Rapport du projet PSID

LoveMyPet

Imane Errahmani – 43014746
Faiz Sadikou Adenle – 42000139
Malek Messaoudi – 42000319

19 avril 2025

Table des matières

1 Présentation

Ce rapport présente le projet **LoveMyPet**, réalisé dans le cadre du module PSID.

Ce document est une première version du rapport (v0.1) présentant le contexte, les membres de l'équipe, et une première synthèse des travaux réalisés.

1.1 Problématique

Chaque année, de nombreux animaux se retrouvent dans des refuges sans trouver de foyer. Malgré les efforts des associations, une part importante d'entre eux reste en attente d'adoption pendant des mois, ce qui entraîne des conséquences négatives pour leur bien-être. Cette problématique soulève une question essentielle : comment augmenter leurs chances d'être adoptés rapidement ?

Grâce aux données disponibles sur les profils des animaux (description, caractéristiques physiques, informations de santé, etc.), il devient possible d'analyser les éléments qui influencent le temps avant adoption. L'analyse de ces facteurs peut contribuer à mieux présenter les animaux et améliorer les stratégies de sensibilisation.

1.2 Solution

Pour répondre à cette problématique, nous avons développé le projet **LoveMyPet**, une étude basée sur l'analyse de données issues de la plateforme PetFinder. Le but est d'identifier les facteurs influençant la rapidité d'adoption des animaux en appliquant des techniques de data science et d'intelligence artificielle.

Notre solution se structure autour de plusieurs étapes clés :

— Nettoyage et préparation des données (tabulaires, textuelles et visuelles).
— Visualisation exploratoire pour comprendre les tendances.
— Modélisation prédictive pour estimer la rapidité d'adoption.
— Interprétation et recommandations pour améliorer la présentation des profils.

Ce travail vise à outiller les refuges et associations d'adoption avec des indicateurs clairs et des leviers d'action pour favoriser des adoptions plus rapides et mieux ciblées.

1.3 Public cible

Le projet s'adresse principalement :

— aux **associations de protection animale** qui cherchent à optimiser le placement des animaux de compagnie dans des foyers adaptés ;
— aux **refuges pour animaux**, qui doivent prendre des décisions rapides et informées pour augmenter les chances d'adoption ;
— aux **utilisateurs particuliers**, intéressés par l'adoption et désireux d'accéder à des recommandations personnalisées selon leurs préférences ou leur profil ;
— aux **organismes publics ou municipaux**, pour la gestion et l'amélioration des campagnes d'adoption sur leur territoire.

1.4 Personas

Pour mieux illustrer les besoins auxquels répond le projet, nous avons identifié deux personas types :

Persona 1 – Clara, bénévole dans un refuge
— **Âge** : 29 ans
— **Profession** : étudiante vétérinaire et bénévole
— **Besoins** : mieux comprendre pourquoi certains animaux mettent du temps à être adoptés ; savoir comment mettre en avant les bons critères dans les fiches descriptives.
— **Objectif** : améliorer les chances d'adoption des animaux les plus vulnérables.

Persona 2 – Marc, adoptant potentiel
— **Âge** : 45 ans
— **Profession** : enseignant
— **Besoins** : adopter un animal compatible avec son mode de vie et ses contraintes familiales.
— **Objectif** : accéder à des recommandations d'animaux faciles à adopter et correspondant à son environnement.

2 Architecture métier

2.1 Frontend

Le frontend du projet LoveMyPet a été développé en utilisant le framework **React** avec l'outil de bundling **Vite**, qui offre une configuration légère et un temps de compilation rapide. Pour le design et le style de l'interface, nous avons adopté **TailwindCSS**, un framework CSS utilitaire moderne, qui permet une personnalisation rapide et cohérente des composants graphiques.

L'interface utilisateur est conçue pour être simple, interactive et informative. Elle présente les graphiques de visualisation générés à partir des données analysées, notamment des diagrammes interactifs produits avec **Plotly.js**, qui permettent une navigation fluide et dynamique entre différentes catégories d'animaux, races, ou facteurs d'adoption. Des composants personnalisés comme les boutons de filtre ou les titres numérotés ont été développés pour guider la lecture et l'exploration des résultats.

2.2 Backend

Le backend est construit avec **FastAPI**, un framework Python moderne et performant, parfaitement adapté pour construire des API REST. Ce choix permet une communication fluide entre l'interface utilisateur et les données analysées en backend, tout en assurant un temps de réponse rapide.

Les données sont traitées avec **pandas**, une bibliothèque de référence pour la manipulation de données tabulaires en Python. Plusieurs endpoints ont été définis pour servir les résultats des analyses : répartition des types d'animaux, distribution des âges, top races stérilisées, impact de la stérilisation sur l'adoption rapide, etc.

Le backend se charge également du chargement et du nettoyage des jeux de données à partir de fichiers CSV, ainsi que de leur enrichissement par jointures avec des fichiers de correspondance (races, couleurs, états géographiques). Les analyses sont structurées pour fournir des réponses directement exploitables par les composants de visualisation du frontend.

3 Pratiques de Collaboration et de DevOps Project

3.1 Management

La gestion de projet s'est appuyée sur une méthodologie **Scrum**, adaptée aux projets étudiants et favorisant un développement itératif et collaboratif. Pour organiser les tâches, nous avons utilisé un tableau **Kanban** sur GitHub, permettant une répartition claire des responsabilités et une visualisation en temps réel de l'avancement du projet.

Nous avons également mis en place des réunions d'équipe hebdomadaires. Ces sessions régulières, similaires aux "stand-ups" Scrum, ont servi à faire le point sur les tâches réalisées, discuter des obstacles rencontrés et ajuster les priorités collectives. Cette dynamique a renforcé la cohésion de l'équipe et favorisé une communication continue.

3.2 Versionnement

Le versionnement du code a été centralisé via la plateforme **GitHub**, qui a permis un suivi rigoureux de l'évolution du projet. Grâce à l'utilisation des branches, commits et pull requests, chaque membre a pu travailler de manière autonome tout en maintenant l'intégrité du code commun. Les revues de code ont contribué à la détection précoce des erreurs et à la diffusion des bonnes pratiques au sein de l'équipe.

3.3 Intégration Continue et Déploiement Continu

Nous avons mis en place un pipeline d'intégration et de déploiement continu (**CI/CD**) en utilisant **GitHub Actions**. Chaque push sur la branche principale déclenche automatiquement un processus de build et de déploiement. Ce mécanisme garantit que les nouvelles fonctionnalités et corrections sont rapidement intégrées et accessibles.

Ce système permet de livrer plus fréquemment et avec davantage de fiabilité, tout en réduisant les risques d'erreur humaine. Le pipeline inclut des étapes de vérification de la qualité du code, ce qui renforce la stabilité de l'application avant chaque mise en production.

3.4 Qualité du code

Le contrôle qualité du code est assuré par l'outil d'analyse statique **Codacy**, intégré dans notre workflow GitHub. Codacy permet de détecter automatiquement les mauvaises pratiques de développement, les complexités inutiles ou les répétitions de code.

Nous avons personnalisé les règles d'analyse afin de coller aux standards définis par l'équipe, garantissant ainsi une uniformité dans le style de développement. Les rapports générés permettent de suivre en continu l'état du code et d'intervenir de manière proactive pour améliorer la robustesse du projet.

4 Partie Data Analytique

4.1 Source de données

Les données utilisées pour ce projet proviennent de la compétition Kaggle intitulée *PetFinder.my Adoption Prediction*, accessible à l'adresse suivante :

https://www.kaggle.com/competitions/petfinder-adoption-prediction

Nom du dataset : PetFinder.my Adoption Prediction
Source : PetFinder.my, en partenariat avec Kaggle
Pays d'origine des données : Malaisie
Date de mise à disposition : 2019

Contexte :
Chaque jour, des millions d'animaux errants sont abandonnés ou euthanasiés par manque de foyers adoptants. PetFinder.my est la principale plateforme malaisienne de protection animale depuis 2008, avec une base de plus de 150 000 animaux enregistrés.

Cette plateforme collabore avec des ONG, des entreprises, des médias et des organisations internationales pour améliorer le bien-être animal. Dans ce contexte, les profils en ligne des animaux (textes descriptifs, caractéristiques des photos, etc.) ont un rôle majeur dans la rapidité d'adoption.

Objectif de la compétition :
L'objectif est de prédire la rapidité avec laquelle un animal est adopté, en fonction des métadonnées présentes dans son profil. Ces prédictions pourraient être intégrées dans des outils d'intelligence artificielle (IA), comme le *Cuteness Meter*, pour conseiller les refuges sur l'optimisation des profils d'animaux. Cela permettrait d'accélérer les adoptions, d'éviter l'euthanasie, et d'améliorer significativement le bien-être animal à l'échelle mondiale.

Intérêt pour le projet :
Ce dataset constitue une base riche pour une analyse complète des facteurs d'adoption : il comprend des données tabulaires, textuelles et visuelles, et permet de tester des approches analytiques et prédictives variées.

4.2 Présentation du dataset

Structure du dataset :
- `train.csv` : Données d'entraînement (variables tabulaires et textuelles)
- `test.csv` : Données de test à prédire
- `sample_submission.csv` : Exemple de fichier de soumission
- `breed_labels.csv` : Identifiants et noms de races, ainsi que leur type (chien ou chat)
- `color_labels.csv` : Noms associés à chaque code couleur
- `state_labels.csv` : Noms associés à chaque code de région (Malaisie)

Détail des variables : Le fichier `train.csv` contient les variables suivantes, décrivant les caractéristiques des animaux disponibles à l'adoption sur la plateforme PetFinder.my :
- **PetID** : Identifiant unique de chaque profil d'animal.
- **AdoptionSpeed** : Cible à prédire. Elle indique en combien de temps l'animal a été adopté.
  - 0 : le jour même
  - 1 : entre 1 et 7 jours
  - 2 : entre 8 et 30 jours
  - 3 : entre 31 et 90 jours
  - 4 : non adopté après 100 jours
- **Type** : Type d'animal (1 = chien, 2 = chat)

- **Name** : Nom de l'animal (vide si non renseigné)
- **Age** : Âge de l'animal (en mois)
- **Breed1** : Race principale de l'animal (identifiant numérique)
- **Breed2** : Race secondaire (si l'animal est croisé). Peut être 0.
- **Gender** : Sexe de l'animal
  - 1 : Mâle
  - 2 : Femelle
  - 3 : Mixte (groupe d'animaux)
- **Color1, Color2, Color3** : Couleurs dominantes (jusqu'à 3 couleurs)
- **MaturitySize** : Taille de l'animal à l'âge adulte
  - 1 : Petite
  - 2 : Moyenne
  - 3 : Grande
  - 4 : Très grande
  - 0 : Non spécifié
- **FurLength** : Longueur du pelage
  - 1 : Court
  - 2 : Moyenmercredi
  - 3 : Long
  - 0 : Non spécifié
- **Vaccinated** : Statut de vaccination
  - 1 : Oui
  - 2 : Non
  - 3 : Inconnu
- **Dewormed** : Vermifugation effectuée ou non (1 = oui, 2 = non, 3 = inconnu)
- **Sterilized** : Animal stérilisé ou non (1 = oui, 2 = non, 3 = inconnu)
- **Health** : État de santé de l'animal
  - 1 : En bonne santé
  - 2 : Légèrement blessé
  - 3 : Gravement blessé
  - 0 : Non spécifié
- **Quantity** : Nombre d'animaux représentés par le profil (souvent 1)
- **Fee** : Frais d'adoption demandés (0 = gratuit)
- **State** : État géographique de l'animal (en Malaisie)
- **RescuerID** : Identifiant du sauveteur ou refuge ayant publié l'annonce
- **VideoAmt** : Nombre de vidéos associées à l'animal
- **PhotoAmt** : Nombre de photos disponibles
- **Description** : Texte libre décrivant l'animal. Langues utilisées : anglais, malais, ou chinois.

Définition de la variable cible : `AdoptionSpeed`

- **0** : adopté le jour même
- **1** : adopté entre 1 et 7 jours
- **2** : adopté entre 8 et 30 jours
- **3** : adopté entre 31 et 90 jours
- **4** : non adopté après 100 jours

Données complémentaires :

— **Images** : Chaque animal possède une ou plusieurs photos, analysées via l'API Google Vision (annotation faciale, étiquettes, propriétés visuelles, textes).
— **Sentiment Analysis** : Les descriptions textuelles ont été soumises à l'API Google Natural Language pour extraire la tonalité émotionnelle (positif, neutre, négatif).

4.3 Nettoyage et manipulation des Données

Prétraitement des données La phase de prétraitement a été essentielle pour garantir la qualité et la fiabilité des analyses ultérieures. Elle s'est déroulée selon les étapes suivantes :

Contrôle de qualité
— **Détection des doublons** : Aucune duplication n'a été identifiée dans le jeu de données, ce qui écarte les biais potentiels liés à la redondance des profils.
— **Gestion des valeurs manquantes** : Une vérification systématique a révélé l'absence de valeurs nulles. Il n'a donc pas été nécessaire de procéder à une imputation ou à l'exclusion d'observations.

Suppression des variables non pertinentes Certaines variables ont été supprimées, car jugées non informatives ou peu pertinentes pour l'objectif de prédiction de la rapidité d'adoption :
— **VideoAmt** : Le nombre de vidéos associées aux animaux était quasi constant, et donc non discriminant.
— **State** : La localisation (État en Malaisie) ne présentait pas de corrélation significative avec la variable cible *AdoptionSpeed*.
— **Name** : Souvent absente ou trop spécifique, cette variable n'apportait pas de valeur prédictive généralisable.
— **Description** : Bien que potentiellement riche en information, ce champ non structuré n'a pas été exploité dans cette première version, faute de traitement NLP prévu.
— **RescuerID** : Identifiant du sauveteur, non pertinent pour l'analyse, car non directement lié aux caractéristiques des animaux.

Format des données Le jeu de données était déjà majoritairement numérique. Les variables catégorielles comme *Breed* ou *Color* sont fournies sous forme d'identifiants entiers, avec les correspondances disponibles dans des fichiers annexes fournis par Kaggle (`breed_labels.csv`, `color_labels.csv`, etc.).

Encodage et transformations Aucune transformation supplémentaire (encodage one-hot, normalisation, standardisation) n'a été nécessaire à ce stade. Le format initial des données était directement exploitable pour les modèles de machine learning tabulaire.

Résultat À l'issue de cette phase, un jeu de données nettoyé et allégé a été obtenu, contenant uniquement les variables jugées pertinentes. Ce jeu est enregistré dans le fichier `data_clean.csv` et constitue la base des analyses exploratoires et des modèles prédictifs développés par la suite.

4.4 Les graphiques présentés

Nous avons intégré plusieurs visualisations interactives dans notre tableau de bord afin d'analyser les facteurs influençant l'adoption rapide des animaux. Ces graphiques sont construits dynamiquement en React à l'aide de la bibliothèque Plotly.js, et les données sont récupérées depuis notre API backend.

Adoption Animale : Les Clés pour une Adoption Rapide des Chiens et Chats Ce graphique en barres empilées présente la répartition des adoptions selon le délai (le jour même, entre 1 et 7 jours, entre 8 et 30 jours), en fonction de plusieurs caractéristiques : taille à maturité, état de santé, vaccination, vermifugation et stérilisation. L'analyse est segmentée par type d'animal (chiens ou chats).

- Chez les chiens, les individus en bonne santé, vermifugés et de taille moyenne sont les plus rapidement adoptés.
- Les chiens non stérilisés sont adoptés plus rapidement que ceux qui le sont, ce qui peut refléter certaines préférences culturelles ou un manque d'information sur les bénéfices de la stérilisation.
- La vaccination n'a pas un impact majeur sur la rapidité d'adoption chez les chiens.
- Chez les chats, les mêmes tendances s'observent : la santé générale et la vermifugation influencent positivement l'adoption rapide.
- Cependant, un paradoxe apparaît : les chats non vaccinés sont adoptés plus vite que les vaccinés. Cela pourrait être lié à l'âge (les plus jeunes n'étant pas encore vaccinés) ou à des perceptions erronées.
- La stérilisation a également un effet inverse aux attentes : les chats non stérilisés sont adoptés plus rapidement.

En résumé, les facteurs liés à la santé visible jouent un rôle crucial dans l'adoption rapide. Toutefois, certaines perceptions des adoptants – notamment sur la stérilisation et la vaccination – semblent aller à l'encontre des recommandations vétérinaires, d'où l'intérêt de renforcer la sensibilisation.

Analyse du lien entre l'âge et la non-vaccination Ce graphique en barres illustre le nombre d'animaux non vaccinés selon leur tranche d'âge. Il fait suite à l'observation précédente d'un nombre élevé d'animaux proposés à l'adoption sans vaccination, et vise à mieux comprendre cette situation en l'associant à l'âge des animaux.

- Une large majorité des animaux non vaccinés ont moins de 10 mois, avec un total de **6580 cas**, soit une concentration très marquée dans cette première tranche.
- Les chiffres chutent rapidement dans les groupes suivants : **383 individus** entre 11 et 20 mois, puis **134** entre 21 et 30 mois, et seulement quelques dizaines au-delà.
- Cela s'explique par le fait que de nombreux animaux sont proposés à l'adoption très jeunes, souvent avant l'âge recommandé pour administrer les premiers vaccins.
- Ce phénomène reflète aussi une préférence des adoptants pour les chiots ou chatons, qui n'ont pas encore suivi de protocole vaccinal complet.
- Il est également possible que certains refuges laissent volontairement cette responsabilité aux adoptants, dans le cadre d'accords personnalisés ou pour des raisons économiques.

En conclusion, la forte proportion d'animaux non vaccinés s'explique avant tout par leur jeune âge au moment de la mise à l'adoption, et ne constitue pas nécessairement un indicateur de négligence ou de mauvaise prise en charge.

Est-ce que le nombre de photos postées augmente la vitesse d'adoption ? Ce graphique combine deux analyses complémentaires : la première présente la vitesse d'adoption par groupe de photos, la seconde met en lumière le nombre total d'animaux adoptés et non adoptés selon le nombre de photos disponibles.

- Les animaux avec 0 à 5 photos sont ceux qui sont adoptés le plus rapidement, notamment le jour même ou dans la première semaine.
- Lorsque le nombre de photos augmente (au-delà de 10), la rapidité d'adoption diminue fortement. Les adoptions immédiates deviennent rares, voire inexistantes pour les groupes avec plus de 20 photos.
- Ce phénomène suggère un possible effet de saturation ou de surcharge cognitive : trop d'images rendraient la décision plus difficile à prendre.
- Malgré leur forte visibilité, les animaux du groupe "0-5 photos" représentent aussi le plus grand nombre de non-adoptés après 100 jours, ce qui indique qu'un faible nombre de photos n'est pas un gage de succès à long terme.
- Les groupes intermédiaires (6-10 photos) semblent offrir un meilleur équilibre entre visibilité et efficacité d'adoption.

Ainsi, un nombre modéré de photos (entre 5 et 10) apparaît comme le plus efficace pour favoriser l'adoption rapide. Au-delà, il devient essentiel de miser sur la qualité et la pertinence des images, ainsi que sur un renouvellement régulier des visuels.

Analyse de la vitesse d'adoption par genre et par taille Ce graphique en radar explore l'influence du genre (mâle ou femelle) sur la vitesse d'adoption des animaux, tout en intégrant un filtre par taille (petit, moyen, grand, très grand). Il permet de comparer visuellement les comportements d'adoption selon ces deux caractéristiques.

- Les femelles sont généralement adoptées plus rapidement que les mâles, surtout pour les petites et moyennes tailles.
- Pour les grandes tailles, les mâles deviennent plus nombreux à être adoptés, notamment après les premiers jours, et ils dominent aussi parmi les animaux non adoptés.
- En ce qui concerne les très grandes tailles, les adoptions sont initialement équilibrées entre les sexes, mais les mâles finissent par être plus représentés.
- La taille de l'animal semble jouer un rôle dans les préférences d'adoption, probablement lié à l'image projetée (compagnon affectueux pour les petits animaux, protecteur ou utilitaire pour les grands).
- Ces différences pourraient refléter des stéréotypes ou attentes comportementales selon le genre et la taille.

Ce graphique suggère que des stratégies différenciées selon le sexe et la taille des animaux pourraient améliorer les chances d'adoption, en jouant sur les perceptions et les attentes des futurs adoptants.

Plus c'est court, plus c'est adopté ? Étude sur la rapidité d'adoption selon la fourrure Ce graphique en camembert met en évidence la répartition des adoptions rapides (dans les 7 premiers jours) selon la longueur de la fourrure des animaux : courte, moyenne ou longue. Il permet d'observer l'impact d'un critère physique souvent négligé mais potentiellement déterminant dans le processus d'adoption.

- Les animaux à poil court sont adoptés le plus rapidement, représentant 52,7% des adoptions rapides. Ils bénéficient d'une image de facilité d'entretien (moins de toilettage, moins de poils), ce qui séduit les adoptants.

— Les animaux à poil mi-long arrivent ensuite avec 38,5% des adoptions rapides. Bien qu'ils soient un peu moins prisés que les poils courts, ils restent dans une proportion significative.
— Les animaux à poil long ne représentent que 8,9% des adoptions rapides. Leur entretien plus contraignant (brossage, risque de nœuds) pourrait expliquer cette faible attractivité.
— La fourrure est donc un facteur non négligeable dans la prise de décision : elle renvoie à des représentations pratiques (entretien, propreté) et parfois esthétiques.
— Cette analyse peut aider les refuges à adapter leur communication, en valorisant les aspects positifs des animaux à fourrure longue, souvent moins rapidement adoptés.

Ce critère simple mais visuellement marquant joue donc un rôle dans la perception initiale des adoptants, influençant directement la vitesse d'adoption.

Comprendre la stérilisation animale : une question d'âge et de genre Cette double visualisation propose une lecture croisée des comportements de stérilisation chez les chiens et les chats, en distinguant les sexes et les tranches d'âge. L'objectif est de comprendre si certaines populations animales sont plus souvent stérilisées que d'autres, et à quel moment cela intervient dans leur vie.

— Le premier graphique présente le nombre d'animaux stérilisés et non stérilisés selon leur sexe. On observe une nette prédominance des animaux non stérilisés, en particulier chez les mâles.
— Les femelles, quant à elles, sont davantage ciblées par les campagnes de stérilisation, probablement pour des raisons liées à la reproduction et à la gestion des portées.
— Le second graphique, en ligne, illustre l'évolution du pourcentage de stérilisation en fonction de l'âge. Les taux augmentent avec le temps, confirmant que les animaux sont rarement stérilisés très jeunes.
— Chez les chiens, la stérilisation progresse de manière plus linéaire, tandis que chez les chats, la courbe est plus abrupte, suggérant une intervention plus précoce.
— L'écart entre mâles et femelles est aussi plus marqué chez les chiens, là où chez les chats, les différences de genre sont moins prononcées.

Ces résultats révèlent l'existence de pratiques différenciées selon les espèces, les sexes et les âges. Ils peuvent servir de base pour ajuster les stratégies de sensibilisation et les politiques vétérinaires, en favorisant une stérilisation plus équitable et ciblée.

Impact de la stérilisation sur l'adoption rapide selon la race Ce graphique à barres groupées compare le taux d'adoption rapide des animaux stérilisés et non stérilisés pour les 10 races les plus fréquentes, avec un filtre par type d'animal (chien ou chat) et par pureté de race.

— Les animaux stérilisés ont généralement un meilleur taux d'adoption rapide. Cela est perçu comme un indicateur de soin et de responsabilité.
— Les races populaires (Labrador, Poodle, Shih Tzu) montrent un effet amplifié : la combinaison stérilisation + pureté de race améliore fortement les chances d'adoption.
— Pour les races controversées (Pit Bull, Bull Terrier), la stérilisation a un effet modéré, souvent éclipsé par les stéréotypes négatifs.
— Chez les races mixtes, la stérilisation est le principal levier d'amélioration, car la pureté de race n'est pas perçue comme un avantage.

L'analyse croisée des effets montre que la stérilisation et la race pure forment un binôme stratégique pour l'adoption rapide. Toutefois, une communication ciblée reste nécessaire pour certaines races stigmatisées.

Top 10 Races Pures et Mixtes les Plus Rapides à Être Adoptées Ce graphique à barres présente, pour chaque type d'animal (chiens ou chats), les 10 races pures (en vert) et 10 races mixtes (en rouge) adoptées le plus rapidement. L'indicateur utilisé est la vitesse moyenne d'adoption (0 = adopté le jour même, 4 = non adopté après 100 jours).

- **Chez les chiens**, certaines races mixtes surpassent même les pures : le Maltais mixte (1,30) et le Cocker Spaniel mixte (1,50) sont plus rapidement adoptés que leurs homologues de race pure.
- Les races pures restent dominantes dans certains cas : Basset Hound (1,60), Border Collie (1,69) et Pug (1,71) témoignent d'un fort attrait.
- **Chez les chats**, les races mixtes sont clairement en tête : Ragdoll mixte (1,70) et Maine Coon mixte (1,75) sont les plus rapidement adoptées, surpassant les versions pures de ces races.
- Des races pures comme le Domestic Long Hair (1,70) et le Russian Blue (1,91) conservent néanmoins un bon taux d'adoption.
- **Comparaison chiens vs chats** : la pureté de race semble jouer un rôle plus fort chez les chiens que chez les chats, pour lesquels les races mixtes sont davantage valorisées.

Cette analyse montre que les préférences des adoptants varient selon l'espèce. Tandis que les races pures bénéficient d'un capital de confiance chez les chiens, les chats mixtes séduisent davantage, possiblement pour leur originalité perçue. Les refuges peuvent adapter leur communication selon ces tendances pour améliorer les chances d'adoption rapide.

Analyse interactive des facteurs de non-adoption Ce graphique radar met en évidence les caractéristiques typiques des animaux qui ne trouvent pas d'adoptants après plus de 100 jours. Il permet de visualiser les variables influençant négativement l'adoption : âge plus élevé, frais d'adoption élevés, nombre de photos réduit, pelage et taille standards.

- Les animaux non adoptés ont en moyenne 13,7 mois, ce qui les rend moins attractifs que les jeunes chiots ou chatons.
- Un faible nombre de photos (en moyenne 3,32) réduit l'impact émotionnel des profils.
- Les frais d'adoption supérieurs à 21 MYR peuvent décourager les potentiels adoptants.
- La santé ne semble pas être un facteur majeur de rejet : la majorité sont en bonne santé.

L'interprétation globale suggère que l'amélioration de la présentation visuelle (photos, descriptions, vidéos) ainsi qu'une stratégie tarifaire adaptée pourraient significativement accroître les chances d'adoption.

Tous les graphes sont disponibles sur notre site dans la vidéo de démonstration ainsi que sur GitHub. Vous trouverez les liens vers ces deux ressources dans le fichier `lien_code_source.txt`.

5 Analyse statistique

5.1 Résultats de Kaggle : une base de comparaison

Bien que ce jeu de données ait déjà été exploité dans le cadre d'une compétition organisée sur *Kaggle*, les résultats obtenus par les participants ont plafonné autour d'une précision de 45%, malgré l'utilisation de modèles avancés et d'approches algorithmiques sophistiquées. Ce plafond de performance interroge : est-ce dû à la qualité des données, à des facteurs structurels ignorés, ou à des limites des modèles utilisés ?

Ce projet propose de prendre le problème à la racine : plutôt que de chercher uniquement à améliorer les scores, il s'agit de comprendre pourquoi ces performances stagnent, en explorant en profondeur les données, les variables et les relations cachées, ce que les modèles ont peut-être raté. L'objectif est double : mieux expliquer le phénomène d'adoption, et envisager des pistes concrètes pour faire mieux.

5.2 Analyse des valeurs aberrantes (outliers)

Les boxplots ont été utilisés pour analyser la distribution des variables du jeu de données. Pour la variable continue **Age**, des valeurs aberrantes dépassant 150 (jusqu'à 250) ont été détectées, ce qui est anormal pour l'âge d'un animal de compagnie. Ces données aberrantes ont été supprimées afin d'améliorer la qualité des données et d'éviter un impact négatif sur le modèle.

Les autres variables, telles que **Gender**, **Color1**, **Color2**, **Color3**, **MaturitySize**, **FurLength**, **Vaccinated** et **Dewormed**, sont des variables catégoriques codées numériquement (par exemple, de 1 à 7 pour `Color1`). Les points isolés observés dans leurs boxplots correspondent à des catégories moins fréquentes, ce qui est attendu pour des variables catégoriques, et ne constitue donc pas des anomalies.

Cette analyse a permis de confirmer la cohérence globale des données après nettoyage.

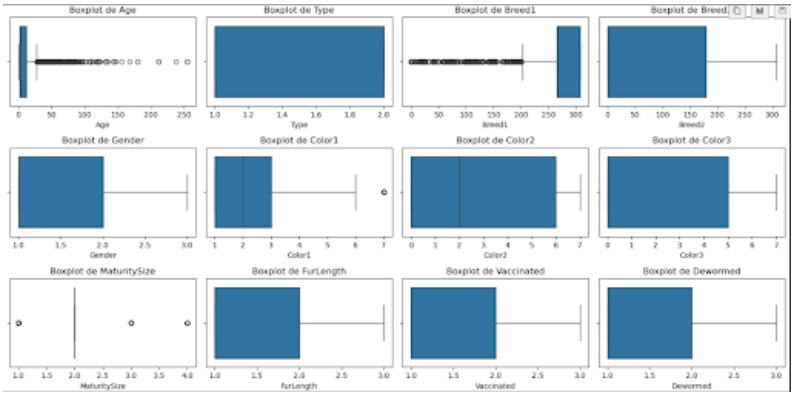


FIGURE 1 – Boxplot illustrant les valeurs aberrantes de la variable Age

5.3 Premier test : Régression logistique multiclasse

Nous avons testé notre modèle à l'aide d'une régression logistique multiclasse afin de vérifier la cohérence des données et d'évaluer les performances de base. L'objectif était notamment de comparer les résultats obtenus à ceux du concours Kaggle, où une précision de 0,45 avait été atteinte.

Pour l'entraînement, nous avons utilisé les variables suivantes comme descripteurs : `Type`, `Age`, `Breed1`, `Breed2`, `Gender`, `Color1`, `Color2`, `Color3`, `MaturitySize`, `FurLength`, `Vaccinated`, `Dewormed`, `Sterilized`, `Health`, `Fee`, `VideoAmt` et `PhotoAmt`. La variable cible que nous cherchons à prédire est `AdoptionSpeed`, qui prend des valeurs entières de 0 à 4.

Le modèle a généré un **MSE de 2,2317**, avec un **biais de 1,9788** et une **variance de 0,2529**. La précision moyenne est d'environ **0,2699**. Ces résultats indiquent clairement un problème de **sous-apprentissage** : le biais est très élevé, ce qui suggère que le modèle n'arrive pas à capter la structure des données, et la précision est globalement très faible.

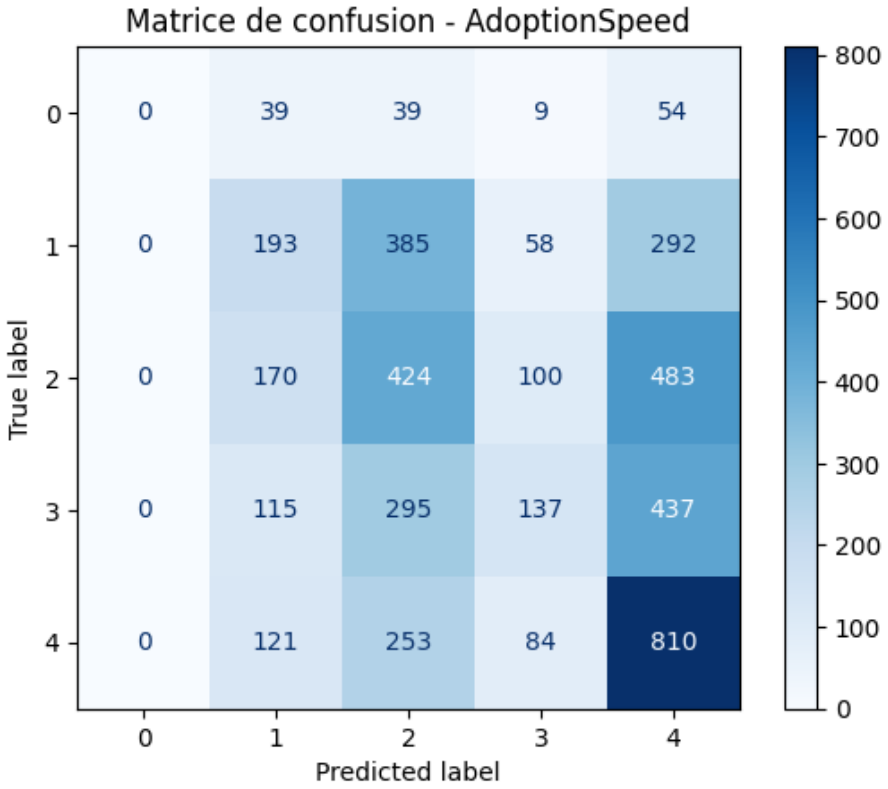
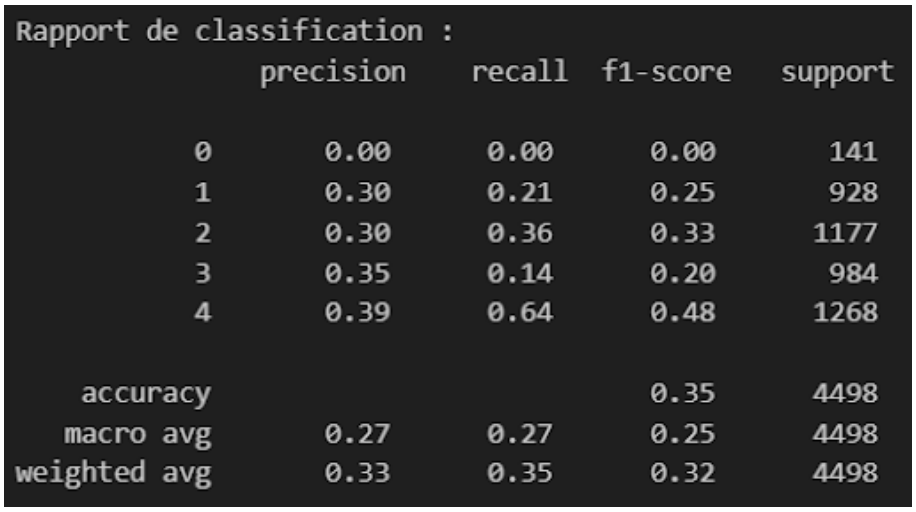


FIGURE 2 – Matrice de confusion obtenue avec la régression logistique



Rapport de classification :

	precision	recall	f1-score	support
0	0.00	0.00	0.00	141
1	0.30	0.21	0.25	928
2	0.30	0.36	0.33	1177
3	0.35	0.14	0.20	984
4	0.39	0.64	0.48	1268
accuracy			0.35	4498
macro avg	0.27	0.27	0.25	4498
weighted avg	0.33	0.35	0.32	4498

FIGURE 3 – Rapport de classification - Régression logistique multiclasse

L'analyse du rapport de classification et de la matrice de confusion confirme ces observations. On remarque notamment que la classe **0 n'est jamais prédite**, avec une précision et un rappel de `0.00`. Les autres classes sont également mal identifiées, avec des *f1-scores* faibles et déséquilibrés. L'exactitude globale du modèle est d'environ **35 %**, ce qui reste bien en dessous des attentes.

Un autre point préoccupant est le fort **déséquilibre dans la distribution des classes** au sein de la variable cible. La classe 0, par exemple, compte environ **six fois moins d'occurrences** que les autres, ce qui impacte fortement sa prédiction. Ce déséquilibre rend l'apprentissage difficile pour le modèle, qui a tendance à ignorer les classes minoritaires.

5.4 Déséquilibre des classes

La figure ci-dessous présente la répartition des animaux selon la variable *AdoptionSpeed*, qui représente le délai d'adoption d'un animal. On observe une forte disparité entre les différentes classes. Plus de 4 000 animaux appartiennent à la classe 4 (non adoptés après plus de 100 jours), tandis que seulement 410 animaux relèvent de la classe 0 (adoptés le jour même).

Ce déséquilibre marqué peut poser un problème lors de l'entraînement de modèles de classification. En effet, un modèle pourrait être biaisé en faveur des classes majoritaires, au détriment des classes minoritaires qui sont pourtant d'un grand intérêt dans le contexte de prédiction. Par exemple, la classe 0, bien que peu représentée, fournit des informations précieuses sur les profils d'animaux très rapidement adoptés.

Ce constat justifie l'exploration de différentes stratégies de regroupement des classes ou de transformation de la cible, dans le but de construire un modèle plus robuste et plus pertinent.

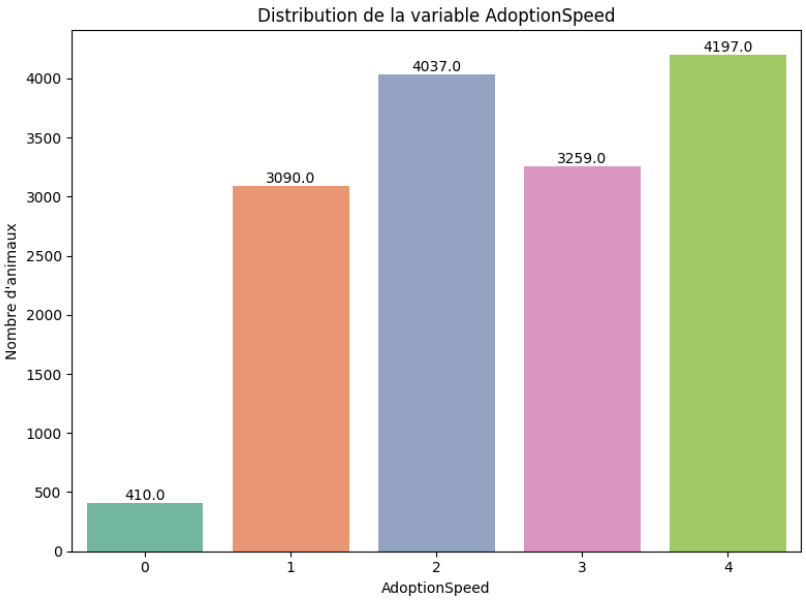


FIGURE 4 – Répartition des animaux selon la vitesse d'adoption (AdoptionSpeed)

5.5 Utilisation de SMOTE pour l'équilibrage des classes

Le dataset présente un déséquilibre marqué entre les classes de la variable *AdoptionSpeed*, avec certaines classes nettement surreprésentées. Par exemple, la classe 0 comporte très peu d'échantillons par rapport aux classes 2 et 4. Ce déséquilibre a un impact négatif sur l'apprentissage du modèle, qui a tendance à privilégier les classes majoritaires au détriment des classes minoritaires.

Pour corriger cela, nous avons utilisé la technique SMOTE (*Synthetic Minority Oversampling Technique*), qui consiste à générer artificiellement de nouveaux exemples pour les classes minoritaires en interpolant les instances existantes.

Distribution des classes avant et après SMOTE

La figure suivante illustre clairement l'effet de SMOTE. Avant son application, les classes sont inégalement réparties, avec une dominance des classes 2 et 4. Après SMOTE, toutes les classes comptent environ 2900 exemples, assurant un apprentissage plus équilibré du modèle.

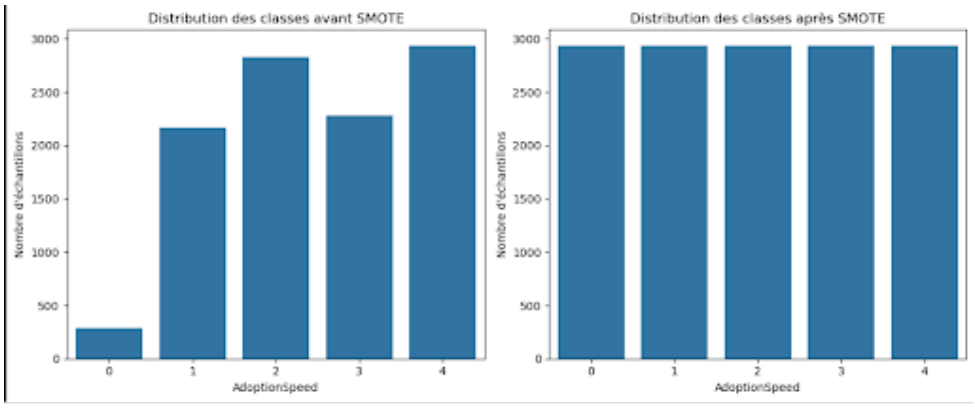


FIGURE 5 – Distribution des classes avant (gauche) et après (droite) SMOTE

Comparaison des performances avant et après SMOTE

Les deux matrices de confusion ci-dessous comparent les performances du modèle avant et après l'équilibrage des classes. Sur le jeu de test original (non équilibré), le modèle montre un fort biais vers les classes fréquentes, comme 2 et 4. Après application de SMOTE, la répartition des prédictions devient plus homogène, traduisant une amélioration sur les classes minoritaires.

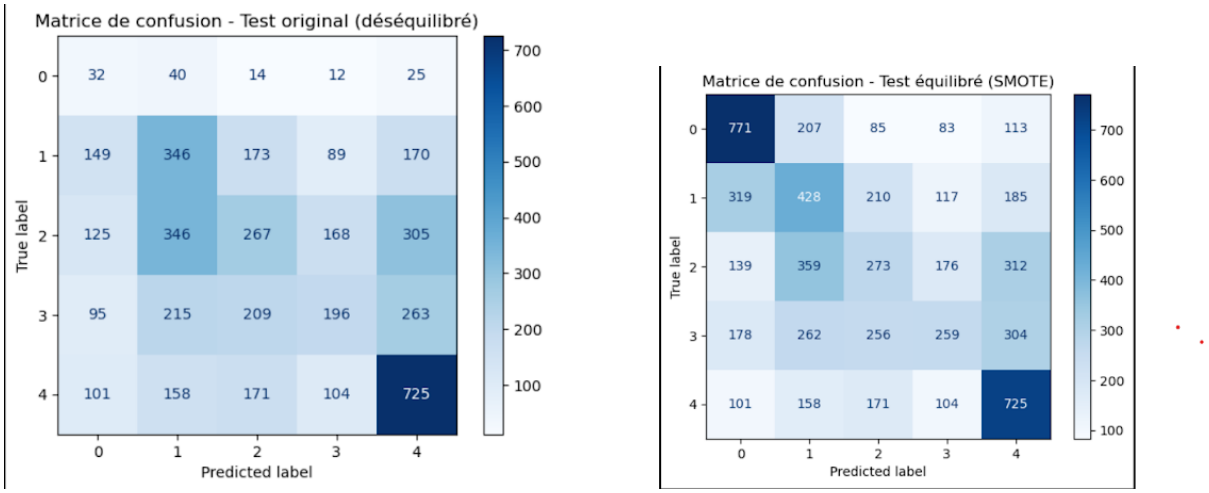


FIGURE 6 – Matrice de confusion avant (gauche) et après (droite) SMOTE

Évolution des métriques globales

Métriques	Précision	Rappel	F1-score
class non équilibré	0.3637	0.3482	0.3461
avec SMOT	0.3762	0.3902	0.3759

FIGURE 7 – Résumé des performances avant et après SMOTE

Les résultats montrent une amélioration modeste mais notable :

— **Avant SMOTE** : précision = 0.3637, rappel = 0.3482, F1-score = 0.3461

— **Après SMOTE** : précision = 0.3762, rappel = 0.3902, F1-score = 0.3759

Ces résultats confirment l'intérêt de SMOTE pour améliorer la reconnaissance des classes rares, tout en limitant l'effet de surapprentissage.

5.6 Amélioration par duplication de la classe 0

Vérifions maintenant si l'absence de prédiction pour la classe `0` est due à son extrême rareté par rapport aux autres classes. Pour cela, nous avons artificiellement augmenté les occurrences de cette classe en multipliant ses **410 lignes par 7**, afin d'obtenir environ **3090 lignes**, ce qui permet d'équilibrer un peu plus les proportions avec les autres classes.

Nous avons ensuite réentraîné notre modèle de régression logistique multiclasse avec ce nouvel ensemble de données. Les résultats obtenus sont les suivants :

— **MSE :** 3.6941

— **Biais :** 3.3316

— **Variance :** 0.3626
— **Précision globale :** 0.2893

La précision a donc légèrement augmenté (**+2 points**), mais cela ne s'est pas traduit par une réelle amélioration du modèle. En réalité, le nombre d'erreurs a augmenté, ce qui montre que le modèle reste peu performant.

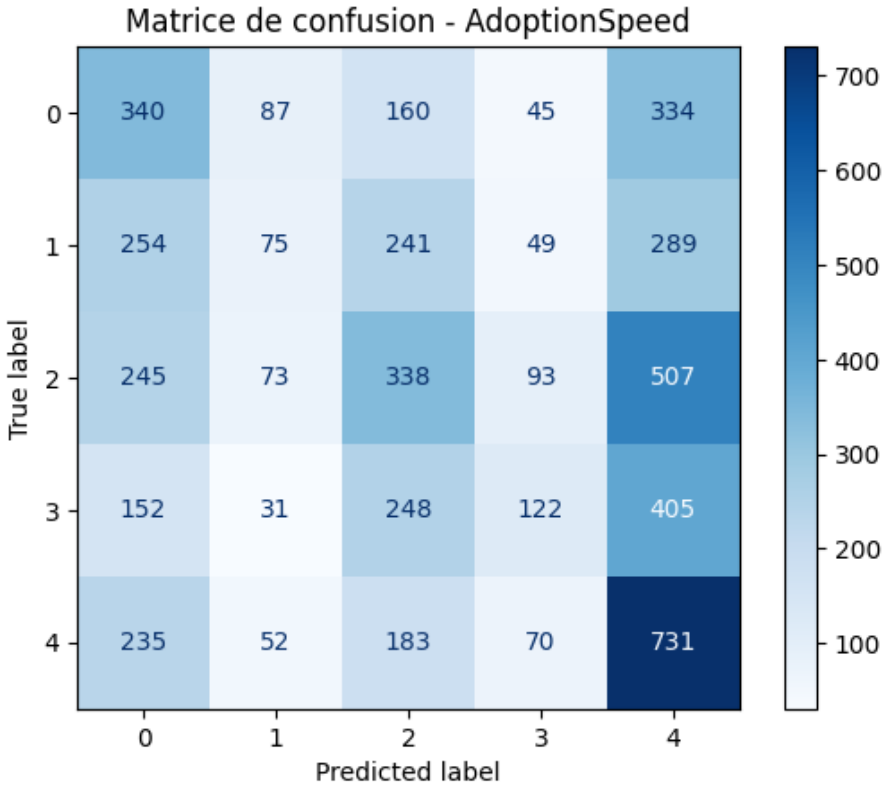


FIGURE 8 – Matrice de confusion après duplication de la classe 0

Rapport de classification :

	precision	recall	f1-score	support
0	0.28	0.35	0.31	966
1	0.24	0.08	0.12	908
2	0.29	0.27	0.28	1256
3	0.32	0.13	0.18	958
4	0.32	0.58	0.41	1271
accuracy			0.30	5359
macro avg	0.29	0.28	0.26	5359
weighted avg	0.29	0.30	0.27	5359

FIGURE 9 – Rapport de classification après duplication de la classe 0

Malgré tout, on observe une **amélioration de la précision pour la classe 0**, qui est désormais mieux prise en compte. Cependant, cette classe est **souvent confondue**

avec la classe 4, ainsi qu'avec d'autres classes, ce qui limite l'efficacité de la prédiction.

En conclusion, le modèle est toujours en situation de **sous-apprentissage**. Il souffrait clairement du manque d'occurrences pour la classe 0, mais il est aussi handicapé par une forte similarité entre les variables descriptives. Cette proximité rend difficile la discrimination entre les différentes classes. Pour améliorer cela, il serait nécessaire d'ajouter de nouvelles variables plus discriminantes, que nous ne possédons pas actuellement.

Au-delà des performances quantitatives obtenues via les modèles supervisés, il est essentiel de mieux comprendre la structure intrinsèque des données. C'est dans cette optique que nous avons mené des analyses exploratoires complémentaires, telles que le clustering non supervisé et l'Analyse des Correspondances Multiples (ACM), afin d'identifier les tendances cachées dans la distribution des classes et de détecter d'éventuels regroupements naturels entre les observations.

5.7 Analyse des tendances globales des variables catégorielles

Les histogrammes ci-dessous illustrent la répartition des variables catégoriques `Type`, `Gender`, `MaturitySize`, `FurLength`, `Vaccinated`, `Dewormed` et `Sterilized` selon les différentes classes de la variable `AdoptionSpeed`.

On observe une tendance générale similaire entre les classes pour toutes ces variables. Pour `Type`, les catégories *Cat* (bleu) et *Dog* (orange) sont globalement équilibrées dans chaque classe (de 0 à 4). En ce qui concerne `Gender`, les animaux de genre *Male* (bleu) et *Female* (orange) sont majoritaires, tandis que la catégorie *Not Specified* (vert) reste marginale.

Pour la variable `MaturitySize`, la taille *Medium* (bleu) domine largement, suivie de *Small* (orange), tandis que les tailles *Large* (vert) et *Extra Large* (rouge) sont peu représentées. Une tendance similaire est observée pour `FurLength`, où les longueurs *Short* (orange) et *Medium* (bleu) prédominent, avec une faible occurrence de *Long* (vert).

Enfin, pour les variables `Vaccinated`, `Dewormed` et `Sterilized`, les réponses *No* (orange) et *Yes* (vert) sont les plus représentées, tandis que *Not Sure* (bleu) est moins fréquente.

Cette homogénéité des répartitions rend difficile l'identification de variables discriminantes permettant de bien séparer les classes de `AdoptionSpeed`. Aucune des variables observées ne présente une tendance distincte ou marquée entre les classes, ce qui peut expliquer la difficulté rencontrée par les modèles pour prédire efficacement la vitesse d'adoption.

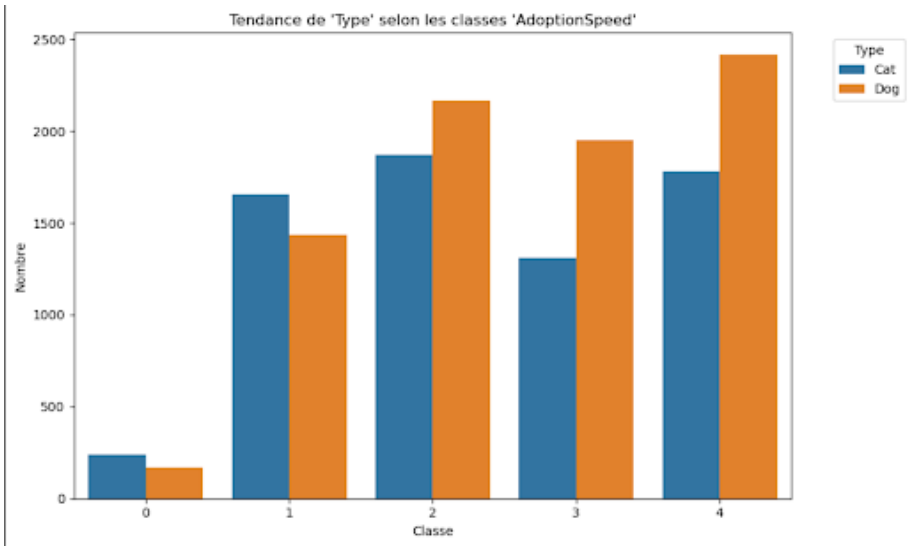


FIGURE 10 – Tendance de la variable `Type` selon les classes d'`AdoptionSpeed`

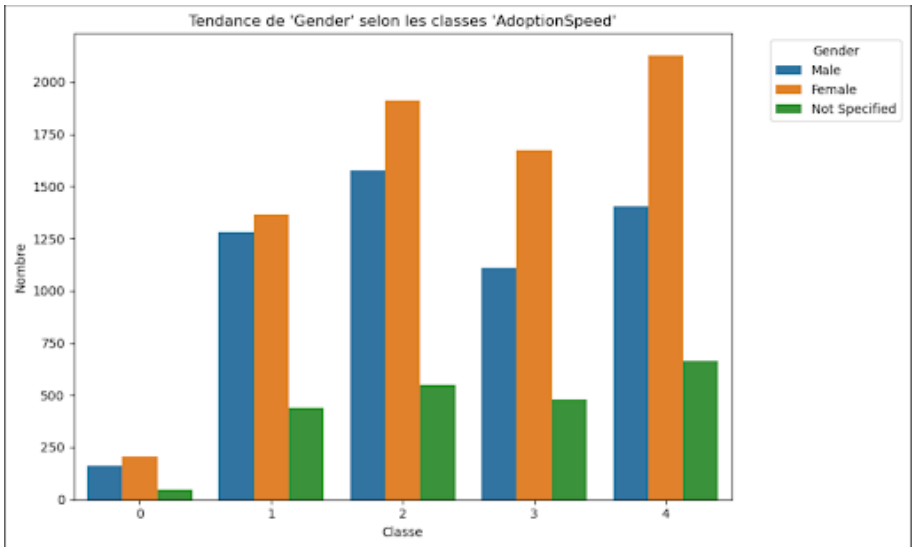


FIGURE 11 – Tendance de la variable `Gender` selon les classes d'`AdoptionSpeed`

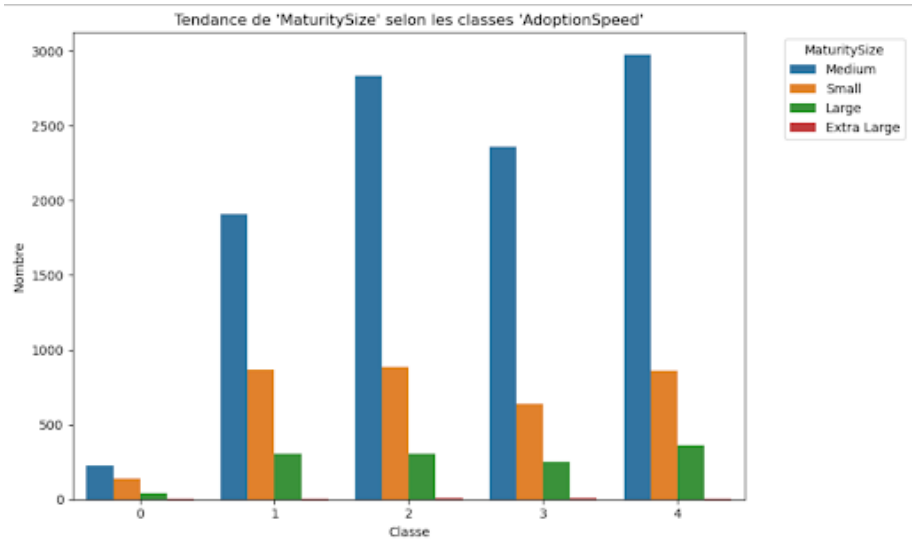


FIGURE 12 – Tendance de la variable `MaturitySize` selon les classes d'`AdoptionSpeed`

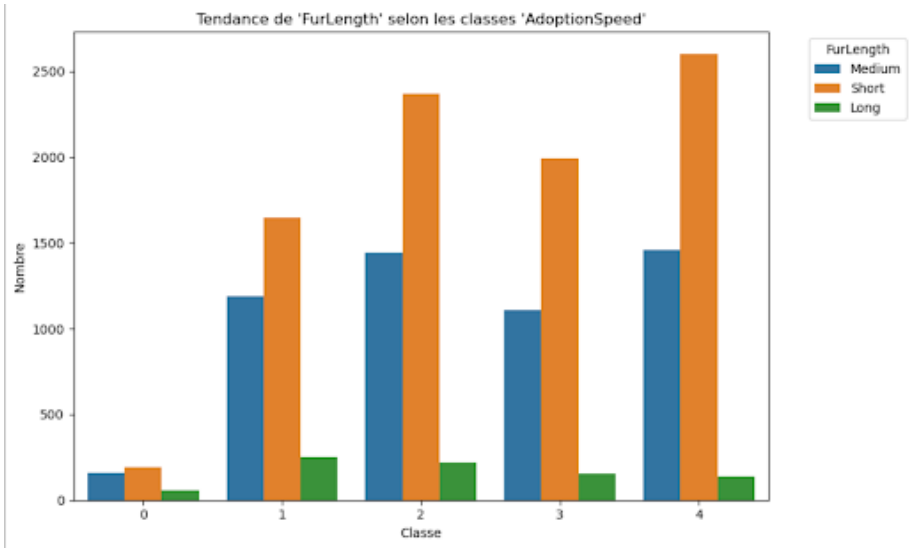


FIGURE 13 – Tendance de la variable `FurLength` selon les classes d'`AdoptionSpeed`

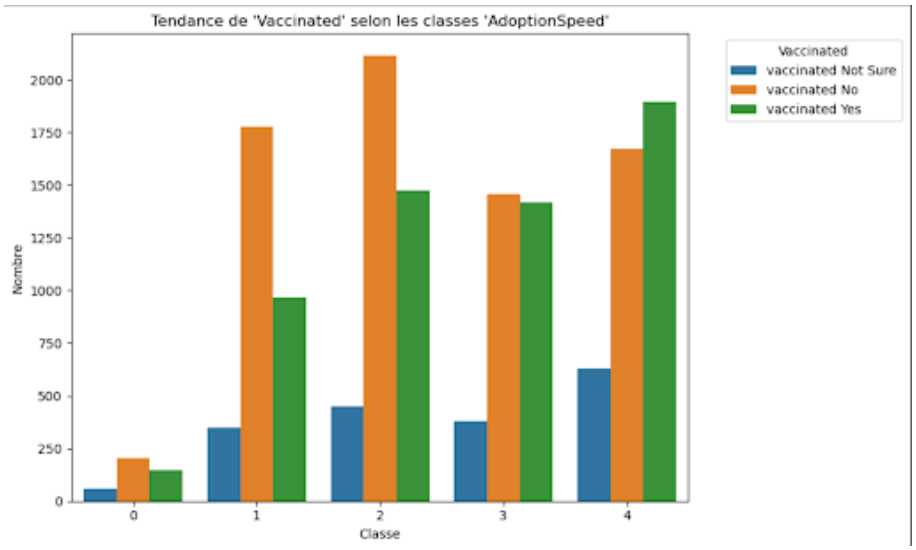


FIGURE 14 – Tendance de la variable `Vaccinated` selon les classes d'`AdoptionSpeed`

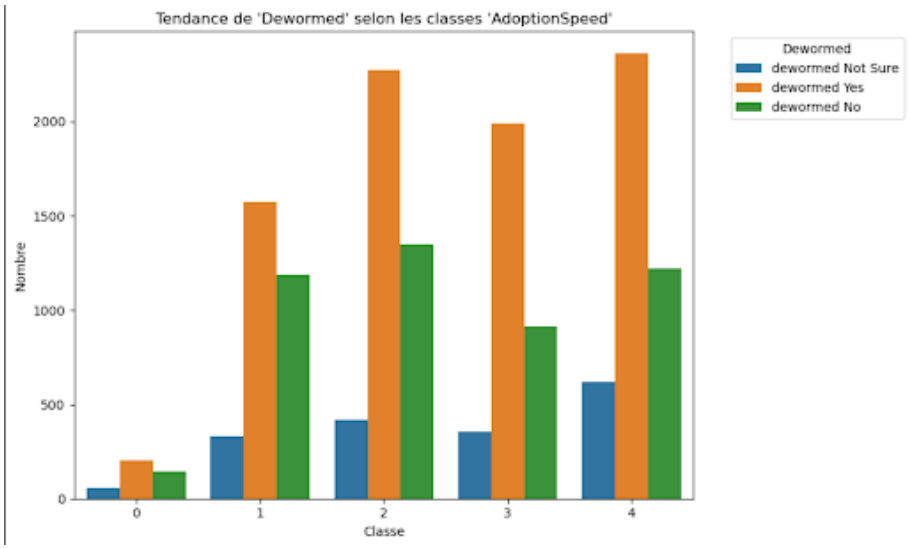


FIGURE 15 – Tendance de la variable `Dewormed` selon les classes d'`AdoptionSpeed`

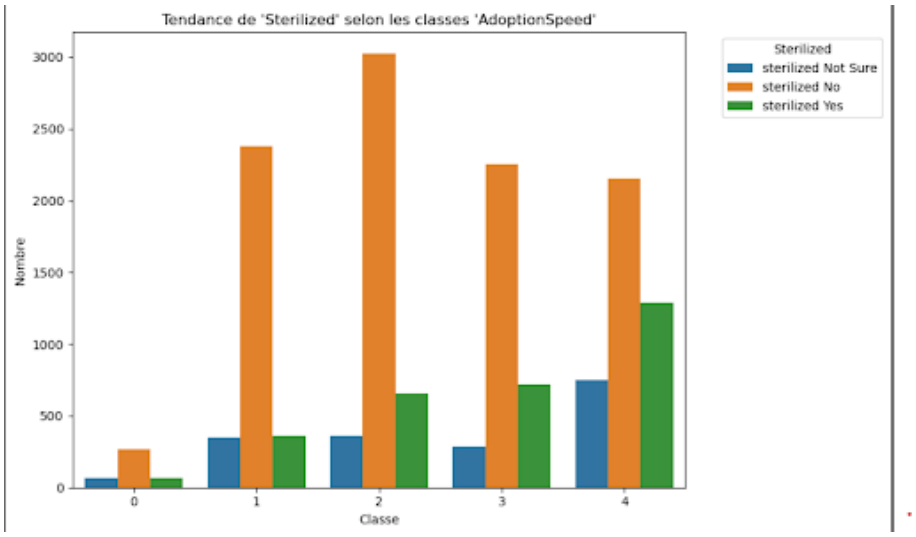


FIGURE 16 – Tendance de la variable `Sterilized` selon les classes d'`AdoptionSpeed` (graphe 1)

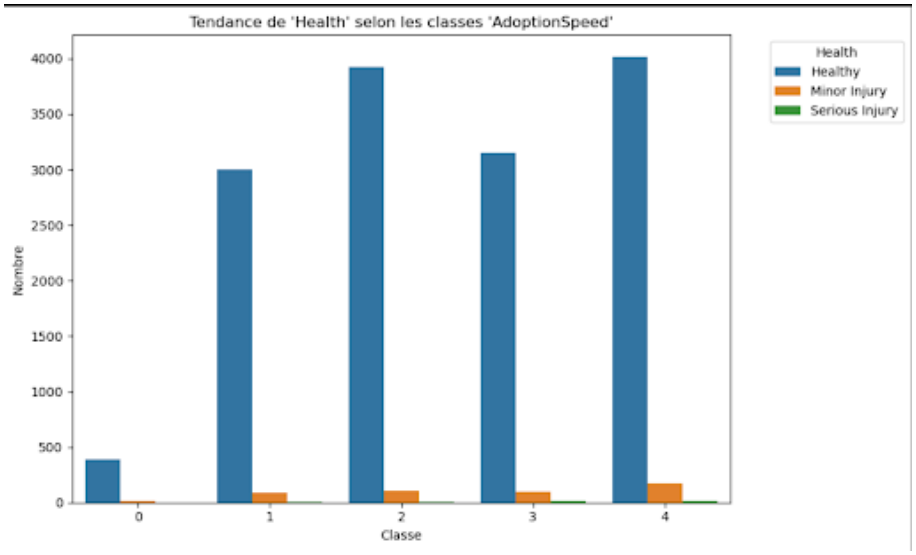


FIGURE 17 – Tendance de la variable `Health` selon les classes d'`AdoptionSpeed`

5.8 Analyse en Composantes Multiples (ACM)

Afin de mieux comprendre les relations entre les variables qualitatives du jeu de données, une Analyse en Composantes Multiples (ACM) a été réalisée. Cette méthode permet de visualiser la structure des données catégorielles et d'identifier d'éventuelles associations entre les modalités des différentes variables.

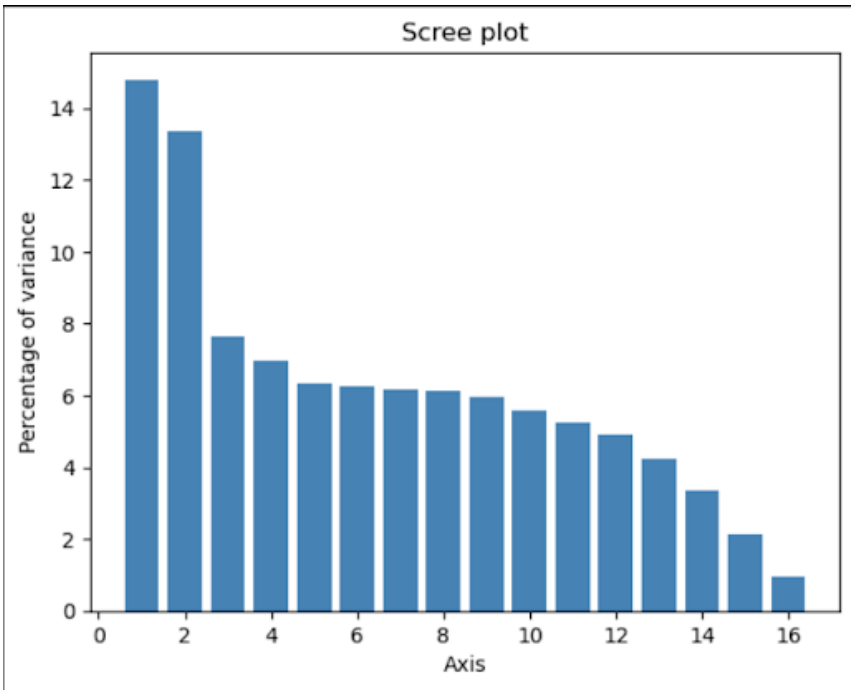


FIGURE 18 – Répartition de la variance expliquée par chaque axe de l'ACM (Scree plot)

Le premier graphe (Figure 18) montre la répartition de la variance expliquée par chaque axe. On observe que la première dimension (Dim 1) explique 14,8% de la variance totale, et la deuxième (Dim 2) 13,35%. Ensemble, ces deux axes capturent environ 28% de l'information globale, ce qui est suffisant pour une visualisation initiale. La décroissance progressive de la variance suggère que les premières dimensions concentrent l'essentiel des variations dans les données.

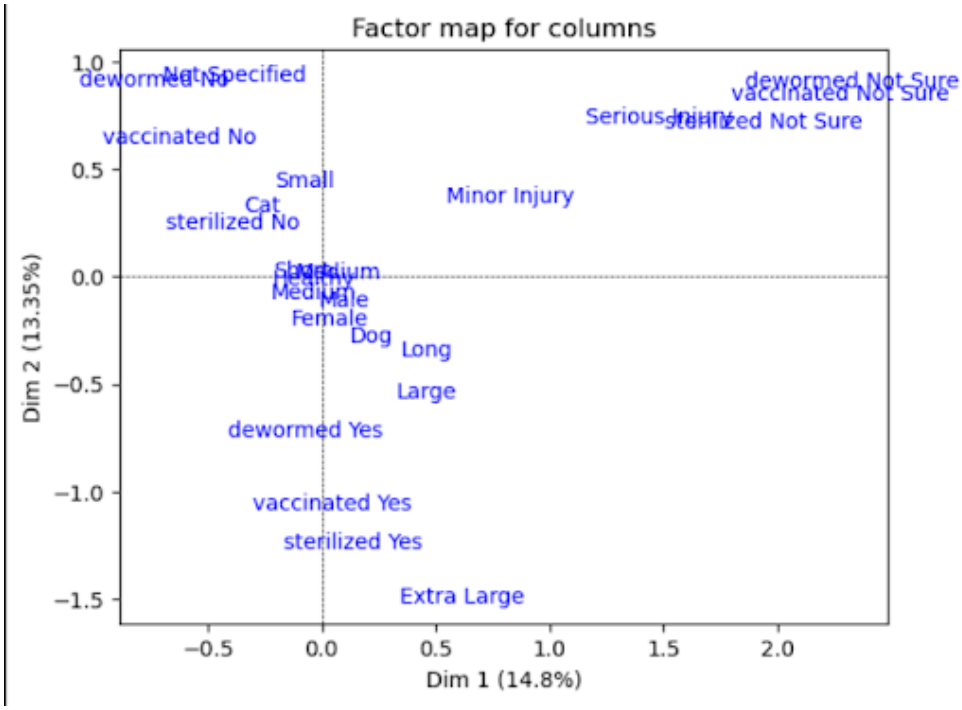


FIGURE 19 – Carte factorielle des modalités (ACM)

La carte factorielle des modalités (Figure 19) illustre les liens entre les modalités des variables catégorielles. Certaines associations apparaissent clairement :

— Les modalités *vaccinated YES*, *dewormed YES*, et *sterilized YES* sont regroupées dans le quadrant inférieur gauche, montrant une forte corrélation entre ces traitements sanitaires.

— Les modalités opposées *vaccinated NO*, *dewormed NO*, et *sterilized NO* apparaissent ensemble dans le quadrant supérieur gauche.

— Un groupe particulier est formé par les modalités *vaccinated NOT SURE*, *dewormed NOT SURE*, et *sterilized NOT SURE*, dans le quadrant supérieur droit.

— Les modalités liées au genre (*Male* et *Female*) et à la taille (*Medium*, *Small*, *Large*, *Extra Large*) sont proches de l'origine. Cela indique une faible contribution discriminante sur les deux premiers axes.

Ces observations confirment les résultats des histogrammes présentés précédemment : les tendances sont globalement homogènes entre les classes d'*AdoptionSpeed*, et aucune variable catégorielle n'émerge clairement comme discriminante. Cela illustre la complexité du problème de classification et justifie le recours à des regroupements de classes pour faciliter l'apprentissage.

5.9 Analyse des clusters : vers une réduction du nombre de classes

Dans le cadre de notre étude, une analyse de clustering non supervisé a été réalisée afin d'évaluer la cohérence des classes d'adoption existantes dans les données. Pour cela, nous avons utilisé l'algorithme *K-Means*, appliqué aux données préalablement standardisées. Une réduction de dimension par **ACM** a été effectuée pour projeter les observations sur deux axes principaux, notés *Composante principale 1* et *Composante principale 2*. Ces deux axes représentent les directions dans lesquelles les données varient le plus, ce qui permet de mieux visualiser la structure des groupes.

Deux configurations ont été testées : un clustering en **5 groupes** et un clustering en **2 groupes**.

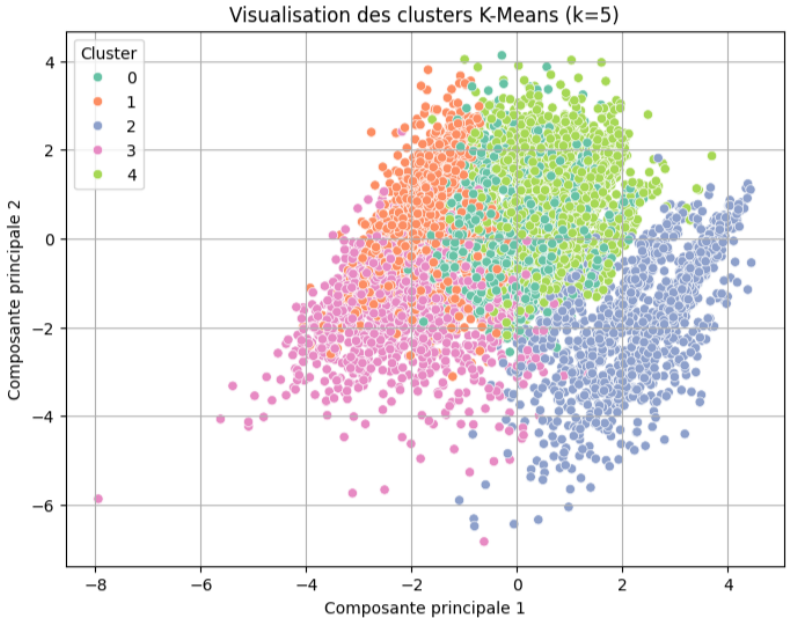


FIGURE 20 – Clustering KMeans avec $k = 5$

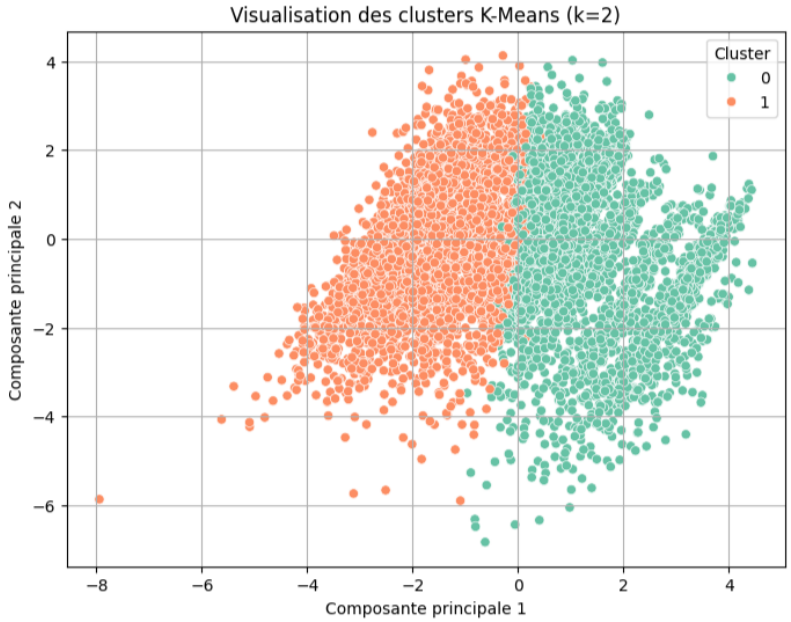


FIGURE 21 – Clustering KMeans avec $k = 2$

Le graphe avec $k = 5$ montre un chevauchement important entre les groupes, ce qui suggère que la distinction entre les 5 catégories d'adoption n'est pas clairement définie dans l'espace des données. À l'inverse, la représentation avec $k = 2$ montre des groupes bien séparés, suggérant qu'un découpage en **deux catégories globales** serait plus pertinent. Cela renforce l'idée que le problème de prédiction est sans doute mal posé avec 5 classes trop proches ou arbitraires, et qu'un regroupement permettrait une classification plus robuste.

5.10 Comparaison de regroupements de classes avec Random Forest

Afin d'optimiser la prédiction de la vitesse d'adoption des animaux, nous avons exploré deux regroupements binaires alternatifs à la classification initiale en 5 classes. L'objectif est de simplifier le problème tout en conservant une interprétabilité pertinente.

Les deux combinaisons testées sont :

— **Combinaison 1** : `[0,1,2]` vs `[3,4]` — où `0` représente une adoption rapide ou modérée, et `1` une adoption lente ou non adoption.

— **Combinaison 2** : `[1,2]` vs `[3,4]` — où la classe `0` (adoption le jour même) est exclue.

Ces regroupements ont été évalués à l'aide d'un modèle **Random Forest** avec une validation croisée à 5 plis, et en mesurant plusieurs métriques : *accuracy*, *precision*, *recall*, *F1-score*, mais aussi le biais, la variance et l'erreur quadratique moyenne (MSE).

Regroupement	Accuracy	Précision	Recall	F1-score	MSE	Biais	Variance
`[0,1,2]` vs `[3,4]`	62.7%	62.7%	62.7%	62.7%	0.376	0.274	0.101
`[1,2]` vs `[3,4]`	61.4%	61.5%	61.4%	61.5%	0.385	0.284	0.101

TABLE 1 – Comparaison des performances entre deux regroupements de classes avec Random Forest

On observe que la première combinaison `[0,1,2]` vs `[3,4]` donne de meilleures performances globales, avec un F1-score de **62.7%**, une *erreur quadratique moyenne (MSE)* plus faible, ainsi qu'un *biais* plus bas. Cela suggère qu'inclure la classe 0 (adoption immédiate) dans les cas d'adoption rapide/modérée améliore la robustesse du modèle.

Ce résultat conforte l'idée que le reclassement judicieux de la variable cible peut améliorer les performances prédictives, tout en gardant du sens dans l'interprétation métier.

5.10.1 Optimisation des performances avec GridSearchCV

Afin d'améliorer les performances du modèle Random Forest, une recherche d'hyperparamètres a été réalisée à l'aide de `GridSearchCV`, en explorant plusieurs combinaisons de paramètres tels que `max_depth`, `n_estimators`, `min_samples_split`, `min_samples_leaf`, et `max_features`. Deux regroupements de classes ont été testés : **(0,1,2) vs (3,4)** et **(1,2) vs (3,4)**.

Les meilleurs paramètres trouvés par GridSearch pour les deux cas sont :

— `max_depth=10`, `n_estimators=200`, `min_samples_split=5`, `min_samples_leaf=2`, `max_features='sqrt'`

Résultats pour le regroupement (0,1,2) vs (3,4) :

— Accuracy : 65.07%
— Précision : 65.09%
— Rappel : 65.07%
— F1-score : 65.05%
— MSE : 0.3553
— Biais : 0.2948
— Variance : 0.0605
— Moyenne de validation croisée : 64.6%

Résultats pour le regroupement (1,2) vs (3,4) :

- Accuracy : 65.07%
- Précision : 65.09%
- Rappel : 65.07%
- F1-score : 65.05%
- MSE : 0.3553
- Biais : 0.2948
- Variance : 0.0605
- Moyenne de validation croisée : 64.6%

Les deux regroupements donnent des résultats quasiment identiques après optimisation, confirmant la robustesse du modèle Random Forest. Toutefois, on remarque que le gain en performance reste marginal après optimisation, suggérant qu'une approche plus fine comme l'analyse de nouvelles variables ou le test d'autres algorithmes pourrait être plus prometteuse pour améliorer les performances globales.

5.11 Comparaison des résultats avec CatBoost sur différents regroupements de classes

Pour approfondir l'analyse des regroupements de classes, nous avons évalué les performances du modèle **CatBoost** sur deux combinaisons binaires :

- Regroupement des classes **(0+1+2)** contre **(3+4)**.
- Regroupement des classes **(1+2)** contre **(3+4)**.

1. Regroupement (0+1+2) vs (3+4) : Dans cette configuration, le modèle CatBoost a montré des résultats globalement satisfaisants. Le **MSE est de 0.3498**, accompagné d'un **biais de 0.2741** et d'une **variance de 0.0758**. La précision globale atteint **66.16%**, tout comme le rappel et le f1-score, ce qui montre une bonne stabilité du modèle et une performance équilibrée entre les deux classes. La matrice de confusion indique que la majorité des exemples sont bien prédits, avec une légère confusion entre les classes.

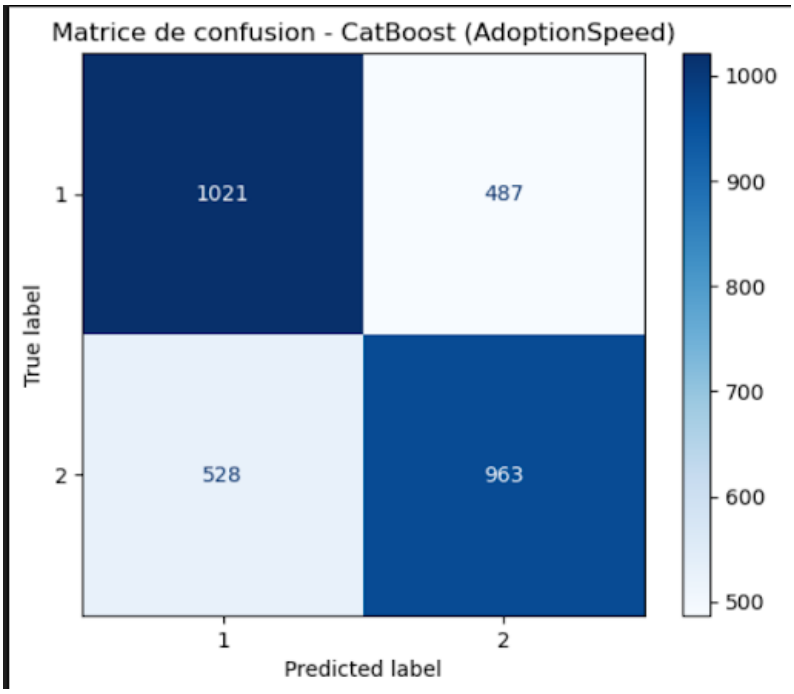


FIGURE 22 – Matrice de confusion - CatBoost (regroupement 0+1+2 vs 3+4)

metric	précision	Rappelle	f1_score	MSE	Biaise	Variance
résultats	0.6527	0.6527	0.6527	0.3600	0.2843	0.0757

FIGURE 23 – Tableau des scores - CatBoost (regroupement 0+1+2 vs 3+4)

2. Regroupement (1+2) vs (3+4) : Ce deuxième regroupement a également été testé afin de voir si l'exclusion de la classe 0 (rare) pouvait améliorer la performance. Le modèle atteint une **précision de 65.27%**, un rappel et un f1-score identiques, ainsi qu'un **MSE de 0.3600**, un biais de 0.2843 et une variance de 0.0757. Les résultats sont légèrement inférieurs à ceux du regroupement précédent, ce qui indique que l'exclusion de la classe 0 n'a pas permis de gains significatifs.

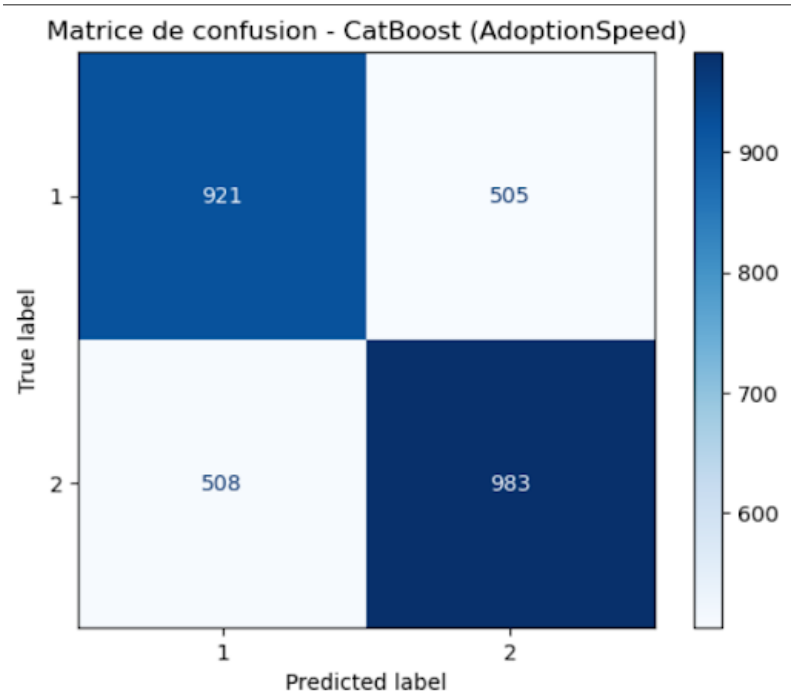


FIGURE 24 – Matrice de confusion - CatBoost (regroupement 1+2 vs 3+4)

metric	précision	Rappelle	f1_score	MSE	Biaise	Variance
résultats	0.6527	0.6527	0.6527	0.3600	0.2843	0.0757

FIGURE 25 – Tableau des scores - CatBoost (regroupement 1+2 vs 3+4)

Conclusion : Ces résultats montrent que CatBoost parvient à modéliser correctement les regroupements binaires, avec une stabilité des prédictions (variance faible) et un compromis intéressant entre biais et variance. Le regroupement **(0+1+2) vs (3+4)** donne des performances légèrement supérieures.

5.12 Comparaison des combinaisons avec Gradient Boosting

5.12.1 Entraînement du modèle Gradient Boosting — combinaison (0,1,2) vs (3,4)

Pour améliorer la performance du modèle, nous avons fusionné les classes 0, 1 et 2 en une classe 0, et les classes 3 et 4 en une classe 1. Cette transformation vise à simplifier le problème en une tâche de classification binaire mieux équilibrée.

Les résultats obtenus montrent un **MSE** de **0.3606**, un biais modéré de **0.2980**, et une variance faible de **0.0626**, indiquant une bonne stabilité des prédictions et une capacité de généralisation convenable. La précision moyenne en validation croisée (macro) atteint **64.83 %** et l'**accuracy globale** est de **64.55 %**, des valeurs satisfaisantes bien qu'améliorables.

Rapport de classification :

	precision	recall	f1-score	support
0	0.64	0.69	0.66	2190
1	0.65	0.60	0.63	2143
accuracy			0.65	4333
macro avg	0.65	0.65	0.64	4333
weighted avg	0.65	0.65	0.64	4333

FIGURE 26 – Rapport de classification — Gradient Boosting (0,1,2) vs (3,4)

Le rapport de classification révèle que pour la classe 0, la précision est de 0.64, le rappel de 0.69 et le f1-score de 0.66, tandis que pour la classe 1, la précision est légèrement meilleure à 0.65, mais avec un rappel plus faible de 0.60, donnant un f1-score de 0.63. Cela montre que la classe 0 est mieux reconnue, mais la classe 1 reste plus difficile à identifier correctement.

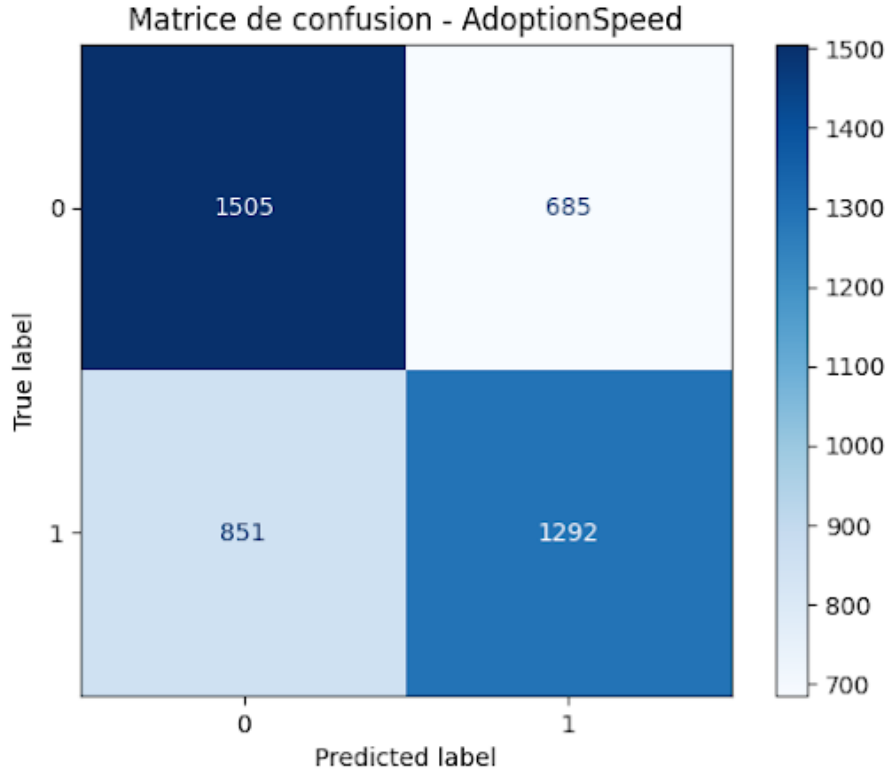


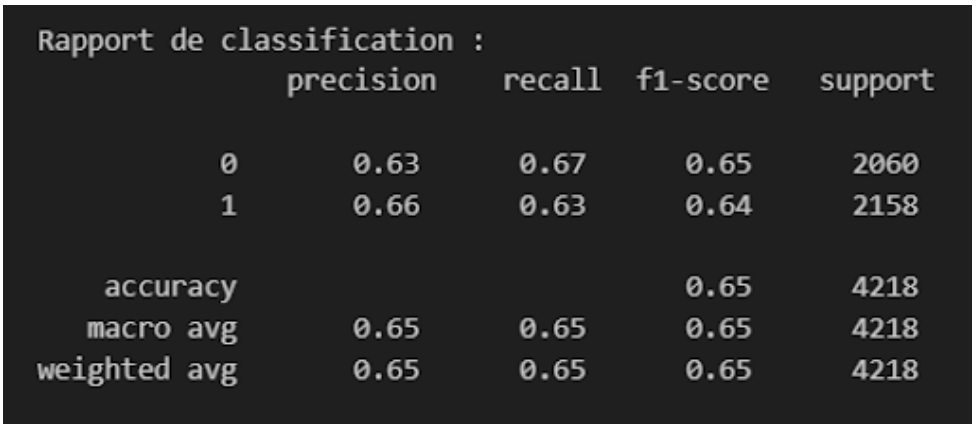
FIGURE 27 – Matrice de confusion — Gradient Boosting (0,1,2) vs (3,4)

La matrice de confusion indique que **1505 exemples de la classe 0** et **1292 exemples de la classe 1** sont bien prédits, mais respectivement **685 exemples de la classe 0** et **851 de la classe 1** sont mal classés. Globalement, le **recall** est de **60.29 %**, et la **précision globale** atteint **65.35 %**, soulignant une confiance correcte dans les prédictions, bien qu'elle se fasse au détriment du rappel.

5.12.2 Entraînement du modèle Gradient Boosting — combinaison (1,2) vs (3,4)

Dans cette seconde approche, la classe 0, très minoritaire, a été supprimée pour réduire le déséquilibre. Les classes 1 et 2 ont été regroupées sous l'étiquette 0, et les classes 3 et 4 sous l'étiquette 1, transformant ainsi le problème en une classification binaire plus équilibrée.

Cette approche a conduit à des résultats prometteurs : le **score de précision macro moyen** atteint **0.6516**, indiquant une nette amélioration par rapport aux modèles multiclasse précédents. Le **MSE**, à **0.3605**, montre une stabilité accrue des prédictions, tandis que le **biais** (**0.2994**) reste modéré et la **variance** faible (**0.0611**), signe d'une bonne généralisation sans surajustement.



Rapport de classification :

	precision	recall	f1-score	support
0	0.63	0.67	0.65	2060
1	0.66	0.63	0.64	2158
accuracy			0.65	4218
macro avg	0.65	0.65	0.65	4218
weighted avg	0.65	0.65	0.65	4218

FIGURE 28 – Rapport de classification — Gradient Boosting (1,2) vs (3,4)

Le rapport de classification reflète un bon équilibre entre les classes : la classe 0 atteint une précision de 0.63 et un rappel de 0.67, tandis que la classe 1 affiche une précision de 0.66 et un rappel de 0.63, avec des f1-scores similaires (**0.64-0.65**), prouvant l'absence de biais en faveur d'une classe.

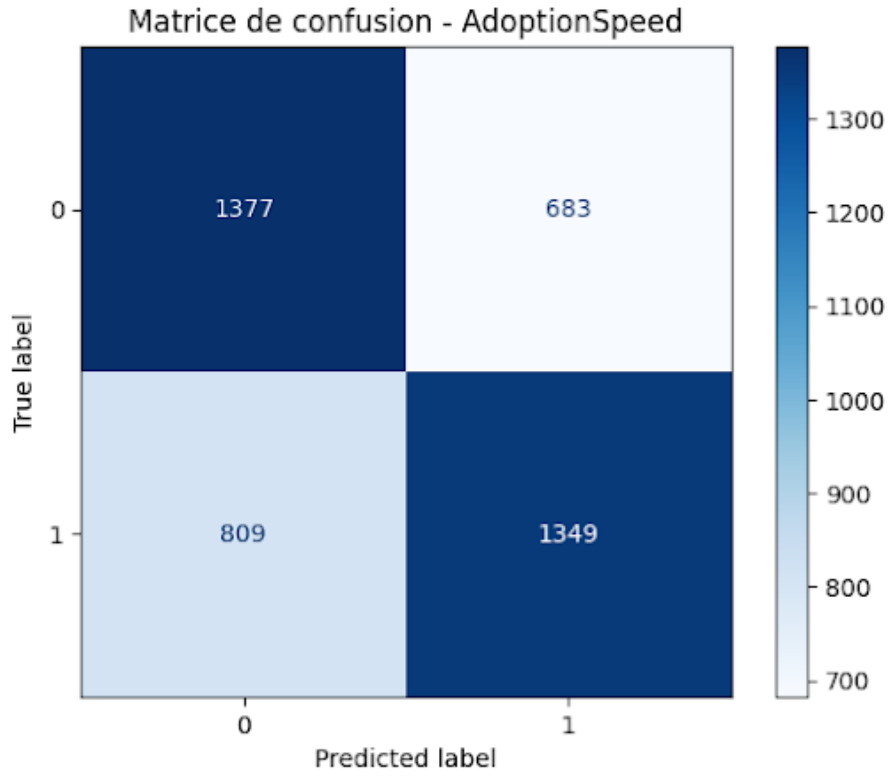


FIGURE 29 – Matrice de confusion — Gradient Boosting (1,2) vs (3,4)

La matrice de confusion montre une classification correcte pour la majorité des exemples : **1377 exemples sur 2060** pour la classe 0 et **1349 sur 2158** pour la classe 1. Enfin, le modèle atteint une **précision globale de 66.39 %**, un **rappel global de 62.51 %** et une **accuracy de 64.63 %**.

5.12.3 Comparaison des deux combinaisons avec Gradient Boosting

Afin d'identifier la meilleure manière de regrouper les classes d'adoption, nous avons comparé deux approches de fusion :

— **Fusion (0,1,2)** contre **(3,4)**

— **Fusion (1,2)** contre **(3,4)**

Les résultats obtenus à l'issue des expérimentations avec l'algorithme Gradient Boosting sont résumés dans le tableau ci-dessous :

Critères	Fusion (0,1,2) & (3,4)	Fusion (1,2) & (3,4)
MSE	0.3606	0.3605
Biais	0.2980	0.2994
Variance	0.0626	0.0611
Précision moyenne (macro)	64.83 %	65.16 %
Accuracy globale	64.55 %	64.63 %
Précision Classe 0	0.64	0.63
Rappel Classe 0	0.69	0.67
F1-Score Classe 0	0.66	0.64
Précision Classe 1	0.65	0.66
Rappel Classe 1	0.60	0.63
F1-Score Classe 1	0.63	0.65
Précision globale	65.35 %	66.39 %
Rappel global	60.29 %	62.51 %

FIGURE 30 – Comparaison des performances entre les deux fusions de classes

La **fusion (0,1,2) & (3,4)** présente un biais légèrement inférieur (0.2980 contre 0.2994), ce qui pourrait indiquer un risque légèrement réduit de sous-apprentissage. Cependant, la variance est un peu plus élevée (0.0626 contre 0.0611), suggérant une stabilité des prédictions légèrement moindre et un risque un peu plus élevé de sur-apprentissage par rapport à l'approche **(1,2) & (3,4)**.

En termes de prédictions, la **fusion (1,2) & (3,4)** offre un équilibre supérieur entre précision et rappel pour les deux classes, avec des f1-scores légèrement meilleurs et une précision globale plus élevée (66.39 % contre 65.35 %), ce qui montre une performance générale légèrement meilleure.

Conclusion : La fusion **(1,2) & (3,4)** s'avère être la meilleure option, offrant une

meilleure généralisation et un équilibre supérieur entre le sous-apprentissage et le surapprentissage.

5.13 Ajout des sentiments textuels dans le Gradient Boosting

Après avoir fusionné les classes **1 et 2** en une classe **0**, et les classes **3 et 4** en une classe **1**, nous avons enrichi notre modèle en y intégrant des variables dérivées de l'analyse textuelle des descriptions des animaux. Ces nouvelles variables correspondent aux scores de sentiment *positif*, *négatif* et *neutre* extraits des textes descriptifs.

Le modèle Gradient Boosting, réentraîné avec ces nouvelles variables, a obtenu les performances suivantes :

— **Précision moyenne (macro)** en validation croisée : 64.71 %
— **Accuracy globale** : 64.25 %
— **MSE** : 0.3627
— **Biais** : 0.2894
— **Variance** : 0.0733

Ces résultats témoignent d'une **stabilité générale du modèle**, avec une faible variance, signe d'un surapprentissage limité.

Rapport de classification :

	precision	recall	f1-score	support
0	0.61	0.67	0.64	2085
1	0.67	0.62	0.64	2290
accuracy			0.64	4375
macro avg	0.64	0.64	0.64	4375
weighted avg	0.65	0.64	0.64	4375

FIGURE 31 – Rapport de classification avec ajout des variables de sentiment

Sur le plan des classes :

— **Classe 0** : Précision = 0.61, Rappel = 0.67, F1-score = 0.64
— **Classe 1** : Précision = 0.67, Rappel = 0.62, F1-score = 0.64

Ces scores sont relativement équilibrés, montrant une reconnaissance modérée mais stable des deux classes.

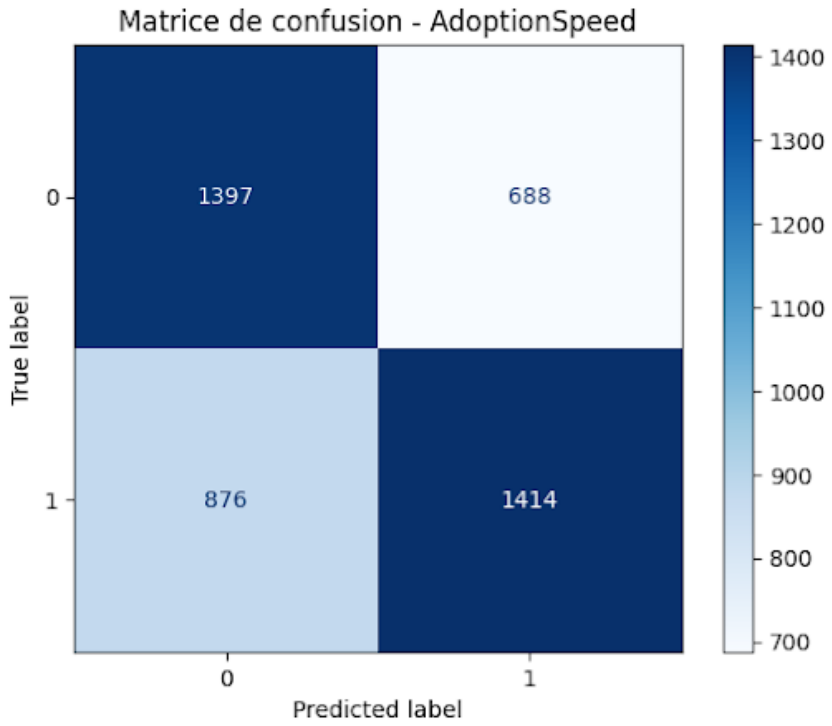


FIGURE 32 – Matrice de confusion avec ajout des variables de sentiment

La matrice de confusion met en évidence des erreurs notables :
— **688** exemples de la classe 0 sont prédit comme appartenant à la classe 1
— **876** exemples de la classe 1 sont mal classés en classe 0

Malgré ces confusions importantes, la **précision globale augmente légèrement à 67.27 %**, ce qui suggère que les variables liées au *sentiment* apportent un signal pertinent, bien que l'amélioration reste modeste.

5.14 Optimisation du Gradient Boosting avec les sentiments (GridSearchCV)

Après optimisation des hyperparamètres du modèle Gradient Boosting à l'aide de `GridSearchCV`, les meilleurs paramètres retenus sont :
— `learning_rate` = 0.01
— `max_depth` = 7
— `n_estimators` = 300
— `subsample` = 0.8

L'objectif était d'améliorer les performances du modèle enrichi par les variables de sentiment, tout en conservant une bonne capacité de généralisation.

Les résultats obtenus après entraînement avec ces paramètres sont :
— **Accuracy** : 64.00 %
— **MSE** : 0.3607
— **Biais** : 0.2875
— **Variance** : 0.0732

Ces métriques traduisent un bon équilibre entre biais et variance, sans réel gain par rapport au modèle précédent sans optimisation.

Rapport de classification :

	precision	recall	f1-score	support
0	0.61	0.69	0.65	2085
1	0.68	0.59	0.63	2290
accuracy			0.64	4375
macro avg	0.64	0.64	0.64	4375
weighted avg	0.64	0.64	0.64	4375

FIGURE 33 – Rapport de classification – Gradient Boosting avec sentiments (optimisé)

D'après le rapport de classification :
— **Classe 0** : Précision = 0.61, Rappel = 0.69, F1-score = 0.65
— **Classe 1** : Précision = 0.68, Rappel = 0.59, F1-score = 0.63
On observe une répartition relativement équilibrée entre les deux classes, bien que la classe 1 présente un rappel plus faible.

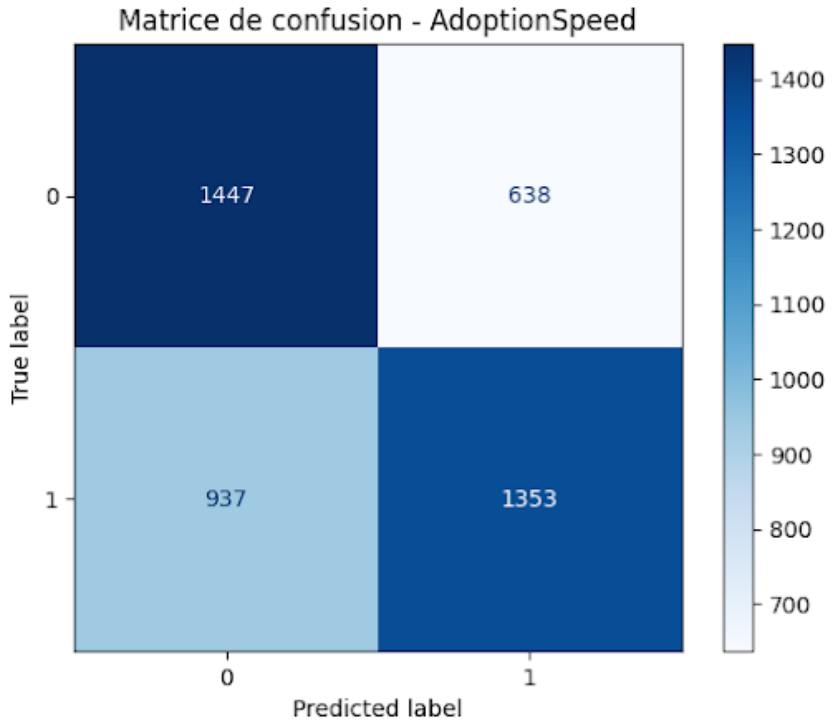


FIGURE 34 – Matrice de confusion – Gradient Boosting avec sentiments (optimisé)

La matrice de confusion montre :
— **Classe 0** : 1447 exemples bien classés, 638 confondus avec la classe 1
— **Classe 1** : 1353 bien classés, 937 mal prédits
Globalement, la **précision atteint 67.96 %**, mais le **rappel global diminue à 59.08 %**, traduisant une meilleure confiance dans les prédictions, mais au prix d'un taux d'erreurs de rappel plus élevé.

Conclusion : comparé au modèle sans GridSearchCV, on note une légère amélioration

de la précision globale, mais une baisse du rappel, suggérant que le modèle devient plus conservateur dans ses prédictions.