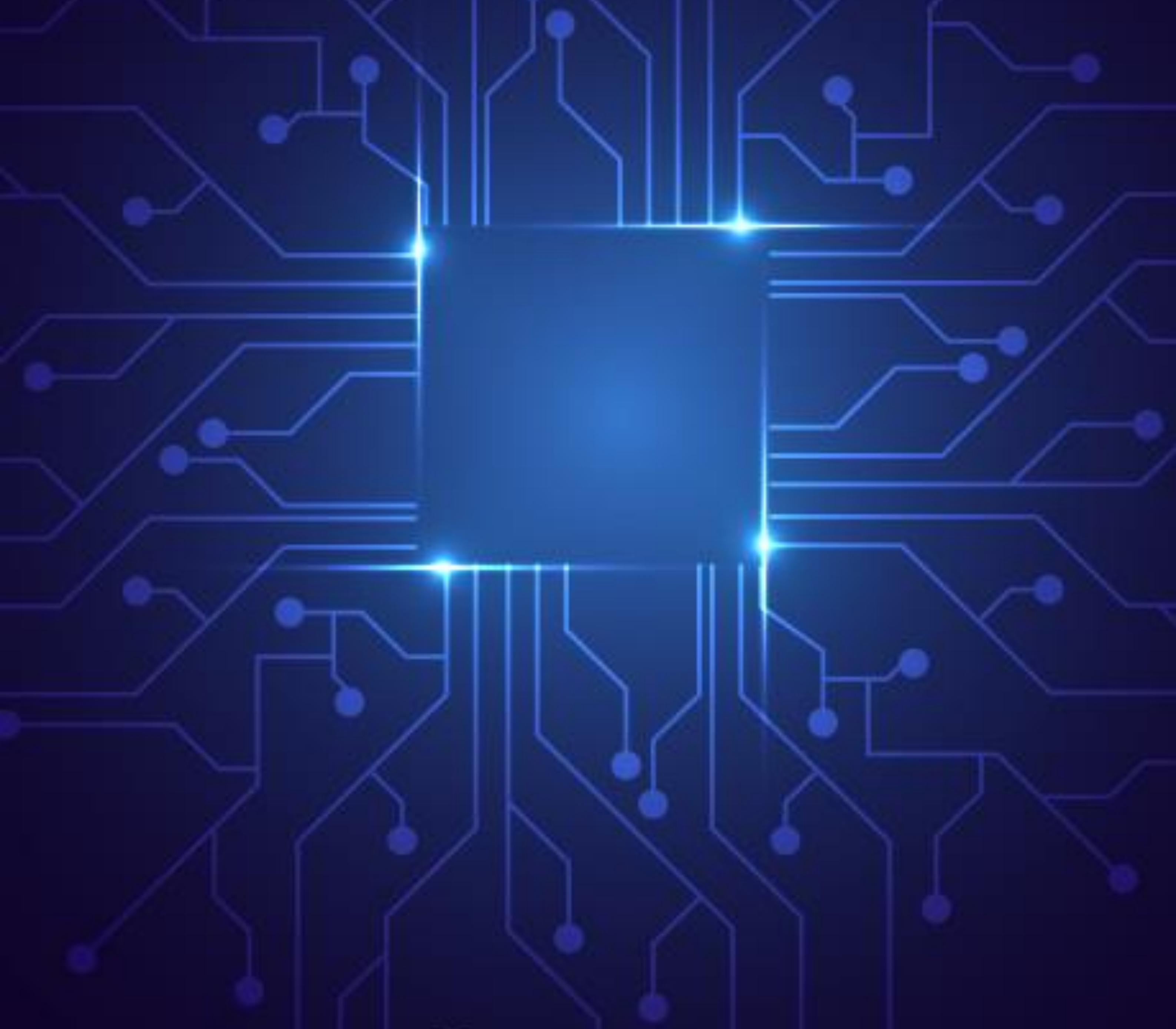


MACHINE LEARNING SIMPLIFIED

By: Kunal Jain

Dedicated to our thriving community – which learns, loves, shares, competes, provides feedback and owns what we have built as much as we do.

A Big Thanks to the entire Analytics Vidhya team, who built this fabulous community and continue to do so. The hard work you all put in enables me to think bigger and give shape to our journey.

Special Thanks to Sunil Ray, Pranav Dar, Tavish Srivastava, Sanad and other authors on Analytics Vidhya - who contributed to the book! This would not be in this form without your contributions.

Analytics Vidhya
Learn everything about analytics

MACHINE LEARNING SIMPLIFIED

www.analyticsvidhya.com

TABLE OF CONTENTS

WHAT IS MACHINE LEARNING?

WHY SHOULD MACHINES LEARN?

Machine Learning is the science of teaching machines how to learn by themselves. Now, you might be thinking – why on earth would we want machines to learn by themselves? Well – it has a lot of benefits. **Machines can do high frequency repetitive tasks with high accuracy without getting bored**.

For example – the task of mopping and cleaning the floor. When a human does the task – the quality of outcome would vary. The human would get exhausted / bored after a few hours of work. The human would also get sick at times.

Depending on the place – it could also be hazardous or risky for a human.

On the other hand, if we can teach machines to detect whether the floor needs cleaning and mopping, how much cleaning is required based on the condition of the floor and the type of the floor - machines would be far better in doing the same job. They can go on to do that job without getting tired or sick!

As this task is well defined, machines can do it well. But machine still needs a way to think that can help it to execute simple things like:

- Whether the floor needs cleaning and mopping?
- How long does the floor need to be cleaned?

This intelligence was traditionally coded by humans in form of simple rules. This approach has serious shortcomings – humans need to figure out precise rules that can solve the problem in varied environment. For instance, the rules that might be required to clean a wooden floor might be very different from that of a mosaic floor. Such specific cases become increasingly complex to code through rules. Hence, rule driven machine intelligence was not that useful a decade back.

But now, we have come a long way. We have figured out a way to make machines learn these rules themselves. All we need to give is a lot and lots of data.

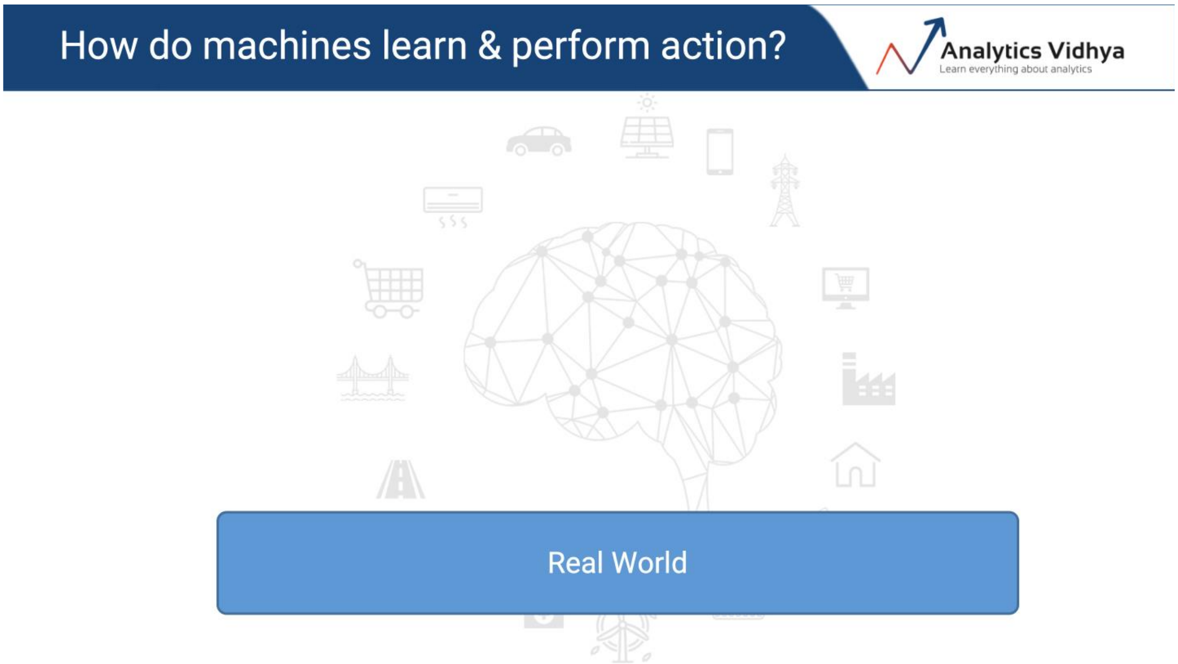
The machines capture data from the environment and feed it to the machine learning model. The model then uses this data to predict whether the floor needs cleaning or not. And, for how long does it need the cleaning.

Now, imagine the world once you have trained your machine to make decisions about cleaning on their own – you no longer need to worry about cleaning, mopping, variability of quality, holidays and many other factors.

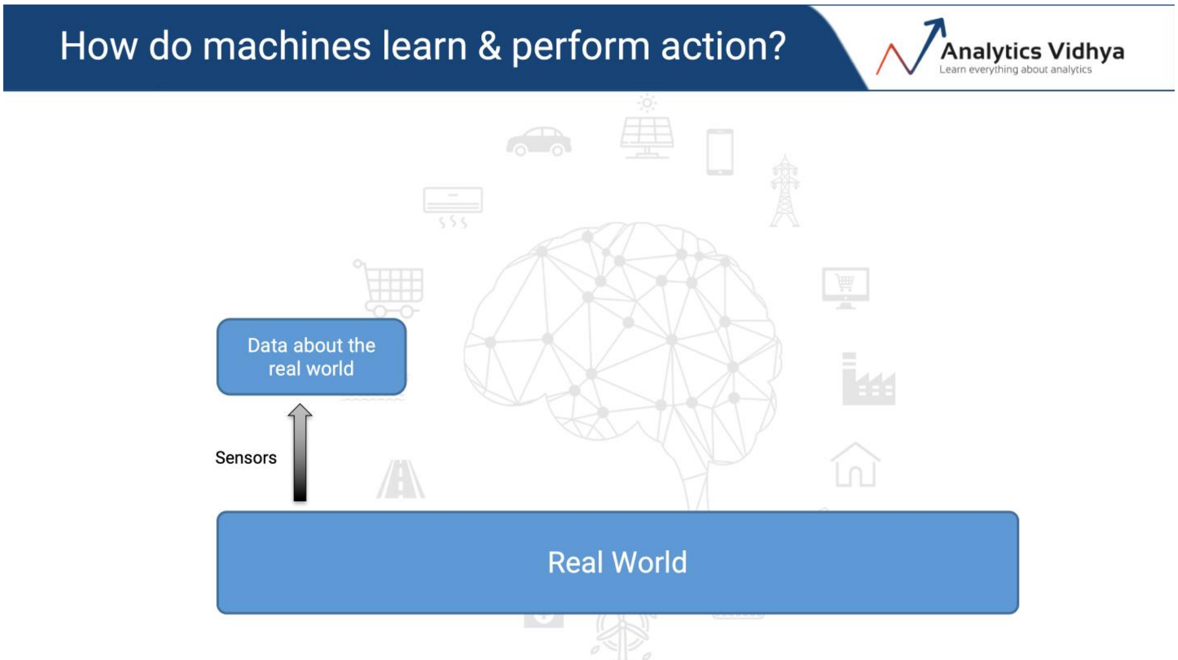
Once learnt, the machine is far better in performing this high frequency repetitive job and can make our life much better by taking care of the same. But, how do we teach machines to learn?

HOW DO MACHINES LEARN?

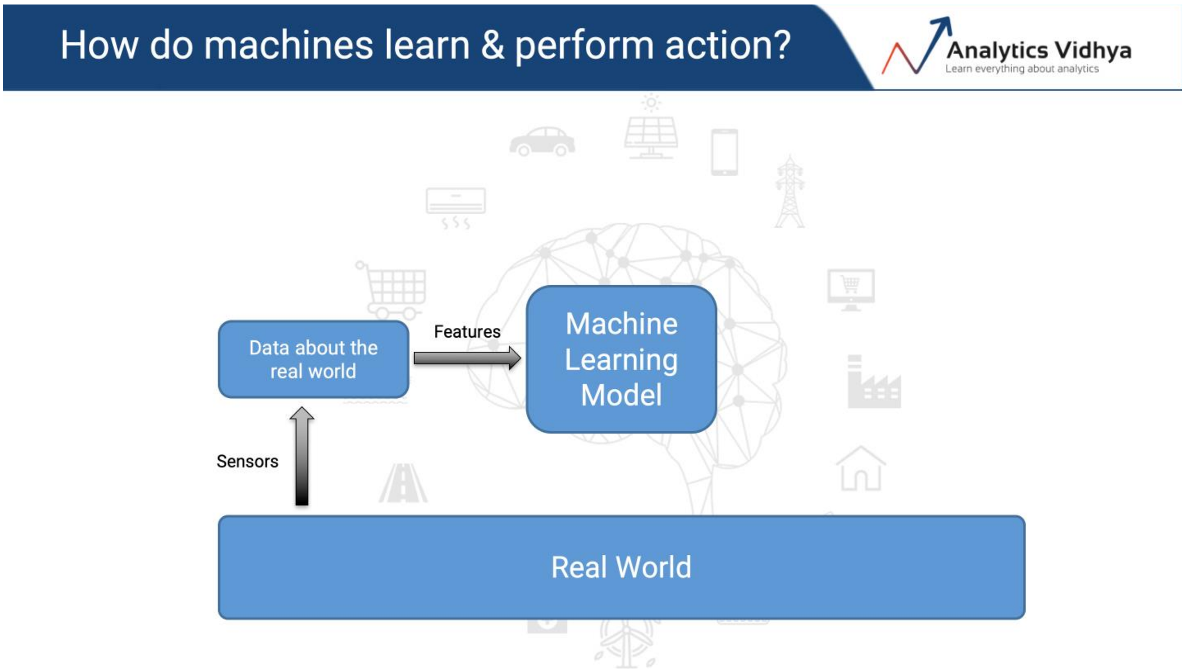
Let us understand how machines learn in a simple manner. We represent the world by a simple box as shown in the Image below:



To understand the current environment in which the machine is - it captures data from this world. This data is captured by various sensors in a machine. This data can also come from existing systems in an Enterprise (which would have been captured by some machine ultimately).

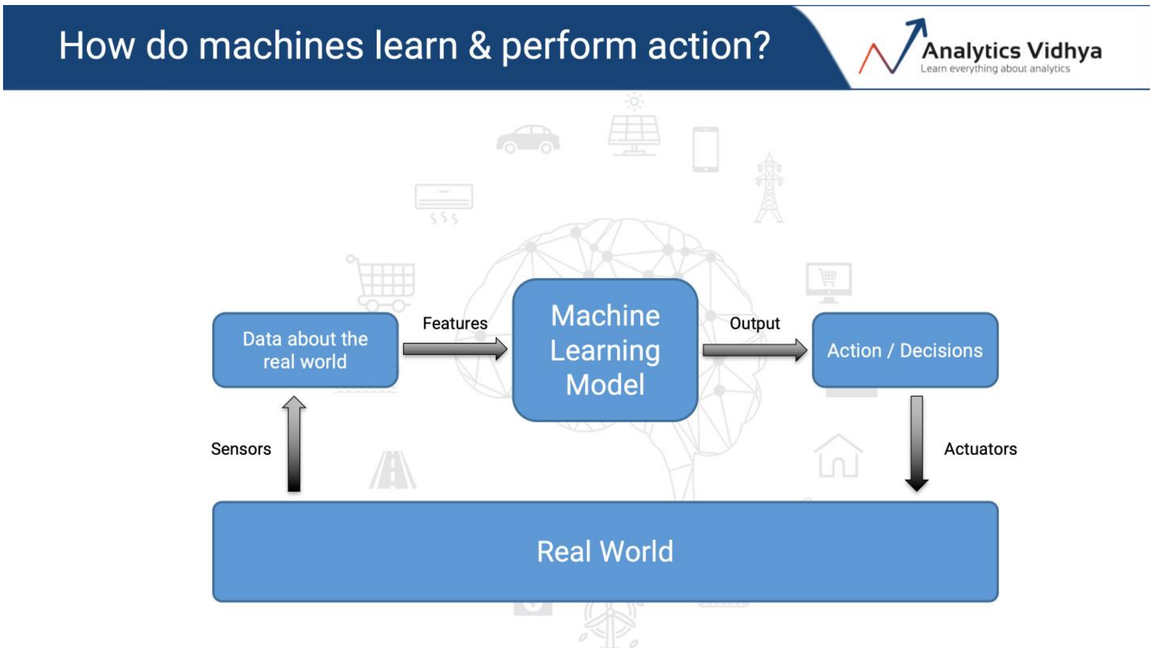


This data flows into machine learning models in the form of variables (or features in machine learning paradigm).



For example, your smartphone captures your phone orientation through several sensors like gyrometers and accelerometers. This data is then fed into a Machine Learning algorithm, which predicts the Orientation of your phone.

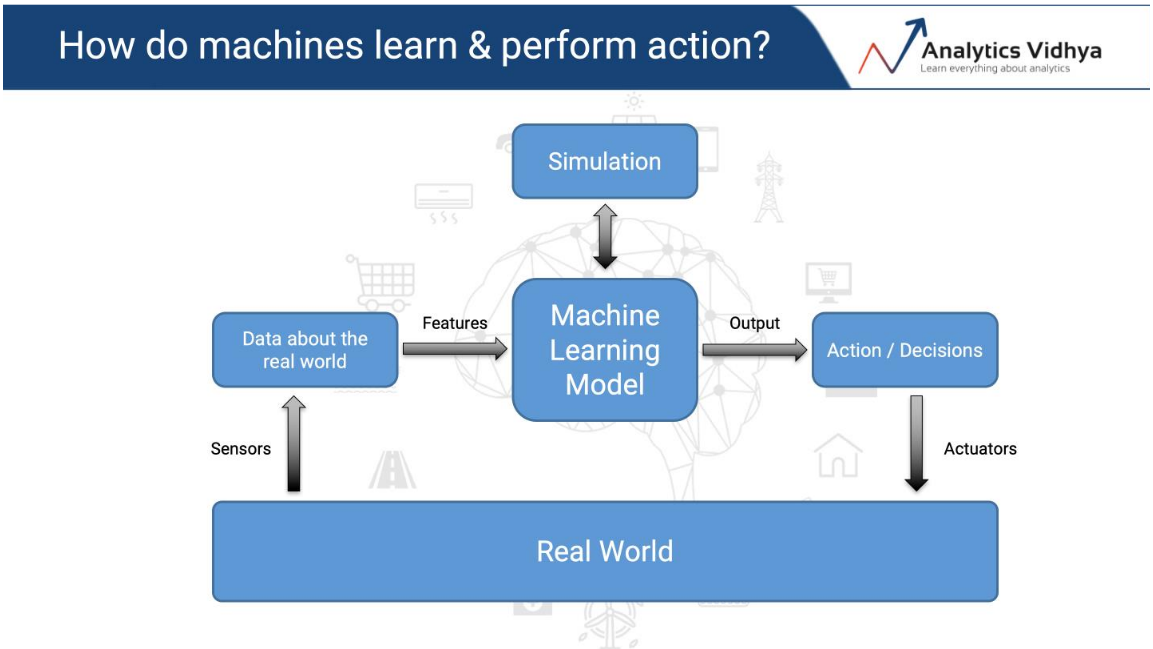
In Summary, this is how the Interaction can be represented:



Once the model makes a prediction, the machines takes a decision or action on that basis.

Depending on the complexity of the system and the requirements from the machine, it might also need to run simulations along with the machine learning model to take decisions.

For example, in a self-driving car, machine learning is used to understand various objects on the road. Once that is done, it runs a simulation to see what could be the potential movement of each of those things.



This schematic represents what we today describe as "Smart Machines" and it represents how machines learn from the real world and take actions.

All smart machines can be represented in this simple schematic. Take a Voice Assistant like Alexa for example. It captures data from various mics embedded on the assistant. Once captured, it runs the machine learning models on this data to create an Output, which it provides to the end user.

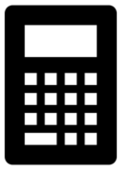
Now that you understand how machines learn – the next obvious question which comes to mind is "How easy or difficult is it to teach machines to learn?"

HOW EASY OR DIFFICULT IS IT TO TEACH MACHINES TO LEARN?

The current design of machines and computers is amazing! Things which look very difficult to a human mind can be easy for computers. For example, try and multiply these 2 numbers in your mind: 3848952348998 x 73289478912.

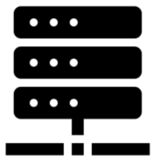
How much time did you take?

Now give it to a Calculator or Computer. You will see what I mean.



On the other hand, things which are usually intuitive to humans can be very difficult for machines. For example, you only need to demonstrate cleaning and mopping to a human a few times before they can perform it on their own.

But, that is not the case with machines. To teach machines how to clean and mop, we need to collect a lot of data along with the desired outcomes. Then we teach machines to predict the outcomes and perform them based on the data. This is where machine learning comes into play.



Machine Learning would help the machine understand the kind of cleaning, intensity of cleaning, duration of cleaning based on the conditions and nature of the floor. But in order to do that, it would need a lot of observations (past decisions) before the machine can become capable of making these decisions on its own.

This is why machine learning in its current form is best suited for high frequency repetitive activities which have a lot of past data.

Recent Developments in Machine Learning

The idea behind machine learning sounds exciting. But, this idea of teaching machines has been around for a while. Remember Asimov's Three laws of robotics? Machine Learning ideas and research have been around for decades. However, there has been a lot of action and buzz recently.

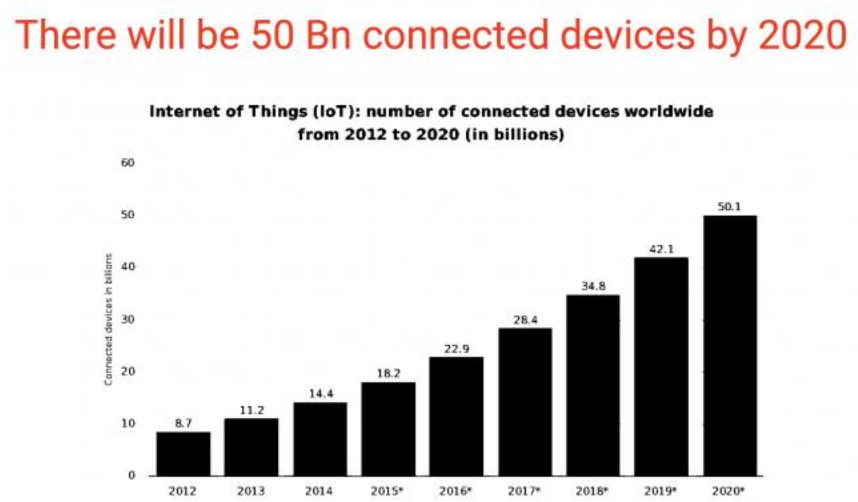
The obvious question is why is this happening now, when machine learning has been around for several decades?

Why is Machine Learning getting so much attention recently?

While machine learning has been around for a few decades, we were not able to use it to its true potential due to multiple reasons. This includes factors like limited availability of data and compute power required by the algorithms.

However, a lot has changed in the last few years. Let us look at the fundamental forces driving the growth of machine learning:

- **Change 1** - The amount of data generation is increasing significantly with reduction in the cost of sensors**.** As the cost of sensors is going down, more and more hardware is coming with sensors connected to the Internet. Health Bands to Connected Microwave ovens – all fall in this category. Each of these devices would continue to generate huge amount of data in the coming years.



- **Change 2** - The cost of storing this data has reduced significantly. So, not only are we generating more data – we can actually store this data cheaply. The following figure shows how storage costs have come down over time.

- **Change 3** - The cost of computing has come down significantly. This enables us to run computations on the stored data in a cost efficient manner.

- **Change 4** - Cloud has democratised Compute for the masses. Not only have the costs for computations come down – they have also been democratised. What this means is that people like you and me can spin servers on the Cloud for the duration of time we need and then shut them off, without buying any hardware. This enables people to train their own machine learning models at a very low cost and no huge upfront hardware costs.

These 4 forces combine to create a world where we are not only creating more data, but we can store cheaply and run huge computation on it. This was not possible before, even though machine learning techniques and algorithms were well known.

LATEST DEVELOPMENTS IN MACHINE LEARNING

Machine Learning is making some big news lately. From winning strategy games to saving electricity to running self-driving cars – machine learning is taking the world by storm.

Here are some recent applications of machine learning:

- **Generation and Detection of Fake News and Fake Videos** - Facebook and Twitter have made it easier than ever to find quality news. They have also enabled putting out misleading or outright false information posted online. A very recent confusion over a video of U.S. House Speaker Nancy Pelosi doctored to make her sound drunk is just the latest example of the threat posed by these technologies.
- **AI warfare** is beginning to dominate military strategy in the US and in China, but it can be easily and dangerously fooled. AI algorithms are far from accurate to be used in military, but the advantages are immense as well.
- A **robot hand taught itself to solve Rubik's cube** after creating its own training regime. Again – a recent news showcasing the power of Artificial Intelligence and Machine Learning.

What is clear is that Artificial Intelligence and Machine Learning can be used to perform things which could not be thought before. Any domain we think of is about to change with the application of these technologies.

HOW GOOD ARE THE MACHINES CURRENTLY?

At the current level of technological advancement, machines are only good at doing specific tasks. A machine, which has been "taught" cleaning can only do cleaning (for now). In fact, if there is a surface of new material or form which the machine has not been trained on – the machine will not be able to work on it in the same manner.

This is usually not the case with humans. So, if a person is responsible for cleaning and mopping, he / she can also be a security guard. He / she can also help in planning logistics.

The current phase of artificial intelligence is typically referred to as "**Artificial Narrow Intelligence (ANI)**". This is very different from creating an **"Artificial General Intelligence (AGI)"**, which is usually referred to Human level intelligence. AGI is generic in nature and refers to a phase where machines can perform multiple tasks with ease. At this stage, AGI is only a concept and no one in the domain knows for sure how far are we from achieving AGI.

According to some of the leading researchers in the field – if and when a machine achieves AGI, it will not take machines a long time to surpass Human Intelligence. This is because the machine will continue to evolve and learn continuously without having any human limitations like memory leaks. This phase of Artificial Intelligence will be called **"Artificial Super Intelligence (ASI)".**

The infographic below summarizes different levels of Artificial Intelligence:

Levels of Artificial Intelligence — Analytics Vidhya (Learn everything about analytics)

AI Stage	**Artificial Narrow Intelligence (ANI)** • Execute Specific tasks well • E.g. Driving, Medical	**Artificial General Intelligence (AGI)** • Machines achieve human level intelligence	**Artificial Super Intelligence (ASI)** • Machine is more intelligent than human across in practically every field
Projected Timelines	Currently	25 – 75 years	Shortly after AGI
Implications	• Machines become better in specific tasks – playing games, driving, flying, listening, translating languages • Jobs displacement with AI taking over jobs which are repetitive in nature	• Machines start learning and creating knowledge • Machines would be able to do creative things – create better movies, write books	• Machines become better than humans at most of the jobs

WHAT ARE SOME OF THE CHALLENGES IN ADOPTION OF MACHINE LEARNING?

While machine learning has made tremendous progress in the last few years, there are some big challenges which still need to be solved. This is an area of active research and I expect a lot of efforts to solve these problems in the coming time.

- **Huge data required** – It takes a huge amount of data to train a model today. For example – if you want to classify Cats vs. Dogs based on images (and you don't use an existing model) – you would need the model to be trained on thousands of images. Compare that to a human – we typically explain the difference between Cat and Dog to a child by using 4 or 5 photos.
- **High compute required** – As of now, machine learning and deep learning models require huge computations to achieve simple tasks (simple according to humans). This is why use of special hardware including GPUs and TPUs is required. The cost of computations needs to come down for machine learning to make a next level impact.
- **Interpretation of models is difficult at times** – Some modeling techniques can give us high accuracy, but are difficult to explain. This can leave the business owners frustrated. Imagine being a bank, but you cannot tell why you declined a loan for a customer!
- **New and better algorithms required** – Researchers are consistently looking out for new and better algorithms to address some of the problems mentioned above.
- **More Data Scientists needed** – Further, since the domain has grown so quickly – there aren't many people with the skill sets required to solve the vast variety of problems. This is expected to remain so for the next few years. So, if you are thinking about building a career in machine learning – you are in good stead!

IS MACHINE LEARNING A COMPLETE BLACK BOX?

No – it is not. There are methods or algorithms in machine learning, which can be interpreted well. These methods can help us understand what are the significant relationships and why the machine has taken a particular decision.

On the other hand, there are certain algorithms which are difficult to interpret. With these methods, even if we achieve a very high accuracy – we may struggle with explanations.

The good thing is that depending on the application or the problem we are trying to solve – we can choose the right method. This is currently a very active field of research and development.

Common Applications of Machine Learning

Given what we've covered so far, you might be asking what are some of the examples of machine learning and how does it affect our life?

Unless you have been living under a rock – your life is already heavily impacted by machine learning. Let us look at a few examples where we use the outcomes of machine learning already:

- Smart phones detecting faces while taking photos or unlocking themselves.
- Facebook, LinkedIn or any other social media site recommending you friends and ads you might be interested in.
- Amazon recommending you the products based on your browsing history.
- Banks using Machine Learning to detect Fraud transactions in real time.

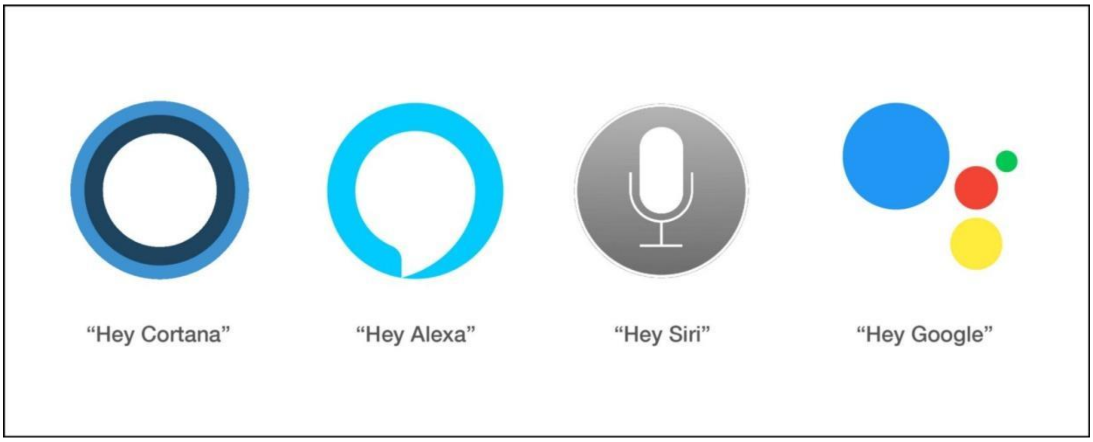
Machine Learning Use Cases in Smartphones

Did you know that machine learning powers most of the features on your smartphone?

That's right! From the voice assistant that sets your alarm and finds you the best restaurants to the simple use case of unlocking your phone via facial recognition – machine learning is truly embedded in our favourite devices.

Voice Assistants

That example we saw in the introduction about talking to our virtual assistant? That was all about the concept of speech recognition – a budding topic in machine learning right now.



Voice assistants are ubiquitous. You must have used (or at least heard about) the below popular voice assistants:

- Apple's Siri
- Google Assistant
- Amazon's Alexa
- Google Duplex
- Microsoft's Cortana
- Samsung's Bixby

And so on. The common thread between all of these voice assistants? They are powered by machine learning algorithms! These voice assistants recognize speech (the words we say), convert them into numbers machine can understand and uss machine learning to formulate a response accordingly.

The field is ripe for these assistants becoming smarter in the future as machine learning techniques become more advanced.

Smartphone Cameras

Wait – what in the world does machine learning have to do with my smartphone camera? Quite a lot, as it turns out.

The incredible images we are able to click these days and the depth of these images – all of that is thanks to machine learning algorithms. They analyze every pixel in a given image to detect objects, blur the background, and a whole host of other tricks.

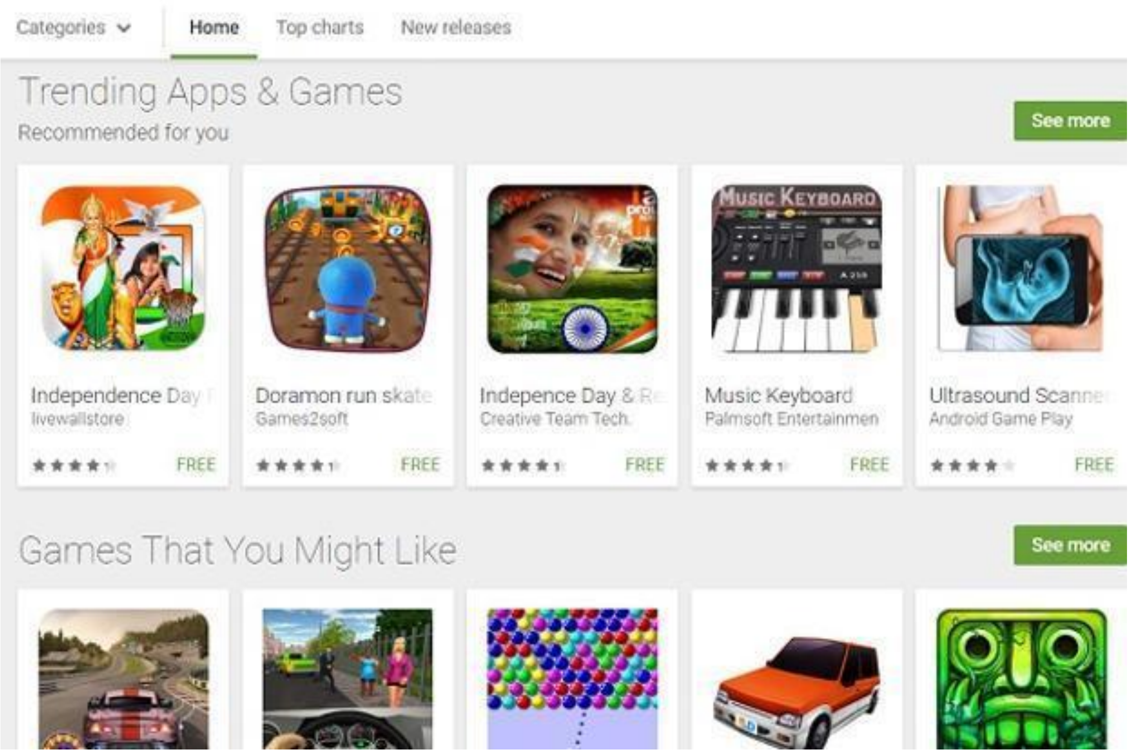
These machine learning algorithms do several things to improve and enhance the smartphone's camera:

- **Object detection** to locate and single out the object(s) (or human) in the image.
- Filling in the missing parts in a picture.
- Using a certain type of neural network using **GANs** to enhance the image or even extend its boundaries by imagining what the image would look like, etc.

App Store and Play Store Recommendations

I love this feature of both Google's Play Store and Apple's App Store. The 'Recommended for you' section is based on the applications I have installed on my phone (or previously used).

For example, if I have a few sports and food-related applications – so my recommended for you section is usually filled with applications that are similar to these apps. I appreciate that the Play Store is personalized to my taste and shows me apps I have a higher chance of downloading.



Face Unlock – Smartphones

Most of us are quite familiar with this. We pick up our smartphone and it unlocks itself by detecting our face. It's smart, efficient, time-saving and frankly superb.

What a lot of people don't know about this is that our smart phones use a technique called facial recognition to do this. And the core idea behind facial recognition is powered by – you guessed it – machine learning.

The applications of facial recognition are vast and businesses around the world are already reaping the benefits:

- Facebook uses it to identify the people in images
- Governments are using it to identify and catch criminals
- Airports are using to verify passengers and crew members, and so on

Machine Learning Use Cases in Transportation

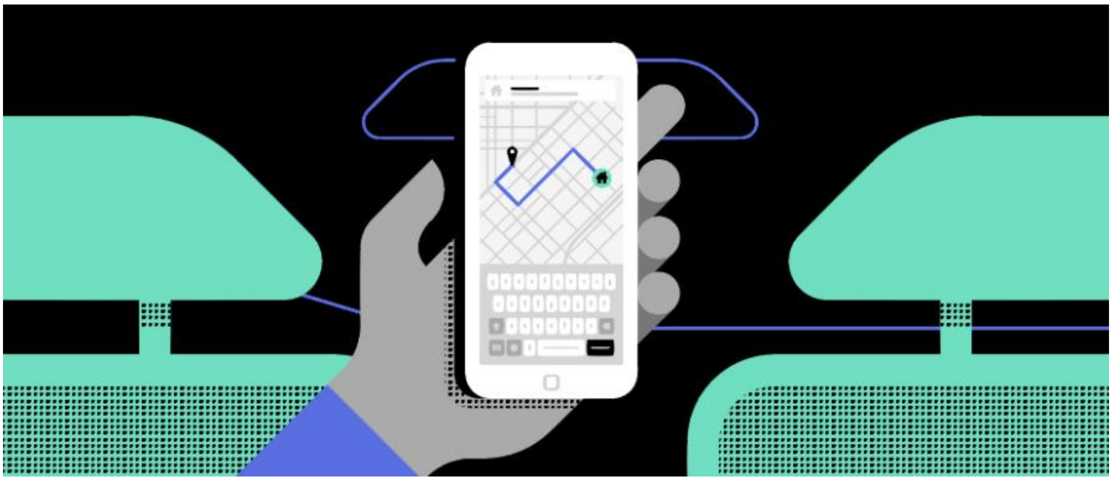
The applications of machine learning in the transport industry has gone to an entirely different level in the last decade. This coincides with the rise of ride-hailing apps like Uber, Lyft, Ola, etc.

These companies use machine learning throughout their many products, from planning optimal routes to deciding prices for the rise we take. So let's look at a few popular use cases in transportation which use machine learning heavily.

Dynamic Pricing in Travel

Do you often get frustrated by the surge pricing cab-hailing companies use? I encounter it on a daily basis for my commute to and from work. Prices seem to be perpetually hiked. Why is this happening?!

So I dug into this a bit more and came across the concept of dynamic pricing – an excellent machine learning use case. To understand this, let's take a simple example.



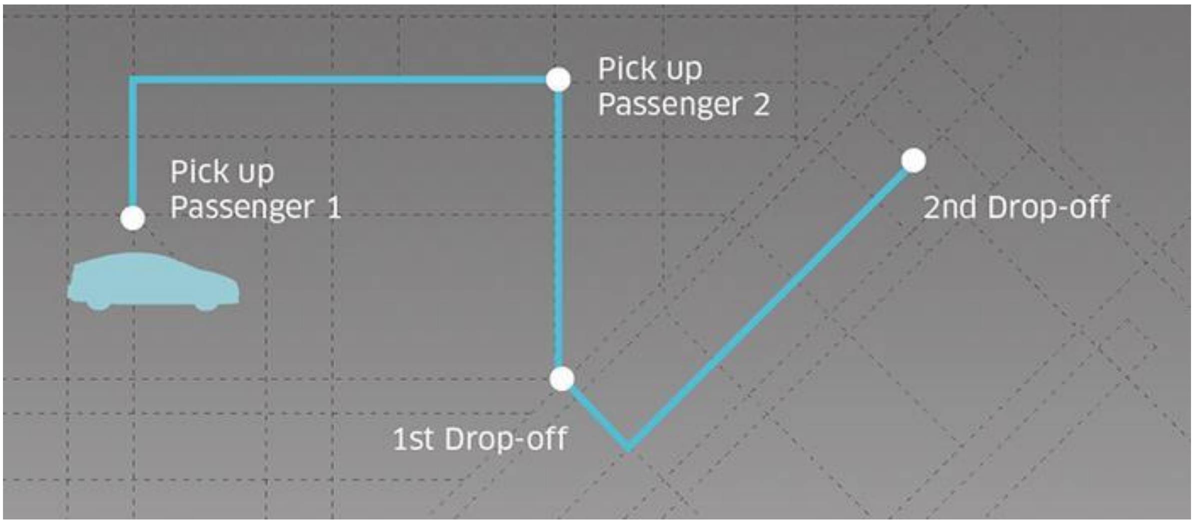
Imagine you're starting a ride-hailing business. You need to plan the ride prices for each route in the city in a way that would attract customers but also improve your bottom line. One way to do it is to manually map prices to each route. Not an ideal solution.

This is where dynamic pricing plays a vital role. It means adjusting your prices to changing market conditions. So prices vary depending on factors like location, time of day, weather, overall customer demand, etc. That's the underlying idea behind why surge pricing was introduced.

Dynamic pricing is a thriving practice in various industries, such as travel, hospitality, transportation, and logistics, among others.

Transportation and Commuting – Uber

Dynamic pricing isn't the only machine learning use case ride-hailing companies like Uber use. They rely heavily on machine learning to identify the most optimal route to get the passenger from point A to B.

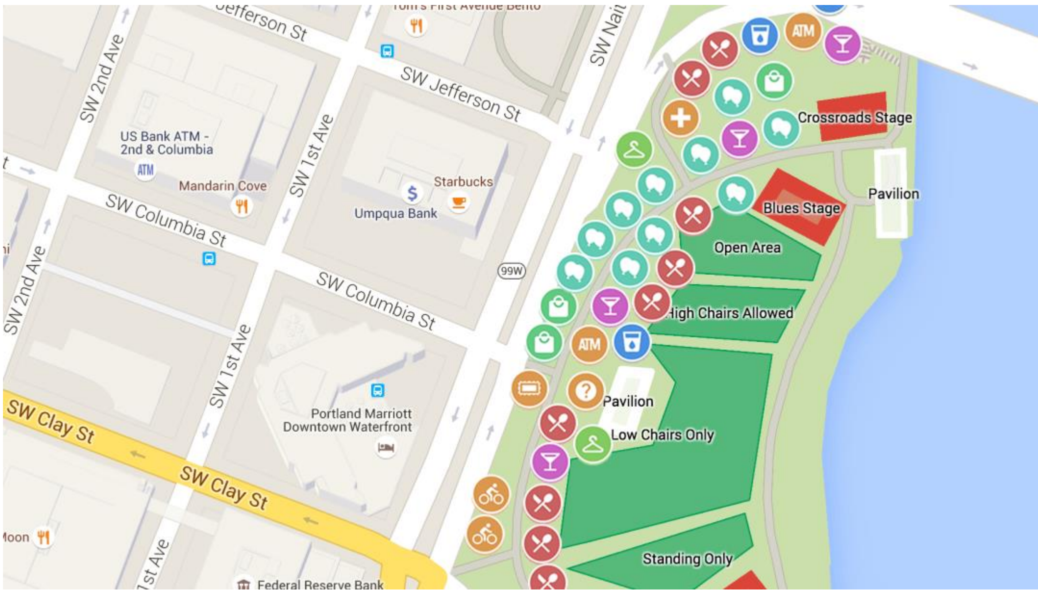


For us, it appears to be a rather simple solution. Put your location, the destination and the nearest driver will come to pick us up. But what appears to be straightforward is actually a complex web of architectures and services on Uber's backend.

There are multiple machine learning techniques at play that aim to optimize the route we take.

GOOGLE MAPS

You must have guessed this one by now. Google Maps is a prime example of a machine learning use case. In fact, I would recommend opening up Google Maps right now and picking out the different features it offers.



Here are some that I can see (and have used extensively):

- Routes: Go from point A to point B
- Estimated time to travel this route
- Traffic along the route
- The 'Explore Nearby' feature: Restaurants, petrol pumps, ATMs, Hotels, Shopping Centres, etc.

Google uses a ton of machine learning algorithms to produce all these features. Machine learning is deeply embedded in Google Maps and that's why the routes are getting smarter with each update.

The estimated travel time feature works almost perfectly. If it shows '40 minutes' to reach your destination, you can be sure your travel time will be approximately around that timeline. Got to love machine learning!

Machine Learning Use Cases in Popular Web Services

You'll love this section. We interact with certain applications every day multiple times. What we perhaps did not realize until recently – most of these applications work thanks to the power and flexibility of machine learning.

Here are four use cases you are ultra-familiar with. Now, look at them from a machine learning perspective.

Email filtering

Dealing with too many emails at work? Or is your personal email inbox bursting with utterly random and spam emails? We've all been there. My inbox count once read 11,000+ unread emails!

Wouldn't it be easier if we could write a rule that would filter emails according to their subject? A marketing mail would go to that folder. An email about work would come into my primary inbox (and so on). This would make life so much easier.

As it turns out, this is exactly what most email services are now doing! They're using machine learning to parse through the email's subject line and categorize it accordingly. Take Gmail for example. The machine learning algorithm Google uses has been trained on millions of emails so it can work seamlessly for the end-user (us).

While Gmail allows us to customize labels, the service offers default labels:

- Primary
- Social
- Promotions

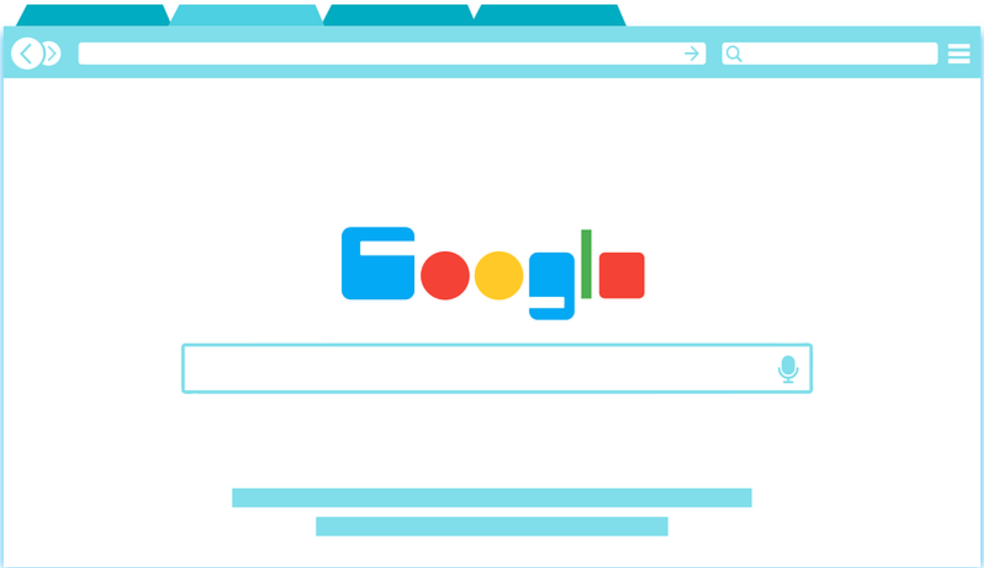
The machine learning algorithms immediately categorize the email into one of these three labels as soon as you receive an email. We get an instant alert if Gmail deems it a 'Primary' email.

Of course, Gmail also uses machine learning to figure out if the email is spam or not. A feature we are all truly grateful for. Google's algorithm has become a lot smarter over the years in deciding if an email is spam or not. This is where getting more data for a machine learning algorithm is so helpful – something Google has in abundance.

The most popular machine learning use case in this (or any) list. Everyone has used Google Search and most of us use it multiple times on a daily basis. I would venture to say we take it for granted that Google will serve us the best results up front.

But how does Google Search work?

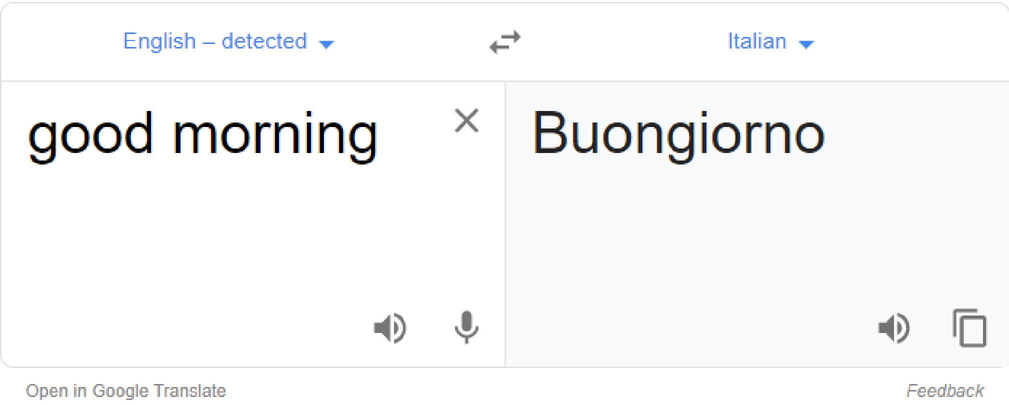
Google Search has become an impenetrable behemoth that mortals cannot crack. How it works underneath is something only those folks who have designed Google Search know. One thing we can say for certain – Google uses machine learning to power its Search engine.



The amount of data Google has to constantly train and refine its algorithms is a number we cannot fathom. No calculator in the world will tell us the number of queries Google has processed in the last two decades. It is a treasure trove for data scientists!

Google Translate

I'm fluent in Google Translate. I've picked up bits and pieces of foreign languages like German, Spanish, and Italian thanks to this wonderful service by Google. Anytime I come across a bit of text in a foreign language, Google Translate immediately offers me the answer.

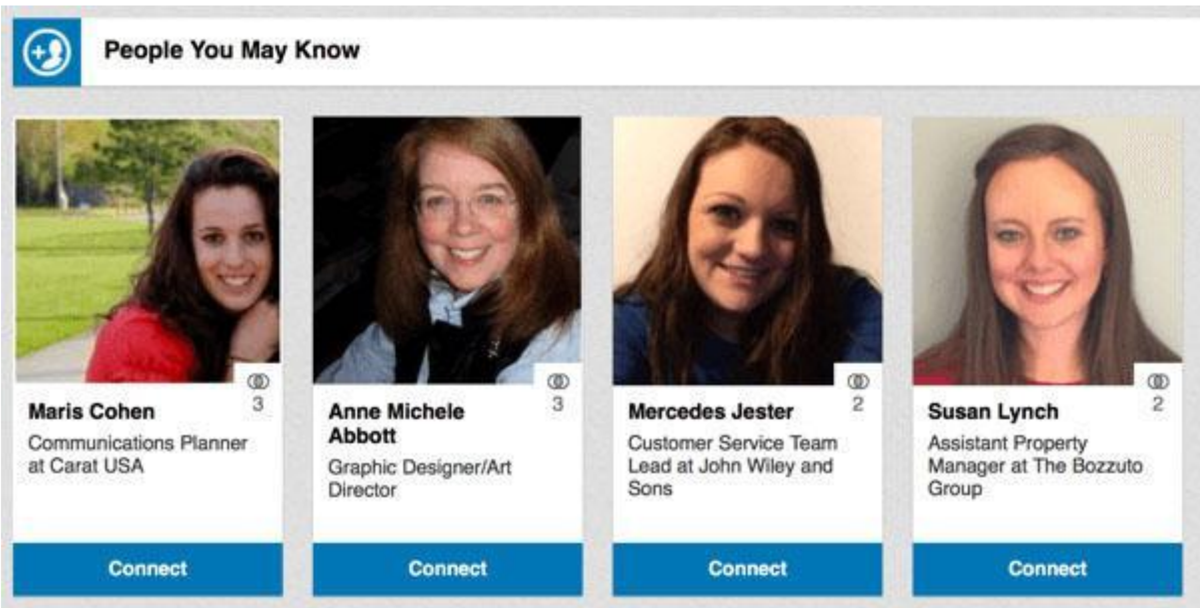


It won't surprise you to know that Google uses machine learning to understand the sentence(s) sent by the user, convert them to the requested language, and show the output. Machine learning is deeply embedded in Google's ecosystem and we are all benefitting from that.

LinkedIn and Facebook Recommendations and Ads

Social media platforms are classic use cases of machine learning. Like Google, these platforms have integrated machine learning into their very fabric. From your home feed to the kind of ads you see, all of these features work thanks to machine learning.

A feature which we regularly see is 'People you may know'. This is a common feature across all social media platforms, Twitter, Facebook, LinkedIn, etc. These companies use machine learning algorithms to look at your profile, your interests, your current friends, their friends, and a whole host of other variables.



The algorithm then generates a list of people that match a certain pattern. These people are then recommended to you with the expectation that you might know them (or at least have profiles very similar to yours).

I have personally connected with a lot of my professional colleagues and college friends thanks to LinkedIn's system. It's a use case of machine learning benefiting everyone involved in the process.

The ads that we see work in a similar fashion. They are tailored to your tastes, interests and especially your recent browsing or purchase history. If you are a member in data science group(s), Facebook or LinkedIn's machine learning algorithm might suggest machine learning / data science courses.

Pay attention to this next time you're using social media. It's all machine learning behind the curtains!

Machine Learning Use Cases in Sales and Marketing

Top companies in the world are using machine learning to transform their strategies from top to bottom. The two most impacted functions? Marketing and Sales!

These days if you're working in the marketing and sales field, you need to know at least one Business Intelligence tool (like Tableau or Power BI). Additionally, marketers are expected to know how to leverage machine learning in their day-to-day role to increase brand awareness, improve the bottom line, etc.

So, here are three popular use cases in marketing and sales where machine learning is changing the way things work.

Recommendation Engines

We briefly spoke about recommendation engines earlier. I mentioned that these systems are ubiquitous. But where are they used in the marketing and sales field? And how?

Let's take a simple example to understand this. Before the advent of IMDb (and Netflix), we all used to go to DVD stores or rely on Google to search for movies to watch. The store clerk would offer suggestions on what to watch and we took a hail mary pass by picking up movies we had no idea about.

That world is almost completely in the past now thanks to recommendation engines. We can log on to a site and it recommends products and services to me based on my taste and previous browsing history. Some popular examples of recommendation engines:

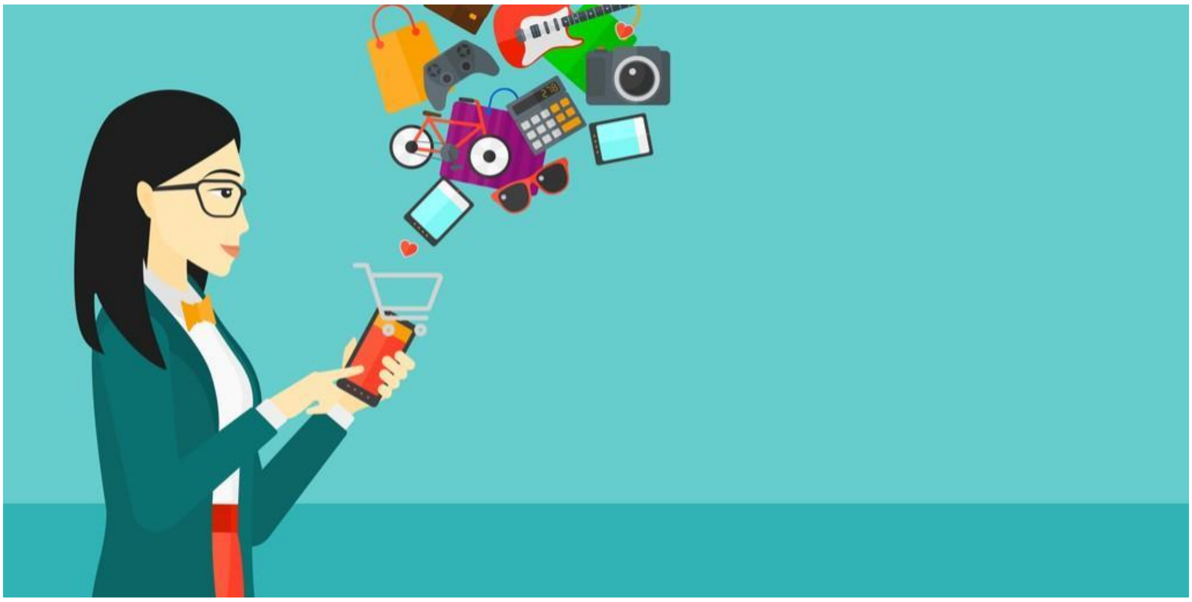
- E-commerce sites like Amazon and Flipkart
- Book sites like Goodreads
- Movie services like IMDb and Netflix
- Hospitality sites like MakeMyTrip, Booking.com, etc.
- Retail services like StitchFix
- Food aggregators like Zomato and Uber Eats

The list is long. Recommendation engines are everywhere around us and marketing and Sales departments are leaning on them more than ever before to attract (and retain) new customers.

Personalized Marketing

Recommendation engines are part of an overall umbrella concept called personalized marketing. The meaning of this concept is in the name itself – it is a type of marketing technique tailored to an individual's need.

Think about this. How many calls do you get from credit card or loan companies offering their services "for free"? These calls offer the same services without understanding what you want (or don't want). It's traditional marketing that is now outdated and well behind the digital revolution.



Now imagine if these calls or emails came highly personalized to your interests. If you're a big shopaholic and that reflects in your purchase history, perhaps the message could be about a new service offering to extend your credit line. Or if you're a machine learning enthusiast, the email could offer courses suited to your taste.

Honestly, the potential for personalized marketing is HUGE. Machine learning helps to identify customer segments and tailor your marketing campaigns for those segments. You can regularly check how your campaign is doing through metrics like open rates, click through rates, and so on.

Customer Support Queries (and Chat bots)

You will understand this at a very personal level if you've ever dealt with customer support (and who hasn't?). Those dreaded phone calls, the interminable wait, the unresolved query – it all adds up to a very frustrating user experience.

Machine learning is helping remove all these obstacles. Using concepts of Natural Language Processing (NLP) and sentiment analysis, machine learning algorithms are able to understand what we're saying and the tone which we are saying it in.

We can broadly divide these queries into two categories:

- Voice-based queries
- Text-based queries

For the former, machine learning algorithms detect the message and the sentiment to redirect the query to the appropriate customer support person. They can then deal with the user accordingly.

Text-based queries, on the other hand, are now almost exclusively being handled by chatbots. Almost all businesses are now leveraging these chatbots on their sites. They remove the impediment of waiting and immediately provide answers – hence, a super useful end-user experience.

Machine Learning Use Cases in Security

Machine learning is disrupting the security industry as well! The days of traditional security, where security guards used to sit for hours on end noting down vehicle numbers and stopping suspicious folks – it's slowly being phased out.

Businesses are using machine learning to better analyze threats and respond to adversarial attacks. These use cases extend to both offline threats as well as online (bank frauds, financial threats, etc.).

Video Surveillance

I’m certain you must have heard or read about a certain country using video surveillance to track its citizens (a quick Google search will tell you anyway). Organizations globally are using video surveillance for various tasks, like detecting intruders, identifying threats of violence, catching criminals, etc.

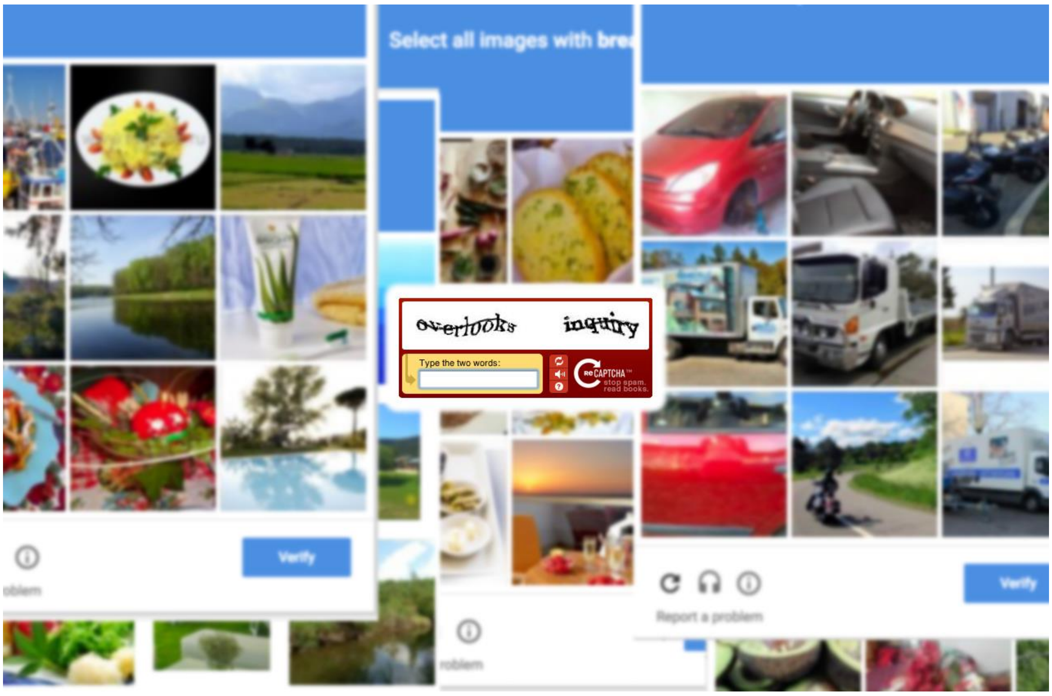
All of this is not being done manually, however. That would be immensely time consuming. So instead, machine learning algorithms are being used for the software that is put inside these surveillance cameras.

These machine learning algorithms use various computer vision techniques (like object detection) to identify potential threats and nab offenders.

Cyber Security (Captchas)

“I’m not a robot” – does this sentence seem familiar? We often encounter this button when a website suspects it is dealing with a machine rather than a human.

These tests are called CAPTCHA, short for Completely Automated Public Turing test. We are asked to identify traffic lights, trees, crosswalks, and all sorts of objects to prove that we are, indeed, human.



The traffic lights and trees are getting covered by other objects, cars are getting obscured, crosswalks are more distant, and all sorts of complications. Why are websites making life more difficult for us? The answer to this lies with machine learning.

The website Verge put it best:

> *"Because CAPTCHA is such an elegant tool for training AI, any given test could only ever be temporary, something its inventors acknowledged at the outset. With all those researchers, scammers, and ordinary humans solving billions of puzzles just at the threshold of what AI can do, at some point the machines were going to pass us by."*

So Google is using machine learning to make CAPTCHA even more complex to decipher. The researchers are using image recognition techniques to crack these CAPTCHAS and consequently enhance their security at the backend.

Machine Learning Use Cases in the Financial Domain

Most of the jobs in machine learning are geared towards the financial domain. And that makes sense – this is the ultimate numbers field. A lot of banking institutions till recently used to lean on logistic regression (a simple machine learning algorithm) to crunch these numbers.

There are tons of use cases of machine learning in finance. Let's look at two very common ones you (most likely) have come across.

Catching Fraud in Banking

Have you ever been a victim of credit card fraud? It's a painful experience to go through. The shock of the fraud is exacerbated by the amount of paperwork the bank asks you to fill out.

Thankfully, machine learning is solving different layers of this process. From fraud detection to fraud prevention, machine learning algorithms are changing the way banks work to improve the customer's experience.

The challenge with this is keeping up with the level of cyber threats. These adversaries are two steps ahead of the curve at each stage. As soon as the latest machine learning solution comes up, these attackers perfect it and build on top of it.

Having said that, machine learning has definitely helped streamline the process. These algorithms are able to identify fraudulent transactions and flag them so the bank can connect with the customers ASAP to check if they made the transaction.

A good example is to look at the spending patterns of consumers. If a purchase does not fit in with this pattern (the amount is too high, or from a different country, etc.), then the algorithms alert the bank and put the transaction on hold.

Personalized Banking

Another use case of recommendation engines! This one is targeted specifically for the banking domain. You must be quite familiar with personalization at this point – so think about what personalized banking could mean before you read further.

We have read about banks targeting customer micro segments and tailoring offers to them. Personalized banking takes this concept to an entirely new level.

The ideal personalization scenario is using machine learning to anticipate the user's need and targeting segments of each individual. As a report by BCG states:

> *"Personalization in banking is not primarily about selling. It's about providing service, information, and advice, often on a daily basis or even several times a day. Such interactions, as opposed to infrequent sales communications, form the crux of the customer's banking experience."*

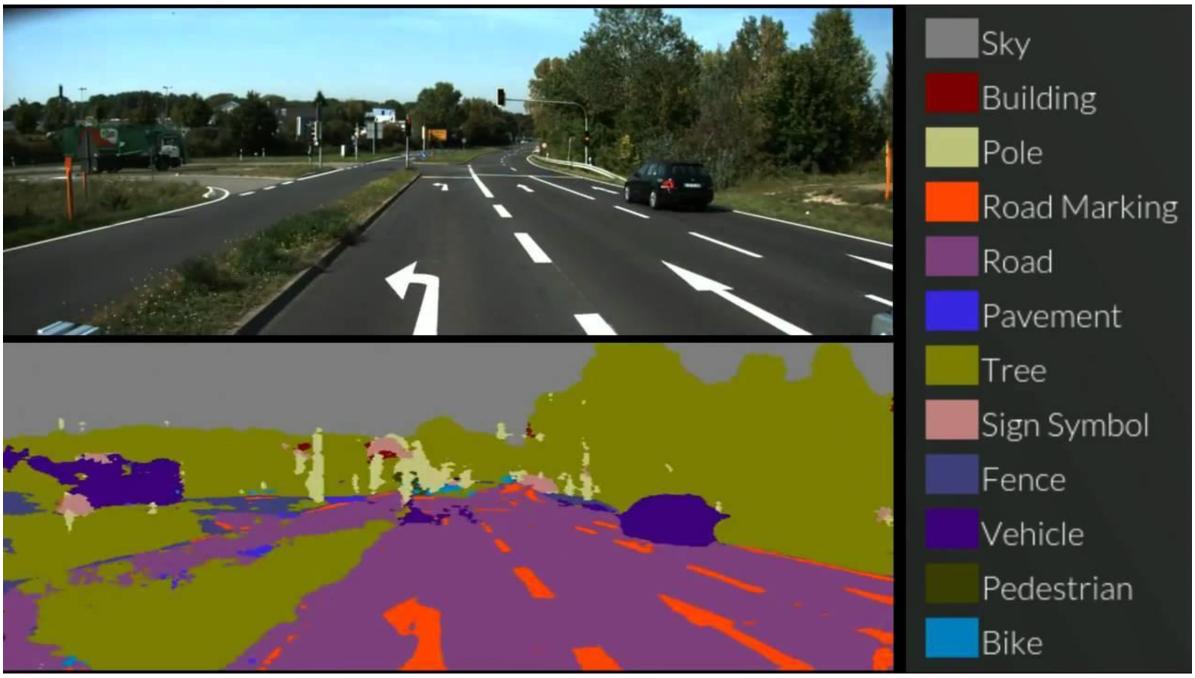
Other Popular Machine Learning Use Cases

I wanted to include this section of machine learning use cases that did not quite fit in our above categories. I will constantly update this section so let's start by looking at a very interesting use case – self-driving cars!

SELF-DRIVING CARS

Out of all the use cases we have covered in this article, self-driving cars fascinate me the most. It is a crowning achievement of what we have been able to accomplish using hardware and machine learning.

The beauty of self-driving cars is that all the three main aspects of machine learning – supervised, unsupervised and reinforcement learning – are used throughout the car's design.



Here are just a few features of self-driving cars where machine learning is used:

- Detecting objects around the car
- Detecting the distance between the car in front, where the pavement is located, and the traffic signal
- Evaluating the condition of the driver
- Scene classification, among many other things.

How is Machine Learning Different from Statistical Modeling?

At this juncture – it is important to understand how Machine Learning is different from some of the earlier used quant techniques. The most commonly used quant technique in businesses in the past was **"Statistical Modeling"**.

So, how are machine learning and statistical modeling different?

Let's start with simple definitions:

Machine Learning is …AN ALGORITHM THAT CAN LEARN FROM DATA WITHOUT RELYING ON RULES-BASED PROGRAMMING.

Statistical Modeling is …the formalization of relationships between variables in the form of mathematical equations.

For people like me, who enjoy understanding concepts from practical applications, these definitions don't help much. So, let's look at a business case here.

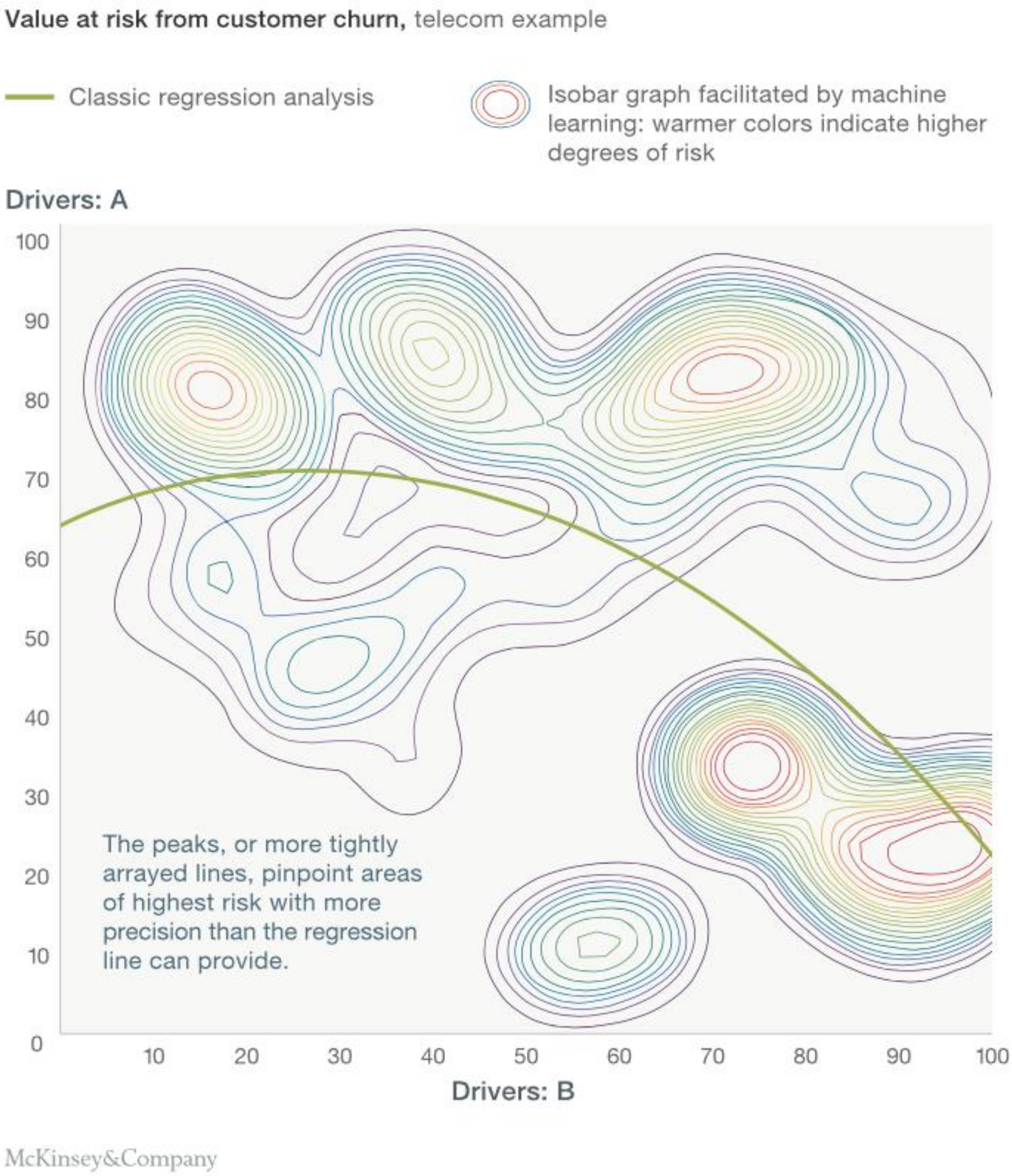
A Business Case

Let us now see an interesting example (published by McKinsey) differentiating the two algorithms :

Case : Understand the risk level of customer churn over a period of time for a Telecom company.

Data Available : Two Drivers – A & B.

What McKinsey shows next is an absolute delight! Just look at the below graph to understand the difference between a statistical model and a Machine Learning algorithm.



What did you observe from the above graph? A statistical model is all about getting a simple formulation of a frontier in a classification model problem. Here, we see a nonlinear boundary which to some extent separates risky people from non-risky people. But when we see the contours generated by the Machine Learning algorithm, we witness **that statistical modelling is in no way comparable to the problem in hand to the Machine Learning algorithm**. The contours of machine learning seem to capture all patterns beyond any boundaries of linearity or even continuity of the boundaries. This is what Machine Learning can do for you.

If this is not inspiring enough, then how about this - machine learning algorithms are used in recommendation engines of YouTube / Google etc. which can churn trillions of observations in a second to come up with almost a perfect recommendation. Even with a laptop of 16 GB RAM I daily work on datasets of millions of rows with thousands of parameters and build an entire model in not more than 30 minutes. A statistical model on the other hand needs a supercomputer to run a million observations with thousands of parameters.

Differences between Machine Learning and Statistical Modeling:

Given the flavour of difference in the output of these two approaches, let us understand the difference in the two paradigms, even though both do almost a similar job :

- Schools they come from
- When did they come into existence?
- Assumptions they work on
- Type of data they deal with
- Nomenclatures of operations and objects
- Techniques used
- Predictive power and human effort involved to implement

All the differences mentioned above do separate the two to some extent, but there is no hard boundary between Machine Learning and statistical modelling.

They belong to different schools

Machine Learning is …

a subfield of computer science and artificial intelligence which deals with building systems that can learn from data, instead of explicitly programmed instructions.

Statistical Modelling is …

a subfield of mathematics which deals with finding relationships between variables to predict an outcome.

They came up in different eras

Statistical modeling has been there for centuries now. However, machine learning is a very recent development. It came into existence in the 1990s as steady advances in digitization and cheap computing power enabled data scientists to stop building finished models and instead train computers to do so. The unmanageable volume and complexity of big data that the world is now swimming in have increased the potential of machine learning—and the need for it.

Extent of assumptions involved

Statistical modeling works on a number of assumptions. For instance, a linear regression assumes :

1. Linear relation between independent and dependent variable

2. Homoscedasticity
3. Mean of error at zero for every dependent value
4. Independence of observations

5. Error should be normally distributed for each value of dependent variable

Similarly, Logistic regression comes with its own set of assumptions. Even a nonlinear model has to comply to a continuous segregation boundary. Machine Learning algorithms do assume a few of these things but in general are spared from most of these assumptions. The biggest advantage of using a Machine Learning algorithm is that there might not be any continuity of boundary as shown in the case study above. Also, we need not specify the distribution of dependent or independent variables in a machine learning algorithm.

Types of data they deal with

Machine Learning algorithms are wide range tools. Online Learning tools predict data on the fly. These tools are capable of learning from trillions of observations one by one. They make predictions and learn simultaneously. Other algorithms like Random Forest and Gradient Boosting are also exceptionally fast with big data. Machine learning does really well with wide (high number of attributes) and deep (high number of observations). However statistical models are generally applied on smaller data with less attributes or they end up over fitting.

Naming Convention

Here are names which refer to almost the same things :

Machine learning	Statistics
network, graphs	model
weights	parameters
learning	fitting
generalization	test set performance
supervised learning	regression/classification
unsupervised learning	density estimation, clustering

Formulation

Even when the end goal for both machine learning and statistical modeling is the same, the formulation of the two is significantly different.

In a statistical model, we basically try to estimate the function f in:

Dependent Variable (Y) = f(Independent Variable) + error function

Machine Learning takes away the deterministic function "f" out of the equation. It simply becomes:

Output(Y) ----- > Input (X)

It will try to find pockets of X in n dimensions (where n is the number of attributes), where occurrence of Y is significantly different.

Predictive Power and Human Effort

Nature does not assume anything before forcing an event to occur.

So the lesser assumptions in a predictive model, higher will be the predictive power. Machine Learning, as the name suggests, needs minimal human effort. Machine learning works on iterations where a computer tries to find patterns hidden in the data. Because the machine does this work on comprehensive data and is independent of all the assumptions, predictive power is generally very strong for these models. Statistical models are mathematics intensive and based on coefficient estimation. They require the modeler to understand the relation between variables before putting them in.

When to apply Machine Learning and When to apply Statistical Modeling?

Now that you understand both these powerful techniques – the obvious questions is when to apply which tools? The rule of thumb you can generally use is when you need to provide explanations for your decisions – statistical modeling will come in more handy. For example, when making decisions for Credit decisioning in a bank – Statistical modeling is far more effective. On the other hand, if you are powering a recommendation engine on your website – machine learning is far more effective.

Deep Learning – A new Lifeline

Deep Learning is nothing but a paradigm of machine learning which has shown incredible promise in recent years. This is because of the fact that Deep Learning shows great analogy with the functioning of the human brain. The superiority of the human brain is an evident fact, and it is considered to be the most versatile and efficient self-learning model that has ever been created.

Let us understand the functioning of a deep learning model with an example:

What do you see in the above image?

The most obvious answer would be “a car”, right? Despite the fact that there is sand, greenery, clouds and a lot of other things, our brain tags this image as one of a car. This is because our brain has learnt to identify the primary subject of an image.

This ability of deriving useful information from a lot of extraneous data is what makes deep learning special. With the amount of data that is being generated these days, we want our models to be better with more of this data being fed into it. And deep learning models get better with the increase in the amount of data.

Now, although Deep Learning has been around for many years, the major breakthroughs from these techniques came only in recent years. This is because of two main reasons – first and foremost, as we saw before, there is an increase in the amount of data generated through various sources. The second is the growth in hardware resources required to run these models. GPUs, which are becoming a requirement to run deep learning models, are multiple times faster and they help us build bigger and deeper deep learning models in comparatively lesser time than we required previously.

This is the reason that Deep Learning has become a major buzz word in the data science industry.

IS DEEP LEARNING JUST HYPE OR DOES IT HAVE REAL-LIFE APPLICATIONS?

Deep Learning has found many practical applications in the recent past. From Netflix's famous movie recommendation system to Google's self-driving cars, deep learning is already transforming a lot of businesses and is expected to bring about a revolution in almost all industries. Deep learning models are being used from diagnosing cancer to winning presidential elections, from creating art and writing literature to making real life money. Thus it would be wrong to say that it is just a hyped topic.

Some major applications of deep learning that are being employed by technology companies are:

- Google and Facebook are **translating text** into hundreds of languages at a time. This is being done through some deep learning models being applied to NLP tasks and is a major success story.
- Conversational agents like Siri, Alexa, Cortana basically work on **simplifying speech recognition** techniques through LSTMs and RNNs. Voice commands have added a whole new domain to the possibilities of a machine.
- Deep learning is being used in **impactful computer vision applications** such as OCR (Optical Character Recognition) and real time language translation
- Multimedia sharing apps like Snapchat and Instagram apply algorithms using **facial features** which is another application of deep learning.
- Deep Learning is being used **in Healthcare domain to locate malignant cells** and other foreign bodies in order to detect complex diseases.

However, some people develop a thinking that deep learning is overhyped because of the fact that labeled data required for training deep learning models is not readily available. Even if the data is available, the computational power required to train such models does not come cheap. Hence, due to these barriers, people are not able to experience the power of deep learning and term it as just hype.

Go through the following blog to build some real life deep learning applications yourself:

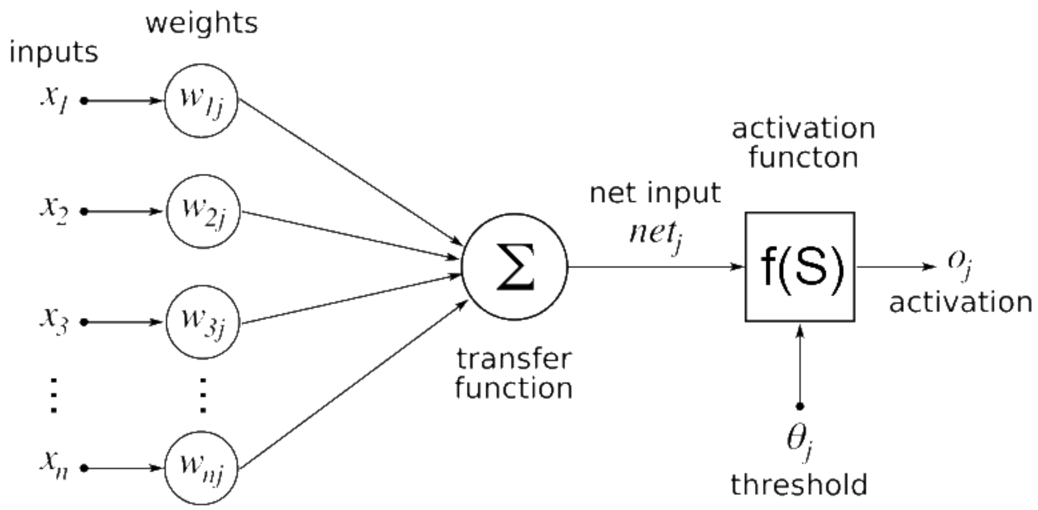
- 6 Deep Learning Applications a beginner can build in minutes (using Python)

WHY ARE GPUS NECESSARY FOR BUILDING DEEP LEARNING MODELS?

When you train a deep learning model, two main operations are performed:

- Forward Pass
- Backward Pass

In forward pass, input is passed through the neural network and after processing the input, an output is generated. Whereas in backward pass, we update the weights of the neural network on the basis of error we get in forward pass.



Both of these operations involve matrix multiplication. A simple matrix multiplication can be represented by the image below:

$$\begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix} \begin{bmatrix} 7 & 8 \\ 9 & 10 \\ 11 & 12 \end{bmatrix} = \begin{bmatrix} 58 & \\ & \end{bmatrix}$$

$$1 \cdot 7 + 2 \cdot 9 + 3 \cdot 11 = 58$$

Here, we can see that each element in one row of the first array is multiplied with one column of the second array. So, in a neural network, we can consider first array as input to the neural network, and the second array can be considered as the weights of the network.

This seems to be a simple task. Now just to give you a sense of what kind of scale deep learning – VGG16 (a convolutional neural network of 16 hidden layers which is frequently used in deep learning applications) has ~140 million parameters; aka weights and biases. Now think of all the matrix multiplications you would have to do to pass just one input to this network! It would take years to train this kind of system if we take traditional approaches.

We saw that the computationally intensive part of a neural network is made up of multiple matrix multiplication. So how can we make it faster?

We can do this by performing all the operations at the same time instead of doing it one after the other. This, in a nutshell, is why we use a GPU (graphics processing units) instead of a CPU (central processing unit) for training a neural network.

When (and where) to apply Neural Networks ?

Deep Learning has been in the spotlight for quite some time now. Its "deeper" versions are making tremendous breakthroughs in many fields such as image recognition, speech and natural language processing, etc.

Now that we know it is so impactful; the main question that arises is when to and when not to apply neural networks? This field is like a gold mine right now, with many discoveries uncovered every day. And to be a part of this "gold rush", you have to keep a few things in mind:

- **Deep learning models require more labeled data to train.** I discussed before in the Cat Vs Dog classification example. A Deep Learning model needs more labeled Cats and Dogs images to train as compared to humans.

- **It is prudent to use Deep Learning for complex problems such as image processing.** Deep Learning algorithms belong to a class of algorithms called representation learning algorithms. These algorithms break down complex problems into simpler form so that they become understandable (or "representable"). Think of it as chewing food before you gulp. This would be harder for traditional (non-representation learning) algorithms.

- **When you have an appropriate type of deep learning to solve the problem.** Each problem has its own twists. So the data decides the way you solve the problem. For example, if the problem is of sequence generation, recurrent neural networks are more suitable. Whereas, if it is image related problem, you would probably be better of taking convolutional neural networks for a change.

- **Last but not the least, hardware requirements are essential for running a deep neural network model.** Neural nets were "discovered" long ago, but they are shining in recent years for the main reason that computational resources are better and more powerful. If you want to solve a real-life problem with these networks, get ready to buy some high-end hardware!

DO WE NEED A LOT OF DATA TO TRAIN DEEP LEARNING MODELS?

It is true that we need a large amount of data to train a typical deep learning model. But we can generally overcome this by using something called transfer learning. Let me explain thoroughly.

One of the barriers for using deep learning models for industry applications is where the data is not in huge amount. A few examples of data needed to train some of the popular deep learning models are:

	Google's Neural Machine Translation	VGG Network	DeepVideo
Objective	Text Translation	Image Category Classification	Video Category Classification
Data Size	6M pairs of English-French sentences	1.2M images with labeled categories	1.1M videos with labeled categories

Parameters	380M	140M	About 100M

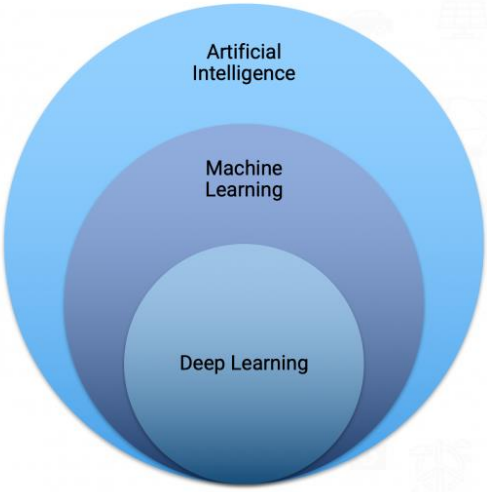
However, a deep learning model trained on a specific task can be reused for different problems in the same domain even if the amount of data is not that huge. This technique is known as **Transfer Learning**.

For instance, we have a set of 1000 images of cats and dogs labeled as 1 and 0 (1 for cat and 0 for dog) and we have another set of 500 test images that we need to classify. So, instead of training a deep learning model on the data of 1000 images, we can use a **pre-trained** VGGNet model and retrain it on our data and use it to classify the unlabeled set of images. A pre-trained model may not be 100% accurate in your application, but it saves huge efforts required to reinvent the wheel.

You may have a look at this article to get a better intuition of using a pre-trained model.

How is Machine Learning Different from Deep Learning?

Deep learning is actually a sub-field of Machine Learning. So, if you were to represent Machine Learning and Deep Learning by a simple Venn-diagram – it will look like this:

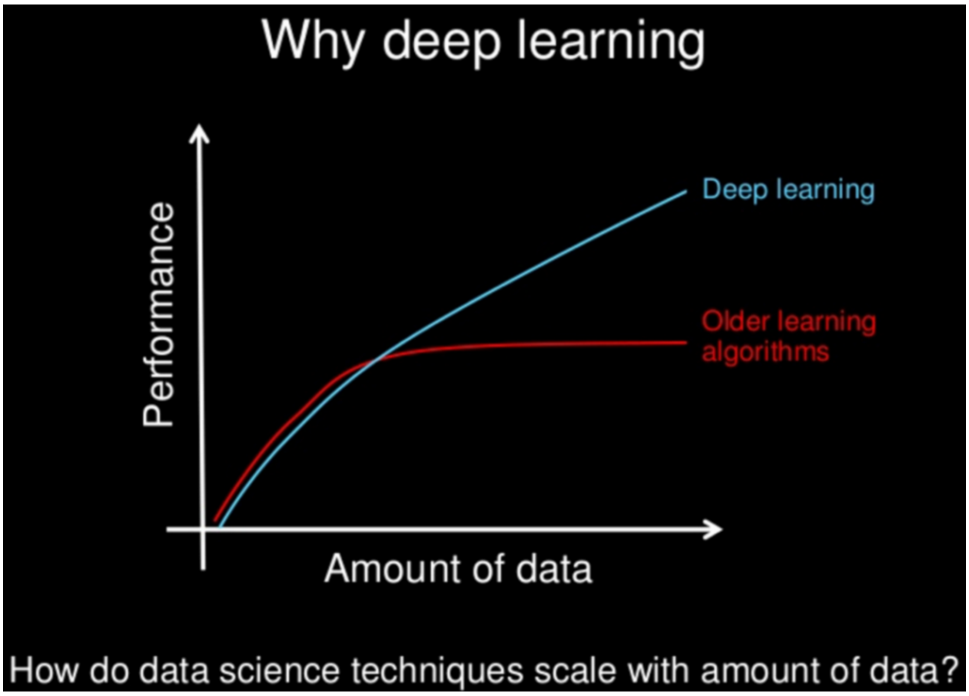


Comparison of Machine Learning and Deep Learning

Now that you have understood an overview of Machine Learning and Deep Learning, we will take a few important points and compare the two techniques.

Data dependencies

The most important difference between deep learning and traditional machine learning is its performance as the scale of data increases. When the data is small, deep learning algorithms don't perform that well. This is because deep learning algorithms need a large amount of data to understand it perfectly. On the other hand, traditional machine learning algorithms with their handcrafted rules/ features prevail in this scenario. The below image summarizes this fact.



Hardware dependencies

Deep learning algorithms heavily depend on high-end machines, contrary to traditional machine learning algorithms, which can work on low-end machines. This is because the requirements of deep learning algorithms include GPUs which are an integral part of its working. Deep learning algorithms inherently do a large amount of matrix multiplication operations. These operations can be efficiently optimized using a GPU because GPU is built for this purpose.

Feature engineering

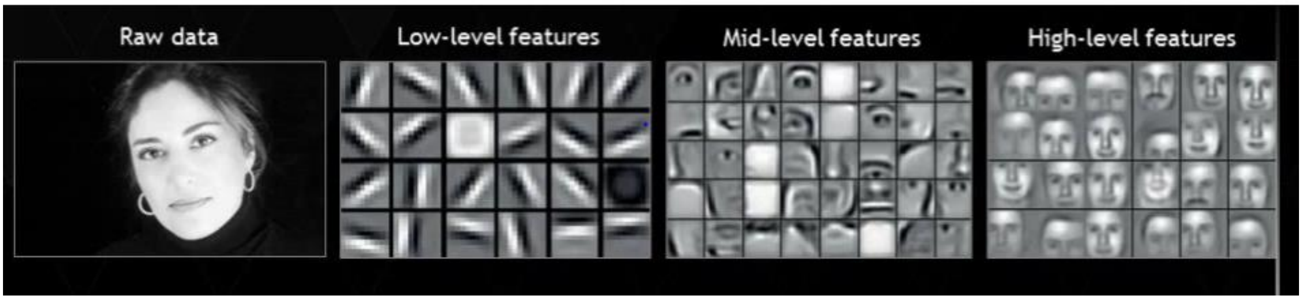
Feature engineering is a process of putting domain knowledge into the creation of feature extractors to reduce the complexity of the data and make patterns more visible for learning algorithms to work. This process is difficult and expensive in terms of time and expertise.

In Machine learning, most of the applied features need to be identified by an expert and then hand-coded as per the domain and data type.

For example, features can be pixel values, shapes, textures, position and orientation. The performance of most Machine Learning algorithms depends on how accurately the features are identified and extracted.

Deep learning algorithms learn these features itself from the data. This is a very distinctive part of Deep Learning and a major step ahead of traditional Machine Learning. Therefore, deep learning reduces the task of developing a new feature extractor for every problem.

Like, Convolutional NN will try to learn low-level features such as edges and lines in early layers then parts of faces of people and then high-level representation of a face.

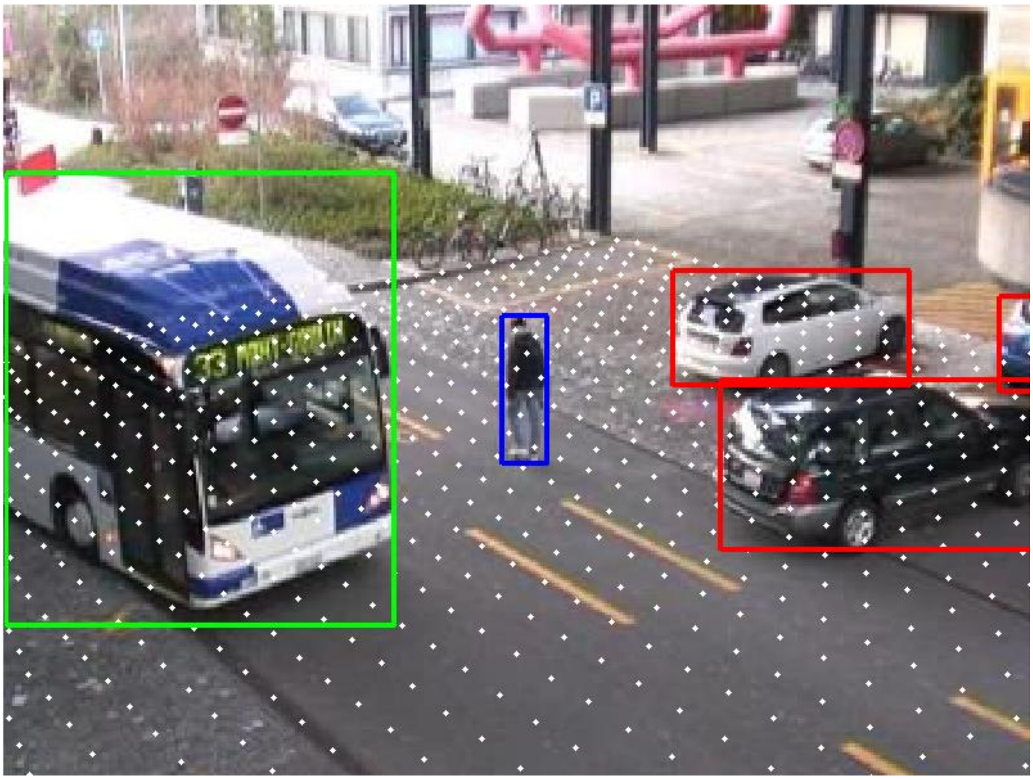


Problem Solving approach

When solving a problem using traditional machine learning algorithms, it is generally recommended to break the problem down into different parts, solve them individually and combine them to get the result. Deep learning in contrast advocates to solve the problem end-to-end.

Let's take an example to understand this.

Suppose you have a task of multiple object detection. The task is to identify what is the object and where it is present in the image.



In a typical machine learning approach, you would divide the problem into two steps - object detection and object recognition. First, you would use a bounding box detection algorithm like grabcut to skim through the image and find all the possible objects. You would then use object recognition algorithms like SVM with HOG to recognize relevant objects.

On the contrary, in a deep learning approach, you would do the process end-to-end. For example, in a YOLO net (which is a type of deep learning algorithm), you would pass in an image, and it would give out the location along with the name of the object.

Execution Time

Usually, a deep learning algorithm takes a long time to train. This is because there are so many parameters in a deep learning algorithm that training them takes longer than usual. State of the art deep learning algorithm ResNet takes about two weeks to train completely from scratch. Whereas machine learning comparatively takes much less time to train, ranging from a few seconds to a few hours.

This in turn is completely reversed on testing time. At test time, deep learning algorithm takes much less time to run. Whereas, if you compare it with k-nearest neighbors (a type of machine learning algorithm), test time increases on increasing the size of data. Although this is not applicable on all machine learning algorithms, as some of them have small testing times too.

Interpretability

Last but not the least, we have interpretability as a factor for comparison of machine learning and deep learning. This factor is the main reason deep learning is still thought 10 times before its use in industry.

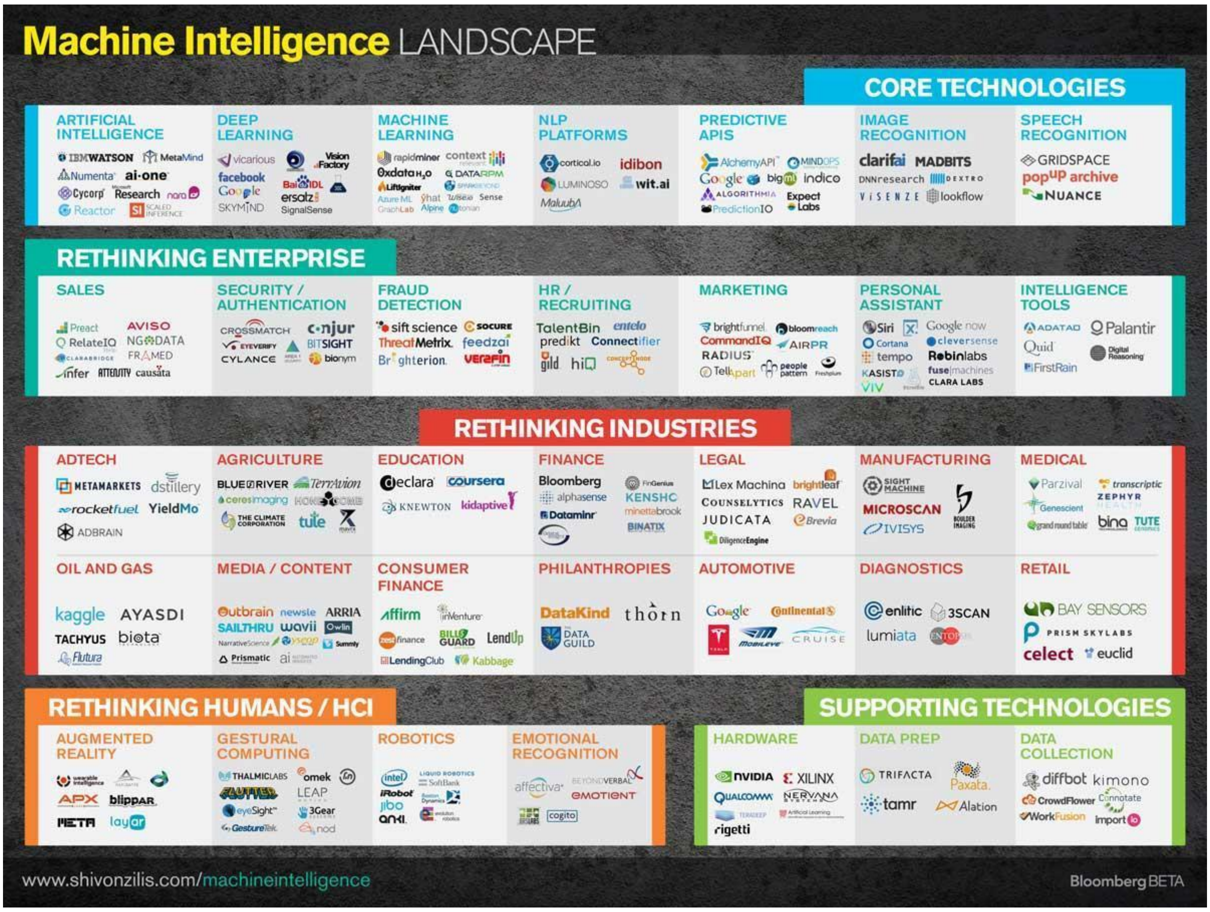
Let's take an example. Suppose we use deep learning to give automated scoring to essays. The performance it gives in scoring is quite excellent and is near human performance. But there is an issue. It does not reveal why it has given that score. Indeed mathematically you can find out which nodes of a deep neural network were activated, but we don't know what the neurons were supposed to model and what these layers of neurons were doing collectively. So we fail to interpret the results.

On the other hand, machine learning algorithms like decision trees give us crisp rules as to why it chose what it chose, so it is particularly easy to interpret the reasoning behind it. Therefore, algorithms like decision trees and linear/logistic regression are primarily used in industry for interpretability.

WHERE IS MACHINE LEARNING AND DEEP LEARNING BEING APPLIED RIGHT NOW?

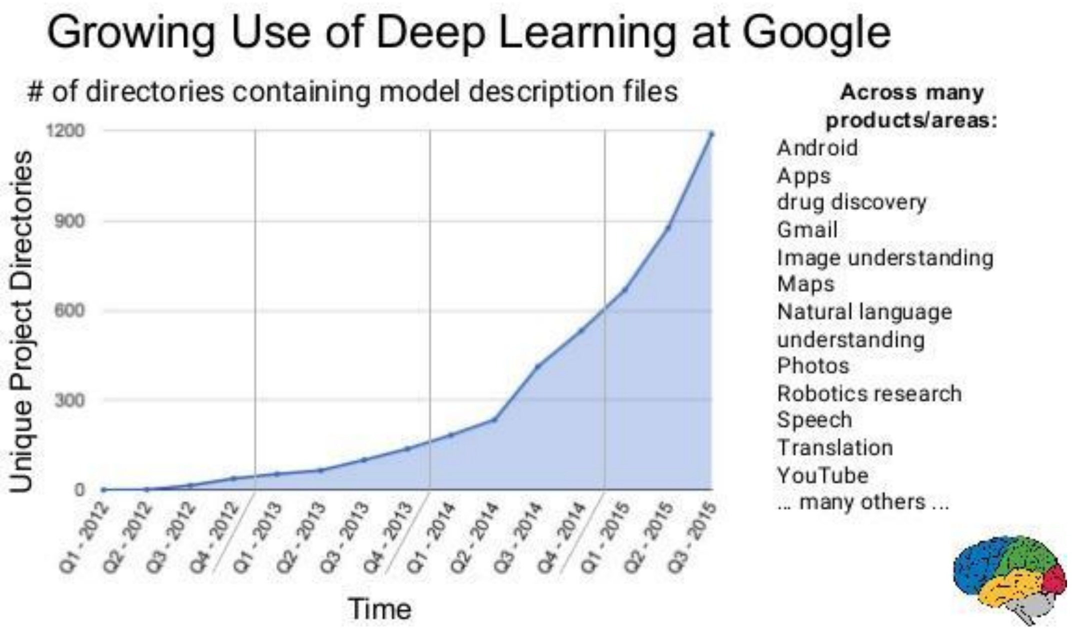
The wiki article gives an overview of all the domains where machine learning has been applied. These include:

- Computer Vision: for applications like vehicle number plate identification and facial recognition.
- Information Retrieval: for applications like search engines, both text search, and image search.
- Marketing: for applications like automated email marketing, target identification
- Medical Diagnosis: for applications like cancer identification, anomaly detection
- Natural Language Processing: for applications like sentiment analysis, photo tagging
- Online Advertising, etc.



The image given above aptly summarizes the application areas of machine learning. Although it covers the broader topic of machine intelligence as a whole.

One prime example of a company using machine learning / deep learning is Google.



In the above image, you can see how Google is applying machine learning in its various products. Applications of Machine Learning/Deep Learning are endless, you just have to look at the right opportunity!

Future Trends

The above section would have given you an overview of Machine Learning and Deep Learning and the difference between them. In this section, I'm sharing my views on how Machine Learning and Deep Learning would progress in the future.

- First of all, seeing the increasing trend of using data science and machine learning in the industry, it will become increasingly important for each company who wants to survive to inculcate Machine Learning in their business. Also, each and every individual would be expected to know the basic terminologies.
- Deep learning is surprising us each and every day, and will continue to do so in the near future. This is because Deep Learning is proving to be one of the best techniques to be discovered with state-of-the-art performances.
- Research is continuous in Machine Learning and Deep Learning. But unlike in previous years, where the research was limited to academia, research in Machine Learning and Deep Learning is exploding in both industry and academia. And with more funds available than ever before, it is more likely to be a keynote in human development overall.

Now that we understand the difference between Machine Learning and Deep Learning – let us see a few areas in the field of Computer Science where Machine Learning and Deep Learning have made a huge difference in the last few years. This includes, but is not restricted to the areas of Computer Vision and Natural Language Processing – which we will briefly cover in the next 2 chapters.

NATURAL LANGUAGE PROCESSING – MACHINES CAN UNDERSTAND LANGUAGE!

Natural Language Processing is a branch of data science that consists of systematic processes for analyzing, understanding, and deriving information from the text data.

By using the techniques of Natural Language Processing, one can organize and analyze the massive chunks of text data, perform numerous automated tasks to solve a wide range of problems such as – automatic summarization, machine translation, and many more.

APPLICATIONS OF NATURAL LANGUAGE PROCESSING IN OUR DAY-TO-DAY-LIFE

Natural Language Processing is essentially teaching machines to understand human language and since it is all about human language, you come across multiple applications of NLP in your daily life without even realizing it!

Here are a few examples that you would have definitely come across:

CHATBOTS OR CONVERSATIONAL AGENTS

Chatbots are everywhere today, from booking your flight tickets to ordering food, chances are that you have already interacted with one.

Customers nowadays don't want to wait for hours just to get their queries resolved. They want instant answers and chatbots come in really handy here, both for your business as well as your customers.

Similarly, we have many conversational agents or AI assistants like Alexa, Siri, Cortana and Google Home that use natural language processing internally.

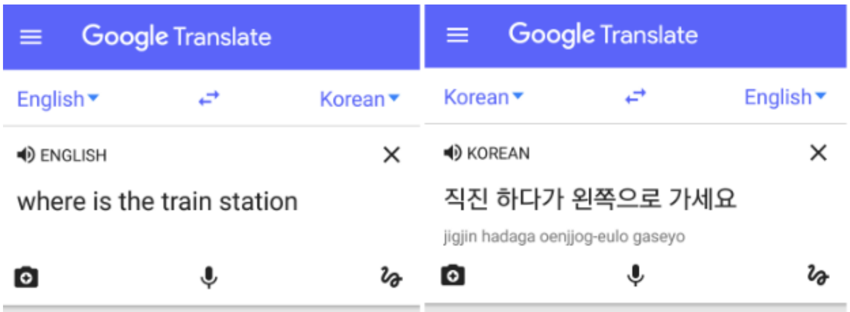
If you curious about Chatbots and want to learn how they work under the hood or how can you build one? You can follow these tutorials below:

- [A Guide to Building an Intelligent Chatbot for Slack using Dialogflow API]
- [Building a FAQ Chatbot in Python – The Future of Information Searching]
- [Learn how to Build and Deploy a Chatbot in Minutes using Rasa (IPL Case Study!)]

MACHINE TRANSLATION

A machine translation system uses natural language processing techniques in collaboration with Machine learning/Deep learning to build systems that are capable of automatic language translation.

If you have ever read a post in another language on Facebook and seen its translation just below it, or opened a website of any language other than English in Chrome or even used Google Translate on a trip to a foreign country, then you have used some kind of a machine translation system.



Applications like Google Translate are a very good example of what we call as Neural Machine Translation, these are language translation systems that are built on top of Neural Networks. If you'd like to build an NMT system of your own, you can follow the below tutorial:

- [A Must-Read NLP Tutorial on Neural Machine Translation – The Technique Powering Google Translate]

SPEECH RECOGNITION

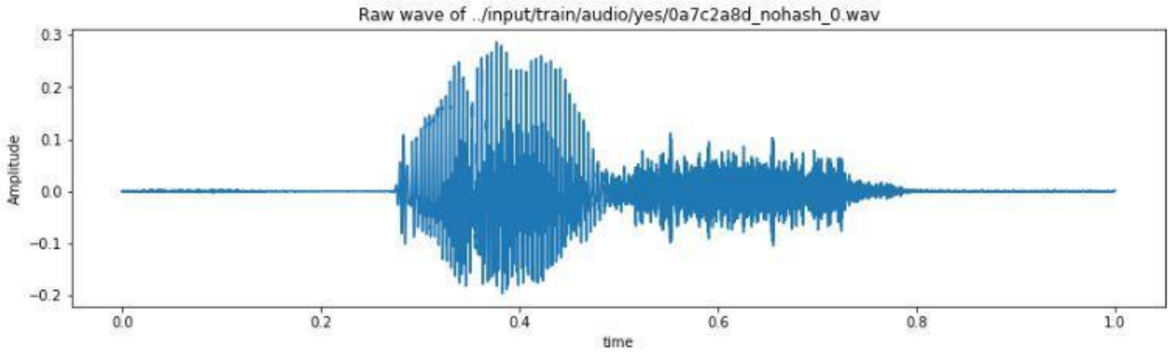
"Hey, Siri. What's the weather like today?"

(Source: Getty Images)

Voice-based personal assistants have become so ubiquitous in the past decade that almost every Smartphone user would be very familiar with the likes of Apple's Siri, Amazon's Alexa, Baidu's Deur, and Google's Assistant.

But have you ever wondered about the technology that powers these assistants under the hood?

Siri and Google are able to easily understand what you are saying, how does the system convert your query into text on your phone's screen?



This is exactly where Natural Language Processing comes in the picture. Every audio that you speak can be considered as "natural language" and if you can convert this audio to text then all you have to do is reuse the models and techniques you used for building the text-based NLP systems.

For example, if you want to build a voice-based AI assistant, then one of the approach can be to convert the audio interaction (commands) into text and feed it into the text-based Chatbot that you have. From there onwards, your Chatbot can take over the interaction with the person and you just have to take care of the part where you convert from text to speech and vice-versa.

Similarly, speech recognition becomes simple text classification once you are able to convert the audio to text. The same is true for language translation.

Now knowing all this, if you are curious about how can you make your first text to speech system in Python then you should read the following article:

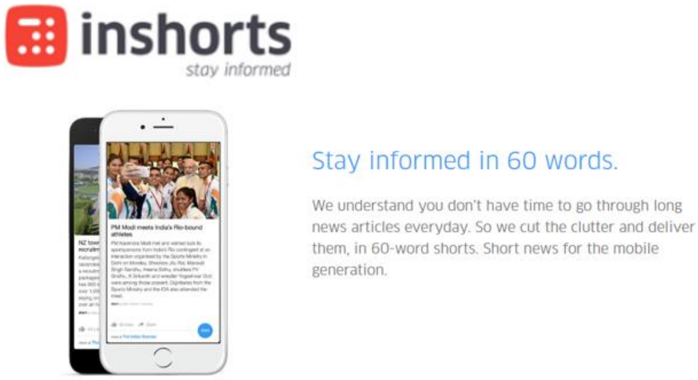
- Learn how to Build your own Speech-to-Text Model (using Python)

Text Summarization

In the busy world of today, people need byte-sized summaries of information in order to effectively take action on it without indulging time more than necessary. Text summarization is one such application of Natural Language Processing (NLP) that is slowly becoming the need of the hour.

Who has the time to go through a myriad of articles/documents/books to decide whether they are useful or not? Wouldn't it help if instead of reading long medical or legal reports, you can simply get the summary of the proceedings with important points?

Thankfully – this technology is already here.



[Automatic Text Summarization]() is one of the most interesting problems in the field of Natural Language Processing (NLP). It is a process of generating a concise, coherent and meaningful summary of text from text resources such as books, news articles, blog posts, research papers, etc.

Applications of Natural Language Processing in the Industry

Now that you are already familiar with NLP based applications that you use in your daily life as a layman, let's understand what kind of applications of NLP create value for businesses!

Sentiment Analysis for Customer Reviews

Natural Language Processing (NLP) is a hotbed of research in data science these days and one of the most common applications of NLP is sentiment analysis. From opinion polls to creating entire marketing strategies, this domain has completely reshaped the way businesses work, which is why this is an area every data scientist must be familiar with.

Thousands of text documents can be processed for the sentiment (and other features including named entities, topics, themes, etc.) in seconds, compared to the hours it would take a team of people to manually complete the same task.

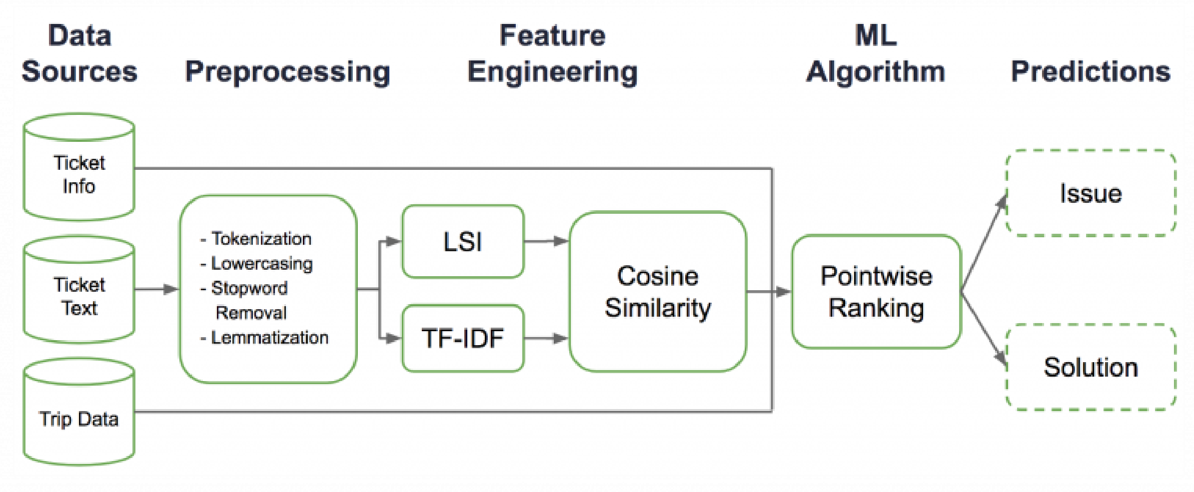
If you want to learn how can you mine user reviews or analyze sentiment from people's tweets, you can follow the below two articles:

- An NLP Approach to Mining Online Reviews using Topic Modeling (with Python codes)
- Comprehensive Hands on Guide to Twitter Sentiment Analysis with dataset and code

CUSTOMER SUPPORT SYSTEMS

Companies like Uber that are large-scale and customer-facing have developed sophisticated systems for customer support requests. With hundreds of thousands of tickets surfacing daily on the platform across 400+ cities worldwide, their team must ensure that agents are empowered to resolve them as accurately and quickly as possible.

Enter COTA, Uber's Customer Obsession Ticket Assistant, a tool that uses machine learning and natural language processing (NLP) techniques to help agents deliver better customer support.



COTA enables quick and efficient issue resolution for more than 90% of Uber's inbound support tickets. This is because you are not only responding to a customer in real-time (now the first response is from a bot rather than a human agent) but also because you are saving both the customer's and the agent's time by giving useful suggestions to the agent regarding the kind of problem the customer could be facing and the possible steps that could solve that problem. This is a very interesting use-case of natural language processing in the real world. You can read more about COTA here.

TEXT ANALYTICS

One of the biggest breakthroughs required for achieving any level of artificial intelligence is to have machines that can process text data. Thankfully, the amount of text data being generated in this universe has exploded exponentially in the last few years.

It has become imperative for an organization to have a structure in place to mine actionable insights from the text being generated. From social media analytics to risk management and cybercrime protection, dealing with text data has never been more important.

Now that you understand the impact and possibilities in Machine Learning, the obvious question you would be asking is how do you benefit from this technology. For doing that you not only need to know what is possible, but more importantly understand how a machine learning project is executed. You need to understand how do we marry machine learning with business knowledge to solve difficult business problems. This is what we will be discussing in the next chapter! Keep reading.

LIFE CYCLE OF A MACHINE LEARNING PROJECT

Till now – we have understood what is machine learning, its applications in different domains and the possibilities it opens up along with Deep Learning. The next logical question which comes up is:

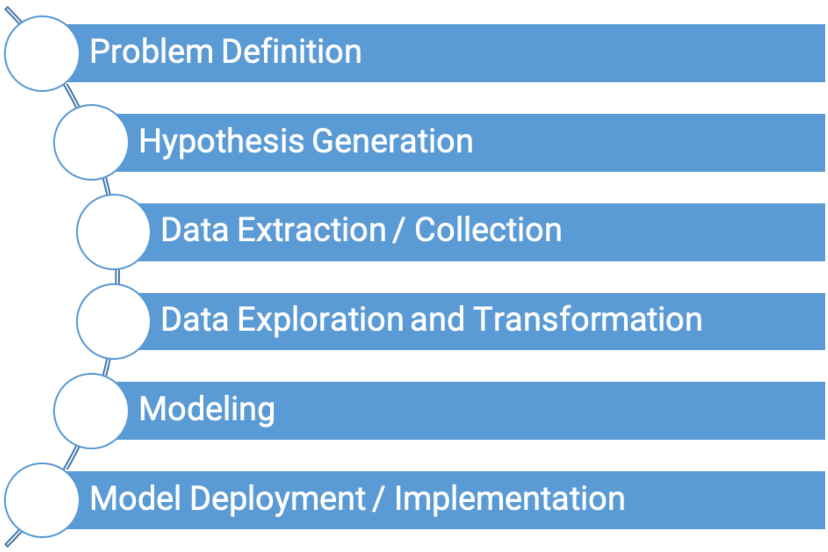
"*How do you solve a business problem using Machine Learning and Artificial Intelligence?*"

In order to answer this question, let me walk you through a typical life cycle of a Machine Learning project.

Any machine learning model development can be divided in 6 steps:

- **Problem definition** involves converting a Business Problem to a machine learning problem.
- **Hypothesis generation** is the process of creating possible business hypothesis and potential features for the model.
- **Data Collection** requires you to collect the data for testing your hypothesis and building the model.
- **Data Exploration** and cleaning helps you remove outliers, missing values and then **transform** data into required format.
- Modelling is where you actually build the **machine learning models.**
- Once built, you will **deploy the models.**

Here is a schematic of these 6 steps:



In order to understand these steps in more details, let me take a case study and share these steps in more details. Here is the business context you need for the case study:

> *ZipKart started as an ecommerce company focused on niche designer products. These products and designers were not served by the likes of Amazon and other companies till ZipKart came in the market. ZipKart grew at a fast pace and created a loyal set of customers looking for designer products. To fuel further growth, ZipKart expanded its product offering started selling larger variety of products, which are available on other sites. The idea was to increase Customer Life Time Value by selling them products while they are on ZipKart.*

> *You have been hired as a data scientist with an aim to increase customer lifetime value.*

Step 1 – Problem Definition

The first step in this process to solve this problem using machine learning, is to convert the business problem to data problem. Now, this can be done in a variety of ways. For example, each of these are valid data problems, which solve the same business problem in different ways:

- For the customers on ZipKart, can we provide them with a Flash offer which helps them purchase more than what they intend to buy initially?
- Segment the customers based on areas of interest based on past purchases and browsing history and then email them products of interest with offers.
- Can we predict which customers are likely to make a purchase in the next 24 hours and send them additional offers proactively?

The decision of which data problem to work on is usually based on the impact of the solution, ease of implementation, business preferences and the business strategy. This could be a matter of a different book altogether. For now, let us say that you along with the business owners zeroed in on the following data problem:

> *Create a Recommendation Engine based on customer profile, browsing history, past purchases and any other available data to tell top products a customer is likely to purchase. Once created, these recommendations can be used across the platform and across multiple communications.*

At this stage, we also discuss about the methodologies to evaluate model performance. To accomplish this we decide evaluation metric or validation data set.

STEP 2 – HYPOTHESIS GENERATION

Once the data problem is defined, the data scientist should sit down with the business to define various hypotheses with regards to the problem.

Wait – what is a hypothesis? Don't worry – if you don't know what is a hypothesis. Simply put, a hypothesis is a possible explanation or relationship which can help you solve the business problem.

For example, in this case following are some valid hypothesis:

- *Recommendations for the users will depend on their Age. Younger population would have different preferences compared to older population.*
- *Any product which a customer has browsed but not purchased in last 24 hours can be recommended to the customer*

HOW MANY HYPOTHESES DO YOU CREATE?

Now, if you are wondering how many hypotheses do you need to build your model. Before answering this question, let us try and understand why we are creating these hypothesis in the first place.

Each of these hypotheses is an attempt to capture the underlying data based relationships which might help us build a better model.

Given that context, you should try and build as many hypotheses as possible. You should not only apply your understanding and thinking, but you should also get all the stakeholders across business to collaborate and brainstorm on building these hypotheses.

A person involved in the Sales would have very different perspectives and inputs compared to someone in Operations. Hence, the more relationships you can capture, the better it would be.

In general, you need hundreds of hypothesis for models used in industry. Obviously, this is a function of the problem you are working on, the domain you are in and many other factors.

Key aspects of Hypothesis

There are 2 key components to every hypothesis you capture. Each hypothesis should have the following components:

- **What relationship do you expect?** This obviously can be wrong or there might be no hypothesis at all. But this helps you understand what variables would you need to capture and what features would you need to create.
- **How would you quantify the hypothesis?** What variables can help you quantify the hypothesis.

Step 3 – Data Extraction / Collection

Once you have a list of hypothesis, you would have a bucket list of all the data points you need to solve this problem. But, they may or may not be available within your Organization. Some of them might be available in the Organization, but the cost of getting them might be very high.

Hence, as a next step, you need to categorize each of these variables into 3 categories:

- Currently Available for project
- Collect for Future
- Not available

Once you have done this, it is time to get in touch with the stakeholders to understand who holds which data and how can you get the data.

Step 4 – Data Exploration / Exploratory Data Analysis

Once you have collected the data, now is the time to explore it. There are no shortcuts for data exploration. If you are in a state of mind, that machine learning can sail you away from every data storm, trust me, it won't. After some point of time, you'll realize that you are struggling at improving model's accuracy. In such situations, data exploration techniques will come to your rescue.

I can confidently say this, because I've been through such situations, a lot.

It is always a good practice to communicate your findings with stakeholders and check that it is also in sync with business understanding or not. Because, in some cases, due to wrong input/ data we may lead to incorrect conclusion.

STEP 5 – MODEL BUILD

After generating features and understanding data stories, we can go ahead with relevant features and build model(s). And, we can select the right algorithm(s) based on the type of problem and data.

We will discuss common machine learning algorithm in upcoming section.

STEP 6 – MODEL DEPLOYMENT

Questions to answer at this stage:

- How frequently would the model be refreshed?
- How will it be implemented?
- What could be the limitations of the model?

This is how a model deployment typically happens in real life. Once implemented, every model is monitored for performance and is improved as we continue to get more and more data.

Here is an exercise for you – please pick one of the problems in your business area and work through these 6 steps for that problem. How would you convert this business problem to data problem? What would be the various set of hypothesis you would have to solve this problem? What data do you need? What model would you build to solve this problem? How would you deploy the model?

HOW MUCH DATA IS REQUIRED TO TRAIN A MACHINE LEARNING MODEL?

There is no simple answer to this question. It depends on the problem you are trying to solve, the cost of collecting incremental data and the benefits coming from incremental data. But here are some guidelines:

- In general – you would want to collect as much data as possible. If the cost of collecting the data is not very high – this ends up working fine.
- If the cost of capturing the data is high, then you would need to do a cost-benefit analysis based on the expected benefits coming from machine learning models.
- The data being captured should be representative of the behaviour / environment you expect the model to work on.

WHAT KIND OF DATA IS REQUIRED TO TRAIN A MACHINE LEARNING MODEL?

Everything you see, hear and do is data. All you need is to capture that in the right manner.

Data is omnipresent these days. From logs on websites and smartphones to health devices – we are in constant process of creating data. In fact 90% of the data in this Universe has been created in the last 18 months.

Data can broadly be classified into 2 types:

1 **Structured Data** – Structured data typically refers to data stored in tabular format in databases in Organizations. This includes data about Customers, interactions with them and several other attributes, which flow through the IT infrastructure of Enterprises.

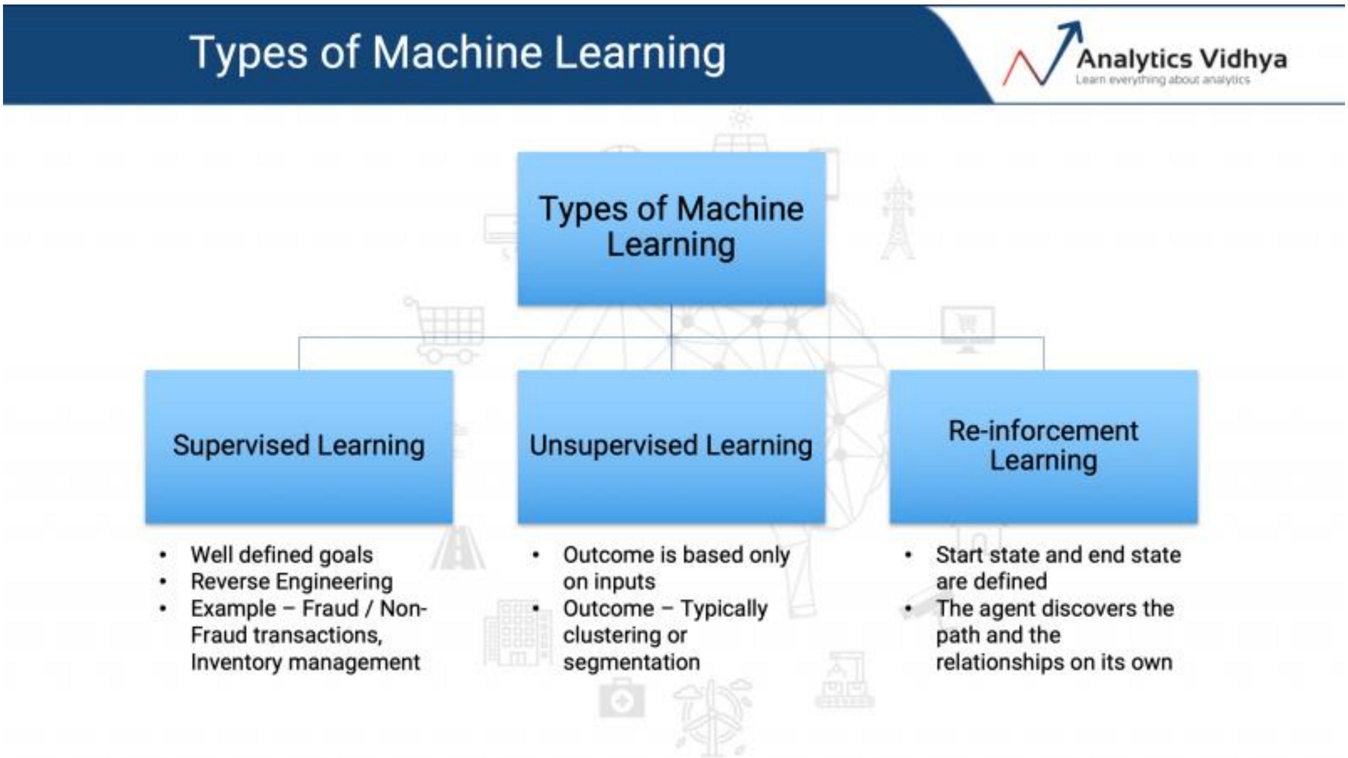
2 **Unstructured Data** – Unstructured Data includes all the data which gets captured, but is not stored in form of tables in enterprises. For example – letter of communications from customers or tweets and pictures from customers. It also includes images, voice records.

Machine Learning models can work on both Structured as well as Unstructured Data. However, you need to convert unstructured data to structured data first.

WHAT ARE THE KIND OF PROBLEMS WHICH CAN BE SOLVED USING MACHINE LEARNING?

Machine Learning problems can be divided into 3 broad classes:

- **Supervised Machine Learning** – When you have past data with outcomes (labels in machine learning terminology) and you want to predict the outcomes for future – you would use Supervised Machine Learning algorithms. Supervised Machine Learning problems can again be divided into 2 kinds of problems
  - **Classification Problems** – When you want to classify outcomes into different classes. For example – whether the floor needs cleaning / mopping is a classification problem. The outcome can fall in one of the classes – Yes or No. Similarly, whether a customer would default on their loan or not is a classification problem which is of high interest to any Bank.
  - **Regression Problem** – When you are interested in answering how much – these problems would be called Regression problem. For example – How much cleaning needs to be done is a Regression problem. Or what is the expected amount of default from a customer is a Regression problem.
- **Unsupervised Machine Learning** – There are times when you don't want to exactly predict an Outcome. You just want to perform a segmentation or clustering. For example – a bank would want to have segmentation of its customers to understand their behaviour. This is Unsupervised Machine Learning problem as we are not predicting any outcomes here.
- **Re-inforcement Learning –** Using this algorithm, the machine is trained to make specific decisions. It works this way: the machine is exposed to an environment where it trains itself continually using trial and error. This machine learns from past experience and tries to capture the best possible knowledge to make accurate business decisions. Example of Reinforcement Learning: Markov Decision Process. To know more about this learning, please refer to the article "Beginner's Guide to Reinforcement Learning".


Types of Machine Learning
Analytics Vidhya
Learn everything about analytics
Types of Machine Learning
Supervised Learning
Unsupervised Learning
Re-inforcement Learning
• Well defined goals
• Reverse Engineering
• Example – Fraud / Non-Fraud transactions, Inventory management
• Outcome is based only on inputs
• Outcome – Typically clustering or segmentation
• Start state and end state are defined
• The agent discovers the path and the relationships on its own

Common Machine Learning Algorithms - Simplified

Here is a list of commonly used machine learning algorithms. These algorithms can be applied to almost any data problem:

1. Linear Regression
2. Logistic Regression
3. Decision Tree
4. SVM
5. Naive Bayes
6. kNN
7. K-Means
8. Random Forest
9. Gradient Boosting algorithms
    1. GBM
    2. XGBoost
    3. LightGBM
    4. CatBoost

1. Linear Regression

It is used to estimate real values (cost of houses, number of calls, total sales etc.) based on continuous variable(s). We basically establish relationship between independent and dependent variables by fitting a best line. This best fit line is known as regression line and represented by a linear equation Y= a *X + b.

In this equation:

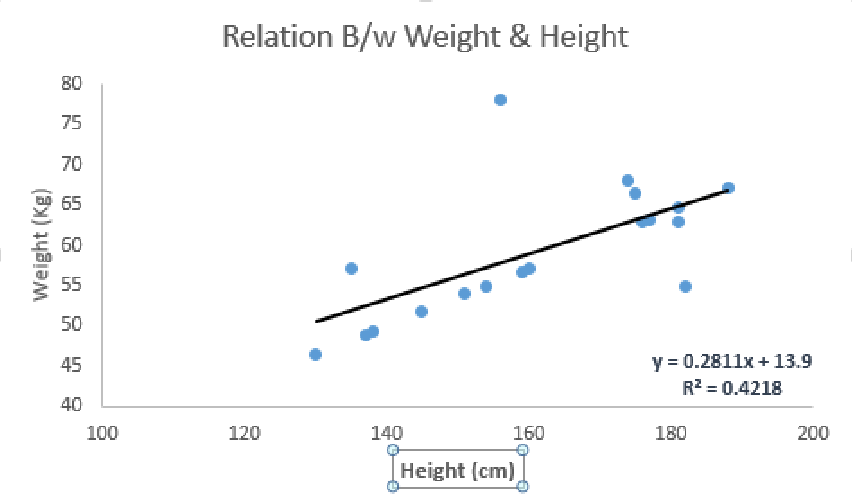
- Y – Dependent Variable
- a – Slope
- X – Independent variable
- b – Intercept

The best way to understand linear regression is to relive this experience of childhood. Let us say, you ask a child in fifth grade to arrange people in his class by increasing order of weight, without asking them their weights! What do you think the child will do? He / she would likely look (visually analyze) at the height and build of people and arrange them using a combination of these visible parameters. This is linear regression in real life!

The child has actually figured out that height and build would be correlated to the weight by a relationship, which looks like the equation above.

The coefficients a and b are derived based on minimizing the sum of squared difference of distance between data points and regression line.

Look at the below example. Here we have identified the best fit line having linear equation **y=0.2811x+13.9**. Now using this equation, we can find the weight, knowing the height of a person.



Linear Regression is of mainly two types: Simple Linear Regression and Multiple Linear Regression. Simple Linear Regression is characterized by one independent variable. And, Multiple Linear Regression (as the name suggests) is characterized by multiple (more than 1) independent variables. While finding best fit line, you can fit a polynomial or curvilinear regression. And these are known as polynomial or curvilinear regression.

2. Logistic Regression

Don't get confused by its name!

It is a classification not a regression algorithm. It is used to estimate discrete values (Binary values like 0/1, yes/no, true/false) based on given set of independent variable(s).

In simple words, it predicts the probability of occurrence of an event by fitting data to a logit function. Hence, it is also known as **logit regression**. Since, it predicts the probability, its output values lies between 0 and 1 (as expected).

Again, let us try and understand this through a simple example:

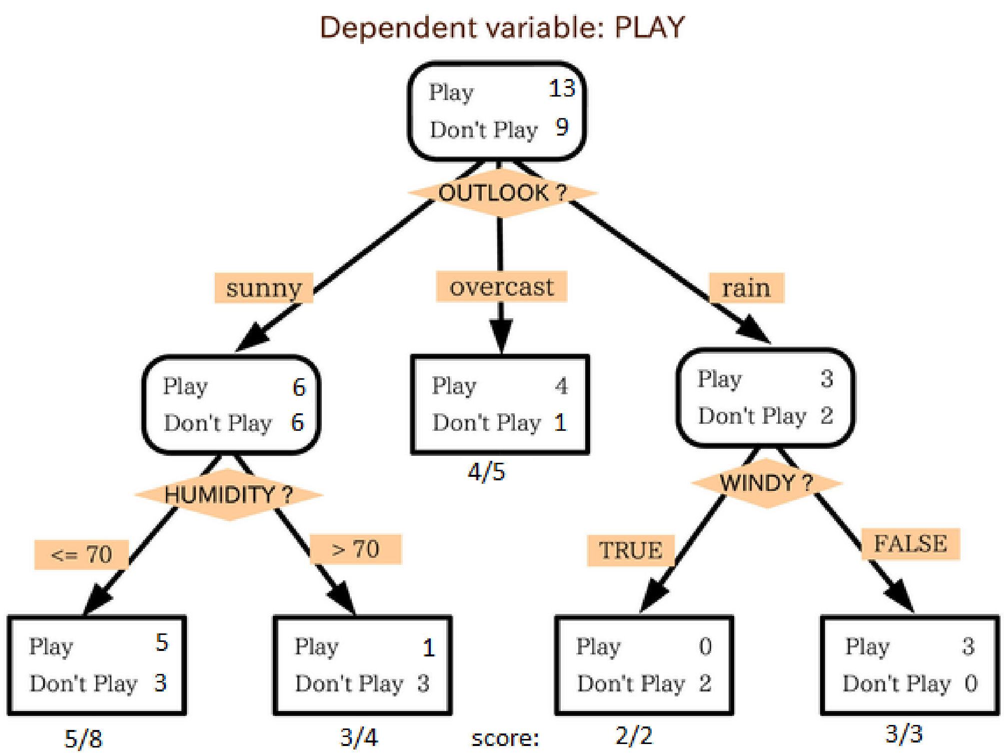
Let's say your friend gives you a puzzle to solve. There are only 2 outcome scenarios – either you solve it or you don't. Now imagine that you are being given a wide range of puzzles / quizzes in an attempt to understand which subjects you are good at.

The outcome of this study would be something like this – if you are given a trigonometric based tenth grade problem, you are 70% likely to solve it. On the other hand, if it is grade fifth history question, the probability of getting an answer is only 30%. This is what Logistic Regression provides you.

3. Decision Tree

This is one of my favorite algorithms and I use it quite frequently.

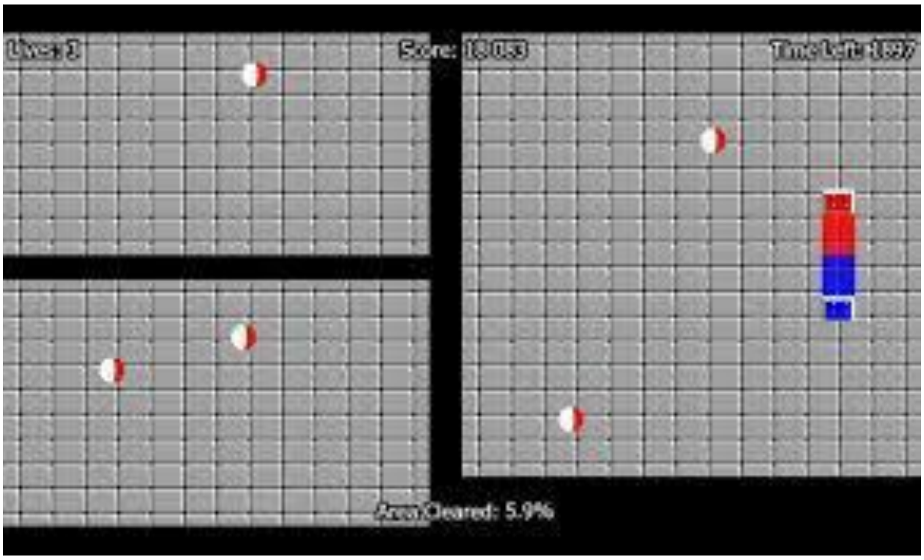
It works for both categorical and continuous dependent variables. In this algorithm, we split the population into two or more homogeneous sets. This is done based on most significant attributes/ independent variables to make as distinct groups as possible. For more details, you can read: Decision Tree Simplified.



source: statsexchange

In the image above, you can see that population is classified into four different groups based on multiple attributes to identify 'if they will play or not'. To split the population into different heterogeneous groups, it uses various techniques like Gini, Information Gain, Chi-square, entropy.

The best way to understand how decision tree works, is to play Jezzball – a classic game from Microsoft (image below). Essentially, you have a room with moving walls and you need to create walls such that maximum area gets cleared off without the balls.



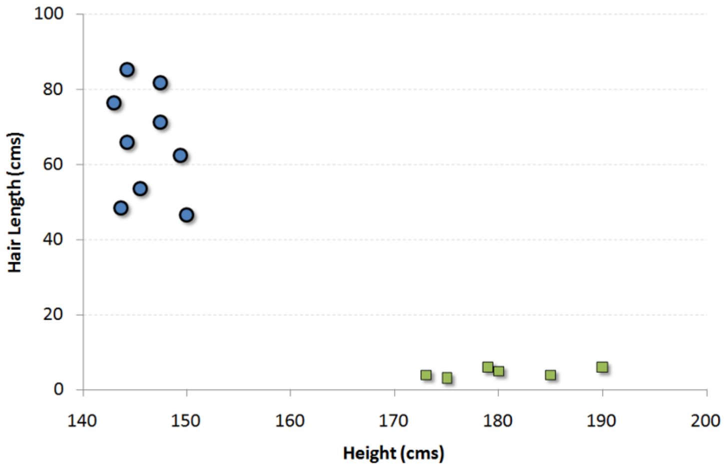
So, every time you split the room with a wall, you are trying to create 2 different populations within the same room. Decision trees work in very similar fashion by dividing a population in as different groups as possible.

MORE: [Simplified Version of Decision Tree Algorithms]

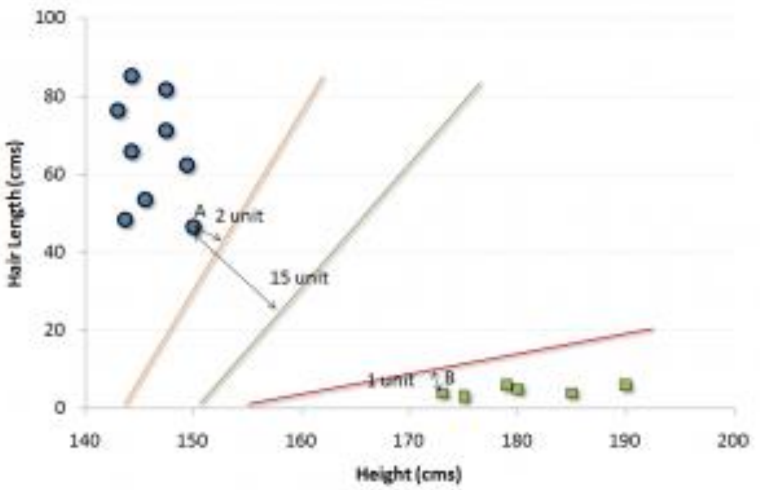
4. SVM (Support Vector Machine)

SVM is a classification method. In this algorithm, we plot each data item as a point in n-dimensional space (where n is the number of features you have) with the value of each feature being the value of a particular coordinate.

For example, if we only had two features like Height and Hair length of an individual, we'd first plot these two variables in two dimensional space where each point has two coordinates (these coordinates are known as **Support Vectors**).



Now, we will find some LINE that splits the data between the two differently classified groups of data. This will be a line such that the distances from the closest point in each of the two groups will be farthest away.



In the example shown above, the line which splits the data into two differently classified groups is the BLACK line, since the two closest points are the farthest apart from the line. This line is our classifier. Then, depending on where the testing data lands on either side of the line, that's what class we can classify the new data as.

More: Simplified Version of Support Vector Machine

Think of this algorithm as playing JezzBall in n-dimensional space. The tweaks in the game are:

- You can draw lines / planes at any angles (rather than just horizontal or vertical as in classic game)

- The objective of the game is to segregate balls of different colors in different rooms.
- And the balls are not moving.

5. Naive Bayes

It is a classification technique based on Bayes' theorem with an assumption of independence between predictors.

In simple terms, a Naive Bayes classifier assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

For example:- a fruit may be considered to be an apple if it is red, round, and about 3 inches in diameter. Even if these features depend on each other or upon the existence of the other features, a naive Bayes classifier would consider all of these properties to independently contribute to the probability that this fruit is an apple.

Naive Bayesian model is easy to build and particularly useful for very large data sets. Along with simplicity, Naive Bayes is known to outperform even highly sophisticated classification methods.

Bayes theorem provides a way of calculating posterior probability P(c|x) from P(c), P(x) and P(x|c). Look at the equation below:

Likelihood   Class Prior Probability

$$P(c|x) = \frac{P(x|c)P(c)}{P(x)}$$

Posterior Probability   Predictor Prior Probability

$$P(c|X) = P(x_1|c) \times P(x_2|c) \times \cdots \times P(x_n|c) \times P(c)$$

Here,

- *P*(*c*|*x*) is the posterior probability of *class* (*target*) given *predictor* (*attribute*).
- *P*(*c*) is the prior probability of *class*.
- *P*(*x*|*c*) is the likelihood which is the probability of *predictor* given *class*.

- *P*(*x*) is the prior probability of *predictor*.

Example: Let's understand it using an example. Below I have a training data set of weather and corresponding target variable 'Play'. Now, we need to classify whether players will play or not based on weather conditions. Let's follow the below steps to perform it.

Step 1: Convert the data set to frequency table

Step 2: Create Likelihood table by finding the probabilities like Overcast probability = 0.29 and probability of playing is 0.64.

Weather	Play
Sunny	No
Overcast	Yes
Rainy	Yes
Sunny	Yes
Sunny	Yes
Overcast	Yes
Rainy	No
Rainy	No
Sunny	Yes
Rainy	Yes
Sunny	No
Overcast	Yes
Overcast	Yes
Rainy	No

Frequency Table		
Weather	No	Yes
Overcast		4
Rainy	3	2
Sunny	2	3
Grand Total	5	9

Likelihood table				
Weather	No	Yes		
Overcast		4	=4/14	0.29
Rainy	3	2	=5/14	0.36
Sunny	2	3	=5/14	0.36
All	5	9		
	=5/14	=9/14		
	0.36	0.64		

Step 3: Now, use Naive Bayesian equation to calculate the posterior probability for each class. The class with the highest posterior probability is the outcome of prediction.

Problem: Players will pay if weather is sunny, is this statement is correct?

We can solve it using above discussed method

so, P(Yes | Sunny) = P(Sunny | Yes) * P(Yes) / P (Sunny)

Here we have P (Sunny |Yes) = 3/9 = 0.33, P(Sunny) = 5/14 = 0.36, P(Yes)= 9/14 = 0.64

Now, P (Yes | Sunny) = 0.33 * 0.64 / 0.36 = 0.60, which has higher probability.

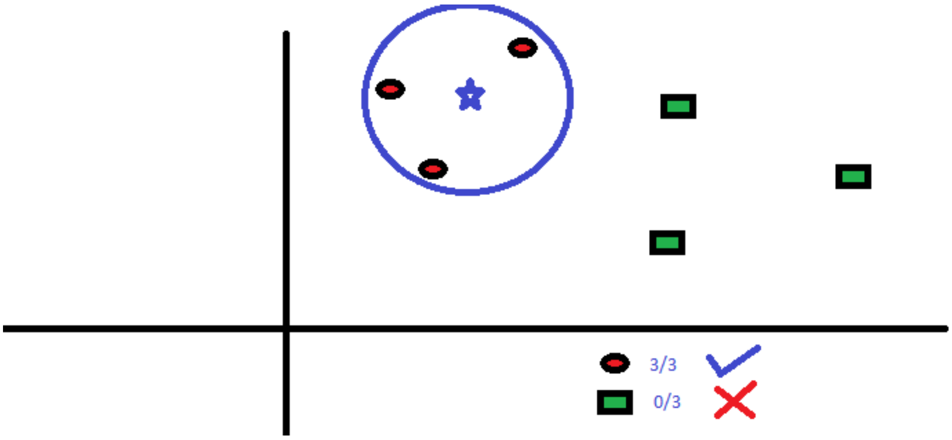
Naive Bayes uses a similar method to predict the probability of different class based on various attributes. This algorithm is mostly used in text classification and with problems having multiple classes.

6. KNN (K- NEAREST NEIGHBORS)

It can be used for both classification and regression problems. However, it is more widely used in classification problems in the industry. K nearest neighbors is a simple algorithm that stores all available cases and classifies new cases by a majority vote of its k neighbors. The case being assigned to the class is most common amongst its K nearest neighbors measured by a distance function.

These distance functions can be Euclidean, Manhattan, Minkowski and Hamming distance. First three functions are used for continuous function and fourth one (Hamming) for categorical variables. If K = 1, then the case is simply assigned to the class of its nearest neighbor. At times, choosing K turns out to be a challenge while performing kNN modeling.

More: Introduction to k-nearest neighbors : Simplified.



KNN can easily be mapped to our real lives. If you want to learn about a person, of whom you have no information, you might like to find out about his close friends and the circles he moves in and gain access to his/her information!

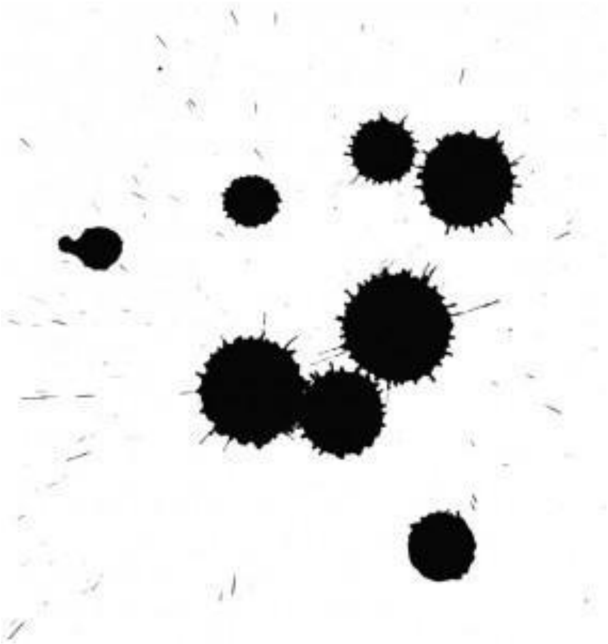
Things to consider before selecting kNN:

- KNN is computationally expensive
- Variables should be normalized else higher range variables can bias it
- Works on pre-processing stage more before going for kNN like outlier, noise removal

7. K-Means Clustering

It is a type of unsupervised algorithm which solves the clustering problem. Its procedure follows a simple and easy way to classify a given data set through a certain number of clusters (assume k clusters). Data points inside a cluster are homogeneous and heterogeneous to peer groups.

Remember figuring out shapes from ink blots? k means is somewhat similar this activity. You look at the shape and spread to decipher how many different clusters / population are present!



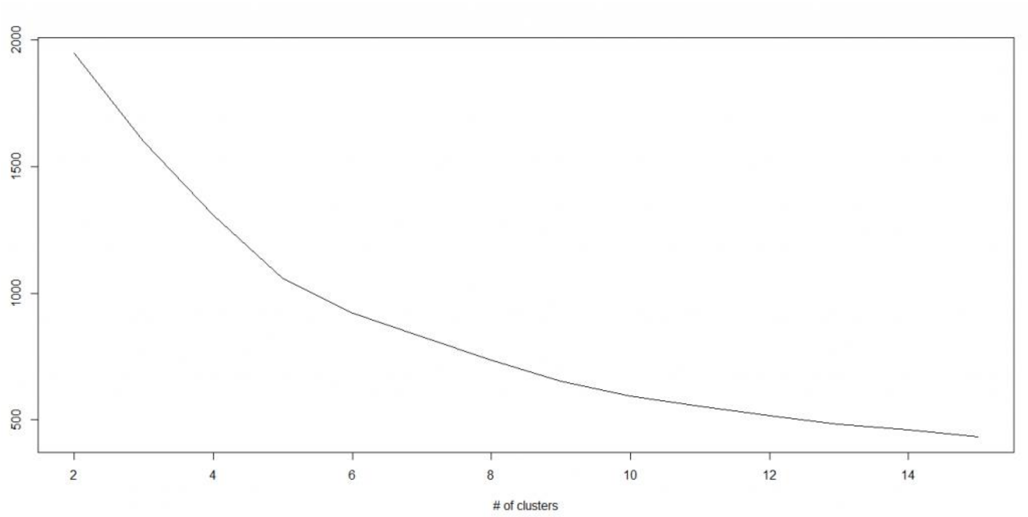
How K-means forms cluster:

1. K-means picks k number of points for each cluster known as centroids.
2. Each data point forms a cluster with the closest centroids i.e. k clusters.
3. Finds the centroid of each cluster based on existing cluster members. Here we have new centroids.
4. As we have new centroids, repeat step 2 and 3. Find the closest distance for each data point from new centroids and get associated with new k-clusters. Repeat this process until convergence occurs i.e. centroids does not change.

How to determine value of K:

In K-means, we have clusters and each cluster has its own centroid. Sum of square of differences between the centroid and the data points within a cluster constitutes within sum of square value for that cluster. Also, when the sum of square values for all the clusters are added, it becomes total within sum of square value for the cluster solution.

We know that as the number of cluster increases, this value keeps on decreasing but if you plot the result you may see that the sum of squared distance decreases sharply up to some value of k, and then much more slowly after that. Here, we can find the optimum number of cluster.



8. RANDOM FOREST

Random Forest is a trademark term for an ensemble of decision trees. In Random Forest, we've collection of decision trees (also known as "Forest"). To classify a new object based on attributes, each tree gives a classification and we say the tree "votes" for that class. The forest chooses the classification having the most votes (over all the trees in the forest).

Each tree is planted & grown as follows:

1. If the number of cases in the training set is N, then sample of N cases is taken at random but WITH REPLACEMENT. This sample will be the training set for growing the tree.

2. If there are M input variables, a number m<<M is specified such that at each node, m variables are selected at random out of the M and the best split on these m is used to split the node. The value of m is held constant during the forest growing.

3. Each tree is grown to the largest extent possible. There is no pruning.

For more details on this algorithm, comparing with decision tree and tuning model parameters, I would suggest you to read these articles:

1. Introduction to Random forest – Simplified
2. Comparing a CART model to Random Forest (Part 1)
3. Comparing a Random Forest to a CART model (Part 2)
4. Tuning the parameters of your Random Forest model

9. Gradient Boosting Algorithms

9.1. GBM

GBM is a boosting algorithm used when we deal with plenty of data to make a prediction with high prediction power. Boosting is actually an ensemble of learning algorithms which combines the prediction of several base estimators in order to improve robustness over a single estimator. It combines multiple weak or average predictors to build strong predictor. These boosting algorithms always work well in data science competitions like Kaggle, AV Hackathon, CrowdAnalytix.

More: Know about Boosting algorithms in detail

GradientBoostingClassifier and Random Forest are two different boosting tree classifier and often people ask about the difference between these two algorithms.

9.2. XGBoost

Another classic gradient boosting algorithm that's known to be the decisive choice between winning and losing in some Kaggle competitions.

The XGBoost has an immensely high predictive power which makes it the best choice for accuracy in events as it possesses both linear model and the tree learning algorithm, making the algorithm almost 10x faster than existing gradient booster techniques.

The support includes various objective functions, including regression, classification and ranking.

One of the most interesting things about the XGBoost is that it is also called a regularized boosting technique. This helps to reduce overfit modelling and has a massive support for a range of languages such as Scala, Java, R, Python, Julia and C++.

Supports distributed and widespread training on many machines that encompass GCE, AWS, Azure and Yarn clusters. XGBoost can also be integrated with Spark, Flink and other cloud dataflow systems with a built in cross validation at each iteration of the boosting process.

To learn more about XGBoost and parameter tuning, visit:

https://www.analyticsvidhya.com/blog/2016/03/complete-guide-parameter-tuning-xgboost-with-codes-python/.

9.3. LightGBM

LightGBM is a gradient boosting framework that uses tree based learning algorithms. It is designed to be distributed and efficient with the following advantages:

- Faster training speed and higher efficiency
- Lower memory usage
- Better accuracy
- Parallel and GPU learning supported
- Capable of handling large-scale data

The framework is a fast and high-performance gradient boosting one based on decision tree algorithms, used for ranking, classification and many other machine learning tasks. It was developed under the Distributed Machine Learning Toolkit Project of Microsoft.

Since the LightGBM is based on decision tree algorithms, it splits the tree leaf wise with the best fit whereas other boosting algorithms split the tree depth wise or level wise rather than leaf-wise. So, when growing on the same leaf in Light GBM, the leaf-wise algorithm can reduce more loss than the level-wise algorithm and hence results in much better accuracy which can rarely be achieved by any of the existing boosting algorithms.

Also, it is surprisingly very fast, hence the word ‘Light’.

Refer to the article to know more about LightGBM:

https://www.analyticsvidhya.com/blog/2017/06/which-algorithm-takes-the-crown-light-gbm-vs-xgboost/

9.4. CatBoost

CatBoost is an open-source machine learning algorithm from Yandex. It can easily integrate with deep learning frameworks like Google's TensorFlow and Apple's Core ML. The best part about CatBoost is that it does not require extensive data training like other ML models, and can work on a variety of data formats; not undermining how robust it can be. Make sure you handle missing data well before you proceed with the implementation.

Catboost can automatically deal with categorical variables without showing the type conversion error, which helps you to focus on tuning your model better rather than sorting out trivial errors.

Learn more about Catboost from this article:

https://www.analyticsvidhya.com/blog/2017/08/catboost-automated-categorical-data/

What tools are used by Machine Learning professionals?

There are several tools and languages being used in machine learning. The exact choice of the tool depends on your need and scale of operations. But, here are the most commonly used tools in machine learning:

- **Programming Languages:**
  - R
  - Python
  - SAS
  - Julia
  - Java
  - Swift
  - Scala
  - Go
- **Databases:**
  - SQL
  - Oracle
  - Hadoop
  - NoSQL databases like MongoDB
- **Visualisation tool:**
  - D3.js
  - Tableau
  - QlikView / QlikSense
- **Other commonly used tools:**
  - Excel
  - PowerPoint
  - Git
  - Bash Commands

Connection for Next Section

How can I build a career in Machine Learning?

Now you are asking! Given the shortage of talent in this domain, it definitely makes sense to look at building a career in data science and machine learning. But before you decide, you should keep the following things in mind:

- You would need to be comfortable with coding in order to build a career as a data scientist. Sure, there are GUI tools available – but data scientists need to code their own algorithms to be up to speed with the latest developments in the domain.
- You do not need a background or Ph.D. in mathematics. You can always pick up the things you need. If you are from this background – it helps, but it is not mandatory.
- For those of you transitioning from any other domain or field – plan for at least 18 months of transition. If you get a break before – consider this as a bonus.

If you are ready to build a career in data science after reading the tips above – we have a plan for you. You can check out the FREE learning path to becoming a data scientist by Analytics Vidhya here. If you need guidance and mentorship – check out our AI & ML Blackbelt+ program.

In addition, you can do the following:

- Taking MOOCs or University Courses
  - Coursera
  - Analytics Vidhya Courses
- Attending Meetups, Webinars, Conferences
- Solving problems yourself and learning along the way
- Becoming part of data science communities and learning from the experts

What are the skills needed to build a career in Machine Learning?

- **Structured thinking, communication and problem solving**

  This is probably the most important skill required in a data scientist. You need to take business problems and then convert them to machine learning problems. This requires putting a framework around the problem and then solving it.

- **Mathematics and Statistics**

  You need mathematics and statistics to understand how the algorithms work and what are their limitations.

- **Business Understanding**

  At the end of the day, you will be solving business problems using machine learning. So, you would need to have a good understanding of the current processes, limitations and the options.

- **Software Skills and Programming**

  Data Scientists not only need to build algorithms, they need to code them and integrate them into the products seamlessly. That is where software skills come into play.

How can I prepare for Data Science and Machine Learning Interviews?

Check out our awesome course "Ace Data Science Interviews" for a detailed and Structured preparation modules. Check out the seven step process and a comprehensive guide to help you out.

WHO ARE THE TOP RESEARCHERS IN MACHINE LEARNING?

As mentioned multiple times – Machine Learning is a very active field of research. You can read this article to get a list of the top researchers in the field of machine learning.

Machine Learning and Deep Learning are fast evolving areas. We expect this to continue in the coming years. If you want to learn about the latest developments in the field, gain perspective from the best brains and be future proof – there cannot be a better way but to follow these experts!

The ordering is in no particular order!

GEOFFREY HINTON

- **He is best known for his work on artificial neural networks (ANNs).** His contributions in the field of deep learning are a key reason behind the success of the field and he is often called the “Godfather of Deep Learning” (with good reason).
- His research on the backpropagation algorithm brought about a drastic change in the performance of deep learning models. Mr. Hinton’s other notable research works are Boltzmann machines and Capsule neural networks. Both major breakthroughs in our field.
- Hinton recently won the 2018 Turing Award for his groundbreaking work around deep neural networks, along with Yann LeCun and Yoshua Bengio. He has also won the BBVA Foundation Frontiers of Knowledge Award (2016) and IEEE/RSE Wolfson James Clerk Maxwell Award.

MICHAEL I JORDAN

- Michael Jordan is a professor at the University of California, Berkeley.
- He has been a major advocate of Bayesian networks and has made a significant contribution towards probabilistic graphical models, spectral methods, natural language processing (NLP), and much more.
- He has won many well-known awards, including the IEEE Neural Networks Pioneer Award, the best paper award (with R. Jacobs) at the American Control Conference (ACC 1991) and the ACM – AAAI Allen Newell Award. He has also been named a Neyman Lecturer and a Medallion Lecturer by the Institute of Mathematical Statistics.

ANDREW NG

- Andrew Ng is considered as one of the most significant researchers in Machine Learning and Deep Learning in today's time.
- **He is the co-founder of Coursera and deeplearning.ai and an Adjunct Professor of Computer Science at Stanford University. Professor Andrew also co-founded the Google Brain project and was previously the Chief Scientist at Baidu.**
- He has over 300 published papers in machine learning and robotics! He is also a recipient of prestigious awards like IJCAI Computers and Thought award, ICML Best Paper Award, ACL Best Paper Award and many, many more.

YANN LECUN

- Yann LeCun is another iconic name in machine learning. He is a professor, researcher, and R&D manager with academic and industry experience in machine learning, deep learning, computer vision, and robotics.
- **Mr. LeCun is currently the Chief AI Scientist and VP at Facebook.**
- **Yann LeCun is the founding father of convolutional nets.** He made convolutional neural networks work with backpropagation, which is widely used in computer vision applications.
- He has over 150 papers published under his name and has received a number of awards for his contributions. He has won the 2014 IEEE Neural Network Pioneer Award and the 2015 PAMI Distinguished Researcher Award. LeCun also won the 2018 Turing award, along with Geoffrey Hinton and Yoshua Bengio.

YOSHUA BENGIO

- Yoshua Bengio is a professor at the Department of Computer Science and Operations Research at the Université de Montréal. **He is also the co-founder of Element AI**, a Montreal-based business incubator that seeks to transform AI research into real-world business applications.
- **Yoshua is well known for his work on artificial neural networks and deep learning** in the 1980s and 1990s. He co-created the prestigious ICLR conference with Yann LeCun.
- He is one of the most-cited computer scientists in the areas of deep learning, recurrent networks, probabilistic learning, and natural language.
- He has received the prestigious award of Canada Research Chair in Statistical Learning Algorithms and also won the 2018 Turing award

Jürgen Schmidhuber

- **He is a computer scientist, known for his work around artificial neural networks and deep learning.**
- He, along with some of his students, published sophisticated versions of long short-term memory (LSTM), an improved version of recurrent neural networks.
- His research work also included the speeding up of convolutional neural networks using GPUs.
- Mr. Schmidhuber is the recipient of numerous awards, author of over 350 peer-reviewed papers, and Chief Scientist of the company NNAISENSE, which aims at building the first practical general-purpose AI.

Zoubin Ghahramani

- Zoubin is a professor of Information Engineering at the University of Cambridge.
- **His research interests include Bayesian approaches to machine learning, statistics, information retrieval, bioinformatics, and artificial intelligence.**
- He completed his Ph.D. from the Department of Brain and Cognitive Sciences at the Massachusetts Institute of Technology, under Michael I. Jordan and Tomaso Poggio.
- In 2014, he co-founded a startup, Geometric Intelligence that focuses on object or scenario recognition.
- Uber acquired Geometric Intelligence and Zoubin joined Uber's A.I. Labs in 2016. He has published over 250 research papers and was elected Fellow of the Royal Society (FRS) in 2015.

Sebastian Thrun

- Sebastian Thrun is currently the CEO of Kitty Hawk Corporation and the co-founder of Udacity.
- Sebastian founded the Google X Lab and Google's self-driving team. He led the project from the start and is widely considered a leader when it comes to autonomous vehicles.
- As you might expect from a person of Sebastian's stature, he is deeply integrated into the academic side of machine learning as well. He is Adjunct Professor at Stanford University and at Georgia Tech.
- Sebastian was named one of 'Brilliant 5' by Popular Science magazine in 2005. He has also been awarded the Max-Planck-Research Award (2011).

Yaser S. Abu-Mostafa

- Professor Yaser is a Professor of Electrical Engineering and Computer Science at the California Institute of Technology.
- **He co-founded the most renowned machine learning conference for researchers – NIPS, or the Conference on Neural Information Processing Systems.**
- He was awarded the Richard Feynman award prize for excellence in teaching (which is no surprise to anyone who has seen his talk about machine learning) and has numerous technical publications.

Peter Norvig

- Peter Norvig is among the godfathers of modern-day AI.
- He is currently the Director of Research at Google. Before his current role, **Peter was head of Google's core search algorithms group, and of NASA Ames's Computational Sciences Division.** He won the NASA Exceptional Achievement Award in 2001.
- Peter is also a bestselling author and has written numerous books on the field of artificial intelligence. We loved his article titled 'Teach Yourself Programming in Ten Years" where he put forth an impassioned argument against introductory books that promised to teach you programming in one go.

Ian Goodfellow

- Ian Goodfellow is best known for inventing Generative Adversarial Networks (GANs). GANs have become ubiquitous in deep learning and are popularly used at companies like Facebook and Google.
- Ian is currently a Director of Machine Learning at Apple. He is a researcher at heart and has previously worked as a research scientist at Google Brain and OpenAI.
- Ian's list of mentors is enviable. He completed his MS in computer science under Andrew Ng and his Ph.D. under Yoshua Bengio and Aaron Courville.

Andrej Karpathy

- Andrej Karpathy is already a legend in the AI community. He is currently working as the Director of AI at Tesla but has long been involved in the machine learning domain.
- His interest and specialization lies in deep learning, computer vision and image recognition.
- He completed his Ph.D. from Stanford University under the supervision of the great Dr. Fei-Fei Li. He has previously worked at OpenAI as a research scientist as well.

Which books should I read about Machine Learning?

As a beginner with no background in machine learning, you can read the following books (the links are affiliate links):

- SuperIntelligence
- The Master Algorithm
- Life 3.0 – Being Human in the Age of Artificial Intelligence
- AI Superpowers
- Moneyball
- Scoring Points
- The Singularity is Near

On the other hand, if you already have the required background and want to learn machine learning – these are the books you should read:

- Machine Learning Yearning
- The Hundred-Page Machine Learning Book
- Programming Collective Intelligence
- Machine Learning for Hackers
- Machine Learning by Tom M Mitchell
- The Elements of Statistical Learning
- Learning from Data
- Pattern Recognition and Machine Learning
- Natural Language Processing with Python

Books on Artificial Intelligence

- Artificial Intelligence: A Modern Approach
- Artificial Intelligence for Humans
- Paradigm of Artificial Intelligence Programming
- Artificial Intelligence: A New Synthesis
- Superintelligence
- The Singularity is Near
- Life 3.0 – Being Human in the Age of Artificial Intelligence
- The Master Algorithm

A Few Products from the Future

I am a data science evangelist. I see opportunities to make our world data driven all around me. I love talking to people about the possibilities with Data Science and honestly, I couldn't have asked for a better decade to live in!

In the last decade, data science and machine learning have gone mainstream. We have seen smartphones becoming omnipresent, the apps in those smartphones ruling our information consumption, virtual assistants coming up and connecting our homes.

But, this is just the beginning. I believe that the next 5 years would see even bigger and faster changes all around us. In the next few years, more and more developments in machine learning will move from research labs to our home. We will see AI enabled chips all around us. Everything in our houses and offices, sooner or later will become smart.

The clothes we wear, the software we use, the gadgets we have, the car we drive (or we don't) - all of these are becoming smart as I write this article. Sounds exciting? Scary? Nerve-wrecking? Well - irrespective of how you feel - this is bound to come! Remember Thanos saying "Dread it! Run from it! Destiny arrives all the same." Data Science feels like a similar force coming into our lives.

In this section, I will layout a vision for a few products which are technically already feasible.

The Refrigerator or the Food Selector

Today, when you think about a Refrigerator, you think of a place to store food. What if we make the same Refrigerator smart?

Let us add the following sensors to the Refrigerator:

- Weighing scale
- Image recognition setup – camera and the chip running the algorithm

Let us also change the way we interact with a refrigerator. The "New" refrigerator is an enhanced vending machine. You can see what is stored, you have places to add input and get output.

Why would we do this? For multiple reasons – when we get an input, this new setup allows us to capture images from multiple angles create an accurate image recognition system. By capturing images from multiple angles – we can also estimate the health of a fruit. We could It also lets the Refrigerator create a database on what was added at what time.

Is your notebook smarter than you?

What if your notebook is constantly connected to the Internet and pops up references to you as you are making notes? How will your life change?

All you need is adding a connection between your notebook and a smartphone. There are already some products in the market today trying to do this, but it is still early days. Check out Wacom Bamboo Folio which can convert everything you write on the notepad to Digital format and convert handwritten notes to text. Once done – you just need to find the right references related to the notes on Internet / Intranet.

Imagine the wonders it can do to your research. Imagine going to a conference and writing notes and getting references along with it.

Wardrobe or Personal Styling Specialist?

Take your wardrobes - they have been like that for years now. We use them to store our dresses efficiently. The only thing which has changed about wardrobes in last decade would be their looks, may be materials. But, it has broadly remained the same.

Now, think of a wardrobe which is fitted with a Smart Camera and an Assistant. This assistant can see (through the camera) what you wear, store what you wore from your wardrobe, when you wore it, how you wore it and what would be the style quotient of your dress up.

Once the assistant has seen you for some time, it can provide you personalized recommendations to improve your style quotient. Which shoes will go best on your dress? Which belt? Which bag do you pick? All the answers can be arrived by studying available combinations of clothes, ranking them and then suggesting them to you.

This is a big step forward for people like me, who end up consulting their better halves for everything we wear! But, the beauty does not stop there, it actually just starts from there. This assistant can warn you when you repeat the same dresses often or when you are not dressing casually enough on a Friday.

How about syncing this assistant with your Calendar? The assistant can see the meetings you have and plan the right style and dresses for you. Have long travel - the assistant will pick the dress with the right fabric based on the weather conditions.

Not sure what to buy when you are refreshing your wardrobe. The assistant can keep track of what is trending on the internet and which fashion lines will match your style. It can recommend you the styles and the collections you are likely to buy without you going through the hassle of shopping.

You see how data science and machine learning changes the entire experience by adding a few sensors, capturing the data and running computations on this data? Now imagine this being applied to any instrument you have in your house - Is it a refrigerator or a Food Recommender System? A Sofa set or a Posture monitoring set?

It is for you to decide! We will know the answers in the decade to come.