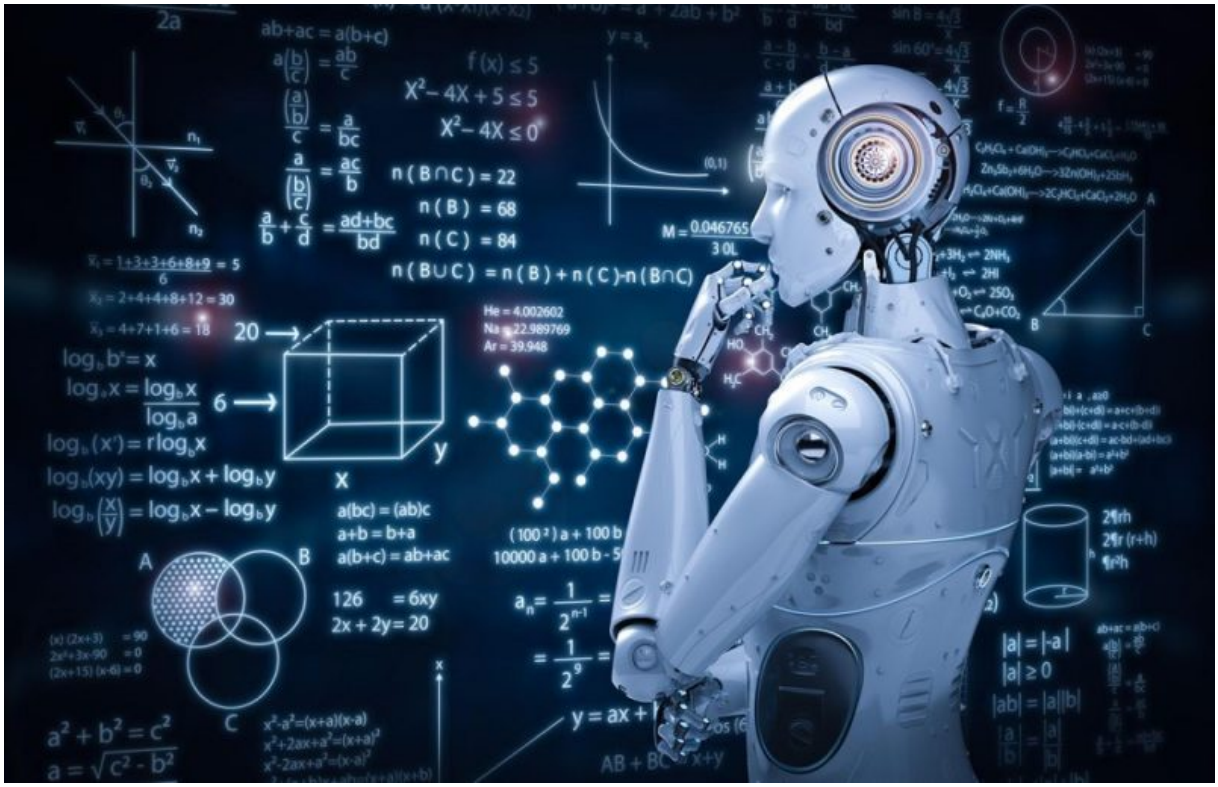
Clermont Auvergne INP ISIMA

Analyse et Prédiction du Churn Client à l'aide du Machine Learning

Rédigé par :

EBEYITE Mohamed
ELHOUARY Mohammed



Le 10 Janvier 2024

Contents

1 Introduction

Ce rapport a pour objectif principal d'étudier et de modéliser le comportement des clients d'une entreprise afin de prévoir si un client est susceptible de quitter (churn) ou de rester. Cette prédiction est essentielle pour permettre aux entreprises de développer des stratégies préventives et ainsi réduire leur taux de désertion.

Contexte

Le churn client est un problème critique dans de nombreux secteurs tels que la télécommunication, la finance et les abonnements numériques. La variable cible considérée dans ce projet est binaire :

- **Exited = 1** : le client quitte l'entreprise.
- **Exited = 0** : le client reste fidèle.

Objectifs du Projet

Les étapes clés du projet incluent :

1. Une exploration approfondie des données pour comprendre les facteurs influençant le churn.
2. La préparation des données, y compris le traitement des valeurs manquantes et l'encodage des variables catégoriques.
3. L'entraînement de plusieurs modèles de Machine Learning pour prévoir le churn avec une grande précision.
4. L'évaluation et la comparaison des performances des modèles afin de déterminer le meilleur.

Les résultats obtenus dans ce rapport mettent en avant les décisions clés prises à chaque étape et les métriques obtenues pour valider les performances des modèles proposés.

2 Exploration des Données (EDA)

L'exploration des données (EDA) est une étape cruciale pour comprendre les caractéristiques principales du dataset et détecter les relations potentielles entre les variables. Voici un résumé des analyses effectuées :

2.1 Analyse Statistique

Le dataset contient `73,955` entrées et `14` colonnes. Les variables peuvent être classées comme suit :

- **Variables continues** : `CreditScore`, `Age`, `Tenure`, `Balance`, `EstimatedSalary`.
- **Variables binaires** : `HasCrCard`, `IsActiveMember`, `Exited`.
- **Variables catégoriques** : `Geography`, `Gender`.

2.2 Visualisations

Pour mieux comprendre les relations entre les variables explicatives et la variable cible (`Exited`), plusieurs visualisations ont été générées :

2.2.1 Distribution de l'Âge selon `Exited`

La distribution de l'âge montre des différences entre les clients qui quittent l'entreprise (`Exited=1`) et ceux qui restent (`Exited=0`). Les clients plus âgés semblent avoir une tendance plus forte à quitter.

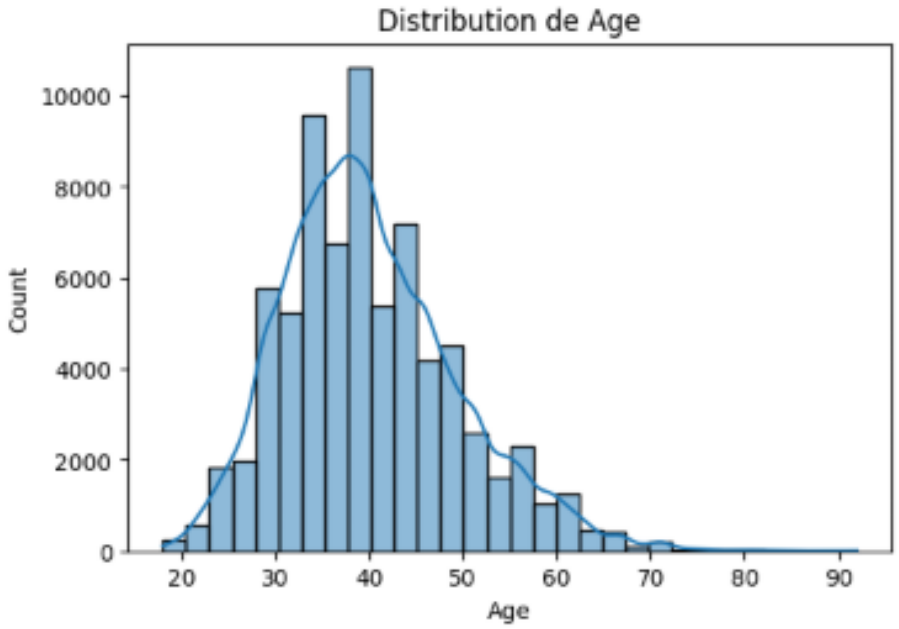


Figure 1: Distribution de l'Âge selon la variable cible `Exited`.

2.2.2 Matrice de Corrélation

La matrice de corrélation illustre les relations entre les variables continues. Par exemple, `Age` montre une corrélation modérée avec `Exited` (`0.40`), ce qui en fait une variable explicative importante.

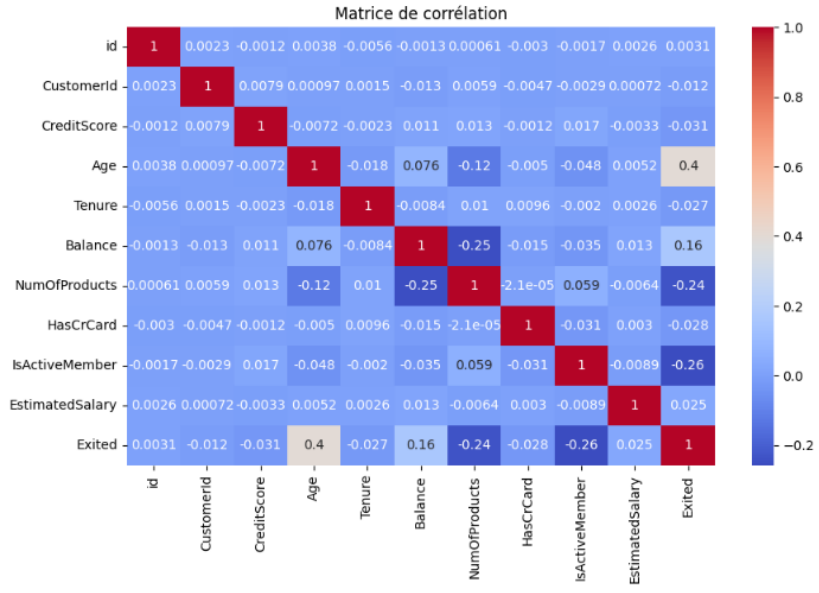


Figure 2: Matrice de corrélation des variables continues.

2.2.3 Relation entre `Geography` et `Exited`

La relation entre `Geography` et `Exited` montre que les clients situés en Allemagne ont un taux de churn significativement plus élevé par rapport à ceux en France ou en Espagne.

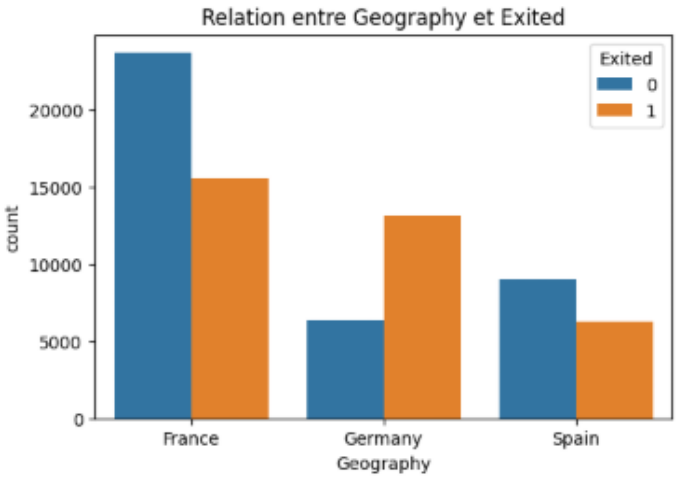


Figure 3: Relation entre `Geography` et `Exited`.

Ces visualisations mettent en évidence les principales variables explicatives et justifient leur inclusion dans les modèles de prédiction.

3 Prétraitement des Données

Le prétraitement des données est une étape cruciale pour s'assurer que les algorithmes de Machine Learning fonctionnent efficacement. Cela inclut la préparation des variables continues et catégoriques, ainsi que le traitement des classes déséquilibrées.

3.1 Standardisation

Les variables continues comme `CreditScore`, `Age`, `Tenure`, `Balance`, et `EstimatedSalary` ont été standardisées à l'aide de `StandardScaler`. Cette étape est essentielle pour :

- S'assurer que toutes les variables ont une échelle comparable.
- Réduire l'impact des variables avec des plages de valeurs très différentes sur les modèles.

La standardisation consiste à transformer les variables pour qu'elles aient une moyenne de 0 et un écart-type de 1, selon la formule :

$$x' = \frac{x - \mu}{\sigma}$$

où x est la valeur d'origine, μ la moyenne et σ l'écart-type.

3.2 Encodage des Variables Catégoriques

Les variables catégoriques (`Geography`, `Gender`) ont été transformées à l'aide de l'encodage `OneHotEncoder`, qui permet de convertir les catégories en colonnes binaires tout en évitant la redondance grâce à l'option `drop='first'`. Cela assure que les modèles interprètent correctement ces variables.

3.3 Gestion des Classes Déséquilibrées

Pour traiter le déséquilibre des classes dans la variable cible (`Exited`), nous avons utilisé la technique **SMOTE** (Synthetic Minority Oversampling Technique). Cela consiste à générer artificiellement des échantillons pour la classe minoritaire, ce qui permet aux modèles d'apprendre équitablement sur les deux classes.

- Augmente les échantillons de la classe minoritaire (`Exited = 1`).
- Réduit le risque de surapprentissage lié au déséquilibre des données.

Ces étapes garantissent que les données sont prêtes pour l'entraînement des modèles et maximisent les performances des algorithmes de classification.

4 Entraînement des Modèles

Pour prédire la variable cible (`Exited`), plusieurs algorithmes de classification ont été testés. L'objectif est d'identifier le modèle offrant les meilleures performances en termes de précision et de robustesse.

4.1 Algorithmes Utilisés

Les algorithmes suivants ont été utilisés :

- **Logistic Regression** : Un modèle linéaire de base permettant d'obtenir une première référence.
- **Decision Tree** : Un modèle basé sur des règles qui est intuitif et rapide à entraîner.
- **Random Forest** : Un ensemble d'arbres de décision pour améliorer la robustesse et réduire le surapprentissage.
- **XGBoost** : Un modèle de boosting performant, connu pour ses excellentes performances sur les données tabulaires.

4.2 Optimisation des Hyperparamètres

Pour affiner les performances des modèles, une recherche d'hyperparamètres a été réalisée à l'aide de `GridSearchCV`. Cette technique effectue une recherche exhaustive sur une grille de combinaisons de paramètres en appliquant une validation croisée. Voici les étapes clés :

- Définition d'une grille de paramètres pour chaque modèle. Par exemple, pour `Decision Tree`, les paramètres explorés incluent :
  - `max_depth` : Limite de la profondeur des arbres.
  - `criterion` : Métrique d'impureté (`gini` ou `entropy`).
  - `min_samples_split` : Nombre minimal d'échantillons requis pour effectuer une division.
- Application de la recherche sur les données d'entraînement (`X_train`, `Y_train`).
- Sélection des meilleurs hyperparamètres en fonction de la métrique de précision (`accuracy`).

4.3 Exemple de Résultat pour XGBoost

Pour **XGBoost**, les meilleurs hyperparamètres trouvés via `GridSearchCV` sont :

- `n_estimators` : 200
- `max_depth` : 3
- `learning_rate` : 0.1

Ces paramètres ont permis d'améliorer significativement les performances sur les données de test, en atteignant une précision de 81%.

4.4 Conclusion

L'optimisation des hyperparamètres a montré que `XGBoost` et `Random Forest` offrent les meilleures performances grâce à leur capacité à capturer des relations complexes dans les données.

5 Évaluation des Performances

L'évaluation des performances des modèles est essentielle pour identifier celui qui offre les meilleurs résultats. Plusieurs métriques clés ont été utilisées pour comparer les modèles.

5.1 Métriques Clés

Les métriques suivantes ont été employées pour évaluer les performances :

- **Accuracy** : Proportion des prédictions correctes parmi l'ensemble des données.
- **Précision (Precision)** : Capacité du modèle à éviter les faux positifs. Calculée comme :

$$\text{Précision} = \frac{\text{TP}}{\text{TP} + \text{FP}}$$

- **Rappel (Recall)** : Capacité du modèle à détecter correctement les vrais positifs. Calculée comme :

$$\text{Rappel} = \frac{\text{TP}}{\text{TP} + \text{FN}}$$

- **F1-Score** : Moyenne harmonique entre la précision et le rappel, utile pour les ensembles déséquilibrés.

5.2 Résultats Préliminaires

Les résultats des différents modèles, en termes d'accuracy, sont les suivants :

Modèle	**Accuracy (%)**
Logistic Regression	72
Random Forest (optimisé)	79
XGBoost (optimisé)	81

Table 1: Comparaison des performances des modèles en termes d'accuracy.

Parmi les modèles testés, **XGBoost** a démontré les meilleures performances avec une accuracy de 81%. Ce résultat est dû à sa capacité à capturer des relations complexes dans les données et à éviter le surapprentissage grâce à son mécanisme de boosting.

5.3 Conclusion

L'évaluation des modèles montre que **XGBoost** est le modèle le plus performant, suivi de près par **Random Forest**. Ces résultats guideront le choix du modèle final pour prédire efficacement le churn client.

6 Pipeline Final

Le pipeline final regroupe les étapes nécessaires pour traiter les données et entraîner un modèle performant. Voici sa structure et les résultats obtenus.

6.1 Structure

Le pipeline final comprend les étapes suivantes :

- **Prétaitement** :
  - Standardisation des variables continues (`CreditScore`, `Age`, `Balance`, etc.) pour aligner les échelles.
  - Encodage des variables catégoriques (`Geography`, `Gender`) en variables binaires avec One-Hot Encoding.
- **Équilibrage des classes** : Utilisation de SMOTE pour générer artificiellement des exemples de la classe minoritaire (`Exited = 1`).
- **Modèle final** : Entraînement de XGBoost avec les hyperparamètres optimisés (`n_estimators = 200`, `max_depth = 3`, `learning_rate = 0.1`).

6.2 Résultats

Le pipeline a été évalué sur le jeu de test, avec les métriques suivantes :

Métrique	Valeur (%)
Accuracy	81
Précision	84
Recall	78
F1-Score	81

Table 2: Performances du modèle XGBoost sur le jeu de test.

6.3 Conclusion

Le pipeline final, basé sur **XGBoost optimisé**, a démontré des performances solides. Il offre un bon équilibre entre précision et rappel, ce qui en fait un choix adapté à la problématique de la prédiction du churn.

7 Conclusion

7.1 Récapitulatif des Performances

L'analyse et l'évaluation des modèles ont permis d'identifier les algorithmes offrant les meilleures performances pour prédire le churn des clients :

- **XGBoost** s'est avéré être le modèle le plus performant, avec une accuracy de 81%, équilibrant précision (84%) et rappel (78%).
- **Random Forest**, bien que légèrement en deçà, a également montré des performances solides avec une accuracy de 79%.

Ces résultats démontrent la capacité des modèles d'ensemble comme XGBoost et Random Forest à capturer des relations complexes et à traiter efficacement les déséquilibres dans les données.

7.2 Améliorations Futures

Bien que les performances soient satisfaisantes, plusieurs améliorations pourraient être envisagées pour renforcer encore plus la prédiction du churn :

- **Intégrer de nouvelles variables explicatives** : Ajouter des caractéristiques pertinentes qui pourraient mieux expliquer le comportement des clients.
- **Affiner davantage les hyperparamètres** : Tester des plages de valeurs plus larges ou utiliser des techniques avancées comme `Bayesian Optimization` pour optimiser les modèles.
- **Explorer des techniques avancées** :
  - Le stacking, qui combine plusieurs modèles pour améliorer les performances globales.
  - Des méthodes de boosting hybrides, qui peuvent offrir un meilleur équilibre entre biais et variance.

Ces pistes ouvrent la voie à des performances encore plus robustes et à des modèles mieux adaptés aux spécificités des données étudiées.