AVIGNON UNIVERSITÉ

Démissions d'un organisme bancaire

Ibrahim Jallouli

6 avril 2025

Master 2 informatique
ILSEN

UE Business intelligence & Systèmes décisionnels
ECUE Application Business Intelligence

UFR
SCIENCES
TECHNOLOGIES
SANTÉ

AVIGNON UNIVERSITÉ
CERI

CENTRE
D'**ENSEIGNEMENT**
ET DE **RECHERCHE**
EN **INFORMATIQUE**
ceri.univ-avignon.fr

Sommaire

1 Présentation

Ce projet s'inscrit dans un cadre académique et vise à appliquer les connaissances acquises dans des disciplines telles que l'intelligence artificielle, le data mining et les systèmes d'aide à la décision. Il constitue une mise en pratique des méthodes d'analyse de données et de modélisation prédictive en contexte réel, permettant d'expérimenter des techniques avancées tout en répondant à une problématique concrète du secteur bancaire.

1.1 Contexte

Dans un environnement bancaire de plus en plus concurrentiel, la fidélisation des clients est un enjeu stratégique majeur. Pour répondre à cette problématique, nous allons développer une solution d'analyse prédictive permettant d'identifier les clients susceptibles de quitter un organisme financier. L'objectif de ce projet est de développer un modèle prédictif permettant d'évaluer le risque de démission des sociétaires en fonction de leurs caractéristiques. Ce modèle repose sur l'analyse de données historiques et actuelles, incluant des informations telles que les revenus, l'âge, la situation familiale et d'autres facteurs pertinents. En attribuant un score de risque à chaque client, la banque pourra ainsi détecter les profils à risque et mettre en place des actions ciblées pour renforcer la relation client et prévenir les départs.

Ce projet s'inscrit dans une démarche d'intelligence d'affaires et repose sur des techniques de science des données, allant du nettoyage et de la préparation des données à la mise en œuvre de modèles de classification supervisée. Le modèle principal utilisé sera la régression logistique, complété par d'autres algorithmes afin d'évaluer et comparer leur efficacité.

L'objectif final est de fournir une solution fiable, interprétable et facilement intégrable aux processus décisionnels de la banque, permettant ainsi d'améliorer la gestion de la relation client et de réduire le taux de démission.

1.2 Organisation

Initialement conçu pour être réalisé en binôme, ce projet a finalement été mené individuellement en raison de la saturation des groupes disponibles. Cette situation a représenté un véritable défi, notamment en raison de la charge de travail conséquente qui devait être assumée seul. La progression du projet a parfois été difficile, nécessitant des efforts supplémentaires et certains sacrifices personnels.

Malgré ces difficultés, une gestion rigoureuse du temps et une organisation méthodique ont permis d'atteindre les objectifs fixés. Le projet a été mené à bien grâce à une planification minutieuse des différentes phases : exploration et préparation des données, modélisation, validation, ainsi que la rédaction du rapport, réalisée parallèlement à l'avancement du travail.

Ce contexte a exigé une implication constante et une grande autonomie, rendant l'expérience particulièrement formatrice sur les plans technique et personnel.

1.3 Implémentation

L'implémentation du projet s'articule autour de plusieurs scripts Python organisés en étapes successives. Les bibliothèques utilisées et certaines parties du pipeline ont été inspirées du TP réalisé en classe dans le cadre du cours de fouille de données.

- **Fusion et nettoyage des données** : ce traitement est effectué dans le fichier `fusion_nettoyage_clients.py`. Il utilise principalement les bibliothèques `pandas` et `matplotlib`. Les fonctions permettent de charger deux tables, détecter et corriger les anomalies, calculer des variables comme l'âge ou l'ancienneté (ADH), classifier les clients en tranches, corriger les revenus et générer une figure de visualisation.
- **Prétraitement et visualisation** : dans `pretraitement_et_visualisation.py`, les données nettoyées sont préparées pour la modélisation. Cela inclut l'encodage des variables catégorielles (avec `OneHotEncoder` et `OrdinalEncoder`), la normalisation (avec `StandardScaler`), le traitement des valeurs manquantes et la génération de visualisations comme des histogrammes, boxplots et matrices de corrélation. Les bibliothèques utilisées sont notamment `pandas`, `numpy`, `seaborn` et `sklearn`.
- **Modélisation** : le script `modele_prediction.py` applique différents modèles de classification pour prédire les démissions. Il utilise des modèles de `scikit-learn` comme `LogisticRegression`, `SVC`, `CategoricalNB` et `KNeighborsClassifier`, avec suréchantillonnage via `RandomOverSampler`. Les performances sont évaluées avec des scores (accuracy, recall, F1) et visualisées par des matrices de confusion, histogrammes de probabilités et des représentations 3D (avec `PCA`).
- **Orchestration** : le fichier `main.py` orchestre les trois étapes précédentes dans un pipeline unique, assurant une exécution séquentielle cohérente du projet.

Un fichier `README.md` accompagne le projet. Il contient les instructions nécessaires à l'exécution du code dans n'importe quel environnement, incluant la liste des dépendances à installer et une description des étapes à suivre.

2 Présentation des données

L'analyse des données constitue une étape essentielle dans ce projet, car elle permet de mieux comprendre la structure des informations disponibles et d'anticiper les éventuelles difficultés liées à la qualité des données. Cette phase d'exploration vise à identifier les différentes catégories d'attributs, leur structuration et la manière dont ils sont codés. Une compréhension approfondie des données est nécessaire avant d'appliquer les méthodes de traitement et de modélisation.

2.1 Sources des données

Les données utilisées dans ce projet proviennent de deux fichiers CSV distincts :

- **table1.csv** : contient des informations sur **30 332** clients ayant quitté la banque entre **1999 et 2006**. Ces clients sont qualifiés de démissionnaires.
- **table2.csv** : représente un échantillon de **15 022** clients actuels de la banque, comprenant à la fois des clients actifs et des démissionnaires.

2.2 Liste des attributs et leur nature

Les attributs présents dans ces fichiers sont de plusieurs types :

- **Attributs d'identification** : `ID`.
- **Attributs démographiques** : `CDSEXE`, `DTNAIS`, `NBENF`.
- **Attributs financiers** : `MTREV`.
- **Attributs de relation bancaire** : `DTADH`, `CDTMT`, `CDCATCL`.
- **Attributs de démission** : `DTDEM`, `ANNEEDEM`, `CDMOTDEM`.
- **Attributs calculés et catégorisés** : `AGEAD`, `RANGAGEAD`, `AGEDEM`, `RANGAGEDEM`, `ADH`, `RANGADH`.

2.3 Nature et codage des valeurs

- **Variables catégorielles** :
  - `CDSEXE` : Code numérique représentant le sexe du client. Les valeurs précises doivent être décodées.
  - `CDSITFAM` : Code représentant la situation familiale du client.
  - `CDTMT` : Indique le statut du sociétaire dans la banque.
  - `CDCATCL` : Catégorie de client (classification interne).
  - `CDMOTDEM` : Motif de démission (uniquement pour les clients démissionnaires).
- **Variables numériques** :
  - `MTREV` : Montant des revenus du client (exprimé en unités monétaires).
  - `NBENF` : Nombre d'enfants (valeur entière).
  - `AGEAD`, `AGEDEM` : Âge du client lors de l'adhésion et au moment de la démission (en années).
  - `ADH` : Durée d'adhésion en années.
- **Variables temporelles** :
  - `DTNAIS` : Date de naissance du client (format date).
  - `DTADH` : Date d'adhésion (format date).
  - `DTDEM` : Date de démission (format date, uniquement pour les démissionnaires).
  - `ANNEEDEM` : Année de démission (extrait de `DTDEM`).
- **Variables catégorisées** :
  - `RANGAGEAD`, `RANGAGEDEM` : Tranche d'âge du client à l'adhésion et à la démission.
  - `RANGADH` : Tranche de la durée d'adhésion.

2.4 Interprétation des variables et unités utilisées

Certaines variables nécessitent une attention particulière :

- `CDSEXE` : Ce champ contient des valeurs numériques qui ne correspondent pas simplement aux genres homme/femme mais incluent des sous-catégories.
- `MTREV` : Cette variable exprime un montant en unités monétaires, mais l'échelle exacte n'est pas précisée dans les fichiers.
- `DTDEM` : Certaines valeurs sont renseignées comme `31/12/1900`, ce qui peut indiquer un client non démissionnaire dans `table2.csv`.
- `AGEAD` et `AGEDEM` : Ces valeurs peuvent être recalculées à partir de `DTNAIS` et `DTDEM` pour assurer la cohérence.
- `RANGAGEAD`, `RANGAGEDEM`, `RANGADH` : Ces champs classent les individus par tranche, ce qui permet d'éviter un traitement strictement numérique des âges et durées d'adhésion.

L'ensemble de ces informations permet de mieux appréhender la structure des données et d'anticiper les étapes de préparation nécessaires pour garantir un bon traitement lors de l'analyse prédictive.

3 Préparation des données

3.1 Nettoyage des données

3.1.1 Observation

Avant de pouvoir exploiter les données dans le cadre d'une analyse prédictive, il est essentiel de garantir leur qualité. Des données brutes peuvent contenir des erreurs, des incohérences ou des valeurs manquantes susceptibles d'entraîner des biais ou des performances médiocres lors de l'entraînement du modèle.

Un script Python a été développé pour analyser les fichiers `table1.csv` et `table2.csv`. Cette analyse a mis en évidence plusieurs problèmes :

- **Revenus nuls (`MTREV`)** : une large proportion de clients ont un revenu enregistré à zéro, ce qui peut refléter soit une réalité économique (étudiants, chômeurs), soit une absence de saisie.
- **Valeurs manquantes** : certaines colonnes présentent de nombreuses valeurs nulles, notamment `RANGADH` ou `CDMOTDEM`, cette dernière étant logiquement vide pour les clients encore actifs.
- **Codages complexes** : des variables comme `CDSEXE`, `CDSITFAM`, `CDTMT` ou `CDDEM` utilisent des codes numériques ou alphanumériques peu lisibles sans dictionnaire explicite.
- **Dates non standardisées** : les colonnes `DTADH`, `DTDEM` et `DTNAIS` sont exprimées sous forme de chaînes avec des formats irréguliers, contenant parfois des dates par défaut comme `31/12/1900` ou `0000-00-00`.
- **Redondances** : certaines informations sont présentes à la fois sous forme continue (`AGEAD`, `AGEDEM`, `ADH`) et sous forme discrétisée (`RANGAGEAD`, `RANGAGEDEM`, `RANGADH`).

- **Tranches mal codées** : les variables de type `RANG*` contiennent des valeurs mixtes combinant un indice et un libellé (ex. : `5 41-45`, `b 71-+`).

3.1.2 Traitements appliqués

À partir de ces constats, plusieurs opérations ont été menées :

- **Fusion des fichiers** : j'ai regroupé les deux jeux de données dans une seule table. Les colonnes manquantes ont été complétées par des valeurs nulles, afin d'obtenir une structure homogène entre les anciens clients et les clients actuels.
- **Ajout de la variable cible `DEMISSIONNAIRE`** : j'ai créé une variable binaire permettant d'identifier les clients démissionnaires (valeur 1) et les clients encore actifs (valeur 0). Pour cela, je me suis basé sur la date de démission : si la date est égale à la valeur par défaut `31/12/1900`, le client est considéré comme actif (0), sinon il est considéré comme démissionnaire (1).
- **Création de la variable `ADH`** : cette variable essentielle, représentant la durée d'adhésion, a été calculée à partir des colonnes `DTADH` et `DTDEM`. Pour les clients encore actifs (ayant `DTDEM` = `31/12/1900`), j'ai utilisé l'année 2007 comme référence.
- **Correction des revenus nuls (`MTREV`)** : la variable `MTREV`, représentant les revenus des clients, est essentielle dans le contexte bancaire, en particulier pour anticiper les risques de démission. Cependant, un nombre important d'observations contiennent une valeur nulle, ce qui rend cette variable peu informative dans l'état brut. Une imputation globale par la moyenne aurait pu fausser les résultats et masquer des différences structurelles importantes. Pour remédier à ce problème, une stratégie plus fine a été adoptée : j'ai calculé la moyenne des revenus pour chaque modalité de la variable `CDSITFAM` (situation familiale), puis utilisé cette moyenne pour remplacer les valeurs nulles (zéro) de `MTREV`. Cela permet de conserver la cohérence des profils clients tout en restaurant la valeur prédictive de la variable.
- **Complétion des variables ordinales (`RANG*`)** : les modèles de classification étant souvent plus efficaces avec des variables discrètes ou catégorielles, j'ai souhaité exploiter les tranches déjà présentes dans `table1.csv`, comme `RANGAGEAD`, `RANGAGEDEM` ou `RANGADH`. Ces informations étaient absentes ou incomplètes dans les données issues de `table2.csv`.
  Pour y remédier, j'ai d'abord calculé les âges à l'adhésion et à la démission (`AGEAD`, `AGEDEM`) à partir des dates disponibles (`DTADH`, `DTDEM`, `DTNAIS`), puis j'ai transformé ces valeurs continues en tranches ordinales, selon les intervalles définis dans le projet. De même, j'ai reconstitué la variable `RANGADH` à partir de la variable `ADH`. Ce travail m'a permis d'uniformiser l'ensemble des données, en renforçant la cohérence et la qualité des variables utilisées pour la modélisation.

Toutes les étapes précédentes ont été automatisées dans un script Python pour assurer la reproductibilité du nettoyage.

3.1.3 Adaptation des données pour l'apprentissage

La fusion des fichiers a entraîné des valeurs manquantes, dues à des différences structurelles. Par exemple, certaines variables (`RANGAGEDEM`, `RANGADH`) sont naturellement absentes pour les clients encore actifs. Ces valeurs nulles ont été traitées selon leur nature et leur utilité.

Certaines variables présentaient des formats complexes, notamment les `RANG*`, mêlant texte et chiffres. Un nettoyage a permis d'en extraire la partie numérique, exploitée ensuite comme variable ordinale.

Des valeurs par défaut (comme `31/12/1900` ou `0000-00-00`) ont été identifiées et traitées comme des indicateurs spécifiques. Les variables continues (`MTREV`, `ADH`, etc.) ont été standardisées pour une meilleure homogénéité.

L'objectif global était de transformer un maximum de variables en format numérique afin de faciliter les traitements statistiques et l'entraînement des modèles.

Les étapes principales ont été :

- **Traitement des valeurs manquantes** : imputations, exclusions ou transformation en modalités distinctes selon les cas.
- **Nettoyage des formats** : extraction des valeurs numériques, conversion des dates, harmonisation des représentations.
- **Encodage des variables catégorielles** : transformation en valeurs numériques adaptées.
- **Standardisation des variables numériques** : centrage-réduction pour optimiser les performances des modèles.
- **Suppression des attributs temporaires ou redondants** : comme `AGEAD`, `AGEDEM`, `ADH`, `DTNAIS`, `DTADH`, `ID`, après dérivation des variables utiles.

3.2 Analyse descriptive

Afin de mieux comprendre la structure et les propriétés des données, une analyse descriptive a été réalisée après l'application de plusieurs transformations. Des techniques telles que le *one-hot encoding*, l'encodage ordinal, ainsi que l'imputation des valeurs manquantes ont été appliquées aux attributs. Ces étapes ont permis d'obtenir une représentation exploitable des données et de visualiser la répartition des valeurs pour chaque variable, en identifiant notamment les valeurs atypiques (outliers), les déséquilibres et les éventuelles redondances entre attributs.

À l'issue des opérations de nettoyage, de transformation et de réduction, seuls les attributs suivants ont été conservés pour l'analyse et la modélisation :

`MTREV`, `CDSEXE`, `NBENF`, `CDSITFAM`, `CDTMT`, `CDMOTDEM`, `CDCATCL`, `RANGAGEAD`, `RANGAGEDEM`, `RANGADH`, `DEMISSIONNAIRE`

Chaque sous-section suivante décrit les traitements appliqués à un attribut, accompagnée d'une visualisation (histogramme ou boxplot) et d'une brève interprétation.

3.2.1 MTREV : Montant des revenus

Définition. `MTREV` représente le montant des revenus mensuels d'un client. Dans le contexte bancaire, cette variable est particulièrement pertinente car elle peut influencer fortement le comportement des clients vis-à-vis de leur fidélité ou de leur probabilité de démission.

Traitements appliqués. La variable `MTREV` a été préalablement nettoyée afin de résoudre le problème des revenus nuls, fréquents dans les données brutes. Plutôt qu'une imputation uniforme, une méthode par moyenne conditionnelle selon la situation familiale (`CDSITFAM`) a été retenue pour préserver les différences de profil client. Après remplacement, les valeurs ont été standardisées afin de les rendre plus compatibles avec les algorithmes d'apprentissage supervisé. Cette stratégie est justifiée par les différences observées entre les groupes, comme illustré dans cette Figure.

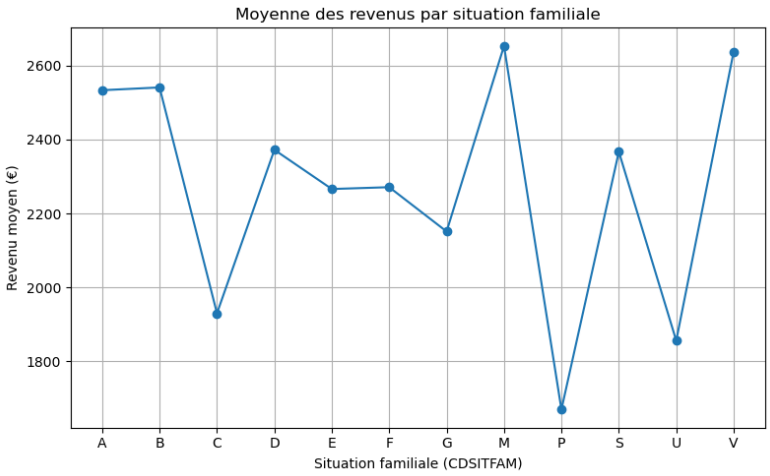

Figure 1. Revenu moyen par situation familiale (`CDSITFAM`)

Visualisation et analyse.

- **Boxplot** : le boxplot montre que la majorité des revenus se situent entre environ **1 500** et **3 100**. La médiane est proche de **2 500**. On observe également de nombreux *outliers* en dessous de 1 500 et au-dessus de 3 100, indiquant une forte dispersion. Cela confirme que certains clients présentent des revenus très faibles ou très élevés comparés à la majorité.
- **Histogramme** : on observe une distribution très asymétrique, avec une concentration massive autour de **2 500**. Cette concentration peut être due à l'effet de l'imputation ou à une réalité économique typique du profil client de la banque. Une deuxième bosse est visible autour de **1 900**, ce qui peut correspondre à une autre catégorie de sociétaires (étudiants, retraités...).

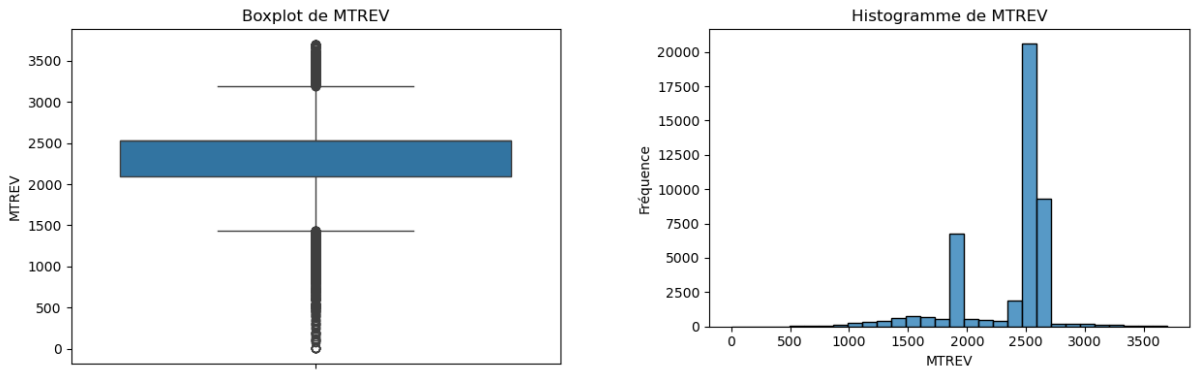

Figure 2. Distribution et boxplot de la variable `MTREV`

3.2.2 CDSEXE : Code sexe du client

Définition. L'attribut `CDSEXE` est un code catégoriel indiquant le sexe ou le genre du sociétaire. Contrairement à une classification binaire classique (homme/femme), ce champ contient plusieurs modalités (1, 2, 3, 4), correspondant probablement à des sous-catégories internes à la banque.

Traitements appliqués. Aucun nettoyage spécifique n'a été appliqué sur cet attribut, car les valeurs étaient déjà bien codées. Une transformation de type encodage ordinal a été utilisée pour conserver l'information tout en permettant son exploitation dans les modèles d'apprentissage.

Visualisation et analyse.

Le boxplot montre que les valeurs sont concentrées entre 1 et 4 sans valeurs aberrantes. L'histogramme permet d'observer une distribution fortement déséquilibrée : les valeurs 2 et 3 dominent largement, tandis que la catégorie 1 est marginale. Cela reflète probablement des codes internes propres à la banque, avec une majorité répartie sur deux sous-genres principaux.

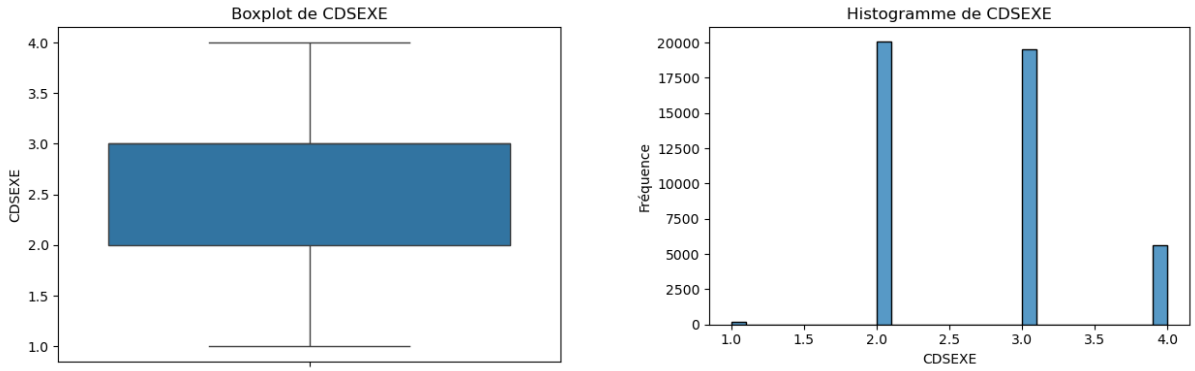

Figure 3. Distribution et boxplot de la variable `CDSEXE`

3.2.3 NBENF : Nombre d'enfants à charge

Définition. L'attribut `NBENF` représente le nombre d'enfants à charge pour chaque sociétaire. Il s'agit d'une variable numérique discrète, potentiellement corrélée à des

aspects économiques et sociaux du client.

Traitements appliqués. Aucune imputation ou transformation spécifique n'a été nécessaire pour cette variable, car elle était déjà bien structurée. Toutefois, elle a été normalisée (centrage-réduction) afin d'être compatible avec les modèles de classification utilisés.

Visualisation et analyse.

Le boxplot met en évidence la présence de valeurs atypiques (jusqu'à 13 enfants), alors que la majorité des individus ont entre 0 et 2 enfants. L'histogramme montre une forte concentration sur la valeur 0, suivie de loin par 1 et 2 enfants. Cette distribution très asymétrique, fortement biaisée à droite, reflète une majorité de clients sans enfants ou avec une famille restreinte.

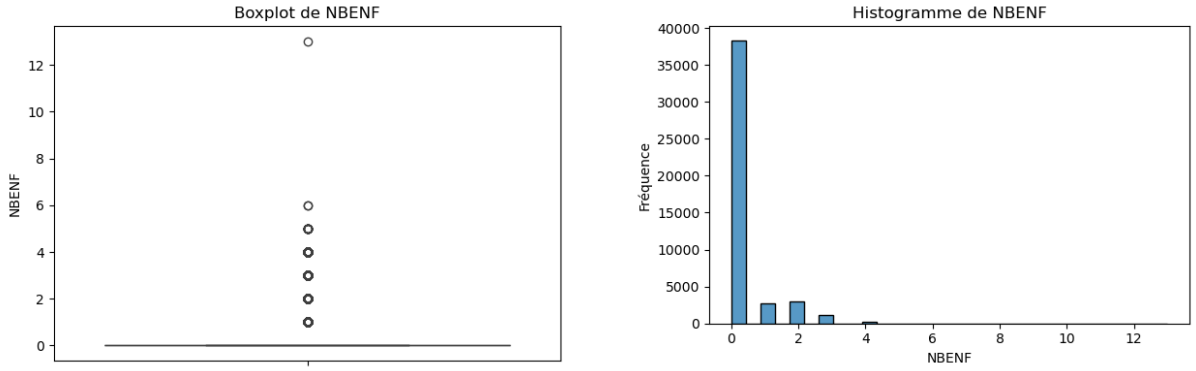

Figure 4. Distribution et boxplot de la variable `NBENF`

3.2.4 CDSITFAM : Situation familiale

Définition. La variable `CDSITFAM` représente la situation familiale du client à l'aide de codes alphabétiques. Même si les libellés exacts ne sont pas fournis, on peut supposer que certaines lettres renvoient à des catégories courantes comme « Célibataire », « Marié », « Divorcé », etc. Cette information est pertinente dans un contexte bancaire, car elle peut influencer la stabilité financière ou les décisions de fidélisation.

Traitements appliqués. Les modalités de `CDSITFAM` étant exprimées sous forme alphabétique, un encodage numérique a été appliqué afin de rendre la variable exploitable par les algorithmes de classification. Étant donné l'absence d'un ordre logique entre les catégories, un encodage arbitraire mais stable a été mis en place via une table de correspondance explicite (mapping). Ce codage a été injecté manuellement, en se basant sur les valeurs observées dans les données.

Visualisation et analyse. Le boxplot montre une concentration sur les modalités les plus basses de l'encodage (entre 1 et 4), ce qui reflète une forte présence des situations familiales les plus courantes. Les modalités plus élevées apparaissent comme valeurs atypiques, mais restent valides.

L'histogramme met en évidence un déséquilibre marqué : la majorité des clients sont regroupés dans les modalités A (probablement « Célibataire »), B (« En union libre ou concubinage ») et M (« Marié »), codées respectivement 1, 2 et 3. Ces catégories

couvrent à elles seules une très grande part de la population. En revanche, d'autres modalités comme P, F ou G sont très peu fréquentes, ce qui suggère une distribution déséquilibrée et potentiellement utile pour la segmentation des profils clients.

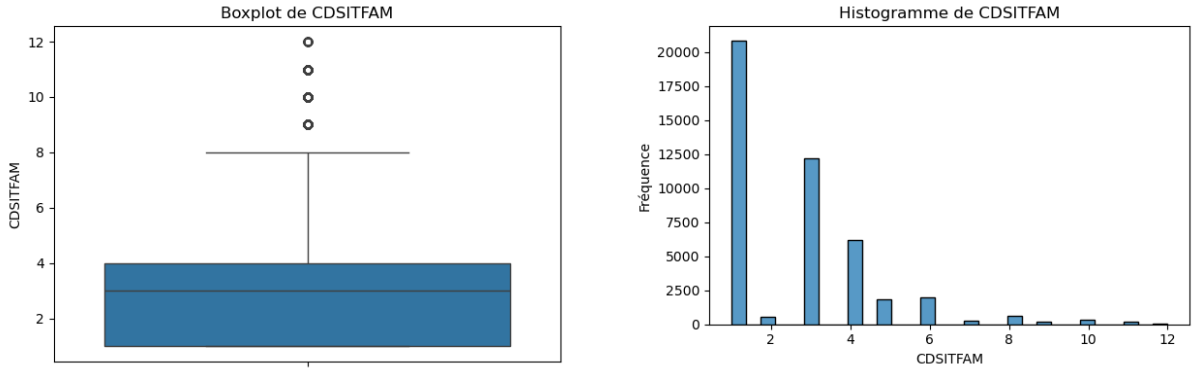

Figure 5. Distribution et boxplot de la variable `CDSITFAM`

CDMOTDEM

Définition Cette variable indique le motif de départ pour les clients démissionnaires. Elle est vide (non définie) pour les clients encore actifs, ce qui est logique puisque ces derniers n'ont pas quitté la banque.

Analyse des valeurs L'analyse montre que cette variable contient un nombre important de valeurs manquantes (14195 en total), qui correspondent aux clients toujours actifs. Les quatre modalités présentes sont :

- `DV` : 24 092 occurrences,
- `DA` : 5 166 occurrences,
- `RA` : 1 565 occurrences,
- `DC` : 336 occurrences.

Les valeurs manquantes (`NaN`, 14 195 cas) sont donc interprétées comme une **absence de démission** plutôt qu'un problème de qualité des données.

```
--- Attribut : CDMOTDEM ---
Type : object
Valeurs uniques : 4

> Valeurs les plus fréquentes :
CDMOTDEM
DV     24092
NaN    14195
DA      5166
RA      1565
DC       336
Name: count, dtype: int64
------------------------------------------------
```

Figure 6. Modèle en étoile

Décision La variable a été conservée et encodée à l'aide d'un **OneHotEncoding**. Chaque modalité a été transformée en une colonne binaire indépendante. Les valeurs manquantes ont été automatiquement traitées comme une catégorie à part entière par l'encodeur, ce qui permet de conserver l'information "non démissionnaire" sans introduire de biais.

3.2.5 CDCATCL : Catégorie de client

Définition. La variable `CDCATCL` indique la catégorie à laquelle appartient chaque client, selon une classification interne de la banque. Elle est fournie sous forme numérique, mais ses valeurs ne sont pas continues et leur signification exacte n'est pas précisée. Elle reste néanmoins utile pour distinguer différents profils d'usagers.

Traitements appliqués. Bien que `CDCATCL` soit déjà codée numériquement, ses modalités étaient dispersées (par exemple : 10, 21, etc.) et fortement déséquilibrées. Un encodage ordinal a été appliqué à l'aide de la classe `OrdinalEncoder`, afin de transformer ces valeurs en entiers consécutifs (0, 1, 2, ...), tout en conservant une structure exploitable par les modèles.

Visualisation et analyse. Le boxplot montre une forte concentration des observations dans les premières catégories, avec une série de valeurs plus élevées considérées comme atypiques (outliers). Cela reflète un déséquilibre important entre les différentes catégories de clients.

L'histogramme confirme ce constat : les catégories 0 et 1 (issues de l'encodage) dominent très largement l'échantillon, tandis que les autres classes sont marginales. Cette distribution déséquilibrée pourrait introduire un biais dans certains modèles, mais elle est aussi révélatrice de la structure commerciale de la banque.

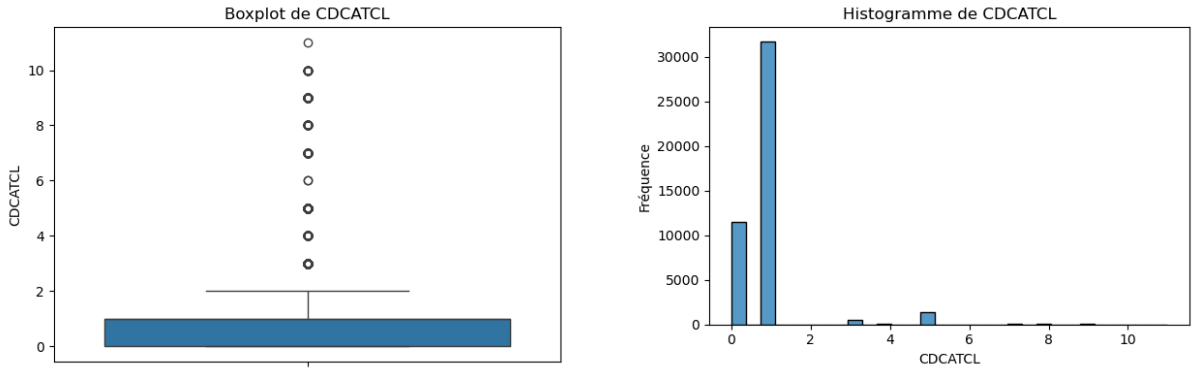

Figure 7. Distribution et boxplot de la variable `CDCATCL`

3.2.6 CDTMT : Statut du sociétaire

Définition. La variable `CDTMT` indique le statut du client vis-à-vis de la banque. Elle est exprimée sous forme numérique discrète, chaque valeur représentant probablement un type de lien ou d'adhésion (par exemple : client standard, sociétaire, partenaire, etc.).

Traitements appliqués. Aucun traitement particulier n'a été appliqué à cette variable. Les valeurs ont été conservées telles quelles, sans transformation ni encodage supplémentaire.

Visualisation et analyse. Le boxplot révèle une forte concentration sur la valeur 0, avec quelques modalités plus élevées considérées comme atypiques. L'histogramme confirme ce déséquilibre : la majorité des clients appartiennent à la modalité 0, suivis à distance par ceux en modalité 2. Les modalités plus rares (4, 6) représentent une part négligeable de l'échantillon.

Cette répartition reflète une structure client fortement dominée par un seul type de statut.

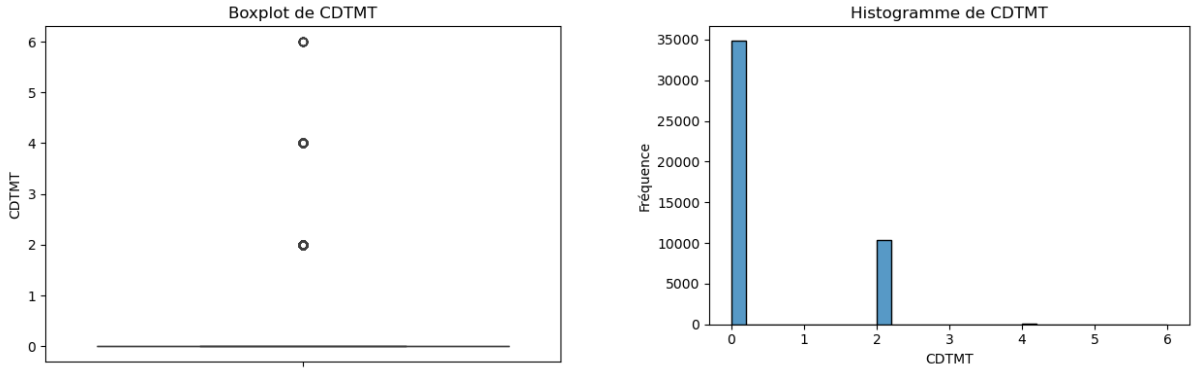

Figure 8. Distribution et boxplot de la variable `CDTMT`

3.2.7 RANGADH : Tranche d'ancienneté

Définition. La variable `RANGADH` représente la tranche d'ancienneté d'un client dans la banque. Elle est issue de la discrétisation de la variable `ADH` (durée d'adhésion en années) et permet de regrouper les clients selon des intervalles cohérents d'expérience ou de fidélité.

Traitements appliqués. Plusieurs valeurs textuelles arbitraires ont été détectées dans cette variable (par exemple : `'na'`, `'null'`, `'none'`). Ces valeurs ont été considérées comme manquantes, puis directement remplacées par la valeur la plus fréquente (mode) observée dans la colonne. Ensuite, l'ensemble de la variable a été converti au format numérique `float` pour garantir sa compatibilité avec les modèles d'apprentissage.

Visualisation et analyse. Le boxplot montre une répartition relativement symétrique, avec une médiane autour de la modalité 3 et des valeurs allant de 1 à 9 sans outliers apparents. Cela indique une bonne couverture de l'ensemble des tranches d'ancienneté.

L'histogramme révèle une forte présence des clients dans les tranches basses (1 à 3), avec un pic à la modalité 2. On observe ensuite une répartition plus étalée jusqu'à la modalité 6, et une chute nette après la modalité 7. Cette distribution traduit une population majoritairement récente ou moyenne en ancienneté, ce qui peut être un facteur explicatif du comportement client.

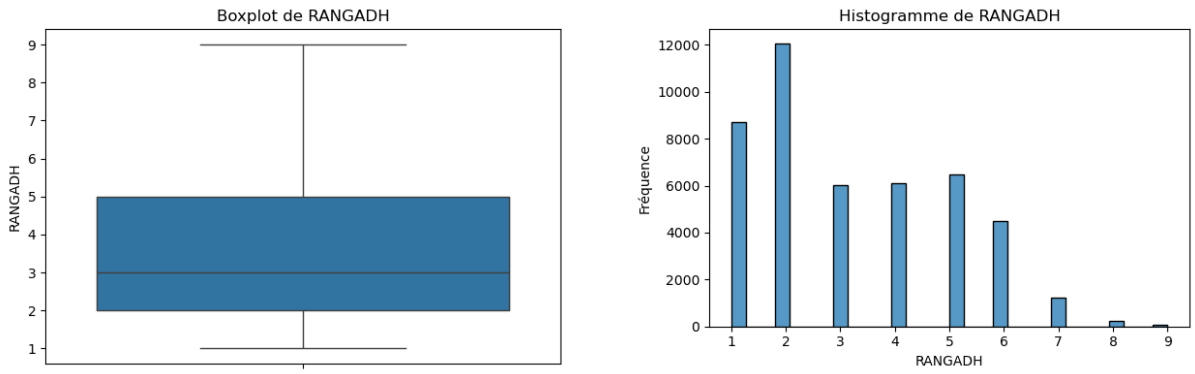

Figure 9. Distribution et boxplot de la variable `RANGADH`

3.2.8 RANGAGEDEM : Tranche d'âge à la démission

Définition. La variable `RANGAGEDEM` représente la tranche d'âge dans laquelle se trouvait le client au moment de sa démission. Elle est dérivée de la variable continue `AGEDEM`, et permet une catégorisation utile pour la modélisation.

Traitements appliqués. Les données de cette variable présentaient un format mixte (indice + étiquette texte). Plusieurs étapes de nettoyage ont été nécessaires :

- suppression des libellés textuels superflus (tranches d'âge en texte),
- transformation en minuscules et suppression des espaces inutiles,
- conversion des lettres spéciales : `a` remplacé par 10, `b` par 11,
- conversion finale de toutes les valeurs en entiers via `pandas.to_numeric()`.

Visualisation et analyse. Le boxplot montre une répartition assez équilibrée entre les différentes tranches, avec une médiane proche de 4, ce qui indique que la majorité des démissions surviennent entre les tranches d'âge intermédiaires.

L'histogramme met en évidence une anomalie sur la modalité 0, probablement liée à des erreurs ou valeurs par défaut présentes dans les données brutes. En dehors de cette modalité, la distribution est relativement homogène entre les tranches 2 et 10, ce qui témoigne d'une variété d'âges au moment de la démission.

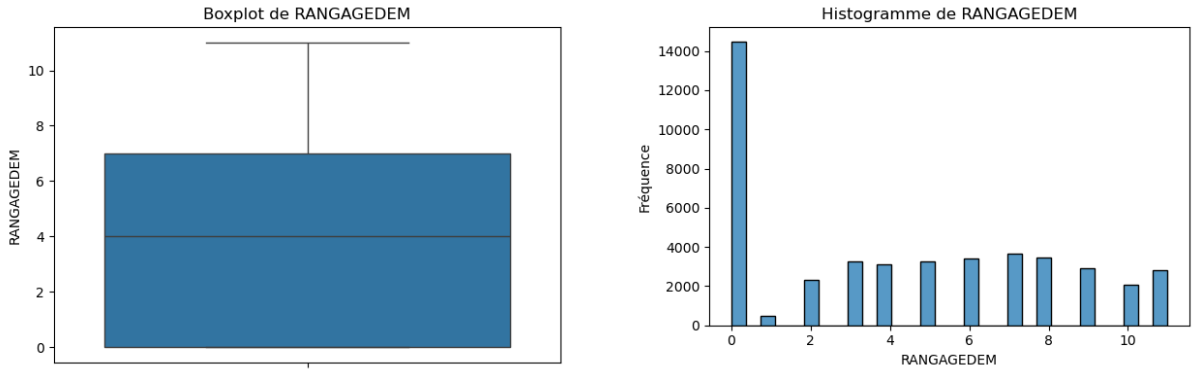

Figure 10. Distribution et boxplot de la variable `RANGAGEDEM`

3.2.9 RANGAGEAD : Tranche d'âge à l'adhésion

Définition. La variable `RANGAGEAD` représente la tranche d'âge du client au moment de son adhésion à la banque. Elle est dérivée de la variable continue `AGEAD`, et permet de regrouper les clients selon leur âge d'entrée, ce qui peut influencer leur comportement à long terme.

Traitements appliqués. La variable initiale contenait des chaînes mélangées (index + texte). Le traitement a consisté à :

- supprimer les espaces superflus et uniformiser les chaînes en minuscules,
- convertir les tranches en valeurs numériques discrètes,
- imputer les valeurs manquantes par la modalité la plus fréquente (mode).

Visualisation et analyse. Le boxplot indique une distribution équilibrée avec une médiane située autour de la modalité 4, ce qui correspond à une tranche d'âge intermédiaire au moment de l'adhésion. Aucune valeur atypique n'est observée.

L'histogramme révèle une légère asymétrie : la majorité des clients ont adhéré dans les premières tranches (2 à 4), correspondant probablement à une population jeune ou active. Les tranches supérieures (6 à 8) sont moins fréquentes, traduisant une baisse progressive d'adhésion avec l'âge.

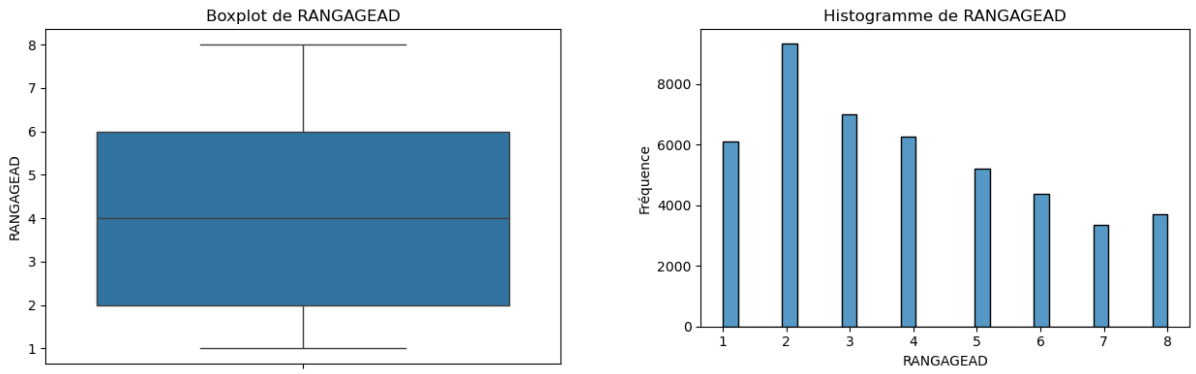

Figure 11. Distribution et boxplot de la variable `RANGAGEAD`

3.3 Correspondance des tranches ordinales

Table 1. Tranches d'âge à la démission (`RANGAGEDEM`)

Valeur	Tranche d'âge
1	19-25 ans
2	26-30 ans
3	31-35 ans
4	36-40 ans
5	41-45 ans
6	46-50 ans
7	51-55 ans
8	56-60 ans
9	61-65 ans
a	66-70 ans
b	71 ans et plus

Table 2. Tranches de durée d'adhésion (`RANGADH`)

Valeur	Durée
1	1 à 4 ans
2	5 à 9 ans
3	10 à 14 ans
4	15 à 19 ans
5	20 à 24 ans
6	25 à 29 ans
7	30 à 34 ans
8	35 à 39 ans
9	40 ans et plus

RANGAGEAD

Val.	Tranche d'âge
1	19–25
2	26–30
3	31–35
4	36–40
5	41–45
6	46–50
7	51–55
8	56+

CDCATCL

Code	Original
0	10
1	21
2	25
3	23
4	24
5	50
6	40
7	61
8	22
9	82
10	32
11	98

CDSITFAM

Lettre	Code
A	1
B	2
M	3
C	4
D	5
U	6
S	7
V	8
E	9
G	10
P	11
F	12

3.4 Analyse des corrélations entre attributs

Afin d'étudier les relations linéaires entre les variables du jeu de données, une matrice de corrélation a été calculée à partir des variables numériques et ordinales. Cette approche permet de repérer d'éventuelles redondances, mais aussi d'identifier les variables les plus informatives vis-à-vis de la variable cible `DEMISSIONNAIRE`.

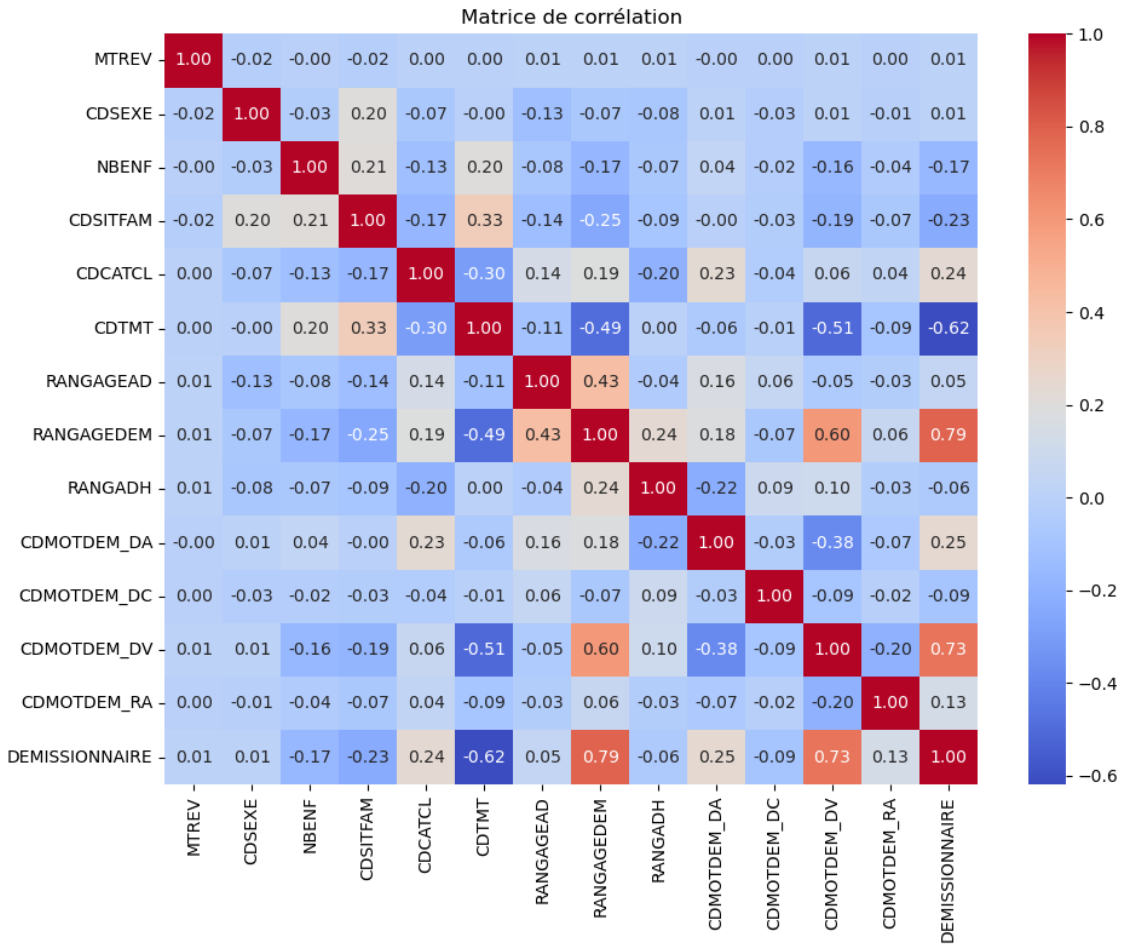

Figure 12. Matrice de corrélation des variables numériques et ordinales

Analyse des corrélations

Corrélations avec la variable cible `DEMISSIONNAIRE`

- **RANGAGEDEM** présente une très forte corrélation positive ($+0.79$) avec `DEMISSIONNAIRE`. Cela signifie que plus un client est âgé au moment de sa démission, plus il est probable qu'il fasse effectivement partie des démissionnaires.
- **CDMOTDEM_DV** (motif de départ « DV ») est aussi fortement corrélée ($+0.73$) à la variable cible. Cela suggère que ce motif est typiquement associé aux démissionnaires.
- **CDMOTDEM_DA** (motif « DA ») a une corrélation plus modérée mais toujours positive ($+0.25$), indiquant également un lien avec les départs.
- **CDMOTDEM_DC** (clients non démissionnaires) affiche une très forte corrélation négative (-0.99) avec `DEMISSIONNAIRE`, ce qui est attendu puisqu'il identifie précisément les clients restés actifs.
- **CDTMT** présente une forte corrélation négative (-0.62), ce qui indique que certains statuts de sociétaires sont liés à une meilleure fidélité.
- **CDCATCL** affiche une corrélation positive modérée ($+0.24$) : certaines catégories de clients semblent plus enclines à quitter la banque.
- **CDSITFAM** (-0.23) et **NBENF** (-0.17) montrent que la stabilité familiale (couple, enfants) pourrait être un facteur de rétention.
- Les autres variables telles que **CDSEXE**, **RANGAGEAD**, **RANGADH**, **CDMOTDEM_RA**, **CDMOTDEM_DC** présentent des corrélations faibles (proches de zéro) et semblent avoir peu d'influence directe sur la démission.

Remarque importante *: les variables présentant une corrélation supérieure à* $|\pm 0.70|$ *avec la variable cible peuvent introduire un fort biais dans les modèles de prédiction (sur-apprentissage ou perte de généralisation). Il est recommandé de les exclure ou de les encoder avec précaution. Cela concerne ici* ***`RANGAGEDEM`****,* ***`CDMOTDEM_DV`*** *et* ***`CDMOTDEM_DC`****.*

Corrélations entre variables explicatives

- **RANGAGEDEM** et **RANGAGEAD** sont corrélés positivement ($+0.43$), ce qui est logique puisqu'ils décrivent tous deux des aspects liés à l'âge du client.
- **CDMOTDEM_DV** est fortement corrélée à **CDMOTDEM_DA** ($+0.60$), suggérant que ces motifs peuvent être associés à des profils similaires.
- **CDMOTDEM_DC** (non démissionnaires) est très négativement corrélée à **RANGAGEDEM** (-0.78), **CDMOTDEM_DV** (-0.72) et **CDMOTDEM_DA** (-0.24), confirmant que ces variables sont liées à des profils quittant la banque.
- **RANGAGEDEM** montre également une corrélation modérée avec **CDCATCL** ($+0.19$), ce qui peut refléter un lien indirect entre âge et type de clientèle.
- Les relations entre **CDSEXE**, **NBENF**, **CDSITFAM** sont globalement faibles (entre -0.17 et $+0.21$), traduisant une relative indépendance entre ces variables.

Conclusion. L'analyse de la matrice de corrélation a permis d'identifier plusieurs variables fortement liées à la variable cible `DEMISSIONNAIRE`, notamment `RANGAGEDEM`, `CDMOTDEM_DV`, `CDMOTDEM_DC` et `CDTMT`. Ces attributs devront être manipulés avec précaution, car une trop forte corrélation peut induire un sur-apprentissage si le modèle s'appuie uniquement sur ces signaux.

Cependant, il est important de ne pas négliger les variables présentant une corrélation faible avec la cible. Des attributs comme `CDSEXE`, `NBENF`, `RANGADH`, `CDSITFAM` ou encore `RANGAGEAD` peuvent, lorsqu'ils sont combinés entre eux, contribuer à améliorer la performance d'un modèle d'apprentissage. En effet, certaines relations complexes ou non linéaires ne sont pas détectées par une simple corrélation linéaire, mais peuvent être capturées par des algorithmes comme les arbres de décision, les forêts aléatoires ou les réseaux de neurones.

Ainsi, toutes les variables nettoyées et préparées seront initialement conservées pour la phase de modélisation, avant de procéder à une éventuelle sélection automatique des plus pertinentes.

4 Questionnements

4.1 Pourquoi fusionner les deux tables ?

Les fichiers `table1.csv` et `table2.csv` contiennent respectivement les données de clients démissionnaires (historiques) et un échantillon de clients actuels, parmi lesquels on trouve à la fois des démissionnaires et des clients toujours actifs. Afin de constituer un jeu de données exploitable pour une tâche de classification supervisée, il a été nécessaire de fusionner ces deux tables.

Cette fusion permet d'avoir une base cohérente regroupant l'ensemble des profils disponibles. Elle facilite la comparaison entre les différents types de clients et la détection de patterns communs. Pour distinguer les deux groupes dans le jeu final, une variable cible binaire `DEMISSIONNAIRE` a été créée : elle prend la valeur 1 pour les clients ayant quitté la banque (identifiés par une date de démission réelle) et 0 pour les clients encore actifs (indiqués par une date par défaut comme `31/12/1900`).

Résultat : un tableau unique, harmonisé et annoté, prêt pour les phases d'analyse, d'apprentissage automatique et de prédiction du risque de départ.

4.2 Comment gérer les valeurs manquantes ou aberrantes ?

Les deux fichiers `table1.csv` et `table2.csv` ne partagent pas exactement les mêmes attributs. Lors de la fusion, certaines colonnes se sont retrouvées partiellement vides, notamment pour les clients actifs qui ne disposent pas, par exemple, d'une date de démission ou d'une tranche d'âge à la démission. Cela a généré un nombre important de valeurs nulles.

Pour gérer ces cas, un traitement spécifique a été mis en place :

Certaines variables manquantes ont été recalculées à partir d'autres attributs. Par exemple, les variables `RANGAGEAD`, `RANGAGEDEM` et `RANGADH` ont été déterminées à

partir des dates d'adhésion, de naissance et, si disponible, de démission. De même, la durée d'adhésion `ADH` a été calculée en soustrayant les dates pertinentes.

Pour les attributs catégoriels présentant des valeurs manquantes, j'ai appliqué une imputation par la valeur la plus fréquente (mode)

Une attention particulière a été portée aux valeurs nulles attendues, comme l'absence de motif de démission (`CDMOTDEM`) chez les clients actifs. Ces cas n'ont pas été imputés pour ne pas biaiser le modèle.

Enfin, certains attributs trop incomplets ou peu informatifs ont été supprimés afin de ne pas nuire à la qualité du jeu final.

Résultat Un jeu de données plus propre, structuré et cohérent, prêt à être utilisé pour la modélisation. Le taux de valeurs manquantes a été réduit au strict nécessaire, sans perte d'information pertinente, tout en maintenant une logique métier respectée.

4.3 Pourquoi certaines variables ont-elles été supprimées ?

Certaines colonnes ont été utilisées uniquement à des fins de transformation, de vérification ou n'étaient pas pertinentes pour la modélisation finale. Elles ont donc été supprimées afin d'obtenir un jeu de données plus cohérent, sans redondance ni bruit.

En particulier, les variables suivantes ont été retirées :

- `AGEAD`, `AGEDEM`, `ADH`, `DTNAIS`, `DTADH`, `DTDEM`, `ANNEEDEM`, `RANGDEM` : utilisées pour reconstruire les tranches ordinales (`RANGAGEAD`, `RANGAGEDEM`, `RANGADH`) ou pour calculer la durée d'adhésion.
  `CDDEM` : trop corrélées à la cible (risque de fuite d'information).
- `BPADH` : attribut absent ou non exploitable dans la majorité des cas.
- `ID` : identifiant unique sans valeur explicative.

Une attention particulière a été portée aux attributs numériques et temporels, dont les informations ont été condensées dans des versions ordinales ou catégorielles plus robustes pour les modèles.

Résultat Un jeu de données allégé, plus homogène et orienté vers l'apprentissage supervisé, avec uniquement les variables pertinentes et exploitables pour la modélisation prédictive

4.4 Quel type d'encodage utiliser pour les variables catégorielles ?

Le choix d'encodage dépend de la nature des variables :

OrdinalEncoder a été appliqué sur des variables discrètes avec un ordre implicite ou arbitraire mais stable (comme `CDCATCL`, `CDSITFAM`), et les attributs RANG

OneHotEncoder a été préféré pour les variables sans ordre logique, comme `CDMOTDEM`, afin d'éviter d'introduire un biais de rang.

Résultat : toutes les variables catégorielles ont été converties en format numérique exploitable par les algorithmes de machine learning.

4.5 Pourquoi normaliser ou standardiser les variables ?

Les modèles d'intelligence artificielle et d'apprentissage automatique sont généralement plus performants et plus stables lorsque les données sont exprimées sous forme numérique et sur des échelles similaires.

En effet, certains algorithmes (comme les k-plus proches voisins, SVM, ou les réseaux de neurones) sont sensibles à l'échelle des variables. Si une variable contient des valeurs très grandes (ex. : `MTREV`), elle peut dominer le calcul des distances ou gradients et fausser l'apprentissage, même si elle n'est pas la plus informative. De plus, les modèles ont souvent une meilleure capacité à détecter des patterns sur de petites valeurs centrées autour de zéro.

Par ailleurs, les attributs textuels ou catégoriels doivent être transformés en valeurs numériques pour être interprétables par les modèles. Sans cette transformation, les algorithmes ne peuvent pas traiter ces variables.

Résultat Une base de données homogène, numérique, et normalisée, facilitant la convergence des algorithmes, améliorant leur stabilité et évitant que certaines variables n'écrasent l'influence des autres lors de l'apprentissage.

4.6 Est-ce que je dois appliquer l'ACP ?

L'Analyse en Composantes Principales (ACP) est une technique de réduction de dimension utilisée lorsqu'il y a beaucoup de variables et des corrélations élevées entre elles. Elle permet de simplifier le modèle en combinant les variables redondantes.

Dans notre cas, l'analyse de corrélation montre peu de redondance, à l'exception des variables liées à l'âge et à la durée d'adhésion. De plus, l'ACP rend les variables moins interprétables, ce qui n'est pas souhaitable ici.

Conclusion : l'ACP n'a pas été retenue, mais elle pourrait être testée ultérieurement dans une approche comparative.

5 Analyse prédictive et création des modèles

Après avoir nettoyé, transformé et préparé les données, l'étape suivante consiste à entraîner des modèles de classification supervisée pour prédire le départ d'un client.

L'objectif est de construire un modèle de classification binaire, produisant un **score de probabilité de démission** pour chaque client. Cette approche vise à anticiper les départs et à fournir un outil d'aide à la décision pour les stratégies de fidélisation.

Conformément au sujet, quatre algorithmes de classification sont évalués :

- **SVM (Séparateur à Vaste Marge)**,
- **k plus proches voisins (kNN)**,
- **Classificateur bayésien naïf (Naive Bayes)**,
- ainsi qu'un **modèle libre** choisi pour sa robustesse : la **régression logistique**.

Tous les modèles sont entraînés et testés sur le même découpage des données, afin d'assurer une comparaison équitable. En revanche, certains prétraitements

(comme l'encodage ou la normalisation) peuvent différer selon les exigences spécifiques de chaque algorithme.

L'analyse porte à la fois sur la **performance prédictive** des modèles (via des métriques comme la précision, le rappel, le F1-score ou l'AUC ROC), mais aussi sur leur **interprétabilité**, afin d'identifier les attributs les plus influents dans le processus de démission.

5.1 Découpage des données

Lors de l'exploration des données, un fort déséquilibre entre les classes a été observé : environ 68% de clients démissionnaires contre seulement 32% de non-démissionnaires. Ce déséquilibre peut biaiser les modèles de classification, les poussant à favoriser la classe majoritaire.

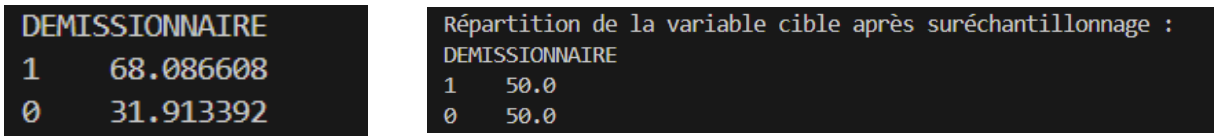

Figure 13. Équilibrage de la variable cible : avant (déséquilibré) et après (équilibré) suréchantillonnage.

Pour corriger ce problème, une méthode de fouille de données a été appliquée : le **suréchantillonnage aléatoire** via `RandomOverSampler` de la bibliothèque `imblearn`. Cette technique permet de rééquilibrer les classes en dupliquant aléatoirement des instances de la classe minoritaire, menant à une répartition équilibrée de 50% / 50%.

Ensuite, les données ont été divisées en trois ensembles :

- **Apprentissage** (60%) : utilisé pour entraîner les modèles, avec données rééquilibrées.
- **Validation** (20%) : pour le réglage des hyperparamètres et la comparaison des modèles.
- **Test** (20%) : pour l'évaluation finale et la généralisation du modèle retenu.

Ce découpage a été réalisé avec `train_test_split` (`random_state=42`) pour assurer la reproductibilité.

5.2 Présentation des modèles et métriques utilisées

Conformément au sujet, quatre algorithmes de classification ont été sélectionnés :

- **Support Vector Machine (SVM)** : efficace pour les données non linéaires grâce au noyau RBF.
- **k plus proches voisins (kNN)** : méthode intuitive basée sur la proximité, sensible à la dimension des données.
- **Naive Bayes catégoriel** : simple, rapide, adapté aux variables discrètes.
- **Régression Logistique** : modèle linéaire robuste, choisi pour sa bonne interprétabilité.

Tous les modèles sont entraînés sur le même jeu de données équilibré, et validés sur le même ensemble pour garantir une comparaison équitable. Un seuil de classification de 0.6 a été appliqué sur les probabilités pour produire la classe prédite.

Pour évaluer les performances, plusieurs métriques ont été utilisées :

- **Accuracy** : proportion de bonnes prédictions.
- **Recall** : capacité à bien identifier les démissionnaires (classe 1).
- **F1-score** : compromis entre précision et rappel, utile en cas de classes déséquilibrées.

Des matrices de confusion et des histogrammes de probabilités ont également été produits pour une analyse visuelle plus fine des résultats.

5.3 Entraînement des modèles et résultats

Chaque modèle a été entraîné sur l'ensemble d'apprentissage équilibré, puis évalué sur l'ensemble de validation et enfin testé sur l'ensemble de test. Les résultats sont présentés ci-dessous pour chaque modèle.

5.3.1 Régression Logistique

Le modèle de régression logistique a été entraîné sur les données équilibrées, puis évalué sur les ensembles de validation et de test. Malgré plusieurs essais sur les hyperparamètres, notamment le paramètre de régularisation C et le solveur (`liblinear`), les performances sont restées similaires. Cela montre que le modèle atteint rapidement ses limites sur ce jeu de données. Il offre des résultats acceptables, mais reste moins performant que d'autres méthodes plus flexibles.

1. Performance globale (validation) :

- **Accuracy** : 84.33%
- **F1 Score** : 89.15%
- **Recall** : 95.28%

2. Matrice de confusion (test) :

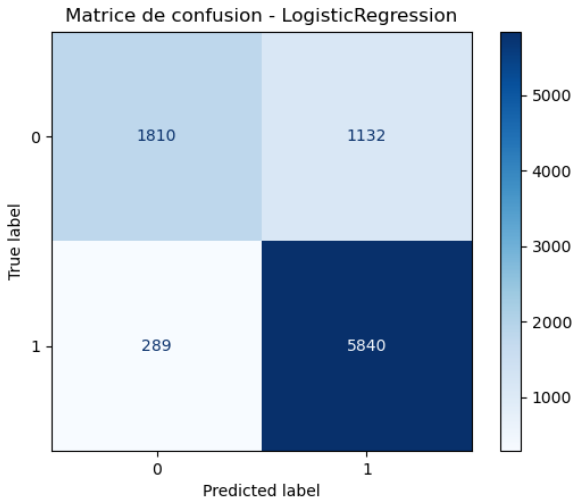

Figure 14. Matrice de confusion — Régression Logistique (test)

On constate que 1153 clients non démissionnaires ont été classés à tort comme démissionnaires (faux positifs), ce qui affecte la précision globale. Cependant, seule-

ment 324 démissionnaires ont été mal détectés (faux négatifs), ce qui confirme le bon rappel du modèle.

3. Distribution des probabilités (test) :

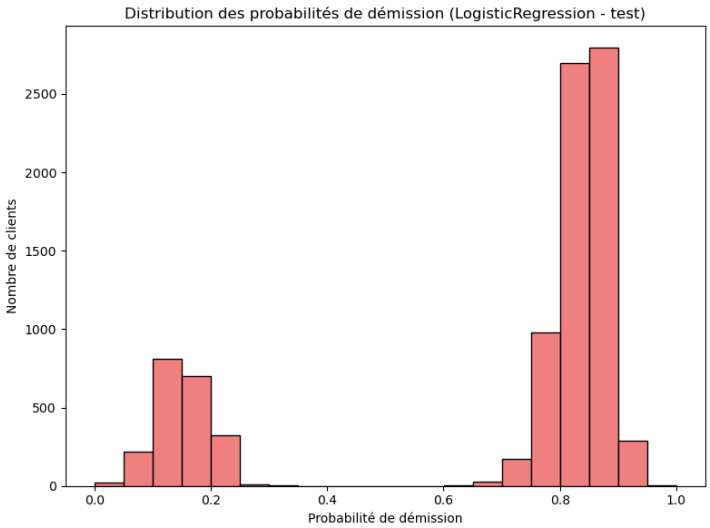

Figure 15. Distribution des probabilités de démission — Régression Logistique (test)

La distribution montre une séparation claire entre les classes avec deux pics bien distincts. Le modèle est donc globalement confiant dans ses prédictions.

4. Visualisation PCA 3D (validation) :

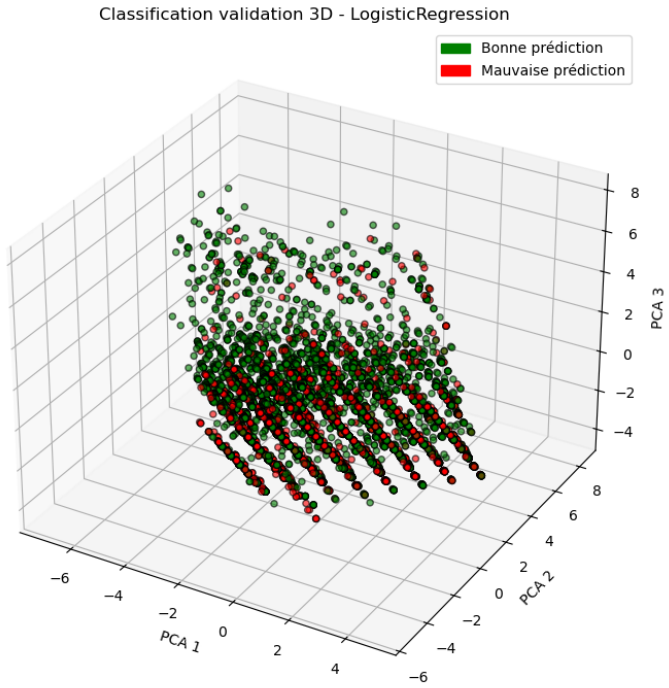

Figure 16. Classification 3D sur validation — Régression Logistique

La projection montre que les erreurs (en rouge) sont principalement concentrées dans des zones intermédiaires, ce qui est cohérent avec une séparation linéaire.

Conclusion :

Les résultats obtenus montrent que, malgré un bon rappel, la régression logistique reste limitée en précision. Même en ajustant ses hyperparamètres, les performances n'ont pas significativement évolué. Cela suggère que les modèles linéaires ne sont peut-être pas les plus adaptés à la complexité et à la structure de ce jeu de données.

5.3.2 Support Vector Machine (SVM)

Le modèle SVM a été entraîné avec un noyau RBF, permettant une séparation non linéaire des classes. Contrairement à la régression logistique, ce modèle est plus flexible et s'adapte mieux à des frontières de décision complexes.

1. Performance globale (validation) :

- **Accuracy** : 92.43%
- **F1 Score** : 94.52%
- **Recall** : 96.67%

Le SVM obtient les meilleures performances globales parmi tous les modèles testés, avec un excellent compromis entre précision et rappel.

2. Matrice de confusion (test) :

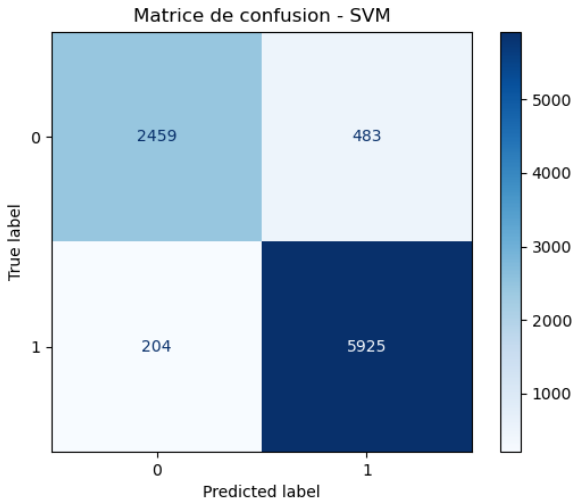

Figure 17. Matrice de confusion — SVM (test)

On observe que 483 clients non démissionnaires ont été classés à tort comme démissionnaires (faux positifs), tandis que seulement 204 démissionnaires n'ont pas été détectés (faux négatifs), ce qui confirme la fiabilité du modèle.

3. Distribution des probabilités (test) :

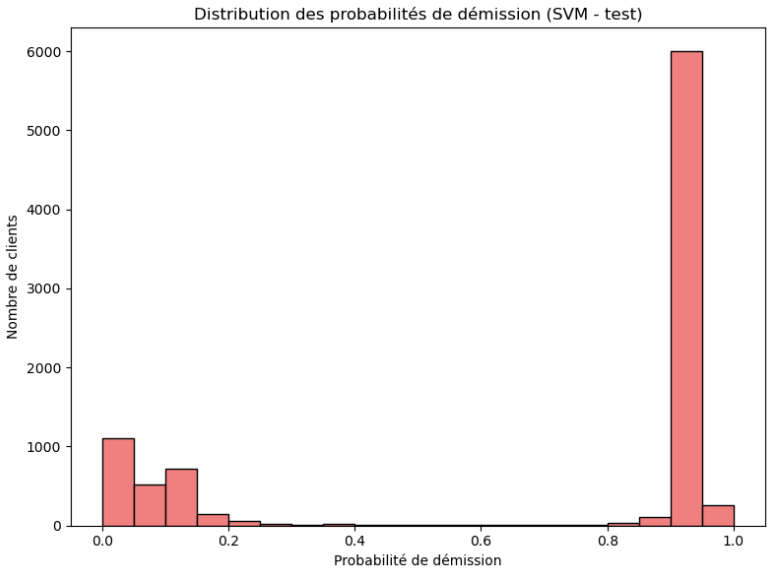

Figure 18. Distribution des probabilités de démission — SVM (test)

Le modèle montre une forte confiance dans ses prédictions, avec une polarisation nette des probabilités vers 0 et 1. Cela témoigne d'une bonne séparation des classes.

4. Visualisation PCA 3D (validation) :

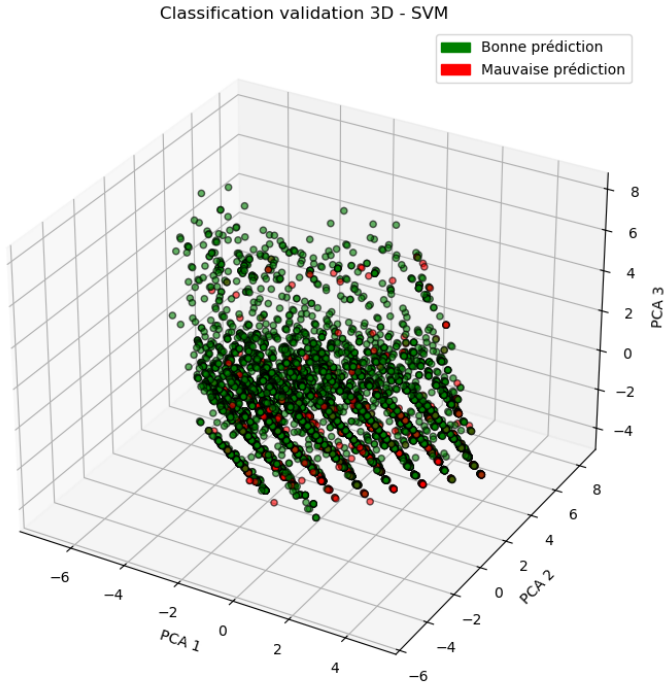

Figure 19. Classification 3D sur validation — SVM

La visualisation montre peu d'erreurs (points rouges) et une bonne séparation visuelle des classes, confirmant la robustesse du modèle SVM sur ce jeu de données.

Conclusion :

Grâce à sa capacité à modéliser des frontières non linéaires, SVM surpasse

les performances des modèles linéaires dans ce contexte. Il s'agit d'un excellent candidat pour prédire le risque de démission avec fiabilité.

5.3.3 k plus proches voisins (kNN)

Le modèle kNN a été utilisé avec $k = 10$, conformément à la recommandation du professeur de ne pas utiliser une valeur inférieure à 10. Plusieurs tests avec différentes valeurs de k ont été réalisés, mais les performances sont restées globalement similaires. Ce modèle repose sur la distance entre les points pour effectuer les prédictions, ce qui le rend simple mais sensible aux données bruitées et à la dimensionnalité.

1. Performance globale (validation) :

- **Accuracy** : 91.21%
- **F1 Score** : 93.67%
- **Recall** : 96.17%

Le modèle obtient de très bonnes performances, avec un rappel élevé et un bon équilibre entre les classes.

2. Matrice de confusion (test) :

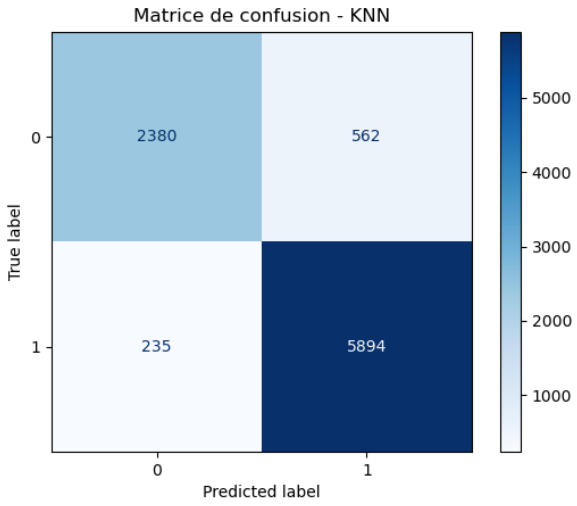

Figure 20. Matrice de confusion — kNN (test)

On observe que 562 clients non démissionnaires ont été faussement identifiés comme démissionnaires (faux positifs), tandis que 235 démissionnaires ont été manqués (faux négatifs). Cela reste globalement satisfaisant compte tenu de la nature simple du modèle.

3. Distribution des probabilités (test) :

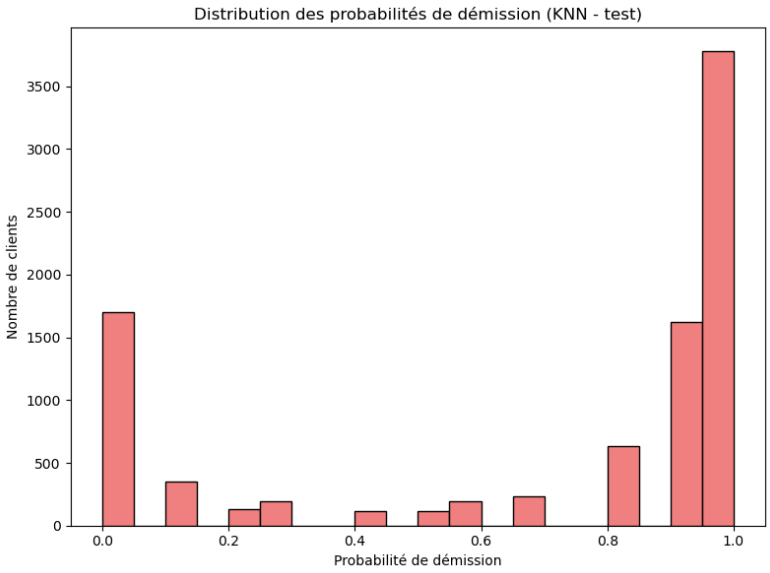

Figure 21. Distribution des probabilités de démission — kNN (test)

La distribution présente une concentration importante autour des extrémités (0 et 1), signe que le modèle est souvent très confiant dans ses prédictions, mais montre aussi quelques cas intermédiaires plus incertains.

4. Visualisation PCA 3D (validation) :

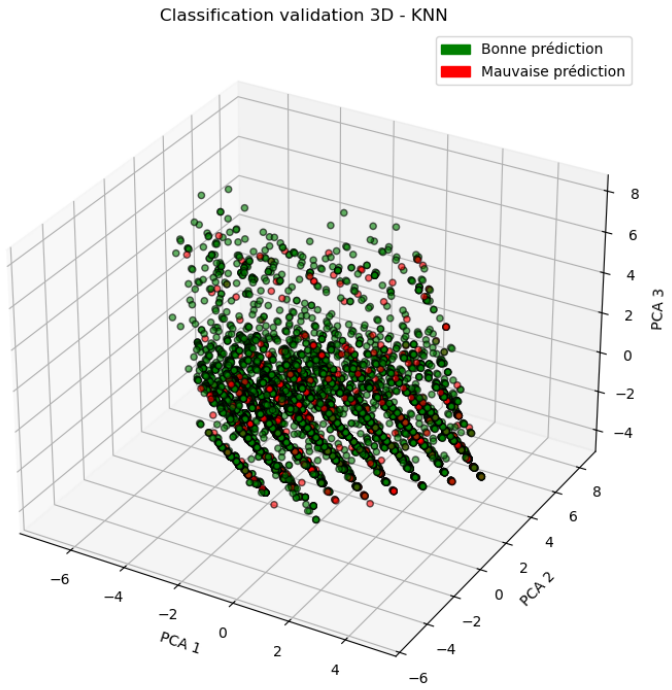

Figure 22. Classification 3D sur validation — kNN

La projection en trois dimensions met en évidence un bon regroupement des prédictions correctes, bien que certaines erreurs soient visibles dans les zones de frontière, ce qui est typique pour un modèle à base de voisinage.

Conclusion :

Les performances de kNN se rapprochent de celles obtenues avec SVM, mais avec l'avantage notable d'un temps d'entraînement beaucoup plus rapide. Ce compromis entre rapidité et performance en fait une alternative intéressante, notamment lorsque la rapidité d'exécution est cruciale.

5.3.4 Classificateur Bayésien naïf

Le modèle Naive Bayes utilisé ici est la version catégorielle (`CategoricalNB`), bien adaptée aux variables discrètes. Il repose sur l'hypothèse forte d'indépendance conditionnelle entre les attributs, ce qui simplifie le calcul mais peut être irréaliste dans certains cas.

1. Performance globale (validation) :

- **Accuracy** : 91.16%
- **F1 Score** : 93.60%
- **Recall** : 95.66%

Malgré sa simplicité, le modèle atteint des résultats très proches de ceux de SVM et kNN sur les données de validation.

2. Matrice de confusion (test) :

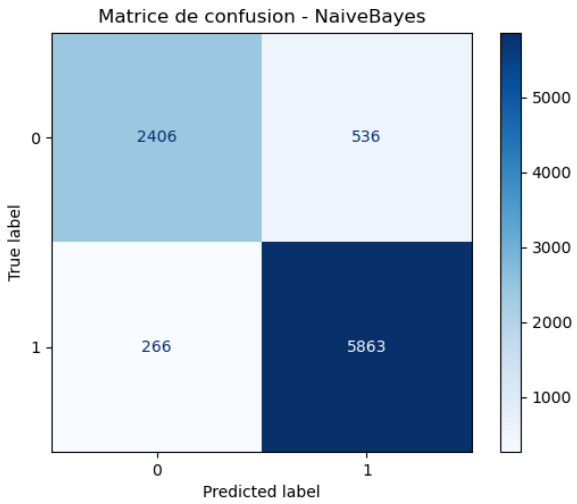

Figure 23. Matrice de confusion — Naive Bayes (test)

Le modèle a mal classé 536 non démissionnaires (faux positifs) et 266 démissionnaires (faux négatifs), ce qui reste un très bon équilibre compte tenu de la nature naïve du modèle.

3. Distribution des probabilités (test) :

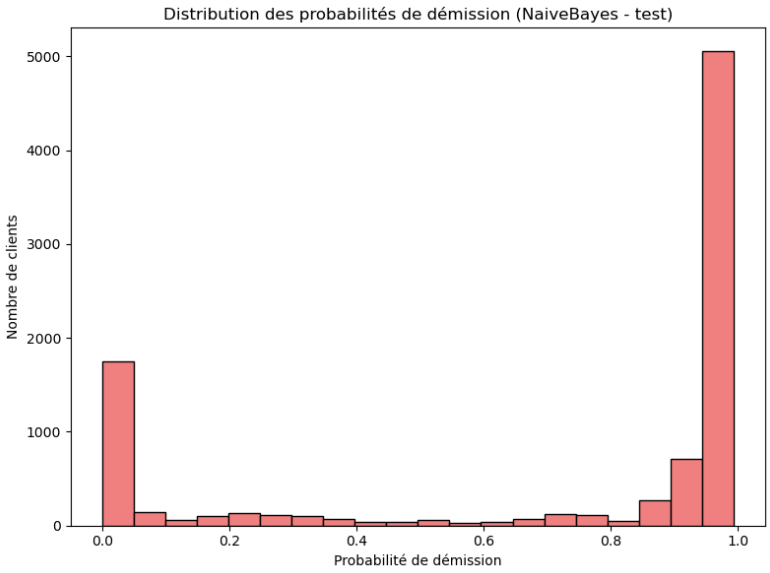

Figure 24. Distribution des probabilités de démission — Naive Bayes (test)

Les prédictions sont très polarisées, concentrées autour de 0 et 1, ce qui traduit une forte confiance du modèle, même si certaines prédictions intermédiaires persistent.

4. Visualisation PCA 3D (validation) :

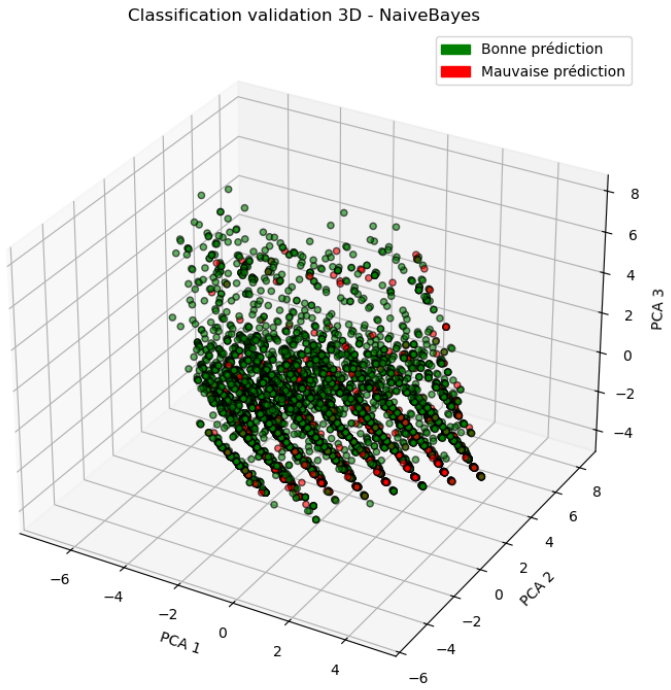

Figure 25. Classification 3D sur validation — Naive Bayes

La représentation en 3D montre un bon regroupement des prédictions correctes, malgré quelques erreurs réparties dans l'espace, signe que le modèle reste efficace malgré l'hypothèse d'indépendance.

Conclusion :

Naive Bayes se révèle étonnamment performant malgré son approche simplifiée. Il rivalise avec des modèles plus complexes tout en offrant une rapidité d'entraînement exceptionnelle, ce quiF en fait un excellent choix de base pour une première modélisation.

5.4 Interprétation et importance des attributs

Pour mieux comprendre les facteurs influençant la démission des employés, deux approches ont été utilisées pour analyser l'importance des attributs :

- **Pour les modèles linéaires (Régression Logistique)** : Les coefficients du modèle ont été analysés directement, car ils représentent l'impact de chaque variable sur la probabilité de démission.
- **Pour les modèles non linéaires (SVM, Naive Bayes, kNN)** : Une méthode de permutation (`permutation_importance`) a été utilisée, avec $n\_repeats = 10$ pour assurer la stabilité des résultats. Cette méthode mesure l'effet de la perturbation d'une variable sur les performances du modèle.

Régression Logistique :

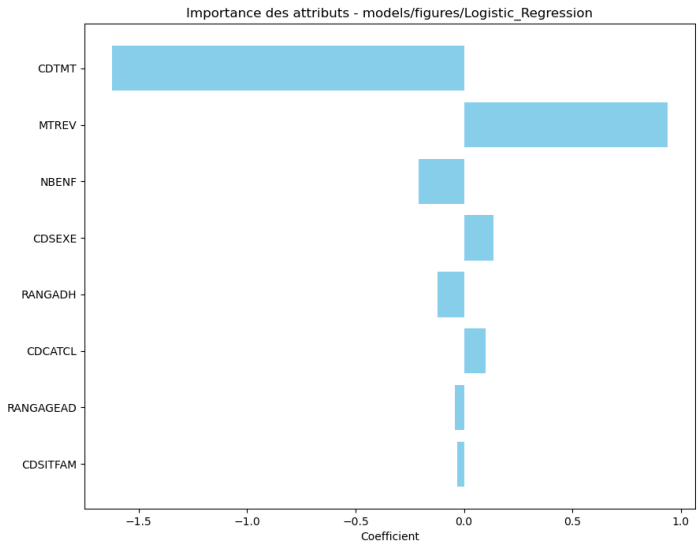

Figure 26. Importance des attributs — Régression Logistique

L'analyse des coefficients montre que l'attribut `CDTMT` (type de contrat) a une forte influence **négative** sur la probabilité de démission. Cela signifie que certaines modalités de contrat sont fortement associées à une réduction du risque de départ.

En revanche, l'attribut `MTREV` (montant de la rémunération) présente un effet positif marqué : plus le revenu mensuel est élevé, plus la probabilité de démission augmente selon le modèle. Cela peut paraître contre-intuitif, mais peut indiquer que **les profils les mieux rémunérés ont plus d'opportunités ailleurs ou plus d'exigences.**

Les autres attributs (`NBENF`, `CDSEXE`, `RANGADH`, etc.) ont des effets plus modérés, et leur contribution au modèle reste plus faible.

Il est important de noter que :

- un **coefficient positif** indique que l'augmentation de la variable augmente la probabilité de démission,
- un **coefficient négatif** signifie au contraire qu'une valeur plus élevée diminue cette probabilité.

k plus proches voisins (kNN) :

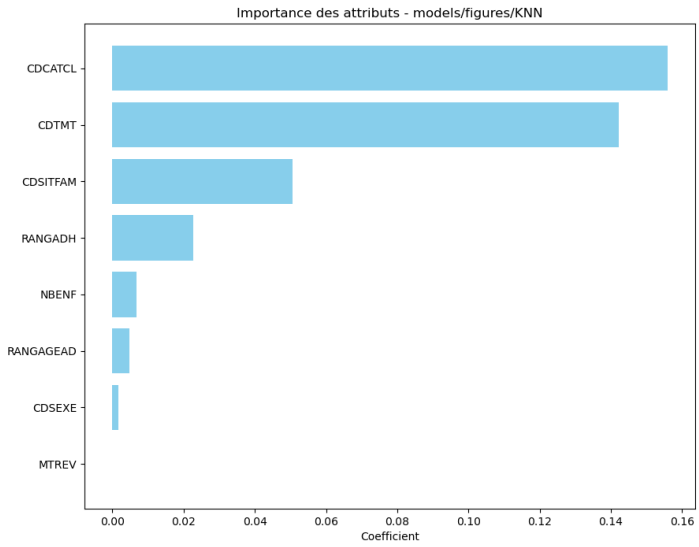

Figure 27. Importance des attributs — kNN (permutation)

L'importance des attributs a été mesurée via la méthode de permutation. On observe que les variables `CDCATCL` (catégorie contractuelle) et `CDTMT` (type de contrat) ont la plus grande influence sur les prédictions de kNN. Leur permutation entraîne une forte dégradation de la performance du modèle, ce qui montre leur rôle crucial dans la séparation des classes.

`CDSITFAM` (situation familiale) et `RANGADH` (grade hiérarchique) apparaissent également comme des facteurs significatifs, bien que dans une moindre mesure.

Les variables comme `NBENF` (nombre d'enfants), `RANGAGEAD` (tranche d'âge), ou `CDSEXE` (sexe) ont une importance marginale. Enfin, `MTREV` (montant de la rémunération) semble avoir un impact très limité selon cette méthode.

Ces résultats indiquent que kNN se base essentiellement sur des caractéristiques contractuelles et administratives pour déterminer les similarités entre individus, au détriment des variables personnelles ou financières.

Support Vector Machine (SVM) :

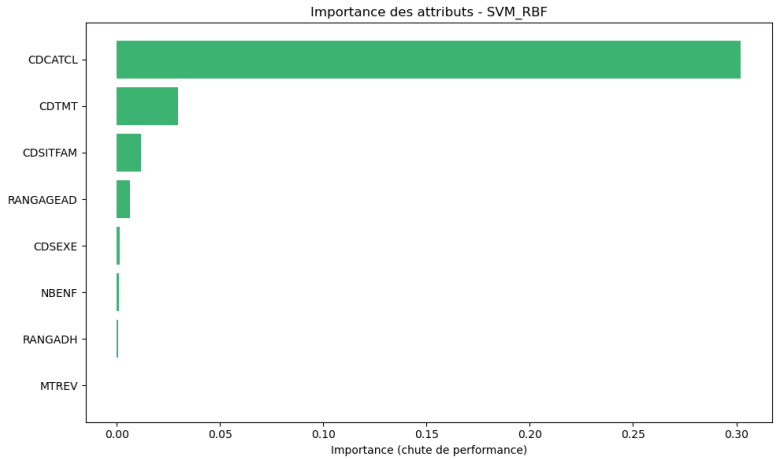

Figure 28. Importance des attributs — SVM (permutation)

Le modèle SVM (avec noyau RBF) montre une dépendance très forte à la variable `CDCATCL`, dont la permutation entraîne une chute de performance significative. Les attributs `CDTMT` (type de contrat) et `CDSITFAM` (situation familiale) suivent avec un impact plus modéré. Cela confirme que les catégories contractuelles sont essentielles pour ce modèle. D'autres variables comme `MTREV` ou `NBENF` semblent avoir peu ou pas d'influence.

Naive Bayes :

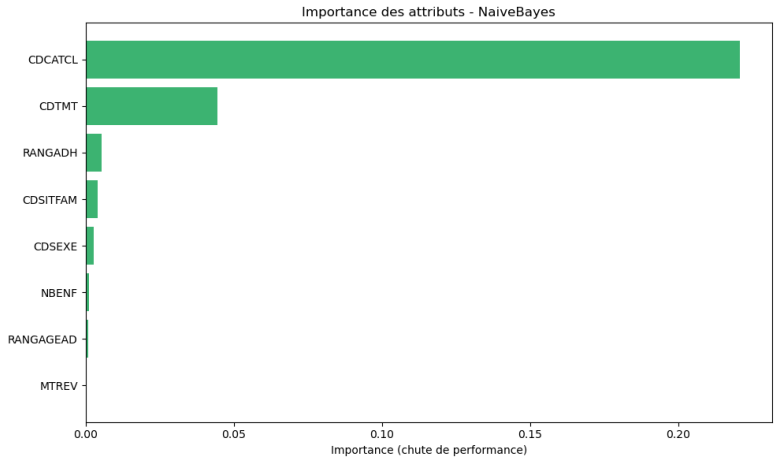

Figure 29. Importance des attributs — Naive Bayes (permutation)

Le modèle Naive Bayes montre une très forte dépendance à l'attribut `CDCATCL` (catégorie contractuelle), dont la permutation entraîne une perte de performance significative. L'attribut `CDTMT` (type de contrat) suit en seconde position, bien que son impact soit nettement inférieur.

Les autres variables (`RANGADH`, `CDSITFAM`, `CDSEXE`, etc.) ont une influence marginale, avec une importance proche de zéro. Cela est cohérent avec la nature du classificateur bayésien, qui repose sur des hypothèses d'indépendance entre les attributs, et favorise donc les variables fortement discriminantes individuellement.

On note également que des variables comme `MTREV` ou `NBENF` n'apportent presque aucune valeur informative pour ce modèle.

Conclusion :

Quel que soit le modèle utilisé, certains attributs comme `CDCATCL` (catégorie du contrat) et `CDTMT` (type de contrat) apparaissent systématiquement parmi les plus influents. Cette convergence suggère qu'ils jouent un rôle clé dans le processus de démission et doivent faire l'objet d'une attention particulière dans les stratégies de rétention.

5.5 Choix du modèle final

Le choix du modèle final repose sur un compromis entre plusieurs critères :

- **Performance prédictive** : mesurée via l'accuracy, le F1-score et le rappel.
- **Temps de traitement** : temps d'entraînement et rapidité d'inférence.
- **Interprétabilité** : capacité à comprendre et justifier les décisions du modèle.
- **Simplicité de déploiement et robustesse**.

Dans notre cas, bien que le modèle **SVM** ait offert les meilleures performances globales, les modèles **kNN** et **Naive Bayes** ont montré des résultats proches avec des temps de calcul bien plus faibles. Selon les besoins (précision maximale ou exécution rapide), l'un ou l'autre pourrait être retenu.

5.6 Résultats produits et fichiers générés

À l'issue de l'entraînement et de l'évaluation, plusieurs fichiers ont été générés pour documenter les résultats et faciliter leur exploitation :

- **Fichiers `.pkl`** : chaque modèle entraîné a été sauvegardé sous forme de fichier `pickle` dans le dossier `models/`, permettant une réutilisation sans réentraînement.
- **Fichiers `.csv`** : pour chaque modèle, un fichier contenant les résultats sur les ensembles de validation et de test a été généré. Ces fichiers incluent les colonnes :
  - `Probabilité_démission` : score prédit par le modèle,
  - `Classe_prévue` : classe binaire après application du seuil (0.6),
  - `Classe_réelle` : classe cible réelle.

  Ces fichiers sont enregistrés dans le dossier `models/CSV/`.
- **Visualisations** : tous les graphiques générés (matrices de confusion, histogrammes des probabilités, visualisations PCA 3D, importance des attributs...) sont sauvegardés dans le dossier `models/figures/`.

Ces résultats fournissent à la fois une vision quantitative et visuelle du comportement des modèles sur les différentes phases du projet.

6 Conclusion

Ce projet m'a permis de mettre en pratique l'ensemble des étapes d'un processus de modélisation prédictive, en partant d'un jeu de données brut jusqu'à la production de résultats exploitables. Je me suis notamment appuyé sur le TP de fouille de données réalisé en cours, qui m'a servi de base précieuse pour structurer mon travail et avancer de manière méthodique.

Le projet s'est déroulé selon les grandes étapes classiques de la data science :

- Nettoyage et préparation des données,
- Traitement du déséquilibre de classes par suréchantillonnage,
- Découpage en jeux d'entraînement, validation et test,
- Entraînement de plusieurs modèles supervisés (régression logistique, SVM, kNN, Naive Bayes),
- Évaluation des performances par différentes métriques (accuracy, F1-score, rappel),
- Interprétation des résultats et identification des attributs les plus influents.

Ce travail m'a permis de mieux comprendre le fonctionnement des algorithmes de classification, leurs forces et leurs limites, ainsi que l'importance de chaque étape du pipeline. Les résultats obtenus sont globalement satisfaisants et montrent qu'une modélisation simple mais bien structurée peut donner des informations utiles à la prise de décision.