République Algérienne Démocratique et Populaire

Ministère de l'enseignement supérieur et de la recherche scientifique

Université de 8 Mai 1945 – Guelma -

Faculté des Mathématiques, d'Informatique et des Sciences de la matière

Département d'Informatique

Mémoire de Fin d'études Master

Filière : Informatique

Option : Science et Technologie de L'information et Communication

Intitulé du mémoire :

Impact des méthodes analytiques dans le contexte des données massives

Encadré Par :

Dr BENHAMZA KARIMA

Présenté par :

BOURAHDOUN MOHAMMED ILYAS

Octobre 2020

Remerciements

Au terme de ce travail, je tiens tout d'abord à remercier Dieu le tout puissant et miséricordieux qui m'a donné la force et la patience durant ces longues années d'étude.

Je tiens à exprimer ma profonde gratitude et mes sincères remerciements à mon encadreur Dr. **BENHAMZA Karima** pour son soutien, sa patience ses précieux conseils, son aide, sa disponibilité tout au long de mes études et sans qui ce mémoire n'aurait jamais vu le jour. Qu'elle trouve dans ce travail un hommage vivant à son grand dévouement et à sa haute personnalité.

Je tiens tout particulièrement à remercier les enseignants du département d'informatique pour leur disponibilité et encouragement, ainsi que tous les enseignants qui ont contribué à notre formation.

Ma reconnaissance va aussi aux membres de jury, pour l'honneur qu'ils auront fait en acceptant de juger ce travail.

Je remercie, enfin tous ceux qui, d'une manière ou d'une autre, ont contribué à la réussite de ce travail et qui n'ont pas pu être cités ici.

Résumé

Face à l'explosion volumineuse des données, le Big Data (données massives) offre une nouvelle alternative aux solutions traditionnelles de bases de données et d'analyse. Le but de ce travail est de montrer l'impact des méthodes analytiques appliquées aux données massives dans la classification des données et la prédiction. Tout l'intérêt est de faire ressortir les pépites d'informations cachées dans ces Méga-données. Les résultats de l'application des méthodes de partitionnement et de régression linéaire sur un dataset médical a permis de souligner l'importance de ces méthodes analytiques.

Mots clés : Big Data ; Méthodes Analytiques ; K-means ; Régression linéaire ; Dataset Médical.

Abstract

Faced massive data explosion, Big Data offers a new alternative to traditional database and analytics solutions. The aim of this work is to show the impact of analytical methods applied to big data in classification and prediction. The whole point is to bring out the nuggets of information hidden in this Big Data. The results of partitioning and linear regression methods applied to a medical dataset have showed the importance of these analytical methods.

Keywords: Big Data; Analytical Methods; K-means; Linear regression; Medical Dataset.

ملخص

في مواجهة الانفجار الهائل للبيانات ، تقدم البيانات الضخمة بديلاً جديدًا لقواعد البيانات التقليدية وحلول التحليلات. الهدف من هذا العمل هو إظهار تأثير الأساليب التحليلية المطبقة على البيانات الضخمة في تصنيف البيانات والتنبؤ بها. بيت القصيد هو إبراز شذرات المعلومات المخبأة في هذه البيانات الضخمة. سلطت نتائج تطبيق أساليب التقسيم والانحدار الخطي على مجموعة بيانات طبية الضوء على أهمية هذه الأساليب التحليلية.

الكلمات المفتاحية : البيانات الضخمة ؛ طرق تحليلية؛ K-means .الانحدارالخطي ؛ مجموعة البيانات الطبية.

Sommaire

Chapitre 1 : BIG DATA

Chapitre 2 : Analyse de données.

Chapitre 3 Conception et Implémentation

Liste des figures

Chapitre 1 Introduction au BIG DATA

Chapitre 2 Analyse de Données

Chapitre3 : Conception et implémentation.

Liste des tableaux

Chapitre1 : BIG DATA.

Chapitre2 : Analyse de données.

Chapitre3 : Conception et implémentation.

Introduction Générale

Depuis quelques années, nous sommes confrontés à une explosion de données structurées ou non (textes, photographies, vidéos, …) produites massivement par les différentes sources de données numériques (internet, réseaux de capteurs, traces de GPS, ….) et qu'il faudrait savoir traiter avec une croissance rapide. De rudes contraintes opposent les différents chercheurs dans le domaine, quant au stockage et à l'analyse de ces masses de données et qui dépassent les limites des technologies traditionnelles (bases de données relationnelles).

Cette révolution scientifique, qui a envahi le monde de l'information, a imposé aux différents chercheurs, de nouveaux défis et les a poussé à concevoir de nouvelles technologies pour contenir, analyser et traiter ces volumes énormes de données. C'est ainsi que naissait une nouvelle technologie : le « Big Data » ou les « Données massives ».

Plusieurs modèles d'analyse, de traitements parallèles et de systèmes de gestion de fichiers, venait enrichir les concepts qui sont liés à cette technologie. Ces analyses, appelées « Big Analytics », reposent généralement sur des méthodes de calcul distribué. Elles mettent en œuvre divers algorithmes relevant des statistiques de la fouille de données, de l'apprentissage machine automatique.

Notre projet de fin d'étude a pour but d'étudier les technologies et les méthodes du Big Data, Nous nous intéresserons particulièrement aux méthodes analytiques et leurs impacts dans l'analyse et le traitement des données massives. L'intérêt est de souligner leurs utilités face à la masse importante des données.

Ce mémoire sera départagé en trois parties :

Dans le premier chapitre, on présentera la technologie « Big Data » avec ses concepts et ses caractéristiques. Ensuite, dans le deuxième chapitre, les méthodes analytiques seront détaillées : On s'est intéressé particulièrement à la méthode de partitionnement (k-means) et la méthode de régression pour l'analyse des données massives.

La dernière partie (troisième chapitre) traitera la conception et l'implémentation du modèle proposé pour l'analyse des données médicales. La plateforme Spark est utilisée pour le traitement des données et la présentation des résultats obtenus.

Finalement, ce mémoire est clôturé par une conclusion générale, les perspectives de ce travail et les références bibliographiques utilisées.

Chapitre 1
BIG DATA

1.Introduction :

Avec l'évolution de la technologie, l'utilisation des données a augmenté ces dernières années, rapidement et dans tous les domaines comme le marketing, la météorologie, le trafic, la santé, etc. A cause de ces informations de grande taille appelées « Méga-données », les informaticiens se sont forcés de trouver de nouveaux concepts et de nouvelles méthodes de traitement. Ces derniers sont inclus dans un volet appelé "Big data" ou "Données massives". Ces recherches avaient pour objectifs de faciliter l'analyse des données de grandes tailles pour extraire les informations importantes.

Dans ce chapitre, nous allons présenter les notions liées à ce domaine "Big Data".

2.Historique :

Le terme « Big Data » est utilisé depuis le début des années 90. Bien que l'on ne sache pas exactement qui a utilisé le terme pour la première fois, la plupart des gens attribuent à John R. Mashey (qui travaillait à l'époque chez Silicon Graphics) d'avoir rendu le terme populaire.

Le Big Data n'est pas quelque chose de complètement nouveau. Au cours des années précédentes, les chercheurs ont essayé d'utiliser des techniques d'analyse et de traitement de données pour soutenir leur processus de prise de décision.

Néanmoins la quantité totale de données dans le monde, qui était de 4,4 zettaoctets en 2013, a grimpé à 44 zettaoctets en 2020. Ces 44 zettaoctets sont équivaut à 44 trillions de gigaoctets. Même avec les technologies les plus avancées aujourd'hui, il est impossible d'analyser toutes ces données. La nécessité de traiter ces ensembles de données, de plus en plus volumineux et non structurés, a contraint l'analyse de données traditionnelle à se transformer en « Big Data » au cours de la dernière décennie. [1]

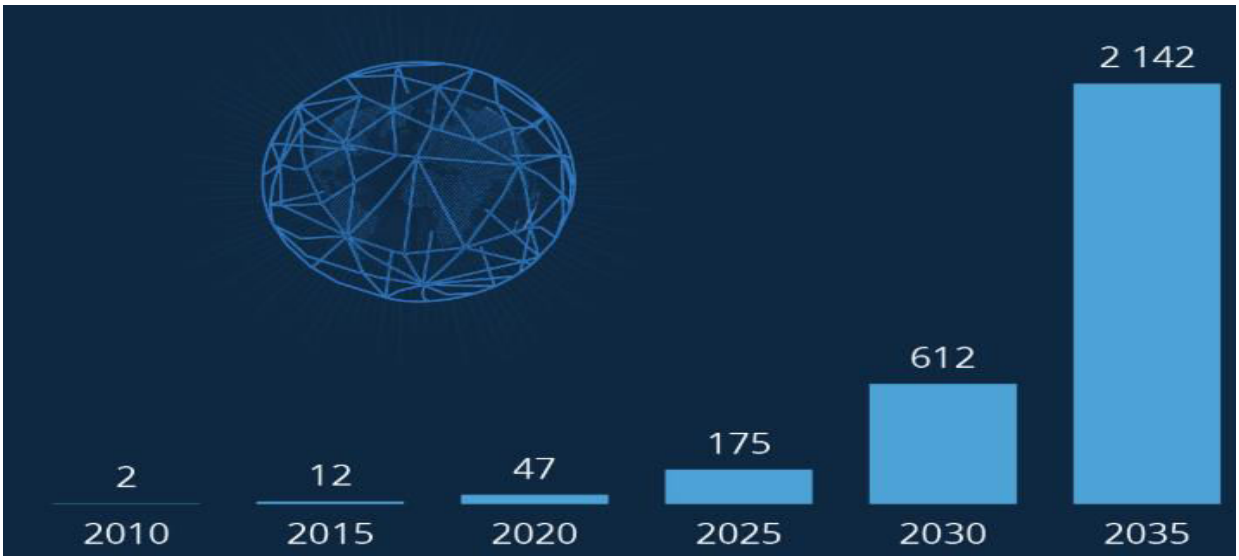

Figure 1 : Volume annuel des données numériques créées à l'échelle mondiale depuis 2010 en zettaoctets [2]

3. Définition du "Big Data" :

Littérairement, le Big Data désigne les « données massive » ou « mégadonnées ». C'est une expression anglophone utilisée pour désigner des ensembles de données volumineux, ne permettant pas l'utilisation d'outils classiques de gestion de base de données. Ceci se traduit par une difficulté de traitement, de stockage, d'analyse et de gestion de ces données avec les anciennes méthodes [3].

Plusieurs définitions ont été données au Concept "Big Data". Cependant, aucune n'a été universellement adoptée car ce concept est assez complexe et sa définition différée selon les usagers et les fournisseurs de service qui s'y intéresse.

Parmi ces définitions, nous citons :

- Les Big data sont des ressources d'informations volumineuses, à grande vélocité et à grande variété qui exigent des formes inventes et rentables de traitement de l'information pour améliorer la compréhension et la prise de décision [4]
- Le Big Data est un ensemble de technologies, d'architecture et de procédures permettant d'analyser et de traiter de larges quantités de données hétérogènes, et d'en extraire les informations pertinentes à un coût accessible [5].

Ainsi, on peut conclure que le Big Data fait référence à l'explosion du volume des données, à leur variété, et aux nouvelles solutions proposées pour gérer cette volumétrie tant par la capacité de stockage, d'analyse et d'exploitation de ces données en temps réel.

4. Type des données Big Data :

Une structure de données est une collection de valeurs de données, de leurs relations et des fonctions ou opérations qui peuvent être appliquées aux données. C'est un moyen qui permet d'organiser et de stocker des données dans un ordinateur, afin qu'elles puissent être consultées et modifiées efficacement.

Dans le cadre des Big Data, les données collectées, stockées et traitées peuvent être issues de différents domaines et crées par plusieurs sources de données hétérogènes, ce qui génère une masse de données de types différents structurés et non structurés[43], semi- structurées :

4.1 Données structurées :

Les données structurées font référence aux données avec un format et une longueur définie, faciles à stocker et à analyser, et hautement organisées. Cela signifie que les données sont organisées dans une structure reconnaissable afin de pouvoir répondre aux requêtes pour récupérer des informations à des fins d'organisation.

Une base de données relationnelle comme le langage de requête structuré (SQL) représente un bon exemple pour les données structurés, il contient des nombres organisés, des dates, des groupes de mots et des nombres appelés chaînes / texte.

En raison de la structure transparente de la base de données, elle peut être recherchée avec des algorithmes de recherche simples et directs qui peuvent être par type de données dans le contenu réel [6].

4.2. Données non-structurées :

Les données non structurées sont des informations, sous de nombreuses formes différentes, qui ne correspondent pas aux modèles de données conventionnels et qui ne conviennent donc généralement pas à une base de données relationnelle traditionnelle. Cela rend le traitement et l'analyse des données non structurées très difficiles et longues.[6]

D'après Feldman et Sanger, Les données non structurées n'ont pas de structure spécifique. Les données non structurées comprennent généralement des images / objets bitmap, du texte, des e-mails et d'autres types de données qui ne font pas partie de la base de données. [7]

4.3 Données semi-structurées :

Les données semi-structurées sont des données irrégulières qui peuvent être incomplètes et avoir une structure qui change rapidement ou de manière imprévisible mais qui ne se conforme pas à un schéma fixe ou explicite.

Hanig, Schierle et Trabold ont précisé que le modèle de données semi-structurées permet aux informations provenant de plusieurs sources, avec des propriétés liées mais différentes, d'être regroupées en un tout, par exemple, des fichiers de courrier électronique, XML et Doc [8].

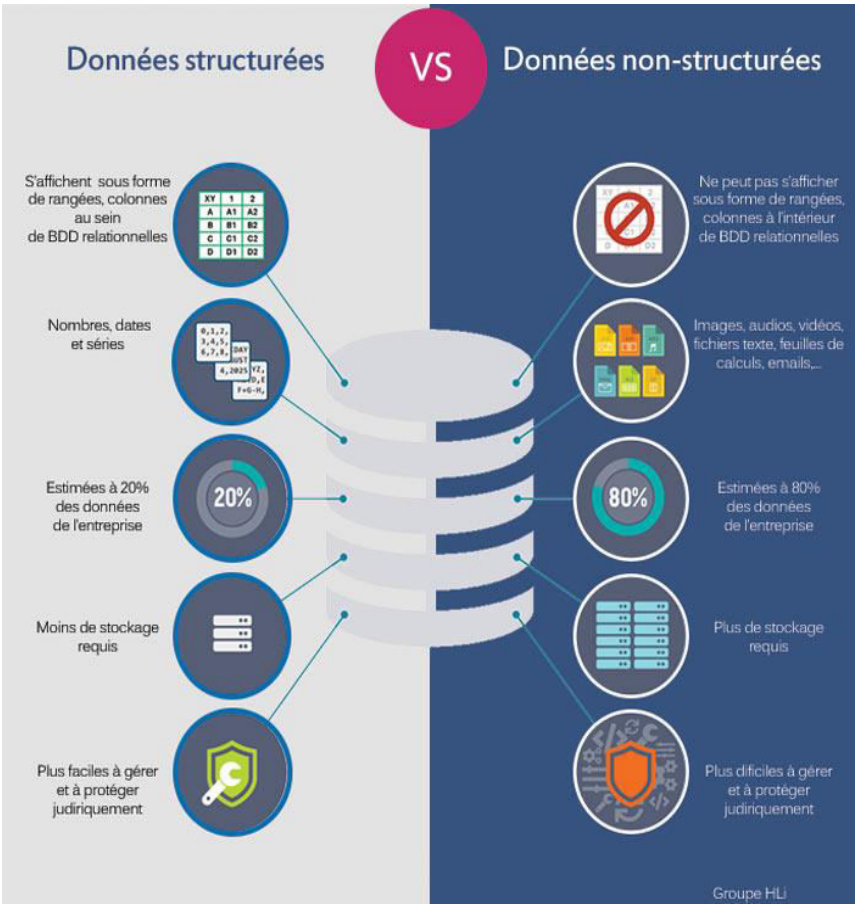

Figure 2 : Données structuré Versus non-structuré [9]

5. Caractéristiques de Big Data :

Les caractéristiques sous-jacentes des mégadonnées ou Big Data comprennent :

5.1 Volume : Le volume se réfère à la quantité de données générées quotidiennement par des entreprises ou des personnes.

5.2 Vitesse : La vitesse est une caractéristique importante du Big Data, elle signifie la vitesse de génération des données. Par exemple une grande vitesse de génération des résultats des données aboutit à une large quantité des données en peu de temps.

5.3 Variété : Les données à traiter sont sous forme structurés ou non structurés : data bases, textes, données de capteurs, sons, vidéos, de parcours, fichiers journaux etc.[44]

5.4 Véracité : La véracité fait référence à la précision et la fiabilité des données.

Pour extraire la valeur des données, les données doivent être nettoyées pour supprimer le bruit, ce nettoyage des données est important pour que les données incorrectes et défectueuses puissent être filtrées [10].

5.5 Valeur : La valeur des données se réfère à l'utilité des données en fonction du but prévu. L'objectif final de tout le système d'analyse des mégadonnées consiste à extraire cette valeur des données.

La valeur des données est également liée à la véracité ou à l'exactitude des données. Pour certaines applications, la valeur dépend également de la vitesse à laquelle nous pouvons traiter les données

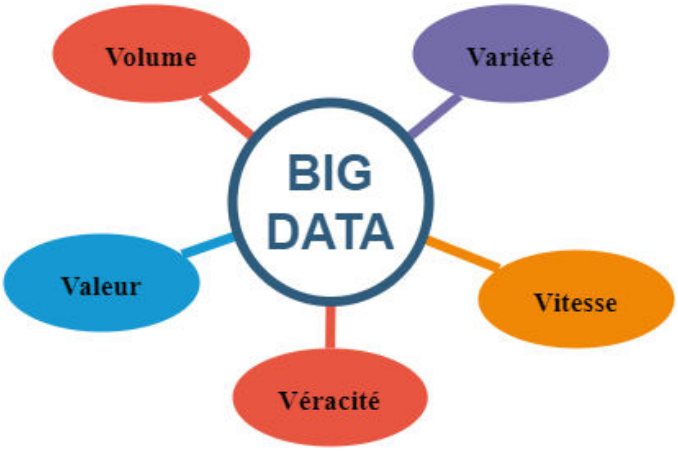

Figure 3 : 5V du Big data [11]

6. Technologies et plateformes pour Big Data :

Pour pouvoir traiter des bases de données volumineuses, plusieurs solutions ont été proposées :

***Des bases de données NoSQL** (comme MongoDB, Cassandra) qui implémentent des systèmes de stockage plus performants pour l'analyse de données non structurées en masse.

* **Des infrastructures de serveurs pour distribuer les traitements** sur des nœuds (traitement massivement parallèle).

*** Le stockage des données en mémoire** : On parle de traitement in-memory pour évoquer les traitements qui sont effectués en mémoire. L'avantage du traitement in-memory est celui de la vitesse puisque les données sont immédiatement accessibles.[45]

Plusieurs plateformes existent pour l'analyse et le traitement des données massives. On peut citer les plus utilisées :

6.1 Hadoop :

Le nom Hadoop a évolué pour signifier beaucoup de choses différentes, en 2002, il a été créé en tant que projet logiciel unique pour soutenir un moteur de recherche Web. Depuis ce temps, il est devenu un écosystème d'outils et d'applications qui sont utilisés pour analyser de grandes quantités et types de données [12].

Hadoop ne peut plus être considéré comme un projet unique monolithique, mais plutôt comme une approche du traitement des données qui diffère radicalement du modèle de base de données relationnelle traditionnel [13]. Il s'agit d'un logiciel "open-source" qui fonctionne dans le réseau d'ordinateurs en parallèle pour trouver des solutions au Big Data et le traiter à l'aide de l'algorithme MapReduce.

6.1.1 MapReduce :

MapReduce est un modèle de programmation qui permet d'effectuer un traitement parallèle et distribué sur d'énormes ensembles de données. MapReduce se compose de deux tâches distinctes : Map et Reduce.

Le premier est le « Map job », où un bloc de données est lu et traité pour produire des paires clé-valeur en tant que sorties intermédiaires. La sortie d'un Mapper ou d'un « Map job » (paires clé-valeur) est une entrée dans le réducteur « Reducer », ce dernier reçoit la paire clé-valeur de plusieurs « Map jobs ». Finalement, le réducteur « Reducer » agrège des tuples de données intermédiaires (paires clé-valeur intermédiaires) en un ensemble plus petit de tuples ou de paires clé-valeur, qui est la sortie finale.[14]. La méthode « MapReduce » peut être programmé sur plusieurs langages come python, java, scala, R etc.

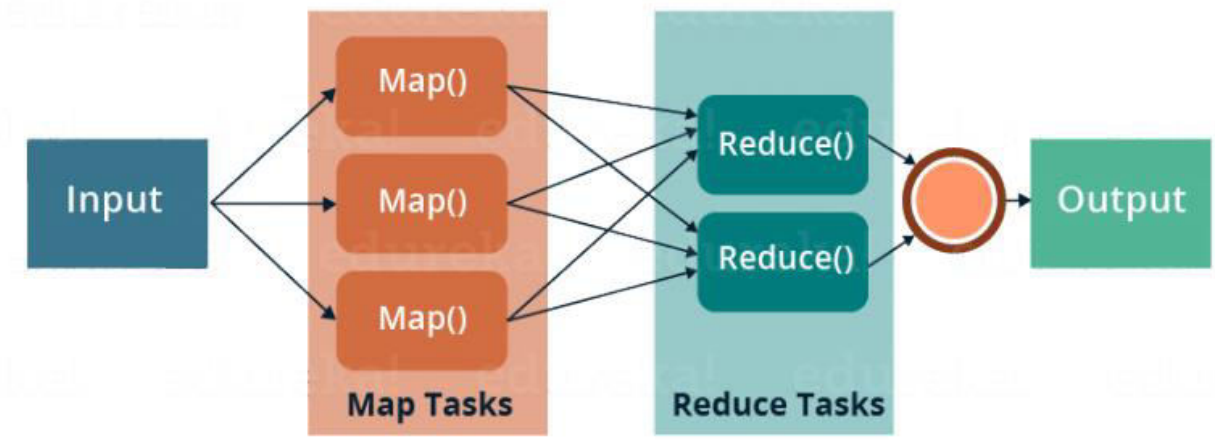

Figure 4 : Hadoop MapReduce.[14]

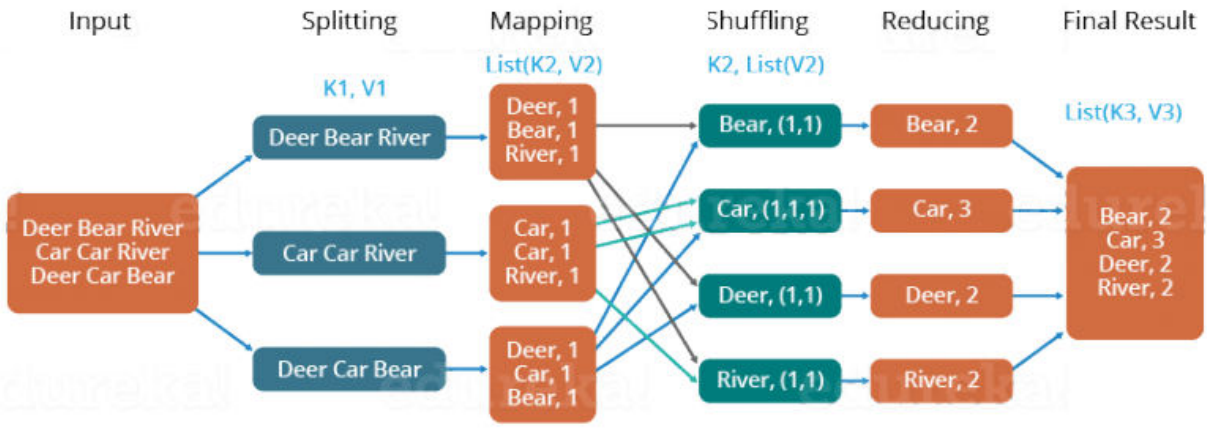

Figure 5 : Exemple d'utilisation de MapReduce.[14]

6.2 Apache Spark :

Apache Spark est un Framework de traitement distribué open source pour les charges de travail Big Data. Il utilise la mémoire cache et la fonction d'exécution de requête optimisée pour interroger rapidement des données de toute taille. Spark est donc un moteur rapide et polyvalent pour le traitement de données à grande échelle. Moteur Spark Core utilise l'ensemble de Données Distribué Résilient, ou RDD (Resilient Distributed Data), comme type de données de base [15].

Apache Spark peut effectuer rapidement des tâches de traitement sur de très grands ensembles de données, et peut également distribuer des tâches de traitement de données sur plusieurs ordinateurs, seul ou avec d'autres outils informatiques distribués. Ces deux qualités sont essentielles aux Concepts « Big Data » et l'apprentissage automatique, qui nécessitent la mobilisation d'une puissance de calcul massive pour parcourir les grands magasins de données.

Spark peut être implémenté avec différents langages de programmation fonctionnels Scala (qui s'exécute dans Java VM), java, python et R. [16]

En 2009, Apache Spark est devenu l'un des principaux frameworks de traitement distribué Big Data dans le monde.[46] Spark peut être déployé de différentes manières. On peut le trouver utilisé par les banques, les entreprises de télécommunications, les sociétés de jeux, les gouvernements et tous les grands géants de la technologie tels qu'Apple, Facebook, IBM et Microsoft.[46]

6.2.1 RDD (Resilient Distributed Data) :

RDD est une collection de données en lecture seule, qui peuvent être partitionnées dans un sous-ensemble de machines de cluster Spark et constituent les principaux composants de travail. RDD est un élément indispensable de la fonctionnalité de Spark, de sorte que toute l'API Spark est considérée comme un ensemble d'opérations de création, de transformation et d'export de RDD.[17]

Les données d'entrée qui forment un RDD sont partitionnées en morceaux et distribuées sur tous les nœuds du cluster Spark, chaque nœud effectuant ensuite un calcul en parallèle Une série d'opérations parallèles peut être effectuée sur les RDD à l'aide de l'API Spark Core. Ces opérations sont divisées en deux catégories différentes: les transitions et les actions.

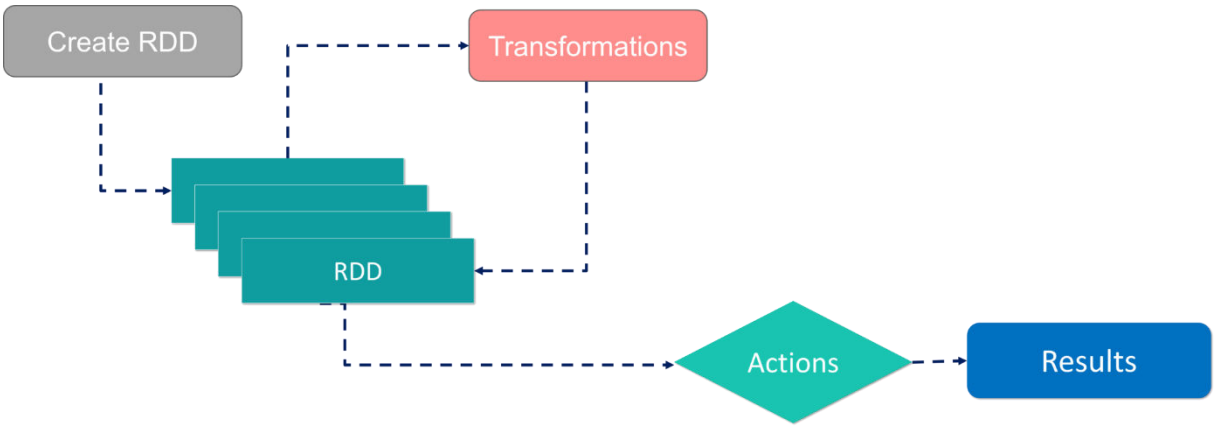

Figure 6 : Spark RDD.[17]

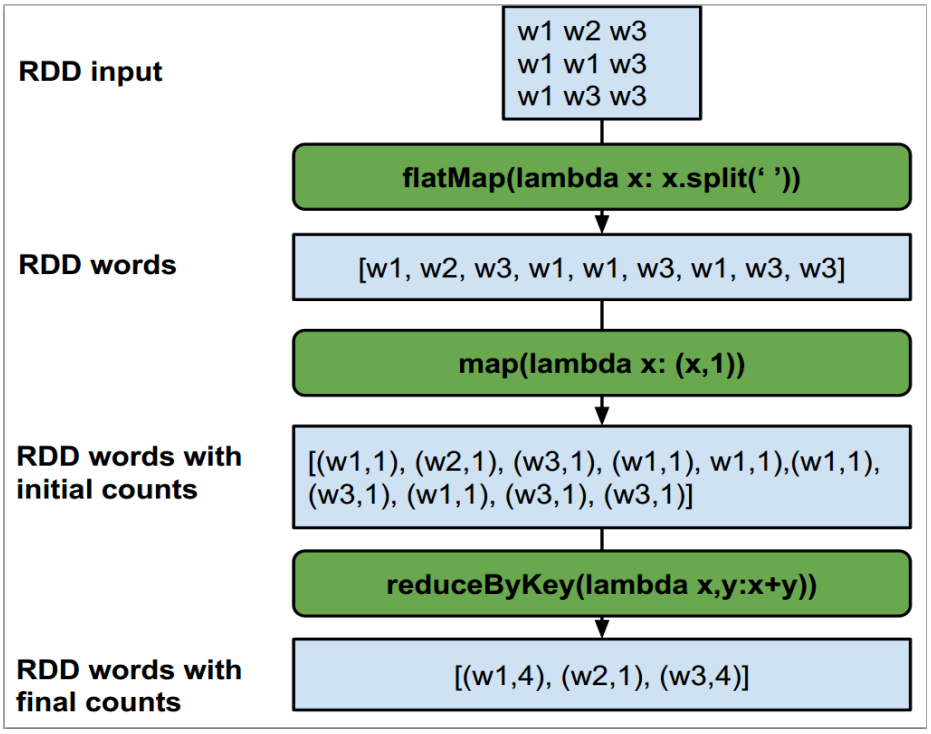

Figure 7 : Exemple d'utilisation de RDD [17]

Dans cet exemple :

La fonction " flatMap " : Prend le fichier d'entrée qui est retourné par la fonction sc.textFile qui renvoie les lignes du fichier. Elle applique la fonction lambda à chaque ligne et créant une liste de mots séparés par des espaces.

L'opération " map " : Applique la fonction lambda fournie à chaque élément du RDD,elle donne 1 pour chaque éléments (valeur).

La fonction " reductionByKey " : périmètre de faire la somme, donc a chaque fois elle trouve les mêmes clés (exemple w1=w1) elle fait la somme des valeur (1+1=2) après elle prend le résultat et continuer l'opération pour chaque élément.

6.2.2 Les composants de spark :

Spark se compose de plusieurs composants étroitement intégrés, parmi lesquels Spark Core est le composant principal, permettant aux utilisateurs d'utiliser d'autres composants comme bibliothèques, et chaque composant dépend des composants Spark Core de niveau inférieur.

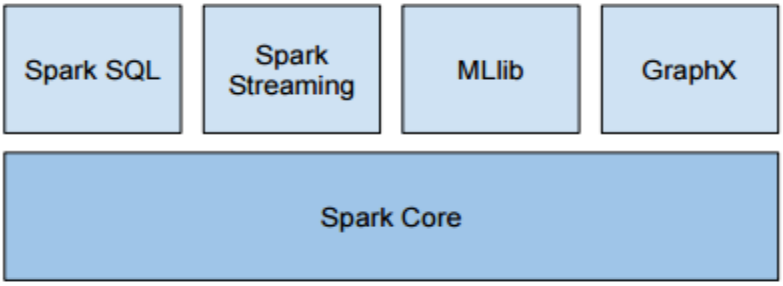

Figure 8 : Les composants de spark.[47]

- **Spark core :**
  Les fonctions de base de Spark incluent la planification des tâches, la gestion de la mémoire, la récupération après une panne, l'interaction avec les systèmes de stockage, etc. Démontrez le concept de base de l'ensemble de données distribuées élastiques de Spark, qui peut traiter les données via l'API RDD de Spark.
- **Spark SQL :**
  Spark SQL est un module de traitement de données construit via l'interface SQL. Les données sont accessibles à partir de diverses sources de données (telles que JSON, Parquet, tables Hive et JDBC). Spark SQL fournit les fonctions suivantes: manipuler des données via SQL, HQL2 ou une API d'ensemble de données de haut niveau personnalisée, et mélanger SQL avec la manipulation de données fournie par l'API (Application Programming Interface)Spark RDD.[18]

  Spark streaming :

  Spark streaming est utilisé pour le traitement des données en temps réel. Bien que Spark soit conçu comme une infrastructure de traitement par lots, il peut être utilisé pour traiter des données en continu à l'aide du concept de micro-lots fourni par Spark Streaming. Toutes ces bibliothèques fonctionnent sur les RDD en tant qu'abstraction de données, de sorte qu'elles peuvent toutes être combinées ensemble dans une seule application de manière transparente pour offrir des avantages significatifs aux utilisateurs.[19]
- **MLlib :**
  MLlib est une bibliothèque d'apprentissage automatique évolutive, incluse dans Apache Spark, qui contient des fonctionnalités d'apprentissage automatique. MLlib contient des algorithmes d'apprentissage automatique courants (classification, régression, clustering et filtrage collaboratif) et offre des fonctionnalités telles que l'évaluation de modèle et l'importation de données.[17]
- **GraphX :**
  GraphX est un composant pour la manipulation de graphes et les calculs parallèles aux graphes.[20]

"Resilient Distributed Dataset" (RDD) est une abstraction tolérante aux pannes pour le calcul en cluster en mémoire. D'un autre côté, RDD est également défini comme une collection d'objets immuable, qui est partitionnée et distribuée entre les nœuds de calcul dans le cluster.

Il peut être comparé à la mémoire partagée distribuée (DSM), qui a également fait l'objet d'études approfondies. Il y a deux différences principales.[21]

- DSM gagne en tolérance aux pannes en créant des points de contrôle et annule en cas de panne. Cela entraîne une surcharge importante car la reconstruction peut interférer avec plusieurs ensembles de données autres que l'ensemble de données défaillant. RDD est plus efficace. Si un échec se produit lors de la création d'un RDD, il peut être reconstruit à l'aide des données d'ancêtre. Spark conserve le diagramme de lignage de chaque RDD, de sorte que le RDD perdu peut être recalculé à tout moment.
- DSM extrait les données dans l'espace de noms global et met à jour les données à granularité fine. RDD agit comme MapReduce et pousse le calcul des données vers le nœud local et créé par des transformations grossières en grai persistantes sur les opérations.

L'utilisation efficace de la mémoire est la clé d'une performance optimale. Les dernières recherches montrent que seule une petite partie des données est extraite de grands ensembles de données en mémoire. Cela prend en charge l'utilisation du stockage en colonnes, car l'analyse est généralement effectuée par un petit nombre d'attributs par rapport à tous les attributs de l'ensemble de données.[22]

Il existe deux types d'opérations sur RDD: les transformations et les actions. Les transformations renvoient des pointeurs vers de nouveaux RDD tandis que les actions renvoient des valeurs ou des résultats au programme pilote, plusieurs transformations et actions peuvent être enchaînées pour effectuer des analyses complexes sur des ensembles de données. [23]

6.2.3 Les transformations:

Les transformations créent un nouveau RDD en exécutant une opération sur un RDD existant (la figure suivante), par exemple « filter (func) » est une transformation qui utilise une fonction de prédicat pour faire correspondre les éléments qu'elle renverra et « map (func) » est une transformation qui applique une fonction à chaque élément de l'ensemble de données, formant un nouvel ensemble de données à partir des valeurs renvoyées par la fonction.

Les transformations Spark sont évaluées paresseusement (calculées à la demande), ce qui signifie qu'elles ne sont calculées que lorsqu'elles sont nécessaires pour calculer un résultat par une action, jusque-là, seul l'ordre des transformations pour l'ensemble de données est stocké.[22]

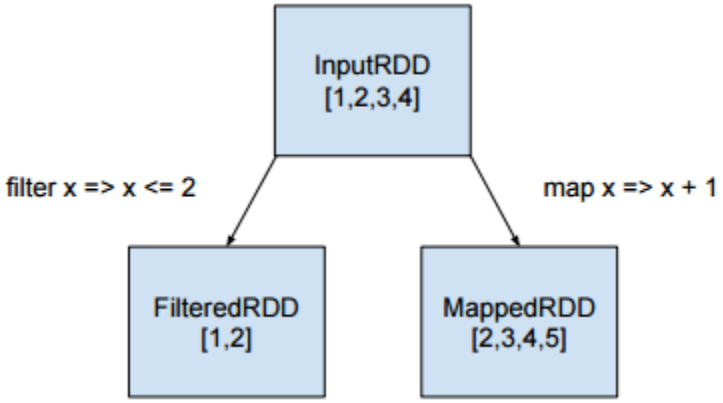

Figure 9 : Exemple pour les fonction filter/map [22]

6.2.4 Les action :

Les actions sont des opérations qui renvoient le résultat au programme pilote après avoir effectué le calcul sur le RDD d'entrée.

Par exemple, "réduire (func)" est une action qui réduit tous les éléments de l'ensemble de données d'entrée en utilisant une fonction donnée, en opérant sur deux éléments et en renvoyant un élément (chacun du type d'ensemble de données), réduisant finalement l'ensemble de données à une valeur unique, "count ()" est une action qui renvoie le nombre d'éléments dans l'ensemble de données et "first ()" est une action qui renvoie simplement le premier élément peut écrire des données sur un stockage externe.[22]

Opération	Utilisation de la fonction	Entrée => sortie
Les transformations	map(T=>U)	RDD[T]=>RDD[U]
	flatMap(T=>Sequence(U))	RDD[T]=>RDD[U]
	filter(T=>Boolean)	RDD[T]=>RDD[T]
	groupByKey()	RDD[(K,V)] => RDD[(K, Sequence[V])]
	reduceByKey((V,V)=>V)	RDD[(K, V)]) => RDD[(K, V)]
	join ()	(RDD[(K, V],RDD[(K, W)])) => RDD[(K, (V, W))]
Les action	Count(), sum()	RDD[T] => numeric
	collect()	RDD[T] => Sequence[T]
	reduce()	RDD[T] => T

Tableau 1 : Utilisation du RDD du framework Spark.[24]

6.2.5 Ensembles de données de paires clé-valeur :

Il existe certaines opérations et actions qui ne sont disponibles que sur les RDD avec des données de paires clé-valeur (par exemple reduceByKey ()). Pour que ces opérations fonctionnent, la structure des RDD doit être stockée dans des tuples.[22]

5.2.6 Récupération RDD via les données de lignage :

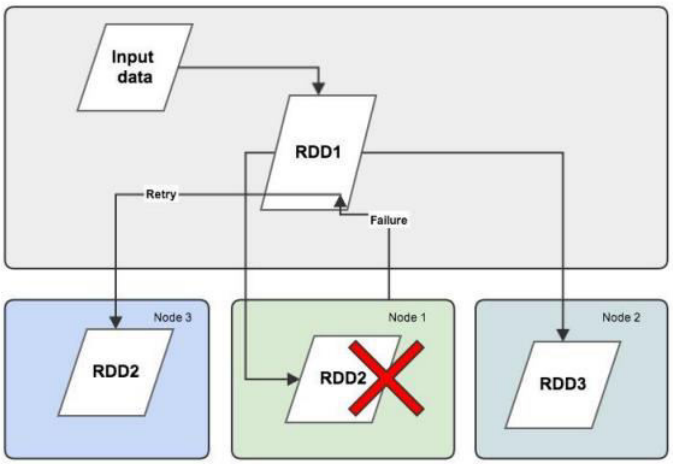

Figure 10 : Schéma explique la récupération via les données lignage.

L'entrée est la base de RDD1. Cela pourrait représenter une fraction des données filtrées à partir du fichier d'entrée.RDD2 et RDD3 pourraient représenter un filtrage ou une agrégation de niveau supérieur. Si RDD2 ne rentre pas dans la mémoire ou si le nœud échoue, alors RDD2 peut être facilement reconstruit sur les données de lignage, car le nœud maître sait que RDD2 a été construit sur la base de RDD1.

La reconstruction est effectuée dans un autre nœud disponible au hasard.

6.3 Comparaison entre Hadoop MapReduce et Spark RDD :

Facteurs	SPARK	HADOOP
Vitesse	Apache Spark exécute des applications jusqu'à 100 fois plus vite en mémoire et 10 fois plus vite sur disque que Hadoop.	MapReduce lit et écrit à partir du disque, en conséquence, il ralentit la vitesse de traitement.
Difficulté	Spark est facile à programmer car il dispose d'un grand nombre d'opérateurs de premier plan avec RDD	Dans MapReduce, les développeurs doivent coder manuellement chaque opération, ce qui rend le travail très difficile.

Latence	Spark fournit un calcul à faible latence	MapReduce est un framework de calcul à latence élevée.
Mode interactif	Spark peut traiter les données de manière interactive.	MapReduce n'a pas de mode interactif.
Diffusion	Spark peut traiter des données en temps réel via Spark Streaming.	Avec MapReduce, vous ne pouvez traiter les données qu'en mode batch.
Facilité d'utilisation	Spark est plus facile à utiliser	MapReduce est complexe à utiliser

Tableau 2 : Hadoop MapReduce Versus Spark RDD

7. Domaine d'application du Big data :

7.1 La santé :

Le Big Data a déjà commencé à créer une énorme différence dans le secteur de la santé. Grâce à l'analyse prédictive, les professionnels et les professionnels de la santé sont désormais en mesure de fournir des services de soins de santé personnalisés aux patients individuels.

En dehors de cela, les appareils portables de fitness, la télémédecine, la surveillance à distance sont tous alimentés par les données massives ou Big Data et contribuent à changer les vies pour le mieux [25].

7.2 Le secteur bancaire :

Le secteur bancaire s'appuie sur le Big Data pour la détection des fraudes. Les outils Big Data peuvent détecter efficacement les actes frauduleux en temps réel tels que l'utilisation abusive des cartes de crédit / débit, l'archivage des pistes d'inspection ou la modification défectueuse des statistiques client [25].

7.3 L'internet des objets :

C'est l'un des plus grands utilisateurs de Big Data, les entreprises informatiques du monde entier utilisent le Big Data pour optimiser leur fonctionnement, améliorer la productivité des employés et minimiser les risques dans les opérations commerciales. En combinant les technologies Big Data avec le domaine du Machine Learning (ML) et l'intelligence Artificielle (IA ou AI), le secteur informatique propulse continuellement l'innovation pour trouver des solutions même pour les problèmes les plus complexes [25].

8. Conclusion :

Toutes ces informations convergent vers un point commun : Dans un futur proche le Big Data serait très utilisé pour l'analyse et le traitement des données dans plusieurs domaines : la création de nouvelles entreprises, l'amélioration de la satisfaction clients, la détection d'épidémie, la détection de foyer de tension …etc.

Le Big data et l'analyse de données massives sont des concepts très liés et importants pour le développement de tous les domaines. Dans le chapitre 2, on présentera l'essentiel des méthodes analytiques dans le cadre des Big Data.

Chapitre 2
Analyse de Données

1. Introduction :

L'analyse des données est un sous-domaine des statistiques. Ce dernier englobe des méthodes qui fournissent des liens existants entre différentes données. Dans ce chapitre, on va exposer la définition et l'objectif de l'analyse des données et les méthodes les plus utilisées. La relation "Big data" et "Analyse de données » va être soulignée à travers quelques applications importantes.

2. Analyse de données :

2.1 Définition :

L'analyse de données est la science qui consiste à analyser des données brutes afin de tirer des conclusions sur ces informations, cette science englobe les processus, les technologies, les cadres et les algorithmes pour extraire des informations significatives à partir des données [26].

L'analyse est ce processus d'extraction et de création d'informations à partir de données brutes en filtrant, traitant, catégorisant, condensant et contextualisant les données. Ces informations obtenues sont ensuite organisées et structurées pour inférer des connaissances sur le système pour avoir des informations prêtes pour la consommation humaine, et cela veut dire une connaissance sur le système et / ou ses utilisateurs, son environnement et ses opérations et progresser vers ses objectifs.

Le choix des technologies, des algorithmes et des cadres d'analyse est guidé par les objectifs d'analyse de l'application. De toutes façons les données brutes en elles-mêmes n'ont pas de sens tant qu'elles ne sont pas contextualisées et transformées en informations utiles [26-27].

2.2 Type d'analyse de données :

Il y a quatre types d'analyse :

2.2.1 Analyse descriptive :

L'analyse descriptive comprend l'analyse des données passées pour les présenter sous une forme résumée qui peut être facilement interprétée et faire une amélioration des connaissances, de la compréhension et de l'application du lecteur liées à la recherche.

L'utilisation de fonctions statistiques telles que le nombre, le maximum, le minimum, la moyenne, les N premiers, le pourcentage, par exemple représente une partie importante des analyses effectuées aujourd'hui. On peut prendre comme exemple le calcul de nombre moyen de visiteurs par mois sur un site web.

Finalement pour simplifier, l'analyse des données descriptive vis à répondre à la question Que s'est-il passé ? [26-28].

2.2.2 Analyse diagnostique :

Ce type d'analyse se concentre davantage sur les raisons pour lesquelles quelque chose s'est passé, cela implique des entrées de données plus diversifiées et un peu d'hypothèses.
Bien que l'analyse descriptive puisse être utile pour résumer les données en calculant diverses statistiques (telles que la moyenne, le minimum, le maximum, la variance ou le N supérieur).
On peut prendre un exemple comme un système qui collecte et analyse les données des capteurs des machines pour surveiller leur état de santé et prévoir les pannes, ici le rôle de l'analyse diagnostique fournir plus d'informations sur les raisons pour lesquelles une erreur s'est produite en fonction des modèles du données du capteur pour les défauts précédents.

Donc l'analyse diagnostique vise de répondre à la question Pourquoi est-ce arrivé ?[29].

2.2.3 Analyse prédictive :

L'analyse prédictive est la branche de l'analyse avancée qui est utilisée à faire des prédictions sur des événements futurs inconnus. L'analyse prédictive consiste donc à prédire l'occurrence d'un événement ou le résultat probable d'un événement ou encore à prévoir les valeurs futures à l'aide de modèles de prédiction. On utilise l'analyse prédictive par exemple pour prédire quand un défaut se produira dans une machine, ou bien si une tumeur est bénigne ou maligne, et aussi prévoir les niveaux de pollution.
Donc on voit que l'analyse prédictive vise à répondre à la question Que va-t-il se passer ?[30]

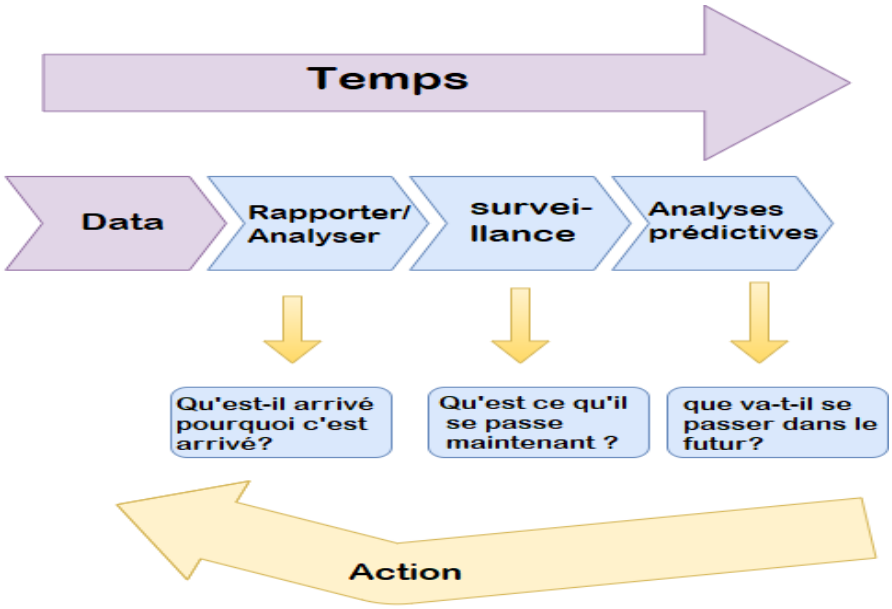

Figure 11 : Chaîne de valeur de l'analyse prédictive [31]

2.2.4 Analyse perspective :

Ce type d'analyse utilise différents modèles prédictifs pour différentes entrées. Tant que l'analyse prédictive utilise des modèles de prédiction pour prédire le résultat probable d'un événement, alors l'analyse prescriptive utilise plusieurs modèles de prédiction pour prédire divers résultats et le meilleur plan d'action pour chaque résultat.

L'analyse prescriptive peut prédire les résultats possibles en fonction du choix actuel des actions, elle prescrit des actions ou la meilleure option à suivre parmi les options disponibles. Exemple : l'analyse prescriptive peut être utilisée pour prescrire le meilleur médicament pour le traitement d'un patient en fonction des résultats de divers médicaments pour des patients similaires [26].

Donc cette analyse vise à répondre à la question Que pouvons-nous faire pour y arriver ?

Une caractérisation des tâches de calcul pour l'analyse massive de données (appelées les sept « géants ») a été effectuée par le Conseil national de la recherche [32]. Ces tâches de calcul comprennent :

1- Statistiques de base.

2-Problèmes généralisés à N-body.

3- Calculs algébriques linéaires.

4-Calculs théoriques des graphes.

5- Optimisation.

6-Intégration.

7- Problèmes d'alignement.

Cette caractérisation des tâches de calcul vise à fournir une classification des tâches qui se sont avérées utiles pour l'analyse des données, et à les regrouper grossièrement selon des structures mathématiques et des stratégies de calcul. [32]

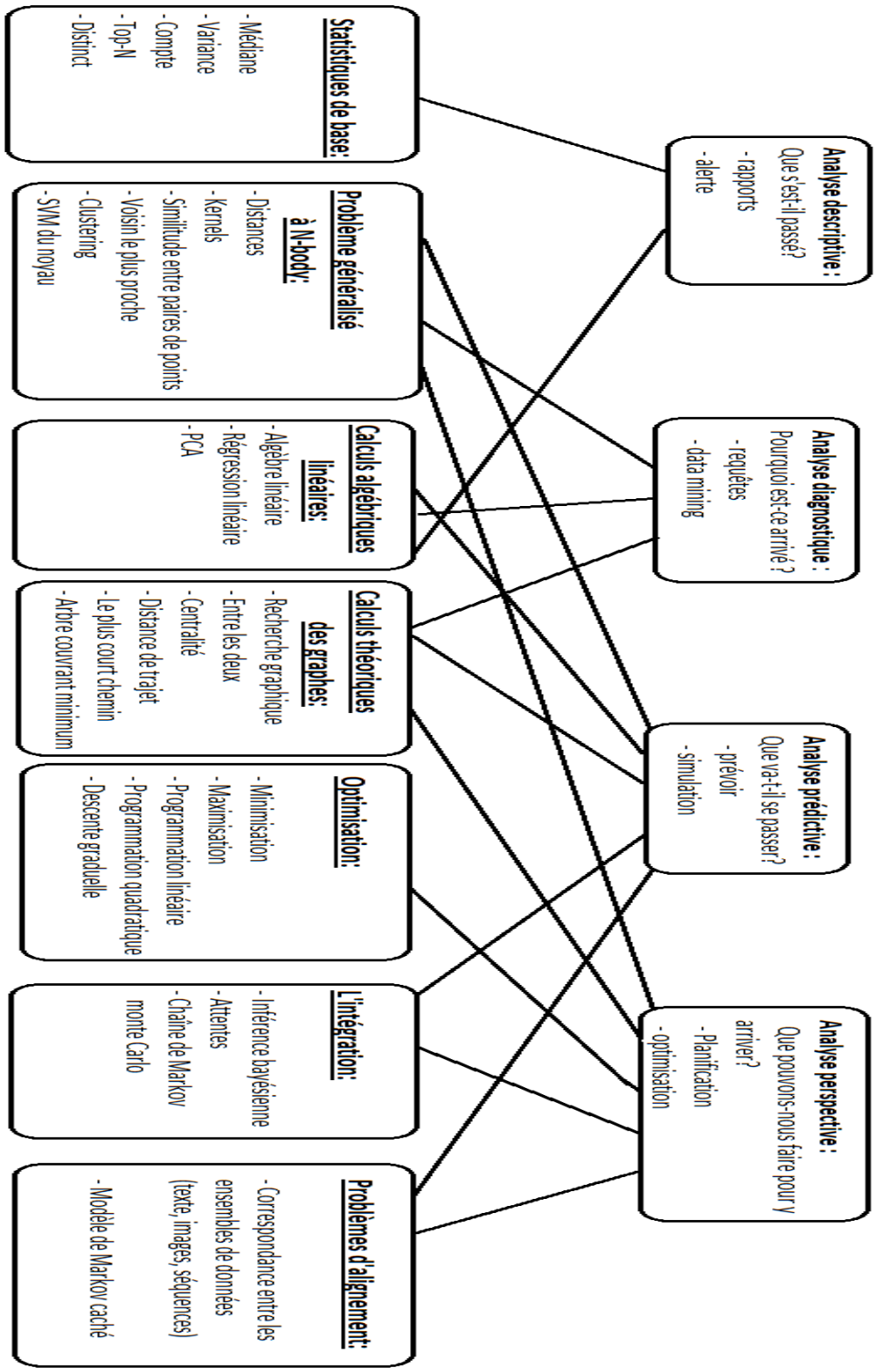

Figure 12 : Cartographie entre les types d'analyse et les tâches de calcul [32]

3. Application de l'analyse des données :

L'analyse des données est essentielle pour comprendre les résultats, ou bien pour obtenir des renseignements sur les lacunes en matière de données. Ces analyses nous donnent l'occasion de prendre des décisions en avance pour éviter les dommages prévoir ce qui se passera dans le futur.

Le tableau ci-dessous présente quelque exemple applicatif des méthodes d'analyse des données dans différents domaines :

Marketing	Gestion des risques	Gouvernement	Web	Logistiques	Autre
Modélisation de la réponse	Modélisation du risque de crédit	Évasion fiscale	Analyses de web	Prévision de la demande	Analyse de texte
Modélisation du net lift	Modélisation du risque de marché	Fraude à la sécurité sociale	Analyse des médias sociaux	Analyse de la chaîne d'approvisionnement	Analyse des processus métier
Modélisation de la rétention	Modélisation des risques opérationnels	Blanchiment d'argent	Test multivarié		
Analyse du panier de marché	Détection de fraude	Détection du terrorisme			
Systèmes de recommandation					
Segmentation de la clientèle					

Tableau 3 : Exemples d'applications d'analyse de données [33]

4. Big Data et l'analyse de données :

Le monde est guidé par les données et il est analysé à chaque instant. Le domaine de l'Analyse Des Données (ADD) intervient dans tous les domaines pour extraire le sens des données collectées et pourrait ainsi conduire à un avenir incroyable.

Exemple la construction de nouvelles voitures sûres et autonomes ou bien des médicaments efficaces ou encore améliorer nos décisions avec des machines intelligentes etc.

L'acronyme de l'Analyse Des Données (ADD) peut être différent de celui du big data, Mais c'est la clé pour extraire le sens de toutes les informations que nous recueillons.

Parmi les méthodes d'ADD utilisées pour l'analyse des informations :

4.1 Méthode des k plus proches voisins (kpp ou knn; acronyme anglais):

On va expliquer cette méthode avec un exemple simple :

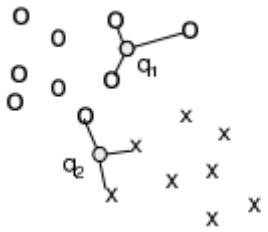

Figure 13 : Exemple explicatif pour knn [34]

L'idée de base est celle illustrée à la figure précédente (figure3) qui représente un classificateur de voisin le plus proche à 3 sur un problème à deux classes dans un espace de caractéristiques bidimensionnel.

Ici on a 2 classes, la classe « O » et la classe « x », et aussi 2 éléments « q1, q2 » à classer : La décision pour q1 est simple, ses trois voisins les plus proches sont de classe O donc il est classé comme O. Mais la situation pour q2 est un peu plus compliquée car il a deux voisins de classe X et un de classe O, cela peut être résolu par un vote à la majorité simple ou par un vote pondéré à distance.

La classification knn (ou kpp) comporte donc deux étapes [34] : La première étape est la détermination des voisins les plus proches.

$$d(q, x_i) = \sum_{f=F} \omega_f \delta(q_f, x_{if}) \quad (1)$$

La deuxième étape est la détermination de la classe à l'aide de ces voisins.

$$vote(y_j) = \sum_{c=1}^{k} \frac{1}{d(q,x_c)^n} 1(y_j, y_c) \quad (2)$$

4.2 Partitionnement en K-moyennes (acronyme anglais : K-means) :

Le clustering K-means est une méthode couramment utilisée pour partitionner automatiquement un ensemble de données en k groupes, il procède en sélectionnant k centres de cluster initiaux, puis en les affinant de manière itérative comme suit : [35]

1. Par exemple on place 2 (K) centroïdes aléatoires (jaune et bleu) :

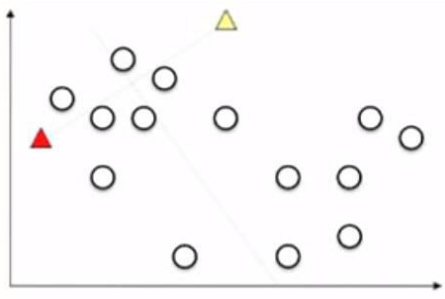

Figure 14 : Placement des centroïdes [36]

2. On calcule la distance des points à chaque centroïde pour voir quels points sont les plus proches à chaque centroïde, en utilisant la distance euclidienne (par exemple), le résultat va être comme suit :

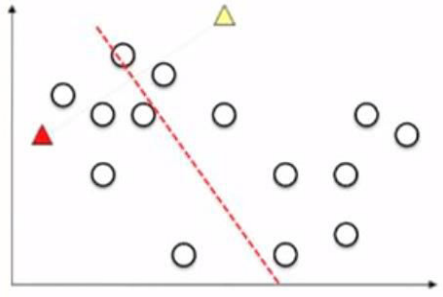

Figure 15 : Division des clusters. [36]

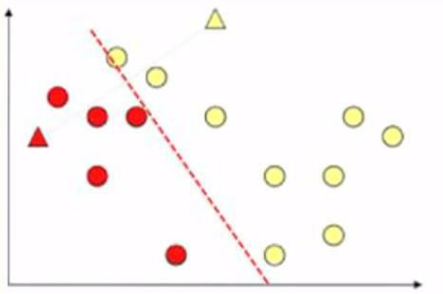

Figure 16 : L'affectation d'après la distance aux centroïdes. [36]

3. On répète la même opération plusieurs fois avec le changement de position des centroïdes. (Le changement doit être pour tous les centroïdes ensemble).

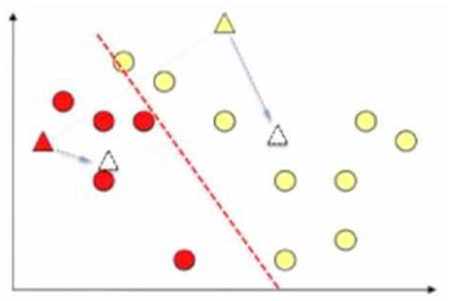

Figure 17 : Changement des centroïdes. [36]

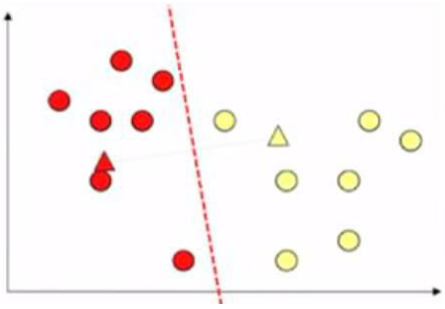

Figure 18 : Le résultat après le changement des centroïdes. [36]

4. L'algorithme converge lorsqu'il n'y a plus de changement dans l'affectation des instances aux clusters.

4.3 Régression linéaire :

Il existe deux types de régression :

4.3.1 Régression simple :

C'est l'une des méthodes statistiques la plus utilisées dans les sciences appliquées et dans les sciences de l'homme et de la société.

Son objectif est double : il consiste tout d'abord à décrire les relations entre une variable privilégiée, appelée variable expliquée (ou dépendante), et plusieurs variables jouant un même rôle par rapport à la première, appelées variables explicatives (ou indépendantes). (c-a-d: c'est la présentation de Y utilisant des X)

Elle permet aussi d'effectuer des prévisions de la variable expliquée en fonction des variables explicatives. Les liaisons entre les variables explicatives exercent une influence très importante sur l'efficacité de la méthode, quel que soit l'objectif dans lequel elle est utilisée. [37]

Avec l'équation générale du modèle de régression linéaire simple : $\mathbf{Y = b_0 + b_1X + \varepsilon}$

Y : la variable à expliquer.

X : la variable explicative

B_0 B_1 : les coefficients de régression (ou les paramètres de modèle). Et $\boldsymbol{\varepsilon}$: l'erreur.

Pour bien expliquer : L'estimation de b0 est :

$$\hat{b}_0 = \bar{y} - \hat{B}_1 \times \bar{x} \qquad (3)$$

$$\bar{y} = \frac{1}{n}\sum y \qquad (4)$$

$$\text{Et : } \bar{x} = \frac{1}{n}\sum x \qquad (5)$$

L'estimation de b1 est : $\hat{b}_1 = \text{covariance}(x, y) \div \text{variance}\ (x)$ (6)

Et ε=0.

Ci-dessous l'exemple représentant la tension artérielle en fonction de l'âge :

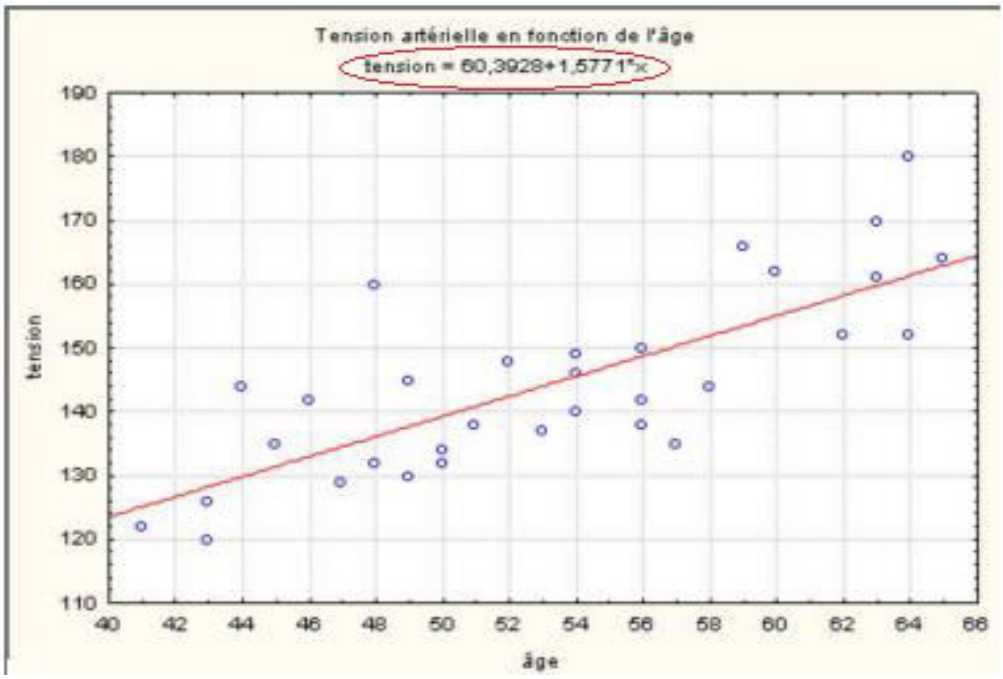

Figure 19 : Exemple explicatif pour la régression simple.[38]

4.3.2 Régression multiple :

Le modèle de régression multiple est une généralisation du modèle de régression simple lorsque les variables explicatives sont en nombre fini.[39]

$$y_i = b_0 + b_1 x_{i1} + b_2 x_{i2} + \cdots + b_p x_{ip} + \varepsilon_i \qquad (7)$$

$$i = 1, \cdots, n$$

4.3.3 Mesure de la qualité de l'ajustement :

Pour mesurer la qualité on doit calculer R^2 :

$$R^2 = \frac{SCE}{SCT} \qquad (8)$$

$$\text{Avec } (SCT = \Sigma(y_i - \bar{y})^2) \qquad (9)$$

Ou bien (SCT=SCR+SCE) (10)

$$\text{Et} \quad SCR = \sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \qquad (11)$$

$$\text{Et } SCE = \sum_{i=1}^{n}(\hat{y}_i - \bar{y})^2 \qquad (12)$$

SCE représente : la variance expliquée par la régression.

SCR représente : la variance résiduelle ou non expliquée.

- si R2 = 0, le modèle n'explique rien, les variables X et Y ne sont pas corrélées linéairement.
- si R2 = 1, les points sont alignés sur la droite, la relation linéaire explique toute la variation.

5. Exemples d'applications d'analyse de données avec les données massives (BIG DATA) existant dans la littérature :

Le Big Data a changé la façon dont nous gérons, analysons et exploitons les données dans n'importe quel secteur. Les domaines les plus prometteurs sont les secteurs des prévisions météorologiques de la santé et du commerce électronique.

5.1 Prévision météorologique

L'augmentation des changements météorologiques évidents devient un problème sérieux. Les fluctuations météorologiques quotidiennes attirent non seulement l'attention des météorologues mais aussi des analystes, en particulier des données de prévision. La recherche sur le changement climatique est nécessaire pour obtenir de nombreux avantages, tels que sauver des vies, surmonter les risques et augmenter les profits et la qualité de vie en fonction des conditions météorologiques.

On peut prendre un exemple comme le tonnerre profond, la détection des information sur ce tonnerre avec des capteurs peut fournir des informations importantes telles que l'évaluation des zones où les inondations sont plus susceptibles de se produire, l'estimation de la direction et de l'ampleur des tempêtes tropicales, la détermination de la quantité de fortes chutes de neige, des précipitations et la chute de lignes électriques dans une zone, l'estimation des zones où les routes et les ponts sont endommagés et bien d'autres. [39]

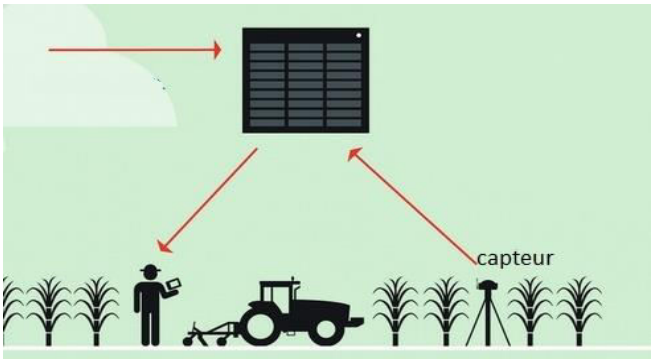

Figure 20 : Explication pour la détection de tonnerre [39]

5.2 La santé (Health care) :

L'analyse des soins de santé a le potentiel de réduire les coûts de traitement, de prévoir les épidémies, d'éviter les maladies évitables et d'améliorer la qualité de vie globale. L'analyse prédictive est l'une des plus grandes tendances de l'intelligence d'affaires deux années de suite, mais les applications potentielles vont bien au-delà des affaires et bien plus loin dans le futur. Optum Labs, une collaboration de recherche américaine, a collecté les EHR (Electronic health records) de plus de 30 millions de patients pour créer une base de données pour des outils d'analyse prédictive qui amélioreront la prestation des soins. [40-41]

5.3 Commerce électronique

En raison des défis et des opportunités engendrés par la révolution de l'information, l'analyse Big Data (BDA : big data Analytics) est devenue un nouveau domaine d'innovation et de concurrence dans le domaine du commerce électronique (ou e-commerce).

L'analyse Big data utilise la dynamique des personnes, des processus et de la technologie pour convertir les données en informations afin de parvenir à une prise de décision fiable et à des solutions aux problèmes commerciaux, augmentant ainsi la valeur des activités de commerce électronique. [42]

Année	Croissance du nombre de clients e-commerce dans le monde (en millions)	Croissance des ventes e-commerce par client dans le monde (en DOLLARS AMÉRICAINS$)	Croissance de l'analyse du Big Data (BDA) marché mondial (en milliards)
2011	792.6	1162	7.3
2012	903.6	1243	11.8
2013	1015.8	1318	18.6
2014	1124.3	1399	28.5
2015	1228.5	1459	38.4
2016	1321.4	1513	45.3

Tableau 4 : La croissance mondiale du commerce électronique et de l'analyse du Big Data [42].

6. Conclusion :

Après cette étude, on peut confirmer que l'analyse des données est un ensemble de méthodes essentielles, utilisées pour traiter les données de nature hétérogène et extraire des informations importantes. Nous allons présenter dans le chapitre suivant l'application d'un modèle proposé pour l'analyse de données et son implémentation**.**

Chapitre 3
Conception et Implémentation

1.Introduction :

Il n'existe pas de choix objectif pour un modèle d'analyse statistique de données. Le modèle qui sera considéré le meilleur est le plus prédictif et dont la justification théorique est la plus élaborée. Dans ce chapitre, nous allons présenter la conception de notre système suivi par son implémentation et terminera avec la présentation des résultats.

2. Modélisation et conception :

2.1 Méthodologie et objectifs

Le processus d'analyse prédictive suivi dans notre projet est constitué des étapes suivantes :

1. La compréhension des objectifs : consiste à comprendre les questions auxquelles on essaie d'apporter et de prédire une réponse.
2. La définition du modèle prédictif selon la forme, la taille et la complexité des données.
3. Le test et vérification de la fiabilité du modèle sur les données existantes permet son évaluation, la réalisation des corrections et l'application des prédictions aux nouvelles données.

Dans ce travail, on a choisi d'appliquer une méthode de classification non supervisée (K-means) pour l'étude d'une population (Dataset). Ceci va permettre de regrouper les individus en plusieurs classes : ces classes sont les plus distinctes possibles et les individus d'une même classe sont les plus semblables possible.

On applique, ensuite une méthode supervisée (Régression Linéaire) sur le résultat obtenu pour une amélioration de la classification et la prédiction de nouveaux cas. Les données étant massives, le framework Spark (présenté dans le chapitre 1) a été adopté pour faire ce traitement.

2.2 Architecture proposée :

L'architecture du modèle proposé est représentée dans la figure suivante. Dans ce qui suit, on expliquera en détail chaque étape suivie.

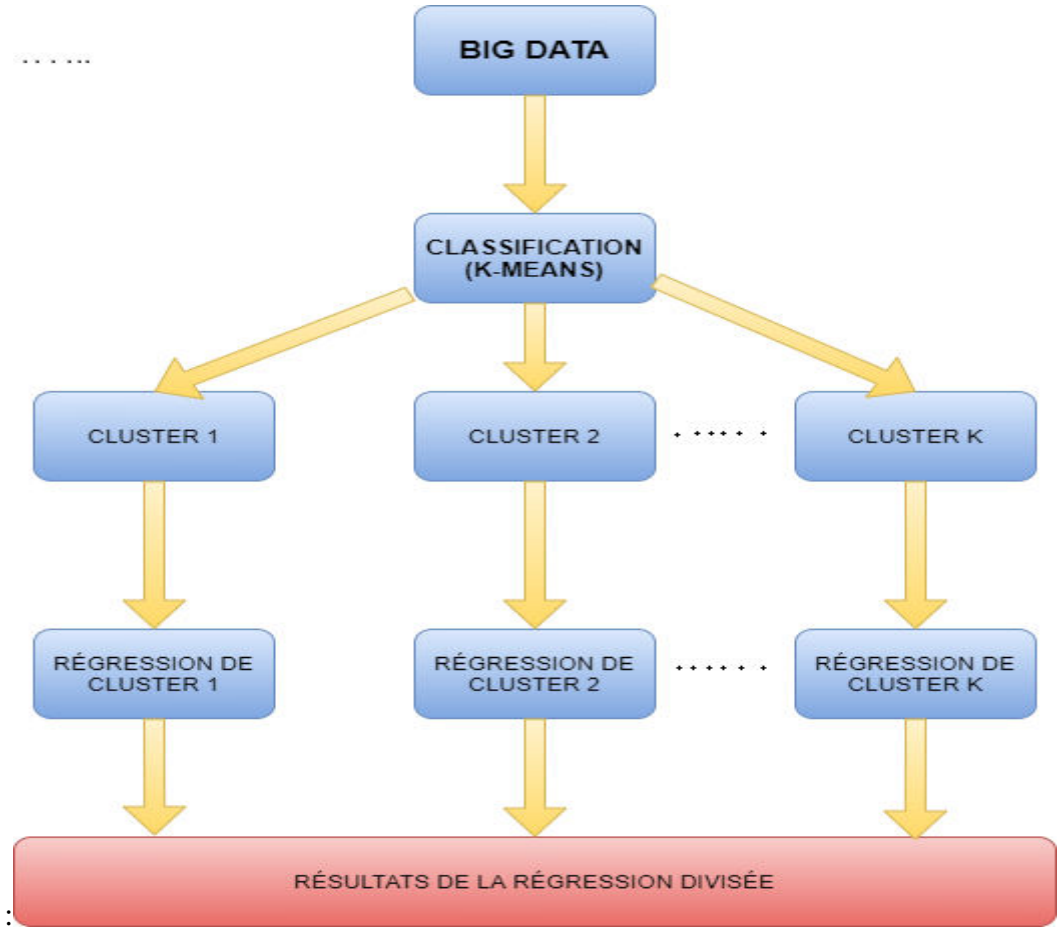

Figure 21 : Architecture proposée

2.3 Diagramme de cas d'utilisation :

On représente dans ce qui suit le diagramme d'utilisation de notre application :

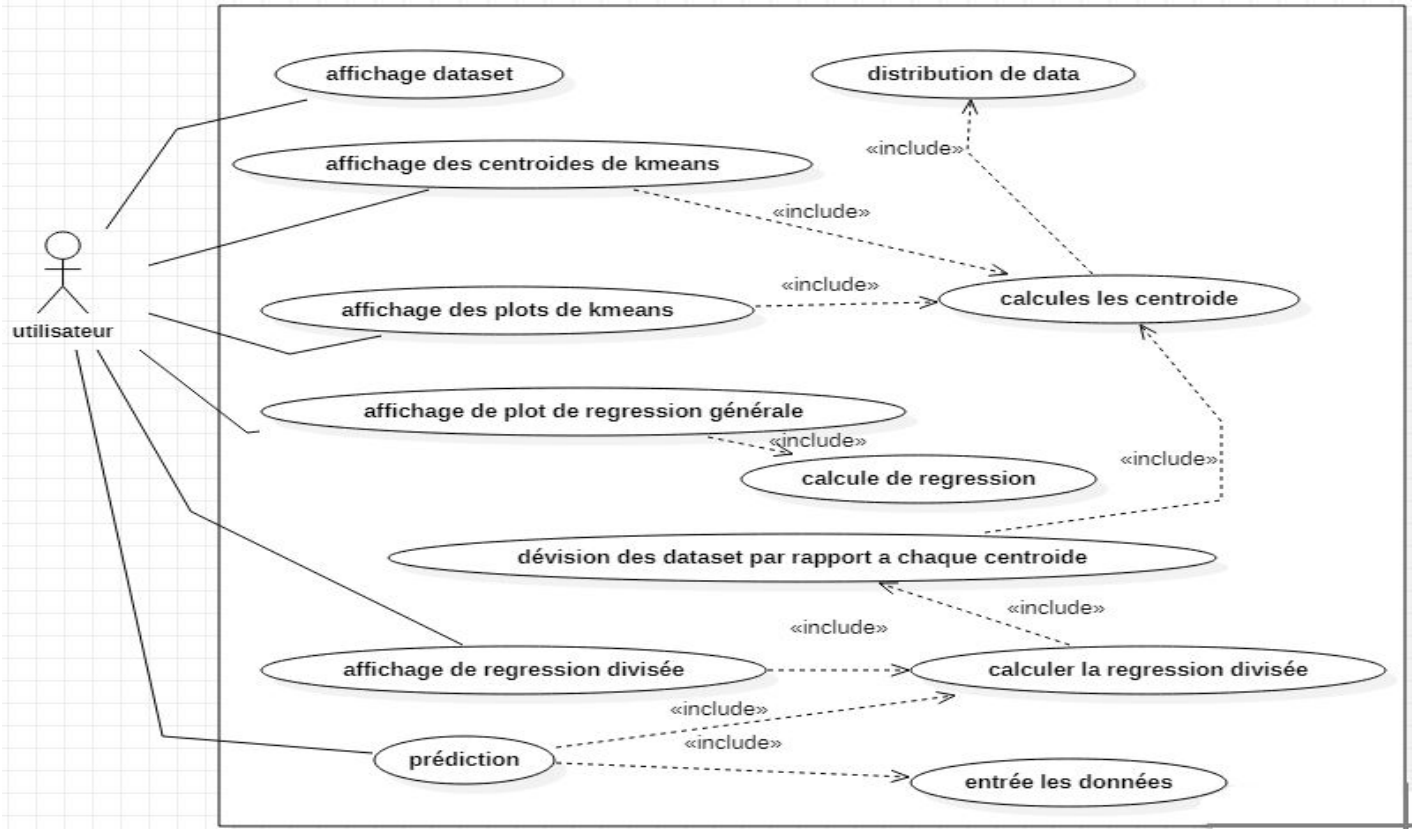

Figure 22 : Diagramme de cas d'utilisation.

Cas d'utilisation	Affichage des centroïdes k-means
Acteur principal	L'utilisateur
Objectif	Présentation du centre de chaque cluster
Pré-condition	Calculer des centroïdes
Post-condition	L'affichage des centroïdes
Scénario principal	• Le système affiche les centroïdes

Table 5 : Scénario affichage des centroïdes k-means

Cas d'utilisation	Affichage de plot de k-means
Acteur principal	L'utilisateur
Objectif	Présentation de chaque cluster
Pré-condition	Calcule les centroïdes
Post-condition	Le plot de k-means affiché
Scénario principal	• Le système calcule les centroïdes • Le système affiche en couleur chaque cluster.

Table 6 : Scénario d'affichage plot de k-means

Cas d'utilisation	Affichage des plots de régression générale
Acteur principal	L'utilisateur
Objectif	Présentation de régression de notre population
Pré-condition	Calcul de régression générale
Post-condition	L'affichage de régression générale
Scénario principal	• Le système calcule la régression pour la dataset entrée. • Le système affiche la régression générale

Table 7 : Scénario d'affichage des plots de la régression générale.

Cas d'utilisation	Affichage de régression divisée
Acteur principale	L'utilisateur
Objectif	Présentation la régression de chaque cluster.
Pré-condition	Calculer de régression divisée
Post-condition	L'affichage de la régression divisée
Scénario principal	• Le système calcul les centroïde • Le système décompose la dataset par rapport à chaque centroïde • Le système affiche la régression divisée

Table 8 : Scénario de l'Affichage de régression divisée.

Cas d'utilisation	Prédiction
Acteur principale	L'utilisateur
Objectif	Prédire les résultats d'après les paramètres d'entrée
Pré-condition	Entrée les données Calculer la régression divisée
Post-condition	Les résultats de Prédiction
Scénario principal	• L'utilisateur entre les facteurs • Le système calcul et affiche la prédiction

Table 9 : Scénario de prédiction.

2.4 Diagramme de séquence :

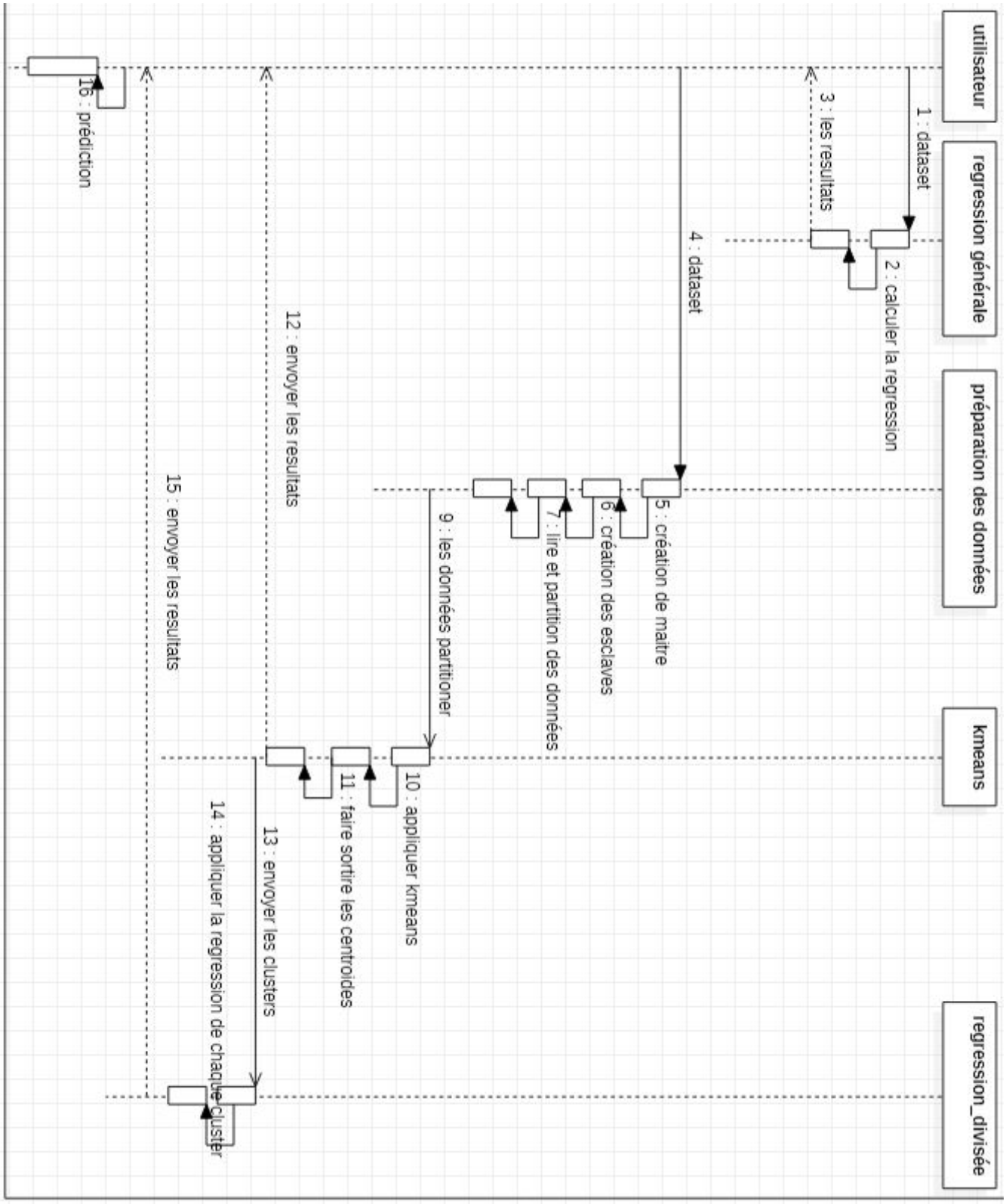

Figure 23 : Diagramme de séquence du système.

2.5 Diagramme de séquence de la Prédiction :

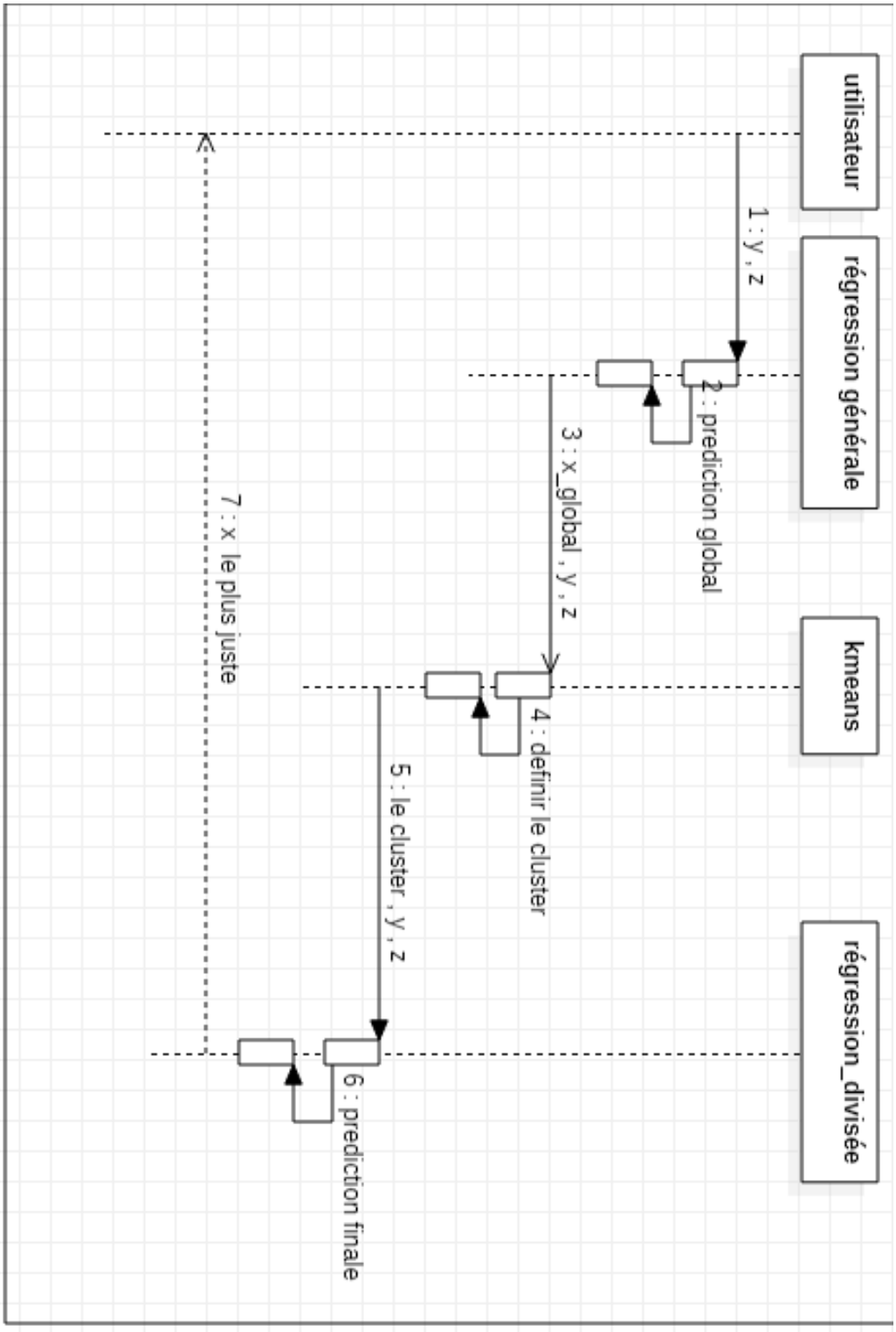

Figure 24 : Diagramme de séquence pour l'opération de prédiction.

2.6 Modalisation d'exécution de K-means avec spark :

Le RDD est le cœur du framework Spark. On représente ci-dessous l'exécution du K-means avec RDD_Spark :

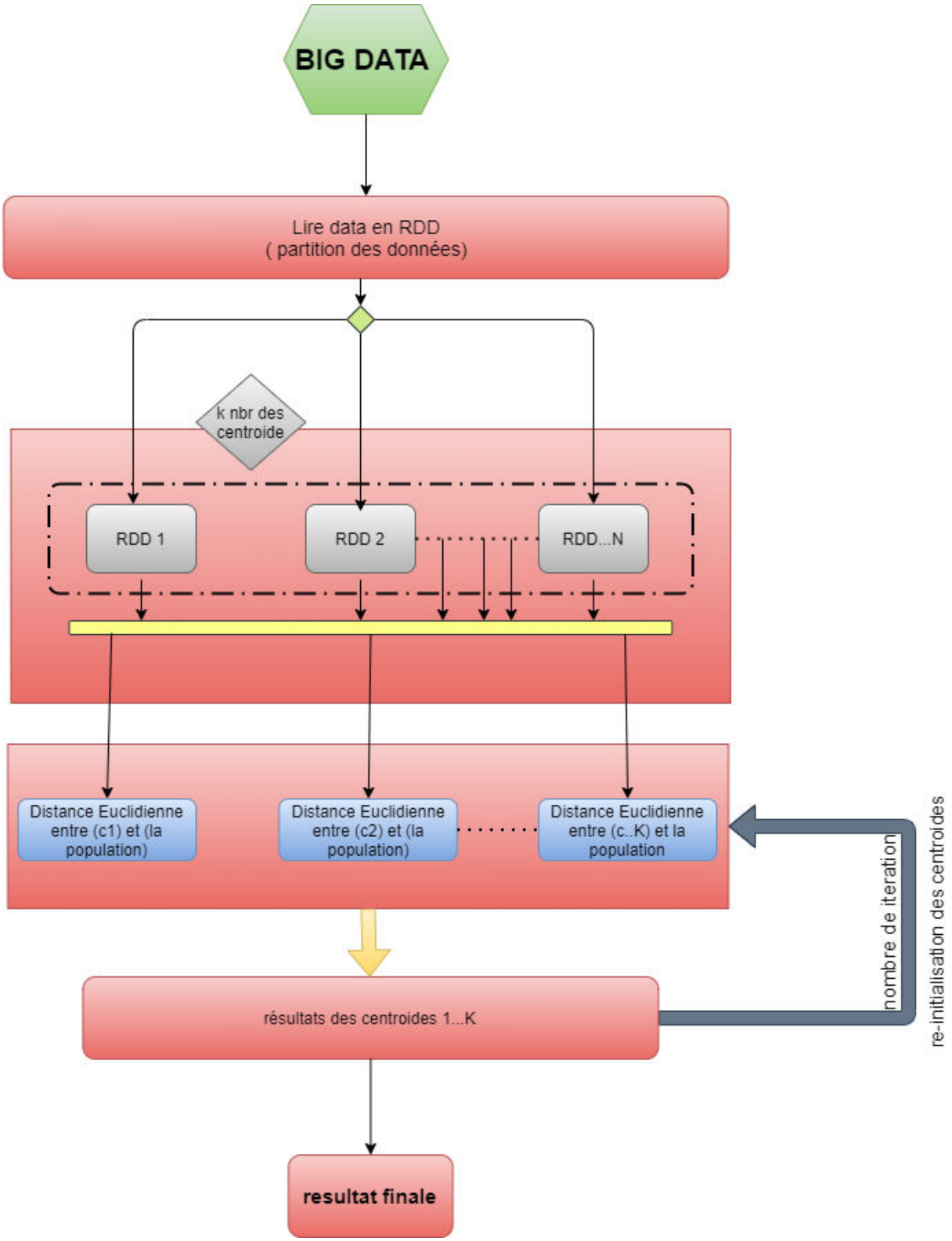

Figure 25 : Exécution de l'algorithme k-means avec RDD_SPARK

3. Implémentation

On décrit ci-dessous les étapes d'executions sur Spark :

3.1.1 Creation de Maitre :

A chaque application Spark, la première opération consiste à se connecter au Maître Spark et à obtenir une session Spark. C'est une opération qu'il faut faire à chaque foi :

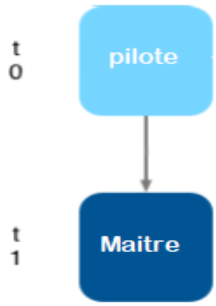

Figure 26 : Création de maitre sur spark.

Le pilote se connecte au maître et obtient une « session Spark ». La flèche indique le déroulement de la séquence : à t0, nous démarrons notre application et à t1, nous obtenons notre session Spark.

3.1.2Chargement du fichier CSV :

Ensuite, on **demande** à charger les données contenues dans notre fichier CSV, Spark peut utiliser la lecture distribuée via les différents nœuds du cluster, pour cela, il s'appuie sur des esclaves (ouvriers). On prend l'exemple de 3 esclaves (3 partitions) pour expliquer.

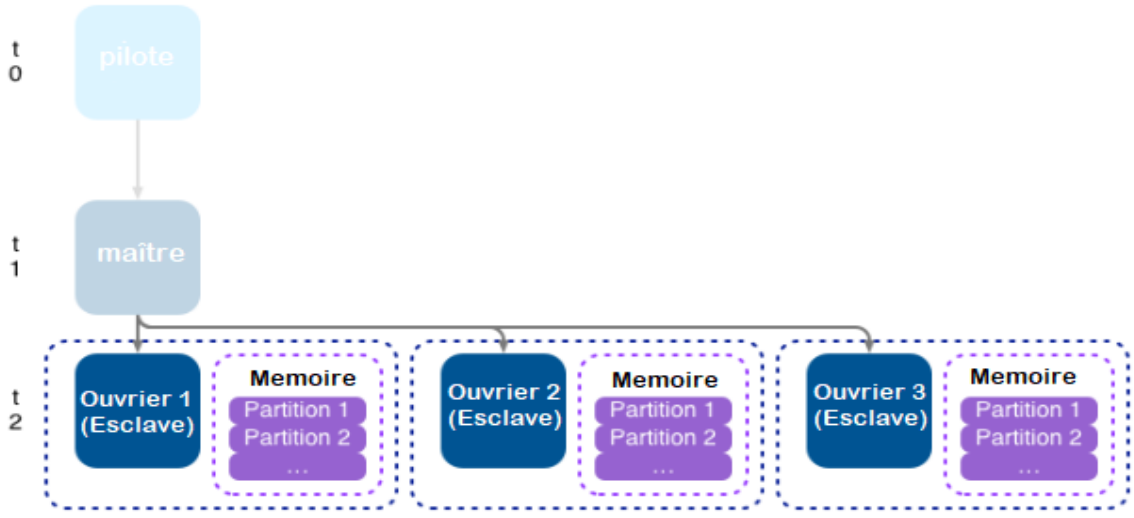

Figure 27 : Création des esclaves.

À t2, le maître ordonne aux ouvriers de charger le fichier. Les esclaves créent des tâches pour lire le fichier. Chaque esclave a accès à la mémoire du nœud et attribue une partition de

mémoire à la tâche. Les tâches sont créées en fonction des ressources disponibles. Le Maitre peut créer plusieurs tâches et attribuer une partition de mémoire à la tâche.

Les tâches fonctionnelles sont en cours d'exécution (elles ont également un point vert), contrairement aux tâches non fonctionnelles (d'autres applications par exemple) ont un point rouge.

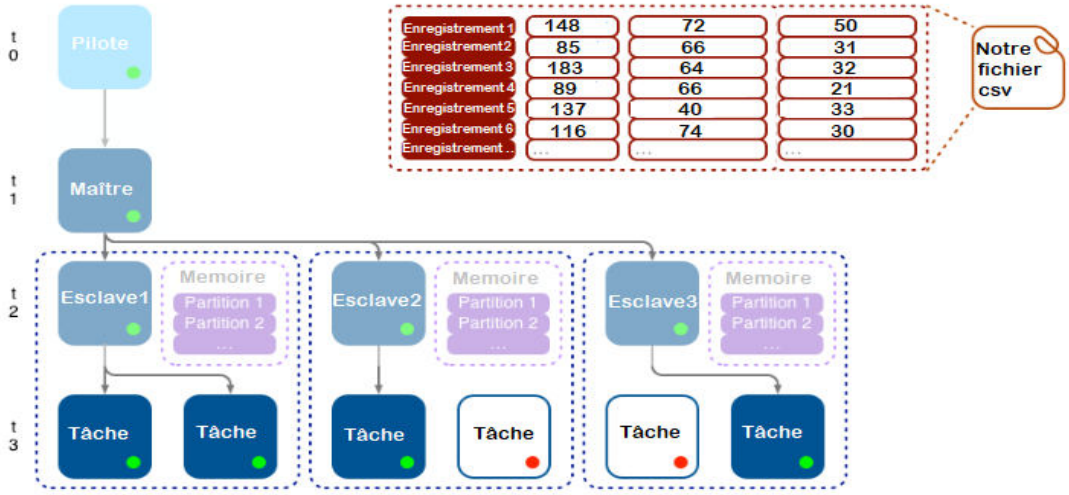

Figure 28 : La sélection du dataset.

La figure suivante montre l'enregistrement en cours de copie du fichier CSV vers la partition pendant le processus de lecture, dans le R ▶ P (enregistrement (record) vers la partition)

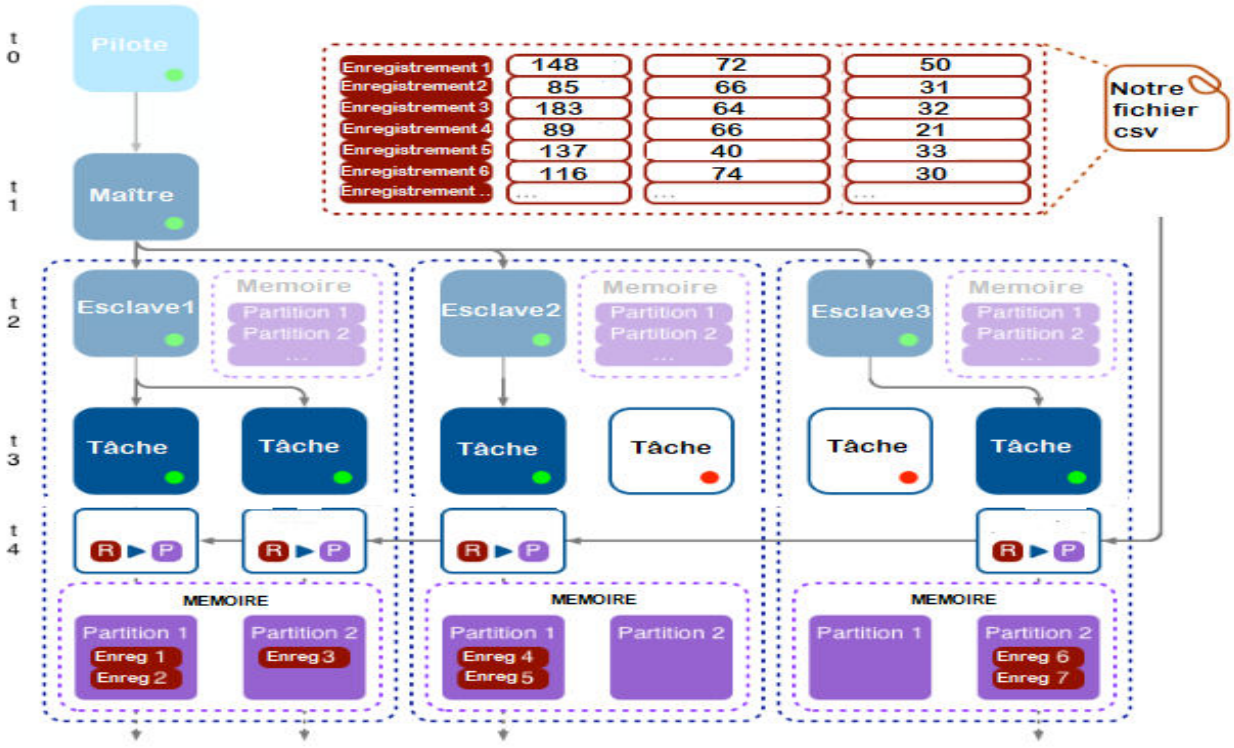

Figure 29 : Enregistrement vers la partition.

À t4, chaque tâche continue en lisant une partie du fichier CSV. Au fur et à mesure que la tâche lit des lignes, elle les stocke dans une partition dédiée.

3.1.2 Transformation des données :

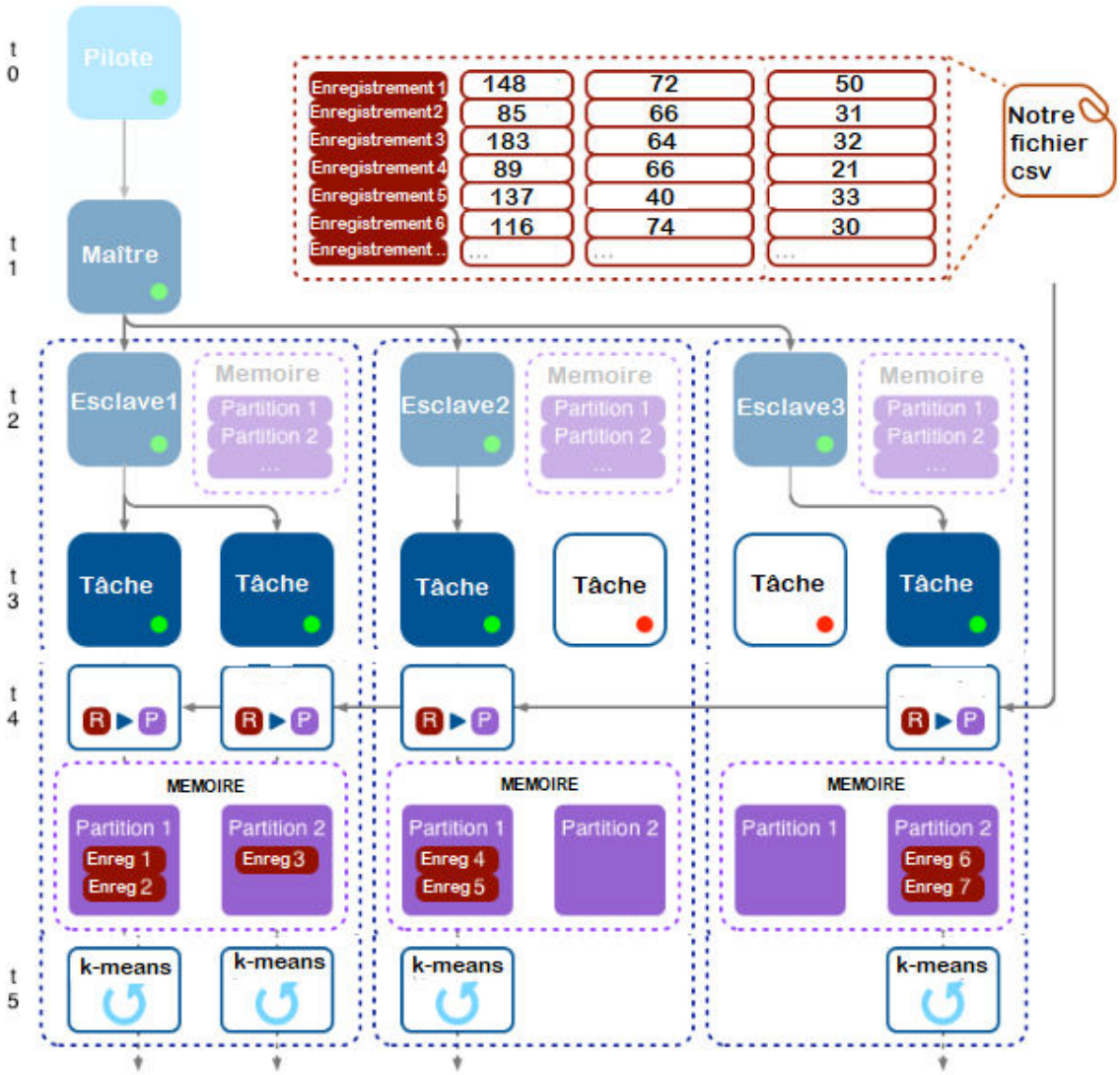

Figure 30 : Exécution du code k-means.

Une fois les données chargées, à t5, nous pouvons traiter les enregistrements, Le traitement est l'application du code de partitionnement k-means.

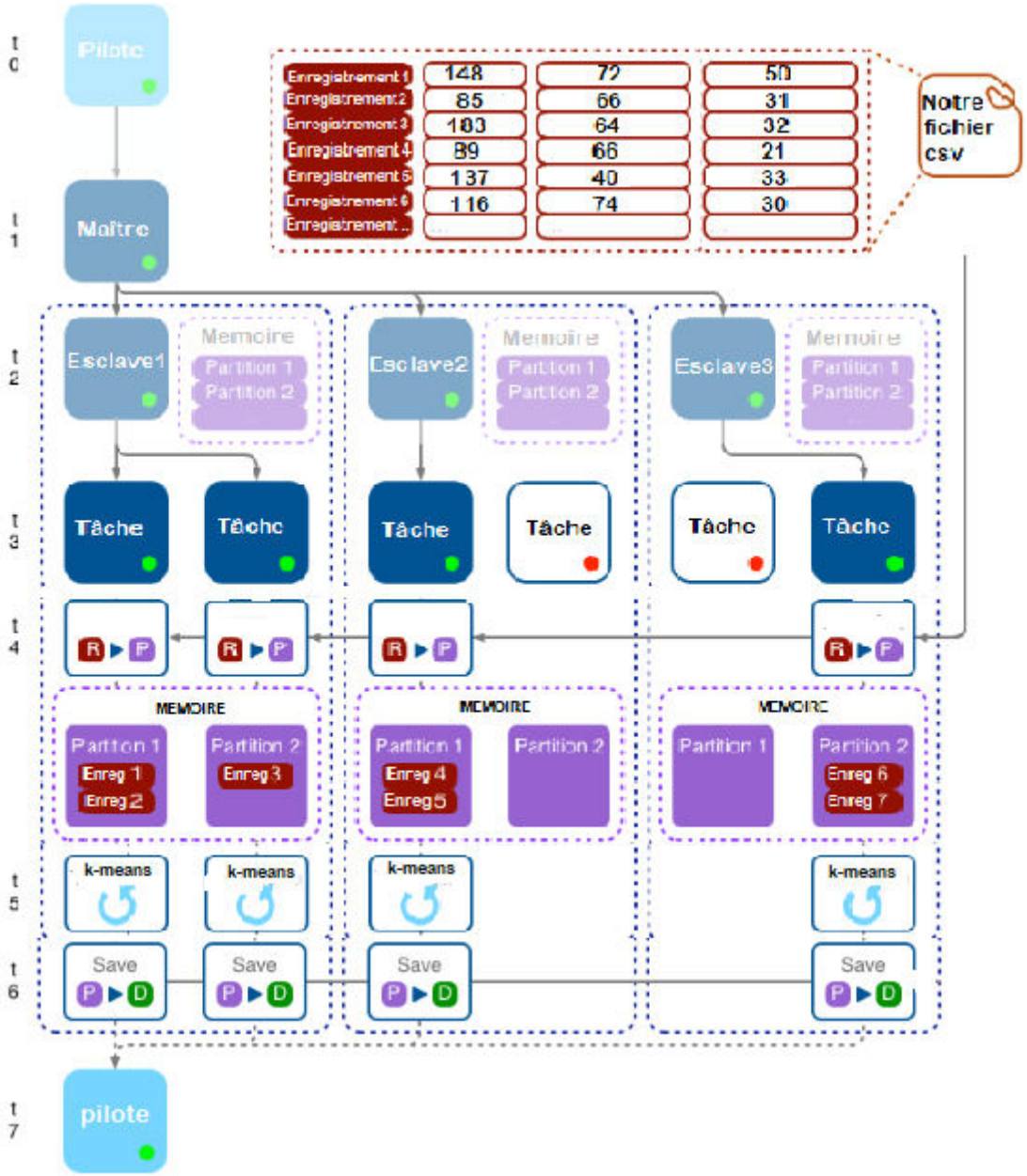

Figure 31 : Renvoyer les résultats à l'utilisateur.

Après lecture, partition et transformation du dataset, on peut ainsi sauvegarder à (t6) les résultats (centroïdes de chaque cluster), et les renvoyer à l'utilisateur (t7).

Enfin, nous citons ci-après les points importants de ce processus :

- L'ensemble de données a été divisé en partitions sur les esclaves, pas sur le pilote.
- L'ensemble du traitement a eu lieu dans les esclaves.
- Les résultats de k-means sont des centroïde (dépend des K entrées).

Après avoir présenté dans les sections précédentes les concepts de base liés à notre travail et en se basant sur l'architecture proposée, un système d'analyse de données a été implémenté Dans cette section, je défini d'abord le choix du logiciel et matériel pour la réalisation de l'application. Ensuite je présente les étapes d'installation suivie pat le déroulement du programme d'analyse de données appliqué sur le Dataset choisi. Enfin, je terminerai par une discussion des résultats.

3.2. Les ressources matérielles et logicielles :

Dans cette étape, je présenterai les ressources matérielles et logiciels utilisées :

3.2.1. Matériels utilisés :

L'implémentation de notre système a été réalisée sur une machine possédant les caractéristiques suivantes :

Processeur : 1.70 GHz

Mémoire :4.00 Go

Disque dur :500 GB

3.2.2. Logiciels utilisés :

a) Système d'exploitation : Windows 10

b) Outils de développement :

Python version 3.2 : Python est un langage de programmation de haut niveau avec une syntaxe simple et une puissance remarquable.

Bibliothèque : pyspark, pandas, pyplot

c) Apache Spark version 2.4.6 : il permet d'effectuer des traitements sur de large volume de donnée.

4. Préparation de données :

Après avoir installé les logiciels requis ; on passe à la préparation des données et l'implémentation des fonctions. Afin de tester le système, on a utilisé un Dataset médical disponible sur :

https://www.kaggle.com/

4.1 Notre dataset :

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1
...	...	...	...	...	...	...	...	...	...
763	10	101	76	48	180	32.9	0.171	63	0
764	2	122	70	27	0	36.8	0.340	27	0
765	5	121	72	23	112	26.2	0.245	30	0
766	1	126	60	0	0	30.1	0.349	47	1
767	1	93	70	31	0	30.4	0.315	23	0

768 rows × 9 columns

Figure 32 : dataset choisi.

Grossesses : nombre de fois enceinte

Glucose : concentration plasmatique de glucose

BloodPressure : tension artérielle diastolique (mm Hg)

Insuline : insuline sérique 2 heures (mu U / ml)

BMI : Indice de masse corporelle (poids en kg / (taille en m) ^ 2)

Age : Âge (années)

5. Description détaillée :

La figure suivante représente la création de Maitre (Master)

```
#la création de MASTER

sc =SparkContext()
appName = "k-kmaens dans Spark"
spark = SparkSession \
    .builder \
    .appName(appName) \
    .config("spark.some.config.option", "some-value") \
    .getOrCreate()
print ("la creation de maitre a été bien faite")
```

```
la creation de maitre a été bien faite
```

Figure 33 : création de maitre.

La figure suivante représente la lecture de Dataset en RDD, dans le quel SparkConext(sc) est responsable de la décomposition en RDD.

Et aussi l'application du partitionnement (méthode k-means) avec l'initialisation du nombre des clusters, et du nombre d'itérations.

```
#Lire dataset et choix des facteur utiliser
dataset = sc.textFile("C:/Users/2016/Downloads/diabetess.csv"). \
        map(lambda x: x.split(",")).\
        map(lambda x: [float(x[1]), float(x[2]),float(x[7])])

#notre model de kmeans
model = KMeans.train(dataset, k = 3, maxIterations = 200)
```

Figure 34: lecture de dataset.

```
In [6]: dataset.take(10)

Out[6]: [[148.0, 72.0, 50.0],
         [85.0, 66.0, 31.0],
         [183.0, 64.0, 32.0],
         [89.0, 66.0, 21.0],
         [137.0, 40.0, 33.0],
         [116.0, 74.0, 30.0],
         [78.0, 50.0, 26.0],
         [115.0, 0.0, 29.0],
         [197.0, 70.0, 53.0],
         [125.0, 96.0, 54.0]]
```

Figure 35 : notre dataset en RDD.

```
print ('nombre des RDD créer ---->',dataset)

nombre des RDD créer ----> PythonRDD[43] at RDD at PythonRDD.scala:53
```

Figure 36 : nombre des RDD créer.

La figure suivante représente les résultats de la méthode de partitionnement k-means, dans lequel, on affiche les coordonnés de chaque centroïde.

```
In [10]: #les centroides
         model.clusterCenters

Out[10]: [array([122.509375,  75.0125  ,  34.546875]),
          array([92.3041958 , 59.11888112, 28.54195804]),
          array([168.17901235,  75.06790123,  38.95679012])]
```

Figure 37 : les résultats de Kmeans.

La figure suivante représente le partitionnement des données par rapport au centroïdes, donc chaque table représente les coordonnées des points qui appartiennent à chaque centroïde (c.-à-d. qui forment la partition).

```
In [19]: #la division par raport a les centroides
         c0=0
         c1=0
         c2=0

         for m in range(dataset.count()):
             AA = model.predict(array([x[m], y[m], z[m]]))
             if AA==0:
                 df0.loc[c0] = [x[m], y[m], z[m]]
                 c0=c0+1

             elif AA==1:
                 df1.loc[c1] = [x[m], y[m], z[m]]
                 c1=c1+1

             elif AA==2 :
                 df2.loc[c2] = [x[m], y[m], z[m]]
                 c2=c2+1

         print('on a diviser la dataset pour chaque centroide')

         on a diviser la dataset pour chaque centroide
```

Figure 38 : division de dataset.

df0

	x0	y0	z0
0	137.0	40.0	33.0
1	116.0	74.0	30.0
2	125.0	96.0	54.0
3	110.0	92.0	30.0
4	139.0	80.0	57.0
...	...	...	...
315	106.0	76.0	26.0
316	101.0	76.0	63.0
317	122.0	70.0	27.0
318	121.0	72.0	30.0
319	126.0	60.0	47.0

320 rows × 3 columns

df1

	x1	y1	z1
0	85.0	66.0	31.0
1	89.0	66.0	21.0
2	78.0	50.0	26.0
3	115.0	0.0	29.0
4	100.0	0.0	32.0
...	...	...	...
281	81.0	74.0	32.0
282	108.0	62.0	25.0
283	88.0	58.0	22.0
284	89.0	62.0	33.0
285	93.0	70.0	23.0

286 rows × 3 columns

df2

	x2	y2	z2
0	148.0	72.0	50.0
1	183.0	64.0	32.0
2	197.0	70.0	53.0
3	168.0	74.0	34.0
4	189.0	60.0	59.0
...	...	...	...
157	162.0	62.0	50.0
158	181.0	88.0	26.0
159	154.0	78.0	45.0
160	190.0	92.0	66.0
161	170.0	74.0	43.0

162 rows × 3 columns

Figure 39 : Partitionnement en groupes.

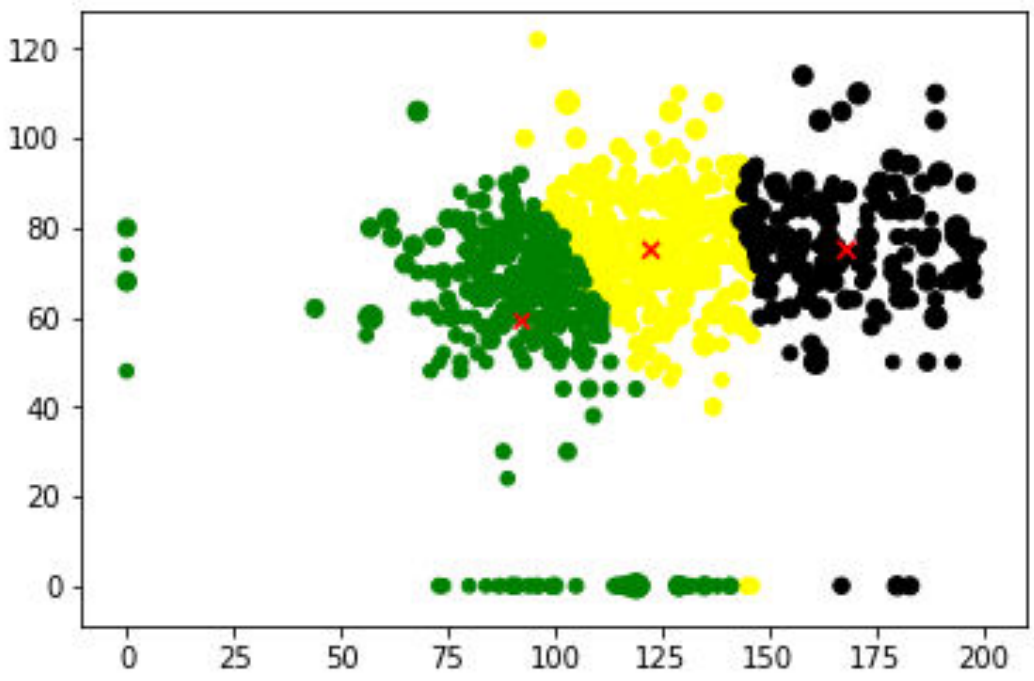

Figure 40 : Représentation des individus de la population en 2D.

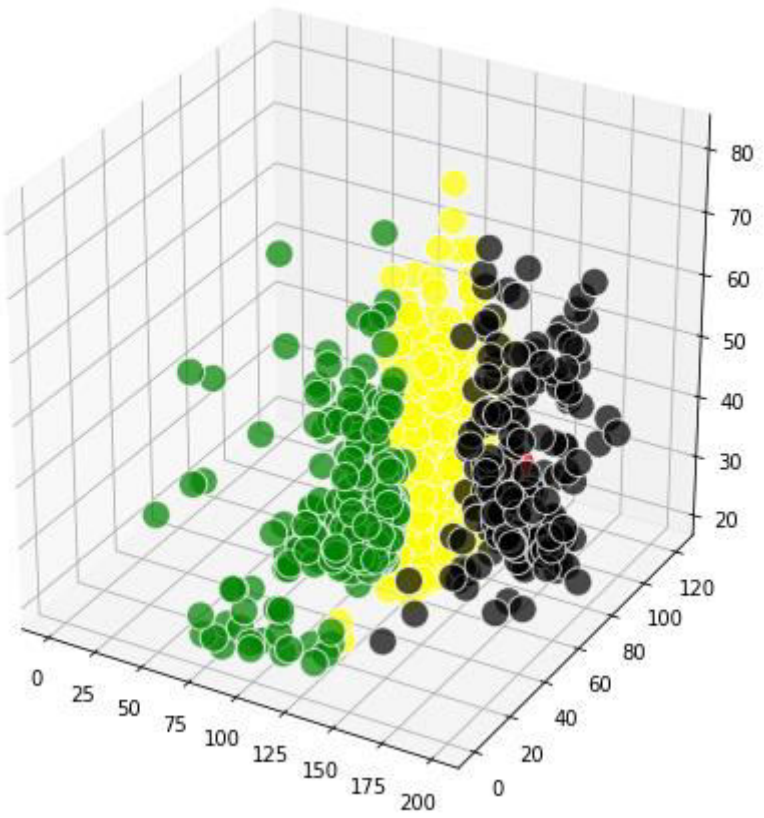

Figure 41 : Représentation des individus de la population en 3D.

La figure suivante représente la régression générale sur nos données, on a choisi les variables (colonnes) de BloodPressue, l'Age et Glucose pour applique la régression sur ces derniers

```
In [56]: df= pandas.read_csv("C:/Users/2016/Downloads/diabetes.csv")

         xx = df[['BloodPressure','Age']]
         yy = df['Glucose']

         regr_generale = linear_model.LinearRegression()
         regr_generale.fit(xx, yy)

         xx,yy
```

Figure 42 : Régression générale.

La figure ci-dessous représente la régression linéaire appliquée dans chaque cluster (régression divisée) en utilisant les résultats du partitionnement du dataset (**figure 39**).

```
In [33]: #regression pour cluster 1
         xgroupe0= df0[['y0', 'z0']]
         ygroupe0 = df0['x0']

         reg_groupe0 = linear_model.LinearRegression()
         reg_groupe0.fit(xgroupe0, ygroupe0)

         print(' la regression de cluster 1 a été bien faite')
         print('  ')

         #regression pour cluster 2
         xgroupe1= df1[['y1', 'z1']]
         ygroupe1 = df1['x1']

         reg_groupe1 = linear_model.LinearRegression()
         reg_groupe1.fit(xgroupe1, ygroupe1)

         print(' la regression de cluster 2 a été bien faite')
         print('  ')

         #regression pour cluster 3
         xgroupe2= df2[['y2', 'z2']]
         ygroupe2 = df2['x2']

         reg_groupe2 = linear_model.LinearRegression()
         reg_groupe2.fit(xgroupe2, ygroupe2)

         print(' la regression de cluster 3 a été bien faite')
         print('  ')
```

Figure 43 : Régression divisée.

5.1 Prédiction de données :

Dans la figure suivante on présente la fonction de prédiction générale qui permet à l'utilisateur de retrouver les valeurs de taux de glucose en fonction de la tension et de l'age(âge, tension et glucose sont les variables choisies) .

```
#prédiction avec la regression_génélrale
la_tension=66
l_age=31
p_glocose=regr_generale.predict([[la_tension,l_age]])
print('d aprés la regression globale on trouve le glucose = ',p_glocose)
```

```
d aprés la regression globale on trouve le glucose =  [118.94073543]
```

Figure 44 : prédiction globale.

La figure suivante représente la prédiction divisée, cette prédiction est plus spécifique que la régression générale, donc elle donne la valeur la plus proche de la réalité que la régression générale.

```
#prédiction par regression divisé
p=model.predict(array([p_glocose,la_tension,l_age]))

if p==0:
    print( 'd aprés la regression divisée on trouve le glucose = ',reg_groupe0.predict([[la_tension,l_age]]))
elif p==1:
    print( 'd aprés la regression divisée on trouve le glucose = ', reg_groupe1.predict([[la_tension,l_age]]))
elif p==2:
    print( 'd aprés la regression divisée on trouve le glucose = ', reg_groupe2.predict([[la_tension,l_age]]))
```

```
d aprés la regression divisée on trouve le glocose =  [124.06081315]
```

Figure 45 : Prédiction divisée.

Le système proposé permet ainsi d'offrir une prédiction plus exacte que l'utilisation de la prédiction générale. Apres partitionnement avec la méthode k-means et l'application de la régression divisé à l'intérieur de la partition (cluster), le résultat de la prédiction est plus proche des valeurs exacte existantes. Ceci permet de retrouver les valeurs manquantes et de compléter les données absentes du dataset.

5.2 Tableau comparatif pour les résultats :

Glucose (dataset)	La tension (dataset)	L'Age (dataset)	Prédiction générale de Glucose	Prédiction divisée de Glucose
183	**64**	**32**	**119**	**144**
118	**72**	**46**	**129.7**	**120.4**
170	**120**	**80**	**159**	**163.6**
150	**62**	**38**	**122**	**122.8**
115	**70**	**35**	**132.3**	**120.8**

Tableau 10 : comparaison des résultats.

D'après le résultat on confirme l'efficacité de notre système, les prédictions divisées sont les plus proche.

6. Conclusions :

Dans ce chapitre, nous avons commencé avec par une introduction, après nous avons proposé une modélisation et conception bien détaillé et on a terminé par l'implémentation de l'application qui représente des captures pour les résultats obtenus.

Conclusion Générale

L'analyse de données est un domaine multidisciplinaire. Il repose principalement sur l'analyse statistique et la fouille de données. L'analyse de données utilise des techniques et des algorithmes d'exploration permettant de découvrir les relations qui relient les données et mettre ainsi les résultats à disposition des utilisateurs. L'application de ces techniques permet de mieux comprendre les données qui nous entourent et de procéder à des améliorations de performances pour anticiper les résultats.

La méthode de partitionnement K-means est largement utilisée dans l'analyse de données. Cette technique est simple et fournit des résultats rapides. Cependant, ses performances ne sont généralement pas aussi compétitives car de légères variations dans les données pourraient entraîner une variance élevée des résultats. D'autre part, la régression linéaire est une méthode d'apprentissage supervisé consistant à apprendre une fonction de prédiction mais à partir de données connues.

Dans ce travail, la méthode du k-means a été combinée avec la régression linéaire. Cette hybridation a permis d'apporter une amélioration aux résultats de classification et aussi prédire les valeurs des données manquantes.

Dans ce projet,

- ✓ Un état de l'art sur les concepts du « Big Data » a été présenté dans le chapitre 1
- ✓ Une étude des méthodes analytiques appliquées au domaine du Big data a été exposée dans le chapitre 2.
- ✓ Une implémentation de la méthode de partitionnement K-means a été proposée. L'algorithme a été combiné avec la méthode de la régression linéaire et validé sur des données massives « Médicale ».
- ✓ L'utilisation du Framework Spark a été d'un grand apport dans le traitement de ces données massives.

Ce travail m'a permis de :

- ✓ Maitriser les concepts clés des domaines en challenge « BIG DATA » et « Analyse de données »
- ✓ Proposer une hybridation de deux méthodes analytiques pour profiter de leurs avantages et améliorer leurs lacunes. Cette hybridation a permis l'amélioration des résultats et aussi faire ressortir des pépites d'informations pour le traitement et la prédiction.
- ✓ Maitriser les concepts clés de « Spark », le Framework de traitement des données massives.
- ✓ Ce système peut être appliqué sur des Datasets avec des données manquantes.

En perspective, ce travail peut être compléter par les point suivants :

- ✓ Implémentation parallèle et récursive de l'algorithme Kmeans
- ✓ Automatisation du choix des paramètres des méthodes utilisées
- ✓ D'autre Datasets peuvent faire aussi l'objet de bases de tests pour valider le système proposé

Ce travail a permis de souligner l'impact des méthodes d'analyse données appliquées dans le cadre des données massives ou « Big Data ».

Références Bibliographiques

[1] Clavert, Frédéric. "Patrick Manning, Big Data In History". Lectures, 2014. Openedition,.

[3] Bahga, Arshdeep, and Vijay Madisetti. Big data science & analytics: A hands-on approach. VPT, 2016.

[4] Beyer, Mark A., and Douglas Laney. "The importance of 'big data': a definition." Stamford, CT: Gartner (2012): 2014-2018.

[5] Gupta, Richa. "Journey from data mining to Web Mining to Big Data." arXiv preprint arXiv:1404.4140 (2014).

[6] Eberendu, Adanma Cecilia. "Unstructured Data: an overview of the data of Big Data." International Journal of Computer Trends and Technology 3.1 (2016): 46-50.

[7] Feldman, Ronen, and James Sanger. The text mining handbook: advanced approaches in analyzing unstructured data. Cambridge university press, 2007.

[8]. Hänig, Christian, Martin Schierle, and Daniel Trabold. "Comparison of structured vs. unstructured data for industrial quality analysis." Proceedings of The World Congress on Engineering and Computer Science. 2010.

[10] Demchenko, Yuri, Cees De Laat, and Peter Membrey. "Defining architecture components of the Big Data Ecosystem." 2014 International Conference on Collaboration Technologies and Systems (CTS). IEEE, 2014.

[13 Nandimath, Jyoti, et al. "Big data analysis using Apache Hadoop." 2013 IEEE 14th International Conference on Information Reuse & Integration (IRI). IEEE, 2013. [15] Meng, Xiangrui, et al. "Mllib: Machine learning in apache spark." The Journal of Machine Learning Research 17.1 (2016): 1235-1241.

[16] Chambers, Bill, and Matei Zaharia. Spark: The definitive guide: Big data processing made simple. " O'Reilly Media, Inc.", 2018.

[17] Mistrík, Ivan, et al., eds. Software architecture for big data and the cloud. Morgan Kaufmann, 2017.

[18] Armbrust, Michael, et al. "Spark sql: Relational data processing in spark." Proceedings of the 2015 ACM SIGMOD international conference on management of data. 2015.
[19] Zaharia, Matei, et al. "Discretized streams: an efficient and fault-tolerant model for stream processing on large clusters." Presented as part of the. 2012.

[20] Xin, Reynold S., et al. "Graphx: A resilient distributed graph system on spark." First international workshop on graph data management experiences and systems. 2013.

[21] Zaharia, Matei, et al. "Resilient distributed datasets: A fault-tolerant abstraction for in-memory cluster computing." Presented as part of the 9th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 12). 2012.

[22] Xin, Reynold S., et al. "Shark: SQL and rich analytics at scale." Proceedings of the 2013 ACM SIGMOD International Conference on Management of data. 2013.

[23] Zaharia, Matei, et al. "Spark: Cluster computing with working sets." HotCloud 10.10-10 (2010): 95.

[26] Bahga, Arshdeep, and Vijay Madisetti. Big data science & analytics: A hands-on approach. VPT, 2016.

[28] Hussain, Mehwish. "Descriptive statistics--presenting your results I." JPMA. The Journal of the Pakistan Medical Association 62.7 (2012): 741-743.

[30] Shah, Nilay D., Ewout W. Steyerberg, and David M. Kent. "Big data and predictive analytics: recalibrating expectations." Jama 320.1 (2018): 27-28.

[32] National Research Council. Frontiers in massive data analysis. National Academies Press, 2013.

[33] Baesens, Bart. Analytics in a big data world: The essential guide to data science and its applications. John Wiley & Sons, 2014.

[34] Cunningham, Padraig, and Sarah Jane Delany. "k-Nearest Neighbour Classifiers--." arXiv preprint arXiv:2004.04523 (2020).

[35] Wagstaff, Kiri, et al. "Constrained k-means clustering with background knowledge." Icml. Vol. 1. 2001.

[37] Foucart, Thierry. "Colinéarité et régression linéaire." Mathématiques et sciences humaines. Mathematics and social sciences 173 (2006).

[39] Abhigna, P., et al. "Analysis of feed forward and recurrent neural networks in predicting the significant wave height at the moored buoys in Bay of Bengal." 2017 International Conference on Communication and Signal Processing (ICCSP). IEEE, 2017.

[41] Dash, Sabyasachi, et al. "Big data in healthcare: management, analysis and future prospects." Journal of Big Data 6.1 (2019): 54.

[42] AKTER, Shahriar et WAMBA, Samuel Fosso. Big data analytics in E-commerce: a systematic review and agenda for future research. Electronic Markets, 2016, vol. 26, no 2, p. 173-194.

[43] Hurwitz, Judith S., et al. Big data for dummies. John Wiley & Sons, 2013.

[44] docshare01.docshare. tips

[45]mMsseguem abdeldjalil , Analyse des données avec apache spark, mémoire master Mssila,2019

[48]Colinéarité et régression linéaire – OpenEdition journals , journals. openedition.org> msh> pdf.

[2] https://www.statista.com (dernier accès 9/6/2020).

[9]https://www.groupe-hli/comprocessus-metiers-donnees-non-structurees. (denier accès 19/09/2020)

[11] https://www.techentice.com/the-data-veracity-big-data (dernier accès05/05/2020).

[12] http://hadoop.apache.org/ (dernier accès 20/09/2020)

[14]https://www.edureka.co/blog/mapreduce-tutorial/ (dernier accès 20/09/2020)

[24] https://data-flair.training/blogs/spark-rdd-operations-transformations-actions/

[25]https://www.upgrad.com/blog/what-is-big-data-types-characteristics-benefits-and-examples/ (dernier accès8/05/2020).

[27] https://www.investopedia.com/terms/d/data-analytics.asp (dernier accès 01/08/2020)

[29] https://www.investopedia.com/terms/d/data-analytics.asp (dernier accès 09/08/2020)

[31] https://www.predictiveanalyticstoday.com/what-is-predictive-analytics/ (dernier accès 10/08/2020)

[36] https://www.kdnuggets.com/2014/05/watch-basics-machine-learning.html (dernier accès 28/08/2020)

[38] http://fermin.perso.math.cnrs.fr/Files/Chap3.pdf (dernier accès 14/09/2020).

[40]https://www.optumlabs.com/search-results.html?q=ehr (denier accès 30/8/2020).

[46] https://www.hebergeurcloud.com/quest-ce-que-apache-spark-la-plate-forme-danalyse-des-megadonnees

[47] https://www.veonum.com/apache-spark-pour-les-nuls/