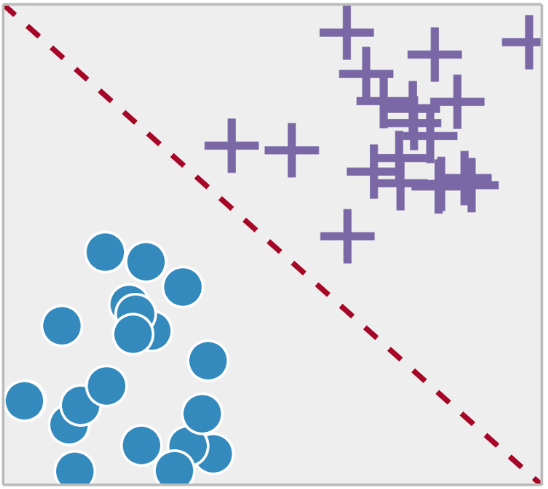
MACHINE **LEARNING**

# Supervised Learning

1. Build the Model
2. Train the Model
3. Test the Model

## Classification

🡪 Entrainer un modèle pour classifier des choses et leur associer des labels.



### Support vector machines (SVM)

### Decision Tree

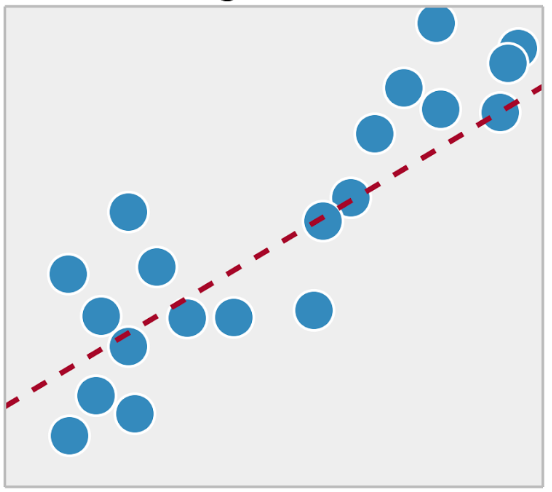
### Neural Networks

### Nearest neighbors (KNN)

### Naïve Bayes classifier

## Regression

🡪 Prédiction de valeurs



### Linear regression

### Nonlinear regression

### Support Vector Regression (SVR)

### Partial Least Squares (PLS)

### Neural Network

# Unsupervised Learning

## Clustering

### K-Means Clustering

### Hierarchical Clustering

### Density based Clustering

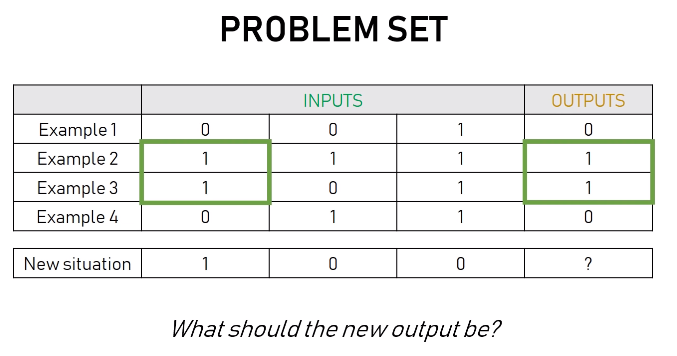
## Association

# Semi-supervised learning

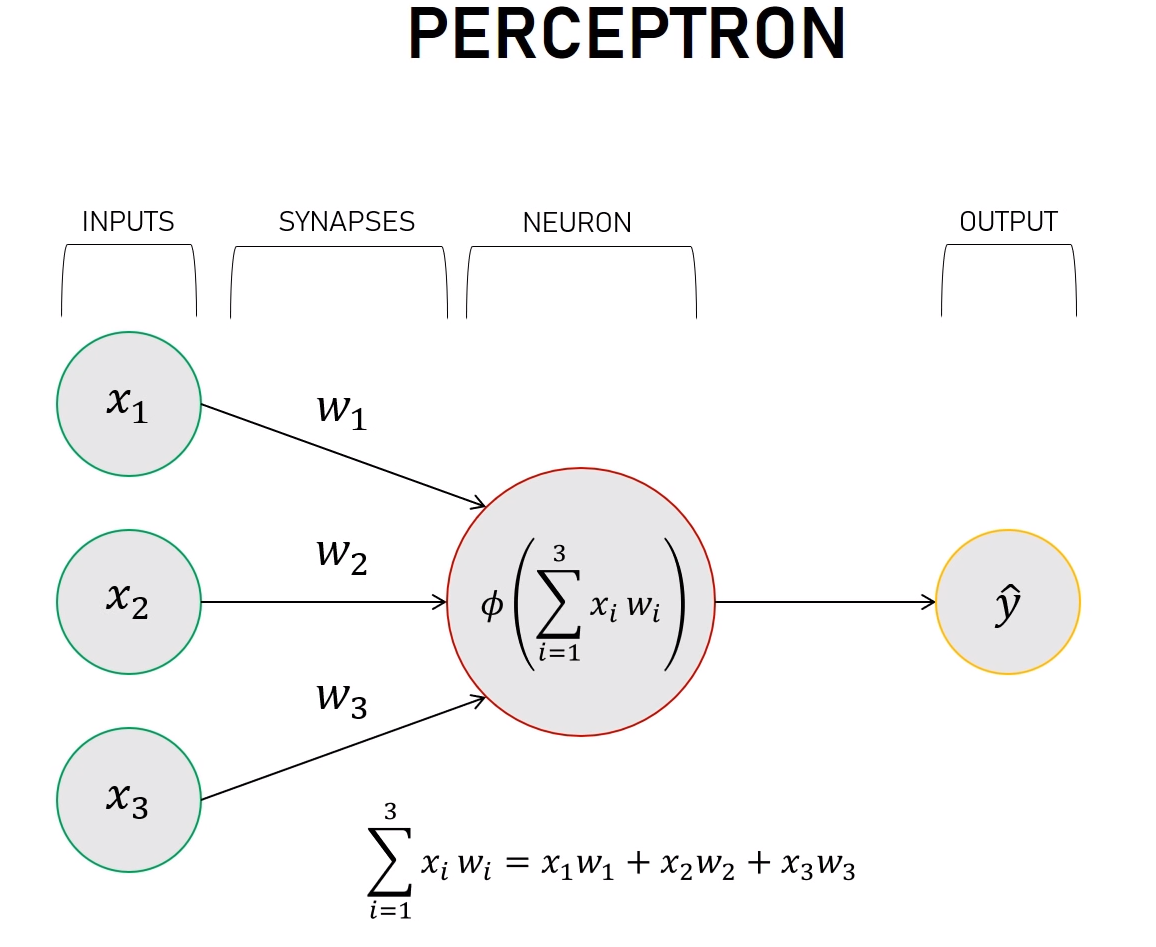
# Reinforcement Learning

DEEP LEARNING

# Neural Network

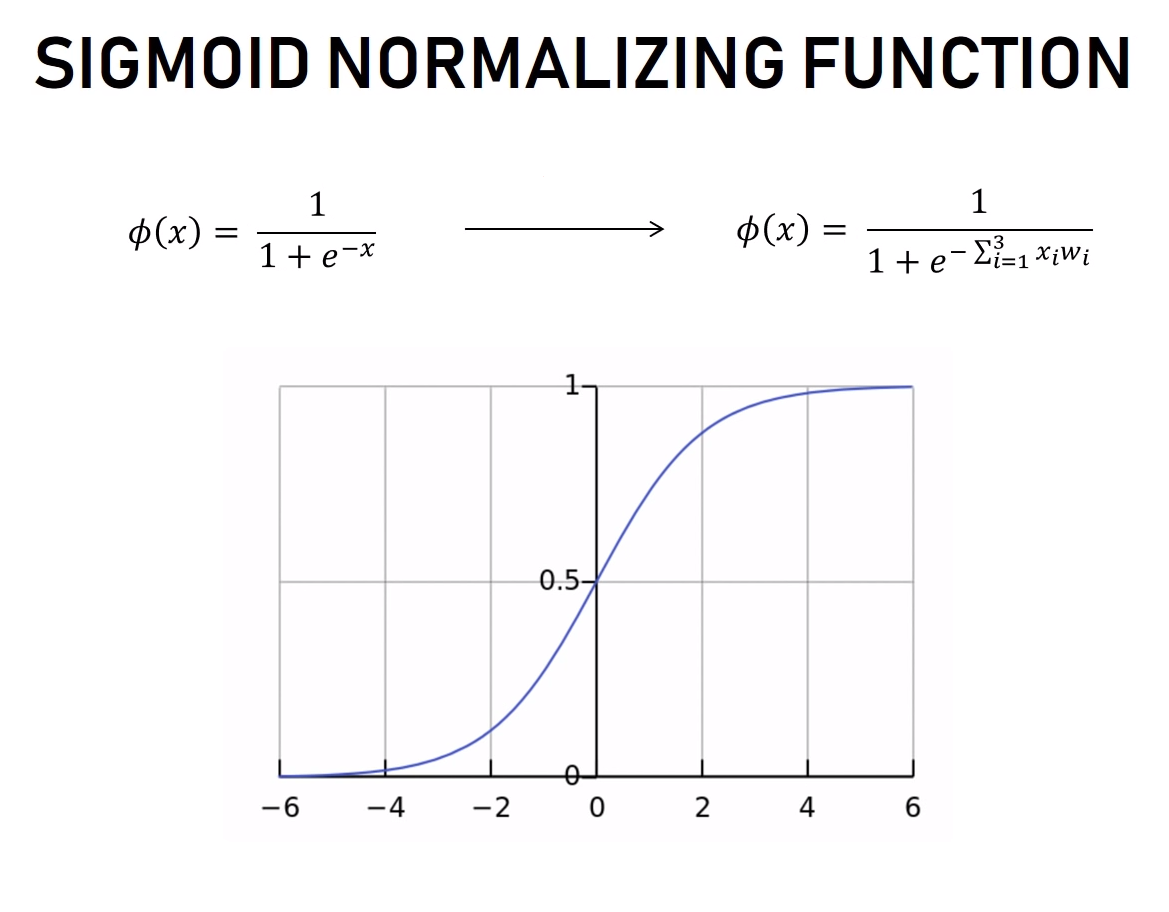


## Perceptron



## Activation and normalizing function

### Sigmoid



### Softmax

## Forward propagation

## Back propagation

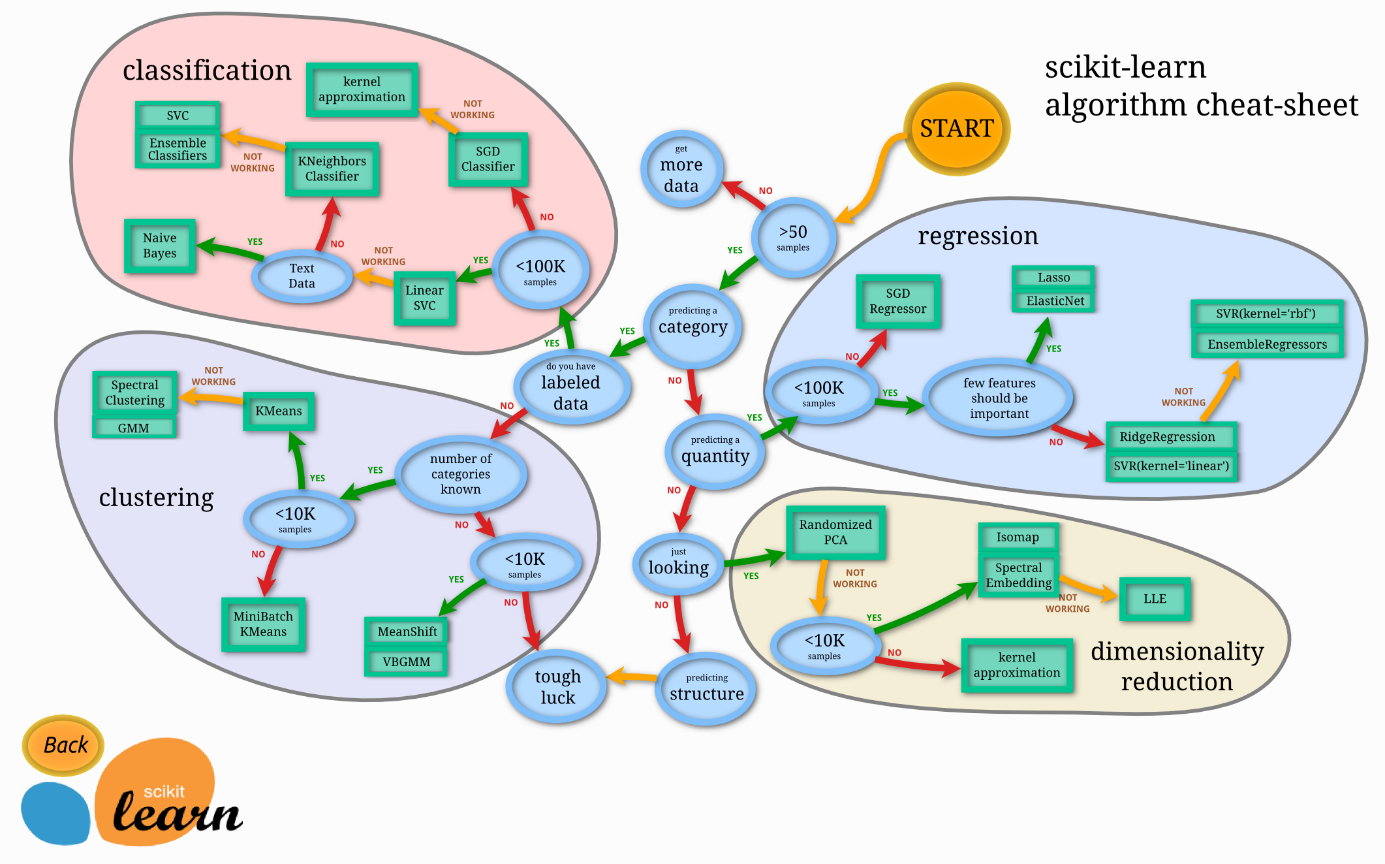
PYTHON

# Scikit

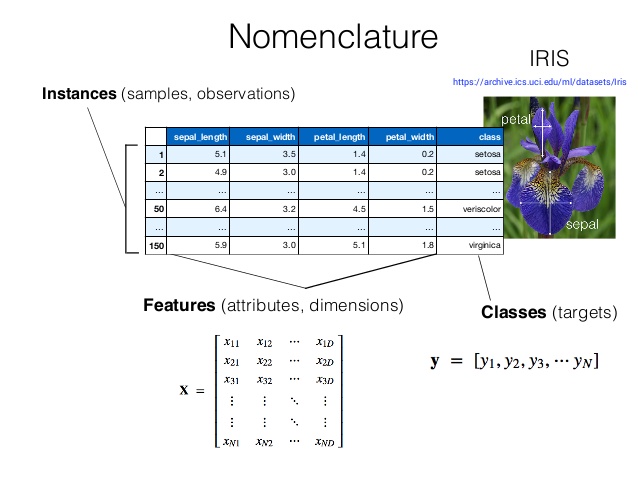
🡪 Scikit-learn is a machine learning library for the Python programming language.It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

* **NumPy** : Base n-dimensional array package
* **SciPy** : Fundamental library for scientific computing
* **Matplotlib** : Comprehensive 2D/3D plotting
* **IPython** : Enhanced interactive console
* **Sympy** : Symbolic mathematics
* **Pandas** : Data structures and analysis

## Algorithm cheat-sheet



## Dataset



# Tensorflow

🡪 Machine Learning framework created by Google for creating Deep Learning models.

## Modèle pré-entrainé

### MobilNet

[*https://ai.googleblog.com/2017/06/mobilenets-open-source-models-for.html*](https://ai.googleblog.com/2017/06/mobilenets-open-source-models-for.html)

### Retrain with new images

[*https://codelabs.developers.google.com/codelabs/tensorflow-for-poets/index.html#0*](https://codelabs.developers.google.com/codelabs/tensorflow-for-poets/index.html#0)

IMAGE\_SIZE=224

ARCHITECTURE="mobilenet\_0.50\_${IMAGE\_SIZE}"

python -m scripts.retrain

--bottleneck\_dir=tf\_files/bottlenecks

--how\_many\_training\_steps=500

--model\_dir=tf\_files/models/

--summaries\_dir=tf\_files/training\_summaries/"${ARCHITECTURE}"

--output\_graph=tf\_files/retrained\_graph.pb

--output\_labels=tf\_files/retrained\_labels.txt

--architecture="${ARCHITECTURE}"

--image\_dir=tf\_files/flower\_photos

### Tensorboard

python -m tensorboard.main --logdir tf\_files/training\_summaries &

### Test

python -m scripts.label\_image

--graph=tf\_files/retrained\_graph.pb

--image=tf\_files/flower\_photos/daisy/3475870145\_685a19116d.jpg

### Convert for mobile with TOCO (TensorFlow Lite)

[*https://codelabs.developers.google.com/codelabs/tensorflow-for-poets-2-tflite/#0*](https://codelabs.developers.google.com/codelabs/tensorflow-for-poets-2-tflite/#0)

/home/patrick/.local/bin/toco

--input\_file=tf\_files/retrained\_graph.pb

--output\_file=tf\_files/optimized\_graph.lite

--input\_format=TENSORFLOW\_GRAPHDEF

--output\_format=TFLITE

--input\_shape=1,224,224,3

--input\_array=input

--output\_array=final\_result

--inference\_type=FLOAT

--input\_data\_type=FLOAT

🡪 optimized\_graph.lite and retrained\_labels

### Android App (Android Studio)

Replace graph.lite and labels.txt by optimized\_graph.lite and retrained\_labels