Chapitre 3 : Classification supervisée

Introduction à la classification

La classification est une technique de data Mining qui consiste à attribuer des étiquettes ou des classes à des données en fonction de leurs caractéristiques ou attributs. Cela implique la construction d'un modèle à partir d'un ensemble de données d'apprentissage, puis l'utilisation de ce modèle pour prédire les étiquettes de nouvelles données non étiquetées. La classification trouve des applications dans une variété de domaines, de la médecine à la finance, en passant par le marketing et la recherche.

1. Définition de la Classification en Data Mining

La classification est un processus d'apprentissage supervisé où l'objectif est de prédire la classe ou la catégorie d'un objet en fonction de ses attributs. Dans le contexte du Data Mining, l'objectif est d'automatiser ce processus en utilisant des algorithmes et des modèles.

Un exemple simple de classification pourrait être la classification des e-mails en « spam » ou « non-spam ». Dans ce cas, les attributs pourraient inclure des mots clés, la longueur du texte, la présence d'URL, etc. Le modèle de classification apprendrait à partir d'un ensemble d'e-mails étiquetés pour ensuite classer de nouveaux e-mails comme spam ou non-spam en fonction de ces attributs.

2. Types de Classement

Classification Binaire : Il s'agit de la classification en deux classes (par exemple, oui/non, spam/non spam).

Classification Multi classe : Ici, il y a plus de deux classes possibles. Par exemple, dans la classification des fleurs Iris, les classes sont les différentes espèces d'iris (setosa, versicolor, virginica), et les caractéristiques sont les mesures des sépales et des pétales.

3. Données d'Apprentissage et de Test

Avant de commencer à classer des données, il est essentiel de diviser votre ensemble de données en deux parties : l'ensemble d'apprentissage, appelé aussi training set et l'ensemble de test appelé aussi test set. L'ensemble d'apprentissage est utilisé pour entraîner le modèle de classification (généralement 2/3 des données), tandis que l'ensemble de test est utilisé pour évaluer ses performances (généralement 1/3 des données).

4. Évaluation de la Classification

Une fois que vous avez utilisé un algorithme de classification pour classer vos données, vous devez évaluer les performances du modèle. Voici quelques métriques utilisées pour l'évaluation :

Précision : La précision mesure le nombre de prédictions corrigées par rapport au nombre total de prédictions. Une précision élevée signifie que le modèle fait peu d'erreurs.

Rappel : Le rappel mesure la capacité du modèle à identifier toutes les instances positives (vraies positives) parmi toutes les instances réellement positives. Un rappel élevé signifie que le modèle trouve la plupart des exemples positifs.

F-mesure : La F-mesure est une mesure qui combine la précision et le rappel en une seule métrique. Elle est utile lorsque vous souhaitez trouver un équilibre entre la précision et le rappel.

5. Applications de la Classification dans Divers Domaines

La classification est largement utilisée dans de nombreux domaines pour résoudre une variété de problèmes. Voici quelques exemples d'applications de classification dans différents domaines :

Médecine : Diagnostic médical basé sur des données telles que les symptômes du patient, les résultats de tests et les résultats médicaux. Par exemple, la classification peut aider à prédire si un patient a une maladie particulière (par exemple, le cancer) en fonction de ses caractéristiques.

Finance : Évaluation du risque de crédit pour les prêts bancaires en fonction des financiers confirmés et des données de crédit des emprunteurs.

Marketing : Ciblage des clients pour des campagnes publicitaires en fonction de leurs comportements d'achat antérieurs et de leurs préférences.

Détection de Fraude : Identification des transactions frauduleuses dans les transactions par carte de crédit en fonction de modèles de dépenses inhabituelles.

Vision par Ordinateur : Classification d'images, par exemple pour la reconnaissance d'objets dans des photos ou la détection de visages.

Analyse de Sentiments : Classification de textes (avis, commentaires) en fonction de leurs sentiments (positif, négatif, neutre).

6. Méthodes de classification

6.1. K-Voisins les plus proches (KNN)

Les K-Voisins les plus Proches (KNN) est une méthode de classification qui repose sur le principe que des points de données similaires tendent à appartenir à la même classe. KNN attribue une classe à un point de données non étiqueté en se basant sur la classe majoritaire parmi ses K voisins les plus proches dans l'espace des attributs. La distance entre les points de données est utilisée pour déterminer leur proximité.

Principe de Base : Classification Basée sur la Proximité

L'idée fondamentale derrière KNN est que les points de données qui sont proches les uns des autres dans l'espace des attributs sont susceptibles de partager des caractéristiques similaires et donc d'appartenir à la même classe. La proximité est généralement réglée en utilisant des mesures de distance telles que la distance euclidienne, la distance de Manhattan, etc.

6.1.1. Paramètre K et Son Influence sur la Performance

Le paramètre K spécifique le nombre de voisins les plus proches que le modèle KNN prendra en compte pour prendre une décision de classification. Un K plus élevé implique une prise de décision plus lissée, tandis qu'un K plus faible peut rendre le modèle plus sensible aux fluctuations locales. Le choix de la valeur optimale de K est crucial et peut varier en fonction de la nature des données et du problème.

6.1.2. Fonctionnement du KNN

- Choix de k : Lors de l'utilisation de k-NN, il est essentiel de déterminer un nombre k, représentant le nombre de voisins les plus proches pris en compte pour la classification ou la régression. Par exemple, si k = 3, l'algorithme considérera les trois voisins les plus proches pour prendre une décision.
- Calcul de la Distance : L'algorithme k-NN mesure la distance entre l'instance de test à classer et toutes les instances de l'ensemble d'apprentissage. Cette distance peut être utilisée en utilisant diverses métriques telles que la distance euclidienne ou la distance de Manhattan.
- Sélection des k Voisins les Plus Proches : L'algorithme identifie les k voisins les plus proches de l'instance de test en fonction des distances calculées. Ces voisins sont choisis en raison de la similarité de leurs caractéristiques avec celles de l'instance de test.
- Classification : Dans le cas de la classification, pour attribuer une classe à l'instance de test, l'algorithme prend en compte la classe majoritaire parmi les k voisins les plus proches. Par exemple, si parmi les k voisins, 2 appartiennent à la classe A et 1 à la classe B, l'instance de test sera classée dans la classe A.

- Régression : Lorsqu'il s'agit de régression, au lieu de prendre une décision basée sur des classes, l'algorithme calcule la moyenne des valeurs cibles des k voisins les plus proches. Cette moyenne est ensuite attribuée à l'instance de test.
- Résultats : L'instance de test est ainsi classée ou estimée pour la régression. Ce processus est répété pour toutes les instances de test, et les résultats sont collectés pour évaluer la performance du modèle.

6.1.3. Avantages et inconvénients de KNN

Avantages :

- Facile à comprendre et à mettre en œuvre.
- Ne nécessite pas d'hypothèses sur la distribution des données.
- Peut être utilisé pour des problèmes de classification binaire et multiclasse.

Inconvénients :

- Peut-être être sensible à la valeur de k.
- Peut être coûteux en termes de calcul, en particulier avec de grandes bases de données.
- Ne fonctionne pas bien avec des données de grande dimensionnalité.

6.1.4. Exemples d'Utilisation de KNN

Classification d'Iris : L'un des exemples classiques est la classification des espèces d'iris (Setosa, Versicolor, Virginica) en fonction de leurs longueurs de pétale et de sépale. En utilisant KNN, on peut prédire l'espèce d'iris pour une nouvelle fleur en se basant sur les caractéristiques de ses pétales et sépales.

Recommandation de Films : Dans les systèmes de recommandation de films, KNN peut être utilisé pour recommander des films similaires à ceux que l'utilisateur a aimé. Les films similaires sont déterminés en fonction des évaluations et des préférences d'autres utilisateurs.

Exercice sur l’algorithme KNN

Soit l’ensemble de données sur des véhicules contenant des informations sur la vitesse maximale (en km/h) et la consommation de carburant (en litres par 100 km) de différents modèles de voitures. Vous avez également des étiquettes de classe indiquant si chaque modèle est économe en carburant (E) ou gourmand en carburant (G).

Vitesse Maximale (km/h)	220	180	210	200	230	190
Consommation (l/100km)	8,5	11,2	9,0	10,5	8,0	12,0
Classe	E	G	E	G	E	G

Pour K = 3 utilisez l'algorithme KNN pour classer un nouveau modèle de voiture avec une vitesse maximale de 195 km/h et une consommation de carburant de 9,5 litres par 100 km.

Solution

Pour le nouveau modèle de voiture (195 Km/h, 9,5 L/100 Km), calculons la distance euclidienne avec chaque modèle de voiture de l'ensemble de données.

Distance entre (220, 8,5) et (195, 9,5) = sqrt ((220-195)^2 + (8,5-9,5)^2) ≈ 25,02

Distance entre (180, 11,2) et (195, 9,5) = sqrt ((180-195)^2 + (11,2-9,5)^2) ≈ 15,10

Distance entre (210, 9,0) et (195, 9,5) = sqrt ((210-195)^2 + (9,0-9,5)^2) ≈ **15,01**

Distance entre (200, 10,5) et (195, 9,5) = sqrt ((200-195)^2 + (10,5-9,5)^2) ≈ **5,10**

Distance entre (230, 8,0) et (195, 9,5) = sqrt ((230-195)^2 + (8,0-9,5)^2) ≈ 35,03

Distance entre (190, 12,0) et (195, 9,5) = sqrt ((190-195)^2 + (12,0-9,5)^2) ≈ **5,59**

Identification des 3 voisins les plus proches :

Les trois distances les plus courtes sont : 15,01 et 5,10 et 5,59

Les trois voisins les plus proches sont donc : (210, 9), (200, 10.5) et (190, 12).

Parmi les trois voisins les plus proches, deux appartiennent à la classe G (gourmand en carburant) et un à la classe E (économique en carburant). Par conséquent, la classe la plus probable pour le nouveau modèle de voiture est G (gourmand en carburant).

6.2. Arbre de Décision

Les arbres de décision sont des outils de classification puissants qui fonctionnent en divisant les données en sous-groupes homogènes en fonction des valeurs des attributs. Chaque nœud de l'arbre représente une décision basée sur une caractéristique particulière, et chaque feuille attribue une classe ou une étiquette à un groupe de données.

6.2.1. Structure d'un Arbre de Décision : Nœuds, Branches, Feuilles

Un arbre de décision est organisé en plusieurs composants clés :

Nœuds : Les nœuds sont les points de décision dans l'arbre. Le nœud racine est le point de départ, et chaque nœud représente une question sur un attribut. Les nœuds intérieurs guident le processus de décision, tandis que les feuilles attribuent une classe aux données.

Branches : Les branches dépendent les nœuds entre eux. Chaque branche représente une réponse possible à la question posée par le nœud précédent.

Feuilles : Les feuilles sont les nœuds terminaux de l'arbre. Chaque feuille attribue une classe spécifique aux données qui y aboutissent.

6.2.2. Algorithmes utilisés pour la construction d'un Arbre de décision

Dans les arbres de décision, plusieurs algorithmes sont utilisés pour la construction et la classification (CART, ID3, C4.5, C5.0, CHAID...). Nous allons parler plus en détail des deux algorithmes de construction d'arbres de décision les plus fameux : CART (Classification and Régression Trees) et ID3 (Itérative Dichotomiser 3).

A. ID3 (Itérative Dichotomiser 3) :

L'algorithme ID3 est l'un des premiers algorithmes populaires pour construire des arbres de décision. Il a été développé par Ross Quinlan. Voici comment fonctionne l'algorithme ID3 :

Sélection de la Caractéristique de Division : L'algorithme ID3 commence par la racine de l'arbre et sélectionne la caractéristique de division qui sépare le mieux les données d'entraînement en classes distinctes. Il utilise généralement l'entropie ou le gain d'information pour mesurer la pureté de la division.

Création des Sous-Arbres : Une fois la caractéristique de division sélectionnée, l'ensemble de données est divisé en sous-ensembles en fonction des valeurs possibles de cette caractéristique. Un nœud est créé pour chaque valeur, et le processus est récursivement appliqué à chaque sous-ensemble jusqu'à ce que les feuilles soient atteintes.

Arrêt : L'algorithme s'arrête lorsque l'une des conditions suivantes est remplie :

Toutes les instances d'un nœud appartiennent à la même classe.

Il n'y a plus de caractéristiques pour diviser les données.

D'autres critères d'arrêt spécifiques sont atteints.

B. Algorithme CART (Arbres de Classification et de Régression)

L'algorithme CART est un autre algorithme populaire pour construire des arbres de décision. Il a été développé par Breiman et al. (1984). Voici comment fonctionne l'algorithme CART :

Sélection de la Caractéristique de Division : Contrairement à ID3, qui utilise l'entropie, CART utilise le critère de Gini pour mesurer la pureté de la division. Il sélectionne la caractéristique qui minimise le critère de Gini.

Création des Sous-Arbres : Comme ID3, CART divise l'ensemble de données en sous-ensembles en fonction des valeurs possibles de la caractéristique de division. Il crée des sous-arbres récursivement.

Arrêt : Les critères d'arrêt pour CART sont similaires à ceux d'ID3, à savoir toutes lorsque les instances d'un nœud appartiennent à la même classe ou lorsque d'autres critères spécifiques sont remplis.

6.2.3. Fonctions Mathématiques Utilisées

Les deux algorithmes, ID3 et CART, utilisent des fonctions mathématiques pour mesurer la pureté des divisions et sélectionner la meilleure caractéristique de division.

Entropie : L'entropie mesure le désordre ou l'incertitude dans un ensemble de données. Plus l'entropie est élevée, plus l'ensemble de données est mélangé. L'entropie est utilisée par ID3 pour mesurer la pureté.

Formule mathématique de l'entropie :

$$Entropie : H(X) = -\sum_{i=1}^{n} Pi\ log_b(Pi)$$

Où :

(Pi) est la probabilité de sélectionner au hasard un exemple dans la classe i.

(b) est la base du logarithme utilisée dans le calcul de l'entropie et du gain, elle dépend du nombre de classes (ou de catégories) dans le problème de classification. La base la plus couramment utilisée est le logarithme en base 2, ce qui donne des unités d'entropie en "bits".

Gain d'Information : Le gain d'information mesure la réduction de l'entropie après une division. Il est utilisé par ID3 pour sélectionner la caractéristique de division qui **maximise** la réduction de l'entropie.

$$Gain : Gain\ (X, ai) = H(X) - \sum \frac{|Xai = v|}{|X|} H(Xai = v)$$

Critère de Gini : Le critère de Gini mesure l'impureté d'un ensemble de données en termes de probabilité qu'une instance soit mal classée en aléatoirement une classe. CART utilise le critère de Gini pour mesurer la pureté et sélectionner la caractéristique de division.

$$Gini - index\ (Attribut = Valeur) = \ 1 - \sum (Pi)^2$$

$$Gini - index\ (Attribut) = \sum Pv * GI(v)$$

6.2.4. Élagage pour Éviter le Surapprentissage

Les algorithmes ID3 et CART peuvent être vulnérables au surapprentissage, un phénomène où un modèle s'adapte excessivement aux données d'entraînement au point de perdre sa capacité à généraliser sur de nouvelles données. Pour contrer ce risque, ils mettent en œuvre des méthodes d'élevage, qui consistent à retirer certaines parties de l'arbre pour prévenir le surapprentissage. Cette opération vise à supprimer les branches qui ne contribuent pas de manière significative à l'information utile. Voici comment chaque algorithme gère l'élagage :

Élagage de l'Arbre ID3 : Pour atténuer le surapprentissage, ID3 peut être soumis à un élagage en imposant une limite à la profondeur de l'arbre ou en fusionnant des branches qui apportent peu d'informations distinctes.

Élagage de l'Arbre CART : Pour CART, un processus connu sous le nom d'« élagage récursif » est utilisé. Ce processus opère de bas en haut dans l'arbre, en remplaçant des sous-arbres entiers par des feuilles si cela conduit à une réduction de l'erreur de classification.

6.2.5. Avantages et Inconvénients des Arbres de Décision

Avantages :

- Facilité d'interprétation et de visualisation.
- Gère naturellement les données catégorielles et numériques.
- Peut capturer des relations complexes entre les attributs.

Inconvénients :

- Tendance au surapprentissage, surtout avec des arbres profonds.
- Sensibles aux variations mineures dans les données d'entraînement.
- L'optimisation de la structure peut être coûteuse.

6.2.6. Exemples Concrets d'Arbres de Décision

Diagnostic Médical : Un arbre de décision peut être utilisé pour diagnostiquer des maladies en se basant sur les symptômes des patients.

Décision de Crédit : Dans le secteur financier, un arbre de décision peut aider à décider si un demandeur de crédit est à faible ou à haut risque en fonction de critères tels que le revenu, l'historique de crédit, etc.

Exercice sur les arbres de décision

Contexte : Vous décidez si vous devez jouer au tennis ou non. Vous disposez de deux paramètres pour prendre votre décision : la météo et la température. Voici les données que vous avez recueillies :

Jour	Ciel	Température	Jouer ?
1	Ensoleillé	Chaude	Oui
2	Ensoleillé	Chaude	Oui
3	Nuageux	Chaude	Oui
4	Pluvieux	Fraiche	Oui
5	Pluvieux	Froide	Non
6	Pluvieux	Froide	Non
7	Nuageux	Froide	Oui
8	Ensoleillé	Fraiche	Oui
9	Ensoleillé	Froide	Non

Objectif : Votre tâche est de construire un arbre de décision pour déterminer si vous devez jouer au tennis ou non en fonction de la météo et de la température, en utilisant les algorithmes ID3 et CART.

Solution avec l'algorithme ID3 :

L'algorithme commence par évaluer chaque attribut disponible pour déterminer lequel serait le meilleur choix pour diviser les données. Pour ce faire, nous examinons les différentes valeurs de chaque attribut par rapport aux deux classes cibles, c'est-à-dire « Jouer" ou "Ne pas Jouer". Le tableau suivant présente le nombre de cas "Oui" et "Non" pour chaque combinaison de valeurs d'attributs :

Ciel		Température		Jouer ?
Ensoleillé	3	Chaude	3	Oui
Ensoleillé	1	Chaude	0	Non
Nuageux	2	Fraiche	2	Oui
Nuageux	0	Fraiche	0	Non
Pluvieux	1	Froide	1	Oui
Pluvieux	2	Froide	3	Non

Formule mathématique de l'entropie :

$$Entropie : H(X) = -\sum_{i=1}^{n} Pi\ log_b(Pi)$$

Où :

(Pi) est la probabilité de sélectionner au hasard un exemple dans la classe i.

(b) est la base du logarithme utilisée dans le calcul de l'entropie et du gain, elle dépend du nombre de classes (ou de catégories) dans le problème de classification. La base la plus couramment utilisée est le logarithme en base 2, ce qui donne des unités d'entropie en "bits".

Gain d'Information : Le gain d'information mesure la réduction de l'entropie après une division. Il est utilisé par ID3 pour sélectionner la caractéristique de division qui **maximise** la réduction de l'entropie.

$$Gain : Gain\ (X, ai) = H(X) - \sum \frac{|Xai = v|}{|X|} H(Xai = v)$$

Notre base de données contient 6 personnes qui ont joué au Tennis, et 3 personnes qui n'ont pas joué au Tennis, alors l'entropie est :

H(X) = - (6/9 $\log_2$ 6/9 + 3/9 $\log_2$ 3/9) ≈ 0,918…

NB: Pour simplifier le calcul de log base 2 du 6/9, vous pouvez utiliser le changement de base du logarithme. Voici comment vous pouvez le faire :

$\log_2$ (6/9) = log (6/9) / log (2)

Calcul du Gain « Ciel » :

H(X Ciel = Ensoleillé) = - (3/4 $\log_2$ 3/4 + 1/4 $\log_2$ 1/4) ≈ 0,811…

H(X Ciel = Nuageux) = - (2/2 $\log_2$ 2/2 + 0/2 $\log_2$ 0/2) = 0

H(X Ciel = Pluvieux) = - (1/3 $\log_2$ 1/3 + 2/3 $\log_2$ 2/3) ≈ 0,918…

Gain(X, Ciel) = H(X) – (4/9 H(X Ciel = Ensoleillé) + 2/9 H(X Ciel = Nuageux) + 3/9 H(X Ciel = Pluvieux))

Gain(X, Ciel) = 0,918 – (4/9 * 0,811 + 3/9 * 0,918) ≈ 0,252…

Calcul du gain « Température » :

H(X Température = Chaude) = - (3/3 $\log_2$ 3/3 + 0/3 $\log_2$ 0/3) = 0

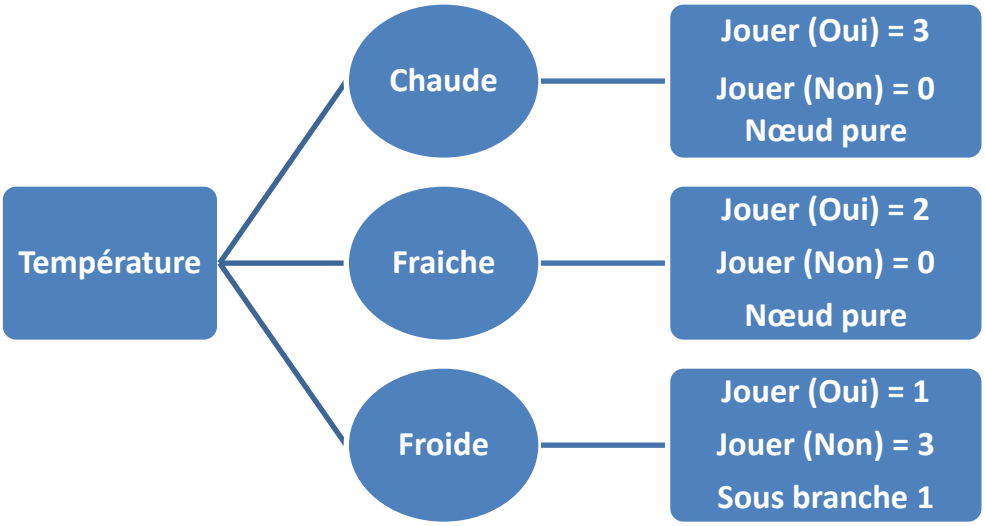
H(X Température = Fraiche) = - (2/2 $\log_2$ 2/2 + 0/2 $\log_2$ 0/2) = 0

H(X Température = Froide) = - (1/4 $\log_2$ 1/4 + 3/4 $\log_2$ 3/4) ≈ 0,811

Gain(X, Pages vues) = H(X) – (3/9 H(X Température = Chaude) + 2/9 H(X Température = Fraiche) + 4/9 H(X Température = Froide))

Gain(X, Pages vues) = 0,918-(4/9 * 0,811) ≈ 0,558…

Nous prenons le plus grand gain, c'est celui de « Température », donc il sera la racine.



Sous branche 1 (Température = Froide) :

Jour	Ciel	Température	Jouer ?
1	Ensoleillé	Chaude	Oui
2	Ensoleillé	Chaude	Oui
3	Nuageux	Chaude	Oui
4	Pluvieux	Fraiche	Oui
5	Pluvieux	Froide	Non
6	Pluvieux	Froide	Non
7	Nuageux	Froide	Oui
8	Ensoleillé	Fraiche	Oui
9	Ensoleillé	Froide	Non

Ciel		Jouer ?
Ensoleillé	0	Oui
Ensoleillé	1	Non
Nuageux	1	Oui
Nuageux	0	Non
Pluvieux	0	Oui
Pluvieux	2	Non

Notre nouvelle base de données contient 1 personne qui a joué au tennis et 3 personnes qui n'ont pas joué au tennis, répartis comme suit :

Ciel = Ensoleillé ➔ Non

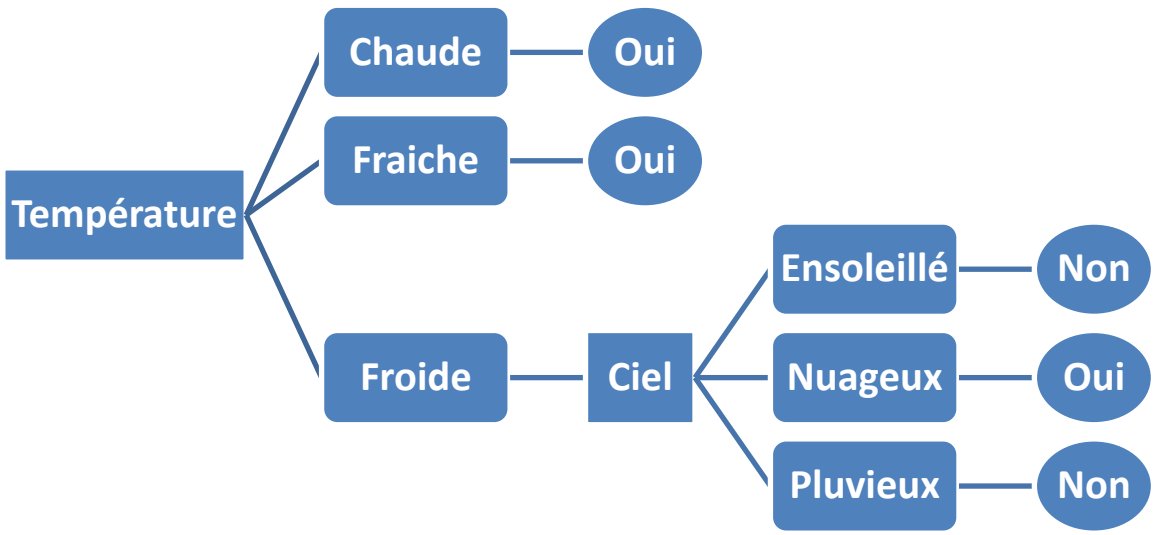
Ciel = Nuageux ➔ Oui

Ciel = Pluvieux ➔ Non

Ciel = Pluvieux ➔ Non

À ce stade, l'attribut « Ciel » est l'attribut final qui détermine si quelqu'un joue au tennis ou non. L'arbre de décision s'arrête ici, car il n'y a pas d'autres attributs à considérer pour prendre des décisions supplémentaires

L'arbre final avec l'algorithme ID3 :



Les règles de classification :

Si température = Chaude **alors** classe = **Oui**

Si température = Fraiche **alors** classe = **Oui**

Si température = Froide **^** Ciel = Ensoleillé **alors** classe = **Non**

Si température = Froide **^** Ciel = Nuageux **alors** classe = **Oui**

Si température = Froide **^** Ciel = Pluvieux **alors** classe = **Non**

Solution avec l'algorithme CART :

La même chose comme le ID3, l'algorithme CART commence par évaluer chaque attribut disponible pour déterminer lequel serait le meilleur choix pour diviser les données. Pour ce faire, nous examinons les différentes valeurs de chaque attribut par rapport aux deux classes cibles, c'est-à-dire « Jouer" ou "Ne pas Jouer". Le tableau suivant présente le nombre de cas "Oui" et "Non" pour chaque combinaison de valeurs d'attributs :

Ciel		Température		Jouer ?
Ensoleillé	3	Chaude	3	Oui
Ensoleillé	1	Chaude	0	Non
Nuageux	2	Fraiche	2	Oui
Nuageux	0	Fraiche	0	Non
Pluvieux	1	Froide	1	Oui
Pluvieux	2	Froide	3	Non

CART utilise le critère de Gini pour mesurer la pureté et sélectionner la caractéristique de division.

$$Gini - index\ (Attribut = Valeur) = \ 1 - \sum (Pi)^2$$

$$Gini - index\ (Attribut) = \sum Pv * GI(v)$$

Calcul de l'indice de Gini pour l'attribut « Ciel » :

Gini-index (Ciel = Ensoleillé) = 1 – ((3/4)² + (1/4)²) = 0,375

Gini-index (Ciel = Nuageux) = 1 – ((2/2)² + (0/2)²) = 0

Gini-index (Ciel = Pluvieux) = 1 – ((1/3)² + (2/3)²) ≈ 0,444

Gini-index (Ciel) = ((4/9) * 0,375 + (2/9) * 0 + (3/9) * 0,444) ≈ **0,315**

Calcul de l'indice de Gini pour l'attribut « Température » :

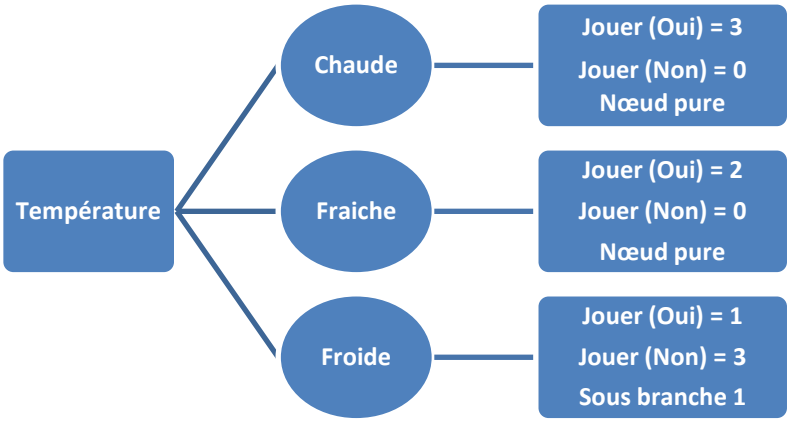
Gini-index (Température = Chaude) = 1 – ((3/3)² + (0/3)²) = 0

Gini-index (Température = Fraiche) = 1 – ((2/2)² + (0/2)²) = 0

Gini-index (Température = Froide) = 1 – ((1/4)² + (3/4)²) = 0,375

Gini-index (Température) = ((3/9) * 0 + (2/9) * 0 + (4/9) * 0,375) ≈ **0,166**

L'attribut qui minimise l'indice de Gini est sélectionné comme variable de partitionnement, dans notre cas c'est l'attribut « température » donc il sera la racine de notre arbre. Une fois l'attribut de partitionnement sélectionné, l'algorithme divise les données en sous-ensembles en fonction des différentes valeurs de cet attribut, chaque sous-ensemble représente un nœud ou une branche de l'arbre.



Sous branche 1 (Température = Froide) :

Jour	Ciel	Température	Jouer ?
1	Ensoleillé	Chaude	Oui
2	Ensoleillé	Chaude	Oui
3	Nuageux	Chaude	Oui
4	Pluvieux	Fraiche	Oui
5	Pluvieux	Froide	Non
6	Pluvieux	Froide	Non
7	Nuageux	Froide	Oui
8	Ensoleillé	Fraiche	Oui
9	Ensoleillé	Froide	Non

Ciel		Jouer ?
Ensoleillé	0	Oui
Ensoleillé	1	Non
Nuageux	1	Oui
Nuageux	0	Non
Pluvieux	0	Oui
Pluvieux	2	Non

Notre nouvelle base de données contient 1 personne qui a joué au tennis et 3 personnes qui n'ont pas joué au tennis, répartis comme suit :

Ciel = Ensoleillé ➔ Non

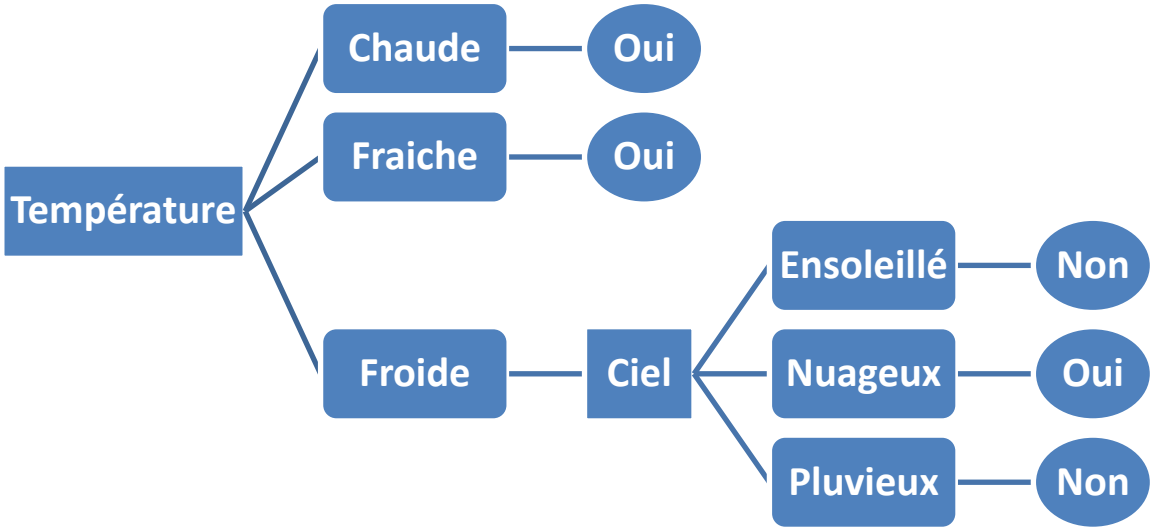
Ciel = Nuageux ➔ Oui

Ciel = Pluvieux ➔ Non

Ciel = Pluvieux ➔ Non

À ce stade, l'attribut « Ciel » est l'attribut final qui détermine si quelqu'un joue au tennis ou non. L'arbre de décision s'arrête ici, car il n'y a pas d'autres attributs à considérer pour prendre des décisions supplémentaires

L'arbre final avec l'algorithme CART :



Les règles de classification :

Si température = Chaude **alors** classe = **Oui**

Si température = Fraiche **alors** classe = **Oui**

Si température = Froide **^** Ciel = Ensoleillé **alors** classe = **Non**

Si température = Froide **^** Ciel = Nuageux **alors** classe = **Oui**

Si température = Froide **^** Ciel = Pluvieux **alors** classe = **Non**

NB:

Dans l'exercice que nous avons abordé, les deux algorithmes, ID3 et CART, ont abouti à des arbres de décision identiques, mais cette convergence n'est pas une règle universelle, et ces deux algorithmes ne produisent pas toujours les mêmes résultats. Le choix entre ID3, CART ou d'autres algorithmes de data mining repose sur plusieurs paramètres, notamment la nature des données, les objectifs du modèle, et le niveau de performance souhaité. Pour

évaluer et comparer ces algorithmes, nous faisons appel à des métriques telles que la précision, le rappel, la F-mesure, l'aire sous la courbe ROC, Etc.

6.3. Classification de Bayes

La classification de Bayes est une technique fondamentale dans le domaine du data Mining qui repose sur les principes de la théorie des probabilités et de la statistique. Elle est largement utilisée pour classer des données en catégories ou en classes en se basant sur des caractéristiques ou des attributs.

La classification de Bayes est nommée d'après le mathématicien Thomas Bayes, qui a développé le théorème de Bayes. Le but de la classification de Bayes est de prédire la classe d'un exemple en utilisant des informations probabilistes.

6.3.1. Classification Naïve de Bayes : Hypothèses et Fonctionnement

La classification naïve de Bayes est une variante qui simplifie le calcul en faisant l'hypothèse que les attributs sont indépendants les uns des autres, même si ce n'est pas toujours le cas dans la réalité. Malgré cette simplification, cette méthode est souvent très efficace.

6.3.2. Estimation des Probabilités à partir des Données d'Apprentissage

La classification de Bayes repose sur les probabilités conditionnelles. Elle calcule la probabilité qu'un exemple appartienne à une classe donnée en fonction de ses caractéristiques ou attributs.

6.3.3. Théorème de Bayes

Le théorème de Bayes est la pierre angulaire de la classification de Bayes. Il permet de calculer la probabilité conditionnelle de la classe C étant donné un ensemble d'attributs A. La formule de base est la suivante :

$$P(C|X) = \frac{P(C) * P(X|C)}{P(X)}$$

Ou :

P(C|X) : Probabilité que l'exemple appartienne à la classe C sachant que ses attributs sont X.

P(C) : Probabilité a priori de la classe C.

P(X|C) : Probabilité conditionnelle des attributs X étant donné la classe C.

P(X) : Probabilité marginale des attributs X.

Cette formule permet de mettre à jour nos croyances sur la classe d'un exemple en fonction des informations fournies par les attributs. Le résultat est une probabilité conditionnelle qui peut être utilisée pour la classification.

Il est à noter que pour la classification bayésienne naïve, nous supposons que les attributs sont mutuellement indépendants conditionnellement à la classe, ce qui simplifie le calcul des probabilités conditionnelles $P(X|C)$.

6.3.4. Classification Bayésienne Naïve

La classification de Bayes peut être simplifiée en utilisant l'hypothèse d'indépendance conditionnelle, ce qui donne naissance à la classification bayésienne naïve. Dans ce cas, nous supposons que les attributs sont mutuellement indépendants conditionnellement à la classe. Cela simplifie les calculs et permet de construire des modèles de classification plus rapidement.

6.3.5. Étapes pour Construire un Modèle de Classification de Bayes

Pour construire un modèle de classification de Bayes, suivez ces étapes :

- Collecte des Données : Rassemblez un ensemble de données avec des exemples étiquetés (attributs et classes).
- Calcul des Probabilités a priori : Calculez les probabilités a priori de chaque classe.
- Calcul des Probabilités Conditionnelles : Calculez les probabilités conditionnelles des attributs pour chaque classe.
- Appliquer le Théorème de Bayes : Utilisez le théorème de Bayes pour calculer la probabilité conditionnelle de chaque classe pour un nouvel exemple.
- Choisir la Classe : Sélectionnez la classe avec la probabilité la plus élevée comme prédiction.

6.3.6. Avantages et limites de la classification de Bayes

Avantages :

- Facile à comprendre et à mettre en œuvre.
- Peut être efficace même avec des échantillons de données relativement petits.
- Performant lorsque les attributs sont indépendants ou peu dépendants.

Limites :

- L'hypothèse d'indépendance dans la classification naïve peut être peu réaliste pour certaines données.
- Sensible aux attributs manquants.
- Peut-être ne pas être performant lorsque les attributs sont fortement dépendants.

6.3.7. Applications Pratiques de la Classification de Bayes

Diagnostic Médical : Elle peut être utilisée pour diagnostiquer des maladies en fonction des symptômes du patient et des probabilités de présence de la maladie pour différentes classes.

Exercice sur la classification de Bayes

Considérez un ensemble de données représentant différentes caractéristiques et leurs classes associées. Dans cet exercice, nous utiliserons la classification bayésienne naïve pour développer un modèle de décision basé sur cet ensemble de données. Voici les données d'entraînement :

	Cheveux	Taille	Poids	Crème Solaire	Classe
1	Blonds	Moyenne	Leger	Non	Coup de soleil
2	Blonds	Grande	Moyen	Oui	Bronzé
3	Bruns	Petite	Moyen	Oui	Bronzé
4	Blonds	Petite	Moyen	Non	Coup de soleil
5	Roux	Moyenne	Lourd	Non	Coup de soleil
6	Bruns	Grande	Lourd	Non	Bronzé
7	Bruns	Moyenne	Lourd	Non	Bronzé
8	Blonds	Petite	Leger	Oui	Bronzé

Donner le modèle de décision déduit de cette base en utilisant la classification Bayésienne naïve

Trouver les classes des exemples suivants :

	Cheveux	Taille	Poids	Crème Solaire	Classe
1	?	Petite	?	Oui	?
2	?	Grande	Moyen	?	?
3	Bruns	?	?	Non	?

Solution

- **Calcul des probabilités marginales pour chaque classe**

Nous devons calculer P(Coup de soleil) et P(Bronzé). Pour ce faire, nous comptons combien d'exemples de chaque classe que nous avons dans l'ensemble d'entraînement.

P(Coup de soleil): Il y a 3 exemples de "Coup de soleil" sur 8, donc, P(Coup de soleil) = 3/8

P(Bronzé) : Il y a 5 exemples de "Bronzé" sur 8, donc P(Bronzé) = 5/8

- **Calcul des probabilités conditionnelles pour chaque attribut donné a une classe**

Nous allons calculer les probabilités conditionnelles pour chaque attribut (Cheveux, Taille, Poids, Crème Solaire) étant donné chaque classe (Coup de soleil ou Bronzé). Nous allons utiliser la classification bayésienne naïve, ce qui signifie que nous supposons que les attributs sont indépendants conditionnellement à la classe. Par exemple P (Cheveux = Blonds).

Calculons ces probabilités conditionnelles :

P(Cheveux = Blonds | Coup de soleil): Il y a 2 exemples de "Coup de soleil" avec des cheveux blonds sur 3, donc :

P(Cheveux = Blonds | Coup de soleil) = 2/3

P(Cheveux = Blonds | Bronzé) : Il y a 1 exemple de "Bronzé" avec des cheveux blonds sur 5, donc :

P(Cheveux = Blonds | Bronzé) = 1/5.

Et ainsi de suite pour les autres attributs et classes.

Tableau des probabilités :

		P(Classe = Coup de soleil) = 3/8	P(Classe = Bronzé) = 5/8
Cheveux	Blonds = 4	2/3	2/5
	Bruns = 3	**0/3**	3/5
	Roux = 1	1/3	**0/5**
Taille	Petite = 3	1/3	2/5
	Moyenne = 3	2/3	1/5
	Grande = 2	**0/3**	2/5
Poids	Leger = 2	1/3	1/5
	Moyen = 3	1/3	2/5
	Lourd = 3	1/3	2/5
Crème solaire	Oui = 3	**0/3**	3/5
	Non = 5	3/3	2/5

Dans l'ensemble de données, nous observons la présence de valeurs nulles, ce qui peut poser des problèmes lors des calculs statistiques. Pour éviter ces problèmes, nous utilisons une technique appelée "estimateur de Laplace" (ou "lissage de Laplace"). Cette technique nous permet de remplacer les valeurs nulles par des valeurs non nulles en ajoutant une petite quantité à chacune des occurrences de l'attribut concerné (l'attribut 1/nombres de valeurs de l'attribut). En appliquant cette méthode, nous obtenons un nouveau tableau de données qui tient compte de toutes les valeurs de l'attribut, y comprenant celles qui étaient initialement nulles.

		P(Classe = Coup de soleil) = 3/8	P(Classe = Bronzé) = 5/8
Cheveux	Blonds = 4	2/3 → (1+2)/(3+3) = 3/6	2/5→ (1+2)/(3+5) = 3/8
	Bruns = 3	0/3→ (1+0)/(3+3) = 1/6	3/5→ (1+3)/(3+5) = 4/8
	Roux = 1	1/3→ (1+1)/(3+3) = 2/6	0/5→ (1+0)/(3+5) = 1/8
Taille	Petite = 3	1/3→ (1+1)/(3+3) = 2/6	2/5
	Moyenne = 3	2/3→ (1+2)/(3+3) = 3/6	1/5
	Grande = 2	0/3→ (1+0)/(3+3) = 1/6	2/5
Poids	Leger = 2	1/3	1/5
	Moyen = 3	1/3	2/5
	Lourd = 3	1/3	2/5
Crème solaire	Oui = 3	0/3→ (1+0)/(2+3) = 1/5	3/5
	Non = 5	3/3→ (1+3)/(2+3) = 4/5	2/5

- **Utilisation du modèle de Bayes naïf pour classer les exemples de test**

Maintenant, nous pouvons utiliser ces probabilités conditionnelles pour classer les exemples de test en utilisant le modèle de Bayes naïf.

Pour chaque exemple de test, nous allons calculer la probabilité conditionnelle pour chaque classe, puis choisir la classe avec la probabilité la plus élevée.

Exemple de test 1 : Cheveux : ?, Taille : Petite, Poids : ?, Crème Solaire : Oui, Classe : ?

- P(Coup de soleil |Taille = Petite, Crème Solaire = Oui) =

P(Coup de soleil) * P(Taille=petite|Coup de soleil) * P(Crème solaire=Oui|Coup de soleil) = 3/8*2/6*1/5 = 6/240 = 0,025

- P(Bronzé | Taille = Petite, Crème Solaire = Oui) =

P(Bronzé) * P(Taille=petite|Bronzé) * P(Crème solaire=Oui|Bronzé) = 5/8*2/5*3/5=30/200 = 0,15

- 0,15 > 0,025 Donc Classe = Bronzé

Exemple de test 2 : Cheveux : ?, Taille : Grande, Poids : Moyen, Crème Solaire : ?, Classe : ?

- P(Coup de soleil | Taille = Grande, Poids = Moyen) =

P(Coup de soleil) * P(Taille=Grande|Coup de soleil) * P(Poids= Moyen|Coup de soleil) = 3/8*1/6*1/3 = 3/144 = 0,02

- P(Bronzé |Taille = Grande, Poids = Moyen) =

P(Bronzé) * P(Taille=Grande|Bronzé) * P(Poids = Moyen|Bronzé) = 5/8*2/5*2/5 = 20/200 = 0,1

- 0,1 > 0,02 Donc Classe = Bronzé

Exemple de test 3 : Cheveux : Bruns, Taille : ?, Poids : ?, Crème Solaire : Non, Classe : ?

- P(Coup de soleil | Cheveux = Bruns, Crème solaire = Non) =

P(Coup de soleil) * P(Cheveux = Bruns |Coup de soleil) * P(Crème solaire = Non |Coup de soleil) = 3/8*1/6*4/5 = 12/240 = 0,05

- P(Bronzé | Cheveux = Bruns, Crème solaire = Non) =

P(Bronzé) * P(Cheveux = Bruns) * P(Crème solaire = Non) = 5/8*4/8*2/5 = 40/320 = 0,125

- 0,125 > 0,05 Donc Classe = Bronzé

	Cheveux	Taille	Poids	Crème Solaire	Classe
1	?	Petite	?	Oui	**Bronzé**
2	?	Grande	Moyen	?	**Bronzé**
3	Bruns	?	?	Non	**Bronzé**

6.4. Machines à Vecteurs de Support (SVM)

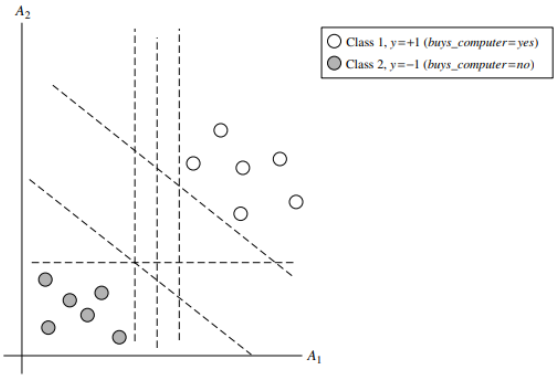
Les Machines à Vecteurs de Support (SVM) sont une technique d'apprentissage automatique largement utilisée pour la classification et la régression. Elles ont été développées pour la première fois par Vladimir Vapnik dans les années 1960. Les SVM sont particulièrement efficaces pour résoudre des problèmes de classification dans des espaces de grande dimension. Ils sont souvent utilisés dans des domaines tels que la vision par ordinateur, la classification de textes, la bioinformatique et plus encore.

6.4.1. Principes de base

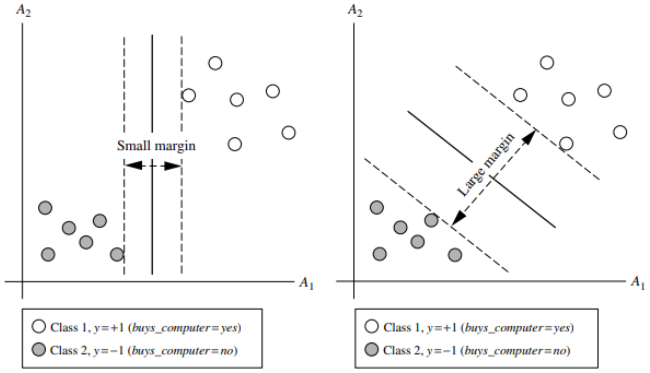
Les SVM sont basés sur le concept de trouver un hyperplan qui maximise la marge entre les classes de données (une ligne en 2D, un plan en 3D, etc.). L'hyperplan est défini comme une frontière de décision qui sépare les données en deux classes. Le but est de trouver l'hyperplan optimal qui maximise la marge tout en minimisant l'erreur de classification.

6.4.2. SVM Linéaires (Cas Séparables)

- Séparation Linéaire : L'idée de base des SVM est de trouver un hyperplan de séparation entre les classes. Cela signifie qu'un SVM cherche à tracer une ligne ou un plan qui maximise la marge entre les points de données des différentes classes tout en minimisant l'erreur de classification.



- Marges : La marge est la distance entre l'hyperplan de séparation et les exemples de données les plus proches de chaque classe. L'objectif est de maximiser cette marge en utilisant la formule suivante : Marge = 2/||W||
Ou ||W|| est la norme du vecteur de poids W.



- Vecteurs de Support : Les vecteurs de support sont les exemples de données les plus proches de l'hyperplan de séparation. Ce sont ces points qui déterminent la position de l'hyperplan et donc la performance du modèle.

Fonction de décision linéaire

La fonction de décision linéaire dans ce cas peut être exprimée comme suit :

$f(x) = w \cdot X + b$

où :

$f(x)$: est la fonction de décision.

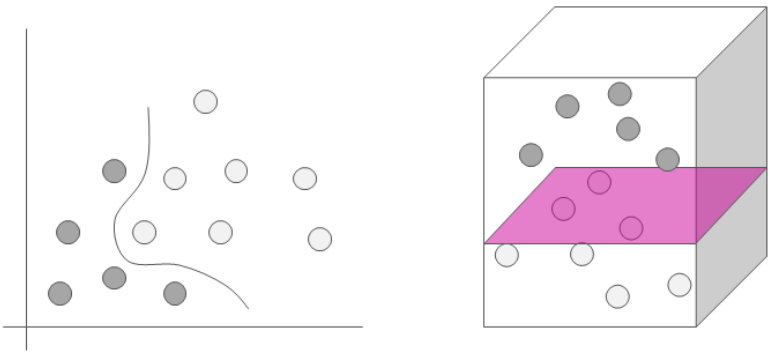
X : est le vecteur d'entrée.

W : est le vecteur de poids qui définit l'orientation de l'hyperplan.

b : est le biais.

6.4.3. SVM Non Linéaires (Cas Non Séparable)

Dans de nombreux cas réels, les données ne sont pas linéairement séparables. Pour résoudre de tels problèmes, les SVM utilisent une technique appelée "noyau" (noyau) pour projeter les données dans un espace de dimension supérieure où elles sont linéairement séparables. Les noyaux sont des fonctions qui mesurent la similarité entre deux vecteurs de données.



Types de noyaux

Les SVM peuvent utiliser différents types de noyaux, tels que le noyau linéaire, le noyau polynomial, le noyau gaussien (RBF), etc. Le choix du noyau dépend souvent de la nature des données et de la complexité du problème.

6.4.4. Avantages et inconvénients des SVM

Avantages

Efficacité dans les espaces de grande dimension

Bonne généralisation

Flexibilité avec les noyaux

Gestion des données déséquilibrées

Inconvénients du SVM

Sensibilité aux hyper paramètres

Complexité informatique

Les SVM ne sont pas aussi faciles à interpréter que certains autres modèles.

Exercice sur le SVM

Vous avez un ensemble de données bidimensionnelles contenant deux classes, les points rouges (classe A) et les points bleus (classe B) :

Classe A (points rouges) : (1, 2), (2, 3), (3, 3)

Classe B (points bleus) : (6, 6), (7, 5), (8, 7)

Votre tâche est de trouver l'hyperplan de séparation entre ces deux classes en utilisant les linéaires SVM. Vous devrez également déterminer les vecteurs de support et la marge.

Solution

……

Conclusion

Ce chapitre se concentre sur la classification supervisée, une technique de data Mining essentielle pour attribuer des étiquettes à des données en fonction de leurs caractéristiques. Il explore en détail des méthodes clés, KNN et les arbres de décision, en mettant en évidence leurs avantages et inconvénients, ainsi que leurs applications pratiques. De plus, il plonge dans la classification bayésienne et les SVM, offrant une compréhension approfondie de ces approches. Les exercices pratiques associés à chaque méthode offrent une occasion précieuse d'appliquer ces concepts dans des contextes réels, renforçant ainsi la maîtrise de la classification supervisée.