

# Rapport de Projet

## Prédiction des Annulations de Rendez-vous pour Doctolib

*Master 2 DATA & IA*

**Présenté par :**

FOUEJIO Francky Joël

**Encadrants Académique :**

- M. DAOUDI Ahmed
- M. AMAR Mehdaoui

## SOMMAIRE :

SOMMAIRE : .....	2
<b>1. Introduction générale.....</b>	<b>4</b>
<b>2. Contexte et problématique métier .....</b>	<b>4</b>
<b>3. Objectifs du projet.....</b>	<b>5</b>
3.1 Objectifs techniques .....	5
3.2 Objectifs opérationnels.....	5
3.3 Objectifs stratégiques .....	6
<b>4 Gestion de projet .....</b>	<b>6</b>
4.1 Contexte général et philosophie de gestion .....	6
4.2 Organisation du projet Agile .....	6
4.3 Rôles, équipe et dynamique collective .....	8
4.4 Backlog fonctionnel.....	9
4.5 Organisation des sprints .....	12
4.6 Planification des sprints .....	12
4.7 Conclusion .....	15
<b>5. Collecte et préparation des données .....</b>	<b>15</b>
5.1 Sources de données utilisées .....	15
5.2 Nettoyage et traitement des données .....	17
5.3 Analyse exploratoire des données .....	20
<b>6. Conception du modèle IA .....</b>	<b>23</b>
6.1 Choix des modèles.....	23
6.2 Conception du modèle XGBoost.....	23
6.2.a Oversampling ou Suréchantillonnage .....	23
6.2.b Entraînement, optimisation et sauvegarde du modèle. ....	25
6.3 Interprétabilité du modèle (SHAP).....	28
6.4 Analyse des résultats .....	29
<b>7. Application Streamlit et l'API / Base de données .....</b>	<b>31</b>
7.1 Application Streamlit .....	31
7.2 Conception de l'API et de la base de données .....	32
<b>8. Architecture Cloud .....</b>	<b>35</b>

<b>8.1 Architecture générale</b> .....	36
<b>8.2 Architecture Cloud AWS</b> .....	37
8.2.a Présentation.....	37
8.2.b Schéma technique .....	38
8.2.c Description des composants .....	38
8.2.d Chiffrage estimatif .....	41
<b>8.3 Architecture Cloud Azure</b> .....	43
8.3.a Présentation.....	43
8.3.b Schéma technique .....	45
8.3.c Description des composants .....	45
8.3.d Chiffrage estimatif .....	48
<b>8.4 Choix du fournisseur cloud</b> .....	51
<b>8.5 Conclusion</b> .....	51

## 1. Introduction générale

La digitalisation du secteur de la santé est un mouvement profond, amorcé depuis plusieurs années, et s'inscrivant dans une volonté sociétale d'améliorer l'accès, la qualité et l'efficacité des soins médicaux. Dans cette transformation, Doctolib joue un rôle central en tant que plateforme de prise de rendez-vous en ligne, permettant à des millions de patients d'accéder facilement à un professionnel de santé. Fondée en 2013, cette entreprise française a su imposer un modèle de service novateur, connecté et intuitif, qui simplifie le parcours de soins, réduit les frictions administratives et fluidifie la gestion des plannings médicaux.

Cependant, derrière ce succès technologique, un défi majeur subsiste : le phénomène des annulations de rendez-vous, et plus particulièrement des annulations tardives ou des absences sans préavis (appelées "no-shows"). Ces comportements perturbent l'organisation des praticiens, gênent la prise en charge des patients, et nuisent à l'efficacité globale du système de soins. En moyenne, des dizaines de milliers de rendez-vous sont annulés chaque jour sur Doctolib sans qu'aucune solution ne soit automatiquement proposée pour combler ces absences. À l'échelle du territoire, cela représente une perte considérable de temps médical et une insatisfaction croissante des patients.

Ce projet répond à cette problématique en explorant les apports de l'intelligence artificielle et de la science des données pour prévoir les risques d'annulation. Il propose une démarche complète : collecte et analyse des données, développement d'un modèle de machine learning, mise en place d'une architecture cloud sécurisée et scalable (AWS/Azure), intégration dans un environnement métier réel. L'objectif final est de fournir à Doctolib un outil d'aide à la décision pour anticiper, agir et optimiser la gestion des rendez-vous. En créant un système prédictif fiable et éthique, ce projet contribue à la transformation digitale responsable du système de santé.

## 2. Contexte et problématique métier

Le système de santé français, reconnu pour sa qualité de soins, est confronté à de fortes tensions organisationnelles. La désertification médicale, le vieillissement de la population, l'augmentation des pathologies chroniques et les attentes croissantes des usagers mettent à mal les modèles traditionnels de prise en charge. Dans ce contexte, l'efficacité devient un enjeu prioritaire. Pourtant, un facteur souvent sous-estimé grève la productivité des professionnels de santé : les absences non prévues aux rendez-vous.

Ce phénomène d'absentéisme médical, en augmentation constante, se traduit par des pertes de temps considérables, une complexification des plannings, et une expérience dégradée pour les patients en attente. Selon les données internes de Doctolib et les rapports institutionnels, on estime que jusqu'à 10 % des rendez-vous sont concernés par des no-shows. Ce taux peut grimper à 20 % dans certaines spécialités ou zones géographiques.

Les conséquences sont multiples :

- **Désorganisation des praticiens** : une plage horaire vide représente un manque à gagner, mais aussi une perte de temps précieux pour d'autres patients.
- **Augmentation des délais de rendez-vous** : un rendez-vous non honoré aurait pu être attribué à une personne en attente urgente.
- **Surcoûts administratifs** : les secrétariats sont mobilisés pour recontacter les patients et réorganiser les plannings.
- **Inégalité d'accès aux soins** : les patients fragiles ou en situation de priorité peuvent être injustement pénalisés.

Face à ce constat, la plateforme Doctolib doit renforcer ses outils de prévention et d'anticipation. L'intégration d'un système de prédiction des annulations permettrait, par exemple :

- d'envoyer des rappels ciblés aux patients à risque,
- de proposer du surbooking intelligent (comme dans le transport aérien),
- de notifier les praticiens en cas de forte probabilité d'annulation,
- d'adapter dynamiquement la gestion des plages horaires.

Ces leviers, combinant data science, automatisation et sens clinique, permettent une gestion médicale plus réactive, éthique et performante.

### 3. Objectifs du projet

Le projet s'articule autour de trois dimensions essentielles : technique, opérationnelle et stratégique. Chacune d'elles est étroitement liée à la mission de Doctolib : offrir un accès plus rapide, plus fluide et plus personnalisé à la santé.

#### 3.1 Objectifs techniques

- Concevoir un pipeline de données robuste : ingestion, traitement, nettoyage, enrichissement, équilibrage.
- Réaliser une analyse exploratoire détaillée des variables influentes sur les annulations (ex. : âge, météo, antécédents).
- Choisir et entraîner un modèle de classification binaire performant, avec métriques d'évaluation pertinentes (AUC, précision, rappel, F1-score).
- Implémenter des outils d'interprétabilité comme SHAP afin de comprendre les déterminants des prédictions.
- Déployer le modèle dans un environnement cloud sécurisé (AWS ou Azure) avec API REST et monitoring.

#### 3.2 Objectifs opérationnels

- Réduire le taux d'annulations non signalées de 15 à 30 %.

- Doter les praticiens d'outils d'aide à la décision : alertes, tableaux de bord, notifications intelligentes.
- Permettre une meilleure gestion du temps médical, en revalorisant les créneaux à forte incertitude.
- Offrir aux secrétariats un outil d'optimisation des plannings en temps réel.

### 3.3 Objectifs stratégiques

- Consolider l'image de Doctolib en tant qu'acteur innovant, responsable et orienté vers les données.
- Renforcer la fidélisation des professionnels de santé par un service plus efficace.
- Poser les fondations d'un système intelligent, extensible à d'autres cas d'usage (gestion des urgences, planification opératoire, hôpital intelligent).

## 4 Gestion de projet

### 4.1 Contexte général et philosophie de gestion

Dès le lancement du projet "Prédiction des annulations de rendez-vous pour Doctolib", nous avons fait un choix fort : adopter une approche agile, inspirée du cadre Scrum, afin de structurer notre travail tout en nous laissant la flexibilité nécessaire pour réagir aux imprévus. Pourquoi ce choix ? Parce qu'en matière d'intelligence artificielle, les besoins peuvent évoluer rapidement en fonction des données disponibles, des performances du modèle ou des retours métier. Une méthode rigide n'aurait pas permis cette réactivité.

Nous avons ainsi privilégié une organisation itérative, incrémentale et collaborative, où chaque membre de l'équipe a pu s'exprimer, proposer, ajuster et apprendre. La gestion de projet n'a pas été une simple affaire de planning : elle a constitué le socle vivant de notre dynamique d'équipe.

### 4.2 Organisation du projet Agile

**Le projet est structuré autour des éléments suivants :**

- **Backlog Produit** : Le backlog produit est l'élément central de la gestion de projet Scrum. Il s'agit d'une liste dynamique et priorisée de tout ce qui doit être réalisé dans le projet

Le backlog est organisé selon deux niveaux hiérarchiques principaux :

- ✓ **Les EPICs** : grands ensembles fonctionnels ou thématiques représentant les grandes phases ou composants du projet (ex. : "Modélisation IA", "Architecture cloud", "Suivi de projet").

- ✓ **Les User Stories** : tâches spécifiques exprimées du point de vue de l'utilisateur ou du métier, décrivant une fonctionnalité ou une action à réaliser (ex. : "En tant que data scientist, je veux entraîner un modèle prédictif afin de détecter les rendez-vous à risque").

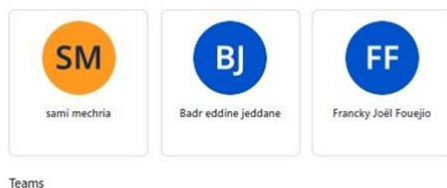
Chaque user story est associée à :

- ❖ Un **rôle utilisateur** (qui fait l'action),
- ❖ Un **objectif clair** (ce qui est attendu),
- ❖ Et une **valeur métier** (le pourquoi).

- **Sprints** : Le travail est divisé en sprints de 02 semaines. Chaque sprint donne lieu à une planification, une revue, et une rétrospective. Les livrables sont développés de manière incrémentale.
- **Daily Meetings** : Des réunions quotidiennes permettent à l'équipe de synchroniser ses efforts, d'identifier les blocages et d'ajuster les priorités si nécessaire.
- **Rétrospectives de sprint** : À la fin de chaque sprint, l'équipe analyse ce qui a bien fonctionné, ce qui peut être amélioré, et ajuste ses méthodes de travail en conséquence.
- **Livrables intermédiaires** : À chaque fin de sprint, des livrables exploitables (dashboards, notebooks, API, documentation...) sont produits et partagés avec les parties prenantes pour validation.
- **Jira** : L'ensemble du projet est géré dans Jira, un outil de gestion de projet Agile largement adopté dans l'industrie, qui permet :
  - ❖ De structurer le backlog de manière claire et collaborative,
  - ❖ D'assurer une traçabilité complète des décisions et des évolutions du projet,
  - ❖ De centraliser les informations, facilitant la communication entre les membres de l'équipe et les parties prenantes externes.Grâce à Jira, le suivi quotidien du projet est simplifié : chaque sprint est planifié avec des objectifs clairs, les tâches sont visuellement réparties dans un tableau Scrum (board), et l'équipe peut ajuster la charge de travail en temps réel.

### 4.3 Rôles, équipe et dynamique collective

Le projet a été porté par **une équipe réduite mais hautement polyvalente de trois membres**, chacun assumant plusieurs rôles clés à différents moments, en fonction des besoins. Cette organisation souple nous a permis de couvrir l'ensemble des dimensions du projet, de la conception à la mise en production, tout en favorisant une dynamique de **co-construction et de responsabilisation partagée**.



Les rôles assurés au cours du projet étaient les suivants :

- ❖ **Product Owner** : responsable de la vision produit, de la structuration du backlog et de la priorisation des user stories en lien avec la valeur métier. Ce rôle a permis de garder un cap fonctionnel clair et de s'assurer que chaque livrable répondait à un besoin réel.
- ❖ **Data Engineer** : en charge de la préparation, du nettoyage et de l'enrichissement des données. Cette fonction a assuré la qualité des jeux de données, condition essentielle à la performance des modèles.
- ❖ **Data Scientist** : responsable de la modélisation prédictive, de l'évaluation des performances (AUC, F1-score, etc.) et de l'interprétabilité des résultats. Ce rôle a été central dans la transformation des données en insights exploitables.
- ❖ **Architecte Cloud** : en charge de la conception de l'architecture technique pour le déploiement de la solution. Cela inclut le choix du fournisseur cloud, la gestion des coûts, la scalabilité et la sécurité de l'infrastructure.
- ❖ **Chef de projet** : rôle transversal de coordination, d'organisation des sprints, de suivi du planning et de fluidification des échanges. Ce rôle a permis de structurer le travail d'équipe en mode Agile (Scrum) et de garantir une progression continue.



Nom	Rôle	Contributions principales
<b>Badr Eddine Jeddane</b>	Chef de projet / Data analyst	Coordination globale du projet, gestion Agile, planification des sprints, traitement et analyse des données
<b>Sami Mechria</b>	Product Owner / Cloud & DevOps	Définition de la vision produit, gestion du backlog, conception de l'architecture cloud, estimation des coûts, déploiement .
<b>Francky Joël Fouejio</b>	Data scientist / ML engineer/ data engineer	Extraction et préparation des données, conception et entraînement des modèles, évaluation des performances, développement et intégration de l'API prédictive,

**L'agilité et la communication ont été les piliers de notre organisation.** Plutôt que d'appliquer une séparation rigide des responsabilités, nous avons fonctionné avec une forte complémentarité. Les décisions étaient prises en commun, les doutes partagés sans crainte, et les priorités ajustées collectivement à chaque sprint.

Notre force résidait dans notre capacité à apprendre ensemble, à itérer rapidement, à nous entraider sur les aspects techniques comme fonctionnels, et à progresser **comme une équipe unie**, tournée vers un objectif commun.

#### 4.4 Backlog fonctionnel

Dès les premiers jours du projet, nous avons structuré un backlog produit clair et hiérarchisé, en identifiant les grandes thématiques sous forme d'EPICs. Chacun de ces EPICs a ensuite été décomposé en user stories, rédigées systématiquement du point de vue de l'utilisateur final. Cette approche nous a permis de maintenir un ancrage fort sur les enjeux métier, en évitant de nous disperser dans des tâches purement techniques sans lien direct avec leur valeur d'usage.

Vous trouverez ci-dessous la liste détaillée des différents EPICs, accompagnés des user stories correspondantes.

##### EPIC 1 – Cadrage et exploration initiale

###### Description :Haut du formulaire

Définir les objectifs, rechercher des données, cadrer le projet.

###### User Story – Recherche de jeux de données

**En tant que** data scientist,  
**je veux** rechercher des jeux de données pertinents et publics,  
**afin de** disposer de données réalistes pour entraîner un modèle prédictif.

###### User Story – Rédaction du dossier de cadrage

**En tant que** chef de projet IA,  
**je veux** formaliser les enjeux, objectifs, contraintes techniques et juridiques,  
**afin de** poser une base claire et alignée pour le projet.

## **EPIC 2 – Préparation et analyse des données**

### **Description :**

Définir les objectifs, rechercher des données, cadrer le projet.

### **User Story – Nettoyage des données**

**En tant que** data analyst,  
**je veux** nettoyer et préparer les jeux de données sélectionnés,  
**afin de** garantir leur qualité et éviter les biais ou erreurs en entrée de modèle.

### **User Story – Analyse exploratoire (EDA)**

**En tant que** analyste,  
**je veux** analyser visuellement et statistiquement les données,  
**afin de** comprendre les facteurs associés aux annulations et guider la modélisation.

## **EPIC 3 – Modélisation IA**

### **Description :**

Concevoir, entraîner et évaluer le modèle prédictif.

### **User Story – Entraînement d'un modèle prédictif**

**En tant que** data scientist,  
**je veux** entraîner un modèle de classification binaire (annulé / non annulé),  
**afin de** prédire le risque d'annulation avec de bonnes performances.

### **User Story – Évaluation des performances du modèle**

**En tant que** développeur IA,  
**je veux** analyser les métriques (F1-score, AUC...),  
**afin de** mesurer la qualité des prédictions.

### **User Story – Interprétation des prédictions**

**En tant qu'**utilisateur métier,  
**je veux** comprendre pourquoi un rendez-vous est classé à risque,  
**afin de** pouvoir justifier et utiliser les résultats du modèle de façon responsable.

## **EPIC 4 – Création de l'API**

### **Description :**

Déployer le model IA sous forme d'API

### **User Story – Création et test de l' API prédictive**

**En tant que** développeur IA,  
**je veux** déployer le modèle via une API REST,  
**afin de** permettre aux équipes métier ou front-end d'y accéder facilement.

### **EPIC 5 – Suivi et pilotage de projet**

#### **Description :**

Assurer le bon déroulement du projet IA avec des pratiques agiles, une communication efficace, et un suivi rigoureux des tâches, du temps, et des risques.

### **User Story – Mise en place du backlog Jira et planification des sprints**

**En tant que** Product Owner,  
**je veux** structurer le backlog Jira avec les EPICs, user stories, et sprints,  
**afin de** suivre l'avancement du projet de façon agile et visible par tous.

### **User Story – Documentation projet et livrables intermédiaires**

**En tant que** chef de projet,  
**je veux** centraliser tous les livrables (EDA, modèles, API, dashboard...)  
**afin de** faciliter les échanges avec les parties prenantes et l'évaluation.

### **EPIC 6 – Architecture Cloud & Déploiement**

#### **Description :**

Concevoir, chiffrer et présenter une architecture cloud complète pour déployer la solution IA de prédiction en production, de façon sécurisée et scalable. Bas du formulaire

### **User Story – Conception de l'architecture cloud**

**En tant qu'**ingénieur cloud,  
**je veux** définir l'architecture (stockage, entraînement, API),  
**afin de** déployer une solution scalable et sécurisée.

### **User Story – Estimation du coût cloud**

**En tant que** responsable budget,  
**je veux** calculer les coûts de stockage, calcul, API,  
**afin d'**assurer la faisabilité financière

### **User Story – Choix du fournisseur cloud**

**En tant que** chef de projet,  
**je veux** sélectionner AWS, Azure ou GCP,  
**afin de** garantir compatibilité et maîtrise des coûts.

### User Story – Déploiement d’une API prédictive

**En tant que** développeur IA,  
**je veux** exposer le modèle via une API ,  
**afin de** l’intégrer dans les outils Doctolib

## 4.5 Organisation des sprints

Nous avons découpé le projet en quatre sprints de travail, réparties sur une durée de huit semaines. Chaque sprint, d’une durée de deux semaines, est dédié à un objectif précis. Cette organisation nous permet d’avancer de manière progressive et structurée, de produire des livrables concrets à intervalles réguliers, et d’ajuster notre approche agile en fonction des résultats obtenus et des besoins identifiés en cours de réalisation :

- Une revue de sprint : présentation des livrables réalisés à l’équipe et aux parties prenantes.
- Une rétrospective : analyse des points forts, difficultés rencontrées et axes d’amélioration pour le sprint suivant.

## 4.6 Planification des sprints

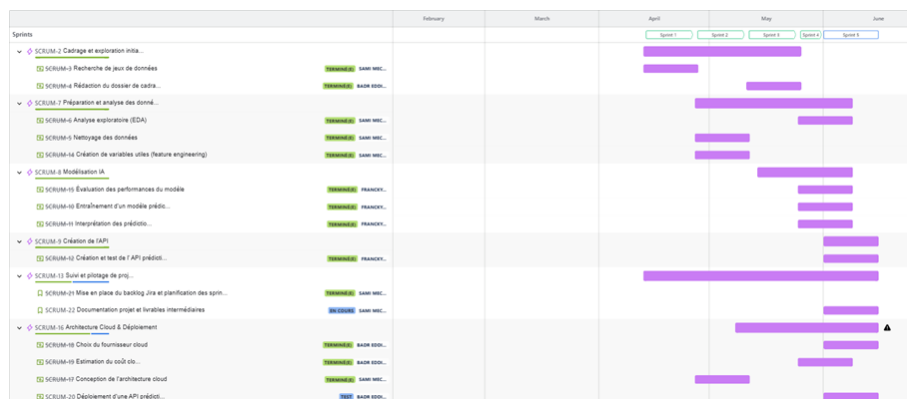
Sprint	Objectifs principaux	Livrables attendus
<b>Sprint 1 : Cadrage &amp; exploration initiale</b>	- Recherche et sélection de jeux de données publics ou synthétiques - Rédaction du dossier de cadrage - Création du backlog Jira et planification globale du projet	- Dossier de cadrage fonctionnel - Backlog structuré (EPICs et User Stories) - Données collectées et analysées en termes de pertinence
<b>Sprint 2 : Préparation et analyse des données</b>	- Nettoyage des jeux de données - Création des features (ex. : quartier, météo) - Analyse exploratoire (EDA)	- Jeux de données propres et enrichis - Notebook EDA - Rapport d’analyse synthétique
<b>Sprint 3 : Modélisation et évaluation</b>	- Entraînement de plusieurs modèles de classification (Random Forest, Regression...) - Évaluation des performances (AUC, F1-score...)	- Pipeline de modélisation IA - Rapport d’évaluation - Visualisations d’explicabilité

<b>Sprint 4 : Déploiement et finalisation</b>	- Déploiement du modèle IA via une API - Conception de l'architecture cloud (stockage, calcul, API) - Estimation budgétaire et choix du fournisseur cloud	- Maquette d'architecture cloud - Chiffrage cloud - API fonctionnelle - Rapport de présentation de la solution complète
---	---	--

## Sprints planning

Initialement, le projet était planifié sur **quatre sprints de deux semaines** chacun, avec des objectifs clairement définis pour chaque phase. Cependant, dans un souci d'adaptation dynamique à l'avancement réel et aux imprévus rencontrés, nous avons dû ajuster notre organisation en modifiant le contenu du quatrième sprint et en ajoutant un **cinquième sprint**. Cette flexibilité a permis de mieux finaliser le déploiement et d'assurer la qualité des livrables.

Vous trouverez ci-dessous le planning détaillé des différents sprints.

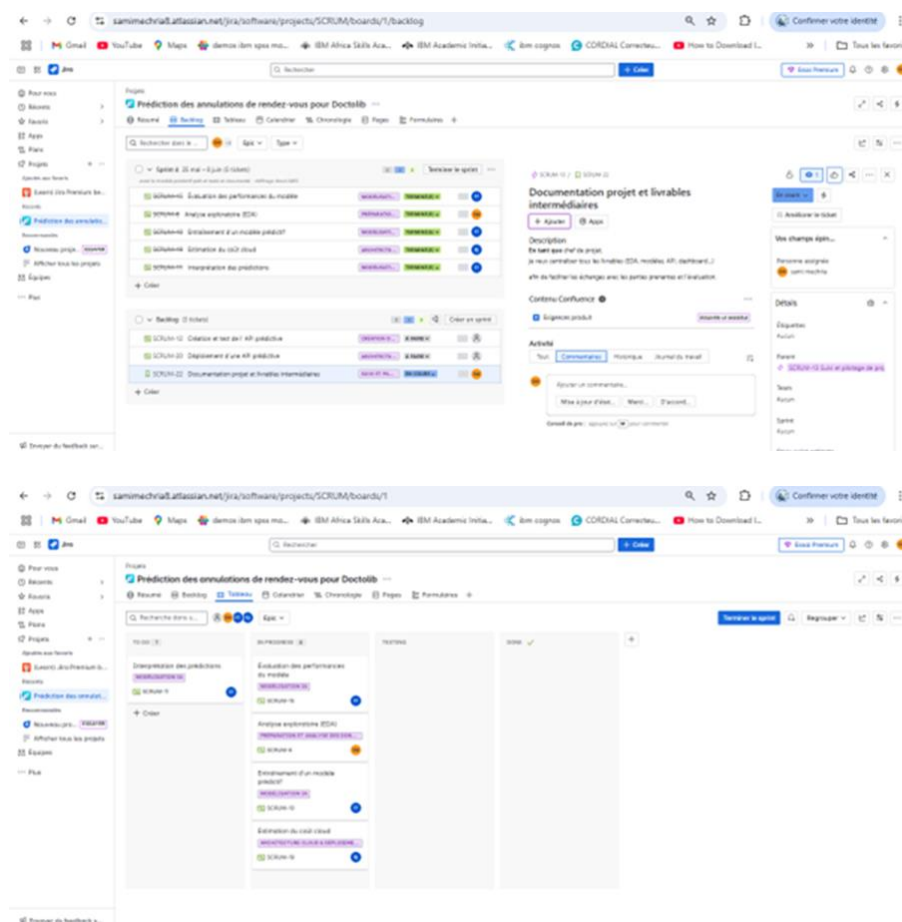


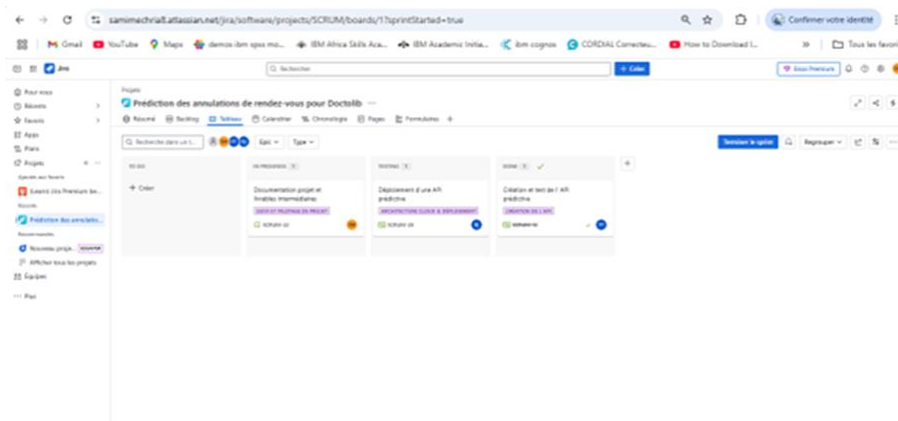
## Suivi de l'avancement

Tout au long du projet, l'outil Jira est utilisé pour :

- Gérer les tâches affectées à chaque sprint,
- Suivre l'avancement quotidien à l'aide du tableau Scrum,
- Et assurer une visibilité totale sur l'état du projet auprès des parties prenantes.

Nous présentons quelques exemples de tableaux de bord, incluant le suivi des sprints ainsi qu'une vue d'ensemble des EPICs. Ces illustrations nous permettent de donner une idée concrète des outils et méthodes que nous avons utilisées pour piloter le projet.





## 4.7 Conclusion

En somme, la gestion du projet s'est appuyée sur une organisation agile, flexible et collaborative, adaptée à la complexité technique et aux exigences métier du projet. La répartition claire des rôles, combinée à une communication fluide et à un suivi rigoureux via des outils comme Jira, a permis de maintenir une vision partagée, d'anticiper les risques, et d'assurer une progression cohérente et maîtrisée tout au long des différentes phases. Cette approche a favorisé l'implication de chacun, l'adaptabilité face aux imprévus, et la livraison régulière de livrables de qualité, en adéquation avec les objectifs définis. Ainsi, la chefferie de projet a joué un rôle central dans la réussite globale de cette initiative de prédiction d'annulation des rendez-vous pour Doctolib.

## 5. Collecte et préparation des données

### 5.1 Sources de données utilisées

La collecte de données constitue une étape cruciale dans la réussite de tout projet d'analyse de données. Dans le cadre de cette étude, les données proviennent principalement des plateformes de données ouvertes tels que Kaggle, Data.gouv et un processus de strapping du site Wikipedia via les codes en langage python. Ces sources contiennent des informations relatives à Hypertension, Diabète, Alcoolisme, Handicap, Jours\_entre\_la\_prise\_de\_renseignements\_et\_le\_rendez-vous, Zone\_hospitalière, Spécialité, Type\_d'établissement, Distance\_km, Type\_de\_soins, Précédemment traité, Âge et plus

encore. L'intégration de ces données dans un format de type CSV structuré a été effectuée via outils Microsoft Excel.

Nom	Type de données	Description
Appointment_Booking_Date	object	Date à laquelle le rendez-vous a été réservé.
Appointment_Date	object	Date réelle du rendez-vous.
Scholarship	int64	Indicateur (binaire) si la personne bénéficie d'une bourse.
Hypertension	int64	Indicateur (binaire) si la personne souffre d'hypertension.
Diabetes	int64	Indicateur (binaire) si la personne est diabétique.
Alcoholism	int64	Indicateur (binaire) si la personne souffre d'alcoolisme.
Disability	int64	Indicateur (binaire) si la personne est en situation de handicap.
Days_Between_Scheduling_and_Appointment	int64	Nombre de jours entre la planification du rendez-vous et le jour du rendez-vous.
Hospital_Area	object	Zone géographique de l'hôpital (par exemple, quartier ou région).
Specialty	object	Spécialité médicale pour laquelle le rendez-vous a été pris.
Facility_Type	object	Type d'établissement médical (par exemple, conventionné, privé, etc.).
Distance_km	float64	Distance en kilomètres entre le domicile du patient et l'établissement médical.
Type_of_Care	object	Type de soin (par exemple, suivi, consultation initiale, urgence).
Previously_Treated	int64	Indicateur (binaire) si la personne a été traitée auparavant dans cet établissement.
Age	int64	Âge de la personne.
Social_Status	object	Statut social du patient (par exemple, salarié, indépendant, etc.).
SMS_Received	int64	Indicateur (binaire) si la personne a reçu un SMS de rappel pour le rendez-vous.
Weather_Conditions	int64	Indicateur (binaire) si les conditions météorologiques



		sont défavorables (pluie, neige, etc.).
Appointment_Time	object	Heure prévue du rendez-vous.
Gender	object	Sexe de la personne (par exemple, Homme ou Femme).
Consultations_Last_12_Months	int64	Nombre de consultations dans les 12 derniers mois.
Waiting_Time_Minutes	int64	Temps d'attente en minutes avant de voir un professionnel de santé.
Hospital_Rating	float64	Note attribuée à l'hôpital (échelle de 1 à 5).
Average_Fee	float64	Frais moyens associés au rendez-vous.
Appointment_Status	int64	Statut du rendez-vous (0 : non annulé, 1 : annulé).

## 5.2 Nettoyage et traitement des données

Une fois les données collectées, un processus de nettoyage et de traitement a été appliqué afin de garantir leur qualité et leur cohérence. Ce processus inclut plusieurs étapes, telles que :

- **Traitement des valeurs manquantes** : Les valeurs manquantes dans certaines colonnes ont été identifiées et traitées par suppression. Pour certaines colonnes critiques, des stratégies d'imputation avancées ont été mises en place afin d'éviter la perte de données importantes.
- **Correction des incohérences et des erreurs** : Des anomalies telles que des valeurs aberrantes, des doublons ou des erreurs de format (par exemple, des dates mal formatées) ont été corrigées. Cela a impliqué l'utilisation de l'outil **pandas**, pour assurer l'intégrité des données.

## 1 - Suppression des doublons - Enlever les lignes doubles

```
#trouver les lignes dupliquées  
duplicates = data.duplicated()  
duplicates
```

```
0      False  
1      False  
2      False  
3      False  
4      False  
...  
110522  False  
110523  False  
110524  False  
110525  False  
110526  False  
Length: 110527, dtype: bool
```

```
#Compte des lignes dupliquées et non dupliquées  
duplicates.value_counts()
```

```
False    110527  
dtype: int64
```

*Capture 1*

- **Transformation des variables** : Certaines variables ont nécessité une transformation pour correspondre aux formats requis par les algorithmes de machine learning. Cela inclut encodage des variables catégorielles, normalisation des valeurs numériques, conversion de dates en variables temporelles.

```
# Convertir les colonnes en datetime
data['Appointment_Booking_Date'] = pd.to_datetime(data['Appointment_Booking_Date'])
data['Appointment_Date'] = pd.to_datetime(data['Appointment_Date'])

# Calculer la différence en jours entre ScheduledDay et AppointmentDay
data['Number_days'] = (data['Appointment_Date'] - data['Appointment_Booking_Date']).dt.days
data = data.drop(["Appointment_Booking_Date", "Appointment_Date"], axis= 1)

# Afficher les résultats
data
```

Capture 2

```
# Fonction de conversion HH:MM → secondes
def time_to_seconds(time_str):
    h, m = map(int, time_str.split(':'))
    return h * 3600 + m * 60

# Appliquer la fonction sur la colonne
data['Appointment_Time'] = data['Appointment_Time'].apply(time_to_seconds)
data['Appointment_Time'].head()
```

```
0    18060
1    65400
2    80040
3    16080
4    85260
Name: Appointment_Time, dtype: int64
```

Capture 3

```
encoder = LabelEncoder()
data['Gender'] = encoder.fit_transform(data['Gender'])
# Afficher le dataset avec la colonne encodée
data
```

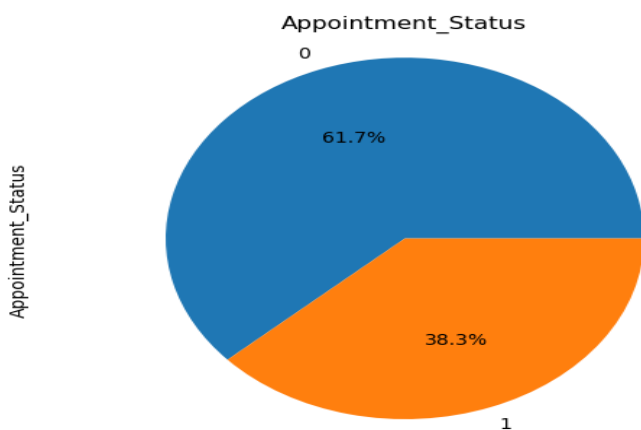
Capture 4

- **Équilibrage des données** : Si nécessaire, un **équilibrage des classes** a été effectué pour éviter le biais dans les modèles de prédiction, notamment dans le cas d'un déséquilibre entre les classes. exemples pour notre cas on n'a annulation de rendez-vous (No\_show) et non-annulation avec une répartition déséquilibré.

```
# Sélectionner les variables catégorielles
cat_vars = ['Appointment_Status']

# Générer un pie plot pour chaque variable catégorielle
for var in cat_vars:
    data[var].value_counts().plot(kind='pie', autopct='%1.1f%%')
    plt.title(var)
    plt.axis('equal')
    plt.show()
```

*Capture 5*



*Figure 1*

### 5.3 Analyse exploratoire des données

Avant de procéder à la modélisation, une **analyse exploratoire des données (EDA)** a été réalisée afin de mieux comprendre les caractéristiques du dataset et identifier des tendances, des relations et des patterns intéressants. Cette étape est essentielle pour :

- **Visualiser les distributions des variables** : Des graphiques tels que des histogrammes, des diagrammes de boîte et des diagrammes de dispersion ont été

utilisés pour visualiser la répartition des variables numériques et détecter d'éventuelles anomalies.

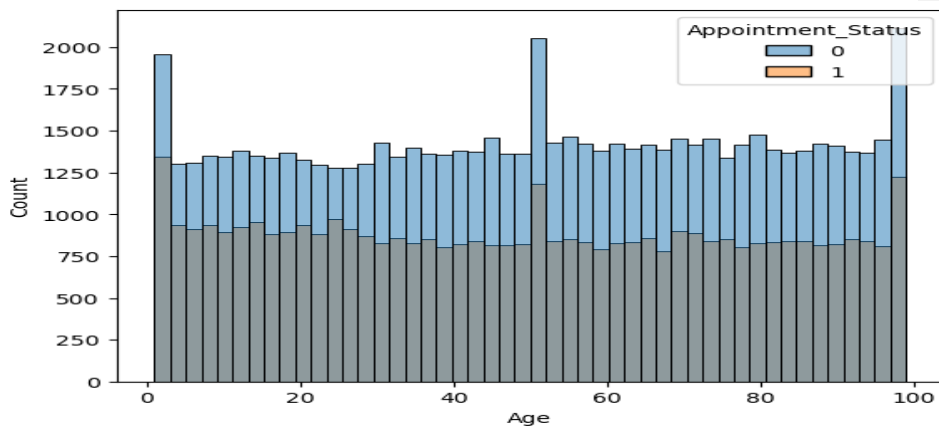


Figure 2

- **Explorer les relations entre les variables** : Des matrices de corrélation et des diagrammes de chaleur ont été utilisés pour observer les relations entre les différentes variables, par exemple, entre **Age** et **No\_show**, ou entre **Neighbourhood** et **DaysBetween**.

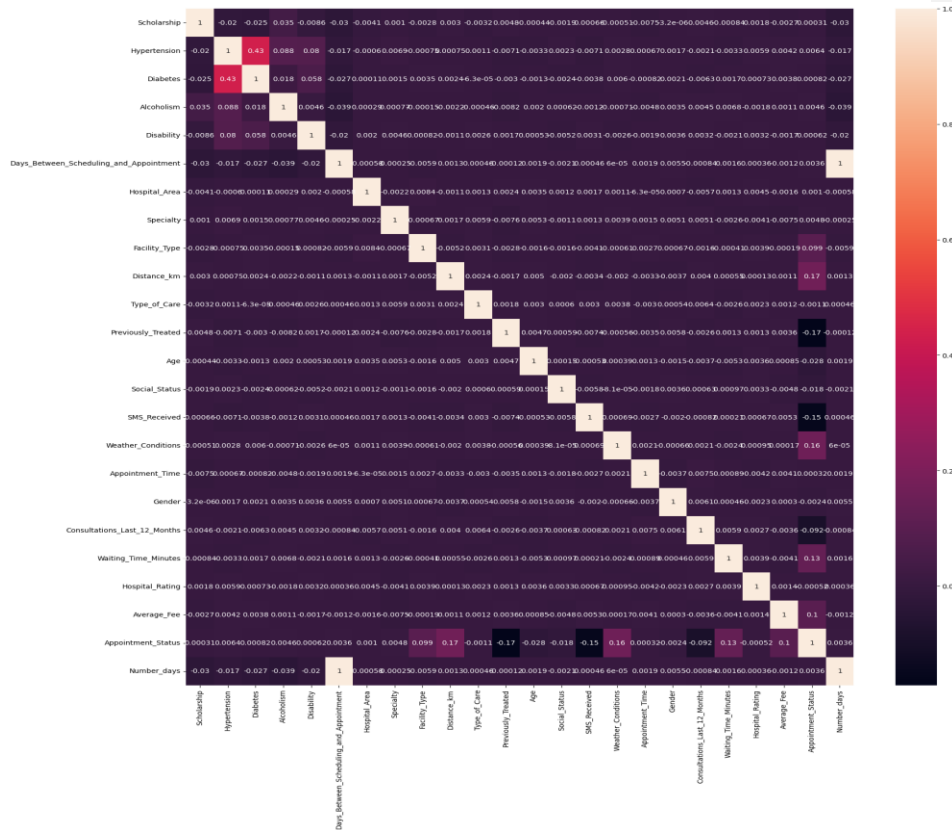


Figure 3

- **Analyse des variables catégorielles** : Pour les variables catégorielles comme **Gender**, **Neighbourhood**, et **SMS\_received**, des graphiques en barres ont été utilisés pour observer les fréquences de chaque catégorie. Cela a permis d'identifier des schémas dans la répartition des données et des différences de comportement en fonction des différentes catégories.
- **Identification des tendances temporelles** : L'analyse des dates de **ScheduledDay** et **AppointmentDay** a permis d'examiner les tendances saisonnières ou hebdomadaires concernant les annulations de rendez-vous, et d'éventuelles relations avec des périodes spécifiques.

#### 5.4 Visualisations interactives ( dashboard sur POWER BI) ----- laissez vite

## 6. Conception du modèle IA

### 6.1 Choix des modèles

Le choix des modèles est une étape essentielle dans la conception du système de prédiction. Pour ce projet, nous avons étudié plusieurs modèles de machine learning qui sont bien adaptés à la tâche de prédiction des annulations de rendez-vous :

- **Logistic Regression** : Un modèle linéaire de classification souvent utilisé pour des problèmes de classification binaire. Il est simple à interpréter et efficace pour des données bien structurées.
- **Random Forest Classifier** : Un modèle basé sur l'agrégation de plusieurs arbres de décision, capable de gérer des relations complexes entre les variables. Il est également robuste face aux données manquantes et aux variables peu pertinentes.
- **Gradient Boosting Classifier** : Un modèle d'ensachage qui combine plusieurs modèles faibles pour créer un modèle robuste et performant. Il est particulièrement utile lorsque les interactions complexes entre les caractéristiques sont présentes.
- **Extreme Gradient Boosting (XGBoost)** : Une version améliorée du gradient boosting qui est plus rapide et souvent plus performante sur des datasets avec beaucoup de données.
- **Naive Bayes** : Utilisé pour des données catégorielles et fondé sur l'hypothèse de l'indépendance des variables. Il est simple, mais peut être puissant si les hypothèses sont valides.

Ces modèles ont été sélectionnés pour leurs performances éprouvées sur des tâches de classification binaire, ainsi que pour leur capacité à s'adapter aux spécificités des données disponibles. Après études nous avons opté pour le modèle **Extreme Gradient Boosting (XGBoost)** du fait de sa rapidité et sa performance sur des datasets avec beaucoup de données.

### 6.2 Conception du modèle XGBoost

#### 6.2.a Oversampling ( Suréchantillonnage )

L'oversampling est une technique de rééchantillonnage utilisée pour gérer les ensembles de données déséquilibrés en augmentant le nombre d'échantillons de la classe minoritaire. Cela peut être fait en dupliquant les échantillons existants ou en générant de nouveaux échantillons synthétiques à partir des données existantes, par exemple en utilisant des méthodes telles que SMOTE (Synthetic Minority Over-sampling Technique) ou ADASYN (Adaptive Synthetic Sampling).

### Avantages

- Améliore la performance du modèle sur la classe minoritaire en augmentant la quantité d'informations disponibles pour l'apprentissage.
- Réduit le biais envers la classe majoritaire, ce qui peut améliorer la précision globale du modèle.
- Facilite la découverte de modèles significatifs dans les données en permettant aux algorithmes d'apprentissage d'explorer plus en profondeur la structure de la classe minoritaire.

### Inconvénients

- Peut entraîner un surapprentissage, car les échantillons dupliqués ou synthétiques peuvent augmenter la complexité du modèle sans apporter d'informations nouvelles.
- Augmente la taille de l'ensemble de données, ce qui peut augmenter les temps d'apprentissage et de prédiction.

```
# séparation de x et y
x = data.drop(['Appointment_Status'], axis = 1)
y = data['Appointment_Status']

# training
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.25, random_state=42)

# Initialiser l'objet RandomOverSampler
ros = RandomOverSampler(sampling_strategy=1, random_state=42)

# Appliquer l'oversampling sur les données d'apprentissage
X_train_oversampled, y_train_oversampled = ros.fit_resample(X_train, y_train)

# Créer un nouveau DataFrame avec les données oversampled
data_oversampled = pd.concat([X_train_oversampled, y_train_oversampled], axis=1)
```

Capture 6

```
# Fonction pour créer un pie chart avec les proportions et les nombres
def plot_pie_chart(y, title):
    labels = ['0', '1']
    sizes = y.value_counts().values
    colors = ['#66b3ff', '#ff9999']

    # Modifier le format des labels pour inclure les proportions et les nombres
    def autopct_format(pct, all_values):
        absolute = int(round(pct / 100 * sum(all_values)))
        return f"{pct:1f}% ({absolute})"

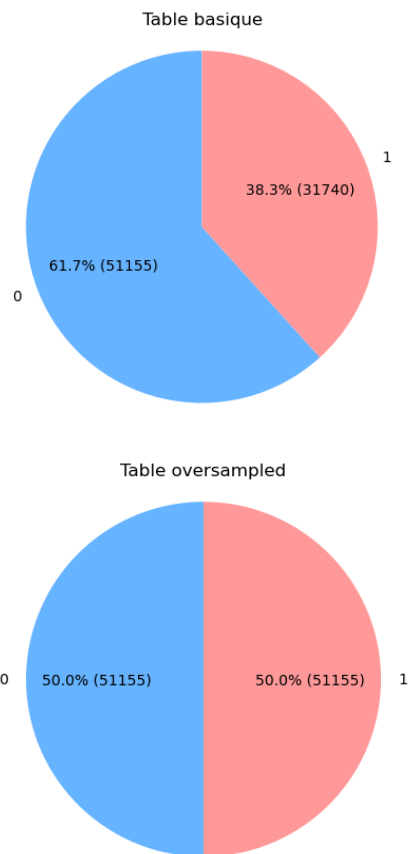
    fig, ax = plt.subplots()
    ax.pie(sizes, labels=labels, colors=colors, autopct=lambda pct: autopct_format(pct, sizes), startangle=90)
    ax.axis('equal') # Pour assurer que le diagramme est bien rond
    plt.title(title)
    plt.show()

# Créer un pie chart pour la table basique
plot_pie_chart(y_train, "Table basique")

# Créer un pie chart pour la table oversampled
plot_pie_chart(y_train_oversampled, "Table oversampled")
```

Capture 7





*Figure 4*

### 6.2.b Entraînement, optimisation et sauvegarde du modèle.

L'entraînement des modèles a été effectué en utilisant un ensemble de données d'entraînement **X\_train** et des étiquettes **y\_train**. Pour optimiser les performances, les hyperparamètres le modèle a été ajustés par **GridSearchCV**, ce qui permet de tester une grille d'hyperparamètres et de sélectionner la meilleure combinaison pour maximiser les performances du modèle.

**XGBoost** : Des hyperparamètres comme le taux d'apprentissage (`learning_rate`), le nombre d'arbres (`n_estimators`), et la profondeur des arbres (`max_depth`) ont été affinés pour améliorer l'accuracy.

```
#Entraîner et évaluer l'xgb sur la base oversampled
xgb_oversampled = xgb(n_estimators=100, learning_rate=0.1, max_depth=3, random_state=42)
xgb_oversampled = xgb_oversampled.fit(X_train_oversampled, y_train_oversampled)
y_pred_oversampled = xgb_oversampled.predict(X_test)
auc_oversampled = roc_auc_score(y_test, y_pred_oversampled)
y_pred_oversampled_train = xgb_oversampled.predict(X_train_oversampled)
auc_oversampled_train = roc_auc_score(y_train_oversampled, y_pred_oversampled_train)
```

Capture 8

```
import joblib

# Sauvegarder le modèle XGBoost entraîné sur les données oversampled
joblib.dump(xgb_oversampled, "xgb_oversampled_model.joblib")

['xgb_oversampled_model.joblib']
```

Capture 9

### 6.3 Métriques d'évaluation

Une fois les modèles entraînés, il est crucial d'évaluer sa performance sur un ensemble de test distinct. Pour évaluer les modèles de manière robuste, les métriques suivantes ont été utilisées :

- **Accuracy** : La proportion des prédictions correctes. C'est une métrique simple, mais elle peut être biaisée si les classes sont déséquilibrées.
- **Précision** (Precision) : La proportion de prédictions positives correctes parmi toutes les prédictions positives faites par le modèle.
- **Rappel** (Recall) : La proportion de prédictions positives correctes parmi toutes les instances réellement positives.
- **F1-Score** : La moyenne harmonique entre la précision et le rappel, qui donne un bon compromis entre les deux.
- **Courbe ROC et AUC** : La courbe ROC (Receiver Operating Characteristic) montre le compromis entre le taux de vrais positifs et le taux de faux positifs. L'AUC (Area Under the Curve) quantifie l'aire sous cette courbe, une mesure importante pour des modèles équilibrés.

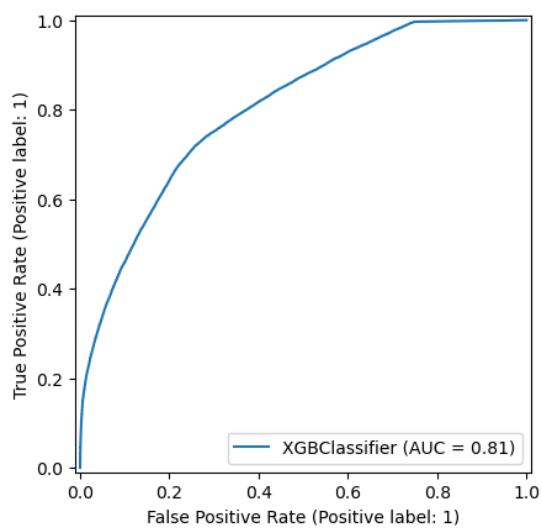


Figure 5

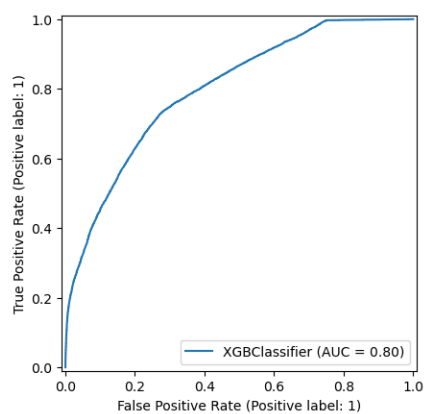


Figure 6

```

from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

# Prédiction sur le jeu de test
y_pred_test = xgb_oversampled.predict(X_test)

# Calcul des métriques
accuracy = accuracy_score(y_test, y_pred_test)
precision = precision_score(y_test, y_pred_test)
recall = recall_score(y_test, y_pred_test)
f1 = f1_score(y_test, y_pred_test)

print(f"Accuracy: {accuracy:.4f}")
print(f"Precision: {precision:.4f}")
print(f"Recall: {recall:.4f}")
print(f"F1-score: {f1:.4f}")

```

✓ 0.0s

Accuracy: 0.7305  
Precision: 0.6336  
Recall: 0.7029  
F1-score: 0.6664

*Capture 10*

Ces métriques ont permis de comparer les performances des différents modèles et de sélectionner celui offrant les meilleurs résultats pour la prédiction des annulations.

## 6.4 Interprétabilité du modèle (SHAP)

L'interprétabilité des modèles est cruciale, notamment pour des applications pratiques où il est important de comprendre les raisons derrière les prédictions. **SHAP (SHapley Additive exPlanations)** est une méthode d'explication des modèles basée sur les valeurs de Shapley, qui attribue une importance à chaque caractéristique pour une prédiction donnée.

- **SHAP values** : Les valeurs SHAP fournissent une explication quantifiée de l'impact de chaque caractéristique sur une prédiction particulière. Par exemple, dans le cas d'une annulation de rendez-vous, SHAP permet de déterminer si des facteurs comme l'âge, la distance ou la réception d'un SMS ont contribué positivement ou négativement à la prédiction.

L'usage de SHAP dans ce projet a permis de visualiser l'impact de chaque caractéristique sur les prédictions des modèles, ce qui est essentiel pour rendre les prédictions plus transparentes et compréhensibles pour les utilisateurs.

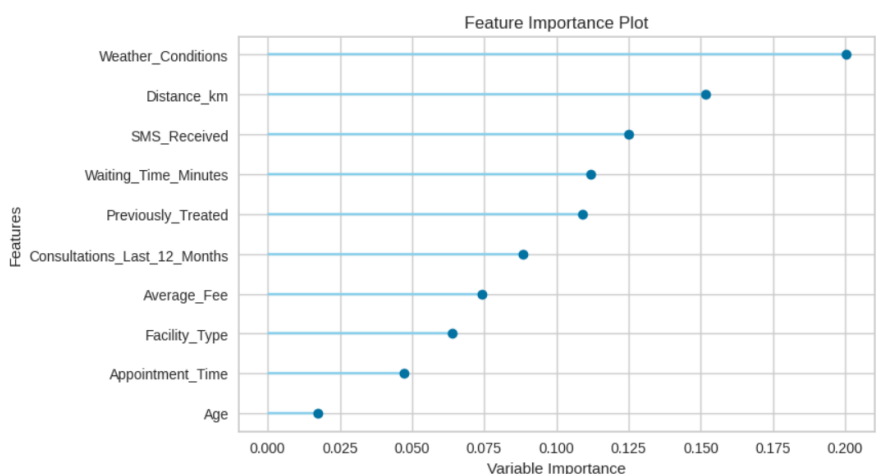


Figure 7

Le graphique présenté illustre l'importance des variables dans le modèle de prédiction des annulations de rendez-vous. On y observe que la variable **Weather\_Conditions** est la plus influente dans la prédiction, suivie par **Distance\_km**, **SMS\_Received** et **Waiting\_Time\_Minutes**. Cela signifie que des conditions météorologiques défavorables, une distance importante entre le patient et l'établissement, l'absence de SMS de rappel, ou encore un long temps d'attente contribuent fortement aux annulations. À l'inverse, des caractéristiques comme **Age** ou **Appointment\_Time** ont un impact plus faible dans la décision du modèle. Cette hiérarchisation permet de cibler les leviers d'action pour réduire les absences.

## 6.5 Analyse des résultats

Après l'entraînement et l'évaluation des modèles, une analyse approfondie des résultats a été réalisée. Voici quelques points clés de cette analyse :

- Analyse des résultats :** L'analyse combinée des courbes ROC et des métriques de classification montre que votre modèle XGBoost atteint une **performance globalement correcte mais perfectible**. L'AUC de 0.80–0.81 sur les deux courbes ROC indique une **bonne capacité discriminante** du modèle à différencier les rendez-vous annulés des présents. Du côté des métriques, l'**accuracy est de 73.05%**, ce qui est acceptable pour une tâche déséquilibrée. Le **recall de 70.29%** est un point positif, car il signifie que le modèle détecte bien les annulations (vrais positifs). Toutefois, la **précision reste modeste à 63.36%**, ce qui suggère qu'il y a encore trop de faux positifs. Le **F1-score de 66.64%** reflète un compromis intermédiaire entre rappel et précision. Globalement, votre modèle est plutôt bon pour **repérer les rendez-vous à risque d'annulation**, mais peut être encore amélioré en réduisant les fausses alertes.

- **Sur-apprentissage (Overfitting)** : Une attention particulière a été portée à éviter le sur-apprentissage en ajustant les hyperparamètres. Les modèles ont été évalués en utilisant des jeux de données de test et de validation pour s'assurer qu'ils généraient des résultats robustes et généralisables.

## 7. Application Streamlit et l'API / Base de données

### 7.1 Application Streamlit

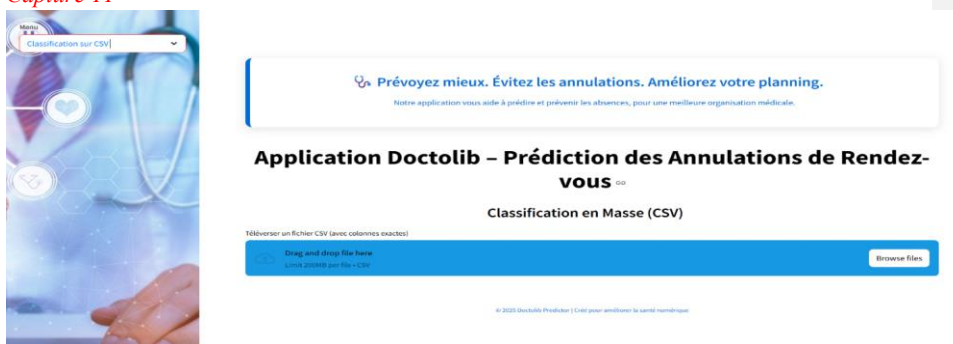
L'application Streamlit a été développée dans le but de fournir une interface utilisateur interactive pour prédire les annulations de rendez-vous médicaux. Elle s'appuie sur un modèle de machine learning (XGBoost) préalablement entraîné, et permet plusieurs modes d'utilisation :

- **Prédiction en temps réel** : L'utilisateur peut saisir manuellement les informations d'un patient pour obtenir une prédiction immédiate.
- **Classification sur fichier CSV** : Un fichier contenant plusieurs rendez-vous peut être téléversé et classifié automatiquement.
- **Système automatique (API)** : La plateforme peut interagir avec une API FastAPI pour récupérer dynamiquement les rendez-vous en attente et envoyer des notifications aux patients à risque.



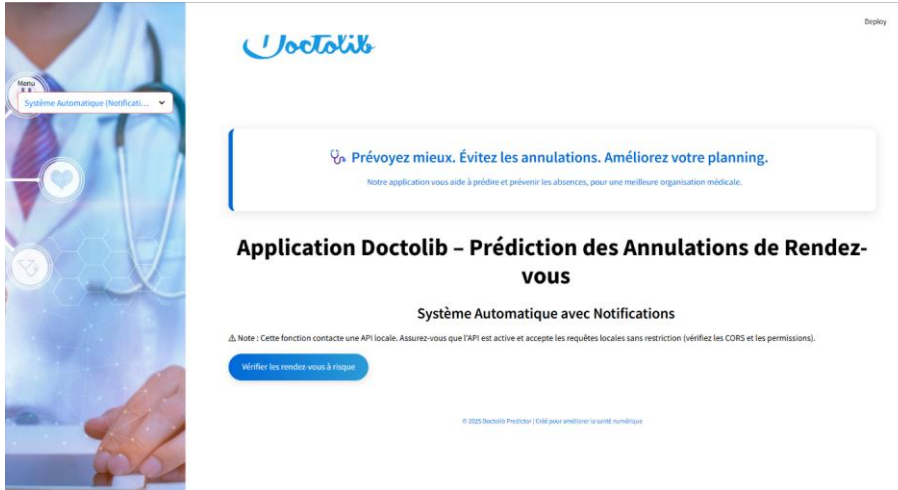
The screenshot shows the 'Prédiction en Temps Réel' section of the Doctolib application. It features a sidebar with a medical-themed image and a main content area. The main area has a header with the Doctolib logo and a mission statement: 'Prévoyez mieux. Évitez les annulations. Améliorez votre planning.' Below this, the title 'Application Doctolib – Prédiction des Annulations de Rendez-vous' is displayed. The section is titled 'Prédiction en Temps Réel'. There are three input fields: 'Date de prise de rendez-vous' (set to 2025/06/06), 'Date du rendez-vous' (set to 2025/06/06), and 'Schedulings (Dati=1, Nou=0)' (set to 0). A 'Predict' button is at the bottom.

Capture 11



The screenshot shows the 'Classification en Masse (CSV)' section of the Doctolib application. It features a sidebar with a medical-themed image and a main content area. The main area has a header with the Doctolib logo and a mission statement: 'Prévoyez mieux. Évitez les annulations. Améliorez votre planning.' Below this, the title 'Application Doctolib – Prédiction des Annulations de Rendez-vous' is displayed. The section is titled 'Classification en Masse (CSV)'. There is a text input field for 'Téléverser un Fichier CSV (avec colonnes exactes)'. Below it, there is a 'Drag and drop file here' area with a 'Browse Files' button. A 'Classify' button is at the bottom.

Capture 12



*Capture 13*

- La capture 11 représente le menu **Prédiction en temps réel**.
- La capture 12 représente le menu **Classification sur fichier CSV**
- La capture 13 représente le menu **Système automatique (API)**

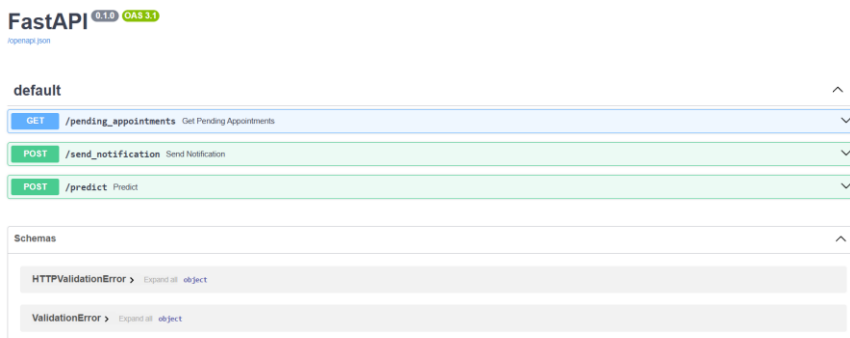
L'interface a été stylisée avec des éléments CSS intégrés pour correspondre à l'identité graphique de Doctolib : palette de bleus, logo intégré, éléments visuels médicaux. Des animations, styles d'encadrement et options de téléchargement au format .csv ou .xlsx ont également été intégrés pour renforcer l'expérience utilisateur.

## 7.2 Conception de l'API et de la base de données

L'API a été développée avec **FastAPI**, un framework Python moderne et performant. Elle fournit trois points d'entrée principaux :

- **/pending\_appointments** : Récupère les rendez-vous en attente depuis la base de données.
- **/predict** : Permet de prédire l'annulation d'un rendez-vous donné.
- **/send\_notification** : Enregistre dans la base un message d'alerte pour un rendez-vous identifié comme risqué.





### Capture 14

Cette capture représente l'interface où nous testons notre API.

La **base de données** est hébergée via **phpMyAdmin** (MySQL) pour le test en local et contient une table principale **appointments** avec les champs correspondant aux caractéristiques du modèle, ainsi qu'un champ **status** et **notification\_message**. La table suit la structure suivante :

```
CREATE TABLE appointments ( id INT AUTO_INCREMENT PRIMARY KEY,
scholarship TINYINT, hypertension TINYINT, diabetes TINYINT, alcoholism TINYINT,
disability TINYINT, days_between INT, hospital_area VARCHAR(50), specialty
VARCHAR(50), facility_type VARCHAR(50), distance_km FLOAT, type_of_care
VARCHAR(50), previously_treated TINYINT, age INT, social_status VARCHAR(50),
sms_received TINYINT, weather_conditions TINYINT, appointment_time INT, gender
VARCHAR(10), consultations_last_12_months INT, waiting_time_minutes INT,
hospital_rating FLOAT, average_fee FLOAT, number_days INT, status VARCHAR(50)
DEFAULT 'pending', notification_message TEXT );
```

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
<input type="checkbox"/> 1	id	int			Non	Aucun(e)		AUTO_INCREMENT	Modifier  Supprimer Plus
<input type="checkbox"/> 2	scholarship	tinyint(1)			Oui	NULL			Modifier  Supprimer Plus
<input type="checkbox"/> 3	hypertension	tinyint(1)			Oui	NULL			Modifier  Supprimer Plus
<input type="checkbox"/> 4	diabetes	tinyint(1)			Oui	NULL			Modifier  Supprimer Plus
<input type="checkbox"/> 5	alcoholism	tinyint(1)			Oui	NULL			Modifier  Supprimer Plus
<input type="checkbox"/> 6	disability	tinyint(1)			Oui	NULL			Modifier  Supprimer Plus
<input type="checkbox"/> 7	days_between	int			Oui	NULL			Modifier  Supprimer Plus
<input type="checkbox"/> 8	hospital_area	varchar(255)	utf8mb4_unicode_ci		Oui	NULL			Modifier  Supprimer Plus
<input type="checkbox"/> 9	specialty	varchar(255)	utf8mb4_unicode_ci		Oui	NULL			Modifier  Supprimer Plus
<input type="checkbox"/> 10	facility_type	varchar(255)	utf8mb4_unicode_ci		Oui	NULL			Modifier  Supprimer Plus
<input type="checkbox"/> 11	distance_km	float			Oui	NULL			Modifier  Supprimer Plus
<input type="checkbox"/> 12	type_of_care	varchar(255)	utf8mb4_unicode_ci		Oui	NULL			Modifier  Supprimer Plus
<input type="checkbox"/> 13	previously_treated	tinyint(1)			Oui	NULL			Modifier  Supprimer Plus
<input type="checkbox"/> 14	age	int			Oui	NULL			Modifier  Supprimer Plus
<input type="checkbox"/> 15	social_status	varchar(255)	utf8mb4_unicode_ci		Oui	NULL			Modifier  Supprimer Plus
<input type="checkbox"/> 16	sms_received	tinyint(1)			Oui	NULL			Modifier  Supprimer Plus
<input type="checkbox"/> 17	weather_conditions	varchar(255)	utf8mb4_unicode_ci		Oui	NULL			Modifier  Supprimer Plus
<input type="checkbox"/> 18	appointment_time	int			Oui	NULL			Modifier  Supprimer Plus
<input type="checkbox"/> 19	gender	varchar(10)	utf8mb4_unicode_ci		Oui	NULL			Modifier  Supprimer Plus
<input type="checkbox"/> 20	consultations_last_12_months	int			Oui	NULL			Modifier  Supprimer Plus
<input type="checkbox"/> 21	waiting_time_minutes	int			Oui	NULL			Modifier  Supprimer Plus
<input type="checkbox"/> 22	hospital_rating	float			Oui	NULL			Modifier  Supprimer Plus
<input type="checkbox"/> 23	average_fee	float			Oui	NULL			Modifier  Supprimer Plus
<input type="checkbox"/> 24	number_days	int			Oui	NULL			Modifier  Supprimer Plus
<input type="checkbox"/> 25	status	varchar(20)	utf8mb4_unicode_ci		Oui	pending			Modifier  Supprimer Plus
<input type="checkbox"/> 26	notification_message	varchar(255)	utf8mb4_unicode_ci		Oui	NULL			Modifier  Supprimer Plus

Capture 15

## 8. Architecture Cloud

### Introduction

Dans le cadre de l'automatisation et de l'optimisation des services numériques liés à la gestion des rendez-vous médicaux sur la plateforme Doctolib, le cloud computing s'est imposé comme une solution incontournable. La mise en place d'un système de prédiction en temps réel, capable d'apprendre en continu à partir de volumes croissants de données, combinée à la nécessité de proposer une application dédiée à la gestion des annulations de rendez-vous, requiert une infrastructure souple, résiliente et évolutive.

La partie métier du projet repose sur une API capable de prédire, dès la prise de rendez-vous, la probabilité qu'un patient ne se présente pas ou annule. Afin d'exploiter pleinement le potentiel des technologies de Machine Learning tout en garantissant la haute disponibilité du service, l'architecture cloud a été conçue de manière modulaire, scalable et hautement disponible.

L'objectif principal est de garantir :

- o Une exposition sécurisée de l'API à la plateforme Doctolib,
- o Une infrastructure capable de gérer la charge de manière dynamique,
- o Une application dédiée à la gestion et l'analyse des annulations des rendez-vous
- o Un accès transparent et encadré pour les équipes Data Science pour entraîner, réentraîner et améliorer les modèles prédictifs,
- o Et un Plan de Continuité d'Activité (PCA) assuré par un déploiement en multi-zones de disponibilité (AZ) dans un tenant cloud dédié.

Dans les sections suivantes, nous présentons en détail les architectures techniques proposées sur AWS et Azure, leurs composants, leur chiffrage estimatif, ainsi que les raisons motivant le choix du fournisseur cloud retenu.

## 8.1 Architecture générale

L'architecture technique du projet repose sur une approche cloud-native, privilégiant les services managés pour réduire la charge d'exploitation, améliorer la résilience du système et permettre un passage à l'échelle automatique. Ce choix s'inscrit dans une logique de modernisation des systèmes d'information de santé et d'optimisation des coûts liés à l'infrastructure.

La solution proposée s'articule autour de plusieurs couches fonctionnelles distinctes, déployées dans un tenant cloud multi-AZ afin d'assurer un Plan de Continuité d'Activité (PCA) robuste :

### o Couche d'API (Interface d'exposition)

Cette couche est responsable de la réception et de la gestion des requêtes HTTP émanant de l'application Doctolib. Elle comprend des endpoints POST pour la prédiction à la réservation d'un rendez-vous, et PUT pour la mise à jour du statut réel du rendez-vous (honoré ou annulé).

### o Fonctions serverless

Deux fonctions cloud distinctes sont implémentées pour exécuter la logique métier. La première déclenche un appel au modèle prédictif lors d'une réservation (POST), tandis que la seconde enregistre le retour réel du rendez-vous (PUT) et met à jour le jeu de données d'entraînement. Le choix du serverless permet un dimensionnement automatique en fonction de la charge, sans gestion manuelle de serveurs.

### o Service de machine learning managé

Le modèle prédictif est entraîné, déployé et exposé via un service managé de machine learning (AWS SageMaker ou Azure Machine Learning). Ce service permet une gestion fine du cycle de vie du modèle (training, déploiement, versioning) tout en assurant une haute disponibilité.

### o Système de stockage sécurisé et hybride

Deux types de stockage sont utilisés :

- ❖ Une base de données relationnelle (AWS RDS / Azure SQL) pour stocker les données structurées relatives aux rendez-vous et aux patients.

- ❖ Un stockage d'objets (S3 / Azure Blob Storage) pour les données brutes et les fichiers de datasets utilisés dans l'entraînement des modèles.

#### o Espace collaboratif Data Science

Les data scientists ont accès à un environnement isolé et sécurisé (SageMaker Notebooks ou Azure ML Studio) pour réaliser les travaux de modélisation, d'entraînement et d'évaluation. Cet environnement est connecté au stockage et au pipeline d'automatisation pour permettre l'intégration continue de nouvelles versions de modèles.

#### o Application de gestion des annulations des rendez-vous

L'application dédiée à la gestion des annulations offre aux utilisateurs (patients, praticiens, ou personnel administratif) une interface centralisée, accessible via le web, pour consulter, planifier ou mettre à jour les rendez-vous. Connectée au moteur de prédiction et à la base de données, elle permet un suivi en temps réel du statut des rendez-vous, et facilite la prise de décision grâce aux prédictions de probabilité d'annulation. Son intégration au sein de l'écosystème cloud garantit une disponibilité élevée, une sécurité renforcée des données, et une interaction fluide avec les services d'analyse et de stockage.

Dans les sections suivantes, nous détaillerons l'implémentation de cette architecture sur deux fournisseurs cloud majeurs : AWS et Microsoft Azure, en analysant leurs composants, leurs mécanismes de sécurité et de haute disponibilité, ainsi que les implications budgétaires.

## 8.2 Architecture Cloud AWS

### 8.2.a Présentation

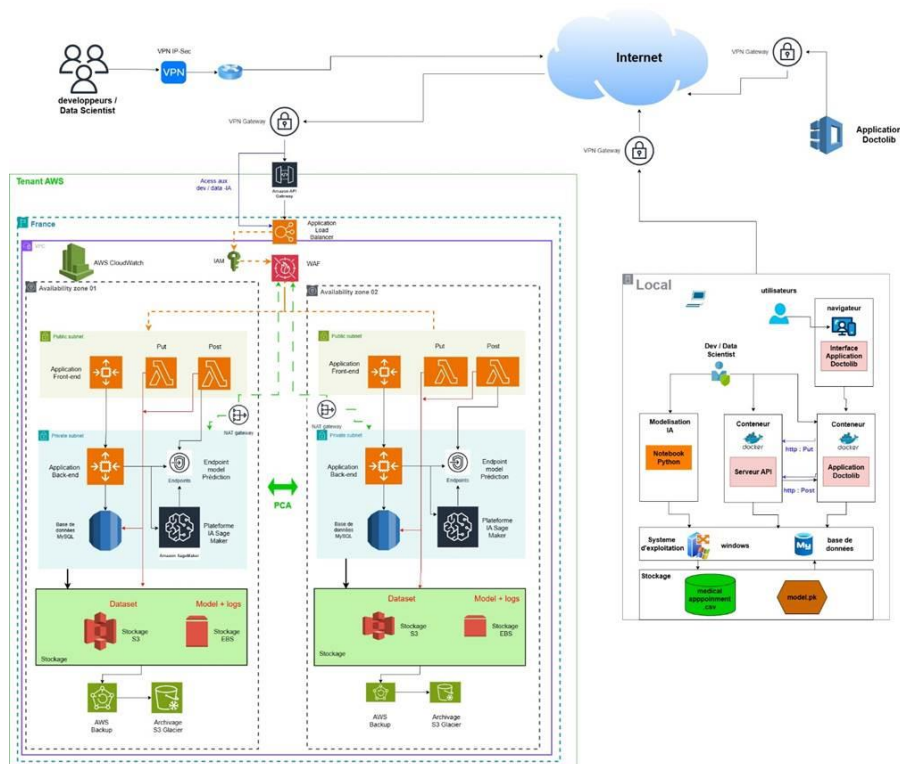
Pour concevoir notre architecture cloud autour du projet Doctolib, nous avons choisi d'exploiter les services proposés par Amazon Web Services (AWS). Cette plateforme nous permet de mettre en place une infrastructure moderne, sécurisée et capable de s'adapter aux évolutions du projet.

Grâce à AWS Lambda, nous gérons nos API de manière flexible, sans avoir à administrer de serveurs. Nous déployons notre application sur des machines virtuelles EC2, afin de contrôler

pleinement l'environnement d'exécution du front-end et du back-end. Côté machine learning, SageMaker nous offre un cadre complet pour entraîner, tester et déployer nos modèles de prédiction d'annulation.

Nous nous appuyons également sur Amazon S3 pour le stockage des données, et Amazon RDS (MySQL) pour la gestion de notre base relationnelle. L'ensemble de ces services nous permet de répondre efficacement aux enjeux de performance, de scalabilité et de conformité réglementaire, notamment en matière de protection des données personnelles (RGPD).

### 8.2.b Schéma technique



### 8.2.c Description des composants

#### a. VPN Site-to-Site

Deux tunnels VPN sont mis en place afin d'assurer la sécurité des échanges :

- VPN Doctolib - AWS : Ce tunnel garantit un canal de communication chiffré entre l'application Doctolib (côté client) et les services hébergés dans AWS. Cela permet de

s'affranchir de l'exposition des données sur Internet, renforçant la sécurité et la conformité RGPD.

- VPN Data Science - AWS : Ce tunnel permet aux data scientists d'accéder à leur environnement d'analyse et d'expérimentation (SageMaker Studio, fichiers S3) via un accès réseau privé depuis leur poste de travail, en toute sécurité.

### **b. Amazon API Gateway**

API Gateway sert de point d'entrée pour les appels HTTP provenant de l'application Doctolib. Il permet de créer, surveiller et sécuriser les endpoints REST suivants :

- `POST /predict` : Reçoit les données d'un nouveau rendez-vous et déclenche une prédiction d'annulation.
- `PUT /update` : Met à jour les données existantes avec le statut réel du rendez-vous (annulé ou honoré).

L'API Gateway joue également un rôle dans la gestion des quotas, de la journalisation, et de l'authentification (via IAM ou tokens JWT).

### **c. AWS Lambda**

Deux fonctions sans serveur assurent la logique applicative :

- Lambda POST : Cette fonction traite les requêtes `POST /predict`. Elle effectue les opérations suivantes :
  - Valide les données reçues.
  - Enregistre les données dans un fichier JSON ou CSV dans un bucket S3, avec un nom unique ou horodaté et mis à jour la base de données mysql
  - Interroge le modèle hébergé sur SageMaker pour obtenir une prédiction d'annulation.
  - Retourne la prédiction à l'application cliente.
- Lambda PUT : Cette fonction gère les mises à jour provenant de `PUT /update`. Elle :
  - Récupère le fichier correspondant au rendez-vous depuis S3 et ce correspondant à la base de données MySQL.
  - Met à jour le statut du rendez-vous avec les informations réelles (annulé, honoré).
  - Sauvegarde de nouveau le fichier dans S3 et la base de données, créant ainsi une trace complète du cycle de vie du rendez-vous.

### **d. Amazon EC2 (VM Front-End et Back-End)**

Pour bien déployé l'application d'aide à la gestion d'annulation des rendez-vous nous avons opté à mettre en place deux machines virtuelles EC2:

- **VM Front-End** : Héberge l'interface utilisateur de l'application, accessible aux utilisateurs via HTTPS et elle interagit avec le back-end via des appels REST.
- **VM Back-End** : Contient la logique métier de l'application, Elle agit comme intermédiaire entre le front-end, l'API du prédiction et la base de données RDS.

Les instances EC2 sont sécurisées par des groupes de sécurité restreints et supervisées via CloudWatch.

#### e. Amazon RDS (Base de données MySQL)

Une base de données relationnelle MySQL est déployée via **Amazon RDS** pour stocker les informations structurées et critiques, telles que :

- Les utilisateurs, professionnels de santé, rendez-vous, et statuts.
- Les métadonnées liées aux prédictions et résultats.

Le choix de RDS offre la haute disponibilité, les sauvegardes automatiques, la réplication et la sécurité au niveau réseau (VPC).

#### f. Amazon SageMaker

SageMaker est au cœur du dispositif de machine learning. Il est utilisé pour :

- Le déploiement du modèle sous forme d'un endpoint HTTPS que Lambda peut interroger pour obtenir des prédictions en temps réel.
- L'entraînement automatique via des jobs planifiés qui relisent les données stockées dans S3 pour mettre à jour le modèle avec les derniers rendez-vous et statuts collectés.
- L'environnement SageMaker Studio, qui offre aux data scientists un espace collaboratif pour explorer les données, tester de nouveaux modèles, visualiser les résultats, et monitorer les performances.

#### e. Amazon S3

S3 est utilisé comme stockage central unique, permettant de gérer :

- Les fichiers entrants contenant les informations des nouveaux rendez-vous.
- Les mises à jour effectuées après la réalisation ou annulation.
- Les datasets d'entraînement utilisés périodiquement par SageMaker.



- Les résultats de prédiction retournés par le modèle, ainsi que les logs d'exécution produits par Lambda.

Grâce à ses fonctionnalités de versioning et de cycle de vie, S3 facilite aussi la gestion de l'historique des données et l'archivage.

## f. IAM et CloudWatch

- **IAM (Identity and Access Management)** : Définit les rôles et politiques d'accès pour contrôler les interactions entre les composants (ex. : Lambda ne peut écrire dans S3 que sur un répertoire donné, SageMaker ne lit que certains fichiers).
- **CloudWatch** : Fournit des métriques, logs et tableaux de bord pour surveiller :
  - Le bon déroulement des fonctions Lambda.
  - Le trafic des endpoints API.
  - L'état des jobs d'entraînement SageMaker.
  - Les alarmes déclenchées en cas d'anomalie (latence, erreurs 5xx, échecs de job, etc.).

### 8.2.d Chiffrage estimatif

Selon le site officiel de Doctolib, en 2024, la plateforme comptait 50 millions de patients utilisateurs en France. Chaque mois, ce sont 15 millions de rendez-vous pris en ligne, dont près de la moitié sont programmés en dehors des heures d'ouverture des cabinets médicaux, illustrant une flexibilité et une accessibilité accrues pour les patients. Cela se traduit par une moyenne quotidienne d'environ 500 000 rendez-vous.

L'estimation du coût mensuel de l'infrastructure AWS reposera donc sur cette volumétrie de 15 millions de rendez-vous par mois, soit 15 millions de requêtes POST et 15 millions de requêtes PUT, représentant une charge de production de 30 millions d'appels API mensuels.

### Résumé du Chiffrage de l'Architecture Cloud AWS

*Total mensuel estimé : 5 442,65 \$ / 4 764,07 €*

*Total annuel estimé : 65 311,8 \$ / 57 168,84 €*

*Taux de conversion utilisé : 1 USD = 0,87535 EUR (au 4 juin 2025)*

Services	Détails	Quantité	Tarification	Prix unitaire	Prix total mensuel	Prix total mensuel

				mens uel	uel USD	EUR O
Amazon API Gateway	Requêtes HTTP, REST, WebSocket, mémoire cache 6,1 Go	1	à l'usage	252,4 7 USD	252,4 7 USD	221.0 1 €
AWS Lambda	1M requêtes/jour, 1000 simultanéités provisionnées, 512 Mo de stockage éphémère	4	à l'usage	75,26 USD	301.0 4 USD	263.5 6 €
Amazon SageMaker	Studio (5 users), Data Wrangler, Entraînement (ml.g6.4xlarge) , stockage SSD	2	à l'usage	175,3 9 USD	350.7 8 USD	307.1 3 €
Amazon EC2	m5a.large   Family: m5a   2vCPU   8 GiB Mémoire	2	à l'usage	73,73 USD	147,4 6 USD	128.9 9 €
Amazon EC2	m6gd.xlarge   Family: m6gd   4vCPU   16 GiB Mémoire	2	à l'usage	154,1 8 USD	308,3 6 USD	269.9 1 €
Amazon RDS for MySQL	db.m4.2xlarge vCPU: 8 Memory: 32 GiB stockage SSD 500 Go	2	à l'usage	1 517, 81 USD	3035, 62 USD	2655. 59 €
Amazon S3	15 Go, 45M requêtes, 2 To de données traitées via Select	2	à l'usage	168,7 8 USD	337.5 6 USD	295,4 7 €

AWS Backup	1 To sauvegardé, variation quotidienne et annuelle estimée	2	à l'usage	59,14 USD	118,28 USD	103,51 €
Amazon CloudWatch	1 tableau de bord, alarmes, 10M logs/mois, 2 fonctions Lambda	1	à l'usage	15,06 USD	15,06 USD	13,18 €
Amazon Elastic Block Store	1 To, 750 h/mois, 2 snapshots/jour	2	à l'usage	177,81 USD	355,62 USD	311,44 €
Amazon VPC	Connexions VPN Site-to-Site, sous-réseaux associés	1	à l'usage	198,80 USD	198,80 USD	174,00 €
AWS Web Application Firewall	2 ACL Web, 2 règles/ACL, 2 groupes/règles, 2 règles/groupe	1	à l'usage	26,60 USD	26,60 USD	23,28 €

## 8.3 Architecture Cloud Azure

### 8.3.a Présentation

En parallèle de notre approche avec AWS, nous avons également exploré les solutions proposées par Microsoft Azure. Cette plateforme nous offre une alternative robuste, avec une

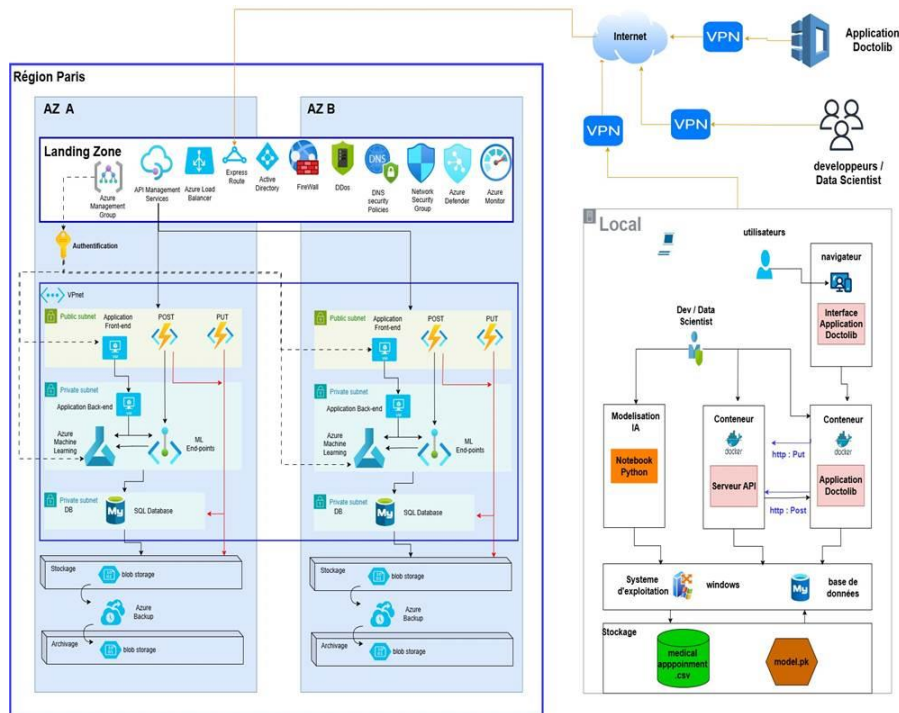
gamme complète de services pour construire une infrastructure cloud sécurisée et évolutive, adaptée aux besoins de notre application.

Avec Azure Functions, nous orchestrans les appels API de manière serverless, tout en conservant une bonne maîtrise des flux de données. L'hébergement du front-end et du back-end est assuré via des machines virtuelles Azure, ce qui nous garantit souplesse et continuité dans le déploiement.

Pour la partie intelligence artificielle, Azure Machine Learning nous fournit un environnement de travail puissant et collaboratif, idéal pour entraîner, tester et affiner nos modèles prédictifs. Les données sont stockées de manière fiable dans Azure Blob Storage, et la base de données relationnelle est gérée via Azure Database for MySQL, assurant sécurité et conformité avec les réglementations européennes.

Cette alternative nous permet de comparer les offres et de choisir la solution la plus adaptée selon nos contraintes techniques, financières ou organisationnelles.

### 8.3.b Schéma technique



### 8.3.c Description des composants

#### a. VPN Site-to-Site

Deux tunnels VPN sont mis en place afin d'assurer la sécurité des échanges :

- **VPN Doctolib - Azure** : Ce tunnel garantit un canal de communication chiffré entre l'application Doctolib (côté client) et les services hébergés dans Azure. Cela permet d'éviter l'exposition des données sur Internet, renforçant la sécurité et la conformité RGPD.
- **VPN Data Science - Azure** : Ce tunnel permet aux data scientists d'accéder à leur environnement d'analyse et d'expérimentation (Azure Machine Learning Studio, fichiers Storage Account) via un accès réseau privé sécurisé depuis leur poste de travail.

#### b. Azure API Management

Azure API Management sert de point d'entrée pour les appels HTTP provenant de l'application Doctolib. Il permet de créer, surveiller et sécuriser les endpoints REST suivants :

- **POST /predict** : Reçoit les données d'un nouveau rendez-vous et déclenche une prédiction d'annulation.
- **PUT /update** : Met à jour les données existantes avec le statut réel du rendez-vous (annulé ou honoré).  
API Management joue également un rôle dans la gestion des quotas, la journalisation, et l'authentification (via Azure AD, tokens JWT, ou autres mécanismes).

### c. Azure Functions

Deux fonctions serverless assurent la logique applicative :

- **Function POST** : Cette fonction traite les requêtes POST /predict. Elle effectue les opérations suivantes :
  - o Valide les données reçues.
  - o Enregistre les données dans un fichier JSON ou CSV dans un Storage Account, avec un nom unique ou horodaté, et met à jour la base de données Azure Database for MySQL.
  - o Interroge le modèle hébergé sur Azure Machine Learning pour obtenir une prédiction d'annulation.
  - o Retourne la prédiction à l'application cliente.
- **Function PUT** : Cette fonction gère les mises à jour provenant de PUT /update. Elle :
  - o Récupère le fichier correspondant au rendez-vous depuis le Storage Account et les données associées dans la base Azure Database for MySQL.
  - o Met à jour le statut du rendez-vous avec les informations réelles (annulé, honoré).
  - o Sauvegarde de nouveau le fichier dans le Storage Account et la base de données, créant ainsi une trace complète du cycle de vie du rendez-vous.

#### d. Machines virtuelles Azure (VM Front-End et Back-End)

Pour le déploiement de l'application d'aide à la gestion des annulations de rendez-vous, deux machines virtuelles Azure sont mises en place :

- **VM Front-End** : Héberge l'interface utilisateur de l'application, accessible aux utilisateurs via HTTPS, et interagit avec le back-end via des appels REST.
- **VM Back-End** : Contient la logique métier de l'application, servant d'intermédiaire entre le front-end, l'API de prédiction et la base de données MySQL.  
Les VM sont sécurisées par des groupes de sécurité Azure et surveillées via Azure Monitor.

#### e. Azure Database for MySQL

Une base de données relationnelle MySQL est déployée via Azure Database for MySQL pour stocker les informations structurées et critiques telles que :

- Les utilisateurs, professionnels de santé, rendez-vous, et statuts.
- Les métadonnées liées aux prédictions et résultats.

Ce service offre haute disponibilité, sauvegardes automatiques, réplication et sécurité réseau intégrée via le Virtual Network.

#### f. Azure Machine Learning

Azure Machine Learning est au cœur du dispositif de machine learning. Il est utilisé pour :

- Le déploiement du modèle sous forme d'un endpoint HTTPS que Azure Functions peut interroger pour obtenir des prédictions en temps réel.

- L'entraînement automatique via des pipelines ou jobs planifiés qui relisent les données stockées dans le Storage Account pour mettre à jour le modèle avec les derniers rendez-vous et statuts collectés.
- L'environnement Azure ML Studio, qui offre aux data scientists un espace collaboratif pour explorer les données, tester de nouveaux modèles, visualiser les résultats et monitorer les performances.

#### g. Azure Storage Account

Le Storage Account est utilisé comme stockage centralisé, permettant de gérer :

- Les fichiers entrants contenant les informations des nouveaux rendez-vous.
- Les mises à jour effectuées après la réalisation ou annulation.
- Les datasets d'entraînement utilisés périodiquement par Azure Machine Learning.
- Les résultats de prédiction retournés par le modèle, ainsi que les logs d'exécution produits par Azure Functions.

Grâce à ses fonctionnalités de versioning et de cycle de vie, le Storage Account facilite la gestion de l'historique des données et leur archivage.

#### h. Azure Active Directory (Azure AD) et Azure Monitor

- **Azure AD** : Gère les identités et les accès, définissant les rôles et politiques pour contrôler les interactions entre les composants (ex. : Azure Functions ne peut écrire que dans un conteneur spécifique du Storage Account, Azure ML ne lit que certains fichiers).
- **Azure Monitor** : Fournit des métriques, logs et tableaux de bord pour surveiller :
  1. Le bon déroulement des fonctions Azure Functions.
  2. Le trafic des endpoints API Management.
  3. L'état des jobs d'entraînement Azure Machine Learning.
  4. Les alertes déclenchées en cas d'anomalie (latence, erreurs, échecs de job, etc.).

#### 8.3.d Chiffrage estimatif

Pour Microsoft Azure, l'estimation du coût mensuel s'appuie également sur un volume de 15 millions de rendez-vous traités chaque mois, impliquant 15 millions de requêtes POST et 15 millions de requêtes PUT, soit un total de 30 millions d'appels API mensuels. À cela s'ajoutent l'usage de machines virtuelles Azure (VM Azure) pour l'hébergement du front-end



et du back-end de l'application, ainsi que la gestion de la base de données relationnelle via Azure Database for MySQL, qui contribuent tous deux à la consommation des ressources et au coût global de l'infrastructure.

### Résumé du Chiffrage de l'Architecture Cloud Azure

*Total mensuel estimé : 4 924,49 €*

*Total annuel estimé : 59 093,88 €*

Services	Détails	Quantité	Tarification	Prix unitaire mensuel	Prix total mensuel
Azure Functions	1024 Mo RAM, 1000 ms exécution, 1M exécutions/mo (France Central & South)	4	à l'usage	8,44 €	16,88 €
API Management	Niveau Basic, 1 unité, 730 h (France Central)	1	Forfait mensuel	129,33 €	129,33 €
Azure Machine Learning	2 NC4as T4 v3 (4 cœurs, 28 Go RAM), 1 mois (France Central)	2	à l'usage	789,09 €	1587,18 €
Load Balancer	5 règles, 1 000 Go données traitées (France Central)	1	à l'usage	20,43 €	23.25 €
VPN Gateway	VPN de base, 10 tunnels S2S, 128 connexions P2S, 1000 Go (France Central)	2	à l'usage	30,76 €	61.52 €

Virtual Network	2 réseaux virtuels, 100 Go sortants chacun (France Central)	1	à l'usage	3,52 €	3,52 €
Machine virtuel	B2as v2 (2 vCPUs, 8 GB RAM)	2	à l'usage	54,53 €	109,06 €
Machine virtuel	1 B8as v2 (8 vCPUs, 32 GB RAM)	2	à l'usage	218,12 €	436,24 €
Base de données Azure MySQL	1 D8DS v4 (8 vCores) Stockage : 500 Go	2	à l'usage	566,53 €	1133,06 €
Storage Account	Stockage blob, 50 To, opérations diverses, LRS, niveau accès chaud (France South)	1	à l'usage	1124,49 €	1124,49 €
Protection DDos Azure	01 adresse ip	1	à l'usage	174,88 €	174,88 €
AzureExpress Route	standard	1	à l'usage	48,33 €	48,33 €
Passerelle VPN -app doctolib	24/7 h 500 Go	1	à l'usage	50,90 €	50,90 €
Passerelle VPN – dev et datascientists	24/7 h 100 Go	1	à l'usage	25,85 €	25,85 €

## 8.4 Choix du fournisseur cloud

Après une analyse approfondie des différentes options disponibles, et notamment une comparaison avec Microsoft Azure, le choix s'est porté sur Amazon Web Services (AWS) comme plateforme cloud principale pour ce projet Doctolib. Plusieurs raisons motivent cette décision :

1. **Maturité et richesse de l'écosystème**

AWS est reconnu comme un leader mondial du cloud computing, offrant une large palette de services managés matures, fiables et innovants. Son écosystème complet couvre tous les besoins du projet, de l'hébergement des applications aux services avancés de machine learning, en passant par le stockage sécurisé et la gestion fine des accès.

2. **Flexibilité et scalabilité**

Les services AWS, notamment Lambda et SageMaker, permettent de concevoir une architecture serverless hautement scalable, capable de s'adapter automatiquement à des volumes très importants de données et de requêtes, ce qui est essentiel pour gérer les 30 millions d'appels API mensuels prévus.

3. **Sécurité et conformité**

AWS met à disposition des outils robustes pour sécuriser les échanges (VPN Site-to-Site), gérer les identités et accès (IAM), et assurer la conformité avec le RGPD. Cette priorité à la sécurité est indispensable dans le contexte sensible des données médicales traitées par Doctolib.

4. **Performances et disponibilité**

Avec son infrastructure mondiale, AWS garantit une faible latence et une haute disponibilité des services, assurant une expérience utilisateur fluide et fiable pour les professionnels de santé et patients.

5. **Coût compétitif par rapport à Azure**

Comparativement à Azure, AWS propose une tarification souvent plus compétitive et transparente pour des charges similaires, ce qui permet d'optimiser le budget sans compromettre la qualité ni la sécurité.

6. **Support et communauté**

AWS bénéficie d'une large communauté d'utilisateurs et d'un support technique de qualité, facilitant le développement, le déploiement et la maintenance de l'application.

En résumé, AWS offre un cadre robuste, éprouvé, évolutif et économique parfaitement adapté aux enjeux techniques, réglementaires et opérationnels de Doctolib. Ce choix stratégique, renforcé par une comparaison favorable avec Azure, assure la pérennité et la réussite du projet sur le long terme.

## 8.5 Conclusion

L'architecture cloud conçue pour le projet Doctolib repose sur des principes de résilience, de sécurité, de scalabilité et d'efficacité opérationnelle. Elle répond aux besoins critiques de

traitement en temps réel des données, d'automatisation de la prédiction des annulations, ainsi qu'à la nécessité d'offrir un environnement de travail collaboratif pour les data scientists.

Deux alternatives majeures ont été étudiées : **Amazon Web Services (AWS)** et **Microsoft Azure**. Chacune de ces plateformes propose un ensemble robuste de services, avec des fonctionnalités similaires en matière de virtualisation, de stockage, d'intelligence artificielle, de gestion d'API et de sécurité réseau. Toutefois, l'analyse comparative a révélé plusieurs avantages clés en faveur d'AWS.

D'une part, **AWS se distingue par la maturité de son écosystème**, notamment avec des services comme Lambda et SageMaker, qui permettent une intégration fluide entre l'inférence en temps réel et l'entraînement continu des modèles. D'autre part, **le coût global estimé sur AWS s'est révélé inférieur à celui sur Azure**, pour une architecture équivalente en termes de charge et de performance.

En somme, le **choix d'AWS comme cloud principal** repose sur une combinaison de critères objectifs : meilleure maîtrise des coûts, richesse des outils managés, fiabilité éprouvée à grande échelle, et conformité rigoureuse aux exigences de sécurité des données de santé. Ce choix stratégique offre ainsi à Doctolib une infrastructure solide, évolutive et prête à soutenir l'innovation continue autour de la gestion intelligente des rendez-vous.