



Rapport du Projet de fin d'année

Filère : Génie logiciel

Marketing précis : une approche basée Big Data et Machine learning

Réalisé par :

EL MAHDI Zouhair

HADFI Abdel Moumene

Encadré par :

Pr.NASSAR Mahmoud

Membres du jury :

Pr.NASSAR Mahmoud

Pr.GUERMAH Hatim

Année Universitaire : 2019 - 2020

Remerciements

Nous remercions notre DIEU, qui nous a donné la force, la volonté et le courage pour terminer ce modeste travail. Nous tenons à exprimer d'abord toute nos gratitude à notre encadrant Pr.NASSAR Mahmoud pour sa proposition on de ce projet, pour son encadrement, son écoute, ses élucidations, ses conseils, ses directives et encouragements.

Nous tenons également à remercier profondément tous le cadre professoral de l'ENSIAS pour la formation de la qualité qu'ils nous assurent.

Enfin, Veuillez accepter, Messieurs les membres du jury, *Pr.NASSAR Mahmoud* et *Pr.GUERMAH Hatim* tous nos connaissances et nos gratitude.

Résumé

Aujourd'hui le marketing précis est une méthode de business qui permet aux entreprises d'augmenter leurs revenus jusqu'à 30% . En fait, fournir des publicités pour des produits de cuisine à un enfant naviguant sur des sites de jeux sera généralement inutile, en revanche, lui fournir des publicités pour des produits de jouets augmentera la possibilité d'acheter ce produit. C'est le rôle du marketing précis qui visent à comprendre le comportement de l'utilisateur et lui offrir les meilleures offres.

En effet, de nombreux secteurs utilisent les systèmes intelligents en but de fournir des nouvelles suggestion de produits q'un client pourrait aimer.

Le présent document est la synthèse du travail effectué dans le cadre du projet de fin d'année de 2eme année à l'Ecole nationale supérieur d'informatique et d'analyse des systèmes.

Notre projet consiste en l'exploration et l'implémentation d'un système de recommandation basé sur le Big Data et Machine Learning. Pour la réalisation de ce projets, on a commencé par une étude des différents systèmes de recommandation, puis on a entamé la préparation du données basées sur un Dataset d'utilisateurs et leurs ratings et ensuite réalisation de notre système avec quelques interfaces graphiques.

Mots clés

- Marketing précis
- Systeme de recommandation
- Big Data
- Hadoop
- KNN

Liste d'abréviations

- **HDFS** **H**adoop **D**istributed **F**ile **S**ystem
- **API** **A**pplication **P**rogramming **I**nterface
- **YARN** **Y**et **A**nother **R**esource **N**egotiator
- **RMSE** **R**oot **M**ean **S**quare **E**rror
- **KNN** **k**-NearestNeighbors

TABLE DES FIGURES

1.1	Les 3V du Big Data	9
1.2	Types des systemes de recommandation	10
1.3	Système de recommandation basé sur le contenu	11
1.4	Système de recommandation collaboratif	13
1.5	Système de recommandation hybride	15
2.1	Hadoop distributed file system	17
2.2	Schéma de fonctionnement du MapReduce	18
2.3	Exemple de matrice du rating	19
2.4	graphique d'évaluation de deux produits.	22
2.5	formule de calcule de la distance euclidienne.	22
2.6	graphique des lignes joignant les points à l'origine	23
2.7	programme de calcul du cosinus	24
2.8	formule de calcule de similitude	25
2.9	formule de calcule du rating	25
2.10	formule de calcule du rating à l'aide de la moyenne pondérée	26
3.1	exemple du DATASET	28
3.2	statistique sur utilisateur , produit , rating	29
3.3	statistique sur nombre de rating	29
3.4	utilisateur statistique	33
3.5	produit statistique	33
3.6	rating statistique	34

3.7	fonctionnement du methode KNN	35
3.8	Benchmark fait par surprise	36
3.9	notre Benchmark	36
3.10	Implementation du methode KNN	37
3.11	RMSE du modele KNN	37
3.12	Diagramme de sequence	38
4.1	Flask logo	40
4.2	ReactJs logo	40
4.3	API	41
4.4	Rating Page	42
4.5	Prédiction Page	42

TABLE DES MATIÈRES

Table des figures	2
Introduction générale	6
1 Présentation du projet	7
1.1 Contexte général du projet	8
1.1.1 Problématique :	8
1.1.2 Objectifs :	8
1.2 Le Big Data et le marketing	9
1.3 Les Systèmes de recommandation :	10
1.3.1 Définition :	10
1.3.2 Approche basée sur le contenu :	11
1.3.3 Approche basée sur le filtrage collaboratif :	13
1.3.4 Approche hybride :	15
2 Hadoop et le Filtrage collaboratif pour les systèmes de recommandation	16
2.1 Hadoop pour le Big Data	17
2.1.1 Distribution données et traitements	17
2.1.2 Hadoop File System (HDFS)	17
2.1.3 L'algorithme MapReduce	18
2.2 le concept du filtrage collaboratif	19
2.3 Les étapes du filtrage collaboratif	20
2.4 Le Type du filtrage collaboratif : Memory Based	21

2.4.1	Trouver des utilisateurs similaires	21
2.4.2	Calcul du rating.	25
3	Realisation :	27
3.1	Compréhension du DATASET :	28
3.2	Outils :	30
3.3	HADOOP :	33
3.3.1	Introduction :	33
3.3.2	Utilisateur :	33
3.3.3	Produit :	33
3.3.4	Rating :	34
3.3.5	conclusion :	34
3.4	algorithme k plus proches voisins centre :	35
3.4.1	Théorie :	35
3.4.2	Adoption du methode KNN :	36
3.4.3	Implementation :	37
3.4.4	Evaluation du modèle :	37
3.5	Diagramme de sequence :	38
4	Interface utilisateur graphique	39
4.1	Outils	40
4.1.1	Flask :	40
4.1.2	ReactJs :	40
4.2	Realisation :	41
4.2.1	API :	41
4.2.2	Front-end :	42
	Conclusion générale	44

INTRODUCTION GÉNÉRALE

Avec la facilité d'accès à Internet, nous sommes de plus en plus exposés à une multitude d'informations. Les blogs, les journaux, les sites de commerce etc. apportent beaucoup de diversités aux utilisateurs. Toutefois, avec cette multitude de sources d'informations, la surcharge d'information et les données massives collectées peuvent devenir une problématique. Afin de remédier à ces problèmes, divers outils ont été développés pour filtrer les informations avant de les transmettre aux utilisateurs. Le Big Data et les systèmes de recommandations sont des outils permettant un tel filtrage et traitement de données. Le but principal de ces systèmes est d'analyser les données et faciliter la prise de décisions pour les utilisateurs en leur offrant des informations selon leurs préférences. Il existe actuellement une multitude de systèmes de recommandations pour divers produits tels que des films (Netflix), des livres (Amazon) etc.

Le présent document est articulé en quatre parties : la première présente le contexte général, la problématique et l'objectif du projet. Dans la deuxième partie, nous avons défini l'outil hadoop ainsi que l'approche du filtrage collaboratif, puis la troisième partie est dédiée à la description du modèle utilisé. La quatrième partie est consacrée pour la présentation des interface utilisateur.

CHAPITRE 1

PRÉSENTATION DU PROJET

Introduction

Dans ce premier chapitre, nous présentons le projet dans son cadre général à savoir : le Big Data et les concepts de bases de la recommandation notamment les approches adoptées pour l'élaboration des systèmes de recommandation.

1.1 Contexte général du projet

1.1.1 Problématique :

Le marketing est une démarche visant à satisfaire les désirs et besoins du consommateur, et aussi un état d'esprit et un comportement dont le but est d'adapter l'offre à la demande du client. et cela se fait grâce à des études de clientèle pour définir le produit qui pourrait intéresser les clients. Toute entreprise a une légitimité vis à vis du client, Il ne faut pas sans cesse satisfaire les désirs des clients.

Aujourd'hui, la concurrence est vive sur les marchés, les consommateurs dont le pouvoir d'achat et le niveau d'éducation ont considérablement évolué, automatiquement deviennent exigeants.

Dans ce cadre, l'adaptation du produit à la demande du client à aussi évalué grâce à l'innovation technologique, d'où nombreux entreprises ont déjà commencé à utiliser des systèmes intelligents qui peuvent apporter de nombreux avantages, qui comprend généralement des outils analytiques développés dans le cadre du Big Data, et qui permettent de mettre en place des solutions prédictives, de suivre des tendances en temps réel et de mieux anticiper les risques éventuels liés à l'activité et à la relation client. et aussi un système de recommandation pour garantir et faciliter l'interaction des clients, et aussi pour exploiter leurs avis pour les aider à prendre des décisions.

1.1.2 Objectifs :

Notre projet a pour objectif principal la mise en oeuvre d'un système qui permet de recommander des produits pour l'utilisateur en se basant sur l'extraction des informations pertinentes des utilisateurs, comparer leurs profils, et ensuite chercher à prédire l'avis que donnerait un utilisateur pour un produit.

pour atteindre cet objectif on doit passer par les étapes suivantes :

- Etude et Compréhension des différents systèmes de recommandation.
- Préparation de données.
- Evaluation des différents modèles.
- Réalisation de l'application.

1.2 Le Big Data et le marketing

Le Big Data est un concept permettant de stocker un nombre indicible d'informations sur une base numérique, et représente des enjeux stratégiques regroupés sous la notion des « 3V » :



FIGURE 1.1 – Les 3V du Big Data

- **Volume des données** : La quantité de données traitées.
- **Vitesse de traitement** : Vitesse à laquelle ces données sont traitées, accès en temps réel aux informations par exemple.
- **Variété des données** : Issues des réseaux sociaux, données textes, images, vidéos, etc. . .

L'objectif global du Big Data est de pouvoir analyser rapidement de grands volumes de données aux formats hétérogènes pour en sortir une information utile. La personnalisation des offres devenant un argument inévitable dans le but d'être attractif et encore plus dans le domaine du web, le Big data devient une solution permettant au marketing d'être au plus proche du consommateur et de proposer des offres qu'il sera difficile de retrouver chez la concurrence.

1.3 Les Systèmes de recommandation :

1.3.1 Définition :

Le système de recommandation est un système de filtrage de l'information qui cherche à prédire le rating ou la préférence qu'un utilisateur donnerait à un article ou film, livre, nouvelle. Le système de recommandation a un super avantage soit pour le service qui le fournit ou bien pour l'utilisateur. Car Il réduit les coûts de transaction liés à la recherche et à la sélection d'articles dans un environnement d'achat en ligne.

Il fait partie des systèmes d'apprentissage automatique les plus puissants qui ont pour but stimuler les ventes des grandes surfaces commerciales en ligne.

Les systèmes de recommandation sont classés en fonction de l'approche utilisée pour estimer l'avis d'un utilisateur, sachant qu'ils existent trois méthode :

- méthode basée sur le contenu.
- méthode collaborative ou basée sur le filtrage collaboratif.
- méthode hybride.

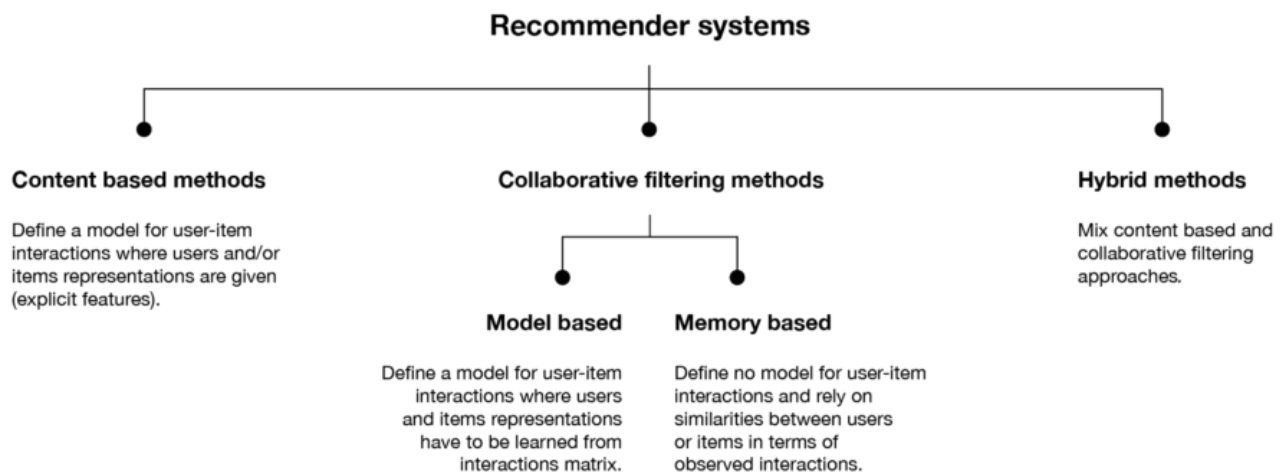


FIGURE 1.2 – Types des systemes de recommandation

1.3.2 Approche basée sur le contenu :

Pour les recommandations basées sur le contenu , la tâche consiste à déterminer quels éléments 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.

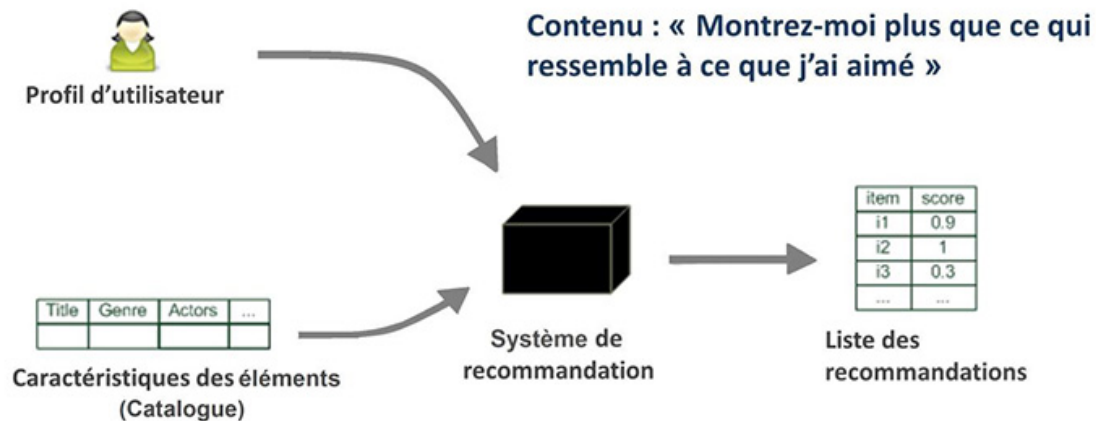


FIGURE 1.3 – Système de recommandation basé sur le contenu

Les systèmes de recommandation basés sur le contenu présentent les avantages suivants :

- recommandent des éléments similaires à ceux que les utilisateurs ont aimés dans le passé.
- avoir les recommandations les plus pertinentes pour chacun des utilisateurs.
- les données relatives aux autres utilisateurs sont inutiles.
- possible de recommander de nouveaux éléments ou même des éléments qui ne sont pas populaires.
- pas de problème de démarrage à froid lorsqu'un nouvel élément est ajouté ou de faible

Et même temps, ces systèmes ont aussi des inconvénients :

- un risque de « sur-spécialisation » apparaît, c'est-à-dire que l'on se limite aux éléments similaires et que les réponses sont trop homogènes.
- pour que le système produise des recommandations précises, l'utilisateur doit fournir un feedback sur les suggestions retournées.
- les éléments représentés par le même ensemble de mots-clés ne peuvent pas être distingués.
- pas d'historique pour un nouvel utilisateur.

1.3.3 Approche 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.

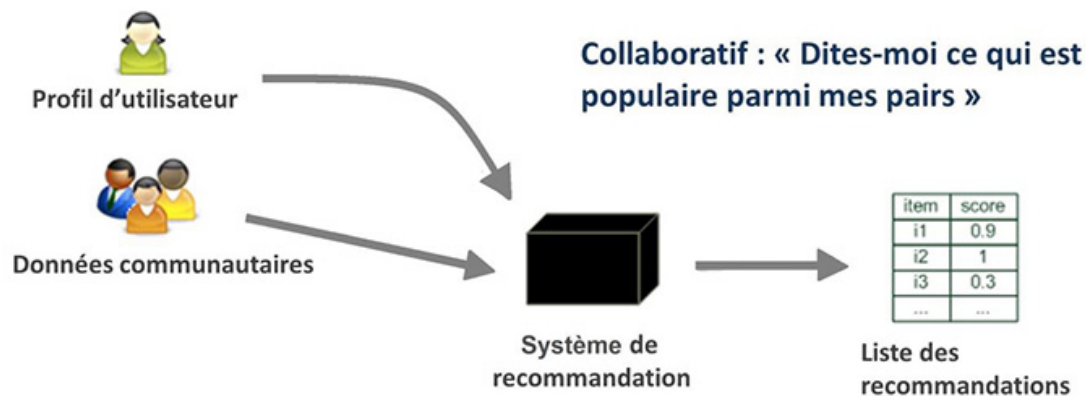


FIGURE 1.4 – Système de recommandation collaboratif

L'idée est d'essayer de prédire l'opinion d'un utilisateur sur les différents éléments en se basant sur les goûts et avis précédents de l'utilisateur et sur une mesure de similarité avec d'autres utilisateurs. Les principales étapes de cette approche sont :

1. collecter les préférences des utilisateurs.
2. un groupe d'utilisateurs est considéré similaires à l'utilisateur qui cherche la recommandation.
3. calculer la moyenne des préférences pour ce groupe.
4. utiliser une fonction de préférence pour recommander des éléments à l'utilisateur.

Les systèmes de recommandation collaboratifs présentent les avantages suivants :

- utiliser les avis d'autres utilisateurs pour évaluer l'utilité des produits.
- trouver des groupes dont les utilisateurs correspondent aux mêmes intérêts.
- l'utilisation des avis de plusieurs utilisateurs donne des meilleurs résultats de recommandation.

En même temps, de tels systèmes ont aussi des inconvénients :

- le regroupement des utilisateurs similaires est difficile.
- le système de recommandation est faible si la matrice Utilisateur X Éléments n'est pas dense
- le calcul croît linéairement lorsque le nombre d'utilisateurs et des éléments est grand.

Nous allons plus détailler cette approche dans un prochain chapitre.

1.3.4 Approche hybride :

Un système de recommandation hybride peut utiliser à la fois des connaissances extérieures et les caractéristiques des éléments, combinant ainsi des approches collaboratives et basées sur le contenu.

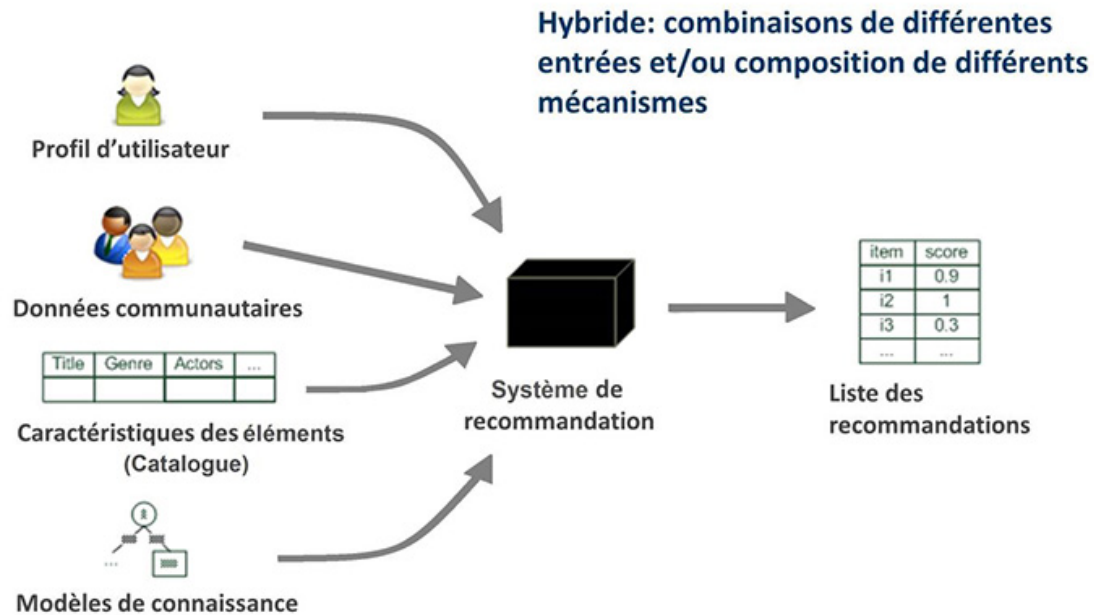


FIGURE 1.5 – Système de recommandation hybride

Conclusion

dans ce chapitre, nous avons bien défini la problématique et notre objectif qui mène à implémenter un système de recommandation. Et dans un deuxième lieu on a expliqué la relation entre le Big Data et le marketing. Ensuite nous avons abordé les différents types de systèmes de recommandation avant de se concentrer sur la méthode collaborative dans un autre chapitre.

CHAPITRE 2

HADOOP ET LE FILTRAGE COLLABORATIF POUR LES SYTÈMES DE RECOMMANDATION

Introduction

le Big Data et le filtrage collaboratif est la technique la plus utilisée pour le traitement d'aussi grandes quantités de donnée et créer des systèmes de recommandation intelligents qui peuvent apprendre à donner de meilleures recommandations à mesure d'avoir davantage d'informations sur les utilisateurs. dans ce chapitre nous allons présenter en détails le fonctionnement de ce type des systèmes intelligents.

2.1 Hadoop pour le Big Data

2.1.1 Distribution données et traitements

Le traitement d'aussi grandes quantités de données impose des méthodes particulières. Un SGBD classique, même haut de gamme, est dans l'incapacité de traiter autant d'informations. C'est pour cela, il sera indispensable de répartir les données sur plusieurs machines (jusqu'à plusieurs millions d'ordinateurs) dans des Data Centers.

et pour cette raison, il a appris Hadoop qui est un système de gestion de données et de traitements distribués. Il contient de beaucoup de composants, dont :

- **HDFS** : un système de fichier qui répartit les données sur de nombreuses machines.
- **YARN** : un mécanisme d'ordonnancement de programmes de type MapReduce.

On va d'abord présenter HDFS puis MapReduce.

2.1.2 Hadoop File System (HDFS)

Le HDFS est un système de fichiers distribué, extensible et portable développé par Hadoop à partir du GoogleFS. Écrit en Java, il a été conçu pour stocker de très gros volumes de données sur un grand nombre de machines équipées de disques durs banalisés. Il permet l'abstraction de l'architecture physique de stockage, afin de manipuler un système de fichiers distribué comme s'il s'agissait d'un disque dur unique.

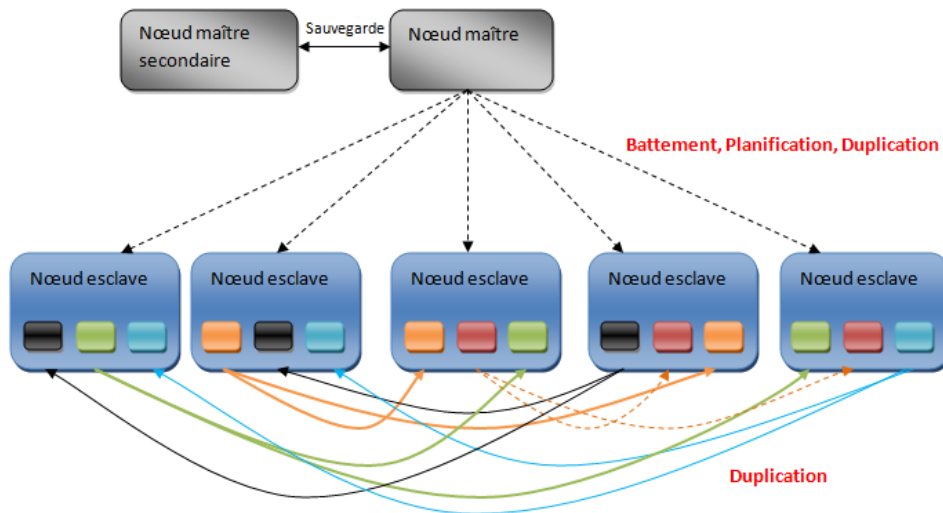


FIGURE 2.1 – Hadoop distributed file system

HDFS est un système de fichiers distribué. C'est à dire :

- les fichiers et dossiers sont organisés en arbre (comme Unix).
- ces fichiers sont stockés sur un grand nombre de machines de manière à rendre invisible la position exacte d'un fichier. L'accès est transparent, quelle que soient les machines qui contiennent les fichiers.
- les fichiers sont copiés en plusieurs exemplaires pour la fiabilité et permettre des accès simultanés multiples

HDFS permet de voir tous les dossiers et fichiers de ces milliers de machines comme un seul arbre, contenant des Po de données, comme s'ils étaient sur le disque dur local. HDFS permet de voir tous les dossiers et fichiers de ces milliers de machines comme un seul arbre, contenant des données, comme s'ils étaient sur le disque dur local.

2.1.3 L'algorithme MapReduce

MapReduce permet de manipuler de grandes quantités de données en les distribuant dans un cluster de machines pour être traitées. Ce modèle connaît un vif succès auprès de sociétés possédant d'importants centres de traitement de données.

MapReduce est principalement utilisé pour la manipulation et le traitement d'un nombre important de données au sein d'un cluster de nœuds.

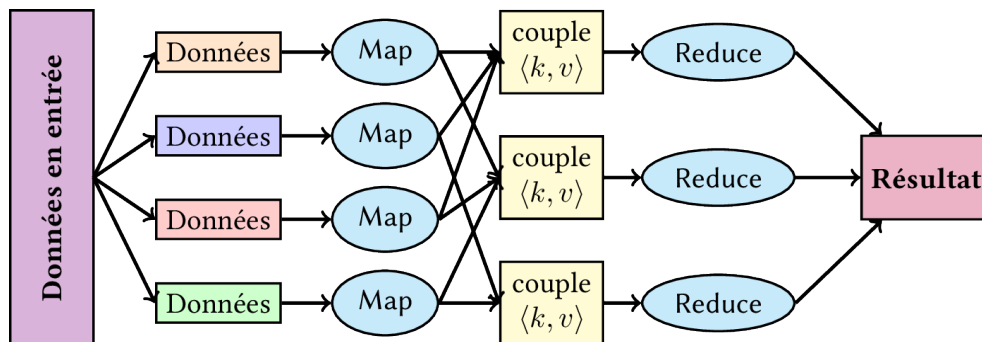


FIGURE 2.2 – Schéma de fonctionnement du MapReduce

2.2 le concept du filtrage collaboratif

Le filtrage collaboratif est une technique qui peut filtrer les éléments qu'un utilisateur pourrait aimer sur la base des réactions d'utilisateurs similaires. Il fonctionne en recherchant un grand groupe de personnes et en trouvant un plus petit ensemble d'utilisateurs avec des goûts similaires à un utilisateur particulier. Il examine les éléments qu'ils aiment et les combine pour créer une liste classée de suggestions.

Il existe de nombreuses façons de décider quels utilisateurs sont similaires et de combiner leurs choix pour créer une liste de recommandations.

Pour tester développer les algorithmes de recommandation, nous avons besoin de données contenant un ensemble d'éléments et un ensemble d'utilisateurs qui ont réagi à certains éléments.

La réaction peut être explicite (évaluation sur une échelle de 1 à 5, j'aime ou n'aime pas) ou implicite (visualiser un article, l'ajouter à une liste de souhaits, le temps passé sur un article...).

	i_1	i_2	i_3	i_4	i_5
u_1	5		4	1	
u_2		3		3	
u_3		2	4	4	1
u_4	4	4	5		
u_5	2	4		5	2

FIGURE 2.3 – Exemple de matrice du rating

Dans la plupart des cas, les cellules de la matrice sont vides, car les utilisateurs ne notent que quelques éléments. Il est très peu probable que chaque utilisateur évalue ou réagisse à chaque élément disponible. Une matrice avec des cellules principalement vides est appelée **clairsemée** (sparse matrix), et l'opposé de celle est appelée **dense** (dense matrix).

2.3 Les étapes du filtrage collaboratif

Pour créer un système capable de recommander automatiquement des éléments aux utilisateurs en fonction des préférences des autres utilisateurs, il faut passer par les deux étapes suivantes :

- **la première étape** consiste à rechercher des utilisateurs ou des éléments similaires.
- **La deuxième étape** consiste à prédire les notes des articles qui ne sont pas encore notées par un utilisateur.

Donc, nous aurons besoin des réponses à ces questions :

- Comment déterminer les utilisateurs ou les éléments similaires ?
- Comment prédire le rating qu'un utilisateur attribuait à un élément en fonction des notes d'utilisateurs similaires ?
- Comment mesurez-vous l'exactitude des rating calculés ?

Les deux premières questions n'ont pas de réponse unique. Le filtrage collaboratif est une famille d'algorithmes où il existe plusieurs façons de trouver des utilisateurs ou des éléments similaires et plusieurs façons de calculer la note en fonction des notes d'utilisateurs similaires. Selon les choix que nous faisons, nous nous retrouvons avec un type d'approche de filtrage collaboratif.

Une chose importante à garder à l'esprit est que dans une approche basée uniquement sur le filtrage collaboratif, la similitude n'est pas calculée à l'aide de facteurs tels que l'âge des utilisateurs, le genre ou toute autre donnée sur les utilisateurs ou les éléments. Il est calculé uniquement sur la base de la notation qu'un utilisateur attribue à un article.

La troisième question sur la façon de mesurer la précision des prédictions a également plusieurs réponses, qui incluent des techniques de calcul d'erreur qui peuvent être utilisées à de nombreux endroits et pas seulement des recommandations basées sur le filtrage collaboratif.

L'une des approches pour mesurer la précision de votre résultat est l'erreur quadratique moyenne (RMSE), dans laquelle vous prédisiez les évaluations d'un ensemble de données de test de paires utilisateur-élément dont les valeurs d'évaluation sont déjà connues. La différence entre la valeur connue et la valeur prédite serait l'erreur.

2.4 Le Type du filtrage collaboratif : Memory Based

La catégorie de ce type comprend des algorithmes basés sur la mémoire, dans lesquels des techniques statistiques sont appliquées à l'ensemble des données pour calculer les prédictions.

Pour trouver la note R qu'un utilisateur U donnerait à un élément I , l'approche comprend :

- Recherche d'utilisateurs similaires à U qui ont évalué l'élément I .
- Calcul de la note R en fonction des notes des utilisateurs trouvées à l'étape précédente

2.4.1 Trouver des utilisateurs similaires

Pour comprendre le concept de similitude, nous donnons un exemple de données simple.

Les données incluent quatre utilisateurs A , B , C et D , qui ont évalué deux produits. Les classements sont stockés dans des listes, et chaque liste contient deux chiffres indiquant le classement de chaque produit :

- évaluation de A est $[1, 2]$.
- évaluation de B est $[2, 4]$.
- évaluation de C est $[2.5, 4]$.
- évaluation de D est $[4.5, 5]$.

Pour commencer avec un indice visuel, nous traçons les notes de deux produits données par les utilisateurs sur un graphique. Le graphique ressemble à ceci :

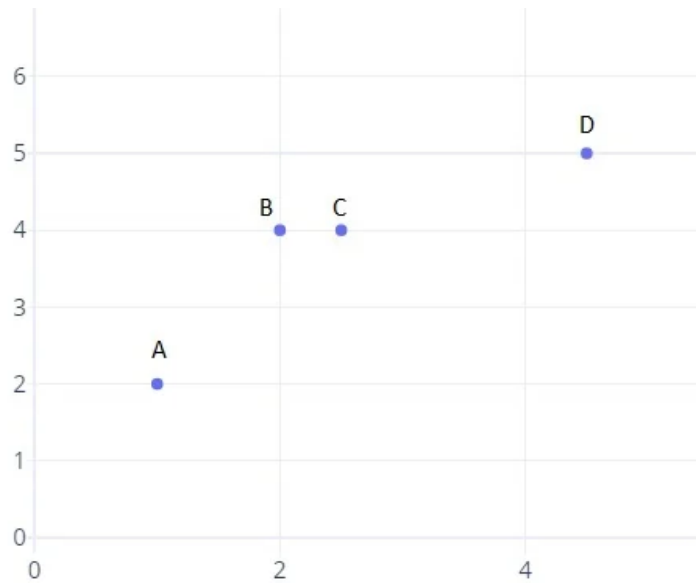


FIGURE 2.4 – graphique d'évaluation de deux produits.

en regardant la distance entre les points semble être un bon moyen d'estimer la similitude, nous pouvons trouver la distance en utilisant la formule de la distance euclidienne entre deux points.

$$AB = \sqrt{(x_B - x_A)^2 + (y_B - y_A)^2}.$$

FIGURE 2.5 – formule de calcul de la distance euclidienne.

Comme indiqué ci-dessus, nous pouvons utiliser la formule pour calculer la distance entre deux points.

- AC=2.5
- BC=0.5.
- DC=2.23606797749979.

nous pouvons voir que l'utilisateur C est le plus proche de B, même en regardant le graphique. Mais sur A et D seulement, qui est le plus proche de C ?

On pourrait dire que C est plus proche de D en termes de distance. Mais en regardant le classement, il semblerait que les choix de Cs s'alignaient sur celui de A plus que D. Alors, on peut utiliser l'angle entre les lignes joignant les points à l'origine pour identifier de tels modèles que la distance euclidienne ne peut pas.

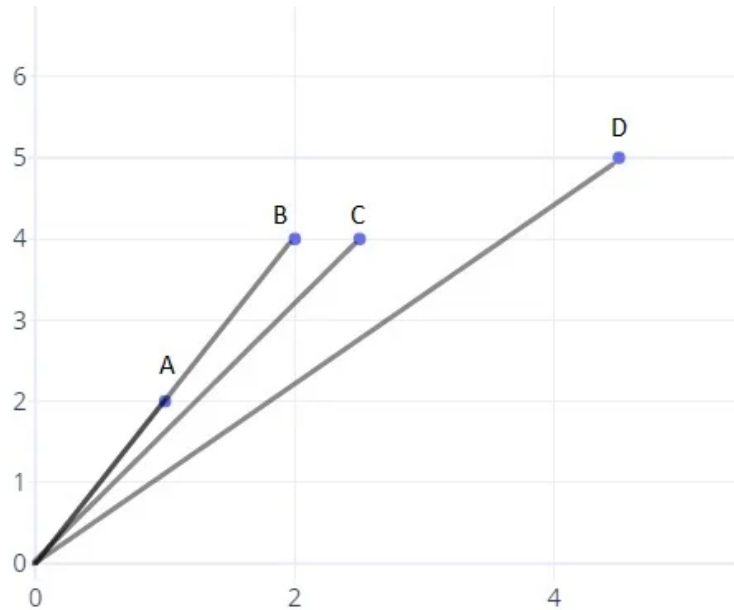


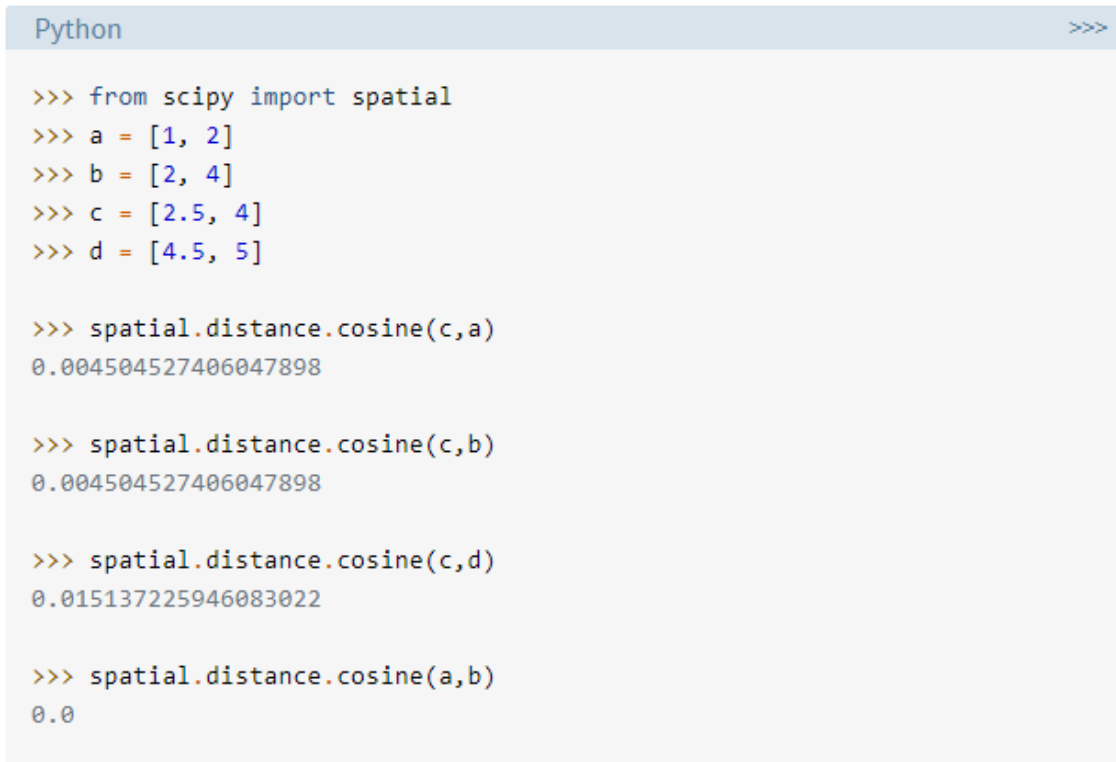
FIGURE 2.6 – graphique des lignes joignant les points à l'origine .

Le graphique montre quatre lignes joignant chaque point à l'origine. Les lignes pour A et B coïncident, rendant l'angle entre elles nul. Nous pouvons considérer que si l'angle entre les lignes augmente, la similitude diminue et si l'angle est nul, les utilisateurs sont très similaires.

Pour calculer la similitude à l'aide d'un angle, nous avons besoin d'une fonction qui renvoie une similitude supérieure ou une distance plus petite pour un angle inférieur et une similitude inférieure ou une distance plus grande pour un angle supérieur. Le cosinus d'un angle est une fonction qui diminue de 1 à -1 lorsque l'angle augmente de 0 à 180.

Nous pouvons utiliser le cosinus de l'angle pour trouver la similitude entre deux utilisateurs. Plus l'angle est grand, plus le cosinus sera faible et donc, plus la similitude des utilisateurs sera faible. Nous pouvons inverser la valeur du cosinus de l'angle pour obtenir la distance cosinus entre les utilisateurs en la soustrayant de 1 .

pour l'exemple ci-dessus, nous calculons le cosinus à l'aide d'un programme python :



```
Python >>>

>>> from scipy import spatial
>>> a = [1, 2]
>>> b = [2, 4]
>>> c = [2.5, 4]
>>> d = [4.5, 5]

>>> spatial.distance.cosine(c,a)
0.004504527406047898

>>> spatial.distance.cosine(c,b)
0.004504527406047898

>>> spatial.distance.cosine(c,d)
0.015137225946083022

>>> spatial.distance.cosine(a,b)
0.0
```

FIGURE 2.7 – programme de calcul du cosinus .

L'angle inférieur entre les vecteurs de C et A donne une valeur de distance cosinus inférieure. la classification les similitudes des utilisateurs de cette manière, utilise la distance cosinus.

Dans l'exemple ci-dessus, seuls deux produits sont pris en compte, ce qui facilite la visualisation des vecteurs de notation en deux dimensions. Cela n'est fait que pour rendre l'explication plus facile.

Les cas d'utilisation réels avec plusieurs éléments impliquent plus de dimensions dans les vecteurs d'évaluation. C'est pour cela, il faut utiliser la formule suivante :

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}},$$

FIGURE 2.8 – formule de calcul de similitude

2.4.2 Calcul du rating.

Après avoir déterminé une liste d'utilisateurs similaires à un utilisateur U, nous devons calculer la note R que U attribuait à un certain élément I.

Nous pouvons prédire que la note R d'un utilisateur pour un élément I sera proche de la moyenne des notes attribuées à I par les 5 premiers utilisateurs les plus similaires à U. La formule mathématique pour la note moyenne donnée par n utilisateurs ressemblerait à :

$$R_U = \left(\sum_{u=1}^n R_u \right) / n$$

FIGURE 2.9 – formule de calcul du rating

Cette formule montre que la note moyenne donnée par les n utilisateurs similaires est égale à la somme des notes données par eux divisée par le nombre d'utilisateurs similaires, qui est n.

Il y aura des situations où les n utilisateurs similaires que nous avons trouvés ne sont pas également similaires à l'utilisateur cible U . Les 3 premiers d'entre eux peuvent être très similaires et les autres peuvent ne pas être aussi similaires à U que les 3 premiers. Dans ce cas, nous pouvons envisager une approche dans laquelle la note de l'utilisateur le plus similaire importe plus que celle du deuxième utilisateur le plus similaire. La moyenne pondérée peut nous aider à y parvenir.

Dans l'approche moyenne pondérée, nous multiplions chaque note par un facteur de similitude qui indique à quel point les utilisateurs sont similaires. En multipliant avec le facteur de similitude, nous ajoutons des pondérations aux notes. Plus le poids est lourd, plus la note importe.

Le facteur de similitude, qui agirait comme des poids, devrait être l'inverse de la distance discutée ci-dessus, car moins de distance implique une similitude plus élevée. Par exemple, vous pouvez soustraire la distance cosinus de 1 pour obtenir une similitude cosinus.

Avec le facteur de similitude S pour chaque utilisateur similaire à l'utilisateur cible U , nous pouvons calculer la moyenne pondérée à l'aide de cette formule :

$$R_U = (\sum_{u=1}^n R_u * S_u) / (\sum_{u=1}^n S_u)$$

FIGURE 2.10 – formule de calcul du rating à l'aide de la moyenne pondérée

Dans la formule ci-dessus, chaque note est multipliée par le facteur de similitude de l'utilisateur qui a donné la note. La note finale prévue par l'utilisateur U sera égale à la somme des notes pondérées divisée par la somme des poids.

Conclusion

dans ce chapitre, nous avons appris comment fonctionne Hadoop pour le traitement des données massives, ainsi que le modèle du filtrage collaboratif qui mène à implémenter un système de recommandation en se basant sur des utilisateurs similaires. Dans le chapitre suivant, nous allons élaborer la phase de réalisation.

CHAPITRE 3

REALISATION :

Introduction

Après le traitement de données avec HADOOP et la réalisation du modèle de prédiction maintenant c'est l'étape de réaliser une interface utilisateur .

3.1 Compréhension du DATASET :

On a choisit de travailler avec AMAZON REVIEW DATA qui se compose des attributs suivant :

- reviewerID - ID of the reviewer,
- asin - ID of the product,
- reviewerName - name of the reviewer
- helpful - helpfulness rating of the review,
- reviewText - text of the review
- overall - rating of the product
- summary - summary of the review
- unixReviewTime - time of the review (unix time)
- reviewTime - time of the review (raw)

```
{
  "reviewerID": "A2SUAM1J3GNN3B",
  "asin": "0000013714",
  "reviewerName": "J. McDonald",
  "helpful": [2, 3],
  "reviewText": "I bought this for my husband who plays the piano.
He is having a wonderful time playing these old hymns. The music is
at times hard to read because we think the book was published for
singing from more than playing from. Great purchase though!",
  "overall": 5.0,
  "summary": "Heavenly Highway Hymns",
  "unixReviewTime": 1252800000,
  "reviewTime": "09 13, 2009"
}
```

FIGURE 3.1 – exemple du DATASET

On a travaillé seulement avec les attributs :

- reviewerID - ID of the reviewer,
- asin - ID of the product,
- overall - rating of the product

Voici des statistiques sur DATASET

Total data

```
-----  
Total No of ratings : 9472  
Total No of Unique Users   : 85  
Total No of Unique products : 4450
```

FIGURE 3.2 – statistique sur utilisateur , produit , rating

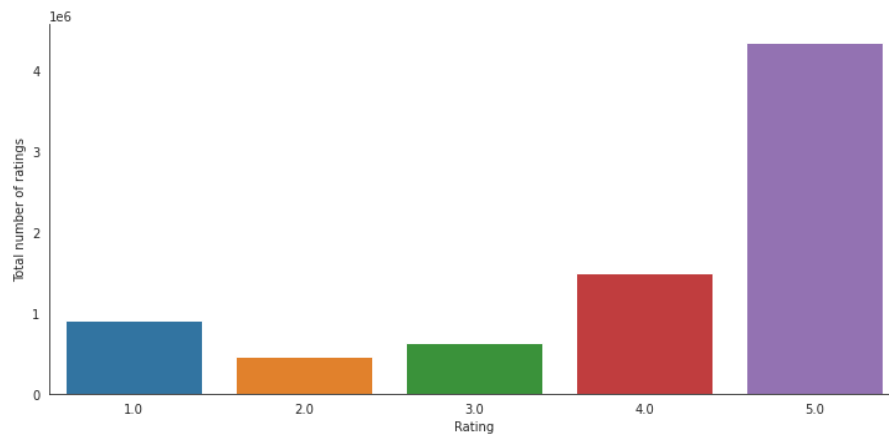


FIGURE 3.3 – statistique sur nombre de rating

3.2 Outils :

L'implémentation du modèle s'est faite en utilisant un ensemble de technologies dont nous citons :

Python



Python est le langage de programmation open source le plus employé par les informaticiens. Ce langage s'est propulsé en tête de la gestion d'infrastructure, d'analyse de données ou dans le domaine du développement de logiciels. En effet, parmi ses qualités, Python permet notamment aux développeurs de se concentrer sur ce qu'ils font plutôt que sur la manière dont ils le font. Il a libéré les développeurs des contraintes de formes qui occupaient leur temps avec les langages plus anciens. Ainsi, développer du code avec Python est plus rapide qu'avec d'autres langages.

Google Colab



Google Colab ou Colaboratory est un service cloud, offert par Google (gratuit), basé sur Jupyter Notebook et destiné à la formation et à la recherche dans l'apprentissage automatique. Cette plateforme permet d'entraîner des modèles de Machine Learning directement dans le cloud. Sans donc avoir besoin d'installer quoi que ce soit sur notre ordinateur à l'exception d'un navigateur.

NumPy



NumPy est une extension du langage de programmation Python, destinée à manipuler des matrices ou tableaux multidimensionnels ainsi que des fonctions mathématiques opérant sur ces tableaux. Plus précisément, cette bibliothèque logicielle libre et open source fournit de multiples fonctions permettant notamment de créer directement un tableau depuis un fichier ou au contraire de sauvegarder un tableau dans un fichier, et manipuler des vecteurs, matrices et polynômes.

Surprise

surprise

Surprise est une bibliothèque Python qui construit et analyse des systèmes de recommandation qui traitent des données de notation explicites.

Surprise a été conçu pour les objectifs suivants :

- Donner aux utilisateurs un contrôle parfait sur leurs expériences.
- Fournir divers algorithmes de prédiction prêts à l'emploi tels que des algorithmes de base, des méthodes de voisinage, basés sur la factorisation matricielle .
- Facilitez la mise en œuvre de nouvelles idées d'algorithmes.
- Fournir des outils pour évaluer, analyser et comparer les performances des algorithmes.

Pandas



Pandas est une bibliothèque écrite pour le langage de programmation Python permettant la manipulation et l'analyse des données. Elle propose en particulier des structures de données et des opérations de manipulation de tableaux numériques et de séries temporelles. Les principales structures de données sont les séries, les DataFrames et les Panels.

3.3 HADOOP :

3.3.1 Introduction :

Comme on a déjà expliqué dans le chapitre précédent HADOOP se base sur l'algorithme Map Reduce pour analyser les données . dans cette partie on va citer

3.3.2 Utilisateur :

Depuis notre DATASET on génère une DATA qui contient l'ID d'utilisateur et combien de fois cet utilisateur a fait un rating .

A11KZ906QD08C5	104
A12DQZKRKTNF5E	129
A17BUUB0U0598B	123
A18HE80910BTZI	91
A19W47CXJJP1MI	111

FIGURE 3.4 – utilisateur statistique

3.3.3 Produit :

Depuis notre DATASET on génère une DATA qui contient l'ID de produit et combien de fois ce produit a eu un rating .

B00000J1EQ	2
B00000J1UQ	1
B00000J1V5	2
B00000J4FS	1
B00000K135	3

FIGURE 3.5 – produit statistique

3.3.4 Rating :

Depuis notre DATASET on génère une DATA qui contient le rating et combien de fois il est repeter .

1.0	226
2.0	322
3.0	774
4.0	2739
5.0	5411

FIGURE 3.6 – rating statistique

3.3.5 conclustion :

Ces données généré par HADOOP ils vont être utilisée soit par système de recommandation ou par application web.

3.4 algorithme k plus proches voisins centre :

3.4.1 Théorie :

Les k plus proches voisins est une méthode non paramétrique dans laquelle le modèle mémorise les observations de l'ensemble d'apprentissage pour la classification des données de l'ensemble de test.

En effet, Pour prédire la classe d'une nouvelle donnée d'entrée, il va chercher ses K voisins les plus proches (en utilisant la distance euclidienne, ou autres) et choisira la classe des voisins majoritaires.

Pour appliquer cette méthode, les étapes à suivre sont les suivantes :

1. On fixe le nombre de voisins k.
2. On détecte les k-voisins les plus proches des nouvelles données d'entrée que l'on veut classer.
3. On attribue les classes correspondantes par vote majoritaire.

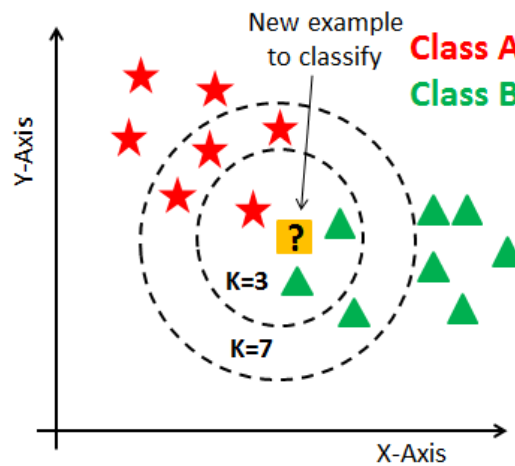


FIGURE 3.7 – fonctionnement du methode KNN

3.4.2 Adoption du methode KNN :

On a choisi d'adopter ce modèle à la base de se benchmark , qu'est fait sur un Intel Core i5 7th gen (2.5 GHz) avec 8Go RAM , sur un DATASET de 1 million.

Movielens 1M	RMSE	MAE	Time
SVD	0.873	0.686	0:02:13
SVD++	0.862	0.673	2:54:19
NMF	0.916	0.724	0:02:31
Slope One	0.907	0.715	0:02:31
k-NN	0.923	0.727	0:05:27
Centered k-NN	0.929	0.738	0:05:43
k-NN Baseline	0.895	0.706	0:05:55
Co-Clustering	0.915	0.717	0:00:31
Baseline	0.909	0.719	0:00:19
Random	1.504	1.206	0:00:19

FIGURE 3.8 – Benchmark fait par surprise

et ce benchmark qu'on a fait sur notre DATASET

	test_rmse	fit_time	test_time
Algorithm			
BaselineOnly	0.878706	0.012734	0.013251
SVD	0.883038	0.370465	0.019038
SVDpp	0.883460	6.301008	0.185200
KNNBaseline	0.967598	0.015092	0.041624
KNNWithMeans	0.975625	0.004045	0.031318
KNNWithZScore	0.987437	0.008417	0.057643
KNNBasic	1.012712	0.002013	0.028293
CoClustering	1.023518	0.402045	0.015021
SlopeOne	1.049080	0.303288	0.094094
NMF	1.094583	0.554745	0.018506
NormalPredictor	1.331251	0.007581	0.023854

FIGURE 3.9 – notre Benchmark

même si on a un RMSE qu'est un peut grand mais le temps pour faire la prédiction est petit , pour cela on a choisit d'adopter cet methode .

3.4.3 Implementation :

Pour Implementation on a choisit d'utiliser la librairie surprise , fonction de prédiction et configurer d'utiliser la méthode similarité cousine pour trouver les similaires item.

```
algo = surprise.KNNWithMeans(k=4, sim_options={'name': 'cosine', 'user_based': False})  
train_pred = algo.fit(trainset)  
test_pred = train_pred.test(testset)
```

FIGURE 3.10 – Implementation du methode KNN

3.4.4 Evaluation du modèle :

Pour mesurer la précision du modèle , on a choisi de travailler avec le paramètre RMSE, et voici les résultats :

```
Item-based Model : Recommender  
RMSE: 1.0107  
1.0106502286432053
```

FIGURE 3.11 – RMSE du modele KNN

3.5 Diagramme de sequence :

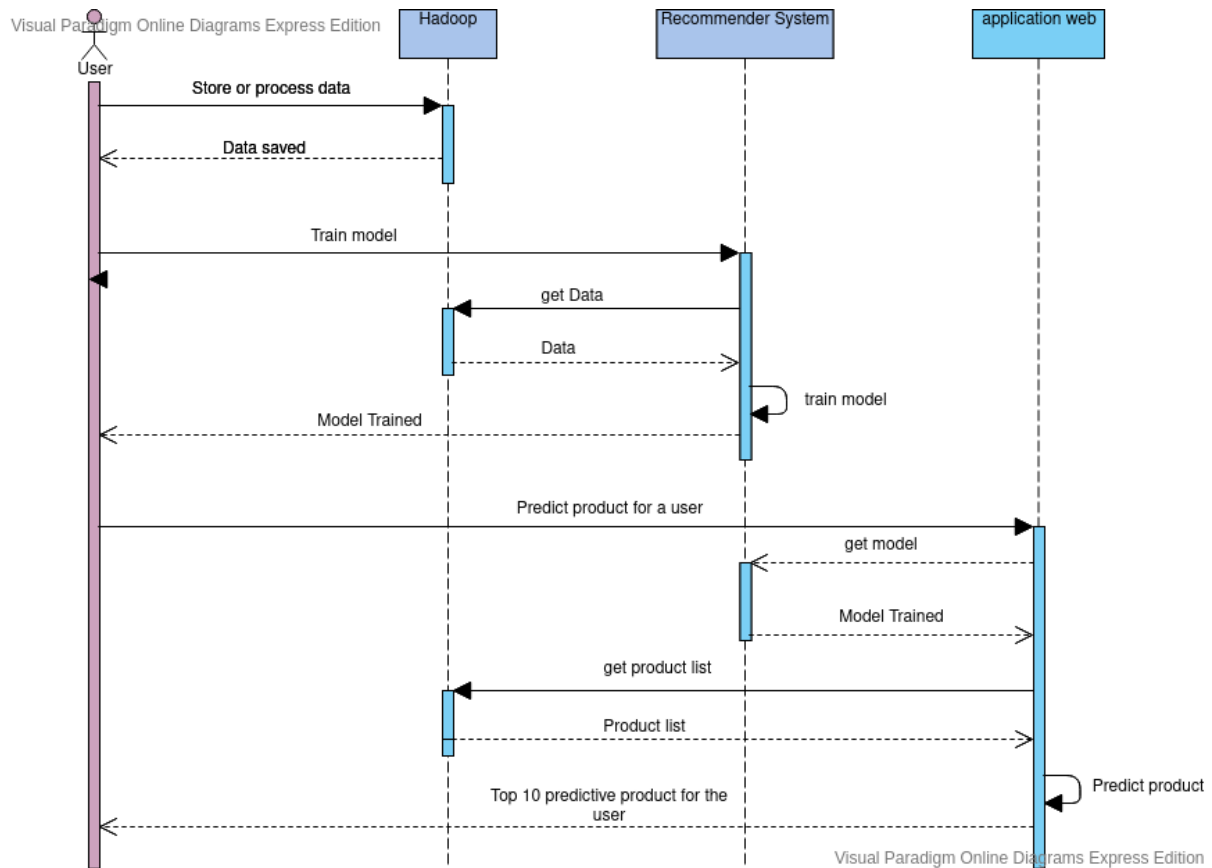


FIGURE 3.12 – Diagramme de sequence

Conclusion

dans ce chapitre, nous avons parler sur les outils utilisé pour élaborer ce système , puis sur programme mapreduce qui ne permet d'analyser notre DATASET , et enfin la méthode KNN pour réaliser le système de recommandation .

CHAPITRE 4

INTERFACE UTILISATEUR GRAPHIQUE

Introduction

Après le traitement de données avec HADOOP et la réalisation du modèle de prédiction maintenant c'est l'étape de réaliser une interface utilisateur .

4.1 Outils

4.1.1 Flask :

Flask est un framework open-source de développement web en Python. Son but principal est d'être léger, afin de garder la souplesse de la programmation Python, associé à un système de templates.



FIGURE 4.1 – Flask logo

4.1.2 ReactJs :

ReactJs est une bibliothèque JavaScript libre développée par Facebook depuis 2013. Le but principal de cette bibliothèque est de faciliter la création d'application web monopage, via la création de composants dépendant d'un état et générant une page (ou portion) HTML à chaque changement d'état.



FIGURE 4.2 – ReactJs logo

4.2 Realisation :

4.2.1 API :

On a utilisée framework Flask pour créer une API qui va recevoir les HTTP requests et va les traites puis envoyer les réponses.

```
@app.route('/pred_product',methods=['POST'])
@cross_origin()
def pred_product():
    if not request.json or not 'Id' in request.json :
        abort(400)

    if not find_user(request.json["Id"],userl) :
        return "user not found",404
    return jsonify(predict_user(request.json["Id"],productl))

@app.route('/rating',methods=['POST'])
@cross_origin()
def rating_func():
    return jsonify(rating)

@app.route('/product',methods=['POST'])
@cross_origin()
def top_product():
    return jsonify(product_top(productl))
```

FIGURE 4.3 – API

4.2.2 Front-end :

On a utilisée Reactjs pour créer le FRONT du application web qui va envoyer les requests vers API et représenter les réponses du API.

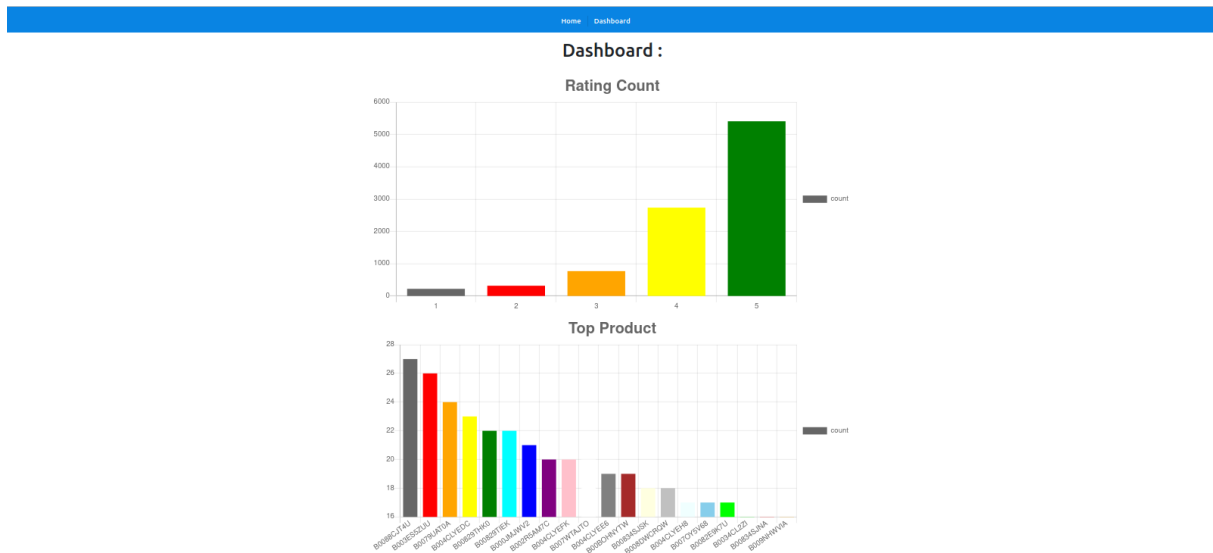


FIGURE 4.4 – Rating Page

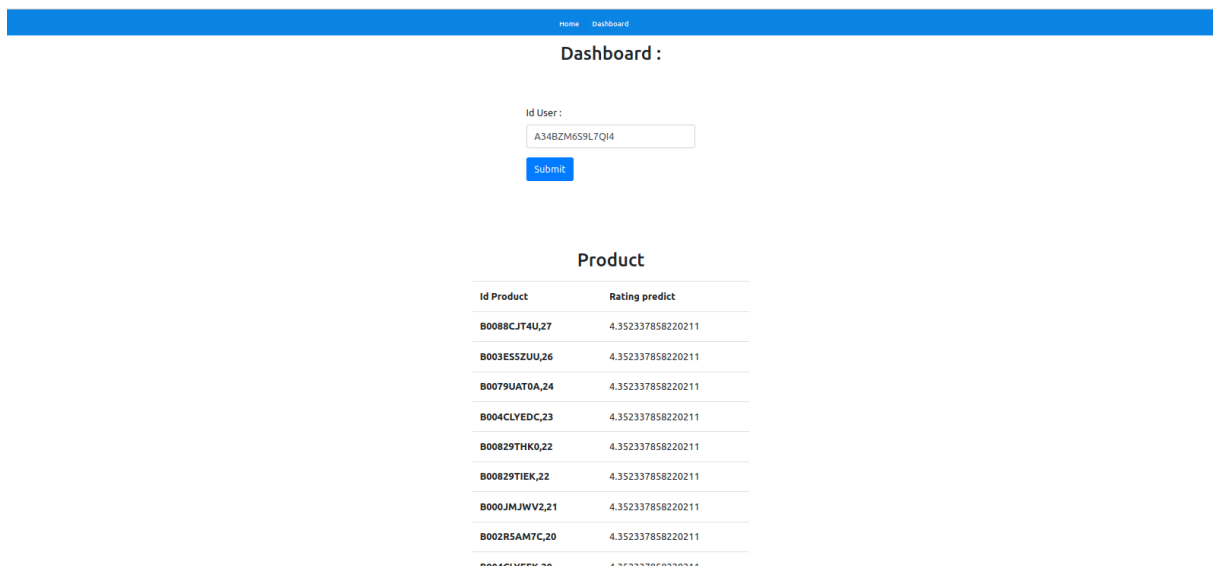


FIGURE 4.5 – Prédiction Page

Conclusion

dans ce chapitre, nous avons donné l'architecture d'application web et aussi les outils utilisé .

CONCLUSION GÉNÉRALE

Dans le cadre du projet de fin de 2ème année à l'Ecole National supérieur d'informatique et d'analyse des systèmes filière GL, on a réalisé un projet qui a pour but d'explorer et d'implémenter une approche des systèmes de recommandation basé sur le Big Data et le filtrage collaboratif. Pour atteindre ces objectifs, on a décidé d'utiliser le framework Hadoop qui a permis de dépasser les obstacles que posent la complexité du traitement des données massives d'une manière automatique.

Durant ce projet on a pu acquérir plusieurs connaissances et compétences comme l'extraction et le filtrage des informations de littérature, le travail en équipe, le management du temps et aussi approfondir et enrichir nos connaissances en sciences des données Big Data et Machine Learning, ainsi que le marketing précis.

BIBLIOGRAPHIE ET WEBOGRAPHIE

- [1] [En ligne] (Machine Learning for Recommender systems) Disponible sur :
<https://medium.com/recombee-blog/machine-learning-for-recommender-systems>
<http://surpriselib.com>
<https://realpython.com/build-recommendation-engine-collaborative-filtering>
- [2] [En ligne] (Collaborative Filtering) Disponible sur :
<https://medium.com/sfu-csmp/recommendation-systems-user-based-collaborative-filtering-using-n-nearest-neighbors>
<https://medium.com/datadriveninvestor/k-nearest-neighbors-knn>
- [3] [En ligne] (Hadoop) Disponible sur :
<https://www.udacity.com/course/intro-to-hadoop-and-mapreduce>
<https://clubhouse.io/developer-how-to/how-to-set-up-a-hadoop-cluster-in-docker>
<https://medium.com/better-programming/what-is-hadoop>
- [4] [En ligne] (github) Disponible sur :
<https://github.com/AbdelMoumene-Hadfi/Product-Recommendation>