

Algorithmes de classification

Noura BENHAJJI





noura.benhajji@gmail.com

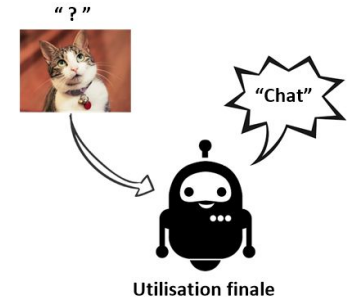
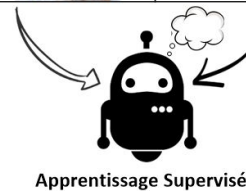
Introduction

- ❑ La classification supervisée est l'approche de machine learning **la plus utilisée et la mieux maîtrisée** à l'heure actuelle.
- ❑ La classification permet de résoudre des problèmes pratiques de la vie réelle
 - ❑ la détection de défaut d'usinage, de fraude, de maladie
 - ❑ le tri automatique de courrier, de document ou de vidéo
 - ❑ la reconnaissance d'images
- ❑ La classification permet de résoudre **les tâches où un choix est requis**.


Classification supervisée | Quésaco ?

- ❑ La classification supervisée consiste à **attribuer automatiquement une catégorie (ou une classe) à des données** dont on ne connaît pas la catégorie.
- ❑ Pour cela, **un classifieur** (algorithme de machine learning) est **entraîné** sur des données similaires ou très proches des données que l'on souhaite classer.

x	y
	"Chien"
	"Chien"
	"Chat"
	"Chien"



Crédit : machine Learnia

Veille individuelle  **20min**

Algorithmes de classification supervisée

- ❑ Lister les algorithmes de classification les plus utilisés
- ❑ Comment choisir “le meilleur” algorithme ?



- ❑ <https://www.kdnuggets.com/2020/05/guide-choose-right-machine-learning-algorithm.html>
- ❑ <https://docs.microsoft.com/fr-fr/azure/machine-learning/how-to-select-algorithms>

Algorithmes de classification supervisée

❑ Le k-plus proche voisin

- ❑ La méthode de k-proche voisins consiste à chercher dans une base de données l'exemple le plus proche de celui que l'on est entrain de traiter.

❑ L'arbre de décision

- ❑ Un outil d'aide à la décision représentant un ensemble de choix sous la forme graphique d'un arbre.
- ❑ les différentes décisions possibles sont situées aux extrémités des branches.

Algorithmes de classification supervisée

❑ Le random Forest

- ❑ On construit plusieurs arbres de décision de moins bonne qualité individuelle qui possède une vision réduite du problème.
- ❑ On réunit l'ensemble de ces estimateurs (classifieurs) pour avoir une vision globale.

❑ Le perceptron multicouches

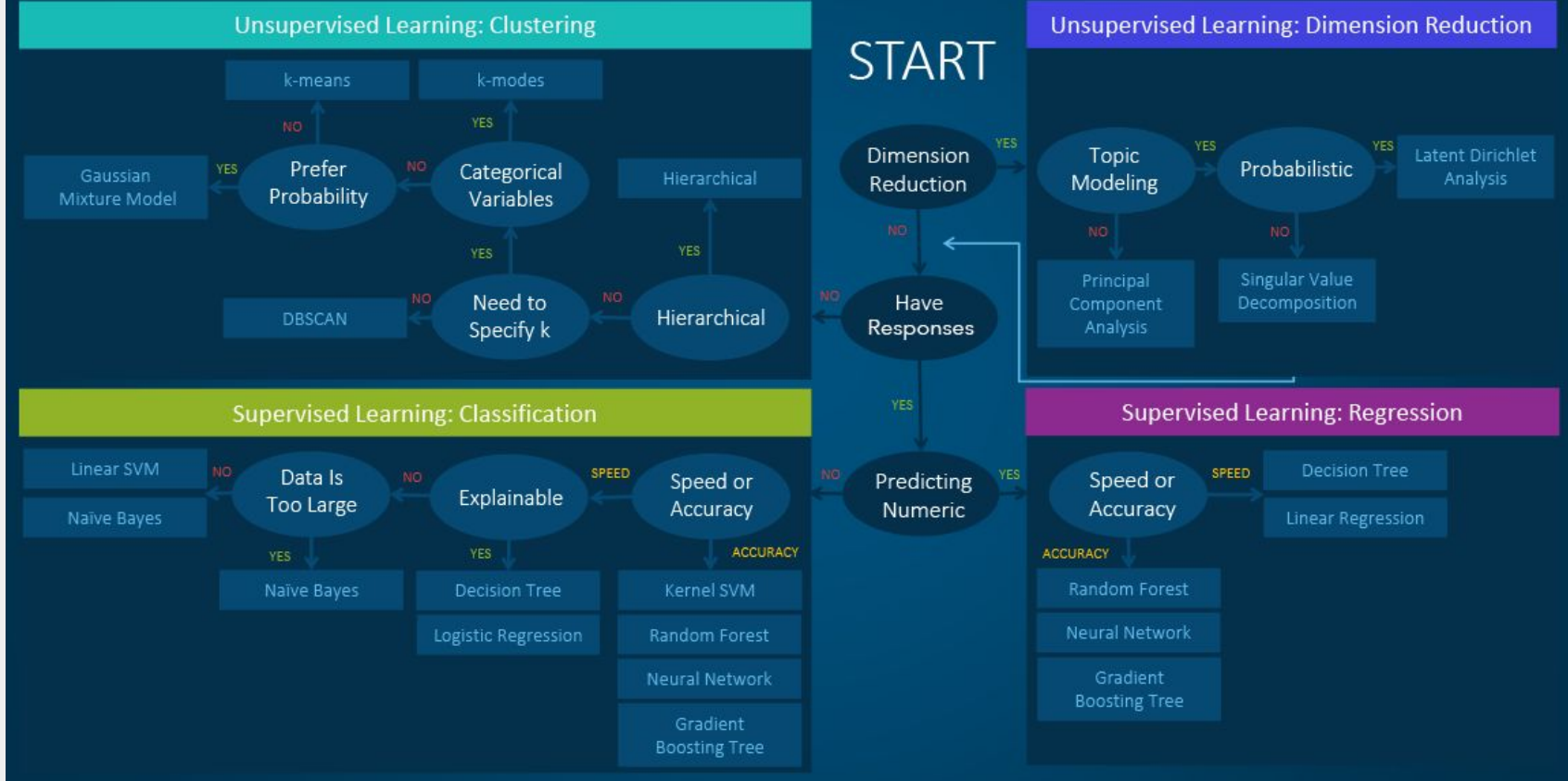
- ❑ Un ensemble de neurones connectés.
- ❑ Il est composé d'une couche d'entrée, de n couches cachées, et d'une couche de sortie.

Algorithmes de classification supervisée

❑ Régression logistique

- ❑ un modèle statistique qui permet d'étudier les relations entre un ensemble de variables d'entrée et une variable de sortie.
- ❑ il s'agit d'un modèle linéaire généralisé utilisant une fonction logistique comme fonction de lien.

Machine Learning Algorithms Cheat Sheet



Source : SAS Algorithm Flowchart

Veille individuelle



30min

Régression logistique

- ❑ Qu'est ce que la régression logistique ?
- ❑ Quels sont les types de régression logistiques ?
- ❑ Comment un modèle de régression logistique fonctionne ?



Régression logistique | Quésaco

- ❑ La régression logistique est utilisée pour estimer une valeur **discrète** (**classe** ou **catégorie**).
- ❑ La régression logistique est un **modèle statistique** qui permet d'étudier les relations entre un ensemble d'entrée **X** et une variable de sortie **y**.
- ❑ Un modèle de régression logistique permet aussi de **prédire la probabilité** qu'un événement arrive (valeur de 1) ou non (valeur de 0) à partir de l'**optimisation des coefficients de régression**.
- ❑ Lorsque la valeur prédite est supérieure à un seuil, l'événement est susceptible de se produire, alors que lorsque cette valeur est inférieure au même seuil, il ne l'est pas.

Régression logistique | Quésaco

Pourquoi régression logistique ?

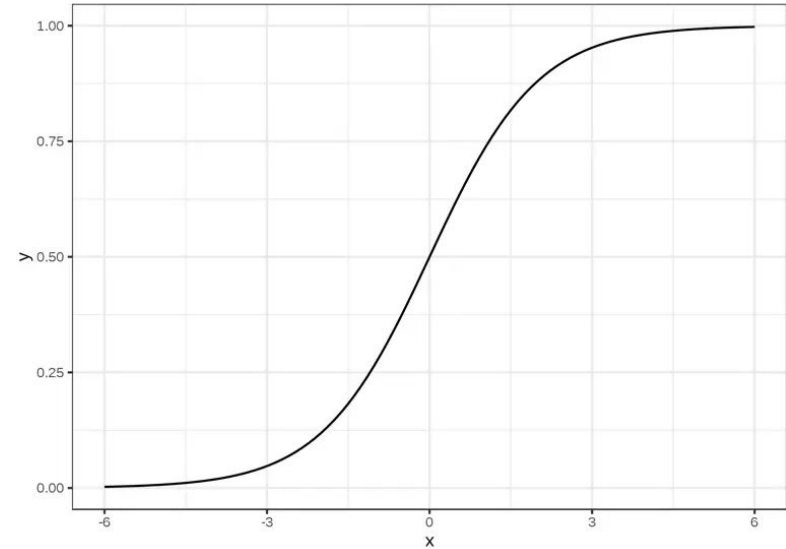
- ❑ **“Régression”** : on cherche à montrer une relation de dépendance entre une variable à expliquer et des variables explicatives. Cette dépendance s'exprime en terme de probabilité d'appartenir à chacune des classes.
- ❑ **“Logistique”** : la loi de probabilité est modélisée à partir d'une loi logistique.

Régression logistique | Quésaco

Mathématiquement

- ❑ Considérons une entrée
 $X = x_1, x_2, \dots, x_n$
- ❑ La régression a pour objectif de trouver une fonction h telle que :
$$y = \begin{cases} 1 & \text{si } h(X) \geq \text{seuil} \\ 0 & \text{si } h(X) < \text{seuil} \end{cases}$$
- ❑ On utilise la fonction **sigmoïde**

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$



Représentation de la fonction sigmoïde

Régression logistique | Quésaco

Mathématiquement

- ❑ La fonction h qui définit la régression logistique s'écrit alors :

$$\forall (X \in \mathbb{R}^n) \quad h(X) = \frac{1}{1 + e^{-\sum_{i=1}^n \theta_i x_i}}$$

- ❑ Le problème de classification par régression logistique est un problème d'**optimisation**, où, à partir de données, nous essayons d'**obtenir le meilleur jeu de paramètre** qui permettent à la courbe sigmoïde de **coller au mieux aux données**.
- ❑ C'est dans cette étape qu'intervient l'**apprentissage automatique**.

Régression logistique | Types

- ❑ Il existe plusieurs types de régression logistiques :
- ❑ **Régression logistique binaire**
 - ❑ oui/non
- ❑ **Régression logistique multinomiale** : la variable dépendante a trois catégories nominales ou plus.
 - ❑ prédiction de la catégorie d'iris
- ❑ **Régression logistique ordinale** : la variable dépendante a trois catégories ordinales ou plus.
 - ❑ la notation d'un produits de 1 à 5

Régression logistique | ChatGPT

NO

qu'est ce que la régression logistique



La régression logistique est un algorithme de classification utilisé pour résoudre les problèmes de classification binaire. Il utilise une fonction de probabilité logistique pour modéliser la probabilité qu'un élément appartienne à une certaine classe. Il peut également être utilisé pour résoudre les problèmes de classification multi-classes en utilisant des techniques dites de "one-vs-all" ou "softmax".

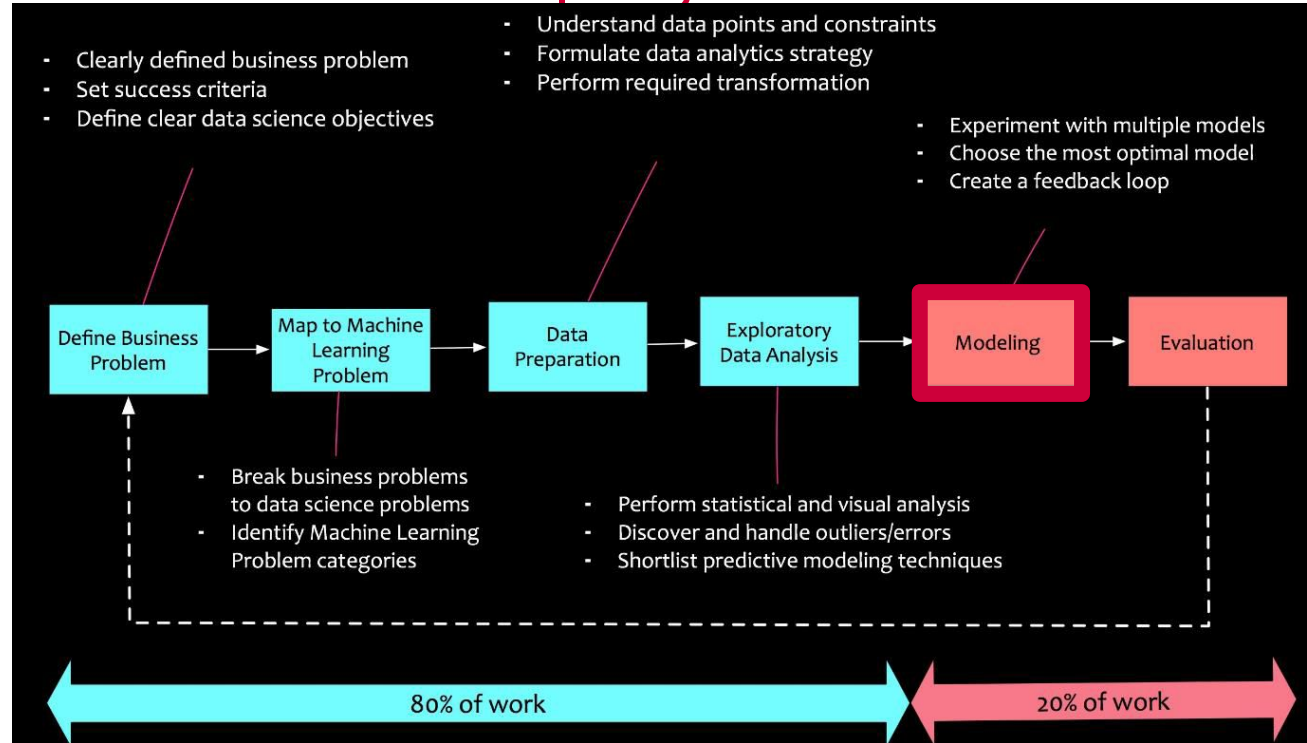


La régression logistique est basée sur la méthode des moindres carrés pour ajuster les coefficients d'un modèle linéaire aux données d'entraînement. Une fois que le modèle est entraîné, il peut être utilisé pour prédire la probabilité d'appartenance à une classe pour de nouveaux éléments de données.

La régression logistique est souvent utilisée lorsque les données sont linéairement séparables, c'est-à-dire qu'il existe une droite ou un hyperplan qui sépare les différentes classes. Elle est également utilisée pour résoudre les problèmes de classification binaire et multi-classes, pour des problèmes d'analyse de risque, de marketing, de biologie, de médecine, etc.

Mon premier modèle de ML

Cycle de vie d'un projet IA



Cycle de vie d'un projet IA

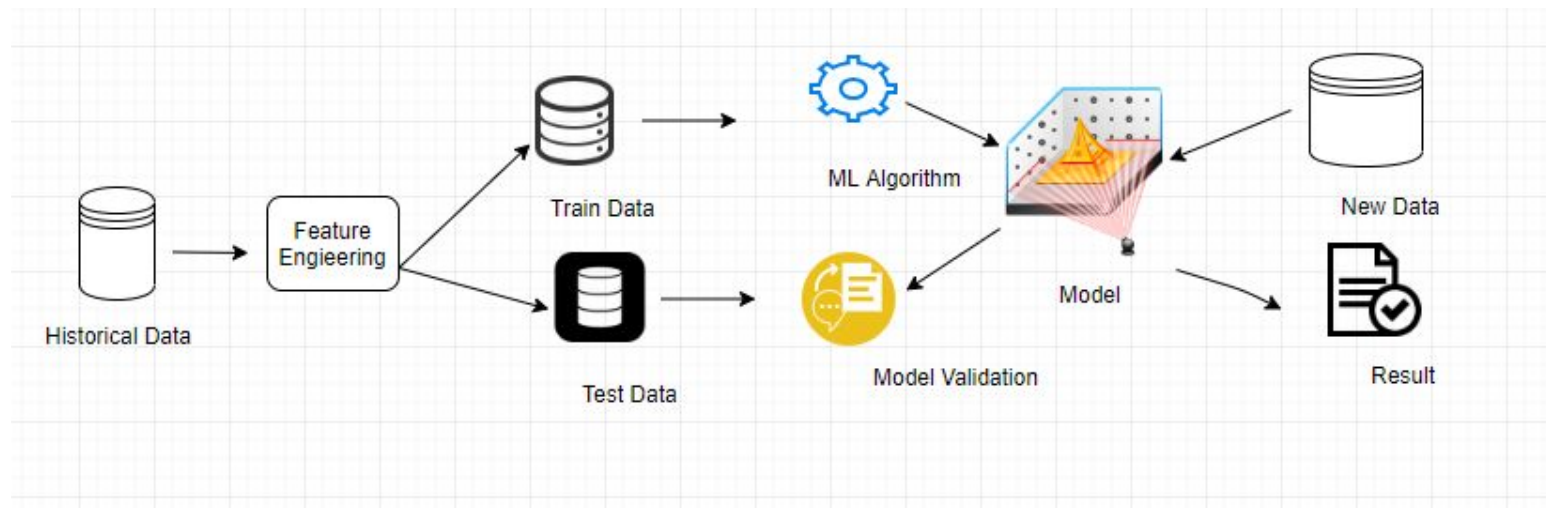
source ; [Data Science Simplified Part I : Principles and Process – Pradeep Menon](#)

Etapes clés



7 étapes vers l'apprentissage automatique

Source : [7 Steps to Machine Learning: How to Prepare for an Automated Future](#)



Etapes de création d'un modèle ML
Source : Machine Learning Workflow

Etape de construction d'un modèle

1. Importer les données
2. Séparation des données en sous ensemble d'entraînement et un sous ensemble de test. -> *train_test_split*
3. construction du modèle -> *LogisticRegression*
4. Entrainement du modèle avec le sous ensemble d'entraînement -> *fit*
5. Prédictions -> *predict*
6. Evaluation du modèle -> *score*

Travail en groupe



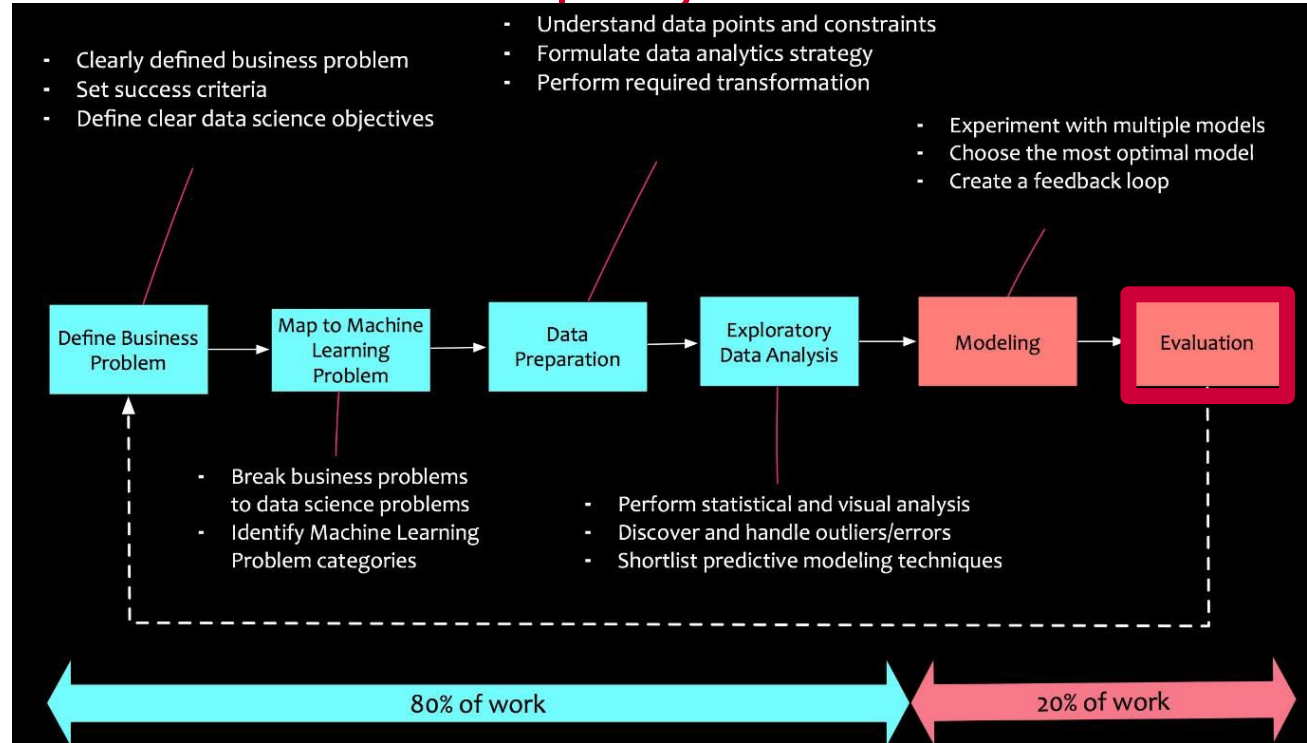
30min

Modèle de régression logistique

- ❑ Notebook : Régression
logistique – apprenants
- ❑ Partie I uniquement
- ❑ Niveau : Imiter



Cycle de vie d'un projet IA



Cycle de vie d'un projet IA

source ; [Data Science Simplified Part I : Principles and Process – Pradeep Menon](#)

Veille individuelle



45min

Evaluation du modèle

- ❑ Comment évaluer un modèle de classification ?
 - ❑ la matrice de confusion
 - ❑ l'exactitude
 - ❑ la précision
 - ❑ le rappel
 - ❑ F1 score
 - ❑ la courbe roc



<https://datascientest.com/comment-gerer-les-problemes-de-classification-desequilibree-partie-i#:~:text=Qu'est%20ce%20qu'une,m%C3%A9trique%20utilis%C3%A9e%20pour%20l'%C3%A9valuer.>

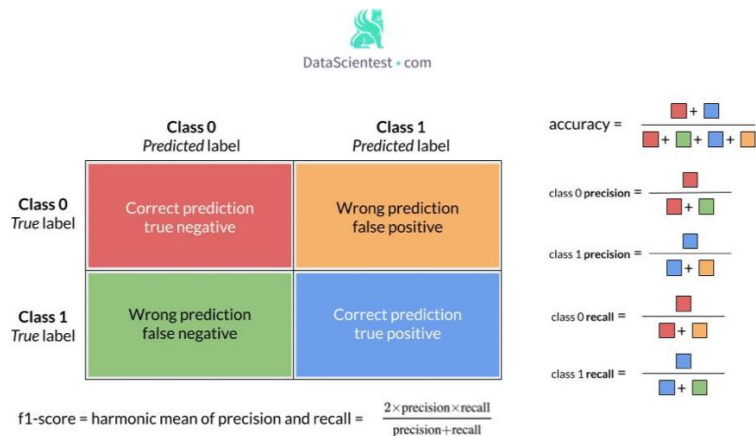
<https://datascientest.com/comment-gerer-les-problemes-de-classification-desequilibree-partie-i#:~:text=Qu'est%20ce%20qu'une,m%C3%A9trique%20utilis%C3%A9e%20pour%20l'%C3%A9valuer.>

<https://docs.microsoft.com/fr-fr/azure/machine-learning/component-reference/evaluate-model#metrics-for-classification-models>

<https://blog.octo.com/quel-sens-metier-pour-les-metriques-de-classification/>

Evaluation d'un modèle ML

- ❑ Métriques pour les modèles de classification
- ❑ **L'exactitude (accuracy)** : mesure l'adéquation d'un modèle de classification sous forme de proportion de résultats réels sur le nombre total de cas.
- ❑ La **précision (precision)** : correspond à la proportion de résultats réels sur tous les résultats positifs.



Source : [datascientest](https://datascientest.com)



DataScientest • com

	Class 0 Predicted label	Class 1 Predicted label
Class 0 True label	Correct prediction true negative	Wrong prediction false positive
Class 1 True label	Wrong prediction false negative	Correct prediction true positive

f1-score = harmonic mean of precision and recall = $\frac{2 \times \text{precision} \times \text{recall}}{\text{precision} + \text{recall}}$

$$\text{accuracy} = \frac{\text{red} + \text{blue}}{\text{red} + \text{green} + \text{blue} + \text{orange}}$$

$$\text{class 0 precision} = \frac{\text{red}}{\text{red} + \text{green}}$$

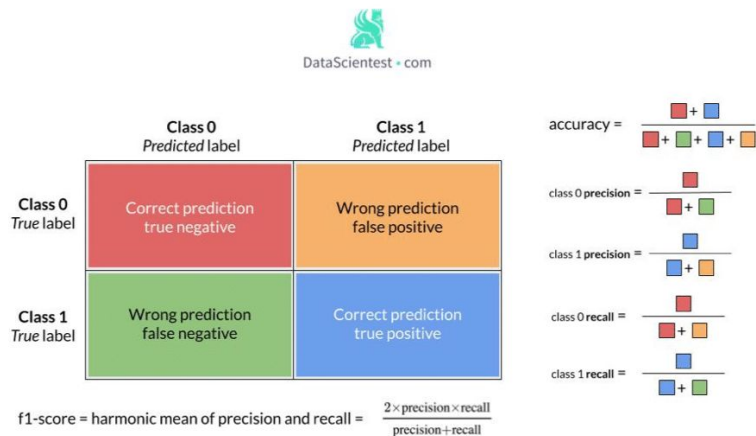
$$\text{class 1 precision} = \frac{\text{blue}}{\text{blue} + \text{orange}}$$

$$\text{class 0 recall} = \frac{\text{red}}{\text{red} + \text{orange}}$$

$$\text{class 1 recall} = \frac{\text{blue}}{\text{blue} + \text{green}}$$

Evaluation d'un modèle ML

- ❑ Métriques pour les modèles de classification
- ❑ Le **rappel (recall)** : est la fraction de la quantité totale d'instances pertinentes qui ont été réellement récupérées.
- ❑ Le **F1 Score** : la moyenne harmonique de précision et de rappel.



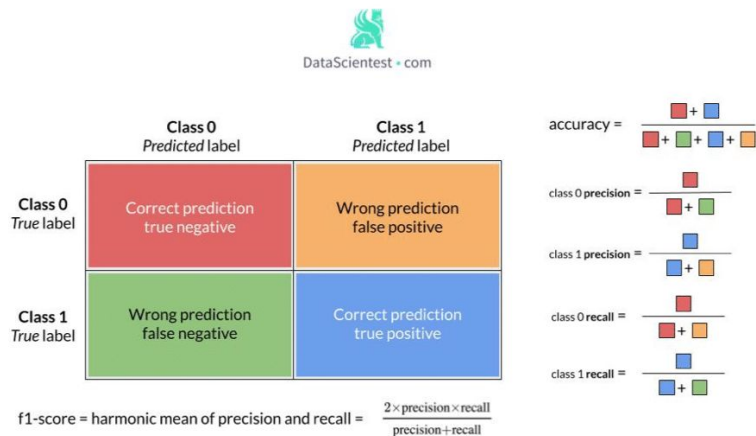
Source : [datascientest](https://datascientest.com)

Evaluation d'un modèle ML

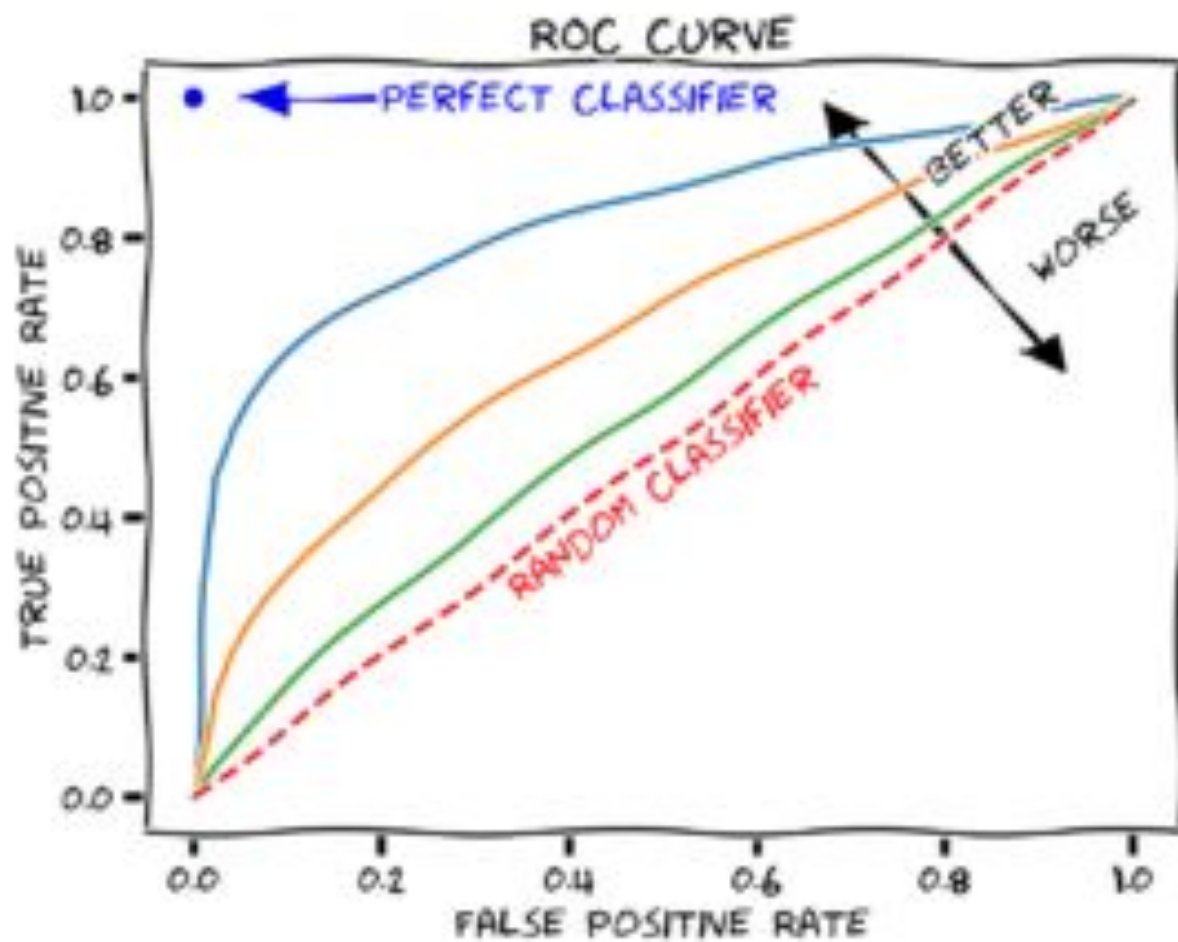
❑ Métriques pour les modèles de classification

❑ **AUC** : mesure la zone sous la courbe tracée avec les vrais positifs sur l'axe y et les faux positifs sur l'axe x.

Cette métrique est utile car elle fournit un nombre unique qui vous permet de comparer les modèles de types différents.



Source : [datascientest](https://datascientest.com)



Travail en binôme



45 min

Modèle de régression logistique

- ❑ Notebook : Régression
logistique – apprenants
- ❑ Partie II
- ❑ Niveau : Adapter



Etapes clés



7 étapes vers l'apprentissage automatique

Source : [7 Steps to Machine Learning: How to Prepare for an Automated Future](#)

Veille individuelle



45 min

La régularisation

- ❑ Quel est l'objectif de la régularisation ?
- ❑ Quels sont les types de régularisation ?
- ❑ Comment fonctionne l'algorithme de descente de gradient ?
- ❑ Quels sont les options possibles du **solver** avec scikit-learn



La régularisation

- ❑ La régularisation a pour objectif
 - ❑ de minimiser la complexité du modèle
 - ❑ de minimiser la fonction coût
 - ❑ de parer à le overfitting
- ❑ Pour cela, on ajoute un terme de **régularisation** λ à la fonction

La régularisation

- ❑ Il existe 3 types de régularisation :
- ❑ **Ridge** (ou l_1) $ax + by + cz$
 - ❑ permet de réduire l'amplitude des coefficients d'une régression linéaire
 - ❑ groupe les variables corrélés, en leur affectant des coefficients similaires.
- ❑ **Lasso** (ou l_2)
 - ❑ permet d'annuler certains coefficients
 - ❑ si plusieurs variables sont corrélées, une d'entre elles est choisie de façon aléatoire

La régularisation

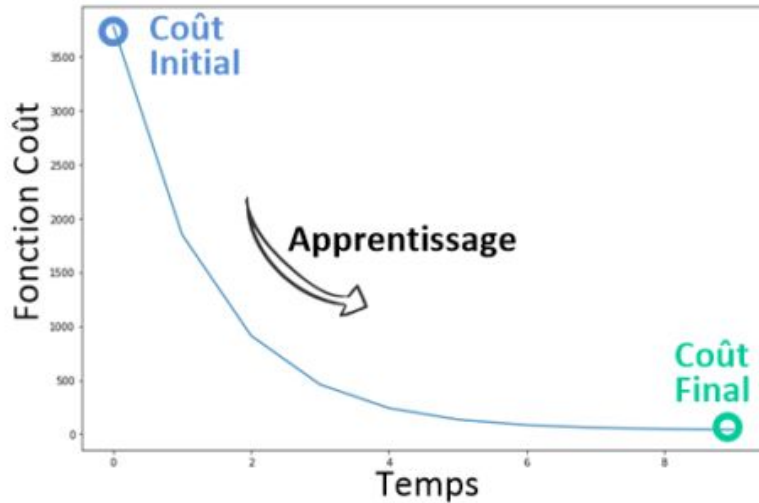
- ❑ Il existe 3 types de régularisation :
- ❑ **Elastic net**
 - ❑ combine les normes l_1 et l_2
 - ❑ capacité de sélection de variables : exclusion des variables non pertinentes
 - ❑ groupe de variables prédictives corrélées, partage de poids et non plus sélection arbitraire.

La régularisation

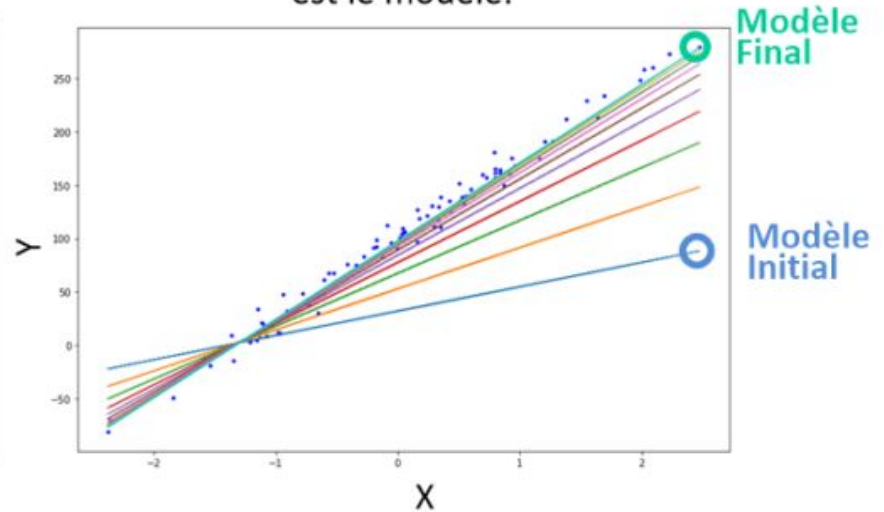
- ❑ Le paramètre **solver** représente l'algorithme à utiliser pour le problème d'optimisation.
- ❑ Les options possibles sont :
 - ❑ **linlinear** : (option par défaut) le meilleur choix pour les petits ensembles de données. Il gère la pénalité L1.
 - ❑ **newton-cg** : gère que la pénalité L2.
 - ❑ **lbfgs** : gère la perte multinomiale pour les problèmes multi classes. Il ne gère également que la pénalité L2.
 - ❑ **saga** : représente un bon choix pour les grands ensembles de données. Pour les problèmes multi classes, il gère les pertes multinomiales. Il prend en charge la pénalité L1 et la pénalité « elasticnet ».
 - ❑ **sag** : cette option est aussi idéale pour les grands ensembles de données et gère la perte multinomiale pour les problèmes multi classes.

La descente de Gradient

Minimisation de la Fonction Coût



Plus la Fonction Coût est faible, meilleur est le modèle.



Source : [Descente de gradient - Machine Learnia](#)

La descente de Gradient

- ❑ La formule générale est : $x_{t+1} = x_t - \eta \Delta x_t$
 - ❑ η taux d'apprentissage
 - ❑ Δx_t direction de la descente

Note : L'algorithme de descente du gradient décide de suivre comme direction de descente l'opposé du gradient (dérivée). Le gradient indique la croissance maximale d'une fonction à partir d'un point.

La descente de Gradient

L'algorithme de descente de gradient est comme suit :

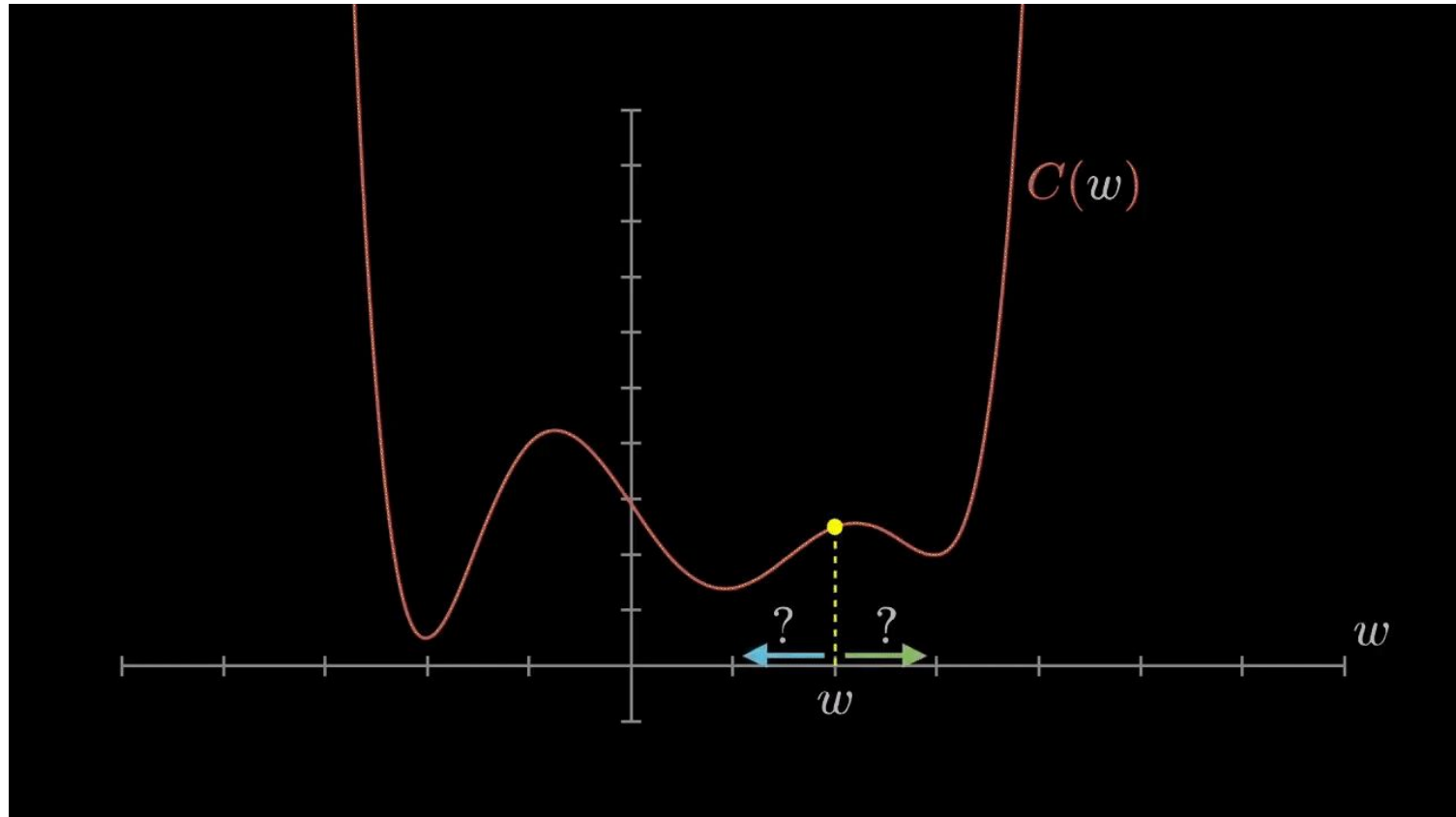
1. Soit un point d'initialisation x_0
2. calculer $f(x_t)$
3. mettre à jour les coordonnées : $x_{t+1} = x_t - \eta \Delta f(x_t)$
4. répéter 2 et 3 jusqu'au critère d'arrêt $|f(x_{t+1}) - f(x_t)| \leq \epsilon$

La descente de Gradient

Attention

- ❑ deux éléments cruciaux pour le bon fonctionnement de l'algorithme de descente de gradient :
 - ❑ le point d'initialisation x^0 et la valeur du taux d'apprentissage.
- ❑ Un mauvais point d'initialisation ou un taux d'apprentissage peu adapté peut empêcher l'algorithme de converger vers le minimum.
- ❑ Plus le taux d'apprentissage est élevé, plus on suivra loin la direction indiquée par le gradient. → minimum local

La descente de Gradient

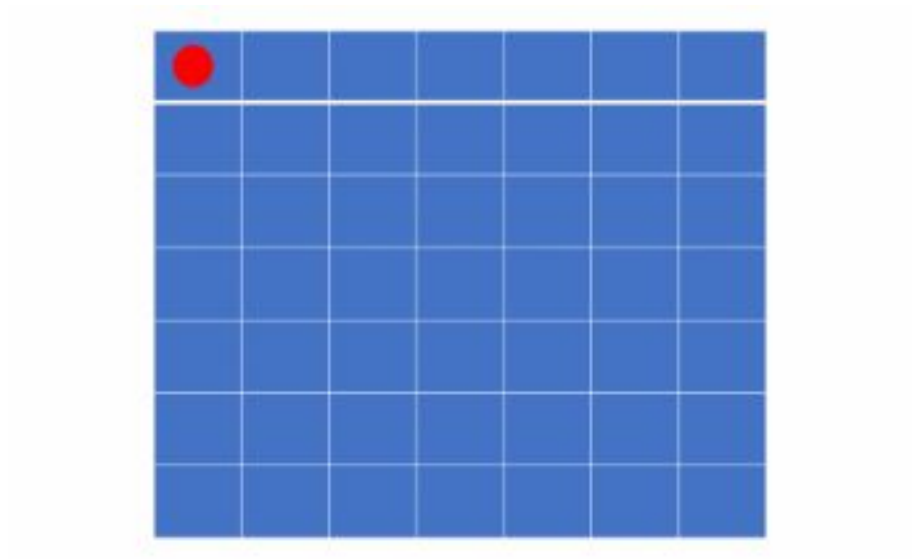


Optimisation des hyper paramètres

❑ Gridsearch

- ❑ le problème d'optimisation est considéré comme un problème de recherche dans une grille.
- ❑ On fixe pour chaque hyperparamètre un ensemble de valeurs qu'il peut prendre
- ❑ Pour chaque combinaison d'hyperparamètres, on entraîne le modèle et on conserve les performances en mémoire
- ❑ on choisi les hyperparamètres du modèle avec les meilleurs performances.

Optimisation des hyper paramètres



❑ Limite

❑ capacité de calcul

Optimisation des hyper paramètres

```
>>> from sklearn import svm, datasets
>>> from sklearn.model_selection import GridSearchCV
>>> iris = datasets.load_iris()
>>> parameters = {'kernel':('linear', 'rbf'), 'C':[1, 10]}
>>> svc = svm.SVC()
>>> clf = GridSearchCV(svc, parameters)
>>> clf.fit(iris.data, iris.target)
GridSearchCV(estimator=SVC(),
              param_grid={'C': [1, 10], 'kernel': ('linear', 'rbf')})
>>> sorted(clf.cv_results_.keys())
['mean_fit_time', 'mean_score_time', 'mean_test_score', ...
 'param_C', 'param_kernel', 'params', ...
 'rank_test_score', 'split0_test_score', ...
 'split2_test_score', ...
 'std_fit_time', 'std_score_time', 'std_test_score']
```

Hide prompts
and outputs

Optimisation des hyper paramètres

Vocabulaire : les paramètres des modèles de machine learning que l'on peut modifier sont appelés des hyperparamètres. Attention à ne pas les confondre avec les paramètres du modèle qui eux sont calculés automatiquement pendant l'entraînement.

Exemple : le nombre de couches d'un réseau de neurones est un hyperparamètre, le biais d'un neurone donné est un paramètre du réseau.

Travail en binôme



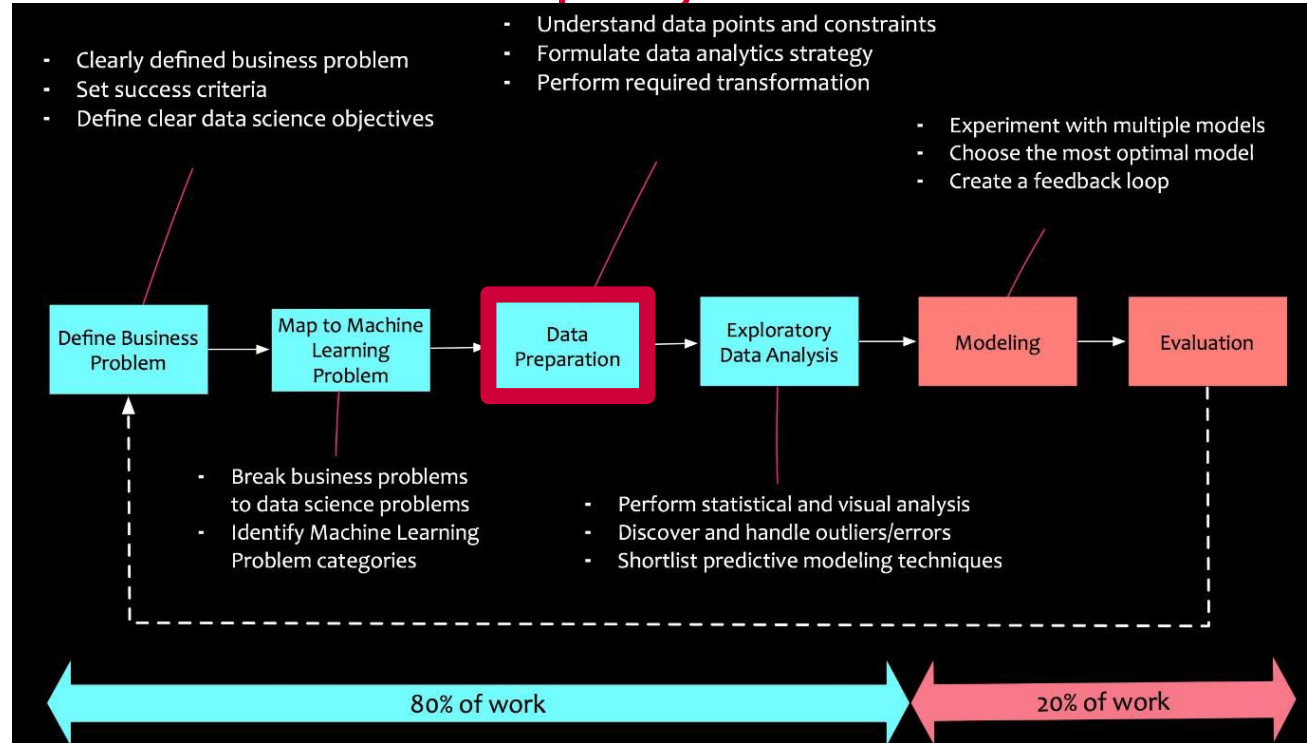
45 min

Modèle de régression logistique

- ❑ Créer trois modèles de régression logistique avec trois solvers différents.
- ❑ Conclure sur les performances du modèle.
- ❑ Refaire le même exercice en utilisant une gridsearch pour optimiser l'hyperparamètre solver



Cycle de vie d'un projet IA



Cycle de vie d'un projet IA

source ; [Data Science Simplified Part I : Principles and Process – Pradeep Menon](#)

Préparation des données

Les actions principales sont :

- ❑ **Unifier les intitulés** : erreurs de saisie ou d'incompatibilités entre la source de données et la base.
- ❑ **Identifier les données aberrantes** : erreurs de saisie, outliers.
- ❑ **Traiter les données numériques** : dans le cas où les échelles de grandeur sont incompatibles (normalisation)
- ❑ **Traiter les valeurs manquantes** : suppression, amputation avec des données calculer, aléatoires, etc.
- ❑ **Supprimer les doublons.**

Préparation des données

- ❑ Les modèles d'apprentissage basé sur des données non préparées présentent des biais qui peuvent fausser l'analyse, voire, la rendre impossible

=> **la préparation des données permet d'optimiser l'efficacité des algorithmes.**

Nettoyage des données

Les problèmes de qualité des données les plus fréquents sont

- ❑ **Données erronées** : erreurs de saisie ou d'incompatibilités entre la source de données et la base.
- ❑ **Données incomplètes (Imputation)** : seuls les champs obligatoires sont renseignés.
- ❑ **Données non normées** : donnée identique sous des formats différents.
- ❑ **Doublons**

Données manquantes

Les valeurs manquantes impactent fortement la qualité du modèle de machine learning. Deux stratégies possibles :

- ❑ Supprimer les observations avec des valeurs manquantes.
- ❑ Imputer les valeurs manquantes.

Veille individuelle



30min

Valeurs manquantes

- Types de valeurs manquantes
- Méthodes d'imputation
- Méthodes d'imputation avec scikit learn
 - ❑ SimpleImputer
 - ❑ KNNImputer



Valeurs manquantes

Les valeurs manquantes peuvent être :

- ❑ **MCAR** (Missing Completely at random) : une donnée est manquante de façon aléatoire si la probabilité d'absence est la même pour toutes les observations. La probabilité ne dépend que de paramètres extérieurs.
- ❑ **MAR** (Missing at random) : une donnée n'est pas manquante d'une façon complètement aléatoire si la probabilité d'absence est liée à une ou plusieurs autres variables observées
- ❑ **MNAR** (Missing not at random) : une donnée est manquante de façon non aléatoire si la probabilité d'absence dépend de la variable en question.

Pour aller plus loin

Valeurs manquantes

Les valeurs manquantes impactent fortement la qualité du modèle de machine learning.

Deux stratégies possibles :

- ❑ Supprimer les observations avec des valeurs manquantes.
- ❑ Imputer les valeurs manquantes.

Méthodes d'imputation

Plusieurs stratégies peuvent être utilisées :

- ❑ supprimer les observations avec des données manquantes.
- ❑ supprimer les colonnes avec un certain seuil de valeurs manquantes.
- ❑ remplacer les valeurs manquantes par une valeur par défaut.
- ❑ remplacer les valeurs manquantes par la moyenne ou la médiane des autres valeurs de la colonne (variables numériques)
- ❑ remplacer les valeurs manquantes par la valeur la plus fréquente (variables catégorielles)

Travail en groupe



Modèle de régression logistique

- ❏ Notebook : Brief projet
Titanic apprenants
- ❏ Questions 1 à 4



Valeurs aberrantes (outliers)

Plusieurs stratégies peuvent être utilisées :

- ❑ Plusieurs algorithmes de ML sont sensibles aux données d'entraînement et à leurs distributions.
- ❑ Les valeurs aberrantes peuvent rendre la phase d'entraînement plus longue.
- ❑ Les valeurs aberrantes influencent certains paramètres statistiques. Détecter les outliers permet de formuler de bonnes hypothèses.

Veille individuelle



30min

Valeurs aberrantes

- ❑ Comment les détecter ?
- ❑ Quelle stratégie adopter ?

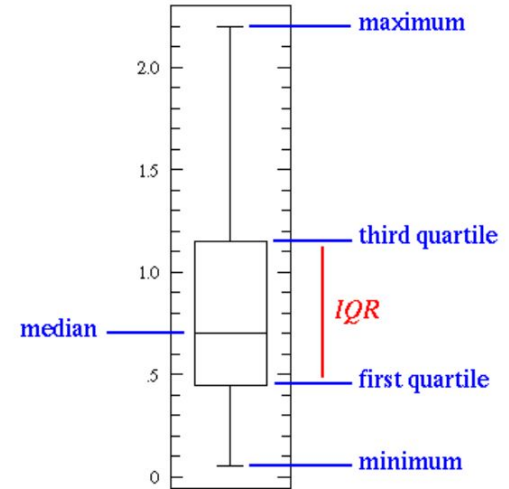


Valeurs aberrantes (outliers)

La détection des outliers se fait à l'aide de méthodes de visualisation

❑ Box plot

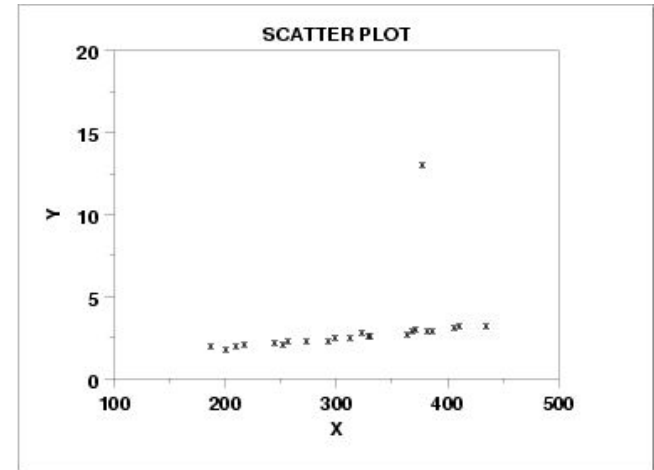
- ❑ permet de visualiser la distribution d'une seule variable
- ❑ un outlier est toute valeur extrême, supérieure ou inférieure à 1.5 fois l'écart interquartile IQR



Valeurs aberrantes (outliers)

La détection des outliers se fait à l'aide de méthodes de visualisation

- ❑ Scatter plot
 - ❑ détecter les valeurs aberrantes en fonction de plusieurs données
 - ❑ mettre en évidence une corrélation entre deux features



Valeurs aberrantes (outliers)

Avant de décider de la stratégie de comment gérer les outliers, il faut se poser les questions suivantes :

- ☐ est ce que une valeur due à une erreur de mesure ou de collecte d'information ?
- ☐ est ce que d'un point de vue métier cette valeur est possible ?
- ☐ est ce que cette observation est bénéfique pour le modèle ?

Valeurs aberrantes (outliers)

- ❑ Si le modèle prédictif est sensible aux outliers, **il faut les supprimer.**
 - ❑ les valeurs aberrantes biaiseront le modèle
- ❑ Si le modèle doit détecter un comportement anormal, comme les fraudes, **il faut les garder.**
 - ❑ le modèle doit apprendre un comportement anormal.

Travail en groupe



15 min

Modèle de régression logistique

- ❏ Notebook : Brief projet
Titanic apprenants
- ❏ Question 5



Feature Scaling

- ❑ Le feature scaling est une technique souvent appliquée sans le cadre de la préparation des données pour le ML.
- ❑ L'objectif est de modifier les valeurs des colonnes numériques du jeu de données pour utiliser une échelle commune, sans que les différences de plages de valeurs ne soient faussées et sans perte d'informations.
- ❑ Certains algorithmes ont besoin de données mises à l'échelle.
- ❑ Le feature scaling permet de préparer les données quand elles ont des échelles différentes.

Veille individuelle



30min

Valeurs aberrantes

- ❑ Comment les détecter ?
- ❑ Quelle stratégie adopter ?



Feature Scaling | Normalisation

- ❑ **Min-Max Scaling** peut- être appliqué quand les données varient dans des échelles différentes.
- ❑ A l'issue de cette transformation, les features seront comprises dans un intervalle fixe [0,1].
- ❑ Le but d'avoir un tel intervalle restreint est de réduire l'espace de variation des valeurs d'une feature et par conséquent réduire l'effet des outliers.

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

Feature Scaling | Standardisation

- ❑ La standardisation (aussi appelée Z-Score normalisation) peut-être appliquée quand les input features répondent à des distributions normales avec des moyennes et des écart-types différents.
- ❑ Par conséquent, cette transformation aura pour impact d'avoir toutes nos features répondant à la même loi normale.
- ❑ La standardisation peut également être appliquée quand les features ont des unités différentes.
- ❑ La Standardisation est le processus de transformer une feature en une autre qui répondra à la loi normale

Feature Scaling | Scikit Learn

- ❑ La **normalisation** peut- être appliquée par le min-max scaling. Python propose pour cela une classe nommée **MinMaxScaler** dans le package preprocessing.
- ❑ Pour appliquer la **standardisation**, on peut utiliser la classe **StandardScaler** de Scikit Learn.

Travail en groupe



Modèle de régression logistique

- ❏ Notebook : Brief projet
Titanic apprenants
- ❏ Question 5



Encodage des variables catégorielles

- ❑ Les algorithmes de ML ne traitent en entrée que des valeurs numériques.
- ❑ Les dataset sont majoritairement constitué de variables catégorielles.
- ❑ Il faut donc les transformer pour pouvoir les exploiter

Veille individuelle



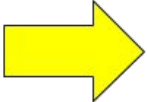
30min

Méthodes d'encodage



Encodage one-hot

- ❑ Le **One-hot encoding** permet de traiter les données sous formes vectorielles et de répondre à la problématique de transformation des données catégorielles en données numériques.
- ❑ Un vecteur de valeurs binaires de dimension nombre de classes est automatiquement créé et utilisée.



Color
Red
Red
Yellow
Green
Yellow

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

Encodage one-hot

❑ Implémentation avec scikit-learn OneHotEncoder

```
>>> from sklearn.preprocessing import OneHotEncoder
```

```
>>>
```

One can discard categories not seen during `fit`:

```
>>> enc = OneHotEncoder(handle_unknown='ignore')
>>> X = [['Male', 1], ['Female', 3], ['Female', 2]]
>>> enc.fit(X)
OneHotEncoder(handle_unknown='ignore')
>>> enc.categories_
(array(['Female', 'Male'], dtype=object), array([1, 2, 3], dtype=object))
>>> enc.transform([['Female', 1], ['Male', 4]]).toarray()
array([[1., 0., 1., 0., 0.],
       [0., 1., 0., 0., 0.]])
>>> enc.inverse_transform([[0, 1, 1, 0, 0], [0, 0, 0, 1, 0]])
array([['Male', 1],
       [None, 2]], dtype=object)
>>> enc.get_feature_names_out(['gender', 'group'])
array(['gender_Female', 'gender_Male', 'group_1', 'group_2', 'group_3'], ...)
```

Encodage ordinal

- ❑ Le **Ordinal Encoding** attribue à chaque valeur ordinal une valeur entière.
- ❑ Implémentation avec la classe **OrdinalEncoder** de Scikit-learn

Original Encoding	Ordinal Encoding
Poor	1
Good	2
Very Good	3
Excellent	4

Avantages et Inconvénients

Avantages

- ❑ Algorithme simple, efficace et facile à mettre en oeuvre
- ❑ ne nécessite pas une grande puissance de calcul
- ❑ fournit des scores de probabilité pour les observations
- ❑ très utilisés pour le scoring

Inconvénients

- ❑ Ne peut pas résoudre le problème de non-linéarité ce qui nécessite la transformation des caractéristiques non linéaires.
- ❑ Il faut faire très attention à l'interprétabilité des paramètres

Ressources

- Fundamental Techniques of Feature Engineering for ML
- Imputation de données manquantes
- Tout savoir sur les valeurs aberrantes
- Data preprocessing : Feature Scaling avec Python
- Smarter Ways to Encode Categorical Data for Machine Learning