CentraleSupélec

PRIMEVER

Projet Industriel - 07

Smart ETL : IA et MLOps au service de la performance de la data

Antoine VALETTE - Leyla YUSUPJANOVA - Manon KOBSCH - Guillaume RAMIREZ

Le projet vise à améliorer le suivi des processus ETL (*Extract, Transform, Load*) pour garantir la fluidité des flux de données au profit des utilisateurs du Groupe Primever. Avec la montée en puissance de l'intelligence artificielle (IA), l'objectif est aussi d'intégrer ces nouvelles technologies pour renforcer le *monitoring* proactif et améliorer la robustesse les pipelines de données.

Client : M. Balthazar MÉHUS (Data Scientist – Groupe Primever)

Année académique 2024 - 2025

Sommaire

CentraleSupélec

Projet Industriel Ms MSI & SIO P2025

P07

Smart ETL : IA et MLOps au service de la performance de la data

Client : Groupe Primever – Leader français du transport de fruits et légumes.

Equipe : Service Data, composé de 6 personnes (1 responsable data, 2 docteurs en data science, 2 data analysts, 1 data scientist).

Projet : Analyse, optimisation, monitoring, alerting et prédiction sur des centaines de data pipelines (type ETL).

Encadrant :

Responsabilités : Data scientist au sein de l'équipe

CV rapide : Issu des systèmes d'information de l'armée de Terre**, Balthazar Mehus**, s'est réorienté vers la data science grâce au MS SIO (P2023), après un stage en tant que chef de projet cloud devOps dans une ESN d'infogérance cloud.

Coordonnées : balthazar.mehus@primever.com, 06 84 78 93 94.

• Contexte du projet :

Vous intégrerez l'équipe dédiée à la data (traitement, analyse et valorisation de données à grande échelle) du groupe Primever, spécialisé dans le transport de fruits et légumes, du producteur au consommateur. Ce groupe met la DSI, et plus particulièrement le service Data, au cœur de sa croissance. Notre équipe regroupe des experts en **data engineering, data analyse et data science.** Ces domaines sont au cœur de l'évolution des entreprises modernes. Vous aurez l'opportunité de travailler directement avec ces professionnels pour développer des solutions qui auront un impact concret.

Aujourd'hui, l'un des défis majeurs de l'équipe est de **monitorer efficacement les processus ETL** (Extract, Transform, Load) pour assurer la fluidité des flux de données, au profit des utilisateurs du groupe Primever. Avec la montée en puissance des technologies d'**intelligence artificielle (IA)** et des **grands modèles de langage (LLM)**, nous souhaitons intégrer ces concepts afin d'améliorer le monitoring proactif et automatiser nos pipelines de données.

Faites partie d'une équipe data expérimentée, dans une dynamique d'entreprise « data driven ».

• Objectifs du projet :

Ce projet propose de **développer un prototype** par une approche progressive avec des objectifs variés :

1. **Amélioration des processus de monitoring** des ETL en temps réel, avec un focus sur la fiabilité et la performance des pipelines de données.
2. **Exploitation des technologies d'intelligence artificielle** pour introduire des capacités de prédiction et d'anticipation des erreurs dans les ETL.
3. **Bonus**, exploration d'une solution de type **Retrieval-Augmented Generation (RAG)**, où un **LLM** pourrait interagir avec une base de données pour répondre à des requêtes en langage naturel sur les logs d'ETL, sans passer par un tableau de bord classique.

CentraleSupélec

Projet Industriel Ms MSI & SIO P2025

P07

Smart ETL : IA et MLOps au service de la Performance de la Data

• **Travail et livrables attendus** :

1. **Rapport d'analyse historique** des ETL (nombre de chargements, temps d'exécution moyens, erreurs fréquentes).
2. **Base de données structurée** prête à être utilisée pour l'analyse des logs d'ETL.
3. **Dashboard interactif de monitoring** des ETL, permettant une visualisation en temps réel des indicateurs de performance.
4. **Système d'alerting automatisé**, pour notifier les équipes en cas de plantage ou d'anomalie dans les processus ETL.
5. **Modèle prédictif** capable de détecter et anticiper les risques de plantage des ETL, avec une intégration dans une démarche **MLOps** pour assurer le déploiement et le monitoring du modèle en production.

• **Aspects techniques :**

L'accent est mis sur l'adaptabilité des outils. Voici des exemples d'outils couramment utilisés pour réaliser les différentes tâches du projet :

- **Gestion des bases de données et stockage de logs** : PostgreSQL, SQLserver, MongoDB…
- **Intelligence artificielle :** Scikit-learn, Pytorch, TensorFlow et Hugging Face pour les LLMs.
- **Orchestration et tracking** : Mlflow, prefect, Mage, Python, Bash (sur VM Debian).
- **Visualisation et monitoring en temps réel** : Grafana, Kibana...
- **Déploiement d'infrastructure :** Docker, Kubernetes… et Terraform (IaC).
- **Systèmes d'alerting :** Outlook, Microsoft Teams, Slack
- **Développement collaboratif, versioning et DevOPs :** Gitlab.
- **Méthodologie :** Agile, Lean Startup.

Ces outils ne sont que des suggestions. En effet, chaque étape du projet pourra être réalisée avec des technologies différentes en fonction des préférences de l'équipe projet. Donc, libre à vous de choisir d'autres alternatives adaptées aux objectifs, à vos compétences, à vos appétences et à nos contraintes.

1. Introduction

A l'ère du *Big Data*, la donnée devient de plus en plus centrale à l'ensemble des processus de gestion et de pilotage de l'entreprise. Il devient donc primordial d'améliorer l'accessibilité et la facilité d'utilisation des données pour les différents services interne. En conséquence, il en découle de forts enjeux autour des processus d'acquisition, de transformation et de mise à disposition des données.

Ces processus, communément appelés ETL *(Extract, Transform, Load)*, sont donc au cœur de la stratégie de développement de l'entreprise [1][2]. Ils peuvent même devenir critiques au bon fonctionnement de l'entreprise. En effet, les erreurs, les interruptions ou encore les défaillances des ETL peuvent perturber les opérations ou impacter les services supports internes. Face à ce constat, la nécessité de suivre ces ETL et de mettre en place des pratiques de *monitoring* devient évidente.

C'est par ailleurs dans ce contexte qu'a été penser le **projet « Smart ETL »** du groupe Primever. Suite à son orientation technologique vers le *Big Data*, cette entreprise française spécialiste du le transport de fruits et légumes en Europe, comptabilise plus d'une centaine d'ETL pour garantir la fluidité des flux de données.

Concrètement, le projet se traduit par un premier objectif qui consiste à mettre en place un *monitoring* des ETL capable de surveiller et détecter les anomalies. De plus, un second objectif porte sur le suivi en temps réel et même la prédiction des erreurs pour apporter davantage de résilience aux équipes opérationnelles. L'ambition du projet est donc d'optimiser et de fiabiliser pipelines de données de Primever.

2. Contexte & Présentation de l'entreprise

2.1. Présentation de l'entreprise Primever

Avec son expertise unique et un maillage intelligent des territoires, le groupe Primever se spécialise dans le transport et la logistique de la filière Fruits et Légumes (température dirigée) et de certaines marchandises industrielles (température ambiante) à la fois en France et à l'international. Fondée en 1963 pour répondre aux besoins croissants du marché des denrées périssables, l'entreprise opère aujourd'hui avec un large réseau de sites stratégiquement localisés pour assurer le transport du producteur (coopératives, expéditeurs) au distributeur (grandes et moyennes surfaces, grossistes, spécialistes).

La figure 1 présente quelques chiffres clés du groupe et davantage d'informations sur Primever est disponible sur leur site web.

Figure 1 : Chiffres clés de Primever [3]

Avec un engagement envers l'innovation et la durabilité, Primever ne cesse d'adapter ses technologies et processus pour répondre aux défis contemporains, comme la gestion optimisée des ressources et la réduction de son empreinte environnementale. Pour se faire, l'entreprise s'appuie sur un système d'information (SI) alimenté par des données opérationnelles en permanence.

2.2. L'équipe Data de Primever

Pour exploiter son SI à son plein potentiel, Primever a structuré une équipe *Data* en interne. Composée de 6 personnes (1 *data manager*, 2 docteurs en *data science*, 2 *data analysts* et 1 *data scientist*), cette équipe a plusieurs rôles :

- Mettre à disposition et maintenir des tableaux de bords pour les autres services de l'entreprise (*business intelligence*) ;
- Mener des analyses spécifiques (analyse de données et/ou *data science*) ;
- Explorer l'utilisation de nouvelles technologies telles que l'intelligence artificielle (IA) ou encore les *Large Language Models* (LLMs) pour différents cas d'usage.

Dans ce contexte, l'équipe Data rencontre aujourd'hui plusieurs défis. L'un d'entre eux porte sur le suivi efficace des processus ETL (*Extract, Transform, Load*) pour garantir les flux de données. On désigne cette action de suivi, ou de surveillance, par son terme anglosaxon *monitoring*. De plus, les processus ETL désignent l'ensemble des processus de collecte, de transformation et de mise à disposition des données de l'entreprise au sein d'un entrepôt de données. La finalité des ETL est bien évidemment de tendre à devenir 100% automatique.

Pour aller dans ce sens, l'équipe *Data* cherche donc à optimiser et rendre robuste ces ETL. Malheureusement elle ne dispose actuellement pas d'un moyen simple et automatique pour monitorer ces processus ETL.

2.3. Enjeux et objectifs du projet industriel

L'équipe *Data* de Primever, représentée par Balthazar MÉHUS, nous a donc missionnés afin de concevoir un tableau de bord interactif pour le *monitoring* en temps réel des processus ETL avec un système de notification en cas de défaillance ou d'évènement critique. Ce travail sera réalisé en grande partie sur la base de l'analyse des différents logs (journal de bord des évènements) du SI et des ETL.

Pour aller au-delà du "simple" *monitoring*, un second objectif est aussi de permettre la prévision d'erreurs et d'anomalies des ETL. Concrètement, il s'agit de développer un modèle prédictif capable d'anticiper les erreurs des ETL (sur la semaine à venir) et l'intégrer dans un *workflow* MLOps pour assurer un déploiement et un suivi automatisé.

Enfin, selon notre avancement, un troisième objectif viendrait en complément des deux premiers. Il porterait sur la conception d'une solution permettant l'interrogation directe des logs via des requêtes en langage naturel. Cet outil permettrait à l'équipe *Data* de diagnostiquer des erreurs d'ETL en complément des *dashboards* visuels utilisés pour le *monitoring*.

3. Points clés et choix techniques

3.1. Architecture fonctionnelle

Pour répondre à l'objectif de développement d'un outil de *monitoring* et d'analyse des logs, nous pouvons décomposer notre future solution en 4 fonctionnalités distinctes (voir figure 2) :

- Un mécanisme de récupération des données (différents logs) ;
- Un espace de stockage des données transmises ;
- Un outil de *monitoring* et de notifications ;
- Une capacité d'analyse des logs par des algorithmes de *Machine Learning*.

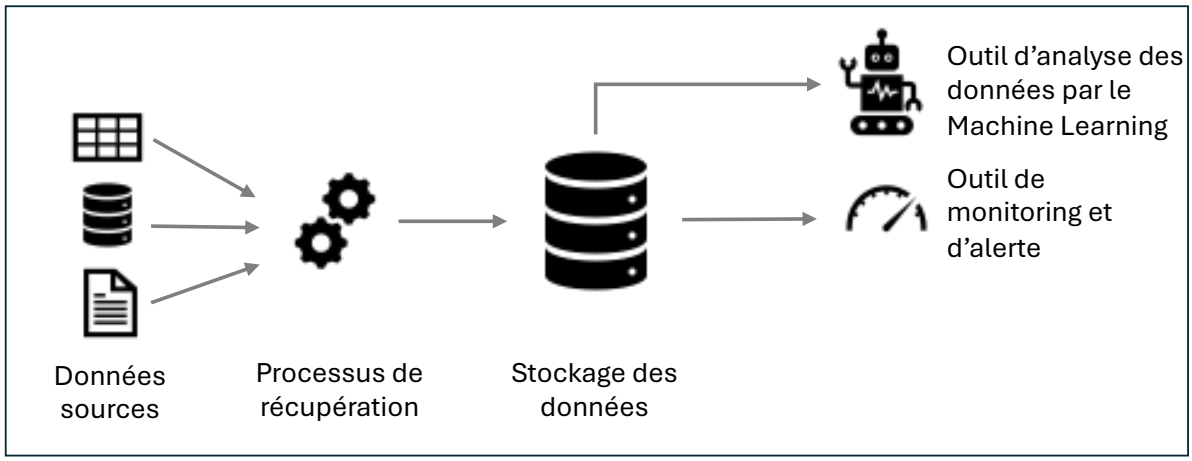

Figure 2 Schéma conceptuel de la solution « Smart ETL »

3.2. Problématiques & enjeux du projet

Des problématiques fortes se dégagent à partir de notre schéma fonctionnel de la solution Smart ETL et sont listées ci-dessous.

Comment collecter et stocker les données des logs ETL ?

Primever n'utilise pas de solutions Cloud et dispose de sa propre infrastructure dite *on-premise*. Par exemple, l'entreprise possède son propre serveur pour la collecte et le stockage des données.

Tout d'abord, nous devons être capable de collecter les données de logs ETL issues de leur infrastructure simplement et de les stocker durablement. C'est un enjeu important du projet car toute l'architecture de l'application se basera sur cette structure.

Cette collecte, qui constitue notre propre système ETL, se base sur le langage Python et s'effectue avec des *scripts* appropriés pour importer, transformer et charger les données. Ces données sont stockées dans une base de données relationnelle requêtable en SQL, langage maîtrisé par l'ensemble du groupe.

Comment analyser ces données ?

Primever souhaite comprendre la signification des logs et des pipelines de données associées mais surtout comprendre quand et comment elles dysfonctionnent. Le choix de la méthode pour mener à bien cette analyse est cruciale afin d'obtenir des résultats exploitables. Dans le cas présent, ces logs seront traités avec des librairies Python telles que Pandas et visualisés avec Matplotlib.

Comment obtenir un flux de données continu et comment les monitorer ?

Nous devrons également créer une infrastructure permettant la collecte des données en continu en s'appuyant sur des outils complets, par exemple avec Apache Airflow. Un second outil nous permettra de superviser ces données en temps réel et de créer un système d'alerte pour renforcer la valeur ajoutée au quotidien pour les équipes de Primever.

Comment mettre en place un modèle d'IA automatisé et déployer l'application ?

Pour mettre en place une analyse des données automatisée avec du *Machine Learning*, il faut une approche interdisciplinaire qui combine les automatisations du DevOps avec celui du *Machine Learning* : le MLOps.

Dans le cadre du projet nous allons mettre en œuvre certaines briques du DevOps comme la containerisation pour faciliter les développements et le déploiement. Nous mettrons également en place des *pipelines* de CI/CD afin d'automatiser la génération d'image Docker dans la Dépôt d'images de conteneur dans GitLab. Cela facilitera le déploiement sur une plateforme de test pour les présentations à l'entreprise.

Chaque livrable de ce projet, détaillé dans Identification des livrables et aperçu de leurs contenus, répond à un des enjeux cités ici, afin que l'application complète soit stable et robuste.

4. Environnement de travail (matériel et logiciel)

4.1. Matériel

Pour mener à bien notre projet, nous disposons de plusieurs ressources.

Notre outil de travail principal restera nos machines personnelles : chacun des membres de l'équipe dispose de son ordinateur portable pour travailler. En complément, ayant 3 membres de l'équipe au sein de MS SIO, nous avons également 3 postes de travail (mac mini, écran 27 pouces, clavier, souris) de la salle du MS SIO. Physiquement, nous pourrons donc utiliser la salle du MS SIO ou d'autres salles de Centrale Supélec pour nos sessions de travail en groupe.

4.2. Logiciel

Sur le plan logiciel, nous avons sélectionné plusieurs outils pour travailler.

Pour la communication :

- **Notion** : initialement un outil *no-code* de productivité, cette application *Freemium* propose un espace de travail collaboratif pour la gestion des notes, des tableaux, des tâches, des projets, ainsi que des documentations internes. Il est entièrement personnalisable et accessible facilement par navigateur Web. Nous l'utilisons pour partager et stocker des notes et des documents qui ne sont ni du code ni de la documentation sur le code. Nous l'utilisons également pour effectuer le suivi des tâches (voir Gestion des tâches avec Kanban).
- **Suite Microsoft Office** : cette suite applicative est bien connue du grand public. Nous utilisions certaines de ses applications, à savoir :
    - **Teams** : pour les réunions en visioconférence avec Primever ;
    - **Word** : pour la rédaction de documents, qui peut se faire à plusieurs, en synchrone ou asynchrone, en travaillant en ligne ;
    - **Outlook** : pour communiquer par mail en asynchrone avec Primever.
- **WhatsApp** : cette application mobile/*desktop* de Meta nous permet de communiquer facilement. Elle nous sert de canal de messagerie interne à notre groupe.

Pour le DevOps :

- **Gitlab CentraleSupélec** : plateforme de développement logiciel open source qui combine les fonctionnalités de gestion de code source, d'intégration continue (CI), de livraison continue (CD), de sécurité et de collaboration. Elle est conçue pour aider les équipes de développement à gérer leur cycle de vie DevOps de manière efficace et sécurisée. Nous l'utilisons pour partager et stocker du code et de la documentation uniquement.
- **Docker** : outil permettant de créer, déployer et exécuter des applications à l'aide de conteneurs. Cela permet de faire abstraction des différents environnements pour des applications s'exécutent de manière cohérente sur n'importe qu'elle machine.

Pour le développement, l'analyse de données et la *data science* :

- **IDE** : environnement intégré de développement, chacun de nous utilise l'IDE de son choix pour coder.

- **Python** (libraires, notebooks) : nous avons choisi Python comme langage de programmation pour plusieurs raisons. Premièrement, c'est un langage que nous connaissons et que nous avons pu pratiquer dans le cadre du MS SIO. Deuxièmement, c'est un des langages le plus utilisé actuellement. Enfin troisièmement, la possibilité de travailler avec des notebooks est une autre force de Python. Ce troisième point nous a permis d'explorer les logs et de rendre le code plus accessible pour Leyla.
- **SQL** (PostgreSQL, PgAdmin) : nous avons choisi une base de données SQL initialement car nous n'avions pas de contraintes/besoins qui auraient nécessitée du NoSQL. Nous avons opté pour PostgreSQL comme serveur et PGAdmin4 comme client car Manon avait déjà travaillé avec ces outils.

5. Organisation

5.1. Organisation et rôles au sein de l'équipe

Dans le cadre de ce projet, nous avons adopté une méthodologie **Agile** pour structurer notre travail de manière itérative et collaborative. Cette méthodologie nous permet de rester flexibles face aux besoins évolutifs du projet tout en maintenant un cadre rigoureux pour la planification et l'exécution des tâches.

5.2. Rôles au sein de l'équipe

Pour garantir une organisation efficace, chaque membre de l'équipe s'est vu attribuer des responsabilités spécifiques, comme illustré dans le la figure 3. Ces rôles peuvent évoluer dans le temps en fonction de nos besoins.

Rôle	Antoine	Guillaume	Leyla	Manon
Scrum Master / Chef de Projet		X		X (Adjointe)
Développeur	X	X		
Analyste des données	X	X	X	X
Administrateur base de données			X	X
DevOps	X		X	X
Support Utilisateur	X	X	X	X

Figure 3 : Répartition des rôles au sein de l'équipe

Description des rôles :

- ***Scrum Master* :** supervise la méthodologie Agile, définit les grandes fonctionnalités du *backlog*, assure le bon déroulement des sprints et sert de principal point de contact entre l'équipe et l'entreprise.
- ***Scrum Master* adjointe :** soutient et apporte un appui au Scrum Master dans ses responsabilités.
- **Développeur :** conçoit, code, intègre les outils nécessaires et développe les fonctionnalités.
- **Analyste des données :** analyse les historiques des ETL en utilisant Python, identifie les anomalies et les tendances pour orienter les décisions. Responsable de la prédiction à l'aide de Machine Learning.
- **Administrateur base de données :** administre la base de données en utilisant PostgreSQL et vérifie la qualité des données.

- **DevOps :** déploie et surveille l'infrastructure des ETL et des modèles en utilisant Docker, GitLab CI/CD, et des outils de *monitoring*.
- **Support Utilisateur :** Rédige les supports documentaires et/ou les livrables rédactionnels.

Nous pouvons aussi considérer que Balthazar MÉHUS agit comme ***Product Owner***. Il détermine les priorités et prend les décisions stratégiques pour orienter le projet, et valide les livrables à chaque étape.

5.3. Gestion des tâches avec Kanban

Pour organiser et suivre nos activités, nous utilisons un tableau Kanban qui nous permet de :

- Planifier les sprints en remplissant le *backlog* ;
- Visualiser l'état des tâches à travers les colonnes « À faire », « En cours », « À tester » et « Terminé » ;
- Attribuer des responsabilités à chaque membre selon ses domaines de compétence ;
- Fixer des échéances claires pour chaque tâche afin de respecter les délais convenus.

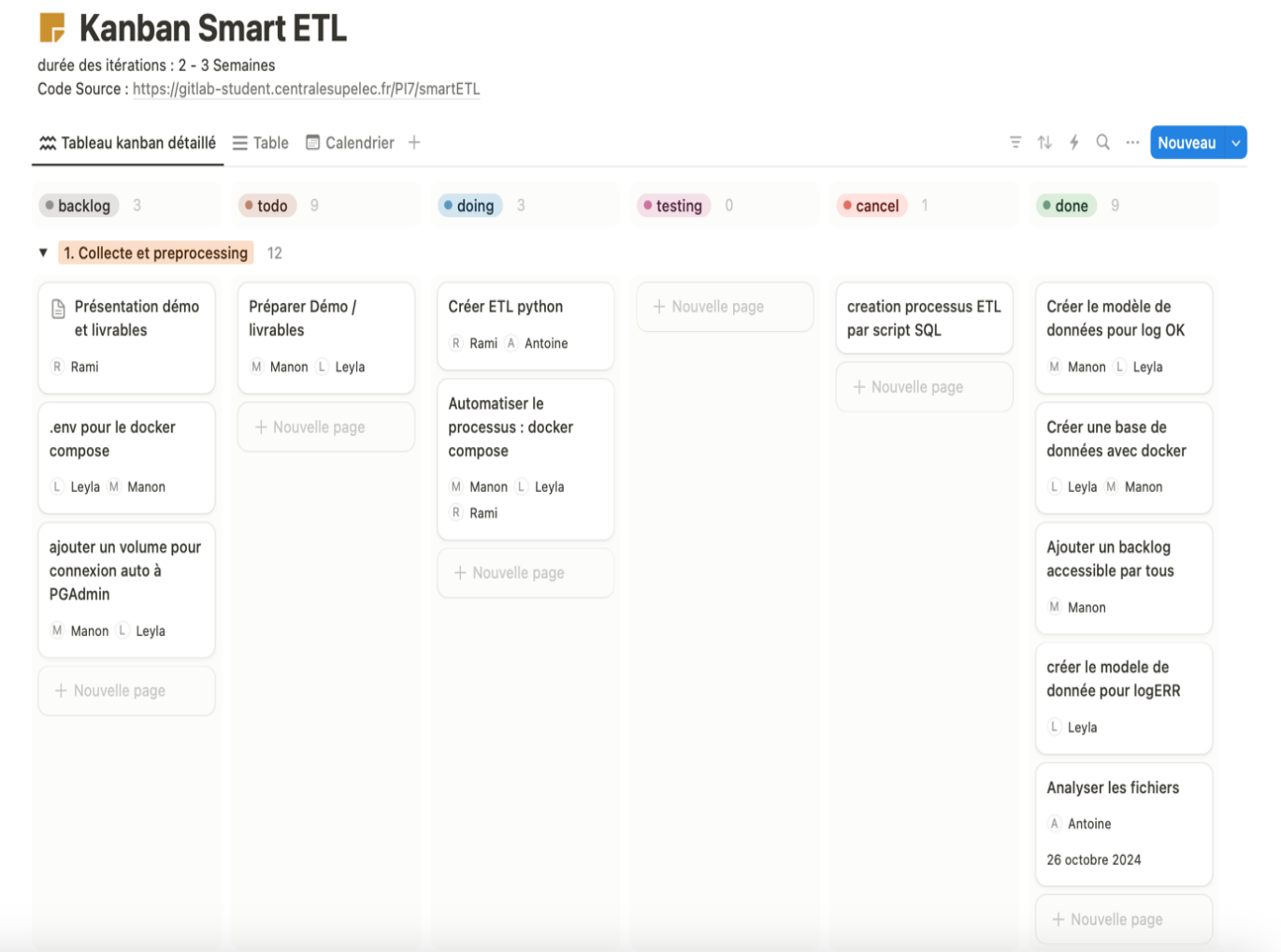

Figure 4 : Tableau de suivi des Kanban

5.4. Sprints, Livrables & Cérémonies

Sprint : Nous avons structuré notre travail en sprints de trois semaines, qui incluent les phases de conception, développement, tests et mise en production.

Livrables : Nous avons défini des livrables avec l'aide de Primever. Ces livrables sont à voir comme des *Milestones* et sont détaillées dans Identification des livrables et aperçu de leurs contenus.

Réunions d'équipe : Les réunions quotidiennes (*Daily Meeting*) ne sont pas réalisables à cause de nos contraintes horaires académiques. À la place, nous organisons des réunions hebdomadaires tous les mercredis pour profiter de ce créneau qui est dédié au projet industriel dans notre emploi du temps.

Réunions avec le client : Nous organisons des réunions avec le client toutes les deux à trois semaines. Ces points permettent de présenter les progrès réalisés, identifier les obstacles rencontrés, et ajuster les priorités en fonction des besoins du projet.

6. Identification des livrables et aperçu de leurs contenus

Tout au long du projet, nous prévoyons la remise de quatre livrables principaux, en cohérence avec la fiche projet proposée par le client.

6.1. Premier livrable : Collecte et pré-processing

A partir de la collecte des logs issus de l'activité de Primever, nous devons être capable de nettoyer et structurer ces données, afin de pouvoir les analyser. L'enjeu du premier livrable est donc de définir quelles transformations à appliquer sur la donnée afin de pouvoir facilement l'exploiter et la stocker.

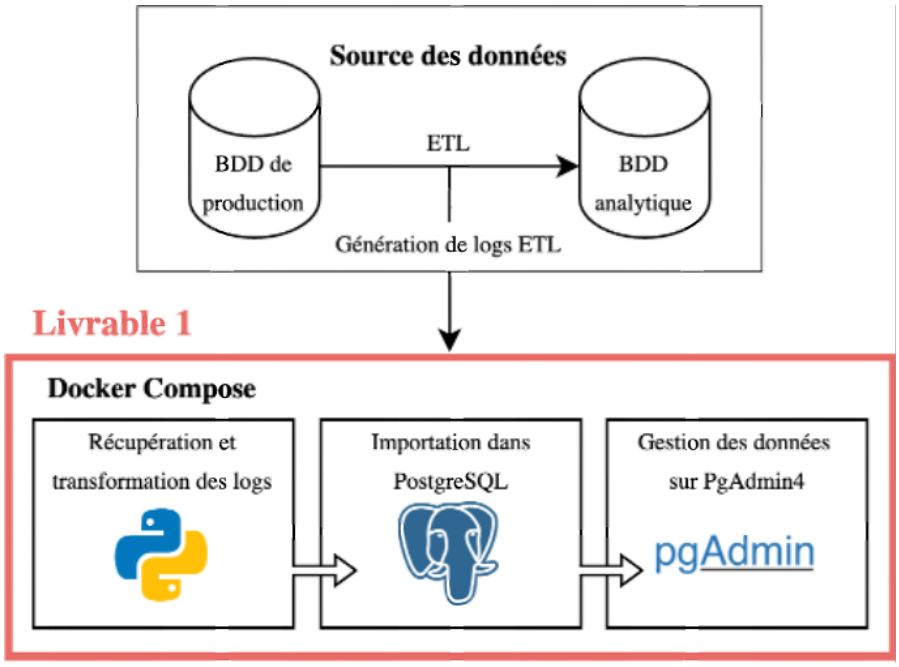

Figure 5 : Schéma du livrable 1

Ce premier livrable aura la forme d'une base de données prête pour l'analyse. Nous transmettrons tout l'environnement et ses variables, ainsi que le code nécessaire à l'accès à la base de données structurée.

Concrètement, les données transmises étant sous la forme de fichiers .csv, nous avons fait le choix de les transformer via des scripts Python avant de les charger dans une base PostgreSQL. De plus, nous avons choisi PgAdmin4 comme interface cliente.

Pour éviter des soucis d'environnements différents sur différentes machines, nous avons choisi de conteneuriser ce travail avec Docker. Le rendu final de ce livrable sera donc un fichier Docker compose qui gèrera tout le processus d'importation et de création de la base de données, avec des volumes pour la persistance de la base de données.

Lot	Tâche	Description	Critère de validation	Degré d'avancement
1 - Nettoyage et structuration des données	1 – Analyse des fichiers CSV	Identifier les types de données, les anomalies, et les champs clés.	Rapport documenté des anomalies et des transformations nécessaires.	100%
	2 – Transformation des données	Créer un script Python pour nettoyer et transformer les fichiers CSV.	Les fichiers transformés doivent respecter le schéma défini.	100%
2 - Mise en place de PostgreSQL	1 – Définition du schéma de la base de données	Créer une structure adaptée pour stocker les logs ETL.	*Script* SQL du schéma validé par l'équipe.	100%
	2 – Importation des données dans PostgreSQL	Développer un *script* Python pour automatiser l'import des données transformées.	Données disponibles dans PostgreSQL avec intégrité garantie.	100%
3 - Conteneurisation	1 – Configuration Docker	Conteneuriser PostgreSQL, PgAdmin, et les *scripts* Python.	Docker Compose opérationnel pour déployer la base de données et les outils associés.	100%

Figure 6 : Tableau récapitulatif du livrable 1

6.2. Deuxième Livrable : Analyse des données

Sur la base du premier livrable, nous aurons la matière nécessaire pour démarrer une analyse approfondie des données. À l'aide des logs issus des ETL (structurés dans une base de données) ainsi que les performances des serveurs *on-premise* du client (que nous avons récupéré par la suite), nous serons en mesure de mener une analyse descriptive des ETL existants.

L'objectif de cette phase est d'aboutir à un rapport détaillé de l'historique des logs ETL. Seront identifiés les patterns récurrents, les erreurs communes, les requêtes plus couteuses en termes de ressources pour le serveur, etc.

Le document principal de ce livrable prendra la forme d'un fichier PDF reprenant la méthodologie d'analyse, les différents supports d'analyses descriptives, les principaux résultats et la description de ceux-ci. La dernière partie se focalisera sur une synthèse des problèmes identifiés, les axes d'amélioration et les recommandations classées par priorité et faisabilité.

Les annexes de ce livrable seront des requêtes SQL et surtout des *notebooks* Jupyter. Ces notebooks contiendront des graphiques et autres outils d'analyses. Ils s'appuieront sur les librairies Pandas ou Matplotlib. Enfin, un outil de BI (*Business Intelligence*) sera utilisé pour créer une première version, simple, de *dashboard* (probablement semi-statique). Ce dernier sera ajouté au Docker Compose.

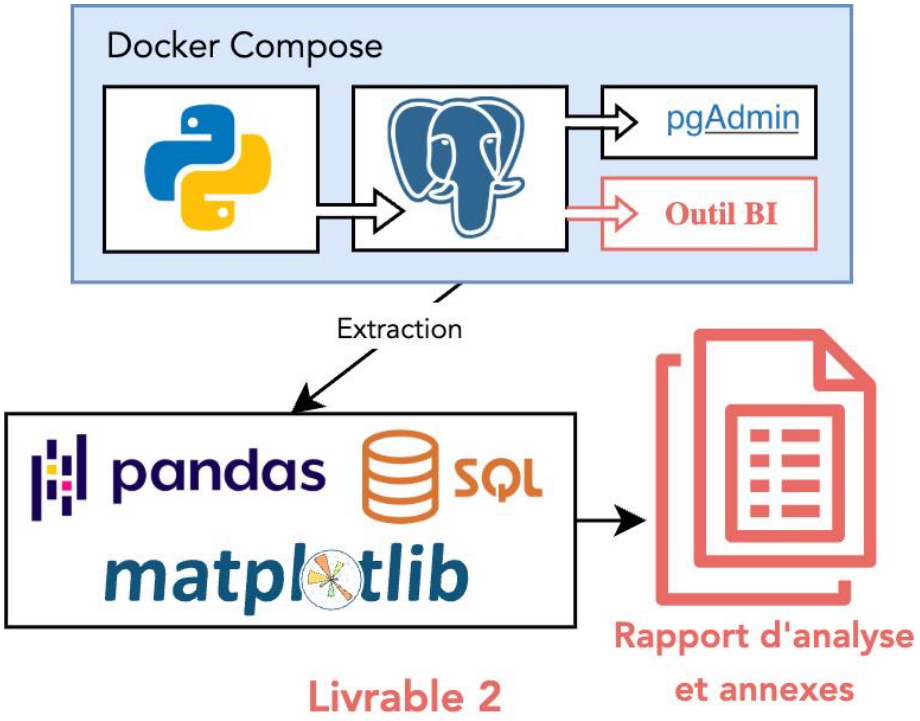

Figure 7 : Schéma du livrable 2

Lot	Tâche	Description	Critère de validation	Degré d'avancement
1 - Analyse exploratoire des données	1 – Identification des *patterns* et anomalies	Utiliser Python (Pandas, Matplotlib) pour analyser les tendances et les erreurs fréquentes.	Graphiques et tableaux des *patterns* principaux documentés.	0%
	2 – Analyse des performances des ETL	Étudier les temps d'exécution, les pics d'utilisation des ressources.	Résultats analysés et synthétisés dans un fichier PDF.	0%
2 - Production du rapport	1 – Rédaction du rapport	Intégrer les résultats, méthodologie, et recommandations.	Rapport PDF validé par l'équipe et le client.	0%
	2 – Compilation des annexes	Inclure *scripts* Python, requêtes SQL, et extraits de données.	Annexes complètes et bien organisées.	0%
3 - Dashboard initial	1 – Proposition de dashboard	Créer un prototype avec un outil de BI pour visualiser les analyses.	*Dashboard* fonctionnel intégré dans Docker Compose.	0%

Figure 8 : Tableau récapitulatif du livrable 2

6.3. Troisième Livrable : Monitoring et Alerting

L'étape suivante du projet sera la livraison d'un système de *monitoring* des processus ETL en temps réel. Les objectifs de ce livrable sont multiples : éviter un ralentissement système, détecter une erreur dans la transmission des données, détecter un *batch* incomplet ou encore alerter sur un évènement critique.

Pour résumer, en prenant appui sur le rapport d'analyse et les annexes issues du livrable précédent, nous serons en mesure de définir des mesures et/ou indicateurs plus approfondis pour établir un *monitoring* pour ces processus. Enfin, en partant de la première version du dashboard du deuxième livrable, nous mettrons alors un en place la version finale du *dashboard* pour le suivi des ETL.

En parallèle, il sera nécessaire de collaborer avec Primever pour identifier un moyen d'obtenir leurs données en continu. L'intégration de ce flux continu de données permettra un suivi en temps réel (ou avec un délai faible) permettant d'alerter en cas de dysfonctionnement. La difficulté de ce livrable sera de réussir à réaliser cette opération de *monitoring* en temps réel.

Pour se faire, nous utiliserons des outils tels que Kibana ou Grafana. De plus, nous pouvons imaginer faire un envoi d'alerte via la messagerie Teams ou encore par mail. Quoiqu'il en soit, notre solution devra être compatible avec le *stack* technologique de Primever. Elle sera donc conteneurisée afin d'être déployable facilement sur une machine virtuelle. Le livrable prendra

donc la forme d'une application complètement packagée sur Docker, dont les *Dockerfiles*, dépendances, Docker compose et tout autre élément seront transmis.

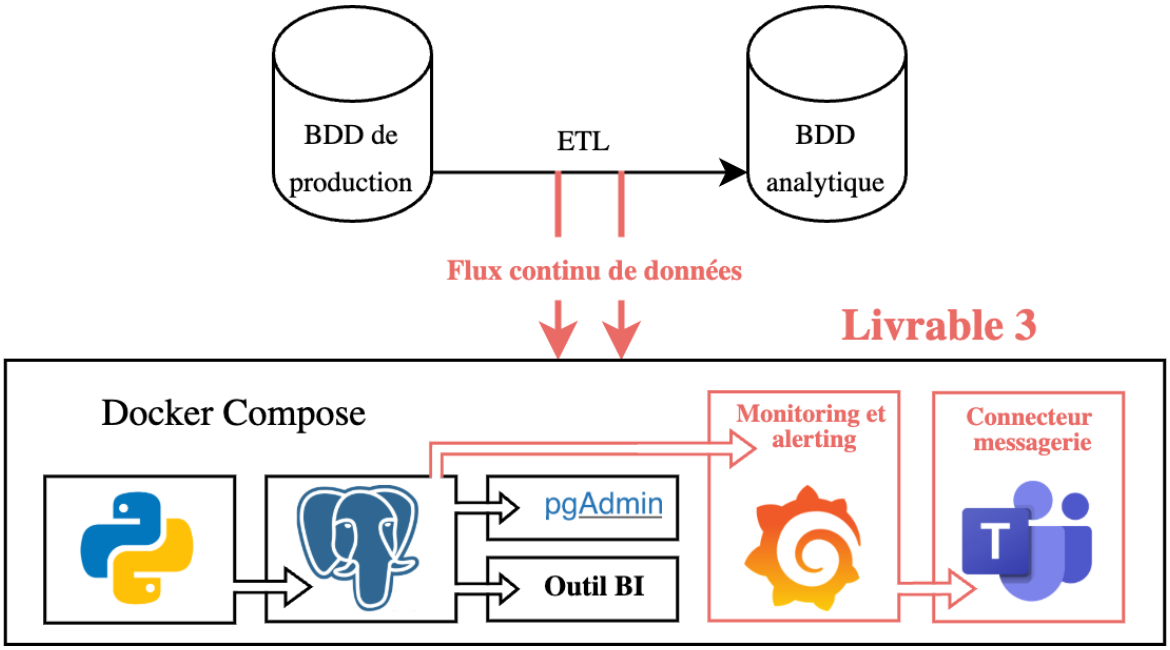

Figure 9 : Schéma du livrable 3

Lot	Tâche	Description	Critère de validation	Degré d'avancement
1 - Intégration d'un flux de données continu	1 – Configuration du système de streaming	Intégrer un outil pour capter les données en temps réel.	Flux de données stable entre les ETL et PostgreSQL.	0%
	2 – Adaptation des scripts Python	Modifier les *scripts* pour traiter les données entrantes en temps réel.	Données enrichies disponibles dans PostgreSQL.	0%
2 - Mise en place du *monitoring*	1 – Configuration de Grafana	Créer des tableaux de bord pour surveiller les processus ETL en temps réel.	*Dashboards* affichant les données en direct.	0%
	2 – Connexion avec Teams	Développer un connecteur pour envoyer des alertes sur Teams.	Alertes envoyées automatiquement en cas d'événement critique.	0%
3 - Conteneurisation	1 – Mise à jour du Docker Compose	Inclure les conteneurs Grafana et connecteur Teams.	Déploiement complet et opérationnel avec Docker Compose.	0%

Figure 10 : Tableau récapitulatif du livrable 3

6.4. Quatrième Livrable : Modélisation prédictive et intégration MLOps

L'objectif de ce dernier livrable sera d'intégrer un modèle de *Machine Learning (ML)* afin de prédire les erreurs probables à venir sur les processus ETL avant qu'elles ne se produisent. La cible sera d'identifier des processus et/ou les situations présentant un risque élevé d'incident avant qu'elles ne se produisent.

Il faudra réfléchir à l'intégration d'un *workflow* MLOps automatisé, avec une fréquence à établir, afin de gérer l'exécution du modèle MS. Il est possible que ce travail nécessite d'exposer une ou des API, via FastAPI par exemple, afin de faire communiquer la base de données, l'outil de *monitoring* ou encore le *workflow* MLOps.

Le livrable final reprendra tous les éléments précédents pour former une application complètement conteneurisée et fonctionnelle. Elle prendra la forme du schéma ci-dessous. Bien que la conteneurisation simplifie cette phase déploiement, il est possible que des petites modifications soient apportées à ce schéma fonctionnel.

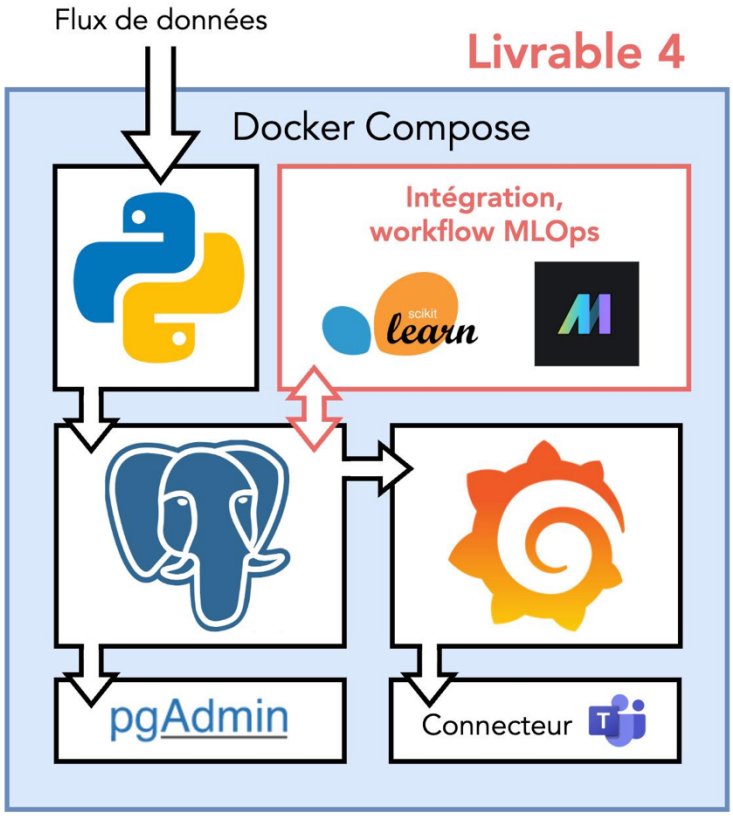

Figure 11 : Schéma du livrable 4

Lot	Tâche	Description	Critère de validation	Degré d'avancement
1 - Développement du modèle prédictif	1 – Préparation des données d'entraînement	Extraire les données historiques depuis PostgreSQL et les préparer.	Jeu de données final prêt à l'emploi pour l'entraînement.	0%
	2 – Entraînement du modèle	Créer et entraîner un modèle avec scikit-learn.	Modèle performant validé sur un jeu de test.	0%
2 - Déploiement et intégration	1 – Conteneurisation du modèle	Exposer le modèle via une API (FastAPI) dans un conteneur Docker.	API fonctionnelle répondant aux requêtes en temps réel.	0%
	2 – Intégration avec le flux de données	Connecter le modèle au flux pour traiter les données en temps réel.	Prédictions disponibles et stockées dans PostgreSQL.	0%
3 - *Workflow* MLOps	1 – Automatisation de l'entraînement	Mettre en place un *pipeline* MLOps pour entraîner le modèle périodiquement.	Réentraînement automatisé fonctionnel.	0%
	2 – *Monitoring* du modèle	Suivre les performances du modèle (*drift*, qualité des prédictions) avec Grafana.	Alertes déclenchées en cas de dérive des performances.	0%

Figure 12 : Tableau récapitulatif du livrable 4

7. Problèmes rencontrés & solutions

Malgré une organisation bien définie et un projet bien découpé en *milestone* avec des livrables, nous avons rencontré quelques obstacles. Ces difficultés ont nécessité des ajustements et une réflexion collective pour garantir la progression des travaux.

7.1. Manque de données

Dès la réception des premiers logs, nous avons débuté une analyse exploratoire des données. Nous nous sommes vite aperçus que nous n'avions pas suffisamment de données pour le livrable 2. Après échange avec Primever, l'équipe *Data* a abouti au même constat et nous a envoyé, quelques semaines plus tard, des nouvelles sources de données. Ainsi, notre capacité d'anticipation nous a permis d'éviter de perdre du temps plus tard pendant notre projet.

7.2. Gestion du temps

La charge de travail importante liée au programme du mastère a rendu difficile la conciliation entre les cours et le temps à consacrer au projet industriel. En conséquence, nous nous répartissons les tâches en prenant soin de prendre en compte les périodes de forte charge académique, permettant à chacun de contribuer selon ses disponibilités. Nous essayons également de prioriser les tâches apportant le plus de valeur pour les sprints, afin d'assurer des avancées significatives même pendant les semaines chargées.

7.3. Différence d'emploi du temps entre deux programmes MS SIO et MS MSI

Étant donné que nous provenons de deux mastères différents – MS SIO pour Guillaume, Manon et Antoine et MS MSI pour Leyla – avec des emplois du temps qui ne coïncident pas toujours, la coordination se fait principalement lors des créneaux dédiés au projet industriel. Ce fonctionnement est d'autant plus facile grâce à la planification des réunions de travail et le découpage du travail en sprint. Nous restons aussi flexibles face aux changements pour garder une capacité d'adaptation.

7.4. Estimations incorrectes des efforts nécessaires pour certaines tâches

Certaines tâches, notamment liées à l'intégration de technologies spécifique ou à l'analyse des données, ont nécessité beaucoup plus de temps et d'efforts que prévu initialement. Pour éviter une surcharge et/ou des retards, nous avons su aborder le problème lors des réunions hebdomadaires pour trouver une solution collectivement. Cela a parfois nécessité de revoir les *sprints* ou de réaffecter certaines personnes sur des tâches plus conséquentes.

8. Analyse des risques

Afin de structurer notre analyse des risques, nous avons choisi de nous inspirer d'un outil issu de l'industrie, l'AMDEC. L'AMDEC (Analyse des Modes de Défaillance, de leurs Effets et de leur Criticité) est une méthode d'analyse préventive utilisée pour identifier et évaluer les risques potentiels liés à des défaillances dans un produit, un processus, ou un système. Son utilisation a même été étendu à des projets ou encore des organisations [4].

En contexte industriel, l'AMDEC consiste à recenser les modes de défaillance possibles, à en analyser les causes et les effets, puis à évaluer leur criticité. Appliqué à une organisation ou un projet, il s'agit surtout d'identifier les risques majeurs et les classer selon une échelle commune: la criticité. Il s'agit du produit de 3 indices :

- **Indice de fréquence (F) :** Fréquence ou probabilité d'apparition
  - 1 : peu fréquent/probable
  - 10 : très fréquent/probable
- **Indice de gravité (G) :** Impact du risque s'il se réalise
  - 1 : peu d'impact
  - 10 : remise en cause de la viabilité du projet
- **Indice de détection (D) :** Possibilité de non-détection
  - 1 : détectable quasi-systématiquement
  - 10 : probabilité de détection faible

Risque	F	G	D	C	Solution préventive	Solution curative/palliative
Tension au sein du groupe	3	8	5	120	Communiquer et être à l'écoute. Respecter l'avis de chacun. Définir des valeurs communes à respecter au sein du groupe.	Désigner un médiateur, interne ou externe au groupe, pour désamorcer la situation et rétablir un climat constructif au sein du groupe.
Démotivation du groupe	3	6	5	90	Choisir des tâches qui nous intéresse. Maintenir une bonne ambiance au sein du groupe. Rester soudé et ne pas laisser quelqu'un à l'écart.	Évoquer les éventuelles baisses de motivation et réagir. Utiliser le collectif comme force.
Manque de données	6	5	2	60	Demander d'avoir les données dès les premières semaines du projet.	Générer un jeu de données factice en dernière mesure.
Non gratuité inattendue d'un outil utilisé	2	8	2	32	Privilégier des outils open source.	Changer d'outil dans la mesure du possible.
Perte de contact avec l'entreprise	3	9	1	27	Avoir des points réguliers avec l'entreprise. Avoir le contact de plusieurs personnes internes au cas où le contact privilégier ne serait plus joignable.	S'appuyer sur la direction de Centrale en cas de problème majeur.
Vol de données et/ou de matériels	4	5	1	20	Enregistrer notre travail sur des espaces collaboratifs (Gitlab, Notion). Faire des sauvegardes supplémentaires deux fois par mois en désignant un responsable sauvegarde.	Aucune.
Absence prolongée d'un membre du groupe	3	6	1	18	Bien répartir les tâches pour éviter qu'une personne soit seule sur une dimension du projet.	Réaffecter les tâches rapidement.
Perte du travail réalisé	1	10	1	10	Enregistrer notre travail sur des espaces collaboratifs (Gitlab, Notion). Faire des sauvegardes supplémentaires deux fois par mois en désignant un responsable sauvegarde.	Aucune.
Faillite de l'entreprise	1	10	1	10	Aucune.	S'appuyer sur la direction de Centrale en cas de problème majeur.

Figure 13 : Tableau des risques

Cette liste de risques n'est sans doute pas exhaustive, mais elle permet déjà d'identifier clairement les principaux risques de notre projet industriel. Nous remarquons notamment que les plus gros risques identifiés sont avant tout lié à la cohésion du groupe et la motivation du groupe : ceux sont des risques humains. Les autres risques sont des risques techniques ou organisationnels que dont l'impact peut être limité avec des actions préventives et/ou des actions correctives.

9. Conclusion

Le projet industriel du groupe Primever démontre bien l'importance cruciale d'une gestion intelligente des données pour maintenir l'excellence opérationnelle, en particulier dans un secteur aussi exigeant que la logistique des produits périssables.

Ce projet, intitulé **« Smart ETL »,** vise premièrement à analyser les pipelines ETL de Primever afin d'optimiser et fiabiliser ces processus critiques au bon fonctionnement de l'entreprise. Deuxièmement, le projet porte aussi sur la mise en place d'une solution sur mesure pour le monitoring des ETL en temps réel et leur suivi grâce à un *dashboard*. Enfin, le projet a également pour ambition d'utiliser du *Machine Learning* dans le but de prédire les erreurs et anomalies des ETL avant qu'elles ne se produisent.

Il s'agit donc de répondre à des besoins concrets du groupe Primever. Pour se faire, nous avons décomposé le travail à réaliser en 4 livrables et nous avons imaginé une architecture fonctionnelle capable de répondre aux objectifs ambitieux du projet. Le contenu de chaque livrable a été validé par l'équipe *Data* de Primever et nous sommes donc bien alignés avec elle.

Nous avons également identifié plusieurs risques et mis en avant plusieurs problèmes rencontrés pendant les premières semaines de ce projet. Les solutions et/ou actions pour s'en protéger ont d'ailleurs été explicitées dans ce document. Enfin, nous avons posé le cadre de notre organisation et lister les moyens qui nous sont disponibles. Nous avons donc toutes les clés en main pour accomplir notre travail sur notre feuille de route.

Bien que les résultats soient prometteurs, le projet devra relever certains défis, notamment en ce qui concerne le *monitoring* en temps réel, le déploiement ou encore la précision des modèles prédictifs. Bien que nous n'ayons pas encore toutes les réponses, nous sommes confiants dans notre démarche et nous espèrerons que ce rapport aura rassurer son lecteur.

10. Table des figures

11. Références bibliographiques

[1] **“Extract-transform-load.”** *Wikipédia*, Wikimedia Foundation, 11 déc. 2024, https://fr.wikipedia.org/wiki/Extract-transform-load.

[2] **Alnoukari, Mouhib.** “Business Intelligence and Data Mining.” *International Journal of Information and Communication Technology Research*, vol. 2, no. 2, 2011, pp. 53-59. ScienceDirect, https://www.sciencedirect.com/science/article/pii/S131915781100019X.

[3] **Primever.** “Qui sommes-nous ?” *Primever*. https://www.primever.com/groupe/qui-sommes-nous/. Consulté le 25 novembre 2024.

[4] **Les Cahiers de l’Innovation.** “La méthode AMDEC : Analyse des risques.” *Les Cahiers de l’Innovation*. https://www.lescahiersdelinnovation.com/la-methode-amdec-analyse-des-risques/. Consulté le 25 novembre 2024.