



République Tunisienne

Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

Université de Carthage- Ecole Supérieure de la Statistique et de l'Analyse de l'Information



Rapport de Projet de Fin d'Études présenté pour l'obtention du
Diplôme National d'Ingénieur en Statistique et Analyse de l'Information



Abdennacer BOUABID

Modélisation des Paramètres de Risque de Crédit : Probabilité
de Défaut (PD) et Perte en Cas de Défaut (LGD)

Soutenu le 25/09/2024 devant le Jury composé de :

- Nom et prénom du member 1, Grade, Affiliation (Président)
- Nom et prénom du member 2, Grade, Affiliation (Rapporteur)
- Mr Marwen ben nasr (Encadrant entreprise)
- Mme Leila Zaghdoud (Encadrant entreprise)
- Mme Tasnim Hamdeni (Encadrant universitaire)

Stage de Fin d'Études effectué à Banque Zitouna



مصرف الزيتونة
BANQUE ZITOUNA

Année universitaire 2023-2024

Une dédicace À mes chers parents, Salem Bouabid et Souad Bouabid, dont Leur amour inconditionnel, leur soutien et leur innombrables sacrifices ont ouvert la voie de ma réussite.

À travers ce travail, je rends hommage à votre présence constante et à vos précieux conseils qui ont été mes solides fondations. Je vous exprime ma gratitude éternelle pour les valeurs, l'éducation, et l'encouragement que vous m'avez transmis. Je dédie également ce travail à mon frère Mohamed, à mes sœurs Dorra et Dorsaf, ainsi qu'à toute ma famille, qui a été une source continue d'inspiration et de soutien. Leur exemple a éclairé mon chemin à chaque étape de ma vie. Enfin, je tiens à adresser mes remerciements à tous mes professeurs et enseignants dont le dévouement et la qualité de l'enseignement se reflètent dans ce travail.

Remerciements

Je souhaite tout d'abord exprimer mes plus vifs et sincères remerciements envers tous ceux qui m'ont aidé et soutenu tout au long de ce projet. Leur aide et leur intérêt pour mon projet de fin d'études ont été essentiels à mon avancement et L'achèvement de ce projet n'aurait pas été possible sans leur aide et encouragement .

Je tiens en premier lieu à exprimer toute ma gratitude à Mme. Leila ZAGHDOUD et M. Marwen BEN NASR d'avoir accepté de superviser mon projet. Malgré leurs nombreuses responsabilités, ils ont pris le temps de m'accompagner et de me conseiller. Je les remercie pour leur gentillesse et leur soutien. C'est également grâce à ses conseils et à que j'ai pu m'accomplir totalement dans mes missions.

Je tiens également à remercier mon encadrante universitaire Mme. Tasnim Hamdeni pour sa supervision de ce projet de fin d'études. Ses conseils, son suivi attentif et la valeur ajoutée qu'elle a apportée ont été d'une grande aide tout au long de mon parcours. Je souhaite également remercier tous mes amis, dont la présence a été une source constante de soutien et d'encouragement durant cette période.

Enfin, j'apprécie la présence de .. en tant que président du jury et de .. en tant que rapporteuse de ce travail.

Résumé

Dans un contexte marqué par des exigences réglementaires accrues et une concurrence intense, ce rapport se concentre sur l'amélioration des processus décisionnels de crédit à la Banque Zitouna, une institution spécialisée dans la finance islamique en Tunisie. L'objectif principal est de développer des modèles prédictifs robustes pour estimer la Probabilité de Défaut (PD) et la Perte en Cas de Défaut (LGD), afin de renforcer la gestion des risques de crédit et d'assurer une conformité accrue aux normes réglementaires.

En utilisant un portefeuille de données historiques et comportementales des clients de la banque, nous avons mis en œuvre une approche rigoureuse combinant des techniques avancées de machine learning et d'intelligence artificielle. Après une évaluation approfondie des performances, le modèle XGBoost a été identifié comme le plus performant pour prédire la PD, tandis qu'une approche pratique de type "LGD Workout" a été adoptée pour estimer la LGD.

Une application web a également été développée pour intégrer ces modèles et offrir des résultats dynamiques et interactifs. Cette solution contribue à l'amélioration de la performance économique de la Banque Zitouna, tout en optimisant sa capacité à gérer efficacement les risques de crédit dans un environnement en constante évolution.

Mots clés : Intelligence artificielle, probabilité de défaut, perte en cas de défaut, machine learning, LGD Workout, ré-équilibrage, XGBoost.

Abstract

In a context marked by heightened regulatory requirements and intense competition, this report focuses on enhancing the credit decision-making processes at Banque Zitouna, an institution specialized in Islamic finance in Tunisia. The primary objective is to develop robust predictive models to estimate the Probability of Default (PD) and Loss Given Default (LGD), thereby strengthening credit risk management and ensuring greater compliance with regulatory standards.

Using a portfolio of historical and behavioral data from the bank's clients, we implemented a rigorous approach that combines advanced machine learning and artificial intelligence techniques. After a thorough evaluation of performance, the XGBoost model was identified as the most effective for predicting PD, while a practical "LGD Workout" approach was adopted to estimate LGD.

A web application was also developed to integrate these models and provide dynamic and interactive results. This solution contributes to improving the economic performance of Banque Zitouna while optimizing its ability to manage credit risks effectively in a constantly evolving environment.

Keywords: Artificial intelligence, probability of default, loss given default, machine learning, LGD Workout, rebalancing, XGBoost.

Table des matières

Introduction Générale	1
1. Présentation générale de projet	3
1.1 Introduction	3
1.2 Présentation de l'Organisme d'Accueil	3
1.3 Présentation du Projet	4
1.3.1 Contexte	4
1.3.2 Objectifs	5
1.3.3 Problématique	5
1.4 Définitions et Notions de Bases	6
1.4.1 La Norme IFRS 9	6
1.4.2 Transition de l'IAS 39 à l'IFRS 9	7
1.4.3 Paramètres des Risques	7
1.4.4 Workout LGD	9
1.4.5 Méthode Chain-Ladder	11
1.4.6 Intelligence Artificielle	12
1.4.7 L'Apprentissage Automatique	13
1.5 Environnement de Travail	15
1.6 Méthodologie de Travail	15
1.7 Plan de Travail	16
1.8 Conclusion	16

2. Compréhension du Projet	17
2.1 Introduction	17
2.2 Sélection des Variables	17
2.3 Manipulation des Variables Catégorielles	19
2.3.1 Encodage des Variables Catégorielles	19
2.3.2 Méthode One Hot Encoder	19
2.3.3 Méthode Label Encoder	20
2.4 Traitement de Déséquilibre de Classes	20
2.4.1 Techniques de rééchantillonnage	21
2.5 Augmentation des Données	21
2.5.1 Méthode d'Injection de Bruit Aléatoire	22
2.6 Les modèles de Machine Learning	22
2.6.1 Régression Logistique	23
2.6.2 Bagging	25
2.6.3 Boosting	27
2.7 Mesures de Qualité de l'Ajustement du Modèle	30
2.8 Réglage des hyperparamètres	30
2.9 Interprétation des Variables	31
2.9.1 Méthode SHAP	31
2.9.2 Méthode LIME	32
2.10 Framework Django	34
2.10.1 Caractéristiques principales de Django	35
2.10.2 Backend de Django (Python)	36
2.10.3 Frontend de Django (HTML, CSS, et JavaScript)	36
2.11 Conclusion	36
3. EXPLORATION DES DONNEES	37
3.1 Introduction	37

3.2	Perte en Cas de Défaut	37
3.2.1	Collecte et Présentation de la Base de Données	37
3.2.2	Hypothèses et Règles de Gestion	39
3.2.3	Calcul des LGD pour chaque financement	39
3.2.4	Analyse Des Données	39
3.3	Probabilité de Défaut	42
3.3.1	Présentation de la Base de Données	42
3.3.2	Analyse de Données	46
3.3.3	Traitement des Variables	50
3.3.4	Équilibrage des Données	52
3.3.5	Augmentation des Données	53
3.3.6	Sélection des Variables	53
3.4	Conclusion	55
4.	EVALUATION ET IMPLEMENTATION DES MODELES	56
4.1	Introduction	56
4.2	Perte en Cas de Défaut	57
4.2.1	Résultat du Calcul	57
4.2.2	Dashboard	57
4.3	Probabilité de Défaut	58
4.3.1	Travail Préliminaire	58
4.3.2	Résultats Empiriques	59
4.3.3	Comparaison des Modèles Appliqués	67
4.3.4	Interprétabilité des Modèles	70
4.4	Application Web	74
4.4.1	Interface d'Authentification	74
4.4.2	Page d'Accueil	75
4.4.3	Prédiction de la Probabilité de Défaut(PD)	76

4.4.4 Estimation de la Perte en Cas de Défaut(LGD)	77
4.5 Conclusion	78
Conclusion Générale	80
Annexe A	81
Annexe B	83
Annexe C	84
Annexe D	87

Table des figures

1.1	Logo de la banque Zitouna (Source : banquezitouna.com/fr)	4
1.2	Domaines d'intelligence artificielle (Source : bial-x.com)	13
1.3	Types d'apprentissage automatique (Source : deeplogicaitech.com)	14
1.4	Logo Jupiter Notebook (Source : jupyter.org)	15
1.5	Diagramme de Gantt	16
2.1	Courbe de la fonction logit (Source : mixture.fr)	23
2.2	Principe de fonctionnement des forêts aléatoires (Source : sciviews.org)	27
2.3	Logo Django (Source : djangoproject.com)	35
3.1	Distribution du LGD	40
3.2	Répartition des catégories	41
3.3	Répartition de maturité	41
3.4	Distribution de la variable cible	46
3.5	Répartition des clients entreprises selon leurs secteurs d'activité	47
3.6	Distribution des clients selon leur total d'engagement	48
3.7	Matrice de corrélation : Spearman	48
3.8	Pourcentage des données manquantes	51
4.1	Statistiques descriptives de la LGD	57
4.2	Interface du Dashboard	58
4.3	Courbe de ROC pour le modèle XGBoost	69
4.4	Importance des variables dans le modèle XGBoost	70

4.5	Impact des variables sur le modèle selon SHAP	71
4.6	Explication LIME des prédictions du modèle	72
4.7	Page d'authentification	74
4.8	Page d'accueil	75
4.9	Page de prédiction PD	76
4.10	Page de résultat de prédiction PD	77
4.11	Page d'estimation LGD	77
4.12	Page de résultat d'estimation LGD	78

Liste des tableaux

1.1	Triangle de liquidation des récupérations observées avec complétion des récupérations futures	12
3.1	Corrélation entre les variables qualitatives et la variable cible	49
3.2	Corrélation entre les variables quantitatives et la variable cible	50
3.3	Résultats des méthodes d'équilibrage des données	53
3.4	Liste des variables sélectionnées selon SelectKbest	54
3.5	Liste des variables sélectionnées selon stepwise	54
4.1	Résultats des métriques pour la régression logistique avec OneHotEncoder	60
4.2	Résultats des métriques pour la régression logistique avec LabelEncoder .	60
4.3	Hyperparamètres pour Arbre de Décision	61
4.4	Résultats des métriques pour l'Arbre de décision avec OneHotEncoder . .	61
4.5	Résultats des métriques pour l'Arbre de décision avec LabelEncoder . . .	62
4.6	Hyperparamètres pour Forêt Aléatoire	62
4.7	Résultats des métriques pour la Forêt Aléatoire avec OneHotEncoder . .	63
4.8	Résultats des métriques pour la Forêt Aléatoire avec LabelEncoder . . .	63
4.9	Hyperparamètres pour XGBoost	64
4.10	Résultats des métriques pour XGBoost avec OneHotEncoder	64
4.11	Résultats des métriques pour XGBoost avec LabelEncoder	64
4.12	Hyperparamètres pour LightGBM	65
4.13	Résultats des métriques pour LightGBM avec OneHotEncoder	65
4.14	Résultats des métriques pour LightGBM avec LabelEncoder	66

4.15	Hyperparamètres pour AdaBoost	66
4.16	Résultats des métriques pour AdaBoost avec OneHotEncoder	67
4.17	Résultats des métriques pour AdaBoost avec LabelEncoder	67
4.18	Mesures de performance des différents modèles	68
.19	Synthèse des Hypothèses et Règles de Gestion	83

Introduction Générale

Le projet, intitulé "Modélisation des paramètres des risques LGD et PD", répond à la nécessité de créer un outil capable de prédire avec précision la Probabilité de Défaut (PD) et d'estimer la perte en cas de défaut (LGD) conformément aux exigences de la Banque Centrale de Tunisie pour la norme IFRS 9.

Ce projet de fin d'études vise à développer des modèles prédictifs pour estimer la Probabilité de Défaut (PD) et à utiliser une méthode rigoureuse pour estimer la Perte en Cas de Défaut (LGD), afin de renforcer la capacité de la banque à gérer efficacement les risques de crédit.

Pour atteindre cet objectif, nous avons recueilli et analysé des données historiques et comportementales des clients de la Banque Zitouna. En utilisant des techniques avancées d'intelligence artificielle et de machine learning, nous avons développé et validé plusieurs modèles prédictifs pour la Probabilité de Défaut (PD). Pour l'estimation de la Perte en Cas de Défaut (LGD), nous avons appliqué la méthode LGD Workout, qui repose sur le calcul de la valeur actuelle nette des flux de trésorerie de récupération.

Le projet est structuré en quatre chapitres principaux. Le premier chapitre présente l'organisme d'accueil, le cadre général du projet et les concepts de base de l'étude. Le deuxième chapitre explore les mécanismes de fonctionnement des différents modèles de machine learning utilisés pour la PD, ainsi que la méthode LGD Workout. Il aborde également les techniques de prétraitement des données et les mesures de performance employées pour sélectionner le modèle le plus robuste. Le troisième chapitre est consacré à l'analyse exploratoire des données, détaillant les étapes de nettoyage et de préparation

des données, ainsi que les méthodes de sélection des variables, tant pour la PD que pour la LGD. Enfin, le quatrième chapitre présente les résultats obtenus des modèles de machine learning pour la PD, les estimations de la LGD à l'aide de la méthode LGD Workout, et décrit le développement d'une application web intuitive permettant une utilisation pratique et interactive des résultats par les décideurs de la banque.

Chapitre 1

Présentation générale de projet

1.1 Introduction

Ce premier chapitre offre une vue d'ensemble du contexte général du projet de fin d'études. Nous débutons par une présentation de l'organisme d'accueil, puis nous exposons la problématique et les objectifs visés. Ensuite, nous allons définir quelques notions de base. Enfin, nous concluons en détaillant le planning et la méthodologie adoptés pour ce travail.

1.2 Présentation de l'Organisme d'Accueil

La Banque Zitouna, fondée en 2009, est une institution bancaire tunisienne spécialisée dans la finance islamique. Depuis son ouverture au public le 28 mai 2010, la banque s'est engagée à offrir des produits et services financiers conformes aux principes de la charia. Cette orientation permet à la banque de répondre aux besoins spécifiques de ses clients en matière de finance éthique et respectueuse des valeurs islamiques.

Dès ses débuts, la Banque Zitouna a mis en place des solutions bancaires innovantes, alliant performance économique et conformité religieuse. Son engagement envers la finance islamique se reflète dans tous les aspects de ses opérations, de la conception des produits financiers à la gestion des transactions.

Cependant, la banque a traversé des périodes de défis. En janvier 2011, la Banque Centrale de Tunisie a pris en charge la gestion provisoire de la Banque Zitouna, marquant une phase de transition et de réorganisation. Malgré ces obstacles, la banque a continué à maintenir ses valeurs fondamentales et à offrir des services bancaires fiables et conformes à la charia.

Un changement significatif est survenu en 2018, lorsque le gouvernement tunisien a décidé de vendre sa participation majoritaire de 69,15 % dans la banque au groupe qatari Majda. Cette acquisition a renforcé la position de la Banque Zitouna sur le marché et a ouvert de nouvelles perspectives de croissance et d'expansion.

Aujourd'hui, la Banque Zitouna se distingue par son engagement envers l'éthique et l'innovation dans le secteur financier tunisien. Elle continue de jouer un rôle crucial dans le développement économique de la Tunisie en offrant des solutions bancaires qui respectent les convictions religieuses de ses clients tout en assurant une performance économique durable.



Figure 1.1 – Logo de la banque Zitouna (Source : banquezitouna.com/fr)

1.3 Présentation du Projet

1.3.1 Contexte

La Banque Zitouna cherche à améliorer son processus décisionnel de crédit en s'appuyant sur les données historiques, les informations personnelles et les comportements

de ses clients. Dans ce cadre, ce projet de fin d'études vise à développer des modèles prédictifs pour l'estimation de la Perte en cas de défaut (LGD) et de la Probabilité de défaut (PD) et la validation de leur précision. En d'autres termes, l'objectif de la banque est de renforcer sa capacité à gérer efficacement les risques de crédit pour évaluer ses clients selon leur habilité de remboursement.

1.3.2 Objectifs

L'objectif de ce projet de fin d'études est de développer des modèles de modélisation des paramètres de risque de crédit, notamment la Probabilité de Défaut (PD) et la Perte en cas de Défaut (LGD). Dans un premier temps, nous entreprendrons une analyse approfondie des données historiques des clients de la Banque Zitouna ainsi que leurs comportements pendant une durée précise et leurs informations personnelles pour comprendre les facteurs qui influencent ces paramètres de risque. En utilisant des techniques avancées d'intelligence artificielle et de machine Learning, nous créerons des modèles prédictifs pour estimer la PD et la LGD pour chaque client. En outre, nous cherchons à développer des outils de visualisation pour présenter de manière claire et compréhensible les résultats de nos modèles, permettant ainsi aux décideurs de prendre des décisions éclairées en matière de gestion du risque de crédit. Enfin, nous visons à développer une application web ou mobile pour présenter les résultats de nos modèles et aider la Banque Zitouna à mieux gérer son risque de crédit.

1.3.3 Problématique

L'adoption de la norme IFRS 9 a imposé des exigences plus strictes et sophistiquées pour la gestion du risque de crédit dans les institutions financières. En remplaçant l'approche réactive de l'IAS 39 par une méthode proactive fondée sur les pertes de crédit attendues (ECL), cette transition nécessite une expertise avancée en modélisation statistique et l'intégration de divers paramètres de risque.

Cependant, la complexité et le temps nécessaires pour évaluer les dossiers de crédit, associés à l'augmentation du volume de demandes, posent des défis importants pour les banques. Pour cette raison, un nombre croissant d'institutions financières se tournent vers les algorithmes d'apprentissage automatique afin d'automatiser et d'optimiser le processus de notation de crédit. Ces algorithmes peuvent potentiellement réduire les délais de traitement et améliorer la précision des évaluations de risque.

Dans ce contexte, la banque Zitounasouhaite intégrer la modélisation des Pertes en cas de Défaut (LGD) et des Probabilités de Défaut (PD) dans ses systèmes. Cela permettra non seulement d'affiner l'évaluation des risques de crédit, mais aussi de prévoir avec une plus grande précision les pertes potentielles en cas de défaut de paiement des clients.

1.4 Définitions et Notions de Bases

1.4.1 La Norme IFRS 9

La norme IFRS (International Financial Reporting Standards) est une nouvelle norme comptable et financière adoptée à l'échelle internationale pour établir de nouvelles règles, marquant une évolution significative par rapport aux anciennes méthodes de classification. Son objectif est de créer un système financier plus stable, permettant une comptabilisation plus rapide et plus efficace des pertes de crédit. Les origines de la norme IFRS 9 remontent à la crise financière de 2008 [1], lorsque les règles de provisionnement en vigueur, définies par la norme IAS 39, ont été jugées trop complexes et insuffisantes. Pour instaurer un système financier plus résilient, la norme IFRS 9 a été introduite en juillet 2014 et est appliquée mondialement depuis 2018. Selon IFRS 9, les banques doivent comptabiliser la probabilité de défaut sur une période de crédit complète de 12 mois. Bien que toutes les banques possèdent actuellement des systèmes de notation interne (SNI), leur niveau de maturité varie, et certaines institutions disposent de SNI pas suffisamment développés pour être utilisés dans le cadre d'IFRS 9.

1.4.2 Transition de l'IAS 39 à l'IFRS 9

La transition de l'IAS 39 à l'IFRS 9 a été motivée par les critiques de la complexité et de l'inefficacité de l'IAS 39, notamment sa dépendance à un modèle de pertes subies qui retardait la reconnaissance des pertes de crédit jusqu'à la survenue d'événements de perte spécifiques. Introduite en 2014 et appliquée depuis 2018, l'IFRS 9 a remplacé l'IAS 39 pour offrir une approche plus proactive avec un modèle de pertes attendues, obligeant les banques à comptabiliser les pertes de crédit potentielles sur une période de 12 mois, même en l'absence d'événements de perte identifiés.

En outre, l'IFRS 9 simplifie la classification et l'évaluation des actifs financiers en les regroupant en trois catégories principales contrairement à l'IAS 39, qui avait plusieurs catégories avec des règles complexes pour le reclassement, l'IFRS 9 permet un reclassement uniquement lorsque le modèle d'affaires de gestion des actifs change.

De plus, l'IFRS 9 introduit un modèle unique de dépréciation basé sur les pertes de crédit attendues, remplaçant le modèle de dépréciation de l'IAS 39 qui était basé sur les pertes de crédit encourues. Cela permet une reconnaissance plus rapide des pertes potentielles et améliore ainsi la résilience et la stabilité du système financier mondial. Adoptée par des institutions financières dans plus de 110 pays, l'IFRS 9[2] vise à aligner la comptabilisation des pertes de crédit avec la gestion des risques des entités, alors que les États-Unis utilisent la méthode CECL[3].

1.4.3 Paramètres des Risques

- **Pertes de Crédit Attendues ECL**

La perte de crédit attendue ou Expected Credit Loss (ECL), représente la perte moyenne estimée qu'une institution financière s'attend à subir sur une période déterminée en raison d'événements de défaut sur ses actifs financiers.

La formule de la Perte Crédit Espérée peut être exprimée comme suit :

$$ECL = PD \times LGD \times EAD \quad (1.1)$$

Où :

ECL : Expected Credit Loss (Perte de crédit attendue)

PD : Probability of Default (Probabilité de défaut)

EAD : Exposure At Default (Exposition en cas de défaut)

LGD : Loss Given Default (Pertes en cas de défaut)

- **Probabilité de Défaut(PD)**

La probabilité de défaut représente la probabilité qu'un client ne puisse pas respecter ses obligations de remboursement. Cette probabilité est dynamique et peut être affectée par de multiples facteurs de risque, qu'ils soient spécifiques au client ou systémiques liés à l'environnement du client. C'est pourquoi la probabilité de défaut est considérée comme une variable aléatoire que l'on peut modéliser et estimer afin d'analyser et de prédire le risque de défaut d'une contrepartie.

- **Pertes en cas de Défaut(LGD)**

LGD (Loss Given Default), ou perte en cas de défaut, représente la mesure de la perte potentielle en cas de défaut. Cette mesure est exprimée en pourcentage et correspond à la part de l'exposition totale qui n'est pas récupérée en cas de défaut, en prenant en compte les actifs du client et les garanties éventuellement fournies dans le cadre du contrat de prêt.

- **Exposition en cas de Défaut (EAD)**

L'exposition en cas de défaut désigne le montant que la contrepartie est engagée à perdre lorsqu'un événement de défaut se produit. Elle représente ainsi la perte maximale potentielle lorsqu'un événement de défaut survient.

1.4.4 Workout LGD

- **Processus de "Workout"**

Le processus de "Workout" dans la gestion des risques se réfère aux actions entreprises pour maximiser la récupération des prêts non performants (NPL) ou des actifs en difficulté après un défaut. Cela commence par l'identification des NPL et l'évaluation de leur situation, y compris la valeur des garanties. Ensuite, des stratégies de restructuration personnalisées sont développées, souvent en négociant avec les emprunteurs pour établir des modalités de remboursement mutuellement acceptables. Ces plans sont mis en œuvre avec un suivi rigoureux et des ajustements effectués si nécessaire. L'objectif principal est de réduire les pertes en maximisant les recouvrements. Une communication claire et un engagement proactif avec les emprunteurs sont essentiels pour garantir le succès du processus de workout. Ce processus contribue à améliorer la stabilité financière des institutions en minimisant les impacts négatifs des défauts de paiement.

- **Workout LGD**

Le "Workout LGD" fait référence à l'estimation de la perte en cas de défaut (LGD) pendant la restructuration des prêts non performants et d'autres instruments de crédit. Il quantifie la part du montant en défaut que les prêteurs ou créanciers risquent de perdre après avoir mis en place diverses stratégies de recouvrement. L'analyse des prêts non performants nécessite souvent l'estimation du Workout LGD pour une gestion efficace des risques. Cette évaluation permet aux institutions financières de mesurer les pertes potentielles liées à leurs portefeuilles de crédit et d'informer leurs décisions sur les provisions, l'allocation de capital, et les stratégies d'atténuation des risques. Une bonne compréhension du Workout LGD permet aux prêteurs de mieux gérer les actifs non performants et de maximiser les recouvrements lors des restructurations.

- **Calcul de Workout LGD pour un Défaut**

La méthodologie de workout LGD est basée sur le calcul de la valeur actuelle nette des flux de trésorerie de récupération.

La formule de calcul du LGD peut être exprimée comme suit :

$$LGD = 1 - RR \quad (1.2)$$

$$RR = \frac{\sum_{i=1}^n VAN(\text{recup}, a_i) + VAN(\text{Fluxartificiel}, a_i)}{EAD} \quad (1.3)$$

avec :

LGD représente la perte en cas de défaut, tandis que RR (Recovery Rate) désigne le taux de recouvrement. VAN(Recup) correspond à la valeur actuelle nette des recouvrements, VAN(Fluxartificiel) à la valeur actuelle nette de l'engagement au moment de la sortie de défaut, et EAD à l'exposition en cas de défaut.

Et telque :

a_i : Taux d'actualisation

$$\text{Recup} = (\text{Engagement}_t - \text{Engagement}_{t+1}) + \max(\text{Impayenprofit}_{t+1} - \text{Impayenprofit}_t, 0) \quad (1.4)$$

$$\text{Engagement}_t = \text{Encours}_t + \text{Impayé}_t \quad (1.5)$$

$$EAD = \text{Engagement}_d \quad (1.6)$$

Remarques

- Si le client est encore en défaut, alors on projette les récupérations jusqu'à la date de fin de financement en suivant la méthode Chain-Ladder.

- Si le client est encore en défaut et que la période de défaut dépasse 7 ans, alors le taux de recouvrement (RR) est égal à 0.
- Si le client subit un rééchelonnement de l'un de ses financements, alors on projette le montant du rééchelonnement après l'année de fin de financement initiale, avant le rééchelonnement, et on calcule selon la formule générale.
- Tous les clients qui passent en contentieux seront considérés en déchéance du terme.
- Si on a plusieurs défauts pour un seul financement, la LGD de ce financements est la moyenne des LGDs pour ces défauts.

1.4.5 Méthode Chain-Ladder

Le prolongement par Chain-Ladder consiste à estimer les récupérations futures non observées en se basant sur les récupérations déjà observées et réalisées. L'algorithme Chain-Ladder est une approche traditionnelle, intuitive et facile à mettre en œuvre [4]. Fondamentalement, la méthode Chain-Ladder opère sur l'hypothèse que les récupérations cumulées futures seront similaires à celles observées dans le passé. Pour que cette hypothèse soit valide, il est essentiel que les données historiques de récupération soient fiables et précises. Plusieurs facteurs peuvent influencer cette précision, notamment les changements dans les produits financiers de la banque, les modifications réglementaires et juridiques, ainsi que les évolutions dans les processus de recouvrement.

L'approche repose sur l'observation de l'évolution des récupérations cumulées actualisées au moment du défaut pour chaque défaut identifié (ID_D_i), en supposant que les schémas de récupération futurs seront similaires à ceux du passé.

Selon cette méthode de complétion, la récupération future sur un défaut est égale à la récupération passée actualisée au moment du défaut multipliée par un coefficient multiplicatif près, dit facteur de développement, noté f_k .

Les facteurs de développement sont obtenus avec la formule suivante :

$$f_k = \frac{\sum_{i=1}^{n-k+1} R_{i,k}}{\sum_{i=1}^{n-k+1} R_{i,k-1}} \quad (1.7)$$

Ainsi les récupérations cumulées futures estimées (et actualisées) sont obtenues avec la formule suivantes :

$$R_{i,j} = R_{i,n-i+1} \prod_{k=j-i+2}^j f_k \quad (1.8)$$

A travers ces deux formules listées ci-dessus le triangle de liquidation est ainsi prolongé pour les défauts non clos comme suit :

Temps en défaut Année d'origine	1	2	3	4	5	6	7	8	9	10
ID_D_1	$R_{1,1}$	$R_{1,2}$	$R_{1,3}$	$R_{1,4}$	$R_{1,5}$	$R_{1,6}$	$R_{1,7}$	$R_{1,8}$	$R_{1,9}$	$R_{1,10}$
ID_D_2	$R_{2,1}$	$R_{2,2}$	$R_{2,3}$	$R_{2,4}$	$R_{2,5}$	$R_{2,6}$	$R_{2,7}$	$R_{2,8}$	$R_{2,9}$	$R_{2,10} \times f_{10}$
ID_D_3	$R_{3,1}$	$R_{3,2}$	$R_{3,3}$	$R_{3,4}$	$R_{3,5}$	$R_{3,6}$	$R_{3,7}$	$R_{3,8}$	$R_{3,9} \times f_9$	$R_{3,10} \times f_{10}$
ID_D_4	$R_{4,1}$	$R_{4,2}$	$R_{4,3}$	$R_{4,4}$	$R_{4,5}$	$R_{4,6}$	$R_{4,7}$	$R_{4,8} \times f_8$	$R_{4,9} \times f_9$	$R_{4,10} \times f_{10}$
ID_D_5	$R_{5,1}$	$R_{5,2}$	$R_{5,3}$	$R_{5,4}$	$R_{5,5}$	$R_{5,6}$	$R_{5,7} \times f_7$	$R_{5,8} \times f_8$	$R_{5,9} \times f_9$	$R_{5,10} \times f_{10}$
ID_D_6	$R_{6,1}$	$R_{6,2}$	$R_{6,3}$	$R_{6,4}$	$R_{6,5}$	$R_{6,6} \times f_6$	$R_{6,7} \times f_7$	$R_{6,8} \times f_8$	$R_{6,9} \times f_9$	$R_{6,10} \times f_{10}$
ID_D_7	$R_{7,1}$	$R_{7,2}$	$R_{7,3}$	$R_{7,4}$	$R_{7,5} \times f_5$	$R_{7,6} \times f_6$	$R_{7,7} \times f_7$	$R_{7,8} \times f_8$	$R_{7,9} \times f_9$	$R_{7,10} \times f_{10}$
ID_D_8	$R_{8,1}$	$R_{8,2}$	$R_{8,3}$	$R_{8,4} \times f_4$	$R_{8,5} \times f_5$	$R_{8,6} \times f_6$	$R_{8,7} \times f_7$	$R_{8,8} \times f_8$	$R_{8,9} \times f_9$	$R_{8,10} \times f_{10}$
ID_D_9	$R_{9,1}$	$R_{9,2}$	$R_{9,3} \times f_3$	$R_{9,4} \times f_4$	$R_{9,5} \times f_5$	$R_{9,6} \times f_6$	$R_{9,7} \times f_7$	$R_{9,8} \times f_8$	$R_{9,9} \times f_9$	$R_{9,10} \times f_{10}$
ID_D_10	$R_{10,1}$	$R_{10,2} \times f_2$	$R_{10,3} \times f_3$	$R_{10,4} \times f_4$	$R_{10,5} \times f_5$	$R_{10,6} \times f_6$	$R_{10,7} \times f_7$	$R_{10,8} \times f_8$	$R_{10,9} \times f_9$	$R_{10,10} \times f_{10}$
Facteur de développement	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}

Table 1.1 – *Triangle de liquidation des récupérations observées avec complétion des récupérations futures*

1.4.6 Intelligence Artificielle

L'intelligence artificielle (IA) est une technologie qui vise à doter les machines de capacités similaires à celles des humains, telles que la perception, le raisonnement et l'apprentissage. Elle repose sur des algorithmes et des modèles informatiques pour résoudre des problèmes complexes, automatiser des tâches et prendre des décisions. Les applications courantes de l'IA incluent les chatbots, les systèmes de recommandation, la vision par ordinateur et la reconnaissance vocale. L'IA est largement utilisée dans divers

secteurs tels que la santé, la finance, l'automobile et les technologies de l'information pour améliorer l'efficacité, la précision et la productivité. L'apprentissage automatique, l'apprentissage profond et les réseaux neuronaux représentent des branches spécifiques de l'intelligence artificielle, qui permettent aux machines d'apprendre à partir de données, de reconnaître des motifs complexes et de prendre des décisions autonomes.

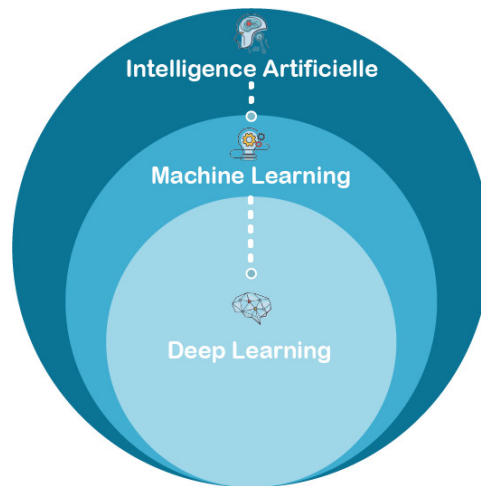


Figure 1.2 – Domaines d'intelligence artificielle (Source : bial-x.com)

1.4.7 L'Apprentissage Automatique

L'apprentissage automatique, également connu sous le nom de machine learning, est une branche de l'intelligence artificielle (IA) qui se concentre sur l'utilisation de données et d'algorithmes pour permettre aux machines d'apprendre à effectuer des tâches sans être explicitement programmées[5]. En se basant sur des échantillons de données, les algorithmes d'apprentissage automatique sont capables d'améliorer progressivement leur performance, en ajustant leurs paramètres pour optimiser leur précision. Elle s'appuie sur des techniques statistiques, probabilistes et mathématiques pour créer des modèles capables de faire des prédictions ou des classifications. Il existe 3 méthodes d'apprentissage automatique :

- **Apprentissage supervisé :** Cette méthode d'apprentissage implique l'identification des règles dans les systèmes automatisés à partir de données historiques étiquetées. Ainsi, elle permet de prédire ou de classifier de nouvelles données non étiquetées en se basant sur ces règles préalablement établies.
- **Apprentissage non supervisé :** Cette méthode est utilisée lorsque les données d'apprentissage ne sont pas préalablement classées ou étiquetées. Elle permet au système de découvrir des structures cachées à partir d'échantillons de données non étiquetées. Bien qu'elle ne puisse pas prédire de réponse précise, elle explore les données pour en extraire des interprétations.
- **Apprentissage par renforcement :** Cette méthode implique l'apprentissage basé sur l'interaction entre le système et son environnement. Le système prend des actions et apprend de ses erreurs ainsi que des résultats de ces actions pour s'améliorer au fil du temps.

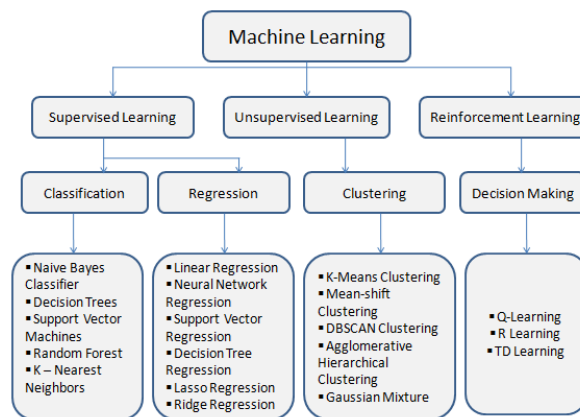


Figure 1.3 – Types d'apprentissage automatique (Source : deeplogicaitech.com)

1.5 Environnement de Travail

Tout au long de mon projet de développement, j'ai principalement utilisé Jupyter Notebook comme plateforme de travail. Jupyter Notebook offre un environnement interactif qui facilite le développement de code, la visualisation des données et la création de narration textuelle.



Figure 1.4 – *Logo Jupiter Notebook (Source : jupyter.org)*

1.6 Méthodologie de Travail

Pour maintenir une méthodologie de travail structurée et organisée, j'ai intégré MLflow dans mon processus de développement. MLflow est une plateforme qui facilite la gestion des différentes étapes de modélisation, en permettant de suivre et d'enregistrer les résultats des expériences, y compris les métriques et les hyperparamètres utilisés dans chaque itération. Cela nous a permis de suivre efficacement le progrès du développement et de l'implémentation des modèles, tout en assurant la reproductibilité et la facilité de déploiement des modèles grâce au stockage intégré. En utilisant MLflow, nous avons bénéficié d'une meilleure traçabilité et d'une gestion plus efficace de nos expériences de modélisation.

1.7 Plan de Travail

Afin de représenter visuellement la progression de notre projet de fin d'études, nous avons créé un diagramme de Gantt qui met en évidence chaque tâche accomplie ainsi que sa durée. Pour structurer notre travail et guider nos étapes d'analyse des données, nous avons également adopté l'approche CRISP-DM (Cross-Industry Standard Process for Data Mining), qui est une méthodologie largement utilisée pour la gestion de projets d'exploration de données.



Figure 1.5 – *Diagramme de Gantt*

1.8 Conclusion

Dans ce premier chapitre, nous avons débuté par une présentation de l'organisme d'accueil, la banque Zitouna, ainsi que son secteur d'activité. Nous avons défini les normes référentiels du système bancaire et les techniques d'apprentissage automatique. Nous avons également énoncé la problématique, les objectifs et la méthodologie, accompagnés d'un planning détaillé grâce au diagramme de Gantt. Ce chapitre établit les fondements de notre projet, fournissant une base solide pour la suite de notre travail.

Chapitre 2

Compréhension du Projet

2.1 Introduction

Dans ce chapitre, nous exposons les techniques de sélection des variables, les méthodes d'encodage et le mécanisme de travail de chaque modèle de machine learning que nous allons utiliser lors de la phase de modélisation[hastie2009elements].

Nous présentons également les les critères de performance ainsi que les mesures d'évaluation que nous avons sélectionnés pour évaluer nos modèles.

Enfin, nous clôturons ce chapitre en abordant quelques techniques de rééchantillonnage qui seront utiles pour traiter le déséquilibre de notre variable cible , ainsi que le réglage des hyperparamètres.

2.2 Sélection des Variables

L'objectif de sélection des variables est d'identifier un ensemble de variables qui permettra de créer le modèle le mieux adapté pour effectuer des prédictions précises.[6] En choisissant soigneusement les variables qui ont une forte corrélation avec la variable cible, nous cherchons à améliorer les performances du modèle tout en maintenant une interprétabilité élevée et des temps d'entraînement rapides.

Les quatre options de cette sélection sont : en avant (**Forward**), en arrière (**Backward**), pas à pas (**stepwise**) et **SelectKBest** :

- **Forward Feature Selection (FFS)** : Cette technique commence avec un modèle vide puis essaie d'ajouter les variables une par une au modèle, ne sélectionnant que celles qui améliorent les performances du modèle. Ce processus se poursuit jusqu'à ce qu'aucune variable non sélectionnée n'améliore significativement les performances du modèle.
- **Backward Feature Selection (BFS)** : À l'opposé de la FFS, cette méthode commence avec toutes les variables incluses dans le modèle puis les élimine une par une en fonction de leur significativité statistique, ne conservant que celles qui contribuent de manière significative aux résultats du modèle.
- **Recursive Feature Selection (ou Stepwise Selection)** : Cette méthode est une combinaison des approches FFS et BFS. Elle essaie différentes combinaisons de variables pour trouver la plus pertinente. Elle commence souvent par un modèle vide puis ajoute et supprime des variables à chaque étape, en fonction de leur contribution aux performances du modèle, jusqu'à ce qu'aucune autre amélioration significative ne soit possible.
- **SelectKBest** : Contrairement aux méthodes de sélection de caractéristiques comme la Forward Feature Selection (FFS) et la Backward Feature Selection (BFS), qui impliquent des processus itératifs pour ajouter ou éliminer des variables en fonction de leur contribution au modèle, le SelectKBest adopte une approche plus directe. Au lieu de procéder par étapes successives, cette technique choisit directement les 'k' variables les plus pertinentes en se basant sur une mesure statistique prédéfinie. Le SelectKBest évalue chaque caractéristique en utilisant une fonction de score, telle que la corrélation ou le test de chi-deux, pour déterminer son importance relative. Il sélectionne ensuite les k caractéristiques avec les scores les plus élevés,

excluant les autres. Cette approche est souvent utilisée pour simplifier les modèles en ne conservant que les caractéristiques les plus significatives, ce qui peut améliorer la performance du modèle et réduire le temps de calcul.

2.3 Manipulation des Variables Catégorielles

La présence de variables catégorielles dans les données peut rendre l'apprentissage plus complexe. La plupart des modèles d'apprentissage automatique requièrent des données numériques en entrée, ce qui nécessite de trouver des méthodes pour convertir les différentes modalités de ces variables en valeurs numériques.

2.3.1 Encodage des Variables Catégorielles

L'encodage des variables catégorielles consiste à transformer les données catégorielles en une forme que les algorithmes de machine learning peuvent traiter. Cela implique souvent de convertir les catégories en nombres, comme dans le cas du One Hot Encoder ou Label Encoding. Ce processus est essentiel pour permettre aux modèles de machine learning de comprendre et d'utiliser efficacement ces données dans leurs prédictions. Pour un modèle donné, une méthode d'encodage peut être meilleure qu'une autre, donc Nous devons accorder une attention particulière à cette étape cruciale de transformation.

2.3.2 Méthode One Hot Encoder

Le One Hot Encoder est une méthode largement adoptée pour convertir les variables catégorielles en variables numériques. Elle est couramment utilisée car elle produit généralement des résultats satisfaisants et est simple à mettre en œuvre dans la plupart des cas.

Cette méthode convertit chaque modalité d'une variable catégorielle en une nouvelle variable binaire, où chaque variable indique la présence ou l'absence de la modalité dans un exemple donné. Ainsi, une variable catégorielle avec n modalités sera transformée en

n variables binaires distinctes. Cette représentation permet aux algorithmes d'apprentissage automatique de mieux interpréter les variables catégorielles en évitant toute notion d'ordre ou de magnitude entre les modalités.

L'inconvénient de cette technique est qu'elle peut entraîner une augmentation importante de la dimensionnalité des données, compliquant ainsi les modèles et augmentant la consommation de mémoire. De plus, elle peut conduire à une dispersion des données et à des problèmes de multicollinéarité lorsque les modalités sont peu fréquentes ou fortement corrélées.

2.3.3 Méthode Label Encoder

Le Label Encoding est l'une des techniques les plus simples pour convertir des variables catégorielles en variables numériques. Elle consiste à attribuer à chaque modalité un numéro arbitraire, généralement compris entre 0 et le nombre total de modalités moins 1.

Cependant, l'utilisation de cette technique peut entraîner des problèmes de priorité dans l'ensemble des modalités de la variable encodée. Les modalités ayant des valeurs numériques plus élevées peuvent être interprétées comme étant plus importantes que celles ayant des valeurs numériques plus basses.

Ainsi, nous avons réalisé des tests comparatifs entre le Label Encoding et d'autres méthodes d'encodage pour déterminer laquelle fonctionne le mieux pour nos modèles.

2.4 Traitement de Déséquilibre de Classes

Le traitement du déséquilibre de classes revêt une importance capitale pour assurer des performances de prédiction fiables dans les problèmes de classification caractérisés par une répartition inégale des classes. Face à cette situation, diverses approches peuvent être mises en œuvre pour pallier ce déséquilibre.[7]

2.4.1 Techniques de rééchantillonnage

- **Sous-échantillonnage** : Cette méthode vise à atténuer le déséquilibre du jeu de données en réduisant la taille de la classe majoritaire.
- **Sous-échantillonnage aléatoire** : Le sous-échantillonnage aléatoire implique la suppression aléatoire d'instances de la classe majoritaire. Cependant, cela peut augmenter la variance du classifieur et entraîner une perte d'informations en éliminant des échantillons utiles.
- **Sur-échantillonnage** : Cette approche consiste à rééquilibrer le jeu de données en augmentant le nombre d'instances de la classe minoritaire.
- **Sur-échantillonnage aléatoire** : Le sur-échantillonnage aléatoire implique l'augmentation de l'ensemble de données en créant des copies aléatoires de certaines instances de la classe minoritaire.
- **SMOTE (Synthetic Minority Over-sampling Technique)** : Le SMOTE est une technique de sur-échantillonnage des données de la classe minoritaire qui consiste à créer des observations synthétiques en se basant sur des segments entre éléments proches de cette classe.

2.5 Augmentation des Données

L'augmentation des données enrichit un ensemble limité ou déséquilibré en générant de nouvelles instances à partir des données existantes. En créant des variations des données d'entraînement, cette technique permet de mieux capturer la complexité des données et de réduire les risques de surapprentissage. Cela accroît la diversité des scénarios d'entraînement et améliore la généralisation du modèle, le rendant plus performant face à des données non vues.

Pour augmenter les données, diverses méthodes peuvent être employées. L'une des approches courantes est l'injection de bruit aléatoire.

2.5.1 Méthode d'Injection de Bruit Aléatoire

Cette méthode ajoute du bruit gaussien, généré à partir d'une distribution normale $\mathcal{N}(\mu, \sigma^2)$, aux caractéristiques d'origine. Le bruit suit une distribution avec une moyenne $\mu = 0$ et un écart-type $\sigma = 1$:

$$\mathcal{N}(\mu, \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x - \mu)^2}{2\sigma^2}\right) \quad (2.1)$$

Pour chaque observation, le bruit ϵ_i est ajouté à chaque caractéristique x_i pour obtenir une valeur perturbée x'_i :

$$x'_i = x_i + \epsilon_i \quad (2.2)$$

Les données d'origine X sont combinées avec les données augmentées X_{bruit} pour former un nouvel ensemble X_{aug} :

$$X_{aug} = [X; X_{bruit1}; X_{bruit2}; \dots; X_{bruitn}] \quad (2.3)$$

Cette approche renforce la robustesse du modèle en le rendant plus résilient aux variations des données d'entrée.

2.6 Les modèles de Machine Learning

Les modèles de classification jouent un rôle crucial dans la modélisation prédictive. Ces modèles permettent d'établir des relations entre les variables explicatives et la variable cible, facilitant ainsi des prévisions précises des résultats binaires ou multiclass. Des approches telles que le Bagging, le Boosting, ainsi que les modèles de régression logistique traditionnels, offrent des solutions flexibles pour analyser des relations complexes et améliorer la précision des prédictions [8].

2.6.1 Régression Logistique

La régression logistique est une méthode statistique fréquemment utilisée pour modéliser des problèmes de classification binaire. Ce modèle permet d'examiner la relation entre un ensemble de variables explicatives, souvent notées X_i , et une variable cible binaire Y . Contrairement à la régression linéaire classique, la régression logistique est adaptée aux situations où la variable cible prend deux valeurs discrètes, telles que 0 ou 1. Le modèle est largement utilisé dans divers domaines, y compris le secteur bancaire, pour évaluer la solvabilité des clients. Il offre une méthode efficace pour résoudre des problèmes de classification.

Le modèle de régression logistique repose sur la fonction logistique, qui transforme une combinaison linéaire des variables explicatives en une probabilité comprise entre 0 et 1. Cette probabilité représente la probabilité qu'un événement se produise, comme la probabilité qu'un client soit en défaut de paiement. La fonction logistique est définie comme suit :

$$\text{logistique}(x) = \frac{1}{1 + e^{-x}} \quad (2.4)$$

où x représente la combinaison linéaire des variables explicatives.

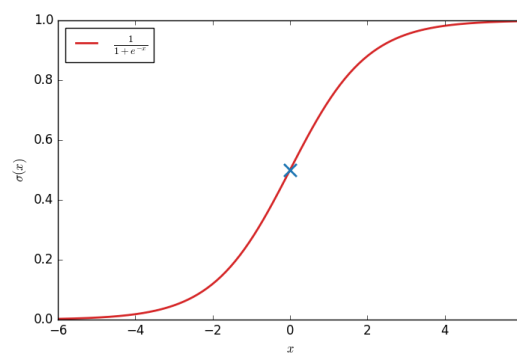


Figure 2.1 – Courbe de la fonction logit (Source : *maximum.fr*)

En général, on pose :

$$Y_i^* = \theta_0 + \sum_{k=0}^n \theta_k X_k + \epsilon_i \quad (2.5)$$

avec ϵ_i terme d'erreur généré par le modèle (variable aléatoire) et Y_i^* une variable aléatoire. La variable cible Y est définie par :

$$Y = \begin{cases} 1 & \text{si } Y_i^* > 0 \\ 0 & \text{sinon} \end{cases} \quad (2.6)$$

Le modèle peut être exprimé sous la forme :

$$\log \left(\frac{\pi(x)}{1 - \pi(x)} \right) = \beta_0 + \sum_{i=1}^p \beta_i x_i \quad (2.7)$$

Dans cette équation, $\pi(x)$ correspond à la probabilité que $Y = 1$ conditionnellement à $X = x$, avec β_0 comme intercept et β_i représentant les coefficients associés aux variables explicatives x_i . La fonction logistique garantit que la sortie du modèle est comprise entre 0 et 1, ce qui est approprié pour modéliser des probabilités.

Un modèle de régression logistique permet d'expliquer linéairement le logit $\left(\log \left(\frac{\pi(x)}{1 - \pi(x)} \right) \right)$ par les variables explicatives X .

Les coefficients β sont estimés en utilisant la méthode du maximum de vraisemblance. La fonction de vraisemblance pour un ensemble d'observations est donnée par :

$$L(\beta) = \prod_{i=1}^n \pi(x_i)^{y_i} \times (1 - \pi(x_i))^{1-y_i} \quad (2.8)$$

où y_i est la valeur observée de la variable cible pour l'observation i . Pour maximiser cette fonction de vraisemblance et obtenir les meilleurs paramètres β , on résout l'équation suivante :

$$\frac{\partial L(\beta)}{\partial \beta} = 0 \quad (2.9)$$

Cependant, la solution analytique de cette équation est souvent complexe.

2.6.2 Bagging

Le Bagging est une méthode d'ensemble visant à diminuer la variance d'un classifieur basé sur les arbres de décision. Cette technique commence par la génération de plusieurs sous-ensembles de données à partir d'un échantillon d'apprentissage, choisi au hasard et avec remise. Chaque sous-ensemble est utilisé pour entraîner un arbre de décision distinct, produisant ainsi un ensemble de modèles variés. En combinant les prédictions de tous ces classifieurs par moyenne, on obtient un modèle plus robuste comparé à un unique classifieur à arbre de décision.

2.6.2.1 Arbre de Décision

L'arbre de décision est un modèle prédictif qui explique une variable d'intérêt à partir de variables indépendantes. Il s'applique à la classification et à la régression. L'arbre est construit en segmentant les données en sous-ensembles homogènes en choisissant les variables les plus discriminantes et en créant des divisions successives jusqu'à atteindre un critère d'arrêt.

L'arbre final se compose de nœuds de décision, qui ont plusieurs branches représentant les valeurs des attributs testés, et de nœuds feuilles, représentant les décisions ou les prédictions. Le nœud racine est le nœud de décision le plus élevé.

Deux critères principaux pour choisir la meilleure séparation d'un nœud sont :

- **L'indice de diversité de Gini** : Mesure la fréquence à laquelle un élément serait incorrectement classé de manière aléatoire. Il est calculé par :

$$I_G(f) = 1 - \sum_{i=1}^m f_i^2 \quad (2.10)$$

où f_i est la fraction des éléments de la classe i .

- **L'entropie** : Évalue le désordre au sein des données et est utilisée pour maximiser le gain d'information. Elle est donnée par :

$$I_E(f) = - \sum_{i=1}^m f_i \log_2(f_i) \quad (2.11)$$

2.6.2.2 Forêt Aléatoire

La forêt aléatoire est une méthode d'apprentissage automatique qui repose sur la construction d'un ensemble d'arbres de décision, chacun étant généré à partir de sous-échantillons de données et de sous-ensembles de variables sélectionnés de manière aléatoire. Cette approche s'appuie sur deux mécanismes clés :

- Le *bagging* consiste à effectuer un tirage aléatoire avec remplacement des observations dans l'ensemble de données, ce qui permet de créer des sous-échantillons distincts pour chaque arbre.
- *Feature sampling* sélectionne de manière aléatoire un sous-ensemble de variables explicatives pour chaque arbre, ce qui contribue à la diversité structurelle des arbres au sein de la forêt.

Cette diversification des arbres réduit la corrélation entre eux. Une fois les arbres construits, leurs prédictions sont combinées pour obtenir une prédiction finale. En régression, cette combinaison se fait par la moyenne des prédictions de tous les arbres. En classification, la prédiction finale est déterminée par la classe la plus fréquente parmi celles proposées par les arbres.

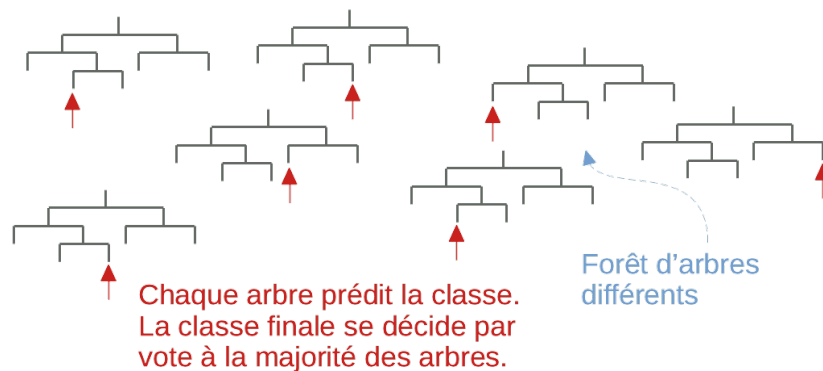


Figure 2.2 – Principe de fonctionnement des forêts aléatoires (Source : sciviews.org)

2.6.3 Boosting

Le boosting est une méthode d'ensemble itérative conçue pour développer une série de prédicteurs. Dans ce processus, les classifieurs sont formés de manière séquentielle. On commence par ajuster un modèle simple aux données, puis on analyse ces données pour identifier les erreurs. Des arbres successifs sont ensuite ajustés, chaque étape visant à améliorer la précision de l'arbre précédent. Lorsqu'une observation est mal classée par un arbre, son poids est augmenté, ce qui incite le modèle suivant à mieux la classer. Ce procédé permet de convertir des classifieurs faibles en un modèle globalement plus performant.

2.6.3.1 XGBoost

XGBoost, ou *eXtreme Gradient Boosting*, est une technique avancée de machine learning conçue pour maximiser la performance des modèles prédictifs, notamment dans les tâches de régression et de classification.

Principes fondamentaux :

- **Boosting de gradient** : Ce mécanisme est au cœur de XGBoost. Il ajuste les modèles en minimisant la fonction de perte, qui mesure l'écart entre les prédictions et les valeurs réelles. À chaque étape, les erreurs des prédictions antérieures sont utilisées

pour affiner les modèles suivants, permettant ainsi d'améliorer la précision des prédictions pour les observations les plus complexes.

- **Pruning**, ou élagage : Cette technique est essentielle dans XGBoost pour optimiser les arbres de décision. Contrairement aux méthodes traditionnelles, qui construisent les arbres jusqu'à une certaine profondeur avant de les tailler, XGBoost applique un élagage anticipé. Cela permet d'ajuster la taille des arbres pendant leur croissance, améliorant ainsi la rapidité d'exécution et réduisant le risque de surapprentissage en contrôlant la complexité des modèles.

2.6.3.2 LightGBM

LightGBM, ou Light Gradient Boosting Machine, est un algorithme de boosting développé par Microsoft, conçu pour optimiser la vitesse et l'efficacité dans les tâches de classification et de régression. Ce modèle repose sur la combinaison d'arbres de décision, où chaque arbre successeur est construit pour corriger les erreurs des arbres précédents, améliorant ainsi progressivement la précision du modèle.

Une innovation majeure de LightGBM est sa méthode de construction des arbres basée sur les histogrammes. Plutôt que de traiter chaque valeur de caractéristique séparément, l'algorithme regroupe les valeurs similaires en bins (intervalles), ce qui simplifie le processus de calcul des meilleurs points de séparation. Cette approche allège la charge computationnelle et accélère le temps d'entraînement.

LightGBM se distingue également par l'utilisation de deux techniques avancées pour améliorer la performance de l'algorithme :

- **GOSS (Gradient-based One-Side Sampling)** : Cette technique privilégie les échantillons avec des gradients élevés, qui apportent plus d'informations utiles pour l'amélioration du modèle, tout en réduisant le nombre d'exemples avec des gradients faibles. Cela concentre les ressources sur les données les plus pertinentes et accélère l'entraînement.

- **EFB (Exclusive Feature Bundling)** : Pour gérer les ensembles de données avec un grand nombre de caractéristiques, LightGBM utilise EFB pour regrouper les caractéristiques mutuellement exclusives. Cette méthode réduit la dimensionnalité des données, simplifiant ainsi l'entraînement tout en préservant l'intégrité des informations.

2.6.3.3 AdaBoost

AdaBoost, ou Adaptive Boosting, est un algorithme de boosting développé par Yoav Freund et Robert Schapire. Ce modèle est conçu pour améliorer la précision des classificateurs en combinant plusieurs modèles faibles en un classificateur fort. AdaBoost est particulièrement efficace pour les tâches de classification et est réputé pour sa capacité à augmenter la performance des modèles simples.

La méthode d'AdaBoost repose sur la combinaison de plusieurs classificateurs faibles, généralement des arbres de décision peu profonds appelés "stumps". À chaque itération, l'algorithme ajuste les poids des observations pour se concentrer davantage sur les exemples mal classifiés par les classificateurs précédents. Cela permet à AdaBoost de créer un modèle global qui corrige progressivement les erreurs des modèles précédents.

AdaBoost utilise deux concepts clés pour améliorer la performance du modèle :

- **Pondération des Observations** : Lors de chaque itération, AdaBoost ajuste les poids des observations en fonction des erreurs des classificateurs précédents. Les observations mal classifiées reçoivent un poids plus élevé, ce qui incite les classificateurs suivants à se concentrer davantage sur ces exemples difficiles.
- **Combinaison des Classificateurs** : Les classificateurs faibles sont combinés en un modèle fort en pondérant leur contribution selon leur précision. Les modèles plus performants ont une influence plus grande sur la décision finale, ce qui renforce la précision globale du modèle.

2.7 Mesures de Qualité de l'Ajustement du Modèle

L'évaluation du modèle est basée sur plusieurs méthodes principalement les mesures de performance et la courbe du ROC et AUC. Ces méthodes sont mieux expliquées dans la partie Annexe A.

2.8 Réglage des hyperparamètres

- **Le GridSearch** Le Grid Search est une méthode systématique pour trouver les meilleurs hyperparamètres d'un modèle d'apprentissage automatique. Il fonctionne en testant toutes les combinaisons possibles de valeurs d'hyperparamètres spécifiées dans une grille prédéfinie. Pour chaque combinaison, le Grid Search utilise une validation croisée pour évaluer les performances du modèle. Cette technique est largement utilisée car elle permet d'explorer efficacement l'espace des hyperparamètres et de trouver ceux qui maximisent les performances du modèle. Donc nous devons utiliser le Grid Search pour explorer et trouver les combinaisons optimales d'hyperparamètres pour nos modèles.
- **Le RandomSearch** Le Random Search est une méthode utilisée pour optimiser les hyperparamètres d'un modèle d'apprentissage automatique en explorant l'espace des hyperparamètres de manière aléatoire. Contrairement au Grid Search, qui teste toutes les combinaisons possibles d'un ensemble prédéfini de valeurs, le Random Search sélectionne des combinaisons d'hyperparamètres de manière aléatoire à partir d'une distribution spécifiée. Cette méthode échantillonne un nombre fixe de combinaisons au lieu d'explorer toutes les possibilités.

Le Random Search est souvent plus efficace que le Grid Search, surtout lorsque l'espace des hyperparamètres est vaste et complexe. En testant des combinaisons aléatoires, cette technique peut découvrir des configurations intéressantes que le Grid Search pourrait manquer, tout en réduisant le coût computationnel. Par consé-

quent, le Random Search est une alternative pratique pour trouver des hyperparamètres optimaux en limitant le nombre d'évaluations nécessaires tout en conservant une bonne chance de trouver des valeurs performantes

2.9 Interprétation des Variables

2.9.1 Méthode SHAP

Dans le domaine de l'apprentissage automatique, il est essentiel de comprendre la contribution des différentes caractéristiques aux prédictions des modèles complexes. Une des méthodes les plus avancées pour l'interprétation des modèles est l'utilisation des valeurs SHAP (SHapley Additive exPlanations).

Les valeurs SHAP s'appuient sur la théorie des jeux coopératifs, et plus spécifiquement sur le concept de valeurs de Shapley. En théorie des jeux, la valeur de Shapley est une solution qui permet de répartir équitablement un gain total entre les contributeurs en fonction de leurs contributions marginales. Transposées à l'apprentissage automatique, les valeurs SHAP fournissent une mesure unifiée pour attribuer la sortie d'un modèle à ses caractéristiques d'entrée.

Fonctionnement des valeurs SHAP

Les valeurs SHAP mesurent la contribution de chaque caractéristique à la prédiction finale d'un modèle. Elles y parviennent en examinant toutes les combinaisons possibles de caractéristiques d'entrée, puis en calculant la contribution marginale de chaque caractéristique dans ces différentes combinaisons. Ainsi, les valeurs SHAP permettent de répartir équitablement la prédiction entre les caractéristiques d'entrée. Les principaux éléments des valeurs SHAP sont :

- **Valeur de référence (Valeur attendue)** : Il s'agit de la prédiction moyenne du modèle si aucune caractéristique n'est connue. Cette valeur représente la sortie attendue en l'absence d'information sur les caractéristiques.

- **Valeur SHAP** : Pour chaque caractéristique, la valeur SHAP indique à quel point cette caractéristique contribue à la différence entre la valeur de référence et la prédiction réelle. Une valeur SHAP positive signifie que la caractéristique augmente la prédiction, tandis qu'une valeur SHAP négative signifie qu'elle la diminue.

Calcul des valeurs SHAP

Le calcul des valeurs SHAP repose sur l'examen de tous les sous-ensembles possibles de caractéristiques et sur la détermination de la contribution marginale de chaque caractéristique. La formule pour la valeur de Shapley ϕ_i d'une caractéristique i est :

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|! \cdot (|N| - |S| - 1)!}{|N|!} \cdot [v(S \cup \{i\}) - v(S)] \quad (2.12)$$

où :

- N représente l'ensemble de toutes les caractéristiques.
- S est un sous-ensemble de N qui ne contient pas la caractéristique i .
- $v(S)$ correspond à la prédiction du modèle en se basant sur les caractéristiques présentes dans le sous-ensemble S .

Cette approche offre une compréhension fine et transparente de l'influence de chaque caractéristique sur les prédictions du modèle, rendant ainsi l'interprétation des modèles d'apprentissage automatique plus accessible et précise.

2.9.2 Méthode LIME

LIME (Local Interpretable Model-agnostic Explanations) est une méthode permettant d'interpréter les modèles d'apprentissage automatique en boîte noire. Contrairement aux approches globales, LIME se concentre sur des explications locales des prédictions individuelles.

Principe de LIME

LIME crée des modèles de substitution locaux pour éclairer les prédictions spécifiques d'un modèle complexe. En introduisant de légères variations dans les données d'entrée, LIME génère un nouvel ensemble de données et entraîne un modèle interprétable pour approximer le comportement du modèle complexe dans le voisinage de l'instance à expliquer. Cette approximation est appelée fidélité locale.

L'explication locale $\xi(x)$ est obtenue par la minimisation suivante :

$$\xi(x) = \arg \min_{g \in G} [L(f, g, \pi_x) + \Omega(g)] \quad (2.13)$$

Avec :

- f : Modèle complexe à expliquer, où $f(x)$ est la probabilité ou l'indicateur binaire pour une classe.
- g : Modèle de substitution approximant f dans le voisinage de x .
- G : Ensemble des modèles interprétables (ex. Lasso, arbres de décision).
- π_x : Voisinage local de x .
- $L(f, g, \pi_x)$: Mesure de l'infidélité de g par rapport à f dans π_x .
- $\Omega(g)$: Mesure de la complexité de g , influençant son interprétabilité.

L'objectif est de maximiser la fidélité locale tout en maintenant une complexité $\Omega(g)$ réduite, assurant ainsi l'interprétabilité.

Processus de LIME

1. **Sélectionner le modèle de ML et un point de référence à expliquer** : Choisissez le modèle d'apprentissage automatique (ML) dont vous souhaitez expliquer les prédictions. Identifiez également une instance spécifique (point de données) pour laquelle vous souhaitez obtenir une explication locale de la prédiction du modèle.
2. **Générer des points de données autour du point de référence** : Créez un ensemble de données autour de l'instance à expliquer en générant de nouvelles observations. Ces observations sont créées en échantillonnant les valeurs des caractéristiques X

autour de l'instance de référence à partir d'une distribution normale, basée sur les caractéristiques de l'ensemble d'entraînement. L'idée est de perturber légèrement chaque caractéristique pour voir comment ces perturbations affectent la prédiction du modèle.

3. **Prédire les résultats pour ces points à l'aide du modèle complexe** : Utilisez le modèle complexe (boîte noire) pour prédire les résultats pour les points générés. Ces prédictions permettent de comprendre comment le modèle complexe réagit aux variations dans les données autour du point de référence.
4. **Attribuer des poids aux points en fonction de leur proximité avec le point de référence** : Évaluez la proximité de chaque point généré par rapport à l'instance de référence. Plus un point est proche de l'instance de référence, plus il aura un poids élevé. Un noyau RBF (Radial Basis Function) est souvent utilisé pour attribuer ces poids, où les points plus proches reçoivent un poids plus élevé et les points plus éloignés reçoivent un poids plus faible.
5. **Entraîner un modèle interprétable sur les points pondérés** : Entraînez un modèle interprétable sur les données générées et pondérées. Le modèle interprétable apprend à prédire les résultats du modèle complexe en fonction des points de données perturbés. Les coefficients β du modèle linéaire obtenu fournissent l'explication de la manière dont chaque caractéristique contribue à la prédiction du modèle complexe pour l'instance de référence.

2.10 Framework Django

Django est un framework de développement web robuste et complet écrit en Python. Il est conçu pour rendre le développement d'applications web aussi rapide et simple que possible tout en encourageant les meilleures pratiques en matière de sécurité et de maintenance [9]. Django suit une architecture basée sur le modèle MVC (Model-View-Controller), un patron de conception qui sépare une application en trois parties inter-

connectées : le **Model** (Modèle) qui gère les données et la logique métier, la **View** (Vue) qui s'occupe de la présentation des données, et le **Controller** (Contrôleur) qui gère les interactions entre le modèle et la vue. Dans le contexte de Django, ce modèle est appelé MVT (Model-View-Template), où le **Model** (Modèle) définit la structure des données, la **View** (Vue) gère la logique métier et la réponse aux requêtes, et le **Template** (Template) s'occupe de la présentation des données dynamiques dans les pages HTML.



Figure 2.3 – Logo Django (Source : [djangoproject.com](https://www.djangoproject.com))

2.10.1 Caractéristiques principales de Django

- **Complet et Structuré** : Django propose un ensemble complet d'outils intégrés, facilitant le développement rapide d'applications web tout en imposant une structure claire pour les projets de grande envergure.
- **Sécurisé** : Il intègre des fonctionnalités de sécurité robustes, incluant la gestion des utilisateurs, la protection contre les attaques CSRF et XSS, ainsi que le cryptage des mots de passe.
- **Extensible** : Son architecture modulaire permet d'ajouter des fonctionnalités via des packages tiers ou des applications personnalisées.
- **Support pour les Templates** : Django utilise un moteur de templates pour générer des pages web dynamiques, assurant une séparation claire entre la logique et la présentation.
- **ORM (Object-Relational Mapping)** : Il inclut un ORM puissant pour interagir avec les bases de données relationnelles, permettant de manipuler les données en Python sans écrire de SQL.

2.10.2 Backend de Django (Python)

Django est largement utilisé comme cadre backend pour le développement d'applications web en Python. Il offre un environnement robuste et structuré pour créer des serveurs web et gérer les requêtes HTTP. Django utilise des vues pour traiter les requêtes des clients, qui peuvent être associées à des modèles pour manipuler les données et des templates pour générer des réponses dynamiques. Il prend également en charge les sessions utilisateur, l'authentification, la gestion des formulaires et l'accès aux bases de données, ce qui en fait un choix populaire pour le développement backend.

2.10.3 Frontend de Django (HTML, CSS, et JavaScript)

Bien que Django soit principalement un cadre backend, il est couramment associé à des technologies frontend telles que HTML, CSS et JavaScript pour offrir une expérience utilisateur complète. Django permet de générer des pages HTML dynamiques via ses templates, qui peuvent être stylisées avec CSS pour une présentation visuelle attrayante. JavaScript peut être utilisé pour ajouter des fonctionnalités interactives et des comportements dynamiques. Django facilite l'intégration de ces technologies frontend en offrant des méthodes pour servir des fichiers statiques et en permettant une communication efficace entre le backend Django et le frontend pour créer des applications web interactives.

2.11 Conclusion

Ce chapitre a été dédié à l'explication théorique des modèles de machine learning utilisés dans ce projet. Nous avons également abordé quelques techniques de sélection des variables et d'encodage des variables catégorielles, ainsi que des méthodes de ré-échantillonnage telles que le suréchantillonnage, SMOTE et le sous-échantillonnage. La prochaine étape consistera à examiner les résultats obtenus suite à l'application de ces techniques.

Chapitre 3

EXPLORATION DES DONNEES

3.1 Introduction

Ce chapitre est dédié à l'exploration des données nécessaires à l'estimation de la LGD et à la modélisation de la PD[10] . Nous commençons par une présentation des bases de données utilisées, suivie de la collecte et du prétraitement des données. Ensuite, nous procédons à des analyses univariées et bivariées pour comprendre les distributions et les relations entre les variables. Enfin, nous appliquons des techniques d'équilibrage et de sélection des variables pour préparer les données à la modélisation, assurant ainsi une base solide pour les étapes d'estimation et de modélisation à venir.[11]

3.2 Perte en Cas de Défaut

3.2.1 Collecte et Présentation de la Base de Données

- **Collecte de Données**

La phase initiale du projet était dédiée à l'exploration et à l'extraction de données à partir de la base de données de la banque Zitouna. Nous avons développé des requêtes SQL pour extraire les variables brutes nécessaires à la calcul de LGDs. Nous avons récupéré les variables concernant les entreprises ayant des comptes

bancaires de type compte professionnel, sans restriction particulière émanant de la banque. Ces entreprises peuvent être soit tunisiennes soit étrangères.

- **Création de la Base de Données**

Nous avons créé à partir des données collectées une nouvelle base de données que nous allons utiliser pour calculer le LGD. Cette étape comprend deux phases essentielles :

-Une première phase consiste à filtrer les lignes non pertinentes qui ne contribuent pas aux données, notamment les clients dont le solde impayé et les encours par financement sont nuls, surtout qu'elles sont liées au calcul de l'EAD.

-Une deuxième phase consiste à créer des nouvelles colonnes à partir des variables brutes de la base de données comme :

- Colonne "Date d'entrée" : représente La date à laquelle le client ne peut plus respecter ses obligations de paiement selon les termes du contrat.
- Colonne "Date de sortie" : La date à laquelle le client a résolu son état de défaut et retour en sain.
- Colonne "EAD" (Exposure at Default) : représente la valeur du EAD, valeur du montant exposée au risque de défaut.
- Colonnes "Récupération" : représente le montant récupéré chaque année en utilisant la formule préalablement mentionnée.

- **Présentation de la Base de Données**

La base de données contient des informations sur les clients professionnels de la Banque Zitouna. Elle comprend 2 016 observations et 25 variables. Elle renferme des informations sur les financements ainsi que sur les situations financières et comportementales des clients. La période d'observation couvre 10 ans, de 2013 à 2023.

3.2.2 Hypothèses et Règles de Gestion

Le tableau, présenté dans l'Annexe B, résume les hypothèses et les règles de gestion significatives qui ont été utilisées dans le cadre de notre analyse. Ces hypothèses et règles sont cruciales pour la compréhension et l'interprétation des résultats obtenus. Elles définissent les critères et les méthodes appliquées pour identifier et gérer les défauts, calculer les engagements, et traiter les flux financiers associés.

3.2.3 Calcul des LGD pour chaque financement

Après la préparation de notre base de données, nous passons maintenant à la phase cruciale du calcul du LGD (Loss Given Default) pour chaque client. En suivant les formules théoriques mentionnées précédemment, nous appliquons les remarques et règles spécifiques établies pour garantir la précision et la pertinence de nos calculs.

3.2.4 Analyse Des Données

- **Distribution du LGD**

La figure 3.1 montre la distribution du LGD WORKOUT . Cette variable, que nous avons calculée.

Observations de la Distribution :

-Asymétrie :

La distribution de la LGD WORKOUT est asymétrique vers la droite. Cela signifie que la plupart des observations ont des valeurs de LGD relativement basses. -

Longue queue :

La présence d'une longue queue à droite indique qu'il y a des cas rares avec LGD élevé.

-Densité :

La densité élevée près de zéro suggère que beaucoup de clients ont une faible valeur de LGD ou un LGD nul.

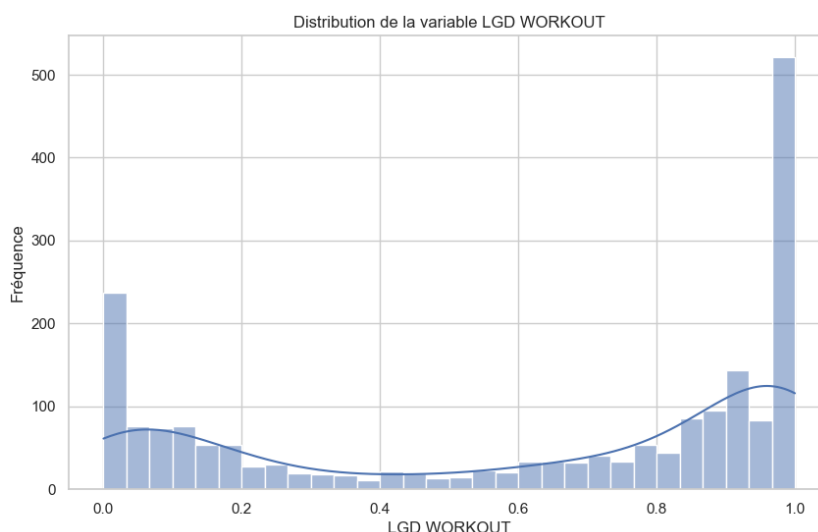


Figure 3.1 – *Distribution du LGD*

- **Catégories de financement**

La figure 3.2 montre le nombre de clients pour chaque catégorie de financement. Les catégories représentées sont "Ijara mobilier", "Financement à la gestion", "CMT d'investissement", "Ijara immobilier", "CMT d'acquisition de matériel de transport" et "CMT financement des équipements professionnels".

Observations :

-Ijara mobilier :

C'est la catégorie de financement la plus fréquente avec plus de 1300 occurrences. Cela pourrait indiquer que la majorité des financements dans le dataset sont destinés à l'Ijara mobilier.

- Financement à la gestion et CMT d'investissement :

Ce sont les deux catégories les plus fréquentes après l'Ijara mobilier, avec environ 750 occurrences. Les financements à la gestion et les investissements représentent une part significative de la base.

- CMT financement des équipements professionnels :

La catégorie la moins fréquente avec environ 10 occurrences, indiquant que cette

catégorie de financement est la moins courante dans notre base de données.

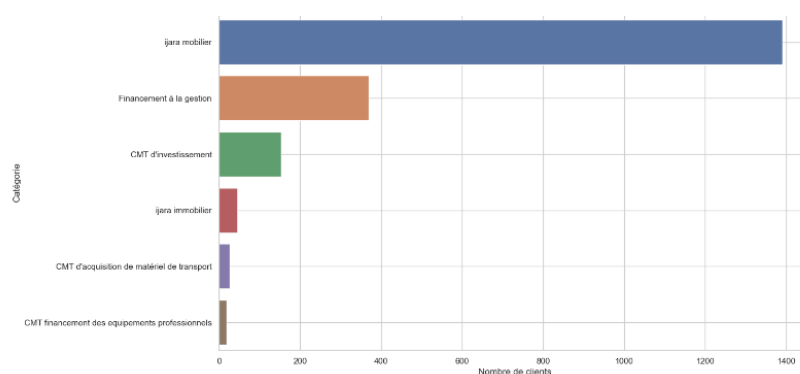


Figure 3.2 – Répartition des catégories

- **Maturité de financement**

La figure 3.3 illustre la répartition du nombre de clients en fonction de la maturité de leur engagement.

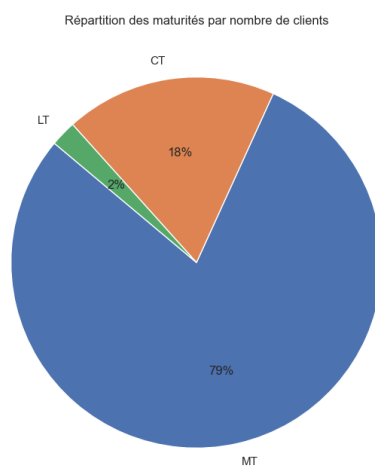


Figure 3.3 – Répartition de maturité

Observations :

-La majorité des clients, 79%, sont engagés sur le moyen terme (MT), ce qui en fait la

catégorie la plus populaire.

- Les engagements à court terme (CT) regroupent environ 18% des clients, montrant un intérêt modéré pour cette durée d'engagement.
- Les engagements à long terme (LT) sont les moins fréquents, avec un pourcentage de 2%.

Cette distribution indique une préférence marquée des clients pour des engagements de durée moyenne, tandis que les engagements à long terme restent relativement rares.

3.3 Probabilité de Défaut

3.3.1 Présentation de la Base de Données

La base de données sur laquelle repose cette étude est un échantillon constitué de données provenant de 140 entreprises emprunteuses. Les informations ont été collectées à partir des dossiers de financements de ces entreprises, permettant ainsi d'évaluer leur performance en matière de crédit. Ces données ont été extraites des bilans comptables des entreprises clientes de la banque Zitouna pour une période spécifique. Elles contiennent des informations détaillées sur 48 ratios financiers, ainsi que 4 données supplémentaires de nature qualitative et quantitative, qui seront définies comme suit :

3.3.1.1 Les Ratios Financiers

Les ratios financiers sélectionnés pour cette analyse fournissent des indications précises sur divers aspects de la santé financière des entreprises, y compris leur solvabilité, leur rentabilité, leur structure financière et leur trésorerie. Ces ratios sont utilisés pour évaluer la performance des entreprises, particulièrement en vue d'octroi de crédits, de rachat d'entreprises ou pour comparer les performances avec d'autres entreprises du même secteur. Dans la suite nous allons expliquer les ratios que nous considérons les plus pertinents et vous trouverez le reste dans l' Annexe C.

- **Capitaux propres / Total passif et cap prop** : Ce ratio mesure la proportion des capitaux propres par rapport au total des passifs et aux capitaux propres, reflétant la capacité de l'entreprise à financer ses activités sans recourir à des financements externes. Un ratio élevé indique une autonomie financière accrue et une réduction des risques liés à l'endettement.
- **ANC / Total Actif** : Ce ratio indique la part des actifs non courants dans le total des actifs, montrant ainsi l'importance des investissements à long terme de l'entreprise, tels que les immobilisations corporelles et incorporelles. Ce ratio est crucial pour la stabilité et la croissance future de l'entreprise.
- **AC / Total Actif** : Ce ratio représente la proportion des actifs courants par rapport au total des actifs, soulignant la liquidité de l'entreprise et sa capacité à couvrir ses obligations à court terme. Un pourcentage plus élevé suggère une gestion efficace des ressources disponibles pour les opérations quotidiennes.
- **Stocks / Total Actif** : Ce ratio évalue le pourcentage des stocks par rapport au total des actifs, indiquant la part des biens destinés à la vente ou à la production. Il reflète l'efficacité de la gestion des inventaires dans le cycle d'exploitation de l'entreprise.
- **PNC / Total Passifs** : Ce ratio indique la proportion des passifs non courants dans le total des passifs, illustrant la part des dettes à long terme de l'entreprise et sa capacité à supporter des engagements financiers sur une période prolongée.
- **PC / Total Passifs** : Ce ratio représente la part des passifs courants dans le total des passifs, révélant l'importance des obligations à court terme dans la structure financière de l'entreprise et sa capacité à honorer ses dettes à court terme.
- **Résultat de l'exercice / Capital Social** : Ce ratio exprime la rentabilité du capital social de l'entreprise en comparant le résultat de l'exercice avec le capital apporté par les actionnaires. Il offre un aperçu du rendement généré par les investissements des propriétaires.

- **BFR en j / CAHT** : Ce ratio indique le besoin en fonds de roulement en jours de chiffre d'affaires hors taxes, permettant d'évaluer la durée pendant laquelle l'entreprise doit financer son cycle d'exploitation avant de récupérer ses liquidités.
- **Stocks en j / CAHT** : Ce ratio exprime le stock en jours de chiffre d'affaires hors taxes, permettant d'évaluer l'efficacité de la gestion des stocks dans le cycle d'exploitation.
- **Fournisseurs en j / Achats TTC** : Ce ratio évalue les dettes fournisseurs en jours d'achats toutes taxes comprises, indiquant la durée moyenne de paiement des fournisseurs, un indicateur clé de la gestion des obligations à court terme.
- **Chg Pers / VA** : Ce ratio exprime les charges de personnel par rapport à la valeur ajoutée, indiquant l'efficacité des coûts de main-d'œuvre dans la génération de valeur économique.
- **EBE / CA** : Ce ratio compare l'excédent brut d'exploitation au chiffre d'affaires, offrant un aperçu de la rentabilité opérationnelle par rapport aux ventes totales.
- **R.Ex / Capitaux Propres (ROE)** : Ce ratio mesure le rendement des capitaux propres en comparant le résultat de l'exercice aux capitaux propres, un indicateur essentiel pour les investisseurs.
- **R.Ex / Total Actifs (ROA)** : Ce ratio évalue le rendement des actifs en comparant le résultat de l'exercice au total des actifs, offrant un aperçu de l'efficacité de l'utilisation des actifs.

3.3.1.2 Les Autres Variables

Dans cette partie allons présenter les autres variables que nous considérons les plus pertinents et vous trouverez le reste dans l'Annexe avec les autres ratios.

- **Total Engagement** : Ce variable représente le montant total des engagements financiers d'une entreprise, incluant tous les crédits, prêts et autres obligations contrac-

tées qui doivent être remboursés à l'avenir. C'est un indicateur de l'exposition globale de l'entreprise aux dettes et obligations financières.

- **Total Impayé** : Ce variable représente le montant total des créances qui restent impayées par les clients ou débiteurs à une date donnée. Elle reflète le niveau de retard dans les paiements et peut influencer la liquidité et la solvabilité de l'entreprise.
- **Variation VA** : Ce variable presente la variation de la valeur ajoutée d'une période à l'autre, fournissant un aperçu des changements dans la contribution nette à l'économie.
- **CA Total** : Ce variable représente le chiffre d'affaires total généré par l'entreprise, un indicateur global de la performance commerciale sur une période donnée.
- **Résultat de l'Exercice (R.Ex)** : Ce variable presente le résultat net de l'exercice, indiquant le bénéfice ou la perte totale après toutes les charges et produits.
- **Trésorerie nette fin exercice (EENE)** : Ce variable presente la trésorerie nette en incluant les éléments économiques et non économiques, offrant une vision plus complète de la situation financière à court terme de l'entreprise.
- **Besoin en Fonds de Roulement (EENE)** : Ce variable presnte le besoin en fonds de roulement en tenant compte des éléments économiques et non économiques, offrant une vue plus complète des besoins financiers de l'entreprise.
- **Secteur PC** : Ce variable qualitatie indique le secteur d'activité de l'entreprise souvent désigné comme le "secteur de portefeuille client" .
- **Indicateur de Défaut** : Cette variable représente un indicateur binaire qui indique la présence de défaut de paiement par un client ou une entreprise.

3.3.2 Analyse de Données

3.3.2.1 Analyse Univariée des Variables

L'analyse univariée des variables permet d'évaluer leur distribution, leur fréquence et le pourcentage des modalités au sein de chaque variable. Dans cette section, nous présentons la répartition de variable cible ainsi que des variables explicatives

■ Variable cible

L'histogramme ci-dessous illustre la distribution des clients qui sont soit sains, soit à défaut :

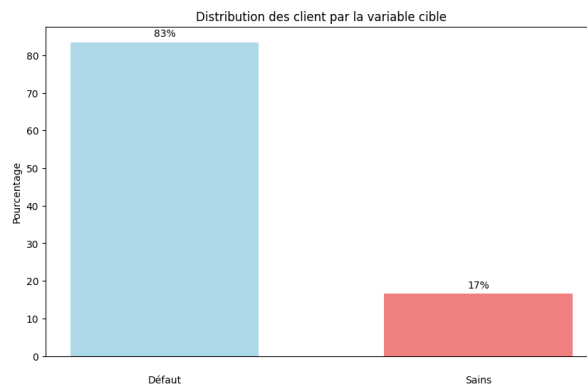


Figure 3.4 – *Distribution de la variable cible*

D'après la figure 3.4, on observe un déséquilibre notable entre les deux classes [défaut (1) et sain (0)].

En effet, 83% des entreprises n'ont pas fait défaut, tandis que seulement 17% ont fait défaut. Ce déséquilibre doit être pris en compte, car si la classe minoritaire est négligée, les modèles construits risquent de ne pas être capables de détecter efficacement les clients en défaut.

■ Variables explicatives

Dans cette partie, nous avons présenté que les variables les plus pertinentes pour

nos modèles, vu que notre base de données possède 53 variables explicatives.

- Variable **"Secteur PC"**

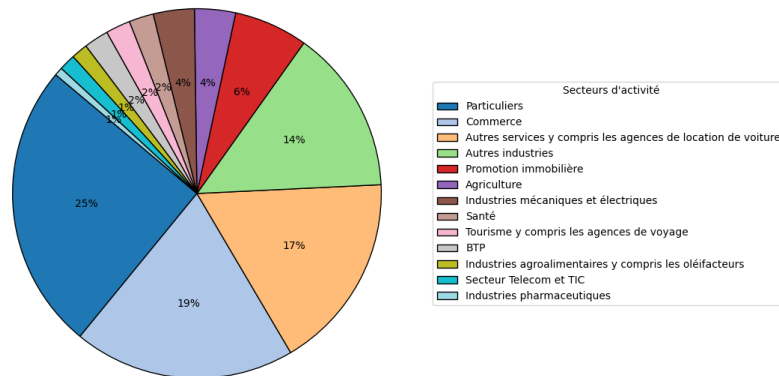


Figure 3.5 – Répartition des clients entreprises selon leurs secteurs d'activité

La figure ci-dessous montre que 25% des clients sont issus du secteur des particuliers, suivi du commerce avec 19% et des services divers à 17%. Les autres industries représentent 14% des clients, tandis que la promotion immobilière en compte 6%. Les secteurs restants, tels que l'agriculture, l'industrie mécanique, la santé, et le tourisme, constituent chacun environ 4% des clients, avec des secteurs comme la construction, l'agroalimentaire, les télécommunications, et la pharmacie occupant une part encore plus réduite.

- Variable **"Total Engagement"**

La figure 3.6 représente la distribution des clients en fonction de leur total d'engagement. On peut observer que la majorité des clients, soit environ 75%, ont un total d'engagement compris entre 0 et 1M. Le reste des clients se répartit équitablement dans les autres classes d'engagement, avec chaque classe représentant environ 5% à 10% des clients.

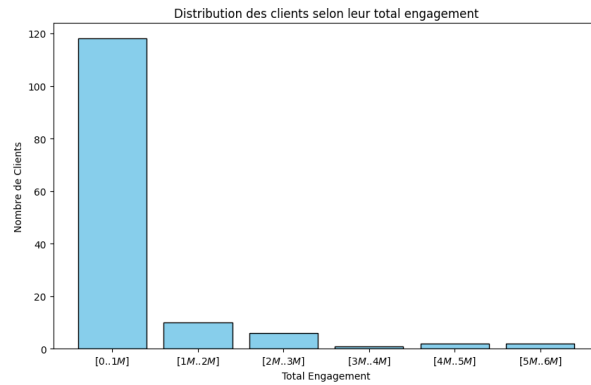


Figure 3.6 – *Distribution des clients selon leur total d'engagement*

3.3.2.2 Analyse Bivariée des Variables

- **Etude de la corrélation entre les variables explicatives**

L'analyse bivariée constitue une méthode élémentaire d'analyse quantitative permettant d'examiner la relation empirique entre deux variables. Elle est utile pour tester des hypothèses d'association entre ces variables. À cet égard, nous commençons par utiliser la matrice de corrélation de Spearman pour explorer les relations entre les variables quantitatives.

Matrice de corrélation : Spearman



Figure 3.7 – *Matrice de corrélation : Spearman*

Il est clair que toutes les variables quantitatives du modèle sont indépendantes, avec des coefficients de corrélation variant de -0,41 à 0,35

- **Etude de la Corrélation entre la variable cible et les variables explicatives**

Pour évaluer la dépendance entre les variables explicatives et la variable cible, nous avons utilisé le test du chi carré pour les variables qualitatives et le test ANOVA pour les variables quantitatives.

Test de chi-deux

L'hypothèse de base du test de chi-deux est H_0 : absence de relation entre les deux variables catégorielles. L'hypothèse alternative H_1 : Il y a une dépendance significative entre les deux variables.

Si $p_value < 0.05$ on rejette H_0 et on accepte l'hypothèse H_1 de dépendance des deux variables.

Variables qualitatives	Statistique du test	P-value
Secteur PC	13.87	0.309

Table 3.1 – *Corrélation entre les variables qualitatives et la variable cible*

La variable SECTEUR PC a une p-value supérieure au seuil de 5%, ce qui indique qu'elle est indépendante de la variable cible

Test d'ANOVA

L'hypothèse de base du test d'ANOVA est H_0 : égalité des moyennes au sein des deux groupes.

L'hypothèse alternative H_1 : il y a une différence significative entre la moyenne des deux groupes.

Si $p_value < 0.05$ on rejette H_0 et on accepte l'hypothèse H_1 .

Variable quantitative	Statistique du test	P-value
Résultat de l'exercice (R.Ex)	8.83	0.004
CA total	4.69	0.032
Trésorerie nette fin exercice TN (EENE)	4.51	0.035
Besoin en fonds de roulement BFR (EENE)	7.35	0.0075
Total Impayé	21.70	0.000

Table 3.2 – *Corrélation entre les variables quantitatives et la variable cible*

Le tableau 3.3 ci-dessus présente les résultats du test :

On constate que toutes les variables ont des valeurs inférieures ou égales à 0,05, ce qui indique qu'elles sont significativement liées à la variable cible.

3.3.3 Traitement des Variables

Avant d'implémenter un modèle statistique, il est crucial de prétraiter les variables de la base de données pour gérer les valeurs manquantes, les incohérences, et les variables catégorielles complexes. Cette étude analyse ces défis en détail.

3.3.3.1 Changement des Types de Variables

Parfois le type d'une variable ne reflète pas son type réel. Il est donc essentiel de modifier son type de manière interactive, par exemple en convertissant des variables numériques enregistrées en objets en types float ou entier, avant de poursuivre l'analyse.

3.3.3.2 Traitement des Valeurs Manquantes

Dans l'analyse statistique, les valeurs manquantes ne peuvent être ignorées car la plupart des modèles de machine learning nécessitent des données complètes pour

fonctionner efficacement. La suppression des observations incomplètes peut entraîner une perte importante d'informations, surtout avec des ensembles de données limités. Ainsi, nous privilégions l'imputation des valeurs manquantes en utilisant diverses techniques adaptées à la proportion et au type des données manquantes pour préserver la qualité des données.

Le graphique ci-dessous montre le pourcentage de valeurs manquantes pour les variables concernées :

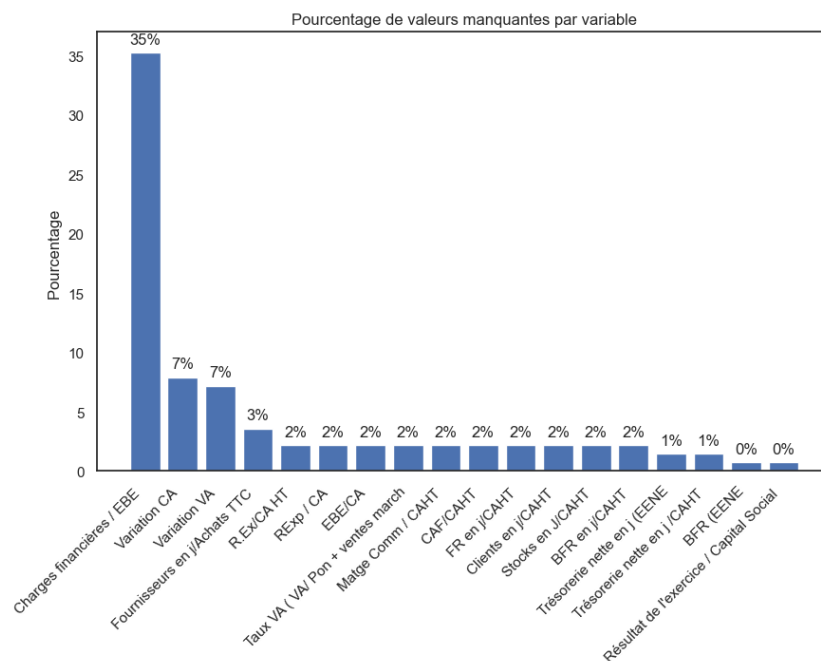


Figure 3.8 – *Pourcentage des données manquantes*

Avant de sélectionner la méthode d'imputation, il est nécessaire de déterminer le pourcentage maximum de valeurs manquantes acceptable.

En effet, les variables dont le pourcentage de valeurs manquantes dépasse ce seuil sont supprimées. Dans le traitement réalisé, un seuil de 10 % a été fixé, ce qui entraîne la suppression automatique de la variable "**Charges financières / EBE**".

La Figure 3.10 met en évidence le pourcentage de données manquantes pour les

variables concernées . Toutes les autres variables ne présentent pas de valeurs manquantes.

Ensuite, nous avons imputé les variables restantes en utilisant la méthode d'imputation basée sur une valeur fixe, telle que la moyenne, la médiane, le maximum ou le minimum. Étant donné que toutes les variables sont numériques et que le nombre de valeurs manquantes est relativement faible, cette approche est adaptée pour compléter les données de manière appropriée.

3.3.3.3 Regroupement des Variables Catégorielles

Afin de préparer ces données pour l'analyse, nous avons appliqué deux techniques de codage : **one-hot encoding** et **label encoding** pour la variable qualitative "**Secteur PC**". Le **one-hot encoding** crée 11 nouvelles colonnes binaires correspondant à chaque modalité : 'Particuliers', 'Agriculture', 'Autres services y compris les agences de location de voitures', 'Promotion immobilière', 'Commerce', 'Santé', 'Autres industries', 'Tourisme y compris les agences de voyage', 'Industries agroalimentaires y compris les oléifacteurs', 'Industries mécaniques et électriques', 'BTP', 'Industries pharmaceutiques', et 'Secteur Telecom et TIC'. Chaque colonne indique la présence ou l'absence de la catégorie correspondante. En revanche, le **label encoding** attribue un nombre unique à chaque modalité, simplifiant le traitement des données pour les algorithmes de machine learning .

3.3.4 Équilibrage des Données

Pour améliorer la performance de notre modèle et traiter le déséquilibre des classes, nous avons appliqué plusieurs techniques d'équilibrage des données. Les méthodes utilisées incluent le suréchantillonnage, le sous-échantillonnage, et le SMOTE. Les résultats sont introduits dans le tableau ci-dessous :

Méthode	Nombre d'observations
SMOTE	152
Suréchantillonnage	232
Sous-échantillonnage	46

Table 3.3 – *Résultats des méthodes d'équilibrage des données*

3.3.5 Augmentation des Données

En raison de la taille limitée et du déséquilibre de notre base de données, nous avons choisi d'utiliser la méthode d'injection de bruit aléatoire pour l'augmentation des données. Cette technique consiste à ajouter du bruit gaussien aux données d'origine afin de générer des variations supplémentaires. Grâce à cette approche, la taille de notre base de données a été étendue de 152 à 912 observations, ce qui permet de mieux équilibrer les données et d'améliorer la performance du modèle.

3.3.6 Sélection des Variables

Dans cette section, nous employons deux techniques de sélection de caractéristiques. Tout d'abord, nous utilisons la méthode **SelectKBest** pour évaluer toutes les caractéristiques en fonction de l'information décrite dans la section 2.2. Ensuite, nous utilisons la **Sélection de Caractéristiques Séquentielle (SFS) Stepwise**, une méthode itérative qui ajoute ou supprime des caractéristiques une à une en fonction des performances du modèle, optimisant les ensembles de caractéristiques séparément pour les deux approches.

3.3.6.1 Sélection des Variables par SelectKbest

En utilisant la méthode SelectKBest, 12 variables ont été sélectionnées, comme le montre le tableau ci-dessous.

Variables Sélectionnées	
AC/ total Actif	Stocks / total actif
ANC/total Actif	Résultat de l'exercice / Capital Social
Total Engagement	Total Impayé
Besoin en fonds de roulement BFR (EENE)	Résultat d'exploitation (RExp)
BFR en j/CAHT	Stocks en J/CAHT
CA total	Trésorerie nette fin exercice TN (EENE)

Table 3.4 – Liste des variables sélectionnées selon SelectKbest

3.3.6.2 Sélection des Variables par la régression STEPWISE

Pour identifier les facteurs les plus importants influençant notre analyse, nous avons utilisé une méthode de sélection de caractéristiques séquentielle . Ce processus a permis de sélectionner 14 variables critiques, présentées dans le Tableau ci-dessous :

Variables Sélectionnées	
Capitaux propres/total passif et cap prop	ANC/total Actif
Trésorerie nette en j/CAHT	R.Ex/total Actifs(ROA)
Fournisseurs en j/Achats TTC	BFR en j / CAHT
Chg Pers/VA	EBE/CA
Stocks/total actif	RExp / CA
R.Ex/CA HT	Total Engagement
Total Impayé	Secteur PC

Table 3.5 – Liste des variables sélectionnées selon stepwise

3.4 Conclusion

Dans ce chapitre, nous avons décrit les différentes étapes du prétraitement de notre base de données, incluant le traitement des valeurs manquantes et le regroupement des modalités pour certaines variables présentant un nombre élevé de catégories, ainsi que l'augmentation des données. Nous avons également présenté divers graphiques illustrant notre jeu de données et exposé deux méthodes de sélection de variables que nous utiliserons dans le quatrième chapitre dédié à la modélisation.

Chapitre 4

EVALUATION ET IMPLEMENTATION DES MODELES

4.1 Introduction

Dans ce chapitre, nous nous concentrons sur l'évaluation et l'implémentation des modèles prédictifs développés pour estimer la Probabilité de Défaut (PD) [12] et évaluer les résultats de la Perte en Cas de Défaut (LGD). Après avoir préparé et analysé les données dans les chapitres précédents[11], nous mettons en œuvre divers algorithmes de machine learning, incluant la régression logistique, les arbres de décision, les forêts aléatoires, XGBoost, LightGBM et AdaBoost. Cette phase implique la construction, l'optimisation et la validation rigoureuse des modèles pour déterminer les plus efficaces. Enfin, nous développons une application web intuitive pour permettre aux décideurs de la banque d'utiliser ces modèles de manière pratique et interactive.

4.2 Perte en Cas de Défaut

4.2.1 Résultat du Calcul

Les résultats du calcul du LGD WORKOUT à partir des autres variables sont résumés dans la figure 3.1 suivante. Cette figure présente les statistiques descriptives essentielles de la variable LGD WORKOUT, qui est une variable numérique continue avec des valeurs comprises entre 0 et 1.

On observe que La valeur moyenne de LGD WORKOUT est d'environ 0.606. Cela signifie que, en moyenne, Les clients font face à une perte de 60.6 % sur leur financement.

```
count    2002.000000
mean      0.605591
std       0.389290
min       0.000000
25%       0.160065
50%       0.783312
75%       0.983277
max       1.000000
Name: LGD WORKOUT, dtype: float64
```

Figure 4.1 – *Statistiques descriptives de la LGD*

4.2.2 Dashboard

Pour présenter notre travail d'une manière interactive, nous avons créé un tableau de bord en utilisant Microsoft Power BI. Ce tableau de bord contient une analyse exploratoire de nos données, illustrant plusieurs statistiques descriptives pertinentes. Nous montrons, à titre d'exemple, la distribution du LGD (Loss Given Default) ainsi que la répartition des clients par catégorie de financement et par maturité. Le tableau de bord inclut également la moyenne du LGD par catégorie, par maturité,

et selon la combinaison des deux, offrant une vue détaillée des pertes potentielles en cas de défaut.

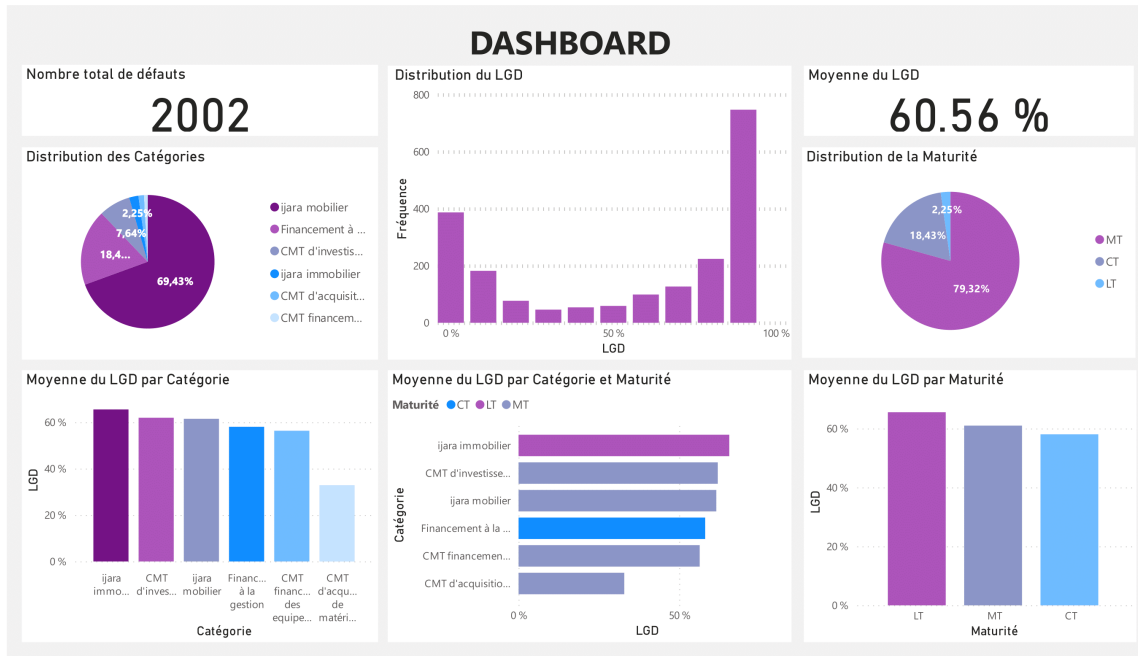


Figure 4.2 – Interface du Dashboard

Notre dashboard a été conçu de manière dynamique en utilisant des filtres qui permettent de sélectionner et de visualiser différentes informations en fonction d'un indicateur choisi. Par exemple, nous pouvons choisir une catégorie ou une maturité spécifique et observer comment les autres caractéristiques se distribuent pour cette référence, ce qui facilite une exploration approfondie et ciblée des données.

4.3 Probabilité de Défaut

4.3.1 Travail Préliminaire

La base de données initiale a été améliorée par divers traitements, et avant la modélisation, elle a été divisée en deux parties : 70% pour l'apprentissage (**Train**),

où la simulation a été appliquée, et 30% pour la validation du modèle (**Test**), où les modèles ont été évalués.

4.3.2 Résultats Empiriques

Dans cette section, nous présenterons les différents résultats obtenus à l'aide des modèles suivants : Régression logistique, Classification naïve bayésienne, Arbre de décision, Forêt aléatoire, XGBoost, LightGBM et AdaBoost.

Pour chaque modèle, nous avons comparé plusieurs techniques de ré-échantillonnage de la variable cible, à savoir le sous-échantillonnage aléatoire, le sur-échantillonnage aléatoire et SMOTE. Nous avons utilisé également les deux types d'encodage des variables qualitatives à savoir le OneHotEncoder et LabelEncoder .

Nous mettrons en avant leurs mesures de performance, en insistant sur leur pouvoir discriminant et leur capacité à estimer la probabilité de défaut.

Nous avons choisi les mesures de performance les plus pertinentes pour notre contexte de modélisation. Tout d'abord, l'accuracy, présentée dans le chapitre 2, est couramment utilisée comme indicateur de performance. Cependant, étant donné le déséquilibre des classes de notre variable cible, l'accuracy ne constitue plus un indicateur fiable pour valider nos modèles. En effet, cela pourrait conduire au paradoxe de l'accuracy [13], où l'on peut obtenir une précision élevée malgré des classifications erronées importantes. De plus, l'accuracy ne permet pas de différencier les erreurs de classification en fonction des classes. Par conséquent, nous avons décidé d'exclure cet indicateur au profit d'autres mesures de performance.

Ensuite, la précision et le rappel fournissant ainsi une indication sur la classification des clients en défaut. Il est également crucial pour une banque de prédire correctement les clients sains (Classe 0), car une mauvaise classification de ces derniers pourrait entraîner leur perte en refusant leur demande de crédit sous le prétexte erroné qu'ils sont en défaut. C'est pourquoi la précision et le rappel sont les deux

mesures les plus importantes dans notre sélection de modèles.

Afin d'optimiser les performances des modèles, nous avons réalisé une recherche des hyperparamètres à l'aide de GridSearch détaillé dans l'Annexe D.

4.3.2.1 Régression Logistique

Cette partie est consacrée pour présenter les résultats de régression logistique.

- **Encodage par OneHotEncoder**

La méthode d'équilibrage	Rappel	Précision	F1-Score	AUC
Sous-échantillonnage (Undersampling)	0.63	0.83	0.69	0.64
Sur-échantillonnage (Oversampling)	0.85	0.87	0.86	0.77
SMOTE	0.83	0.87	0.83	0.74

Table 4.1 – Résultats des métriques pour la régression logistique avec OneHotEncoder

Pour un encodage des variables par la méthode **OneHotEncoder**, le modèle de régression logistique qui donne les meilleures métriques est celui dont la méthode d'équilibrage est le **sur-échantillonnage (Oversampling)**.

- **Encodage par LabelEncoder**

La méthode d'équilibrage	Rappel	Précision	F1-Score	AUC
Sous-échantillonnage (Undersampling)	0.63	0.83	0.69	0.64
Sur-échantillonnage (Oversampling)	0.85	0.88	0.86	0.77
SMOTE	0.83	0.87	0.84	0.75

Table 4.2 – Résultats des métriques pour la régression logistique avec LabelEncoder

Pour un encodage des variables par la méthode **LabelEncoder**, le modèle de régression logistique qui donne les meilleures métriques est celui dont la méthode

d'équilibrage est le **sur-échantillonnage (Oversampling)**.

Conclusion : Le modèle de régression logistique à retenir est celui qui combine les techniques **LabelEncoder** et **sur-échantillonnage (Oversampling)**.

4.3.2.2 Arbre de Décision

Le tableau 4.3 ci-dessous présente les valeurs optimisées des hyperparamètres :

Hyperparamètre	criterion	max_depth	min_samples_split	min_samples_leaf	max_features
Valeur	gini	None	2	1	auto

Table 4.3 – Hyperparamètres pour Arbre de Décision

Dans la partie suivante, nous allons expliciter les résultats du modèle Arbre de décision en utilisant les trois techniques de rééchantillonnage et les deux techniques d'encodage mentionnées précédemment :

- Encodage par **OneHotEncoder**

La méthode d'équilibrage	Rappel	Précision	F1-Score	AUC
Sous-échantillonnage (Undersampling)	0.65	0.84	0.71	0.65
Sur-échantillonnage (Oversampling)	0.80	0.84	0.82	0.67
SMOTE	0.76	0.86	0.79	0.73

Table 4.4 – Résultats des métriques pour l'Arbre de décision avec *OneHotEncoder*

Pour un encodage des variables par la méthode **OneHotEncoder**, le modèle d'Arbre de décision qui donne les meilleures métriques est celui dont la méthode d'équilibrage est le **sur-échantillonnage (Oversampling)**.

- Encodage par **LabelEncoder**

La méthode d'équilibrage	Rappel	Précision	F1-Score	AUC
Sous-échantillonnage (Undersampling)	0.80	0.87	0.82	0.72
Sur-échantillonnage (Oversampling)	0.78	0.84	0.80	0.66
SMOTE	0.84	0.88	0.86	0.83

Table 4.5 – Résultats des métriques pour l'Arbre de décision avec **LabelEncoder**

Pour un encodage des variables par la méthode **LabelEncoder**, le modèle d'Arbre de décision qui donne les meilleures métriques est celui dont la méthode d'équilibrage est **SMOTE**.

Conclusion : Le modèle d'Arbre de décision à retenir est celui qui combine les techniques **LabelEncoder** et **SMOTE**.

4.3.2.3 Forêt Aléatoire

Le tableau 4.6 ci-dessous présente les valeurs optimisées des hyperparamètres :

Hyperparamètre	n_estimators	max_depth	min_samples_split	min_samples_leaf	bootstrap
Valeur	50	None	2	1	False

Table 4.6 – Hyperparamètres pour Forêt Aléatoire

Cette partie se concentrera sur l'interprétation des résultats du modèle Forêt aléatoire, en utilisant les trois méthodes de rééchantillonnage et les deux techniques d'encodage présentées précédemment :

- Encodage par **OneHotEncoder**

La méthode d'équilibrage	Rappel	Précision	F1-Score	AUC
Sous-échantillonnage (Undersampling)	0.61	0.80	0.67	0.56
Sur-échantillonnage (Oversampling)	0.85	0.75	0.80	0.50
SMOTE	0.85	0.84	0.84	0.63

Table 4.7 – Résultats des métriques pour la Forêt Aléatoire avec OneHotEncoder

Pour un encodage des variables par la méthode **OneHotEncoder**, le modèle de Forêt Aléatoire qui donne les meilleures métriques est celui dont la méthode d'équilibrage est **SMOTE**.

- Encodage par LabelEncoder

La méthode d'équilibrage	Rappel	Précision	F1-Score	AUC
Sous-échantillonnage (Undersampling)	0.52	0.78	0.60	0.51
Sur-échantillonnage (Oversampling)	0.85	0.81	0.83	0.55
SMOTE	0.85	0.86	0.85	0.70

Table 4.8 – Résultats des métriques pour la Forêt Aléatoire avec LabelEncoder

Pour un encodage des variables par la méthode **LabelEncoder**, le modèle de Forêt Aléatoire qui donne les meilleures métriques est également celui dont la méthode d'équilibrage est **SMOTE**.

Conclusion : Le modèle de Forêt Aléatoire à retenir est celui qui combine les techniques **LabelEncoder** et **SMOTE**.

4.3.2.4 XGBoost

Le tableau 4.9 ci-dessous présente les valeurs optimisées des hyperparamètres :

Cette partie est dédiée à l'analyse des résultats obtenus avec le modèle XGBoost,

Hyperparamètre	n_estimators	max_depth	learning_rate	gamma	delta_step
Valeur	30	6	0.2	0	0

Table 4.9 – Hyperparamètres pour XGBoost

en appliquant les trois méthodes de rééchantillonnage et les deux techniques d'encodage précédemment mentionnées :

- Encodage par **OneHotEncoder**

La méthode d'équilibrage	Rappel	Précision	F1-Score	AUC
Sous-échantillonnage (Undersampling)	0.70	0.87	0.74	0.75
Sur-échantillonnage (Oversampling)	0.80	0.84	0.82	0.68
SMOTE	0.83	0.83	0.81	0.76

Table 4.10 – Résultats des métriques pour XGBoost avec *OneHotEncoder*

Pour un encodage des variables par la méthode **OneHotEncoder**, le modèle XGBoost qui donne les meilleures métriques est celui dont la méthode d'équilibrage est **SMOTE**.

- Encodage par **LabelEncoder**

La méthode d'équilibrage	Rappel	Précision	F1-Score	AUC
Sous-échantillonnage (Undersampling)	0.67	0.81	0.72	0.59
Sur-échantillonnage (Oversampling)	0.80	0.84	0.82	0.67
SMOTE	0.85	0.90	0.87	0.85

Table 4.11 – Résultats des métriques pour XGBoost avec *LabelEncoder*

Pour un encodage des variables par la méthode **LabelEncoder**, le modèle XGBoost

qui donne les meilleures métriques est également celui dont la méthode d'équilibrage est **SMOTE**.

Conclusion : Le modèle XGBoost à retenir est celui qui combine les techniques **LabelEncoder** et **SMOTE**.

4.3.2.5 LightGBM

Le tableau 4.12 ci-dessous présente les valeurs optimisées des hyperparamètres :

Hyperparamètre	num_leaves	max_depth	learning_rate	n_estimators	min_child_samples
Valeur	31	6	0.1	30	1

Table 4.12 – Hyperparamètres pour *LightGBM*

Dans cette section, nous examinerons les résultats du modèle LightGBM, en tenant compte des trois techniques de rééchantillonnage et des deux méthodes d'encodage discutées précédemment :

- Encodage par **OneHotEncoder**

La méthode d'équilibrage	Rappel	Précision	F1-Score	AUC
Sous-échantillonnage (Undersampling)	0.59	0.86	0.65	0.70
Sur-échantillonnage (Oversampling)	0.80	0.82	0.81	0.61
SMOTE	0.80	0.87	0.83	0.74

Table 4.13 – Résultats des métriques pour *LightGBM* avec *OneHotEncoder*

Pour un encodage des variables par la méthode **OneHotEncoder**, le modèle LightGBM qui donne les meilleures métriques est celui dont la méthode d'équilibrage est **SMOTE**.

- Encodage par **LabelEncoder**

La méthode d'équilibrage	Rappel	Précision	F1-Score	AUC
Sous-échantillonnage (Undersampling)	0.57	0.81	0.64	0.60
Sur-échantillonnage (Oversampling)	0.72	0.79	0.75	0.55
SMOTE	0.78	0.86	0.81	0.73

Table 4.14 – Résultats des métriques pour *LightGBM* avec *LabelEncoder*

Pour un encodage des variables par la méthode **LabelEncoder**, le modèle *LightGBM* qui donne les meilleures métriques est celui dont la méthode d'équilibrage est **SMOTE**.

Conclusion : Le modèle *LightGBM* à retenir est celui qui combine les techniques **OneHotEncoder** et **SMOTE**.

4.3.2.6 AdaBoost

Le tableau 4.15 ci-dessous présente les valeurs optimisées des hyperparamètres :

Hyperparamètre	n_estimators	learning_rate
Valeur	200	1

Table 4.15 – Hyperparamètres pour *AdaBoost*

Nous allons maintenant détailler les performances du modèle *AdaBoost* en intégrant les trois approches de rééchantillonnage ainsi que les deux techniques d'encodage :

- Encodage par **OneHotEncoder**

Pour un encodage des variables par la méthode **OneHotEncoder**, le modèle *AdaBoost* qui donne les meilleures métriques est celui dont la méthode d'équilibrage est **SMOTE**.

La méthode d'équilibrage	Rappel	Précision	F1-Score	AUC
Sous-échantillonnage (Undersampling)	0.74	0.88	0.78	0.78
Sur-échantillonnage (Oversampling)	0.81	0.86	0.84	0.67
SMOTE	0.83	0.85	0.84	0.69

Table 4.16 – Résultats des métriques pour AdaBoost avec OneHotEncoder

- Encodage par LabelEncoder

La méthode d'équilibrage	Rappel	Précision	F1-Score	AUC
Sous-échantillonnage (Undersampling)	0.59	0.82	0.65	0.63
Sur-échantillonnage (Oversampling)	0.83	0.85	0.84	0.68
SMOTE	0.83	0.87	0.84	0.75

Table 4.17 – Résultats des métriques pour AdaBoost avec LabelEncoder

Pour un encodage des variables par la méthode **LabelEncoder**, le modèle AdaBoost qui donne les meilleures métriques est aussi celui dont la méthode d'équilibrage est **SMOTE**.

Conclusion : Le modèle AdaBoost à retenir est celui qui combine les techniques **LabelEncoder** et **SMOTE**.

4.3.3 Comparaison des Modèles Appliqués

Nous avons testé plusieurs algorithmes de modélisation, notamment la régression logistique, la classification naïve bayésienne, l'arbre de décision, la forêt aléatoire, XGBoost, LightGBM et AdaBoost. Dans la section précédente, nous avons sélectionné la base rééquilibrée et les techniques d'encodage les plus appropriées pour chaque modèle[14]. Nous allons maintenant comparer ces modèles afin d'identifier

celui offrant les meilleures performances. Cette comparaison sera réalisée en examinant les métriques de performance, comme le montre le tableau ci-dessous, qui illustre les performances des différents algorithmes d'apprentissage en fonction des techniques de rééchantillonnage et d'encodage les plus adéquates.

Modèle	Technique de rééchantillonnage	Type d'encodage	Rappel	Précision	F1-Score	AUC
Régression logistique	Sur-échantillonnage	LabelEncoder	0.85	0.88	0.86	0.77
Classification naïve bayésienne	Sous-échantillonnage	OneHotEncoder	0.78	0.89	0.81	0.80
Arbre de décision	SMOTE	LabelEncoder	0.84	0.88	0.86	0.83
Forêt aléatoire	SMOTE	LabelEncoder	0.85	0.86	0.85	0.70
XGBoost	SMOTE	LabelEncoder	0.85	0.90	0.87	0.85
LightGBM	SMOTE	OneHotEncoder	0.80	0.87	0.83	0.74
AdaBoost	SMOTE	LabelEncoder	0.83	0.87	0.84	0.75

Table 4.18 – *Mesures de performance des différents modèles*

Le rééchantillonnage de la base d'apprentissage, ainsi que l'encodage et le paramétrage des algorithmes, ont permis d'améliorer les différentes mesures de performance. Parmi les modèles évalués, Le modèle de régression logistique utilisant le sur-échantillonnage avec LabelEncoder présente de bonnes performances, avec un rappel de 0.85 et un F1-Score de 0.86. Cependant, il est surpassé par l'Arbre de Décision utilisant SMOTE avec LabelEncoder, qui atteint un rappel de 0.84 et un F1-Score équivalent de 0.86, mais avec une AUC plus élevée de 0.83. Cela suggère une meilleure capacité de détection des cas positifs par rapport à la régression logistique.

La Forêt Aléatoire avec SMOTE et LabelEncoder affiche également des résultats solides, notamment avec un rappel de 0.85 et un F1-Score de 0.85. Cependant, son AUC est plus faible à 0.70 par rapport à l'Arbre de Décision, indiquant une discrimination légèrement inférieure.

Le modèle XGBoost, utilisant SMOTE avec LabelEncoder, se distingue comme le meilleur en termes de précision et d'AUC, avec une précision de 0.90 et une AUC

de 0.85. Ces valeurs montrent une excellente capacité à prédire correctement les classes positives et à discriminer entre les classes.

Le modèle LightGBM avec SMOTE et OneHotEncoder présente un rappel de 0.80 et une AUC de 0.74, ce qui est inférieur à ceux des autres modèles comme XGBoost et Arbre de Décision. AdaBoost avec SMOTE et LabelEncoder montre un rappel de 0.83 et une AUC de 0.75, mais ne surpasse pas XGBoost en termes de précision et d'AUC.

En conclusion, le modèle XGBoost avec SMOTE et LabelEncoder offre les meilleures performances globales en termes de précision et d'AUC, le rendant ainsi le modèle le plus performant parmi ceux évalués.

— Courbe de ROC

La figure 4.3 présente la courbe de Roc du modèle le plus performant XGBoost. Nous constatons que la courbe ROC est proche de la courbe idéale à savoir la courbe qui passe par le point (0,1), cette courbe nous renseigne ainsi sur la pertinence de ce modèle choisi.

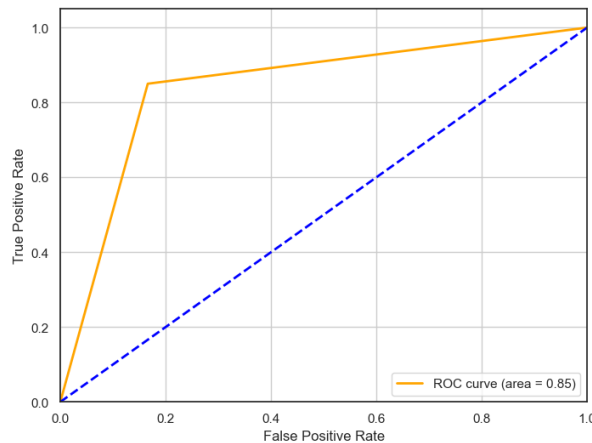


Figure 4.3 – Courbe de ROC pour le modèle XGBoost

— Importance des variables

Les variables explicatives les plus importantes selon le modèle le plus perfor-

mant, XGBoost, sont illustrées dans la figure suivante :

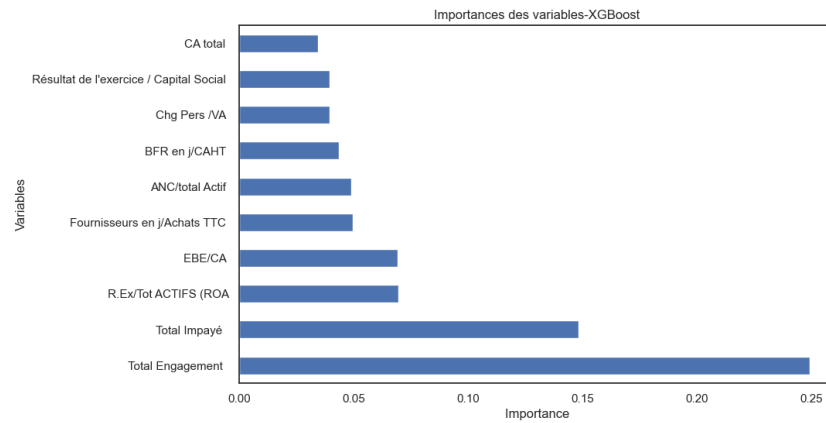


Figure 4.4 – Importance des variables dans le modèle XGBoost

Il est clair d'après le graphique que la variable **Total Engagement** est la plus importante, avec une contribution qui dépasse 20% dans la précision des prédictions du modèle. Nous notons également que **Total Impayé** est une variable cruciale avec un impact substantiel. Parmi ces variables, **R.Ex/Tot ACTIFS (ROA)** et **EBE/CA** se distinguent également, soulignant leur importance dans le modèle. Les variables **Fournisseurs en j/Achats TTC**, **ANC/total Actif**, **BFR en j/CAHT**, ainsi que **CA total** et **Résultat de l'exercice/Capital Social** apportent une valeur prédictive supplémentaire, mais avec une influence moindre comparée à **Total Engagement** et **Total Impayé**.

4.3.4 Interprétabilité des Modèles

4.3.4.1 SHAP

Le graphique SHAP présenté dans la figure 4.5 ci-dessous illustre l'influence des différentes variables sur les prédictions du modèle de probabilité de défaut, où la

variable cible est 0 (clients sains) ou 1 (clients en défaut).

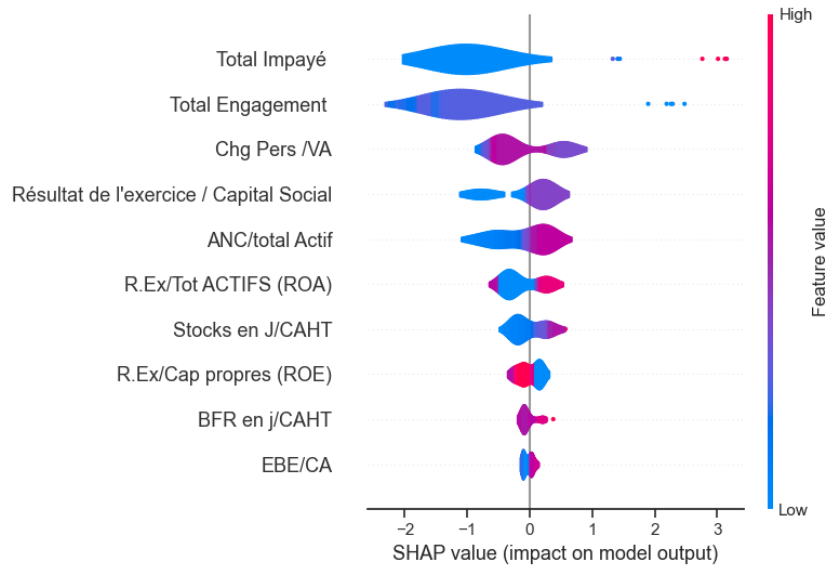


Figure 4.5 – *Impact des variables sur le modèle selon SHAP*

Total Impayé est la variable la plus influente sur les prédictions. Les valeurs élevées de cette variable augmentent considérablement la probabilité que le client soit en défaut (indiqué par 1). Cela signifie que plus le montant impayé est important, plus le modèle prédit que le client a un risque élevé de défaut.

De même, **Total Engagement** est également une variable significative. Les entreprises avec un engagement total plus élevé sont plus susceptibles d'être en défaut, ce qui reflète un risque accru avec des niveaux d'engagement plus élevés.

Chg Pers /VA et **Résultat de l'exercice / Capital Social** montrent un impact notable sur les prédictions, bien que moins important que les deux premières variables. Des charges du personnel élevées par rapport à la valeur ajoutée ou un faible résultat d'exercice par rapport au capital social augmentent la probabilité que le client soit en défaut.

ANC/total Actif a également un certain impact. Cela suggère que la proportion

d'actifs immobilisés comparée au total des actifs a un effet sur la probabilité de défaut, bien que cet effet soit plus modéré. **R.Ex/Tot ACTIFS (ROA)** et **Stocks en J/CAHT** influencent également la probabilité de défaut, bien que de manière moins marquée.

R.Ex/Cap propres (ROE) , **BFR en j/CAHT** , et **EBE/CA** ont un impact moindre mais néanmoins présent sur la probabilité de défaut. Ces ratios financiers, bien que moins influents, contribuent tout de même à la détermination du risque de défaut.

4.3.4.2 LIME

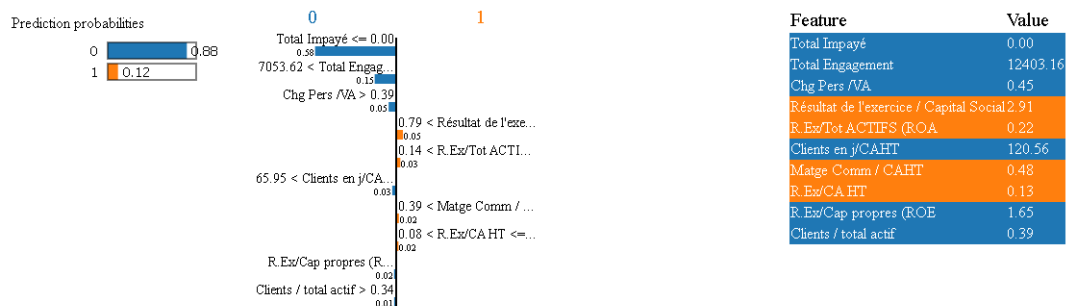


Figure 4.6 – Explication LIME des prédictions du modèle

Le graphique LIME (Local Interpretable Model-agnostic Explanations) présenté dans la Figure 4.6 ci-dessus montre comment les différentes caractéristiques influencent les prédictions du modèle pour un client spécifique, avec la variable cible définissant les clients en défaut (1) et les clients sains (0).

Dans ce cas, Le modèle prédit que ce client a une probabilité de 88% d'être classé comme sain , et une probabilité de 12 % d'être classé comme en défaut .

Les caractéristiques qui favorisent la probabilité que ce client soit sain incluent, en premier lieu, l'absence de montant impayé (Total Impayé 0.00), ce qui constitue le facteur le plus déterminant pour la santé financière de l'entreprise. Ensuite, un engagement total plus faible (7053.62) réduit également le risque de défaut,

suggérant une exposition modérée aux obligations financières. De plus, des charges du personnel relativement faibles par rapport à la valeur ajoutée ($\text{Chg Pers} / \text{VA} = 0.39$) sont indicatives d'une gestion efficace des coûts opérationnels, renforçant la stabilité financière. La gestion des créances clients, mesurée en jours par rapport au chiffre d'affaires ($\text{Clients en j} / \text{CAHT} = 65.95$), joue également un rôle clé en maintenant la liquidité de l'entreprise. Enfin, une rentabilité des capitaux propres modérée ($\text{R.Ex} / \text{Cap propres (ROE)} = 1.65$) indique une performance équilibrée sans prise de risque excessive.

À l'opposé, certaines caractéristiques augmentent la probabilité que le client soit en défaut. Un résultat d'exercice élevé par rapport au capital social ($\text{Résultat de l'exercice} / \text{Capital Social} = 2.91$) peut suggérer une prise de risque accrue, ce qui, bien que potentiellement profitable à court terme, pourrait mener à des difficultés financières à long terme. Une faible rentabilité des actifs ($\text{R.Ex} / \text{Tot ACTIFS (ROA)} = 0.22$) peut indiquer une inefficacité dans l'utilisation des ressources de l'entreprise, augmentant le risque de défaut. Une marge commerciale modérée ($\text{Matge Comm} / \text{CAHT} = 0.48$) pourrait également être un signe de faiblesse financière (Alerte de solvabilité). De plus, une faible rentabilité du chiffre d'affaires hors taxes ($\text{R.Ex} / \text{CA HT} = 0.13$) et une proportion élevée de créances clients par rapport aux actifs totaux ($\text{Clients} / \text{total actif} = 0.39$) sont des indicateurs supplémentaires de risque potentiel.

En conclusion, dans cette analyse LIME, on observe que les caractéristiques qui réduisent la probabilité de défaut (favorisant la classe 0) sont principalement liées à une bonne gestion financière et à des niveaux d'engagement modérés. À l'inverse, des ratios financiers suggérant une performance sous-optimale ou un haut niveau de risque augmentent la probabilité de défaut (favorisant la classe 1).

4.4 Application Web

Pour agréger toutes les étapes de notre travail, une application web a été réalisée à l'aide du framework Django de Python. Django inclut un ensemble d'outils pour gérer la base de données, l'authentification des utilisateurs, et la gestion des templates.

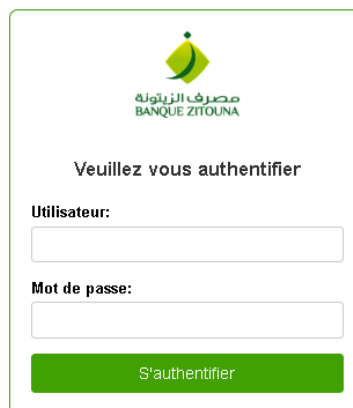
Pour la partie Frontend, nous avons utilisé HTML, CSS, et JavaScript afin de rendre les pages web plus attrayantes et interactives.

L'application se compose principalement de quatre pages :

- Interface d'authentification
- Page d'accueil
- Page de prédiction de la probabilité de défaut (PD) et son résultat
- Page d'estimation de la perte en cas de défaut (LGD) et son résultat

Dans la partie suivante, nous allons détailler les éléments de l'interface graphique.

4.4.1 Interface d'Authentification



مصرف الزيتونة
BANQUE ZITOUNA

Veuillez vous authentifier

Utilisateur:

Mot de passe:

S'authentifier

Figure 4.7 – Page d'authentification

Le client peut se connecter en tant qu'administrateur ou utilisateur standard en utilisant un identifiant unique et un mot de passe en cliquant sur "S'authentifier".

L'administrateur peut gérer les utilisateurs, ajouter ou supprimer des comptes, et exiger des mises à jour des mots de passe.

4.4.2 Page d'Accueil

Une fois connecté, la page d'accueil s'affiche avec un design épuré et une palette de couleurs vertes et blanches, accompagnée d'un logo qui renforce l'identité visuelle de la banque. Deux boutons principaux, " **Prédire PD** " et " **Estimer LGD** ", ont bien distincts et facilement accessibles. Le bouton " **Prédire PD** " permet d'estimer la probabilité de défaut à l'aide du modèle sélectionné, tandis que " **Estimer LGD** " offre une évaluation de la perte en cas de défaut en utilisant la méthode **LGD Workout**. Cette page permet aux clients de choisir entre ces deux fonctions.

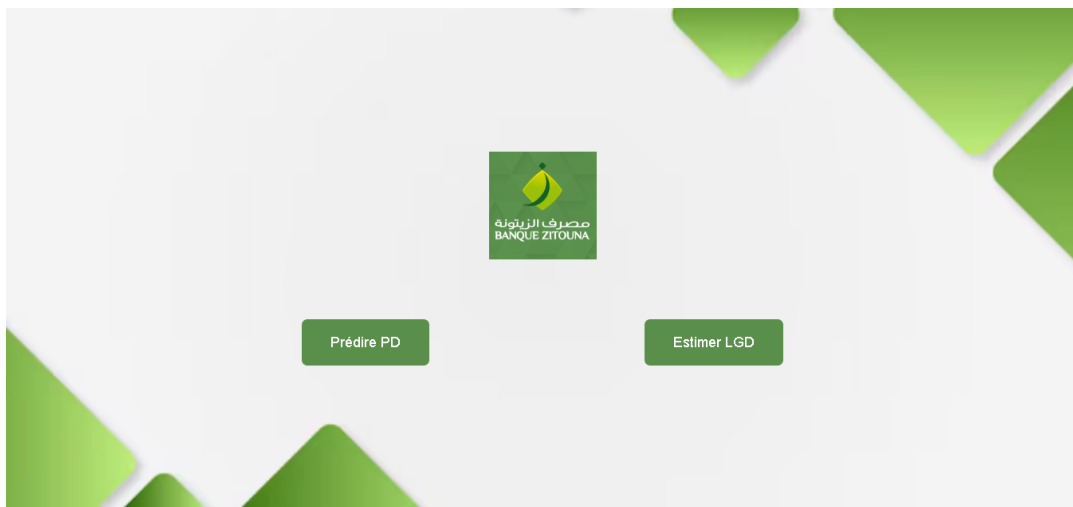


Figure 4.8 – Page d'accueil

4.4.3 Prédiction de la Probabilité de Défaut(PD)

4.4.3.1 Page de Prédiction

Cette page comprend des champs à remplir correspondant aux variables sélectionnées pour notre modèle. Ces champs serviront d'entrées pour le modèle. Elle comporte également un bouton " **Prédire** ". Une fois les informations nécessaires soumises, l'API prédit dynamiquement la valeur cible, comme le montre la Figure 4.9, et fournit une prédiction de la probabilité de défaut.

Entrez les informations nécessaires	
ANC/total Actif :	AC/ total Actif :
<input type="text"/>	<input type="text"/>
Total Engagement :	Chg Pers IVA :
<input type="text"/>	<input type="text"/>
Variation CA :	Stocks / total actif :
<input type="text"/>	<input type="text"/>
Stocks en J/CAHT :	Résultat de l'exercice / Capital Social :
<input type="text"/>	<input type="text"/>
R.Ex/Cap propres ROE :	Fournisseurs en j/Achats TTC :
<input type="text"/>	<input type="text"/>
Total Impayé :	
<input type="text"/>	
<input type="button" value="Prédire"/>	

Figure 4.9 – Page de prédiction PD

4.4.3.2 Page de Résultat

Cette page affiche le résultat de la prédiction de la probabilité de défaut (PD) basée sur notre modèle et les variables d'entrée. Elle permet de déterminer si le client est considéré comme solvable ou en défaut en fonction de la probabilité estimée, avec un bouton " **Prédire à nouveau** " est disponible, permettant à l'utilisateur de réaliser une nouvelle prédiction s'il le souhaite, comme le montre la Figure 4.10 ci-dessous.

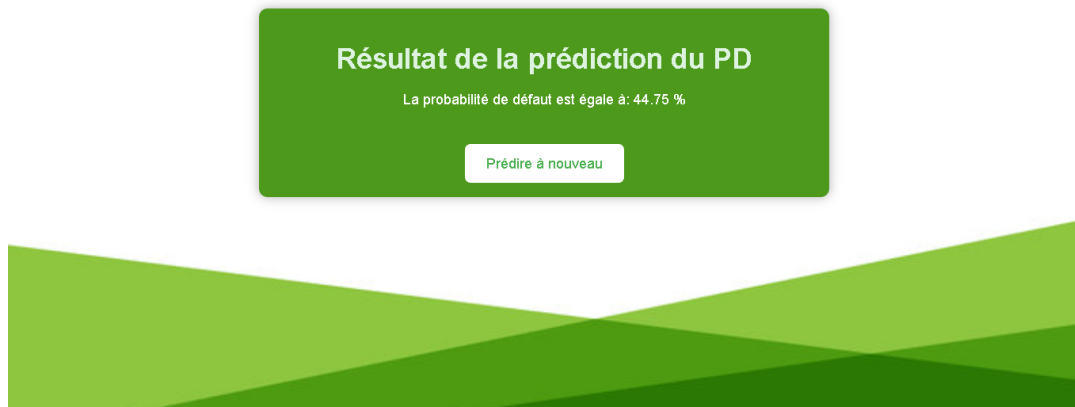


Figure 4.10 – *Page de résultat de prédiction PD*

4.4.4 Estimation de la Perte en Cas de Défaut(LGD)

4.4.4.1 Page d'Estimation

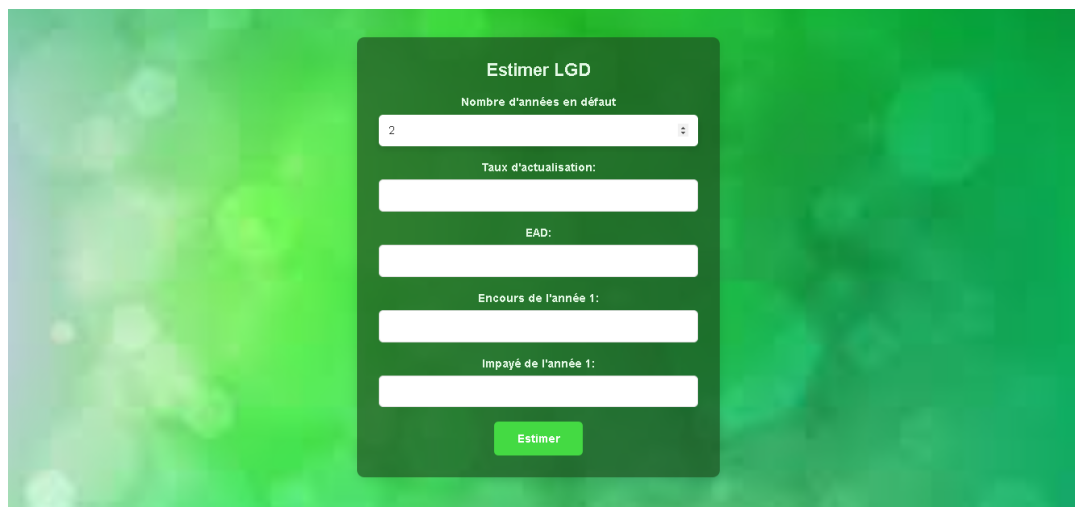


Figure 4.11 – *Page d'estimation LGD*

La page de calcul de la LGD permet aux utilisateurs de saisir des données pour obtenir une estimation de la LGD comme le montre la Figure 4.11, les utilisateurs saisissent des paramètres tels que le nombre d'années en défaut, l'Exposition en cas

de défaut (EAD), le taux d'actualisation, l'encours et l'impayé de l'année. Après avoir cliqué sur ” **Estimer** ”, l'API estime la LGD à l'aide de la méthode LGD Workout.

4.4.4.2 Page de Résultat

Cette page présente le résultat de l'estimation de la perte en cas de défaut (LGD), calculée à l'aide de la méthode LGD Workout et des données d'entrée avec possibilité de revenir à la page d'accueil en utilisant le bouton ”**Retourner à l'accueil**”, comme le montre la Figure 4.12.



Figure 4.12 – Page de résultat d'estimation LGD

4.5 Conclusion

Dans ce chapitre, nous avons exploré et évalué plusieurs modèles de machine learning pour prédire la Probabilité de Défaut (PD) et exploré le résultat d'estimation de la Perte en Cas de Défaut (LGD) par un tableau de bord. Après avoir comparé les performances des différents modèles, nous avons identifié le modèle

XGBoost comme étant le plus performant pour la prédiction de la PD. En ce qui concerne la LGD, nous avons utilisé la méthode LGD Workout pour fournir des estimations précises. De plus, nous avons développé une application web intuitive permettant aux utilisateurs de calculer et de visualiser ces métriques de manière interactive.

Conclusion Générale

Au cours de ce projet de fin d'études effectué au sein de la Banque Zitouna, nous avons cherché à mettre en place une solution pour prédire la Probabilité de Défaut (PD) des clients et estimer la Perte en Cas de Défaut (LGD). Nous avons suivi une démarche rigoureuse de prétraitement et de nettoyage des données, ainsi que de sélection des variables pertinentes pour la phase de modélisation. Nous avons également appliqué plusieurs techniques d'équilibrage de la variable cible, la classe des clients en défaut étant minoritaire. Deux techniques d'encodage des variables qualitatives ont été testées pour optimiser les modèles.

Pour la Probabilité de Défaut (PD), nous avons développé et validé plusieurs modèles prédictifs en combinant ces différentes techniques. Après une comparaison approfondie, le modèle XGBoost encodé avec la technique LabelEncoder et suréchantillonné avec SMOTE s'est révélé être le plus performant. Ce modèle a offert des mesures de performance pertinentes, notamment un rappel de 0.85, une précision de 0.90, un F1-score de 0.87 et une AUC de 0.85.

Pour l'estimation de la Perte en Cas de Défaut (LGD), nous avons appliqué la méthode LGD Workout, qui repose sur le calcul de la valeur actuelle nette des flux de trésorerie de récupération. Cette méthode a fourni des estimations fiables des pertes potentielles en cas de défaut de paiement.

Pour l'avenir, il serait bénéfique d'explorer l'intégration de nouvelles données et d'améliorer continuellement les modèles développés, tout en formant les utilisateurs à l'utilisation de l'application web et à l'interprétation des résultats.

Annexe A

Mesures de Qualité de l'Ajustement du Modèle

— **Mesure de Performance**

La matrice de confusion permet de définir plusieurs métriques clés pour évaluer la performance d'un modèle :

— **Accuracy (Précision Globale)**

L'accuracy mesure le pourcentage de bonnes prédictions sur l'ensemble des observations :

$$\text{Accuracy} = \frac{\text{Vrais Positifs} + \text{Vrais Négatifs}}{\text{Vrais Positifs} + \text{Vrais Négatifs} + \text{Faux Positifs} + \text{Faux Négatifs}} \quad (.1)$$

Cependant, elle peut être trompeuse en cas de déséquilibre des classes.

— **Rappel (Sensibilité)**

Le rappel, ou taux de vrais positifs, mesure la proportion des instances positives correctement identifiées :

$$\text{Rappel} = \frac{\text{Vrais Positifs}}{\text{Vrais Positifs} + \text{Faux Négatifs}} \quad (.2)$$

— **Précision**

La précision évalue la proportion de prédictions positives correctes :

$$\text{Précision} = \frac{\text{Vrais Positifs}}{\text{Vrais Positifs} + \text{Faux Positifs}} \quad (.3)$$

Ces deux mesures, bien que cruciales, ne suffisent pas à elles seules pour évaluer un modèle. Des métriques combinées comme le F1-score sont souvent utilisées pour une évaluation plus complète.

— **F1-Score**

Le F1-score est la moyenne harmonique de la précision et du rappel :

$$\text{F1-Score} = \frac{2 \times \text{Précision} \times \text{Rappel}}{\text{Précision} + \text{Rappel}} \quad (.4)$$

— **Spécificité**

La spécificité mesure la proportion de négatifs correctement identifiés :

$$\text{Spécificité} = \frac{\text{Vrais Négatifs}}{\text{Vrais Négatifs} + \text{Faux Positifs}} \quad (.5)$$

— **Taux de Faux Positifs (FPR)**

Le FPR indique la proportion de négatifs incorrectement classifiés comme positifs :

$$\text{FPR} = \frac{\text{Faux Positifs}}{\text{Vrais Négatifs} + \text{Faux Positifs}} \quad (.6)$$

— **Courbe de ROC et l'AUC**

Une courbe ROC (Receiver Operating Characteristic) est un graphique qui évalue les performances d'un modèle de classification à travers tous les seuils de décision possibles. Elle illustre comment le taux de vrais positifs (sensibilité) varie en fonction du taux de faux positifs (1 - spécificité). Autrement dit, la courbe montre la capacité du modèle à distinguer entre les classes positives et négatives à différents niveaux de seuil.

En complément, l'efficacité de la courbe ROC est souvent mesurée par l'aire sous la courbe, désignée par AUC (Area Under the Curve). L'AUC quantifie la performance globale du modèle : une valeur élevée de l'AUC indique que le modèle possède une meilleure capacité discriminatoire, c'est-à-dire qu'il est plus efficace pour séparer les classes positives des classes négatives.

Annexe B

Hypothèses et Règles de Gestion

Date d'entrée en défaut	<p>La date où le client a été classé en classe 2 (Classe SNI comportementale basé sur les impayés).</p> <p>Les défauts déjà en force à la date du 31/12/2013 seront exclus de la base de calcul de la LGD.</p>
Date de clôture du défaut	<p>Le compte est radié : Le 31/12 de l'année de cession.</p> <p>Retour en sain : Date de retour à une classe inférieure à la Classe 2.</p>
Défauts liés	Deux défauts sont considérés comme liés si la période qui les sépare est inférieure ou égale à 9 mois(un ans dans notre cas).
EAD au moment du défaut	EAD = Engagement lors de l'entrée en défaut
Flux financiers après le défaut	<p>Les flux financiers sont calculés sur la base de la variation d'engagement.</p> <p>Un delta d'engagement positif sera traité comme étant un flux de récupération.</p>
Traitement des retours en sain	Flux de récupération artificiel à la suite du retour en sain

Table .19 – *Synthèse des Hypothèses et Règles de Gestion*

Annexe C

Les Variables

- **Fonds de Roulement** : Mesure la liquidité disponible pour financer les activités courantes.
- **Besoin en Fonds de Roulement** : Indique les ressources nécessaires pour financer le cycle d'exploitation.
- **Taux de Couverture FR / BFR** : Compare le fonds de roulement au besoin en fonds de roulement, indiquant la capacité de financement à court terme.
- **Ratio de Solvabilité (Ress Prop / Passifs)** : Évalue la capacité à rembourser les dettes à long terme en comparant les ressources propres aux passifs.
- **Trésorerie nette fin exercice** : Évalue la liquidité disponible après déduction des dettes à court terme.
- **Trésorerie nette en j / CAHT** : Exprime la trésorerie nette en jours de chiffre d'affaires hors taxes.
- **Trésorerie nette en j (EENE) / CAHT** : Exprime la trésorerie nette, y compris les éléments non économiques, en jours de chiffre d'affaires.
- **CAF (RN + Dot Amt)** : Calcule la capacité d'autofinancement en ajoutant le résultat net aux dotations aux amortissements.
- **Marge d'Endettement (FP - DLMT)** : Évalue la marge d'endettement en soustrayant les dettes à long terme des fonds propres.
- **Capacité de Remboursement (DLMT / CAF)** : Mesure la capacité à rem-

- bourser les dettes à long terme par rapport à la capacité d'autofinancement.
- **Dettes Structurelles / CAF** : Indique la charge des dettes structurelles par rapport à la capacité d'autofinancement.
 - **Dettes Structurelles / Cap Prop** : Compare les dettes structurelles aux capitaux propres, montrant l'équilibre financier.
 - **Variation CA** : Mesure la variation du chiffre d'affaires d'une période à l'autre.
 - **Marge Brute** : Calcule la différence entre le chiffre d'affaires et le coût des biens vendus.
 - **Marge Comm / CAHT** : Exprime la marge commerciale par rapport au chiffre d'affaires hors taxes.
 - **Valeur Ajoutée** : Mesure la contribution nette de l'entreprise à la richesse économique.
 - **Taux VA (VA / Pon + Ventes March)** : Mesure le taux de valeur ajoutée par rapport au produit net et aux ventes marchandes.
 - **Résultat d'Exploitation (RExp)** : Représente le bénéfice généré par les opérations principales avant charges financières et impôts.
 - **R.Ex / CA HT** : Exprime la rentabilité nette par rapport au chiffre d'affaires hors taxes.
 - **CAF / CAHT** : Mesure la proportion des bénéfices pouvant être réinvestis dans l'entreprise.
 - **Taux de Marge (EBE / VA)** : Mesure la rentabilité opérationnelle en comparant l'excédent brut d'exploitation à la valeur ajoutée.
 - **Excédent Brut d'Exploitation (EBE)** : Mesure le résultat avant intérêts, impôts, amortissements, et provisions.
 - **Clients en j / CAHT** : Mesure les créances clients en jours de chiffre d'affaires hors taxes.

-
- **BFR (EENE) en j / CAHT** : Exprime le besoin en fonds de roulement, y compris les éléments non économiques, en jours de chiffre d'affaires.
 - **FR en j / CAHT** : Exprime le fonds de roulement en jours de chiffre d'affaires hors taxes.
 - **Résultat de l'exercice / Capitaux propres** : Évalue la rentabilité des capitaux propres après distribution des résultats.
 - **Clients / Total Actif** : Mesure la part des créances clients dans le total des actifs.
 - **RExp / CA** : Compare le résultat d'exploitation au chiffre d'affaires.
 - **Charges Financières / EBE** : Compare les charges financières à l'excédent brut d'exploitation.

Annexe D

Hyperparamètres des Modèles

Arbre de Décision

- **criterion** : Fonction utilisée pour mesurer la qualité d'une division.
- **max_depth** : Nombre maximum de niveaux dans l'arbre.
- **min_samples_split** : Nombre minimum de données requises pour diviser un nœud.
- **min_samples_leaf** : Nombre minimum de données requises pour être dans une feuille.
- **max_features** : Nombre maximum de caractéristiques à considérer pour la meilleure division.

Forêt Aléatoire

- **n_estimators** : Nombre d'arbres dans la forêt aléatoire.
- **max_features** : Nombre maximum de fonctionnalités prises en compte pour diviser un nœud.
- **max_depth** : Nombre maximum de niveaux dans chaque arbre de décision.
- **min_samples_split** : Nombre minimum de données placées dans un nœud pour qu'il soit divisé.
- **min_samples_leaf** : Nombre minimal de données autorisées dans un nœud feuille.

- **bootstrap** : Méthode d'échantillonnage des points de données (avec ou sans remplacement).

XGBoost

- **n_estimators** : Nombre d'arbres dans le modèle XGBoost.
- **max_depth** : Profondeur maximale de chaque arbre.
- **learning_rate** : Taux d'apprentissage, contrôle la contribution de chaque arbre.
- **gamma** : Réduction minimale de la perte requise pour faire une division supplémentaire.
- **delta_step** : Pas minimum par lequel les poids sont mis à jour

LightGBM

- **num_leaves** : Nombre maximum de feuilles par arbre.
- **max_depth** : Profondeur maximale de chaque arbre.
- **learning_rate** : Taux d'apprentissage.
- **n_estimators** : Nombre d'arbres dans le modèle LightGBM.
- **min_child_samples** : Nombre minimum d'échantillons requis dans un nœud feuille.

AdaBoost

- **n_estimators** : Nombre de classifieurs faibles dans l'ensemble.
- **learning_rate** : Taux d'apprentissage, influence la contribution de chaque classifieur.

Bibliographie

- [1] Julien TEMIM. “The IFRS 9 impairment model and its interaction with the Basel framework”. *Moody’s analytics, Electronic Journal* 8 (2019).
- [2] Besma CHOUCANE. “Pertinence des normes comptables IAS/IFRS au contexte culturel tunisien”. *La Revue des Sciences de Gestion* 5 (2010), p. 129-140.
- [3] Arindam BANDYOPADHYAY. “IFRS 9 and CECL Credit Risk Modelling and Validation”. *Prajnan (National Institute of Bank Management)* 48.4 (2020), p. 369-370.
- [4] Leigh Joseph HALLIWELL. “Chain-ladder bias : Its reason and meaning”. *Variance* 1.2 (2007), p. 214-247.
- [5] E. ALPAYDIN. *Introduction to Machine Learning*. 4th. MIT Press, 2020.
- [6] Jason BROWNLEE. “How to choose a feature selection method for machine learning”. *Machine Learning Mastery* 10 (2019), p. 1-7.
- [7] Jason BROWNLEE. “Tactics to combat imbalanced classes in your machine learning dataset”. *Machine Learning Mastery* 19 (2015).
- [8] Z.-H. ZHOU. *Ensemble Methods : Foundations and Algorithms*. T. 1. CRC Press, 2012, p. 1-222.
- [9] William S VINCENT. *Django for Beginners : Build websites with Python and Django*. WelcomeToCode, 2022, p. 1-349.

-
- [10] L. ALLEN et A. SAUNDERS. “A Comparative Analysis of Current Credit Risk Models”. *Journal of Banking and Finance* 28.1 (2004), p. 59-117.
 - [11] Dodo Zaenal ABIDIN et al. “RSSI Data Preparation for Machine Learning”. *2020 International Conference on Informatics, Multimedia, Cyber and Information System (ICIMCIS)*. IEEE. 2020, p. 284-289.
 - [12] T. SCHUERMANN. “Estimating Probabilities of Default”. *Journal of Banking & Finance* 28.6 (2004), p. 1293-1311.
 - [13] Fabian FISCHER. “The Accuracy Paradox of Algorithmic Classification”. *Conference Proceedings of the STS Conference Graz 2019, Critical Issues in Science, Technology and Society Studies, 6-7 May 2019*. Verlag der Technischen Universität Graz. 2019.
 - [14] S. M. LUNDBERG et S.-I. LEE. “A Unified Approach to Interpreting Model Predictions”. *Proceedings of the 31st International Conference on Neural Information Processing Systems*. 2017, p. 4768-4777.