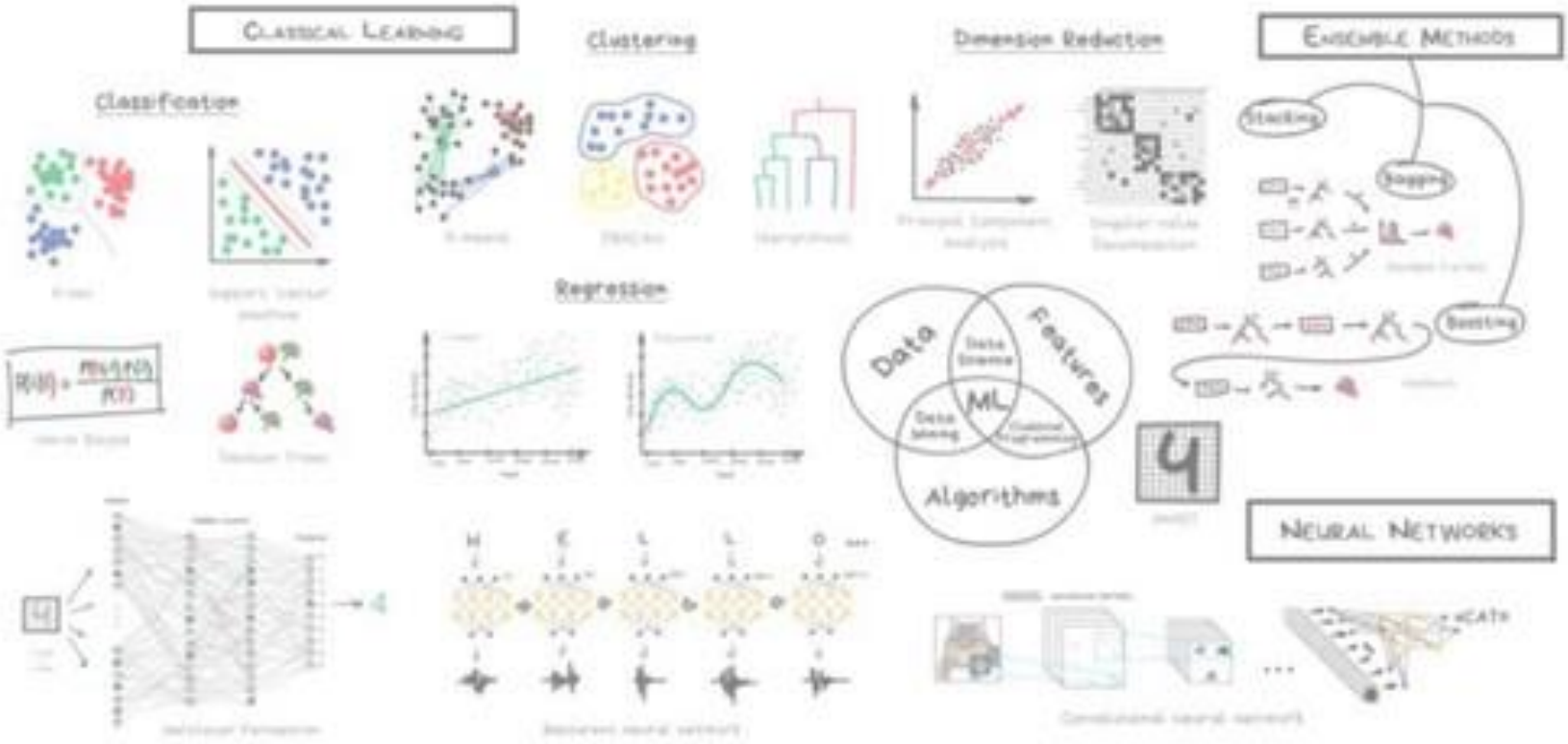
Машинне навчання



Лекція №1

ВСТУП

Перш ніж говорити про навчання, почнемо з аналізу термінології.
Data Science - це загальне найменування дисциплін з вивчення даних,
Machine Learning - це підрозділ Data Science, який займається розбудовою розумних моделей. Такі моделі можуть використовуватися для передбачення покупки товару користувачем, рекомендацій у соцмережах (рекомендаційні системи), розпізнавання зображень тощо.
Data Science спеціалісти займаються дослідженнями. В іноземних компаніях такій посаді відповідають позиції research-інженерів — це переважно математики, які працюють з теоретичною частиною алгоритмів та досліджують різноманітні закономірності.
Machine Learning інженери, своєю чергою, займаються побудовою моделей з урахуванням отриманих даних. Але такий поділ існує лише теоретично чи лише деяких країнах.
В Україні Data Science та Machine Learning раніше використовувалися як слова-синоніми, зараз ці поняття вже починають розділяти. У наших реаліях вакансії, де необхідне знання Machine Learning, найчастіше називаються Data Scientist і навпаки. Тому, якщо ви хочете працювати з даними, вам слід вивчити і те, й інше.

Класи застосування МН

- *Навчання з учителем (Supervised learning).* Ми маємо чимало прикладів і правильні, еталонні, відповіді до кожного з них. Наприклад, історичні дані успішности студентів з минулих років.
- *Навчання без учителя (Unsupervised learning).* Це знаходження сенсу в даних без наявности конкретної, правильної, відповіді. Інакше кажучи, ми не знаємо, що шукаємо, але дуже сильно хочемо знайти.
- *Навчання із закріпленням (Reinforcement learning).* Це розширений варіант навчання з учителем, коли замість еталонних відповідей програма наприкінці отримує зворотний зв'язок. Наприклад, якщо ми вчимо машину грати в шахи, оцінка кожного можливого ходу є надто затратною й не завжди правильною, але наприкінці гри дуже легко визначити виграш або програш алгоритму.
- *Перенесення навчання (Transfer learning).* Навчаємо модель для однієї проблеми й використовуємо результат навчання для модифікації даних від іншої. Наприклад, фільтр, за допомогою якого ви робите селфі подібними до творінь Вінсента ван Гога та який домальовує заячі вуха у відеоконференції або змушує Обаму вимовляти ваші слова його ж голосом і з його мімікою.

Процес навчання Data Science та Machine Learning можна розділити на 4 блоки:

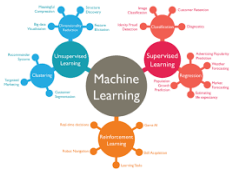
1

Математика – основа DataScience та машинного навчання Знадобиться для глибинного розуміння аналізу даних та принципів MachineLearning

2

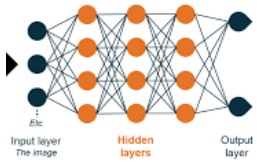
Мова програмування

3



Базові алгоритми машинного навчання

4



Deep learning (нейронні мережі)

Математика

Для початку давайте розберемося, чи потрібна математика в роботі з Data Science і Machine Learning. Короткою відповіддю буде: так, потрібна. Безумовно, є багато прикладів того, як успішні Data Scientists займають призові місця на Kaggle-змаганнях, не маючи технічної освіти. Але навіть вони погодяться, що знання математики дає значну перевагу у роботі з Data Science.
Незважаючи на те, що майже всі алгоритми реалізуються в бібліотеках Python і R, розуміння базових математичних концепцій значно спростить ваше навчання та виконання прикладних завдань. Крім того, у більшості статей про машинне навчання містяться математичні викладки, читати які без знань математики буде важко.

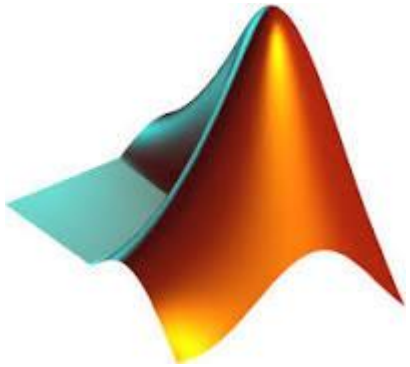
Для успішної роботи мінімально потрібно розуміти три розділи математики:

- Основи лінійної алгебри
- Основи математичного аналізу (інтегрування, похідні та часткові похідні)
- Основи теорії ймовірностей та математична статистика

Мова програмування

Для роботи з даними ви повинні вміти програмувати.
Наприклад, щоб завантажити дані, розпарсувати, синтезувати нові ознаки або втілити в життя будь-яку іншу вашу ідею.
Основною мовою програмування більшості Data Science фахівців є **Python**.

Python сама по собі дуже проста мова, в ній реалізовано безліч бібліотек для обробки та аналізу даних.
Популярні раніше **R** і **Matlab** сьогодні зустрічаються дедалі рідше.



Базові алгоритми

Для того щоб розпочати свій професійний шлях у машинному навчанні, вам необхідно знати основні класи задач Machine Learning, які існують алгоритми та які підходи дозволяють вирішити той чи інший клас задач. Ви також повинні розрізняти алгоритми різних спеціалізацій, розуміти їх переваги та недоліки. Машинне навчання є практичною дисципліною, тому дуже важливо застосовувати отримані знання на реальних даних. Візьміть за правило заходити на Kaggle - це платформа для змагань з Data Science. Тут ви знайдете безліч датасетів, на яких зможете розібрати рішення інших учасників та попрактикувати свої аналітичні навички. І згодом зможете спробувати щастя у якомусь відкритому конкурсі.

Deep learning

Маючи базове розуміння принципів машинного навчання і знання Python, можна приступити до вивчення Deep Learning. Це один із розділів машинного навчання, в основі якого лежить використання нейронних мереж.

Процес вивчення Data Science

- Вирішувати практичні завдання на Kaggle.
- Вирішуйте їх самостійно, розбирайте рішення інших людей – все це допомагає розвивати логіку та аналітику.
- Зверніть увагу на блоги різних Data Scientists, YouTube-канали з розбором та описом того, як вони будували модель, яку логіку вкладали у рішення.
- Крім того, у вільному доступі є багато даних, на яких можна практикуватись. Візьміть, наприклад, статистику із захворюваності COVID-19 і спробуйте знайти закономірності (такий конкурс нещодавно проводили на Kaggle).
- Ви можете подивитися на хороші рішення, розібрати логіку і поступово покращувати свої знання алгоритмів.
- При постійній практиці та наявності аналітичного мислення дуже скоро ви почнете робити перші успіхи у Data Science.

Що почитати

Хоча профільна література може допомогти у вашому навчанні, не забувайте, що технології розвиваються дуже швидко, а інформація у книгах застаріває. Для успіху в Data Science важлива практика, розуміння предметної галузі, завдань та інструментів, якими володієте.

Мова програмування Python

Сьогодні існує величезна кількість бібліотек для машинного та глибокого навчання. Щоб полегшити завдання вибору ми розглянемо лише найпопулярніші та найнеобхідніші бібліотеки, які покривають усі базові потреби для початку роботи з ML та DL.

Jupyter Notebook: робота з даними, кодом та графіками

Якщо в традиційному програмуванні більшу частину часу ви проводите в текстових редакторах або IDE-шках, то в Data Science більшість коду пишеться в Jupyter Notebook.
Це простий та потужний інструмент для аналізу даних. Він дозволяє писати код Python, R та інших мовах, додавати текстові описи в Markdown, вбудовувати графіки та діаграми безпосередньо в інтерактивну веб-сторінку. Плюс до всього Google випустив безкоштовний сервіс Google Colab, який надає хмарну версію Jupyter Notebook і дає можливість проводити обчислення на CPU і GPU. Всі потрібні пітонівські ML бібліотеки вже встановлені, так що можна починати відразу там.

Scikit-learn: найкраща бібліотека для класичних ML алгоритмів

Scikit-learn – одна з найпопулярніших ML бібліотек на сьогодні. Вона підтримує більшість алгоритмів навчання як з учителем, так і без: лінійна та логістична регресія, метод опорних векторів (SVM), Naive Bayes класифікатор, градієнтний бустинг, кластеризація, KNN, k-середні та багато інших.

Крім цього, Scikit-learn містить безліч корисних утиліт для підготовки даних та аналізу результатів. Ця бібліотека в основному призначена для класичних алгоритмів машинного навчання, тому її функціонал для нейронних мереж дуже обмежений, а для завдань глибокого навчання вона не може бути використана зовсім.На додаток до дуже якісної документації, Scikit-learn містить розділ із туторіалами, в якому показано, як працювати з бібліотекою, а також даються базові знання з машинного навчання.

Pandas:
вилучення та підготовка даних

Аналіз та підготовка даних найчастіше займає більшу частину часу при вирішенні ML завдань. Дані можуть бути отримані в CSV, JSON, Excel або в іншому структурованому (або не дуже) форматі, і вам потрібно обробити їх для того, щоб використовувати в ML моделях. Для цього використовується бібліотека Pandas. Це потужний інструмент, який дозволяє швидко аналізувати, модифікувати та готувати дані для подальшого використання в інших ML та DL бібліотеках, таких як Scikit-learn, TensorFlow або PyTorch. У Pandas можна завантажувати дані з різних джерел: SQL баз, CSV, Excel, JSON файлів та інших менш популярних форматів. Коли дані завантажені у пам'ять, з ними можна виконувати безліч різних операцій для аналізу, трансформації, заповнення відсутніх значень та очищення набору даних. Pandas дозволяє виконувати безліч SQL-подібних операцій над наборами даних: об'єднання, угруповання, агрегування тощо. Також вона надає вбудований набір популярних статистичних функцій для базового аналізу. Jupyter Notebook також підтримує Pandas та реалізує гарну візуалізацію його структур даних. Сайт Pandas містить докладну документацію. Але почати можна з 10-хвилинного туторіалу, який показує всі основні фішки та можливості бібліотеки.

Бібліотека NumPy: багатовимірні масиви та лінійна алгебра

Основний функціонал NumPy полягає у підтримці багатовимірних масивів даних та швидких алгоритмів лінійної алгебри. Саме тому NumPy – ключовий компонент Scikit-learn, SciPy та Pandas.Зазвичай NumPy використовують як допоміжну бібліотеку до виконання різних математичних операцій із структурами даних Pandas, тому варто вивчити її базові можливості.Для цього відмінно підійде вступний туторіал Numpy, а також основи NumPy.

Matplotlib та Seaborn: побудова графіків та візуалізація даних

Matplotlib – це стандартний інструмент у наборі дата-інженера. Він дозволяє створювати різноманітні графіки та діаграми для візуалізації отриманих результатів.Графіки, створені Matplotlib, легко інтегруються в Jupyter Notebook. Це дозволяє візуалізувати дані та результати, отримані при обробці моделей.Для цієї бібліотеки створено багато додаткових пакетів. Один із найбільш популярних – це Seaborn. Його основна фішка - готовий набір найчастіше використовуваних статистичних діаграм і графіків.Традиційно, обидві бібліотеки мають розділ із туторіалами на їхніх сайтах, але більш ефективним підходом буде зареєструватися на сайті Kaggle та подивитися в розділі «Kernels» готові приклади використання, наприклад Comprehensive Data Exploration with Python.

Tensorflow та Keras: бібліотеки глибокого навчання

Будь-яка бібліотека глибокого навчання містить три ключові компоненти: багатовимірні масиви (вони ж тензори), оператори лінійної алгебри та обчислення похідних.У **Tensorflow**, бібліотеці глибокого навчання від **Google**, добре реалізовані всі три компоненти. Поряд із **CPU**, вона підтримує обчислення на **GPU** та **TPU** (тензорних процесорах **Google**).В даний час це найпопулярніша бібліотека глибокого навчання, внаслідок чого по ній створено безліч туторіалів та онлайн-курсів. Але зрілість має і зворотний бік - дуже корявий **API** і вищий поріг входу, порівняно з тією ж **PyTorch**.
Keras - це надбудова над **Tensorflow**, яка вирішує безліч юзабіліті-проблем останньої. Її головна фішка це можливість будувати архітектуру нейронної мережі з використанням красивого **Python DSL**. Для **Keras** також написано безліч навчальних матеріалів, тому розібратися з нею нескладно.

PyTorch: альтернативна бібліотека глибокого навчання

PyTorch – це друга за популярністю DL бібліотека після Tensorflow, яка створена у Facebook. Її сильна сторона в тому, що вона була розроблена для Python і тому використовує його стандартні ідіоми. У порівнянні з Tensorflow тут поріг входу набагато нижчий, а будь-яку нейронну мережу можна побудувати з використанням стандартних ООП класів та об'єктів.Також її легше налагоджувати, тому що код виконується як звичайний Python код – немає етапу компіляції, як у TensorFlow. Тому можна скористатися навіть пітонівським відладчиком.Якщо порівнювати з Keras, PyTorch — багатослівніший, але менш магічний.PyTorch теж має свою надбудову — це бібліотека fastai. Вона дозволяє вирішити більшість стандартних DL завдань у кілька рядків коду. Але що робить fastai справді особливою – це їхній неймовірний онлайн-курс Practical Deep Learning for Coders.

Дякую за увагу