# **Roadmap for learning Machine Learning (More focus on DL):**

## : Andrew Ng Coursera Course

* [Coursera](https://www.coursera.org/learn/machine-learning).
* [Youtube](https://www.youtube.com/playlist?list=PLLssT5z_DsK-h9vYZkQkYNWcItqhlRJLN).
* [Python implementation.](https://medium.com/analytics-vidhya/python-implementation-of-andrew-ngs-machine-learning-course-part-1-6b8dd1c73d80)

## Kaggle :

* Browse scikit-learn library - use their NN tool (and others if you want).
* [Getting started with kaggle](https://blogs.mathworks.com/loren/2015/06/18/getting-started-with-kaggle-data-science-competitions/).
* [Titanic (classification)](https://www.kaggle.com/c/titanic).
* [Boston housing prices competitions (regression).](https://www.kaggle.com/c/house-prices-advanced-regression-techniques)

## Neural Network (NN) :

* [Calltech youtube.](https://www.youtube.com/watch?v=Ih5Mr93E-2c)
* [MIT youtube.](https://www.youtube.com/watch?v=uXt8qF2Zzfo)
* [Stanford youtube.](https://www.youtube.com/watch?v=d14TUNcbn1k)
* [Backpropagation step by step.](https://mattmazur.com/2015/03/17/a-step-by-)
* Reimplement NN from scratch.
* Apply them to Kaggle’s titanic (classification) and boston housing prices (regression) courses.
* Customize them to support Multi-class classification (softmax).
* Customize them to support Multi-label classification (sigmoid).
* Do regression using them.
* Try to make Auto Encoder using them.
* Learn and Implement Dropout layer (their forward and backward propagation.
* [Arabic playlist](https://www.youtube.com/playlist?list=PLQkyODvJ8ywsLydDYORIlJxV9KarhXp9O) (Neural Networks in Arabic).
* Try to read a chapter about NN (for extra help).

## Evaluation Metrics:

* Learn [those](https://towardsdatascience.com/metrics-to-evaluate-your-machine-learning-algorithm-f10ba6e38234).
* Then learn [this](https://medium.com/@jonathan_hui/map-mean-average-precision-for-object-detection-45c121a31173).

## AFTER you finish the above, do the following time split strategy:

* 25% of time for classical/projects .
* 75% for deep learning.

## 25% of your time for classical ML :

* Do kaggle projects.
* Browse kaggle blogs to learn from others (how they solved particular problems).
* Go deeper in ML, study [Andrew’s specialization course](https://www.coursera.org/specializations/aml).
* Try to get some experience in Feature Engineering.
* Read Google’s ML small [book](https://github.com/hktxt/bookshelf/blob/master/Machine%20Leraning/Rules%20of%20Machine%20Learning%20Best%20Practices%20for%20ML%20Engineering.pdf) ( about their experience in Feature Engineering)
* Read “Feature Engineering for Machine Learning: Principles and Techniques for Data Scientists” book.
* After doing the above you can go deeper (SVM, Random forest trees, pose estimation and so on).

## 75% of your time for deep learning :

* Study [Deep Learning Specialization course](https://www.coursera.org/specializations/deep-learning) as a first step.
* You can look at [Anrew 's Deep Learning playlists](https://www.youtube.com/channel/UCcIXc5mJsHVYTZR1maL5l9w/playlists) if you want (not necessary).
* You can also look at [this](https://classroom.udacity.com/courses/ud730) if you want(not necessary).
* Learn some Frameworks (TF/Pytorch).
* Learn Either 2D Vision or NLP and learn the other after finishing the first.
* [Vision Stanford Course.](http://cs231n.stanford.edu/syllabus.html)
* [NLP Stanford Course.](http://web.stanford.edu/class/cs224n/)
* Computer vision typical problems like(classification, detection, segmentation, pose estimation, object traction and action detection/recognition).
* Study every track’s problems, implement them..(this takes a significant amount of time).

## After doing the above : Basics + 25% ML and 75% DL do some projects then:

* Read more: A fundamental book (may be heavy but necessary).
* ML books like ( Bishop’s [book](https://www.amazon.com/Pattern-Recognition-Learning-Information-Statistics/dp/0387310738) , Kalivin Murphy’s [book](https://www.amazon.com/Machine-Learning-Probabilistic-Perspective-Computation/dp/0262018020) or Analysis of statistical learning book).
* Understand more models, fields, more experience.
* Field follow up: E.g. in Vision (CVPR, ICCV, ECCV).
* DL Experience: Paper-to-code skill ( find papers and code them, better if their code is available and published).
* Build up the abstracting skills to learn new technologies faster.
* Vision elements (later): look at [this](https://classroom.udacity.com/courses/ud810).
* Other ML areas (later).
* Generative models: PGM, EM (GMM), VI, MCMC)
* Generative model DL: (VAE, GAN - Study GAN)
* GAN’s applications are like [this](https://medium.com/@jonathan_hui/gan-some-cool-applications-of-gans-4c9ecca35900) and [that](https://github.com/nashory/gans-awesome-applications).
* Reinforcement Learning.

## 

## What if you are also interested in competitive programming? :

* ML is time-consuming to be good. In this case, divide your vacation to 50% competitive and 50% for ML.