AICOMPANY

Training Material

Become the expert in Artificial Intelligence in your industry

Purpose of the Machine Learning Bible

Lets have some fun with Machine Learning and see how we can use it in order to generate business value. In order to successfully complete your assessment, AIcompany maintains the highest-quality standards for our training material. With help and guidance from industry leaders we created the AIcompany Machine Learning Guide that enables all students to obtain a profound theoretical understanding of the subject as well as recognise sustainable business value opportunities. The field of Artificial Intelligence is broad, complex, and filled with its own jargon and definitions. We have curated the most important topics for this guide and listed them below.

1.Theory Pages:

- History and Definition
- Machine Learning Definitions
- Data Pre-processing
- Eight different Machine Learning methods
- Deep Learning / Artificial Neural Networks
- API's
- Data Security & Ethics

2.Management Pages

- Organisation Readiness
- Hardware Requirements
- Agile Scrum
- AIcompany Talent Framework (ATF)
- AIcompany Pilot Management Framework (APMF)
- Further Readings

These topics have been carefully selected in order to learn about the most recent theoretical fields as well as gain practical knowledge and applicable skills to accomplish a true organisational transition.

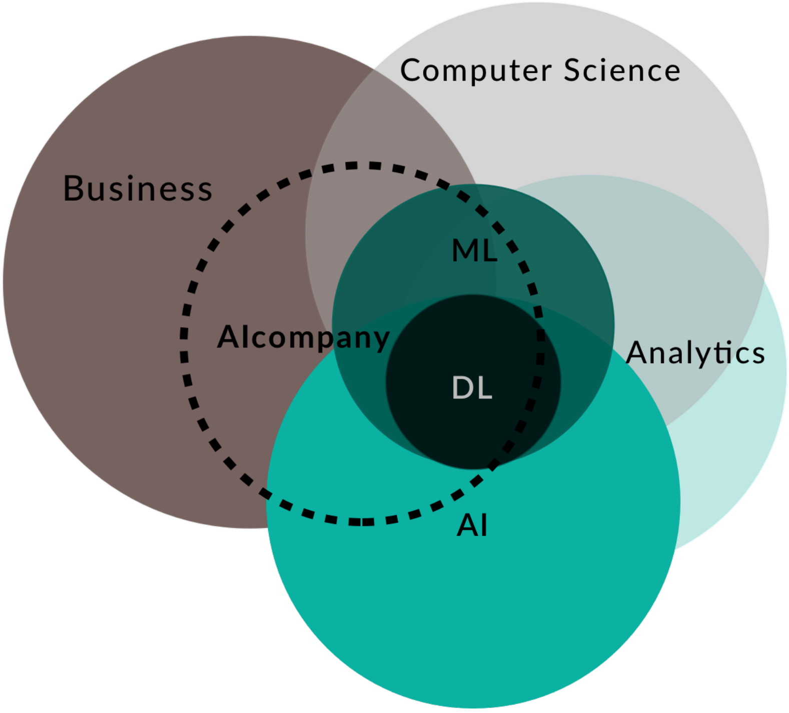
Theory Pages

History and Definition

Artificial Intelligence (AI) is usually defined as the science of making machines or computers accomplish tasks that require intelligence and reasoning. Scientists have been exploring the field of AI since the 1960's. Progress in the field was slow and challenging initially. Finding appropriate and useful tasks for AI has proved difficult due to its complexity. However, in the last decade, a significant foundation has been established for competing with human-level intelligence. In the next decade, AI will likely outperform human intelligence in many fields as the technology is adapted to modernised society.

So why the hype? The interest in machine learning is due to the same factors that have made Data mining and Bayesian analysis more popular than ever: growing volumes and varieties of available data, cheaper and more powerful computing processing, and the rise of affordable data storage.

With the rise of new computing technologies, machine learning today is far from machine learning of the past. It was born from pattern recognition in data sets and the theory that computers can learn, without being programmed, to perform specific tasks with the help of algorithms. An algorithm is a procedure or formula for solving a problem, based upon conducting a series of specified actions. In other words, the operator explicitly instructs what the computer needs to do. A computer program can be viewed as an elaborate algorithm. Algorithms are widely used throughout all areas of IT (information technology). Researchers became interested in AI because they wanted to program computers to learn from data. Machine learning is considered a field within AI. You will come across many different terms and definitions, with varying degrees of overlap. The image below demonstrates the topics and fields that will be covered in the learning materials and their relationship to each other:



Machine Learning is considered a field within Artificial Intelligence and Data (Advanced) Analytics that allows software applications to become more accurate in predicting outcomes without being explicitly programmed. Classifying precisely whether a solution is considered AI or Advanced Analytics can be difficult. We believe that whenever a computer is able to make autonomous decisions based upon predictions derived from machine learning models, it classifies as Artificial Intelligence instead of Advanced Analytics. Deep Learning (DL) is a subfield of machine learning and is responsible for the most exciting capabilities in diverse areas like natural language processing, image recognition and robotics.

Applying statistics

The mathematical science called statistics is what helps us to deal with this information overload. Statistics is the study of numerical information, called data. The statistical and iterative aspect of machine learning is important because as models are exposed to new data, they are able to independently adapt. They learn from previous computations to produce reliable, repeatable decisions and results. Currently this technology is applied to complex mathematical calculations on large amounts of data sets.
Both P-Value and Generalization are important statistical definitions in order for you to understand further machine learning concepts and how well they perform.

P-value

P-Value is defined informally as the probability of obtaining a result that is 'more extreme' than what was actually observed. In statistics, the p-value is the probability for any given statistical model that the null hypothesis is true (your model doesn't show any significant result). This implies that you want to observe a low p-value (normally <0.05). Understanding the p-value helps you to verify whether your machine learning model is accurate and consistent.

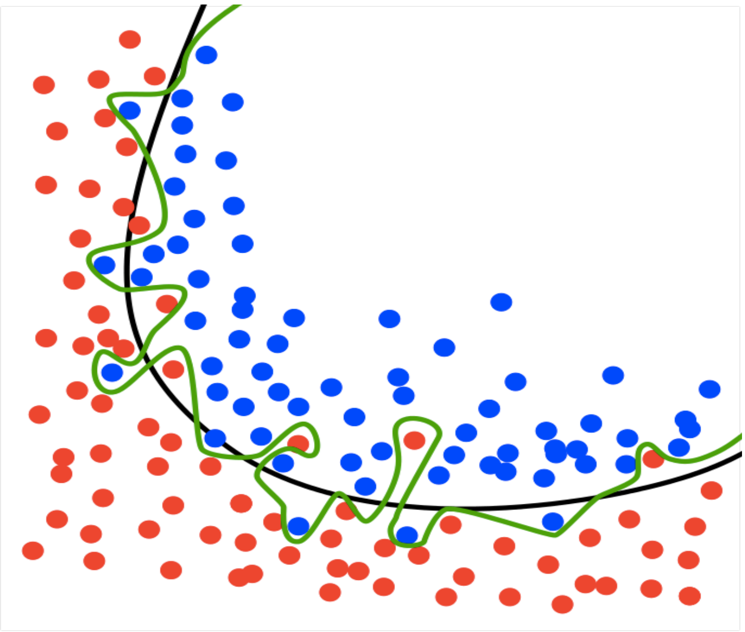
Generalisation

In machine learning, we describe the learning of the target function from training data as 'inductive learning'. Induction refers to learning general concepts from specific data sets. The opposite would be deduction, in which you seek to learn specific concepts pulled, or drawn out of data sets with the help of general rules. Generalisation refers to how well the concepts learned by a machine learning model apply to specific new data input that are considered new by the model (test sets). The goal of many machine learning business applications is to generalise successfully from the training data and apply the concepts to any new data to solve problems. The terminology "overfitting" and "underfitting" are used to describe how well the machine learning model learns and is able to generalise new data. We often encounter that overfitting is responsible for poor performance.

What is fitting? In machine learning a "fit" refers to how well you approximate a target function. The model's algorithm seeks to approximate the unknown underlying mapping function for the output variables (*y's*).

Overfitting

Overfitting occurs when a model learns too many specifics, details, and noise in the training data set to the extent that it negatively impacts the performance on new data. This means that any noise or random fluctuations in the training data is detected and learned as concepts by the model. In the image we provided a visual example:



The green line represents an overfitted model, and the black line represents a regularised model. Even though the green line best follows the training data, it is likely to have a higher error rate on new unseen data, compared to the black line.

Overfitting is more likely to happen when working with non-linear models since they are more flexible in learning a target function. Therefore it is recommended to include parameters or techniques to limit and constrain how much detail the model learns and removing some of the details it detects.

Underfitting

Underfitting refers to a model that is not able to generalise to new data. An underfit machine learning model is obviously not a suitable model and will have poor performance. Underfitting is often not discussed; it is easily detected since the generalised output it produces is nonsense.

The solution is to remove the model and apply new learning algorithms. Nevertheless, we included the description of underfitting because it provides contrast to what overfitting does.

Machine Learning Definitions

In order to understand how we can implement machine learning models to generate sustainable business value, it is important to gain some basic theoretical knowledge. In short, there are three types of machine learning flavours: Supervised Machine Learning, Unsupervised Machine Learning and Semi-Supervised Machine Learning.

Supervised Machine Learning

The majority of practical machine learning uses supervised learning. Supervised machine learning utilises an algorithm, consisting of input variables (*x*) and output variables (*y*), that allows the model to learn the 'mapping' from input to output. The goal is to train the model so well that when you provide a new *x* variable it is able to predict *y*. It is called supervised learning because the process of learning is similar to a teacher supervising the learning process of a student. The model knows the correct answer, whereas the algorithm iteratively makes predictions on the training data and is corrected by the teacher. Learning stops when the algorithm achieves an acceptable level of performance (Accuracy). In this guide, we will refer to *x* as a variable, however in machine learning lingo the correct name for a variable is "feature". With the help of several features, the teacher in the above-mentioned example is training his or her students to accomplished a specific performance or a specific target level of knowledge. In machine learning, the target is referred to as a "label". In many machine learning applications the model's label or goal is to predict some sort of specific outcome.

Supervised machine learning allows you to solve (*y*) problems. These learning problems can be further grouped into regression and classification problems.

- Classification: A classification problem is when the output variable is a category 0 or 1, such as 'black' or 'white' or 'buy' and 'No buy'.
- Regression: A regression problem is when the output variable can take on a range of values, such as 'Euro's' or 'Weight'

Many online businesses use classification and regression models in order to include recommendations and predictions.

Unsupervised Machine Learning

Unsupervised learning uses input data (*x*) but no corresponding output variables (*y*). In simple terms, you will not supervise the model by providing feedback if it is right or wrong. The goal of unsupervised learning is to allow the model to understand underlying structures and distributions in data sets in order to learn more about these data sets. Therefore, there is no correct answer and there is no teacher. Algorithms are 'left alone', in order to discover and present interesting structures in data sets. A simple example is a AI model used in gaming. The model is not explicitly instructed how to play the game. The model must determine, on its own, how to get to the next level. After many attempts a model will learn what works best.

Unsupervised learning is useful when you want to explore your data but don't yet have a specific goal or are not sure what information the data contains. It's also a good way to reduce the dimensions of your data.

Unsupervised learning problems can be further grouped into clustering and association problems.

- Clustering: A cluster problem allows you to investigate the reason why certain data groups are 'sticking' together. For example, it will discover grouping of customers by their purchasing behaviour. The model will be able to discover which features (*x's*) are causing these different groups.

- Association: An association rule learning problem allows to investigate certain rules that describe large portions of your data set. For example, customers that buy product A, also tend to buy product C in many cases.

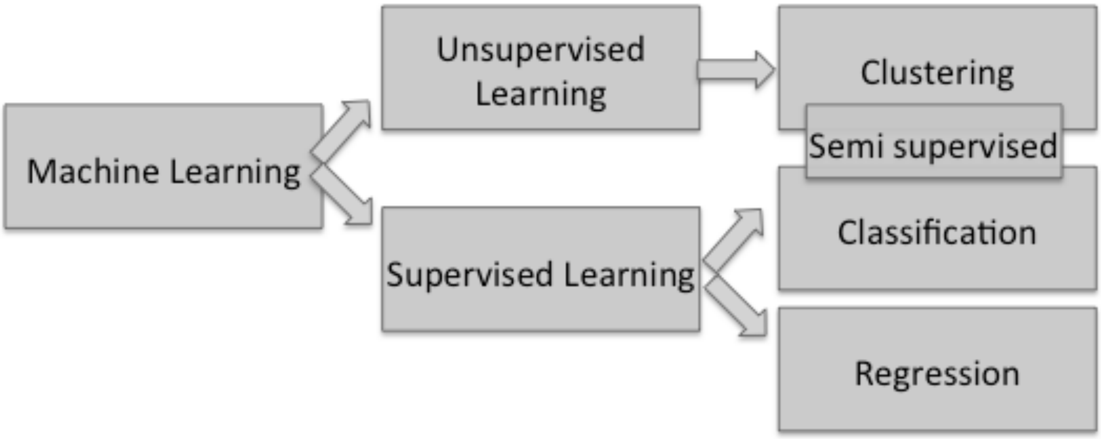
Semi-Supervised Machine Learning

From a more practical perspective, there is no need to decide which flavour to use in the early stages of your model build. It is common to switch or set up a supervised setting while adding several unsupervised modules. That's where semi-supervised Machine Learning comes in. Whenever you have a large amount of input data (*x*) and not all of your data is labelled and structured, you combine supervised and unsupervised learning techniques into a semi-supervised model. Lets expound the two types of data a semi-supervised model would encounter:

- Structured data. Structured data is information, usually text files, displayed in titled columns and rows, which can easily be ordered and processed by machine learning tools. In simple terms, it resembles perfectly organised spread sheets and data from machine sensors that is identified, labelled and easy to access. Most organisations are likely to be familiar with this form of data and already use it effectively.
- Unstructured data. Unstructured data doesn't fit nicely to data warehouse settings. Unstructured data is probably around 80% of all data. Examples include recorded human language, videos, images, and text documents.

A useful example of Semi-Supervised Machine Learning is a photo archive where only some of the images are labelled (eg. car, cat, person) and the majority are unlabeled. This requires a combination of both flavours, because it can be expensive or time-consuming to label data as it may require access to several domain experts. Unlabeled data is relatively cheap and easy to collect and store. We also see that in some cases a best-guess-predictions model for the unlabeled data is used to 'label' the data and allows you to feed that data back into the supervised learning model. This training model is able to make predictions on new data sets.

In chapter three we will describe eight machine learning methods. But before we dive into machine learning methods it is important to understand the way in which clustering, classification, and regression relate with respect to supervised & unsupervised learning.



Data Pre-processing

AIcompany believes it is important to have a basic understanding of what, and why data pre-processing is needed. Data pre-processing is not considered the sexiest part of machine learning but is important to understand. But don't worry, almost all of the necessary tools and algorithms for data pre-processing already exist and can be imported onto your data set.

Data pre-processing describes any type of processing performed on raw data sets in order to prepare it for further procedures (i.e. the use of machine learning). It transforms the data into a format that will be more easily and effectively processed for the purpose of the user.

To begin, your data set is split into training sets and test sets is done in order to verify whether the model is performing properly. If you use 100% of your data set to train the model, you will not be able to use that same data to test its accuracy (since it was already trained on those input features). Test sets allow you to examine if your model is working appropriately. AIcompany recommends an 80/20 ratio: 80% of the dataset available is used to train the model and 20% is used to test. Examples of data pre-processing methods are:

- ***Sampling***: which (randomly) selects a representative subset from a large population of data. To solve the specific problem, we might not need all the data. The goal is to obtain a representative sample of the data. In the real world, we see that sampling is typically used whenever it is too expensive and time consuming to process all of the data.
- ***Missing data***: deals with missing data sets. Missing values in a data set might influence the performance of a machine learning algorithm. A common solution to this problem is to use the mean, median, or mode (most frequent) value in this section. Deleting the entire observation or data group is often unnecessary because the remaining values can still be of great help to enhance the performance of your model.
- ***Transformation***: manipulates raw data to produce a single input.
- ***Dummy features***: allows you to distinct categories. In order for category features to be useful in regression analysis, all of the features need to be numerical. However, you might want to include an attribute or nominal scale feature such as 'Country', 'Type of Education' etc. Dummy features don't have a causal relation to the output, but they are created to 'trick' the regression algorithm into correctly analysing the features of interest. Say you have three countries; USA, The Netherlands, and Hong Kong. Labelling those countries to a number (1,2, and 3) would not mean anything since you can't subtract country 1 from country 3. In this case, you would add two additional columns, and each value could be either 1 or 0 (take for instance Hong Kong and USA). For example, specifying that the observation happened in the Netherlands would result in 0,0. (or in Hong Kong; 1,0).
- ***Normalisation***: organises data for more efficient access. Data is scaled to fall within a small, specified range. The goal of standardisation or normalisation is to make an entire set of values have a particular property. Lets look at income as an example. Suppose that the minimum and maximum values for the feature income are €98,000 and €128,000, respectively. If these numbers are too large compared to other data sets, it would automatically give more 'weight' to the higher income numbers. We would like to map income to the range 0.0-1.0 . By min-max normalisation, a value of €98000 for income is transformed to: 98,000 − 128,000 / 98,000 − 128,000 1.0 − 0.0 + 0 = 0.872
- ***Feature extraction***: identifies specified data that is significant in some particular context. Feature extraction involves reducing the amount of resources required to describe a large data set. When performing analysis on complex data, an often-encountered problem is that there are too many features involved.

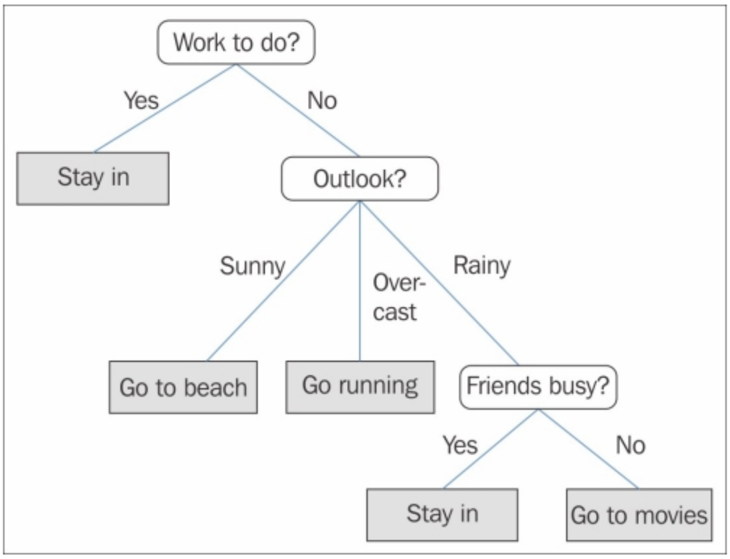
Analysis with a large number of features generally requires a large amount of memory and computational power. It may also cause a classification algorithm to be overfitting (described in the next section) to the model, generalising poorly new input data.

Machine Learning Methods

AIcompany selected 8 machine learning methods crucial in understanding how machine learning models work and create business value.

1. Decision Trees

A decision tree is considered a very simple, but sometimes, very effective decision support tool. Decision trees have a tree-like graph or model of decisions and their possible consequences, including chance-event outcomes, resource costs, and utility. From a business (decision) perspective, a decision tree is usually the minimum number of yes/no questions that one has to ask, to assess the probability of making a correct decision. As a method, it allows you to approach the problem in a structured and systematic way to arrive at a logical conclusion. See image:



Machine learning models that make use of a decision tree are able to identify various ways of splitting data sets into branch-like segments. These segments form an inverted decision tree that originates with a root node at the top of the tree.

In a decision tree each segment or branch is called a node. The bottom nodes of the decision tree are called leaves (or terminal nodes). For each leaf, the decision rule provides a unique path for data to enter the class that is defined as the leaf. All nodes, including the bottom leaf nodes, have mutually exclusive assignment rules; as a result, records or observations from the training data set can be found in one node only.

The object of analysis is reflected in this root node as a simple, one-dimensional display in the decision tree interface. In this simple display, the decision tree would most likely be a classification tree. But a decision tree can reflect both a continuous and categorical object of analysis (regression tree). The discovery of the decision rule is used to form the segments underneath the root node, a method that extracts the relationship between the object of analysis (that serves as the target field in the data). The values in the input field are used to estimate the likely value in the target field.

The target field is also called an outcome, response, or dependent variable (y). Once the relationship is extracted, one or more decision rules can be derived that describe the relationships between inputs and targets. Once the decision rules have been determined, it is possible to use the rules to predict new node values based on new or unseen data.

When to use classification, and when a regression tree?

This might seem like a difficult issue, but can something be relative easy to figure out. Classification trees, as the name implies are used to separate the dataset into classes belonging to the response variable. Usually the response variable has two classes: Yes or No (1 or 0). If the target variable has more than 2 categories, use a regression variant.

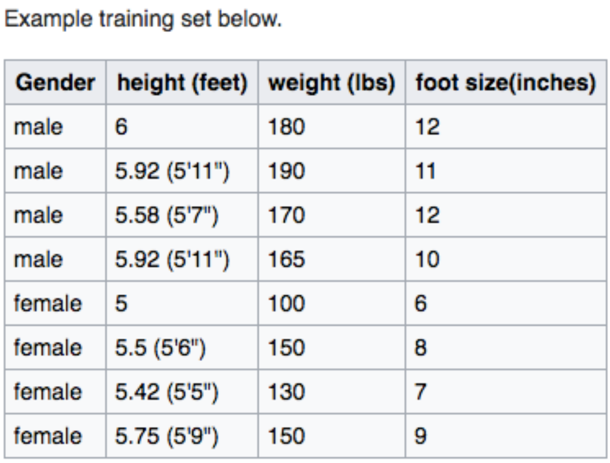
2. Naïve Bayes Classification.
Naïve Bayes classifiers are a family of simple probabilistic classifiers based upon applying Bayes theorem with strong independence assumptions between the features. In the below mentioned equation (not as difficult as it might look), the P(A|B) is posterior probability (The revised probability of an event occurring after taking into consideration new information), P(B|A) is likelihood, P(A) is class prior probability, and P(B) is predictor prior probability.

$$P(A|B)= \frac{P(B|A)\ P(A)}{P(B)}$$

Prior probabilities are based on previous 'experience'. Think of a data set where there is twice as many females as males, in this case it is reasonable to believe that a new observation (which hasn't been observed yet) is twice as likely to be female rather than male. In Bayesian analysis, this belief is known as the prior probability.

Applications
Divide them and it is useful for many real world classification problems.
- To mark an email as spam or not spam
- Classify news articles about technology, politics, or sports.
- Check whether a piece of text has expressing positive / negative emotions,
- Provides support in facial recognition

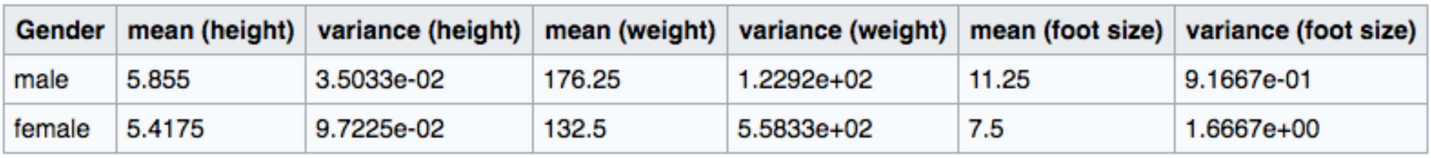
A simple but great example is given in the paper of Stuart Russell & Peter Norvig, AI, A Modern Approach, to explain how Naïve Bayes Classification works. The example that is given: classify whether a given person is a male of or a female based upon the measured features. The features include height, weight, and foot size.



Example training set below.

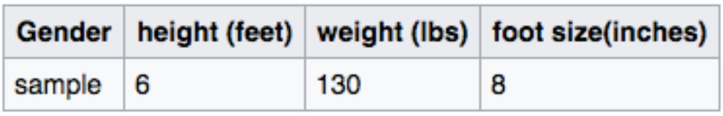
Gender	height (feet)	weight (lbs)	foot size(inches)
male	6	180	12
male	5.92 (5'11")	190	11
male	5.58 (5'7")	170	12
male	5.92 (5'11")	165	10
female	5	100	6
female	5.5 (5'6")	150	8
female	5.42 (5'5")	130	7
female	5.75 (5'9")	150	9

After categorising the data to either male or female the data looks like:



Gender	mean (height)	variance (height)	mean (weight)	variance (weight)	mean (foot size)	variance (foot size)
male	5.855	3.5033e-02	176.25	1.2292e+02	11.25	9.1667e-01
female	5.4175	9.7225e-02	132.5	5.5833e+02	7.5	1.6667e+00

Imagine the following data occurs as new input.

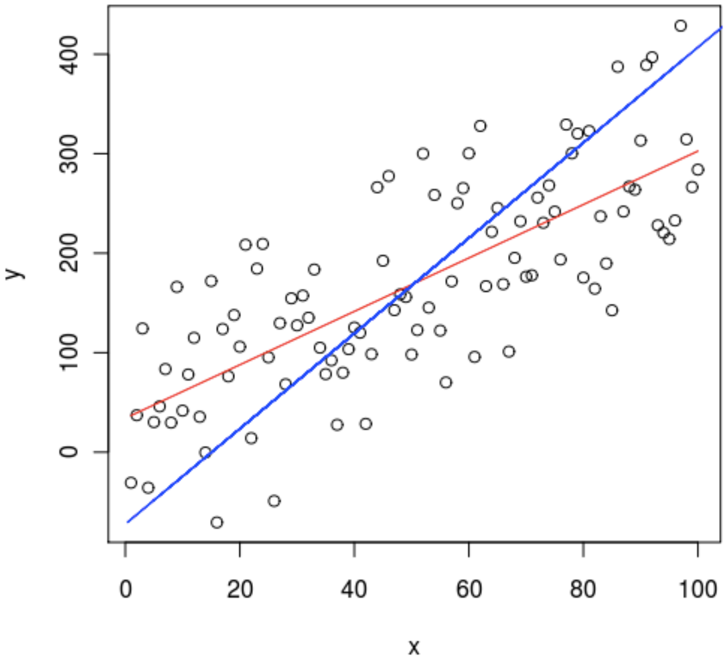


Gender	height (feet)	weight (lbs)	foot size(inches)
sample	6	130	8

Determining whether this input is considered male or female can be quite difficult for us. Using Naïve Bayes Classification in machine learning models could be highly accurate.

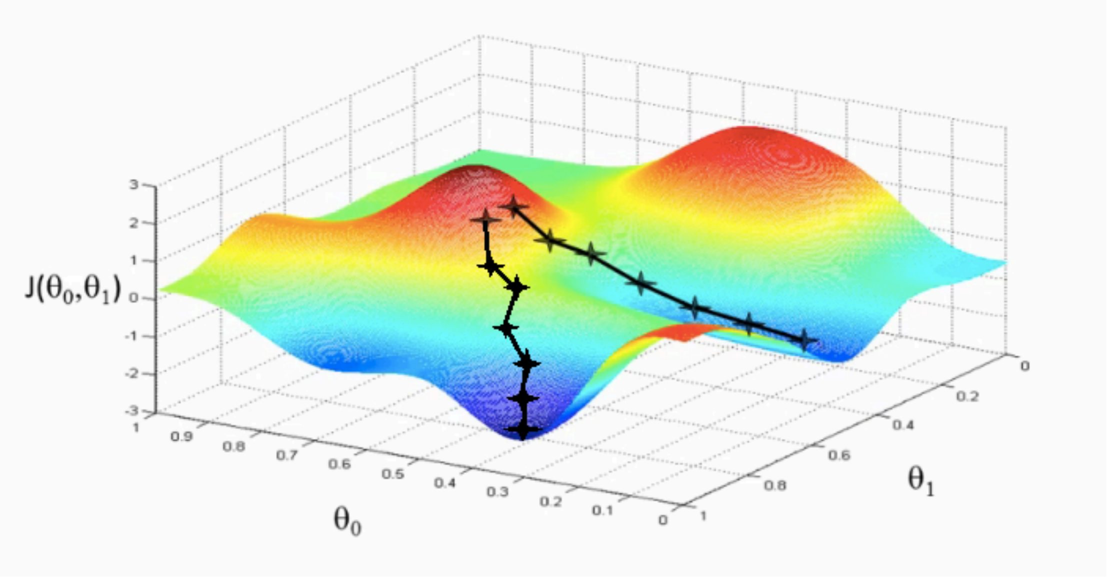
3. Ordinary Least Squares Regression

If you had any statistics classes in school, you probably heard of linear regressions before. 'Least squares' is a method for performing linear regressions. You can think of linear regression as the computer task of fitting a straight line through a set of data points. There are several possible strategies to do this, and "ordinary least squares" strategy is one of them. When visualising this method, the model will draw a line (like in the image below), and will (for each of the data points) measure the vertical distance between the point and the line, and add these values.



This method helps you to predict where new input data will stand relatively to existing data. The sum of the squared vertical distances between each data point in the set and the corresponding point on the regression line will tell you how great de model fits. The smaller the differences, the better the models fits the data.

4. Logistic Regression:
Logistic regression is considered a powerful statistical method for binomial outcomes with one or more explanatory variables. It measures the relationship between the categorical dependent variable and one or more independent variables by estimating probabilities using a logistic function, which is the cumulative logistic distribution.



How to recognise a question that could be answered with help of logistic regression models? In simple terms, look for a predictive analysis problem that could be answered with yes or no, and, has many possible causes (*x's*), such as: How does the probability of getting lung cancer (yes vs. no) change for every additional pound of overweight and for every pack of cigarettes smoked per day?

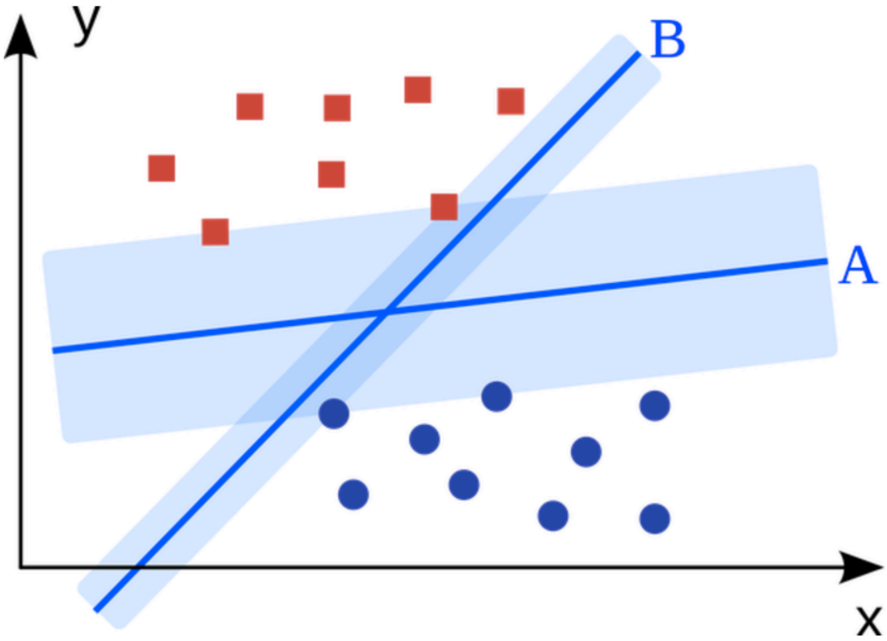
Applications
Logistic Regressions applications that are used in real world business solutions:

- Credit Scoring
- Measuring the success rates of marketing campaigns
- Predicting the revenues of a certain product
- Is there going to be an earthquake on a particular day?

5. Support Vector Machines:

Support Vector Machines are perhaps one of the most popular machine learning algorithms. Developed back in the 1990's, but still considerable relevant and high performing.

SVM is (binary) classification algorithm. Given a set of two different types of data points within a dimensional place, SVM generates a dimensional hyper plane to separate those sets into 2 groups. SVM will allow you to find straight lines which separates those points into 2 types and situated as 'far as possible' from all data points (for visualisation, see image below).



The distance between the line and the closest data points is referred to as the margin. The best or optimal line that can separate the two classes is the line that has the largest margin. This is called the Maximal-Margin hyperplane.

The margin is calculated as the perpendicular distance from the line to only the closest points. Only these points are relevant in defining the line and in the construction of your classifier. These points are called the support vectors (hence the name). They support or define the hyperplane. The hyperplane is learned from training data sets using an optimization algorithm that maximizes the margin.

Elaboration on Dimensional Space

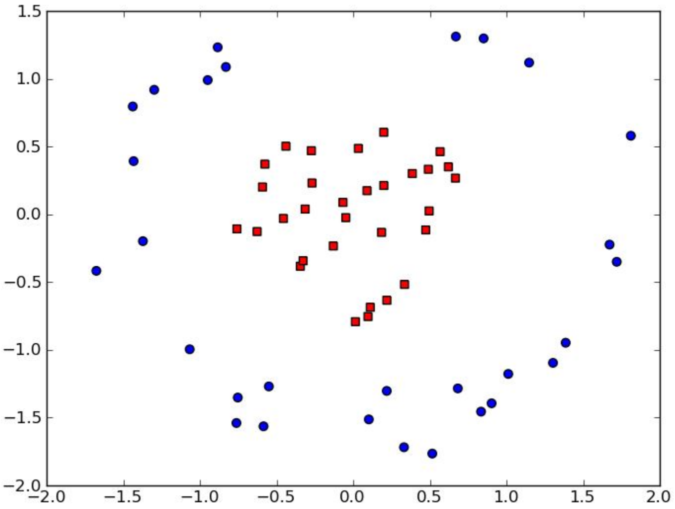
However, in practice we see that data sets are a bit messier and can't be separated perfectly with a hyperplane. A commonly used solution is to use 'soft margin classifier' or add more (up to infinity) dimensions. The 'soft margin classifier' approach allows some of the points in the training data to 'violate' the separating line and ensures that the model is not overfitting. For example, suppose you wanted to predict weight from height. You have all the data sets that correspond to people's weights and heights. In general, they follow a simple linear relationship. If you keep the data in the 2-dimensional space, and fit a line through it, it might not hit all the data points. But that is okay, because you know the relationship is linear, and you want a good approximation anyway.

What will happen if we add more dimensions in this example?

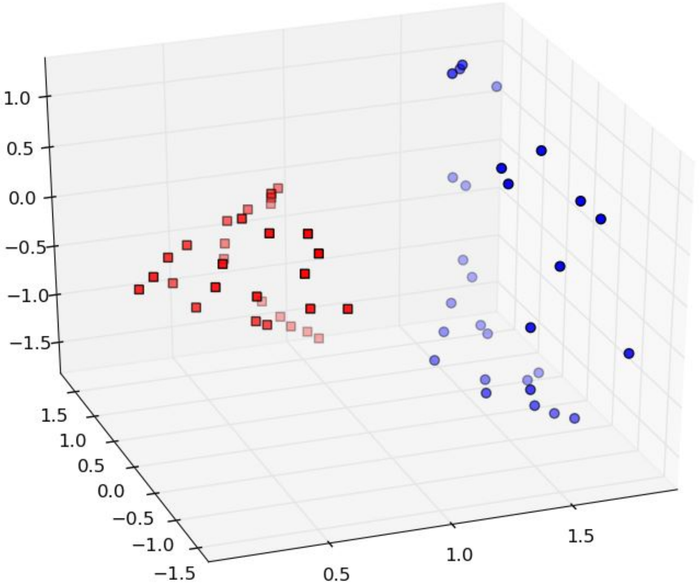
Now let's say that you took this 2-dimensional data and transformed it into higher dimensional space. So instead of only two, you also add 5 more dimensions. All these dimensions will (polynomial) try to 'best fits' the data points. You will end up with a line that is all over the place and you will have 'overfit' data. Because you forced the machine learning algorithm to take into account higher dimensions that have nothing to do with anything. Weights just depends (generally) on height linearly. It does not depend on the other 4 dimensions or higher order nonsense. This is why if you transform the data to higher order dimensions blindly, you run a very big risk of overfitting, and not generalizing data properly.

How does going to a higher dimension help classification effectively?

One or more of the additional dimensions may create distance between the classes that can be modelled with a linear function. The best way we can explain this is with help of the following visualized images:
The following data set is not separable in a two-dimension format since it would be impossible to draw a straight line between the red and blue dots.



However, these data points might be separable in three dimensions. A vertical plane between the dots would separates the colours in this case perfectly!



In practice we see that the main problem with transforming data to higher dimensions (up to infinite dimensions) is that it will cost a lot more processing and computing power! Working with big data and accurate SVM models will eventually lead to applications that are just not fast enough to be applicable to real world problems. However, that is why we have the Kernel Trick.

Kernel Trick

The word 'Kernel' is used in statistics and mathematics to denote a weighting function for a weighted integral. However, the Kernel Trick allows you to separate your data in a higher dimensional space without having to actually transform the data. This will result in less computation compared to actually transforming the data first. Understand that the Kernel is essentially a mapping function, one that transforms a given space into some other space. The Kernel is to define a (x) value in terms of original space of itself without even defining (or in face even knowing) what the transformation function is. Why is it even called a trick? For a kernel to be useful, it would typically need to have a large number of dimensions (in most applicable kernels even infinite). But transforming input data into such large dimensional space can take a lot of computation (hardware) power. Kernel allows you to work with high dimensions and yet it is easy to compute to similarity score in the original space. There are many different Kernel methods, such as Linear Kernel, Polynomial Kernel, and Radial Kernel, but in the AIcompany Essentials Guide we touch only the high level applications since we see that in practice that SVM algorithms are implemented using a kernel.

Applications

SVM applications that are used in real world business solutions:

- Display advertising
- Human splice site recognition
- Large-scale image classifications
- Image- based gender detection (however nowadays Deep Learning or Artificial Neural Networks are far more accurate, see chapter: ANN)

6. Ensemble Methods:
Ensemble methods are a combination of mentioned machine learning methods. In practice we see that Ensemble methods usually produce a more accurate solution than a single model would. If you want to relate this to real life, a group of people are likely to make better decisions compared to individuals, especially when group members come from diverse background.

Machine learning models that we use as inputs of Ensemble methods are called Base Models. Ensemble methods are learning algorithms that construct a set of classifiers and use a weighted vote of their predictions. The original ensemble method is called 'Bayesian Average', but more recent algorithms include error-correcting output coding, bagging, and boosting (basically a more complex way to apply Bayesian Average, but in essence they all have similar objectives).

So how do ensemble methods work? Why are they superior to individual models? And what are their benefits?

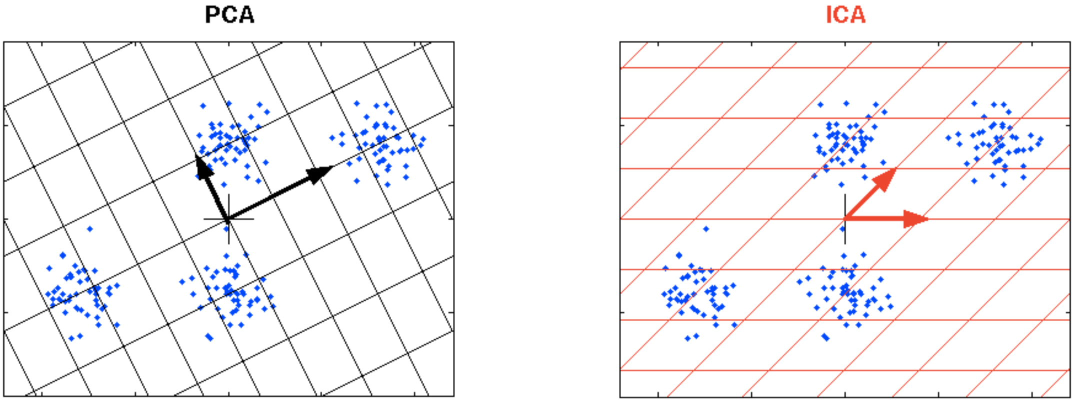
- They average out biases. For example, if you average a bunch of democratic-leaning polls and republican-leaning polls together, you will get an average something that isn't leaning either way (biased).
- They reduce the variance: The aggregate opinion of a bunch of models is less noisy than the single opinion of one of the models.

From a more business perspective, in finance, this is called diversification—a mixed portfolio of many stocks will be much less variable than just one of the stocks alone.

- They are unlikely to over-fit: If you have individual models that didn't over-fit, and you are combining the predictions from each base model in a simple way (average, weighted average, or logistic regression) there is less room for over-fitting.

7. Independent component analysis

Independent component analysis (ICA) is a statistical technique for 'revealing' hidden factors that underlie sets of random variables, measurements, or signals. ICA defines a generative model for the observed multivariate data, which is typically given as a large database of samples. Data variables are assumed to be a linear mixture of some unknown latent variables, and the system that combines the sets is also unknown. The latent variables are assumed non-Gaussian and mutually independent, and they are called independent components of the observed data. In the image we can see how new data will be automatically categorised and 'pushed' towards a group.



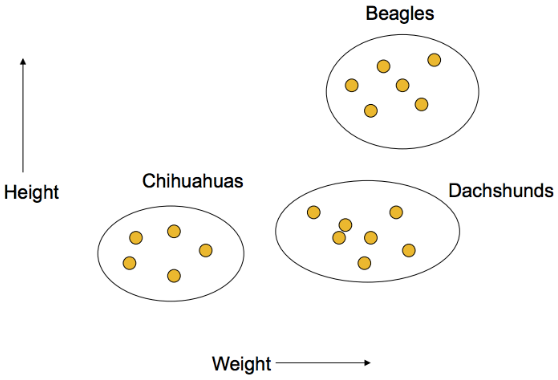
ICA is related to PCA, but it is a more capable technique for finding underlying factors if pervious mentioned methods fail or have poor output. It is commonly used for digital images, document databases, economic indicators and psychometric measurements.

8. Clustering Algorithms.

Clustering is the task of grouping data points into clusters so that:
- points within each cluster are similar to each other
- points from different clusters are dissimilar

Usually, points are in a high-dimensional space, and similarity is defined using a distance measure.

In the image below a clustering task is visualised and what it can do for different data points of dogs for example.



As mentioned, in cluster analysis, data is partitioned into groups based on some

measure of similarity or shared characteristic. Clusters are formed so that objects in the same cluster are very similar and objects in different clusters are very distinct.
Clustering algorithms fall into two broad groups:
- Hard clustering, where each data point belongs to only one cluster
- Soft clustering, where each data point can belong to more than one cluster

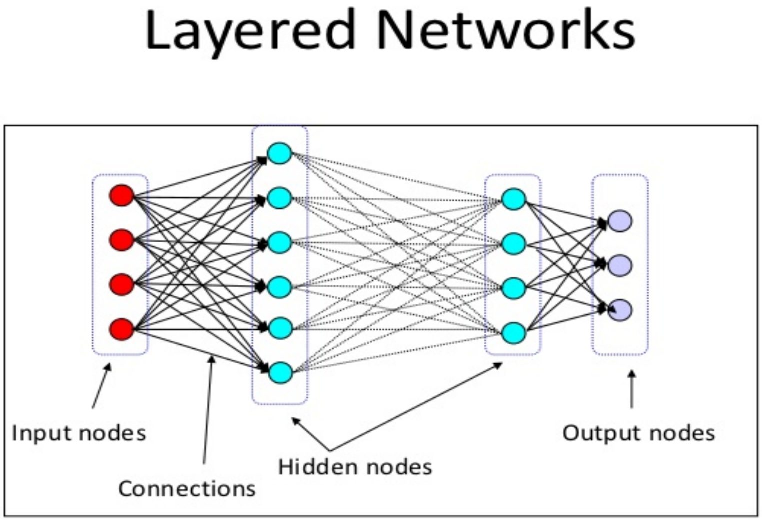
You can use hard or soft clustering techniques if you already know
the possible data groupings.

Every clustering algorithm is different. These six methods we often encounter:
- Centroid-based algorithms
- Connectivity-based algorithms
- Density-based algorithms
- Probabilistic
- Dimensionality reduction
- Neural networks / Deep Learning

Nowadays many of the considered disruptive AI solutions include Deep Learning or Artificial Neural Networks (ANN). That is why this guide will dedicate a special chapter to this specific method.

Deep Learning / Neural Networks

It is commonly known that Deep Learning is great for clustering and pattern recognition, but what is Deep Learning exactly? Deep Learning Neural Networks are inspired by our understanding of the biology of the human brain, more specifically, the trillions interconnections between neurons. Unlike a biological brain where neurons can connect to any other neuron within a certain physical distance, the artificial neural networks have discrete layers, connections and directions. Each neuron assigns a weighting to its inputs. Determining how correct or incorrect it is relative to the task being performed. The final output is determined by the total of those weightings.
The method of using artificial neural networks to process information is not considered new. The first sufficient working neural networks date back almost four decades ago. However, back then the computing power was not effectively able to handle great amounts of data and complexity to successfully implement artificial neural networks. This image visualises how complex neural networks work.



$$\begin{aligned} Output: \quad y_i &= f(w_i^1x_1 + w_i^2x_2 \quad w_i^3x_3 + \cdots + w_i^m x_m) \\ &= f(\sum_j w_i^j x_j) \end{aligned}$$

In order to get a better understanding of these layered Networks, we listed all key definitions in the process.

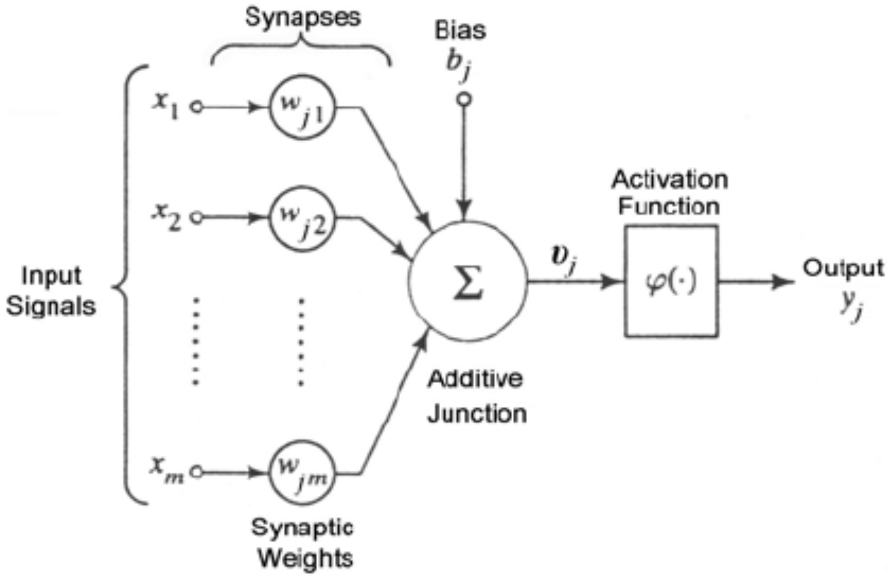
Layer

A layer is the highest-level building block in Deep Learning. The layer is the container that usually receives weighted input, transforms it with a set of mostly non-linear functions, and then computes these values as output to the next layer. A layer is usually uniform, and contains one type of Activation Function, making it comparable to other parts in the network. The first and last layers in a network are called Input and Output Layers, respectively, and all layers in between are called Hidden Layers. The output is, as mentioned, connected to the inputs of other neurons/units and therefore not visible as a network output (hence the term hidden layer).

The Neuron, The Unit.

The fact that the term 'artificial neuron' is often simply referred to as neuron, implies a close connection between AI and neurobiology, however Deep Learning has very little to do with the brain (for example, we see that biological neurons are more similar to entire multilayer perceptrons rather than a single unit in a neural network). Therefore this term is very misleading and potentially dangerous for the perception of the field of deep learning. Consequentially, it is encouraged to use the more descriptive term ‘Unit’ to define an artificial neuron.

Here we have visualised the setup of a simple unit, receiving multiple inputs:



Activation Function of a Neuron

So what does a neuron or unit do? Basically it assigns a (synaptic) weight to all the individual inputs (based on past experience) and calculates a ‘weighted sum’. Then it adds a bias (a statistical mediator), and then decides whether it should be ‘fired’ or not and outputs a transformation of the data. The Activation Function is the mechanism that makes this ‘fire’ decision. Units are more complex than natural neurons and can have multiple activation functions.

Unit Step Activation Function

As mentioned above, the Activation Function decides whether to ‘fire’ or not. But how is this decision made? The first thing that comes to mind is to have a binary threshold based Activation Function (relevant or not relevant). If the value is above a certain value, the function is activated. If it is less then the threshold, then it is not activated. However, in reality it can be much more complicated.

What will happen during a binary classifier is that more than one unit is activated. If all (or more than one) units are activated, there should be a selection made. Ideally, a binary classifier network would activate only one neuron, and not all the others. This approach makes it difficult for Deep Learning to train and converge. Consequently, it is more effective if the activation was not binary and instead allows for varying degrees of activation (for example in percentages, 70% activated). Therefore, if more than one unit activates, the ‘highest activation’ is identified and processed. There are a number of functions that can be utilised for this identification process such as: Linear Function, Sigmoid Function, Tanh Function, ReLu Function and so on. In practice, the Rectifier function is often used. The ML Essentials Guide has only touched this topic in order for you to get some understanding to the complexity in determining the most successful strategy for this step.

How do neural networks learn?

A neural network ‘learns’ by basically attempting to solve an optimisation problem to select a set of parameters that minimises an Error / Cost Function, which is the typical sum of squared errors. The cost function is an important concept in learning, as it is a measure of how far away the particular practical solution is from an optional solution to the problem that it is trying to solve. The goal here is to find weights in a certain space to find a function that has the smallest possible cost. The 'trial and error' procedure of computationally finding a slick result that produces minimal errors is called 'backpropagation'.

Backpropagation and Gradient descent

The goal of fitting the model is to find a suitable set of 'weights'. One way to do this is to choose a set of weights that minimises the sum of squared errors (see error arrow) from training data sets. The computational approach Gradient Descent is commonly accepted as the best method to complete backpropagation. Gradient descent takes steps in the direction of the greatest error decrease in parameters space, allowing it to quickly find a minimum Error Function. Linear Regression is not considered ideal for this task because of the shortcomings of the binary threshold functions as described in the Activation Function paragraph above.

Issues nowadays: deep / shallow (and wide) neural networks

With the development and increasing accessibility to Deep Learning, it is common for organisations to adopt shallow networks to save on budget and schedule, however there are a number of difficulties with using extremely wide, shallow networks. Deep refers to the amount of hidden layers (horizontal layers in the image above; layered Networks) and wide to the amount of different features (*x's*). The main issue is that very wide, shallow networks are very good at memorisation, but not effective at generalisation and in-field learning. Therefore, the ability of the network is limited by its training. The network will generate errors when real life inputs exceed the training knowledge developed. It is not efficient, and in some cases not possible, to train networks with every possible input value.

The advantage of multiple (deep) layers is that the network can learn features at various levels. For example, if you train a deep neural network to classify images, you will find that the first layer will train itself to recognise very basic things like edges, the next layer will train itself to recognise collections of edges such as shapes, the next layer will train itself to recognise collections of shapes like eyes or noses, and the last layer will learn even higher order features like faces. Multiple layers like these are much better at generalising because they learn all the intermediate features. Simply put, the likelihood that the network will be able to recognise a certain face correlates to the number of layers, since the network knows really well what a pair of eyes looks like.

In conclusion, all artificial neural networks need training to see hundreds of thousands, even millions of images, until the weightings of the neuron inputs are tuned so precisely that the network will identify the correct answer practically every time — clouds or no clouds, sun or rain.

Deep Learning has enabled many practical applications of Machine Learning and there are endless possibilities. A true pioneer in the overall field of AI.

API's

We chose to address the topic of Application Programming Interface (API's), since we see that often API's are used while implementing machine learning models in real world organisations. API's are integrated in order to have more access to data or more successful round-end running applications. For example, the Neural Network solutioning of IBM Watson uses many different API's to create output.

API's are not new. Whenever you use a smart phone, desktop or laptop computer, API's will be involved. In simple terms, API's are sets of requirements that govern how one application is able to talk to another, and what makes it possible to move information between programs. For example, an API is used whenever you log into an account with the help of Facebook or Google log-in. Services like Google Maps let other apps 'piggyback' on their offerings and software.

What are some of the benefits for implementing API's into your solution?
- Time: API's allow you to leverage existing services and free up time so you are able to focus on the problems at hand.
- Focus: Most of the existing API's are superior and far more comprehensive than what you can do by yourself. You can choose an API solution in order to avoid distraction and focus on core delivery.
- Cost: Third party services are not nearly as expensive as the cost of an engineer or developer. Not only is it cheaper but you are able to save long term cost associated with maintaining the technology and keeping up to date with the latest advancements.
- User Satisfaction: API's offer familiar features to your product, which can bridge the gap created by the use of an artificial model.

However, there are some drawbacks in including API's.
- Security: From a security point of view, there is a larger surface area for potential breaches. If one of the services becomes compromised, your own sensitive data may be at risk.
- Reliability: Using a third party exposes the program to the potential technical limitations of their product or service.

If a third party API seems expensive at first, AIcompany recommends you to do look at your specific situation and compare the costs. Take both the monetary and opportunity cost of building this service in-house in consideration (including future maintaining implications).

Data Security & Ethics

Starting a machine learning project in real world working environments can be a very sensitive topic. For example, think of private information about patients in the healthcare industry. Using their data could result in predictions or decisions made by Artificial Intelligence that may have a great impact on their lives. And potentially be life threatening. While most of these implicates are not unique to AI, some of them are. Predicting future behaviour based on previous patterns raises challenging questions. Companies in the financial sector already use machine learning to predict credit risk, for example. Governments run prisoner details through machine learning algorithms to predict the likelihood of recidivism when considering parole. In these cases, it is a technical challenge to ensure that factors such as race and sexual orientation are not being used to inform machine learning based decision-making. Even though such features are not directly provided to the algorithms, they may still correlate strongly with seemingly innocuous features such as zip codes. With careful design, testing, and deployment, machine learning algorithms may be able to make less biased decisions than a typical person. Anthropomorphic interfaces increasingly associated with machine learning raise novel privacy concerns. That is why AIcompany always insists on having anonymous data strategies with 5 overall scoring points.

1. Aggregate

When data is aggregated, it is unable to be used to identify the sources of the data. For example, if we provide the mean house income in combination with a street. The downside of this approach is that aggregating data too far will imply and lead to a machine learning model that will interpret data wrong and makes the model less accurate.

2. Remove

A very simple approach to insure privacy protection is to remove some values that were originally collected and presented in the data set. For example, omit gender, age, location or any other variable that has to remain confidential.

3. Top and Bottom Coding

Top and bottom coding means to replace extreme values of sensitive numerical variables with the weighted group mean for those values in order to mask outlying values, which are potentially identifiable. For example, if you do not want to give up specific revenue performance indicators of one particular division within your company, use the overall company revenue performance indicators.

4. Group

We can group multiple values together to protect individual's privacy. Consider that you can categorise an individual instead of linking their personal information directly at one observation.

5. Hash digests

Used often in the Healthcare industry. Using cryptographic hash of a string can make it impossible for someone to determine what the original string was, while being able to allow machine learning models (with third party API's) to play around with the data.

Management Pages

Welcome to the AIcompany Management Pages, congratulations you made it, now it is time to get down to business.
We previously taught you the essentials of machine learning and the next step is to make it applicable and relevant in a more professional business setting. We created several machine learning management frameworks combined with successful existing theories in order to equip you for real life projects and strategic management decisions. We will first help you to analyse and assess the current state of your organisation.

For that purpose, we created a free industry-specific readiness test at aicompany.co/business-value. Now we will explain every step and aspect of our readiness guide, by studying the underlying principles of the organisational AI pillars:

Pillar 1: Organisation structure (AMPF)
Pillar 2: IT infrastructure / Hardware requirements
Pillar 3: Data Strategy
Pillar 4: Talent (ATF)
Pillar 5: Business processes

Pillar 1: Organisation Structure

Both organisations and governments are starting to find many processes where knowledge-based tools can make a huge difference. When using machine learning to interact with online customers, keep in mind that piecemeal approaches will not work. Most organisations are deploying department-level solutions and standalone tools without sufficient funding. The results are typically inconsistent and haphazard, forcing time-consuming and costly fixes.
Even the perception of data value might differ from one department to another. One department may see the customer through a transactional lens, while another puts the emphasis on promotions. The differences in their data models will slow, and perhaps negate, the entire effort of applying machine learning. Starting any machine learning initiative must involve analysis and preparation, which takes into account each department. It requires formal nuanced governance structures, and senior management level facilitators and sponsors (see AIcompany Pilot Management Framework (APMF)).

Agile Scrum

Agile working environments are more likely to succeed.
In any machine learning initiative, an agile work environment is considered key. Within a scrum team we are dealing with various team members that have different backgrounds. Any traditional way of working encompasses different departments working separately and with their own, not aligned targets. This creates unnecessary distance between your employees resulting in loss a of efficiency. Agile working encourages teams to work together and jointly tackle impediments. Nowadays many organisations shift to agile working environments since their effectiveness has been proven time and time again. We are not demanding a complete and immediate organisational transition, but we do advise you to start embracing agile working for machine learning projects.

Pillar 2.1: IT infrastructure / Hardware Requirements

When deploying machine learning solutions in real world organisations, we see that having a Service or Container Oriented Architecture (SOA) can speed up the process. SOA has the ability to leverage existing services. One of the keys to SOA is that interactions occur between loosely coupled services that operate independently. SOA allows for service reuse, making it unnecessary to start from scratch when upgrades and other modifications are needed. This is a benefit to organisations that seek ways to save time and money. SOA is known to provide both time-to-market advantages, as well as business agility. The use of orchestration engines, or leveraging development environments that leverage services and SOA, allows machine learning projects to be more successful since the services might provide much of what the solution requires. These quick successes provide a time-to-market advantage.

Cloud infrastructure can also be very helpful for machine learning solutions since the system builds models from incoming transactional data. The overhead cost of integrating machine learning systems can be relatively high. But today, we have the option to bring most of our systems hardware to the Cloud. Amazon Web Services, for example, supports machine learning using AWS's algorithms to read native AWS data, such as RDS, Redshift, and S3 (all standards for Data Warehouse Services). This allows for computing services such as Machine Learning as a service by cloud infrastructure providers.

Pillar 2.2: Hardware Requirements

We at AIcompany found that some communication problems are caused by misconceptions about hardware requirements. For example, you might encounter some problems with managers (or Product Owners) if they assume that all big data machine learning solutions can be processed and computed on an iPhone. We want to discuss these misconceptions, which is why we incorporated the following hardware definitions.

The required memory and processing power for machine learning mainly depends on the tasks and architectural environment of the solution. When it comes down to hardware, you will probably hear a lot about GPU, CPU, and RAM.

CPU: The Central Processing Unit (CPU) is the computer's component that is responsible for interpreting and executing most of the commands from the computer's software. A modern CPU is usually small and square shaped, with many short, rounded, metallic connectors on its underside. What does the CPU do for machine learning? The CPU will do all; writing and reading variables in the code, executing instructions such as 'function calls' (explanation; see Deep Learning pages) creasing mini-batches from data, initiating transfers.

GPU: A graphics processing unit (GPU) is a computer chip that performs rapid mathematical calculations, primarily for the purpose of rendering images. AIcompany assumes that you will use a GPU for machine learning solutions, especially for deep learning. It would not be sensible to leave out the GPU when computing deep learning since it has great processing speed.

RAM: (Random Access Memory) is the physical hardware inside a computer that temporarily stores data, serving as the computer's "working" memory.

Additional RAM allows a computer to work with more information at the same time, which usually has a dramatic effect on total system performance. Having sufficient RAM is extremely important to run smooth machine learning solutions. The CPU memory clock and RAM are intertwined. The CPU memory clock determines the maximum clock rate of your RAM and overall memory bandwidth. However, usually the RAM computing

power determines the overall available bandwidth because it is on many devices slower than the CPU memory rate.
During the pilot phase you are able to experiment with any of the below mentioned specifications. In collaboration with the organisation's IT department, you are able to check whether you at least comply with;
- RAM (16GB)
- PCI SSD (512 GB)
- Quad Core Intel Core (i7 Skylake equivalent or higher)
- Graphics cards on a company laptop, GTX 980 (980Ms)
- Graphics cards on a desktop setup, 1080s or 1070s

Once you are ready and plan to include more data sets into your model you will require a lot more RAM and CPU power! To achieve this, it is a relatively cheap and quick solution to rent cloud infrastructure and only download the CSV predictions.

AIcompany created an overview for all commonly used backend systems and hardware solutions (Amazon, Google, e. Tensorflow / Theano) and corresponding languages (Phyton / C++).
We at AIcompany advise you to install a dashboard app (such as NIVDIA tooling), which allows you to browse through the performance results of implemented algorithms. You are able to measure running time, CPU utilisation, memory utilisation, and disk activity models up to 1,000,000 search results, and to apply models up to 10,000,000 search results, each with up to 50 fields, which is great for pilots and experiments.

Pillar 3. Data Governance

Real-world datasets can be messy, incomplete, and come in a variety of formats. You might just have simple numeric data. But sometimes you are combining several different data types, such as sensor signals, text, and streaming images from a camera. Research results from Harvard University (HBR, Data Strategy, May 2017) show that less than 1% of an organisation's unstructured data is analysed or used at all. More than 70% of employees have access to data they should not, and 80% of analyst's time is spent simply discovering and preparing data. Since Machine learning algorithms learn from data, we highly recommend having both 'Defensive' and 'Offense' data strategies. In simple terms, data defence is about minimising downside risk. For example, by ensuring compliance regulations, using analytics to detect and limit fraud, and setting up systems to prevent theft. Defensive efforts also preserve the integrity of data flowing through an organisation's internal system by identifying, standardising, and governing authoritative data sources. For instance, by having your customer and supplier information in a single source of truth. AIcompany believes that having Data Custodians might be a great way to start improving data quality. Data custodians are accountable for the technical control of data including scalability, configuration management, availability accuracy, consistency, audit trail, backup and restore, policies and business rule implementation. In practice we see that many problems concerning data sets are caused by the inconsistencies between individual annotators, especially if there are no clear data definitions. Data Custodians could have great impact since they are and most likely feel responsible for managing the use of data. This will increase data quality over time.

Pillar 4. Talent

Talent management has become a critical component in an organisation's ability to compete and succeed in the digital economy. Chief Information Officers (CIO's) must collaborate with Human Resource leaders in order to find, attract, and retain IT talent that is capable of keeping up. However, creating the right setting for a team to achieve their goals is most important. Implementing machine learning models that generate true business value requires the right skillset and talent. AIcompany came up with a talent framework for organisations.

AIcompany Talent Framework (ATF)

The team should consist of 4 up to 6 members. Our guideline; "A team should not have more members than you are able to share a pizza with". Having shared quite a number of pizzas in respectively effective teams, we kept this guideline. A common mistake is assuming that a highly technical stack is needed. We broke the requirements back to 4 capabilities.

Required Skillset:

- Domain Chief.

The domain chief has enough domain expertise in order to understand business behaviours and industry specific knowledge (Business Intelligence). The domains chief usually has frequent contact with the customers / clients. He or she does not necessarily cover technical domains but has to be able to use data to open up new product, business, and revenue opportunities.

- AI Specialist / Machine Learning Scientist.

The Machine Learning Scientist builds the model and algorithms. There is a difference between AI specialists and Data Scientists in terms of research capabilities. Across industries the job description 'Data Scientist' is used for all kinds tasks related to data. We believe this makes it harder for employers and clients to distinguish these specialist capabilities. We see AI specialists that are able to understand and adjust algorithms while taking the smallest details into account, whereas Data Scientists are usually limited to applying and training proven methods.

- Developer/Programmer.

The Developer is not only able to code in one or two languages, but is also able to fix interactions with existing services and current IT-landscapes. Think of creating integrated adaptors in order to connect machine learning solutions to the rest of the organisation's infrastructure. This also includes integrating third party's software packages (such as API's).

- Promoter Chief.

The Promoter Chief is responsible for creating a hospitable environment, knows the organisational culture, different departments, policies, guidelines, back-end systems, and funding streams. In simple terms, they ensure that all stakeholders are up and running (in an agile working environment this includes Scaled Agile Alignments). The Promoter Chief is preferably someone with a relative high position within the organisation.

Pillar 5. Business processes.

Search for procedures and decisions that are made frequently and consistently, such as approving or denying a loan. Another example are 'contact us' forms on websites, which got leaner in recent years. Machine learning already helped streamline those business processes. Instead of requiring users to select the topic of an issue, machine learning can interpret the content of a request and route it to the correct place.
However, we recommend businesses to start looking for suitable core processes. Make sure you are collecting data about how the decision was made. In the example of the loan approval collect what client data was important for approving the loan. Who made the decision, at what time, how confident did they feel taking the decision? This is the kind of data that can be used to fuel machine learning in the future. Please take a look at our ReadinessTest in order to find more information on how to analyse your core business processes.

AIcompany Pilot Management Framework (APMF)

In previous chapters we have talked about all major knowledge areas of machine learning, and now we will be talking about creating real business value with the help of starting a machine learning pilot.

AIcompany designed a framework to help managers get a quick-scan selection of all possible processes where Artificial Intelligence would have a great impact. Ideally, we look for projects and models that would be able to make a difference and deliver business value within 12 months or less (production). The AIcompany framework mainly focuses on the 'what' questions instead of the 'how', more technical aspects. Most Pilot frameworks and tools start from a company goal, vision and mission perspective; what does the organisation wants to accomplish. However, we believe that with the help of APMF, organisations are able to get highly innovative innovation initiatives started that could have a great, and often unexpected, impact on the entire organisation.

APMF

Step one: Client First?

We see organisations starting machine learning initiative because they see an opportunity to generate more revenue (Return on investment driven). The APMF requires you to start identifying from a customer satisfaction point of view. Not only will it strengthen the business case but will also help you find support within the organisation. Therefore, write down three (maximum) or less Client First themes where an increase of customer satisfaction (i.e. NPS scores) will have great impact (Core Business). These themes should be described however your customers /clients perceive the full service / product. For example, if you sell mortgages, please remember that customers are not looking for mortgages, most likely, customers would much rather get a house without the burden of a mortgage. A question that might help you to come u with customer satisfaction themes; 'Where would our customer really appreciate a more proactive approach?'

Step two: What must be done manually and repeats itself often?

Answering this question would in most organisations result in a long list of activities and processes. Please try to group and label them, keep it simple (measured in time, hours for example, %).
As mentioned, search for procedures and decisions that are made frequently and consistently. We recommend businesses to start looking for suitable core processes. Try

to collect as much data as possible about how the decision was made. In the example of the loan approval, what customer data was important for approving the loan? (Who made it? At what time of day? How confident did they feel in the decision?).

Step three: What do we actually know?
Very important, identify what data is available and is currently stored. Similar data sets might have been stored in different formatted labels or back office systems. Are processes or activities labelled? In what quantities? Is the data structured or unstructured? Are there any intermediate results stored, or linked overall results? What kind of job title or person made the decision? Again, how confident did they feel in this decision? All these answers will help you determine what kind of model to use and how to make it accurate.

Step four: Design your model
Step four is considered the most time consuming step. Here it comes down to the creativity and skillset of the AI specialist or Data Scientist you are working with in order to deliver value. What could an entire dedicated team of the best employees do for this one special client? What data would be required to design an appropriate learning model that will have a high accuracy performance? Design a model that is able to predict, handle, advise and support this delivery. Remember, finding the perfect fit requires time and many Trial-and-error's. The pitfall we see often encounter in real world organisations is a project team that started out with a highly complex model to solve a problem that can be easily simplified and solved by a relative simple model. Unsupervised machine learning might be the end goal (agents), but try to combine methods. For example, while doing market research you might want to segment consumer groups to target specific website behaviours, a clustering algorithm will be sufficient. Apply clustering techniques to derive smaller number of features and use those features as inputs for training a classifier model. Image 4 shows a machine learning workflow we often see in real world organisations.

Step Five: Select your dreamteam
(See AIcompany Talent Framework for details) Please do not hesitate or be shy in creating the pilot dreamteam. In Large organisations we noticed that even a one-hour commitment from the right domain specialist could make a huge difference. Selecting your team is also a great way to create ambassadors across several departments and divisions.

Step Six: Pilot Time
The pilot should demonstrate what a successful implementation of a machine learning model would look like. Deployment in a production system is recommended. How accurate is the machine learning model performing to new real world data? During the pilot there should be more than enough data and feedback from end-users available that allows you to fine-tune the business case, and rethink return on investments for further scaling.
Good luck! Please do not hesitate to contact us if you would like to have more information or want to see practical cases (in your industry) where APMF designs were implemented.