

Projet Tutoré

M2 Bases de Données Intelligence Artificielle (BDIA)

**Étude et caractérisation de données disponibles dans la
base de données Open Food Facts**

Élèves :

Guillaume BELDILMI
Assia HAMMANI

Enseignants :

Nadine CULLOT
nadine.cullot@u-bourgogne.fr

2024-2025

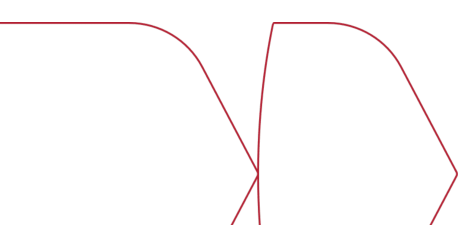


Table des matières

1	Remerciements	3
2	Introduction	4
2.1	Contexte et présentation d'Open Food Facts	4
2.2	Objectifs du projet	4
2.3	Environnement de travail	4
3	Pré-analyse et filtrage des données	7
3.1	Données disponibles	7
3.2	Structure des données	7
3.3	Méthodologie d'exploration des données	8
3.4	Exploration de données avec Pandas	8
3.5	Choix des colonnes redondantes	10
3.6	Choix des colonnes à ignorer	11
3.7	Reconstruction du schéma de données	12
3.8	Bilan de la pré-analyse	13
4	Modelisation	13
4.1	Modèle de données (BDR)	13
4.2	Diagramme UML	15
4.3	Description d'Analyse	16
5	Implémentation technique	16
6	Realisation du Notebook	18
6.1	Fonctionnalités du Notebook	18
6.2	Bibliothèques utilisées	18
6.3	Exploration initiale des données	19
6.4	Analyses nutritionnelles	19
6.5	Analyses environnementales	20
6.6	Fonctionnalités avancées	21
6.6.1	Prédiction du Nutri-Score à l'aide du Machine Learning	21
7	Annexes	23
8	Bibliographie	29

Table des figures

1	Utilisation de l'outil Trello pour la gestion de tâches	5
2	Résultats exploration de données	9
3	Tables base de données	14
4	Schéma ELT	17
5	Processus ELT	17
6	Diagramme UML	26
7	Visualisation des occurrences des données manquantes - Excel	28

1 Remerciements

Nous tenons à remercier notre encadrante Madame Nadine Cullot pour ses conseils, sa disponibilité et son accompagnement tout au long de ce travail.

Nous souhaitons également remercier l'ensemble de nos enseignants du Master, et tout particulièrement Madame Annabelle Gillet pour ses conseils avisés et notamment pour avoir répondu à nos questions sur un sujet spécifique.

2 Introduction

2.1 Contexte et présentation d'Open Food Facts

Le projet d'étude et de caractérisation des données d'Open Food Facts s'inscrit dans un contexte où l'accès à une information transparente et fiable sur les produits alimentaires devient essentiel pour les consommateurs, les autorités publiques et les entreprises.

Open Food Facts est une base de données collaborative qui regroupe des informations détaillées sur des milliers de produits alimentaires, permettant ainsi de mieux comprendre leur composition, leur impact sur la santé et l'environnement.

Ce projet vise à exploiter ces données pour en tirer des analyses significatives, mettant en évidence des tendances et des anomalies susceptibles d'éclairer les choix des consommateurs et de favoriser une consommation plus responsable.

2.2 Objectifs du projet

L'objectif du projet est d'analyser en profondeur les données disponibles dans le jeu de données **Open Food Facts**, mises à disposition par le gouvernement, afin de les caractériser et d'en extraire des informations pertinentes.

Le premier objectif consiste à étudier les ressources mises à disposition sur le portail **open-data** du gouvernement, en examinant les métadonnées, la documentation et les informations concernant la structure et la qualité des données. Cela permettra de mieux comprendre leur contexte et leurs limites avant de passer à l'analyse proprement dite.

Ensuite, il s'agira de concevoir un schéma de stockage optimisé des données, en définissant une ou plusieurs bases de données adaptées pour faciliter leur exploitation. Une fois cette étape réalisée, l'objectif sera d'identifier des critères d'analyse pertinents, tels que les caractéristiques nutritionnelles des produits ou l'impact environnemental, tout en explorant des possibilités d'utilisation des techniques d'intelligence artificielle pour des analyses plus avancées.

Enfin, le projet prévoit la création d'un notebook **Jupyter** qui présentera de manière interactive les résultats des analyses, en permettant aux utilisateurs de paramétrer facilement certains critères d'analyse pour répondre aux besoins spécifiques des spécialistes du secteur alimentaire. Ce notebook visera à rendre l'analyse des données accessible et utile, tout en favorisant une meilleure transparence et une utilisation éclairée des données dans l'industrie alimentaire. Ce notebook offrira donc la possibilité aux spécialistes de l'alimentaire et aux consommateurs d'explorer les données, de comparer les produits et d'obtenir des recommandations personnalisées basées sur différents critères.

2.3 Environnement de travail

Outils, technologies et ressources utilisés

Pour la gestion du projet, nous avons utilisé **Trello**, un outil de gestion de tâches basé sur la méthode Kanban. Trello nous a permis de visualiser l'état d'avancement des

différentes tâches grâce à des tableaux organisés en trois catégories : *À faire*, *En cours*, et *Terminé*. Cet outil a été essentiel pour suivre le progrès du projet, identifier les tâches en retard, et coordonner nos efforts.

Voici un exemple illustrant l'utilisation de l'outil Trello :

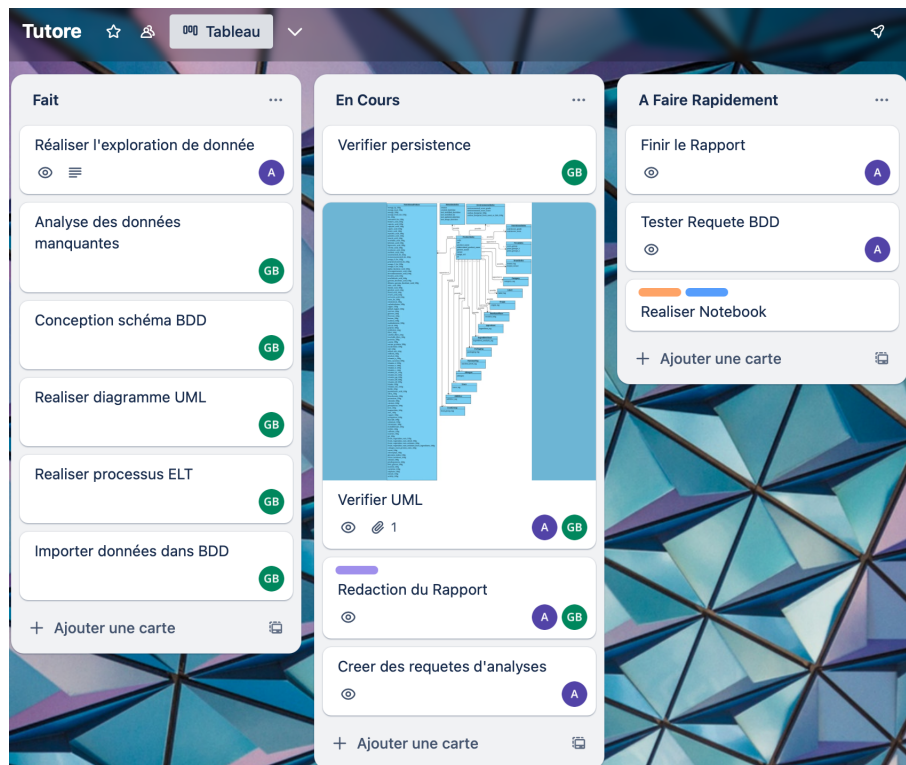


FIGURE 1 – Utilisation de l'outil Trello pour la gestion de tâches

En ce qui concerne la communication, nous avons utilisé **Discord**, une plateforme intuitive qui facilite les échanges en temps réel. Grâce à Discord, nous avons pu discuter, partager des fichiers, et organiser des réunions vocales lorsque nécessaire. L'utilisation de cet outil a simplifié nos interactions, étant donné que nous sommes une équipe de deux personnes. Nous avons également veillé à répondre rapidement aux messages pour maintenir une communication fluide.

Nous n'avons pas utilisé d'outils spécifiques pour le suivi du temps. À la place, nous avons fixé ensemble des délais pour chaque tâche à réaliser. Cependant, certains délais ont dû être dépassés en raison de difficultés techniques ou d'autres priorités académiques liées à d'autres matières.

Pour le développement de ce projet, nous avons mobilisé nos compétences en **Machine Learning** et utilisé le langage **Python** ainsi que ses bibliothèques principales, telles que **Pandas** pour la manipulation des données et **Matplotlib** pour leur visualisation. Ces

outils nous ont permis d'effectuer des analyses approfondies et de présenter des résultats pertinents.

En ce qui concerne la gestion des données, nous avons opté pour **PostgreSQL** comme système de gestion de base de données en raison de sa capacité à gérer de grandes quantités de données et de sa simplicité. Il a parfaitement répondu aux besoins du projet en matière de stockage et de gestion des informations. **Kafka** a été sélectionné en tant que serveur **kafka.iem** sur lequel nous avons pu nous connecter à la base de données.

Pour garantir une collaboration fluide et un suivi efficace du projet, nous avons utilisé **GitHub** pour le versionnage du code et la gestion de la collaboration. Cette plateforme a permis à chaque membre de l'équipe d'accéder aux dernières modifications et d'intégrer ses propres contributions au projet de manière synchronisée. Chaque ajout de travail réalisé était ainsi mis à jour en temps réel, facilitant l'intégration continue et l'avancée de chacun.

Répartition du travail et organisation

Pour la répartition du travail, nous avons adopté une méthodologie collaborative basée sur nos compétences respectives et nos affinités. Chaque membre de l'équipe était responsable d'une partie spécifique du projet. Par exemple, pour la répartition des tâches principales, nous avons :

- **Guillaume** : Conception de l'ELT (Extract - Load -Transform) ; schéma et gestion de la partie Base de Données.
- **Assia** : Explorations de données ; requête d'analyse et rédaction notebook Jupyter

Certaines tâches ont été réalisées individuellement par chaque membre de l'équipe, puis corrigées et modifiées par l'autre. Nous avons également travaillé ensemble sur d'autres tâches, combinant nos compétences respectives afin de réaliser le travail de manière optimale.

Importance de l'environnement de travail

L'environnement de travail a joué un rôle important dans la réussite du projet. Travailler dans un cadre calme et organisé nous a permis de rester concentrés et productifs. Étant dans le même groupe académique, nous avons pu collaborer facilement en présentiel lorsque nécessaire. Ces rencontres physiques ont renforcé notre coordination et facilité les échanges d'idées.

Nous avons également mis en place des réunions régulières pour discuter des choix stratégiques et techniques ayant un impact majeur sur le projet. Lors de ces échanges, chaque membre a pu présenter ses idées, argumenter sur les solutions proposées, et démontrer les avantages de sa vision. En cas de divergence, nous avons adopté une méthode d'argumentation constructive, où chacun exposait les raisons de son choix, suivie d'un processus de décision par élimination. Cette approche nous a permis de concilier les points de vue de manière efficace, de prendre des décisions éclairées et de progresser tout en

apprenant les uns des autres.

En résumé, un environnement collaboratif bien structuré, combiné à des outils adaptés comme Trello et Discord, a été déterminant pour mener à bien ce projet malgré les défis rencontrés.

3 Pré-analyse et filtrage des données

3.1 Données disponibles

Open Food Facts propose un export de ses données sous diverses formes : un "dump" MongoDB (une image brute de toute la base de données ayant pour but d'être réintégrée directement au sein d'une base de données MongoDB), un fichier CSV, un fichier JSONL, ainsi qu'un fichier RDF.

Pour notre analyse, nous avons choisi d'utiliser le fichier CSV, car il est plus facile à manipuler et à corriger hors-ligne, et plus facile à importer et à analyser au sein d'une base de données relationnelle.

Nous ne voulions pas utiliser le fichier JSONL ou l'export MongoDB du fait de la complexité de la structure des données et de la difficulté de trouver un schéma de données adéquat pour les importer dans une base de données relationnelle.

Nous n'avons pas utilisé le fichier RDF, car nous n'avons pas trouvé de moyen simple de l'ouvrir au sein d'outils comme Protégé (<https://protege.stanford.edu/>) dont nous avons l'habitude au sein de notre formation. Ce dernier nous retournait une erreur de syntaxe. Notre tuteur nous a alors conseillé d'examiner le fichier RDF avec un autre outil, Neo4j (<https://neo4j.com/>). Cependant nous n'avons pas pu extraire le graphe de connaissance à partir du fichier RDF fourni par Open Food Facts. Nous avons donc décidé de renoncer à cette approche.

Enfin, le fichier CSV obtenu d'Open Food Facts est un fichier trop volumineux pour être importé dans un tableur. Nous avons donc décidé de l'importer dans une base de données relationnelle pour pouvoir l'analyser plus facilement colonne par colonne.

3.2 Structure des données

Le fichier CSV décrit une table unique dans laquelle chaque ligne correspond à un produit alimentaire. Chaque colonne de la table correspond à un attribut de la fiche de ce produit.

À noter que, selon la langue choisie lors de l'export, certaines colonnes peuvent être dupliquées avec un suffixe de deux lettres indiquant la langue choisie. Par exemple, la

colonne `labels` peut être dupliquée en `labels_fr` si la langue de l'export choisie est le français, ou en `labels_en` si la langue de l'export choisie est l'anglais. Dans la suite de notre analyse, nous désignerons le suffixe de langue par `_lang`.

Voici la liste des colonnes présentes dans le fichier CSV (variante `fr`) : Voir Annexe.

3.3 Méthodologie d'exploration des données

Afin de pouvoir explorer ces données, nous nous sommes d'abord tournés vers le site Open Food Facts pour comprendre les informations disponibles et leur signification. Nous avons également consulté la documentation de l'API Open Food Facts (lien vers la documentation) pour comprendre la signification des colonnes du fichier CSV. Cependant, cette documentation n'est pas exhaustive et ne décrit pas toutes les colonnes du fichier CSV ni leur contenu.

Nous avons ensuite importé le fichier CSV en une table unique `off_origin` dans une base de données relationnelle (PostgreSQL) pour pouvoir manipuler les données plus facilement. La table n'ayant pas de colonne remplissant une contrainte d'unicité, nous avons ajouté une nouvelle colonne `id` pour identifier chaque ligne de la table.

Par la suite, nous avons séparé la table en de multiples tables, isolant chaque colonne avec la colonne `id` pour pouvoir les analyser et les traiter individuellement.

Le but de cette pré-analyse est de déterminer les colonnes pertinentes pour notre analyse, de repérer les colonnes redondantes et de pouvoir les nettoyer et les traiter en conséquence. Nous chercherons ensuite à reconstruire un schéma de données plus simple et mieux orienté pour notre analyse.

3.4 Exploration de données avec Pandas

Pour cette partie, une première exploration des données a été réalisée en appliquant nos connaissances en machine learning. Nous avons utilisé un notebook Jupyter, accompagné des bibliothèques Python essentielles, telles que **Pandas** pour la manipulation et l'analyse des données, **Matplotlib** pour la visualisation graphique des résultats, et **Seaborn** pour la création de visualisations statistiques avancées. L'exploration des données a été rendue plus efficace grâce à ces outils.

Étant donné la taille importante du fichier, environ 11 Go, l'exécution des codes était initialement lente. Afin de surmonter cette contrainte, nous avons utilisé **Dask**, une bibliothèque permettant de traiter les données de manière distribuée. Dask nous a ainsi permis de gérer des volumes de données importants en répartissant les calculs sur plusieurs noyaux de processeurs, ce qui a considérablement accéléré les opérations.

Cette exploration nous a permis de recueillir plusieurs informations de base telles que le nombre de lignes et de colonnes du jeu de données, le nombre de valeurs manquantes pour chaque colonne, ainsi que les valeurs distinctes présentes dans chaque colonne, et leurs occurrences respectives. Ci-dessous les valeurs uniques de la colonne `main_category_fr` et leurs occurrences (image gauche) et les valeurs uniques de la colonne `environmental_score_grade` (image droite) et leurs occurrences :

Valeurs uniques pour la colonne main_category_fr:	
main_category_fr	
Compléments alimentaires	5
Boissons	3
Madeleines au chocolat	3
en:supplement	2
Madeleines longues	2
en:protein	2
Chia	2
Vitamines	2
Protéines en poudre	2
Quarks	1
en:clean-antioxydant-energy-drink	1
it:olio-di-mandorle-dolci	1
Madeleines natures	1
en:tortilla	1
en:cinnamon-roll	1
Risottos	1
it:bieta-da-costa	1
en:butfalo-mac-and-cheese	1
en:sample	1
Légumes	1
Bouillons cubes	1
it:integratore-alimentare-di-vitamina-b6	1
Wraps	1

Valeurs uniques pour la colonne environmental_score_grade:	
environmental_score_grade	
unknown	38
d	5
c	4
b	4
e	3
f	1
a	1
not-applicable	1

(a) Liste et occurrence des valeurs d'une colonne spécifique

(b) Valeur unique d'une colonne spécifique

FIGURE 2 – Résultats exploration de données

Ces premières analyses nous ont non seulement donné une vue d'ensemble du jeu de données, mais ont également mis en lumière des aspects cruciaux comme la présence de données manquantes ou des colonnes redondantes. Cela nous a été d'une grande aide pour la phase suivante du projet, notamment dans le choix des données à conserver et de celles à ignorer ou traiter.

Grâce à cette exploration, nous avons constaté que près de la moitié des colonnes de notre fichier de données comportaient un grand nombre de valeurs manquantes. Dans le document suivant *Voir Annexe (Figure 7)*, nous avons inclus un tableau répertoriant toutes les colonnes, triées par ordre décroissant du nombre de valeurs manquantes. On observe que la première ligne correspond à la colonne `cities`, qui présente un nombre de valeurs manquantes égal au nombre total de lignes du fichier, ce qui signifie que cette colonne est entièrement vide.

De plus, nous avons inclus deux graphiques illustrant la répartition des valeurs manquantes dans le fichier. Par exemple, dans le premier graphique, on remarque que plus de la moitié des colonnes contiennent plus de 80 % de valeurs manquantes. Cela suggère que notre fichier de données n'est pas dans un état optimal pour une analyse approfondie.

Informations obtenues lors de l'exploration des données :

- Le fichier contient plus de 3 millions de lignes et 206 colonnes.
- Les données sont présentes dans plusieurs langues.
- Certaines colonnes contiennent des listes de valeurs.
- Les types des valeurs numériques ont été déterminés, spécifiant si elles sont décimales ou entières.
- Nous avons obtenu une vue d'ensemble des données.
- Des valeurs manquantes ou des erreurs ont été détectées dans les données.

Toutefois, il est important de noter que nous avons eu une liberté totale dans le choix du fichier de données pour ce projet. Nous aurions pu abandonner ce fichier et en rechercher un autre, avec moins d'erreurs et de valeurs manquantes, afin de garantir des analyses plus fiables et précises. Cependant, nous avons opté pour une approche différente. En effet, dans le cadre professionnel, il est probable que nous soyons amenés à travailler avec des fichiers de données imparfaits, voire fortement incomplets. Il est donc essentiel de savoir comment aborder ce type de situation et en tirer le maximum d'informations.

Ainsi, nous avons décidé de poursuivre avec ce fichier, non seulement pour comprendre les défis associés à ce type de données, mais aussi pour apprendre à exploiter au mieux ce genre de situation. Cela nous permettra, à terme, de mieux appréhender de futurs projets, où nous devrons peut-être traiter des données incomplètes ou non optimisées, et savoir quelles stratégies adopter pour en tirer des analyses pertinentes et exploitables.

3.5 Choix des colonnes redondantes

Nous pouvons remarquer que certaines colonnes sont redondantes, comme `categories`, `categories_tags` et `categories_fr`. Globalement, les colonnes sans suffixe ou avec le suffixe `_text` contiennent des informations brutes, peu formatées, tandis que les colonnes avec le suffixe `_tags` contiendront des informations plus formatées, privilégiant des tags normalisés et privilégiant l'anglais. Ce sont ces colonnes qui seront privilégiées pour les analyses. Les colonnes avec le suffixe de la langue de l'export contiendront des informations formatées pour les langues étrangères (exemple : `de:<texte>` pour des informations en allemand) mais également non formatées pour la langue choisie lors de l'export (exemple : `fr:<texte>` deviendra `<texte>` dans le cas d'un export en français).

Nous pouvons remarquer que, pour certains groupes de colonnes, l'utilisation de la variante avec la langue choisie lors de l'export peut sembler plus pertinente. Par exemple, la colonne `additives_fr` donne également le nom des additifs en français (exemple : `E132 - Indigotine carmin d'indigo`), alors que la colonne `additives_tags` donne les tags des additifs en anglais sans complément d'information (exemple : `en:e132`).

Dans d'autres cas, la colonne avec le suffixe de langue peut perdre sa pertinence. Par exemple, la colonne `food_groups_fr` ne contient que 4 valeurs traduites en français sur les 52 valeurs présentes dans les données.

Cependant, ces observations ne s'appliquant pas à toutes les langues, nous avons quand même décidé de privilégier les colonnes suffixées par `_tags` pour notre analyse afin de rester cohérents sur le schéma global ainsi que pour des raisons de simplification des requêtes dans notre ETL.

De même que pour les variations de langues, certaines colonnes indiquant des informations d'horodatage peuvent être dupliquées avec un suffixe `_t` ou `_datetime`, ces derniers représentant respectivement le timestamp Unix (nombre de secondes écoulées depuis le 1er janvier 1970 à 00 :00 :00 UTC) et la date formatée en chaîne de caractères selon le format ISO 8601 (`yyyy-mm-ddThh:mn:ssZ`). Nous avons décidé de privilégier les colonnes suffixées par `_datetime` pour notre analyse, car elles sont plus faciles à lire pour un humain.

3.6 Choix des colonnes à ignorer

Parmi les colonnes présentes dans le fichier CSV, certaines ne nous semblent pas pertinentes pour notre analyse. Il s'agit des colonnes suivantes :

- `allergens_lang`, `cities` : Ces deux colonnes sont vides.
- `allergens_n` : Cette colonne contient le nombre d'allergènes du produit et pourra être restituée par la suite.
- `states`, `states_tags`, `states_lang` : Ces colonnes contiennent des informations sur l'état du produit dans la base de données Open Food Facts.
- `data_quality_errors_tags` : Cette colonne contient des informations sur les erreurs de remplissage de la fiche produit.
- `unique_scans_n`, `popularity_tags` : Ces colonnes contiennent des informations sur la popularité du produit.
- `completeness` : Cette colonne contient des informations sur le taux de remplissage de la fiche produit, mais puisque nous allons supprimer des colonnes, nous n'en aurons plus besoin.
- `quantity`, `serving_size`, `serving_quantity`, `product_quantity` : Ces colonnes contiennent des informations sur la quantité de produits et celle d'une portion, mais nous avons décidé de ne pas les utiliser pour notre analyse.

- **no_nutrition_data** : Cette colonne contient des informations sur la présence ou non de données nutritionnelles pour le produit, elle était utile jusqu'à ce que les colonnes de données nutritionnelles soient retirées de l'export.
- **first_packaging_code_geo** : Cette colonne contient des informations sur le code géographique du premier emballage du produit.
- **image_small_url, image_ingredients_url, image_ingredients_small_url, image_nutrition_url, image_nutrition_small_url** : Ces colonnes contiennent des liens vers des images du produit et de ses étiquettes, mais nous avons décidé de ne pas les utiliser pour notre analyse. Elles nécessiteraient un traitement de reconnaissance d'image que nous ne sommes pas en mesure de réaliser. Nous garderons cependant la colonne **image_url** pour pouvoir éventuellement présenter un visuel des produits à l'utilisateur.
- **purchase_places, stores, cities_tags** : Ces colonnes contiennent des informations sur les lieux d'achat du produit, nous nous limiterons au pays d'achat.
- **manufacturing_places, manufacturing_places_tags** : Ces colonnes contiennent des informations sur les lieux de fabrication du produit, nous nous limiterons au pays d'origine.
- **nutrition_score_fr_100g, nutrition_score_uk_100g** : Ces colonnes sont redondantes avec la colonne **nutriscore_score**.
- **main_category_fr** : Cette colonne est redondante avec **categories_tags**.
- **ingredients_text** : Cette colonne est redondante avec **ingredients_tags**, avec un texte brut parfois peu exploitable.

3.7 Reconstruction du schéma de données

Les colonnes restantes nous semblent pertinentes pour notre analyse, nous avons donc décidé de les réorganiser en plusieurs tables pour simplifier notre schéma de données et les rendre plus faciles à manipuler. Voici les tables que nous allons créer :

Les colonnes restantes de la table **off_nutritional_values** sont disponibles en annexe, accessible via le lien suivant *Voir Annexe* .

Nom de la table	Colonnes
off_product_infos	code, url, product_name, abbreviated_product_name, generic_name, owner, image_url
off_brands_infos	brands_tags, brand_owner
off_metadata_infos	creator, created_datetime, last_modified_datetime, last_modified_by, last_updated_datetime, last_image_datetime
off_nova_infos	nova_group, pnns_groups_1, pnns_groups_2

Nom de la table	Colonnes
off_environmental_infos	environmental_score_grade, environmental_score_score, carbon_footprint_100g, carbon_footprint_from_meat_or_fish_100g
off_nutritional_infos	nutriscore_grade, nutriscore_score
off_nutritional_values	energy_kj_100g, energy_kcal_100g, energy_100g, energy_from_fat_100g, ...
off_categories	categories_tags
off_origins	origins_tags
off_labels	labels_tags
off_purchase_places	countries_fr
off_ingredients	ingredients_tags
off_ingredients_anal	ingredients_analysis_tags
off_packaging	packaging_tags
off_nutrient_tags	nutrient_levels_tags
off_allergens	allergens
off_traces	traces_tags
off_additives	additives_tags
off_food_groups	food_groups_tags

Ces tables reprendront toutes la colonne **id** afin de les lier entre elles, elle sera une clé primaire pour `off_product_infos` et une clé étrangère sur cette dernière pour les autres. Les tables allant de `off_categories` à `food_groups_tags` reprennent des colonnes en séparant les éléments de chaque liste sur des lignes distinctes.

3.8 Bilan de la pré-analyse

Au terme de cette pré-analyse, nous avons pu identifier les colonnes pertinentes pour notre analyse et celles à ignorer, comme les colonnes redondantes, les colonnes vides et celles n'apportant pas d'informations utiles pour le but recherché. Nous avons également pu reconstruire un schéma de données plus simple et mieux orienté pour notre analyse. Nous allons maintenant pouvoir passer à la création de notre schéma conceptuel et à l'analyse des données.

4 Modelisation

4.1 Modèle de données (BDR)

Le diagramme de base de données présente la structure des différentes tables utilisées dans le modèle, avec les clés primaires et étrangères clairement indiquées. Chaque table

est détaillée avec le type de données associé à chaque colonne, assurant ainsi une bonne gestion des relations entre les entités. Ce diagramme permet de visualiser les connexions logiques entre les données, et d'identifier comment les tables interagissent les unes avec les autres.

Le diagramme suivant (figure 3) illustre le modèle de données relationnel, où chaque table est définie par ses colonnes, avec des spécifications concernant les types de données, les clés primaires et les clés étrangères.

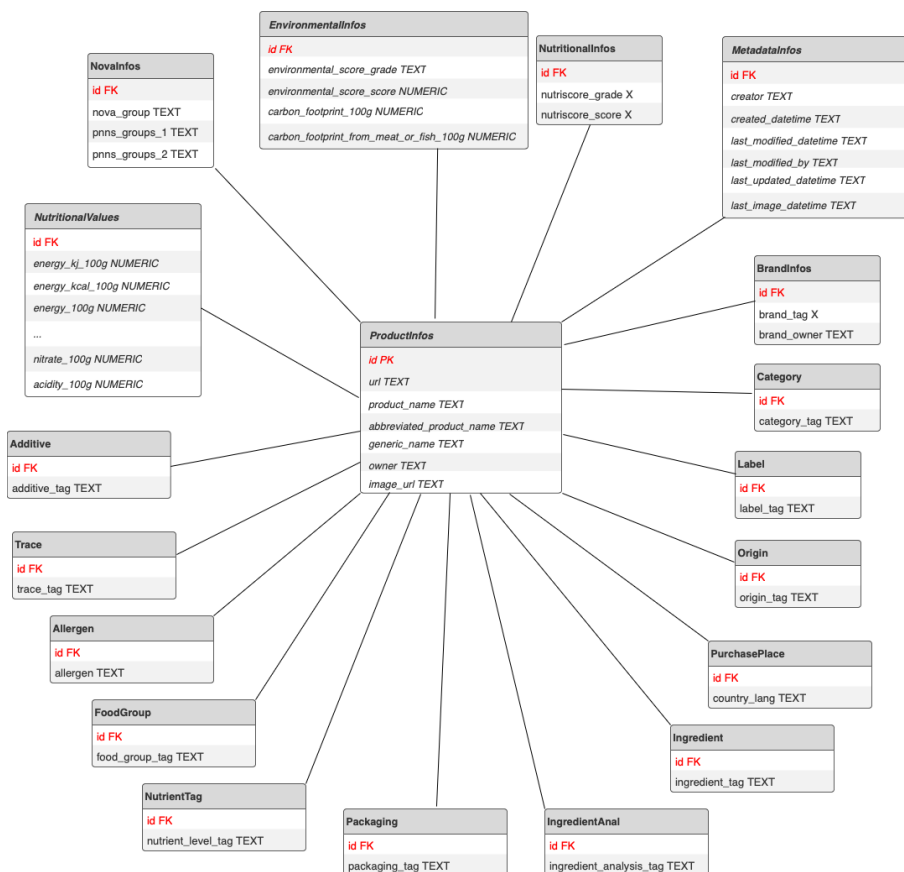


FIGURE 3 – Tables base de données

Nous avons donc la table ProductInfos qui a un identifiant unique en clé primaire et toutes les autres tables ont des clés étrangères qui référencent la table ProductInfos. Pour les types de données nous avons utilisé le type **NUMERIC** pour les colonnes numériques et **TEXT** pour toutes les autres colonnes. Cela nous a permis d'importer rapidement les données et de traiter les valeurs non pertinentes, telles que les champs vides ou ceux remplis uniquement de ponctuation. Cela nous a ainsi permis de les nullifier et de les exclure des analyses futures.

4.2 Diagramme UML

Le diagramme UML : Voir Annexe (Figure 6)

Le diagramme UML présenté représente la modélisation des tables de la base de données relationnelle destinée à la gestion des informations nutritionnelles, environnementales et générales des produits alimentaires.

La table centrale est **ProductInfos**, qui regroupe les informations principales sur chaque produit telles que son nom, son nom générique, son image ou encore ses catégories. Cette table est liée à plusieurs autres entités représentant des caractéristiques spécifiques du produit.

Par exemple, chaque produit est associé à une marque (**BrandInfos**), un groupe de transformation (**NovaInfos**), ainsi qu'à des données nutritionnelles (**NutritionalInfos**) telles que les apports en énergie, protéines, lipides, vitamines ou minéraux. De plus, les informations environnementales (**EnvironmentalInfos**) comme l'empreinte carbone peuvent également être rattachées à un produit.

D'autres tables viennent enrichir la description des produits, comme **Label**, **Origin**, **PurchasePlace**, **Ingredient**, **Packaging**, **NutrientLevels**, **Allergen**, **Additive** ou encore **FoodGroup**, chacune pouvant contenir plusieurs éléments associés à un produit.

Concernant les cardinalités, nous distinguons différents types de relations :

- *Relations de type un-à-un :*
 - **ProductInfos** et **MetadataInfos** : Un produit possède un ensemble d'informations metadata.
- *Relations de type 0-à-1 (zéro-à-un) :*
 - **ProductInfos** et **EnvironmentalInfos** : Un produit peut posséder un (et un seul) ensemble d'informations environnementales, ou aucune.
 - **ProductInfos** et **NutritionalInfos** : Un produit peut posséder un (et un seul) ensemble d'informations nutritionnelles, ou aucune.
 - **ProductInfos** et **NovaInfos** : Un produit appartient à un (et un seul) groupe, ou aucun.
 - **ProductInfos** et **NutritionalValues** : Un produit peut posséder un (et un seul) ensemble de valeurs nutritionnelles, ou aucun.
- *Relations de type 1..* à 0..* (au moins un à plusieurs) :*
 - **ProductInfos** et **BrandInfos**, **Label**, **Origin**, **Ingredient**, **IngredientAnal**, **Packaging**, **NutrientTag**, **Allergen**, **Trace**, **Additive** : Un produit possède au moins un label, une origine, un ingrédient, etc.
 - **ProductInfos** et **Category**, **FoodGroup** : Un produit appartient à au moins une catégorie ou un groupe alimentaire.
 - **ProductInfos** et **PurchasePlace** : Un produit est vendu dans au moins un lieu de vente.

4.3 Description d'Analyse

Afin de surmonter les défis posés par des données mal formatées lors de l'importation, notamment des problèmes de compatibilité de types où certaines colonnes numériques contenaient des valeurs incohérentes ou non conformes, une nouvelle stratégie a été mise en place. Nous avons choisi d'attribuer le type **NUMERIC** uniquement aux colonnes contenant des données numériques, tandis que toutes les autres colonnes ont été définies avec le type **TEXT**. Cette approche nous a permis de conserver une structure de données cohérente tout en assurant une certaine robustesse lors de l'importation. En effet, elle permettait de préserver la nature quantitative des données essentielles à l'analyse, tout en évitant les erreurs d'importation bloquantes pour les autres colonnes. Ce compromis a facilité le nettoyage ultérieur des données et leur exploitation dans les analyses, sans compromettre l'intégrité des informations importantes.

De plus, les tables ayant été créées à l'aide de l'instruction **CREATE TABLE AS SELECT**, les contraintes de clés primaires (PK) et de clés étrangères (FK) n'ont pas été automatiquement établies. Par conséquent, il est nécessaire d'ajouter manuellement ces contraintes après la création des tables, en utilisant l'instruction **ALTER TABLE**. Cette étape est cruciale pour rétablir l'intégrité des données et assurer la cohérence des relations entre les tables. Les données sont donc correctes mais le SGBD ne les surveille pas activement car les contraintes qui forceraient le respect des règles ne sont pas définies.

Nous disposons d'un schéma en étoile, structuré autour de la table centrale **ProductInfos**, qui joue le rôle de table de faits. Toutes les autres tables représentent des dimensions qui viennent enrichir les informations liées aux produits. En effet, l'ensemble des données nutritionnelles, environnementales, d'étiquetage, d'origine, d'ingrédients, de lieu d'achat, d'emballage ou encore de catégories, gravitent autour de cette table centrale. Ce modèle permet une organisation claire et optimisée des données, facilitant ainsi les opérations d'analyse et de requête. Chaque table dimensionnelle apporte un axe de lecture complémentaire sur le produit, ce qui reflète parfaitement la logique d'un entrepôt de données orienté produit.

5 Implémentation technique

Le schéma ci-dessous (figure 4) décrit le processus à partir du téléchargement des données à l'importation des données dans la base de données.

Dans le cadre de notre projet, nous avons opté pour une approche **ELT (Extract - Load - Transform)** plutôt que **ETL (Extract - Transform - Load)**.

L'ETL, bien qu'adapté aux entrepôts de données, s'avère souvent trop lent pour les volumes importants en raison des transformations préalables coûteuses.

L'**ELT (Extract - Load - Transform)** (figure 5) consiste également à extraire les données depuis une ou plusieurs sources distantes, mais celles-ci sont d'abord chargées telles quelles, sans transformation préalable, dans l'entrepôt de données cible.

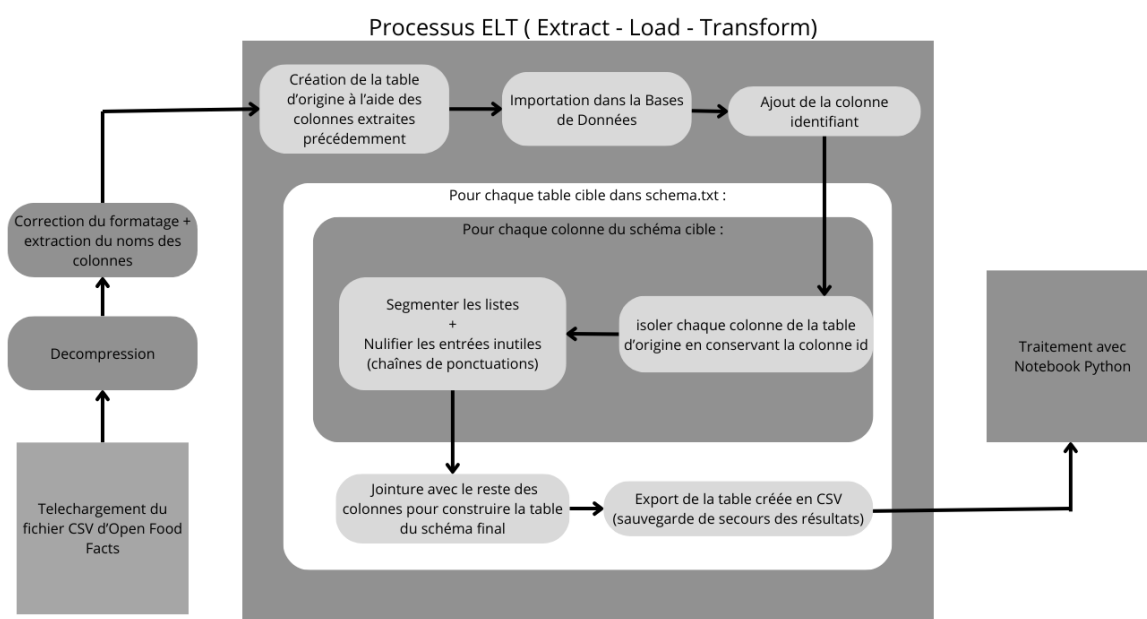


FIGURE 4 – Schéma ELT

Contrairement au processus ETL, les transformations ne sont pas réalisées en amont, mais directement au sein de la base de données cible. Cette approche se concentre sur l’exploitation des **données brutes**, sans nécessiter une préparation préalable des sources, ce qui permet de simplifier l’intégration initiale tout en offrant plus de flexibilité.

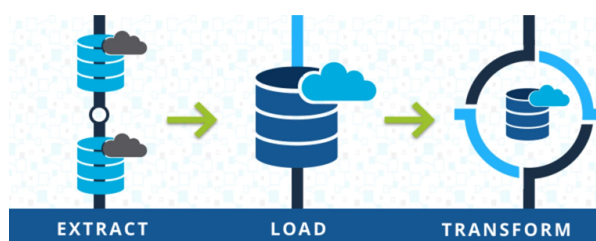


FIGURE 5 – Processus ELT

Les transformations, y compris le nettoyage des données, sont ensuite effectuées directement dans la base, offrant ainsi une flexibilité accrue et la possibilité de procéder par étapes. Cette méthode s’est avérée particulièrement efficace pour notre projet, nous permettant d’importer nos données dans PostgreSQL et de les traiter de manière progressive.

Après le téléchargement du fichier CSV d’**Open Food Facts**, une étape de correction du formatage a été réalisée, accompagnée de l’extraction des noms des colonnes. Certaines colonnes étant mal formatées, un nettoyage a été nécessaire afin de récupérer correctement

leurs intitulés.

Une table initiale a ensuite été créée dans PostgreSQL, dans laquelle toutes les colonnes ont été définies avec le type **TEXT**. Ce choix a permis de garantir l'importation complète des données, quelles que soient les variations de type. Le fichier a alors été importé dans cette table temporaire. Ensuite nous avons pu modifier les colonnes numériques en modifiant leur type de **TEXT** à **NUMERIC**.

Afin de faciliter les opérations de transformation ultérieures, une colonne identifiant unique a été ajoutée à chaque ligne de la table. Un traitement a ensuite été appliqué colonne par colonne, permettant d'isoler, nettoyer et transformer les données selon le schéma cible.

Enfin, les colonnes traitées ont été regroupées pour reconstituer les tables finales conformément au modèle de données. Les résultats obtenus ont été exportés et sauvegardés, assurant ainsi la conservation des données transformées.

Notre approche **ELT** va consister à enregistrer les données source du fichier CSV dans un espace temporaire, du côté de la base cible (la table en gris). Ensuite les données seront intégrées dans la table cible (les tables vertes).

Pour notre projet, nous disposons d'un fichier de plus de 3 millions de lignes à nettoyer avant de pouvoir les analyser. Grâce à l'approche **ELT (Extract, Load, Transform)**, on charge d'abord les données brutes dans la base cible, ici **PostgreSQL**, puis on effectue les transformations directement dans la base de données. Ainsi, les données restent dans leur environnement d'origine, ce qui permet d'optimiser les performances et de réduire la charge sur les ressources réseau.

6 Realisation du Notebook

6.1 Fonctionnalités du Notebook

Un notebook Jupyter a été développé pour permettre aux spécialistes de l'alimentaire d'effectuer diverses analyses sur les données. Il offre plusieurs fonctionnalités utiles pour mener à bien ces analyses.

Le notebook commence par importer les données présentes dans la base de données PostgreSQL, puis les requêtes d'analyse sont exécutées directement sur ces données. Après l'exécution des requêtes, des visualisations des résultats sont générées. Une explication détaillée des résultats est également fournie afin d'aider les utilisateurs à comprendre et à interpréter correctement les graphiques.

Il est conçu pour être facilement paramétrable, permettant ainsi aux utilisateurs de modifier les paramètres des requêtes selon leurs besoins.

6.2 Bibliothèques utilisées

Dans le notebook, nous avons utilisé plusieurs bibliothèques pour interagir avec la base de données et réaliser des visualisations. Nous avons utilisé **pandas** pour charger les données et les manipuler sous forme de DataFrames, ce qui nous a permis de travailler facilement

avec les résultats des requêtes. Pour la connexion à la base de données PostgreSQL, nous avons utilisé `psycopg2`, une bibliothèque fiable et performante pour interagir avec cette base de données. Ensuite, pour les visualisations, nous avons fait appel à `matplotlib` pour créer des graphiques statiques simples, et à `seaborn` pour réaliser des visualisations statistiques plus avancées et esthétiques. Enfin, nous avons utilisé `train_test_split` de `sklearn.model_selection` pour diviser les données en ensembles d'entraînement et de test, ce qui nous a permis de préparer les données pour les modèles de machine learning. Ces bibliothèques nous ont permis de traiter, de visualiser et de préparer efficacement les données tout au long du processus d'analyse.

6.3 Exploration initiale des données

Dans une première partie, l'objectif était de fournir un aperçu général de la base de données. Ainsi, nous avons affiché les premières lignes de certaines tables et listé les différentes colonnes disponibles dans chaque table. Cette étape permet à l'utilisateur de se familiariser avec la structure des données s'il souhaite réaliser des analyses complémentaires.

6.4 Analyses nutritionnelles

Nous avons ensuite entamé des requêtes d'analyse plus spécifiques :

- **Moyenne et médiane des lipides, glucides et protéines par catégorie de produit :**
Cette requête permet d'identifier les catégories de produits les plus riches ou pauvres en ces éléments nutritionnels, facilitant une première évaluation de la qualité nutritionnelle globale des différentes catégories.
- **Top 10 des additifs les plus utilisés dans les produits :**
Une visualisation accompagnait cette requête afin de mettre en évidence la fréquence d'utilisation des additifs. Cette analyse permet de repérer les additifs les plus courants et soulève des questions sur leur présence dans l'alimentation.
- **Moyenne des sucres par lieu de vente (Top 10 uniquement) :**
Cette requête permet d'observer comment les niveaux de sucre varient en fonction des points de distribution, ce qui peut refléter une variation des gammes de produits commercialisés selon les enseignes.
- **Moyenne des sucres par origine des produits :**
Cette analyse permet de comparer les teneurs en sucre en fonction de l'origine géographique, mettant en évidence d'éventuelles tendances ou différences entre pays.
- **Distribution de la quantité de sucres ajoutés par catégorie de produit :**
Cette requête fournit des indications sur les catégories où les sucres ajoutés sont les plus fréquents, ce qui peut être utile pour des recommandations nutritionnelles ciblées.
- **Distribution du nombre d'occurrences des ingrédients par catégorie (Top 10) :**

Cette analyse, accompagnée d'une visualisation, permet d'identifier les ingrédients les plus couramment utilisés et leur répartition dans les différentes catégories de produits.

- **Marques avec une teneur moyenne en sucre inférieure ou égale à 10g :**
Cette requête vise à mettre en avant les marques proposant des produits globalement moins sucrés, ce qui peut aider les consommateurs à faire des choix plus sains.
- **Vérification du Nutri-Score des marques identifiées précédemment :**
Pour compléter l'analyse, nous avons extrait les Nutri-Scores de trois marques issues de la requête précédente, afin de valider si ces marques proposent effectivement des produits à bon profil nutritionnel.
- **Top catégories de produits avec une teneur en fer élevée (Top 10) :**
Cette requête identifie les catégories riches en fer, critère nutritionnel important. Une visualisation a été ajoutée pour illustrer la répartition des produits en fonction du seuil de fer défini.
- **Matrice de corrélation des valeurs nutritionnelles :**
L'objectif est de comprendre les relations entre certains nutriments (protéines, sucres, glucides, énergie, etc.). Des corrélations trop fortes (supérieures ou égales à 0,99) ont été observées, ce qui a soulevé des questions d'incohérence dans les données. Pour approfondir, nous avons ensuite vérifié le nombre de valeurs manquantes, car une forte proportion de données manquantes peut fausser les résultats statistiques.

6.5 Analyses environnementales

Nous avons également mené une série d'analyses relatives à l'impact environnemental des produits :

- **Distribution des scores environnementaux par origine des produits :**
Cette requête permet d'identifier les origines des produits les plus représentées dans la base et d'étudier leur répartition selon les scores environnementaux. Nous avons focalisé l'analyse sur les trois pays les plus présents : la France, l'Italie et l'Union européenne.
- **Distribution des grades environnementaux par origine (avec visualisation) :**
Cette analyse visuelle permet de comparer la répartition des grades (de A+ à F, ainsi que "unknown") entre les pays. Elle met en évidence des différences significatives dans le profil environnemental des produits selon leur origine.
- **Top 10 des produits avec les scores environnementaux les plus élevés :**
Cette requête identifie les produits les plus vertueux d'un point de vue environnemental, ce qui peut orienter les recommandations d'achats responsables.
- **Comparaison de l'empreinte carbone entre produits avec et sans viande/-poisson :**
Cette analyse met en évidence l'impact environnemental plus élevé des produits contenant des ingrédients d'origine animale, soulignant leur contribution plus importante aux émissions de gaz à effet de serre.

6.6 Fonctionnalités avancées

6.6.1 Prédiction du Nutri-Score à l'aide du Machine Learning

Enfin, nous avons mis en œuvre un modèle de machine learning afin de prédire les valeurs de la colonne `nutriscore_grade`, qui attribue un score nutritionnel aux produits (valeurs possibles : a, b, c, d, e).

Nous avons fait ce choix car cette colonne représente une information clé dans l'analyse des produits alimentaires. Or, un grand nombre de valeurs y sont manquantes, ce qui limite considérablement les analyses statistiques et les visualisations basées sur cette variable. L'objectif était donc de tester s'il était possible de prédire les valeurs manquantes du Nutri-Score à partir d'autres variables nutritionnelles disponibles.

Pour cela, plusieurs caractéristiques nutritionnelles ont été sélectionnées comme variables explicatives : `proteins_100g`, `sugars_100g`, `carbohydrates_100g`, et `energy_kcal_100g`. Un nettoyage préalable des données a été effectué, incluant notamment la suppression des valeurs non exploitables telles que "unknown" ou "not-applicable".

Les données ont ensuite été divisées en deux ensembles : un ensemble d'entraînement et un ensemble de validation, afin de permettre une évaluation objective des performances du modèle.

Nous avons d'abord opté pour un algorithme de classification : le **Random Forest**, connu pour sa performance sur des problèmes de classification multiclasse. Toutefois, lors de l'entraînement du modèle, le programme s'est interrompu après quelques minutes d'exécution. Cela pourrait s'expliquer par le volume conséquent des données manipulées (plus de 300 000 lignes et de nombreuses colonnes), ce qui a pu entraîner une surcharge de mémoire et un ralentissement important du traitement.

Face à ces contraintes, nous avons ensuite choisi d'utiliser un second modèle plus léger : la **régression logistique**, adaptée à la classification de variables catégorielles. Ce modèle a pu être entraîné correctement et a fourni des résultats.

Une visualisation des prédictions a été ajoutée afin d'illustrer la répartition des Nutri-Scores prédits. Cette démarche permet non seulement de combler les valeurs manquantes, mais aussi d'ouvrir la voie à des analyses plus complètes et fiables sur la base de données enrichie.

Conclusion générale et bilan

Ce projet a constitué une expérience particulièrement enrichissante sur les plans technique, méthodologique et personnel. Il nous a permis de mettre en œuvre une démarche complète de traitement et d'analyse de données, depuis l'acquisition d'un jeu de données ouvert jusqu'à la restitution d'une analyse claire, pertinente et accessible à des utilisateurs finaux.

Nous avons ainsi pu extraire des données issues d'une base ouverte, les structurer dans une base relationnelle, puis en proposer une analyse à la fois rigoureuse et intelligible. Ce travail revêt une importance particulière à une époque où l'accessibilité et la valorisation

des données ouvertes constituent des enjeux majeurs, en particulier lorsqu'il s'agit de thématiques sensibles comme la santé publique.

Ce projet nous a également permis de prendre du recul sur les avantages mais aussi les limites des données ouvertes. Si ces données sont précieuses par leur accessibilité, elles présentent souvent des lacunes en termes de qualité, de cohérence ou encore de formatage. Nous avons rencontré de nombreuses erreurs de types ou de formats, parfois dues à des valeurs aberrantes ou manquantes, ce qui a rendu certaines étapes de traitement plus complexes. Par ailleurs, certains fichiers étaient d'une taille trop volumineuse pour être facilement manipulables sur une machine grand public, ce qui nous a poussés à chercher des solutions techniques adaptées, notamment des bibliothèques optimisées pour l'exploration de données massives, afin d'éviter les crashes de notre environnement de développement.

L'exploration et la pré-analyse des données ont joué un rôle clé pour identifier les informations pertinentes à conserver. Après avoir procédé au nettoyage. Cependant, les étapes de normalisation des données, notamment pour les valeurs aberrantes, n'ont pas été réalisées. Cette étape nous a réellement montré à quel point la qualité des données conditionne la qualité des analyses.

Sur le plan méthodologique, nous avons appris à concevoir un projet structuré, à établir un cahier des charges simplifié, à planifier les étapes et à anticiper les difficultés potentielles. Une autre dimension importante de ce projet a été l'aspect métier. Étant donné que ce travail s'adresse à des spécialistes du secteur alimentaire, il nous a fallu nous interroger sur leurs besoins concrets : quelles données exploitent-ils au quotidien ? Quels types d'informations leur seraient réellement utiles ? Cette réflexion nous a permis d'orienter nos choix techniques en fonction d'une cible métier claire et pertinente, et d'apprendre à adapter notre travail à des utilisateurs finaux.

En terme de progression, nous avons pu progresser significativement dans plusieurs domaines. Nous avons déjà eu l'occasion de faire de l'exploration de données sur des fichiers de taille modeste, mais c'était la première fois que nous devions traiter un jeu de données volumineux. Nous avons dû apprendre à optimiser notre environnement de travail, rechercher des outils adaptés, et gérer efficacement les ressources mémoire.

Concernant la base de données, les notions étudiées dans le module *Informatique Décisionnelle* notamment les architectures de type Data Warehouse ont été mises en pratique ici. Nous avons ainsi pu concrétiser ces concepts dans un cas réel, ce qui a renforcé notre compréhension de leur utilité.

L'analyse dans le *notebook Jupyter* a aussi été un levier d'apprentissage. Si nous avons déjà utilisé cet outil dans le cadre du module *Machine Learning*, ce projet nous a permis d'approfondir son usage, notamment en connectant directement le notebook à une base PostgreSQL plutôt que de passer par des fichiers CSV. Le traitement de grandes quantités de données et la combinaison de requêtes SQL avec des techniques d'analyse avancée ont représenté un vrai défi, surtout en termes de performances et d'optimisation.

Nous avons également appris à surmonter les difficultés liées à l'importation des

données : erreurs de typage, valeurs manquantes ou erronées. Cela nous a appris que le pré-nettoyage des données est une étape clé, et nous avons exploré diverses méthodes pour y remédier (suppression).

Ce projet nous a aussi permis de renforcer nos compétences transversales : travail collaboratif, communication, organisation, gestion du temps, rédaction technique. Nous avons travaillé en parallèle d'autres projets, ce qui nous a poussés à être rigoureux dans la répartition des tâches, à anticiper les charges de travail, et à respecter des délais réalistes. La rédaction du rapport au fil du temps s'est révélée être une stratégie très efficace pour documenter chaque étape sans perdre d'informations clés.

Enfin, ce projet nous a permis de prendre conscience de l'importance d'une bonne gestion de projet : se fixer des objectifs clairs, estimer les durées des tâches avec réalisme, et maintenir une communication constante au sein de l'équipe. La collaboration avec mon binôme s'est très bien déroulée : nous avons su nous écouter, nous conseiller mutuellement, et avancer dans la même direction, ce qui a grandement contribué à la réussite de ce travail.

En résumé, ce projet nous a permis de consolider nos acquis, de développer de nouvelles compétences pratiques, et surtout de mieux appréhender les enjeux et contraintes réelles d'un projet data destiné à des utilisateurs finaux. Nous en tirons une grande satisfaction et une réelle motivation pour la suite de notre parcours professionnel.

7 Annexes

Nom de la table	Colonnes
-----------------	----------

Nom de la table	Colonnes
off_nutritional_values	energy_kj_100g, energy_kcal_100g, energy_100g, energy_from_fat_100g, fat_100g, saturated_fat_100g, butyric_acid_100g, caproic_acid_100g, caprylic_acid_100g, capric_acid_100g, lauric_acid_100g, myristic_acid_100g, palmitic_acid_100g, stearic_acid_100g, arachidic_acid_100g, behenic_acid_100g, lignoceric_acid_100g, cerotic_acid_100g, montanic_acid_100g, melissic_acid_100g, unsaturated_fat_100g, monounsaturated_fat_100g, omega_9_fat_100g, polyunsaturated_fat_100g, omega_3_fat_100g, omega_6_fat_100g, alpha_linolenic_acid_100g, eicosapentaenoic_acid_100g, docosahexaenoic_acid_100g, linoleic_acid_100g, arachidonic_acid_100g, gamma_linolenic_acid_100g, dihomo_gamma_linolenic_acid_100g, oleic_acid_100g, elaidic_acid_100g, gondoic_acid_100g, mead_acid_100g, erucic_acid_100g, nervonic_acid_100g, trans_fat_100g, cholesterol_100g, carbohydrates_100g, sugars_100g, added_sugars_100g, sucrose_100g, glucose_100g, fructose_100g, lactose_100g, maltose_100g, maltodextrins_100g, starch_100g, polyols_100g, erythritol_100g, fiber_100g, soluble_fiber_100g, insoluble_fiber_100g, proteins_100g, casein_100g, serum_proteins_100g, nucleotides_100g, salt_100g, added_salt_100g, sodium_100g, alcohol_100g, vitamin_a_100g, beta_carotene_100g, vitamin_d_100g, vitamin_e_100g, vitamin_k_100g, vitamin_c_100g, vitamin_b1_100g, vitamin_b2_100g, vitamin_pp_100g, vitamin_b6_100g, vitamin_b9_100g, folates_100g, vitamin_b12_100g, biotin_100g, pantothenic_acid_100g, silica_100g, bicarbonate_100g, potassium_100g, chloride_100g, calcium_100g, phosphorus_100g, iron_100g, magnesium_100g, zinc_100g

Nom de la table	Colonnes
off_nutritional_values	copper_100g, manganese_100g, fluoride_100g, selenium_100g, chromium_100g, molybdenum_100g, iodine_100g, caffeine_100g, taurine_100g, ph_100g, fruits_vegetables_nuts_100g, fruits_vegetables_nuts_dried_100g, fruits_vegetables_nuts_estimate_100g, fruits_vegetables_nuts_estimate_from_ingredients_100g, collagen_meat_protein_ratio_100g, cocoa_100g, chlorophyll_100g, glycemic_index_100g, water_hardness_100g, choline_100g, phylloquinone_100g, beta_glucan_100g, inositol_100g, carnitine_100g, sulphate_100g, nitrate_100g, acidity_100g

Revenir en haut

Ci-dessous le tableau des occurrences des valeurs manquantes triées par ordre décroissant de chaque colonne (non complet).

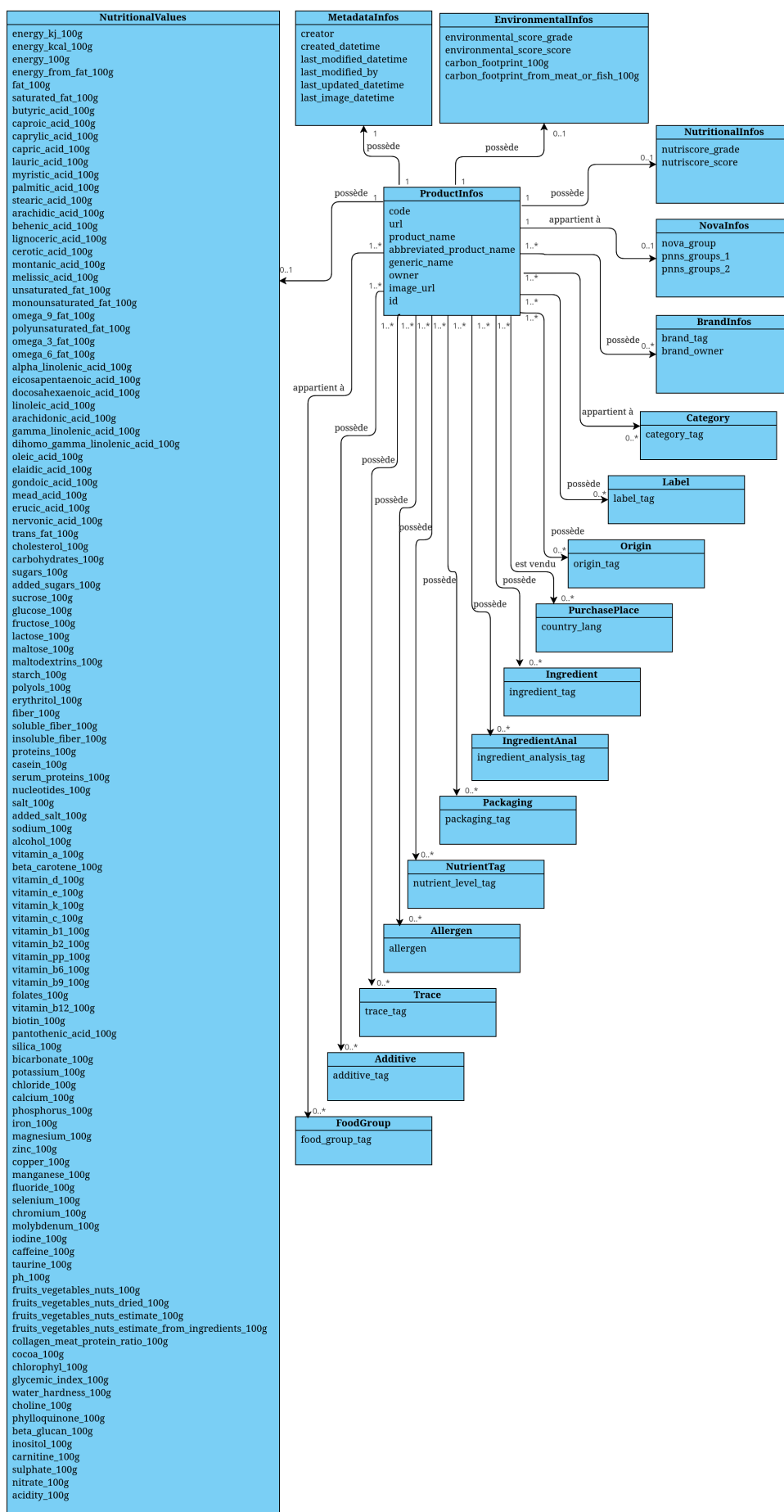


FIGURE 6 – Diagramme UML
Revenir en haut

code	url	creator
created_t	created_datetime	last_modified_t
last_modified_datetime	last_modified_by	last_updated_t
last_updated_datetime	product_name	abbreviated_product_name
generic_name	quantity	packaging
packaging_tags	packaging_fr	packaging_text
brands	brands_tags	categories
categories_tags	categories_fr	origins
origins_tags	origins_fr	manufacturing_places
manufacturing_places_tags	labels	labels_tags
labels_fr	emb_codes	emb_codes_tags
first_packaging_code_geo	cities	cities_tags
purchase_places	stores	countries
countries_tags	countries_fr	ingredients_text
ingredients_tags	ingredients_analysis_tags	allergens
allergens_fr	traces	traces_tags
traces_fr	serving_size	serving_quantity
no_nutrition_data	additives_n	additives
additives_tags	additives_fr	nutriscore_score
nutriscore_grade	nova_group	pnns_groups_1
pnns_groups_2	food_groups	food_groups_tags
food_groups_fr	states	states_tags
states_fr	brand_owner	environmental_score_score
environmental_score_grade	nutrient_levels_tags	product_quantity
owner	data_quality_errors_tags	unique_scans_n
popularity_tags	completeness	last_image_t
last_image_datetime	main_category	main_category_fr
image_url	image_small_url	image_ingredients_url
image_ingredients_small_url	image_nutrition_url	image_nutrition_small_url
energy_kj_100g	energy_kcal_100g	energy_100g
energy_from_fat_100g	fat_100g	saturated_fat_100g
butyric_acid_100g	caproic_acid_100g	caprylic_acid_100g
capric_acid_100g	lauric_acid_100g	myristic_acid_100g
palmitic_acid_100g	stearic_acid_100g	arachidic_acid_100g
behenic_acid_100g	lignoceric_acid_100g	cerotic_acid_100g
montanic_acid_100g	melissic_acid_100g	unsaturated_fat_100g
monounsaturated_fat_100g	omega_9_fat_100g	polyunsaturated_fat_100g
omega_3_fat_100g	omega_6_fat_100g	alpha_linolenic_acid_100g
eicosapentaenoic_acid_100g	docosahexaenoic_acid_100g	linoleic_acid_100g
arachidonic_acid_100g	gamma_linolenic_acid_100g	dihomo_gamma_linolenic_acid_100g
oleic_acid_100g	elaidic_acid_100g	gondoic_acid_100g
mead_acid_100g	erucic_acid_100g	nervonic_acid_100g
trans_fat_100g	cholesterol_100g	carbohydrates_100g
sugars_100g	added_sugars_100g	sucrose_100g
glucose_100g	fructose_100g	lactose_100g
maltose_100g	maltodextrins_100g	starch_100g
polyols_100g	erythritol_100g	fiber_100g
soluble_fiber_100g	insoluble_fiber_100g	proteins_100g
casein_100g	serum_proteins_100g	nucleotides_100g
salt_100g	added_salt_100g	sodium_100g
alcohol_100g	vitamin_a_100g	beta_carotene_100g
vitamin_d_100g	vitamin_e_100g	vitamin_k_100g
vitamin_c_100g	vitamin_b1_100g	vitamin_b2_100g
vitamin_pp_100g	vitamin_b6_100g	vitamin_b9_100g
folates_100g	vitamin_b12_100g	biotin_100g
pantothenic_acid_100g	silica_100g	bicarbonate_100g
potassium_100g	chloride_100g	calcium_100g
phosphorus_100g	iron_100g	magnesium_100g
zinc_100g	copper_100g	manganese_100g
fluoride_100g	selenium_100g	chromium_100g
molybdenum_100g	iodine_100g	caffeine_100g
taurine_100g	ph_100g	fruits_vegetables_nuts_100g
fruits_vegetables_nuts_dried_100g	fruits_vegetables_nuts_estimate_100g	fruits_vegetables_nuts_estimate_from_ingredients_100g
collagen_meat_protein_ratio_100g	cocoa_100g	chlorophyl_100g
carbon_footprint_100g	carbon_footprint_from_meat_or_fish_100g	nutrition_score_fr_100g
nutrition_score_uk_100g	glycemic_index_100g	water_hardness_100g
choline_100g	phyloquinone_100g	beta_glucan_100g
inositol_100g	carnitine_100g	sulphate_100g
nitrate_100g	acidity_100g	

TABLE 2 – Tableau des colonnes du fichier CSV (variante **fr**)

Revenir en haut

valeurs_manquantesTut	
cities	3600423
allergens_fr	3600423
additives	3600423
nutrition-score-uk_100g	3600423
elaidic-acid_100g	3600415
glycemic-index_100g	3600415
chlorophyl_100g	3600414
erucic-acid_100g	3600411
water-hardness_100g	3600410
caproic-acid_100g	3600407
gamma-linolenic-acid_100g	3600405
nervonic-acid_100g	3600403
lignoceric-acid_100g	3600402
dihomo-gamma-linolenic-acid_100g	3600401
caprylic-acid_100g	3600399
cerotic-acid_100g	3600398
capric-acid_100g	3600397
acidity_100g	3600397
mead-acid_100g	3600394
myristic-acid_100g	3600394
montanic-acid_100g	3600392
melissic-acid_100g	3600388
stearic-acid_100g	3600387
lauric-acid_100g	3600383
butyric-acid_100g	3600380
nucleotides_100g	3600371
palmitic-acid_100g	3600354
beta-glucan_100g	3600351
carnitine_100g	3600343
casein_100g	3600332
gondoic-acid_100g	3600328
behenic-acid_100g	3600327
maltose_100g	3600322
serum-proteins_100g	3600319
fructose_100g	3600315
beta-carotene_100g	3600309
inositol_100g	3600299
added-salt_100g	3600296
oleic-acid_100g	3600284
nitrate_100g	3600283
glucose_100g	3600275
sulphate_100g	3600268
erythritol_100g	3600266
unsaturated-fat_100g	3600246
choline_100g	3600243
omega-9-fat_100g	3600215
eicosapentaenoic-acid_100g	3600178
arachidonic-acid_100g	3600174
maltodextrins_100g	3600168
sucrose_100g	3600160
silica_100g	3600159
arachidic-acid_100g	3600145
collagen-meat-protein-ratio_100g	3600104
taurine_100g	3600059
docosaheptaenoic-acid_100g	3600033
chromium_100g	3599974
carbon-footprint_100g	3599948
molybdenum_100g	3599869
linoleic-acid_100g	3599789
ph_100g	3599753
fluoride_100g	3599680
starch_100g	3599670
bicarbonate_100g	3599414
fruits-vegetables-nuts-dried_100g	3599188
energy-from-fat_100g	3599166
alpha-linolenic-acid_100g	3599109
omega-6-fat_100g	3599106
lactose_100g	3598839

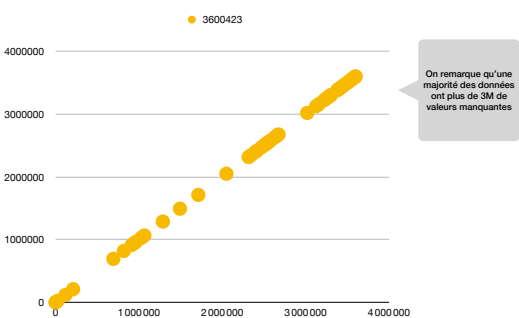
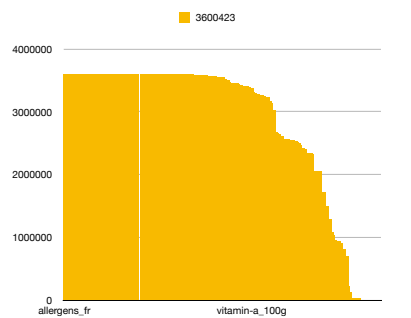


FIGURE 7 – Visualisation des occurrences des données manquantes - Excel

Revenir en haut

8 Bibliographie

Références

- [1] Data.gouv, *Open Food Facts : Produits alimentaires - Ingrédients, nutrition, labels*, <https://www.data.gouv.fr/fr/datasets/open-food-facts-produits-alimentaires-ingredients-nutrition-labels/>.
- [2] Talend, *ELT vs ETL : Comprendre les différences*, <https://www.talend.com/fr/resources/elt-vs-etl/>.
- [3] Nalron, *Project Public Health Study Notebook*, https://github.com/nalron/project_public_health_study/blob/french_version/p3_notebook01.ipynb.
- [4] Logilab, *Article sur le Blog de Logilab*, <https://www.logilab.fr/blogentry/13252264>.
- [5] Open Food Facts, *Données Open Food Facts*, <https://fr.openfoodfacts.org/data>.

Résumé du rapport

Dans un contexte où l'accès à une information fiable et transparente sur les produits alimentaires devient essentiel pour les consommateurs, les autorités et les professionnels du secteur, ce projet s'inscrit dans l'étude et la caractérisation des données issues de la base collaborative **Open Food Facts**. Celle-ci regroupe des informations détaillées sur des milliers de produits alimentaires, permettant d'éclairer les choix de consommation et d'encourager une alimentation plus responsable.

L'objectif du projet est de fournir un **notebook Jupyter interactif et paramétrable**, destiné aux spécialistes de l'alimentaire, afin de faciliter l'exploration et l'analyse de ces données. Les étapes du projet ont consisté en l'exploration initiale des jeux de données, la conception d'une base de données sous **PostgreSQL**, ainsi que le développement de requêtes analytiques ciblées.

Les résultats permettent de mettre en évidence des tendances nutritionnelles, des anomalies dans les produits et d'ouvrir des perspectives vers des analyses avancées. Ce travail constitue une base utile pour des études futures intégrant des techniques d'intelligence artificielle.

Mots-clés : Open Food Facts, PostgreSQL, Jupyter Notebook, données alimentaires, open data, exploration de données, analyse nutritionnelle.