Data Mining

NIVEAU IV
SPECIALITE : Data Sciences et Master Informatique
Année académique : 2022-2023

UMa
ENSPM
INFOTEL

Touza Isaac
isaac_touza@outlook.fr

Informations générales

- **Code UE** : DSC 438 - MIF 438
- **Intitulé de L'UE** : Data Mining
- **Crédit** :
- **Durée** : 45h
  - CM : 10h
  - TP : 15h
  - TPE : 5h
  - TD : 15h
- **Évaluations** :
  - CC (théorique ou pratique) : 2h
  - Examen (théorique ou pratique) : 2h
  - Rattrapage (théorique ou pratique) : 2h

Références

- Mohamed NEMICHE,Data mining Master MASI, Faculté des Sciences d'Agadir(2014/2015)
- Bing Liu, Web Data Mining Exploring Hyperlinks, Contents, and Usage Data Second Edition, 2011
- Hanane Ezzikouri & Mohammed Erritali, Web Mining, Extraction des connaissances à partir des données du Web.
- HADJ-TAYEB Karima , Le Data Mining appliqué au WEB, 2018
- ZDR AVKO MARKOV AND DANIEL T. L AROSE , DATA MINING THE WEB, Uncovering Patterns in Web Content, Structure, and Usage , 2007

Objectifs du cours

Objectif général :

Apprendre à maîtriser les techniques et modèles du data Mining, afin notamment de pouvoir les utiliser/appliquer dans des situations réelles.

Objectifs spécifiques :

- Rechercher et collecter les données
- Nettoyer et analyser les données collectées
- Classer et recommander des informations/données.
- Prendre ou dégager une décision à partir des résultats des analyses.

Près-requis et consignes

Près-requis

- Utilisation des bases de données
- Connaissances du langage Python
- Théorie de graphe

Consignes

- Assister à tous les cours
- Être attentifs et actifs pendant le cours
- Faire tous les exercices de TD et TP
- Refaire plusieurs fois les exercices d'applications
- **Ne manquez jamais un TP**

Plan du cours I

1. **Chapitre 1 : Introduction au data mining**
   - Data Mining : Définitions et tâches
   - Data Mining : Objectifs
   - Axes du Data Mining
   - Processus du Data Mining
   - Types de données
   - Techniques du Data Mining
   - Domaines d'applications
2. **Chapitre 2 : Collecte et pré-traitement des données**
   - Collecte des données
     - Introduction
     - Les sources des données collectées
     - Outils de collecte des données
     - Cas du Web
   - Pré-traitement des données
     - Définition

Plan du cours II

- Techniques de pré-traitement des données
- Bibliothèques python pour le pré-traitement des données
- Etapes de pré-traitement des données

3 Chapitre 3 : Les règles d'association

- Introduction
- Les transactions
- Concepts de base des règles d'association
- Algorithme Apriori
- FP-Growth

4 Chapitre 4 : La Classification supervisée

- Introduction
- Formes de classification
- Types de classification
- Processus de classification
- Pondérations ou calcul des fréquences

Plan du cours III

- Étiquetage du document
- Extractions des termes
- Algorithmes de classification supervisée
  - ID3
  - C4.5
  - Knn(k-nearest neighbor)
  - Bayes naïf
- Évaluation de la performance de classification
- Problèmes de classification

5 Chapitre 5 : Le Clustering

- Définitions et objectifs
- Domaines d'application
- Concepts de base du clustering
- Méthodes de clustering
- K-means

CHAPITRE 1

Introduction au Data Mining

Introduction

- Aujourd'hui, des milliards de données sont collectées chaque jour dans le monde.
- L'accroissement exponentiel des utilisateurs d'Internet, entraine la massification des données
- L'accès à des informations utiles et pertinentes devient donc de plus en plus complexe.
- Par ailleurs cette augmentation massive des données crée chez les utilisateurs plusieurs besoins (classification , recherche d'information, relation entre objets...)
- Ainsi, des outils performants devront être mis à la disposition des utilisateurs et des entreprises pour répondre de manière performante et intelligente à leurs besoins : d'où la naissance du **data Mining**

Introduction

Définitions

data mining ou fouille de données

- Extraction d'informations intéressantes (non triviales, implicites, probablement inconnues,et potentiellement utiles) à partir d'une grande bases des données
- Ensemble de techniques d'exploration de données permettant d'extraire d'une base de données des connaissances sous la forme de modèles afin de **décrire** le comportement actuel des données et/ou **prédire** le comportement futur des données.

Définitions

Le **data mining** se sert des techniques et méthodes :

- mathématiques,
- statistiques,
- et algorithmiques

pour extraire une connaissance ou une décision à partir des données élémentaires disponibles dans une entrepôt des données (data warehouse ou big data)ou d'une base des données quelconque : On parle de **(Knowledge Discovery in Databases – KDD)**

Définitions

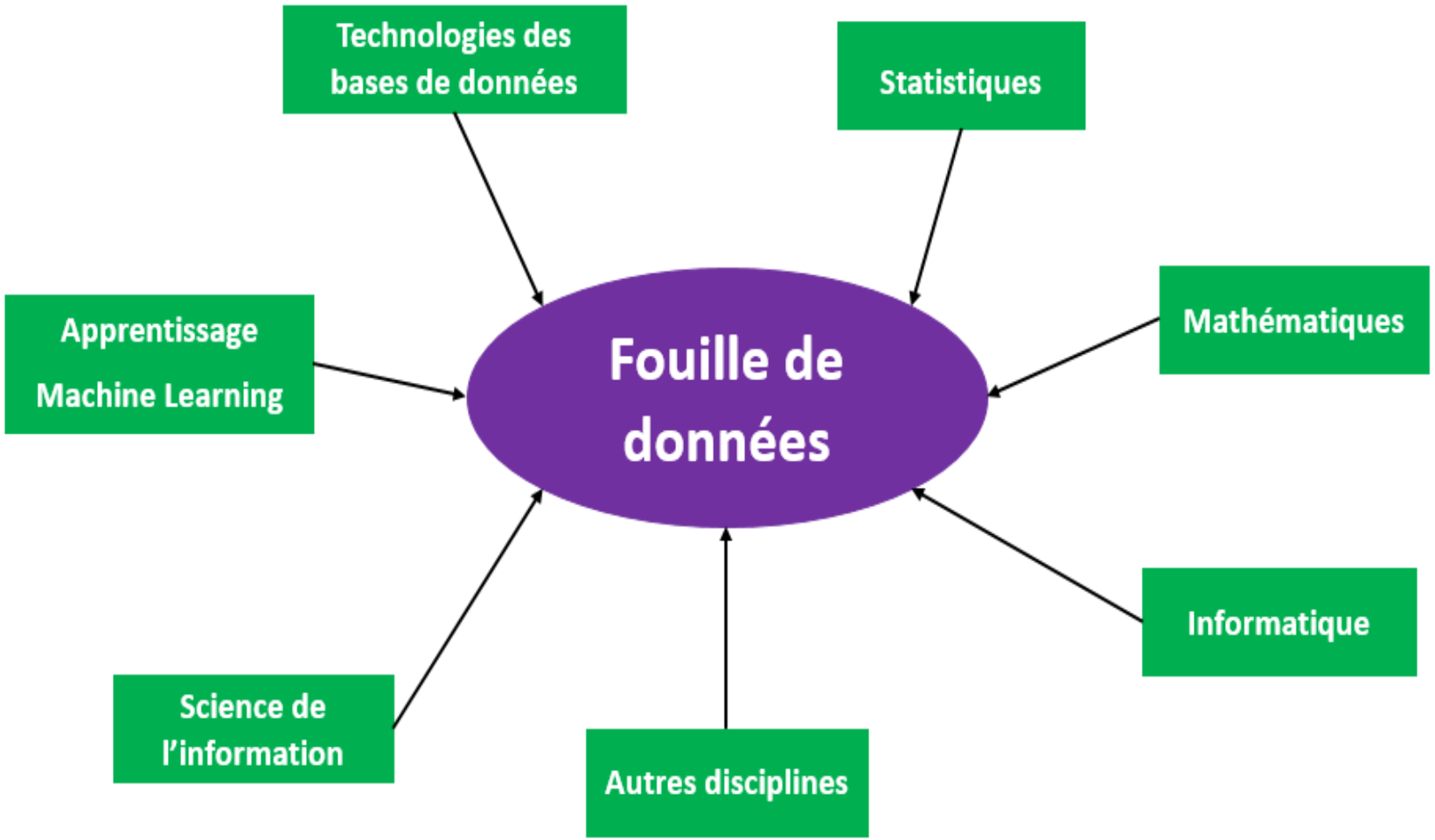
Ce qui n'est pas de Data Mining

- En générale Data Mining n'est pas basé sur des modèles déterministes.
- Un modèle déterministe ne fait intervenir aucune variable aléatoire. Les relations entre variables sont strictement fonctionnelles.

Ce qui est de la fouille de données

- En générale Data Mining est basé sur des modèles probabilistes.
- Un modèle probabiliste est un modèle mathématique qui nous aide à prévoir le comportement des futures répétitions d'une expérience aléatoire en se basant sur l'estimation d'une probabilité d'apparition de cet évènement concret

Définitions



Data Mining vs KDD

- Habituellement les deux termes sont interchangés.
- KDD (Knowledge Discovery in Databses) : C'est le processus de trouver information utiles à partir de données.
- Data Mining : C'est l'utilisation des algorithmes pour extraire une information
- Data Mining : C'est une partie du processus KDD
- Data Mining : Le cœur du processus d'extraction de connaissances

Statistique vs Data mining

1. En statistique :
   - Quelques centaines d'individus
   - Quelques variables
   - Fortes hypothèses sur les lois statistiques
   - Importance accordée au calcul
   - Échantillon aléatoire.
2. En Data mining
   - Des millions d'individus
   - Des centaines de variables
   - Données recueillies sans étude préalable
   - Nécessité de calculs rapides
   - Corpus d'apprentissage

Data Mining vs Data Warehouse

- **Data warehouse** est un entrepôt de données d'une entreprise. Ces données sont stockées dans une ou plusieurs base de données relationnelle et sont accessibles par toutes les applications orientées aide à la décision.
- **Data Warehouse** et **Data Mining** sont deux choses très différentes. DataWarehouse est usuellement le point le départ de Data Mining.
- Data Warehouse et Data Mining sont des parties du processus KDD.

Data Mining et Machine Learning

- Machine Learning : C'est un sujet de l'intelligence artificielle (IA) qui s'occupe de la façon d'écrire des programmes qui peuvent apprendre.
- Dans Data Mining machine learning est habituellement utilisés pour la réalisation de certaines tâches telles que la prédiction, la classification, etc.

Tâches du data mining

1. **Classification** : on examine les caractéristiques d'un nouvel objet pour l'affecter à une classe prédéfinie.
2. **Estimation** : Elle porte sur des variables continues(numérique) et établit le lien entre une combinaison de critères
3. **Segmentation** : déterminer quelles observations vont naturellement ensemble sans privilégier aucune variable.
4. **Prédiction** : cette fonction est proche de la classification ou de l'estimation, mais les observations sont classées selon un comportement ou une valeur estimée futurs.
5. **Recherche d'association** : rechercher et découvrir dans une grande base des données les relations et/ou les règles cachées qu'il existe entre les attributs (variables) et qu'ils sont utiles pour la prise de décision.

Pourquoi le Data Mining ?

Quelques raisons d'être du data Mining sont :

- **L'explosion des données** : Les outils de collecte automatique des données et les bases de données conduisent à d'énormes masses de données stockées dans des entrepôts
- **Submergés par les données, manque de connaissance !** :
- **Données en trop grandes quantités pour être traitées manuellement ou par des algorithmes classiques** : Nombre d'enregistrements en million ou milliard, Donnée de grande dimension (trop de champs/attributs/caractéristiques), Sources de données hétérogènes
- **Nécessité économique** : e-commerce, Haut degré de concurrence, personalisation, fidélisation de la clientèle, market segmentation

Data Mining : Objectifs

Le Data Mining présente deux principaux objectifs que sont :

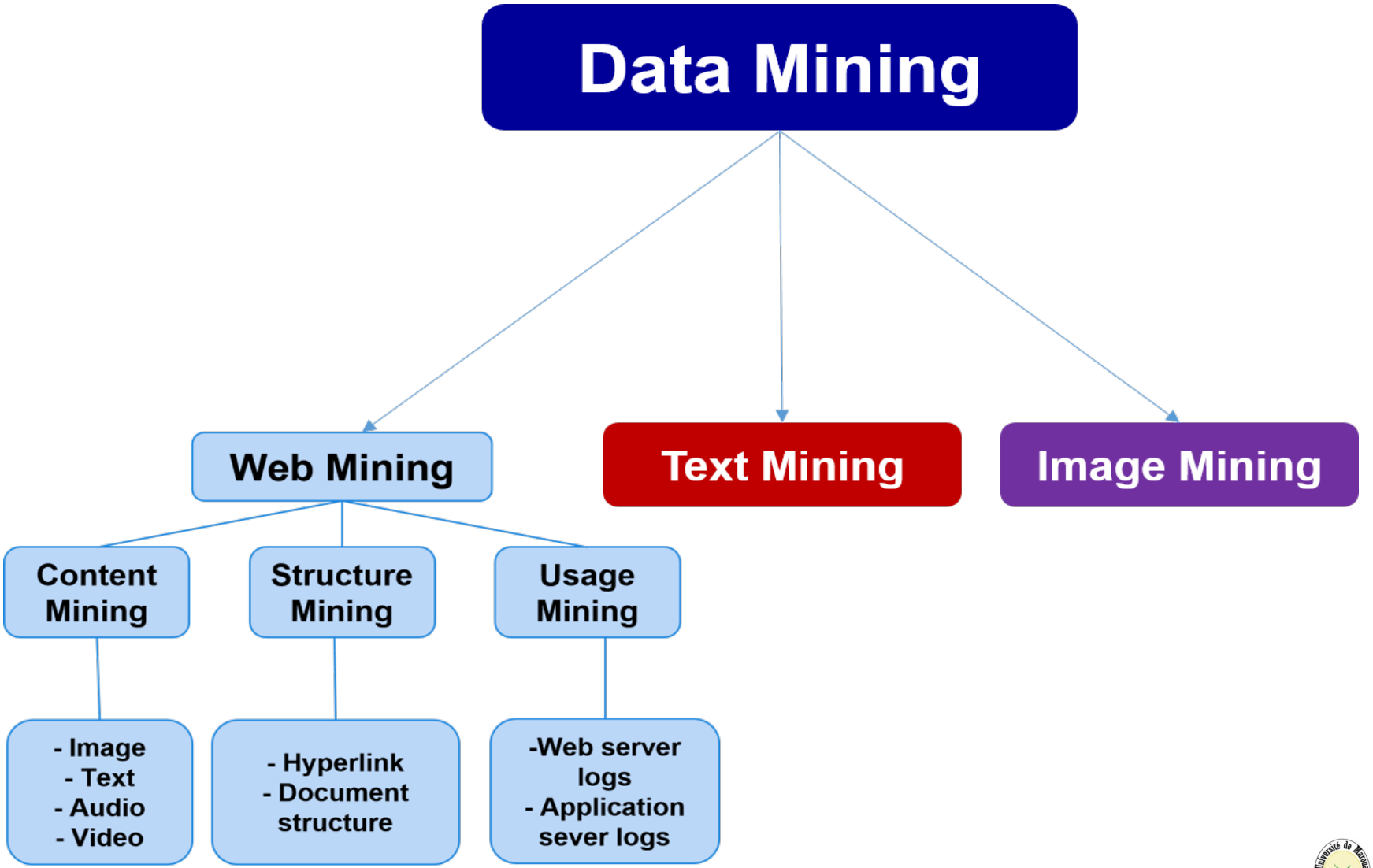
1. **Décrire les comportements des données** : La description vise à comprendre les données en les résumant et en les expliquant de manière significative. Les techniques de description comprennent la visualisation de données, la réduction de la dimensionnalité, le clustering et la règle d'association. L'objectif principal est de découvrir des modèles et des relations intéressantes dans les données qui peuvent aider les analystes à mieux comprendre les données.
2. **Prédire les comportements futurs des données** : Elle vise à utiliser les modèles découverts pour faire des prédictions sur des données inconnues. Les techniques de prédiction comprennent la régression, les arbres de décision, les réseaux neuronaux, les algorithmes de classification, etc.

Axes du Data Mining I

Selon ces cibles, la fouille des données peut être divisée en trois types :

1. la fouille des données du web : **Web Mining**
2. la fouille des données textuelles : **Text Mining**
3. la fouille des images : **Image Mining**

Axes du Data Mining II



Processus du Data Mining

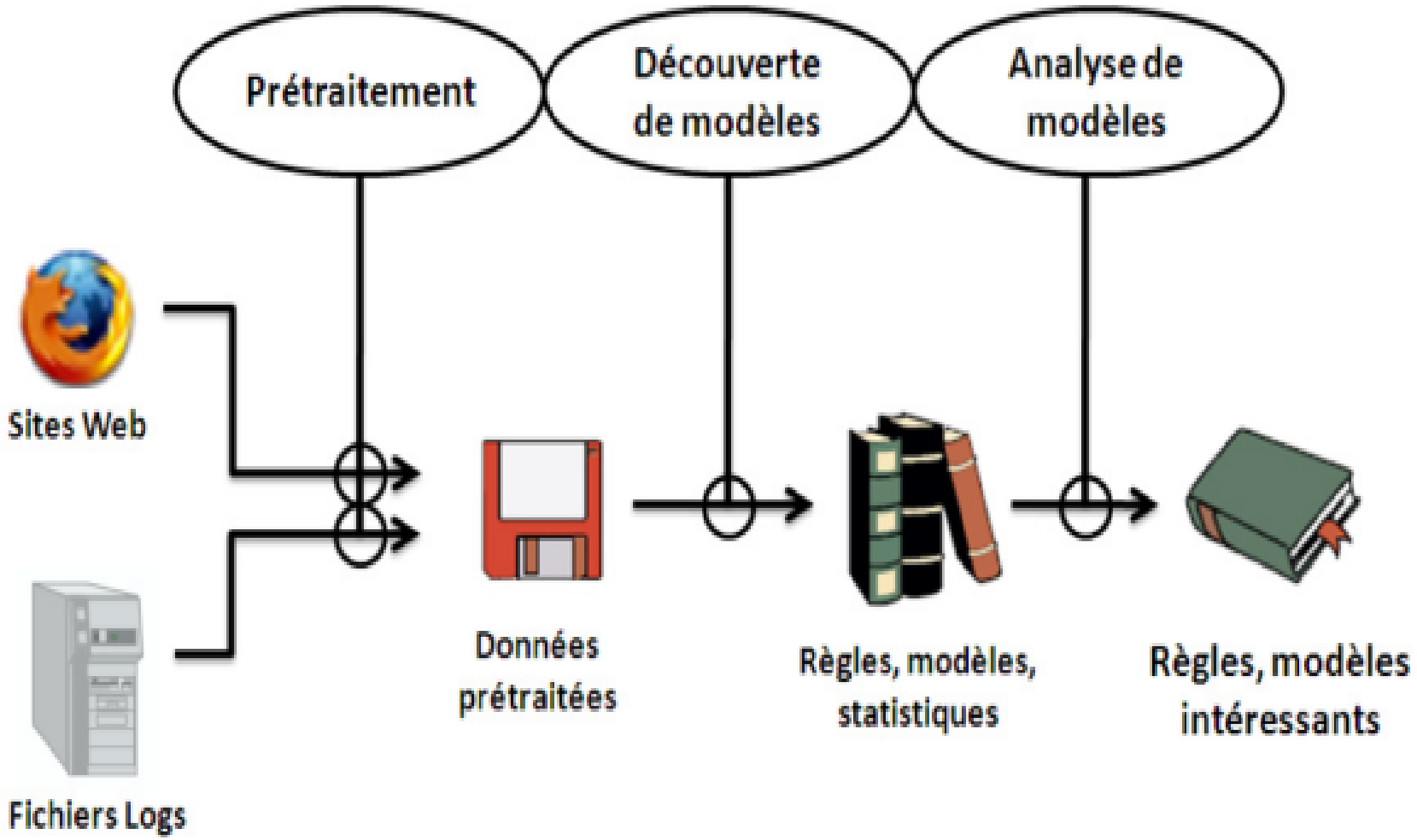
Le processus du Data Mining se déroule en quatre étapes telles que décrites ci-dessous :

1. **Collecte des données** : Cette étape consiste à collecter les données brutes provenant de différentes sources, telles que des fichiers texte, des bases de données, des fichiers Excel, etc.
2. **Le Pré-traitement des données collectées** : Cette étape vise à nettoyer les données collectées en supprimant les données manquantes, les valeurs aberrantes et les données dupliquées.
3. **Exploration des données** Cette étape implique l'analyse exploratoire des données pour identifier les tendances, les modèles et les relations entre les différentes variables
4. **La transformation des données** Cette étape consiste à transformer les données en un format utilisable pour l'analyse en utilisant des techniques telles que la normalisation, l'encodage des variables catégorielles, etc.

Processus du Data Mining

5. **Modélisation et extraction d'informations** : Cette étape consiste à construire des modèles statistiques ou des algorithmes d'apprentissage automatique pour extraire des informations utiles à partir des données.
6. **Évaluation des modèles** : Cette étape consiste à évaluer les performances des modèles construits en utilisant des métriques telles que la précision, le rappel, la F-mesure, etc.
7. **Interprétation des résultats** : Cette étape implique l'interprétation des résultats de l'analyse pour en tirer des conclusions utiles et des recommandations pour la prise de décision.
8. **Mise en œuvre** : Cette dernière étape consiste à mettre en œuvre les résultats de l'analyse dans l'entreprise ou l'organisation en utilisant les connaissances et les recommandations pour améliorer les processus, les produits ou les services.

Processus du Data Mining



Les types de données du Data Mining

Les types de données exploitées en data Mining sont :

- Les graphiques
- Les textes
- Les images
- Les vidéos
- HTML
- XML
- les hyperliens
- les adresses IP
- les adresses URL
- fichiers logs...

Techniques du Data Mining

Ils permettent d'accomplir des analyses qui peuvent être regroupées en deux catégories :

1. **Les techniques descriptives :**consiste à trouver les caractéristiques générales relatives aux données fouillées .
   - **Classification**
     - ID3
     - C4.5
     - Classifieur Bayésien
     - Knn(k-nearest neighbor)
   - **Association**
     - Apriori
     - FP-Growth
     - Eclat
     - OPUS

Techniques de Data Mining

2. **Les techniques prédictives :** Consiste à utiliser certaines variables pour prédire les valeurs futures inconnues de la même variable ou d'autres variables.
   - **Estimation**
     - EM
     - EDA(Estimation of Distribution Algorithms)
     - SEM(Stochastique,Estimation,Maximisation)
   - **Clustering**
     - k-means
     - EM (Maximisation d'espérance)
     - OPTICS
     - BDSCAN
   - **Prévision**
     - Regression logistique
     - knn(K plus proches voisins)
     - Bayes naïf
     - SVM(Machine à support de vecteur)

Domaines d'applications

- **Domaine des assurances :**
  - analyse des risques (caractérisation des clients à hauts risques, etc.)
  - automatisation du traitement des demandes (diagnostic des dégâts et détermination automatique du montant des indemnités)
- **Services financiers :**
  - Attribution de prêts automatisés, support à la décision de crédit
  - Détection de fraude
  - Marketing ciblé
- **Grande distribution :**
  - profils de consommateurs et modèles d'achats
  - marketing ciblé

Domaines d'applications

- **Médecine :**
  - Aide au diagnostic
- **Internet :**
  - Personnalisation des publicités affichées
  - Optimisation des sites web
- **Ticket de caisse :**
  - Liste des achats.
  - Heure de passage en caisse.
  - Quels sont les articles le plus souvent achetés ensemble ?
  - Promotions groupées, agencement du magasin . .

CHAPITRE 2

Collecte et prétraitement des données

Introduction

- L'étape de collecte des données est la première étape du processus de data mining. Elle consiste à recueillir les données brutes à partir de diverses sources, telles que des fichiers texte, des bases de données, des fichiers Excel, des fichiers de log, des flux de données en direct, etc.
- Le succès de l'ensemble du processus de data mining dépend en grande partie de la qualité et de la quantité des données collectées. Par conséquent, il est important de veiller à ce que les données collectées soient pertinentes et complètes, car cela garantira que l'analyse des données est précise et pertinente pour l'objectif visé.

Sources des données

Voici les principales sources de données utilisées pour le data mining :

1. **Les fichiers de données stockés sur les systèmes de stockage** : Les entreprises stockent souvent des données importantes dans des fichiers stockés sur des systèmes de stockage tels que les systèmes de fichiers distribués, les systèmes de stockage cloud et les disques durs. Ces fichiers contiennent souvent des informations précieuses telles que des données de vente, des données de transactions, des données de suivi de l'inventaire, etc.
2. **Les bases de données relationnelles** : Les bases de données relationnelles stockent des données dans des tables organisées en colonnes et en lignes. Les bases de données peuvent contenir des informations sur les clients, les ventes, les stocks, les employés, les fournisseurs, etc.

Sources des données

3. **Les fichiers log** : Les fichiers log sont des fichiers générés par les applications et les systèmes d'exploitation pour enregistrer les événements tels que les erreurs, les avertissements et les informations sur l'utilisation. Les fichiers log contiennent souvent des informations précieuses sur les performances du système, les erreurs et les activités des utilisateurs.
4. **Les flux de données en direct** : Les flux de données en direct sont des données qui sont générées en temps réel à partir de diverses sources telles que les capteurs, les caméras, les enregistreurs de données, etc. Les flux de données en direct peuvent fournir des informations précieuses sur les tendances en temps réel, les anomalies et les modèles.

Sources des données

5. **Les données provenant de sources tierces** : Les données provenant de sources tierces, telles que les réseaux sociaux, les sources d'informations publiques, les données environnementales, les données démographiques, etc., peuvent fournir des informations précieuses pour le data mining.

Sources des données

Remarques

1. Une fois les données collectées, il est **important de veiller à ce qu'elles soient stockées dans un format qui facilite leur traitement et leur analyse**. Cela peut inclure la conversion de fichiers de données en un format standard tel que CSV (Comma Separated Values), la normalisation des données et la suppression des données redondantes ou non pertinentes.
2. En suivant ces bonnes pratiques, vous serez en mesure de collecter des données de qualité et de les stocker de manière appropriée pour le processus de data mining.

Sources des données

Important !

- Il est également important de noter que les données collectées doivent être conformes aux lois et réglementations en matière de protection des données personnelles. Les entreprises doivent se conformer à des normes strictes pour assurer la confidentialité et la sécurité des données collectées, afin d'éviter tout risque de violation de la vie privée des individus.
- Il est également essentiel de définir les objectifs de l'analyse des données dès le début du processus de collecte de données. Cela permet de déterminer quelles données sont nécessaires pour atteindre les objectifs et quelles sont les sources de données appropriées pour les obtenir

Outils de collecte des données

Il existe plusieurs outils de collecte de données pour collecter et extraire des données à partir de diverses sources. Voici quelques-uns des outils les plus couramment utilisés :

- **Google Forms** : Google Forms est un outil de collecte de données gratuit qui permet de créer des formulaires personnalisés pour collecter des données auprès des utilisateurs. Les données collectées peuvent être exportées dans différents formats, notamment CSV et Excel.
- **SurveyMonkey** : SurveyMonkey est un outil de collecte de données en ligne qui permet de créer des sondages et des enquêtes en ligne pour collecter des données auprès des participants. Les résultats peuvent être exportés dans différents formats, y compris CSV et Excel.

Outils de collecte des données

- Les **navigateurs** gratuites qui permet de collecter des données à partir de sites web en utilisant des techniques de web scraping. L'outil peut extraire des données à partir de pages web, de tableaux et de fichiers PDF.
- **Scrapy** : Scrapy est un framework open source de web scraping en Python. Il permet de collecter des données à partir de sites web en utilisant des scripts Python personnalisés pour extraire les données souhaitées.

Cas du Web

Python est un langage de programmation populaire utilisé dans l'analyse des données et le data mining. Il propose plusieurs bibliothèques et frameworks qui facilitent la collecte des données à partir de différentes sources de données. Voici quelques-unes des méthodes courantes pour collecter des données en Python :

- Utilisation de la bibliothèque **requests** pour récupérer des données à partir d'API web : Il suffit de spécifier l'URL de l'API et d'envoyer une requête GET ou POST pour récupérer les données.

Exemple 1 : récupération des données à partir de l'API Github

```
import requests
url = "https ://api.github.com/users/octocat/repos"
response = requests.get(url)
data = response.json()
```

Cas du Web

- Utilisation de la bibliothèque **BeautifulSoup** :c'est un outil de parsing HTML et XML en Python. Elle permet de récupérer les données à partir d'un site web en spécifiant des balises HTML ou des attributs de balises.

Exemple 2 : extraire les titres des articles d'un site web :

```
import requests
from bs4 import BeautifulSoup
url = "https ://www.example.com"
response = requests.get(url)
soup = BeautifulSoup(response.content, 'html.parser')
articles = soup.find_all('h2', class_='article-title')
titles = [article.text for article in articles]
```

Cas du Web

- Utilisation de la bibliothèque **pandas** pour importer des données à partir de fichiers CSV , Excel,SQL ou de bases de données NoSQL

Exemple 3 : importer des données à partir d'un fichier CSV :

```
import pandas as pd
data = pd.read_csv('data.csv')
```

Pré-traitement des données

Définition

Le pré-traitement des données est une étape essentielle dans le processus de data mining car les données brutes peuvent contenir des erreurs, des données manquantes, des valeurs aberrantes et d'autres problèmes qui peuvent fausser les résultats de l'analyse.

Définition

Le pré-traitement des données est un ensemble de techniques et de procédures utilisées pour nettoyer, transformer et préparer les données brutes en vue de leur analyse ultérieure.

1. **Nettoyage des données** :
   Cette étape consiste à éliminer les données erronées, manquantes ou dupliquées des données brutes. Il existe plusieurs techniques courantes de nettoyage des données, notamment :

- **La suppression des données manquantes** : les données manquantes peuvent être éliminées ou remplacées par des valeurs moyennes ou médianes.
- **La détection et la suppression des valeurs aberrantes** : les valeurs aberrantes peuvent être supprimées ou remplacées par des valeurs moyennes ou médianes.
- **La déduplication des données** : les données en double peuvent être supprimées ou fusionnées pour éviter les doublons.

2. **Transformation des données** : Elle est une autre étape importante du pré-traitement des données. Cette étape consiste à transformer les données brutes en une forme plus appropriée pour l'analyse ultérieure. Les techniques courantes de transformation des données comprennent :
   - **La normalisation** : cette technique est utilisée pour ramener toutes les valeurs d'un ensemble de données à une échelle commune.
   - **La discrétisation** : cette technique consiste à convertir des données continues en données discrètes en créant des intervalles de valeurs.
   - **La réduction de la dimensionnalité** : cette technique est utilisée pour réduire le nombre de variables ou de caractéristiques dans un ensemble de données, ce qui peut simplifier l'analyse et réduire la complexité.

3. **Sélection des caractéristiques** :
   La sélection des caractéristiques est une étape importante du pré-traitement des données qui consiste à sélectionner les variables ou les caractéristiques les plus pertinentes pour l'analyse ultérieure. Cette étape peut aider à éliminer les variables redondantes et à réduire la complexité de l'analyse.

4. **Gestion des données manquantes** :La gestion des données manquantes est une étape importante du prétraitement des données, car les données manquantes peuvent avoir un impact significatif sur les résultats de l'analyse. Les techniques courantes de gestion des données manquantes comprennent :

- **L'imputation de données** : cette technique consiste à remplacer les données manquantes par des valeurs estimées à partir des autres données disponibles.
- **La suppression de données** : cette technique consiste à éliminer les lignes ou les colonnes qui contiennent des données manquantes

5. **Traitement des données textuelles :** Le traitement des données textuelles est une étape importante du prétraitement des données pour les données non structurées telles que les textes. Les techniques courantes de traitement des données textuelles comprennent :

- **La normalisation des textes** : cette technique consiste à réduire tous les textes à une forme commune, telle que la suppression des majuscules, de la ponctuation et des espaces.
- **La suppression des mots vides** : cette technique consiste à éliminer les mots couramment utilisés tels que "le", "la", "et", qui n'ont pas de signification particulière dans le contexte de l'analyse

Pré-traitement des données

Techniques de pré-traitement des données

- **L'analyse de la fréquence des mots** : cette technique consiste à identifier les mots les plus fréquemment utilisés dans le texte et à leur attribuer des poids en fonction de leur fréquence.
- **Tokenisation** : Elle consiste à segmenter un document texte en tokens.Un token est défini comme une séquence de caractères comprise entre deux séparateurs (les blancs, les signes de ponctuation et certains autres caractères comme les guillemets ou les parenthèses)
- La **racinisation (Stemming)** : Elle considère uniquement la racine des mots plutôt que les mots en entier sans se soucier de son analyse grammaticale ;
- **Lemmatisation** : elle ramène les termes à leur forme canonique en mettant tous les noms au singulier, les adjectifs au masculin singulier, et tous les verbes conjugués à l'infinitif.

Pré-traitement des données

Bibliothèques python pour pré-traitement des données

1. **Pandas :**
   - Charger des données à partir de fichiers (csv, Excel, etc.) ou de bases de données ;
   - Nettoyer les données en éliminant les doublons, les données manquantes, etc. ;
   - Transformer les données en ajoutant, supprimant ou modifiant des colonnes ;
   - Regrouper les données pour réaliser des analyses
2. **NumPy :**
   - Stocker et manipuler les données ;
   - Effectuer des opérations mathématiques et statistiques sur les données ;
   - Générer des nombres aléatoires.

Pré-traitement des données

Bibliothèques python pour pré-traitement des données

1. **Scikit-learn :**
   - La normalisation des données ;
   - La gestion des données manquantes ;
   - La réduction de la dimensionnalité ;
   - La sélection des attributs ;
   - La discrétisation des données.

2. **NLTK :**
   - Nettoyer les données textuelles ;
   - Normaliser les textes en minuscules et en retirant la ponctuation ;
   - Supprimer les mots vides ;
   - Analyser la fréquence des mots.

Pré-traitement des données

Etapes de pré-traitement des données

1. Charger les données à partir d'une source externe (fichier, base de données, etc.) en utilisant Pandas.
2. Effectuer une exploration de données pour comprendre la nature des données et identifier les erreurs et les valeurs aberrantes.
3. Nettoyer les données en éliminant les doublons, les données manquantes et les valeurs aberrantes.
4. Transformer les données en ajoutant, supprimant ou modifiant des colonnes.
5. Réduire la dimensionnalité des données en sélectionnant les attributs les plus importants.
6. Normaliser les données pour éviter les biais dus aux différences de mesure.
7. Diviser les données en ensembles d'entraînement et de test pour l'analyse ultérieure.

CHAPITRE 3

Les règles d'association

Introduction

- Les **règles d'association** sont une méthode d'analyse de données qui permettent de découvrir des relations entre des variables dans une grande quantité de données.
- Elles sont utilisées pour trouver des relations entre des articles vendus ensemble dans un magasin, des symptômes liés à une maladie, des mots fréquemment utilisés ensemble dans un texte, etc.
- Dans ce cours, nous allons examiner les concepts de base des règles d'association , présenter l'algorithme Apriori et appliquer les règles d'association en Python à l'aide de la bibliothèque mlxtend.

Introduction

Définitions

1. Une **règle d'association** est une application de la forme X$\rightarrow$ Y où X et Y sont des ensembles d'items disjoints.
   Dans notre cas , une règle peut s'écrire :

   SI Produit1 **ALORS** Produit2
2. **item** : un élément d'un ensemble (un produit)
3. **itemset** : ensemble de produits (par exemple :{Pain, Lait})
4. **sup(itemset)** : nombre de transactions d'apparition simultanée des produits
5. **card(itemset)** : nombre de produits dans l'ensemble

Les transactions

- Une **transaction** représente un ensemble d'articles ou objets dans une base des données
- Nous pouvons représenter les transactions comme :
  1. Liste
  2. Représentation verticale
  3. Représentation horizontale

Les transactions

Une Liste

- Chaque ligne représente une transaction
- Chaque ligne liste les items achetés par le consommateur
- Les lignes peuvent avoir un numéro différent de colonnes

Représentation verticale :seulement deux colonnes

- une colonne pour les numéros de la transaction (id)
- Une colonne indiquant un item présent

Représentation horizontale : Les transactions se représentent avec une matrice binaire

- Chaque ligne de la matrice représente une transaction
- Chaque colonne représente un article ou item
- Si un item est présent dans une transaction sera représenté avec un 1
- Si un item est absent sera représenté avec un 0

Les transactions

Exemple

Le tableau ci-dessous contient un ensemble de données relatives aux transactions de vente dans un magasin.

Transactions	Articles
1	{ Yaourt, Tomate, Banane}
2	{ Orange, Pomme }
3	{ Yaourt, Tomate, Orange, Pomme }
4	{ Tomate, Pomme, Banane }
5	{ Yaourt, Tomate,Orange,Banane }

1. Sous quelle forme ces transactions ont été représentées
2. Donner une représentation binaire de ce tableau.

Les transactions

Solution exemple

1. Il s'agit d'une représentation verticale
2. La représentation binaire de ce tableau (représentation horizontale) est :

TID	Yaourt	Tomate	Orange	Pomme	Banane
1	1	1	0	0	1
2	0	0	1	1	0
3	1	1	1	1	0
4	0	1	0	1	1
5	1	1	1	0	1

Concepts de base des règles d'association

Les règles d'association sont basées sur les mesures suivantes : la **fréquence** , le **support** et la **confiance** .

1. La **fréquence** est le nombre de fois qu'un élément ou une combinaison d'éléments apparaît dans un ensemble de données.
2. Le **support** d'un ensemble d'items fait référence au nombre de transactions (observées) qui le contient. Mathématiquement, le support $\sigma(X)$ d'un ensemble d'items X est défini par : $\sigma(X) = Card(\{t_i | X \subset t_i, t_i \in T\})$ où $Card(A)$ représente le cardinal de l'ensemble A.
3. La **confiance** est la probabilité que deux articles soient vendus ensemble. Elle est définie comme le nombre de fois qu'une combinaison d'articles a été vendue ensemble, divisé par le nombre de fois que le premier article apparaît dans l'ensemble de données.

Concepts de base des règles d'association

1. Calcul du support d'une règle d'association :

$$\sigma(X \rightarrow Y) = \frac{\sigma(X \bigcup Y)}{N}$$

N est le nombre de transactions.

2. Calcul de la confiance d'une règle d'association :

$$c(X \rightarrow Y) = \frac{\sigma(X \bigcup Y)}{\sigma(X)}$$

Concepts de base des règles d'association

Remarques

- le support est un indicateur de « fiabilité » d'une règle
- La confiance est un indicateur de « précision » d'une règle
- Il existe un autre métrique permettant d'avaluer une règle d'association. Il s'agit de **LIFT** qui est un indicateur de pertinence des règles.

$$Lift(X \rightarrow Y) = \frac{\sigma(X \rightarrow Y)}{\sigma(X) * \sigma(Y)}$$

Concepts de base des règles d'association

Exercice

Supposons que nous avons un ensemble de données contenant des transactions de vente dans un magasin.

Transactions	Articles
1	{ Pain, Lait, Fromage}
2	{ Pain, Lait, Beurre }
3	{ Pain, Lait, Beurre, Fromage }
4	{ Pain, Lait, Beurre }
5	{ Pain, Lait, Fromage }

1. Combien d'items possibles peut-on former à partir de ce tableau des transactions ?
2. Que représente un item ? Donner un exemple.
3. Considérons la règle suivante : {Pain, Lait}→ {Fromage}. Déterminer son support et sa confiance et son Lift.

Algorithme Apriori

Présentation

- L'algorithme APriori est un algorithme d'exploration de données conçu en 1994, par Rakesh Agrawal et Ramakrishnan Sikrant, dans le domaine de l'apprentissage des règles d'association.
- Il sert à reconnaitre des propriétés qui reviennent fréquemment dans un ensemble de données et d'en déduire une catégorisation.

Apriori :Principe

L'algorithme Apriori s'execute en deux étapes :

1. Génération de tous les itemsets fréquents c'est-à-dire :

$$\text{IF} = \{ X_i \subseteq T | \text{ supp } (X_i) = X_i.count \geqslant \text{minsupp}, \quad \text{i=1,2..,n } \}$$

2. Génération de toutes les règles d'associations de confiance à partir des itemsets fréquents, c'est-à-dire

$$\{ X_i, Y_j \subseteq IF | X_i \cap Y_j = \varnothing \wedge \text{Conf}(X_i \rightarrow Y_j) \geqslant \text{minconf} \quad \text{i=1,2,...,p j=1,2,...,q } \}$$

minsupp est l'indice de support minimum donné, et **minconf** l'indice de confiance donné.

Apriori

Le Pseudo-code de l'algorithme Apriori est :

Algorithme 4 Apriori

Entrée(s): Base de données de transactions D, Seuil de support minimum σ ;

Sorties(s): Ensemble des items fréquents

1: i←1
2: $C_1 \leftarrow$ ensemble des motifs de taille 1 (un seul item)
3: **Tant que** $C_i \neq \emptyset$ **faire**
4: Calculer le Support de chaque motif m $\in C_i$ dans la base
5: $F_i \leftarrow \{m \in C_i | \text{ support}(m) \geqslant \sigma \}$
6: $C_{i+1} \leftarrow$ toutes les combinaisons possibles des motifs de F_i de taille i + 1
7: i← i + 1
8: **Fin Tant que**
9: **Retourner** $\bigcup_{(i \geqslant 1)} F_i$

Apriori :Exemple

Le tableau ci-dessous représente le contenu du panier d'une ménagère.

TID	Items
1	{ Pain, Lait}
2	{ Pain, Couches,Bière, Oeufs }
3	{ Lait, Couches, Bière, Djino }
4	{ Pain, Lait, Couches, Bière }
5	{ Pain, Lait, Couches, Djino }

Pour découvrir les relations cachées dans cette base de données, nous allons effectuer l'analyse des associations.

Apriori :Exemple

La représentation binaire des données du tableau précédent est donnée ci-dessous :

TID	Pain	Lait	Couches	Bière	Oeufs	Djino
1	1	1	0	0	0	0
2	1	0	1	1	1	0
3	0	1	1	1	0	1
4	1	1	1	1	0	0
5	1	1	1	0	0	1

Apriori :Exemple

Par exemple pour la règle {Lait, Couches} $\rightarrow$ {Bière}. Le support de l'ensemble {Bière, Couches, Lait} étant 2 et le nombre total de transactions étant 5, le support de la règle est donc : $\frac{2}{5}$=0,4

Sa confiance = $\frac{support\{Bière, Couches, Lait\}}{support\{Lait, Couches\}} = \frac{2}{3} = 0{,}67$ car on a 3 transactions contenant {Lait, Couches}.

■ **Recherche de règles d'associations en utilisant l'algorithme Apriori**

Fixons un degré d'exigence sur les règles à extraire. Par exemple :
Support min : 2 transactions et **Confiance min** =75%
L'idée est surtout de contrôler (limiter) le nombre de règles produites.
Nous allons procéder ici en deux étapes :

Apriori :Exemple

Étape 1 : Génération des ensembles d'items fréquents (support $\geq$ support min.)

Pour le faire, nous allons utiliser le graphe pour énumérer tous les ensembles d'items possibles.

Sur ce graphe ci-dessous :

P= Pain

L = Lait

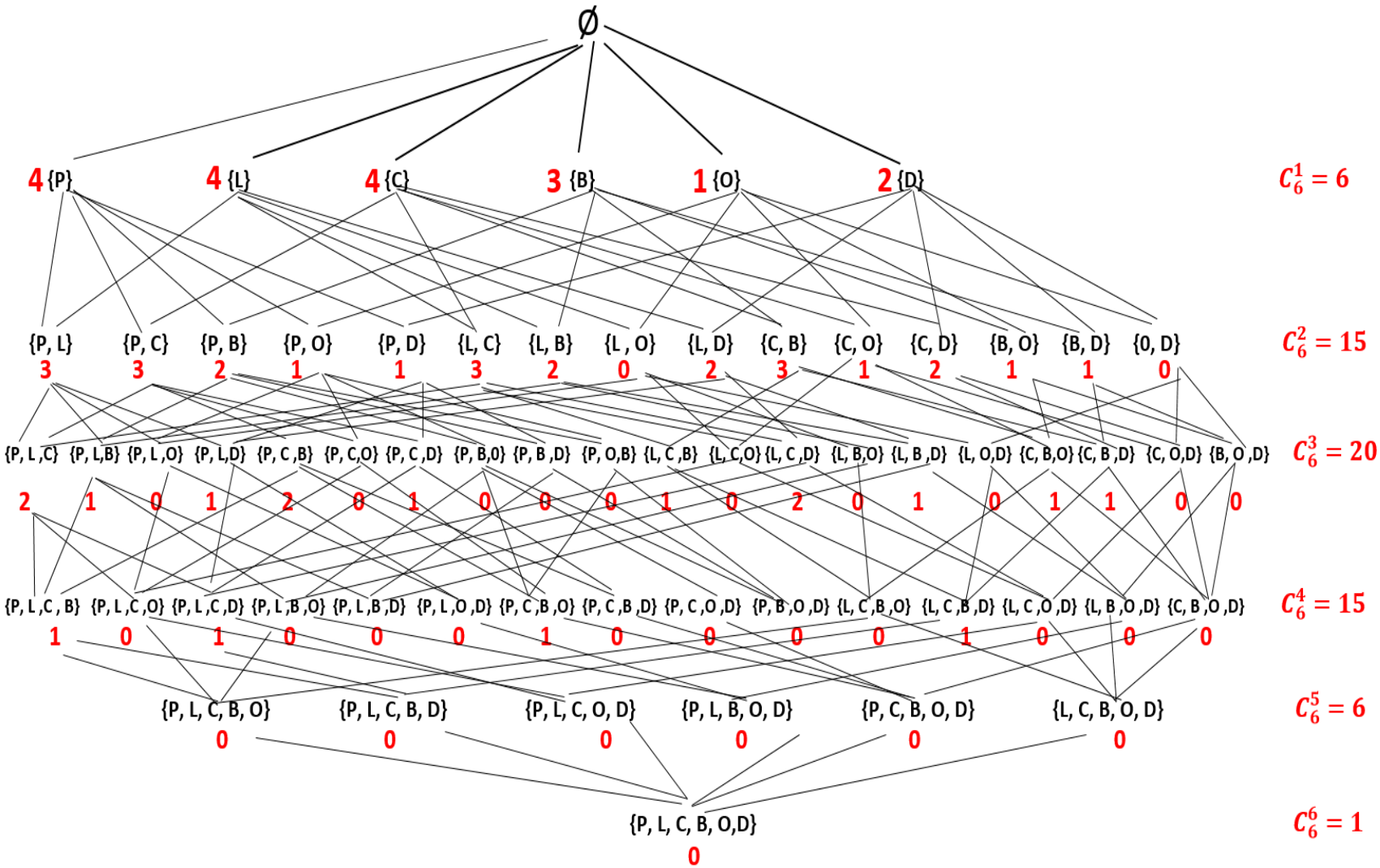
C =Couches

B = Biere

O = Oeufs

D = Djino

Apriori :Exemple



Apriori :Exemple

Le support minimal étant de 2, nous retenons donc de ce graphe, tous les itemsets fréquents c'est-à-dire les itemsets donc le support est supérieur ou égal à 2.
Le tableau ci-dessous, donne la liste de ces itemsets fréquents :

Item	{P}	{L}	{C}	{B}	{D}	{P,L}	{P,C}	{P,B}	{L,C}	{L,B}	{L,D}	{C,B}	{C,D}	{P,L,C}	{P,C,B}	{L,C,D}
Sup	4	4	4	3	2	3	3	2	3	2	2	3	2	2	2	2

Apriori :Exemple

Étape 2 : Génération des règles à grande confiance à partir des items fréquents précédents trouvés précédemment : Règles fortes (conf. $\geq$ conf. min.)

Il s'agit ici de déduire les règles à partir des itemsets fréquents et limiter par la suite la prolifération des règles en utilisant le critère **confiance min**=0,75.

Nous utiliserons par conséquent les itemsets fréquents avec au moins deux éléments et nous allons tester toutes les combinaisons possibles :

Itemset	{P,L}	{P,C}	{P,B}	{L,C}	{L,B}	{L,D}	{C,B}	{C,D}	{P,L,C}	{P,C,B}	{L,C,D}
Support	3	3	2	3	2	2	3	2	2	2	2

Apriori :Exemple

Calculons donc la confiance de chaque règle :

{ P, L }	P ⟶ L : conf = 3 / 4 = 75% **(accepté)**
	L ⟶ P : conf = 3 / 4 = 75% **(accepté)**
{ P, C }	P ⟶ C : conf = 3 / 4 = 75% **(accepté)**
	C ⟶ P : conf = 3 / 4 = 75% **(accepté)**
{ P, B }	P ⟶ B : conf = 2 / 4 = 50% **(refusé)**
	B ⟶ P : conf = 2 / 3 = 66,6% **(refusé)**
{ L, C }	L ⟶ C : conf = 3 / 4 = 75% **(accepté)**
	C ⟶ L : conf = 3 / 4 = 75% **(accepté)**

Apriori :Exemple

{ L, B }	L ⟶ B : conf = 2 / 4 = 50% **(refusé)**
	B ⟶ L : conf = 2 / 3 = 66,6% **(refusé)**
{ L, D }	L ⟶ D : conf = 2 / 4 = 50% **(refusé)**
	D ⟶ L : conf = 2 / 2 = 100% **(accepté)**
{ C, B }	C ⟶ B : conf = 3 / 4 = 75% **(accepté)**
	B ⟶ C : conf = 3 / 3 = 100% **(accepté)**
{ C, D }	C ⟶ D : conf = 1 / 4 = 25% **(refusé)**
	D ⟶ C : conf = 1 / 2 = 50% **(refusé)**

Apriori :Exemple

{ P,L, C }	P ⟶ {L,C} : conf = 2 / 4 = 50% **(refusé)**
	{L,C} ⟶ P : conf = 2 / 3 = 66,6% **(refusé)**
{ P,C, B }	P ⟶ {C,B} : conf = 2 / 4 = 50% **(refusé)**
	{C,B} ⟶ P : conf = 2 / 3 = 66,6% **(refusé)**
{ L,C, D }	L ⟶{C,D} : conf = 2 / 4 = 50% **(refusé)**
	{C,D} ⟶ L : conf = 2 / 2 = 100% **(accepté)**

Toutes les règles ayant une confiance supérieure ou égale à 75% sont acceptées.

Apriori :Limites

Malgré sa simplicité, l'algorithme Apriori présente quelques limites :

- Il n'est pas envisageable de chercher toutes les règles d'associations pour ensuite sélectionner celles qui ont un support et une confiance suffisants.Cela est très couteux à gérer un grand nombre d'ensembles candidates. Par exemple pour un ensemble de d items, le nombre total de règles possibles est de $R = 3^d - 2^{d+1} + 1$. Si d =6, on R=602.
- Il es fastidieux de de numériser plusieurs fois sur la base des données et vérifier un grand nombre de candidats par correspondance de motif,ce qui est particulièrement vrai pour l'exploitation des longs motifs.

Afin de surmonter les inconvénients rencontrés dans l'algorithme **Apriori**, l'on utilisera l'algorithme **FP-Growth**.

FP-Growth

Présentation

FP-growth (frequent pattern growth) est une version améliorée de l'algorithme Apriori qui est largement utilisé pour le minage de modèles fréquent. Il est utilisé comme un processus analytique qui trouve des modèles ou des associations fréquents à partir d'ensembles de données.

L'algorithme FP-growth consiste en deux étapes :

1. Utilise une structure d'arbre (FP-tree) pour stocker une forme compressée d'une base de données.
2. Adopte une stratégie de découpage pour décomposer les tâches d'exploration de données et les bases de données (**diviser pour régner**)et utilise une méthode pour éviter le coûteux processus de génération et de test des candidats, utilisé par Apriori.

FP-Growth : Pseudo-code

Ci-dessous le Pseudo-code de l'algorithme FP-Growth

Algorithme 5 FP-Growth

Entrée(s): Base de données de transactions D

Sorties(s): modèles fréquents

1: Déduire les articles fréquents commandés. Pour les éléments de même fréquence, l'ordre est donné par ordre alphabétique.
2: Construire l'arbre FP à partir des données ci-dessus
3: À partir de l'arbre FP ci-dessus, construire l'arbre conditionnel FP pour chaque élément (ou ensemble d'éléments).
4: Déterminer les modèles fréquents.

Exercice

Exercice

Vous disposez d'un ensemble de données représentant les transactions des clients dans un magasin de vêtements. Chaque transaction se compose d'articles achetés par un client.
Transaction 1 : {Chemise, Pantalon, Cravate}
Transaction 2 : {Chemise, Veste, Chaussures}
Transaction 3 : {Pantalon, Chaussures, Chapeau}
Transaction 4 : {Chemise, Pantalon, Chaussures}
Transaction 5 : {Veste, Chapeau}

1. Donner une représentation binaire de ces données
2. Utilisez l'algorithme Apriori pour trouver tous les ensembles fréquents avec un support minimum de 2 transactions.
3. En utilisant les ensembles fréquents trouvés à l'étape précédente, générez toutes les règles d'association possibles avec une confiance minimale de 50%.

Exercice

Exercice

4. Calculez le support et la confiance pour chaque règle d'association générée.
5. Identifiez les règles d'association intéressantes qui satisfont à la fois le support minimum et la confiance minimale.
6. Interprétez les règles d'association intéressantes trouvées en termes de comportement d'achat des clients dans le supermarché.

CHAPITRE 4

La Classification supervisée

Classification supervisée

Introduction

- **Classer un document** c'est tout simplement mettre une étiquette sur son contenu , lui faisant ainsi appartenir à un groupe ou classe bien définie.
- La classification (ou catégorisation) d'un document est l'une des tâches de traitement du langage naturel (NLP) les plus courantes.
- Elle joue un rôle essentiel dans de nombreuses tâches de gestion et de récupération de l'information.

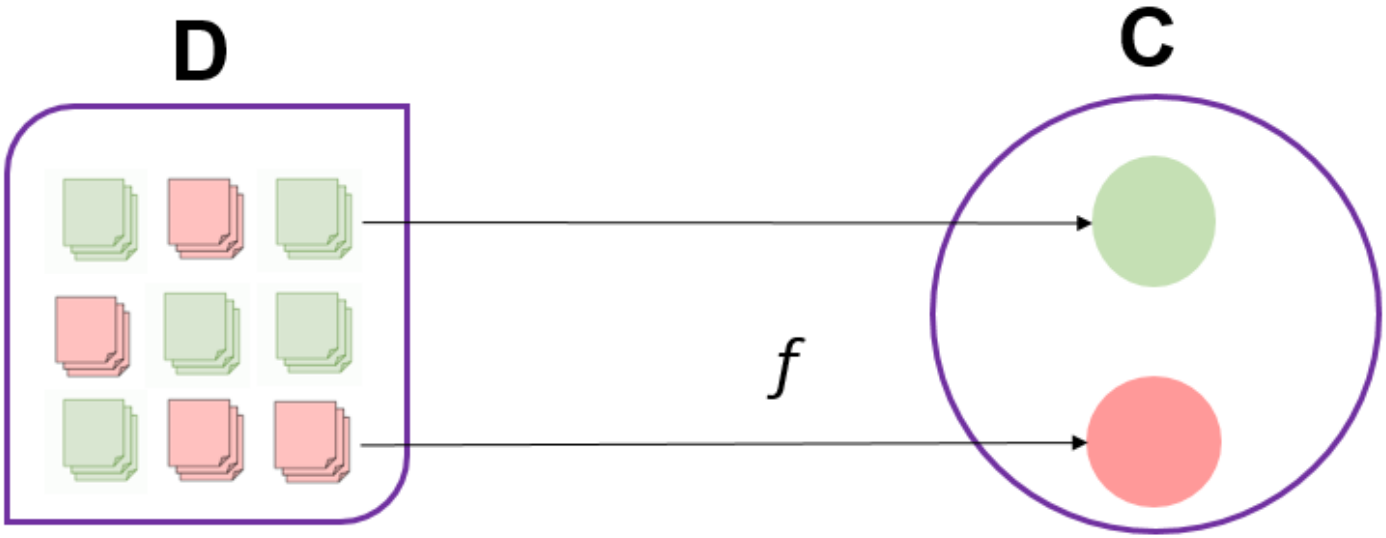
Définition

Un classificateur d'un document est une fonction booléenne (f) qui associe automatiquement un document (D) à la classe (C) :

$$f : D \rightarrow C$$

Classification supervisée

Définition



$f(\square, \bigcirc) = 0$

$f(\square, \bigcirc) = 1$

Où

0 = faux

1= vrai

Formes de classification

- **Classification Binaire :** Il s'agit tout simplement d'une classification à 2 classes comme l'illustre la figure 1.1. Par exemple, un système de détection de SPAM classifie un e-mail comme étant soit un "SPAM", soit un "NON-SPAM".
- **Classification Multi-Classe :** Elle consiste à associer un document à une classe parmi plusieurs.
- **Classification Multi-Label :** Elle consiste à associer le texte en entrée à une ou plusieurs classes.
- **Classification en Cascade :** Il s'agit d'un classifieur composé de plusieurs classifieurs mis l'un à la suite de l'autre. Le but étant de classifier suivant des sous-classes.

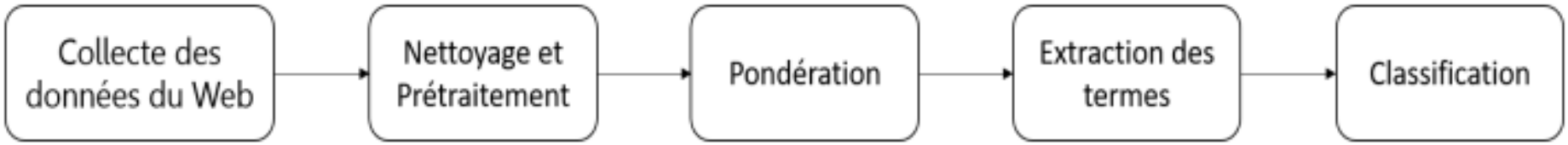
Types de classification

La classification automatique d'un texte peut se faire de deux façons :

1. **La classification non supervisée (clustering)** : Elle consiste à apprendre à classer sans supervision. Au début du processus on ne dispose ni de la définition des classes, ni du nombre. C'est l'algorithme de classification qui va déterminer ces informations.
2. **La classification supervisée (catégorisation)** : Contrairement à l'apprentissage non supervisé, on commence ici par un ensemble de classes connues et définies à l'avance. on dispose aussi d'une sélection initiale de données dont la classification est connue. Ces données sont supposées indépendantes et identiquement distribuées. Elles nous servent pour l'apprentissage de l'algorithme. L 'algorithme réalise donc la classification selon le modèle qu'il a appris.

Processus de classification

Pour déterminer la classe d'un document, un ensemble étapes est habituellement suivies.



Pondérations ou calcul des fréquences

Frequence d'un terme

Elle désigne le nombre de fois qu'un certain descripteur est apparu dans chacun des documents d'un corpus.
À partir de ces fréquences, que nous nommons communément "poids", nous pouvons dire qu'un descripteur quelconque est discriminant ou pas, par rapport à un document donné, si son poids est élevé ou pas, respectivement

Il existe plusieurs manières de définir un ensemble de descripteurs :

- Le sac à mots (bag of words)
- Les N-grams
- Fréquence des termes (TF)
- Fréquence documents inverses (IDF)
- TFIDF

Pondérations ou calcul des fréquences

Le sac à mots (bag of words)

- La façon la plus simple et la plus évidente pour la représentation d'un document texte par un document vecteur, est d'utiliser les mots comme descripteurs
- Cette technique conserve le sens naturel des descripteurs.
- Les mots sont regroupés en vrac et traités d'une manière indépendante, ce qui nuit considérablement à la sémantique du texte.

Pondérations ou calcul des fréquences

Le sac à mots (bag of words)

Le gouvernement Camerounais s'est engagé dans un programme de développement de l'enseignement des TIC à travers la généralisation de l'enseignement de l'informatique à tous les niveaux. La mise en œuvre de cette détermination dans le secteur de l'éducation s'exprime à travers des actes administratifs et réglementaires dont le plus important est : L'arrêté N° 3745/D/63/ MINEDUC/CAB du 17/06/2003 portant introduction de l'Informatique dans les programmes de formation des 1er et 2nd Cycles de l'enseignement secondaire général et des ENIEG, rappelons que ces cours étaient déjà dispensés en enseignement technique (EST) et l'entrée en vigueur des programmes d'enseignement dès l'année scolaire 2003/2004.

Mot	Occurr.	Mot	Occurr.	Mot	Occurr.
à	2	en	3	MINEDUC	1
actes	1	engagé	1	mise	1
administratifs	1	ENIEG	1	niveaux	1
arrêté	1	enseignement	5	œuvre	1
CAB	1	entrée	1	plus	1
camerounais	1	est	2	portant	1
cette	1	et	4	programme	3
cycles	1	étaient	1	rappelons	1
dans	3	exprime	1	réglementaires	1
de	8	formation	1	scolaire	1
déjà	1	général	1	secondaire	1
des	5	généralisation	1	secteur	1
dès	1	gouvernement	1	technique	1
détermination	1	important	1	TIC	1
développement	1	informatique	2	tous	1
dispensés	1	introduction	1	travers	1
dont	1	la	2	un	1
du	1	le	3	vigueur	1
éducation	1	les	2		

Pondérations ou calcul des fréquences

Les N-grams

N-grams est une méthode de représentation de documents texte qui consiste à partager ce dernier en séquences de *n* caractères.
Prenons l'exemple de la phrase "**La classification supervisée** " et essayons de la représenter en ngrams caractères.

- si n=2 nous aurons : "La", "a ", " c", "cl", "la", "as", "ss", "si", "if", etc.
- si n=3 nous aurons : "La ","a c", " cl", "cla", "las", "ass", "ssi", "sif", "ifi", etc.
- si n=4 nous aurons : "Lac ", "a cl", " cla", "clas", "lass", "assi", "ssif", "sifi", "ific", etc

Pondérations ou calcul des fréquences

Fréquence des termes (TF)

Nous dénombrons plusieurs manières de calcul de la TF :

- **TF absolue** : c'est le nombre de fois qu'un terme apparait dans un texte donné.

$$\mathbf{TF = NT}$$

(où NT est le nombre de fois où le terme est apparu dans le texte.)

- **TF relative** : C'est le rapport entre le nombre de fois qu'un terme est apparu dans le texte sur le nombre de tous les termes du texte.

$$\mathbf{TF = \frac{NT}{ST}}$$

- **TF booléenne** : se contente juste de la présence ou de l'absence du terme dans le texte.

$$\mathbf{TF = 0 \text{ ou } 1}$$

Pondérations ou calcul des fréquences

Fréquence des termes (TF)

- **Fréquence documents inverses (IDF)** : Elle mesure en quelque sorte le degré de rareté d'un terme, non pas dans un document, mais dans tous les documents d'un corpus. Elle est définie par cette équation :

$$\mathbf{IDF} = \mathbf{log}\left(\frac{\mathrm{N_{doc}}}{\mathrm{Doc_T}}\right)$$

Où N_{doc} est le nombre de documents dans le corpus, et Doc_T est le nombre de documents dans lesquels le terme est apparu. Si le terme est très présent dans tout le corpus alors le rapport sera égal à 1 et $\mathbf{IDF = 0}$ donc le terme est neutralisé. Si par contre il apparait dans un seul document la valeur est maximale.

$$\mathbf{IDF} = \mathbf{log}(\mathbf{N_{doc}})$$

Pondérations ou calcul des fréquences

Fréquence des termes (TF)

- **TFIDF** : Elle est une combinaison de l'abondance particulière et de la rareté générale d'un terme dans un corpus.Elle est calculée avec la formule suivante :

$$\mathbf{TFIDF = TF * log}\left(\frac{\mathbf{N_{doc}}}{\mathbf{Doc_T}}\right)$$

où **TF** est relative ou absolue.

Étiquetage du document

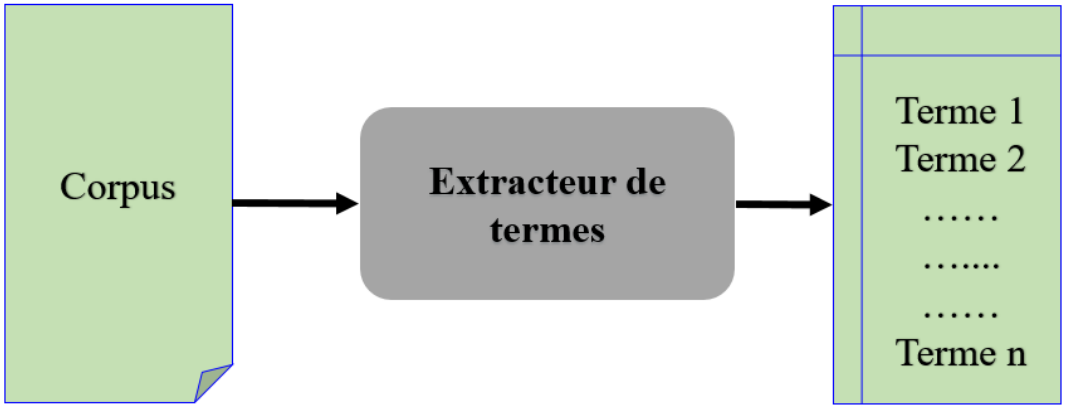
- L'étiqueteur réalise l'analyse morpho-syntaxique qui est une étape qui peut être considérée comme préliminaire à tout traitement linguistique plus poussé sur un texte, notamment l'analyse syntaxique.
- Elle consiste à affecter des étiquettes morpho-syntaxiques propres à chaque mot d'une phrase d'un texte (catégorie grammaticale, informations morphologiques comme le genre, le nombre...).

Par exemple l'étiquetage correct de la phrase "**Isaac a mangé une pomme**" est :

Isaac	a	mangé	une	pomme
Nom	verbe	verbe	déterminant	Nom

Extraction des termes

L'extraction des termes est une étape du processus de classification automatique qui consiste à analyser un corpus et proposer à l'utilisateur les termes qui s'y trouvent.



L'identification d'un terme repose sur le lien qu'on peut établir entre son sens et un domaine de spécialité, donc sur des connaissances extra-linguistiques. Par exemple, après avoir soumis un texte d'informatique à un extracteur de termes, on pourra avoir des termes comme : ordinateur, logiciel, programme, page web, internet, site web, navigateur, tablette, écran, souris,...

Présentation

L'Algorithme ID3 a été développé à l'origine par Ross Quinlan. Il a tout d'abord été publié dans le livre "Machine Learning" en 1986. C'est un algorithme de classification supervisé, c'est-à-dire qu'il se base sur des exemples déjà classés dans un ensemble de classes pour déterminer un modèle de classification. Le modèle que produit ID3 est un arbre de décision. Cet arbre servira à classer de nouveaux échantillons.

Principe

Chaque exemple en entrée pour cet algorithme est constitué d'une liste d'attributs. Un de ces attributs est l'attribut « cible » et les autres sont les attributs « non cibles ». On appelle aussi cette "cible" la "classe". En fait l'arbre de décision va permettre de prédire la valeur de l'attribut « cible » à partir des autres valeurs. Bien entendu, la qualité de la prédiction dépend des exemples : plus ils sont variés et nombreux, plus la classification de nouveaux cas sera fiable.
ID3 construit l'arbre de décision récursivement. À chaque étape de la récursion, il calcule parmi les attributs restant pour la branche en cours, celui qui maximisera le gain d'information. C'est-à-dire l'attribut qui permettra le plus facilement de classer les exemples à ce niveau de cette branche de l'arbre. On appelle ce calcul **l'entropie de Shannon** dont voici la formule utilisée :

$$\textbf{Entropie(S)} = -\sum_{c \in \text{classes(S)}} P_c \times log_2(P_c)$$

- S est le set d'exemples.
- La quantité minimale de données redondantes à ajouter pour qu'un message ayant une probabilité p d'arriver sans être corrompu est $-log_2(P)$
- P_c est la proportion d'exemples de S ayant pour classe résultante c.

En bref, voila comment fonctionne l'algorithme ID3 :

1. Calculer l'entropie de tous les attributs en utilisant l'ensemble d'apprentissage S
2. Partitionner l'ensemble S en utilisant l'attribut pour lequel l'entropie est minimum (gain d'information maximum)
3. Construire le noeud de l'arbre avec cet attribut
4. Recommencer récursivement sur chaque sous arbre avec chaque sous-ensemble.

Le pseudo-code de l'algorithme ID3 est donné ci-dessous :

Algorithmes de classification supervisée

ID3 : Pseudo-code

Algorithme 1 ID3

Entrée(s): exemples(liste d'exemples étiquetés), questions(liste des attributs non utilisés jusqu'à présent);

Sorties(s): un nœud

```
Si exemples est vide alors
    Finir la fonction sans construire de nœud
Fin Si
Si tous les exemples sont la même classe alors
    Retourner une feuille ayant cette classe
Fin Si
Si questions est vide alors
    Retourner une feuille avec la classe la plus fréquente
Fin Si
q = attribut optimal (avec le plus grand gain d'entropie)
n = nouveau nœud créé qui testera l'attribut q
Pour chaque v = valeur possible de q faire
    e = l'ensemble des éléments de exemples ayant v comme valeur à l'attribut q
    chaque nœud fils de n est créé par ID3(e, questions\{q})
Fin pour
Retourner n
```

Algorithmes de classification supervisée

ID3 : Exemple

Considérons les données du tableau ci-dessous :

Jour	**Attributs des exemples**				**Classe**
Jour	Prévisions	Température	Humidité	Vent	Jouer
1	Ensoleillé	Chaud	Élevée	Faible	Non
2	Ensoleillé	Chaud	Élevée	Fort	Non
3	Nuageux	Chaud	Élevée	Faible	Oui
4	Pluvieux	Moyen	Élevée	Faible	Oui
5	Pluvieux	Frais	Normale	Faible	Oui
6	Pluvieux	Frais	Normale	Fort	Non
7	Nuageux	Frais	Normale	Fort	Oui
8	Ensoleillé	Moyen	Élevée	Faible	Non
9	Ensoleillé	Frais	Normale	Faible	Oui
10	Pluvieux	Moyen	Normale	Faible	Oui
11	Ensoleillé	Moyen	Normale	Fort	Oui
12	Nuageux	Moyen	Élevée	Fort	Oui
13	Nuageux	Chaud	Normale	Faible	Oui
14	Pluvieux	Moyen	Élevée	Fort	Non

Prévisions, **Température**, **Humidité** et **Vent** sont les quatre attributs qui déterminent chacun des exemples qui vont être fournis. On peut voir qu'il existe uniquement 36 exemples différents pour cette configuration d'attributs :

$$\{\text{Ensoleillé,Nuageux,Pluvieux}\} \times \{\text{Chaud,Moyen,Frais}\} \times \{\text{Élevée,Normale}\} \times \{\text{Faible,Fort}\} = 3 \times 3 \times 2 \times 2 = 9 \times 4 = 36$$

Dans cet exemple, les seules classes possibles sont **Oui** et **Non**. Autrement dit la classification de la cible est "**devrions-nous jouer au tennis ?**" qui peut être **oui** ou **non**.
Donc l'entropie vaut :

$$-P_{Oui} \times log_2(P_{Oui}) - P_{Non} \times log_2(P_{Non})$$

Initialement, l'algorithme prend tout le set S=$\{J_1, J_2, J_3, ..., J_{14}\}$. Et comme 9 des 14 exemples donnent la réponse (ou classe) **Oui** et 5 sur 14 donnent la réponse (ou classe) **Non**,

$$P_{Oui} = \frac{9}{14}$$
$$P_{Non} = \frac{5}{14}$$

On peut donc calculer que :

$$Entropie(S) = -(\frac{9}{14}) \times log_2(\frac{9}{14}) - (\frac{5}{14}) \times log_2(\frac{9}{14}) = 0.94$$

Calculons le gain d'entropie pour chacun des attributs :
Rappelons que ce gain est donnée par la formule :

$$\text{Gain(S,A)} = \text{Entropie(S)} - \sum_{v \in valeurs(A)} \frac{|S_v|}{|S|} \times \text{Entropie}(S_v)$$

Prévisions

L'attribut **Prévisions** a trois valeurs possibles : Ensoleillé,Nuageux et Pluvieux

Gain(S,Prévisions) = Entropie(S) - $\frac{5}{14}\times$ Entropie($S_{Ensoleillé}$) - $\frac{4}{14}\times$ Entropie($S_{Nuageux}$) - $\frac{5}{14}\times$ Entropie($S_{Pluvieux}$)

$$\begin{aligned} &= 0{,}94 - \tfrac{5}{14} \times \left(-\tfrac{3}{5} log_2\left(\tfrac{3}{5}\right) - \tfrac{2}{5} log_2\left(\tfrac{2}{5}\right)\right) - \tfrac{4}{14} \times \\ &\left(-\tfrac{0}{5} log_2\left(\tfrac{0}{4}\right) - \tfrac{4}{4} log_2\left(\tfrac{4}{4}\right)\right) - \tfrac{5}{14} \times \left(-\tfrac{3}{5} log_2\left(\tfrac{3}{5}\right) - \tfrac{2}{5} log_2\left(\tfrac{2}{5}\right)\right) \\ &= 0{,}94 - 0{,}357 \times (0{,}97) \\ &= 0{,}24742 \end{aligned}$$

Donc **Gain(S,Prévisions) = 0,24742**

En procédant de la même manière avec les calculs de gain d'attributs des autres variables, nous obtenons :

Gain(S,Vent) = Entropie(S) - $\frac{8}{14}\times$ **Entropie(**S_{Faible}**) -** $\frac{6}{14}\times$ **Entropie(**S_{Fort}**) = 0.048**

Gain(S,Humidité) = Entropie(S) - $\frac{8}{14}\times$ **Entropie(**$S_{\text{Élevée}}$**) -** $\frac{6}{14}\times$ **Entropie(**$S_{Normale}$**)=0.153**

Gain(S,Température) = Entropie(S) - $\frac{4}{14}\times$ **Entropie(**S_{Chaud}**) -** $\frac{6}{14}\times$ **Entropie(**S_{Moyen}**) -** $\frac{4}{14}\times$ **Entropie(**S_{Frais}**) = 0.028**

Récapitulons :

Gain(S,Température) $<$ Gain(S,Vent) $<$ Gain(S,Humidité)$<$ Gain(S,Prévisions).
On peut voir que le plus grand gain est pour **Prévisions**. C'est donc Prévisions qui est le premier attribut testé dans l'arbre c'est à dire il est l'attribut de décision dans le nœud racine de notre arbre.

Nous obtenons donc l'arbre temporaire ci-dessous :

Algorithmes de classification supervisée

ID3 : Exemple

Ensoleillé

Jour	Température	Humidité	Vent	jouer
1	Chaud	Elevée	Faible	Non
2	Chaud	Elevée	Fort	Non
8	Moyen	Elevée	Faible	Non
9	Frais	Normale	Faible	Oui
11	Moyen	Normale	Fort	Oui

?

Nuageux

Jour	Température	Humidité	Vent	jouer
3	Chaud	Elevée	Faible	Oui
7	Frais	Normale	Fort	Oui
12	Moyen	Elevée	Fort	Oui
13	Chaud	Normale	Faible	Oui

Jouer = Oui

Pluvieux

Jour	Température	Humidité	Vent	jouer
4	Moyen	Elevée	Faible	Oui
5	Frais	Normale	Faible	Oui
6	Frais	Normale	Fort	Non
10	Moyen	Normale	Faible	Oui
14	Moyen	Elevée	Fort	Non

?

ID3 : Exemple

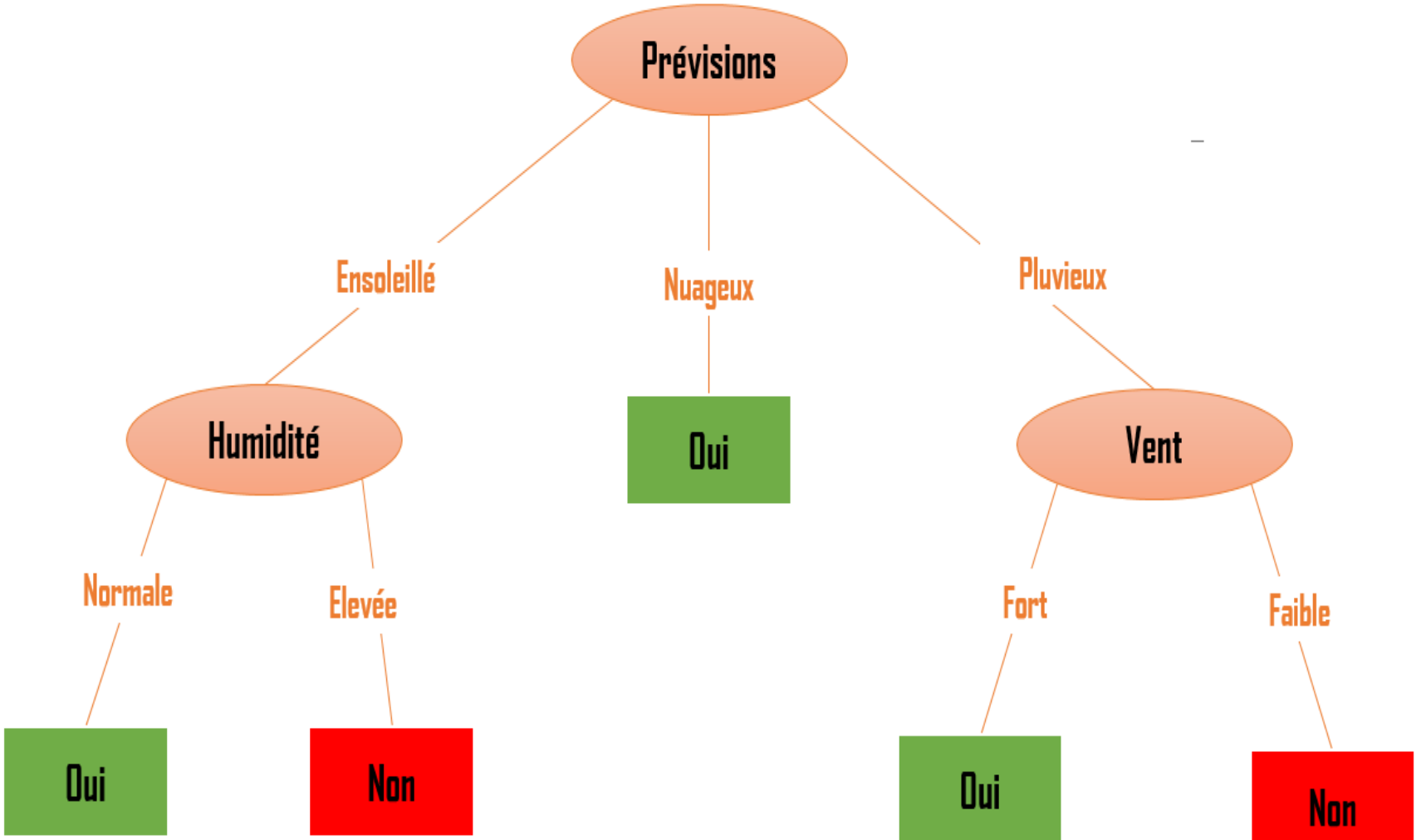
Il faut maintenant continuer à mettre des nœuds après **Ensoleillé** et **Pluvieux** car tous les exemples ne donnent pas le même résultat.

Il faut donc à cet effet déterminer pour **Ensoleillé** et **Pluvieux** le meilleur attribut à tester en utilisant à nouveau le gain. Cependant il n'est plus utile de tester le gain de Prévisions étant donné qu'il vient d'être utilisé.

. L'arbre de décision finale obtenu par ID3 donné ci-dessous :

Algorithmes de classification supervisée

ID3 : Exemple



Algorithmes de classification supervisée

C4.5

Présentation

L'algorithme C4.5 utilisé pour générer un arbre de décision développé a été proposé en 1993, toujours par Ross Quinlan, pour pallier les limites de l'algorithme ID3 vu précédemment.Il est classé numéro 1 dans le top 10 des algorithmes dans le domaine de l'exploration de données publié par Springer LNCS en 2008.

Algorithmes de classification supervisée

C4.5 : Améliorations

Quelques extensions à ID3 introduit par l'algorithme C4.5 sont :

1. **Gestion des attributs de valeur inconnue**
2. **Gestion des attributs à valeur sur intervalle (attributs continus)**
3. **L'élagage de l'arbre de décision après création**
4. **Gestion des données d'apprentissage avec des valeurs d'attribut manquantes**
5. **Gestion des attributs avec des coûts différents**.

En bref, C4.5 est une amélioration d'ID3 qui permet de travailler à la fois avec des données discrètes et des données continues. Il permet également de travailler avec des valeurs d'attribut absentes. Enfin, C4.5 élague l'arbre construit afin de supprimer les règles inutiles et de rendre l'arbre plus compact.

Le Pseudo-code de l'algorithme C4.5 est :

Algorithme 2 C4.5

Si Tous les échantillons de la liste appartiennent à la même classe **alors**
    Créer simplement un nœud feuille pour l'arbre de décision disant de choisir cette classe
Fin Si
Si Aucune des fonctionnalités ne fournit de gain d'information **alors**
    Créer un nœud de décision plus haut dans l'arbre en utilisant la valeur attendue de la classe.
Fin Si
Si Instance de classe inconnue précédemment rencontrée **alors**
    Créer un nœud de décision plus haut dans l'arbre en utilisant la valeur attendue.
Fin Si
Pour chaque attribut a , trouvez le rapport de gain d'information normalisé à partir du fractionnement sur a .
Soit **a'** l'attribut avec le gain d'information normalisé le plus élevé.
Créez un nœud de décision qui se divise sur **a'** .
Réexaminez les sous-listes obtenues en divisant sur **a'** , et ajoutez ces nœuds en tant qu'enfants de node .

Présentation

L'algorithme des **K plus proches voisins** ou **K-nearest neighbors (kNN)** est un algorithme de Machine Learning qui appartient à la classe des algorithmes d'apprentissage supervisé simple et facile à mettre en œuvre qui peut être utilisé pour résoudre les problèmes de classification et de régression.

- En classification knn, le résultat est une **classe d'appartenance**.
- En régression kNN, le résultat est la **valeur** pour cet objet. Cette valeur est la moyenne des valeurs des k plus proches voisins.

Algorithmes de classification supervisée

Knn : Principe

L'intuition derrière l'algorithme des **K plus proches voisins** est l'une des plus simples de tous les algorithmes de Machine Learning supervisé . Il fonctionne de la maniere suivante :

1. Sélectionner le nombre K de voisins
2. Calculer la distance
3. Prendre les K voisins les plus proches selon la distance calculée.
4. Parmi ces K voisins, compter le nombre de points appartenant à chaque catégorie.
5. Attribuer le nouveau point à la catégorie la plus présente parmi ces K voisins.

Les distances pouvant être utilisés pour la classification sont :

- **La distance de Minkowski** : avec p un entier positif non nul.

$$d(i,j) = \sqrt[p]{\sum_{r=1}^{M} |x_{ir} - x_{jr}|^p}$$

- **La distance euclidienne :** Elle est un cas particulier de la distance de Minkowski pour p = 2 :

$$d(i,j) = \sqrt{\sum_{r=1}^{M} |x_{ir} - x_{jr}|^2}$$

- **La distance de Manhattan :** Elle est également un cas particulier de la distance de Minkowski pour p = 1

$$d(i,j) = \sum_{r=1}^{M} |x_{ir} - x_{jr}|$$

Le Pseudo-code de l'algorithme knn est :

Algorithme 3 Knn

Entrée(s):

- un ensemble de données D
- une fonction de définition distance d
- Un nombre entier K

Sorties(s): la moyenne ou mode des variables **y** des **K** plus proches observations.

Pour Une nouvelle observation **X** dont on veut prédire sa variable de sortie **y** **faire**
    Calculer toutes les distances de cette observation **X** avec les autres observations du jeu de données **D**
    Retenir les **K** observations du jeu de données **D** les proches de **X** en utilisation le fonction de calcul de distance **d**
    Prendre les valeurs de **y** des **K** observations retenues :
    Si on effectue une régression **alors**
        calculer la moyenne (ou la médiane) de **y** retenues
    Fin Si
    Si on effectue une classification **alors**
        calculer le mode de **y** retenues
    Fin Si
Fin pour
Retourner la valeur calculée dans l'étape 3 comme étant la valeur qui a été prédite par K-NN pour l'observation X.

Exemple

Supposons que vous ayez un ensemble de données représentant des animaux avec trois caractéristiques : le poids, la taille et la vitesse. Chaque animal est également associé à une étiquette qui indique son type : soit "Chien", soit "Chat".

Poids (kg)	Taille (cm)	Vitesse (km/h)	Type
25	50	30	Chien
10	30	20	Chat
15	35	25	Chat
30	60	35	Chien
20	40	28	Chien
12	32	22	Chat

Exemple

Votre tâche consiste à prédire le type d'un nouvel animal ayant un poids de 18 kg, une taille de 38 cm et une vitesse de 26 km/h en utilisant l'algorithme des k plus proches voisins avec k=3.

1. Calculez les distances entre le nouvel animal et tous les points d'entraînement.
2. Sélectionnez les k=3 points les plus proches.
3. Déterminez la classe majoritaire parmi les k voisins.
4. Prédisez le type du nouvel animal en fonction de la classe majoritaire.

Soluton de l'exemple

1. Pour le nouvel animal avec un poids de 18 kg, une taille de 38 cm et une vitesse de 26 km/h, voici les distances entre chaque point d'entraînement et le nouvel animal :

Poids (kg)	Taille (cm)	Vitesse (km/h)	Distance
25	50	30	20.396
10	30	20	18.439
15	35	25	15.132
30	60	35	23.345
20	40	28	10.630
12	32	22	12.083

Soluton de l'exemple

2. Les trois points les plus proches du nouvel animal sont les suivants :

Poids (kg)	Taille (cm)	Vitesse (km/h)	Distance
15	35	25	15.132
20	40	28	10.630
12	32	22	12.083

3. Parmi les trois voisins les plus proches, deux appartiennent à la classe "Chat" et un appartient à la classe "Chien". Donc la classe majoritaire est "Chat".
4. La prédiction finale pour le nouvel animal est donc "Chat".

Algorithmes de classification supervisée

Bayes naïf

Présentation

La méthode de classification naïve bayésienne ou classifieur bayésien (ou encore estimateur bayésien)est un algorithme d'apprentissage supervisé (supervised machine learning) qui permet de classifier un ensemble d'observations selon des règles déterminées par l'algorithme lui-même. c'est un type de classification probabiliste simple basée sur le théorème de Bayes avec une forte indépendance (dite naïve) des hypothèses. Leur particularité est de prédire la valeur des paramètres du modèle en termes de probabilité.

A la base de la classification naïve bayésienne se trouve le théorème de Bayes avec l'hypothèse simplificatrice, dite naïve, d'indépendance entre toutes les paires de variables.
Le théorème de Bayes est donnée par la relation :

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)}$$

Le problème de classification ici revient à estimer la probabilité de chaque classe c_i sachant un vecteur de caractéristiques $\vec{f}$.

La probabilité de chaque caractéristique f_j sachant une classe c_i est estimée selon le type de ces valeurs : discrètes, binaires ou continues.

- **Cas des valeurs discrètes** : On utilise la ici loi multinomiale.

  Par exemple, la couleur des cheveux avec les valeurs : brun, auburn, châtain, roux, blond vénitien, blond et blanc. La probabilité d'une caractéristique f_j sachant une classe c_i est le nombre des occurrences de ce critère dans la classe ($|c_i|f_j$) divisé par le nombre de ces occurrences dans tout l'ensemble de données.

$$P(f_j|c_i) = \frac{|c_i|f_j}{\sum_{c_j} |c_j|f_j}$$

- **Cas des valeurs binaires** : on utilise la loi de Bernoulli.
- **Cas des valeurs continues** : on utilise la loi normale (loi gaussienne). Par exemple, le poids, le prix, etc. En se basant sur les données d'entrainement avec N échantillons, on calcule l'espérance μ et la variance σ^2 de chaque caractéristique f_j et chaque classe c_i.
  Donc, la probabilité qu'une caractéristique f_j ait une valeur x sachant une classe c_i suit la loi normale.

$$P(f_j = x|c_i) = \frac{1}{2\pi\sigma_{ij}^2} e^{-\frac{(x-\mu_{ij})^2}{2\sigma_{ij}^2}}$$

L'espérance μ et la variance σ^2 sont données par :

$$\mu_{ij} = \frac{1}{\sum_{|c_i|} \sum_{k=1}^{|c_i|} x_k} / x_k$$

$$\sigma_{ij}^2 = \frac{1}{\sum_{|c_i|-1} \sum_{k=1}^{|c_i|} (x_k - \mu_{ij})^2} / x_k$$

Soit le tableau des données ci-dessous contenant 8 échantillons :

Sexe	Taille (cm)	Poids (kg)	Pointure
masculin	182	81.6	30
masculin	180	86.2	28
masculin	170	77.1	30
masculin	180	74.8	25
féminin	152	45.4	15
féminin	168	68.0	20
féminin	165	59.0	18
féminin	175	68.0	23

On souhaite donc vérifier si un échantillon avec les caractéristiques suivantes : {taille=183, poids=59, pointure=20} est féminin ou masculin.

Les classes sont : **masculin** et **féminin**.

On a les probabilités suivantes :
P(masculin) = 4/8 = 0.5 P(féminin) = 4/8 = 0.5

La phase d'apprentissage consiste à calculer l'espérance et la variance de chaque caractéristique et classe.

Sexe	μ **(taille)**	σ^2 **(taille)**	μ **(poids)**	σ^2 **(poids)**	μ **(pointure)**	σ^2 **(pointure)**
masculin	178	29.333	79.92	25.476	28.25	5.5833
féminin	165	92.666	60.1	114.04	19.00	11.333

On peut à présent déterminer le sexe de l'échantillon avec :

$$f_{i,j}(x) = \frac{1}{2\pi\sigma_{i,j}^2} exp(\frac{-1}{2\sigma_{i,j}^2}(x - \mu_{i,j})^2)$$

pour une variable j dans le groupe i.

Pour la variable taille (t) dans le groupe masculin (m) on a donc :

$$P(t|m) = f_{t,m}(x) = \frac{1}{2\pi \times 29.333} exp(\frac{-1}{2 \times 29.333}(183 - 178)^2)$$
$$\approx 0,048102$$

En réalisant ce calcul pour chacune des autres caractéristiques on obtient :

$P(\text{taille}|\text{masculin}) = 4.8102 \times 10^{-2}$
$P(\text{poids}|\text{masculin}) = 1.4646 \times 10^{-5}$
$P(\text{pointure}|\text{masculin}) = 3.8052 \times 10^{-4}$

$P(\text{taille}|\text{féminin}) = 7.2146 \times 10^{-3}$
$P(\text{poids}|\text{féminin}) = 3.7160 \times 10^{-3}$
$P(\text{pointure}|\text{féminin}) = 1.1338 \times 10^{-1}$

Nous rappelons que nous avons obtenu plus haut les probabilités suivantes :

$P(\text{masculin}) = 0.5$

$P(\text{féminin}) = 0.5$

Posons : m=masculin, f=féminin. On aura donc :

$$P_p(\text{m}) = P(\text{m})P(\text{taille}|\text{m})P(\text{poids}|\text{m})P(\text{pointure}|\text{m})$$

$$P_p(\text{f}) = P(\text{f})P(\text{taille}|\text{f})P(\text{poids}|\text{f})P(\text{pointure}|\text{f})$$

Ainsi on a donc :

$P_p(\text{féminin}) = 1.5200 \times 10^{-5}$

$P_p\ (\text{masculin}) = 1.3404 \times 10^{-10}$

Comme la probabilité (postérieure) féminin est supérieure à la probabilité (postérieure) masculin, l'échantillon est plus probablement de sexe féminin.

Évaluation d'un classificateur

L'évaluation consiste à mesurer la différence entre un résultat attendu et un résultat obtenu. Elle peut se faire en utilisant plusieurs mesures dont quelques unes sont :

- **Matrice de contingence ou de confusion** : Cette technique utilise un corpus étiqueté de documents pour lequel on connaît la vraie catégorie de chaque document, et le résultat obtenu par le classifieur.Pour un corpus, on construit la matrice de contingence pour chaque classe , qui fournit 4 informations essentielles :
  - **Vrai Positif (VP)** : documents correctement classés ;
  - **Vrai Négatif(VN)** : documents correctement non classés ;
  - **Faux Positif (FP)** : documents incorrectement classés ;
  - **Faux Négatif (FN)** : documents incorrectement non classés.

Évaluation d'un classificateur

- **Matrice de confusion d'une classe**

Catégorie C_i		**Jugement expert**	
		Oui	Non
Jugement classifieur	Oui	**VPi**	**FPi**
	Non	**FNi**	**VNi**

- **Matrice de confusion de plusieurs classes**

		Expert	
	$C = \{c_1, c_2, ..., c_{\|C\|}\}$	C_i	$\neg C_i$
Classifieur	C_i	$VP = \sum_{i=1}^{\|C\|} VP_i$	$FP = \sum_{i=1}^{\|C\|} FP_i$
	$\neg C_i$	$FN = \sum_{i=1}^{\|C\|} FN_i$	$VN = \sum_{i=1}^{\|C\|} VN_i$

Évaluation d'un classificateur

- **Le Rappel** : Elle est la proportion de documents correctement classés par le système par rapport à tous les documents de la classe Ci, elle mesure la capacité d'un système de classification à détecter les documents correctement classés . Elle est donnée par la relation :

$$\begin{aligned}\textbf{Rappel}(C_i) &= \frac{\textbf{Nombredededocumentsbienclassésdans}\textbf{C}_i}{\textbf{Nombredededocumentsdela}\textbf{C}_i} \\ &= \frac{\textbf{VP}_i}{\textbf{VP}_i + \textbf{FN}_i}\end{aligned}$$

Évaluation d'un classificateur

- **La précision :** est la proportion de documents correctement classés parmi ceux classés par le système dans Ci .Elle mesure la capacité d'un système de classification à ne pas classer un document dans une classe, un document qui ne l'est pas. Comme elle peut aussi être interprétée par la probabilité conditionnelle qu'un document choisi aléatoirement dans la classe soit bien classé par le classifieur. Elle est donnée par la relation

$$\begin{aligned}\textbf{Précision}(C_i) &= \frac{\textbf{Nombrededocumentsbienclassésdans}\textbf{C}_i}{\textbf{Nombrededocumentsclassésdans}\textbf{C}_i} \\ &= \frac{\textbf{VP}_i}{\textbf{VP}_i + \textbf{FP}_i}\end{aligned}$$

Évaluation d'un classificateur

Calcul de précision et rappel pour plusieurs classes

$$\boldsymbol{P} = \frac{VP}{VP+FP} = \frac{\sum_{i=1}^{|C|} VP_i}{\sum_{i=1}^{|C|} \left(VP_i + FP_i\right)}$$

$$\boldsymbol{R} = \frac{VP}{VP+FN} = \frac{\sum_{i=1}^{|C|} VP_i}{\sum_{i=1}^{|C|} \left(VP_i + FN_i\right)}$$

Évaluation d'un classificateur

- **Le bruit** est le pourcentage de textes incorrectement associés à une classe par le système. Il est donné par la formule :

$$\textbf{Bruit(B)} = \textbf{1-Précision(P)} = \frac{\textbf{FP}_i}{\textbf{VP}_i + \textbf{FP}_i}$$

- **Le silence** est le pourcentage de textes à associer à une classe incorrectement non classés par le système. Il est donné par la formule

$$\textbf{Silence(S)} = \textbf{1-Rappel(R)} = \frac{\textbf{FN}_i}{\textbf{VP}_i + \textbf{FN}_i}$$

Évaluation d'un classificateur

- La **F-mesure** est le plus usuel des indicateurs. Elle prend en compte la valeur relative de la précision et du rappel.Elle est calculée par la formule :

$$\textbf{F-measure} = \frac{2 \times \textbf{précision} \times \textbf{rappel}}{\textbf{précision} + \textbf{rappel}}$$

- **Le taux de succès** est le rapport entre les documents bien classés sur le nombre total des documents du corpus.Il se calcule par la formule :

$$\textbf{Taux succès} = \frac{\textbf{VP} + \textbf{VN}}{\textbf{VP} + \textbf{FP} + \textbf{VN} + \textbf{FN}}$$

Évaluation d'un classificateur

- **Taux d'erreur :** est le rapport entre les documents mal classés sur le nombre total des documents du corpus. Il se calcule par la formule :

$$\textbf{Taux d'erreur} = 1 - \textbf{Taux succès} = \frac{\textbf{FP} + \textbf{FN}}{\textbf{VP} + \textbf{FP} + \textbf{VN} + \textbf{FN}}$$

Problèmes de classification

La classification automatique de texte, tout comme d'autres domaine de la science est soumis lui aussi à plusieurs contraintes qui sont, d'ailleurs, essentielles à l'essor de cette discipline. Quelques uns sont :

1. La polysémie
2. La redondance sémantique
3. Le temps d'apprentissage
4. Problème d'étiquetage

Exercice

Exercice

Considérons l'ensemble des données d'entrainement ci-dessous contenant des documents (d) auquel on a associé leur classes (A ou B).

d	c	d	c
aa	A	ba	A
ab	A	bb	B

1. Calculer P(A), P(B), P(a|A), P(b|A), P(a|B), P(b|B). Utiliser la technique de Laplace Smoothing pour le calcul de probabilités.
2. Classer chacune des nouvelles données ci-dessous, supprimer tous les mots inconnus : aaba , a, bbba, bccbba, bbb

Exercice

Exercice (suite)

Considérons l'ensemble des données d'entrainnement précédent comme étant cette fois ci des données de test

Documents	Bonne classe
aaba	A
a	A
bbba	A
bccbba	A
bbb	B

3. Construire la matrice de confusion
4. Calculer les mesures suivantes : le rappel, la précision, le bruit, le silence,la F-measure , le taux de succès et le taux d'erreur.
5. Que conclusion faire ?

CHAPITRE 5

Le Clustering

Définition et objectifs

- Le **clustering**, également appelé **regroupement**, est une technique d'exploration de données non supervisée qui vise à organiser un ensemble d'objets en groupes homogènes appelés clusters.
- L'objectif du clustering est de regrouper des objets similaires ensemble, tout en les séparant des objets qui sont différents.
- En d'autres termes, le clustering cherche à identifier des structures ou des relations cachées dans les données en se basant uniquement sur les caractéristiques intrinsèques des objets, sans avoir de connaissances préalables sur des classes ou des étiquettes spécifiques.

Clustering

Domaines d'application

Le clustering trouve de nombreuses applications dans divers domaines tels que :

- le marketing pour la segmentation de clients ;
- la biologie pour la classification de gènes ;
- l'analyse des réseaux sociaux pour la détection de communautés ;
- la reconnaissance de formes ;
- la recherche d'informations, etc

Concepts de base du clustering

1. Un **cluster** est un groupe ou un ensemble d'objets similaires qui sont regroupés ensemble en fonction de leur similarité ou de leur proximité. Les objets au sein d'un cluster partagent des caractéristiques communes et sont plus similaires entre eux qu'avec les objets d'autres clusters.
2. Un **point** représente une entité individuelle ou un objet dans l'espace des données. Chaque point est décrit par un ensemble de caractéristiques ou de variables, qui servent à évaluer leur similarité ou leur distance par rapport aux autres points.
3. La **Distance** : La distance mesure la dissimilitude ou l'éloignement entre deux points. Différentes mesures de distance sont utilisées, telles que la distance euclidienne, la distance de Manhattan, la distance de Minkowski, etc. La distance est utilisée pour quantifier la différence ou la similitude entre les points et forme la base de nombreux algorithmes de clustering.

Concepts de base du clustering

4. Une **Similarité** : La similarité mesure la similitude ou la proximité entre deux points. Elle est souvent utilisée comme mesure alternative à la distance, où une similarité élevée entre deux points indique une plus grande similitude ou proximité entre eux.

Méthodes de clustering

1. **Clustering hiérarchique** : qui crée une structure de clusters emboîtés
   - Le **clustering agglomératif** commence par considérer chaque point comme un cluster séparé, puis fusionne itérativement les clusters en fonction de leur similarité, jusqu'à obtenir un seul cluster global.
   - Le **clustering divisif** commence par un seul cluster global, puis divise itérativement le cluster en sous-clusters plus petits en fonction de la dissimilarité entre les points.

Méthodes de clustering

1. **Clustering non hiérarchique** : qui forme des clusters indépendants sans structure hiérarchique.
   - Le **K-means** Le K-means est l'une des méthodes les plus populaires de clustering non hiérarchique. Il vise à partitionner les points en K clusters en minimisant la distance entre les points et les centres de clusters
   - Le **K-medoids** Le K-medoids est une variante du K-means où les centres de cluster sont choisis parmi les points de données réels plutôt que comme des moyennes. Cela rend le K-medoids plus robuste aux valeurs aberrantes.
   - **DBSCAN (Density-Based Spatial Clustering of Applications with Noise)** c'est une méthode de clustering basée sur la densité. Elle identifie les zones de haute densité de points et les considère comme des clusters, tout en identifiant les points isolés comme du bruit.

Présentation

- K-means est un algorithme non supervisé de clustering . Il permet de regrouper en **K clusters** distincts les observations du data set(ensemble des données).
- Les données similaires se retrouveront dans un même cluster. Par ailleurs, une observation ne peut se retrouver que dans un cluster à la fois (exclusivité d'appartenance). Une même observation, ne pourra donc, appartenir à deux clusters différents.
- Pour pouvoir regrouper un jeu de données en K cluster distincts, l'algorithme K-Means a besoin d'un moyen de comparer le degré de similarité entre les différentes observations.
- Ainsi, deux données qui se ressemblent, auront une distance de dissimilarité réduite, alors que deux objets différents auront une distance de séparation plus grande.

K-means

Application

Les champs d'application de K-Means sont nombreux, il est notamment utilisé en :

- la segmentation de la clientèle en fonction d'un certain critère (démographique, habitude d'achat etc. ...)
- Utilisation du clustering en Data Mining lors de l'exploration de données pour déceler des individus similaires. Généralement, une fois ces populations détectées, d'autres techniques peuvent être employées en fonction du besoin.
- Clustering de documents (regroupement de documents en fonction de leurs contenus. Pensez à comment Google Actualités regroupe des documents par thématiques.)

K-means

Fonctionnement

Le K-means divise les données en k clusters en minimisant la somme des distances au carré entre chaque point et le centroïde de son cluster. Il alterne entre l'affectation des points aux centroïdes les plus proches et la mise à jour des positions des centroïdes jusqu'à convergence.

1. Sélectionner aléatoirement k centroïdes initiaux.
2. Répéter jusqu'à convergence :
   - Assigner chaque point au centroïde le plus proche.
   - Mettre à jour les positions des centroïdes en calculant les moyennes des points assignés.

K-means

Pseudo-code

Ci-dessous le Pseudo-code de l'algorithme K-means

1. **Entrées :**
   - K : le nombre de cluster à former
   - training set : Ensemble ou matrices des données
2. choisir aléatoirement le k points qui seront considérés comme les centroides (les centres de clusters)
3. **Répéter** :
   - (Ré)affecter chaque objet O au cluster C_i de centre M_i tel que dist(O,M_i) est minimal
   - Recalculer M_i de chaque cluster (le barycentre)
4. **jusqu**'à la convergence

K-means I

Exemple

Soit A={1,2,3,6,7,8,13,15,17} un ensemble des données numériques. Créer 3 clusters à partir de A

- On prend 3 objets au hasard. Supposons que c'est 1, 2 et 3. Ca donne C1={1}, M1=1, C2={2}, M2=2, C3={3} et M3=3
- Chaque objet O est affecté au cluster au milieu duquel, O est le plus proche. 6 est affecté à C3 car dist(M3,6)<dist(M2,6) et dist(M3,6)<dist(M1,6) On a :
  C1={1}, M1=1
  C2={2}, M2=2
  C3={3, 6,7,8,13,15,17}, M3=69/7=9.86
- dist(3,M2)<dist(3,M3)→ 3 passe dans C2. Tous les autres objets ne bougent pas. C1={1}, M1=1, C2={2,3}, M2=2.5,C3={6,7,8,13,15,17} et M3= 66/6=11

K-means II

Exemple

- dist(6,M2)<dist(6,M3) $\rightarrow$ 6 passe dans C2. Tous les autres objets ne bougent pas. C1=$\{1\}$, M1=1, C2=$\{2,3,6\}$, M2=11/3=3.67, C3=$\{7,8,13,15,17\}$, M3= 12
- dist(2,M1)<dist(2,M2) $\rightarrow$ 2 passe en C1. dist(7,M2)<dist(7,M3)$\rightarrow$ 7 passe en C2. Les autres ne bougent pas. C1=$\{1,2\}$, M1=1.5, C2=$\{3,6,7\}$, M2=5.34, C3= $\{8,13,15,17\}$, M3=13.25
- dist(3,M1)<dist(3,M2)$\rightarrow$ 3 passe en 1.
- dist(8,M2)<dist(8,M3)$\rightarrow$ 8 passe en 2 C1=$\{1,2,3\}$, M1=2, C2=$\{6,7,8\}$, M2=7, C3=$\{13,15,17\}$, M3=15
- Plus rien ne bouge, on s'arrete car l'algorithme converge. On a donc les classes finales suivantes : **C1=$\{1,2,3\}$**, **C2=$\{6,7,8\}$** et **C3=$\{13,15,17\}$**

K-means

Limites

Quelques limites de cet algorithme sont :

- Un nombre K grand peut conduire à un partitionnement trop fragmenté des données. Ce qui empêchera de découvrir des patterns intéressants dans les données. Par contre, un nombre de clusters trop petit, conduira à avoir, potentiellement, des cluster trop généralistes contenant beaucoup de données. Dans ce cas, on n'aura pas de patterns "fins" à découvrir.
- Pour un même jeu de données, il n'existe pas un unique clustering possible. La difficulté résidera donc à choisir un nombre de cluster K qui permettra de mettre en lumière des patterns intéressants entre les données. Malheureusement il n'existe pas de procédé automatisé pour trouver le bon nombre de clusters.

K-means : Exercice

Exercice : Utilisation de l'algorithme K-means

Soit l'ensemble D des entiers suivants : D= { 2, 5, 8, 10, 11, 18, 20 } . On veut répartir les données de D en trois (3) clusters, en utilisant l'algorithme Kmeans. La distance d entre deux nombres a et b est calculée ainsi : d(a , b) = |a - b| (la valeur absolue de a moins b)

1. Appliquez l'algorithme Kmeans en choisissant comme centres initiaux des 3 clusters respectivement : 8, 10 et 11. Montrez toutes les étapes de votre calcul.
2. Donnez le résultat final et précisez le nombre d'itérations qui ont été nécessaires.
3. Peut-on avoir un nombre d'itérations inférieur pour ce problème ? Discutez.