Journal de bord — Projet “Recipe Finder” (Data Visualisation)

Projet réalisé dans le cadre du cours **Conception en Data Visualisation**, Master 2 MBFA.

Journal de bord — Projet *Recipe Finder*

Master 2 MBFA — Conception en Data Visualisation

---

1) Prise en main du dataset et premières explorations

Je commence le projet par une analyse complète du dataset de recettes afin de :

- Comprendre la structure et les types de données,
- Identifier les colonnes essentielles (ingrédients, temps, nutrition),
- Repérer les incohérences et formats non exploitables.

Je charge d’abord le dataset dans un notebook puis j’explore les colonnes.
Je remarque rapidement plusieurs problèmes :

- Beaucoup de listes sont stockées sous forme de chaînes de caractères type c("milk", "eggs"),
- de nombreuses valeurs nutritionnelles sont manquantes,
- certains temps de préparation sont aberrants.

Pour répondre à cela :

- Je développe une fonction dédiée parse_r_vector(),
- Je crée un script d’exploration pour valider la cohérence des données,
- J’ajoute les colonnes cook_time_min, prep_time_min et total_time_min.

Ces premières étapes me permettent d’obtenir une base propre, exploitable pour la suite du projet.

---

2) Construction du pipeline de données

L’objectif suivant est de transformer ce nettoyage manuel en un pipeline robuste.
Je crée ainsi un module data_pipeline.py contenant :

- La lecture du fichier,
- Le nettoyage automatisé de chaque colonne,
- La transformation des listes en objets Python,
- La normalisation des données nutritionnelles.

Je mets également en place des contrôles pour :

- Gérer les quantités ne correspondant pas aux ingrédients,
- Remplacer les valeurs incohérentes par "N/A",

- Eviter les erreurs lors de l'affichage dans Streamlit.

Le pipeline devient alors une fonction unique load_recipes(), réutilisable dans toute l'application.

---

3) Création de l'application Streamlit

Je mets en place la structure générale de l'application via app.py :

- Une navigation sur trois pages : **Home**, **Recipes**, **Global Analysis**,
- Un thème graphique verts,
- Un CSS pour améliorer l'esthétique générale (sliders, menus déroulants…).

Je construis une page d'accueil simple présentant les objectifs du projet et expliquant le fonctionnement de l'application.

Difficultés sur cette partie :

- Streamlit limite fortement les personnalisations visuelles,
- les suggestions des multiselects occupaient trop d'espace à l'écran.

---

4) Développement de la page Recettes (recherche par ingrédients)

L'objectif est de permettre une recherche intuitive selon les ingrédients disponibles.

Je mets en place :

- Un multiselect d'ingrédients,
- Une contrainte de minimum 3 ingrédients sélectionnés,
- Des filtres optionnels : temps maximal, calories max,
- Une sélection aléatoire de 10 recettes parmi les résultats,
- Des cartes cliquables affichant les informations principales.

Pour stabiliser l'interface, j'utilise st.session_state afin de conserver la sélection entre deux rafraîchissements.
J'ajoute également des messages d'avertissement lorsque la recette sélectionnée n'appartient plus aux résultats filtrés.

---

5) Ajout d'un second mode : recherche par nom de recette

Afin de rendre l'application plus flexible, j'ajoute un mode alternatif « *By recipe name* ».
Ce mode supprime tous les filtres et affiche uniquement la recette correspondant au mot-clé saisi.

Je crée :

- Un text_input pour saisir un nom partiel,
- Une recherche dynamique insensible à la casse,

- Un affichage direct de la fiche complète de la recette.

Challenges rencontrés :

- La selectbox proposait des valeurs prédéfinies,
- Certaines requêtes ne renvoyaient aucun résultat.

J'adapte donc la logique pour n'afficher aucune suggestion et pour tolérer les correspondances partielles.

---

6) Analyse globale : création de visualisations pertinentes

Je construis plusieurs graphiques interactifs :

1. **Recettes les plus populaires :** basé sur un score combinant note et nombre d'avis.
2. **Temps moyen par catégorie :** simplifié pour une meilleure lisibilité.
3. **Score composite Qualité / Temps / Calories :** utilisant une normalisation des indicateurs.
4. **Un fingerprint nutritionnel par catégorie (barplot)**
5. **Un scatter Calories vs Popularité limité**

---

7) Diffusion du projet : mise en ligne sur GitHub et Streamlit Cloud

Une fois l'application fonctionnelle, je procède à la mise en ligne.

Mise en ligne sur GitHub

Je rencontre une difficulté majeure :
Le dataset d'origine (≈ 650 Mo) dépasse la limite autorisée de GitHub (100 Mo).

Solutions testées :

- Git LFS → trop limité en quota,
- Compression → encore trop volumineux.

Solution retenue :

- Créer une **GitHub Release**,
- Y déposer un fichier CSV compressé,
- Adapter le pipeline pour télécharger le dataset depuis cette release lors de l'exécution.

Même après cela, Streamlit n'arrive pas à lire le CSV qui reste trop volumineux.

Ma dernière solution possible est donc de prendre uniquement un échantillon de ma base de données, réduisant alors considérablement sa taille.

Je conserve uniquement 40% de ma base afin qu'elle puisse se lancer correctement.