Introduction to Machine Learning

This post does not explain what ML is and the usual stuff. Rather it mentions some basic details about ML. This is based on Aurelion’s ML book.

# Types of ML

Based on how they are classified, below are some classifications

* Whether or not they are trained with human supervision
  + Supervised
  + Unsupervised
  + Semi supervised
  + Reinforcement Learning
* Whether or not they can learn incrementally on the fly
  + Online
  + batch learning
* Whether they work by simply comparing new data points to known data points, or instead by detecting patterns in the training data and building a predictive model, much like scientists do
  + instance-based
  + model-based learning

## Common Supervised Learning Algorithms

* k-Nearest Neighbors
* Linear Regression
* Logistic Regression
* Support Vector Machines (SVMs)
* Decision Trees and Random Forests
* Neural networks

## Common Unsupervised Learning Algorithms

* Clustering
  + K-Means
  + DBSCAN
  + Hierarchical Cluster Analysis (HCA)
* Anomaly detection and novelty detection
  + One-class SVM
  + Isolation Forest
* Visualization and dimensionality reduction
  + Principal Component Analysis (PCA)
  + Kernel PCA
  + Locally Linear Embedding (LLE)
  + t-Distributed Stochastic Neighbor Embedding (t-SNE)
* Association rule learning
  + Apriori
  + Eclat

## Most semi supervised learning algorithms are combinations of unsupervised and supervised algorithms

For example, *deep belief networks* (DBNs) are based on unsupervised components called *restricted Boltzmann machines* (RBMs) stacked on top of one another. RBMs are trained sequentially in an unsupervised manner, and then the whole system is fine-tuned using supervised learning techniques.

## Reinforcement Learning Algorithms

There are no common algorithms. These work on Reward and Penalties and the learning happens over time by running it on multiple real-life examples

* Algorithms which play Chess or Go are an example
* Programs used in Robots are another example

## Batch and Online Learning algorithms

As the name says there are algorithms which have to be trained offline i.e. Batch Algorithms and algorithms which can learn on the fly i.e. Online algorithms

## Instance vs Model algorithms

Depending on whether the algorithm uses learned instances to predict for new inputs like say Classification Algorithms (k-NearestNeighbors for example) or uses a Model like say Regression Algorithms where you have a line/plane.

# Challenges in ML

* Bad Data
* Bad Model

## What to do with the Bad data or Bad model?

* Feature Selection
* Feature Extraction
* Regularization
* Hyperparameters

## Data Load

Below function will be useful to get data from online datasets.

**import** **os**

**import** **tarfile**

**import** **urllib**

DOWNLOAD\_ROOT = "https://raw.githubusercontent.com/ageron/handson-ml2/master/"

HOUSING\_PATH = os.path.join("datasets", "housing")

HOUSING\_URL = DOWNLOAD\_ROOT + "datasets/housing/housing.tgz"

**def** fetch\_housing\_data(housing\_url=HOUSING\_URL, housing\_path=HOUSING\_PATH):

os.makedirs(housing\_path, exist\_ok=True)

tgz\_path = os.path.join(housing\_path, "housing.tgz")

urllib.request.urlretrieve(housing\_url, tgz\_path)

housing\_tgz = tarfile.open(tgz\_path)

housing\_tgz.extractall(path=housing\_path)

housing\_tgz.close()

Look at the next few posts on ML End to End process for more information.