République Algérienne Démocratique et Populaire

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université des Sciences et de la Technologie Houari Boumediene

Faculté d'Electronique et Informatique

Département Informatique

Rapport du Projet : Partie 3

Data Mining

Extraction de motifs fréquents, règles d'associations et corrélations

Rédigé par

GUELLATI Mehdi Anis

ZAIT Fouad

Enseignant

HOUACINE Naila

Année universitaire :2022/2023

Sommaire

Introduction

L'exploration des motifs fréquents conduit à la découverte d'associations et de corrélations entre les éléments d'un dataset transactionnel ou relationnel. Avec des quantités massives de données collectées et stockées en permanence, de nombreuses industries s'intéressent à l'extraction de ces modèles dans leurs bases de données. La découverte de relations de corrélation intéressantes entre d'énormes quantités d'enregistrements de transactions commerciales peut aider dans de nombreux processus décisionnels commerciaux tels que la conception de catalogues, le marketing croisé et l'analyse du comportement d'achat ou de consommation des clients.

Dans cette troisième partie du projet, nous allons procéder à l'analyse, le prétraitement des données et l'extraction des motifs fréquents, des règles d'associations et de corrélation du dataset 2.

Cette étude va nous permettre d'analyser la consommation d'un ensemble d'individus en termes de contenu YouTube. Les règles d'associations et de corrélation qu'on va extraire nous permettront d'établir un système de recommandation de vidéos YouTube aux différents utilisateurs.

A. Jeux de données

a. Analyse du nouveau jeu de données dataset 2

1-Visualisation et description du Dataset et Description des attributs

Le dataset qu'on va étudier dans cette partie contient 4 attributs qui sont 'Watcher', 'VideoCategoryId', 'VideoCategorieLabel' et 'Definition' et 115 instances.

Watcher : Il est de type string et il ne contient pas de valeurs null.
VideoCategoryId : Il est de type int et il ne contient pas de valeurs null.
VideoCategoryLabel : Il est de type string et il ne contient pas de valeurs null.
Definition : Il est de type string et il contient des valeurs null.

2-Moyenne, Mode, Medianne de nos attributs

Attributs	Moyenne	Médiane	Mode
Watcher	/	/	Jeff
videoCategorieId	24.41	24	22
Definition	/	/	hd

On peut déduire la symétrie que de l'attribut videoCategorieId, on remarque que cet attribut est asymétrique à droite.

3-Mesures de dispersions de L'attribut videoCategorieId

Min : 1
Q1 : 22
Q2 : 22
Q3 : 28
Max : 29

On remarque de ces résultats que cet attribut a des valeurs entre 1 et 29 et qu'il a au moins un outliers en bas qui est le 1.

4-Les valeurs aberrantes

L'attribut videoCategoryId possède 2 outliers en bas qui sont 10 et 1, On ne remplacera pas ces valeurs par une autre, car ce sont deux catégories différentes et on ne les supprimera pas car elles sont catégoriques et donc elles ne fausseront pas nos résultats.

5-Histogrammes

-Watcher :

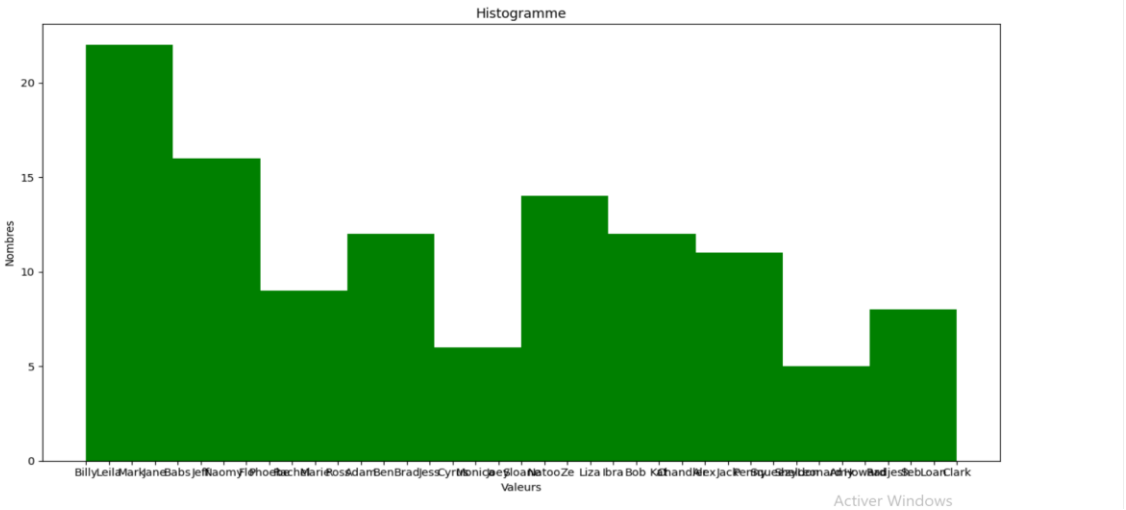

Figure 1 : Histogramme de l'attribut Watcher

On peut déduire de cet histogramme que Billy et Leila sont les utilisateurs apparaissant le plus dans ce dataset.

-**VideoCategoryId** :

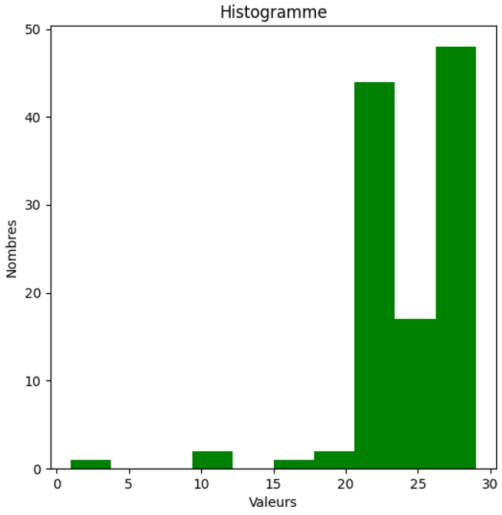

Figure 2 : Histogramme de l'attribut videoCategoryId

On remarque de cet histogramme que la plupart des valeurs sont entre 15 et 29 et que les valeurs qui apparaissent le plus sont : 22 et 28 ,29.

-VideoCategoryLabel :

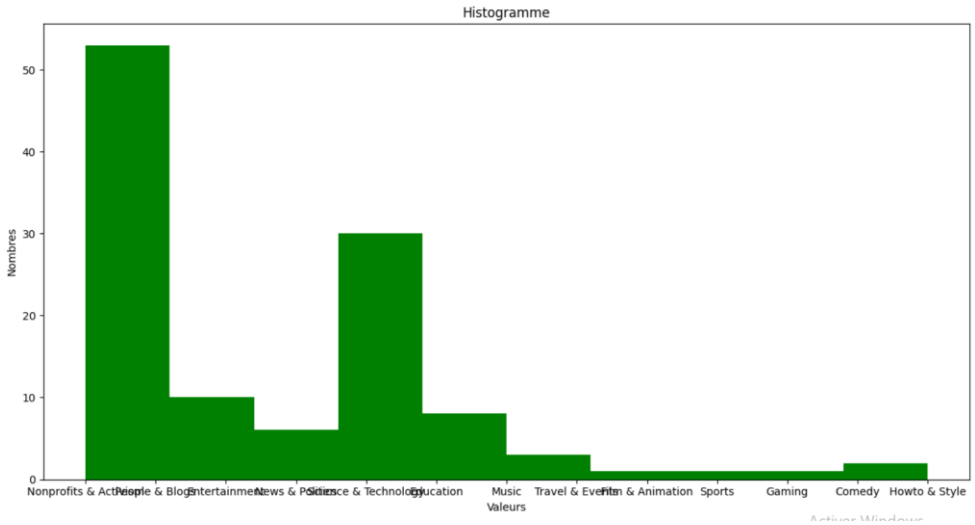

Figure 3 : Histogramme de l'attribut VideoCategoryLabel

Comme l'attribut précèdent les labels qui apparaissent le plus dans cette colonne sont Nonprofits & Activism, People & Blogs et Science & Technologie.

-Definition :

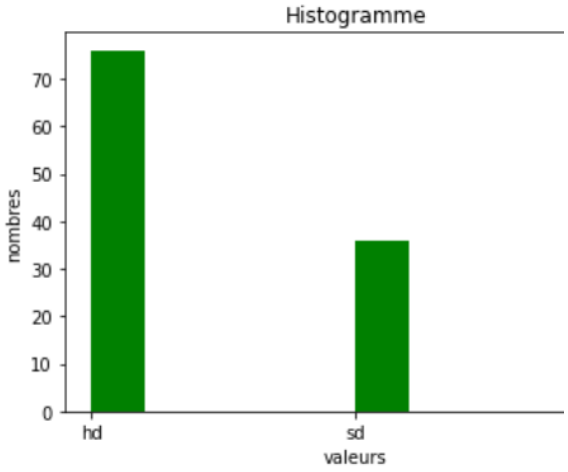

Figure 4 : Histogramme de l'attribut definition

On remarque que la plus part des videos visionnées par les utilisateurs sont en hd.

6-Boite a moustache

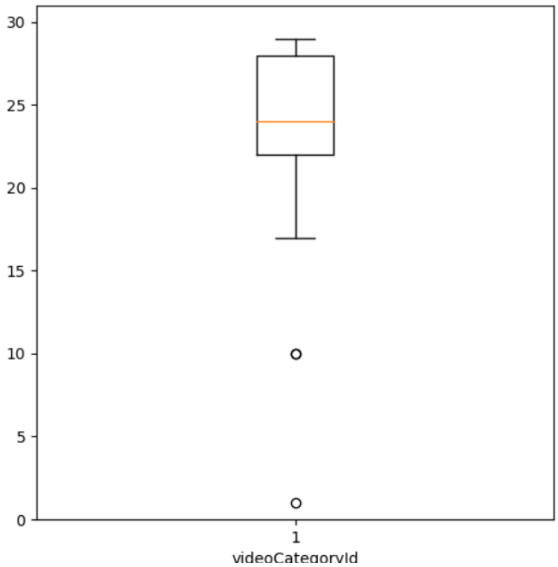

Figure 5 : Boite à moustache de l'attribut VideoCategoryId

A partir de cette boite a moustache on peut voir clairement 2 outliers en bas 1 et 10 pour l'attribut VideoCategoryId.

7-Diagramme de dispersion des données

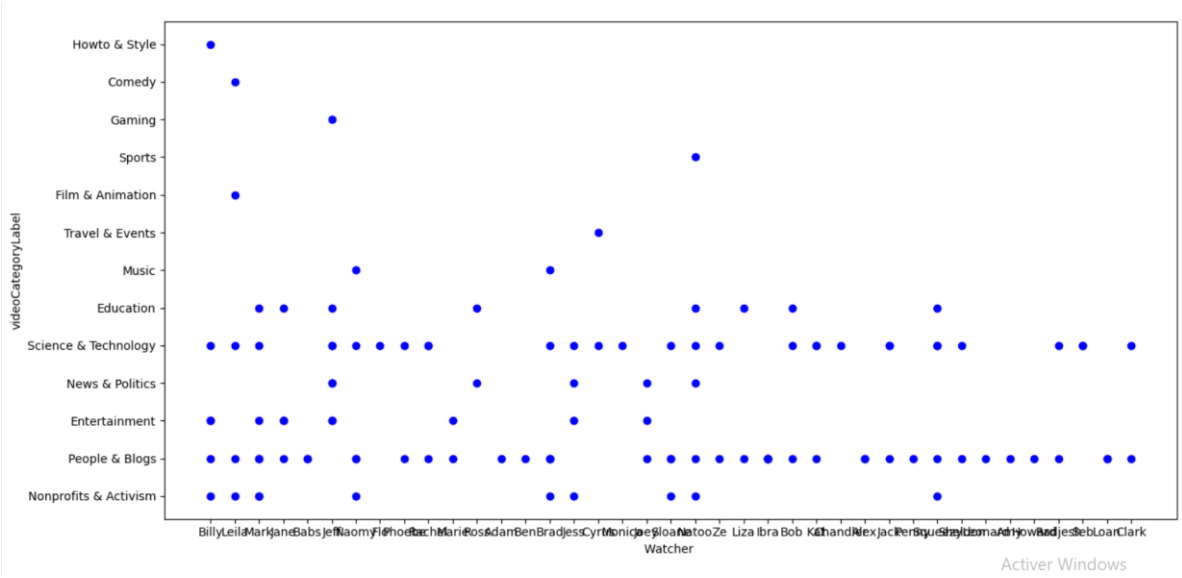

Figure 6 : Diagramme de dispersion entre l'attribut Watcher et l'attribut videoCategoryLabel

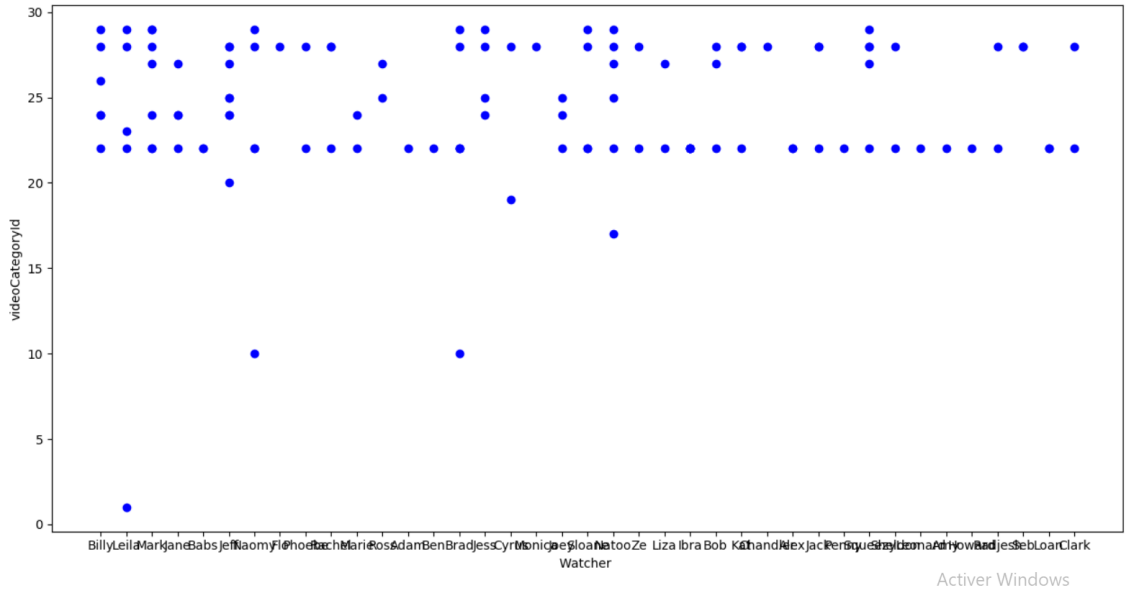

Figure 7 : Diagramme de dispersion entre l'attribut Watcher et l'attribut videoCategoryId

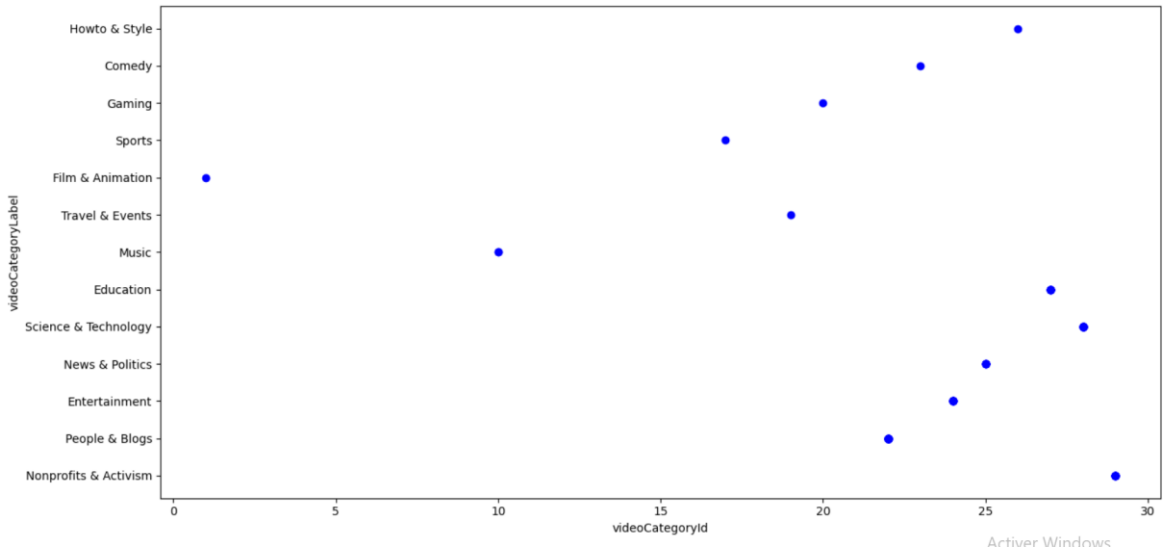

Figure 8 : Diagramme de dispersion entre l'attribut videoCategoryLabel et l'attribut videoCategoryId

D'après ces diagrammes on peut déduire qu'on peut travailler uniquement avec les attributs Watcher et VideoCategorylabel ou videocategorylabel et définition (on supprime un des deux (VideoCategorylabel ou videocategorylabel).

b. Appliquer un prétraitement adapté aux données manipulées

-Le seul attribut contentant des valeurs manquantes est l'attribut definition, on a remplacé ces valeurs manquantes par le mode ' hd '.
-Codification des valeurs de définition (hd en 1 et sd en 0).
-Reduction des valeurs redondantes (les doublons), on a 13 doublons dans notre dataset, après suppression des doublons notre dataset contiendra 102 instances.

-On a adapté notre dataset pour le problème des modèles fréquents, pour ceci nous avons pris en considération uniquement les deux premiers attributs (Watcher et videoCategoryId). L'attribut Watcher détermine les transactions et l'attribut videoCategoryId représente les items. On a donc regroupé pour chaque valeur de l'attribut watcher toutes les valeurs de l'attribut videoCategoryId qui lui correspondent, ceci afin d'obtenir une table de transaction comme suit :

	Itemset
Billy	[29.0, 22.0, 24.0, 28.0, 26.0]
Leila	[22.0, 28.0, 29.0, 1.0, 23.0]
Mark	[24.0, 22.0, 29.0, 27.0, 28.0]
Jane	[24.0, 27.0, 22.0]
Babs	[22.0]
Jeff	[25.0, 28.0, 24.0, 27.0, 20.0]
Naomy	[22.0, 10.0, 28.0, 29.0]
Flo	[28.0]
Phoebe	[28.0, 22.0]
Rachel	[22.0, 28.0]
Marie	[24.0, 22.0]
Ross	[25.0, 27.0]

Figure 9 : Une partie du dataset après traitement

B. Application de l'algorithme apriori

L'algorithme Apriori fait partie d'une famille d'algorithmes (ECLAT, FP-Growth…) qui ont pour but l'extraction des associations entre des entités dans un ensemble de données. Leur objectif principal est de déterminer des relations cachées dans des grandes bases de données afin de pouvoir émettre des prédictions par la suite.

Apriori est appliqué sur un type de dataset nommé table de transaction, qui est une liste de transactions avec les items correspondant à chacune d'entres elles. Un paramètre indispensable doit être préciser pour cet algorithme, et c'est le support minimum.

Le support minimum est sous la forme d'un nombre (ou un pourcentage) qui détermine le minimum de fois ou un item doit apparaitre dans la table de transaction afin de le prendre en considération dans la liste des items fréquents.

L'algorithme crée une liste C1 de candidats contenant tous les items de la table associés à leurs fréquences, il réduit ensuite cette liste en enlevant les items avec une fréquence inférieure au MinSup (on nommera cette liste L1), et il réitère ces étapes en combinant entre les items pour créer les listes C2 et L2… jusqu'à ce qu'il ne puisse plus.

Voici le pseudo de l'algorithme

Algorithme Apriori

Entrée : T : Table de transactions (Matrice) ; MinSup: le support minimum

Sortie : L: l'ensemble des motifs fréquents;

Var

C1, C2, ... , Ck: Matrice des candidats (itemsets - support)

L, L1, L2, ... , Lk: Tableaux des motifs fréquents (itemsets)

Début

/*Construction des candidats C1 (1-itemsets)*/

Calculer pour chaque item le nombre d'apparitions (support);

/*Réduction des candidat C1 en liste de motifs fréquents L1 */

Pour chaque candidat c de C1 faire

Si support(c) ≥ MinSup Alors L1 ← L1 ∪ c ; FinSi;

Fait;

/*Réitération du processus avec les 2-itemsets, 3-itemsets, ... , k-itemsets*/

k ← 1 ;

Tant que Lk!= ϕ faire

L ← L Lk ∪ ; /*sauvegarde de l'ensemble des motifs fréquents*/

k ← k + 1;

/*Générer toutes les combinaisons possibles formant des k-itemsets*/

Générer les candidats Ck à partir de la jointure de Lk-1 * Lk-1

/*Extraction des k-itemsets fréquents : Lk à partir de Ck */

Pour chaque candidat c de Ck faire

Si support(c) ≥ MinSup Alors Lk ← Lk ∪ c ; FinSi;

Fait;

Fait;

retourner L;

Fin.

a. Extraction de règles d'association (support, confiance)

L'extraction des règles d'association se fait à partir des liste Li de motifs fréquents qui contiennent les items (ou bien ensembles d'items) et leurs fréquences.

Support :

Le support fait référence à la popularité d'un item dans la base de données, il peut être calculé en trouvant le nombre de transactions contenant un item particulier divisé par le nombre total de transactions. Il est calculé en utilisant la liste des motifs fréquents avec la formule suivante :

$$\text{Support (A)} = \frac{nombre\ de\ transactions\ qui\ contiennent\ l'item\ (A)}{le\ nombre\ totale\ des\ transactions}$$

Règles d'association :

Les règles d'associations sont formées à partir des ensembles d'item fréquents, de manière à créer toutes les règles possibles en associant les différents items de l'ensemble entre eux, elles se présente sous la forme suivante :

{Ensemble d'antécédents} -> {Ensemble de conséquents}

Par Example :

Si on a cet ensemble d'item fréquents

{22, 28, 29}

Les règles d'associations qu'on pourra formés sont les suivants :

{22} -> {28, 29}, {28} -> {22, 29}, {29} -> {28, 22},

{22, 28} -> {29}, {22, 29} -> {28}, {29, 28} -> {22}

Confiance :

La confiance fait référence à la probabilité qu'un item B soit également présent si l'item A est présent. Il peut être calculé en trouvant le nombre de transactions ou A et B apparaissent ensemble, divisé par le nombre total de transactions ou A est apparue. Mathématiquement, on utilise la formule suivante :

$$\text{Confiance (A} \rightarrow \text{B)} = \frac{nombre\ de\ transactions\ qui\ contiennent\ les\ items\ (A\ et\ B)}{nombre\ de\ transactions\ qui\ contiennent\ l'item\ (A)}$$

On a donc extrait toutes les règles d'associations des nos ensemble de motifs fréquents et on a calculé pour chacune d'entre elles son support et sa confiance.

b. Extraction de règles de corrélation (support, confiance, lift)

Règles de corrélation :

Les règles de corrélation sont les mêmes règles qu'on a extrait dans la question a, la seule différence est le paramètre en plus qu'on doit calculer et qui le Lift.

Lift (Augmentation) :

Lift (A->B) fait référence à l'augmentation du ratio d'apparition de B lorsque A apparait. Il peut être calculé en divisant la confiance de la règle d'association (A->B) par le support de B. La formule suivante le montre :

$$Lift(A \rightarrow B) = \frac{Confiance(A \rightarrow B)}{Support(B)}$$

c. Expérimenter quelques valeurs de Min_Supp et Min_Conf

Première expérimentation :

On a exécuté l'algorithme Apriori avec les valeurs MinSup = 3, MinConf =0.02 et on a obtenu les résultats suivants :

```
Les motifs fréquents :
L1 {29: 9, 22: 31, 24: 7, 28: 24, 27: 8, 25: 5}


L2 {(29, 22): 8, (29, 24): 3, (29, 28): 9, (29, 27): 3, (22, 24): 5, (22, 28): 17,
(22, 27): 6, (24, 28): 4, (24, 27): 3, (24, 25): 3, (28, 27): 5, (28, 25): 3, (27,
25): 3}


L3 {(29, 22, 28): 8, (29, 22, 27): 3, (29, 24, 28): 3, (29, 28, 27): 3, (22, 28, 27
): 4}


L4 {(29, 22, 28, 27): 3}


L {29: 9, 22: 31, 24: 7, 28: 24, 27: 8, 25: 5, (29, 22): 8, (29, 24): 3, (29, 28):
9, (29, 27): 3, (22, 24): 5, (22, 28): 17, (22, 27): 6, (24, 28): 4, (24, 27): 3, (
24, 25): 3, (28, 27): 5, (28, 25): 3, (27, 25): 3, (29, 22, 28): 8, (29, 22, 27): 3
, (29, 24, 28): 3, (29, 28, 27): 3, (22, 28, 27): 4, (29, 22, 28, 27): 3}
```

Règles d'association avec 1 conséquent:
{'29 -> 22': [0.205, 0.889, 1.118], '22 -> 29': [0.205, 0.258, 1.118], '29 -> 24': [0.077, 0.333, 1.857], '24 -> 29': [0.077, 0.429, 1.857], '29 -> 28': [0.231, 1.0, 1.625], '28 -> 29': [0.231, 0.375, 1.625], '29 -> 27': [0.077, 0.333, 1.625], '27 -> 29': [0.077, 0.375, 1.625], '22 -> 24': [0.128, 0.161, 0.899], '24 -> 22': [0.128, 0.714, 0.899], '22 -> 28': [0.436, 0.548, 0.891], '28 -> 22': [0.436, 0.708, 0.891], '22 -> 27': [0.154, 0.194, 0.944], '27 -> 22': [0.154, 0.75, 0.944], '24 -> 28': [0.103, 0.571, 0.929], '28 -> 24': [0.103, 0.167, 0.929], '24 -> 27': [0.077, 0.429, 2.089], '27 -> 24': [0.077, 0.375, 2.089], '24 -> 25': [0.077, 0.429, 3.343], '25 -> 24': [0.077, 0.6, 3.343], '28 -> 27': [0.128, 0.208, 1.016], '27 -> 28': [0.128, 0.625, 1.016], '28 -> 25': [0.077, 0.125, 0.975], '25 -> 28': [0.077, 0.6, 0.975], '27 -> 25': [0.077, 0.375, 2.925], '25 -> 27': [0.077, 0.6, 2.925]}

Règles d'association avec 2 conséquents:
{'29 -> 22, 28': [0.205, 0.889, 2.039], '22 -> 29, 28': [0.205, 0.258, 1.118], '28 -> 29, 22': [0.205, 0.333, 1.625], '29, 22 -> 28': [0.205, 1.0, 1.625], '29, 28 -> 22': [0.205, 0.889, 1.118], '22, 28 -> 29': [0.205, 0.471, 2.039], '29 -> 22, 27': [0.077, 0.333, 2.167], '22 -> 29, 27': [0.077, 0.097, 1.258], '27 -> 29, 22': [0.077, 0.375, 1.828], '29, 22 -> 27': [0.077, 0.375, 1.828], '29, 27 -> 22': [0.077, 1.0, 1.258], '22, 27 -> 29': [0.077, 0.5, 2.167], '29 -> 24, 28': [0.077, 0.333, 3.25], '24 -> 29, 28': [0.077, 0.429, 1.857], '28 -> 29, 24': [0.077, 0.125, 1.625], '29, 24 -> 28': [0.077, 1.0, 1.625], '29, 28 -> 24': [0.077, 0.333, 1.857], '24, 28 -> 29': [0.077, 0.75, 3.25], '29 -> 28, 27': [0.077, 0.333, 2.6], '28 -> 29, 27': [0.077, 0.125, 1.625], '27 -> 29, 28': [0.077, 0.375, 1.625], '29, 28 -> 27': [0.077, 0.333, 1.625], '29, 27 -> 28': [0.077, 1.0, 1.625], '28, 27 -> 29': [0.077, 0.6, 2.6], '22 -> 28, 27': [0.103, 0.129, 1.006], '28 -> 22, 27': [0.103, 0.167, 1.083], '27 -> 22, 28': [0.103, 0.5, 1.147], '22, 28 -> 27': [0.103, 0.235, 1.147], '22, 27 -> 28': [0.103, 0.667, 1.083], '28, 27 -> 22': [0.103, 0.8, 1.006]}

Règles d'association avec 3 conséquents:
{'29 -> 22, 28, 27': [0.077, 0.333, 3.25], '22 -> 29, 28, 27': [0.077, 0.097, 1.258], '28 -> 29, 22, 27': [0.077, 0.125, 1.625], '27 -> 29, 22, 28': [0.077, 0.375, 1.828], '29, 22 -> 28, 27': [0.077, 0.375, 2.925], '29, 28 -> 22, 27': [0.077, 0.333, 2.167], '29, 27 -> 22, 28': [0.077, 1.0, 2.294], '22, 28 -> 29, 27': [0.077, 0.176, 2.294], '22, 27 -> 29, 28': [0.077, 0.5, 2.167], '28, 27 -> 29, 22': [0.077, 0.6, 2.925], '29, 22, 28 -> 27': [0.077, 0.375, 1.828], '29, 22, 27 -> 28': [0.077, 1.0, 1.625], '29, 28, 27 -> 22': [0.077, 1.0, 1.258], '22, 28, 27 -> 29': [0.077, 0.75, 3.25]}

Deuxième expérimentation :

MinSup=9 et MinConf=0.01, et on a obtenu les résultats suivants :

```
Les motifs fréquents :
L1 {29: 9, 22: 31, 28: 24}


L2 {(29, 28): 9, (22, 28): 17}


L {29: 9, 22: 31, 28: 24, (29, 28): 9, (22, 28): 17}



Règles d'association avec 1 conséquent:
{'29 -> 28': [0.231, 1.0, 1.625], '28 -> 29': [0.231, 0.375, 1.625], '22 -> 28': [0
.436, 0.548, 0.891], '28 -> 22': [0.436, 0.708, 0.891]}
```

C. Options avancées de l'IHM

a. Exécution de la méthode de DataMining à appliquer (Apriori) et Choix du Min_Supp et Min_Conf.

On a rajouté dans l'interface graphique des champs textuels pout permettre à l'utilisateur de choisir la valeur du support minimum et celle de la confiance minimum avant de cliquer sur le buton afin de lancer l'exécution de l'algorithme Apriori avec les paramètres spécifiés, l'affichage des règles d'association est optionnel (un buton spécial pour cet effet).

b. Insertion des données sur un utilisateur et en déduire une recommandation

L'utilisateur pourra utiliser notre interface afin d'ajouter les informations sur un utilisateur (nom de l'utilisateur, et les ses catégories), le système déduira par la suite les catégories les plus susceptibles de lui plaire, en se basant sur les règles d'association extraient précédemment, et en les triant par rapport à la valeur de leurs confiances, car c'est le paramètre qui désigne la probabilité que l'utilisateur qui s'intéresse à la catégorie A s'intéressera à une autre catégorie B.

On a ajouté les informations sur un utilisateur avec les catégories 22 et 29, et on a obtenu les résultats suivants :


```
{'28': 1.0, '27': 0.375, ('28', '27'): 0.375, '24': 0.25, '10': 0.25, ('24', '28'):
0.25, ('28', '10'): 0.25, '26': 0.125, '1': 0.125, '23': 0.125, '25': 0.125, '17':
0.125, ('24', '26'): 0.125, ('24', '27'): 0.125, ('28', '26'): 0.125, ('28', '1'):
0.125, ('28', '23'): 0.125, ('28', '25'): 0.125, ('28', '17'): 0.125, ('1,', '23'):
0.125, ('27', '25'): 0.125, ('27', '17'): 0.125, ('25', '17'): 0.125, ('24', '28',
'26'): 0.125, ('24', '28', '27'): 0.125, ('28', '1,', '23'): 0.125, ('28', '27', '2
5'): 0.125, ('28', '27', '17'): 0.125, ('28', '25', '17'): 0.125, ('27', '25', '17'
): 0.125, ('28', '27', '25', '17'): 0.125}
```


Comme on peut le voir 100% des utilisateurs qui sont intéressé pas les catégories 22 et 28, sont intéressé par la catégorie 28. Cette catégorie est donc une très bonne recommandation pour l'utilisateur introduit.

Conclusion

Durant cette 3eme partie du projet d'exploration de données, nous avons procéder à toutes les étapes nécessaires pour arriver à extraire les items fréquents en appliquant l'algorithme Apriori allant de l'analyse de notre dataset, le prétraitement des données jusqu'à l'extraction des règles d'association et les règles de corrélation en appliquant Apriori. Cela nous a permis d'identifier l'ensemble d'éléments qui se produisent ensemble.

Cette étude nous a permis dans un premier temps d'analyser la consommation d'un ensemble d'individus en termes de contenu YouTube ensuite nous avons pu établir un système de recommandation de vidéos YouTube aux différents utilisateurs grâces aux règles d'association que l'on a extraites.