Université Catholique de l'Afrique de l'Ouest
Faculté des Sciences de Gestion
Foi – Science – Action
INSTITUT SUPERIEUR DE GESTION SAINT MICHEL,
Ecole Affiliée

DEPARTEMENT SCIENCES APPLIQUEES & DISCIPLINES CONNEXES

MEMOIRE DE FIN D'ETUDES

En vue de l'obtention du diplôme de **Licence Professionnelle en Informatique de Gestion**

TITRE DU MEMOIRE :

« Conception d'un modèle de scoring de crédit fiable et interprétable, et création d'un tableau de bord décisionnel à l'aide du Machine Learning »

Mémoire réalisé par ***Ndeye Maimouna GUEYE*** et ***Adji Awa NDAW***

Dirigé par ***Chamsedine AIDARA***

Promotion **2024-2025**

Sommaire

INTRODUCTION GÉNÉRALE

Dans un contexte économique mondial de plus en plus incertain et concurrentiel, les institutions financières sont confrontées à de nombreux défis pour préserver leur rentabilité et leur stabilité. Parmi ces défis, la maîtrise du risque de crédit occupe une place centrale, car il représente l'un des risques majeurs pour les banques et sociétés de financement. La capacité d'une institution à évaluer correctement la solvabilité de ses clients et à anticiper les défauts de paiement conditionne non seulement sa pérennité financière mais aussi sa compétitivité sur le marché.

Depuis plusieurs années, le secteur bancaire connaît une transformation profonde, portée par l'essor des technologies numériques. La digitalisation des services financiers a permis d'automatiser de nombreuses procédures, de diversifier les canaux de distribution, et de proposer des services personnalisés et accessibles à distance. Dans ce cadre, la gestion du risque de crédit s'appuie de plus en plus sur des outils numériques et des systèmes d'aide à la

décision exploitant les données clients et les capacités de l'intelligence artificielle. Cette digitalisation favorise le traitement rapide et fiable des demandes de crédit, tout en renforçant la maîtrise des risques associés.

Le risque de crédit, défini comme la probabilité qu'un emprunteur ne rembourse pas tout ou partie de son prêt aux échéances convenues, constitue un enjeu stratégique pour les établissements financiers. Une évaluation imprécise de ce risque peut engendrer des pertes importantes, affecter les résultats financiers, et nuire à la réputation de l'institution. À l'inverse, une gestion efficace du risque de crédit permet non seulement de sécuriser le portefeuille de prêts, mais aussi d'optimiser les décisions d'octroi et de fidéliser les clients solvables. C'est pourquoi les banques s'appuient sur des modèles statistiques et des systèmes automatisés pour prédire le comportement des emprunteurs et ajuster leurs décisions en conséquence.

Dans ce contexte de transformation numérique et de pression concurrentielle, le recours au **scoring automatisé** s'est généralisé au sein des institutions financières. Il s'agit d'attribuer à chaque demandeur de crédit un score représentant son niveau de risque, calculé à partir de ses caractéristiques personnelles, professionnelles et financières. Cette approche permet de standardiser et d'objectiver le processus de décision, de réduire les délais de traitement, et d'améliorer la qualité des portefeuilles de crédits. L'intégration de techniques avancées de **Machine Learning** dans ces systèmes offre la possibilité de concevoir des modèles plus performants et mieux adaptés à la diversité des profils clients.

Malgré les avancées technologiques, concevoir un modèle de scoring de crédit fiable et explicable demeure un défi, notamment dans des environnements marqués par des données hétérogènes et parfois déséquilibrées. Par ailleurs, les régulateurs et les décideurs financiers exigent des systèmes transparents, capables de justifier les décisions prises. Dès lors, la problématique à laquelle ce mémoire souhaite répondre est la suivante :

Comment concevoir un modèle de scoring de crédit fiable, performant et interprétable, capable de prédire le risque de non-remboursement d'un client à partir de ses données personnelles et financières, afin d'optimiser la gestion des risques dans une institution bancaire ?

Ce mémoire vise à atteindre plusieurs objectifs.

- **Objectif général** :
  Concevoir et évaluer un modèle de scoring de crédit s'appuyant sur des techniques de Machine Learning, capable de prédire le risque de défaut des clients à partir de données personnelles et financières.
- **Objectifs spécifiques** :
  - Réaliser une revue théorique et conceptuelle du scoring de crédit et des techniques de Machine Learning adaptées.
  - Préparer et analyser un jeu de données représentatif issu du secteur bancaire.
  - Développer et comparer plusieurs modèles prédictifs en utilisant différentes techniques d'apprentissage supervisé.

- Évaluer les performances et l'interprétabilité des modèles retenus à l'aide de métriques et d'outils explicatifs adaptés.
- Intégrer le modèle final dans un tableau de bord décisionnel interactif pour faciliter son exploitation par les analystes financiers.

Pour répondre à cette problématique, une démarche expérimentale sera adoptée. Elle consistera dans un premier temps à collecter et explorer un dataset contenant des informations personnelles et financières de clients. Ensuite, différentes étapes de préparation des données seront réalisées : nettoyage, transformation, création de nouvelles variables et gestion du déséquilibre éventuel des classes.
Plusieurs algorithmes de Machine Learning seront ensuite implémentés et testés dans l'environnement **Python**, via des bibliothèques spécialisées telles que **Scikit-learn**, **XGBoost** et **SHAP** pour l'interprétabilité. L'expérimentation sera conduite à l'aide de **Jupyter Notebook**, et les résultats seront exploités à travers un **tableau de bord interactif développé avec Power BI ou Streamlit**.

Enfin, le mémoire sera structuré autour de **quatre chapitres complémentaires et progressifs**. Le **premier chapitre** sera consacré au cadre théorique et conceptuel du scoring de crédit. Il présentera les définitions, les objectifs, les typologies de scoring ainsi que les notions clés du Machine Learning appliquées à la gestion du risque de crédit. Ce chapitre permettra d'établir les bases nécessaires à la compréhension des concepts mobilisés dans ce travail.

Le **deuxième chapitre** décrira la méthodologie adoptée pour la mise en œuvre du projet. Il détaillera la nature et l'origine des données utilisées, les techniques de préparation des données, les outils et technologies choisis ainsi que l'approche expérimentale suivie pour la conception du modèle de scoring.

Le **troisième chapitre** sera dédié à la construction du modèle de scoring. Il exposera les étapes de sélection des variables explicatives, l'entraînement et la validation de plusieurs algorithmes, ainsi que l'analyse de leurs performances respectives. Une

attention particulière sera accordée à l'interprétabilité du modèle retenu, à travers des méthodes explicatives adaptées au contexte bancaire.

Le **quatrième chapitre** traitera du déploiement et de l'exploitation des résultats obtenus. Il expliquera comment le modèle développé peut être utilisé pour scorer de nouveaux clients, analyser les prédictions, et être intégré dans un tableau de bord décisionnel interactif à destination des analystes et conseillers bancaires. Ce chapitre évoquera également les limites du modèle et proposera des pistes d'amélioration pour de futurs travaux.

Enfin, le mémoire s'achèvera par une **conclusion générale** qui résumera les principaux apports du projet, ses contributions tant sur le plan technique que décisionnel, et

ouvrira des perspectives de recherche et d'applications concrètes dans le domaine de la finance et du crédit.

CHAPITRE 1 : CADRE THÉORIQUE ET CONCEPTUEL

1.1. Définition et principes du scoring de crédit

Le **scoring de crédit** est un processus statistique et analytique qui vise à évaluer la solvabilité d'un individu ou d'une entreprise en attribuant un score synthétique à partir d'informations personnelles, professionnelles et financières. Ce score reflète la probabilité qu'un demandeur de crédit respecte ses engagements de remboursement.

Concrètement, le scoring repose sur l'analyse de variables explicatives telles que le niveau de revenu, l'ancienneté professionnelle, l'historique de crédit ou encore le taux d'endettement. Ces données sont croisées et traitées à l'aide de modèles statistiques ou algorithmiques qui permettent de prédire le comportement futur de remboursement d'un client, sur la base de comportements observés sur des populations similaires.

Les principes fondamentaux du scoring de crédit reposent sur :

- **La quantification du risque** : assigner une valeur numérique au risque de défaut.
- **L'objectivité de l'évaluation** : remplacer le jugement subjectif de l'analyste par un modèle fondé sur des données et des probabilités.
- **La segmentation des populations** : distinguer les bons et les mauvais payeurs pour ajuster les politiques d'octroi et de tarification du crédit.
- **L'automatisation des décisions de crédit** : rendre le processus plus rapide, standardisé et reproductible.

1.1.1. Historique du scoring dans le secteur bancaire

Le concept de scoring de crédit trouve son origine au début du XXe siècle, mais c'est véritablement à partir des années 1950 qu'il a pris son essor dans le secteur bancaire, en particulier aux **États-Unis**, avant de se généraliser à l'échelle mondiale.

Les prémices du scoring de crédit

Avant l'apparition des techniques de scoring, l'octroi de crédit reposait principalement sur le **jugement subjectif des agents de crédit** et des responsables d'agences bancaires. Les décisions étaient fondées sur des entretiens personnels, des relations de proximité, et l'analyse manuelle des dossiers financiers. Ce système, bien que intuitif et humain, présentait de nombreux inconvénients : il était lent, coûteux, difficile à standardiser et susceptible d'être influencé par des biais personnels ou des préférences discriminatoires.

L'avènement des modèles statistiques dans les années 1950

Avec l'essor des méthodes statistiques et des premiers ordinateurs dans les années 1950, plusieurs institutions financières américaines ont commencé à formaliser et à automatiser le processus de décision en matière de crédit. **Fair, Isaac and Company** (aujourd'hui connue sous le nom de **FICO**), fondée en 1956 par William R. Fair et Earl J. Isaac, fut pionnière en la matière. Ils ont mis au point les premiers modèles de scoring de crédit basé sur des méthodes statistiques, notamment la **régression linéaire** et plus tard la **régression logistique**.

Ces premiers systèmes attribuaient un score numérique à chaque demandeur de crédit en fonction de caractéristiques objectives comme l'âge, le revenu, le nombre d'années de travail, ou encore l'historique de crédit. Cette notation permettait d'évaluer de façon plus standardisée le risque de non-remboursement.

Généralisation dans les années 1970-1980

Dans les années 1970, sous l'effet de la croissance économique et de la diversification des produits de crédit (cartes bancaires, prêts automobiles, prêts personnels), le scoring s'est démocratisé. Aux États-Unis, des lois telles que le **Fair Credit Reporting Act (FCRA)** de 1970 ont encadré l'utilisation des informations personnelles dans le cadre de l'octroi de crédit, ce qui a encouragé la professionnalisation et la transparence des méthodes de scoring.

Dans les années 1980, les banques européennes et asiatiques ont progressivement adopté ces techniques face à la montée des risques liés au crédit à la consommation et à la nécessité de moderniser leurs processus décisionnels.

Les mutations technologiques et le Big Data

L'arrivée des **technologies de l'information** dans les années 1990 et 2000 a considérablement transformé le scoring de crédit. Les banques ont pu collecter, stocker et exploiter de grands volumes de données structurées et non structurées, issues notamment des comportements en ligne, des transactions par carte bancaire, et des interactions avec les services clients.

Cette évolution a conduit au développement de modèles de scoring plus complexes, intégrant de nouvelles variables et reposant sur des méthodes avancées telles que les **arbres de décision**, les **réseaux de neurones** et les **algorithmes de machine learning**. Ces outils offrent des performances prédictives supérieures et permettent de détecter des signaux faibles dans le comportement des emprunteurs.

Le scoring à l'ère de l'intelligence artificielle

Aujourd'hui, avec l'essor de l'**intelligence artificielle** et du **Big Data**, le scoring de crédit est en pleine mutation. Les institutions financières exploitent des sources de données toujours

plus diversifiées (réseaux sociaux, données mobiles, historiques de navigation) et utilisent des techniques de **machine learning** et de **deep learning** pour améliorer la précision des scores.

Toutefois, cette sophistication s'accompagne de nouveaux défis, notamment en matière de **transparence**, **d'interprétabilité** des modèles et de **protection des données personnelles**. Les régulateurs, comme la **Banque Centrale Européenne (BCE)** et la **Commission Nationale de l'Informatique et des Libertés (CNIL)** en France, imposent désormais des règles strictes pour garantir un usage éthique et responsable de ces outils.

1.1.2. Objectifs principaux du scoring

Le scoring de crédit remplit plusieurs fonctions majeures pour les établissements financiers :

- **L'acceptation du crédit** : Il permet de décider rapidement si une demande de prêt peut être acceptée ou refusée, et à quelles conditions (montant, taux, garanties).
- **La gestion du risque** : Il sert à segmenter la clientèle en fonction de son profil de risque et à adapter les stratégies de gestion et de surveillance du portefeuille.
- **Le recouvrement** : En phase de suivi et de recouvrement, le scoring permet d'anticiper les difficultés de paiement et de prioriser les actions de relance.

1.1.3. Avantages du scoring automatisé par rapport au jugement humain

Contrairement aux décisions reposant uniquement sur l'expertise humaine, le scoring automatisé présente plusieurs avantages :

- **Objectivité** : Il applique les mêmes critères et règles à tous les clients, ce qui réduit les biais subjectifs et garantit un traitement équitable.
- **Rapidité** : Les scores sont générés en quelques secondes, permettant des décisions quasi instantanées.
- **Cohérence** : Les décisions prises sont homogènes et reproductibles, indépendamment de l'agent ou du moment de la demande.
- **Optimisation du portefeuille** : Les institutions peuvent mieux contrôler leur exposition au risque et adapter leurs stratégies commerciales et financières.

1.1.4. Limites et critiques des systèmes de scoring classiques

Malgré leurs avantages, les systèmes de scoring traditionnels présentent certaines limites :

- **Risque d'exclusion financière** : Certains profils atypiques, comme les travailleurs indépendants ou les jeunes sans historique bancaire, peuvent être injustement défavorisés.
- **Rigidité des modèles** : Les modèles statistiques classiques peinent à intégrer des données non structurées ou de nouvelles variables issues du Big Data.

- **Manque d'interprétabilité** : Certains modèles avancés, notamment issus du Machine Learning, peuvent être perçus comme des « boîtes noires », ce qui complique leur utilisation dans des environnements réglementés.

1.2. Typologies de scoring

Le scoring de crédit se décline en plusieurs catégories selon l'objectif recherché et le stade du cycle de vie du crédit.

1.2.1. Scoring d'acceptation

Ce type de scoring intervient **avant l'octroi du crédit**. Il permet d'évaluer la **capacité de remboursement potentielle d'un nouveau client** à partir de données historiques ou déclaratives. L'objectif est de déterminer si le client est "accepté" ou "refusé" pour l'obtention du crédit.

Principaux critères analysés

- Informations personnelles : âge, situation familiale, niveau d'éducation
- Données professionnelles : statut (CDI, CDD, freelance), ancienneté, secteur d'activité
- Données financières : salaire mensuel, charges, taux d'endettement
- Historique bancaire : incidents de paiement, nombre de prêts en cours

Ce score est déterminant car il engage la banque à moyen ou long terme.

1.2.2. Scoring comportemental

Ce type de scoring est utilisé **pendant la durée de vie du crédit**. Il permet d'évaluer l'évolution du risque client en fonction de son comportement réel, observé au fil du temps. Cela peut influencer le réajustement des conditions de crédit (limite, taux, durée, etc.)

Critères pris en compte :

- Historique des paiements : ponctualité, retards, impayés récents
- Solde moyen du compte bancaire
- Utilisation des produits financiers (cartes, découverts, etc.)
- Fréquence des appels ou litiges avec le service client

Le scoring comportemental permet aussi de **prévenir les défauts de paiement** en activant des alertes précoces.

1.2.3. Scoring de recouvrement : prédiction de la probabilité de remboursement

Appliqué **lorsqu'un client est déjà en situation d'impayé**, ce scoring estime la **probabilité de récupération du crédit en défaut**. Il aide à prioriser les actions de recouvrement en fonction du profil du client.

Variables typiques :

- Montant impayé
- Durée de l'impayé
- Historique de remboursement partiel
- Données socio-économiques (revenu, emploi actuel, etc.)

Selon le score, le dossier peut être traité par relance simple, recouvrement amiable, ou transmis à une agence judiciaire.

1.2.4. Comparaison et interactions entre les types de scoring

Bien que distincts, ces trois types de scoring sont complémentaires. Un scoring d'acceptation efficace réduit le risque initial, un scoring comportemental permet de détecter précocement les incidents de paiement, et un scoring de recouvrement maximise les chances de récupération en cas de défaut.

1.3. Notions clés en Machine Learning appliqué au scoring

L'utilisation du **Machine Learning** dans le scoring de crédit permet d'améliorer les performances prédictives en exploitant des techniques plus complexes et en intégrant un plus grand volume de données.

1.3.1. Apprentissage supervisé : principe et utilité

Le scoring de crédit repose principalement sur l'apprentissage supervisé, une catégorie d'algorithmes de Machine Learning qui apprennent à partir de données étiquetées (avec une variable cible indiquant le remboursement ou le défaut). Le modèle est entraîné sur des données historiques pour identifier les relations entre les variables explicatives et la probabilité de défaut.

1.3.2. Algorithmes couramment utilisés

Parmi les algorithmes de scoring les plus utilisés, on retrouve :

- **Régression logistique** : méthode classique offrant une bonne interprétabilité.
- **Arbres de décision** : outil simple et visuel.

- **Random Forest** : ensemble d'arbres de décision réduisant le risque de surapprentissage.
- **XGBoost** : algorithme de gradient boosting performant et adapté aux grands volumes de données.

1.3.3. Concepts techniques essentiels

- **Overfitting (sur-apprentissage)** : lorsque le modèle est trop complexe et s'adapte trop aux données d'entraînement, au détriment des nouvelles données.
- **Underfitting (sous-apprentissage)** : lorsque le modèle est trop simple pour capter la complexité des relations.
- **Validation croisée** : technique d'évaluation qui consiste à diviser les données en plusieurs sous-ensembles pour tester la robustesse du modèle.
- **Tuning des hyperparamètres** : processus d'ajustement des paramètres internes de l'algorithme pour optimiser ses performances.

1.3.4. Mesures de performance

Pour évaluer les performances d'un modèle de scoring, plusieurs indicateurs sont utilisés :

- **Accuracy** : proportion de prédictions correctes.
- **Recall (sensibilité)** : capacité à identifier les cas de défaut.
- **F1-Score** : moyenne harmonique entre précision et rappel.
- **Courbe ROC et AUC** : indicateurs graphiques de la performance globale.

1.3.5. Importance de l'interprétabilité des modèles ML en milieu bancaire

Dans le secteur bancaire, les décisions doivent être justifiables, tant pour des raisons réglementaires que pour maintenir la confiance des clients. Il est donc crucial que les modèles soient interprétables, notamment via des outils comme :

- **SHAP (SHapley Additive exPlanations)** : qui quantifie l'impact de chaque variable sur la décision.
- **LIME (Local Interpretable Model-agnostic Explanations)** : qui explique localement le comportement du modèle pour un cas particulier.

CHAPITRE 2 : MÉTHODOLOGIE DU PROJET

2.1. Description et exploration des données

2.1.1. Présentation du dataset utilisé (origine, structure, volume)

Le dataset utilisé dans ce projet provient d'un ensemble de données lié à l'octroi de prêts bancaires. Il est couramment utilisé pour des projets de classification en apprentissage automatique, notamment dans le cadre de la prédiction de l'acceptation ou du rejet d'une demande de prêt.

Ce jeu de données contient initialement **614 enregistrements** et **13 variables**, dont une variable cible appelée Loan_Status, qui indique si le prêt a été approuvé (Y) ou non (N). Les autres colonnes comprennent des informations personnelles, professionnelles et financières sur les demandeurs de prêt.

Les colonnes du dataset sont les suivantes :

- **Personnelles** : Gender, Married, Dependents, Education, Self_Employed, Property_Area
- **Financières** : ApplicantIncome, CoapplicantIncome, LoanAmount, Loan_Amount_Term
- **Historiques** : Credit_History, Loan_Status

Voici un aperçu de la structure du dataset :

```
df = pd.read_csv('../data/train.csv')
df.head()
```

	Loan_ID	Gender	Married	Dependents	Education	Self_Employed	ApplicantIncome	CoapplicantIncome	LoanA
0	LP001002	Male	No	0	Graduate	No	5849	0.0	
1	LP001003	Male	Yes	1	Graduate	No	4583	1508.0	
2	LP001005	Male	Yes	0	Graduate	Yes	3000	0.0	
3	LP001006	Male	Yes	0	Not Graduate	No	2583	2358.0	
4	LP001008	Male	No	0	Graduate	No	6000	0.0	

2.1.2. Types de variables disponibles (personnelles, financières, historiques)

Le dataset se divise en différentes catégories de variables qui peuvent être classées comme suit :

- **Variables Personnelles** : Ces variables se rapportent aux informations personnelles des demandeurs de prêts, telles que :
    - Gender : Sexe du demandeur (catégorielle).
    - Married : État civil du demandeur (catégorielle).
    - Dependents : Nombre de personnes à charge (catégorielle).
    - Education : Niveau d'éducation du demandeur (catégorielle).
    - Self_Employed : Statut professionnel du demandeur (catégorielle).
    - Property_Area : Zone géographique de la propriété (catégorielle).

- **Variables Financières** : Ces variables concernent les aspects financiers du demandeur de prêt :
    - ApplicantIncome : Revenu mensuel du demandeur (numérique).
    - CoapplicantIncome : Revenu mensuel du co-demandeur (numérique).
    - LoanAmount : Montant demandé pour le prêt (numérique).
    - Loan_Amount_Term : Durée du prêt (numérique).
- **Variables Historiques** : Ces variables sont liées à l'historique de crédit du demandeur :
    - Credit_History : Historique de crédit du demandeur, indiquant s'il a eu des antécédents de remboursement de crédit (numérique, où 1 indique un bon historique et 0 un mauvais historique).
    - Loan_Status : Statut du prêt (cible), indiquant si le prêt a été approuvé (Y) ou refusé (N).

2.1.3. Analyse exploratoire des données : statistiques descriptives, visualisation, identification des outliers

Une analyse exploratoire des données (EDA) a été menée pour mieux comprendre la distribution et les relations entre les différentes variables.

- **Statistiques descriptives** :
  Les statistiques de base ont été calculées pour les variables numériques afin d'obtenir des informations sur leur distribution, leur moyenne, leur médiane, leur écart-type, etc. Cela permet de détecter rapidement des anomalies ou des valeurs extrêmes.
  Voici un aperçu à travers quelques captures :

    - La moyenne

```
[ ]  df.mean(numeric_only=True)
```

```
ApplicantIncome      5403.459283
CoapplicantIncome    1621.245798
LoanAmount            146.412162
Loan_Amount_Term      342.000000
Credit_History          0.842199
dtype: float64
```

- La mediane

```
] df.median(numeric_only=True)
```

```
ApplicantIncome      3812.5
CoapplicantIncome    1188.5
LoanAmount            128.0
Loan_Amount_Term      360.0
Credit_History          1.0
dtype: float64
```

- l'ecart type

```
] df.std(numeric_only=True)
```

```
ApplicantIncome      6109.041673
CoapplicantIncome    2926.248369
LoanAmount             85.587325
Loan_Amount_Term       65.120410
Credit_History          0.364878
dtype: float64
```

- **Identification des valeurs manquantes** :
  La présence de valeurs manquantes dans les données a été vérifiée. Par exemple, dans notre dataset, des valeurs manquantes (Nan) ont été repérées notamment : la colonne Gender a 13 valeurs manquantes, et Credit_History en compte 50. Ces valeurs ont été identifiées comme suit :

  - Les valeurs Manquantes

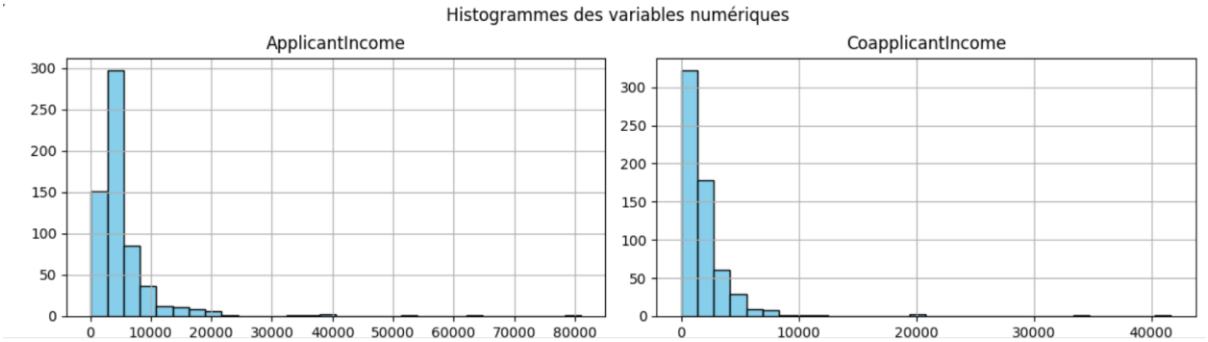
```
[ ] df.isnull().sum()
```

```
Loan_ID              0
Gender              13
Married              3
Dependents          15
Education            0
Self_Employed       32
ApplicantIncome      0
CoapplicantIncome    0
LoanAmount          22
Loan_Amount_Term    14
Credit_History      50
Property_Area        0
Loan_Status          0
dtype: int64
```

- **Visualisation des données** :
  Des histogrammes ont été générés pour les variables numériques afin de visualiser la

répartition des données. De plus, des boxplots ont été créés pour identifier les outliers dans des variables comme ApplicantIncome, CoapplicantIncome et LoanAmount. Cela permet de détecter d'éventuelles anomalies dans les données qui pourraient influencer les modèles prédictifs.
Voici des exemples de captures :

```
import matplotlib.pyplot as plt
import seaborn as sns
df.hist(bins=30, figsize=(12, 10), color='skyblue', edgecolor='black')
plt.suptitle("Histogrammes des variables numériques")
plt.tight_layout()
plt.show()
```



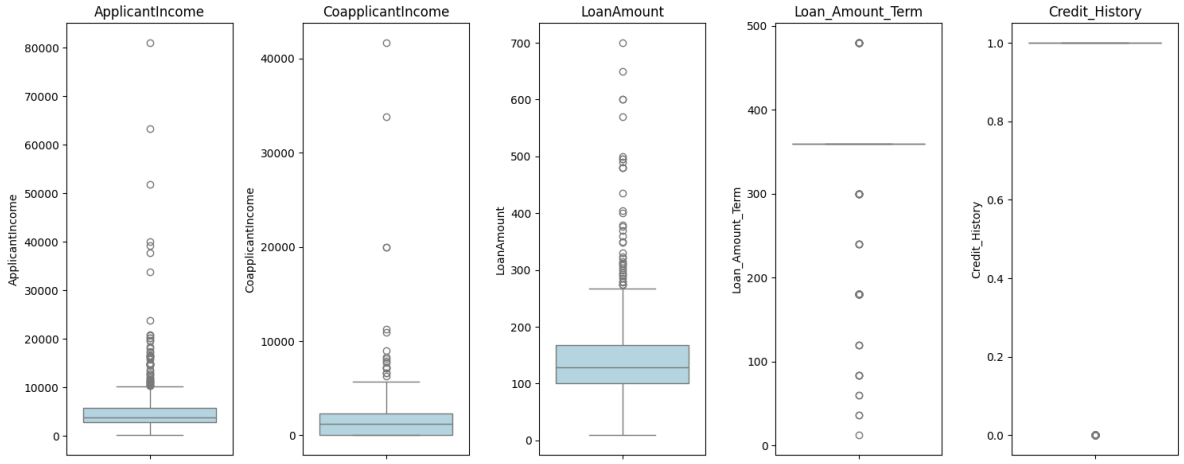

```python
import matplotlib.pyplot as plt
import seaborn as sns

# Sélection automatique des colonnes numériques
num_cols = df.select_dtypes(include=['int64', 'float64']).columns

# Création des box plots
plt.figure(figsize=(15, 6))

for i, col in enumerate(num_cols):
    plt.subplot(1, len(num_cols), i + 1)
    sns.boxplot(y=df[col], color='lightblue')
    plt.title(col)
    plt.tight_layout()

plt.show()
```



2.1.4. Corrélations et relations entre les variables

Une analyse des corrélations a été réalisée pour identifier les relations entre les variables. Par exemple, il a été observé que le revenu des demandeurs de prêts (ApplicantIncome) est fortement corrélé avec le montant du prêt demandé (LoanAmount). Ces relations sont cruciales pour la création des modèles prédictifs, car elles aident à comprendre l'impact relatif des différentes variables sur le statut du prêt.

2.2. Préparation des données

2.2.1. Nettoyage des données : traitement des valeurs manquantes et aberrantes

Une fois l'analyse exploratoire effectuée, le nettoyage des données a commencé. Cela a impliqué la gestion des valeurs manquantes et des outliers.

- **Traitement des valeurs manquantes** :
  Les valeurs manquantes dans les variables catégorielles telles que Gender, Married, Dependents, Self_Employed ont été imputées par la modalité la plus fréquente, c'est-à-dire le **mode**. Cela garantit que les informations manquantes sont remplacées par des valeurs représentatives du groupe.
  Les variables numériques, telles que LoanAmount, ont été imputées par la **médiane**, car cela permet de réduire l'impact des valeurs extrêmes sur l'analyse.

Imputation des Valeurs Manquantes (NaN)

```
# Catégorielles : Remplacement par la valeur la plus fréquente (mode)
df['Gender'].fillna(df['Gender'].mode()[0], inplace=True)
df['Married'].fillna(df['Married'].mode()[0], inplace=True)
df['Dependents'].fillna(df['Dependents'].mode()[0], inplace=True)
df['Self_Employed'].fillna(df['Self_Employed'].mode()[0], inplace=True)

# Numériques : Remplacement par la médiane ou mode selon la nature
df['LoanAmount'].fillna(df['LoanAmount'].median(), inplace=True)
df['Loan_Amount_Term'].fillna(df['Loan_Amount_Term'].mode()[0], inplace=True)
df['Credit_History'].fillna(df['Credit_History'].mode()[0], inplace=True)
```

✓ 0.0s

- **Traitement des outliers** :
  Les outliers ont été identifiés dans des variables financières telles que ApplicantIncome, CoapplicantIncome et LoanAmount. Ces valeurs extrêmes ont été détectées en utilisant l'IQR (Interquartile Range). Par exemple, un revenu de demandeur de plus de 10 000 a été considéré comme un outlier et supprimé.
  Après cette suppression, une nouvelle analyse des statistiques descriptives a montré des valeurs plus cohérentes pour ces variables.

```
# SUPRESSION DES OUTILIERS

borne_inf_applicant = -1498.75
borne_sup_applicant = 10171.25

borne_inf_coapplicant = -3445.875
borne_sup_coapplicant = 5743.125

borne_inf_loan = 3.5
borne_sup_loan = 261.5

# Suppression des outliers pour toutes les colonnes concernées
df = df[
    (df['ApplicantIncome'] >= borne_inf_applicant) & (df['ApplicantIncome'] <= borne_sup_applicant) &
    (df['CoapplicantIncome'] >= borne_inf_coapplicant) & (df['CoapplicantIncome'] <= borne_sup_coapplicant) &
    (df['LoanAmount'] >= borne_inf_loan) & (df['LoanAmount'] <= borne_sup_loan)
]
```

2.2.2. Transformation des données : normalisation, standardisation, encodage

Certaines variables ont été transformées pour préparer les données à l'analyse.

- **Normalisation et standardisation** : Les variables comme ApplicantIncome et LoanAmount ont été normalisées afin de garantir que les modèles de Machine

Learning, comme la régression logistique ou les arbres de décision, ne soient pas influencés par des écarts d'échelle importants.

Standardisation

```
### Certaines colonnes ont des chiffres beaucoup plus grands que d'autres
# Les modèles peuvent mal apprendre si les échelles sont trop différentes
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()
df[['ApplicantIncome', 'CoapplicantIncome', 'LoanAmount']] = scaler.fit_transform(
    df[['ApplicantIncome', 'CoapplicantIncome', 'LoanAmount']]
)
```

✓ 10.2s

- **Encodage des variables catégorielles** : Les variables catégorielles comme Gender, Married, et Education ont été encodées à l'aide de **One-Hot Encoding**, ce qui a permis de transformer ces variables en valeurs binaires (0 et 1).

Encodage

```
### Transformons les colonnes categorielles en numerique
df['Gender'] = df['Gender'].map({'Male': 1, 'Female': 0})
df['Married'] = df['Married'].map({'Yes': 1, 'No': 0})
df['Education'] = df['Education'].map({'Graduate': 1, 'Not Graduate': 0})
df['Self_Employed'] = df['Self_Employed'].map({'Yes': 1, 'No': 0})
df['Loan_Status'] = df['Loan_Status'].map({'Y': 1, 'N': 0})  # Cible
```

✓ 0.0s

```
### La colonne Dependents a des valeurs comme 0, 1, 2, 3+. On transforme 3+ en 3
df['Dependents'] = df['Dependents'].replace('3+', 3).astype(int)
```

✓ 0.0s

```
### One-Hot Encoding : pour les colonnes avec plus de 2 catégories (comme Property_Area),
# Cela va créer 2 nouvelles colonnes : Urban et Rural
df = pd.get_dummies(df, columns=['Property_Area'], drop_first=True)
```

✓ 0.0s

2.2.3. Création de nouvelles variables pertinentes (feature engineering)

Afin d'augmenter la capacité explicative des modèles prédictifs, des variables dérivées ont été créées à partir des données existantes. Ces transformations ont pour but de mieux refléter la situation financière globale du demandeur, au-delà des informations brutes initiales.

Les nouvelles variables incluent notamment :

- **TotalIncome** : somme des revenus du demandeur principal et du co-demandeur (TotalIncome = ApplicantIncome + CoapplicantIncome). Cette variable permet d'estimer plus justement la capacité de remboursement globale du foyer.
- **EMI (Equivalent Monthly Installment)** : estimation de la mensualité du prêt, calculée comme le montant du prêt divisé par la durée (EMI = LoanAmount / Loan_Amount_Term). Elle permet d'évaluer la charge mensuelle du crédit.
- **Debt_to_Income Ratio** : ratio entre le montant du prêt demandé et le revenu total (LoanAmount / TotalIncome), utilisé pour mesurer l'effort financier demandé au foyer.

Ces variables ont été ajoutées au dataset après normalisation pour assurer leur intégration homogène dans les modèles de Machine Learning.

```
# TotalIncome (Addition des revenus du demandeur principal et du co-demandeur.
df['TotalIncome'] = df['ApplicantIncome'] + df['CoapplicantIncome']
```

✓ 0.0s

```
# EMI (Equivalent Monthly Installment) = Estimation de la mensualité à rembourser.
df['EMI'] = df['LoanAmount'] / df['Loan_Amount_Term']
```

✓ 0.0s

```
# Debt-to-Income Ratio (DTI) = Rapport entre la mensualité estimée et le revenu total.
df['Debt_to_Income'] = df['EMI'] / df['TotalIncome']
```

2.2.4. Gestion du déséquilibre des classes : techniques de sur-échantillonnage (SMOTE), sous-échantillonnage

L'analyse du jeu de données a révélé un déséquilibre significatif entre les deux classes Y et N de la variable cible Loan_Status. En effet, la majorité des prêts ont été approuvés (Y), ce qui peut biaiser les modèles prédictifs en faveur de la classe majoritaire.

Pour remédier à cela, nous avons appliqué la technique de **SMOTE (Synthetic Minority Over-sampling Technique)**. Elle consiste à générer artificiellement de nouveaux exemples

appartenant à la classe minoritaire, à partir des instances existantes, en interpolant les valeurs entre les plus proches voisins.

Cette approche permet d'obtenir un jeu de données équilibré, améliorant ainsi la capacité des modèles à détecter les cas de refus de prêt sans sur-représenter les cas d'acceptation.

```python
# Séparation des variables et de la cible
X = df.drop('Loan_Status', axis=1)
y = df['Loan_Status']
```

✓ 0.0s

```python
# Application de SMOTE
from imblearn.over_sampling import SMOTE

smote = SMOTE(random_state=42)
X_resampled, y_resampled = smote.fit_resample(X, y)
```

✓ 0.0s

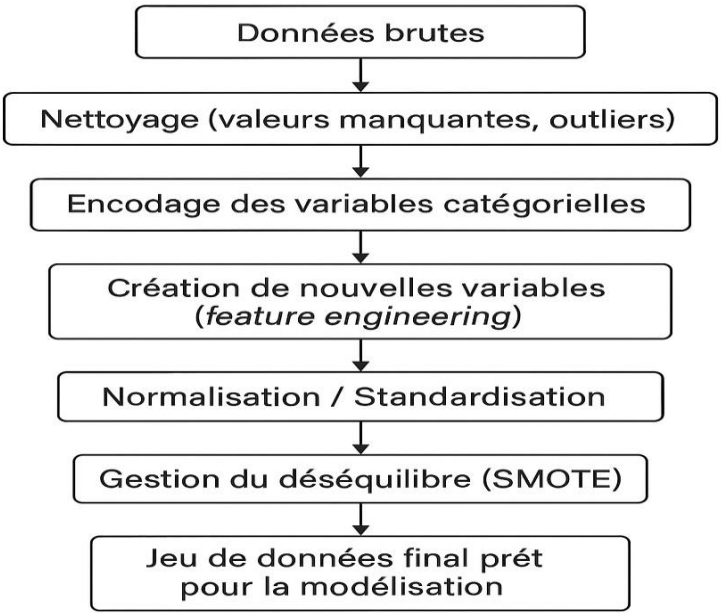
```python
# VERIFICATIONS , on doit avoir un resultat equilibre
import numpy as np

unique, counts = np.unique(y_resampled, return_counts=True)
print(dict(zip(unique, counts)))
```

✓ 0.0s

```
{np.int64(0): np.int64(372), np.int64(1): np.int64(372)}
```

Synthèse visuelle du pipeline de traitement des données :



2.3. Choix des outils et technologies

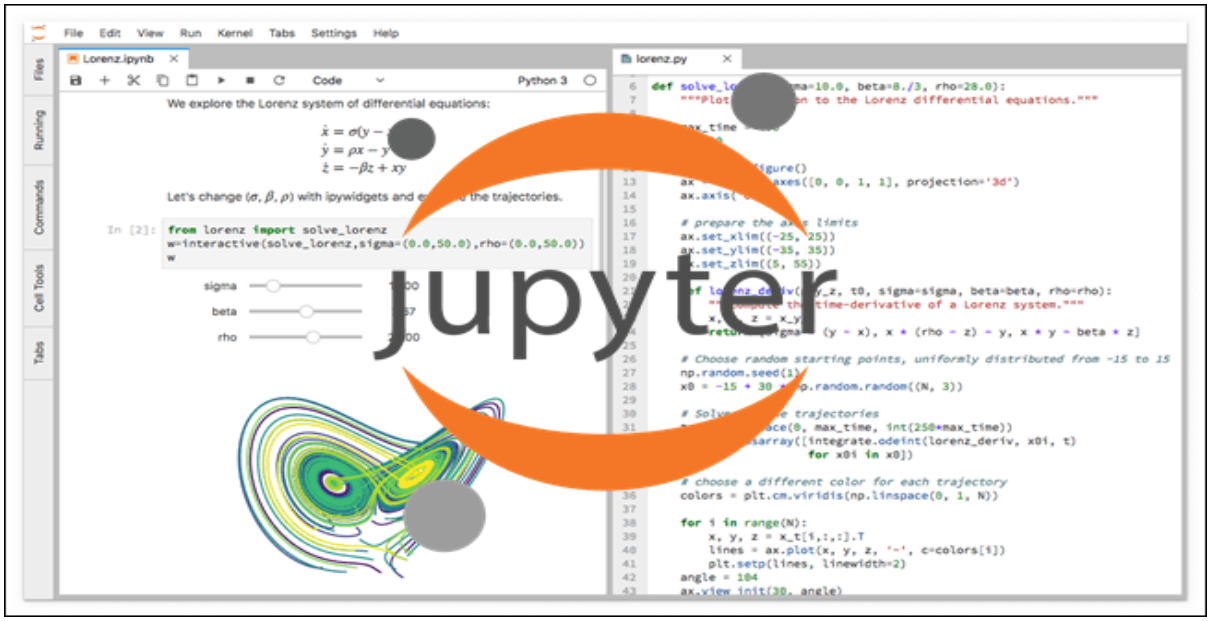
2.3.1. Environnement Python : bibliothèques pandas, numpy, scikit-learn, xgboost

L'environnement de développement pour ce projet est Python, avec plusieurs bibliothèques utilisées pour le traitement et la modélisation des données :

- **NumPy** : Calculs numériques et manipulation de matrices.
- **Pandas** : Manipulation des données et traitement des valeurs
- **Scikit-learn** : Création et évaluation des modèles de Machine Learning.
- **XGBoost** : Un algorithme de boosting qui a montré de bons résultats pour ce type de problème.

2.3.2. Jupyter Notebook pour les expérimentations et tests

Les analyses et expérimentations ont été réalisées dans Jupyter Notebook, un environnement interactif basé sur le langage Python, largement utilisé en data science et en Machine Learning. Il permet de combiner du **code exécutable, des visualisations graphiques, des commentaires textuels (markdown)** et des résultats en temps réel dans un même document. Cela offre une grande souplesse pour **tester différents modèles, analyser les performances, et documenter l'ensemble du processus** de façon claire et reproductible. Grâce à Jupyter Notebook, il est possible de suivre pas à pas l'évolution des données, les traitements appliqués, ainsi que les résultats obtenus, ce qui en fait un outil idéal.



2.3.3. Streamlit pour la visualisation des résultats

Pour la visualisation des résultats, le choix s'est porté sur **Streamlit**, une solution légère et rapide à déployer, particulièrement adaptée à un projet de data science. Elle permet de créer des applications web interactives en Python, sans compétences avancées en développement web. Streamlit a été préféré à Power BI en raison de sa flexibilité d'intégration directe avec les notebooks Jupyter et les bibliothèques Python utilisées dans ce projet.

2.3.4. GitHub pour la gestion de versions et la collaboration

GitHub est utilisé pour la gestion de version du code source et des fichiers associés, permettant un suivi efficace des modifications et la collaboration avec d'autres membres de l'équipe.


GitHub