Machine Learning in Python

Mahmoud Elsawy and Jean Luc Bouchot

Centre Inria d'Université Côte d'Azur

June 10, 2025

Outline

Qu'est-ce que le Machine Learning?

Définition

Le **machine learning** permet à un système d'apprendre à partir de données afin d'effectuer des prédictions sur de nouvelles données.

- Domaine de l'intelligence artificielle permettant aux systèmes d'apprendre de manière autonome.
- Utilisation d'un échantillon de données d'entraînement pour prédire de nouvelles données.
- Résolution de problèmes complexes inaccessibles à la programmation traditionnelle.

Applications du Machine Learning

Technologie et automatisation

- Véhicules autonomes
- Pilotage de drones
- Robotique

Prédiction et analyse

- Estimation des prix immobiliers
- Publicités personnalisées
- Systèmes de recommandation

Trois Types de Machine Learning

Comment une machine apprend?

Apprentissage supervisé

On fournit à la machine des exemples avec leurs réponses. Elle apprend à faire des prédictions correctes.

Apprentissage non supervisé

La machine découvre des groupes et des tendances toute seule, sans réponses données.

Apprentissage par renforcement

La machine apprend par essais et erreurs en recevant des récompenses ou des pénalités.

Apprentissage Supervisé : Idée

Comment ça fonctionne?

Étape 1 : On donne des exemples avec labels/étiquettes (ex : photos de chats et chiens).
Étape 2 : La machine apprend à reconnaître les différences.
Étape 3 : Une fois entraînée, elle peut prédire la sortie (ex: cette photo est celle d'un chien).

Applications

- Filtrage des emails (Spam vs Non-Spam).
- Prédiction du prix d'une maison.
- Reconnaissance vocale (ex : Siri, Alexa).

Apprentissage supervisé: les grandes familles

Selon les étiquettes on distingue plusieurs catégories:

- Les étiquettes sont des catégories non numériques. On parle alors de **classification** ou de **regression logistique**. Quels exemples peut on citer?
- Les étiquettes sont des valeurs continues. On parle alors de **regression**. Quels seraient des exemples?
- Les étiquettes sont des nombres ordonnées, mais ceux ci ne représentent pas des classes. On parle alors de **regression ordinale** ou **classification ordinale**. Quels exemples a-t-on?

Dans quelle catégorie se trouve la reconnaissance de nombres manuscrits à partir d'images?

Apprentissage Non Supervisé : Idée

Comment ça fonctionne?

Étape 1 : On donne beaucoup de données sans labels.
Étape 2 : La machine trouve des similitudes et regroupe les données.
Étape 3 : On analyse les groupes trouvés.

Applications

- Segmentation client (groupes d'acheteurs).
- Détection de fraudes bancaires.
- Organisation automatique des photos.

Apprentissage par Renforcement : Idée

Comment ça fonctionne?

Étape 1 : La machine explore différentes actions.
Étape 2 : Elle reçoit une récompense ou une pénalité selon l'action.
Étape 3 : Elle améliore sa stratégie avec le temps.

Applications

- Jeux vidéo (ex : AlphaGo).
- Robots autonomes (marche, vol).
- Trading algorithmique (optimisation des investissements).

Apprentissage Supervisé pour la Classification

Principe

L'**apprentissage supervisé** consiste à entraîner un modèle à partir d'exemples étiquetés, c'est-à-dire des données accompagnées de leur valeur réelle (on parle aussi de cible, de label, d'étiquette).

Exemple : Reconnaissance de Chiffres

- Chaque image représente un chiffre manuscrit (de 0 à 9).
- L'image est associée à une étiquette : le chiffre correct.
- Le modèle apprend à reconnaître les motifs visuels et à prédire la bonne classe.

Objectif

Généraliser les connaissances apprises pour classer correctement de *nouvelles images* jamais vues.

Qu'est-ce que l'Apprentissage Supervisé ?

Définition

L'apprentissage supervisé permet d'apprendre une fonction $f(x)$ qui associe une entrée x à une sortie y.

Objectif

Trouver une fonction f telle que :

$$f(x) \approx y$$

Cela signifie que les prédictions du modèle doivent être **aussi proches que possible** des vraies valeurs.

Exemple

- Entrée : Une image d'un chiffre manuscrit. - Sortie : Le chiffre correspondant (0-9). - Le modèle apprend une relation entre **image** et **chiffre**.

Les étapes d'un projet d'IA

Un projet passe par plusieurs phases de développement

- Cadrage: phase de préparation des coûts, besoins, ... Cette phase intervient avant le développement de la solution. – Non traitée ici
- Collection/laélisation des données – Non traitée ici.
- Exploration et préparation des données: cette étape importante force l'analyste à regarder les données manquantes et aberrantes et faire des suppressions ou corrections. Les données sont aussi normalisées pour éviter les différences de dynamique dans les valeurs. – Pas de correction de valeurs aberrantes / manquantes durant ce projet.
- Une étape de features (caractéristiques) engineering
- Une étape de modélisation pour le modèle d'apprentissage
- L'apprentissage et validation

Il est souvent indispensable de revenir en arrière (ajuster la détection d'*outliers*, revoir la normalisation, repenser les features ...)

L'objectif de ce module est de mener un projet d'IA de bout en bout, itérant les diverses étapes pour améliorer les résultats.

Étape 0 : Analyse et Exploration des données

Étape importante dans un projet, il est indispensable de faire un peu de visualisation de données

Représentation d'une Image

- Chaque image est une matrice de pixels dont les intensiés sont représentées par des entiers entre 0 et 16. - Exemple : Une image de chiffre manuscrit 8x8 pixels devient une matrice X de taille 8×8. - Chaque pixel a une intensité de niveau de gris: 0 pour noir et 16 pour blanc.

$$\longrightarrow X \begin{bmatrix} 0 & 45 & 200 & 255 \\ 10 & 150 & 230 & 100 \\ \vdots & \vdots & \vdots & \vdots \\ 50 & 240 & 210 & 0 \end{bmatrix}$$

Étape 0: Analyse et prération des données

L'exploration, c'est aussi le moment de comprendre la distribution de l'échantillon.
TODO: Commencer un script chargeant les données depuis `scikit learn, dataset` et utiliser `matplotlib` pour afficher une image aléatoire par classe. Ajouter avec des informations (et visualisations) sur la dimensionalité des données, la distribution des différentes classes.

Étape 1 : Visualisation et préparation des données pour le Modèle

On cherche maintenant à visualiser non plus les données uniques (l'image d'un chiffre), mais la base de données dans son ensemble.

Normalisation pour le Modèle

- L'image est normalisée pour simplifier le traitement.
- Ex : On divise l'intensié de chaque pixel par 255.

$$X_{\text{normalisé}} = \begin{bmatrix} 0 & 0.18 & 0.78 & \dots & 1 \\ 0.04 & 0.59 & 0.90 & \dots & 0.39 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 0.19 & 0.94 & 0.82 & \dots & 0 \end{bmatrix}$$

TODO: Convertir les données en données normalisées. Reflechir à la normalisation utile parmi les choix disponibles dans `sklearn`.

Étape 1 : Visualisation de la base de données

La base de données ne peut être visualisée facilement quand dans 2 ou 3 dimensions. On procède donc à une rérudction de dimensions.

Réduction de dimension avec une ACP (PCA) pour la visualisation

- Concentre l'information globale des pixels.
- Transformation de l'image $8 \times 8 = 64$ dimensions en un espace à 2 ou 3 dimensions.
- Conserve uniquement les **composantes principales** les plus pertinentes.

TODO: Continuer le script en ajoutant une ACP à 2 ou 3 dimensions et afficher l'ensemble des données avec une couleur par label. Faire a minima un visualisation 2D. Comparer une image originale et une image reconstruite après une réduction de dimension. Comment évolue la qualité de la reconstruction après une ACP en changeant le nombre de composantes principales? Comment quantifer cette qualité par image? De manière globale?

Étape 2 : Extraction des Caractéristiques

Pourquoi est-ce nécessaire ?

- Une image brute est trop complexe pour être directement analysée.
- Le modèle identifie des motifs simples

Exemple

- Un "5", peut être décrit comme:
  - Une ligne horizontale en haut
  - Une ligne vertical à gauche
  - Une courbe ouverte à gauche sur la moité inférieure
  - Zones sombres et claires qui définissent la forme générale du chiffre.

L'objectif de la phase de *feature engineering* est de trouver comment transformer l'information des pixels en caratéristiques utiles et intelligentes pour le modèle.

Extraction des Caractéristiques avec un filtre de convolution W

Pourquoi un Filtre ?

- Le modèle doit identifier les motifs essentiels du chiffre "5".
- Nous appliquons un filtre W qui détecte les bords et courbes.
- Une convolution permet un calcul de caractéristique localisée

Un exemple de filtre W pour Identifier "5"

$$W = \begin{bmatrix} -1 & -1 & -1 & -1 & -1 \\ 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix}$$

- Met en lumière de grosses différences d'intensités verticales
- S'étend et agglomère ces différences le long d'un axe horizontal
- Met en lumière des lignes/contours horizontales

Application du Filtre W sur la Matrice X

Multiplication Matricielle

- Nous appliquons le filtre W sur la matrice X.
- Cette opération met en évidence lune spécificité du chiffre "5" (en l'occurrence une ligne horizontale).

Résultat après Filtrage

$$F = X \circledast W$$

$$F = \begin{bmatrix} 0.9 & 0.2 & 0.0 & \dots & 0.0 \\ 0.1 & 0.8 & 0.5 & \dots & 0.2 \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 0.0 & 0.3 & 0.9 & \dots & 1.0 \end{bmatrix}$$

- Cette nouvelle matrice F contient l'intensité de la réponse de l'image au filtre W à une certaine location.

Extraction des caractéristiques pour la reconnaissance des chiffres

1- Réduction de dimension avec PCA

- Réduire le nombre de pixels tout en préservant l'essence du chiffre.
- Transformer l'espace de dimension 64 en un espace de faible dimension.
- Conserver les **composantes principales** les plus pertinentes.
- Diminuer la complexité et améliore la classification.
- Question: Comment choisir la dimensionalité de la projection?

2- Partitionnement en zones

- Diviser l'image en plusieurs régions pour mieux analyser la structure.
- Découpage en **trois zones** : Haut (lignes 1,2,3), Milieu (4,5), Bas (6,7,8).
- Calcul de la **moyenne des intensités** par zone.
- Capture les éléments distinctifs, comme les boucles ou les barres

Extraction des caractéristiques pour la reconnaissance des chiffres

3- Détection des contours avec (la moyenne d')un filtre de Sobel

- Identifier les contours des chiffres pour distinguer les formes.
- Calcul du gradient d'intensité pour repérer les changements brusques.
- Met en évidence les structures principales du chiffre.
- Améliore la différenciation entre les chiffres similaires (ex: "8" et "6").

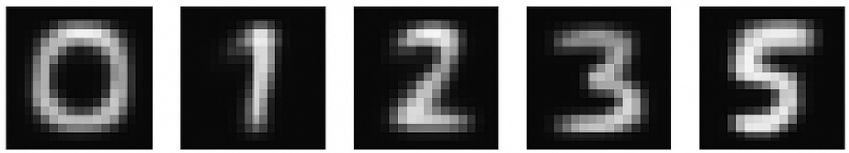
4- Pourquoi ces méthodes ?

- Réduction du nombre de caractéristiques tout en préservant l'essence des chiffres.
- Amélioration de la robustesse de la classification.
- Réduction du bruit et des redondances dans les données.

Les Étapes Réalisées Jusqu'à Présent

Votre premier script devrait vous avoir donné un compréension de l'exploration et la visualisation des données, incluant

- Collection/chargement des images des chiffres 0 à 9.
- Normalisation des données.
- Exploration et visualisation de la base de données
- Calculs de trois *features*:
  - une réduction de dimensions par ACP (combien de composantes?)
  - des composantes local de mesure d'intensités
  - une mesure de l'intensité globale des gradients
- Création d'une nouvelle matrice des caractéristiques F par concaténation des features.



Feature 0 Feature 1 Feature 2 Feature 3 Feature 4

Division des données pour l'apprentissage

- **Pourquoi diviser les données ?**
  - Évite le sur-apprentissage (*overfitting*).
  - Permet de tester le modèle sur des données inédites.
  - Améliore la généralisation du modèle.
- **Types de division des données :**
  - **Entraînement/Test** : - 80% pour l'entraînement, 20% pour le test.
  - **Validation croisée (Cross-Validation)** : découpage en plusieurs sous-ensembles pour tester plusieurs configurations.
  - **Stratification** : assure une distribution équilibrée des classes dans les ensembles d'apprentissage et de test.
- **Impact sur l'apprentissage :**
  - Un mauvais découpage peut fausser les résultats.
  - Une proportion trop faible d'exemples de test peut limiter la fiabilité de l'évaluation.
- Trois ensembles:
  - Ensemble d'apprentissage (training): optimization des paramètres
  - Ensemble de validation: hyperparameter tuning
  - Ensemble de test: évaluation d'un modèle

Méthodes de classification avec SVM

- **Support Vector Machine (SVC)**
  - Trouve une frontière optimale pour séparer les classes.
  - Utilise des **kernels** pour gérer des distributions complexes.
- **Validation croisée (Cross-Validation)**
  - Utilisation de **K-Fold** : divise les données en K sous-ensembles et entraîne/teste plusieurs fois.
  - Permet de mieux évaluer la performance réelle du modèle.
- **Stratification**
  - Assure une **répartition équilibrée des classes**.
  - Utile pour éviter les biais dus à un déséquilibre des données.
- **One-vs-One vs One-vs-Rest**
  - **One-vs-One (OvO)** : entraîne un classificateur pour chaque paire de classes.
  - **One-vs-Rest (OvR)** : entraîne un modèle pour chaque classe contre toutes les autres.
  - Le choix dépend de la taille et de la complexité du jeu de données.

Implémentation

Pour le second script, vous devez:

- Charger les données bruts,
- Implémenter une première division train/test,
- Visualiser les distributions des classes dans chacun des ensembles, les comparer
- Mettre en place des fonctions utiles à une `Pipeline` de traitement des données
- Mettre en place un simple SVClassifier avec un noyau linéaire, optimiser les paramètres, et rapporter l'erreur de test
- Consider un SVC plus complexe avec divers (hyper)paramètres à optimiser et mettre en place un validation croisée.
- Commenter sur les temps de calculs et fiabilité des diverses stratégies d'apprentissage (OvO, OvR)
- Si le temps permet, tester d'autres modèles d'apprentissage. Lesquels? Pourquoi? Quels paramètres optimiser?