

Code : 1332

Année Universitaire : 2021/2022

Université de la Manouba  
École Supérieure d'Économie Numérique



## MÉMOIRE DE MASTÈRE

PRÉSENTÉ EN VUE DE L'OBTENTION DU DIPLÔME DE

MASTÈRE PROFESSIONNEL EN  
"DATA SCIENCE AND SOFTWARE DEVELOPMENT"

Élaboré par :

**Narjess NEBLI**

Sujet :

---

Implémentation d'un système intelligent de Early Warning  
pour détecter les clients risqués pour le département risque  
d'une Banque de détails

---

Organisme d'accueil :



Sous la direction de :

**Encadrant pédagogique : Mme.DHOUHA MAATAR**

**Encadrant professionnel : M.AHMED REBAI**

# Remerciement

*S'il faut beaucoup de motivation, de rigueur et d'enthousiasme pour mener à bien ce mémoire, alors, ce travail a eu besoin de la contribution de plusieurs personnes, que je tiens à remercier en signe de ma gratitude et profonde reconnaissance.*

*Je commence par exprimer mes plus vifs remerciements à mon encadrante universitaire **Dr. Dhouha Maatar**, qui m'a toujours soutenue durant toute la période du stage et qui s'est dévoué corps et âme afin de mener à bien le présent travail. Son écoute active, sa disponibilité, son empathie et sa bonté tout au long de ce parcours m'ont poussé à travailler plus fort et à dépasser toutes les difficultés et les mauvais moments que j'ai rencontré. Ce projet n'a abouti que grâce à la confiance qu'elle m'a accordée et les efforts qu'elle a consenti.*

*Je tiens également à exprimer ma profonde gratitude et sincères reconnaissance à mon maitre de stage **M. Ahmed Rebai**, Tech Lead chez Value. Finir la totalité du mémoire en si peu de temps, n'a pas était une tâche facile, et je n'aurais pas tant réussi si je n'avais pas reçu ses conseils, ainsi que sa force de persuasion. Grâce au partage de son expertise au quotidien et à sa confiance j'ai pu m'accomplir totalement dans mes missions. Il fut d'une aide précieuse dans les moments les plus délicats.*

*Je remercie également toute **l'équipe Data de Value** , mes collègues et mes amis, travailler avec vous ces derniers mois m'a aidé à grandir personnellement et professionnellement et je vous en suis très reconnaissante. Je tiens à remercier également le manager **M. Faress Souissi**, de m'avoir accueilli si chaleureusement au sein de son équipe.*

*Je termine par exprimer mes vifs remerciements aux membres du jury pour l'honneur qu'ils me présentent en acceptant d'évaluer le présent travail et mon respect à tout le corps administratif et professoral d **l'École Supérieure d'Économie Numérique** pour l'intérêt qu'ils portent à notre formation.*

# *Dédicace*

*Je dédie ce modeste travail accompagné d'un profond amour :*

## ***À mes très chers parents Hatem et Souad***

*Ma mère, tu as fait de moi la personne que je suis. Tu m'as donné la vie et l'envie de vivre, d'avancer et de faire de mon mieux rien que pour te rendre fière de moi, tu m'a toujours arrosé d'amour et d'espoirs.*

*Mon père, mon support dans la vie qui m'a appris, m'a supporté et m'a dirigé vers la gloire. Que ce travail traduit ma gratitude et mon affection.*

*Je vous dois tout mon succès et mon bonheur.*

## ***À ma chère sœur Syrine et mon adorable frère Tarek***

*En reconnaissance de leur soutien moral et leur affection toujours constante.*

*Je vous souhaite tout le succès et le bonheur dans le monde.*

## ***À mon Cher Ahmed***

*Depuis que je t'ai connu, tu n'as cessé de me soutenir et de m'épauler.*

*Ton amour ne m'a procuré que confiance et stabilité.*

*J'aimerais bien que tu trouves dans ce travail l'expression de mes sentiments de reconnaissance les plus sincères car grâce à ton aide et à ta patience avec moi que ce travail a pu voir le jour.*

## ***À mon adorable cousine Sarra***

*Pour ses encouragements et son aide qui m'a permis de mener à bien le présent travail.*

## ***À tous mes amis***

*Pour tous les moments de pure joie qu'on a partagés ensemble et les merveilleux souvenirs qu'on a vécus.*

*A ma famille, mes proches et à tous ceux que j'aime et tous ceux qui m'aiment, que ce mémoire soit le plus petit des témoignages de mes sentiments les plus affectueux.*

*- Narjess -*

# Table des matières

<b>Liste des abréviations</b>	<b>11</b>
<b>Introduction générale</b>	<b>13</b>
<b>1 Étude du projet</b>	<b>14</b>
1.1 Cadre du projet . . . . .	15
1.1.1 Présentation de l'organisme d'accueil . . . . .	15
1.1.2 Activités de l'entreprise . . . . .	16
1.2 Contexte général du projet . . . . .	16
1.2.1 Contexte du projet . . . . .	16
1.2.2 Problématique . . . . .	17
1.2.3 Solution proposée . . . . .	17
1.3 Méthodologie adoptée . . . . .	18
1.3.1 Etude comparative . . . . .	18
1.3.2 Focus sur la méthodologie CRISP-DM . . . . .	23
1.4 État de l'art . . . . .	25
1.4.1 Aperçu sur la fintech . . . . .	25



1.4.2	Focalisation sur le Risk Management et les risques de crédit . . .	25
1.4.3	Aperçu sur la Data Science . . . . .	26
1.4.4	Application de la Data Science à la Fintech et au risk management . . . . .	27
<b>2</b>	<b>Compréhension et préparation des données</b>	<b>29</b>
2.1	Identification de source des données . . . . .	30
2.1.1	Compréhension de la structure des données . . . . .	30
2.1.2	La forme des tables du DataSet . . . . .	31
2.1.3	La forme des tables du DataSet . . . . .	32
2.2	Analyse exploratoire des données (EDA) . . . . .	32
2.3	Préparation des données . . . . .	41
2.3.1	Suppression des valeurs aberrantes . . . . .	41
2.3.2	Feature Engineering . . . . .	41
2.3.3	Transformation des données . . . . .	43
2.3.4	Intégration des données . . . . .	44
2.3.5	Nettoyage des données . . . . .	45
2.3.6	Division des données . . . . .	46
2.3.7	Échantillonnage aléatoire . . . . .	46
<b>3</b>	<b>Modélisation et évaluation</b>	<b>48</b>
3.1	Modélisation . . . . .	49
3.1.1	Classification supervisée . . . . .	49
3.1.2	Synthèse . . . . .	61
3.2	Évaluation . . . . .	63



3.3	Pipeline de la prédiction . . . . .	68
<b>4</b>	<b>Déploiement</b>	<b>70</b>
4.1	Analyse des besoins . . . . .	71
4.1.1	Identification des acteurs . . . . .	71
4.1.2	Identification des besoins fonctionnels . . . . .	71
4.1.3	Les besoins non fonctionnels . . . . .	72
4.1.4	Diagramme de cas d'utilisation . . . . .	72
4.1.5	Analyse . . . . .	73
4.2	Implémentation . . . . .	74
4.2.1	Architecture physique adoptée . . . . .	74
4.2.2	Intégration du modèle de prédiction . . . . .	76
4.2.3	Réalisation des interfaces . . . . .	77
4.2.4	Outils de développement . . . . .	79
4.2.5	Environnement matériel . . . . .	80
	<b>Conclusion et perspectives</b>	<b>81</b>
	<b>Références</b>	<b>82</b>

# Table des figures

1.1	Logo de Value Digital Services[1]	16
1.2	Activités de Value	16
1.3	Cycle de vie de la méthodologie « Microsoft TDSP» [2]	19
1.4	Rôles définis par la méthodologie « RAMSYS»	19
1.5	Cycle de vie de la méthodologie « Data Science Edge»	20
1.6	Couches de la méthodologie « Analytics Canvas »	20
1.7	Cycle de vie de la méthodologie « Domino DS Lifecycle » [3]	21
1.8	Cycle de vie de la méthodologie « CRISP-DM » [4]	22
1.9	Cycle de vie de la méthodologie « IBM Data Science Master Plan »[5]	22
2.1	La structure du DataSet	30
2.2	Le diagramme de la structure du DataSet	31
2.3	Le nombre de ligne et de features dans chaque table	32
2.4	La distribution du statut de remboursement des prêts	33
2.5	La distribution des revenus totaux des clients	34
2.6	Boxplot de la variable AMT_INCOME	34
2.7	La ditribution des Motants des crédits	35



2.8	La ditribution de nombre d'années pendant lesquelles une personne a travaillé . . . . .	36
2.9	Boxplot de la variable Days_Employed . . . . .	36
2.10	Boxplot de la variable Days_Employed . . . . .	37
2.11	Analyse de la variable genre des clients . . . . .	37
2.12	Analyse des variables EXT_SOURCE_1, EXT_SOURCE_2, EXT_SOURCE_3	38
2.13	Analyse de la table 'Application_train' . . . . .	39
2.14	Analyse de la table 'Bureau' . . . . .	39
2.15	Analyse de la table 'Bureau_Balance' . . . . .	39
2.16	Analyse de la table 'POS_CASH_Balance' . . . . .	40
2.17	Analyse de la table 'Credit_Card_Balance.' . . . .	40
2.18	Analyse de la table 'Previous_Application' . . . . .	40
2.19	Analyse de la table 'Installments_Payments' . . . . .	41
2.20	Un échantillon du résultat de l'encodage de la variable 'NAME_FAMILY_STATUS'	43
2.21	Un échantillon du résultat de l'agrégation de la variable AMT_CREDIT'	44
2.22	Un échantillon du résultat de l'imputation des valeurs manquantes . .	46
2.23	Résultat de l'Oversampling sur les données d'entraînement . . . . .	47
3.1	La fonction Sigmoid [6] . . . . .	50
3.2	Une illustration du concept d'agrégation bootstrap (Bagging) [7] . . .	53
3.3	Les étape de fonctionnement de l'algorithme des forêts aléatoires[8] .	54
3.4	Une illustration présentant l'intuition derrière l'algorithme de boosting)[7] . . . . .	55
3.5	Développement de l'arbre en leaf-wise par l'algorithme LightGBM [9]	55





3.6	Développement de l'arbre en level-wise par les autres algorithmes de Boosting [9] . . . . .	56
3.7	Architecture d'un ANN[10] . . . . .	59
3.8	L'architecture de notre modèle ANN . . . . .	61
3.9	Matrice de confusion du modèle Random Forest . . . . .	63
3.10	Matrice de confusion du modèle Decision Tree . . . . .	63
3.11	Matrice de confusion du modèle XGboost . . . . .	64
3.12	Matrice de confusion du modèle Logistic Regression . . . . .	64
3.13	Matrice de confusion du modèle XGboost . . . . .	64
3.14	Matrice de confusion du modèle ANN . . . . .	64
3.15	ROC-AUC curves pour le modèle Random Forest . . . . .	65
3.16	ROC-AUC curves pour le modèle Decision Tree . . . . .	65
3.17	ROC-AUC curves pour le modèle Logistic Regression . . . . .	65
3.18	ROC-AUC curves pour le modèle Lightgbm . . . . .	66
3.19	ROC-AUC curves pour le modèle XGboost . . . . .	66
3.20	ROC-AUC curves pour le modèle ANN . . . . .	66
3.21	Pipeline de la prédiction . . . . .	68
4.1	Diagramme de cas d'utilisation global . . . . .	72
4.2	Diagramme de séquence système du cas d'utilisation «Charger un fichier de demandes de crédits» . . . . .	73
4.3	Diagramme de séquence système du cas d'utilisation «Prédire le risque d'une demande de crédit» . . . . .	74
4.4	Architecture à deux niveaux . . . . .	75
4.5	Architecture Modèle-Vue-Contrôleur . . . . .	76



4.6	Intégration du modèle de prédiction [11]	76
4.7	Interface d'authentification	77
4.8	Interface de chargement de fichier de demandes de crédits	77
4.9	Interface de consultation du rapport d'EDA	78
4.10	Interface de prédiction de défaut de paiement « Cas d'un client non-risqué »	78
4.11	Interface de prédiction de défaut de paiement « Cas d'un client Risqué »	79

# Liste des tableaux

3.1	Modèles utilisées et leurs hyperparamètres . . . . .	62
3.2	Les performances des modèles . . . . .	63
4.1	Environnement matériel . . . . .	80



# Liste des abréviations

---

- **EDA** : Exploratory Data Analysis
- **Lgbm** : Lightgbm classifier
- **ANN** : réseau de neurones artificiels
- **PoC** : Proof of Concept
- **MLP** : Perceptron multicouches
- **AUC** : Area Under Curve
- **ROC** : Receiver Operating Characteristic
- **CSV** : Comma Separated Values
- **MVC** : Modèle-Vue-Contrôleur



# Introduction générale

---

Depuis quelques années, le monde a été témoin d'une évolution technologique qui a permis aux organisations d'automatiser leurs processus administratifs compte tenu de la puissance de la transformation digitale.

Cette automatisation a donné naissance à des processus qui sont gérés entièrement à travers des transactions numériques ce qui a généré des quantités massives de données dont la gestion et l'exploitation avec les méthodes traditionnelles sont devenues intraitables.

C'est dans ce contexte que le concept de Data Science a vu le jour. La majorité des entreprises ont eu recours à ce domaine afin de pouvoir exploiter les données excessives dont elles disposent tout en ayant comme objectif ultime l'extraction de connaissances qui auront comme but d'assurer une prise de décision efficace et une conception de meilleurs produits et services plus innovants.

Aujourd'hui, nous pouvons constater l'application de la Data Science dans les différents secteurs d'activités notamment le secteur bancaire. En effet, Dans un contexte de concurrence accrue et de dématérialisation croissante des transactions, les banques, les grandes pourvoyeuses de données, mettent à profit l'intelligence artificielle pour mieux satisfaire les besoins de leurs clients et améliorer la gestion des risques.

Afin d'anticiper les défauts de paiement, les banques de détail dont l'activité principale est l'affectation des crédits aux entreprises et aux particuliers, se sont servies de la Data Science pour faire le scoring des dossiers de crédits.

Dans ce contexte s'inscrit notre projet de fin d'études. En effet, Value, notre entreprise d'accueil, nous a confié la mission de l'implémentation d'un système intelligent de Early Warning pour la détection des clients risqués d'un défaut de paiement pour le département risque d'une banque de détail. Ce projet a pour but le Scoring des dossiers de crédits en vue de faciliter la détection des clients ayant une relation avec la probabilité d'un défaut.

Ce présent rapport est ainsi composé de quatre chapitres :

- Le premier chapitre «Étude du projet» est consacré pour la présentation de l'organisme d'accueil, le cadre général du projet qui va nous donner une idée



sur le problème métier et la solution proposée ainsi que la méthodologie de travail adoptée et finalement il mettra l'accent sur l'état de l'art de la Fintech, le risk management, les risques de crédit et la Data Science.

- Le deuxième chapitre « Compréhension et préparation des données » portera sur l'identification de la source des données, l'analyse exploratoire des données (EDA) et les actions de prétraitements appliquées.
- Le troisième chapitre intitulé « Modélisation et évaluation » abordera la phase de modélisation qui contiendra la construction des modèles et une étude théorique des différents algorithmes construits ainsi que la phase d'évaluation des résultats obtenus afin de valider le modèle le plus performant.
- Le quatrième chapitre « Déploiement » mettra l'accent sur l'analyse et l'implémentation de l'application web.

# Chapitre 1

## Étude du projet



## Introduction

Dans ce chapitre, nous présenterons le cadre général de notre projet, allant de la présentation de l'organisme d'accueil au sein duquel nous avons eu l'opportunité d'effectuer notre stage, à la description du contexte général de notre projet. Nous enchaînons par la suite par une étude comparative de quelques méthodologies de gestion de projets en Data Science et nous allons finir par mettre en évidence la démarche à suivre pour concrétiser la solution proposée.

### 1.1 Cadre du projet

Dans le cadre de mémoire de fin d'études au sein de la société Value Digital Services, l'achèvement de deux ans d'études universitaires au sein de l'École Supérieure d'Économie Numérique (ESEN) se concrétise avec ce projet qui consiste à l'implémentation d'un système intelligent de Early Warning pour détecter les clients risqués pour le département risque d'une Banque de détail, en vue de l'obtention du diplôme de Mastère professionnel en "Data Science and Software Development".

#### 1.1.1 Présentation de l'organisme d'accueil

Value Digital Services est un cabinet de conseil en stratégie et en transformation digitale, fondée en 2019, qui accompagne ses clients dans l'élaboration et le pilotage de leurs stratégies et dans le déploiement et la mise en œuvre opérationnelle de leurs feuilles de route digitales et data.[1]

Value s'engage aux côtés de ses partenaires et clients pour une création de valeur durable dans le temps pour la société et l'économie. [1]

D'après ses fondateurs :

« L'objectif de Value, c'est de contribuer à créer de la valeur dans un monde en pleine mutation.

Ma conviction est que ceci n'est possible qu'à travers une combinaison alliant crédibilité et relations de confiance avec nos clients et nos partenaires ainsi qu'un état d'esprit d'ouverture et de diversité permettant aux talents de s'exprimer. » [1]





FIGURE 1.1 – Logo de Value Digital Services[1]

### 1.1.2 Activités de l'entreprise

Value aide ses collaborateurs à s'épanouir et à réaliser pleinement leur potentiel, en contribuant à des projets majeurs et à très fort impact tout en développant leurs expertises métiers et technologiques au sein de ses différentes divisions. [1]

La figure ci-dessous détaille les différentes activités de Value :

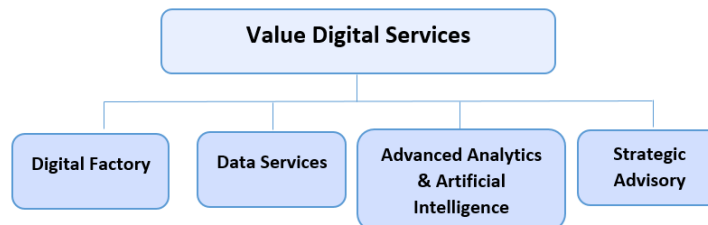


FIGURE 1.2 – Activités de Value

## 1.2 Contexte général du projet

### 1.2.1 Contexte du projet

Une banque de détail est une banque de distribution de services bancaires, son activité principale consiste en la collection des dépôts, la gestion des moyens de paiement du public ainsi que l'affectation des crédits aux entreprises et aux particuliers. Ce sont alors les banques de détail qui réalisent l'essentiel de la création monétaire à travers la production de crédit.

A cet égard, une institution financière spécialisée dans la banque de détail est face à un risque de crédit.

En effet, en finance, le risque de défaut est la probabilité qu'une entreprise ou un particulier ne soit pas en mesure de payer ses dettes en respectant les conditions exigées par la banque. Dans ce contexte, la banque peut faire face à des risques de dépassement de date limite de paiement ou encore le non-paiement d'une partie ou de la totalité de la somme empruntée.



Dès qu'un créancier accorde alors un prêt à un débiteur, il court le risque que ce dernier ne le rembourse pas.

### 1.2.2 Problématique

À chaque fois qu'une institution financière accorde un prêt à une personne ou à une société, elle évalue le niveau de risque lié à la transaction. En effet, parfois, le particulier ou l'entreprise sera dans l'impossibilité de rembourser son prêt. Dans le monde dynamique dans lequel nous vivons, des imprévus se produisent et les circonstances varient. De nombreuses personnes ont ainsi du mal à obtenir des prêts en raison d'antécédents de crédit insuffisants ou inexistants. Et d'autres obtiennent des crédits et confronteront des difficultés à les rembourser.

L'objectif d'une telle institution consiste à offrir une expérience d'emprunt positive et sûre à ses clients, en s'assurant que les clients capables de rembourser leurs prêts ne sont pas rejetés, et ne pas se trouver ainsi face à un risque de défaut.

Le risque de crédit a des facteurs ou des caractéristiques liés à la personne ou à l'entreprise qui ont une relation avec la probabilité d'un défaut. Le point principal est que les facteurs de risque de défaut peuvent être mesurés et analysés pour les modèles liés au défaut. Par conséquent, la probabilité de risque d'une personne ou d'une institution n'est pas aléatoire. C'est dans ce contexte que nous pourrions constater la contribution du Machine Learning.

### 1.2.3 Solution proposée

Nous proposons de développer un "proof of concept"<sup>1</sup> permettant de prédire si un client sera en mesure de rembourser un prêt ou non, en se basant sur les données précédentes du comportement des clients et en utilisant des techniques de la data science.

Notre livrable se présentera sous la forme d'une application web contenant quelques fonctionnalités permettant à l'utilisateur de gérer en premier lieu le fichier de demandes de crédits, avoir un aperçu sur ses données sous la forme d'un rapport présentant toute une Analyse Exploratoire (EDA) et il y aura également un persona prédictif où l'utilisateur aura la possibilité de choisir une demande de crédit à traiter et avoir comme output une prédiction finale ainsi qu'une probabilité prédite de défaut de paiement afin de décider d'accepter ou non sa demande.

---

1. Une preuve de concept (PoC), ou encore démonstration de faisabilité, est un produit ayant pour objectif de démontrer la faisabilité d'un système.



## 1.3 Méthodologie adoptée

Une méthodologie est une stratégie générale pour guider les processus et les activités dans un domaine donné.

Les méthodologies utilisées en Data Science sont agiles et itératives. Cela commence par un raisonnement inductif, qui consiste à construire des connaissances à partir des données. L'approche est construite par étapes, en formulant d'abord des hypothèses, puis en validant ces hypothèses à l'aide d'algorithmes statistiques et/ou d'apprentissage automatique. [12]

### 1.3.1 Etude comparative

Il existe évidemment plusieurs méthodologies développées dans le but d'assurer le bon déroulement du cycle de vie d'un projet Data Science. Afin de réussir à bien choisir la démarche adéquate pour orienter notre projet et créer une meilleure qualité de livrable, nous avons choisi de précéder le choix de la méthodologie par une étude comparative entre les méthodologies les plus connues en Data Science.

En effet, les principaux défis d'un projets Data Science sont d'assurer :

- **Gestion d'équipe** : Il s'agit de la définition des rôles pour faciliter la coordination entre les membres de l'équipe et entre les parties prenantes.
- **Gestion de projet** :
  - Définition du flux de travail du cycle de vie de la Data Science.
  - Normalisation de la structure des dossiers.
  - Documentation continue du projet.
  - Visualisation de l'état du projet.
  - Alignement des objectifs de la Data Science et de l'entreprise.
  - Consolidation des mesures de performance et de réussite.
- **Gestion des données et des informations** :
  - Reproductibilité : création d'un référentiel de connaissances.
  - Déploiement robuste : création du code, des données et des modèles.
  - Création de modèles pour la génération de connaissances et de valeurs.

En se basant sur les défis cités auparavant nous allons comparer sommairement quelques méthodologies de gestion de projet en Data Science.



- **Microsoft TDSP**

Etant l'acronyme pour « Team Data Science Process ». Microsoft a tenté de créer une approche englobante, qui permet de gérer toute sorte de projets en science des données. [13]

Le cycle de vie de la méthodologie TDSP se présente comme suit :

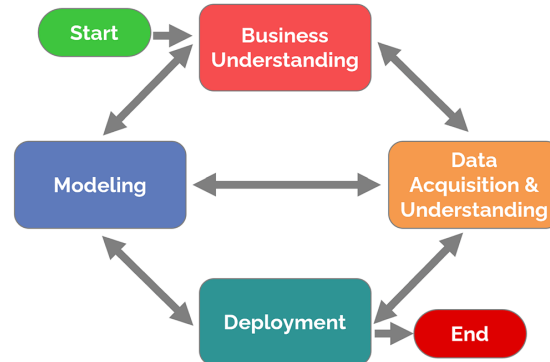


FIGURE 1.3 – Cycle de vie de la méthodologie « Microsoft TDSP» [2]

- **Avantages** : Méthodologie intégrale : fournit des processus à la fois sur le projet, l'équipe et la gestion des données et de l'information. [13]
- **Inconvénients** : Dépendance excessive à l'égard des outils et technologies Microsoft.[13]

- **RAMSYS**

Cette méthodologie suit et étend la méthodologie CRISP-DM et permet au travail du Data Mining d'être dépensé à des endroits très différents communiquant via un outil basé sur le web. [13]

RAMSYS définit trois rôles :

Data Master	✓ responsable de la maintenance de la base de données et de l'application des transformations nécessaires
Modellers	✓ tester la validité de chaque hypothèse ✓ produire de nouvelles connaissances, ils peuvent suggérer de nouvelles transformations de données au 'Data Master'.
Management Committee	✓ Assure que les informations circulent au sein du Network et qu'une bonne solution est fournie ✓ Gérer l'interface avec le client et la mise en œuvre du défi lié au projet de Data Science

FIGURE 1.4 – Rôles définis par la méthodologie « RAMSYS»

- **Avantages** :Tient compte des équipes distribuées et permet le partage des informations et des connaissances. [13]
- **Inconvénients** : Manque de solution pour le partage des ensembles de données et des modèles.[13]
- **Data Science Edge**  
Il s'agit d'un modèle de processus amélioré pour s'adapter aux technologies du big data et aux activités de la Data Science. [13]  
Le cycle de vie du DSE est divisé en quatre quadrants :

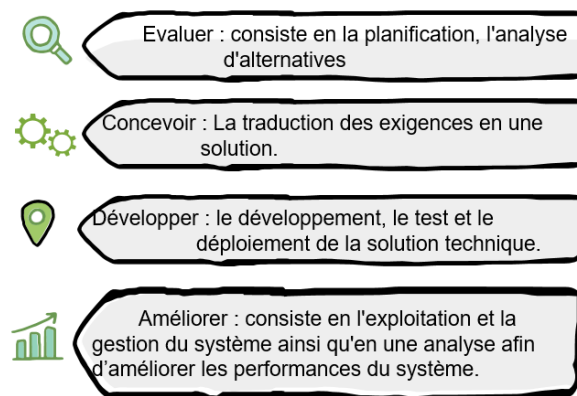


FIGURE 1.5 – Cycle de vie de la méthodologie « Data Science Edge»

- **Avantages** : Amélioration du modèle de processus CRISP-DM pour prendre en compte les technologies du big data et les activités de La Data Science. [13]
- **Inconvénients** : Les défis liés à la gestion de l'équipe sont négligés.[13]
- **Analytics Canvas**  
Il s'agit d'une technique de spécification semi-formelle pour la conception de projets d'analyse de données.  
Décrit un cas d'utilisation analytique et documente l'infrastructure de données nécessaire au cours de la planification initiale d'un projet d'analyse de données. [13]  
Il propose un modèle à quatre couches et attribue un rôle spécialisé à chaque phase, comme suit :



FIGURE 1.6 – Couches de la méthodologie « Analytics Canvas »



- **Avantages** : Difficile à mettre en œuvre en tant que framework évolutif tout au long de l'ensemble du développement du projet.[13]
- **Inconvénients** : Les défis liés à la gestion de l'équipe sont négligés.[13]

- **Domino DS Lifecycle**

Une approche holistique de l'ensemble du cycle de vie du projet, de la conception à la livraison et au monitoring.

Le cycle de vie de cette méthodologie est illustré comme suit, en effet, au cours de ce cycle chaque phase doit être terminée avant que la suivante puisse commencer. Il n'y a pas de chevauchement entre les phases.[13]

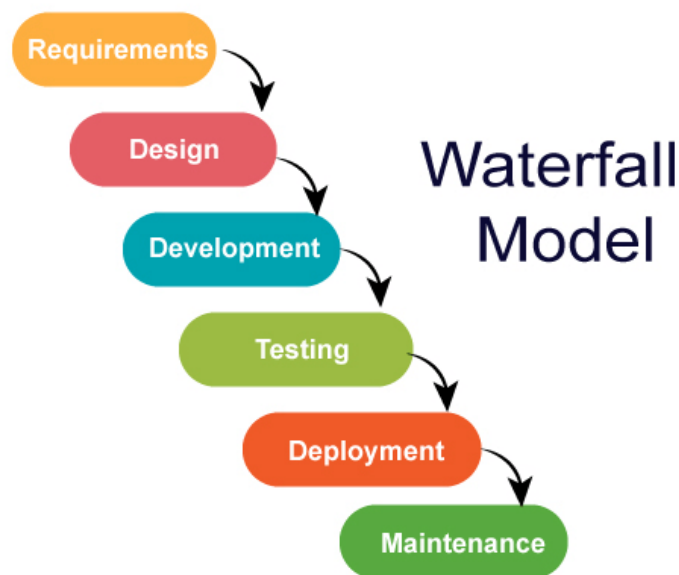


FIGURE 1.7 – Cycle de vie de la méthodologie « Domino DS Lifecycle » [3]

- **Avantages** : Approche holistique du cycle de vie du projet. Intègre efficacement la data science, le génie logiciel et les approches agiles. [13]
- **Inconvénients** : Une méthodologie informative plutôt que prescriptive. [13]

- **CRISP-DM**

Cross Industry Standard Process for Data Mining » été créé dans l'objectif de servir des projets de « data mining ». Considérée comme la plus utilisée. [13]

Son cycle de vie est défini comme suit :

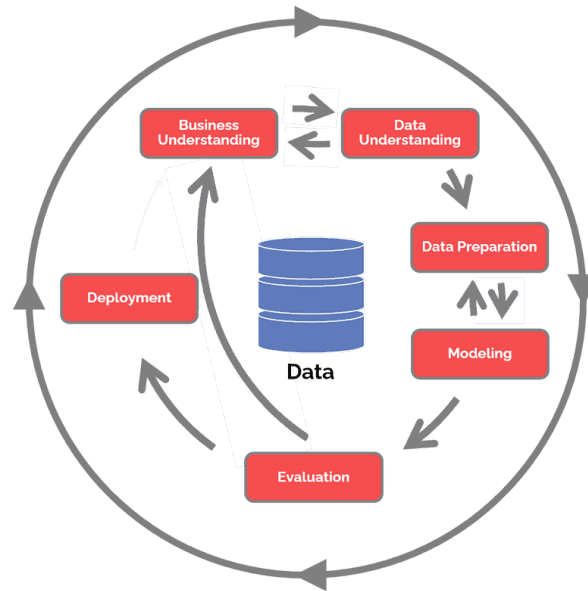


FIGURE 1.8 – Cycle de vie de la méthodologie « CRISP-DM » [4]

- **Avantages** : Processus itératif cohérent et bien documenté. [13]
- **Inconvénients** : N'explique pas comment les équipes doivent s'organiser. [13]

### • IBM Data Science Master Plan

Présente certaines ressemblances avec CRISP-DM, mais il fournit un certain nombre de nouvelles pratiques. Globalement, cette méthodologie structure un projet de Data Science en plus de phases (10) que CRISP-DM (6). Le cycle de vie de la méthodologie IBM Data Science Master Plan se présente comme suit : [13]

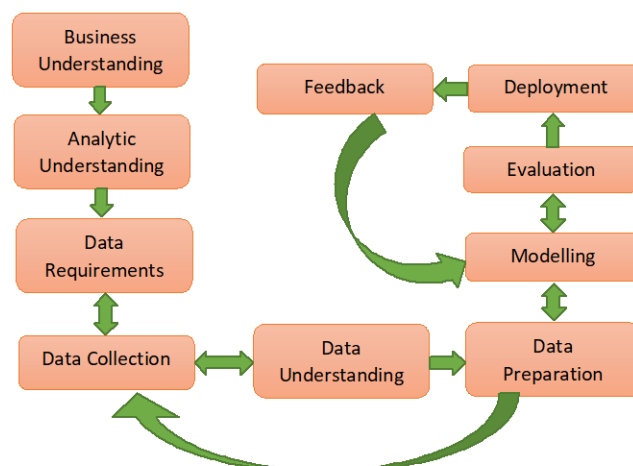


FIGURE 1.9 – Cycle de vie de la méthodologie « IBM Data Science Master Plan » [5]



- **Avantages** : Fournit de nouvelles pratiques et étend le processus modèle CRISP-DM. [13]
- **Inconvénients** : Hérite certains des inconvénients de CRISP-DM, spécialement la négligence de la gestion d'équipes . [13]

Après cette étude comparative, notre choix s'est orienté vers la méthodologie CRISP-DM.

En effet, un facteur important dans l'utilisation de cette approche est qu'elle peut être implémentée dans n'importe quel projet de Data Science, quel que soit son domaine ainsi que l'agilité et l'automatisation des tâches puisque la méthodologie suit une démarche itérative et chaque itération apporte de la connaissance métier supplémentaire qui permet de mieux aborder l'itération suivante.

### 1.3.2 Focus sur la méthodologie CRISP-DM

CRISP-DM, qui signifie Cross-Industry Standard Process for Data Mining, est une méthode mise à l'épreuve sur le terrain permettant d'orienter les travaux d'exploration de données. Elle a été développée par IBM dans les années 60 pour réaliser des projets Data Mining. [14]

Le cycle de vie d'un projet de Data Mining se compose de six phases, illustrées dans la figure 1.8. L'ordre des phases n'est pas rigide. L'alternance entre les différentes phases est toujours nécessaire. Le résultat de chaque phase détermine quelle phase, ou tâche particulière d'une phase, doit être exécutée ensuite. Les flèches indiquent les dépendances les plus importantes et les plus fréquentes entre les phases.

Le cercle extérieur dans la figure 1.8 symbolise la nature cyclique de Data Mining elle-même. La Data Mining ne s'arrête pas une fois qu'une solution est déployée. Les leçons tirées du processus et de la solution déployée peuvent susciter de nouvelles "Business questions", souvent plus ciblées. [14]

Dans ce qui suit, nous décrivons brièvement chaque phase :

#### 1. La compréhension du problème métier

Cette première étape consiste à la compréhension des éléments métiers et problématiques que la Data Science vise à améliorer ou à résoudre.[15]

#### 2. La compréhension des données





La phase de compréhension des données cherche à une détermination précise des données à analyser, à une identification de la qualité des données disponibles et à lier entre les données et leur signification d'un point de vue métier.[15]

### 3. Préparation des données

Cette phase tend à regrouper les activités liées à la construction de l'ensemble précis des données à analyser, faite à partir des données brutes. Elle inclut ainsi le classement des données en fonction de critères choisis, le nettoyage des données, et surtout leur recodage pour les rendre compatibles avec les algorithmes qui seront utilisés.

La paramétrique des données numériques et leur recodage en données catégorielles sont extrêmement importantes et à réaliser avec soin afin d'éviter que les algorithmes utilisés donnent des résultats faux dans la phase suivante. Toutes ces données doivent en effet être centralisées dans une base de données structurée et qui porte le nom de Data Hub.[15]

### 4. La modélisation

C'est la phase de Data Science proprement dite. La modélisation comprend le choix, le paramétrage et le test de différents algorithmes ainsi que leur enchaînement, qui constitue un modèle.

Ce processus est d'abord descriptif pour générer de la connaissance, en expliquant pourquoi les choses se sont passées. Il devient ensuite prédictif en expliquant ce qu'il va se passer, puis prescriptif en permettant d'optimiser une situation future.[15]

### 5. Évaluation

L'évaluation vise à vérifier le(s) modèle(s) ou les connaissances obtenues afin de s'assurer qu'ils répondent aux objectifs formulés au début du processus.

Elle contribue aussi à la décision de déploiement du modèle ou, si besoin est, à son amélioration. A ce stade, on teste notamment la robustesse et la précision des modèles obtenus.[15]

### 6. Le déploiement

Il s'agit de l'étape finale du processus. Elle consiste en une mise en production pour les utilisateurs finaux des modèles obtenus. Son objectif : mettre la connaissance obtenue par la modélisation, dans une forme adaptée, et l'intégrer au processus de prise de décision.



Le déploiement peut ainsi aller, selon les objectifs, de la simple génération d'un rapport décrivant les connaissances obtenues jusqu'à la mise en place d'une application, permettant l'utilisation du modèle obtenu, pour la prédiction de valeurs inconnues d'un élément d'intérêt.[15]

## 1.4 État de l'art

Aborder un sujet typique, tel que la Data Science dans la Fintech n'est pas une tâche simple, mais une analyse de la littérature pourrait faciliter la compréhension de la relation entre les deux domaines, et donner au lecteur une idée de ce que d'autres auteurs voient concernant le sujet présenté. Cette section est ainsi consacrée à la description de l'état de l'art de la Fintech et de la Data Science.

### 1.4.1 Aperçu sur la fintech

D'après Investopedia, la FinTech est définie comme une combinaison des deux termes Finance et Technologie. Il a été utilisé pour identifier des technologies pouvant inclure tout logiciel, algorithme, ordinateur, smartphone, structure, processus ou tout autre moyen pouvant être utilisé pour automatiser des tâches financières dans le but d'améliorer la qualité des services financiers et d'aider les entreprises et les consommateurs à gérer leurs transactions et opérations financières avec efficacité.[16]

La FinTech décrit aujourd'hui de multiples activités financières possibles telles que le dépôt d'un chèque avec son smartphone et les transferts d'argent. On peut également citer l'encouragement à l'investissement dans de nouvelles entreprises ou la gestion des investissements sans l'assistance d'une personne ou l'intervention humaine en utilisant les nouvelles technologies telles que les robots conseillers ou les bots d'assistance par chat. En utilisant certains de ces services FinTech, les consommateurs sont de plus en plus conscients que la FinTech est intégrée implicitement dans leur vie quotidienne et sont en train d'en tirer profit.[16]

### 1.4.2 Focalisation sur le Risk Management et les risques de crédit

La gestion des risques ne repose pas uniquement sur les outils financiers et les indicateurs tels que les taux d'intérêt ou l'évolution économique mais également sur la



situation financière des clients et leur historique.

L'un des risques majeurs que les banques essaient toujours de gérer est le risque de crédit. Le risque de crédit est défini de manière générale comme le risque qu'un prêteur ne puisse pas recouvrer son prêt auprès d'un emprunteur. La diminution du risque lié au crédit permanent d'un emprunteur est également considérée comme un risque de crédit. Cependant, ce genre de situation n'est pas considéré comme un défaut, mais la probabilité de défaut augmentera. [17]

De nombreuses banques et entreprises FinTech estiment que le risque de crédit est l'un des risques importants qu'elles doivent traiter. Cela renvoie à la nature de leur activité qui est basée sur le crédit, ne pas être en mesure de gérer ce risque pourrait entraîner une perte de crédit. La perte de crédit peut survenir lorsqu'un emprunteur est incapable de rembourser ou de rembourser à temps son emprunt.

Dans la plupart des cas, la raison d'un client pourrait être une situation financière difficile ou face à des procédures de faillite. Mais ce n'est pas la seule raison ; il peut aussi s'agir d'un refus de l'emprunteur de respecter ses obligations comme en cas de fraude ou de litige.

Plus de conséquences pourraient survenir ; une perte importante causée par la défaillance des emprunteurs peut conduire à la faillite d'une banque. Il peut même s'agir du déclenchement d'une crise bancaire ou financière. Par conséquent, les banques et les institutions financières accordent une grande importance au suivi et à la gestion des risques, en particulier le risque de défaut pour éviter ce genre de dangers.[18]

### 1.4.3 Aperçu sur la Data Science

La Data Science peut être définie à travers plusieurs perspectives.

La première est une définition qui donne une vision plutôt générique sur le domaine : elle déclare que la Data Science, ou encore la science des données est l'étude et le traitement de données en général qui permet d'en tirer de la valeur.[19]

La seconde est une perspective disciplinaire qui définit la science des données comme un nouveau champ interdisciplinaire qui s'appuie sur la statistique, l'informatique, le traitement, la communication, la gestion et la sociologie, où l'objectif est d'étudier la donnée et ses environnements, en d'autres termes, c'est étudier les données et les aspects contextuels qui lui sont relatifs et qui aident à les comprendre. Ce processus permet de transformer les données en des idées et connaissances suivant une méthodologie spécifique ce qui assure l'analyse décisionnelle.[19]

En effet, les domaines d'application de la data science sont innombrables, elle est



devenu présente dans le domaine de la santé, le marketing, l'industrie, le transport, l'aéronautique et dans le domaine de la Banque et assurance également.

En effet, les banques mettent à profit la Data Science pour mieux satisfaire les besoins de leurs clients et améliorer la gestion des risque.

Afin de mieux connaître et fidéliser ses clients, le secteur bancaire a fait recours à l'élaboration des process de marketing prédictif afin de mieux connaître les habitudes de consommation des clients et le ciblage de publicités proposés aux clients.

Mieux connaître ses clients permet également aux banques de se protéger contre les transactions frauduleuses et les défauts de paiement.[20]

#### **1.4.4 Application de la Data Science à la Fintech et au risk management**

Une énorme quantité de données est générée par les institutions financières, notamment les banques. Ces données peuvent être analysées et fournissent ainsi des connaissances aux entreprises. Dans le même contexte, les techniques de la Data Science peuvent être appliquées par les entreprises et les banques pour renforcer la révolution du FinTech.

Le risque de crédit est connu comme une perte économique qui survient lorsqu'une contrepartie ne remplit pas ses obligations contractuelles (par exemple, le paiement du prêt ou des intérêts), ou lorsque le risque de défaut augmente pendant la durée de la transaction.

Les méthodes traditionnelles de modélisation du risque de crédit reposent sur des régressions linéaires et logit classiques. Néanmoins, ces techniques ne sont pas les meilleures puisque l'apprentissage automatique et l'Intelligence Artificielle en général ont une capacité plus importante à le modéliser en raison de la compréhension sémantique des données non structurées.[21]

Le risque de crédit est un élément essentiel dans la procédure de prêt commercial. Une banque ne pourra pas faire une analyse objective des clients potentiels, et du prix des prêts sans l'évaluation de ce risque. Ce risque de défaut est souvent mesuré et évalué avec un score de crédit. Les modèles de notation ou de scoring de crédit sont construits sur la base de différents indices qui décrivent les multiples aspects de la situation financière de l'emprunteur. Ces modèles sont bénéfiques pour les institutions financières et sont utilisés pour classer les demandeurs de crédit soit dans un « bon profil de demandeur » qui est susceptible de rembourser, soit dans un « mauvais profil de demandeur » qui a une forte probabilité de ne pas rembourser le prêt.[21]



## Conslusion

Dans ce premier chapitre, nous avons présenté l'organisme d'accueil. Ensuite, nous avons décrit le cadre du projet dans son contexte général ainsi que les objectifs majeurs à prendre en compte et nous avons également évoqué le choix de la méthodologie adoptée. Enfin, nous avons passé en revue le domaine d'application de notre projet, il s'agit de la description de l'état de l'art de la Fintech, le risk management, les risques de crédit et la Data Science.

# Chapitre 2

## Compréhension et préparation des données



## Introduction

Ce chapitre est consacré à la compréhension et la préparation des données. Dans cette étape nous allons commencer par identifier la source des données, nous enchaînons par une analyse exploratoire des données (EDA) et nous finissons par présenter les actions de pré-traitements appliquées.

### 2.1 Identification de source des données

Pour la réalisation de notre PoC, nous avons opté pour des données disponibles sur "Kaggle", fournis par "Home Credit Group".<sup>1</sup>, qui mentionne clairement dans sa licence, que l'utilisation du DataSet est autorisée pour la recherche et l'éducation universitaires, et à d'autres usages non commerciaux.

#### 2.1.1 Compréhension de la structure des données

Comme tout problème de Machine Learning, le risque débiteur comporte des facteurs, ou des caractéristiques liées au client, qui a une relation avec la probabilité d'un défaut.

L'identification et la mesure de ces facteurs sont essentielles pour prévoir les défauts de paiement.

À cet égard, les données sont provenues de plusieurs sources de données. Il s'agit d'un ensemble de fichiers sous format CSV présentant des données statiques et d'autres transactionnelles .









 application_test.csv	31/03/2022 10:04	Microsoft Excel Co...	25 945 Ko
 application_train.csv	31/03/2022 10:04	Microsoft Excel Co...	162 240 Ko
 bureau.csv	31/03/2022 10:04	Microsoft Excel Co...	166 032 Ko
 bureau_balance.csv	31/03/2022 10:04	Microsoft Excel Co...	366 790 Ko
 credit_card_balance.csv	31/03/2022 10:04	Microsoft Excel Co...	414 632 Ko
 installments_payments.csv	31/03/2022 10:04	Microsoft Excel Co...	706 171 Ko
 POS_CASH_balance.csv	31/03/2022 10:04	Microsoft Excel Co...	383 500 Ko
 previous_application.csv	31/03/2022 10:04	Microsoft Excel Co...	395 482 Ko

FIGURE 2.1 – La structure du DataSet

1. Il s'agit d'une institution financière internationale non bancaire fondée en 1997 en République tchèque et basée aux Pays-Bas. La société opère dans 9 pays et se concentre sur les prêts à tempérament principalement aux personnes ayant peu ou pas d'antécédents de crédit.[22]



## 2.1.2 La forme des tables du DataSet

Le diagramme de la structure du DataSet suivant montre les interrelations entre les fichiers de données fournis.

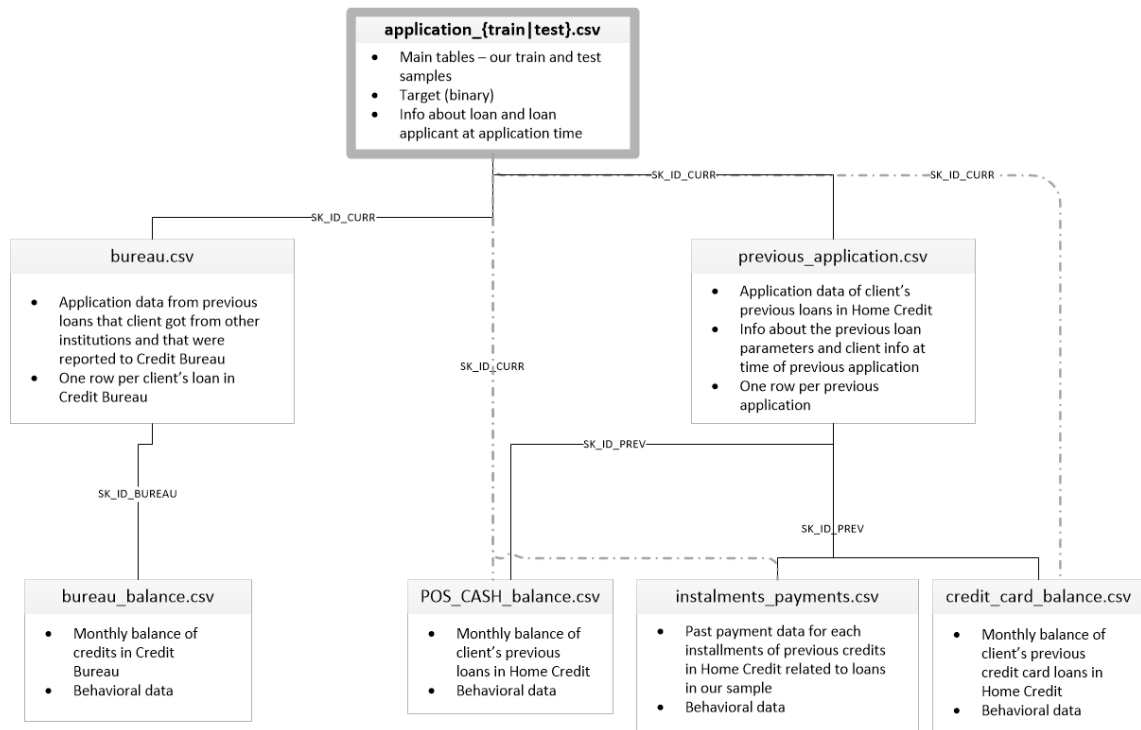


FIGURE 2.2 – Le diagramme de la structure du DataSet

- **Application {train|test}.csv** : C'est la table main, divisée en deux fichiers pour le train (avec la valeur cible) et pour le test (sans la valeur cible). Il s'agit de données statiques de toutes les demandes de crédit, chaque ligne représente les caractéristiques liées à une demande de prêt.
- **Bureau.csv** : Il s'agit de données de tous les prêts précédents du client fournis par d'autres institutions financières avant la date de sa demande de crédit.
- **bureau balance.csv** : Chaque ligne représente un mois de chaque crédit antérieur du client, ainsi un seul ancien crédit peut avoir plusieurs lignes, une pour chaque mois de la durée du crédit.
- **POS CASH balance.csv** : Des aperçus des soldes mensuels des prêts POS (point of sales) et cash précédents que le demandeur de crédit a eu avec Home Credit.
- **credit card balance.csv** : Ce fichier représente des aperçus des soldes mensuels des cartes de crédit précédentes que le client possède chez Home Credit.





- **previous application.csv** : Il s'agit de toutes les demandes précédentes de prêts pour un crédit immobilier des clients qui ont des prêts dans l'échantillon de données de Home Credit.
- **installments payments.csv** : C'est l'historique de remboursement des crédits précédemment payés dans le cadre de l'institution Home Credit et liés aux prêts de leur échantillon de données.

### 2.1.3 La forme des tables du DataSet

La figure ci-dessous illustre le output du code de " DataFrame shape " qui nous montre un aperçu de nombre de features et de lignes de chaque table de notre DataSet :

```
La table application_train a : 307511 lignes avec 122 features
-----
La table application_test a : 48744 lignes avec 121 features
-----
La table previous_application a : 1670214 lignes avec 37 features
-----
La table POS_CASH_balance a : 10001358 lignes avec 8 features
-----
La table bureau a : 1716428 lignes avec 17 features
-----
La table bureau_balance a : 27299925 lignes avec 3 features
-----
La table credit_card_balance a : 3840312 lignes avec 23 features
-----
La table installments_payments a : 13605401 lignes avec 8 features
```

FIGURE 2.3 – Le nombre de ligne et de features dans chaque table

## 2.2 Analyse exploratoire des données (EDA)

L'analyse exploratoire des données est une tâche effectuée par un data scientist pour se familiariser avec les données. Toutes les tâches initiales que nous effectuons pour bien comprendre nos données sont connues sous le nom d'EDA.

L'objectif principal de l'EDA est de réaliser une analyse statistique afin de déduire des hypothèses qui nous permettent de nous préparer à une modélisation plus avancée.

En effet, toutes les tables que nous possédons comptent ensemble plus de 212 variables.



À cet égard, visualiser et analyser chacune d'entre elles ne créerait aucune valeur et rendrait la lecture difficile. Par conséquent, seul un petit ensemble de variables sera sélectionné et discuté dans cette section. Mais la sélection des variables n'est pas arbitraire, tout d'abord, nous allons nous concentrer essentiellement sur les données statiques de notre dataset. Ensuite, la sélection sera basée sur l'importance de cette variable par rapport à la variable cible et sur ce qui semble être utile à comprendre les données et le problème métier sur la base de l'intuition et l'expertise du domaine.

### 1. La distribution du statut de remboursement des prêts (la valeur cible)

Le statut de remboursement des prêts est la variable cible dans notre Dataset. Selon la figure 2.4, il est bien clair qu'on est face à un Dataset Imbalanced. Les clients qui sont capables de rembourser le prêt sont plus de 250000, 91.98% des clients, (Encodé 0) et ceux qui ont des difficultés à rembourser le prêt sont moins de 50000, 8.1 % des clients (Encodé 1).

En effet, un dataset déséquilibré peut influencer négativement l'algorithme d'apprentissage, étant donné que la plupart des algorithmes automatiques fonctionnent mal avec un tel dataset, ce qui peut mener dans notre cas à ignorer complètement la classe minoritaire et causera par conséquent une grande perte pour la banque.

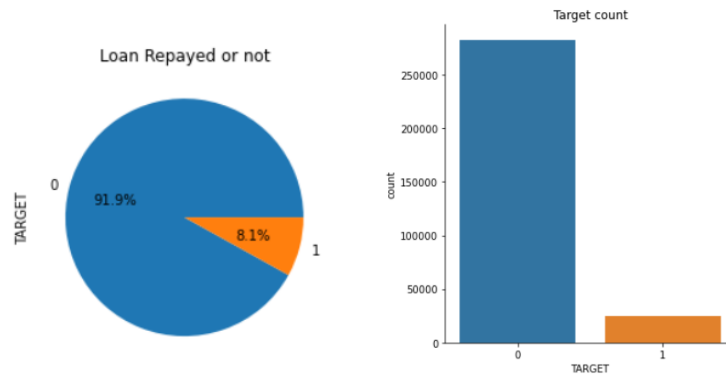


FIGURE 2.4 – La distribution du statut de remboursement des prêts

### 2. La distribution des revenus totaux des clients

Le `AMT_INCOME_TOTAL` est la variable présentant le revenu total des clients.

Le premier graphique de la figure 2.5 montre que le revenu des deux types de clients est inférieur à 200 millions sauf pour un client qui a un revenu de 117 millions et qui est un défaillant. Si nous le supprimons et traçons un graphique de "`AMT_INCOME_TOTAL < (0.2X1e8)`" (représentant un focus sur la distribution des clients ayant un revenu inférieur à 200 millions), nous avons supprimé la valeur aberrante qui est bien claire dans le boxplot présenté



par la figure 2.6, mais nous ne pouvons pas clairement déterminer si la majorité des deux classes se chevauchent ou non. Après avoir tracé le graphique  $AMT\_INCOME\_TOTAL < (1 \times 10^6)$  (représentant un focus sur la distribution des clients ayant un revenu inférieur à 1 millions), il est bien clair que les valeurs de la majorité sont fortement 'Overlapping' (C'est à dire qu'une grande partie des populations couvrent la même gamme de valeurs) pour les deux classes, comme le montre le dernier graphique de la figure 2.5. A cet égard, cette colonne n'est pas très utile.

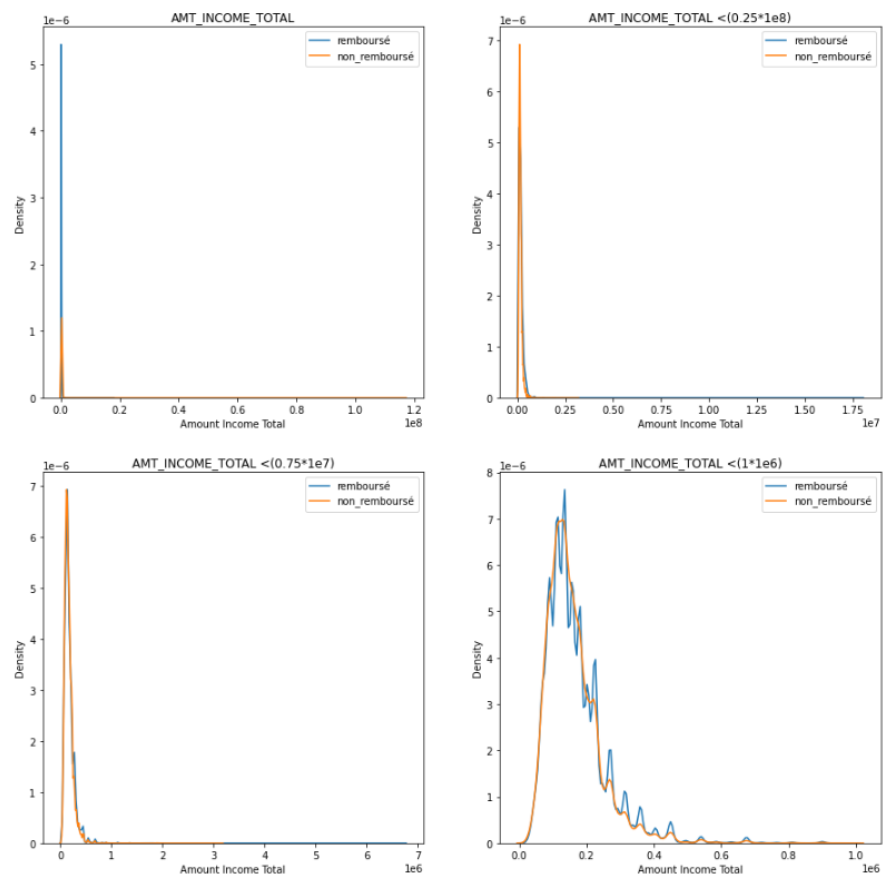


FIGURE 2.5 – La distribution des revenus totaux des clients

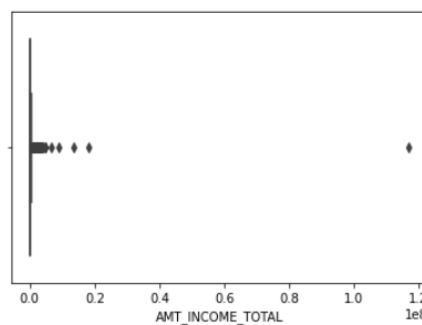


FIGURE 2.6 – Boxplot de la variable AMT\_INCOME



### 3. La ditribution des Motants de crédits

Le AMT\_CREDIT est le montant de crédit demandé par le client, comme illustré par la figure ci-dessous, il est bien clair que la densité des montants inférieurs à 1 million pour les deux types de clients (qui ont remboursé leurs prêts et qui sont des défaillants) et que le graphique est asymétrique à gauche c'est-à-dire que les petites valeurs observées sont plus fréquentes que les valeurs plus élevées. Il est également remarquable que les données se chevauchent fortement entre les deux classes. Ce qui implique que la valeur du montant de crédit demandé n'est pas très utile.

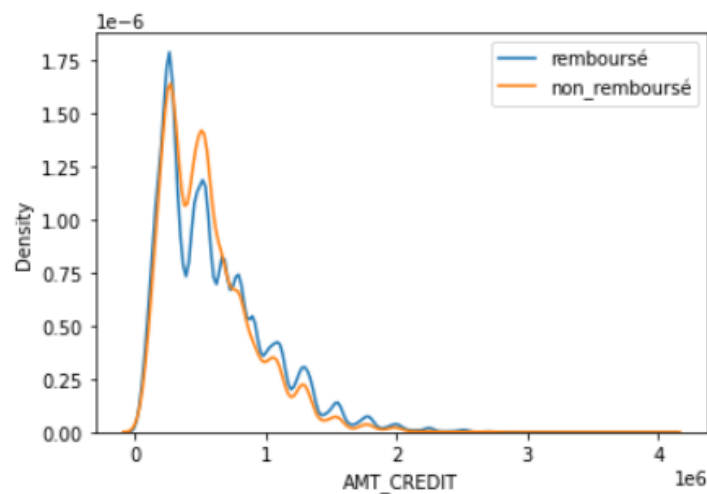


FIGURE 2.7 – La ditribution des Motants des crédits

### 4. La ditribution de nombre d'années pendant lesquelles une personne a travaillé

Cette variable représente le nombre d'années qu'un client a travaillé. Après l'avoir plotté, nous remarquons qu'il y a quelques clients qui travaillent pendant 1000 années qui sont évidemment des valeurs aberrantes, comme détaillé dans la figure 2.9.

Nous avons donc supprimé la valeur aberrante 365243, et nous remarquons alors que le nombre maximum d'années est 50 ans. Il est également bien clair que les clients ayant moins de 10 ans d'expérience ont eu des difficultés à rembourser leurs prêts comme illustré dans la figure ci-dessous.

Nous pouvons alors conclure que cette colonne serait utile.

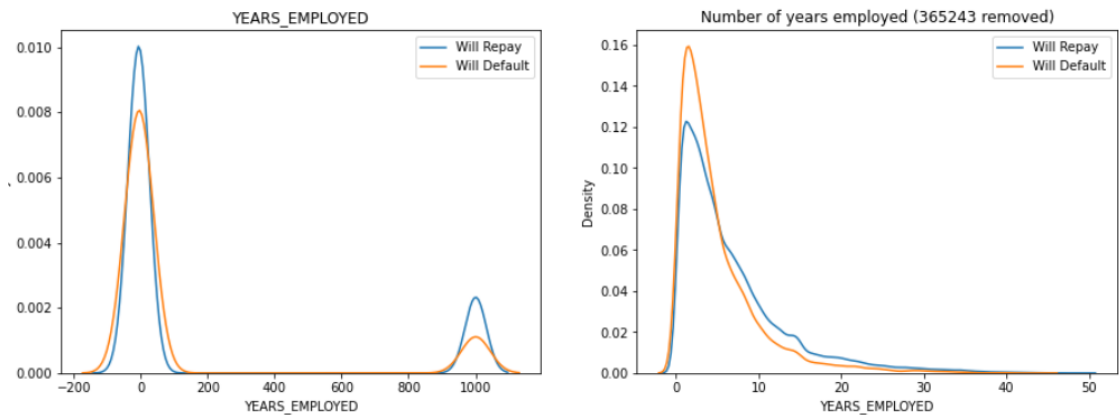


FIGURE 2.8 – La distribution de nombre d’années pendant lesquelles une personne a travaillé

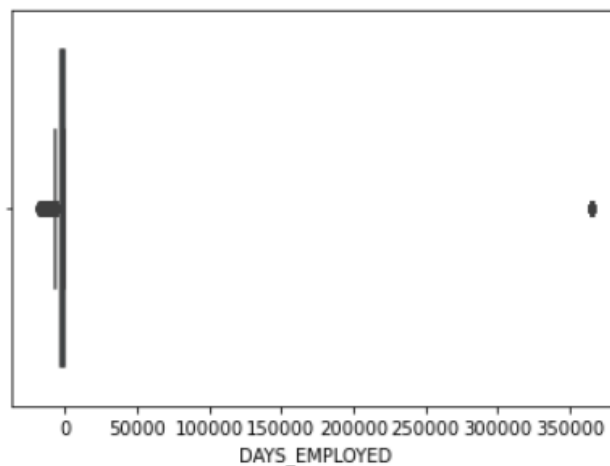


FIGURE 2.9 – Boxplot de la variable Days\_Employed

## 5. Analyse de la variable âge des clients

D’après le traçage de l’âge des clients en années comme montré dans la figure ci-dessous, nous pouvons conclure que la majorité de ceux âgés de 20 à 40 ans ont eu une difficulté à rembourser leurs prêts.

À mesure que l’âge augmente, nous constatons que la majorité de groupe d’âge (50 à 70 ans) ont remboursé leurs prêts.

Nous pouvons déduire que cette colonne est utile.

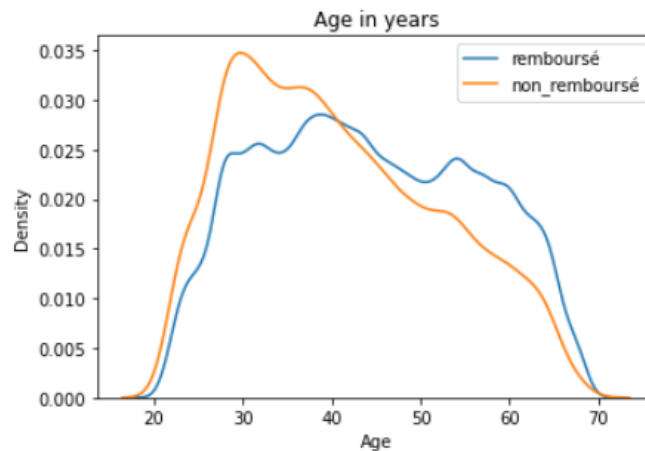


FIGURE 2.10 – Boxplot de la variable Days\_Employed

## 6. Analyse de la variable genre des clients

D'après le diagramme de la figure ci-dessous, on remarque que le pourcentages des clients de genre Femme et plus élevé que ceux de genre Homme, et que pour les deux genres la densité de ceux qui ont remboursé leur prêt est élevé mais mais cela pourrait être dû au déséquilibre des données. Nous constatons également la présence d'une valeur 'XNA' qui est inutile.

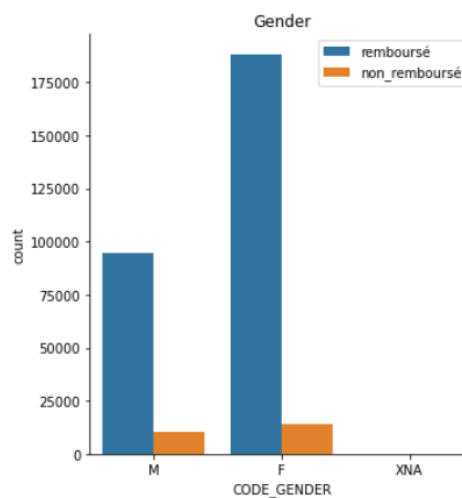


FIGURE 2.11 – Analyse de la variable genre des clients

## 7. Analyse des variables EXT\_SOURCE\_1, EXT\_SOURCE\_2, EXT\_SOURCE\_3

Ces variables représentent un score normalisé de sources de données externes. D'après les subplots définis par la figure ci-dessous on peut déduire quelques hypothèses :



- Si External source 1  $< 0.4$  alors target=1 et si (External source 1  $> 0.4$  et External source 1  $< 0.1$ ) target=0.
- Si External source 2  $< 0.4$  alors target=1 et (External source 2  $> 0.4$  et External source 2  $< 0.8$ ) target=0.
- Si External source 3  $< 0.4$  alors target=1 et (External source 3  $> 0.4$  et External source 3  $< 0.1$ ) alors target=0.

Nous pouvons en conclure alors qu'il y a une séparation visible entre les deux classes, ainsi que ces valeurs indiquent une colinéarité négative avec la variable cible vu qu'elles sont faibles pour la classe 1 et élevés pour la classe 0.

Ces variables sont alors très utiles.

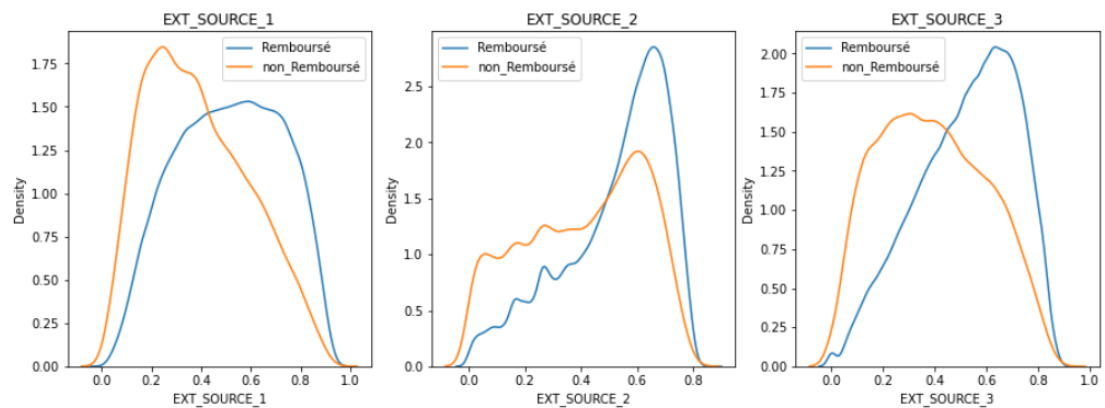


FIGURE 2.12 – Analyse des variables EXT\_SOURCE\_1, EXT\_SOURCE\_2, EXT\_SOURCE\_3

## 8. Aperçu des différentes tables

En utilisant le module 'Pandas Profiling'<sup>2</sup> de Python, nous analysons ci-dessous l'état des différentes tables de notre Datamart final.

- **Analyse de la table 'Application\_train'**

Comme détaillé dans la figure ci-dessous, la table Application\_train comporte 123 features (dont 70 variables numériques, 49 variables catégorielles et 3 variables booléennes) et 307 511 lignes, avec 24.4% de valeurs manquantes.

---

2. Pandas profiling est un module Python open source grâce auquel nous pouvons rapidement faire un EDA avec seulement quelques lignes de code. Il génère également des rapports interactifs au format web qui peuvent être présentés à toute personne, même si elle n'est pas spécialisée dans le domaine de l'informatique.[23]



Dataset statistics		Variable types	
Number of variables	122	Numeric	70
Number of observations	307511	Categorical	49
Missing cells	9152465	Boolean	3
Missing cells (%)	24.4%		
Duplicate rows	0		
Duplicate rows (%)	0.0%		
Total size in memory	536.7 MiB		
Average record size in memory	1.8 KiB		

FIGURE 2.13 – Analyse de la table 'Application\_train'

- **Analyse de la table 'Bureau'**

Comme illustré dans la figure ci-dessous, la table Bureau comporte 17 features (dont 14 variables numériques et 3 variables catégorielles) et 1 716 428 lignes, avec 13.5% de valeurs manquantes.

Dataset statistics		Variable types	
Number of variables	17	Numeric	14
Number of observations	1716428	Categorical	3
Missing cells	3939947		
Missing cells (%)	13.5%		
Duplicate rows	0		
Duplicate rows (%)	0.0%		
Total size in memory	512.1 MiB		
Average record size in memory	312.9 B		

FIGURE 2.14 – Analyse de la table 'Bureau'

- **Analyse de la table 'Bureau\_Balance'**

Comme illustré dans la figure ci-dessous, la table Bureau\_Balance comporte 3 features (dont 2 variables numériques et 1 variable catégorielle) et 27 299 925 lignes, avec 0% de valeurs manquantes.

Dataset statistics		Variable types	
Number of variables	3	Numeric	2
Number of observations	27299925	Categorical	1
Missing cells	0		
Missing cells (%)	0.0%		
Duplicate rows	0		
Duplicate rows (%)	0.0%		
Total size in memory	1.9 GiB		
Average record size in memory	74.0 B		

FIGURE 2.15 – Analyse de la table 'Bureau\_Balance'

- **Analyse de la table 'POS\_CASH\_Balance'**

Comme illustré dans la figure ci-dessous, la table POS\_CASH\_Balance comporte 8 features (dont 7 variables numériques et 1 variable catégorielle) et 10 001 358 lignes, avec 0.1% de valeurs manquantes.





Dataset statistics		Variable types	
Number of variables	8	Numeric	7
Number of observations	10001358	Categorical	1
Missing cells	52158		
Missing cells (%)	0.1%		
Duplicate rows	0		
Duplicate rows (%)	0.0%		
Total size in memory	1.1 GiB		
Average record size in memory	119.2 B		

FIGURE 2.16 – Analyse de la table 'POS\_CASH\_Balance'

- **Analyse de la table 'Credit\_Card\_Balance.'**

Comme illustré dans la figure ci-dessous, la table Credit\_Card\_Balance comporte 8 features (dont 2 variables numériques et 1 variable catégorielle) et 10 001 358 lignes, avec 0.1% de valeurs manquantes.

Dataset statistics		Variable types	
Number of variables	23	Numeric	22
Number of observations	3840312	Categorical	1
Missing cells	5877356		
Missing cells (%)	6.7%		
Duplicate rows	0		
Duplicate rows (%)	0.0%		
Total size in memory	875.7 MiB		
Average record size in memory	239.1 B		

FIGURE 2.17 – Analyse de la table 'Credit\_Card\_Balance.'

- **Analyse de la table 'Previous\_Application'**

Comme illustré dans la figure ci-dessous, la table Previous\_Application comporte 37 features (dont 19 variables numériques et 17 variable catégorielle et 1 variable booléenne) et 1 670 214 lignes, avec 0% de valeurs manquantes.

Dataset statistics		Variable types	
Number of variables	37	Numeric	19
Number of observations	1670214	Categorical	17
Missing cells	11109336	Boolean	1
Missing cells (%)	18.0%		
Duplicate rows	0		
Duplicate rows (%)	0.0%		
Total size in memory	1.9 GiB		
Average record size in memory	1.2 KiB		

FIGURE 2.18 – Analyse de la table 'Previous\_Application'

- **Analyse de la table 'Installments\_Payments'**

Comme illustré dans la figure ci-dessous, la table Installments\_Payments comporte 8 features (8 variables numériques ) et 13 605 401 lignes, avec 0% de valeurs manquantes.



Dataset statistics		Variable types	
Number of variables	8	Numeric	8
Number of observations	13605401		
Missing cells	5810		
Missing cells (%)	< 0.1%		
Duplicate rows	0		
Duplicate rows (%)	0.0%		
Total size in memory	830.4 MiB		
Average record size in memory	64.0 B		

FIGURE 2.19 – Analyse de la table 'Installments\_Payments'

## 2.3 Préparation des données

Dans cette phase de préparation des données, un certain nombre d'actions de pré-traitements seront appliquées à chaque table de notre ensemble de données, et aux données fusionnées par la suite, afin de permettre aux algorithmes de prédiction d'éviter des résultats erronés.

Cette tâche comprendra quelques activités clés, tel que :

- **Suppression des valeurs aberrantes**
- **Feature Engineering**
- **Transformation des données**
- **Intégration des données**
- **Nettoyage des données**
- **Division des données**
- **Echantillonnage aléatoire**

La forme finale des données sera introduite dans le modèle de prédiction de la phase suivante.

### 2.3.1 Suppression des valeurs aberrantes

Comme expliqué dans la partie EDA de la section précédente, nous avons extrait quelques valeurs aberrantes que nous avons opté pour leur suppression.

### 2.3.2 Feature Engineering

Il s'agit de la création de nouvelles variables prédictives à partir de variables existantes.



Cette tâche consiste à enrichir les données d'apprentissage de features supplémentaires pour rendre les algorithmes plus performants, ce qui en fait une activité essentielle dans le processus de Machine Learning.

En effet, certaines de nouvelles features qui sont conçues pourraient nous fournir de nouvelles informations précieuses sur les clients, ce qui aidera le modèle à prédire des valeurs plus précises pour la variable cible.

Un exemple de features que nous avons créé à partir de la table de demandes de crédits :

- La proportion du crédit par rapport au revenu total des clients

$$df['INCOME\_PER\_CREDIT'] = \frac{df['AMT\_INCOME\_TOTAL']}{df['AMT\_CREDIT']}$$

- La proportion de revenu du client par rapport aux membres de famille

$$df['INCOME\_PER\_PERSON'] = \frac{df['AMT\_INCOME\_TOTAL']}{df['CNT\_FAM\_MEMBERS'] + 1}$$

- Montant payé annuellement par rapport au montant crédité

$$df['PAYMENT\_RATE'] = \frac{df['AMT\_ANNUITY']}{df['AMT\_CREDIT']}$$

- Montant payé pour la demande de prêt précédente mensuellement selon le nombre de jours employés

$$df['ANNUITY\_DAYS\_EMPLOYED\_PERC'] = \frac{df['DAYS\_EMPLOYED']}{df['AMT\_ANNUITY']}$$

$$df['AMT\_CREDIT\_DAYS\_EMPLOYED'] = \frac{df['DAYS\_EMPLOYED']}{df['AMT\_CREDIT']}$$

- Pourcentage de vie de client passé à travailler

$$df['DAYS\_WORKING\_PER'] = \frac{df['DAYS\_EMPLOYED']}{df['DAYS\_BIRTH']}$$




### 2.3.3 Transformation des données

#### 1. Traitement des variables catégorielles

En effet, les algorithmes de Machine Learning ne traitent pas des variables qualitatives catégorielles, et comme notre dataset représente un grand nombre de variables catégorielles nous devons procéder à une transformation de ces variables pour pouvoir les exploiter.

À cet égard, nous avons opté pour l'encodage des ces données en utilisant la méthode de **One Hot Encoding**, qui permet un simple encodage des valeurs qualitatives sans biaiser l'information de la donnée par une relation d'ordinalité inexistante. Il s'agit de représenter chaque variable catégorielle par un vecteur binaire comportant un élément pour chaque label unique et de marquer le label de classe par un '1' et tous les autres éléments par un '0'.



NAME_FAMILY_STATUS		NAME_FAMILY_STATUS_Married	NAME_FAMILY_STATUS_Separated	NAME_FAMILY_STATUS_Single / not married	NAME_FAMILY_STATUS_Widow
0	Single / not married	0	0	1	0
1	Married	1	0	0	0
2	Single / not married	0	0	1	0
3	Civil marriage	0	0	0	0
4	Single / not married	0	0	1	0
...	...	...	...	...	...
307506	Separated	0	0	0	1
307507	Widow	1	0	0	0
307508	Separated	1	0	0	0
307509	Married	1	0	0	0
307510	Married	1	0	0	0

FIGURE 2.20 – Un échantillon du résultat de l'encodage de la variable 'NAME\_FAMILY\_STATUS'

#### 2. Traitement des variables numériques

- Agrégation des données

Le terme d'agrégation désigne la combinaison de deux ou plusieurs attributs (ou objets) en un seul. L'objectif de l'agrégation est de réduire le nombre d'objets ou d'attributs.

L'agrégation de nos données a été spécifiée pour chaque table puisqu'il existe plusieurs prêts pour l'Id de chaque demandeur.

Par exemple, dans les tables 'Bureau' et 'Bureau\_balance', nous avons pour chaque Id de prêt (par exemple, SK\_ID\_CURR) un sous-ensemble de lignes, chaque ligne est un ancien prêt qui est lié au prêt actuel par un Id (par exemple, SK\_ID\_BUREAU).



L'agrégation des variables numériques a été donc faite pour analyser la moyenne, la somme, les valeurs maximales et minimales des prêts de chaque Id client spécifique. En utilisant ces agrégations, de nouvelles colonnes seront créées, comme le montre la figure ci-dessous.

SK_ID_CURR	AMT_CREDIT		PREVA_AMT_CREDIT_MEAN	PREVA_AMT_CREDIT_MIN	PREVA_AMT_CREDIT_MAX	
0	100002	406597.5	0	179055.0000	179055.000	179055.0
1	100003	1293502.5	1	484191.0000	68053.500	1035882.0
2	100004	135000.0	2	20106.0000	20106.000	20106.0
3	100006	312682.5	3	343728.9000	24219.000	675000.0
4	100007	513000.0	4	166638.7500	14616.000	284400.0
...	...	...	...	...	...	...
307506	456251	254700.0	356245	254700.0000	254700.000	254700.0
307507	456252	269550.0	356246	98704.1250	26788.500	208854.0
307508	456253	677664.0	356247	132516.8325	111295.665	153738.0
307509	456254	370107.0	356248	247171.5000	40468.500	453874.5
307510	456255	675000.0	356249	158396.6250	66091.500	239850.0

FIGURE 2.21 – Un échantillon du résultat de l'agrégation de la variable AMT\_CREDIT'

- Normalisation des données (StandardScaler)

La normalisation est une méthode permettant de réduire la complexité des modèles. Ce qui peut permettre aux algorithmes de s'entraîner plus rapidement et d'éviter la saturation, pour la normalisation de nos données nous avons opté pour le StandardScaler.

Le StandardScaler suppose que les données sont normalement distribuées dans chaque élément et les met à l'échelle de telle sorte que la distribution devient centrée autour de 0, avec un écart type de 1. La moyenne et l'écart type sont calculés pour le feature, puis ils sont mise à l'échelle en fonction de :

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

$\mu$  : *Mean*

$\sigma$  : *StandardDeviation*

### 2.3.4 Intégration des données

Cette étape consiste à combiner les données de différentes sources pour obtenir une structure unifiée avec des informations plus précieuses et plus significatives.



En effet, les tables 'application\_train' et 'application\_test' ont les mêmes informations sur les demandes actuelles, nous avons donc choisi de les fusionner à l'aide de la fonction 'append()'. L'ensemble des données sur les demandes précédentes contient des informations sur les anciens prêts qui sont liés aux prêts actuels, et toutes les autres tables sont également liées aux données principales présentées par la table 'Application\_{train|test}'.

Nous avons donc fusionné les différentes dataframe pandas à l'aide de la fonction `merge()`.

### 2.3.5 Nettoyage des données

L'objectif de cette étape, consiste à avoir un dataset cohérent avec les autres données similaires dans l'ensemble de données.

En effet, les incohérences détectées peuvent être initialement causées par des erreurs de saisie par l'utilisateur, par une corruption au cours de la transmission ou le stockage des données ou par un manque de données à saisir.

Dans nos données, nous sommes face à deux problèmes, les valeurs infinies et les valeurs manquantes.

#### 1. Traitement des valeurs infinies

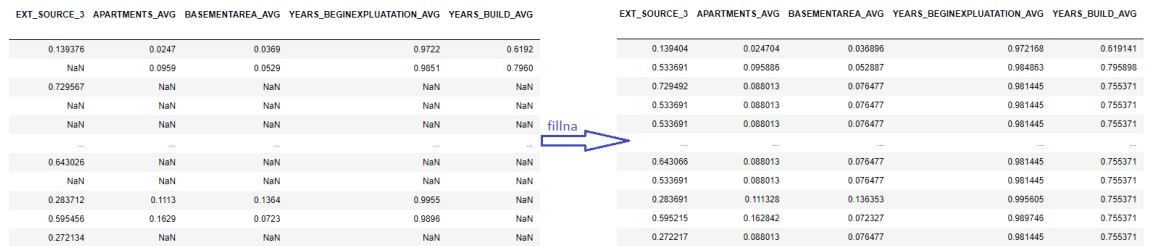
Étant donné que, les algorithmes de machine learning ne peuvent pas traiter les valeurs infinies de la même manière, il est essentiel de prendre cela en considération lors de la construction des modèles.

La première étape de nettoyage des données consiste donc à traiter les valeurs infinies. Le processus de correction des valeurs infinies est assez simple, nous avons simplement remplacé ses valeurs par la valeur maximale.

#### 2. Traitement des valeurs manquantes

Comme notre dataset contient un grand nombre de valeurs manquantes, la suppression des colonnes ayant beaucoup de valeurs manquantes peut impacter négativement la performance des modèles, la suppression des lignes avec des valeurs manquantes pourrait avoir le même effet, car la taille du dataset serait trop petite.

Afin d'éviter ce genre de problèmes, nous avons décidé de les remplacer par la valeur médiane .



The diagram illustrates the process of imputing missing values in a dataset. It shows two side-by-side tables. The left table represents the original dataset with missing values (NaN) in the 'YEARS\_BUILD\_AVG' column. The right table shows the result after applying the 'fillna' function, where the missing values have been replaced with a specific value (0.619141). A blue arrow labeled 'fillna' points from the left table to the right table.

EXT_SOURCE_3	APARTMENTS_AVG	BASEMENTAREA_AVG	YEARS_BEGINEXPLUATION_AVG	YEARS_BUILD_AVG
0.139376	0.0247	0.0369	0.9722	0.6192
NaN	0.0959	0.0529	0.9851	0.7960
0.729567	NaN	NaN	NaN	NaN
NaN	NaN	NaN	NaN	NaN
NaN	NaN	NaN	NaN	NaN
...	...	...	...	...
0.643026	NaN	NaN	NaN	NaN
NaN	NaN	NaN	NaN	NaN
0.283712	0.1113	0.1364	0.9955	NaN
0.595456	0.1629	0.0723	0.9896	NaN
0.272134	NaN	NaN	NaN	NaN

EXT_SOURCE_3	APARTMENTS_AVG	BASEMENTAREA_AVG	YEARS_BEGINEXPLUATION_AVG	YEARS_BUILD_AVG
0.139404	0.024704	0.036896	0.972168	0.619141
0.533691	0.095886	0.052887	0.984863	0.795898
0.729492	0.088013	0.076477	0.981445	0.755371
0.533691	0.088013	0.076477	0.981445	0.755371
0.533691	0.088013	0.076477	0.981445	0.755371
...	...	...	...	...
0.643066	0.088013	0.076477	0.981445	0.755371
0.533691	0.088013	0.076477	0.981445	0.755371
0.283691	0.111328	0.136353	0.995605	0.755371
0.595215	0.162842	0.072327	0.989746	0.755371
0.272217	0.088013	0.076477	0.981445	0.755371

FIGURE 2.22 – Un échantillon du résultat de l'imputation des valeurs manquantes

### 2.3.6 Division des données

Cette étape consiste à diviser le jeu de données en deux parties, une pour l'entraînement du modèle et l'autre pour le test. La première partie des données est consacré pour développer un modèle prédictif, et l'autre pour évaluer ses performances.

À cet égard, nous avons spécifié un ensemble de données d'entraînement de 70% et alloué la partie restante qui présente 30% pour le test en utilisant la fonction 'train\_test\_split' de la bibliothèque 'Scikit-Learn'.

### 2.3.7 Échantillonnage aléatoire

Comme cité dans la section de l'EDA, ce projet nous a mis face à un dataset dés-équilibré où la distribution des classes est fortement asymétrique, ce qui peut influencer l'algorithme d'apprentissage, conduisant à ignorer complètement la classe minoritaire. Il s'agit d'un problème car c'est la classe minoritaire sur laquelle les prédictions sont les plus importantes ,si une demande de crédit risqué est prédite comme non risquée, la conséquence peut être très mauvaise pour la banque.

Ainsi, nous avons opté pour le module 'Random Over Sampling' de la bibliothèque 'Scikit-Learn' qui consiste à dupliquer des exemples de la classe minoritaire. En effet, le sur-échantillonnage de tout l'ensemble de données peut provoquer la présence de mêmes observations à la fois dans les ensembles de test et de l'entraînement, ce qui peut permettre au modèle d'apprendre simplement des points de données spécifiques et entraîner un overfitting et une mauvaise généralisation aux données de test.

À cet égard nous avons appliqué le OverSampling uniquement sur les données d'entraînement afin de l'équilibrer comme détaillé dans la figure ci-dessous.

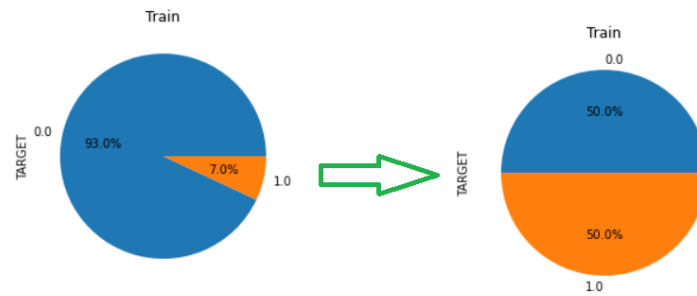


FIGURE 2.23 – Résultat de l'Oversampling sur les données d'entraînement

## Conclusion

Au cours de ce chapitre nous avons présenté la source de nos données, nous avons également fait une EDA qui nous a permis de se familiariser avec les données, et finalement nous les avons préparées. Nous pouvons donc entamer la phase de "modélisation et évaluation".



# Chapitre 3

## Modélisation et évaluation



## Introduction

Dans ce chapitre nous allons aborder la partie de la modélisation et l'évaluation dans le processus CRISP-DM. Nous allons sélectionner plusieurs algorithmes, les construire et régler leurs hyperparamètres afin d'évaluer leurs performances.

### 3.1 Modélisation

Cette phase de modélisation consiste à la sélection de plusieurs algorithmes, leur construction ainsi que le réglage de leurs hyperparamètres.

#### 3.1.1 Classification supervisée

Dans cette partie nous allons détailler un ensemble d'algorithmes de classification supervisée que nous avons utilisé pour notre problème de défaut de paiement.

Comme notre domaine d'application se focalise sur le secteur bancaire qui s'agit d'un secteur fortement réglementé, nous avons opté tout d'abord pour des algorithmes qui sont directement interprétables tel que la **régression logistique**, l'**arbre de décision** et le **Random Forest**, nous avons également fait recours aux deux algorithmes qui sont surpuissants pour les jeux de données déséquilibrés : le **XGboost** et le **Lightgbm** et finalement nous avons choisi de construire un réseau de neurones (ANN) afin d'évaluer sa performance pour notre problème.

Tandis que les performances des modèles peuvent être fortement dépendantes du choix des hyperparamètres, nous avons fait un «**HyperParameter Tuning**», c'est à dire un ajustement des hyperparamètres de chaque modèle, à l'aide de la fonction **RandomizedSearchCV()** du module **sickit-learn** ainsi que le module **Hyperopt**, qui nous ont permis d'avoir un baseline sur lequel nous avons fait des ajustements individuels pour quelques paramètres afin de mieux optimiser les performances des différents modèles.

#### 1. Méthode mathématique : Régression logistique

La régression logistique est un algorithme mathématique supervisé de classification qui est l'équivalent de la régression linéaire.[6]

Étant donné que le modèle est linéaire, la fonction hypothèse nommée fonction Score, notée S, pourra s'écrire comme suit :

$$S(X^{(i)}) = \sum_{i=0}^{n+1} (\theta_i x_i)$$



avec :

- $X^{(i)}$  : une observation, qui représente un vecteur contenant  $x_1, x_2, \dots, x_n$
- $x_i$  : est une variable prédictive (feature)
- $\theta_i$  : est un poids/paramètre de la fonction hypothèse qu'on cherche à déterminer afin d'obtenir la fonction de prédiction. [6]

L'objectif de cette fonction est de trouver les coefficients  $\theta_0, \theta_1, \dots, \theta_n$  de sorte que :

- $S(X^{(i)}) > 0$  lorsque la classe vaut 1
- $S(X^{(i)}) < 0$  lorsque la classe vaut 0

Ensuite il s'agit de calculer la probabilité d'une classe, afin de modéliser les données, la fonction Sigmoidale sera appliquée sur la fonction Score, qui est interprétée comme la probabilité que l'observation  $X$  soit une classe positive "1". [6]

La fonction Logistique est ainsi définie par la formule suivante :

$$\text{Sigmoid}(x) = \frac{1}{1 + e^{-x}}$$

La fonction Sigmoidale est illustrée par la représentation graphique de la figure ci-dessous :

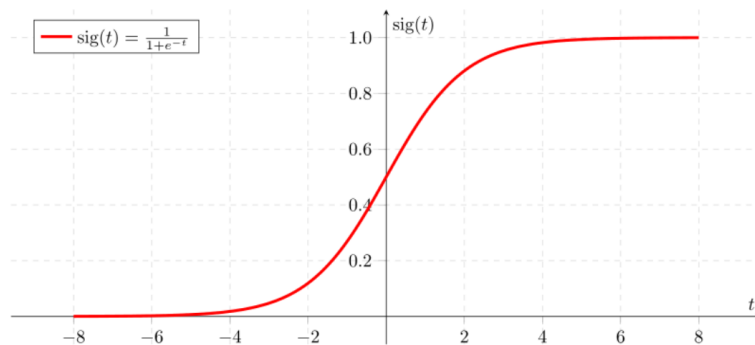


FIGURE 3.1 – La fonction Sigmoidale [6]

L'application de la fonction sigmoïde sur la fonction Score, résulte la fonction hypothèse pour la régression logistique suivante :

$$\text{Sigmoid}(\Theta X) = \frac{1}{1 + e^{-(\Theta X)}}$$

Son interprétation est simple :



Le nombre renvoyé par la fonction Sigmoid représente la probabilité que l'observation soit de la classe 1 (la classe positive) en considérant les paramètres  $\Theta$ . [6]

Cette probabilité, se définit par :

$$H(X) = P(y = 1|X; \Theta) \text{ [6]}$$

Ce qui en déduit que celle de la classe négative 0, est :

$$P(y = 0 | X; \Theta) = 1 - P(y = 1 | X; \Theta) \text{ [6]}$$

\* **Les hyperparamètres ajustés :**

**C** = 0.5 : Cette valeur définit l'inverse de la force de régularisation qui doit être un flottant positif.

**solver** = saga : Il s'agit de l'algorithme à utiliser dans le problème d'optimisation.

**max\_iter** = 500 : C'est le nombre maximal d'itérations nécessaires pour que les solveurs convergent.

**penalty** = l1 : Utilisé pour préciser la norme utilisée dans la pénalisation.

## 2. Arbre de décision

L'arbre de décision permet la construction des règles explicites et métiers à partir des données exploitables en fonction d'une variable cible qu'on cherche à expliquer. Il s'agit d'un outil élémentaire et indispensable qu'il faut maîtriser afin de réussir à facilement comprendre les algorithmes ensemblistes.[24]

### - Principe de fonctionnement

L'idée c'est d'expliquer une variable cible à partir d'autres variables explicatives, à cet égard le rôle de l'algorithme est de chercher à partitionner les individus en groupes d'individus les plus similaires possibles du point de vue de la variable à prédire. Ce qui en résulte un arbre qui révèle des relations hiérarchiques entre les variables. [24]

### - Construction des règles

Il s'agit en fait d'un algorithme itératif, qui consiste à séparer les individus en K groupes afin d'expliquer la variable cible à chaque itération.

Une première division ou encore split est obtenue à partir du choix d'une variable explicative qui permet la meilleure séparation des individus. Cette division donne des sous-populations correspondant au premier nœud de l'arbre.



Ce processus de split est ensuite répété plusieurs fois pour chaque sous-population ou noeuds précédemment calculés jusqu'à ce que le processus de séparation s'arrête. [24]

#### - Limites de l'algorithme

Les arbres de décisions peuvent facilement mener à un Overfitting. Qui est dû au fait que l'algorithme a trouvé une règle qui semble parfaite pour comprendre et décrire les données tandis que réellement cette règle ne peut pas être généralisée. Avec une possibilité que la règle trouvée peut changer radicalement si des observations supplémentaires sont introduites dans les données initiales. [24]

#### \* Les hyperparamètres ajustés :

**criterion** = gini : C'est la fonction de mesure de la qualité d'une division.

**max\_depth** = 8 : Définit la profondeur maximale de l'arbre.

**random\_state** = 50 : C'est une instance qui permet de contrôler le caractère aléatoire de l'estimateur.

**min\_samples\_leaf** = 10 : Représente le nombre minimum d'échantillons requis pour être à un nœud feuille.

**min\_samples\_split** = 2 : Il s'agit du nombre minimum d'échantillons requis pour diviser un nœud interne.

### 3. Méthodes ensemblistes

Les méthodes ensemblistes consistent à combiner plusieurs modèles d'apprentissage dans l'objectif d'avoir un modèle plus performant ayant des performances prédictives supérieures à celles de chacun des modèles pris indépendamment. D'où le fameux dicton «L'unité fait la force».

Les méthodes ensemblistes sont divisées en deux types ce sont les méthodes parallèles et les méthodes séquentielles.

#### (a) Les méthodes parallèles : Bagging

Le bagging (appelé bootstrap aggregating) s'agit d'un modèle homogène d'apprenants faibles qui apprennent les uns des autres indépendamment en parallèle et les combine afin de déterminer la moyenne du modèle.[7]

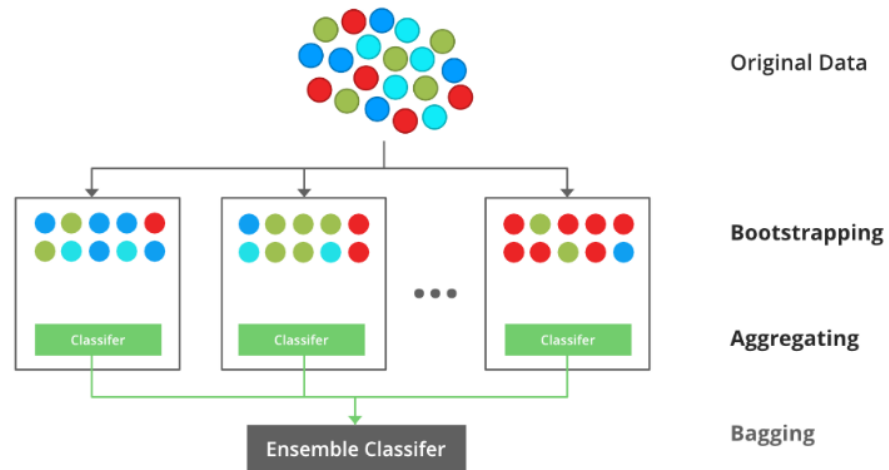


FIGURE 3.2 – Une illustration du concept d'agrégation bootstrap (Bagging) [7]

- **Forêt aléatoire**

Il s'agit d'un algorithme d'apprentissage supervisé utilisé à la fois pour la classification et la régression, basé sur le principe de Bagging.

Le Random Forest consiste à créer des arbres de décisions sur des échantillons de données sélectionnés aléatoirement, afin d'obtenir des prédictions de chaque arbre et sélectionner la meilleure solution au moyen du vote. D'où la nomination de Forêt aléatoire : " Une forêt est composée d'arbres. On dit que plus elle a d'arbres, plus une forêt est robuste." [8]

Il fonctionne en quatre étapes :

- 1 - Sélectionnez des échantillons aléatoires à partir d'un jeu de données donné.[8]
- 2 - Construisez un arbre de décision pour chaque échantillon et obtenez un résultat de prédiction de chaque arbre de décision.[8]
- 3 - Effectuez un vote pour chaque résultat prévu.[8]
- 4 - Sélectionnez le résultat de la prédiction avec le plus de votes comme prédiction finale. [8]

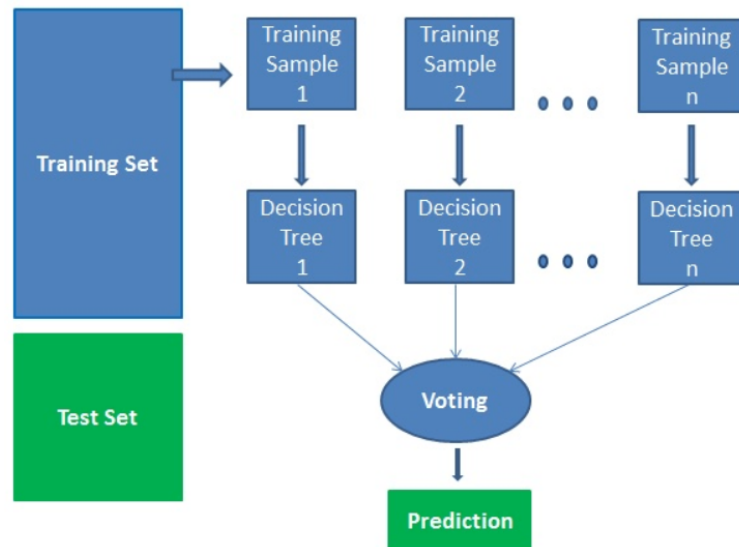


FIGURE 3.3 – Les étapes de fonctionnement de l'algorithme des forêts aléatoires[8]

\* **Les hyperparamètres ajustés :**

**n\_estimators** = 100 : C'est le nombre d'arbres dans la forêt.

**max\_depth** = 8 : Définit la profondeur maximale de l'arbre.

**random\_state** = 50 : C'est une instance qui permet de contrôler le caractère aléatoire de l'estimateur.

**Verbose** = 1 : Contrôle la verbosité lors de l'ajustement et de la prédiction.

**n\_jobs** = -1 : Définit le nombre de tâches à exécuter en parallèle, -1 signifie l'utilisation de tous les processeurs.

(b) **Les méthodes séquentielles : Boosting**

Tout comme le Bagging, le boosting est également un modèle homogène d'apprenants faibles mais fonctionne différemment du Bagging.

Dans ce modèle, les apprenants apprennent de manière séquentielle et adaptative pour améliorer les prédictions du modèle d'un algorithme d'apprentissage. [7]

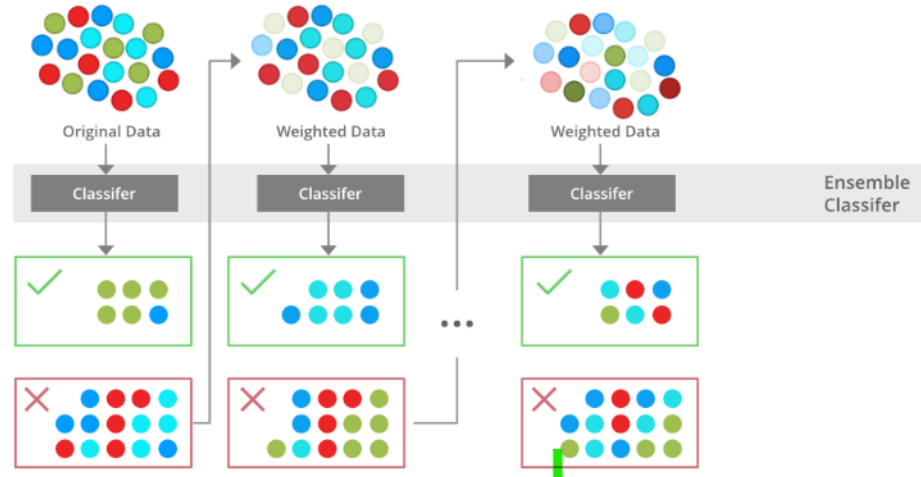


FIGURE 3.4 – Une illustration présentant l'intuition derrière l'algorithme de boosting)[7]

- **Lightgbm classifier**

Il s'agit d'un framework de gradient boosting rapide, distribué et de haute performance, qui utilise des algorithmes d'apprentissage basés sur des arbres qui sont considérés comme un algorithme très puissant en matière de calcul.

Tandis que d'autres arborescences d'algorithmes sont développées horizontalement, l'algorithme LightGBM se développe verticalement c'est à dire qu'il se développe au niveau des feuilles (leaf-wise), comme expliqué par la figure 3.5, alors que les autres se développent par niveau (level-wise), comme expliqué par la figure 3.6. [9]

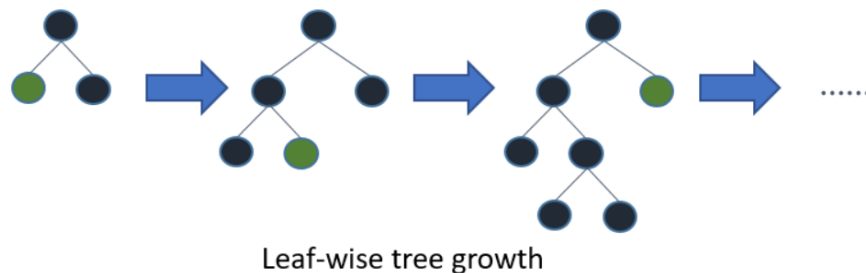


FIGURE 3.5 – Développement de l'arbre en leaf-wise par l'algorithme LightGBM [9]





FIGURE 3.6 – Développement de l'arbre en level-wise par les autres algorithmes de Boosting [9]

En effet l'appellation de « Light » vient de sa puissance de calcul et sa vitesse d'entraînement rapide ainsi que son utilisation réduite de la mémoire et sa capacité à traiter des données à grande taille.

Il n'est pas recommandé de l'utiliser avec de petits ensembles de données, vu qu'il est sensible à l'Overfitting et facilement overfitter les petits data. Ce qui explique son utilisation dans notre cas vu que nous possédons d'un ensemble de données volumineux contenant plus de 100 000 lignes. [9]

La seule complexité de cet algorithme c'est qu'il couvre plus de 100 paramètres. À cet égard les hyperparamètres qu'on a implémenté sont :

\* **Les hyperparamètres ajustés :**

**n\_estimators = 1000** : Nombre d'arbres boostés à ajuster.

**learning\_rate = 0.01** : Définit l'augmentation du taux d'apprentissage.

**max\_depth = 8** : Il décrit la profondeur maximale de l'arbre.

**num\_leaves = 58** : C'est le nombre des feuilles d'arbre maximales pour les lerners de base.

**colsample\_bytree = 0.613** : c'est le rapport de sous-échantillonnage des colonnes lors de la construction de chaque arbre.

**subsample = 0.708** : Rapport de sous-échantillon de l'instance de training.

**max\_bin = 407** : Définit le nombre maximum de cases dans lesquelles



la valeur de la feature sera placée.

**reg\_alpha = 3.564** : C'est un terme de régularisation L1 sur les poids

**reg\_lambda = 4.930** : C'est un terme de régularisation L2 sur les poids.

**min\_child\_weight = 6** : Somme minimale du poids de l'instance nécessaire dans un enfant (feuille).

**min\_child\_samples = 165** : Nombre minimale de données de l'instance nécessaire dans un enfant (feuille).

**Feature\_fraction = 0.7** : Signifie que le LightGBM sélectionnera 70 % des paramètres au hasard à chaque itération pour la construction d'arbres.

**bagging\_fraction = 0.7** : Signifie que la fraction de données à utiliser pour chaque itération est 70% afin d'accélérer le training et éviter le overfitting.

**Min\_data\_in\_leaf = 20** : Permet d'éviter de développer un arbre trop profond,

**Bagging\_freq = 20** : permet une vitesse plus rapide.

- **XgBoost classifier**

Grâce à sa performances et sa vitesse, cet algorithme est parmi les algorithmes les plus utilisés, il est également caractérisé par son évolutivité qui favorise un apprentissage rapide vu qu'il est de base parallélisable et distribuée et offre une utilisation efficace de la mémoire, il a le pouvoir d'exploiter la puissance des ordinateurs multicoeurs et il est aussi possible de le paralléliser avec les GPU et les réseaux d'ordinateurs, ce qui permet de d'entraîner sur des ensembles de données volumineux.

Cet algorithme est basé sur le principe de Boosting, il est donc itératif. Selon la distribution des exemples d'entraînement, il y aura une sélection d'un classifieur faible à chaque itération. Les exemples sont pondérés selon leurs difficultés avec le classifieur courant. Afin d'obtenir le classifieur final, les différents classifieurs seront agrégés.[25]

Le XGBoost, présente des améliorations algorithmiques de point de vue **Régularisation** vu qu'il a la possibilité de pénaliser les modèles complexes par la régularisation ce qui permet d'éviter le Overfitting, **structure de bloc d'apprentissage parallèle** comme il introduit l'utilisation



de plusieurs coeurs sur le CPU afin d'accélérer la vitesse de calcul, et de point de vue **connaissance du cache** en considérant qu'il vise toujours à faire le meilleur usage des ressources matérielles.[25]

\* **Les hyperparamètres ajustés :**

**learning\_rate** =0.01 : Définit le taux d'apprentissage, peut être défini pour contrôler la pondération des nouveaux arbres ajoutés au modèle.

**n\_estimators** =500 : C'est le nombre d'arbres.

**max\_depth** =7 : C'est la profondeur maximale d'un arbre

**gamma**=10 : Cette valeur permet la réduction minimale des pertes requise pour créer une partition supplémentaire sur un nœud feuille de l'arbre.

**colsample\_bytree** =0.4 : C'est le rapport de sous-échantillon des colonnes lors de la construction de chaque arbre.

**Objective** =binary :logistic : Définit la tâche d'apprentissage et l'objectif d'apprentissage correspondant. Dans notre cas il s'agit d'une classification binaire.

**tree\_method** ='hist' : prend en charge l'entraînement distribué.

**eta** = 0.4 : Permet la réduction de taille de pas utilisée dans la mise à jour pour empêcher le sur-ajustement.

**subsample** =0.4 : C'est le rapport de sous échantillonnage, qui signifie que XGBoost échantillonnerait au hasard 40% des données d'entraînement avant de faire pousser des arbres afin d'éviter le Overfitting.

**scale\_pos\_weight** =8 : Permet de contrôler l'équilibre des poids positifs et négatifs dans le cas des classes déséquilibrées.

#### 4. Modélisation des réseaux de neurones artificiels

Les réseaux de neurones artificiels sont l'un des principaux outils utilisés dans le Machine Learning. Comme le terme "neuronal" de leur nom l'indique, il s'agit de systèmes inspirés du cerveau qui visent à reproduire la manière dont les humains apprennent. Les réseaux neuronaux se composent de couches d'entrée et de sortie, ainsi que (dans la plupart des cas) d'une couche cachée constituée

d'unités qui transforment l'entrée en quelque chose que la couche de sortie peut utiliser.

Les RNA comportent trois couches inter-connectées. La première couche est constituée de neurones d'entrée. Ces neurones envoient des données à la deuxième couche, qui à son tour envoie les neurones de sortie à la troisième couche.

Les ANN sont considérés comme des outils de modélisation statistique non linéaire des données qui permettent de modéliser les relations complexes entre les entrées et les sorties ou de trouver des modèles. Il est à noter qu'un neurone peut également être appelé un perceptron.[10]

## Principe de fonctionnement

On dit que le fonctionnement des ANN prend ses racines dans le réseau neuronal qui réside dans le cerveau humain. En effet, l'ANN fonctionne sur un élément appelé "état caché", ces états cachés sont similaires aux neurones. Chacun de ces états cachés est une forme transitoire qui a un comportement probabiliste. Une grille de ces états cachés agit comme un pont entre l'entrée et la sortie.

Nous avons ainsi une couche d'entrée qui représente les données que nous fournissons à l'ANN, et nous avons également les couches cachées, qui sont l'endroit où la magie se produit. Enfin, nous avons la couche de sortie, qui est l'endroit où les calculs terminés du réseau sont placés pour que nous puissions les utiliser. [10]

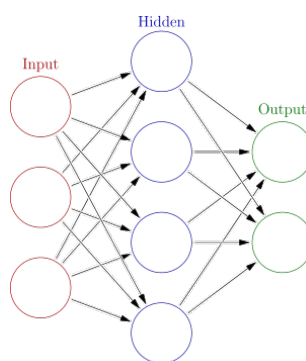


FIGURE 3.7 – Architecture d'un ANN[10]

Au départ, les poids du réseau peuvent être aléatoires. Lorsque l'entrée est donnée à la couche d'entrée, le processus avance et la couche cachée reçoit l'entrée combinée avec les poids. Ce processus se poursuit jusqu'à ce que la couche finale de sortie soit atteinte et que le résultat soit donné. Lorsque le résultat est



donné, il est comparé à la valeur réelle et un algorithme de rétro-propagation entre en jeu pour ajuster les poids des liens du réseau afin d'améliorer le résultat.

Quant aux neurones des couches, ils sont responsables de l'apprentissage individuel. Ils sont constitués d'une fonction d'activation qui permet au signal de passer ou non en fonction de la fonction d'activation utilisée et de l'entrée provenant de la couche précédente. [10]

### Construction de notre réseau de neurones

Un bon modèle de réseau neuronal est instancié en utilisant des paramètres particuliers tel que le nombre d'entrées, de sorties et de couches cachées.

- **Le nombre de neurones d'entrées, et de neurones des couches cachées :** Le choix de nombre de neurones a été fait par la fonction `RandomizedSearchCV()`, nous avons donc opté pour 47 neurones en entrée et pour les couches cachés également.
- **Le nombre de couches cachées :** Dans la plupart des cas, une ou deux, voire trois couches cachées se sont avérées suffisantes, et l'augmentation du nombre de ces couches accroît également le risque de Overfitting. Le nombre de couches cachées ne peut pas être déduit facilement, il a fallu entraîner des réseaux avec un nombre différent de couches cachées et faire des comparaisons.

A la fin de notre itération, nous optons pour trois couches cachées comme solution optimisée du nombre.

- **Le nombre de neurones de sorties :** Dans notre cas, nous sommes face à un problème de classification binaire (le client a des difficultés à rembourser le prêt ou non) nous avons donc opté pour un neurone de sortie.

La figure ci-dessous illustre l'architecture de notre modèle ANN, en effet nous avons utilisé une couche de type Dense, et il s'agit d'une couche ordinaire de réseau neuronal à connexion profonde. Il s'agit de l'une des couches les plus courantes et les plus fréquemment utilisées.

La couche d'entrée a 47 neurones, chacune des couches cachées a également 47 neurones.

À la fin de chaque couche cachée, nous ajoutons une batch normalisation, qui est une technique permettant d'améliorer la vitesse, les performances et la stabilité du réseau neuronal artificiel. La couche de sortie comporte un neurone qui transmet les valeurs au programme.

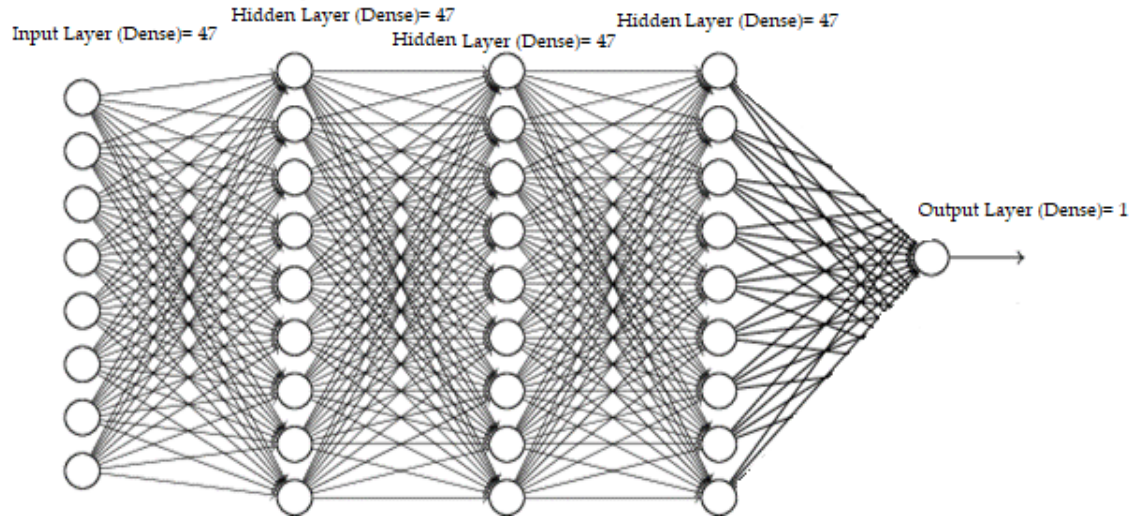


FIGURE 3.8 – L'architecture de notre modèle ANN

### Les paramètres de notre réseau de neurones

Pour un ANN donné, les entrées sont multipliées par les poids d'un neurone et additionnées. Cette valeur est ensuite transformée via une fonction d'activation. Nous avons utilisé deux types de fonctions d'activation :

- Le premier type est la fonction **ReLU**, qu'on a utilisé pour la couche d'entrée et les couches cachées.
- La deuxième fonction qu'on a utilisé est la fonction **Sigmoïde** pour la couche de sortie.

Et finalement, nous avons opté pour **Adam** comme optimiseur dans notre modèle.

### 3.1.2 Synthèse

Le tableau ci-dessous résume les différents modèles de classification que nous avons utilisés ainsi que leurs hyperparamètres ajustés :



Modèles	Hyperparamètres
Logistic Regression	$C = 0.4$ Solver = liblinear max_iter = liblinear Penalty = l1
Decision Tree	Criterion = gini max_depth = 8 random_state = 50 min_samples_leaf = 10 min_samples_split = 2
Random Forest	n_estimators = 100 max_depth = 8 random_state = 50 Verbose = 1 n_jobs = -1
Lightgbm	n_estimators = 1000 learning_rate = 0.01 max_depth = 8 num_leaves = 58 colsample_bytree = 0.613 subsample = 0.708 max_bin = 407 reg_alpha = 3.564 reg_lambda = 4.930 min_child_weight = 6 min_child_samples = 165 Feature_fraction = 0.7 Bagging_fraction = 0.7 Min_data_in_leaf = 20 Bagging_freq = 20
XGboost	learning_rate = 0.01 n_estimators = 500 max_depth = 7 gamma = 10 colsample_bytree = 0.4 Objective = binary : logistic tree method = 'hist' eta = 0.4 subsample = 0.4 scale_pos_weight = 8
ANN	Nombre de neurones d'entrée : 47 Nombre de neurones de couches cachées : 47 Nombre de couches cachées : 3 Nombre de neurones de sortie : 1 Optimiseur : Adam

TABLE 3.1 – Modèles utilisées et leurs hyperparamètres



## 3.2 Évaluation

Afin de détecter les clients présentant un risque de défaut, nous avons testé plusieurs modèles de classification dans le but de choisir deux modèles à implémenter dans notre système de Early warning.

Sur la base des métriques d'évaluation des modèles de classification nous avons évalué chacun de ces modèles.

la comparaison des performances des modèles obtenus est présentée dans le tableau ci-dessous :

Comparaison des modèles							
Modèle	Accuracy	AUC Score	Recall(Non_risqué)	Recall(risqué)	Recall_Avg	precision	f1-score
Random Forest	0.92	0.75980	0.98	0.16	0.57	0.63	0.58
Decision Tree	0.87	0.71862	0.93	0.29	0.61	0.58	0.59
Logistic Regression	0.88	0.78302	0.91	0.39	0.65	0.60	0.62
LGBMClassifier	0.89	0.80081	0.93	0.38	0.65	0.62	0.63
Xgboost	0.76	0.79030	0.76	0.67	0.72	0.57	0.57
ANN	0.82	0.76387	0.84	0.51	0.68	0.58	0.59

TABLE 3.2 – Les performances des modèles

Les figures ci-dessous montrent les différentes matrices de confusion obtenues à partir des modèles construits. Les colonnes de chaque matrice représentent les classes prédites par le modèle et les lignes représentent les classes réelles. Le nombre de prédictions correctes pour chaque classe est ainsi représenté sur la diagonale principale de la matrice.

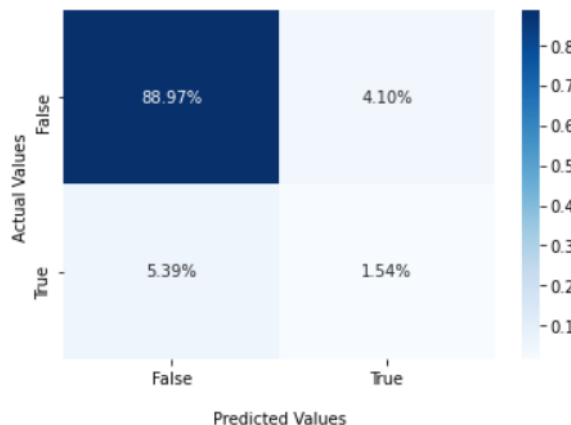


FIGURE 3.9 – Matrice de confusion du modèle Random Forest

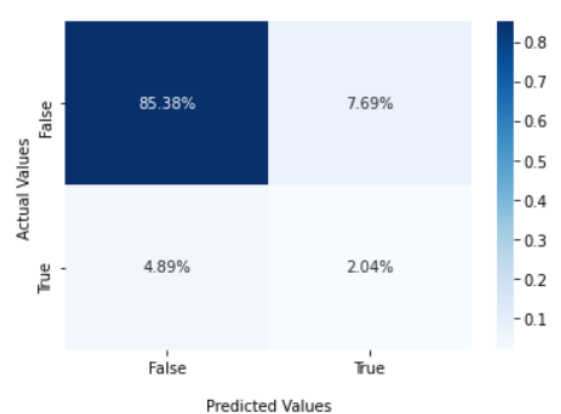


FIGURE 3.10 – Matrice de confusion du modèle Decision Tree



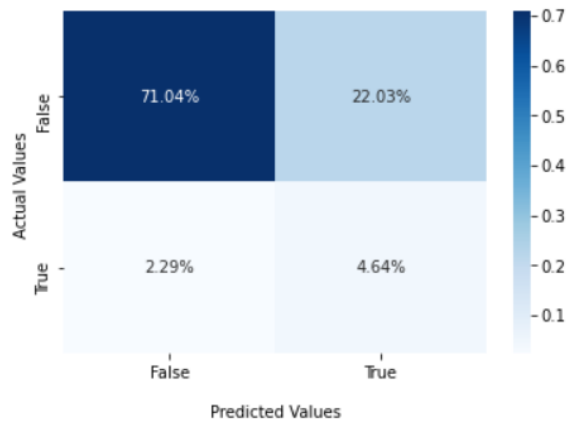


FIGURE 3.11 – Matrice de confusion du modèle XGboost

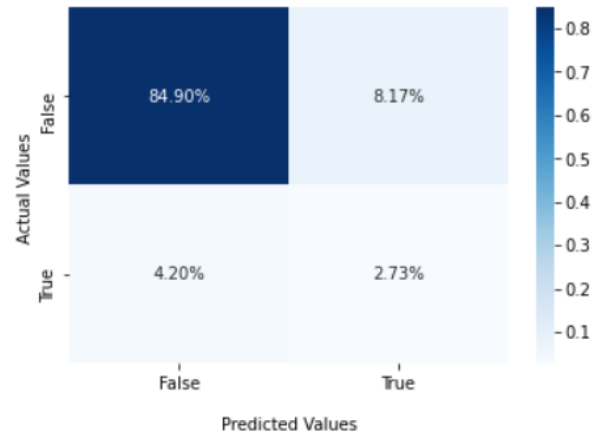


FIGURE 3.12 – Matrice de confusion du modèle Logistic Regression

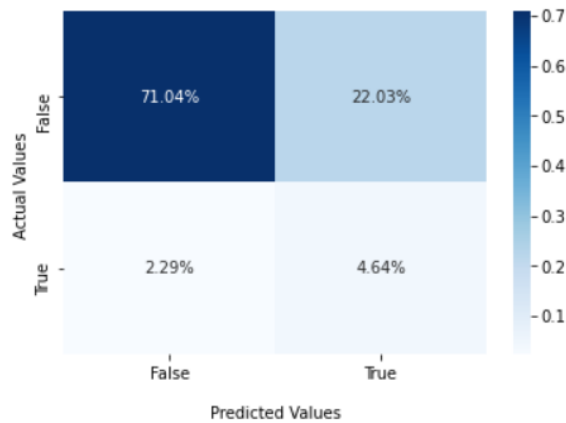


FIGURE 3.13 – Matrice de confusion du modèle XGboost

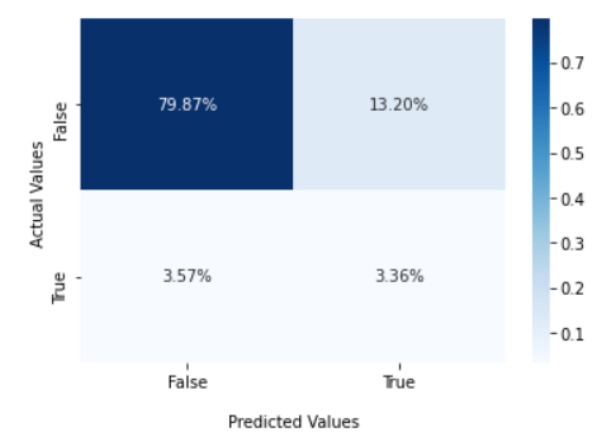


FIGURE 3.14 – Matrice de confusion du modèle ANN

Nous détaillons également dans les figures ci-dessous les courbes ROC-AUC au niveau de l'entraînement et de la validation pour chacun de nos modèles :

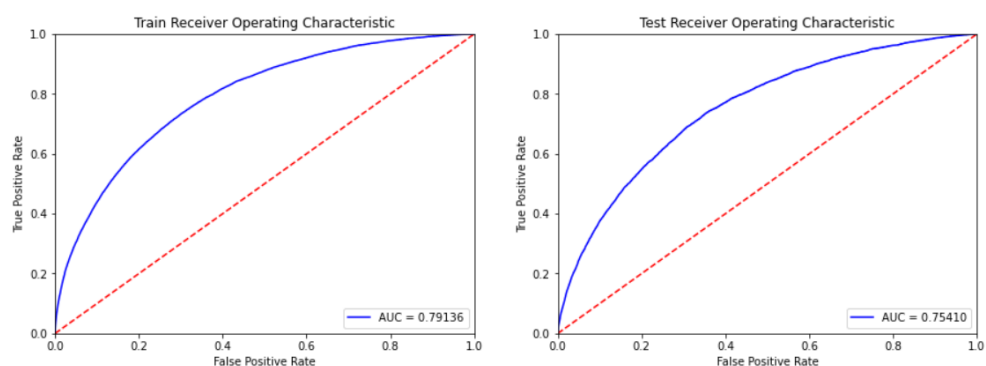


FIGURE 3.15 – ROC-AUC curves pour le modèle Random Forest

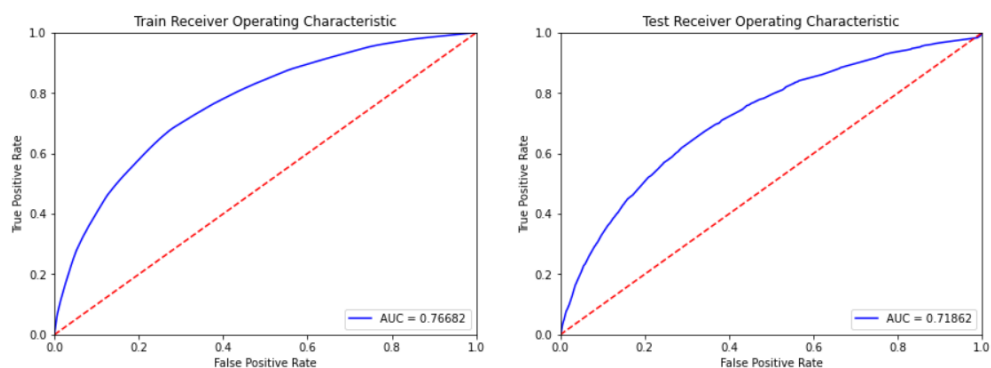


FIGURE 3.16 – ROC-AUC curves pour le modèle Decision Tree

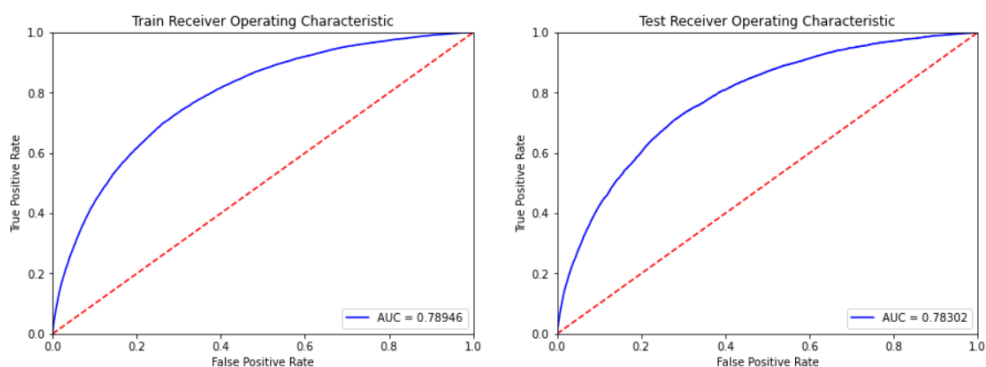


FIGURE 3.17 – ROC-AUC curves pour le modèle Logistic Regression

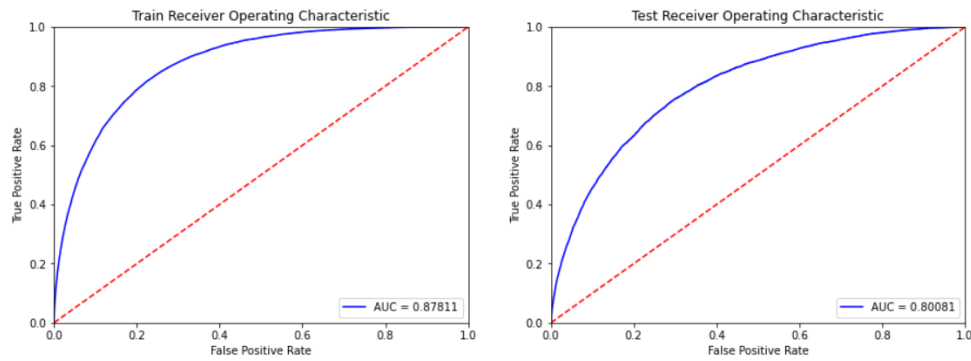


FIGURE 3.18 – ROC-AUC curves pour le modèle Lightgbm

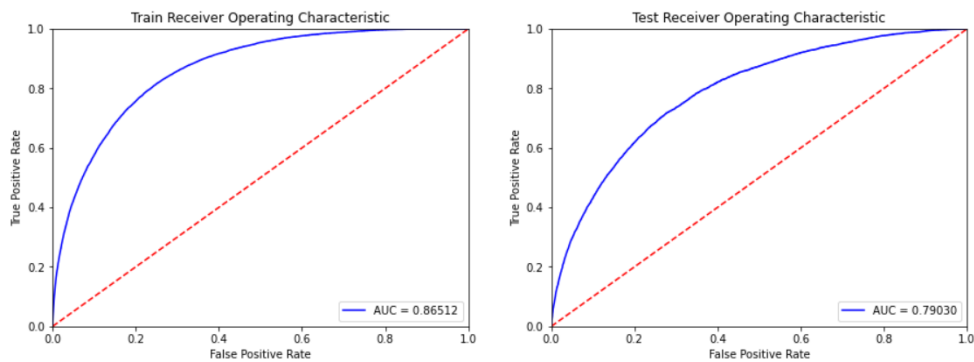


FIGURE 3.19 – ROC-AUC curves pour le modèle XGboost

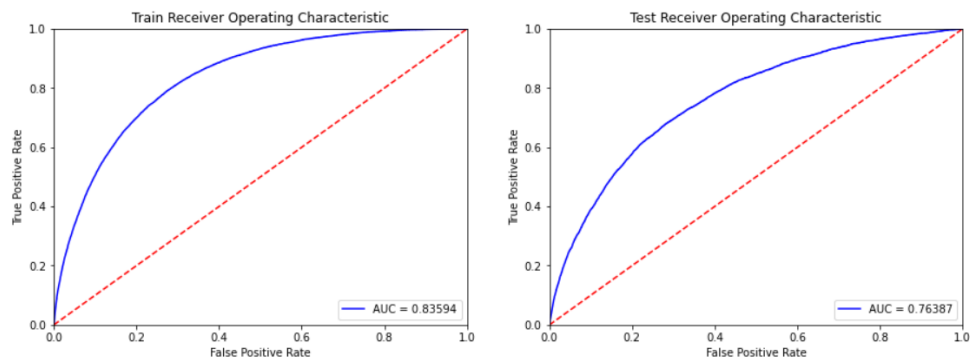


FIGURE 3.20 – ROC-AUC curves pour le modèle ANN

Comme expliqué dans la section de l'EDA, ce projet nous a mis face à un dataset déséquilibré, où il y a un grand nombre de clients qui ont été capables de rembourser leurs prêt par rapport aux clients qui sont des défaillants.

Dans ce cas deux scénarios se présentent où la banque subira des pertes si la prédiction est erronée :



- Scénario 1 : Si le modèle a prédit que le client remboursera le prêt alors qu'en réalité il s'agit d'un défaillant.
- Scénario 2 : Si le modèle a prédit que le client est un défaillant alors qu'en réalité il est capable de rembourser le prêt, ainsi le candidat méritant n'obtient pas de prêt et la banque perd des intérêts en retour.

La perte sera plus importante dans le premier scénario, ce qui implique que notre but est de savoir parmi tous les points réellement positifs (Il est à noter que pour notre cas la classe 1 des clients risqué présente les positifs), combien d'entre eux sont prédits positifs.

- **L'Accuracy** ne peut pas être utilisé pour un dataset déséquilibré, elle n'est donc pas la bonne métrique à considérer dans notre cas.
- **La précision** montre parmi les points qui sont prédits positifs combien d'entre eux sont réellement positifs. La précision ne tient pas donc compte des points qui étaient réellement positifs et qui sont prédits positifs.
- **Le rappel (Recall)** est important vu qu'il indiquera de tous les points qui sont réellement positifs combien d'entre eux sont prédits correctement. Il est donc important pour décrire le premier scénario. Tant que le Recall est élevé on peut dire que les défaillants sont correctement prédits.
- **Le F1 Score** c'est la moyenne géométrique de la précision et le Recall, comme la précision n'est pas très utile dans notre cas donc le F1 score n'est pas vraiment important.
- **Le score ROC-AUC** contient le taux de vrais positifs et le taux de faux positifs.  
Le taux de vrais positifs est le même que le Recall, donc le premier scénario est couvert. Tandis que le taux de faux positifs définit sur tous les points qui sont réellement négatifs, combien sont prédits comme positifs, donc le deuxième scénario est également couvert.  
Ce qui implique que le Le score AUC est une mesure importante dans notre cas vu qu'elle couvre les deux scénarios à la fois dans lesquels la banque peut subir une perte.

Pour conclure, notre évaluation des différents modèles sera basé essentiellement sur le **AUC score** et le **Recall** en prenant en compte le recall de la classe Risqué afin de satisfaire le scénario 1.

En analysant le tableau 4.1, on peut déduire que le AUC score est élevé pour le LGBMClassifier qui est de l'ordre de 80% avec un Recall de 65% avec prédiction correcte de 38% des points positifs pour la Classe risqué, ce qui est



insuffisant pour satisfaire le premier scénario, la perte sera importante pour la banque dans ce cas. Dans ce cas nous devons passer à l'analyse au modèle XGboost qui a un AUC score de l'ordre de 79% avec un Recall de 72% et une prédiction correcte de 67% des points positifs pour la Classe risqué, nous validons donc le modèle XGboost.

Pour le choix du deuxième modèle, nous pouvons opter pour le ANN, tant qu'il a un AUC Score de l'ordre de 76% avec un Recall de 68%.

Finalement, notre choix s'est focalisé sur le modèle Xgboost de Machine Learning et le ANN du Deep Learning.

À l'aide de la fonction `dump()` du module `pickle` nous avons enregistré le modèle XGboost et à l'aide de la fonction `save()` du Keras nous avons enregistré le modèle ANN, pour une utilisation ultérieure dans la prédiction.

### 3.3 Pipeline de la prédiction

Pour récapituler, nous avons commencé par un data construit de plusieurs tables sur lesquelles nous avons appliqué un ensemble d'actions allant du feature engineering jusqu'à l'échantillonnage aléatoire afin d'atteindre une forme finale de données que nous avons introduit dans les deux modèles choisis au niveau de l'étape d'évaluation (le XGboost et l'ANN) afin de prédire le défaut de paiement. La figure ci-dessous détaille le pipeline que nous avons suivi :

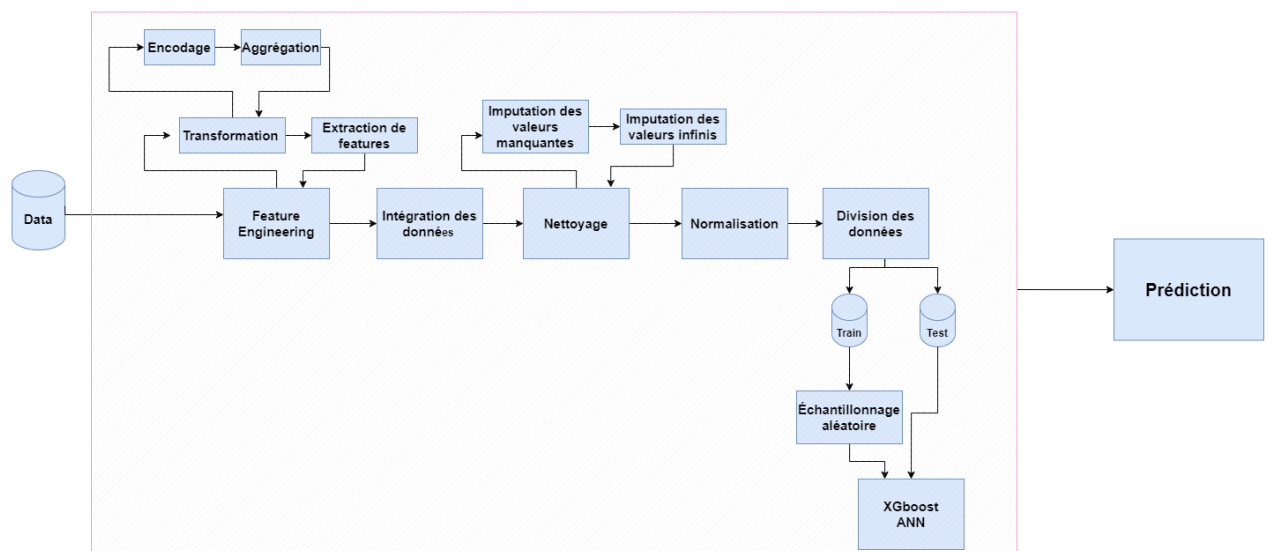


FIGURE 3.21 – Pipeline de la prédiction



## Conclusion

Dans ce chapitre, nous nous sommes concentrés sur la phase de modélisation et évaluation où nous avons choisi d'évaluer des différents types de modèles tel qu'un modèle mathématique, arbre de décision, méthodes ensemblistes et nous avons également fait recours au Deep Learning en construisant un réseau de neurones. Nous avons fini par valider deux modèles.

Nous clôturons notre projet par la dernière étape qui est le déploiement présenté par le chapitre suivant.

# Chapitre 4

## Déploiement



## Introduction

Dans ce chapitre nous allons mettre en production les deux modèles validés dans la section précédente dans une application web afin de permettre à l'utilisateur final de les utiliser dans le but d'obtenir des résultats faciles à comprendre. À cet égard, nous allons analyser les besoins et présenter le diagramme de cas d'utilisation de l'application conçue et nous aborderons également son implémentation.

### 4.1 Analyse des besoins

Dans cette section, nous allons définir les acteurs et les fonctionnalités de notre système.

#### 4.1.1 Identification des acteurs

L'acteur est l'utilisateur final qui interagira avec le système de façon à se bénéficier des fonctionnalités souhaitées et à réaliser ses objectifs. Dans notre cas l'utilisateur de l'application est le responsable crédit ou "credit manager" de la banque, celui qui est responsable de gérer le risque client.

#### 4.1.2 Identification des besoins fonctionnels

Il s'agit des attentes de l'utilisateur final du système que nous avons conçu. En effet, l'idée c'est l'intégration de concepts d'apprentissage automatique dans une applications web, afin de permettre à un simple utilisateur de l'utiliser et d'obtenir des résultats faciles à comprendre. Les besoins fonctionnels de notre PoC sont :

- **S'authentifier**
- **Charger un fichier de demandes de crédits** : Visualiser le fichier dans un Datatable avec le possibilité de filtrer les données, les modifier, supprimer et exporter la version finale sous format csv.
- **Consulter le rapport d'EDA d'un fichier exporté** : Après avoir uploader le fichier un rapport est crée contenant un aperçu sur les données, tel que les valeurs manquantes, le nombres de variable, le nombre de lignes, les types de





variables, etc..

- **Prédire le risque d'une demande de crédit** : En sélectionnant l'identifiant d'une demande de crédit une prédiction est affichée contenant si le client est risqué ou non avec une probabilité prédite.

### 4.1.3 Les besoins non fonctionnels

Notre application devrait être en mesure d'assurer les besoins suivants :

- **L'extensibilité** : Vu que notre application présente un PoC, elle doit être flexible à l'ajout de nouvelles fonctionnalités ou à la mise à jour de celles qui sont déjà existantes.
- **La performance** : L'application doit répondre à toutes les attentes des utilisateurs et satisfaire notre client.
- **La portabilité** : L'application doit être capable à être déployées, accessibles et gérées de manière portable,
- **L'ergonomie** : Notre application doit être facile à utiliser avec des interfaces simples et conviviales.

### 4.1.4 Diagramme de cas d'utilisation

La figure ci-dessous offre une vue d'ensemble sur les fonctionnalités de notre application, il s'agit du diagramme de cas d'utilisation global :

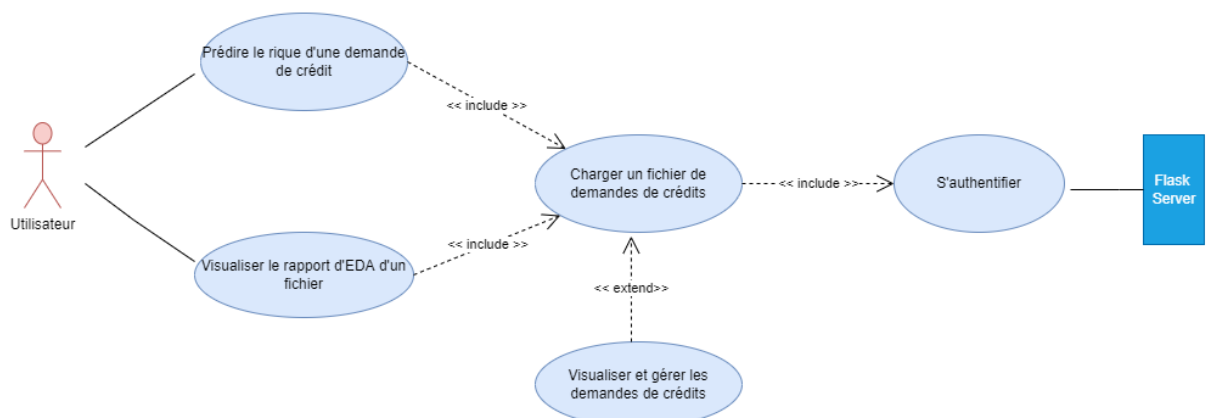


FIGURE 4.1 – Diagramme de cas d'utilisation global



### 4.1.5 Analyse

#### Diagramme de séquence système du cas d'utilisation «Charger un fichier de demandes de crédits» :

La figure 4.2 détaille le diagramme de séquence système du cas d'utilisation «Charger un fichier de demandes de crédits».

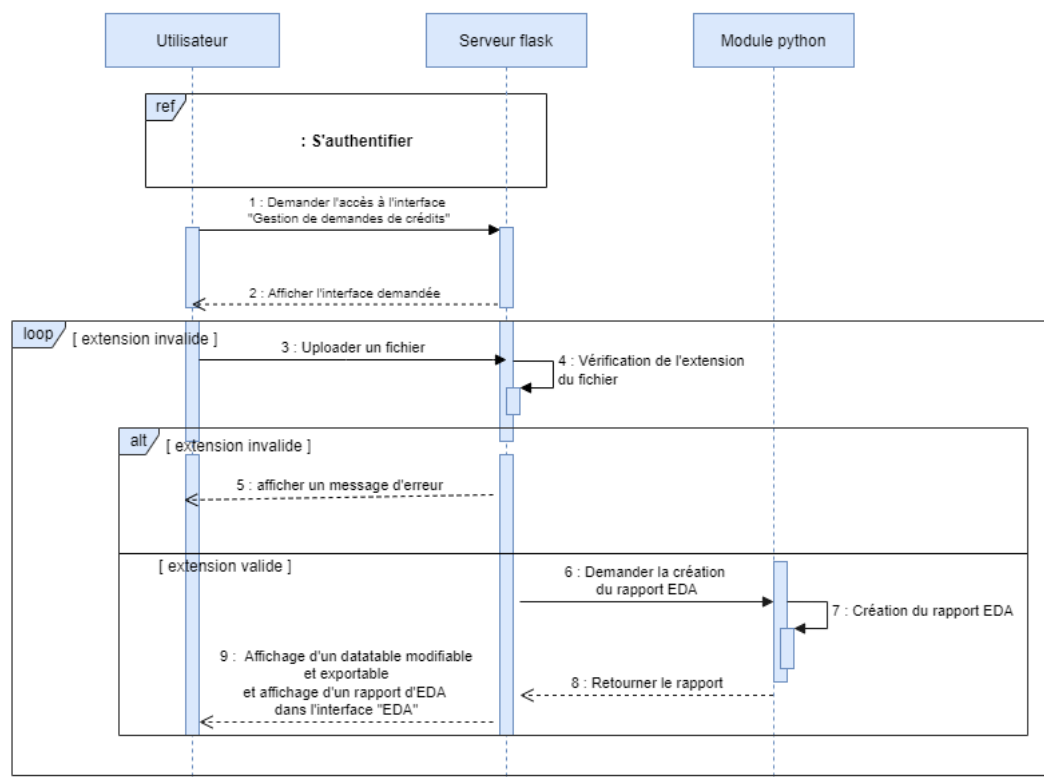


FIGURE 4.2 – Diagramme de séquence système du cas d'utilisation «Charger un fichier de demandes de crédits»

#### Diagramme de séquence système du cas d'utilisation «Prédire le risque d'une demande de crédit» :

La figure 4.3 détaille le diagramme de séquence système du cas d'utilisation «Prédire le risque d'une demande de crédit».

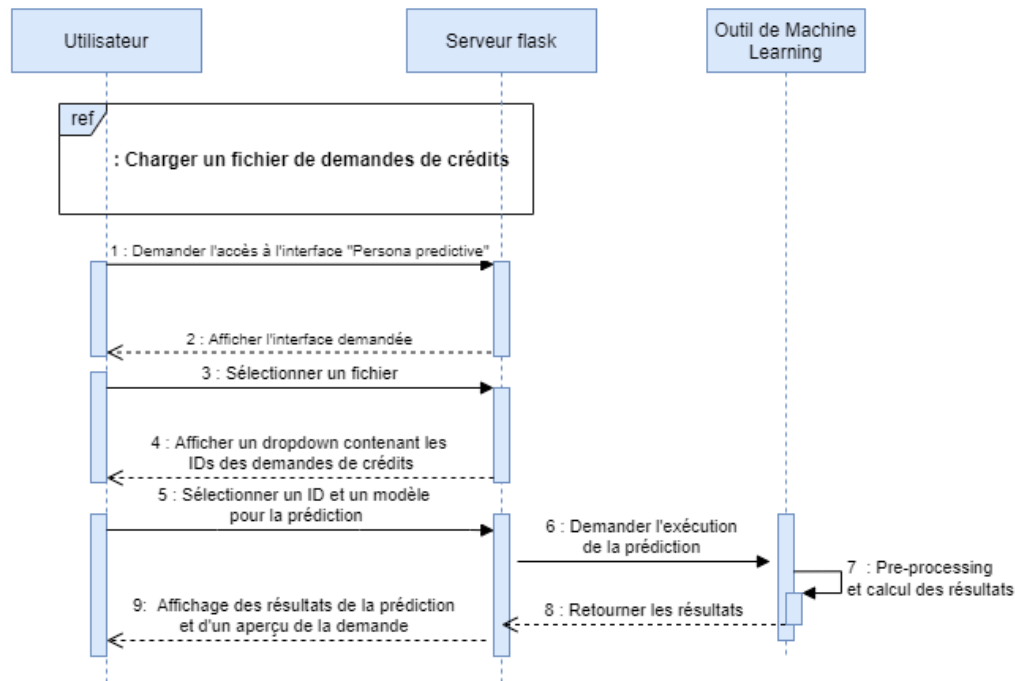


FIGURE 4.3 – Diagramme de séquence système du cas d'utilisation «Prédire le risque d’une demande de crédit»

## 4.2 Implémentation

Dans le but de produire des prédictions de défaut de paiement en temps réel pour chacun des modèles choisi dans le section précédente, nous avons créé une application Dash Plotly intégrée dans Flask qui utilise chacun de nos modèles pour faire des prédictions sur chaque demande de crédit choisie par l'utilisateur et renvoie les prédictions et les probabilités prédites.

### 4.2.1 Architecture physique adoptée

Pour la réalisation de notre application, nous avons opté pour une architecture à deux niveaux qui réunit à la fois le framework flask et Dash Plotly.

L'idée c'est de mettre en place une application sécurisée avec une interface d'authentification robuste et comme le framework Dash est construit sur Flask, nous avons opté pour une page d'authentification avec Flask, et pour la connexion à la base de



données et la création du compte utilisateur nous avons fait recours à la bibliothèque `flask_mysql`, à cet égard nous avons intégré notre application Dash sur un serveur Flask.

Notre application adopte ainsi une architecture basée sur le modèle client serveur, ou encore Front-end/Back-end comme illustré par la figure 4.4.

Le front-end est le code exécuté du côté du client, Il s'agit dans notre cas de l'application Dash qui résume HTML/CSS, React, Flask et Plotly, enveloppant le tout dans une API Python unifiée et facile à utiliser offrant un front-end réactif, et révolutionne également le paradigme Modèle-Vue-Contrôleur (MVC) du niveau de développement d'applications :

- `@app.callbacks` : fournit une API Python pour définir les méthodes du contrôleur .
- `app.layout` : Fournit également une API Python pour structurer les vues qui s'accrochent à la couche de contrôle.

Ce qui offre une intégration étroite d'un contrôleur en temps réel avec la vue. Donc, le front-end permettra aux clients d'interagir avec le serveur et lui envoyer des requêtes. La figure 4.5 détaille le modèle MVC expliqué.

Le back-end présente le code qui s'exécute sur le serveur, qui reçoit les requêtes des clients et contient la logique permettant de renvoyer des réponses aux clients, qui est dans notre cas le serveur Flask et MySQL le serveur de base de données.

Cette architecture est rapide, facile à implémenter, robuste et convient au micro-framework utilisé.

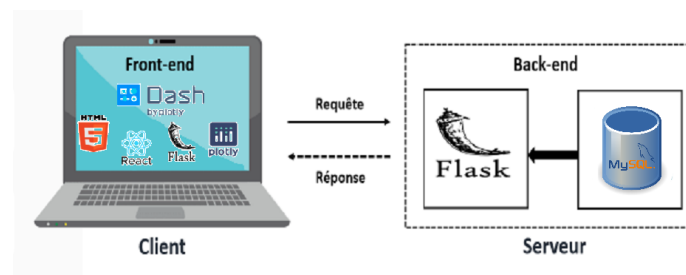


FIGURE 4.4 – Architecture à deux niveaux

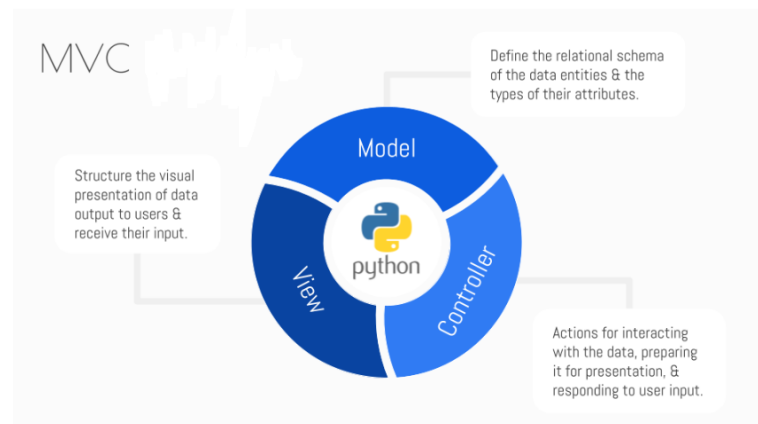


FIGURE 4.5 – Architecture Modèle-Vue-Contrôleur

### 4.2.2 Intégration du modèle de prédiction

Afin de pouvoir effectuer des prédictions sur notre application web, nous allons intégrer les deux modèles qui sont implémentés et validés au cours de la partie du modélisation et enregistrés dans un fichier sous forme binaire à l'aide de la fonction `dump()` du module `pickle` pour le modèle XGboost et la fonction `save()` du Keras pour le modèle ANN.

Cette intégration consiste à utiliser les fonctions `load()` et `load_model()` dans Flask.

Lorsque l'utilisateur sélectionne une demande de crédit et choisit un des deux algorithmes de prédictions, les données passent par l'étape de pre-processing et elles sont finalement dirigées par le serveur flask vers le modèle afin de faire la prédiction.

La figure ci-dessous détaille ce processus :

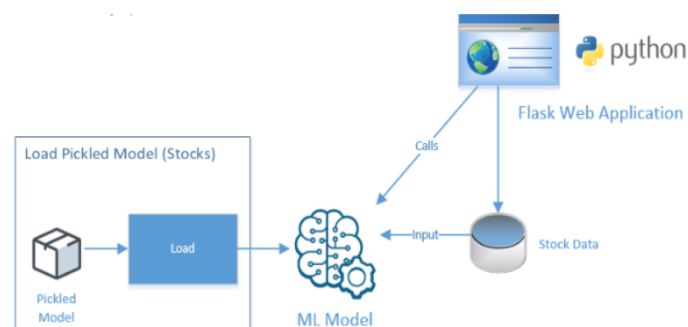


FIGURE 4.6 – Intégration du modèle de prédiction [11]



### 4.2.3 Réalisation des interfaces

- Interface d'authentification :

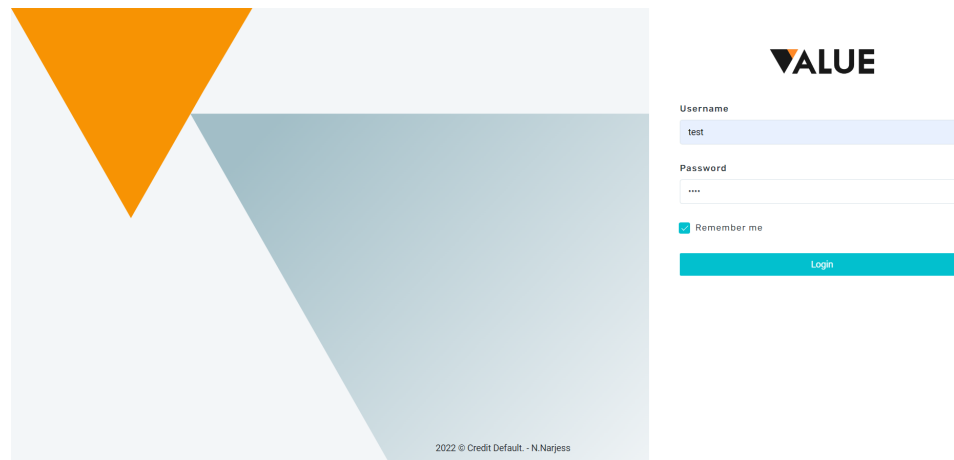


FIGURE 4.7 – Interface d'authentification

- Interface de chargement de fichier de demandes de crédits :

Cette interface permet à l'utilisateur d'uploader un fichier contenant une ou plusieurs demandes de crédits à traiter, et affichera un Datatable modifiable où on a la possibilité de gérer les données, les filtrer, les trier et exporter une version finale mise à jour.

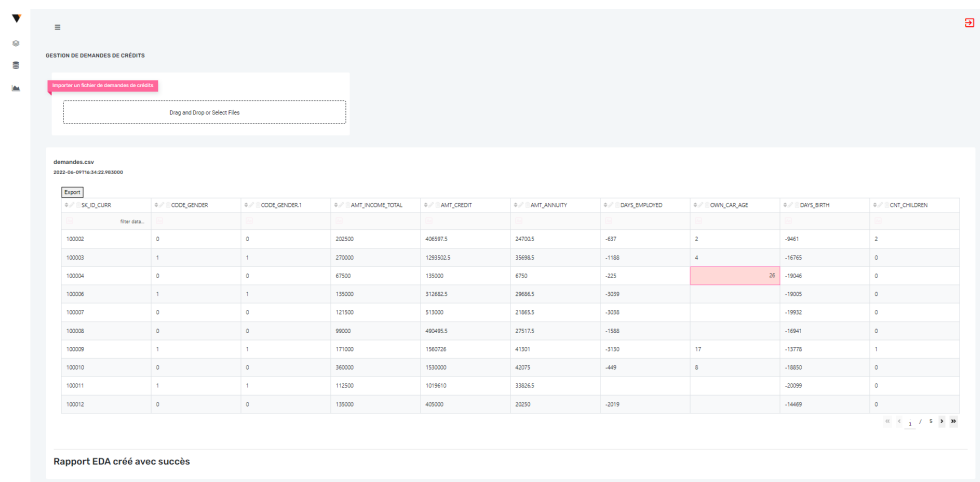


FIGURE 4.8 – Interface de chargement de fichier de demandes de crédits

- Interface de consultation du rapport d'EDA :

Pour chaque fichier chargé un rapport d'EDA est créé afin d'être consulté par l'utilisateur, contenant un aperçu sur les données tel que les valeurs manquantes, le nombre de variable, le nombre de lignes, les types de variables, les



corrélations entre eux, etc..

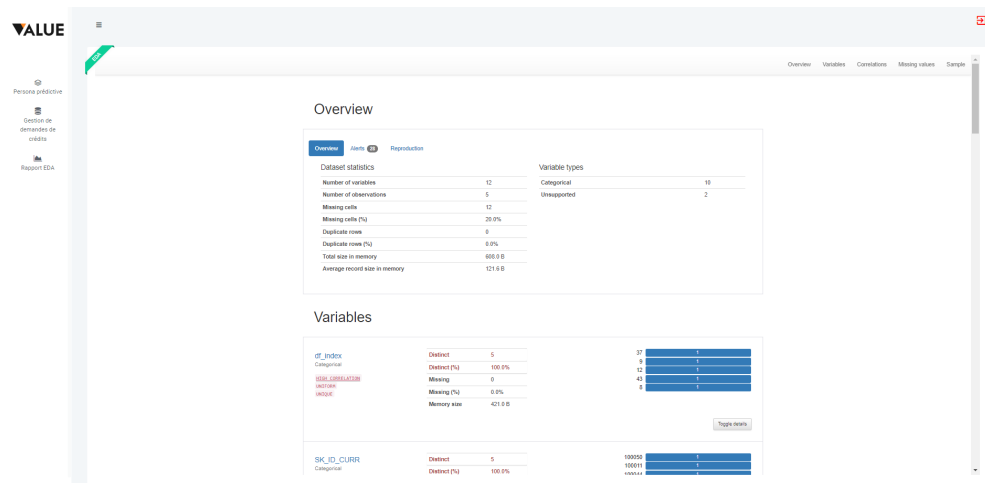


FIGURE 4.9 – Interface de consultation du rapport d'EDA

- **Interface de prédiction de défaut de paiement :**

Cette interface permet à l'utilisateur une prédiction de défaut de paiement en temps réel.

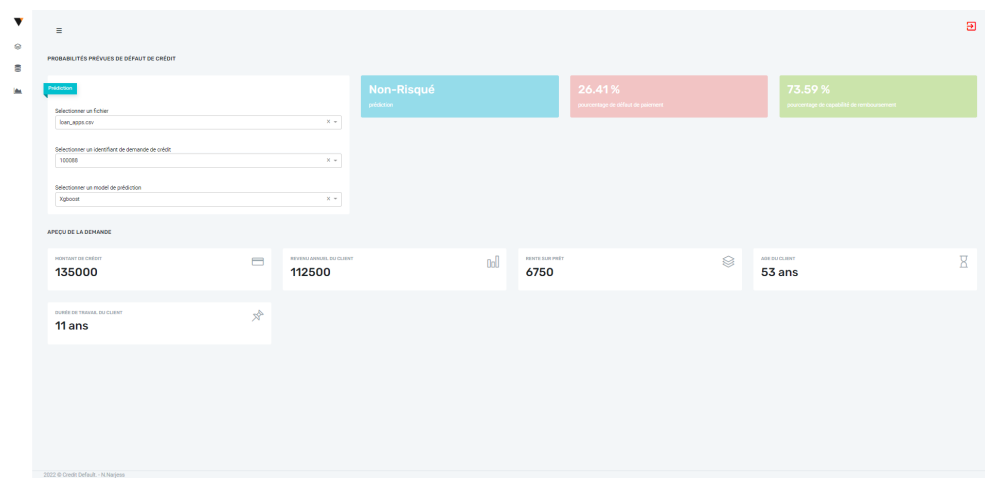


FIGURE 4.10 – Interface de prédiction de défaut de paiement « Cas d'un client non-risqué »

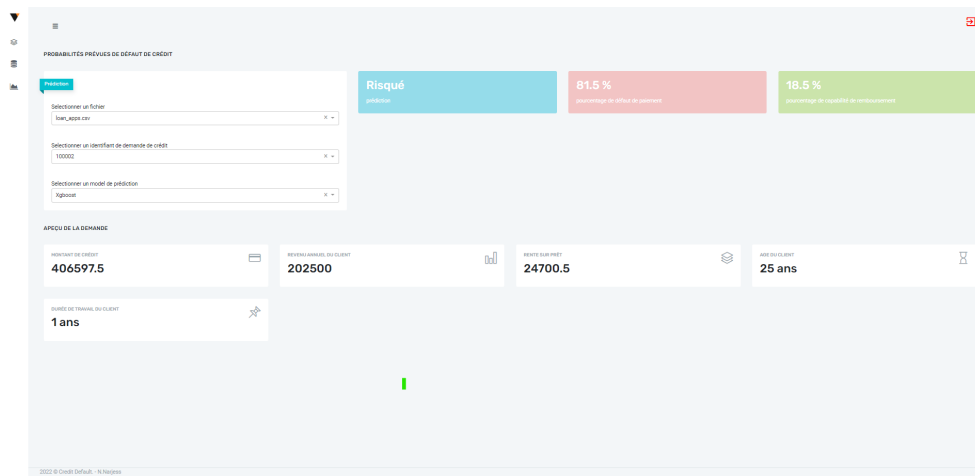


FIGURE 4.11 – Interface de prédiction de défaut de paiement « Cas d’un client Risqué »

#### 4.2.4 Outils de développement

- **Python** : Il s’agit du langage de programmation open source le plus utilisé. il s’est propulsé en tête de la gestion d’infrastructure, d’analyse de données ou dans le domaine du développement de logiciels.[26]
- **Flask** : C’est un framework web léger qui fait partie des catégories du micro-framework, fournissant des outils, des bibliothèques et des technologies qui vous permettent de créer une application Web.[27]
- **Dash Plotly** : Il s’agit d’un framework low-code permettant la création rapide des applications de données en Python, R, Julia et F (expérimental). Il est idéal pour la création et le déploiement des applications de données avec des interfaces utilisateur personnalisées. Il est particulièrement adapté à tous ceux qui travaillent avec des données.[28]
- **MySQL** : Il s’agit d’un système de gestion de base de données relationnelle, qui s’appuie sur le langage de requête structuré SQL (Structured Query Language). Utilisé pour toutes sortes d’applications, il est le plus souvent associé aux applications Web et à la publication de contenus en ligne.[29]
- **HTML5** : (HyperText Markup Language 5), est un format de données conçu pour représenter les pages web. [30]
- **Bootstrap** : Est une collection d’outils utiles à la création du design (graphisme, animation et interactions avec la page dans le navigateur, etc..) de





sites et d'applications web. [31]

#### 4.2.5 Environnement matériel

Notre projet a été réalisé sur une machine possédant les caractéristiques suivantes :

Marque	HUAWEI
Processeur	Intel Core i7-10510U 10ème génération
Disque dur	512 ssd
Mémoire vive	16 GO
Système d'exploitation	Windows 11

TABLE 4.1 – Environnement matériel

## Conclusion

Nous clôturons notre rapport avec ce dernier chapitre à travers lequel nous avons mis l'accent sur l'analyse et l'implémentation de notre application web qui a abouti à la réalisation d'un produit fonctionnel qui répond parfaitement aux besoins spécifiés au début du chapitre.



# Conclusion et perspectives

---

Ce présent mémoire a été rédigé dans le cadre du projet de fin d'études effectué au sein de la société Value Digital Services, pour l'obtention du diplôme de master professionnel en Data Science et développement des logiciels.

L'objectif principal de ce projet consiste à l'implémentation d'un système intelligent de Early Warning pour la détection des clients risqués pour le département risque d'une banque de détails. Il s'agit de la réalisation d'un proof of concept à proposer à notre client, permettant d'offrir tout un processus automatisé pour le scoring des dossiers des crédits en vue d'évaluer la capacité d'un client de remboursement du crédit sollicité.

Ce stage nous a donné l'opportunité d'entamer un projet assez volumineux et de comprendre la conduite des projets en Data Science. C'était une occasion intéressante qui nous a permis de mettre en oeuvre nos différentes connaissances acquises tout au long de notre cursus universitaire et de pratiquer la méthodologie CRISP-DM qui nous a appris à apprendre la gestion d'un projet de façon méthodique et organisée.

Cette expérience était enrichissante sur les deux plans technique et professionnel. Elle nous a permis de renforcer mes compétences techniques, admettre des nouvelles connaissances et nous a offert une initiation à la vie professionnelle tout en faisant partie d'une grande entreprise et d'une équipe bien collaborative et motivante.

Bien que nous avons pu effectuer les tâches qui nous étaient confiées et le produit final était à la hauteur des attentes de notre superviseur au sein de Value, ce projet représente le prototype qui sera livré au client en vue de recevoir ses remarques et feedbacks. Ces derniers vont nous servir comme un appui pour améliorer davantage la solution présentée et la rendre ainsi plus adéquate à ses besoins tout en satisfaisant les exigences préalablement définies de sa part.

En effet, comme perspective pour ce projet nous avons prévu d'implémenter un simulateur de modèles de classification dans notre application qui permettra de construire manuellement son modèle de prévision et visualiser sa performance et après l'avoir validé un job sera créé afin d'entraîner le modèle et le tester à chaque fois que des nouvelles données sont intégrés dans l'application.

## Références

- [1] Value, “Value digital services.” <https://www.value.com.tn/>. (En ligne, consulté le 20 Mars 2022).
- [2] N. HOTZ, “What is tdsp ?.” <https://www.datascience-pm.com/tdsp/>. (En ligne, consulté le 18 Mars 2022).
- [3] C. Jayathilaka, “Waterfall methodology.” <https://medium.com/@chathmini96/waterfall-vs-agile-methodology-28001a9ca487>. (En ligne, consulté le 18 Mars 2022).
- [4] N. HOTZ, “What is crisp dm ?.” <https://www.datascience-pm.com/crisp-dm-2/>. (En ligne, consulté le 18 Mars 2022).
- [5] N. HOTZ, “Data science methodology and approach.” <https://www.geeksforgeeks.org/data-science-methodology-and-approach/>. (En ligne, consulté le 18 Mars 2022).
- [6] Y. Benzaki, “Logistic regression pour machine learning – une introduction simple.” <https://www.homecredit.net/about-us.aspx>. (En ligne, consulté le 30 Avril 2022).
- [7] A. Lima, “Bagging vs boosting dans le machine learning.” <https://fr.acervolima.com/bagging-vs-boosting-dans-le-machine-learning/>. (En ligne, consulté le 10 Juin 2022).
- [8] A. Navlani, “Understanding random forests classifiers in python tutorial.” <https://www.datacamp.com/tutorial/random-forests-classifier-python>. (En ligne, consulté le 3 Juin 2022).
- [9] P. Mandot, “What is lightgbm, how to implement it? how to fine tune the parameters ?.” <https://medium.com/@pushkarmandot/https-medium-com-pushkarmandot-what-is-lightgbm-how-to-implement-it-how-to-fine-tune-the-parameters-7e7e7e7e7e7e>. (En ligne, consulté le 5 Juin 2022).
- [10] S. IYYER, “Deep tutorial 1 ann and classification.” <https://www.kaggle.com/code/shrutimechlearn/deep-tutorial-1-ann-and-classification/notebook>. (En ligne, consulté le 8 Juin 2022).
- [11] “An end to end machine learning experience – scikit learn to python flask.” <https://francisjohnpicaso.wordpress.com/2018/10/14/>



- an-end-to-end-machine-learning-experience-scikit-learn-to-python-flask-part-  
(En ligne, consulté le 1 Juin 2022).
- [12] D. Gaultier, “Les 5 pratiques clés de la data science.” <https://fr.blog.businessdecision.com/les-5-pratiques-cles-data-science/>. (En ligne, consulté le 17 Mars 2022).
- [13] I. Martinez, E. Viles, and I. G. Olaizola, “Data science methodologies : Current challenges and future approaches,” *Big Data Research*, vol. 24, p. 100183, 2021.
- [14] P. Chapman, J. Clinton, R. Kerber, T. Khabaza, T. P. Reinartz, C. Shearer, and R. Wirth, “Crisp-dm 1.0 : Step-by-step data mining guide,” 2000.
- [15] D. Gaultier, “Méthode crisp : la clé de la réussite en data science.” <https://fr.blog.businessdecision.com/methode-crisp-la-cle-de-la-reussite-en-data-science/>. (En ligne, consulté le 21 Mars 2022).
- [16] J. KAGAN, “Financial technology – fintech.” <https://www.investopedia.com/terms/f/fintech.asp>. (En ligne, consulté le 20 Avril 2022).
- [17] V. Castro, “Macroeconomic determinants of the credit risk in the banking system : The case of the gipsi,” *Economic Modelling*, vol. 31, pp. 672–683, 2013.
- [18] T. Van Gestel and B. Baesens, *Credit Risk Management : Basic concepts : Financial risk components, Rating analysis, models, economic and regulatory capital*. Oxford University Press, 2009.
- [19] L. Cao, “Data science : a comprehensive overview,” *ACM Computing Surveys (CSUR)*, vol. 50, no. 3, pp. 1–42, 2017.
- [20] B. Recrutement, “A quoi sert la data science dans le secteur bancaire?.” [https://blogrecrutement.bpce.fr/-a-quoi-sert-data-science-dans-domaine-bancaire-?fbclid=IwAR168kZpRTWuMKcaZE0v9\\_rlp\\_-CCxGan\\_rhNE0r7SJscShVpKIBRBv3Hwo](https://blogrecrutement.bpce.fr/-a-quoi-sert-data-science-dans-domaine-bancaire-?fbclid=IwAR168kZpRTWuMKcaZE0v9_rlp_-CCxGan_rhNE0r7SJscShVpKIBRBv3Hwo). (En ligne, consulté le 21 Avril 2022).
- [21] T. Lynn, J. G. Mooney, P. Rosati, and M. Cummins, *Disrupting finance : Fin-Tech and strategy in the 21st century*. Springer Nature, 2019.
- [22] H. C. I. a.s., “About us.” <https://www.homecredit.net/about-us.aspx>. (En ligne, consulté le 10 Avril 2022).
- [23] A. S. Lafuente, “Exploratory data analysis with pandas profiling.” <https://towardsdatascience.com/exploratory-data-analysis-with-pandas-profiling-de3aae2ddff3>. (En ligne, consulté le 01 Juin 2022).
- [24] A. Bénard, “Algorithme n°1 - comprendre ce qu’est un arbre de décision en 5 min.” <https://blog.ysance.com/algorithme-n1-comprendre-ce-quest-un-arbre-de-decision-en-5-min>. (En ligne, consulté le 30 Avril 2022).
- [25] guest blog, “An end-to-end guide to understand the math behind xgboost.” <https://www.analyticsvidhya.com/blog/2018/09/an-end-to-end-guide-to-understand-the-math-behind-xgboost/>. (En ligne, consulté le 8 Juin 2022).



- [26] “Python : définition et utilisation de ce langage informatique.” <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1445304-python-definition-et-utilisation-de-ce-langage-informatique/>. (En ligne, consulté le 10 Juin 2022).
- [27] “Introduction to flask.” <https://pymbook.readthedocs.io/en/latest/flask.html>. (En ligne, consulté le 10 Juin 2022).
- [28] “Introduction to dash.” <https://dash.plotly.com/introduction>. (En ligne, consulté le 10 Juin 2022).
- [29] “definition mysql.” <https://www.lemagit.fr/definition/MySQL>. (En ligne, consulté le 10 Juin 2022).
- [30] “Html5 (hypertext markup language 5) : définition de ce langage informatique.” <https://www.journaldunet.fr/web-tech/dictionnaire-du-webmastering/1203257-html5-hypertext-markup-langage5-definition-traduction/>. (En ligne, consulté le 10 Juin 2022).
- [31] “Bootstrap :définition, tutoriels, astuces, pratiques.” [journaldunet.com/web-tech/developpeur/1159810-bootstrap-definition-tutoriels-astuces-pratiques/](https://www.journaldunet.com/web-tech/developpeur/1159810-bootstrap-definition-tutoriels-astuces-pratiques/). (En ligne, consulté le 10 Juin 2022).