

Le Machine Learning au service du Data Scientist



Les outils pour exploiter les données

La fois précédente

- Principes du Machine Learning
- Différentes familles de modèles
 - Supervisé
 - Non-supervisé
- Forte dépendance aux données

Workflow

1. Identifier une problématique, une question métier
2. Identifier les données pertinentes
3. Valider les données
 1. Nettoyage
 2. Preprocessing et feature engineering
4. Formaliser le use-case
 1. Modèle prédictif (ML)
 2. Protocole de validation
5. Industrialisation

Moteur de recommandations

L'archétype du produit data

Objectifs et enjeux

Client

Augmenter les revenus à travers la hausse de la valeur moyenne du panier.

Utilisateur

Avoir des propositions pertinentes et éviter la surcharge d'informations.

Exploiter les données pendant que le visiteur est sur le site

Données disponibles et informations exploitables

Session en cours



Intérêt immédiat

Historique d'achats



Profil général et
centres d'intérêt

Historique de navigation

Fiches produits



Connaissance
produits

Stratégies de recommandation

Intérêt immédiat



Profil général et
centres d'intérêt



Utilisateurs
similaires

Intérêt immédiat



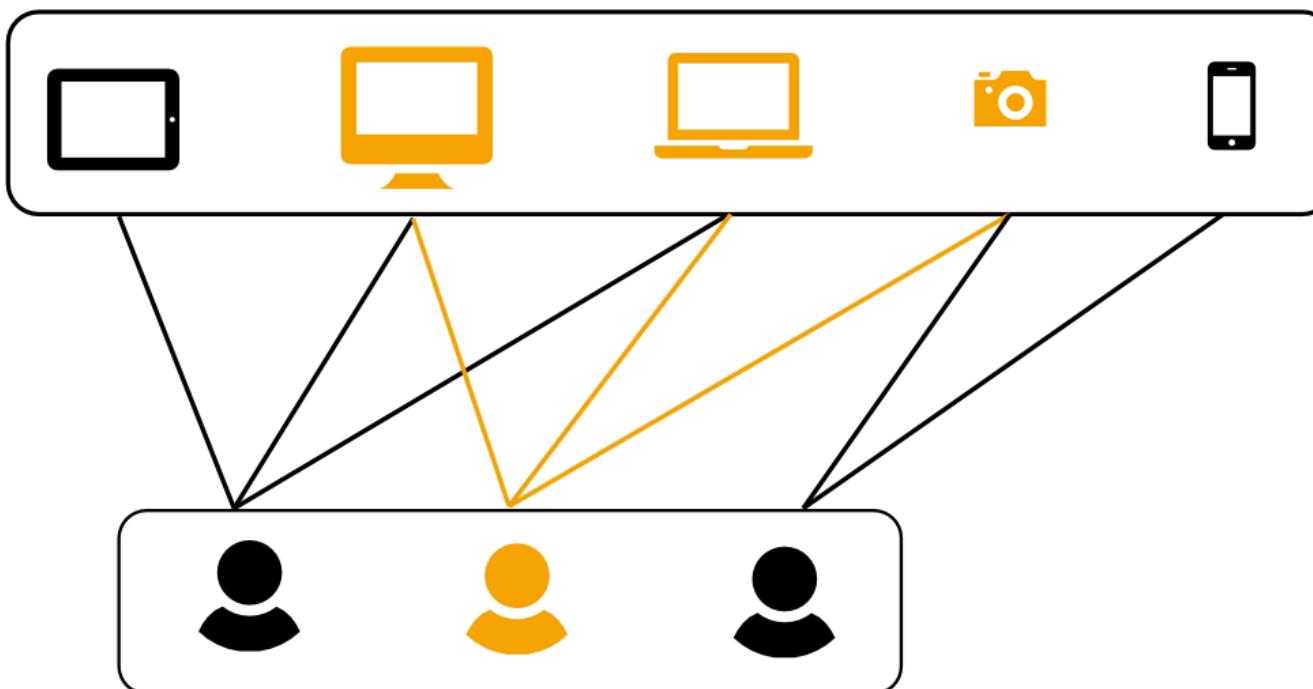
Fiches produits



Produits similaires

Formalisation mathématique

user	?	1	1	1	?
------	---	---	---	---	---



Matrice Utilisateurs - Produits

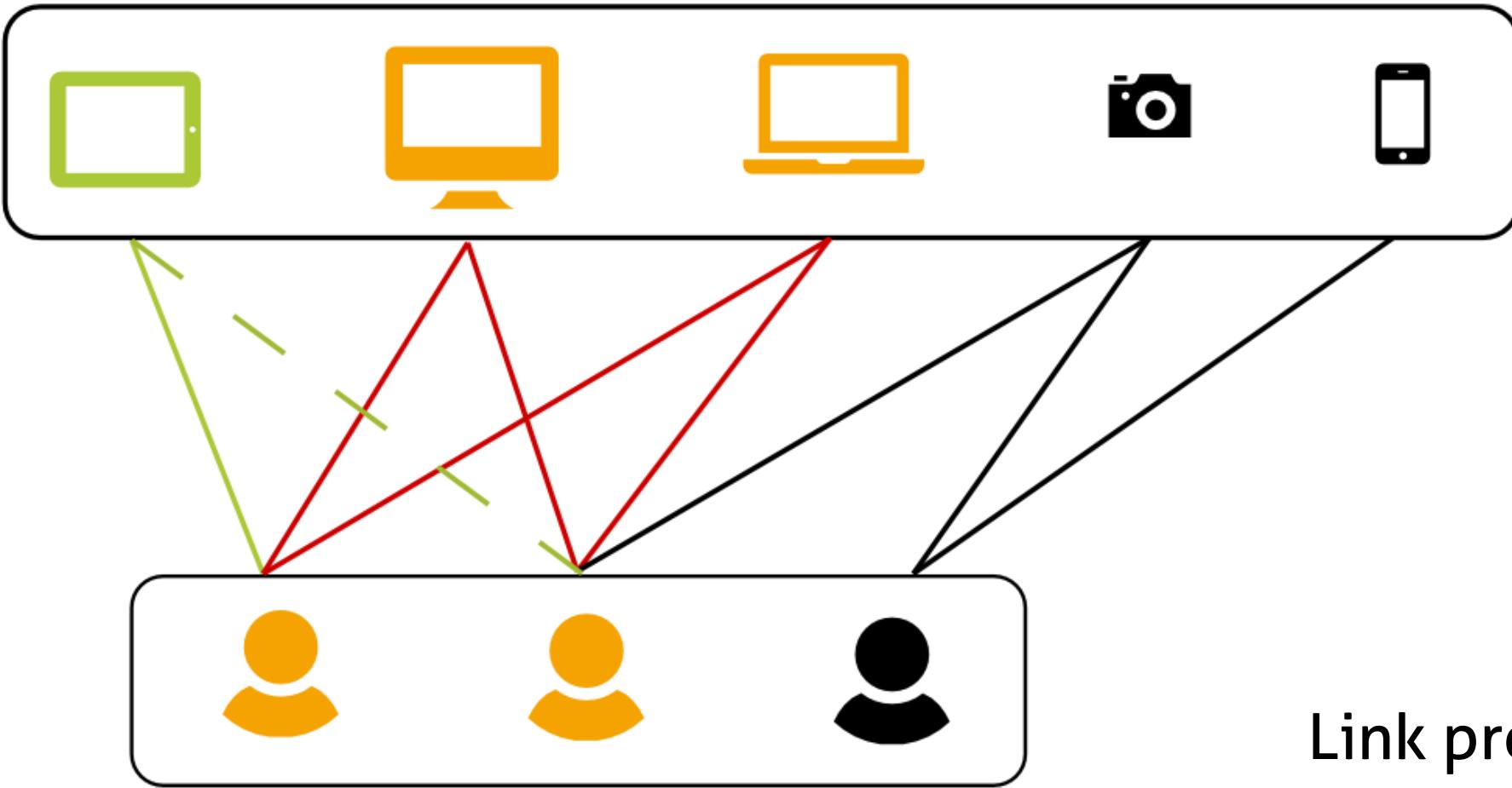
	item 1	item 2	item 3	item 4	item 5
A	1	1	1		
B		1	1	1	
C				1	1
.			.	.	.
.					
.					

Matrice Utilisateurs - Produits

	item 1	item 2	item 3	item 4	item 5
A	1	1	1	?	?
B	?	1	1	1	?
C	?	?	?	1	1
.			.	.	.
.					
.					

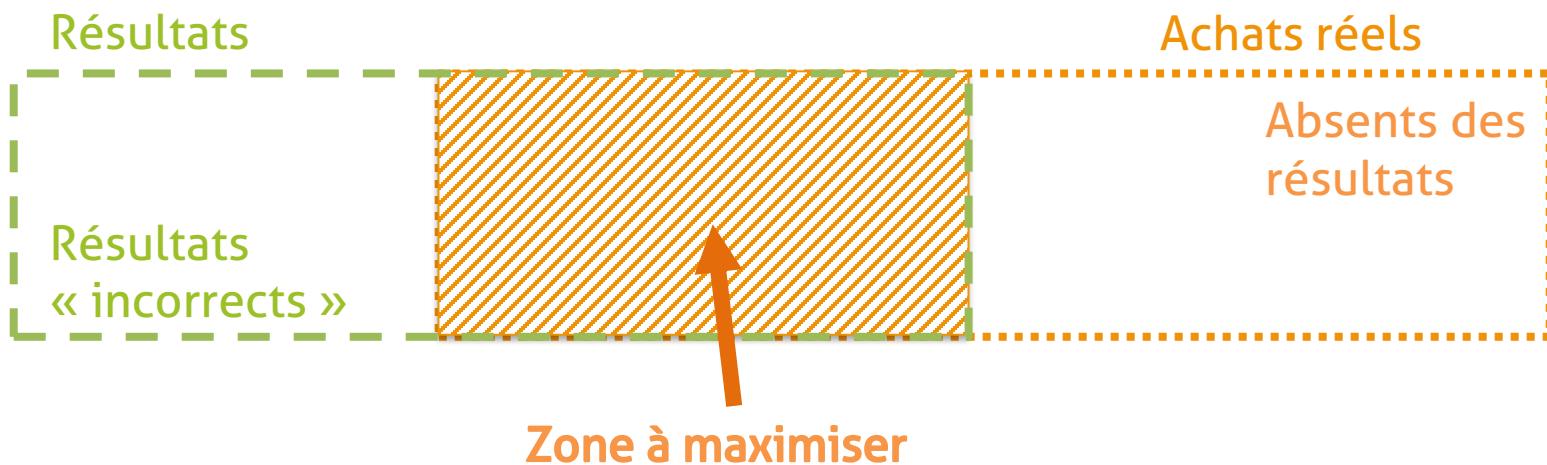
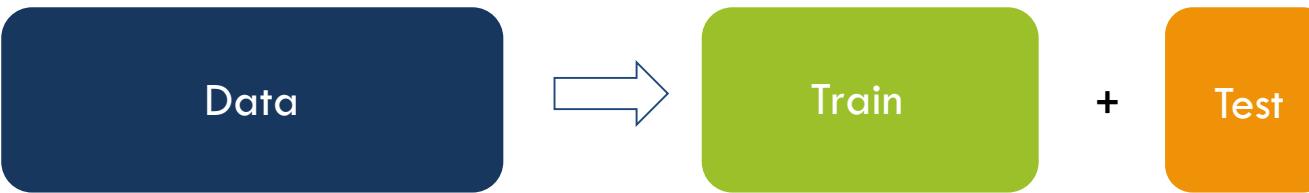
Factorisation de matrice
SVD / NMF

Modèle en graphe



Link prediction
Nearest-neighbour

Validation des algorithmes prédictifs



$$\text{Précision} = \frac{\text{Résultats corrects}}{\text{Résultats retournés}}$$

$$\text{Rappel} = \frac{\text{Résultats corrects}}{\text{Achats réels}}$$

$$F_1 = \frac{2*P*R}{P+R}$$

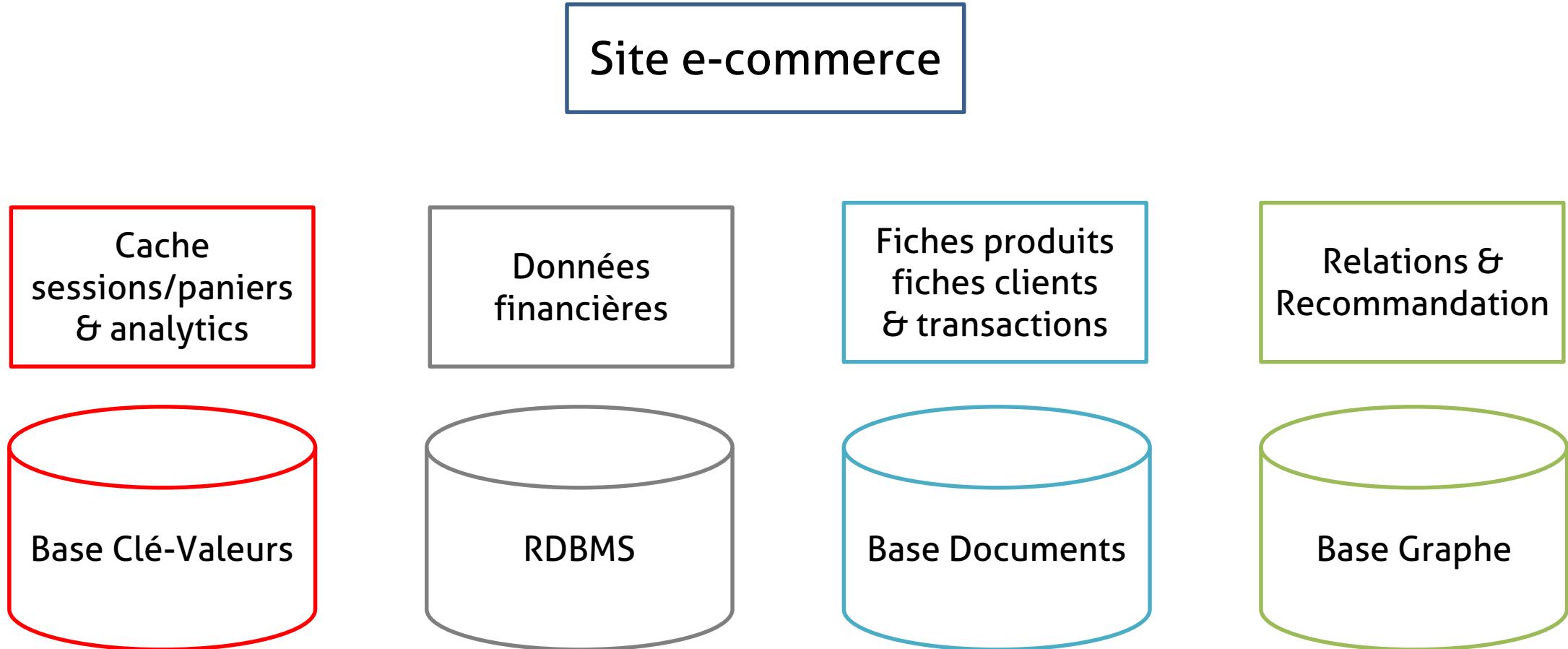
Matrice de confusion

		Prédit	
		Positif	Négatif
Réel	Positif	TP	FN
	Négatif	FP	TN

$$\text{Précision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

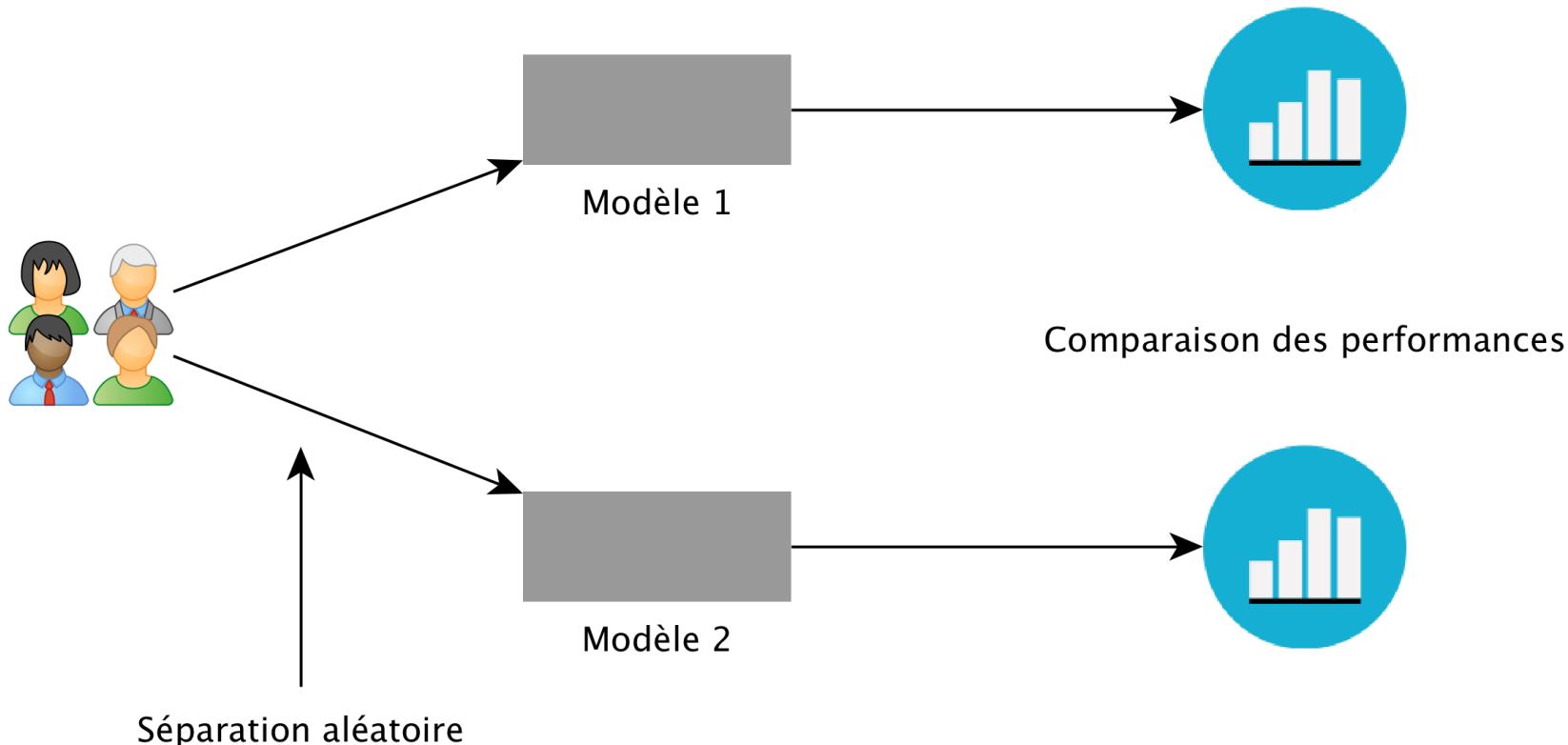
$$\text{Rappel} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

Exemple d'architecture



Validation en conditions réelles

KPI suivi : valeur moyenne du panier

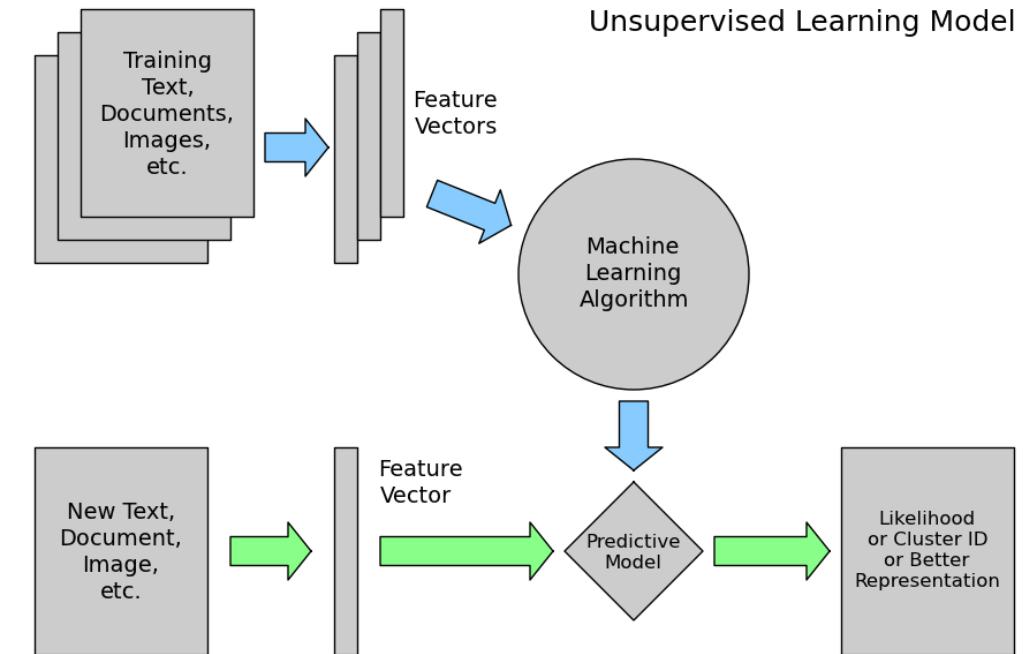
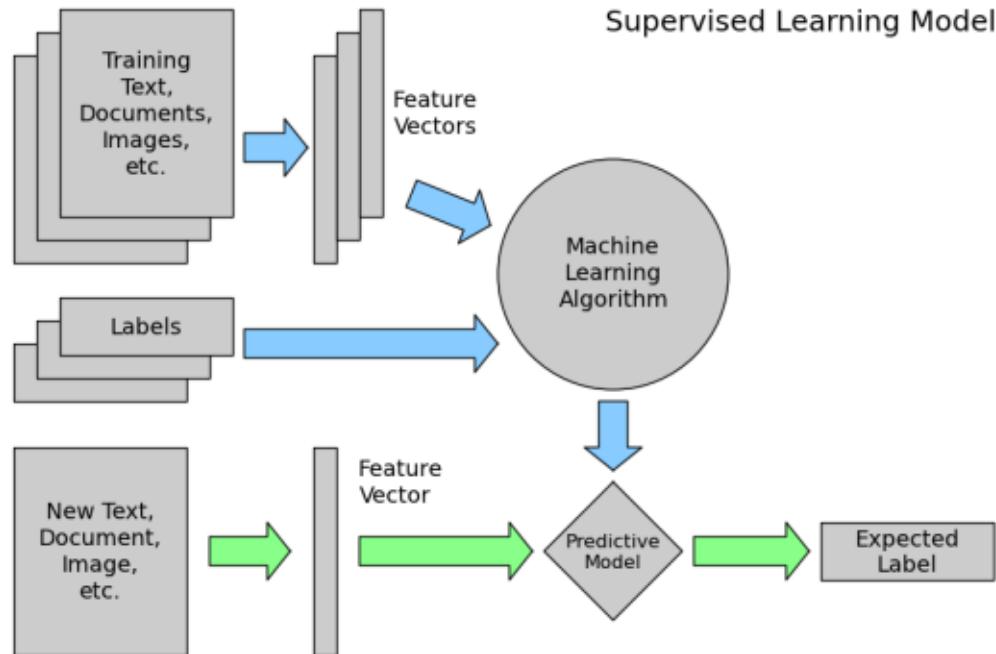


A/B testing :
Comparaison avec
existant ou
baseline à définir.

Machine Learning

Principes et algorithmes

Workflow Machine Learning



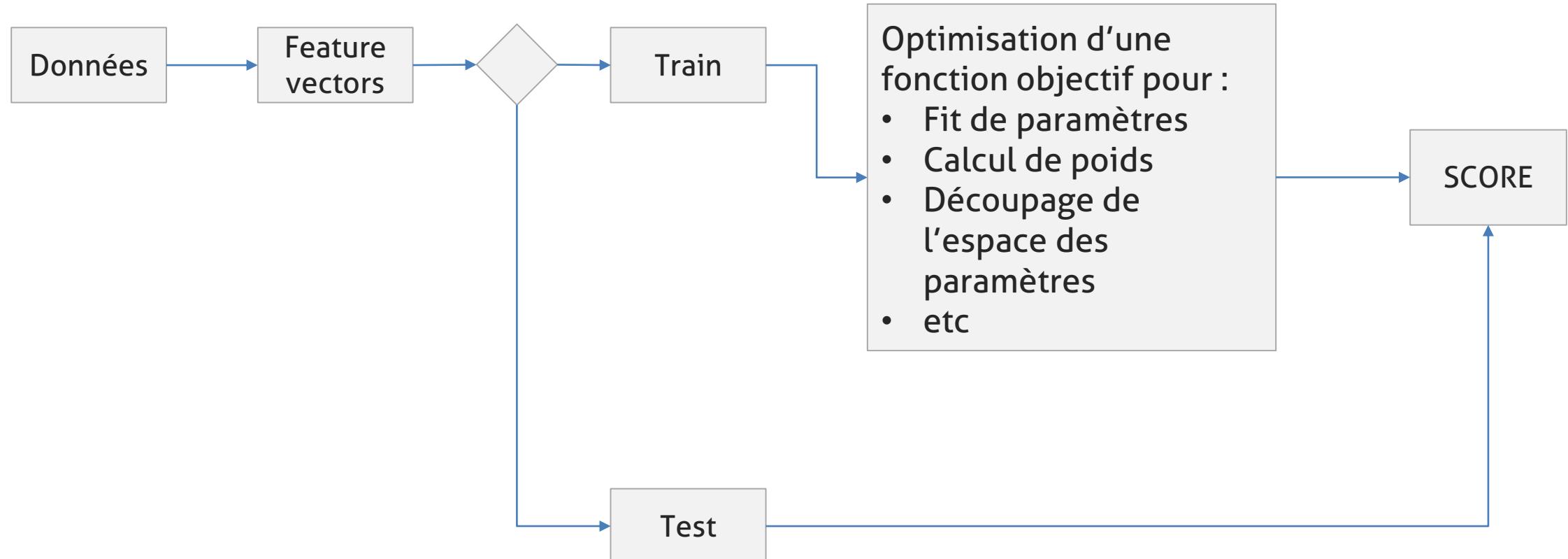
Construction de modèles

Feature engineering

Cross-validation

Apprentissage

Erreur



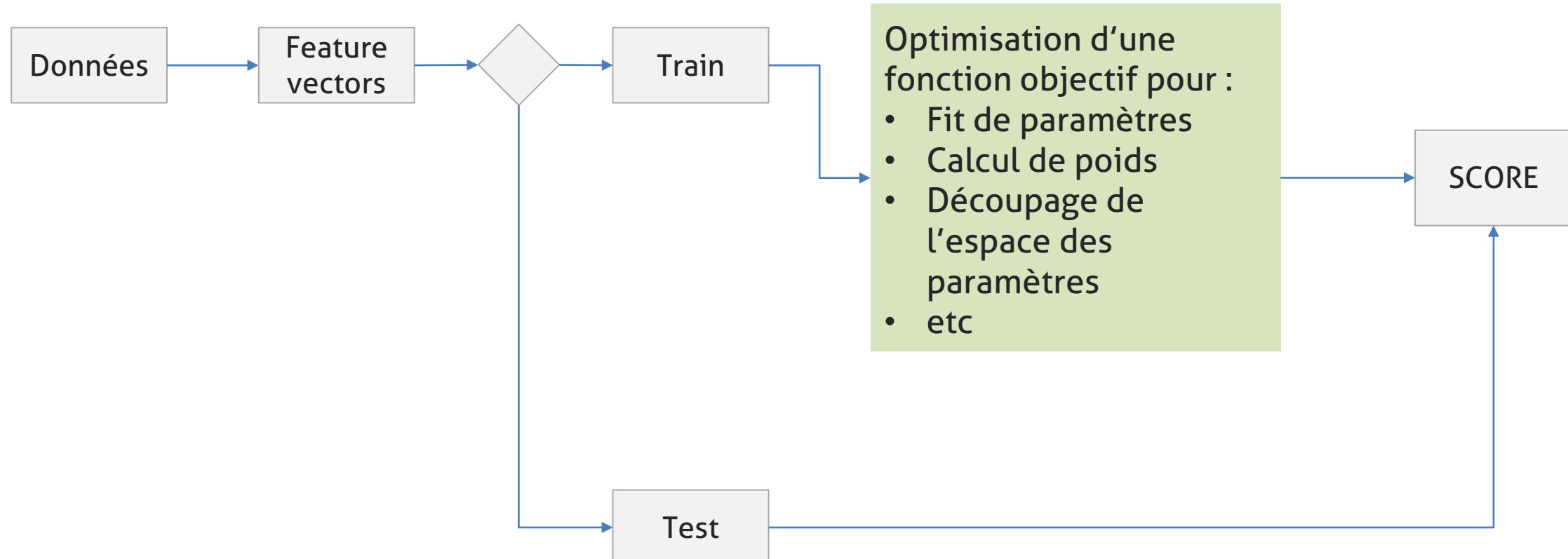
Construction de modèles

Feature engineering

Cross-validation

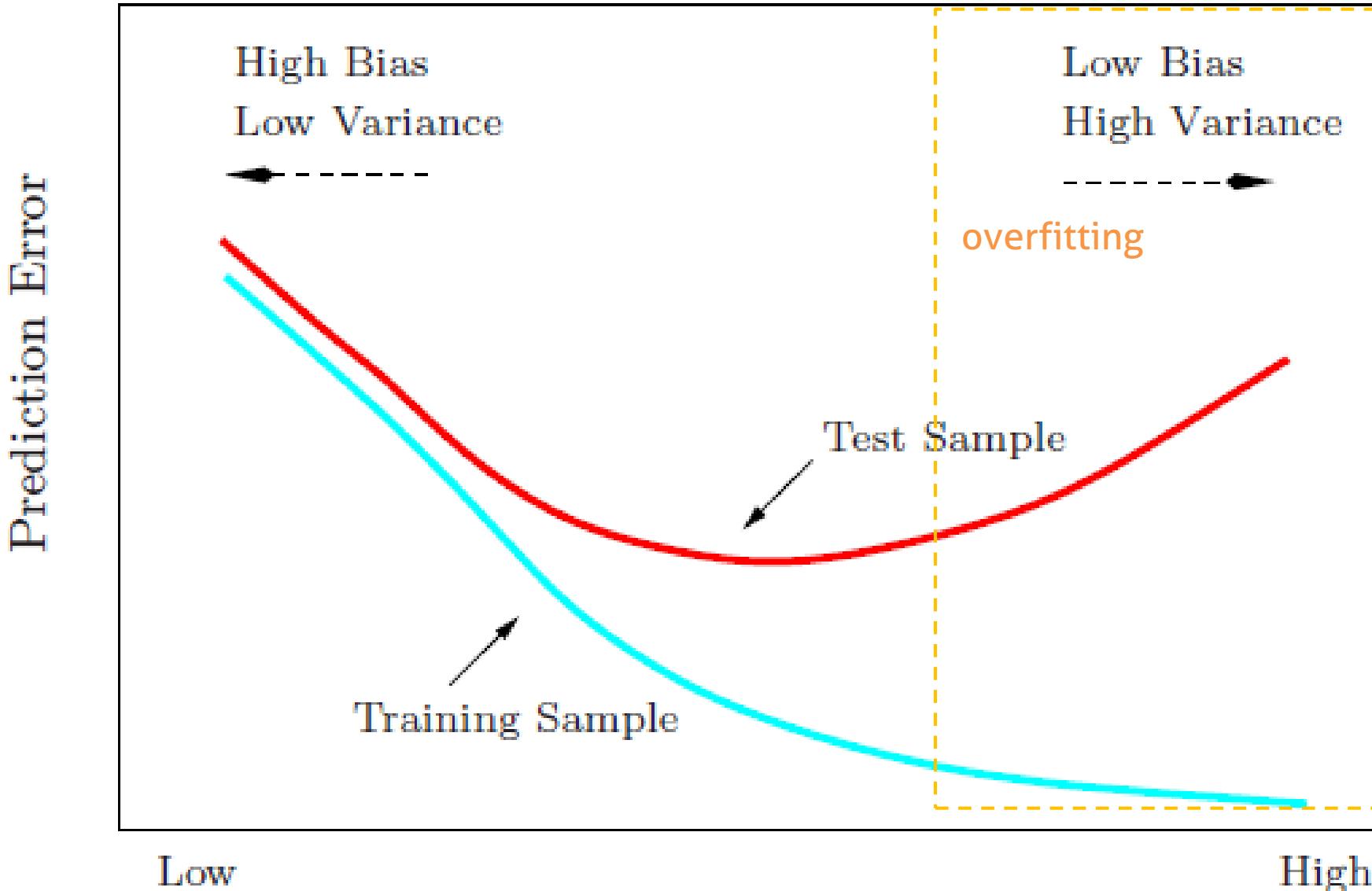
Apprentissage

Erreur

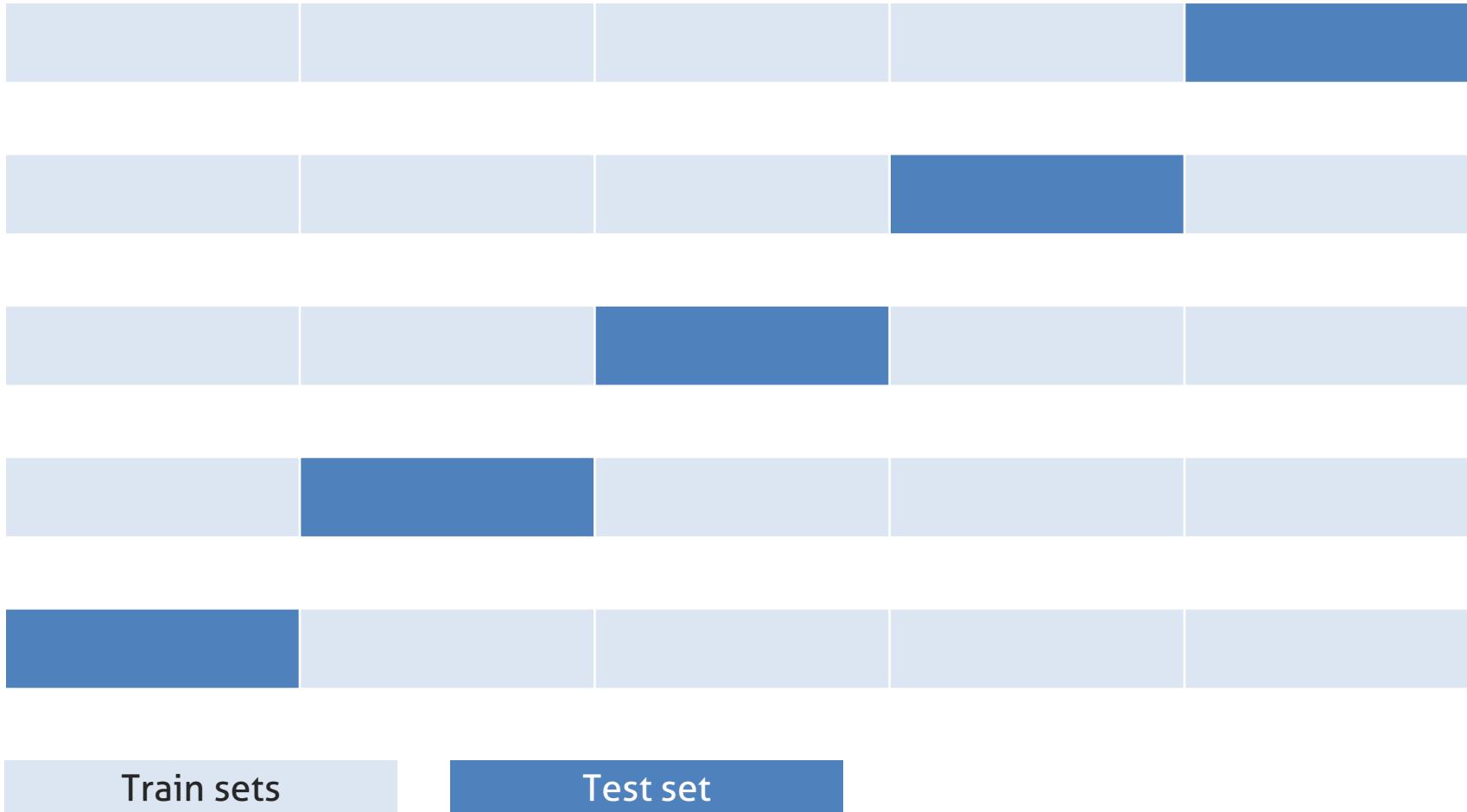


Notebook associé :
1 - Apprentissage et cross-validation.ipynb

Cross-validation

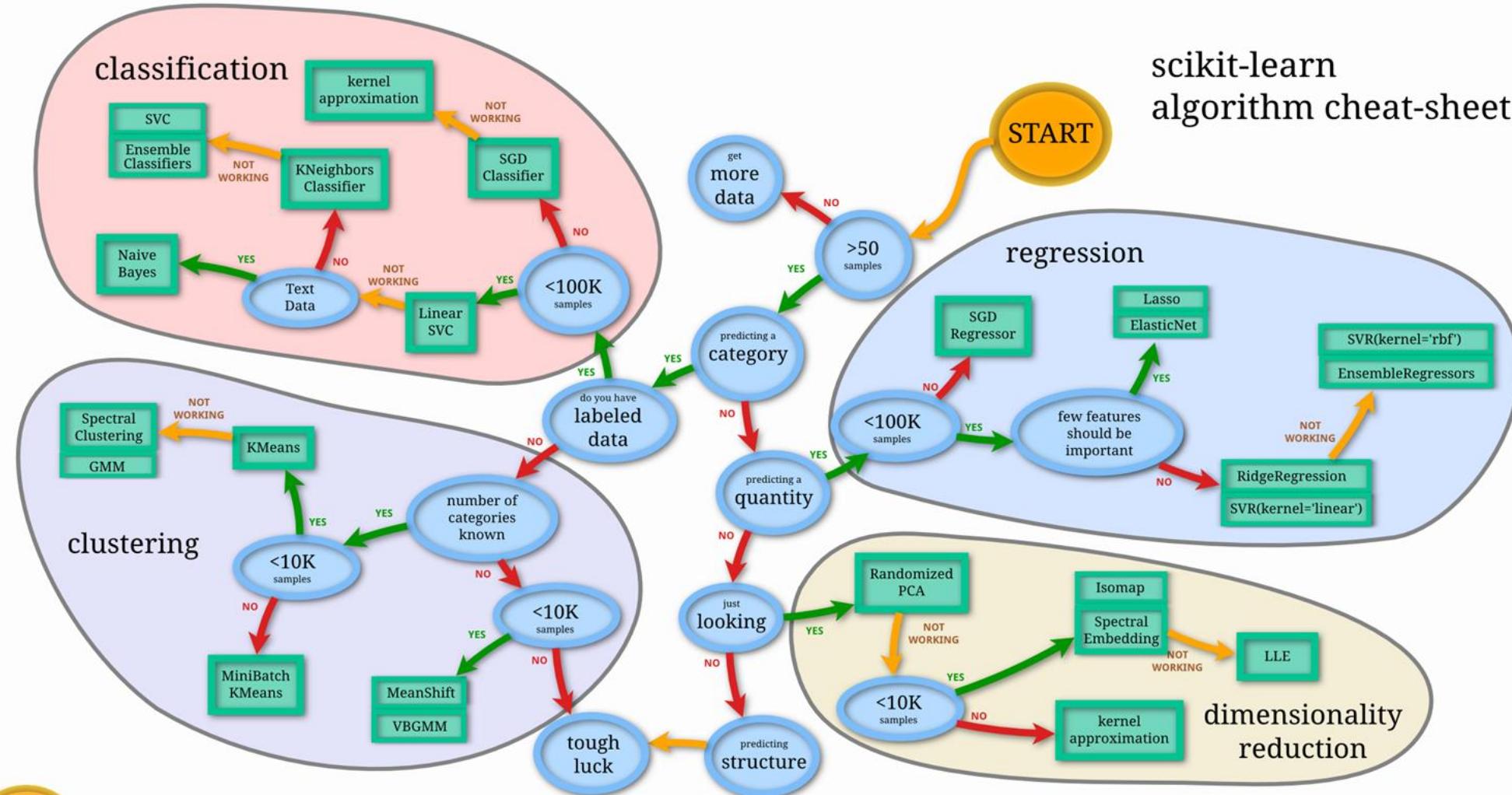


Cross-validation : k-fold



Cas extrême : Leave One Out

Choisir le bon algorithme

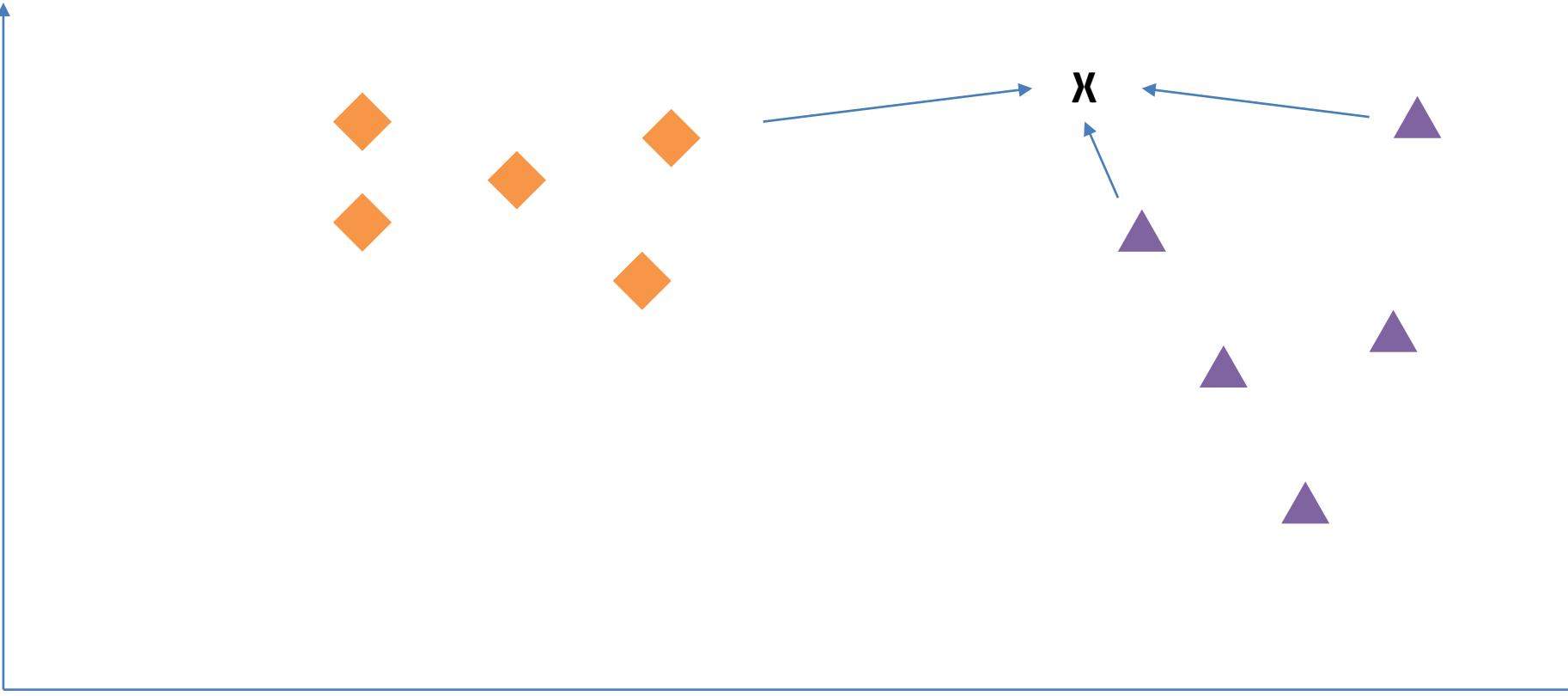


Nearest Neighbours

Feature 1

Vote des plus proches voisins

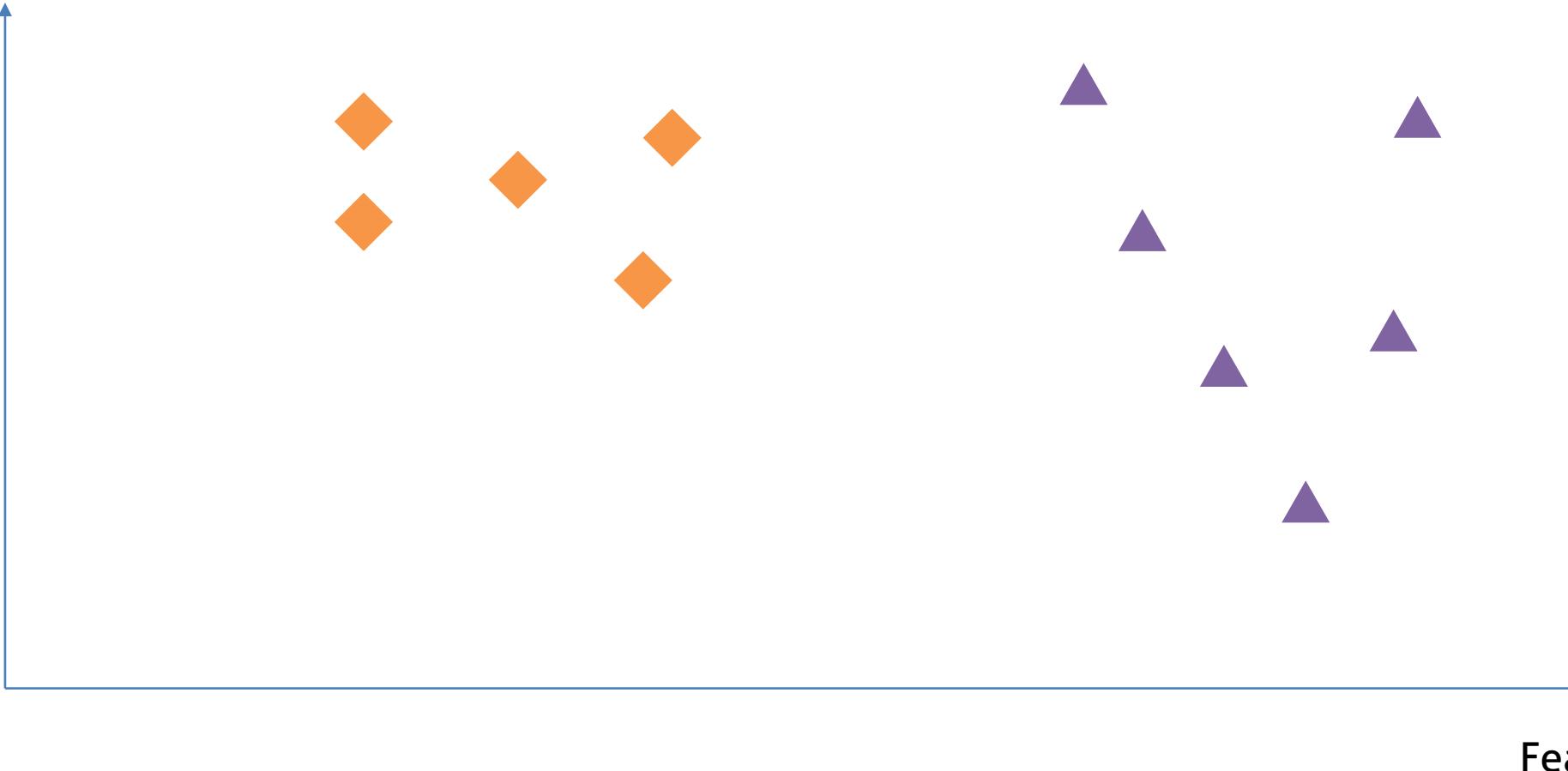
Feature 2



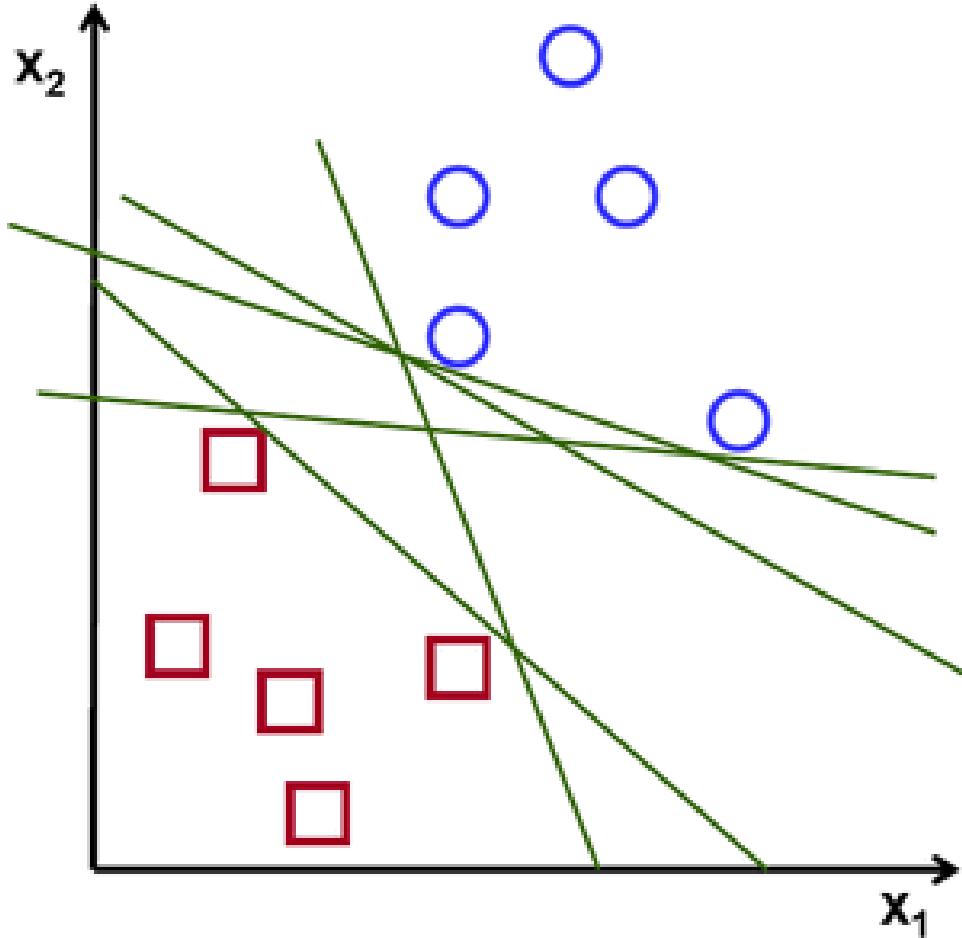
Nearest Neighbours

Feature 1

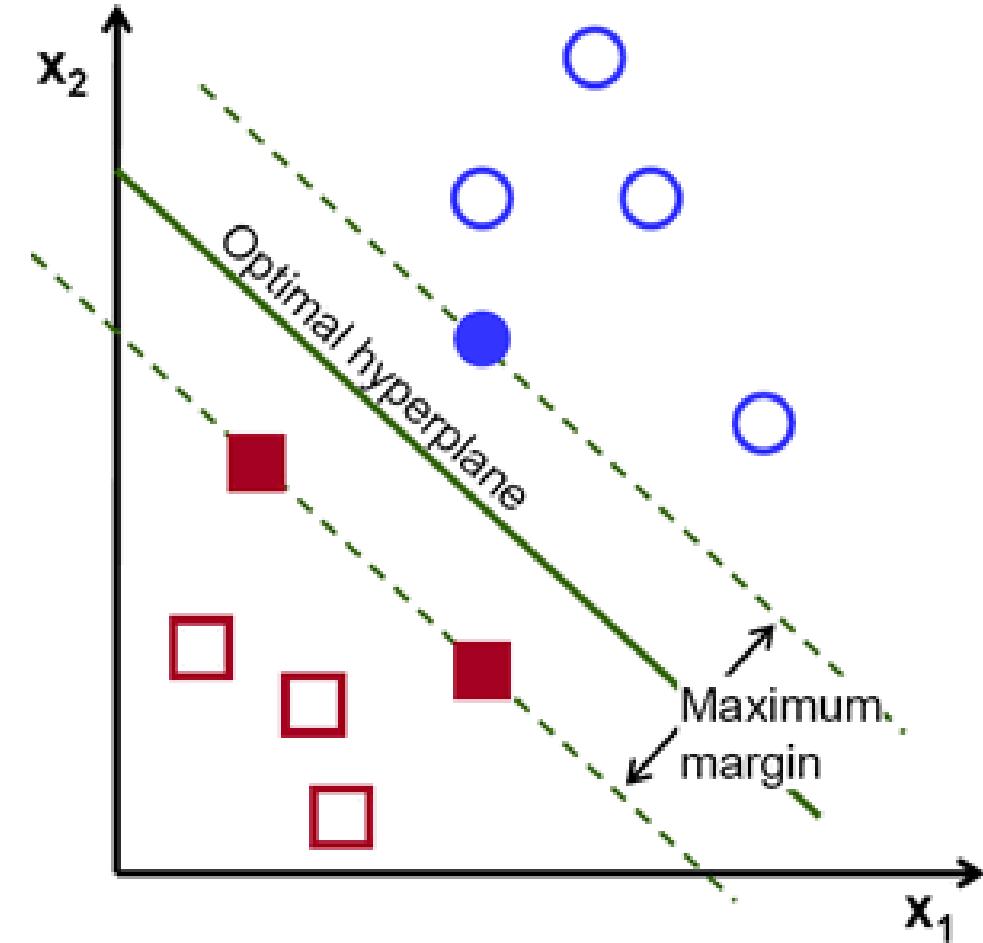
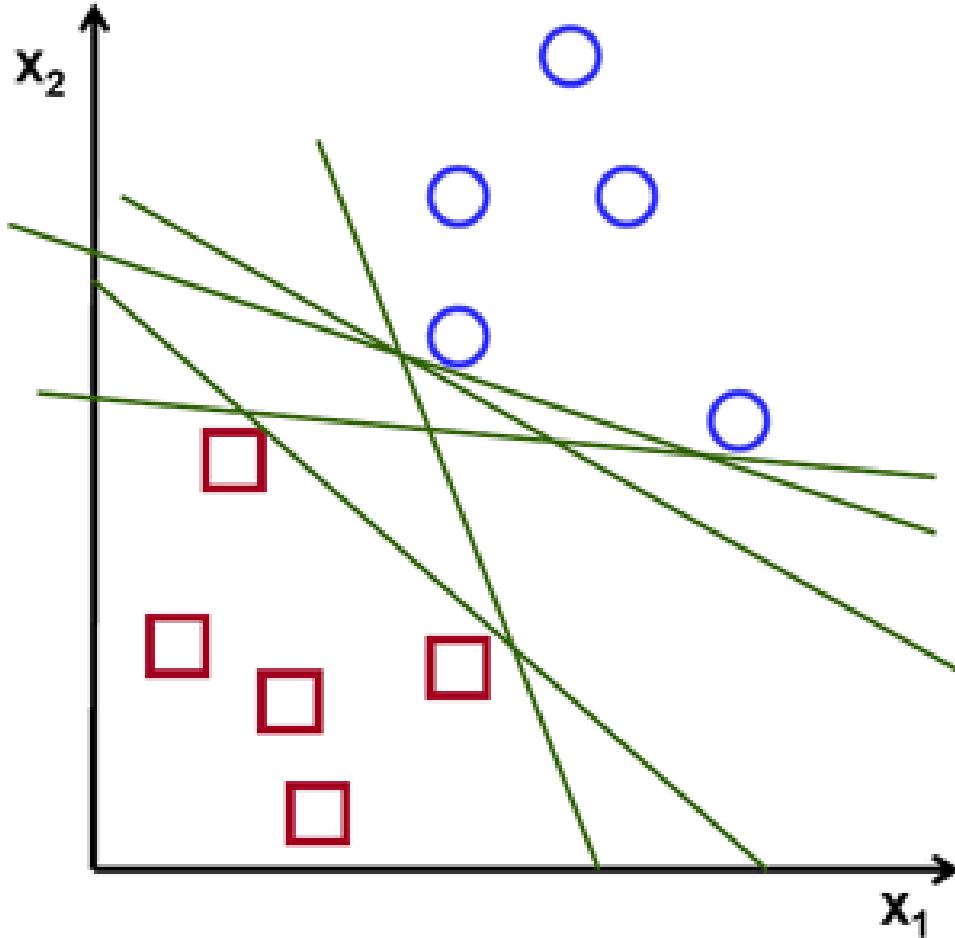
Vote des plus proches voisins



SVM

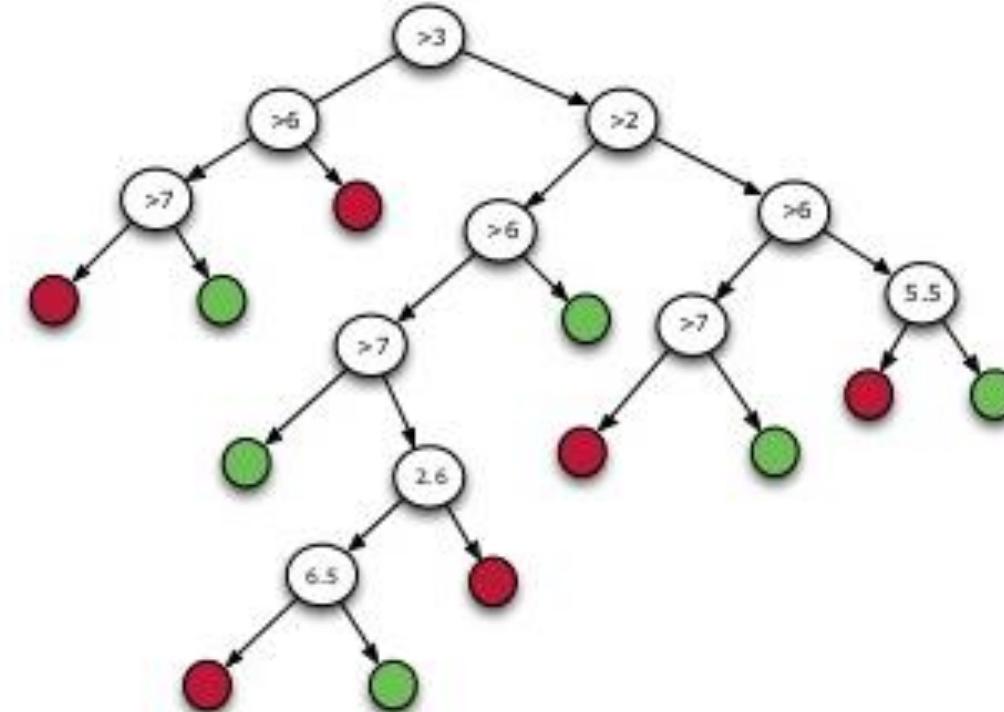
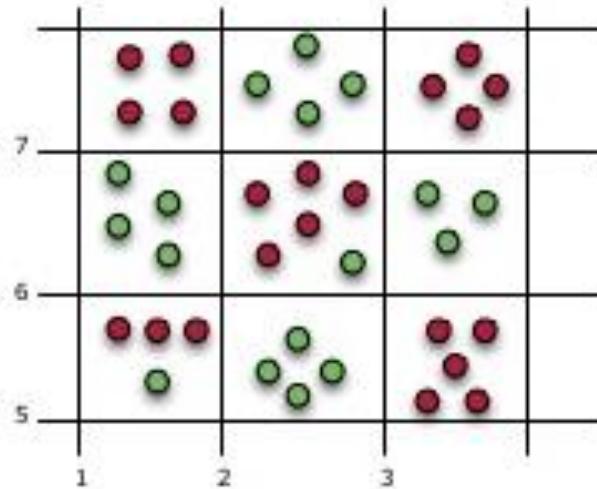


SVM

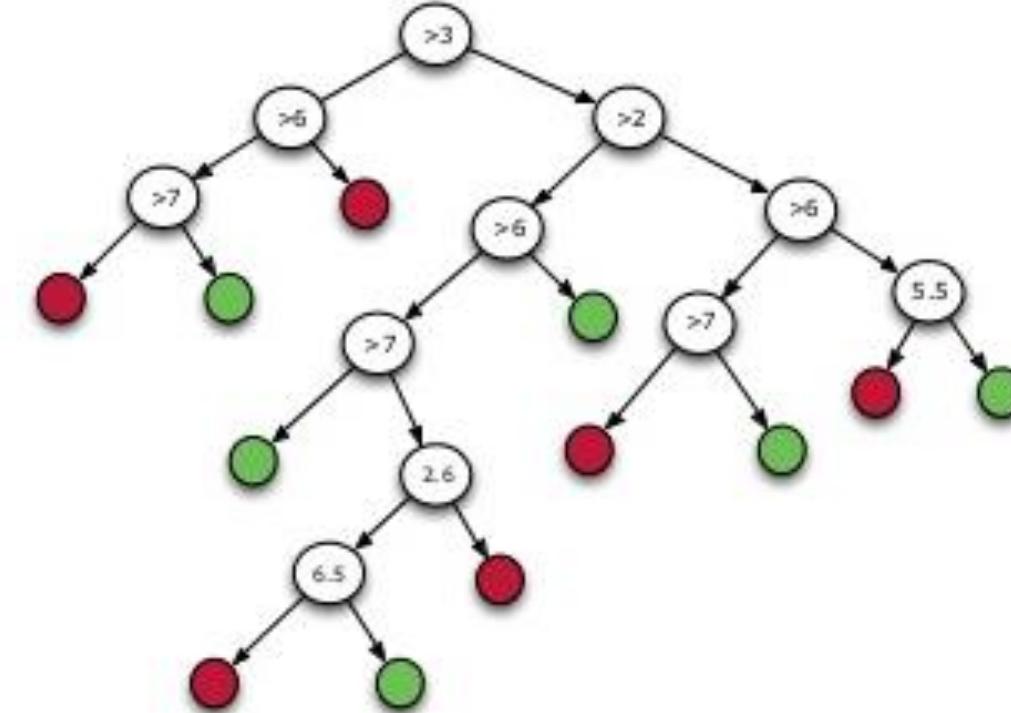
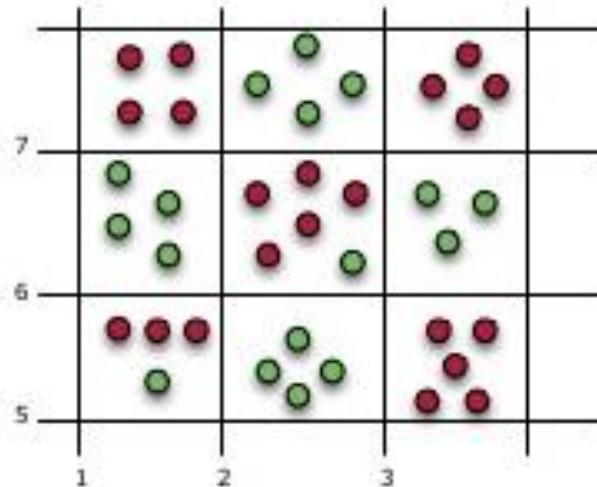


Maximiser la distance entre les points et l'hyperplan : marge maximale

Random Forest

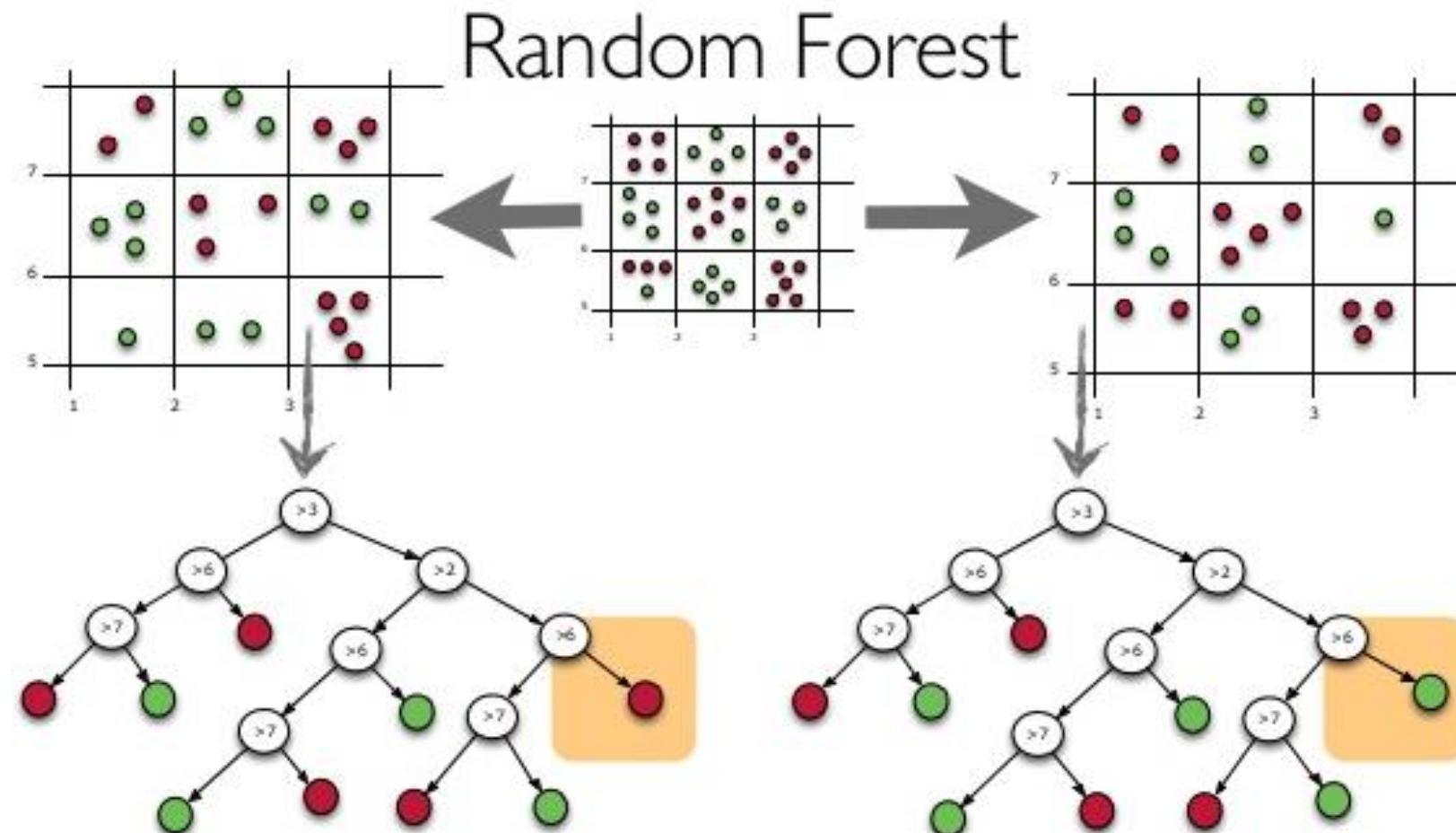


Random Forest

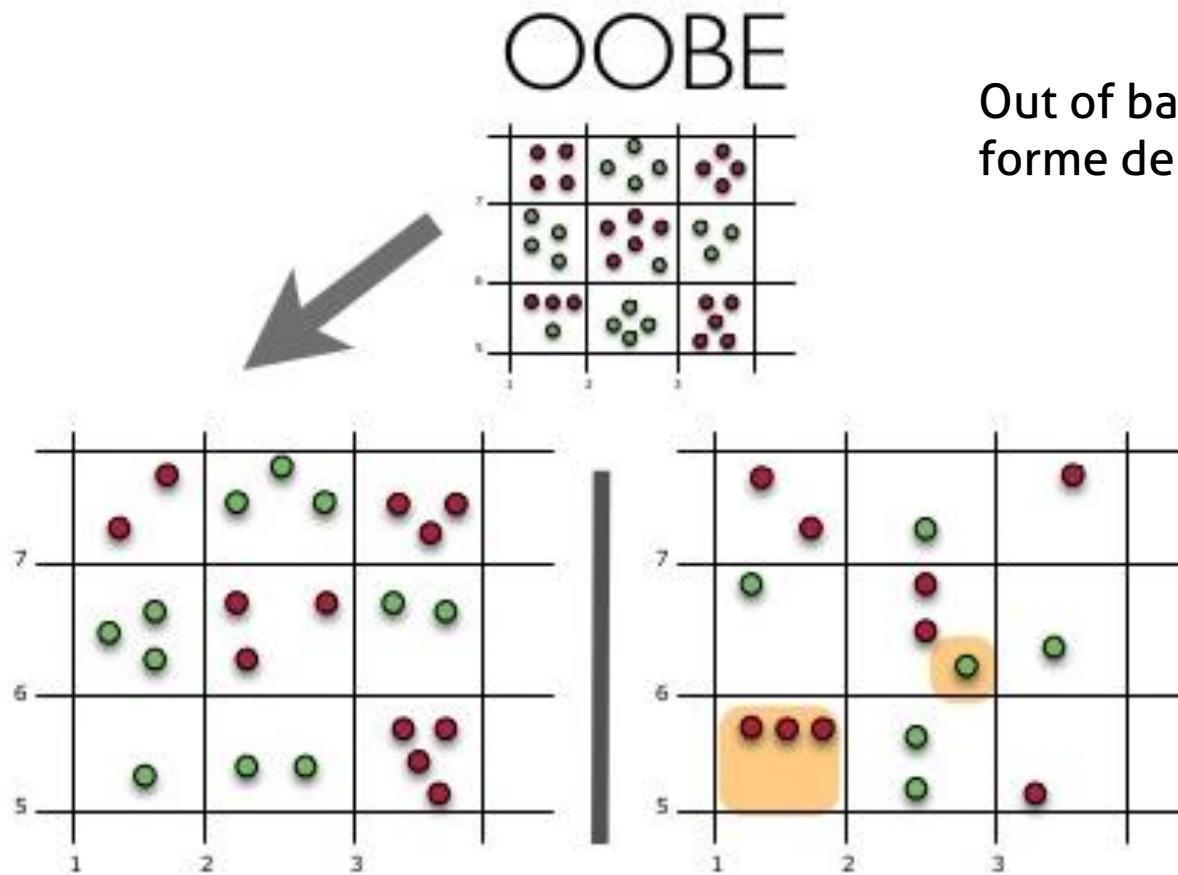


Un arbre de décision a tendance à overfitter !

Random Forest

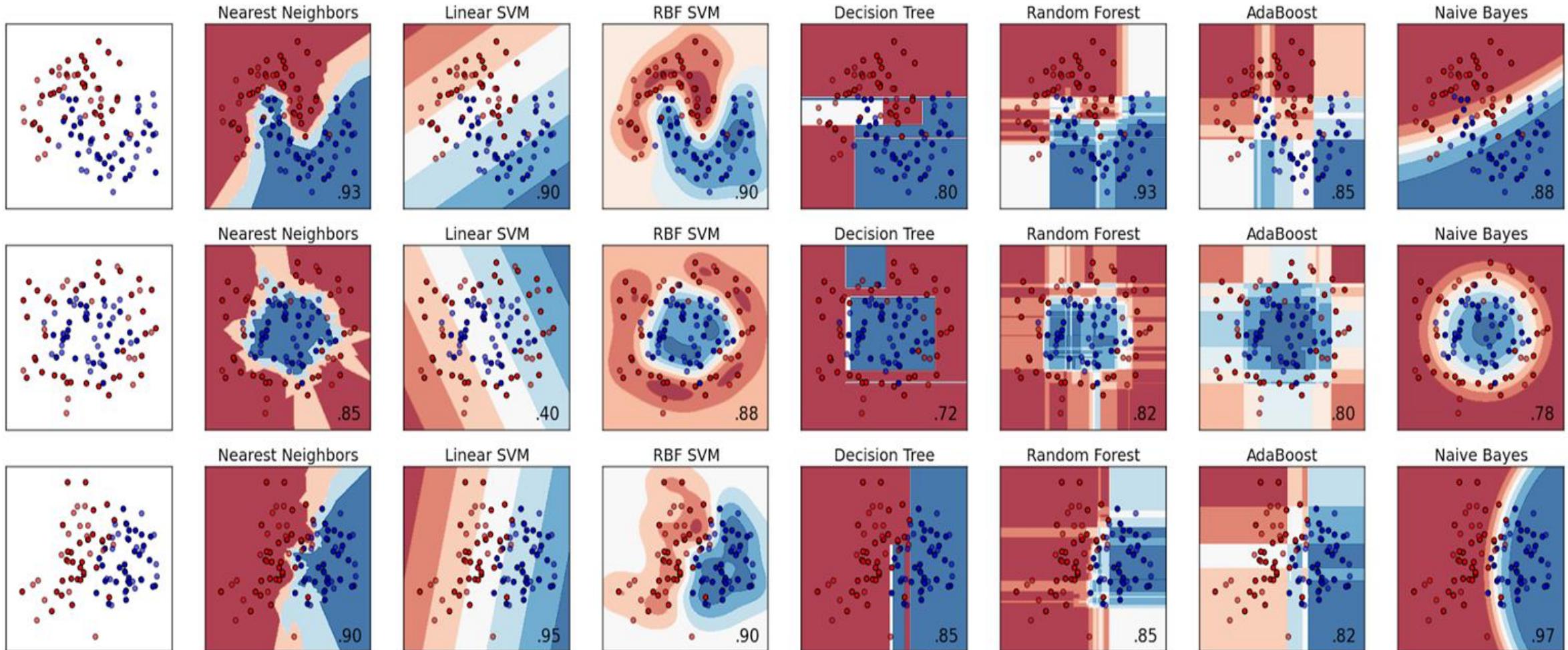


Random Forest



Out of bag error :
forme de cross-validation interne

Classification



ML supervisé avec scikit-learn

```
data = pandas.read_csv()  
# some feature engineering
```

Chargement et
traitement des données

```
train_data, test_data = train_test_split(data)
```

Cross-validation

```
model.fit(train_data)  
predicted = model.predict(test_data)
```

Fit du modèle
et prédictions

```
model.score(predicted, test_data)
```

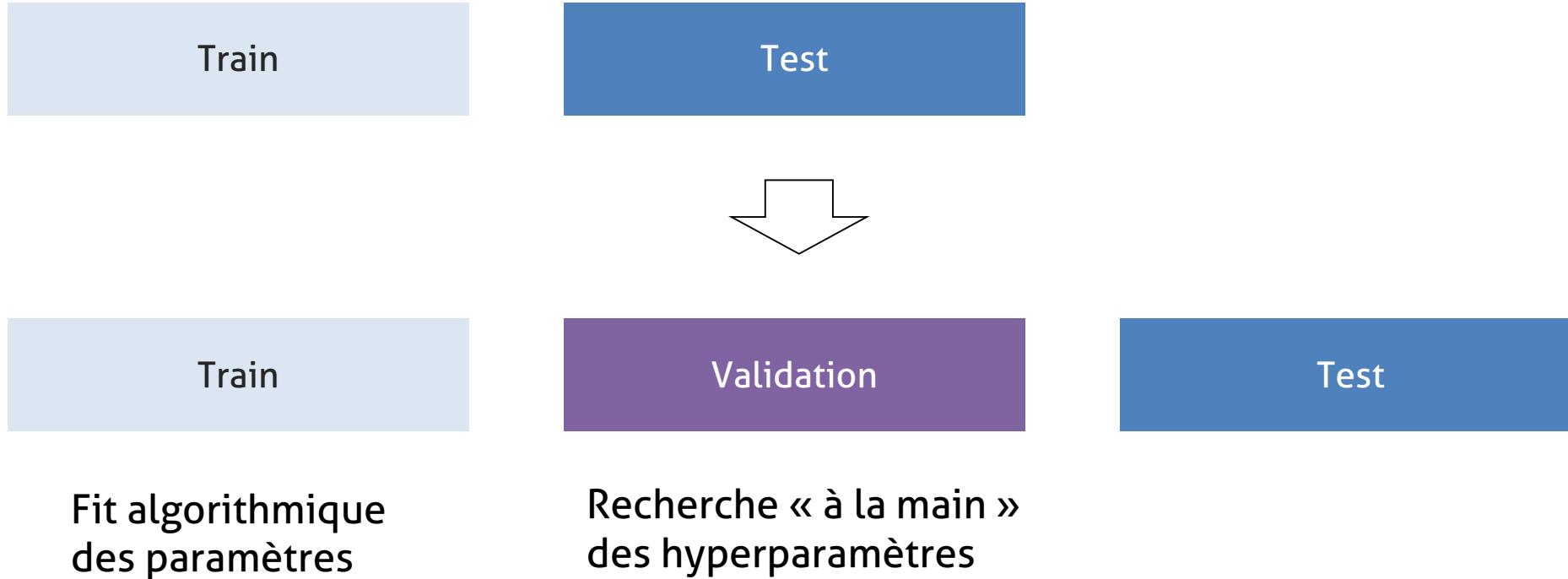
Comparaison

Notebook associé :
2 – Demo Titanic.ipynb

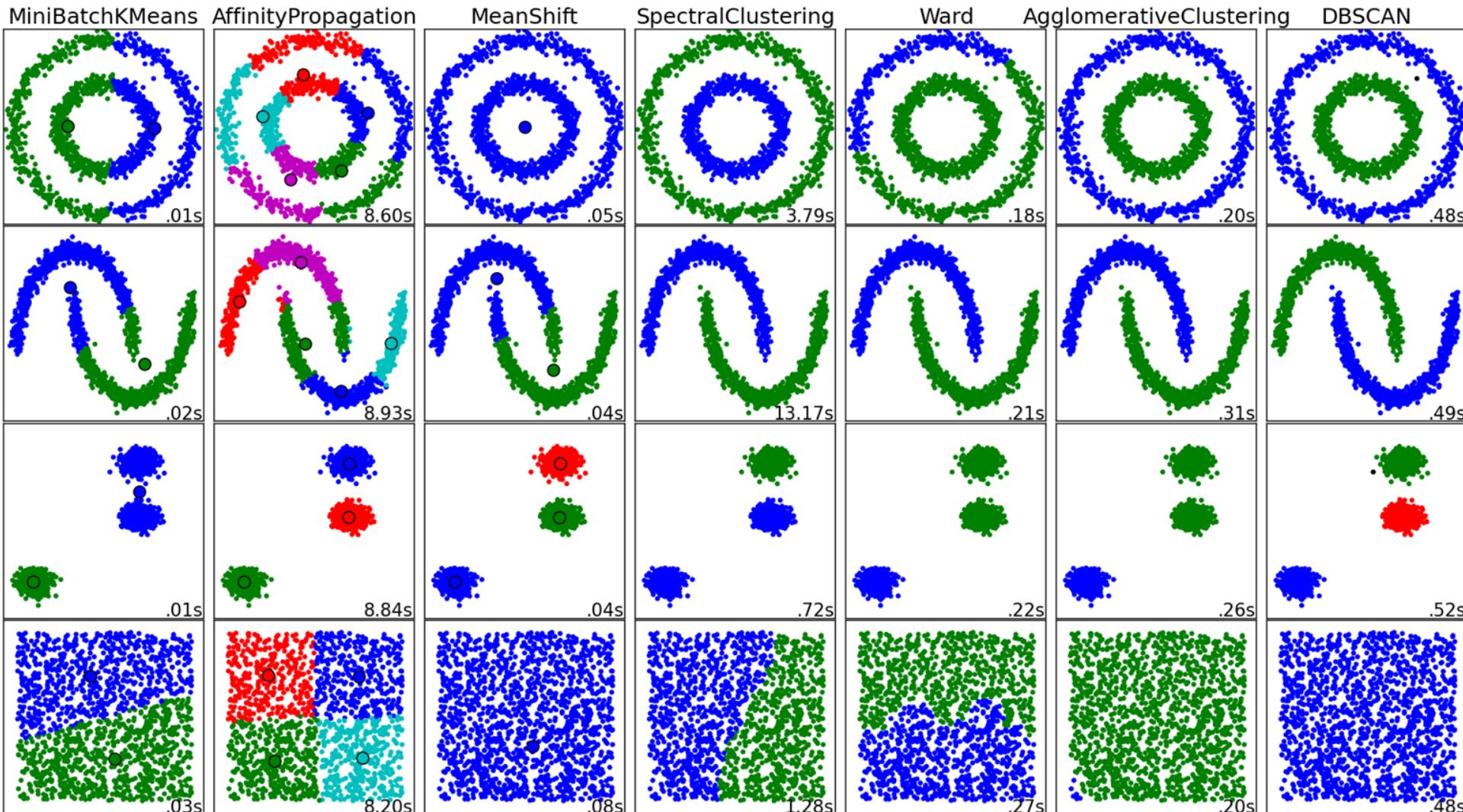
Hyperparamètres et cross-validation

Hyperparamètres :

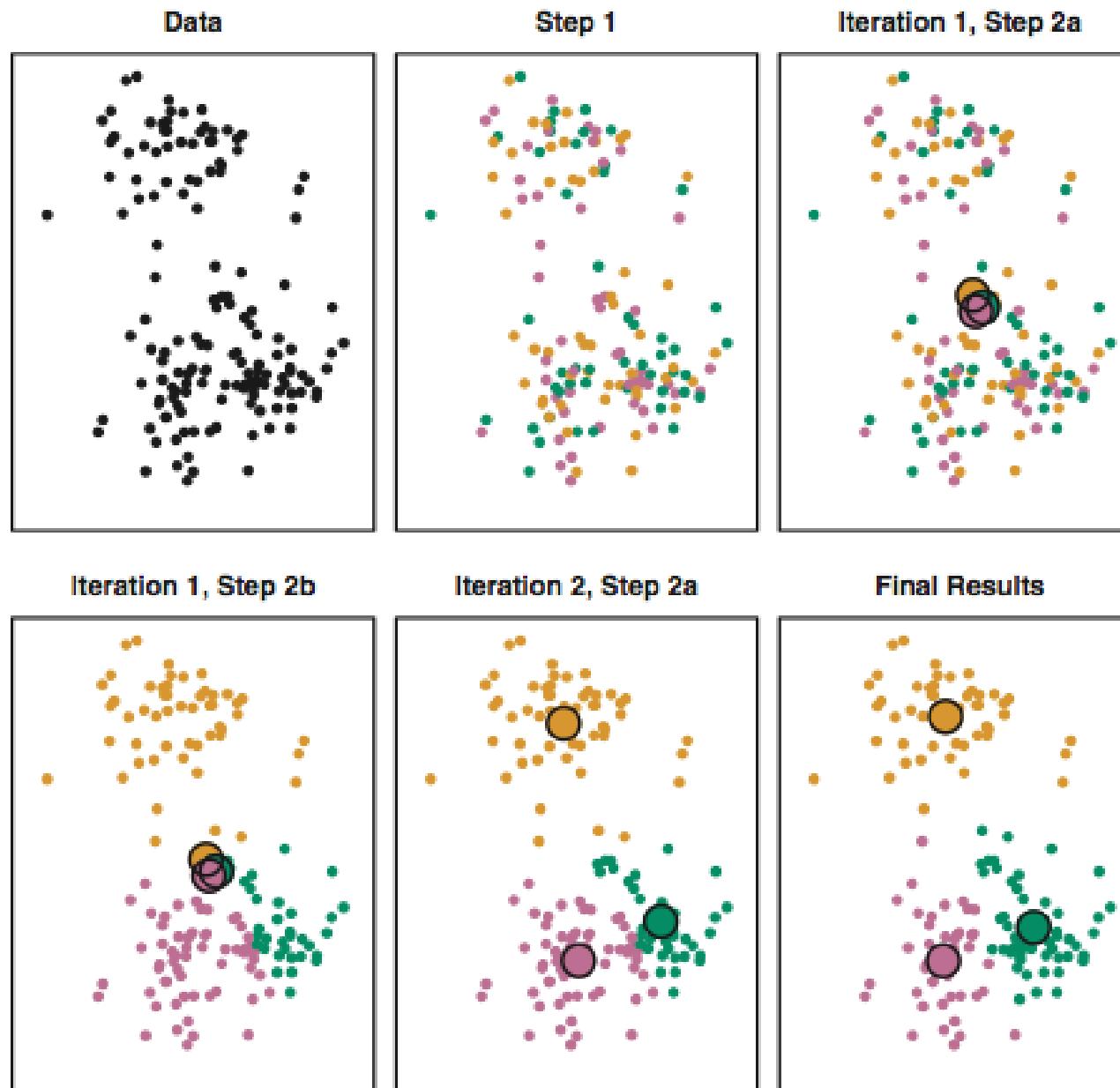
Un modèle est caractérisé par des paramètres qui sont fixés à la main



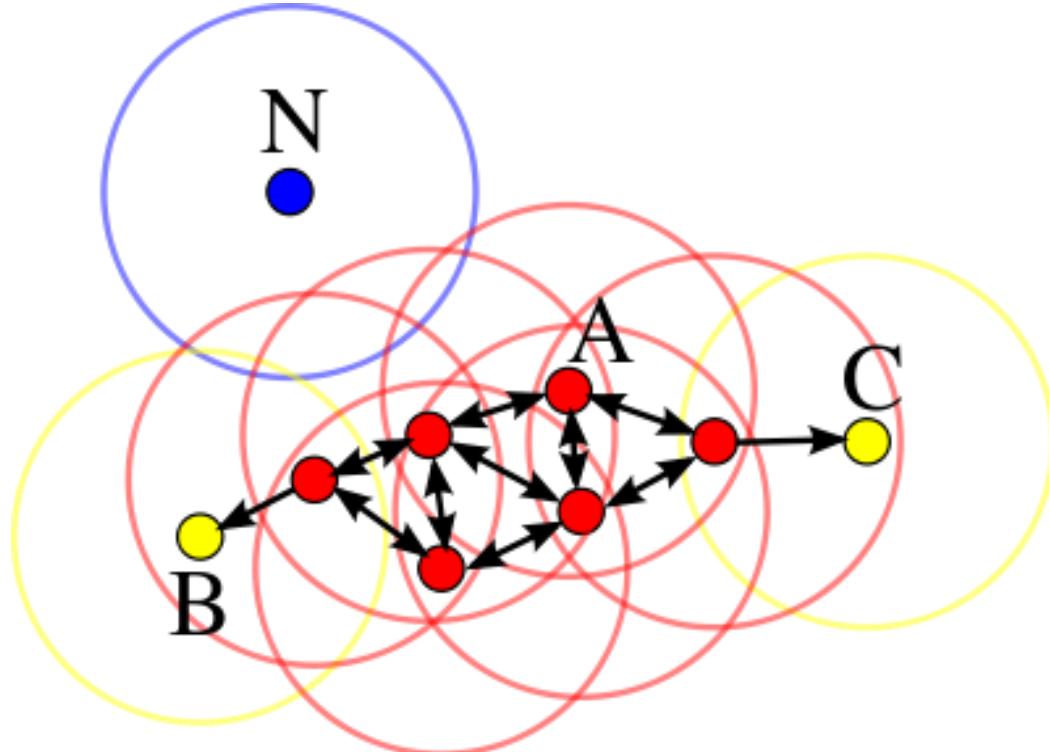
Clustering



K-means



DBSCAN



- Densité suffisante au voisinage d'un point
- On remplit le cluster de proche en proche

Intérêts :

- On ne fixe pas le nombre de clusters
- Des points peuvent rester isolés
- Géométrie quelconque des cluster

Par contre, K-mean reste plus simple et rapide à mettre en œuvre.

Et le Deep Learning ?

ImageNet Challenge

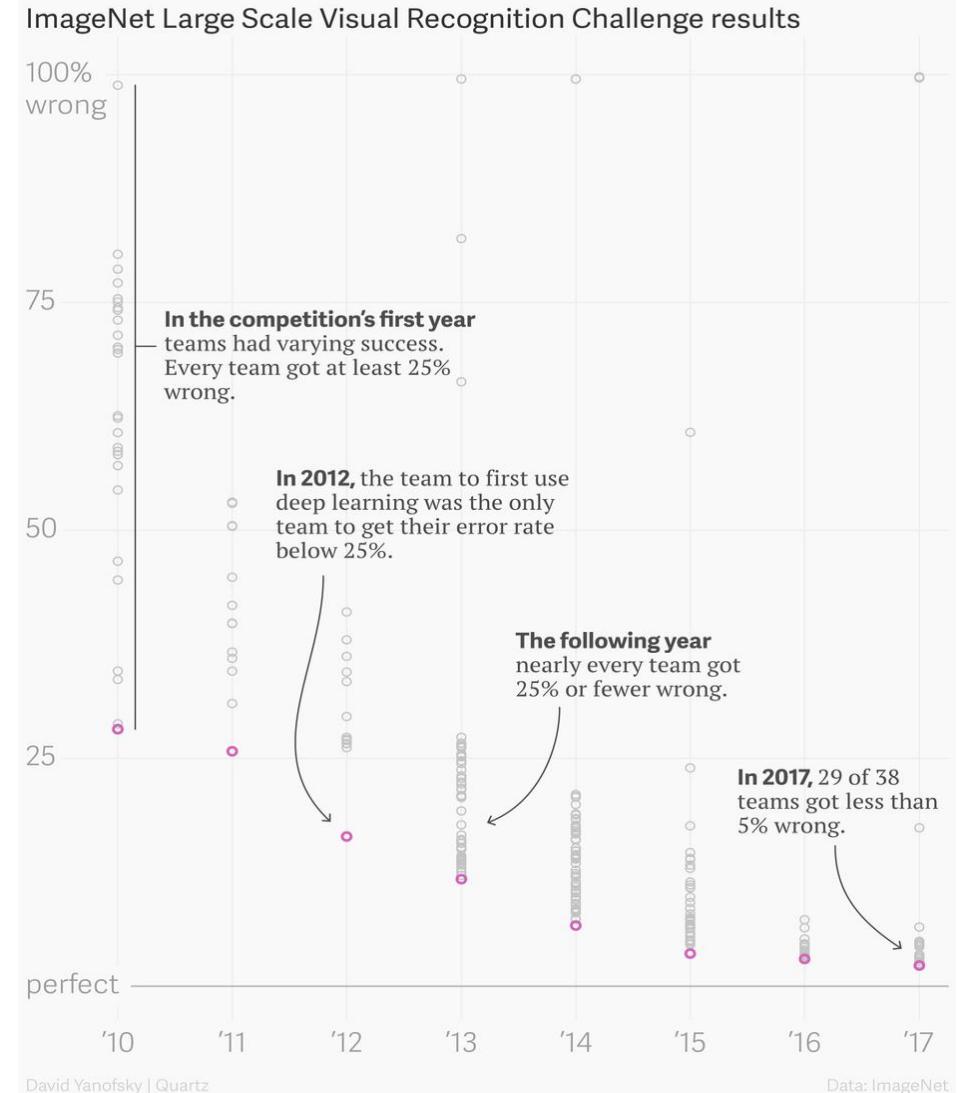
IMAGENET

- 1,000 object classes (categories).
- Images:
 - 1.2 M train
 - 100k test.

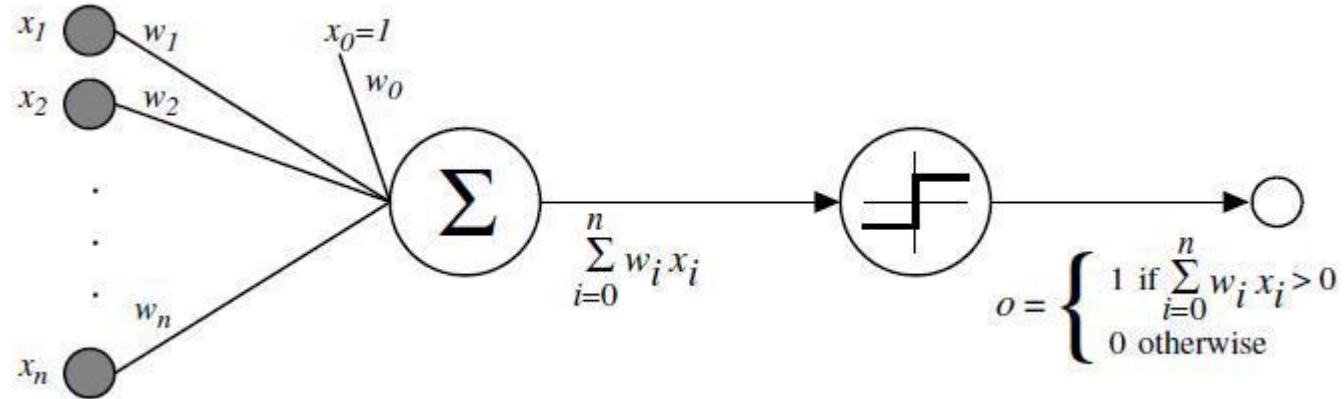


Geoffrey Hinton :

1. Our labeled datasets were thousands of times too small.
2. Our computers were millions of times too slow.
3. We initialized the weights in a stupid way.
4. We used the wrong type of non-linearity.

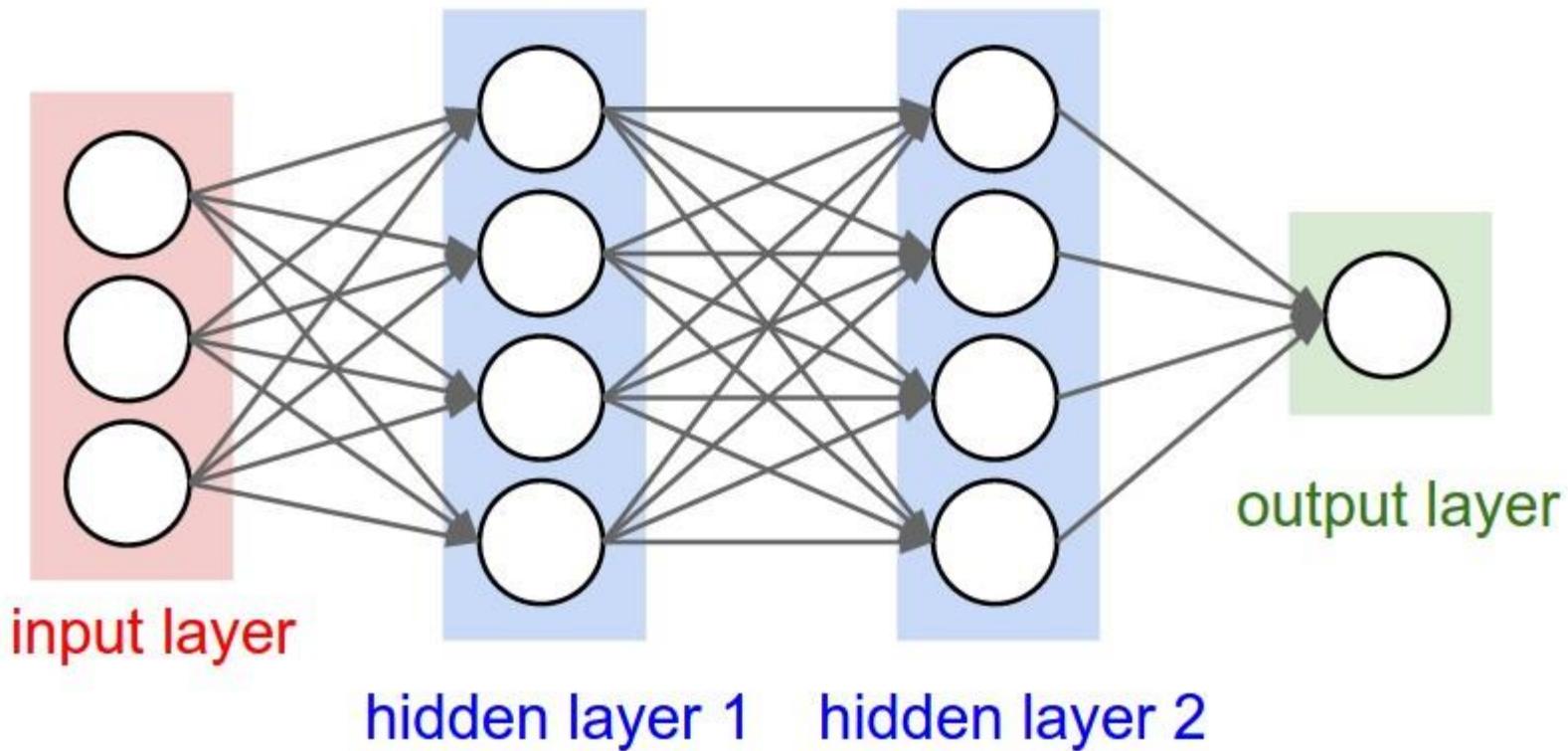


Réseaux de neurones : le perceptron



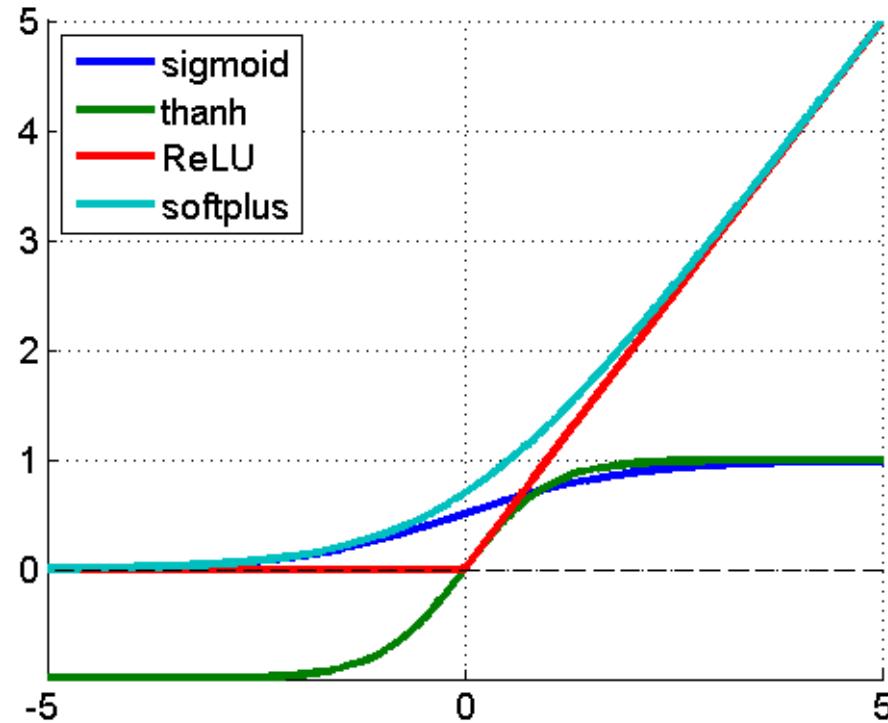
Limité à des frontières linéaires

Multi-layer perceptron



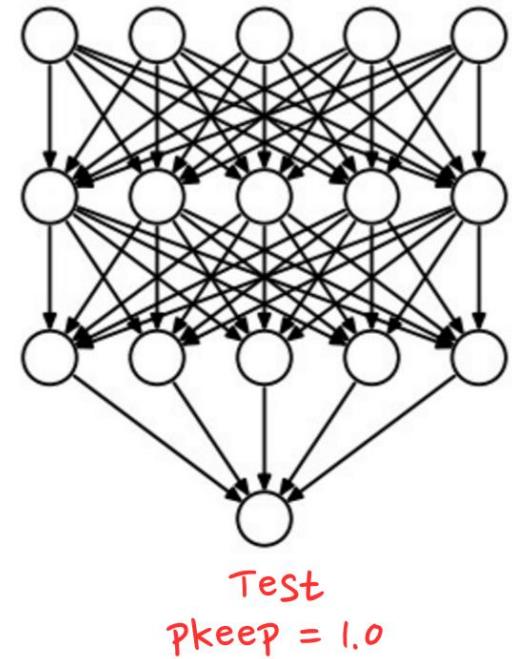
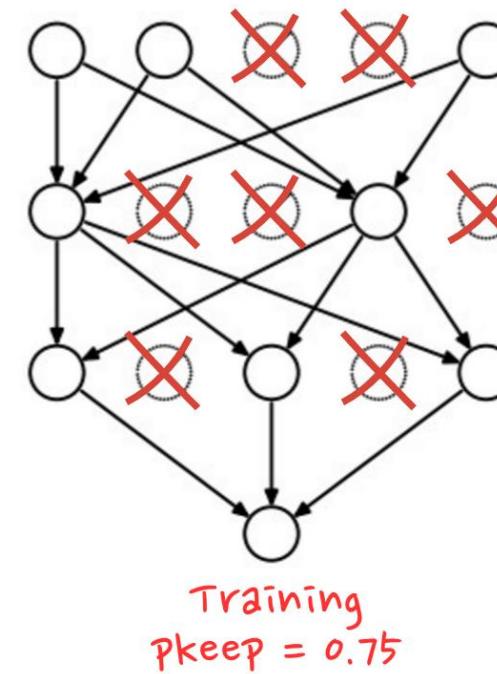
Capable d'apprendre des frontières non-linéaires
Mais des limitations (vanishing gradient, impact de l'initialisation, etc)

Activation et dropout



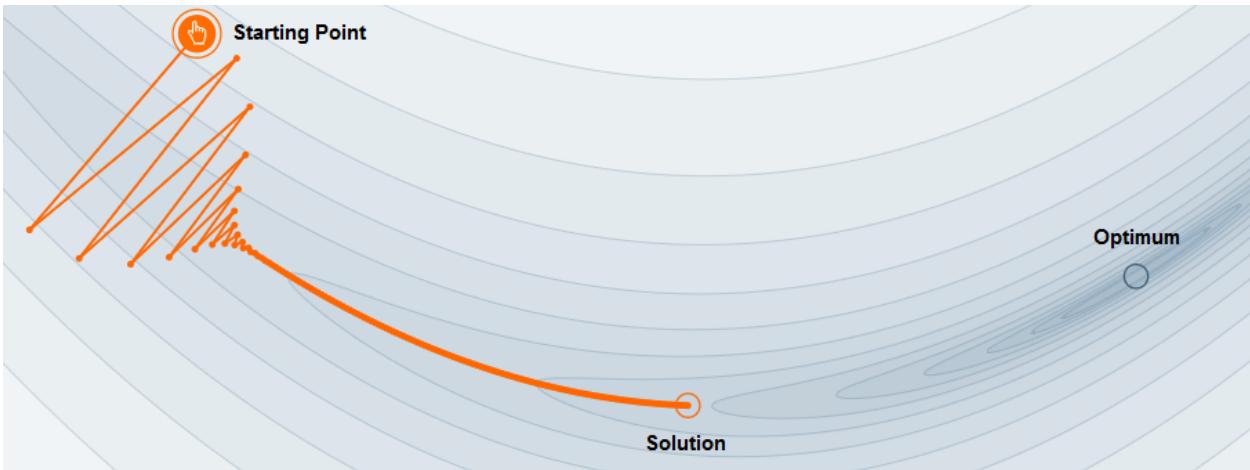
Différentes fonctions d'activations

ReLU permet un meilleur apprentissage dans des réseaux profonds



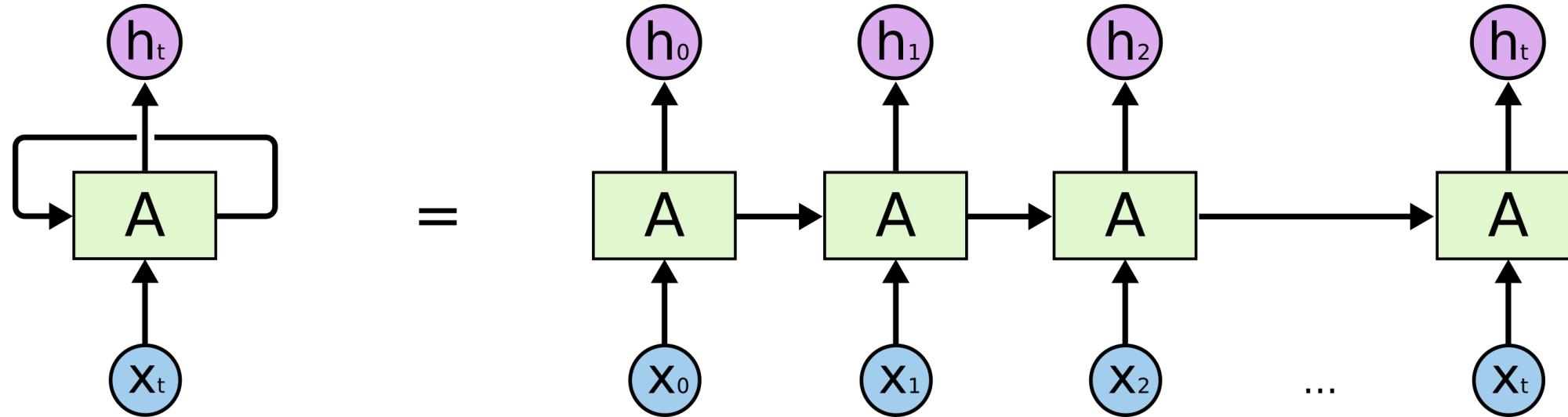
Dropout : on ignore des nœuds aléatoirement pour éviter de renforcer certains patterns (lutte contre l'overfitting)

Descente de gradient avec moment



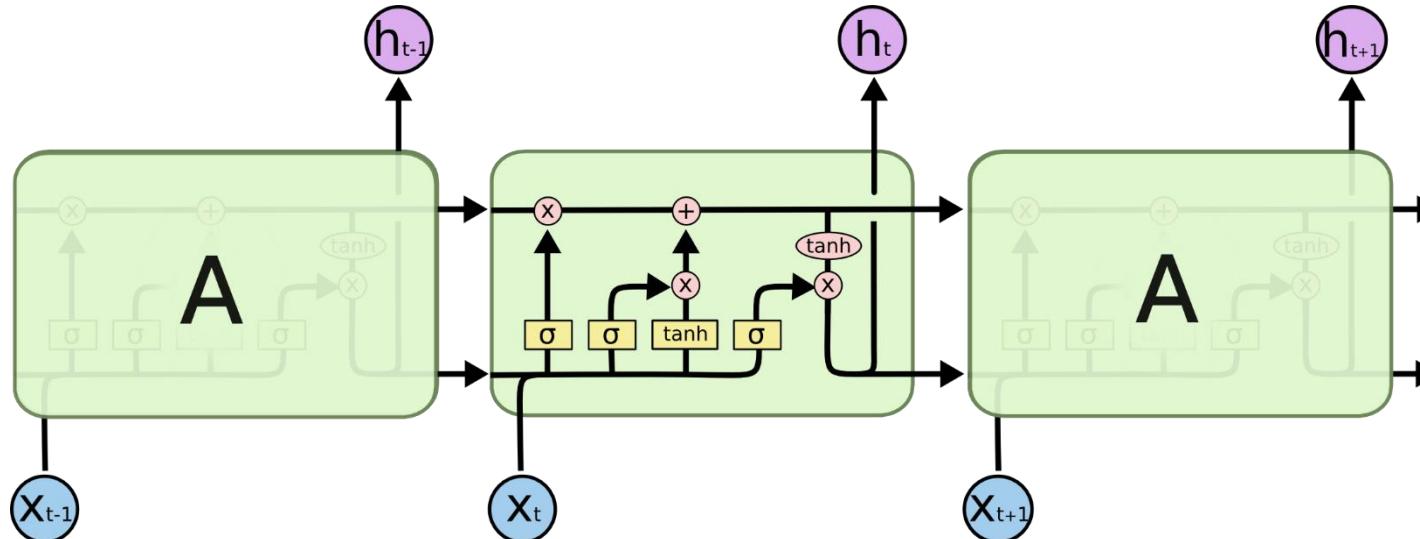
<https://distill.pub/2017/momentum/>

Sequence to sequence learning : Réseaux récurrents



"the cat sat on the mat" -> [Seq2Seq model] -> "le chat était assis sur le tapis"

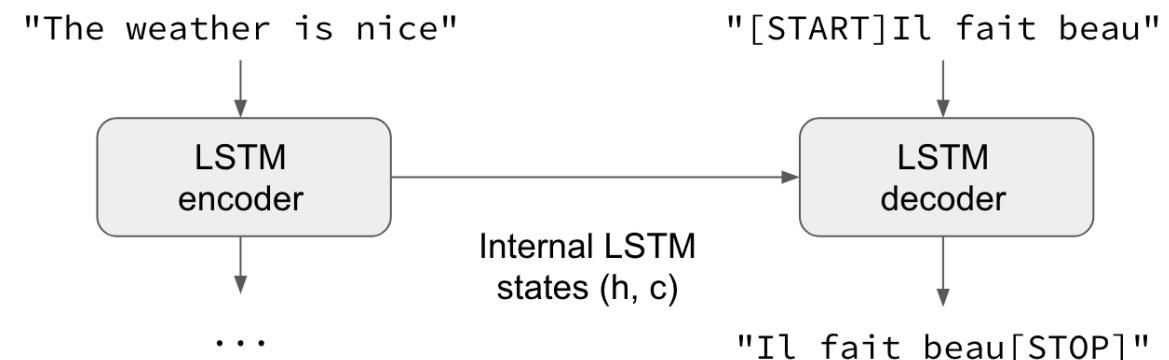
Long Short Term Memory



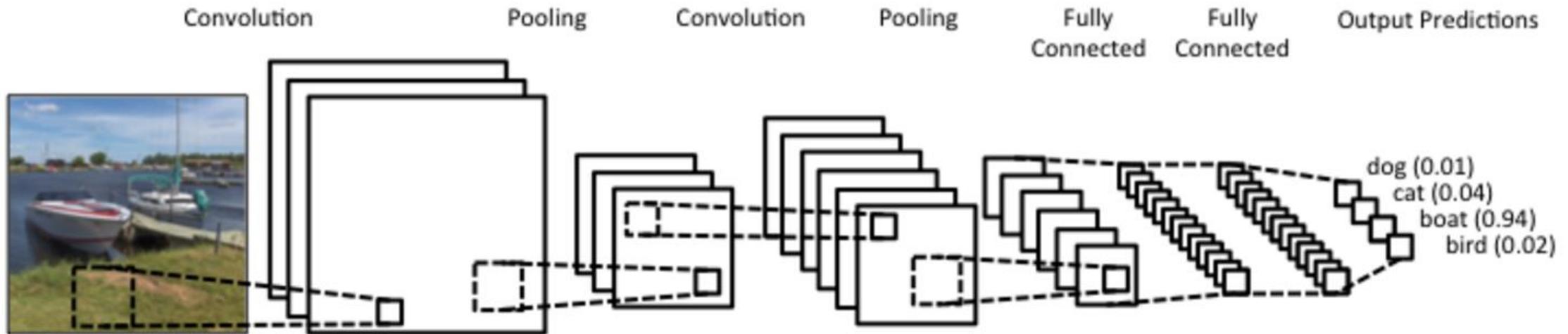
<http://colah.github.io/posts/2015-08-Understanding-LSTMs/>

Applications :

- Traitement du langage
- Speech to text / text to speech
- Analyse de séries temporelles



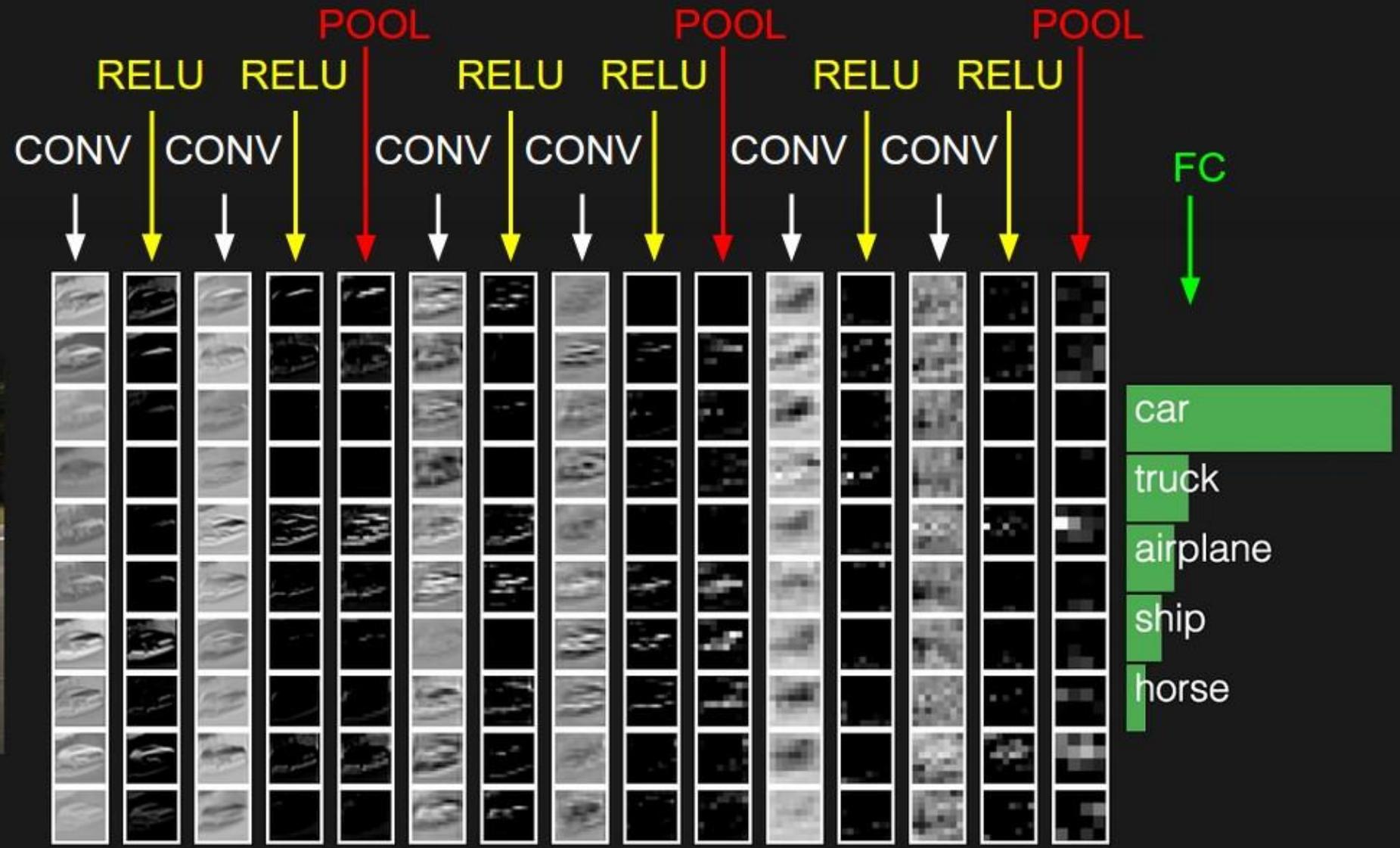
Réseaux convolutifs



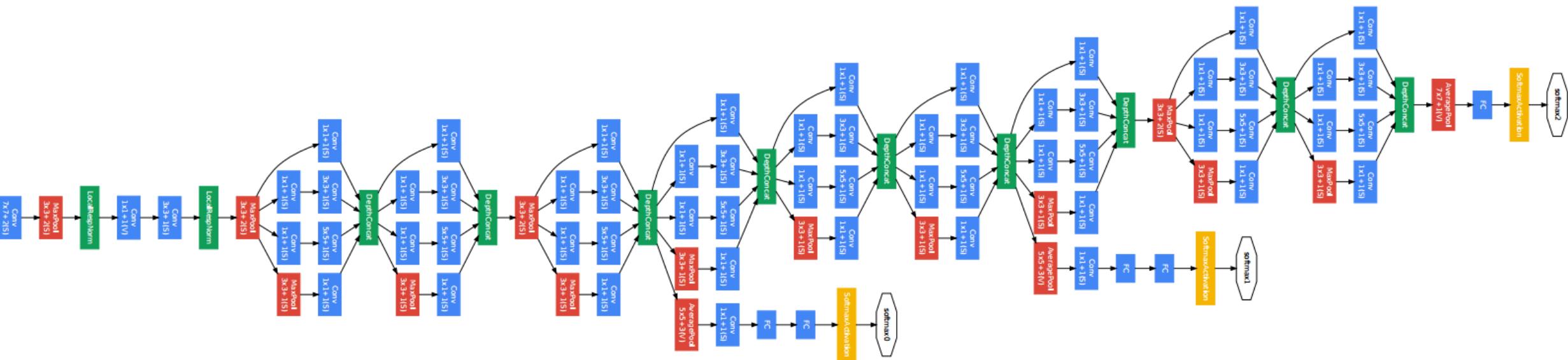
Convolution : Travailler (en 2D) sur des portions d'image

Pooling : Réduire la dimension progressivement jusqu'au couches « fully connected »

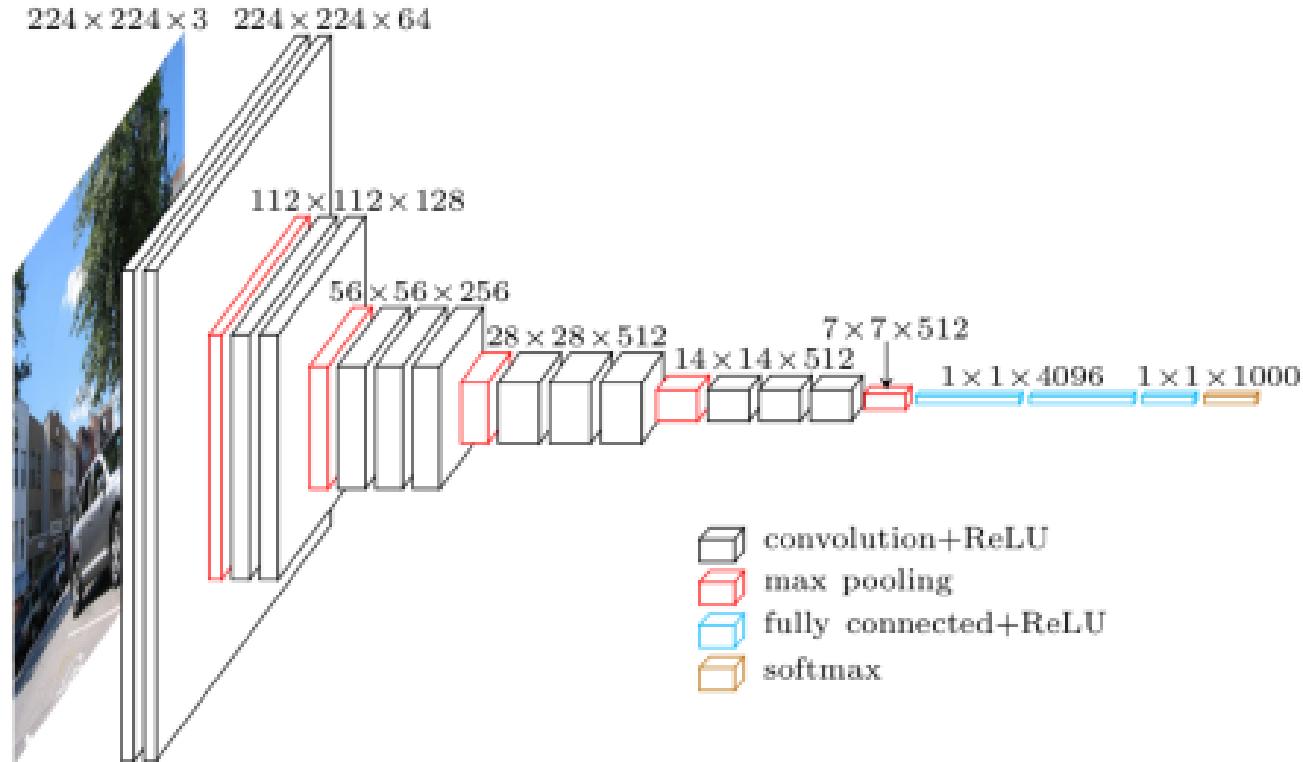
Réseaux convolutifs



GoogleNet

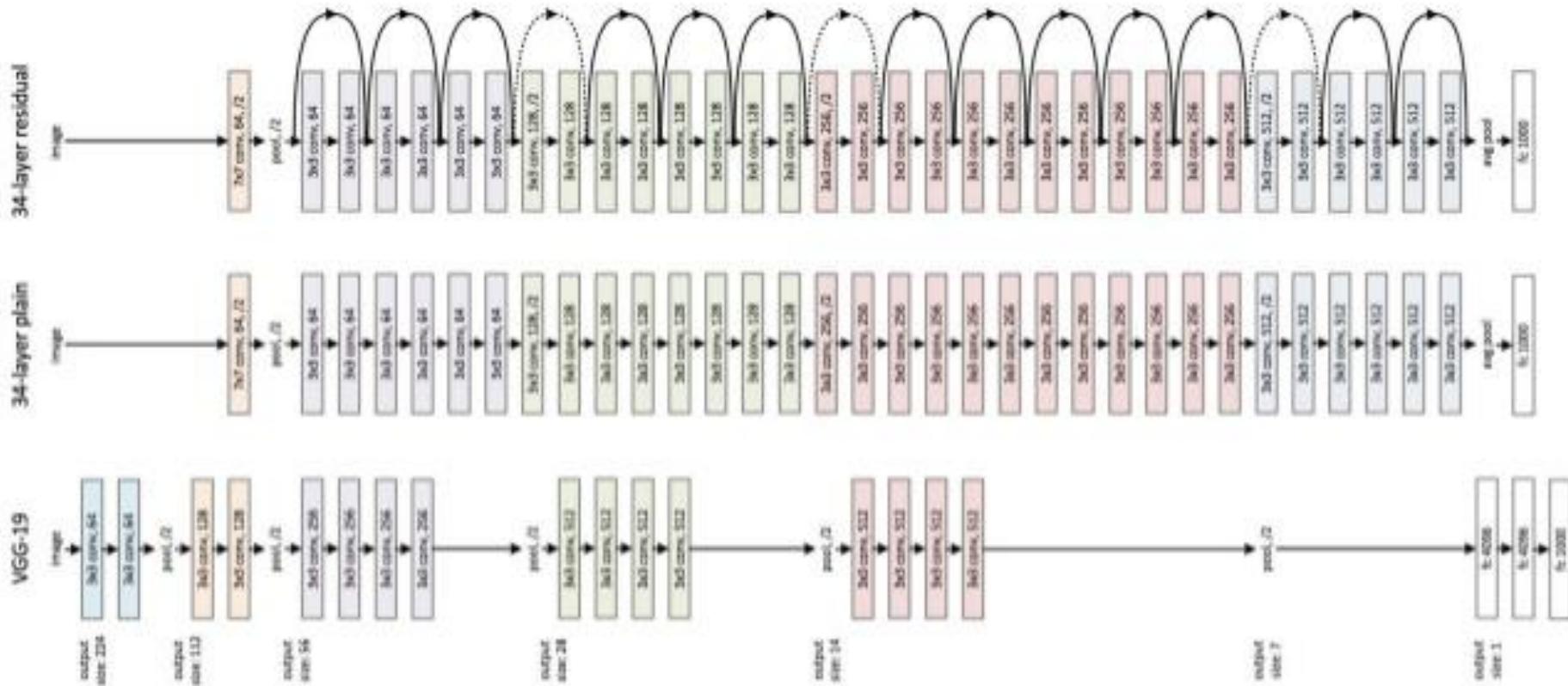


VGG16



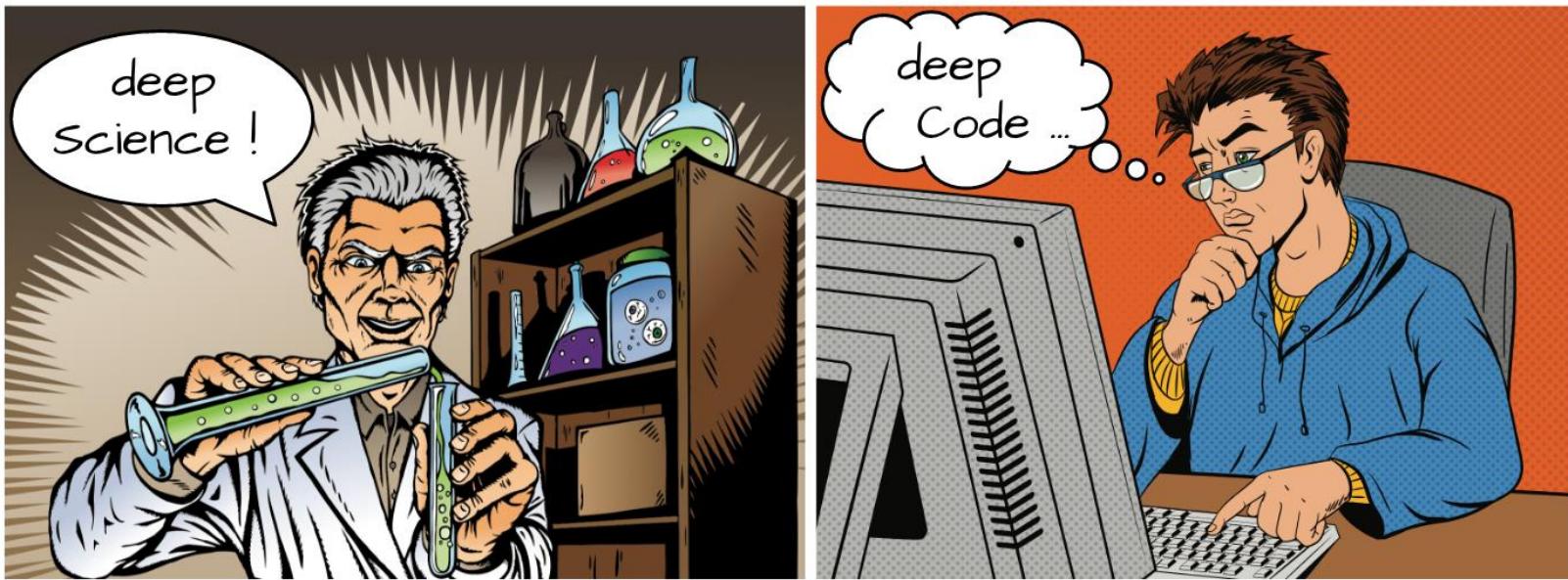
ResNet, encore plus profond

ResNet



Réseaux convolutifs : aller plus loin

>TensorFlow and deep learning_
without a PhD



#Tensorflow

Google Cloud Platform

@martin_gorner

<https://codelabs.developers.google.com/codelabs/cloud-tensorflow-mnist/#0>
<https://www.youtube.com/watch?v=vq2nnJ4g6N0>

Implémenter un use-case data

De la donnée brute à la mise en production

Workflow

1. Identifier une problématique, une question métier
2. Identifier les données pertinentes
3. Valider les données
 1. Nettoyage
 2. Preprocessing et feature engineering
4. Formaliser le use-case
 1. Modèle prédictif (ML)
 2. Protocole de validation
5. Industrialisation

Les compétitions de ML ne sont pas réalistes



Compétitions Data Science
Compétitions Machine Learning

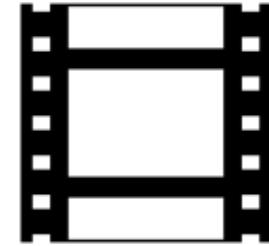
À considérer :

- Jeux de données propres
- Questions bien posées
- Focus sur la maximisation du score prédictif

Très bon pour monter en compétence sur le ML

Peu de rapport avec le travail du Data Scientist

Le syndrome Netflix



1M\$ pour améliorer la recommandation de film de 10%

3 ans pour un algorithme inutilisable en production

Le pipeline : au cœur de la Data Science

Il faut avoir une vision globale du flot de données (ETL, nettoyage, etc)

- Il est préférable de conserver les données brutes, et de reproduire les étapes de transformation
- On peut garder des étapes intermédiaires en cas de besoin
- Tester les différentes étapes et automatiser pour valider les étapes à chaque itération / changement

Exemple : F-score borné entre précision et rappel
 $\min(\text{precision}, \text{recall}) < \text{f-score} < \max(\text{precision}, \text{recall})$

- Attention à la gestion des dépendances et ordonnancement

Le pipeline doit être automatisable et réplicable

End-to-end Data Science : quelques outils*

Gestion de flux



Persistance



Processing et prédictif



Restitution



UX/UI

API &
webservices

*Image incomplète et biaisée par mon expérience personnelle

Sources et formats de données

Sources de données

BDD
API
Dépôts
serveur/cloud
Documents
Excel
HDFS



Modalités de transfert

Requêtes (BDD, API)
Échanges indirects
(dépôts ftp, cloud)
Échanges directs
(sérialisation, sockets)



Formats des données

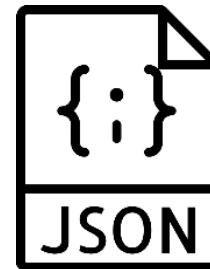
CSV
JSON
XML
txt
pdf
Binaires
médias

Formats « populaires »

**SI, data warehouse
datalake**

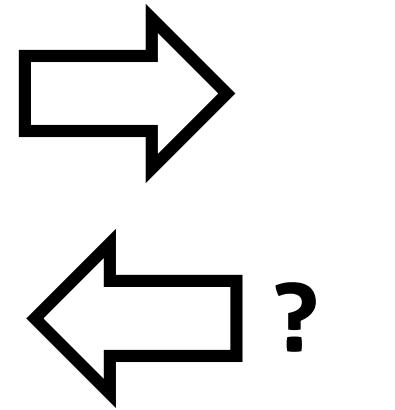


**Serveurs emails
APIs (twitter, google, etc)**



Jongler entre les formats

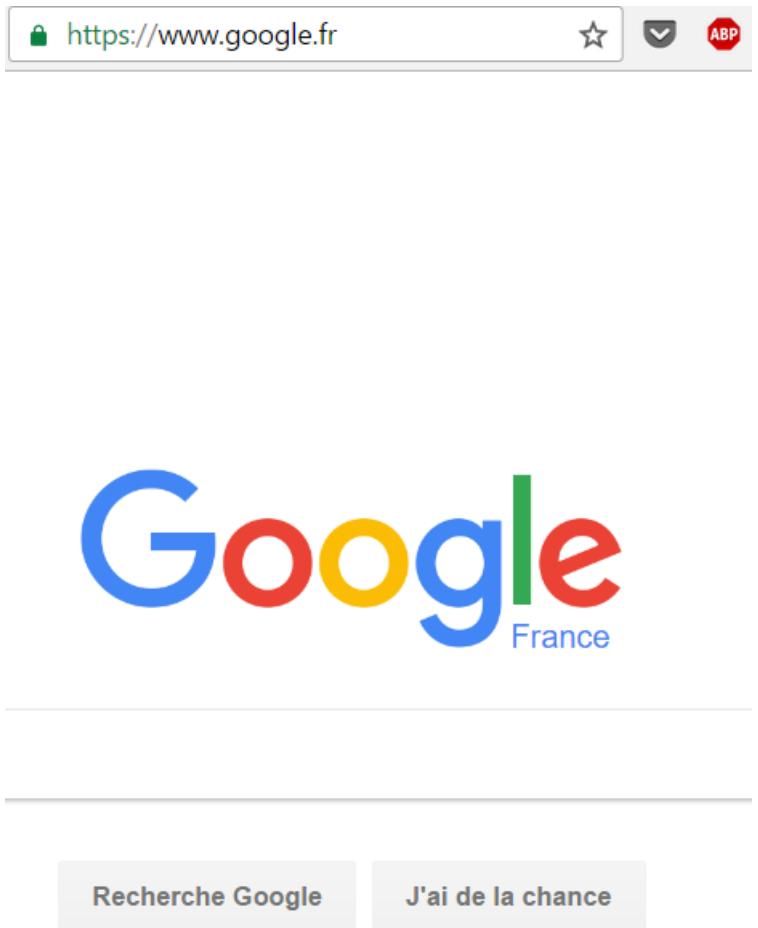
Id	First name	Last name	Entreprise
1	Amine	Benhenni	EPA, Dataswati



Mettre à plat des structures hiérarchiques complexes nécessitent de repenser en relationnel et faire de la normalisation

Plus facile de passer dans un sens que dans l'autre

Interroger le web : scraping



```
curl -X GET http://www.google.fr
```

Interroger le web : API

The screenshot shows the Twitter Developer Documentation homepage. At the top, there's a navigation bar with links for Developers, Products, Documentation, Community, Build, My apps, and a blue 'Join' button. Below the navigation is a large blue header with the text 'Twitter Developer Documentation'. Underneath the header, a breadcrumb trail shows 'Docs / REST APIs'. On the left side, there's a dark sidebar titled 'Products & Services' containing links for Best practices, API overview, Websites, Cards, OAuth, REST APIs, API Rate Limits, Rate Limits: Chart, The Search API, The Search API: Tweets by Place, and Working with Timelines. The main content area has a title 'REST APIs' and a paragraph explaining that the REST APIs provide programmatic access to read and write Twitter data. It mentions OAuth and JSON format. A note says to consider the Streaming API for real-time monitoring. Below this is an 'Overview' section with a list of documents to get started with the REST APIs.

Twitter Developers Products Documentation Community Build My apps Join

Twitter Developer Documentation

Docs / REST APIs

Products & Services

- Best practices
- API overview
- Websites
- Cards
- OAuth
- REST APIs
 - API Rate Limits
 - Rate Limits: Chart
 - The Search API
 - The Search API: Tweets by Place
 - Working with Timelines

REST APIs

The REST APIs provide programmatic access to read and write Twitter data. Create a new Tweet, read user profile and follower data, and more. The REST API identifies Twitter applications and users using OAuth; responses are in JSON format.

If your intention is to monitor or process Tweets in real-time, consider using the Streaming API instead.

Overview

Below are some documents that will help you get going with the REST APIs as quickly as possible

- API Rate Limiting
- API Rate Limits
- Working with Timelines
- Using the Twitter Search API
- Finding Tweets about Places
- Uploading Media
- Reference Documentation

Reference Documentation

These are the REST API endpoint reference docs.

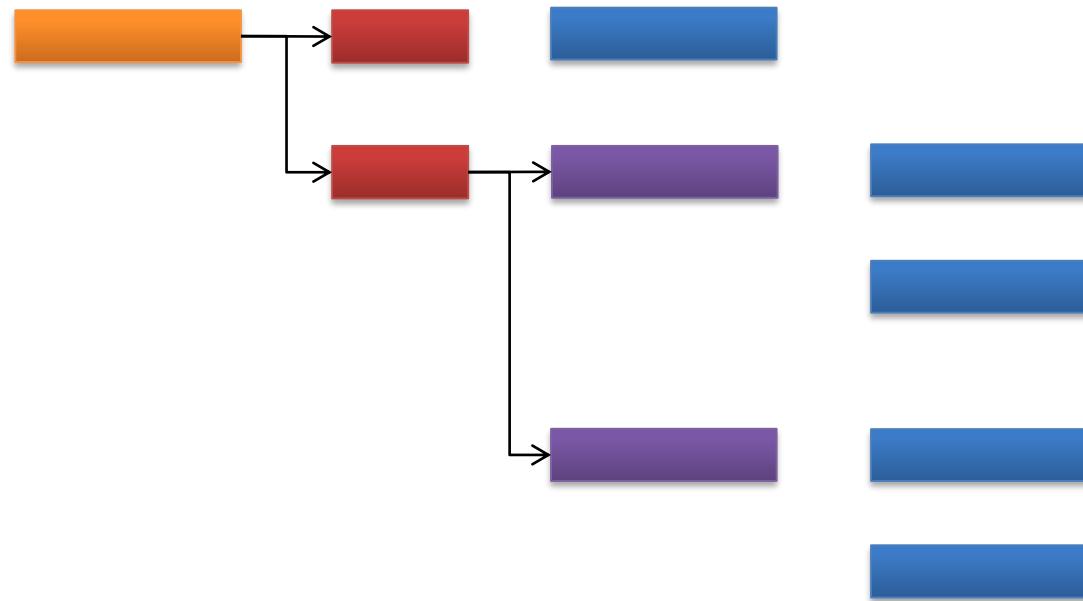
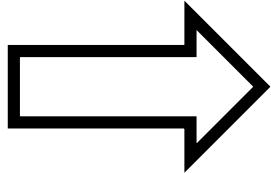
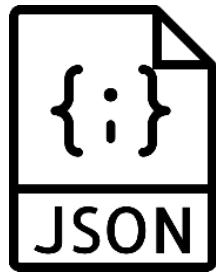
GET

- GET account/settings
- GET account/verify_credentials
- GET application/rate_limit_status
- GET blocks/ids
- GET blocks/list
- GET collections/entries
- GET collections/list

Différents points d'accès pour avoir différentes informations

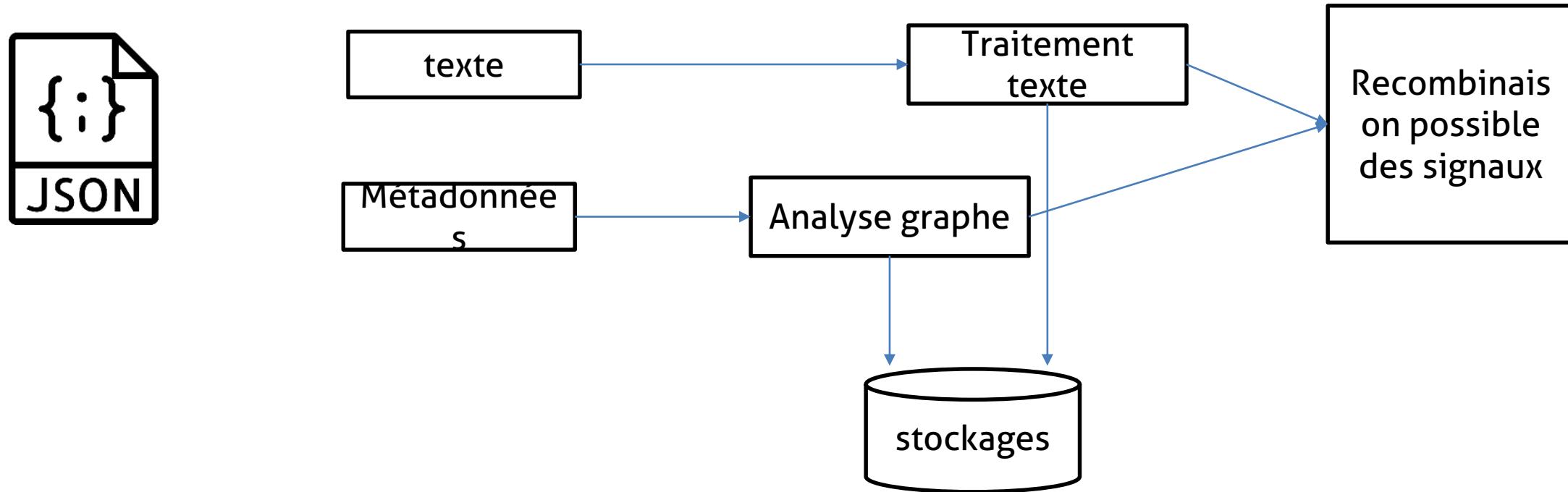
API et MongoDB

GET http://.../api



Le stockage conserve la forme exacte de la donnée brute

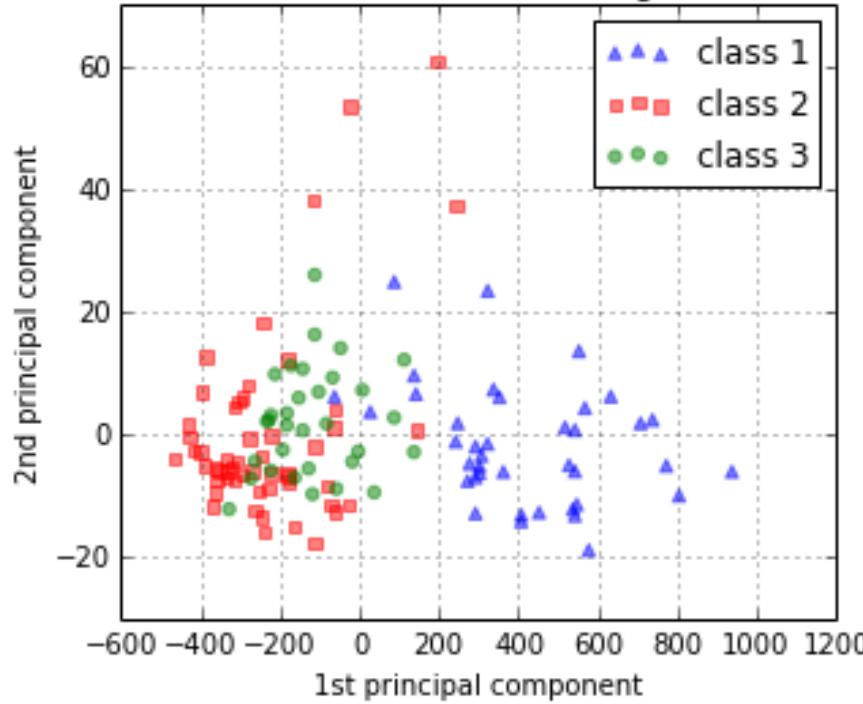
Schéma d'un pipeline de traitement d'un tweet



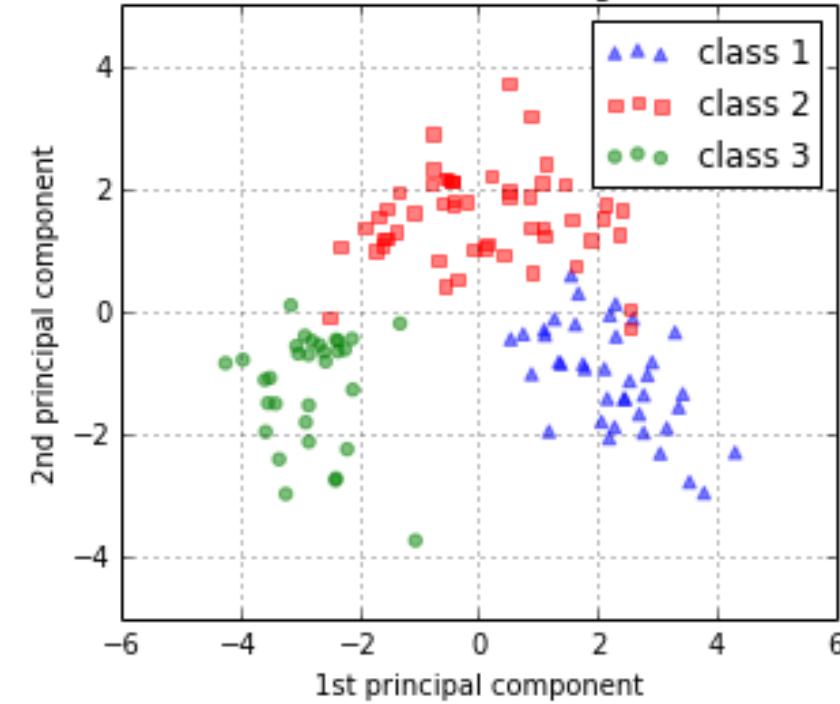
Également pour email et des pages web

Transformations sur les variables

Transformed NON-standardized training dataset after PCA



Transformed standardized training dataset after PCA



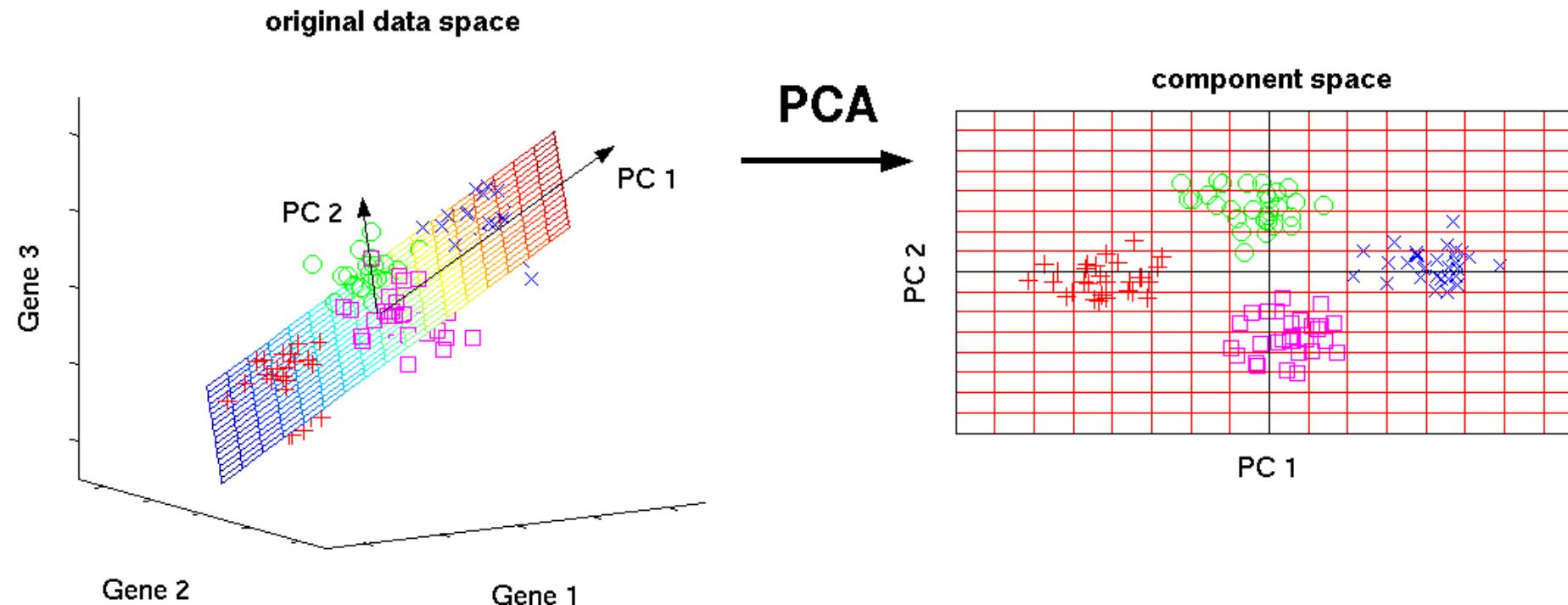
$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

Borner entre valeur min et valeur max

$$z = \frac{x - \mu}{\sigma}$$

Centrer (on enlève la moyenne), et réduire de l'écart-type

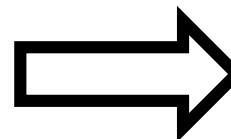
Réduire l'espace initial



Réduction de dimension par PCA

Travailler avec des dates

2016-12-01 14:00:00

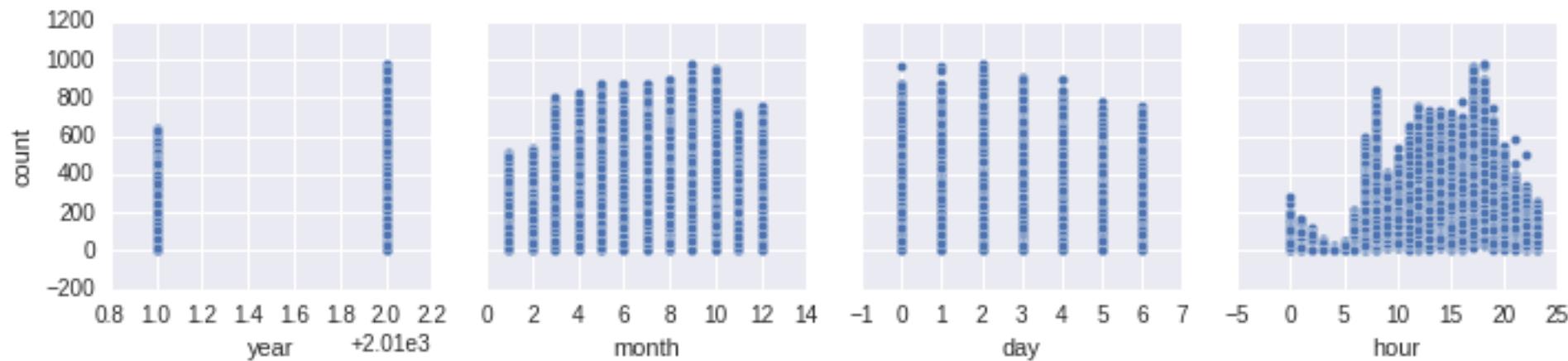


Heure = 14

Jour_semaine = jeudi

Mois = 12

Année = 2013

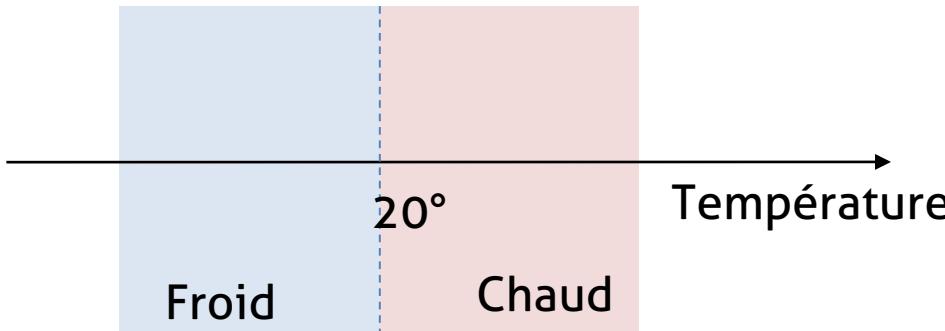


Capturer explicitement des phénomènes saisonniers

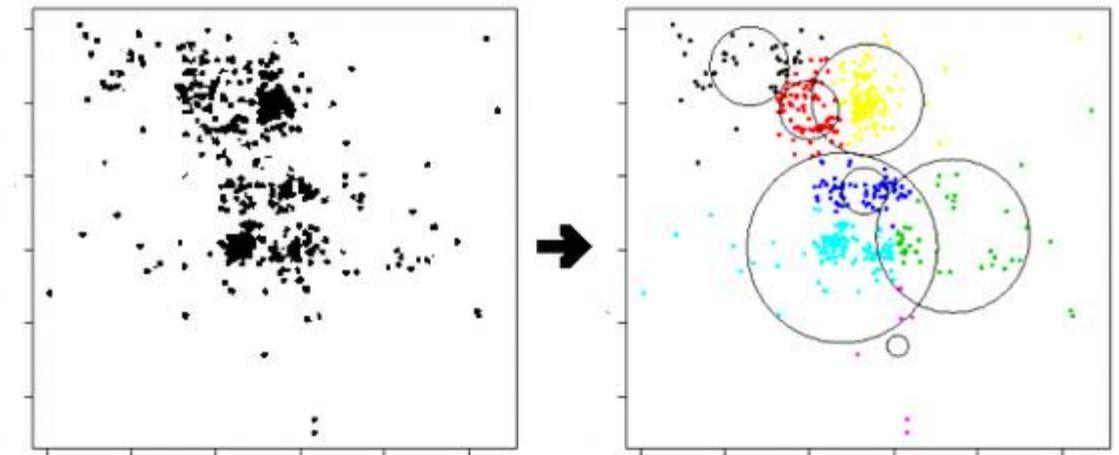
Discrétisation

Passer de valeurs continues à des valeurs discrètes

Simplifier certaines tâches prédictives, ou niveau de détail trop fin pour le besoin



Cas simple :
autant de catégories que nécessaire



Cas plus complexe : par clustering

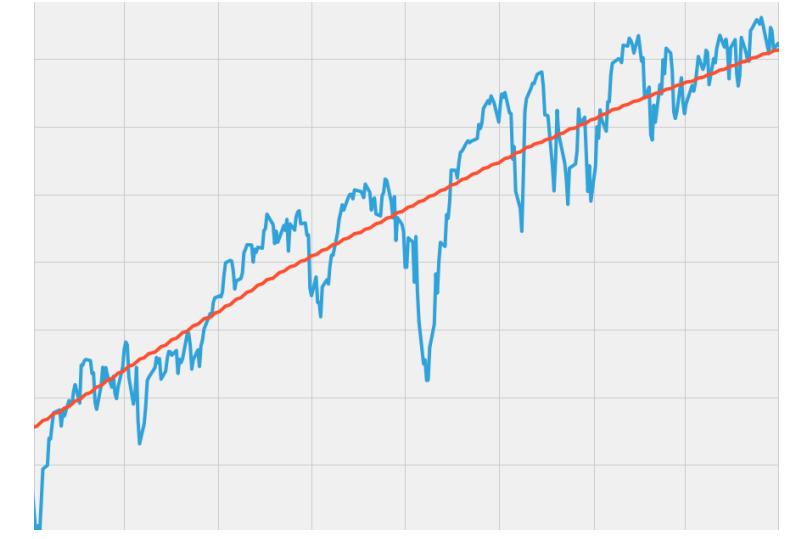
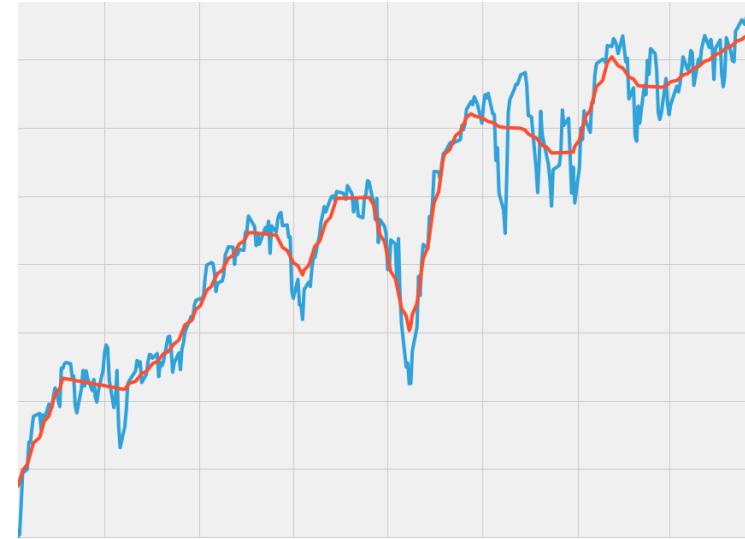
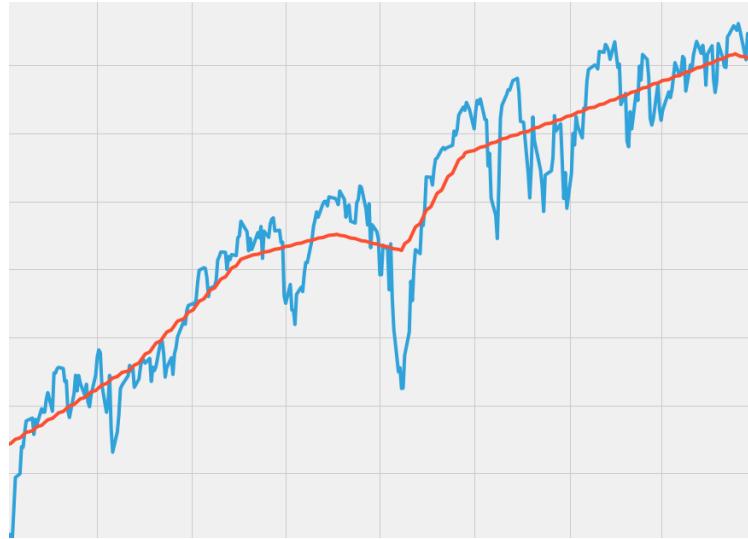
One-hot encoding

Pour des raisons pratiques, il est parfois nécessaire de représenter des variables discrètes par des nombres

lundi	→	1	→	(1,0,0,0,0,0)
mardi	→	2	→	(0,1,0,0,0,0)
...	
jeudi	→	4	→	(0,0,0,1,0,0)

Vecteurs indépendants (produit scalaire nul)

Lissage de timeseries



Atténuer le bruit et ne garder que le signal, mais jusqu'où aller ?

Mise en production

Des compétences et des outils complémentaires

Pousser une API prédictive avec python

```
from flask import Flask, jsonify, request
from sklearn import ...

app = Flask(__name__)

@app.route('/train', methods=['POST'])
def model_train():
    ...
    return jsonify(response)

@app.route('/predict', methods=['POST'])
def model_predict():
    ...
    return jsonify(response)

if __name__ == '__main__':
    app.run(host='0.0.0.0', port=8080)
```

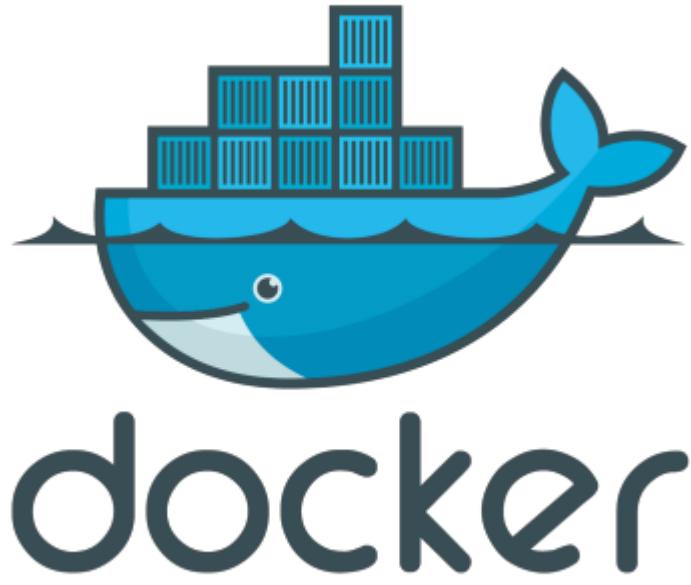
Flask : framework web léger en python
sklearn : librairie ML

Fonction de training,
associé à l'URL : http://.../train

Fonction prédictive,
associé à l'URL : http://.../predict

Boucle d'événements qui attend les requêtes

Prêt à être déployé avec docker



- Environnement « portable »
- Projets isolés
- Plus modulaire et plus souple
- Container peuvent être sur la même machine ou sur différentes machines

Exemple : Testez rapidement tensorflow

The screenshot shows a portion of the TensorFlow website's "Getting Started" page. At the top, there is a yellow header bar with the TensorFlow logo. Below it, a sidebar on the left lists navigation options: "Version: r0.12" (with a dropdown arrow), "Introduction", "Recommended Next Steps", "Download and Setup", and "Requirements". The main content area contains instructions for running a Docker container. It says: "After Docker is installed, launch a Docker container with the TensorFlow binary image as follows." Below this is a code block:

```
$ docker run -it -p 8888:8888 gcr.io/tensorflow/tensorflow
```

. Further down, it explains: "The option -p 8888:8888 is used to publish the Docker container's internal port to the host machine, in this case to ensure Jupyter notebook connection."

After Docker is installed, launch a Docker container with the TensorFlow binary image as follows.

```
$ docker run -it -p 8888:8888 gcr.io/tensorflow/tensorflow
```

The option -p 8888:8888 is used to publish the Docker container's internal port to the host machine, in this case to ensure Jupyter notebook connection.

Créer rapidement un container

Dockerfile

```
FROM ubuntu:16.04
MAINTAINER Amine Benhenni "albenhenni@dataswati.com"

RUN apt-get update -y

RUN apt-get install -y python-pip python-dev libev4 libev-dev gcc libxml2-dev libffi-dev vim curl
RUN pip install --upgrade pip

RUN apt-get install -y python-numpy python-scipy
RUN pip install scikit-learn
RUN pip install flask-restful
RUN pip install pandas
```

docker build .

```
$ docker build .
Sending build context to Docker daemon 3.619 GB
Step 1 : FROM ubuntu:16.04
--> e4415b714b62
Step 2 : MAINTAINER Amine Benhenni "albenhenni@dataswati.com"
--> Using cache
--> 8cc5c9101295
Step 3 : RUN apt-get update -y
--> Using cache
--> 14604f8d7596
Step 4 : RUN apt-get install -y python-pip python-dev libev4 libev-dev gcc libxml2-dev libffi-dev vim curl
--> Using cache
--> 56e7a1fc18e8
Step 5 : RUN pip install --upgrade pip
--> Using cache
--> 2d35685ed9b9
Step 6 : RUN apt-get install -y python-numpy python-scipy
--> Using cache
--> 29d0f8a5d36a
Step 7 : RUN pip install scikit-learn
--> Using cache
--> 7a5662ecc81a
Step 8 : RUN pip install flask-restful
--> Using cache
--> 8e75e3df8891
Step 9 : RUN pip install pandas
--> Using cache
--> f7ef51151c1c
Successfully built f7ef51151c1c
SECURITY WARNING: You are building a Docker image from Windows against a non-Windows Docker host. All files and direct
ories added to build context will have '-rwxr-xr-x' permissions. It is recommended to double check and reset permissio
ns for sensitive files and directories.
```

Le container est disponible

Architectures modulaires

docker-compose.yml

api : contient le code de l'application qui est accessible sur le port 8181

api a besoin de redis pour fonctionner (cache local)

« docker-compose up » :
l'architecture est déployé en une ligne de commande

```
api:
  build: ./docker/api/
  links:
    - redis
  volumes:
    - ./app:/app
  expose:
    - "8081"
  ports:
    - "8081:8081"
mongodb:
  image: mongo:3.1.5
  command: mongod --config /etc/mongod.conf
#  command: mongod --auth --setParameter enableLocalhostAuthBypass=1
  volumes:
    - ./data/etc/mongod.conf:/etc/mongod.conf
    - ./data/db:/data/db
  expose:
    - "27017"
  ports:
    - "27017:27017"
rabbitmq:
  image: rabbitmq:3.5.3-management
  environment:
    - RABBITMQ_DEFAULT_USER=[REDACTED]
    - RABBITMQ_DEFAULT_PASS=[REDACTED]
  expose:
    - "15672"
  ports:
    - "15672:15672"
neo4j:
  image: tpires/neo4j
  volumes:
#    - /var/lib/neo4j/data
    - ./data/etc/neo4j-wrapper.conf:/var/lib/neo4j/conf/neo4j-wrapper.conf
    - ./data/graph:/var/lib/neo4j/data
  environment:
    - NEO4J_AUTH=[REDACTED]
  expose:
    - "7474"
  ports:
    - "7474:7474"
redis:
```

Dockerfile

Code de l'application

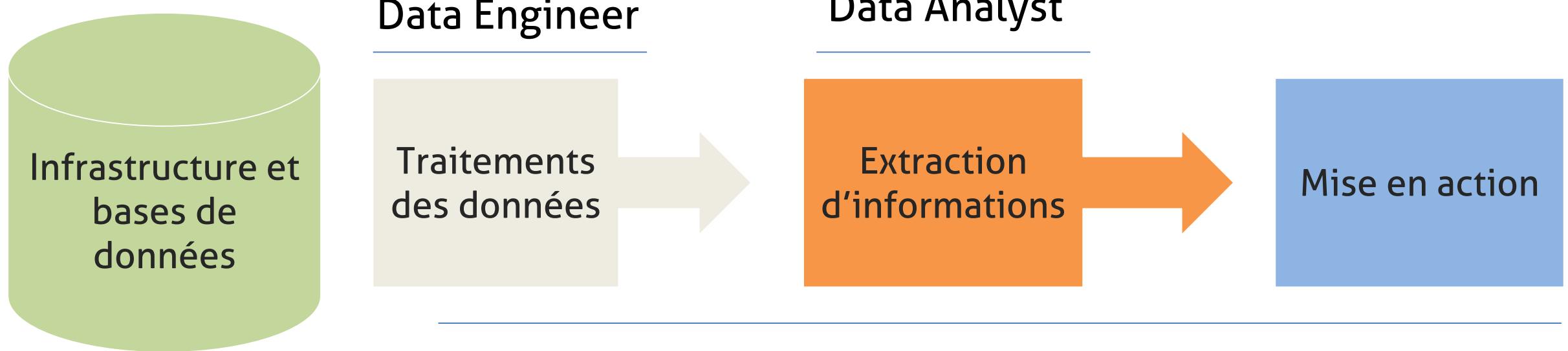
Kitematic (windows)

 kitematic hello-world-nginx A light-weight nginx container that demonstrates the features of Kitematic	 official ghost Ghost is a free and open source blogging platform written in JavaScript	 official jenkins Official Jenkins Docker image
 official redis Redis is an open source key-value store that functions as a data structure server.	 official rethinkdb RethinkDB is an open-source, document database that makes it easy to build and scale realtime	 kitematic minecraft The Minecraft multiplayer server allows two or more players to play Minecraft together
 official solr Solr is the popular, blazing-fast, open source enterprise search platform built on Apache	 official elasticsearch Elasticsearch is a powerful open source search and analytics engine that makes data easy to	 official postgres The PostgreSQL object-relational database system provides reliability and data integrity.
 official ubuntu-upstart Upstart is an event-based replacement for the /sbin/init daemon which starts processes a...	 official memcached Free & open source, high-performance, distributed memory object caching system.	 official rabbitmq RabbitMQ is a highly reliable enterprise messaging system based on the emerging AMQP
 official celery Celery is an open source asynchronous task queue/job queue based on distributed...	 official mysql MySQL is a widely used, open-source relational database management system (RDBMS).	 official mongo MongoDB document databases provide high availability and easy scalability.

Modalités de restitution (UX/UI)

- API
- Dashboards
- Dataviz
- Chatbot
- Emails
- SMS
- Objets connectés
- Chatbot
- Lunettes de réalité augmentée
- ...

Data driven : mettre en action la donnée



Data Architect

Data Scientist

Pas de barrières de la donnée brute aux use cases métiers

Data Scientist et compétences techniques

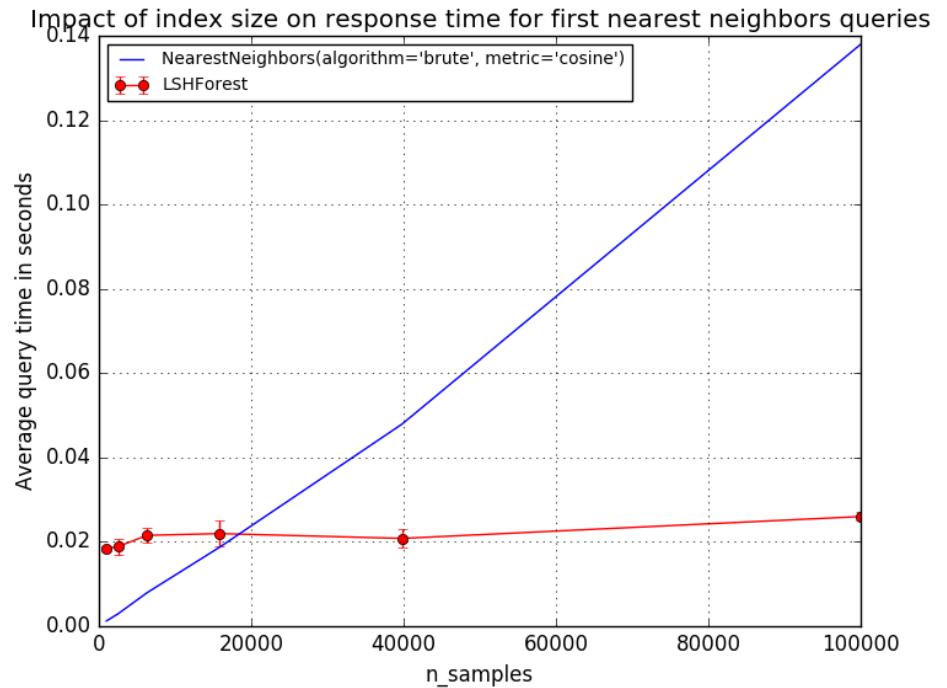
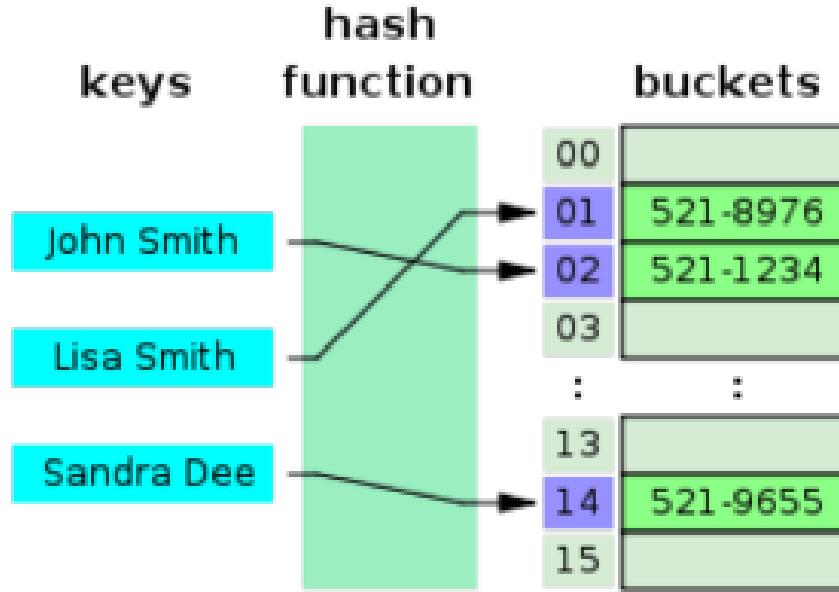
Multiplication naïve de matrices

1000	< 1 seconde
10000	~ 20 secondes
100000	MemoryError !

Tf-Idf, calculs de similarité, etc

Le Data Scientist doit maîtriser les contraintes techniques de ses projets (en amont, et en aval sur les contraintes d'industrialisation), même dans le cas où il n'assure pas la mise en production (si Data Engineer dans l'équipe)

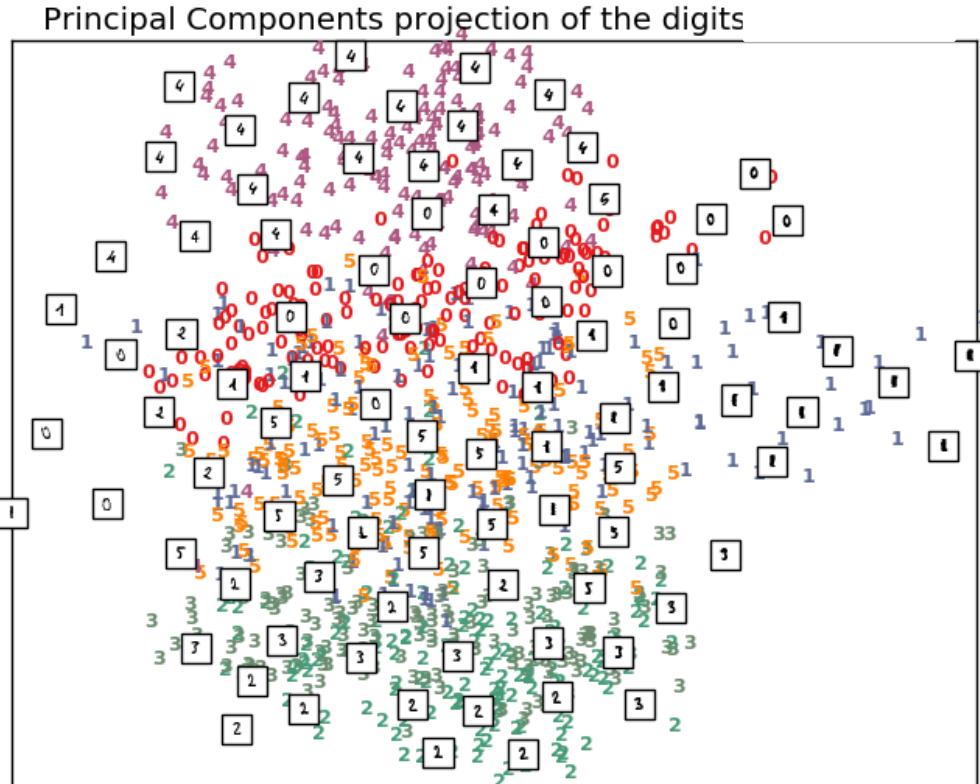
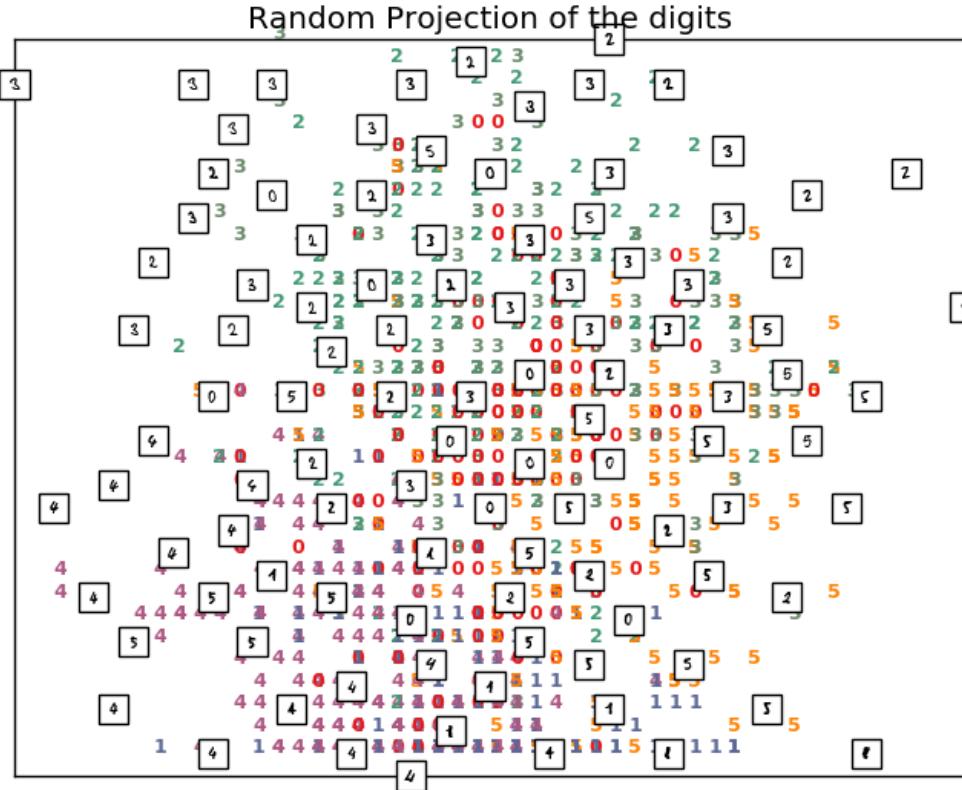
Prédiction à l'échelle : exemple calcul de similarité



Comparaison avec les implémentations de sklearn

On ne compare plus tous les éléments 1 à 1, mais on les classe dans des cases de sorte que les voisins se trouvent au même endroit

Réduction de dimension



Des résultats moins robustes (mais pertinents avec un volume suffisant)
mais avec de meilleures performances à l'échelle

Formalisation des use cases

Transformer des questions métiers en problèmes prédictifs

Vous êtes Data Scientist

Un client vient vous voir (interne ou externe)

Je souhaite faire :

- De la connaissance client
- De la lutte contre la fraude
- De la maintenance prédictive
- De l'optimisation de process
- ...

Attention au techno-push !

Métriques ML et indicateurs métiers

Métriques :

- RMSE
- RMLSE
- Log Loss
- etc



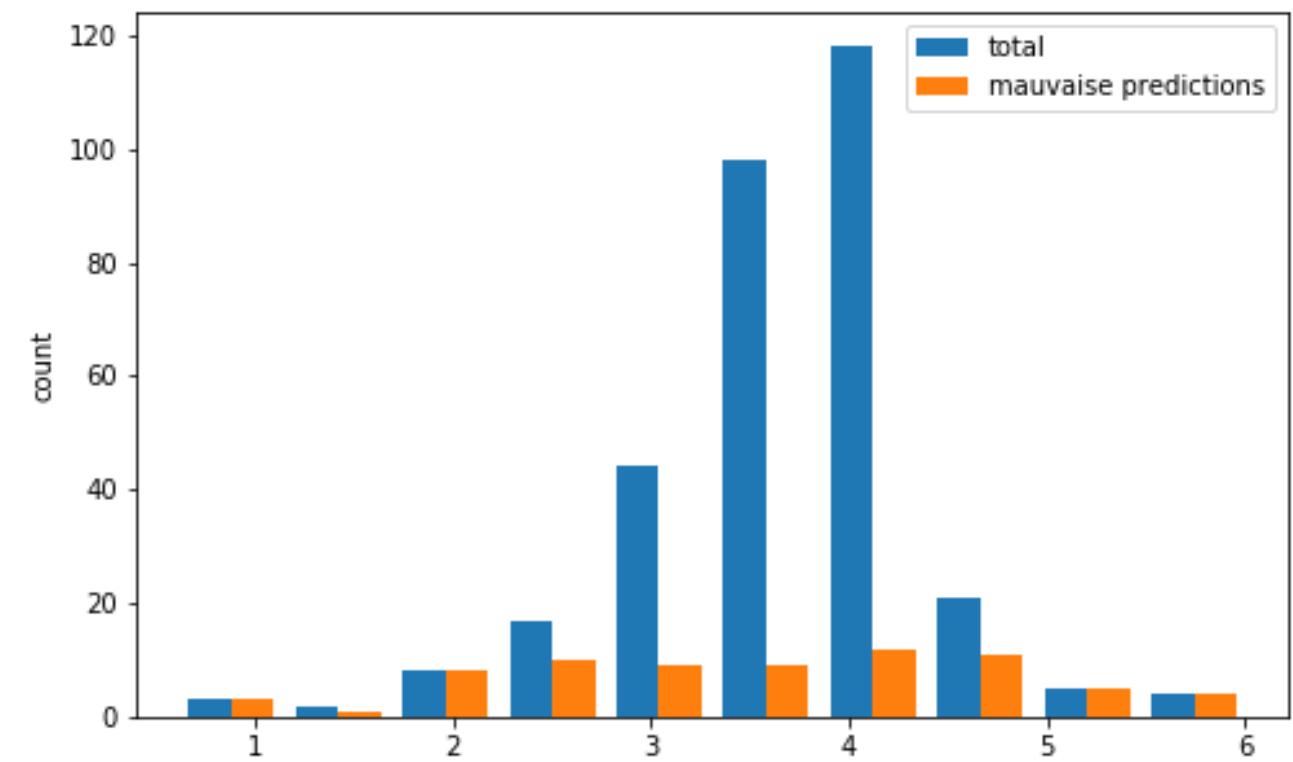
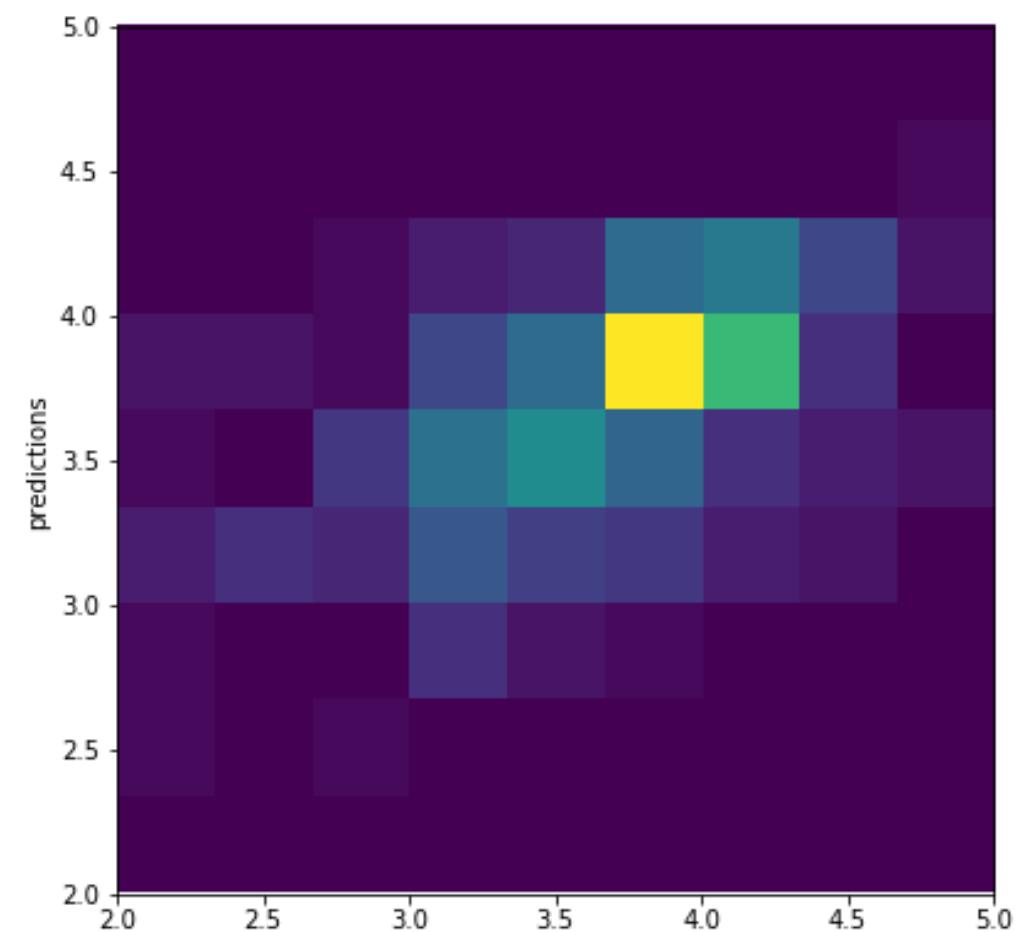
N'ont aucun sens métier !

Utiles pour choisir et valider des modèles, mais aucun intérêt pour le client.

Il faut faire le travail de convertir ses résultats en quelque chose de quantifiable pour le client

Par exemple : mes prédictions sont 80% du temps dans l'incertitude de mesure

Articuler l'impact business des résultats obtenus



Indicateurs définis en fonction des priorités

		Prédit	
		Positif	Négatif
Réel	Positif	TP	FN
	Négatif	FP	TN

$$\text{Précision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

$$\text{Rappel} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

On peut avoir d'autres priorités

- Pas de **faux positifs** car coûtent trop chers et mobilisent des ressources
- Pas de **faux négatifs**, on ne peut rien laisser passer car trop dangereux

Attentions aux indicateurs

On utilise des indicateurs tous les jours dans notre vie courante pour prendre des décisions
Il s'agit souvent d'indicateurs indirects qu'on estime corrélés à ce qu'on souhaite mesurer

Quelques exemples dans différents domaines

- Nombre de personnes faisant la queue devant un restaurant
- Nombre de lignes de code pour juger le travail d'un développeur
- Nombre de publications pour juger le travail d'un chercheur
- Nombre de CV envoyés pour une personne en recherche de travail
- Volume de données disponible pour valider la faisabilité d'un projet data
- etc

Data Use Case Canvas

Définition et pilotage d'un produit data

Causes d'échec qui peuvent être facilement évitées

Vous entassez la donnée dans un datawarehouse ou un datalake mais ne savez pas encore quoi en faire ?

Vous avez un « lab » qui a validé 2/3 algos, mais rien ne tourne encore en production ?

Vous confondez data science et analyse de données ?

Vous avez un cahier des charges fixe et un projet de développement sur plusieurs mois ?

Votre modèle en production est largement moins bon que lors de la validation hors-ligne ?

Data Use Case Canvas

USERS

Désigne l'ensemble des utilisateurs impactés par le projet data considéré.
(ex : régleur)



USE CASE

Définit l'ensemble des fonctions, actions et contenus du projet data considéré.
(ex : afficher le calcul de performance, ...)



UX DESIGN

Définit l'ensemble des contraintes d'interfaces du projet data considéré.
(ex : rendu visuel sur tablette 12", rendu dataviz sur écran (>27"))



CONSTRAINTES

Définit les contraintes obligatoires pour la réussite du projet data considéré.
(ex : calcul de la performance en moins de 2')



DATA

Définit les sources de données nécessaires, leur format et leur accessibilité du projet data considéré.
(ex : données réglage machines X20032 | CSV | SFTP)