N°d'ordre :
N° de série :

RÉPUBLIQUE ALGÉRIENNE DÉMOCRATIQUE ET POPULAIRE
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

---

UNIVERSITE ECHAHID HAMMA LAKHDAR - EL OUED
FACULTÉ DES SCIENCES EXACTES
Département D'Informatique

Mémoire de Fin D'étude
Présenté pour l'obtention du Diplôme de

MASTER ACADEMIQUE

Domaine:**Mathématique et Informatique**
Filière : **Informatique**
Spécialité : **Systèmes Distribués et Intelligence Artificielle**

Présenté par :

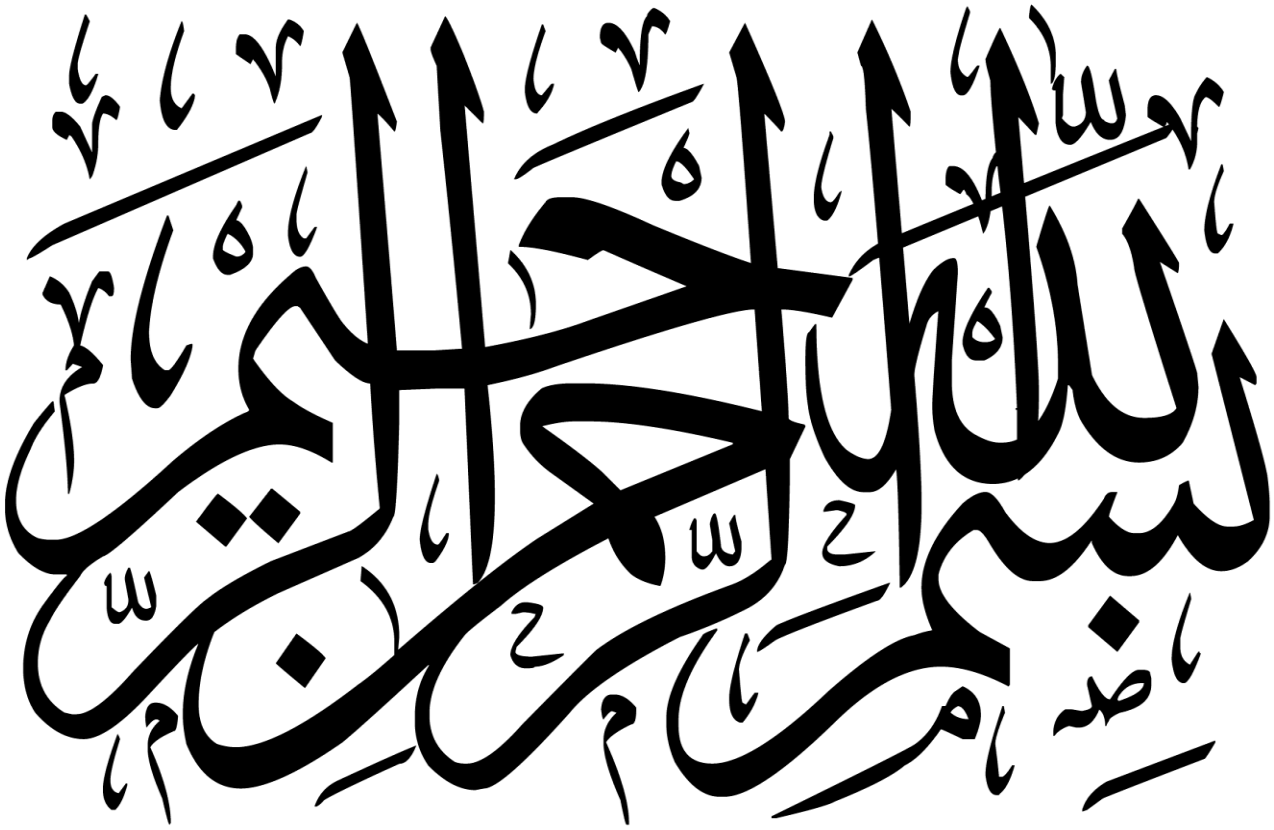
- BARAIKA Imane
- OMRANI Asma

Thème

Modélisation et implémentation d'un système de régression pour une architecture Big Data

M.	……	MCA	Président
M.	……	MAA	Rapporteur
M.	NAOUI Med Anouar	MAA	Encadreur

Année Universitaire:2016-2017


بسم الله الرحمن الرحيم

Remerciements

Nous remercions tout d'abord notre Dieu qui nous a donné la force et la volonté pour élaborer ce travail.

Nous adressons nos vifs remerciements à notre encadreur "Mr naoui med anouar ", qui n'a pas cessé de nous donner les conseils et les bonnes orientations et nous prive pas de son temps et Sans son aide, notre travail n'aurait pas vu la lumière .

Notre reconnaissance va aussi à tous ceux qui ont collaboré à notre formation en particulier les enseignants du département d'Informatique ,

Universitaire Hama Lakhdar d'El-Oued

Aussi à nos collègues de la promotion 2016-2017 On remercie également tous ceux qui ont participé de près ou de loin à élaborer ce travail

DEDICACE

Avant tout, je remercie ALLAH le tout puissant de m'avoir donné le
courage et la patience pour réaliser ce travail malgré toutes les difficultés rencontrées.
Tous d'abord Je dédie ce modeste travail
À mon très cher père MOSTAPHA qui m'a
encouragé par tous les moyens à avoir la confiance et l'espoir au cours
de la rédaction de ce travail.
À ma très cher mère adorée qui m'a aidé, grâce à leur prière et à leur bénédiction.
A tous mes frères
A ma binôme IMANE BARAIKA
À tous mes amies
À tous les étudiants de la faculté Informatique surtout les étudiants de la 2éme année master promotion 2017
À tous les habitants de SOUIHLA.

Asma

DEDICACE

Avant tout, je remercie ALLAH le tout puissant de m'avoir donné le
courage et la patience pour réaliser ce travail malgré toutes les difficultés rencontrées.
Tous d'abord Je dédie ce modeste travail
À mon très cher père
À ma très cher mère adorée qui m'a aidé, grâce à leur prière et à leur bénédiction.
A tous mes frères
À tous mes amies
À ma très ses enfant "Mabrouk" et " Aya" et "Nehal"
Et à toute la famille.
A ma binôme Asma Omrani
À tous les étudiants de la faculté Informatique surtout les étudiants de la 2éme année master promotion 2017
À tous les habitants de Djedaida.

Imane

Résume

Big Data est une discipline de recherche importante, la régression linaire des données dans cet architecture est basé sur l'algorithme MapReduce . Notre objectif est la proposition d'une MapReduce algorithme pour la régression de données.

Mots clés : Analyse de données ; MapReduce algorithme; Big data architecture .

Abstract

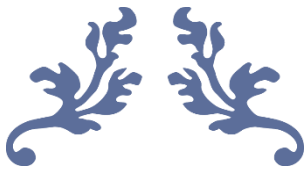
Big Data is an important research discipline; data regression in this architecture is based on Map Reduce algorithm . Our goal is to propose a MapReduce algorithm for data regression.

Key words: Data analysis; MapReduce algorithm; Big data architecture.

ملخص

البيانات الكبيرة هو عمل بحثي كبير، يستند الانحدار الخطي في هذه الهيكلة على خوارزمية مابريديوس.هدفنا هو اقتراح خوارزمية مابريديوس لأجل الانحدار الخطي.

كلمات البحث: تحليل البيانات, خوارزمية مابريديوس. هيكلة البيانات الكبيرة.



SOMMAIRE

Sommaire

liste de figure

Chapitre I

Chapitre II

Chapitre III

Chapitre IV

Introduction Général

Introduction général

De nos jours, les données augmentent très rapidement au jour dans le texte, les journaux, la musique, les vidéos, etc., et que les données doivent être stockées pour un accès ultérieur et Big Data stocke ces données de manière bien adaptée, à laquelle il faut accéder et affiché aux utilisateurs. Les utilisateurs normaux ne peuvent pas analyser les données directement et il est très difficile d'analyser les données normales, de sorte que les données doivent être affichées de manière graphique ou simple pour analyser facilement les données. Pour cela, certains outils et techniques sont utilisés comme le langage R qui utilise les statistiques et Hadoop pour le traitement parallèle des données afin que les données puissent être facilement accessibles pour l'analyse. Dans ce projet, différentes façons sont là pour décrire ou montrer les données collectées,

Problématique : Comment appliquer une régression linéaire sur une architecture Big Data .
L'objectif : Le but de notre étude est de trouver une solution de appliquer une système de régression linaire pour une architecture Big Data .En second lieu, comme notre sujet est d'essayer de développer un système de régression linéaire pour les Big Data en utilisant l'algorithme parallèle mapreduce.

Ce mémoire s'articule autour de quatre chapitres :

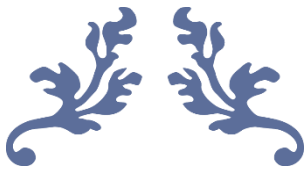
- Le premier chapitre donne un état de l'art qui contient des concepts de base de Fouille de types et Donnée techniques qui y sont utilisés, et la régression linaire .

- Le deuxième chapitre montre et explique big data de donnée set de l'infrastructure, et ses propriétés et de la structure, ainsi que de toucher et positifs à l'algorithme et son principe mapreduce.

- Le troisième chapitre Le chapitre III, nous examinerons à l'algorithme mapreduce et les grandes données et Des études antérieures sur la régression linéaire et grandes quantités de données Nous offrirons des algorithmes utilisés et les grandes lignes du système.

- Le dernier chapitre présente l'expérimentation et la validation de système réalisé.

Enfin, nous conclurons ce projet par une conclusion général.



CHAPITRE I

Analyse de Donnée

1- Introduction

Avec le développement dans tous les domaines des médias à devenir difficile de traiter avec les bases de données ont montré la fouille des données

Pour faciliter le traitement de grandes bases de données pour analyser et en extraire des informations

Dans ce chapitre, nous allons examiner le concept de classification et traitement de la fouille donnée, étapes et les champs utilisés et les moyens pour y faire face en termes de classification et de régression et les types de régression linéaire.

2- histoire de Fouille de Donnée

Le “data mining” que l’on peut traduire par “fouille de données” apparait au milieu des années**1990** aux États-Unis comme une nouvelle discipline a l’interface de la statistique et des technologies de l’information : bases de données, intelligence artificielle, apprentissage.

3- Définition de Fouille de Donnée

Fouille de données signifie l’extraction de connaissance a travers l’analyse d’une grande quantité de données pour utiliser ces connaissances dans le processus de décision. On peut trouver les données stocker et organisé dans les data-martes et les entrepôts de données, ou dans autre sources non structurées.

Le processus de fouille de données implique plusieurs étapes avant de trouver un modèle de décision qui peut être un ensemble de règles, de l’équation ou des fonctions de transfert complexes. Selon leur objectif le fouille de données se compose en deux catégories supervisé et non supervisé, qui nous en parler dans les technique de fouille de données.[1]

4- Processus de Fouille de Donnée

Il est très important de comprendre que le fouille de données n’est pas seulement le problème de découverte de modèles dans un ensemble de donnée. Ce n’est qu’une seule étape dans tout un processus suivi par les scientifiques, les ingénieurs ou toute autre personne qui cherche à extraire les connaissances à partir des données. En **1996** un groupe d’analystes définit le fouille de données comme étant un processus composé de cinq étapes sous le standard CRISP-DM (Cross-Industry Standard Process for Data Mining) comme schématisé ci-dessous(figure I-1) : [2]

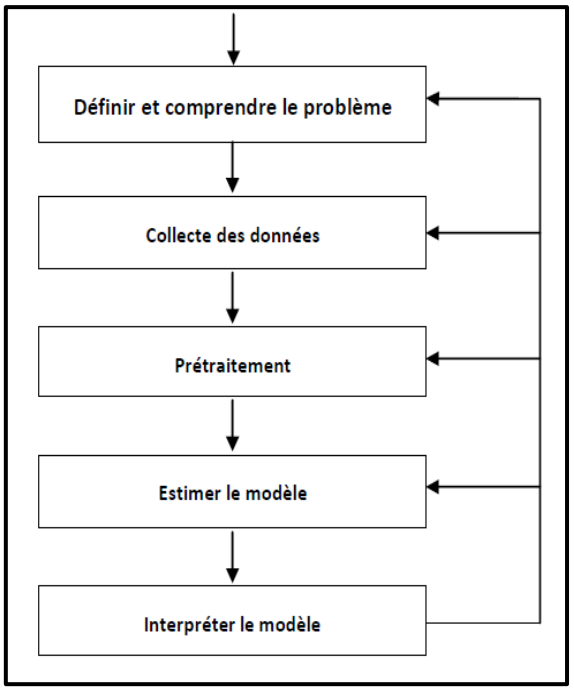


Figure I - 1: Processus de fouille de données .

4.1. Définition et compréhension du problème :

Dans la plus part des cas, il est indispensable de comprendre la signification des données et le domaine à explorer. Sans cette compréhension, aucun algorithme ne va donner un résultat fiable. En effet, Avec la compréhension du problème, on peut préparer les données nécessaires à l'exploration et interpréter correctement les résultats obtenus.

4.2. Collecte des données:

D'après la définition du problème et des objectifs du fouille de données, on peut avoir une idée sur les données qui doivent être utilisées. Ces données n'ont pas toujours le même format et la même structure. On peut avoir des textes, des bases de données, des pages web, ...etc. Parfois, on est amené à prendre une copie d'un système d'information en cours d'exécution, puis ramasser les données de sources éventuellement hétérogènes (fichiers, bases de données relationnelles, temporelles, ...).

4.3. Prétraitement :

Les données peuvent contenir plusieurs types d'anomalies : des données peuvent être omises à cause des erreurs de frappe ou à causes des erreurs dues au système lui-même, dans ce cas il faut remplacer ces données ou éliminer complètement leurs enregistrements.

Des données peuvent être incohérentes c.-à-d. qui sortent des intervalles permis, on doit les écarter où les normaliser. Parfois on est obligé à faire des transformations sur les données

pour unifier leur poids. Le prétraitement comporte aussi la réduction des données qui permet de réduire le nombre d'attributs pour accélérer les calculs et représenter les données sous un format optimal pour l'exploration. Dans la majorité des cas, le prétraitement doit préparer des informations globales sur les données pour les étapes qui suivent tel que la tendance centrale des données (moyenne, médiane, mode), le maximum et le minimum, le rang, les quartiles, la variance, ... etc.

Plusieurs techniques de visualisation des données telles que les courbes, les diagrammes, les graphes,... etc., peuvent aider à la sélection et le nettoyage des données. Une fois les données collectées, nettoyées et prétraitées on les appelle entrepôt de données (data warehouse).

4.4. Estimation du modèle :

Dans cette étape, on doit choisir la bonne technique pour extraire les connaissances (exploration) des données. Des techniques telles que les réseaux de neurones, les arbres de décision, les réseaux bayésiens, le clustering, ... sont utilisées. Généralement, l'implémentation se base sur plusieurs de ces techniques, puis on choisit le bon résultat. Dans le titre suivant on va détailler les différentes techniques utilisées dans l'exploration des données et l'estimation du modèle.

4.5. Interprétation du modèle et établissement des conclusions :

Généralement, l'objectif de la fouille de données est d'aider à la prise de décision en fournissant des modèles compréhensibles aux utilisateurs. En effet, les utilisateurs ne demandent pas des pages et des pages de chiffres, mais des interprétations des modèles obtenus. Les expériences montrent que les modèles simples sont plus compréhensibles mais moins précis, alors que ceux complexes sont plus précis mais difficiles à interpréter.

5. Techniques fouille de données

5.1. Techniques non supervisées

Dans les modèles non supervisés ou non orientés, il n'y a pas de champ de sortie, il n'y a que des entrées. La reconnaissance de formes est non orienté; elle n'est pas guidée par un attribut cible spécifique. Le but de ces modèles est de découvrir des motifs de données dans l'ensemble des champs d'entrée.

Les modèles non supervisés comprennent :

• Les modèles de dispersion : Dans ces modèles les groupes ne sont pas connus à l'avance. Au contraire, nous voulons que les algorithmes pour analyser les schémas de données d'entrée et

d'identifier les regroupements naturels de données ou de cas. Lorsque de nouveaux cas sont marqués par le modèle de cluster généré ils sont affectés à l'un des groupes révélés.

• Les modèles d'association de séquences : Ces modèles font également partie de la classe de la modélisation non supervisé. Ils ne comportent pas de prédiction directe d'un seul champ. En fait, tous les champs concernés ont un double rôle, car ils agissent comme des entrées et des sorties en même temps. Des modèles d'association de détecter des associations entre des événements discrets, des produits ou des attributs. Les modèles de séquence détectent des associations au fil du temps.

5.1.1. Clustring hiérarchique

Il considère comme la "mère" de tous les modèles de clustering. Il est appelé hiérarchique ou d'agglomération, car il commence avec une solution où chaque enregistrement comprend un groupe et peu à peu les groupes se former jusqu'au point où tous tomber dans un super-cluster (figure I - 2). À chaque étape, il calcule les distances entre toutes les paires d'enregistrements et les groupes les plus similaires. Une table (horaire d'agglomération) ou un graphique (dendrogramme) résume les étapes de regroupement et les distances respectives.

L'analyste doit consulter ces informations, identifier le point où l'algorithme commence à cas disjoints de groupe, et de décider ensuite sur le nombre de grappes à conserver. Cet algorithme ne peut pas traiter efficacement plus de quelques milliers de cas. Ainsi, il ne peut pas être directement appliqué dans la plupart des tâches de regroupement d'entreprise. Une solution habituelle consiste à une utilisation sur un échantillon de la population de clustering. Cependant, de nombreux autres algorithmes efficaces qui peuvent facilement gérer des millions d'enregistrements, le regroupement par échantillonnage n'est pas considéré comme une approche idéale.

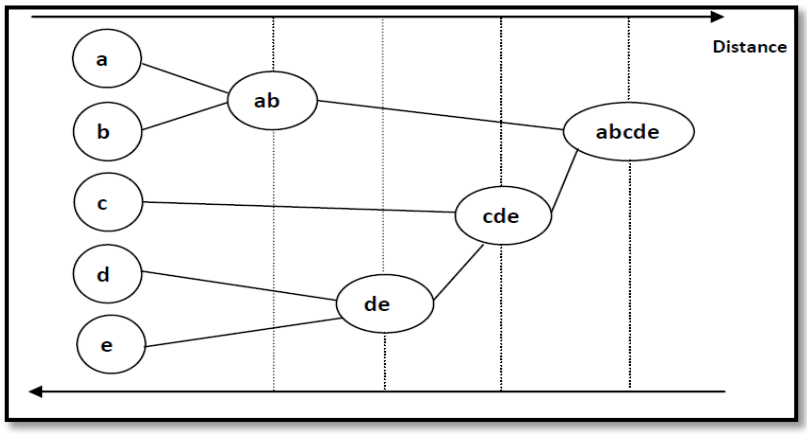


Figure I - 2 : Clustring hiérarchique [3].

5.1.2. K-means

C'est un moyen efficace et peut-être l'algorithme de segmentation le plus rapide qui peut gérer deux longues (plusieurs enregistrements) et des ensembles de données larges (de nombreuses dimensions de données et des champs d'entrée). Il s'agit d'une technique de segmentation basé sur la distance et, à la différence de l'algorithme hiérarchique, il n'a pas besoin de calculer les distances entre toutes les paires d'enregistrements. Le nombre de grappes d'être formés et est prédéterminée spécifiée par l'utilisateur à l'avance. Habituellement, un certain nombre de solutions différentes doit être jugé et évalué avant d'approuver le plus approprié.

5.1.3. Carte auto-organisatrice de Kohonen

Réseaux de Kohonen sont basés sur des réseaux de neuronaux et produisent typiquement une grille à deux dimensions ou une carte des grappes, où les cartes d'auto-organisation. Réseaux de Kohonen prennent généralement plus de temps à former que les K-means, mais ils fournissent un point de vue différent sur le regroupement qui est la peine d'essayer.

5.2. Techniques supervisées [4]

Dans la modélisation supervisée, ou prédictive, l'objectif est de prédire un événement ou d'estimer les valeurs d'un attribut numérique continue. Dans ces modèles, il existe des champs où les attributs d'entrée et une zone de sortie ou de la cible. Les champs d'entrée sont également appelés prédicteurs, car ils sont utilisés par le modèle pour identifier une fonction de prédiction de champ de sortie. Nous pouvons penser à des prédicteurs que la partie X de la fonction et le domaine cible que la partie Y, le résultat.

Le modèle utilise les champs de saisie qui sont analysées en ce qui concerne leur effet sur le champ cible. La reconnaissance de formes est "surveillé" par le domaine cible. Des relations sont établies entre les champs d'entrée et de sortie. Une cartographie " fonction d'entrée-sortie " est généré par le modèle, qui associe des prédicateurs et à la sortie permet la prédiction des valeurs de sortie, étant donné les valeurs des champs d'entrée.

Les modèles prédictifs sont subdivisés en modèles de classification et d'estimation :

5.2.1. Les modèle classification

Dans ces modèles les groupes ou classes cibles sont connus dès le départ. Le but est de classer les cas dans ces groupes prédéfinis ; en d'autres termes, à prévoir un événement. Le modèle généré peut être utilisé comme un moteur de marquage pour l'affectation de nouveaux cas pour les classes prédéfinies. Il estime aussi un score de propension pour chaque cas. Le score de propension dénote la probabilité d'occurrence du groupe cible ou d'un événement.

• Les modèle d'estimation : Ces modèles sont similaires à des modèles de classification, mais avec une différence majeure. Ils sont utilisés pour prédire la valeur d'un champ continu en fonction des valeurs observées des attributs d'entrée.

a. Arbre de décision

Les arbres de décision fonctionnent en séparant de façon récursive la population initiale. Pour chaque groupe, ils sélectionnent automatiquement l'indicateur le plus significatif, le prédicteur qui donne la meilleure séparation par rapport au champ cible. À travers des cloisons successives, leur objectif est de produire sous-segments pures, avec un comportement homogène en termes de production. Ils sont peut-être la technique la plus populaire de classification. Une partie de leur popularité, c'est parce qu'ils produisent des résultats transparents qui sont facilement interprétables, offrant un aperçu de l'événement à l'étude. Les résultats obtenus peuvent avoir deux formats équivalents. Dans un format de règle, les résultats sont représentés dans un langage simple que les règles ordinaires :
SI (VALEURS PREDICTIVES) ALORS (RESULTAT CIBLE ET SCORE DE CONFIANCE).

Dans une forme d'arborescence, les règles sont représentés graphiquement sous forme d'arbre dans laquelle la population initiale (nœud racine) est successivement divisé en des nœuds terminaux ou feuilles de sous-segments ayant un comportement similaire en ce qui concerne le champ cible.

Les algorithmes d'arbres de décision constituent selon la vitesse et l'évolutivité. Algorithmes disponibles sont:

- C5.0
- CHAID
- Classification et arbres de régression
- QUEST.

b. Règles de décision

Ils sont assez semblables à des arbres de décision et de produire une liste de règles qui ont le format des états humains compréhensible:
SI (VALEURS PREDICTIVES) ALORS (RESULTAT CIBLE ET SCORE DE CONFIANCE).

Leur principale différence par arbres de décision, c'est qu'ils peuvent produire plusieurs règles pour chaque enregistrement. Les arbres de décision génèrent des règles exhaustives et mutuellement exclusifs qui couvrent tous les records. Pour chaque enregistrement une seule règle s'applique. Au contraire, les règles de décision peuvent générer un ensemble de règles de

chevauchement. Plus d'une règle, avec des prédictions différentes, peut être vrai pour chaque enregistrement. Dans ce cas, les règles sont évaluées, à travers une procédure intégrée, afin de déterminer l'une pour l'évaluation. Habituellement, une procédure de vote est appliquée, qui combine les règles et les moyennes de leurs confidences individuelles pour chaque catégorie de sortie. Enfin, la catégorie ayant la confiance la moyenne la plus élevée est sélectionnée comme la prédiction.

Les algorithmes de règles de décision comprennent :

- C5.0
- Liste de décision.

c. Réseaux de neurone

Les réseaux de neurones sont des puissants algorithmes d'apprentissage automatique qui utilisent des fonctions de cartographie complexe, non linéaire pour l'estimation et classification. Ils sont constitués de neurones organisés en couches. La couche d'entrée contient les prédicateurs ou neurones d'entrée. La couche de sortie comprend dans le champ cible. Ces modèles permettent d'estimer des poids qui relient les prédicteurs (couche d'entrée à la sortie). Modèles avec des topologies plus complexes peuvent également inclure, couches cachées intermédiaires, et les neurones. La procédure de formation est un processus itératif. Enregistrements en entrée, avec des résultats connus, sont présentés sur le réseau et la prédiction du modèle est évaluée par rapport aux résultats observés. Erreurs observées sont utilisés pour ajuster et d'optimiser les estimations du poids initial. Ils sont considérés comme des solutions opaques ou "boîte noir" car ils ne fournissent pas une explication de leurs prédictions. Ils fournissent seulement une analyse de sensibilité, qui résume l'importance prédictive des champs d'entrée. Ils nécessitent une connaissance statistique minimum mais, selon le problème, peut nécessiter un temps de traitement à long pour la formation

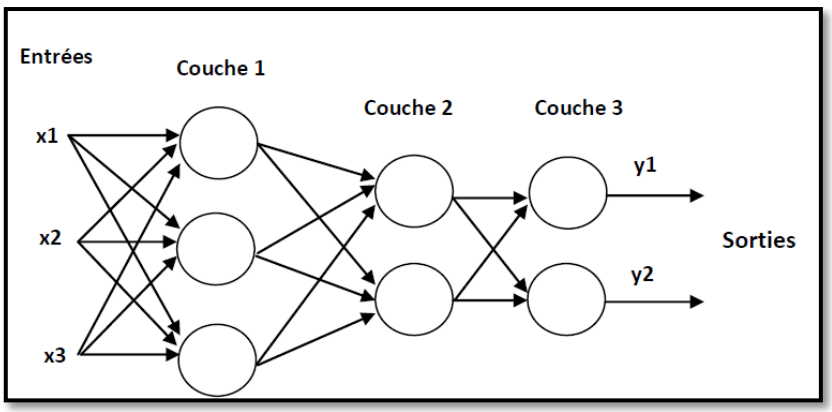


Figure I - 3: Réseau de neurones artificiel .

d. Machines à vecteurs supports (SVM)

SVM est un algorithme de classification qui peut modéliser les profils de données non linéaires hautement complexes, et d'éviter les sur-apprentissages, c'est-à-dire la situation dans laquelle un modèle mémorise les modèles ne concernent que des cas spécifiques analysés. SVM fonctionne en données cartographiques à un espace de grande dimension caractéristique dans lequel les enregistrements deviennent plus facilement séparables (ie, séparés par des fonctions linéaires) à l'égard des catégories de cibles.

Les données d'entraînement d'entrée sont transformés de manière appropriée par les fonctions du noyau non linéaires et cette transformation est suivie d'une recherche de fonctions plus simples, c'est-à des fonctions linéaires, qui enregistre de façon optimale distincts. Les analystes expérimentent généralement avec différentes fonctions de transformation et de comparer les résultats. Globalement SVM est un algorithme efficace et exigeant, en termes de ressources de mémoire et de temps de traitement. En outre, il manque de transparence puisque les prévisions ne sont pas expliquées et seulement l'importance des prédicteurs est résumée.

e. Réseaux bayésiens

Les modèles bayésiens sont des modèles de probabilité qui peuvent être utilisées dans des problèmes de classification pour estimer la probabilité d'occurrences. Ils sont des modèles graphiques qui fournissent une représentation visuelle des relations d'attributs, en assurant la transparence, et une explication de la justification du modèle.

-En plus des types dont nous avons parlé il y a un autre type d'une régression qui fait l'objet de notre étude, nous allons le regarder comme suit:

6- Régression

La régression est la méthode utilisée pour l'estimation des valeurs continues. Son objectif est de trouver le meilleur modèle qui décrit la relation entre une variable continue de sortie et une ou plusieurs variables d'entrée. Il s'agit de trouver une fonction f qui se rapproche le plus possible d'un scénario donné d'entrées et de sorties [2].

Il existe deux type de le régression :

6.1.régression linaire

La régression linéaire se classe parmi les méthodes d'analyses multi variées qui traitent des données quantitatives. C'est une méthode d'investigation sur données d'observations, ou d'expérimentations, où l'objectif principal est de rechercher une liaison linéaire entre une variable Y quantitative et une ou plusieurs variables X également quantitatives.[5]

il y a existé deux type de la régression linaire :

6.1.1. régression linaire simple

- Pour décrire une relation linéaire entre deux variables quantitatives ou encore pour pouvoir prédire Y pour une valeur donnée de X, nous utilisons une droite de régression:

$$Y = \beta 0 + \beta 1X + \varepsilon$$

- Puisque tout modèle statistique n'est qu'une approximation (nous espérons la meilleure possible!!), il y a toujours une erreur, notée ε dans le modèle, car le lien linéaire n'est jamais parfait.

•S 'il y avait une relation linéaire parfaite entre Y et X, le terme d 'erreur serait toujours égale à 0, et toute la variabilité de Y serait expliquée par la variable indépendante X. [6]

6.1.2. Régression linaire multiple

- Il est fort possible que la variabilité de la variable dépendante Y soit expliquée non pas par une seule variable indépendante X mais plutôt par une combinaison linéaire de plusieurs variables indépendantes $\mathbf{X_1, X_2, \ldots, X_p}$.

- Dans ce cas le modèle de régression multiple est donné par:

$$\mathbf{Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_p X_p + \varepsilon}$$

- Aussi, à l'aide des données de l'échantillon nous estimerons les paramètres$\boldsymbol{\beta_0, \beta_1, \ldots, \beta_p}$du modèle de régression de façon à minimiser la somme des carrés des erreurs.

- Le coefficient de corrélation multiple$\mathbf{R^2}$, aussi appelé coefficient de détermination, nous indique le pourcentage de la variabilité de **Y**expliquée par les variables indépendantes$\mathbf{X_1, X_2, \ldots, X_p}$.

- Lorsqu'on ajoute une ou plusieurs variables indépendantes dans le modèle, le coefficient $\mathbf{R^2}$augmente.

- La question est de savoir si le coefficient$\mathbf{R^2}$augmente de façon significative.

- Notons qu'on ne peut avoir plus de variables indépendantes dans le modèle qu'il y a d'observations dans l'échantillon (règle générale: $n \geq 5p$)[6].

6.2.régression non-linéaire

On modélise les variables considérées comme des var (définies sur un espace probabilisé (Ω; A; p)), en gardant les mêmes notations par convention. À partir de celles-ci, le modèle de régression non-linéaire est caractérisé par : pour tout i ∈ (1;.... ; n),

- $(x_{1;i}; ... ; x_{p;i})$ est une réalisation du vecteur aléatoire réel $(X_1; .... ;X_p)$,
- sachant que $(X_1 .... X_p) = (x_{1;i}.....x_{p;i})$, yi est une réalisation de

  $Y i = f (x_{1;i} .....x_{p;i}.\beta_0.....\beta_p) + \varepsilon_i$

  oùε_i est une var indépendante de X1; ....; X p avec $E(\varepsilon_i) = 0$.

6.2.1. Modèle de régression polynomiale:

Un modèle de régression polynomiale est un modèle de régression non-linéaire avec f(X1 .... X p. β0.....βp) étant un polynôme en X1 .... X p de coefficients. β0.....βp
Cette représentation polynomiale peut être motivée par l'intuition, le contexte de l'expérience ou des critères statistiques précis (comme les résidus partiel présentés ci-après).

6.2.2. Modèle de régression logistique

On supposera que la variable Y à laquelle on s'intéresse est la survenue ou non d'une maladie, dont les deux catégories seront notées M + et M-.

Dans le cas d'une seule variable X explicative (équivalent d'une régression simple), le modèle s'écrit :$P(M |X)= f(X)= \frac{\exp(\propto+\beta X)}{1+\exp(\alpha+\beta X)}$

Il s'agit de la probabilité de maladie si la variable X est prise en compte et quand sa valeur est connue P (M + |X) se lit : probabilité de maladie si X. f (X) est la fonction logistique.
L'intérêt de cette fonction réside dans la simplicité de passage à l'estimation d'un odds-ratio (OR) ou rapport des cotes qui mesure la force de l'association entre la maladie M et une variable d'exposition. En effet, si l'exposition est codée en 0/1 (non exposé/exposé), le modèle permet d'arriver après simplification à OR = exp (β). Le coefficient Ede la variable d'exposition dans le modèle logistique est donc le logarithme de l'odds-ratio mesurant l'association entre cette variable et la maladie, ce qui permet d'interpréter facilement les résultats d'une régression logistique. De plus, la fonction logistique a une forme sigmoïde (fig. 1) qui correspond à une forme de relation souvent observée entre une « dose » d'exposition et la fréquence d'une maladie. Ainsi, le modèle logistique permet de décrire l'association entre le degré d'exposition à un facteur quantitatif et l'accroissement du risque de la maladie, car pour tous les degrés d'exposition, la valeur du modèle logistique reste entre

0 et 1 ce qui correspond à une probabilité de maladie. L'extension vers un modèle à plusieurs variables (régression multiple) se fait facilement :

$$P\ M\ |X\ X1\ \dots\ Xn = \frac{\exp\{(\propto + \sum_{i=1}^{n} \beta iXi)\}}{1+\exp\{(\propto + \sum_{i=1}^{n} \beta iXi)\}}$$

À chaque variable Xi est associé un coefficient βi et ORi (mesurant l'association entre Xi et M+) se calcule par exp (βi).

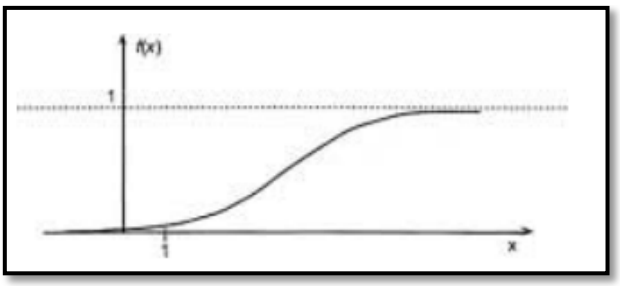


Figure I - 4: Représentation graphique de la fonction logistique.

7. Conclusion

Nous avons présenté dans ce chapitre dans sa première partie la fouille de données définition types et classifications, tandis que dans la seconde partie nous présentons la régression, l'équation et les différente forme de régression.

La régression linaire simple est l'un des type de régression le plus reconnue, nous abordons dans les prochaine chapitres les relations entre ce type de régression et big data architecture.

CHAPITRE II

Architecture de Big Data

1. Introduction

Face à un terme auquel nous sommes de plus en plus exposés, il est indispensable de comprendre l'ensemble des éléments qui compose le Big data. Effet de mode ou véritable bouleversement technologique, il est certain que cette expression anglophone ne laisse personne indifférent.

Ce chapitre a pour but d'éclaircir les notions fondamentales liées à cette masse historique de données : ses origines, son évolution, ses caractéristiques qui permettront par la suite d'aborder le reste du sujet de manière plus concrète.

2. Histoire de Big Data

Les années 1990 sont marquées par un fort contexte concurrentiel qui pousse les entreprises à innover en matière de stockage et d'utilisation de l'information. On assiste à l'arrivée de logiciels de type ERP (Entreprise Ressource Planning) aussi appelé "Progiciel de gestion intégrée".

Ces logiciels ont permis durant des années de stocker et de collecter de la donnée utile pour les postes de stratégie. Cependant, les modes de stockage (majoritairement en base de données relationnelles) ont par la suite montré leurs limites.

Vers la fin des années 1990, internet bouleverse l'économie mondiale, notamment avec la croissance des e-commerces et la montée en puissance d'une bulle financière qui prendra fin en 2000. Pour la première fois, l'information devient mondiale.

Post 2000 nous assistons à l'arrivée de technologies mobiles et tablettes qui modifient notre rapport à l'information. Tout le monde peut désormais accéder et partager de l'information partout, tout le temps. Les objets connectés accentuent ce phénomène. Les montres connectées, les appareils de soin, les moteurs de recherche ou encore les voitures autonomes représentent autant de "machines" qui génèrent de la data en continue.

Tout cela est en partie dû à l'émergence du Cloud Computing et de ses différentes couches (IaaS, PaaS, SaaS) qui permettent aux entreprises de générer et stocker de la donnée de plus en plus facilement. L'infrastructure et le logiciel deviennent commodité.

La forte diversité des objets permettant de générer de l'information induit une très grande **variété des données.** Les contenus sont aujourd'hui multiples : tweets, vidéos, données GPS, sons, communications…

Les entreprises du secteur se sont donc progressivement adaptées en proposant de nouvelles méthodes de stockage. Google publie en 2006 Map Reduce, une architecture de programmation efficace sous des fortes contraintes de volume. D'autres acteurs tel que Mongo DB, Apache Cassandra ou Redis apparaissent par la suite en se basant sur des systèmes de stockage non relationnels. Ces technologies sont une des réponses à la **forte variété** des données engendrée par le Big Data.[7]

3. Definition de Big Data

Le terme de Big Data a été évoquée la première fois par le cabinet d'études Gartner en 2008 mais la naissance de ce terme effective remonte à 2001 et a été évoquée par le cabinet Meta Group.

Il fait référence à l'explosion du volume des données (de par leur nombre, la vitesse à laquelle elles sont produites et leur variété) et aux nouvelles solutions proposées pour gérer cette volumétrie tant par la capacité à stocker et explorer, et récemment par la capacité à analyser et exploiter ces données dans une approche temps réel.[8]

Big data, littérairement les grosses données, est une expression anglophone utilisée pour désigner des ensembles de données qui deviennent tellement volumineux qu'ils en deviennent difficiles à travailler avec des outils classiques de gestion de base de données. Il s'agit donc d'un ensemble de technologies, d'architecture, d'outils et de procédures permettant à une organisation très rapidement de capter, traiter et analyser de larges quantités et contenus hétérogènes et changeants, et d'en extraire les informations pertinentes à un coût accessible.[9]

4. Infrastructures de Big Data

La croissance phénoménale de Big Data nécessite des infrastructures efficaces et de nouveaux paradigmes informatiques afin de répondre aux énormes volumes de données et à leur grande vitesse. Ci-après, nous présentons trois modèles d'infrastructure communs qui offrent d'excellents moyens de gérer Big Data.

4.1. Clusters

L'informatique en grappes consiste à connecter un ensemble d'ordinateurs à leurs logiciels habituels (Par exemple, des systèmes d'exploitation) ensemble via, communément, des

réseaux locaux (LAN) à Construire un système global. L'informatique en grappes a été l'une des principales sources Informatique distribuée. Le principal objectif de ce paradigme informatique est de fournir La performance et la disponibilité à un faible coût.

Au fil des ans, les outils et les paradigmes de l'informatique de cluster ont évolué pour atteindre une efficacité élevée.

De nos jours, la construction d'un cluster est devenue facile avec le soutien d'une planification efficace des tâches, Et l'équilibrage de charge. De plus, les utilisateurs du cluster peuvent s'appuyer sur un ensemble de Des outils dédiés tels que les systèmes d'exploitation Linux, les outils d'interface de passage de messages (MPI)

Et divers outils de surveillance pour exécuter un large éventail d'applications.

À l'ère de Big Data, il est courant pour les organisations de créer des clusters dédiés quiSe composent généralement de matériel de base. Ces clusters exécutent des plates-formes Big Data (comme Hadoop et No SQL), afin d'accueillir de manière significative de grands volumes de données. Ils Permettre en attendant, le traitement en temps réel et le degré élevé de disponibilité. De telles tailles de grappes Peut varier de dizaines de nœuds à des dizaines de milliers de nœuds.

Dans le domaine de l'informatique de haute performance (HPC), les superordinateurs qui contiennent des centaines de milliers de cœurs peuvent exécuter des applications hautes performances qui présentent des E / S élevées demandes. Ces superordinateurs ont des architectures massivement parallèles dédiées pour Les exigences élevées de calcul et les besoins de stockage croissants de HPC

4.2. grilles

Dans leur tentative de définir "Grid Computing", Foster et Kesselman qualifient un calcul Réseau comme «une infrastructure matérielle et logicielle fournissant des services fiables, cohérents, Un accès peu coûteux aux capacités de calcul haut de gamme.

Quatre ans plus tard, Ian Foster reformule sa définition comme liste de contrôle [60].

En conséquence, un Grille est un système qui:

- ✓ Fournit une coordination sur les ressources pour les utilisateurs sans contrôle centralisé.
- ✓ Fournit des protocoles et des interfaces génériques standard et ouverts.
- ✓ Offrir des qualités de service non triviales .Grid Computing s'adresse, en général, à la fédération et à la coordination des

Des sous-organisations hétérogènes et des ressources qui pourraient être dispersées géographiquement les régions éloignées. Elle vise à dissimuler la complexité de la

coordination des sous – systèmes Utilisateurs et les expose l'illusion d'un système global à la place.

4.3. Des nuages

L'informatique parallèle et distribuée évolue au fil des ans afin de mieuxLa qualité du service aux utilisateurs tout en assurant l'efficacité en termes de performance, de coût et d'échec tolérance. Un nouveau paradigme de l'informatique émergé est Cloud Computing. Dans ce paradigme, Logiciel et / ou matériel sont fournis en tant que service sur un réseau informatique, généralement Internet. Les services en nuage diffèrent dans leur niveau d'abstraction. Ils peuvent être classés en trois Niveaux: infrastructure, plate-forme et logiciel.

Depuis son émergence, Cloud Computing a été immédiatement adopté par les applications Big Data. Pour de nombreuses organisations, l'acquisition de ressources en mode Pay-as-You-Go À l'échelle de leur Big Data plates-formes lorsque nécessaire, a été longtemps nécessaire, et le nuage Fournisseurs offrent juste cela. De plus, la plupart des fournisseurs de Cloud offrent des plates-formes Big Data un service. Ils permettent aux clients d'exécuter leurs applications sans se soucier de la gestion de l'infrastructure et de son entretien coûteux. De nos jours, il est tout naturel Les entreprises, comme Netflix [27] qui offre un service vidéo à la demande, d'accueil et de gérer tous Leurs données dans le nuage (Amazon EC2 [9]).

Dans la section suivante, nous soulignons ce paradigme du Cloud Computing comme un Et excellent moyen pour Big Data.

5. Caractéristique de Big Data

Le Big Data (en français "Grandes données") regroupe une famille d'outils qui répondent à une triple problématiques : C'est la règle dite des 3V[10]

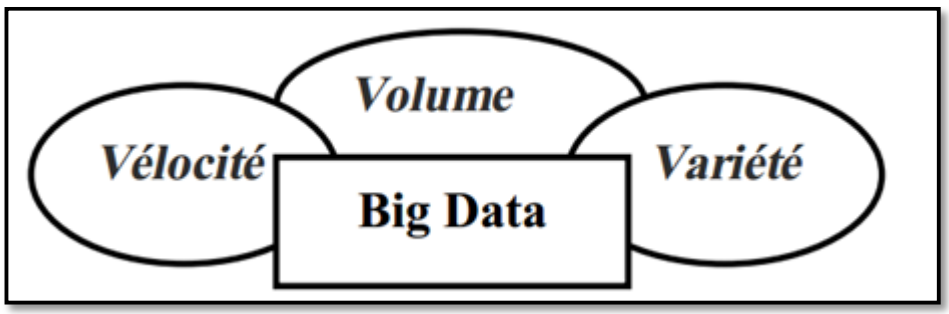


Figure II - 1: Le Big Data, les 3 V[9]

§ **Volume**, c'est le poids des données à collecter. Confrontées à des contraintes de stockage, les entreprises doivent aussi gérer le tsunami des réseaux sociaux. La montée en puissance des réseaux sociaux a accentué cette production de données.

§ **Variété**, l'origine variée des sources de données qui sont générées. premièrement, l'époque où les entreprises s'appuyaient uniquement sur les informations qu'elles détenaient dans leurs archives et leurs ordinateurs est révolue. De plus en plus de données nécessaires à une parfaite compréhension du marché sont produites par des tiers. Deuxièmement, les sources se sont multipliées : banques de données, sites, blogs, réseaux sociaux, terminaux connectés comme les Smartphones, puces RFID, capteurs, caméras... Les appareils produisant et transmettant des données par l'Internet ou d'autres réseaux sont partout, y compris dans des boîtiers aux apparences anodines comme les compteurs électriques.

§ **Vélocité**, la vitesse à laquelle les données sont traitées simultanément. à l'ère d'internet et de l'information quasi instantanée, les prises de décision doivent être rapides pour que l'entreprise ne soit pas dépassée par ses concurrents. La vélocité va du batch au temps réel.

A ces « 3V », les uns rajoutent la **visualisation** des données qui permet d'analyser les tendances ; les autres rajoutent la **variabilité** pour exprimer le fait que l'on ne sait pas prévoir l'évolution des types de données.

6. Processus de chargement et de collecte de données dans Big Data

La couche responsable du chargement de données dans Big Data, devrait être capable de gérer d'énorme volume de données, avec une haute vitesse, et une grande variété de données. Cette couche devrait avoir la capacité de valider, nettoyer, transformer, réduire (compression), et d'intégrer les données dans la grande pile de données en vue de son traitement. La Figure illustre le processus et les composants qui doivent être présent dans la couche de chargement de données. [11][12]

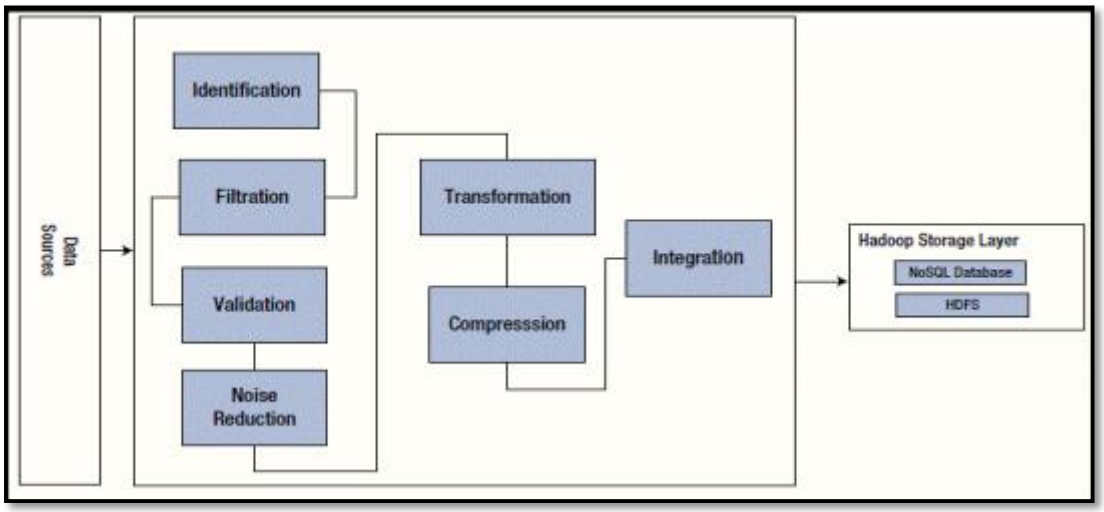


Figure II - 2: Couche de chargement des données dans le Big Data [11]

La couche de chargement de données de Big Data collecte les informations pertinentes finales, sans bruit, et les charge dans la couche de stockage de Big Data (HDFS ou No SQL base).Elle doit inclure les composants suivants :

- **Identification** des différents formats de données connues, par défaut Big Data cible les données non structurées ;
- **Filtration et sélection** de l'information entrante pertinente pour l'entreprise
- **Validation** et analyse des données en permanence ;
- **Réduction** de bruit implique le nettoyage des données en supprimant le bruit ;
- **La transformation** peut entrainer le découpage, la convergence, la normalisation ou la synthèse des données ;
- **Compression** consiste à réduire la taille des données, mais sans perdre de la pertinence des données ;
- **Intégration** consiste à intégrer l'ensemble des données dans le stockage de données de Big Data (HDFS ou NoSQL base).

7. Architecture Big Data

On distingue principalement les couches suivantes :

- Couche matériel (infrastructure Layer) :peut-être des serveurs virtuels VMware, ou des serveurs lame blade ;

- Couche stockage (Storage layer) : les données seront stockées soit dans une base No SQL, ou bien directement dans le système de fichier distribué ou les Data warehouse
- Couche management et traitement : on trouve dans cette couche les outils de traitement et analyse des données comme MapReduce ou Pig ; [11]
- Couche visualisation : pour la visualisation du résultat du traitement.

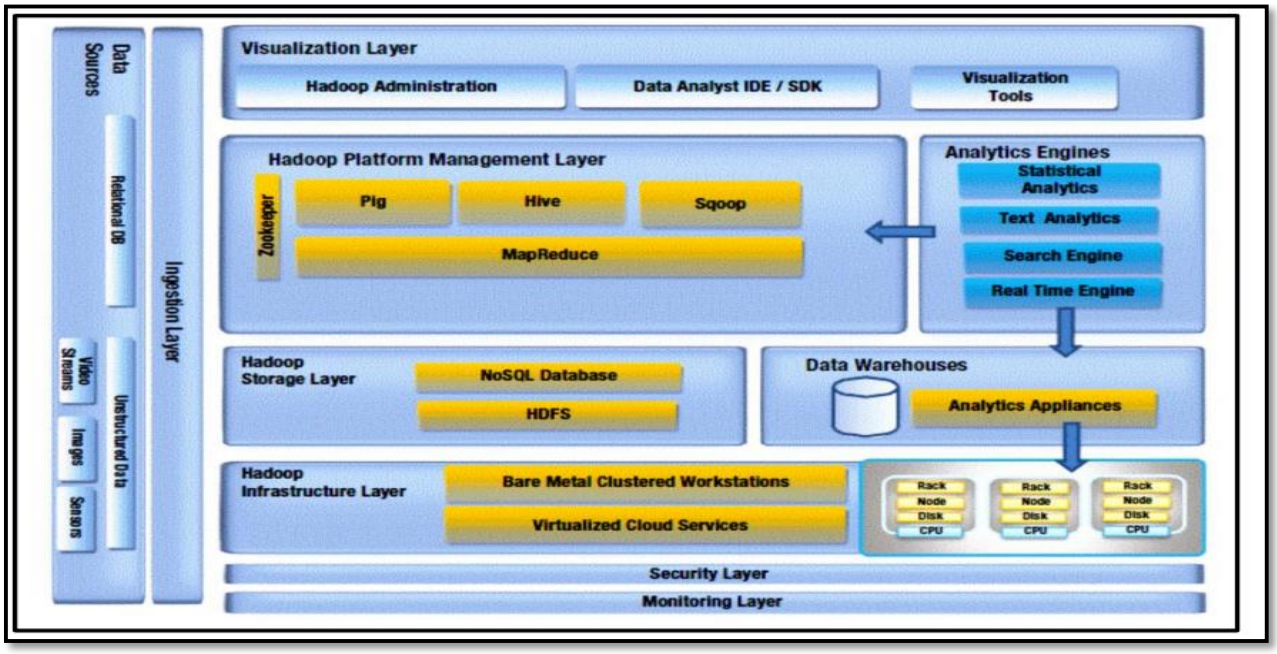


Figure II - 3: Architecture de Big Data [11]

8. Avantages de l'architecture Big Data

Plusieurs avantages peuvent être associés à une architecture Big Data, nous pouvons citer par exemple :

- **Evolutivité (scalabilité) :** Quelle est la taille que devra avoir votre infrastructure ? Combien d'espace disque est nécessaire aujourd'hui et à l'avenir ? le concept Big Data nous permet de s'affranchir de ces questions, car il apporte une architecture scalable.
- **Performance :** Grâce au traitement parallèle des données et à son système de fichiers distribué, le concept Big Data est hautement performant en diminuant la latence des requêtes.
- **Coût faible :** Le principal outil Big Data à savoir Hadoop est en Open Source,en plus on n'aura plus besoin de centraliser les données dans des baies de stockage souvent excessivement chère, avec le Big Data et grâce au système de fichiers distribués les disques internes des serveurs suffiront.
- **Disponibilité :** On a plus besoin des RAID disques, souvent coûteux . L'architecture Big Data apporte ses propres mécanismes de haute disponibilité.

9. Dimensions du Big Data

Il est difficile de visualiser l'ampleur de cette nouvelle révolution numérique. Pour pouvoir s'en faire une idée un peu plus précise, Meta Group présenta en 2001

un rapport faisant état 3 de dimensions à travers lesquelles l'amoncellement et le traitement de données diverses pouvaient être représentés : le volume, la vitesse et la variété. Le terme de Big Data n'était alors pas encore apparu et peu de gens étaient conscients que chacune de leur trace laissée sur Internet était enregistrée et stockée. Néanmoins, la définition du Big Data vient s'enrichir d'autres termes et interrogations qui en découlent.

10. Différence entre BI (Business intelligence) et Big Data

La méthodologie BI traditionnel fonctionne sur le principe de regrouper toutes les données de l'entreprise dans un serveur central (Data warehouse ou entrepôt de données). Les données sont généralement analyse en mode déconnecté.
Les données sont généralement structurées en SGBDR avec très peu de données non structurées. [11][12]
Une solution Big Data, est différente d'une BI traditionnel dans les aspects suivants :

- ✓ Les données sont conservées dans un système de fichiers distribué et scalable plutôt que sur un serveur central ;
- ✓ Les données sont de formats différents, à la fois structurées ainsi que non structurées
- ✓ Les données sont analysées en temps réel ;
- ✓ La technologie Big Data s'appuie sur un traitement massivement parallèle (concept MPP).

11. Algorithme MapReduce

11.1. Le paradigme MapReduce

MapReduce est un modèle de programmation massivement parallèle adapté au traitement de très grandes quantités de données [13]. Ce modèle de programmation s'articule autour de deux étapes principales Map et Reduce (cf. Figure II - 4). Dans l'étape Map le nœud (machine) à qui est soumis un problème, le découpe en sous-problèmes, et les délègue à d'autres nœuds (qui peuvent en faire de même récursivement). Dans l'étape Reduce les nœuds les plus bas font remonter leurs résultats aux nœuds parents qui les avaient sollicités. À la fin du processus, le nœud d'origine peut recomposer une réponse au problème qui lui avait été

soumis. MapReduce s'appuie sur la manipulation de couples (clé, valeur), la tâche initiale Map est un couple (clé, valeur), et chacune des tâches intermédiaires également. Chacune de ces dernières retournent un résultat sous la forme d'un couple (clé, valeur). La fonction Reduce combine tous ces résultats en un couple (clé, valeur) unique.

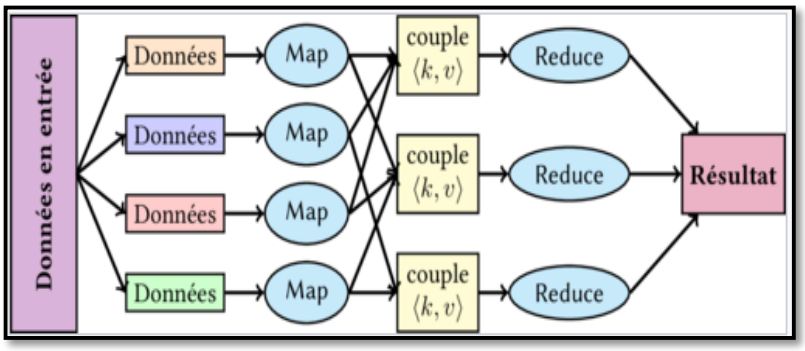


Figure II - 5: Schéma du paradigme MapReduce.

L'implémentation la plus populaire de MapReduce est le Framework Hadoop, qui permet aux applications d'être exécutées sur de larges clusters déployés sur des machines à faible coût, ouvrant ainsi la voie aux solutions basées sur ce paradigme vers les architectures du type Cloud [14]. D'autres implémentations du paradigme MapReduce sont disponibles pour différentes architectures, telles que les architectures multi-cores [15], les architectures à multiples machines virtuelles [16], les environnements de Grid Computing [17] ou encore les environnements mobiles [18].

11.2. Principe de fonctionnement de MapReduce

MapReduce joue un rôle majeur dans le traitement des grandes quantités de données. La distribution des données au sein de nombreux serveurs permet le traitement parallélisé de plusieurs tâches portant chacune sur des morceaux de fichiers. La fonction Map accomplit une opération spécifique sur chaque élément. L'opération Reduce combine les éléments selon un algorithme particulier, et fournit le résultat. Soulignons que le principe de délégation peut être récursif : les nœuds à qui sont confiées des tâches peuvent aussi déléguer des opérations à d'autres nœuds.

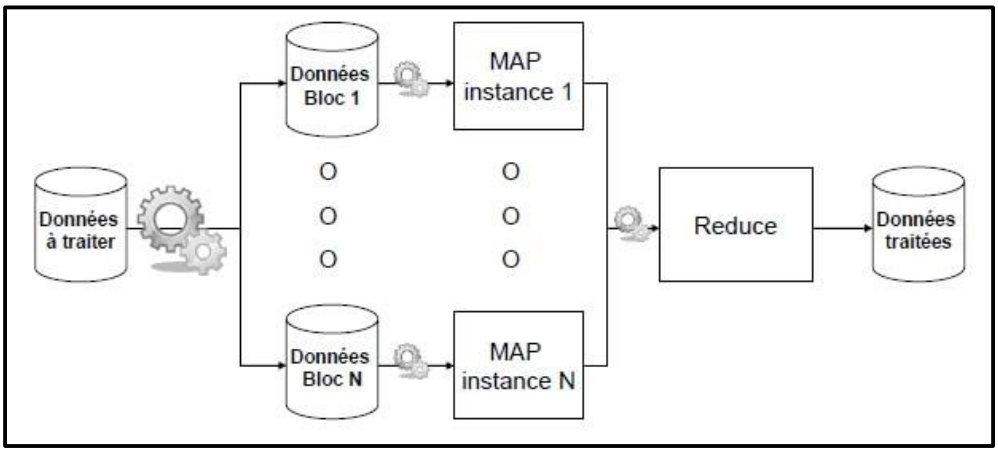
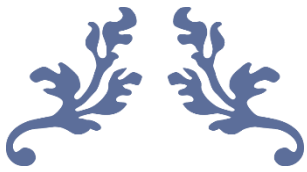


Figure II - 6: Principe de fonctionnement de MapReduce [19].

12. Conclusion

Le Big Data, la gestion des grands volumes de données à un champ d'application très vaste et varié. Dans un futur proche le Big Data serait très utile dans la création de nouvelles entreprises, de l'amélioration de la satisfaction clients, la détection d'épidémie, la détection de foyer de tension …etc.

Selon un rapport publié par Gartner le Big data est la technologie qui va générer le plus d'emploi dans l'informatique dans les trois (03) années venir.



CHAPITRE III

Modélisation

1. Introduction

Dans ce chapitre nous présentons notre conception, qui s'articule sur le modèle mathématique, les algorithmes Map/reduce proposé, les flux de données de notre architecture .et les diagrammes qui décrit les aspects statique et dynamique de notre proposition.

2. Big data et Map Reduce Algorithm

Map Reduce [20] primitives implémente une traitement parallèle, il est composé par deux algorithmes, Map et Reduce, l'algorithme Map prend un ensemble de données et convertir en un autre ensemble de données, il prend une paire de (clé, valeur) et émet (clé, valeur) au reduce phase . Reduce phase prend le résultat de map phase .

Map reduce algorithme constitué par master Job tracker, et un ensemble de serveur esclave appelé Task Tracker [21] [22]. Hadoop [23] fournissent Map Reduce run times avec tolérance de panne et support de flexibilité dynamique.

3. Algorithm Map reduce :

Map/Reduce [24, 25] est un modèle de programmation qui permet le développement et le test de programmes dédies a l'analyse des grandes masses de données distribuées sur un ensemble de nœuds. Son objectif est d'automatiser le parallélisme et la distribution du calcul sans exiger (de l'utilisateur) une supervision du système ni une expertise dans le calcul parallèle. Etant donné que le système gère l'exécution parallèle, la coordination et l'échec d'exécution des taches, le rôle du programmeur est de définir et d'implémenter les deux fonctions Map et Reduce. La Figure III - 1 [24] résume les étapes d'exécution d'un programme Map/Reduce qui sont :

1- La phase Map : chaque Mapper (nœud qui exécute la fonction Map) travaille sur un ou plusieurs morceaux des données initiales qui se trouvent dans son nœud. Suivant le traitement décrit par la fonction Map, les Mappers produisent des résultats sous formats de paires (clé, valeur).

2- Les paires (clé, valeur) produites par les différents Mapper sont regroupées et triées suivant leurs clés. Ensuite, elles sont dirigées vers les différents nœuds Reduce de façon que toutes les paires qui ont la même clé soient dans le même nœud Reduce.

3- Chaque Reducer traite les valeurs associées à chaque clé à la fois. Ce traitement est fixe par la fonction Reduce écrite par le programmeur.

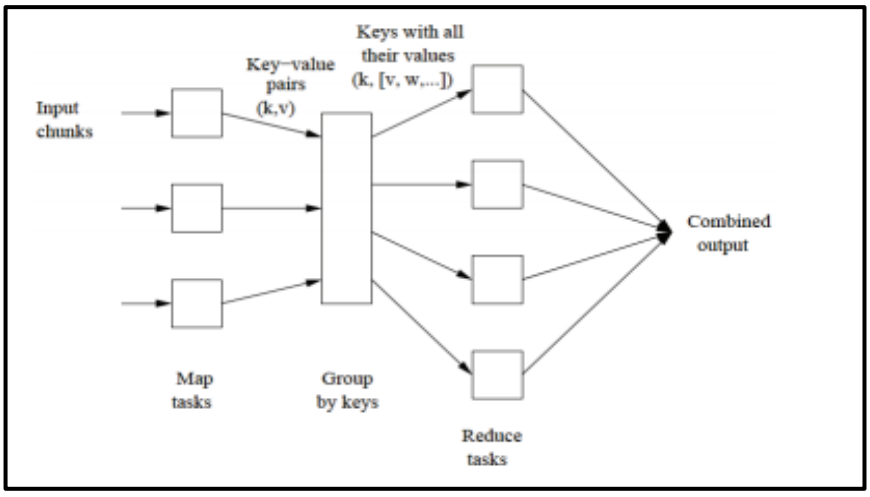


Figure III - 1: Etapes d'exécution d'un algorithme Map/Reduce [24].

Pour résumer, la fonction Map doit d'écrire le traitement qui peut être exécute sur des parties des données initiales indépendamment des résultats sur les autres parties. Alors que la fonction Reduce décrit comment agréger des résultats de la fonction Map pour atteindre l'objectif final.

4. Travaux similaires

Il existe plusieurs travaux de recherche intéressés par régression, linéaire ou courbe dans les grandes données [26, 27, 28, 29,30].Plusieurs orienté à proposer des approches mathématique de régression dans les grandes données telles que [26,28,29,30] .Autres visant à propose des algorithmes MapReduce et son mises en œuvre dans les grands systèmes de données[27].

4.1.modèles linéaires

Dans [24] a présenté une analyse de régression linéaire multiple en utilisant divisé où forme de régression est:

$$y = a1x1 + a2x2 + ...amxm + b$$

Les auteurs utilisent des données d'échantillonnage aléatoire pour gros volumes de données divisé en sous échantillons. Ils considèrent tous les attributs ont une chance égale d'être sélectionné dans l'échantillon .La figure III - 2.

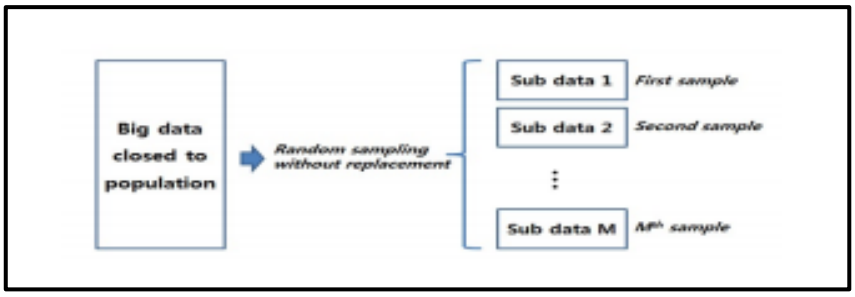


Figure III - 2: Diviser de grandes données en utilisant la méthode d'échantillonnage [26]

Dans [27] auteur présente un moyen de résoudre la régression linéaire dans les grandes données, ils proposent un algorithme MapReduce exprimé à la moindre erreur carrée, pour la mise en œuvre utilisent la bibliothèque R-Studio et Rhadoop.

4.2. K-means algorithme :

k-means est un algorithme itératif qui minimise la somme des distances entre chaque objet et le centroïde de son cluster. La position initiale des centrio les conditionne le résultat final, de sorte que les centroïdes doivent être initialement placés le plus loin possible les uns des autres de façon à optimiser l'algorithme. K-means change les objets de cluster jusqu'à ce que la somme ne puisse plus diminuer .Le résultat est un ensemble de clusters compacts et clairement séparés, sous réserve qu'on ait choisi la bonne valeur du nombre de clusters[31]

Algorithm 1: K-MEANS algorithm

Input: k,the number of cluster. D,data containing n attribute.
Output: A, a set of k cluster

1 Choose k attribute from D
2 **repeat**
3 (re)assign each attribute to the cluster to which the attribute is the most similar, based on the mean value of the attributes in the cluster
4 update the cluster means, i.e., calculate the mean value of the attributes for each cluster;
5 **until** *No change*

Figure III - 3: Algorithm K-means[31].

5. Proposition

Dans cette partie, nous présentons notre proposition modèle de mathématiques, algorithme de map, reduce l'algorithme et l'architecture du flux de données.

5.1. Le Modèle mathématique :

Supposant que les données X={x1,x2,…xn}.Big data architecture diviser les données en m sous données $\{x^1,x^2,\ldots x^m\}$.la première étape dans le modèle mathématique consiste à construire pour chaque sous données$\{x^1,x^2,\ldots x^m\}$. leurs modèle linaire$\{Z^1,Z^2,\ldots Z^m\}$.

et $Z^i=\{z_0^i, z_1^i, \ldots z_l^i\}$, l'expression de modèle linaire est :

$$y^i = a_0^i z_0^i + a_1^i z_1^i \ldots + a_l^i z_l^i + b^i$$

Le résultat de cette étape retourne les vecteurs $v=\{v^1,v^2,\ldots v^m\}$ dont $v^i = (a_0^i, a_1^i \ldots + a_l^i, b^i)$.

On applique k-means algorithme sur l'ensemble des vecteurs afin de trouver la meilleure solution de la régression linaire.

5.2. Description général de l'algorithme map/reduce pour la régression linaire :

Notre algorithme composé par MapReduce algorithme, Map phase consiste à traiter les données de chaque nœud, dans la partie reduce, s'intéressent par les transformations de données de chaque nœud en régression linaire.

- **phase map :**

Phase MAP pour diviser les données sur le nœud et retourne le résultat sous la forme de pair clé et valeurs pour chaque nœud.

Il est divisé les données sur un ensemble de nœuds .La formule ci-dessous montre comment diviser les données.

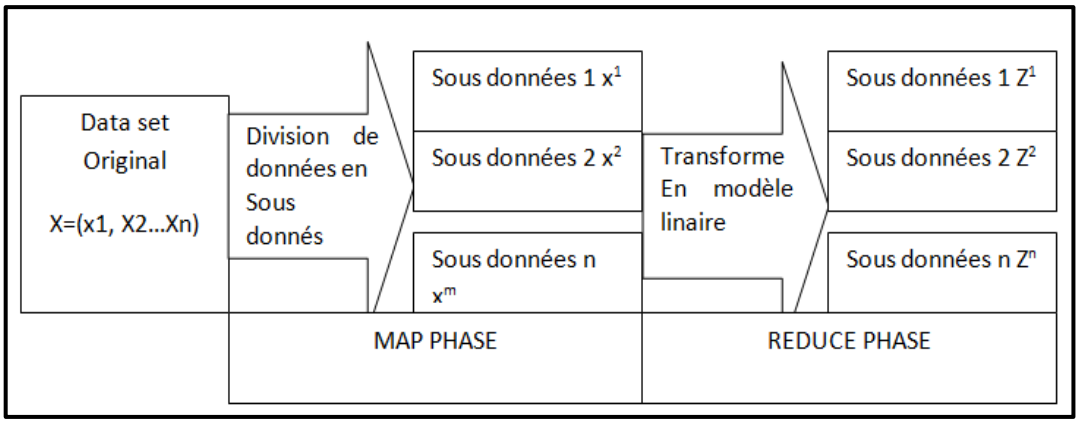


Figure III - 4: MAP/REDUCE actions.

- **phase reduce :**

On applique reduce algorithme pour chaque sous données de chaque nœud, reduce algorithme est une fonction qui permet de transformer les données sous une description linaire figure III.6.

```
Algorithme mapreduce(d : donnée)
clé : réel
Début
Phase map
Clé ← d[x1]
Return (Keyval(clé,d))
Fin.
Phase reduce(key ,val)
Ecrire(''reduce '')
Ecrire (d)
X ← reg(d[x1])
Y ← reg (d[x1])
Liste ← list(X,Y)
Return (Keyval(clé,Liste )
fin.
```

Figure III - 5: Algorithme Map Reduce.

- **Fonction de la régression linéaire**

Pour chaque nœud l'équation de la régression linaire décrit sous la forme suivante y=a1z1+a2z2+…anZn+b .(ou a1,a2,…b) sont les coefficient de la régression. Figure III-6.

```
Algorithme  Reg(d : donnée)
X,Y :[1...n]
Base: [1...n][1...n]
D : String ; s1,s2 : réel
Début
X ← crée une matrice à nrow lignes et 1 colonnes.
Y ← crée une matrice à nrow lignes et 1 colonnes.
Base ←crée un data frame(x , y)
D ← appliquer le régression linaire (y , base)
S1 ← D.coefficients[1]
S2 ←D. coefficients[2]
List1 ← list(s1,s2)
Return(List1)
fin.
```

Figure III - 6 : Notre fonction de régression linaire.

5.3. Algorithme k-means pour Map/Reduce:

Après l'exécution de Map/reduce qui retourne les régressions linaire de chaque nœud, nous appliquons k-means Map/reduce algorithme, le rôle de cette taches est d'extraire k-cluster des résulta des régressions linaires.

Les k-clusters sont les clusters qui mieux représenter le problème de la régression linaire.

```
Algorithme  mapreduce-kmeans (d : donnée)
Phase reduce (key , val)
Ecrire (''reduce '');
Ecrire (''val'');
Km1 ← kmeans(val,3,nstart= 100)
Ecrire (" k-means  resultat with 3 cluster ")
Plot(km1)
Liste ← km1 ;
Return (KeyVal (cl,Liste);
Fin .
```

Figure III - 7 : algorithme mapreduce pour le K-means.

6. Flux de données de notre architecture

Le flux de données Figure III - 8, présente les nœuds de données (Data Node), et les algorithmes s'exécutent sur celui-ci. L'algorithme (Map $Algo_1$, Map $algo_2$, ... Map algo $_m$) exécutent dans chaque nœud afin d'extraire la liste de clé et valeur des données. L'algorithme de réduction de la phase (algorithme Reduce) et retourne la régression linaire de chaque nœud, En fin nous appliquons un algorithme k-means Map/Reduce algorithme qui nous donne les k- modèle linaire sous la forme de cluster.

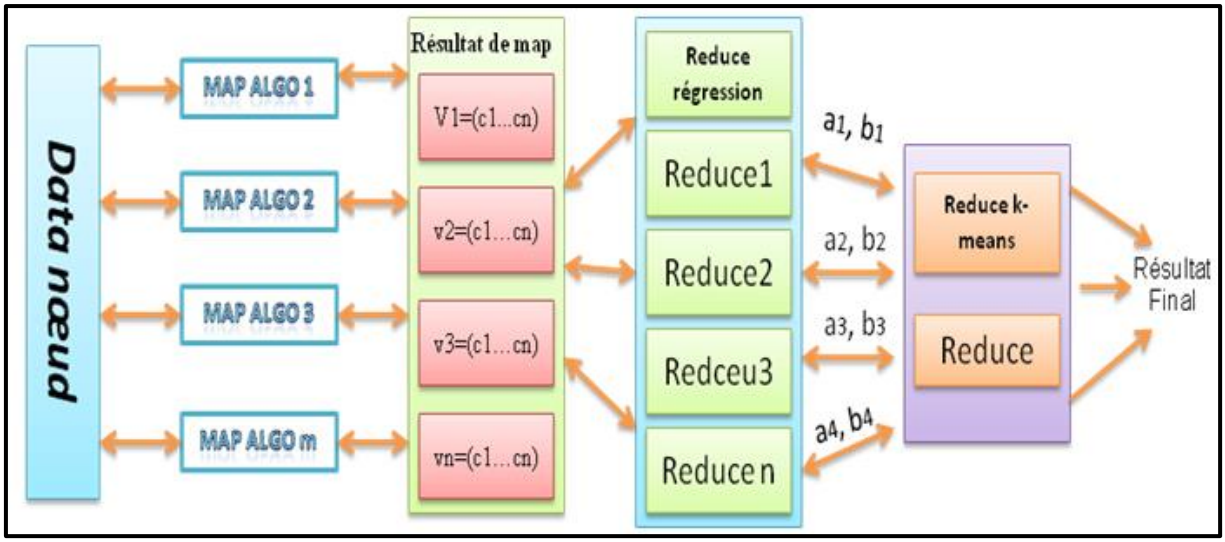


Figure III - 8 : Architecture du flux de données.

A. **data nœud :** C'est la première étape où nous entrons les données.
B. **map algo: Deuxième** Map à ce stade, nous divisons les données et sélectionnée les clés et calcule les valeurs.
C. **résultat de map :** Cette phase présente les résultats de chaque map phase des clés et ses valeurs.
D. **reduce algo:** les résultats de chaque map algorithme transforme en model linaire.
E. **Map reduce k-means :** Enfin, on applique la méthode k-means dans Map /reduce sur les résultats des régressions.

7. Model UML

UML 2.0 comporte treize types de diagrammes représentant autant de vues distinctes pour représenter des concepts particuliers du système d'information. Ils se répartissent en deux grands groupes (diagrammes structurels ou statiques, diagrammes comportementaux ou dynamiques).

Un diagramme UML est une représentation graphique, qui s'intéresse à un aspect précis du modèle ; c'est une perspective du modèle. Chaque type de diagramme UML possède une

structure (les types des éléments de modélisation qui le composent sont prédéfinis) et véhicule une sémantique précise (il offre toujours la même vue d'un système [32].

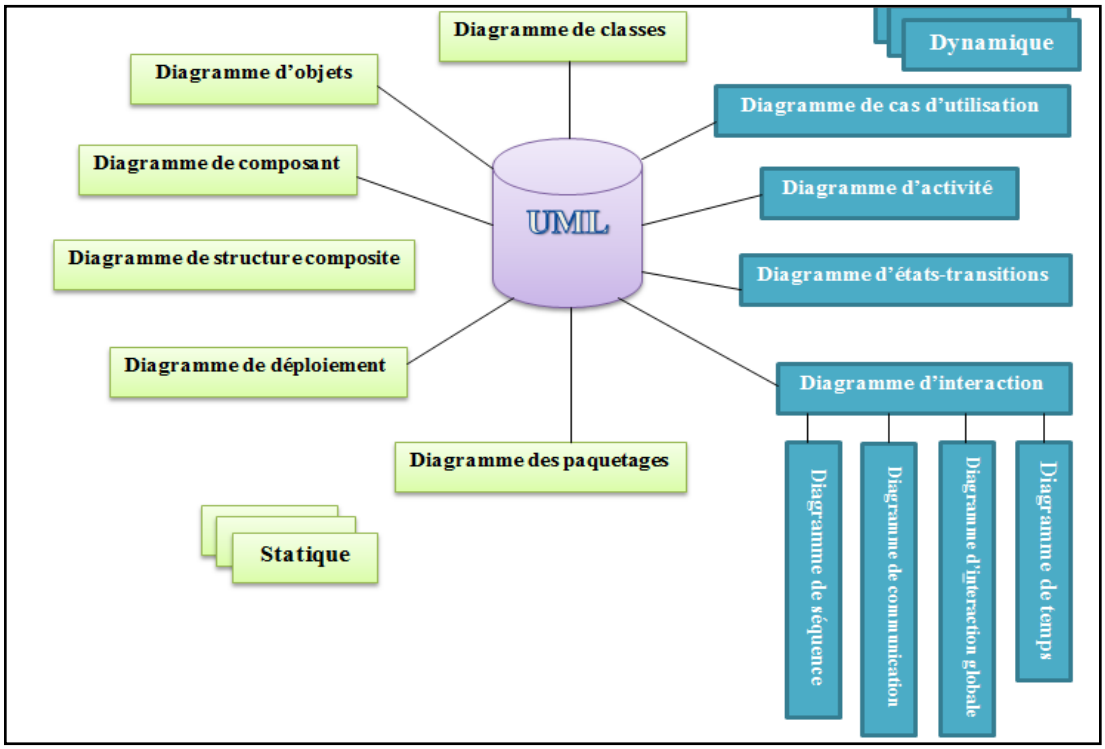


Figure III - 9: Organigramme de l'UML.

7.1.Diagramme de séquence :

Ce schéma Diagramme de séquence en UML illustre les étapes de la conduite des opérations dans le système

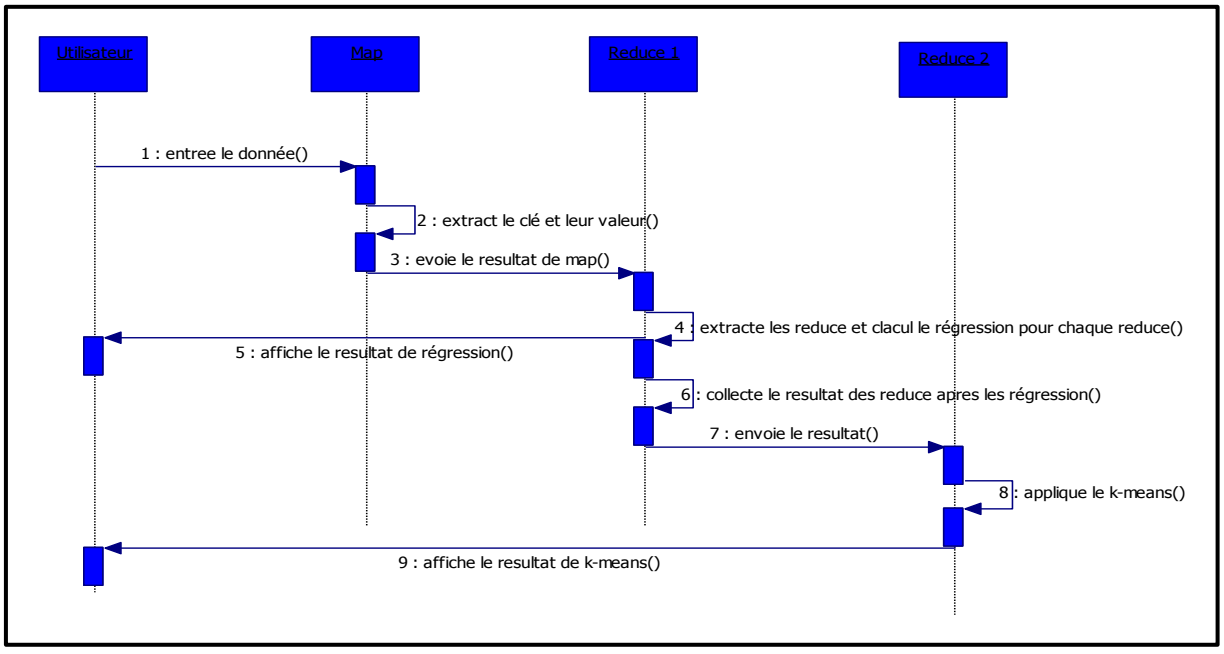


Figure III - 10 : Diagramme séquence.

7.2.Diagramme de class :

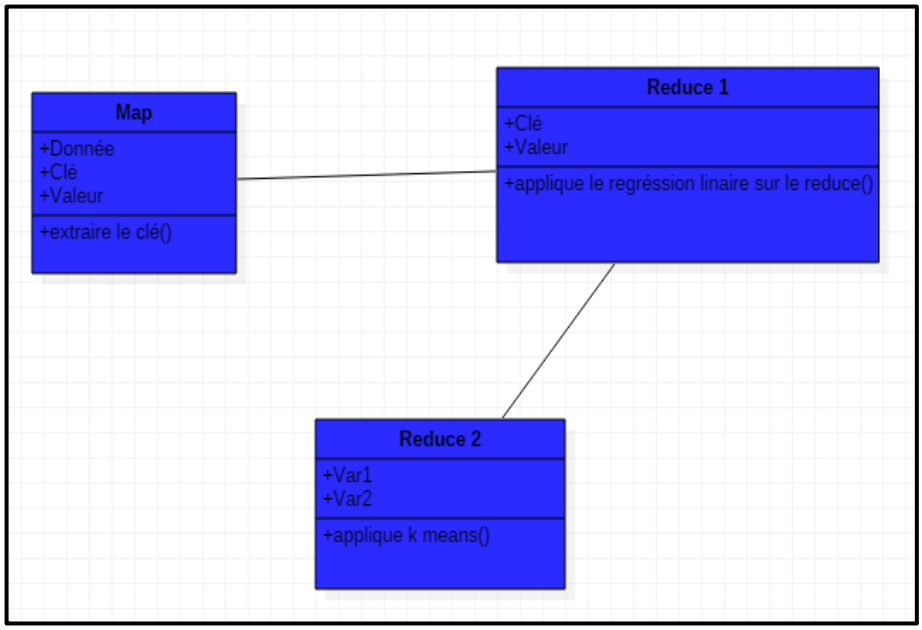
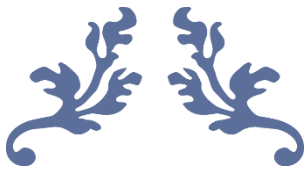


Figure III - 11 : Diagramme de classe.

8. conclusion

Dans ce chapitre nous avons donné une spécification sur notre conception du système, cette phase représente l'une des phases les plus importantes dans le processus de développement d'un logiciel. Elle décrit le système de point de vue général et détaillée pour comprendre et assister la phase de programmation .Nous avons vu dans ce chapitre l'architecture globale du système et les différente Map Reduce algorithme.

Dans le prochain chapitre, nous allons illustrer la réalisation de notre système en représentant quelques interfaces et quelques résultats obtenus, avec les structures des données qui sont choisis pour implémenter ce système.



CHAPITRE IV

Expérimentation et Validation

1. Introduction

Ce chapitre, consacré à la réalisation et la mise en œuvre de notre proposition sous studio R plus avec le package rmr2.

2. Environnement de travail

2.1.Environnement R

R est une suite intégrée d'équipements logiciels pour la manipulation de données, le calcul et l'affichage graphique. il comprend une installation de traitement des données efficace et stockage ,une série d'opérateurs pour les calculs sur les tableaux, dans des matrices particulières, une grande collection cohérente et intégrée d'outils intermédiaires pour l'analyse des données, installations graphiques pour l'analyse des données et l'affichage soit à l'écran ou sur papier, et un bien développé, langage de programmation simple et efficace qui comprend conditionnels, les boucles, les fonctions récursives définies par l'utilisateur et d'entrée et de sortie des installations.

Le terme « environnement » vise à caractériser comme un système entièrement planifié et cohérent, plutôt que d'une accumulation progressive d'outils très spécifiques et rigides, comme cela est souvent le cas avec d'autres logiciels d'analyse de données.

R, comme S, est conçu autour d'un véritable langage informatique, et permet aux utilisateurs d'ajouter des fonctionnalités supplémentaires en définissant de nouvelles fonctions. Une grande partie du système lui-même est écrit dans le dialecte R de S, ce qui le rend facile pour les utilisateurs de suivre les choix algorithmiques faits. Pour les tâches informatiquement intensives, C, C ++ et Fortran peuvent être liés et appelé au moment de l'exécution. Les utilisateurs avancés peuvent écrire du code C pour manipuler des objets R directement.

De nombreux utilisateurs pensent de R en tant que système statistique. Nous préférons penser d'un environnement dans lequel sont mis en œuvre des techniques statistiques. R peut être étendu (facilement) par paquets. Il y a environ huit paquets fournis avec la distribution de R et beaucoup d'autres sont disponibles dans la famille CRAN de sites Internet couvrant un très large éventail de statistiques modernes.

R a son propre format de documentation comme Latex, qui est utilisé pour fournir une documentation complète, à la fois en ligne dans plusieurs formats et sur papier.[33]

2.2.R Studio

RStudio est livré en deux versions : RStudio Desktop, pour une exécution locale du logiciel comme tout autre application, et RStudio Server qui, lancé sur un serveur Linux, permet

d'accéder à RStudio par un navigateur web. Des distributions de R Studio Desktop sont disponibles pour Microsoft Windows, OS X et GNU/Linux.

RStudio a été écrit en langage C++, et son interface graphique utilise l'interface de programmation Qt.

Depuis la version 1.0 (novembre 2016), RStudio intègre la possibilité d'écrire des notebooks combinant de manière interactive du code R, du texte mis en forme en markdown et des appels à du code Python ou Bash [34].

2.3. Rhadoop

L'intégration entre R et Hadoop pour traiter de gros volumes de données Il y a maintenant un grand nombre de paquets de R ou des scripts pour l'analyse de traitement et des données. Leur utilisation avec Hadoop normalement besoin de les réécrire en Java, le langage naturel pour Hadoop, mais l'activité de réécriture peut conduire à de nombreuses erreurs. Par conséquent, il est plus efficace pour l'interface système Hadoop avec R afin que nous ils peuvent travailler avec des scripts écrits en données R et stockées avec Hadoop.

2.4.rmr2

Rmr est un package R qui permet aux développeurs R d'effectuer l'analyse statistique dans R via la fonctionnalité MapReduce de Hadoop sur un cluster Hadoop. Avec l'aide de ce package, le travail d'un programmeur R a été réduit, où il suffit de diviser leur logique d'application dans la carte et de réduire les phases et de les soumettre avec les méthodes Rmr. Après cela, le Rmr appelle l'API MapReduce en streaming de Hadoop avec plusieurs paramètres de travail tels que le répertoire d'entrée, le répertoire de sortie, le mappeur, le réducteur, etc., pour effectuer le travail R MapReduce sur le cluster Hadoop. Ce paquet devrait être installé sur chaque nœud du cluster [35].

- **Forfaits requis pour l'installation**

Nous avons besoin de plusieurs paquets R pour installer R avec Hadoop. La liste des paquets est la suivante [34]:

- ✓ rJava
- ✓ RJSONIO
- ✓ itertools
- ✓ digest
- ✓ Rcpp
- ✓ httr

- ✓ functional
- ✓ devtools
- ✓ plyr
- ✓ reshape2

2.4.1. Installation de rmr2

Après avoir téléchargé le package rmr2 du site [36] que nous avons installé à l'aide du code indiqué dans l'image suivant

```
> install.packages("C:/Users/Admin/Downloads/rmr2_3.3.1.zip", repos = NULL, type = "win.binary")
```

Figure IV - 1: Instruction de installation de package rmr2.

3. Importation de la base de données

Nous avons sélectionné la base de données appelée " Universel Bank " pour le travail dans le projet et sont du type Excel contenant 14 colonne et élément 5000[37].

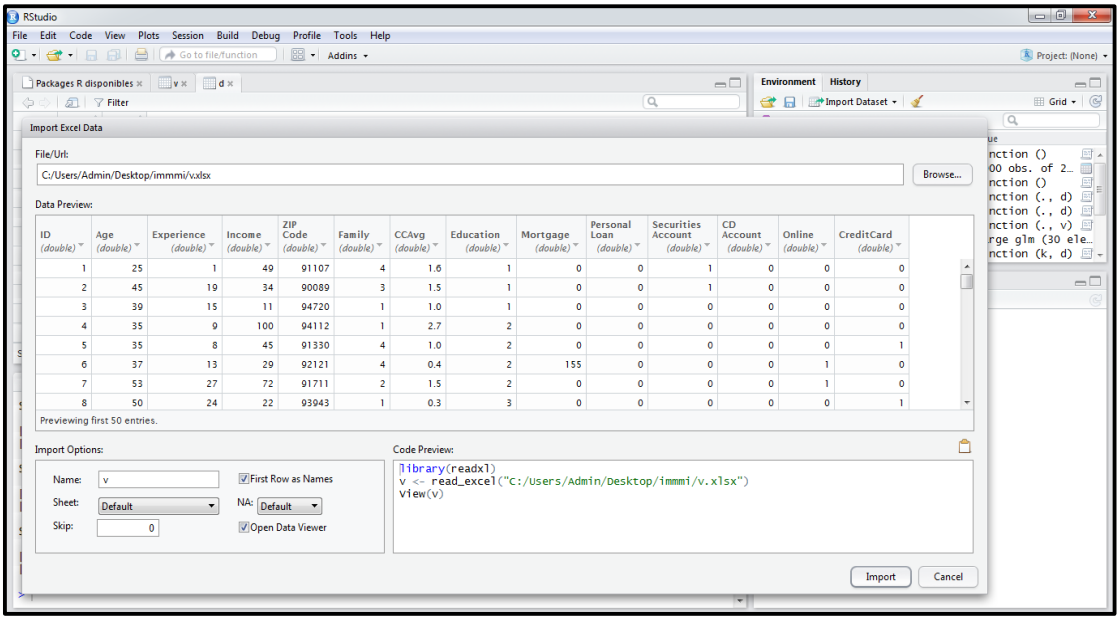


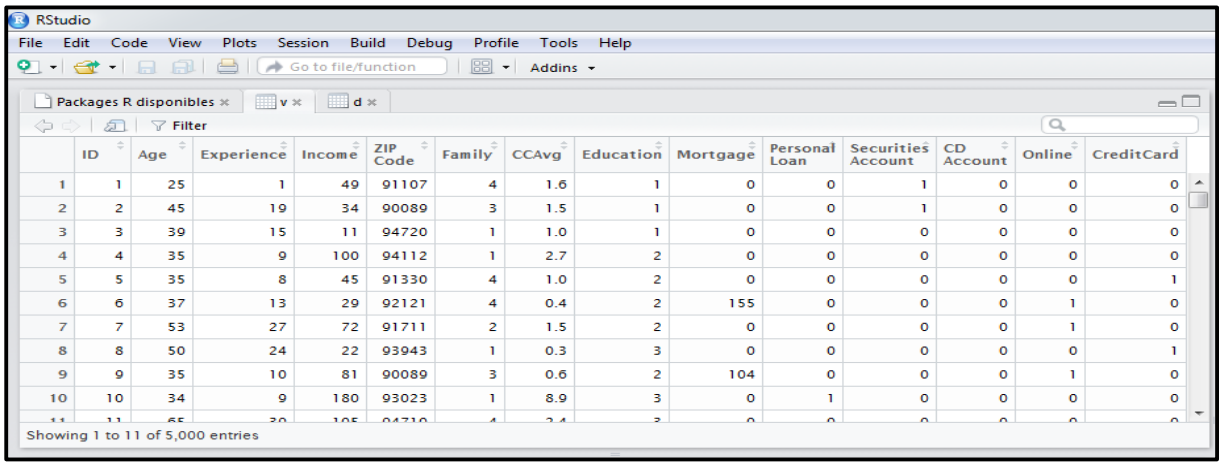
Figure IV - 2: Import la base de données.

➢ **affichage de la base de donnée:**

Pour l'affichage de base de données utilise l'instruction suivant :

```
> View(v)
```

Figure IV - 3: Instruction de l'affichage de base de données.



	ID	Age	Experience	Income	ZIP Code	Family	CCAvg	Education	Mortgage	Personal Loan	Securities Account	CD Account	Online	CreditCard
1	1	25	1	49	91107	4	1.6	1	0	0	1	0	0	0
2	2	45	19	34	90089	3	1.5	1	0	0	1	0	0	0
3	3	39	15	11	94720	1	1.0	1	0	0	0	0	0	0
4	4	35	9	100	94112	1	2.7	2	0	0	0	0	0	0
5	5	35	8	45	91330	4	1.0	2	0	0	0	0	0	1
6	6	37	13	29	92121	4	0.4	2	155	0	0	0	1	0
7	7	53	27	72	91711	2	1.5	2	0	0	0	0	1	0
8	8	50	24	22	93943	1	0.3	3	0	0	0	0	0	1
9	9	35	10	81	90089	3	0.6	2	104	0	0	0	1	0
10	10	34	9	180	93023	1	8.9	3	0	1	0	0	0	0

Showing 1 to 11 of 5,000 entries

Figure IV - 4: affichage de la base de donnée.

Après l'entrée de la base de données e nous choisissons seulement deux commencé à travailler sur les deux ' iconom' et 'personnel loan ' et combiné avec certaines données Frame

	X1	X2
1	49	0
2	34	0
3	11	0
4	100	0
5	45	0
6	29	0
7	72	0
8	22	0
9	81	0
10	180	1
11	105	0

Showing 1 to 11 of 5,000 entries

Figure IV - 5: affichage de donnée sélectionnée

4. programmation de les algorithms

4.1. programmation de régression

Appliquer l'algorithme de régression pour la sortie est le a , b de équation rectal

```
> reg =  function(.,classe1){
      x<- matrix(rnorm(classe1) ,ncol = 1)
      y<- as.matrix(rnorm(classe1))
      base<- data.frame("y"=y,x)
      d<- lm(y~.,data = base)
      s1<- d$coefficients[1]
      s2 <- d$coefficients[2]
      list1 <- list(s1, s2)
      return(list1)
  }
```

Figure IV - 6: fonction de régression.

4.2.programmation d'algorithme mapreduce pour régression linaire

- **algorithme map :**

L'image suivante représente l'implémentation de l'algorithme Map.

```
> map_im = function(., d){
      #the column x2is the key
      cle <- d$X2
      #return key and the entire data frame
      return(keyval(cle,d))
  }
```

Figure IV - 7: code map dans r.

- **algorithme reduce :**

Le figure suivante représente la implémentation de fonction reduce et où appel de fonction régression linéaire compte " reg " pour chaque Reduce,

```
> reduce_im = function(k,d){
      #print
      print("reduce");print(d)
      # calcul
      x <- reg(.,d$X1) ;y <- reg(.,d$X1)
      liste <- list(x, y);
      return(keyval(k,liste))
  }
```

Figure IV - 8: code reduce dans r.

4.3.programmation de mapreduce pour K-means

- **programmation de map K-means**

L'image ci-dessous représente la fonction de map algorithme mapreduce avoir les mêmes résultats map première (la même clé)

```
> map_km = function(., l){
      if (l[[1]][[1]] | l[[1]][[2]])
          cle <- 0
      else
          cle <- 1
      return(keyval(cle,d))

  }
```

Figure IV - 9: code map_km dans r

➢ **programmation de reduce pour K-means**

Le figure suivante représente l'implémentation de la fonction reduce et où appel de fonction k-means pour Reduce,

```
> reduce_km = function(cl,l){
    # ****print ****
    print("reduce"); print(l)
    km1 = kmeans(l, 3, nstart=100)
    plot(l, col =(km1$cluster +1) , main="K-Means result with 3 clusters", pch=20, cex=2)
    liste <- l
    return(keyval(cl,liste))
 }
```

Figure IV - 10: code reduce_km dans r

5. Exécution de programme

5.1. Exécution d'algorithme mapreduce 1

✓ pour charge la bibliothèque rmr2 utiliser le instruction représenter dans Figure

```
> library(rmr2)
```

Figure IV - 11: l'instruction de chargement la bibliothèque

✓ l' instruction suivant pour utiliser RHadoop

```
> rmr.options(backend="local")
NULL
```

Figure IV - 12: instruction pour fonctionner sous Hadoop

✓ Convertir les données et le copier dans un fichier temporaire sur HDFS.

```
> d1.dfs <- to.dfs(d1)
```

Figure IV - 13: instruction de convertir la donnée.

✓ Dans cette étape est le lancement du processus Mapreduce géré la saisie des données, la carte et réduire les emplois.

```
> calcul <- mapreduce(input=d1.dfs,map=map_im,reduce=reduce_im)
```

Figure IV - 14: instruction de lancement de mapreduce.

✓ récupération de la sortie temporaire sur HDFS

```
> resultat <- from.dfs(calcul)
```

Figure IV - 15: Récupérer la sortie d'un fichier temporaire sur HDFS.

✓ **L'affichage de résultat :**

- L'affichage de résultat de mapreduce le figure suivant représenter le résultat de Key.

```
> resultat$key
[1] 0 0 1 1
```

Figure IV - 16: le résultat de Key

- le figure suivant représenter le résultat de valeur de chaque reduce.

```
> resultat$val
[[1]]
[[1]][[1]]
 (Intercept)
-0.004073019

[[1]][[2]]
                x
-0.02729041


[[2]]
[[2]][[1]]
(Intercept)
 0.03517225

[[2]][[2]]
                x
-0.009197347


[[3]]
[[3]][[1]]
(Intercept)
 0.02128459

[[3]][[2]]
                x
-0.04039503


[[4]]
[[4]][[1]]
(Intercept)
0.009352975

[[4]][[2]]
                x
-0.09834727
```

Figure IV - 17: l'affichage de résultat de valeur de chaque reduce.

5.2. Comparaison entre le résultat de l'algorithme et le système classique

Notez que les taux d'erreur entre l'approche de l'algorithme classique et l'approche de algorithme mapreduce et la figure IV – 18 suivante.

```
> resultat$val
[[1]]
[[1]][[1]]
 (Intercept)
-0.004073019

[[1]][[2]]
                x
-0.02729041


[[2]]
[[2]][[1]]
(Intercept)
 0.03517225

[[2]][[2]]
                x
-0.009197347


[[3]]
[[3]][[1]]
(Intercept)
 0.02128459

[[3]][[2]]
                x
-0.04039503


[[4]]
[[4]][[1]]
(Intercept)
0.009352975

[[4]][[2]]
                x
-0.09834727
```

```
> summary(sum)

Call:
lm(formula = d1$X2 ~ d1$X1)

Residuals:
     Min       1Q   Median       3Q      Max
-0.57910 -0.12567 -0.02919  0.04799  0.94830

Coefficients:
             Estimate Std. Error t value Pr(>|t|)
(Intercept) -1.412e-01  6.806e-03  -20.75   <2e-16 ***
d1$X1        3.216e-03  7.827e-05   41.09   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.2548 on 4998 degrees of freedom
Multiple R-squared:  0.2525,    Adjusted R-squared:  0.2523
F-statistic:  1688 on 1 and 4998 DF,  p-value: < 2.2e-16
```

Figure IV - 18: comparaison entre le résultat de régression linaire classique et régression avec mapreduce

5.3. Exécution d'algorithme mapreduce pour k-means :

✓ Pour la compilation des résultats de chacun reduce de l'algorithme mapreduce nous utilisons les data. Frame pour crée un tableau de données.

```
> l <- data.frame(cbind(S,P))
```

Figure IV - 19: instruction pour l'agrégation de résultat de mapreduce régression

✓ dans cette étape exécuter le algorithme mapreduce par le entrée de donnée, la fonction map et reduce.

```
> cal_km <- mapreduce(input=l.dfs,map=map_km,reduce=reduce_km)
```

Figure IV - 20: Lancement de la fonction mapreduce.

✓ récupération de la sortie temporaire sur HDFS.

```
> resultat1 <- from.dfs(cal_km)
```

Figure IV - 21: instruction pour une récupération de fichier temporaire est HDFS

✓ l'affichage de résultat de Key de mapreduce.

```
> resultat$key
[1] 0 0 1 1
```

Figure IV - 22: résultat de Key de mapreduce_km

✓ l'affichage de résultat de valeur de reduce

```
> resultat1$val
$cluster
1 2 3 4
2 1 2 3

$centers
           S             P
1 0.035172251 -0.009197347
2 0.008605785 -0.033842722
3 0.009352975 -0.098347272

$totss
[1] 0.005298239

$withinss
[1] 0.0000000000 0.0004073696 0.0000000000

$tot.withinss
[1] 0.0004073696

$betweenss
[1] 0.004890869

$size
[1] 1 2 1

$iter
[1] 2

$ifault
[1] 0
```

Figure IV - 23: résultat de Key de mapreduce_km

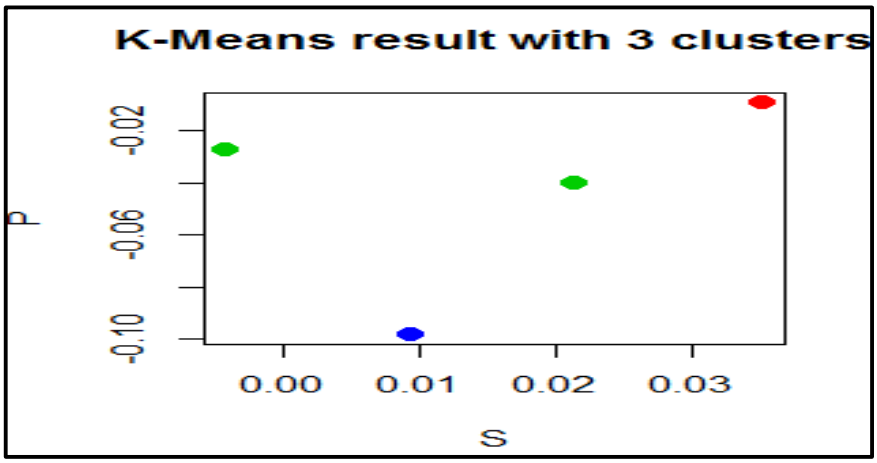


Figure IV - 24: Affichage de résultat de k-means

5.4. Comparaison entre le résultat de l'algorithme et le système classique

Nous notons la grande différence de valeurs entre le approche classique et l'utilisation d'un algorithme mapreduce dans le valeur d'erreur.

```
> print(k1$centers)
          x1          x2
1 152.25380 0.41323210
2  81.17891 0.05194124
3  33.96225 0.00000000
> print(k1$totss)
[1] 10593836
> print(k1$withinss)
[1] 529966.2 483855.8 354148.9
> print(k1$tot.withinss)
[1] 1367971
> print(k1$betweenss)
[1] 9225865
> print(k1$size)
[1]  922 1906 2172
> print(k1$iter)
[1] 2
> print(k1$ifault)
[1] 0
```

```
$totss
[1] 0.005298239

$withinss
[1] 0.0000000000 0.0000000000 0.0004073696

$tot.withinss
[1] 0.0004073696

$betweenss
[1] 0.004890869

$size
[1] 1 1 2

$iter
[1] 2

$ifault
[1] 0
```

Figure IV - 25:comparaison entre le résultat d'approche classique et algorithme Mapreduce.

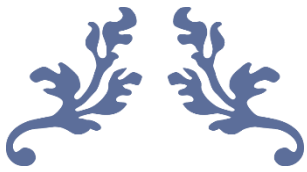
5.5. Interprétation des résultats

Dans ce travail, nous avons extrait les équations de lignes de données en utilisant un algorithme Mapreduce et nous sommes allés à la suite des résultats peut

- Moins que le taux d'erreur par rapport à l'approche classique .
- Moins que le taux d'erreur plus nous dans le nombre de nœud dans l'architecture .

6. Conclusion

Dans ce chapitre, nous avons présenté l'implémentation de notre système, pour cela, nous avons décrit les outils de réalisation system de régression pour le Big data, en commençant par la présentation de l'environnement du travail, ainsi les outils utilisés, enfin nous avons montré quelques interfaces qui représentent les résultats.



Conclusion Général

Conclusion général

Dans notre mémoire nous avons étudiés, le problème de la régression linaire dans les big data architecture. Cette étude constitué par quatre chapitre, dans les chapitres 1 et 2, nous présentons les concepts de base de fouille de donnés, régression linaire les architecture big data et le paradigme des algorithmes Map/Reduce.

Dans la troisième chapitre nous abordons notre proposition par un modèle mathématique formel exprime les relations entre les données, sous données, nœud et les équations linaire de la régression. Ensuite on à présenter notre algorithmes Map/Reduce qui répond au besoin d'un problème de régression linaire. k-means algorithme joue un rôle important, il sélectionne les équations de régression le plus adéquat .En fin nous terminons par flux de données de notre architecture et une description UML de système.

Dans le quatrième chapitre nous testons et validons notre proposition par l'utilisation de Universel Bank data set .les équations linaire trouvés , permettons de diminuer le taux d'erreur par rapport à un système classique.

Bibliographie

[1] Gilbert Saporta; RST « Epidémiologie »; Chapitre 4.2 .« DATA MINING » 5/12/04

[2] Abdelhamid Djeffal. « Cours sur la fouille de données avancées » Université Mohamed Khider, Biskra, 2013.

[3]Morgan Kaufmann. « Data Mining: Concepts and Techniques » Academic press, USA, 2001.

[4] KonstantinosTsiptsis, AntoniosChorianopoulos. « Data Mining Techniques in CRM: Inside Customer Segmentation» Wiley publishing, UK, 2009.

[5] Josiane Confais et Monique Le Guen ;PREMIERS PAS en REGRESSION LINEAIRE avec SAS

[6] http://neumann.hec.ca/sites/cours/30-636-01/seance7-8/regression.ppt

[7] https://medium.com/big-data-et-sante/petite-introduction-au-big-data-95713ba89034

[8]http://webcache.googleusercontent.com/search?q=cache:2DfXDe1Z_mEJ:www.aubay.com/fileadmin/user_upload/Publications_FR/Regard_Aubay__Big_Data_Web.pdf+&cd=1&hl=ar&ct=clnk&gl=dz

[9] http://www.redsen-consulting.com/2013/06/big-data/

[10] M.CORINUS, T.Derey, J.Marguerie, W.Techer, N.Vic, Rapport d'étude sur le Big Data, SRS Day, 54p, 2012.

[11] Big data application and archetecteure; de himanshu et soumendramohanty.

[12] Mekideche Mounir, Conception et implémentation d'un moteur de recherche à base d'une architecture Hadoop (Big Data), Avril 2015.

[13] Bernard ESPINASSE, Introduction aux systèmes NoSQL (Not Only SQL), Ecole Polytechnique Universitaire de Marseille, 19p, Avril 2013.

[14]Zhang, Q., L. Cheng, et R. Boutaba. Cloud computing: state-of-the-art and research challenges. Journal of Internet Services and Applications 1, (2010) 7–18.

[15]Chen, R., H. Chen, et B. Zang. MapReduce: Optimizing resource usages of data-parallel applications on multicore with tiling. Proceedings of the 19th PACT, NY, USA (2010) 523–534

[16]Ibrahim, S., H. Jin, et B. Cheng. Hadoop: The definitive guide. In HPDC, NY, USA, (2009) 65–66.

[17] Dou, A., V. Kalogeraki, et D. Gunopulos. Misco: a mapreduce framework for mobile systems. Proceedings of the 3rd

[18] Miceli, C., M. Miceli, et S. Jha. Programming abstractions for data intensive computing on clouds and grids. Proceedings of the CCGRID09, IEEE Computer Society, Washington, DC, USA, (2009) 478–483.

[19]http://www.journaldunet.com/developpeur/outils/les-solutions-du-big-data/principe-de-fonctionnement-de-mapreduce

[20] Dean, J., Ghemawat, S. (2010). MapReduce: a flexible data processing tool. Communications of the ACM, 53(1), 72-77.

[21] V.Martha, W.Zhao, XiaoweiXu, h-MapReduce: A Framework for Workload Balancing in MapReduce, IEEE 27th International Conference on Advanced Information Networking and Applications, pp.637-644, 25-28 March 2013.

[22] Shafer, J., Rixner, S., Cox, A. L. (2010, March). The hadoop distributed filesystem: Balancing portability and performance. In Performance Analysis of Systems Software (ISPASS), 2010 IEEE International Symposium on (pp. 122-133). IEEE

[23] Naoui, M. A., Mcheick, H., Kazar, O. (2014, May). Mobile Agent approach based on mobile strategic environmental Scanning using Android and JADELEAP systems. In Electrical and Computer Engineering (CCECE), 2014 IEEE 27th Canadian Conference on (pp. 1-7). IEEE.c

[24] Jun, S., Lee, S. J., Ryu, J. B. (2015). A Divided Regression Analysis for Big Data. Statistics, 9(5).

[25] Oancea., B.(2015).Linear Regression With R And HADOOP. International Conference : CKS - Challenges of the Knowledge Soc;2015, p1007.

[26]Ma, P., Sun, X. (2015).Leveraging for big data regression. Wiley Interdisciplinary Reviews: Computational Statistics, 7(1), 70-76.

[27]Neyshabouri, M. M., Demir, O., Delibalta, I., Kozat, S. S. (2016). Highly efficient nonlinear regression for big data with lexicographical splitting. Signal, Image and Video Processing, 1-8.

[28]Willems, F. M., Shtarkov, Y. M., Tjalkens, T. J. (1996). Context weighting for general finite-context sources. IEEE transactions on information theory, 42(5), 1514-1520

[29]Montgomery, D. C., Peck, E. A., Vining, G. G. (2015). Introduction to linear regression analysis.John Wiley Sons.

[30]Wang, Y., Li, Y., Xiong, M., Jin, L. (2015). Random Bits Regression: a Strong General Predictor for Big Data. arXiv preprint arXiv:1501.02990.

[31] Han, J., Pei, J., Kamber, M. (2011). Data mining: concepts and techniques. Elsevier.

[32] Bali Ahmed /Cours UML/ Chapitre II/ 2013

[33] Vincent Goulet ,Introduction à la programmation en R

[34] https://fr.wikipedia.org/wiki/RStudio

[35]https://acadgild.com/blog/integration-r-hadoop/

[36] https://github.com/RevolutionAnalytics/RHadoop/wiki/Downloads

[37] https://www3.nd.edu/ busiforc/problems/DataMining/UniversalBank.xls.