

Apprentissage automatique

January 2023



Définition de Machine Learning

- L'apprentissage automatique est une branche de l'intelligence artificielle (IA) qui se concentre sur le développement de programmes informatiques capables d'apprendre et de s'adapter grâce à l'expérience.
- Il s'agit d'un sous-ensemble de l'IA qui existe depuis les années 1950, lorsque les premiers algorithmes ont été développés.
- L'objectif principal de l'apprentissage automatique est de créer des programmes informatiques capables d'apprendre à partir de données, d'identifier des modèles et de prendre des décisions avec un minimum d'interférences humaines.
- Les systèmes d'apprentissage automatique peuvent être utilisés dans une variété d'applications telles que la vision par ordinateur, le traitement du langage naturel, la robotique et l'analyse financière.

Inventeur Machine Learning

- L'invention de l'apprentissage automatique est attribuée à Arthur Samuel, qui a développé le premier programme d'auto-apprentissage en 1959.
- Samuel était un pionnier américain dans le domaine des jeux informatiques et de l'intelligence artificielle (IA).
- Il a créé un programme de jeu de dames qui pouvait apprendre de ses erreurs et ajuster sa stratégie en fonction de son expérience.



L'apprentissage automatique prend son envol

- L'apprentissage automatique a vraiment pris son envol à la fin des années 1990, lorsque les progrès de la puissance de calcul et des algorithmes ont permis d'analyser des ensembles de données plus importants et de créer des modèles plus complexes.
- Ce nouveau niveau de puissance de calcul a permis aux chercheurs de créer des algorithmes plus puissants capables de traiter et d'apprendre de grandes quantités de données.
- Au début des années 2000, le développement de bibliothèques logicielles open source telles que TensorFlow et PyTorch a permis aux chercheurs de mettre en œuvre plus facilement leurs idées et de les partager avec d'autres chercheurs.
- Cela a conduit à une explosion de la recherche dans le domaine, avec de nouveaux algorithmes et applications développés à un rythme rapide.
- Au cours de la dernière décennie, l'apprentissage automatique est devenu de plus en plus populaire et est utilisé dans une variété d'applications, y compris la vision par ordinateur, le traitement du langage naturel, la robotique et l'analyse financière.
- Au fur et à mesure que la technologie continue de s'améliorer, on s'attend à ce que l'apprentissage automatique devienne une partie encore plus intégrante de la vie moderne.
-

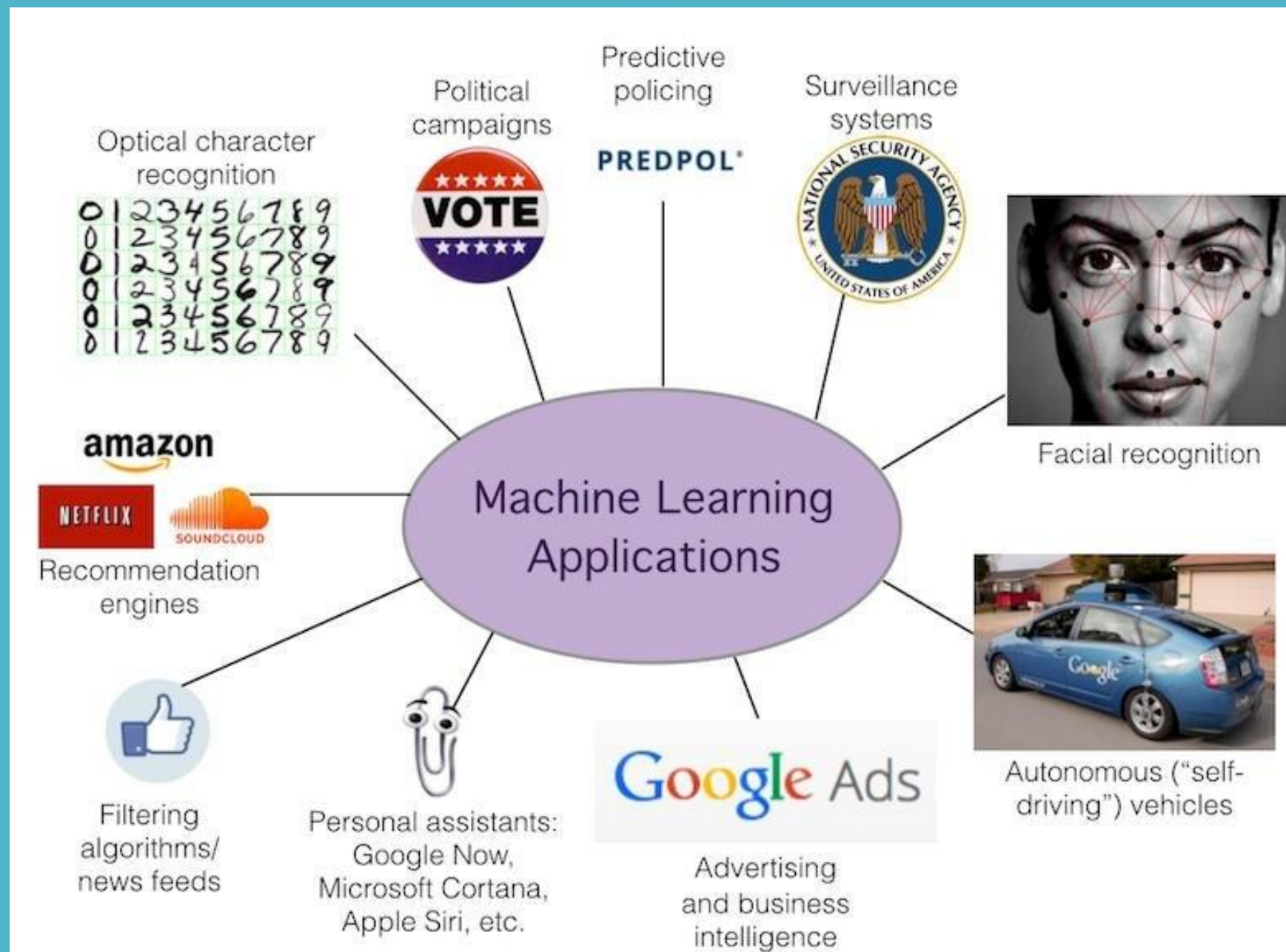


Machine Learning et informatique

- L'amélioration de l'informatique qui a rendu possible l'apprentissage automatique dans les années 1990 a été les progrès de la puissance de calcul et des algorithmes.
- Cela a permis aux chercheurs d'analyser de plus grands ensembles de données et de créer des modèles plus complexes.
- La puissance de calcul a été améliorée par l'invention de processeurs plus puissants, d'algorithmes plus efficaces et de périphériques de stockage plus rapides.
- Cela a permis aux chercheurs de créer des algorithmes d'apprentissage automatique plus puissants capables de traiter et d'apprendre à partir d'ensembles de données plus importants.
- En outre, le développement de bibliothèques de logiciels open source telles que TensorFlow et PyTorch a permis aux chercheurs de mettre en œuvre plus facilement leurs idées et de les partager avec d'autres chercheurs.
- La vitesse de traitement informatique et la capacité de mémoire spécifiques qui ont rendu possible l'apprentissage automatique étaient la capacité d'analyser des ensembles de données plus importants et de créer des modèles plus complexes.
- Cela a été rendu possible par l'invention de processeurs plus puissants, de périphériques de stockage plus rapides et d'algorithmes plus efficaces.
- La combinaison de ces avancées a permis aux chercheurs de créer des algorithmes d'apprentissage automatique plus puissants capables de traiter et d'apprendre à partir d'ensembles de données plus importants.

Applications d'apprentissage automatique

1. **Vision par ordinateur:** utilisé pour analyser des images et des vidéos afin de détecter des objets, des visages et d'autres caractéristiques.
2. **Traitement du langage naturel:** utilisé pour comprendre et répondre au langage humain.
3. **Robotique:** utilisé pour contrôler les robots et les véhicules autonomes.
4. **Analyse financière:** utilisée pour prédire les cours des actions et d'autres tendances financières.
5. **Systèmes de recommandation :** utilisés pour fournir des recommandations personnalisées aux utilisateurs.
6. **Détection de fraude:** utilisé pour détecter les activités frauduleuses dans les transactions financières.
7. **Soins de santé:** utilisés pour analyser des images médicales et diagnostiquer des maladies.
8. **Reconnaissance vocale :** utilisée pour reconnaître la parole et répondre aux commandes vocales.
9. **Traitement d'image:** utilisé pour reconnaître des objets dans des images et détecter des objets dans une vidéo.
10. **Traduction automatique :** utilisée pour traduire du texte d'une langue à une autre.



Intelligence artificielle vs Machine Learning

- L'intelligence artificielle (IA) et l'apprentissage automatique (ML) sont des domaines étroitement liés de l'informatique.
- L'IA est le concept plus large de machines capables d'effectuer des tâches d'une manière que nous considérerions comme « intelligente ».
- Le ML est un type d'IA qui permet à une machine d'apprendre à partir de données sans être explicitement programmée.
- La principale différence entre l'IA et le ML est que l'IA se concentre sur le développement de machines intelligentes capables de penser et d'agir comme des humains, tandis que le ML se concentre sur le développement de programmes informatiques capables d'apprendre et de s'adapter à l'évolution des données.
- L'IA est généralement utilisée pour décrire des machines capables d'imiter le comportement humain et de résoudre des problèmes, tandis que le ML est utilisé pour décrire des machines capables d'apprendre de grandes quantités de données et d'identifier des modèles.
- Les algorithmes ML sont utilisés pour développer des applications d'IA telles que les voitures autonomes, la reconnaissance faciale et le traitement du langage naturel.

Machine Learning vs Deep Learning

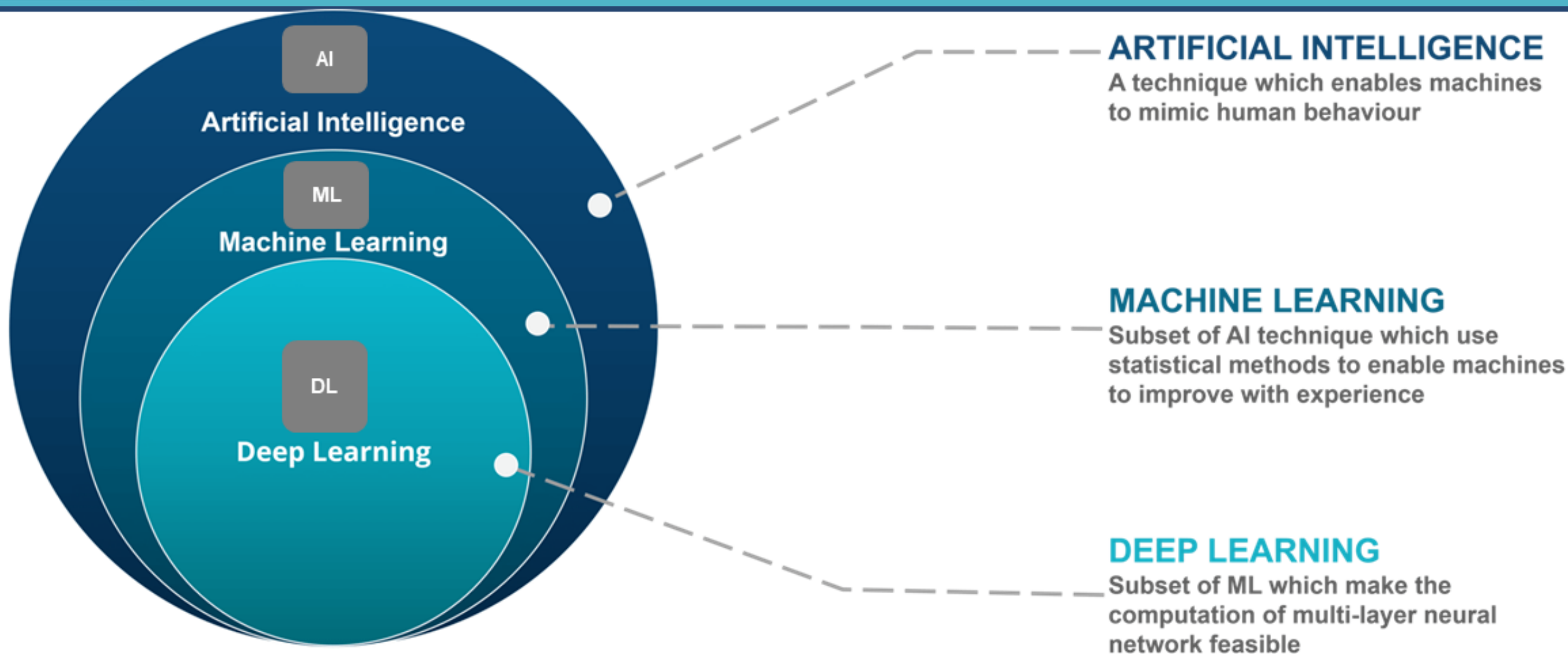
- La principale différence entre l'apprentissage automatique et l'apprentissage profond est que les algorithmes d'apprentissage automatique sont conçus pour apprendre à partir des données et identifier des modèles, tandis que les algorithmes d'apprentissage profond sont conçus pour apprendre des données et prendre des décisions.
- Les algorithmes d'apprentissage automatique sont utilisés pour développer des applications d'IA telles que la vision par ordinateur, le traitement du langage naturel, la robotique et l'analyse financière.
- Les algorithmes d'apprentissage profond sont utilisés pour développer des applications d'IA telles que les voitures autonomes, la reconnaissance faciale et le traitement du langage naturel.

Algorithmes d'apprentissage automatique

1. Arbres de décision : utilisés pour les tâches de classification et de régression.
2. Support Vector Machines (SVM) : utilisées pour les tâches de classification et de régression.
3. Naïve Bayes Classifier : utilisé pour les tâches de classification.
4. K-Nearest Neighbors (KNN) : utilisé pour les tâches de classification et de régression.
5. Forêt aléatoire : utilisée pour les tâches de classification et de régression.
6. Régression logistique : utilisée pour les tâches de classification.
7. Gradient Boosting: utilisé pour les tâches de classification et de régression.
8. Réseaux de neurones : utilisés pour les tâches de classification et de régression.
9. K-Means Clustering: utilisé pour les tâches d'apprentissage non supervisées.
10. Il existe de nombreux autres algorithmes d'apprentissage automatique tels que les modèles de Markov cachés (HMM), la maximisation des attentes (EM) et les réseaux de croyances profondes (DBN). Il existe également de nombreux autres algorithmes d'apprentissage profond tels que les machines de Boltzmann, les réseaux antagonistes génératifs (GAN) et l'apprentissage par renforcement profond.

Algorithmes d'apprentissage profond

1. Réseaux de neurones convolutifs (CNN) : utilisés pour analyser des images et des vidéos.
2. Réseaux neuronaux récurrents (RNN) : utilisés pour le traitement du langage naturel et la reconnaissance vocale.
3. Réseaux antagonistes génératifs (GAN) : utilisés pour générer des images et des vidéos.
4. Réseaux de mémoire à long terme (LSTM) : utilisés pour le traitement du langage naturel et la reconnaissance vocale.
5. Autoencodeurs : utilisés pour apprendre des représentations compressées de données.
6. Reinforcement Learning : utilisé pour l'entraînement des robots et des véhicules autonomes.



Artificial Intelligence

TRANSFORMA
INSIGHTS

Mimicking of
human
intelligence by
computers

Business rules
including
IFTTT

Machine Learning

Application of
statistical
techniques

Learn to
improve with
experience

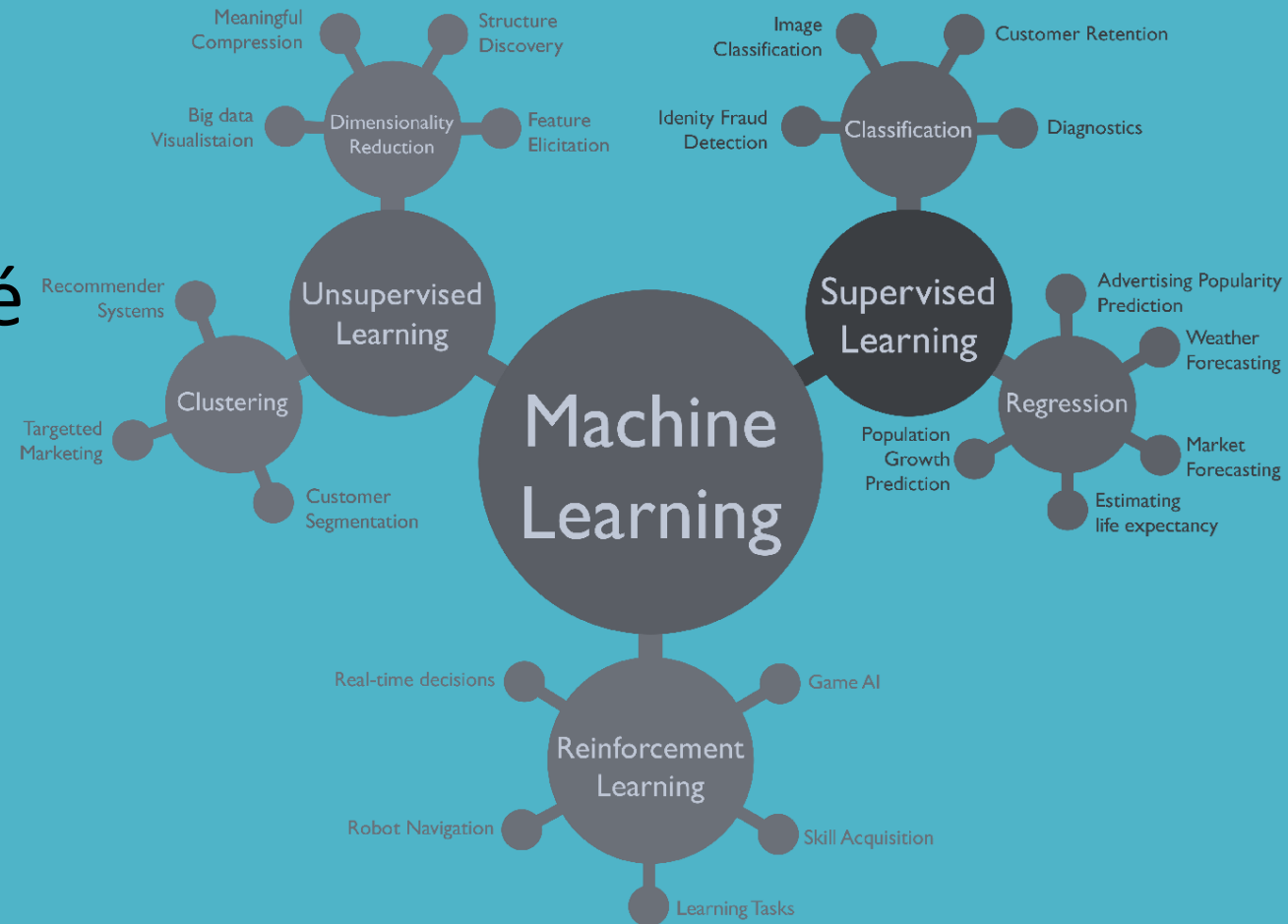
Deep Learning

Self-training software

Application of neural
networks

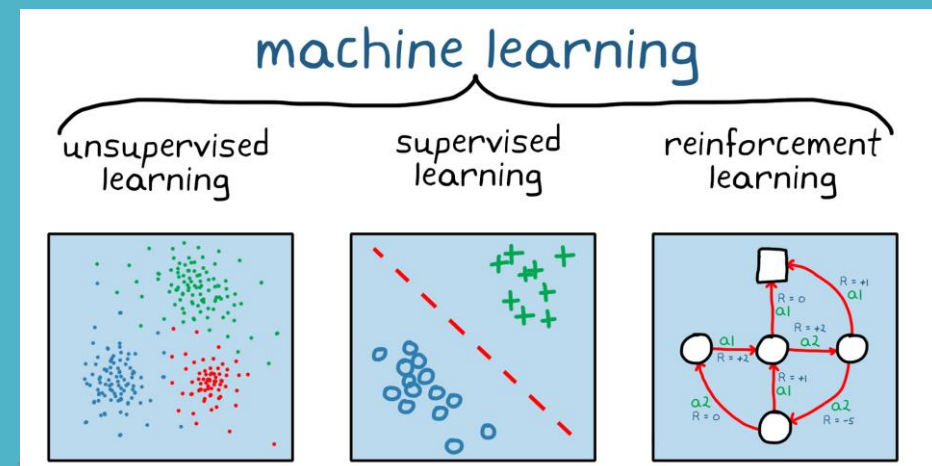
Types d'apprentissage automatique

1. Apprentissage supervisé
2. Apprentissage non supervisé
3. Apprentissage par renforcement
4. Apprentissage semi-supervisé



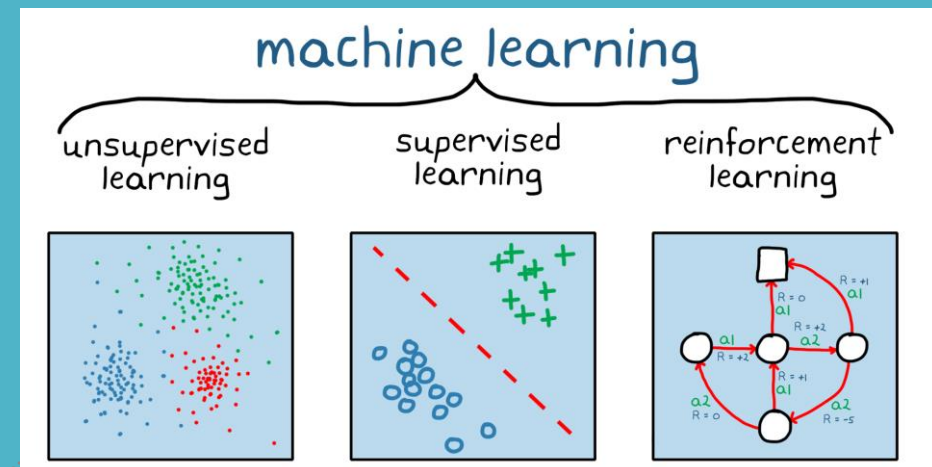
Apprentissage supervisé

- L'apprentissage supervisé implique la formation d'un algorithme d'apprentissage automatique avec des données étiquetées.
- L'algorithme est entraîné pour reconnaître les modèles dans les données et faire des prédictions basées sur ces modèles.
- Les algorithmes d'apprentissage supervisé courants incluent les arbres de décision, les machines à vecteurs de support, les classificateurs bayésiens naïfs, la régression logistique et les forêts aléatoires.



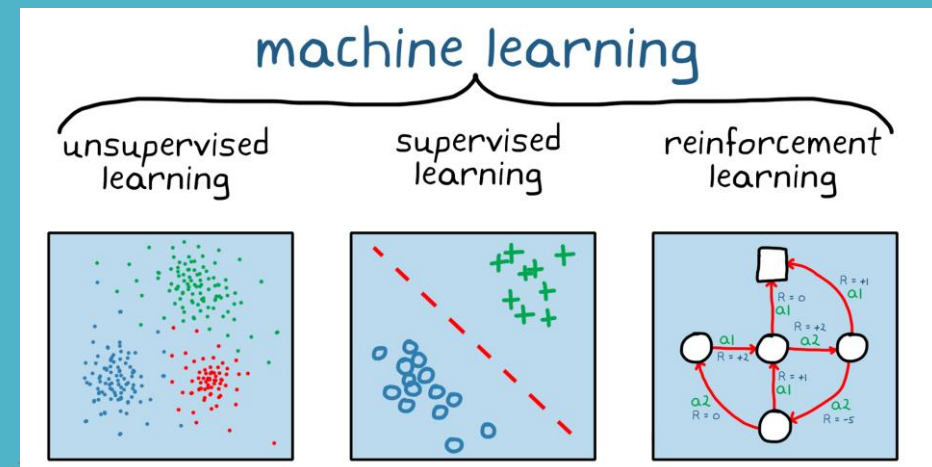
Apprentissage non supervisé

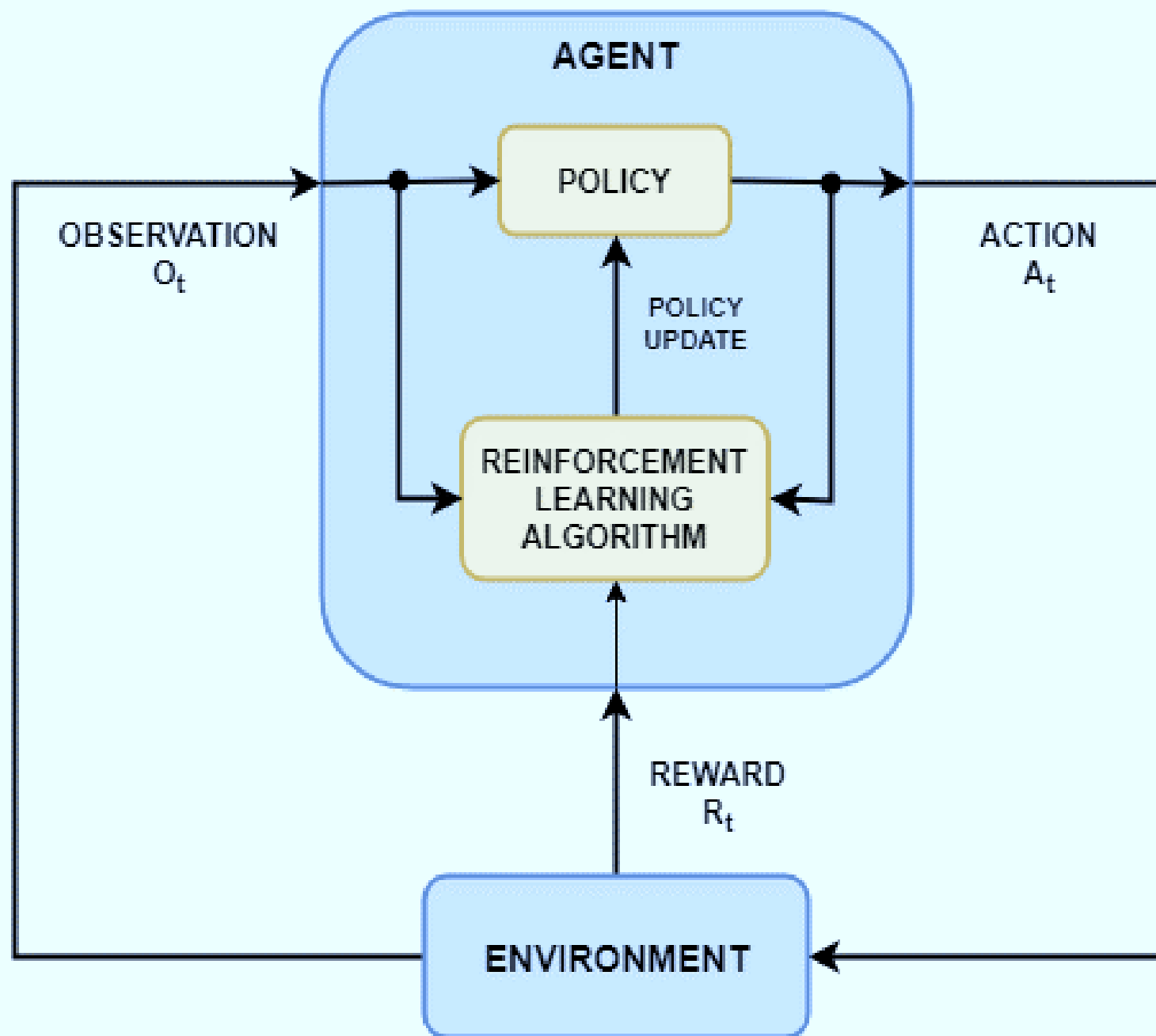
- L'apprentissage non supervisé implique l'entraînement d'un algorithme d'apprentissage automatique avec des données non étiquetées.
- L'algorithme est entraîné pour identifier des modèles dans les données sans aucune étiquette.
- Les algorithmes d'apprentissage non supervisé courants incluent le clustering K-means et le clustering hiérarchique.



Apprentissage par renforcement

- L'apprentissage par renforcement est un type d'apprentissage automatique où un agent est formé pour prendre des mesures dans un environnement afin de maximiser une récompense.
- Ce type d'apprentissage automatique est utilisé pour entraîner des robots et des véhicules autonomes.
- Les algorithmes courants d'apprentissage par renforcement incluent le Q-learning et la recherche d'arbres de Monte Carlo.





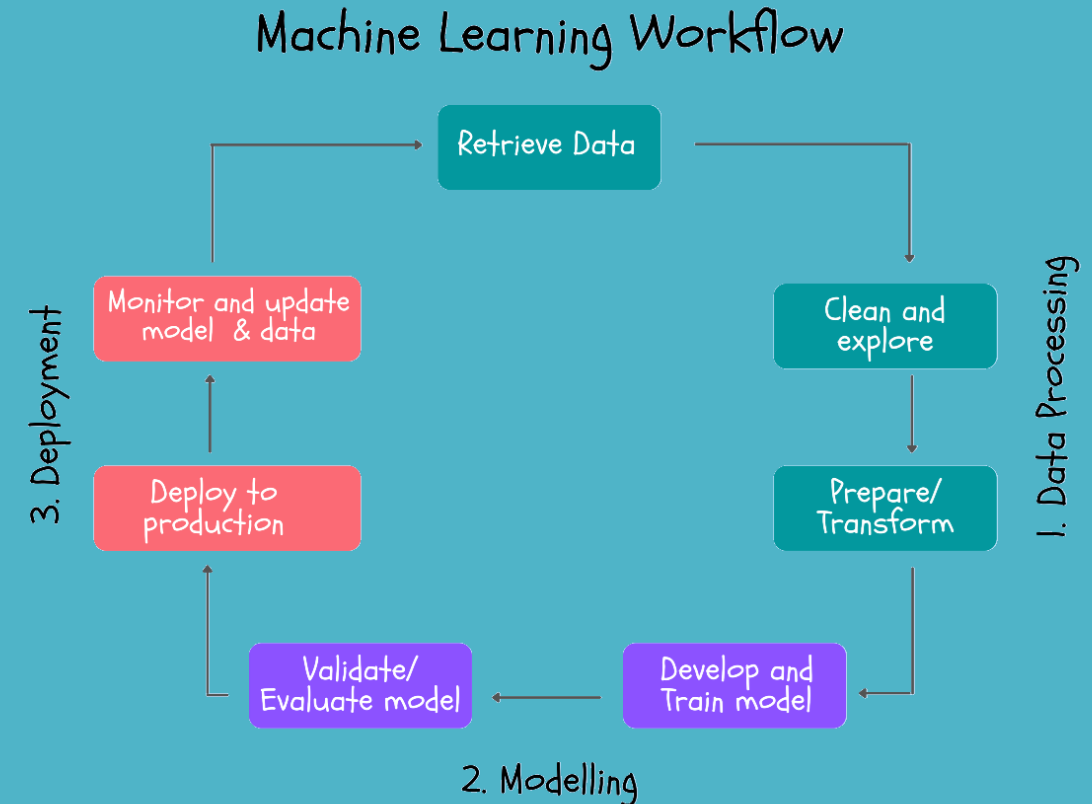
Apprentissage semi-supervisé

- L'apprentissage semi-supervisé est un type d'apprentissage automatique qui combine l'apprentissage supervisé et non supervisé.
- Dans ce type d'apprentissage automatique, l'algorithme est entraîné avec des données étiquetées et non étiquetées.
- Les données étiquetées sont utilisées pour entraîner l'algorithme à reconnaître les modèles dans les données, tandis que les données non étiquetées

Model	Learning task
Supervised Learning Algorithms	
Nearest Neighbor	Classification
Naive Bayes	Classification
Decision Trees	Classification
Classification Rule Learners	Classification
Linear Regression	Numeric prediction
Regression Trees	Numeric prediction
Model Trees	Numeric prediction
Neural Networks	Dual use
Support Vector Machines	Dual use
Unsupervised Learning Algorithms	
Association Rules	Pattern detection
k-means clustering	Clustering
Meta-Learning Algorithms	
Bagging	Dual use
Boosting	Dual use
Random Forests	Dual use

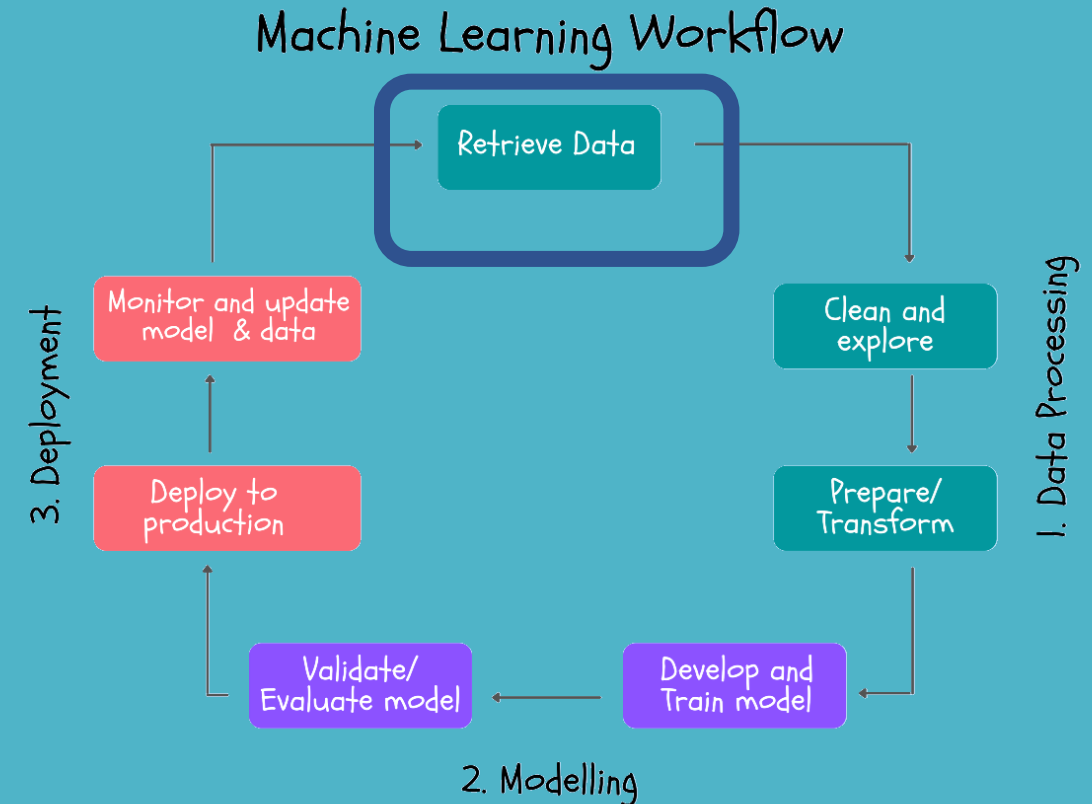
Flux de travail d'apprentissage automatique

- **Collecte de données** : collecte de données provenant de diverses sources.
- **Prétraitement des données** : nettoyage, transformation et normalisation des données.
- **Model Training** : formation du modèle sur les données traitées.
- **Évaluation du modèle** : évaluation du modèle sur les données de test.
- **Déploiement de modèle** : déploiement du modèle pour une utilisation dans un environnement de production.
- **Surveillance du modèle** : surveillance de la précision et des performances du modèle.



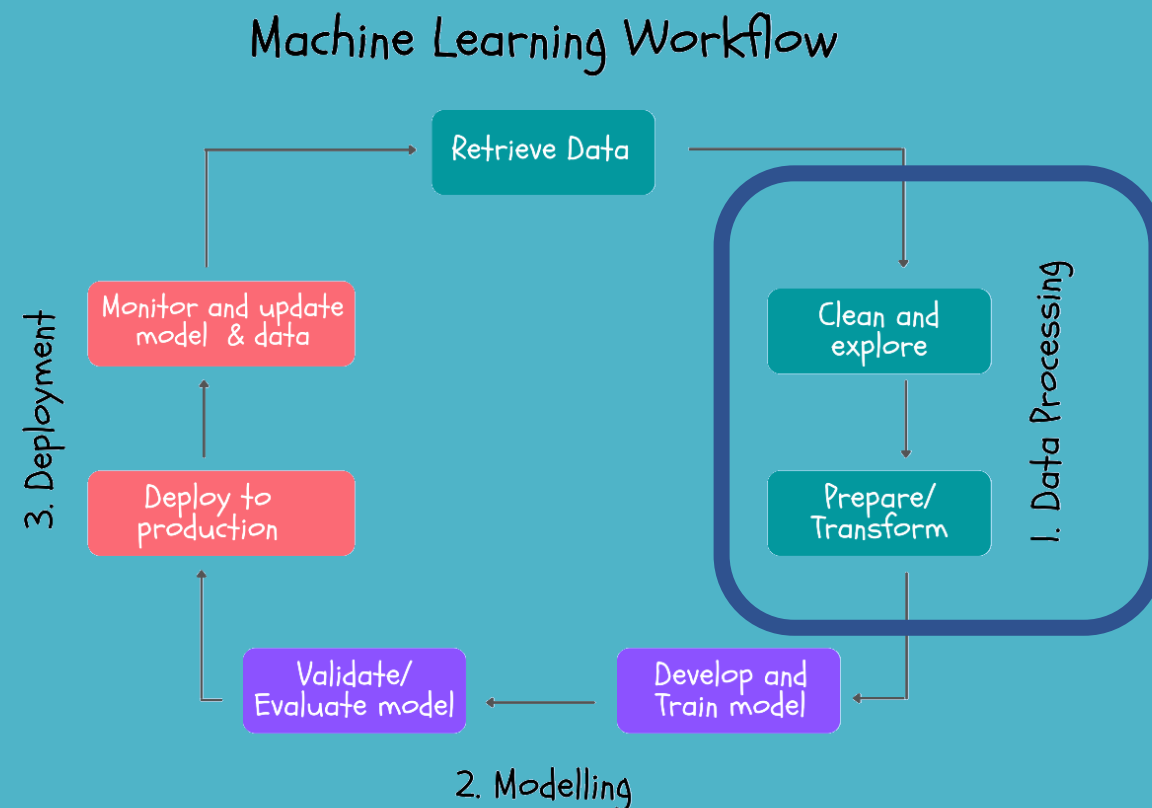
Machine learning : collecte de données

- La collecte de données à partir de diverses sources est la première étape du flux de travail d'apprentissage automatique.
- Cela implique la collecte de données à partir de bases de données, d'API Web et d'autres sources.
- Il est important de s'assurer que les données sont complètes, exactes et pertinentes pour la tâche à accomplir.



Machine learning : Prétraitement des données

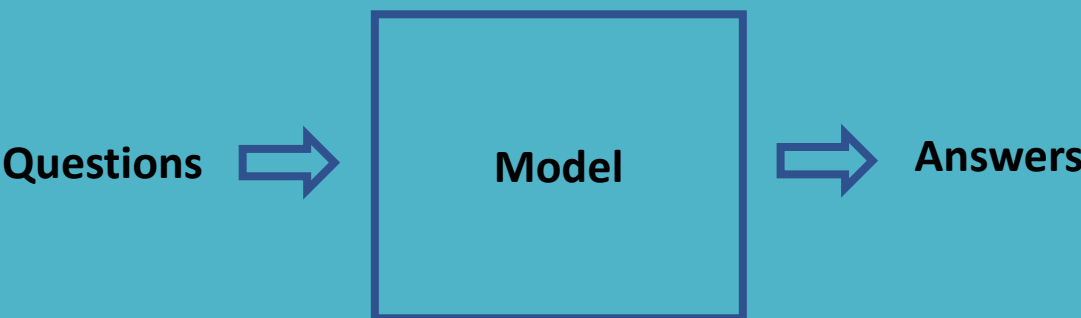
- Nettoyez, transformez et normalisez les données.
- Cela implique de supprimer les valeurs aberrantes, de transformer les variables catégorielles en variables numériques et de mettre à l'échelle les données.
- Il est également important de diviser les données en ensembles d'apprentissage, de validation et de test.



Phase 1: Training – deriving a model

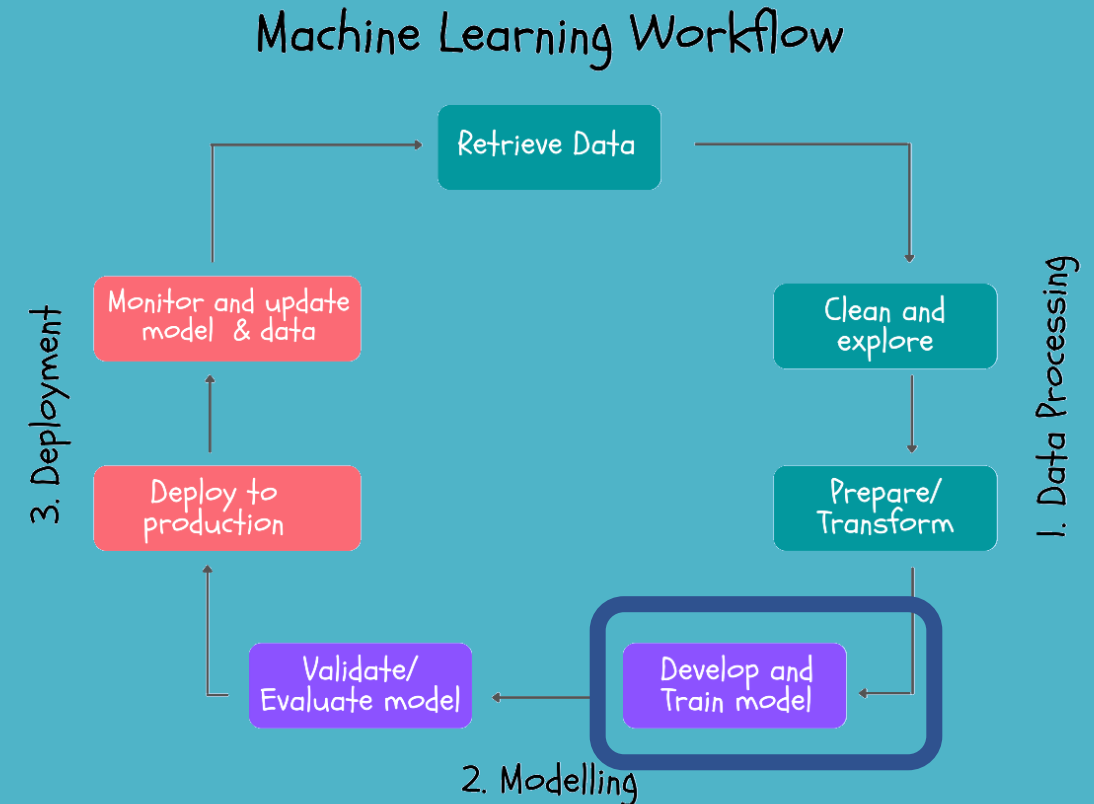


Phase 2: Inference – using the model



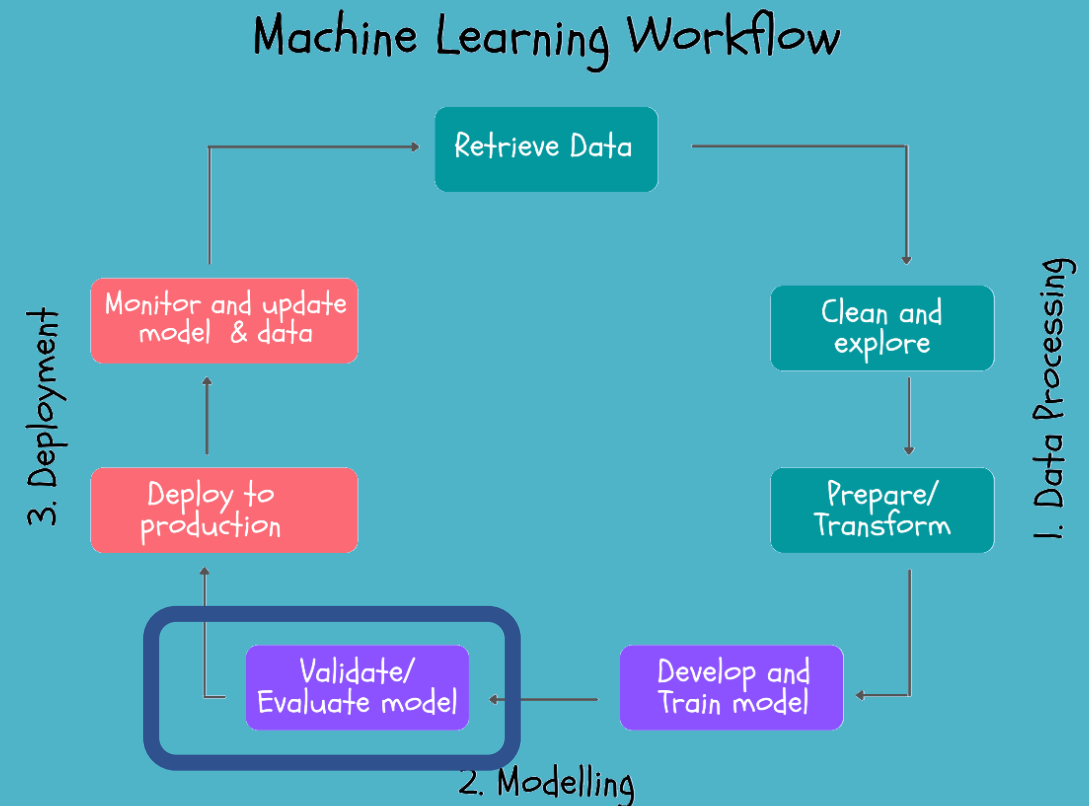
Machine learning : formation de modèles

- Après le prétraitement des données, le modèle est formé sur les données traitées.
- Cela implique d'ajuster le modèle aux données et d'ajuster les paramètres.
- Il est important de choisir le bon modèle pour la tâche à accomplir et de s'assurer qu'il ne surajuste pas ou ne sous-ajuste pas les données.



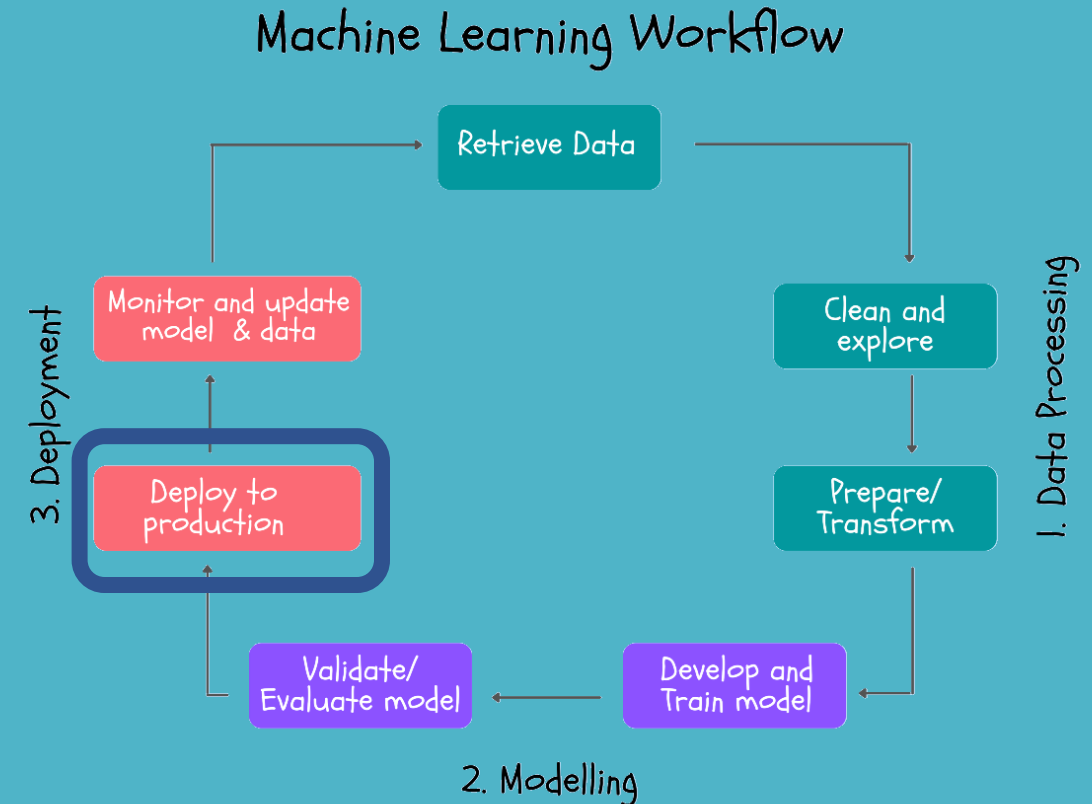
Machine learning : évaluation de modèle

- Après l'entraînement du modèle, il est évalué sur les données de test.
- Cela implique de mesurer la précision et la performance du modèle et d'apporter les ajustements nécessaires.
- Les mesures courantes pour évaluer les modèles d'apprentissage automatique incluent l'exactitude, la précision, le rappel et le score F1.
- Collecte de données : collecte de données auprès de



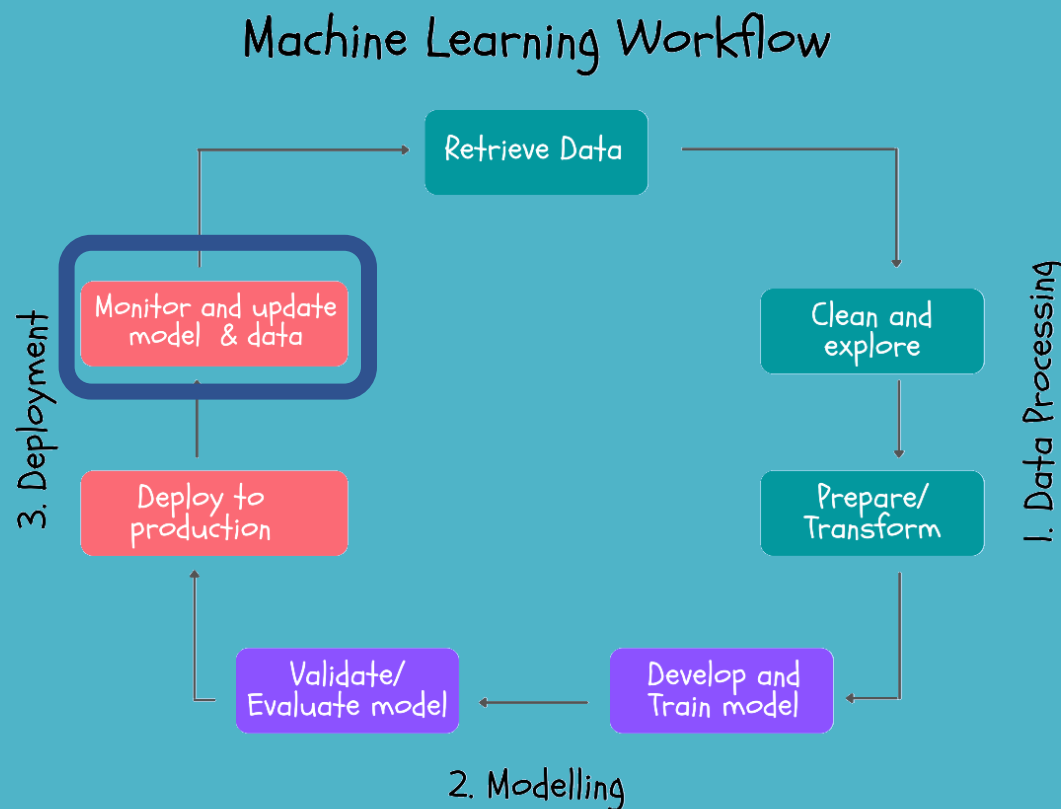
Machine learning : déploiement de modèles

- Le modèle est ensuite déployé dans un environnement de production, ce qui lui permet d'être utilisé en continu dans des applications réelles.
- Il est important de tenir compte du coût de la formation et du déploiement du modèle, ainsi que des implications en matière de confidentialité et de sécurité des données.



Machine learning : Surveillance des modèles

- Le modèle est ensuite surveillé pour la précision et la performance.
- Cela implique de mettre à jour le modèle au besoin et de suivre ses performances au fil du temps.
- Les mesures courantes pour la surveillance des modèles d'apprentissage automatique incluent l'exactitude, la précision, le rappel et le score F1



k-NN

Apprentissage automatique : k-NN

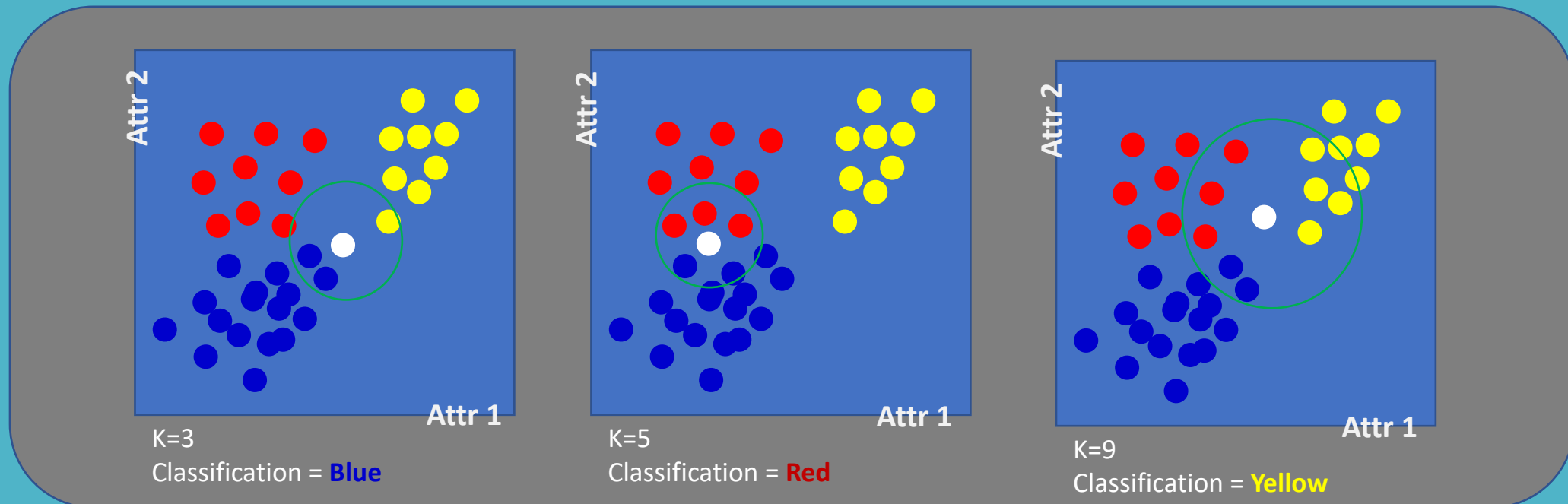
- Classer les échantillons non étiquetés en les affectant à une classe d'échantillons étiquetés similaires
- La similarité est déterminée par l'une des nombreuses mesures de distance
- Le k dans kNN est le nombre de voisins auxquels l'échantillon est comparé
- En règle générale, un vote majoritaire détermine à quelle classe appartient l'échantillon.
- Apprenant paresseux : l'algorithme kNN classe sans apprendre grand-chose sur la répartition des données dans le jeu de données. C'est le contraire des apprenants enthousiastes
- kNN est une méthode de classification basée sur les instances : le calcul est retardé jusqu'à la classification
- La classification kNN est un algorithme de classification d'apprentissage supervisé
- La régression kNN n'est pas la même chose que la classification kNN
- La régression kNN fait la moyenne des valeurs des k plus proches voisins au lieu de prendre un vote majoritaire
- Ils sont connus sous le nom de régression pondérée localement

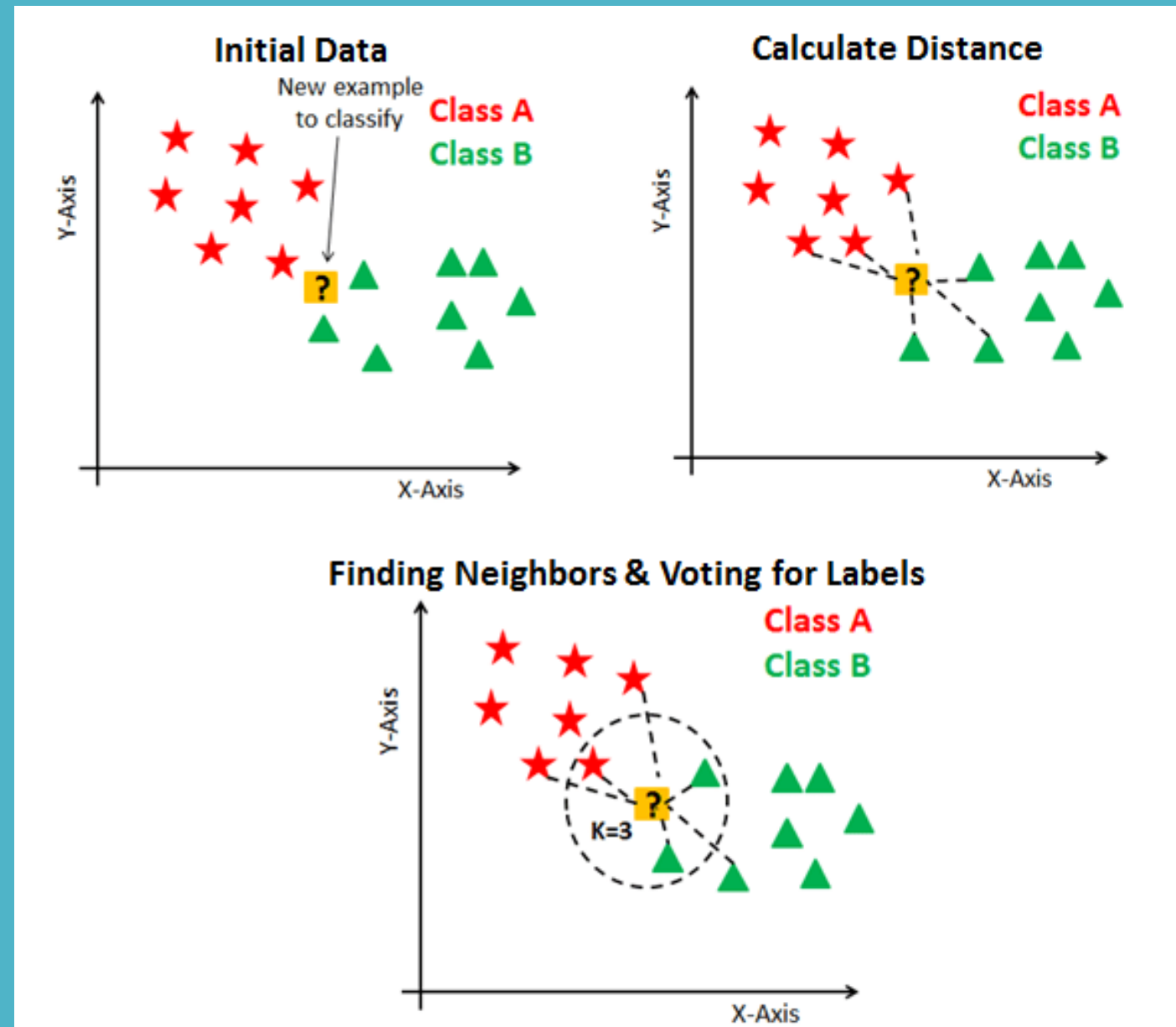
k-NN : terminologie

- k : nombre de voisins dans la classification
- Instances : nombre d'observations, échantillons utilisés dans la classification (formation et essais)
- Attributs : nombre de caractéristiques, paramètres utilisés pour comparer la distance entre les échantillons
- Mesure de la distance : méthode de calcul de la distance entre les échantillons, par exemple distance euclidienne
- Elbow Method : technique utilisée pour déterminer le k optimal pour un ensemble de données donné

k-NN : visualisation

- Dans cette illustration, le cercle vert capture les k plus proches voisins. Ensuite, la classification se fait à la majorité des voix. Quelle que soit la classe ayant la majorité au sein du cercle, remporte le vote.





k-NN : avantages et inconvénients

- **Avantages:**

- kNN est robuste même avec des ensembles de données bruyants
- Efficacité relative avec de grands ensembles de données
- L'algorithme est exécuté en un seul passage, chaque échantillon est évalué une fois
- kNN peut modéliser facilement des modèles relativement complexes

- **Inconvénients:**

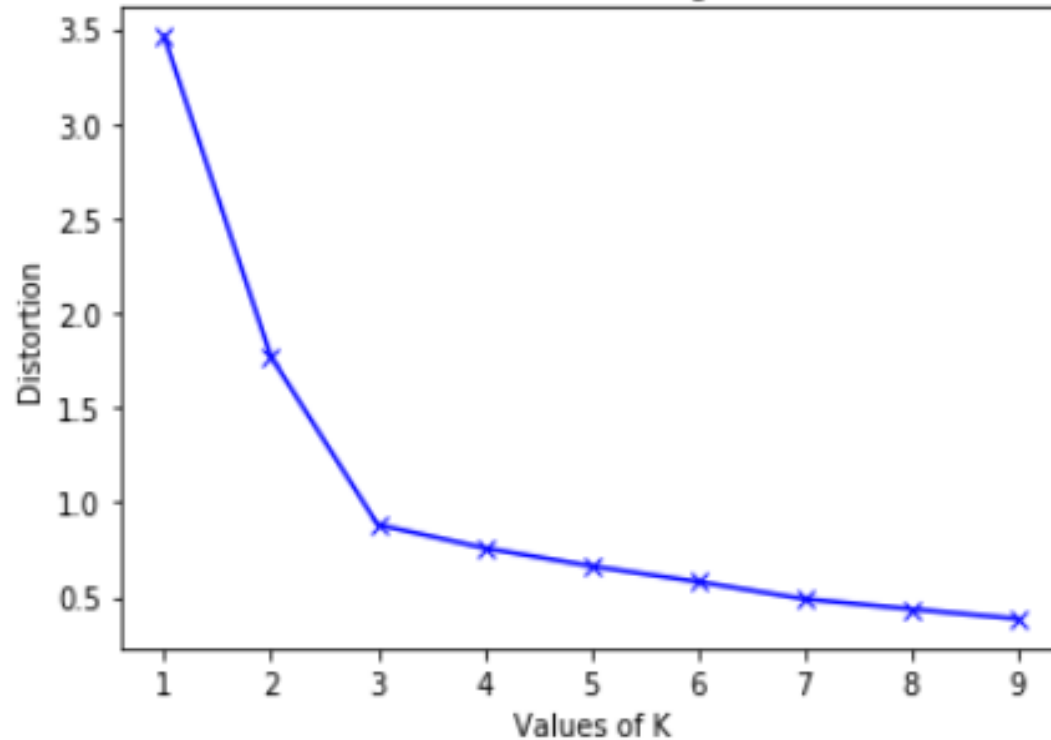
- Trouver le k optimal n'est pas systématique
- Calculs de distance par « force brute » : chaque échantillon est comparé à toutes les données du jeu de données d'apprentissage
- Les grands ensembles de données peuvent causer des problèmes de performance en ce qui concerne le coût de calcul, bien que la précision finira par être relativement élevée
- Les valeurs aberrantes peuvent entraîner une augmentation des erreurs de classification

k-NN : sélection de k

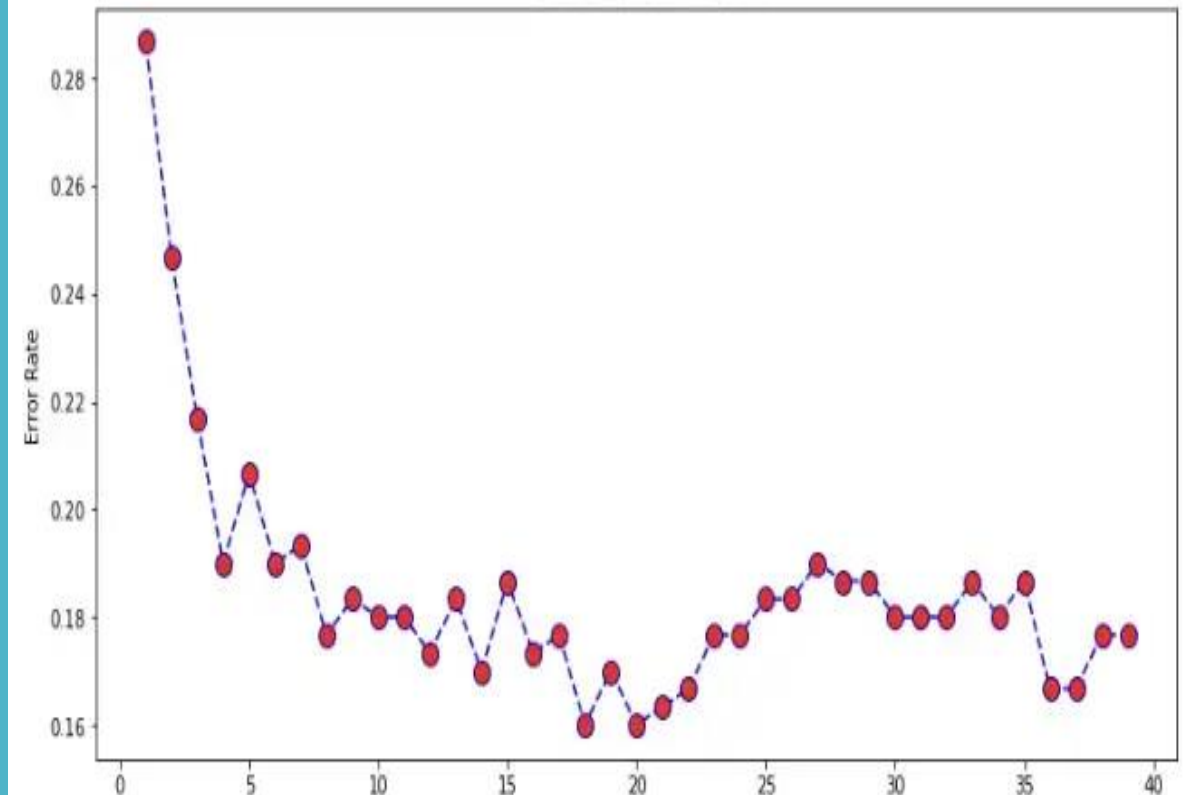
- Règle empirique : choisissez k à peu près égal à la racine carrée du nombre de classes
- k devrait être un nombre impair pour éviter l'égalité des voix dans le vote majoritaire
- k ne doit pas être un multiple du nombre de classes pour éviter les liens N-way ($N = \#$ classes)
- k ne peut pas être trop petit ou trop grand
- K Trop petit : problèmes de surajustement
- k trop grand:
- Problèmes d'ajustement dus à l'apprentissage de la distribution locale des données
- Coût de calcul élevé du calcul des distances pour chaque paire d'instances possible
- Les performances de classification diminuent à mesure que k et la taille du jeu de données augmentent

K-NN: elbow method

The Elbow Method using Distortion

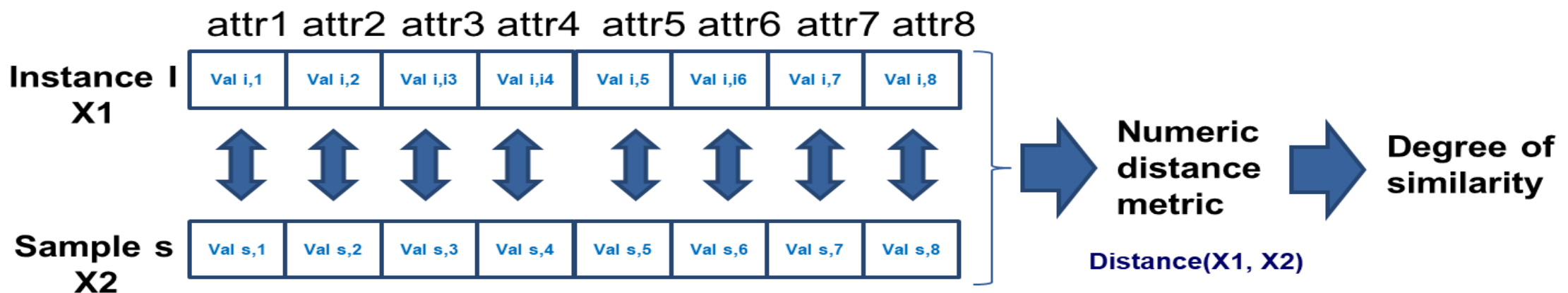


Error Rate vs. K Value



K-NN : concept de similitude

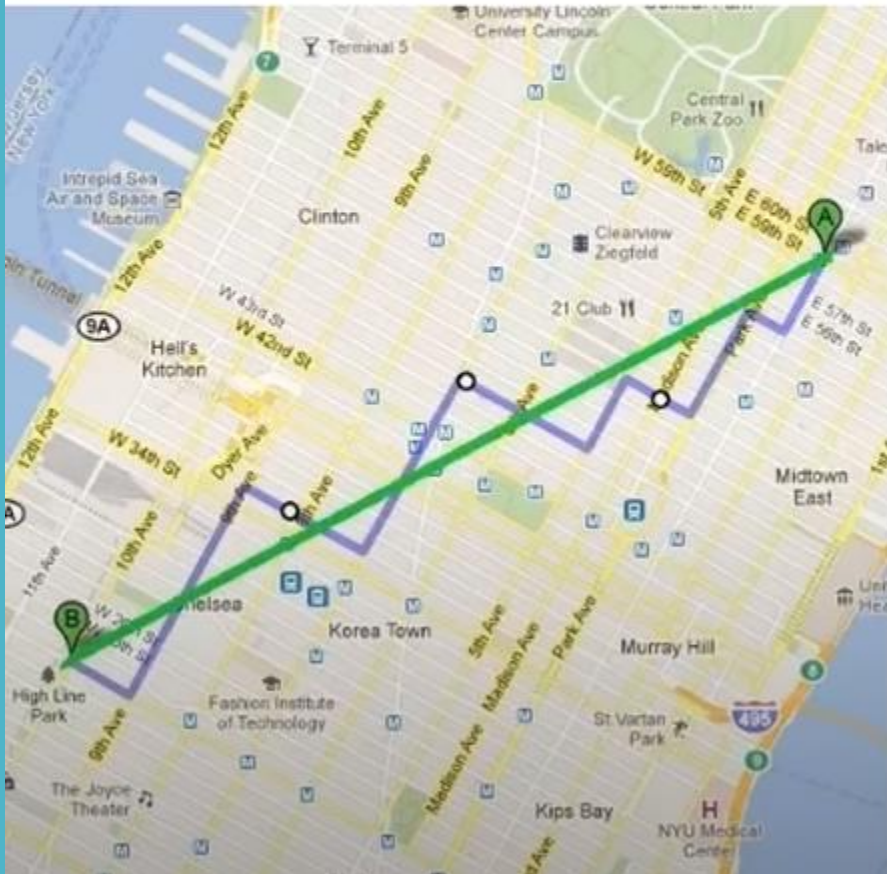
- La similitude entre les données d'échantillon et d'entraînement en kNN est calculée en appliquant une métrique de distance
- Une façon de visualiser le concept est de penser à chaque instance de l'ensemble d'apprentissage comme un vecteur de valeurs numériques $val_{i,j}$ de chaque attribut $attr_j$
- Le calcul de la similarité entre l'échantillon et les instances revient à calculer les distances en N dimensions entre ces vecteurs



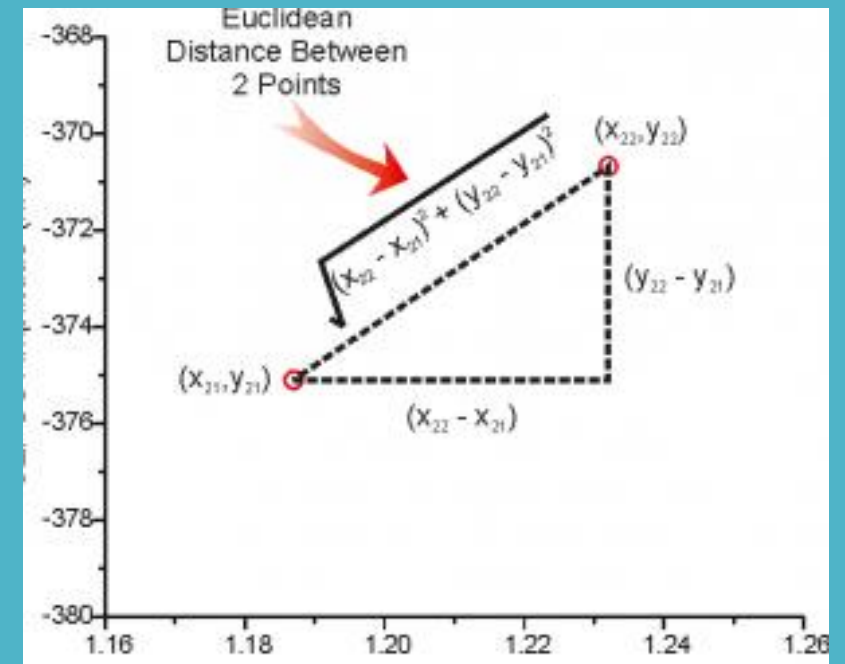
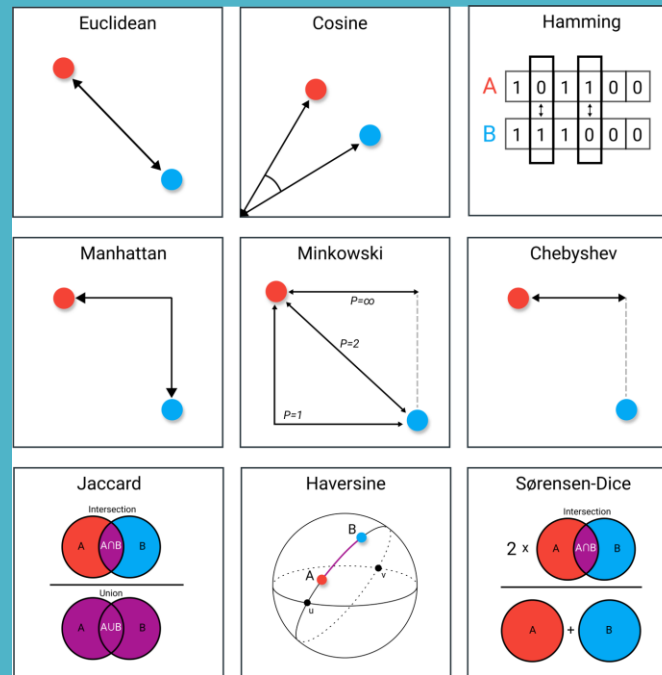
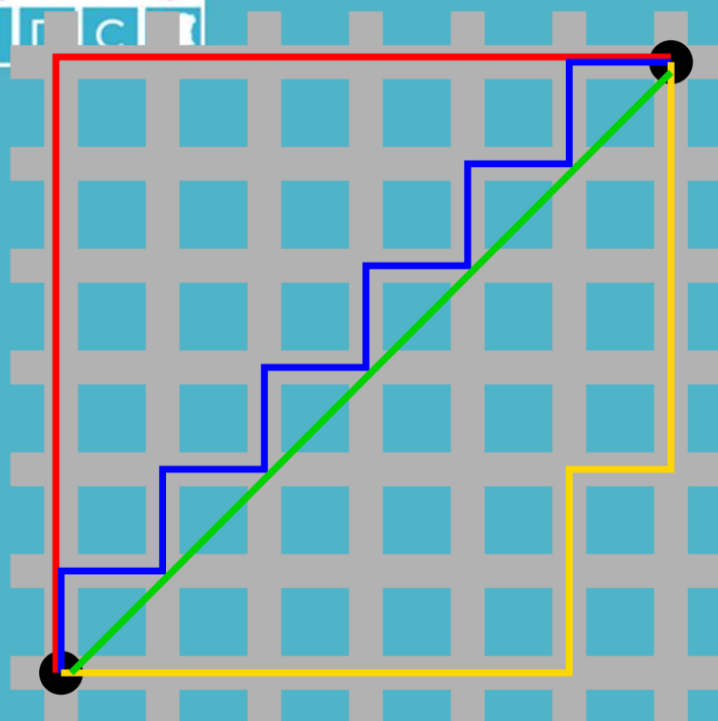
k-NN : mesures de distance

- Distance euclidienne: distance basée sur la formule de distance pythagoricienne dans un système de coordonnées cartésiennes. Couramment utilisé dans la classification kNN
- Distance de Manhattan: également connue sous le nom de distance de pâté de maisons ou géométrie de taxi
- Distance de Minkowski: formule de distance généralisée dans laquelle les cas spéciaux $p = 2$ est la distance euclidienne et $p = 1$ est la distance de Manhattan
- Distance de Hamming: nombre de positions par lesquelles deux vecteurs de longueur égale diffèrent. Très populaire en théorie des codes
- Distance cosinus : mesure la similitude entre deux vecteurs non nuls d'un espace produit interne.
- Indépendamment de leur magnitude vectorielle, la similitude cosinus est égale à
- 1 si deux vecteurs sont orientés dans la même direction
- -1 si deux vecteurs sont orientés dans des directions opposées
- 0 si deux vecteurs sont perpendiculaires

Manhattan Distance VS Euclidean Distance



- La distance euclidienne est la métrique de distance par défaut et la plus populaire utilisée dans la classification k-NN.
- D'autres métriques, comme la distance Manhattan, doivent être prises en compte pour voir si leur utilisation améliore le taux de précision de la classification (réduit le taux d'erreur)



Some frequently used distance functions.	
<p>Camberra :</p> $d(x, y) = \sum_{i=1}^m \frac{ x_i - y_i }{ x_i + y_i } \quad (2)$	<p>Euclidean :</p> $d(x, y) = \sqrt{\sum_{i=1}^m (x_i - y_i)^2} \quad (5)$
<p>Minkowsky :</p> $d(x, y) = \left(\sum_{i=1}^m x_i - y_i ^r \right)^{1/r} \quad (3)$	<p>Manhattan / city - block :</p> $d(x, y) = \sum_{i=1}^m x_i - y_i \quad (6)$
<p>Chebychev :</p> $d(x, y) = \max_{i=1}^m x_i - y_i \quad (4)$	

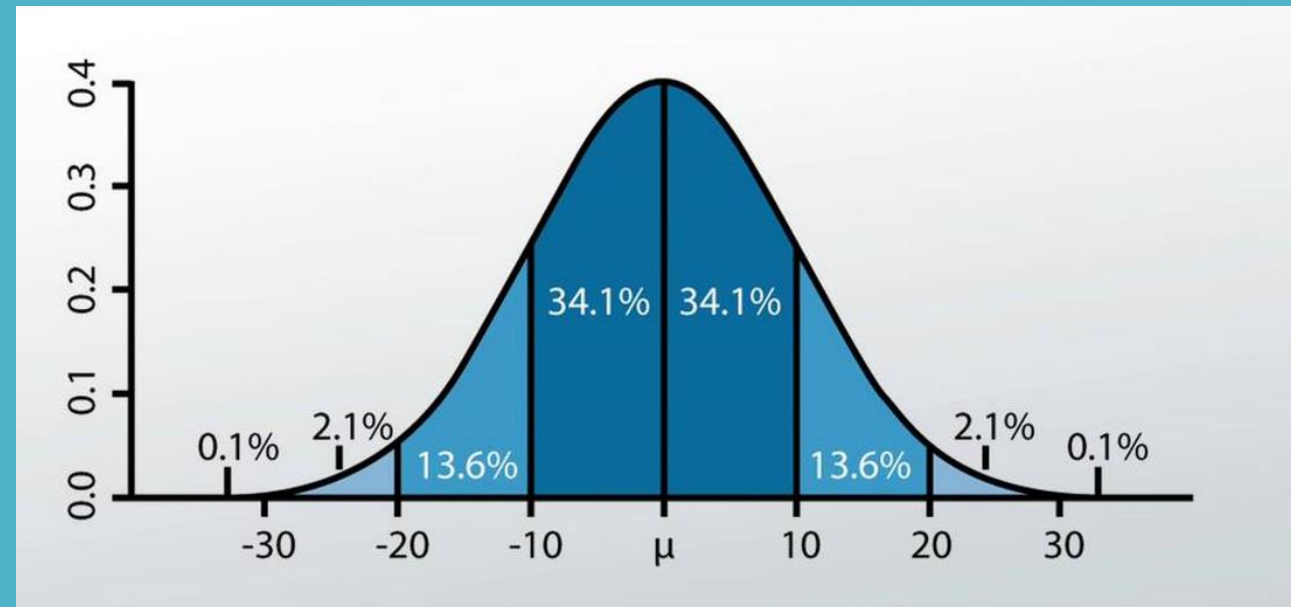
Normalisation des données

- Pourquoi normaliser les données ?
- Les classificateurs calculent la distance entre deux entités (attributs) à l'aide d'une métrique de distance.
- Si l'une des entités a une large gamme de valeurs et/ou de nombreuses valeurs aberrantes, la distance sera dominée par cette caractéristique, ce qui affectera les performances du classificateur.
- La normalisation des valeurs des caractéristiques garantit que chaque entité contribue proportionnellement aux calculs de distance
- Les méthodes de normalisation comprennent la normalisation du score z , la normalisation min-max, la normalisation moyenne et la mise à l'échelle à la longueur unitaire.

Normalisation du Z-score

$$X_{new} = \frac{X - \mu}{\sigma} = \frac{X - \text{Mean}(X)}{\text{StdDev}(X)}$$

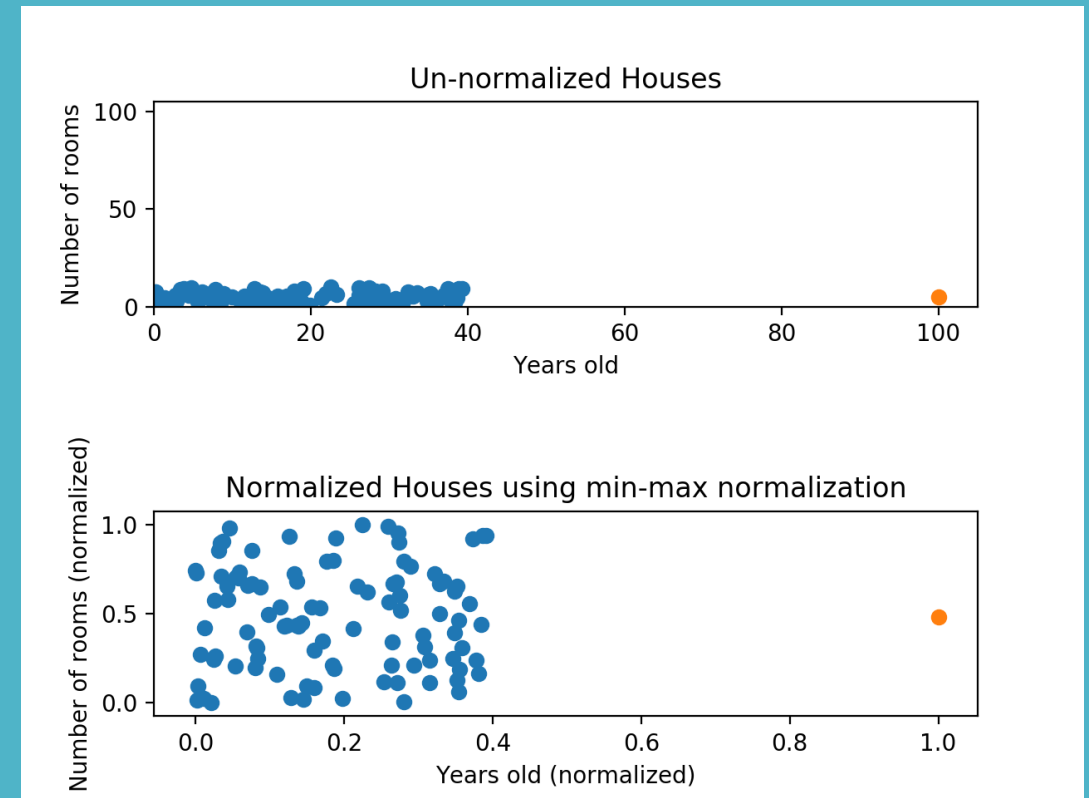
- Normalisation du score Z: Pour chaque caractéristique, la moyenne de distribution et l'écart-type sont calculés. La moyenne est soustraite de chaque caractéristique. Ensuite, le résultat est divisé par son écart-type
- Cette transformation conduit à des valeurs centrées autour de chaque caractéristique moyenne
- Les valeurs peuvent, en théorie, aller de $-\infty$ à $+\infty$. Cependant, pour les données normalement distribuées, près de 100 % des valeurs se situeront à moins de 3 écarts-types de la moyenne



Normalisation Min-Max

- Normalisation Min-Max : pour chaque entité (attribut), la valeur minimale de cette caractéristique est transformée en 0, la valeur maximale en 1 et toutes les autres valeurs sont transformées en décimales comprises entre 0 et 1.

$$X_{new} = \frac{X - \min(X)}{\max(X) - \min(X)}$$



Naive Bayes

Naïve Bayes

- Naive Bayes est un algorithme de classification probabiliste basé sur le théorème de Bayes, qui stipule que la probabilité d'une hypothèse (dans ce cas, une étiquette de classe) étant donné certaines observations (dans ce cas, les valeurs de caractéristique) est égale à la probabilité des observations étant donné l'hypothèse, multipliée par la probabilité préalable de l'hypothèse, divisée par la probabilité des observations. En d'autres termes,
- $P(\text{classe} | \text{données}) = (P(\text{données} | \text{classe}) * P(\text{classe})) / P(\text{données})$
- Naive Bayes est naïf parce qu'il fait une hypothèse forte et irréaliste sur l'indépendance des caractéristiques, ce qui signifie que la présence ou l'absence d'une caractéristique particulière dans les données n'est pas liée à la présence ou à l'absence de toute autre caractéristique.
- Malgré cette hypothèse, l'algorithme fonctionne souvent bien dans la pratique, en particulier lorsque le nombre de fonctionnalités est important.

Naïve Bayes Types

- Il existe plusieurs variantes de Bayes naïfs, notamment les bayes naïfs gaussiens, les bayésiens naïfs multinomiaux et les bayésiens naïfs de Bernoulli.
- **Bayes naïf gaussien:** suppose que les données de chaque entité sont normalement distribuées. Il est utilisé pour les données continues.
- **Bayes naïf multinomial:** suppose que les données de chaque entité sont distribuées de manière multinomiale. Il est utilisé pour les données discrètes telles que la classification de texte.
- **Bernoulli Naive Bayes:** suppose que les données de chaque fonctionnalité sont distribuées par Bernoulli. Il est utilisé pour les données binaires.

Naïve Bayes entraînement

- Le processus d'entraînement pour Naive Bayes implique l'estimation des paramètres des distributions de probabilité pour chaque caractéristique et classe, généralement en utilisant l'estimation du maximum de vraisemblance.
- Une fois le modèle formé, faire des prédictions pour les nouvelles instances est relativement efficace, car il s'agit simplement de calculer les probabilités conditionnelles pour chaque classe en fonction des valeurs de fonctionnalité de la nouvelle instance et de choisir la classe avec la probabilité la plus élevée.

Naive Bayes

	Spam		No spam	
Total	25		75	
Buy	20		5	
Cheap	15		10	
Buy & Cheap	12		2/3	

Naive Bayes

$$P(\text{"Buy"} \ \& \ \text{"Cheap"}) = P(\text{"Buy"}) \ P(\text{"Cheap"})$$

Bayes Theorem

S: Spam

H: Ham (not spam)

B: 'Buy'

$$P(S | B) = \frac{P(B | S) P(S)}{P(B | S) P(S) + P(B | H) P(H)}$$

S: Spam

H: Ham (not spam)

B: 'Buy'

C: 'Cheap'

Naive Bayes

$$P(S | B \cap C) = \frac{P(B|S)P(C|S)P(S)}{P(B|S)P(C|S)P(S) + P(B|H)P(C|H)P(H)}$$

Bayes Theorem

S: Spam

H: Ham (not spam)

B: 'Buy'

$$P(S | B) = \frac{P(B | S) P(S)}{P(B | S) P(S) + P(B | H) P(H)}$$

$$P(\text{spam if "Buy"}) = \frac{\frac{20}{25} \frac{25}{100}}{\frac{20}{25} \frac{25}{100} + \frac{5}{75} \frac{75}{100}} = 80\%$$

S: Spam

H: Ham (not spam)

B: 'Buy'

C: 'Cheap'

Naive Bayes

$$P(S | B \cap C) = \frac{P(B|S)P(C|S)P(S)}{P(B|S)P(C|S)P(S) + P(B|H)P(C|H)P(H)}$$

$$\begin{aligned} P(\text{spam if "Buy" \& "Cheap"}) &= \frac{\frac{20}{25} \frac{15}{25} \frac{25}{100}}{\frac{20}{25} \frac{15}{25} \frac{25}{100} + \frac{5}{75} \frac{10}{75} \frac{75}{100}} \\ &= 94.737\% \end{aligned}$$

Avec trois

Naive Bayes

	Spam		No Spam	
Total	25		75	
Buy	20	4/5	5	1/15
Cheap	15	3/5	10	2/15
Work	5	1/5	30	6/15
Buy, Cheap, & Work		12/125		

Naive Bayes

	Spam		No Spam	
Total	25		75	
Buy	20	$4/5$	5	$1/15$
Cheap	15	$3/5$	10	$2/15$
Work	5	$1/5$	30	$6/15$
Buy, Cheap, & Work	$12/5$	$12/125$		

Naive Bayes

	Spam		No Spam	
Total	25		75	
Buy	20	$4/5$	5	$1/15$
Cheap	15	$3/5$	10	$2/15$
Work	5	$1/5$	30	$6/15$
Buy, Cheap, & Work	$12/5$	$12/125$	$4/15$	$12/3375$

Naive Bayes

	Spam		No Spam	
Total	25		75	
Buy	20	4/5	5	1/15
Cheap	15	3/5	10	2/15
Work	5	1/5	30	6/15
Buy, Cheap, & Work	12/5	12/125	4/15	12/3375

$$\frac{12/5}{12/5 + 4/15} = \frac{36}{40} = 90\%$$

S: Spam

H: Ham (not spam)

B: 'Buy'

C: 'Cheap'

Naive Bayes

$$P(S | B \cap C) = \frac{P(B|S)P(C|S)P(S)}{P(B|S)P(C|S)P(S) + P(B|H)P(C|H)P(H)}$$

$$P(\text{spam if "Buy" \& "Cheap"}) = \frac{\frac{20}{25} \frac{15}{25} \frac{25}{100}}{\frac{20}{25} \frac{15}{25} \frac{25}{100} + \frac{5}{75} \frac{10}{75} \frac{75}{100}}$$

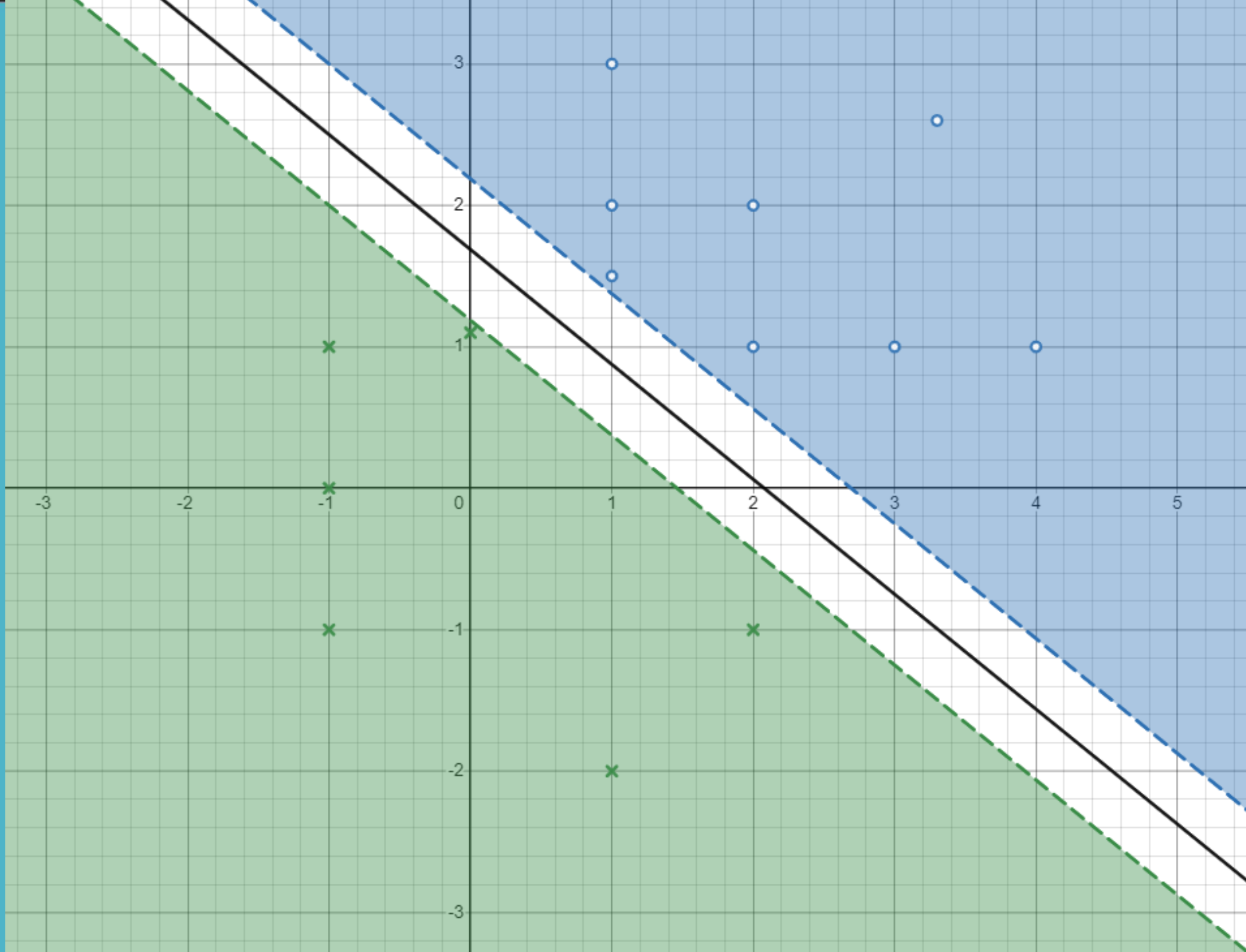
Support Vector Machines

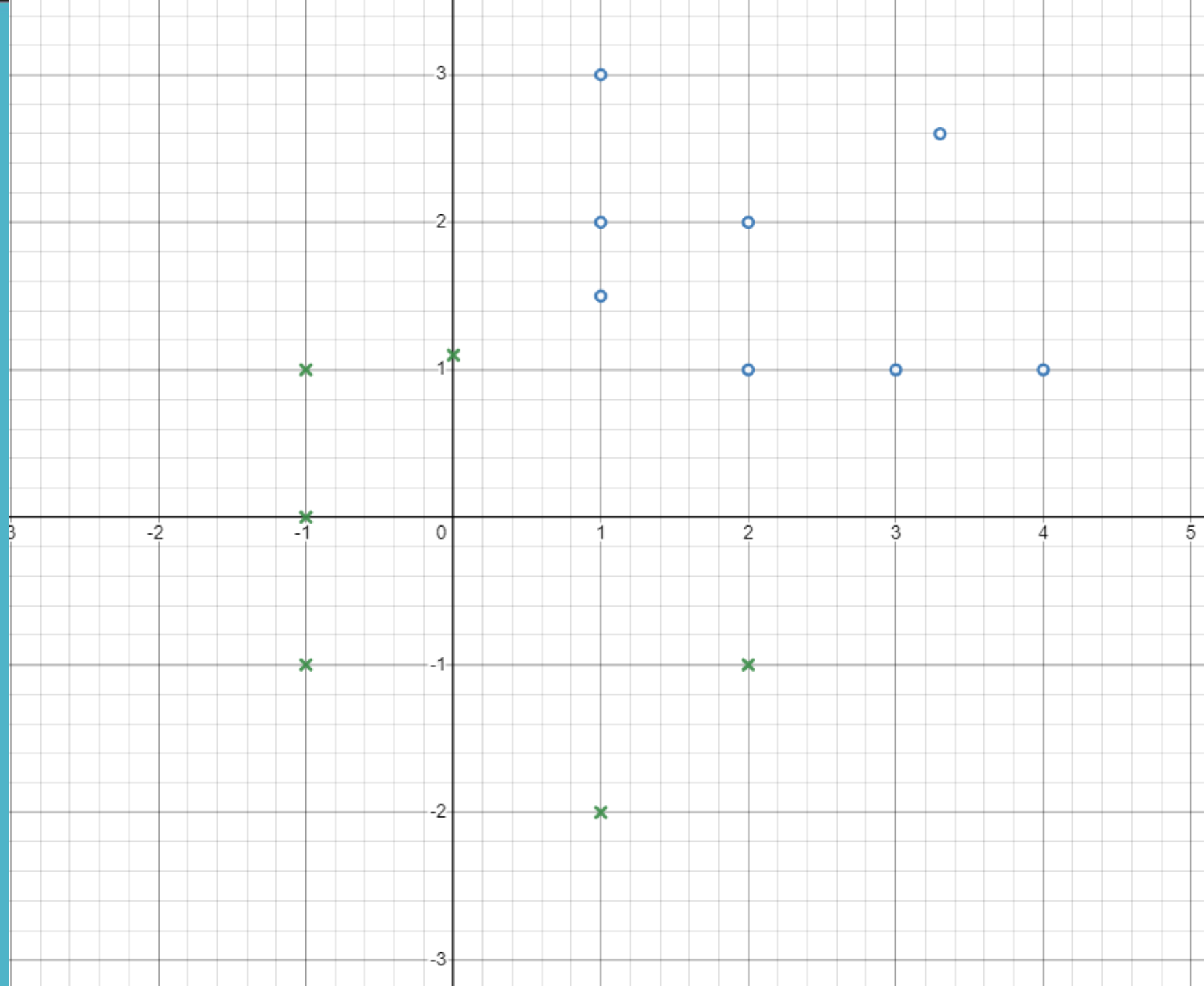
SVM Definition

- Les machines à vecteurs de support (SVM) sont un type d'algorithme d'apprentissage supervisé qui peut être utilisé pour des tâches de classification ou de régression.
- L'idée de base derrière les SVM est de trouver un hyperplan qui sépare au maximum les différentes classes dans les données.

SVM linéaire

- La première étape de la création d'une SVM linéaire consiste à choisir une fonction du noyau.
- Le noyau le plus couramment utilisé est le noyau linéaire, qui calcule simplement le produit ponctuel entre les vecteurs d'entrée.
- D'autres fonctions du noyau, telles que les noyaux de fonctions de base polynomiale et radiale (RBF), peuvent également être utilisées.





Formation du modèle

- Once the kernel is selected, the SVM algorithm will find the hyperplane that maximally separates the different classes in the data.
- The hyperplane is chosen so that it has the largest margin, which is the distance between the hyperplane and the closest data points from each class.
- These closest points are called the support vectors.

Limites de la décision

- La limite de décision d'une SVM linéaire est un hyperplan qui sépare les différentes classes dans les données.
- Les points d'un côté de la frontière appartiennent à une classe et les points de l'autre côté appartiennent à l'autre classe.

SVM non linéaire

- Les SVM non linéaires utilisent une astuce du noyau pour transformer les données d'entrée en un espace de dimension supérieure, où une limite linéaire peut séparer les classes.
- Certains des noyaux populaires sont:
 - Noyau polynomial
 - Noyau de la fonction de base radiale (RBF)
 - Noyau sigmoïde

Astuce du noyau SVM (1)

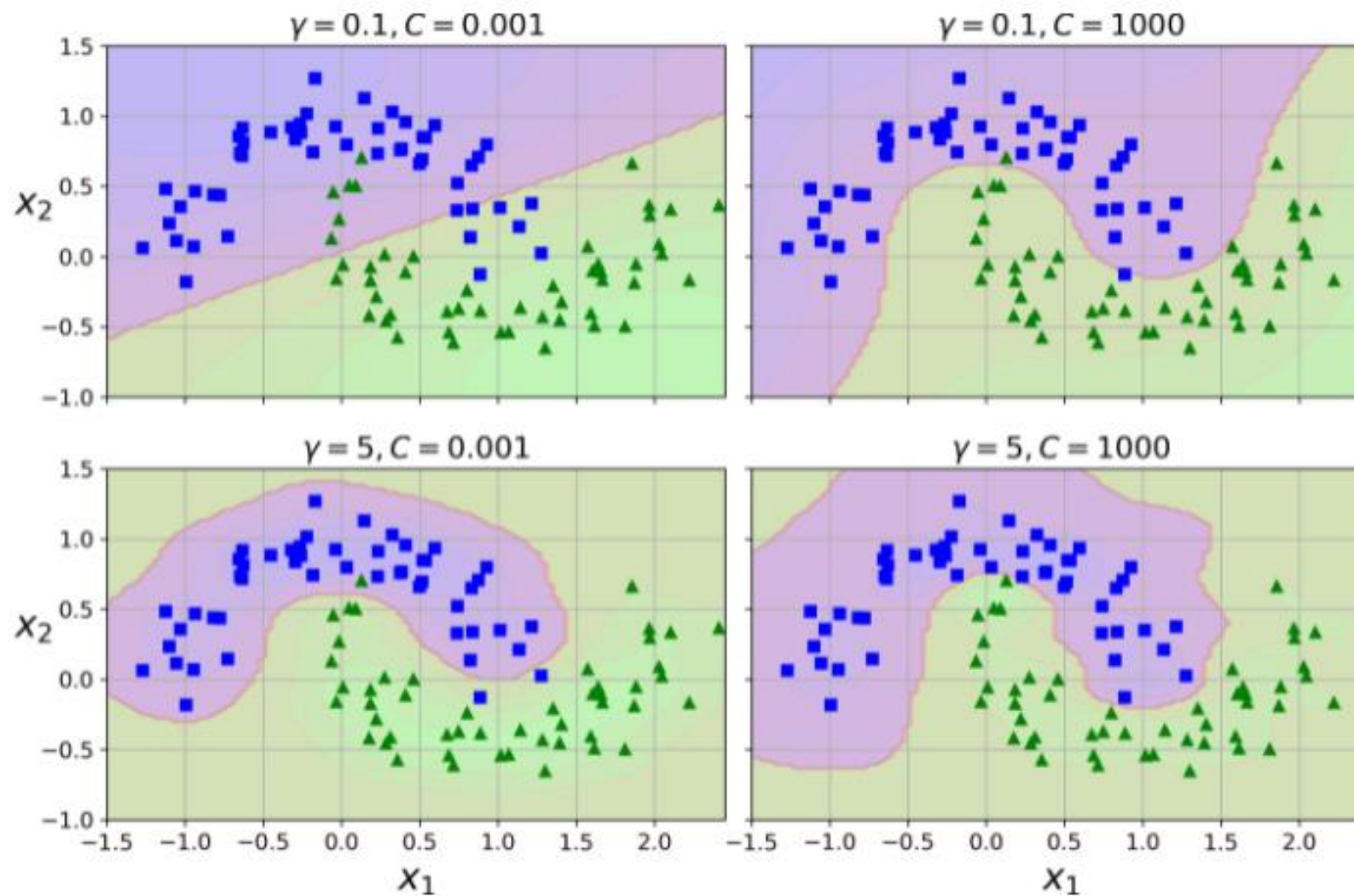
- L'astuce du noyau est une technique mathématique utilisée dans les machines à vecteurs de support (SVM) pour transformer les données d'entrée en un espace de dimension supérieure, où une frontière linéaire peut séparer les classes.
- L'idée principale derrière l'astuce du noyau est de mapper les données d'entrée dans un espace d'entités de dimension supérieure, où les données deviennent linéairement séparables.
- L'astuce du noyau permet aux SVM d'apprendre une limite de décision non linéaire sans calculer explicitement les coordonnées des données dans l'espace de dimension supérieure.
- Plusieurs fonctions de noyau peuvent être utilisées dans les SVM, telles que les noyaux linéaires, polynomiaux et de fonction de base radiale (RBF).
- Le noyau linéaire calcule simplement le produit ponctuel entre les vecteurs d'entrée et est utilisé pour les problèmes de classification linéaire et de régression.
- Le noyau polynomial élève le produit ponctuel à une puissance, qui peut être utilisée pour apprendre les limites de décision non linéaires.
- Le noyau RBF est un type de fonction de base radiale qui calcule la distance entre les vecteurs d'entrée, qui peut être utilisée pour apprendre les limites de décision non linéaires.

Astuce du noyau SVM (2)

- L'astuce du noyau est particulièrement utile lorsque les données ne sont pas séparables linéairement dans leur espace d'entités d'origine, car elles permettent aux SVM d'apprendre une limite de décision non linéaire sans avoir besoin de calculer explicitement les coordonnées des données dans un espace de dimension supérieure.
- L'astuce du noyau rend les SVM efficaces sur le plan informatique, car elle évite d'avoir à effectuer le mappage explicite des données dans un espace de dimension supérieure.
- Il est important de noter que le choix de la fonction noyau peut grandement affecter les performances du modèle SVM, et la meilleure fonction du noyau dépendra du problème spécifique et des caractéristiques des données.
- Les méthodes de réglage des hyperparamètres telles que la validation croisée peuvent être utilisées pour sélectionner la fonction optimale du noyau et ses paramètres.

Réglage des hyperparamètres

- Une chose importante à noter est que les SVM ont plusieurs hyperparamètres qui peuvent être réglés pour améliorer les performances du modèle, tels que le paramètre de régularisation et les paramètres du noyau.
- Ces hyperparamètres peuvent être sélectionnés à l'aide de techniques telles que la validation croisée.

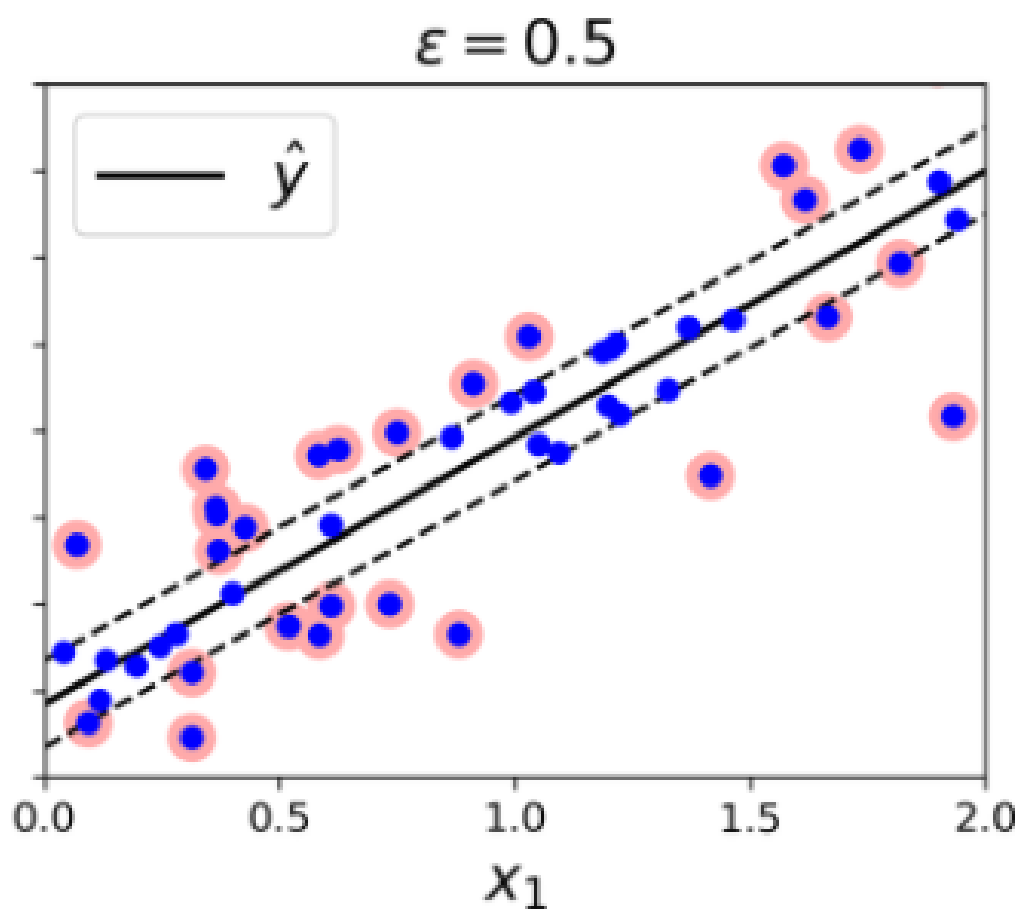
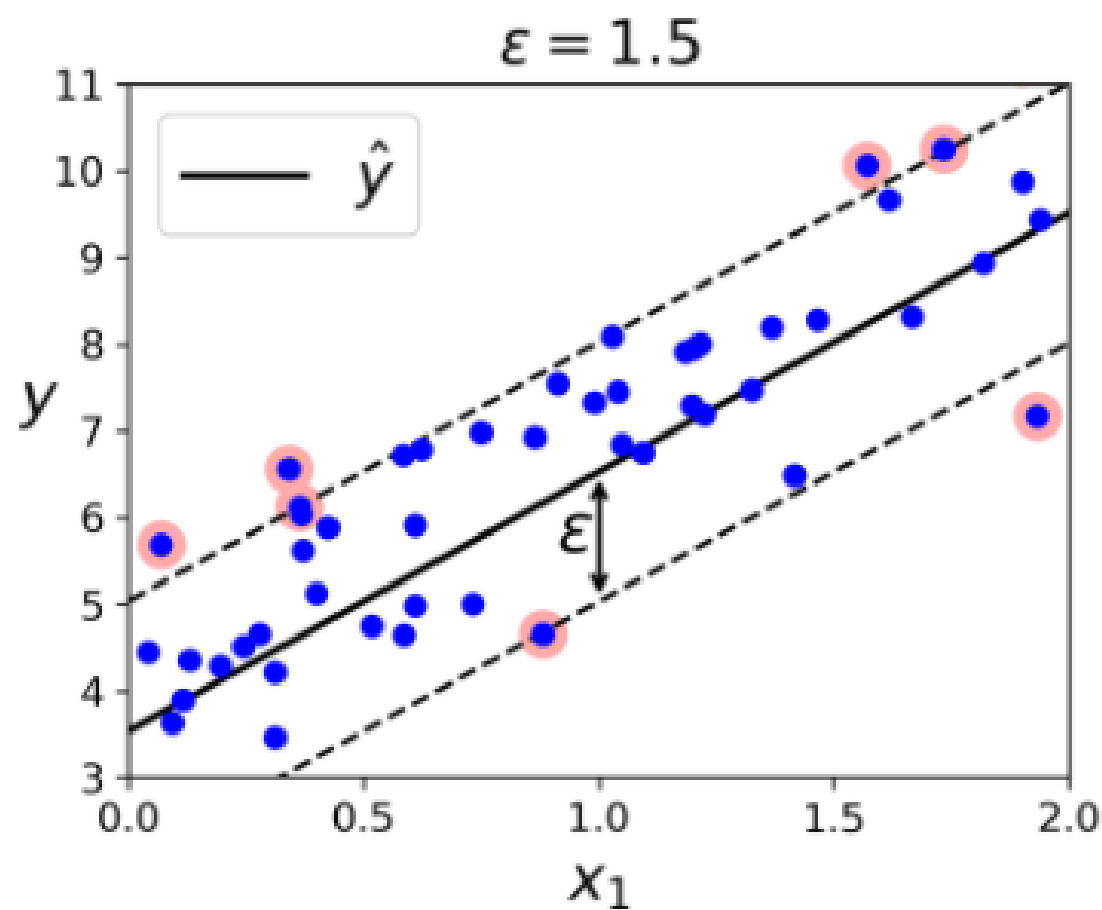


Évaluation de la SVM

- L'évaluation des modèles SVM peut être effectuée en utilisant des mesures d'évaluation courantes telles que l'exactitude, la précision, le rappel et le score F1 pour les problèmes de classification et l'erreur quadratique moyenne (MSE) pour les problèmes de régression.

Régression SVM

- La régression SVM (Support Vector Machine) est une extension de l'algorithme SVM utilisée pour les problèmes de régression, plutôt que pour les problèmes de classification.
- L'idée principale derrière la régression SVM est de trouver une fonction qui se rapproche le mieux des données en minimisant l'erreur entre les valeurs prédites et les valeurs réelles
- La fonction objective de régression SVM est similaire à celle de la classification SVM, mais avec une légère modification :
- Dans la régression SVM, l'objectif est de minimiser l'erreur entre les valeurs prédites et les valeurs réelles, tout en s'assurant que les erreurs sont inférieures à un seuil spécifié, appelé epsilon.



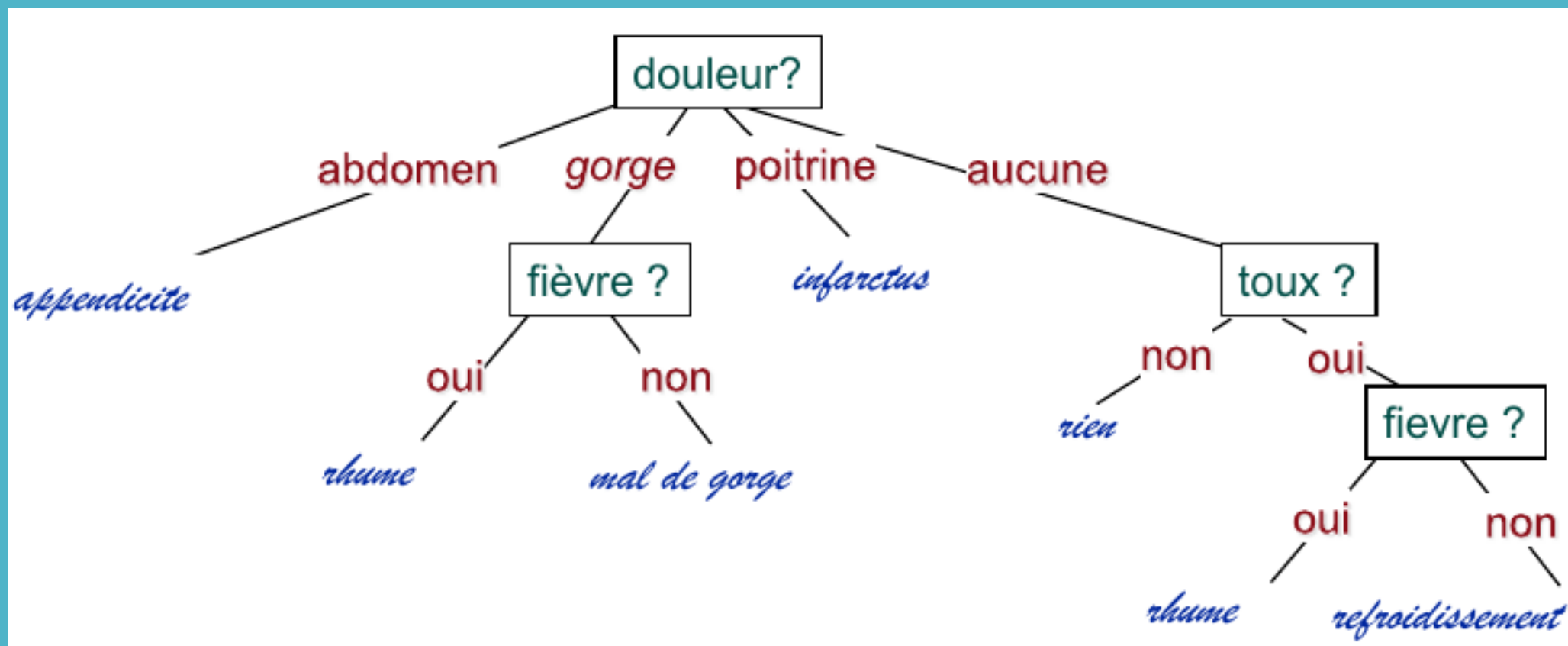
Résumé de la SVM

- Les SVM sont des algorithmes puissants pour les données linéaires et non linéaires, mais ce n'est pas toujours la meilleure méthode pour chaque problème, il est important d'évaluer les performances du modèle sur un ensemble de test retenu et de le comparer à d'autres modèles.

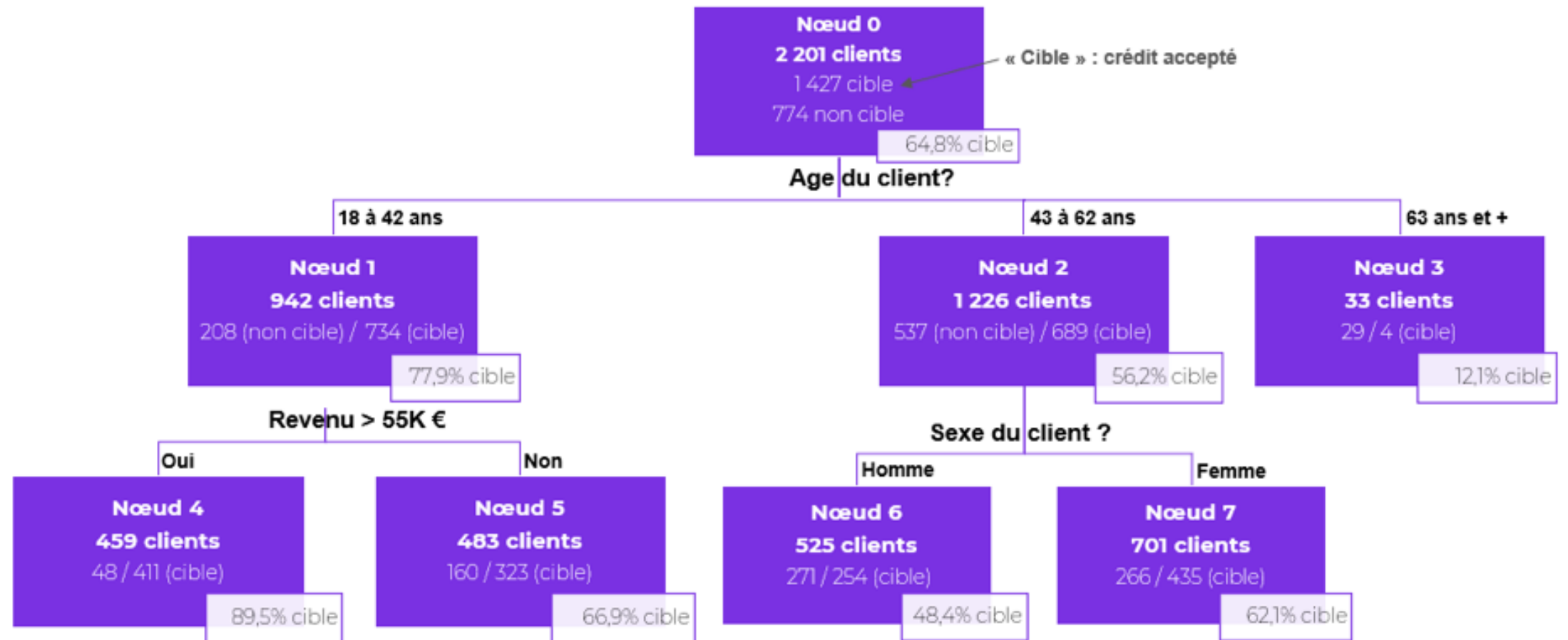
Arbres de décision

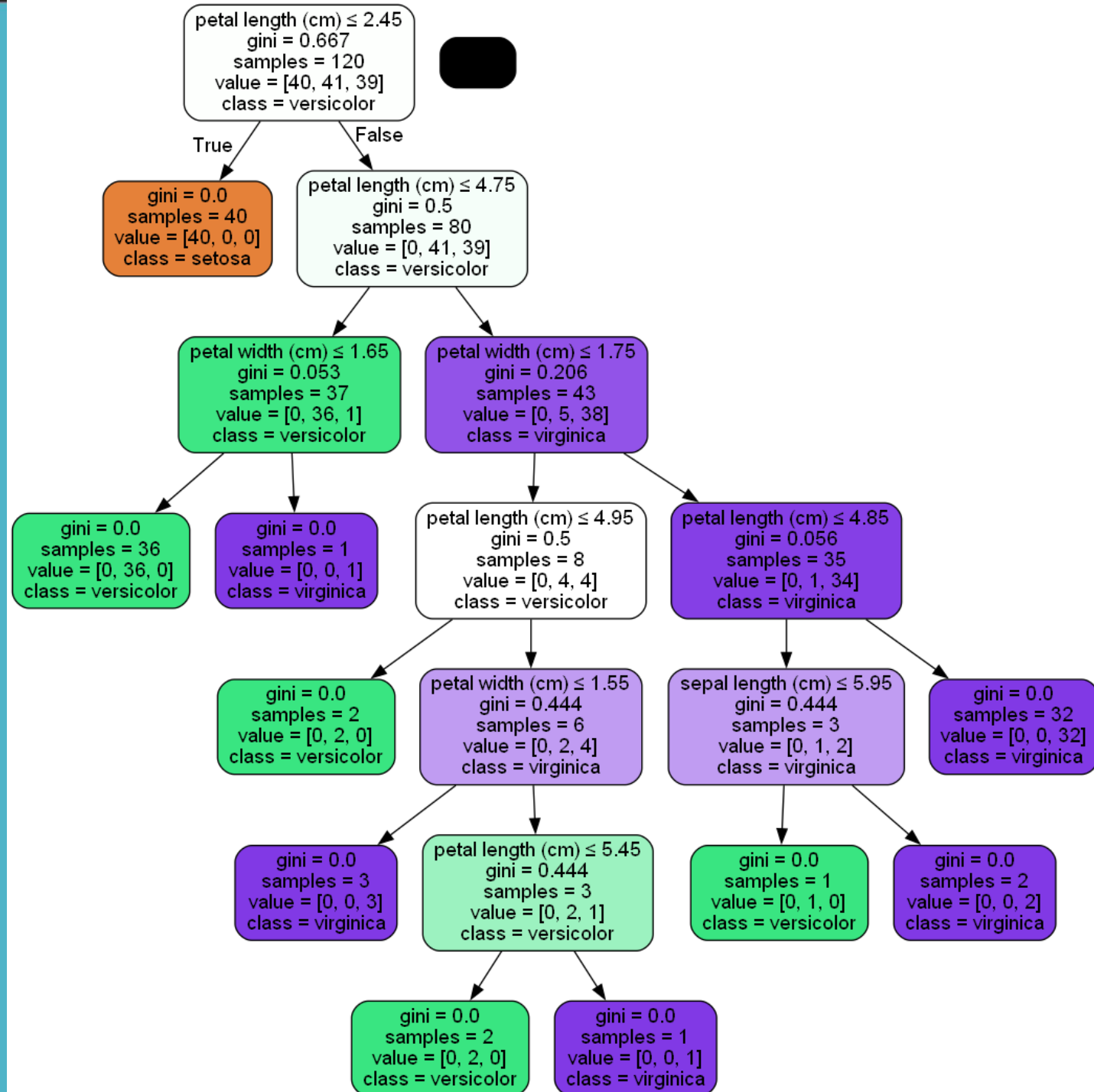
Arbre de décision - Définition

- Un arbre de décision est un modèle arborescent qui représente un ensemble de décisions et leurs conséquences possibles.
- Il s'agit d'un algorithme d'apprentissage automatique supervisé qui peut être utilisé à la fois pour les problèmes de classification et de régression.
- Dans un arbre de décision, chaque nœud interne représente un test sur un attribut, chaque branche représente le résultat du test et chaque nœud feuille représente une étiquette de classe ou une valeur numérique.
-



Acceptation des crédits bancaires





Applications des arbres de décision

- Les arbres de décision sont applicables dans un large éventail de domaines tels que la finance, la santé, le marketing et bien d'autres.
- Quelques exemples incluent la notation de crédit, la détection de la fraude, le diagnostic de maladies, la segmentation de la clientèle et la prévision du cours des actions.
-

Arbres de décision dans la classification et la régression

- Les arbres de décision sont utilisés à la fois dans l'apprentissage automatique de classification et dans les régressions

Classification de l'arbre de décision

- Pour la classification, un arbre de décision est construit en partitionnant récursivement les données en sous-ensembles aussi homogènes que possible en termes de variable cible.
- L'algorithme le plus couramment utilisé pour cela s'appelle l'algorithme ID3, qui utilise le critère de gain d'information pour déterminer le meilleur attribut pour diviser les données.

Régression de l'arbre de décision

- Les arbres de décision peuvent également être utilisés pour les problèmes de régression, où l'objectif est de prédire une variable de sortie continue. Dans les arbres de régression, la valeur de sortie est prédite par la moyenne de la variable cible pour les échantillons dans le nœud feuille.
- L'idée de base de l'algorithme est la même que pour les arbres de classification, mais au lieu de diviser les données en fonction de la mesure des impuretés, l'algorithme divise les données en fonction de la réduction de la variance. La réduction de la variance est mesurée par l'erreur quadratique moyenne (EMS), qui est la somme des différences au carré entre les valeurs prédites et les valeurs réelles.
- L'algorithme CART peut également être utilisé pour les arbres de régression, avec quelques modifications à la mesure des impuretés et au critère de fractionnement. Dans les arbres de régression, la mesure des impuretés est la variance de la variable cible, qui est la somme des différences au carré entre la variable cible et sa moyenne. L'indice de Gini et l'entropie ne sont pas utilisés dans les arbres de régression.
- Le critère de division dans les arbres de régression est la réduction de la variance, qui est la différence entre la variance du nœud parent et la somme pondérée des variances des nœuds enfants. Les poids sont les proportions des échantillons dans chaque nœud enfant.
-

Algorithme des arbres de classification et de régression (CART)

- L'algorithme le plus couramment utilisé pour cela s'appelle l'algorithme CART, qui utilise l'indice de Gini pour déterminer le meilleur attribut pour diviser les données.

L'algorithme CART fonctionne dans les étapes suivantes

1. Choisissez le meilleur attribut pour fractionner les données. Cela se fait en calculant l'indice de Gini pour chaque attribut et en choisissant l'attribut avec le plus petit indice de Gini. L'indice de Gini mesure l'impureté d'un ensemble de données, qui est définie comme la probabilité de mal classer un élément choisi au hasard dans l'ensemble de données s'il est étiqueté au hasard en fonction de la distribution des étiquettes dans le sous-ensemble.
2. Divisez les données en deux sous-ensembles en fonction de la valeur de l'attribut choisi. Le sous-ensemble qui contient les valeurs de l'attribut est considéré comme le sous-ensemble « gauche », et le sous-ensemble qui ne contient pas les valeurs de l'attribut est considéré comme le sous-ensemble « droit ».
3. Appliquez de manière récursive les étapes 1 et 2 à chaque sous-ensemble jusqu'à ce qu'un critère d'arrêt soit rempli. Le critère d'arrêt pourrait être, par exemple, une profondeur maximale de l'arbre, un nombre minimum d'échantillons dans un nœud foliaire ou une réduction minimale des impuretés.
4. Taillez l'arbre pour éviter de trop l'ajustement. Pour ce faire, supprimez les nœuds qui n'améliorent pas les performances de l'arborescence sur un jeu de validation. Les performances de l'arbre sont généralement mesurées par une mesure telle que l'exactitude, la précision, le rappel ou l'erreur quadratique moyenne.

Algorithme CART avec attributs catégoriels

- Pour les attributs catégoriels, l'indice de Gini est calculé comme suit:

$$Gini(D) = 1 - \sum_{i=1}^k p_i^2$$

où D est l'ensemble de données, k est le nombre de classes et p_i est la probabilité de classe i.

Algorithme CART avec attributs numériques

- Pour les attributs numériques, l'algorithme CART fonctionne en triant les valeurs de l'attribut dans l'ordre croissant et en calculant l'indice de Gini pour tous les points de division possibles.
- Le point de division qui donne le plus petit indice de Gini est choisi comme le meilleur split. L'indice de Gini est calculé comme suit:

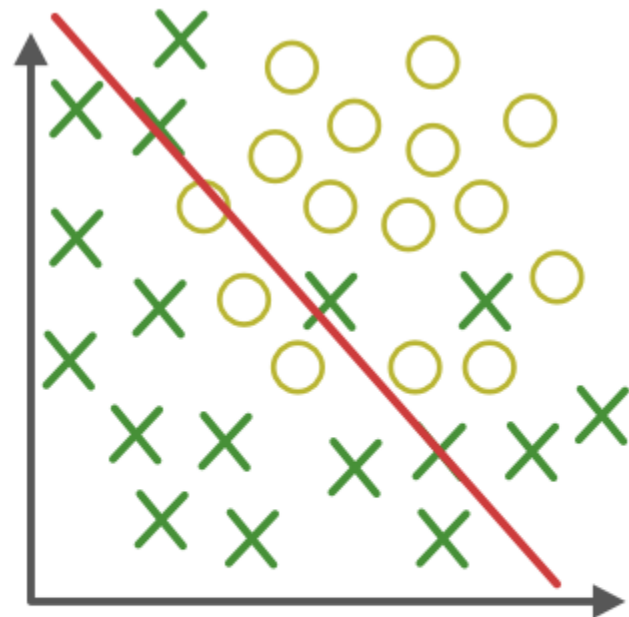
$$Gini(D, a, t) = (|D1|/|D|) * Gini(D1) + (|D2|/|D|) * Gini(D2)$$

où D est l'ensemble de données, k est le nombre de classes et p_i est la probabilité de classe i .

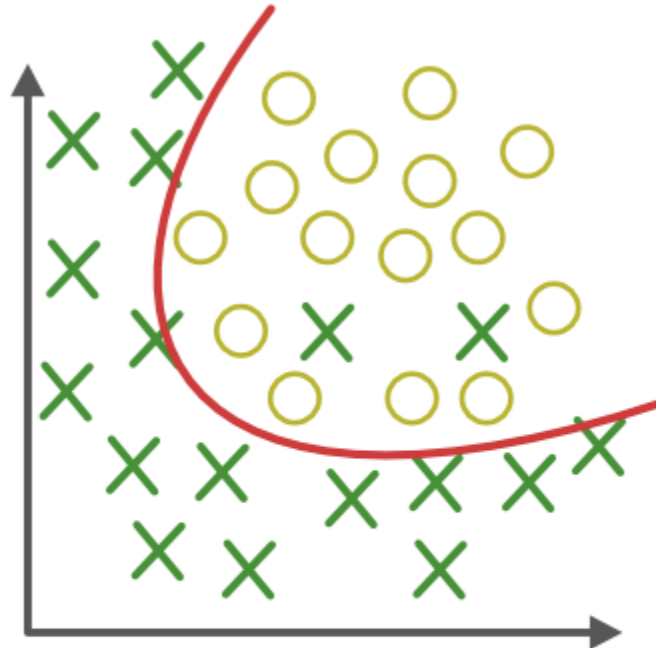
Gestion des valeurs manquantes de l'algorithme CART

- L'algorithme CART peut gérer les valeurs manquantes à l'aide de fractionnements de substitution.
- Une division de substitution est une division utilisée comme sauvegarde lorsque la division principale ne peut pas être utilisée en raison de valeurs manquantes.

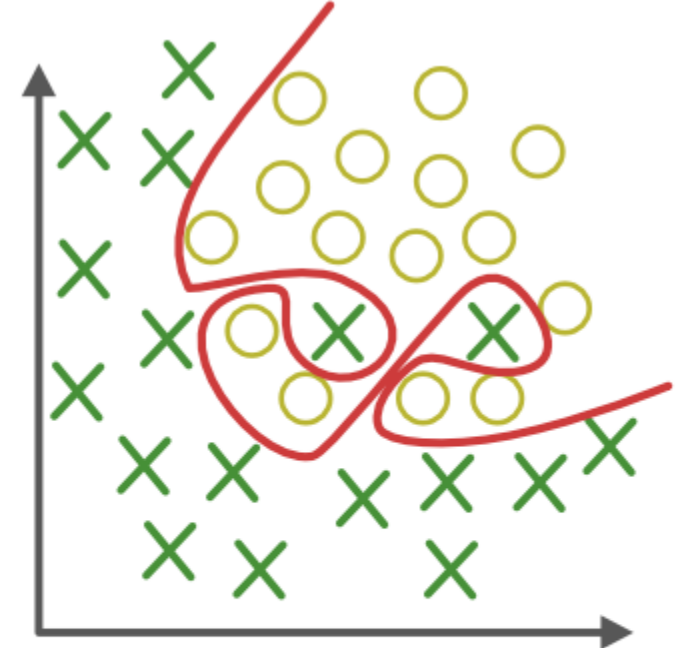
Sur-ajustement vs sous-ajustement



Under-fitting
(too simple to
explain the variance)



Appropriate-fitting



Over-fitting
(forcefitting--too
good to be true) 

Sur-ajustement vs sous-ajustement

	Underfitting	Just right	Overfitting
Symptoms	<ul style="list-style-type: none"> • High training error • Training error close to test error • High bias 	<ul style="list-style-type: none"> • Training error slightly lower than test error 	<ul style="list-style-type: none"> • Very low training error • Training error much lower than test error • High variance
Regression illustration			
By Afshine Amidi and Shervine Amidi, Via: eml.india			
Classification illustration			
Deep learning illustration			

Sous-ajustement

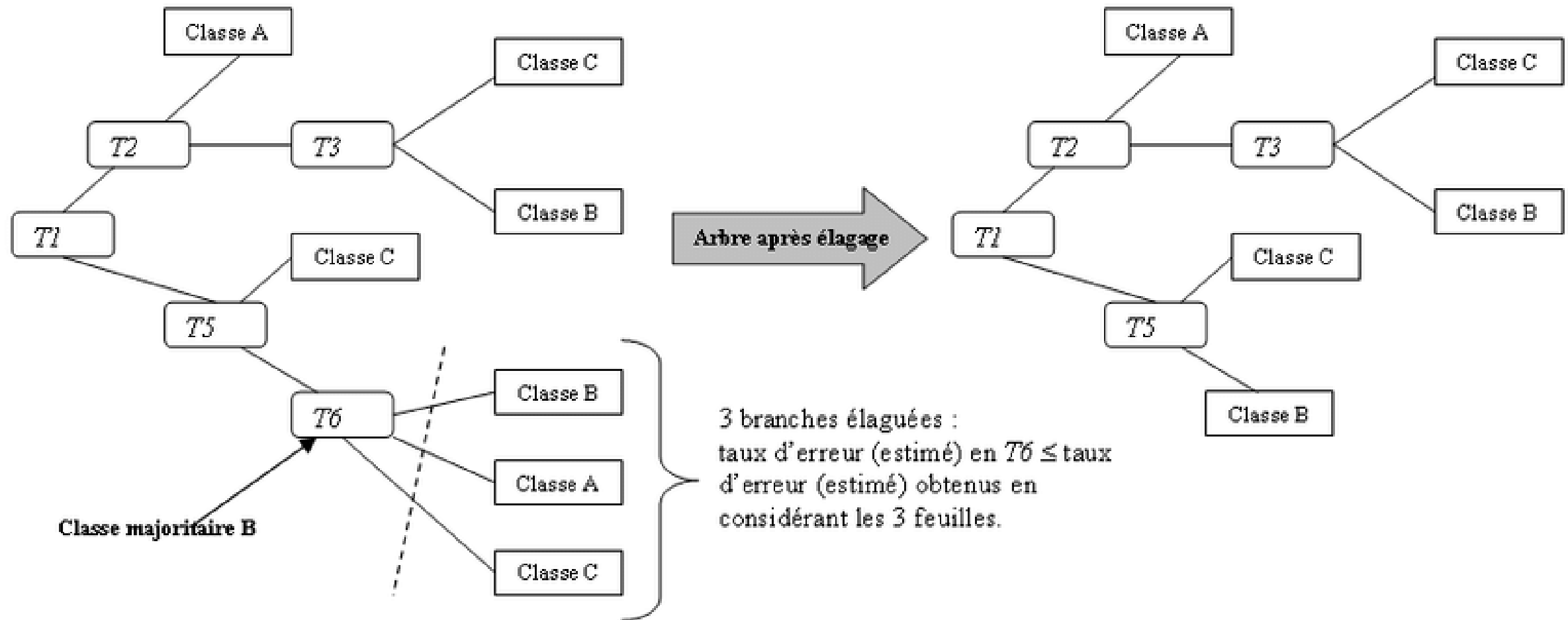
- Le sous-ajustement est un problème courant dans les algorithmes d'arbre de décision, y compris l'algorithme CART. Le sous-ajustement se produit lorsque l'arborescence est trop simple et ne parvient pas à capturer les modèles sous-jacents dans les données. Cela peut entraîner un manque de précision à la fois sur les données d'entraînement et sur les nouvelles données invisibles.
- Le sous-ajustement est souvent causé par des méthodes de pré-taille qui empêchent l'arbre de pousser trop tôt ou par l'utilisation d'un critère d'arrêt trop strict. Par exemple, la définition d'une faible profondeur maximale pour l'arbre ou d'un nombre minimum élevé d'échantillons dans un nœud feuille peut entraîner un arbre trop simple.
- Pour éviter le sous-ajustement, il est important de permettre à l'arbre de grandir à sa taille optimale et d'utiliser un critère d'arrêt plus détendu. Cela peut être fait en utilisant des méthodes post-élagage, telles que l'élagage réduit les erreurs ou l'élagage de complexité des coûts, qui suppriment les nœuds de l'arbre une fois qu'il a atteint sa taille maximale. Ces méthodes peuvent supprimer les nœuds qui n'améliorent pas la précision de l'arborescence, tout en conservant les nœuds qui capturent les modèles sous-jacents dans les données.
- Une autre façon d'éviter le sous-ajustement consiste à utiliser des algorithmes d'arbre de décision plus complexes, tels que les forêts aléatoires ou l'amplification de gradient, qui peuvent capturer des modèles plus complexes dans les données. Ces algorithmes utilisent plusieurs arbres de décision, chacun étant formé sur un sous-ensemble de données, et combinent leurs prédictions pour améliorer la précision du modèle.
-

Sur-ajustement

- Le surajustement est un problème courant dans les algorithmes d'arbre de décision, y compris l'algorithme CART.
- Le surajustement se produit lorsque l'arborescence est trop complexe et capture le bruit dans les données, plutôt que les modèles sous-jacents.
- Cela peut entraîner de mauvaises performances de généralisation et une faible précision sur les nouvelles données invisibles.

Élagage

- L'élagage est une technique courante utilisée pour éviter le surajustement dans les arbres de décision.
- L'élagage consiste à supprimer de l'arborescence les nœuds qui n'améliorent pas ses performances de généralisation.
- Il existe plusieurs types de méthodes d'élagage, y compris le pré-élagage et le post-taille.



Pré-élagage

- La pré-élagage est une méthode d'élagage qui empêche l'arbre de pousser avant qu'il ne devienne trop complexe.
- Cela se fait en établissant un critère d'arrêt basé sur une mesure d'impureté, telle que l'indice de Gini ou l'entropie.
- Le critère d'arrêt pourrait être, par exemple, une profondeur maximale de l'arbre, un nombre minimum d'échantillons dans un nœud foliaire ou une réduction minimale des impuretés.
- La pré-élagage est rapide et simple, mais peut entraîner un sous-ajustement si l'arbre n'est pas autorisé à atteindre sa taille optimale.

Post-élagage

- Le post-élagage est une méthode d'élagage qui enlève les nœuds de l'arbre après qu'il ait atteint sa taille maximale.
- Cela se fait en cultivant d'abord l'arbre sur un ensemble d'entraînement, puis en l'élaguant sur un ensemble de validation.
- Le jeu de validation est un sous-ensemble de l'ensemble d'apprentissage qui n'est pas utilisé pour l'apprentissage, mais qui est utilisé pour évaluer les performances de l'arborescence.
- Les performances de l'arbre sont généralement mesurées par une mesure telle que l'exactitude, la précision, le rappel ou l'erreur quadratique moyenne.
- Il existe plusieurs types de méthodes post-élagage, notamment l'élagage à erreur réduite, l'élagage de complexité des coûts et l'élagage de la longueur minimale de description.

Types de méthodes post-élagage (1)

- Réduction de l'élagage des erreurs:
- L'élagage à erreurs réduites est une méthode d'élagage qui supprime de l'arborescence les nœuds qui n'améliorent pas ses performances de généralisation.
- Pour ce faire, supprimez récursivement chaque nœud de l'arborescence et évaluez les performances du sous-arbre résultant sur le jeu de validation.
- Si les performances de la sous-arborescence sont meilleures que celles de l'arborescence d'origine, le nœud est supprimé. Ce processus est répété jusqu'à ce qu'aucun autre nœud ne puisse être supprimé.
-

Types de méthodes post-élagage (2)

- **Complexité des coûts d'élagage**

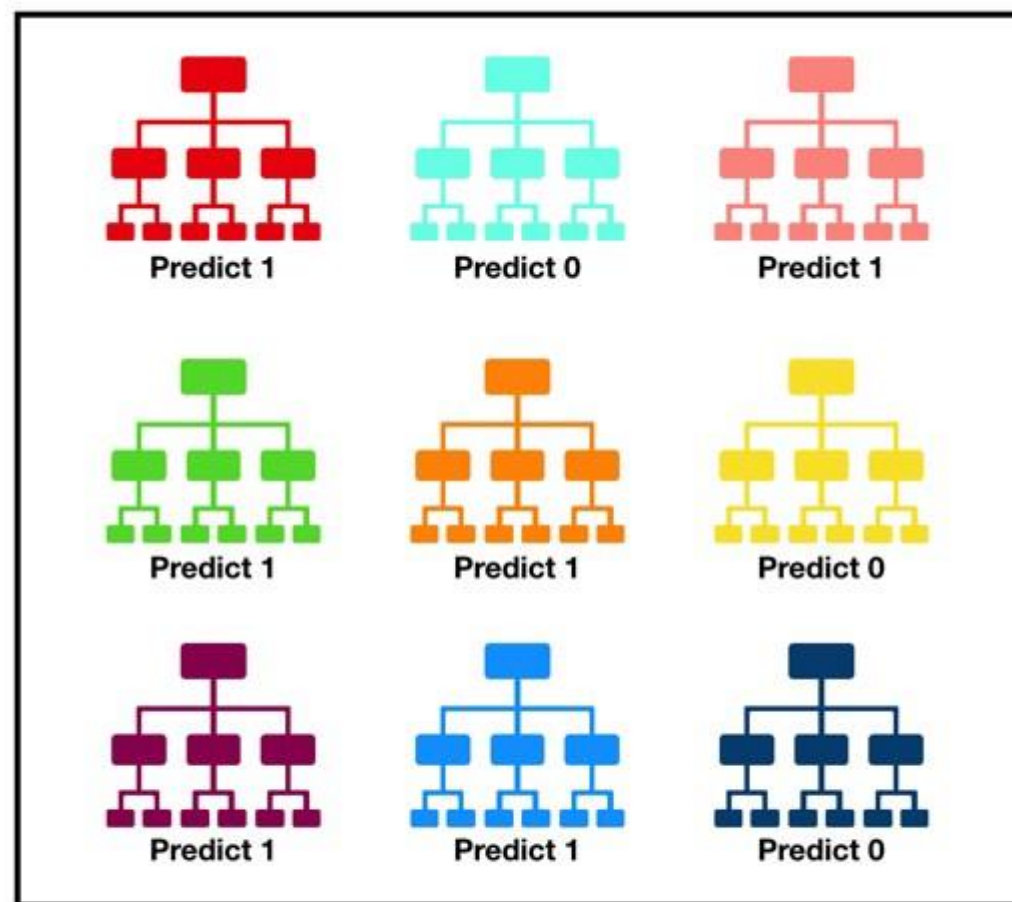
- L'élagage de la complexité des coûts, également connu sous le nom d'élagage de la complexité des coûts minimales, est une méthode d'élagage qui équilibre la complexité et la précision de l'arbre.
- Cela se fait en introduisant un terme de pénalité qui pénalise la complexité de l'arbre, en plus du terme d'erreur qui mesure la précision de l'arbre.
- Le terme de pénalité est contrôlé par un hyperparamètre appelé paramètre de complexité, qui détermine le compromis entre précision et complexité.

Types de méthodes post-élagage (3)

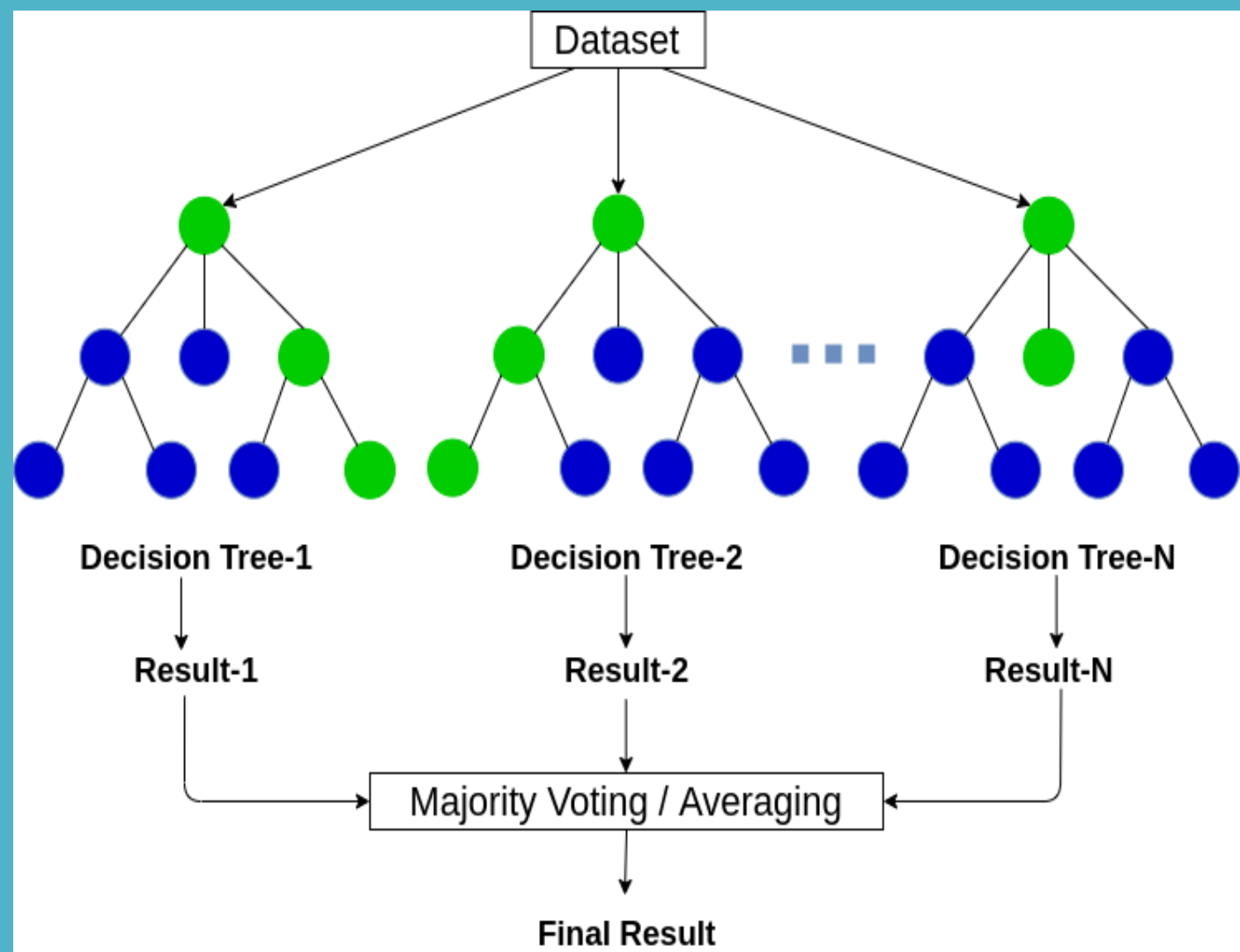
- **Longueur minimale de description élagage**
 - L'élagage de la longueur minimale de description est une méthode d'élagage qui minimise la longueur de description de l'arbre et des données, en partant du principe que l'explication la plus simple est la meilleure.
 - Cela se fait en codant l'arbre et les données à l'aide d'un schéma de codage, tel que le codage de Huffman ou le codage arithmétique, puis en calculant la longueur totale du code.
 - L'arbre est élagué pour minimiser la longueur totale du code, sous réserve d'une contrainte sur la précision de l'arbre.

Méthode de la forêt aléatoire

- Une forêt aléatoire est une collection d'arbres de décision qui sont formés sur des sous-ensembles aléatoires des données d'apprentissage et des sous-ensembles aléatoires des entités. L'idée derrière les forêts aléatoires est d'introduire le caractère aléatoire dans l'algorithme de l'arbre de décision, ce qui peut réduire le surajustement et améliorer les performances de généralisation du modèle.
- L'algorithme de forêt aléatoire fonctionne comme suit :
- Sélectionnez au hasard un sous-ensemble des données d'entraînement, avec remplacement. C'est ce qu'on appelle l'agrégation bootstrap ou l'ensachage.
- Sélectionnez au hasard un sous-ensemble des entités, généralement la racine carrée du nombre total d'entités. C'est ce qu'on appelle l'ensachage de caractéristiques ou la méthode du sous-espace aléatoire.
- Entraînez un arbre de décision sur les données et fonctionnalités sélectionnées.
- Répétez les étapes 1 à 3 pour un nombre fixe de fois, généralement 100 ou plus.
- Faire des prédictions en combinant les prédictions de tous les arbres, soit par vote majoritaire (pour la classification), soit par moyenne (pour la régression).



Tally: Six 1s and Three 0s
Prediction: 1



Avantages de la méthode de la forêt aléatoire

- L'algorithme de forêt aléatoire présente plusieurs avantages par rapport à un seul arbre de décision:
- Il peut gérer un grand nombre de fonctionnalités et de données bruyantes, ce qui peut être problématique pour un seul arbre de décision.
- Il peut capturer des modèles plus complexes dans les données, en combinant les prédictions de plusieurs arbres de décision.
- Il peut réduire le surajustement et améliorer les performances de généralisation du modèle, en introduisant le caractère aléatoire dans l'algorithme de l'arbre de décision.

Applications de la méthode de la forêt aléatoire

- Les forêts aléatoires sont devenues l'un des algorithmes d'apprentissage automatique les plus populaires et sont largement utilisées dans de nombreuses applications, telles que la classification d'images, la bioinformatique et la finance.
- Le package Python sklearn fournit une implémentation simple et facile à utiliser de l'algorithme de forêt aléatoire, qui peut être personnalisé à l'aide d'une gamme d'hyperparamètres, tels que le nombre d'arbres, la profondeur maximale des arbres et le nombre minimum d'échantillons requis pour diviser un nœud.