

# Table des matières

<b>Introduction</b>	<b>2</b>
<b>1 Méthodologie</b>	<b>3</b>
1.1 Étapes du développement . . . . .	3
1.1.1 Prétraitement des données . . . . .	3
1.1.2 Réalisation du tableau de bord . . . . .	3
1.1.3 Réalisation de l'API pour la prédiction . . . . .	4
1.1.4 Hébergement du tableau de bord . . . . .	4
1.2 Outils . . . . .	4
1.3 Techniques utilisées . . . . .	4
<b>2 Fonctionnalités du BloodHub Dashboard</b>	<b>5</b>
<b>3 Difficultés rencontrées</b>	<b>5</b>
3.1 Variation des datasets et choix des feuilles de calcul . . . . .	5
3.2 Problème de cartographie pour la représentation des villes du Cameroun . . . . .	6
3.3 Problème lors de l'hébergement . . . . .	6
<b>Conclusion</b>	<b>7</b>

# Introduction

Le don de sang est un enjeu majeur de santé publique à l'échelle mondiale. Chaque année, des millions de personnes dépendent des transfusions sanguines pour traiter des maladies graves, effectuer des interventions chirurgicales ou faire face aux urgences médicales. Malgré son importance vitale, l'approvisionnement en sang repose sur la générosité des donateurs volontaires, rendant ainsi la gestion des stocks sanguins complexe et cruciale. Dans ce contexte, l'analyse des données liées aux donateurs de sang permet d'optimiser la collecte et la distribution des poches sanguines. Grâce à l'essor des technologies de l'information et des techniques avancées d'analyse de données, il est désormais possible d'exploiter ces informations pour prédire les besoins, améliorer le suivi des donateurs et renforcer l'efficacité des campagnes de sensibilisation. Ce rapport s'inscrit dans le cadre du challenge portant sur l'étude et la modélisation des comportements des donateurs de sang. L'objectif principal est d'explorer les données disponibles afin de mieux comprendre les tendances de donation, identifier les profils des donateurs réguliers et concevoir un modèle prédictif permettant d'anticiper les futurs dons. À travers ce travail, nous espérons apporter une contribution pertinente à la gestion des dons de sang en exploitant les outils de la science des données pour une meilleure planification et un approvisionnement plus efficace en produits sanguins.

# 1 Méthodologie

La mise en place d'un projet aussi structurant que la prédiction des besoins en don de sang repose sur une approche rigoureuse. La présente section détaille les différentes étapes du développement, les outils employés et les techniques adoptées pour assurer un résultat fiable et exploitable.

## 1.1 Étapes du développement

Le développement de notre solution s'est articulé autour de trois grandes étapes : le prétraitement des données, la réalisation du tableau de bord et la mise en place de l'API pour la prédiction.

### 1.1.1 Prétraitement des données

Afin de garantir une base de données propre et cohérente, plusieurs opérations ont été effectuées :

- **Fusion et harmonisation des données** : uniformisation des noms de colonnes (remplacement des espaces par des underscores, conversion en minuscules, standardisation des variables) et fusion des fichiers sources.
- **Suppression des colonnes non pertinentes** : les colonnes contenant un trop grand nombre de valeurs manquantes ou jugées inutiles ont été supprimées.
- **Traitement des valeurs manquantes** : les valeurs absentes ont été imputées par des moyennes ou des valeurs calculées.
- **Standardisation et nettoyage** : uniformisation des formats, correction des erreurs et vérification des données temporelles.

### 1.1.2 Réalisation du tableau de bord

Un tableau de bord interactif a été conçu pour faciliter la visualisation et l'analyse des données. Cet outil permet d'explorer les tendances et de mieux comprendre les besoins en don de sang.

Pour garantir une expérience utilisateur fluide et une interactivité optimale, nous avons utilisé **Dash**, un framework Python spécialisé dans la création d'interfaces web interactives. En exploitant ses *callbacks*, nous avons réussi à créer une interface réactive capable de s'adapter dynamiquement aux entrées des utilisateurs.

Sur le plan de l'esthétique et de la lisibilité, nous avons opté pour **Pico**, un framework CSS léger, permettant d'obtenir un design simple et efficace sans surcharge de styles complexes.

### 1.1.3 Réalisation de l'API pour la prédiction

L'API constitue le cœur du système de prédiction. Son rôle est de prendre en entrée des données utilisateur et de retourner une prédiction basée sur les modèles entraînés. Pour cela, nous avons mis en place une API RESTful avec **Flask**, garantissant une communication fluide entre le backend et les services clients.

L'API permet ainsi d'envoyer des requêtes avec les informations nécessaires et d'obtenir une prédiction en retour, assurant une intégration facile avec d'autres applications.

### 1.1.4 Hébergement du tableau de bord

Une fois le système d'analyse mis en place, il était nécessaire de le rendre accessible en ligne. Pour cela, nous avons opté pour l'hébergement sur un serveur **Contabo**, reconnu pour ses performances et sa fiabilité. Ce choix nous a permis de déployer efficacement l'API et le tableau de bord, garantissant un accès rapide et stable aux utilisateurs finaux.

L'hébergement sur un serveur dédié a facilité l'intégration continue et la maintenance, assurant ainsi une disponibilité constante du service.

## 1.2 Outils

Différents outils ont été utilisés pour chaque étape du projet :

- **Python** : Utilisé pour le prétraitement des données et le développement de l'API.
- **Pandas et NumPy** : Manipulation et traitement des données.
- **Dash** : Outil principal pour la création du tableau de bord interactif.
- **Flask** : Mise en place de l'API permettant d'accéder aux prédictions.
- **Scikit-learn** : Entraînement des modèles de prédiction.
- **Pico** : Framework CSS simple pour améliorer l'interface visuelle.

## 1.3 Techniques utilisées

Les techniques employées pour le prétraitement et l'analyse des données comprennent :

- **Imputation des valeurs manquantes** : remplacement des valeurs absentes par des moyennes pour maintenir la cohérence des données.
- **Standardisation** : normalisation des noms de colonnes et nettoyage des entrées utilisateur.
- **Modélisation** : Entraînement de modèles de machine learning pour prédire les besoins en don de sang.

## 2 Fonctionnalités du BloodHub Dashboard

L'objectif du **BloodHub Dashboard** est d'offrir un tableau de bord à la fois fonctionnel et intuitif, permettant une meilleure compréhension et gestion des données liées aux dons de sang.

- **Visualisation des foyers de donneurs** : Une cartographie interactive permettant d'identifier les zones à forte concentration de donneurs au sein des villes.
- **Analyse des conditions de santé des donneurs** : Mise en avant des problèmes de santé les plus récurrents chez les donneurs, facilitant ainsi l'évaluation des tendances médicales.
- **Définition du profil de donneur idéal** : Grâce à des filtres avancés, possibilité d'affiner les caractéristiques du donneur recherché (ex. : déterminer le meilleur profil pour une femme sans antécédents médicaux).
- **Comparaison des dons par arrondissement** : Une analyse comparative permettant de mesurer la contribution de chaque arrondissement par rapport aux dons globaux.
- **Impact de chaque arrondissement dans la campagne** : Identification des zones les plus actives et évaluation de leur apport dans la dynamique globale des dons.

## 3 Difficultés rencontrées

Comme dans tout projet complexe, plusieurs obstacles se sont dressés sur notre chemin tout au long du développement du **BloodHub Dashboard**. Toutefois, chaque difficulté a été abordée de manière pragmatique, en cherchant des solutions adaptées aux contraintes rencontrées. Voici un aperçu des principaux problèmes et des solutions que nous avons mises en place pour les surmonter.

### 3.1 Variation des datasets et choix des feuilles de calcul

L'un des premiers problèmes rencontrés fut la **variabilité des datasets**. En effet, notre jeu de données se composait de plusieurs feuilles de calcul contenant des informations différentes, ce qui a rendu la gestion des données complexe. Chaque feuille avait des colonnes et des structures variées, et il n'était pas clair laquelle choisir comme base pour notre analyse.

**Solution** : Pour résoudre ce problème, nous avons adopté une approche de travail sur chaque feuille de calcul individuellement, en fonction des données qu'elle contenait. Cela nous a permis d'examiner chaque cas de manière plus ciblée et de traiter les données de façon plus structurée. En analysant les informations disponibles dans chaque feuille, nous avons pu opter pour la feuille qui correspondait le mieux aux objectifs de notre projet.

### 3.2 Problème de cartographie pour la représentation des villes du Cameroun

Un autre challenge majeur a été la cartographie des données géographiques, notamment la représentation des villes du Cameroun. Nous avons envisagé plusieurs bibliothèques Python pour la gestion de cartes interactives, mais nous avons rapidement constaté que certaines d'entre elles **manquaient de performance** et n'étaient pas adaptées à nos besoins, en particulier en termes de vitesse de traitement.

**Solution** : Afin de pallier cette limitation, nous avons opté pour l'utilisation de l'**API Google Maps**, qui nous a permis d'extraire les coordonnées géographiques des zones présentes dans notre dataset. Cette API s'est révélée **plus rapide** et efficace pour gérer de grandes quantités de données géographiques, ce qui a facilité la représentation des villes sur notre carte interactive.

### 3.3 Problème lors de l'hébergement

Lors de la phase de déploiement, nous avons rencontré des problèmes de **restriction d'hébergement**. Nous avons testé plusieurs plateformes populaires comme **Heroku** et **PythonAnywhere**. Cependant, ces services gratuits avaient des limitations, notamment en termes d'espace de stockage. Par exemple, sur PythonAnywhere, l'espace de stockage du compte gratuit était insuffisant pour installer toutes les dépendances mentionnées dans le fichier `requirements.txt`, ce qui a rendu l'hébergement difficile.

**Solution** : Face à ces contraintes financières et techniques, nous avons pris la décision d'héberger le tableau de bord sur un serveur **Contabo**, auquel nous avons déjà accès. Ce choix nous a permis de déployer facilement l'application, tout en garantissant une gestion optimale des ressources serveur, sans être soumis aux limitations des services gratuits.

## Conclusion

Le développement du **BloodHub Dashboard** a été une expérience enrichissante, marquée par une série de défis techniques, mais également par la mise en place de solutions efficaces permettant de surmonter ces obstacles. En adoptant une approche méthodique et en utilisant les outils appropriés, nous avons réussi à construire une solution robuste et performante pour la prédiction des besoins en don de sang, avec une interface utilisateur interactive et un système d'API de prédiction fonctionnel. L'hébergement de l'application, bien que contraint par des limitations budgétaires et techniques, a été résolu grâce à l'utilisation d'un serveur dédié Contabo, garantissant une disponibilité stable du service. Le projet a non seulement permis de mettre en œuvre des compétences techniques en traitement de données, en développement web, et en déploiement, mais il a aussi offert une occasion de renforcer notre capacité à résoudre des problèmes complexes. Nous espérons que ce tableau de bord pourra, à terme, contribuer à une meilleure gestion des besoins en don de sang et offrir des perspectives intéressantes pour de futurs travaux dans le domaine de la santé.