**Модуль 1.1: Знайомство з Data Science**

**Вступ до Data Science**

Вітаємо усіх на першому занятті курсу по Data Science!

Сьогодні ми дізнаємося про основні поняття та інструменти, які використовуються у сфері науково-дослідницького аналізу даних. Ми розглянемо всі необхідні кроки, починаючи від підготовки даних та закінчуючи викладенням результатів у зрозумілій формі.

Data Science - це галузь, що займається збором, обробкою, аналізом, та інтерпретацією даних, з метою здобуття нових знань та розуміння явищ в певній сфері. Вона поєднує методології, алгоритми та інструменти з різних дисциплін, включаючи статистику, математику, інформатику та інженерію.

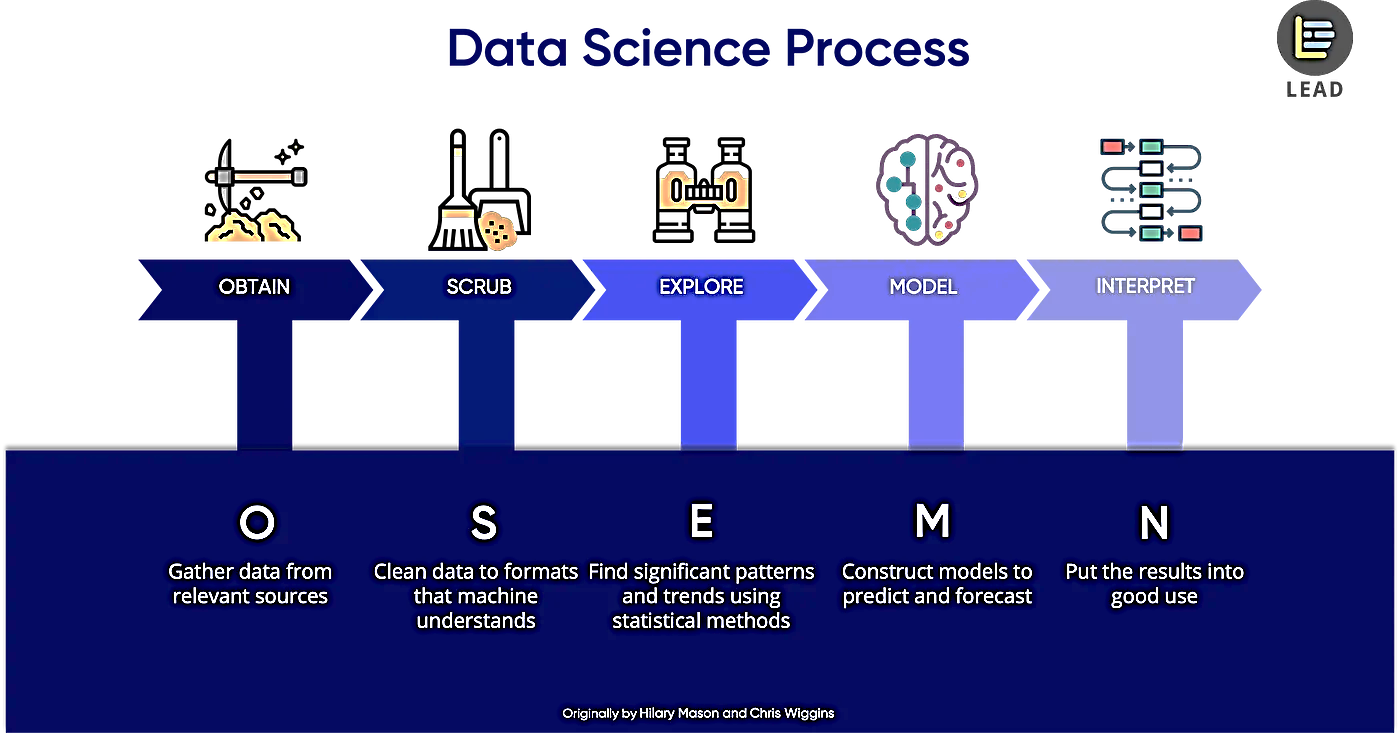
Data Science є ключовим елементом в сучасному світі, де дані грають важливу роль у прийнятті рішень, плануванні, та стратегічному розвитку різних галузей. Із розвитком технологій та зростанням об'ємів даних, значення Data Science тільки збільшується, тому вона стає все більш популярною серед фахівців різних напрямків.

У курсі ми розглянемо основні інструменти та методи Data Science, а також їх застосування в реальних проектах. Ви дізнаєтеся про роботу з даними, візуалізацію, машинне навчання, глибоке навчання, рекомендаційні системи, та інші теми, що допоможуть вам стати ефективними Data Scientist'ами.

Метою цього першого заняття є ознайомлення з основними концепціями Data Science, а також інструментами, які ви будете використовувати у курсі. Давайте почнемо!

**Життєвий цикл проекту Data Science**[​](https://textbook.edu.goit.global/python/data-science-remaster/v1/docs/module-01/main" \l "%D0%B6%D0%B8%D1%82%D1%82%D1%94%D0%B2%D0%B8%D0%B9-%D1%86%D0%B8%D0%BA%D0%BB-%D0%BF%D1%80%D0%BE%D0%B5%D0%BA%D1%82%D1%83-data-science" \t "_blank)

Життєвий цикл проекту Data Science - це важлива тема для розуміння того, як правильно підходити до роботи з даними. Без належного розуміння цього процесу можна втратити багато часу та ресурсів, що не завжди буде ефективним. У цьому блоку ми розглянемо 5 кроків життєвого циклу проекту Data Science, що допоможе вам розуміти, як ефективно працювати з даними та досягати успіху в цій області.



<https://towardsdatascience.com/5-steps-of-a-data-science-project-lifecycle-26c50372b492>

**Збір даних**

[​](https://textbook.edu.goit.global/python/data-science-remaster/v1/docs/module-01/main#%D0%B7%D0%B1%D1%96%D1%80-%D0%B4%D0%B0%D0%BD%D0%B8%D1%85)Перший етап життєвого циклу проекту Data Science - збір даних. Цей етап є важливим, оскільки точність і достовірність будь-якого аналізу даних залежить від якості вхідних даних. Зазвичай дані збираються з різних джерел, таких як бази даних, датчики, API тощо. Залежно від проекту, дані можуть бути у форматі структурованого або неструктурованого тексту, зображень, відео або звуку.

Наприклад, якщо ми хочемо побудувати модель передбачення ціни нерухомості на підставі її характеристик, то ми можемо зібрати дані з баз даних нерухомості, від агентів нерухомості, а також з соціальних мереж, де користувачі діляться інформацією про купівлю та продаж нерухомості. Дані можуть бути в різних форматах, таких як CSV, JSON або XML, і їх можна зібрати за допомогою різних інструментів та бібліотек, наприклад, [Python](https://www.python.org/) (<https://www.python.org/>) бібліотеки [pandas](https://pandas.pydata.org/) (<https://pandas.pydata.org/>) або [BeautifulSoup](https://pypi.org/project/beautifulsoup4/) (<https://pypi.org/project/beautifulsoup4/>) для збору даних з веб-сторінок.

**Очищення та підготовка даних**

[​](https://textbook.edu.goit.global/python/data-science-remaster/v1/docs/module-01/main#%D0%BE%D1%87%D0%B8%D1%89%D0%B5%D0%BD%D0%BD%D1%8F-%D1%82%D0%B0-%D0%BF%D1%96%D0%B4%D0%B3%D0%BE%D1%82%D0%BE%D0%B2%D0%BA%D0%B0-%D0%B4%D0%B0%D0%BD%D0%B8%D1%85)Етап очищення даних є одним з найважливіших етапів у життєвому циклі проекту Data Science. Незалежно від того, наскільки у нас хороша модель, більшість залежить від нашого датасету, і є хороший вислів: сміття на вхід - сміття на вихід. Очищення даних означає видалення або коригування неточностей, помилок, дублікатів, відсутності значень та інших некоректних даних. На цьому етапі можна проводити різні операції, наприклад:

* Видалення дублікатів: у великих наборах даних можуть бути дублікати записів, що може призвести до перекручення результатів. Наприклад у таблиці можуть бути стовпці User та Name, проте у кожному з них буде зберігатись одна й та ж інформація - ім'я користувача. Видалення дублікатів забезпечує точність даних і зменшує їх об'єм.
* Видалення відсутніх значень: відсутні значення в даних можуть виникати з різних причин, включаючи помилки вводу, відмову учасників дослідження відповідати на певні питання або недоступність даних. Відсутні значення можуть бути замінені середніми значеннями або іншими вирахованими значеннями.
* Кодування категоріальних даних: багато даних містять категоріальні дані, які потрібно закодувати в числові значення для подальшої обробки. Це можна зробити за допомогою різних методів, таких як метод one-hot encoding або label encoding.
* Нормалізація даних: дані можуть містити значення, які відрізняються за порядком величин, наприклад, вік та дохід. Для того, щоб забезпечити рівні ваги для всіх факторів, дані можуть бути нормалізовані за допомогою різних методів.

Даний перелік не є вичерпним, проте описує основні проблеми, з якими ви можете зіткнутись.

**Дослідження даних**

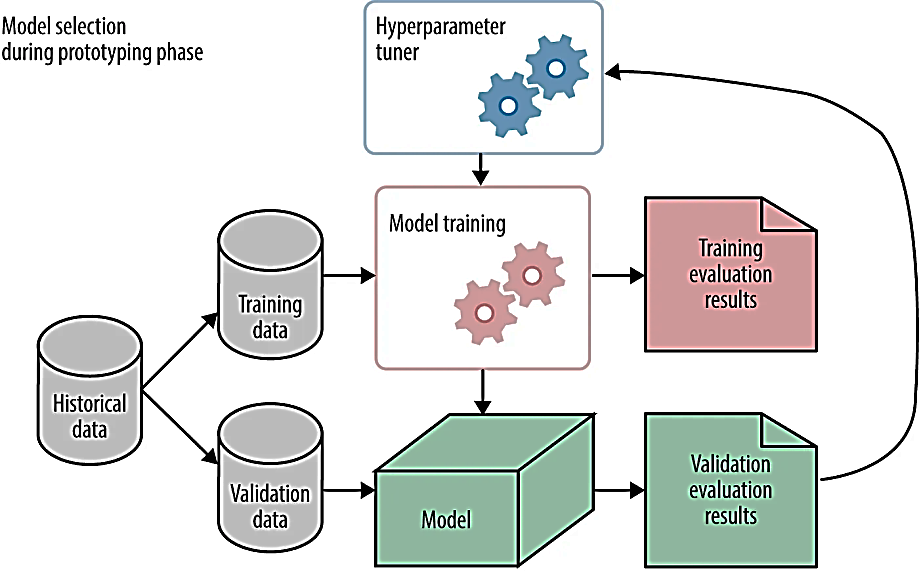
[​](https://textbook.edu.goit.global/python/data-science-remaster/v1/docs/module-01/main#%D0%B4%D0%BE%D1%81%D0%BB%D1%96%D0%B4%D0%B6%D0%B5%D0%BD%D0%BD%D1%8F-%D0%B4%D0%B0%D0%BD%D0%B8%D1%85)EDA (Exploratory Data Analysis) - це перший крок у будь-якому проекті зі статистичним моделюванням та машинним навчанням. Його метою є дослідження даних, що входять до дослідження, їхнього структуровання, залежностей та можливих проблем.

Першим кроком при EDA є збір інформації про дані. Наскільки вони великі, які вони мають типи даних, чи присутні в них пропущені значення, яка розподіленість даних і т.д. Ця інформація допомагає зрозуміти, які можливості наявні у ваших даних і як їх краще аналізувати.

Сюди входить також візуалізація даних, яка може допомогти виявити кореляції між змінними та можливі виклики, пов'язані з даними. Найпоширенішими інструментами для візуалізації є бібліотеки matplotlib та seaborn, за допомогою яких ми можемо будувати різні лінійні графіки, діаграми, гістограми та інші.

**Створення моделі**[​](https://textbook.edu.goit.global/python/data-science-remaster/v1/docs/module-01/main#%D1%81%D1%82%D0%B2%D0%BE%D1%80%D0%B5%D0%BD%D0%BD%D1%8F-%D0%BC%D0%BE%D0%B4%D0%B5%D0%BB%D1%96)

Це мабуть найцікавіша, проте й найскладніша частина всієї роботи Data Scient'иста. Саме тут відбувається магія і ми лише завдяки наших даних можемо навчитись розрізняти спам, передбачати ціну на будинок чи розблоковувати телефон дотиком чи своїм лицем.



<https://towardsdatascience.com/5-steps-of-a-data-science-project-lifecycle-26c50372b492>

Ще раз, перш, ніж перейти до цього етапу, майте на увазі, що етап очищення та дослідження даних не менш важливий для створення корисних моделей. Тому не поспішайте із застосуванням моделювання, а зверніть увагу на ці етапи.

У моделюванні є кілька завдань. Ми також можемо навчати моделі виконувати класифікацію для розрізнення електронних листів, які ви отримали як "Вхідні" та "Спам", використовуючи логістичну регресію. Ми також можемо прогнозувати значення, використовуючи лінійну регресію. Ми також можемо використовувати моделювання для групування даних, щоб зрозуміти логіку за цими кластерами. Наприклад, ми групуємо клієнтів електронної комерції, щоб зрозуміти їх поведінку на вашому веб-сайті. Це вимагає від нас ідентифікувати групи точок даних з алгоритмами кластеризації, такими як k-середніх або ієрархічна кластеризація.

Коротко кажучи, ми використовуємо регресію та прогнозування для передбачення майбутніх значень, класифікацію для ідентифікації та кластеризацію для групування значень.

Для реалізації даного етапу нам вже потрібно знати бібліотеку [scikit-learn](https://scikit-learn.org/stable/) (https://scikit-learn.org/stable/) а також бібліотеки для глибокого навчання такі як [TensorFlow](https://www.tensorflow.org/) (https://www.tensorflow.org/) чи [PyTorch](https://pytorch.org/) (https://pytorch.org/), у разі використання великої к-сті даних та складніших моделей. А також ми повинні вміти робити оцінку роботи нашої моделі, щоб розуміти, яка модель краща, як можна покращити її роботу і т.д. Відповідно для оцінки моделі нам потрібні деякі метрики, як наприклад RMSE, MAE чи F1 score.

**Інтерпретація (Застосування)**

[​](https://textbook.edu.goit.global/python/data-science-remaster/v1/docs/module-01/main#%D1%96%D0%BD%D1%82%D0%B5%D1%80%D0%BF%D1%80%D0%B5%D1%82%D0%B0%D1%86%D1%96%D1%8F-%D0%B7%D0%B0%D1%81%D1%82%D0%BE%D1%81%D1%83%D0%B2%D0%B0%D0%BD%D0%BD%D1%8F)Ми на останньому етапі проекту Data Science - інтерпретації моделей та даних. Прогностична сила моделі полягає у її здатності узагальнювати. Те, як пояснити модель, залежить від її здатності узагальнювати майбутні невідомі на теперішній момент дані.

Інтерпретація даних означає представлення ваших даних неспеціалісту в цій галузі. Ми демонструємо результати для відповіді на бізнес-питання, які ми ставили на початку проекту, разом з дієвими інсайтами, які ми знайшли через власне Data Science, тобто наш процес дослідження даних. Це і є основна робота Data Scientist'а - принести користь компанії, зробивши якісь оптимізації чи діставши якісь інсайти з даних. До прикладу запропонувати той товар, котрий купить користувач, правильно застосувати таргетовану рекламу і т.д.

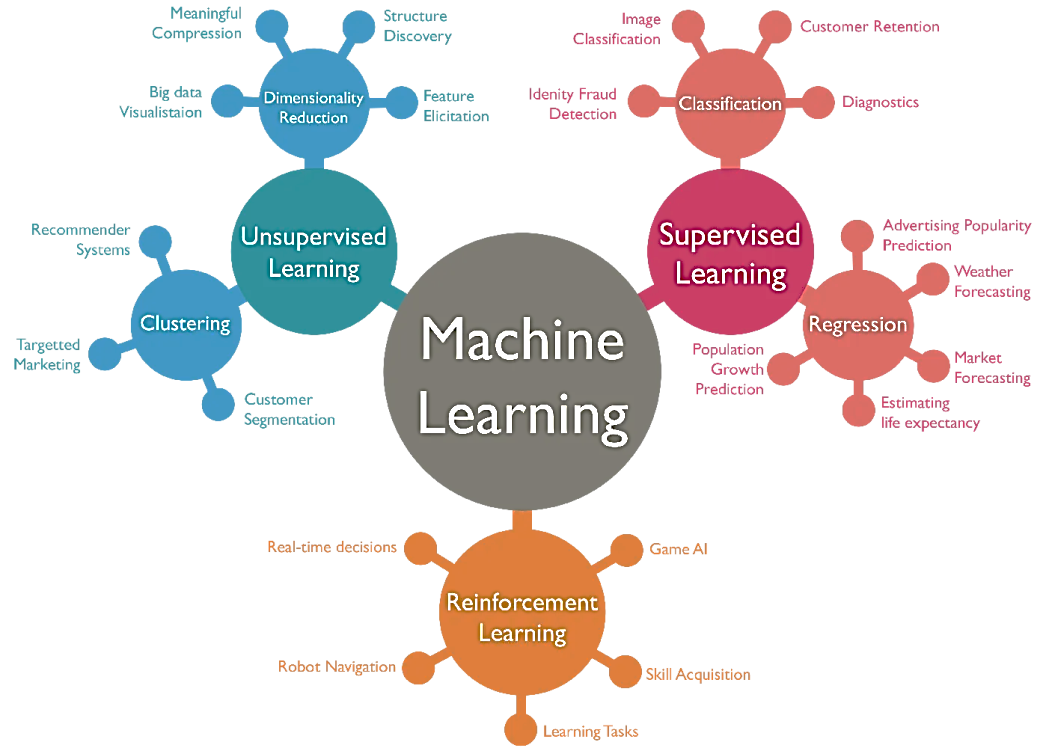
Дієвий інсайт є ключовим результатом. Ми вивчаємо, як повторити позитивний результат або запобігти негативному результату.

Крім того, вам потрібно візуалізувати свої результати відповідно до вашого бізнес-питання. Важливо представити ваші результати таким чином, щоб вони були корисні організації, інакше це буде безглуздо.

На цьому етапі тільки технічні навички недостатні. Одним із ключових навичок, яку ви повинні мати, є здатність розповісти чітку та дієву історію. Якщо ваша презентація не спонукає до дій вашу аудиторію, це означає, що ваша комунікація була неефективною. Пам'ятайте, що ви будете презентувати перед аудиторією без технічного фону, тому спосіб, яким ви спілкуєтесь з аудиторією, є ключовим.

**Класифікація алгоритмів машинного навчання**

[​](https://textbook.edu.goit.global/python/data-science-remaster/v1/docs/module-01/main#%D0%BA%D0%BB%D0%B0%D1%81%D0%B8%D1%84%D1%96%D0%BA%D0%B0%D1%86%D1%96%D1%8F-%D0%B0%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC%D1%96%D0%B2-%D0%BC%D0%B0%D1%88%D0%B8%D0%BD%D0%BD%D0%BE%D0%B3%D0%BE-%D0%BD%D0%B0%D0%B2%D1%87%D0%B0%D0%BD%D0%BD%D1%8F)На етапі побудови моделі, ми згадували деякі задачі та алгоритми, котрі існують для їх вирішення. Для кращого розуміння, які взагалі існують алгоритми, давайте зануримось у поточну класифікацію алгоритмів.



<https://towardsdatascience.com/machine-learning-algorithms-in-laymans-terms-part-1-d0368d769a7b>

**Навчання з учителем**

[​](https://textbook.edu.goit.global/python/data-science-remaster/v1/docs/module-01/main#%D0%BD%D0%B0%D0%B2%D1%87%D0%B0%D0%BD%D0%BD%D1%8F-%D0%B7-%D1%83%D1%87%D0%B8%D1%82%D0%B5%D0%BB%D0%B5%D0%BC)Навчання з учителем (supervised learning) - це один з видів машинного навчання, в якому модель навчається на основі попередньо позначених (маркованих) даних. Іншими словами, для кожного набору вхідних даних є правильна відповідь, яку модель намагається передбачити. Модель тренується на цих позначених даних з метою знаходження патернів та створення правильних передбачень для нових, невідомих даних.

Один з прикладів застосування supervised learning - це задача класифікації електронної пошти на "спам" або "інші повідомлення". В даному випадку, модель будується на основі попередньо позначених даних, які містять інформацію про зразки спаму та нормальних повідомлень. Потім, модель навчається розпізнавати зразки та встановлювати, чи є нові повідомлення "спамом" чи ні.

Ще один приклад - це прогнозування цін на нерухомість. Модель може бути навчена на попередньо позначених даних, які містять інформацію про характеристики, ознаки нерухомості та їх продажну ціну. Потім, модель навчається розпізнавати зв'язок між характеристиками та їх ціною, щоб передбачити ціну для нових будинків (нерухомостей).

У обох прикладах, модель тренується на попередньо позначених даних з метою передбачення правильних відповідей для нових, невідомих даних. Навчання з учителем дозволяє створювати точні прогнози та класифікації з високим рівнем достовірності, що робить його корисним для широкого спектру задач у різних галузях.

**Навчання без учителя**

[​](https://textbook.edu.goit.global/python/data-science-remaster/v1/docs/module-01/main#%D0%BD%D0%B0%D0%B2%D1%87%D0%B0%D0%BD%D0%BD%D1%8F-%D0%B1%D0%B5%D0%B7-%D1%83%D1%87%D0%B8%D1%82%D0%B5%D0%BB%D1%8F)Навчання без учителя (unsupervised learning) - це один з підходів машинного навчання, який дозволяє виявляти приховані залежності в даних, без використання міток або результатів. Основна мета полягає у знаходженні структури в даних та кластеризації, де алгоритми виявляють групи схожих елементів у наборі даних, що не мають явних міток.

Наприклад, розглянемо задачу кластеризації, де маємо набір зображень та бажаємо розділити їх на різні групи. Натомість, у наборі даних немає інформації про те, який об'єкт до якої групи належить. Алгоритми навчання без вчителя, такі як k-means, можуть бути використані для кластеризації зображень на основі їхньої схожості.

Іншим прикладом є зменшення розмірності даних, де маємо великий набір даних із багатьма функціями, та хочемо зменшити кількість ознак для подальшого аналізу. У такому випадку, можна використовувати алгоритми пониження розмірності, такі як Principal Component Analysis (PCA), для зменшення кількості ознак із збереженням якомога більше інформації.

Узагальнюючи, навчання без учителя - це підхід до машинного навчання, який дозволяє знаходити структуру в даних без попереднього навчання на вибірці даних. Це відкриває можливості для виявлення нових залежностей та патернів, які можуть бути корисними в багатьох областях, таких як рекомендації, обробка природних мов, аналіз даних та багато інших.

**Рекомендаційні системи**

[​](https://textbook.edu.goit.global/python/data-science-remaster/v1/docs/module-01/main#%D1%80%D0%B5%D0%BA%D0%BE%D0%BC%D0%B5%D0%BD%D0%B4%D0%B0%D1%86%D1%96%D0%B9%D0%BD%D1%96-%D1%81%D0%B8%D1%81%D1%82%D0%B5%D0%BC%D0%B8)Рекомендаційні системи - це програмні рішення, що аналізують дані про користувачів та предмети, що їх цікавлять, з метою надання персоналізованих рекомендацій користувачам. Такі системи використовуються в різних сферах, включаючи електронну комерцію, засоби масової інформації, музику, соціальні мережі тощо.

У рекомендаційних системах використовуються два основних підходи: колаборативний та контент-базований. Колаборативний підхід базується на історії взаємодії користувача з системою та аналізу спільності цих взаємодій з іншими користувачами. Контент-базований підхід використовує відомості про властивості та характеристики предметів для порівняння з вподобаннями користувача.

Один з прикладів рекомендаційної системи - Netflix. Система аналізує історію перегляду користувача, їх вподобання та рейтинги фільмів, а також аналізує дані про характеристики самого фільму, такі як жанр, акторський склад, режисер тощо. На основі цих даних система робить персоналізовані рекомендації користувачам.

Інший приклад - Amazon. Рекомендаційна система Amazon аналізує історію покупок користувачів, їх відгуки та рейтинги товарів, які вони придбали. Система також аналізує властивості товарів, щоб порівняти їх з вподобаннями користувача та знайти товари, які він може бути зацікавлений купити.

Рекомендаційні системи є важливим інструментом для підвищення рівня задоволення користувачів і збільшення прибутку компанії. Їх використовують в різних галузях, таких як електронна комерція, засоби масової інформації, музика, ігри та інші.

**Навчання з підкріпленням**

[​](https://textbook.edu.goit.global/python/data-science-remaster/v1/docs/module-01/main#%D0%BD%D0%B0%D0%B2%D1%87%D0%B0%D0%BD%D0%BD%D1%8F-%D0%B7-%D0%BF%D1%96%D0%B4%D0%BA%D1%80%D1%96%D0%BF%D0%BB%D0%B5%D0%BD%D0%BD%D1%8F%D0%BC)Навчання з підкріпленням (reinforcement learning) є однією з головних підгалузей машинного навчання, яка полягає у тому, щоб навчити агента взаємодіяти з довкіллям, максимізуючи нагороду або мінімізуючи покарання.

У навчанні з підкріпленням немає правильних або неправильних відповідей. Замість цього, агент знаходиться у взаємодії з довкіллям і робить певні дії, за які отримує нагороду або покарання. Метою агента є вивчення стратегії, яка дозволяє йому максимізувати нагороду на довгострокову перспективу.

Один з прикладів застосування навчання з підкріпленням - гра в Го. Агент (наприклад, комп'ютерна програма) знаходиться взаємодії з довкіллям (Го-дошка) і має робити певні ходи, за які отримує очки, що відображають його успішність. Метою агента є навчитися грати в Го краще, щоб заробляти якомога більше очок. AlphaGo компанії Google навчилася грати в гру Гo та перемогла одного з найкращих гравців у світі.

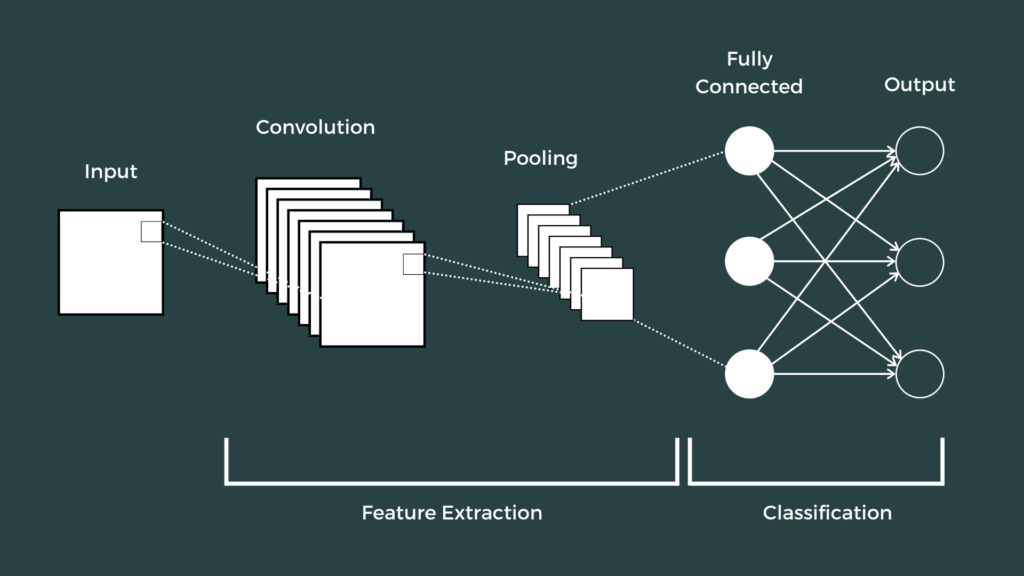
Інший приклад застосування навчання з підкріпленням - роботи, які виконують різні завдання. Робот взаємодіє з довкіллям, роблячи певні дії, і отримує нагороду або покарання в залежності від того, наскільки ефективно він виконує завдання.

Навчання з підкріпленням є досить складним процесом, оскільки агент має вирішувати проблеми з довгостроковою перспективою, де кожен крок може впливати на подальшу поведінку. Однак це дає можливість розв'язувати складні завдання, для яких немає простого рішення або алгоритму.

**Нейронні мережі (глибоке навчання)**

[​](https://textbook.edu.goit.global/python/data-science-remaster/v1/docs/module-01/main#%D0%BD%D0%B5%D0%B9%D1%80%D0%BE%D0%BD%D0%BD%D1%96-%D0%BC%D0%B5%D1%80%D0%B5%D0%B6%D1%96-%D0%B3%D0%BB%D0%B8%D0%B1%D0%BE%D0%BA%D0%B5-%D0%BD%D0%B0%D0%B2%D1%87%D0%B0%D0%BD%D0%BD%D1%8F)Deep Learning - це підгалузь машинного навчання, яка використовує нейронні мережі з багатьма шарами для автоматичного виконання завдань, таких як класифікація зображень, розпізнавання мовлення та обробка природної мови. Глибоке навчання базується на технології нейронних мереж, яка вивчає абстрактні характеристики даних, розпізнаючи взаємозв'язки між вхідними даними та вихідними мітками. Воно широко використовується в багатьох галузях, таких як медицина, фінанси та інші.

Залежно від того з якими даними ми працюємо, у нас є різні специфічні нейронні мережі, котрі більш ефективно використовувати у даному завданні.

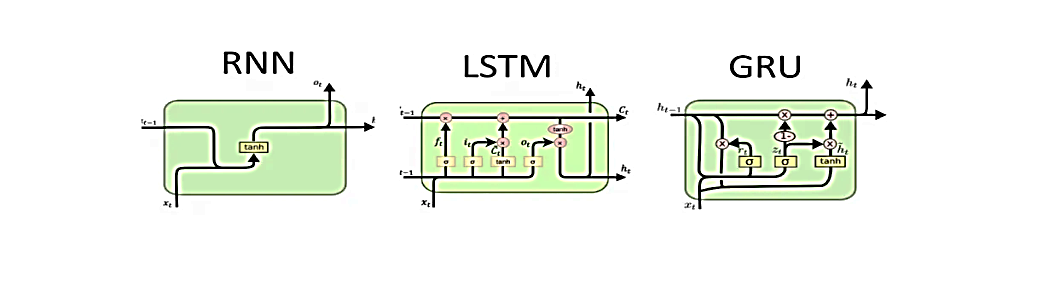


**Приклад згорткової нейронної мережі**

Згорткові нейронні мережі (Convolutional Neural Networks або CNN) є видом алгоритмів глибокого навчання, який спеціалізується на обробці даних, що мають сітчасту структуру, такі як зображення, відео та звук. Вони використовуються для автоматичної обробки та аналізу великої кількості даних, що дозволяє їм виконувати завдання, такі як розпізнавання об'єктів на зображеннях, класифікацію зображень та інші пов'язані з цим задачі.

Основна ідея згорткових нейронних мереж полягає в тому, що вони використовують невеликі фільтри для згортки зображення та отримання відповідей на кожну з цих згорток. Далі, за допомогою шару підвибірки (pooling), отримані відповіді зменшуються, що дозволяє зменшити кількість параметрів та спростити обчислення. Застосування цих операцій дозволяє згортковим нейронним мережам розпізнавати об'єкти на зображеннях з високою точністю та ефективністю.

Прикладами застосування згорткових нейронних мереж є класифікація зображень, розпізнавання облич, відстеження руху та розпізнавання мови. Зокрема, вони широко використовуються в комп'ютерному зорі, рекомендаційних системах, робототехніці та інших областях, що пов'язані з обробкою даних зі зображень та відео.



Рекурентні нейронні мережі (RNN) - це тип нейронних мереж, який може аналізувати послідовності даних, де кожен елемент взаємозв'язаний з попередніми елементами послідовності.

Одна з основних особливостей RNN полягає в тому, що вони зберігають стан (пам'ять) з попередніх елементів вхідних даних і використовують його для здійснення прогнозів для наступного елементу. Це дозволяє їм враховувати контекст, що допомагає у більш точному прогнозуванні.

RNN зазвичай використовують для роботи з послідовностями даних, такими як мова, часові ряди, музика. Вони знаходять застосування в різних областях, включаючи машинний переклад, розпізнавання мови, генерацію тексту, прогнозування часових рядів та інше.

Один з найбільш відомих прикладів RNN - це Long Short-Term Memory (LSTM), який є спеціальним типом RNN. Він може довгостроково запам'ятовувати інформацію та уникнути проблеми зниклих градієнтів, що дозволяє йому бути дуже ефективним у роботі з послідовними даними. LSTM використовується для різноманітних завдань, включаючи автокомплітацію тексту, стиснення тексту, генерацію музики та багато іншого.

**Середовище роботи**

[​](https://textbook.edu.goit.global/python/data-science-remaster/v1/docs/module-01/main#%D1%81%D0%B5%D1%80%D0%B5%D0%B4%D0%BE%D0%B2%D0%B8%D1%89%D0%B5-%D1%80%D0%BE%D0%B1%D0%BE%D1%82%D0%B8)Окрім того, що ми будемо працювати у середовищі мови програмування Python, Data Scientist'ам зручно користуватись [Jupyter Notebook](https://jupyter.org/) (https://jupyter.org/) або [Google Colab](https://colab.research.google.com/) (https://colab.research.google.com/), а не більш звичний IDE, як наприклад PyCharm. І на це є декілька причин:

* обидва ці інструменти надають можливість виконувати код покроково, а також зберігати та переглядати результати виконання кожного кроку окремо. Це дозволяє легко зрозуміти, що відбувається в коді, а також відлагоджувати та вдосконалювати моделі;
* Jupyter Notebook та Google Colab підтримують безліч бібліотек та фреймворків машинного навчання, що значно спрощує розробку та навчання моделей. Крім того, обидва інструменти мають можливість зберігати та перезавантажувати стан моделей, що зберігає час та зусилля при налагодженні та оптимізації моделей;
* дані інструменти є безкоштовними та легко доступними з будь-якого пристрою з підключенням до Інтернету. Це дає можливість легко обмінюватись результатами та колаборувати з колегами;
* також ви можете користуватись Jupyter Notebook на віддалених серверах, наприклад користуючись інструментом [SageMaker](https://aws.amazon.com/sagemaker/) ([https://aws.amazon.com/ sagemaker/](https://aws.amazon.com/%20sagemaker/)) від AWS.

Отже, Jupyter Notebook та Google Colab є потужними інструментами для Data Scientist'ів, що дозволяють швидко та ефективно розробляти, навчати та налагоджувати моделі машинного навчання.

**Встановлення**

[​](https://textbook.edu.goit.global/python/data-science-remaster/v1/docs/module-01/main#%D0%B2%D1%81%D1%82%D0%B0%D0%BD%D0%BE%D0%B2%D0%BB%D0%B5%D0%BD%D0%BD%D1%8F)Якщо Google Colab не вимагає від нас якогось додаткового налаштування, встановлення, ми легко ним користуємось зі свого диску, як і гугл документом чи таблицею, то для ноутбука ми повинні додатково встановити даний інструмент. Якщо у вас вже встановлений Python, то ви легко можете встановити ноутбук завдяки pip:

pip install notebook

та запустити його відповідно за допомогою наступної команди:

jupyter notebook

Іншим способом встановлення це використання [Anaconda](https://www.anaconda.com/) (https://www.anaconda.com/). Це безкоштовний відкритий дистрибутив Python, який містить інструменти для наукових обчислень та аналізу даних. Він містить більше 1500 пакетів, включаючи наукові бібліотеки, такі як NumPy, pandas, SciPy, scikit-learn та TensorFlow, а також інструменти для візуалізації даних, такі як Matplotlib.

Anaconda постачається з власним менеджером пакетів, який дозволяє легко установлювати, оновлювати та керувати пакетами, а також з додатковими інструментами, такими як Jupyter Notebook, Spyder та JupyterLab.

Тому встановлення Anaconda може відразу забезпечити нас всіма необхідними інструментами.

**Додаткові матеріали**

[​](https://textbook.edu.goit.global/python/data-science-remaster/v1/docs/module-01/main#%D0%B4%D0%BE%D0%B4%D0%B0%D1%82%D0%BA%D0%BE%D0%B2%D1%96-%D0%BC%D0%B0%D1%82%D0%B5%D1%80%D1%96%D0%B0%D0%BB%D0%B8)Сьогодні ми вперше занурились у світ Data Science та ознайомились з аспектами роботи фахівця у даній сфері та базовою термінологією та деякими інструментами. Для кращого запам'ятовування та розуміння пропоную переглянути додаткові матеріали на дану тему:

* [5 кроків в життєвому циклі Data Science проекту](https://towardsdatascience.com/machine-learning-algorithms-in-laymans-terms-part-1-d0368d769a7b)

(https://towardsdatascience.com/machine-learning-algorithms-in-laymans-terms-part-1-d0368d769a7b)

* Навчання з учителем та без учителя - [стаття](https://www.ibm.com/cloud/blog/supervised-vs-unsupervised-learning) (https://www.ibm.com/products) та [відео](https://youtu.be/W01tIRP_Rqs) (https://www.youtube.com/watch?v=W01tIRP\_Rqs)
* [Відеоогляд основних середовищ роботи для Data Scientist'а](https://youtu.be/5pf0_bpNbkw)

(https://www.youtube.com/watch?v=5pf0\_bpNbkw)