Free and Open Machine Learning Documentation

Release 0.1

Maikel

Nov 20, 2019

Contents:

CHAPTER 1

Abstract

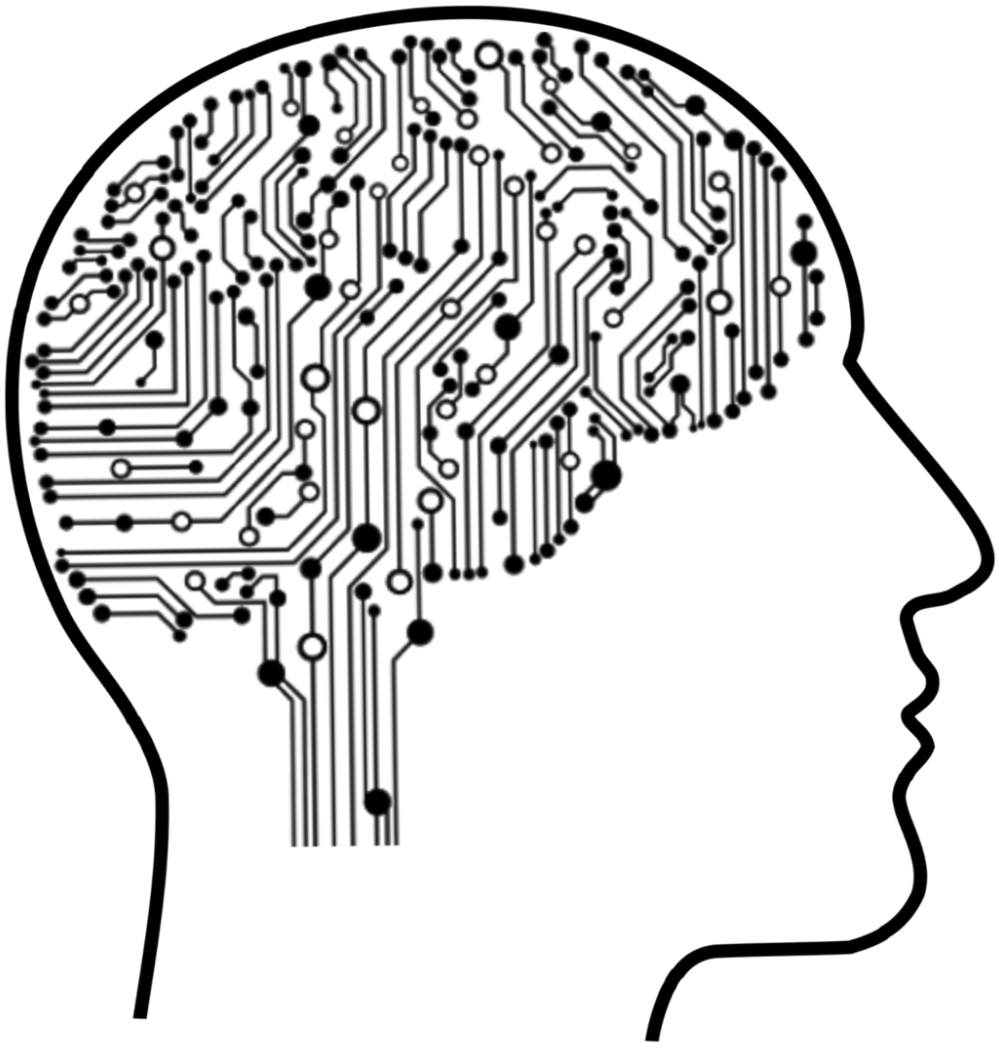
This book is all about applying open machine learning solutions for real practical use cases. So the core focus is on outlining how you can apply open machine learning in a simple way. Books with lots of mathematical background information on how machine learning works are available for more than 70 years.

Machine learning is an exciting powerful technology. The continuous use and growth of machine learning technology opens new opportunities. It should also give improvements for everyone. But this is our, so also your, responsibility! Machine learning technology should be available for everyone. So without barriers. But with the rise of new machine learning enabled products it is important to be able to ask critical questions on how safety, security, privacy and ethical issues are handled.

This book is created to give you a head start to use and apply new Open Source machine learning technologies to solve your business problems. Important machine learning concepts are explained, but the main emphasis is on providing insights in the possibilities that are now available within the growing open source machine learning ecosystem. This so you can start applying machine learning in your business today. And be able to judge answers regarding important quality aspects, like safety, security and privacy.

This book gives an overview of all important OSS machine learning frameworks and OSS support tools that you can use for prototyping with machine learning or when using machine learning for real production use cases.

Warning: This document is in alfa-stage!! Collaboration is fun, so *Help Us* by contributing ! There are some chapters currently written and editing work (typos,spelling) is yet to be done! Some more background information of the project can be found in the readme on github.com. And do not forget to join the ROI movement!



CHAPTER 2

Table of Contents

2.1 Preface

We humans are since the beginning of the development of modern computers obsessed with creating computers that have super powers. Even before the birth of computers research has been done on artificial intelligence (AI). The question what artificial intelligence really is, is hard and fuel for philosophical discussions.

Currently (as of writing 2019) we see more and more products developed that claim to have super powers that come close to AI. A look under the surface shows that the real progress on AI is made by a tangible technique, called machine learning.

Machine learning today is able of solving challenging problems that impact everyone around the world. Problems that were impossible to solve for long or where too expensive or too complex to solve. Now solving these problems is possible using this new machine learning technology. Currently very complex problems and meaningful problems are solved using applications based on machine learning algorithms. Many firms involved are willing to tell and show you how easy it is! But you must be aware: machine learning is a buzzword in the industry! So the ML field is full of companies that use fads, all kind of vendor lock-in options and marketing buzz to take your money without delivering long running solutions.

This publication is aimed to give you solid information so you can start applying the new machine learning tools and frameworks too. However with no strings attached. So the focus for this publication is on openness. The core focus is outlining concepts and showing an open architecture that make machine learning possible for real business use cases. And of course this publication is focused on outlining open source solutions that make it possible to start your machine learning journey. So the aim of this book is to be a practical grounding in open machine learning and its business applications. This to help you transform your organization into an innovative, efficient, and sustainable company of the future using new open machine learning technology.

Machine learning is and should not be the exclusive domain commercial companies, data scientists, mathematics, computer scientists or hackers. Our belief is that every business and everyone should be able to take advantage of the machine learning techniques and applications available. This is possible within the field of machine learning as we will show in this publication.

Nowadays knowledge is more and more openly shared, thanks to open access, open publication licenses and open source software. So everyone can and should benefit from the possibilities that open machine learning frameworks and tools provide.

To create this publication a lot of papers, books and reports on machine learning have been examined. And doing crucial 'hands-on' to experiences and feel the power of machine learning algorithms turned out to be crucial for understanding and creating this publication.

In the journey on learning how to apply machine learning for real business cases many books turned out to be either too theoretical, or to much focused on programming algorithms only. As an IT architect I missed the overall machine learning picture from an typical architecture point of view. So business, information, application, infrastructure, security and privacy perspective. This books fills up that gap.

This publication is not an end, but is constructed as an continuous effort to provide usable open and non commercial information for applying machine learning technology.

This publication was only possible with the help of you! If you have a suggestion or corretion, please send an email to info [at] bm-support [dot] org. Ik will add you to the contributor list, unless you ask to be omitted.

2.2 Introduction

With the use of Machine learning you can solve challenging problems that impact everyone around the world. Machine Learning (ML) and Artificial Intelligence (AI) are rapidly emerging technologies that have the potential to change our world with speed that humankind has never experienced before. Machine Learning and Artificial Intelligence are not the same, although the current technologies developed for ML do help research and developments on AI. ML can be characterized with a stricter definition from an engineering perspective. Trying to define AI raises more philosophical discussions on what intelligence is. This publication is focused on machine learning. But beware that the terms machine learning and artificial intelligence are intertwined and many so called AI applications are in fact driven by machine learning technology.

However mind the buzz and fads surrounding AI and ML: Machine Learning, deep learning and a lot of tools developed are not "a universal solvent' for solving all problems on the road to perfect AI. No perfect or magic machine learning tool or method exist (yet) that can solve all your complex problems.Machine learning is just a solution for a **certain type** of problems. In future the use of machine learning tools and can be applied to a broader landscape of problems. But do not try to solve all your problems with one (new)technology.

Artificial Intelligence and Machine Learning are now in the forefront of global discourse, garnering increased attention from practitioners, industry leaders, policymakers, and the general public.

2.2.1 What is covered in this book?

Nowadays many people are talking about the transformative power of machine learning and how it will revolutionize the economy, but what does that mean for your business and how do you get started? How do get solid independent advice to learn how machine learning and can improve or disrupt your business? This book gives you a quick introduction to get started with machine learning.

The field of machine learning and artificial intelligence is making rapid progress. Do you know what kind of applications for direct business use are already possible today? Are you aware of the very low entry barriers to take advantage of machine learning? Is your knowledge of free and open source solutions available in the machine learning eco system up to date? How do you classify safety, security and privacy risk when using machine learning? These and many other questions are the foundation of this book.

With new machine learning applications and companies being created on a daily basis, it is difficult to get a hold of what applications are viable, and which are hype, fads or simple a hoax. Especially when the terms ML and AI are intertwined. This book directs you to tangible working open source machine learning software. This category machine learning software is used at large for real business use cases. And because it is OSS it can be studied and improved.

Given that machine learning tools and techniques are increasingly part of our everyday lives, it is critical for professionals in the IT industry to gain knowledge on machine learning and start asking questions yourself. What will you

be doing with machine learning tools and applications the coming 3 years? Are you really aware of the safety and privacy concerns evolving that are part of this technology? Do you really understand and control the working?

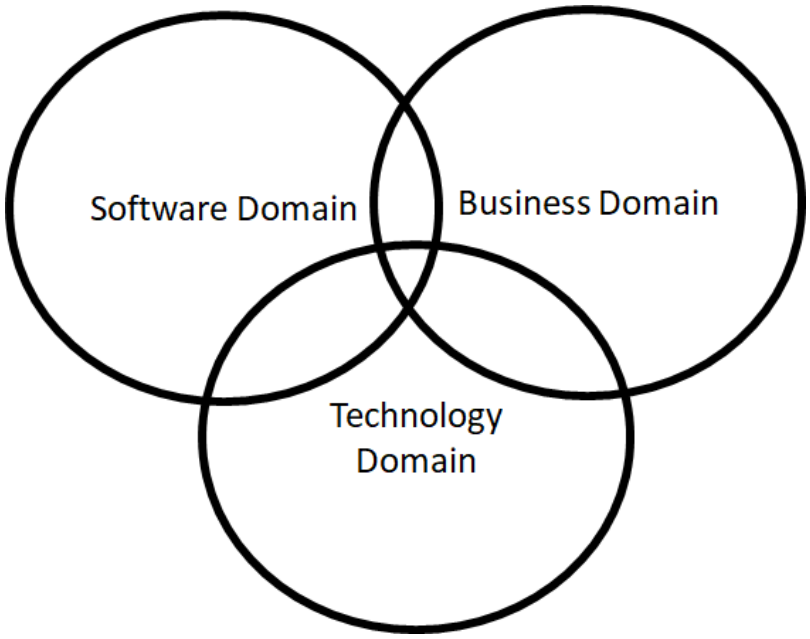
This book is all about taking advantage of the new OSS machine learning technologies for your business. The major machine learning concepts are explained, but the main emphasis of this book is giving insights in the various possibilities that are available within the open source machine learning ecosystem. This so you can start applying machine learning in your business today, without unclear dependencies towards a vendor.

This book gives an overview of all important OSS machine learning frameworks and OSS machine learning support tools that you can use for prototyping or to use for real production systems.

This book will not explain and dive into statistics and basic math or the advanced mathematical algebra functions that form the foundation under machine learning algorithms. If you are interested in learning the mathematical foundations on which machine learning is developed you can find good starting material in the reference section of this book.

This book aims to cover the high level machine learning concepts and gives you information to get started to work with machine learning for your business use case.

So this book is concentrated on machine learning aspects where software, business and technology touch each other.



(* When we write Open Source Software or OSS in this report we explicitly mean FOSS as defined by the Free Software Foundation - FSF.org)

2.2.2 Who should read this book?

This book is created for IT professionals who wants to learn of machine learning without being already forced into a specific solution. So I you like architecture, concepts to create your own solution, than this publication if for you.

This book is primary written with IT managers, directors, business owners, system engineers, quality managers and IT architects in mind.

This book crucial outlines concepts, but will not go into too much mathematical or technical details. However after reading this book you should have a more complete and realistic overview of the possibilities applying machine learning (ML) or artificial intelligence (AI) for your use cases.

2.2.3 Why another book on Machine Learning?

There are many books and courses developed to learn you what machine learning is. Most of these books and courses are focused on hands on learning and require you to program. However not many books and resources are focused on explaining the concepts with a clear focus on real business use cases.

Despite the enormous buzz and attention for machine learning currently it is proven to be hard to apply machine learning for real profitable use cases. Applying machine learning starts with a broad overview of the concepts, the architecture, the technology components and pitfalls that are present.

2.2.4 Is Machine Learning complex?

You might get the impression when visiting presentations from commercial vendors that machine learning is simple. The hard work is already done and all you have to do is get your credit card and make use of the incredible machine learning cloud offering. This machine learning as a service (MaaS) will take your company to the next level and the advise of the sales consultant is clear: Using their MaaS service is so simple that entering your credit card number is probably the hardest part. Maybe it will take a minute, maybe more. But you will find out that things are maybe not that simple after all. And you are right. The great offerings of many large and small vendors selling MaaS from a fantastic cloud offering will not solve your business problem in a simple way. As with all new technologies and especially IT technology: There are over promises on advantages and getting the return on your investments is not that simple. You will be confronted with complex terminology, a machine learning back-box from your vendor that is of course great at billing, data collection and data cleaning problems you had never heard of, and security, privacy and even safety issues. And if you think it can not get worse also legal and ethical issues will slow your project down. By using an 100% open approach (tools, methods) for machine learning a lot of risks can be mitigated. E.g. it is easier to control spending in the important ramp up phase of your project. If needed for production and scalability you can always move calculation to a cloud platform in a later stage.

There have been tremendous advances made in making machine learning more accessible over the past few years. This book outlines some great OSS applications ready to be used, even if you really hate difficult mathematical formulas. Multiple developments are in progress that now really make it possible to drop your data and let a complex ML algorithm do the hard work.

But don't be fooled. Machine learning remains a relatively 'hard' problem. Solving soft problems with machine learning requires far more than a good computer scientist alone. Using ML for soft problems requires a variety of disciples and creativity, experimentation and tenacity.

2.2.5 Organization of this book

The topics explored in this book include: Chapter 'tbd ' outlines why openness and OSS is so important for machine learning. Chapter 'tbd ' dives into the basic concept and terms that come with machine learning.

Todo: Complete this when all chapters are clear and ready!

2.2.6 Errata, updates and support

We have made serious efforts to create a first readable version of this book. However if you notice typos, spelling and grammar errors please notify us so we can improve this book. Since the world of machine learning is rapidly evolving some parts of this book will needs updates to present to you the latest machine learning solution building blocks. That's why there is also an on-line version of this book available that will incorporate the latest updates.

If like to contribute to make this book better: Please CONTRIBUTE! See [chapter contribution]

If you need support for your business use case and need some guidance with your pilot or project using machine learning: Please see our sponsor list [chapter consultancy]

2.3 Why focus on open source?

Open Source is an approach for the design, development, and distribution of new products & knowledge offering practical accessibility to its source. Real open source solutions have a license that is approved by the FSF.org or the OSI foundation. Open source is all about collaboration. Collaboration is key for developing, applying and using machine learning functionality.

Open Source Software(OSS) is the norm for machine learning. However using open source software will still be new and innovative for a lot of companies. However if you want really want to benefit from using machine learning software you must go for a solid OSS machine learning ecosystem. This makes you flexible, independent and you can still use thousands of consultancy firms and hosting companies that can help you.

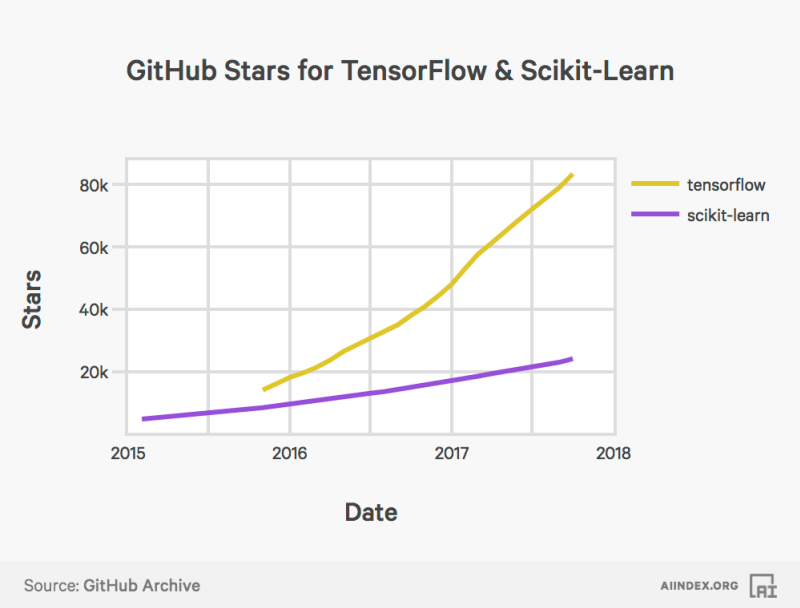
A transition towards OSS can already be very hard and can be disruptive. And applying machine learning for real business cases is also already complex and challenging. But using machine learning without taken direct benefit from the OSS ecosystems that come with is like learning to swim without hitting the water. So hit the water as soon as possible, after a while you will see the benefits.

Machine learning applications are expensive to develop and to adopt. This accounts for the development process itself, but even more for the needed infrastructure and resources to develop meaningful applications for your business. This means that currently big firms like Google, IBM, Microsoft, Facebook and Amazon are at the front of the queue and smaller counterparts get left behind. Since most of the scientific knowledge is freely available and more and more infrastructure needed is available within the open source domain, this book is entirely focussed on open source. The technique behind ML is too much fun and often requires adjustments and tweaking, which is hard when you are using black-box solutions.

OSS developments in the machine learning field are no hobby projects. Almost all major OSS machine learning developments are backed by small or large companies(e.g. Google, Microsoft, Facebook, Uber) active in the deep learning ecosystem. Small machine learning OSS projects are often developed by researchers from backed by a strong foundation or by universities.

A focus on OSS for applying machine learning for real is crucial. OSS machine learning applications and frameworks have the following benefits:

- Create solutions software faster, better and with less friction. You can adjust what you want without limitations.
- Lower cost for creating your first pilot project. Mind: Your first attempts will fail. And the faster your pilot projects fail, the better. This since applying the new machine learning capabilities requires some learning curve. Technical, but even more on the organization and business side.
- Flexibility and changeability.
- No vendor lock ins. Of course the ML cloud offerings of the major tech companies are great (Azure ML, IBM Watson, Amazon, Google etc). But playing around without any strings attached and limitations set for you gives you a head start.



Almost all companies advertise with machine learning powered software products nowadays. This also means that all existing software that is already been sold for decades is now suddenly re-branded with the new machine learning buzz words. Like cognitive, artificial intelligence (AI) powered and data driven. You can easily be fooled since massive marketing effort (time, money, material) has been invested to sell the old buggy solutions as new innovative machine learning powered solutions. In reality black box solutions from small or large vendors are almost always based on fads. This is why you should be very suspicious when using cloud based machine offerings without an option to DIY. So if the new solution looks to good to be true, be aware. When using machine learning OSS solutions you can inspect yourself the working or ask someone to audit the software you directly know what the promise is based on. Because in the end: The security, safety and privacy of your customers are at risk.

2.4 What is machine learning

To understand the basic principles of machine learning you do not need a master or PhD in computer science or another complex mathematical or technological study. Machine learning should be beneficial for everyone, it is important that everyone is able to understand the basics and the underlying principles.

This section outlines the most used terms used within the machine learning field. If you are short on time and want to know what the machine learning buzz is all about: This is the section you should read!

Before introducing the terms and definitions: Be aware that no unified de-facto definition of machine learning exist. So when you are new in this area be aware that when people are writing and talking about 'machine learning' they can be talking about totally different subjects. Since investments in machine learning by commercial companies are still growing, a lot of documentation freely available on machine learning is biased. In the reference section of this book we collect open access resources, but also open access is not free from commercial interest and not always objective and unbiased. So be aware of facts and fads when reading machine learning papers and books. And yes to determine fads in texts you can use machine learning as well.

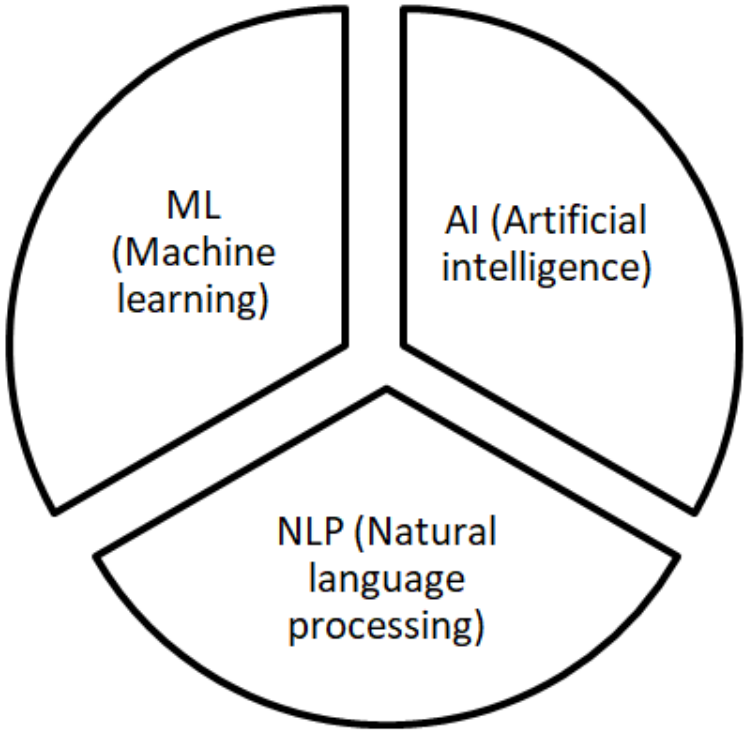
This section outlines the essential concept of surrounding machine learning more in depth.

2.4.1 ML, AI and NLP: What is what

Machine Learning (ML) and Artificial Intelligence (AI) are terms that are crucial to learn about when creating machine learning driven solutions. But also the term NLP (Natural language processing) is a term that is crucial for

understanding machine learning. A lets start with a high level separation of these terms and their meaning:

- AI (Artificial intelligence) is concerned with solving tasks that are easy for humans but hard for computers.
- ML (Machine learning) is the science of getting computers to act without being explicitly programmed. Machine learning (ML) is basically a learning through doing. Often ML is regarded as a subset of AI.
- NLP (Natural language processing) is the part of machine learning that has to do with language (usually written). NLP will be outlined more in depth in another chapter of this book.



A clear distinguishing between AI and ML is hard to make. Discussions on making a clear distinguishing are often a waste of time and heavily biased. For this publication we use the term ML, since machine learning can be brought down to tangible hard mathematical algebra and software implementations. Philosophical discussions on questions 'what is intelligence?' are mostly related to AI discussions.

At its core, machine learning is simply a way of achieving AI. Machine learning can be seen as currently the only viable approach to building AI systems that can operate in complicated real-world environments.

A few other definitions of artificial intelligence:

- A branch of computer science dealing with the simulation of intelligent behaviour in computers.
- The capability of a machine to imitate intelligent human behaviour.
- A computer system able to perform tasks that normally require human intelligence, such as visual perception, speech recognition, decision-making, and translation between languages.

There are a lot of ways to simulate human intelligence, and some methods are more intelligent than others. AI raises questions on the philosophical spectrum, like 'What is intelligence?', 'How do we measure intelligence?' and gives a lot of fuel to good ethical discussions that arise like:

- Should AI driven machine be a legal entity?

- How do we prevent AI machines to kill human life, since AI machines will be 'smarter' than human intelligence ever will be.

These ethical should not be neglected. In the section ['ML in Business'] a more deep dive in the ethical issues for applying machine learning is given.

Machine Learning is the most used current application of AI based around the idea that we should really just be able to give machines access to data and let them learn for themselves.

2.4.2 Statistics is not machine learning

Statistics is not machine learning. So let repeat this one more time:Statistics is not machine learning. But when the truth this that statistics and machine learning are intertwined and can not be separated. So for a good understanding basic knowledge of statistics is important. The question 'What's the difference between Machine Learning and Statistics?' is a questions that occurs often and leads to heavy discussion among scientist. To get is straight: A very clear separation between machine learning and statistics is hard to make. Machine Learning is however more a hybrid field than statistics. Some answers on this question are:

- Machine learning is essentially a form of applied statistics
- Machine learning is glorified statistics
- Machine learning is statistics scaled up to big data
- Machine learning improves a model by learning using data, where a statistical model is not automatically improved feeding it more data.
- Statistics emphasizes inference, whereas machine learning emphasized prediction.

Of course all answers are a bit true. With Machine Learning insights improves based using on more data. Using pure statistical models, learning and improving is not automatically guaranteed. So summarized statistical and machine learning methods and reasoning do have a large overlap, but the the purpose of using statistics or machine learning is often very different.

Machine Learning can be defined as:

- Machine learning is a field of computer science that uses statistical techniques to give computer systems the ability to "learn" (i.e., progressively improve performance on a specific task) with data, without being explicitly programmed. (source Wikipedia)

The underlying algorithms used for machine learning are essentially based around statistics. Machine learning is similar to the concepts around data mining. An algorithm attempts to find patterns in data to classify, predict, or uncover meaningful trends. Machine learning is only useful if enough data is available, and if the data has been prepared correctly. So despite the promises of machine learning, when you want to apply machine learning you always will have a data challenge. Not only for getting enough quality data, but also to manage the retrieved data. And most of the time storage and performance are the easiest problems to solve regarding data.

For machine learning, four things are needed:

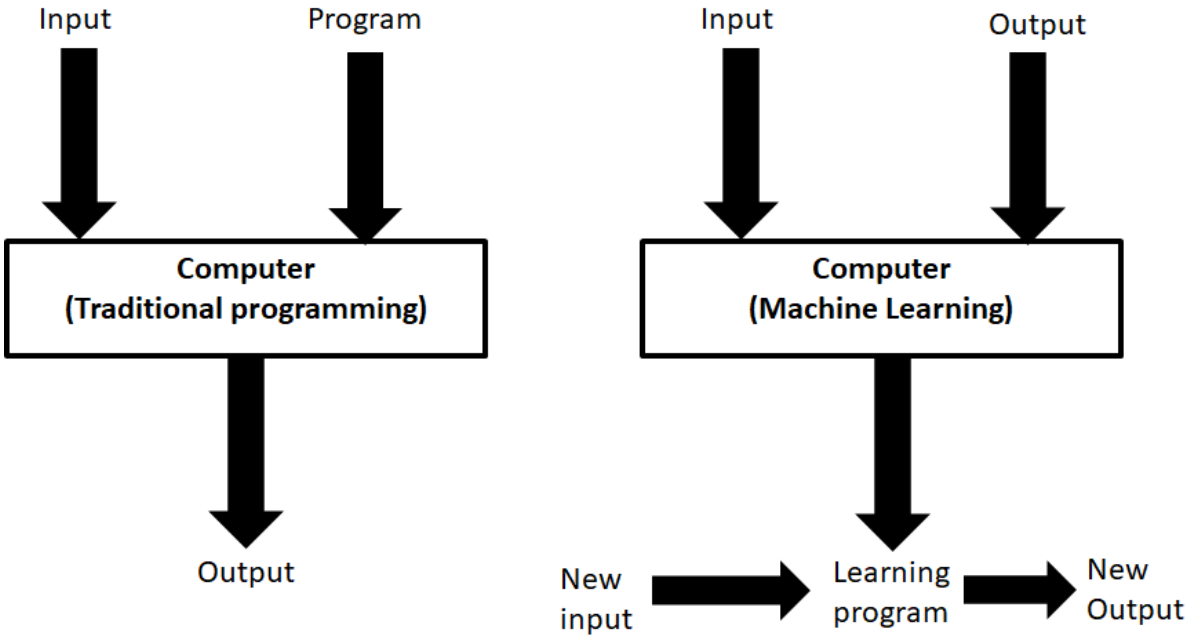
1. Data. More is better.
2. A model of how to transform the data.
3. A loss function to measure how good the model is performing.
4. An algorithm to tweak the model parameters such that the loss function is minimized

2.4.3 The paradigm shift: Creating smart software

To really understand machine learning a new view on how software can be created and works is needed. Most of our current computer programs are coded by using requirements, logic and design principles for creating good software. E.g. When you add an item to your shopping cart, you trigger an application component to store an entry in a shopping cart database table. However many real problems, solutions aren't so easy. A good solution requires knowledge of the context and a lot of hard to point knowledge built from your experience.

Determining the exact context of a car in traffic and in order to make a decision within milliseconds to go left or right is very hard programming challenge. It will take you decades and you will never do it right. This is why a paradigm shift in creating software for the next phase of automation is needed.

Programming computers the traditional way made it possible to put a man on the moon. To break new barriers in automation in our daily lives and science requires new ways of thinking about creating intelligent software. Machine learning is a new way to 'program' computers. When a programming challenge is too large to solve with traditional programming methods (requirements, getting input, etc) a program for a computer should be 'generated'. Generated based on some known desired output types. But knowing all desired output types in front is impossible. So your new 'program' will get it wrong sometimes. Large amounts of input data will increase the quality of the generated prediction model. In the old traditional paradigm called 'the program'.



Difference between general programming and (supervised) machine learning.

In essence machine learning makes computers learn the same way people learn: Through experience. And just as with humans algorithms exist that makes it possible to make use of learned experience of other computers to make your machine learning application faster and better.

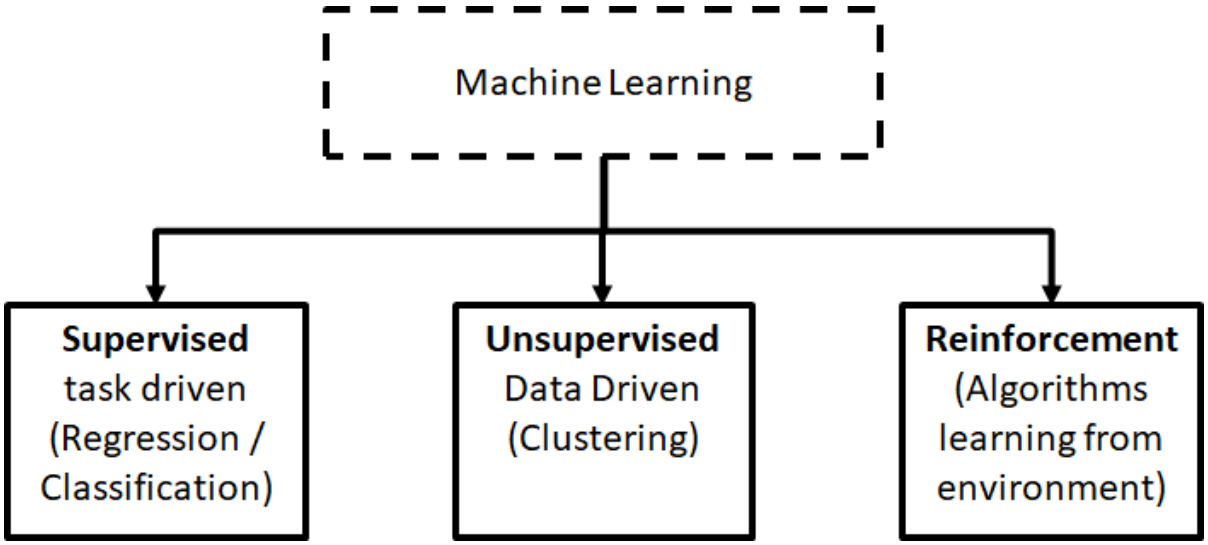
2.4.4 Overview machine learning methods

Whenever you will be confronted with machine learning it is good to known that different methods, and thus approaches, exist.

At the highest level, machine learning can be categorized into two main types:

- Supervised learning and

- Unsupervised learning.



Supervised Learning

Supervised Learning: Most practical solutions use supervised learning. Supervised learning encompasses approaches to satisfy the need to classify things into categories, known as classification. It also includes approaches to address the need to provide variable real-value solutions such as weight or height known as regression.

Unsupervised Learning

The goal of this type of learning is to model data and uncover trends that are not obvious in its original state. This type of learning is used to learn about data. Unsupervised learning methods are suited for unlabeled data. It is used is to find patterns where the patterns are still unknown. Unsupervised learning seems attractive since it does not require a lot of hard work of data cleaning before starting. However there are also serious challenges when applying unsupervised learning.

To name a few:

- Without a possibility to tell the machine learning algorithm what you want (like in classification), it is difficult to judge the quality of the results.
- You have to select a lot of good examples from each class while you are training the classifier. If you consider classification of big data that can be a real challenge.
- Training needs a lot of computation time, so do the classification.
- Unsupervised learning is more subjective than supervised learning, as there is no clear goal set for the analysis, such as prediction of a response.
- The order of the data can have an impact on the final results.
- Rescaling your datasets can completely change results.

In machine learning no single algorithm works best for every problem, and it's especially relevant for supervised learning (i.e. predictive modelling).

Reinforcement learning (RL)

Reinforcement Learning is close to human learning. Reinforcement learning differs from standard supervised learning in that correct input/output pairs are never presented, nor sub-optimal actions explicitly corrected. Instead the focus is on performance. Reinforcement learning can be seen as learning best actions based on reward or punishment.

Reinforcement learning (RL) is learning by interacting with an environment. An RL agent learns from the consequences of its actions, rather than from being explicitly taught and it selects its actions on basis of its past experiences (exploitation) and also by new choices (exploration), which is essentially trial and error learning.

In reinforcement learning (RL) there's no answer key, but your reinforcement learning agent still has to decide how to act to perform its task. In the absence of existing training data, the agent learns from experience. It collects the training examples ("this action was good, that action was bad") through trial-and-error as it attempts its task, with the goal of maximizing long-term reward.

RL methods are employed to address the following typical problems:

- The Prediction Problem and
- the Control Problem.

Supervised learning

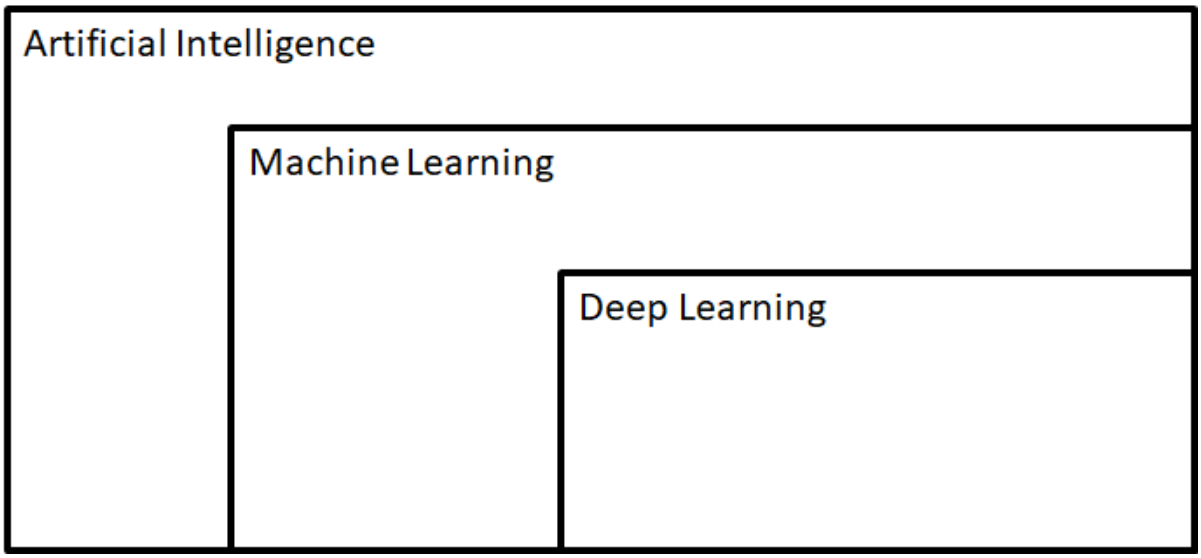
Supervised learning addresses the task of predicting targets given input data.

Deep learning (DL)

Deep Learning(DL) is an approach to machine learning which drives the current hype wave of self riding cars and more.

Deep Learning (DL) is a type of machine learning that enables computer systems to improve with experience and data. Deep learning is a subfield of machine learning.

To position Deep Learning in the spectrum of AI and ML see the next figure.



AutoML

Every technology will be evolving continuously. So when you have mastered machine learning you will be faced with yet another machine learning innovation. The big next thing beyond machine learning is automated machine learning in short autoML.

AutoML can be defined as: the automated process of algorithm selection, hyperparameter tuning, iterative modelling, and model assessment. AutoML will accelerate the model building process, the time consuming 'human' part within ML.

So with the current machine learning we have:

> Solution = ML expertise + data + computation

With AutoML the challenge is to turn this into:

> Solution = data + 100X computation

2.4.5 Other common terms used in the ML world

Within the world of ML you will read and hear about concepts and terms as networks, deep learning, reinforcement learning and more. Many of these terms are derived from years of scientific progress and discussions.

Data science

Data science can be defined as:

- The practice of, and methods for, reporting and decision making based on data.

So Data science is a umbrella term for several disciplines (technical and non technical) that deal with data. Even storing data in a retrievable way is a real science with many pitfalls.

Generative model

A Generative model can be defined as:

- A model for generating all values for a phenomenon, both those that can be observed in the world and "target" variables that can only be computed from those observed

Neural networks (NNs)

Neural networks (NNs) can be defined as:

- The algorithms in machine learning are implemented by using the structure of neural networks. These neural networks model the data using artificial neurons. Neural networks thus mimic the functioning of the brain.

The 'thinking' or processing that a brain carries out is the result of these neural networks in action. A brain's neural networks continuously change and update themselves in many ways, including modifications to the amount of weighting applied between neurons. This happens as a direct result of learning and experience.

NN are can be regarded as statistical models directly inspired by, and partially modelled on biological neural networks. They are capable of modelling and processing non-linear relationships between inputs and outputs in parallel. The related algorithms are part of the broader field of machine learning, and can be used in many applications.

Features (also called attributes): Properties of an data object to train a ML system. Think of features as number of colours in your street,the number of leafs on a tree, or the size of a garden. A smart selection of features is crucial to train a ml system.

Vision

A lot of machine learning application work on vision. But vision for computers is different than vision for humans. Humans can not see without thinking. And when we see something our mind is concepts playing with us.

Vision for computers can be defined as:

- The ability of computers to "see" by recognizing what is in a picture or video.

Speech

One of the great things we can do with computers to create applications that transfer words to speech or when we need a lot of data transfer speech to data. Great progress has been made on automatically analysing conversations without human intervention needed.

Speech:

- the ability of computers to listen by understanding the words that people say and to transcribe them into text.

Language

Understanding each other is hard. But this is typical a field where machine learning applications, mainly NLP driven have made great progress using (new)machine learning techniques and technologies.

A definition of language as used within the ML field:

- The ability of computers to comprehend the meaning of the words, taking into account the many nuances and complexities of language (such as slang and idiomatic expressions).

Knowledge

Defining knowledge is hard, but crucial for many machine learning applications. An attempt to define knowledge in the context of ML:

Knowledge:

- The ability of a computer to reason by understanding the relationship between people, things, places, events and context.

2.5 ML Reference Architecture

When you are going to apply machine learning for your business for real you should develop a solid architecture. A good architecture covers all crucial concerns like business concerns, data concerns, security and privacy concerns. And of course a good architecture should address technical concerns in order to minimize the risk of instant project failure.

Unfortunately it is still not a common practice for many companies to share architectures as open access documents. So most architectures you will find are more solution architectures published by commercial vendors.

Architecture is a minefield. And creating a good architecture for new innovative machine learning systems and applications is an unpaved road. Architecture is not by definition high level and sometimes relevant details are of the utmost importance. But getting details of the inner working on the implementation level of machine learning algorithms can be very hard. So a reference architecture on machine learning should help you in several ways.

Unfortunately there is no de-facto single machine learning reference architecture. Architecture organizations and standardization organizations are never the front runners with new technology. So there are not yet many mature

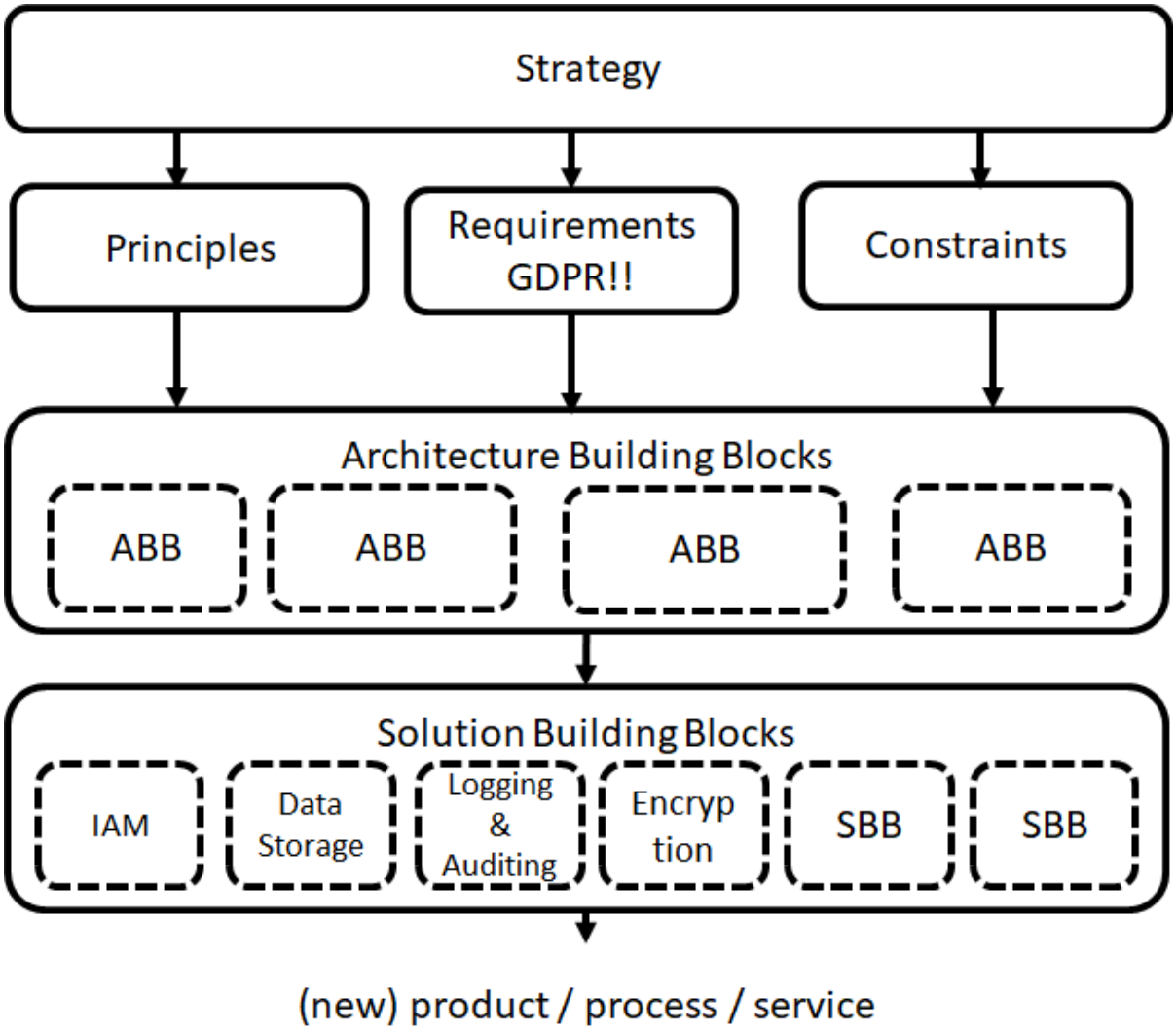
machine learning reference architectures that you can use. You can find vendor specific architecture blueprints, but these architecture mostly lack specific architecture areas as business processes needed and data architecture needed. Also the specific vendor architecture blueprints tend to steer you into a vendor specific solution. What is of course not always the most flexible and best fit for your business use case in the long run.

In this section we will describe a (first) version of an open reference architecture for machine learning. Of course this reference architecture is an open architecture, so open for improvements and discussions. So all input is welcome to make it better! See section *Help*.

The scope and aim of this open reference architecture for machine learning is to enable you to create better and faster solution architectures and designs for your new machine learning driven systems and applications.

You should also be aware of the important difference between:

- Architecture building Blocks and
- Solution building blocks



This reference architecture for machine learning describes architecture building blocks. So you could use this reference architecture and ask vendors for input on for delivering the needed solution building blocks. However in another section of this book we have collected numerous great FOSS solution building blocks so you can create an open architecture and implement it with FOSS solution building blocks only.

Before describing the various machine learning architecture building blocks we briefly describe the machine learning process. This because in order to setup a solid reference architecture high level process steps are crucial to describe the most needed architecture needs.

Applying machine learning for any practical use case requires beside a good knowledge of machine learning principles and technology also a strong and deep knowledge of business and IT architecture and design aspects.

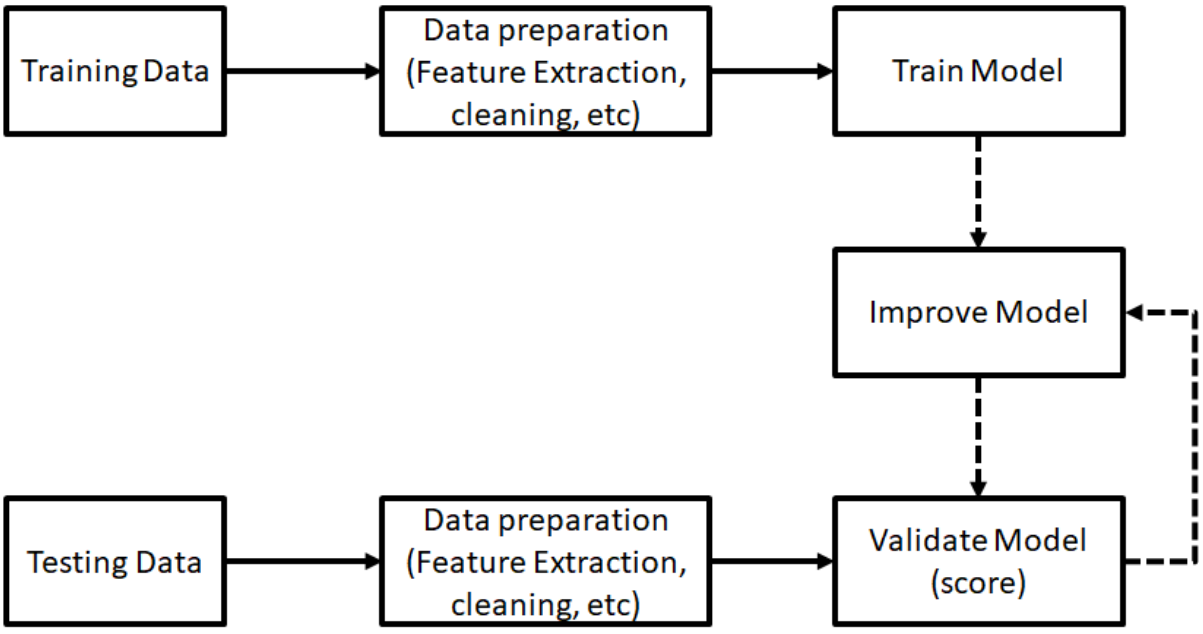
2.5.1 The machine learning process

Setting up an architecture for machine learning systems and applications requires a good insight in the various processes that play a crucial role. So to develop a good architecture you should have a solid insight in:

- The business process in which your machine learning system or application is used.
- The way humans interact or act (or not) with the machine learning system.
- The development and maintenance process needed for the machine learning system.
- Crucial quality aspects, e.g. security, privacy and safety aspects.

In its core a machine learning process exist of a number of typical steps. These steps are:

- Determine the problem you want to solve using machine learning technology
- Search and collect training data for your machine learning development process.
- Select a machine learning model
- Prepare the collected data to train the machine learning model
- Test your machine learning system using test data
- Validate and improve the machine learning model. Most of the time you will need to search for more training data within this iterative loop.



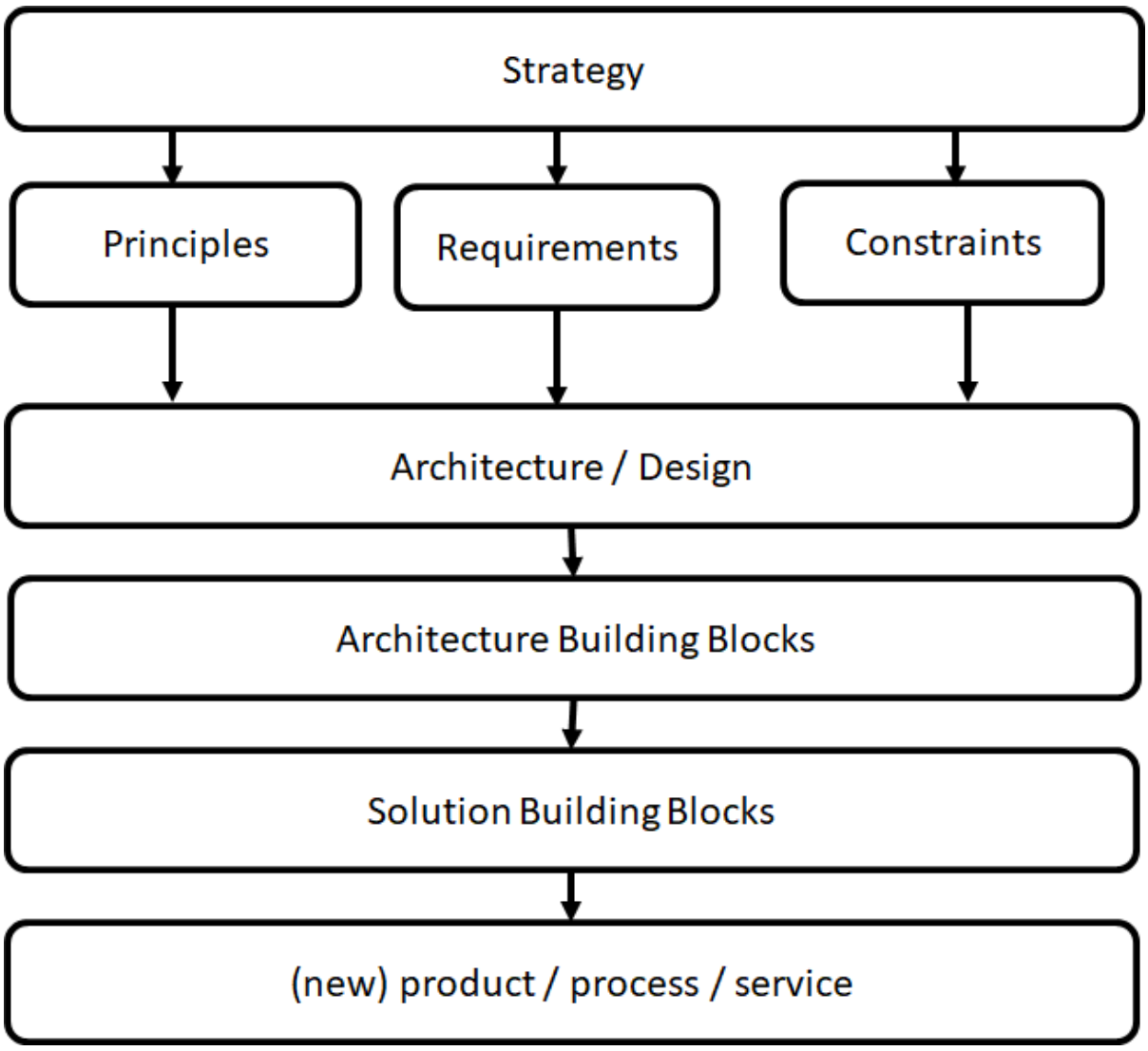
You will need to improve your machine learning model after a first test. Improving can be done using more training data or by making model adjustments.

2.5.2 ML Architecture Building Blocks

This reference architecture for machine learning gives guidance for developing solution architectures where machine learning systems play a major role. Discussions on what a good architecture is, can be a senseless use of time. But input on this reference architecture is always welcome. This to make it more generally useful for different domains and different industries. Note however that the architecture as described in this section is technology agnostics. So it is aimed at getting the architecture building blocks needed to develop a solution architecture for machine learning complete.

Every architecture should be based on a strategy. For a machine learning system this means an clear answer on the question: What problem must be solved using machine learning technology? Besides a strategy principles and requirements are needed.

The way to develop a machine learning architecture is outlined in the figure below.



In essence developing an architecture for machine learning is equal as for every other system. But some aspects require special attention. These aspects are outlined in this reference architecture.

Principles are statements of direction that govern selections and implementations. That is, principles provide a foundation for decision making.

Principles are commonly used within business design and successful IT projects. A simple definition of a what a principle is:

- A principle is a qualitative statement of intent that should be met by the architecture.

The key principles that are used for this reference machine learning architecture are:

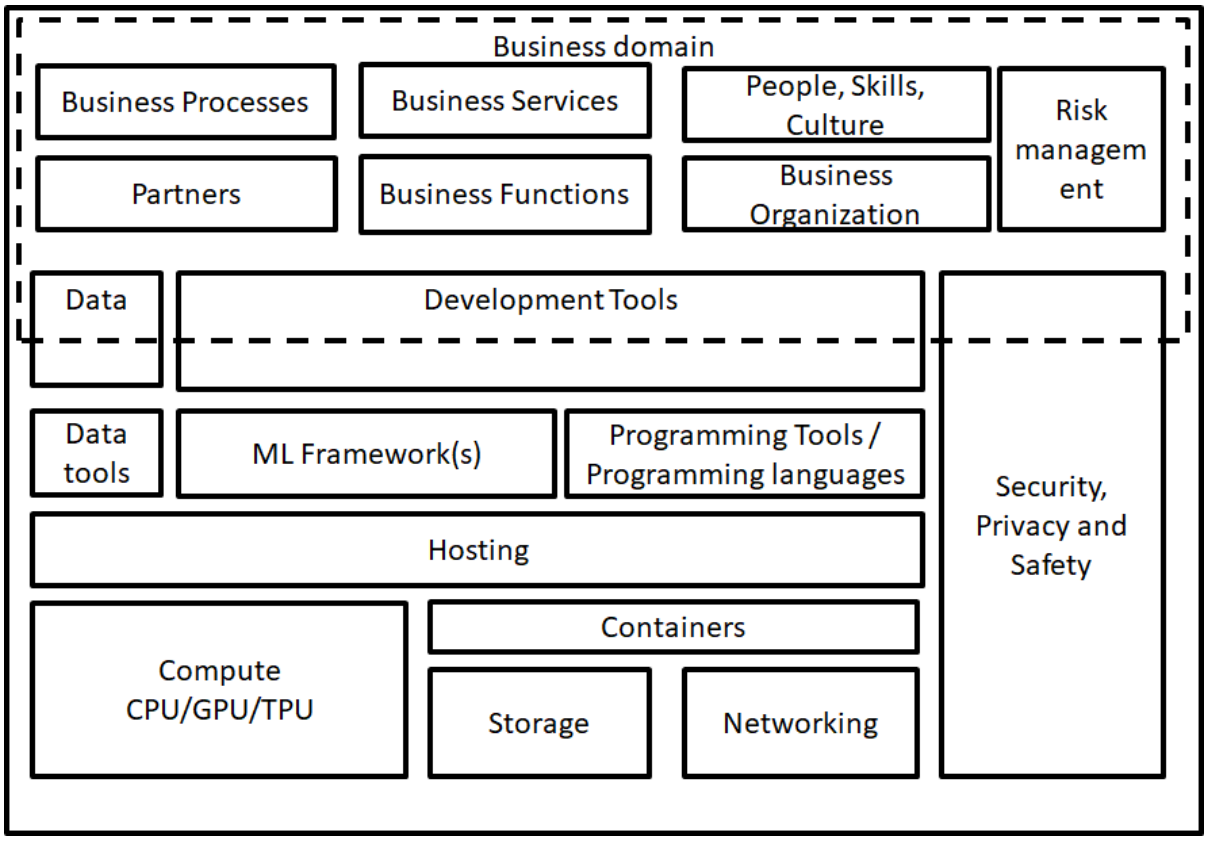
1. The most important machine learning aspects must be addressed.
2. The quality aspects: Security, privacy and safety require specific attention.
3. The reference architecture should address all architecture building blocks from development till hosting and maintenance.
4. Translation from architecture building blocks towards FOSS machine learning solution building blocks should be easily possible.
5. The machine learning reference architecture is technology agnostics. The focus is on the outlining the conceptual architecture building blocks that make a machine learning architecture.

By writing down these principles is will be easier to steer discussions on this reference architecture and to improve this machine learning architecture.

Machine learning architecture principles are used to translate selected alternatives into basic ideas, standards, and guidelines for simplifying and organizing the construction, operation, and evolution of systems.

Important concerns for this machine learning reference architecture are the aspects:

- Business aspects (e.g capabilities, processes, legal aspects, risk management)
- Information aspects (data gathering and processing, data processes needed)
- Machine learning applications and frameworks needed (e.g. type of algorithm, easy of use)
- Hosting (e.g. compute, storage, network requirements but also container solutions)
- Security, privacy and safety aspects
- Maintenance (e.g. logging, version control, deployment, scheduling)
- Scalability, flexibility and performance



Conceptual overview of machine learning reference architecture

Since this simplified machine learning reference architecture is far from complete it is recommended to consider e.g. the following questions when you start creating your solution architecture where machine learning is part of:

- Do you just want to experiment and play with some machine learning models?
- Do you want to try different machine learning frameworks and libraries in to discover what works best for your use case? Machine learning systems never work directly. You will need to iterate, rework and start all over again. Its innovation!
- Is performance crucial for your application?
- Are human lives direct or indirect dependent of your machine learning system?

In the following sections a more in depth description of the various machine learning architecture building blocks is given.

Business Processes

To apply machine learning with success it is crucial that the core business processes of your organization that will be affected with this new technology are determined. In most cases secondary business processes will benefit more than primary processes. Think of marketing, sales and quality aspects that make your primary business processes better.

Business Services

Business services are services that your company provides to customers, both internally and externally. When applying machine learning for business use you should create a map to outline what services are impacted, changed or disappear

when using machine learning technology. Are customers directly impacted or will your customer experience indirect benefits?

Business Functions

A business function delivers business capabilities that are aligned to your organization, but not necessarily directly governed by your organization. For machine learning it is crucial that the information that a business function needs is known. Also the quality aspects of this information should be taken into account. To apply machine learning it is crucial to know how information is exactly processes and used in the various business functions.

People, Skills and Culture

Machine learning needs a culture where experimentation is allowed. When you start with machine learning you and your organization need to build up knowledge and experience. Failure will happen and must be allowed. Fail hard and fail fast. Take risks. However your organization culture should be open to such a risk based approach. IT projects in general fail often so doing an innovative IT project using machine learning will be a risk that must be able to cope with. To make a shift to a new innovative experimental culture make sure you have different types of people directly and indirectly involved in the machine learning project. Also make use of good temporary independent consultants. So consultants that have also a mind set of taking risks and have an innovative mindset. Using consultants for machine learning of companies who sell machine learning solutions as cloud offering do have the risk that needed flexibility in an early stage is lost. Also to be free on various choices make sure you are not forced into a closed machine learning SaaS solution too soon. Since skilled people on machine learning with the exact knowledge and experience are not available you should use creative developers. Developers (not programmers) who are keen on experimenting using various open source software packages to solve new problems.

Business organization

Machine learning experiments need an organization structure that does not limit creativity. In general hierarchical organizations are not the perfect placed where experiments and new innovative business concepts can grow. Applying machine learning in an organization requires an organization that is data and IT driven. A perfect blueprint for a 100% good organization structure does not exist, but flexibility, learning are definitely needed. Depending on the impact of the machine learning project you are running you should make sure that the complete organization is informed and involved whenever needed.

Partners

Since your business is properly not Amazon, Microsoft or Google you will need partners. Partners should work with you together to solve your business problems. If you select partners pure doing a functional aspect, like hosting, data cleaning ,programming or support and maintenance you will miss the needed commitment and trust. Choosing the right partners for your machine learning project is even harder than for ordinary IT projects, due to the high knowledge factor involved. Some rule of thumbs when selecting partners: Big partners are not always better. With SMB partners who are committed to solve your business challenge with you governance structures are often easier and more flexible. Be aware for vendor lock-ins. Make sure you can change from partners whenever you want. So avoid vendor specific and black-box approaches for machine learning projects. Machine learning is based on learning, and learning requires openness. Trust and commitment are important factors when selecting partners. Commitment is needed since machine learning projects are in essence innovation projects that need a correct mindset. Use the input of your created solution architecture to determine what kind of partners are needed when. E.g. when your project is finished you need stability and continuity in partnerships more than when you are in an innovative phase.

Risk management

Running machine learning projects involves risk. Within your architecture it is crucial to address business and projects risks early. Especially when security, privacy and safety aspects are involved mature risks management is recommended. To make sure your machine learning project is not dead at launch, risk management requires a flexible and create approach for machine learning projects. Of course when your project is more mature openness and management on all risks involved is crucial. To avoid disaster machine learning projects it is recommended to create your:

- solution architecture using:
- Safety by design principles.
- Security by design principles and
- Privacy by design principles

In the beginning this will slow down your project, but doing security/privacy or safety later as 'add-on' requirements is never a real possibility and will take exponential more time and resources.

Development tools

In order to apply machine learning you need good tools to do e.g.:

- Create experiments for machine learning fast.
- Create a solid solution architecture
- Create a data architecture
- Automate repetitive work (integration, deployment, monitoring etc)

Fully integrated tools that cover all aspects of your development process (business design and software and system design) are hard to find. Even in the OSS world. Many good architecture tools, like Arch for creating architecture designs are still usable and should be used. A good overview for general open architecture tools can be found here https://nocomplexity.com/architecture-playbook/. Within the machine learning domain the de facto development tool is 'The Jupyter Notebook'. The Jupyter notebook is an web application that allows you to create and share documents that contain live code, equations, visualizations and narrative text. A Jupyter notebook is perfect for various development steps needed for machine learning suchs as data cleaning and transformation, numerical simulation, statistical modeling, data visualization and testing/tuning machine learning models. More information on the Jupyter notebook can be found here https://jupyter.org/ .

But do not fall in love with a tool too soon. You should be confronted with the problem first, before you can evaluate what tool makes your work more easy for you.

ML Frameworks

Machine Learning frameworks offer software building blocks for designing, training and validating your machine learning model. Most of the time you will only be confronted with your ML framework using a high level programming interface. All major FOSS ML frameworks offer APIs for all major programming languages. Almost all 'black magic' needed for creating machine learning application is hidden in a various software libraries that make a ML framework.

In another section of this book a full overview of all major ML frameworks will be presented. But for creating your architecture within your specific context choosing a ML framework that suits your specific use case is a severe difficult task. Of course you can skip this task and go for e.g. Tensorflow in the hope that your specific requirements are offered by simple high level APIs.

Some factors that must be considered when choosing a ML framework are:

- Stability. How mature, stable is the framework?

- Performance. If performance really matters a lot for your application (training or production) doing some benchmark testing and analysis is always recommended.
- Features. Besides the learning methods that are supported what other features are included? Often more features, or support for more learning methods is not better. Sometimes simple is enough since you will not change your machine learning method and model continuously.
- Flexibility. How easy is it to switch to another ML framework, learning method or API?
- Transparency. Machine learning development is a very difficult tasks that involves a lot of knowledge of engineers and programmers. Not many companies have the capabilities to create a ML framework. But in case you use a ML framework: How do you known the quality? Is it transparent how it works, who has created it, how it is maintained and what your business dependencies will be!
- License. Of course we do not consider propriety ML frameworks. But do keep in mind that the license for a ML framework matters. And make sure that no hooks or dual-licensing tricks are played with what you think is an open ML Framework.
- Speeding up time consuming and recurrent development tasks.

Debugging a machine learning application is no fun and very difficult. Most of the time you spend time with model changes and retraining. But knowing why your model is not working as well as expected is a crucial task that should be supported by your ML framework.

There are too many open source machine learning frameworks available which enable you to create machine learning applications. Almost all major OSS frameworks offer engineers the option to build, implement and maintain machine learning systems. But real comparison is a very complex task. And the only way to do some comparison is when ML frameworks are open source. And since security, safety and privacy should matter for every use case there is no viable alternative than using an mature OSS ML framework.

Programming Tools

You can use every programming language for developing your machine learning application. But some languages are better suited for creating machine learning applications than others. The top languages for applying machine learning are:

- Python.
- Java and
- R

The choice of the programming language you choice depends on the ML framework, the development tools you want to use and the hosting capabilities you have. For fast iterative experimentation a language as Python is well suited. And besides speeds for running your application in production also speed for development should be taken into concern.

There is no such thing as a 'best language for machine learning'.

There are however bad choices that you can make. E.g. use a new development language that is not mature, has no rich toolset and no community of other people using it for machine learning yet.

Within your solution architecture you should justify the choice you make based upon dependencies as outlined in this reference architecture. But you should also take into account the constraints that account for your project, organisation and other architecture factors that will drive your choice. If have e.g. a large amount of Java applications running and all your processes and developers are Java minded, you should take this fact into account when developing and deploying your machine learning application.

Data

Data is the heart of the machine earning and many of most exciting models don't work without large data sets. Data is the oil for machine learning. Data is transformed into meaningful and usable information. Information that can be used for humans or information that can be used for autonomous systems to act upon.

In normal architectures you make a clear separation when outlining your data architecture. Common view points for data domains are: business data, application data and technical data For any machine learning architecture and application data is of utmost importance. Not all data that you use to train your machine learning model needs can be originating from you own business processes. So sooner or later you will need to use data from other sources. E.g. photo collections, traffic data, weather data, financial data etc. Some good usable data sources are available as open data sources. For a open machine learning solution architecture it is recommended to strive to use open data. This since open data is most of the time already cleaned for privacy aspects. Of course you should take the quality of data in consideration when using external data sources. But when you use data retrieved from your own business processes the quality and validity should be taken into account too.

Free and Open Machine learning needs to be feed with open data sources. Using open data sources has also the advantage that you can far more easily share data, reuse data, exchange machine learning models created and have a far easier task when on and off boarding new team members. Also cost of handling open data sources, since security and privacy regulations are lower are an aspect to take into consideration when choosing what data sources to use.

For machine learning you will need 'big data'. Big data is any kind of data source that has one the following properties:

- Big data is data where the volume, velocity or variety of data is (too) great.So big is really a lot of data!
- The ability to move that data at a high Velocity of speed.
- An ever-expanding Variety of data sources.
- Refers to technologies and initiatives that involve data that is too diverse, fast-changing or massive for conventional technologies, skills and infra- structure to address efficiently.

Every Machine Learning problem starts with data. For any project most of the time large quantities of training data are required. Big data incorporates all kind of data, e.g. structured, unstructured, metadata and semi-structured data from email, social media, text streams, images, and machine sensors (IoT devices).

Machine learning requires the right set of data that can be applied to a learning process. An organization does not have to have big data in order to use machine learning techniques; however, big data can help improve the accuracy of machine learning models. With big data, it is now possible to virtualise data so it can be stored in the most efficient and cost-effective manner whether on- premises or in the cloud.

Within your machine learning project you will need to perform data mining. The goal of data mining is to explain and understand the data. Data mining is not intended to make predictions or back up hypotheses.

One of the challenges with machine learning is to automate knowledge to make predictions based on information (data). For computer algorithms everything processed is just data. Only you know the value of data. What data is value information is part of the data preparation process. Note that data makes only sense within a specific context.

The more data you have, the easier it will be to apply machine learning for your specific use case. With more data, you can train more powerful models.

Some examples of the kinds of data machine learning practitioners often engage with:

- Images: Pictures taken by smartphones or harvested from the web, satellite images, photographs of medical conditions, ultrasounds, and radiologic images like CT scans and MRIs, etc.
- Text: Emails, high school essays, tweets, news articles, doctor's notes, books, and corpora of translated sentences, etc.
- Audio: Voice commands sent to smart devices like Amazon Echo, or iPhone or Android phones, audio books, phone calls, music recordings, etc.

- Video: Television programs and movies, YouTube videos, cell phone footage, home surveillance, multi-camera tracking, etc.
- Structured data: Webpages, electronic medical records, car rental records, electricity bills, etc
- Product reviews (on Amazon, Yelp, and various App Stores)
- User-generated content (Tweets, Facebook posts, StackOverflow questions)
- Troubleshooting data from your ticketing system (customer requests, support tickets, chat logs)

When developing your solution architecture be aware that data is most of the time:

- Incorrect and
- useless.

So meta data and quality matters. Data only becomes valuable when certain minimal quality properties are met. For instance if you plan to use raw data for automating creating translating text you will soon discover that spelling and good use of grammar do matter. So the quality of the data input is an import factor of the quality of the output. E.g. automated Google translation services still struggle with many quality aspects, since a lot of data captures (e.g. captured text documents or emails) are full of style,grammar and spell faults.

Data science is a social process. Data is generated by people within a social context. Data scientists are social people who will have to do a lot of communication with all kind of business stakeholders. Data scientist should not work in isolation because the key thing is to find out what story is told within the data set and what import story is told over the data set.

Data Tools

Without data machine learning stops. For machine learning you will be dealing with large complex data sets (maybe even big data) and the only way to make machine learning applicable is data cleaning and preparation. So you need good tools to handle data.

The number of tools you will need will depend of the quality of your data sets, your experience, development environment and other choice you will have to make in your solution architecture. But a view use cases where good solid data tools will help are:

- Data visualization and viewer tools; Good data exploration tools give visual information about the data sets without a lot of custom programming.
- Data filtering, data transformation and data labelling;
- Data anonymiser tools;
- Data encryption / decryption tools
- Data search tools (analytics tools)

Without good data tools you are lost when doing machine learning for real. The good news is: There are a lot of OSS data tools you can use. Depending if you have raw csv, json or syslog data you will need other tools to prepare the dataset. The challenge is to choose tools that integrate good in your landscape and save you time when preparing your data for starting developing your machine learning models. Since most of the time when developing machine learning applications you will be fighting with data, it is recommended to try multiple tools. Most of the time you will learn that a mix of tools is the best option, since a single data tool will never cover all your needs. So leave some freedom within your architecture for your team members who will be dealing with data work (cleaning, preparation etc).

The field of 'data analytics' and 'business intelligence' is a mature field for decades within IT. So you will find many tools that are excellent for data analytics and/or reporting. But keep in mind that the purpose of fighting with data for machine learning is in essence only for data cleaning and feature extraction. So be aware of 'old' tools that are rebranded as new data science tools for machine learning. There is no magic data tool preparation of data for machine learning. Sometimes old-skool unix tool like awk or sed will just do the job.

Besides tools that assist you with preparing the data pipeline, there are also good (open) tools for finding open datasets that you can use for your machine learning application. See the reference section for some tips.

To prepare your data working with the data within your browser seems a nice idea. You can visual connect data sources and e.g. create visuals by clicking on data. Or inspecting data in a visual way. There is however one major drawback: Despite the great progress made on very good and nice looking JavaScript frameworks for visualization, handling data within a browser DOM is and will take your browser over the limit. You can still expect hang-ups, indefinitely waits and very slow interaction. At least when not implemented well. But implementation of on screen data visualisation (Drag-and-Drop browser based) is requires an architecture and design approach that focus on performance and usability from day 1. Unfortunately many visual web based data visualization tools use an generic JS framework that is designed from another angle. So be aware that if you try to display all your data, it will eat all your resources(CPU, memory) and you will get a lot of frustration. So most of the time using a Jupyter Notebook will be a safe choice when preparing your data sets.

Hosting

Hosting infrastructure is platform that is capable of running your machine learning application(s). Hosting is a separate block in this reference architecture to make you aware that you must make a number of choices. These choices concerning hosting your machine learning application can make or break your machine learning adventure.

It is a must to make a clear distinguishing in:

1. Hosting infrastructure needed for development and training and
2. Hosting infrastructure needed for production

Depending on your application it is e.g. possible that you need a very large and costly hosting infrastructure for development, but you can do deployment of your trained machine learning model on e.g. a Raspberry PI or Arduino board.

Standard hosting capabilities for machine learning are not very different as for 'normal' IT services. Expect scalability and flexibility capabilities requires solid choices from the start. The machine learning hosting infrastructure exist e.g. out of:

- Physical housing and power supply.
- Operating system (including backup services).
- Network services.
- Availability services and Disaster recovery capabilities.
- Operating services e.g. deployment,, administration, scheduling and monitoring.

For machine learning the cost of the hosting infrastructure can be significant due to performance requirements needed for handling large datasets and training your machine learning model.

A machine learning hosting platform can make use of various commercial cloud platforms that are offered(Google, AWS, Azure, etc). But since this reference architecture is about Free and Open you should consider what services you will use from external Cloud Hosting Providers (CSPs) and when. The crucial factor is most of the time cost and the number of resources needed. To apply machine learning it is possible to create your own machine learning hosting platform. But in reality this is not always the fasted way if you have not the required knowledge on site.

All major Cloud hosting platforms do offer various capabilities for machine learning hosting requirements. But since definitions and terms differ per provider it is hard to make a good comparison. Especially when commercial products are served instead of OSS solutions. So it is always good to take notice of:

- Flexibility (how easy can you switch from your current vendor to another?).
- Operating system and APIs offered. And
- Hidden cost

For experimenting with machine learning there is not always a direct need for using external cloud hosting infrastructure. It all depends on your own data center capabilities. In a preliminary phase even a very strong gaming desktop with a good GPU can do.

When you want to use machine learning you need a solid machine learning infrastructure. Hosting Infrastructure done well requires a lot of effort and is very complex. E.g. providing security and operating systems updates without impacting business applications is a proven minefield.

For specific use cases you can not use a commodity hosting infrastructure of a random cloud provider. First step should be to develop your own machine learning solution architecture. Based on this architecture you can check what capabilities are needed and what the best way is for starting.

The constant factor for machine learning is just as with other IT systems: **Change**.

So to minimize the risks make sure you a good view on all your risks. Your solution architecture should give you this overview, including a view of all objects and components that will be changed (or updated) sooner or later. Hosting a machine learning application is partly comparable with hosting large distributed systems. And history learns that this can still be a problem field if not managed well. So make sure what dependencies you will accept regarding hosting choices and what depencies you want to avoid.

Containers

Understanding container technology is crucial for using machine learning. Using containers within your hosting infrastructure can increase flexibility or if not done well decrease flexibility due to the extra virtualization knowledge needed.

The advantage and disadvantages of the use of Docker or even better Kubernetes or LXD or FreeBSD jails should be known. However is should be clear: Good solid knowledge of how to use and manage a container solution so it benefits you is hard to get.

Using containers for developing and deploying machine learning applications can make life easier. You can also be more flexible towards your cloud service provider or storage provider. Large clusters for machine learning applications deployed on a container technology can give a great performance advantage or flexibility. All major cloud hosting providers also allow you to deploy your own containers. In this way you can start small and simple and scale-up when needed.

Summarized: Container solutions for machine learning can be beneficial for:

- Development. No need to install all tools and frameworks.
- Hosting. Availability and scalability can be solved using the container infrastructure capabilities.
- Integration and testing. Using containers can simplify and ease a pipeline needed to produce quality machine learning application from development to production. However since the machine learning development cycle differs a bit from a traditional CICD (Continuous Integration - Continuous Deployment) pipeline, you should outline this development pipeline to production within your solution architecture in detail.

GPU - CPU or TPU

Not so long ago very large (scientific) computer cluster were needed for running machine learning applications. However due to the continuous growth of power of 'normal' consumer CPUs or GPUs this is no longer needed.

GPUs are critical for many machine learning applications. This because machine learning applications have very intense computational requirements. GPUs are general better equipped for some massive number calculation operations that the more generic CPUs.

You will also read and hear about TPUs. A tensor processing unit (TPU) is an AI accelerator application-specific integrated circuit (ASIC). First developed by Google specifically for neural network machine learning. But currently more companies are developing TPUs to support machine learning applications.

Within your solution architecture you should be clear on the compute requirements needed. Some questions to be answered are:

- Do you need massive compute requirements for training your model?
- Do you need massive compute requirements for running of your trained model?

In general training requires far more compute resources than is needed for production use of your machine learning application. However this can differ based on the used machine learning algorithm and the specific application you are developing.

Many machine learning applications are not real time applications, so compute performance requirements for real time applications (e.g. real time facial recognition) can be very different for applications where quality and not speed is more important. E.g. weather applications based on real time data sets.

2.6 Security,Privacy and Safety

2.6.1 Introduction

This section outlines security, privacy and safety concerns that matter when applying machine learning for real business use.

The complexity of ML technologies has fuelled fears that machine learning applications will cause harm in unforeseen circumstances, or that they will be manipulated to act in harmful ways. Think of a self driving car with its own ethics or algorithms that make prediction based on your personal data that really scare you. E.g. Predicting what diseases will hit you based on data from your grocery store.

As with any technology: Technology is never neutral. You have to think before starting what values you implicitly use to design your new technology. All technology can and will be misused. But it is up to the designers to think of the risks when technology will be misused. On purpose or by accident.

Machine learning systems should be operated reliably, safely and consistently. Not only under normal circumstances but also in unexpected conditions or when they are under attack for misuse.

Machine learning software differs from traditional software because:

- The outcome is not easily predictable.
- The used trained models are a black box, with very few options for transparency.
- Logical reasoning (or cause and effect) is not present. Predictions are made based on statistical number crunching complex algorithms which are non linear.
- Both Non IT people and trained IT people will have a hard time figuring out machine learning systems, due to the new paradigms in use.

What makes security and safety more than normal aspects for machine learning driven applications is that by design neural networks are not designed to to make the inner workings easy to understand for humans and quality and risk managers.

Without a solid background in mathematics and software engineering evaluating the correct working of most machine learning application is impossible for security researchers safety auditors.

However more and more people will dependent on the correct outcome of decisions made by machine learning software. So we should ask some critical questions:

- Is the system making any mistakes?
- How do you know what alternatives were considered?
- What is the risk of trusting the outcome blind?

Understanding how output produced by machine learning software is created will make more people comfortable with self-driving cars and other safety critical systems that will be machine learning enabled. In the end systems that can kill you must be secure and safe to use. So how do we get the process and trust chain to a level that we are not longer depended of:

- Software bugs
- Machine learning experts
- Auditors
- A proprietary certification process that end with a stamp (if paid enough)

From other sectors, like finance or oil industry we know that there is no simple solution. However regarding the risks involved only FOSS machine learning applications have the right elements needed to start working on processes that will give enough trust to use machine learning system for society at large.

2.6.2 Security

Using machine learning technology gives some serious new threads. More and more new ways for exploiting the technology are published. IT security is proven to be hard and complex to control and manage. But machine learning technology makes the problem of IT security even worse. This is due to the fact that the special created machine learning exploits are very hard to determine.

Machine learning challenges many current security measurements. This because machine learning software:

- Lowers the cost of applying current known attacks on all devices which depend on software. So almost all modern technology devices.
- Machine learning software enables the easy creation of new threats and vulnerabilities on existing systems. E.g. you can take the CVE security vulnerability database (https://www.cvedetails.com/) and train a machine learning model how to create attack on the published omissions.
- When machine learning software will be in hospitals, traffic control systems, chemical fabrics and IoT devices machine learning gives easier options to create a complete new attack surface as with traditional software.

Security aspects for machine learning accounts for the application where machine learning is used, but also for the developed algorithms self. So machine learning security is divided into two main categories:

1. Machine learning attacks aimed to fool the developed machine learning systems. Since machine learning is often a 'black-box' these attacks are very hard to determine.
2. Machine learning offers new opportunities to break existing traditional software systems.
3. Usage threats. The outcome of many machine learning systems is far from correct. If you base decisions or trust on machine learning application you can make serious mistakes. This accounts e.g. for self driving vehicles, health care systems and surveillance systems. Machine learning systems are known for producing racially biased results often caused by using biased data sets. Think about problematic forms of "profiling" based on surveillance cameras with face detection.

Some examples of machine learning exploits:

- Google's Cloud Computing service can be tricked into seeing things that are not there. In one test it perceived a rifle as a helicopter.
- Fake videos made with help from machine learning software are spreading online, and the law can't do much about it. E.g. videos with speeches given by political leaders created by machine learning software are created and spread online. E.g. a video where some president declares a war to another country is of course very dangerous. Even more dangerous is the fact that the fake machine learning created videos are very hard to diagnose as machine learning creations. This since besides machine learning a lot of common Hollywood special effects are also used to make it hard to distinguish real videos from fake video's. Creating online fake

porn video sites were you can use a photo of a celebrity or someone you really do not like is nowadays only just three mouse clicks away. And the reality is that you can do very little against these kinds of damaging threads. Even from a legal point of view.

Users and especially developers of machine learning applications must be more paranoid from a security point of view. But unfortunately security cost a lot of effort and money and a lot of special expertise is needed to minimize the risks.

2.6.3 Privacy

Machine learning raises serious privacy concerns since machine learning is using massive amounts of data that contain often personal information.

It is a common believe that personal information is needed for experimenting with machine learning before you can create good and meaningful applications. E.g. for health applications, travel applications, eCommerce and of course marketing application. Machine learning models are often loaded with massive amounts of personal data for training and to make in the end good meaningful predictions.

The belief that personal data is needed for machine learning creates a tension between developers and privacy aware consumers. Developers want the ability to create innovative new products and services and need to experiment, while consumers and GDPR regulators are concerned for the privacy risks involved.

The applicability of machine learning models is hindered in settings where the risk of data leakage raises serious privacy concerns. Examples of such applications include scenar- ios where clients hold sensitive private information, e.g., medical records, financial data, or location.

It is commonly believed that individuals must provide a copy of their personal information in order for AI to train or predict over it. This belief creates a tension between developers and consumers. Developers want the ability to create innovative products and services, while consumers want to avoid sending developers a copy of their data.

Machine learning models can be trained in environments that are not secure on data it never has access to. Secure machine learning that works on anonymized data sets is still an obscure and unpaved path. But some companies and organizations are already working on creating deep learning technology that work on encrypted data. Using encryption on data to train machine learning models raises the complexity in various ways. It is already hard to get inside the 'black-box' of the working of machine learning. Using advanced data encryption will require even more knowledge and competences for all engineers involved when developing machine learning applications.

In the EU the use of personal data is protected by law in all countries by a single law. The EU General Data Protection Regulation (GDPR). This GDPR does not prohibit the use of machine learning. But when you use personal data you will have a severe challenge to explain to DPOs (Data Protection Officers) and consumers what you actually do with the data and how you comply with the GDPR.

When you apply machine learning for your business application you should consider the following questions:

- In what way will your customers be happy with their data usage for their and your benefit?
- Do you really have a clear and good overview of all GDPR implications when using personal data in your machine learning model? What happens if you invite other companies to use your model?
- What are the ethical concerns when using massive amounts of data of your customers to develop new products? Is the way you use the data to train your model congruent with you business vision and moral?
- What are the privacy risks involved for your machine learning development chain and application?

2.7 Machine Learning for Business Problems

Reading and talking about the in potential endless options for machine learning is nice and should be done. But applying machine learning for your business is where you can make a difference. This section is focussed on applying

machine learning technology for real business use. Some business use cases where machine learning is already applied are outlined. This to give you some inspiration. But besides the technology more is needed for applying machine learning in a business with success. This section will also present some more in depth discussion of the several factors that should be taken into account when applying machine learning for real business use.

2.7.1 When to use machine learning for business problems?

With the use of machine learning it is possible to learn from patterns and conditions to get new solid outcomes or predictions based on new data. Machine learning is able to learn from change patterns (data) at a pace that the human mind can not. This makes that machine learning is useful for a set of use cases were learning from data is possible or needed.

Machine learning should not be used for use cases that can be easily solved in another way. For example do not use machine learning driven solutions if your use case matches on of the following criteria:

- If it's possible to structure a set of rules or "if-then scenarios" to handle your problem entirely, then there may be no need for machine learning at all.
- Your problem can be solved using statistical tools and software.

2.7.2 Common business use cases

Healthcare

Healthcare is due to the large amounts of data already available a perfect domain for solving challenges using machine learning. E.g. a challenging question for machine learning for healthcare is: Given the current use of machine learning for healthcare is given a patient's electronic medical record data, can we prevent a person getting sick?

Machine learning is more and more used for automatic diagnostics. This can be data provided by X-ray scans or data retrieved from blood and tissue samples. machine learning has already proven to be valuable in detecting and predicting diseases for real people.

Predictive tasks for healthcare is maybe the way to keep people healthier and lower healthcare cost. The transformation from making people better towards preventing people getting sick will be long and hard, since this will be a real shift for the healthcare industry.

But given a large set of training data of de-identified medical records it is already possible to predict interesting aspects of the future for a patient that is not in training set.

Machine learning applications for healthcare are also to create better medicines by making use of all the data already available.

Language translation

Machine learning is already be used for automatic real-time message translation. E.g. Rocket Chat (The OSS Slack alternative, https://rocket.chat/) is using machine learning for this reason. Since language translation needs context and lots of data, typically this use case is often NLP driven. Language translation is as speech recognition a typical NLP application. Natural language processing (NLP) is area of machine learning that operates on (human)text and speech. See the section on NLP in this book for more use cases and insight in the specific NLP technologies.

Other areas for language translation are Speech recognition. Some great real time machine learning driven application already exist.

Chat bots

Currently all major tech companies like Amazon(Alexis), Google, Apple (Siri) have built a smart chatbot for the consumer market. Creating a chatbot (e.g. IRQ bot) was not new and difficult, however building a real 'intelligent' chat bot that has learning capabilities is another challenge.

Machine learning powered chatbots with real human like voices will help computers communicate with humans. But algorithms still have a hard time trying to figure out what you are saying, because context and tone of voice are hard to get right. Even for us humans, communication with other humans is most of the time hard. So building a smart chatbot that understands basic emotions in your voice is difficult. Machine learning isn't advanced enough yet to carry on a dialogue without help, so a lot of the current chat bot software needs to be hand-coded.

Warning: This document is in alfa-stage!! We are now working on some great NLP things. So not yet a NLP section in this alfa verion of this eBook! Collaboration is fun, so *Help Us* by contributing !

eCommerce Recommendation systems

A well known application for machine learning for eCommerce systems is a machine learning enabled recommendation system system. Whether you buy a book, trip, music or visit a movie: On all major online ecommerce site you get a recommendation for a product that seems to fit your interest perfectly. Of course the purpose is to drive up the online sale, but the algorithms used are examples of still evolving machine learning algorithms. Examples of these systems are:

- Selling tickets to concerts based on your profile.
- NetFlix or cinema systems to make sure you stay hooked on watching more series and films you like.
- Finding similar products in an eCommerce environment with a great chance you buy it. E.g. similar hotels, movies, books, etc.

Quality inspection and improvement

When computer vision technologies are combined with machine learning capabilities new opportunities arise. Examples of real world applications are:

- Inspecting tomatoes (and other fruit / vegetables) for quality and diseases.
- Inspecting quality of automatic created constructions (e.a. Constructions made by robots)

Vision

Since vision is captured in data machine learning is a great tool for building applications using vision (images, movies) technology. E.g.:

- Face detection. Writing software to detect faces and do recognition is very very hard to do using traditional programming methods.
- Image classification. In the old days we were happy when software was able to distinguish a cat and dog. In 2018 far more advanced applications are possible. E.g. giving details on all kind of aspects of photos. E.g. when you organize a conference you can use software to check the amount of suits or hoodies visiting your conference. Which is of course great for marketing.

- Image similarity. Given an image, the goal of an image similarity model is to find "similar" images. Just like in image classification, deep learning methods have been shown to give incredible results on this challenging problem. However, unlike in image similarity, there isn't a need to generate labelled images for model creation. This model is completely unsupervised.
- Object Detection. Object detection is the task of simultaneously classifying (what) and localizing (where) object instances in an image.

Security

- Email spam filters. Although simple rules can and should be applied, the enormous creativity of spammers and the amount send good fighting spam is a solid use case for a supervised machine learning problem.
- Network filtering. Due to the learning capability of machine learning network security devices are improved using machine learning techniques.
- Fraud detection. Fraud detection is possible using enormous data and searching for strange patterns.

Besides fraud detection machine learning can also applied for IT security detections since intrusion detection systems and virus scanners are more and more shipped with self learning algorithms. Also Complex financial fraud schemes can be easily detected using predictive machine learning models.

Risk and compliance

Evaluating risks can be done using large amounts of data. Natural language processing techniques can be used to validate highly automatic if your company meets regulations. Since audit and inspecting work is mostly based on standardized rules performed by knowledge workers this kind of work can be automated using machine learning techniques.

2.7.3 Example use cases

To outline some use cases that have been realized using machine learning technology, this paragraph summarize some real world cases to get some inspiration.

Applications for real business use of machine learning to solve real tangible problems is growing at a rapid pace. Below some examples of practical use of machine learning applications:

- Medical researchers are using machine learning to assess a person's cardiovascular risk of a heart attack and stroke.
- Air Traffic Controllers are using TensorFlow to predict flight routes through crowded airspace for safe and efficient landings.
- Engineers are using TensorFlow to analyze auditory data in the rainforest to detect logging trucks and other illegal activities.
- Scientists in Africa are using TensorFlow to detect diseases in Cassava plants to improving yield for farmers.
- Finding free parking space. http://www.peazy.in has developed an app using machine learning to assist with finding a free parking space in crowded cities.

2.7.4 Exiting ML business examples

In this section some worth mentioning exiting real business examples for companies that really make use of new ML solutions possible.

- AI Driven Logos. An AI solution which selects the best possible logos for your brand based on a large number of designs it has seen over time. Check https://www.designwithai.com/

2.7.5 Business principles for Machine Learning applications

Every good architecture is based on principles, requirements and constraints.This machine learning reference architecture is designed to easy the process of creating machine learning solutions. So below some general principles for machine learning applications. For your specific machine learning application use the principles that apply and make them smart. So include implications and consequences per principle.

Collaborate

Statement: Collaborate Rationale: Successful creation of ML applications require the collaboration of people with different expertises. You need e.g. business experts, infrastructure engineers, data engineers and innovation experts. Implications: Organisational and culture must allow open collaboration.

Unfair bias

Statement: Avoid creating or reinforcing unfair bias Rationale: Machine learning algorithms and datasets can reflect, reinforce, or reduce unfair biases. Recognize fair from unfair biases is not simple, and differs across cultures and societies. However always make sure to avoid unjust impacts on sensitive characteristics such as race, ethnicity, gender, nationality, income, sexual orientation, ability, and political or religious belief. Implications: Be transparent about your data and training datasets. Make models reproducible and auditable.

Built and test for safety

Statement: Built and test for safety. Rationale: Use safety and security practices to avoid unintended results that create risks of harm. Design your machine learning driven systems to be appropriately cautious Implications: Perform risk assessments and safety tests.

Privacy by design

Statement: Incorporate privacy by design principles. Rationale: Privacy by principles is more than being compliant with legal constraints as e.g. EU GDPR. It means that privacy safeguards,transparency and control over the use of data should be taken into account from the start. This is a hard and complex challenge.

2.7.6 Business ethics

There are good and bad uses for any technology. So with ML technology. Working with machine learning can, will and must raise severe ethical questions. Machine learning can be used in many bad ways. Saying that you 'Don't be evil' , like the mission statement of Google (https://en.wikipedia.org/wiki/Don%27t_be_evil) was for decades, will not save you. Any business that uses machine learning should develop a process in order to handle ethical issues before they arrive. And ethical questions will arise!

A growing number of experts believe that a third revolution will occur during the 21st century, through the invention of machines with intelligence which surpasses our own intelligence. The rapid progress in machine learning technology turns out to be input for all kind of disaster scenarios. When the barriers to apply machine learning will be lowered more one of the fears is that knowledge work and various mental tasks currently performed by humans will become obsolete.

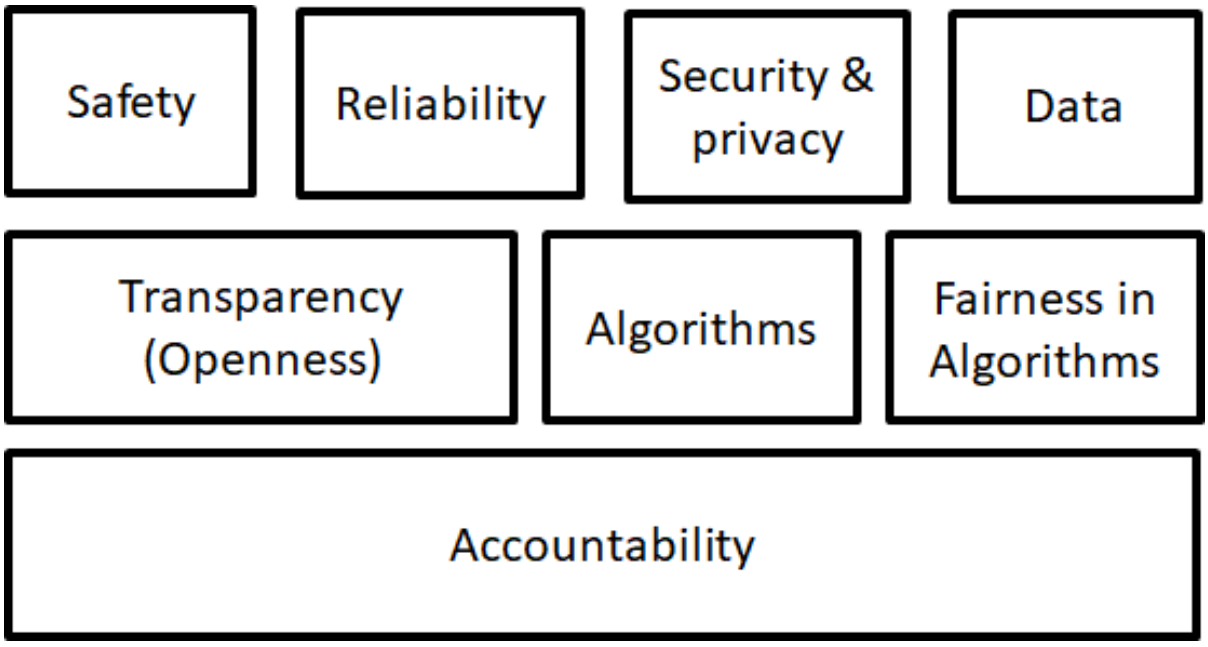
When machine learning develops and the border with artificial intelligence will be hit many more philosophical and ethical discussions will take place. The core question is of course: What is human intelligence? Or to put it in the context of machine learning: What is the real value of human intelligence when machine learning algorithms can take over many common mental tasks of humans?

Many experts believe that there is a significant chance we will develop machines more intelligent than ourselves within a few decades. This could lead to large, rapid improvements in human welfare, or mass unemployment and poverty on large scale. And yes history learns that there are good reasons to think that it could lead to disastrous outcomes for our current societies. If machine learning research advances without enough research work going on security, safety on privacy, catastrophic accidents are likely to occur.

With FOSS machine learning capabilities you should be able to take some control over the rapid pace machine learning driven software is hitting our lives. So instead of trying to stop developments and use, it is better to steer developments into a positive, safe, human centric direction. So apply machine learning using a decent machine learning architecture were also some critical ethical business questions are addressed.

Advances within machine learning could lead to extremely positive developments, presenting solutions to now-intractable global problems. But applying machine learning without good architectures where ethical questions are also addressed, using machine learning at large can pose severe risks. Humanity's superior intelligence is the sole reason that we are the dominant species on our planet. If technology with advanced machine learning algorithms surpass humans in intelligence, then just as the fate of gorillas currently depends on the actions of humans, the fate of humanity may come to depend more on the actions of machines than our own.

To address ethical questions for your machine learning solution architecture you can use the high level framework below.



Some basic common ethical questions for every machine learning architecture are:

- Bias in data sets. How do you weight this? Are you fully aware of the impact?
- Impact on your company.
- Impact on your employees.
- Impact on your customers (short and long term).
- Impact on society.
- Impact on available jobs and future man force needed.

- Who is responsible and who is liable when the application developed using machine learning goes seriously wrong?
- Do you and your customers find it acceptable all kinds of data are combined to make more profit?
- How transparent should you inform your customers on how privacy aspects are taken into account when using the machine learning software? Legal baselines, like the EU GDPR do not answer these ethical questions for you!
- How transparent are you towards stakeholders regarding various direct and indirect risks factors involved when applying machine learning applications?
- Who is responsible and liable when risks in your machine learning application do occur?

A lot of ethical questions come back to crucial privacy and other risks questions like safety and security. We live in a digital world were our digital traces are everywhere. Most of the time we are fully unaware. In most western countries mass digital surveillance cameras generates great data to be used for machine learning algorithms. This can be noble by detecting diseases based on cameras, but all nasty use cases thinkable are of course also under development. Continuous track and trace of civilians including face recognition is not that uncommon any more!

The question regarding who is responsible for negative effects regarding machine learning technology is simple to answer. You are! If you do not understand the technology, the impact for your business and on society you should not use it.

Regulations for applying machine learning are not yet developed. Although some serious thinking is already be done in the field regarding:

- Safety and
- Liability

Government rules, laws will be formed during the transition the coming decade. Machine learning techniques are perfect to use for autonomous weapons. So drones will in near future decide based on hopefully predefined rules when to launch a missile and when not. But as with all technologies: Failures will happen! And we all hope it will not hit us.

2.8 Catalogue of Open ML Software

This section presents the most widespread, mature and promising open source ML software available. The purpose of this section is just to make you curious to maybe try something that suits you.

ML software comes in many different forms. A lot can be written on the differences on all packages below, the quality or the usability. Truth is however there is never one best solution. Depending your practical use case you should make a motivated choice for what package to use.

As with many other evolving technologies in heavy development: Standards are still lacking, so you must ensure that you can switch to another application with minimal pain involved. By using a real open source solution you already have taken the best step! Using OSS makes you far more independent than using ML cloud solutions. This because these work as 'black-box' solutions and by using OSS you can always build your own migration interfaces if needed. Lock-in for ML is primarily in the data and your data cleansing process. So always make sure that you keep full control of all your data and steps involved in the data preparation steps you follow. The true value for all ML solutions is of course always in the data.

Todo: The OSS ML list will be completed, filtered, adjusted and corrected soon! Want to help?

2.8.1 Acumos AI

Acumos AI is a platform and open source framework that makes it easy to build, share, and deploy AI apps. Acumos standardizes the infrastructure stack and components required to run an out-of-the-box general AI environment.

Acumos is a platform which enhances the development, training and deployment of AI models. Its purpose is to scale up the introduction of AI-based software across a wide range of industrial and commercial problems in order to reach a critical mass of applications. In this way, Acumos will drive toward a data-centric process for producing software based upon machine learning as the central paradigm. The platform seeks to empower data scientists to publish more adaptive AI models and shield them from the task of custom development of fully integrated solutions. Ideally, software developers will use Acumos to change the process of software development from a code-writing and editing exercise into a classroom-like code training process in which models will be trained and graded on their ability to successfully analyze datasets that they are fed. Then, the best model can be selected for the job and integrated into a complete application.

Acumos is part of the LF Deep Learning Foundation, an umbrella organization within The Linux Foundation that supports and sustains open source innovation in artificial intelligence, machine learning, and deep learning while striving to make these critical new technologies available to developers and data scientists everywhere.

SBB License	Apache License 2.0
Core Technology	Java
Project URL	https://www.acumos.org/
Source Location	https://gerrit.acumos.org/r/#/admin/projects/
Tag(s)	ML

2.8.2 AdaNet

AdaNet is a lightweight TensorFlow-based framework for automatically learning high-quality models with minimal expert intervention. AdaNet builds on recent AutoML efforts to be fast and flexible while providing learning guarantees. Importantly, AdaNet provides a general framework for not only learning a neural network architecture, but also for learning to ensemble to obtain even better models.

This project is based on the *AdaNet algorithm*, presented in "AdaNet: Adaptive Structural Learning of Artificial Neural Networks" at ICML 2017, for learning the structure of a neural network as an ensemble of subnetworks.

AdaNet has the following goals:

- *Ease of use*: Provide familiar APIs (e.g. Keras, Estimator) for training, evaluating, and serving models.
- *Speed*: Scale with available compute and quickly produce high quality models.
- *Flexibility*: Allow researchers and practitioners to extend AdaNet to novel subnetwork architectures, search spaces, and tasks.
- *Learning guarantees*: Optimize an objective that offers theoretical learning guarantees.

Documentation at https://adanet.readthedocs.io/en/latest/

SBB License	Apache License 2.0
Core Technology	Python
Project URL	https://adanet.readthedocs.io/en/latest/
Source Location	https://github.com/tensorflow/adanet
Tag(s)	ML

2.8.3 AllenNLP

An open-source NLP research library, built on PyTorch. AllenNLP is a NLP research library, built on PyTorch, for developing state-of-the-art deep learning models on a wide variety of linguistic tasks. AllenNLP makes it easy to design and evaluate new deep learning models for nearly any NLP problem, along with the infrastructure to easily run them in the cloud or on your laptop.

AllenNLP was designed with the following principles:

- *Hyper-modular and lightweight.* Use the parts which you like seamlessly with PyTorch.
- *Extensively tested and easy to extend.* Test coverage is above 90% and the example models provide a template for contributions.
- *Take padding and masking seriously*, making it easy to implement correct models without the pain.
- *Experiment friendly.* Run reproducible experiments from a json specification with comprehensive logging.

SBB License	Apache License 2.0
Core Technology	Python
Project URL	http://allennlp.org/
Source Location	https://github.com/allenai/allennlp
Tag(s)	ML, NLP, Python

2.8.4 Apache MXNet

Lightweight, Portable, Flexible Distributed/Mobile Deep Learning with Dynamic, Mutation-aware Dataflow Dep Scheduler; for Python, R, Julia, Scala, Go, Javascript and more.

All major GPU and CPU vendors support this project, but also the real giants like Amazon, Microsoft, Wolfram and a number of very respected universities. So watch this project or play with it to see if it fits your use case.

Apache MXNet (incubating) is a deep learning framework designed for both *efficiency* and *flexibility*. It allows you to **mix** symbolic and imperative programming to **maximize** efficiency and productivity. At its core, MXNet contains a dynamic dependency scheduler that automatically parallelizes both symbolic and imperative operations on the fly. A graph optimization layer on top of that makes symbolic execution fast and memory efficient. MXNet is portable and lightweight, scaling effectively to multiple GPUs and multiple machines.

MXNet is also more than a deep learning project. It is also a collection of blue prints and guidelines for building deep learning systems, and interesting insights of DL systems for hackers.

Gluon is the high-level interface for MXNet. It is more intuitive and easier to use than the lower level interface. Gluon supports dynamic (define-by-run) graphs with JIT-compilation to achieve both flexibility and efficiency. The perfect starters documentation with a great crash course on deep learning can be found here:http://gluon.mxnet.io/

Part of the project is also the the Gluon API specification (see https://github.com/gluon-api/gluon-api)

The Gluon API specification (Python based) is an effort to improve speed, flexibility, and accessibility of deep learning technology for all developers, regardless of their deep learning framework of choice. The Gluon API offers a flexible interface that simplifies the process of prototyping, building, and training deep learning models without sacrificing training speed.

SBB License	Apache License 2.0
Core Technology	CPP
Project URL	https://mxnet.apache.org/
Source Location	https://github.com/apache/incubator-mxnet
Tag(s)	ML

2.8.5 Apache Spark MLlib

Apache Spark MLlib. MLlib is Apache Spark's scalable machine learning library.

Apache Spark is a OSS platform for large-scale data processing. The Spark engine is written in Scala and is well suited for applications that reuse a working set of data across multiple parallel operations. It's designed to work as a standalone cluster or as part of Hadoop YARN cluster. It can access data from sources such as HDFS, Cassandra or Amazon S3. MLlib can be seen as a core Spark's APIs and interoperates with NumPy in Python and R libraries. And Spark is very fast!

MLlib library contains many algorithms and utilities, e.g.:

- Classification: logistic regression, naive Bayes,...
- Regression: generalized linear regression, survival regression,...
- Decision trees, random forests, and gradient-boosted trees
- Recommendation: alternating least squares (ALS)
- Clustering: K-means, Gaussian mixtures (GMMs),...
- Topic modeling: latent Dirichlet allocation (LDA)
- Frequent itemsets, association rules, and sequential pattern mining

SBB License	Apache License 2.0
Core Technology	Java
Project URL	https://spark.apache.org/mllib/
Source Location	https://github.com/apache/spark
Tag(s)	ML

2.8.6 Apollo

Apollo is a high performance, flexible architecture which accelerates the development, testing, and deployment of Autonomous Vehicles.

SBB License	GNU General Public License (GPL) 2.0
Core Technology	C++
Project URL	http://apollo.auto/
Source Location	https://github.com/ApolloAuto/apollo
Tag(s)	ML

2.8.7 auto_ml

Automated machine learning for analytics & production.

Automates the whole machine learning process, making it super easy to use for both analytics, and getting real-time predictions in production.

SBB License	MIT License
Core Technology	Python
Project URL	http://auto-ml.readthedocs.io
Source Location	https://github.com/ClimbsRocks/auto_ml
Tag(s)	ML

2.8.8 BigDL

BigDL is a distributed deep learning library for Apache Spark; with BigDL, users can write their deep learning applications as standard Spark programs, which can directly run on top of existing Spark or Hadoop clusters.

- **Rich deep learning support.** Modeled after Torch, BigDL provides comprehensive support for deep learning, including numeric computing (via Tensor) and high level neural networks; in addition, users can load pre-trained Caffe or Torch or Keras models into Spark programs using BigDL.
- **Extremely high performance.** To achieve high performance, BigDL uses Intel MKL and multi-threaded programming in each Spark task. Consequently, it is orders of magnitude faster than out-of-box open source Caffe, Torch or TensorFlow on a single-node Xeon (i.e., comparable with mainstream GPU).
- **Efficiently scale-out.** BigDL can efficiently scale out to perform data analytics at "Big Data scale", by leveraging Apache Spark (a lightning fast distributed data processing framework), as well as efficient implementations of synchronous SGD and all-reduce communications on Spark.

SBB License	Apache License 2.0
Core Technology	Java
Project URL	https://bigdl-project.github.io/master/
Source Location	https://github.com/intel-analytics/BigDL
Tag(s)	ML

2.8.9 Blocks

Blocks is a framework that is supposed to make it easier to build complicated neural network models on top of Theano.

Blocks is a framework that helps you build neural network models on top of Theano. Currently it supports and provides:

- Constructing parametrized Theano operations, called "bricks"

- Pattern matching to select variables and bricks in large models
- Algorithms to optimize your model
- Saving and resuming of training
- Monitoring and analyzing values during training progress (on the training set as well as on test sets)
- Application of graph transformations, such as dropout

SBB License	MIT License
Core Technology	Python
Project URL	http://blocks.readthedocs.io/en/latest/
Source Location	https://github.com/mila-udem/blocks
Tag(s)	ML

2.8.10 ConvNetJS

ConvNetJS is a Javascript library for training Deep Learning models (Neural Networks) entirely in your browser. Open a tab and you're training. No software requirements, no compilers, no installations, no GPUs, no sweat.

ConvNetJS is a Javascript implementation of Neural networks, together with nice browser-based demos. It currently supports:

- Common **Neural Network modules** (fully connected layers, non-linearities)
- Classification (SVM/Softmax) and Regression (L2) **cost functions**
- Ability to specify and train **Convolutional Networks** that process images
- An experimental **Reinforcement Learning** module, based on Deep Q Learning

For much more information, see the main page at convnetjs.com

Note: Not actively maintained, but still useful to prevent reinventing the wheel.

SBB License	MIT License
Core Technology	Javascript
Project URL	https://cs.stanford.edu/people/karpathy/convnetjs/
Source Location	https://github.com/karpathy/convnetjs
Tag(s)	Javascript, ML

2.8.11 Cookiecutter Data Science

A logical, reasonably standardized, but flexible project structure for doing and sharing data science work.

SBB License	MIT License
Core Technology	Python
Project URL	https://drivendata.github.io/cook iecutter-data-science/
Source Location	https://github.com/drivendata/coo kiecutter-data-science
Tag(s)	Data tool, ML

2.8.12 Data Science Version Control (DVC)

Data Science Version Control or **DVC** is an **open-source** tool for data science and machine learning projects. With a simple and flexible Git-like architecture and interface it helps data scientists:

1. manage **machine learning models** – versioning, including data sets and transformations (scripts) that were used to generate models;
2. make projects **reproducible**;
3. make projects **shareable**;
4. manage experiments with branching and **metrics** tracking;

It aims to replace tools like Excel and Docs that are being commonly used as a knowledge repo and a ledger for the team, ad-hoc scripts to track and move deploy different model versions, ad-hoc data file suffixes and prefixes.

SBB License	Apache License 2.0
Core Technology	Python
Project URL	https://dvc.org/
Source Location	https://github.com/iterative/dvc
Tag(s)	ML, Python

2.8.13 Dataexplorer

View, visualize, clean and process data in the browser.

Some features:

- Classic spreadsheet-style "grid" view
- Import CSV data from online
- Geocode data (convert "London" to longitude and latitude)
- Data and scripts automatically saved and accessible from anywhere
- "Fork" support – build on others work and let them build on yours

SBB License	MIT License
Core Technology	javascript
Project URL	http://explorer.okfnlabs.org
Source Location	https://github.com/okfn/dataexplorer
Tag(s)	Data viewer, ML

2.8.14 Datastream

An open-source framework for real-time anomaly detection using Python, ElasticSearch and Kiban. Also uses scikit-learn.

SBB License	Apache License 2.0
Core Technology	Python
Project URL	https://github.com/MentatInnovations/datastream.io
Source Location	https://github.com/MentatInnovations/datastream.io
Tag(s)	ML, Monitoring, Security

2.8.15 DeepDetect

DeepDetect implements support for supervised and unsupervised deep learning of images, text and other data, with focus on simplicity and ease of use, test and connection into existing applications. It supports classification, object detection, segmentation, regression, autoencoders and more.

It has Python and other client libraries.

Deep Detect has also a REST API for Deep Learning with:

- JSON communication format
- Pre-trained models
- Neural architecture templates
- Python, Java, C# clients
- Output templating

SBB License	MIT License
Core Technology	C++
Project URL	https://deepdetect.com
Source Location	https://github.com/beniz/deepdetect
Tag(s)	ML

2.8.16 Deeplearn.js

Deeplearn.js is an open-source library that brings performant machine learning building blocks to the web, allowing you to train neural networks in a browser or run pre-trained models in inference mode. And since Google is behind this project, a lot of eyes are targeted on this software. Deeplearn.js is an open source hardware accelerated implementation of deep learning APIs in the browser. So there is no need to download or install anything.

Deeplearn.js was originally developed by the Google Brain PAIR team to build powerful interactive machine learning tools for the browser.

SBB License	Apache License 2.0
Core Technology	Javascript
Project URL	https://deeplearnjs.org/
Source Location	https://github.com/PAIR-code/deeplearnjs
Tag(s)	Javascript, ML

2.8.17 Deeplearning4j

Deep Learning for Java, Scala & Clojure on Hadoop & Spark With GPUs.

Eclipse Deeplearning4J is an distributed neural net library written in Java and Scala.

Eclipse Deeplearning4j a commercial-grade, open-source, distributed deep-learning library written for Java and Scala. DL4J is designed to be used in business environments on distributed GPUs and CPUs.

Deeplearning4J integrates with Hadoop and Spark and runs on several backends that enable use of CPUs and GPUs. The aim of this project is to create a plug-and-play solution that is more convention than configuration, and which allows for fast prototyping. This project is created by Skymind who delivers support and offers also the option for machine learning models to be hosted with Skymind's model server on a cloud environment

SBB License	Apache License 2.0
Core Technology	Java
Project URL	https://deeplearning4j.org
Source Location	https://github.com/deeplearning4j/deeplearning4j
Tag(s)	ML

2.8.18 Detectron

Detectron is Facebook AI Research's software system that implements state-of-the-art object detection algorithms, including Mask R-CNN. It is written in Python and powered by the Caffe2 deep learning framework.

The goal of Detectron is to provide a high-quality, high-performance codebase for object detection *research*. It is designed to be flexible in order to support rapid implementation and evaluation of novel research.

A number of Facebook teams use this platform to train custom models for a variety of applications including augmented reality and community integrity. Once trained, these models can be deployed in the cloud and on mobile devices, powered by the highly efficient Caffe2 runtime.

SBB License	Apache License 2.0
Core Technology	Python
Project URL	https://github.com/facebookresearch/Detectron
Source Location	https://github.com/facebookresearch/Detectron
Tag(s)	AI, ML, Python

2.8.19 Dopamine

Dopamine is a research framework for fast prototyping of reinforcement learning algorithms. It aims to fill the need for a small, easily grokked codebase in which users can freely experiment with wild ideas (speculative research).

Our design principles are:

- *Easy experimentation*: Make it easy for new users to run benchmark experiments.
- *Flexible development*: Make it easy for new users to try out research ideas.
- *Compact and reliable*: Provide implementations for a few, battle-tested algorithms.
- *Reproducible*: Facilitate reproducibility in results.

SBB License	Apache License 2.0
Core Technology	Python
Project URL	https://github.com/google/dopamine
Source Location	https://github.com/google/dopamine
Tag(s)	ML, Reinforcement Learning

2.8.20 Fabrik

Fabrik is an online collaborative platform to build, visualize and train deep learning models via a simple drag-and-drop interface. It allows researchers to collaboratively develop and debug models using a web GUI that supports importing, editing and exporting networks written in widely popular frameworks like Caffe, Keras, and TensorFlow.

SBB License	GNU General Public License (GPL) 3.0
Core Technology	Javascript, Python
Project URL	http://fabrik.cloudcv.org/
Source Location	https://github.com/Cloud-CV/Fabrik
Tag(s)	Data Visualization, ML

2.8.21 Fastai

The fastai library simplifies training fast and accurate neural nets using modern best practices. Fast.ai's mission is to make the power of state of the art deep learning available to anyone. fastai sits on top of PyTorch, which provides the foundation.

Docs can be found on:http://docs.fast.ai/

SBB License	Apache License 2.0
Core Technology	Python
Project URL	http://www.fast.ai/
Source Location	https://github.com/fastai/fastai/
Tag(s)	ML

2.8.22 Featuretools

Featuretools is a python library for automated feature engineering. Featuretools can automatically create a single table of features for any "target entity". Featuretools is a framework to perform automated feature engineering. It excels at transforming transactional and relational datasets into feature matrices for machine learning.

SBB License	BSD License 2.0 (3-clause, New or Revised) License
Core Technology	Python
Project URL	https://www.featuretools.com/
Source Location	https://github.com/Featuretools/featuretools
Tag(s)	ML, Python

2.8.23 Featuretools

"One of the holy grails of machine learning is to automate more and more of the feature engineering process." — Pedro

Featuretools is a python library for automated feature engineering. Featuretools automatically creates features from temporal and relational datasets. Featuretools works alongside tools you already use to build machine learning pipelines. You can load in pandas dataframes and automatically create meaningful features in a fraction of the time it would take to do manually.

SBB License	BSD License 2.0 (3-clause, New or Revised) License
Core Technology	Python
Project URL	https://www.featuretools.com/
Source Location	https://github.com/Featuretools/featuretools
Tag(s)	ML

2.8.24 Flair

A very simple framework for **state-of-the-art NLP**. Developed by Zalando Research.

Flair is:

- **A powerful NLP library.** Flair allows you to apply our state-of-the-art natural language processing (NLP) models to your text, such as named entity recognition (NER), part-of-speech tagging (PoS), sense disambiguation and classification.
- **Multilingual.** Thanks to the Flair community, we support a rapidly growing number of languages. We also now include '*one model, many languages*' taggers, i.e. single models that predict PoS or NER tags for input text in various languages.
- **A text embedding library.** Flair has simple interfaces that allow you to use and combine different word and document embeddings, including our proposed Flair embeddings, BERT embeddings and ELMo embeddings.
- **A Pytorch NLP framework.** Our framework builds directly on Pytorch, making it easy to train your own models and experiment with new approaches using Flair embeddings and classes.

SBB License	MIT License
Core Technology	Python
Project URL	https://github.com/zalandoresearch/flair
Source Location	https://github.com/zalandoresearch/flair
Tag(s)	ML, NLP, Python

2.8.25 Fuel

Fuel is a data pipeline framework which provides your machine learning models with the data they need. It is planned to be used by both the Blocks and Pylearn2 neural network libraries.

- Fuel allows you to easily read different types of data (NumPy binary files, CSV files, HDF5 files, text files) using a single interface which is based on Python's iterator types.
- Provides a a series of wrappers around frequently used datasets such as MNIST, CIFAR-10 (vision), the One Billion Word Dataset (text corpus), and many more.
- Allows you iterate over data in a variety of ways, e.g. in order, shuffled, sampled, etc.
- Gives you the possibility to process your data on-the-fly through a series of (chained) transformation procedures. This way you can whiten your data, noise, rotate, crop, pad, sort or shuffle, cache it, and much more.
- Is pickle-friendly, allowing you to stop and resume long-running experiments in the middle of a pass over your dataset without losing any training progress.

SBB License	MIT License
Core Technology	Python
Project URL	http://fuel.readthedocs.io/en/latest/index.html
Source Location	https://github.com/mila-udem/fuel
Tag(s)	Data tool, ML

2.8.26 Gensim

Gensim is a Python library for *topic modelling*, *document indexing* and *similarity retrieval* with large corpora. Target audience is the *natural language processing* (NLP) and *information retrieval* (IR) community.

SBB License	MIT License
Core Technology	Python
Project URL	https://github.com/RaRe-Technologies/gensim
Source Location	https://github.com/RaRe-Technologies/gensim
Tag(s)	ML, NLP, Python

2.8.27 Golem

The aim of the Golem project is to create a global prosumer market for computing power, in which producers may sell spare CPU time of their personal computers and consumers may acquire resources for computation-intensive tasks. In technical terms, Golem is designed as a decentralised peer-to-peer network established by nodes running the Golem client software. For the purpose of this paper we assume that there are two types of nodes in the Golem network: requestor nodes that announce computing tasks and compute nodes that perform computations (in the actual implementation nodes may switch between both roles).

SBB License	GNU General Public License (GPL) 3.0
Core Technology	Python
Project URL	https://golem.network/
Source Location	https://github.com/golemfactory/golem
Tag(s)	Distributed Computing, ML

2.8.28 HyperTools

HyperTools is a library for visualizing and manipulating high-dimensional data in Python. It is built on top of matplotlib (for plotting), seaborn (for plot styling), and scikit-learn (for data manipulation).

Some key features of HyperTools are:

1. Functions for plotting high-dimensional datasets in 2/3D
2. Static and animated plots
3. Simple API for customizing plot styles
4. Set of powerful data manipulation tools including hyperalignment, k-means clustering, normalizing and more
5. Support for lists of Numpy arrays or Pandas dataframes

SBB License	MIT License
Core Technology	Python
Project URL	http://hypertools.readthedocs.io/en/latest/
Source Location	https://github.com/ContextLab/hypertools
Tag(s)	Data tool, ML

2.8.29 JeelizFaceFilter

Javascript/WebGL lightweight face tracking library designed for augmented reality webcam filters. Features : multiple faces detection, rotation, mouth opening. Various integration examples are provided (Three.js, Babylon.js, FaceSwap, Canvas2D, CSS3D. . .).

Enables developers to solve computer-vision problems directly from the browser.

Features:

- face detection,
- face tracking,
- face rotation detection,
- mouth opening detection,
- multiple faces detection and tracking,
- very robust for all lighting conditions,
- video acquisition with HD video ability,
- interfaced with 3D engines like THREE.JS, BABYLON.JS, A-FRAME,
- interfaced with more accessible APIs like CANVAS, CSS3D.

SBB License	Apache License 2.0
Core Technology	Javascript
Project URL	https://jeeliz.com/
Source Location	https://github.com/jeeliz/jeelizFaceFilter
Tag(s)	face detection, Javascript, ML

2.8.30 Keras

Keras is a high-level neural networks API, written in Python and capable of running on top of TensorFlow, CNTK, or Theano. It was developed with a focus on enabling fast experimentation. Being able to go from idea to result with the least possible delay is key to doing good research.

Use Keras if you need a deep learning library that:

- Allows for easy and fast prototyping (through user friendliness, modularity, and extensibility).
- Supports both convolutional networks and recurrent networks, as well as combinations of the two.

- Runs seamlessly on CPU and GPU.

SBB License	MIT License
Core Technology	Python
Project URL	https://keras.io/
Source Location	https://github.com/keras-team/keras
Tag(s)	ML

2.8.31 Klassify

Redis based text classification service with real-time web interface.

What is Text Classification: Text classification, document classification or document categorization is a problem in library science, information science and computer science. The task is to assign a document to one or more classes or categories.

SBB License	MIT License
Core Technology	Python
Project URL	https://github.com/fatiherikli/klassify
Source Location	https://github.com/fatiherikli/klassify
Tag(s)	ML, Text classification

2.8.32 Lore

Lore is a python framework to make machine learning approachable for Engineers and maintainable for Data Scientists.

Features

- Models support hyper parameter search over estimators with a data pipeline. They will efficiently utilize multiple GPUs (if available) with a couple different strategies, and can be saved and distributed for horizontal scalability.
- Estimators from multiple packages are supported: Keras (TensorFlow/Theano/CNTK), XGBoost and SciKit Learn. They can all be subclassed with build, fit or predict overridden to completely customize your algorithm and architecture, while still benefiting from everything else.
- Pipelines avoid information leaks between train and test sets, and one pipeline allows experimentation with many different estimators. A disk based pipeline is available if you exceed your machines available RAM.
- Transformers standardize advanced feature engineering. For example, convert an American first name to its statistical age or gender using US Census data. Extract the geographic area code from a free form phone number string. Common date, time and string operations are supported efficiently through pandas.
- Encoders offer robust input to your estimators, and avoid common problems with missing and long tail values. They are well tested to save you from garbage in/garbage out.

- IO connections are configured and pooled in a standard way across the app for popular (no)sql databases, with transaction management and read write optimizations for bulk data, rather than typical ORM single row operations. Connections share a configurable query cache, in addition to encrypted S3 buckets for distributing models and datasets.
- Dependency Management for each individual app in development, that can be 100% replicated to production. No manual activation, or magic env vars, or hidden files that break python for everything else. No knowledge required of venv, pyenv, pyvenv, virtualenv, virtualenvwrapper, pipenv, conda. Ain't nobody got time for that.
- Tests for your models can be run in your Continuous Integration environment, allowing Continuous Deployment for code and training updates, without increased work for your infrastructure team.
- Workflow Support whether you prefer the command line, a python console, jupyter notebook, or IDE. Every environment gets readable logging and timing statements configured for both production and development.

SBB License	GNU General Public License (GPL) 2.0
Core Technology	Python
Project URL	https://github.com/instacart/lore
Source Location	https://github.com/instacart/lore
Tag(s)	ML, Python

2.8.33 Ludwig

Ludwig is a toolbox built on top of TensorFlow that allows to train and test deep learning models without the need to write code. Ludwig provides two main functionalities: training models and using them to predict. It is based on datatype abstraction, so that the same data preprocessing and postprocessing will be performed on different datasets that share data types and the same encoding and decoding models developed for one task can be reused for different tasks.

All you need to provide is a CSV file containing your data, a list of columns to use as inputs, and a list of columns to use as outputs, Ludwig will do the rest. Simple commands can be used to train models both locally and in a distributed way, and to use them to predict on new data.

A programmatic API is also available in order to use Ludwig from your python code. A suite of visualization tools allows you to analyze models' training and test performance and to compare them.

Ludwig is built with extensibility principles in mind and is based on data type abstractions, making it easy to add support for new data types as well as new model architectures.

It can be used by practitioners to quickly train and test deep learning models as well as by researchers to obtain strong baselines to compare against and have an experimentation setting that ensures comparability by performing standard data preprocessing and visualization.

SBB License	Apache License 2.0
Core Technology	Python
Project URL	https://uber.github.io/ludwig/
Source Location	https://github.com/uber/ludwig
Tag(s)	ML

2.8.34 Luminoth

Luminoth is an open source toolkit for computer vision. Currently, we support object detection and image classification, but we are aiming for much more. It is built in Python, using TensorFlow and Sonnet.

SBB License	BSD License 2.0 (3-clause, New or Revised) License
Core Technology	Python
Project URL	https://luminoth.ai
Source Location	https://github.com/tryolabs/luminoth
Tag(s)	ML

2.8.35 MacroBase

MacroBase is a new analytic monitoring engine designed to prioritize human attention in large-scale datasets and data streams. Unlike a traditional analytics engine, MacroBase is specialized for one task: finding and explaining unusual or interesting trends in data. Developed by Stanford Future Data Systems

Documentation can be found at: https://macrobase.stanford.edu/docs/

SBB License	Apache License 2.0
Core Technology	Java
Project URL	https://macrobase.stanford.edu/
Source Location	https://github.com/stanford-futur edata/macrobase/tree/v1.0
Tag(s)	Data analytics, ML

2.8.36 ml5.js

ml5.js aims to make machine learning approachable for a broad audience of artists, creative coders, and students. The library provides access to machine learning algorithms and models in the browser, building on top of TensorFlow.js with no other external dependencies.

The library is supported by code examples, tutorials, and sample data sets with an emphasis on ethical computing. Bias in data, stereotypical harms, and responsible crowdsourcing are part of the documentation around data collection and usage.

ml5.js is heavily inspired by Processing and p5.js.

SBB License	MIT License
Core Technology	Javascript
Project URL	https://ml5js.org/
Source Location	https://github.com/ml5js/ml5-library
Tag(s)	Javascript, ML

2.8.37 MLflow

MLflow offers a way to simplify ML development by making it easy to track, reproduce, manage, and deploy models. MLflow (currently in alpha) is an open source platform designed to manage the entire machine learning lifecycle and work with any machine learning library. It offers:

- Record and query experiments: code, data, config, results
- Packaging format for reproducible runs on any platform
- General format for sending models to diverse deploy tools

SBB License	Apache License 2.0
Core Technology	Python
Project URL	https://mlflow.org/
Source Location	https://github.com/databricks/mlflow
Tag(s)	ML, Python

2.8.38 Mljar

MLJAR is a platform for rapid prototyping, developing and deploying machine learning models.

MLJAR makes algorithm search and tuning painless. It checks many different algorithms for you. For each algorithm hyper-parameters are separately tuned. All computations run in parallel in MLJAR cloud, so you get your results very quickly. At the end the ensemble of models is created, so your predictive model will be super accurate.

There are two types of interface available in MLJAR:

- you can run Machine Learning models in your browser, you don't need to code anything. Just upload dataset, click which attributes to use, which algorithms to use and go! This makes Machine Learning super easy for everyone and make it possible to get really useful models,
- there is a python wrapper over MLJAR API, so you don't need to open any browser or click on any button, just write fancy python code! We like it and hope you will like it too! To start using MLJAR python package please go to our github.

SBB License	MIT License
Core Technology	Python
Project URL	https://mljar.com/
Source Location	https://github.com/mljar/mljar-supervised
Tag(s)	ML, Python

2.8.39 MLPerf

A broad ML benchmark suite for measuring performance of ML software frameworks, ML hardware accelerators, and ML cloud platforms.

The MLPerf effort aims to build a common set of benchmarks that enables the machine learning (ML) field to measure system performance for both training and inference from mobile devices to cloud services. We believe that a widely accepted benchmark suite will benefit the entire community, including researchers, developers, builders of machine learning frameworks, cloud service providers, hardware manufacturers, application providers, and end users.

SBB License	MIT License
Core Technology	Python
Project URL	https://mlperf.org/
Source Location	https://github.com/mlperf/reference
Tag(s)	ML, Performance

2.8.40 ModelDB

A system to manage machine learning models.

ModelDB is an end-to-end system to manage machine learning models. It ingests models and associated metadata as models are being trained, stores model data in a structured format, and surfaces it through a web-frontend for rich querying. ModelDB can be used with any ML environment via the ModelDB Light API. ModelDB native clients can be used for advanced support in spark.ml and scikit-learn.

The ModelDB frontend provides rich summaries and graphs showing model data. The frontend provides functionality to slice and dice this data along various attributes (e.g. operations like filter by hyperparameter, group by datasets) and to build custom charts showing model performance.

SBB License	MIT License
Core Technology	Python, Javascript
Project URL	https://mitdbg.github.io/modeldb/
Source Location	https://github.com/mitdbg/modeldb
Tag(s)	administration, ML

2.8.41 Netron

Netron is a viewer for neural network, deep learning and machine learning models.

Netron supports ONNX (`.onnx`, `.pb`), **Keras** (`.h5`, `.keras`), **CoreML** (`.mlmodel`) and **TensorFlow Lite** (`.tflite`). Netron has experimental support for **Caffe** (`.caffemodel`), **Caffe2** (`predict_net.pb`), **MXNet** (`-symbol.json`), **TensorFlow.js** (`model.json`, `.pb`) and **TensorFlow** (`.pb`, `.meta`).

SBB License	GNU General Public License (GPL) 2.0
Core Technology	Python, Javascript
Project URL	https://www.lutzroeder.com/ai/
Source Location	https://github.com/lutzroeder/Netron
Tag(s)	Data viewer, ML

2.8.42 Neuralcoref

State-of-the-art coreference resolution based on neural nets and spaCy.

NeuralCoref is a pipeline extension for spaCy 2.0 that annotates and resolves coreference clusters using a neural network. NeuralCoref is production-ready, integrated in spaCy's NLP pipeline and easily extensible to new training datasets.

SBB License	MIT License
Core Technology	Python
Project URL	https://huggingface.co/coref/
Source Location	https://github.com/huggingface/neuralcoref
Tag(s)	ML, NLP, Python

2.8.43 NLP Architect

NLP Architect is an open-source Python library for exploring the state-of-the-art deep learning topologies and techniques for natural language processing and natural language understanding. It is intended to be a platform for future research and collaboration.

How can NLP Architect be used:

- Train models using provided algorithms, reference datasets and configurations
- Train models using your own data
- Create new/extend models based on existing models or topologies
- Explore how deep learning models tackle various NLP tasks
- Experiment and optimize state-of-the-art deep learning algorithms
- integrate modules and utilities from the library to solutions

SBB License	Apache License 2.0
Core Technology	Python
Project URL	http://nlp_architect.nervanasys.com/
Source Location	https://github.com/NervanaSystems/nlp-architect
Tag(s)	ML, NLP, Python

2.8.44 NNI (Neural Network Intelligence)

NNI (Neural Network Intelligence) is a toolkit to help users run automated machine learning (AutoML) experiments. The tool dispatches and runs trial jobs generated by tuning algorithms to search the best neural architecture and/or hyper-parameters in different environments like local machine, remote servers and cloud. (Microsoft ML project)

Who should consider using NNI:

- Those who want to try different AutoML algorithms in their training code (model) at their local machine.
- Those who want to run AutoML trial jobs in different environments to speed up search (e.g. remote servers and cloud).
- Researchers and data scientists who want to implement their own AutoML algorithms and compare it with other algorithms.
- ML Platform owners who want to support AutoML in their platform.

SBB License	MIT License
Core Technology	Python
Project URL	https://nni.readthedocs.io/en/latest/
Source Location	https://github.com/Microsoft/nni
Tag(s)	ML

2.8.45 ONNX

ONNX provides an open source format for AI models. It defines an extensible computation graph model, as well as definitions of built-in operators and standard data types. Initially we focus on the capabilities needed for inferencing (evaluation).

Caffe2, PyTorch, Microsoft Cognitive Toolkit, Apache MXNet and other tools are developing ONNX support. Enabling interoperability between different frameworks and streamlining the path from research to production will increase the speed of innovation in the AI community. We are an early stage and we invite the community to submit feedback and help us further evolve ONNX.

Companies behind ONNX are AWS, Facebook and Microsoft Corporation and more.

SBB License	MIT License
Core Technology	Python
Project URL	http://onnx.ai/
Source Location	https://github.com/onnx/onnx
Tag(s)	AI, ML

2.8.46 OpenCV: Open Source Computer Vision Library

OpenCV (Open Source Computer Vision Library) is an open source computer vision and machine learning software library. OpenCV was built to provide a common infrastructure for computer vision applications and to accelerate the use of machine perception in the commercial products. Being a BSD-licensed product, OpenCV makes it easy for businesses to utilize and modify the code.

The library has more than 2500 optimized algorithms, which includes a comprehensive set of both classic and state-of-the-art computer vision and machine learning algorithms. These algorithms can be used to detect and recognize faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D models of objects, produce 3D point clouds from stereo cameras, stitch images together to produce a high resolution

image of an entire scene, find similar images from an image database, remove red eyes from images taken using flash, follow eye movements, recognize scenery and establish markers to overlay it with augmented reality, etc.

SBB License	BSD License 2.0 (3-clause, New or Revised) License
Core Technology	C
Project URL	https://opencv.org/
Source Location	https://github.com/opencv/opencv
Tag(s)	ML

2.8.47 OpenML

OpenML is an on-line machine learning platform for sharing and organizing data, machine learning algorithms and experiments. It claims to be designed to create a frictionless, networked ecosystem, so that you can readily integrate into your existing processes/code/environments. It also allows people from all over the world to collaborate and build directly on each other's latest ideas, data and results, irrespective of the tools and infrastructure they happen to use. So nice ideas to build an open science movement. The people behind OpemML are mostly (data)scientist. So using this product for real world business use cases will take some extra effort.

Altrhough OpenML is exposed as an foundation based on openness, a quick inspection learned that the OpenML platform is not as open as you want. Also the OSS software is not created to be run on premise. So be aware when doing large (time) investments into this OpenML platform.

SBB License	BSD License 2.0 (3-clause, New or Revised) License
Core Technology	Java
Project URL	https://openml.org
Source Location	https://github.com/openml/OpenML
Tag(s)	ML

2.8.48 Orange

Orange is a comprehensive, component-based software suite for machine learning and data mining, developed at Bioinformatics Laboratory.

Orange is available by default on Anaconda Navigator dashboard. Orange is a component-based data mining software. It includes a range of data visualization, exploration, preprocessing and modeling techniques. It can be used through a nice and intuitive user interface or, for more advanced users, as a module for the Python programming language.

One of the nice features is the option for visual programming. Can you do visual interactive data exploration for rapid qualitative analysis with clean visualizations. The graphic user interface allows you to focus on exploratory data analysis instead of coding, while clever defaults make fast prototyping of a data analysis workflow extremely easy.

SBB License	GNU General Public License (GPL) 3.0
Core Technology	
Project URL	https://orange.biolab.si/
Source Location	https://github.com/biolab/orange3
Tag(s)	Data Visualization, ML, Python

2.8.49 Pattern

Pattern is a web mining module for Python. It has tools for:

- Data Mining: web services (Google, Twitter, Wikipedia), web crawler, HTML DOM parser
- Natural Language Processing: part-of-speech taggers, n-gram search, sentiment analysis, WordNet
- Machine Learning: vector space model, clustering, classification (KNN, SVM, Perceptron)
- Network Analysis: graph centrality and visualization.

SBB License	BSD License 2.0 (3-clause, New or Revised) License
Core Technology	Python
Project URL	https://www.clips.uantwerpen.be/pages/pattern
Source Location	https://github.com/clips/pattern
Tag(s)	ML, NLP, Web scraping

2.8.50 Plait

plait.py is a program for generating fake data from composable yaml templates.

With plait it is easy to model fake data that has an interesting shape. Currently, many fake data generators model their data as a collection of IID variables; with plait.py we can stitch together those variables into a more coherent model.

Example uses for plait.py are:

- generating mock application data in test environments
- validating the usefulness of statistical techniques
- creating synthetic datasets for performance tuning databases

SBB License	MIT License
Core Technology	Python
Project URL	https://github.com/plaitpy/plaitpy
Source Location	https://github.com/plaitpy/plaitpy
Tag(s)	Data Generator, ML, text generation

2.8.51 Polyaxon

An open source platform for reproducible machine learning at scale.

Polyaxon is a platform for building, training, and monitoring large scale deep learning applications.

Polyaxon deploys into any data center, cloud provider, or can be hosted and managed by Polyaxon, and it supports all the major deep learning frameworks such as Tensorflow, MXNet, Caffe, Torch, etc.

Polyaxon makes it faster, easier, and more efficient to develop deep learning applications by managing workloads with smart container and node management. And it turns GPU servers into shared, self-service resources for your team or organization.

SBB License	MIT License
Core Technology	Python
Project URL	https://polyaxon.com/
Source Location	https://github.com/polyaxon/polyaxon
Tag(s)	ML

2.8.52 Pylearn2

Pylearn2 is a library designed to make machine learning research easy.

SBB License	BSD License 2.0 (3-clause, New or Revised) License
Core Technology	Python
Project URL	http://deeplearning.net/software/pylearn2/
Source Location	https://github.com/lisa-lab/pylearn2
Tag(s)	ML

2.8.53 Pyro

Deep universal probabilistic programming with Python and PyTorch. Pyro is in an alpha release. It is developed and used byUber AI Labs.

SBB License	GNU General Public License (GPL) 2.0
Core Technology	Python
Project URL	http://pyro.ai/
Source Location	https://github.com/uber/pyro
Tag(s)	AI, ML, Python

2.8.54 PyTorch

PyTorch is:

- a deep learning framework that puts Python first.
- a research-focused framework.
- Python package that provides two high-level features:

Pytorch uses tensor computation (like NumPy) with strong GPU acceleration. It can use deep neural networks built on a tape-based autograd system.

You can reuse your favorite Python packages such as NumPy, SciPy and Cython to extend PyTorch when needed.

Note: PyTorch is still in an early-release beta phase (status January 2018). PyTorch was released as OSS by Google January 2017.

SBB License	MIT License
Core Technology	Python
Project URL	http://pytorch.org/
Source Location	https://github.com/pytorch/pytorch
Tag(s)	AI, ML

2.8.55 Rant

Rant is an all-purpose procedural text engine that is most simply described as the opposite of Regex. It has been refined to include a dizzying array of features for handling everything from the most basic of string generation tasks to advanced dialogue generation, code templating, automatic formatting, and more.

The goal of the project is to enable developers of all kinds to automate repetitive writing tasks with a high degree of creative freedom.

Features:

- Recursive, weighted branching with several selection modes
- Queryable dictionaries
- Automatic capitalization, rhyming, English indefinite articles, and multi-lingual number verbalization
- Print to multiple separate outputs
- Probability modifiers for pattern elements
- Loops, conditional statements, and subroutines
- Fully-functional object model
- Import/Export resources easily with the .rantpkg format
- Compatible with Unity 2017

SBB License	MIT License
Core Technology	.NET
Project URL	https://berkin.me/rant/
Source Location	https://github.com/TheBerkin/rant
Tag(s)	.NET, ML, NLP, text generation

2.8.56 RAPIDS

The RAPIDS suite of software libraries gives you the freedom to execute end-to-end data science and analytics pipelines entirely on GPUs. It relies on NVIDIA® CUDA® primitives for low-level compute optimization, but exposes that GPU parallelism and high-bandwidth memory speed through user-friendly Python interfaces.

RAPIDS also focuses on common data preparation tasks for analytics and data science. This includes a familiar DataFrame API that integrates with a variety of machine learning algorithms for end-to-end pipeline accelerations without paying typical serialization costs–. RAPIDS also includes support for multi-node, multi-GPU deployments, enabling vastly accelerated processing and training on much larger dataset sizes.

SBB License	Apache License 2.0
Core Technology	C++
Project URL	http://rapids.ai/
Source Location	https://github.com/rapidsai/
Tag(s)	ML

2.8.57 Ray

Ray is a flexible, high-performance distributed execution framework for AI applications. Ray is currently under heavy development. But Ray has already a good start, with good documentation (http://ray.readthedocs.io/en/latest/index.html) and a tutorial. Also Ray is backed by scientific researchers and published papers.

Ray comes with libraries that accelerate deep learning and reinforcement learning development:

- Ray Tune: Hyperparameter Optimization Framework
- Ray RLlib: A Scalable Reinforcement Learning Library

SBB License	Apache License 2.0
Core Technology	Python
Project URL	https://ray-project.github.io/
Source Location	https://github.com/ray-project/ray
Tag(s)	ML

2.8.58 Scikit-learn

scikit-learn is a Python module for machine learning.

Simple and efficient tools for data mining and data analysis

- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib

SBB License	BSD License 2.0 (3-clause, New or Revised) License
Core Technology	Python
Project URL	http://scikit-learn.org
Source Location	https://github.com/scikit-learn/scikit-learn
Tag(s)	ML

2.8.59 Skater

Skater is a python package for model agnostic interpretation of predictive models. With Skater, you can unpack the internal mechanics of arbitrary models; as long as you can obtain inputs, and use a function to obtain outputs, you can use Skater to learn about the models internal decision policies.

The project was started as a research idea to find ways to enable better interpretability(preferably human interpretability) to predictive "black boxes" both for researchers and practioners.

Documentation at:https://datascienceinc.github.io/Skater/overview.html

SBB License	MIT License
Core Technology	Python
Project URL	https://www.datascience.com/resources/tools/skater
Source Location	https://github.com/datascienceinc/Skater
Tag(s)	ML

2.8.60 Snorkel

Snorkel is a system for rapidly **creating, modeling, and managing training data**, currently focused on accelerating the development of *structured or "dark" data extraction applications* for domains in which large labeled training sets are not available or easy to obtain.

SBB License	Apache License 2.0
Core Technology	Python
Project URL	https://hazyresearch.github.io/snorkel/
Source Location	https://github.com/HazyResearch/snorkel
Tag(s)	ML

2.8.61 Tensorflow

TensorFlow is an Open Source Software Library for Machine Intelligence. TensorFlow is by far the most used and popular ML open source project. And since the first initial release was only just in November 2015 it is expected that the impact of this OSS package will expand even more.

TensorFlow™ is an open source software library for numerical computation using data flow graphs. Nodes in the graph represent mathematical operations, while the graph edges represent the multidimensional data arrays (tensors) communicated between them. The flexible architecture allows you to deploy computation to one or more CPUs or GPUs in a desktop, server, or mobile device with a single API. TensorFlow was originally developed by researchers and engineers working on the Google Brain Team within Google's Machine Intelligence research organization for the purposes of conducting machine learning and deep neural networks research, but the system is general enough to be applicable in a wide variety of other domains as well.

TensorFlow comes with a tool called TensorBoard which you can use to get some insight into what is happening. TensorBoard is a suite of web applications for inspecting and understanding your TensorFlow runs and graphs.

There is also a version of TensorFlow that runs in a browser. This is TensorFlow.js (https://js.tensorflow.org/). TensorFlow.js is a WebGL accelerated, browser based JavaScript library for training and deploying ML models.

SBB License	Apache License 2.0
Core Technology	C
Project URL	https://www.tensorflow.org/
Source Location	https://github.com/tensorflow/tensorflow
Tag(s)	AI, ML

2.8.62 TextBlob: Simplified Text Processing

TextBlob is a Python (2 and 3) library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more.

2.8.63 Features

- Noun phrase extraction
- Part-of-speech tagging
- Sentiment analysis
- Classification (Naive Bayes, Decision Tree)
- Language translation and detection powered by Google Translate
- Tokenization (splitting text into words and sentences)
- Word and phrase frequencies
- Parsing

- n-grams
- Word inflection (pluralization and singularization) and lemmatization
- Spelling correction
- Add new models or languages through extensions
- WordNet integration

SBB License	MIT License
Core Technology	Python
Project URL	https://textblob.readthedocs.io/en/dev/
Source Location	https://github.com/sloria/textblob
Tag(s)	ML, NLP, Python

2.8.64 Theano

Theano is a Python library that allows you to define, optimize, and evaluate mathematical expressions involving multi-dimensional arrays efficiently. It can use GPUs and perform efficient symbolic differentiation.

Note: After almost ten years of development the company behind Theano has stopped development and support(Q4-2017). But this library has been an innovation driver for many other OSS ML packages!

Since a lot of ML libraries and packages use Theano you should check (as always) the health of your ML stack.

SBB License	MIT License
Core Technology	Python
Project URL	http://www.deeplearning.net/
Source Location	https://github.com/Theano/Theano
Tag(s)	ML, Python

2.8.65 Thinc

Thinc is the machine learning library powering spaCy. It features a battle-tested linear model designed for large sparse learning problems, and a flexible neural network model under development for spaCy v2.0.

Thinc is a practical toolkit for implementing models that follow the "Embed, encode, attend, predict" architecture. It's designed to be easy to install, efficient for CPU usage and optimised for NLP and deep learning with text – in particular, hierarchically structured input and variable-length sequences.

SBB License	GNU General Public License (GPL) 2.0
Core Technology	Python
Project URL	https://explosion.ai/
Source Location	https://github.com/explosion/thinc
Tag(s)	ML, NLP, Python

2.8.66 Turi

Turi Create simplifies the development of custom machine learning models.Turi is OSS machine learning from Apple.

Turi Create simplifies the development of custom machine learning models. You don't have to be a machine learning expert to add recommendations, object detection, image classification, image similarity or activity classification to your app.

SBB License	BSD License 2.0 (3-clause, New or Revised) License
Core Technology	Python
Project URL	https://github.com/apple/turicreate
Source Location	https://github.com/apple/turicreate
Tag(s)	ML

2.8.67 TuriCreate

This SBB is from Apple. Apple, is with Siri already for a long time active in machine learning. But even Apple is releasing building blocks under OSS licenses now.

Turi Create simplifies the development of custom machine learning models. You don't have to be a machine learning expert to add recommendations, object detection, image classification, image similarity or activity classification to your app.

- **Easy-to-use:** Focus on tasks instead of algorithms
- **Visual:** Built-in, streaming visualizations to explore your data
- **Flexible:** Supports text, images, audio, video and sensor data
- **Fast and Scalable:** Work with large datasets on a single machine
- **Ready To Deploy:** Export models to Core ML for use in iOS, macOS, watchOS, and tvOS apps

SBB License	BSD License 2.0 (3-clause, New or Revised) License
Core Technology	Python
Project URL	https://turi.com/index.html
Source Location	https://github.com/apple/turicreate
Tag(s)	ML, Python

2.8.68 VisualDL

VisualDL is an open-source cross-framework web dashboard that richly visualizes the performance and data flowing through your neural network training. VisualDL is a deep learning visualization tool that can help design deep learning jobs. It includes features such as scalar, parameter distribution, model structure and image visualization.

SBB License	Apache License 2.0
Core Technology	C++
Project URL	http://visualdl.paddlepaddle.org/
Source Location	https://github.com/PaddlePaddle/VisualDL
Tag(s)	ML

2.8.69 What-If Tool

The What-If Tool (WIT) provides an easy-to-use interface for expanding understanding of a black-box ML model. With the plugin, you can perform inference on a large set of examples and immediately visualize the results in a variety of ways. Additionally, examples can be edited manually or programatically and re-run through the model in order to see the results of the changes. It contains tooling for investigating model performance and fairness over subsets of a dataset.

The purpose of the tool is that give people a simple, intuitive, and powerful way to play with a trained ML model on a set of data through a visual interface with absolutely no code required.

SBB License	Apache License 2.0
Core Technology	Python
Project URL	https://pair-code.github.io/what- if-tool/
Source Location	https://github.com/tensorflow/tensorboard/tree/master/tensorboard/plugins/interactive_inference
Tag(s)	ML

2.8.70 XAI

XAI is a Machine Learning library that is designed with AI explainability in its core. XAI contains various tools that enable for analysis and evaluation of data and models. The XAI library is maintained by The Institute for Ethical AI & ML, and it was developed based on the 8 principles for Responsible Machine Learning.

You can find the documentation at https://ethicalml.github.io/xai/index.html.

SBB License	MIT License
Core Technology	Python
Project URL	https://ethical.institute/index.html
Source Location	https://github.com/EthicalML/xai
Tag(s)	ML, Python

End of SBB list

2.9 Catalogue of Open NLP Software

Todo: The OSS NLP list will be completed, filtered, adjusted and corrected soon! Want to help?

2.9.1 AllenNLP

An open-source NLP research library, built on PyTorch. AllenNLP is a NLP research library, built on PyTorch, for developing state-of-the-art deep learning models on a wide variety of linguistic tasks. AllenNLP makes it easy to design and evaluate new deep learning models for nearly any NLP problem, along with the infrastructure to easily run them in the cloud or on your laptop.

AllenNLP was designed with the following principles:

- *Hyper-modular and lightweight.* Use the parts which you like seamlessly with PyTorch.
- *Extensively tested and easy to extend.* Test coverage is above 90% and the example models provide a template for contributions.
- *Take padding and masking seriously*, making it easy to implement correct models without the pain.
- *Experiment friendly.* Run reproducible experiments from a json specification with comprehensive logging.

SBB License	Apache License 2.0
Core Technology	Python
Project URL	http://allennlp.org/
Source Location	https://github.com/allenai/allennlp
Tag(s)	ML, NLP, Python

2.9.2 Apache OpenNLP

The Apache OpenNLP library is a machine learning based toolkit for the processing of natural language text.

The Apache OpenNLP library is a machine learning based toolkit for the processing of natural language text. It supports the most common NLP tasks, such as tokenization, sentence segmentation, part-of-speech tagging, named entity extraction, chunking, parsing, and coreference resolution. These tasks are usually required to build more advanced text processing services. OpenNLP also included maximum entropy and perceptron based machine learning.

The goal of the OpenNLP project will be to create a mature toolkit for the abovementioned tasks. An additional goal is to provide a large number of pre-built models for a variety of languages, as well as the annotated text resources that those models are derived from.

SBB License	Apache License 2.0
Core Technology	Java
Project URL	http://opennlp.apache.org/
Source Location	http://opennlp.apache.org/source-code.html
Tag(s)	NLP

2.9.3 Apache Tika

The Apache Tika™ toolkit detects and extracts metadata and text from over a thousand different file types (such as PPT, XLS, and PDF). All of these file types can be parsed through a single interface, making Tika useful for search engine indexing, content analysis, translation, and much more.

Several wrappers are available to use Tika in another programming language, such as Julia or Python

SBB License	Apache License 2.0
Core Technology	Java
Project URL	https://tika.apache.org/
Source Location	https://tika.apache.org/
Tag(s)	NLP

2.9.4 Bling Fire

A lightning fast Finite State machine and REgular expression manipulation library. Bling Fire Tokenizer is a tokenizer designed for fast-speed and quality tokenization of Natural Language text. It mostly follows the tokenization logic of NLTK, except hyphenated words are split and a few errors are fixed.

SBB License	MIT License
Core Technology	CPP
Project URL	https://github.com/Microsoft/BlingFire
Source Location	https://github.com/Microsoft/BlingFire
Tag(s)	NLP

2.9.5 ERNIE

An Implementation of ERNIE For Language Understanding (including Pre-training models and Fine-tuning tools)

'ERNIE 2.0 <https://arxiv.org/abs/1907.12412v1>'__ is a continual pre-training framework for language understanding in which pre-training tasks can be incrementally built and learned through multi-task learning. In this framework, different customized tasks can be incrementally introduced at any time. For example, the tasks including named entity prediction, discourse relation recognition, sentence order prediction are leveraged in order to enable the models to learn language representations.

SBB License	Apache License 2.0
Core Technology	Python
Project URL	https://github.com/PaddlePaddle/ERNIE
Source Location	https://github.com/PaddlePaddle/ERNIE
Tag(s)	NLP, Python

2.9.6 fastText

fastText is a library for efficient learning of word representations and sentence classification. Models can later be reduced in size to even fit on mobile devices.

Created by Facebook Opensource, now available for us all. Also used for the new search on StackOverflow, see https://stackoverflow.blog/2019/08/14/crokage-a-new-way-to-search-stack-overflow/

SBB License	MIT License
Core Technology	CPP, Python
Project URL	https://fasttext.cc/
Source Location	https://github.com/facebookresearch/fastText
Tag(s)	NLP

2.9.7 Flair

A very simple framework for **state-of-the-art NLP**. Developed by Zalando Research.

Flair is:

- **A powerful NLP library.** Flair allows you to apply our state-of-the-art natural language processing (NLP) models to your text, such as named entity recognition (NER), part-of-speech tagging (PoS), sense disambiguation and classification.
- **Multilingual.** Thanks to the Flair community, we support a rapidly growing number of languages. We also now include '*one model, many languages*' taggers, i.e. single models that predict PoS or NER tags for input text in various languages.
- **A text embedding library.** Flair has simple interfaces that allow you to use and combine different word and document embeddings, including our proposed **'Flair embeddings <https://drive.google.com/file/d/17yVpFA7MmXaQFTe-HDpZuqw9fJlmzg56/view?usp=sharing>'__**, BERT embeddings and ELMo embeddings.
- **A Pytorch NLP framework.** Our framework builds directly on Pytorch, making it easy to train your own models and experiment with new approaches using Flair embeddings and classes.

SBB License	MIT License
Core Technology	Python
Project URL	https://github.com/zalandoresearch/flair
Source Location	https://github.com/zalandoresearch/flair
Tag(s)	ML, NLP, Python

2.9.8 Gensim

Gensim is a Python library for *topic modelling*, *document indexing* and *similarity retrieval* with large corpora. Target audience is the *natural language processing* (NLP) and *information retrieval* (IR) community.

SBB License	MIT License
Core Technology	Python
Project URL	https://github.com/RaRe-Technologies/gensim
Source Location	https://github.com/RaRe-Technologies/gensim
Tag(s)	ML, NLP, Python

2.9.9 Neuralcoref

State-of-the-art coreference resolution based on neural nets and spaCy.

NeuralCoref is a pipeline extension for spaCy 2.0 that annotates and resolves coreference clusters using a neural network. NeuralCoref is production-ready, integrated in spaCy's NLP pipeline and easily extensible to new training datasets.

SBB License	MIT License
Core Technology	Python
Project URL	https://huggingface.co/coref/
Source Location	https://github.com/huggingface/neuralcoref
Tag(s)	ML, NLP, Python

2.9.10 NLP Architect

NLP Architect is an open-source Python library for exploring the state-of-the-art deep learning topologies and techniques for natural language processing and natural language understanding. It is intended to be a platform for future research and collaboration.

How can NLP Architect be used:

- Train models using provided algorithms, reference datasets and configurations
- Train models using your own data
- Create new/extend models based on existing models or topologies
- Explore how deep learning models tackle various NLP tasks
- Experiment and optimize state-of-the-art deep learning algorithms
- integrate modules and utilities from the library to solutions

SBB License	Apache License 2.0
Core Technology	Python
Project URL	http://nlp_architect.nervanasys.com/
Source Location	https://github.com/NervanaSystems/nlp-architect
Tag(s)	ML, NLP, Python

2.9.11 NLTK (Natural Language Toolkit)

NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries.

Check also the (free) online Book (OReily published)

SBB License	Apache License 2.0
Core Technology	Python
Project URL	http://www.nltk.org
Source Location	https://github.com/nltk/nltk
Tag(s)	NLP

2.9.12 Pattern

Pattern is a web mining module for Python. It has tools for:

- Data Mining: web services (Google, Twitter, Wikipedia), web crawler, HTML DOM parser
- Natural Language Processing: part-of-speech taggers, n-gram search, sentiment analysis, WordNet
- Machine Learning: vector space model, clustering, classification (KNN, SVM, Perceptron)
- Network Analysis: graph centrality and visualization.

SBB License	BSD License 2.0 (3-clause, New or Revised) License
Core Technology	Python
Project URL	https://www.clips.uantwerpen.be/pages/pattern
Source Location	https://github.com/clips/pattern
Tag(s)	ML, NLP, Web scraping

2.9.13 PDFx

Extract references (pdf, url, doi, arxiv) and metadata from a PDF. Optionally download all referenced PDFs and check for broken links.

Features

- Extract references and metadata from a given PDF
- Detects pdf, url, arxiv and doi references
- **Fast, parallel download of all referenced PDFs**
- **Find broken hyperlinks (using the ‘‘-c‘‘ flag)** (more)
- Output as text or JSON (using the `-j` flag)
- Extract the PDF text (using the `--text` flag)
- Use as command-line tool or Python package
- Compatible with Python 2 and 3
- Works with local and online pdfs

SBB License	Apache License 2.0
Core Technology	Python
Project URL	https://www.metachris.com/pdfx/
Source Location	https://github.com/metachris/pdfx
Tag(s)	NLP, Text Extraction

2.9.14 Rant

Rant is an all-purpose procedural text engine that is most simply described as the opposite of Regex. It has been refined to include a dizzying array of features for handling everything from the most basic of string generation tasks to advanced dialogue generation, code templating, automatic formatting, and more.

The goal of the project is to enable developers of all kinds to automate repetitive writing tasks with a high degree of creative freedom.

Features:

- Recursive, weighted branching with several selection modes
- Queryable dictionaries
- Automatic capitalization, rhyming, English indefinite articles, and multi-lingual number verbalization
- Print to multiple separate outputs
- Probability modifiers for pattern elements
- Loops, conditional statements, and subroutines
- Fully-functional object model
- Import/Export resources easily with the .rantpkg format
- Compatible with Unity 2017

SBB License	MIT License
Core Technology	.NET
Project URL	https://berkin.me/rant/
Source Location	https://github.com/TheBerkin/rant
Tag(s)	.NET, ML, NLP, text generation

2.9.15 SpaCy

Industrial-strength Natural Language Processing (NLP) with Python and Cython

Features:

- Non-destructive **tokenization**
- **Named entity** recognition
- Support for **26+ languages**
- **13 statistical models** for 8 languages
- Pre-trained **word vectors**
- Easy **deep learning** integration
- Part-of-speech tagging
- Labelled dependency parsing
- Syntax-driven sentence segmentation
- Built in **visualizers** for syntax and NER
- Convenient string-to-hash mapping
- Export to numpy data arrays
- Efficient binary serialization
- Easy **model packaging** and deployment
- State-of-the-art speed
- Robust, rigorously evaluated accuracy

SBB License	MIT License
Core Technology	Python
Project URL	https://spacy.io/
Source Location	https://github.com/explosion/spaCy
Tag(s)	NLP

2.9.16 Stanford CoreNLP

Stanford CoreNLP provides a set of human language technology tools. It can give the base forms of words, their parts of speech, whether they are names of companies, people, etc., normalize dates, times, and numeric quantities, mark up the structure of sentences in terms of phrases and syntactic dependencies, indicate which noun phrases refer to the same entities, indicate sentiment, extract particular or open-class relations between entity mentions, get the quotes people said, etc.

Choose Stanford CoreNLP if you need:

- An integrated NLP toolkit with a broad range of grammatical analysis tools
- A fast, robust annotator for arbitrary texts, widely used in production
- A modern, regularly updated package, with the overall highest quality text analytics
- Support for a number of major (human) languages
- Available APIs for most major modern programming languages
- Ability to run as a simple web service

SBB License	GNU General Public License (GPL) 3.0
Core Technology	Java
Project URL	https://stanfordnlp.github.io/CoreNLP/
Source Location	https://github.com/stanfordnlp/CoreNLP
Tag(s)	NLP

2.9.17 Sumeval

Well tested & Multi-language evaluation framework for text summarization. Multi-language.

SBB License	Apache License 2.0
Core Technology	Python
Project URL	https://github.com/chakki-works/sumeval
Source Location	https://github.com/chakki-works/sumeval
Tag(s)	NLP, Python

2.9.18 TextBlob: Simplified Text Processing

TextBlob is a Python (2 and 3) library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more.

2.9.19 Features

- Noun phrase extraction

- Part-of-speech tagging
- Sentiment analysis
- Classification (Naive Bayes, Decision Tree)
- Language translation and detection powered by Google Translate
- Tokenization (splitting text into words and sentences)
- Word and phrase frequencies
- Parsing
- n-grams
- Word inflection (pluralization and singularization) and lemmatization
- Spelling correction
- Add new models or languages through extensions
- WordNet integration

SBB License	MIT License
Core Technology	Python
Project URL	https://textblob.readthedocs.io/en/dev/
Source Location	https://github.com/sloria/textblob
Tag(s)	ML, NLP, Python

2.9.20 Thinc

Thinc is the machine learning library powering spaCy. It features a battle-tested linear model designed for large sparse learning problems, and a flexible neural network model under development for spaCy v2.0.

Thinc is a practical toolkit for implementing models that follow the "Embed, encode, attend, predict" architecture. It's designed to be easy to install, efficient for CPU usage and optimised for NLP and deep learning with text – in particular, hierarchically structured input and variable-length sequences.

SBB License	GNU General Public License (GPL) 2.0
Core Technology	Python
Project URL	https://explosion.ai/
Source Location	https://github.com/explosion/thinc
Tag(s)	ML, NLP, Python

2.9.21 Torchtext

Data loaders and abstractions for text and NLP. Build on PyTorch.

SBB License	BSD License 2.0 (3-clause, New or Revised) License
Core Technology	
Project URL	https://github.com/pytorch/text
Source Location	https://github.com/pytorch/text
Tag(s)	NLP

End of SBB list

2.10 ML Learning resources

Learning machine learning does not have to be very expensive or time consuming. Great learning material for machine learning is licensed under a Creative Commons license. So in most cases also free available for anyone who is eager to learn this technology.

In this section an opinionated list of great machine learning resources is published for learning this technology. Of course only resources that are open, so only resources published using a Creative Commons license (cc-by mostly) or other real open license are included. So all references are open access resources.

Most learning resource include hands-on tutorials. So be ready to use a notebook, but most tutorials offer notebooks ready to use directly.

- A Course in Machine Learning, http://ciml.info/

- Advanced NLP with spaCY, https://course.spacy.io/

- AutoML: Methods, Systems, Challenges, https://www.ml4aad.org/wp-content/uploads/2019/05/AutoML_Book.pdf

- Building Safe A.I., A Tutorial for Encrypted Deep Learnig, https://iamtrask.github.io/2017/03/17/safe-ai/

- Collection of Interactive Machine Learning Examples, http://tools.google.com/seedbank/

- Cryptography and Machine Learning, Mixing both for privacy-preserving machine learning, https://mortendahl.github.io/

- Dive into Deep Learning, An interactive deep learning book with code, math, and discussions, http://numpy.d2l.ai/

- Foundations of Machine Learning, Understand the Concepts, Techniques and Mathematical Frameworks Used by Experts in Machine Learning, https://bloomberg.github.io/foml/#home

- Interpretable Machine Learning,A Guide for Making Black Box Models Explainable,Christoph Molnar, https://christophm.github.io/interpretable-ml-book/

- Machine Learning Crash Course with TensorFlow APIs, https://developers.google.com/machine-learning/crash-course/ This is a great course published by Google's. It is advertised as a 'A self-study guide for aspiring machine learning practitioners'

- Machine Learning Guides, Simple step-by-step walkthroughs to solve common machine learning problems using best practices , https://developers.google.com/machine-learning/guides/

- Mathematics for Machine Learning, https://mml-book.github.io/ Examples and tutorials for this book are placed on: https://github.com/mml-book/mml-book.github.io

- NLP concepts with spaCy ,Allison Parrish (http://www.decontextualize.com/), https://gist.github.com/nocomplexity/b7c4c0aa5a0b53f4f5ff1c4784084be6

- Practical Deep Learning for Coders v3, https://course.fast.ai/index.html

- Python Machine Learning course, https://machine-learning-course.readthedocs.io/en/latest/index.html

- The Elements of AI, https://www.elementsofai.com/

2.11 NLP Learning resources

There is a large overlap between machine learning and current NLP technology. But is makes sense to outline specific NLP resources seperate. This to make searching for good open NLP resources easier.

So this section an opinionated list of great NLP learning resources. Of course also only resources that are open, so only resources published using a Creative Commons license (cc-by mostly) or other real open license are included. So all references are open access resources.

- Natural Language Processing with Python, http://www.nltk.org/book/

2.12 Help

We encourage all information professionals (developers, architects, consultants) to help to improve this Free and Open Machine Learning Guide.

Join the team to:

- Review and correct the content. (Yes there are still too much typos)
- Add new content blocks and/or images
- Create better graphics and text.
- Discuss the content so it gets better. We do discussion on-line and off-line (meetups)

2.13 License

You are free to: - Share — copy and redistribute the material in any medium or format - Adapt — remix, transform, and build upon the material for any purpose, even commercially.

The licensor cannot revoke these freedoms as long as you follow the license terms.

Under the following terms:

- Attribution — You must give appropriate credit, provide a link to the license, and indicate if changes were made. You may do so in any reasonable manner, but not in any way that suggests the licensor endorses you or your use.
- ShareAlike — If you remix, transform, or build upon the material, you must distribute your contributions under the same license as the original.

No additional restrictions — You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

Notices:

You do not have to comply with the license for elements of the material in the public domain or where your use is permitted by an applicable exception or limitation.

No warranties are given. The license may not give you all of the permissions necessary for your intended use. For example, other rights such as publicity, privacy, or moral rights may limit how you use the material.

CHAPTER 3

Contributors

The following people have contributed to the Free and Open Machine Learning project:

[name] [OPTIONAL email] [Optional Organization name]

If you like your name stated here: This book is open source. Issues and pull requests are welcome. All contributors will be added to this list.

So Get involved in the discussion to make it better!

If you wish to make comments regarding this document, please raise them as GitHub issues. Or send comments by email if you are unable to raise issues on GitHub. All comments are welcome!