Machine Learning Supervisé en R

Niveau Master 2

Mehdi MUNIM, mehdi.munim.int@groupe-gema.com

Plan du cours

Jour 1 : Modèles de régression

- régression simple / multiple (caret)
- régularisation ridge / lasso / elastic net
- TP 1 & 2 (mtcars, AmesHousing)

Jour 2 : Modèles de classification

- arbres de décision, xgboost...
- TP 3 (Titanic)
- Exposé (application ML supervisé)

Jour 3 : Projet

- Projet Scoring
- Présentation orale

Objectifs du cours (cf. IA School)

Machine Learning supervisé avec R (M2-DA)

Niveau: M2-DA

Rythme: Tronc commun

Format de l'évaluation

- *50% contrôle continu (format apprécié: projet)*
- *50% partiel*

Objectifs pédagogiques

- Connaître les bibliothèques R pour le machine learning: *caret, tidyverse, randomForest, etc.*
- Installer un environnement de travail et explorer des données simples avec **ggplot2** et **dplyr**
- Préparer des données pour du machine learning:
  - nettoyage (valeurs manquantes, outliers, variables catégorielles)
  - normalisation
  - standardisation
  - transformation des variables
- Avec les bibliothèques **glmnet** et **caret**:
  - Réaliser une régression linéaire
  - Réaliser une régression multiple
  - Maîtriser les techniques de régularisation pour éviter l'overfitting
    - Régression Ridge
    - Régression Lasso
- Réaliser une régression logistique pour résoudre un problème de classification
  - arbre de décision avec **rpart**
- Maîtriser les algorithmes d'ensemble avec R
  - Utiliser la bibliothèque **randomForest** pour implémenter un modèle
  - Utiliser **xgboost** pour implémenter un modèle Gradient Boosting

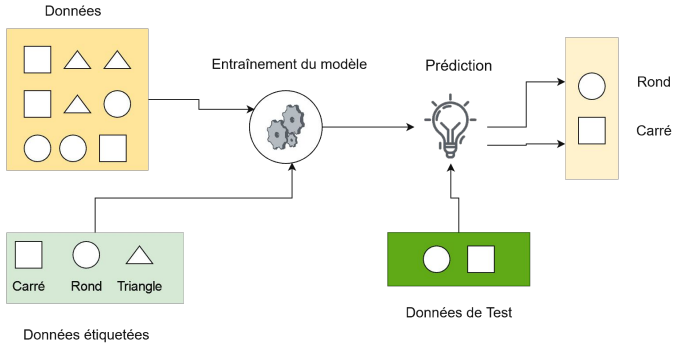
Machine Learning Supervisé

JOUR 1

2 aspects :

- Machine Learning :
  - Un algorithme apprend à partir de données au lieu d'être explicitement programmé

- ML **supervisé** : apprendre à partir de **données labellisées** (data + labels)



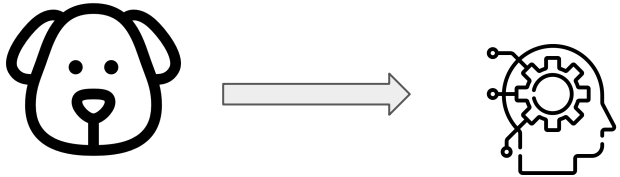
Machine Learning Supervisé

Chats

Chiens

1) Apprentissage

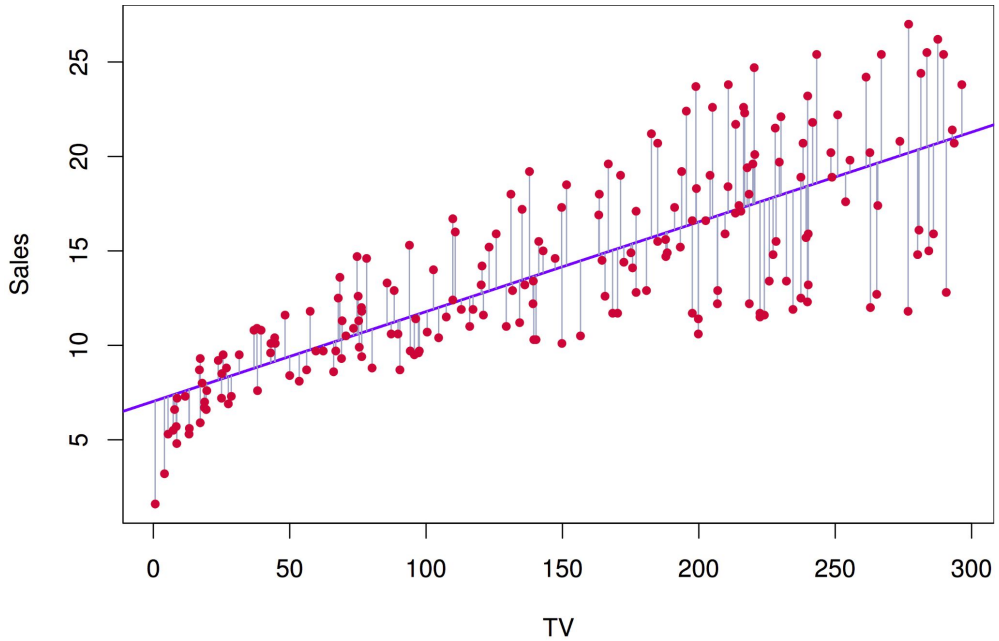
2) Inférence



Chien

Types de problèmes : régression vs classification

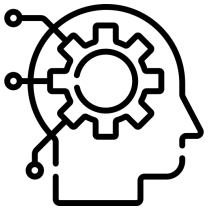
Régression : Prédire une variable continue (ex: prix, température, demande).

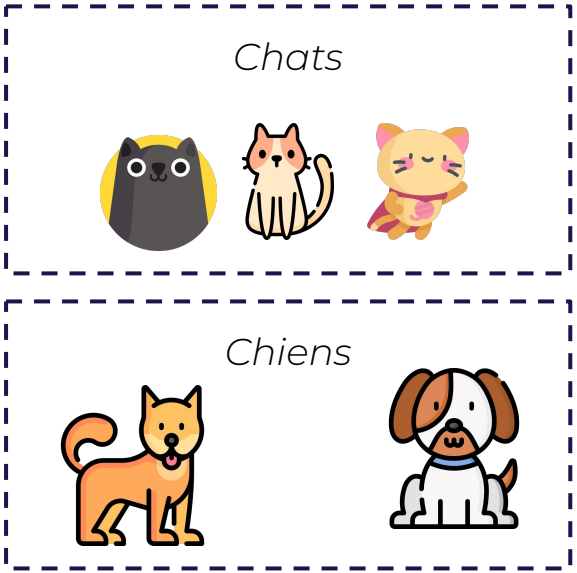


Types de problèmes : régression vs classification

Classification : Prédire une variable catégorielle (ex: classe, type, étiquette).

Distinction importante pour le choix des algorithmes et des métriques d'évaluation.





Régression ou classification ?

A. Classer des emails en spam ou non spam

B. Prédire le prix d'une maison en fonction de sa superficie

C. Identifier des images de chats et de chiens.

D. Prévoir la demande d'un produit en fonction du prix.

E. Diagnostiquer des maladies à partir d'images médicales.

Régression ou classification ?

A. Classer des emails en spam ou non spam **(Classification)**

B. Prédire le prix d'une maison en fonction de sa superficie **(Régression)**

C. Identifier des images de chats et de chiens. **(Classification)**

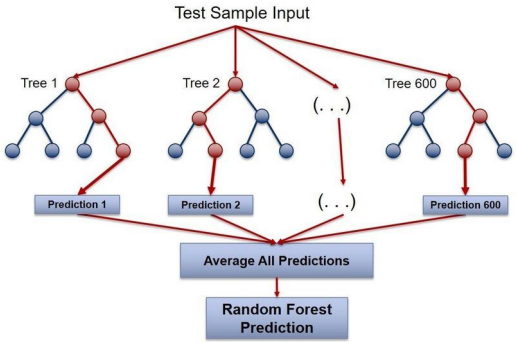
D. Prévoir la demande d'un produit en fonction du prix. **(Régression)**

E. Diagnostiquer des maladies à partir d'images médicales. **(Classification)**

R pour le Machine Learning

Plusieurs packages R pour le Machine learning :

- **caret**
- **tidyverse**
- **randomForest**
- **xgboost**
- ....



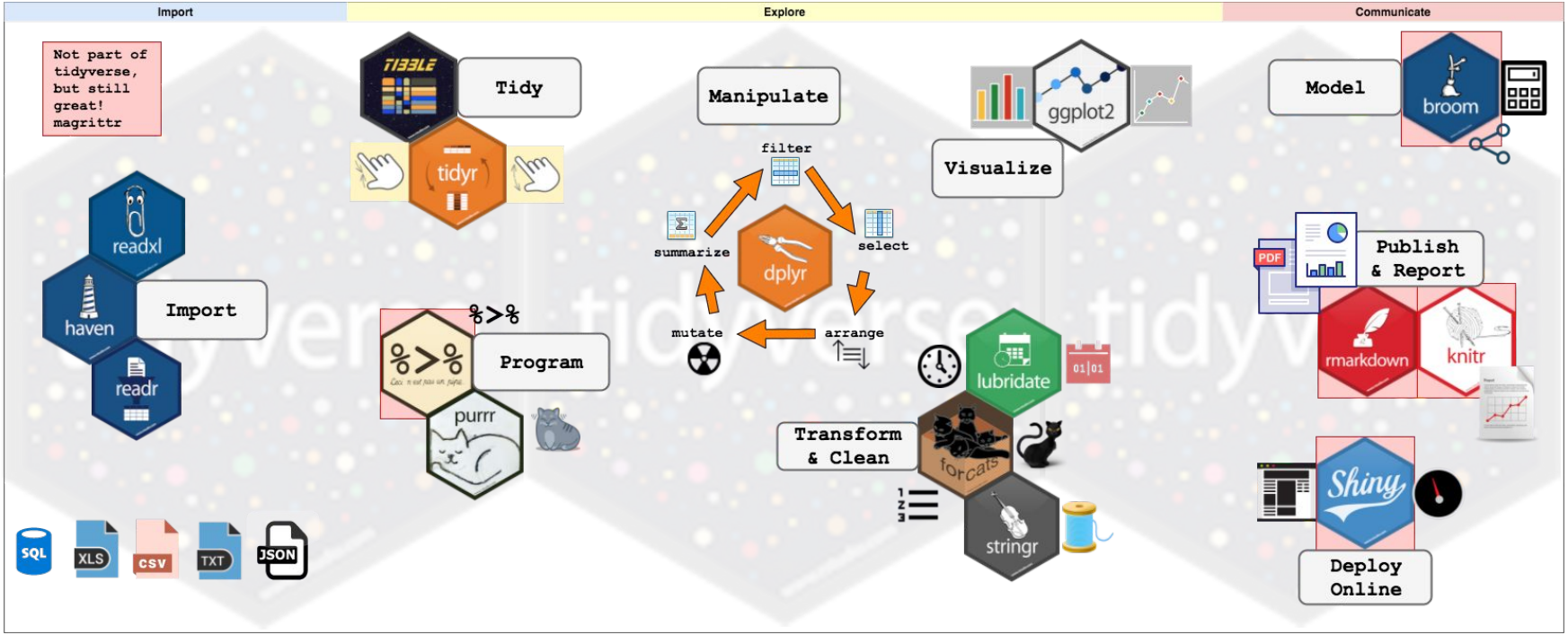
Focus sur tidyverse (dplyr et ggplot2)

- tidyverse est une **collection** de packages R pour la manipulation, l'exploration et la visualisation de données.

- dplyr : permet de manipuler facilement les données (filtrer, sélectionner, transformer).

- ggplot2 : permet de créer des graphiques de haute qualité.

Focus sur tidyverse (dplyr et ggplot2)



Focus sur caret

- caret (*Classification And REgression Training*) est un package incontournable pour le Machine Learning supervisé en R.
- Fournit une interface unifiée pour de nombreux algorithmes de Machine Learning.
- Simplifie l'entraînement, l'évaluation et la comparaison des modèles.
- Fonctions clés : **train()**, **predict()**, **confusionMatrix()**.

Autres packages utiles

randomForest : implémente l'algorithme Random Forest.

xgboost : implémente l'algorithme Gradient Boosting.

rpart : permet de construire des arbres de décision.

glmnet : permet d'entraîner des modèles de régression linéaire, logistique et régularisés.

Source :
https://r-craft.org/the-3-reasons-you-should-learn-r-for-data-science/

A QUICK AND DIRTY COMPARISON OF R AND PYTHON AS DATA SCIENCE LANGUAGES

Feature	R	Python
Ease of doing data manipulation	5	4
Ease of doing data visualization	5	3
Ease of doing data analysis	5	3
Ease of doing machine learning	3	4
Ease of doing general programming & automation	2	5
Size of Data Science Community	2	5
Number of jobs that require the language	2	5

Introduction aux modèles de régression

Rappel : la régression sert à prédire une variable continue.

Différents types de modèles de régression : linéaire, polynomiale, non linéaire.

Focus sur la régression linéaire simple et multiple.

Avant de commencer… la préparation du dataset

Préparation en vue du Machine Learning

Nettoyage des données :

- Gérer les valeurs manquantes (NA)
- Gérer les valeurs aberrantes (outliers).

Transformation des données :

- Mettre les variables à la même échelle (normalisation, standardisation).
- Encoder les variables catégorielles (variables indicatrices, encodage numérique).

Sélection des variables :

- Choisir les variables les plus pertinentes pour le modèle.
- Supprimer les variables redondantes ou non informatives.

Gestion des valeurs manquantes

Techniques courantes :

- Supprimer les lignes ou les colonnes avec des valeurs manquantes (na.omit()).
- Remplacer les valeurs manquantes par une valeur estimée (moyenne, médiane, mode) (ifelse(), imputeTS::na_mean()).
- Utiliser des méthodes d'imputation plus avancées (kNN, régression).

⇒ Choisir la méthode la plus adaptée en fonction du type de données et du problème.

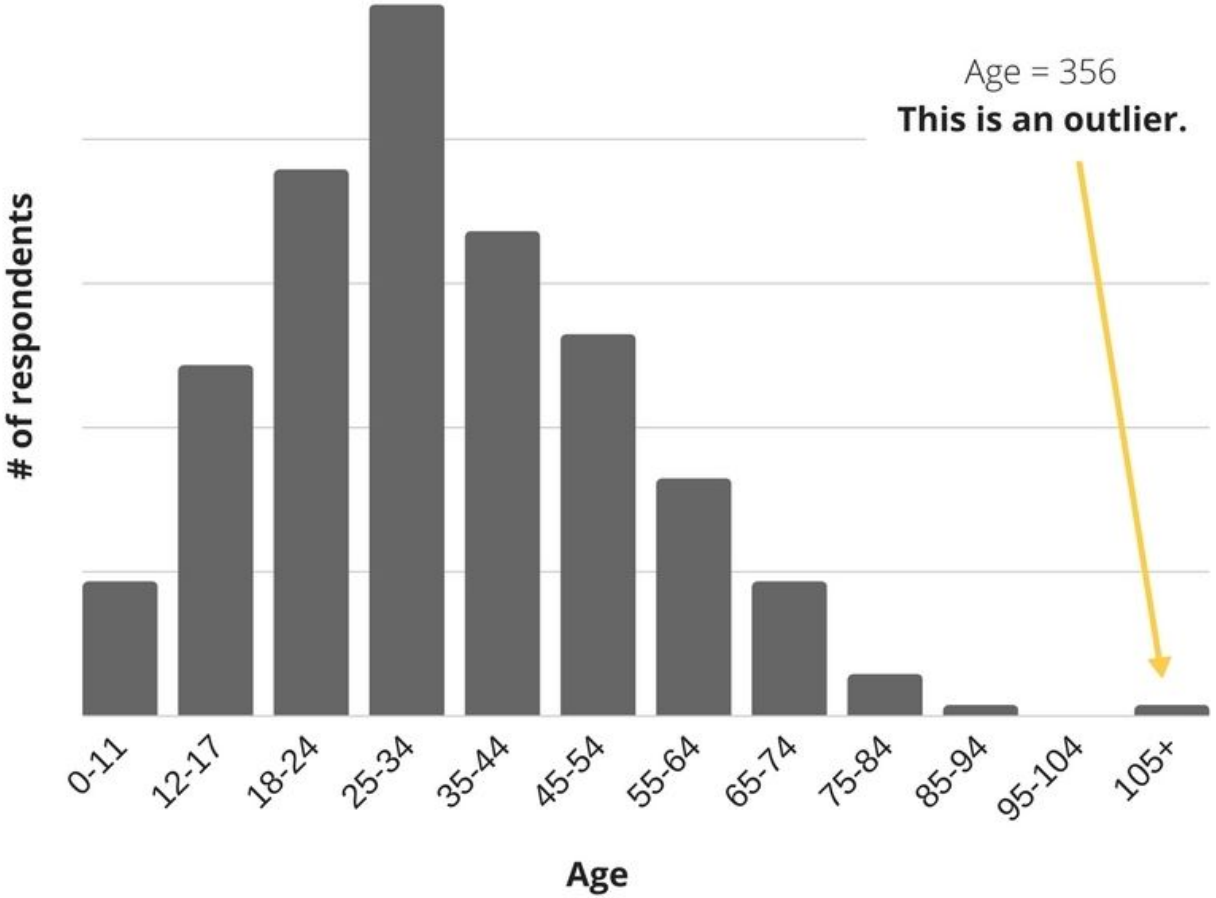
Traitement des valeurs aberrantes

Identifier les outliers :

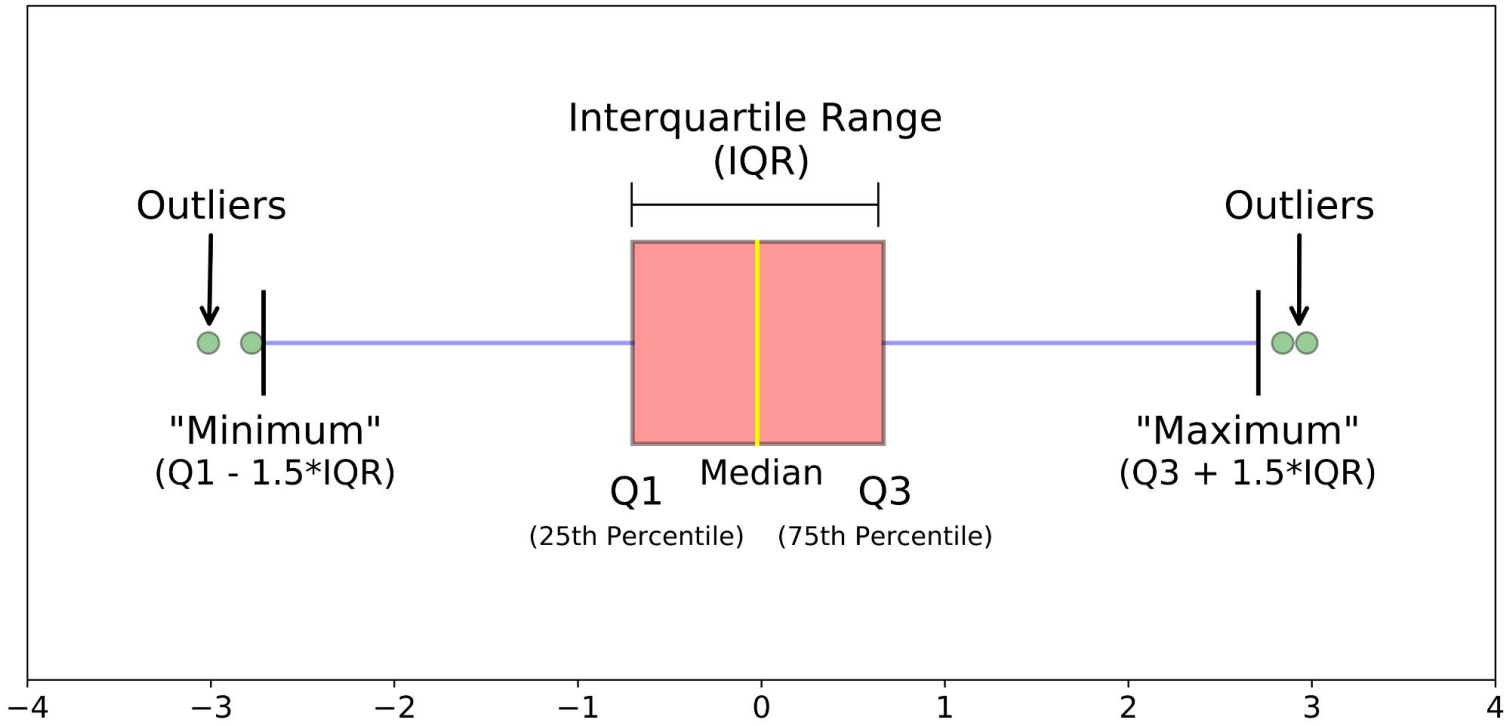
- Visualisation (boîtes à moustaches, histogrammes, nuages de points).
- Méthodes statistiques (z-score, IQR).

Méthodes de traitement :

- Supprimer les outliers.
- Remplacer par des valeurs moins extrêmes (winsorisation).
- Transformer les données (ex: logarithmique).



https://humansofdata.atlan.com/2018/03/when-delete-outliers-dataset/



https://medium.com/@agarwalvishal819/outlier-detection-with-boxplots-1b6757faf421

Encodage des variables catégorielles

Variables nominales (sans ordre) :

- Créer des variables indicatrices (dummy variables) (model.matrix(), dummyVars()).

Variables ordinales (avec ordre) :

- Attribuer un ordre numérique aux catégories.

Feature (Color)
Red
Green
Yellow
Green
Red

One Hot Encoding

Red	Green
1	0
0	1
0	0
0	1
1	0

Yellow Column dropped to avoid the Dummy Variable Trap

SCALER Topics

Normalisation et standardisation

- **Normalisation :**
    - Mettre les variables à l'échelle [0, 1].
    - Utile pour les algorithmes sensibles aux différences d'échelle.
- **Standardisation :**
    - Centrer les variables (moyenne = 0) et réduire (écart-type = 1) (scale()).
    - Utile pour les algorithmes basés sur la distance.

Sélection de variables

Objectif :

- Améliorer la performance du modèle.
- Réduire la complexité du modèle.
- Améliorer l'interprétabilité du modèle.

Méthodes :

- Méthodes de filtrage (corrélation, variance).
- Méthodes wrapper (stepwise, recherche exhaustive).
- Méthodes embedded (régularisation L1 (Lasso)).

Régression linéaire simple - Principe et équation

Objectif : modéliser la relation linéaire entre une variable explicative (prédicteur) et une variable à prédire.

Equation :

$$y = \beta_0 + \beta_1 x + \epsilon$$

- y : variable à prédire
- x : variable explicative
- β_0 : ordonnée à l'origine (intercept)
- β_1 : coefficient de régression (pente)
- ε : erreur du modèle

Régression linéaire multiple - Principe et équation

- **Extension de la régression simple à plusieurs variables explicatives.**
- **Equation :**
  - x_1, x_2, ..., x_p : variables explicatives

$$y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_p x_p + \epsilon$$

Évaluation des performances - RMSE

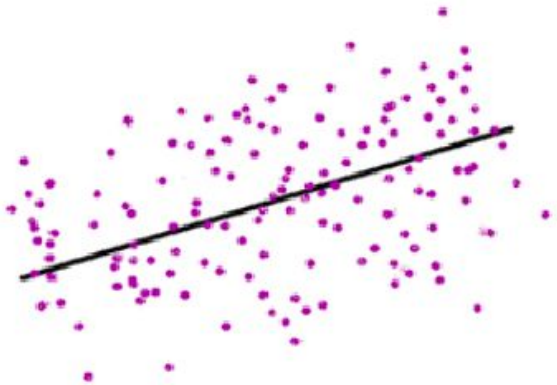
- **RMSE (Root Mean Squared Error) :** mesure l'erreur moyenne du modèle en unités de la variable à prédire.
- **Formule :**
  - y : valeurs observées
  - ŷ : valeurs prédites

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} \|y(i) - \hat{y}(i)\|^2}{N}},$$

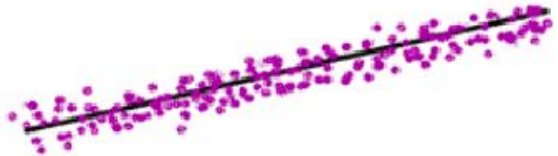
Coefficient de détermination

- **R^2 (coefficient de détermination) :** mesure la proportion de la variance de la variable à prédire expliquée par le modèle.

- **Valeurs entre 0 et 1 :** plus R^2 est proche de 1, meilleur est le modèle.



R-squared : 17%



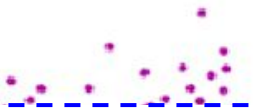
R-squared : 83%

https://www.knowledgehut.com/blog/data-science/interpret-r-squared-and-goodness-fit-regression-analysis

Coefficient de détermination

- **R^2 (coefficient de détermination) :** mesure la proportion de la variance de la variable à prédire expliquée par le modèle.

- **Valeurs entre 0 et 1 :** plus R^2 est proche de 1, meilleur est le modèle.



Autres métriques :

- ***RMSE* : erreur moyenne du modèle (carré des erreurs)**

- ***MAE* : moyenne des valeurs absolues des erreurs.**

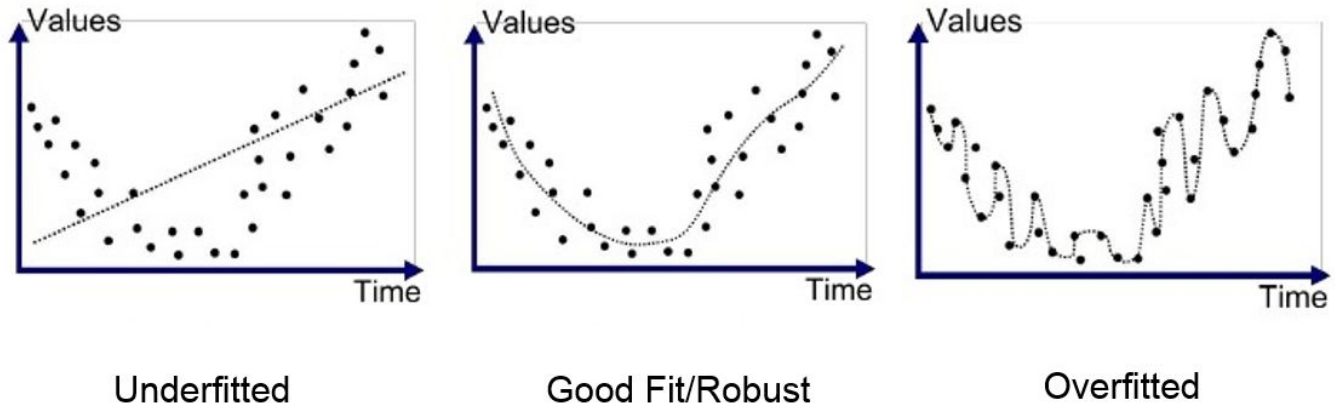
https://www.knowledgehut.com/blog/data-science/interpret-r-squared-and-goodness-fit-regression-analysis

Overfitting / Surapprentissage

Surapprentissage : le problème

L'overfitting (surajustement) se produit lorsque votre modèle apprend "trop bien" les données d'entraînement, au point de **capturer le bruit** et les variations aléatoires.

Il en résulte un modèle qui performe très bien sur les données d'entraînement, mais **généralise mal** aux nouvelles données.

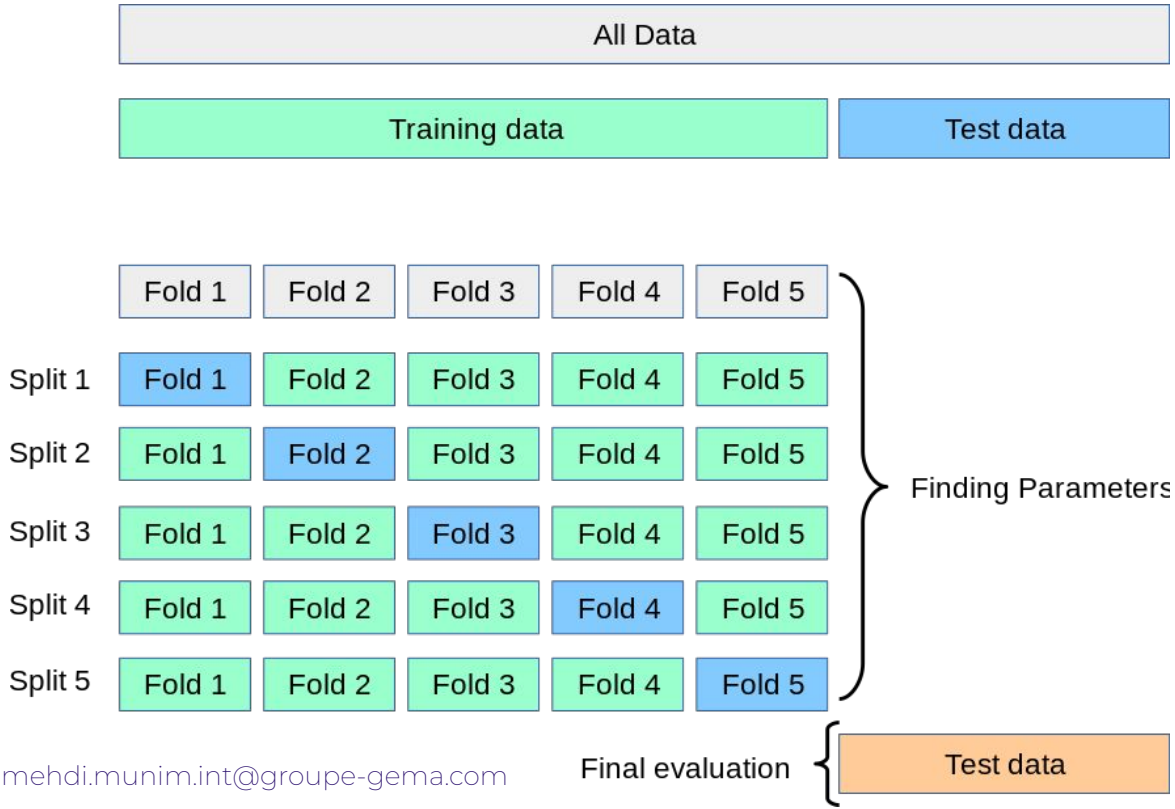


Comment gérer l’overfitting ?

Augmenter la quantité de données: Plus de données permettent au modèle de mieux généraliser.

Utiliser des techniques de **validation croisée** (ex: k-fold cross-validation) pour estimer la performance du modèle sur de nouvelles données et ajuster les hyperparamètres.

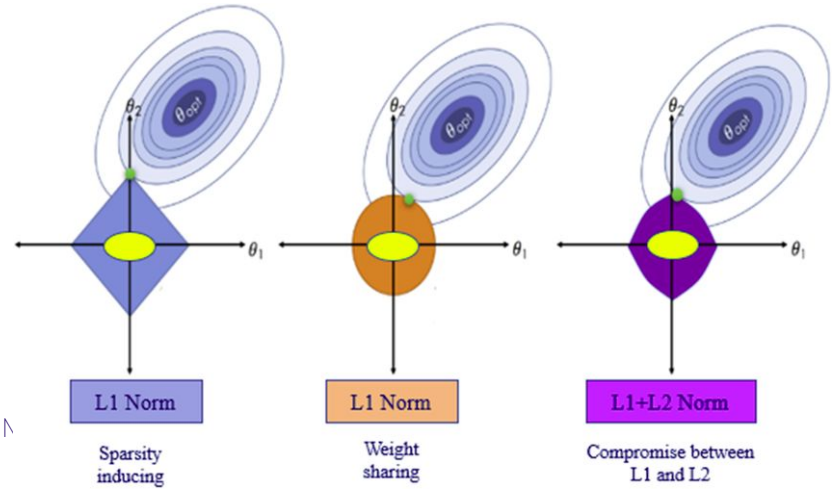
Gérer l'overfitting : Validation croisée

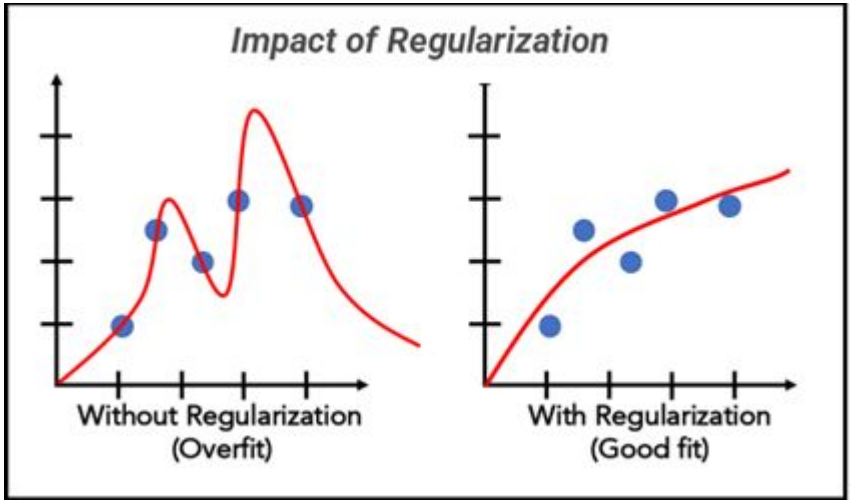


https://towardsmachinelearning.blogspot.com/2023/02/what-is-k-fold-cross-validation-working.html

Comment gérer l'overfitting ?

Régularisation: Ajouter une pénalité à la complexité du modèle pour éviter le surajustement (ex: Ridge, Lasso, Elastic Net).





Introduction aux modèles de classification

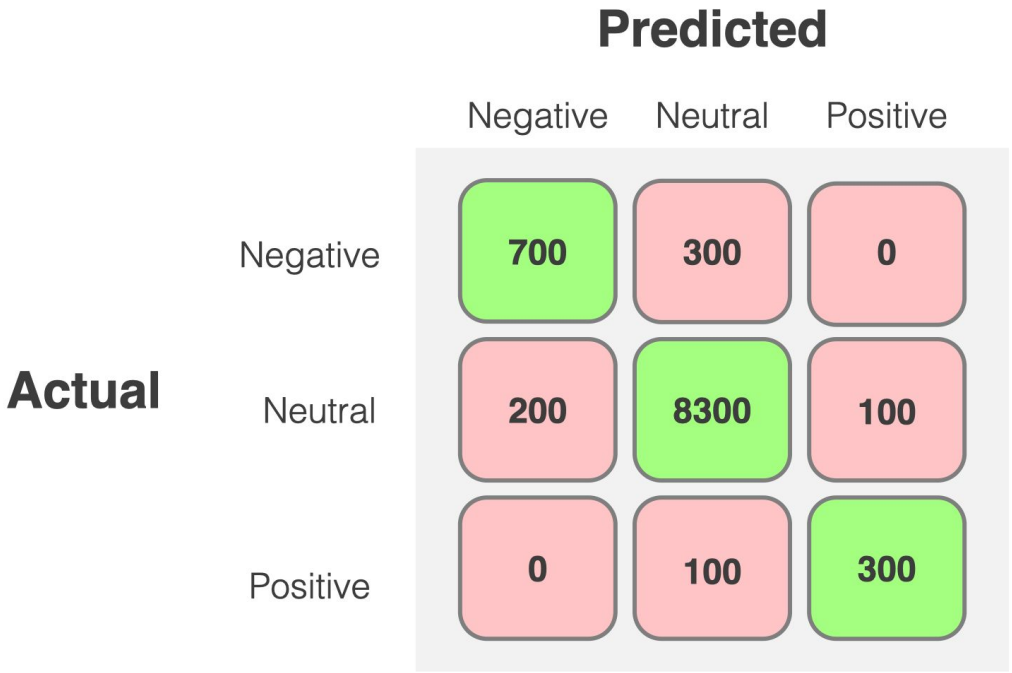
Rappel : la classification sert à prédire une variable catégorielle.

Différents types de modèles de classification : régression logistique, arbres de décision, SVM, etc.

Focus sur la régression logistique.

Source :
https://www.evidentlyai.com/classification-metrics/confusion-matrix

Évaluation des performances - Matrice de confusion



Métriques de classification - AUC

AUC (Area Under the Curve) : aire sous la courbe ROC (Receiver Operating Characteristic).

Courbe ROC : représente le taux de vrais positifs en fonction du taux de faux positifs pour différents seuils de classification.

AUC = 1 : modèle parfait.

AUC = 0.5 : modèle aléatoire.