

Introduction to: Machine Learning

Diane Lingrand



UNIVERSITÉ
CÔTE D'AZUR
SI 4

2023 - 2024

Pour communiquer

- le **slack** du cours #si4-ml, pour les questions d'ordre général pouvant intéresser les autres étudiants du cours.
 - Tout le monde peut répondre aux questions posées, pas seulement les enseignants !
 - Cela permet également de corriger les fausses vérités lors des révisions.
- la page **moodle** du cours :
<https://lms.univ-cotedazur.fr/course/view.php?id=11331>
 - les supports de cours, les sujets de TP, les rendus et autres documents
 - Si vous n'êtes pas déjà inscrit, il faut s'auto-inscrire sur le cours moodle pour pouvoir accéder aux rendus !
- les discussions privées sur **slack** ou par **mail** avec les enseignants :
 - pour prévenir d'une absence (non, ce n'est pas ringuard, c'est de la politesse), ...
 - prenom.nom@univ-cotedazur.fr

1 Les algorithmes d'apprentissage

2 Préparation des données

3 Analyse des résultats

4 Des datasets pour 'jouer'

- **Introduction**

- Termes généraux de l'apprentissage automatique
- Préparation des données
- Évaluation des classifications
- Des données pour 'jouer' avec les différents algorithmes
- Des bibliothèques (`scikit-learn`, `tensorflow.keras`)

Outline

1 Les algorithmes d'apprentissage

2 Préparation des données

3 Analyse des résultats

4 Des datasets pour 'jouer'

Familles d'algorithmes

- Apprentissage non supervisé : nécessite des données
 - Clustering / Partitionnement
 - Représentation
 - Génération de nouvelles données
 - Détection d'anomalies
- Apprentissage supervisé : nécessite des données ET des labels
 - Régression
 - Régression linéaire / non-linéaire, Réseaux de neurones (incluant les réseaux profonds ou *deep learning*), SVM, arbres de décision et forêts aléatoires
 - Classification
 - Régression logistique, Réseaux de neurones (incluant les réseaux profonds ou *deep learning*), SVM, arbres de décision et forêts aléatoires
 - Légendes, Segmentation, Compréhension (texte, oral)

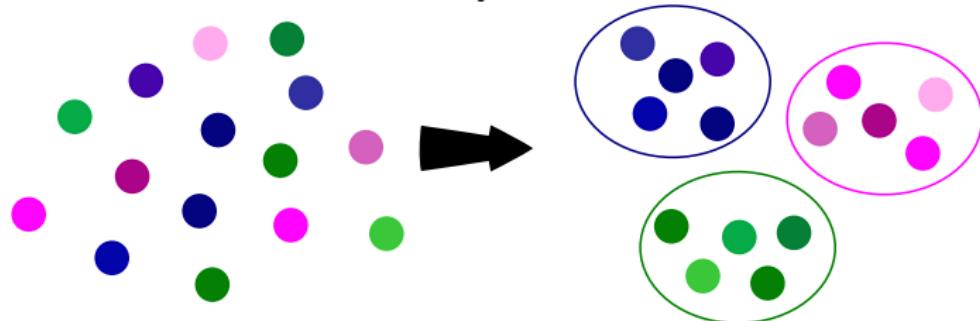
Familles d'algorithmes

- Apprentissage non supervisé : nécessite des données
 - Clustering / Partitionnement
 - Représentation
 - Génération de nouvelles données
 - Détection d'anomalies
- Apprentissage supervisé : nécessite des données ET des labels
 - Régression
 - Régression linéaire / non-linéaire, Réseaux de neurones (incluant les réseaux profonds ou *deep learning*), SVM, arbres de décision et forêts aléatoires
 - Classification
 - Régression logistique, Réseaux de neurones (incluant les réseaux profonds ou *deep learning*), SVM, arbres de décision et forêts aléatoires
 - Légendes, Segmentation, Compréhension (texte, oral)
- Autres modes d'apprentissage :
 - apprentissage semi-supervisé
 - apprentissage actif
 - apprentissage zéro exemple (ZSL) ou très peu d'exemples
 - renforcement (données de récompenses), imitation
 - apprentissage fédéré

- Algorithmes de clustering
 - k-means, mixture de gaussienne, hierarchiques, basés sur la densité
- Algorithmes de regression
 - linéaire, arbres de décision, SVM, réseaux de neurones
- Algorithmes de classification
 - linéaire, arbres de décision, SVM, réseaux de neurones
- Algorithmes ensemblistes :
 - boosting, forêts aléatoires
- Introduction à l'apprentissage profond
 - CNN (réseaux convolutionnels)

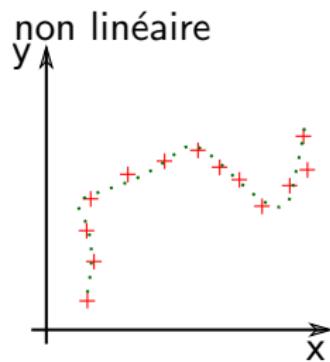
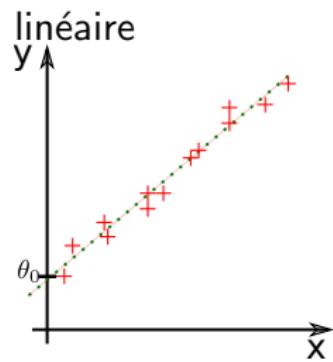
Partitionnement ou *clustering*

- **Unsupervised learning** : no class label needed.
- Find the class labels directly from the data.

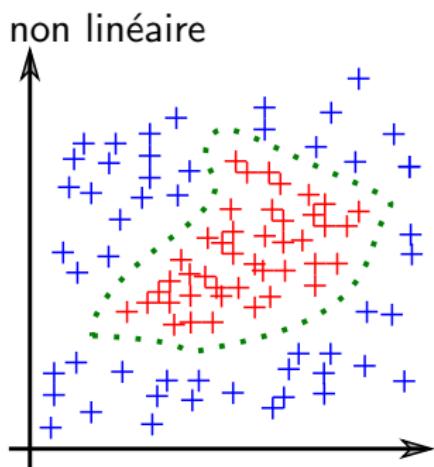
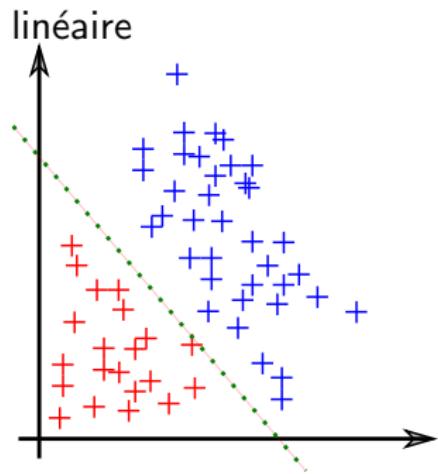


- Cluster : a collection of data objects
 - Similar to one another within the same cluster
 - Dissimilar to the objects in other clusters
 - Need a way to calculate object similarity/distance

Regression



Classification



- Arbres de décision
 - chaque décision concerne un paramètre
- Régression logistique
 - descente de gradient pour minimiser une fonction d'erreur logistique (sigmoïde)
- Réseau de neurones
 - modélisation de la séparation par différents neurones artificiels
- SVM
 - maximisation de la marge

Outline

1 Les algorithmes d'apprentissage

2 Préparation des données

3 Analyse des résultats

4 Des datasets pour 'jouer'

- Sélection des données
- Nettoyage des données
 - données dupliquées, entrées erronées, panne de capteurs
 - recherche des données aberrantes (pics, hors d'un intervalle)
 - données manquantes
- Codage / Transformation des variables
 - discrétisation
 - affectation d'un attribut discret
 - normalisation ou standardisation
 - agrégation (somme, moyenne ...)
 - calcul de variable
 - transformation
 - exemple des données géographiques (codes postaux)
 - exemple des durées / dates
 - combiner plusieurs variables (revenu/nb. d'enfants)
 - données multimédias (images, sons, vidéos, texte ...)

Outline

1 Les algorithmes d'apprentissage

2 Préparation des données

3 Analyse des résultats

4 Des datasets pour 'jouer'

- Données pour l'apprentissage (train)
 - classification : doit comporter des éléments de chaque classe
- Données de validation
 - afin de stopper l'apprentissage
 - afin de choisir des hyper-paramètres
- Données de test (test)
 - pour évaluer les performances de l'algorithme
 - classification : doit comporter des éléments de chaque classe dans les mêmes proportions que pour l'apprentissage
- Proportions couramment utilisées (train/valid/test) : 60% 20% 20% ou 80% 0% 20%
- Si le nombre de données est faible :
 - validation croisée (*cross-validation*)

Evaluation d'un algorithme de regression

- mse : mean square error

$$E = \frac{1}{n} \sum_{i=0}^n (y_{pred}^i - y_{truth}^i)^2$$

- rmse : root mean square error

$$E = \sqrt{\frac{1}{n} \sum_{i=0}^n (y_{pred}^i - y_{truth}^i)^2}$$

Evaluation d'une classification binaire : positifs et négatifs

vrais positifs (VP) : données positives calculées comme positives

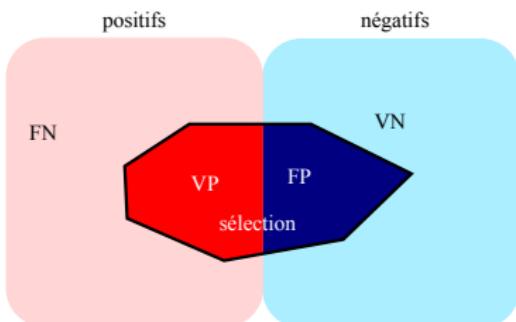
vrais négatifs (VN) : données négatives calculées comme négatives

faux positifs (FP) : données négatives calculées comme positives

faux négatifs (FN) : données positives calculées comme négatives

Matrice de confusion :

		classes estimées	
		positif	négatif
classes réelles	positif	9	1
	négatif	1	9



Vrais faux positifs négatifs

vrais positifs (VP) : données positives calculées comme positives

vrais négatifs (VN) : données négatives calculées comme négatives

faux positifs (FP) : données négatives calculées comme positives

faux négatifs (FN) : données positives calculées comme négatives

sensibilité, rappel, taux de VP : $\frac{VP}{VP+FN}$

spécificité, sélectivité, taux de VN : $\frac{VN}{VN+FP}$

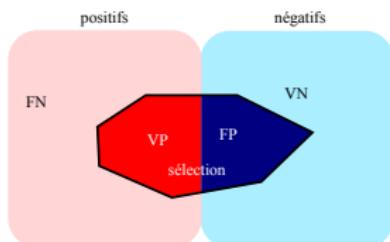
précision (*precision*), v. de préd. pos. : $\frac{VP}{VP+FP}$

valeur de prédition négative : $\frac{VN}{VN+FN}$

F-mesure (*F1 score*) :

$$2 \frac{\text{précision.rappel}}{\text{précision+rappel}} = \frac{2VP}{2VP+FP+FN}$$

précision (*accuracy*) : $\frac{VP+VN}{P+N} = \frac{VP+VN}{VP+FP+VN+FN}$

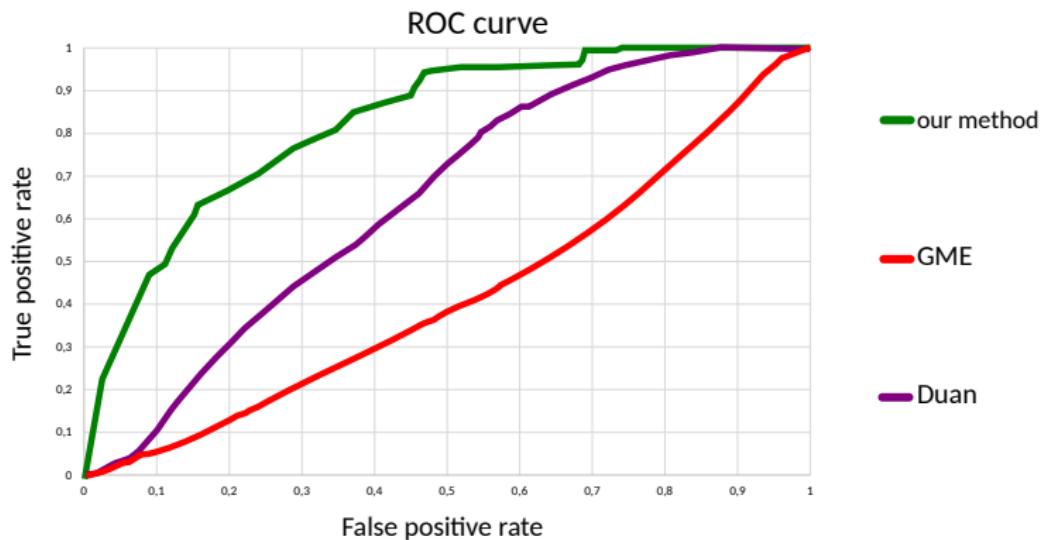


Evaluation metrics using scikit-learn

Documentation : https://scikit-learn.org/stable/modules/model_evaluation.html

```
from sklearn.metrics import f1_score, accuracy_score,  
    ConfusionMatrixDisplay  
  
yPredTest = clf.predict(xTest)  
  
# binary classification  
print("F1 score (test): ", f1_score(yTest, yPredTest))  
# multi-classes classification  
print("F1 score (test): ", f1_score(yTest, yPredTest,  
    average = 'micro'))  
  
print("accuracy (test): ", accuracy_score(yTest, yPredTest))  
ConfusionMatrixDisplay.from_predictions(yTest, y_predTest)
```

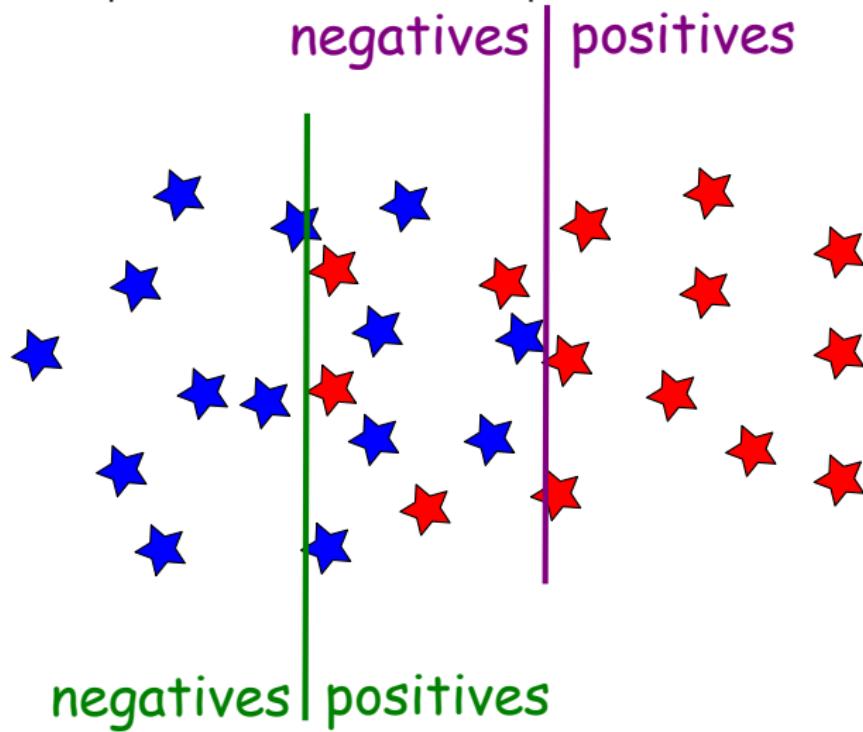
ROC curve



- ROC : Receiver Operating Characteristic
- FP rate = $\frac{FP}{FP+TN}$
- AUC : Area Under Curve
- Ressource <https://www.youtube.com/watch?v=4jRBRDbJemM>

How to compute a ROC curve ?

Example of a vertical linear separation :



How to plot a ROC curve ?

```
import scikitplot as skplt
import matplotlib.pyplot as plt

y_true = # ground truth labels
y_pred = # predicted probabilities generated by sklearn classifier
skplt.metrics.plot_roc_curve(y_true, y_pred)
plt.show()
```

- précision *accuracy* : fraction des prédictions correctes
- rappel *recall* : mêmes options que pour F1
- mesure F1 : plusieurs options sont disponibles, entre autres :
 - 'micro' : calcul global des VP, FP, FN
 - None : une valeur de score F1 est calculée pour chaque classe
 - classe considérée comme positive, les autres classes étant alors considérées comme négatives
 - 'macro' : calcul du score pour chaque classe et moyenne de ces scores
 - 'weighted' : calcul du score pour chaque classe et moyenne pondérée de ces scores en fonction de la taille des classes

- classes équilibrées : chaque (vraie) classe contient le même nombre de données
- classification binaire
 - classification aléatoire : 1 chance sur 2 d'être positif

Quelques ordres de grandeur : cas des classes équilibrées

- classes équilibrées : chaque (vraie) classe contient le même nombre de données
- classification binaire
 - classification aléatoire : 1 chance sur 2 d'être positif
 - $VP = FP = VN = VP = 0.5 = \text{toute métrique}$
- example de classification :

		classes estimées	
		positif	négatif
classes réelles	positif	80	20
	négatif	10	90

Quelques ordres de grandeur : cas des classes équilibrées

- classes équilibrées : chaque (vraie) classe contient le même nombre de données
- classification binaire
 - classification aléatoire : 1 chance sur 2 d'être positif
 - $VP = FP = VN = VP = 0.5 = \text{toute métrique}$
- example de classification :

		classes estimées	
		positif	négatif
classes réelles	positif	80	20
	négatif	10	90
- $VP = 80, VN = 90, FN = 20, FP = 10$

Quelques ordres de grandeur : cas des classes équilibrées

- classes équilibrées : chaque (vraie) classe contient le même nombre de données
- classification binaire
 - classification aléatoire : 1 chance sur 2 d'être positif
 - $VP = FP = VN = VP = 0.5 = \text{toute métrique}$
- example de classification :

		classes estimées	
		positif	négatif
classes réelles	positif	80	20
	négatif	10	90

 - $VP = 80, VN = 90, FN = 20, FP = 10$
 - rappel = 0.8, accuracy = 0.85, $F1 = 16/19 = 0.84$

- classification multi-classes

- example : cas de 3 classes équilibrées
- classification aléatoire : 1 chance sur 3 d'être de la classe 0

- classification multi-classes

- example : cas de 3 classes équilibrées
- classification aléatoire : 1 chance sur 3 d'être de la classe 0
 - accuracy = $1/3$, rappel(micro) = $30/90 = 1/3$, F1(micro) = $1/3$

- classification multi-classes

- example : cas de 3 classes équilibrées
- classification aléatoire : 1 chance sur 3 d'être de la classe 0
 - accuracy = $1/3$, rappel(micro) = $30/90 = 1/3$, F1(micro) = $1/3$
 - n classes équilibrées

- classification multi-classes

- example : cas de 3 classes équilibrées
- classification aléatoire : 1 chance sur 3 d'être de la classe 0
 - accuracy = $1/3$, rappel(micro) = $30/90 = 1/3$, F1(micro) = $1/3$
 - n classes équilibrées
 - accuracy = $1/n$, rappel(micro) = $1/n$, F1(micro) = $1/n$

Quelques ordres de grandeur : cas des classes équilibrées

- classification multi-classes

- example : cas de 3 classes équilibrées
- classification aléatoire : 1 chance sur 3 d'être de la classe 0
 - accuracy = $1/3$, rappel(micro) = $30/90 = 1/3$, F1(micro) = $1/3$
 - n classes équilibrées
 - accuracy = $1/n$, rappel(micro) = $1/n$, F1(micro) = $1/n$

- example de classification :

		classes estimées		
		classe 0	classe 1	classe 2
classes réelles	classe 0	70	10	20
	classe 1	10	90	0
	classe 2	5	15	80

Quelques ordres de grandeur : cas des classes déséquilibrées

- Vu que la classe 'George W Bush' est majoritaire, il peut être tentant de construire un classifieur qui répond toujours 'George W Bush'. Dans ce cas, quelles seraient les valeurs de chaque métrique ?

classe	numéro de classe	nombres d'images
Colin Powell	0	236
Donald Rumsfeld	1	121
George W Bush	2	530
Gerhard Schroeder	3	109
Tony Blair	4	144

Quelques ordres de grandeur : cas des classes déséquilibrées

- Vu que la classe 'George W Bush' est majoritaire, il peut être tentant de construire un classifieur qui répond toujours 'George W Bush'. Dans ce cas, quelles seraient les valeurs de chaque métrique ?

classe	numéro de classe	nombre d'images
Colin Powell	0	236
Donald Rumsfeld	1	121
George W Bush	2	530
Gerhard Schroeder	3	109
Tony Blair	4	144

- précision (*accuracy*) = $530/(236+121+530+109+144)=0.4649$
- rappel (*micro*) : $\frac{VP}{VP+FN}$
 - $VP = 530$ et $FP = (236+121+109+144) = FN$
 - '*micro*' : rappel = $530/(236+121+530+109+144) = F1 = 0.4649$
- None :
 - pour chaque classe sauf 'Bush' : $VP=0$, $FP=0$, $FN=\text{nb.images}$
 - pour la classe 'Bush' : $VP = 530$, $FP = (236+121+109+144)$, $FN = 0$ donc $\text{recall}=1$ et $F1=2*0.4649/(0.4649+1)=0.6347$
- '*macro*' : $0.6347/5 = 0.1269$
- '*weighted*' : $0.6347*530/(236+121+530+109+144)=0.2951$

Quelques ordres de grandeur (2)

- Et que donnerait un classifieur qui tire au hasard les classes ?
 - En réalisant 1000 tirages aléatoires des prédictions de chaque données et en calculant la moyenne des performances, on obtient :
 - F1 micro : 0.2005
 - F1 None : [0.2037 0.1389 0.2800 0.1300 0.1550]
 - F1 macro : 0.1815
 - F1 weighted : 0.2191
 - Peut-on expliquer ce résultat ? (en supposant que chaque donnée a autant de chances d'être classée dans chacune des classes)

Quelques ordres de grandeur (2)

- Et que donnerait un classifieur qui tire au hasard les classes ?
 - En réalisant 1000 tirages aléatoires des prédictions de chaque données et en calculant la moyenne des performances, on obtient :
 - F1 micro : 0.2005
 - F1 None : [0.2037 0.1389 0.2800 0.1300 0.1550]
 - F1 macro : 0.1815
 - F1 weighted : 0.2191
 - Peut-on expliquer ce résultat ? (en supposant que chaque donnée a autant de chances d'être classée dans chacune des classes)
 - Probabilité qu'une donnée soit classée en classe i : $1/5 = 0.2$ avec $0 \leq i \leq 4$
 - Calcul pour F1 None, classe 0 'Colin Powell' : $P = 236$; $N = 904$
 - Parmi les positifs, $1/5$ e seront classés positifs : $VP = 236 * 0.2$, $FN = 236 * 0.8$
 - Parmi les négatifs, $1/5$ e seront classés positifs : $FP = 904 * 0.2$, $VN = 904 * 0.8$
 - $F1 = 2VP / (2VP + FN + FP) = 236 * 0.4 / (236 * 0.4 + 236 * 0.8 + 904 * 0.2) = 0.2035$
- Pour le TP, vous utiliserez la valeur de F1 'micro' pour comparer les différentes approches.
 - Il faudra faire mieux que $F1('micro') = 0.4649$

Outline

1 Les algorithmes d'apprentissage

2 Préparation des données

3 Analyse des résultats

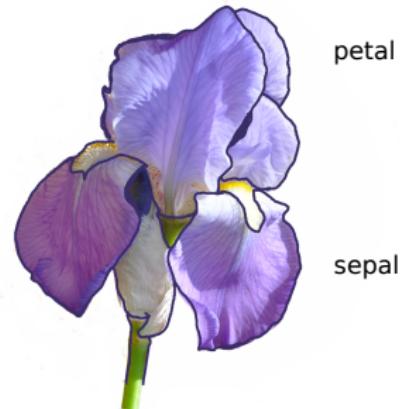
4 Des datasets pour 'jouer'

Determine species using Iris dataset



<https://archive.ics.uci.edu/ml/datasets/iris>

- 150 data
- 4d data :
 - sepal length and width in cm
 - petal length and width in cm
- classes :
 - Iris Setosa
 - Iris Versicolour
 - Iris Virginica



Charger le *dataset* Iris

```
from sklearn import datasets
iris = datasets.load_iris()
X = iris.data
y = iris.target
```

Classify handwritten digits using MNIST dataset

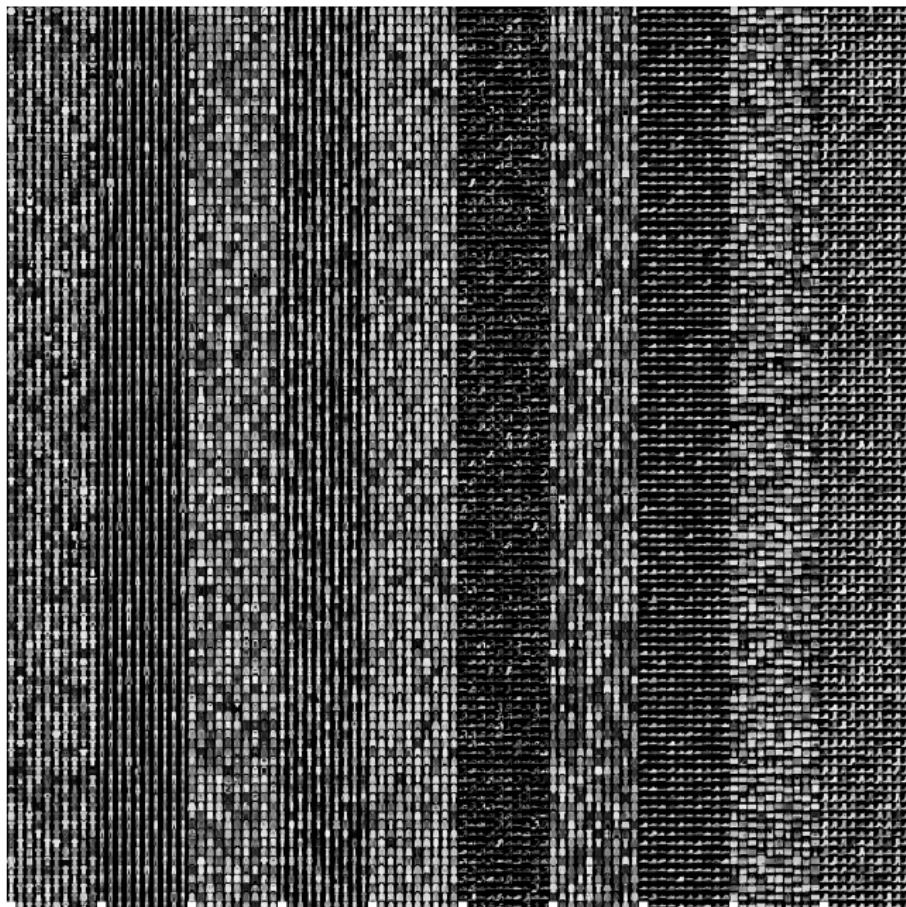
Dataset of 60000+10000 grayscale squared images (28x28).



Charger le *dataset* MNIST

```
from tensorflow.keras.datasets import mnist  
(x_train, y_train), (x_test, y_test) = mnist.load_data()
```

Fashion MNIST

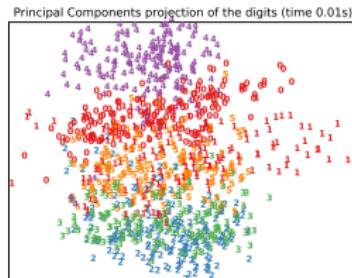


Dataset of
60000+10000
grayscale squared
images (28x28).

Label	Description
0	T-shirt/top
1	Trouser
2	Pullover
3	Dress
4	Coat
5	Sandal
6	Shirt
7	Sneaker
8	Bag
9	Ankle boot

Visualisation des données

- C'est un domaine de recherche
- Pour cette semaine :
 - données 2D



- On étudiera d'autres algorithmes les prochaines semaines : PCA et tSNE

```
# X data and y labels
colors = [cm.nipy_spectral(float(i) / nbCl) for i in range(nbCl)]
col = [colors[c] for c in y]
plt.figure(figsize=(10, 6))
plt.scatter(X[:, 0], X[:, 1], c=col, marker=".")
```

- tests /QCM au cours du module
- Rendus/démonstrations de TP
- Devoir sur table en fin de module

- mise en place de l'environnement de travail
 - python3, numpy, scikit-learn, notebook jupyter
- chargement du dataset Iris
 - découverte des données : nombre, dimensions
 - visualisation des données
 - Evaluation des métriques pour une classification aléatoire
 - séparation train / test : comparaison des métriques
 - classification binaire “à la main”
- mêmes choses avec les datasets MNIST ou fMNIST