See https://github.com/skorsens/ai-foundations-ml

# ML types

* Supervised learning.
  + Tains the model with labeled data – known inputs and outputs. E.g. houses prices vs. house features; or flower images vs flower types.
  + Regression problems – predicting a continuous value. House pricse is a regression problem that predicts the value of a house.
  + Classification problems – predict discrete values, tag data, classify data instance. E.g. given a flower image define its type.
* Unsupervised learning.
  + Does not use labeled data, instead it groups data based on the input only.
  + Clustering. E.g. group customers by their preferences.
* Reinforcement learning. Trial and error learning. Does not rely on a defined data set, but operates in a dynamic environment where the model learns from the experience. Used in robotics, board games

# ML Life Cycle

## Problem formation and understanding.

* Define a problem to solve and if ML can provide a solution.
* Define the inputs and outputs and prediction error rates.

When to use ML:

* Predict an outcome
* Uncover trends and patterns in data
* Have rules and steps that can’t be coded
* The dataset is too large to process by a human.

Frame the problem to solve

* Define the questions for the model.
* Define the inputs (features) and outputs. The outputs can define the learning algorithm.
* Define the success metrics (e.g. how faster a house shall be sold).
* Define the costs for running and maintaining the model.

## Data collection and preparation.

* Source the raw data required.
* Prepare the data: label the data, remove irrelevant features and outliers; transform the data; input missing values.
* Split the data into training (80%), validation (10%), testing (10%).
* Select the ML algorithm. Often the required output defines the learning algorithm.

## Model training and testing.

* Select if to train the model from scratch or if to use a pretrained model (transfer learning – adding a new data on top of a pre-trained model).
  + [ModelZoo](https://modelzoo.co/) is the pre-trained model catalog.
  + [AWS SageMaker](https://aws.amazon.com/sagemaker-ai/?trk=af448c55-a335-4b68-909c-3589fef8d1a7&sc_channel=ps&ef_id=Cj0KCQjwmqPDBhCAARIsADorxIY2FiCdLwA1Fx39Z_xtgqUwESU3YdqdLZzan39i8x--Ctp8aoCp2UAaAj2GEALw_wcB:G:s&s_kwcid=AL!4422!3!724139679532!p!!g!!amazon%20sagemaker%20ai!19574556908!168879877422&gad_campaignid=19574556908&gbraid=0AAAAADjHtp9PA4wy1DgjnimsBImyuWPFm&gclid=Cj0KCQjwmqPDBhCAARIsADorxIY2FiCdLwA1Fx39Z_xtgqUwESU3YdqdLZzan39i8x--Ctp8aoCp2UAaAj2GEALw_wcB).
  + <https://huggingface.co/>.
* Try the learning algorithm on the data trying to produce a model.

## Model deployment and maintenance.

* Deploy the model. Retrain the model and monitor its performance.

# Python libraries for ML

* Pandas. Data structures, matrices, data analysis, reading data from files.
* Numpy. Numerical data processing in arrays and matrices.
* Matplotlib, Seaborn. Visualization.
* Scikit-learn, TensorFlow, MXNext, PyTorch, Keras. Learning algorithms.

# Preparing data for ML

## Obtaining data

Data sources: internal; from clients; open-source; public datastore; commercial.

### E.g. problem. Use supervised learning linear regression to predict the house prices

Features: location (long, lat), house age, total rooms, bedrooms, population, house value etc…

House value is the target output.