# PROSIT 3

Intelligence Artificielle

MARDI 7 MARS 2023

CESI
Mathilde Ballouhey

# Table des matières

**TABLE DES MATIERES** ..... **1**

**1. CONTEXTE** ..... **2**

**2. MOTS CLES** ..... **2**

2.1. LE PERCEPTRON ..... 2
2.2. REGRESSION LOGISTIQUE ..... 2

**3. BESOINS / PROBLEMATIQUES** ..... **2**

**4. CONTRAINTES** ..... **2**

**5. LIVRABLES** ..... **2**

**6. GENERALISATION** ..... **2**

**7. HYPOTHESES** ..... **2**

**8. PLAN D'ACTION** ..... **2**

**COURS** ..... **4**

**9. CLASSIFICATION** ..... **4**

9.1. DIFFERENCE CLASSIFICATION ET REGRESSION ..... 4
9.2. QU'EST-CE QUE LA CLASSIFICATION ..... 4
9.3. K-PLUS PROCHES VOISINS ..... 4
9.4. CLASSIFIEURS LINEAIRES ..... 4
9.5. ALGORITHMES ..... 5
*9.5.1. Méthode des centroïdes* ..... *5*
*9.5.2. Perceptron* ..... *5*
*9.5.3. Classifieurs à vaste marge* ..... *6*
*9.5.4. Classifieurs à soft marge (Soft-Margin SVM)* ..... *6*
*9.5.5. Arbre de décision* ..... *7*

**10. METRIQUE** ..... **7**

10.1. MATRICE DE CONFUSION ..... 7
10.2. COURBE ROC & AUC (AIRE SOUS LA COURBE) ..... 8

**11. LIVRABLE** ..... **9**

***12.*** **VALIDATION DES HYPOTHESES** ..... **9**

**BIBLIOGRAPHIE** ..... **10**

# 1. Contexte

Une entreprise immobilière veut faire des projections financières, et elle veut savoir si un bien sera vendu dans les 6 mois en fonction du quartier (block).

# 2. Mots clés

## 2.1. Le perceptron

Le perceptron est un type de modèle d'apprentissage automatique supervisé qui appartient à la famille des réseaux de neurones artificiels. Il est utilisé pour résoudre des problèmes de classification binaire, c'est-à-dire des problèmes où les données sont classées en deux catégories distinctes.

## 2.2. Régression logistique

La régression logistique est une technique d'apprentissage automatique supervisé utilisée pour prédire une variable de sortie catégorielle à partir d'un ensemble de variables d'entrée. Contrairement à la régression linéaire qui est utilisée pour prédire des variables continues, la régression logistique est utilisée pour prédire des variables catégorielles.

# 3. Besoins / Problématiques

- Comment réalisé un modèle de classification sur la vente ou non sur une période de 6 mois dans un quartier donné ?

# 4. Contraintes

- S'appuyer sur des indicateurs de qualité pertinents
- Un attribut booléen nommé *Sold6M*
- Régression logistique
- Perceptron
- Python ❤️

# 5. Livrables

- Notebook
  - Étude pour l'utilisation de la régression logistique et/ou le Perceptron
  - Model implémenté s'appuyant sur des indicateurs qualités pertinentes

# 6. Généralisation

Utilisation de la classification avec l'IA.

# 7. Hypothèses

- La régression logistique donne des entiers naturels ?
- Le Perceptron est un algorithme supervisé ?
- La régression linéaire ne suffit pas car elle est en sous apprentissage ?
- Il faut utiliser des métriques pour valider le model ? RMSE, MAE, matrice de confusion

# 8. Plan d'action

- Mots clés
- Cours
  - Classification (voir la différence avec la régression)
    - Perceptron

        - Régression logistique
        - …
    - Métrique
        - Courbe ROC
        - AUC
        - Matrice de confusion
        - Et plus encore !! (👉ﾟヮﾟ)👉

# Cours

## 9. Classification

### 9.1. Différence Classification et Régression

**Classification**

- Point un point x on cherche sa classe
- Souvent binaire
- Donner une prédiction à une échelle discrète (appartient à l'ensemble N nombre entier)

**Régression**

- Pour un point donné, prédit une valeur numérique
- Donne une prédiction quantitative
- Souvent une échelle continue

### 9.2. Qu'est-ce que la classification

La tâche qui permet d'assigner à un point en d-dimensions (nombre de caractéristiques) un nombre discret (appelé classe) qui est en régression un nombre continu.

Il existe 2 types de modèles :

- Modèle Génératifs : Basé sur les probabilités. Exemple : x sachant y
- Modèle Discriminatifs : prennent un *dataset*, essayer d'estimer les poids et essayer d'apprendre les frontières de décisions. (Quelle est la courbe qui sépare mon nuage de point)

### 9.3. K-Plus proches Voisins

Le K-Plus Proches Voisins (K-NN) peut être utilisé pour la classification et la régression. Il est interprétable et sa complexité est O(1).


*Figure 1 : Entrainement et Test d'erreur d'une fonction de k*

### 9.4. Classifieurs linéaires

Les classifieurs linéaires représentent les points dans un espace à d-dimensions, essaient de tracer une courbe qui va séparer les deux classes et utilisent les courbes pour faire des prédictions.

Ils ont une complexité en $O(k)$ et sont toujours une droite ou un hyperplan de dimension $(d-1)$. Ils utilisent souvent une fonction de décision qui est elle-même linéaire, de type $ax+b$ ou $\omega x+\alpha$.

On va :

- Représenter les points dans un espace à d-dimentions
- Essayer de tracer une courbe qui va séparer les 2 classes
- Utiliser les courbes pour faire des prédictions

## 9.5. Algorithmes

### 9.5.1. Méthode des centroïdes

- Calculer la moyenne de tous les points dans la classe C
- On calcule le centroïde (le centre de gravité) du nuage de points de chaque classe noté $\mu_C$ et $\mu_x$
- On dessine un segment entre $\mu_C$ et $\mu_x$
- On cherche la droite qui passe au milieu du segment.

Un segment (2d) ou un hyperplan si la classe est en 3 dimensions.

⚠ Si les points sont dispersés la droite aura généralement beaucoup d'erreurs. Mais elle généralise assez bien. Le *biais* est assez grand, la *variance* est assez faible.

### 9.5.2. Perceptron

Utilisé dans le réseau de neurone donc la base est le nœud qui s'appelle *Perceptron*. En connectant et en imbriquant plusieurs *Perceptrons* ça donne ce qu'on appelle **Perceptron multicouche** (ou réseau de neurones feed-forward).

Le *Perceptron* a été inventé par *Frank Rosenblatt* au Cornell Aeronautical Laboratory. Il était initialement conçu pour reconnaitre des images de 20x20 pixel. La première machine s'appelait le **Mark I Perceptron**.

**Les aspects intéressants**

- C'est un **algorithme en ligne** : on peut l'alimenter en données pendant l'exécution pour qu'il s'améliore pendant l'exécution.
- Il dispose d'un **théorème de convergence** : si les données sont linéairement séparables, indépendantes et identiquement distribuées, alors on va certainement converger vers un classifieur avec un minimum d'erreur.

Il est **lent** car il utilise un algorithme d'optimisation, la **descente de gradient**, pour trouver des droites correctes pour séparer les points.

Il prend un **échantillon de n-points** de $X_1 \dots X_n$. Pour chaque point on a le label :

$$y_i = \begin{cases} 1 \; si \; X_i \in class \; C \\ -1 \; sinon \end{cases}$$

Le **but** : trouver les **poids ω** tel que :

$$X\_i \,.\, \omega \geq 0 \; \{si \;\; y\_i \;=\; 1$$

$$X_i . \omega \leq 0 \; si \; y_i = -1$$

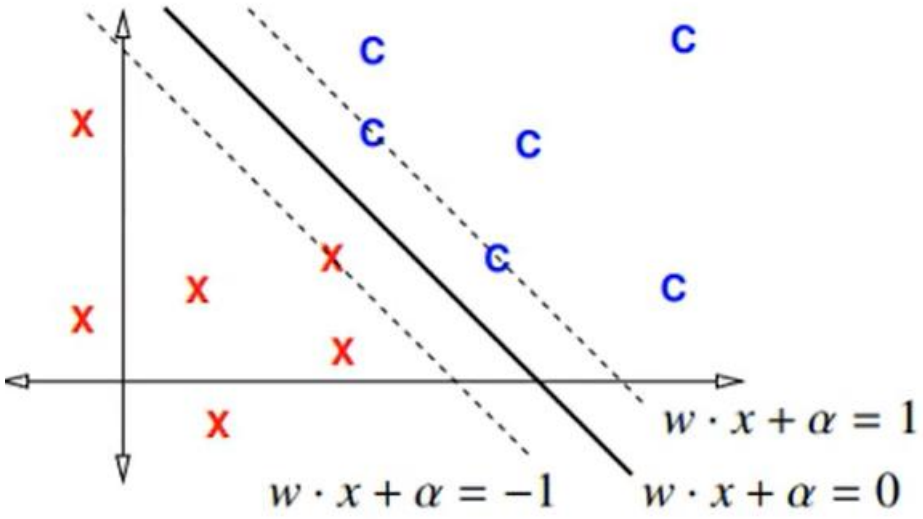
Soit trouver ω en **minimisant la fonction coût** : $R(\omega) = \sum_{i \in V} - y_i X_i . \omega$

La descente de gradient est ensuite utilisée et bouclée jusqu'à ce que la fonction coût ne varie plus. La fonction de **descente de gradient** peut être **remplacé** par la **descente de gradient stochastique** qui est un meilleur algorithme.

### 9.5.3. Classifieurs à vaste marge

La **marge** est la distance entre la ligne de décision (définie par le vecteur de poids ω et l'intercept α) et le point le plus proche de chaque classe. Cela ne fonctionne que si les **données** sont **linéairement séparables**. On utilise un **programme ou problème quadratique** pour trouver une **solution unique** qui **maximise la marge** tout en **minimisant** la taille du vecteur de **poids** $||\omega||$.

Pour garantir que chaque point est correctement classé, on ajoute une nouvelle contrainte : $y_i(\omega . X_i + \alpha) \geq 1 \; pour \; i \in [1, n]$




### 9.5.4. Classifieurs à soft marge (Soft-Margin SVM)

Les classifieurs à soft marge permettent de répondre au problème des classifieurs à vaste marge lorsque les données ne sont pas linéairement séparables ou lorsque des points aberrants (*outliers*) sont présents.

Ils permettent à quelques points de violer la marge en introduisant des **variables d'écart** (slack variables) $\xi_i$. Cela donne la nouvelle contrainte :

$$y_i(X_i . \omega + \alpha) \geq 1 - \xi_i$$

On impose la contrainte :

$$\xi_i \geq 0$$

On utilise la descente du gradient pour résoudre ce problème. On choisit la **valeur de C** (qui **contrôle la balance entre la marge et l'erreur d'entraînement**) en utilisant la *cross-validation*.

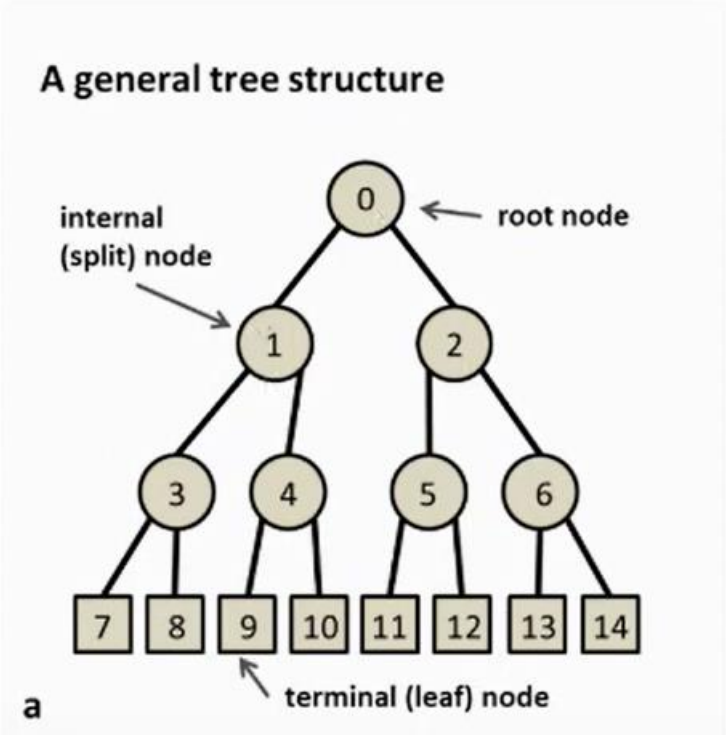
| | Small C | Big C |
|---|---|---|
| Goal | Maximize margin $^1/_{\|\|\omega\|\|}$ | Keep most slack variables zero or small |
| Danger | Underfitting | Overfitting |
| Outliers | Less sensitive | Very sensitive |
| Boundary | More « flat » | More sinuous |

*Figure 2 : Régularisation de l'hyperparamètre*

Lorsque les données ne sont pas linéairement séparables, on peut utiliser une transformation non linéaire pour projeter les points sur une dimension supérieure, puis revenir à la dimension initiale. Cela permet de trouver une frontière de décision non linéaire.

### 9.5.5. Arbre de décision

Les arbres de décision sont une méthode d'apprentissage supervisé qui divise le problème en sous-problèmes plus simples. Chaque nœud de l'arbre représente une caractéristique (ou attribut) et chaque branche représente une valeur possible de cette caractéristique.




**Comment choisir les nœuds**

- On choisit aléatoirement un attribut A
- On calcule l'**Entropie Estimée** $EH(A)$ pour cette caractéristique. L'entropie estimée mesure la **quantité d'incertitude** dans les données **après avoir divisé** selon cette caractéristique.

$$EH(A) = \sum_{i=1}^{K} \frac{p_i + n_i}{p + n} H\left(\frac{p_i}{p_n + n_i}, \frac{n_i}{p_i + n_i}\right)$$

- On calcule également le **gain d'information** I(A), qui mesure la **réduction de l'entropie** que l'on obtient en divisant selon cette caractéristique.

$$I(A) = H\left(\frac{p}{p+n}, \frac{n}{p+n}\right) - EH(A)$$

- On choisit l'attribut A avec le plus grand I

# 10. Métrique

## 10.1. Matrice de confusion

La **matrice de confusion** est une méthode permettant d'évaluer la **performance** d'un algorithme de **classification** en comparant les prédictions faites par l'algorithme avec les véritables étiquettes des données. Elle permet de visualiser le nombre de **vrais positifs**, de **faux positifs**, de **vrais négatifs** et de **faux négatifs**.

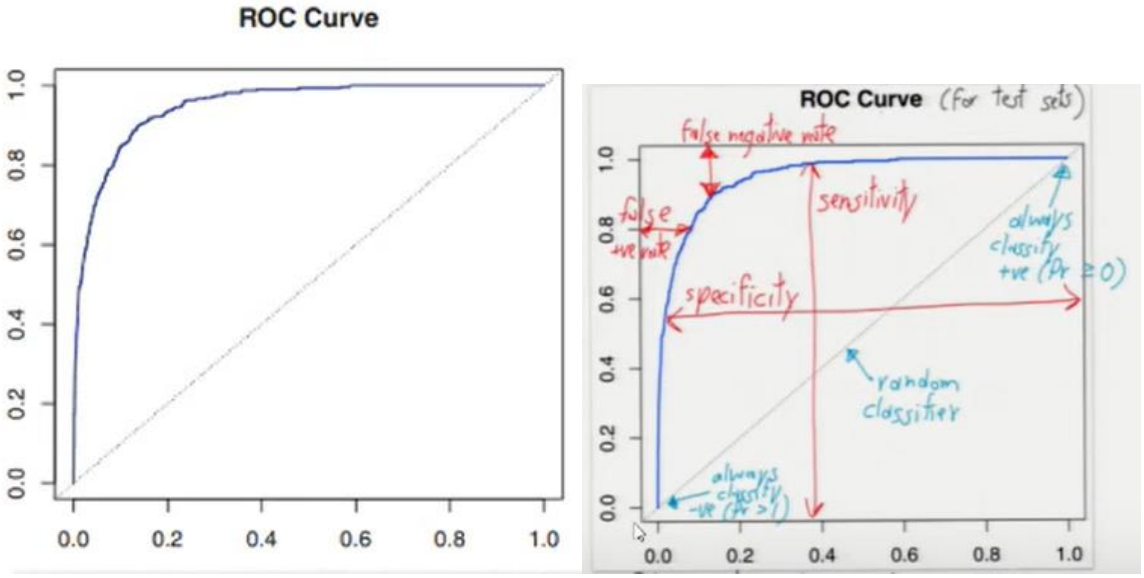
The loss used for model fitting is typically not the best evaluation metric.

| | | Truth: 1 | Truth: 0 |
|---|---|---|---|
| Prediction | 1 | True positive (TP) | False positive (FP) |
| | 0 | False negative (FN) | True negative (TN) |

| | | |
|---|---|---|
| Accuracy: | (TP + TN) / n | The most obvious/common evaluation metric |
| Error rate: | (FP + FN) / n | |
| Precision: | TP / (TP + FP) | Used when detecting rare outcomes |
| Recall: | TP / (TP + FN) | |

## 10.2. Courbe ROC & AUC (Aire sous la courbe)

La courbe **ROC (*Receiver Operating Characteristic*)** est une méthode **d'évaluation des classifieurs** qui compare le taux de faux positifs et le taux de vrais positifs pour différents seuils de décision. Avec *x* le taux de faux positifs et *y* taux de vrais positifs. Elle est générée en **exécutant le classifieur** sur les données de **test** et de **validation**. Plus la courbe ROC est proche du **coin supérieur gauche**, mieux le **classifieur binaire fonctionne**.




L'aire sous la courbe ROC (AUC) donne une mesure de la qualité globale du classifieur, avec une valeur proche de 1, indiquant une très bonne performance et une valeur proche de 0.5 indiquant une performance aléatoire.

La courbe ROC et l'AUC sont des outils utiles pour évaluer la performance d'un classifieur binaire, mais il est important de garder à l'esprit que ces **métriques ne sont pas toujours suffisantes** pour évaluer la performance d'un modèle dans des situations réelles. Il est souvent nécessaire de considérer **d'autres facteurs** tels que les **coûts de classification erronée** pour chaque classe, la **distribution des classes** dans les données de test, ou la **capacité du modèle à généraliser** à de nouvelles données.

11. Livrable

*12.* Validation des Hypothèses

# Bibliographie

**Aucune source spécifiée dans le document actif.**