

**Projet en big data**

Réalisation d’une application machine learning permet de choisir le meilleur produit avec mahout et oozie

**Réalise Par :**

Elhoussine AIT EZZOUINE

Youssef ELMOUMEN

**Encadré par :**

Professeur Ahmed EZZATIE

Anne scolaire : 2019-2020

Table des matières

[II. Introduction : 4](#_Toc36131570)

[III. Introduction à Mahout : 5](#_Toc36131571)

[1. Définition : 5](#_Toc36131572)

[2. Caractéristiques de Mahout 5](#_Toc36131573)

[3. Applications de Mahout 5](#_Toc36131574)

[4. Mahout – Recommendation 6](#_Toc36131575)

[a) Définition 6](#_Toc36131576)

[b) Moteur de recommandation Mahout 7](#_Toc36131577)

[5. Mahout-Clustering 8](#_Toc36131578)

[a) Applications du clustering 8](#_Toc36131579)

[b) Procédure de clustering 8](#_Toc36131580)

[6. Mahout – Classification 9](#_Toc36131581)

[a) Définition 9](#_Toc36131582)

[b) Applications de la classification 9](#_Toc36131583)

[c) Naive Bayes Classifier 9](#_Toc36131584)

[d) Procédure de classement 10](#_Toc36131585)

[IV. Introduction à Oozie: 10](#_Toc36131586)

[1. Définition 10](#_Toc36131587)

[2. Cas d'utilisation d'Apache Oozie 11](#_Toc36131588)

[3. Editeurs d'Oozie 11](#_Toc36131589)

[a) Éditeur de teintes pour Oozie 11](#_Toc36131590)

[b) Plugin Oozie Eclipse (OEP) 12](#_Toc36131591)

[4. Oozie workflow 12](#_Toc36131592)

[5. Cordinators 13](#_Toc36131593)

[6. Bundle 13](#_Toc36131594)

[7. Outils de ligne de commande 13](#_Toc36131595)

[V. Projet en Maout-oozie : 15](#_Toc36131596)

# Introduction :

Nous vivons à une époque où les informations sont disponibles en abondance. La surcharge d'informations a atteint des niveaux tels qu'il devient parfois difficile de gérer nos petites boîtes aux lettres! Imaginez le volume de données et d'enregistrements que certains des sites Web populaires (comme Facebook, Twitter et Youtube) doivent collecter et gérer quotidiennement. Il n'est pas rare, même pour des sites Web moins connus, de recevoir d'énormes quantités d'informations en vrac.

Normalement, nous nous appuyons sur des algorithmes d'exploration de données pour analyser des données en vrac afin d'identifier les tendances et de tirer des conclusions. Cependant, aucun algorithme d'exploration de données ne peut être suffisamment efficace pour traiter de très grands ensembles de données et fournir des résultats en un temps record, à moins que les tâches de calcul ne soient exécutées sur plusieurs machines réparties sur le cloud.

Nous avons maintenant de nouveaux frameworks qui nous permettent de décomposer une tâche de calcul en plusieurs segments et d'exécuter chaque segment sur une machine différente.

Du côté du traitement des données, Mahout est un tel cadre d'exploration de données qui fonctionne normalement couplé à l'infrastructure Hadoop en arrière-plan pour gérer d'énormes volumes de données.

De l'autre côté du hadoop multitâche, Apache Oozie est un système de planification pour exécuter et gérer les travaux Hadoop dans un environnement distribué. Il permet de combiner plusieurs tâches complexes à exécuter dans un ordre séquentiel pour réaliser une tâche plus importante. Dans une séquence de tâches, deux ou plusieurs tâches peuvent également être programmées pour s'exécuter en parallèle.

La combinaison de ces outils nous donne un puissant moteur de Big Data pour nettoyer et traiter les données rapidement.

# Introduction à Mahout :

## Définition :

Un Mahout est celui qui conduit un éléphant comme maître. Le nom vient de son association étroite avec Apache Hadoop qui utilise un éléphant comme logo.

Hadoop est un framework open-source d'Apache qui permet de stocker et de traiter des mégadonnées dans un environnement distribué sur des clusters d'ordinateurs à l'aide de modèles de programmation simples.

Apache Mahout est un projet open source qui est principalement utilisé pour créer des algorithmes d'apprentissage automatique évolutifs. Il met en œuvre des techniques d'apprentissage automatique populaires telles que:

* Recommandation
* Classification
* Regroupement

Apache Mahout a commencé comme sous-projet de Lucene d'Apache en 2008. En 2010, Mahout est devenu un projet de haut niveau d'Apache.

## Caractéristiques de Mahout

Les fonctionnalités primitives d'Apache Mahout sont répertoriées ci-dessous.

* Les algorithmes de Mahout sont écrits sur Hadoop, donc cela fonctionne bien dans un environnement distribué. Mahout utilise la bibliothèque Apache Hadoop pour évoluer efficacement dans le cloud.
* Mahout offre au codeur un cadre prêt à l'emploi pour effectuer des tâches d'exploration de données sur de gros volumes de données.
* Mahout permet aux applications d'analyser de grands ensembles de données de manière efficace et rapide.
* Comprend plusieurs implémentations de clustering activées par MapReduce, telles que k-means, k-means floue, Canopy, Dirichlet et Mean-Shift.
* Prend en charge les implémentations de classification Naive Bayes distribuées et Naive Bayes complémentaires.
* Livré avec des capacités distribuées de fonction de fitness pour une programmation évolutive.
* Comprend des bibliothèques matricielles et vectorielles.

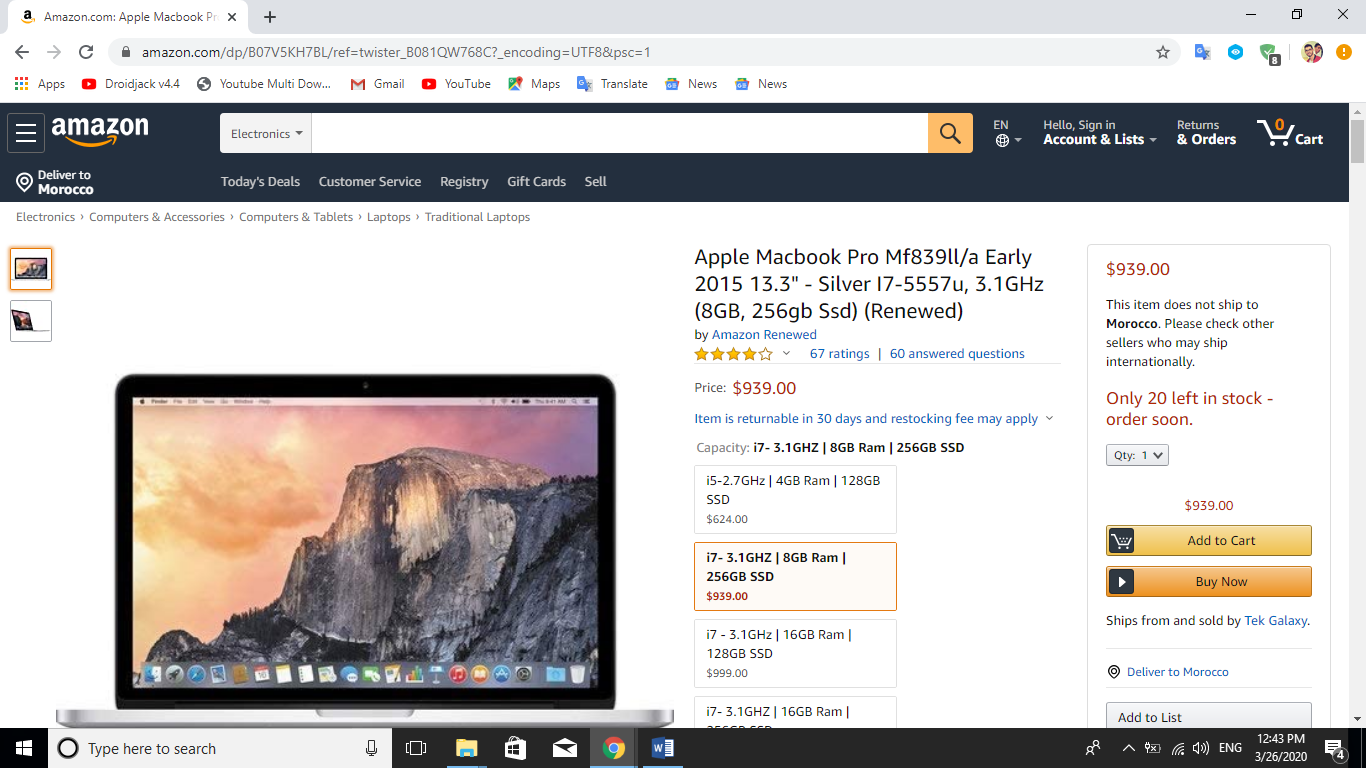
## Applications de Mahout

* Des entreprises telles qu'Adobe, Facebook, LinkedIn, Foursquare, Twitter et Yahoo utilisent Mahout en interne.
* Foursquare vous aide à trouver des endroits, de la nourriture et des divertissements disponibles dans une zone particulière. Il utilise le moteur de recommandation de Mahout.
* Twitter utilise Mahout pour la modélisation des intérêts des utilisateurs.
* Yahoo! utilise Mahout pour l'extraction de motifs

## Mahout – Recommendation

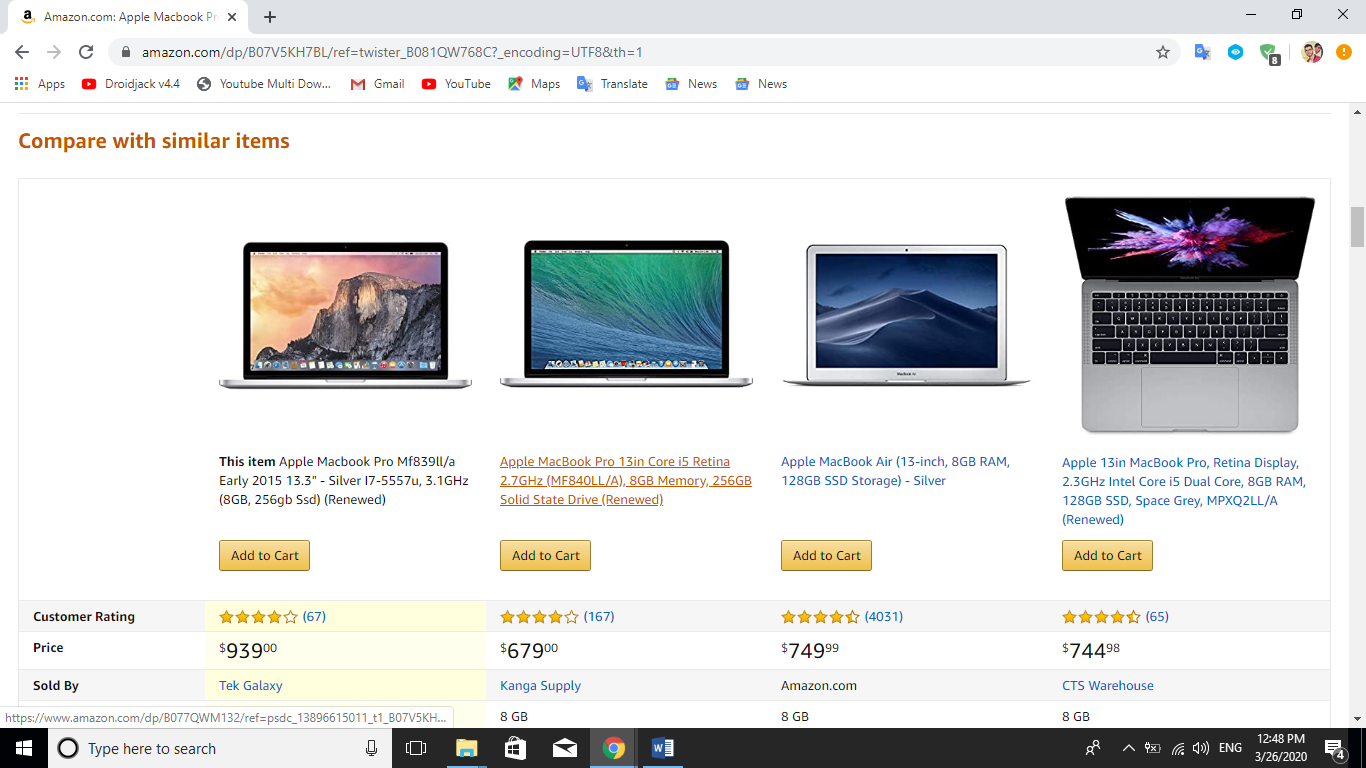
### Définition

Mahout Recommandation implémente un cadre de filtrage collaboratif - popularisé par Amazon et autres - utilise des données hystoriques (évaluations, clics et achats) pour fournir des recommandations basé sur l'utilisateur: recommander des articles en trouvant des utilisateurs similaires.

 supposons que nous voulions acheter un Macbook pro sur Amazon, nous avons fait quelques recherches puis nous avons cliqué sur celui-ci:

faisons défiler la page, tout à coup, nous verrons qu'Amazon propose une liste d'articles recommandés pour attirer notre attention sur des produits particuliers qui pourraient nous intéresser!

comme indiqué ci-dessous, Amazon nous recommande 3 autres articles en plus de celui que nous choisissons afin de les comparer

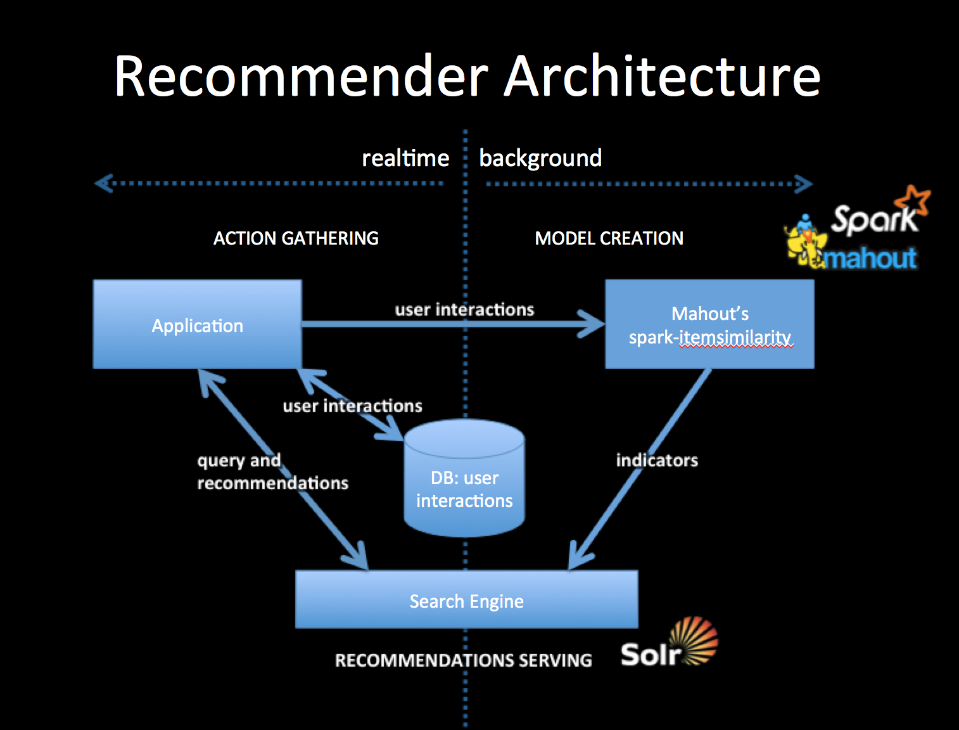


Cet liste de recommandations est produite à l'aide de moteurs de recommandation. Mahout fournit des moteurs de recommandation de plusieurs types tels que:

* des recommandations basées sur les utilisateurs,
* des recommandateurs basés sur des articles, et
* plusieurs autres algorithmes.

### Moteur de recommandation Mahout

Mahout a un moteur de recommandation non distribué et non basé sur Hadoop. Vous devez transmettre un document texte ayant des préférences utilisateur pour les éléments. Et la sortie de ce moteur serait les préférences estimées d'un utilisateur particulier pour d'autres éléments.



## Mahout-Clustering

Clustering est la procédure pour organiser les éléments ou les éléments d'une collection donnée en groupes en fonction de la similitude entre les éléments. Par exemple, les applications liées à l'édition de nouvelles en ligne regroupent leurs articles de presse en utilisant le clustering

### Applications du clustering

* Le clustering est largement utilisé dans de nombreuses applications telles que les études de marché, la reconnaissance de formes, l'analyse de données et le traitement d'images.
* Le clustering peut aider les spécialistes du marketing à découvrir des groupes distincts dans leur clientèle. Et ils peuvent caractériser leurs groupes de clients en fonction de leurs habitudes d'achat.
* Dans le domaine de la biologie, il peut être utilisé pour dériver des taxonomies végétales et animales, classer les gènes avec des fonctionnalités similaires et obtenir un aperçu des structures inhérentes aux populations.
* Le regroupement aide à identifier les zones d'utilisation similaire des terres dans une base de données d'observation de la Terre.
* Le clustering permet également de classer les documents sur le Web pour la découverte d'informations.
* Le clustering est utilisé dans les applications de détection de valeurs aberrantes telles que la détection de fraude par carte de crédit.
* En tant que fonction d'exploration de données, l'analyse de cluster sert d'outil pour obtenir un aperçu de la distribution des données afin d'observer les caractéristiques de chaque cluster.

En utilisant Mahout, nous pouvons regrouper un ensemble de données donné. Les étapes requises sont les suivantes:

* **Algorithme** Vous devez sélectionner un algorithme de clustering approprié pour regrouper les éléments d'un cluster.
* **Similarité et dissimilarité** Vous devez avoir une règle en place pour vérifier la similitude entre les éléments récemment rencontrés et les éléments des groupes.
* **Condition d'arrêt** Une condition d'arrêt est requise pour définir le point où aucun regroupement n'est requis.

### Procédure de clustering

Pour regrouper les données fournies, vous devez -

* Démarrez le serveur Hadoop. Créez les répertoires requis pour le stockage des fichiers dans le système de fichiers Hadoop. (Créez des répertoires pour le fichier d'entrée, le fichier de séquence et la sortie en cluster en cas de canopée).
* Copiez le fichier d'entrée dans le système de fichiers Hadoop à partir du système de fichiers Unix.
* Préparez le fichier de séquence à partir des données d'entrée.
* Exécutez l'un des algorithmes de clustering disponibles.
* Obtenez les données en cluster.

## Mahout – Classification

### Définition

La classification est une technique d'apprentissage automatique qui utilise des données connues pour déterminer comment les nouvelles données doivent être classées dans un ensemble de catégories existantes. Par exemple :

* L'application iTunes utilise la classification pour préparer des listes de lecture.
* Les fournisseurs de services de messagerie tels que Yahoo! et Gmail utilise cette technique pour décider si un nouveau courrier doit être classé comme spam. L'algorithme de catégorisation se forme en analysant les habitudes des utilisateurs de marquer certains courriers comme des spams. Sur cette base, le classificateur décide si un futur courrier doit être déposé dans votre boîte de réception ou dans le dossier spam.

### Applications de la classification

* **Détection de fraude par carte de crédit -** Le mécanisme de classification est utilisé pour prédire les fraudes par carte de crédit. En utilisant les informations historiques des fraudes précédentes, le classificateur peut prédire quelles transactions futures pourraient se transformer en fraudes.
* **Courriers indésirables -** En fonction des caractéristiques des courriers indésirables précédents, le classificateur détermine si un courrier électronique récemment rencontré doit être envoyé dans le dossier de courrier indésirable.

### Naive Bayes Classifier

Mahout utilise l'algorithme de classification Naive Bayes. Il utilise deux implémentations:

* Classification des baies naïves distribuées
* Classification complémentaire Naive Bayes

Naive Bayes est une technique simple pour construire des classificateurs. Ce n'est pas un algorithme unique pour former de tels classificateurs, mais une famille d'algorithmes. Un classificateur Bayes construit des modèles pour classer les instances de problème. Ces classifications sont faites en utilisant les données disponibles.

Un avantage de Bayes naïf est qu'il ne nécessite qu'une petite quantité de données d'entraînement pour estimer les paramètres nécessaires à la classification.

Pour certains types de modèles de probabilité, les classificateurs bayésiens naïfs peuvent être formés très efficacement dans un cadre d'apprentissage supervisé.

Malgré ses hypothèses trop simplifiées, les classificateurs naïfs de Bayes ont très bien fonctionné dans de nombreuses situations complexes du monde réel.

### Procédure de classement

Les étapes suivantes doivent être suivies pour implémenter la classification :

* Générer des exemples de données
* Créer des fichiers de séquence à partir des données
* Convertir des fichiers de séquence en vecteurs
* Former les vecteurs
* Testez les vecteurs

# Introduction à Oozie:

## Définition

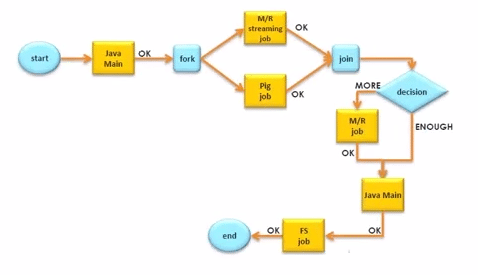
Apache Oozie est un système de planification pour exécuter et gérer les travaux Hadoop dans un environnement distribué. Il permet de combiner plusieurs tâches complexes à exécuter dans un ordre séquentiel pour réaliser une tâche plus importante. Dans une séquence de tâches, deux ou plusieurs tâches peuvent également être programmées pour s'exécuter en parallèle.

L'un des principaux avantages d'Oozie est qu'il est étroitement intégré à la pile Hadoop prenant en charge divers travaux Hadoop comme Hive, Pig, Sqoop ainsi que des travaux spécifiques au système comme Java et Shell.

Oozie est une application Web Java Open Source disponible sous licence Apache 2.0. Il est responsable du déclenchement des actions de workflow, qui à son tour utilise le moteur d'exécution Hadoop pour réellement exécuter la tâche. Par conséquent, Oozie est en mesure de tirer parti des machines Hadoop existantes pour l'équilibrage de charge, le basculement, etc.

Oozie détecte l'achèvement des tâches par le biais de rappels et d'interrogations. Lorsque Oozie démarre une tâche, il fournit une URL HTTP de rappel unique à la tâche et avertit cette URL lorsqu'elle est terminée. Si la tâche ne parvient pas à appeler l'URL de rappel, Oozie peut interroger la tâche pour l'achèvement.

Les trois types d'emplois suivants sont courants à Oozie -

* **Oozie Workflow jobs:** Ils sont représentés sous forme de graphiques acycliques dirigés (DAG) pour spécifier une séquence d'actions à exécuter.
* **Oozie Coordinator Jobs**: Il s'agit de travaux de workflow déclenchés par le temps et la disponibilité des données.
* **Oozie Bundle :** Ceux-ci peuvent être appelés un ensemble de plusieurs tâches de coordinateur et de workflow.

la figure dans le côté droit est un exemple simple de flux de travail oozie

## Cas d'utilisation d'Apache Oozie

Apache Oozie est utilisé par les administrateurs système Hadoop pour exécuter une analyse de journal complexe sur HDFS. Les développeurs Hadoop utilisent Oozie pour effectuer des opérations ETL sur des données dans un ordre séquentiel et enregistrer la sortie dans un format spécifié (Avro, ORC, etc.) dans HDFS.

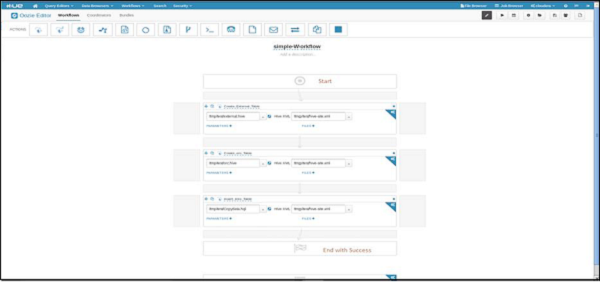
Dans une entreprise, les jobs Oozie sont planifiés en tant que coordinateurs ou bundles.

## Editeurs d'Oozie

### Éditeur de teintes pour Oozie



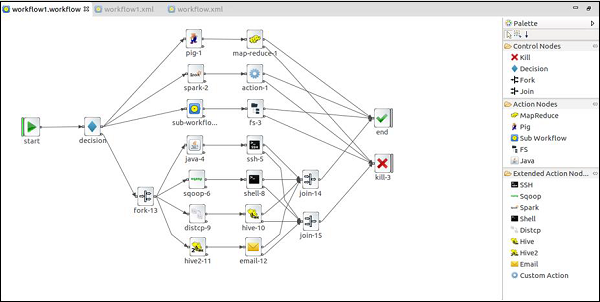
Cet éditeur est très pratique à utiliser et est disponible avec presque toutes les solutions des fournisseurs Hadoop.



### Plugin Oozie Eclipse (OEP)



Le plugin Oozie Eclipse (OEP) est un plugin Eclipse permettant de modifier graphiquement les workflows Apache Oozie. Il s'agit d'un éditeur graphique permettant de modifier les workflows Apache Oozie dans Eclipse.



## Oozie workflow

Le workflow dans Oozie est une séquence d'actions organisées dans un DAG de dépendance de contrôle (Direct Acyclic Graph). Les actions sont en dépendance contrôlée car l'action suivante ne peut s'exécuter que selon la sortie de l'action en cours. Les actions suivantes dépendent de son action précédente. Une action de flux de travail peut être une action Hive, une action Pig, une action Java, une action Shell, etc. Il peut y avoir des arbres de décision pour décider comment et à quelle condition un travail doit être exécuté.

Une fourchette est utilisée pour exécuter plusieurs travaux en parallèle. Les workflows Oozie peuvent être paramétrés (des variables comme $ {nameNode} peuvent être passées dans la définition du workflow). Ces paramètres proviennent d'un fichier de configuration appelé fichier de propriétés. (Plus d'informations à ce sujet dans les chapitres suivants)

## Cordinators

Les applications de coordinateur permettent aux utilisateurs de planifier des flux de travail complexes, y compris des flux de travail régulièrement planifiés. Oozie Coordinator modélise les déclencheurs d'exécution du flux de travail sous forme de prédicats de temps, de données ou d'événements. Le travail de workflow mentionné à l'intérieur du coordinateur n'est démarré qu'après que les conditions données sont remplies.

## Bundle

Le système Oozie Bundle permet à l'utilisateur de définir et d'exécuter un ensemble d'applications de coordination souvent appelé pipeline de données. Il n'y a pas de dépendance explicite entre les applications de coordination dans un bundle. Cependant, un utilisateur peut utiliser la dépendance aux données des applications de coordination pour créer un pipeline d'application de données implicite.

L'utilisateur pourra démarrer / arrêter / suspendre / reprendre / réexécuter au niveau du bundle, ce qui se traduira par un contrôle opérationnel meilleur et facile.

## Outils de ligne de commande

Oozie fournit un utilitaire de ligne de commande, Oozie, pour effectuer des tâches de travail et d'administration.

oozie version : show client version

Voici quelques-unes des autres opérations de travail

oozie job <OPTIONS> :

-action <arg> coordinator rerun on action ids (requires -rerun); coordinator log

   retrieval on action ids (requires -log)

-auth <arg> select authentication type [SIMPLE|KERBEROS]

-change <arg> change a coordinator/bundle job

-config <arg> job configuration file '.xml' or '.properties'

-D <property = value> set/override value for given property

-date <arg> coordinator/bundle rerun on action dates (requires -rerun)

-definition <arg> job definition

-doas <arg> doAs user, impersonates as the specified user

-dryrun Supported in Oozie-2.0 or later versions ONLY - dryrun or test run a

   coordinator job, job is not queued

-info <arg> info of a job

-kill <arg> kill a job

-len <arg> number of actions (default TOTAL ACTIONS, requires -info)

-localtime use local time (default GMT)

-log <arg> job log

-nocleanup do not clean up output-events of the coordinator rerun actions (requires

   -rerun)

-offset <arg> job info offset of actions (default '1', requires -info)

-oozie <arg> Oozie URL

-refresh re-materialize the coordinator rerun actions (requires -rerun)

-rerun <arg> rerun a job (coordinator requires -action or -date; bundle requires

   -coordinator or -date)

-resume <arg> resume a job

-run run a job

-start <arg> start a job

-submit submit a job

-suspend <arg> suspend a job

-value <arg> new endtime/concurrency/pausetime value for changing a coordinator

    job;new pausetime value for changing a bundle job

-verbose verbose mode

Pour vérifier l'état du travail, les commandes suivantes peuvent être utilisées.

-auth <arg> select authentication type [SIMPLE|KERBEROS]

-doas <arg> doAs user, impersonates as the specified user.

-filter <arg> user = <U>; name = <N>; group = <G>; status = <S>; ...

-jobtype <arg> job type ('Supported in Oozie-2.0 or later versions ONLY -

   coordinator' or 'wf' (default))

-len <arg> number of jobs (default '100')

-localtime use local time (default GMT)

-offset <arg> jobs offset (default '1')

-oozie <arg> Oozie URL

-verbose verbose mode

Par exemple - Pour vérifier l'état du système Oozie, vous pouvez exécuter la commande suivante -

oozie admin -oozie http://localhost:8080/oozie -status

Validation d'un XML de workflow –

oozie validate myApp/workflow.xml

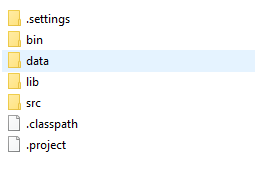
# Projet en Maout-oozie :

## Introduction

Mahout Joue un grand rôle dans de nombreux sites Web en tant que cadre d'apprentissage automatique pour fournir une expérience utilisateur fluide et parfaite, l'une des puissantes opérations que Mahout peut faire est la recommandation comme mentionné dans les chapitres précédents, dans ce projet, nous avons fait une recommandation que nous sera développé en détail dans les lignes suivantes.

## Structure du projet

**bin** : contient le bytecode des fichiers de nos classes de projet



**data** : contient les fichiers de données pour créer des modèles de données et alimenter les algorithmes Mahout

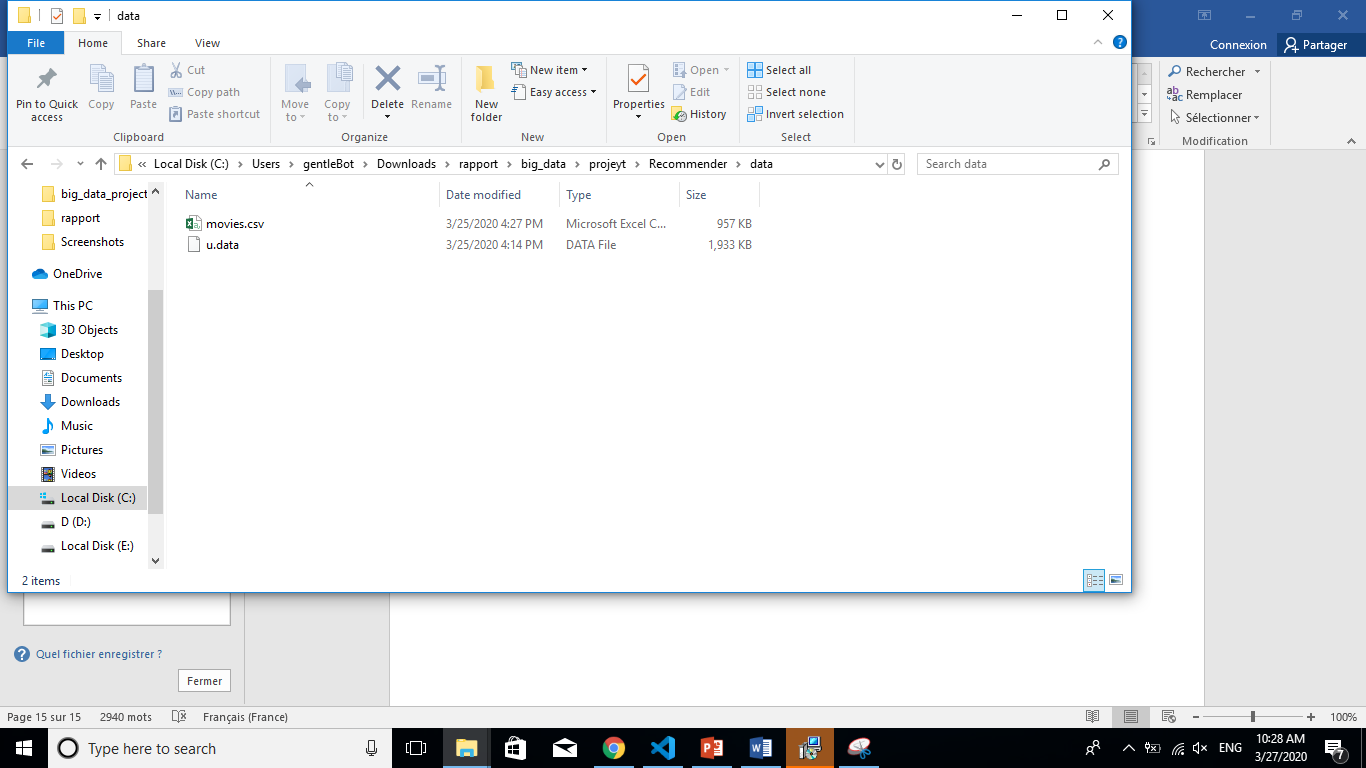
**lib** : contient des bibliothèques Mahout

**src :** ce dossier est le conteneur de nos fichiers Java

## Recommander étape par étape

### Créer un objet de modèle de données

Le constructeur de la classe **LogLikelihoodSimilarity** nécessite un objet de modèle de données, qui contient un fichier qui contient les détails des éléments et des préférences d'un produit.

dans les données du dossier, nous mettons un fichier csv appelé movies.csv pour aider à créer des modèles de données

L'objet Data Model nécessite l'objet fichier, qui contient le chemin du fichier d'entrée. Créez l'objet Data Model comme indiqué ci-dessous.

*DataModel* dm = new FileDataModel(new File("data/movies.csv"));

### Créer un objet UserSimilarity

Créez un objet ItemrSimilarity à l'aide de la classe **LogLikelihoodSimilarity**, comme indiqué ci-dessous:

*ItemSimilarity* sim = new LogLikelihoodSimilarity(dm);

### Créer un objet de recommandation

Créez un objet. **itemBasedRcommender** Passez tous les objets créés ci-dessus à son constructeur comme indiqué ci-dessous.

*GenericItemBasedRecommender* recommender = new GenericItemBasedRecommender(dm, sim);

### Recommander des articles à un utilisateur

maintenant la méthode **mostSimilarItems** () obtient deux paramètres le premier est l'ID de l'article, le second est le nombre de recommandations à louer

puis récupérez la liste des recommandations, montrez-la à l'utilisateur comme indiqué ci-dessous :

*int* x=1;

            for(*LongPrimitiveIterator* items = dm.getItemIDs(); items.hasNext();) {

*long* itemId = items.nextLong();

*List*<*RecommendedItem*> recommendetions = recommender.mostSimilarItems(itemId, 5);

                for(*RecommendedItem* recommendation:recommendetions)

                {

                    System.out.println(itemId+", "+recommendation.getItemID()+", "+recommendation.getValue());

                }

                x++;

                if(x>10) System.exit(1);

            }

maintenant nous avons terminé notre simple projet Mahout. le prochain un planificateur oozie.

# Projet en Oozie

## Introduction