Ministère de l’enseignement technique et de la Formation Professionnelle

Orange Digital Center

sonatel ACADEMY

Référentiel DEV-DATA

MEMOIRE DE FIN DE FORMATION

Optimisation des Campagnes Téléphoniques dans le Secteur Bancaire : Utilisation de l'Analyse Prédictive pour Prédire la Souscription aux Dépôts à Terme

Presente par :

Serigne Modou Diop

Encadre par :

Coach Mbaye

Promo 5 - Année 2023

I Avant-propos

Au cours de ma formation au sein de la Sonatel Academy, j'ai eu l'opportunité d'explorer divers aspects de la science des données, de l'analyses de données et de l'ingégneurie de données. Ce mémoire représente le fruit de mes efforts pour appliquer ces connaissances dans un contexte concret, en l'occurrence l'optimisation des campagnes téléphoniques dans le secteur bancaire.

En effet, ce travail s'inscrit dans une démarche visant à améliorer les performances des campagnes téléphoniques, en utilisant les techniques d'analyse prédictive pour prédire la souscription aux dépôts à terme. À travers ce mémoire, je souhaite partager les étapes de réflexion, d'analyse et de mise en œuvre qui ont permis de parvenir à des résultats concrets et utiles pour les professionnels du secteur.

Je tiens à exprimer ma gratitude envers toutes les personnes qui m'ont soutenu et guidé tout au long de ce projet, en particulier mes formateurs à la Sonatel Academy qui m'ont transmis leur savoir-faire et leur passion pour les données.

Je vous invite à parcourir ces pages qui représentent une étape importante de mon parcours académique et professionnel, en espérant qu'elles puissent également apporter des éléments de réflexion et d'amélioration aux lecteurs intéressés par le sujet.

II Remerciements

Je tiens à exprimer ma profonde gratitude envers toutes les personnes qui ont contribué de près ou de loin à l'élaboration de ce mémoire et à mon parcours de formation au sein de la Sonatel Academy en développement de données.

Je tiens tout particulièrement à remercier mon coach, M. Mbaye, pour sa guidance, ses conseils avisés et son soutien constant tout au long de ma formation. Son expertise et son dévouement ont été des atouts précieux dans l'acquisition de compétences solides en science des données.

Mes remerciements s'adressent également à l'ensemble des formateurs de la Sonatel Academy, dont les connaissances pointues et les méthodes pédagogiques ont largement contribué à mon apprentissage. Le partage de leur expérience a été une source d'inspiration et d'enrichissement professionnel.

Je souhaite exprimer ma reconnaissance envers mes collègues de formation, qui ont été des partenaires précieux dans cette aventure académique. Leurs échanges et leurs collaborations ont été des moments forts de partage et d'apprentissage mutuel.

Enfin, je tiens à remercier ma famille et mes amis pour leur soutien indéfectible, leur encouragement constant et leur compréhension durant cette période de formation exigeante. Mes collègues et supérieurs ont également contribué à mon épanouissement professionnel, et je leur suis reconnaissant pour leur support.

Ce mémoire est le fruit de l'engagement et du travail collectif de nombreuses personnes, et je leur suis reconnaissant pour leur précieuse contribution à mon développement professionnel.

III Les Sommaires

Table des matières

IV La liste des figures et tableaux

Index des figures

V La liste des abréviations

- **API** : Interface de Programmation Applicative
- **AUC-ROC** : Surface sous la courbe ROC
- **ML** : Machine Learning
- **SVM** : Support Vector Machines
- **VS Code** : Visual Studio Code

VI Le glossaire

Algorithme de Gradient Boosting : Méthode ensembliste qui combine plusieurs modèles faibles pour former un modèle plus robuste.

Algorithme de Naive Bayes : Classificateur basé sur le théorème de Bayes qui suppose que les caractéristiques sont indépendantes entre elles.

Algorithme de Random Forest : Algorithme d'ensemble basé sur des arbres de décision.

Algorithme de Régression Logistique : Modèle linéaire adapté à la classification binaire.

Algorithme SVM : Algorithmes de classification puissants.

Flask : Framework web léger en Python, idéal pour créer des applications web simples et rapides.

Jupyter Notebook : Application web open source permettant de créer et de partager des documents contenant du code, des équations, des visualisations et du texte explicatif.

Kaggle : Plateforme en ligne pour les scientifiques des données et les professionnels de l'apprentissage automatique, où ils peuvent trouver et publier des ensembles de données, explorer et créer des modèles dans un environnement de machine learning.

Pandas : Bibliothèque open source de Python pour la manipulation et l'analyse des données.

NumPy : Bibliothèque open source de Python pour effectuer des calculs numériques efficaces, en particulier sur des tableaux multidimensionnels.

Scikit-learn : Bibliothèque open source de Python pour l'analyse de données et l'apprentissage automatique.

Matplotlib : Bibliothèque open source de Python pour la visualisation des données.

Seaborn : Bibliothèque open source de Python pour la visualisation des données basée sur Matplotlib.

Scikit-plot : Extension de Scikit-learn offrant des outils de visualisation supplémentaires pour évaluer les performances des modèles de machine learning.

GitHub : Plateforme web populaire pour le stockage et la gestion du code source.

Hyperparamètres : Paramètres qui contrôlent le processus d'apprentissage du modèle et qui sont définis avant le démarrage de l'apprentissage.

Encodage des variables catégorielles : Processus de conversion des variables catégorielles en un format numérique pour être utilisé par les algorithmes de machine learning.

Modèle déséquilibré : Ensemble de données dans lequel les classes ne sont pas représentées de manière égale, ce qui peut poser des défis lors de l'apprentissage d'un modèle de classification.

Précision : Mesure de performance d'un modèle qui représente le nombre de prédictions correctes parmi toutes les prédictions effectuées par le modèle.

Rappel : Mesure de performance d'un modèle qui représente le nombre de prédictions correctes parmi toutes les instances réelles appartenant à la classe donnée.

Score F1 : Moyenne harmonique de la précision et du rappel, utilisée comme mesure de performance d'un modèle.

Aire sous la courbe ROC (AUC-ROC) : Mesure de performance d'un modèle qui représente la capacité du modèle à discriminer entre les classes positives et négatives.

1 INTRODUCTION

Dans un contexte où la concurrence est de plus en plus forte, les entreprises, notamment dans le secteur bancaire, cherchent à optimiser leurs campagnes marketing pour maximiser leur efficacité et atteindre leurs objectifs commerciaux. Les campagnes téléphoniques sont l'un des outils les plus utilisés pour atteindre les clients potentiels, mais leur efficacité peut être limitée par divers facteurs, tels que le ciblage inadéquat des clients ou le manque de personnalisation des offres.

Dans ce contexte, l'analyse prédictive et l'utilisation d'algorithmes de machine learning offrent des opportunités intéressantes pour améliorer la performance des campagnes téléphoniques. En prédisant avec précision la probabilité qu'un client souscrive à un produit financier, comme un dépôt à terme, les entreprises peuvent cibler de manière plus efficace les clients les plus susceptibles d'être intéressés, adaptant ainsi leurs offres et leurs messages pour augmenter le taux de conversion.

Ce mémoire vise à explorer comment l'analyse prédictive peut être utilisée pour prédire la souscription aux dépôts à terme lors de campagnes téléphoniques dans le secteur bancaire. Nous examinerons différentes techniques d'analyse de données et d'apprentissage automatique, telles que la régression logistique, les arbres de décision, les forêts aléatoires et les méthodes ensemblistes, pour construire des modèles prédictifs.

2 PRÉSENTATION

2.1 Orange Digital Center Sénégal

Orange Digital Center Sénégal est un hub d'innovation et de développement numérique basé à Dakar, au Sénégal. Il a été créé en 2018 par Orange, l'un des principaux opérateurs de télécommunications en Afrique, pour soutenir et accompagner l'écosystème numérique en Afrique de l'Ouest.

Le centre offre une variété de programmes et de services pour les entrepreneurs, les startups, les développeurs, les étudiants et les professionnels de l'industrie. Il dispose d'espaces de coworking, de laboratoires de fabrication, de salles de formation, d'ateliers et d'un programme d'incubation pour aider les startups à développer leurs idées et à se lancer sur le marché.

Le centre organise également des événements, des hackathons, des formations et des conférences pour promouvoir l'innovation et la collaboration entre les acteurs du numérique en Afrique. Il est un lieu de rencontre pour les innovateurs, les investisseurs, les experts et les décideurs de l'industrie numérique.

Orange Digital Center Sénégal fait partie du réseau Orange Digital Center, qui compte plusieurs centres d'innovation en Afrique et en Europe. Il est un symbole de l'engagement d'Orange à soutenir l'innovation et le développement numérique en Afrique, en offrant des opportunités et des outils pour aider les acteurs du numérique à réussir dans un monde de plus en plus numérique.

2.2 Sonatel Academy

La Sonatel Academy (Coding For Better Life) est la première école de codage gratuite d'Afrique de l'Ouest, ouverte en 2017 par Sonatel. Son objectif est de former des jeunes, en particulier des profils sous-représentés, notamment les femmes, aux métiers techniques du numérique pour favoriser leur insertion professionnelle. Elle complète le dispositif ministériel existant dans le domaine du numérique, soutenant ainsi la Stratégie Sénégal Numérique 2025.

Proposant des formations de qualité en développement web/mobile, en référencement digital et en développement data, la Sonatel Academy dispense des cours donnés par des experts du secteur. Elle met l'accent sur la pratique pour permettre aux apprenants de développer des compétences opérationnelles conformes aux exigences du marché.

La mission de la Sonatel Academy est de fournir une formation de qualité, de promouvoir l'innovation et d'accompagner les apprenants dans leur insertion professionnelle. Elle représente un véritable tremplin pour les jeunes talents désireux de se lancer dans une carrière dans les TIC.

2.3 Présentation du référentielle DEVELOPPEMENT DATA

Le référentiel "Développement Data" de la Sonatel Academy prépare les apprenants à devenir des spécialistes de la gestion et de l'exploitation des données, depuis l'analyse des besoins jusqu'à la visualisation des données. Ces développeurs sont chargés de concevoir et d'utiliser des bases de données, en gérant tout le cycle de vie des données, de la collecte à la livraison de données exploitables. Ils sont capables de traiter n'importe quel format de données, de les stocker, de les interroger et de les présenter de manière visuelle ou adaptée à un usage tiers. Ils peuvent également automatiser des processus d'acquisition, d'importation, d'extraction et de visualisation des données, tout en garantissant la qualité, l'intégrité et la cohérence des données.

La formation "Développement Data" de la Sonatel Academy est un programme intensif destiné à former les futurs professionnels du domaine des données, qu'ils aspirent à devenir des développeurs de logiciels axés sur les données, des data scientists, des data analysts ou des data engineers. Ce programme complet offre une immersion totale dans le monde des données, couvrant un large éventail de compétences et de connaissances essentielles pour réussir dans le domaine.

L'objectif principal de cette formation est de fournir aux apprenants les compétences nécessaires pour concevoir, développer et déployer des solutions innovantes basées sur l'exploitation et l'analyse des données. Pour ce faire, les apprenants sont formés sur divers aspects techniques, tels que les langages de programmation (notamment Python et R), les bases de données, les

technologies de stockage et d'analyse de données, ainsi que les méthodologies de développement de modèles de machine learning.

Encadrés par des formateurs expérimentés et passionnés par le domaine des données, les apprenants bénéficient d'un apprentissage interactif et personnalisé, mettant la pratique au centre de l'enseignement. En travaillant sur des projets concrets tout au long de la formation, les apprenants acquièrent une expérience pratique précieuse et peuvent consolider leurs connaissances théoriques en les appliquant dans des contextes réels.

En conclusion, la formation "Développement Data" de la Sonatel Academy représente une opportunité unique pour les individus désireux de se lancer dans une carrière axée sur les données, ainsi que pour les entreprises cherchant à renforcer leurs équipes avec des professionnels hautement qualifiés et compétents dans le domaine des données.

3 METHODOLOGIE DE GESTION DE PROJET

3.1 Description des étapes de réalisation du projet

La réalisation de ce projet repose sur une méthodologie structurée visant à garantir une gestion efficace des différentes étapes de développement et d'évaluation des modèles prédictifs. Cette méthodologie détaillée ci-dessous permettra d'assurer la cohérence, la qualité et la rigueur tout au long du processus.

3.1.a Collecte des données

- Les données sont extraites de la plateforme Kaggle, une communauté en ligne dédiée à l'apprentissage automatique et à l'analyse de données. Cette étape a impliqué la recherche, la sélection et le téléchargement du jeu de données approprié pour notre étude.

3.1.b Exploration des données

- Une analyse exploratoire approfondie a été menée pour comprendre la structure, la qualité et les caractéristiques des données. Cette phase a permis d'identifier d'éventuelles anomalies, tendances et relations entre les variables.

3.1.c Prétraitement des données

- Les données ont été nettoyées et préparées pour l'analyse en traitant les valeurs manquantes, en détectant et en corrigeant les valeurs aberrantes, ainsi qu'en normalisant les caractéristiques lorsque cela était nécessaire. De plus, les variables catégorielles ont été encodées pour être utilisées dans les modèles.

3.1.d Ingénierie des fonctionnalités

- De nouvelles caractéristiques ont été créées à partir des données existantes pour améliorer la capacité des modèles à capturer les informations pertinentes. Cette étape a impliqué la transformation et la combinaison de variables pour créer de nouvelles informations significatives..

3.1.e Division des données

- Les données traitées ont été divisées en deux ensembles distincts : un ensemble d'entraînement utilisé pour former les modèles et un ensemble de test utilisé pour évaluer les performances des modèles entraînés.

3.1.f Entraînement des modèles

- Différents algorithmes de classification, tels que la régression logistique, les arbres de décision et les méthodes ensemblistes, ont été utilisés pour entraîner des modèles sur les données d'entraînement.

3.1.g Évaluation des modèles

- Les performances des modèles ont été évaluées en utilisant plusieurs métriques telles que l'accuracy, la précision, le rappel, le F1-score et l'AUC-ROC sur les données de test. Cette évaluation a permis de comparer et de sélectionner les meilleurs modèles pour la prédiction de la souscription aux dépôts à terme.

3.1.h Optimisation des modèles

- Les hyperparamètres des modèles ont été ajustés en utilisant des techniques telles que la validation croisée et le GridSearchCV pour améliorer les performances prédictives des modèles sélectionnés.

3.1.i Interprétation des résultats

- Les résultats des modèles ont été analysés et interprétés pour tirer des conclusions sur leur efficacité et leur pertinence par rapport à l'objectif de prédire la souscription aux dépôts à terme.

3.2 Méthodes utilisées pour chaque étape

Chaque étape de la méthodologie a été réalisée en utilisant des outils et des techniques appropriés. Les fonctionnalités spécifiques de ces outils ont été exploitées pour effectuer les tâches nécessaires à chaque étape de la réalisation du projet.

- Collecte des données:

  - Utilisation de bibliothèques Python telles que Pandas pour télécharger et manipuler les données depuis Kaggle.

- Exploration des données:
  - Utilisation de Pandas, Matplotlib et Seaborn pour visualiser les distributions, les corrélations et détecter les valeurs aberrantes.
- Prétraitement des données:
  - Utilisation de scikit-learn pour nettoyer les données, remplir les valeurs manquantes, encoder les variables catégorielles et normaliser les caractéristiques.
- Ingénierie des fonctionnalités:
  - Utilisation de Pandas et de méthodes de transformation de données de scikit-learn pour créer de nouvelles caractéristiques à partir des variables existantes.
- Division des données:
  - Utilisation de la fonction train_test_split de scikit-learn pour diviser les données en ensembles d'entraînement et de test.
- Entraînement des modèles:
  - Utilisation des implémentations d'algorithmes de classification de scikit-learn pour entraîner différents modèles sur les données d'entraînement.
- Évaluation des modèles:
  - Utilisation des métriques fournies par scikit-learn pour évaluer les performances des modèles sur les données de test.
- Optimisation des modèles:
  - Tester différentes combinaisons d'hyperparamètres et à sélectionner celle qui donne les meilleurs résultats en termes de métriques de performance telles que l'accuracy, le rappel, la précision, le F1-score, etc.
- Interprétation des résultats:
  - Utilisation des techniques d'interprétabilité des modèles, telles que les diagrammes d'importance des caractéristiques, les graphiques de partial dependence plots (PDP) ou les Shapley values, pour expliquer les prédictions de vos modèles de manière compréhensible.

4 OUTILS ET TECHNOLOGIES UTILISÉS

4.1 Langage de programmation

Python a été le langage de programmation principal utilisé pour ce projet en raison de sa polyvalence et de sa richesse en bibliothèques adaptées à l'apprentissage automatique et à l'analyse de données.

Figure 1: python

4.1.a Pandas :

Pandas est une bibliothèque open source de Python qui offre des structures de données et des outils de manipulation de données puissants et flexibles. Cette bibliothèque a été utilisée pour la manipulation et l'analyse des données, grâce à ses fonctionnalités de manipulation de données rapides et efficaces.

Figure 2: pandas

4.1.b NumPy :

NumPy est une bibliothèque open source de Python qui offre des fonctions pour effectuer des calculs numériques efficaces, en particulier sur des tableaux multidimensionnels. Dans ce projet, NumPy a été utilisé pour traiter les données dans un format adapté aux modèles d'apprentissage automatique.

Figure 3: NumPy

4.1.c Scikit-learn :

Scikit-learn est une bibliothèque open source de Python qui offre des outils simples et efficaces pour l'analyse de données et l'apprentissage automatique. Scikit-learn a été utilisé pour implémenter les différents algorithmes de machine learning, permettant ainsi de former, évaluer et optimiser les modèles de prédiction.

Figure 4: Scikit learn

4.1.d Matplotlib et Seaborn :

Matplotlib et Seaborn sont des bibliothèques open source de Python utilisées pour la visualisation des données. Ces bibliothèques ont permis de créer des graphiques et des diagrammes pour mieux comprendre les caractéristiques des données et les résultats des modèles.

Figure 5: matplotlib

Figure 6: Seaborn

4.1.e Scikit-plot :

Scikit-plot est une extension de Scikit-learn qui offre des outils de visualisation supplémentaires pour évaluer les performances des modèles de machine learning. Dans ce projet, Scikit-plot a été utilisé pour créer des courbes ROC et des matrices de confusion afin d'évaluer les performances des modèles.

4.2 Environnement de développement

Visual Studio Code (VS Code) a été utilisé comme logiciel permettant de développer notre code. VS Code est un éditeur de code source léger mais puissant qui offre une interface utilisateur intuitive, des fonctionnalités d'édition avancées et une intégration transparente avec des outils de développement populaires.

L'extension Jupyter Notebook de VS Code a également été utilisée comme environnement de développement interactif pour écrire, exécuter et visualiser le code Python. Jupyter Notebook est une application web open source qui permet de créer et de partager des documents contenant du code, des équations, des visualisations et du texte explicatif.

Figure 7: Logo VS Code

Figure 8: Jupyter Notebook

4.3 Outils de gestion des ressources de code

GitHub est une plateforme web populaire pour le stockage et la gestion du code source. Ses caractéristiques en font un outil idéal pour le développement collaboratif de logiciels, y compris le contrôle des versions, le suivi des problèmes et la gestion de projet.

GitHub a été utilisé comme plateforme de gestion de versions pour le code source du projet. Cela a permis de suivre les modifications apportées au code, de collaborer avec d'autres personnes et de conserver une trace de l'historique du projet.

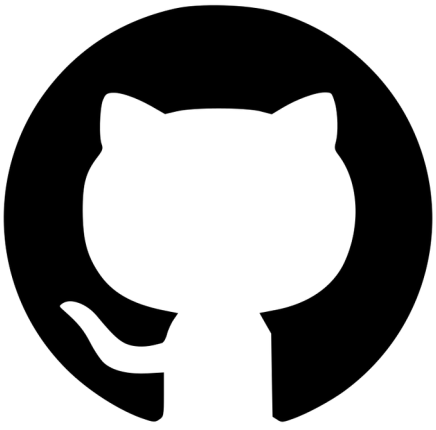

Figure 9: Github

4.4 Outil de déploiment

Flask est un framework web léger en Python, idéal pour créer des applications web simples et rapides. L'utilisation de Flask nous permet de déployer notre modèle de manière accessible à distance, ce qui est essentiel pour son utilisation dans des applications ou des systèmes externes.

Figure 10: Logo de Flask

5 Analyse et conceptions

5.1 Analyse des besoins

Pour notre projet de prédiction de la souscription aux dépôts à terme, nous avons utilisé un jeu de données provenant de Kaggle. Ce jeu de données contient des informations sur les clients, des données relatives à leurs transactions bancaires précédentes ainsi que des informations sur l'économie et le contexte financier au moment des contacts avec les clients . Les informations disponibles dans le dataset sont les suivantes :

5.1.a Données clients bancaires

- **Age** : L'âge du client. Cette information est importante car elle peut indiquer la maturité financière et les préférences d'investissement du client. Les personnes plus âgées peuvent être plus enclines à souscrire à un dépôt à terme pour sécuriser leurs économies.
- **Job** : La profession du client. La profession peut influencer la décision d'épargne du client. Par exemple, les personnes ayant des emplois stables et bien rémunérés peuvent être plus enclines à épargner.
- **Marital** : La situation familiale du client (célibataire, marié, divorcé, etc.). La situation familiale peut également influencer la décision d'épargne. Par exemple, les personnes mariées avec des enfants peuvent être plus enclines à épargner pour l'avenir de leur famille.

- **Education** : Le niveau d'éducation du client. Le niveau d'éducation peut être un indicateur de la stabilité financière et des connaissances en matière d'investissement du client.
- **Default** : Indique si le client a déjà fait défaut sur un prêt ou non. Les personnes ayant déjà fait défaut sur un prêt peuvent être moins enclines à souscrire à un dépôt à terme en raison de leur historique de crédit.
- **Housing** : Indique si le client a un prêt immobilier en cours ou non. Les personnes ayant déjà un prêt immobilier en cours peuvent être moins enclines à souscrire à un dépôt à terme en raison de leurs obligations financières existantes.
- **Loan** : Indique si le client a un prêt personnel en cours ou non. De même, les personnes ayant déjà un prêt personnel en cours peuvent être moins enclines à souscrire à un dépôt à terme en raison de leurs obligations financières existantes.

5.1.b Dernier contact de la campagne en cours

- **Contact** : Le type de contact établi avec le client (téléphone, cellulaire, etc.). Le type de contact peut influencer la capacité à atteindre le client pour proposer des produits financiers.
- **Month** : Le mois où le dernier contact a été établi avec le client. De même, le mois peut être un indicateur de saisonnalité dans les décisions d'épargne.
- **Day_of_week** : Le jour de la semaine où le dernier contact a été établi avec le client. Cette information peut être utilisée pour analyser les tendances de contact en fonction des jours de la semaine.
- **Duration** : La durée du dernier contact en secondes. La durée du contact peut être un indicateur de l'intérêt du client pour le produit proposé.
- **Campaign** : Le nombre de contacts effectués lors de cette campagne. Le nombre de contacts peut également être un indicateur de l'intérêt du client pour le produit proposé.
- **Pdays** : Le nombre de jours écoulés depuis le dernier contact avec le client. Cette information peut être utilisée pour analyser la fréquence des contacts avec le client et son impact sur la décision d'épargne.
- **Previous** : Le nombre de contacts effectués avant cette campagne. De même, le nombre de contacts précédents peut être un indicateur de l'intérêt du client pour le produit proposé.
- **Poutcome** : Le résultat de la campagne marketing précédente. Le résultat de la campagne précédente peut être un indicateur de l'efficacité des campagnes précédentes sur ce client spécifique.

5.1.c Attributs du contexte social et économique

- Emp_var_rate : Taux de variation de l'emploi - indicateur trimestriel.
- Cons_price_idx : Indice des prix à la consommation - indicateur mensuel.

- Cons_conf_idx : Indice de confiance des consommateurs - indicateur mensuel.
- Euribor3m : Taux Euribor à 3 mois - indicateur quotidien.
- Nr_employed : Nombre d'employés - indicateur trimestriel.

5.1.d Variable de sortie (cible souhaitée) :

- **y** : Indique si le client a souscrit à un dépôt à terme ou non. C'est notre variable cible, que nous chercherons à prédire à partir des autres variables.

Les attributs du contexte social et économique fournissent des informations sur l'économie et le contexte financier au moment des contacts avec les clients. Elles peuvent être importantes pour comprendre le comportement des clients. En analysant ces informations, nous pourrons mieux comprendre le comportement des clients et utiliser ces connaissances pour développer un modèle prédictif précis.

5.2 Préparation des données

La qualité des données est fondamentale pour la réussite d'un modèle d'apprentissage. Ainsi, la préparation des données est une étape essentielle qui vise à nettoyer, normaliser et enrichir les données brutes afin de les rendre exploitables par les algorithmes. Cette phase garantit la cohérence et la pertinence des données pour une analyse efficace.

Nous avons initié cette étape par une exploration minutieuse des données pour en comprendre la structure, la distribution et les relations entre les variables. À l'aide de visualisations, nous avons identifié les tendances ce qui a permis d'acquérir une meilleure compréhension du jeu de données.

Ensuite, le prétraitement des données a été effectué pour les nettoyer, les transformer et les préparer en vue de l'analyse. Ce processus a inclus la suppression de la variable « duree_appel » , la normalisation des variables numériques pour les mettre à la même échelle, et l'encodage des variables catégorielles pour les rendre utilisables dans les modèles d'apprentissage automatique. Ces transformations ont abouti à des données de haute qualité, prêtes à être utilisées dans la modélisation.

5.2.a Exploration des données

i Exploration de la valeur à prédire

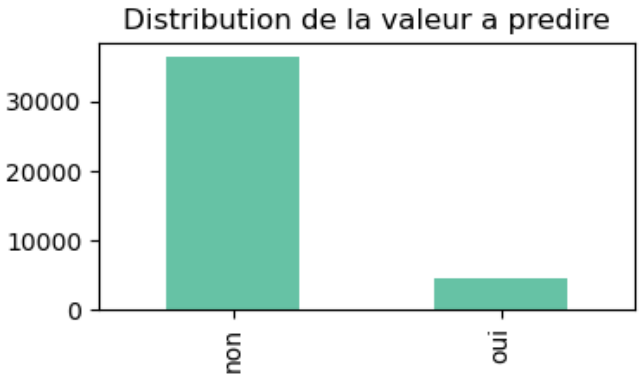

Figure 11: Distribution de la valeur à predire

- La variable y est une variable binaire qui prend les valeurs "non" et "oui". Elle indique si le client a souscrit à un prêt ou non.
- La distribution de la variable y est déséquilibrée. Il y a beaucoup plus de clients qui n'ont pas souscrit à un prêt que de clients qui ont souscrit à un prêt.
- Cette déséquilibre peut poser problème lors de l'apprentissage d'un modèle de classification. Il est important de prendre en compte ce déséquilibre lors de l'évaluation du modèle.

ii Exploration des corrélations entre les variables

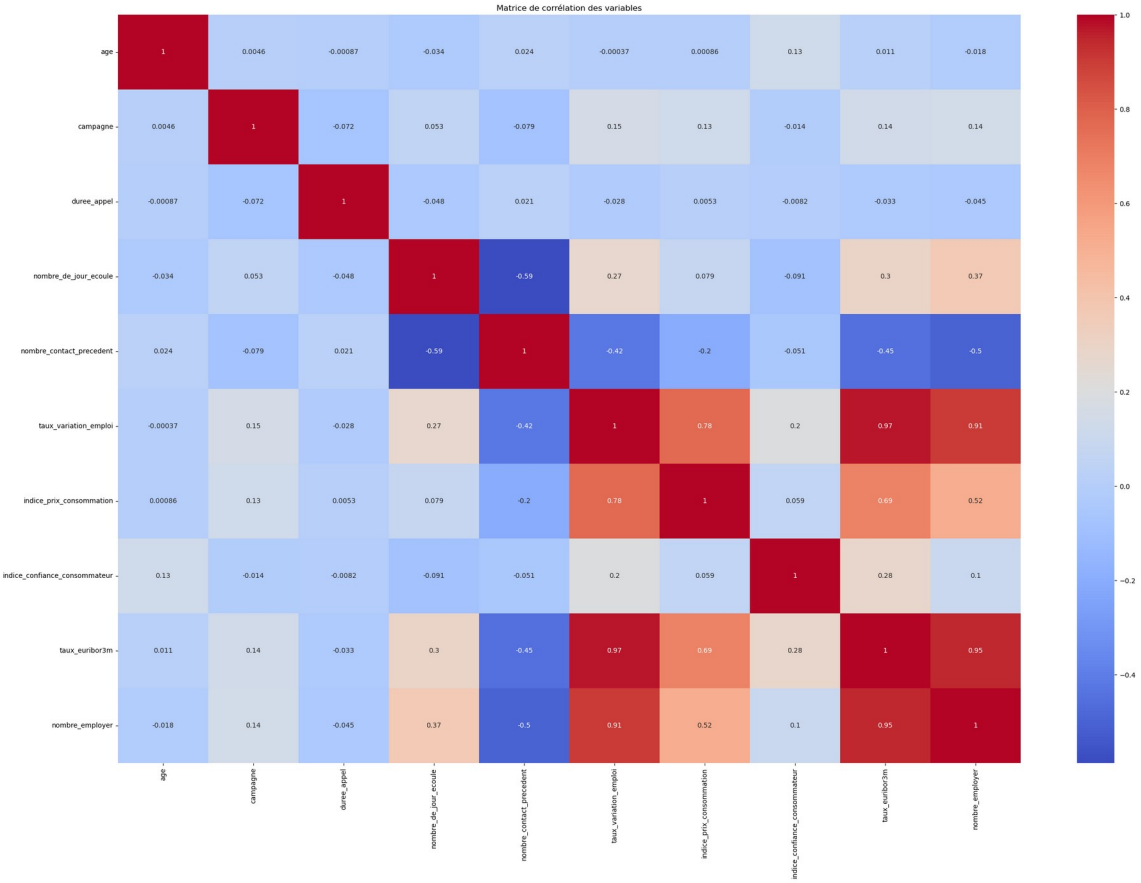

Figure 12: Matrice de corrélation

L'analyse des corrélations entre les variables de notre ensemble de données a révélé des liens significatifs entre certaines variables indicatrices et d'autres caractéristiques. Plus précisément, les variables telles que 'taux_variation_emploi', 'indice_prix_consommation', 'indice_confiance_consommateur', 'taux_euribor3m', et 'nombre_employer' présentent des corrélations importantes entre eux. Étant des indicateurs spécifiques à un pays, ces variables pourraient introduire un biais géographique dans nos modèles de prédiction.

Afin de garantir l'objectivité et la généralisation des modèles, nous avons pris la décision de supprimer ces variables fortement corrélées. Cette démarche vise à améliorer la robustesse de nos analyses et à limiter les biais potentiels introduits par des facteurs non pertinents pour notre problématique. En éliminant ces variables, nous nous assurons que nos modèles se concentrent sur des caractéristiques plus générales et prédictives, favorisant ainsi des résultats plus fiables et applicables à un contexte plus large.

Cette décision s'inscrit dans une démarche rigoureuse visant à garantir la qualité et la pertinence de nos analyses, tout en soulignant l'importance de la sélection des caractéristiques dans le processus de modélisation en science des données.

iii Exploration de la distribution des données catégorielles par souscription

- Metier (type d'emploi) :

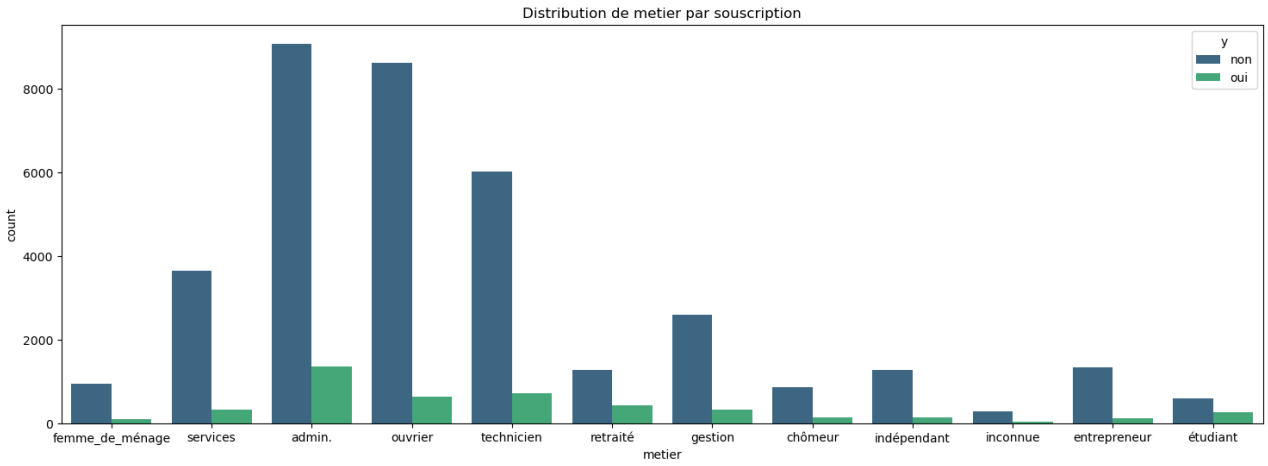

Figure 13: Distribution du métier par souscription

  - La plupart des clients semblent être dans les catégories « admin », « ouvrier » et « technicien ».
  - Les étudiants ont une tendance plus forte à souscrire à un dépôt à terme par rapport à d'autres catégories.

- Situation matrimoniale (état civil) :

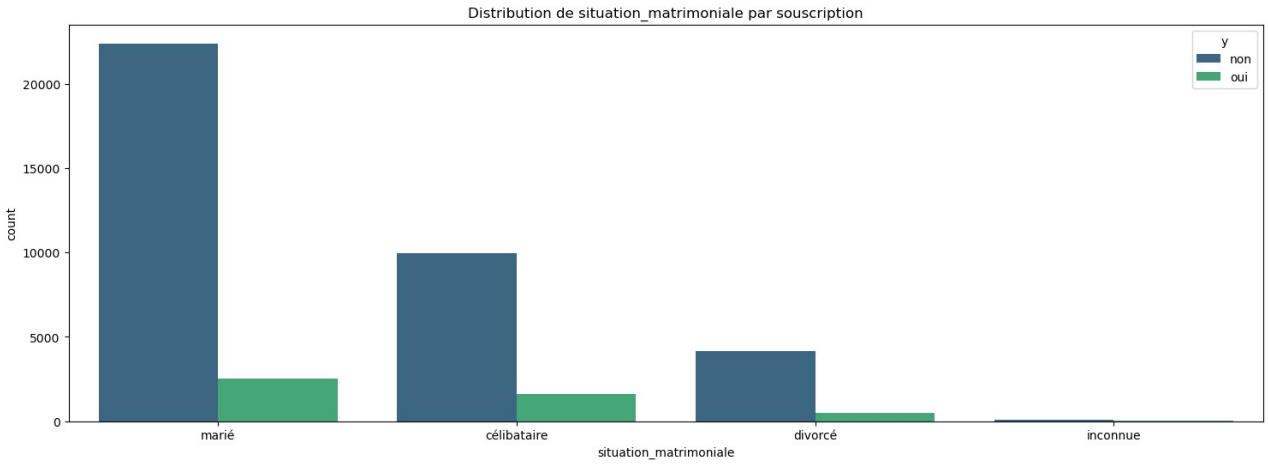

Figure 14: Distribution de la situation matrimoniale par souscription

  - La majorité des clients sont mariés, suivis par des célibataires.

- Les célibataires ont une légère tendance à souscrire davantage.

- Education (niveau d'éducation) :

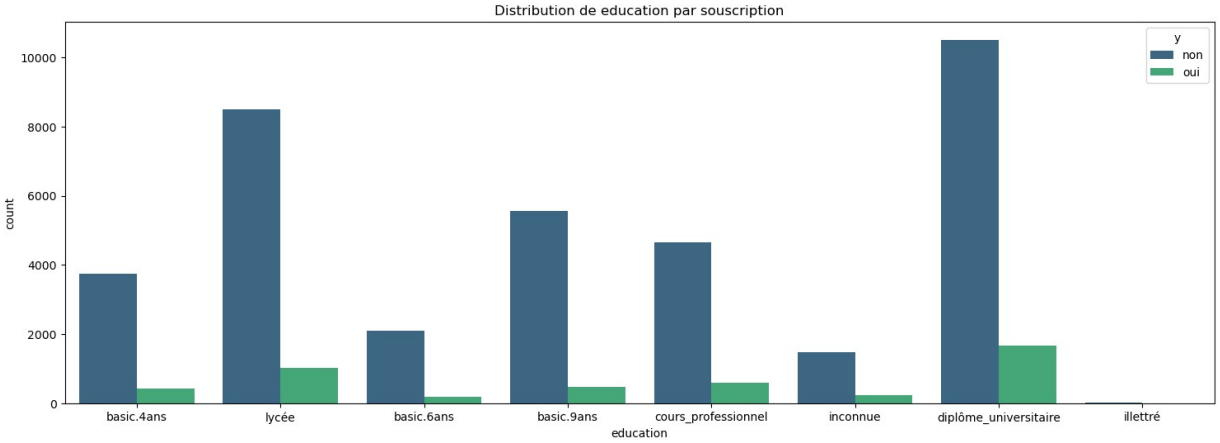

Figure 15: Distribution de la variable éducation par souscription

- La plupart des clients ont un niveau d'éducation de base ou universitaire.
- Les clients avec un diplôme universitaire semblent avoir une tendance plus forte à souscrire.

- Defaut_credit (crédit en défaut) :

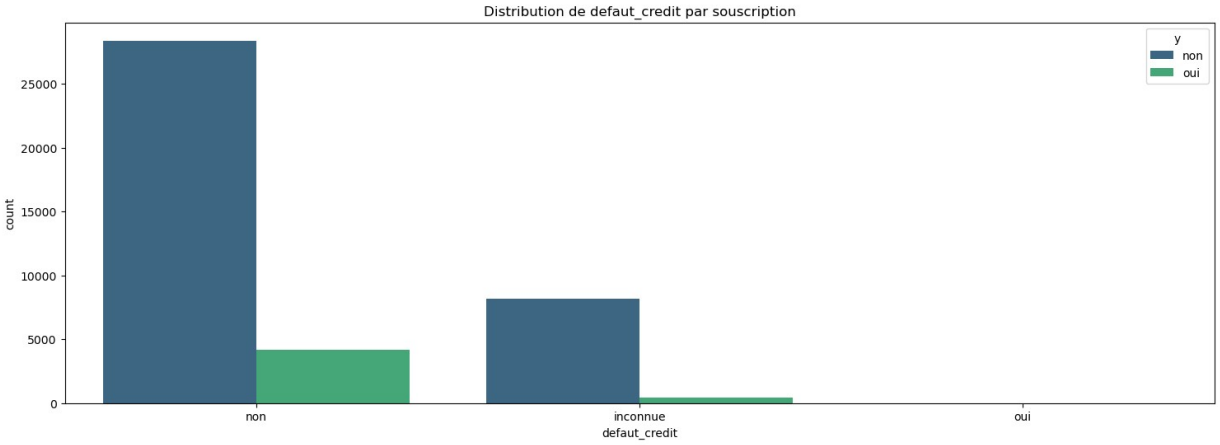

Figure 16: Distribution du défaut de crédit par souscription

- La plupart des clients n'ont pas de défaut de crédit.
- Les clients sans défaut de crédit ont une tendance plus forte à souscrire.

- Logement (prêt logement) :

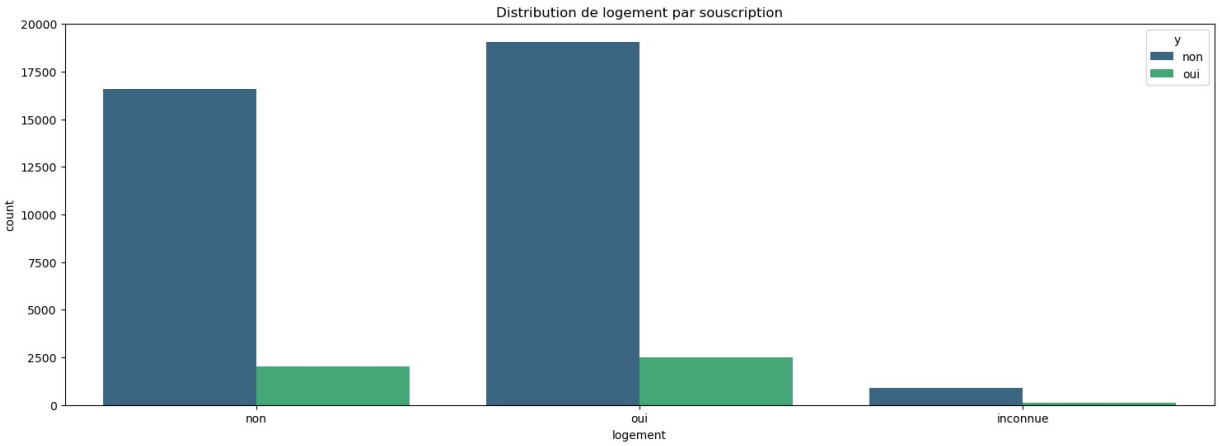

Figure 17: Distribution du logement par souscription

  - La plupart des clients disposent d'un prêt logement.
  - Les clients avec prêt logement ont une tendance plus forte à souscrire.

- Pret (prêt personnel) :

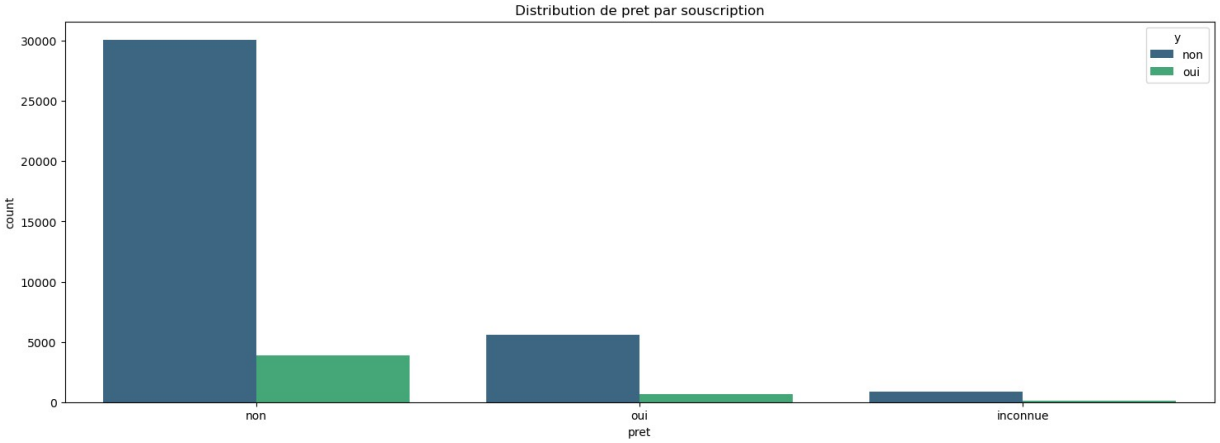

Figure 18: Distribution du pret personnel par souscription

  - La plupart des clients n'ont pas de prêt personnel.
  - Les clients sans prêt personnel ont une tendance plus forte à souscrire.

- Contact (type de communication) :

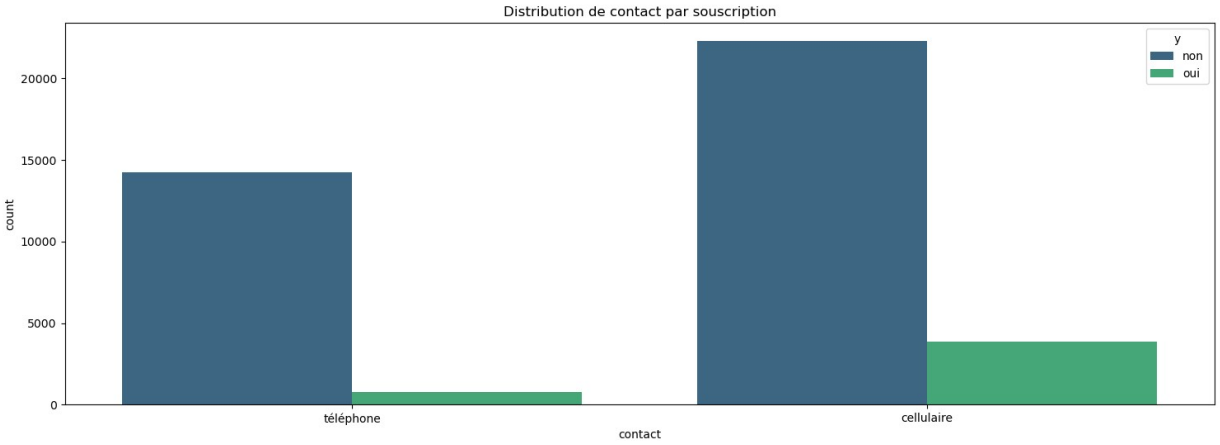

Figure 19: Distribution de contact par souscription

- La communication cellulaire semble plus fréquente que la communication téléphonique.
- Les clients contactés par téléphone semblent avoir une légère tendance à souscrire davantage.

- Mois (mois du dernier contact) :

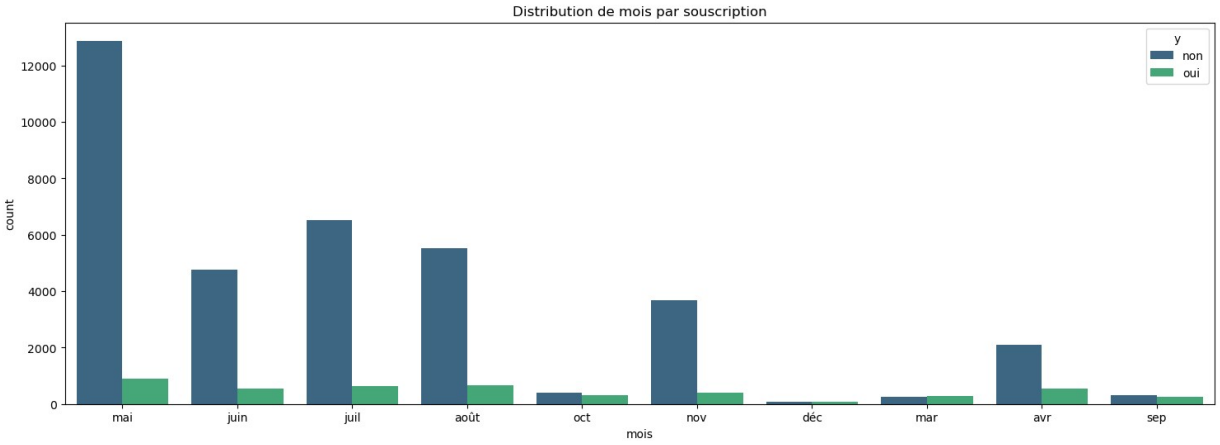

Figure 20: Distribution du mois par souscription

- Les contacts semblent répartis sur plusieurs mois.
- Il n'y a pas de tendance claire en fonction du mois.

- Jour_de_semaine (jour de la semaine du dernier contact) :

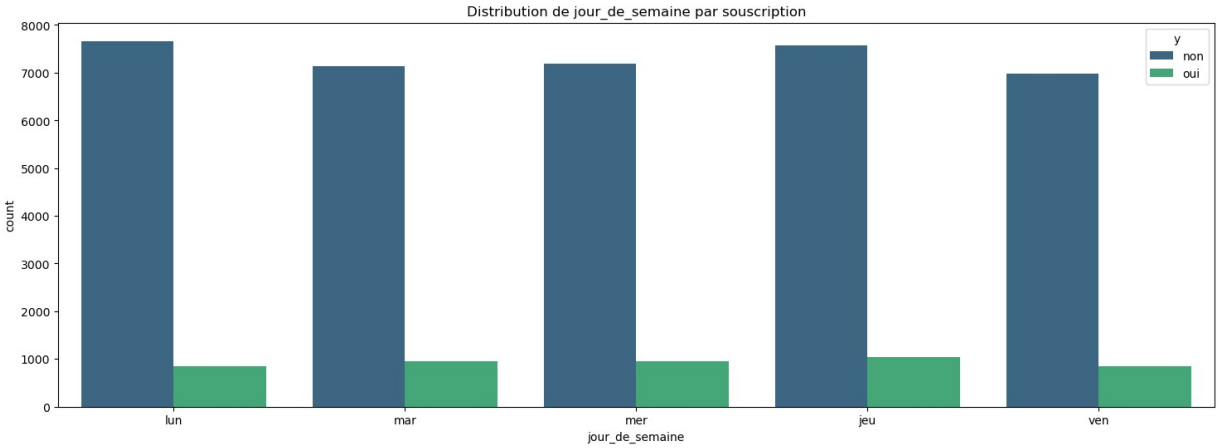

Figure 21: Distribution du jour de la semaine par souscription

- Les contacts sont également répartis sur tous les jours de la semaine.
- Il n'y a pas de tendance claire en fonction du jour de la semaine.

- Resultat_campagne_precendente (résultat de la campagne marketing précédente) :

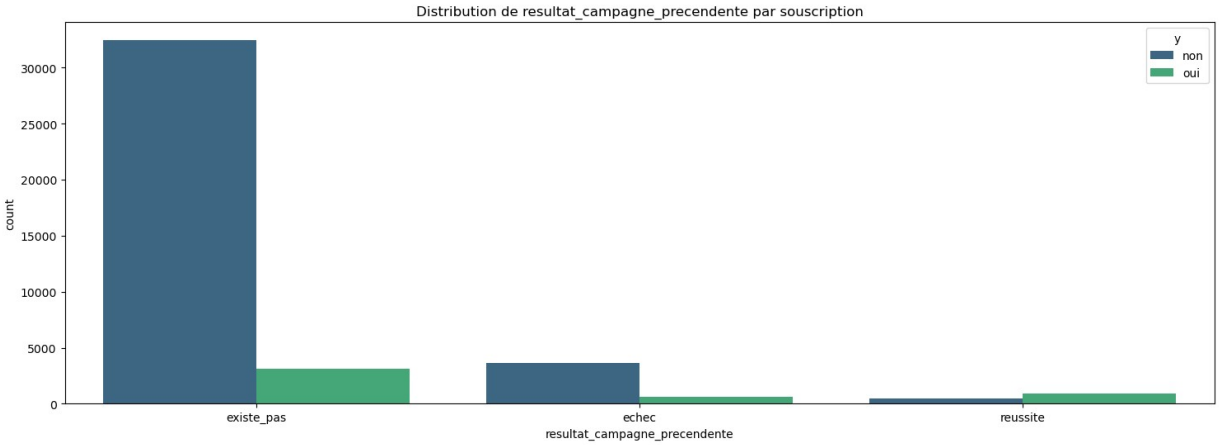

Figure 22: Distribution du résultat de la campagne précedente par souscription

- La plupart des clients n'ont pas de résultat de campagne précédente.
- Les clients avec un résultat de campagne précédente "succès" ont une tendance plus forte à souscrire.

- y (souscription à un dépôt à terme) :
  - Il est essentiel de vérifier la distribution des classes dans la variable cible.

- La classe "no" semble être plus fréquente, ce qui indique un déséquilibre dans les données.

iv Exploration des données numériques

Pour chaque variable numérique, nous avons comparé la distribution entre les clients qui ont souscrit à un dépôt à terme ("oui") et ceux qui n'ont pas souscrit ("non").

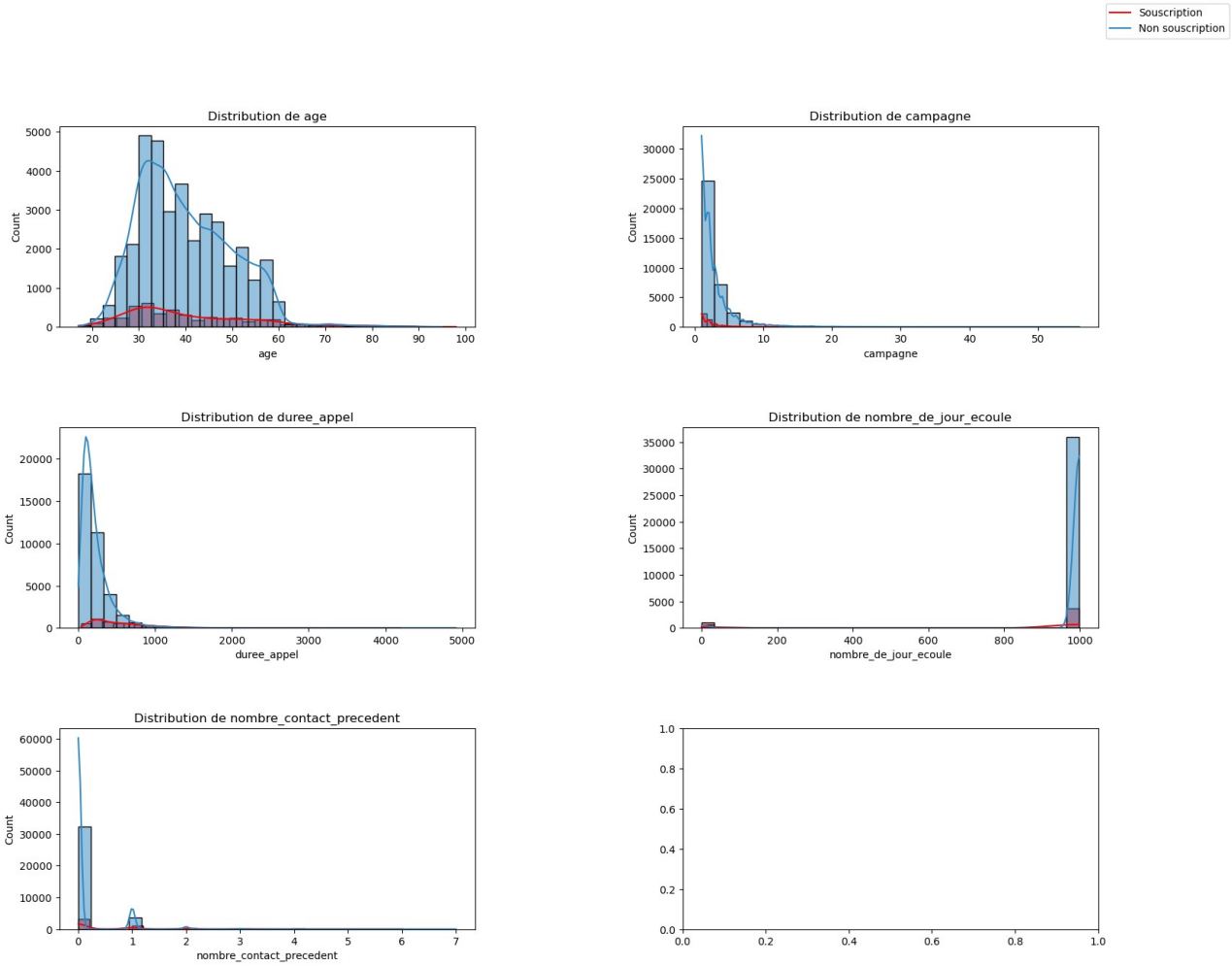

Figure 23: Distribution des variables numeriques par souscription

- age
  - Les clients qui souscrivent à un dépôt à terme ont une répartition plus large en termes d'âge par rapport à ceux qui ne souscrivent pas.

- On observe une concentration plus importante de clients plus jeunes parmi ceux qui souscrivent.

- duree_appel (Durée du dernier contact en secondes) :
  - La durée du dernier contact semble significativement plus longue pour les clients qui souscrivent à un dépôt à terme.

- campagne (Nombre de contacts effectués lors de cette campagne) :
  - Les clients qui souscrivent à un dépôt à terme ont tendance à avoir été contactés un nombre plus faible de fois au cours de cette campagne.

- nombre_de_jour_ecoule (Nombre de jours écoulés après le dernier contact lors d'une campagne précédente) :
  - La majorité des clients, qu'ils souscrivent ou non, n'ont pas été contactés lors d'une campagne précédente (pdays = 999).
  - Cependant, parmi ceux qui ont été contactés précédemment, il semble y avoir une distribution similaire entre les deux groupes.

- nombre_contact_precedent (Nombre de contacts effectués avant cette campagne) :
  - Les clients qui souscrivent à un dépôt à terme ont tendance à avoir un nombre plus élevé de contacts précédents.

Conclusions Générales :

- Le nombre de contacts précédents et le taux de variation de l'emploi semblent avoir une influence sur la souscription.

5.2.b Prétraitement des données

- Suppression des variables indicatrices

```
numerical_columns = ['taux_variation_emploi', 'indice_prix_consommation',
                     'indice_confiance_consommateur', 'taux_euribor3m',
                     'nombre_employer']

data.drop(numerical_columns, axis=1, inplace=True)
```
✓ 0.0s

Figure 24: Suppression de variable indicatrices

- Transformation de la variable à prédire en variable binaire

```
from sklearn.preprocessing import LabelEncoder

label_encoder = LabelEncoder()
data['y'] = label_encoder.fit_transform(data['y'])
```
✓ 0.0s

Figure 25: Transformation de la variable à predir en varaible binaire

- Encodage des variables catégorielles

```
data_one_hot = pd.get_dummies(data, dtype=int)
data_one_hot
```
Python

	age	y	duree_appel	campagne	nombre_de_jour_ecoule	nombre_contact_precedent	education_basic.4ans	education_basic.6ans	education_basic.9ans	education_cours_professionnel
0	56	0	261	1	999	0	1	0	0	0
1	57	0	149	1	999	0	0	0	0	0
2	37	0	226	1	999	0	0	0	0	0
3	40	0	151	1	999	0	0	1	0	0
4	56	0	307	1	999	0	0	0	0	0
...	...	...	...	...	...	...	...	...	...	...
41183	73	1	334	1	999	0	0	0	0	1
41184	46	0	383	1	999	0	0	0	0	1
41185	56	0	189	2	999	0	0	0	0	0
41186	44	1	442	1	999	0	0	0	0	1
41187	74	0	239	3	999	1	0	0	0	1

41188 rows × 59 columns

Figure 26: Encodage des variables catégorielles

- Normalisation et standardisation des variables numériques

```
from sklearn.preprocessing import StandardScaler

# Sélection des variables numériques à standardiser
numerical_columns = ['age', 'campagne', 'nombre_de_jour_ecoule', 'nombre_contact_precedent', 'duree_appel']

# Création d'un objet StandardScaler
scaler = StandardScaler()

data_scaled = scaler.fit_transform(data_one_hot[numerical_columns])

data_scaled
```

```
array([[ 1.53303429, -0.56592197,  0.1954139 , -0.34949428,  0.01047142],
       [ 1.62899323, -0.56592197,  0.1954139 , -0.34949428, -0.42150051],
       [-0.29018564, -0.56592197,  0.1954139 , -0.34949428, -0.12451981],
       ...,
       [ 1.53303429, -0.20490853,  0.1954139 , -0.34949428, -0.26722482],
       [ 0.38152696, -0.56592197,  0.1954139 , -0.34949428,  0.70856893],
       [ 3.26029527,  0.15610492,  0.1954139 ,  1.67113606, -0.07438021]])
```

Figure 27: Normalisation et standardisation des variables numérique

- Fusionner les deux dataframes (Dataframe finale)

```
merged_data
```

0.0s Python

ir_de_semaine_mar	jour_de_semaine_mer	jour_de_semaine_ven	resultat_campagne_precendente_existe_pas	resultat_campagne_precendente_reussite	age	campagne	nombre_de_jour_ecoule	nombre_contact_precedent
0	0	0	1	0	1.533034	-0.565922	0.195414	-0.349494
0	0	0	1	0	1.628993	-0.565922	0.195414	-0.349494
0	0	0	1	0	-0.290186	-0.565922	0.195414	-0.349494
0	0	0	1	0	-0.002309	-0.565922	0.195414	-0.349494
0	0	0	1	0	1.533034	-0.565922	0.195414	-0.349494
...	...	...	...	...	...	...	...	...
0	0	1	1	0	3.164336	-0.565922	0.195414	-0.349494
0	0	1	1	0	0.573445	-0.565922	0.195414	-0.349494
0	0	1	1	0	1.533034	-0.204909	0.195414	-0.349494
0	0	1	1	0	0.381527	-0.565922	0.195414	-0.349494
0	0	1	0	0	3.260295	0.156105	0.195414	1.671136

Figure 28: Dataframe finale

6 Implémentation

Dans cette section, nous décrirons en détail les étapes de mise en œuvre de notre projet de prédiction de la souscription aux dépôts à terme. Nous commencerons par la division des données en ensembles d'entraînement et de test, puis nous explorerons plusieurs algorithmes de machine learning pour identifier le meilleur modèle pour notre problème. Enfin, nous évaluerons les performances du modèle retenu et discuterons des résultats obtenus.

6.1 Division des données

Avant de commencer l'implémentation des modèles, nous avons divisé nos données en ensembles d'entraînement et de test. L'ensemble d'entraînement est utilisé pour entraîner les modèles, tandis que l'ensemble de test est utilisé pour évaluer les performances des modèles sur des données non vues.

```python
# Séparer les caractéristiques (features) et la variable cible
X = data.drop('y', axis=1)
y = data['y']

# Diviser les données en ensembles d'entraînement et de test
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Python

Figure 29: Division des données

6.2 Sélection des Modèles

Nous avons sélectionné plusieurs algorithmes de classification pour notre problème de prédiction de la souscription aux dépôts à terme. Les algorithmes que nous avons considérés sont les suivants :

- **Gradient Boosting :** Le boosting de gradient est une méthode ensembliste qui combine plusieurs modèles faibles pour former un modèle plus robuste.
- **Naive Bayes :** Le classificateur naïf de Bayes est basé sur le théorème de Bayes et suppose que les caractéristiques sont indépendantes entre elles.
- **Régression Logistique :** La régression logistique est un modèle linéaire adapté à la classification binaire.
- **SVM (Support Vector Machines) :** Les SVM sont des algorithmes de classification puissants.
- **Random Forest :** Random Forest est un algorithme d'ensemble basé sur des arbres de décision.

```python
# Créer les modèles
models = {
    'Logistic Regression': LogisticRegression(),
    'Random Forest': RandomForestClassifier(),
    'Support Vector Machine': SVC(probability=True),
    'Gradient Boosting': GradientBoostingClassifier(),
    'Naive Bayes': GaussianNB()
}
```

Python

6.3 Entraînement et évaluation des modèles

Nous avons entraîné chaque modèle sur l'ensemble d'entraînement. Une fois les modèles entraînés, nous les avons évalués sur l'ensemble de test en utilisant différentes mesures de performance telles que la précision, le rappel, le score F1 et l'aire sous la courbe ROC (AUC-ROC). Ces mesures nous permettent d'évaluer à la fois la capacité de prédiction des modèles et leur capacité à généraliser sur de nouvelles données.

```
# Entraîner et évaluer les modèles
for model_name, model in models.items():
    model.fit(X_train, y_train)
    y_pred = model.predict(X_test)

    # Accuracy
    accuracy = accuracy_score(y_test, y_pred)
    print(f'{model_name} - Accuracy: {accuracy:.4f}')

    # Matrice de Confusion
    confusion_mat = confusion_matrix(y_test, y_pred)
    print(f'{model_name} - Confusion Matrix:\n{confusion_mat}')

    # Précision, Recall, F1-Score
    precision = precision_score(y_test, y_pred)
    recall = recall_score(y_test, y_pred)
    f1 = f1_score(y_test, y_pred)
    print(f'{model_name} - Precision: {precision:.4f}, Recall: {recall:.4f}, F1-Score: {f1:.4f}')

    # AUC-ROC
    y_pred_prob = model.predict_proba(X_test)[:, 1]
    auc_roc = roc_auc_score(y_test, y_pred_prob)
    print(f'{model_name} - AUC-ROC: {auc_roc:.4f}')

    print('\n')  # Ajout d'une ligne vide entre les résultats de différents modèles
```

6.4 Sélection du meilleur modèle

Après avoir entraîné et évalué plusieurs modèles d'apprentissage automatique, nous devons maintenant sélectionner le meilleur modèle pour notre projet de prédiction de la souscription aux dépôts à terme.

En analysant les performances de chaque algorithme sur l'ensemble de test, nous avons observé les métriques suivantes :

- **Gradient Boosting :**
  - Précision : 0.6506
  - Rappel : 0.4079

  - F1-Score : 0.5014
  - AUC-ROC : 0.9183
- **Random Forest :**
  - Précision : 0.6600
  - Rappel : 0.3932
  - F1-Score : 0.4928
  - AUC-ROC : 0.9265
- **Régression Logistique :**
  - Précision : 0.6656
  - Rappel : 0.3648
  - F1-Score : 0.4713
  - AUC-ROC : 0.9069
- **Naive Bayes :**
  - Précision : 0.3565
  - Rappel : 0.5225
  - F1-Score : 0.4238
  - AUC-ROC : 0.7951
- **SVM (Support Vector Machines) :**
  - Précision : 0.6679
  - Rappel : 0.3281
  - F1-Score : 0.4401
  - AUC-ROC : 0.9054

En considérant ces métriques, le modèle de **Random Forest** semble offrir les meilleures performances initiales avec un bon équilibre entre précision, rappel et score F1, ainsi qu'une AUC-ROC élevée. Il est donc judicieux de le choisir comme modèle de prédiction principal pour notre projet. Cependant, il est important de noter que ces résultats sont basés sur les performances initiales des modèles. Après avoir amélioré tous les algorithmes et ajusté les hyperparamètres, il sera nécessaire de réévaluer ces performances pour confirmer notre choix final.

7 Déploiment du modèle

Pour le déploiement du meilleur modèle, nous avons choisi d'utiliser Flask pour créer une API.

Nous avons mis en place une route spécifique dans notre application Flask pour gérer les requêtes de prédiction. Cette route prend les données en entrée, les prépare pour le modèle (encodage des variables catégorielles, normalisation des données numériques), puis utilise le modèle pré-entraîné pour faire une prédiction.

```python
# Créer une route pour les prédictions
@app.route('/predict', methods=['POST'])
def predict():
    # Obtenir les données à partir de la requête POST
    data = request.json
    df = pd.DataFrame([data])

    # Encoder les variables catégorielles
    categorical_cols = ['metier', 'situation_matrimoniale', 'education', 'defaut_credit', 'logement', 'pret', 'contact', 'mois', 'jour_de_semaine', 'resultat
    df_encoded = pd.get_dummies(df, dtype=int)

    # Vérifier et ajouter les colonnes manquantes
    missing_cols = set(encoder.columns) - set(df_encoded.columns)
    for col in missing_cols:
        df_encoded[col] = 0

    # Réorganiser les colonnes selon l'ordre de l'encoder
    df_encoded = df_encoded[encoder.columns]


    # Normaliser les données numériques
    numerical_cols = ['age', 'campagne', 'nombre_de_jour_ecoule', 'nombre_contact_precedent', 'duree_appel']
    df_normalized = scaler.transform(df_encoded[numerical_cols])

    # Remplacer les données numériques par les données normalisées
    df_encoded[numerical_cols] = df_normalized


    # Faire la prédiction
    prediction = model.predict(df_encoded)


    # Renvoyer la prédiction sous forme de JSON
    return jsonify({'prediction': prediction.tolist()})

# Lancer l'application Flask
if __name__ == '__main__':
    app.run(port=5000)
```

Figure 30: Application Flask avec une route pour la prédiction

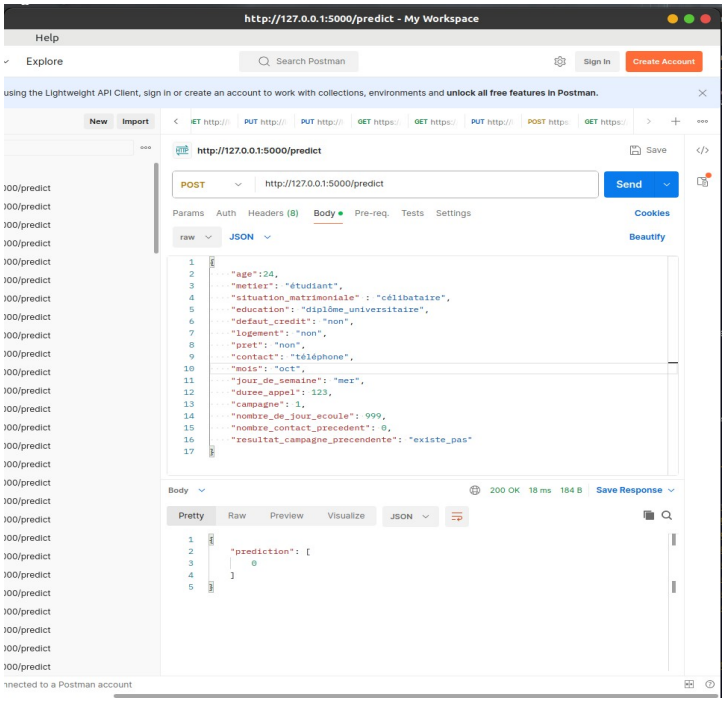

Figure 31: Test de prédiction pour le cas d'une non-souscription à un dépôt à terme

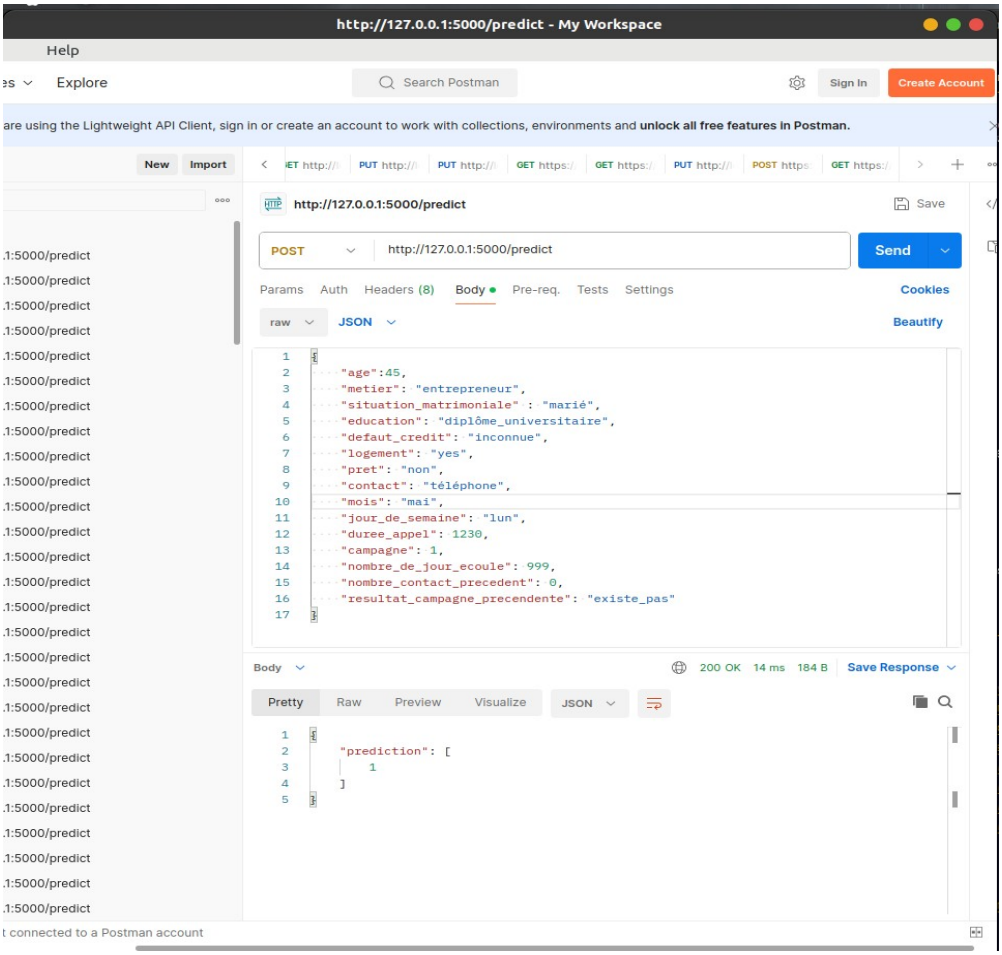

Figure 32: Test de prédiction pour le cas d'une souscription à un dépôt à terme

8 Conclusion et Perspectives

Ce mémoire a présenté une approche complète pour la prédiction de la souscription à un dépôt à terme dans le secteur bancaire, en mettant en œuvre différentes techniques de data science et de machine learning. Nous avons débuté par l'exploration et la préparation des données, en identifiant les caractéristiques les plus pertinentes pour notre modèle. Ensuite, nous avons entraîné plusieurs algorithmes de classification et évalué leurs performances, en accordant une attention particulière à la gestion des données déséquilibrées.

Notre étude a démontré que le modèle de forêt aléatoire était le plus performant pour prédire la souscription à un dépôt à terme, avec une précision et un rappel élevés. Nous avons également mis en place une API Flask pour déployer ce modèle, permettant ainsi une utilisation facile et accessible à distance.

En conclusion, ce travail a permis de développer un modèle de prédiction précis et robuste pour anticiper la souscription à un dépôt à terme dans le secteur bancaire. Pour les perspectives futures, il serait intéressant d'explorer davantage l'impact de différentes variables sur la décision de souscription, notamment en utilisant des techniques d'interprétabilité de modèles telles que LIME ou SHAP. De plus, l'ajout de nouvelles fonctionnalités ou l'exploration de modèles plus avancés tels que les réseaux neuronaux pourraient améliorer encore les performances prédictives du modèle.

Ce travail ouvre la voie à de nombreuses opportunités de recherche et d'application dans le domaine de la banque et de la finance, en offrant des outils précieux pour anticiper et optimiser les campagnes de marketing et de fidélisation des clients.

Bibliographie

Sklearn. (n.d.). Scikit-learn: Machine learning in Python. https://scikit-learn.org/stable/

Breiman, L. (2001). Random forests. Machine learning, 45(1), 5-32.

Friedman, J. H. (2001). Greedy function approximation: A gradient boosting machine. Annals of statistics, 1189-1232.

Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. Journal of machine learning research, 3(Mar), 1157-1182.

Zheng, A., & Casari, A. (2018). Feature engineering for machine learning: principles and techniques for data scientists. O'Reilly Media, Inc.

Han, J., Kamber, M., & Pei, J. (2011). Data mining: concepts and techniques. Elsevier.