

SOMMAIRE

Introduction

1 Architecture du système

1.1 Le module de Prétraitement

1.1.1 Base de données de GroupLens

1.1.2 Prétraitement

1.1.3 La base de données après le prétraitement

1.2 Module de filtrage et contextualisation

1.2.1 Les arbres de décision

1.2.2 Similarité User- User

1.2.3 Similarité Item-Item

1.3 Le module de recommandation

1.3.1 La recommandation cotée profil

1.3.2 La recommandation cotée Historique

2 Module d'évaluation de notre système

2.1 Les différents types d'évaluations

2.1.1 Evaluation du système de recommandation basé sur les arbres de décision basic

2.1.2 Evaluation du système de recommandation basé sur les arbres de décision et sur les contenus

2.1.3 Evaluation du système de recommandation basé sur les arbres de décision et sur la contextualisation

2.1.4 Evaluation du système de recommandation basé sur les arbres de décision basé sur contenus et sur la contextualisation

2.1.5 Système de recommandation basé sur l'historique d'un utilisateur et la contextualisation utilisons les arbres de décision

Conclusion

Introduction

Le but de notre travail est de concevoir un système, permettant de mieux connaître nos utilisateurs : leurs genres, leurs goûts et les utilisateurs qui partagent des informations similaires selon le temps pour recommander des films et chercher le meilleur titre à proposer lorsqu'un utilisateur se connecte, cette recommandation se fait selon l'historique ou le profil de l'utilisateur.

Pas de hasard sur notre système, il sait comment vous vous accrochez grâce à les différentes méthodes sophistiquées, permettant de classifier les Jours, calculer la similarité entre les utilisateurs et les films, et d'utiliser un classifieur appelé arbre de décision pour la classification et la prédiction.

1. Architecture du système

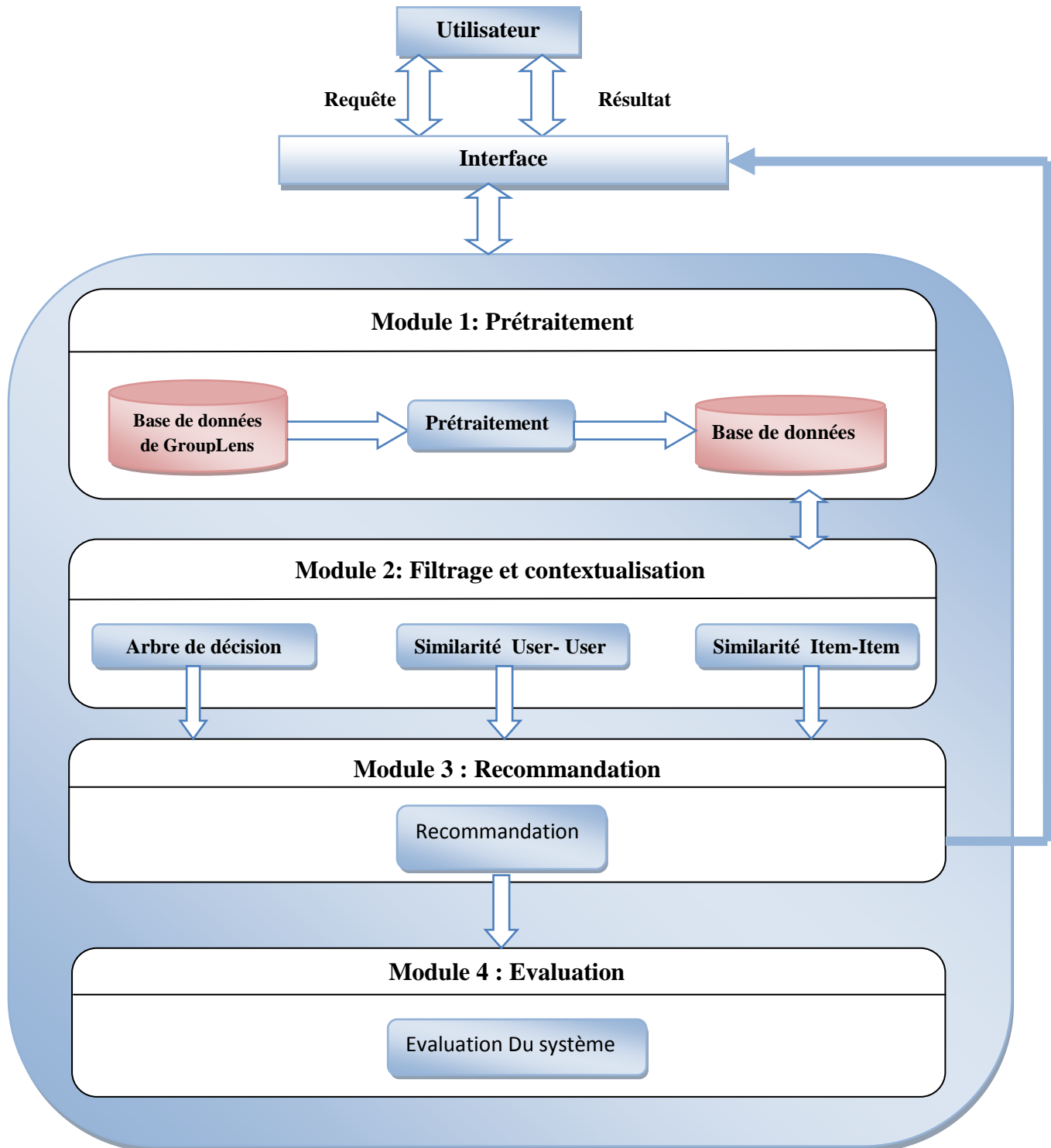


Figure 36 : Architecture du système.

L'architecture de notre système de recommandation comporte les modules suivants :

- **Un module de Prétraitement** : permettant d'effectuer un prétraitement sur notre base « GroupLens » pour obtenir une base donnée. On va l'utiliser dans le module suivant.
- **Un module de Recommandation** : permettant la recommandation des films à l'utilisateur selon son profil ou son historique
- **Un module d'Évaluation** : Utilisant ce module pour calculer le taux d'erreur de notre système.

1.1 Le module de Prétraitement :

1.1.1 Base de données de GroupLens :

MovieLens est un système de recommandation et de communauté virtuelle, c'est un site qui recommande des films pour ses utilisateurs, en fonction de leurs préférences de film créé en 1997 pour recueillir des données de recherche sur des recommandations personnalisées.

GroupLens recherche, est un laboratoire de recherche au département d'informatique et de génie de l'Université du Minnesota, qui a collecté une base de données « Rating data » à partir du site <http://movielens.org>. Cette collection a été effectuée dans des périodes de temps variées en fonction de sa taille par exemple :

- MovieLens 100k : un ensemble contient 100.000 notes de 1000 utilisateurs sur 1700 films. Sorti 4/1998.
- MovieLens 1M : un ensemble contient 1 millions notes de 6000 utilisateurs sur 4000 films. Sorti 2/2003.
- MovieLens 10M : un ensemble contient 10 millions notes et 100.000 applications d'étiquette sont appliquées à 10.000 films par 72.000 utilisateurs Sorti 1/2009.
- MovieLens 20M : un ensemble contient 20 millions notes et 465 applications d'étiquette sont appliquées à 27.000 films par 138.000 utilisateurs Sorti 4/2015.

Parmi ses derniers on a choisit la version MovieLens 100k car le temps d'exécution des requêtes par les utilisateurs est petit donc on n'aura pas besoin d'un processeur rapide pour la gérer et qui contient (cf. [figure 37](#)):

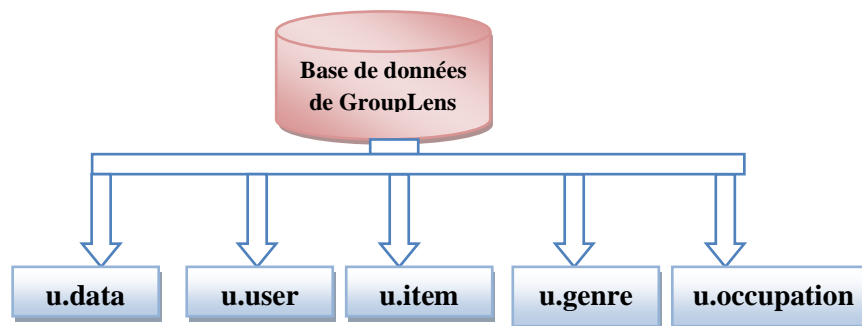


Figure 37 : Les composants de Base de données de GroupLens.

- **u.data** : C'est un ensemble de données se compose de : 10.0000 note par 943 utilisateur sur 1682 films. Chaque utilisateur a déjà noté au moins 20 films, L'ensemble est ordonné aléatoirement et contient ces colonnes (cf. [tableau 6](#)) et (cf. [figure 38](#)).

User id	Chaque utilisateur possède un seul identifiant
Iteme id	Chaque film possède un seul identifiant
Rating	La note donnée par l'utilisateur a cet item
Timestamp	Le temps de notation d'un film par un utilisateur en seconde depuis 1/1/1970 utc.

Tableau 6: Colonnes des données de u.data.

	IdUser	IdItem	Note	TimesTemp
1	196	242	3	881250949
2	186	302	3	891717742
3	22	377	1	878887116
4	244	51	2	880606923
5	166	346	1	886397596
6	298	474	4	884182806
7	115	265	2	881171488
8	253	465	5	891628467
9	305	451	3	886324817
10	6	86	3	883603013
11	62	257	2	879372434
12	286	1014	5	879781125
13				

Figure 38: Fichier u.data.

- **u.user** : sont les informations démographique des utilisateurs qui contient les colonnes suivants (cf. **tableau 7** et **figure 39**).

User id	Chaque utilisateur possède un seul identifiant
Age	L'âge de chaque utilisateur
Gender	Le sex de chaque utilisateur
Occupation	La profession de chaque utilisateur
Zip code	

Tableau 7 : Colonnes des données de u.user

1	IdUser	Age	Gender	Proffession	ZipCode
2	1	24	M	technician	85711
3	2	53	F	other	94043
4	3	23	M	writer	32067
5	4	24	M	technician	43537
6	5	33	F	other	15213

Figure 39: Le fichier u.user.

- **u.item** : sont les informations de chaque film, qui contient les colonnes suivants (cf. **tableau 8** et **figure 40**).

User id	Chaque film possède un seul identifiant
Movie title	Le titre de film
Release date	La date de film
video release date	La datefilm
IMDb URL	URL de film
Genres	Le genre de film

Tableau 8 : Colonnes des données de u.item

Chaque film appartient a plusieurs genres et le genre se compose de 19 colonne :| unknown | Action | Adventure | Animation | Children's | Comedy | Crime | Documentary | Drama | Fantasy | Film-Noir | Horror | Musical | Mystery | Romance | Sci-Fi |Thriller | War | Western.

1 : indique que le film appartient a ce genre et 0 indique le contraire par exemple :

01000000000000000000000011 = Action|War|Western

[illegible]

Figure 40: Fichier u.item.

- **u. occupation** : c'est la liste des professions (cf. **figure 41**).

1	administrator
2	artist
3	doctor
4	educator
5	engineer
6	entertainment
7	executive
8	healthcare
9	homemaker
10	lawyer
11	librarian
12	marketing
13	none
14	other
15	programmer
16	retired
17	salesman
18	scientist
19	student
20	technician
21	writer

Figure 41: Fichier u.occupation.

- **u. genres** : c'est la liste de genre que peut prendre un film (cf. **figure 42**).

1	unknown 0
2	Action 1
3	Adventure 2
4	Animation 3
5	Children's 4
6	Comedy 5
7	Crime 6
8	Documentary 7
9	Drama 8
10	Fantasy 9
11	Film-Noir 10
12	Horror 11
13	Musical 12
14	Mystery 13
15	Romance 14
16	Sci-Fi 15
17	Thriller 16
18	War 17
19	Western 18

Figure 42 : Fichier u.genre

1.1.2 Prétraitement :

On a effectué un prétraitement sur les ensembles des données suivantes :

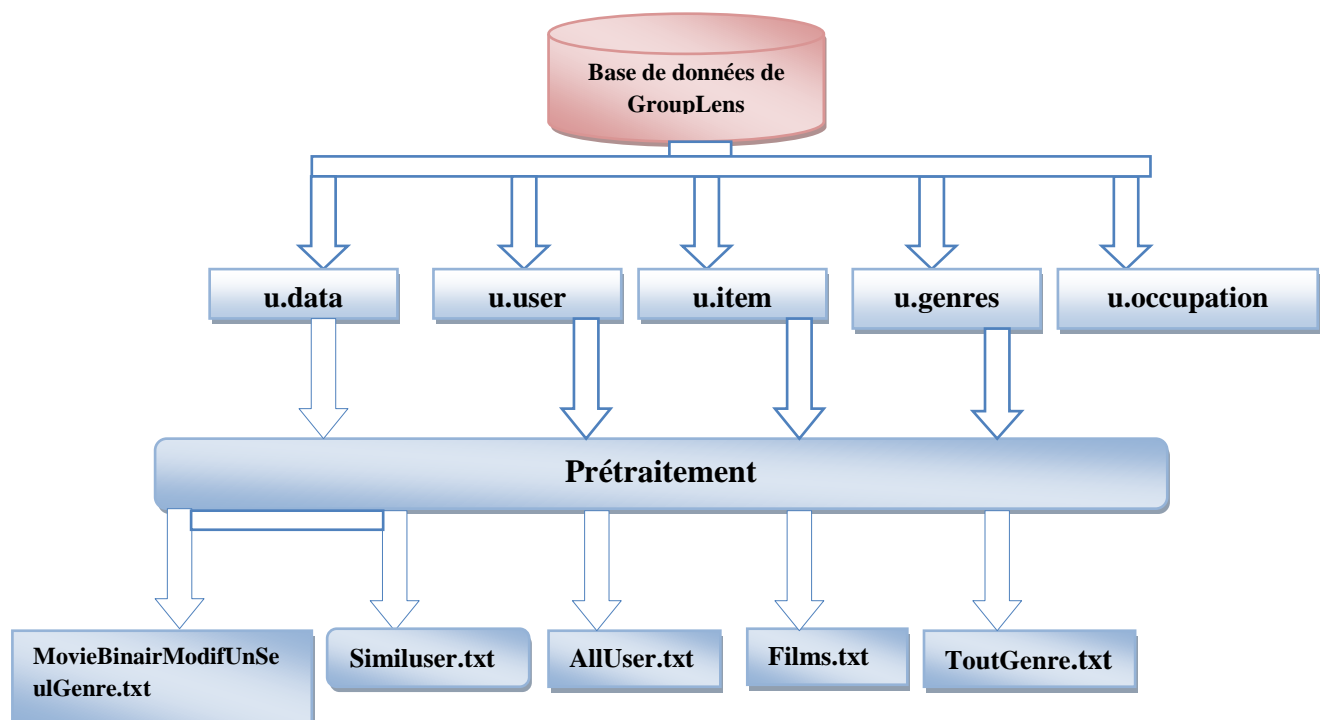


Figure 43 : Prétraitement sur les ensembles de Base de données de GroupLens.

• **MovieBinaireFinalModfUnSeulGenre.txt:**

Après un prétraitement effectué sur l'ensemble `u.data`, on a obtenu l'ensemble **MovieBinaireFinalModfUnSeulGenre** sous forme d'un fichier (.txt) (cf. **tableau 9** et **figure 44**).

User id	Age	12ans → 18ans == Adolescent 18ans → 30ans == Jeune Homme 30ans → 40ans == Homme 40ans → 70ans == Adulte 70ans → 80ans == Vieux
	Gender	Homme / Femme
	Profession	u.occupation
Item Id	Genre de film	u.Genre
Note	Mode binaire	Yes / No
Timestamp	Day	Samedi / Dimanche / Lundi / Mardi / Mercredi / Jeudi / Vendredi.
	KindOfDay	Vacance/ Normal
	Season	Hiver / Printemps / été / Automne
	Time	05h → 12h == Matin 12h → 15h == Après Midi 15h → 20h == Soir 20h → 05h ==Nuit

Tableau 9 : colonnes des données MovieBinaireFinalUnSeulGenre.txt

- ✓ **User Id** : on a utilisé les informations démographiques (Content-Based Chapitre 1) qui nous résoudre le problème de démarrage à froid donc d'après le profil d'un nouveau utilisateur on va recommander des films, mais si il est ancien dans notre système on peut utilisée sont Id.

On a divisé l'âge en :

Age : (12ans → 18ans == Adolescent, 18ans → 30ans == Jeune Homme, 30ans → 40ans == Homme, 40ans → 70ans == Adulte, 70ans → 80ans == Vieux)

- ✓ **Item id** : utilisant l'identifiant de film on obtient le Genre.

- ✓ **Rating** : on a changé le mode d'évaluation (étoile) en mode binaire (Yes /No) si l'utilisateur a aimé le film alors la note est varié entre 3 ,4 ou 5 dans ce cas la recommandation est effectué sinon (on va expliquer le but de ce prétraitement dans la partie d'évaluation).
- ✓ **Timestamp** : dans (le chapitre 2 contextualisation) on a évoqué l'utilisation des données temporelle qui sont importantes pour effectuer une recommandation fiable et qui satisfait le mieux possible notre utilisateur.

On a divisé et classifier le temps en :

- **Day** : (Samedi, Dimanche, Lundi, Mardi, Mercredi, Jeudi, Vendredi).
- **KindOfDay** : (Vacance, Normal).
- **Season** : (Hiver, Printemps, été, Automne).
- **Time**: (05h → 12h == Matin, 12h → 15h == Après Midi, 15h → 20h == Soir, 20h → 05h == Nuit).

1	Age	Gender	Proff	Genre	Note	Day	KindOfDay	Season	Time
2	Adulte	M	writer	Comedy	yes	jeudi	Normal	Hiver	ApréMidi
3	Homme	F	executive	Crime Film-Noir	yes	samedi	Vacance	Printemp	Soir
4	JeuneHomme	M	writer	Children's Comedy	no	vendredi	Vacance	Automne	Matin
5	JeuneHomme	M	technician	Drama	no	jeudi	Vacance	Automne	Matin

Figure 44: Fichier MovieBinairModif.txt

- **Similuser.txt:**

Après le prétraitement effectué sur l'ensemble u.data on a obtenu l'ensemble Similuser sous forme d'un fichier (.txt) (cf. **tableau 10** et **figure 45**).

Id user	Chaque utilisateur possède un seul identifiant
Id film	Chaque film possède un seul identifiant
Note	La note donnée par l'utilisateur a un film Yes / No
Type de jour	Vacance / Normal
Temps	05h → 12h == Matin 12h → 15h == Après Midi 15h → 20h == Soir 20h → 05h ==Nuit

Tableau 10: Colonnes des données Similuser.txt

Cet ensemble nous aide à calculer la similarité entre les utilisateurs, on le détaillera dans le module suivant.

1	IdUser	IdFilm	Note	TypeJr	Time
2	196	242	Yes	Normal	ApréMidi
3	186	302	Yes	Vacance	Soir
4	22	377	No	Vacance	Matin
5	244	51	No	Vacance	Matin
6	166	346	No	Normal	Matin
7	298	474	Yes	Normal	ApréMidi
8	115	265	No	Normal	ApréMidi

Figure 45: Fichier Similuser.txt

- **Alluser .txt :**

Après le prétraitement effectué sur l'ensemble u.user on a obtenu l'ensemble Alluser sous forme d'un fichier (.txt). le **tableau 11** illustre ce fichier.

User id	Chaque utilisateur possède un seul identifiant
Age	12ans → 18ans == Adolescent 18ans → 30ans == Jeune Homme 30ans → 40ans == Homme 40ans → 70ans == Adulte 70ans → 80ans == Vieux
Gender	Homme / Femme
Occupation	profession de chaque utilisateur

Tableau 11 : Colonnes des données Alluser .txt

- **Films.txt :**

Après le prétraitement effectué sur l'ensemble u.item on a obtenu l'ensemble Films sous forme d'un fichier (.txt) le **tableau 12** représente ce fichier.

Idfilm	Chaque film possède un seul identifiant
Titre	Le titre de filme
Genre	Chaine de caractère
Nombre de vue	Le nombre des utilisateurs qu'ont vu le film
Mieux noté	Le nombre des utilisateurs qu'ont aimés ce film

Tableau 12: Les colonnes des données Films.txt

- ✓ **Idfilm** : on a gardé l'identifiant du film.
- ✓ **Titre** : on a gardé le titre du film tel qu'il est.
- ✓ **Genre** : on a changé le genre de mode binaire (19 colonne) en chaîne de caractères, prenant en considération que les 1.
Exemple : 0|1|0|0|0|0|0|0|0|0|0|0|0|0|0|1|1| == Action
- ✓ **Nombre de vue** : si l'utilisateur a vu le film on rajoute +1. (qu'il ait aimé le film ou non).
- ✓ **Mieux noté** : si l'utilisateur a aimé le film on rajoute +1.

1	IdFilms	Titre	Genre	NbreVu	MieuxNoté
2	1/	Toy Story (1995)	/Animation Children's Comedy	/100001/	219
3	2/	GoldenEye (1995)	/Action Adventure Thriller	/100001/	57
4	3/	Four Rooms (1995)	/Thriller	/100001/	30
5	4/	Get Shorty (1995)	/Action Comedy Drama	/100001/	23
6	5/	Copycat (1995)	/Crime Drama Thriller	/100001/	111

Figure 46: Fichier films .txt

• ToutGenre.txt

Après le prétraitement effectué sur l'ensemble u.genre on a obtenu l'ensemble ToutGenre sous forme d'un fichier (.txt).

La seule différence entre u.genre et ToutGenre.txt, le premier ensemble contient tous les genres qu'un film peut les prendre, mais le second contient tous les genres de films quand peut trouver dans notre base de données (cf. [figure 47](#)).

```

1 Animation|Children's|Comedy|
2 Action|Adventure|Thriller|
3 Thriller|
4 Action|Comedy|Drama|
5 Crime|Drama|Thriller|
6 Drama|
7 Drama|Sci-Fi|
8 Children's|Comedy|Drama|
9 Drama|War|
10 Crime|Thriller|
11 Comedy|
12 Drama|Romance|
13 Comedy|Romance|
14 Action|Comedy|Crime|Horror|Thriller|
15 Action|Adventure|Comedy|Musical|Thriller|
16 Action|Drama|War|
17 Drama|Thriller|
18 Action|Adventure|Crime|
19 Action|
20 Action|Drama|Thriller|
21 Action|Adventure|Comedy|Crime|

```

Figure 47: Fichier ToutGenre.txt.

1.1.3 La base de données après le prétraitement:

Après avoir effectué le prétraitement sur l'ensemble des données de MovieLens (u.user, u.data, u.item, u.genre, u.occupation) ; on a obtenu des ensembles de données sous forme des fichiers.txt (MovieBinaireModif.txt ,Similuser.txt ,Alluser.txt ,Films.txt ,ToutGenre.txt) , on utilise ses dernier pour construire notre base de données.

1.2 Module de filtrage et contextualisation :

1.2.1 Les arbres de décision :

- **L'objectif de l'utilisation :**

On a utilisé les arbres de décision parce qu'ils sont plus lisible, il permet l'extraction des règles d'association et de prédire si un film recommandé satisfait l'utilisateur ou non dans un moment donné se basant sur les informations démographique de cet utilisateur et les données temporelles.

- **Création de l'arbre de décision :**

Dans le cas ou l'on effectue une recommandation basé sur le profil de l'utilisateur (nouveau, ancien) on crée un seul arbre de décision on utilisant l'ensemble des données MovieBinaireFinalModfUnSeulGenre.txt

D'après le chapitre 3, il existe plusieurs algorithmes pour la sélection d'attributs, et chaque algorithme a des avantages et des inconvénients selon les données traités ; donc on ne peut distinguer qui est le meilleur algorithme. Pour y remédier on a utilisé le logiciel SPSS qui inclue les différentes méthodes de splitage pour la création des arbres de décision, cela nous a permis de conclure que le meilleure algorithme est ID3 selon nos données.

En appliquant cette algorithme (ID3) sur MovieBinaireFinalModfUnSeulGenre.txt ou les entrées sont : l'age, Gender Profession, Genre de film, Day, KindOfDay, Season, Time et l'attribut cible que l'arbre doit prédie, c'est la note. Qui est basé sur l'entropie et le gain d'information. On obtient l'arbre illustré par la **figure 48**.

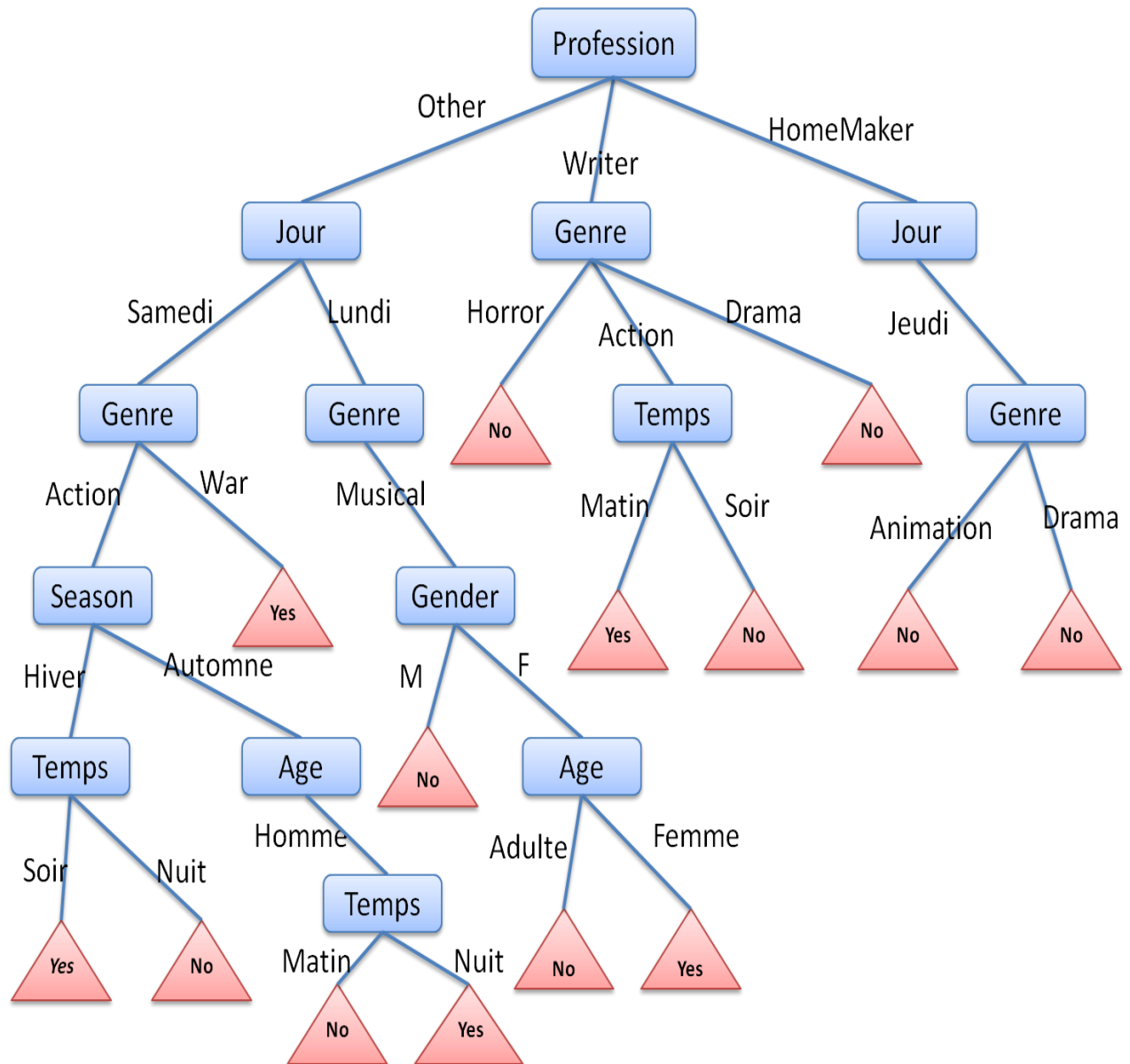


Figure 48: Une partie de l'arbre de décision

Après application de l'algorithme ID3 sur l'ensemble défini on a obtenu l'attribut « Genre » qui représente la racine de notre arbre caractérisé par un gain d'information optimum. Cela nous a permis d'extraire les règles d'associations suivantes :

- ✓ **If** (Profession =Other) **And** (jour = Samedi) **And** (Genre = Action) **And** (Season = Hiver) **And** (Temps = Soir) **Then** (Note = Yes).
- ✓ **If** (Profession =Other) **And** (Jour = Samedi) **And** (Genre = Action) **And** (Season = Hiver) **And** (Temps = Nuit) **Then** (Note = No).
- ✓ **If** (Profession =Other) **And** (Jour = Lundi) **And** (Genre = Musical) **And** (Gender = M) **Then** (Note = No).

- ✓ **If** (Profession = Other) **And** (Jour = Lundi) **And** (Genre = Musical) **And** (Gender = F) **And** (Age = Adulate) **Then** (Note = No).
- ✓ **If** (Profession = Other) **And** (Jour = Lundi) **and** (Genre = Musical) **And** (Gender = F) **And** (Age = Femme) **Then** (Note = Yes).
- ✓ **If** (Profession = Writer) **and** (Genre = Horror) **Then** (Note = No).
- ✓ **If** (Profession = Writer) **and** (Genre = Action) **And** (Temps = Matin) **Then** (Note = Yes).
- ✓ **If** (Profession = Writer) **and** (Genre = Action) **And** (Temps = Soir) **Then** (Note = No).
- ✓ **If** (Profession = Writer) **And** (Genre = Drama) **Then** (Note = No).
- ✓ **If** (Profession = HomeMaker) **And** (Jour = Jeudi) **And** (Genre = Animation) **Then** (Note = No).
- ✓ **If** (Profession = HomeMaker) **And** (Jour = Jeudi) **And** (Genre = Drama) **Then** (Note = No).

Dans « le chapitre 1, la section 3.3.2 » on a cité les différentes approches qui ont été proposées dans la littérature et Parmi ses approches on a choisit l'approche de la confiance maximale.

Dans les systèmes de recommandation des films, la confiance est calculée soit on se basant sur les utilisateurs ou les films. La probabilité de mettre une confiance dans les films est supérieure à celle des utilisateurs.

Comment calculer la confiance ?

Pour calculer la confiance on doit suivre les étapes suivantes :

- Préciser le genre de film
- Définir le nombre de tous les utilisateurs qui existe dans notre système « T »
- Définir la note (l'attribut Mieux Noté dans l'ensemble films.txt) de chaque film appartienne a ce genre « Ni » telle que le i varié de 1 a N.

$$\text{La confiance} = \sum_{i=1}^N N_i / T$$

Les genres sont ensuite triés par ordre décroissant.



1.2.2 Similarité User- User :

Pour calculer la similarité entre les utilisateurs voir « Chapitre 1, section 3.1 » il y'a plusieurs méthodes et parmi elles, on a choisit la Distance de Jaccard après avoir et Arrondi les données.

Comme on a utilisé des données temporelles (Contextualisation) dans notre système, et après utilisation de la distance de Jaccard pour calculer la similarité on a remarqué que cette dernière ne prend pas en considération ces données ceci est illustré par la **tableau 13**.

	HP1	HP2	HP3	HP4	HP5	HP6
A	1			1		
B	1	1	1			
C					1	1
D			1	1	1	
E	1	1		1		

Tableau 13 : Les données de l'exemple

 Même Type Jour  Même Temps

Utilisant la distance de jaccard :

$$\text{Similarité (A-B)} = 1/4 - 4/4 = 3/4.$$

$$\text{Similarité (A-D)} = 1/4 - 4/4 = 3/4.$$

Qui est l'utilisateur le plus similaire à A ? B ou D ?

On ne peut pas distinguer l'utilisateur le plus similaire car : Similarité (A-B) = Similarité (A-D) , pour cela on a *proposé* une méthode basée sur la distance de jaccard et les données temporelles

Ou on a divisé la Note on trois :

- ✓ **Si** le Type de Jour est le même **Alors** Note =Note + 0.2.
- ✓ **Si** le Temps est le même **Alors** Note =Note + 0.3.
- ✓ **Si** les utilisateurs aiment le même film qu'elle que soit le temps **Alors** Note =Note + 0.5.

Avec cette méthode on recalcule la similarité entre :

$$\text{Similarité (A-B)} = 0.5/4 - 4/4 = 3.5/4. \dots\dots\dots 1$$

$$\text{Similarité (A-D)} = 0.8/4 - 4/4 = 3.2/4 \dots\dots\dots 2$$

Similarité (A-D) < Similarité (A-B) donc l'utilisateur le plus similaire est D.

$$\text{Similarité (A-E)} = 2/3 - 3/3 = 1/3 \dots\dots\dots 3$$

D'après 1,2 et 3 on conclue que l'utilisateur E est le plus similaire.

❖ **Remarque :**

On calcule la similarité selon l'historique des utilisateurs existant dans notre base et on a obtenu le résultat (cf. **figure 49**) après une exécution qui a duré 3 jours

1	IdUser	IdUserSimil
2	1	823
3	2	931
4	3	616
5	4	220
6	5	804
7	6	474
8	7	308
9	8	638
10	9	744
11	10	321
12	11	59

Figure 49: Le fichier SimilUserUser.txt

L'arbre de décision obtenue après l'exécution d'ID3 sur ce fichier est illustré par la **figure 50**.

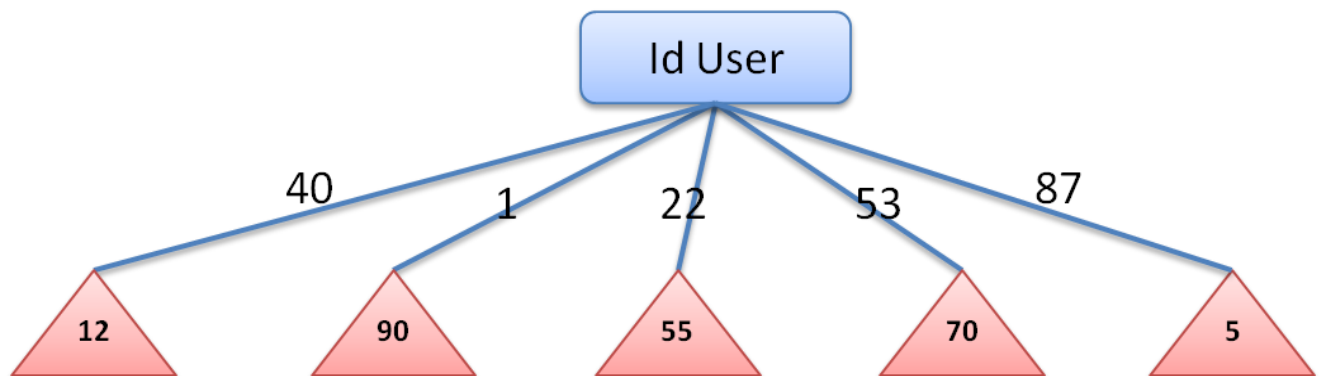


Figure 50: Représentation d' une partie d'un arbre de décision Obtenu après l'exécution sur SimilUserUser.txt d'ID3

1.2.3 Similarité Item-Item :

Elle permet de calculer la similarité entre les films vus par un utilisateur donné. Dans ce cas on utilise la distance de Jaccard, après avoir arrondi les données tenant compte aussi de l'historique de l'utilisateur. (Voir **Figure 51**).

1	IdItem	IdItemSimil
2	1	7
3	2	393
4	3	92
5	4	655
6	5	234
7	6	279
8	7	276
9	8	393
10	9	276
11	10	13
12	11	7
13	12	234

Figure 51: Le fichier SimilItem-Item.txt

L'arbre de décision obtenue après l'exécution d'ID3 est illustré par la **figure 52**

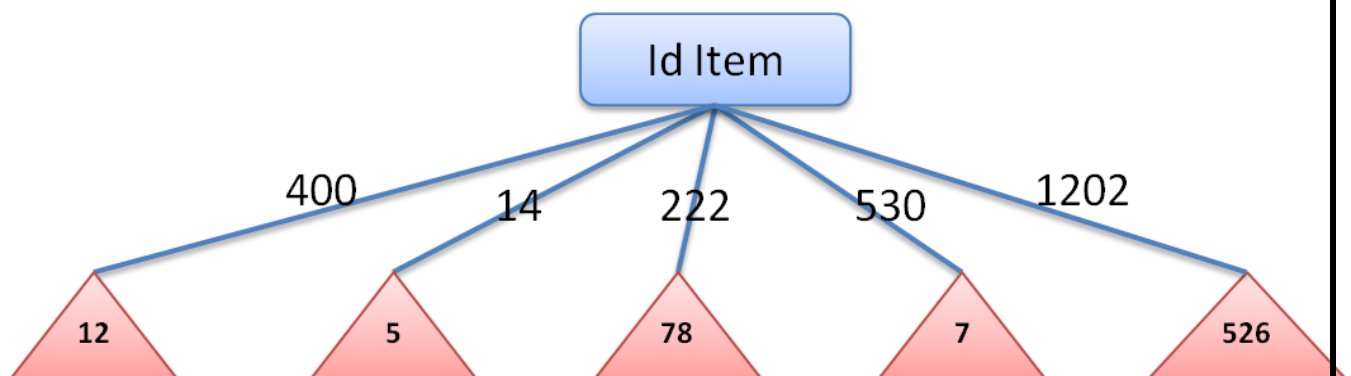


Figure 52 : Représentation d'une partie d'un arbre de décision Obtenu après l'exécution sur SimilItem-Item.txt d'ID3

1.3 Module de recommandation :

1.3.1 La recommandation cotée profil :

- **Recommandation utilisant l'arbre de Décision :**

Après la création de l'arbre de décision et le calcul de la confiance on a obtenu comme résultat tous les genres des films triés par ordre décroissant que l'utilisateur peut aimer à un moment donné. A partir de ces genres triés on sélectionne les films les mieux notés.

- **Recommandation utilisant la similarité User-User:**

Comme on n'a pas beaucoup d'information sur les films aimés par cet utilisateur, le calcul de similarité ne nous donne pas un résultat satisfaisant.

- **Recommandation utilisant la similarité Item-Item:**

Faute d'absence d'historique de ce profil, on peut pas connaître les films les plus similaires. Dans ce cas la, on ne peut pas effectuer la recommandation basé sur la similarité item-item.

- **Recommandation Final :**

D'après la recommandation qu'on a formulée dans la partie arbre de décision on a obtenu une liste de films. A partir de cette dernière on fait une union pour obtenir la liste finale de recommandation des films. Représentée par la **figure 53**.

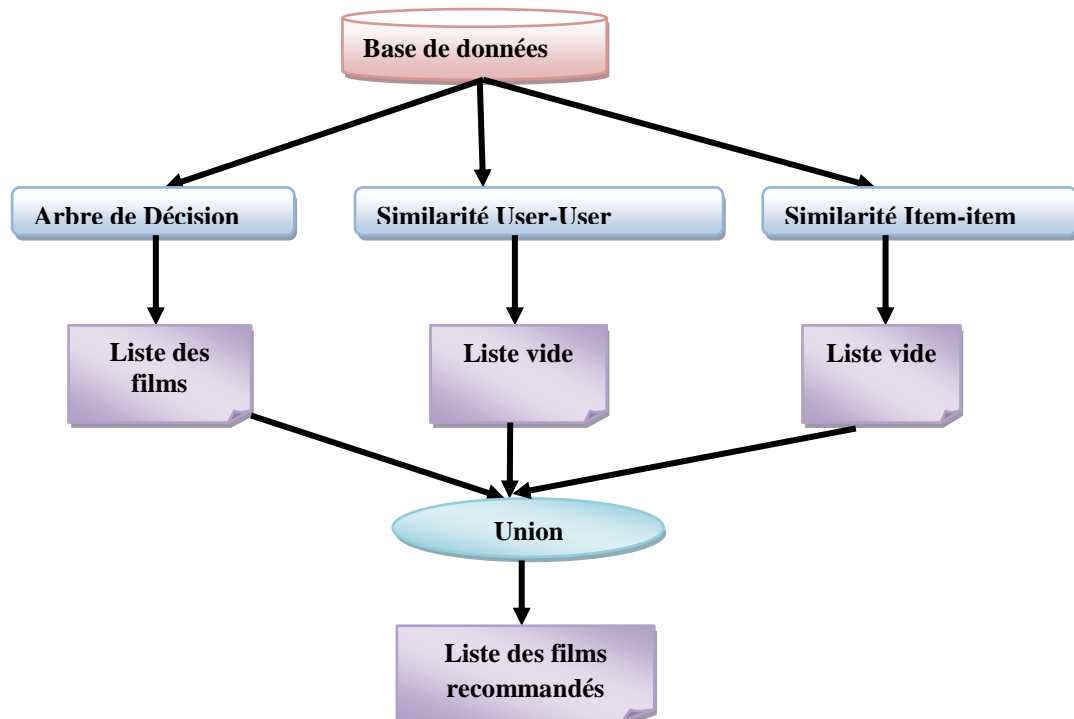


Figure 53 : Etapes de recommandation finale cotée profil

1.3.2 Recommandation cotée Historique :

- **Recommandation utilisant l'Arbre de Décision :**

D'après l'historique on se propose de créer un arbre de décision seulement pour un utilisateur qui nous permettra de prédire à la fin une liste de films. Susceptibles d'être préféré par cet utilisateur

- **Recommandation utilisant la similarité User-User:**

Après identification de l'utilisateur le plus similaire dans un moment donné, on prend tous les films les mieux notés par cet utilisateur (ou note = 4 ou 5 étoile).

- **Recommandation utilisant la similarité Item-Item:**

Utilisant l'historique de l'utilisateur on calcule la similarité entre les films aimés par cet utilisateur.

- **Recommandation Final :**

D'après la recommandation effectuée dans la partie Arbre de Décision, User-User et Item-Item on a obtenu trois listes de films. A partir de ses derniers on fait une union pour obtenir la liste finale de recommandation des films qui est schématisé par la **figure 54**.

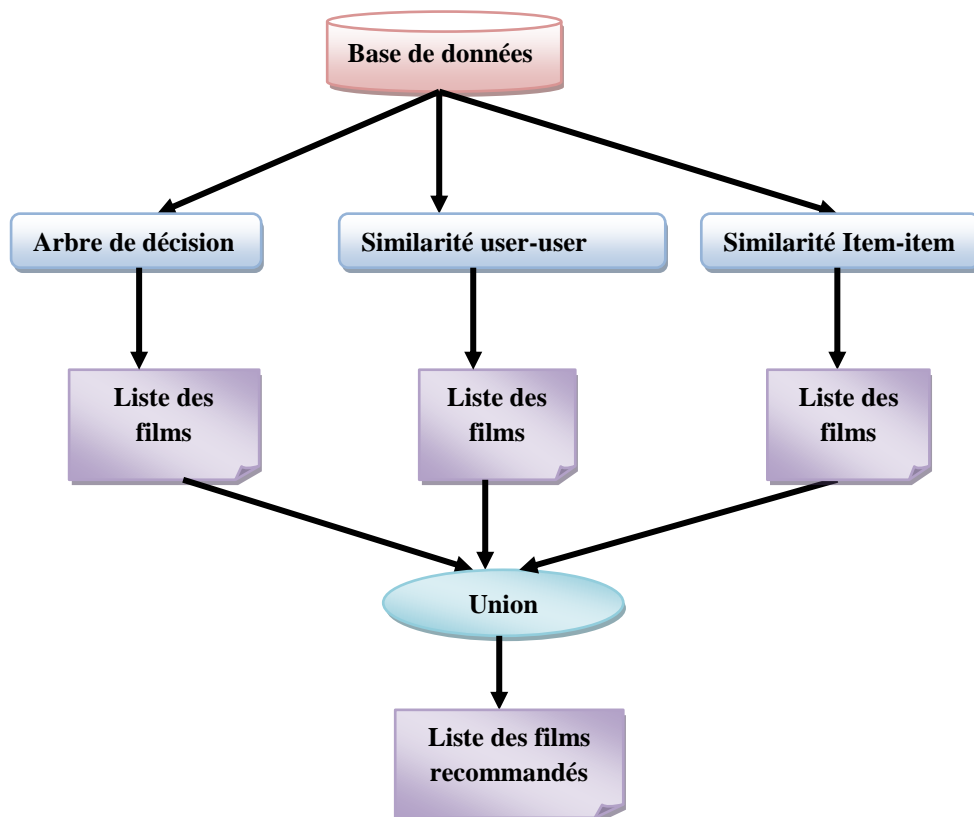


Figure 54 : Recommandation finale cotée Historique

2 Module d'évaluation de notre système :

La phase d'évaluation va donc en premier lieu essayer de déterminer la performance de notre système, dans le cas où l'erreur est grande on essayera de proposer de nouvelles solutions pour le système.

GroupLens ont collecté aussi un ensemble de données dans la base MoviLens 100k et ces ensembles sont utilisés soit pour :

- **Validation et Teste :**

$\langle \text{ua.base}, \text{ua.test} \rangle, \langle \text{ub.base}, \text{ub.test} \rangle$

Les sous ensembles $\langle \text{ua.base}, \text{ua.test} \rangle, \langle \text{ub.base}, \text{ub.test} \rangle$ sont obtenus après la division de l'ensemble de données $u.data$, pour former un ensemble représentant l'apprentissage et un sous ensemble de test avec 10 notes données par un seul utilisateur dans le test. Les sous ensembles $\langle \text{ua.base}, \text{ua.test} \rangle, \langle \text{ub.base}, \text{ub.test} \rangle$ sont disjoints.

- **Validation Croisé :**

$\langle u1.base, u1.test \rangle, \langle u2.base, u2.test \rangle, \langle u3.base, u3.test \rangle, \langle u4.base, u4.test \rangle, \langle u5.base, u5.test \rangle$

A partir de l'ensemble de données $u.data$ on a obtenu Ces sous ensembles de données qui sont divisés en 80% pour l'apprentissage et 20% pour le test chaque sous ensembles d'apprentissage $\langle u1, \dots, u5 \rangle$ à un sous ensemble de tests disjoints (5 fold cross validation). Les étapes d'évaluation de notre système sont portées dans la **figure 55**.

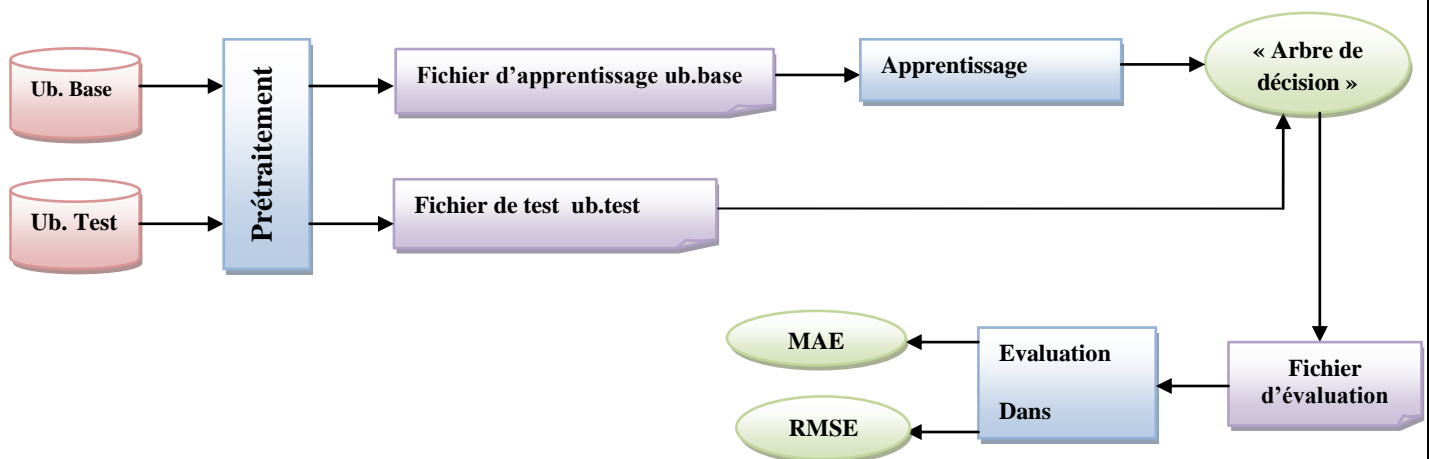


Figure 55: étapes d'évaluation de notre système

2.1 Différents types d'évaluations

2.1.1 Evaluation du Système de recommandation basé sur les arbres de décision basic :

Dans les systèmes de recommandation les données utilisées pour construire les arbres de décisions basics sont composés de « UserId, ItemId, Note »

- **Etapes d'évaluations :**

- a) **Phase de Prétraitement :**

Après le prétraitement sur le sous ensemble d'apprentissage et de teste < ub.base, ub.test> qui contient les colonnes suivants : IDUser, IDItem, Note, Temps. On obtient deux sous ensembles qui sont représentés dans les deux figures (58 et 59).

1	IDUser	IDItem	Note	Temp
2	1	1	5	874965758
3	1	2	3	876893171
4	1	3	4	878542960
5	1	4	3	876893119
6	1	5	3	889751712
7	1	6	5	887431973
8	1	7	4	875071561
9	1	8	1	875072484
10	1	9	5	878543541
11	1	10	3	875693118
12	1	11	2	875072262
13	1	12	5	878542960

Figure 56: Fichier d'apprentissage ub.base

7	IdUser	IdItem	Note	Temp
8	1	222	4	878873388
9	1	227	4	876892946
10	1	228	5	878543541
11	1	253	5	874965970
12	2	257	4	888551062
13	2	279	4	888551745
14	2	299	4	888550774
15	2	301	4	888550631
16	2	303	4	888550774
17	2	307	3	888550066

Figure 57 : Fichier de test ub.test

	IDUser	IDItem	Note
1	1	1	5
2	1	2	3
3	1	3	4
4	1	4	3
5	1	5	3
6	1	6	5
7	1	7	4
8	1	8	1
9	1	9	5
10	1	10	3
11	1	11	2

Figure 58 : fichier ub.base après le prétraitement

	IdUser	IdItem	Note
7	1	222	4
8	1	227	4
9	1	228	5
10	1	253	5
11	2	257	4
12	2	279	4
13	2	299	4
14	2	301	4
15	2	303	4

Figure 59 : fichier ub.test après le prétraitement

b) La phase d'apprentissage :

L'apprentissage de l'arbre de décision se fait à partir du fichier traité ub.base, où les entrées sont : IdUser, IDItem, et l'attribut cible est la Note. L'arbre de décision est représenté par la **figure 60**)

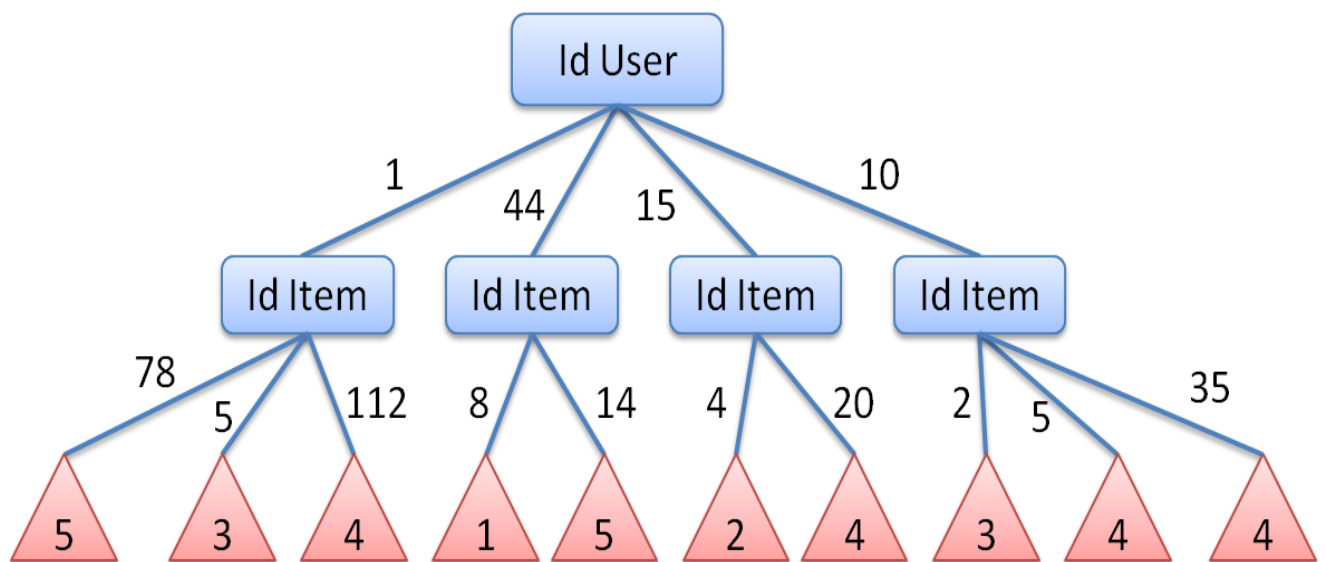


Figure 60: partie d'un arbre de décision lors de l'apprentissage utilisant le fichier ub.base

c) La phase de teste :

Dès que l'arbre de décision est créé on utilisera le fichier ub.test traité pour évaluer le résultat, et le sauvegarder dans un fichier d'évaluation qui contient les colonnes : IdUser, IdItem, Note, Note Prédicat, ce fichier est présenté dans la **figure 61**.

1	IdUser	IdItem	Note	NotePredict
2	1	17	3	Null
3	1	47	4	Null
4	2	64	5	Null
5	2	90	4	Null
6	3	92	3	Null
7	3	113	5	Null
8	4	222	4	Null
9	4	227	4	Null

Figure 61 : fichier d'évaluation utilisant l'arbre de décision basic

d) La phase d'évaluation :

Pour l'évaluation de notre système on a calculé les deux erreurs présentés dans le « chapitre 1, section 6 » RMSE et MAE (cf. [tableau 13](#)).

Erreur	Valeur
RMSE	3,75
MAE	3,58

Table 13 : valeurs des erreurs selon l'arbre de décision basic

2.1.2 Système de recommandation basé sur les arbres de décision et sur les contenus

- Les étapes d'évaluations :

a) La phase de prétraitement :

Après le prétraitement sur le sous ensemble d'apprentissage et de teste $\langle \text{ub.base}, \text{ub.test} \rangle$, on utilise *IdUser* et on le remplace par d'autre informations : Age, Sexe et la profession, puis on supprime la colonne du temps.

La [figure 62](#) représente le fichier *ub.base* ou *u.test* est après le prétraitement.

<i>Age</i>	<i>Gender</i>	<i>Profession</i>	<i>IdItem</i>	<i>Note</i>
<i>JeuneHomme</i>	<i>M</i>	<i>technician</i>	<i>1</i>	<i>5</i>
<i>JeuneHomme</i>	<i>M</i>	<i>technician</i>	<i>2</i>	<i>3</i>
<i>Adulte</i>	<i>F</i>	<i>other</i>	<i>310</i>	<i>4</i>
<i>Adulte</i>	<i>F</i>	<i>other</i>	<i>311</i>	<i>5</i>
<i>Homme</i>	<i>F</i>	<i>other</i>	<i>1</i>	<i>4</i>
<i>Homme</i>	<i>F</i>	<i>other</i>	<i>2</i>	<i>3</i>
<i>Adulte</i>	<i>M</i>	<i>executive</i>	<i>1</i>	<i>4</i>
<i>Adulte</i>	<i>M</i>	<i>executive</i>	<i>7</i>	<i>2</i>

Figure 62 : Le fichier ub.base après le prétraitement (basé sur le contenu)

b) La phase d'apprentissage :

L'apprentissage de l'arbre de décision (cf. **figure 63**) se fait à partir du fichier traité ub.base, ou les entrées sont : Idfilm , Profession, Age, Gender et l'attribut cible est la note.

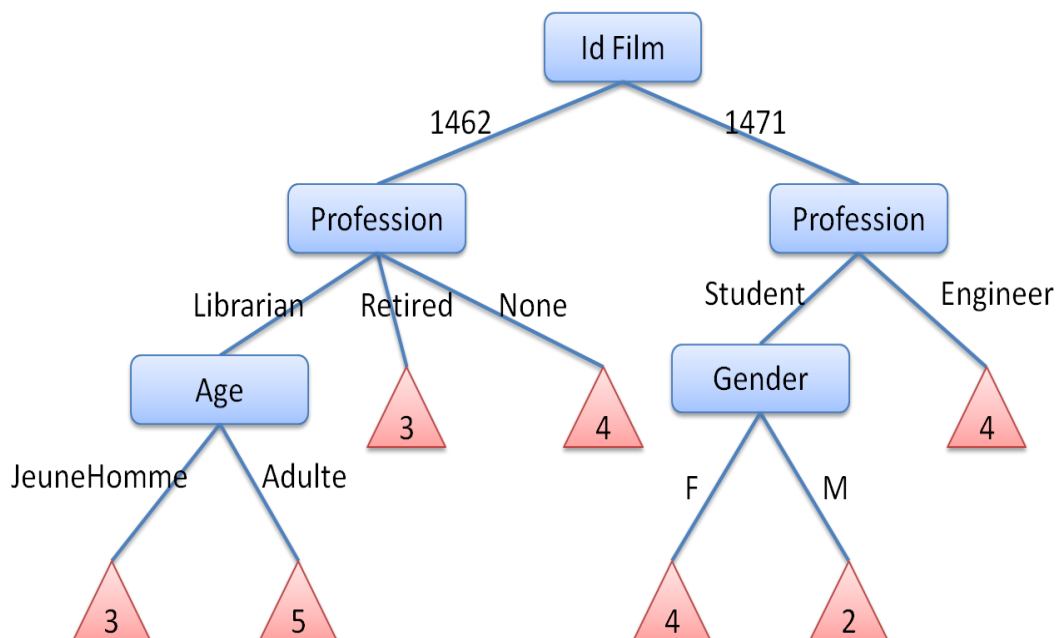


Figure 63 : Une partie d'un arbre de décision lors de l'apprentissage sur ub.base

c) La phase de teste :

Dés que l'arbre de décision est créé on utilisera le fichier `ub.test` traité pour évaluer le résultat et le sauvegarder dans un fichier d'évaluation qui contient les colonnes : Age, Gender, Profession, IdItem, Note. La **figure 64** illustre ce fichier.

<i>Age</i>	<i>Gender</i>	<i>Profession</i>	<i>IdItem</i>	<i>Note</i>	<i>NotePredict</i>
<i>JeuneHomme</i>	<i>M</i>	<i>technician</i>	<i>17</i>	<i>3</i>	<i>2</i>
<i>JeuneHomme</i>	<i>M</i>	<i>technician</i>	<i>47</i>	<i>4</i>	<i>2</i>
<i>JeuneHomme</i>	<i>M</i>	<i>technician</i>	<i>64</i>	<i>5</i>	<i>5</i>
<i>Adulte</i>	<i>F</i>	<i>other</i>	<i>303</i>	<i>4</i>	<i>3</i>
<i>Adulte</i>	<i>F</i>	<i>other</i>	<i>307</i>	<i>3</i>	<i>2</i>
<i>Adulte</i>	<i>F</i>	<i>other</i>	<i>308</i>	<i>3</i>	<i>3</i>
<i>Homme</i>	<i>F</i>	<i>other</i>	<i>154</i>	<i>4</i>	<i>3</i>
<i>Homme</i>	<i>F</i>	<i>other</i>	<i>196</i>	<i>3</i>	<i>5</i>
<i>Homme</i>	<i>F</i>	<i>other</i>	<i>199</i>	<i>3</i>	<i>4</i>
<i>Homme</i>	<i>F</i>	<i>other</i>	<i>387</i>	<i>4</i>	<i>3</i>

Figure 64 : Fichier d'évaluation utilisant l'arbre de décision basé sur le contenu

d) La phase d'évaluation :

Les deux erreurs sont calculées et présentées dans le **tableau 14**.

Erreur	Valeur
RMSE	1,58
MAE	1,14

Tableau 14 : valeurs des erreurs selon l'arbre de décision basé sur le contenu

2.1.3 Système de recommandation basé sur les arbres de décision et sur la contextualisation

- Les étapes d'évaluations :

a) La phase de prétraitement :

Après le prétraitement sur le sous ensemble d'apprentissage et de teste `< ub.base, ub.test >`, nous avons classifié le temps en milliseconde depuis 1970 selon : Jour, Type de Jour, Season et Temps.

La **figure 65** représente le fichier ub.base ou u.test après le prétraitement.

1	IdUser	IdItem	Note	Jour	TypeJour	Season	Temp
2	1	1	5	lundi	Normal	Automne	Soir
3	1	2	3	mercredi	Normal	Automne	Matin
4	1	3	4	lundi	Normal	Automne	Matin
5	2	274	3	vendredi	Vacance	Hiver	Nuit
6	2	275	5	dimanche	Vacance	Printemp	AprèsMidi
7	2	276	4	jeudi	Vacance	Hiver	Nuit

Figure 65 : fichier ub.base après le prétraitement (basé sur le contexte)

b) La phase d'apprentissage :

L'apprentissage de l'arbre de décision se fait à partir du fichier traité ub.base, où les entrées sont : Id User, Id Item, Jour, Type de Jour, Season, Temps l'attribut cible est la note. La **figure 66** Représente l'arbre de d écision.

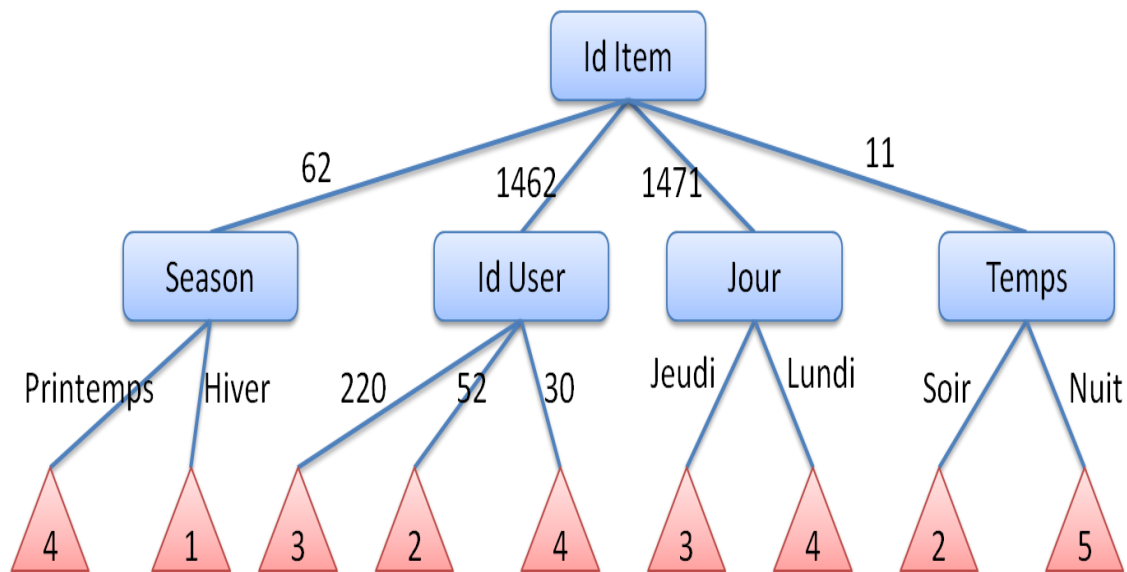


Figure 66: L'Arbre de décision obtenu (basé sur le contexte)

c) La phase de teste :

Dés que l'arbre de décision est crée on utilisera le fichier ub.test traité pour évaluer le résultat et le sauvegarder dans un fichier d'évaluation (cf. **figure 67**) qui contient les colonnes : Id User, Id Item, Note, Jour, Type de Jour, Season, Temps.

1	IdUser	IdItem	Note	Jour	TypeJr	Season	Time	NotePredict
2	1	17	3	mercredi	Normal	Automne	Nuit	null
3	1	47	4	mercredi	Normal	Automne	Nuit	null
4	9	50	5	dimanche	Normal	Hiver	ApréMidi	null
5	2	313	5	vendredi	Vacance	Hiver	Soir	null
6	3	330	2	samedi	Vacance	Printemp	Nuit	null
7	6	357	4	mercredi	Normal	Hiver	Soir	null
8	8	686	3	mercredi	Normal	Automne	Soir	null

Figure 67 : Fichier d'évaluation utilisant l'arbre de décision (basé sur le contexte)

d) La phase d'évaluation :

Les deux erreurs sont calculées et présentées dans le [tableau 15](#).

Erreur	Valeur
RMSE	3,75
MAE	3,57

Table 15 : valeurs des erreurs selon l'arbre de décision basé sur le contexte

❖ Remarque :

Si on teste la règle suivante **If** (idItem = 1462) **And** (idUser = 5) **Then** (note = Nuls) ; selon l'arbre de décision présenté dans la figure N° la prédiction obtenu est nulle (dans le cas si l'utilisateur numéro 5 n'a pas regardé le film numéro1462), c'est pour cette raison soit on n'utilise pas l'algorithme ID3 soit on va laisser l'attribut Id User en dernier

2.1.4 Système de recommandation basé sur les arbres de décision basé sur contenus et sur la contextualisation

• Les étapes d'évaluations :

a) La phase de prétraitement :

Après le prétraitement sur le sous ensemble d'apprentissage et de teste < ub.base, ub.test>, nous avons classifié le temps selon : Jour, Type de Jour, Saison et le Temps. Et nous avons remplacé le :

- Id User par l'Age, Gender, et profession
- IdItem par le gère

La [figure 68](#) représente le fichier ub.base ou u.test après le prétraitement

Age	Gender	Profession	Genre	Note	Jour	TypeJr	Season	Time
JeuneHomme	M	technician	Animation	5	lundi	Normal	Automne	Soir
JeuneHomme	M	technician	Action	3	mercredi	Normal	Automne	Matin
Homme	F	other	Action	5	mardi	Normal	Automne	ApréMidí
Homme	M	administrator	Comedy	1	mercredi	Normal	Automne	Soir
Adulte	F	other	Comedy	3	vendredi	Vacance	Hiver	Nuit
Adulte	F	other	Action	1	mercredi	Normal	Printemp	Nuit
Homme	F	other	Animation	5	jeudi	Normal	Automne	Soir
Homme	M	administrator	Action	4	mercredi	Normal	Automne	Soir

Figure 68 : fichier ub.base après le prétraitement (basé sur le contenu et la contextualisation)

b) phase d'apprentissage :

Cette arbre est représentée dans la [figure 48](#).

c) La phase de teste :

Dés que l'arbre de décision est créé (cf. [figure 69](#)), on utilisera le fichier ub.test traité pour évaluer le résultat et le sauvegarder dans un fichier d'évaluation qui contient les colonnes : Age, Gender, Profession, Genre, Note, Jour, Type de Jour, Season, Time

1	Age	Gender	Profession	Genre	Note	Jour	TypeJr	Season	Time	NotePredict
2	JeuneHomme	M	technician	Action	4	mercredi	Normal	Automne	Matin	3
3	JeuneHomme	M	technician	Action	5	lundi	Normal	Automne	Matin	4
4	JeuneHomme	M	technician	Drama	5	lundi	Normal	Automne	Soir	5
5	Adulte	F	other	Action	4	vendredi	Vacance	Hiver	Nuit	4
6	Adulte	F	other	Drama	4	vendredi	Vacance	Hiver	Nuit	4
7	Adulte	F	other	Crime	4	vendredi	Vacance	Hiver	Nuit	4
8	Adulte	F	other	Comedy	4	vendredi	Vacance	Hiver	Nuit	4

Figure 69: fichier d'évaluation utilisant l'arbre de décision (basé sur le contexte et contextualisation)

d) La phase d'évaluation :

Les deux erreurs sont calculées et présentées dans le [tableau 16](#).

Erreur	Valeur
RMSE	1,43
MAE	1,02

Tableau 16 : Valeurs des erreurs selon l'arbre de décision basé sur le contexte et la contextualisation

2.1.5 Système de recommandation basé sur l'historique d'un utilisateur et la contextualisation utilisons les arbres de décision.

- Les étapes d'évaluations :

- a) La phase prétraitement :

Après le prétraitement sur le sous ensemble d'apprentissage et de teste $\langle \text{ub.base}, \text{ub.test} \rangle$, nous avons on a remplacé le Id Item par le genre et on a classifié le temps : Jour, Type de Jour, Season, Time. (cf. **figure 70**).

	IdUser	GenreFilm	Like/Dislike	Day	KindOfDay	Season	Time
1	1	Drama	Yes	lundi	Normal	Automne	Matin
2	1	Animation	Yes	dimanche	Normal	Printemp	Matin
3	1	Action	Yes	lundi	Normal	Automne	Matin
4	1	Drama	Yes	mercredi	Normal	Automne	Nuit
5	1	Drama	Yes	samedi	Vacance	Hiver	Soir
6	1	Comedy	Yes	mercredi	Normal	Automne	Nuit
7	1	Comedy	Yes	vendredi	Vacance	Printemp	Nuit
8	1	Comedy	Yes	vendredi	Vacance	Printemp	Nuit

Figure 70 : fichier ub.base après le prétraitement (sur l'historique de l'utilisateur N°=1 basée sur la contextualisation)

- b) La phase d'apprentissage:

L'apprentissage de l'arbre de décision (**figure 71**) se fait a partir du fichier traité ub.base, ou les entrées sont et l'attribut cible est la note.

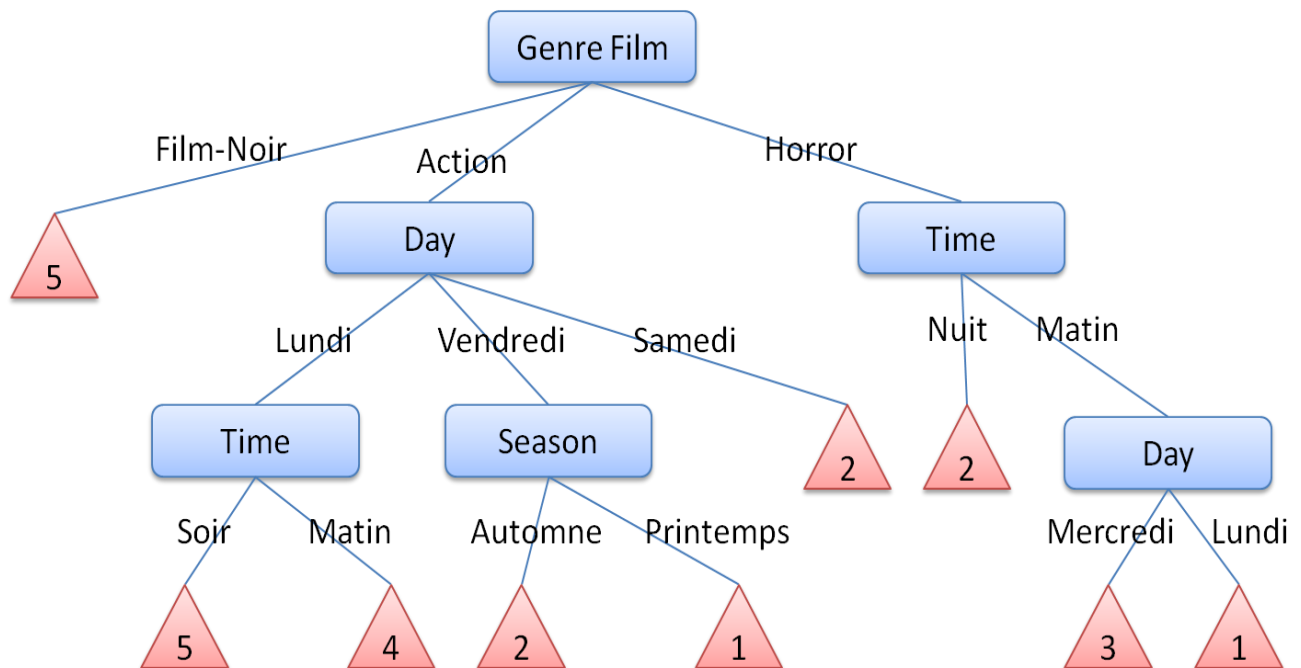


Figure 71 : L'Arbre de décision obtenu basé sur l'historique d'un utilisateur et la contextualisation utilise les arbres de décision

c) La Phase de teste :

Dés que l'arbre de décision est crée (cf. **figure 72**) on utilisera le fichier ub.test traité pour évaluer le résultat et le sauvegarder dans un fichier d'évaluation qui contient les colonnes :

1	IdUser	GenreFilm	Note	Day	KindOfDay	Season	Time	NotePredict
2	1	Action	3	mercredi	Normal	Automne	Nuit	4
3	1	Comedy	4	mercredi	Normal	Automne	Nuit	4
4	1	Drama	5	mercredi	Normal	Automne	Nuit	4
5	1	Comedy	4	lundi	Normal	Automne	Matin	3
6	1	Action	3	mercredi	Normal	Automne	Matin	3
7	1	Drama	5	lundi	Normal	Automne	Matin	5
8	1	Action	4	vendredi	Vacance	Automne	Nuit	2
9	1	Action	4	mercredi	Normal	Automne	Matin	3
10	1	Action	5	lundi	Normal	Automne	Matin	4
11	1	Drama	5	lundi	Normal	Automne	Soir	5

Figure 72 : fichier d'évaluation basé sur l'historique d'un utilisateur et la contextualisation utilisons les arbres de décision)

d) phase d'évaluation :

Les deux erreurs sont calculées et présentées dans le tableau suivant, ce résultat est encourageant du moment que nous avons obtenus une valeur d'erreur similaire a celle obtenu par NETFLIX avant que cette dernière ne parvienne a une erreur de 0.80 après avoir utilisé 100 méthodes de prédictions

Erreur	Valeur
RMSE	0,90
MAE	0,63

Tableau 17 : valeurs des erreurs selon l'historique d'un utilisateur et la contextualisation utilisant les arbres de décision

Conclusion :

Notre système a pour objectif de proposer une nouvelle méthode de recommandation on utilise les arbres de décision et des données temporelles pour comparer les résultats obtenus avec la recommandation on utilisant les arbres de décision basique.

Dans la partie suivante, nous allons présenter les outils utilisés pour implémenter notre système, puis nous allons effectuer un ensemble de tests de performance, pour comparer les deux méthodes.