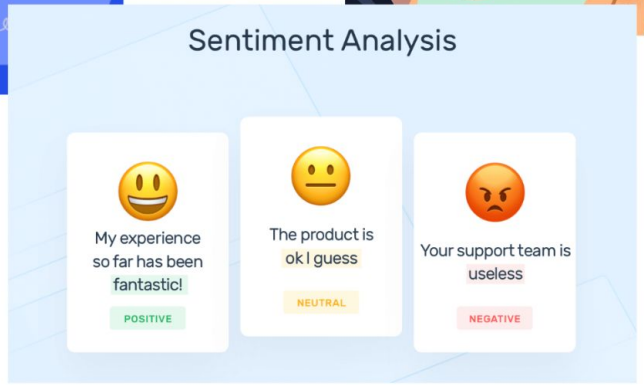




Обробка природньої мови (NLP)



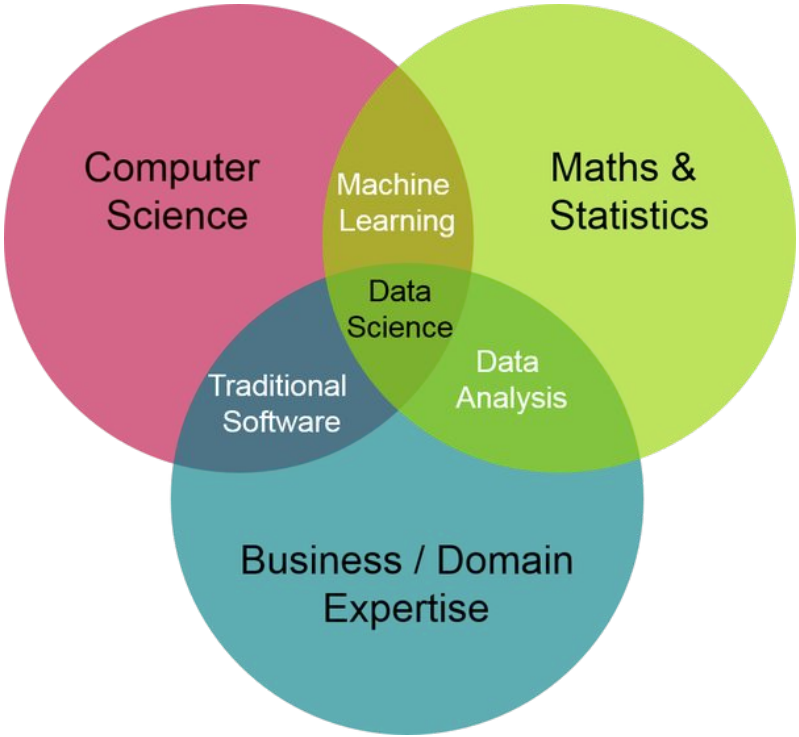




Використань Data Science та ШІ набагато, набагато більше!

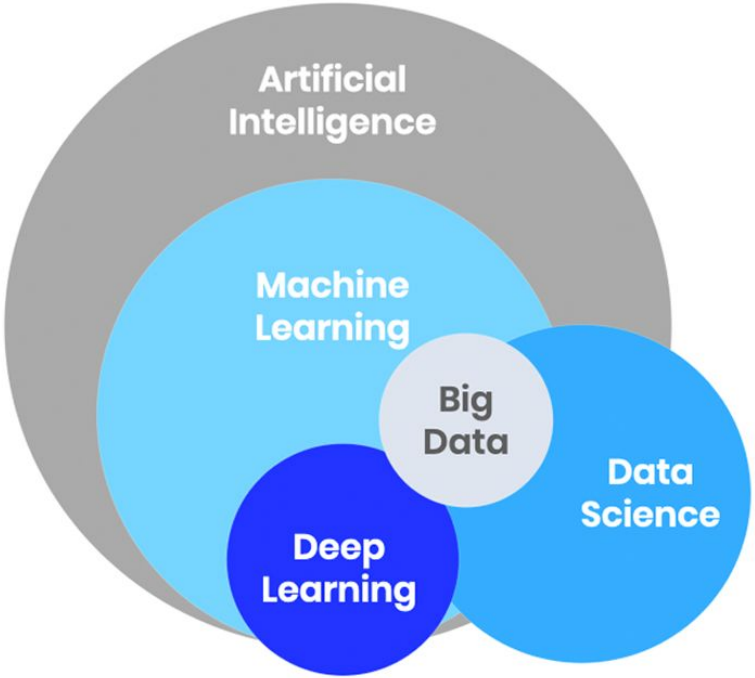
Стоп, а що до чого тут Data Science та ШІ? Ми ж про машинне навчання говорили 🤔

Де знаходиться Data Science?



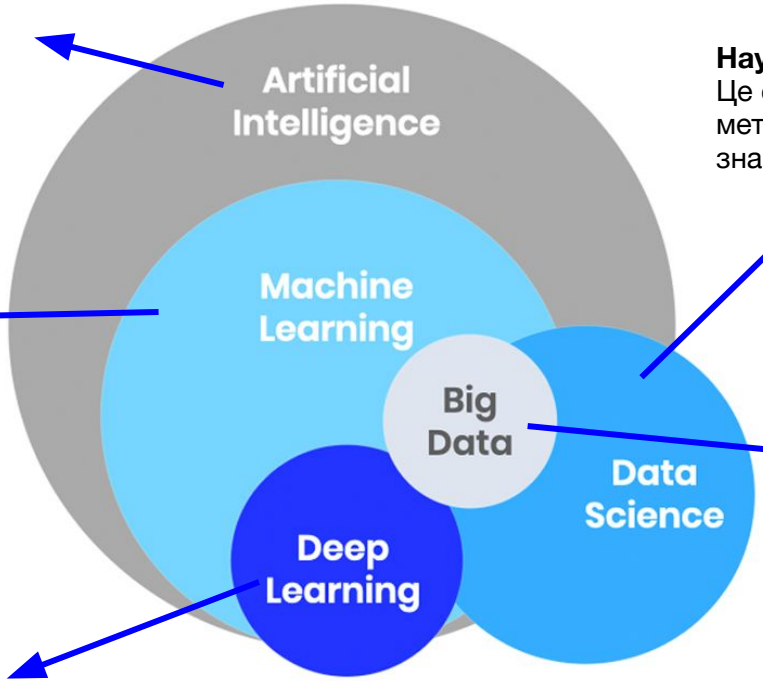
А де ж ШІ?

Організація виглядає наступним чином



Організація виглядає наступним чином

Штучний інтелект — означає створення розумних машин для імітації людської поведінки



Наука про дані є підмножиною AI. Це сфера статистики, наукових методів тощо, щоб витягнути значення та зрозуміти дані.

Машинне навчання є підмножиною AI й створює модель на основі навчальних даних, щоб робити прогнози

Великі дані — означає набір методів і інфраструктуру для роботи з настільки великими даними, що їх важко, довго та дорого обробляти іншими способами

Глибоке навчання — підмножина ML, клас алгоритмів ML для розв'язання складних проблем.

Тому ми почнемо обробляти дані на першому ж тижні 💪

Перед цим ще оглянемо загальний процес Data Science проєкту.

В цьому уроці

- Процес розв’язку Data Science задачі
- Роль роботи з даними при розв’язку Data Science задачі
- Інструменти, необхідні для роботи з DS задачею

Життєвий цикл розв'язку DS задачі: CRISP-DM

1. **Розуміння бізнесу:** Важливе питання полягає в тому, чи потрібен нам ML для проєкту. Мета проєкт має бути вимірною.

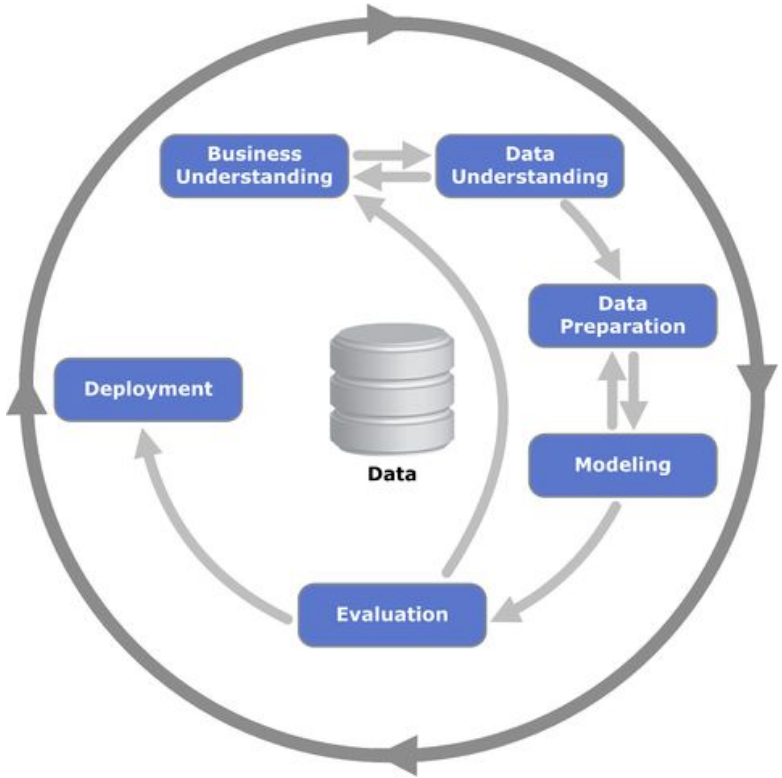
2. **Розуміння даних:** Тут ми аналізуємо доступні джерела даних і вирішуємо, чи потрібні додаткові дані.

3. **Підготовка даних:** Очищуємо дані, видаляємо дублікати, викиди, заповнюємо пропущені значення, можливо також автоматизуємо обробку даних. Також, дані повинні бути перетворені в структурований числовий формат, щоб ми могли передати їх у ML-модель.

4. **Моделювання:** навчання моделей. Різні моделі та виберіть найкращу. Враховуючи результати цього кроку, правильно вирішити, чи потрібно додати нові функції, чи виправити проблеми з даними.

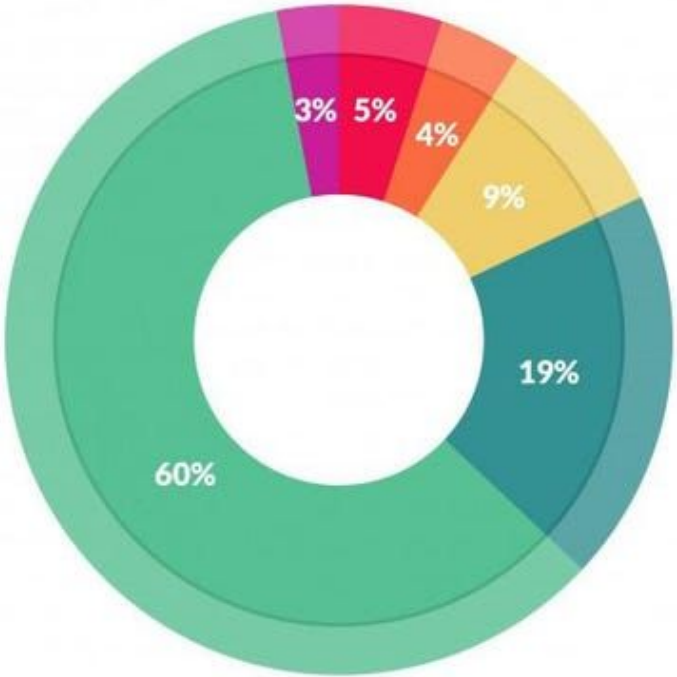
5. **Оцінка:** Виміряйте, наскільки добре працює модель і чи вирішує вона бізнес-проблему.

6. **Розгортання:** Розгортання робочої версії для всіх користувачів. Оцінка та розгортання часто відбуваються разом – в такому випадку ми говоримо про онлайн-оцінку.



https://en.wikipedia.org/wiki/Cross-industry_standard_process_for_data_mining

Чи важлива робота з даними?

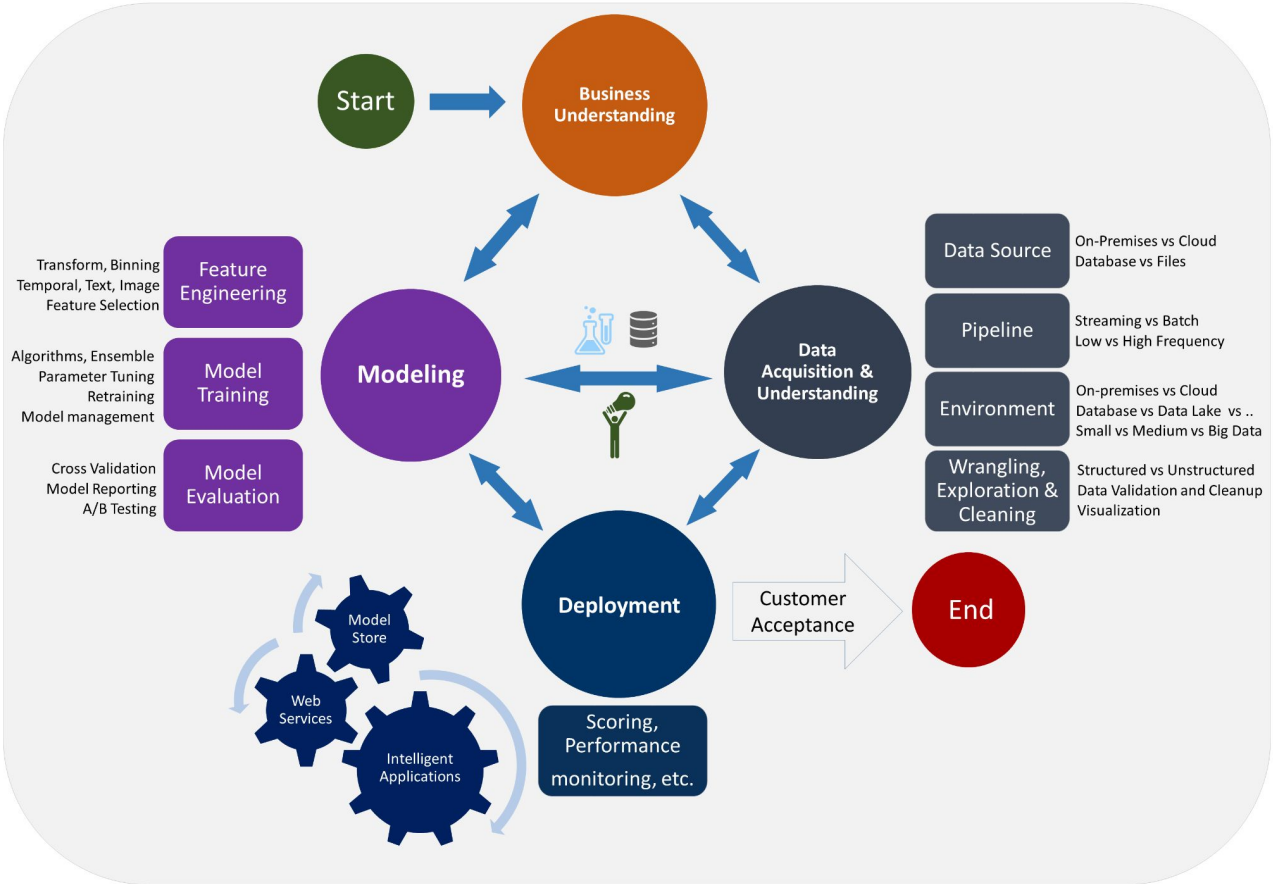


What data scientists spend the most time doing

- *Building training sets: 3%*
- *Cleaning and organizing data: 60%*
- *Collecting data sets; 19%*
- *Mining data for patterns: 9%*
- *Refining algorithms: 4%*
- *Other: 5%*

Cleaning Big Data: Most Time-Consuming, Least Enjoyable Data Science Task, Survey Says

Data Science Lifecycle



https://learn.microsoft.com/en-us/azure/architecture/data-science-process/lifecycle-business-understanding

Інструменти, необхідні для роботи з DS задчею

- **Pandas:** робота з табличними даними
- **Numpy:** робота з матрицями та векторами, математичні обчислення
- **Matplotlib/Seaborn/Plotly:** візуалізація даних
- **Scikit-learn, scipy:** побудова ML моделей
- **Statsmodels:** статистичні моделі, тестування стат. гіпотез, дослідження даних
- **Keras/PyTorch:** побудова моделей глибокого навчання
- **Huggingface:** тренування та використання готових моделей для NLP, CV задач
- **Streamlit:** деплоймент і простий UI для презентації роботи моделі
- **FastAPI/Flask:** створення RestAPI
- **Docker:** контейнеризація (для деплойменту)
- **openai/langchain/llamaindex:** робота з genAI моделями
- Допоміжні інструменти, що прискорюють роботу на різних етапах вирішення DS завдання: **sweetwiz, pandas-profiling, missingno**

Дослідницький аналіз даних

В цій темі

- Роль дослідницького аналізу даних в Data Science задачі
- Завдання дослідницького аналізу даних
- Приклад покрокового дослідницького аналізу

Дослідницький аналіз даних (Exploratory Data Analysis, EDA)

Дослідницький аналіз даних (EDA) у контексті машинного навчання - це перший крок у процесі аналізу та підготовки даних, який передує побудові моделей машинного навчання.

Цей процес включає **збір**, **очищення** та **візуалізацію** даних для виявлення *закономірностей*, *аномалій* і *взаємозв'язків* між змінними, що можуть вплинути на вибір алгоритмів машинного навчання та їх налаштування.

Завдання в рамках EDA

В рамках дослідницького аналізу даних аналітики і науковці в галузі даних виконують такі завдання:

- **Очищення даних**: знаходження та виправлення помилок у даних, усунення аномалій та викидів, заповнення або видалення відсутніх значень.

- **Візуалізація даних**: використання графіків і діаграм для виявлення тенденцій, закономірностей та взаємозв'язків між змінними.

- **Аналіз розподілів та кореляцій:** вивчення як розподіляються окремі змінні та як вони взаємодіють одна з одною.

- **Попередня обробка даних:** нормалізація або стандартизація даних, перетворення змінних для підвищення ефективності алгоритмів машинного навчання.

- **Генерація нових ознак:** створення додаткових змінних (фіч), що можуть покращити якість моделей.

Завдяки дослідницькому аналізу даних, ми отримуємо важливі інсайти про дані, з якими працюємо, що дозволяє їм краще зрозуміти проблему, яку вони намагаються вирішити, і вибрати методи та алгоритми, які найкраще пасують для цього.

Ми наразі проведемо максимально простий EDA, а з часом навчимось більш витонченим методам 👌

Numpy

Numpy = Numeric Python

пакет для наукових обчислень мовою Python.

Надає:

+ об'єкт багатовимірного масиву та різні похідні об'єкти (масиви з масками та матриці);
+ процедури для швидких операцій над масивами, включно з математичними, логічними, маніпуляціями з формами, сортуванням, вибіркою, введенням/виведенням, дискретними перетвореннями Фур'є, базовою лінійною алгеброю, базовими статистичними операціями, методами генерації випадкових значень із заданих розподілів і багатьом іншим.

Застосування:

+ будь-які операції над матрицями (фільтрація сигналу, обробка зображень). Numpy масиви - стандарт у Python для роботи з багатовимірними масивами.
+ написання алгоритмів ML з нуля, зокрема нейронних мереж.

https://runebook.dev/ru/docs/numpy/user/whatisnumpy

Ndarray – основа Numpy

Numpy ndarray	**Python list**
+ Мають фіксований розмір під час створення. Зміна розміру `ndarray` створить новий масив і видалить оригінал	+ Можуть зростати динамічно
+ Елементи масиву NumPy повинні бути одного типу даних і, отже, мати однаковий розмір у пам'яті	+ Елементи можуть бути різних типів даних
+ Спрощують складні математичні та інші типи операцій з великою кількістю даних	+ На великій кількості даних можуть призводити до тривалого часу обчислень

Чому Numpy швидкий?

В основі більшої частини можливостей Numpy: векторизація і трансляція.

Демо швидкості Numpy, векторизації, трансляції:

Переходимо до практики — [Lecture 1. Numpy Basics.ipynb]

Векторизація

Векторизація — відсутність циклів, індексації. У коді ці речі відбуваються неявно, в оптимізованому, попередньо скомпільованому коді на C.

Переваги векторизованого коду:

- більш лаконічний і легше читається
- менше рядків коду зазвичай означає менше помилок
- код більш схожий на стандартну математичну нотацію (що спрощує правильне кодування математичних конструкцій)
- векторизація призводить до більш "пітонічного" коду. Без векторизації наш код був би завалений неефективними і важкими для читання циклами for.

Трансляція (broadcasting)

Трансляція — це термін, що використовується для опису неявної поелементної поведінки операцій. У NumPy всі операції (не тільки арифметичні, а й логічні, побітові, функціональні тощо) поводяться таким неявним поелементним чином, тобто транслюються.

Демо

Демо роботи з масивами: [Lecture 1. Numpy Basics.ipynb]

Запитання

Рекомендовані матеріали

1. Про Jupyter-ноутбуки: https://thecode.media/jupyter/
2. Вступ до Numpy https: //habr.com/ru/post/352678/
3. Застосування Numpy для обробки зображень: https: //habr.com/ru/post/469355/
4. Простими словами про Машинне навчання https: //vas3k.ru/blog/machine_learning/
5. Математичні функції numpy https: //docs.scipy.org/doc/numpy/reference/routines.math.html