# **Machine Learning**

You might be interested in 🤖 [*Interactive Machine Learning Experiments*](https://github.com/trekhleb/machine-learning-experiments)

For Octave/MatLab version of this repository please check [*machine-learning-octave*](https://github.com/trekhleb/machine-learning-octave) project.

This repository contains examples of popular machine learning algorithms implemented in **Python** with mathematics behind them being explained. Each algorithm has interactive **Jupyter Notebook** demo that allows you to play with training data, algorithms configurations and immediately see the results, charts and predictions **right in your browser**. In most cases the explanations are based on [this great machine learning course](https://www.coursera.org/learn/machine-learning) by Andrew Ng.

The purpose of this repository is not to implement machine learning algorithms by using 3rd party library one-liners but rather to practice implementing these algorithms from scratch and get better understanding of the mathematics behind each algorithm. That's why all algorithms implementations are called "homemade" and not intended to be used for production.

## Supervised Learning

In supervised learning we have a set of training data as an input and a set of labels or "correct answers" for each training set as an output. Then we're training our model (machine learning algorithm parameters) to map the input to the output correctly (to do correct prediction). The ultimate purpose is to find such model parameters that will successfully continue correct input→output mapping (predictions) even for new input examples.

### Regression

In regression problems we do real value predictions. Basically we try to draw a line/plane/n-dimensional plane along the training examples.

Usage examples: stock price forecast, sales analysis, dependency of any number, etc.

#### 🤖 Linear Regression

* 📗 [Math | Linear Regression](https://github.com/trekhleb/homemade-machine-learning/blob/master/homemade/linear_regression) - theory and links for further readings
* ⚙️ [Code | Linear Regression](https://github.com/trekhleb/homemade-machine-learning/blob/master/homemade/linear_regression/linear_regression.py) - implementation example
* ▶️ [Demo | Univariate Linear Regression](https://nbviewer.jupyter.org/github/trekhleb/homemade-machine-learning/blob/master/notebooks/linear_regression/univariate_linear_regression_demo.ipynb) - predict country happiness score by economy GDP
* ▶️ [Demo | Multivariate Linear Regression](https://nbviewer.jupyter.org/github/trekhleb/homemade-machine-learning/blob/master/notebooks/linear_regression/multivariate_linear_regression_demo.ipynb) - predict country happiness score by economy GDP and freedom index
* ▶️ [Demo | Non-linear Regression](https://nbviewer.jupyter.org/github/trekhleb/homemade-machine-learning/blob/master/notebooks/linear_regression/non_linear_regression_demo.ipynb) - use linear regression with polynomial and sinusoid features to predict non-linear dependencies

### Classification

In classification problems we split input examples by certain characteristic.

Usage examples: spam-filters, language detection, finding similar documents, handwritten letters recognition, etc.

#### 🤖 Logistic Regression

* 📗 [Math | Logistic Regression](https://github.com/trekhleb/homemade-machine-learning/blob/master/homemade/logistic_regression) - theory and links for further readings
* ⚙️ [Code | Logistic Regression](https://github.com/trekhleb/homemade-machine-learning/blob/master/homemade/logistic_regression/logistic_regression.py) - implementation example
* ▶️ [Demo | Logistic Regression (Linear Boundary)](https://nbviewer.jupyter.org/github/trekhleb/homemade-machine-learning/blob/master/notebooks/logistic_regression/logistic_regression_with_linear_boundary_demo.ipynb) - predict Iris flower class based on petal\_length and petal\_width
* ▶️ [Demo | Logistic Regression (Non-Linear Boundary)](https://nbviewer.jupyter.org/github/trekhleb/homemade-machine-learning/blob/master/notebooks/logistic_regression/logistic_regression_with_non_linear_boundary_demo.ipynb) - predict microchip validity based on param\_1 and param\_2
* ▶️ [Demo | Multivariate Logistic Regression | MNIST](https://nbviewer.jupyter.org/github/trekhleb/homemade-machine-learning/blob/master/notebooks/logistic_regression/multivariate_logistic_regression_demo.ipynb) - recognize handwritten digits from 28x28 pixel images
* ▶️ [Demo | Multivariate Logistic Regression | Fashion MNIST](https://nbviewer.jupyter.org/github/trekhleb/homemade-machine-learning/blob/master/notebooks/logistic_regression/multivariate_logistic_regression_fashion_demo.ipynb) - recognize clothes types from 28x28 pixel images

## Unsupervised Learning

Unsupervised learning is a branch of machine learning that learns from test data that has not been labeled, classified or categorized. Instead of responding to feedback, unsupervised learning identifies commonalities in the data and reacts based on the presence or absence of such commonalities in each new piece of data.

### Clustering

In clustering problems we split the training examples by unknown characteristics. The algorithm itself decides what characteristic to use for splitting.

Usage examples: market segmentation, social networks analysis, organize computing clusters, astronomical data analysis, image compression, etc.

#### 🤖 K-means Algorithm

* 📗 [Math | K-means Algorithm](https://github.com/trekhleb/homemade-machine-learning/blob/master/homemade/k_means) - theory and links for further readings
* ⚙️ [Code | K-means Algorithm](https://github.com/trekhleb/homemade-machine-learning/blob/master/homemade/k_means/k_means.py) - implementation example
* ▶️ [Demo | K-means Algorithm](https://nbviewer.jupyter.org/github/trekhleb/homemade-machine-learning/blob/master/notebooks/k_means/k_means_demo.ipynb) - split Iris flowers into clusters based on petal\_length and petal\_width

### Anomaly Detection

Anomaly detection (also outlier detection) is the identification of rare items, events or observations which raise suspicions by differing significantly from the majority of the data.

Usage examples: intrusion detection, fraud detection, system health monitoring, removing anomalous data from the dataset etc.

#### 🤖 Anomaly Detection using Gaussian Distribution

* 📗 [Math | Anomaly Detection using Gaussian Distribution](https://github.com/trekhleb/homemade-machine-learning/blob/master/homemade/anomaly_detection) - theory and links for further readings
* ⚙️ [Code | Anomaly Detection using Gaussian Distribution](https://github.com/trekhleb/homemade-machine-learning/blob/master/homemade/anomaly_detection/gaussian_anomaly_detection.py) - implementation example
* ▶️ [Demo | Anomaly Detection](https://nbviewer.jupyter.org/github/trekhleb/homemade-machine-learning/blob/master/notebooks/anomaly_detection/anomaly_detection_gaussian_demo.ipynb) - find anomalies in server operational parameters like latency and threshold

## Neural Network (NN)

The neural network itself isn't an algorithm, but rather a framework for many different machine learning algorithms to work together and process complex data inputs.

Usage examples: as a substitute of all other algorithms in general, image recognition, voice recognition, image processing (applying specific style), language translation, etc.

#### 🤖 Multilayer Perceptron (MLP)

* 📗 [Math | Multilayer Perceptron](https://github.com/trekhleb/homemade-machine-learning/blob/master/homemade/neural_network) - theory and links for further readings
* ⚙️ [Code | Multilayer Perceptron](https://github.com/trekhleb/homemade-machine-learning/blob/master/homemade/neural_network/multilayer_perceptron.py) - implementation example
* ▶️ [Demo | Multilayer Perceptron | MNIST](https://nbviewer.jupyter.org/github/trekhleb/homemade-machine-learning/blob/master/notebooks/neural_network/multilayer_perceptron_demo.ipynb) - recognize handwritten digits from 28x28 pixel images
* ▶️ [Demo | Multilayer Perceptron | Fashion MNIST](https://nbviewer.jupyter.org/github/trekhleb/homemade-machine-learning/blob/master/notebooks/neural_network/multilayer_perceptron_fashion_demo.ipynb) - recognize the type of clothes from 28x28 pixel images

## Machine Learning Map

The source of the following machine learning topics map is [this wonderful blog post](https://vas3k.ru/blog/machine_learning/)

## Prerequisites

#### Installing Python

Make sure that you have [Python installed](https://realpython.com/installing-python/) on your machine.

You might want to use [venv](https://docs.python.org/3/library/venv.html) standard Python library to create virtual environments and have Python, pip and all dependent packages to be installed and served from the local project directory to avoid messing with system wide packages and their versions.

#### Installing Dependencies

Install all dependencies that are required for the project by running:

pip install -r requirements.txt

#### Launching Jupyter Locally

All demos in the project may be run directly in your browser without installing Jupyter locally. But if you want to launch [Jupyter Notebook](http://jupyter.org/) locally you may do it by running the following command from the root folder of the project:

jupyter notebook

After this Jupyter Notebook will be accessible by http://localhost:8888.

#### Launching Jupyter Remotely

Each algorithm section contains demo links to [Jupyter NBViewer](http://nbviewer.jupyter.org/). This is fast online previewer for Jupyter notebooks where you may see demo code, charts and data right in your browser without installing anything locally. In case if you want to change the code and experiment with demo notebook you need to launch the notebook in [Binder](https://mybinder.org/). You may do it by simply clicking the "Execute on Binder" link in top right corner of the NBViewer.

## Datasets

The list of datasets that is being used for Jupyter Notebook demos may be found in [data folder](https://github.com/trekhleb/homemade-machine-learning/blob/master/data)

## **Deep Learning Track**

## Week 1 - Feedforward Neural Networks and Backpropagation

* Read Part I of the Deep Learning Book found [here](http://www.deeplearningbook.org/)
* Use this cheat sheet to help understand any math notation, found [here](https://www.flickr.com/photos/95869671@N08/40544016221)
* Watch [Build a Neural Net in 4 Minutes](https://www.youtube.com/watch?v=h3l4qz76JhQ)
* Read [Neural Net in 11 lines](https://iamtrask.github.io/2015/07/12/basic-python-network/)
* Type out the neural network code yourself in a text editor, compile, and run it locally (using no ML libraries)
* Watch [Backpropagation in 5 minutes](https://www.youtube.com/watch?v=q555kfIFUCM)

## Week 2 - Convolutional Networks

* Watch the Convolutional Networks Specialization on Coursera, found [here](https://www.coursera.org/learn/convolutional-neural-networks).
* Read all 3 lecture notes under Module 2 for Karpathy CNN course found [here](http://cs231n.github.io/)
* Watch my video on CNNs [here](https://www.youtube.com/watch?v=FTr3n7uBIuE&t=1782s) and [here](https://www.youtube.com/watch?v=cAICT4Al5Ow&t=4s)
* Write out a simple CNN yourself (using no ML libraries)

## Week 3 - Recurrent Networks

* Watch the Sequence Models Specialization on Coursera, found [here](https://www.coursera.org/learn/nlp-sequence-models)
* Watch my videos on recurrent networks, [here](https://www.youtube.com/watch?v=BwmddtPFWtA&t=4s), [here](https://www.youtube.com/watch?v=cdLUzrjnlr4), and [here](https://www.youtube.com/watch?v=9zhrxE5PQgY&t=25s)
* Read Trask's blogpost on LSTM RNNs found [here](https://iamtrask.github.io/2015/11/15/anyone-can-code-lstm/)
* Write out a simple RNN yourself (using no ML libraries)

## Week 4 - Tooling

* Watch CS20 (Tensorflow for DL research). Slides are [here](http://web.stanford.edu/class/cs20si/syllabus.html). Playlist is [here](https://www.youtube.com/watch?v=g-EvyKpZjmQ&list=PLDuNt91tg0urwwTQNKyUbncSDvMEl74ww)
* Watch my intro to tensorflow playlist [here](https://www.youtube.com/watch?v=2FmcHiLCwTU&list=PL2-dafEMk2A7EEME489DsI468AB0wQsMV)
* Read Keras Example code to quickly understand its structure [here](https://keras.io/getting-started/sequential-model-guide/)
* Learn which GPU provider is best for you [here](https://medium.com/@rupak.thakur/aws-vs-paperspace-vs-floydhub-choosing-your-cloud-gpu-partner-350150606b39)
* Write out a simple image classifier using Tensorflow

## Week 5 - Generative Adversarial Network

* Watch the first 7 videos you see [here](https://www.youtube.com/results?search_query=generative+adversarial+network)
* Build a GAN using no ML libraries
* Build a GAN using tensorflow
* Read this to understand the math of GANs, but don't worry if you dont understand it all. This is the bleeding edge [here](https://lilianweng.github.io/lil-log/2017/08/20/from-GAN-to-WGAN.html)

## Week 6 - Deep Reinforcement Learning

* Watch CS 294 [here](http://rail.eecs.berkeley.edu/deeprlcourse/)
* Build a Deep Q Network using Tensorflow

### **Computer Science Track**

### 5 Month Curriculum

## Week 1-2 (Learn Python)

* <https://automatetheboringstuff.com/>
* <https://www.codecademy.com/learn/learn-python>

## Week 3-4 (Data Structures)

* <https://www.edx.org/course/data-structures-fundamentals-uc-san-diegox-algs201x>

## Week 5-6 (Algorithms)

* <https://courses.csail.mit.edu/6.006/fall11/notes.shtml>

## Week 7 (Databases)

* <https://www.coursera.org/learn/python-databases>

## Week 8 (Networking)

* <https://www.coursera.org/learn/computer-networking>

## Week 9-10 (Web Development)

* <https://www.youtube.com/watch?v=1u2qu-EmIRc&list=PLhQjrBD2T382hIW-IsOVuXP1uMzEvmcE5>
* <https://github.com/melanierichards/just-build-websites>

## Week 11-12 (Mobile Development)

* <https://developer.apple.com/library/content/referencelibrary/GettingStarted/DevelopiOSAppsSwift/>
* <https://developer.android.com/training/basics/firstapp/index.html>

## Week 13-14 (Data Science)

* <https://www.edx.org/course/python-for-data-science>

## Week 15-16 (Computer Vision)

* <https://www.udacity.com/course/introduction-to-computer-vision--ud810>

## Week 17-18 (Natural Language Processing)

* <https://www.coursera.org/learn/python-text-mining>

## Week 19 (Software Engineering Practices)

* <https://www.coursera.org/learn/software-processes>

## Week 20 (Blockchain)

* <https://www.youtube.com/watch?v=cjbHqvr4ffo&list=PL2-dafEMk2A7jW7CYUJsBu58JH27bqaNL>

# **Top-down learning path:**

# **Machine Learning for Software Engineers**

Inspired by [Coding Interview University](https://github.com/jwasham/coding-interview-university).

Translations: [Brazilian Portuguese](https://github.com/ZuzooVn/machine-learning-for-software-engineers/blob/master/README-pt-BR.md) | [中文版本](https://github.com/ZuzooVn/machine-learning-for-software-engineers/blob/master/README-zh-CN.md) | [Français](https://github.com/ZuzooVn/machine-learning-for-software-engineers/blob/master/README-fr-FR.md) | [臺灣華語版本](https://github.com/ZuzooVn/machine-learning-for-software-engineers/blob/master/README-zh-TW.md)

## What is it?

This is my multi-month study plan for going from mobile developer (self-taught, no CS degree) to machine learning engineer.

My main goal was to find an approach to studying Machine Learning that is mainly hands-on and abstracts most of the Math for the beginner. This approach is unconventional because it’s the top-down and results-first approach designed for software engineers.

Please, feel free to make any contributions you feel will make it better.

## Table of Contents

* [What is it?](https://github.com/ZuzooVn/machine-learning-for-software-engineers#what-is-it)
* [Why use it?](https://github.com/ZuzooVn/machine-learning-for-software-engineers#why-use-it)
* [How to use it](https://github.com/ZuzooVn/machine-learning-for-software-engineers#how-to-use-it)
* [Follow me](https://github.com/ZuzooVn/machine-learning-for-software-engineers#follow-me)
* [Don't feel you aren't smart enough](https://github.com/ZuzooVn/machine-learning-for-software-engineers#dont-feel-you-arent-smart-enough)
* [About Video Resources](https://github.com/ZuzooVn/machine-learning-for-software-engineers#about-video-resources)
* [Prerequisite Knowledge](https://github.com/ZuzooVn/machine-learning-for-software-engineers#prerequisite-knowledge)
* [The Daily Plan](https://github.com/ZuzooVn/machine-learning-for-software-engineers#the-daily-plan)
* [Motivation](https://github.com/ZuzooVn/machine-learning-for-software-engineers#motivation)
* [Machine learning overview](https://github.com/ZuzooVn/machine-learning-for-software-engineers#machine-learning-overview)
* [Machine learning mastery](https://github.com/ZuzooVn/machine-learning-for-software-engineers#machine-learning-mastery)
* [Machine learning is fun](https://github.com/ZuzooVn/machine-learning-for-software-engineers#machine-learning-is-fun)
* [Inky Machine Learning](https://github.com/ZuzooVn/machine-learning-for-software-engineers#inky-machine-learning)
* [Machine Learning: An In-Depth Guide](https://github.com/ZuzooVn/machine-learning-for-software-engineers#machine-learning-an-in-depth-guide)
* [Stories and experiences](https://github.com/ZuzooVn/machine-learning-for-software-engineers#stories-and-experiences)
* [Machine Learning Algorithms](https://github.com/ZuzooVn/machine-learning-for-software-engineers#machine-learning-algorithms)
* [Beginner Books](https://github.com/ZuzooVn/machine-learning-for-software-engineers#beginner-books)
* [Practical Books](https://github.com/ZuzooVn/machine-learning-for-software-engineers#practical-books)
* [Kaggle knowledge competitions](https://github.com/ZuzooVn/machine-learning-for-software-engineers#kaggle-knowledge-competitions)
* [Video Series](https://github.com/ZuzooVn/machine-learning-for-software-engineers#video-series)
* [MOOC](https://github.com/ZuzooVn/machine-learning-for-software-engineers#mooc)
* [Resources](https://github.com/ZuzooVn/machine-learning-for-software-engineers#resources)
* [Becoming an Open Source Contributor](https://github.com/ZuzooVn/machine-learning-for-software-engineers#becoming-an-open-source-contributor)
* [Games](https://github.com/ZuzooVn/machine-learning-for-software-engineers#games)
* [Podcasts](https://github.com/ZuzooVn/machine-learning-for-software-engineers#podcasts)
* [Communities](https://github.com/ZuzooVn/machine-learning-for-software-engineers#communities)
* [Conferences](https://github.com/ZuzooVn/machine-learning-for-software-engineers#conferences)
* [Interview Questions](https://github.com/ZuzooVn/machine-learning-for-software-engineers#interview-questions)
* [My admired companies](https://github.com/ZuzooVn/machine-learning-for-software-engineers#my-admired-companies)

## Why use it?

I'm following this plan to prepare for my near-future job: Machine learning engineer. I've been building native mobile applications (Android/iOS/Blackberry) since 2011. I have a Software Engineering degree, not a Computer Science degree. I have an itty-bitty amount of basic knowledge about: Calculus, Linear Algebra, Discrete Mathematics, Probability & Statistics from university. Think about my interest in machine learning:

* [Can I learn and get a job in Machine Learning without studying CS Master and PhD?](https://www.quora.com/Can-I-learn-and-get-a-job-in-Machine-Learning-without-studying-CS-Master-and-PhD)
  + "You can, but it is far more difficult than when I got into the field." [Drac Smith](https://www.quora.com/Can-I-learn-and-get-a-job-in-Machine-Learning-without-studying-CS-Master-and-PhD/answer/Drac-Smith?srid=oT0p)
* [How do I get a job in Machine Learning as a software programmer who self-studies Machine Learning, but never has a chance to use it at work?](https://www.quora.com/How-do-I-get-a-job-in-Machine-Learning-as-a-software-programmer-who-self-studies-Machine-Learning-but-never-has-a-chance-to-use-it-at-work)
  + "I'm hiring machine learning experts for my team and your MOOC will not get you the job (there is better news below). In fact, many people with a master's in machine learning will not get the job because they (and most who have taken MOOCs) do not have a deep understanding that will help me solve my problems." [Ross C. Taylor](https://www.quora.com/How-do-I-get-a-job-in-Machine-Learning-as-a-software-programmer-who-self-studies-Machine-Learning-but-never-has-a-chance-to-use-it-at-work/answer/Ross-C-Taylor?srid=oT0p)
* [What skills are needed for machine learning jobs?](http://programmers.stackexchange.com/questions/79476/what-skills-are-needed-for-machine-learning-jobs)
  + "First, you need to have a decent CS/Math background. ML is an advanced topic so most textbooks assume that you have that background. Second, machine learning is a very general topic with many sub-specialties requiring unique skills. You may want to browse the curriculum of an MS program in Machine Learning to see the course, curriculum and textbook." [Uri](http://softwareengineering.stackexchange.com/a/79717)
  + "Probability, distributed computing, and Statistics." [Hydrangea](http://softwareengineering.stackexchange.com/a/79575)

I find myself in times of trouble.

AFAIK, [There are two sides to machine learning](http://machinelearningmastery.com/programmers-can-get-into-machine-learning/):

* Practical Machine Learning: This is about querying databases, cleaning data, writing scripts to transform data and gluing algorithm and libraries together and writing custom code to squeeze reliable answers from data to satisfy difficult and ill-defined questions. It’s the mess of reality.
* Theoretical Machine Learning: This is about math and abstraction and idealized scenarios and limits and beauty and informing what is possible. It is a whole lot neater and cleaner and removed from the mess of reality.

I think the best way for practice-focused methodology is something like ['practice — learning — practice'](http://machinelearningmastery.com/machine-learning-for-programmers/#comment-358985), that means where students first come with some existing projects with problems and solutions (practice) to get familiar with traditional methods in the area and perhaps also with their methodology. After practicing with some elementary experiences, they can go into the books and study the underlying theory, which serves to guide their future advanced practice and will enhance their toolbox of solving practical problems. Studying theory also further improves their understanding on the elementary experiences, and will help them acquire advanced experiences more quickly.

It's a long plan. It's going to take me years. If you are familiar with a lot of this already it will take you a lot less time.

## How to use it

Everything below is an outline, and you should tackle the items in order from top to bottom.

I'm using Github's special markdown flavor, including tasks lists to check progress.

* Create a new branch so you can check items like this, just put an x in the brackets: [x]

[More about Github-flavored markdown](https://guides.github.com/features/mastering-markdown/#GitHub-flavored-markdown)

|  |
| --- |
|  |

## Don't feel you aren't smart enough

I get discouraged from books and courses that tell me as soon as I open them that multivariate calculus, inferential statistics and linear algebra are prerequisites. I still don’t know how to get started…

* [What if I’m Not Good at Mathematics](http://machinelearningmastery.com/what-if-im-not-good-at-mathematics/)
* [5 Techniques To Understand Machine Learning Algorithms Without the Background in Mathematics](http://machinelearningmastery.com/techniques-to-understand-machine-learning-algorithms-without-the-background-in-mathematics/)
* [How do I learn machine learning?](https://www.quora.com/Machine-Learning/How-do-I-learn-machine-learning-1)

## About Video Resources

Some videos are available only by enrolling in a Coursera or EdX class. It is free to do so, but sometimes the classes are no longer in session so you have to wait a couple of months, so you have no access. I'm going to be adding more videos from public sources and replacing the online course videos over time. I like using university lectures.

## Prerequisite Knowledge

This short section consists of prerequisites/interesting info I wanted to learn before getting started on the daily plan.

* [What is the difference between Data Analytics, Data Analysis, Data Mining, Data Science, Machine Learning, and Big Data?](https://www.quora.com/What-is-the-difference-between-Data-Analytics-Data-Analysis-Data-Mining-Data-Science-Machine-Learning-and-Big-Data-1)
* [Learning How to Learn](https://www.coursera.org/learn/learning-how-to-learn)
* [Don’t Break The Chain](http://lifehacker.com/281626/jerry-seinfelds-productivity-secret)
* [How to learn on your own](https://metacademy.org/roadmaps/rgrosse/learn_on_your_own)

## The Daily Plan

Each subject does not require a whole day to be able to understand it fully, and you can do multiple of these in a day.

Each day I take one subject from the list below, read it cover to cover, take notes, do the exercises and write an implementation in Python or R.

# Motivation

* [Dream](https://www.youtube.com/watch?v=g-jwWYX7Jlo)

## Machine learning overview

* [A Visual Introduction to Machine Learning](http://www.r2d3.us/visual-intro-to-machine-learning-part-1/)
* [A Gentle Guide to Machine Learning](https://blog.monkeylearn.com/a-gentle-guide-to-machine-learning/)
* [Introduction to Machine Learning for Developers](http://blog.algorithmia.com/introduction-machine-learning-developers/)
* [Machine Learning basics for a newbie](https://www.analyticsvidhya.com/blog/2015/06/machine-learning-basics/)
* [How do you explain Machine Learning and Data Mining to non Computer Science people?](https://www.quora.com/How-do-you-explain-Machine-Learning-and-Data-Mining-to-non-Computer-Science-people)
* [Machine Learning: Under the hood. Blog post explains the principles of machine learning in layman terms. Simple and clear](https://georgemdallas.wordpress.com/2013/06/11/big-data-data-mining-and-machine-learning-under-the-hood/)
* [What is machine learning, and how does it work?](https://www.youtube.com/watch?v=elojMnjn4kk&list=PL5-da3qGB5ICeMbQuqbbCOQWcS6OYBr5A&index=1)
* [Deep Learning - A Non-Technical Introduction](http://www.slideshare.net/AlfredPong1/deep-learning-a-nontechnical-introduction-69385936)

## Machine learning mastery

* [The Machine Learning Mastery Method](http://machinelearningmastery.com/machine-learning-mastery-method/)
* [Machine Learning for Programmers](http://machinelearningmastery.com/machine-learning-for-programmers/)
* [Applied Machine Learning with Machine Learning Mastery](http://machinelearningmastery.com/start-here/)
* [Python Machine Learning Mini-Course](http://machinelearningmastery.com/python-machine-learning-mini-course/)
* [Machine Learning Algorithms Mini-Course](http://machinelearningmastery.com/machine-learning-algorithms-mini-course/)

## Machine learning is fun

* [Machine Learning is Fun!](https://medium.com/@ageitgey/machine-learning-is-fun-80ea3ec3c471#.37ue6caww)
* [Part 2: Using Machine Learning to generate Super Mario Maker levels](https://medium.com/@ageitgey/machine-learning-is-fun-part-2-a26a10b68df3#.kh7qgvp1b)
* [Part 3: Deep Learning and Convolutional Neural Networks](https://medium.com/@ageitgey/machine-learning-is-fun-part-3-deep-learning-and-convolutional-neural-networks-f40359318721#.44rhxy637)
* [Part 4: Modern Face Recognition with Deep Learning](https://medium.com/@ageitgey/machine-learning-is-fun-part-4-modern-face-recognition-with-deep-learning-c3cffc121d78#.3rwmq0ddc)
* [Part 5: Language Translation with Deep Learning and the Magic of Sequences](https://medium.com/@ageitgey/machine-learning-is-fun-part-5-language-translation-with-deep-learning-and-the-magic-of-sequences-2ace0acca0aa#.wyfthap4c)
* [Part 6: How to do Speech Recognition with Deep Learning](https://medium.com/@ageitgey/machine-learning-is-fun-part-6-how-to-do-speech-recognition-with-deep-learning-28293c162f7a#.lhr1nnpcy)
* [Part 7: Abusing Generative Adversarial Networks to Make 8-bit Pixel Art](https://medium.com/@ageitgey/abusing-generative-adversarial-networks-to-make-8-bit-pixel-art-e45d9b96cee7)
* [Part 8: How to Intentionally Trick Neural Networks](https://medium.com/@ageitgey/machine-learning-is-fun-part-8-how-to-intentionally-trick-neural-networks-b55da32b7196)

## [Inky Machine Learning](https://triskell.github.io/2016/11/15/Inky-Machine-Learning.html)

* [Part 1: What is Machine Learning ?](https://triskell.github.io/2016/10/23/What-is-Machine-Learning.html)
* [Part 2: Supervised Learning and Unsupervised Learning](https://triskell.github.io/2016/11/13/Supervised-Learning-and-Unsupervised-Learning.html)

## Machine Learning: An In-Depth Guide

* [Overview, goals, learning types, and algorithms](http://www.innoarchitech.com/machine-learning-an-in-depth-non-technical-guide/)
* [Data selection, preparation, and modeling](http://www.innoarchitech.com/machine-learning-an-in-depth-non-technical-guide-part-2/)
* [Model evaluation, validation, complexity, and improvement](http://www.innoarchitech.com/machine-learning-an-in-depth-non-technical-guide-part-3/)
* [Model performance and error analysis](http://www.innoarchitech.com/machine-learning-an-in-depth-non-technical-guide-part-4/)
* [Unsupervised learning, related fields, and machine learning in practice](http://www.innoarchitech.com/machine-learning-an-in-depth-non-technical-guide-part-5/)

## Stories and experiences

* [Machine Learning in a Week](https://medium.com/learning-new-stuff/machine-learning-in-a-week-a0da25d59850#.tk6ft2kcg)
* [Machine Learning in a Year](https://medium.com/learning-new-stuff/machine-learning-in-a-year-cdb0b0ebd29c#.hhcb9fxk1)
* [How I wrote my first Machine Learning program in 3 days](http://blog.adnansiddiqi.me/how-i-wrote-my-first-machine-learning-program-in-3-days/)
* [Learning Path : Your mentor to become a machine learning expert](https://www.analyticsvidhya.com/learning-path-learn-machine-learning/)
* [You Too Can Become a Machine Learning Rock Star! No PhD](https://backchannel.com/you-too-can-become-a-machine-learning-rock-star-no-phd-necessary-107a1624d96b#.g9p16ldp7)
* How to become a Data Scientist in 6 months: A hacker’s approach to career planning
  + [Video](https://www.youtube.com/watch?v=rIofV14c0tc)
  + [Slide](http://www.slideshare.net/TetianaIvanova2/how-to-become-a-data-scientist-in-6-months)
* [5 Skills You Need to Become a Machine Learning Engineer](http://blog.udacity.com/2016/04/5-skills-you-need-to-become-a-machine-learning-engineer.html)
* [Are you a self-taught machine learning engineer? If yes, how did you do it & how long did it take you?](https://www.quora.com/Are-you-a-self-taught-machine-learning-engineer-If-yes-how-did-you-do-it-how-long-did-it-take-you)
* [How can one become a good machine learning engineer?](https://www.quora.com/How-can-one-become-a-good-machine-learning-engineer)
* [A Learning Sabbatical focused on Machine Learning](http://karlrosaen.com/ml/)

## Machine Learning Algorithms

* [10 Machine Learning Algorithms Explained to an ‘Army Soldier’](https://www.analyticsvidhya.com/blog/2015/12/10-machine-learning-algorithms-explained-army-soldier/)
* [Top 10 data mining algorithms in plain English](https://rayli.net/blog/data/top-10-data-mining-algorithms-in-plain-english/)
* [10 Machine Learning Terms Explained in Simple English](http://blog.aylien.com/10-machine-learning-terms-explained-in-simple/)
* [A Tour of Machine Learning Algorithms](http://machinelearningmastery.com/a-tour-of-machine-learning-algorithms/)
* [The 10 Algorithms Machine Learning Engineers Need to Know](https://gab41.lab41.org/the-10-algorithms-machine-learning-engineers-need-to-know-f4bb63f5b2fa#.ofc7t2965)
* [Comparing supervised learning algorithms](http://www.dataschool.io/comparing-supervised-learning-algorithms/)
* [Machine Learning Algorithms: A collection of minimal and clean implementations of machine learning algorithms](https://github.com/rushter/MLAlgorithms)

## Beginner Books

* [Data Smart: Using Data Science to Transform Information into Insight 1st Edition](https://www.amazon.com/Data-Smart-Science-Transform-Information/dp/111866146X)
* [Data Science for Business: What you need to know about data mining and data­ analytic-thinking](https://www.amazon.com/Data-Science-Business-Data-Analytic-Thinking/dp/1449361323/)
* [Predictive Analytics: The Power to Predict Who Will Click, Buy, Lie, or Die](https://www.amazon.com/Predictive-Analytics-Power-Predict-Click/dp/1118356853)

## Practical Books

* [Machine Learning for Hackers](https://www.amazon.com/Machine-Learning-Hackers-Drew-Conway/dp/1449303714)
  + [GitHub repository(R)](https://github.com/johnmyleswhite/ML_for_Hackers)
  + [GitHub repository(Python)](https://github.com/carljv/Will_it_Python)
* [Python Machine Learning](https://www.amazon.com/Python-Machine-Learning-Sebastian-Raschka-ebook/dp/B00YSILNL0)
  + [GitHub repository](https://github.com/rasbt/python-machine-learning-book)
* [Programming Collective Intelligence: Building Smart Web 2.0 Applications](https://www.amazon.com/Programming-Collective-Intelligence-Building-Applications-ebook/dp/B00F8QDZWG)
* [Machine Learning: An Algorithmic Perspective, Second Edition](https://www.amazon.com/Machine-Learning-Algorithmic-Perspective-Recognition/dp/1466583282)
  + [GitHub repository](https://github.com/alexsosn/MarslandMLAlgo)
  + [Resource repository](http://seat.massey.ac.nz/personal/s.r.marsland/MLbook.html)
* [Introduction to Machine Learning with Python: A Guide for Data Scientists](http://shop.oreilly.com/product/0636920030515.do)
  + [GitHub repository](https://github.com/amueller/introduction_to_ml_with_python)
* [Data Mining: Practical Machine Learning Tools and Techniques, Third Edition](https://www.amazon.com/Data-Mining-Practical-Techniques-Management/dp/0123748569)
  + Teaching material
    - [Slides for Chapters 1-5 (zip)](http://www.cs.waikato.ac.nz/ml/weka/Slides3rdEd_Ch1-5.zip)
    - [Slides for Chapters 6-8 (zip)](http://www.cs.waikato.ac.nz/ml/weka/Slides3rdEd_Ch6-8.zip)
* [Machine Learning in Action](https://www.amazon.com/Machine-Learning-Action-Peter-Harrington/dp/1617290181/)
  + [GitHub repository](https://github.com/pbharrin/machinelearninginaction)
* [Reactive Machine Learning Systems(MEAP)](https://www.manning.com/books/reactive-machine-learning-systems)
  + [GitHub repository](https://github.com/jeffreyksmithjr/reactive-machine-learning-systems)
* [An Introduction to Statistical Learning](http://www-bcf.usc.edu/~gareth/ISL/)
  + [GitHub repository(R)](http://www-bcf.usc.edu/~gareth/ISL/code.html)
  + [GitHub repository(Python)](https://github.com/JWarmenhoven/ISLR-python)
  + [Videos](http://www.dataschool.io/15-hours-of-expert-machine-learning-videos/)
* [Building Machine Learning Systems with Python](https://www.packtpub.com/big-data-and-business-intelligence/building-machine-learning-systems-python)
  + [GitHub repository](https://github.com/luispedro/BuildingMachineLearningSystemsWithPython)
* [Learning scikit-learn: Machine Learning in Python](https://www.packtpub.com/big-data-and-business-intelligence/learning-scikit-learn-machine-learning-python)
  + [GitHub repository](https://github.com/gmonce/scikit-learn-book)
* [Probabilistic Programming & Bayesian Methods for Hackers](https://camdavidsonpilon.github.io/Probabilistic-Programming-and-Bayesian-Methods-for-Hackers/)
* [Probabilistic Graphical Models: Principles and Techniques](https://www.amazon.com/Probabilistic-Graphical-Models-Principles-Computation/dp/0262013193)
* [Machine Learning: Hands-On for Developers and Technical Professionals](https://www.amazon.com/Machine-Learning-Hands-Developers-Professionals/dp/1118889061)
  + [Machine Learning Hands-On for Developers and Technical Professionals review](https://blogs.msdn.microsoft.com/querysimon/2015/01/01/book-review-machine-learning-hands-on-for-developers-and-technical-professionals/)
  + [GitHub repository](https://github.com/jasebell/mlbook)
* [Learning from Data](https://www.amazon.com/Learning-Data-Yaser-S-Abu-Mostafa/dp/1600490069)
  + [Online tutorials](https://work.caltech.edu/telecourse.html)
* [Reinforcement Learning: An Introduction (2nd Edition)](https://webdocs.cs.ualberta.ca/~sutton/book/the-book-2nd.html)
  + [GitHub repository](https://github.com/ShangtongZhang/reinforcement-learning-an-introduction)
* [Machine Learning with TensorFlow(MEAP)](https://www.manning.com/books/machine-learning-with-tensorflow)
  + [GitHub repository](https://github.com/BinRoot/TensorFlow-Book)
* [How Machine Learning Works (MEAP)](https://www.manning.com/books/how-machine-learning-works)
  + [GitHub repository](https://github.com/Mostafa-Samir/How-Machine-Learning-Works)
* [Succeeding with AI](https://www.manning.com/books/succeeding-with-ai)

## Kaggle knowledge competitions

* [Kaggle Competitions: How and where to begin?](https://www.analyticsvidhya.com/blog/2015/06/start-journey-kaggle/)
* [How a Beginner Used Small Projects To Get Started in Machine Learning and Compete on Kaggle](http://machinelearningmastery.com/how-a-beginner-used-small-projects-to-get-started-in-machine-learning-and-compete-on-kaggle)
* [Master Kaggle By Competing Consistently](http://machinelearningmastery.com/master-kaggle-by-competing-consistently/)

## Video Series

* [Machine Learning for Hackers](https://www.youtube.com/playlist?list=PL2-dafEMk2A4ut2pyv0fSIXqOzXtBGkLj)
* [Fresh Machine Learning](https://www.youtube.com/playlist?list=PL2-dafEMk2A6Kc7pV6gHH-apBFxwFjKeY)
* [Machine Learning Recipes with Josh Gordon](https://www.youtube.com/playlist?list=PLOU2XLYxmsIIuiBfYad6rFYQU_jL2ryal)
* [Everything You Need to know about Machine Learning in 30 Minutes or Less](https://vimeo.com/43547079)
* [A Friendly Introduction to Machine Learning](https://www.youtube.com/watch?v=IpGxLWOIZy4)
* [Nuts and Bolts of Applying Deep Learning - Andrew Ng](https://www.youtube.com/watch?v=F1ka6a13S9I)
* BigML Webinar
  + [Video](https://www.youtube.com/watch?list=PL1bKyu9GtNYHcjGa6ulrvRVcm1lAB8he3&v=W62ehrnOVqo)
  + [Resources](https://bigml.com/releases)
* [mathematicalmonk's Machine Learning tutorials](https://www.youtube.com/playlist?list=PLD0F06AA0D2E8FFBA)
* [Machine learning in Python with scikit-learn](https://www.youtube.com/playlist?list=PL5-da3qGB5ICeMbQuqbbCOQWcS6OYBr5A)
  + [GitHub repository](https://github.com/justmarkham/scikit-learn-videos)
  + [Blog](http://blog.kaggle.com/author/kevin-markham/)
* [My playlist – Top YouTube Videos on Machine Learning, Neural Network & Deep Learning](https://www.analyticsvidhya.com/blog/2015/07/top-youtube-videos-machine-learning-neural-network-deep-learning/)
* [16 New Must Watch Tutorials, Courses on Machine Learning](https://www.analyticsvidhya.com/blog/2016/10/16-new-must-watch-tutorials-courses-on-machine-learning/)
* [DeepLearning.TV](https://www.youtube.com/channel/UC9OeZkIwhzfv-_Cb7fCikLQ)
* [Learning To See](https://www.youtube.com/playlist?list=PLiaHhY2iBX9ihLasvE8BKnS2Xg8AhY6iV)
* [Neural networks class - Université de Sherbrooke](https://www.youtube.com/playlist?list=PL6Xpj9I5qXYEcOhn7TqghAJ6NAPrNmUBH)
* [21 Deep Learning Videos, Tutorials & Courses on Youtube from 2016](https://www.analyticsvidhya.com/blog/2016/12/21-deep-learning-videos-tutorials-courses-on-youtube-from-2016/)
* [30 Top Videos, Tutorials & Courses on Machine Learning & Artificial Intelligence from 2016](https://www.analyticsvidhya.com/blog/2016/12/30-top-videos-tutorials-courses-on-machine-learning-artificial-intelligence-from-2016/)
* [Practical Deep Learning For Coders](http://course.fast.ai/index.html)
* [Practical Deep Learning For Coders Version 2 (PyTorch)](http://forums.fast.ai/t/welcome-to-part-1-v2/5787)

## MOOC

* [Coursera’s AI For Everyone](https://www.coursera.org/learn/ai-for-everyone)
* [edX's Introduction to Artificial Intelligence (AI)](https://www.edx.org/course/introduction-artificial-intelligence-ai-microsoft-dat263x)
* [Udacity’s Intro to Machine Learning](https://www.udacity.com/course/intro-to-machine-learning--ud120)
  + [Udacity Intro to Machine Learning Review](http://hamelg.blogspot.com/2014/12/udacity-intro-to-machine-learning-review.html)
* [Udacity’s Supervised, Unsupervised & Reinforcement](https://www.udacity.com/course/machine-learning--ud262)
* [Machine Learning Foundations: A Case Study Approach](https://www.coursera.org/learn/ml-foundations)
* [Machine Learning & AI Foundations: Value Estimations](https://www.lynda.com/Data-Science-tutorials/Machine-Learning-Essential-Training-Value-Estimations/548594-2.html)
* [Kaggle's Hands-On Data Science Education](https://www.kaggle.com/learn/overview)
* [Microsoft Professional Program for Artificial Intelligence](https://academy.microsoft.com/en-us/professional-program/tracks/artificial-intelligence/)
* [Coursera’s Machine Learning](https://www.coursera.org/learn/machine-learning)
  + [Video only](https://www.youtube.com/playlist?list=PLZ9qNFMHZ-A4rycgrgOYma6zxF4BZGGPW)
  + [Coursera Machine Learning review](https://rayli.net/blog/data/coursera-machine-learning-review/)
  + [Coursera: Machine Learning Roadmap](https://metacademy.org/roadmaps/cjrd/coursera_ml_supplement)
* [Machine Learning Distilled](https://code.tutsplus.com/courses/machine-learning-distilled)
* [BigML training](https://bigml.com/training)
* [Coursera’s Neural Networks for Machine Learning](https://www.coursera.org/learn/neural-networks)
  + Taught by Geoffrey Hinton, a pioneer in the field of neural networks
* [Machine Learning - CS - Oxford University](https://www.cs.ox.ac.uk/people/nando.defreitas/machinelearning/)
* [Creative Applications of Deep Learning with TensorFlow](https://www.kadenze.com/courses/creative-applications-of-deep-learning-with-tensorflow/info)
* [Intro to Descriptive Statistics](https://www.udacity.com/course/intro-to-descriptive-statistics--ud827)
* [Intro to Inferential Statistics](https://www.udacity.com/course/intro-to-inferential-statistics--ud201)
* [6.S094: Deep Learning for Self-Driving Cars](http://selfdrivingcars.mit.edu/)
* [6.S191: Introduction to Deep Learning](http://introtodeeplearning.com/index.html)
* [Coursera’s Deep Learning](https://www.coursera.org/specializations/deep-learning)

## Resources

* [Absolute Beginning into Machine Learning](https://hackernoon.com/absolute-beginning-into-machine-learning-e90ceda5a4bc)
* [Learn Machine Learning in a Single Month](https://elitedatascience.com/machine-learning-masterclass)
* [The Non-Technical Guide to Machine Learning & Artificial Intelligence](https://medium.com/@samdebrule/a-humans-guide-to-machine-learning-e179f43b67a0#.cpzf3a5c0)
* [Programming Community Curated Resources for learning Machine Learning](https://hackr.io/tutorials/learn-machine-learning-ml)
* [Best practices rule book for Machine Learning engineering from Google](http://martin.zinkevich.org/rules_of_ml/rules_of_ml.pdf)
* [Machine Learning for Software Engineers on Hacker News](https://news.ycombinator.com/item?id=12898718)
* [Machine Learning for Developers](https://xyclade.github.io/MachineLearning/)
* [Machine Learning for Humans🤖👶](https://medium.com/machine-learning-for-humans/why-machine-learning-matters-6164faf1df12)
* [Machine Learning Advice for Developers](https://dev.to/thealexlavin/machine-learning-advice-for-developers)
* [Machine Learning For Complete Beginners](http://pythonforengineers.com/machine-learning-for-complete-beginners/)
* [Getting Started with Machine Learning: For absolute beginners and fifth graders](https://medium.com/@suffiyanz/getting-started-with-machine-learning-f15df1c283ea#.yjtiy7ei9)
* [How to Learn Machine Learning: The Self-Starter Way](https://elitedatascience.com/learn-machine-learning)
* [Machine Learning Self-study Resources](https://ragle.sanukcode.net/articles/machine-learning-self-study-resources/)
* [Level-Up Your Machine Learning](https://metacademy.org/roadmaps/cjrd/level-up-your-ml)
* [An Honest Guide to Machine Learning](https://medium.com/axiomzenteam/an-honest-guide-to-machine-learning-2f6d7a6df60e#.ib12a1yw5)
* Enough Machine Learning to Make Hacker News Readable Again
  + [Video](https://www.youtube.com/watch?v=O7IezJT9uSI)
  + [Slide](https://speakerdeck.com/pycon2014/enough-machine-learning-to-make-hacker-news-readable-again-by-ned-jackson-lovely)
* [Dive into Machine Learning](https://github.com/hangtwenty/dive-into-machine-learning)
* [{Machine, Deep} Learning for software engineers](https://speakerdeck.com/pmigdal/machine-deep-learning-for-software-engineers)
* [Deep Learning For Beginners](https://deeplearning4j.org/deeplearningforbeginners.html)
* [Foundations for deep learning](https://github.com/pauli-space/foundations_for_deep_learning)
* [Machine Learning Mindmap / Cheatsheet](https://github.com/dformoso/machine-learning-mindmap)
* Machine Learning courses in Universities
  + [Stanford](http://ai.stanford.edu/courses/)
  + [Machine Learning Summer Schools](http://mlss.cc/)
  + [Oxford](https://www.cs.ox.ac.uk/people/nando.defreitas/machinelearning/)
  + [Cambridge](http://mlg.eng.cam.ac.uk/)
* Flipboard Topics
  + [Machine learning](https://flipboard.com/topic/machinelearning)
  + [Deep learning](https://flipboard.com/topic/deeplearning)
  + [Artificial Intelligence](https://flipboard.com/topic/artificialintelligence)
* Medium Topics
  + [Machine learning](https://medium.com/tag/machine-learning/latest)
  + [Deep learning](https://medium.com/tag/deep-learning)
  + [Artificial Intelligence](https://medium.com/tag/artificial-intelligence)
* Monthly top 10 articles
  + [Machine Learning](https://medium.mybridge.co/search?q=%22Machine%20Learning%22)
  + [Algorithms](https://medium.mybridge.co/search?q=Algorithms)
* [Comprehensive list of data science resources](http://www.datasciencecentral.com/group/resources/forum/topics/comprehensive-list-of-data-science-resources)
* [DigitalMind's Artificial Intelligence resources](http://blog.digitalmind.io/post/artificial-intelligence-resources)
* [Awesome Machine Learning](https://github.com/josephmisiti/awesome-machine-learning)
* [Awesome Graph Classification](https://github.com/benedekrozemberczki/awesome-graph-classification)
* [Awesome Community Detection](https://github.com/benedekrozemberczki/awesome-community-detection)
* [CreativeAi's Machine Learning](http://www.creativeai.net/?cat%5B0%5D=machine-learning)
* [Machine Learning Online Courses](https://classpert.com/machine-learning)

## Games

* [Halite: A.I. Coding Game](https://halite.io/)
* [Vindinium: A.I. Programming Challenge](http://vindinium.org/)
* [General Video Game AI Competition](http://www.gvgai.net/)
* [Angry Birds AI Competition](https://aibirds.org/)
* [The AI Games](http://theaigames.com/)
* [Fighting Game AI Competition](http://www.ice.ci.ritsumei.ac.jp/~ftgaic/)
* [CodeCup](http://www.codecup.nl/intro.php)
* [Student StarCraft AI Tournament](http://sscaitournament.com/)
* [AIIDE StarCraft AI Competition](http://www.cs.mun.ca/~dchurchill/starcraftaicomp/)
* [CIG StarCraft AI Competition](https://sites.google.com/site/starcraftaic/)
* [CodinGame - AI Bot Games](https://www.codingame.com/training/machine-learning)

## Becoming an Open Source Contributor

* [tensorflow/magenta: Magenta: Music and Art Generation with Machine Intelligence](https://github.com/tensorflow/magenta)
* [tensorflow/tensorflow: Computation using data flow graphs for scalable machine learning](https://github.com/tensorflow/tensorflow)
* [cmusatyalab/openface: Face recognition with deep neural networks.](https://github.com/cmusatyalab/openface)
* [tensorflow/models/syntaxnet: Neural Models of Syntax.](https://github.com/tensorflow/models/tree/master/syntaxnet)

## Podcasts

### Podcasts for Beginners:

* + [Talking Machines](http://www.thetalkingmachines.com/)
  + [Linear Digressions](http://lineardigressions.com/)
  + [Data Skeptic](http://dataskeptic.com/)
  + [This Week in Machine Learning & AI](https://twimlai.com/)
  + [Machine Learning Guide](http://ocdevel.com/podcasts/machine-learning)

### Interviews with ML Practitioners, Researchers and Kagglers about their Joureny

* + [Chai Time Data Science](https://www.youtube.com/playlist?list=PLLvvXm0q8zUbiNdoIazGzlENMXvZ9bd3x), [Audio](http://anchor.fm/chaitimedatascience), [Writeups](https://sanyambhutani.com/tag/chaitimedatascience/)

### "More" advanced podcasts

* + [Partially Derivative](http://partiallyderivative.com/)
  + [O’Reilly Data Show](http://radar.oreilly.com/tag/oreilly-data-show-podcast)
  + [Not So Standard Deviation](https://soundcloud.com/nssd-podcast)

### Podcasts to think outside the box:

* + [Data Stories](http://datastori.es/)

## Communities

* Quora
  + [Machine Learning](https://www.quora.com/topic/Machine-Learning)
  + [Statistics](https://www.quora.com/topic/Statistics-academic-discipline)
  + [Data Mining](https://www.quora.com/topic/Data-Mining)
* Reddit
  + [Machine Learning](https://www.reddit.com/r/machinelearning)
  + [Computer Vision](https://www.reddit.com/r/computervision)
  + [Natural Language](https://www.reddit.com/r/languagetechnology)
  + [Data Science](https://www.reddit.com/r/datascience)
  + [Big Data](https://www.reddit.com/r/bigdata)
  + [Statistics](https://www.reddit.com/r/statistics)
* [Data Tau](http://www.datatau.com/)
* [Deep Learning News](http://news.startup.ml/)
* [KDnuggets](http://www.kdnuggets.com/)

## Conferences

* Neural Information Processing Systems ([NIPS](https://nips.cc/))
* International Conference on Learning Representations ([ICLR](http://www.iclr.cc/doku.php?id=ICLR2017:main&redirect=1))
* Association for the Advancement of Artificial Intelligence ([AAAI](http://www.aaai.org/Conferences/AAAI/aaai17.php))
* IEEE Conference on Computational Intelligence and Games ([CIG](http://www.ieee-cig.org/))
* IEEE International Conference on Machine Learning and Applications ([ICMLA](http://www.icmla-conference.org/))
* International Conference on Machine Learning ([ICML](https://2017.icml.cc/))
* International Joint Conferences on Artificial Intelligence ([IJCAI](http://www.ijcai.org/))
* Association for Computational Linguistics ([ACL](http://acl2017.org/))

## Interview Questions

* [How To Prepare For A Machine Learning Interview](http://blog.udacity.com/2016/05/prepare-machine-learning-interview.html)
* [40 Interview Questions asked at Startups in Machine Learning / Data Science](https://www.analyticsvidhya.com/blog/2016/09/40-interview-questions-asked-at-startups-in-machine-learning-data-science)
* [21 Must-Know Data Science Interview Questions and Answers](http://www.kdnuggets.com/2016/02/21-data-science-interview-questions-answers.html)
* [Top 50 Machine learning Interview questions & Answers](http://career.guru99.com/top-50-interview-questions-on-machine-learning/)
* [Machine Learning Engineer interview questions](https://resources.workable.com/machine-learning-engineer-interview-questions)
* [Popular Machine Learning Interview Questions](http://www.learn4master.com/machine-learning/popular-machine-learning-interview-questions)
* [What are some common Machine Learning interview questions?](https://www.quora.com/What-are-some-common-Machine-Learning-interview-questions)
* [What are the best interview questions to evaluate a machine learning researcher?](https://www.quora.com/What-are-the-best-interview-questions-to-evaluate-a-machine-learning-researcher)
* [Collection of Machine Learning Interview Questions](http://analyticscosm.com/machine-learning-interview-questions-for-data-scientist-interview/)
* [121 Essential Machine Learning Questions & Answers](https://elitedatascience.com/mlqa-reading-list)
* [Minimum Viable Study Plan for Machine Learning Interviews](https://github.com/khangich/machine-learning-interview)

## My admired companies

* [ELSA - Your virtual pronunciation coach](https://www.elsanow.io/home)