République Algérienne Démocratique et Populaire
Ministère de l'enseignement Supérieur et de la Recherche Scientifique
Université de Mohamed El Bachir El-Ibrahimi de Bordj Bou Arréridj
Faculté des Mathématiques et d'Informatique
Département d'informatique

MEMOIRE

Présenté en vue de l'obtention du diplôme
Master en informatique
Spécialité: **T**echnologies de l'**I**nformation et de la **C**ommunication

THEME

Extraction des règles d'association à partir du texte

Présenté par :
CHARIF Khier
SOUALEM Charifa

Soutenu publiquement le : / /2023
Devant le jury composé de:
Président : BOUTOUHAMI Sara
Examinateur : BENABID Sonia
Encadreur : SAIFI Lynda MCB à l'université de BBA

2022/2023

Remerciements

Remercions tout d'abord **ALLAH** qui nous a donné la foi et le courage pour accomplir ce projet. Nous tenons aussi à exprimer notre reconnaissance et profonde gratitude à notre promoteur **Dr Sayfi Lynda** pour nous avoir encadrés durant cette année, pour sa forte présence et sa disponibilité, pour son exigence scientifique et ses précieuses orientations méthodologiques, pour son encouragement et sa patience.

Que les membres du jury trouvent ici nos plus vifs remerciements pour avoir accepté d'honorer par leur jugement notre travail. Aussi, nous adressons nos remerciements à tous nos enseignants de notre université pour nous avoir appris le goût de l'effort et du travail.

Un grand merci aussi à toute personne qui de près ou de loin a contribué à ce que cet humble travail voit le jour.

Dédicace

Je Dédie ce Modeste Travail A mes Très Chers Parents A mes Chers Frères et Sœurs A tout ma Famille Particulièrement ma Petite Sœur Aicha A Tous mes Amis Particulièrement : Oussama, Halim, Jouzef, Ayoub, Issam A mon Chikh Houssemeddine.

A mon Encadreur : Sayfi Lynda

A mon Partenaire : Soualem Charifa.

Khier

À la Mémoire de Mon Père qui est Toujours Présent dans mon Esprit et mon Cœur.

À ton âme Papa, et la Mémoire du Grand Amour que tu m'as offert je dédie ma réussite.

CHARIFA

Résumé :

Ce mémoire présente une application d'extraction de règles d'association à partir de données textuelles en utilisant deux méthodes. La première méthode utilise l'algorithme traditionnel Apriori avec CountVectorizer. La deuxième méthode combine TF-IDF avec les algorithmes Apriori et FP-Growth pour donner la priorité aux termes importants. Des évaluations expérimentales sont réalisées pour comparer l'efficacité de ces méthodes dans l'extraction de règles d'association. Les résultats sont analysés et discutés, en fournissant des aperçus sur les performances de chaque approche.

Les mots clés : Extraction de règles d'association, Données textuelles, Apriori, CountVectorizer, TF-IDF, FP-Growth

Abstract:

This memory presents an application of extracting association rules from textual data using two methods. The first method utilizes the traditional Apriori algorithm with CountVectorizer. The second method combines TF-IDF with the Apriori and FP-Growth algorithms to prioritize important terms. Experimental evaluations are conducted to compare the effectiveness of memory methods in extracting association rules. The results are analyzed and discussed, providing insights into the performance of each approach.

Keywords: Association rule extraction, Textual data, Apriori, CountVectorizer, TF-IDF, FP-Growth

الملخص:

هذه المذكرة تقدم تطبيقًا لاستخراج قواعد الارتباط من البيانات النصية باستخدام طريقتين. تستخدم الطريقة الأولى خوارزمية Apriori التقليدية بواسطة CountVectorizer. تجمع الطريقة الثانية بين TF-IDF وخوارزميات Apriori و FP-Growth لإعطاء الأولوية للمصطلحات المهمة. يتم إجراء تقييمات تجريبية لمقارنة فعالية هذه الطرق في استخراج قواعد الارتباط. يتم تحليل ومناقشة النتائج، وتوفير رؤى حول أداء كل نهج.

الكلمات المفتاحية: استخراج قواعد الارتباط، البيانات النصية، Apriori، CountVectorizer، TF-IDF، FP-Growth.

Liste des abréviations

ECLAT : **E**quivalence **C**lass **C**lustering and bottom-up **L**attice **T**raversal

FP : **F**requent **P**attern

IDE : **E**nvironnement de **D**éveloppement **I**ntégré

IDF : **I**nverse **D**ocument **F**requency

IE : **E**xtraction d'**I**nformation

IR : **I**nformation **R**etrieval

GUI : **I**nterface **U**tilisateur **G**raphique

KNN : **Kn**earest **N**eighbors

NER : **N**amed **E**ntity **R**ecognition

NLP : **N**atural **L**anguage **P**rocessing

NLTK : **N**atural **L**anguage **T**ool**k**it

PIL **P**ython **I**maging **L**ibrary

TF : **T**erm **F**requency

Table des matières

Table des figures

Liste des tableaux

Introduction Générale

1. Contexte

Ces dernières années, la fouille de textes est devenue un sujet d'étude de plus en plus important dans le domaine de l'informatique. Cette discipline vise à extraire des informations à partir de données textuelles, afin de les analyser et d'en tirer des conclusions utiles pour la prise de décision ou la résolution de problèmes spécifiques. Dans ce contexte, l'extraction d'association de règles est une technique fondamentale qui permet de découvrir des relations entre des éléments d'un corpus de textes, en se basant sur des critères statistiques ou sémantiques. Cette technique est essentielle dans de nombreux domaines tels que la recherche d'informations, la veille technologique, la recommandation de produits ou de services, etc.

2. Problématique

Avec le volume croissant de données textuelles, il est devenu une tâche ardue pour les experts d'analyser et d'extraire manuellement des informations utiles. Par conséquent, il est nécessaire de développer des techniques automatisées pour l'extraction efficace des connaissances à partir des données textuelles.

3. Objectif

Dans cette étude, notre objectif est l'extraction des association entre les termes fréquents à partir de données textuelles en utilisant deux méthodes différentes. La première méthode est basée sur l'algorithme Apriori et la seconde méthode utilise TF-IDF avec deux algorithmes, Apriori avec TF-IDF et FP-Growth avec TF-IDF. Nous allons comparer les performances de ces méthodes en termes de temps de traitement, de mémoire utilisée et de précision des résultats obtenus. L'objectif est de déterminer quelle méthode est la plus efficace pour extraire des associations de termes fréquents à partir de données textuelles.

4. Plan de mémoire

Ce mémoire contient cinq chapitres, chacun pour leur partie, vont développer les manières de comprendre, d’analyser et de maîtriser l’extraction des règles d’association à partir du texte.

- Dans **l'introduction générale**, nous allons présenter le contexte général de notre étude, la problématique, l'objectif et le plan de ce mémoire.

- **Chapitre 1 : Text mining** ; au début du premier chapitre, nous allons expliquer le processus de Text mining et ses techniques.
- **Chapitre 2 :** Se focalise sur **les Règles d'association** ; Nous expliquerons les principes de base des règles d'association, les différentes mesures utilisées pour évaluer les règles d'association, ainsi que les différents algorithmes utilisés pour générer ces règles.
- **Chapitre 3 :** Ce chapitre sera consacré à **description de notre approche**. Nous nous intéressons à l'extraction de règles d'association à partir de données textuelles.
- **Chapitre 4 : Réalisation de l'application** ; Dans cette partie nous allons faire présenter le langage de programmation, les outils et packages utilisés dans le développement de notre application, ensuite, on va présenter l'interface utilisateur graphique de ce dernier, à la fin nous allons expliquer la technologie derrière l'application.
- **Chapitre 5 : Etude expérimental** ; dans le dernier chapitre, nous présenterons note étude expérimental ; nous allons inclure quelques exemples de documents sur un domaine d'étude choisi.
- La section **Conclusion générale** et perspectives conclut notre travail en expliquant les problèmes et en résumant les contributions apportées et les perspectives d'avenir.
- La section **Annexe** présente des codes python importants de notre application.

1 TEXT MINING

Introduction

L'exploration de texte a attiré une attention croissante ces dernières années en raison des grandes quantités de données textuelles, qui sont créées dans une variété de réseaux sociaux, du Web et d'autres applications centrées sur l'information. Les données non structurées sont la forme de données la plus simple pouvant être créée dans n'importe quel scénario d'application. En conséquence, il y a eu un énorme besoin de concevoir des méthodes et des algorithmes qui peuvent traiter efficacement une grande variété d'applications de texte.

1.1 Le Data mining

Le data mining ou fouilles de donnes, est un processus d'extraction des informations utiles à partir des grandes bases de données en utilisant des technique de statistique, de Machine Learning et de visualisations de données, pour identifier dans ces données des relations, des modèles, des tendances cachée, ces informations après peuvent être utilisées pour prendre des décisions basées sur les données.**[1]**

1.2 Le Textmining

Le Textmining ou fouilles de textes ou l'exploration de données, est un sous-discipline de Data mining qui traite spécifiquement les données textuels non structuré comme les emailes, les fiches Word, the news headline et tous les fiches textuels, et après ça l'extraction des connaissances utiles à l'aide de spécifique techniques (comme **NLP (**Techniques de Traitement de Langage Naturel**)** par exemple). Les connaissances extraites peuvent être utilisées pour le développement des algorithmes d'apprentissages capable d'analyser des données non structurées, classer les documents, analysé les sentiments, NER (**N**ame d**E**ntity **R**ecognition), l'exploration de données peut être appliqué dans divers domaines tel que le Marketing, réseaux sociaux, la Santé…etc.

1.3 Le processus de Textmining

Les étapes nécessaires pour effectuer le processus de Textmining sont :

- **La sélection ou collection des données :**

  Reçue il de données textuelles non structurées à partir de différentes sources, telles que des sites web, des plateformes de médias sociaux, des articles de presse, des avis de clients, etc.

- **Pré-traitement de données :**

  Prétraiter les données textuelles, nettoyé et normalisé les textes par la tokenisation et la lemmatization et enlevant des informations irrelevantes, telles que des mots vides, des caractères spéciaux**.**

- **Représentation du texte :**

  Convertissez les données textuelles après le pré-traitement en un format numérique tel qu'un sac de mots (bag of words) ou une matrice de term-document, pour les préparer à l'analyse.

- **Exploration du texte :**

  Explorez les données textuelles pour identifier des motifs, des relations et des insights à l'aide de spécifiquement techniques telles que l'analyse de la fréquence des mots, (les nuages de mots) et l'analyse de sentiment**.**

- **Analyse et visualisation du texte :**

  Appliquer des algorithmes d'apprentissage automatique tel que la classification, et visualisez les résultats pour les rendre plus faciles à comprendre et à interpréter.

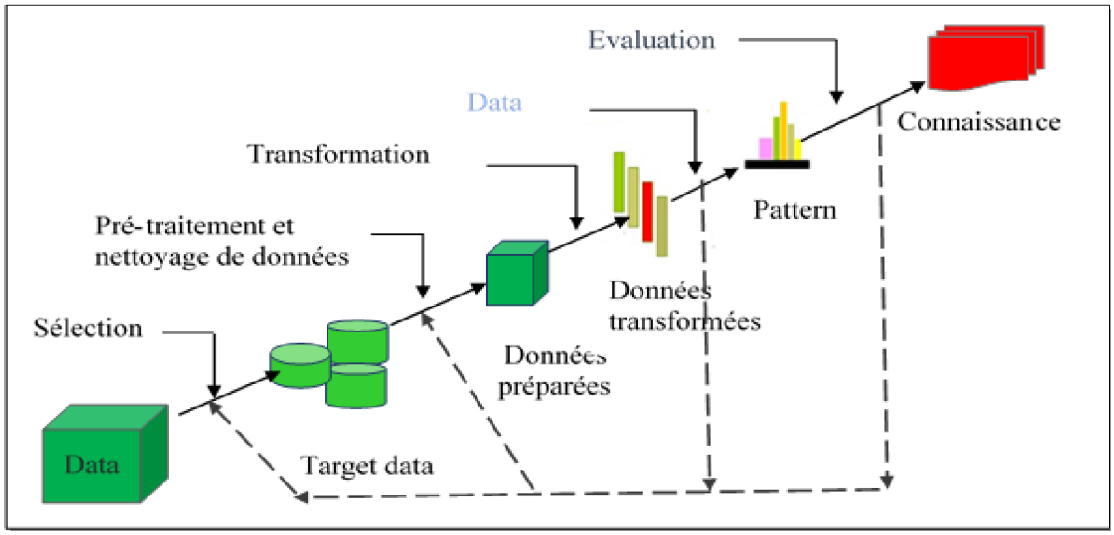


Figure 1-1 :Processus de fouille de données [2]

1.4 Les Techniques de Textmining :

Dans cette partie on va présenter les déférentes techniques sur le contenu des textes et vise à extraire et structurer les connaissances, et parmi ces techniques on trouve**[3]** :

1.4.1 Techniques de Traitement de Langage Naturel (NLP) :

Un traitement automatique est :

✓ Une suite d'actions ou calculs à faire par la machine. Le Traitement Automatique des Langues a pour objectif de traiter des données linguistiques (textes) exprimées dans une langue dite "naturelle". **[4]**

✓ La Conception de programmes capables de traiter automatiquement des données linguistiques de type : textes écrits ; dialogues écrits ou oraux ; unités linguistiques (mots, phrases, énoncés, ...)

Les tâches impliquées dans cette technique peuvent inclure la tokenization, élimination
des mots vides et filtrage de textes, Lemmatisation, La racinisation (ou troncature) :

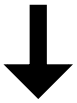
- **La tokenization**

Est le processus de décompositions d'un texte en plusieurs pièces appelé tokens (jetons).

Exemple:

"Full sanctions on Russian exports would be a pivotal moment for the oil market, potentially touching off a sustained high price cycle with few precedents"

Tokenization



Tokens:| Full | sanctions | on | Russian | exports | would | be | a | pivotal | moment | for | the| oil |market | potentially | touching | off | a |sustained |high| price | cycle | with| few |precedents |

- **Le filtrage :**

Le filtrage est une technique utilisée pour prétraiter les données textuelles en vue de les analyser et de les manipuler plus facilement.

Il peut inclure différentes techniques, telles que :

- Convertir en minuscules
- Retirer les mots vides tels que "le", "la", "et"… etc.
- Retirer les signes de ponctuation et les caractères spéciaux.

- **Lemmatization :**

La lemmatization est une technique qui réduit les mots à leur forme de base, ou **lemme,** par exemple pour un verbe, ce sera son infinitif. Pour un nom, son masculin singulier. L'idée étant encore une fois de ne **conserver que le sens des mots** utilisés dans le corpus

Exemples :

Cherche, cherchent -> chercher

ai, as, a, -> avoir

Belles, beaux, beau -> beau

1.4.2 L'extraction d'information ("IE")

Est la technique la plus utilisé parmi les techniques de Textmining, qui se concentre sur l'extraction automatique d'informations structuré à partir de données textuelles non structuré. L'objectif d'IE est de convertir un texte non structuré en un format structuré qui peuvent être facilement analysé et traité par l'ordinateur, cela implique l'identification et l'extraction d'informations spécifiques, tel que les entité nommées(noms de personnes, noms d'organisations, emplacement…etc.),les relations entre les entités Les informations extraites sont ensuite organisées dans un format structuré, tel qu'une base de données ou un tableau, pour une analyse et un traitement ultérieurs. L'extraction d'informations est largement utilisée dans diverses applications, telles que la recherche d'informations, l'analyse des sentiments et la réponse aux questions.

1.4.3 La recherche d'informations (information retrieval "IR")

L'IR est un processus d'extraction qui fournit des informations pertinentes à l'utilisateur en fonction de sa requête et de classer les résultats en fonction de leur pertinence. Cette technique utilise déférentes algorithmes pour suivre et surveiller les comportements des utilisateurs et découvrir et évaluer la pertinence de chaque document et les classer basé sur sa pertinence. Parmi les systèmes IR les plus connues sont les moteurs de recherches Google et Yahoo.

1.4.4 La catégorisation (classification supervisé)

Dans l'apprentissage supervisé, on fournit des données d'entrée qui sont étiquetées avec les sorties souhaitées. Le but est que l'algorithme puisse "apprendre" en comparant sa sortie réelle avec les sorties enseignées pour trouver des erreurs et adapter le modèle. Ce processus permet à l'algorithme de prédire les valeurs d'étiquettes pour des données non étiquetées.

Les algorithmes d'apprentissage supervisé peuvent être utilisés pour des taches telles que la classification de texte, l'analyse de sentiments et pour faire des prédictions.

Parmi ses méthodes :

Les arbres de décision**,** les réseaux neurones**,** la méthode des k plus proche voisins (KNN) ou la classification bayésienne.**[5]**

1.4.5 Clustering (classification non supervisée)

La classification non supervisée regroupe des points de données similaires en fonction de leurs caractéristiques. Le but du clustering est de diviser les données en groupes ou clusters homogènes, de sorte que les points de données d'un cluster soient plus similaires les uns aux autres qu'ils ne le sont avec les points de données d'autres clusters. Les algorithmes de clustering utilisent des métriques de similarité pour déterminer les relations entre les points de données et les regrouper en clusters. Il est utilisé dans diverses applications telles que la segmentation d'images, les études de marché et autres. Parmi les méthodes que nous utilisons dans le clustering :

- **Les règles d'association**

Les règles d'association sont des techniques d'apprentissage non supervisé qui identifient les relations entre les éléments ou les variables dans un ensemble de

données. Elles révèlent des motifs, ce qui aide à prendre des décisions éclairées et à formuler des recommandations.

- **K-means**

Tous simplement le K-means est une technique de segmentation qui s'objectif est de partitionner des données textuelles en utilisant la mesure de la distance ou de la similarité entre les observations en K clusters.

Conclusion

En conclusion, l'exploration de texte et l'exploration de données sont des techniques puissantes pour extraire des connaissances à partir de grandes quantités de données. La classification non supervisée est une technique d'exploration de données importante qui peut être utilisée pour résoudre une variété de problèmes réels. L'utilisation de cette technique est devenue de plus en plus importante dans de nombreuses industries et leur potentiel pour des applications supplémentaires est vaste.

2 LES REGLES D'ASSOCIATION

Introduction

Dans ce chapitre, nous discuterons des règles d'association, un concept fondamental dans la fouille de données qui nous permet d'identifier les relations entre les éléments d'un ensemble de données. Nous expliquerons les principes de base de la fouille de règles d'association, les différentes mesures utilisées pour évaluer les règles d'association, ainsi que les différents algorithmes utilisés pour générer ces règles. En comprenant les règles d'association, nous pouvons extraire des informations précieuses à partir de données textuelles et prendre des décisions éclairées en fonction de ces informations.

2.1 Les Règles d'association

2.1.1 Définition

On peut définir une règle d'association comme une règle de la forme B→H dans laquelle B et H sont des ensembles de terme. Une telle règle peut être interprétée de la façon suivante : "Les documents qui possèdent les termes de B possèdent également les termes de H". B et H sont appelés des motifs (ou itemsets). **[7]**

2.1.2 Motif

Soit ''T'' et ''D'' deux ensembles et R une matrice. 'T'est un ensemble de termes et 'D'' est un ensemble de textes tel que **:**

T= {a, b, c, d, e} et D= {d1, d2, d3, d4, d5, d6}

La matrice R représente la relation binaire qui existe entre l'ensemble T et D.

R	A	B	c	d	E
d1	1	0	1	0	1
d2	0	1	1	1	1
d3	1	1	1	0	1
d4	1	0	0	0	0
d5	0	1	1	1	1
d6	1	0	0	1	0

Tableau 2-1 **: Matrice Termes-Documents**

On appelle un motif tous les sous-ensembles de T. Un motif ''t'' est inclus dans un texte 'di' si $\forall t \in$ T, (t', di) = 1. Un motif de taille K est appelé k-motif.
Par exemple : d3 et d5 contiennent le 4-motif. **[8]**

2.1.3 Motif fréquent

On dit un motif est fréquenté si un support d'un motif "t" supérieur à un support minimal (qui est déterminé par l'utilisateur).
Support (t)≥ min_sup ; tel que le min_sup est le support minimal

2.1.4 Support d'une règle d'association

Le support d'une règle d'association (A→C) est une mesure de la fréquence d'apparition de la Règle, il représente le pourcentage de documents qui contiennent A et C "support (A U C)" divisé par le nombre total de document :

$$support(A \rightarrow C) = \frac{\text{support}(A \cup C)}{D}$$

$$support(A \rightarrow C) \in [0,1]$$

D : le nombre total de documents

Le support est un indicateur de fiabilité de la règle

2.1.5 Confiance d'une règle d'association

La confiance, est une mesure exprimée à l'aide de la probabilité conditionnelle d'avoir l'évènement C sachant que l'évènement A s'est produit, c'est une mesure descriptive qui prend ses valeurs dans l'intervalle [0, 1]. **[7]**

Il est défini par la formule:

$$conf(A \rightarrow C) = \frac{\text{suppot}(A \cup C)}{\text{suppot}(A)}$$

2.1.6 Le Lift

Le Lift est une mesure statistique, symétrique, représente le rapport d'indépendance entre l'antécédent et le conséquent de la règle. Il prend ses valeurs dans l'intervalle [0, +∞ [, mais en pratique, il est rare que le Lift dépasse 10 ou 20.[9]

Elle est définie par la formule:

Lift(X→Y) = (supp (X, Y) / N) / (supp(X) / N * supp(Y) / N)

2.1.7 Le leverage

Le "leverage" est une mesure couramment utilisée dans l'exploration de règles d'association pour évaluer le degré de corrélation entre l'occurrence simultanée d'un antécédent et d'un conséquent dans la même transaction par rapport à ce qui serait attendu s'ils étaient indépendants. Une valeur de zéro pour le "leverage" indique une absence de corrélation, tandis qu'une valeur supérieure à zéro indique une corrélation positive. Plus la valeur du "leverage" est élevée, plus la corrélation est forte.

La formule pour calculer la liaison (leverage) entre deux éléments X et Y est:

Leverage (X -> Y) = support (X -> Y) - (support(X) * support(Y))

2.2 Extraction des règles d'association

La plupart des algorithmes de recherche de règles d'association (parmi eux : apriori) adopte une stratégie qui consiste à décomposer le problème en deux étapes:

1. Extraction des ensembles d'items fréquents
2. Génération des règles d'association.

Dont l'objectif est d'extraire toutes les règles de grande confiance à partir des ensembles d'items fréquents trouvés dans l'étape précédente. **[10]**

2.3 Les Algorithmes de Génération de Règles d'association

Il existe plusieurs algorithmes de génération de règles d'association. Ils utilisent suivants les notions de support et de confidence pour déterminer la pertinence des règles
d'association. Parmi eux on site :

2.3.1 Algorithme Apriori :

Apriori est un algorithme classique de recherche de règles d'association introduit par Agrawal et Srikant en 1993. **[7]**
C'est le premier algorithme destiné à la recherche de règles d'association. Apriori génère les motifs fréquents puis les relie entre eux pour générer les règles d'association. Il se base essentiellement sur la propriété d'anti-mono tonicité existante entre les motifs. Elle est utilisée à chaque itération de l'algorithme Apriori afin minimiser au maximum le nombre de motifs candidats à tester.

2.3.1.1 Le principe de l'algorithme Apriori :

La description de l'algorithme Apriori se résume dans les étapes suivantes :

- **Trouver les 1-Itemsets :** Parcourir la base de données pour identifier les éléments individuels et collecter les ensembles d'items ayant un support supérieur ou égal à min_sup.
- **Générer les (k + 1)-Itemsets :** Générer des candidats pour les (k + 1)-Itemsets en combinant des k-Itemsets fréquents en utilisant la propriété d'Apriori.
- **Filtrer les candidats :** Vérifier le support de chaque candidat (k + 1)-Itemset et conserver uniquement ceux qui satisfont le seuil min_sup.
- **Répéter les étapes 2 et 3 :** Itérer les étapes 2 et 3 jusqu'à ce qu'aucun nouveau k-Itemset fréquent ne puisse être trouvé.

L'algorithme Apriori explore de manière itérative les ensembles d'items de taille k en se basant sur les ensembles d'items fréquents de taille k-1 déjà trouvés. Il utilise la propriété d'Apriori, qui stipule que si un ensemble d'articles est infréquenté, tous ses ensembles supérieurs (ensembles plus larges le contenant) seront également infréquentés. Cette propriété permet à l'algorithme de générer et de filtrer efficacement les candidats d'ensembles d'items, réduisant l'espace de recherche et

améliorant l'efficacité. En répétant le processus jusqu'à ce qu'aucun nouveau k-Items et fréquent ne puisse être trouvé, l'algorithme Apriori découvre les ensembles d'items fréquents dans la base de données.

2.3.1.2 Avantages et inconvénients

L'algorithme Apriori réduit considérablement la taille d'articles candidats de plus qu'il est facile à mettre en œuvre. Cependant, il souffre des limitations par rapport à la nécessité de nombreuses analyses de base de données ainsi que le grand nombre d'ensembles éléments candidats qui peuvent être encore généré si le nombre total d'ensembles des éléments fréquents augmente. **[11]**

2.3.2 AlgorithmeFP-Growth

L'algorithme FP-Growth a été introduit par Han et Al. en 2000, ils ont dit qu'il est actuellement l'une des approches les plus rapides pour l'extraction fréquente d'ensembles d'articles .C'est une méthode différente des approches par niveaux qui permet d'extraire des ensembles d'articles fréquents sans générer de candidats, évitant ainsi les parcours et les visites répétées de la base de données.**[12]**

2.3.2.1 Le principe de l'algorithme FP-Growth :

La description de l'algorithme FP-Growth se résume dans les deux étapes suivantes:

- **Construire l'arbre FP :** Parcourir la base de données pour construire un arbre FP, qui est une structure de données compacte représentant les itemsets fréquents et leurs relations. L'arbre FP se compose d'un nœud racine et de branches conditionnelles représentant les itemsets.

- **Générer les itemsets fréquents :** Parcourir l'arbre FP pour trouver les itemsets fréquents en exploitant de manière récursive les motifs conditionnels. Cela implique de trouver les items individuels fréquents, de construire des arbres FP conditionnels, et de répéter le processus jusqu'à ce qu'aucun itemset fréquent supplémentaire ne puisse être trouvé.

En résumé, l'algorithme FP-Growth construit un arbre FP à partir de la base de données et extrait de manière récursive les itemsets fréquents en exploitant la

structure de l'arbre et les motifs conditionnels. Il élimine le besoin de générer des candidats itemsets, ce qui le rend plus rapide que les autres algorithmes.

2.3.2.2 Avantages et inconvénients

L'algorithme FP-Growth résout le problème de la nécessité de nombreuses analyses de base de données, vu qu'il ne fait que deux balayages de la base des transactions.

Néanmoins, cela ne garantit pas, dans le cas où la base de transactions est trop volumineuse, que toute la structure de l'arbre FP sera maintenue en mémoire centrale. De plus, la construction de la structure FP-tree peut prendre du temps et peut consommer beaucoup de ressources système.**[12]**

2.3.3 Algorithme ECLAT

ECLAT (Equivalence Class Transformation) a été introduit par Zaki, Parthasarathy, Ogihara et Li en 1997. **[13]** Eclat a été conçu pour sur monter les inconvénients de l'algorithme Apriori. Il utilise la mémoire agrégée du système en partitionnant les candidats en ensembles disjoints à l'aide du partitionnement par classe d'équivalence. Il dissocie la dépendance entre les transformateurs en partant de manière à ce que le coût de redistribution puisse être amorti par les itérations ultérieures. Eclat utilise une structure de base de données verticale qui regroupe toutes les informations pertinentes dans la liste d'objets.**[14]**

2.4 Utilisation des règles d'association

L'intérêt des règles d'association dans la fouille de textes est multiple, on peut citer :

2.4.1 Analyse de concepts formels :

Les règles d'association permettent d'organiser des concepts dans une structure hiérarchique à partir de laquelle il est possible d'observer des corrélations entre les individus et leurs propriétés communes. En effet, on peut hiérarchiser les concepts on utilisant correspondance de Galois pour créer un graphe de concept muni d'une relation d'ordre entre les concepts. On appelle ce graphe un treillis de Galois. La construction d'un treillis de Galois permet de se donner une structure mathématique pour l'analyse de concepts issus d'un domaine.

2.4.2 Extraction d'information (EI) :

L'extraction de règles d'association permet de réaliser des tâches d'extraction d'information pour remplir des patrons. À ce titre, le système TEXTRISE[16]illustre l'application d'un processus de fouille de Texte pour l'EI. TEXTRISE apprend à remplir certains attributs de patrons pour de nouveaux textes à partir de règles d'association apprises sur d'autres patrons. Dans une notice bibliographique par exemple, un patron possède un attribut auteur inconnu mais un attribut mots-clés complet {mc_1, mc_2, mc_3}.

Si durant la phase d'apprentissage nous avons une règle d'association (mc_1, $mc_2 \Rightarrow$ aut_1) appelée soft-Matching Rule, ce texte est attribué, à un degré de confiance près, àaut_1**. [17]**

2.4.3 Veille technologique et stratégique :

La veille stratégique (appelée également business intelligence). C'est un processus de mise à jour périodique d'informations. Il offre une aide précieuse à la prise de décision pour les gérants d'entreprise. Les règles d'association révèlent des implications entre termes et permettent de faire des suivis scientifiques.

2.4.4 Recherche d'information (RI) :

La liste de documents pertinents qui constitue une réponse relative à une requête est fondée sur le lien de cooccurrence entre les termes de la requête et leur fréquence d'apparition ensemble dans les textes. L'utilisation des motifs fermés fréquents permet, par navigation dans le treillis de Galois correspondant, de répondre à une requête par les documents constituant l'extension d'un concept.

Conclusion

En conclusion, les règles d'association sont un outil puissant dans la fouille de textes qui nous permettent de découvrir des relations précieuses entre les éléments d'un ensemble de données. En utilisant différentes mesures et algorithmes pour évaluer les règles d'association, nous pouvons obtenir des informations qui peuvent guider les processus de prise de décision. Dans le prochain chapitre, nous

appliquerons ces principes pour extraire des règles d'association à partir de données textuelles et démontrer leur utilisation pratique dans des scénarios réels.

3 Description de notre approche

Introduction

Comme nous l'avons mentionné dans le chapitre précèdent que la fouille de règles d'association est une technique utilisée pour découvrir des modèles et des relations entre des éléments ou des variables dans un ensemble de données. Cette technique a été largement utilisée dans divers domaines tels que l'analyse de panier d'achat, l'exploration de l'utilisation du web et la bio-informatique.

Dans cette étude, nous nous intéressons à l'extraction de règles d'association à partir de données textuelles telles que des articles de presse, des actualités…etc. Nous utilisons ensuite ces règles pour déterminer les facteurs qui ont un impact sur les prix du pétrole. Pour ce faire, nous comparons deux méthodes couramment utilisées pour l'extraction de règles d'association : CountVectorizer avec l'algorithme Apriori, et TF-IDF avec deux algorithmes différents, Apriori et FP-Growth. Nous explorons spécifiquement comment chaque approche affecte la qualité et la quantité des règles d'association extraites, y compris des métriques telles que le support, la confiance et le lift.

3.1 La Méthodologie

Pour extraire des règles d'association à partir des données textuelles, deux méthodes ont été utilisées : Countvectorizer et TF-IDF. Le but était de comparer l'efficacité des deux méthodes pour extraire des règles d'association de haute qualité.

3.1.1 Countvectorizer

Dans notre projet, la première méthode que nous avons utilisée s'appelle CountVectorizer. Nous avons choisi ce nom car nous avons utilisé la bibliothèque CountVectorizer de Python comme composant clé de cette méthode. CountVectorizer

est une technique utilisée pour convertir des données textuelles en une représentation numérique pouvant être traitée par des algorithmes d'apprentissage automatique.

Il se compose de 4 étapes :

- Prétraitement de données
- Représentation numérique (Création une matrice term-document)
- Extraction des motifs fréquents
- Génération des règles d'association

3.1.1.1 Prétraitement de données

Convertir tous les documents en minuscules, supprimer les mots vides, tokenizer les documents et les avoir lemmatisés en utilisant un tagger de parties du discours (POS). Cette étape vise à standardiser les données textuelles et à éliminer toute information non pertinente qui pourrait affecter les performances de l'algorithme.

3.1.1.2 Création une matrice terme-document

On a utilisé python Countvectorizer pour convertir les documents prétraités en une matrice de fréquence de termes. Cette méthode a compté la fréquence de chaque terme dans chaque document, créant ainsi une représentation vectorielle du document. Cette matrice est appelée matrice terme-document, qui représente la fréquence de chaque terme dans chaque document, dans laquelle chaque ligne représente un document et chaque colonne représente un terme unique trouvé dans la collection. La forme de ma matrice que nous intéresse exprime l'apparition des termes avec une méthode binaire.

Par exemple : A est une matrice term document et α une cellule de cette matrice

$$\alpha \text{ ij} = \begin{cases} 1, si\ le\ term\ j\ est\ present\ dans\ le\ decument\ i \\ 0, sinon \end{cases}$$

A	**Term1**	**Term2**	**Term3**
D1	1	0	0
D2	0	1	0
D3	1	0	1

Tableau 3-1 : **Matrice term document**

3.1.1.3 Extraction des motifs frequents

Avant d'extraire les motifs fréquents, il est nécessaire de convertir la matrice en un format Pandas Data Frame afin de la passer à l'algorithme Apriori. Après cette conversion, nous aurons une matrice qui contiendra et représentera toutes les données de notre collection. En utilisant l'algorithme Apriori que nous avons déjà expliqué dans le chapitre précédent, nous recherchons les ensembles d'items qui apparaissent fréquemment ensemble dans les documents analysés. Le support est utilisé comme mesure pour déterminer quels itemsets sont considérés comme fréquents. Une fois que les motifs fréquents sont identifiés, ils servent de base pour la génération des règles d'association dans l'étape suivante.

3.1.1.4 Génération des règles d'association

Une fois que les ensembles d'itemsets fréquents sont extraite en utilisant l'algorithme Apriori, l'étape suivante consiste à générer des règles d'association à partir d'eux. Les règles d'association sont simplement des relations logiques entre deux éléments ou plus dans un ensemble de données transactionnelles.

Le processus de génération de règles d'association implique les étapes suivantes :

- Déterminez les seuils de support et de confiance minimums : Tout comme lors de l'extraction d'ensembles des motifs fréquents, définirez des seuils de support et de confiance minimums pour générer des règles d'association. Ces seuils sera aideront à filtrer les règles qui ne sont pas assez fortes.
- Générez toutes les règles possibles : Pour chaque ensemble d'éléments fréquents, on générer toutes les règles possibles en divisant l'ensemble en deux sous-ensembles non vides. Par exemple, si nous avons un ensemble d'itemsets fréquents de taille 3 : {Term1, Term2, Term3}, on peut générer les règles {Term1, Term2} → {Term3}, {Term1, Term3} → {Term2}, {Term2, Term3} → {Term1}, {Term1} → {Term2, Term3}, {Term3} → {Term1, Term2} et {Term2} → {Term1, Term3}.
- Calculez les mesures d'évaluation pour les règles d'association : Pour chaque règle générée, calculez son support, sa confiance, son lift, son leverage.
- Filtrez les règles faibles : Ensuite, nous pouvons filtrer les règles en fonction des seuils minimums que nous avons définis pour chacune de ces mesures d'évaluation.

- Triez et affichez les règles restantes : Triez les règles restantes par leur confiance et afficher-les dans l'ordre décroissant.

Cependant, il est également possible de trier les règles en fonction d'autres mesures d'évaluation telles que le lift ou la conviction. En fin de compte, la méthode de tri dépendra des objectifs spécifiques de l'analyse et des besoins de l'utilisateur.

3.1.2 TF-IDF

Dans notre projet, la deuxième méthode que nous avons utilisée s'appelle TF-IDF. Nous avons choisi ce nom car nous avons utilisé la mesure TF-IDF (Term Frequency-Inverse Document Frequency) comme composant clé de cette méthode. Nous avons appliqué cette mesure en conjonction avec deux algorithmes différents, à savoir Apriori et FP-Growth.

TF-IDF: TF-IDF signifie**T**erm **F**requency **I**nverse **D**ocument Frequency of records. Il peut être défini comme le calcul de la pertinence d'un mot d'une série ou d'un corpus par rapport à un texte.

Il se compose de 6 étapes :

- Prétraitement de données
- Calcul de l'importance des termes (TF-IDF)
- Sélection les n term ayant les valeurs d'importance les plus élevées
- Représentation numérique (Création une matrice term-document)
- Extraction des motifs fréquentes avec l'algorithme Apriori / FP-Growth
- Génération des règles d'association

3.1.2.1 Prétraitement de données

Le texte a été prétraité de la même manière que dans la première méthode de Countvectorizer.

3.1.2.2 Calcul de l'importance des termes (TF-IDF)

TF-IDF a été utilisé pour calculer l'importance de chaque terme dans le corpus. Cette mesure prend en compte à la fois la fréquence du terme dans le document et sa rareté dans le corpus. Le calcul implique deux étapes :

- **Fréquence de terme (TF) :** Cela mesure la fréquence d'un terme dans un document. Il est calculé en divisant le nombre de fois où un terme apparaît dans un document par le nombre total de termes dans ce document.
  TF (t, d) = nombre d'occurrence de t dans d / nombre de mots dans d
- **Fréquence inverse de document (IDF) :** cela mesure la rareté d'un terme dans le corpus. Il est calculé en divisant le nombre total de documents dans le corpus par le nombre de documents contenant le terme, puis en prenant le logarithme de ce quotient.
  IDF(t) = $log10\frac{N}{N(t)}$
  N = Nombre Totale de documents
  N(t) = Nombre de documents contenant le terme t
  TF-IDF est ensuite calculé en multipliant la valeur TF par la valeur IDF.
  TF-IDF (t, d) = TF (t, d) * IDF(t)

3.1.2.3 Sélection les n term ayant les valeurs d'importance les plus élevées

Après avoir calculé les valeurs de TF-IDF pour ce terme dans le corpus, nous avons sélectionné les n termes ayant les valeurs les plus levées pour une analyse plus ciblée et précise. Dans notre application, nous avons donné à l'utilisateur le contrôle sur le choix de la valeur de n en pourcentage. Par défaut, nous avons fixé cette valeur à 0,25 (25%). Cela signifie que le système sélectionnera les premiers 25% des termes de chaque document, en se basant sur leur score TF-IDF.

Cette étape de sélection est cruciale dans notre méthode, car elle nous permet de focaliser notre analyse sur les termes les plus importants et pertinents pour chaque document. En choisissant les termes avec les scores TF-IDF les plus élevés, nous nous assurons de prendre en compte les termes qui ont le plus d'influence dans la représentation numérique et l'analyse des données textuelles.

En donnant à l'utilisateur la possibilité de définir le pourcentage, nous offrons une flexibilité pour adapter l'analyse à ses besoins spécifiques. Si l'utilisateur souhaite une analyse plus complète, il peut augmenter le pourcentage (n) pour inclure plus de termes. Cela se traduira par une couverture plus large des termes, offrant une vue plus complète de l'ensemble de données. D'autre part, si l'utilisateur souhaite une analyse plus ciblée et concise, il peut diminuer le pourcentage (n) pour se concentrer sur un sous-ensemble plus restreint de termes très importants.

3.1.2.4 Représentation numérique (Création une matrice terme-document)

À l'aide des termes sélectionnés, nous créons manuellement une matrice term-document qui représente numériquement les données textuelles. Dans cette méthode, chaque document est représenté par un vecteur où chaque terme sélectionné correspond à une caractéristique et la valeur associée représente son importance dans le document. Contrairement à la première méthode où nous avons utilisé CountVectorizer pour créer la matrice.

3.1.2.5 Extraction des motifs fréquents

Dans cette étape, nous utilisons deux méthodes :

- **L'algorithme Apriori** pour extraire les motifs fréquents à partir de la matrice term-document créée précédemment. L'algorithme Apriori explore progressivement l'espace des itemsets possibles, recherchant les ensembles d'items qui apparaissent fréquemment ensemble dans les documents analysés.
- **L'algorithme FP-Growth** pour extraire les motifs fréquents à partir de la matrice term-document. L'algorithme FP-Growth utilise une structure d'arbre compacte pour exploiter les relations entre les items et extraire efficacement les motifs fréquents.

1.2.6 Génération des règles d'association

Dans la deuxième méthode, les règles d'association sont générées de la même manière que dans la première méthode. Une fois que les motifs fréquents sont identifiés à l'aide des algorithmes Apriori et FP-Growth, nous utilisons les mêmes étapes de génération des règles d'association que celles décrites dans la première méthode.

Conclusion

En utilisant les méthodes CountVectorizer et TF-IDF, nous avons progressé vers notre objectif d'extraire des règles à partir de données textuelles. Les approches différentes nous ont permis d'acquérir une meilleure compréhension des données et d'identifier les termes importants pour extraire les règles de manière plus efficace.

4 Réalisation de l'application

Introduction

Dans ce chapitre, nous allons explorer les outils et packages utilisés dans le développement d'une application de génération de règles d'association à partir de documents texte. Nous examinerons également de plus près l'interface de l'application et ses différents composants, notamment ses widgets, ses boutons et sa barre d'outils…etc. En comprenant la technologie derrière l'application et son interface utilisateur, nous pouvons mieux comprendre le processus de génération des règles d'association ainsi que la flexibilité et le contrôle que l'application offre à ses utilisateurs.

4.1 Outils et langages utilisés

4.1.1 Langage de programmation : Python

Python est un langage de programmation de haut niveau, interprété, orienté objet et multi-paradigme. Il a été conçu dans les années 1990 par Guido van Rossum, et est connu pour sa syntaxe claire et concise, ainsi que pour sa grande facilité d'utilisation. Python est utilisé dans une variété de domaines, tels que le développement web, l'analyse de données, l'intelligence artificielle, l'automatisation de tâches, et bien plus encore.

4.2.2 Environnement de développement intégré (IDE):

PyCharm est un environnement de développement intégré (IDE) utilisé pour programmer en Python. Il fournit une analyse de code, un débogueur graphique, un testeur d'unité intégré, une intégration avec les systèmes de contrôle de version, et prend en charge le développement web avec Django. PyCharm est développé par la société tchèque JetBrains.

Il est multiplateforme, fonctionnant sur Microsoft Windows, MacOs et Linux. PyCharm a une édition professionnelle, publiée sous licence propriétaire et une édition communautaire publiée sous la licence Apache. L'édition communautaire de PyCharm est moins étendue que l'édition professionnelle.**[18]**

4.1.2 Les packages utilisés :

On a utilisé différentes bibliothèques de Python dans notre application tels que :

Nltk, sklearn, mlxtend, pandas, PIL, Tkinter

4.1.2.1 Nltk

NLTK (Natural Language Toolkit) est une bibliothèque largement utilisée en Python pour les tâches de traitement du langage naturel (NLP) telles que le prétraitement et le nettoyage de texte. Elle offre divers outils et méthodes pour la tokenization, le stemming, la lemmatisation et plus encore. Parmi ses méthodes que nous utilisons dans notre application, nous utilisons la méthode **nltk.word_tokenize** pour tokenizer les documents, la méthode **stopwords** pour supprimer les mots courants des documents et la méthode **WordNetLemmatizer** pour lemmatiser les documents. NLTK est considéré comme l'un des packages les plus essentiels pour les tâches de NLP en Python en raison de sa polyvalence et de sa facilité d'utilisation.**[19]**

4.1.2.2 Sklearn

Scikit-learn (ou sklearn) est une bibliothèque très utilisée en Python pour l'apprentissage automatique et l'exploration de données.[20]Dans notre application, nous avons importé la méthode **CountVectorizer** de sklearn, qui permet de convertir une collection de documents texte en une matrice de comptage de termes-document (terme-document matrix) pour l'analyse des fréquences des termes dans les documents. Nous avons utilisé CountVectorizer pour obtenir une matrice term-doc (terme-document) que nous avons ensuite utilisée comme entrée pour l'algorithme Apriori.

4.1.2.3 Mlxtend

La bibliothèque mlxtend est une bibliothèque Python pour l'apprentissage automatique et l'exploration de données. **[21]** Dans notre application, nous importons

l'algorithme **Apriori** et **FP-Growth** à partir de **mlxtend.frequent_patterns,** qui sont des algorithmes couramment utilisés pour trouver les itemsets fréquenté. Nous importons également la méthode **association_rules**, qui est responsable de la génération de règles d'association à partir des données traitées. La bibliothèque mlxtend est considérée comme une bibliothèque essentielle pour les tâches de fouille de données en raison de son efficacité et de sa simplicité d'utilisation.

4.1.2.4 Pandas

Pandas est une bibliothèque populaire en Python utilisée pour la manipulation de données.**[22]** Dans notre application, nous avons utilisé pandas pour convertir la matrice terme-document en un DataFrame pandas. Cela nous permet de mieux visualiser et manipuler les données, notamment pour la génération de règles d'association à l'aide de l'algorithme Apriori et FP-Growth de la bibliothèque mlxtend.

4.1.2.5 PIL

PIL (Python Imaging Library) est une bibliothèque de traitement d'images en Python qui permet de manipuler des images numériques en effectuant diverses opérations telles que l'ouverture, l'enregistrement et l'affichage d'images.**[23]** Nous avons utilisé PIL dans notre application pour charger et afficher des images, notamment pour afficher des icônes de fichiers et de boutons. PIL est largement utilisé pour le traitement d'images en Python et offre une grande variété de fonctions et de méthodes pour la manipulation d'images.

4.1.2.6 Tkinter

Tkinter est une bibliothèque GUI (Interface Utilisateur Graphique) standard de Python qui permet de créer des interfaces utilisateur fonctionnelles et attrayantes. [24]Elle est utilisée dans notre application pour créer l'interface utilisateur, visualiser les résultats et personnaliser les paramètres de l'algorithme.

4.2 Interface utilisateur graphique de notre application(G.U.I)

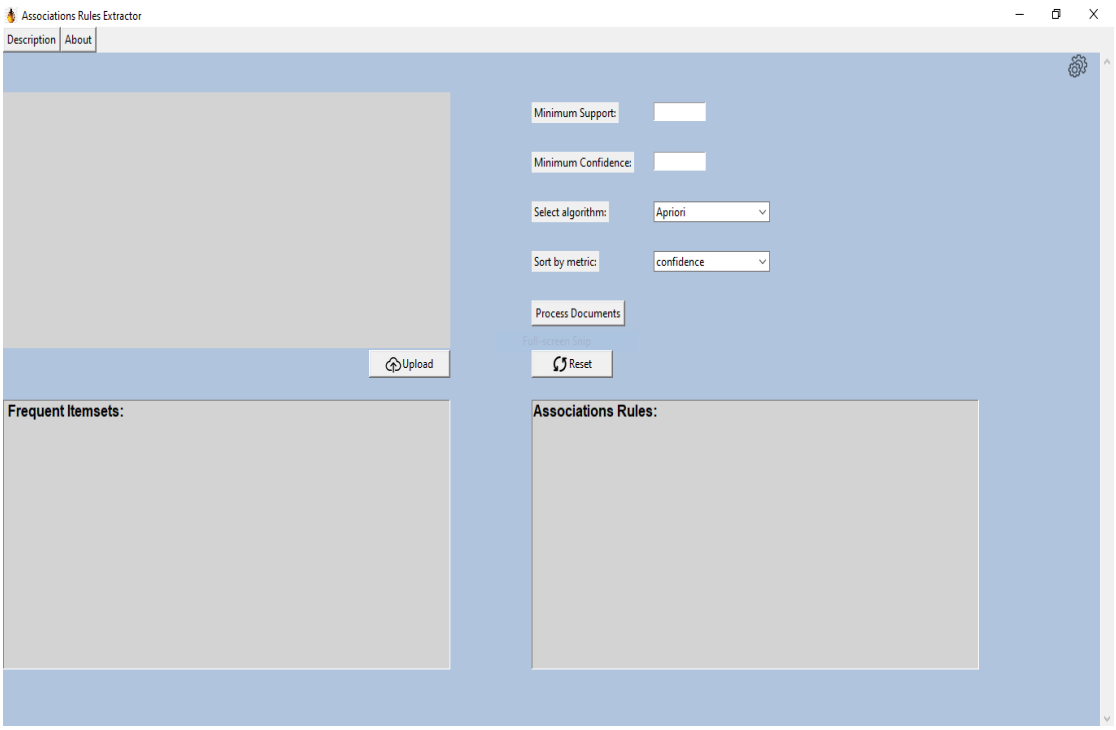


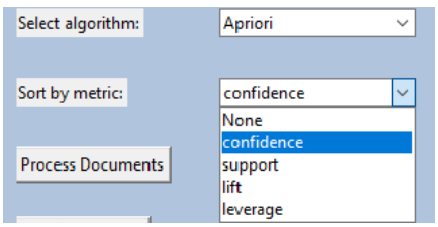
Figure 4-1 **: Interface Graphique de notre application**

Notre application dispose d'une interface graphique intuitive et conviviale pour les utilisateurs. Cette interface est composée de différents widgets, boutons, menus déroulants...etc. Dans cette section, nous allons examiner de plus près les éléments de l'interface utilisateur et expliquer leur fonctionnement.

1. **Les widgets :** Au coin supérieur droit, nous avons un widget qui affiche les noms et les icônes des fichiers téléchargés. Cela permet aux utilisateurs d'identifier facilement les fichiers qu'ils ont téléchargés pour l'analyse. En outre, nous avons deux autres widgets, l'un pour afficher les éléments fréquents et l'autre pour afficher les règles générées par l'algorithme choisi.

2. **Les zones de saisie :** nous avons deux cases à remplir. La première est destinée à la saisie du seuil de support minimal (min_sup) et le deuxième est destinée à la saisie du seuil de confiance minimal (min_conf).

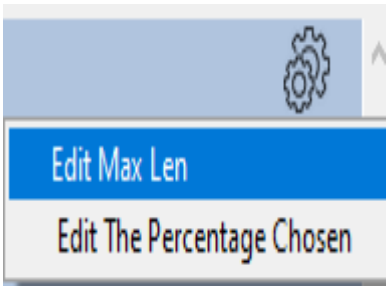
3. **Les listes déroulantes :** nous avons deux listes déroulantes. La première liste déroulante permet à l'utilisateur de sélectionner l'algorithme

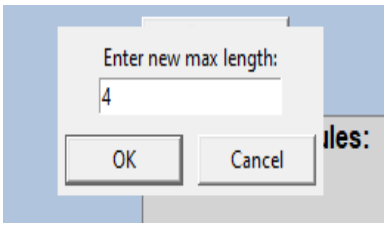


qu'il souhaite utiliser pour la génération des règles d'association, soit l'algorithme Apriori classique, soit l'algorithme Apriori avec TF-IDF qui est une amélioration de l'algorithme Apriori, qui utilise TF-IDF pour identifier les items fréquents qui ont une forte importance dans le corpus de documents, Soit l'algorithme FP-Growth avec TF-IDF. La deuxième liste déroulante permet à l'utilisateur de sélectionner la métrique qu'il souhaite utiliser pour ordonner les règles générées, soit la confiance (confidence), le support (support), le lift (lift) ou le levier (leverage). Cela donne à l'utilisateur un contrôle total sur le processus de génération de règles et la possibilité d'explorer les différentes options disponibles.

4. **Les boutons :** nous avons trois boutons. Le premier bouton « Upload » permet à l'utilisateur de télécharger les fichiers qu'il souhaite analyser. Le deuxième bouton « process documents » est utilisé pour lancer le processus d'analyse des documents après que l'utilisateur a sélectionné l'algorithme, la métrique, le minimum de support et de confiance appropriés. Le dernier bouton « Reset » est le bouton de réinitialisation, qui permet à l'utilisateur de réinitialiser toutes les valeurs précédemment saisies et de recommencer avec un nouvel ensemble de fichiers ou de paramètres.

5. **Le bouton icône :** Dans le coin supérieur droit de notre interface, il y a un bouton d'icône intitulé "Paramètres avancés". Lorsqu'on clique dessus, une liste de deux options s'affiche. La première option permet à l'utilisateur de modifier la longueur maximale des éléments extraits de chaque document, qui est un paramètre d'entrée pour les algorithmes Apriori et FP-Growth. Ceci est effectué via une boîte d'entrée Tkinter. La deuxième option permet à l'utilisateur de modifier le pourcentage des termes les plus importants sélectionnés à partir de chaque document lors de l'utilisation de TF-IDF. Cette fonctionnalité offre une plus grande flexibilité et un contrôle à l'utilisateur, lui permettant d'adapter l'analyse à ses besoins spécifiques.





6. La barre d'outils (toolbar) : elle contient deux boutons. Le premier bouton est intitulé "Description" et fournit des informations sur notre application, y compris notre approche et les algorithmes utilisés pour la génération de règles d'association. Le deuxième bouton est intitulé "about" et affiche des informations générales sur l'application, telles que le numéro de version et les détails du développeur. Ces boutons offrent aux utilisateurs un moyen rapide et pratique d'accéder aux informations de développement de l'application.

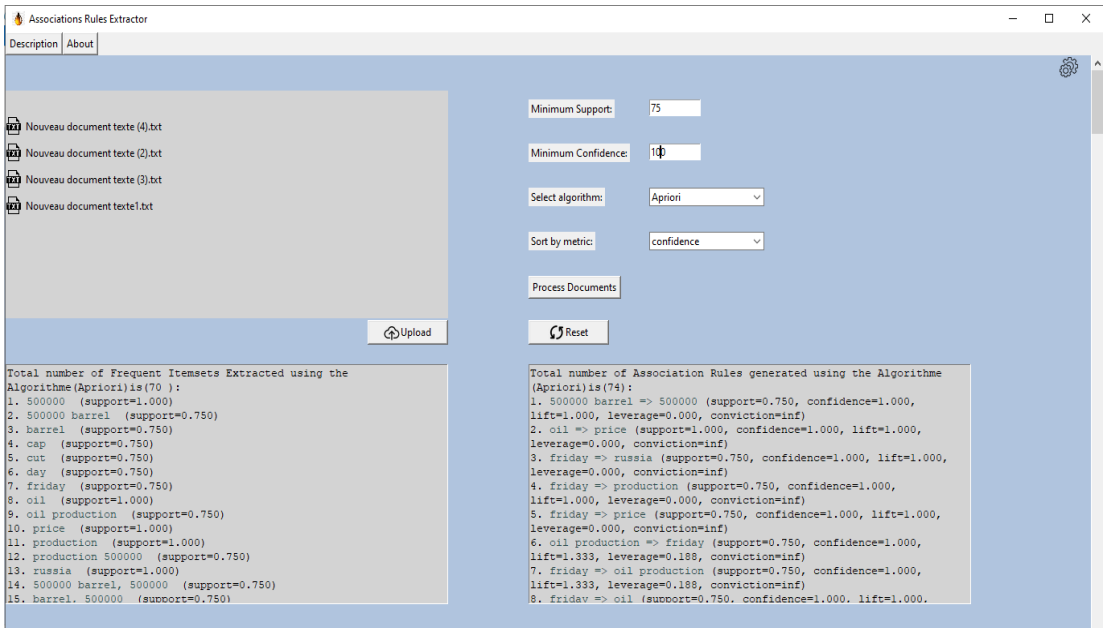


Figure 4-2 **: Interface Graphique de notre application après l'affichage**

Conclusion

En résumé, ce chapitre a discuté des outils et des packages utilisés dans la mise en œuvre de l'application d'extraction de règles d'association, ainsi que les fonctionnalités de l'interface. Avec ces informations, les utilisateurs peuvent avoir une compréhension claire des options offertes de l'application et de la manière d'interagir avec elle.

5 Etude expérimental

Introduction

Dans ce chapitre, nous allons inclure quelques exemples de documents sur un domaine d'étude, à savoir les facteurs et les accidents passés ayant un impact sur les prix du pétrole. Nous allons étudier ces exemples et discuter des résultats obtenus à partir de l'application d'association de règles.

5.1 Exemple 1 (Russia's invasion of Ukraine)

Remarque : Les textes utilisés sont en anglais

Text 1:

It seems that the invasion of Ukraine by Russia has led to an increase in crude oil prices, which has resulted in higher petrol prices. Waitomo Group, a New Zealand fuel company, has stated that they expect to see an increase of 10 cents per liter at the pump within the next week or two due to the recent rise in crude prices. The global oil price is expected to continue to rise as tensions between Russia and Ukraine worsen. Emeritus Professor Ralph Sims from Massey University suggests that for every dollar increase in the world oil price, there is roughly a one-cent increase in the purchase price of petrol at the pumps in New Zealand.

Text 2:

The invasion of Ukraine by Russia is expected to have a significant impact on global oil prices. , oil prices have already surged past $100 a barrel, and many experts predict that they will continue to rise. The global energy supply could be disrupted if the conflict escalates further, leading to reduced oil exports from Russia and other affected countries. This, in turn, would put further upward pressure on oil prices and lead to higher prices at the pump for consumers worldwide.

Text 3:

Since the invasion of Ukraine on February 24, crude oil prices spiked above $110-$120 at times. Gasoline and diesel fuel prices also rose around the world.

The prices of gasoline and diesel fuel, which are set by contract, tend to rise and fall more gradually than daily spot prices of crude oil, following crude prices with a lag. Retail prices of gasoline do not change every day with yesterday's spot price of crude, which is something that may only make sense to politicians.

Russia's invasion of Ukraine has raised the retail price of a gallon of gasoline by at least a dollar in the US (and much more in Europe), but that expensive energy is far more problematic for industry, transportation, and farming than just the "price at the pump.

5.1.1 Résultat 1 de l'exemple 1

Support=100% (3/3), confiance=100%,

Algorithm : **Apriori with Countvectorizer**

Itemsets	Support
invasion	100% (3/3)
invasion Ukraine	100% (3/3)
oil	100% (3/3)
oil price	100% (3/3)
price	100% (3/3)
pump	100% (3/3)
rise	100% (3/3)
Ukraine	100% (3/3)
invasion Ukraine, invasion	100% (3/3)
oil, invasion	100% (3/3)
oil price, invasion	100% (3/3)

Id	La règle d'association	Support	Confiance	Lift
1	oil price, oil, rise => invasion Ukraine	100% (3/3)	100%	1.0
2	invasion Ukraine => oil price, oil, rise	100% (3/3)	100%	1.0
3	oil price => invasion	100% (3/3)	100%	1.0
4	price, rise => Ukraine, invasion	100% (3/3)	100%	1.0

5	invasion => oil price	100% (3/3)	100%	1.0
6	pump => oil price, oil, invasion Ukraine	100% (3/3)	100%	1.0
7	price, oil, invasion Ukraine => pump	100% (3/3)	100%	1.0
8	pump, rise => Ukraine, invasion	100% (3/3)	100%	1.0
9	price, pump => oil price, invasion Ukraine	100% (3/3)	100%	1.0
10	Ukraine, invasion Ukraine => pump	100% (3/3)	100%	1.0
11	pump => oil price, invasion	100% (3/3)	100%	1.0
12	invasion ukraine => oil price, pump, oil	100% (3/3)	100%	1.0
13	price, pump, oil => invasion ukraine	100% (3/3)	100%	1.0
14	ukraine, invasion ukraine => pump, oil	100% (3/3)	100%	1.0
15	pump, oil, rise => invasion	100% (3/3)	100%	1.0
16	ukraine, invasion => price, oil	100% (3/3)	100%	1.0
17	price, oil => ukraine, invasion	100% (3/3)	100%	1.0
18	invasion ukraine => oil price, oil	100% (3/3)	100%	1.0
19	invasion => ukraine	100% (3/3)	100%	1.0
20	ukraine => invasion	100% (3/3)	100%	1.0
21	ukraine, invasion => price	100% (3/3)	100%	1.0
22	ukraine, invasion, invasion ukraine => price	100% (3/3)	100%	1.0
23	oil price, oil => invasion, invasion ukraine	100% (3/3)	100%	1.0

5.1.1.1 Discussion des résultats :

Règle (1…5) : ces règles d'association suggèrent une relation potentielle entre l'invasion de l'Ukraine et la hausse des prix du pétrole.

Règle (6…11) : Ces règles d'association suggèrent qu'il existe une forte relation entre les produits de la pompe (essence), les prix du pétrole et l'invasion de l'Ukraine.

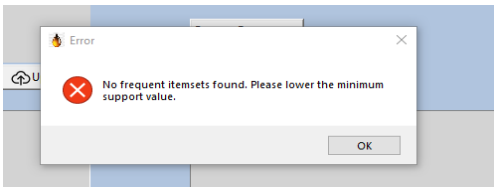
Cela suggérer que le conflit en Ukraine a un impact sur les marchés pétroliers mondiaux et affectent finalement le prix de l'essence à la pompe.

5.1.2 Résulta 2 de l'exemple 1

Support=100% (3/3), confiance=100%,

Algorithm : **Apriori with TF-IDF / FP-Growth with TF-IDF**

Percentagechosen = 0.5 (50%)



Avec un support minimum de 3/3 ou de 2/3 et un pourcentage choisi (n=50%), nous avons rencontré une erreur indiquant qu'il n'y avait pas d'ensembles d'items fréquents, donc il n'y a pas de termes dans les 50% choisis avec unsupport minimum de 2/3 ou plus élevée.

Support=33% (1/3), confiance=100%,

Algorithm : **Apriori with TF-IDF / FP-Growth with TF-IDF**

Percentagechosen = 0.5 (50%)

Itemset	Support
since, gasoline, day, time	33.33%(1/3)
something, gasoline, day, spike	33.33%(1/3)
day, follow, sense, retail	33.33%(1/3)
February, day, lag, yesterdays	33.33%(1/3)
diesel, gasoline, follow, yesterdays	33.33%(1/3)
fall, sense, daily, may	33.33%(1/3)

Id	La règle d'association	Support	Confiance	Lift
1	around, may => 110120, set	33.33% (1/3)	100%	3.000
2	gasoline, lag, also => yesterdays	33.33% (1/3)	100%	3.000
3	spike => gasoline, lag, also	33.33% (1/3)	100%	3.000
4	new, tension, cent => petrol	33.33% (1/3)	100%	3.000
5	week, professor, cent => new	33.33% (1/3)	100%	3.000
6	new, professor, cent => week	33.33% (1/3)	100%	3.000

5.1.2.1 Discussion des résultats :

En abaissant le seuil de support minimum, cela a pour conséquence de générer un plus grand nombre de règles, mais cela conduit également au fait que presque toutes les règles générées sont inutiles ou sans intérêt, ce qui ne permet pas d'obtenir des informations pertinentes et claires.

5.2 Exemple 2 (top factors that affect the price of Oil)

Text: 1

The supply factor is of paramount importance in oil markets, where any changes in supply levels can significantly affect oil prices. OPEC has traditionally used supply as a tool to control oil prices, but geopolitical events such as conflicts or political instability in major oil-producing countries may disrupt the supply chain and cause oil prices to fluctuate. Technological advancements, such as hydraulic fracturing or fracking, can also impact supply levels, which in turn affect the price of oil. Therefore, it is evident that supply remains one of the most critical factors that affect oil market dynamics.

Text: 2

The cost of oil is subject to various factors, among which demand plays a significant role. The level of demand for oil can change in response to fluctuations in global trade and economic activity, leading to shifts in the cost of oil. Growing economies may raise the cost due to an increase in demand, while a recession may cause it to fall. Government policies, geopolitical tensions, and technological advancements that reduce the need for oil can also affect the cost. Therefore, comprehending the intricacies of demand is crucial in anticipating and addressing shifts in the oil market.

Text: 3

Natural or man-made disasters can have a significant impact on petrol markets. Disruptions in supply due to hurricanes, earthquakes, fires, or geopolitical conflicts can lead to sudden price increases. For example, the disaster hurricanes in the Gulf of Mexico have disrupted offshore drilling and production in the past, causing supply shortages and price spikes. Similarly, the 2011 earthquake and tsunami in Japan caused a drop in demand for petrol as the country's economy and energy infrastructure were severely damaged. In contrast, man-made disasters such as wars or terrorist attacks can disrupt supply and cause prices to rise, as seen in the Persian Gulf Wars, the September 11 attacks and lately the Ukraine invasion.

5.2.1 Résultat 1 de l'exemple 2

Support=66% (2/3), confiance=100%,

Algorithm : **Apriori with Countvectorizer**

Itemset	Support
conflict	66.66% (2/3)
demand	66.66% (2/3)
disrupt supply	66.66% (2/3)
economy	66.66% (2/3)
geopolitical	100% (3/3)
impact	66.66% (2/3)
increase demand	66.66% (2/3)
oil market	66.66% (2/3)
price	66.66% (2/3)
supply	66.66% (2/3)

Id	La règle d'association	Support	Confiance	Lift
1	price, cause => supply	66.66% (2/3)	100%	1.50
2	cause, supply => price	66.66% (2/3)	100%	1.50
3	conflict => disrupt, disrupt supply	66.66% (2/3)	100%	1.50
4	disrupt, disrupt supply => conflict	66.66% (2/3)	100%	1.50
5	disrupt supply => geopolitical, cause	66.66% (2/3)	100%	1.00
6	disrupt supply, supply => price	66.66% (2/3)	100%	1.50
7	impact, market => disrupt supply	66.66% (2/3)	100%	1.50
8	demand => geopolitical, cause	66.66% (2/3)	100%	1.00
9	economy => increase, demand	66.66% (2/3)	100%	1.50
10	market, increase => demand	66.66% (2/3)	100%	1.50
11	increase, demand => market	66.66% (2/3)	100%	1.00
12	increase, demand => geopolitical	66.66% (2/3)	100%	1.00
13	demand => increase, geopolitical	66.66% (2/3)	100%	1.50
14	market, demand => geopolitical	66.66% (2/3)	100%	1.00
15	demand => geopolitical	66.66% (2/3)	100%	1.00
16	due, demand => economy, geopolitical	66.66% (2/3)	100%	1.50
17	economy, geopolitical => increase, demand	66.66% (2/3)	100%	1.50
18	market, increase => economy, geopolitical	66.66% (2/3)	100%	1.50

19	economy, geopolitical => market, increase	66.66% (2/3)	100%	1.50

5.2.1.1 Discussion des résultats :

Règle (1…7) : Ces règles d'association suggèrent que l'offre (supply) est Un facteur important, qui peut être influencée par différentes causes. Tel que les événements géopolitiques et les conflits peuvent entraîner des perturbations de l'offre, ce qui affecte à son tour les prix et le marché.

Règle (8…19) : Ces règles suggèrent qu'il existe des relations solides entre la demande, l'économie, le marché et les facteurs géopolitiques dans le contexte des prix du pétrole. Plus précisément, les règles indiquent qu'une augmentation de la demande peut être influencée par des événements géopolitiques et la croissance économique.

5.2.2 Résultat 2 de l'exemple 2

Support=100% (1/3), confiance=100%,

Algorithm Apriori**with TF-IDF / FP-Growth with TF-IDF**

Percentagechosen = 0.2 (20%)

Itemset	Support
supply, price	66.66% (2/3)
supply	66.66% (2/3)
oil	66.66% (2/3)
price	66.66% (2/3)
supply, affect, price	33.33% (1/3)
disaster, gulf, supply	33.33% (1/3)
shift, demand, cost	33.33% (1/3)
disaster, war, earthquake	33.33% (1/3)
disaster, manmade, petrol	33.33% (1/3)
disaster, manmade, price	33.33% (1/3)
oil, supply, opec	33.33% (1/3)
oil, opec, price	33.33% (1/3)

Id	La règle d'association	Support	Confiance	Lift
1	price => supply	66.66% (2/3)	100%	1.50

2	supply, war, hurricane => price	33.33% (1/3)	100%	1.50
3	oil, supply => affect, price	33.33% (1/3)	100%	3.00
4	disaster, supply, war => price	33.33% (1/3)	100%	1.50
5	hurricane, gulf, earthquake => supply	33.33% (1/3)	100%	1.50
6	disaster, gulf => price, supply	33.33% (1/3)	100%	1.50
7	disaster, war, attack => supply	33.33% (1/3)	100%	1.50
8	disaster, manmade, hurricane => supply	33.33% (1/3)	100%	1.50
9	oil, supply => opec, importance	33.33% (1/3)	100%	3.00
10	oil, supply => opec, affect	33.33% (1/3)	100%	3.00
11	oil, demand => cost, shift	33.33% (1/3)	100%	3.00
12	demand => oil, cost	33.33% (1/3)	100%	3.00
13	oil, cost => demand	33.33% (1/3)	100%	3.00
14	petrol => disaster, war	33.33% (1/3)	100%	3.00
15	petrol => disaster, manmade, hurricane	33.33% (1/3)	100%	3.00
16	disaster, manmade => price, petrol	33.33% (1/3)	100%	3.00
17	oil, price => opec, importance	33.33% (1/3)	100%	3.00
18	oil, price => opec, significantly	33.33% (1/3)	100%	3.00
19	oil, price => opec, affect	33.33% (1/3)	100%	3.0

5.2.2.1 Discussion des résultats :

Règle (1,2,3,4) : À partir de ces règles, nous pouvons conclure que l'offre (supply) est un facteur clé qui influence les prix du pétrole. De plus, la survenue d'événements tels que les guerres, les ouragans et les catastrophes peut affecter l'offre de pétrole et avoir un impact sur les prix.

Règle (5…10) : A partir de ces règles, nous pouvons conclure que les catastrophes naturelles telles que les ouragans et les tremblements de terre peuvent affecter l'approvisionnement en pétrole, en particulier dans la région du Golfe. De

plus, les catastrophes d'origine humaine et les guerres peuvent également avoir un impact sur l'approvisionnement en pétrole. Les règles suggèrent également que l'OPEC étant un facteur important qui affecte et contrôle l'approvisionnement en pétrole.

Règle (11,12,13) : Ces règles indiquent qu'il existe une relation entre le pétrole, la demande et le coût. Plus précisément, une augmentation de la demande de pétrole peut entraîner un changement des coûts.

Règle (14,15,16) : Ces règles suggèrent qu'il y a une relation entre le pétrole et les catastrophes, à la fois naturelles et causées par l'homme, ainsi que les guerres. De plus, la survenue de catastrophes causées par l'homme peut avoir un impact sur les prix du pétrole.

Règle (17,18,19) : Ces règles suggèrent qu'il existe une relation entre les prix du pétrole et l'OPEP. En particulier, l'OPEP est considérée comme un facteur important qui affecte significativement les prix du pétrole.

5.3 Comparaisons les résultats des deux exemples

Dans le premier exemple, les résultats obtenus par la méthode 1, qui utilise l'algorithme Apriori avec Countvectorizer suggèrent une relation potentielle entre l'invasion de l'Ukraine et les prix du pétrole, ainsi qu'une forte relation entre les prix de l'essence, les prix du pétrole et l'invasion de l'Ukraine. D'autre part, la deuxième méthode, qui utilise les deux algorithmes Apriori/FP-Growth avec TF-IDF, a conduit à des règles principalement non pertinentes ou non intéressantes, qui ne fournissent pas d'informations claires et pertinentes.

Dans le deuxième exemple, les résultats des deux méthodes fournissent des conclusions similaires sur les relations entre différents facteurs et leur impact sur l'offre, la demande et les prix du pétrole. Cependant, la deuxième méthode (Apriori/ FP-Growth avec TF-IDF) semble fournir des détails plus spécifiques sur les types de catastrophes et d'événements qui peuvent affecter l'offre et la demande de pétrole, et l'importance de l'OPEP en tant que facteur de contrôle des prix du pétrole. De plus, les deux algorithmes avec TF-IDF fournissent un langage plus concis et clair dans leurs conclusions.

Il est important de noter que les deux algorithmes de la méthode 2 ont donné les mêmes résultats dans les deux exemples.

En général, l'algorithme Apriori avec Countvectorizer était plus efficace dans le premier exemple, tandis qu'Apriori/ FP-Growth avec TF-IDF était plus efficace dans le deuxième exemple

5.4 Discussion générale de résultats

Dans le premier exemple, l'algorithme d'Apriori avec Countvectorizer et sans TF-IDF était plus efficace car tous les documents contenaient les mêmes mots-clés liés à l'invasion de l'Ukraine. Par conséquent, l'algorithme a pu facilement trouver des ensembles d'articles fréquents sans avoir besoin de TF-IDF. Cependant, dans le deuxième exemple, chaque document discutait d'un facteur différent affectant les prix du pétrole, avec des mots-clés uniques qui n'étaient pas présents dans les autres documents. Cela rendait nécessaire l'utilisation d'Apriori avec TF-IDF ou FP-Growth avec TF-IDF pour identifier les mots-clés les plus importants et informatifs qui distinguaient chaque document des autres et qui capturaient les aspects uniques de chaque facteur affectant les prix du pétrole.

TF-IDF fonctionne en sélectionnant les mots-clés les plus importants selon leur score, qui est plus élevé pour les mots-clés informatifs et distinctifs. Cependant, lorsque tous les documents contiennent un terme, le score IDF devient 0, ce qui signifie que le score TF-IDF pour ce terme sera également 0, le rendant peu important dans l'analyse. C'est pourquoi les deux algorithmes avec TF-IDF n’ont pas bien fonctionné dans l'exemple 1 car tous les documents contenaient les mêmes mots-clés liés à l'invasion de l'Ukraine, et donc, la plupart des mots-clés avaient des scores IDF de 0. En revanche, dans l'exemple 2, la méthode TF-IDF était efficace car chaque document contenait son propre ensemble unique de mots-clés importants pour capturer les facteurs spécifiques affectant les prix du pétrole.

En ce qui concerne les différences entre les deux algorithmes de la méthode 2 (Apriori avec TF-IDF et FP-Growth avec TF-IDF), il convient de noter qu'ils donnent toujours les mêmes résultats en termes de règles d'association extraites. La différence réside principalement dans la manière dont les itemsets fréquents sont extraits. FP-Growth analyse les données seulement deux fois, tandis qu'Apriori, pour chaque taille d'itemset fréquent, analyse les données pour générer les candidats. En conséquence,

FP-Growth est généralement plus rapide. Cependant, dans notre cas, la différence de performance entre les deux algorithmes n'est pas perceptible car la taille des données n'est pas suffisamment grande pour mettre en évidence cette différence de vitesse.

Conclusion

Le chapitre compare les méthodes Apriori avec CountVectorizer et Apriori/FP Growth avec TF-IDF et évalue l'impact de chaque technique sur la qualité des règles d'association extraites des données.

Conclusion générale et les perspectives

1. Travaux réalisé

En conclusion, ce mémoire s'est concentrée sur l'extraction des règles d'association à partir de données textuelles en utilisant l'algorithme Apriori et une combinaison de TF-IDF avec l'algorithme Apriori et l'algorithme FP-Growth. Cette mémoire a permis d'obtenir des informations précieuses sur l'efficacité de ces méthodes dans différents scénarios.

L'algorithme d'Apriori sans TF-IDF s'est avéré efficace lorsque tous les documents contenaient les mêmes mots-clés ou les mêmes termes. Cependant, lorsque l'on traite de différents facteurs ou aspects dans les documents, l'intégration de TF-IDF avec les algorithmes d'extraction de règles d'association a donné des résultats prometteurs.

2. Perspectives

Malheureusement, les contraintes de temps ont limité la possibilité d'explorer et d'étudier plus en profondeur d'autres approches et algorithmes. Néanmoins, il existe des perspectives potentielles pour de futures recherches dans ce domaine. Celles-ci comprennent :

1. La combinaison des autres techniques d'exploration de texte, telles que : wordembeddings ou topic modeling, avec l'extraction de règles d'association, y compris l'extraction des règles avec d'autres algorithmes tels que ECLAT, afin d'améliorer l'extraction de règles significatives.

2. Utiliser les règles d'association comme étiquettes dans une classification supervisée pour former nos modèles. Cette approche permettrait d'exploiter pleinement les informations contenues dans les règles d'association extraites, en les utilisant comme des indicateurs pour l'entraînement des modèles de classification. Cela pourrait conduire à une meilleure performance des modèles en exploitant les relations et les dépendances entre les termes dans les données textuelles.

En conclusion, bien que les contraintes de temps aient restreint la portée de ce travail, les résultats contribuent à la compréhension et à l'application de l'extraction de

règles d'association à partir de données textuelles. Les futures recherches peuvent s'appuyer sur ces bases en explorant des méthodes et des domaines alternatifs, ce qui permettra de faire progresser le domaine de l'exploration de texte et de l'extraction de règles d'association.

Références

[1] M. NEMICHE, “Data mining,” Master, Faculté des Sciences d’Agadir , Morocco, 2015.

[2] I. H. Witten and E. Frank, “Data mining: practical machine learning tools and techniques with Java implementations,” *Acm Sigmod Record*, vol. 31, no. 1, pp. 76–77, 2002.

[3] A. Samia and L. Arezki, “Extraction de règles d’associations entre les concepts biomédicaux.,” Université Mouloud Mammeri, 2013.

[4] B. Bigi, “TALN Informatique,” Apr. 2006.

[5] S. Raheel, “L’apprentissage artificiel pour la fouille de données multilingues: application à la classification automatique des documents arabes,” Lyon 2, 2010.

[6] S. Raschka, “Naive bayes and text classification i-introduction and theory,” *arXiv preprint arXiv:1410.5329*, 2014.

[7] R. Agrawal, T. Imieliński, and A. Swami, “Mining association rules between sets of items in large databases,” in *Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, 1993, pp. 207–216.

[8] F. HAMZA CHERIF, “Etude et réalisation d’une plateforme dédiée à la télésurveillance de l’état physiopathologique d’une personne au volant,” 2020.

[9] S. Brin, R. Motwani, and C. Silverstein, “Beyond market baskets: Generalizing association rules to correlations,” in *Proceedings of the 1997 ACM SIGMOD international conference on Management of data*, 1997, pp. 265–276.

[10] S. H. Guillaume, “Traitement des données volumineuses. Mesures et algorithmes d’extraction de règles d’association et règles ordinales,” Nantes, 2000.

[11] R. Mishra and A. Choubey, “Comparative analysis of apriori algorithm and frequent pattern algorithm for frequent pattern mining in web log data,” *IJCSIT) International Journal of Computer Science and Information Technologies*, vol. 3, no. 4, pp. 4662–4665, 2012.

[12] J. Han, J. Pei, and Y. Yin, “Mining frequent patterns without candidate generation,” *ACM sigmod record*, vol. 29, no. 2, pp. 1–12, 2000.

[13] M. J. Zaki, S. Parthasarathy, M. Ogihara, and W. Li, “Parallel algorithms for discovery of association rules,” *Data mining and knowledge discovery*, vol. 1, pp. 343–373, 1997.

[14] M. J. Zaki, “Scalable algorithms for association mining,” *IEEE transactions on knowledge and data engineering*, vol. 12, no. 3, pp. 372–390, 2000.

[15] PimpriChinchwad College of Engineering , Akurdi, Pune, Pune University., S. S. Kadam, and S. S.Deshmukh, “Eclat Algorithm for FIM on CPU-GPU co-operative & parallel environment,” *IOSRJCE*, vol. 16, no. 2, pp. 88–96, 2014, doi: 10.9790/0661-16288896.

[16] U. Y. Nahm and R. J. Mooney, “Mining soft-matching rules from textual data,” in *Proceedings of the 17th international joint conference on Artificial intelligence-Volume 2*, 2001, pp. 979–984.

[17] H. Cherfi, A. Napoli, and Y. Toussaint, “Vers une méthodologie de fouille de textes s’ appuyant sur l’extraction de motifs fréquents et de règles d’association,” in *Conférence d’Apprentissage (CAp’03), dans le cadre de la plate-forme (AFIA’03)*, Presses universitaires de Grenoble, 2003, pp. 61–76.

[18] “PyCharm — Wikipédia,” *PyCharm*. https://fr.wikipedia.org/wiki/PyCharm (accessed May 13, 2023).

[19] “Natural Language Toolkit — Wikipédia.” https://fr.wikipedia.org/wiki/Natural_Language_Toolkit (accessed May 13, 2023).

[20] “Scikit-learn — Wikipédia.” https://fr.wikipedia.org/wiki/Scikit-learn (accessed May 13, 2023).

[21] “mlxtend.” http://rasbt.github.io/mlxtend/ (accessed May 13, 2023).

[22] “Pandas : la bibliothèque Python dédiée à la Data Science.” https://datascientest.com/pandas-python-data-science (accessed May 27, 2023).

[23] “Python Imaging Library (PIL) — Documentation Bibliothèques Python 1.0.0.” https://he-arc.github.io/livre-python/pillow/index.html (accessed May 14, 2023).

[24] “Les applications graphiques avec Tkinter — Python 3.X.” https://gayerie.dev/docs/python/python3/tkinter.html (accessed May 14, 2023).

Annexe A

Prétraiter les données

A.1. Lemmatiser les données

```
lemmatizer = WordNetLemmatizer()

def lemmatize_text(text):
    words = text.split()
    lemmatized_words = []
for word, pos in nltk.pos_tag(words):
if pos.startswith("V"):
            lemma = lemmatizer.lemmatize(word, pos="v")
else:
            lemma = lemmatizer.lemmatize(word)
        lemmatized_words.append(lemma.strip())
return ' '.join(lemmatized_words)
```

A.2.Prétraiter les données

```
def preprocess_docs(documents):
# Initialize lemmatizer
lemmatizer= WordNetLemmatizer()

    all_tokens = []
for doc in documents:
# Convert to lowercase
doc = doc.lower()
# Remove punctuation
doc = re.sub(r'[^\w\s]', '', doc)
# Tokenize
tokens = nltk.word_tokenize(doc)
# Remove stop words
stop_words = set(stopwords.words('english'))
        tokens = [token for token in tokens if token not in
stop_words]
# Convert to string
doc_string = ' '.join(tokens)

# Lemmatize by using the def "lemmatize_text"
lem_tokens = lemmatize_text(doc_string)
# Append to all_tokens
all_tokens.append(lem_tokens)
print(all_tokens)

return all_tokens

# Preprocess the documents
all_tokens=preprocess_docs(contents)
```

Annexe B

Calculer le TF-IDF score

B.1. Calculer TF

```
def tf(docs_cleaned):
    tfscore = {}

for documentName, document in enumerate(docs_cleaned):
        wordInfoDictionary = {}
        doc = (document.split())
        wordCounter = Counter(doc)
#print((wordCounter))
lengthofDocument = len(doc)

if lengthofDocument != 0:
for word, repetition in wordCounter.items():
                frequency = repetition / lengthofDocument
                wordInformation = [frequency]
                wordInfoDictionary[word] = wordInformation

            tfscore[documentName] = wordInfoDictionary

return tfscore

#calculate tf score
tfscore=tf(all_tokens)
```

B.2. Calculer IDF

```
def idf(documents):
    idf = {}
    listOfUniqueWords=set(word for doc in documents for word in
doc.split())
    numberOfDocuments = len(documents)
for word in listOfUniqueWords:
        count = 0
for document in documents:
if word in document:
                count += 1
idf[word] = math.log10(numberOfDocuments / (count))
return idf

#calculate idf score
idfscore=idf(all_tokens)
```

B.3. Calculer TF-IDF

```
def tfIdf(tf, idf):
    tfIdfReturnDictionary = {}
    listForWordAndValue = []
    mainDocumentList = []

for documentName in tf:
        wordsInfo = tf[documentName]
if len(wordsInfo) != 0:
for word, wordTf in wordsInfo.items():
                tfIdfValue = wordTf[0] * idf[word]
                listForWordAndValue.append(word)
                listForWordAndValue.append(tfIdfValue)
                mainDocumentList.append(listForWordAndValue)
                listForWordAndValue = []
            mainDocumentList = sorted(mainDocumentList, key=lambda
x: x[1], reverse=True)
            tfIdfReturnDictionary[documentName] = mainDocumentList
            mainDocumentList = []
print(len(tfIdfReturnDictionary))
print(tfIdfReturnDictionary)
return tfIdfReturnDictionary


#calculate tf_idf score
tfidfscore=tfIdf(tfscore,idfscore)
```

Annexe C

Créer le term-document matrix de méthode 02

« avec TF-IDF »

C.1. Créer le term-document matrix

```
def createTermDocMatrix(tf_idf, percentage):
# Create a list of documents containing the top percentage of terms
for each document
documents = []
for documentName, values in tf_idf.items():
        num_terms = len(values)
        topN = int(num_terms * percentage)
        doc = [val[0] for val in values[:topN]]
        documents.append(doc)
# Create a set of unique terms
print(documents)
    terms = set()
for doc in documents:
        terms.update(doc)

# Create a dictionary to map terms to column indices
term_index = {term: i for i, term in enumerate(sorted(terms))}

# Create the term-document matrix
matrix = []
for doc in documents:
        row = [0] * len(terms)
for term in doc:
            row[term_index[term]] = 1
matrix.append(row)
# Convert the matrix and list of terms to a DataFrame
df = pd.DataFrame(matrix, columns=sorted(terms))



# Return the matrix
return df

# Create the document-term matrix
matrix_2 = createTermDocMatrix(tfidfscore,percentage)
```

Annexe D

Appliquer l'algorithme Apriori avec TF-IDF

D.1. Apriori avec TF-IDF

```
def find_associations_rules_using_apriori_tf_idf(df, min_support,
min_confidence,sort_by,max_len):
# Find frequent itemsets using Apriori algorithm
frequent_itemsets = apriori(df, min_support=min_support,
use_colnames=True, max_len=max_len)
# Sort the frequent itemsets by support in ascending order
frequent_itemsets = frequent_itemsets.sort_values('support',
ascending=False)
if frequent_itemsets.empty:
        messagebox.showerror("Error", "No frequent itemsets found.
Please lower the minimum support value.")
return None, None
# Generate association rules from the frequent itemsets
rules = association_rules(frequent_itemsets, metric='confidence',
min_threshold=min_confidence)

# Sort the rules based on the specified metric
if sort_by == 'confidence':
        rules = rules.sort_values('confidence', ascending=False)
elif sort_by == 'support':
        rules = rules.sort_values('support', ascending=False)
elif sort_by == 'lift':
        rules = rules.sort_values('lift', ascending=False)
elif sort_by == 'leverage':
        rules = rules.sort_values('leverage', ascending=False)


# Return the association rules as a pandas DataFrame
return frequent_itemsets, rules

# Extract and generate association rules
frequent_itemsets, associations_rules =
find_associations_rules_using_apriori_tf_idf(matrix_2,
min_support=float(minsupp_entry.get())/100,
min_confidence=float(minconf_entry.get())/100,
sort_by=met_sorted_by,max_len=max_length)
```