l'école d'ingénierie informatique

MSPR 5

Analyse Météorologique

BENCHALAL Nael, LAVERIE Camille, LOUAILECHE Axel, MONIN Bastien

Table des matières

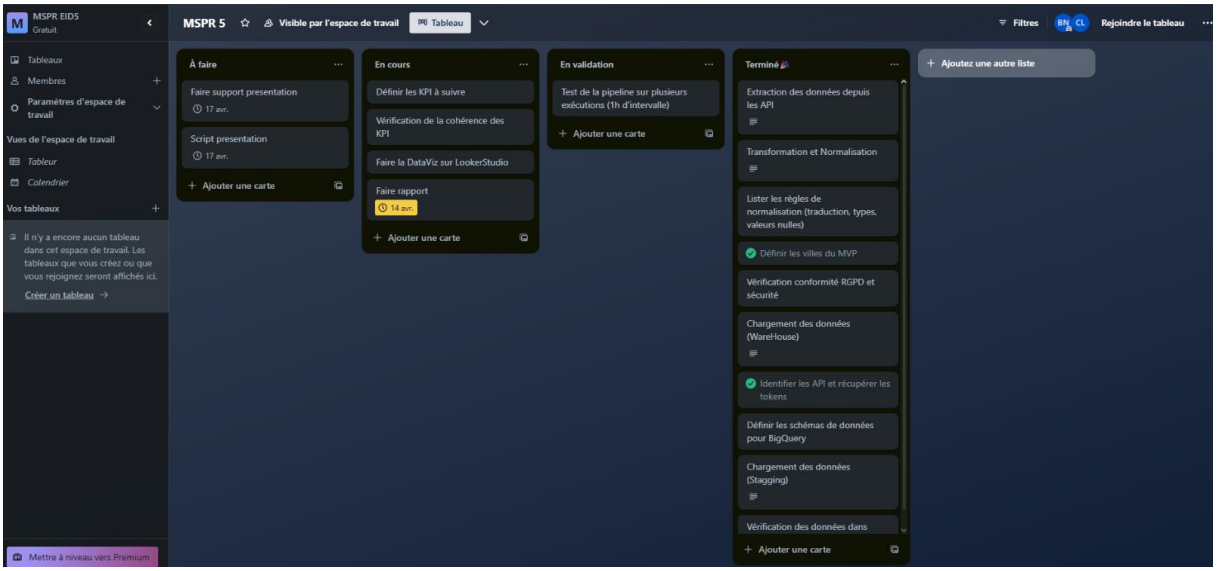
Introduction

GoodAir a pour mission d'analyser la qualité de l'air en France, afin de mieux comprendre les effets du changement climatique et de proposer des recommandations aux citoyens et aux institutions. Pour remplir cette mission, notre équipe souhaite mettre en place une plateforme capable de collecter, traiter, stocker et visualiser des données environnementales provenant de sources ouvertes fiables, tout en garantissant leur qualité, leur sécurité et leur conformité RGPD.

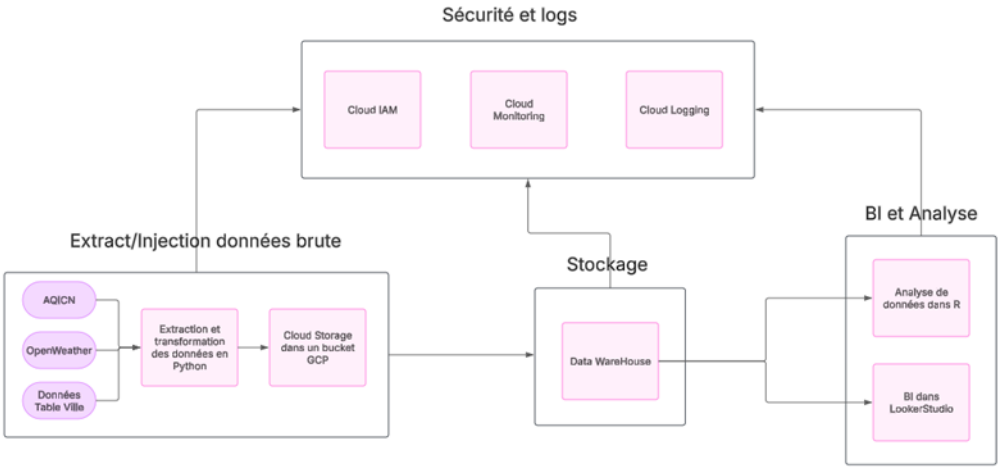
Nos objectifs :

- Mettre en place une architecture complète de gestion de données depuis la collecte jusqu'à la restitution,
- Exploiter les données pour fournir des rapports pertinents aux chercheurs (qualité de l'air, météo, corrélations...),
- Proposer des analyses prédictives futures (modélisation, machine learning),
- Automatiser le tout via une pipeline de traitement.

Dès le début, nous avons opté pour une méthode de travail Agile, en mettant en place un tableau Trello pour organiser les tâches et suivre l'avancement de manière collaborative. Cette approche nous a permis de travailler efficacement en équipe, de répartir les responsabilités selon les compétences de chacun et d'assurer un bon niveau de communication tout au long du projet.



Nous avons également réalisé un schéma explicatif via Lucidchart afin de représenter visuellement l'architecture et le flux de traitement des données de notre projet. Ce schéma met en évidence les différentes étapes clés, de la collecte à l'analyse, en passant par la sécurisation et le stockage :



1. Extraction et injection des données brutes

Les données sont récupérées via les API AQICN et OpenWeatherMap, ainsi qu'à partir d'une table contenant les villes ciblées. Ces données sont ensuite traitées par un script Python centralisé qui :

- Normalise les formats,
- Nettoie les valeurs manquantes ou incohérentes,
- Structure les données pour l'intégration future.

Le résultat est stocké temporairement dans un bucket Google Cloud Storage (GCS).

2. Stockage dans un entrepôt de données

Les données nettoyées sont ensuite chargées dans notre Data Warehouse basé sur BigQuery. Cela permet de centraliser l'ensemble des informations, d'assurer la traçabilité, et de préparer les données pour des analyses avancées.

3. Analyse et visualisation (BI)

Les données stockées sont exploitées à deux niveaux :

- Analyses statistiques via le langage R, pour effectuer des modélisations simples (régression linéaire, par exemple).
- Visualisations dynamiques à l'aide d'un outil de BI (initialement Looker Studio), connecté directement à BigQuery pour suivre les indicateurs environnementaux.

4. Sécurité et supervision

Pour garantir la sécurité et la conformité RGPD, trois briques sont mobilisées :

- Cloud IAM : gestion des accès par rôles, selon le principe du moindre privilège.
- Cloud Monitoring : surveillance de l'état de santé de l'architecture.
- Cloud Logging : journalisation des accès et traitements pour assurer une traçabilité complète.

Architecture de notre Projet

Afin de répondre aux besoins de la plateforme GoodAir, nous avons conçu une architecture de traitement de données complète et automatisée, reposant principalement sur les services cloud de Google Cloud Platform (GCP).

- Collecte des données : Les données sont récupérées toutes les heures via deux appels API pour chaque ville :
- La première API fournit les indicateurs de qualité de l'air (AQICN).
- La seconde donne les données météorologiques (OpenWeatherMap).

Extraction : Un script Python orchestre l'appel des API, l'extraction des données pertinentes, la normalisation des noms de colonnes, la gestion des types, la traduction en français, et le nettoyage (exclusion des valeurs nulles, homogénéisation des formats).

Stockage temporaire (staging) : Les données traitées sont ensuite envoyées dans un bucket Google Cloud Storage, qui sert d'espace de staging avant leur intégration finale dans l'entrepôt de données.

Chargement dans BigQuery : Une fois les données prêtes, elles sont automatiquement importées dans Google BigQuery, une base de données cloud orientée analytique, conçue pour traiter de gros volumes de données rapidement et efficacement.

Machine Learning (R) : Certaines analyses exploratoires, comme la régression linéaire sur l'évolution des températures, sont réalisées via le langage R, afin de démontrer le potentiel de modélisation des données collectées.

Visualisation des données : Un outil de data visualisation est connecté à BigQuery pour créer des dashboards interactifs.

Pipelines automatisés : Toutes ces étapes sont intégrées dans un pipeline automatisé, orchestré à l'aide de Cloud Composer, un service GCP basé sur Apache Airflow. Les DAGS gèrent le déclenchement automatique du traitement toutes les heures.

Collecte des Données

La première étape essentielle de notre projet a consisté à identifier et exploiter des sources de données ouvertes et fiables, disponibles via des API REST. Nous avons choisi deux sources complémentaires afin d'obtenir une vision complète des conditions environnementales dans les grandes villes françaises.

AQICN – Qualité de l'air

Nous utilisons l'API d'AQICN (aqicn.org), qui fournit en temps réel des indicateurs clés liés à la qualité de l'air (L'indice de qualité de l'air (AQI), les concentrations de polluants (PM2.5, PM10, NO2, O3, CO, SO2), la pression atmosphérique et l'humidité).

OpenWeatherMap – Météo

En complément, nous utilisons l'API d'OpenWeatherMap (openweathermap.org) afin de récupérer les conditions météorologiques au moment de la mesure (La température, la vitesse du vent, le taux d'humidité, le type de météo (pluie, brouillard, etc.)).

Nous avons ajouté une **troisième source** de données contenant une table Ville qui nous a donné des ressources donnant plus de détails sur les villes pour faciliter les futurs analyse.

Un script Python effectue l'appel aux deux API toutes les heures, pour chaque ville. Le processus est séquentiel :

1. Un premier appel est effectué vers AQICN pour obtenir les données de qualité de l'air.
2. Un second appel est réalisé vers OpenWeatherMap pour récupérer les données météorologiques correspondantes.

Traitement & Stockage des Données

Une fois les données récupérées via les API, elles sont immédiatement traitées dans un script Python centralisé. Ce script assure à la fois la collecte, la transformation, et la préparation des données, avant leur envoi vers l'infrastructure de stockage sur Google Cloud.

L'étape de transformation des données consiste à convertir les données brutes récupérées des APIs Air Quality et Weather Map en DataFrames Pandas structurés. Ces données sont organisées, nettoyées et normalisées en colonnes dans des DataFrames, facilitant leur analyse et leur sauvegarde ultérieure.

Une fois les données prêtes, elles sont déposées dans un bucket Google Cloud Storage (GCS), utilisé ici comme une zone de staging temporaire. Cela permet de conserver une copie brute "pré-chargement", utile pour la traçabilité ou les replays éventuels. Ensuite, elles sont chargées dans Google BigQuery, qui fait office de Data Warehouse principal du projet. Les données sont insérées de manière incrémentale, permettant la conservation d'un historique complet des mesures environnementales.

Le choix de ne pas créer de datamarts s'explique par le fait que les données, déjà filtrées et nettoyées lors du traitement initial, sont suffisamment structurées pour répondre aux cas d'usage métier identifiés. Cela allège l'architecture sans compromettre la pertinence analytique.

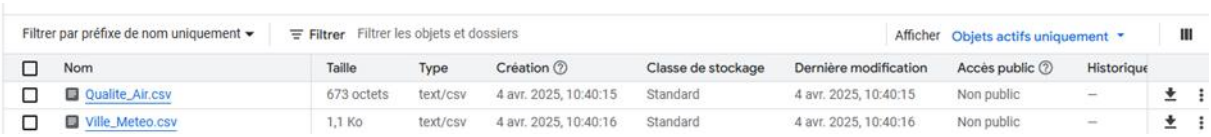
Automatisation du Pipeline de Traitement

Afin de garantir la collecte continue, le traitement horaire et la mise à disposition des données, nous avons mis en place une automatisation complète du processus à l'aide de Cloud Composer, le service managé de Google Cloud Platform basé sur Apache Airflow.

Nous avons créé un DAG principal nommé pipeline_qualite_air_weather, chargé d'exécuter le script Python de collecte et de transformation des données. Comme illustré dans l'interface Airflow ci-dessous, ce DAG est configuré pour s'exécuter toutes les heures. Il est composé de tâches simples et efficaces, chacune étant tracée et historisée dans l'interface

Le script Python utilisé (from_airflow_import_DAG.py) est stocké dans un bucket privé sur GCP, garantissant la sécurité et l'accessibilité par Cloud Composer :

Les fichiers CSV issus du traitement sont également déposés dans le même bucket, dans une logique de staging temporaire, avant d'être chargés dans BigQuery



Le dag " airflow_monitoring" a été mis en place pour surveiller la bonne exécution du pipeline principal :

- Il peut vérifier si `pipeline_qualite_air_weather` a bien tourné à l'heure prévue ;
- En cas d'erreur ou d'échec, une alerte peut être déclenchée (par exemple via email ou log d'erreur dans Cloud Logging).
- Observer et analyser les statistiques de notre dag.

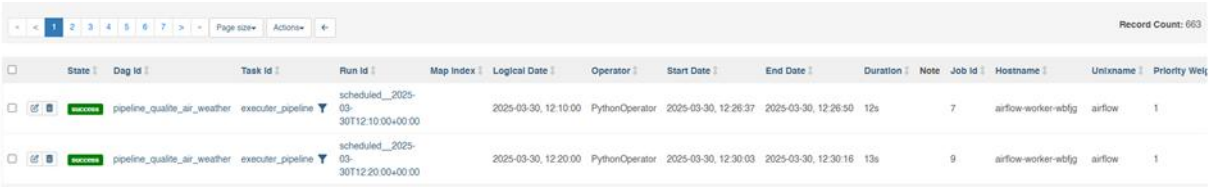
En raison de contraintes budgétaires, l'exécution automatique du pipeline a été temporairement mise en pause. Cependant, toute l'infrastructure est opérationnelle, et les DAGs peuvent être relancés à tout moment si besoin

Sécurité & Conformité RGPD

Conformément à l'article 44 et suivants du RGPD, toutes les données sont hébergées dans des datacenters situés dans l'Union Européenne (Belgique), en utilisant les services Google Cloud Storage et BigQuery, dont la conformité RGPD est assurée par Google Cloud (certifications ISO/IEC 27001, 27017, 27018).

Grâce à Cloud Identity and Access Management (IAM), chaque membre de l'équipe a reçu des permissions minimales strictement nécessaires à ses missions

Nous avons mis en place Cloud Logging et Cloud Monitoring afin de suivre toutes les actions sur les ressources sensibles (lecture, écriture, suppression) . Détecter toute activité inhabituelle ou non autorisée mais aussi recevoir des alertes automatisées via des règles de surveillance Airflow et GCP Monitoring, en cas de pic d'usage ou d'échec de DAG.



Le code source des scripts de traitement (Python et R) est hébergé dans un dépôt Git privé. L'accès est limité aux développeurs identifiés, et nous avons intégré GitGuardian pour détecter automatiquement toute fuite de secrets (ex : API keys, jetons d'authentification).

Dans notre cas, aucune donnée directement personnelle (nom, prénom, email, IP, etc.) n'est collectée.

Analyse Prédictive & Machine Learning

L'objectif de cette partie était d'aller au-delà de la description, pour proposer des modèles permettant :

- D'anticiper les phénomènes critiques (canicule, alerte pollution),
- D'aider les chercheurs à mieux comprendre les liens de causalité entre météo et pollution.

Modèles mis en œuvre :

Objectif	Modèle utilisé	Résultat
Prédire l'indice global AQI	Régression linéaire	RMSE et visualisation
Détecter les pics de pollution	Arbre de décision	Matrice de confusion + arbre
Anticiper les canicules	Classification `rpart`	Prédiction binaire
Détecter des anomalies	LOF (Local Outlier Factor)	Anomalies visuelles
Clustering des profils	K-means	3 groupes types identifiés
Prévision PM10	ARIMA	Prévision à 7 jours

Ce sont des méthodes simples, robustes, facile à interpréter par une équipe non spécialisée en data science et visuellement explicables avec des supports graphiques intégrés.

Les visualisations et les modèles mis en œuvre dans cette phase de la MSPR permettent à GoodAir :

- De suivre en temps réel l'état de l'air et les conditions météo,
- De détecter des situations à risque,
- De prévoir des événements à fort impact (canicule, pics de pollution),
- De disposer d'un socle de travail clair pour approfondir des études scientifiques.

Data Visualisation

Trois solutions s'offraient à nous quant à la data visualisation :

La première option était une solution moins coûteuse par la gratuité de son utilisation (Looker Studio), la seconde est une option plus coûteuse mais plus fiable et demandant moins de préparation pour les données (Power BI) et la dernière moins coûteuse que la seconde mais plus efficace mais demandant plus de compétence métier avec l'utilisation de R. Nous avons conçu plusieurs visualisations simples, claires pour répondre aux enjeux de compréhension métier, d'alerte, et d'analyse exploratoire.

1. Évolution temporelle des polluants

Nous avons voulu suivre la dynamique des indices de qualité de l'air dans le temps. Cela nous a permis de détecter visuellement des tendances, des hausses brutales ou des périodes problématiques par le biais d'une courbe linéaire par polluant, avec seuils d'alerte en fond (ex : AQI > 50).

2. Croisement Météo / Pollution

L'objectif a été d'analyser l'impact des conditions météo (température, humidité, vent) sur la pollution afin de comprendre les interactions climatiques et de pouvoir aider à anticiper les pics de pollution avec un nuage de points "Température vs PM2.5", coloré selon l'humidité ou la vitesse du vent.

3. Corrélations entre variables

Nous avons voulu identifier les facteurs qui influencent fortement la qualité de l'air afin de guider les futurs modèles prédictifs et l'interprétation des phénomènes grâce à une Fmatrice de corrélation avec corrplot, visualisation des coefficients R^2.

4. Clustering météo-pollution (K-means)

Nous avons voulu segmenter automatiquement les observations en profils types (pollution légère/modérée/forte) afin de catégoriser des journées types et d'aider à l'analyse sans supervision avec une projection PCA des clusters (via fviz_cluster) avec légende.

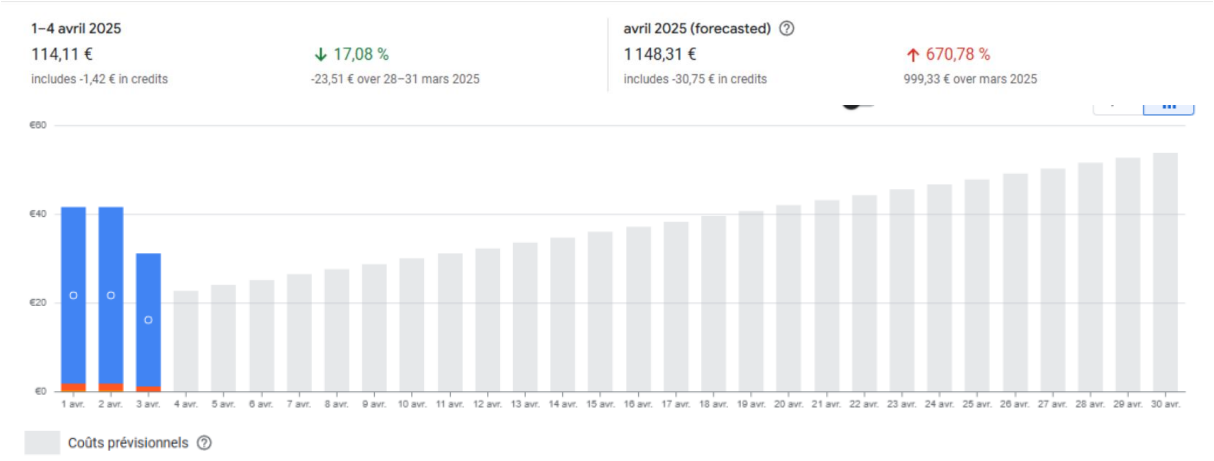
5. Détection d'anomalies (LOF)

Nous avons voulu repérer automatiquement les situations inhabituelles afin d'aider à détecter des erreurs de capteurs, ou des événements extrêmes (canicule, incendies…) avec un Scatter plot avec couleur qui détecte les anomalies.

Limites et Contraintes du Projet

Le projet que nous avons mené dans le cadre de cette MSPR propose une solution fonctionnelle et réaliste, mais reste soumis à un certain nombre de limites techniques et économiques. Il est important de souligner que cette architecture représente une proposition de mise en œuvre, construite dans un contexte pédagogique, avec des moyens restreints.

L'un des obstacles majeurs rencontrés a été le budget limité dont nous disposions pour accéder aux services cloud (notamment Google Cloud Platform). Le pipeline automatisé, bien que fonctionnelle, a dû être désactivée temporairement pour éviter des coûts de traitement récurrent. Nous avons fait le choix d'outils gratuits ou open source chaque fois que possible pour éviter les frais liés à des solutions professionnelles payantes.



Télécharger au format CSV

Service	Coût	Remises	Promotions et autres	Sous-total	Évolution en %
Cloud Composer	109,50 €	0,00 €	–	109,50 €	↓ 16%
Cloud Monitoring	3,96 €	0,00 €	–	3,96 €	↓ 32%
Cloud Storage	0,52 €	0,00 €	–	0,52 €	↓ 34%
Compute Engine	0,13 €	0,00 €	–	0,13 €	↓ 41%
Networking	1,41 €	-1,41 €	–	0,00 €	–
Cloud Run	0,00 €	0,00 €	–	0,00 €	–

Malgré ces contraintes, le système mis en place est opérationnel, automatisable et peut évoluer facilement dès que plus de ressources (humaines, techniques ou financières) seront disponibles.

Conclusion

Au terme de ce projet, nous pouvons affirmer que l'ensemble des exigences formulées dans le cahier des charges ont été respectées et traduites en une solution concrète, fonctionnelle et structurée.

Voici les **objectifs complétés** :

- Développé une phase complète de préparation des données, intégrant la collecte, le nettoyage, la validation et la structuration via des scripts Python robustes ;
- Récolté et intégré des données complémentaires pertinentes à partir de deux API spécialisées (qualité de l'air et météo), assurant un croisement intelligent et enrichi ;
- Conçu une architecture modulaire et évolutive, en exploitant les services de Google Cloud Platform (GCS, BigQuery, Cloud Composer), permettant un stockage sécurisé et une intégration fluide des données ;
- Assuré la qualité des données livrées aux métiers, via un traitement rigoureux, une normalisation des formats, et un monitoring des traitements (Airflow) ;
- Garantit la sécurité des données et leur conformité RGPD, grâce à un hébergement en Union Européenne, à l'authentification sécurisée.
- Proposé un outil de dataviz fiable, interconnecté avec BigQuery, permettant un accès intuitif et dynamique aux données, adapté aux attentes des utilisateurs métiers ;
- Anticipé les besoins d'analyse par la réalisation de dashboards dynamiques et d'une première modélisation statistique (régression linéaire en R), démontrant la valeur d'usage de ces données ;
- Enfin, nous avons démontré notre capacité à organiser et valoriser les données sous forme exploitable, à travers une démarche structurée allant du pipeline technique à la restitution métier.