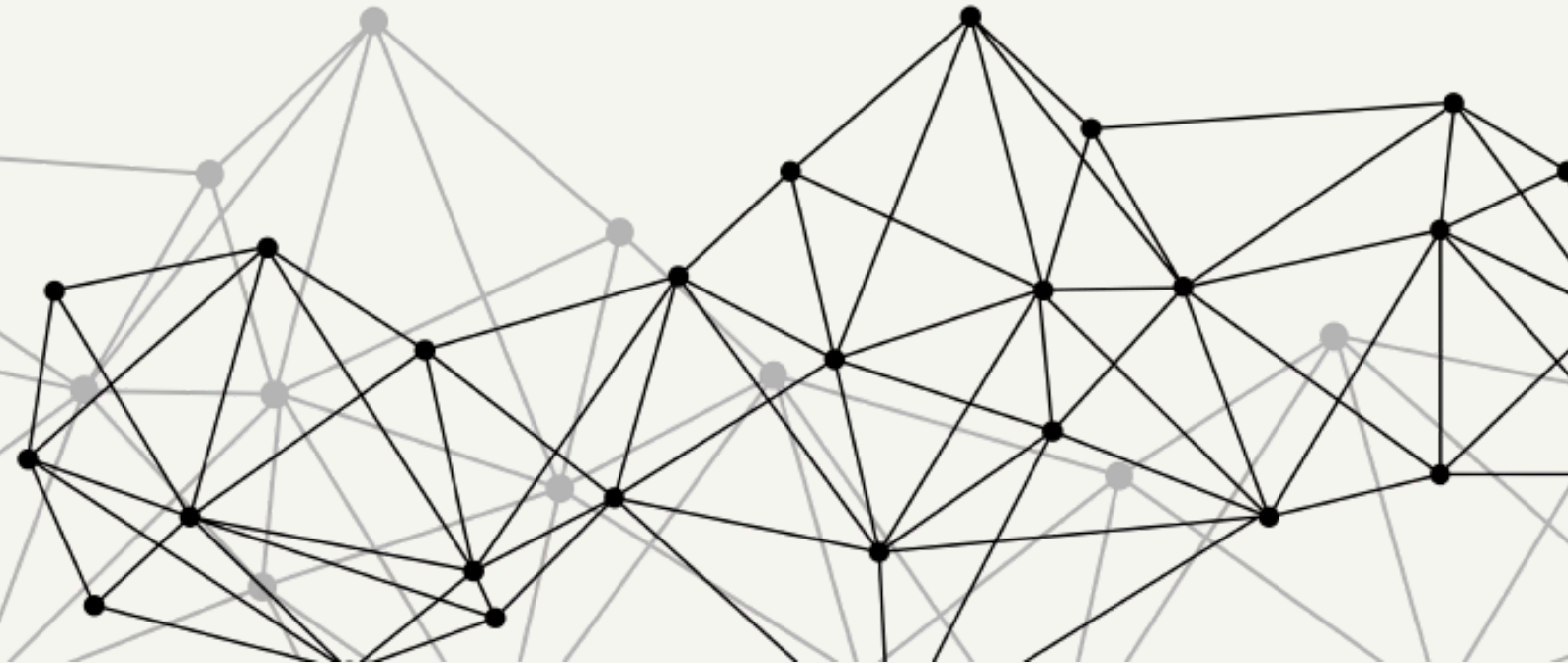
Projet | Big Data

Films Recommandation

Réalisé par :

- HABIBI Mohammed

SOMMAIRE :



INTRODUCTION

Les systèmes de recommandation sont des outils permettant d'interagir avec des espaces d'information vastes et complexes. Les recherches sur les systèmes de recommandation ont intégré une grande variété de techniques d'intelligence artificielle, notamment l'apprentissage automatique, l'exploration de données, la modélisation des utilisateurs, le raisonnement basé sur des cas et la satisfaction des contraintes, entre autres.

L'objectif d'un système de recommandation est de générer des recommandations significatives à un ensemble d'utilisateurs pour des articles ou des produits qui pourraient les intéresser. Nous visons ici à réaliser un système de recommandation des films via une application web. La recommandation se base sur les l'évaluation (Rating) de l'utilisateur à propos les films qui les a déjà regardé.

En règle générale, un système de recommandation compare le profil de l'utilisateur à certaines caractéristiques de référence et cherche à prédire la « note (Rating) » qu'un utilisateur attribuerait à un élément auquel il n'avait pas encore pensé. Et c'est exactement ce qu'on vise réaliser via ce projet.

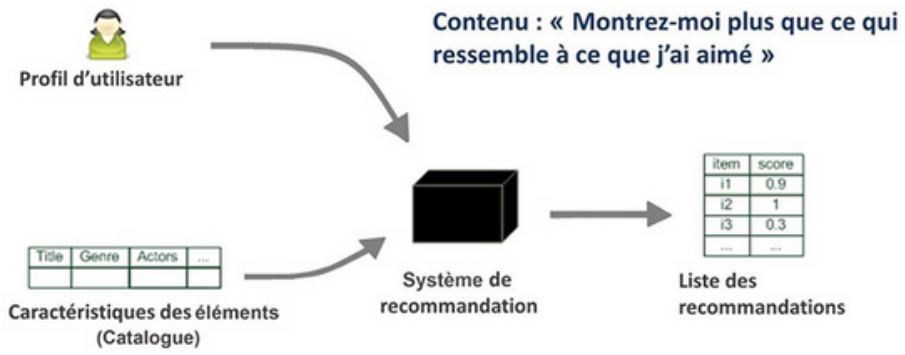
OUTILS UTILISES

• Approche utilisée :

Il existe trois approches pour réaliser un système de recommandation. Nous avons opté pour celle qui est **basée sur le contenu**. Juste pour l'information, Nous allons citer les caractéristiques de chacune de ces trois.

• Approche basée sur le contenu :

Pour les recommandations basées sur le contenu, la tâche consiste à déterminer quels éléments du catalogue coïncident le mieux avec les préférences de l'utilisateur. Une telle approche ne requiert pas une grande communauté d'utilisateurs ou un gros historique d'utilisation du système. La figure 1 illustre ce processus. Ci-dessous, vous trouverez un schéma simplifiant cette approche :



Un système de recommandation basé uniquement sur le contenu fait des recommandations pour un utilisateur uniquement sur la base du profil construit en analysant le contenu des éléments que cet utilisateur a évalués dans le passé. Cette méthode suggère des éléments aux utilisateurs en fonction des informations de contenu des éléments. Par exemple, dans notre cas, nous pouvons suggérer des films basés sur un genre similaire au genre qu'un utilisateur particulier préfère généralement.

Nous avons utilisé le profil d'élément et le profil d'utilisateur pour suggérer des éléments aux utilisateurs. Profil de l'item : I = vecteur de taille 18 (nb de Genres) avec 1 ou 0 marquant la présence d'un Genre. Profil de l'utilisateur:

1. Trouvez la note moyenne donnée par chaque utilisateur.
2. Soustrayez cette moyenne de toutes les notes d'un utilisateur pour obtenir des notes normalisées.
3. Supposons qu'un utilisateur classe "n" films avec le genre "A", puis le poids du profil du genre "A" pour cet utilisateur sera :

$$UA = (r1 + r2 + r3 + \dots + rn) / n$$

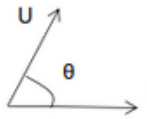
où r1,r2,r3,...rn sont les évaluations normalisées pour cet utilisateur.

De même, recherchez UB ,UC... pour obtenir le profil utilisateur de cet utilisateur sous la forme U = [UA UB UC..... 18]
Prédire :
Pour recommander des articles à un utilisateur, nous trouverons un paramètre de similarité (cosinus similarité) pour tous les articles que l'utilisateur n'a pas évalués. Recommandez ensuite les éléments « h » supérieurs, avec un paramètre de similarité élevé, à cet utilisateur.

Similitude cosinus :

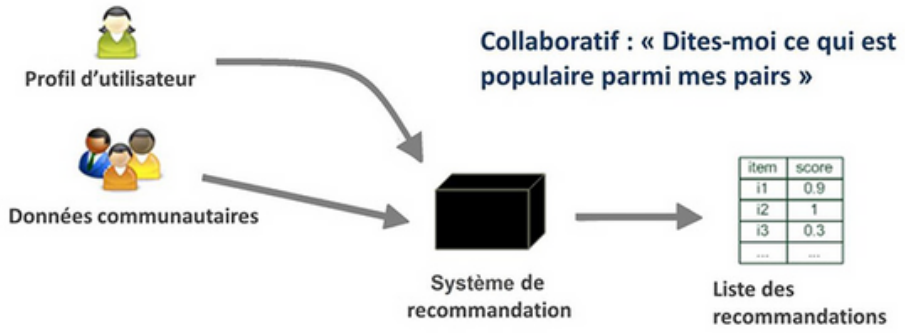
$$\cos\theta = \frac{U \cdot I}{|U| * |I|}$$

qui est en fait le produit scalaire des vecteurs U et I.



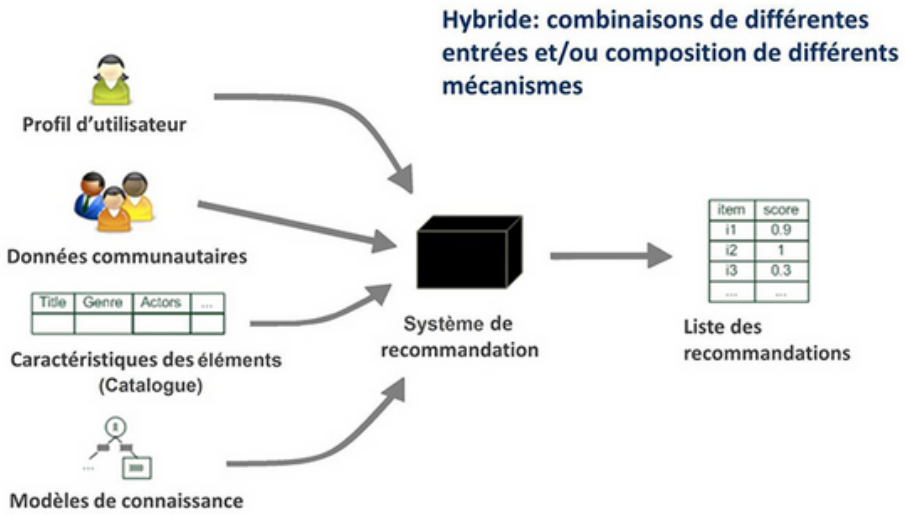
• Approchs basée sur le filtrage collaboratif :

Les systèmes basés sur le filtrage collaboratif produisent des recommandations en calculant la similarité entre les préférences d'un utilisateur et celles d'autres utilisateurs. De tels systèmes ne tentent pas d'analyser ou de comprendre le contenu des éléments à recommander. La méthode consiste à faire des prévisions automatiques sur les intérêts d'un utilisateur en collectant des avis de nombreux utilisateurs. L'hypothèse sous-jacente de cette approche est que ceux qui ont aimé un élément spécifique dans le passé auront tendance à aimer cet élément spécifique, ou un autre très « proche », à nouveau dans l'avenir.



- **Approche hybride :**

Un système de recommandation hybride utilise des composants de différents types d'approches de recommandation ou s'appuie sur leur logique. Par exemple, un tel système peut utiliser à la fois des connaissances extérieures et les caractéristiques des éléments, combinant ainsi des approches collaboratives et basées sur le contenu.



- **Data utilisée :**

Nous avons utilisé l'ensemble de données MovieLens 100k très populaire pour tester nos systèmes. La raison de l'utilisation de cet ensemble de données est que les données sont mises à disposition par une source réputée et fiable (Université du Minnesota) et contiennent les informations sur les films et les données démographiques des utilisateurs qui ont été requises dans notre analyse basée sur le contenu. L'ensemble de données contient 100 000 notes de 943 utilisateurs sur 1682 films. Chaque utilisateur a évalué au moins 20 films et des informations sont également données sur les films, y compris le titre, la distribution, la sortie, la date et son genre

- # Technologie utilisée :

- ## HADOOP :

Hadoop est un framework open source qui repose sur Java. Hadoop prend en charge le traitement des données volumineuses (Big Data) au sein d'environnements informatiques distribués. Hadoop fait partie intégrante du projet Apache parrainé par l'Apache Software Foundation. Il permet d'exécuter des applications sur des systèmes en cluster dotés de milliers de noeuds impliquant des centaines de téraoctets de données.

- ## PIG :

Pig est un outil de traitement de données qui fait partie de la suite Hadoop et qui permet l'écriture de scripts qui sont éxécutés sur l'infrastructure Hadoop sans être obligé de passer par l'écriture de tâche en Java via le framework MapReduce. Il dispose en outre de fonctionnalités permettant le chargement de données depuis une source externe vers le cluster HDFS ou de fonctionnalités permettant l'export de données pour utilisation par des applications tierces.

- **SQOOP :**

Sqoop, abbréviation de SQL-to-Hadoop, est un utilitaire de transfert des données d'une base de données relationnelle au HDFS et du HDFS aux bases de données relationnelles. Vous pouvez utiliser Sqoop pour importer des données des SGBDR tels que MySQL, Oracle, SQL Server ou Teradata au HDFS, transformer les données dans Hadoop via le MapReduce ou un autre modèle de calcul, et les exporter en retour dans le SGBDR. Lorsque vous exportez les données du SGBDR pour Hadoop, vous pouvez les stocker sur Hive, HBase ou directement sur le HDFS.

- **MYSQL SERVER :**

MySQL est un serveur de bases de données relationnelles Open Source.
Un serveur de bases de données stocke les données dans des tables séparées plutôt que de tout rassembler dans une seule table. Cela améliore la rapidité et la souplesse de l'ensemble. Les tables sont reliées par des relations définies, qui rendent possible la combinaison de données entre plusieurs tables durant une requête. Le SQL dans "MySQL" signifie "Structured Query Language" : le langage standard pour les traitements de bases de données.

DESCRIPTION DU Work-flow

Le Work-Flow de ce projet peut être divisé en **5 étapes principales**.

- **Structure du répertoire HDFS :**

A) Répertoires que avons créé :

/RcomSys

- **/OriginalData** (Contains u.data file which is -> User Item Rating)
- **/Items** (Contains u.item -> Item details)
- **/User** (Contains u.user -> User details)
- **/Genre** (Contains u.genre -> Genre details)

B) Répertoires qui seront automatiquement créé :

/RcomSys

- **/UserItemRating** (Output of MapReduce Job -> Modifeid Dataset)
- **/ContentOut** (Output of ContentBased.pig)

- **Steps :**

1) Stockage des données sur HDFS :

Dans cette étape, nous avons uploader nos données directement sur HDFS.

2) Modification de notre Data :

L'outil que nous avons utilisé pour cette étape est **Map Reduce**. Les données téléchargées d'origine contiennent des classements de films qui ont été évalués par l'utilisateur pour le film qu'il/elle a visionné. Mais ces données ne contiennent aucune note pour les films qui n'ont pas été notés par l'utilisateur. Notre tâche consiste à attribuer une note de 0 (zéro) aux films qui n'ont pas été notés par l'utilisateur. Cette étape a été réalisée selon les exigences des algorithmes que nous utilisons. Le code des classes Driver, Mapper et Reducer de Map Reduce Job est joint à ce rapport.

Mapper :

Pour le Mapper nous avons travaillé sur le u.data pour rassembler la valeur output (film rating) et pour la clé on garde l'id de l'utilisateur .

Donc le résultat sera le couple (id_user , film rating) .

```
import java.io.IOException;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Mapper;


public class PMapper extends Mapper<Object,Text,Text,Text>
    {
        public void map(Object key,Text value,Context context) throws IOException, InterruptedException
        {
            String ar[] = value.toString().split("\t");
            Text uid = new Text(ar[0]);
            Text val = new Text(ar[1]+"\t"+ar[2]);
            context.write(uid, val);
        }
    }
```

Shuffling :

Pour le shuffle nous allons retenir la résultat :

(id_user , {id_film 1 , rating 1 } , .. , {id_film n , rating n }) n varie entre 1 et 1682

Reducer :

Le **reducer** va regrouper pour chaque user les films évalués et aussi les non évalués .

Alors on va avoir comme résultat : (id_user , ({film 1, rating 1}, … , {film 1682 , rating 1682})).

```
import java.io.IOException;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapreduce.Reducer;

public class PReducer extends Reducer<Text,Text,Text,Text>
    {
        public void reduce(Text key,Iterable<Text> values,Context context) throws IOException, InterruptedException
        {
            int ar[] = new int[1683];
            for(Text val:values)
            {
                String a[] = val.toString().split("\t");
                int it = Integer.parseInt(a[0].toString());
                ar[it] = 1;
                context.write(key, val);
            }
            for(int i=1;i<=1682;i++)
            {
                if(ar[i]==0)
                    context.write(key,new Text(Integer.toString(i)+"\t"+"0"));
            }
        }
    }
```

Driver :

Pour l'exécution du code MapReduce on emploi la fonction **Driver** qui prend comme input notre data et va retourner un fichier " UserItemRating " dans l' HDFS on va l'utiliser dans le traitement de **PIG** .

```
public class PDriver {

    public static void main(String args[]) throws IOException, ClassNotFoundException, InterruptedException
    {
        System.out.println("**In driver Class**");
        Configuration conf = new Configuration();
        Job job = new Job(conf,"ProfileJob");


        job.setJarByClass(PDriver.class);
        job.setMapperClass(PMapper.class);
        job.setReducerClass(PReducer.class);
        job.setCombinerClass(PReducer.class);

        job.setOutputKeyClass(Text.class);
        job.setOutputValueClass(Text.class);

        FileInputFormat.addInputPath(job, new Path(args[0]));
        FileOutputFormat.setOutputPath(job, new Path(args[1]));

        System.exit(job.waitForCompletion(true)?0:1);
    }
```

3) Traitement des données :

Nous avons choisi de travailler avec **Pig**. C'est l'étape la plus importante de tout le projet car elle contient l'implémentation de l'algorithme que nous utilisons. Comme indiqué ci-dessus, nous avons utilisé trois modèles pour générer des recommandations. WewroteonePig Script pour chaque modèle qui prend des données modifiées en entrée de HDFS. Pour chacun des 948 utilisateurs, nous avons recommandé 5 films générés en suivant les algorithmes.
Nous avons utilisé par le script qui correspond à notre modèle choisi : **pig ContentBased.pig**

4) Placement de la sortie sur MYSQL :

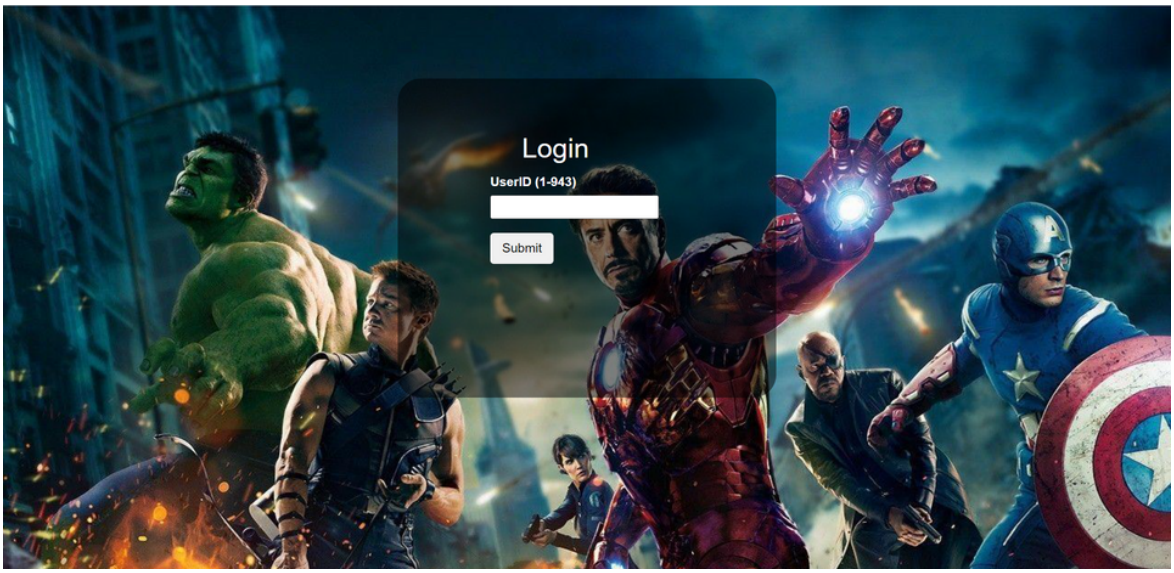
Nous avons opté pour **SQOOP** pour exporter nos données de HDFS vers MYSQL. C'est l'étape qui va nous servir l'exportation de nos données vers MYSQL pour que nous puissions par la suite les utiliser en liaison avec l'interface réaliser dans l'étape suivante.

5) Création de la GUI :

Nous avons utilisé PHP. Décrivant l'environnement industriel et pour montrer notre sortie, nous avons créé une interface graphique en utilisant PHP. Certaines captures d'écran de l'interface graphique sont données à la fin. L'application Web complète est jointe à ce rapport.

Nous avons implémenté une application web qui facilite et aide notre utilisateur à choisir un film ou des films en se basant sur les anciens films évalués par lui-même .

1 . Une interface de Login pour chaque utilisateur : un ID et Mot de passe .



2 . Il le renvoie vers une interface qui affiche les informations de l'utilisateur et les films recommandés pour lui .