ЛЕКЦІЯ 10

МАШИННЕ НАВЧАННЯ ТА ШТУЧНИЙ ІНТЕЛЛЕКТ

ПЛАН

1. Складові машинного навчання та штучний інтеллект
2. Класифікація методів машинного навчання
3. Класичне навчання

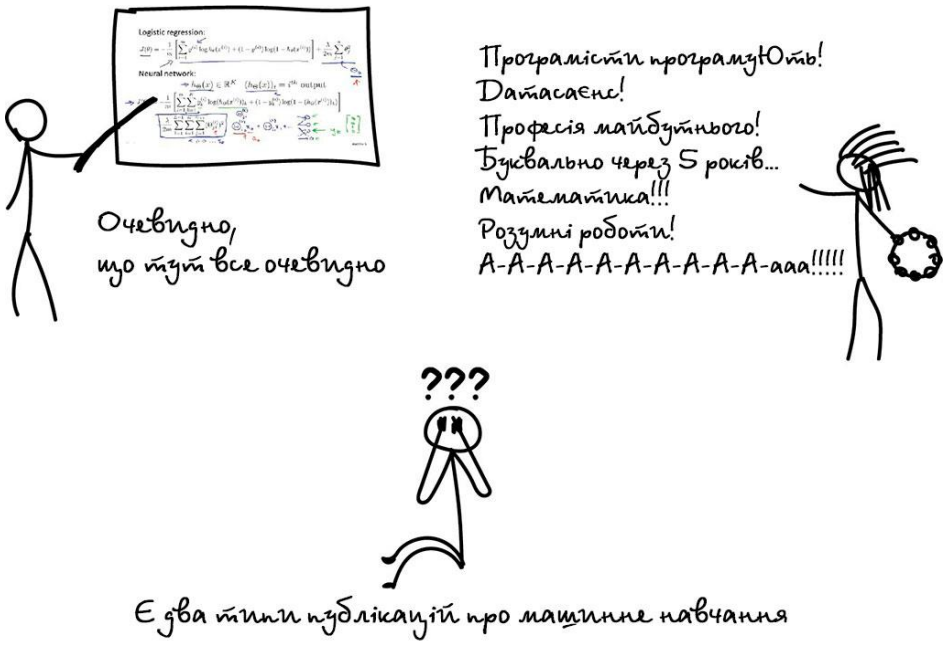
ЛІТЕРАТУРА

Сайт http://www.mmf.lnu.edu.ua/ar/1739

ВСТУП

Загалом статті про машинне навчання діляться на два типи: це або 2-3-4..-томники з формулами і теоремами, які мало хто зміг дочитати навіть до середини, або казки про штучний інтелект, професії майбутнього, чарівних дата-саєнтистів.

Нижче розглянемо пост-вступ для тих, хто хоче нарешті розібратися в машинному навчанні - простою мовою, без формул-теорем, зате з прикладами реальних задач та їх розв'язань.



Навіщо навчати машини

Приклад

Припустимо, що Олег хоче купити автомобіль і обдумує скільки грошей йому потрібно для цього накопити. Він переглянув десяток оголошень в інтернеті і побачив, що нові автомобілі коштують близько $ 20 000, однорічні - приблизно $ 19 000, дворічні - $ 18 000 і так далі.

Зрозуміло, що Олег-аналітик виводить формулу: адекватна ціна автомобіля починається від $ 20 000 і падає на $ 1000 щороку, поки не упреться в $ 10 000.

Олег зробив те, що в машинному навчанні називають регресією - передбачив ціну за відомими даними. Люди роблять це постійно, коли вираховують за скільки продати старий айфон або скільки шашлику взяти на заміську забаву (моя формула - півкіло на людину в добу).

Очевидно, що було б зручно мати формулу під кожну проблему на світі. Але взяти ті ж ціни на автомобілі: крім пробігу є десятки комплектацій, різний технічний стан, сезонність попиту і ще стільки неочевидних факторів, які Олег, навіть при всьому бажанні, не врахував би в голові.

Люди тупі (вибачайте!) і ліниві, а тому треба змусити працювати роботів. Нехай машина подивиться на наші дані, знайде в них закономірності і навчиться передбачати для нас відповідь. Найцікавіше, що в підсумку машина може знаходити навіть такі закономірності, про які люди не здогадуються.

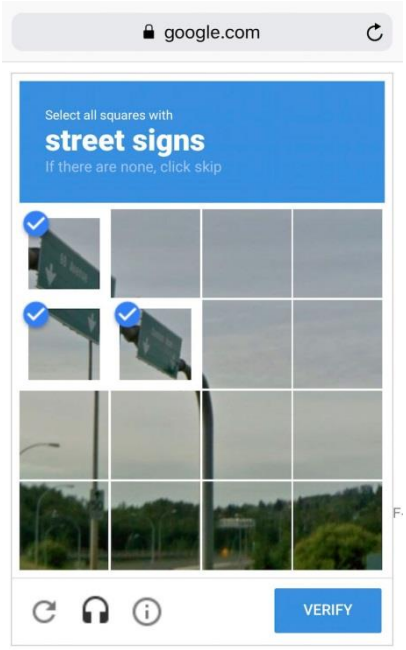
Так народилося машинне навчання.

1. СКЛАДОВІ МАШИННОГО НАВЧАННЯ ТА ШТУЧНИЙ ІНТЕЛЛЕКТ

Мета машинного навчання - передбачити результат за вхідними даними. Чим різноманітніші вхідні дані, тим простіше машині знайти закономірності і тим точніший результат.

Отже, якщо ми хочемо навчити машину, нам потрібні три речі: **дані, ознаки, алгоритми.**

Дані Хочемо виявляти спам - потрібні приклади спам-листів, передбачати курси акцій - потрібна історія цін, дізнатися інтереси користувача - потрібні його лайки або пости. Даних потрібно якомога більше. Десятки тисяч прикладів - це отой злий мінімум для відчайдухів.



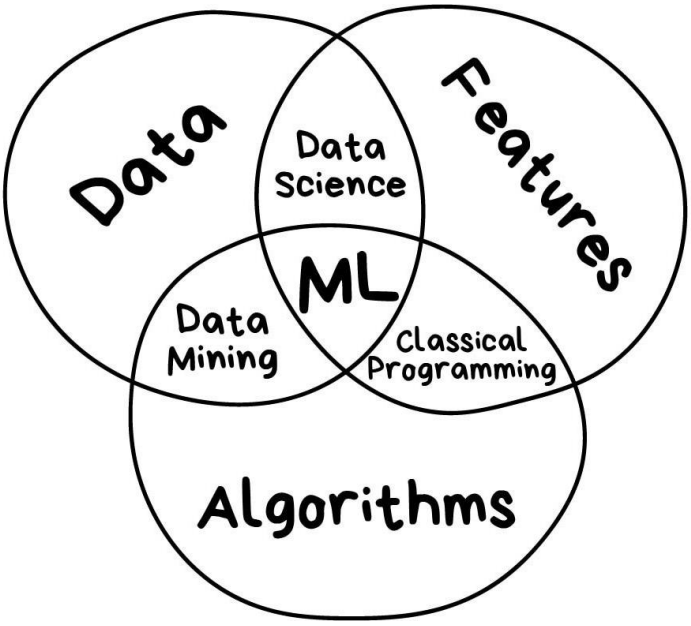
Дані збирають як тільки можуть. Хтось вручну - процес триваліший, даних менше, зате без помилок. Хтось автоматично - просто зливає машині все, що знайшлося, і вірить у щось краще. Найхитріші, типу гугла, використовують своїх же користувачів для безкоштовної розмітки. Згадайте ReCaptcha, яка іноді вимагає «знайти на фотографії всі дорожні знаки» - це воно і є.

За хорошими наборами даних (датасетами) йде велике полювання. Великі компанії інколи (й таке буває) розкривають свої алгоритми, але датасети - вкрай рідко.

Ознаки. Ми називаємо їх фічами (features), так що фаріонщикам доведеться страждати. Фічі, властивості, характеристики, ознаки - ними можуть бути пробіг автомобіля, стать користувача, ціна акцій, навіть лічильник частоти появи слова в тексті.

Машина повинна знати, на що їй конкретно дивитися. Добре, коли дані просто лежать в табличках - назви їх колонок і є фічі. А якщо у нас сто гігабайтів картинок з котами? Коли ознак багато, модель працює повільно і неефективно. Найчастіше відбір правильних фічів займає більше часу, ніж все інше навчання. Але бувають і зворотні ситуації, коли юзер сам вирішує відібрати тільки «правильні» на його погляд ознаки і вносить в модель суб'єктивність - вона починає дико брехати.

Алгоритм Зазвичай, одну й ту ж задачу майже завжди можна розв'язати різними методами-способами. Від вибору методу залежить точність, швидкість роботи і розмір готової моделі. Але є один нюанс: якщо дані - «сміття», то навіть найкращий алгоритм не допоможе. Не зациклюйтеся на відсотках, краще зберіть побільше даних.



Навчання та Інтелект

Одного разу в одному хіпстерскому виданні була стаття під назвою «Чи замінять нейромережі машинне навчання». Піарники в своїх прес-релізах обзивають «штучним інтелектом» будь-яку лінійну регресію, з якою вже дітки у дворі бавляться. Раз і назавжди пояснимо різницю на зображенні.

Штучний інтелект - назва всієї області, як біологія або хімія.

Машинне навчання - це розділ штучного інтелекту. Важливий, але не єдиний.

Нейромережі – можна розглядати як один з типів машинного навчання. Популярний, але є й інші, не гірші.

Глибоке навчання - архітектура нейромереж, один з підходів до їх побудови та навчання. На практиці мало хто відрізняє, де глибокі нейромережі, а де не дуже. Кажуть назву конкретної мережі і все.

Порівнювати можна тільки речі одного рівня, інакше виходить повний буллщит типу «що краще: машина чи колесо?» Не ототожнюйте терміни без причини, щоб не виглядати дурниками.

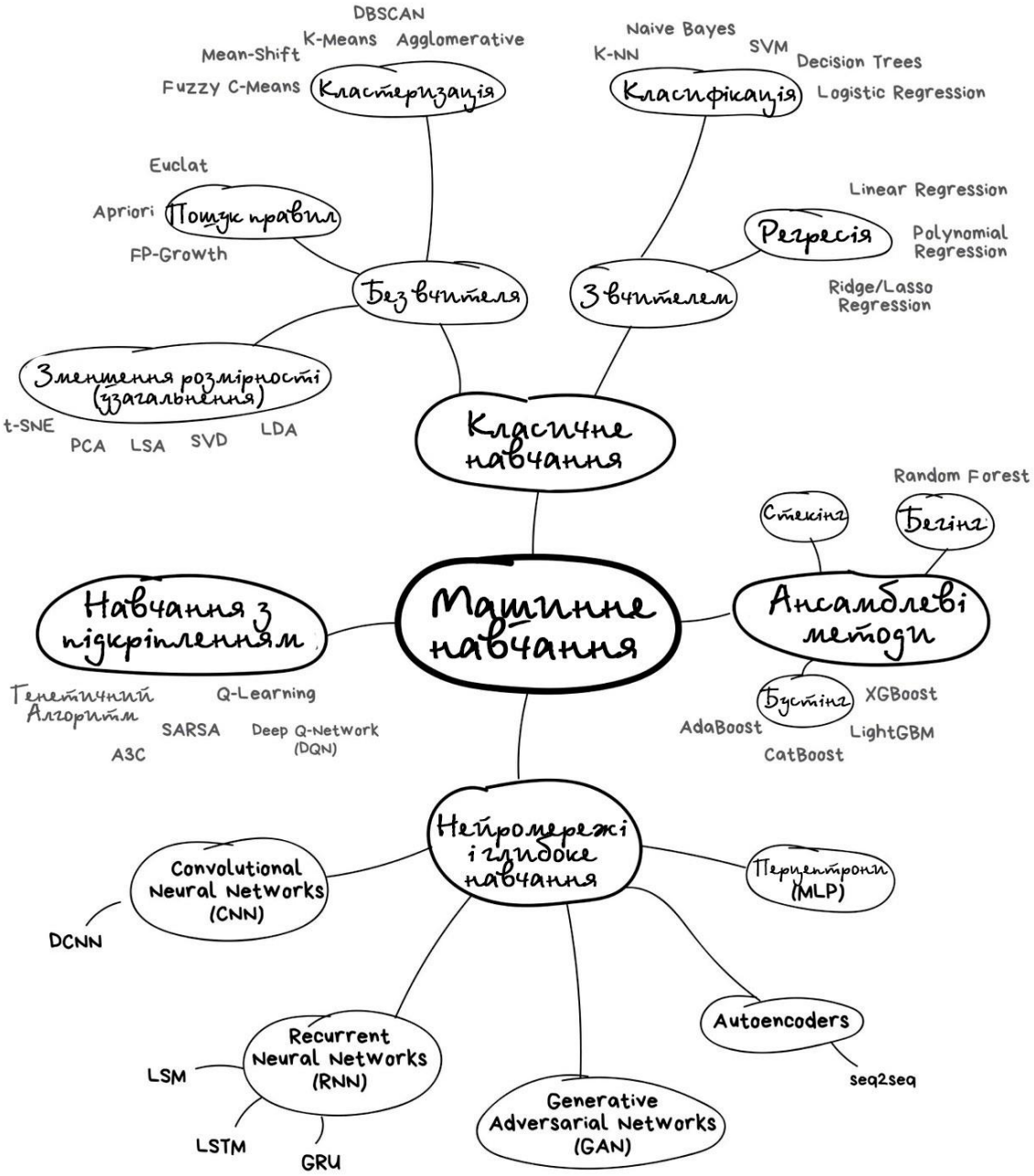
Ось що машини сьогодні вміють, а що не під силу навіть найбільш навченим.

МАШИНА МОЖЕ	МАШИНА НЕ МОЖЕ
Передбачати	Створювати нове

МАШИНА МОЖЕ	МАШИНА НЕ МОЖЕ
Запам'ятовувати	Різко порозумнішати
Відтворювати	Вийти за рамки завдання
Вибирати найкраще	Вбити всіх людей

2. КЛАСИФІКАЦІЯ МЕТОДІВ МАШИННОГО НАВЧАННЯ

Якщо ліньки читати - хоча б гляньте на картинку, буде корисно.



Класифікувати алгоритми можна десятком способів. Оберемо цей, тому що він здається найзручнішим для поясювання. Треба розуміти, що не буває так, щоб задачу

розв'язував лише один метод. Я буду згадувати відомі приклади застосувань, але пам'ятайте, що «син маминої подруги» все це може розв'язати нейромережами.

Розпочнемо з базового огляду. Сьогодні в машинному навчанні є лише чотири основні напрями.

3. КЛАСИЧНЕ НАВЧАННЯ

Перші алгоритми прийшли до нас ще в 1950-х роках з чистої статистики. Вони розв'язували формальні задачі - шукали закономірності в числах, оцінювали близькість точок в просторі і вираховували напрямки.

Сьогодні на класичних алгоритмах тримається добра половина інтернету. Коли ви зустрічаєте блок «Рекомендовані статті» на сайті, або банк блокує всі ваші гроші на картці після першої ж покупки кави за кордоном - це майже завжди справа рук одного з цих алгоритмів.

Зрозуміло, що великі корпорації люблять розв'язувати всі проблеми нейромережами. Це спричинено тим, що зайві 2% точності для них легко конвертуються в додаткові 2 мільярди прибутку. Решті ж варто включати голову. Коли задача розв'язується класичними методами, то дешевше реалізувати скільки-небудь корисну для бізнесу систему саме за їх допомогою, а вже потім думати про якесь покращення. А якщо ви не розв'язали задачу, то не розв'язати її на 2% краще вам не особливо допоможе.

Знаю кілька смішних історій, коли команда три місяці переписувала систему рекомендацій інтернет-магазину на більш точний алгоритм, і тільки потім зрозуміла, що покупці взагалі нею не користуються. Велика частина покупців просто приходить з пошукових систем.

При всій своїй популярності, класичні алгоритми настільки прості, що їх легко пояснити навіть дитині. Сьогодні вони як основи арифметики - застосовуються постійно, але дехто все одно став їх забувати.

3.1. Навчання з вчителем

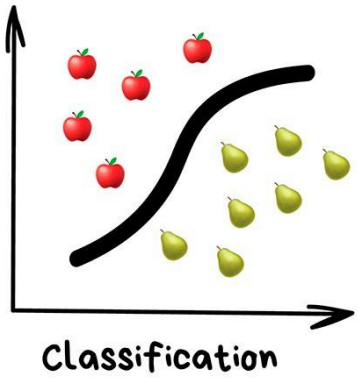
Класичне навчання люблять ділити на дві категорії — з вчителем і без. Часто можна зустріти їх англійські назви — Supervised і Unsupervised Learning.

У першому випадку у машини є якийсь учитель, який говорить їй як правильно. Розповідає, що на цій картинці кішка, а на цій собака. Тобто вчитель вже заздалегідь розділив всі дані на кішок і собак, а машина навчається на конкретних прикладах.

У навчанні без учителя, машині просто вивалюють купу фотографій тварин на стіл і кажуть «розберися, хто тут на кого схожий». Дані не розмічені, у машини немає вчителя, і вона намагається сама знайти будь-які закономірності. Про ці методи поговоримо нижче.

Очевидно, що з учителем машина навчиться швидше і точніше, тому в бойових задачах його використовують набагато частіше. **Ці задачі діляться на два типи: класифікація - передбачення категорії об'єкта, і регресія - передбачення місця на числовій прямій.**

Класифікація



«Поділяє об'єкти за заздалегідь відомою ознакою. Шкарпетки за кольорами, документи за мовами, музику за жанрами»

Сьогодні використовують для:

Спам-фільтри

Визначення мови

Пошук схожих документів

Ааналіз тональності

Розпізнавання рукописних букв і цифр

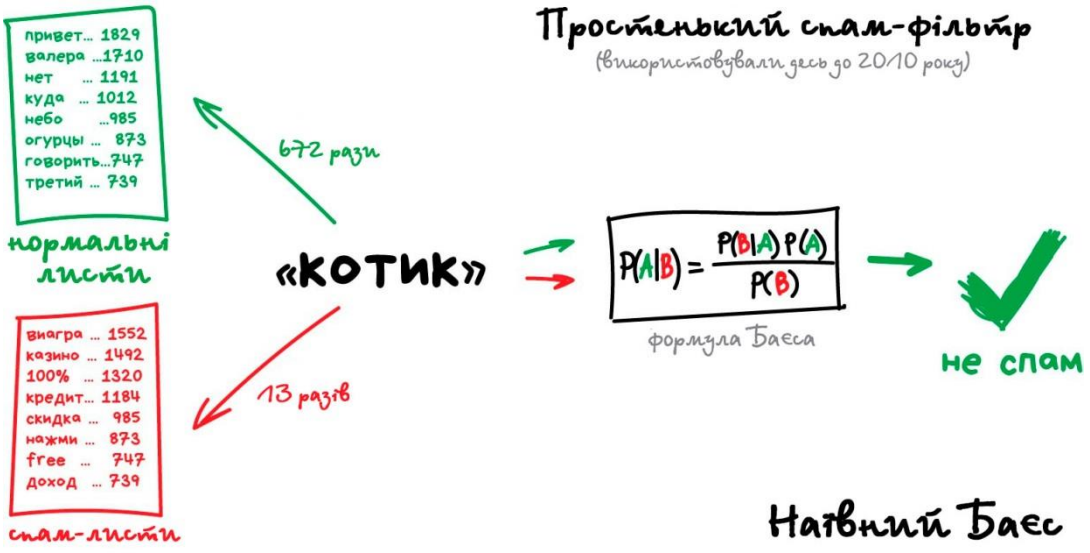
Визначення підозрілих транзакцій

Популярні алгоритми: Наївний Баєс, Дерева Рішень, Логістична Регресія, К-найближчих сусідів, Метод Опорних Векторів.

Класифікація речей - найпопулярніша задача у всьому машинному навчанні. Машина в ній як дитина, яка навчається розкладати іграшки: роботів в один ящик, танки в інший. Опа, а якщо це робот-танк? Що ж, час розплакатися і випасти в помилку.

Для класифікації завжди потрібен учитель - розмічені дані з ознаками і категоріями, які машина буде вчитися визначати за цими ознаками. Далі класифікувати можна що завгодно: користувачів за інтересами - так роблять алгоритмічні стрічки, статті за мовами і тематиками - важливо для пошукових систем, музику за жанрами - згадайте плейлисти, навіть листи у вашій поштовій скриньці.

Раніше всі спам-фільтри працювали на алгоритмі наївного Баєса. Машина вважала скільки разів слово «кредит» зустрічається в спамі, а скільки разів на нормальних листах. Множила ці дві ймовірності за формулою Баєса, складала результати всіх слів і бац, всім лежати, у нас машинне навчання!

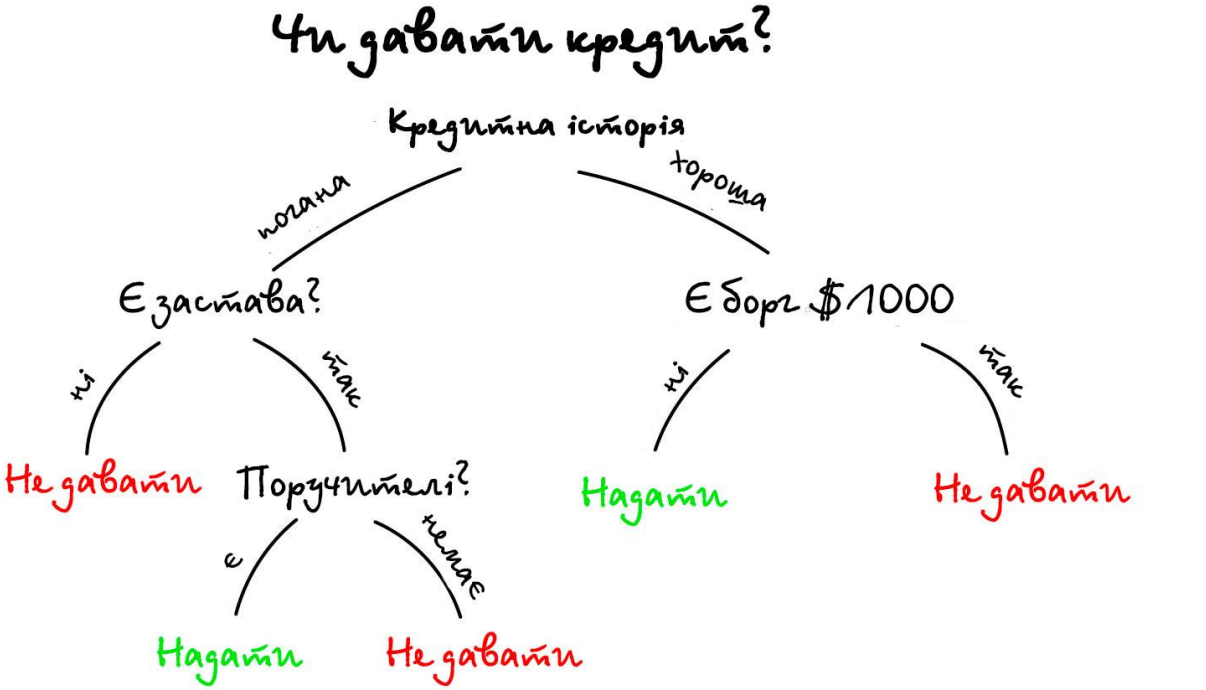


Пізніше спамери навчилися обходити фільтр Байеса, просто вставляючи в кінець листа багато слів з «хорошими» рейтингами. Метод отримав іронічну назву Отруєння Баєса, а фільтрувати спам стали іншими алгоритмами. Але метод назавжди залишився в підручниках як найпростіший, красивий і один з перших практично корисних.

Візьмемо інший приклад корисної класифікації. Ось берете ви кредит в банку. Як банку упевнитися повернете ви його чи ні? Дуже точно ніяк, але банк має тисячі профілів інших людей, які вже брали кредит до вас. Там вказано їхній вік, освіту, посаду, рівень зарплати та головне - хто з них повернув кредит, а з ким виникли проблеми.

Отож, всі здогадалися, де тут дані і який треба передбачити результат. Навчимо машину, знайдемо закономірності, отримаємо відповідь - питання не в цьому. Проблема в тому, що банк не може сліпо довіряти відповіді машини, без пояснень. Раптом збій, злі хакери або бухий адмін вирішив скриптик виправити.

Для цієї задачі придумали Дерево Рішень . Машина автоматично розділяє всі дані за запитаннями, відповідь на які «так» або «ні». Питання можуть бути не зовсім адекватними з точки зору людини, наприклад «зарплата позичальника більше, ніж 25934 гр.?», але машина придумує їх так, щоб на кожному кроці розбиття було найточнішим.



Дерево Рішень

Так отримується дерево питань. Чим вищий рівень, тим загальніше запитання. Потім навіть можна загнати їх аналітикам і вони понавидумують чому саме так.

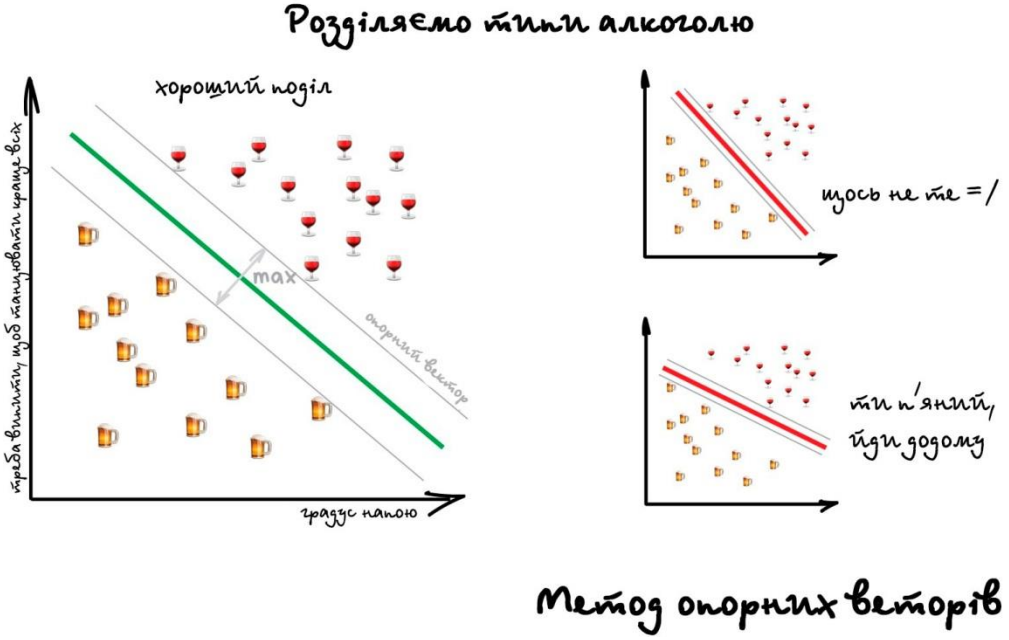
Дерева знайшли свою нішу в областях з високою відповідальністю: діагностиці, медицині, фінансах.

Два найпопулярніших алгоритми побудови дерев - CART і C4.5 .

У чистому вигляді дерева сьогодні використовують рідко, але ось їх ансамблі (про які буде нижче) лежать в основі великих систем і часто обскакують навіть нейромережі. Наприклад, коли ви задаєте запитання Ґуглу, то саме натовп дурних дерев біжить ранжувати вам результати.

Але найпопулярнішим методом класичної класифікації заслужено є Метод Опорних Векторів (SVM). Ним вже класифікували все, що тільки можна класифікувати: види рослин, обличчя на фотографіях, документи за тематиками... Багато років він був головною відповіддю на питання «який би мені взяти класифікатор».

Ідея SVM за своєю ідеєю проста - він шукає, як так провести дві прямі між категоріями, щоб між ними утворився найбільший зазор. На зображенні видно наочніше:



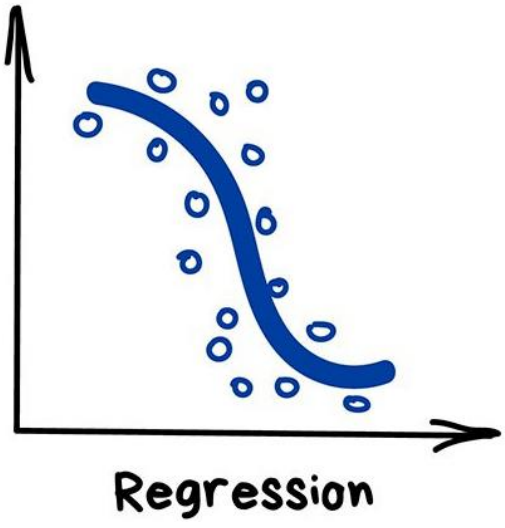
У класифікації є корисна зворотна сторона - пошук аномалій. Коли якась ознака об'єкта аж занадто не вписується в наші класи, ми яскраво підсвічуємо його на екрані. Зараз так роблять в медицині: комп'ютер підсвічує лікарю всі підозрілі області МРТ або виділяє відхилення в аналізах. На біржах таким же чином визначають нестандартних гравців, які швидше за все є інсайдерами. Навчивши комп'ютер «як правильно», ми автоматично отримуємо і зворотний класифікатор - як неправильно.

Сьогодні для класифікації все частіше використовують нейромережі, адже по-суті їх для цього і винайшли.

Правило таке: складніші дані - складніший алгоритм. Для тексту, цифр, табличок я б починав з класики. Там моделі менші, навчаються швидше і працюють зрозуміліше. Для картинок, відео та іншої незрозумілої бігдати - одразу б дивився в сторону нейромереж.

Років п'ять тому ще можна було зустріти класифікатор осіб на SVM, але сьогодні під цю задачу сотня готових сіток по інтернету валяються, чом би їх не взяти. А ось спам-фільтри як на SVM писали, так і не бачу сенсу зупинятися.

Регресія

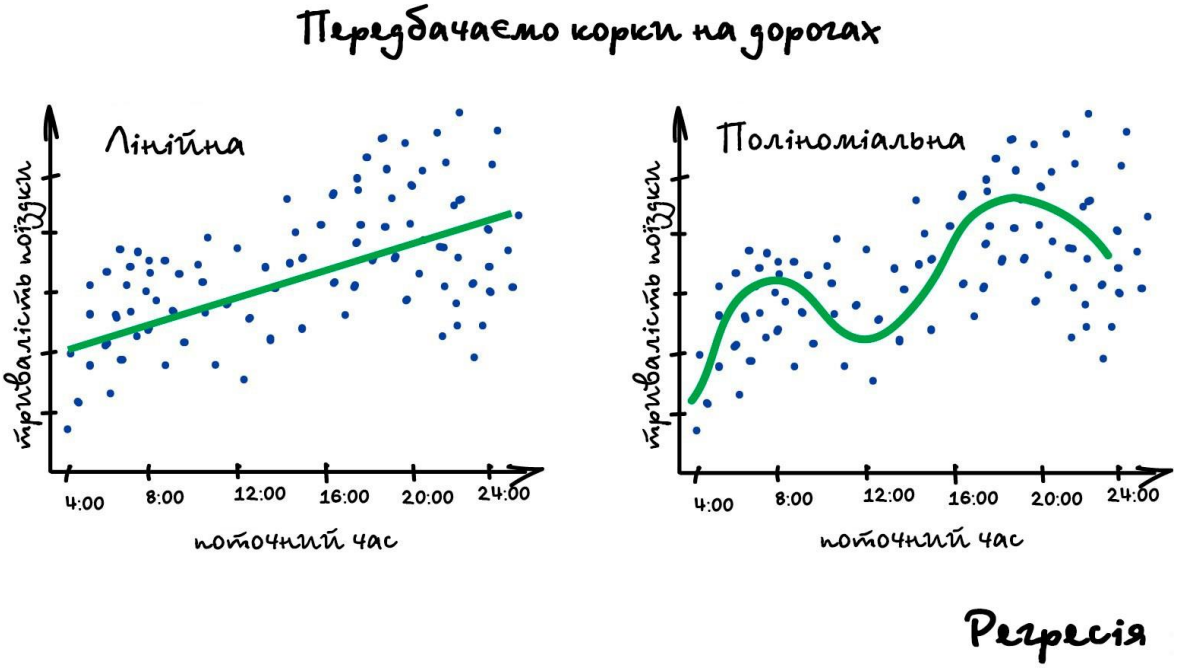


«Намалюй лінію уздовж моїх точок. Саме так, це машинне навчання»

Сьогодні використовують для:
Прогноз вартості цінних паперів
Аналіз попиту, обсягу продажів
Медичні діагнози
Будь-які залежності числа від часу
Популярні алгоритми: Лінійна або Поліноміальна Регресія

Регресія - та ж класифікація, тільки замість категорії ми передбачаємо число. Вартість автомобіля з його пробігом, кількість корків щодо часу доби, обсяг попиту на товар від зростання компанії і.т.д. На регресію ідеально лягають будь-які задачі, де є залежність від часу.

Регресію дуже люблять фінансисти і аналітики, вона вбудована навіть в Excel. Всередині все працює, знову ж таки, банально: машина тупо намагається намалювати лінію, яка в середньому відображає залежність. Правда, на відміну від людини з фломастером і вайтбордом, робить вона це з точністю - обчислюючи середню відстань до кожної точки і намагаючись всім догодити.



Коли регресія малює пряму лінію, її називають лінійною :))), коли криву - поліноміальною :)))))). Це два основні типи регресії, далі вже починаються рідкоземельні методи. Але так як в сім'ї не без виродка, є *Логістична Регресія*, яка насправді не регресія, а метод класифікації, від чого у всіх постійно плутанина. Не робіть так.

Схожість регресії і класифікації підтверджується ще й тим, що багато хто з класифікаторів, після невеликого тюнінгу, перетворюються в регресорів. Наприклад, ми можемо не просто дивитися до якого класу належить об'єкт, а запам'ятовувати, наскільки він близький - і ось, у нас регресія.

3.2. Навчання без вчителя

Навчання без вчителя (Unsupervised Learning) було винайдено пізніше, аж у 90-ті, і на практиці використовується рідше. Але бувають задачі, де у нас просто немає вибору.

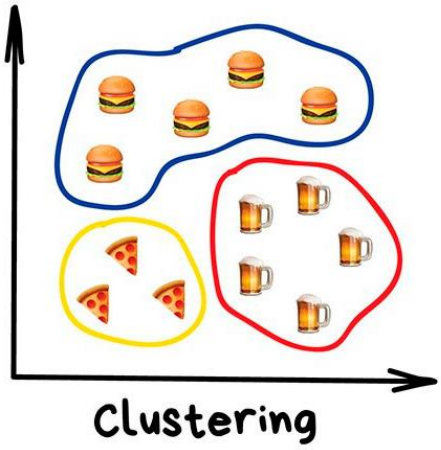
Розмічені дані - дорога рідкість. Але що робити якщо я хочу, наприклад, написати класифікатор автобусів - йти на вулицю, руками фотографувати мільйон Неопланів і підписувати де який? Так і все життя пройде, а у мене ще ігри в танчики не пройдено.

Коли немає міток, є надія на капіталізм, соціальне розшарування і мільйон китайців з сервісів типу Яндекс. Толока, які готові робити для вас що завгодно за п'ять центів. Так зазвичай і роблять на практиці. А ви думали де Яндекс бере більшість своїх датасетів?

Або можна спробувати навчання без учителя. Хоча, відверто кажучи, зі своєї практики я не пам'ятаю, щоб десь воно спрацювало добре.

Навчання без вчителя, все ж, частіше використовують як метод аналізу даних, а не як основний алгоритм. Спеціальний юзер з дипломом котрогось НУ вкидає туди купу сміття і спостерігає. Кластери є? Залежності з'явилися? Ні? Ну що ж, продовжуй, праця облагороджує. Ти ж хотів працювати в датасаєнсі.

Кластеризація



«Розділяє об'єкти за невідомою ознакою. Машина сама вирішує як краще»

Сьогодні використовують для:

Сегментація ринку (типів покупців, лояльності)

Об'єднання близьких точок на карті

Стиснення зображень

Аналіз і розмітки нових даних

Детектори аномальної поведінки

Популярні алгоритми: Метод К-середніх , Mean-Shift , DBSCAN

Кластеризація - це класифікація, але без заздалегідь відомих класів. Вона сама шукає схожі об'єкти та об'єднує їх в кластери. Кількість кластерів можна задати заздалегідь або довірити це машині. Схожість об'єктів машина визначає за тими ознаками, які ми їй розмітили - у кого багато схожих характеристик, тих давай в один клас.

Відмінний приклад кластеризації - маркери на картах в Інтернеті. Коли ви шукаєте всі крафтові бари у Львоі, машині доводиться групувати їх в кружечки з циферками, інакше браузер зависне в потугах намалювати мільйон маркерів.

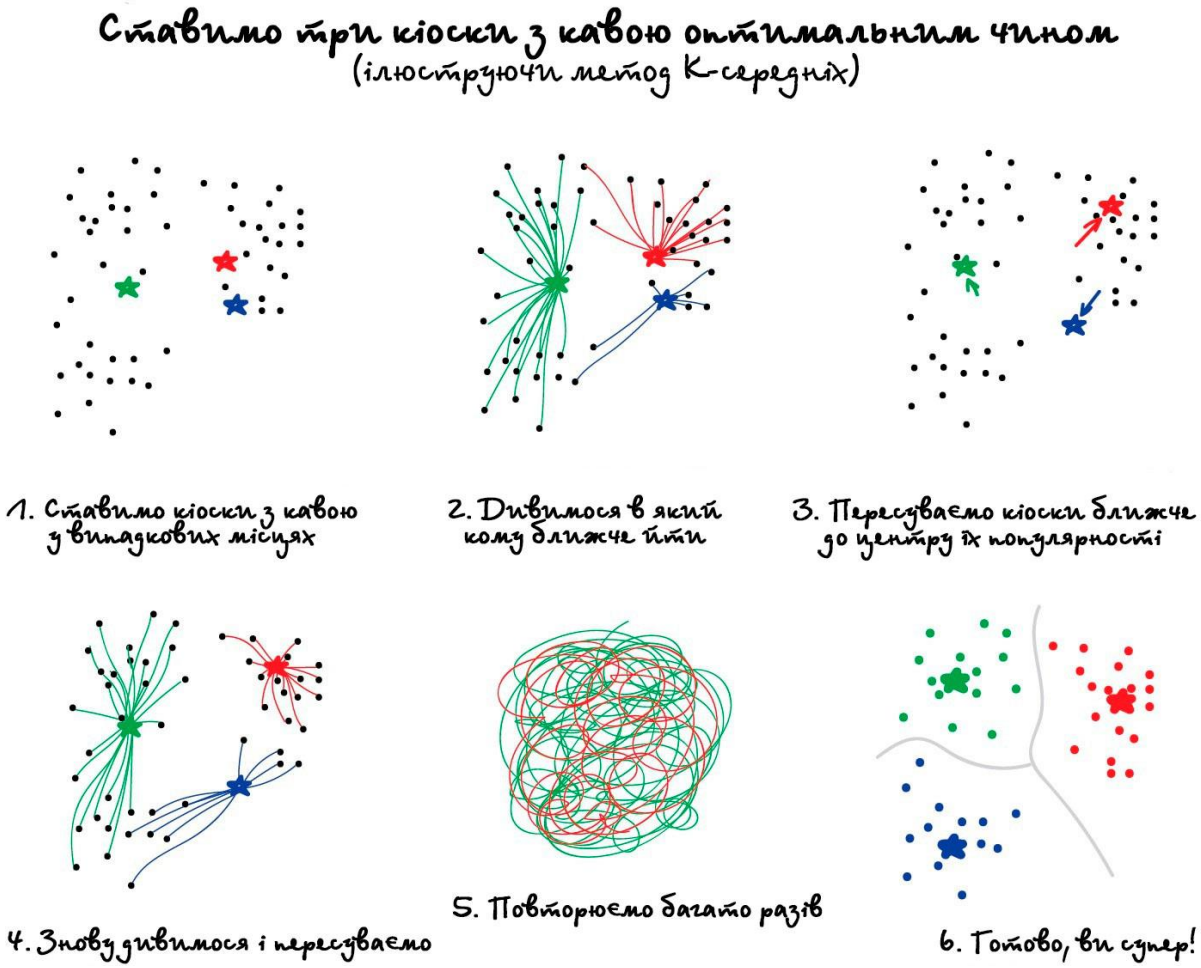
Більш складні приклади кластеризації можна згадати в додатках iPhoto або Google Photos, які знаходять особи людей на фотографіях і групують їх у альбоми. Додаток не знає як звуть ваших друзів, але може відрізнити їх за характерними рисами обличчя. Типова кластеризація.

Правда для початку їм доводиться знайти ці самі «характерні риси», а це вже тільки з учителем.

Стиснення зображень - ще одна популярна проблема. Зберігаючи картинку в PNG, ви можете встановити палітру, скажімо, в 32 кольори. Тоді кластеризація знайде всі «приблизно червоні» пікселі зображення, вирахує з них «середній червоний по лікарні» і замінить всі червоні на нього. Менше кольорів - менший файл.

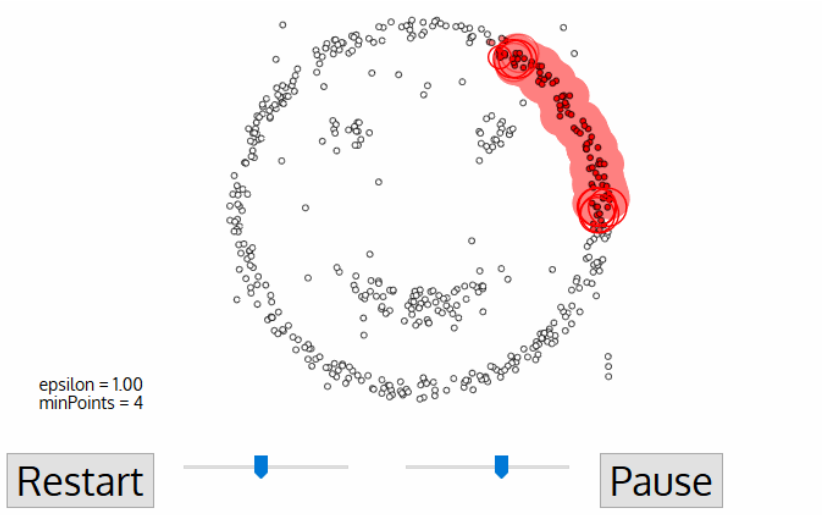
Проблема тільки, як бути з кольорами типу Cyan - ось він ближче до зеленого чи синього? Тут нам допоможе популярний алгоритм кластеризації - Метод К-середніх (K-Means) . Ми випадковим чином кидаємо на палітру кольорів наші 32 точки, називаючи їх центроїдами. Всі інші точки відносимо до найближчого центроїду від них - виходять так би мовити сузір'я з найближчих кольорів. Потім рухаємо центроїд в центр свого сузір'я і

повторюємо поки центроїди не перестануть рухатися. Кластери виявлені, стабільні і їх рівно 32 як і треба було.



Шукати центроїди зручно і просто, але в реальних задачах кластери можуть бути зовсім не круглої форми. Ось ви геолог, якому потрібно знайти на карті схожі за структурою гірські породи - ваші кластери не тільки будуть вкладені одна в одну, але ви ще й не знаєте скільки їх взагалі вийде.

Хитрим завданням - хитрі методи. DBSCAN , наприклад. Він сам знаходить скупчення точок і будує навколо кластери. Його легко зрозуміти, якщо уявити, що точки - це люди на площі. Знаходимо трьох людей, які знаходяться близько одна до одної, і пропонуємо їм взятися за руки. Потім вони починають брати за руку тих, кого можуть досягти. Так по ланцюжку, поки ніхто більше не зможе взяти когось за руку - це і буде перший кластер. Повторюємо, поки не поділимо всіх. Ті, кому взагалі нікого брати за руку - це аномалії, викидаємо. В динаміці виглядає досить красиво:

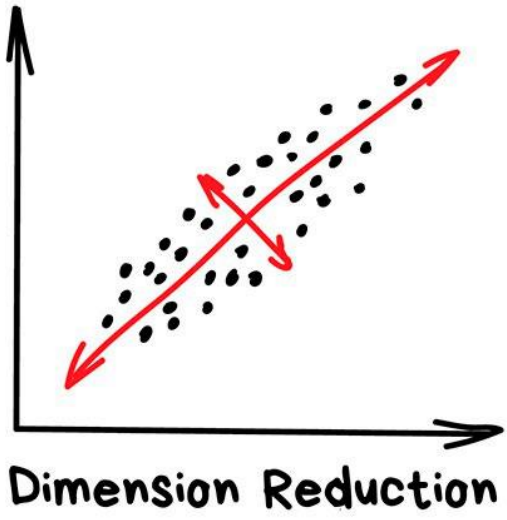


Тим, хто цікавиться кластеризацією, рекомендую статтю [The 5 Clustering Algorithms Data Scientists Need to Know]

Як і класифікація, кластеризація також може використовуватися як детектор аномалій. Поведінка користувача після реєстрації різко відрізняється від нормального? Заблокувати його і створити тікет саппорту, щоб перевірили бот це чи ні. При цьому нам навіть не треба знати, що є «нормальна поведінка» - ми просто вивантажуємо всі дії користувачів в модель, і нехай машина сама розбирається хто тут нормальний.

Працює такий підхід, в порівнянні з класифікацією, не дуже. Але за спробу не б'ють, раптом вийде.

Зменшення Розмірності (Узагальнення)



«Збирає конкретні ознаки в абстракції вищого рівня»

Сьогодні використовують для:

Рекомендаційні Системи (★)

Красиві візуалізації

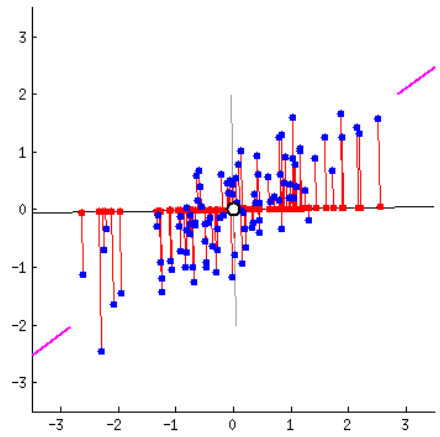
Визначення тематики та пошуку схожих документів

Аналіз фейковий зображень

Ризик-менеджмент

Популярні алгоритми: Метод головних компонент (PCA), Сингулярне розкладання (SVD), Латентне розміщення Діріхле (LDA), Латентно-семантичний аналіз (LSA, pLSA, GLSA), t-SNE (для візуалізації)

Спочатку це були методи хардкорних Data Scientist'ів, яким вивантажували дві фури цифр і говорили знайти там що-небудь цікаве. Коли просто будувати графіки в екселі вже не допомагало, вони придумали напружити машини шукати закономірності замість них. Так у них з'явилися методи, які назвали Dimension Reduction або Feature Learning.

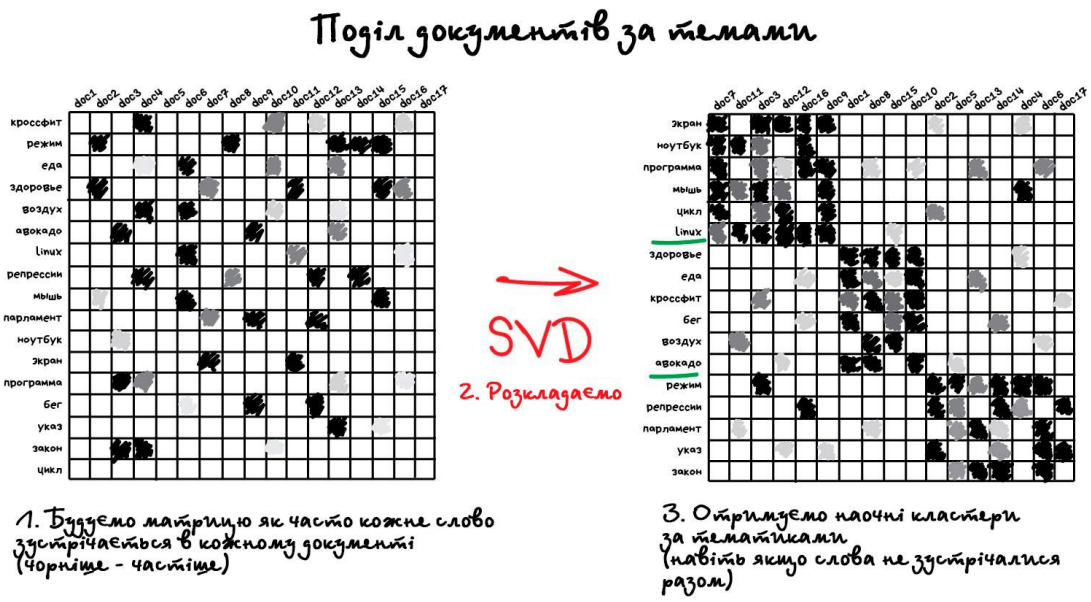


Для нас практична користь їх методів в тому, що ми можемо об'єднати декілька ознак в одну і отримати абстракцію. Наприклад, собаки з трикутними вухами, довгими носами і великими хвостами об'єднуються в корисну абстракцію «вівчарки». Очевидно, що ми втрачаємо інформацію про конкретних вівчарок, але нова абстракція по-любому корисніша цих зайвих деталей. Плюс, навчання на меншій кількості розмірностей йде сильно швидше.

Інструмент на диво добре підійшов для визначення тематик текстів (Topic Modelling). Ми змогли абстрагуватися від конкретних слів до рівня смислів навіть без залучення вчителя зі списком категорій. Алгоритм назвали Латентно-семантичний аналіз (LSA), і його ідея була в тому, що частота появи слова в тексті залежить від його тематики: в наукових статтях більше технічних термінів, в новинах про політику - імен політиків. Так, ми могли б просто взяти всі слова з статей і кластеризувати, як ми робили з кіосками вище, але тоді ми б втратили всі корисні зв'язки між словами, наприклад, що батарейка і акумулятор означають одне і те ж в різних документах.

Точність такої системи - повне дно, навіть не намагайтеся.

Потрібно якось об'єднати слова і документи в одну ознаку, щоб не втрачати ці приховані (латентні) зв'язку. Звідси і з'явилася назва методу. Виявилося, що Сингулярне розкладання (SVD) легко справляється з цим завданням, виявляючи для нас корисні тематичні кластери зі слів, які зустрічаються разом.



Латентно-семантичний Аналіз (LSA)

Для розуміння рекомендую статтю Як зменшити кількість вимірювань і отримати з цього користь, а практичне застосування добре описано в статті Алгоритм LSA для пошуку схожих документів .

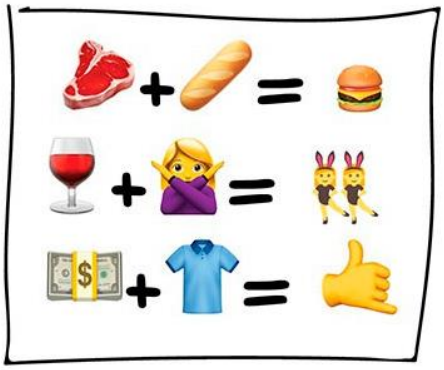
Інше мега-популярне застосування методу зменшення розмірності знайшли в рекомендаційних системах і колаборативній фільтрації. Виявилося, що якщо абстрагувати ними оцінки користувачів фільмів, виходить непогана система рекомендацій кіно, музики, ігор і чого завгодно взагалі.

Отримана абстракція буде важко зрозуміла мозком, але коли дослідники почали пильно розглядати нові ознаки, вони виявили, що якісь з них явно корелюють з віком користувача (діти частіше грали в майнкрафт і дивилися мультфільми), інші з певними жанрами кіно, а треті взагалі з синдромом пошуку глибокого сенсу життя.

Машина, яка не знала нічого, крім оцінок користувачів, змогла дістатися до таких високих матерій, навіть не розуміючи їх. Достойно. Далі можна проводити соцопитування і писати дипломні роботи про те, чому бородаті дядьки люблять дегенеративні мультики.

На цю тему є непогана лекція - Як працюють рекомендаційні системи

Пошук правил (асоціація)



Assiciation Rule Learning

«Шукає закономірності в потоці замовлень»

Сьогодні використовують для:
Прогноз акцій і розпродажів
Аналіз товарів, які купують разом
Розташування товарів на полицях
Аналіз патернів поведінки на веб-сайтах
Популярні алгоритми: Apriori, Euclat, FP-growth

Сюди входять всі методи аналізу продуктових кошиків, стратегій маркетингу і інших послідовностей.

Припустимо, покупець бере в дальньому кутку магазину пиво і йде на касу. Чи варто ставити на його шляху горішки? Чи часто люди беруть їх разом? Горішки з пивом, напевно так, але які ще товари купують разом? Коли ви власник мережі гіпермаркетів, відповідь для вас не завжди очевидна, але одне тактичне поліпшення в розташуванні товарів може принести гарний прибуток.

Це ж стосується інтернет-магазинів, де завдання ще цікавіше - за яким товаром покупець повернеться наступного разу?

З незрозумілих причин, пошук правил - найбільш погано продумана категорія серед всіх методів навчання. Класичні способи полягають в тупому переборі пар всіх куплених товарів за допомогою дерев або множин. Самі алгоритми працюють наполовину - можуть шукати закономірності, але не вміють узагальнювати або відтворювати їх на нових прикладах.

У реальності кожен великий ритейлер пиляє свій велосипед і жодних особливих проривів в цій області не зустрічається. Максимальний рівень технологій тут - створити систему рекомендацій, як в пункті вище. Хоча може я просто далекий від цієї області?