# The Ultimate Business Owners' Guide to Machine Learning

# Contents

"Nowadays barely a single geek talk, tech conference agenda or software product release can do without mentioning Machine Learning. Like "data is the new oil", Machine Learning is the chemical industry that utilizes the raw material to change the way we live. But the excitement surrounding the topic is evolving disproportionately faster than the level of understanding, which we see as a problem potentially harmful for all levels: business, consumers, and scientists.

In this guide, we made an attempt to familiarize decision-makers, regardless of their background and industry, with theoretical and practical basics of Machine Learning. We hope that the knowledge one can acquire by reading this guide will accelerate technological change at companies and organizations and, at the same time, create an unbiased picture of the technology itself."

Yuri Svirid, PhD

CEO, Silk Data

# Foreword

Machine Learning (ML) is a buzzword nowadays that is about to reshape every sphere of our life soon. Well, it already does, and the pace of adoption is only going to accelerate as cost of production and marketing rises and companies find it more challenging than ever to stay on top of competition.

Thus, Netflix save about $1 billion annually by using machine learning to make personalized recommendations while GE claims implementing machine learning software saved them $80 million. Whether in medicine, insurance, transportation or agriculture, every progressive business now realizes the importance of automation and smartification of their processes.

However, there's so much hype around the topic these days, that it is becoming complicated for business owners, investors, and industry professionals to get a clear undistorted picture of the technology and its capabilities.

Some people approach machine learning as a magic wand that can solve their problems with little or no effort, other believe its practical

value is overestimated. With this guide, we try to create a reliable and unbiased image of machine learning and provide people of different backgrounds and from various industries with comprehensible knowledge on the subject.

## You will find this guide useful if you're:

• A business owner or CEO It is essential that business leaders and entrepreneurs not only keep up with their industry trends but can leverage advanced methods and technology to move beyond the traditional practices and stay on top of the competition. Machine Learning is indeed the great source of getting a competitive advantage, which makes the knowledge comprised in this eBook indispensable for a business owner.

• A product leader Machine Learning is extremely powerful when it comes to providing enhanced functionality and personalized user experience for digital products. If you are a product manager, you might have already considered exploiting machine learning capabilities – and this guide will help you explore a more practical side of implementing the technology.

• **A change leader** There are no managers who wouldn't want their work to be done better, faster, and more efficiently. Machine Learning can automate routine tasks at scale and at a much higher speed than legacy software. If you'd like to look a little deeper behind the articles on Digital Trends, this guide will give you a perfect introduction into the basics of machine learning.

**Before you get bored…**

We supplemented the book with lots of visuals and vivid examples, highlighting both glorious triumphs and funny mistakes of AI. We also put together a list of helpful resources that you refer to for a more in-depth study of the subject.

# What is Machine Learning?

Machine Learning is a quite multidimensional concept that uses approaches from statistics, computer science, and data processing.

In simple terms, **Machine Learning**-based product is a computer program (a combination of algorithms and data) designed to get correct answers, generate accurate predictions and make its own sensible decisions without being explicitly coded. Unlike classical programs, it mainly relies on **data**, not algorithms, to progressively improve itself upon a given task, learning from experience. Let's take an example of a predictive text function. Instead of giving the program explicit instructions (which could comprise millions of alternatives for a single step) you can show the examples of thousands of previously sent messages and the machine will use this data to build reasonable predictions of the words and phrases you're going to use next.

In far 1952, IBM's Arthur Samuel created one of the first machine learning programs that played checkers at the human level and got better with every new game (according to Wikipedia, it had been

running on IBM's first commercial computer!)[1]. 54 years later, Google's Alpha Go beat a professional human player at Go, arguably the hardest board game ever created. One can be either excited or scared of the speed at which the progress is being made here. But despite all concerns around whether machines will be able to outsmart humans anytime soon, the primary goal of machine learning is to make people's life better and their work easier.

Over the last 70 years, Machine Learning experienced several waves of hype, with different buzzwords being used to promote new achievements. You might have heard things like Data Mining, Deep Learning, Synthetic Intelligence, Cognitive Computing, Affective Computing, DeepTech, Data Science and so forth. Despite some technical difference, all above-mentioned terms can be explained in a single way: an approach to making the computer solve different tasks where exact rules cannot be easily set. In the rest of this book, we will normally refer to Machine Learning (ML) and Artificial Intelligence (AI), without paying much attention to difference between the terms.

---

[1] See https://en.wikipedia.org/wiki/Arthur_Samuel

In simple words, the typical application case for ML can be formulated as follows:

**Machine Learning Rule of Thumb**

> "If a typical person can do a mental task with less than one second of thought, we can probably automate it using AI either now or in the near future."
>
> Andrew Ng, Stanford University

## The Typical Tasks Solved by Machine Learning

The intelligent systems built on machine learning algorithms have the capability to learn from experience or historical data which make them a powerful tool for solving a wide range of tasks. Currently, ML has been used in multiple areas, from social media (photo tagging) to genetics (identifying disease genes in a human DNA).

Here are some common applications. They demonstrate that the opportunities of practical use of ML are endless and not limited to any specific area.

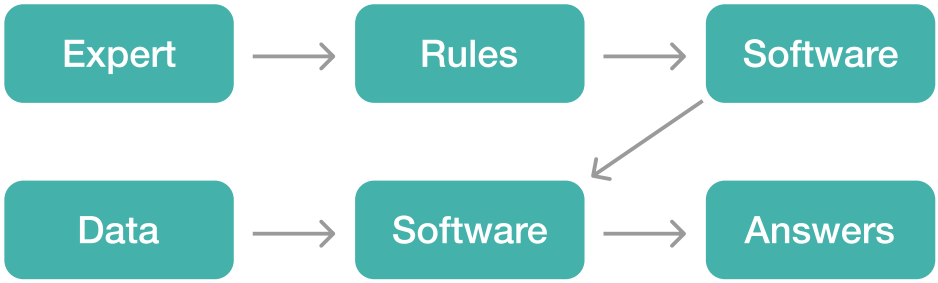
| Input | Response | Application |
|---|---|---|
| Picture | Are there cats? | Photo tagging |
| Loan application | Will they repay the loan? | Loan approvals |
| Ad plus user information | Will user click on ad? | Targeted online ads |
| Audio clip | Transcript of audio clip | Speech recognition |
| English sentence | Chinese language | Language translation |
| Digital text | Text summary | Text summarization |
| Users' transaction history | Customers grouped into segments | Customer segmentation |
| A user's saved tracks | Other tracks he is likely to appreciate | Recommendation systems |
| Utility bill | Extracted personal information | Named entity extraction |
| Text-based document | Similar documents in the collection | Similarity search |

Introducing AI on the company level can help cut down on costs and increase performance manifold in the long run, first and foremost through automating repetitive tasks and reducing human errors. However, companies must be careful in selecting areas they would like to implement ML in as well as in choosing the methods and building the strategy. In many cases it will not worth investments and a level of inherent risk. While technology giants like Google, Amazon, Facebook, Microsoft, Apple and other AI frontrunners are increasing their efforts in integrating Machine Learning, a wide adoption of ML and AI is still at its very roots.

# ML Principles

## Classical Software vs Machine Learning

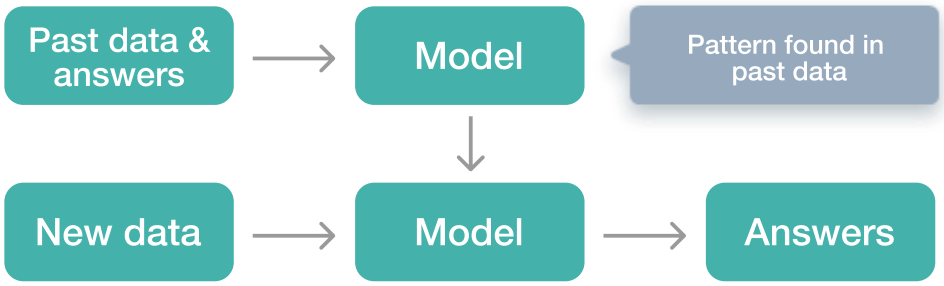
In classical software engineering, there is always an expert – normally a person – that sets precise rules when developing software. The software uses data to generate answers a user wants to get.




In order to alter the results (as environment is changed and new patterns are discovered in the past data) you would need to re-write the rules. More importantly, in many cases formulating the rules can be prohibitively complex, the most common example is how to write down all rules necessary to tell dog from a cat.

Machine Learning algorithms work in a different way. They can extract the rules without being explicitly programmed by human. All

patterns discovered in the past would draw new conclusions to form new data sets. This makes possible to process a really huge amount of information without regular intervention by experts. Instead, experts can only analyze a small number of samples and make ad-hoc adjustments.




**Disclaimer: Now it's going to be a bit of Math. It's OK to skip this part if you're only interested in practical side of ML**

To understand the basic concepts of machine learning, let's refer to related mathematical tool named curve fitting.

## Curve Fitting Example

Curve fitting is the process of engineering a curve that fits best to a series of data, possibly subject to restraints. Machine Learning task is in principle just the same – finding the best approximation from a number of noisy samples, to make predictions not covered by data.

In its simplest variant, the curve fitting is just linear regression: how to tell from the noisy data, if some value (for example business revenue) is actually growing or not, as in illustration below.

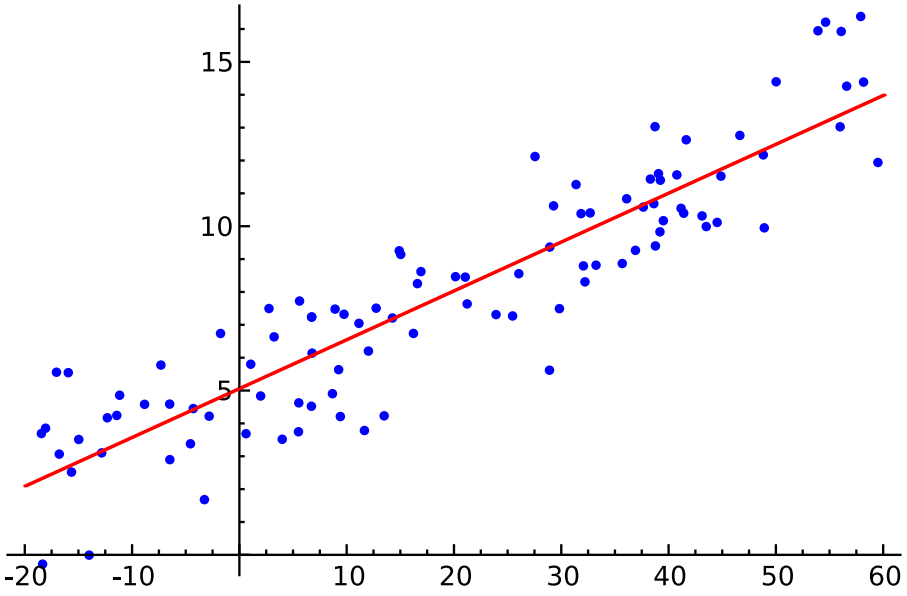



Image source: Wikipedia

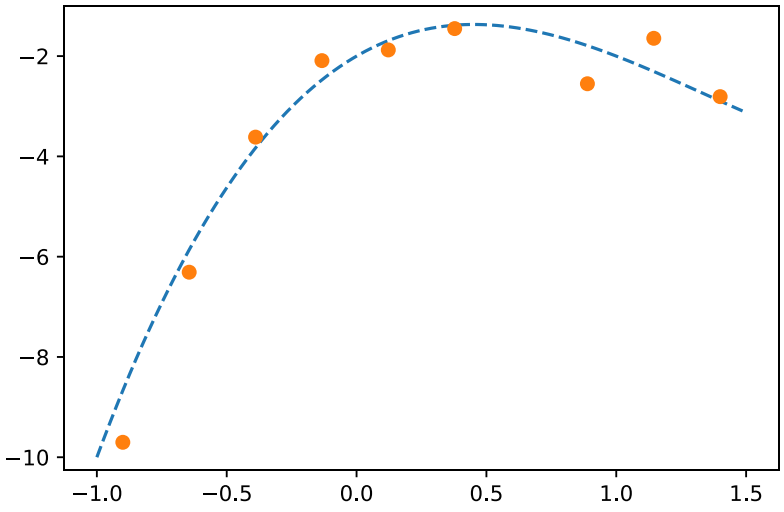
Normally, everyone can draw or imagine a linear approximation of data by just looking on the chart. And the true value of machine

Learning stems from its ability to automatically handle much more advanced cases.

To move forward and to explain few basic concepts of Machine learning, let us take a simple curve:

$$f(x) = x^3 - 4x^2 + 3x - 2 + N(0, 0.25).$$

Here, $x$ is a free variable and $N$ is a noise component. In the plot below, the dashed line is an 'ideal' curve, without noise, and the dots mimic the observed data, affected by noise.

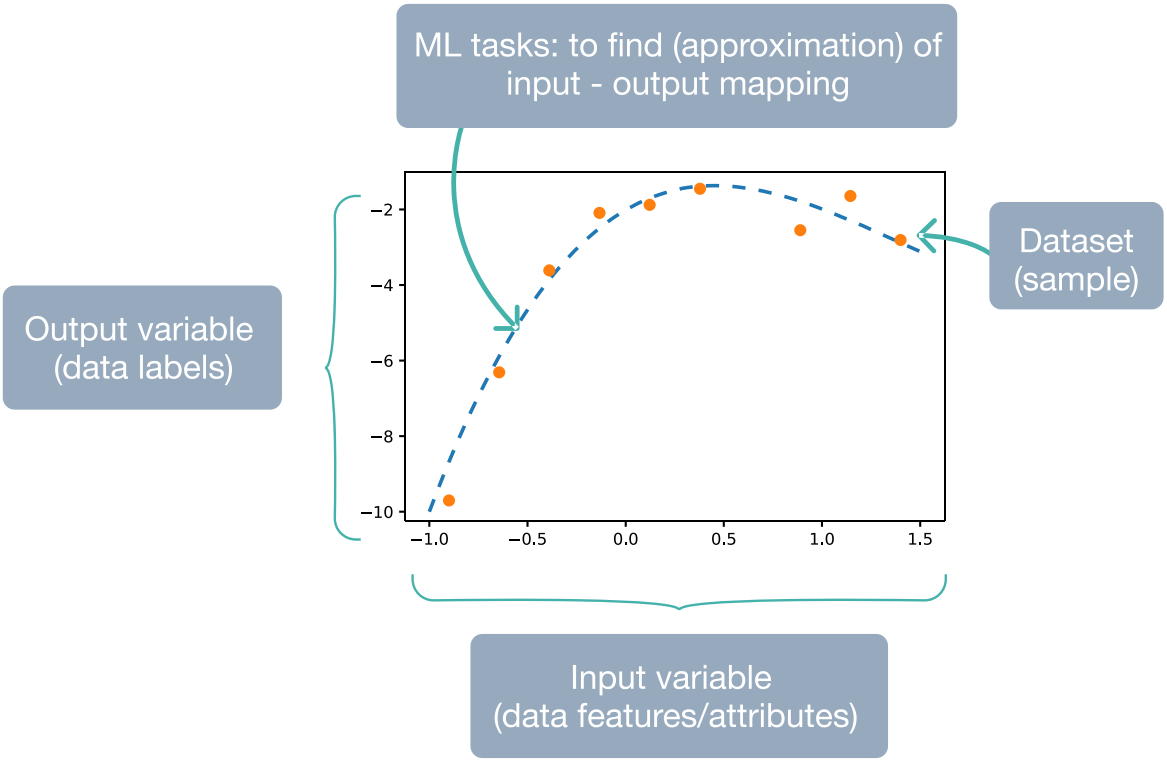



The problem is to find the best polynomial approximation from a few (noisy) samples with no data on the polynomial parameters or order available. Usually, we have an assumption of how the model should

look like, and it is necessary to find the model parameters that best fit the data according to some *metric* (more on metrics below).

Note, that *noise* is unavoidable in almost all practical application. From the mathematical standpoint, the varying light conditions in image and video processing, misprints or minor alterations of text in a document, or changing numbers in the business data – all these are examples of noise. To be useful, a ML system must be sufficiently stable against the noise.

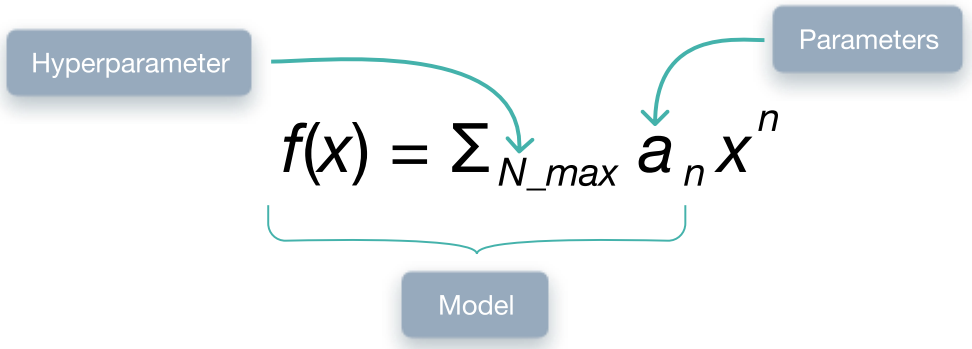
The chart below illustrates how a math textbook example translates to Machine Learning terms:



- An independent variable on the x-axis is called a *data feature* or *attribute*. In most practical cases, there will be not a single feature, but a large set of features represented as array of numbers (or vector).
- The dependent variable on a vertical axis is the so-called *data label.* Machine Learning task is to find the best (most exact/accurate) approximation of (input) – output mapping.
- An individual data point is usually called a *sample*.
- The collection of the samples (data points) is called the *dataset*. Mostly, the Machine Learning task will be centered around the dataset, the data properties and its statistics.

## Model and (Hyper-) Parameters

If we search for the answer (solution) using a polynomial, by optimization of the coefficients of the polynomial, $a_n$, or *model fitting*, we are trying to produce (find) a more exact value.




The order of a polynomial (the number of coefficients or parameters), *N_max*, is the so-called *hyperparameter*. Hyperparameter value distinguishes a specific model from all similar possible models.

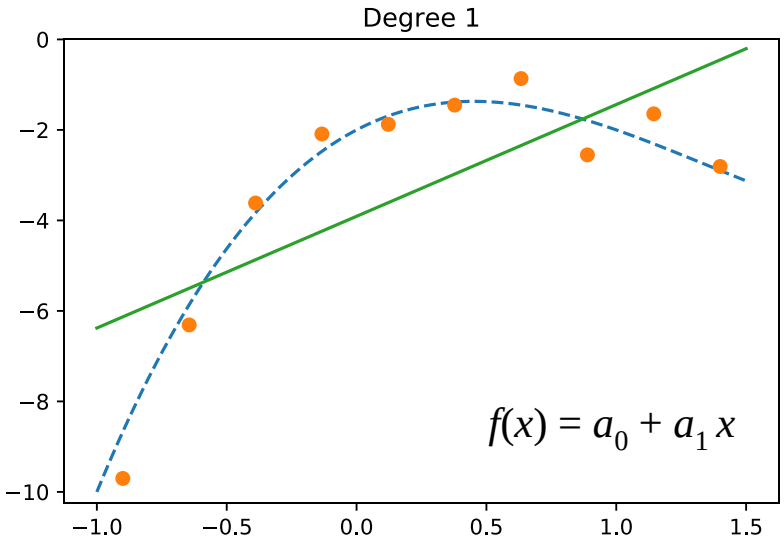
In addition, the specific form of the model (be it polynomial, neural network, decision tree or something else) is commonly named as *model type*.

Wrapping up, ML consists of two stages, as will be further discussed below:

- matching model parameters to the data;
- analyzing which parameters and hyperparameters demonstrated more accurate results.

## Under- and Overfitting

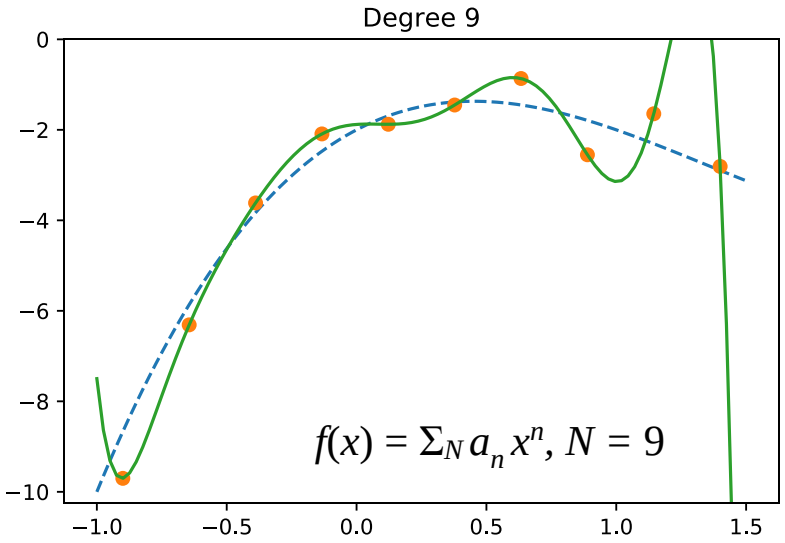
As seen on the pictures below, both a low-order polynomial and a high-order polynomial demonstrate poor performance in ML. These are the so-called cases of under- and overfitting.



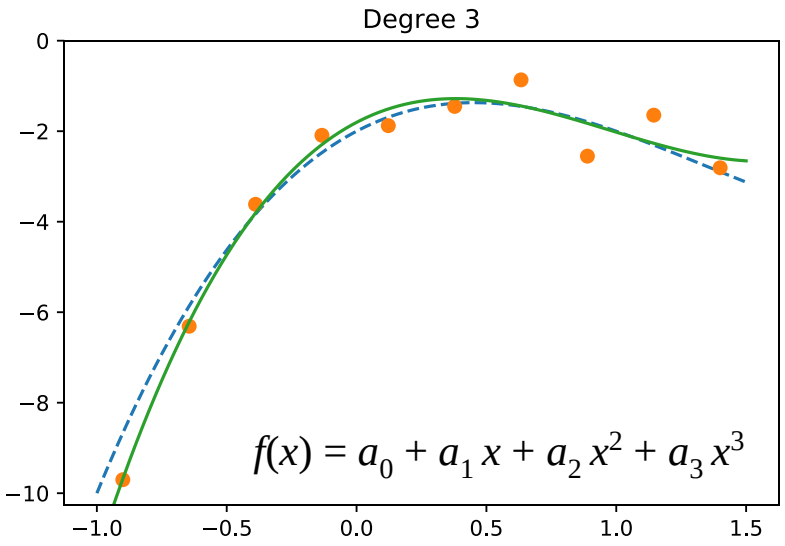

Underfitting happens when the machine learning algorithms cannot capture the underlying trend of the data. For example, for 1st order polynomial (just straight line), as the diagram shows, it does not match (fit) most of the points. In the other words, the model is too simple and subsequently fails to fit the target.

Contrary to underfitting, overfitting occurs when a model memorizes and picks up the noise in the training data.

In other words, if we choose a high degree polynomial it can perfectly fit all the points, but the curve will tend to be “lumpy”. The model gets so attuned to the training data that it fails to perform properly on the new data as a result.




If we guess the order of polynomial (or optimize the hyperparameter properly), see example of 3rd order polynomial below, the Machine Learning result will probably not run through all the points, however, it will run quite close to them.




Under- and -overfitting are the two extremes in which the model performs poorly. In the majority of cases, overfitting is a more common problem than underfitting.

The goal of machine learning is to find a solution that would be a right balance between under- and overfitting.

**Congrats! You did it! Next parts will be easier, we promise.**

# ML Types

ML algorithms can be commonly divided into several categories.

- *Supervised learning* (learning with a teacher)

Requires past data or labeled dataset based on which some predictions are made. The regression (curve fitting) example above is a typical application of the supervised learning: the independent variable is a kind of label, and the aim of regression is to find most likely value for unknown input.

- *Unsupervised learning* (learning without a teacher)

Produces output from unlabeled data (for example, shopping data). the Machine Learning system tries to unearth a certain pattern in the data, the results to be generated are unknown). The most known example is data clustering (for example as used for customer segmentation).

- *Semi-supervised learning* (a mix of both)

Used for expensive data (when getting labeled quality data can be very costly, for example involving experts is required).

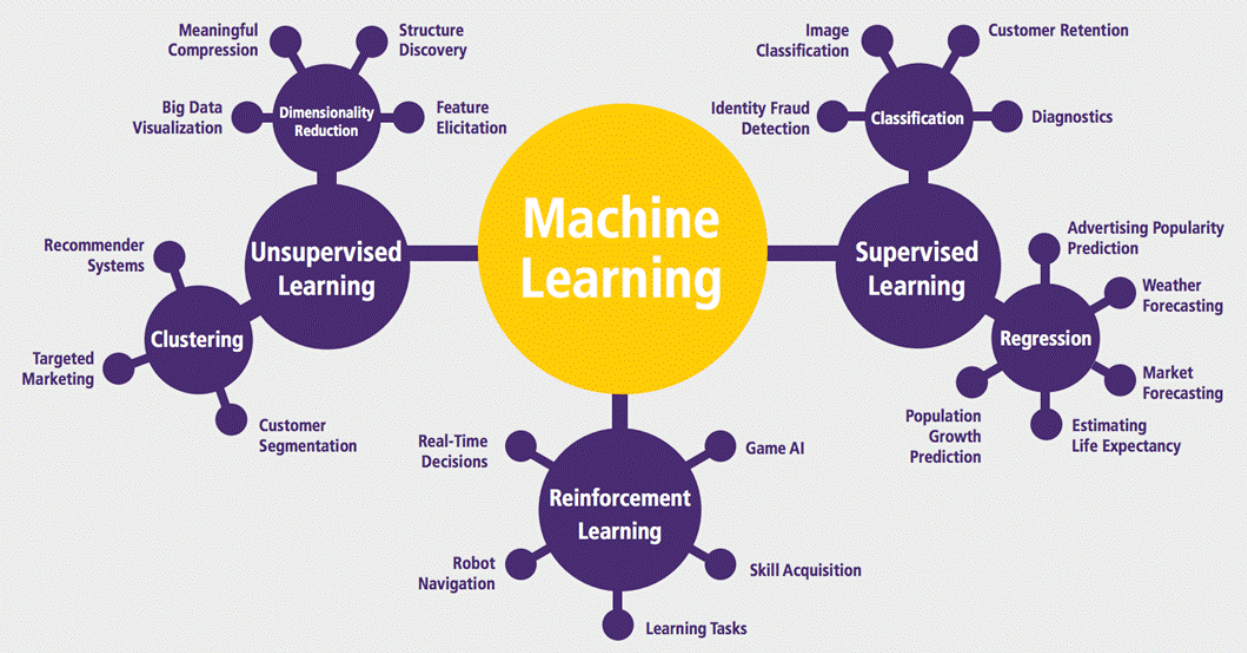
The principle is that data with known labels is mixed with unlabeled data obtained from a similar source. The result is based on combining clustering results with known labels to predict the labels.

- *Manifold learning* (dimensionality reduction)

In case of the high-dimensional data, there are many input parameters (features), and we need to visualize them in some way. As human brain can hardly perceive more than three dimensions, there are some specific ways that allow reducing the data to two or three dimensions keeping the same structure in order to visualize the data. This provides for making better decisions and more accurate predictions.

- *Reinforcement learning* (learning 'on-the-fly')

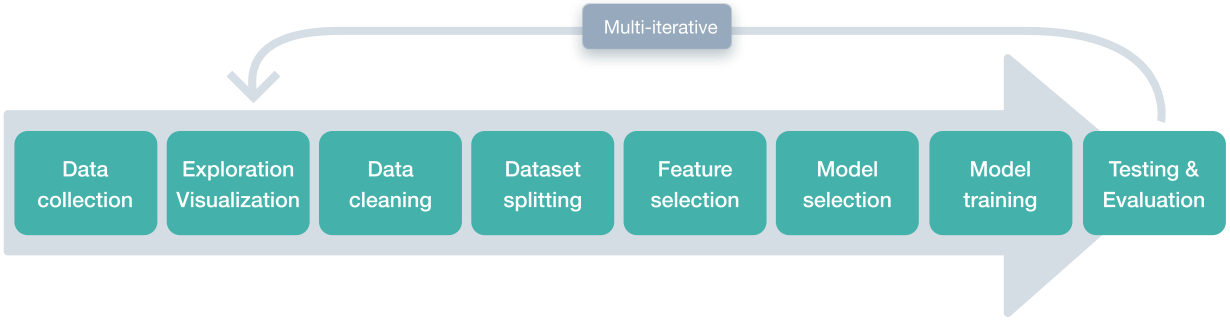
A Machine Learning model learns to respond to changes – obstacles, changing traffic conditions and similar – and learns to take decisions "on the fly". The most known examples of reinforcement learning are self-driving cars and high-quality bots for computer games.




Picture credit: guru99

From the process point of view, Machine Learning is a multi-iterative process that commonly consists of several basic repetitive stages.

The first thing to do is to collect data (or to generate it). Then it is necessary to visualize (explore) the data, to see any patterns it has. At the next stage, the data is subject to cleaning to rule out mistakes and inaccuracies (for example, errors of measuring equipment, database failures).




Generally, datasets are partitioned into training, validation and testing, as detailed in ‘Model Training Principle’ below. Definite features are selected based on available data and model type (one or several) is chosen. The selected model is consequently trained and checked. The process is multi-iterative to ensure proper and efficient model performance.

The above curve fitting (regression) example the dataset is just collection of points (pairs of coordinates), while the model type is a polynomial of certain degree.

# ML Methods

## Feature Selection

By far not all aspects (features) of the data available are equally relevant and important. Usually, we can remove some features to obtain a more stable model. This is what feature selection is all about.

Feature selection is usually performed based on the data on the maximum correlation between the feature and the output variable. This normally improves the model stability and its ability to process new data. Besides, it allows to reduce overfitting as less redundant data decreases chances to make decisions based on noise. Adding new, domain-specific features also contributes to optimizing a model.

A good example is a person's driving records. It has a significant importance for an insurance company that offers personalized coverage. It has little value for a team that build a route recommendation system, at the same time.

Another data processing, somewhat related to the feature selection, is the filtering of data to mitigate the class imbalance. Thus, when analyzing frauds, it is necessary to keep in mind that they normally make up only a tiny fraction of all transactions. Similarly, when dealing with mortality case studies one should consider that lethal rates are commonly low, however, that does not mean lethal cases do not occur at all.

Class imbalance may have adverse effects on the model performance (the ML solution will tend to predict more frequent class for example). Therefore, it is always necessary to analyze imbalances on early stage of a project and later address it properly. Usual solutions include skipping a part of most frequent class or using the models that can tolerate imbalance.

## Feature Engineering

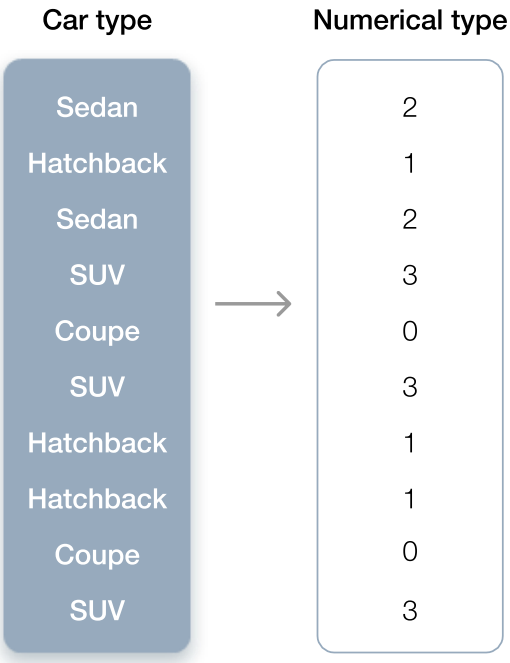
Feature engineering is a process of using domain knowledge of the data to create features and select proper representation of features that make ML algorithms work.

Feature engineering helps to convert the input into data that algorithms can understand. Sometimes the feature engineering is

quite straightforward, for example values of numerical data are features by themselves. Similarly, for pictures, intensities of red green or blue colors in each pixel becomes the features.

In other cases, more tricky transformation is required. For categorical variables, like type of item in catalog are coded using another approach, called ‘one-hot encoding’.

Imagine you need to process data on cars, including the vehicle type, like Sedan, Hatchback and so on. As the first approach, you may just number the available types like as following:



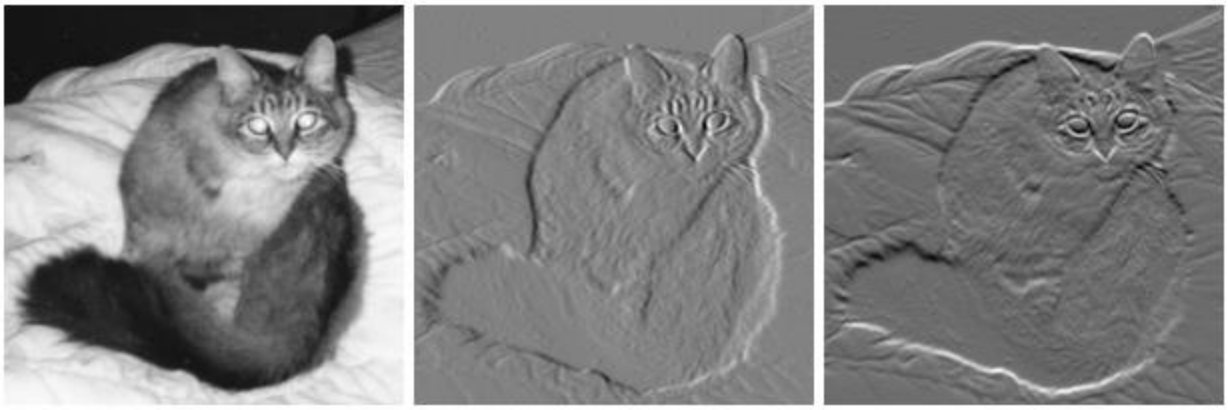
It is a viable approach, but this representation implies an order in data: Hatchback will be closer to Coupe than to SUV. Does it reflect any real fact about the cars? Doubtful!

To mitigate the unwanted correlations, another approach is used: each category becomes a new data column, with 1's for a corresponding value and 0's for all other values, as in example below.

| Car type | Coupe | Hatchback | Sedan | SUV |
|---|---|---|---|---|
| Sedan | 0 | 0 | 1 | 0 |
| Hatchback | 0 | 1 | 0 | 0 |
| Sedan | 0 | 0 | 1 | 0 |
| SUV | 0 | 0 | 0 | 1 |
| Coupe | 1 | 0 | 0 | 0 |
| SUV | 0 | 0 | 0 | 1 |
| Hatchback | 0 | 1 | 0 | 0 |
| Hatchback | 0 | 1 | 0 | 0 |
| Coupe | 1 | 0 | 0 | 0 |
| SUV | 0 | 0 | 0 | 1 |

Such approach provides just a single 1 (hot) value in each row, explaining the name. Beyond categorical features, one-hot encoding is used for text documents.

As more advanced approach to feature engineering is using gradient (derivative) of image for application like edge detection. Such transformations were essentially important before advent of Deep Learning, but is still used today in special cases.



Source: Wikipedia

Wrapping up, proper feature engineering is an essential prerequisite for ML algorithms to make accurate predictions.

## Data Cleaning

Data cleaning in ML is a process used to identify and delete (or correct) irrelevant, unreliable, missing, or inconsistent data to avoid erroneous decisions.

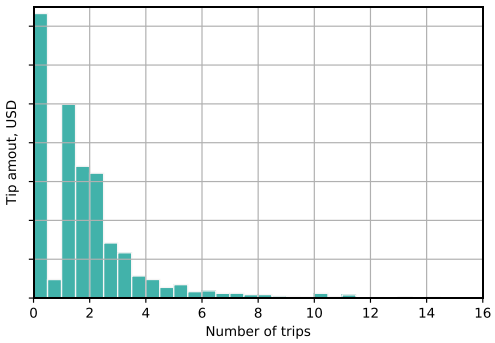
Among others, the categories subject to the data cleaning process include:

- Missing data
- Incorrect values
- Irrelevant data combination

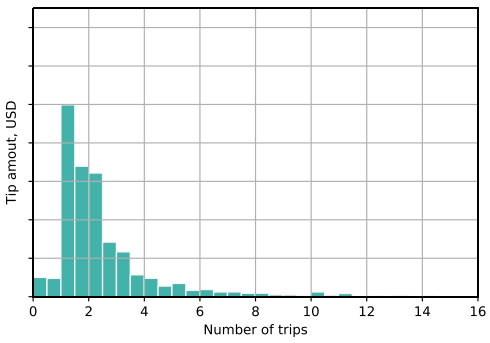
In order to obtain relatively fair results, it is important to filter out, or clean, the data by removing irrelevant data taking into account statistics and domain knowledge. Alongside cleaning unreliable and excessive data, one should also make sure no important values are missing, which is of utmost importance for the ML algorithm to avoid inaccurate predictions and decisions.

Let us take a basic example when cleaning the data is required to get plausible results.

All tips paid




Tips paid in cash excluded



When conducting tipping analysis of taxi drives in Ney York for a certain period it turned out that according to the public (official) sources taxi drivers did not get any tips at all in the majority of cases (see diagram below, left). This runs counter both common sense and common practice in the USA.

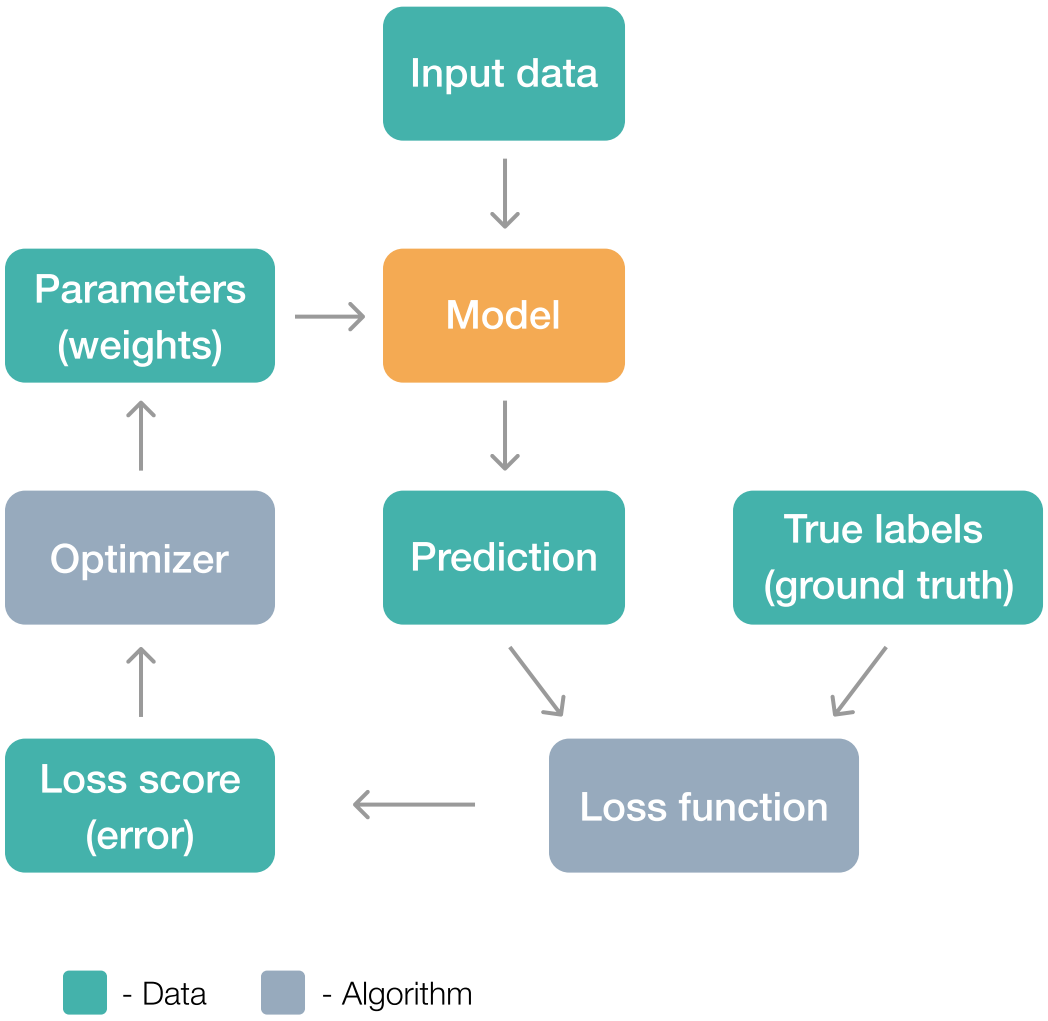
Additional analysis revealed that taxi drivers simply did not keep record of the tips paid in cash (obviously, so as not to pay taxes for tips)[2]. Upon cleaning the data from the tips paid in cash (left), more plausible results were obtained. Thus, to get tidy data, we shall clean the given dataset considering the missing data that may be relevant for further data analysis.

This example shows that why the understanding on how the data is produced is important for cleaning the data. This understanding, or the domain expertise enable on to select not only between relevant and irrelevant data, but also understand if the predictions of ML solution is acceptable.

[2] Example borrowed from Real-World Machine Learning, by H. Brink, J. W. Richards, M. Fetherolf (2016)

# Model Training Principle

As illustrated in the picture, a model training mechanism can be presented in the following scheme.



There are some input data and parameters (weights) used in the model. Based on the input data, the model makes a prediction that is further compared to the initially known correct results.

Loss (or error) function is determined to measure difference predicted and true outcomes. This function is provided to the optimization algorithm that subsequently determines parameters and the process is repeated until the model learns to work perfectly.

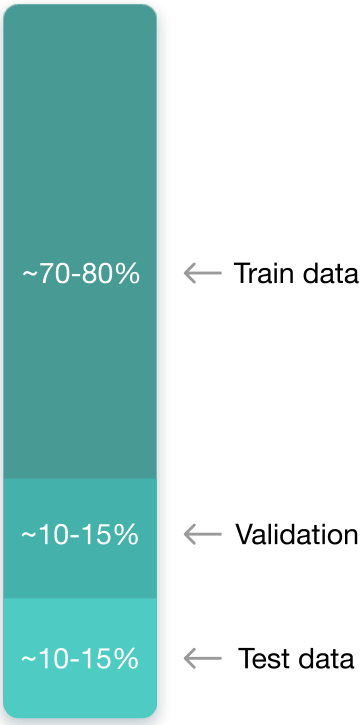
The model training is then just is multi-step optimization aimed to calculation of optimal values of the model weights.

Returning to the curve fitting example, the training process can be explained as follows: the values of independent variable (input data) are used to calculate the values of polynomial (model) according to its coefficients (model parameters or weights). Then the values of vertical coordinate (true labels) are compared with model prediction. Usually, for curve fitting the error (loss function) is MSE (mean-squared error), loss function that is quite important in many real-world ML tasks.

For reliable estimation of the model performance and to avoid overfitting, it is also important to train and test the model on non-overlapping data.

Usually, the following approach is used: the available labeled data is split randomly into training, validation and test parts as depicted below.

Further, the training set is used to optimize the model weights, the validation set is used for on-going monitoring of the model performance (one of the common signs of model overfitting is growth of loss function on validation set). The validation set is also frequently used for hyperparameter optimization.



Finally, the test set is used only for evaluation of performance of model after training. Ideally, to prevent 'leak' of statistics from test set to model, it should be used only once!

## Domain Knowledge

With the rapid progress in the field of artificial intelligence, human beings are apprehensive about robots conquering human thinking! If one extends that argument to expert systems in data science, then one can easily see why some people would see machine learning supplanting domain knowledge.

But the real question lies not in exploring whether data scientists require domain knowledge to build expert systems, but whether the "representation phase" of data can be accurately achieved without involving domain experts.

Domain experts are presumed to be far more capable of identifying, articulating, interpreting, and demonstrating day-to-day process problems in business. As these experts can only well explain specific problem to peers, it is probably absurd to even consider that an expert system can be constructed without their involvement or

guidance. The same should hold true in the case of superior algorithms required to create such systems.

Domain experts usually have an in-depth knowledge of operational processes or tasks and understand the rules of thumb that control the domain. Domain knowledge is gained from actual, practical experience. Then the aim of data scientists is to convert this practical knowledge into meaningful algorithms to automate processing tasks.

Thus, data scientists and domain experts are the two complimentary sides of a complete system development project. In developing expert systems, it is not enough for data scientists to ask questions or find patterns they also need to understand the results.

As an example, Formula-1 pilot is surely capable to evaluate a racing supercar, but it requires full bunch of automotive experts – in engine, tire, transmission, and so on – to design and create such supercar.

Finally, the domain experts are responsible for running the business, so efficient collaboration between data science team and domain experts must be formed.

In particular, good ML solution must include standardized means for data capture, seamless connection of ML tools to existing analytic tools. In other words, the machine learning tool must be tailored to speak the 'language' (tools, visualizations etc.) that domain experts are accustomed with and will embrace quickly.

## Model Evaluation Metrics

Evaluating your machine learning algorithm is an essential part of any project. Most of the times we use classification accuracy to measure the performance of our model, however it is not enough to truly judge our model.

Types of model evaluation metrics:

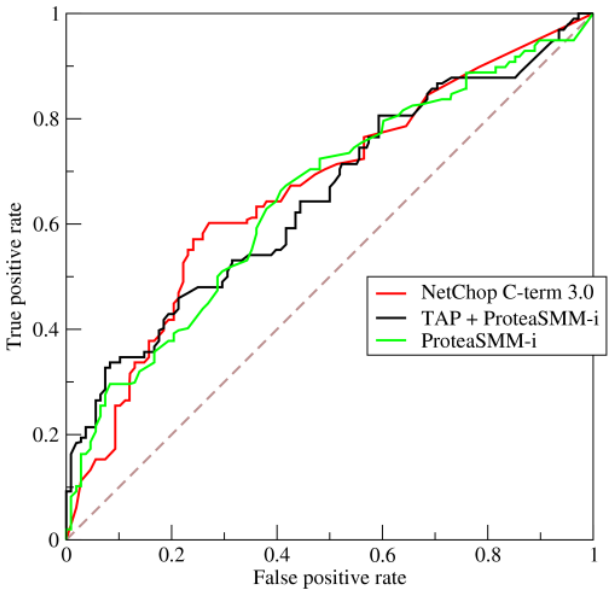
- Classification Precision – ratio of number of correct predictions to the total number of input samples.
- Recall – the ratio of correct predictions in a given class to total number of instances of this class. Contrary to precision, recall characterize the generalization property of ML model: how well the model will work on new data. In practice, it is impossible to have both good precision and recall, and good model is a result of a trade-off.

- **Confusion Matrix** gives us a matrix as output and describes the complete performance of the model (true/false positive and negative outcomes). For example, if an imaginary system trained to tell cats from dogs is evaluated on 13 animals - 8 carts and 5 dogs, its confusion matrix may look like:

| | | **Actual class** | |
|---|---|---|---|
| | | **Cat** | **Dog** |
| **Predicted class** | **Cat** | **5** | 2 |
| | **Dog** | 3 | **3** |

Source: Wikipedia

- **Receiver Operational Characteristic, or ROC curve** is quite widely used metrics for evaluation, used for binary classification problem. In some extent, ROC curve is the graphical view of confusion matrix (for different model hyperparameter) and shows the trade-off between precision and recall. The diagonal (where True Positive rate is equal to False Positive rate) is random classifier (coin flipping) and the closer the ROC curve to upper left corner, the better is the classifier.




Source: Wikipedia

- F1 Score – harmonic mean between precision and recall, and, similarly to ROC curve it helps to overcome the precision-recall trade-off. The range for F1 Score is between 0 and 1 and it tells you how precise your classifier is (how many instances it classifies correctly), as well as how robust it is (it does not miss a significant number of instances).
- Mean Absolute Error – average of the difference between the Original Values and the Predicted Values. It gives us the measure of how far the predictions were from the actual output. However, they don't give us any idea of the direction of the error i.e. whether we are under predicting the data or over predicting the data.

- **Mean Squared Error** is quite similar to Mean Absolute Error, the only difference being that MSE takes the average of the square of the difference between the original values and the predicted values

# Deep Learning

## What is Deep Learning?

Deep learning is a branch of machine learning based on a set of algorithms that attempt to model high level abstractions in data.

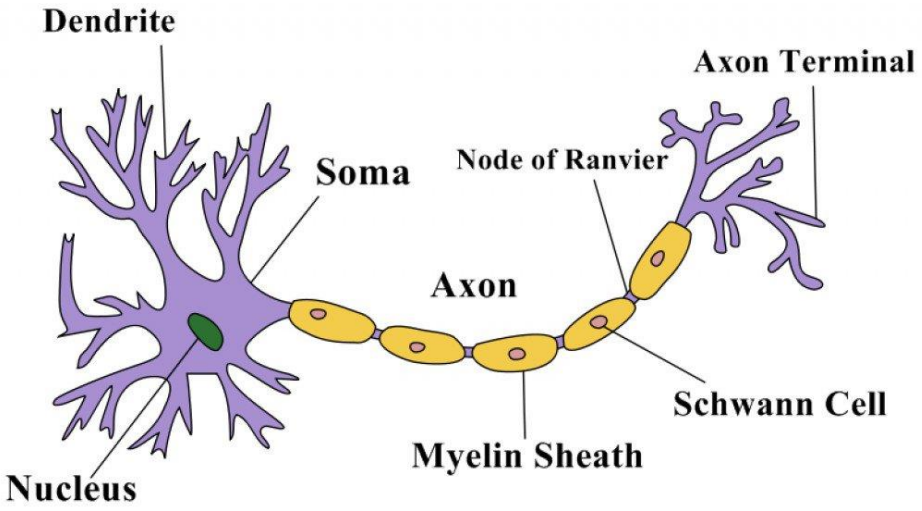
Deep Learning (DL) and Neural Network (NN) are currently driving some of the most significant inventions in last decade. Their outstanding ability to learn complex dependencies from data makes them the enabling technology for many machine learning tasks.

Deep Learning and Neural Network lies in the heart of many innovative products like self-driving cars, image recognition software, automatic text recognition and translation, recommender systems and so forth. Apparently, being a powerful algorithm, it is highly adaptive to various data types as well.

Deep Learning is an important generalization of ideas described above: modern DL methods supports models with tens to hundreds million of parameters and able to support very complex tasks, such as those appearing in machine translation or image recognition.

Returning to the curve fitting, DL methods provides an approach to fit any nonlinear curve (function), including functions with hundreds of arguments.

Neural Network, also referred to as Artificial Neural Network, is named after its artificial representation (or, rather, mimicking) of functioning of a human's *nervous system*, especially the visual cortex. Nervous System comprises of millions of nerve cells or neurons, each having the following structure:

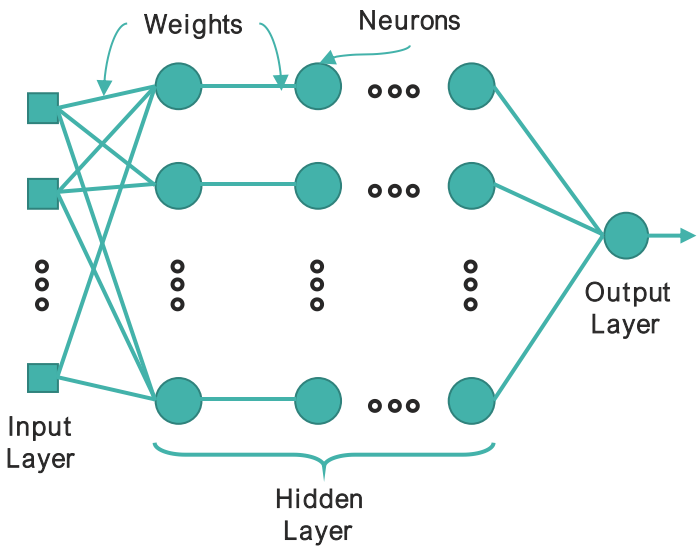



Source: Quasar Jarosz CC BY SA 3.0, via Wikimedia Commons

In simple words, each neuron receives input information in form of electrical impulse from other neurons, does some processing and then transmits it to other cells. Though a single neuron performs only very basic operation, collective actions of all neurons in the

nervous systems results in highly complex behavior and cognitive abilities of animals, including us humans.

Artificial Neural Network works in a very similar manner and can be visually represented as a set of layers of neurons (basically, an input layer, several middle – or hidden – ones, and an output layer). At each layer, the individual neuron is taking in some input information, weighting it accordingly and then passing on a modified version of this input to the next layer. Thus, the output of each layer forms the input of next one.



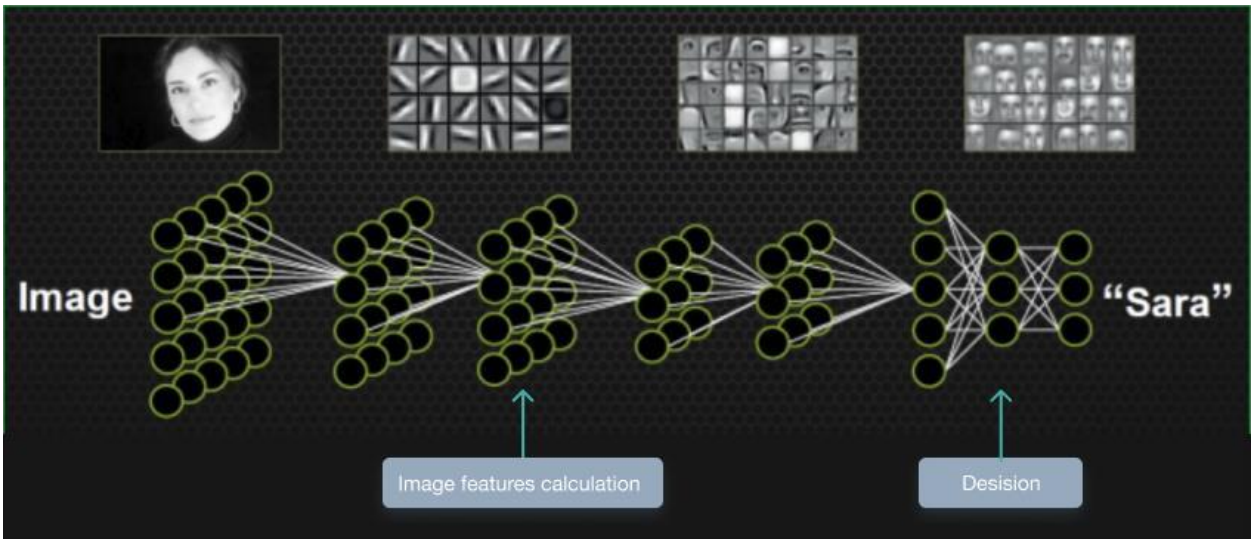

While designing a neural network some random or roughly estimated values are assigned to weights but it's clear that they need to be adjusted as errors occur at the output level. This is resolved through Backpropagation (BP) algorithms that help

determine the loss (or error) at the output and then propagate it back into the network to update our weight values. This is the cornerstone of the learning process of neural networks.

Choosing the right number of layers, neurons and connections between layers (hyperparameters of deep learning model) is also important for building accurate models. Usually, more complex tasks requiring a bigger number of elements in a neural network. In a deep neural network, there are many layers between the input and the output, which allows the algorithm to use multiple processing steps each implying numerous linear and non-linear transformations.

## Example

To explain how Deep Learning works let us take an example of CNN – the Convolutional Neural Network – which is currently the standard approach in Deep Learning for analyzing image data.




Source: *Jason's Machine Learning 101* presentation, Dec 2017

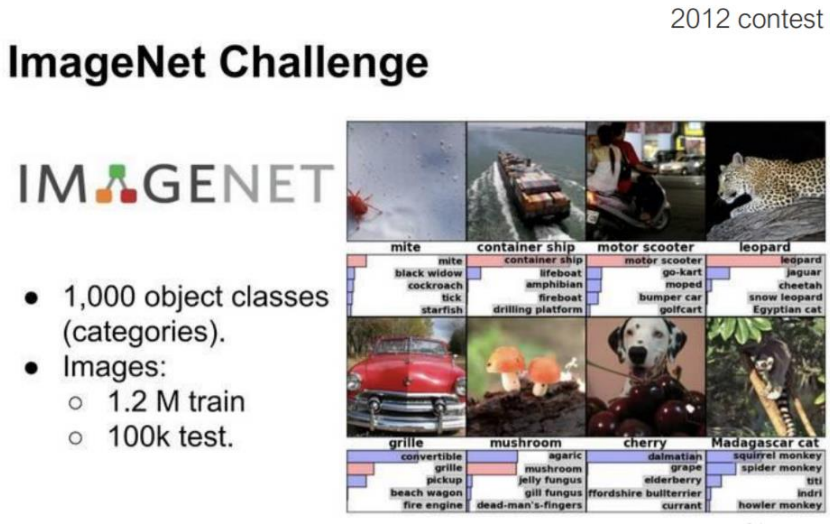
If we submit an image of a human face as an input data, the first layers will detect some high-level features like outlines and edges. The following levels will then face-shaping object like eyes, nose, and lips be recognized. In the end, the neural network will "assemble" the attributes of an individual which enables the system to figure out who is presented on the picture.

## ImageNet Revolution

The history of Neural Networks may be traced back more than 60 years but the real breakthrough that largely influenced its booming

development took place just recently. The **ImageNet**[3] project is a large dataset of pre-annotated images designed for use in research on visual object recognition. And since 2010 it had been running its annual contest where researchers from all over the world competed in correctly classifying and detecting objects using AI-based programs.

It was in 2012 when Geoffrey Hinton, Ilya Sutskever, and Alex Krizhevsky from the University of Toronto submitted a deep convolutional neural network architecture called **AlexNet**, which was 41% better than the next best rival.




Source: Xavier Giro-o-Nieto presentation, Jun 2017

[3] http://www.image-net.org/

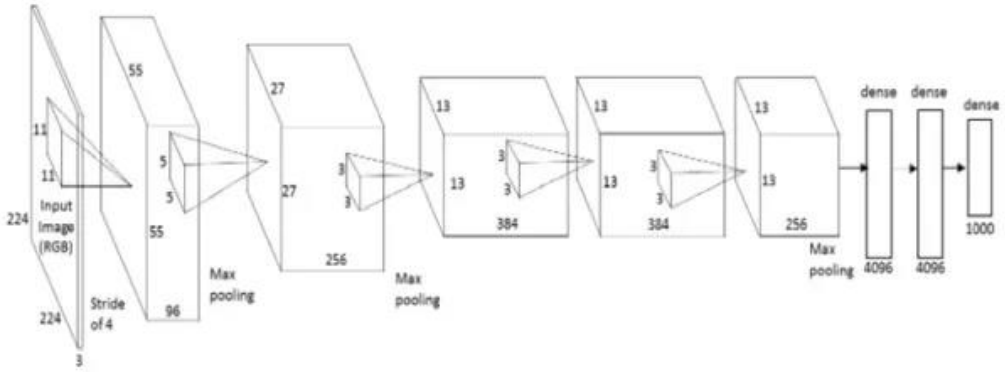
The idea of CNN is to implement a series of digital image filters (similar to the image processing tools like Photoshop are using) but the filters parameters are optimized according to input images and their labels.

Today, these convolutional neural networks[4] are everywhere — Facebook uses them to tag your photos; self-driving cars are using them to detect objects; basically, anything that knows what's in an image or video employs CNN.

[4] https://qz.com/1034972/the-data-that-changed-the-direction-of-ai-research-and-possibly-the-world/

## Deep Learning Architectures: Image Classification

AlexNet was the first deep architecture, which helped pave the way for groundbreaking research in Deep Learning as it is now.



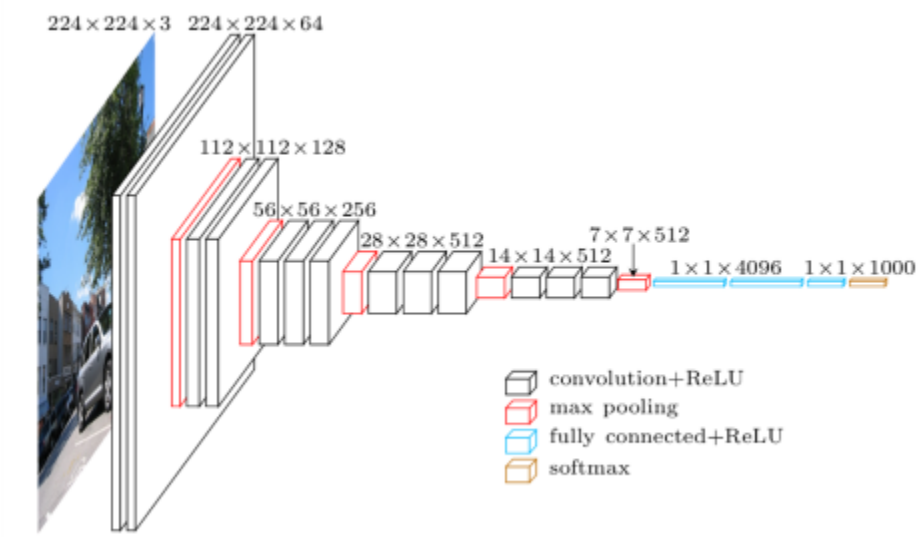

Source: Analytics Vidhya

Its architecture is quite simple and was first conceptualized in 1980s. What helped it stand out is the scale at which it performs the task and the use of GPU for training instead of central processors. This switch alone enabled a 10-fold speed up of the training process.

Although a bit outdated now, AlexNet is still used as a starting point for applying deep neural networks for many tasks, whether it be computer vision or speech recognition.

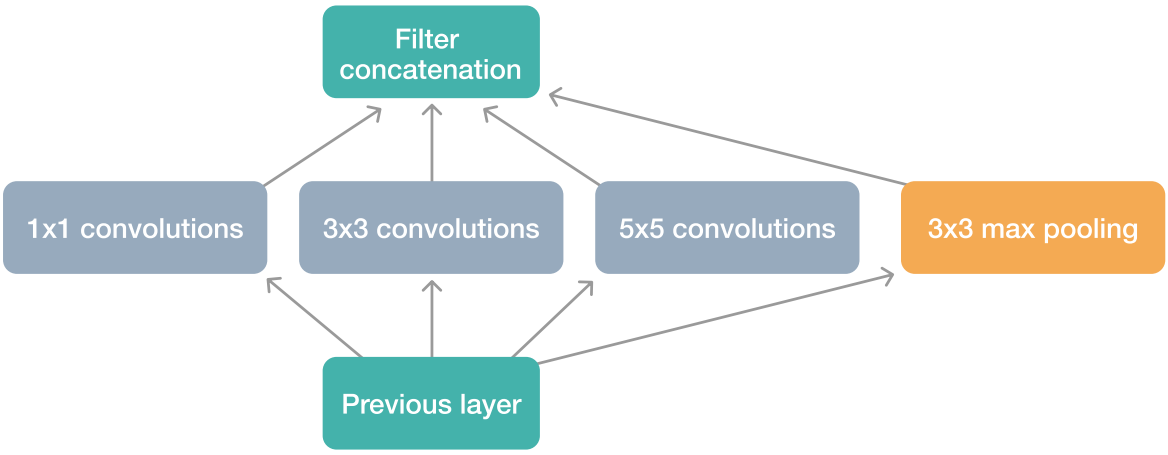
Another widely used neural network architecture is VGGNet that was introduced by Visual Geometry Group from University of Oxford. This network is characterized by its simplicity, using only

3×3 convolutional layers stacked on top of each other in increasing depth. It has a very low error rate but rather slow for training.



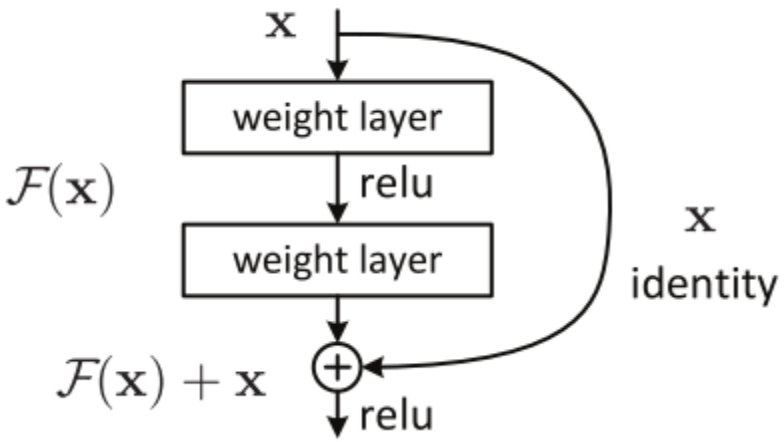

Source: Analytics Vidhya

VGGnet was the 1st runner-up at the ImageNet contest in 2014 giving way to GoogleNet (or Inception Network). Inception architecture achieved a top-5 error rate of 6.67% which was very close to a human-level performance. Inception Network consisted of a 22-layer deep CNN but with the number of parameters reduced from 60 million (AlexNet) to 4 million. The main advantages of Inception model are its training speed and a relatively small size (96MB) compared to VGG (>500MB).




Source: Analytics Vidhya

**ResNet**, introduced in 2015, is another example of how powerful a deep learning architecture can be.




Source: Towards Data Science

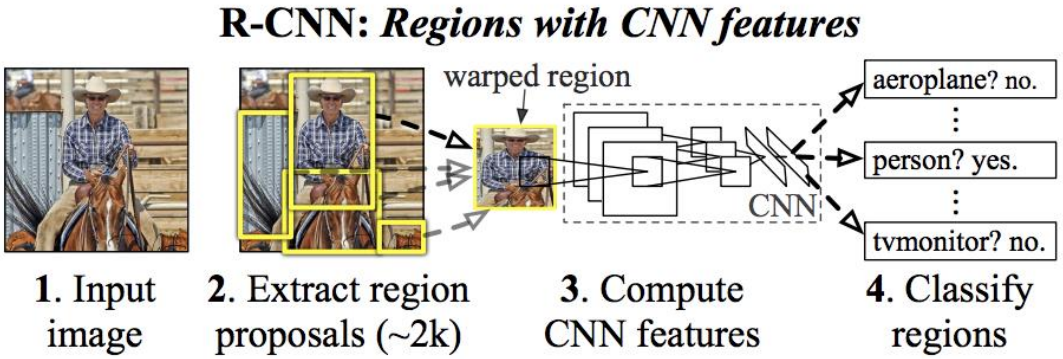
It used a “skip connections” technique (pass-through connection skipping several layers) which enabled hundreds and even

thousands of layers to be created and trained in a single network while remaining a reasonably low complexity. It achieved a top-5 error rate of 3.57% during the ImageNet's 2015 contest and surpassed human level.

## Deep Learning Architectures: Object Detection and Image Segmentation

Another important task that Deep Learning possible is more fine-grained analysis of images: finding the objects like persons or cars in image (object detection) and classifying different parts of image, like road, grass, sky (image segmentation).

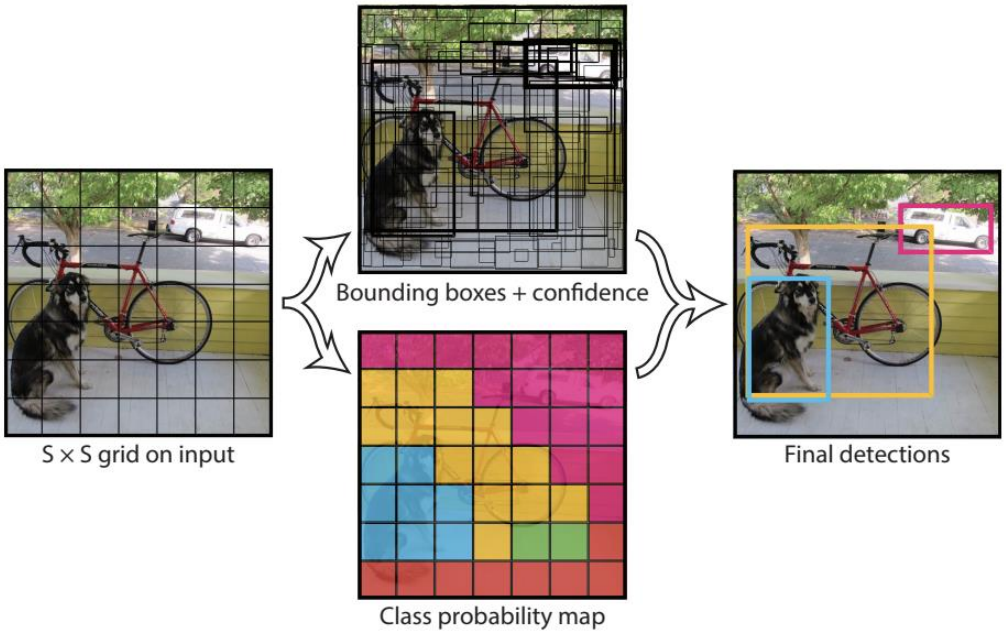
Let's explore several DL architectures used for these tasks.




Source: Towards Data Science

Region Based CNN (or R-CNN) architecture is a popular approach to object detection problem. Instead of processing an entire image and deciding the label for it, the R-CNN finds rectangular regions ('bounding boxes') containing some specific object, like person, car or a house.

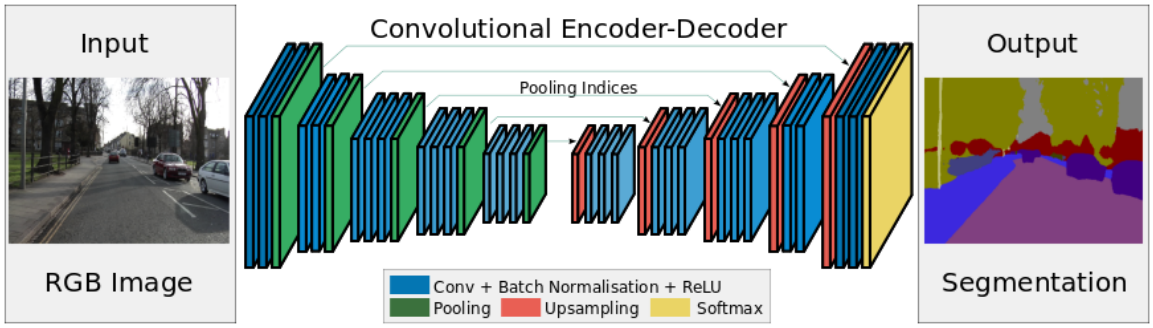
YOLO (You Look Only Once) architecture is also using a "bounding boxes" approach but works faster than may alternatives. As seen on the picture below, it first divides the image into defined bounding boxes and then identifies which object class do they belong to with some confidence level. The boxes with highest class probability then used to locate the object on the image.




Source: Towards Data Science

SegNet is another widely used deep learning architecture originally developed by members of the Computer Vision and Robotics Group[5] at the University of Cambridge. It consists of sequence of processing layers (encoders) followed by a corresponding set of decoders for a pixelwise classification.

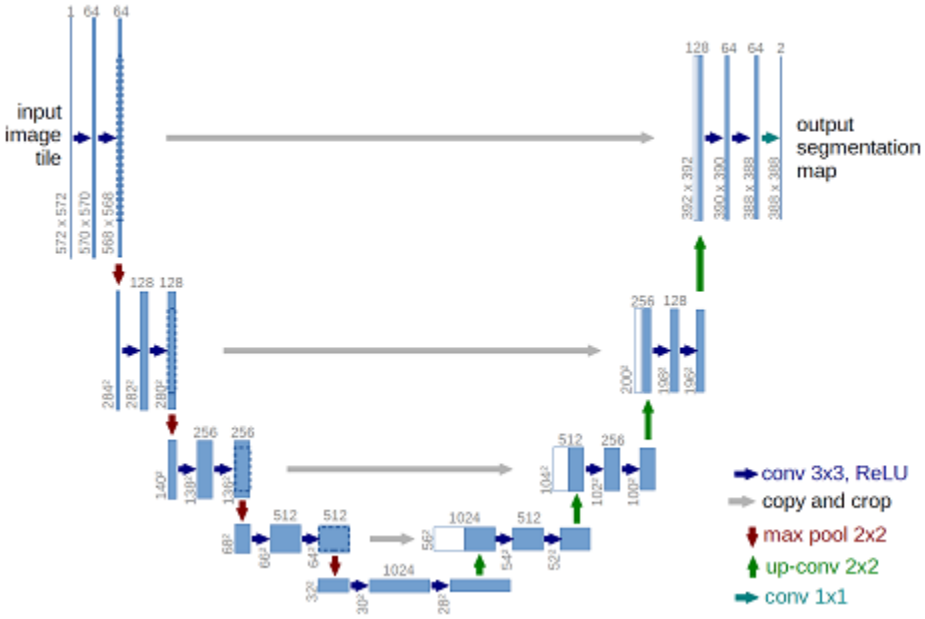
Its core advantage is the use of max pooling in the decoders to up-sample low resolution feature maps. This allows to retain high frequency details in the segmented images while reducing the total number of trainable parameters. It is very useful in image segmentation tasks including road scene understanding for self-driving cars and street monitoring.




Source: http://mi.eng.cam.ac.uk

[5] http://mi.eng.cam.ac.uk/Main/CVR

Another popular solution for Image segmentation is U-net network architecture, initially proposed for segmentation of biomedical images (for example, to find an anomaly in X-ray image or different types of structures in microscopic image).




Source: Freiburg University

The U-net architecture uses contraction-upsampling principle used in SegNet, but complements it by combining the high-resolution features with upsampled output and use of large number of feature channels in upsampling part, which allow the network to propagate context information to higher resolution layers.

Another deep learning architecture worth mentioning is GAN or Generative Adversarial Network. It can generate completely new sets of images that are not initially represented in the dataset but look realistic enough to be hardly distinguished by people from the human-generated ones.

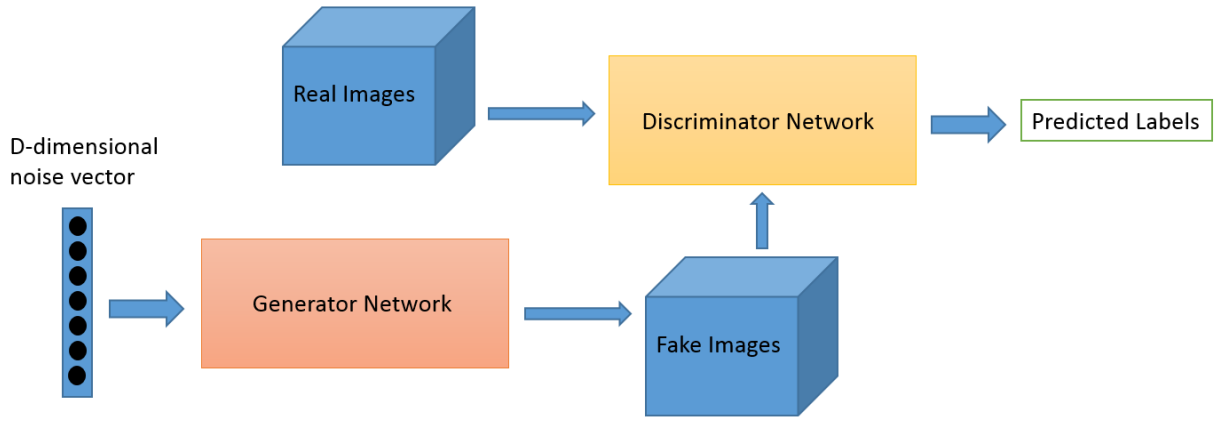



Image credit: https://skymind.ai

The idea of GANs is joint training of two neural network models. The first one (generator) tries to create some image from random noise, while the second model (discriminator) tries to identify if a generated image is close to real one.

Below are two sets of photographs that were generated by GAN in 2014 and in 2017 respectively. If identifying the former one would be an easy task for a human eye, things are getting much trickier with the latter set of images which look just like photographs of real

people. You might have even “recognized a Hollywood star” in one of those faces.

Source: arXiv.org

Source: Medium

# Risks and Challenges

Machine Learning is a very powerful tool that doubtlessly holds an enormous potential for organizations, but it also poses a substantial risk, especially if not treated properly.

## Types of Risks

Deloitte[6] splits machine learning algorithmic risks into 3 main categories, which are the **input data**, **algorithm design**, and **output decisions**.

The first group of risks relates to 2 main problems: insufficient, low quality, outdated or irrelevant **data sets** and **biases**, that can cause a model to be widely inaccurate.

The first risk area is associated with a volume and quality of a data set. Do you know the origin of the data? Wasn't it faked? Does it provide enough information and variability to be used in the model? Answering these questions is necessary in the preparatory step.

[6] https://www2.deloitte.com/us/en/pages/risk/articles/algorithmic-machine-learning-risk-management.html

There might also be different compliance issues depending on your jurisdiction and where does your data come from.

On the other hand, all people have biases, and since machine learning models are made by people, we cannot avoid them being biased. The situation gets even trickier given that data itself can have biases. Many business cases will require data to be adjusted for special circumstances where new parameters or conditions come into play.

That is why, as we already discussed in section Domain Knowledge, a close collaboration between machine learning and industry experts is strongly required, especially at a project's early stage.

**Algorithm design** is vulnerable to risks as well. Let alone coding errors, the logic and assumptions we pledge while training a model are often false and training results are disconnected from the reality. That leads to a model being useless in the real world.

Even when the data is flawless and algorithms were chosen and tweaked perfectly, there is a risk the **results of the machine learning process** will be misinterpreted or will disregard its underlying assumptions. Such situations are very common in business. As machine learning models are built to help in decision-making

process, it is essential that an understanding of its outcome is fully aligned across stakeholders.

## Mitigating Machine Learning Risks

Anticipating your question: given that there are so many risks how can we make sure the machine learning process will go smoothly?

The first thing that all organizations leveraging this technology need to work out is a clear formulation of goals they want to achieve through machine learning projects. Having internal strategy and management scheme is important, as well as to have clearly outlined principles and policies and to assign accountable employees.

Which is indispensable, is to employ a team of experienced data scientists and machine learning engineers (or contracting a trusted service provider) alongside industry professionals. As we previously mentioned, a teamwork of data engineering and business area experts is required to mitigate multiple data and algorithm risks and to help understand how results could differ from expectations.

For some projects (especially on the enterprise level) a group of independent data experts will be needed to provide an unbiased

evaluation of the model on its fairness, compliance with existing privacy rules, and explainability. Making models explainable and interpretable are actually one of the biggest challenges that machine learning engineers face nowadays (especially for most Deep Learning that is necessary for most demanding tasks). Let's have a closer look at these and other natural limitations of machine learning.

## Limitations of Machine Learning

Machine learning has an enormous potential for solving business tasks, improving processes and products. But one should not ignore its limitations when deciding on running a project or implementing the technology.

1) Lack of explainability

But just like personal computers were not built to interpret their decisions to general users, Artificial Intelligence was not designed to explain its predictions. However, as Machine Learning is becoming more pervasive in all spheres of live, human need a clear understanding of its decision-making process. Car manufacturers need to have a full control over AI that works is in the core of their

autopilot systems. Investment brokers that utilize machine learning must be able to explain the grounds for their decisions to a client. The problem is that in many cases, especially those involving complex Deep Learning architectures, data scientists simply don't have an explanation of how these models work.

What is known are the input data and the output results, whereas all the rest is hidden inside a so-called "black box". Some even consider machine learning a new form of alchemy[7]. But the technology is based on computational algorithms, not magic. In reality, even sophisticated non-linear models appeared to be interpretable thanks to a number of recently developed explainability techniques[8], and this field is expected to progress quickly.

2) It requires really big data sets

As we already mentioned, machine learning models require vast amounts of data to be trained – moreover, these data have to be clean, unbiased, and variable enough and to provide quality results. It may seem that obtaining data is not a problem in the world where

[7] https://www.sciencemag.org/news/2018/05/ai-researchers-allege-machine-learning-alchemy

[8] https://medium.com/@Zelros/a-brief-history-of-machine-learning-models-explainability-f1c3301be9dc

2.5 quintillion bytes of information is created each day. In fact, collecting a sufficient amount of data requires either time or money, often both. Data labeling, one of the steps required for tasks like image recognition and classification, is a laborious and time-consuming process that companies often prefer to outsource. Another challenge that organizations face at data collection stage is privacy protection policies that limit collection and utilization of personal information in certain countries or regions.

3) Talent scarcity

When an extensive dataset is in place, you have to inspect and clean the data, select features, then choose or build a custom model, apply a set of machine learning algorithms, pick up and tune parameters, constantly monitor performance and make necessary adjustments.

No wonder this would require a substantial expertise and strong knowledge in data science, machine learning engineering, and software development, all at the same time. Not many companies possess these resources in abundance or can easily hire them. Tencent estimated in 2017 that there were just 300,000 AI engineers worldwide. Although a rapidly growing interest in the field is undeniable, the amount of newly trained labor is nowhere near the

speed at which ML-required jobs are created. That is not to mention high-level and experienced specialists that can manage complex projects and conduct serious artificial intelligence research. There are fewer than 10,000 of those in the world, and the number is not expected to grow exponentially. Talent deficit naturally triggers salaries which have already surpassed reasonable limits in many countries. All it makes machine learning adoption increasingly difficult for companies, let alone startups that are more actively turning their eyes towards external provides and consultants.

Although an effective model for small and middle-sized companies, finding a reliable partner with deep relevant expertise and an appropriate level of compliance emerges as another challenge.

4) It takes time to get desirable results

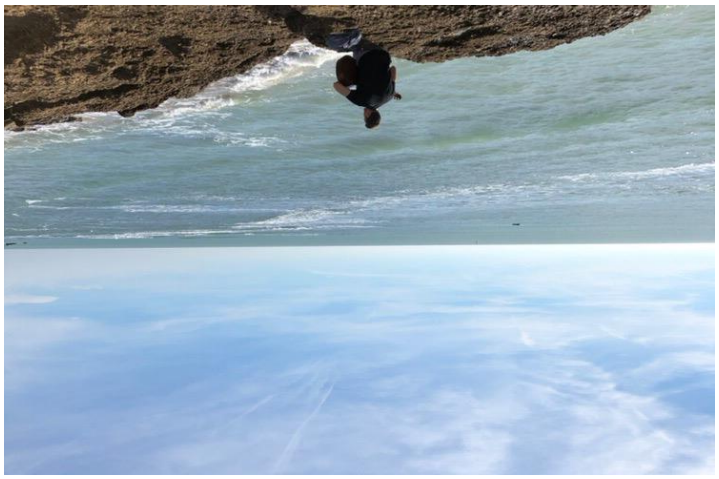
One of the reasons why so many organizations fail in machine learning projects is because they expect immediate results. A lot of business owners and decision makers tend to approach Machine Learning projects the same way they do classic software development, which is wrong.

In software development goals are well-defined, algorithms are transparent, and outcomes are binary, while in machine learning algorithms are designed to improve accuracy (or success rate) over

time and outcomes are often subjective. Time and patience are indispensable prerequisites for maximizing an impact of machine learning within an organization. As more and more companies utilize neural networks and ML algorithms to work on running data that they users generate every day, the process may well become infinite requiring engineers to continuously tweak and adapt parameters to new sets of data.

5) Machine learning is not good at understanding context

Machine Learning algorithms are almost brilliant in recognizing objects and classifying images, but it has some troubles with understanding context.



You can say without hesitation that this image is rotated upside down, but it isn't that obvious for AI. If the model was trained to locate people on images it would do just that. It can learn to

recognize when a picture is upside-down given it has seen a large dataset of both normal and inverted images. But then algorithm runs into something like this:

Image credit: howstuffworks.com

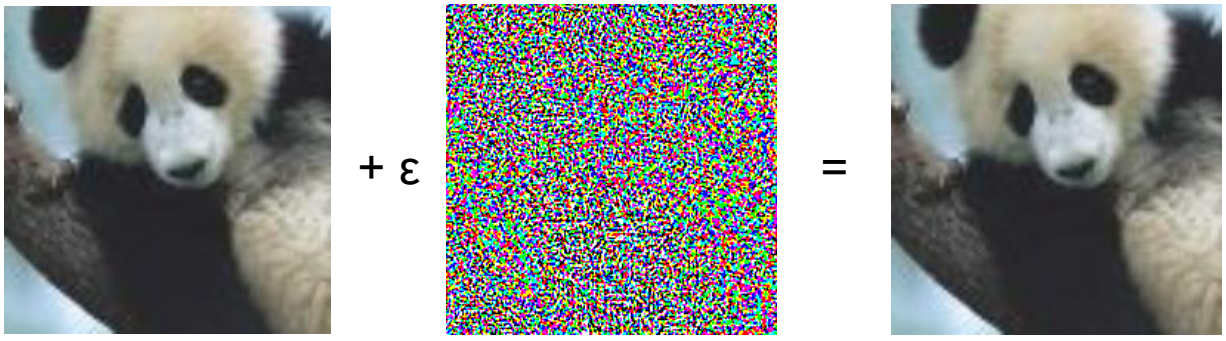
It most likely will make a mistake because context is crucial here. Humans understand the context instinctively. When looking into provisions of a Civil Code, you'll deal with exactly the same content as thousands of other people but the information you would seek there and even the meaning of this information would depend on your context.

Until recently, that was a "mission impossible" for computers but latest achievements give hopes that the solution will be found soon. A number of modern approaches, like vectorization of words in natural language processing, are yielding really promising results in

solving the context understanding problem. A breakthrough in this field can make a true revolution in certain areas like legal practice and knowledge management but as of yet it remains a limitation of machine learning.

5) ML algorithms can be fooled

Even a well-functioning mathematical model — one that relies on good data — can still be tricked, if one knows how it works. Like on the given example, overlaying a minor noise on the image can lead to its miscategorization.




**“panda”**
57.7% confidence

Random noise

**“gibbon”**
99.3% confidence

Source: Explaining and Harnessing Adversarial Examples

This also keeps many companies, organizations, and governments from a large-scale adoption of machine learning, because as long as it can be tricked so easily, the risks would always prevail over benefits.

Another example was published with respect of self-driving cars: a slight change of lightning conditions has causes algorithms to make a wrong decision, killing passengers in a real car.

(a) Input 1 (b) Input 2 (darker version of 1)

Source: DeepXplore: Automated Whitebox Testing of Deep Learning Systems

We are still far away from having an impeccable machine learning which can be used everywhere with little or no limitations and risks. Building transparent and interpretable yet robust and secure algorithms is a mountain to climb for researchers and practitioners. However, by having business owners and decision makers at every level educated on its underlying risks and limitations, we are making the task substantially easier and bringing us closer to solving these challenges.

To wrap-up this section, we want to remind that we humans are also suffer from errors, including visual illusions. Sometimes these errors lead to huge loss.

What is different for Machine Learning algorithms that the latter usually make quite non-human errors. As a remedy, many real-business implementations of Machine Learning employ the ‘human-in-the-loop’ principle: the ML algorithm handles repetitive tasks with human expert fixing the results in case of errors. Besides, practical solutions are usually a combination of ML model and a classical, rule-based algorithms, aiming to compensate limitations of both approaches.

# Hardware and Software

After reading the news on AI progress, one usually expresses a natural concern that using ML implies specialized software and expensive hardware. While it is true to conduct groundbreaking research, practical application of ML techniques can be normally started virtually for free!

## Software

First, recent progress in ML techniques was heavily influenced by the open source movement. Many important software packages and libraries are made free to use and available for commercial use. This concerns both software from academic teams and software implemented by major corporations, like Google, Facebook, AirBnB or Twitter.

Most popular ML libraries are based on Python language ecosystem. Python is quite popular in academia and industry and have very large ecosystem of data preprocessing tools, powerful Deep Learning packages and visualization tools. Beyond this, with

Python it is very easy to wrap the ready ML solution into a web service.

The second popular ML ecosystem is based on the R language. This language is freely available, it was designed especially for statistical calculations and has vast number of libraries, supporting almost any statistical method known. Besides Python and R, Machine Learning solutions can be also implemented in other popular languages, such as C/C++, Java, Matlab or even JavaScript.

## Datasets, Pre-trained Models

Quite frequently, starting a ML project may encounter 'chicken-or-egg' problem. To show the utility of ML in a company it is necessary to have the data, but collection (and annotation!) of data can be expensive or complex process. Sometimes, to convince the stakeholders that ML solution is useful, it is important to show initial result with very limited or even no data!

Fortunately, today it is possible to find open datasets for almost any domain. The size and quality of such datasets may vary, but in many cases, these are enough to make a demo or proof-of-concept.

For examples of open datasets, we recommend looking in the following sources:

- https://www.kaggle.com/ - in the popular ML contest site (now owned by Google), where the contest organizers share data on different topics. The sensitive fields are frequently obfuscated (numerical data rescaled, texts replaced by hash numbers etc.), but such obfuscation normally preserves the patterns in data. Moreover, not business confidential data, such as product reviews or images, is usually provided openly (this is nevertheless visible for anyone). In other words, reusing Kaggle data give access large open datasets with proper annotations.
- https://data.world/ is a big collection of datasets from different public sources.
- https://archive.ics.uci.edu/ml/index.php - the Machine Learning repository at University of California, Irvine is probably oldest collection of research datasets. While most of these are quite small on the modern scale and not suitable for Deep Learning, it is a good resource for research, teaching and testing of algorithms.

- The Google dataset search, available at https://toolbox.google.com/datasetsearch enables to find the data from these and other sources.

Besides the dataset sources discussed above, many open source libraries have the data for training and testing in their repositories.

Beyond the ready datasets, software libraries also frequently contain so called ‘pretrained models’: ready to use neural networks that can be applied just after download. In many cases, it means that a proof-of-concept of ML project can be tested in time span of a day to a week!

## Hardware

When you read about new shiny achievement on AI or related field, you may notice that authors spend thousands of hours on high-performance computing cluster. You may expect that your own ML project will require similar computation power. Is it true?

The answer is **not**.

In many cases, the ML project can be started using just an ordinary office (better, multimedia or gaming) desktop or laptop. Any modern

consumer computer will normally have fast multicore CPU and fast disk storage to process your data. The only additional important requirement is that this consumer computer should have enough RAM.

Using of consumer computer will not permit you to train complex Deep Learning models but pretty sufficient to explore and visualize the data, train classical ML models, implement transfer learning, or test available pre-trained models on your data.

As your requirements will grow, it is advisable to use more powerful computer as available from major cloud providers. Trio of most popular players on this market includes (at the time of writing) AWS from Amazon, Azure from Microsoft and Google Computing Cloud. All these providers also have specific free trial programs, enabling to select the type of a computer (instance) that best suits your needs.

It is important to note that as most ML algorithms benefits from parallel computing architectures. The most available parallel architecture today is the Graphical Processing Unit (GPU) from NVIDIA. GPU-equipped instances are available from most cloud providers (but usually are in high price segments).

In addition, the sufficiently powerful computers are available also from many ‘bare metal’ providers over the world. Check your

favorite hosting provider if it has options suitable for Machine Learning tasks.

Starting by renting computing power also enable try different configurations before building your own computing power.

# Useful Materials

At the end of our somewhat lengthy Guide, we provide a curated list of resources for those who want to dive deeper. This list in by no means exhaustive (after all, Machine Learning is very broad and active developing area, both in industry and in academia). However, we believe the resources below are quite helpful to improve one's understanding of the field.

## Online Courses

- Machine Learning course by Andrew Ng. Probably the best you can find online https://www.coursera.org/learn/machine-learning
- Deep Learning Course at Udacity https://www.udacity.com/course/deep-learning--ud730
- Open Machine Learning Course https://mlcourse.ai/

## Books

- Deep Learning, by Ian Goodfellow, Yoshua Bengio, and Aaron Courville, MIT Press, 2016,
  http://www.deeplearningbook.org/
  Perhaps the most famous recently published book on Deep Learning, though mainly targeted to readers from academia.
- Artificial Intelligence: A Modern Approach, by Stuart Russell and Peter Norvig, Third edition, 2016,
  http://aima.cs.berkeley.edu/
- Deep Learning with Python, by François Chollet, November 2017, ISBN 9781617294433
  https://www.manning.com/books/deep-learning-with-python
- Machine Learning Yearning, by Andrew Y. Ng, 2019
  https://www.deeplearning.ai/machine-learning-yearning/
  (a free book mainly focused on running AI department in real business)
- The Hundred-Page Machine Learning Book, by Anriy Burkov, 2019
  http://themlbook.com/

## Blogs

- Towards Data Science, https://towardsdatascience.com/
- MIT News: Artificial Intelligence, http://news.mit.edu/topic/artificial-intelligence2
- O'Reilly: Artificial Intelligence, https://www.oreilly.com/topics/ai
- Artificial Intelligence on Reddit, https://www.reddit.com/r/artificial/
- Machine Learning Mastery, https://machinelearningmastery.com

## Podcasts

- O'Reilly data show, https://itunes.apple.com/us/podcast/oreilly-data-show/id944929220
- Data Hack Radio, https://soundcloud.com/datahack-radio
- This Week in Machine Learning, https://twimlai.com/

# Thank you for making it till the end!

We've thrown a lot on you but hope it was time worth investing. Now it's your turn to put these ideas into practice. It must be way easier now to see which parts of your product offering or business processes can be improved with Machine Learning and consciously evaluate all benefits and risks connected to implementing the technology. Our team of ML experts is ready to provide you a comprehensive consultation and assist in planning and realization of your projects. Send us a message to hello@silkdata.ai with the promo-code stated below, and we'll grant you 3 hours of expert-level Machine Learning consultancy for free.

Your personal code to claim 3 hours of free Machine Learning consultancy

## SILKMLGUIDE3

Silk Data is a European Machine Learning and IT consulting company founded by PhD titled experts with an average of 25 years of academic and business experience. Our mission is to help companies solve their problems by using full potential of Machine Learning, Artificial Intelligence, and advanced digital technology. For more information, please visit www.silkdata.ai

silk data