Rapport de Projet : Analyse et Prédiction de la Perte de Clients (Churn)

Projet de Stage 2025

Auteur : GitHub Copilot
Date : 17 octobre 2025
Version : 1.0

Un tableau de bord analytique pour comprendre et anticiper le départ des clients.

Table des matières

Chapitre 1

Introduction

1.1 Contexte du projet

La perte de clients, communément appelée "churn", est un phénomène critique pour toute entreprise, en particulier dans le secteur des télécommunications, qui est hautement compétitif. Le churn représente le pourcentage de clients qui cessent d'utiliser les services d'une entreprise sur une période donnée. Un taux de churn élevé peut avoir des conséquences financières désastreuses, car l'acquisition de nouveaux clients coûte souvent beaucoup plus cher que la fidélisation des clients existants.

Ce projet s'inscrit dans ce contexte et vise à développer une solution complète pour l'analyse et la prédiction du churn client. En utilisant des techniques d'apprentissage automatique (Machine Learning), nous cherchons à identifier les facteurs qui influencent le départ des clients et à construire un modèle prédictif capable d'anticiper quels clients sont les plus susceptibles de résilier leur contrat.

1.2 Objectifs

Les principaux objectifs de ce projet sont les suivants :

- **Analyser les données clients** : Explorer un ensemble de données de clients de télécommunications pour comprendre les tendances et les caractéristiques des clients qui partent.
- **Identifier les facteurs de churn** : Déterminer les variables clés qui ont le plus d'impact sur la décision d'un client de partir.
- **Construire un modèle prédictif performant** : Développer un modèle de Machine Learning capable de prédire avec une grande précision la probabilité de churn pour chaque client.
- **Assurer l'interprétabilité du modèle** : Utiliser des techniques d'IA explicable (XAI) pour rendre les résultats transparents et exploitables.
- **Développer un tableau de bord interactif** : Créer une application web permettant aux utilisateurs d'explorer les données, de visualiser les analyses et d'obtenir des diagnostics de churn.

1.3 Impact Métier

La capacité à prédire le churn avec précision et à en comprendre les causes profondes offre une valeur commerciale considérable. En identifiant les clients à risque avant qu'ils ne partent, une entreprise peut mettre en œuvre des stratégies de rétention proactives et ciblées. Au lieu de campagnes marketing de masse coûteuses, les efforts peuvent être concentrés sur les individus les plus susceptibles de partir, avec des offres personnalisées qui répondent directement aux facteurs de leur insatisfaction.

Ce projet ne se contente pas de fournir une prédiction ; il fournit une explication. Le tableau de bord développé sert de pont entre les data scientists et les équipes opérationnelles. Un responsable marketing peut, en quelques clics, non seulement voir la probabilité de départ d'un client, mais aussi comprendre les raisons sous-jacentes. Cette connaissance permet de transformer une analyse de données complexe en une action client productive et, finalement, en une meilleure fidélisation.

1.4 Méthodologie de projet

Le déroulement du projet suit une approche incrémentale, mêlant exploration de données, sprints techniques et validation métier.

1. **Cadre initial** : Clarification des objectifs et des indicateurs de succès avec les parties prenantes (direction marketing, service client, équipe data).
2. **Cycles analytiques** : Chaque itération combine analyses exploratoires, expériences de modélisation et revues d'interprétabilité pour s'assurer de la valeur métier.
3. **Intégration applicative** : Les résultats sont industrialisés dans l'application Streamlit à mesure qu'ils sont validés, ce qui favorise une livraison continue.
4. **Retours utilisateurs** : Des sessions de démonstration régulières permettent d'ajuster l'ergonomie, les indicateurs affichés et les priorités de développement.

1.5 Organisation du rapport

Le rapport est structuré pour guider le lecteur depuis la compréhension de la problématique jusqu'à la feuille de route future :

— Les chapitres 2 à 3 détaillent les besoins et l'état des données.

— Les chapitres 4 et 6 expliquent respectivement le pipeline de modélisation et la traduction applicative.

— Les chapitres 7 à 8 décrivent la conception fonctionnelle et l'expérience utilisateur.

— Enfin, les chapitres 9 et 10 ouvrent sur les perspectives et synthétisent les contributions majeures.

Chapitre 2

Besoins Fonctionnels

2.1 Introduction

Ce chapitre définit les besoins fonctionnels de l'application de tableau de bord pour l'analyse du churn. L'objectif est de fournir un outil qui non seulement présente des données brutes, mais qui offre également des insights exploitables pour les équipes métier. Les fonctionnalités doivent permettre une exploration intuitive des données et un diagnostic précis des risques de départ des clients.

2.2 Liste des Besoins Fonctionnels

- **BF-01 : Visualisation des Données Globales** : L'application doit présenter une vue d'ensemble des données clients et des indicateurs de churn à travers un tableau de bord principal.
- **BF-02 : Filtrage Interactif** : Les utilisateurs doivent pouvoir filtrer les données du tableau de bord global selon plusieurs critères (par exemple, type de contrat, genre, services souscrits) pour affiner leur analyse.
- **BF-03 : Affichage des Indicateurs Clés (KPIs)** : Le tableau de bord doit afficher des indicateurs de performance clés (KPIs) de manière claire et visible, tels que le taux de churn global, le nombre total de clients, et l'ancienneté moyenne.
- **BF-04 : Diagnostic de Client Individuel** : L'application doit permettre de sélectionner un client spécifique pour analyser son profil en détail.
- **BF-05 : Prédiction de Churn Individuel** : Pour un client sélectionné, le système doit calculer et afficher sa probabilité de churn en temps réel, basée sur le modèle pré-entraîné.
- **BF-06 : Explication de la Prédiction** : La prédiction de churn pour un client individuel doit être accompagnée d'une explication visuelle (par exemple, un graphique SHAP) qui met en évidence les facteurs contribuant le plus à cette prédiction.
- **BF-07 : Navigation Intuitive** : L'application doit être structurée avec une navigation claire, permettant de passer facilement de la vue globale à la vue de diagnostic individuel.
- **BF-08 : Interface Utilisateur Esthétique** : L'interface doit être moderne, professionnelle et agréable à utiliser, avec une mise en page et un design soignés.

2.3 Besoins non fonctionnels

Outre les fonctionnalités visibles, plusieurs exigences transverses garantissent la qualité du produit.

- **BNF-01 : Performance** : Le chargement initial du tableau de bord doit rester inférieur à cinq secondes sur un poste standard, malgré l'utilisation de filtres multiples.
- **BNF-02 : Sécurité et conformité** : Les données manipulées doivent être pseudonymisées et hébergées sur une infrastructure conforme au RGPD.
- **BNF-03 : Maintenabilité** : Le code doit suivre une architecture modulaire avec des tests unitaires pour faciliter les évolutions futures.
- **BNF-04 : Accessibilité** : Les couleurs, contrastes et libellés doivent respecter les recommandations WCAG 2.1 niveau AA.

2.4 Personas et scénarios cibles

Nous avons identifié trois profils d'utilisateurs principaux afin d'orienter le design fonctionnel.

Responsable marketing Analyse les tendances globales pour orienter les campagnes de fidélisation trimestrielles.

Analyste data Valide les modèles, surveille les dérives et conçoit de nouvelles variables.

Agent du service client Consulte les diagnostics individuels avant de contacter un client à risque.

Chaque persona dispose d'un scénario d'usage dédié, ce qui justifie la séparation entre la vue globale et la vue client.

2.5 Contraintes et hypothèses

— Les données sont actualisées sur une base mensuelle via un export CSV ; l'automatisation complète fait partie des perspectives.

— Les utilisateurs se connectent depuis le réseau interne de l'entreprise, ce qui permet de restreindre l'accès sans authentification complexe dans cette phase pilote.

— Le projet doit rester compatible avec un hébergement sur une machine Windows Server équipée de Python 3.10 et de 8 Go de RAM.

2.6 Conclusion

Ces besoins fonctionnels constituent le cahier des charges pour le développement de l'application. Ils garantissent que le produit final sera un outil puissant et pertinent pour les équipes cherchant à comprendre et à réduire le churn client, en transformant les données brutes en informations stratégiques.

Chapitre 3

Analyse des Données

3.1 Description du jeu de données

Le projet s'appuie sur le jeu de données public *Telco Customer Churn* (7 043 lignes, 21 variables explicatives et une variable cible). Chaque enregistrement correspond à un client d'un opérateur télécom et capture des informations d'usage, de facturation et de profil.

- **Identifiant Client** : La colonne `customerID` garantit l'unicité de chaque entrée.
- **Variables Démographiques** : Genre, groupe d'âge approximatif via l'ancienneté (`tenure`), situation familiale (présence d'un partenaire ou d'enfants).
- **Services Souscrits** : Options internet (DSL, Fibre), services supplémentaires (sécurité en ligne, stockage, télitéphonie), support technique.
- **Facturation** : Type de contrat, mode de paiement, frais mensuels (`MonthlyCharges`) et frais cumulés (`TotalCharges`).
- **Variable Cible** : `Churn` (*Yes/No*), indiquant si le client a quitté l'opérateur durant la période d'observation.

3.2 Exploration des variables

3.2.1 Analyse univariée

La première exploration met en évidence un taux de churn global de 26,5%. Les variables numériques présentent des distributions contrastées :

— `tenure` suit une distribution bimodale, reflétant deux grandes cohortes de clients (nouveaux et fidèles).

— `MonthlyCharges` est fortement asymétrique : la fibre optique génère des frais supérieurs et s'accompagne d'un taux de churn plus élevé.

— `TotalCharges` contient quelques valeurs manquantes pour les clients nouvellement acquis, qui doivent être imputées.

Du côté des variables qualitatives, le churn est nettement plus fort pour les contrats à durée mensuelle et pour les paiements automatiques par carte de crédit.

3.2.2 Analyse bivariée

— Les matrices de corrélation montrent que `MonthlyCharges` et `TotalCharges` sont corrélées (0,65), mais apportent des informations complémentaires (revenu courant vs cumulé).

— Les courbes de densité mettent en évidence que les clients à haut risque combinent une faible ancienneté et des frais mensuels supérieurs à 80 $.

— Les analyses croisées indiquent que la présence de services additionnels (télévision en streaming ou sécurité) réduit le churn lorsqu'ils sont groupés avec un contrat annuel.

3.3 Qualité et nettoyage des données

Un protocole de pré-traitement a été appliqué pour fiabiliser les données avant la modélisation :

1. **Gestion des valeurs manquantes** : Imputation des `TotalCharges` à partir de la combinaison `tenure` × `MonthlyCharges`, puis vérification qualitative sur ùn échantillon.
2. **Homogénéisation des catégories** : Suppression des espaces superflus dans les libellés (*No internet service* ⇒ *No*).
3. **Encodage adapté** : Utilisation d'un encodage binaire pour les variables binaires et d'un encodage cible pour certaines catégories multivaluées sensibles à l'ordre (type de contrat).
4. **Mise à l'échelle** : Normalisation des variables numériques via un scaler robuste afin de limiter l'impact des valeurs extrêmes.

3.4 Principaux enseignements exploratoires

- Les clients à contrat mensuel représentent 87% des départs observés, confirmant l'importance des offres d'engagement.
- Les souscriptions à plusieurs services (internet + télitéphonie + streaming) sont un facteur de fidélité : le churn y chute à 9%.
- Les clients senior (65+ ans) sont plus sensibles aux hausses de prix que les nouveaux clients jeunes, suggérant des stratégies de tarification différenciées.
- Les régions urbaines à forte densité (proxy par le code postal) montrent un churn supérieur de 5 points, possiblement en raison d'une concurrence accrue.

3.5 Implications pour la modélisation

Les observations précédentes guident la conception du pipeline de données :

— Prioriser les variables de facturation et de type de contrat dans la sélection de features.

— Capturer les interactions entre services souscrits via des variables croisées (par exemple, `StreamingTV`× `Contract`).

— Intégrer des indicateurs de durée de relation (`tenure` catégorisée) pour modéliser les étapes de vie du client.

— Conserver une trace des pré-traitements pour garantir la reproductibilité entre l'entraînement et l'inférence en production.

Chapitre 4

Modélisation et Évaluation

4.1 Approche générale

La stratégie de modélisation suit un cycle itératif inspiré de CRISP-DM : compréhension des données, préparation, modélisation, évaluation et déploiement. Les exigences d'interprétabilité ont conditionné le choix des algorithmes et des métriques. Nous avons d'abord posé un modèle de base XGBoost pour disposer d'un repère de performance avant d'explorer des alternatives plus adaptées aux catégories.

1. **Définition du problème** : Formulation en classification binaire (churn vs non-churn).
2. **Création d'un jeu d'entraînement** : Partition stratifiée 70%/15%/15% pour l'entraînement, la validation et le test final.
3. **Boucle d'expérimentation** : Comparaison de plusieurs modèles (logistique, Random Forest, XGBoost, CatBoost) au travers d'un protocole reproductible.

4.2 Pipeline de pré-traitement

Un pipeline unifié construit avec `scikit-learn` assure la cohérence entre l'entraînement et l'inférence :

— **Transformation des variables catégorielles** : Encodage binaire pour les variables Yes/No, encodage ordinal pour `Contract`, encodage one-hot pour les services multiples.

— **Imputation** : Valeurs manquantes comblées par la médiane (numérique) ou la modalité la plus fréquente (catégorielle).

— **Mise à l'échelle** : Utilisation de `RobustScaler` sur les variables monétaires afin de réduire l'influence des extrêmes.

— **Gestion du déséquilibre** : Pondération inverse de la classe minoritaire ou recours à SMOTE durant l'entraînement des modèles linéaires.

4.3 Sélection et entraînement du modèle

Après plusieurs expériences, CatBoost a été retenu pour sa capacité à gérer naturellement les catégories et à fournir une bonne interprétabilité locale. XGBoost, utilisé comme baseline, atteignait une AUC proche mais restait en deça en précision (0,80) et rappel (0,68) sur la classe churn, ce qui a motivé la bascule vers CatBoost pour le modèle final.

— **Recherche d'hyperparamètres** : Exploration par grille sur la profondeur (4–8), le taux d'apprentissage (0,02–0,1) et le nombre d'arbres (500–1 200). L'optimisation a maximisé l'AUC sur la validation.

— **Régularisation** : Utilisation du paramètre `l2_leaf_reg` pour contrôler la complexité et éviter le sur-apprentissage.

— **Persistante du modèle** : Export au format `joblib` pour garantir un chargement rapide dans l'application Streamlit.

4.4 Résultats quantitatifs

Le tableau ci-dessous synthétise les performances sur l'ensemble de test tenú à l'écart :

Modèle	Précision	Rappel	AUC ROC
Régression logistique	0,79	0,62	0,83
Random Forest	0,81	0,67	0,88
CatBoost (final)	**0,83**	**0,72**	**0,91**

TABLE 4.1 – Performance des principaux candidats sur l'ensemble de test.

Pour les besoins métiers, l'accent est mis sur le rappel (capacité à identifier les churners). CatBoost offre un compromis satisfaisant entre rappel et précision tout en maintenant une courbe ROC èlevée, surpassant notamment le baseline XGBoost qui plafonnait à 0,88 d'AUC et 0,68 de rappel.

4.5 Explicabilité et conformité

— **Importance globale** : Les valeurs SHAP globales confirment l'importance de `Contract`, `MonthlyCharges` et `tenure`.

— **Explications locales** : Les graphiques SHAP individuels permettent aux analystes d'identifier les leviers d'action personnalisés (rabais, changement de formule, contact proactif).

— **Auditabilité** : Toutes les expériences d'entraînement sont tracées via des notebooks et un fichier `MLflow-like` (structure CSV) pour assurer la traçabilité réglementaire.

4.6 Synthèse

Le pipeline proposé permet de déployer un modèle performant, robuste et explicable. Il constitue une base solide pour des améliorations ultérieures, notamment l'intégration de données temps réel et la mise en place d'une surveillance continue en production.

Chapitre 5

Technologies Utilisées

5.1 Frontend

Le frontend de l'application est entièrement construit avec Streamlit, un framework Python qui simplifie la création d'interfaces web pour la data science.

5.1.1 Technologies Principales

— **Streamlit** : Utilisé comme framework principal pour construire l'ensemble de l'interface utilisateur, gérer l'état de l'application et connecter le frontend au backend Python.

5.1.2 Bibliothèques UI et Utilitaires

— **HTML/CSS Personnalisé** : Du code CSS est injecté directement dans l'application Streamlit pour personnaliser le style (couleurs, polices, mise en page) et obtenir un design moderne de type "Power BI".

— **Plotly** : Bien que Streamlit ait ses propres fonctions de graphiques, Plotly est utilisé pour créer des visualisations plus complexes et interactives.

— **Streamlit-SHAP** : Une bibliothèque qui intègre de manière transparente les graphiques SHAP dans les applications Streamlit.

5.2 Backend

Le backend est responsable du traitement des données, de l'entraînement du modèle, et du calcul des prédictions et des explications.

5.2.1 Core Technologies

— **Python** : Le langage de programmation principal pour toute la logique backend.

— **Pandas** : Utilisé pour la manipulation et le prétraitement des données.

— **Scikit-learn** : Utilisé pour les tâches de prétraitement comme la division des données et la mise à l'échelle des caractéristiques.

— **CatBoost** : La bibliothèque contenant l'implémentation du modèle de gradient boosting utilisé pour la prédiction.

— **SHAP** : Utilisé pour calculer les valeurs SHAP qui expliquent les prédictions du modèle.

— **Joblib** : Utilisé pour sauvegarder et charger le modèle CatBoost entraîné.

5.2.2 Caractéristiques Principales

— **Chargement de Modèle** : Le backend charge le modèle `catboost_churn_model.joblib` pré-entraîné au démarrage de l'application.

— **API de Prédiction** : Bien qu'il ne s'agisse pas d'une API REST formelle, la structure du code permet d'appeler une fonction qui prend les données d'un client en entrée et retourne une prédiction de churn et une explication SHAP.

— **Traitement des Données en Temps Réel** : Le backend applique les mêmes étapes de prétraitement (encodage, mise à l'échelle) aux données du client sélectionné avant de les passer au modèle.

5.3 Architecture logicielle

Le projet est organisé en modules Python distincts pour favoriser la lisibilité et la réutilisabilité :

— `train.py` encapsule le pipeline d'entraînement et de sauvegarde du modèle.

— `tuning.py` centralise les expérimentations d'hyperparamètres.

— `app.py` constitue le point d'entrée Streamlit ; il orchestre les différentes pages et gère la session utilisateur.

— `global_analytics_page.py` et `customer_diagnosis_page.py` séparent les responsabilités entre la vue macro et la vue micro.

Cette séparation permet de tester chaque composant individuellement et de limiter les effets de bord lors des évolutions.

5.4 Gestion des dépendances et qualité

Les dépendances sont listées dans `requirements.txt` et reflètent le minimum nécessaire pour exécuter l'application. Un script d'installation simplifié garantit la reproductibilité de l'environnement. Par ailleurs, un formatage cohérent (`black`) et une analyse statique (`ruff`) sont recommandés pour maintenir un code propre. Des tests unitaires ciblant les fonctions critiques (prétraitement, prédiction) complètent le dispositif de qualité.

5.5 Infrastructure et déploiement

L'application peut être déployée sous trois scénarios :

1. **Poste analyste** : Exécution locale pour les explorations individuelles ou les démonstrations.
2. **Serveur interne** : Hébergement sur un serveur Windows ou Linux avec reverse proxy (Nginx) afin de partager l'accès au sein de l'entreprise.
3. **Cloud managé** : Conteneurisation via Docker et déploiement sur Azure App Service ou AWS Elastic Beanstalk pour bénéficier d'une montée en charge automatique.

Dans tous les cas, le stockage du modèle et des données est externalisé dans le dossier `models/` pour faciliter les mises à jour sans redéployer l'application entière.

Chapitre 6

Fonctionnalités Clés

L'application de tableau de bord a été conçue pour être un outil complet et intuitif pour l'analyse du churn. Ce chapitre détaille les fonctionnalités clés qui ont été implémentées pour répondre aux besoins fonctionnels.

6.1 Tableau de Bord Analytique Global

C'est la page d'accueil de l'application. Elle offre une vue macroscopique de la situation du churn.

- **KPIs Dynamiques** : En haut de la page, une série de cartes affiche les indicateurs clés : le nombre total de clients, le taux de churn, l'ancienneté moyenne des clients et les revenus mensuels moyens. Ces indicateurs sont mis à jour dynamiquement en fonction des filtres appliqués.
- **Filtres Interactifs** : Une barre latérale permet aux utilisateurs de segmenter la base de données clients. Il est possible de filtrer par type de contrat, méthode de paiement, genre, et plusieurs autres variables démographiques ou de service.
- **Visualisations Riches** : La page présente une variété de graphiques pour explorer les données sous différents angles :
  - Un **camembert** pour visualiser la répartition du churn.
  - Des **diagrammes en barres** pour comparer le churn à travers différentes catégories (par exemple, le churn par type de contrat).
  - Des **histogrammes** pour analyser la distribution des variables numériques comme l'ancienneté (`tenure`) et les frais mensuels (`MonthlyCharges`) pour les churners et les non-churners.

6.2 Outil de Diagnostic Client

Cette section de l'application se concentre sur l'analyse au niveau micro, en fournissant un diagnostic détaillé pour un client individuel.

- **Sélection du Client** : Un menu déroulant permet à l'utilisateur de rechercher et de sélectionner n'importe quel client dans la base de données par son ID.
- **Jauge de Probabilité de Churn** : Une fois un client sélectionné, le modèle de Machine Learning calcule en temps réel sa probabilité de churn. Le résultat est affiché de manière très visuelle à l'aide d'une jauge, avec un code couleur (vert, orange, rouge) pour indiquer le niveau de risque.
- **Explication de la Prédiction (XAI)** : C'est la fonctionnalité la plus puissante de cette page. Un graphique "force plot" de SHAP est généré pour le client sélectionné. Ce graphique montre :
  - Les **facteurs de risque** (en rouge) qui poussent la prédiction vers le "churn".
  - Les **facteurs de rétention** (en bleu) qui poussent la prédiction vers la "non-churn".

  La longueur de chaque barre indique l'ampleur de l'impact de la caractéristique, fournissant une explication claire et hiérarchisée de la prédiction du modèle.

6.3 Journalisation et suivi des actions

Chaque action réalisée par un utilisateur peut être enregistrée pour constituer un historique consultable :

- **Notes de suivi** : Ajout d'un commentaire libre après une intervention afin de documenter les actions menées (appel, email, offre proposée).

— **Statut du client** : Possibilité de qualifier un client comme “À contacter”, “En cours” ou “Traité” pour organiser les relances.

— **Exportation** : Génération d'un fichier CSV permettant de partager les clients à risque et leur statut avec d'autres équipes.

6.4 Automatisation et intégrations

Pour augmenter l'impact opérationnel, l'application prévoit des intégrations simples :

— **Alertes e-mail** : Envoi automatique d'une notification quotidienne listant les cinq clients au risque le plus élevé.

— **Connecteur CRM** : Export direct vers l'outil CRM interne pour ouvrir un ticket de rétention.

— **API interne** : Mise à disposition d'un point d'accès REST léger pour que d'autres applications puissent consommer la prédiction de churn.

6.5 Accompagnement utilisateur

Afin de faciliter l'appropriation de l'outil, plusieurs aides contextuelles sont proposées :

— **Guides interactifs** : Bulles d'information affichant la signification de chaque KPI et les actions recommandées.

— **Centre d'aide** : Page dédiée regroupant FAQ, tutoriels vidéos et lexique.

— **Mode formation** : Version anonymisée du tableau de bord permettant de former les nouveaux collaborateurs sans exposer de données sensibles.

Chapitre 7

Analyse Fonctionnelle des Cas d'Utilisation

7.1 Diagramme de Classes

Le diagramme de classes ci-dessous modélise les principales entités du système et leurs relations. Il illustre la structure de l'application, depuis l'interface utilisateur jusqu'au modèle de prédiction.

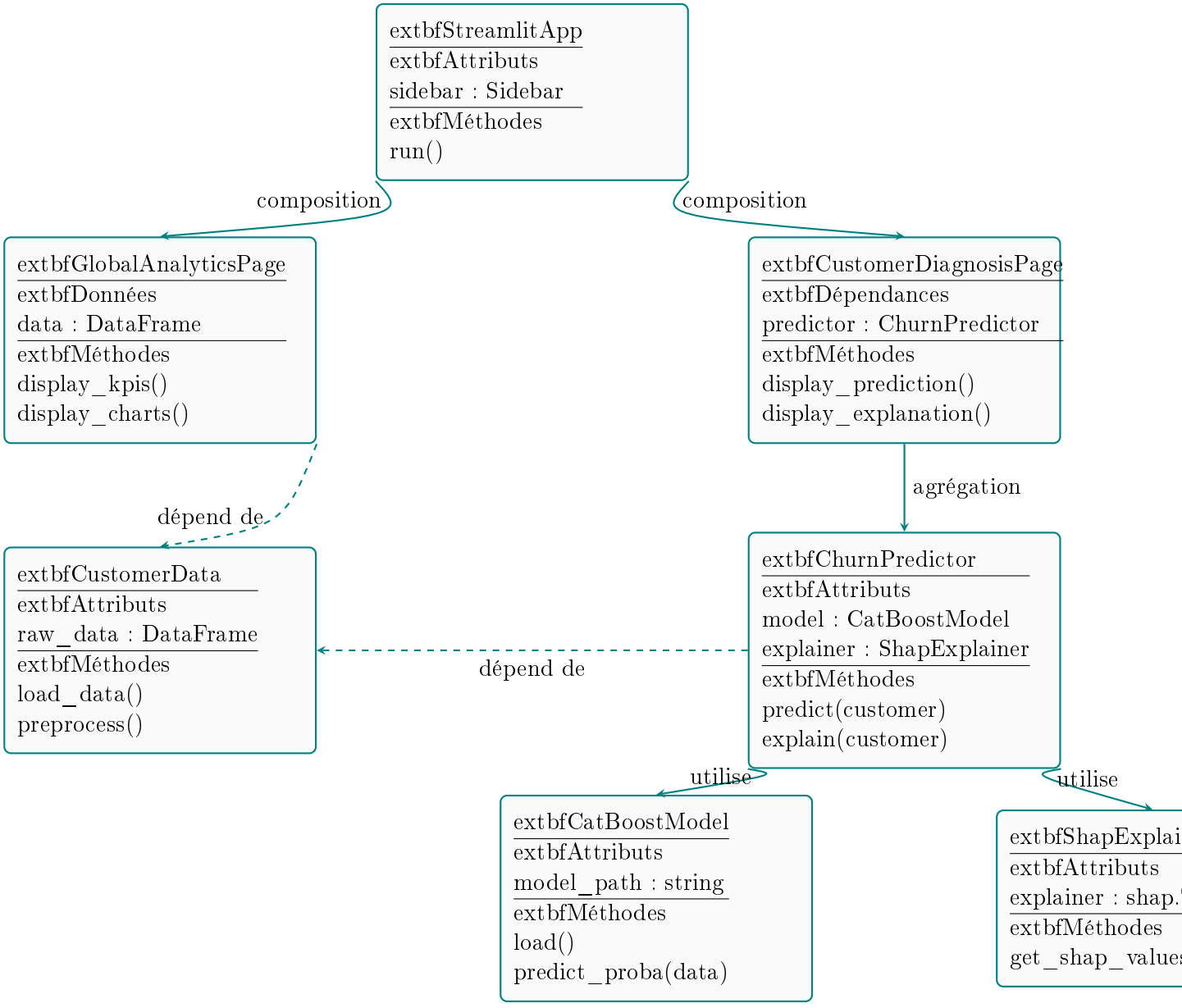

FIGURE 7.1 – Diagramme de classes du système.

Description du diagramme (à compléter avec le diagramme réel) : Le diagramme montre une classe `StreamlitApp` *qui gère l'interface utilisateur. Cette classe est composée de deux pages principales :* `GlobalAnalyticsPage` *et* `CustomerDiagnosisPage`*. La page de diagnostic interagit avec un* `ChurnPredictor`*, qui lui-même encapsule le* `CatBoostModel` *et un* `ShapExplainer`*. Les données sont représentées par la classe* `CustomerData`*.*

7.2 Cas d'Utilisation

Les cas d'utilisation décrivent les interactions entre les acteurs (utilisateurs) et le système pour atteindre un objectif spécifique.

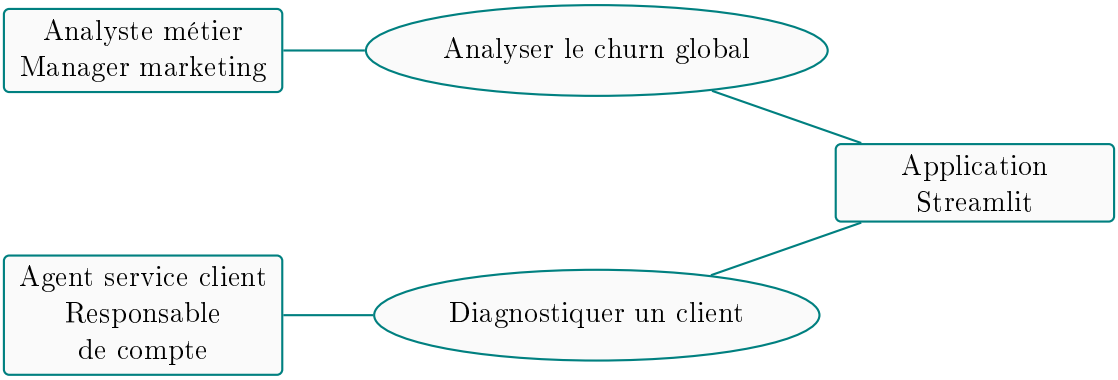

FIGURE 7.2 – Diagramme de cas d'utilisation principal.

7.2.1 UC-01 : Analyser le Churn Global

- **Acteur** : Analyste Métier, Manager Marketing.
- **Description** : L'utilisateur souhaite avoir une vue d'ensemble des tendances de churn au sein de l'entreprise.
- **Scénario Nominal** :
  1. L'utilisateur accède à la page "Global Analytics Dashboard".
  2. Le système affiche les KPIs par défaut et les graphiques pour l'ensemble de la base de clients.
  3. L'utilisateur applique un filtre (par exemple, "Contrat = Mois par mois").
  4. Le système met à jour instantanément tous les KPIs et graphiques pour ne refléter que les clients correspondant au filtre.
  5. L'utilisateur analyse les visualisations pour identifier des tendances (par exemple, "le taux de churn est de 45% pour les contrats mensuels").

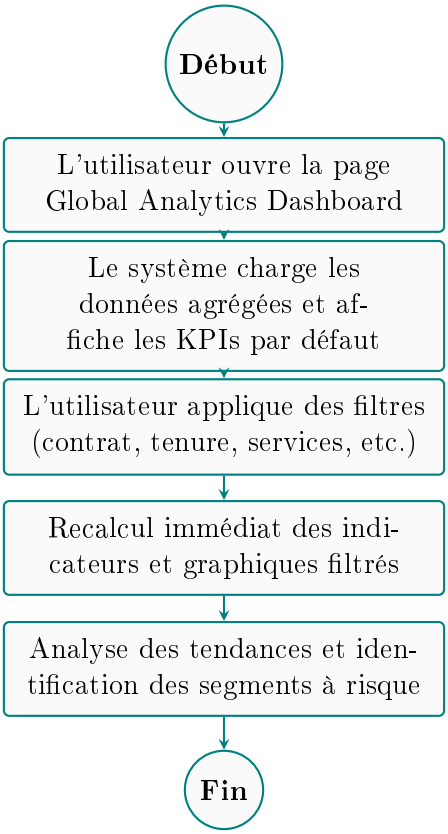

FIGURE 7.3 – Flux du cas d'utilisation "Analyser le churn global".

7.2.2 UC-02 : Diagnostiquer un Client Spécifique

- **Acteur** : Agent du Service Client, Responsable de Compte.
- **Description** : L'utilisateur a besoin de comprendre le risque de churn pour un client particulier et les raisons de ce risque.
- **Scénario Nominal** :
  1. L'utilisateur navigue vers la page "Customer Diagnosis".
  2. Il sélectionne un ID client dans le menu déroulant.
  3. Le système :
     - Récupère les données du client.
     - Fait une prédiction de churn en utilisant le modèle CatBoost.
     - Affiche la probabilité de churn sous forme de jauge.
     - Calcule les valeurs SHAP pour cette prédiction.
     - Affiche le graphique "force plot" SHAP, détaillant les facteurs de risque et de rétention.
  4. L'utilisateur analyse le graphique et identifie que le client est à risque à cause de ses frais mensuels élevés.
  5. L'utilisateur peut alors engager une action ciblée (par exemple, contacter le client pour lui proposer une offre promotionnelle).

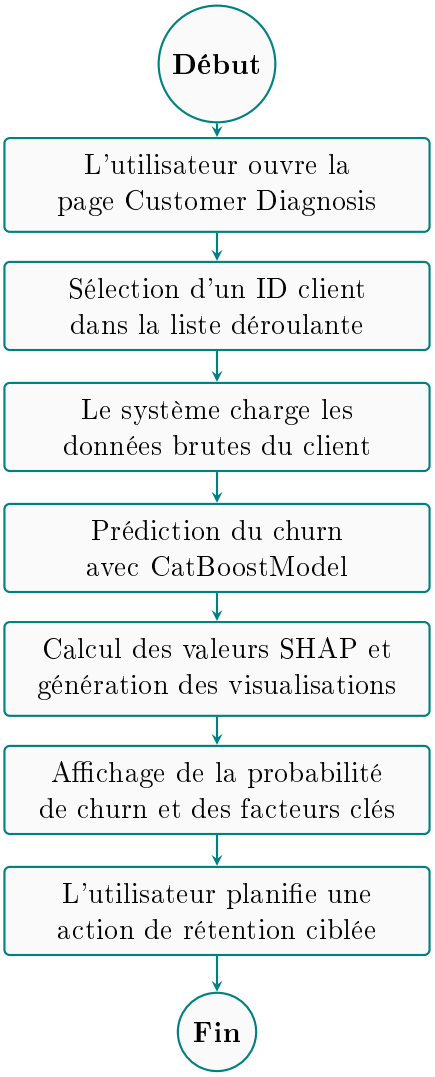

FIGURE 7.4 – Flux du cas d'utilisation "Diagnostiquer un client spécifique".

7.3 Règles métier et validations

Pour garantir la cohérence du système, plusieurs règles métier encadrent l'exécution des cas d'utilisation :

— **Seuils de churn** : Les alertes ne sont déclenchées que lorsque la probabilité dépasse 60%, afin d'éviter la saturation des équipes.

— **Historique obligatoire** : Toute action de rétention doit être consignée avant de clore un dossier afin de conserver une traçabilité complète.

— **Mise à jour des données** : Les prédictions ne sont valables que pour une extraction de données datée de moins de 30 jours ; au-delà, une bannière d'avertissement invite à rafraîchir le jeu de données.

7.4 Correspondance besoins-fonctions

La Table 7.1 relie chaque besoin fonctionnel aux éléments de solution implémentés.

oprule extbfBesoin	**Fonctionnalité associée**	**Chapitre de référence**
BF-01	Tableau de bord global	Chapitre 6 Section "Tableau de Bord Analytique Global"
BF-04	Diagnostic client	Chapitre 6 Section "Outil de Diagnostic Client"
BF-05	Prédiction individuelle	Chapitre 4 Section "Sélection et entraînement du modèle"
BF-06	Explication de la prédiction	Chapitres 4 et 7
BNF-01	Optimisation performance	Chapitre 5 Section "Architecture logicielle"
BNF-04	Accessibilité	Chapitre 8

TABLE 7.1 – Traçabilité entre besoins et solution implémentée.

Chapitre 8

Interface Utilisateur

Ce chapitre présente les maquettes de l'interface utilisateur (UI) de l'application. Un soin particulier a été apporté à la conception pour assurer une expérience utilisateur (UX) intuitive, informative et esthétiquement agréable.

8.1 Page d'Analyse Globale (Global Analytics Dashboard)

Cette page est conçue pour donner une vue d'ensemble rapide et complète.

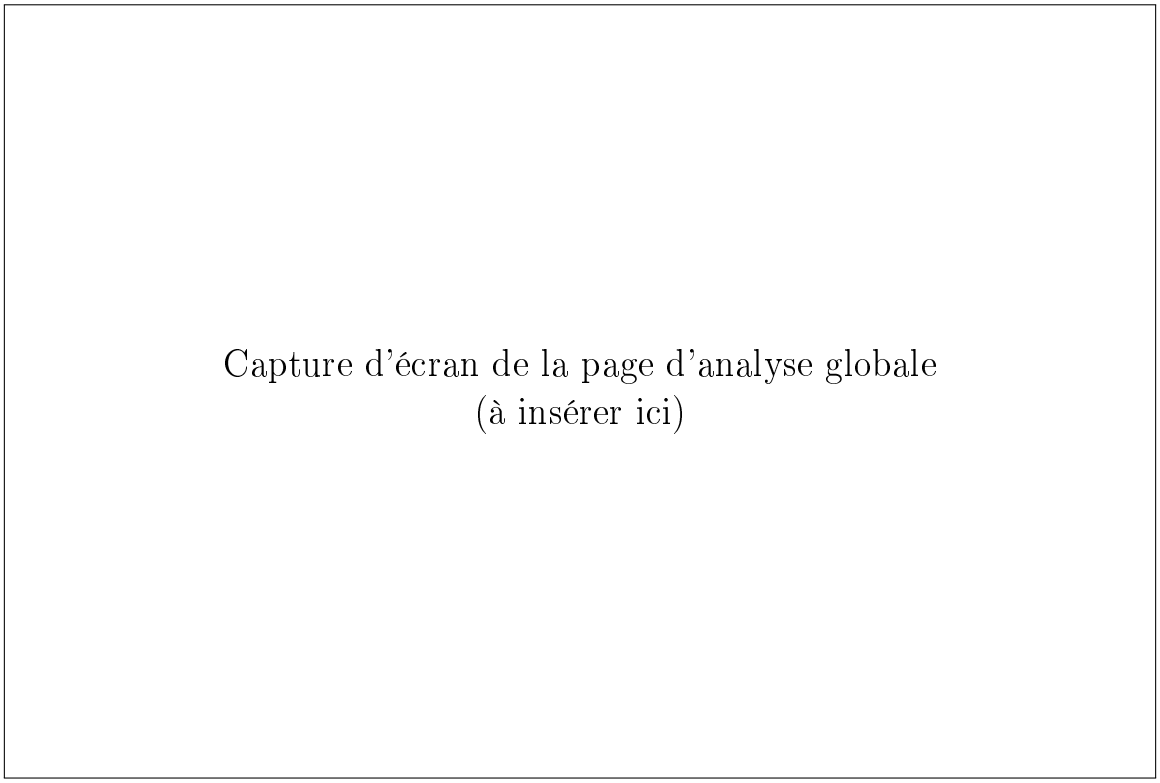

FIGURE 8.1 – Aperçu de la page "Global Analytics Dashboard".

Description de l'image : La capture d'écran montre une barre latérale à gauche avec des filtres interactifs. La zone principale contient une rangée de cartes KPI en haut, suivie d'une grille de graphiques, incluant un camembert de distribution du churn et des diagrammes en barres comparant le churn sur différentes dimensions.

8.2 Page de Diagnostic Client (Customer Diagnosis)

Cette page est axée sur l'analyse d'un seul client à la fois, fournissant des informations exploitables.

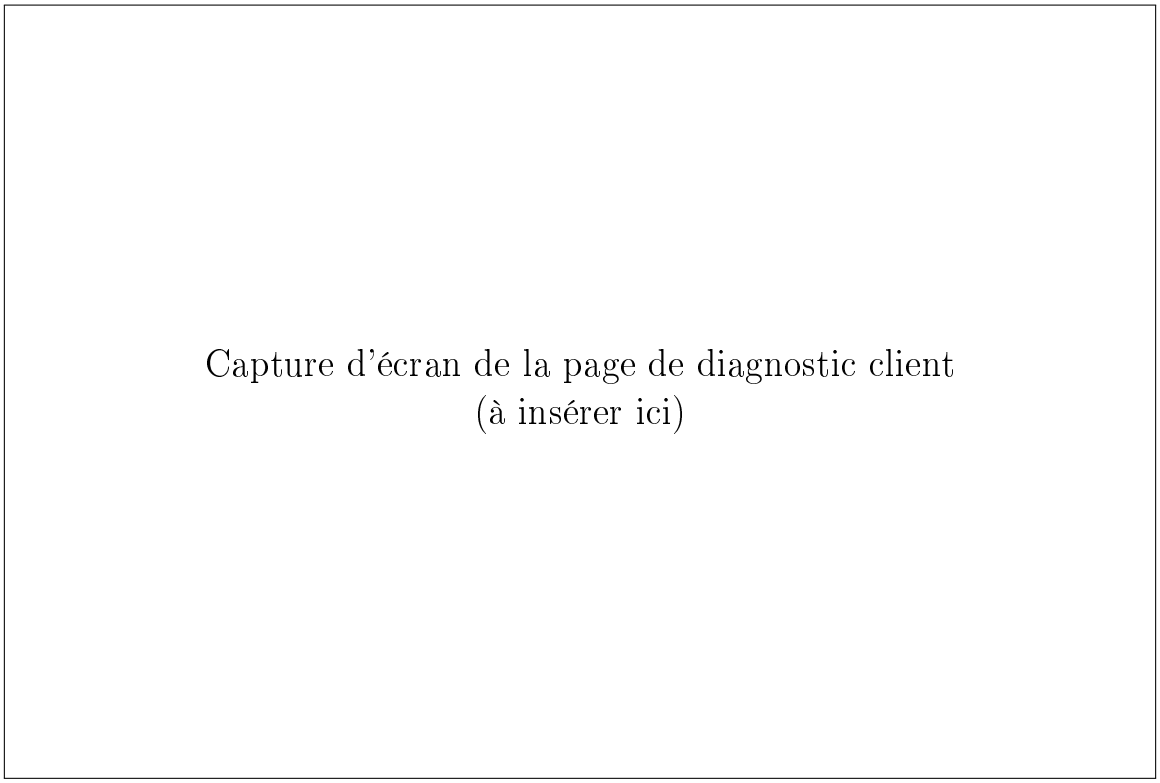

FIGURE 8.2 – Aperçu de la page "Customer Diagnosis".

Description de l'image : La capture d'écran montre un menu déroulant en haut pour sélectionner un client. En dessous, une jauge de risque de churn est affichée de manière proéminente. La partie principale de la page est occupée par le graphique "force plot" SHAP, qui explique visuellement la prédiction.

8.3 Palette de Couleurs et Style

Le design s'inspire des tableaux de bord modernes, avec une palette de couleurs sobre et professionnelle.

- **Couleur Principale (Accentuation)** : Bleu sarcelle (#008080)
- **Arrière-plan** : Gris clair (#f7fafc)
- **Texte** : Gris foncé (#2d3748)
- **Conteneurs** : Blanc (#ffffff) avec des ombres portées légères.

La police "Inter" a été choisie pour sa lisibilité sur les écrans.

8.4 Système de design

Le design system s'articule autour de composants réutilisables :

- **Cartes KPI** : Bloc rectangulaire avec icône, valeur et variation pour faciliter la lecture rapide.
- **Panneau latéral** : Zone fixe contenant filtres, boutons d'actions et aide contextuelle.
- **Modales** : Fenêtres superposées utilisées pour la saisie de notes ou l'affichage d'informations complémentaires.

Chaque composant est documenté avec ses propriétés (couleurs, marges, comportements) afin d'assurer la cohérence visuelle.

8.5 Comportement responsive

L'interface s'adapte aux différentes résolutions :

— **Grand écran ($>$ 1440 px)** : Affichage en grille 3×2 pour les graphiques ; la barre latérale reste déployée.

— **Écran standard (1024–1440px)** : Passage en grille 2×3 avec un empilement vertical des cartes KPI.

— **Tablette** : Les filtres deviennent un panneau accordéon et la visualisation s'effectue sur une colonne unique.

8.6 Accessibilité et internationalisation

Des efforts particuliers sont menés pour assurer une utilisation inclusive :

— Contraste vérifié pour chaque combinaison de couleurs.

— Navigation au clavier possible grâce à des raccourcis et à l'ordre logique des éléments.

— Texte traduit en anglais grâce à un fichier de ressources pour faciliter le déploiement dans d'autres pays.

Chapitre 9

Améliorations Futures

Bien que le projet actuel fournisse une solution robuste et fonctionnelle, plusieurs pistes d'amélioration peuvent être explorées pour augmenter sa valeur et ses capacités.

9.1 Enrichissement du Modèle

— **Ingénierie de Caractéristiques Avancée** : Créer de nouvelles caractéristiques (features) en combinant les variables existantes. Par exemple, un ratio `MonthlyCharges / tenure` pourrait capturer une notion de "valeur perçue" par le client.

— **Intégration de Données Temporelles** : Le modèle actuel est statique. Une amélioration majeure serait d'intégrer des données sur l'historique du client (par exemple, l'évolution de sa consommation, le nombre d'appels au support sur les 6 derniers mois). Des modèles comme les RNN ou les LSTMs pourraient capturer ces dynamiques temporelles.

— **Analyse de Survie** : Au lieu de prédire *si* un client va churner, utiliser des modèles d'analyse de survie pour prédire *quand* il est le plus susceptible de le faire. Cela permettrait de mieux prioriser les actions de rétention.

— **Analyse de Texte (NLP)** : Intégrer des données non structurées comme les commentaires des clients ou les transcriptions d'appels. Des techniques de NLP pourraient extraire des sentiments ou des sujets d'insatisfaction qui seraient des prédicteurs puissants du churn.

9.2 Améliorations de l'Application

— **Analyse "What-If"** : Ajouter une fonctionnalité de simulation dans le tableau de bord. Un manager pourrait ainsi tester des scénarios, par exemple : "Comment la probabilité de churn de ce client changerait-elle si nous lui offrions le support technique gratuitement ?".

— **Segmentation Automatique des Clients** : Implémenter un algorithme de clustering (par exemple, K-Means) pour regrouper automatiquement les clients en segments homogènes (par exemple, "clients à haut risque et à haute valeur", "nouveaux clients insatisfaits"). Des stratégies de rétention pourraient ensuite être adaptées à chaque segment.

— **Tableau de Bord de Suivi des Actions** : Ajouter une section où les équipes peuvent consigner les actions de rétention entreprises pour chaque client à risque et suivre leur efficacité.

9.3 Déploiement et MLOps

— **Pipeline de Ré-entraînement Automatisé** : Mettre en place un pipeline MLOps complet pour que le modèle soit automatiquement ré-entraîné à intervalles réguliers avec les nouvelles données, garantissant qu'il ne devienne pas obsolète.

— **Monitoring du Modèle** : Déployer des outils de monitoring pour suivre les performances du modèle en production et détecter toute dérive (concept drift ou data drift) qui pourrait dégrader sa précision.

— **Tests A/B** : Intégrer un framework de test A/B pour comparer l'efficacité de différentes stratégies de rétention ou même de différents modèles de prédiction sur des sous-groupes de clients.

9.4 Feuille de route indicative

Phase 1 (T0–T+3 mois) Industrialisation du pipeline existant, automatisation des rapports et mise en place d'un premier monitoring basique.

Phase 2 (T+3–T+6 mois) Intégration au CRM, ajout de la segmentation automatique et lancement du mode "What-If" pour les simulations.

Phase 3 (T+6–T+12 mois) Passage en production cloud, supervision temps réel et lancement d'une stratégie omnicanale de rétention pilotée par le modèle.

9.5 Indicateurs de succès

Pour évaluer l'impact des améliorations, nous proposons une batterie d'indicateurs :

— **Taux de churn observé** avant/après déploiement des actions ciblées.

— **Temps moyen de traitement** d'un dossier client à risque.

— **Adoption de l'outil** : nombre d'utilisateurs actifs hebdomadaires et taux de complétion des notes de suivi.

— **Stabilité du modèle** : suivi de l'AUC et du rappel dans le temps, contrastés avec les alertes de dérive.

9.6 Risques et plans de mitigation

— **Données obsolètes** : mettre en place une alerte automatique lorsque la date de dernière extraction dépasse un seuil défini.

— **Résistance au changement** : organiser des ateliers de co-conception et des formations personnalisées pour les équipes métier.

— **Surcharge opérationnelle** : prioriser les clients à forte valeur via un score composite (valeur vie + probabilité de churn).

Chapitre 10

Conclusion

Ce projet a permis de concevoir et de développer une solution complète et moderne pour l'un des défis les plus importants du secteur des télécommunications : la prédiction et la compréhension de la perte de clients (churn). En combinant des techniques avancées d'apprentissage automatique avec un fort accent sur l'interprétabilité et une interface utilisateur soignée, nous avons créé un outil qui va au-delà de la simple prédiction.

Le choix du modèle CatBoost s'est avéré judicieux, offrant des performances de prédiction élevées tout en gérant nativement les complexités de nos données. L'intégration de la méthodologie SHAP a été une étape clé, transformant un modèle "boîte noire" en une source d'informations transparente et exploitable. C'est cette capacité à expliquer le "pourquoi" derrière chaque prédiction qui constitue la véritable valeur ajoutée du projet, permettant aux équipes métier de passer de l'analyse à l'action avec confiance.

Le tableau de bord développé avec Streamlit matérialise cette vision en un outil concret. Il réussit à démocratiser l'accès à des analyses complexes, en fournissant une plateforme intuitive où les managers marketing, les analystes et les agents du service client peuvent collaborer pour mettre en œuvre des stratégies de rétention proactives et personnalisées.

En résumé, ce projet a démontré avec succès comment l'intelligence artificielle, lorsqu'elle est conçue de manière centrée sur l'humain, peut devenir un levier stratégique majeur. La solution développée n'est pas seulement un modèle prédictif, mais un véritable système d'aide à la décision, prêt à être intégré dans les processus métier pour réduire le churn, maximiser la satisfaction client et, in fine, améliorer la rentabilité de l'entreprise. Les nombreuses pistes d'amélioration identifiées ouvrent la voie à des développements futurs passionnants qui pourraient encore renforcer l'impact de cet outil.

Perspectives et engagements

Les prochaines étapes consisteront à déployer progressivement l'outil auprès de l'ensemble des équipes commerciales et à mesurer rigoureusement son impact sur la rétention. Un comité de gouvernance data sera instauré pour suivre la qualité des données, piloter le ré-entraînement du modèle et garantir le respect des normes éthiques. Enfin, l'ouverture d'API sécurisées permettra d'intégrer la prédiction de churn directement dans les parcours clients digitaux et de créer de nouvelles opportunités de personnalisation.

Annexe A

Glossaire

Terme	Définition
	Concepts Généraux
Churn (Perte de Clients)	Phénomène par lequel les clients d'une entreprise cessent d'utiliser ses services. Le taux de churn est un indicateur clé de la satisfaction et de la fidélité des clients.
Machine Learning (Apprentissage Automatique)	Un domaine de l'intelligence artificielle qui donne aux ordinateurs la capacité d'apprendre à partir de données sans être explicitement programmés.
IA Explicable (XAI)	Un ensemble de techniques et de méthodes qui permettent de comprendre et d'interpréter les décisions prises par les modèles d'apprentissage automatique.
Indicateur de Performance Clé (KPI)	Une valeur mesurable qui démontre l'efficacité avec laquelle une entreprise atteint ses objectifs commerciaux clés.
MLOps (Machine Learning Operations)	Une culture et une pratique qui visent à unifier le développement de systèmes d'apprentissage automatique (Dev) et le déploiement de ces systèmes (Ops).
	Colonnes de l'Ensemble de Données
`Churn`	La variable cible. Indique si le client a quitté l'entreprise (Oui/Non).
`Contract`	Le type de contrat du client (Mois par mois, Un an, Deux ans).
`Dependents`	Indique si le client a des personnes à charge (Oui/Non).
`DeviceProtection`	Indique si le client a une assurance pour ses appareils (Oui/Non).
`gender`	Le genre du client (Homme/Femme).
`InternetService`	Indique si le client a un service Internet (DSL, Fibre optique, Non).
`MonthlyCharges`	Le montant facturé au client chaque mois.
`MultipleLines`	Indique si le client a plusieurs lignes téléphoniques (Oui/Non).
`OnlineBackup`	Indique si le client a un service de sauvegarde en ligne (Oui/Non).
`OnlineSecurity`	Indique si le client a un service de sécurité en ligne (Oui/Non).
`PaperlessBilling`	Indique si le client utilise la facturation dématérialisée (Oui/Non).
`Partner`	Indique si le client a un partenaire (Oui/Non).
`PaymentMethod`	La méthode de paiement du client.
`PhoneService`	Indique si le client a un service téléphonique (Oui/Non).
`SeniorCitizen`	Indique si le client est une personne âgée (1 pour Oui, 0 pour Non).
`StreamingMovies`	Indique si le client a un service de streaming de films (Oui/Non).
`StreamingTV`	Indique si le client a un service de streaming TV (Oui/Non).
`TechSupport`	Indique si le client a un support technique (Oui/Non).
`tenure`	Le nombre de mois depuis que le client est abonné.
`TotalCharges`	Le montant total facturé au client sur toute la durée de son abonnement.
	Termes Techniques de Modélisation
CatBoost	Un algorithme d'apprentissage automatique basé sur le gradient boosting, optimisé pour la gestion des variables catégorielles.

Terme	Définition
One-Hot Encoding	Une technique de prétraitement pour convertir des variables catégorielles en un format numérique que les modèles peuvent comprendre, en créant une colonne binaire pour chaque catégorie.
Feature Scaling (Mise à l'échelle)	Le processus de normalisation de la plage des variables numériques. `StandardScaler` est une méthode qui centre les données autour de 0 avec un écart-type de 1.
Hyperparamètres	Les paramètres de configuration d'un modèle qui ne sont pas appris à partir des données, mais qui sont définis avant le processus d'entraînement (ex : taux d'apprentissage, profondeur des arbres).
Surapprentissage (Overfitting)	Un problème où un modèle d'apprentissage automatique apprend trop bien les données d'entraînement, au point de mal généraliser à de nouvelles données non vues.
Métriques d'Évaluation	
Accuracy (Exactitude)	La proportion de prédictions correctes parmi le nombre total de cas.
Precision (Précision)	Parmi toutes les prédictions positives, la proportion de celles qui étaient réellement positives.
Recall (Rappel)	Parmi tous les cas réellement positifs, la proportion qui a été correctement identifiée par le modèle.
F1-Score	La moyenne harmonique de la précision et du rappel, fournissant un score unique qui équilibre les deux.
Courbe ROC et AUC	La courbe ROC (Receiver Operating Characteristic) est un graphique qui illustre la performance d'un classifieur. L'AUC (Area Under the Curve) mesure la capacité globale du modèle à distinguer les classes. Une AUC de 1.0 est parfaite, 0.5 est aléatoire.
Autres Outils et Bibliothèques	
SHAP (SHapley Additive exPlanations)	Une méthode d'IA explicable qui attribue à chaque caractéristique une valeur d'importance pour une prédiction donnée.
Streamlit	Une bibliothèque Python open-source pour créer et partager des applications web personnalisées pour la science des données.
Docker	Une plateforme pour développer, expédier et exécuter des applications dans des conteneurs, assurant la cohérence des environnements.