Année universitaire 2021 - 2022

Institut de Recherche
en Informatique de Toulouse
CNRS - INP - UT3 - UT1 - UT2J
2 Rue du Doyen-Gabriel-Marty 31042 TOULOUSE

UNIVERSITÉ TOULOUSE 1 CAPITOLE
FACULTÉ D'INFORMATIQUE

MASTER MIAGe IPM
L'INFORMATION MAITRISEE

MEMOIRE DE STAGE

Mise en place d'une application pour la création automatique d'entrepôt de données

Stage effectué au sein de l'Institut de recherche en Informatique de Toulouse : IRIT

(du 11/04/2022 au 22/07/2022 et du 22/08/2022 au 31/10/2022)

Prénom : Yuni Nom : CHEN

Mémoire soutenu à l'université le 19/09/2022

Encadrant : Yuzhao YANG Tuteur : Alain BERRO

Titre de stage : Mise en place d'une application pour la création automatique d’entrepôt de données

Nom de l’entreprise : IRIT (Institut de recherche en Informatique de Toulouse)

Équipe : SIG (Systèmes d’Informations Généralisées)

Période du stage : 11/04/2022 – 22/07/2022 et 22/08/2022 - 31/10/2022

Nom stagiaire : Yuni CHEN

Date de soutenance : 19/09/2022

Encadrant : M. Yuzhao YANG et M. Franck RAVAT

Tuteur : M. Alain BERRO

Résumé

L’objectif de ce stage est de concevoir et de développer une application permettant de créer automatiquement un entrepôt à partir des données tabulaires telles que des fichiers csv, de fusionner des entrepôts multiples et de réaliser l’imputation de données. La mission principale porte sur la création automatique d'un entrepôt à partir de données tabulaires. Durant ce stage, j’ai amélioré des algorithmes proposés par mon tuteur entreprise Yuzhao YANG, implémenté toutes d’autres fonctionnalités, réalisé la conception de l’application et le développement de la première version de l’application, le traitement de divers jeux de données et les expérimentations de différentes fonctionnalités. Le stage est composé d'une équipe de quatre personnes, dont une équipe de développement de deux personnes reposant sur la méthode du développement agile.

Dans ce stage, j'ai pu mettre en œuvre certains acquis du master : gestion de projet, modélisation de la base de données et développement d'applications. Ce stage m’a permis de mieux comprendre le travail des développeurs. J'ai fait de grands progrès dans la coopération avec les collèges et la programmation. J'ai également acquis des connaissances pertinentes en gestion des données, en apprentissage automatique, en traitement du langage naturel et en Node.js. Je pense que ces nouvelles connaissances seront d'une grande aide pour mon futur.

Remerciements

Tout d'abord, je tiens à remercier l'équipe de SIG d'IRIT et Yuzhao YANG de m'avoir offert cette opportunité de stage. Ils m'ont offert un stage enrichissant et j'en ai pleinement profité.

Je tiens à remercier toutes les personnes impliquées dans ce projet, Franck RAVAT, Yuzhao YANG et Haoyang YU, qui m'ont apporté un soutien technique, professionnel et personnel.

Je tiens à remercier Franck RAVAT, Yuzhao YANG et mon tuteur stagiaire Alain BERRO, pour ses conseils, explications, suggestions et aide à la rédaction du rapport durant mon stage.

Je tiens à remercier toute l'équipe pédagogique de la formation Master MIAGE-IPM pour apporter des connaissances variées qui m'ont été très utiles durant mon stage.

Je tiens à remercier toute l'équipe ANR BI4PEOPLE[1] de me fournir le projet de stage et de me donner ses conseils techniques.

Enfin, je tiens également à remercier tous les membres du bureau des doctorants pour la bonne ambiance de travail et l'environnement de travail plein d'humour qu'ils ont créé.

[1] https://anr.fr/Project-ANR-19-CE23-0005

Sommaire

Table des matières

Liste des figures

Liste des tables

Glossaire

Business intelligence : Processus qui utilise des stratégies et des technologies tel que l'entrepôt de données, le traitement des données, l'exploration de données et la visualisation des données pour l'analyse des données et la gestion des informations métiers afin d'extraire une valeur métier.

Entrepôt de données : Espace de stockage pour collecter, ordonner, journaliser et stocker des données provenant de base de données opérationnelles et fournir ainsi un socle à l'aide à la décision en entreprise.

Dépendance fonctionnelle : Une dépendance fonctionnelle (FD) est une relation entre deux attributs, généralement entre la primaire clé (PK) et d'autres attributs non-clés dans une table. Pour toute relation R, l'attribut Y dépend fonctionnellement de l'attribut X (généralement le PK), si pour chaque instance valide de X, cette valeur de X détermine de manière unique la valeur de Y. Ex : si X → Y, alors X détermine Y, on dit que X → Y est une dépendance fonctionnelle.

Dépendance fonctionnelle élémentaire : Pour une déterminée C, s'il existe plusieurs déterminants A, B et A est aussi le déterminant de B, donc B détermine C, c'est une dépendance fonctionnelle élémentaire. Ex : A → C, B → C et A → B, mais pas B → A, comme il existe la relation A → B → C, B est le déterminant le plus proche de C, alors on dit que B → C, est une dépendance fonctionnelle élémentaire.

Dépendance transitive : Comme l'exemple de **dépendance fonctionnelle élémentaire**, C'est déterminé par plusieurs déterminants A et B, et B est déterminé par A, mais A n'est déterminée par B. Il existe la relation A → B → C, pour les dépendances fonctionnelles dont les déterminants ne sont pas de déterminant le plus proche du dernier déterminé et dont les déterminés sont le dernier déterminé, tel que A → C, elles sont des dépendances transitives.

Attribut équivalent : Si l'attribut A détermine l'attribut B et B détermine aussi A (A → B et B → A), donc A et B sont les attributs équivalents. On dit que A est l'attribut équivalent de B et B est aussi l'attribut équivalent de A.

Mesure : L'indicateur mesurant l'activité d'une organisation, des types de calcul à appliquer sur les données en général, elle est en numérique et en additive (fonctions d'agrégation). Par exemple, le montant pour une commande.

Dimension : une structure qui catégorise les faits et les mesures afin de permettre aux utilisateurs de répondre aux questions métiers. Elle est un axe d'analyse pour observer les valeurs des mesures du fait. Au moins un attribut et au moins une hiérarchie.

Hiérarchie : un modèle de données dans lequel les données sont stockées sous forme d'enregistrements et organisées dans une structure arborescente, ou structure parent-enfant. Dans la structure parent-enfant, un nœud parent peut avoir plusieurs nœuds enfants connectés par des liens. Dans ce modèle de données, les uns enregistrements relient aux autres par des liens. Un enregistrement est une collection de champs, chaque champ ne contenant qu'une seule valeur. Le type d'un enregistrement définit les champs que l'enregistrement contient.

Paramètre : un attribut identifiant sans ambiguïté un niveau de granularité d'analyse

Attribut faible : association d'attributs faibles aux paramètres (attributs complétant la sémantique d'un paramètre et dont la valeur dépend de la valeur du paramètre associé

Spectral Clustering : une technique qui trouve ses racines dans la théorie des graphes. Il considère que chaque échantillon de l'ensemble de données doit être regroupé comme un point dans l'espace, ces points sont connectés ensemble et les arêtes connectées ont des poids, et la taille des poids indique la similitude entre ces échantillons. Cela forme la matrice de similarité pour cet ensemble de données, puis calcule les valeurs propres et les vecteurs propres de la matrice de corrélation, et sélectionne les vecteurs propres appropriés pour regrouper les différents points de données. La méthode est flexible et nous permet également de regrouper des données non graphiques.

1. Introduction

Mon stage s'est effectué du 11/04/2022 au 22/07/2022 et du 22/08/2022 au 31/10/2022 au sein de l'équipe Systèmes d'Informations Généralisés (SIG) qui est une équipe de recherche du laboratoire Institut de Recherche en Informatique de Toulouse (IRIT).

De nombreuses petites entreprises utilisent encore des fichiers pour stocker des données et analyser des données. Avec le développement des métiers et l'augmentation massive des données, le stockage des données dans l'entrepôt des données devient important pour la Business Intelligence.

L'Entrepôt de Données (ED) est le concept de la Business Intelligence qui a été proposé par Bill Inmon, en 1990. Il est utilisé pour organiser et analyser une grande quantité de données accumulées par les entreprises. Un ED facilite les analyses avec diverses méthodes telles que le traitement analytique en ligne (OLAP) et l'exploration de données (Data Mining), pour aider les décideurs à analyser rapidement et efficacement des informations précieuses à partir d'une grande quantité de données, et à prendre des décisions et à réagir rapidement aux changements de l'environnement externe. Par conséquent, un entrepôt de données, qui est une base de données analytique, est utilisé pour stocker et traiter des données afin d'effectuer une analyse décisionnelle pour une entreprise.

Cependant, certaines entreprises sont confrontées aux problèmes suivants :
1. La conception et la mise en œuvre de l'entrepôt de données, sont des activités importantes dans le processus métier de création d'entrepôt de données à partir des données tabulaires. En général, le processus s'effectué par des techniciens professionnels.
2. De nombreuses petites entreprises ou organisations n'ont pas assez de budget pour embaucher des techniciens professionnels
3. Les personnes qui veulent faire de l'analyse de données, elles n'ont pas d'expertise suffisante ou aucune expertise.

Concernant ce problème, le Doctorant Yuzhao YANG a proposé une solution de la génération automatiquement d'entrepôts de données à partir de données sources stockées sous forme tabulaire. Cette solution va aider les petites entreprises ou les individus manquant de connaissances professionnelles dans ce domaine, à construire plus facilement des entrepôts de données. Le but de mon stage est d'aider le doctorant Yuzhao YANG à réaliser cette solution.

2. Contexte du stage

2.1 Présentation de l'Entreprise

2.1.1 IRIT : Institut de Recherche en Informatique de Toulouse

L'IRIT (Institut de Recherche en Informatique de Toulouse) est une unité mixte de recherche française créée en 1990. L'IRIT qui est l'un des piliers de la recherche en Occitanie. Il est également une des forces structurantes dans le domaine du paysage informatique dans le monde numérique au niveau régional et national.

Actuellement, ses 600 membres sont répartis dans ses vingt-quatre équipes qui constituent les sept départements de L'IRIT (cf. figure 1).

Figure 1 Structure organisationnelle d'IRIT

Les recherches de L'IRIT se structurent en six domaines d'application stratégiques :

1. Santé, Autonomie, Bien-être
2. Aéronautique, Espace, Transports
3. Médias sociaux numériques et diffusion de l'information
4. E-Éducation
5. Cybersécurité, Sécurité des biens et des personnes
6. Ville intelligente

et cinq grands enjeux scientifiques :

1. Transformation des données brutes en informations intelligibles
2. Modélisation numérique du monde réel
3. Conception et construction de systèmes
4. Étude des systèmes autonomes qui s'adaptent à l'environnement
5. Concept de cognition et d'interaction[2]

[2] https://www.irit.fr/le-laboratoire/presentation-du-laboratoire/#

2.1.2 SIG : Système d'Informations Généralisées

Comme mon tuteur entreprise est un membre de l'équipe SIG (Systèmes d'Informations Généralisés) qui m'accueille, une des trois équipes du département GD (Gestion de données) d'IRIT, je deviens la stagiaire de son équipe SIG.

L'équipe SIG est une des équipes importantes d'IRIT avec 21 enseignants et chercheurs des universités de la région Occitanie.

La donnée, qui est au cœur des systèmes d'information modernes, est le domaine de recherche de l'équipe SIG, incluant tout le processus de traitement des données, depuis les données brutes jusqu'aux données traitées, afin que les utilisateurs puissent interroger plus efficacement les informations dont ils ont besoin et les utiliser plus facilement pour l'analyse, la visualisation des données et la prise de décision.[3]

2.1.3 Équipe du Projet de Stage

Notre équipe projet de stage se compose de 4 personnes, deux encadrants (Franck RAVAT et Yuzhao YANG) et deux stagiaires (Moi et Haoyang YU).

Monsieur Franck RAVAT qui est professeur à l'université Toulouse 1 Capitole, est mon premier encadrant. Dans notre équipe, il donne son avis sur l'amélioration du modèle de thèse de Yuzhao YANG, détermine la direction des recherches de l'équipe et suit l'avancement de l'ensemble de notre projet.

Le doctorant Yuzhao YANG est mon tuteur entreprise. Durant mon stage, toutes les tâches se déroulent autour de ses besoins. Il est considéré comme Product Owner pendant mon développement.

Haoyang YU et moi, stagiaires, sommes responsables du développement de l'application dans l'équipe.

2.1.4 Projet ANR BI4PEOPLE

BI4PEOPLE (Business intelligence for the people) est un projet financé par l'ANR (Agence National de Recherche) qui vise à mettre en œuvre une solution permettant d'automatiser les processus de Business intelligence (BI). Ce projet regroupe quatre laboratoires académiques (IRIT, ERIC, ELICO et LIFAT) et une entreprise (Trimane).

Mon stage concerne une partie du projet relative à l'intégration automatique de données dans les entrepôts de données, à savoir le sujet de la thèse de Yuzhao YANG. Plus concrètement, il consiste à créer automatiquement des entrepôts de données à partir des données tabulaires hétérogènes pour faire des analyses en OLAP, conformément à la solution proposée dans la thèse de Yuzhao YANG.

Cette solution contient les cinq étapes suivantes :
1. Identification et transformation des tables par des algorithmes de Machine

[3] https://www.irit.fr/le-laboratoire/organigramme/

Learning.
2. Détection de mesures par des algorithmes de Machine Learning.
3. Détection de dimensions et de hiérarchies par la découverte des dépendances fonctionnelles.
4. Création du schéma conceptuel et intégration de données.
5. Exécution des étapes 1-4 pour chaque source de données, dans le cas où il y en a plusieurs, il faut proposer une fusion automatique des entrepôts de données par des algorithmes proposés.

2.2 Présentation de la Mission

2.2.1 Contexte

Les données sont devenues de plus en plus importantes dans le monde. L'entrepôt de données est également devenu un support indispensable pour les entreprises pour stocker des données. Il est un outil important pour l'aide des décisions des entreprises. L'établissement d'un entrepôt selon le métier est également une étape importante pour les entreprises. Actuellement, cette partie du travail est gérée par des personnes professionnelles qui ont des connaissances en données ; mais pour les petites entreprises ou organisations qui n'ont de connaissances dans ce domaine, cela peut être difficile pour les raisons suivantes :

1. Le coût de la main-d'œuvre de ces professionnels est élevé, ce qui entraîne une augmentation des coûts de main-d'œuvre
2. Les données requises pour l'activité de la plupart des petites et moyennes entreprises ne sont pas très volumineuses et il n'est pas nécessaire d'effectuer fréquemment cette partie du travail, ce qui entraînera facilement un gaspillage de ressources humaines.
3. Certaines étapes sont répétitives, et le développement répétitif de cette partie peut être réduit en définissant un programme spécifique

En réponse à cette carence, le laboratoire a proposé le projet BI4PEOPLE. Pour le travail de réalisation du stockage efficace et de haute qualité des données provenant de différentes sources dans l'entrepôt de données, Yuzhao YANG a proposé le concept d'implémentation automatique d'entrepôt de données.

2.2.2 Problématique

Selon les étapes de conception de la construction d'un entrepôt de données, nous déterminons d'abord les mesures du schéma multidimensionnel. Ensuite, il faut détecter des dimensions et les hiérarchies correspondant à chaque dimension. Enfin, nous identifions les paramètres et ses attributs faibles correspondants pour chaque hiérarchie.

La problématique générale compte à améliorer les propositions effectuées par M. Yuzhao YANG, à proposer de nouvelles implantations.

Premièrement, pour la génération des hiérarchies d'une dimension, M. Yuzhao YANG a implémenté une première version de son algorithme, mais comment optimiser cet algorithme ?

Deuxièmement, comment implémenter l'idée que M. Yuzhao YANG a proposée dans son article pour vérifier si les attributs au niveau le plus haut sont des paramètres ou des attributs faibles ?

Ensuite, comment déterminer si la colonne dont la valeur est un type numérique est un paramètre ou un attribut faible ?

Enfin, concernant l'imputation de données manquantes, pour convertir des données textuelles en numérique, comment trouve-je un corpus adapté pour entraîner un modèle qui convertit les données textuelles en une matrice de similarité ? Comment utiliser un modèle de traitement du langage naturel dans une bibliothèque pour entraîner ce modèle ? Comment combiner les algorithmes KNN et Spectral Clustering pour regrouper chaque ligne de données et obtenir la ligne de données la plus proche de la ligne qui manque des données ?

2.2.3 Enjeux et Objectifs

L'enjeu de ce stage est de mettre en œuvre une application pour aider les personnes ayant peu de connaissances en la modélisation multidimensionnelle conception et l'implantation des schémas.

L'objectif principal de ce stage est de concevoir et de développer une application comprenant les fonctionnalités : création automatique d'un entrepôt de données à partir des données structurées ou semi-structurées, fusion des entrepôts multiples et l'imputation automatique des données d'un entrepôt.

L'objectif ultime de notre développement est de réaliser un outil pour générer automatiquement un entrepôt à partir d'un fichier csv, fusionner des entrepôts multiples pour les tables multiples, compléter automatiquement des données manquantes si nécessaire et réduire le temps de transformation des données tabulaires en un entrepôt.

2.2.4 Missions

D'après les objectifs de mon stage, mes missions se déroulent en 4 grandes parties :

1. Création automatique d'entrepôt à partir données tabulaires

1) Étudier le contexte du projet et sa solution, comprendre le concept de bases relationnelles, du traitement de données, des modèles de Machine Learning, de modèles de Natural Language Processing, de structure d'application et les connaissances liées au développement d'application.
2) Extraire le code de l'algorithme à partir d'un outil et le modifier, afin d'identifier des dépendances fonctionnelles qui sont un des éléments principaux pour détecter des hiérarchies dans un jeu de données
3) Identifier des mesures d'un jeu de données par sources apprentissage automatique
   D'après l'algorithme proposé par mon tuteur entrepris, Yuzhao YANG :
   - Acquérir les caractéristiques des données des colonnes dont les valeurs sont en numérique, ensuite,

- Entraîner un modèle d'apprentissage automatique avec un jeu de données d'entraînement. Le jeu de données enregistre des caractéristiques générales, statistiques et inter-colonnes des colonnes en numérique de différents jeux de données,
- Prédire si les colonnes sont des mesures du jeu de données à l'aide de modèle de données entraîné.

4) Identifier des hiérarchies basées sur les dépendances fonctionnelles acquises lors de l'étape 2)
5) Créer un entrepôt en fonction des mesures et des hiérarchies déterminées aux étapes 3) et 4)

2. Fusion d'entrepôts multiples

1) S'il existe plusieurs tables dans un jeu de données source, créer un entrepôt pour une table du jeu de données selon l'étape 1
2) Fusionner les entrepôts générés

3. Imputation de données

1) Entraîner un modèle de NLP adapté pour transformer les données textuelles en type numérique, y compris la recherche d'un corpus correspondante à notre besoin
2) Implémenter l'algorithme Spectral Clustering en intégrant le modèle de NLP et les algorithmes KNN, KMeans
3) Calculer la distance au centre des centres des clusters de chaque ligne
4) Faire des clusterings pour les lignes basées sur la matrice de similarité via l'algorithme de Spectral Clustering
5) Compléter des données manquantes d'une ligne en fonction des données de la ligne dont la distance est la plus proche de la précédente dans le même cluster.

4. Conception et développement d'application

Concevoir l'architecture d'application, implémenter les fonctionnalités dans l'application selon des missions ci-dessus, et développer une interface ergonomique pour que les utilisateurs puissent retrouver les jeux de données et les résultats des exécutions des processus. (cf. figure 2)

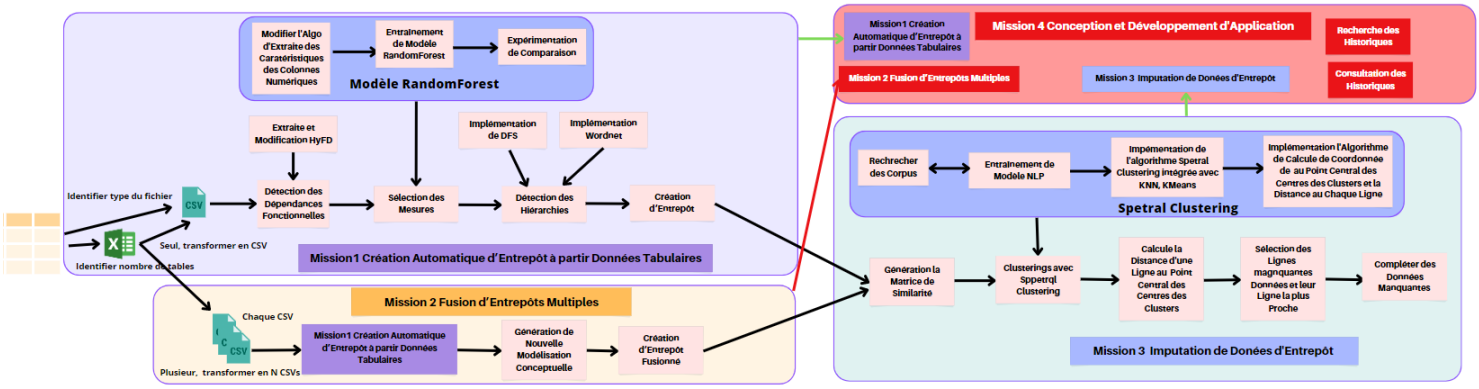


Figure 2 Démarches des missions

3. Gestion de projet

3.1 Méthode de Développement agile

Le développement agile est une approche itérative et incrémentale pour la gestion de projet et le développement d'un logiciel. L'équipe du développement livre itérativement et incrémentalement basée sur la priorité des besoins des utilisateurs. Dans le développement agile, un projet est divisé en plusieurs sous-projets avant développement. Chaque sous-projet est testé et intégré, et il faut aussi être visible et utilisable avant livraison.

Le développement agile permet à l'équipe de développement de livrer en continu des produits, de réduire les risques, d'améliorer constamment la qualité du développement de l'équipe, d'être flexible aux changements des besoins des utilisateurs et d'améliorer la satisfaction client dans un processus de développement itératif.

3.1.1 Développement Scrum

Scrum, il s'agit d'un cadre de développement agile et d'un processus de développement incrémental et itératif. L'équipe de Scrum se compose d'une équipe de développement, le Scrum Master et le Product Owner (PO).

Avec la méthode du développement agile, tous les besoins du client sont considérés comme les User Stories (US). Ils sont fournis par le PO. Tous les US sont listés dans le Product Backlog avec sa priorité donnée par PO. l'équipe du développement livre incrémentalement et itérativement des produits en fonction du Product Backlog. Le développement agile divise la durée du projet en Sprints. Avant le développement, l'équipe de développement et le PO déterminent le cycle de Sprint. L'équipe de développement estime sa vélocité. Après chaque Sprint Planning, l'équipe de développement choisit des User Story de ce Sprint en fonction de sa vélocité et de la priorité des User Story. L'équipe de développement livre les produits développés dans ce Sprint après chaque Sprint, et révise ce Sprint pour améliorer la qualité de développement du Sprint suivant.

3.2 Déroulement du Stage

Durant le développement de mon stage, nous avons utilisé le développement Scrum. Cependant, en raison de la diversité des tâches de stage, je dois non seulement développer une application, mais faire des expérimentations et optimiser le code existant en fonction de l'avancement du projet doctoral. Nous avons personnalisé le développement agile en fonction des besoins de Yuzhao et de l'état d'avancement du projet.

Cette section décrit les principes d'agilité que nous réservons au développement d'applications.

3.2.1 Recueil des Besoins

La collecte des besoins des utilisateurs est une étape essentielle dans le développement d'un projet, en particulier dans le développement agile. Le recueil des besoins utilisateurs consiste à identifier, affiner et hiérarchiser les besoins des utilisateurs par rapport à un sujet précis. Dans un projet de développement, il doit inclure les fonctions à atteindre, les limites du produit et les exigences de qualité à atteindre, la finalité du produit.

L'équipe du développement communique régulièrement avec le PO. Cela permet au PO de participer au développement et de visualiser l'avancement du développement. Tout cela améliore l'expérience et la satisfaction de l'utilisateur. Dans la communication, l'équipe du développement peut découvrir des changements des besoins à temps, afin d'éviter l'incohérence entre le produit réel et les besoins.

Au cours de mon stage, nous avons utilisé la méthode de développement agile pour aider Yuzhao YANG à développer l'application. Il est donc considéré comme le PO dans notre projet de développement. Selon le Product Backlog et la vélocité de Haoyang et moi, nous dictons les cycles de développement pour chaque Sprint.

Généralement, les besoins doivent être collectés au début de chaque Sprint et le produit ou le travail doit être livré à la fin de chaque Sprint. En fonction de la spécificité de notre équipe, nous avons défini le sprint. Après avoir présenté l'avancement du développement du projet lors de chaque réunion de livraison de Sprint, Yuzhao confirme si les fonctions produites dans ce Sprint répondent aux besoins, et propose ses avis pour les fonctionnalités qui doivent encore être améliorées. Nous enregistrons ses avis comme une partie de ses nouveaux besoins. De plus, sauf le premier Sprint dans lequel nous confirmons la priorité des User Stories proposés par Yuzhao, nous collectons ses nouveaux besoins et la priorité des besoins dans la livraison de Sprint. Donc, pour nous, une réunion Sprint est à la fois un Sprint Planning et une livraison. C'est la personnalité de notre Sprint.

3.2.2 Scrum Board et Planning de Travail

1. Scrum Board

Un Scrum Board montre que le travail du projet est réparti dans les différentes étapes. Il permet à tous les membres de l'équipe de visualiser le suivi du projet. Généralement, sur le Scrum Board, il faut montrer l'objectif d'un Sprint. Un Scrum Board présente au Product Owner (PO), les User Stories (US) été développé dans un Sprint, les tâches de chaque US et les colonnes qui présentent les avancements d'une tâche. De plus, une tâche doit être présentée avec l'état de développement tel que bloqué ou modifié, le développeur et son numéro de US.

Donc, le Scrum Board montre visuellement le travail à différentes étapes du processus. Ses colonnes représentent chaque étape du développement. Les cartes de différentes couleurs sur le Scrum Board représentent les tâches avec différentes priorités. Les cartes se déplacent de gauche à droite en fonction de la phase de développement. Cela montre l'avancement du développement et aide les équipes à gérer leur travail.

Figure 3 Scrum Board

Dans un Scrum Board tel que la figure 3 (cf. figure 3), tous les processus du développement se composent de 4 colonnes. La première colonne « Stories » permet de positionnner les cartes des User Stories qui seront développés pendant le Sprint. Les 3 autres colonnes représentent 3 étapes :

- « To Do » montre les tâches de l'US de même ligne à faire,
- « In Progress » présente les tâches en cours de développement et
- « Done » est une colonne pour déposer les tâches terminées.

Un Scrum Board peut fournir une vue plus détaillée de l'avancement du projet en subdivisant la colonne « In Process » en « Develop », « Test » et « Integrate ». Pour visualiser des flux du développement plus détaillés, nous pouvons attacher le symbole du développeur et l'état d'une tâche sur la carte de la tâche.

Durant le développement d'application, pour suivre bien l'avancement du projet, nous utilisons l'application Jira qui est populaire, professionnelle et flexible. Sur Jira, nous pouvons trouver plein de modèles pour le pilotage d'un projet. Pour notre projet, nous choisissons notre type du projet « Software development », ensuite le modèle « Scrum » pour correspondre à notre méthode de développement agile. Nous divisons le Scrum Board en 3 grands états : « To Do », « In Progress » et « Done », et les trois colonnes « Develop », « Test », « Integrate » sont ajoutés comme sous états de « In Progress ». (cf. figure 4)

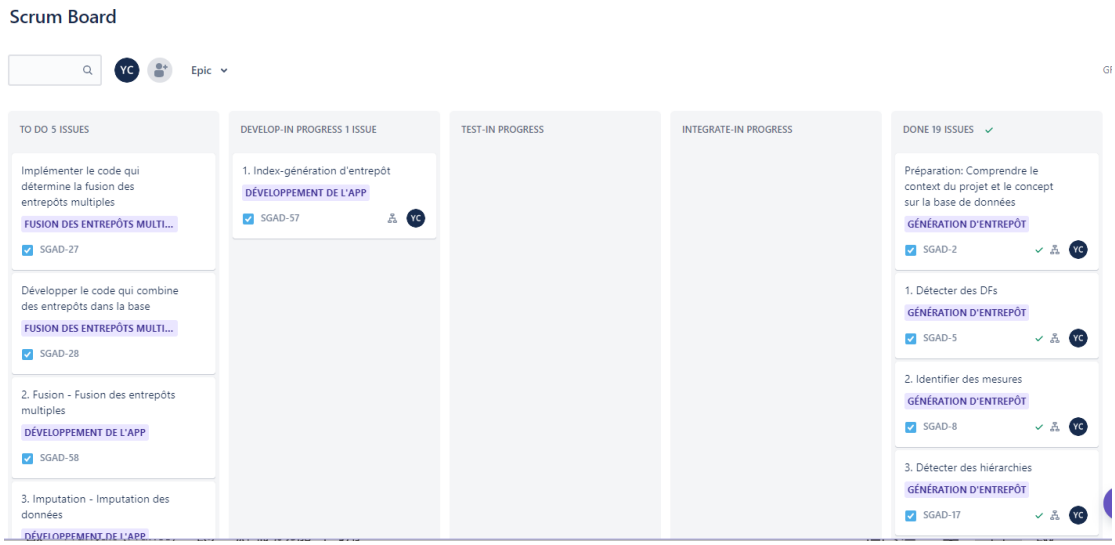


Figure 4 Scrum Board de Jira

Sur Jira, un « Epic » représente un User Story, un « Task » de « Epic » est une tâche de cet User Story. Pour organiser nos besoins, nous ajoutons un User Story en créant

un « Epic » dans le page « Scrum Board ». Ensuite, nous pouvons ajouter un ensemble de « Task » dans la page de « Epic » en cliquant « Add a child issue ». Une fois qu'un « Task » est créé sous le « Epic », il est placé par défaut dans la colonne « To Do » sur la page « Bord ». (cf. figure 5)

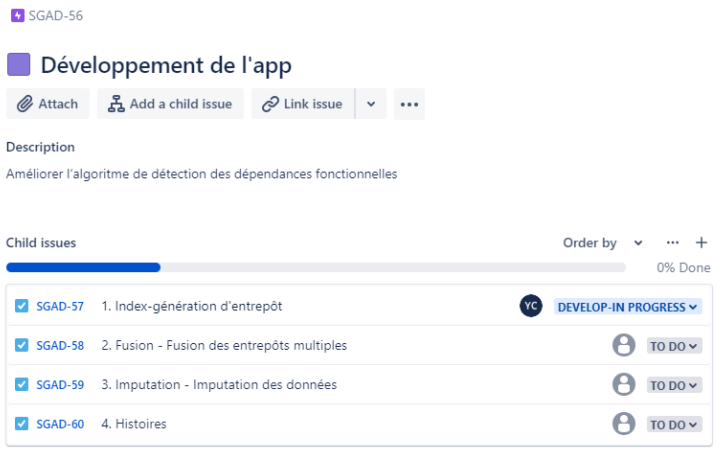
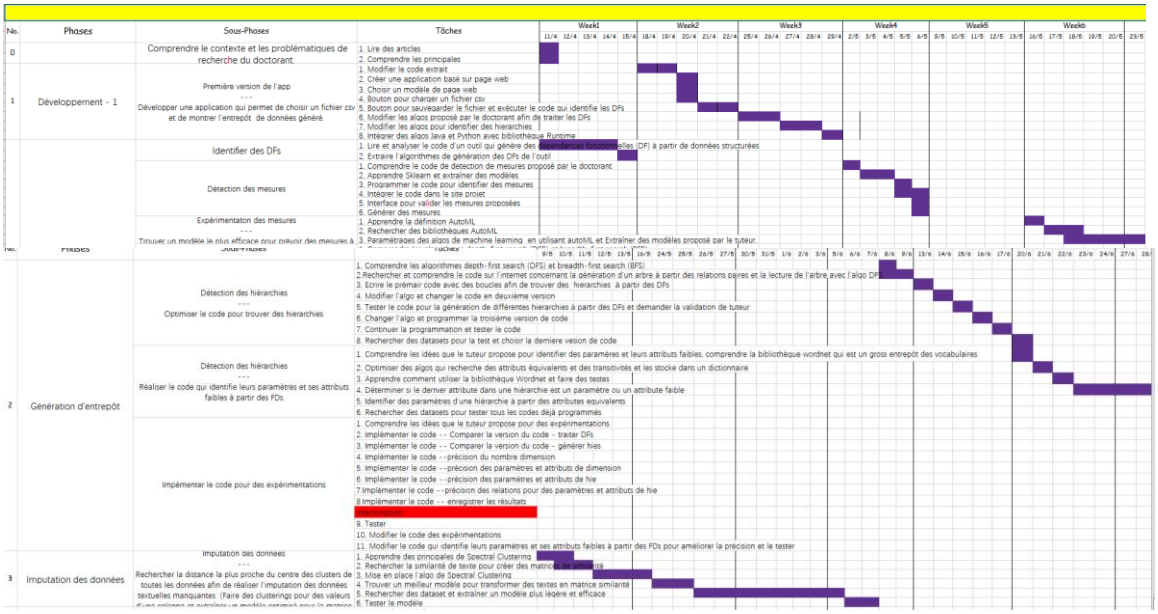


Figure 5 Epic

2. Planning de Travail

Afin de bien suivre en détail toutes mes missions et les étapes que j'ai effectuées de chaque mission, j'ai établi un planning de travail au début de mon stage. Je le mettrais à jour lorsque mon travail de la journée sera terminé.

Afin de bien suivre les démarches des tâches (certaines tâches doivent être lancées après certaines tâches, telles que la détection des dépendances fonctionnelles est faite avant la fusion des entrepôts), le planning est dans l'ordre de la solution proposée par Yuzhao, au lieu de dans l'ordre dans lequel j'ai effectué la tâche. Par exemple, j'ai implémenté l'algorithme « Spectral Clustering » de l'étape 3 avant l'expérimentation de génération d'entrepôt de l'étape 2, mais dans le planning de travail, j'ai mis l'algorithme « Spectral Clustering » dans l'étape 3 qui est lancées après l'étape 2 Génération d'entrepôt (cf. figure 6)



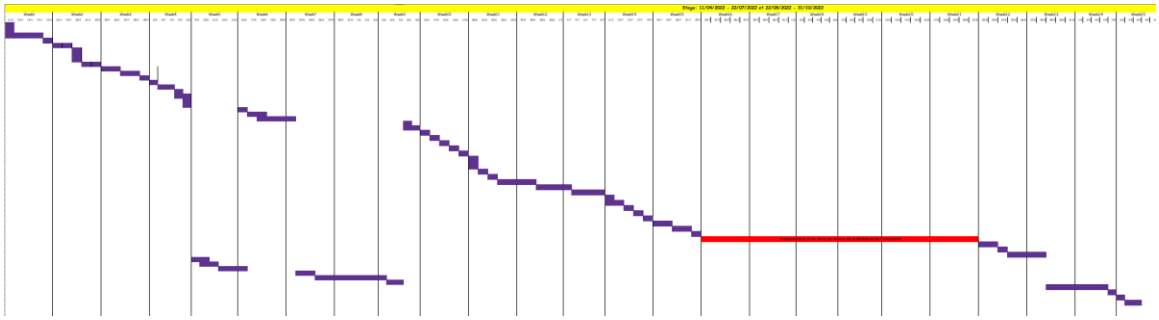


Figure 6 Planning du travail

3.2.3 Les Réunions

Bien que nous utilisions Jira pour suivre l'avancement du développement, nous organisons des réunions presque quotidiennes et hebdomadaires afin que les membres de l'équipe puissent se tenir mutuellement informés de l'avancement de leur tâche.

1. Daily Meeting

Généralement, le Daily Meeting est un rituel organisé tous les matins à une heure fixe. Tous les membres de l'équipe de développement doivent y participer, mais cela n'est pas obligatoire pour le Product Owner. Pendant le Daily Meeting, chaque membre met à jour l'avancement de sa tâche sur Scrum Board et discute avec les autres de mêmes questions suivantes :

(1) Qu'est-ce qu'on a fait hier ?
C'est une question pour suivre l'avancement d'une tâche, il est important pour bien faire la gestion de projet pendant un développement du projet.

(2) Qu'est-ce qu'on va faire aujourd'hui ?
Cette question aide les membres à bien connaître leur propre tâche aujourd'hui. Elle permet à l'équipe de développement de faire la gestion de planification des tâches d'un jour.

(3) Y a-t-il quelque chose qui me freine ?
C'est une bonne façon pour l'équipe de découvrir les problèmes à l'heure. L'équipe peut faire la gestion de risque du projet à l'heure et avoir assez de temps pour rechercher une bonne solution.

Le Daily Meeting est une bonne méthode à l'équipe de développement de suivre l'avancement des tâches, de se coordonner sur les tâches en cours et les difficultés rencontrées chaque jour.

Comme les missions de Haoyang et moi sont indépendantes au début de stage, nous n'avons pas organisé le Daily Meeting ensemble. Généralement, nous avons donc une brève réunion avec Yuzhao séparément presque tous les jours. Pendant le Daily Meeting, nous avons discuté de l'avancement, des problèmes rencontrés et des tâches à faire de ce jour.

Après que nous avons commencé à implémenter l'application ensemble, nous passons généralement de 5 à 10 minutes à faire le Daily Meeting. Yuzhao YANG nous a rejoint parfois comme Product Owner, mais il n'a pas besoin de présenter sa tâche.

Nous avons travaillé dans le même bureau, donc nous avons organisé le Daily Meeting en face à face quand nous étions disponibles. Parfois, Yuzhao a discuté avec nous sur les problèmes par rapport à ses besoins ou des algorithmes qu'il a proposés. Pour des problèmes techniques en commun, HaoYang et moi résoudront ensemble après.

2. Sprint Meeting

Le Sprint Meeting est un rituel informel pour l'équipe de développement de présenter ce qu'elle a produit pendant le sprint et le livrer au Product Owner, sans slides. Toute l'équipe comprenant Product Owner doit participer à ce Sprint Meeting.

Nous avons organisé le Sprint Meeting avec Yuzhao quand nous avons fini nos tâches. Pendant le Sprint Meeting, nous avons présenté l'avancement de développement et démontrons les User Stories produits de ce sprint, les difficultés rencontrées et les problèmes à résoudre. Nous avons confirmé aussi des User Stories (des travaux) à faire dans la semaine prochaine. Si Yuzhao a changé ses besoins ou propose ses nouveaux besoins, nous les avons modifiés User Stories dans notre Product Blog. M. Yuzhao validait les fonctionnalités produites s'il les acceptait, sinon, il proposait son avis de l'amélioration et nous avons ajouté son avis dans notre Product Blog (créer un nouvel Epic » sur Jira).

3.2.4 Programmation en Binôme

La programmation en binôme (Pair Programming) est une façon pour les développeurs d'organiser la programmation. Deux développeurs s'installent sur le même ordinateur pour travailler sur la même tâche tel que la même conception, algorithme, code ou test. Ils ont leur propre rôle. Le développeur qui est chargé de développer est le pilote de la programmation en binôme. L'autre est le copilote. Le copilote suit le pilote avec attention, lit le code écrit par le pilote pour détecter les erreurs du code et les défauts éventuels, et prend la note sur les problèmes à résoudre, de tests unitaires à prédire, etc. Les développeurs changent fréquemment leur rôle, généralement après un sprint du projet.

Les membres en binôme partagent les connaissances sur le code et le métier, livrent rapidement la tâche et trouvent plus facilement une bonne solution. Ils réduisent donc la dette technique. Par conséquent, la programmation en binôme permet l'équipe de développement de produire plus rapidement le code de meilleure qualité, d'améliorer les compétences non seulement individuelles, mais aussi collectives sur le code, l'architecture et le métier. L'équipe de développement est plus motivée.

En général, avant le développement de l'application, Haoyang et moi travaillions séparément, car nos tâches étaient indépendantes. Pendant le développement de l'application, Haoyang et moi travaillons en binôme lorsque nous devons modifier l'architecture de l'application, la structure du code et les mêmes parties du code. Une personne est chargée de corriger le code et l'autre personne est chargée de suivre des corrections et de détecter les erreurs dans le code, comme les noms de variables et de méthodes. Parfois, nous travaillons ensemble pour trouver des solutions aux problèmes. De cette façon, nous pouvons facilement éviter les erreurs éventuelles et réduire le temps nécessaire pour les résolutions des problèmes.

3.2.5 Intégration Continue

L'intégration continue est une pratique de DevOps dans laquelle tous les développeurs d'une équipe fusionnent régulièrement (peut-être plusieurs fois par jour) tout le code dans un référentiel central pendant le développement logiciel. À chaque intégration, le système vérifie automatiquement le code afin de détecter les erreurs d'intégration au plus tôt.[4]

Au cours de notre développement, nous utilisions Visual Studio Code pour écrire notre code et l'outil Github pour fusionner notre code. En général, nous téléchargions notre travail de la journée sur la branche « main » de Github avant la fin de chaque journée. Si nous avions un grand changement du code, nous fusionnons le code plusieurs fois dans la journée, en fonction de la taille de notre travail.

3.2.6 Outil

Pendant tout le long de notre projet, nous utilisons plusieurs outils pour la programmation, la revue de code et l'intégration continue.

1. **Pycharm**

   Pendant le développement d'application, Yuzhao a déjà la première version de certains algorithmes en python. De plus, python est un langage de programmation très puissant dans le traitement des données et l'apprentissage automatique. Alors, nous choisissons Pycharm, l'environnement de développement intégré (IDE) dédié à Python, comme notre outil de développement. C'est parce que PyCharm offre une multitude d'outils pour les développeurs Python. Ces outils sont étroitement intégrés afin de créer un environnement pratique pour un développement rapide en Python, web et science des données.[5]

   Nous utilisions Pycharm pour implémenter les algorithmes sur l'apprentissage automatique et le traitement des données tabulaires.

2. **IntelliJ IDEA**

   IntelliJ IDEA est un outil logiciel d'environnement de développement intégré (IDE) Java. C'est également un JVM IDE puissant et convivial pour le développement en Java. Nous pouvons profiter d'un développement Java efficace, car chaque aspect d'IntelliJ IDEA est conçu pour maximiser la productivité des développeurs. La combinaison d'un support de codage intelligent et d'une conception ergonomique rend le développement non seulement efficace, mais aussi agréable. Lorsque IntelliJ IDEA indexe le code source, il fournit des suggestions pertinentes pour diverses situations du code, ce qui permet aux développeurs de compléter le code rapidement et intelligemment et de l'analyser en temps réel.[6]

   J'utilise IntteliJ IDEA, car il apporte l'efficacité et l'intelligence aux développeurs. Cela rend le développement très fluide.

[4] https://fr.wikipedia.org/wiki/Int%C3%A9gration_continue

[5] https://www.jetbrains.com/help/pycharm/quick-start-guide.html

[6] https://www.jetbrains.com/fr-fr/idea/

Au cours de mon stage, j'ai extrait le code de l'algorithme HyFD à partir d'un projet de java web, le modifié et développé la première version de l'application en java via IntteliJ IDEA.

3. Visual Studio Code

Visual Studio Code (VS Code) est un éditeur de code source multiplateforme gratuit développé par Microsoft. Le logiciel prend en charge la coloration syntaxique, la complétion de code (alias IntelliSense) et la refactorisation du code. Il dispose d'outils de ligne de commande intégrés et d'un système de contrôle de version Git. Les utilisateurs peuvent installer des extensions pour étendre les fonctionnalités de VS Code via la boutique d'extensions intégrée. Visual Studio Code supporte de nombreux langages de programmation par défaut, notamment JavaScript, TypeScript, CSS et HTML. Il est un excellent outil de développement pour les langages de développement autres que Java et Python.[7]

Après que Yuzhao et nous ayons organisé une réunion avec l'équipe BI4PEOPLE, nous avons su que certains membres ont développé leurs applications en Node.js. Alors, nous avons commencé notre développement d'application en Node.js afin d'aligner sur leur développement. Pour programmer plus efficacement, nous avons choisi Visual Studio Code à implémenter la deuxième version de notre application.

4. SonarCloud

SonarCloud est un service d'analyse de code basé sur le cloud. Il utilise une technologie de pointe d'analyse statique du code pour détecter les problèmes et les problèmes potentiels dans le code de 25 langages de programmation différents. De cette façon, il garantit en permanence la maintenabilité, la fiabilité et la sécurité du code. En conséquence, SonarCloud aide les développeurs à identifier les problèmes à un stade précoce, à réduire les problèmes de code plus tard dans le développement et à améliorer la qualité globale du code de production.[8]

Avec SonarCloud, nous repérons facilement les problèmes dans notre code, et réduisons la dette technique produite durant notre programmation.

5. Github

Github est un outil populaire pour le développement collaboratif et l'intégration continue. Il est une plate-forme de services d'hébergement Web et de gestion de développement de logiciels pour le code source des logiciels. Il utilise Git comme logiciel de contrôle de version.[9]

Nous l'avons utilisé pour l'intégration continue et l'hébergement du code au cours de notre développement.

[7] https://en.wikipedia.org/wiki/Visual_Studio_Code
[8] https://docs.sonarcloud.io/
[9] https://en.wikipedia.org/wiki/GitHub

4. Conception

4.1 Modélisation Conceptuelle de l'Application

Notre application se compose de cinq fonctionnalités. En plus des trois grandes fonctionnalités décrites dans la section 2.2.4 Missions, nous avons ajouté deux fonctionnalités relatives à la consultation facile des résultats.

1. Recherche des résultats

La fonctionnalité permet aux utilisateurs de rechercher un résultat en saisissant un mot clé.

2. Création automatique d'entrepôt à partir des données tubulaires

La fonctionnalité se déroule selon les étapes suivantes :

2.1. Identifier le type du fichier téléchargé, transformer le fichier si son format n'est pas en csv, et stocker le fichier

2.2. Générer automatiquement des schémas multidimensionnels en 3 étapes :

- Identifier des dépendances fonctionnelles à partir des jeux de données, et les traiter
- Identifier des mesures à partir des jeux de données et demander la validation des utilisateurs
- Supprimer des dépendances fonctionnelles ayant les attributs des mesures validées, identifier des hiérarchies à partir des dépendances fonctionnelles spécifiées précédemment, et demander la validation des utilisateurs

2.3. Créer un entrepôt selon le schéma multidimensionnel après la validation des utilisateurs

2.4. Retourner à l'utilisateur le résultat final (Schéma multidimensionnel)

3. Fusion d'entrepôts multiples

3.1. S'il existe un jeu de données avec tables multiples, transformer chaque table en un fichier csv

3.2. Créer un entrepôt pour une table selon la fonctionnalité 2

3.3. Fusionner les entrepôts générés ci-dessus une fois que l'utilisateur demande la fusion

3.4. Retourner à l'utilisateur le résultat de la fusion

4. Imputation de données d'entrepôt

4.1. Générer la matrice de similarité à l'aide des modèles de NLP et de Distance euclidienne.

4.2. Faire des clusterings pour les lignes basées sur la matrice de similarité via l'algorithme de Spectral Clustering

4.3. Calculer la distance au centre des clusters de chaque ligne, puis compléter les données manquantes d'une ligne en fonction des données de la ligne dont la distance est la plus proche dans le même cluster.

4.4. Retourner à l'utilisateur le résultat de l'imputation

5. Consultation des historiques

Tous les résultats sont listés dans la page. Les utilisateurs peuvent consulter les fichiers téléchargés, les entrepôts générés et les données complétées d'un entrepôt.

Les solutions suivantes permettent de présenter les maquettes de pages (conception de l'interface), l'interaction des informations avec les utilisateurs, les fonctionnalités de l'application.

4.1.1 Fonctionnalité de Recherche

C'est la page d'accueil qui permet aux utilisateurs de rechercher un fichier ou un entrepôt selon son nom.

1. Maquette

Sur la page d'accueil, il existe une zone et un bouton. Les utilisateurs peuvent saisir le nom d'un fichier ou d'un entrepôt dans la zone, ensuite cliquer sur le bouton pour soumettre l'information saisie. Une fois que les résultats sont retournés, ils sont affichés sur la page. (cf. figure 7)

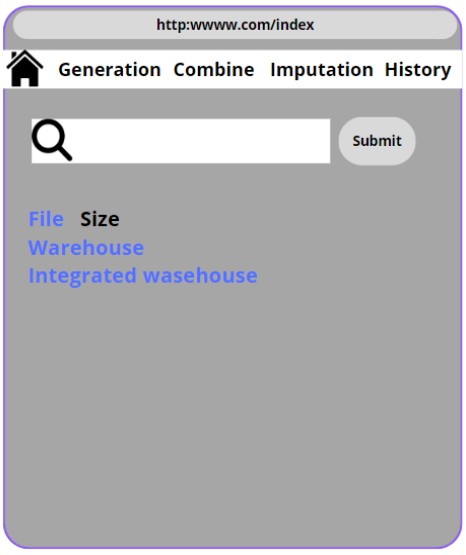


Figure 7 Page Index

2. Visualisation des données sur l'interface

Si le résultat est un fichier, la page affiche :

- *Le nom du fichier* : il est attaché avec un lien qui permet de consulter les informations détaillées du fichier.

- *La taille :* la taille du fichier

Si le résultat est un entrepôt, la page affiche :

- *Le nom d'entrepôt* : c'est le nom d'un entrepôt de données généré par un fichier, il est attaché avec un lien pour consulter ses informations.

Si le résultat est un entrepôt fusionné, la page affiche :

- *Le nom d'entrepôt* : c'est le nom d'un entrepôt fusionné par des entrepôts générés, il est attaché avec un lien pour consulter ses informations.

4.1.2 Création Automatique d'Un Entrepôt

Dans cette partie, nous avons conçu des interfaces principales pour la gestion automatique des données dans les fichiers qui sont téléchargés par les utilisateurs sur notre application.

4.1.2.1 Stockage des Données dans la Base de Données

1. Maquette

Comme ce que la figure 8 montre (cf. figure 8), cette page permet aux utilisateurs de sélectionner un fichier local et de le télécharger. Une fois que les utilisateurs ont soumis leurs fichiers, notre application les aidera à générer automatiquement des mesures en utilisant le modèle NLP et à proposer d'autres mesures possibles. Les mesures sont affichées sur la même page. Au cours de ce processus, les utilisateurs ne sont pas tenus de fournir des informations supplémentaires.

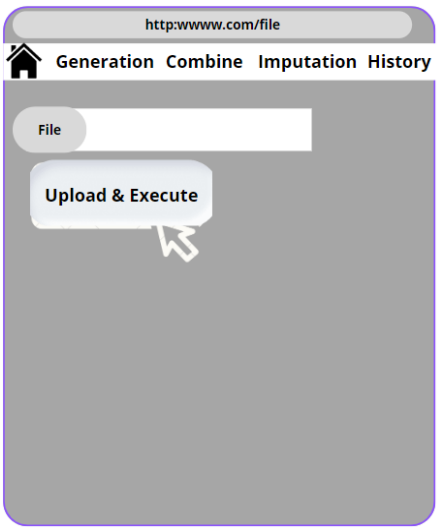


Figure 8 Génération d'entrepôt

2. Visualisation des données sur l'interface

Son résultat est présenté comme la figure 9 Choix-Mesure. (cf. figure 9)

4.1.2.2 Sélection des Mesures

1. Maquette

Cette page affiche les résultats de la soumission des dossiers. Les résultats sont affichés dans deux tableaux. L'utilisateur peut alors sélectionner les mesures qui lui semblent correspondre à son métier ou à ses besoins. Lorsque l'utilisateur finit sa sélection et la confirme, l'application détectera automatiquement les hiérarchies possibles. (cf. figure 9)

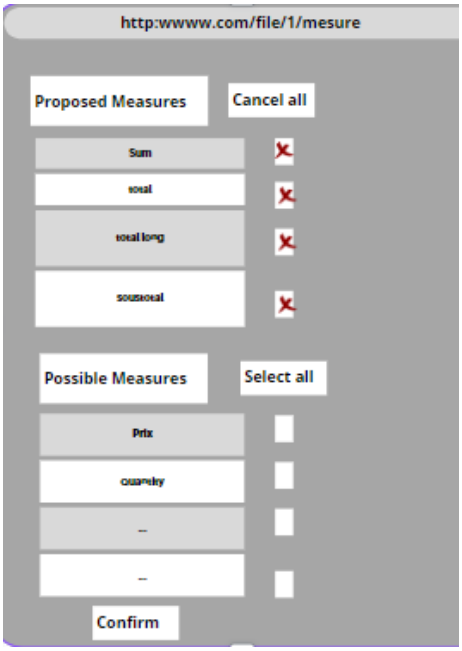


Figure 9 Choix-Mesure

2. Visualisation des données sur l'interface

Le premier tableau montre les mesures prédites par le modèle d'apprentissage automatique. L'autre tableau montre les mesures possibles pour le fichier. (cf. figure 9)

4.1.2.3 Sélection des Hiérarchies

1. Maquette

Les hiérarchies sont affichées sur la même page que les mesures, et l'utilisateur confirme des hiérarchies. S'il a des connaissances métier, il peut sélectionner des hiérarchies en fonction de ses connaissances. (cf. figure 10)

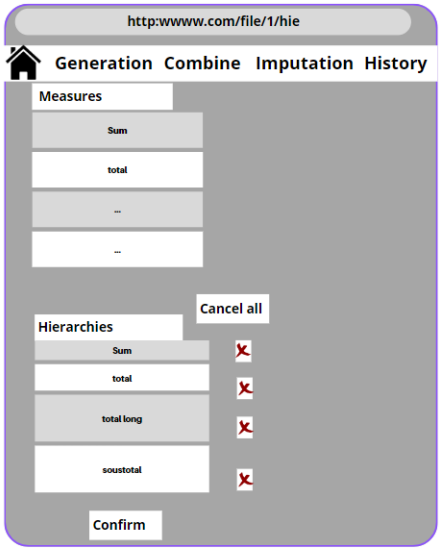


Figure 10 Choix-Hiérarchies

2. Visualisation des données sur l'interface

Le premier tableau stocke les mesures sélectionnées par l'utilisateur. Le deuxième tableau affiche les hiérarchies générées par notre application. (cf. figure 10)

4.1.2.4 Création Finale d'Un Entrepôt

1. Maquette

Après avoir effectué ces actions, l'application fusionne automatiquement les mesures, les dimensions et les hiérarchies confirmées par l'utilisateur. Elle termine ensuite la modélisation de l'entrepôt de données et crée l'entrepôt de données dans Oracle. Enfin, elle retourne à l'utilisateur la réponse. (cf. figure 11)

Les utilisateurs peuvent interroger directement l'entrepôt qu'ils viennent de créer directement sur le site. (cf. figure 11)

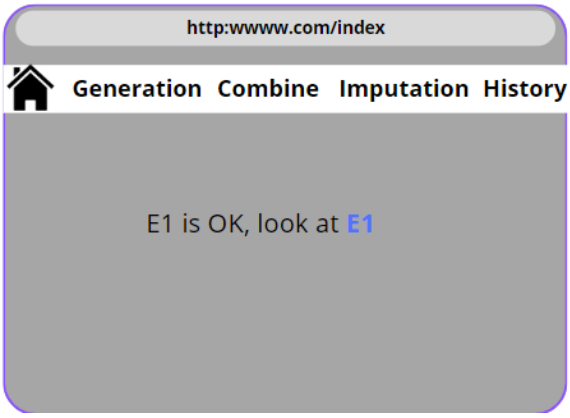


Figure 11 Résultat Création

2. Visualisation des données sur l'interface

La page affiche le résultat de la création de l'entrepôt : le nom de l'entrepôt créé, avec un lien pour consulter ses informations détaillées. (cf. figure 11)

4.1.3 Fusion d'Entrepôts Multiples

La fusion d'entrepôts multiples est une des fonctionnalités principales de notre application. Lorsque des entrepôts sont générés à partir de plusieurs tables contenant des informations communes, ils doivent être fusionnés pour faire des analyses globales. Cette fonctionnalité est donc conçue pour atteindre cet objectif.

1. Maquette

La page permet aux utilisateurs de sélectionner des entrepôts qui sont générés à partir de fichiers contenant plusieurs tables et de les fusionner dans un nouvel entrepôt. (cf. figure 12)

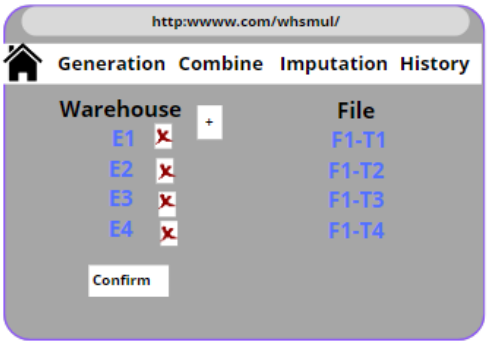


Figure 12 Choix-Entrepôts

2. Visualisation des données sur l'interface

La page suivante affiche le résultat de fusion des entrepôts : le nom de l'entrepôt fusionné avec un lien pour consulter ses informations détaillées. (cf. figure 13)

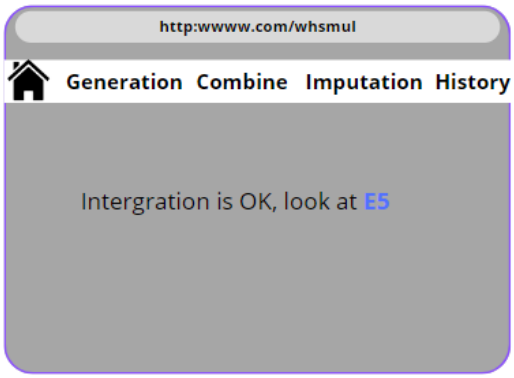


Figure 13 Résultat-Fusion des entrepôts

4.1.4 Imputation de Données

L'objectif de cette fonctionnalité est d'aider les utilisateurs à compléter les données manquantes dans l'entrepôt avec de meilleures valeurs.

4.1.4.1 Sélection d'Un Entrepôt

1. Maquette

Sur cette page, les utilisateurs peuvent sélectionner un entrepôt qui existe dans notre application et remplir des données manquantes dans cet entrepôt. S'il existe des données manquantes, notre application affichera les résultats de l'imputation. Les utilisateurs peuvent également cliquer l'entrepôt pour voir les résultats sur notre site. (cf. figure 14)

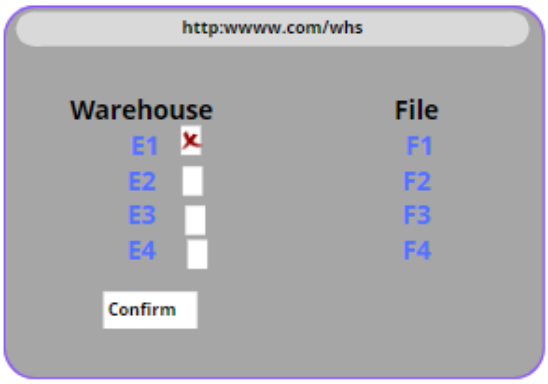


Figure 14 Choix-Entrepôts

2. Visualisation des données sur l'interface

Les résultats de l'imputation sont affichés sur la page suivante : un message de réussite de l'imputation, le nom de l'entrepôt avec un lien permettant de consulter les informations relatives à l'imputation. (cf. figure 15)

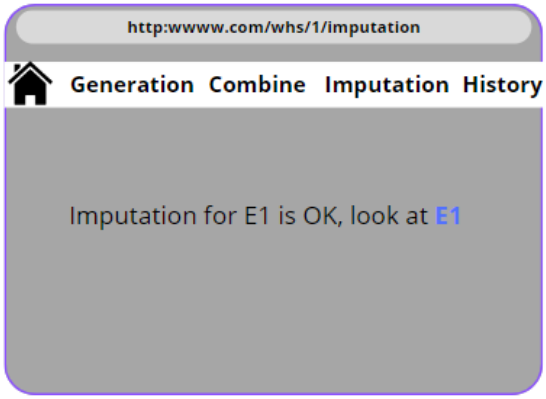


Figure 15 Résultat-Imputation d'un entrepôt

4.1.5 Consultation des Historiques de Génération d'Entrepôt

1. Maquette

Les utilisateurs sont possibles de consulter tous leurs historiques sur la génération d'entrepôt, la fusion des entrepôts multiples et l'imputation de données d'entrepôt. Lorsque l'application génère un entrepôt à partir d'un fichier, le nom du fichier et le nom de l'entrepôt sont stockés dans la base de données. Ils peuvent être consultés sous la rubrique « Génération Entrepôt ». Si des entrepôts sont fusionnés, le nom de nouvel entrepôt est montré sous la rubrique « Fusion Entrepôt ». (cf. figure 16)

Figure 16 historiques

Quel que soit le résultat interrogé par les utilisateurs, notre application peut afficher des informations sur cet historique. (cf. figure 17)

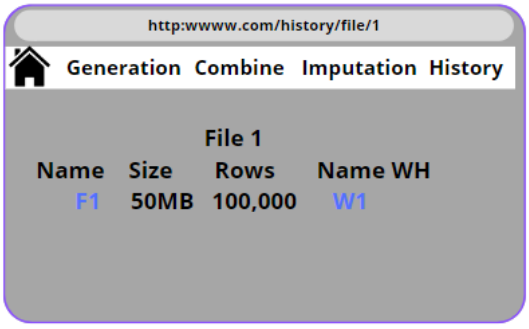


Figure 17 historiques-Fichier

2. Visualisation des données sur l'interface

Si vous consultez un fichier, les résultats affichés sur la page sont les détails du fichier consulté, y compris le nom du fichier avec le lien d'accès, la taille du fichier, le nombre de lignes, et le nom de l'entrepôt généré. Dans le cas d'un entrepôt, le résultat concerne le nom de l'entrepôt avec le lien d'accès et son nombre de lignes. (cf. figure 17)

4.2 Modélisation Conceptuelle des Données

Dans cette partie, je vais présenter les modèles principaux qui montrent les objets interactifs et les paramètres impliqués dans l'application à l'aide de Unified Modeling Language (UML).

UML est un langage de modélisation et de spécification de troisième génération polyvalent et non propriétaire. Il est destiné aux développeurs en phase de conception de logiciels. De plus, il est indépendant des personnes et des langages de programmation spécifiques. Il se dédie à la spécification, la visualisation, la construction et la documentation des logiciels orientés objet en cours de développement. UML est basé sur une combinaison de tous les langages de modélisation de logiciels qui ont existé jusqu'à présent. Il peut être utilisé non seulement pour la modélisation de logiciels, mais aussi pour des travaux de modélisation dans d'autres domaines.

UML se compose de trois modèles principaux :

Modèle fonctionnel :
Il montre la fonctionnalité du système du point de vue d'utilisateur et comprend le diagramme de cas d'utilisation.

Modèle objet :
Il utilise des concepts tels que les objets, les attributs, les opérations et les associations pour montrer la structure et les fondements du système, y compris le diagramme de classe et le diagramme d'objet.

Modèle dynamique :
Il montre des comportements interne du système, incluant le diagramme de séquence, le diagramme d'activité et le diagramme d'état.[10]

Le diagramme de cas d'utilisation et le diagramme de classe sont les deux diagrammes les plus couramment utilisés. J'ai utilisé ces deux diagrammes pendant la conception de notre application.

4.2.1 Diagramme de Cas d'Utilisation

Le diagramme de cas d'utilisation est principalement utilisé pour décrire les rôles et les liens entre les acteurs et les cas d'utilisation, les personnes qui utilisent le système et ce qu'elles peuvent en faire. Le diagramme de cas d'utilisation contient des éléments de modèle, tels que les systèmes, les acteurs et les cas d'utilisation. Il montre de différentes relations entre ces éléments, telles que les généralisations, les associations et les dépendances.[11]

[10] https://en.wikipedia.org/wiki/Unified_Modeling_Language
[11] https://en.wikipedia.org/wiki/Use_case_diagram

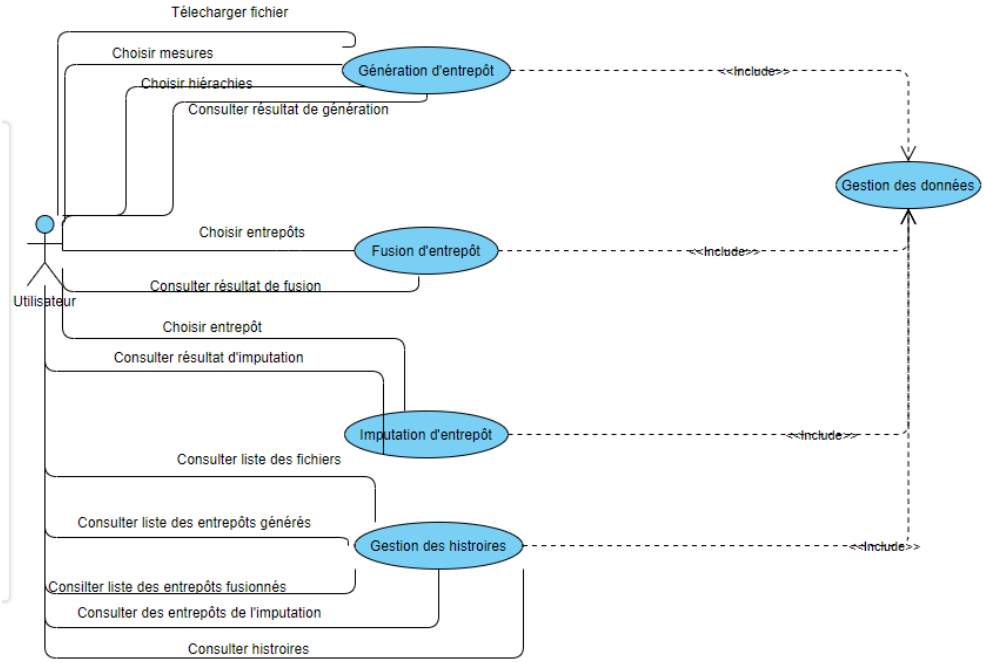


Figure 18 User Case

Selon le diagramme de cas d'utilisation, nous avons quatre packages : génération d'entrepôt, fusion d'entrepôts multiples, imputation de données d'entrepôt et gestion des historiques. De plus, il existe treize activités entre l'utilisateur et notre application. (cf. figure 18)

4.2.2 Diagramme de Classes

Un diagramme de classes est une vue statique qui décrit les classes d'un système et les relations entre elles. Il représente les classes, les interfaces et la collaboration entre elles. Cela nous permet d'avoir une vue complète du système avant de pouvoir écrire correctement le code.[12]

Dans notre cas, nous avons cinq classes : File, Table, Warehouse, Merge et ImputatioinLog.(cf. figure 19)

[12] https://en.wikipedia.org/wiki/Class_diagram

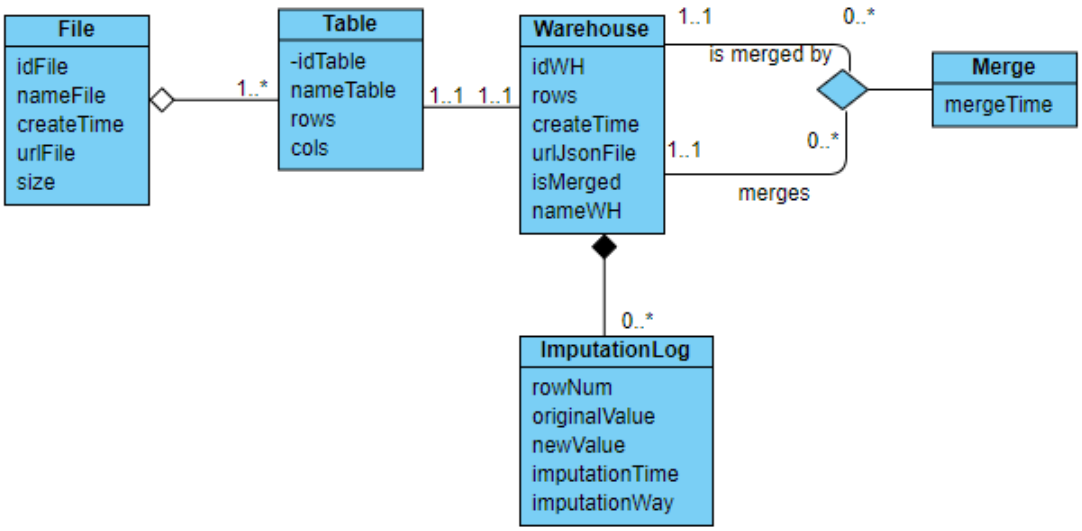


Figure 19 Diagramme de Class

Classe File : Elle comprend les informations du fichier téléchargé par les utilisateurs.

Classe Table : Les informations de chaque table dans le fichier téléchargés sont enregistrées dans la classe Table.

Classe Warehouse : Les informations des entrepôts qui sont générés à partir de fichiers ou fusionnés.

Classe Merge : Elle enregistre principalement le temps de la fusion des entrepôts.

Classe ImputationLog : Cette classe enregistre les informations relatives à l'imputation.

5. Environnement Technique

5.1 Architecture de l'Application

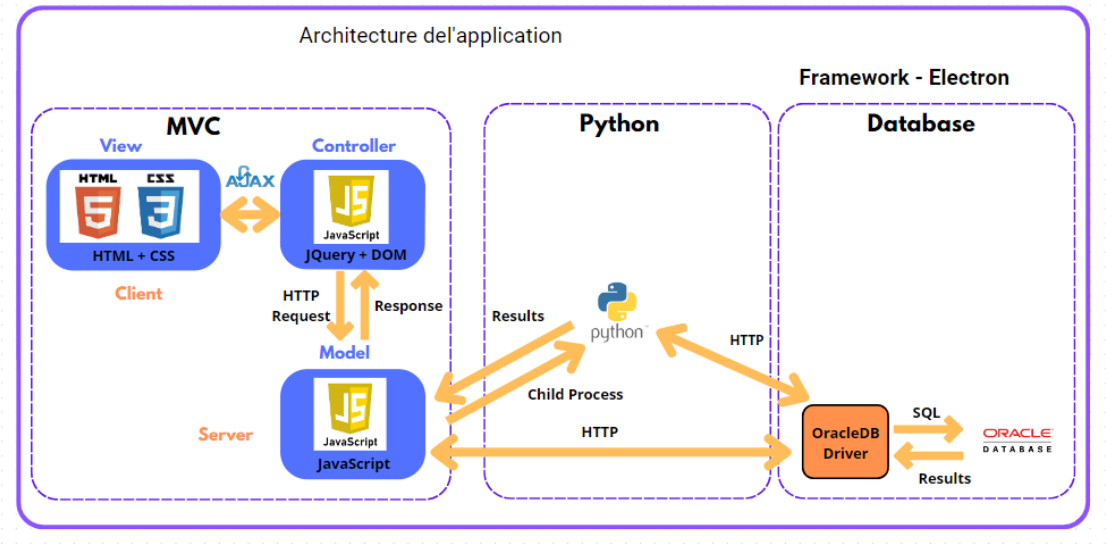


Figure 20 Architecture de l'application

Selon la figure 20 Architecture de l'application, l'architecture se compose de trois parties : MVC, Python et Base de données. (cf. figure 20) :

1. Dans le modèle de MVC, les pages sont écrites en HTML et CSS. Ajax est utilisé pour écouter les comportements des pages et notifier le contrôleur. Le contrôleur est implémenté par Javascript. Il inclut jQuery et DOM. Il envoie les données au serveur via la requête de HTTP. Le serveur utilise ensuite Javascript pour traiter des données acceptées et renvoyer la réponse au contrôleur. La réponse à la demande est récupérée par Ajax dans le contrôleur. Puis, le contrôleur transforme des données de la réponse et les intègre avec la vue via jQuery et DOM. Enfin, il retourne la page entière à l'utilisateur.

2. La deuxième partie concerne les interactions du serveur avec Python, et de Python avec la base de données. Le serveur est implémenté par Node.js. Le module Child Process de Node.js invoque un script Python chaque fois qu'une requête implique des données à traiter par Python. Le serveur obtient le résultat d'exécution et le retourne au contrôleur. Le serveur obtient le résultat de l'exécution et le renvoie au contrôleur. En outre, Python peut se connecter à la base de données pour interroger directement les données.

3. La troisième partie concerne la base de données de l'application, qui stocke des données produites durant les interactions client-serveur.

Les sections suivantes présentent les outils que nous utilisons pour mettre en œuvre notre application.

5.2 Framework Electron

Electron est un Framework de développement côté client basé sur Chromium et Node.js. C'est un open-source qui permet aux développeurs d'établir des applications multiplateformes en utilisant JavaScript, HTML et CSS. Il se compose de trois parties : Chromium, Node.js et Native APIs.

Chromium est une architecture de processus multiple. Elle inclut un processus principal et plusieurs processus de rendu. Node.js supporte les opérations du serveur. Grâce à Native API, les applications développées avec Electron, ont des fonctionnalités natives multi plateformes et de bureau, telles qu'une interface native unifiée et des notifications de message unifiées. La combinaison de ces trois parties rend le développement d'applications plus facile et plus efficace. (cf. figure 21)

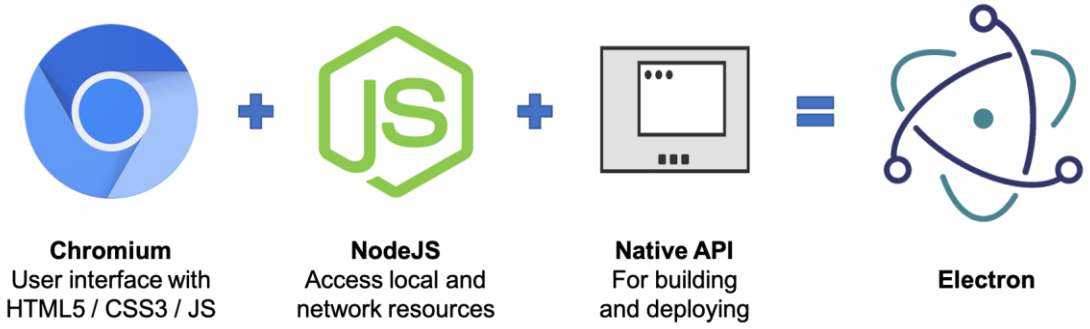


Figure 21 Electron

5.3 MVC Modèle

Modèle-vue-contrôleur (MVC) est un modèle de conception logicielle qui appartient à une architecture en couches. Il divise principalement la logique de l'application en trois couches interdépendantes, afin de séparer le traitement des données et la visualisation d'utilisateur. Chaque couche a ses propres responsabilités et peut communiquer avec la couche suivante. (cf. figure 22)

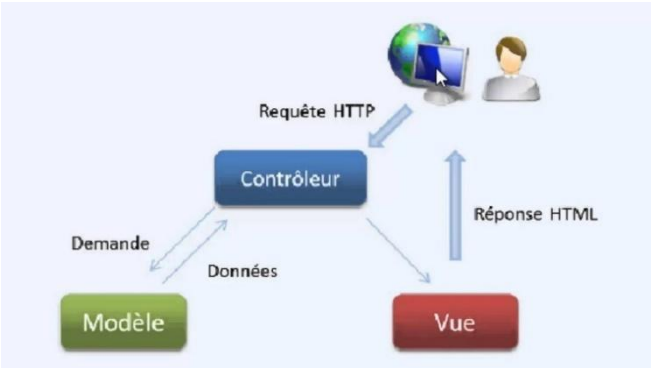


Figure 22 Modèle de Modèle-vue-contrôleur (MVC)

1. **Modèle** est la couche qui est indépendante de l'interface utilisateur. Il encapsule les données de l'application, définit les méthodes de manipulation des données, la

logique et les règles de calcul des données. Il renvoie les données à la vue par l'intermédiaire de la couche contrôleur.

2. **Vue** est l'ensemble des objets d'une application que les utilisateurs peuvent voir. Le but principal de la vue est d'afficher les données du modèle aux utilisateurs et de leur permettre de manipuler ces données. La vue ne peut pas manipuler directement la couche modèle. Elle doit communiquer avec la couche modèle par le contrôleur.

3. **Contrôleur** relie le modèle et la vue. Il peut manipuler les deux couches en acceptant les données et en les transformant du modèle ou de la vue.

5.3.1 Environnement Technique de Vue

Pour implémenter le client de notre application, nous utilisons des langages côté client, tels que HTML, CSS, JavaScript. HTML, CSS et JavaScript sont les langages utilisés pour le développement du page à côté client. Ils permettent aux utilisateurs d'avoir l'interface à communiquer avec l'application.

HTML contrôle la structure des pages Web en défiant les éléments d'interface utilisateur. CSS décide le style de la page qui est l'élément pour la présentation d'une page. Javascript surveille les comportements de la page, tels que cliquer, les activités de souris.

Nous avons utilisé un Framework Bootstrap pour implémenter la couche Vue.

5.3.1.1 Framework Bootstrap

Bootstrap est un open-source et Framework du client basé sur HTML, CSS et JavaScript. Il est principalement utilisé dans la conception et la mise en page de sites réactifs et mobiles. Bootstrap a des fonctions puissantes et riches. Il fournit un ensemble de styles CSS et (éventuellement) de plugins JavaScript pour le design du Web, tels que des barres de navigation, des formulaires, des boutons, des animations et des outils réactifs. Bootstrap permet aux pages Web de s'adapter automatiquement à la taille de l'écran de divers appareils mobiles. Il facilite et accélère le développement Web.[13]

En utilisant Bootstrap, nous avons passé moins de temps à mettre en page notre site, surtout la navigation et la mise en page pour différents appareils.

5.3.2 Environnement Technique de Contrôleur

Au cours du développement de la couche Contrôleur, nous avons utilisé JavaScript et ses techniques jQuery, Ajax et DOM. Cela permet le développement du client d'être plus simple et plus efficace.

[13] https://en.wikipedia.org/wiki/Bootstrap_(front-end_framework)

JQuery est une bibliothèque qui enveloppe JavaScript pour le rendre plus facile à utiliser. Il facilite l'interaction entre la page et l'Ajax.[14]

Ajax (Asynchronous JavaScript And XML) est une technique qui fournit un mécanisme de mise à jour asynchrone. Il est capable de réaliser une mise à jour partielle de la page. C'est-à-dire que les données d'échange entre le client et le serveur peuvent faire partie de la page au lieu de seulement la page entière.[15]

DOM (Document Object Model), il s'agit d'une interface indépendante de la plate-forme. Il est à la fois un langage qui permet aux programmes d'accéder et de mettre à jour dynamiquement le contenu, la structure et le style d'un document. Grâce à lui, JavaScript peut accéder et modifier tous les éléments du document HTML.

5.3.3 Environnement Technique de Modèle

Pendant le développement de la couche Modèle, nous avons utilisé le langage Javascript. Pour cela, Node.js est un environnement nécessaire pour exécuter Javascript.

5.3.3.1 Node.js

Lorsque j'ai commencé mon stage, j'ai d'abord utilisé Java pour développer des pages web. Cependant, lors d'une réunion avec l'équipe de BI4PEOPLE, je me suis rendu compte qu'ils avaient déjà réalisé quelques développements à l'aide de Node.js. Afin de faciliter l'intégration de tous d'autres projets à l'avenir, nous avons également choisi Node.js.

Node.js est un environnement d'exécution JavaScript. Il est un open-source et un multiplateforme basé sur le moteur Chrome V8. La plupart de ses modules sont écrits en JavaScript. Il utilise des techniques étant appliquées à des applications instantanées gourmandes en données. Cela améliore considérablement les performances et la vitesse d'exécution de JavaScript lorsqu'il est exécuté côté serveur, et optimise la taille des données transférées.

L'émergence de Node.js fait de JavaScript devenir un langage côté serveur tel que PHP, Python, Perl. Avant l'avènement de Node.js, JavaScript était un langage de programmation côté client. Il était généralement utilisé uniquement par les développeurs pour développer des pages frontales. En tant que l'environnement d'exécution côté serveur, Node.js fournit un environnement d'exécution backend pour JavaScript. Cela permet à JavaScript de mettre en œuvre la programmation côté serveur. Node.js contient une série de modules intégrés. Donc, les applications développées par JavaScript peuvent être exécutées comme des serveurs indépendants, sans utilisation des serveurs web tels que Apache HTTP Server ou IIS.[16]

Avec les raisons ci-dessus, nous pouvons utiliser JavaScript comme le langage de

[14] https://jquery.com/
[15] https://www.w3schools.com/xml/ajax_intro.asp
[16] https://en.wikipedia.org/wiki/Node.js

développement principal de notre serveur de programme.

5.3.3.2 Art-template

Pour le rendu de page, j'ai trouvé deux solutions : le rendu côté serveur et le rendu côté client. Je préfère le rendu côté client, parce que le module « art-template » est un moteur de template minimaliste et super rapide. Il utilise la technologie de pré-déclaration d'étendue pour optimiser la vitesse de rendu des modèles. Cela permet la performance de l'application d'être proche de la limite de JavaScript. Avec « art-template », les données retournées par le serveur peuvent directement être rendues sur la position indiquée de la page. Donc, il est très utile pour nous d'établir une page dynamique.

5.4 Cube.js

Cube est une plateforme de business intelligence qui agit comme une couche intermédiaire entre les sources de données et les applications. Elle est un open-source pour aider les ingénieurs de données et les développeurs à accéder aux données dans la base de données et à les pousser vers les applications.

Cube.js sont sans-tête, API-first, et découplés avec la visualisation. Premièrement, il peut se connecter, accéder et contrôler toutes les sources de données compatibles avec SQL. Ensuite, il agit comme une couche d'accès aux données. Il peut convertir les requêtes envoyées par les interfaces REST, GraphQL ou SQL en SQL. Cela permet Cube.js de gérer la connexion à la base de données, de consulter la couche de contrôle, et de renvoyer le résultat de l'exécution SQL aux clients. Par conséquent, Cube.js assure la cohérence des données, auxquelles chaque programme accède par des API REST, SQL et GraphQL. Enfin, les développeurs peuvent utiliser une interface de visualisation des données pour les programmes personnalisés écrits en JavaScript.[17]

Comme notre projet de stage est un sous-projet du projet de l'équipe BI4PEOPLE, L'application développée pendant notre stage fait partie de la plateforme qui est développée par BI4PEOPLE. Comme le développement d'autres sous-applications nécessite le framework et les outils fournis par Cube.js, nous avons dû installer Cube.js. Cela facilite l'intégration des différentes applications. Cube.js aide l'équipe BI4PEOPLE à construire une plateforme de visualisation de données rapidement et facilement.

5.5 Python

Python est un langage de programmation populaire. Il a un ensemble de bibliothèques très puissant et très complet pour faciliter le travail dans le traitement des données tabulaires et l'apprentissage automatique.

En effet, avant que je ne commence mon stage, M. Yuzhao avait déjà implémenté

[17] https://cube.dev/docs/introduction

l'algorithme proposé dans ses articles en utilisant python. Après avoir commencé mon stage, j'ai optimisé et intégré le code python qu'il m'a fourni. En outre, la mise en œuvre de l'application implique l'apprentissage automatique et le traitement des données. Par exemple, je dois utiliser l'apprentissage automatique pour entraîner, tester et exporter plusieurs modèles de données, et des bibliothèques python pour traiter les données des fichiers csv.

Par conséquent, Python est un des outils principaux au cours de mon développement. Il est utilisé pour intégrer l'algorithme principal dans les fonctionnalités de traitement des données impliquées dans notre site Web d'application.

5.6 Oracle Database 19c

Oracle Database est un système de gestion de base de données multi-modèles. Elle est aussi une des bases de données relationnelles les plus populaires. La version que nous avons utilisée dans notre projet est Oracle Database 19c. Elle est la dernière version actuelle de support à long terme.

Comme avant mon stage, M. Yuzhao utilisait Oracle pour le stockage des données lors de la mise en œuvre de certains algorithmes. Je continue donc à utiliser son approche.

6. Implémentation de Bases de Données

Au cours de notre développement, nous avons implémenté deux bases de données. L'une est pour stocker des données des entrepôts générés, l'autre est pour stocker des données de l'application.

6.1 Base de Données pour Jeux de Données

L'objectif principal de mon stage est de réaliser la conversion automatique de données tabulaires vers un entrepôt de données. Le type de jeu de données sont donc des données structurées ou semi-structurées, telles que des fichiers xlsx ou csv.

Afin de vérifier l'exactitude des algorithmes et l'efficacité de l'application, j'ai utilisé les cinq jeux de données. (Annex cf. table 10). Avant la génération d'entrepôt, le jeu de données au format non-csv est d'abord converti au format csv. (Annex cf. table 11)

En cours de la génération d'entrepôt, des jeux de données sont stockés dans des entrepôts générés dans Oracle. Généralement, le nom de l'entrepôt de données est le même que celui de jeu de données (cf. table 1). De la manière, le jeu de données peut être utilisé pour réaliser l'imputation de données d'un entrepôt et également autres fonctionnalités du projet BI4PEOPLE.

Nom d'entrepôt	**Nom en CSV**
convidIndicators.db	convidIndicators.csv
devApp. db	devApp.csv
ElectronicsProductsPricingData_V2. db	ElectronicsProductsPricingData_V2.csv
Population2.db	Population2.csv
Sample - Superstore.db	Sample - Superstore.csv

Table 1 Nom de stock - Jeux de données

6.2 Base de Données pour Données de l'Application

Cette partie présente l'implémentation de la base de données pour stocker les données produites durant les interactions entre les utilisateurs et notre application. La structure de la base de données est implémentée selon la modélisation dans la section 4.2.2 Diagramme de classes. (cf. figure 19)

Nous avons 5 tables pour le stock des données.

- **Table File**
  La table est utilisée pour stocker des données détaillées sur les fichiers téléchargés par nos utilisateurs, comme le nom du fichier, sa taille et, surtout, le chemin absolu où les fichiers sont stockés. Grâce à ces données, notre application permet aux utilisateurs de consulter quels fichiers ils ont téléchargés, de quels fichiers proviennent les données des entrepôts générés, etc.

- **Table Table**
  Dans la table, les informations détaillées pour chaque table des fichiers sont enregistrées, telles que le nom de table, le nombre de linges et de colonnes.

- **Table Warehouse**
  La table enregistre les informations des entrepôts générés ou fusionnés. En interrogeant cette table, notre application montre aux utilisateurs sur le site les informations des entrepôts générés, tel que le nom d'entrepôt, la date créée et son fichier associé.

- **Table Merge**
  Dans la table Merge, nous pouvons trouver le temps de la fusion des entrepôts, l'identifiant des entrepôts consolidés et des entrepôts qui sont fusionnés. Ces informations peuvent être interrogées par notre application et mises à la disposition de nos utilisateurs.

- **Table ImputationLog**
  Une fois que l'utilisateur demande l'imputation de données d'un entrepôt, notre application y enregistre les informations de l'imputation. Les informations incluent le temps de l'imputation, l'identifiant de l'entrepôt qui est rempli, le numéro des lignes et les attributs complétés, leur valeur originale et leur nouvelle valeur. Avec ces informations, les utilisateurs peuvent consulter les historiques de l'imputation de données des entrepôts.

7. Implémentation d'Application

7.1 Présentation de l'Application

En suivant la solution de Yuzhao et les missions du stage, l'objet de notre application est de d'aider des petites entreprises et les particuliers ayant peu ou pas de connaissances en entrepôt des données à établir un entrepôt à partir des données tabulaires.

Pour ce faire, elle s'appuie sur trois étapes principales :

1. Créer automatiquement des entrepôts à partir des données tabulaires.
2. Fusionner des entrepôts générés à partir de ces tables s'il existe des tables multiples dans un fichier.
3. Compléter les données manquant dans un entrepôt avec les valeurs de la ligne la plus similaire à la ligne de données manquantes.

En plus des trois fonctionnalités mentionnées ci-dessus, j'ai recommandé d'ajouter des fonctionnalités de recherche et d'interrogation historique, afin d'améliorer la réutilisabilité du programme, la satisfaction et les expériences des utilisateurs, et la commodité de notre application. Notre application comprend donc cinq fonctionnalités principales.

Au cours du développement, nous avons d'abord mis en œuvre les fonctionnalités les plus essentielles, les plus difficiles et les plus complexes via Python. Puis, nous avons implémenté le serveur via JavaScript. Enfin, c'est le développement des pages de l'application.

Actuellement, nous sommes en cours de la mise en place du serveur, nous n'avons pas encore commencé le développement du client. Donc dans cette section, pour la plupart des fonctionnalités, je ne vous présente que ce que nous avons implémenté via Python (cf. figure 30), et ce que nous avons fait pour le serveur (les couches contrôleur et modèle) (cf. figure 35).

7.2 Fonctionnalités de l'Application

Dans cette section, je présente en détail l'implémentation des fonctionnalités de notre application, incluant des techniques, des interfaces, des pages et des opérations.

7.2.1 Recherche des Historiques

Avec cette fonctionnalité, les utilisateurs peuvent rechercher des fichiers téléchargés, des entrepôts générés et des résultats d'imputation de données des entrepôts. Cela aide les utilisateurs à rechercher rapidement l'historique.

1. Bibliothèque et technique

1.1 Interface : /recherche

2. Implémentation de fonctionnalités

A implémenter.

3. Transfert de données

1. Quand les utilisateurs saisissent le nom d'un fichier ou d'un entrepôt sur la page « index », le serveur accepte la requête via l'interface « /recherche ».
2. Ensuite, il se connecte à la base de données pour interroger les informations correspondantes.
3. Puis, le résultat est ensuite renvoyé au contrôleur.
4. Le contrôleur renvoie enfin le résultat au client.

4. Présentation de l'interface

À implémenter.

7.2.2 Génération d'entrepôt

La fonction de la génération d'entrepôt met en œuvre un processus qui commence par un fichier tabulaire téléchargé par l'utilisateur, passe par une série d'activités, et crée finalement un entrepôt de données basé sur la modélisation conceptuelle obtenue à partir des activités. Ces activités incluent la conversion du format de fichier, les étapes de modélisation conceptuelle de l'entrepôt de données.

7.2.2.1 Télécharge des fichiers

C'est la première étape de la génération des entrepôts. Son objectif est de gérer le fichier téléchargé.

1. Bibliothèque et technique

1.1 Interface : /upload

1.2 Technique Node.js :

(1) Module « multer » : Le module tiers de Node.js, il reçoit les fichiers téléchargés et les stocke. Il doit être installé dans le projet Node.js avant l'utilisation.

(2) Module « fs » : Le module propre à Node.js est utilisé pour la gestion des fichiers, comme la lecture, l'écriture, le stockage, le renommage, etc.

2. Implémentation de fonctionnalité

Au cours du développement de cette fonctionnalité, j'ai conçu l'API pour lier la méthode qui accepte et enregistre le fichier téléchargé.

Pour enregistrer le fichier, j'ai implémenté la première méthode. J'ai utilisé principalement les modules « multer » et « fs » de node.js. Le module « multer » est utilisé pour recevoir et enregistrer le fichier téléchargé par les utilisateurs dans un chemin fixe de notre application. C'est pratique pour les autres méthodes d'obtenir des données du fichier à partir du chemin uniforme. Le module « multer » génère

automatiquement un nom aléatoire pour le fichier, le fichier est ensuite stocké sous ce nom. Donc, nous avons besoin du module « fs » pour lire le fichier téléchargé et renommer le fichier à son nom original, lorsque le fichier est enregistré.

De la façon, toutes les méthodes peuvent relire les données du fichier sous son nom original. Cela peut éviter la mauvaise consultation des fichiers. Si le fichier n'est pas au format csv, cette méthode le convertit également au format csv. Si la conversion échoue, la méthode retourne une erreur à l'utilisateur, telle que le fichier est dans un mauvais format, le fichier existe dans notre application, etc.

J'ai ensuite poursuivi le développement de la deuxième méthode. Elle est utilisée pour se connecter à la base de données et enregistrer les informations du fichier dans la table File. Une fois le fichier sauvegardé avec succès, la méthode s'exécute et renvoie le résultat à la première méthode.

Enfin, la première méthode exécute le script Python pour détecter des dépendances fonctionnelles. Elle renvoie le résultat au contrôleur.

3. Transfert de données

Du côté client, l'utilisateur télécharge le fichier. Ici, sa demande est envoyée au côté serveur à l'aide de la méthode POST. Le serveur accepte ensuite la demande de téléchargement de l'utilisateur par le protocole de transport HTTP. Il fait correspondre la demande de l'utilisateur à la méthode associée avec l'interface « /upload » dans la couche modèle.

La méthode enregistre le fichier sur la fiche indiqué et ses informations dans la base de données. Puis, elle appelle le script Python et lui transmet le chemin du fichier. Quand le résultat du script est bien reçu, la méthode le retourne au contrôleur via HTTP Response.

Dans la couche de contrôleur, Ajax prend le résultat et le rend sur la page en utilisant DOM.

4. Présentation de l'interface

L'utilisateur sélectionne d'abord un fichier tabulaire sur son bureau local et clique sur le bouton « Upload & Execute » pour télécharger le fichier. Si le fichier n'est pas de fichier tabulaire ou il existe, l'utilisateur reçoit un message d'erreur sur notre site. Sinon, notre application stock le fichier sur la fiche indiqué, le renomme, et retourne les mesures identifiées à l'utilisateur. (cf. figure 23)

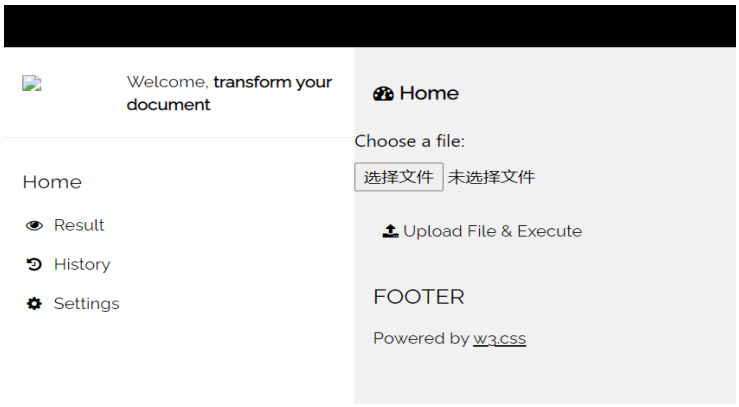


Figure 23 Page Téléchargement

7.2.2.2 Identification des Dépendances Fonctionnelles

C'est l'étape essentielle de la génération d'entrepôt. Les mesures et les hiérarchies du fichier sont des éléments pour établir un entrepôt de données. Pour les détecter, nous devons d'abord identifier des dépendances fonctionnelles du fichier. Donc, l'algorithme qui trouve de bonnes dépendances fonctionnelles est la clé du développement de notre application.

1. Bibliothèque et technique

1.1 Technique Node.js :

(1) Module « child_process » : C'est le module tiers de Node.js. Il peut exécuter des scripts Python éventuellement nécessitant le passage de paramètres.

1.2 Technique Python :

(1) « jpype » : C'est la bibliothèque de Python qui permet Python d'exécuter des méthodes de Java.

2. Implémentation de fonctionnalité

Yuzhao YANG a trouvé le bon algorithme HyFD[11] (cf. figure 32). Selon le résultat du test [1], il obtient les meilleures performances par rapport aux sept algorithmes les plus cités et les plus importants. Comme l'algorithme est placé dans le code d'un outil (cf. figure 31), j'ai dû lire le code et extraire cet algorithme. Pour que cet algorithme réponde aux besoins de notre application, je l'ai également le modifié et ajouté de nouvelles classes et méthodes. Enfin, j'ai empaqueté le code modifié dans un package jar (cf. figure 33). De cette façon, Python peut être invoqué directement. Cela réduit la taille de l'algorithme dans le projet d'application.

Par ailleurs, certaines dépendances fonctionnelles trouvées par l'algorithme HyFD ne sont pas celles dont nous avons besoin. D'après la solution proposée par Yuzhao dans son article [5], les dépendances fonctionnelles transitives et les attributs équivalents doivent d'abord être supprimés. Dans cette partie, j'ai optimisé le code écrit par Yuzhao pour gérer les dépendances fonctionnelles de manière plus précise et plus rapide.

3. Transfert des données

La méthode du serveur (dans Node.js) utilise son module « child_process » pour exécuter le script process.py. Le script cherche ensuite le fichier téléchargé par l'utilisateur en regardant le chemin de stockage absolu enregistré dans notre application. Ensuite, le script exécute le jar HyFD via la bibliothèque « jpype » de Python pour identifier des dépendances fonctionnelles. Après avoir récupéré toutes les dépendances fonctionnelles, process.py supprime les dépendances fonctionnelles transitives et les attributs équivalents, et les stocke.

Après avoir traité les dépendances fonctionnelles, l'application effectue automatiquement la détection des mesures. Ici, le script *process.py* transmet le chemin absolu du fichier et son résultat à l'autre script Python *predictmeasure.py*.

4. Présentation de l'interface

Ceci appartient au développement de la fonctionnalité du serveur, et il n'y a pas d'interface.

7.2.2.3 Détection des Mesures

La méthode est utilisée pour mettre en œuvre la première étape de la modélisation conceptuelle d'entrepôt. Elle exécute le script Python *predictmeasure.py*. Ce script identifie des mesures via notre modèle d'apprentissage automatique RandomForest entraîné, et renvoie les mesures à confirmer par l'utilisateur.

1. Bibliothèque et technique

1.1 Interface : /index

1.2 Technique Python :
(1) Bibliothèque « sklearn » : C'est une grande bibliothèque pour l'apprentissage automatique. Elle permet d'entraîner différents modèles d'apprentissage automatique.
(2) Bibliothèque « joblib » : Elle est utilisée pour sauvegarder un modèle d'apprentissage automatique.
(3) Bibliothèque « pandas » : Elle est la bibliothèque de Python qui traite des fichiers.

2. Implémentation de fonctionnalité

Pour détecter les mesures des données tabulaires, la solution proposée par Yuzhao YANG dans son article est mise en œuvre en trois étapes :

1. Extraire d'abord des caractéristiques de chaque colonne pour les colonnes numériques
2. Entraîner un modèle d'apprentissage automatique RandomForest avec des échantillons de caractéristiques des mesures
3. Prédire si les colonnes numériques sont des mesures en fonction de leurs caractéristiques extraites et le modèle. [1] (cf. figure 24)

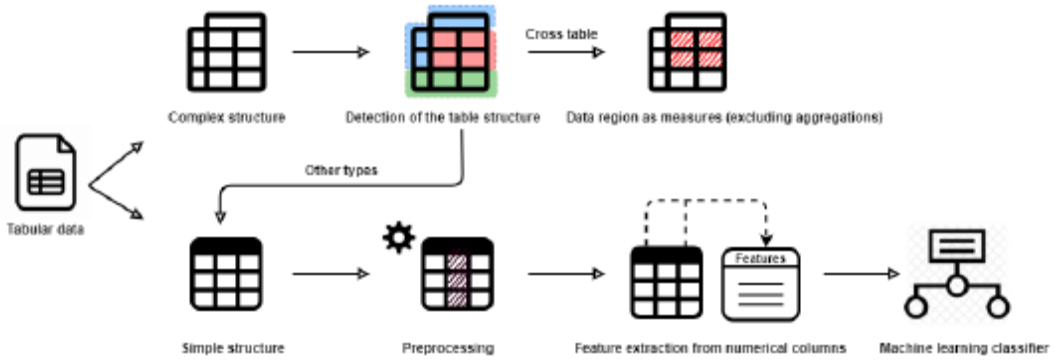


Figure 24 Détection des mesures

En suivant sa solution, j'ai entraîné un modèle d'apprentissage automatique RandomForest en utilisant la bibliothèque de python « Sklearn » et « Joblib » avec les échantillons des caractéristiques fournis par Yuzhao YANG. Pendant l'entraînement, j'ai essayé de diviser le jeu de données en deux parties : un jeu de données d'entraînement et un jeu de données de test. J'ai utilisé les méthodes de vérification croisée et de découpage pour l'entraînement. Après plusieurs comparaisons, j'ai constaté que les résultats obtenus avec la méthode de vérification croisée étaient plus efficaces que la méthode de découpage. J'ai fait part de ce résultat à Yuzhao. Après sa confirmation, j'ai utilisé la méthode de validation croisée pour l'entraînement du modèle.

J'ai également apporté des améliorations au code Python écrit par Yuzhao. Le code est utilisé pour extraire les caractéristiques des colonnes sous forme numérique et construire la fonction pour prédire les mesures à partir d'un fichier. Les caractéristiques contiennent des caractéristiques générales, des caractéristiques statiques et des caractéristiques inter-colonnes. L'expérimentation pour la trois caractéristiques classes est présente sur la section 8.2.3.

3. Transfert de données

Lorsque la fonction accepte le chemin absolu du fichier et les dépendances fonctionnelles traitées, la fonctionnalité

1. D'abord, extrait les caractéristiques des colonnes numériques du fichier.
2. Puis, utiliser le modèle d'apprentissage automatique pour prédire les mesures. Pour les colonnes qui ne sont pas prédites en tant que mesures, elles sont fournies à l'utilisateur en tant que mesures facultatives.

La méthode de Node.js stocke les résultats des dépendances fonctionnelles, des mesures prédites et optionnelles dans un fichier Json. Ensuite, elle stocke les informations du fichier Json dans la base de données, afin que l'utilisateur puisse consulter cet historique. Les mesures sont ensuite renvoyées à l'utilisateur.

4. Présentation de l'interface

Les résultats sont affichés en deux parties : les mesures prédites et les mesures optionnelles. L'utilisateur peut sélectionner les mesures qui lui semble correctes. Généralement, toutes les mesures prédites sont sélectionnées par défaut.

Nous ne disposons que l'interface de la première version de l'application. Comme nous sommes au cours du développement du serveur pour la deuxième version, nous n'avons pas encore implémenté son interface.

7.2.2.4 Détection des Hiérarchies

Après que l'utilisateur a confirmé des mesures, notre application supprime les dépendances fonctionnelles qui contiennent des mesures. Puis, elle détecte des hiérarchies basées sur les dépendances fonctionnelles restantes.

1. Bibliothèque et technique

1.1 Interface : /files/ {id}

1.2 Technique Python :
(1) Bibliothèque « nltk » : NLTK (Natural Language Toolkit) est une bibliothèque de traitement du langage naturel. Elle est la bibliothèque Python couramment utilisée dans le domaine de la recherche en NLP. Elle fournit un grand nombre d'outils pour l'analyse de texte. Elle contient beaucoup de corpus pour le traitement du langage naturel, tels que Wordnet.

2. Implémentation de fonctionnalité

Pour développer cette méthode, j'ai d'abord utilisé le code écrit par Yuzhao YANG. C'est l'algorithme récursif et les hiérarchies sont stockées dans une liste multidimensionnelle. Par la suite, Yuzhao a conseillé d'essayer l'algorithme DFS

(Depth-First-Search). J'ai écrit cet algorithme et j'ai proposé une amélioration : stocker les dépendances fonctionnelles et les hiérarchies dans des dictionnaires. De cette façon, nous pouvons détecter les dimensions et leurs hiérarchies plus facilement et plus rapidement.

En outre, lors de la détection de la hiérarchie, il faut vérifier les propriétés et les paramètres faibles pour s'assurer qu'elle est correcte. Selon des idées proposées par Yuzhao YANG dans son article, la vérification de chaque hiérarchie est implémentée à partir de deux aspects : le paramètre de niveau le plus élevé de la hiérarchie et les attributs équivalents des paramètres de la hiérarchie. [3]

Pendant l'implémentation de la vérification, j'ai utilisé le Wordnet selon la proposition de Yuzhao.

Wordnet est un dictionnaire anglais qui contient des informations sémantiques et il est différent des dictionnaires du sens habituel. WordNet regroupe les termes selon leur sens. Chaque groupe de termes ayant le même sens est appelé un synset. Il fournit une brève définition sommaire pour chaque synset et enregistre la relation sémantique entre les différents synsets. Donc, il est une bonne solution pour vérifier une relation hiérarchique. Pour cela, j'ai importé la bibliothèque Python « nltk » qui contient le corpus Wordnet. Il est utilisé pour rechercher les hyponymes de l'attribut de niveau de granularité le plus élevé et les hyperonymes d'autres attributs.

Pour l'attribut de niveau de granularité le plus élevé de la hiérarchie, j'ai implémenté le code suivant selon les étapea proposé par Yuzhao [3]:

- J'ai d'abord converti le nom de l'attribut en minuscule, ensuite vérifié si son nom contient des chaînes de caractères comme « code », « id », « no » etc.
- J'ai développé la méthode pour détecter une relation hiérarchique sémantique entre le sous-ensemble des chaînes de caractères du nom et le sous-ensemble des chaînes de caractères du nom de l'attribut subordonné via Wordnet. Si les hyponymes de l'attribut contiennent d'autres attributs, ou l'attribut est dans les hyperonymes d'autres attributs, l'attribut est un paramètre. Sinon, c'est un attribut faible.
- Pour vérifier s'il existe une relation hiérarchique sémantique entre les valeurs de l'attribut et les valeurs de l'attribut de niveau plus bas, j'ai utilisé la fonction Random pour sélectionner cent lignes au hasard. Ensuite, j'ai simplement vérifié la méthode ci-dessus. Si le nombre des valeurs qui ont une relation est plus de la moitié du nombre des lignes, il est un paramètre. Sinon, l'étape se passe à l'étape suivante.
- Pour les valeurs de l'attribut sont numériques, j'ai intégré le code de Yuzhao pour vérifier si l'attribut est séquentiel ou ordinal. Si oui, il peut être paramètre.
- Pour les valeurs de l'attribut sont en chaîne de caractère, j'ai intégré le code de Yuhzao pour identifier si l'attribut est catégoriel.
- Après toutes les étapes ci-dessus, si l'attribut est un paramètre, je le garde. Sinon, j'ai supprimé cet attribut et le placé comme attribut faible dans l'attribut de niveau suivant.

Dans un ensemble d'attribut équivalent, il existe un seul attribut qui est un paramètre. En fonction des règles proposées par Yuzhao YANG, j'ai implémenté l'algorithme :

- J'ai appliqué le même code de la première étape de vérification de l'attribut de niveau de granularité le plus élevé.
- J'ai vérifié si les valeurs de l'attribut sont des abréviations des valeurs d'autres attributs.
- Sinon, voir les valeurs de l'attribut sont le type nominal ou ordinal.

- Pour l’attribut dont les valeurs sont en chaîne, j’ai implémenté le code pour identifier si l'attribut dont les valeurs sont composées de chaînes de caractères et de données numériques.
- Enfin, pour les attributs dont la valeur est de type texte, j’ai écrit l'algorithme pour rechercher l'attribut dont la longueur des chaînes de caractères des valeurs est la plus courte via la bibliothèque Pandas de Python.

Notre application peut donc récupérer les dimensions et leurs hiérarchies validées en exécutant le script Python.

3. Transfert de données

Une fois que l’utilisateur soumet sa sélection sur des mesures, le nom du fichier et les mesures sélectionnées sont transférées dans la corp de message par l’interface « /files/ {id} » au serveur. La méthode correspondante recherche le chemin absolu du fichier. De plus, il recherche le chemin du fichier Json qui stocke les hiérarchies et les attributs équivalents. Ensuite, elle les passe au script *hierarchy.py* et exécute *hierarchy.py*. Le script exécute le code pour la détection et la vérification des hiérarchies. Lorsque le processus est déterminé, la méthode récupère les résultats et les stocke dans le même fichier Json. Enfin, elle rend la page *index.html* avec les résultats.

4. Présentation de l’interface

Tout d’abord, l’utilisateur choisit toutes les mesures auxquelles il pense correctement. Ensuite, il clique sur le bouton « confirm » pour soumettre son choix. Notre application accepte son choix et génère les hiérarchies. Enfin, le résultat est montré sur la page.

7.2.2.5 Création d'entrepôt

C’est la dernière étape de création en entrepôt. Notre application se connecte à la base de données. Puis, il exécute la commande de SQL pour créer un entrepôt en fonction des mesures, des dimensions et des hiérarchies générées.

1. Bibliothèque et technique

1.1 Interface : files/{id}/whs

1.2 Technique Python :

（1）Boîte à outil « sqlalchemy » : SQLAlchemy est une boîte à outils SQL, et une boîte à outils ORM (Object Relational Mapping) dans le langage de programmation Python. Les développeurs peuvent exploiter la base de données de manière orientée objet sans écrire d'instructions SQL.

Partant du principe que la taille et les performances d'une base de données SQL sont plus importantes qu'une collection d'objets, mais que l'abstraction d'une collection d'objets est plus importante que les tables et les lignes, SQLAlchemy utilise un modèle de mappage de données similaire à Hibernate en Java.

Selon le concept dans lequel l'ampleur et les performances des bases de données SQL sont plus importantes que les collections d'objets, mais l'abstraction des collections d'objets est plus importante que les tables et les lignes, SQLAlchemy

adopte un modèle de mappage de données similaire à Hibernate en Java.

L'objet « Engine » est le point de départ de l'utilisation de sqlalchemy. D'après le diagramme schématique de l'architecture « Engine » dans la documentation « sqlalchemy documentation - engine congifugration », « Engine » inclut un pool de connexion à la base de données (Pool) et un dialecte (Il fait référence aux différences grammaticales d'instructions SQL dans différentes bases de données) pour interagir avec la base de données d'une manière conforme à la spécification DBAPI. (cf. figure 25)

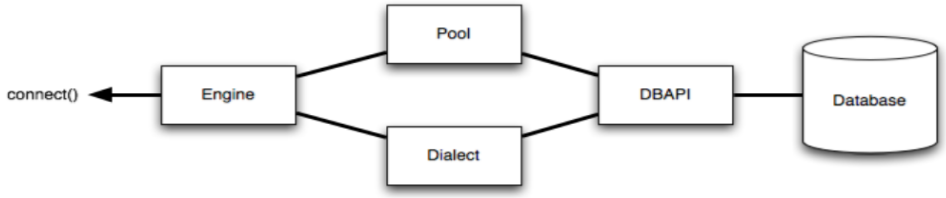


Figure 25 Diagramme schématique de l'architecture Engine

2. Implémentation de fonctionnalité

2.1. J'ai implémenté d'abord le code pour se connecter à la base de données avec le module « oracledb » de Node.js.

2.2. Ensuite, j'ai développé le code pour interroger le chemin de jeu de données et le chemin du fichier Json via Javascript.

2.3. Puis, j'ai développé la méthode pour se connecter à la base de données, créer un entrepôt, et importer les données d'un fichier dans l'entrepôt créé avec la boîte à outil « sqlalchemy » en Python.

2.4. Après, j'ai utilisé le module Child_Process pour exécuter le script Python ci-dessus.

2.5. Enfin, j'ai enregistré les informations d'entrepôt créé dans la base de données via Node.js.

3. Transfert de données

L'application accepte les hiérarchies validées par l'utilisateur via le HTTP « /files/{id}/warehouse ». Elle trouve la méthode qui exécute le script Python *createWH.py* pour créer un entrepôt via la boîte à outil « sqlalchemy ». Si créé avec réussite, la méthode retourne le nom de l'entrepôt créé à l'utilisateur, sinon retourne des erreurs.

4. Présentation de l'interface

L'utilisateur sélectionne des hiérarchies et les valide. Après, il peut recevoir le résultat de la création. Si réussie, il peut trouver le nom de l'entrepôt sur l'interface et le consulter. Sinon, il voit des erreurs produites pour la création.

7.2.3 Fusion d'Entrepôts Multiples

Cette fonctionnalité permet aux utilisateurs de sélectionner des entrepôts et de les

fusionner en un seul entrepôt.

1. Bibliothèque et technique

1.1 Interface : /whsmul/

2. Implémentation de fonctionnalité

À implémenter.

3. Transfert de données

3.1. L'utilisateur sélectionne d'abord les entrepôts à fusionner.

3.2. Les noms des entrepôts sélectionnés sont transférés au contrôleur

3.3. Le contrôleur passe les données ci-dessus au serveur

3.4. Puis la méthode correspondante se connecte à la base de données pour consulter attributs des entrepôts à fusionner et génère le SQL.

3.5. Après, la méthode utilise le SQL de créer un nouvel entrepôt et retourne le nom du nouvel entrepôt au contrôleur.

3.6. Enfin, le contrôleur transfère le résultat à la vue.

4. Présentation de l'interface

À implémenter.

7.2.4 Imputation de Données

Les données manquantes sont un problème rencontré souvent dans le stockage et l'utilisation des données. Afin d'aider nos utilisateurs qui ont peu de connaissances ou sans connaissances en BI, cette fonctionnalité les aide à traiter des données manquantes dans un entrepôt.

Dans notre application, nous ne considérons que l'imputation pour le cas où certaines lignes manquent des valeurs et les valeurs d'autres lignes sont complètes.

Le doctorant Yuzhao YANG a proposé une solution : sélectionner les valeurs d'une ligne qui est la plus similaire que la ligne qui manque des valeurs, ensuite remplir les valeurs manquantes avec les valeurs de la ligne sélectionnée.

Pour réaliser cette fonctionnalité, je l'ai fait en six étapes :

1. Entraîner un modèle de NLP « FastText »
2. Implanter l'algorithme « Spectral Clustering » dont les valeurs propres sont sélectionnées par l'algorithme « KNN »,
3. Faire des clusterings pour le résultat de « Spectral Clustering » via le modèle « KMeans » de la bibliothèque « sklearn »
4. Calculer les distances de chaque ligne au centre des centres des clusterings
5. Récupérer les lignes qui manquent des valeurs de l'entrepôt
6. Remplir les données avec le code ci-dessus.

7.2.4.1 Entraînement de Modèle FastText

Comme il existe des données tabulaires qui sont souvent textuelles, nous devons les transformer en numérique. Pour cela, nous avons besoin d'utiliser le NLP.

1. Bibliothèque et technique

1.1 Technique Python :

(1) Bibliothèque « gensim » : Gensim (Generate Similarity) est une bibliothèque de Python. Elle est simple et efficace pour le traitement du langage naturel qui est utilisé pour extraire des sujets sémantiques à partir de documents. Gensim peut traiter du texte numérique brut et non structuré (Semantic Topics). Elle convertit des mots en vecteurs de mots, afin de juger de la similarité sémantique entre différents mots en fonction des vecteurs de mots.

2. Implémentation de fonctionnalité

Au cours de programmation, j'ai entraîné un modèle « FastText » moins léger avec la moitié du corpus de vecteur de mot « wiki-news-300d-1M-subword.vec », en utilisant la bibliothèque « gensim ». Le modèle peut récupérer la similarité des valeurs de chaque ligne.

3. Transfert de données

C'est la partie de préparation du processus de l'imputation de données. Il n'existe pas d'interface pour transférer des données sur notre application

4. Présentation de l'interface

C'est la partie de préparation du processus de l'imputation de données. Il n'existe pas d'interface pour afficher ce processus.

7.2.4.2 Détection des Lignes Utilisées pour l'Imputation

L'objectif de cette fonctionnalité est d'obtenir des lignes pour compléter des données manquant dans le même entrepôt. Il est principal d'implémenter l'algorithme Spectral Clustering qui intègre les algorithmes KNN et KMeans.

1. Bibliothèque et technique

1.1 Technique Python

(1) Bibliothèque « math » : Elle fournit de nombreuses opérations mathématiques sur les nombres à virgule flottante.

(2) Bibliothèque « numpy » : C'est une bibliothèque composée d'objets de tableaux multidimensionnels et d'une collection de routines pour travailler avec des tableaux.

2. Implémentation de fonctionnalité

Après que le modèle de « FastText » est entraîné, j'ai programmé le code de l'algorithme « Spectral Clustering » en intégrant l'algorithme « KNN ». Elle est utilisée pour faire le clustering des lignes via le modèle « KMeans » de la bibliothèque « sklearn ». Le code se compose par 5 étapes :

1. Récupérer les données des lignes qui n'a aucune valeur vide
2. Obtenir la matrice de poids « Weight » en calculant le nombre des valeurs non

répétitives de chaque colonne

3. Etablir la matrice de similarité A des lignes dont la diagonale égale à 0：
   - Pour calculer la similarité d'une ligne en textuelle : Si les mots des valeurs des lignes sont dans FastText, utiliser le modèle « FastText », Sinon utiliser la bibliothèque « textdistance »
   - Pour calculer la similarité d'une ligne en numérique : utiliser la distance euclidienne avec des bibliothèques « numpy » et « math »
4. Déterminer la matrice d'adjacence W en gardant les cinq similarités plus élevées de la matrice de similarité A (Implémenter KNN). Ensuite, obtenir la matrice de degré D dont les valeurs de la diagonale est la somme de chaque ligne de A et d'autres positions sont 0
5. Obtenir la matrice laplacienne L en utilisant la formule « W – D »
6. Déterminer la matrice de Eigenvector E
7. Normaliser la matrice de Eigenvector E
8. Faire des clusterings en utilisant le modèle « KMeans » de la bibliothèque « sklearn »
9. Puis calculer la coordonnée du point central des centres de chaque clustering en utilisant la bibliothèque « math ».
10. Après, compter la distance de chaque ligne au centre.
11. Enfin, utiliser les valeurs de la ligne qui a la distance la plus proche du centre que la ligne qui manque des valeurs pour compléter les valeurs manquantes.

3. Transfert de données

C'est la partie de préparation du processus de l'imputation de données. Il n'existe pas d'interface pour transférer des données sur notre application.

4. Présentation de l'interface

C'est la partie de préparation du processus de l'imputation de données. Il n'existe pas d'interface pour transférer des données sur notre application.

7.2.4.3 Imputation de Données d'Entrepôt

La fonctionnalité complète les données manquantes avec les valeurs des lignes obtenues ci-dessus dans l'entrepôt.

1. Bibliothèque et technique

1.1 Interface : /entrepôts/{id}/complet

1.2 Technique Python :
Les mêmes bibliothèques comme la section 7.2.2.5

2. Implémentation de fonctionnalités

1. Je vais implémenter une interface permettant à l'utilisateur de sélectionner l'entrepôt à remplir.
2. Ensuite, interroger l'entrepôt pour récupérer les lignes qui manquent des valeurs dans l'entrepôt sélectionné par l'utilisateur.
3. Puis, implémenter une méthode via JavaScript pour exécuter le script de Python *SpectralCulstreing.py* . Le script se connecte à l'entrepôt, sélectionne les instances qui manquent des valeurs à remplir et toutes les instances, ensuite obtient leurs distances et remplit les valeurs vides.

4. Enfin, remplir les valeurs manquantes avec le résultat récupéré ci-dessus et retourner le résultat de l'imputation à l'utilisateur.

3. Transfert de données

1. Une fois que l'utilisateur soumet l'entrepôt, le nom de l'entrepôt se passe à la méthode correspondante via l'interface « /entrepôts/ :id/complet ».
2. La méthode exécute le script *SpectralCulstreing.py* en transférant le nom de l'entrepôt.
3. Puis, la méthode reçoit le résultat du script, le stocke dans le fichier Json et ajoute les informations de cette exécution sur la base de données de l'application.
4. Enfin, elle rend le résultat au client.

4. Présentation de l'interface

À implémenter.

7.2.5 Consultation des Historiques

Chaque fois qu'un utilisateur utilise les méthodes principales décrites ci-dessus, il peut consulter toutes les historiques de ses opérations et les résultats. Il peut savoir à tout moment s'il a déjà exécuté certain programme pour un certain fichier, consulter le résultat correspondant de chaque exécution, et l'entrepôt de données correspond aux données d'un certain fichier, etc.

1. Bibliothèque et technique

1.1 Interface : /historiques

2. Implémentation de fonctionnalités

À implémenter.

3. Transfert de données

1. Quand l'utilisateur interroge la page « historique », le serveur accepte la requête via l'interface « /historique ».
2. Ensuite, il se connecte à la base pour interroger les historiques, tels que les informations des fichiers téléchargés et leur entrepôt généré, les entrepôts fusionnés, les entrepôts d'imputation.
3. Puis, il renvoie le résultat au contrôleur.
4. Le contrôleur le rend finalement au client.

4. Présentation de l'interface

À implémenter.

7.3 Problèmes Rencontrés

Au cours de mon stage, j'ai intégré et amélioré le code existant. Le code est une partie du code de la génération d'entrepôt. J'ai implémenté de nouvelles fonctions, telles que la plupart de code de la génération d'entrepôt, la fusion d'entrepôt et l'imputation de données d'entrepôt. Pendant le développement, j'ai rencontré certains problèmes

techniques en raison de la complexité de certains algorithmes et du manque de connaissances dans de nouveau domaine.

1. Au début de mon stage, j'ai extrait l'algorithme « HyFD » à partir d'un outil. Comme l'outil utilise l'Angular dans lequel j'étais nouveau et sa taille du code est grande, j'ai passé plus d'une semaine à les lire et à trouver le code de l'algorithme. Afin qu'il s'adapte bien à notre besoin, j'ai modifié certaines méthodes et ajouté quelques méthodes. Il était difficile de rechercher des relations entre des classes, car l'algorithme est emballé en jar et le nombre de classes sont nombreux.

2. Dans la première version de l'application en Java (cf. figure 34), j'ai utilisé une bibliothèque « PythonInterpreter » pour exécuter le script Python. La bibliothèque n'a pas intégré certaines nouvelles fonctions de Python, tel que « deepcopy ». Donc, j'ai passé deux jours à rechercher une nouvelle façon pour exécuter bien un script Python. Finalement, j'ai trouvé l'autre bibliothèque « Runtime».

3. Durant le développement de la première version de l'application, les dépendances fonctionnelles transférées au script Python est une liste de Java, mais elle est changée en chaîne de caractères dans le script Python. J'ai programmé le code pour traiter des chaînes de caractères et les stocker dans une liste.

4. Pendant la mise en œuvre de la détection de hiérarchies, pour les colonnes qui ont une seule valeur, l'algorithme « HyFD » ne trouve aucune leur dépendance fonctionnelle. Pour une telle colonne (un attribut), au début, je l'ai considéré comme une nouvelle hiérarchie. Cependant, quand j'ai fait des expérimentations pour vérifier l'exactitude, cela impactait sur l'exactitude de hiérarchies. C'est parce que certains attributs, en effet, doivent être des attributs faibles de certaines hiérarchies, comme la date. Afin de résoudre ce problème, Yuzhao et moi avons discuté de l'ajout de tels attributs à chaque hiérarchie, car ils peuvent être déterminés par aucun autre attribut. Cependant, l'exactitude des expérimentations était moins basse, parce qu'ils n'ont pas dû appartenir à toutes les hiérarchies. Cela avait l'impact sur la vérification des hiérarchies. Puis, nous avons discuté de trouver une hiérarchie dont la similarité moyenne est la plus élevée pour un tel attribut, et d'ajouter l'attribut dans la hiérarchie trouvée. Pour cela, j'ai essayé le « Wordnet », mais il ne fonctionne pas bien sur le traitement des vocabulaires composés de plusieurs mots. Finalement, nous avons utilisé le modèle « FastText » et obtenu un meilleur résultat.

5. Durant les expérimentations pour vérifier l'exactitude des hiérarchies, Pour l'algorithme qui vérifie si un attribut dont le niveau est le plus élevé dans une hiérarchie, il a manqué certaine condition pour la vérification de l'attribut dont les instances sont en chaîne de caractères. Par exemple, quand « Post code » est le dernier niveau dans une hiérarchie, il n'existe pas de relation entre l'attribut « City » et l'attribut « Post code » sur Wordnet. J'ai essayé d'utiliser leur similarité recupérée sur Wordnet et FastText pour examiner leur relation, mais le résultat n'a pas répondu à mon attente. C'est parce que Wordnet ne fonctionne pas bien sur le jugement des vocabulaires incluant plusieurs mots, et la similarité recupérée par FastText n'est pas toujours assez élevée. La solution peut être de trouver une valeur à déterminer si deux attributs ont une relation inclusive.

6. Quand j'ai développé la fonctionnalité de l'imputation de données, j'ai entraîné un modèle de NLP. Pour cela, Yuzhao a proposé au début la méthode « Word2vec ». Elle est une des méthodes principales dans le domaine NLP (Nutral Language Processing). Elle peut transformer un mot en vecteur de mots. Cependant, j'ai trouvé que, pour le modèle « Word2Vec » de la bibliothèque « Gensim » de

Python, son temps de chargement (> 55ms) est très lourd à chaque fois. Après, Yuzhao YANG m'a conseillé d'essayer le modèle « Fasttext », mais nous avons rencontré le même problème (>382ms). Afin d'entraîner un modèle « Fastetext » plus léger et ayant assez de mots, j'ai recherché différents corpus. Cependant, les modèles formés sont soit trop grands en termes de taille, soit trop petits en termes de vocabulaire. Après plus de dix fois d'essai, j'ai trouvé une méthode. J'ai trouvé le corpus du modèle de « Word2vec » mais ne pris que la moitié des vecteurs des mots pour entraîner notre modèle. Enfin, j'ai stocké le modèle entraîné dans notre projet. Chaque fois quand il est importé, la vitesse du chargement est rapide et sa taille des mots est aussi assez pour notre application.

7. Notre deuxième version est développée en Node.js dans laquelle j'étais nouvelle. J'ai passé deux semaines à apprendre de nouvelles connaissances et deux jours à trouver un module de rendu des pages Web dynamiques.

7.4 Evolution de l'Application

Durant mon stage, j'ai modifié et amélioré le code existant et implémenté la fonctionnalité de la première version de notre application pour la génération d'entrepôt et presque deux grandes fonctionnalités de la deuxième version.

Au début de mon stage, j'ai implémenté la première version de l'application en Java et Ajax. Cette application est déjà réalisée toutes les étapes de la génération d'entrepôt sauf la dernière étape : la création d'entrepôt. Comme l'équipe BI4PEOPLE a proposé d'utiliser Node.js comme langage de programmation principale pour l'application générale, nous avons considéré de changer l'architecture de notre application et de recommencer notre développement de l'application. Pour correspondre à l'avancement de la thèse de Yuzhao, et éviter un deux grands changements de l'architecture de l'application, nous nous concentrons d'abord sur les fonctionnalités principales qui sont développées par Python.

J'ai mis en œuvre deux des quatre principales fonctionnalités. Ce sont les fonctionnalités les plus difficiles et les plus complexes pendant le développement. Elles font appel à des techniques et des algorithmes d'apprentissage automatique et de traitement du langage naturel.

De plus, j'ai conçu l'architecture de l'application, les modélisations des données et les interfaces pour la deuxième version. J'ai implémenté la méthode côté serveur qui exécute l'algorithme pour presque tout le processus de la génération d'entrepôt.

7.5 Travail à Faire

J'ai eu quatre grandes missions à effectuer durant mon stage, dont deux plus difficiles et complexes ont été terminé. Pour la mission sur le développement de l'application, j'ai terminé la conception de l'application et mis en œuvre la méthode de génération d'entrepôt du côté serveur. Il me restait d'autres fonctionnalités de l'application à développer, y compris la fusion d'entrepôts. (cf. figure 26)

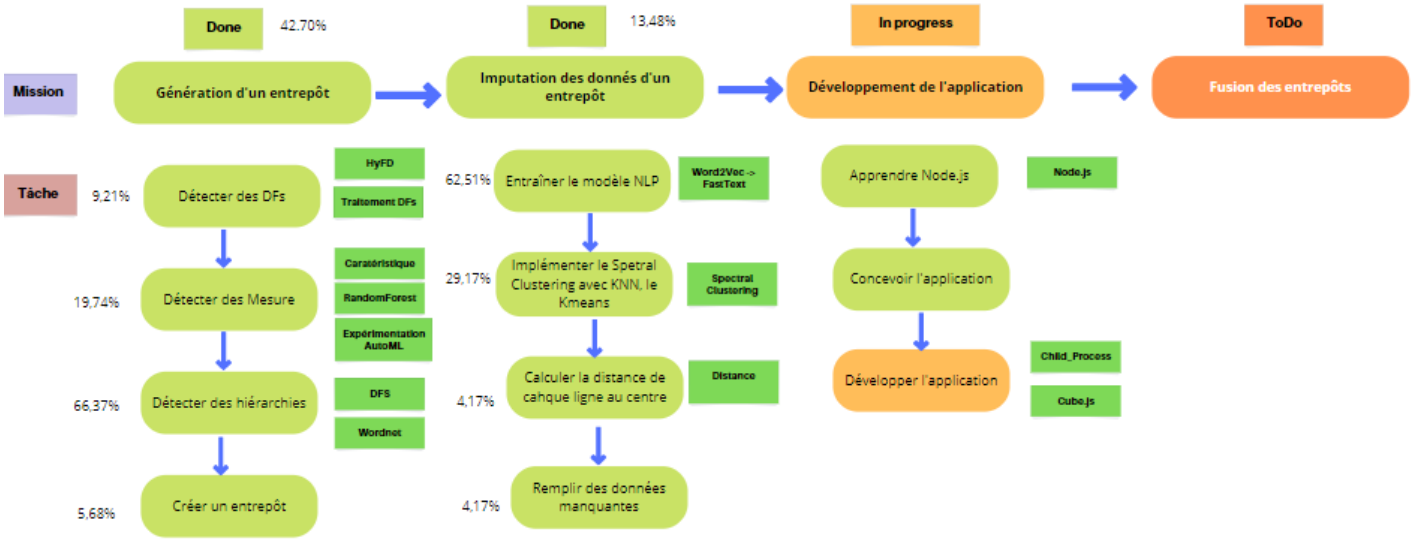


Figure 26 Processus du travail

Selon le temps de travail de chaque tâche, j'ai calculé l'importance de chaque mission. Ma durée est cent septante et huit jours et j'ai déjà bien fini environ 80% de l'importance de mes missions. Il ne restait que 20% de l'importance des missions à finir durant 51 jours, jusqu'au 7 septembre 2022. Selon la figure suivante (cf. Figure), ma vélocité normale par jour est 0,45% qui est plus élevée que la vélocité minimale 0,40% par jour. Donc, je crois que je peux finir toutes mes missions durant mon stage. (cf. figure 27)

No.	Phases	Sous-Phases	Jours	Importance-SP	Importance Total	Importance -SP/P	
1	Développement - 1	Première version de l'app	14	7.87%	7.87%	100.00%	
2	Identifier des DFs	Identifier des DFs	7	3.93%	42.70%	9.21%	
		Détection des mesures + Expérimentaton des mesures	15	8.43%		19.74%	
		Détection des hiérarchies	31	17.42%		40.79%	
		Impémenter des expérimentations et optimiser le code	23	12.92%		30.26%	
3	Imputation des données	Entraîner le modèle NLP	15	8.43%	13.48%	62.51%	
		Implémenter le Spetral Clustering avec KNN, le Kmeans	7	3.93%		29.17%	
		Calculer la distance de cahque ligne au centre	1	0.56%		4.17%	
		Remplir des données manquantes	1	0.56%		4.17%	
4	Développement - 2	Apprendre Node.js	13	7.30%	8.99%	81.25%	
		Conception de l'app	2	1.12%		12.50%	
		Développement de l'app	1	0.56%		6.25%	
5	Fusion des entrepôts	Impémenter le code	0	0.00%	0.00%	100.00%	
		Total	178	Fini	73.03%	Vélocité moyenne	0.41%
		Jours Restants	51	Tâches Restantes	26.97%	Vélocité minimale	0.53%

Figure 27 Importance des tâches

7.6 Point d'Améliorer

Pour la vérification pour l'attribut du niveau le plus élevé dans une hiérarchie dont les valeurs sont en chaîne de caractère, son algorithme peut être amélioré. Je propose deux solutions possibles. L'une est d'ajouter une valeur limite pour la similarité entre deux attributs. L'autre est de trouver l'autre modèle NLP plus correspondant à notre situation. Par exemple, pour la première solution, si utiliser Wordnet, nous pouvons les séparer d'abord les attributs qui contiennent plusieurs mots. Ensuite, calculer la similarité entre chaque élément d'un attribut et chaque élément d'un autre attribut. Enfin, voir le moyen de leurs similarités ou le résultat d'autres façons de calcul. Si on utilise FastText ou Word2Vec, nous devons trouver une valeur limite qui détermine s'il

existe entre eux une relation qui contient ou est contenue.

Si j'en reste à cette partie pour le moment, il y a un risque pour moi de ne pas pouvoir terminer le développement de l'application. Par conséquent, nous avons choisi de mettre en œuvre la deuxième version de notre application en premier. S'il reste du temps pour moi ou pour Yuzhao et Haoyang, nous pouvons améliorer cette partie.

8. Expérimentation

Au cours de mon stage, j'ai amélioré le code pour traiter les dépendances fonctionnelles et aussi le code détectant les hiérarchies. De plus, selon des besoins de Yuzhao, j'ai également fait des expérimentations. Dans cette section, je présente les résultats des améliorations et des expérimentations.

8.1 Comparaison d'Algorithmes pour Traitement des Dépendances Fonctionnelles

L'expérimentation a été conçue pour vérifier les performances de deux versions de mon code modifié : Le code du traitement des dépendances fonctionnelles. Les jeux de données que j'ai utilisés sont les mêmes que la section 6.1.1 (cf. table 10)

J'ai exécuté chaque fichier 10 fois et j'ai récupéré leur temps d'exécution moyen et la taille du stockage résultant. Les temps sont mesurés en millisecondes (ms) et les tailles sont mesurées en Ko.

La version 1 stocke les dépendances fonctionnelles dans une liste multidimensionnelle, mais pour la version 2, elles sont dans un dictionnaire. Selon le résultat présenté, nous constatons que le code de la version 2 s'exécute toujours plus rapidement que celui de la version 1, et que la taille de son stock est inférieure d'environ 90 % à celle de la version 1. (cf. table 2)

Fiche	Version1		Version2		Temps %	Taille %
	Temps1	Taille1	Temps2	Taille2		
convidIndicators.csv	1,044238	312	0	28	-100,00%	-91,03%
devApp.csv	0,117555	184	0	28	-100,00%	-84,78%
ElectronicsProductsPricingData_V1.csv	0,188495	312	0	28	-100,00%	-91,03%
Population2.csv	0,000695	120	0	28	-100,00%	0,00%
Sample - Superstore.csv	0,000995398	312	0	28	-100,00%	-91,03%

Table 2 Expérimentation - Traitement DFs

8.2 Expérimentation sur Mesure

Cette section contient deux expérimentations. L'une d'elles calcule l'exactitude des modèles pour prédire des mesures d'un fichier tabulaire, afin de sélectionner un modèle le plus approprié. L'autre consiste à calculer l'exactitude des dimensions récupérées et des hiérarchies générées. L'exactitude des deux expérimentations est jugée par la Precision (P), le Recall (R) et le F1-Score (F).

8.2.1 Expérimentation sur Modèles Prédits Mesure

Afin d'implémenter cet algorithme, j'ai d'abord appris AutoML (Auto Machine Learning), puis j'ai recherché une bibliothèque Python pour AutoML. Après deux jours de recherche, j'ai trouvé la bibliothèque Pycaret.

Comme la bibliothèque est en cours de développement, certains paramètres de certains modèles ne peuvent pas être modifiés. Pour résoudre ce problème, j'ai utilisé la bibliothèque Sklearn pour l'implémenter.

D'après nos résultats expérimentaux, nous avons constaté que le modèle RandomForest était systématiquement plus performant que les autres modèles sur trois indicateurs. Par conséquent, lors du développement de la détection des mesures, nous avons utilisé le modèle RandomForest [1]. (cf. figure 28)

	TP	FDB	RF	SVM	DT	KNN
R(%)	80.05	75.43	**96.64**	94.77	94.08	90.16
P(%)	73.57	77.50	**90.89**	78.44	88.44	87.61
F(%)	76.67	76.45	**93.65**	85.76	91.12	88.78

Figure 28 Expérimentation des modèles de mesures

8.2.2 Expérimentation sur Exactitude Caractéristique Classe

Yuzhao a proposé de prédire les mesures en utilisant trois classes de caractéristiques : les caractéristiques générales (GE), les caractéristiques statiques (ST) et les caractéristiques inter-colonnes (IC). Il a eu besoin de vérifier l'efficacité de chaque classe des caractéristiques. [1]

Suite à sa demande, j'ai mis en œuvre l'algorithme expérimental pour tester les combinaisons des classes de caractéristiques en utilisant le modèle RandomForest: une seule classe, les combinaisons de deux classes et toutes les classes.

Après l'exécution de l'algorithme, j'ai obtenu le résultat de la figure 29 (cf. figure 29). Nous avons trouvé que Nous avons constaté que les meilleurs résultats pour les trois indicateurs étaient pour la combinaison « toutes les classes » (ALL). Par conséquent, j'ai extrait les caractéristiques de toutes les classes pour les colonnes numériques dans l'algorithme de détection des mesures.

	GE	ST	IC	GE+ST	GE+IC	ST+IC	ALL
R(%)	83.33	94.10	87.20	95.47	92.01	95,41	**96.15**
P(%)	81.98	87.12	83.36	89.22	87.12	89.35	**90.79**
F(%)	82.59	90.41	85.22	92.17	89.48	92.24	**93.36**

Figure 29 Expérimentation pour Caractéristique Classe

8.3 Expérimentation sur Exactitude Dimension

Pour la dimension, nous avons vérifié son nombre et l'exactitude des attributs qu'elle contient.

1. Nombre des dimensions

Nous avons comparé les dimensions détectées par notre algorithme avec celles découvertes par des connaissances professionnelles dans chaque jeu de données. Selon les résultats expérimentaux, nous avons constaté que pour presque tous les jeux de données, les dimensions détectées par l'algorithme étaient les mêmes que celles trouvées par des connaissances professionnelles. (cf. table 3)

Fichier	P	R	F
convidIndicators.csv	100,00%	100,00%	100,00%
devApp.csv	50,00%	100,00%	66,67%
example_mesure2.csv	100,00%	100,00%	100,00%
ElectronicsProductsPricingData_V1.csv	100,00%	100,00%	100,00%
Population2.csv	100,00%	100,00%	100,00%
Sample - Superstore.csv	100,00%	100,00%	100,00%

Table 3 Nombre des dimensions

2. Attributs des dimensions

Selon ce que la table 4 montre, tous les résultats sont à 100%. Donc, tous les attributs trouvés par notre application sont les mêmes que ceux détectés par des connaissances professionnelles. (cf. table 4)

Fichier	Dimension	P	R	F
convidIndicators.csv	Pays	100,00%	100,00%	100,00%
	Indicateur	100,00%	100,00%	100,00%
devApp.csv	App	100,00%	100,00%	100,00%
example_mesure2.csv	Client	100,00%	100,00%	100,00%
	Produit	100,00%	100,00%	100,00%
	Date	100,00%	100,00%	100,00%
ElectronicsProductsPricingData_V1.csv	Produit	100,00%	100,00%	100,00%
Population2.csv	Pays	100,00%	100,00%	100,00%
Sample - Superstore.csv	Commande	100,00%	100,00%	100,00%
	Produit	100,00%	100,00%	100,00%

Table 4 Attributs des dimensions

8.4 Expérimentation sur Hiérarchie

8.4.1 Comparaison d'Algorithme de Détection de Hiérarchies

L'expérimentation teste la performance de deux versions du code que j'ai modifié : le code pour la détection des hiérarchies. Les jeux de données sont présentés dans l'Annexe la table 10 (cf. table 10)

Dans l'expérimentation, j'ai exécuté dix fois pour chaque fichier et récupéré leur moyen du temps exécuté et de la taille du stock de résultat. Le temps est calculé en milliseconde (ms), et la taille est en Ko.

La version 1 utilise l'algorithme récursif pour générer des hiérarchies et les stock dans une liste multidimensionnelle. La version 2 utilise l'algorithme DFS (Depth-First-Search) et stocke les résultats dans un dictionnaire. Nous avons constaté que l'exécution avec la version 2 donnait de meilleurs résultats que la version 1 : la plupart des fichiers s'exécutaient plus rapidement (100%) et avaient une taille de stockage plus petite. (cf. table 5)

Fiche	Version1		Version2		Temps %	Taille %
	Temps 1	Taille1	Temps 2	Taille 2		
convidIndicators.csv	0,001033	248	0	240	-100,00%	-3,23%
devApp.csv	0	184	0	240	0,00%	30,43%
ElectronicsProductsPricingData_V1.csv	0,249332	248	0,000997	240	-99,60%	-3,23%
Population2.csv	0,006022	568	0	240	-100,00%	-57,75%
Sample - Superstore.csv	0	248	0	240	0,00%	0,00%

Table 5 Expérimentation - Détection des hiérarchies

8.4.2 Expérimentation sur Exactitude de Détection de Hiérarchies

Dans cette expérimentation, nous avons testé les hiérarchies de chaque jeu de données sous quatre aspects : les paramètres des hiérarchies, les attributs faibles des hiérarchies, les niveaux de paramètres des hiérarchies et les attributs faibles des paramètres.

1. Paramètres des hiérarchies

Dans la moitié des jeux de données, les paramètres trouvés par l'application correspondent exactement à ceux détectés par des connaissances professionnelles.

Trois jeux de données sur six n'ont pas atteint une précision de 100 %, mais ils étaient égaux ou supérieurs à 75 %. C'est-à-dire que la plupart des paramètres détectés par notre application sont corrects et qu'ils sont inclus dans les attributs trouvés par des connaissances professionnelles.

Pour le recall, nous n'avons trouvé que deux jeux de données avec des résultats inférieurs à 100 %. L'un est supérieur à 90 % et l'autre est supérieur à 50 %. Cela signifie que certains attributs des deux jeux de données ne sont pas détectés par notre application, mais généralement, les paramètres trouvés par l'application sont corrects.

Pour le F1-score, nous avons trois jeux de données avec des résultats de 100%, deux jeux de données avec plus de 85% de résultats et un jeu de données avec le résultat le plus bas (60%). Selon les résultats, nous pouvons dire que la détection des paramètres des hiérarchies par notre application, est valide. (cf. table 6)

Fichier	P	R	F
convidIndicators.csv	100,00%	100,00%	100,00%
devApp.csv	77,78%	100,00%	87,50%
example_mesure2.csv	100,00%	100,00%	100,00%
ElectronicsProductsPricingData_V1.csv	75,00%	50,00%	60,00%
Population2.csv	100,00%	100,00%	100,00%
Sample - Superstore.csv	91,67%	91,67%	91,67%

Table 6 Paramètres des hiérarchies

2. Attributs faibles des hiérarchies

Dans la moitié des jeux de données, les attributs faibles des hiérarchies trouvés par l'application correspondent exactement au résultat détecté par des connaissances professionnelles.

Pour la précision, quatre jeux de données sur six sont à 100%. Pour les trois autres jeux de données, les trois indicateurs qui n'ont pas tous atteint 100%. Les résultats sont affectés par les résultats expérimentaux ci-dessus, ainsi que par les résultats de la détection de la relation entre les deux attributs dont les valeurs sont textuelles. (cf. table 7)

Fichier	P	R	F
convidIndicators.csv	100,00%	100,00%	100,00%
devApp.csv	100,00%	50,00%	66,67%
example_mesure2.csv	100,00%	100,00%	100,00%
ElectronicsProductsPricingData_V1.csv	40,00%	66,67%	50,00%
Population2.csv	100,00%	100,00%	100,00%
Sample - Superstore.csv	75,00%	75,00%	75,00%

Table 7 Attributs faibles des hiérarchies

3. Niveau des paramètres des hiérarchies

D'après les données de la table 8, nous remarquons que les résultats de la plupart des jeux de données sont égaux ou supérieurs à 70%. Les résultats de l'un des jeux de données étaient inférieurs à 50 % en raison du problème que j'ai décrit dans les expériences ci-dessus. Il pourrait également s'agir de certains attributs qui sont censées être des paramètres, mais qui sont traitées comme des attributs faibles par notre application. (cf. table 8)

Fichier	P	R	F
convidIndicators.csv	85,71%	100,00%	92,31%
devApp.csv	42,86%	60,00%	50,00%
example_mesure2.csv	70,00%	87,50%	77,78%
ElectronicsProductsPricingData_V1.csv	50,00%	40,00%	44,44%
Population2.csv	100,00%	100,00%	100,00%
Sample - Superstore.csv	70,00%	70,00%	70,00%

Table 8 Niveau des paramètres des hiérarchies

4. Attributs faibles des paramètres

Selon la table 9, nous observons que tous les attributs faibles corrects sont détectés par notre application pour la plupart des jeux de données. Le F1-score de tous les jeux des données sont supérieurs à 70%. (cf. table 9)

Fichier	P	R	F
convidIndicators.csv	100,00%	85,71%	92,31%
devApp.csv	100,00%	60,00%	75,00%
example_mesure2.csv	100,00%	100,00%	100,00%
ElectronicsProductsPricingData_V1.csv	83,33%	83,33%	83,33%
Population2.csv	100,00%	100,00%	100,00%
Sample - Superstore.csv	71,43%	71,43%	71,43%

Table 9 Attributs faibles des paramètres

9. Bilan

En raison de la fermeture de l'université et de sa non-autorisation de télétravail pour les stagiaires, mon stage a été interrompu pendant un mois (du 23 juillet au 22 août 2022) et reporté au 31 octobre 2022. Par conséquent, je n'ai effectué qu'un stage de quatre mois et demi.

Durant les quatre mois et demi (cent trente-huit jours), j'ai atteint la plupart des objectifs du stage et accompli la plupart des missions requises en fonction de l'importance des missions (cf. figure 27) :

1. La création des entrepôts représente 42,7 % d'importance de toutes mes tâches, car il y a de nombreuses fonctions difficiles et complexes.

2. L'imputation des données représente 13,5% de l'importance en raison de l'entraînement du modèle NLP.

3. L'apprentissage du nouveau langage Node.js et la conception du développement de l'application comptent pour 15,7% de l'importance.

Donc, j'ai livré environ 80% des missions en 138 jours. Jusqu'au 7 septembre 2022, il me reste encore 51 jours pour terminer les 20% de missions restantes. Comme ma vélocité normale est de 0,45% par jour et la vélocité minimale à terminer les missions restantes est de 0,40% par jour. Elle est inférieure à ma vélocité normale, je suis donc confiant de bien accomplir toutes les missions et atteindre les objectifs de mon stage.

9.1 Bilan technique

9.1.1 Difficulté du stage

Durant mon stage, la plupart des problèmes que j'ai rencontré étaient liés au manque de connaissances dans certains domaines. C'est parce que presque toutes les missions étaient mises en œuvre à l'aide de nouvelles connaissances, notamment l'apprentissage automatique et le traitement de langage naturel. Les autres problèmes concernaient le matériel.

1. Tout d'abord, je ne connaissais rien de son architecture quand j'ai extrait l'algorithme HyFD de l'outil. De plus, son architecture est assez compliquée et le code est nombreux. Il m'a fallu trois jours pour comprendre son architecture et trouver l'algorithme. Cependant, en raison de sa mauvaise structure de code (une page de code est consolidée en une seule ligne), j'ai passé deux jours à trouver l'interface liée à l'algorithme.

2. Le deuxième problème est que mes connaissances en Java n'étaient pas suffisantes pour comprendre un package jar. Comme l'algorithme était empaqueté en jar, il y avait une centaine de classes Java, et le jar ne pouvait pas être modifié. Il était difficile pour moi de comprendre tout le code du jar. J'ai donc regardé directement la classe qui imprime le résultat et j'ai trouvé l'entrée et la sortie de l'algorithme. Enfin, j'ai recréé une classe pour modifier l'entrée et la sortie afin que l'algorithme corresponde à l'entrée et à la sortie

que notre programme utilisait.

3. Le troisième problème est que je n'ai eu aucune expérience pour appeler python avec java. Je ne savais pas que Java ne peut pas exécuter des scripts Python avec certains outils tiers utilisant la bibliothèque « PythonInterpreter ». La raison est que le chemin de la bibliothèque n'est pas sous le projet d'exécution. Après, j'ai changé en « Runtime », le script Python a finalement été exécuté. De plus, quelle que soit la forme du paramètre est transmise de Java à Python, il devient une chaîne de caractères. En raison de ce problème, j'ai réécrit le code qui traite des dépendances fonctionnelles.

4. Le quatrième problème porte également sur la technique. Avant mon stage, je ne connaissais pas Node.js et je ne savais pas que Javascript pouvait être un langage de programmation serveur. Pendant mon stage, j'ai eu une expérience systématique de la programmation JavaScript. J'ai beaucoup appris sur Node.js et j'ai une bonne compréhension du framework Electron.

5. Le cinquième problème est que je ne connais rien aux algorithmes de recherche de graphes. Pour implémenter les algorithmes BFS « Breadth-First-Search » et DFS « Depth-First-Search », j'ai d'abord appris leurs concepts, ensuite regardé de différents exemples afin de trouver leurs règles. Après plusieurs essais, j'ai finalement implémenté le code de détection des hiérarchies avec l'algorithme DFS.

6. Le sixième problème est que je n'ai jamais utilisé le traitement de langage naturel et je ne savais qu'il peut être utilisé pour la traduction. En faisant des recherches et en lisant des articles sur ce sujet, j'ai eu une meilleure compréhension du traitement du langage naturel. J'ai eu la première expérience avec les modèles NLP durant mon stage.

7. Le dernier problème concerne le matériel. Mon ordinateur ne pouvait pas installer la base de données Oracle car le compte de mon ordinateur est toujours mal identifié. J'ai essayé plusieurs méthodes, telles que le changement de la configuration de mon ordinateur, mais le problème n'a toujours pas été résolu. J'ai réinstallé le système et j'ai finalement résolu le problème. Après avoir essayé diverses solutions, j'ai appris des solutions à divers problèmes d'installation, notablement sur la base de données Oracle.

9.1.2 Apport du stage

Pendant mon stage, je n'ai cessé d'acquérir de nouvelles connaissances. J'ai appris toujours de nouvelles choses dans différents domaines et je les ai appliquées à mon développement.

1. L'implémentation de la création d'entrepôt à partir des données tabulaires m'a permis d'acquérir une compréhension plus approfondie et plus complète de la modélisation de la base de données relationnelle. Je suis familiarisé avec le traitement des fichiers tabulaires et des données via Python. J'ai acquis de nouvelles connaissances sur l'apprentissage automatique.

2. En travaillant avec des données textuelles, j'ai appris différentes méthodes NLP. Au cours de l'utilisation des modèles Wordnet, Word2Vec et FastText, j'ai

appris comment ils diffèrent et où ils peuvent être appliqués.

3. Au cours de programmation de l'imputation de données, j'ai appris à traiter des valeurs manquantes et des modèles de classification qui sont une partie importante dans le domaine de l'apprentissage automatique.

4. De plus, grâce aux expériences de l'entraînement des modèles de l'apprentissage automatique, j'ai eu une bonne compréhension des très célèbres bibliothèques d'apprentissage automatique en Python. J'ai aussi une bonne compréhension des méthodes pour l'entraînement des modèles de l'apprentissage automatique.

5. En outre, le développement de l'application en JavaScript, m'a permis d'être plus familière avec la programmation en JavaScript, d'avoir d'une bonne expérience dans la conception de l'architecture de l'application et de l'API RestFul. J'ai compris Node.js et ses modules pour le rendu des pages du site. Je savais comment utiliser le module « fs » de Node.js pour la lecture et l'édiction des documents.

6. J'ai utilisé plusieurs langages de programmation pendant le développement, tels que Java, Python et Javascript. C'était la première fois que je développais une application combinant différents langages de programmation. m'a permis de me familiariser non seulement avec les trois langages de programmation, mais aussi avec la façon dont les différents langages de programmation interagissent.

7. Au cours de mon stage, je suis devenu plus compétent dans l'utilisation d'outils tels que Github et Sonnar Cloud. J'étais profondément conscient qu'ils constituent une excellente solution pour l'intégration continue et la réduction de la dette technique.

9.2 Bilan professionnel

Tout d'abord, au cours de mon stage, j'ai communiqué avec Yuzhao et Haoyang presque tous les jours au sujet de nos besoins, de l'avancement du développement de notre application et des difficultés que nous avons rencontrées. Lorsque j'ai eu des questions sur les besoins et que Yuzhao était disponible, je lui ai posé ces questions, afin de mieux comprendre ses besoins. un meilleur algorithme

Lorsque le résultat de l'algorithme ne correspondait pas à notre attente, je l'ai dit à Yuzhao à temps pour ne pas perdre de temps à chercher d'autres solutions. Parfois, nous avons discuté ensemble, ou je lui ai proposé des options alternatives. our les questions plus spécialisées, tels que comment trouver un corpus relativement léger avec suffisamment de mots, j'ai demandé à des collègues du bureau ayant une expérience dans le domaine.

Au cours de mon stage, des expériences de communications me permettent d'avoir plus de confiance dans la communication professionnelle. Avec la communication professionnelle, j'ai seulement besoin d'exprimer mes opinions et mes problèmes avec courage et clarté. De cette manière, l'autre partie peut comprendre mes pensées, comprendre mes problèmes et les résoudre. La communication professionnelle peut m'aider aussi à s'adapter facilement et rapidement à l'environnement du travail.

Puis, lorsque j’ai rencontré un problème technique, j’ai toujours fait de mon mieux pour trouver différentes solutions. Si j'avais assez de temps, je continuerais à essayer ces solutions jusqu'à ce que je puisse le résoudre. Cela se passait également dans mon travail, si j'avais le temps, j'essayais toujours d'améliorer mon travail.

De plus, la gestion de projet a rendu le suivi de l’avancement des missions plus clair et plus facile. Je pouvais bien connaître la vélocité du travail et changer mon rythme de travail. C’est une bonne méthode pour gérer les risques de notre travail ou de notre projet.

Enfin, développer avec la méthode Agile personnalisée durant mon stage, m’a permis de livrer constamment de nouvelles fonctionnalités et de gérer les risques pendant mon développement. En particulier, quand j’étais bloquée sur un problème, je pouvais demander l'aide à Yuzhao ou à d'autres personnes à temps. Avant de trouver une nouvelle solution, je pouvais effectuer d'autres tâches simples au fur et à mesure de l'avancement du projet.

Grâce à cette expérience, je pense pouvoir m'adapter rapidement à mon environnement du travail, résoudre des problèmes rapidement et m'entendre bien avec mes collègues.

9.3 Bilan personnel

Au cours de ce stage, j'ai appliqué beaucoup de ce que j’ai appris pendant mon Master.

Premièrement, étant donné que la tâche principale de mon stage est la mise en place d'entrepôt de données, cela implique beaucoup de connaissances sur bases de données relationnelles, notamment la modélisation d'entrepôt de données. Ces connaissances m’ont permis de comprendre rapidement les besoins de développement et le métier. Je pouvais donc développer les fonctionnalités correspondant aux besoins.

Deuxièmement, j'ai fait un projet de tri des déchets en Master 1. Dans ce projet, j’ai principalement utilisé Python pour filtrer et traiter les données tabulaires sur les informations d'emballage de près d'un million de produits. 'ai également mis en œuvre le scan de QR Code via Python en fonction des données traitées. De plus, dans le cours Big Data de Master 2, j'ai utilisé Python pour la collecte et le traitement des données. Ces techniques ont été pleinement appliquées pendant mon stage. Elles m'ont permis d'apprendre plus facilement des techniques plus complexes pendant mon stage.

Troisièmement, le dernier projet de Master 2 m'a permis d'acquérir une compréhension approfondie de la conception et de l'application de l'architecture de l’application, comme le modèle MVC, la conception d'interfaces API et le rendu de pages JSP. Toutes ces connaissances ont constitué une base importante pour le déroulement de mon stage. Cette expérience m'a permis de développer facilement une application. Cela m'a fait comprendre que même si les langages de programmation sont différents, l'architecture et les principes techniques des applications sont toujours similaires. Dans ce projet, nous avons également travaillé avec la méthode de développement agile pour comprendre comment développer en fonction des besoins des utilisateurs, comment travailler en équipe, comment visualiser l'avancement du projet et gérer les

risques du projet. Cette expérience m'a guidé dans le développement d'applications pendant mon stage.

Par conséquent, les connaissances acquises au cours du master et l'expérience acquise dans le projet, elles ont permis à mon stage de se dérouler bien. Je crois que toutes les expériences, les connaissances et les techniques apprises pendant ces deux ans et au cours de mon stage, seront d'une grande aide pour mon travail futur.

10. Conclusion

La Business Intelligence est une technologie importante qui fournit une aide à la décision aux décideurs dans divers domaines. Elle utilise des techniques d'entrepôt de données, des techniques analytiques, des techniques d'exploration de données et des techniques de visualisation de données pour analyser les données. Parmi elles, l'entrepôt de données est la base de la Business Intelligence. Actuellement, les différentes étapes de la Business Intelligence sont gérées par des personnes ayant des connaissances spécialisées. Cependant, en raison des coûts élevés et d'autres raisons, certaines petites entreprises, organisations et individuelles n'ont pas d'un budget suffisant pour engager des professionnels. Ainsi, BI4PEOPLE (Business intelligence for the people) propose des solutions pour automatiser le processus de Business Intelligence. Mon tuteur entreprise, Yuzhao YANG, était chargé de la première partie du processus : la réalisation de « l'intégration automatique de données dans les entrepôts de données ». Le but de mon stage est de mettre en œuvre la solution proposée par Yuzhao et de développer une application basée sur cette solution : génération automatique d'entrepôts de données à partir de données tabulaires, fusion d'entrepôts multiples, et imputation de données manquantes dans les entrepôts.

Dans ce projet, j'ai extrait l'algorithme « HyFD » d'un outil. J'ai intégré et amélioré le code existant qui traite des dépendances fonctionnelles. Ensuite, j'ai implémenté les missions les plus importantes, y compris toutes d'autres étapes de la génération automatique d'entrepôts qui est la partie la plus importante dans mon stage, et les étapes de l'imputation de données des entrepôts. Puis, j'ai conçu l'application et mis en place certaines fonctionnalités de l'application. De plus, j'ai implémenté le code des expérimentations pour vérifier la faisabilité de la solution de Yuzhao. Selon ma vélocité du développement et le temps restant de mon stage, je crois que je vais bien finir les missions à effectuer : Elles prennent environ 20% de l'importance de toutes mes missions : fusion des entrepôts multiple et mise en œuvre les fonctionnalités restantes de l'application.

Je suis très heureuse d'avoir participé à ce stage : J'en suis honorée et reconnaissante. J'y ai beaucoup gagné : l'augmentation des connaissances, le renforcement des compétences et l'amélioration des capacités. L'expérience de mon stage m'a fourni une bonne opportunité d'appliquer ce que j'ai appris pendant mes deux ans d'études en master. Au cours de mon stage, j'ai de chance d'apprendre des connaissances dans le domaine de l'intelligence artificielle, tel que l'apprentissage automatique et du traitement. Je me suis familiarisé avec le traitement des données et les langages de programmation, tel que Python, Javascript et Java. Il me permet de connaître le Node.js et le Framework Electron, d'améliorer mes capacités de communication, de résolution d'un problème et de gestion du projet.

Dans ce rapport de stage, j'ai détaillé le contexte et l'objet du stage, les méthodes de gestion de projet utilisées, les techniques appliquées, la réalisation des différentes tâches, la conception et les différentes étapes de la mise en place de l'application, ainsi que les résultats et les gains de ce stage. Ce stage m'a beaucoup apporté, et ce fut un succès, car il m'a permis de faire la transition d'étudiant à travailleur et m'a préparé à un futur travail.

Références

[1] Yang, Y., Abdelhédi, F., Darmont, J., Ravat, F., & Teste, O. (2022). Automatic Machine Learning-Based OLAP Measure Detection for Tabular Data. In International Conference on Big Data Analytics and Knowledge Discovery (pp. 173-188). Springer, Cham.
[2] Yang, Y., Abdelhedi, F., Darmont, J., Ravat, F., & Teste, O. (2021, September). Internal Data Imputation in Data Warehouse Dimensions. In International Conference on Database and Expert Systems Applications (pp. 237-244). Springer, Cham.
[3] Yang, Y., Darmont, J., Ravat, F., & Teste, O. (2021, July). An Automatic Schema-Instance Approach for Merging Multidimensional Data Warehouses. In 25th International Database Engineering & Applications Symposium (pp. 232-241).
[4] Yang, Y., Darmont, J., Ravat, F., & Teste, O. (2022). Dimensional Data KNN-Based Imputation. In European Conference on Advances in Databases and Information Systems (pp. 315-329). Springer, Cham.
[5] Yang, Y., Darmont, J., Ravat, F., Teste, O.: Automatic Integration Issues of Tabular Data for On-Line Analysis Processing. In: 16e journ´ees EDA Business Intelligence & Big Data (EDA 2020). vol. B-16, pp. 5–18 (2020)
[6] Yang, Y., Darmont, J., Ravat, F., Teste, O.: Automatic Integration Issues of Tabular Data for On-Line Analysis Processing. In: 16e journ´ees EDA Business Intelligence & Big Data (EDA 2020). vol. B-16, pp. 5–18 (2020)
[7] F Ravat, O Teste, R Tournier, G Zurfluh. (2011). Multidimensional Database Design from Document-Centric XML Documents. In: Cuzzocrea, A., Dayal, U. (eds) Data Warehousing and Knowledge Discovery. DaWaK 2011. Lecture Notes in Computer Science, vol 6862. Springer, Berlin, Heidelberg.
[8] Ravat, F., Teste, O., Tournier, R., Zurfluh, G. (2009). Designing and Implementing OLAP Systems from XML Documents. In: Kozielski, S., Wrembel, R. (eds) New Trends in Data Warehousing and Data Analysis. Annals of Information Systems, vol 3. Springer, Boston, MA.
[9] Pedro J. García-Laencina, José-Luis Sancho-Gómez, Aníbal R. Figueiras-Vidal, Michel Verleysen. (2009). K nearest neighbours with mutual information for simultaneous classification and missing data imputation. Neurocomputing,Volume 72, Issues 7–9,2009,Pages 1483-1493,ISSN 0925-2312.
[10] Vallabhajosyula, Manikya Swathi. (2018). Hypernym Discovery over WordNet and English Corpora - using Hearst Patterns and Word Embeddings. Diss. University of Minnesota.
[11] Papenbrock, T., Naumann, F.: A hybrid approach to functional dependency discovery.In: International Conference on Management of Data. p. 821–833 (2016)
[12] Papenbrock, T., Ehrlich, J., Marten, J., Neubert, T., Rudolph, J., Sch¨onberg, M., Zwiener, J., Naumann, F.: Functional dependency discovery: An experimental evaluation of seven algorithms. In: VLDB Endowment. vol. 8, p. 1082–1093 (2015)

Annexe

1. Les cinq jeux de données sont utilisés pour vérifier l'exactitude des algorithmes et l'efficacité de l'application (cf. table 10).

Nom	Taille	Description	Source
dashboard-data-April.xlsx	18794KB	Le dataset présente l'impact socio-économique du COVID-19 sur les ménages et les individus dans toutes les régions en développement. Il fournit des données sur plus de 155 indicateurs dans 16 domaines, y compris l'éducation, la sécurité alimentaire, les revenus, les filets de sécurité, etc.	https://datacatalog.worldbank.org/search/dataset/0037769/Harmonized-COVID-19-Household-Monitoring-Surveys
20102018DevAppsFinal.xlsx	513KB	Il concerne les demandes de développement soumises au Département des services de planification et de conception entre janvier 2010 et juin 2018.	https://data.louisvilleky.gov/datasets/louisville-metro-ky-development-applications-2010-2018-historical/explore
DatafinitiElectronicsProductsPricingData.csv	81819KB	Il s'agit des informations des produits sur les sites.	https://data.world/datafiniti/electronic-products-and-pricing-data
Popular Indicators.xlsx	14KB	Il présente la population des states américains.	https://databank.worldbank.org/source/world-development-indicators
superstore_sales.csv	11233KB	Il s'agit d'une liste de plus de 7 000 produits électroniques avec des informations sur les prix dans 10 champs uniques fournis par la base de données de produits de Datafiniti. L'ensemble de données comprend la marque, la catégorie, le marchand, le nom, la source, etc.	https://www.kaggle.com/datasets/shreyhenry/superstore-sales-data

Table 10 Origine - Jeux de données

2. Les fichiers CSV sont transformés par les cinq jeux de données ci-dessus.
   Si un jeu de données est un fichier xlsx, il existe éventuellement plusieurs feuilles dans un fichier. Comme notre application génère un entrepôt à partir d'un fichier csv, chaque feuille de table est d'abord transformée en un fichier csv. Donc, nos jeux de données au format xlsx sont stockés au format csv et les autres ne changent pas. (cf. table 11)

Nom	Taille	Original Fichier
convidIndicators.csv	35264KB	dashboard-data-April.xlsx

devApp.csv	248KB	20102018DevAppsFinal.xlsx
ElectronicsProductsPricingData_V2.csv	3487KB	DatafinitiElectronicsProductsPricingData.csv
Population2.csv	4KB	Popular Indicators.xlsx
Sample - Superstore.csv	2197KB	superstore_sales.csv

Table 11 Transformation - Jeux de données

3. Implémentation en Python
   La figure suivante montre le projet Python pour l'implémentation des fonctionnalités sur l'identification des dépendances fonctionnelles, la détection des mesures, la détection des hiérarchies et les expérimentations présentées dans le chapitre 8. (cf. figure 30)

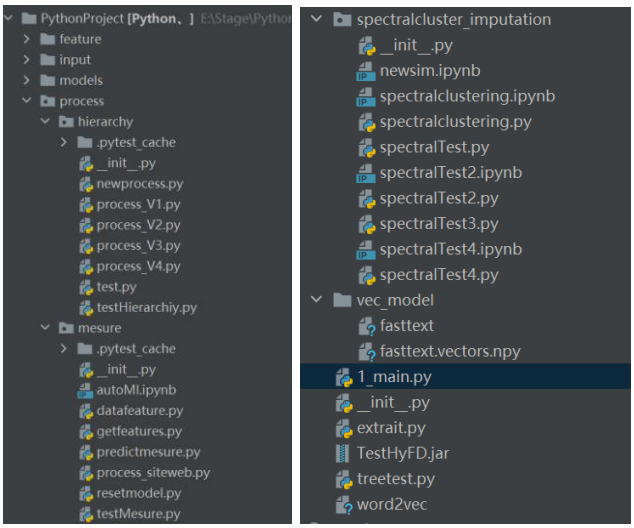


Figure 30 Implémentation Python

4. Extrait et Modification de l'Algorithme HyFD
   4.1. La figure 31 montre le code de l'outil qui inclut HyFD. (cf. figure 31)

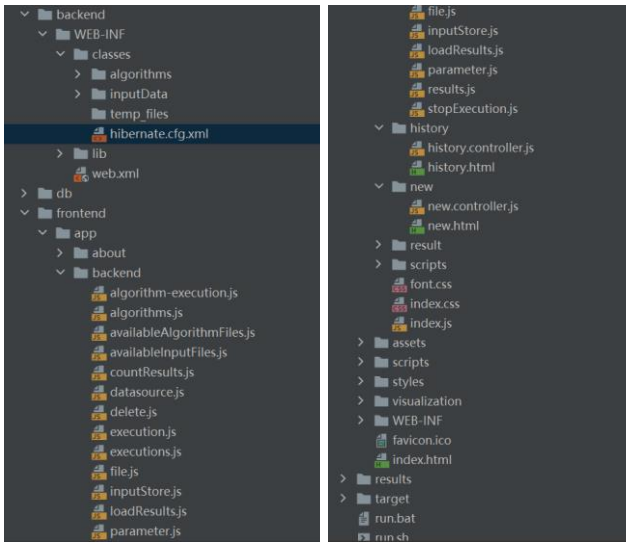


Figure 31 Code d'Outil

4.2. La figure 32 montre le code de l'algorithme HyFD. Dans ce package jar, il existe nombreuses classes (plus de cent classes). (cf. figure 32)

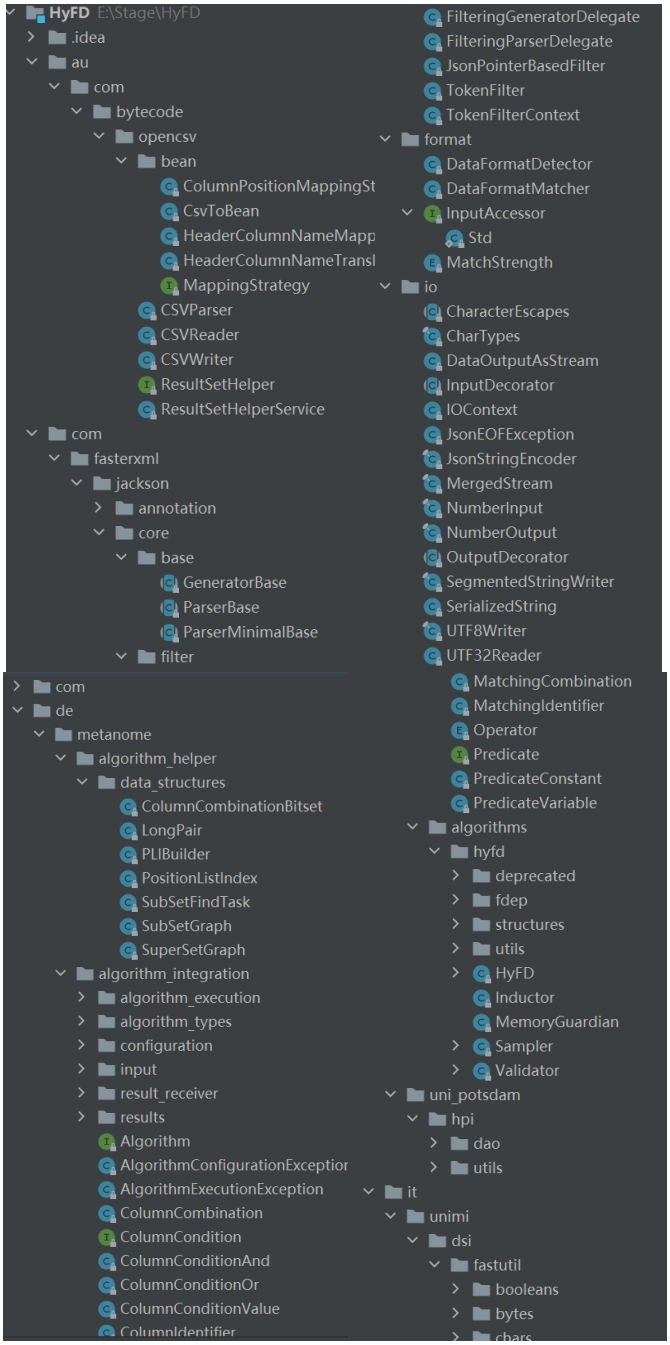


Figure 32 Algorithme HyFD

4.3. La figure 33 présente la partie de modification pour l'algorithme HyFD. Je l'ai empaqueté en TestHyFD.jar après la modification. (cf. figure 33)

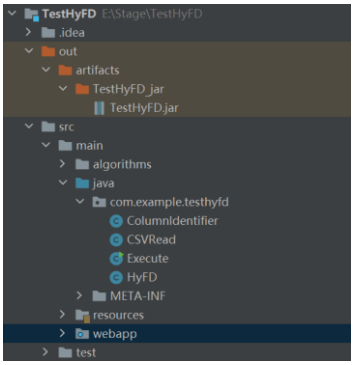


Figure 33 Modification pour HyFD

5. Implémentation de la Première Version de l'Application en Java. (cf. figure 34)

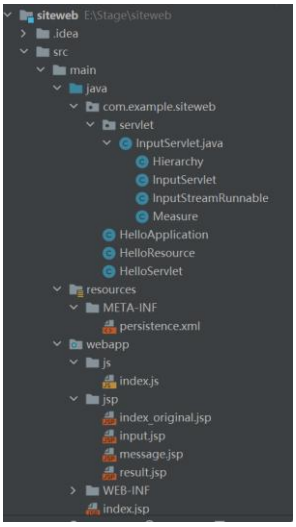


Figure 34 Implémentation Première Version d'Application

6. Implémentation de la Deuxième Version de l'Application en JavaScript. (cf. figure 35)

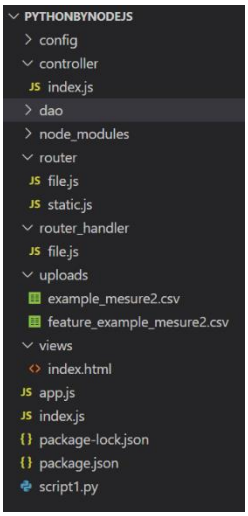


Figure 35 Implémentation Deuxième Version d'Application