Myriam FARIDI - Louis FAYE - Alexandre FAUCOURT
Omar FAKLANI - Buket Meltem OZKAN

GOUVERNEMENT
Liberté
Égalité
Fraternité

Prévision de la demande vaccinale contre la grippe en France :

Modélisation, données et planification des stocks

Myriam FARIDI - Louis FAYE - Alexandre FAUCOURT
Omar FAKLANI - Buket Meltem OZKAN

Myriam FARIDI - Louis FAYE - Alexandre FAUCOURT
Omar FAKLANI - Buket Meltem OZKAN

Introduction

Comment anticiper efficacement la demande en vaccins contre la grippe par région, tranche d'âge ou population à risque afin d'éviter pénuries et surplus?

Chaque année, une mauvaise estimation peut laisser des populations vulnérables non protégées ou générer un gaspillage de doses.

Pour relever ce défi, des modèles prédictifs comme LightGBM, Prophet ou IBM Watson utilisent l'historique des vaccinations contre la grippe, les données démographiques et de santé publique, ainsi que les facteurs saisonniers et épidémiques.

Ces modèles permettent de comprendre et de modéliser la relation entre l'incidence grippale et la demande vaccinale, y compris via des proxys lorsque les données réelles sont partielles ou absentes. Les prévisions générées alimentent des tableaux de bord dynamiques, permettant d'ajuster les commandes et la distribution en temps réel, afin d'optimiser la gestion des stocks et d'assurer une meilleure couverture vaccinale.

Objectif général

Comment anticiper efficacement la demande vaccinale contre la grippe et optimiser la gestion des stocks de vaccins à l'échelle régionale et par tranche d'âge ? Ce projet répond à cet enjeu en construisant un modèle prédictif capable de prévoir les volumes de doses nécessaires pour ajuster les livraisons et éviter les ruptures.

Le pipeline repose sur des données publiques françaises (INSEE, Sentinelles, Data, SurSaUD, Météo-France, Santé Publique France) et suit un processus complet : ingestion et nettoyage des données, construction des variables explicatives, apprentissage supervisé, calibration et génération d'un plan de réassort logistique.

Myriam FARIDI - Louis FAYE - Alexandre FAUCOURT
Omar FAKLANI - Buket Meltem OZKAN

Données d'entrée utilisées

Les données proviennent de sources ouvertes et sont harmonisées par région et par mois:

- **Population** (*INSEE*):
  - Par région et tranche d'âge.
- **Incidence grippale** (IAS,*Sentinelles*):
  - Hebdomadaire, ajustée au format mensuel.
- Lieux de vaccination contre la grippe (pharmacies)
  - Pour la grippe.
- **Températures mensuelles (ODRÉ / Météo-France):**
  - Pour chaque station, transformées en un jeu consolidé avec températures maximales et minimales, associées aux coordonnées géographiques et filtrées sur la période 2020–2025.
- **Volumes de doses administrées, couverture et compagne (**Datagouv):
  - Structurés par jour, groupe et variable, puis agrégés mensuellement.

Toutes les données sont normalisées pour 100 000 habitants et converties dans un format tabulaire exploitable dans Power BI. Les tables sur la vaccination incluent : couverture vaccinale (2021–2024), campagnes annuelles, doses et actes, lieux de vaccination (pharmacies), et prévisions par région et tranche d'âge.

Rôle du proxy vaccinal (filet de sécurité)

Pour les périodes ou régions où les données réelles de vaccination sont absentes ou incomplètes, un **proxy de demande** est généré. Ce proxy repose sur :

- L'incidence Sentinelles décalée dans le temps (t−2 semaines ou t−1 mois),
- La saison
- Un poids par tranche d'âge.

Sans ce proxy, de nombreuses séries seraient plates (0), empêchant tout apprentissage et rendant les métriques inutiles. Dès que les données réelles sont disponibles, le proxy disparaît automatiquement.

Myriam FARIDI - Louis FAYE - Alexandre FAUCOURT
Omar FAKLANI - Buket Meltem OZKAN

Modèle de prédiction utilisé - LightGBM (GBDT tabulaire)

Le modèle principal est un **Light Gradient Boosting Machine (LGBM)**, idéal pour les données tabulaires hétérogènes :

- Rapide et robuste, adapté à l'entraînement sur de multiples séries régionales,
- Peu sensible au scaling, sans besoin de normalisation des colonnes,
- Capable de gérer les interactions et non-linéarités sans feature engineering lourd,
- Explicable via SHAP values, permettant d'identifier l'influence de chaque variable (campagne, météo, incidence, urgences…).

Pour stabiliser les prévisions, un modèle de série temporelle classique (SARIMAX ou Prophet) peut être ajouté, et les deux prévisions combinées par moyenne pondérée pour améliorer la robustesse des métriques comme le SMAPE.

Variables explicatives (features)

Chaque observation mensuelle (région × tranche d'âge) inclut :

- **Lags** : t−1, t−2, t−3, t−6, t−12, représentant les valeurs passées de doses et indicateurs de santé.
- **Moyennes mobiles (MA)** : MA2, MA3, MA6, MA12, capturant les tendances récentes.
- **Variables calendaires** : mois, année, période de campagne, saison hiver/été.
- **Exogènes** : température (max/min), incidence Sentinelles, passages aux urgences, données météo additionnelles.

Cette combinaison assure que le modèle intègre à la fois les tendances historiques, la saisonnalité et l'impact de facteurs externes.

Myriam FARIDI - Louis FAYE - Alexandre FAUCOURT
Omar FAKLANI - Buket Meltem OZKAN

Méthode d'évaluation

Le modèle est évalué en conditions réelles grâce à une validation rolling-origin, et deux métriques principales sont utilisées :

SMAPE (*Symmetric Mean Absolute Percentage Error*) :

$$\mathrm{SMAPE} = \frac{100}{n} \sum_{t=1}^{n} \frac{|F_t - A_t|}{(|A_t| + |F_t|)/2}$$

où A_t est la valeur réelle et F_t la prévision.

MAE (*Mean Absolute Error*) :

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n} |y_i - x_i|}{n} = \frac{\sum_{i=1}^{n} |e_i|}{n}$$

Où y_i est la prédiction et x_i la valeur réelle.

Préparation et traitement des données, modèle sémantique et DataViz.

Préparation et Traitement des Données

L'objectif de cette phase initiale était de consolider un ensemble hétérogène de sources de données relatives à la vaccination contre la grippe pour les rendre exploitables dans Power BI. Cette étape cruciale d'ETL (Extract, Transform, Load) a été entièrement réalisée à l'aide de l'éditeur **Power Query** pour transformer les données brutes en un actif fiable.

Le processus s'est articulé autour de plusieurs opérations clés :

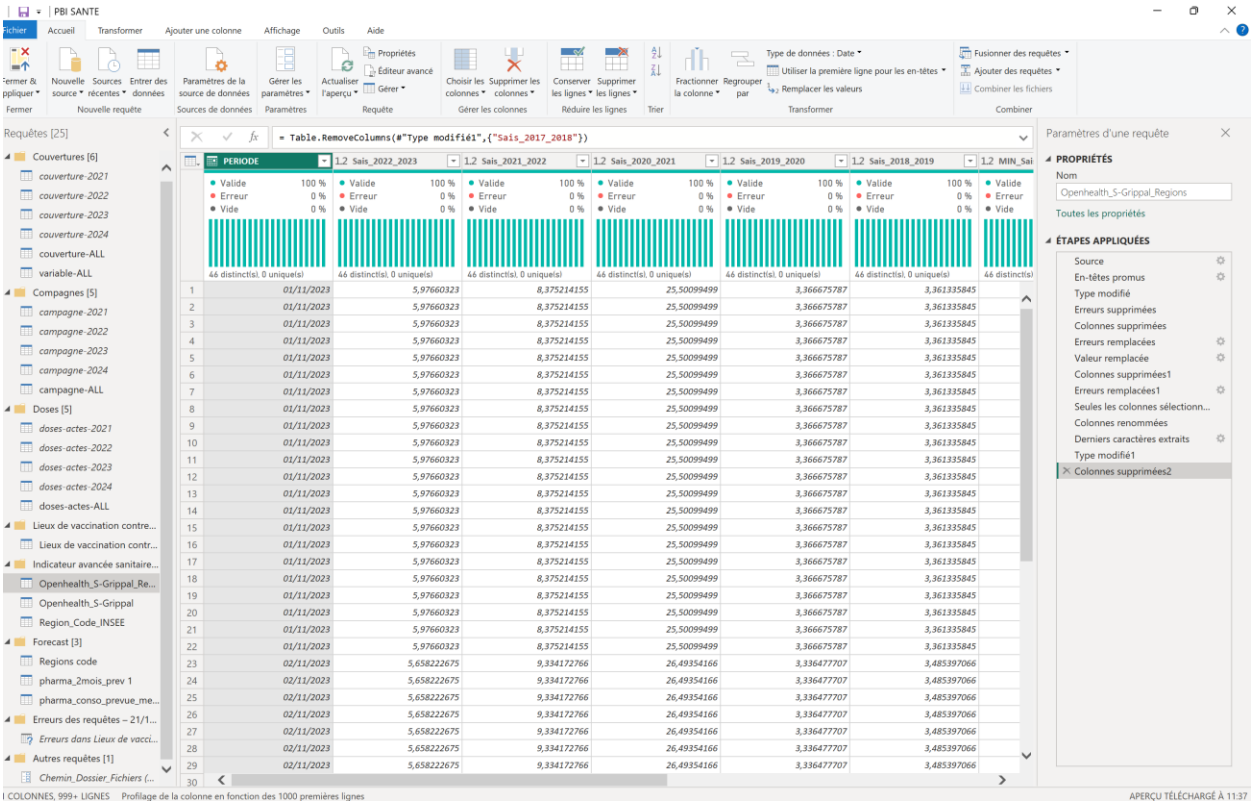
1. **Nettoyage et Standardisation :**
    a. **Harmonisation terminologique :** Un travail de standardisation a été essentiel pour assurer la cohérence. Par exemple, les libellés des tranches d'âge ont été uniformisés (« 0-64 » remplacé par « moins de 65 ans » et « 65+ » par « 65 ans et plus ») sur l'ensemble des fichiers.
    b. **Typage des données :** Chaque colonne a été rigoureusement typée (par exemple, passage des identifiants en texte, des valeurs numériques en nombre décimal et des dates au format Date) pour garantir la fiabilité des calculs futurs.

c. **Gestion des erreurs et des nuls :** Les lignes vides, les doublons et les valeurs aberrantes ont été filtrés ou corrigés pour ne conserver que des données exploitables.

2. **Transformation et Réorganisation :**
   a. **Fusion des données historiques :** Les données qui arrivaient en fichiers CSV annuels (comme les doses, la couverture, les campagnes 2021-2024) ont été consolidées. Via des requêtes d'ajout (*Append Queries*), elles ont été fusionnées en tables historiques uniques (ex: doses-actes-ALL, couverture-ALL), permettant une analyse temporelle continue.
   b. **Harmonisation géographique :** Pour les données météorologiques, l'enjeu majeur a été la normalisation des coordonnées géographiques (latitude et longitude). Celles-ci ont été arrondies pour permettre une jointure fiable entre les relevés de températures minimales et maximales provenant de stations identiques.
3. **Enrichissement et Création de Clés :**
   a. **Colonnes calculées :** Des informations à valeur ajoutée ont été créées. L'exemple le plus notable est le calcul de la **température moyenne** ((TempMax + TempMin) / 2) dans la table météo, un indicateur clé pour l'analyse prédictive.
   b. **Création de clés (Colonnes "Passerelles") :** Pour préparer le modèle sémantique, des **colonnes passerelles** ont été générées. Par exemple, un region_code standardisé a été extrait ou créé dans les différentes tables (couverture, prévisions, IAS) pour servir de clé de relation unique vers la table de dimension Regions code. De même, des colonnes de date propres ont été préparées pour les liaisons vers une future table de calendrier.

L'ensemble de ces données, une fois nettoyé, transformé et enrichi dans Power Query, forme un socle solide et cohérent pour la phase de modélisation.

Myriam FARIDI - Louis FAYE - Alexandre FAUCOURT
Omar FAKLANI - Buket Meltem OZKAN



Modèle Sémantique

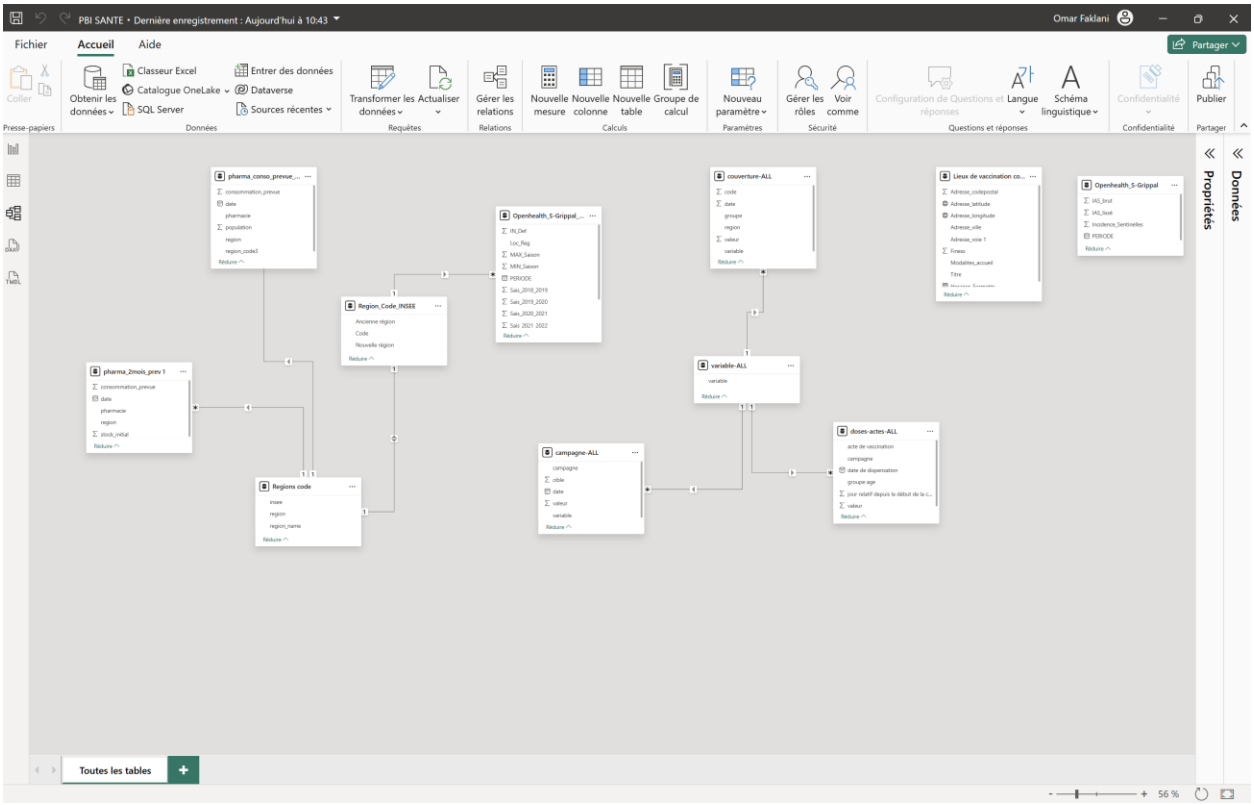
À partir de ces données préparées, un **modèle sémantique robuste** a été construit dans Power BI. Ce modèle est essentiel pour optimiser les performances et permettre des analyses croisées efficaces.

Le modèle s'articule autour d'un **schéma en étoile/flocon** :

- **Tables de faits :** Au centre, on trouve les tables contenant les mesures quantitatives, telles que doses-actes-ALL (suivi des vaccinations), couverture-ALL (taux de couverture) et Openhealth_5-Grippal (données épidémiologiques comme l'IAS).
- **Tables de dimensions :** Ces tables de faits sont reliées à des dimensions communes qui fournissent le contexte et permettent de filtrer les analyses. Les dimensions clés incluent Regions code (pour l'analyse géographique), campagne-ALL (pour la temporalité des campagnes) et variable-ALL (pour distinguer les types d'actes ou de doses).

Cette structuration garantit la cohérence des données et permet de naviguer de manière fluide entre les indicateurs épidémiologiques, les chiffres de vaccination et les contextes géographiques ou temporels.

Myriam FARIDI - Louis FAYE - Alexandre FAUCOURT
Omar FAKLANI - Buket Meltem OZKAN



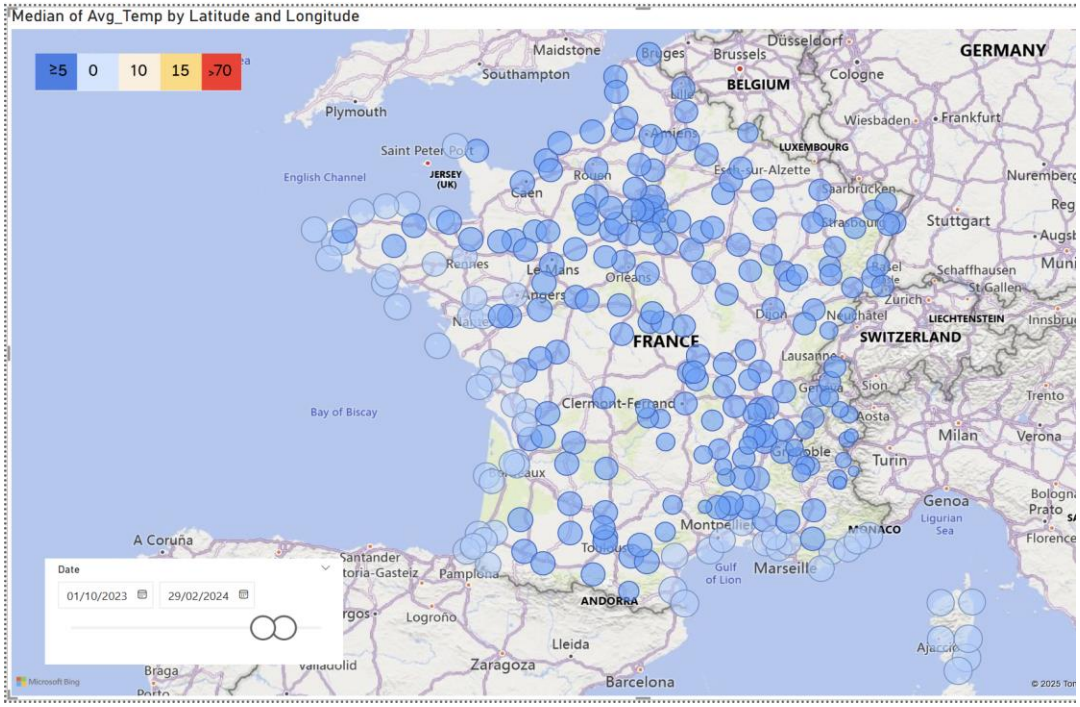
DataViz (Visualisation des Données)

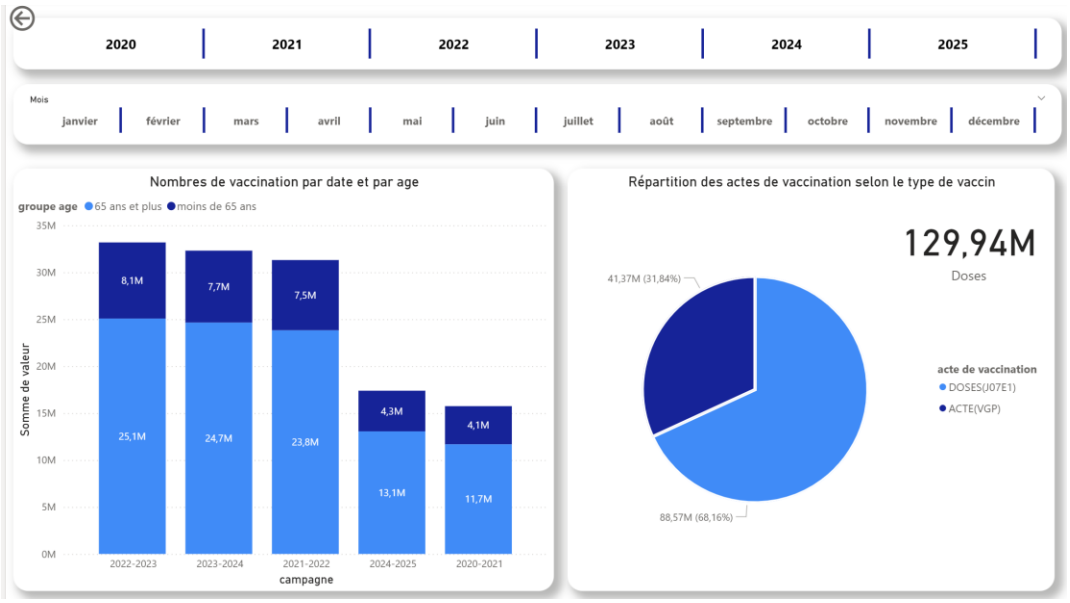
Enfin, la DataViz a été développée via des tableaux de bord interactifs pour explorer, analyser et communiquer les indicateurs clés du projet.

- **Suivi de la Vaccination :** Un tableau de bord principal suit les indicateurs de performance clés (KPI), affichant le **nombre total de doses ,**Il permet une analyse détaillée des actes de vaccination par **campagne** et par **groupe d'âge** (plus ou moins de 65 ans) grâce à un graphique en barres empilées. Un diagramme circulaire détaille également la répartition entre les différents types d'actes (DOSES vs ACTE).
- **Analyse Épidémiologique :** Plusieurs visualisations sont dédiées à l'**indicateur IAS** (Incidence des Syndromes Grippaux).
    - Des graphiques en courbes montrent l'**évolution saisonnière** (pic notable en novembre 2023 pour la saison 2023-24).
    - Des comparaisons historiques sont possibles grâce à un diagramme en anneau comparant l'IAS sur plusieurs saisons
    - Une analyse mensuelle distingue l'IAS **brut** de l'IAS **lissé**, révélant la saisonnalité du virus.

- **Analyse Géographique :** Une **vue cartographique** dynamique localise avec précision les **lieux de vaccination** (pharmacies) agréés sur le territoire français, y compris dans les départements et régions d'outre-mer. Ce visuel permet aux utilisateurs de filtrer par ville ou par code postal pour une analyse locale fine des ressources de vaccination disponibles.

Ces visualisations transforment les données complexes préparées et modélisées en informations actionnables, permettant un pilotage réactif et informé de la campagne vaccinale.





Myriam FARIDI - Louis FAYE - Alexandre FAUCOURT
Omar FAKLANI - Buket Meltem OZKAN

Interprétation des résultats

Les prévisions montrent :

- Des pics nets dans les régions à forte saisonnalité (ex. Bretagne, Bourgogne-Franche-Comté),
- Une demande vaccinale plus marquée pour les 65+ ans, cohérente avec la politique vaccinale,
- Une demande lissée pour les adultes de 18–64 ans.

L'analyse SHAP confirme l'importance des variables d'incidence et de température pour expliquer les variations régionales et temporelles.

Sorties principales

Les fichiers produits incluent :

- features.parquet : toutes les variables calculées,
- metrics_by_series.csv : performance par série région × tranche d'âge,
- forecast_reconciled_calibrated.parquet : prévisions calibrées par région et tranche d'âge,
- reassort_plan_from_latest.csv : plan de réassort basé sur les prévisions.

Le CSV final contient : date, région, tranche d'âge, prévision calibrée, moyenne historique, variation (%), quantité arrondie.

Recommandations pour la suite

- Conserver **LightGBM** comme modèle principal pour la production initiale.
- Ajouter un **baseline SARIMAX ou Prophet** pour ensemblage et comparaison des prévisions.
- Étendre l'horizon de **prévision jusqu'à 12 mois**.
- Intégrer des **scénarios météo et épidémiques** pour simuler différents contextes.
- Développer un **tableau de bord interactif** (**Superset ou Metabase**) pour suivre la demande en temps réel.

Myriam FARIDI - Louis FAYE - Alexandre FAUCOURT
Omar FAKLANI - Buket Meltem OZKAN

VacciBot

Vaccibot est un assistant conversationnel web développé dans le cadre d'un projet d'analyse et d'aide à la décision en santé publique.

L'objectif de ce chatbot est de proposer une interface de discussion interactive capable de:

- Dialoguer avec l'utilisateur,
- Détecter des informations clés (âges, symptômes, statuts vaccinal, ville)
- Et fournir des suggestions adaptées (pharmacies, spécialistes, centre de vaccination)

Important : le code est entièrement codée côté client, sans base de données ni API distante. Tout repose sur une analyse lexicale simple (regex) et une machine à états pour gérer le flux conversationnel.

Technos utilisés

Nous avons décider (par manque de temps) de coder ce chatbot entièrement réalisée en **HTML**, **CSS**, **JavaScript** (avec modules ES6 et un serveur **Python Flask** minimal).

Structure du projet

vaccibot/

├── index.html → Interface principale du chatbot

├── style.css → Design et mise en forme

└── src/

  ├── main.js → Point d'entrée de l'application

  ├── ui.js → Gestion de l'affichage des messages

  ├── state.js → Gestion de l'état global et variables utilisateur

  ├── intents.js → Analyse du texte et détection d'intentions

  ├── flow.js → Logique du dialogue (machine à états)

  ├── actions.js → Réponses et actions du bot

Myriam FARIDI - Louis FAYE - Alexandre FAUCOURT
Omar FAKLANI - Buket Meltem OZKAN

├── links.js → Liens Google Maps / Doctolib

├── pharmacy.js → Gestion et recherche de pharmacies

├── specialists.js → Association symptômes → spécialiste

└── utils.js → Fonctions utilitaires

Détails des fichiers

- **Index.html :**

-Contient la structure complète de l'interface utilisateur.
-Déclare la zone de discussion (`chat-body`), le champ de saisie et le formulaire d'envoi.
-Charge les fichiers `style.css` et les scripts JavaScript du dossier src/.
-Initialise le chatbot avec un message de bienvenue dès le chargement de la page.

- **Style.css** :

- Définit l'apparence visuelle du chatbot : **thème sombre**, bulles de messages, interface responsive.
- Gère les animations du bot (effet "typing…") et les transitions CSS.
- Principales classes : `.chat-container`, `.message.user`, `.message.bot`, `.typing`.

- **Pharmacies.csv** :

- Contient la **base locale** des pharmacies françaises (nom, adresse, ville, coordonnées, horaires, etc.).
- Données **nettoyées et normalisées** pour assurer la cohérence.

- **Main.js** :

- Point d'entrée du chatbot.
- Initialise le flux de discussion au chargement (`FLOW.start()`).
- Écoute la soumission du formulaire utilisateur et appelle la fonction principale de traitement

- **Ui.js** :

- Gère l'affichage dynamique des messages dans le chat.
-Gère le défilement automatique vers le bas après chaque message.

- **State.js** :

-Contient l'état global du chatbot (state) et les informations utilisateur.
-Contient aussi des listes de mots-clés (salutations, remerciements, symptômes, etc) pour enrichir la détection d'intentions et rendre plus intéligent le bot.

- **Intents.js** :

-Analyse le message utilisateur pour déterminer une intention à l'aide de mots-clés et d'expressions régulières.
-Renvoie une chaîne d'intention utilisé par flow.js pour orienter la suite du dialogue.

- **Flow.js** :

-Implémente la logique de conversation sous forme de machine à états.
-Chaque étape du dialogue correspond à un état distinct.
-Gère les transitions d'un état à l'autre et les actions à effectuer via actions.js.
-Garantit un dialogue fluide et logique sans redondance ni incohérence.

- **Actions.js** :

-Contient les actions par le bot selon le contexte de la conversation.
-Intéragit avec les autres modules (link.js, pharmacy.js, state.js) pour produire les réponses adaptées.

- **Link.js** :

-Génère et ouvre des liens externes pertinents et permet d'intégrer des ressources de santé externes (pharmacies, médecins, centre de vaccination).

- **Pharmacy.js** :

-Charge et traite les données du fichier pharmacies.csv.
-Fait le lien entre les données locales et la logique du bot (actions.js).

- **Specialist.js** :

-Identifie le type de spécialiste médical adapté selon les symptômes détectés. Utilisé dans actions.js pour générer les liens Doctolib correspondants.

- **Utils.js** :

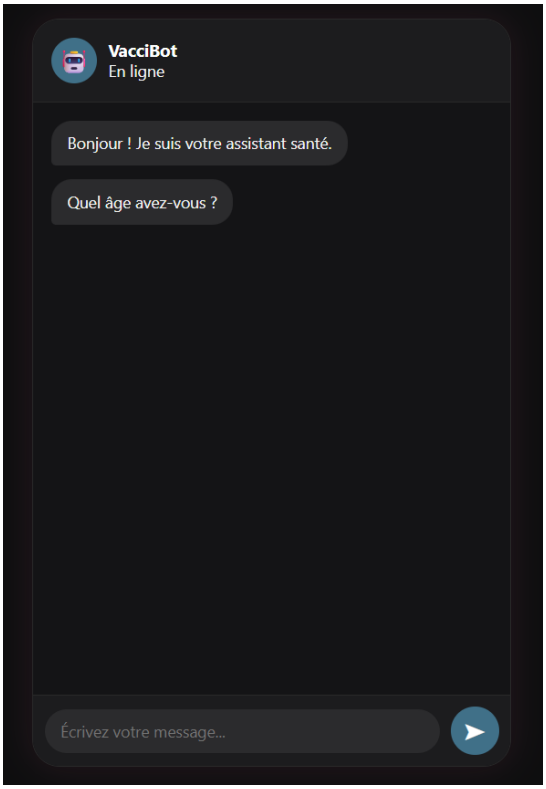
Myriam FARIDI - Louis FAYE - Alexandre FAUCOURT
Omar FAKLANI - Buket Meltem OZKAN

-Contient les fonctions d'aide réutilisables. Utilisé dans plusieurs modules pour améliorer la precision et la cohérence des traitements.

Enfin, le projet fonctionne entièrement côté navigateur mais l'utilisateur d'un serveur Flask local est nécessaire pour charger correctement le fichier pharmacies.csv via HTTP.

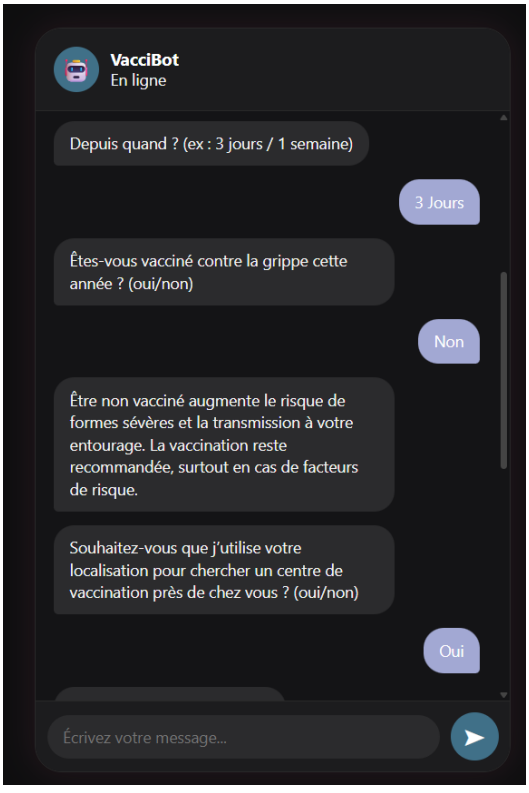
On devait ainsi taper dans notre terminal **python server.py** et ouvrrir le code à l'adresse localhost:8000.

Visuel

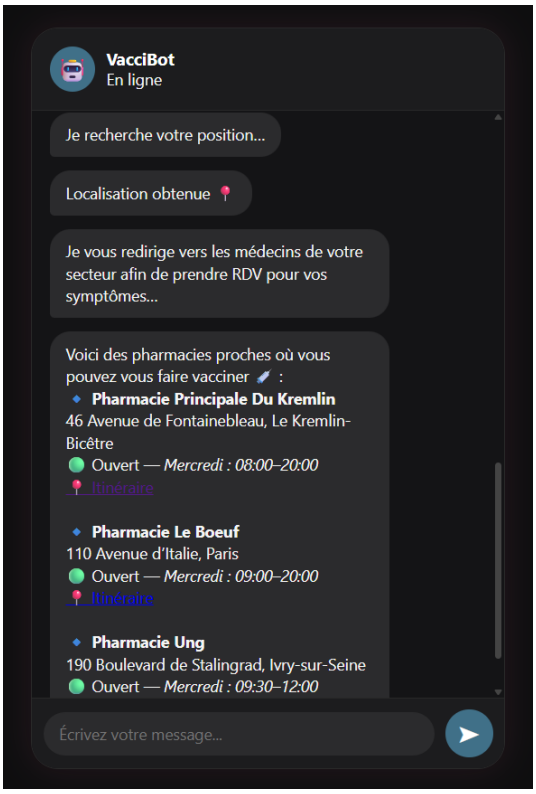


Myriam FARIDI - Louis FAYE - Alexandre FAUCOURT
Omar FAKLANI - Buket Meltem OZKAN

,



Myriam FARIDI - Louis FAYE - Alexandre FAUCOURT
Omar FAKLANI - Buket Meltem OZKAN



I. Amélioration

L'idée du chatbot était de créer un vrai chatbot et implémenter une IA de santé. Le chatbot est une application mobile qui serait téléchargé par les citoyens français. Et chaque données (symptomes, taux de vaccinations, etc...) serait retransmis dans une base de données destiné au ministère de la santé. De plus ont aurait orienté les citoyens (sans qu'ils ne s'en rendent compte) vers des pharmacies de vaccination avec un stock de vaccination décent. Ce qui permettrait aux établissement de gérer mieux leurs stocks.

Les technos qu'on pourrait utilisé pour mettre cette idée en action :

- **BDD** : PostgreSQL
- **API Rest** : FastAPI.
- **Implémentation de l'IA**.

Myriam FARIDI - Louis FAYE - Alexandre FAUCOURT
Omar FAKLANI - Buket Meltem OZKAN

Conclusion

Le modèle Vaccination Demand Forecast constitue une base solide pour anticiper la demande vaccinale et aider les décideurs à planifier les stocks. Il combine rigueur statistique, adaptabilité et transparence, et peut évoluer vers un outil d'aide à la décision en temps réel, intégrant données vaccinales, épidémiologiques et météorologiques.