Université Abdelmalek Essaâdi
Ecole Nationale des Sciences Appliquées - Al-Hoceima
Département Mathématiques et Informatique

Filière: Digital Transformation & Artificial Intelligence
Classe : 2
Module: Big Data

Project

Open Food Facts Dataset: Exploration and Recommender System

Proposed by: Mohamed El Marouani

April 2025

I. Introduction

This project aims to provide a practical, hands-on application of the concepts and techniques covered in the Big Data course. Through this project, students will be challenged to implement a **data pipeline for exploration and recommender system building**.

Conducted within the framework of an academic project, this document outlines the overall context, the functional requirements, the technical architecture, and the expected deliverables. The goal is to simulate a real-world scenario in which students can experiment with large-scale data processing, storage, analysis, and machine learning techniques in a collaborative and structured manner.

II. Context

The project is based on an open dataset known as **Open Food Facts**, a collaborative and crowdsourced database containing detailed information on food products from around the world. This includes ingredients, allergens, nutritional values, labels, categories, and other metadata typically found on product packaging.

Due to the richness and diversity of the dataset, which spans a wide variety of product types and brands, it serves as a valuable resource for gaining insights into the food industry. Moreover, it provides a solid foundation for developing intelligent systems such as product recommendation engines—particularly in contexts like suggesting healthy alternatives or assembling recipe ingredients.

Several open-source initiatives have already been developed around this dataset (see: https://github.com/openfoodfacts), addressing aspects such as data storage, API services, analytics, search capabilities, and artificial intelligence applications.

Each food item in the dataset is described by a comprehensive set of attributes, including but not limited to: product name, image, common name, quantity, brand, packaging format, category tags, ingredients list, nutritional composition, Nutri-Score, and more.

The primary objective of this project is to design and implement a data-driven application that recommends food products based on a user-defined recipe. Users will be able to input a description or a list of ingredients, and the system will suggest relevant products that match the criteria—thus showcasing the integration of data ingestion, processing, and intelligent recommendation in a Big Data environment.

III. Functional requirements

The implementation of this project is structured around three main components, each addressing a critical stage in the development of a data-driven application:

- **A data pipeline** for acquiring, cleaning, enriching, and transforming raw data into a format suitable for analysis and machine learning.
- **A content-based recommender system** for suggesting relevant food products based on recipe ingredients and nutritional preferences.
- **A lightweight web application** to allow user interaction through a simple and intuitive interface.

Business Objective

The overarching business question that the system aims to address is the following:

Given that I am in a specific country and I want to prepare a particular recipe, based on an ingredient list or a textual description, and certain nutritional or dietary criteria, what are the best available food products that I can use to prepare it?

This objective will guide the entire system design, from data preparation to model selection and interface development.

1. Data Pipeline

The data pipeline is the backbone of the system, responsible for transforming the raw Open Food Facts dataset into a clean, structured, and feature-rich dataset that can be used by downstream analytical and machine learning components. The pipeline is composed of several sequential stages:

- **Raw Data Ingestion**: The dataset is initially retrieved from Open Food Facts in JSON, CSV, or MongoDB dump. This includes metadata such as country availability, nutrition facts, categories, and packaging information.
- **Data Cleaning**: Missing values, inconsistent units (e.g., grams vs. milliliters), duplicates, and anomalies are identified and corrected. This ensures uniformity and reliability in the dataset.
- **Feature Engineering**: New features are derived from existing ones to enhance the model's performance. For example, parsing ingredient lists into standardized tokens, generating binary flags for allergens, or computing product similarity metrics (e.g., cosine similarity between ingredient vectors).
- **Exploratory Data Analysis (EDA)**: Visual and statistical analyses are conducted to uncover patterns, detect biases, and better understand the distribution of nutritional scores, categories, and ingredient types. This phase helps in shaping the final design of the recommender engine.

2. Recommender System

The recommendation engine will leverage **content-based filtering**, a machine learning approach that relies solely on the attributes of the items (i.e., food products) rather than user interactions or collaborative behaviors.

Key characteristics of the system include:

- **Vectorization of Recipes and Products**: Recipes provided by users (as text or ingredients list) will be transformed into vector representations using techniques such as word embeddings (e.g., Word2Vec), or sentence embeddings (e.g., Sentence-BERT, LLMs Embed APIs). Food products will also be vectorized based on ingredients, categories, nutrition scores, and other relevant fields.
- **Similarity Computation**: Using cosine similarity or other distance metrics, the engine will match the user's input with the most similar food products in the dataset, tailored to the selected country and constraints.

- **Filtering by Criteria**: Additional filters (e.g., Nutri-Score thresholds, absence of allergens, eco-score) will be applied to refine and personalize recommendations according to user needs.

This approach allows the system to work without requiring historical user behavior or ratings, making it more adaptable to new users or less frequently used products.

3. Web Application

To provide a simple and interactive way to use the system, a web-based interface will be developed. The application will allow users to input:

- Their country (to filter available products).
- A recipe description or a list of ingredients.
- Additional preferences or constraints, such as:
    - Minimum or maximum Nutri-Score (e.g., prefer only A or B scores).
    - Allergen exclusions (e.g., gluten-free, nut-free).
    - Packaging preferences (e.g., plastic-free).
    - Eco-score or environmental impact criteria.

The web application will send user input to the backend, which triggers the recommendation engine and returns a list of suitable food products, optionally with links or images from the Open Food Facts database.

IV. Architecture

The architecture of the system is designed to be modular, efficient, and suitable for handling semi-structured Big Data. It is composed of three main layers: data processing, storage and analytics, and user interaction.

The data processing layer is powered by **Apache Spark**, which handles the ingestion, cleaning, transformation, feature engineering, and exploratory analysis of the raw Open Food Facts dataset. Spark's DataFrame and SQL APIs will be used to process large volumes of data efficiently, leveraging distributed computation. Once the data is prepared and enriched, it is stored in a **MongoDB database**, which provides a flexible and scalable NoSQL environment for managing the processed product data and its attributes. This choice is particularly suitable given the semi-structured nature of the data (e.g., nested fields, optional values).

The recommendation engine is implemented as a Spark job that reads from MongoDB, applies **content-based filtering** using vector similarity techniques, and generates ranked product suggestions based on user input.

The **web application layer** is designed to be lightweight and user-friendly, exposing a **single backend endpoint**—accessible via a RESTful API—through which users can submit queries. These queries include parameters such as the country, recipe description, ingredient list, and dietary constraints (e.g., Nutri-Score, allergens). The backend processes the request, invokes the recommendation engine, and returns the most relevant food products in JSON format. This architecture allows for easy scaling, efficient computation, and clean separation between data processing, storage, and user interaction components.

V. Deliverables

The deliverables of the project will be:

- Code: scripts, notebooks, database, …
- Report or presentation that summarize the different parts of the project.