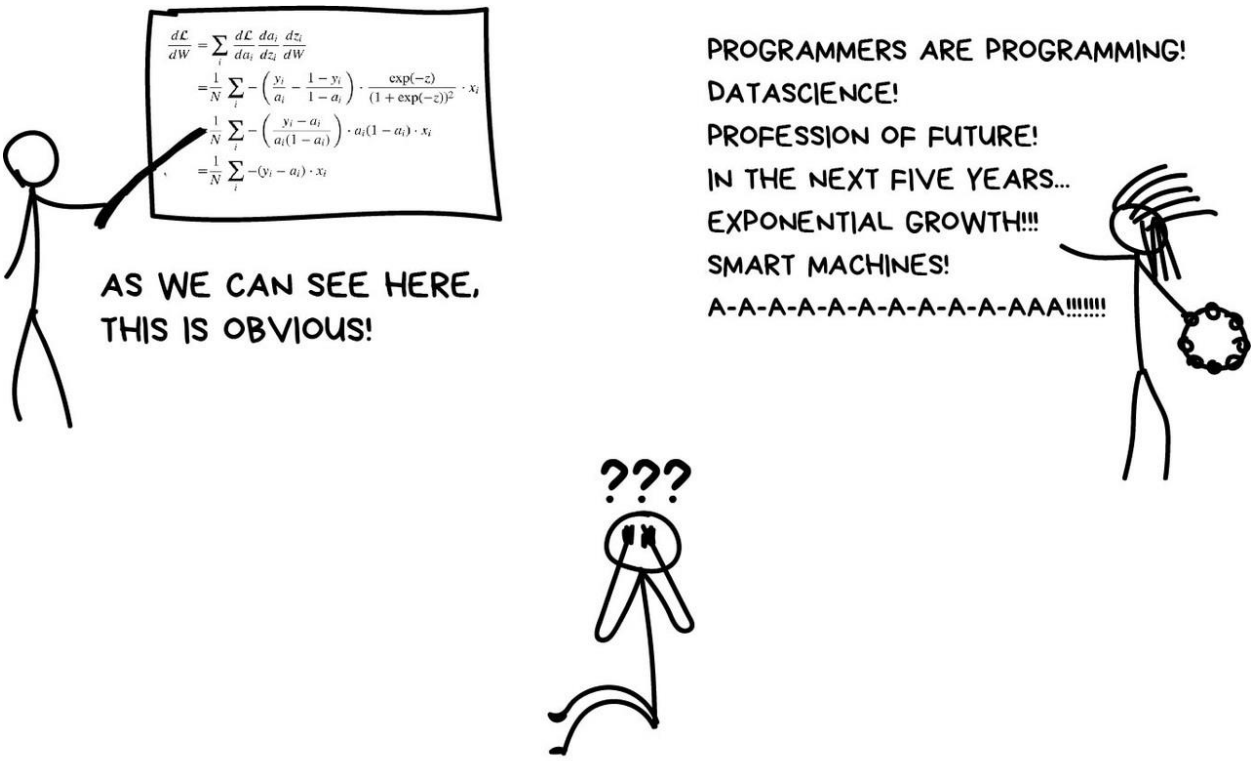
Giải thích dễ hiểu về học máy

Machine Learning (học máy, viết tắt là ML) cũng giống như sex trong trường học vậy. Ai cũng nói về nó, một vài người thực sự làm nó, và chỉ có giáo viên của bạn mới dạy nó. Nếu bạn đã từng thử tìm đọc các bài báo trên mạng về Machine Learning, bạn gần như sẽ tìm được hai kiểu bài: một là các bài nghiên cứu học thuật với hàng tá lý thuyết khô khan (tôi thậm chí còn không thể đọc hết nửa bài nữa), hai là những bài giật tít câu view về **Artificial Intelligence** (trí tuệ nhân tạo, viết tắt là AI), sự thần thông quảng đại của **Data Science** (khoa học dữ liệu), và các công việc mới xuất hiện trong kỉ nguyên 4.0…
Vậy nên tôi quyết định viết, một bài viết mà tôi ước là nó đã phải tồn tại từ lâu rồi. Một bài giới thiệu đơn giản về Machine Learning cho những ai quan tâm và muốn tìm hiểu về nó. Chỉ có những ví dụ thực tế, những giải pháp thực tế, viết bằng ngôn ngữ bình dân không lý thuyết cao siêu gì cả. Một bài viết dành cho tất cả mọi người. Dù bạn là một lập trình viên hay một nhà quản lý.
Ok, bắt đầu thôi!



Hai loại bài viết bạn thường gặp nếu tìm kiếm “Machine Learning” trên Google

Học máy – tại sao máy móc lại phải học?

Đây là Hiệp. Hiệp muốn mua một chiếc ô tô. Hiệp lấy giấy bút ra và tính xem mình sẽ phải tiết kiệm bao nhiêu tiền mỗi tháng để mua được ô tô. Sau khi lướt qua vài cái quảng cáo ô tô trên mạng, Hiệp biết một chiếc xe mới sẽ có giá khoảng 500 triệu, chiếc nào đã đi được một năm thì khoảng 450 triệu, hai năm thì khoảng 400 triệu, và cứ như thế.

Là một người có khả năng phân tích khá, Hiệp bắt đầu nhìn ra một quy luật (pattern): giá của một chiếc ô tô phụ thuộc vào số tuổi của xe, và cứ đi được một năm thì giá xe giảm 50 triệu, nhưng sẽ không hạ xuống dưới 100 triệu.

Nói theo cách của Machine Learning, Hiệp vừa mới phát minh ra **regression** (*tiếng Việt là hồi quy*) – anh ấy dự đoán giá ô tô dựa trên dữ liệu trong quá khứ. Chúng ta vẫn thường làm điều đó, khi cần ước lượng giá của một chiếc iPhone like-new trên Chợ Tốt, hoặc khi tính xem cần mua bao nhiêu rau cho một bữa lẩu 5 người. Mỗi người ăn 2 lạng? Hay 3 lạng?

Sẽ thật tuyệt vời nếu như có một công thức đơn giản cho tất cả các bài toán trong thế giới này. Đặc biệt là cho bữa lẩu. Nhưng rất tiếc là không có.

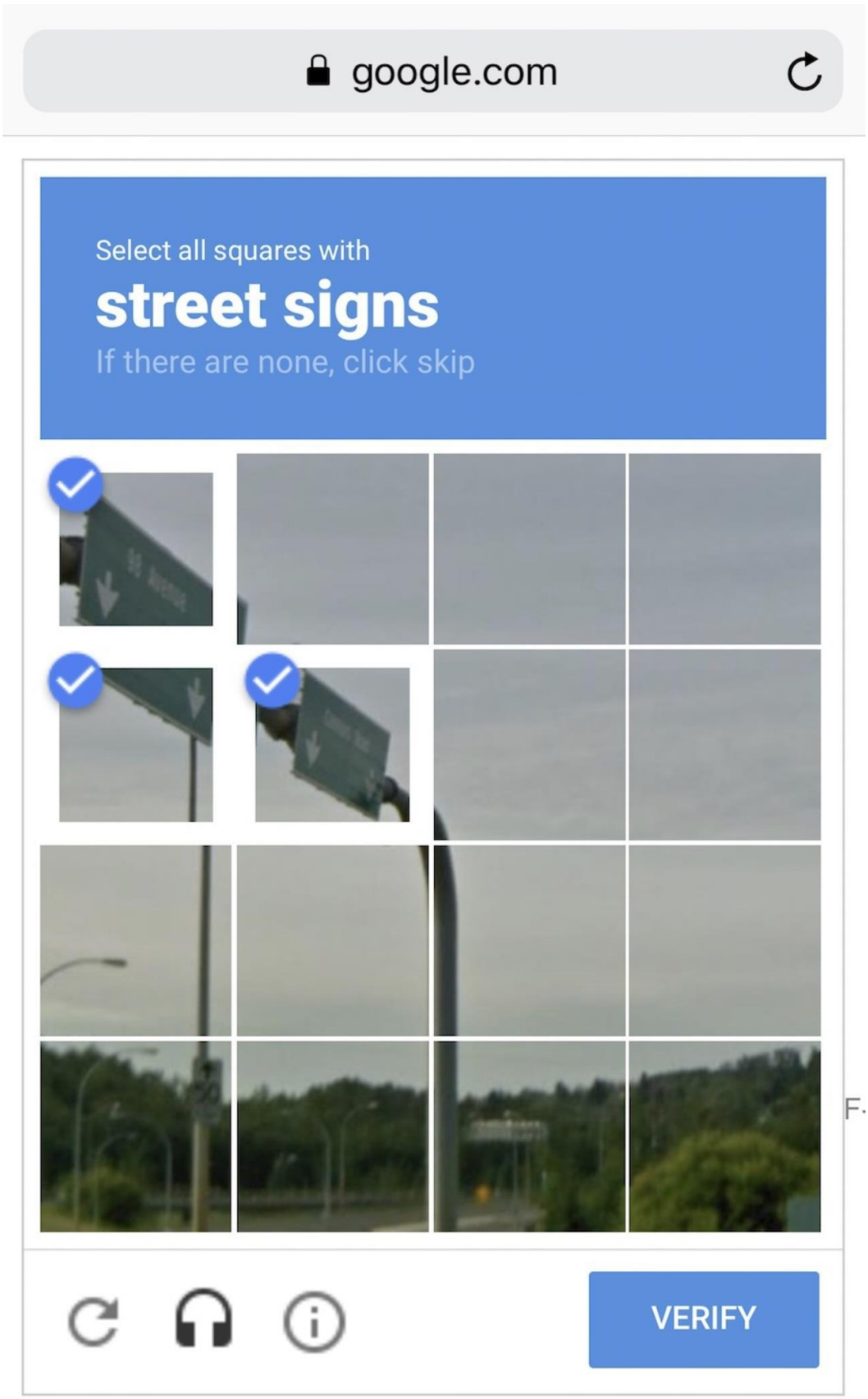
Quay lại vụ mua ô tô. Vấn đề ở đây là, có rất nhiều yếu tố ảnh hưởng đến giá của một chiếc ô tô, ví dụ như hãng sản xuất, thời gian sản xuất khác nhau, các điều kiện kỹ thuật, công nghệ

mới… Một người bình thường như Hiệp không thể nhớ được hết các thông tin đó trong đầu để mà tính ra kết quả được. Và chúng ta cũng vậy.
Con người chúng ta rất ngốc nghếch và lười biếng, vậy nên chúng ta mới cần máy móc làm toán hộ mình. Vậy thử nhờ máy móc xem sao. Hãy thử cho chúng dữ liệu và yêu cầu chúng tìm ra những "quy luật ngầm" (hidden pattenrs) để định giá ô tô.
Ồ, và nó làm được này! May mắn là máy móc xử lý những công việc kiểu này tốt hơn con người rất nhiều, kể cả khi chúng ta xem xét cẩn thận tất cả các mối quan hệ giữa các yếu tố kể trên.
Và từ đó, Machine Learning ra đời.

Ba "nguyên liệu" chính của Machine Learning

Bỏ qua mọi quảng cáo nhảm nhí về AI, thì mục tiêu duy nhất của ML là dự đoán các kết quả dựa trên dữ liệu đầu vào. Tất cả các nhiệm vụ của ML có thể được diễn giải theo cách đó, nếu không thể thì đó không phải một bài toán giải quyết được bằng ML.
Bạn càng có nhiều mẫu thử (*samples, ý nói dữ liệu thực tế*) đa dạng, thì máy móc càng dễ để tìm ra các quy luật liên quan và dự đoán kết quả. Do vậy, chúng ta cần 3 nguyên liệu chính để "dạy" máy móc học:

– Dữ liệu: Bạn muốn phát hiện spam? Hãy thu thập các tin nhắn spam trong thực tế. Bạn muốn dự đoán giá cổ phiếu? Hãy thu thập lịch sử giá. Bạn muốn biết người dùng thích gì? Hãy phân tích hành vi của họ trên Facebook (đừng, Mark, dừng ngay việc đó lại đi, quá đủ rồi!). Dữ liệu càng đa dạng, kết quả càng chính xác. Có những trường hợp cần ít nhất là hàng chục nghìn hàng dữ liệu *(hàng, cột trong bảng biểu Excel ấy).*
Có hai cách để thu thập dữ liệu – thủ công hoặc tự động. Cách thủ công có thể gây ra ít sai sót hơn nhưng lại tốn thời gian, nên nhìn chung thu thập kiểu thủ công khá là tốn kém.
Cách tự động thì rẻ hơn – bạn thu thập tất cả mọi thứ bạn có thể tìm thấy và hi vọng có được kết quả tốt nhất.
Những công ty thông minh như Google tận dụng chính người dùng để dán nhãn dữ liệu miễn phí cho họ. Bạn còn nhớ cái mã captcha mà vợ chồng chị Dậu dùng để bán cái Tí không? Đùa thôi, cái mà hay hiện lên khi bạn click vào ô download, rồi yêu cầu bạn phải *"Chọn những hình vuông chứa biển báo giao thông trong các hình dưới đây"* ấy?


google.com
Select all squares with
street signs
If there are none, click skip
VERIFY

Đó chính xác là những gì Google đang làm đấy. Sử dụng lao động miễn phí! Thực ra, nếu ở vị trí của họ, chắc tôi còn bắt bạn xem nhiều mã captcha hơn nữa ấy chứ. Nhưng, từ từ đã… Rất khó để thu thập được một bộ dữ liệu tốt *(theo thuật ngữ chuyên ngành là dataset – hay tập dữ liệu)*. Chúng quan trọng đến nỗi mà nhiều công ty có thể đồng ý tiết lộ về thuật toán của họ, còn dữ liệu thì không.

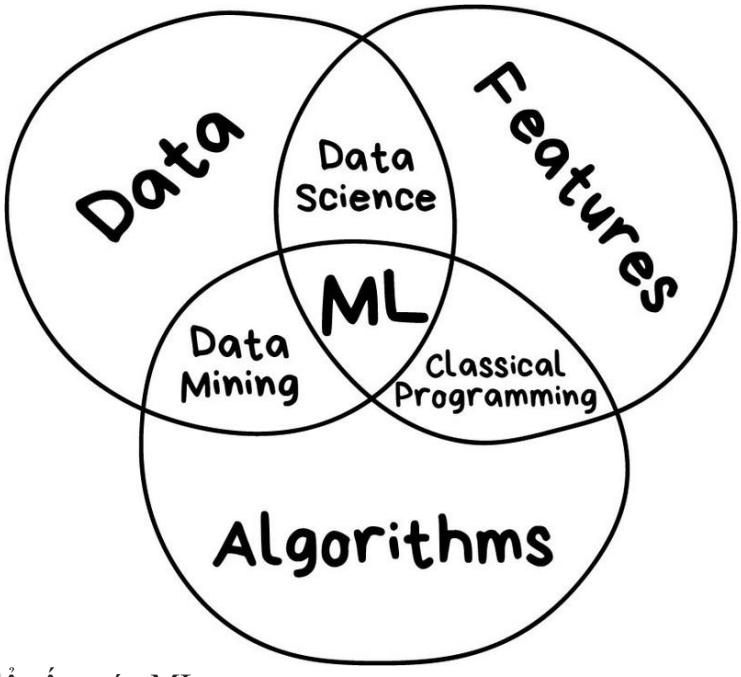
– Tiếp đến là **các đặc trưng (features)**: còn được biết đến với tên gọi là tham số *(parameters)* hoặc các biến *(variables)*. Chúng giống như các yếu tố mà máy móc cần phải xem xét khi "học" vậy. Ví dụ, không cần tính toán bạn cũng biết là một chiếc ô tô được sử dụng càng nhiều thì khi bán lại càng mất giá, vậy nên **giá** của một chiếc ô tô cũ phụ thuộc một phần vào **số km đã đi được** của nó đúng không? Đó chính là một đặc trưng đấy. Các đặc trưng khác có thể kể đến như túi tiền của người ăn với bài toán đi chợ cho bữa lẩu, hay giới tính, với bài toán phân tích sở thích người dùng…
Khi dữ liệu được lưu trữ trong các bảng biểu *(bảng Excel là một ví dụ)*, các đặc trưng đơn giản là tên các cột trong bảng thôi. Nhưng nếu dữ liệu của bạn là 100 Gb ảnh mèo thì sao? Chúng ta không thể coi mỗi pixel ảnh là một đặc trưng được. Đó là lí do tại sao bước chọn lọc ra các đặc trưng thích hợp tốn nhiều thời gian hơn các công việc khác trong một quy trình giải quyết vấn đề bằng học máy. Đó cũng là các nguyên nhân chính gây ra các sai sót trong tính toán. Con người mà, thường rất chủ quan. Chúng ta thường chọn những feature mà mình thích, hoặc mình cho là quan trọng. Vậy nên, bỏ đi đừng làm người.

– Cuối cùng là **thuật toán**: Tất nhiên rồi. Mỗi bài toán đều có cách giải riêng của nó. Phương pháp bạn chọn sẽ ảnh hưởng đến **độ chính xác** *(precision)* , **độ hiệu quả** *(performance),* và quy mô của **mô hình cuối** *(final model)***.** Có một điều quan trọng là: nếu dữ liệu của bạn không phù hợp *(ví dụ: quá ít, hoặc không đủ đa dạng, hoặc đơn giản là dữ liệu không đủ liên quan đến thứ cần dự đoán)* thì dù là thuật toán tốt nhất cũng vô dụng. Đầu xuôi thì đuôi mới lọt. Vậy nên, đừng quá chú trọng vào độ chính xác, hãy thu thập đủ và đúng dữ liệu trước đã.
Giải thích một chút:

– Thuật toán (Algorithm): hiểu nôm na là các bước mà máy tính, hay con người thực hiện để giải quyết một vấn đề nào đó. Ví dụ, để nấu một bát mì bạn có thể: bước 1: xé vỏ mì -> bước 2: đun nước -> bước 3: thả mì vào nồi -> bước 4: thả rau vào nồi -> bước 5: đợi mì và rau chín -> bước 6: vớt mì ra bát. Vậy là xong một thuật toán để nấu mì. Giờ giả sử bạn hoán đổi thứ tự của bước 3 và bước 4, các bước còn lại giữ nguyên, bạn sẽ thu được một thuật toán khác, vẫn để nấu mì, nhưng kết quả bát mì sẽ khác với việc bạn thực hiện thuật toán ban đầu, có thể là rau sẽ chín kĩ hơn. Với máy tính thì ngoài việc thu được kết quả khác, chúng ta còn quan tâm đến thời gian và tài nguyên tiêu tốn để chạy một thuật toán nữa, cùng để thực hiện một nhiệm vụ, thuật toán A có thể chạy nhanh hơn thuật toán B, nhưng sẽ tốn nhiều bộ nhớ hơn, chẳng hạn.

– Precision và accuracy: cùng là độ chính xác, nhưng trong học máy chúng là hai khái niệm khác nhau. Bạn có thể tìm hiểu thêm ở đây: https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9

– Mô hình cuối: hiểu nôm na là mô hình mà máy tính đã "học" được để giải quyết bài toán đó, ví dụ với bài toán dự đoán giá ô tô thì 1 mô hình có thể là: giá ô tô = giá mới – 0.2 x tuổi đời của xe – 0.05 x số km đã đi được – 0.01x số vết xước trên xe, chẳng hạn.

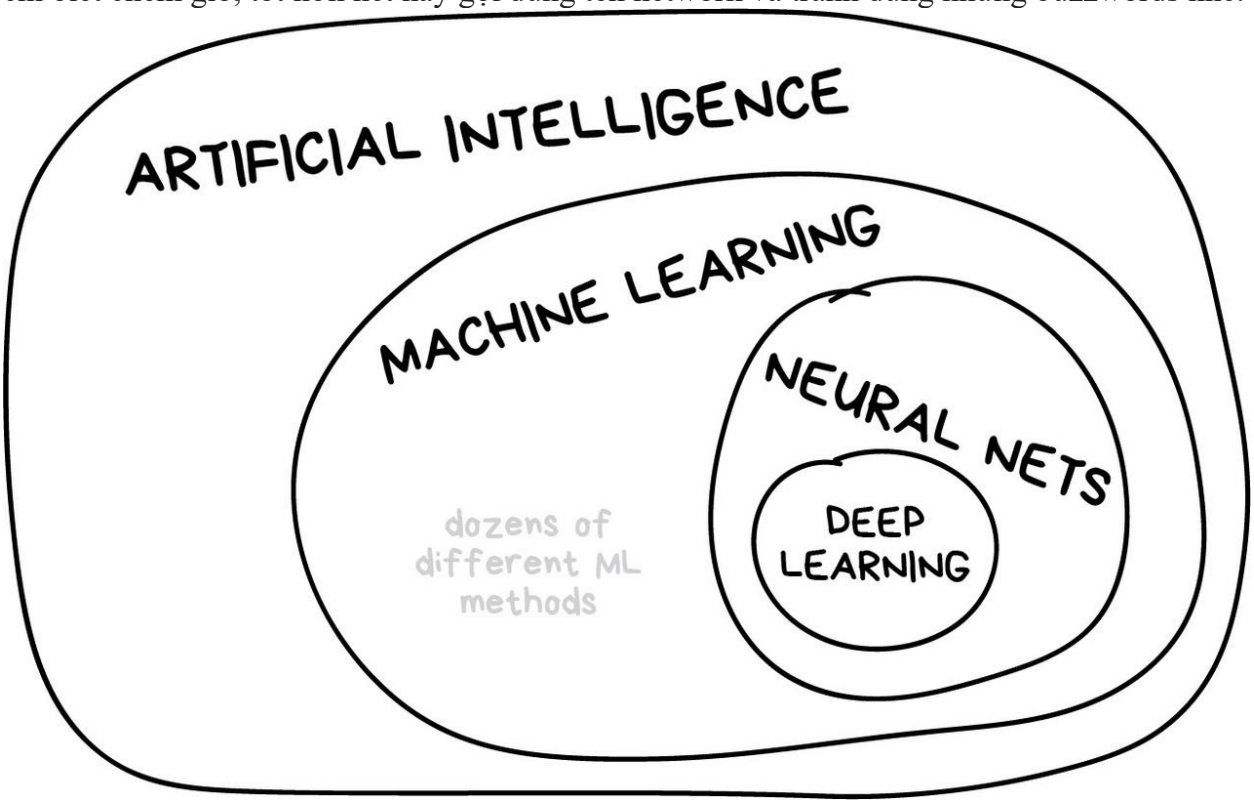


Ba "nguyên liệu" để nấu món ML

Học tập và trí thông minh

Có lần tôi đọc được một bài báo với tiêu đề "Liệu mạng neuron có thay thế được học máy không?" trên một trang báo lá cải. Những tay nhà báo thường xuyên gọi thuật toán **linear regression** *(hồi quy tuyến tính – thuật toán cơ bản nhất trong học máy hoặc thống kê học)* với một cái tên mỹ miều là trí tuệ nhân tạo, hoặc tệ hơn, SkyNet. Để mị giới thiệu cho mà nghe:
– **Trí tuệ nhân tạo** *(Artificial Intelligence):* là tên gọi của cả một lĩnh vực nghiên cứu, tương tự như sinh học hay hoá học ấy.
– **Học máy** *(Machine Learning)*: là một phần của trí tuệ nhân tạo. Một phần quan trọng, nhưng không phải tất cả.
– **Mạng neuron** *(Neural Networks – NN):* là một trong những nhóm thuật toán học máy. Một anh chàng đẹp trai học giỏi trong lớp học ML đấy, nhưng trong lớp vẫn có những anh chàng khác học giỏi không kém *(các mạng neurons sẽ được giải thích kỹ hơn ở các phần sau).*
– **Học sâu** *(Deep Learning- DL)*: một phương pháp hiện đại để xây dựng, huấn luyện và ứng dụng các mạng neuron. Cơ bản là, nó là một kiểu kiến trúc mới. Thực tế là không ai tách riêng deep learning với các kiểu network truyền thống cả. Chúng ta thậm chí vẫn sử dụng các thư viện chung cho chúng đấy thôi *(trong lập trình, thư viện giống như một nơi chứa các dòng code được viết sẵn để bạn có thể ứng dụng vào sản phẩm của bạn mà không cần code lại vậy, trong học máy cũng có các thư viện được viết riêng để lập trình viên có thể dùng*

luôn mà không phải code lại các thuật toán phức tạp từ đầu). Để không bị xem là một thằng chỉ biết chém gió, tốt hơn hết hãy gọi đúng tên network và tránh dùng những buzzwords nhé.



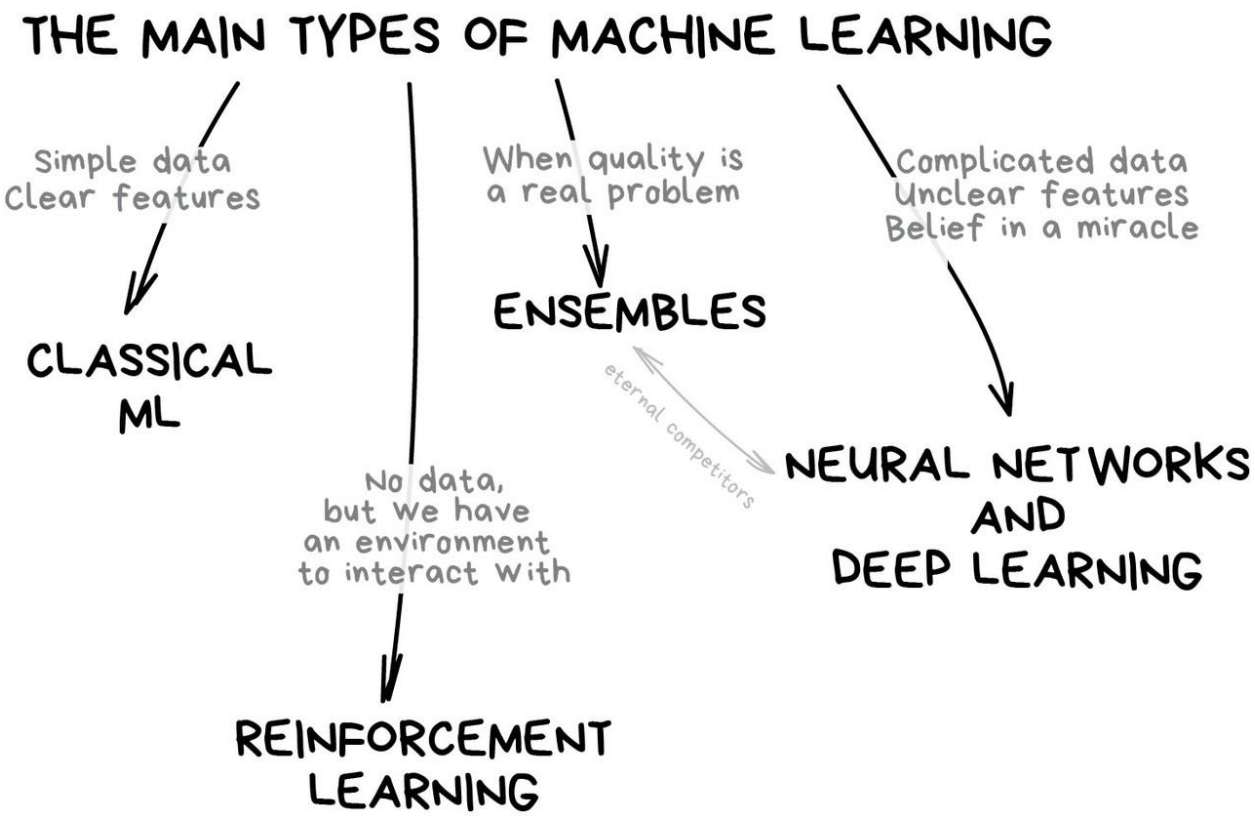
Mối quan hệ giữa AI, ML, NN và DL

Một quy tắc chung nữa là: hãy so sánh những thứ ở cùng một cấp độ thôi. Việc hỏi "Liệu mạng neuron có thay thế học máy hay không" cũng giống như việc hỏi "Liệu cái bánh xe có thay thế được ô tô không" vậy. Truyền thông thân mến, điều đó sẽ làm giảm uy tín của các cậu đi nhiều đấy.

Machine can	Machine cannot
Forecast	Create something new
Memorize	Get smart really fast
Reproduce	Go beyond their task
Choose best item	Kill all humans

Máy móc có thể và không thể làm gì?

Phân loại các mô hình học máy



Hãy bắt đầu với những thứ căn bản. Có bốn hướng chính để giải một bài toán sử dụng học máy:

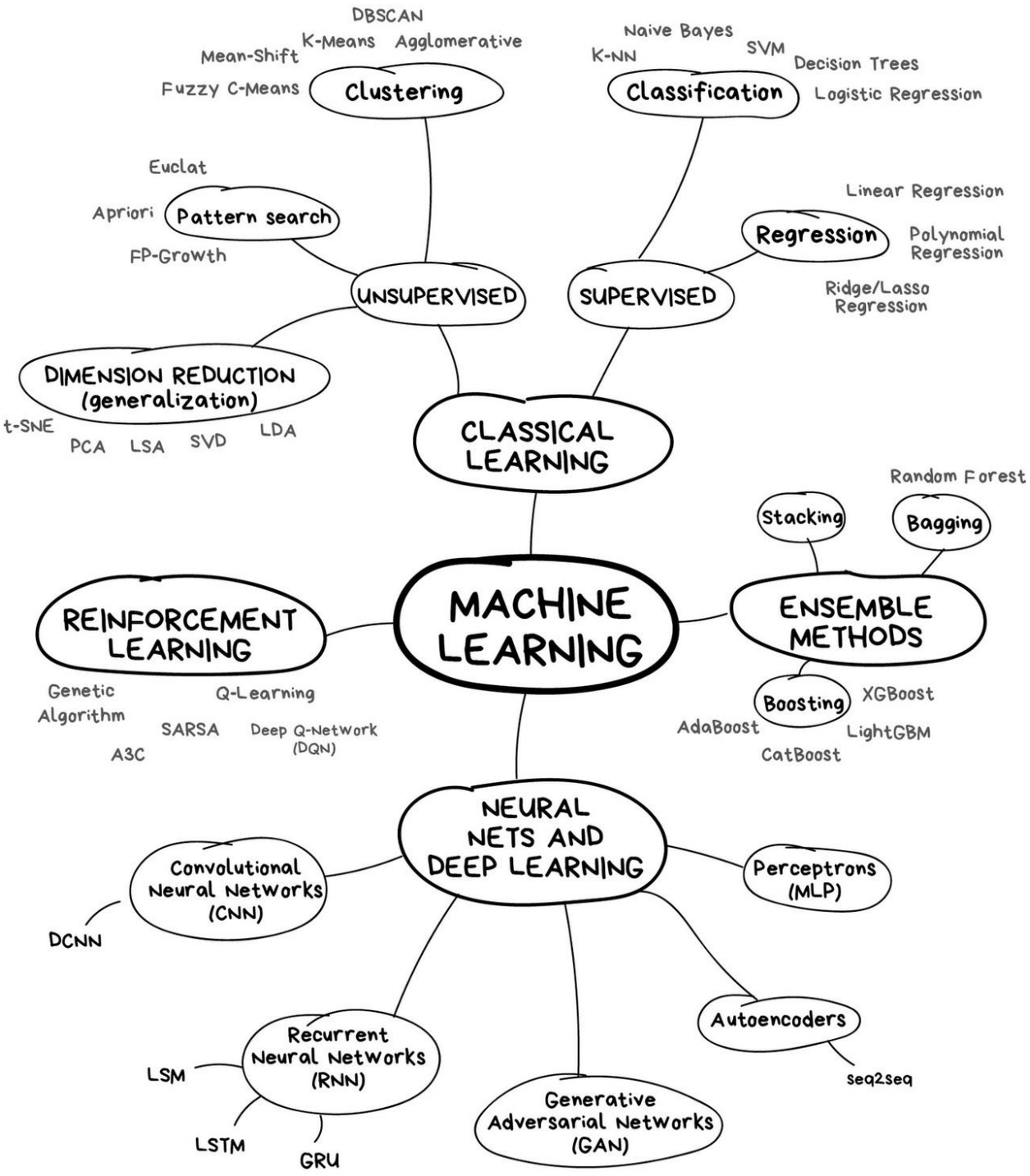
– Classical ML (học máy cổ điển): sử dụng các thuật toán học máy truyền thống, rất hiệu quả với những bài toán có dữ liệu đơn giản và các feature rõ ràng (ví dụ bài toán định giá ô tô).

– Reinforcement learning (Học củng cố – RL): dùng cho những bài toán mà bạn không có dữ liệu đầu vào, nhưng bạn có một môi trường để thoải mái khám phá *(Ví dụ như trong game, sẽ không có dữ liệu từ thực tế, mà chỉ có các quy luật, hệ thống thưởng phạt, và máy tính sẽ học chơi game bằng cách chơi và thất bại nhiều lần, chúng sẽ học được đâu là một bước đi tốt giúp chúng qua cửa và được thưởng điểm, đâu là một bước đi tồi khiến chúng thất bại và phải chơi lại, bị trừ điểm).*

– Ensemble learning (mô hình học máy kết hợp): nói nôm na là bạn so sánh nhiều mô hình trên 1 bài toán cụ thể, với cùng một tập dữ liệu đầu vào, để chọn ra mô hình cho độ chính xác cao nhất. Nên sử dụng phương pháp này cho những bài toán mà bạn ưu tiên chất lượng mô hình *(tất nhiên bài toán nào cũng quan trọng về chất lượng, nhưng nếu bạn dự đoán ung thư thì hẳn nhiên là sẽ cần độ chính xác cao hơn lọc email spam rồi)*.

– Cuối cùng, **neural networks và deep learning:** hãy sử dụng chúng khi bạn có dữ liệu đầu vào phức tạp *(ví dụ như với 100 Gb ảnh mèo ở trên, mỗi tấm có độ phân giải hàng ngàn pixel)*, những đặc trưng không rõ ràng, và bạn thì tin vào phép màu *(nghe có vẻ bí hiểm, nhưng tác giả sẽ giải thích rõ hơn ở các phần sau).*

Và đây là tấm bản đồ full hd không che của thế giới Machine Learning:



Nếu bạn lười đọc, hãy xem qua hình trên để có những hiểu biết chung nhé.

Một điều quan trọng cần nhớ là: luôn luôn có nhiều hơn một cách để giải một bài toán trong thế giới của ML. Sẽ luôn có một vài thuật toán phù hợp, và bạn cần chọn xem cái nào phù hợp hơn thôi. Tất cả mọi thứ có thể được giải quyết bằng một mạng neuron đơn giản, tất

nhiên rồi, nhưng nếu thế ai sẽ mua những chiếc card GeForces đắt tiền đây? *(trong ML, để huấn luyện những mô hình phức tạp, người ta thường dùng card đồ hoạ (GPU) thay cho CPU vì xử lý được nhiều phép toán hơn, nhưng cũng đắt đỏ hơn).*

Chú thích: Link bài viết gốc tại https://bitly.com.vn/uZQKl