UNIVERSITÉ RENNES 2

PRÉFET DE LA RÉGION BRETAGNE
Liberté
Égalité
Fraternité

Rapport d'alternance

Guibert Marie
Master Mathématiques Appliquées, Statistique
Parcours Science des Données, Intelligence Artificielle
Université Rennes 2, Université de Rennes, Institut Agro, ENSAI
Promotion 2023/2024
Tuteur universitaire : Tavenard Romain

Secrétariat Général aux Affaires Régionales de la région Bretagne
3 Rue Martenot
35000 Rennes
Maître d'alternance : Kounowski Julien
Chargé de mission innovation publique, modernisation et développement des usages numériques et administrateur des données - référent cybersécurité

Remerciements

Je tiens à exprimer ma profonde gratitude à toutes les personnes qui ont contribué, de près ou de loin à la réalisation de ce travail ainsi qu'à mon développement professionnel et personnel durant cette année.
Je souhaite adresser mes remerciements les plus sincères à M. Julien KOUNOWSKI et M. Fabrice PHUNG pour leur patience et leur expertise. Leurs conseils avisés et leurs encouragements m'ont été d'une grande aide pour mener à bien les différents projets. Leurs critiques constructives m'ont permis de mieux comprendre les implications de l'utilisation des données et les conséquences potentielles de certains projets dans le secteur public.
Je remercie également M. Romain TAVENARD pour son soutien et ses conseils pertinents tout au long de mon alternance.
Enfin, je remercie toute l'équipe du projet Data Etat pour leur accueil chaleureux et leur coopération, qui ont contribué à créer un cadre de travail stimulant et agréable.

Sommaire

1 Introduction

Dans le cadre de mon parcours universitaire en Master de Mathématiques Appliquées et Statistiques, j'ai eu l'opportunité d'entrer dans le monde professionnel de la donnée au travers d'une alternance. Cette période a été l'occasion de mettre en pratique les connaissances acquises tout au long de mes études et de contribuer activement au sein d'une structure évoluant dans le domaine de la gestion et de l'analyse des données.

Ce rapport retrace mon parcours durant cette année d'alternance au sein du Secrétariat Général des Affaires Régionales (SGAR). Il mettra principalement en lumière un projet visant à comprendre la conduite des communes suite à la mise en place de politiques publiques. Aussi, il présentera différentes missions qui m'ont été confiées, les compétences développées, ainsi que les enseignements tirés de cette immersion professionnelle. À travers cette expérience, j'ai pu comprendre les enjeux de la donnée dans le domaine public et participer aux développements de solutions pour rendre celle-ci accessible et compréhensible par tous.

L'objectif principal de mon alternance était d'appliquer mes connaissances théoriques dans un contexte professionnel et de pouvoir participer aux missions d'une entreprise ou d'une administration. L'intégration au sein du SGAR m'a offert une vision plus centrée sur la mise en oeuvre des politiques publiques, différente de l'approche universitaire.

1.1 Mon environnement d'alternance

Lors de cette année universitaire, j'ai intégré le SGAR, service interministériel à dimension régionale du Préfet de Région Bretagne, en tant que Data Analyst/Scientist. Grâce à la diversité et à la richesse de mon alternance, je suis en lien avec plusieurs acteurs déterminants pour le développement du territoire : le SGAR, le GIP (Groupement d'Intérêt Public) SIB, la Direction Régionale de l'Environnement, de l'Aménagement et du Logement (DREAL) et le Ti Lab, laboratoire d'innovation publique Etat - Région.

SGAR BRETAGNE
Le SGAR constitue mon lieu d'alternance principal, avec Julien KOUNOWSKI, chargé de mission innovation publique, modernisation et développement des usages numériques et administrateur des données de la région Bretagne, comme tuteur pour m'accompagner durant mon apprentissage cette année. En tant que structure interministérielle, le SGAR se compose de deux grandes fonctions sous l'autorité du préfet :

1. Une fonction transversale d'animation et de coordination interministérielle
2. Une fonction d'impulsion des actions de modernisation

Pour ma part, mon alternance s'inscrit davantage dans le second pôle évoqué, celui concernant des actions de modernisation. En effet, le domaine de la data est porteur d'innovation et de progrès offrant la possibilité de moderniser les structures déjà existantes et bien sûr, de fluidifier l'information et l'interconnaissance. Au sein de la région Bretagne, le SGAR facilite la coordination entre les différents départements lors de la mise en œuvre d'actions et de politiques publiques. Travailler au SGAR me permet de comprendre les problématiques réelles auxquelles les administrations régionales sont confrontées dans notre société actuelle. De plus, comme mentionné précédemment, rejoindre le SGAR m'a permis d'avoir une approche plus pratique et orientée métier, en lien avec le monde professionnel.

SIB
Le SIB est un acteur du numérique au service de la santé et du secteur public. En effet, c'est un GIP qui couvre les trois fonctions publiques : la fonction hospitalière, les collectivités et l'Etat.

Concernant la région Bretagne, il joue un rôle essentiel dans l'hébergement et le stockage des données. De plus, de nombreux agents sont essentiels pour effectuer des manipulations techniques concernant le monde de la donnée. En lien avec cette entité, j'ai eu la chance d'être accompagnée par Aristide LEPECULIER qui est chef de projet innovation au sein de ce groupement d'intérêt public et de développeurs. Cela m'a permis d'acquérir de nouvelles connaissances, notamment sur le plan technique, tout en développant des compétences relationnelles.

DREAL

Lors de cette année, je suis aussi en relation avec la Direction Régionale de l'Environnement, de l'Aménagement et du Logement (DREAL), principalement incarnée par Fabrice PHUNG, responsable pour l'Etat de la démarche d'open data reconnue : « GéoBretagne ». La DREAL fait partie des administrations régionales qui demeure à la pointe dans le domaine des données. Collaborer étroitement avec cette administration s'avère extrêmement bénéfique et captivant.

TI LAB

Porté par la Préfecture de Bretagne et la Région Bretagne, le Ti Lab est un laboratoire territorial. Son rôle est d'accompagner la transformation et le décloisonnement des politiques publiques, centrées sur les usagers et définies collaborativement. Au sein de ce laboratoire, ma principale responsabilité consiste à faciliter la connexion entre les individus possédant des compétences essentiellement relationnelles et axées usagers, des designers (UX, graphique, services...) avec les besoins techniques requis pour concrétiser certains projets. Dans cette structure, j'ai principalement été en lien avec Maëlys GLORO, qui est en charge de l'UX/UI Design sur diverses missions entre le SGAR et le Ti Lab.

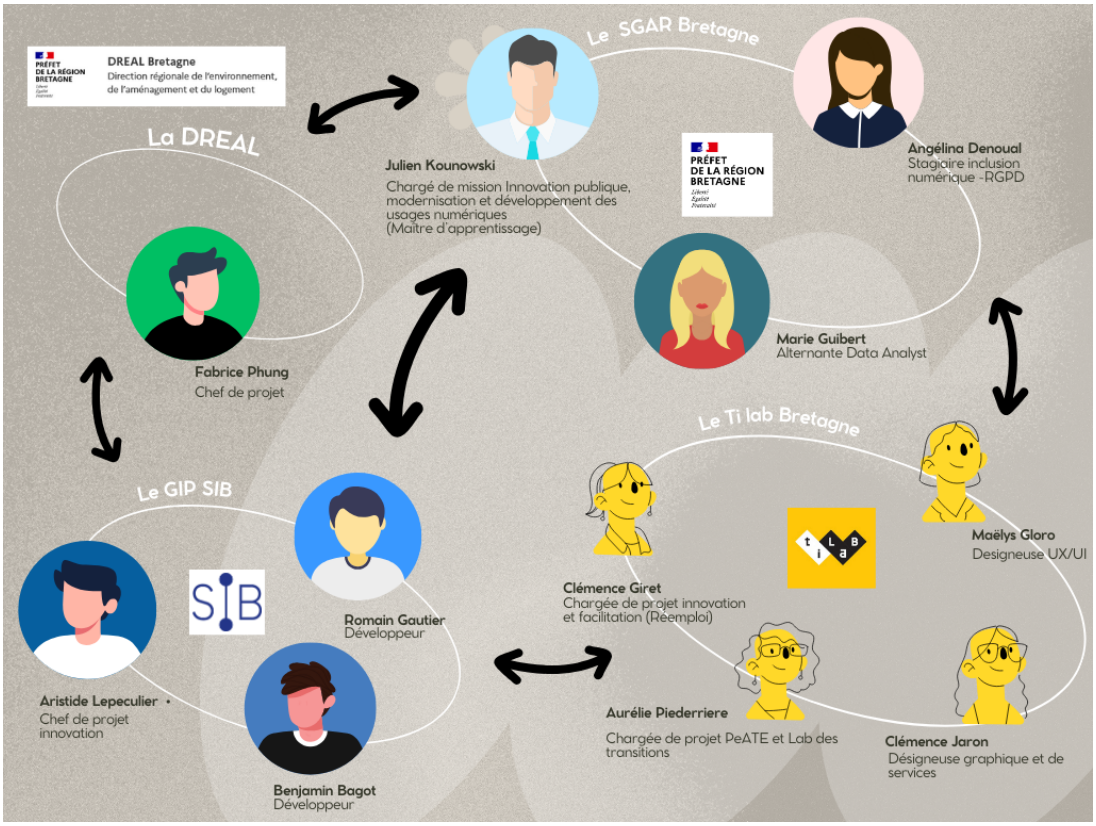


FIGURE 1 – Organigramme

1.2 Enjeux de la donnée pour ces organisations

Les enjeux pour chacune de ces organisations diffèrent selon les besoins mais les objectifs finaux restent communs : optimiser le travail des agents, renforcer la prise de décision, créer des communs numériques et mettre en oeuvre, de manière plus efficace, les politiques publiques pilotées par l'Etat. Cette action s'inscrit dans un contexte budgétaire qui vise à développer des outils pour avoir une meilleure efficience (plus efficace et rentable).
L'objectif premier du SGAR est de faciliter la coordination interministérielle en analysant les données provenant de différentes administrations et ministères.
Le SIB a un rôle clé dans cet objectif car cette organisation détient une partie des compétences techniques pour mettre en place les moyens pour rendre la donnée accessible. Aussi, il garantit le stockage et l'accessibilité aux bases de données pour pouvoir réaliser du traitement de données.
La DREAL joue aussi un rôle important car les politiques menées par cette direction régionale sont primordiales dans notre société. De plus, cette organisation présente une appétence importante pour le monde de la donnée, l'impliquant de fait dans les projets data.
Le Ti Lab étant une source d'innovation pour la région, les données peuvent aussi être utilisées dans les projets proposés au sein du laboratoire.
Aujourd'hui, mon rôle est d'assurer la coordination entre des aspects plutôt techniques (SIB) et métiers (SGAR / Ti Lab). Mon objectif est de répondre aux besoins des agents en facilitant une utilisation optimale des données dans leur quotidien.

2 Missions

2.1 Comprendre l'influence de certains facteurs sur la demande de financements des communes en Bretagne

2.1.1 Introduction au problème

La répartition des financements de l'État, particulièrement dans le domaine de la transition écologique, est un sujet délicat dans notre société. Comme précisé précédemment, au cours de mon alternance au sein du SGAR Bretagne, j'ai été en lien avec le Ti Lab et la DREAL. Grâce à ces interactions, notamment avec des personnes ayant un fort attrait pour l'écologie et l'innovation, le thème de la transition écologique s'est rapidement imposé comme le sujet central pour cette étude.

La transition écologique est aujourd'hui une priorité majeure tant au niveau national que mondial. L'urgence climatique et la nécessité de développer des pratiques durables poussent les gouvernements à investir dans des projets écologiques. Cependant, les facteurs influençant la demande de ces financements demeurent encore peu étudiés. Notre objectif est d'analyser ces facteurs d'influence pour mieux comprendre comment et pourquoi certaines communes de Bretagne demandent et reçoivent des subventions de l'État pour leurs initiatives, en faveur de la transition écologique. En menant cette étude, nous allons faire un état des lieux sur les facteurs qui influencent potentiellement ces demandes de financements et donc la répartition du budget public pour la transition écologique. Cette analyse pourrait ainsi contribuer à améliorer les politiques publiques et à encourager davantage de communes à s'engager dans des actions écologiques.

Pour mener à bien cette étude statistique, nous avons défini une problématique visant à expliquer la demande de financements des communes en Bretagne pour la transition écologique. La question centrale que nous nous sommes posée est : « **Quels sont les facteurs qui influencent les communes en Bretagne dans leurs demandes de financements concernant la transition écologique ?** ».

Pour répondre à cette question , nous avons structuré notre approche en différentes étapes :
Dans un premier temps, nous allons présenter la démarche, les outils utilisés ainsi que le contexte statistique de l'étude. Dans un second temps, nous analyserons le travail réalisé en termes de pré-traitement de données et concernant les statistiques descriptives de la base de données. Dans un troisième temps, nous détaillerons l'analyse statistique, en rappelant la théorie pour les algorithmes mis en place lors de cette étude. Enfin, nous conclurons en synthétisant les principaux résultats de notre étude et en discutant des implications des communes pour obtenir des financements envers certaines politiques publiques.
Pour répondre à notre problématique, nous commencerons par une étape de prédiction afin de déterminer la probabilité qu'une commune demande des subventions pour l'année suivante. Ces prévisions nous permettront d'analyser l'importance des différentes variables et d'identifier les caractéristiques typiques qui influencent les demandes de financements. Finalement, nous réaliserons une cartographie afin d'analyser s'il existe un effet géographique et nous établierons un tableau montrant les communes les moins susceptibles de demander des financements pour la transition écologique.

2.1.2 Démarche et contexte statistique

Données utilisées
Pour répondre à la problématique énoncée ci-dessus, nous allons mettre en place une modélisation statistique.
Les données utilisées sont issues de différentes sources et traitent de divers sujets. Notre objectif est de trouver les facteurs qui influencent les demandes de financements des communes et s'il existe un effet géographique lié à cette demande.

La variable à expliquer correspond à la demande de financement des communes. Pour construire cette variable, nous nous sommes basés sur les données CHORUS (données comptables de l'Etat) et de l'ADEME (Agence de la transition écologique) en rapport avec la transition écologique. La variable à expliquer est donc constitué avec :

1. Le programme 380 - programme financier du fonds vert (CHORUS)
2. Les données de l'ADEME

Afin de n'avoir seulement les données au niveau communal, nous avons utilisé l'outil Budget Data Etat (voir sous-section 2.2.1). Notre recherche était focalisée sur l'année 2023 et sur les financements accordés aux **communes**. Cette démarche nous a permis d'obtenir des montants de financements par communes mais notre objectif étant de déterminer la probabilité qu'une commune demande des financements, nous avons transformé cette donnée de sorte à avoir une donnée **binaire**. Les deux modalités de cette variable sont donc :

— 1 : La commune a demandé des financements en faveur de la transition écologique durant l'année 2023

— 0 : La commune n'a pas demandé de financement en 2023

Concernant les variables explicatives, les sources sont assez variées et citées en annexe. Le choix de ces variables a été subjectif car il a résulté de plusieurs échanges avec des personnes métiers. De plus, nous avons été freiné parfois par la disponibilité des données ouvertes au niveau communal. Les variables explicatives choisies traitent de différentes thématiques : l'énergie, l'urbanisme et le logement, les transports, la démographie, l'économie, l'emploi et le social. Les données utilisées sont au niveau communal et répertoriées dans le tableau situé en annexe. Pour plus de détails sur les données utilisées, on pourra se référer au tableau des données situé en annexe B.

Outils utilisés
Pour réaliser ce travail, divers logiciels et outils ont été utilisés. Premièrement, pour la collecte et le traitement de données, le logiciel R a été employé. Ensuite, j'ai construit un tableau de bord récapitulatif de mon étude avec Apache Superset. Enfin, j'ai utilisé Python concernant la partie machine learning, les prévisions, l'analyse d'importance des variables et la cartographie.

Contexte statistique
Nous allons chercher à expliquer la demande financement d'une commune, on considère donc une variable binaire avec deux valeurs : 0 ou 1. On cherche à expliquer une variable $Y \in \mathcal{Y}$ par p variables $X = (X_1, \ldots, X_p) \in R^p$, où $p = 29$ dans un problème de classification binaire / discrimination. Comme dit précédemment, dans notre cas, Y peut prendre deux valeurs : $0, 1$. Notre échantillon est composé des 1207 communes de Bretagne, soit 1207 individus que l'on considère i.i.d. $D_{1207} = (X_1, Y_1), \ldots, (X_{1207}, Y_{1207})$ de même loi que (X, Y), et nous cherchons à prédire au mieux la sortie y associée à une nouvelle observation x.

2.1.3 Pré-traitements des données & statistiques descriptives

Pré-traitements des données
Afin de résoudre notre problématique, nous avions besoin d'une base de données stable et complète. L'utilisation principale de données ouvertes provenant de sources fiables (voir sources), a facilité le traitement des données.

Tout d'abord, il a été nécessaire de constituer une base de données unifiée puisque les sources de données étaient diverses. Pour ce faire, nous avons effectué de nombreuses jointures et filtré les données sur la région Bretagne (ou ses quatre départements). Pour la variable à expliquer, nous avons sélectionné l'année 2023. Afin d'assurer la fiabilité de notre étude, nous avons repris la même année pour les variables explicatives lorsqu'elle était disponible, ou l'année la plus récente dans le cas contraire.

Les autres traitements résultaient principalement de décisions faites après discussion avec l'équipe :

- Concernant la demande de financements des communes, ou la variable à expliquer Y, nous avons effectué une transformation pour avoir la donnée sous forme binaire. Initialement, nous avions les montants accordés aux communes mais nous avons fait le choix de prédire une probabilité de demander/recevoir des financements et donc de se placer dans un contexte de classification supervisée.
- Du côté des variables explicatives, quand il était nécessaire et pour les variables continues, nous avons calculé les moyennes par habitant pour chaque commune (exemple : consommations d'électricité et de gaz par habitant). En effet, diviser les données par le nombre d'habitants permet de standardiser les informations, facilitant ainsi une analyse statistique plus précise et équitable. Cette étape nous permet de comparer les données entre elles de manière cohérente.
  Aussi, pour pouvoir comparer plus facilement les communes entre elles, nous avons normalisé certaines données comme la part d'actifs ou encore les consommations d'électricité et de gaz dans chaque secteur par exemples.
  Nous avons parfois du trancher entre une variable de comptage ou binaire. Par exemple, la variable concernant France Rénov' montrait beaucoup plus d'intérêt en sommant les actions réalisées plutôt qu'en variable binaire pour évaluer le degré d'implication d'une commune. D'un autre côté, la variable décrivant les éco-quartiers semblaient plus pertinente en variable binaire car elle semblait liée à la superficie/importance de la commune.

Statistiques descriptives
La réalisation de statistiques descriptives permet de comprendre la distribution des données, leurs caractéristiques et si certaines communes présentent des comportements atypiques. Cette étape nous a permis d'examiner les tendances générales, les distributions et les relations initiales entre les variables. Pour avoir une présentation propre, nous avons choisi de réaliser un tableau de bord à l'aide de l'outil Apache Superset, accessible via ce lien. Ce travail a été réalisé avec soin et nous avons mené une courte analyse pour comprendre concrètement notre base de données.

Cette étape nous a permis de constater que nous étions dans une situation de **données déséquilibrées**. En effet, en Bretagne en 2023, 203 communes ont demandé des financements pour la transition écologique. A contrario, 1004 communes n'ont pas fait ces démarches (voir graphique2).

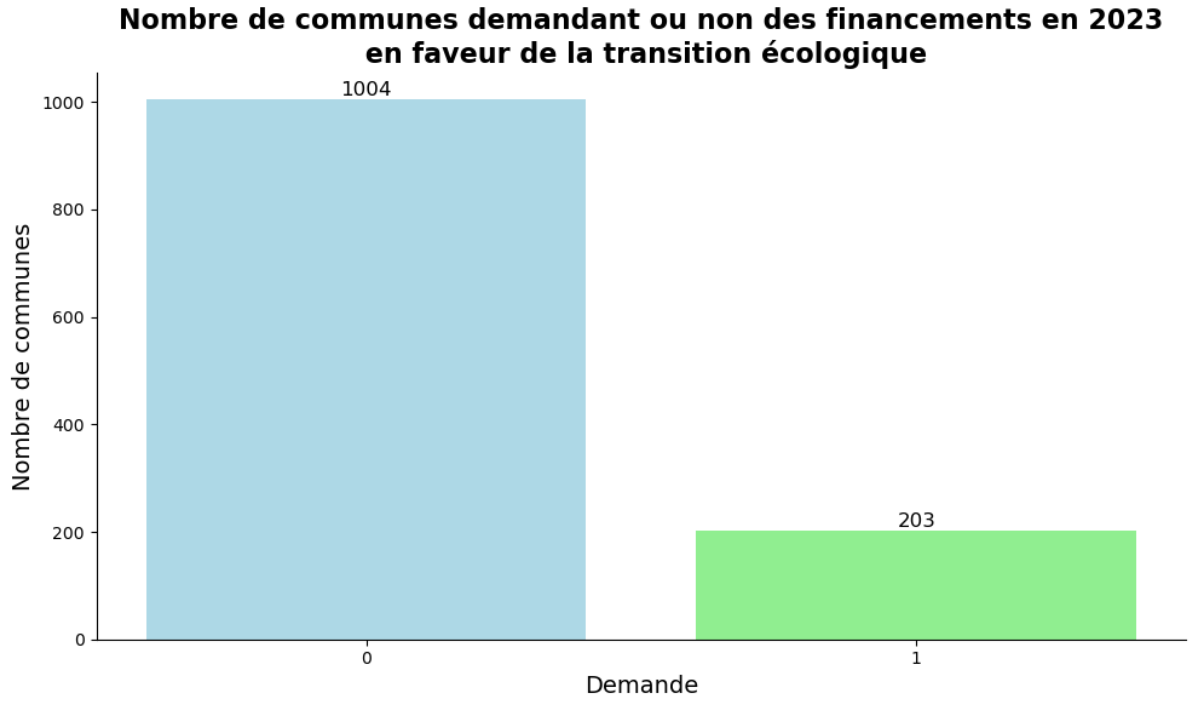


FIGURE 2 – Répartition des commmunes ayant demandé des financments ou non en faveur de la transition écologique en 2023

2.1.4 Analyse statistique

Méthodes utilisées
Nous allons décrire et expliquer la mise en place et la construction des algorithmes employés lors de cette étude :

- La régression logistique
- La régression ridge
- Les arbres de décision
- Les forêts aléatoires
- Les méthodes de boosting : gradient boosting et adaboost

Régression logistique
La régression logistique est une technique statistique qui a pour objectif de produire un modèle permettant de prédire les valeurs prises par une variable à expliquer Y à partir de p va riables explicatives. Elle s'applique lorsque la variable à expliquer Y est binaire et les variables explicatives sont qualitatives ou quantitatives. Elle modélise directement la probabilité qu'une observation appartienne à une classe en fonction des variables explicatives.

ALGORITHME DE RÉGRESSION LOGISTIQUE
Soit Y une variable à valeurs dans $\{0, 1\}$ à expliquer par p variables explicatives $X = (1, X_1, \ldots, X_p)'$. Le modèle logistique propose une modélisation de la loi de $Y|X = x$ par une loi de Bernoulli de paramètre $p_\beta(x) = \mathbb{P}(Y = 1|X = x)$ telle que :

$$\log \frac{p_\beta(x)}{1 - p_\beta(x)} = \beta_0 + \beta_1 x_1 + \ldots + \beta_p x_p = x'\beta \tag{1.1}$$

ou encore

$$\text{logit } p_\beta(x) = x'\beta,$$

logit désignant la fonction bijective et dérivable de $]0,1[$ dans $\mathbb{R}$: $p \mapsto \log\left(\frac{p}{1-p}\right)$.

L'égalité (1.1) peut également s'écrire :

$$p_\beta(x) = \mathbb{P}(Y = 1|X = x) = \frac{\exp(x'\beta)}{1 + \exp(x'\beta)}.$$

La valeur que prend $Y(\{0,1\})$ représente le fait qu'une commune demande, ou non, des financements en faveur de la transition écologique.

L'objectif est de trouver une combinaison linéaire des X qui expliquent bien Y en **estimant les coefficients** $(\beta_1, ..., \beta_p)$ de la fonction logit, qui sont inconnus. La méthode utilisée est celle du maximum de vraisemblance. On cherche à maximiser :

$$\mathcal{L}(y_1, ...y_n, \beta) = \sum_{i=1}^{n} [y_i x_i^t \beta - \log(1 + \exp(x_i^t \beta))]$$

Pour n assez grand, la loi des estimateurs peut être approchée par une loi gaussienne :

$$\mathcal{L}(\hat{\beta}) = \mathcal{N}(\beta, \Sigma_{\hat{\beta}})$$

Choix des paramètres du modèle

Pour paramétrer notre modèle, nous devons effectuer deux choix :

1. une loi pour $Y|X = x$, ici la loi de Bernouilli ;
2. une fonction de lien inversible g : modélisation de $\mathbb{P}(Y = 1|X = x)$ par

$$\text{logit } \mathbb{P}_\beta(Y = 1|X = x) = x'\beta.$$

La fonction **logit** est bijective et dérivable. Elle est appelée fonction de lien. Remarquons également que :

$$\begin{cases} \mathbb{E}_\beta[Y|X = x] = \mathbb{P}_\beta(Y = 1|X = x) \\ \mathbb{V}_\beta(Y|X = x) = \mathbb{P}_\beta(Y = 1|X = x)\,(1 - \mathbb{P}_\beta(Y = 1|X = x)) \end{cases}$$

ce qui implique que la variance n'est pas constante et varie selon x.

Optimisation du modèle

Afin d'optimiser le modèle de régression logistique, d'autres **fonctions de lien** peuvent être utilisées comme les transformations **probit** ou **log-log**.

— **probit**, qui n'est autre que l'inverse de la fonction de répartition de la loi normale centrée réduite :

$$\forall p \in [0,1], \quad \text{probit}(p) = \epsilon \quad \text{avec} \quad \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{\epsilon} \exp\left(-\frac{1}{2}u^2\right) du = p.$$

— **log-log** définie par :

$$\forall p \in [0,1], \quad \text{log-log}(p) = \log\left(-\log(1-p)\right).$$

Ensuite, pour optimiser la performance des estimateurs, le choix de la **pénalité** va jouer un rôle crucial dans les résultats de la méthode.
Si les variables explicatives présentent des colinéarités entre elles, leur variance va augmenter et pourrait nous conduire à du sur-apprentissage. De plus, lorsque le nombre de variables "inutiles" augmente, la précision diminue. L'idée est donc de contraindre la valeur des estimateurs des moindres carrés pour réduire la variance tout en acceptant un biais légèrement supérieur. Il existe des stratégies de régularisation qui mesure la complexité des algortihmes linéaires :

- **Régularisation L1** : qui correspond à la régression lasso qui permet de réduire les valeurs des coefficients à zéro. Cette pénalité peut être très utile pour la sélection de variables. Cependant, dans notre cas, nous avons choisi de ne pas la mettre en place car nous voulions tester toutes les variables puisqu'elles ont été choisi avec des agents métiers.
- **Régularisation L2** : correspondant à la régression ridge, qui sera expliquée dans la section ci-dessous. Cette technique permet aussi de réduire les valeurs des coefficients.
- **Régularisation Elastic Net** : qui combine les deux techniques en ajoutant les termes de pénalité L1 et L2 à la fonction de perte durant l'apprentissage. Cette régularisation n'a pas été testée non plus puisque nous voulions conserver toutes les variables.

Importance des variables

L'importance des variables peut être évaluée à travers les coefficients estimés β dans le modèle. En général, les variables ayant des coefficients absolus plus élevés ont un impact plus important sur les prédiction de Y.
Dans notre situation de données déséquilibrées, les coefficients β mesurent l'effet des variables explicatives sur la probabilité qu'une commune demande des finacements (classe minoritaire).
Cependant, ces résultats sont à analyser avec précaution car l'estimation peut être biaisée. En effet, la régression logistique aura tendance à surestimer l'effet des variables associées à la classe majoritaire (les communes ne demandant pas de financement).

Régression ridge

En appliquant la **régularisation L2**, on se ramène à l'utilisation de la régression ridge qui permet de contrôler la complexité du modèle.
Les estimateurs ridge $\hat{\beta}^R$ s'obtiennent en minimisant la somme des carrés des résidus sous la contrainte que la somme des carrés des coefficients β_j est inférieure à un paramètre λ.

$$\hat{\beta}^R = \arg\min_{\beta} \left\{ \sum_{i=1}^{n} \left(y_i - \beta_0 - \sum_{j=1}^{d} x_{ij}\beta_j \right)^2 + \lambda \sum_{j=1}^{d} \beta_j^2 \right\}.$$

L'estimateur dépend du paramètre lambda : $\hat{\beta}^R = \hat{\beta}^R(\lambda)$

Pour mettre en place la régression ridge, il faut **réduire** les variables explicatives pour éviter les problèmes d'échelle dans la pénalité.
En discrimination binaire, l'estimateur ridge correspond à :

$$\hat{\beta}^R = \arg\min_{\beta} \left\{ -\sum_{i=1}^{n} \left(\tilde{y}_i x_i^t \beta - \log(1 + \exp(x_i^t \beta)) \right) + \lambda \sum_{j=1}^{d} \beta_j^2 \right\}.$$

avec $\tilde{y}_i = (y_i + 1)/2$

Choix des paramètres et optimisation du modèle

Tout d'abord, nous devons choisir le paramètres λ. Pour cela, on réalise :

1. Une estimation d'un critère de choix de modèle pour toutes les valeurs de λ ;
2. Le choix du λ qui minimise le critère estimé.

Lors de l'optimisation du modèle, nous devons faire varier la valeur du **paramètre** λ. Le choix de ce paramètre est crucial car lorsque sa valeur augmente, le biais augmente et la variance diminue (et réciproquement lorsque λ diminue). Il influence donc directement le compromis biais-variance.

Importance des variables

L'analyse de l'importance des variables dans la régression ridge est similaire à celle de la régression logistique car il suffit d'observer les coefficients régularisés $\hat{\beta}^R$. Les variables avec des coefficients non nuls après régularisation contribuent de manière significative aux prédictions
La régression ridge a tendance à pénaliser les coefficients trop grands, ce qui est pratique pour gérer les problèmes de multicollinérarité et de données déséquilibrées. La variance des estimations est réduite grâce à la régularisation et nous aide donc à stabiliser les coefficients. L'analyse de l'impact des variables est donc plus pertinent dans notre situation.

Arbres de décision

Les méthodes par arbres de décision sont des algorithmes où la prévision s'effectue à partir de **moyennes locales**.
Plus précisement, étant donné un échantillon $(x_1, y_1), ..., (x_n, y_n)$, en classification, la méthode consiste à :

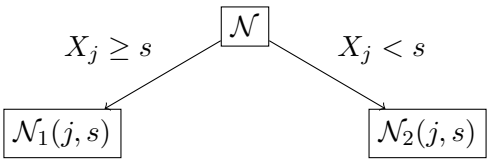
— construire une partition de l'espaces de variables explicatives ($\mathbb{R}^p$)
— prédire la sortie d'une nouvelle observation x en faisant un vote à la majorité parmi les y_i tels que les x_i qui sont dans la même classe que x

L'objectif des arbres de décision est de trouver une **partition** qui sépare au mieux les communes qui demandent et ne demandent pas de financements en faveur de la transition écologique. Pour trouver celle-ci, nous avons utilisé l'**algorithme CART** où la partition est construite par divisions successives parallèles aux axes de $\mathbb{R}^p$.

Algorithme cart et élagage

L'algorithme est récursif : il va à chaque étape séparer un groupe d'observations (noeuds) en deux groupes (noeuds fils) en cherchant la meilleure variable et le meilleur seuil de coupure. Ce choix s'effectue à partir d'un critère d'impureté : la meilleure coupure est celle pour laquelle l'impureté des 2 nœuds fils sera minimale.

Le premier noeud s'apelle le noeud racine et contient toutes les observations.
Une coupure divise un noeud en deux noeuds fils. Soit une coupure représentée par un couple, $(j, s) \in \{1, ..., d\} \times \mathbb{R}$:



Les noeuds qui ne sont pas découpés sont les noeuds terminaux ou les feuilles.

Notre objectif est de prédire la probabilité d'une commune à demander des financements. Nous allons donc utiliser la méthode CART pour prédire celle-ci.
On note $\mathcal{N}(x)$ la feuille de l'arbre qui contient $x \in \mathbb{R}^n$, les prévisions s'obtiennent selon la

proportion d'observations du groupe k :

$$S_{k,n}(x) = \frac{1}{|\mathcal{N}(x)|} \sum_{i:x_i \in \mathcal{N}(x)} \mathbf{1}_{y_i=k} \tag{1}$$

Choix et optimisation des hyperparamètres

Lors de l'implémentation d'un modèle d'arbres de décision, il y a plusieurs paramètres à choisir :

— Le critère de partitionnement : la fonction d'impureté

— Le critère d'arrêt : la profondeur maximale de l'arbre, le nombre minimum d'observations pour continuer le partitionnement.

Premièrement, le critère de partitionnement nous permet de choisir quelle variable sera utilisée pour réaliser la partition à chaque coupure. Le choix des coupures se fait grâce à la définition d'un **critère** mesurant l'homogéinéité des noeuds et en choisissant celle qui optimise celui-ci.
La fonction d'impureté permet de mesurer l'homogénéité d'un noeud. Il faut donc choisir la coupure qui maximise la **pureté** des noeuds fils.

L'impureté $\mathcal{I}$ d'un nœud doit être :

1. faible lorsque un nœud est homogène : les valeurs de Y dans le nœud sont proches.
2. élevée lorsque un nœud est hétérogène : les valeurs de Y dans le nœud sont dispersées

Dans notre situation de classification, les $Y_i, i = 1, ..., n$ sont à valeurs dans $\{1, ..., K\}$. On cherche une fonction $\mathcal{I}$ telle que $\mathcal{I}(\mathcal{N})$ soit :

— petite si un label majoritaire se distingue clairement dans $\mathcal{N}$;

— grande sinon.

En classification, l'**impureté d'un noeud** se mesure selon :

$$I(\mathcal{N}) = \sum_{j=1}^{K} f(p_j(\mathcal{N}))$$

où

— $p_j(\mathcal{N})$ représente la proportion d'observations de la classe j dans le nœud $\mathcal{N}$.

— f est une fonction (concave) $[0, 1] \to \mathbb{R}^+$ telle que $f(0) = f(1) = 0$.

Les 3 mesures d'impureté qui seront utilisées lors de cette étude sont :

1. L'indice de Gini : $f(p) = p(1-p)$;
2. L'entropie : $f(p) = -p\log(p) - (1-p)log(1-p)$.
3. La perte logarithmique : $f(p) = -[y \times log(p) + (1-y) \times log(1-p)]$

Deuxièmement, la **complexité de l'arbre** est représentée par le nombre de coupures, ou la profondeur de l'arbre. Si la profondeur de l'arbre est trop importante, il existe un risque de sur-apprentissage, c'est pourquoi il est nécessaire d'optimiser cet hyperparamètre. Cette étape est aussi appelé élagage CART et consiste à se ramener à une sous-suite d'arbres emboîtés et ensuite à choisir un arbre qui optimise le risque.

Importance des variables

La mesure d'importance des variables dans un arbre de décision repose sur le **gain d'impureté** aux noeuds internes. Lors de chaque division d'un noeud, l'impureté, mesurée par une fonction telle que l'entropie ou l'indice de Gini, est réduite. Le gain d'impureté correspond à la différence entre l'impureté avant et après la division. Plus ce gain est élevé pour une variable donnée, plus

celle-ci est considérée comme importante pour la construction de l'arbre.
Pour une variable ℓ, l'importance est calculée en additionnant les gains d'impureté aux noeuds où cette variable a été utilisée pour la division. Formellement, l'importance d'une variable ℓ dans un arbre T est donnée par :

$$I_\ell(T) = \sum_{t=1}^{|T|-1} \Delta I_t \cdot 1_{j_t=\ell},$$

où ΔI_t est le gain d'impureté au noeud t et $1_{j_t=\ell}$ est égal à 1 si la variable ℓ a été utilisée pour diviser le noeud t, sinon il vaut 0.

Forêts aléatoires

Une forêt aléatoire est définie à partir d'un ensemble d'arbres de décision.
Soit $T_k(x)$, $k = 1, \ldots, B$ des prédicteurs par arbre ($T_k : \mathbb{R}^d \to [0, 1]$) dans le cadre d'une classification binaire. Le prédicteur des forêts aléatoires est obtenu par agrégation de cette collection d'arbres :

$$f_n(x) = \frac{1}{B} \sum_{k=1}^{B} T_k(x).$$

où $f_n(x)$ représente la probabilité estimée que l'observation x appartienne à la classe 1.

Algorithme forêts aléatoires

L'objectif de l'algorithme des forêts aléatoires est de diminuer la corrélation entre les arbres qu'on agrège. Pour cela, Breiman propose de sélectionner la "meilleure" variable dans un ensemble composé de m variables choisies aléatoirement parmi les d variables initiales.
Dans notre situation, Y est binaire et nous souhaitons modéliser la probabilité qu'une commune demande des financements en faveur de la transition écologique.
On a donc $T_k(x, \theta_b, \mathcal{D}_n) \in [0, 1]$ et

$$S_{n,k}(x) = \frac{1}{B} \sum_{k=1}^{B} T_k(x, \theta_b, \mathcal{D}_n) \quad k = 1, ..., K$$

où $S_{n,k}(x)$ est la probabilité agregrée que l'observation x appartienne à la classe positive, calculée à partir des B arbres de la forêt.

Pour réduire la variance des prédictions et améliorer la robustesse du modèle, les forêts aléatoires utilisent une technique appelée bootstrap lors de la construction des arbres.
Le bootstrap est une méthode de rééchantillonnage qui consiste à tirer de manière répétée des échantillons avec remise à partir des données initiales, afin de créer plusieurs échantillons de la même taille. Cette méthode permet de produire des ensembles de données variés pour entraîner les arbres de la forêt aléatoire, renforçant ainsi la robustesse du modèle tout en réduisant le risque de surapprentissage.

Choix et optimisation des hyperparamètres

La calibration des paramètres de l'algorithme des forêts aléatoires comprend plusieurs choix.
Premièrement, le choix du **nombre d'arbres** qu'on agrège est très important. Ce nombre doit être le plus grand possible pour ajuster au mieux les données.

Deuxièmement, le paramètre qui détermine le **nombre de variables à chaque coupure**, soit les variables de coupure. Celui-ci influence directement le compromis biais/variance de la forêt. Lorsque la valeur du paramètre est petit alors la variance de la forêt augmente mais le biais des arbres diminue. Si on prend une valeur trop importante, cela présente un risque de

sur-apprentissage. Pour cela, il faut donc comparer les performances de la forêt avec différentes valeurs.
La valeur par défaut de ce paramètre est : $\sqrt{d}$ avec d variables explicatives.

Aussi, il faut calibrer le paramètre qui détermine le **nombre minimum d'observations** pour qu'un noeud puisse être divisé.

Enfin, il faut déterminer un **seuil de coupure** $s < n$, avec n le nombre d'observations/communes.

Importance des variables

L'importance des variables est mesurée avec le score d'impureté dans notre étude, comme pour les abres de décision. Pour le **score d'impureté**, on peut se référer à la partie sur les arbres de décision. Il suffit donc de faire la moyenne des importances de X_j dans chaque forêt :

$$I_j^{imp} = \frac{1}{B} \sum_{b=1}^{B} \mathcal{I}_j(T_b).$$

Boosting

Le terme de « boosting » vient de l'idée de « booster » un algorithme peu performant, ou faible, pour en faire un algorithme fort. En effet, un algorithme faible a des performances limitées et ne peut pas bien généraliser ou prédire correctement les données.
Le boosting est une méthode d'agrégation, tout comme les forêts aléatoires. Le principe général consiste à construire une famille d'estimateurs qui sont ensuite agrégés par un vote à la majorité pour réaliser une prévision. Les estimateurs, ou arbres, sont construits de manière récursive : chaque estimateur est une version adaptative du précédent en donnant plus de poids aux observations mal ajustées ou mal prédites. Pour pouvoir procéder à la méthode de boosting, nous devons disposer d'un algorithme faible, avec des mauvaises capacités de prédiction. Dans notre situation, ce type d'algorithme est caractérisé par des arbres peu profonds, avec un biais fort et une faible variance. Pour obtenir une seule prévision, on construit puis agrège un grand nombre d'algorithmes "simples".

La méthode procède en plusieurs étapes : On applique l'algorithme faible sur notre échantillon initial, nous donnant un arbre de décision avec des poids pour chaque observation. Puis on procède de manière itérative :

1. Si l'observation est mal classée alors le poids de celle-ci est augmenté. Cela permet à l'algorithme faible de se concentrer sur les observations difficiles à classer. Autrement dit, on obtient des données pondérées par leur poids.
2. On réapplique l'algorithme sur l'échantillon avec des poids pondérés.

A chaque étape, on combine un nouvel arbre pour réaliser les prédictions. Et après plusieurs répétitions de ces deux étapes, les règles faibles sont combinées en une règle unique de prédiction forte. On obtient donc un algorithme « boosté » et avec un biais plus faible.

Lors de cette étude, nous allons utiliser deux algorithmes de boosting : le **Gradient Boosting** et **Adaboost** dans l'objectif de minimiser le risque de la fonction de prévision.

Algorithme de Gradient Boosting

Les algorithmes de Gradient Boosting permettent de minimiser des pertes empiriques de la forme $\frac{1}{n}\sum_{i=1}^{n} \ell(y_i, f(x_i))$ où $\ell : \mathbb{R}^n \to \mathbb{R}$ est une fonction de coût convexe. Notre objectif est de **minimiser le risque** $R_n(g) = \frac{1}{n}\sum_{i=1}^{n} \ell(y_i, g(x_i))$ en trouvant la meilleure combinaison linéaire

d'arbres binaires possible. L'optimisation sur toutes les combinaisons linéaires est complexe c'est pourquoi on utilise la descente de gradient pour construire la combinaison d'arbres de manière récursive.

Etapes de l'algorithme :

1. **Initialisation** : $f_0(\cdot) = \arg\min_c \frac{1}{n}\sum_{i=1}^n \ell(y_i, c)$
2. **Pour** $b = 1$ à B :
   (a) Calculer l'opposé du gradient $-\frac{\partial \ell(y_i, f(x_i))}{\partial f(x_i)}$ et l'évaluer aux points $f_{b-1}(x_i)$:

$$u_i = -\left.\frac{\partial}{\partial f(x_i)}\ell(y_i, f(x_i))\right|_{f(x_i)=f_{b-1}(x_i)}, \quad i = 1, \ldots, n.$$

   (b) Ajuster un arbre de régression à J feuilles sur $(x_i, u_i), \ldots, (x_n, u_n)$.
   (c) Calculer les valeurs prédites dans chaque feuille

$$\gamma_{jb} = \arg\min_{\gamma} \sum_{i: x_i \in \mathcal{N}_{jb}} \ell(y_i, f_{b-1}(x_i) + \gamma).$$

   (d) Mise à jour : $f_b(x) = f_{b-1}(x) + \sum_{j=1}^J \gamma_{jb} 1_{x \in \mathcal{N}_{jb}}$.

Retourner : l'algorithme $f_n(x) = f_B(x)$.

Algorithme Adaboost

L'initialisation du modèle Adaboost consiste à mettre en place un arbre de décision en attribuant des poids égaux pour $(\frac{1}{n})$ chaque observation pour l'estimation du premier modèle. Ils sont ensuite mis à jour pour chaque itération.
L'algorithme ajuste les poids à chaque étape mais d'une manière différente :

1. Si l'observation est mal classée alors le poids de celle-ci est augmenté. Cela permet à l'algorithme faible de se concentrer sur les observations difficiles à classer. Autrement dit, on obtient des données pondérées par leur poids.
2. Si l'observation est mal classée alors le poids de celle-ci est augmenté tandis que les observations bien classées voient leur poids inchangé.

Cette étape a pour but de construire un deuxième arbre et que ses prédictions soient plus précises que celles du premier. Le reste de la méthode fonctionne de la même manière en agrégeant différents arbres avec des poids ajustés.
L'algorithme Adaboost se construit de la manière décrite ci-dessous.

Entrées :

- x l'observation à prévoir
- $d_n = (x_1, y_1), \ldots, (x_n, y_n)$ l'échantillon
- Une règle faible
- M le nombre d'itérations.

Etapes :

1. **Initialiser les poids** $w_i = 1/n$, $i = 1, \ldots, n$
2. **Pour** $m = 1$ à M :
   (a) Ajuster la règle faible sur l'échantillon d_n pondéré par les poids $w_1, \ldots, w_n$, on note $g_m(x)$ l'estimateur issu de cet ajustement

(b) Calculer le taux d'erreur :

$$e_m = \frac{\sum_{i=1}^{n} w_i 1_{y_i \neq g_m(x_i)}}{\sum_{i=1}^{n} w_i}$$

(c) Calculer : $\alpha_m = \log((1 - e_m)/e_m)$

(d) Réajuster les poids :

$$w_i = w_i \exp(\alpha_m 1_{y_i \neq g_m(x_i)}), \quad i = 1, \ldots, n$$

Sortie : $\hat{G}_M(x) = \sum_{m=1}^{M} \alpha_m g_m(x)$

Optimisation des modèles

Afin d'améliorer les performances et la précision des modèles de Gradient Boosting et d'Adaboost, on **ajuste les hyperparamètres**. Le but est de trouver les meilleures valeurs possibles pour ces paramètres afin d'obtenir les meilleurs résultats possibles sans avoir de sur-ajustement.

Lors de l'initialisation des algorithmes, il est nécessaire de calibrer, puis d'ajuster plusieurs paramètres :

— ℓ la fonction de perte
— B nombre d'itérations
— J le nombre de feuilles des arbres
— λ le paramètre de rétrécissement

Tout d'abord, la **fonction de perte** doit :

1. mesurer un coût et caractérise la fonction de prévision à estimer $\Rightarrow \mathrm{f}_n$ est un estimateur de $f^* \in \arg\min_{f:\mathbb{R}^d \to \mathbb{R}} \mathbf{E}[\ell(Y, f(X)]$
2. être convexe et dérivable par rapport à son second argument.

Ensuite, le **nombre de coupures** est à paramétrer lors de la validation croisée et doit être assez faible pour avoir des arbres biaisés (avec peu de coupures). En effet, le boosting réduit le biais et l'objectif est de combiner des algorithmes faibles pour en former un fort.

Ensuite, le **nombre d'itérations**, ou le nombre d'arbres mesure la complexité de notre algorithme. Il faut calibrer correctement ce paramètre pour éviter le sur-ajustement. Plus on itère, mieux on prédit mais on se rapproche du sur-apprentissage si le nombre d'itérations est trop important.
Pour calibrer ce paramètre, on utilise les méthodes d'estimation du risque, ou de ré-échantillonnage comme la validation croisée par exemple.

Enfin, le paramètre de **rétrecissement** correspond au pas de la descente de gradient. Celui-ci est directement lié au nombre d'arbres car lorsque le paramètre de rétrecissement augmente, le nombre d'arbres diminue, et inversement. L'optimisation de cet hyper-paramètre doit être réalisée en regardant les courbes de risque et nous devons nous assurer que le nombre d'itérations optimal se trouve sur un "plateau" pour des raisons de stabilité.

Importance des variables

Le principe du boosting est de construire un modèle de manière itérative en ajoutant de nouveaux modèles qui corrigent les erreurs des modèles précédents. Chaque modèle dans cette séquence est souvent un arbre de décision. Le score d'impureté (I_j^{imp}) mesure l'**importance** d'une variable j

sur l'ensemble de ces arbres.
Le score d'impureté est donné par :

$$I_j^{\text{imp}} = \frac{1}{B} \sum_{b=1}^{B} I_j(T_b)$$

où :

- B est le nombre total d'arbres dans le modèle de gradient boosting.
- $I_j(T_b)$ est l'impureté attribuée à la variable j dans l'arbre b. La mesure de la pureté peut être mesurée par des métriques comme l'indice de Gini ou l'entropie par exemple.

L'importance d'une variable j (I_j^{imp}) est calculée comme la moyenne de son importance individuelle dans chaque arbres b. Cette moyenne donne une indication globale de la contribution de la variable j au modèle de boosting.

Le **score d'impureté** nous permet d'identifier quelles variables sont les plus importantes pour le modèle en fonction de leur capacité à réduire l'erreur de prédiction à chaque étape.

Gestion des problèmes éventuels
Lors de ce projet, nous faisons face à un problème de **données déséquilibrées**. En effet, le nombre de communes demandant des financements pour la transition écologique est nettement inférieur à celui des communes ne faisant pas les démarches. Nous avons donc une majorité d'observations présentant la valeur 0 et peu de communes avec la valeur 1 (voir graphique 2). Si nous utilisons une méthode classique, le risque est que l'algorithme prédise très bien la classe majoritaire, mais très mal la classe minoritaire.

Les données déséquilibrées peuvent mettre en défaut les performances de nos modèles, c'est pourquoi nous avons mis en place des stratégies pour pallier à cette difficulté. La stratégie classique est de prendre en compte le déséquilibre dans la mesure de performance des algorithmes et de mettre en place du ré-échantillonnage (voir ci-après).

Pour évaluer la performance de notre algorithme, nous devons évaluer les critères de performance sur des données non utilisées pour construire l'algorithme (échantillon d'apprentissage). L'objectif est de minimiser le risque d'erreur de prévisions. Pour cela, le critère de prévision, ou la fonction de perte, mesure une perte entre n prévisions $\hat{y}_i, i = 1, ..., n$ et n observations $y_i, i = 1, ..., n$. Elle est définit comme une fonction :

$$\begin{aligned} \mathcal{R} : \mathbb{R}^n \times \mathbb{R}^n &\to \mathbb{R} \\ (y_1^n, \hat{y}_1^n) &\mapsto \mathcal{R}(y_1^n, \hat{y}_1^n). \end{aligned}$$

Des critères comme l'AUC, le balanced accuracy et le F_1-score sont des critères adaptés au cas des données déséquilibrées, c'est pourquoi nous les analyserons par la suite.

Pour choisir le meilleur algorithme, nous comparerons les modèles rééquilibrés et non rééquilibrés dans l'objectif de minimiser la fonction de perte.

Ré-échantillonnage
Pour répondre au problème de données déséquilibrées, nous avons mis en place quelques approches pour tenter de combler le déséquilibre. Nous avons choisi d'utiliser des méthodes qui sur-échantillonne la classe minoritaire pour ne pas supprimer des individus car nous ne disposions que de 1207 communes/observations.

Random Oversampling
Cette méthode consiste à dupliquer aléatoirement des observations de la classe minoritaire. Grâce à ce processus, nous diminuons de façon artificielle la variabilité des données.

SMOTE
L'algorithme Synthetic Minority Over-samplig TEchnique (SMOTE) nous permet de générer de nouvelles observations de la classe minoritaire.

L'algorithme **SMOTE** est construit comme exliqué ci-dessous.
Pour une observation x_m de la plus petite classe, on génèrera une nouvelle observations selon l'algorithme :

1. Calculer les k plus proches voisins de x_m parmi les $x_i = 1, ..., n$ privés de x_m qui sont dans la même classe que x_m
2. Choisir au hasard un des plus proches voisins calculés précédemment, on le note $x_{(m)}$
3. Générer aléatoirement une nouvelle observation x sur le segment reliant x_m à $x_{(m)}$

Les **paramètres** de SMOTE sont à calibrer :

— On doit contrôler le paramètre de rééquilibrage (par défaut il est parfait).
— Le nombre de voisins k à considérer pour générer les nouveaux individus (plus petit que le nombre d'observations de la classe minoritaire)

Borderline SMOTE
La variante Borderline-SMOTE est une version améliorée de la méthode de rééquilibrage de données SMOTE. En effet, la génération de nouvelles observations va un peu plus loin car on se focalise sur les observations de la classe minoritaire qui se trouvent sur les frontières de décision (les cas "borderline"). Ces observations sont plus susceptibles d'être mal classées par nos modèles car ils sont proches de la classe majoritaire.

L'algorithme Borderline-SMOTE est construit en deux étapes :

1. **Identification des observations borderline**
   Pour identifier les observations borderline, on utilise un algorithme des k plus proches voisins.
   Soit S_{min} l'ensemble des observations de la classe minoritaire et S_{maj} l'ensemble des observations de la classe majoritaire.
   Pour chaque observation x_i de la classe minoritaire S_{min} :
   (a) On calcule les k plus proches voisins, notés $N_k(x_i)$.
   (b) On compte le nombre de voisins appartenant à la classe majoritaire $|N_{\text{maj}}(x_i)|$, où $N_{\text{maj}}(x_i) = \{x \in N_k(x_i) \mid x \in S_{\text{maj}}\}$.
2. **Génération d'individus synthétiques**
   Pour chaque observation x_i identifiée comme borderline, Borderline-SMOTE génère des observation synthétiques en interpolant entre x_i et un de ses voisins de la classe minoritaire.
   La formule pour générer une nouvelle observation synthétique x_{new} est :

$$x_{\text{new}} = x_i + \lambda \times (x_j - x_i)$$

   où :
   — x_i est une observation borderline de la classe minoritaire.
   — x_j est un des k plus proches voisins minoritaires de x_i, $x_j \in N_k(x_i)$ et $x_j \in S_{\text{min}}$.

— λ est un facteur aléatoire dans l'intervalle $[0, 1]$.

Cela signifie que x_{new} est un point sur la ligne reliant x_i et x_j, et que la position exacte de x_{new} dépend de la valeur de λ.

Les **paramètres** à calibrer sont :

— k : le nombre de voisins utilisés pour identifier les observations borderline ainsi que pour générer les observations synthétiques

— m : le nombre de voisins majoritaires considérés pour identifier les observations/communes à la frontières

— *kind* : le type de borderline smote à choisir

1. $borderline - 1$: génère des observations uniquement pour les observations de la classe minoritaire qui sont entourés principalement de voisins majoritaires
2. $borderline - 2$: génère des observations synthétiques pour tous les observations de la classe minoritaire à la frontière, même ceux qui sont entourés de nombreux voisins minoritaires.

— le ratio pour déterminer le nombre d'observations synthétiques à générer

Adasyn

L'algorithme ADAptive SYNthetic sampling est assez proche de SMOTE mais le nombre d'observations générés pour un x_i est proportionnel à la densité des observations du groupe majoritaire au voisinage de x_i.
On note qu'il y a plus d'observations qui sont générées aux voisinages des cas isolés.

L'algorithme ADASYN est construit comme ceci :
Entrées : $dth \leq 1, k$ plus petit que $n_1, \beta \in \mathbb{R}^+$.

1. Calculer $d = n_0/n_1$. Si $d \leq dth$ stop.
2. Calculer G le nombre d'observations à générer :
$$G = (n_0 - n_1)\beta.$$
3. Calculer les k-ppv de chaque individu $x_i, i \in \mathcal{X}_1$ de la classe minoritaire et en déduire pour chacun
$$r_i = \frac{\text{card}\{j : y_j = 0 \text{ et } x_j \in k\text{ppv}(x_i)\}}{k}.$$
4. Normalisation :
$$\tilde{r_i} = \frac{r_i}{\sum_{i \in \mathcal{X}_1} r_i}.$$
5. Calculer le nombre d'individus à générer pour chaque x_i de $\mathcal{X}_1$:
$$G_i = G\tilde{r_i}.$$
6. Pour chaque x_i de $\mathcal{X}_1$ *répéter* G_i *fois* :
   (a) Choisir au hasard un de ces $k$ppv du groupe 1 $\implies x_i^{(1)}$
   (b) Générer un point au hasard entre x_i et $x_i^{(1)}$
$$x_{i,j} = x_i + \lambda(x_i^{(1)} - x_i)$$
où λ est générée selon une loi uniforme sur $[0, 1]$.

Sorties : les nouvelles données $x_{i,j}, i \in \mathcal{X}_1, j \in \{1, \ldots, G_i\}$.

Il faut aussi choisir les **paramètres** comme :

— *beta* pour contrôler le niveau de ré-équilibrage

— k pour la taille des voisinages

— la distance pour calculer les k-ppv

Pour pouvoir évaluer si ces méthodes de ré-échantillonage étaient pertinentes, nous avons utilisé des critères de performance pour les comparer à la méthode réalisée mais sans rééquilibrage.

Critères de performance

Tout d'abord, les **critères de performance** utilisés prennent en compte des "critères conditionnels" pour ne pas donner un poids trop important à la classe majoritaire.
Les critères que nous allons analyser seront donc :

— Courbe ROC (Receiver Operating Characteristic)/ AUC (Area Under the ROC curve)

— Balanced accuracy

— F_1-score

Courbe ROC / AUC

La courbe ROC est un critère pertinent pour notre problématique car il est basé sur la manière dont le score ordonne les individus (communes). Aussi, il ne dépend pas des proportions de classes, ce qui le rend plus robuste face aux données de ce type.

Nous allons chercher une fonction de score $S : \mathcal{X} \to \mathbb{R}$ qui donnera une note à chaque nouvel individu $x \in \mathcal{X}$, au lieu de prédire directement le groupe.
La note $S(x)$ est :

— élevée si il a des "chances" de demander des financements

— faible s'il a peu de "chances" de demander des financements

Étant donné un score, on peut déduire une règle de prévision en fixant un seuil :

$$g_s(x) = \begin{cases} 1 & \text{si } S(x) \geq s \\ -1 & \text{sinon.} \end{cases}$$

Cette règle définit la table de confusion :

	$g_s(X) = -1$	$g_s(X) = 1$
$Y = -1$	OK	E_1
$Y = 1$	E_2	OK

Pour chaque seuil, on distingue deux types d'erreur :

$$\alpha(s) = \mathbb{P}(g_s(X) = 1 \mid Y = -1) = \mathbb{P}(S(X) \geq s \mid Y = -1)$$

et

$$\beta(s) = \mathbb{P}(g_s(X) = -1 \mid Y = 1) = \mathbb{P}(S(X) < s \mid Y = 1).$$

On définit également :

— Spécificité : $sp(s) = \mathbb{P}(S(X) < s \mid Y = -1) = 1 - \alpha(s)$

— Sensibilité : $se(s) = \mathbb{P}(S(X) \geq s \mid Y = 1) = 1 - \beta(s)$

La performance d'un score se mesure généralement en visualisant les erreurs $\alpha(s)$ et $\beta(s)$ et/ou la spécificité et la sensibilité pour tous les seuils $s.beta(s)$ et/ou la spécificité et la sensibilité pour tous les seuils s.
La courbe ROC est un graphe qui trace le taux de vrais positifs (sensibilité) par rapport au taux

de faux positifs (1−spécificité) pour tous les seuils s
C'est une courbe paramétrée par le seuil :

$$\begin{cases} x(s) = \alpha(s) = 1 - sp(s) = \mathbb{P}(S(X) > s \mid Y = -1) \\ y(s) = 1 - \beta(s) = se(s) = \mathbb{P}(S(X) \geq s \mid Y = 1) \end{cases}$$

Pour tout score S, on a :

- $x(-\infty) = y(-\infty) = 1$;
- $x(+\infty) = y(+\infty) = 0$;
- La courbe ROC part de $(1, 1)$ pour finir à $(0, 0)$.

Pour un score aléatoire, la courbe ROC est définie par la première bissectrice. Pour la "pire" courbe ROC, on a donc $\forall s\ x(s) = y(s)$. Nous allons mesurer la performance de notre score en fonction de sa capacité à se rapprocher de la droite d'équation $y = 1$ le plus vite rapidement possible.

Par ailleurs, l'AUC mesure l'aire sous la courbe ROC d'un score S et est notée $AUC(S)$. L'objectif est donc de maximiser l'AUC pour avoir une meilleure performance. Ce critère peut ête interprété comme une fonction de perte pour un score S.
Si notre score est parfait, alors nous aurons $AUC(S) = 1$ mais si celui-ci est aléatoire, la valeur de l'AUC sera de $1/2$.

Balanced accuracy
Ce critère donne le même poids aux vrais positifs et aux vrais négatifs, contrairement à l'accuracy. Le balanced accuracy est la moyenne arithmétique des vrais positifs et négatifs. Il est défini comme ci-dessous :

$$\text{Bal Acc} = \frac{1}{2}\mathbb{P}(g(X) = 1 \mid Y = 1) + \frac{1}{2}\mathbb{P}(g(X) = -1 \mid Y = -1) = \frac{\text{TPR} + \text{TNR}}{2}.$$

F_1-score
Le F_1-score est une moyenne harmonique entre :

1. la précision : $\mathbb{P}(Y = 1 \mid g(X) = 1)$ (capacité à identifier les positifs parmi les prédits positifs)
2. le recall : $\mathbb{P}(g(X) = 1 \mid Y = 1)$ (capacité à bien prédire les positifs)

$$F_1 = 2 \cdot \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

Le Balanced accuracy et le F_1-score doivent être le plus proche de 1 pour que l'algorithme soit performant.

2.1.5 Mise en pratique

Modélisation statistique
L'objectif de la mise en place de cette modélisation statistique est de détecter les communes les moins susceptibles de demander des financements en faveur de la transition écologique.
Pour résoudre cela, nous sommes partis de la base de données citée ci-dessus (voir section 2.1.2).
Le processus a été divisé en plusieurs étapes :

1. Construction et le paramétrage des modèles
2. Evaluation de la performance des modèles initiaux
3. Optimisation des modèles initiaux
4. Evaluation de la performance des modèles optimisés
5. Sélection du modèle final
6. Analyse de l'importance des variables
7. Interprétation des résultats

Tout d'abord, pour construire notre modélisation statistique, nous avons divisé notre jeu de données en deux. En effet, nous avons construit un :

— jeu de données d'apprentissage/entraînement
— jeu de données test

Pour procéder à la modélisation statistique, nous entrainerons nos modèles sur les données d'apprentissage et nous testerons nos modèles ensuite sur les données test. Dans notre étude, l'échantillon d'apprentissage contient 80% des données, soit 965 communes et l'échantillon test contient le reste, c'est-à-dire 242 communes. Au total, nous avons bien les 1207 communes de Bretagne. Pour avoir une modélisation statistique correcte et faciliter la généralisation, nous avons veillé à la bonne répartition des classes dans les deux échantillons (voir tableau 1).

Échantillon	**Proportion de 0**	**Proportion de 1**
Apprentissage	83%	17%
Test	85%	15%

TABLE 1 – Proportions des classes dans les ensembles d'apprentissage et de test

1. Construction et paramétrage des modèles
Pour la construction de tous les modèles, nous avons initialisé les méthodes de rééquilibrage des données de la même manière :

- Pour la méthode de rééquilibrage **RandomOversampler**, nous avons définit la stratégie de rééchantillonnage avec un ratio de 0.4 (`sampling_strategy`). On souhaite donc se ramener à un échantillon avec 40% des communes qui demandent des financements en dupliquant aléatoirement des observations de la classe minoritaire.
- Pour les méthodes **SMOTE** (`k_neighbors`), **ADASYN** (`n_neighbors`) et **BorderlineSMOTE** (`k_neighbors`), nous avons gardé le paramétrage initial à 5 voisins pour générer les nouvelles communes "fictives". De plus, le ratio (`sampling_strategy`) sera initialisé en mode automatique.

Pour structurer et automatiser les différentes étapes du processus de construction des modèles, nous avons utilisé des pipelines (package **sklearn**). Pour chaque implémentation, nous avons donc construit les modèles dans un pipeline incluant une méthode de rééquilibrage des classes, une standardisation des données (`StandardScaler()`) et l'algorithme voulu (avec les paramètres par défaut et en renseignant une graine pour avoir les mêmes résultats à chaque éxécution du code). Les pipelines permettent de garantir que le processus de prétraitement et de modélisation

est systématique et reproductible, ce qui facilitera la prise en main du code pour une autre personne.
Nous avons choisi de réaliser une standardisation des données afin de ne privilégier aucune variable. En effet, cette étape permet d'assurer que toutes les variables contribuent de manière égale au modèle en éliminant les biais potentiels dus à des échelles de mesures différentes. Ceci est très important pour les algorithmes sensibles à l'échelle des variables.

Lors de cette initialisation, nous n'avons pas modifié les valeurs par défaut des algorithmes testés. Pour avoir plus de détails sur la valeur des hyperparamètres des modèles initiaux, on pourra se référer au tableau 2.

Algorithme	**Grille de paramètres initiaux**
Régression logistique	`C` : 1 `penalty` : `l2`

Arbres de décision	`criterion` : `gini`
	`max_depth` : `None`
	`min_samples_split` : 2
	`min_samples_leaf` : 1

Forêts aléatoires	`n_estimators` : 100
	`criterion` : `gini`
	`max_depth` : `None`
	`min_samples_split` : 2
	`min_samples_leaf` : 1
	`bootstrap` : `True`

Gradient boosting	`n_estimators` : 100 `learning_rate` : 0.1 `max_depth` : 3
Adaboost	`n_estimators` : 50 `learning_rate` : 1.0

TABLE 2 – Valeurs initiales des hyperparamètres pour chaque algorithme

2. Evaluation de la performance des modèles initiaux
Après avoir construit nos différents modèles, nous allons évaluer leur performance pour avoir une première idée des résultats et des algorithmes qui seront potentiellement sélectionnés. Pour réaliser cette évaluation, nous utilisons les trois métriques :

- Courbe ROC / AUC
- Balanced accuracy
- F_1-score

Lors de cette première étape sans optimisation, le modèle qui a montré les performances les plus élevées est le modèle **adaboost** avec la méthode de rééquilibrage **RandomOversampler**.
Pour faire ce choix, nous avons d'abord sélectionné la meilleure méthode de ré-échantillonnage pour chaque algorithme. Ensuite, nous avons comparé les algorithmes les plus performants entre eux.

ADABOOST
Dans un premier temps, nous allons expliquer la manière dont nous avons choisi la méthode de rééquilibrage pour adaboost. Pour cela, nous allons analyser le tableau contenant les métriques mesurant la performance de cet algorithme.

Le tableau ci-dessous (voir tableau 3) présente les performances de divers modèles d'Adaboost avec ou sans suréchantillonnage.

	AUC-ROC	F_1-score	Balanced Accuracy
AdaBoost sans sur-échantillonnage	0.628	0.197	0.537
AdaBoost avec SMOTE	0.607	0.274	0.566
AdaBoost avec RandomOverSampler	0.653	0.331	0.614
AdaBoost avec ADASYN	0.628	0.297	0.583
AdaBoost avec BorderlineSMOTE	0.608	0.272	0.562

TABLE 3 – Performances des différents modèles initiaux de AdaBoost (avec valeurs maximales en magenta)

Le modèle utilisant RandomOverSampler affiche les meilleures performances concernant les trois métriques. Ces résultats montrent que le suréchantillonnage avec RandomOverSampler améliore nettement les performances d'Adaboost par rapport aux autres méthodes. Nous pouvons noter que la mise en place de l'algorithme Adaboost sans sur-échantillonnage présente tout de même un AUC à 0.628. Cette valeur nous montre que le modèle est globalement performant, nous pourrons réévaluer cela après optimisation des hyperparamètres. Cependant, nous pouvons noter que la valeur du F_1-score est faible peu importe la méthode de rééquilibrage choisie, signifiant que l'algorithme a des difficultés à prédire correctement qu'une commune demande des financements en faveur de la transition écologique.

Dans un second temps, nous analyserons les courbes ROC pour cet algorithme de manière détaillée pour comprendre l'évaluation des différentes méthodes de rééquilibrage des données.

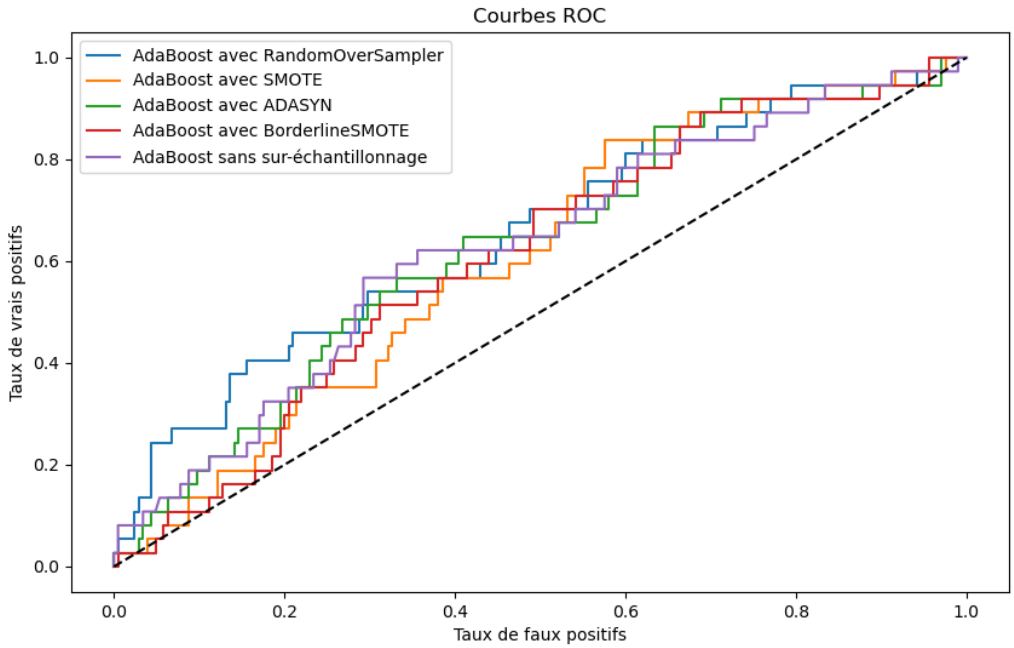


FIGURE 3 – Courbes ROC - Adaboost (modèles initiaux)

Le graphique ci-dessus (figure 3) illustre la performance des modèles en traçant le taux de vrais

positifs (sensibilité) en fonction du taux de faux positifs (1 - spécificité) pour différents seuils de classification.
La courbe la plus proche du coin supérieur gauche du graphique est généralement celle qui présente la meilleure performance et un AUC plus élevé. Dans ce cas, la courbe du modèle avec RandomOversampler semble offrir une meilleure séparation entre les classes comparées aux autres méthodes. Le modèle avec BorderlineSMOTE et SMOTE montrent une performance inférieure, illustrées par des courbes plus proche de la diagonale, indiquant une capacité de discrimination faible. La courbe ROC d'Adaboost avec la méthode ADASYN et sans sur-échantillonnage sont assez proches de celle avec RandomOversampler, montrant des capacités prédictives quasi similaires.
Globalement, ce graphique permet de visualiser l'impact des différentes techniques de rééchantillonnage sur les performances des modèles Adaboost en termes de classification binaire. Ce constat nous permet de comprendre l'utilité du sur-échantillonnage avant même l'optimisation du modèle.

En résumé, l'application de **RandomOverSampler** semble être globalement la méthode la plus efficace pour améliorer les performances d'Adaboost dans notre contexte de données déséquilibrées.

Ces analyses ont été menées pour **chaque algorithme**, afin de choisir celui avec les performances prédictives les plus importantes (tableaux des performances et courbes ROC en annexe D). Après avoir choisi la meilleure méthode pour chaque algorithme, nous avons comparé les performances des algorithmes entre eux (voir le **tableau 4** de comparaison des algorithmes avec les méthodes de rééquilibrage les plus performantes).

Algorithme	**Méthode de rééquilibrage**	**AUC-ROC**	**F_1-score**	**Balanced Accuracy**
Régression logistique	BorderlineSMOTE	0.595	0.283	0.568
Arbres de décision	SMOTE	0.620	0.300	0.586
Forêts aléatoires	SMOTE	0.631	0.219	0.540
Gradient Boosting	ADASYN	0.628	0.293	0.582
AdaBoost	RandomOverSampler	0.653	0.331	0.614

TABLE 4 – Meilleures méthodes de rééquilibrage pour chaque algorithme initial et leurs performances associées

Nous procéderons ensuite à l'optimisation des différents modèles pour vérifier si le modèle **adaboost** utilisant la méthode de rééchantillonnage **RandomOversampler** continue de se démarquer en termes de performances. Lors de cette évaluation des performances des modèles initiaux, nous avons observé des valeurs de F_1-score faibles, ce qui peut nous poser question. Nous analyserons cela au terme de l'optimisation des hyperparamètres si le problème persiste.

3/4. Optimisation des modèles initiaux et évaluation des performances des modèles

PREMIÈRE OPTIMISATION
Pour évaluer la performance de manière efficace entre les modèles, il est nécessaire de procéder à l'optimisation des hyperparamètres de chaque modèle. Pour chaque algorithme, nous avons défini une grille de paramètres qui teste différentes valeurs pour les méthodes de rééquilibrage et pour les paramètres de l'algorithme en question.

D'une part, pour **les méthodes de rééquilibrage**, nous avons testé plusieurs valeurs pour chaque modèle afin d'analyser l'impact de la proportion de la classe minoritaire sur les performances de nos modèles. Pour RandomOversampler, on teste la valeur automatique mais aussi 0.5, 0.75 et 1. Quant aux autres méthodes, on teste avec différents nombres de voisins : 3, 5 et 7. On pourra se référer au tableau 5 ci-dessous pour observer les différentes valeurs testées.

Méthode de rééquilibrage	**Valeurs testées**
Oversampler	`sampling_strategy` : {auto, 0.2, 0.4, 0.5, 0.75, 1.0}
SMOTE	`sampling_strategy` : {auto, 0.2, 0.4, 0.5, 0.75, 1.0} `k_neighbors` : {3, 5, 7}
ADASYN	`sampling_strategy` : {auto, 0.2, 0.4, 0.5, 0.75, 1.0} `n_neighbors` : {3, 5, 7}
Borderline SMOTE	`sampling_strategy` : {auto, 0.2, 0.4, 0.5, 0.75, 1.0} `k_neighbors` : {3, 5, 7}

TABLE 5 – Valeurs testées pour les méthodes de rééquilibrage

D'autre part, pour optimiser les performances de nos modèles, nous avons testé différentes valeurs pour les **paramètres** pour chaque méthode (voir tableau 6). Cette optimisation des hyperparamètres a été réalisée à l'aide de la méthode `RandomizedSearchCV`, associée à une validation croisée à 10-fold. Les paramètres qui sont optimisés pour chaque modèle sont cités et détaillés dans la section 2.1.4.
Nous avons choisi de réaliser une validation croisée **10-fold** et non pas 5-fold comme dans la plupart des cas car après diverses optimisations, les résultats n'étaient pas satisfaisants. En analysant les résultats, nous avions pu observer que nos modèles réalisaient du sur-apprentissage. Le score d'AUC pour l'ensemble d'entrainement était très bon (entre 0.8 et 1) mais présentait des valeurs très faibles pour les données test. Cette observation indiquait une mauvaise capacité de généralisation de notre modèle (voir figure 5).
`RandomizedSearchCV` nous permet de faire une recherche sur la grille des hyperparamètres renseignée. Elle entraine le modèle en sélectionnant aléatoirement un sous-ensemble de combinaisons possibles d'hyperparamètres par rapport à la grille. Pour chaque combinaison sélectionnée, une validation croisée est effectuée pour évaluer la performance du modèle. Au final, elle sélectionne la combinaison de paramètres qui donne la meilleure performance. Nous avons choisi d'utiliser cette méthode plutôt que GridSearchCV par souci de rapidité. Pour avoir plus d'informations sur les paramètres testés pour chaque algorithme, on pourra se référer au tableau ci-dessous.
Nous avons choisi de réaliser une **validation croisée 10-fold** pour avoir une réduction du biais puisque le modèle est testé sur 10 sous-ensembles différents. Le modèle est testé sur plus de sous-ensembles différents, ce qui peut donner une estimation plus stable de la performance moyenne. Nous avons opté pour ce choix puisque nous avons réduit le temps de calcul lors de la recherche des hyperparamètres grâce à `RandomizedSearchCV`. Nous pouvions donc nous permettre de passer plus de temps sur la validation croisée, afin de limiter le sur-apprentissage.

A l'issue de la validation croisée, `RandomizedSearchCV` renvoie le meilleur modèle, celui qui maximise le F_1-score. En effet, nous avons construit un scoring qui est sous forme de dictionnaire où chaque clé est une métrique et la valeur associée est une fonction de score personnalisée avec `make_scorer`. Nous avons utilisé trois métriques : l'AUC/Courbe ROC, le F_1-score et le Balanced accuracy. L'utilisation de plusieurs métriques de scoring dans RandomizedSearchCV permet une évaluation plus complète des performances du modèle. En définissant le scoring de cette manière, on optimise nos modèles en fonction de le F_1-score (qui est ici spécifié comme la métrique principale via `refit='F1'`), mais on peut aussi observer comment nos modèles sur les autres métriques. Ainsi, cela nous donne une vue plus nuancée des performances des modèles et

Algorithme	Grille de paramètres testés
Régression logistique	`C` : [0.01, 0.1, 1, 10, 100] `penalty` : [l2, None]
Arbres de décision	`criterion` : [gini, entropy, log_loss] `max_depth` : [3, 6, 9, 15, 21, 27] `min_samples_split` : [2, 5, 10, 20, 30] `min_samples_leaf` : [1, 2, 4]
Forêts aléatoires	`n_estimators` : [100, 300, 500] `criterion` : [gini, entropy, log_loss] `max_depth` : [3, 15, 21, 27] `min_samples_split` : [2, 7, 15, 27] `min_samples_leaf` : [1, 4] `boostrap` : [True, False]
Gradient boosting	`n_estimators` : [100, 200, 300, 400, 500] `learning_rate` : [0.05, 0.1, 0.5] `max_depth` : [3, 5, 7, 9]
Adaboost	`n_estimators` : [50, 100, 200, 300, 400, 500] `learning_rate` : [0.01, 0.05, 0.1, 0.5]

TABLE 6 – Grille de paramètres testés pour chaque algorithme

nous permet de faire des choix plus logiques pour notre étude. Après l'optimisation des hyperparamètres, nous avons réévalué les performances des modèles afin d'identifier celui présentant les meilleures capacités prédictives. Cette étape permet de vérifier si le modèle ayant montré les meilleures performances initiales conserve son avantage après optimisation. Nous avons aussi comparé les valeurs des hyperparamètres optimisés avec celles par défaut pour observer les changements éventuels.

Dans un premier temps, nous allons analyser les résultats de l'algorithme Adaboost suite à l'optimisation des hyperparamètres. Cette étape va nous permettre de comprendre l'intérêt de l'optimisation des paramètres des modèles.

ADABOOST

	AUC-ROC	F_1-score	Balanced Accuracy
AdaBoost sans sur-échantillonnage	0.617	0.143	0.517
AdaBoost avec SMOTE	0.648	0.280	0.569
AdaBoost avec RandomOverSampler	0.615	0.279	0.563
AdaBoost avec ADASYN	0.645	0.298	0.583
AdaBoost avec BorderlineSMOTE	0.627	0.259	0.550

TABLE 7 – Performances des différents modèles optimisés d'AdaBoost (avec valeurs maximales en magenta)

Auparavant, nous avions observé que la méthode RandomOversampler était la plus performante. Avec l'optimisation des hyperparamètres, la méthode ADASYN se distingue des autres. En effet, les valeurs pour le F_1-score et pour le balanced accuracy sont supérieures aux autres. Par contre, la valeur de l'AUC-ROC est légèrement inférieure à SMOTE (0.645 au lieu de 0.648). Cependant, nous avons des valeurs inférieures lorsque nous optimisons les hyperparamètres du modèle, ce qui peut sembler contre-intuitif. Plusieurs facteurs peuvent expliquer cette situation :

— sur-apprentissage : il est possible que le modèle ait été trop entraîné sur l'ensemble d'entraînement durant l'optimisation, ce qui aurait conduit à du sur-apprentissage.

— rééquilibrage : les méthodes de ré-échantillonnage utilisées génèrent des observations "fictives" ce qui peut dégrader les performances du modèle, surtout s'il sur-ajuste les données.

Le tableau de résultats est cohérent avec le graphique (figure 4) puisque les modèles avec les meilleures performances pour l'AUC-ROC (SMOTE et ADASYN) ont des courbes plus proches du coin supérieur gauche. Les courbes précisent donc que ces modèles ont une meilleure capacité prédictive que les autres modèles. Cependant, ces valeurs restent tout de même assez faibles, ce qui indique que le modèle pourrait encore être amélioré.

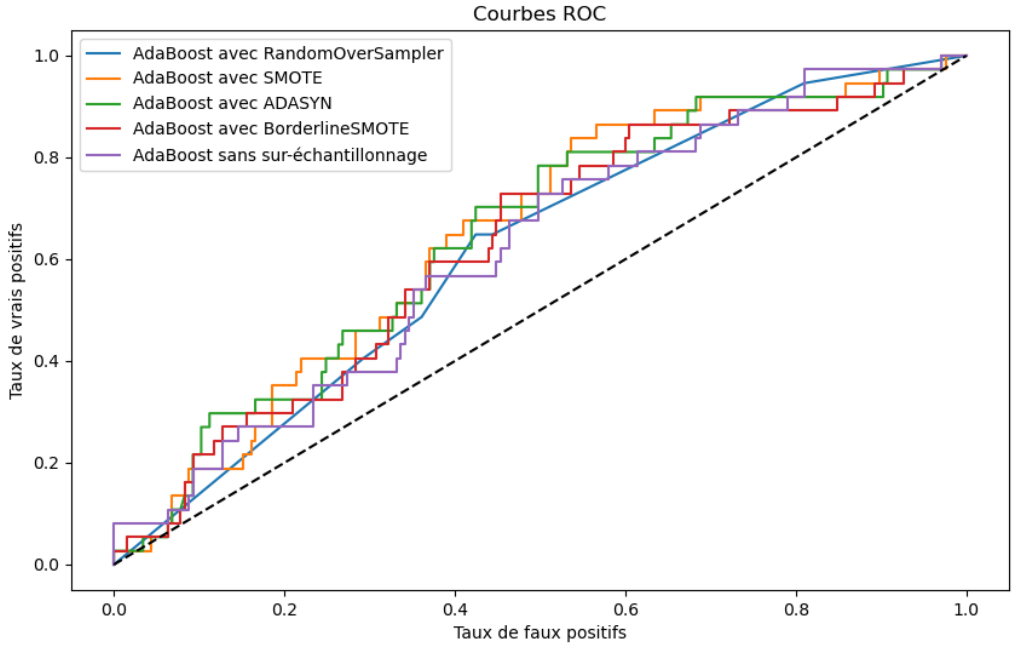


FIGURE 4 – Courbes ROC - Adaboost (modèles optimisés)

Ainsi, nous avons pu déterminer que **ADASYN** est la meilleure méthode de rééquilibrage pour l'algorithme Adaboost.

5. Sélection du modèle final

Puis, nous avons réalisé les mêmes étapes que pour l'algorithme Adaboost pour choisir la meilleure méthode de rééquilibrage pour **chaque algorithme**. En effet, avant et après optimisation, la méthode avec les meilleures performances prédictives peut changer. Nous avons donc réalisé ce travail en trois étapes :

— choix de la méthode de rééquilibrage pour chaque modèle avec les meilleures performances prédictives

— comparaison des modèles sélectionnés entre eux

— analyse des valeurs des hypeparamètres sélectionnées

Ce processus a pour but de déterminer l'algorithme final avec les meilleures performances prédictives.

Le tableau 8 nous présente les performances des modèles avec la meilleure méthode de rééchantillonnage sélectionnée. Nous pouvons remarquer qu'aucun des algorithmes sans rééquilibrage

Algorithme	**Méthode de rééquilibrage**	**AUC-ROC**	F_1**-score**	**Balanced Accuracy**
Régression logistique	SMOTE	0.584	0.297	0.583
Arbres de décision	BorderlineSMOTE	0.605	0.302	0.586
Forêts aléatoires	**ADASYN**	**0.629**	**0.313**	**0.598**
Gradient Boosting	BorderlineSMOTE	0.643	0.273	0.573
Adaboost	ADASYN	0.645	0.298	0.583

TABLE 8 – Meilleures méthodes de rééquilibrage pour chaque algorithme optimisé et leurs performances associées (valeurs inférieures en rouge, supérieures en vert par rapport aux modèles initiaux)

des données n'a été choisi puisque les valeurs des métriques étaient inférieures par rapport aux modèles avec ré-échantillonnage. Cependant, nous pouvons noter que de nombreuses valeurs pour les métriques sont inférieures après optimisation, ce qui nous interroge sur le risque de sur-ajustement de notre modèle. Pour avoir plus d'informations sur les performances de chaque modèle optimisé, on pourra se référer aux tableaux situé en annexe D.
Après avoir analysé les différentes valeurs dans le tableau, nous allons privilégier l'algorithme des forêts aléatoires car c'est celui qui présente les meilleurs F_1-score et balanced accuracy. Nous pouvons analyser les paramètres de ce modèle dans le tableau 9. Les paramètres optimaux sont récupérés via la fonction `get_params()`. Le paramètre `sampling_strategy` d'ADASYN est fixé à 1.0, ce qui indique un équilibrage complet des classes, tandis que `n_neighbors` est à 5, suggérant une génération d'observations fictives basée sur les 5 plus proches voisins. Pour le classifieur RandomForest, le nombre d'estimateurs (`n_estimators`) est réglé à 300, ce qui offre une robustesse au modèle. La profondeur maximale des arbres (`max_depth`) est limitée à 3, contrôlant ainsi la complexité et réduisant le risque de **surapprentissage**, particulièrement pertinent pour notre étude puisque notre jeu de données contient relativement peu de lignes. La contrainte sur le nombre minimum d'échantillons pour diviser un nœud (`min_samples_split`) est fixée à 15, favorisant la stabilité des arbres, tandis que le critère de Gini est utilisé pour mesurer l'impureté. Le modèle est configuré pour utiliser le bootstrap avec un `random_state` fixé à 42, garantissant la reproductibilité des résultats. Ces paramètres montrent un équilibre entre complexité du modèle et prévention du surapprentissage.

Composant	**Paramètres retenus**
ADASYN	`sampling_strategy` : 1.0 `n_neighbors` : 5
RandomForestClassifier	`n_estimators` : 300 `max_depth` : 3 `min_samples_split` : 15 `min_samples_leaf` : 1 `criterion` : gini `bootstrap` : True `random_state` : 42

TABLE 9 – Paramètres retenus pour le modèle de forêts aléatoires avec ADASYN

Afin d'être sur de ne pas réaliser de sur-apprentissage pour ce modèle, nous allons analyser les courbes d'apprentissage et de validation (voir graphique 5).

Le graphique montre que le modèle de forêts aléatoires utilisant ADASYN surapprend initia-

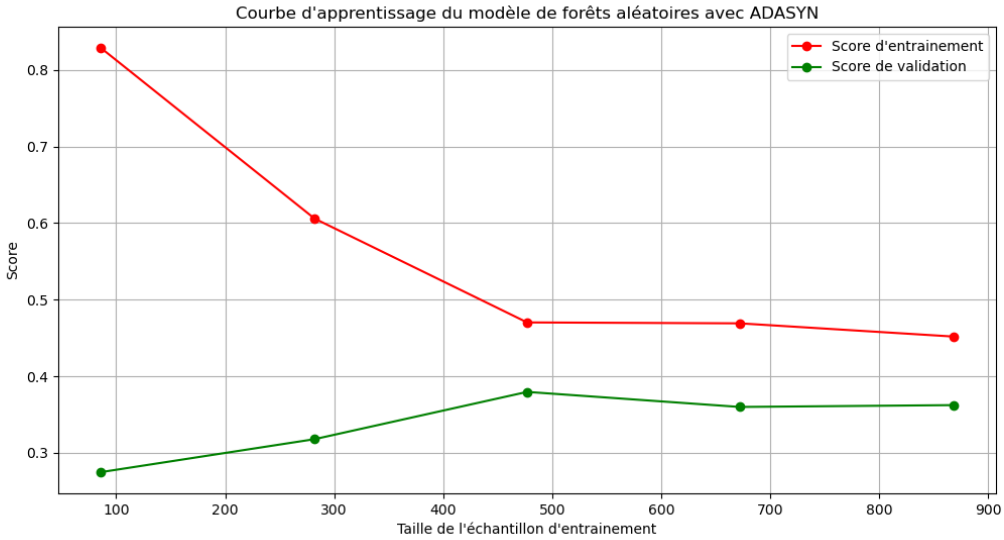


FIGURE 5 – Courbe d'apprentissage et de validation du modèle de forêts aléatoires avec ADASYN

lement avec une petite taille d'échantillon, comme en témoigne le score d'entraînement élevé (autour de 0.8) et le score de validation faible (environ 0.3). À mesure que la taille de l'échantillon augmente, le score d'entraînement diminue et se stabilise autour de 0.4 à 0.5, ce qui indique que le modèle commence à mieux généraliser, grâce aux paramètres choisis. Cependant, malgré cette amélioration, le score de validation reste relativement bas (entre 0.3 et 0.4), suggérant que le modèle continue de souffrir de **surapprentissage**, avec un écart persistant entre les scores d'entraînement et de validation. Cela montre que, bien que les paramètres réduisent le surapprentissage, ils ne permettent pas de détecter une structure/relation dans les données pour améliorer les performances de validation, indiquant que des ajustements supplémentaires des hyperparamètres pourraient être nécessaires pour optimiser la généralisation du modèle.
Par conséquent, nous avons décidé de tester une autre grille de paramètres pour ce modèle, afin de limiter le sur-apprentissage.

DEUXIÈME OPTIMISATION

Afin de limiter au maximum le sur-apprentissage, nous avons tenté d'optimiser notre modèle initial avec une grille d'hyperparamètres différente et moins précise. Le tableau 10 renseigne les différentes valeurs testées pour l'algorithme des forêts aléatoires.

Composant	Paramètres testés
RandomOverSampler	`sampling_strategy` : {0.5, 0.75, 1.0}
SMOTE	`k_neighbors` : {5, 7} `sampling_strategy` : {0.5, 0.75, 1.0}
ADASYN	`n_neighbors` : {5, 7} `sampling_strategy` : {0.5, 0.75, 1.0}
BorderlineSMOTE	`k_neighbors` : {5, 7} `sampling_strategy` : {0.5, 0.75, 1.0}
RandomForestClassifier	`n_estimators` : {50, 100, 300, 500} `max_depth` : {3, 6, 15, 21} `min_samples_split` : {10, 20} `min_samples_leaf` : {2, 4} `criterion` : {gini, entropy, log_loss} `bootstrap` : {True, False}

TABLE 10 – Grille d'hyperparamètres testés pour les modèles de forêts aléatoires avec différentes méthodes de suréchantillonnage (2ème optimisation)

Lors de cette étape, nous avons testé différentes variantes :

- diminution de `n_estimators` à 50, ce qui permet d'explorer des modèles avec moins d'arbres, ce qui peut limiter le risque de sur-ajustement.
- on ne teste pas la valeur maximale 27, ce qui privilégie les arbres moins profonds et donc la complexité du modèle
- un `min_samples_split` plus élevé (10 et 20) limite la division des noeuds dans les arbres, ce qui permet de limiter également le sur-apprentissage.
- nous imposons d'avoir 2 observations par feuille au minimum pour ne pas créer des arbres sur des échantillons très réduits.

Avec cette étape, nous avons généré des modèles moins complexes pour limiter le risque de sur-apprentissage. Les performances des différents modèles de forêts aléatoires, suite à cette seconde optimisation, sont à retrouver dans le tableau 11 et les courbes ROC en annexe **??**.

Algorithme	AUC-ROC	F1-score	Balanced Accuracy
Forêts aléatoires sans sur-échantillonnage	0.573	0.049	0.506
Forêts aléatoires avec SMOTE	0.623	0.284	0.567
Forêts aléatoires avec RandomOverSampler	0.600	0.267	0.553
Forêts aléatoires avec ADASYN	0.634	0.284	0.567
Forêts aléatoires avec BorderlineSMOTE	0.642	0.302	0.586

TABLE 11 – Performances des modèles de forêts aléatoires avec différentes méthodes de rééquilibrage (2ème optimisation)

Nous avons réalisé une étape de sélection de la meilleure méthode de rééquilibrage suite à cette optimisation et le modèle avec la méthode **BorderlineSMOTE** était la plus performante. Cependant, nous pouvons observer que les performances concernant le F_1-score et le balanced accuracy ne se sont pas améliorées par rapport à la recherche de paramètres avec une grille plus exhaustive.

Composant	Paramètres retenus
BorderlineSMOTE	`sampling_strategy` : 1.0 `k_neighbors` : 5 `m_neighbors` : 10
RandomForestClassifier	`n_estimators` : 500 `max_depth` : 3 `min_samples_split` : 20 `min_samples_leaf` : 2 `criterion` : log_loss `bootstrap` : False `random_state` : 42

TABLE 12 – Paramètres retenus pour le modèle de forêts aléatoires avec BorderlineSMOTE (2ème optimisation)

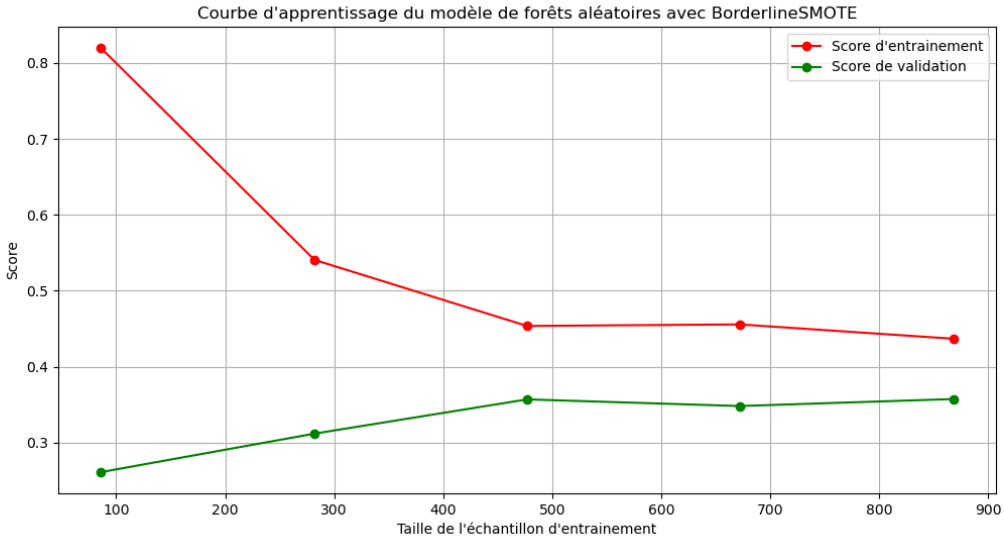


FIGURE 6 – Courbe d'apprentissage du modèle de forêts aléatoires avec BorderlineSMOTE (2ème optimisation)

Les deux courbes montrent que les modèles de forêts aléatoires atteignent une meilleure généralisation en utilisant plus de données. Cependant, BorderlineSMOTE semble offrir une performance légèrement plus stable par rapport à ADASYN (voir graphique 5), surtout en ce qui concerne la stabilité du score de validation après 500 échantillons. ADASYN montre une légère instabilité après avoir atteint un pic de performance, ce qui pourrait indiquer une petite tendance au sur-apprentissage pour certaines tailles d'échantillons.

En conclusion, nous allons choisir le modèle de forêts aléatoires construit lors de la premère optimisation (voir paramètres dans le tableau 9) pour poursuivre notre étude. En effet, les résultats présentés sont les plus optimaux malgré la légère instabilité présentée dans les graphiques des courbes d'apprentissage et de validation. Nous allons donc continuer notre étude avec le modèle de **forêts aléatoires avec la méthode de rééquilibrage ADASYN**.

6. Importance des variables

L'objectif final de cette étude est d'examiner les variables les plus influentes du modèle de forêts

aléatoires sélectionné afin de déterminer si certains facteurs semblent jouer un rôle déterminant dans la demande de financements des communes en faveur de la transition écologique.

Variable	**Importance**
`moyenne_conso_indus_hab`	0.122 396
`p_pop`	0.092 284
`friche`	0.083 144
`nb_actes_france_renov`	0.082 437
`gridens7`	0.076 110
`departement`	0.071 145
`total_entreprises`	0.069 423
`emissions_ges`	0.046 926
`part_inactifs`	0.045 149
`part_actifs`	0.041 877
`superf_choro`	0.041 647
`part_jeunes_sans_diplome`	0.033 485

TABLE 13 – Importance des variables avec le modèle de forêts aléatoires ADASYN

L'analyse des variables importantes révèle que la consommation industrielle moyenne d'électricité et de gaz par habitant (`moyenne_conso_indus_hab`) est le facteur le plus influent dans le modèle. Cela suggère que les communes avec une consommation industrielle d'électricité et de gaz par habitant plus élevée sont potentiellement plus motivées à demander des financements, probablement pour des projets visant à réduire leur impact environnemental. De même, la population (`p_pop`) apparaît comme un facteur significatif, indiquant que les communes plus peuplées pourraient avoir une plus grande demande de financements pour répondre à des besoins écologiques. D'autres variables, comme le nombre de friches (`friche`) et le nombre d'actes liés à la rénovation énergétique (`nb_actes_france_renov`), renforcent l'idée que les caractéristiques locales et les décisions, réalisées par les communes, de réhabilitation écologique peuvent également jouer un rôle important.
Cependant, il est important de nuancer ces résultats. Bien que certaines tendances se dégagent, la performance modeste du modèle laisse supposer que les relations entre les variables et la demande de financements sont complexes et potentiellement non linéaires. Le modèle actuel est capable de capturer certaines influences, mais il reste limité par des erreurs significatives. Cela pourrait être dû à un volume de données insuffisant, à une complexité des interactions entre les variables ou à d'autres facteurs exogènes non pris en compte, comme des éléments politiques ou des initiatives locales/régionales spécifiques.

En conclusion, les résultats montrent que des facteurs tels que la consommation d'électricité et de gaz industrielle par habitant, la population et le nombre de friches semblent influencer la demande de financements pour la transition écologique en Bretagne. Néanmoins, la performance globale du modèle, marquée par des signes de surapprentissage, indique qu'il faudrait explorer d'autres pistes pour améliorer les prévisions.

7. Analyse des résultats

Enfin, notre dernière étape a été d'identifier les communes les moins susceptibles de demander des financements. Cette phase nous a permis de déterminer s'il existait un phénomène purement géographique concernant les démarches réalisées pour demander des financements en faveur de la transition écologique.

Dans un premier temps, une cartographie interactive a été réalisée pour cette étude. Elle vise à offrir une vue d'ensemble des probabilités de demande de financements par les communes bretonnes pour la transition écologique. Ces probabilités ont été générées à partir des résultats d'un modèle de forêts aléatoires, appliqué sur un ensemble de données déséquilibrées. En effet, seules 203 communes parmi l'ensemble des données ont effectivement demandé des financements, ce qui a nécessité l'utilisation de la méthode de rééchantillonnage ADASYN pour tenter de compenser ce déséquilibre.

Cependant, malgré cette tentative d'ajustement, les performances du modèle restent relativement modestes, comme en témoignent les scores cités ci-dessus (modèle en gras dans le tableau 8). Ces résultats indiquent que le modèle a une capacité limitée à discriminer efficacement entre les communes susceptibles de demander des financements et celles qui ne le feront pas. Par conséquent, les prédictions présentées sur la carte doivent être interprétées avec prudence. Les zones en vert, représentant des probabilités élevées, pourraient inclure un certain nombre de faux positifs, tandis que des zones en rouge pourraient, au contraire, contenir des communes qui demanderont effectivement des financements.

Ainsi, bien que cette carte soit un outil visuel utile pour explorer les tendances régionales, elle ne doit pas encore être perçue comme une représentation exacte de la réalité. Les résultats doivent être complétés et d'autres pistes doivent encore être explorées pour l'utiliser comme un outil d'aide à la décision. La carte interactive est accessible via ce lien ou ci-dessous (voir figure 7), offrant un point de départ pour comprendre les dynamiques de demande de financements écologiques dans les communes de Bretagne.

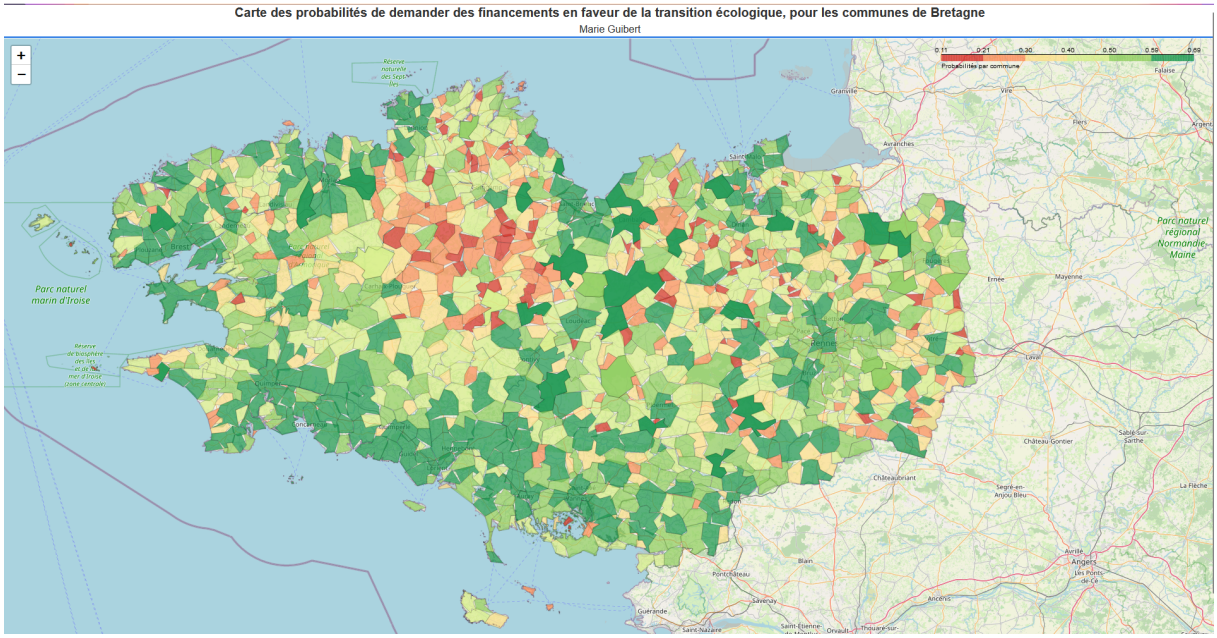


FIGURE 7 – Cartographie de la probabilité des communes à demander des financements en faveur de la transition écologique

Dans un second temps, nous avons classé les communes selon les 20 probabilités les plus faibles de demander des financements pour la transition écologique (voir tableau 14). Ces probabilités, issues du modèle de forêts aléatoires, varient entre 0.108 et 0.154, indiquant pour chaque commune la probabilité qu'elle sollicite des financements. Par exemple, une probabilité de 0.108 pour Saint-Maden signifie qu'il y a environ 10,8% de chances que cette commune fasse une demande de financement. Ces faibles probabilités suggèrent un moindre investissement potentiel de ces

communes, influencé par divers facteurs socio-économiques ou environnementaux. Cependant, étant donné que ces résultats proviennent d'un modèle dont la fiabilité est limitée, il est crucial de les interpréter avec prudence. Cette situation peut entraîner la présence de faux négatifs dans les prédictions, c'est-à-dire des communes susceptibles de demander des financements mais non identifiées comme telles par le modèle. Par conséquent, ces résultats doivent être interprétés avec prudence et considérés comme des estimations plutôt que des certitudes absolues.

Par ailleurs, nous avons comparé ces probabilités avec les émissions de gaz à effet de serre par habitant. Bien que l'importance de cette variable dans notre modèle soit faible (0.047), elle pourrait avoir un rôle crucial dans le processus de décision des politiques publiques. Si certaines communes ont de faibles probabilités de demander des financements mais des émissions de gaz à effet de serre par habitant élevées, cela souligne la nécessité de renforcer la sensibilisation et la communication auprès de ces communes concernant la transition écologique. Même si cette variable n'est pas déterminante dans notre modèle, elle reste essentielle pour une prise de décision éclairée.
Pour effectuer cette comparaison, nous avons calculé la médiane des émissions de gaz à effet de serre pour les communes de Bretagne, qui s'élève à 8.27 tonnes équivalent CO2 pour 2021. L'analyse de ce tableau révèle un constat frappant : 15 communes sur 20 dépassent ce seuil, ce qui soulève de véritables interrogations. Effectivement, cette variable présente un faible impact dans notre modèle alors qu'il faudrait clairement qu'elle influe sur les résultats dans une société idéale et soucieuse de l'environnement.

En conclusion, cette étape rapproche notre étude des enjeux des politiques publiques en transformant une simple prédiction en une problématique concrète de politique publique. L'objectif est désormais de mobiliser les communes qui présentent une faible probabilité de demander des subventions pour la transition écologique. Ce projet, sous réserve d'une amélioration des performances du modèle, pourrait devenir un outil précieux pour les agents de l'Etat pour cibler les communes à "risque".

Code INSEE	Commune	Probabilité	Emissions de gaz à effet de serre par habitant
22312	Saint-Maden	0.108	17.117
22335	Plouzelambre	0.116	19.489
22126	Le Leslay	0.117	25.471
22274	Saint-André-des-Eaux	0.117	9.532
22057	Le Faouët	0.118	6.325
22317	Saint-Méloir-des-Bois	0.119	6.974
22289	Saint-Fiacre	0.132	15.921
35026	Bléruais	0.133	15.663
35336	Le Tiercent	0.142	5.704
22023	Bulat-Pestivien	0.142	13.696
22335	Senven-Léhart	0.143	14.536
35153	Lillémer	0.144	8.246
22322	Saint-Péver	0.144	10.906
35261	Saint-Christophe-de-Valains	0.147	7.183
22036	La Chapelle-Blanche	0.147	12.881
22378	Trévérec	0.148	20.903
22108	Lanleff	0.150	3.631
56019	Billio	0.150	11.642
29116	Lanneuffret	0.154	20.806
22018	Brélidy	0.154	8.680

TABLE 14 – Probabilités de demande de financements et émissions de GES par habitant pour les communes sélectionnées

2.1.6 Conclusion, limites et perspectives

CONCLUSION

Dans notre cadre de recherche des facteurs influençant la demande de financements des communes en Bretagne pour la transition écologique, les résultats obtenus grâce aux différents modèles de classification montrent une amélioration des performances lorsqu'on applique des techniques de rééquilibrage des classes. Ce constat est particulièrement pertinent dans notre contexte où le nombre de communes demandant des financements est bien inférieur à celui des communes ne faisant pas les démarches.

Le modèle de **forêts aléatoires** utilisant la méthode **ADASYN** s'est distingué comme étant le plus performant des algorithmes testés. Cette technique a permis d'identifier les communes ayant une faible probabilité de demander des financements, avec un AUC-ROC de 0.629, un F_1-score de 0.313, et une balanced accuracy de 0.598. Cependant, les performances de notre modèle signifient qu'il n'est pas être encore prêt à être généraliser puisqu'il souffre de surajustement malgré les différentes étapes mises en place. L'absence de rééquilibrage des classes a conduit à des performances inférieures, ce qui démontre l'importance d'utiliser ces techniques dans un contexte où les données sont fortement déséquilibrées. En l'absence de rééquilibrage, le modèle a montré une capacité encore plus limitée à prédire correctement les communes qui sont le moins susceptibles de demander des financements.

En conclusion, notre démarche pourrait être utile pour les décideurs publics, car cela permet de cibler les communes qui pourraient bénéficier de davantage de sensibilisation ou d'incitations pour s'engager dans des initiatives écologiques. L'application de modèles bien calibrés avec des

techniques de rééquilibrage comme ADASYN offre pourraient aussi permettre d'anticiper les demandes de financement des communes **bretonnes**. Cette étude ne se limite pas à la simple réalisation de prédictions ; elle constitue également un véritable outil de ciblage et d'anticipation, permettant d'identifier les communes à mobiliser en priorité.

Limites

Lors de ce projet de modélisation de la demande de financements des communes en Bretagne pour la transition écologique nous nous sommes heurtés à plusieurs limites. Tout d'abord, les performances et la capacité de généralisation de notre modèle sont encore assez limitées. Cette observation nous contraint donc à continuer d'explorer de nouvelles pistes pour la suite de ce projet. Par ailleurs, l'accès compliqué aux données au niveau communal a influencé les variables choisies pour construire les modèles et donc la fiabilité de la modélisation. Surmonter ces limites en renforçant les ressources et en améliorant l'accès aux données permettrait d'améliorer les analyses futures. Aussi, à ce jour, l'absence d'un poste pérenne au sein du SGAR en data science a ralenti mon avancée et mon apprentissage de nouvelles connaissances. De plus, la puissance de calcul limitée a affecté l'exécution de certains algorithmes. Accéder à des plateformes de calcul comme Nubonyxia pourrait améliorer cette situation.

Perspectives

Concernant les perspectives de ce projet, dans un premier temps, nous envisageons de nouvelles pistes pour maximiser la capacité de généralisation de notre modèle. Nous avons mis en place différentes méthodes pour tenter de limiter ce risque (modification des grilles d'hyperparamètres, régularisation, augmentation du nombre de blocs dans la validation croisée) mais il reste encore des possibilités à envisager. Nous pourrions élargir la base de données à la France entière pour ne pas se limiter à la Bretagne afin d'augmenter le nombre de lignes dans notre base de données. Concernant cette piste, j'ai pu commencer à échanger avec les agents en charge de l'API du fonds vert pour avoir accès aux données prochainement. Une autre solution est d'avoir un historique plus important, ce qui n'était pas possible pour cette étude car nous disposions seulement des données pour l'année 2023. Cependant, l'acquisition des données pour l'année 2024 pourrait être une piste d'amélioration. Cette augmentation permettrait peut être de mieux capturer les relations entre les variables. Une autre possibilité est d'intégrer de nouvelles variables comme par exemple le taux de pauvreté ou la part des 65 ans et plus. Ces nouvelles informations pourraient être utiles dans la modélisation. Enfin, il pourrait s'avérer intéressant de réaliser une sélection de variables pour ne capturer que les effets réels des variables et non pas le bruit.
Dans un second temps, nous voudrions étendre nos analyses à d'autres programmes financiers afin de mieux comprendre les facteurs qui influencent les demandes de subventions dans divers domaines. Cette approche nous permettra peut-être d'adapter les politiques publiques en conséquence. De plus, nous souhaitons développer une interface utilisateur qui permettrait aux agents et décideurs de constituer leur propre base de données en sélectionnant les variables d'intérêt. Cette interface offrirait également la possibilité d'entraîner les modèles prédictifs sur ces données personnalisées, facilitant ainsi l'analyse et la prise de décision en temps réel. Ces développements contribueront à rendre les outils d'analyse plus accessibles et adaptés aux besoins spécifiques de chaque utilisateur. Pour l'instant, ces idées restent à l'état de projet, et faute de temps, nous n'avons pas pu les mettre en oeuvre, elles seront donc à rediscuter avec le nouveau data scientist.

2.2 Missions internes

Dans l'ensemble de l'administration publique, l'acculturation à la donnée se heurte à des défis majeurs, souvent liés à la sensibilité des données. On remarque aussi un manque d'outils permettant la réutilisation facile des données et une réticence de personnes ou organisations à partager ou à favoriser une meilleure interconnaissance de la donnée.

2.2.1 Data Etat

Tout d'abord, le projet majeur de l'équipe a émergé en février 2023, avec la création de **Data Etat** : nouvelle infrastructure de partage et de réutilisation sécurisée de la donnée de l'Etat.
Cet outil a pour but de lever le voile sur l'information, principalement financière, pour pouvoir permettre la réutilisation de ces données dans le cadre professionnel. En effet « les services ordonnateurs [...] déplorent le manque d'accès aux données ou le caractère « peu partagé » des informations ». Par exemple, un sous-préfet n'était pas en capacité sur son territoire de savoir ce que l'Etat finance. Notre objectif est donc de permettre progressivement la « libération » de la donnée, au sein de l'administration régionale bretonne dans un premier temps.

Présentation générale

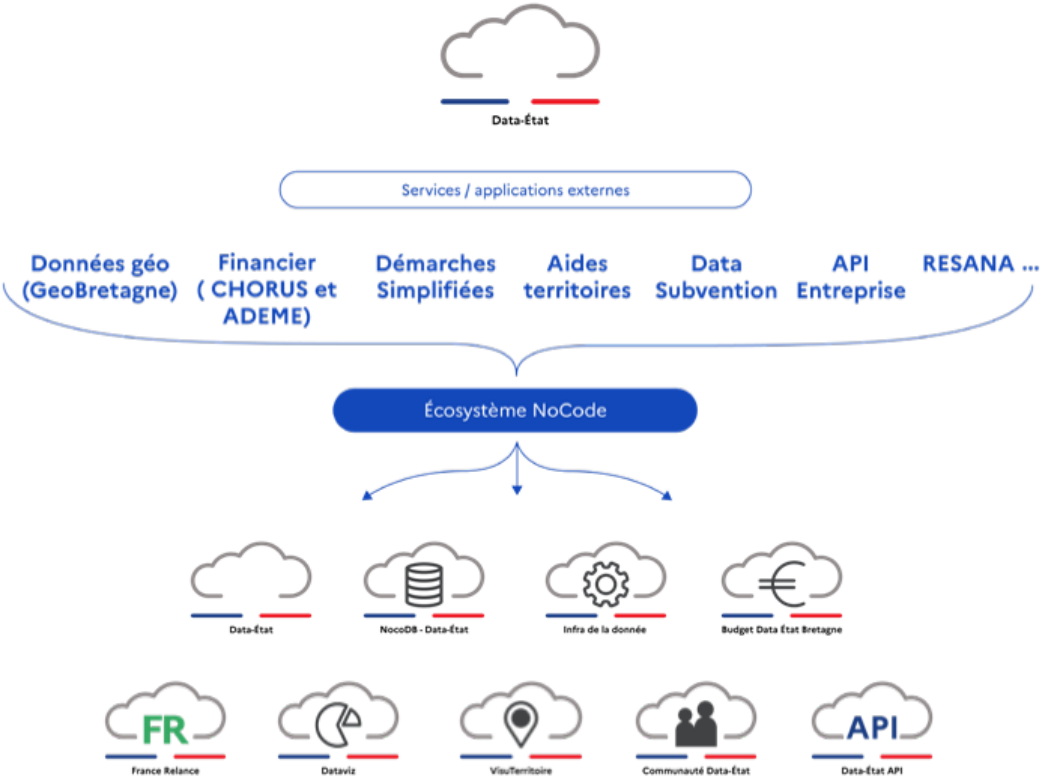
Data État est un projet interministériel piloté par le SGAR Bretagne (Préfecture de région). Son objectif est d'aider au pilotage des politiques publiques, en valorisant les financements de l'État sur les territoires.

Lors de mon alternance, la compréhension des enjeux autour du projet Data Etat faisait parti de mes objectifs. Cela a demandé une phase d'immersion du côté technique mais aussi politique. Il existe quatre plateformes/ applications principales, et une en construction, dans ce projet :

- Budget Data Etat
- France Relance
- Visu'Territoire
- Data'Viz grâce à Superset
- Data QPV (en construction)

Les données

Au sein de la région Bretagne et grâce à l'équipe Data, une avancée significative a été réalisée, permettant l'accès aux données financières de l'Etat (CHORUS). Cette base de données est progressivement complétée par les données financières des opérateurs de l'Etat (exemple : ADEME).



Les données CHORUS sont des données financières et particulièrement complexes (données comptables). Elles sont rendues intelligibles et fonctionnelles pour les agents de l'Etat grâce à Data Etat. Nous avons à notre disposition l'historique des données depuis 2019. Elles sont mises à jour mensuellement. Les domaines recensés sont très variés, allant de la justice, à l'écologie en passant par l'économie... Les natures des actions financées sont aussi diverses : marché public, dotation aux collectivités, subventions associatives.... Le panel de données est donc très large, ce qui permet une analyse approfondie des tendances et des évolutions dans divers domaines.

Nous avons d'autres sources de données complétant la base de données de Data Etat (Data Subvention, Démarches Simplifiées, API Entreprise) grâce à leur API, ce qui nous permet d'avoir des informations qualitatives sur les données financières. Par ailleurs, nous avons à notre disposition les données financières concernant le plan de Relance et les données de France 2030. Et enfin, nous utilisons divers référentiels afin d'avoir un cadre structuré et universel. Toutes les données des référentiels utilisés sont ensuite mises à disposition via l'API de Data Etat.

Communication et encadrement

Dans ce projet, mon rôle n'a pas seulement été technique, mais a également intégré une dimension « communication », car l'outil Data Etat vise à être utilisé par un maximum d'agents afin de créer un « commun numérique » et éviter la multiplication des outils et plateformes internes.
Étant donné que le domaine des données est souvent perçu comme complexe par de nombreux agents des administrations françaises, j'ai travaillé à le rendre plus accessible en organisant des quizz ou des "ice-breakers" en début de réunion. Cette initiative de communication visait également à promouvoir l'utilisation de Data Etat.

Aussi, pour faciliter la prise en main de l'outil, Maëlys GLORO et moi-même avons élaboré

un tutoriel pour guider les utilisateurs sur la plateforme Data Viz'. Cette mission est primordiale pour rendre accessible notre solution à un large public. Le tutoriel est construit avec des captures d'écran, des petites vidéos et des commentaires pour simplifier la compréhension.

Dans le même esprit, nous avons organisé des ateliers pour que les utilisateurs puissent tester la plateforme Data Etat (Budget et Data Viz'). Mon rôle était d'encadrer et de guider les agents, ce qui nécessitait une connaissance approfondie des outils Budget Data Etat et Data Viz' afin de pouvoir expliquer leur intérêt et leur utilité. L'objectif final était de rassembler un maximum d'utilisateurs et d'opérateurs (Banque des Territoires, l'Agence de l'eau, l'Agence Régionale de Santé, l'Agence Nationale du Sport, etc.) dans ce projet, pour centraliser l'information financière.

Fonction support
Par ailleurs, mon rôle dans ce projet a été à la fois une fonction de support et d'analyse.
Au cours de cette année, la plateforme a connu de nombreuses évolutions en réponse aux besoins des utilisateurs. Ces améliorations étaient réalisées selon une méthodologie de sprints mensuels, durant lesquels le SIB traitait les demandes d'évolution avant de les livrer et de les mettre en production. L'une de mes responsabilités était de tester ces améliorations et de signaler les anomalies. Ces tests étaient réalisés en collaboration avec Maëlys GLORO, afin de combiner une perspective utilisateur avec une approche technique.

Il m'était demandé d'être force de propositions en suggérant des améliorations pour la plateforme, telles que l'exploration de nouvelles API. Par exemple, j'ai exploré l'API de Grist avec comme objectif d'automatiser le traitement de certains fichiers pour ensuite les renvoyer sur la plateforme. Grist étant une sorte de "tableur partagé" qui stocke les données sous forme de bases de données, cette proposition a été très intéressante pour le projet. L'objectif final est d'automatiser les mises à jours des tableaux de bord dans Superset pour que les agents puissent suivre en direct les données.

Travail sur les données
Tout d'abord, j'ai du constitué l'historique de nos référentiels de Budget Data Etat et les mettre à jour afin d'avoir une plateforme la plus actualisée possible. La mise à jour des référentiels consiste à récupérer les données via différentes sources (API ou fichiers CSV) puis à joindre les fichiers en fonction de différents codes. Cette tâche nous permet d'historiser les différentes nomenclatures pour avoir tous les programmes financiers disponibles depuis 2019. Ce travail, réalisé sur R, a nécessité beaucoup de rigueur et de compréhension de la donnée.

Durant cette année j'ai réalisé un travail sur le base de données du projet France 2030. L'objectif a été de réaliser une vue SQL pour unifier des tables provenant de différentes sources de données. Durant ce projet, j'ai du être à l'écoute des besoins des utilisateurs pour essayer de répondre au maximum à leurs attentes concernant des statistiques descriptives. Cette production a été conçue avec l'outil Apache Superset et consultable via l'outil Data Viz'.

Dans le cadre de Data QPV, j'ai réalisé de la lecture automatique de documents. L'objectif du projet est de requalifier des financements de l'Etat qui ont bénéficié à des Quartiers Prioritaires de la Ville. Actuellement, Budget Data Etat localise les financements en fonction du SIRET du bénéficiaire. Par exemple si le SIRET d'une association se trouve dans le quartier de Villejean à Rennes, on dit que ce financement a bénéficié au quartier prioritaire de Villejean. Les données CHORUS ne nous permettent pas de localiser les actions qui ont bénéficié à ces quartiers si le SIRET du bénéficiaire n'est pas directement localisé dans le quartier. Cependant, certains bénéficiaires ne sont pas directement implantés dans les quartiers et financent des actions pour

les QPV. Mon but ici a été de lire les CERFA (formulaires administratifs) qui renseignent des informations sur les paiements et pour en ressortir les territoires concernés par les contrats de ville.
Pour se faire, j'ai mis en place un script python qui lit automatiquement tous les fichiers des dossiers concernés par les contrats de ville et qui réalise une correspondance avec les fichiers ressources contenants les quartiers prioritaires des villes concernées.

2.2.2 PeATE ou le parapheur électronique

Le concept du parapheur électronique (PeATE) s'inscrit pleinement dans l'ère actuelle. Un outil autrefois très répandu, le parapheur traditionnel, tend à céder la place à sa version électronique. Cette innovation vise divers objectifs, tels que la réduction de la consommation de papier dans une perspective écologique, l'accélération des processus de validation, la transparence entre les agents de l'Etat, et l'adaptation aux nouvelles méthodes et structures de travail.

Mon rôle a été de recueillir les informations sur les parapheurs électroniques à l'aide de l'API Ixbus. Celle-ci concentre toutes les données nécessaires et permet le calcul de divers indicateurs. Mon objectif a été d'automatiser l'extraction des données pour pouvoir calculer les indicateurs de suivi du parapheur et de sobriété énergétique.
Pour ce travail, j'ai utilisé python et employé des méthodes pour paralléliser les requêtes à l'API car la récupération des données peut être très lente. En effet, il faut se connecter à l'API avec environ 40 tokens différents puisqu'ils correspondent aux diverses instances qui utilisent le parapheur électronique. La parallélisation des requêtes est donc indispensable car les indicateurs concernent toutes les administrations et il faut donc interroger l'API à chaque fois avec une identification différente.

2.2.3 Autres projets

Visualisation de données
Cette alternance a été source de nombreux projets de visualisation de données.

Afin de répondre à divers besoins, j'ai conçu des **cartographies** :

- Pour localiser les tiers-lieux disponibles afin de favoriser le travail à distance des agents publics mais aussi par souci de sous-occupation des locaux publics. Cette cartographie utilisait des données RH sur les résidences administratives et personnelles des agents. L'objectif était d'avoir une vision de la répartition des résidences personnelles des agents par rapport à leur lieu de travail et pour identifier les locaux occupés par les services de l'Etat susceptibles d'accueillir des tiers lieux en proximité des résidences personnelles.
- Pour localiser les employeurs publics en Bretagne dans l'objectif de renforcer l'attractivité du service public et à mieux faire connaître les 1 000 et 1 métiers de la fonction publique. L'objectif, à terme, est donc de pouvoir mettre en visibilité auprès des agents publics, des salariés, des demandeurs d'emploi, des étudiants et de tous les organismes les accompagnant (France travail, missions locales, conseillers d'information et d'orientation de l'éducation nationale...) la diversité et la proximité des employeurs publics dans un périmètre géographique proche. Une meilleure connaissance de l'écosystème de l'emploi public local permettrait de rapprocher l'offre et la demande d'emplois publics en proposant, d'une part, aux agents des parcours professionnels variés et en proximité, et, d'autre part, aux employeurs de recruter des candidats au profil plus riche et expérimenté du fait de ces parcours.

A la demande du SGAR et du préfet, j'ai aussi pu concevoir des **tableaux de bord** pour analyser certaines données comme par exemple le suivi des politiques publiques prioritaires. Ces

visualisations de données permettent de mettre en lumière des informations pertinentes comme les politiques à prioriser dans la région Bretagne. Elles nous indiquent si l'avancée des politiques est plutôt en accord avec le national ou non. De plus, cet outil de suivi est ergonomique et permet d'avoir les informations essentielles très rapidement mais reste sécurisé puisqu'il n'est accessible qu'aux agents de l'Etat. Pour automatiser la mise à jour de ce tableau de bord, nous avons échangé avec la Direction de la Transformation et de l'Innovation Publique (DITP) pour avoir directement accès à l'API qui fournit les données en question. Ce travail est donc un exemple d'une preuve de concept réalisée au local qui permet d'ouvrir des discussions avec le national.

RGPD et catalogue de données
Dans le cadre de la gestion des données du SGAR Bretagne, Angélina DENOUAL et moi-même avons réalisé un recensement des données soumises au Règlement Général sur la Protection des Données.
Cette tâche impliquait une identification rigoureuse et systématique de l'ensemble des données personnelles collectées, traitées, et stockées par le SGAR. Nous avons collaboré étroitement avec divers services pour comprendre les données personnelles ou sensibles utilisées dans leur quotidien. L'objectif pour ces services a été de remplir, en autonomie, un questionnaire créé avec Grist pour que nous puissions récupérer les informations sous forme de base de données. Ce processus nous permet d'avoir une gestion des données plus efficace et l'actualisation de celle-ci sera simplifiée.

Par ailleurs, ce projet nous a aussi permis de recenser les données utilisées au sein du SGAR. Notre second objectif a été de renseigner les données dans le catalogue GéoBretagne afin de permettre un accès aux données facilité. Le formulaire mis en place pour analyser les données soumises au RGPD a été tourné de sorte à avoir une "entrée par la donnée". En effet, nous avons questionné les agents sur les données qu'ils produisaient, utilisaient et échangeaient dans leur quotidien. Afin de faciliter le travail des agents, le catalogue GéoBretagne a été mis en place et notre objectif est de favoriser son utilisation.

En somme, ce projet nous a permis de développer une compréhension approfondie des enjeux liés à la protection des données dans un contexte administratif. Nous avons aussi renforcé notre expertise en matière de réglementation, de gestion des données et nous avons pris conscience de l'importance d'avoir des données disponibles et trouvables pour faciliter le travail des agents.

3 Conclusions et perspectives

Pour conclure, cette étude a marqué une étape importante dans l'application de l'intelligence artificielle pour le pilotage des politiques publiques. En développant un modèle de prédiction des demandes de financements pour la transition écologique par les communes, j'ai défriché un terrain largement inexploré jusqu'à présent. Il est surprenant de constater qu'en 2024, personne n'avait encore réalisé ce type de travail. Pourtant, les retours montrent que ce sujet suscite un grand intérêt parmi les décideurs publics (SGAR, DREAL...) et les chercheurs. Mon approche a permis de démontrer qu'un tel modèle, bien que perfectible, peut servir d'outil de ciblage pour anticiper les besoins des communes et orienter les ressources de manière plus efficace.

Il est évident que l'intelligence artificielle offre un potentiel immense pour améliorer l'efficacité des politiques publiques. Cependant, pour que l'ambition d'une IA souveraine puisse se concrétiser, il est crucial de disposer d'une infrastructure technologique adéquate. La puissance de calcul, actuellement insuffisante, représente un obstacle majeur à surmonter. Une IA souveraine ne pourra pleinement réaliser son potentiel que si elle est soutenue par des capacités de calcul robustes et indépendantes, permettant de déployer à grande échelle des modèles prédictifs et génératifs au service des agents publics.

Par ailleurs, mon intégration dans l'équipe du projet Data Etat a été une expérience particulièrement enrichissante. Au sein de cette équipe, j'ai pu contribuer à des projets de grande envergure, tout en développant mes compétences en gestion de données et en modélisation prédictive. Travailler sur un projet comme celui-ci m'a permis de comprendre les enjeux techniques et stratégiques liés à l'utilisation et la valorisation des données dans le secteur public.

Sur le plan personnel, cette alternance m'a offert des opportunités de croissance tant professionnelle que personnelle. J'ai appris à naviguer dans des environnements complexes, à collaborer avec des équipes pluridisciplinaires, et à développer des solutions innovantes face à des défis réels. Mon travail sur ces projets m'a non seulement permis de consolider mes compétences techniques, mais m'a également donné une meilleure compréhension des dynamiques liées à la mise en œuvre des politiques publiques. Cette expérience a confirmé mon intérêt pour l'IA et m'a permis d'explorer un autre point de vue pour réaliser des statistiques, celui des agents de l'Etat.

Références

[1] Frédéric Gosselin, *Propositions pour améliorer l'équipement biométrique du détective écologique - Application à la modélisation de la relation entre gestion forestière et biodiversité*, Mémoire d'Habilitation à Diriger des Recherches, Université Pierre et Marie Curie Paris, 2011. Disponible en ligne : `https://theses.hal.science/tel-02594486v1`.

[2] Julien Thomas, *Apprentissage supervisé de données déséquilibrées par forêt aléatoire*, Thèse, 2009, disponible en ligne : `https://theses.hal.science/tel-01540283`.

[3] Laurent Rouvière, *Cours d'Apprentissage Supervisé - Notes de Cours*, 2024. Disponible en ligne : `https://lrouviere.github.io/page_perso/cours/docs/machine_learning/cours_app_sup_R2_article.pdf`.

[4] Laurent Rouvière, *Données déséquilibrées*, 2023. Disponible en ligne : `https://lrouviere.github.io/page_perso/cours/docs/machine_learning/cours_article_DD.pdf`.

[5] Roberto D'Ambrosio, *Handling Imbalanced Datasets by Reconstruction Rules in Decomposition Schemes*, Thèse, Université Nice Sophia Antipolis ; Università Campus Bio-Medico di Roma, 2014. Disponible en ligne : `https://theses.hal.science/tel-00995021v1`.

Sources des données

— INSEE. *Codes des communes de France*. A retrouver sur `https://www.insee.fr/fr/information/5057840`

— Data.gouv.fr. *Consommation annuelle d'électricité et de gaz par commune et par secteur d'activité (jusqu'en 2021)*. A retrouver sur `https://www.data.gouv.fr/fr/datasets/consommation-annu`

— Agence ORE. *Consommation annuelle de gaz et d'électricité par secteur d'activité (détail)*. A retrouver sur `https://opendata.agenceore.fr/explore/dataset/conso-elec-gaz-annuelle-par-s information/`

— Observatoire des Territoires. *Émissions totales de GES par secteur*. A retrouver sur `https://www.observatoire-des-territoires.gouv.fr/emissions-totales-de-gaz-effet-de-serre-par-s`

— ADEME. *Conseillers en énergie partagée*. A retrouver sur `https://data.ademe.fr/datasets/sare_tbs`

— Data.gouv.fr. *Base de données nationale des bâtiments*. A retrouver sur `https://www.data.gouv.fr/fr/datasets/base-de-donnees-nationale-des-batiments/#/resources`

— Data.gouv.fr.*Sites référencés dans Cartofriches*. A retrouver sur `https://www.data.gouv.fr/fr/datasets/sites-references-dans-cartofriches/#/resources`

— Data.gouv.fr. *Base des EcoQuartiers*. A retrouver sur `https://www.data.gouv.fr/fr/datasets/base-des-ecoquartiers/#/resources`

— GeoBretagne. *Nombre de résidences secondaires*. A retrouver sur `https://geobretagne.fr/datahub/dataset/insee_rp_hist_1968.part_resid2`

— Observatoire des Territoires. *Communes bénéficiaires des programmes ANCT (ACV, PVD...)*. A retrouver sur `https://www.observatoire-des-territoires.gouv.fr/villages-davenir-communes-`

— Data.gouv.fr. *Liste des communes couvertes par les CRTE*. Retrieved from `https://www.data.gouv.fr/fr/datasets/contrat-de-relance-et-de-transition-ecologique/`

— Data.gouv.fr. *Liste des communes soumises à la loi littoral*. A retrouver sur `https://www.data.gouv.fr/fr/datasets/communes-de-la-loi-littoral-au-code-officiel-geographique-cog-2 #/resources`

— Observatoire des Territoires. *Grille communale de densité en 7 niveaux*. A retrouver sur `https://www.observatoire-des-territoires.gouv.fr/grille-communale-de-densite-en-7-niveau`

— Observatoire des Territoires. *Superficie de la commune*. A retrouver sur `https://www.observatoire-des-territoires.gouv.fr/superficie`

— GeoBretagne. *Liste des gares*. A retrouver sur `https://geobretagne.fr/datahub/dataset/4a9d13f7-17be-4a98-9f8f-907cf223072f`

— Observatoire des Territoires. *Part des déplacements domicile-travail en voiture*. A retrouver sur `https://www.observatoire-des-territoires.gouv.fr/outils/cartographie-interactive/#c=indicator&i=insee_rp_hist_xxxx.part_domtrav_voit&s=2020&view=map59`

— Observatoire des Territoires. *Médiane du revenu disponible*. Retrieved from `https://www.observatoire-des-territoires.gouv.fr/mediane-du-revenu-disponible-par-uc`

— Observatoire des Territoires. *Population au dernier recensement*. Retrieved from `https://www.observatoire-des-territoires.gouv.fr/population-au-dernier-recensement`

— Observatoire des Territoires. *Nombre d'actifs de 15-64 ans*. Retrieved from `https://www.observatoire-des-territoires.gouv.fr/nombre-dactifs-de-15-64-ans`

— Observatoire des Territoires. *Nombre d'inactifs de 15-64 ans*. Retrieved from `https://www.observatoire-des-territoires.gouv.fr/nombre-dinactifs-de-15-64-ans`

— Data.gouv.fr. *CSP du maire*. Retrieved from `https://www.data.gouv.fr/fr/datasets/repertoire-national-des-elus-1/#/resources`

— GeoBretagne. *Taux d'endettement des communes*. Retrieved from `https://geobretagne.fr/datahub/dataset/finances_ep_dette.com_variation_encours_dette_ha_pct`
— Observatoire des Territoires. *Indicateur de dépendance économique*. Retrieved from `https://www.observatoire-des-territoires.gouv.fr/indicateur-de-dependance-economique`
— Observatoire des Territoires. *Taux d'abstention aux municipales (1er tour)*. Retrieved from `https://www.observatoire-des-territoires.gouv.fr/municipales-taux-dabstention-au-1er-tou`
— Observatoire des Territoires. *Taux de création des entreprises*. Retrieved from `https://www.observatoire-des-territoires.gouv.fr/taux-de-creation-dentreprises`
— Observatoire des Territoires. *Nombre d'entreprises par secteurs d'activité*. Retrieved from `https://www.observatoire-des-territoires.gouv.fr/nombre-dentreprises-par-secteurs-dactiv`
— Observatoire des Territoires. *Nombre de licenciés sportifs*. Retrieved from `https://www.observatoire-des-territoires.gouv.fr/nombre-de-licencies-sportifs`
— Observatoire des Territoires. *Part des 20-24 ans sans diplôme*. Retrieved from `https://www.observatoire-des-territoires.gouv.fr/part-des-20-24-ans-sans-diplome`

A Annexe 1 : Liste des graphiques et tableaux

TABLE 15: Liste des graphiques et tableaux

B Annexe 2 : Tableau des données

Nom de la variable	Détail	Nature de la variable
moyenne_conso_agri_hab	Consommation annuelle d'électricité et de gaz (secteur agricole)	variable continue
moyenne_conso_indus_hab	Consommation annuelle d'électricité et de gaz (secteur industrie)	variable continue
moyenne_conso_tertiaire_hab	Consommation annuelle d'électricité et de gaz (secteur tertiaire)	variable continue
moyenne_conso_residentiel_hab	Consommation annuelle d'électricité et de gaz (secteur résidentiel)	variable continue
moyenne_conso_totale_hab	Consommation annuelle d'électricité et de gaz (tous secteurs confondus)	variable continue
emissions_ges	Emission de gaz à effet de serre	variable continue
nb_actes_france_renov	Nombre d'actes en énergie partagée	variable discrète
friches	Nombre de friches	variable discrète
ecoquartiers	Présence d'écoquartiers	variable binaire
part_residences_secondaires	Part des résidences secondaires	variable continue
beneficiaire_prog	Bénéficiaires des programmes ANCT	variable binaire
climat	Climat de la commune	variable qualitative
gridens7	Densité	variable qualitative
superf_choro	Superficie	variable continue
departement	Département de la commune	variable qualitative
gare_tgv	Présence d'une gare TGV	variable binaire
part_trajets_voiture	Part des déplacements domicile-travail	variable continue
med_disp	Médiane du revenu disponible	variable continue
p_pop	Population	variable continue
CSP_maire	CSP du maire	variable qualitative
com_variation_encours_dette_hab_pct	Taux d'endettement de la commune	variable continue
part_actifs	Part d'actifs	variable continue
part_inactifs	Part d'inactifs	variable continue
dependance_eco	Dépendance économique de la commune	variable discrète
abstention_municipales	Taux d'abstention aux élections municipales	variable continue
taux_creation_ent	Taux de création d'entreprises	variable continue
total_entreprises	Nombre total d'entreprises	variable discrète
part_licencies_sportifs	Part des licenciés sportifs	variable continue
part_jeunes_sans_diplome	Part des jeunes sans diplôme	variable continue

Table 16 – Liste et descriptions des variables pour la modélisation statistique

C Annexe 3 : Performances des modèles initiaux

C.1 Tableaux des performances des modèles initiaux

	AUC-ROC	F_1-score	Balanced Accuracy
Régression logistique sans sur-échantillonnage	0.600	0.050	0.509
Régression logistique avec SMOTE	0.589	0.260	0.542
Régression logistique avec RandomOverSampler	0.593	0.188	0.530
Régression logistique avec ADASYN	0.579	0.273	0.554
Régression logistique avec BorderlineSMOTE	0.595	0.283	0.568

TABLE 17 – Performances des différents modèles initiaux de régression logistique (avec valeurs maximales en magenta)

	AUC-ROC	F_1-score	Balanced Accuracy
Arbre de décision sans sur-échantillonnage	0.647	0.203	0.542
Arbre de décision avec SMOTE	0.620	0.300	0.586
Arbre de décision avec RandomOverSampler	0.603	0.105	0.499
Arbre de décision avec ADASYN	0.553	0.248	0.524
Arbre de décision avec BorderlineSMOTE	0.589	0.284	0.563

TABLE 18 – Performances des différents modèles initiaux d'arbre de décision (avec valeurs maximales en magenta)

	AUC-ROC	F_1-score	Balanced Accuracy
Forêts aléatoires sans sur-échantillonnage	0.583	0.000	0.493
Forêts aléatoires avec SMOTE	0.631	0.219	0.540
Forêts aléatoires avec RandomOverSampler	0.613	0.157	0.530
Forêts aléatoires avec ADASYN	0.617	0.187	0.519
Forêts aléatoires avec BorderlineSMOTE	0.592	0.164	0.508

TABLE 19 – Performances des différents modèles initiaux de Forêts aléatoires (avec valeurs maximales en magenta)

	AUC-ROC	F_1-score	Balanced Accuracy
Gradient Boosting sans sur-échantillonnage	0.627	0.133	0.528
Gradient Boosting avec SMOTE	0.632	0.247	0.549
Gradient Boosting avec RandomOverSampler	0.636	0.159	0.516
Gradient Boosting avec ADASYN	0.628	0.293	0.582
Gradient Boosting avec BorderlineSMOTE	0.636	0.214	0.529

TABLE 20 – Performances des différents modèles initiaux de Gradient Boosting (avec valeurs maximales en magenta)

C.2 Courbes ROC des modèles initiaux

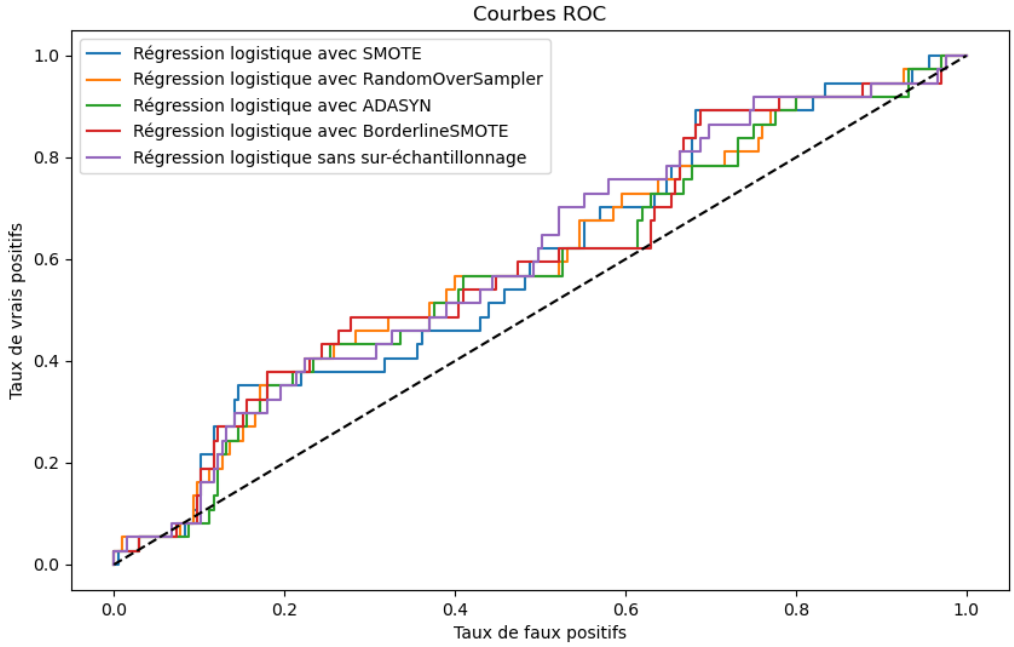


FIGURE 8 – Courbes ROC - Régression logistique (modèles initiaux)

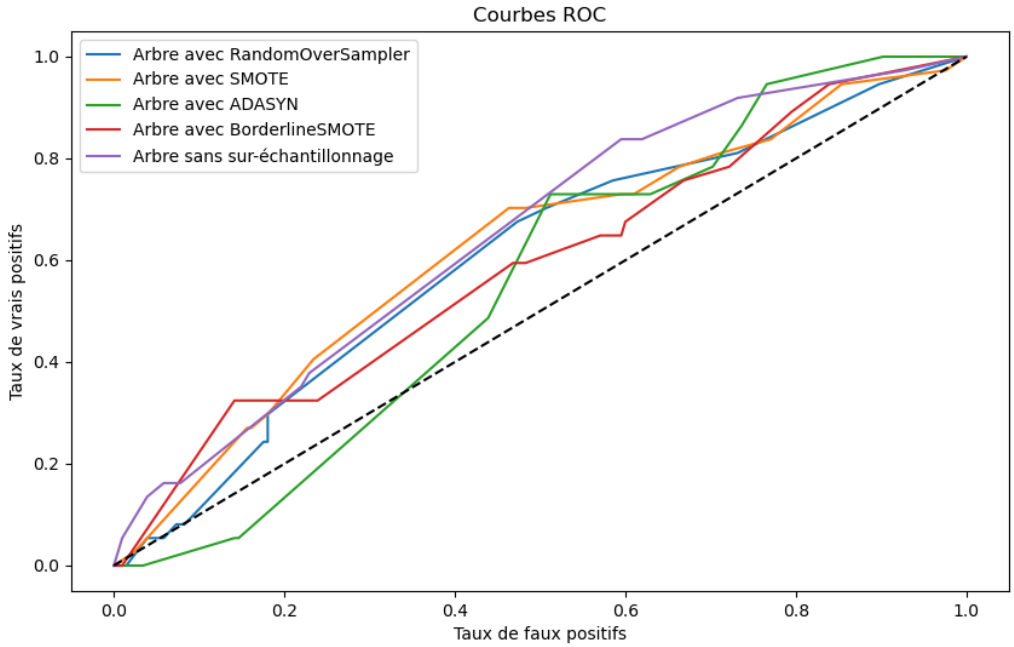


FIGURE 9 – Courbes ROC - Arbres de décision (modèles initiaux)

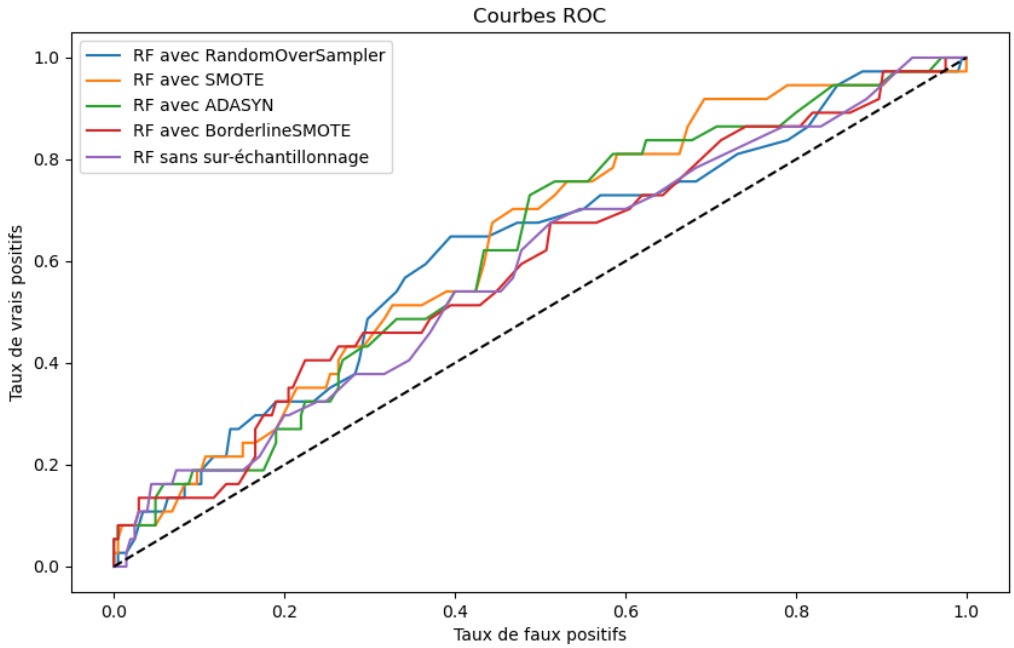


FIGURE 10 – Courbes ROC - Forêts aléatoires (modèles initiaux)

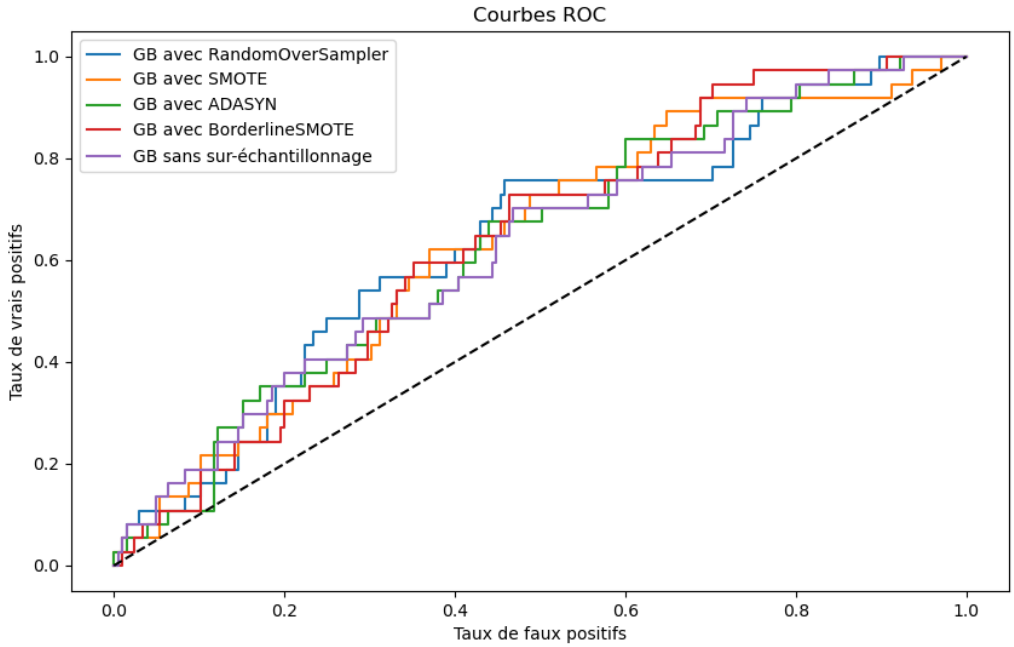


FIGURE 11 – Courbes ROC - Gradient boosting (modèles initiaux)

D Annexe 4 : Performances des modèles optimisés

D.1 Tableaux des performances des modèles optimisés

	AUC-ROC	F_1-score	Balanced Accuracy
Régression logistique sans sur-échantillonnage	0.589	0.050	0.509
Régression logistique avec SMOTE	0.584	0.297	0.583
Régression logistique avec RandomOverSampler	0.597	0.268	0.552
Régression logistique avec ADASYN	0.579	0.294	0.578
Régression logistique avec BorderlineSMOTE	0.597	0.273	0.555

TABLE 21 – Performances des différents modèles optimisés de régression logistique (avec valeurs maximales en magenta)

	AUC-ROC	F_1-score	Balanced Accuracy
Arbre de décision sans sur-échantillonnage	0.578	0.278	0.574
Arbre de décision avec SMOTE	0.614	0.265	0.563
Arbre de décision avec RandomOverSampler	0.532	0.232	0.509
Arbre de décision avec ADASYN	0.532	0.179	0.512
Arbre de décision avec BorderlineSMOTE	0.605	0.302	0.586

TABLE 22 – Performances des différents modèles optimisés d'arbres de décision (avec valeurs maximales en magenta)

	AUC-ROC	F_1-score	Balanced Accuracy
Forêts aléatoires sans sur-échantillonnage	0.584	0.050	0.509
Forêts aléatoires avec SMOTE	0.621	0.220	0.534
Forêts aléatoires avec RandomOverSampler	0.619	0.230	0.553
Forêts aléatoires avec ADASYN	0.629	0.313	0.598
Forêts aléatoires avec BorderlineSMOTE	0.635	0.279	0.564

TABLE 23 – Performances des différents modèles optimisés de forêts aléatoires (avec valeurs maximales en magenta)

	AUC-ROC	F_1-score	Balanced Accuracy
Gradient Boosting sans sur-échantillonnage	0.597	0.179	0.533
Gradient Boosting avec SMOTE	0.635	0.238	0.545
Gradient Boosting avec RandomOverSampler	0.620	0.263	0.556
Gradient Boosting avec ADASYN	0.622	0.206	0.536
Gradient Boosting avec BorderlineSMOTE	0.643	0.273	0.573

TABLE 24 – Performances des différents modèles optimisés de Gradient Boosting (avec valeurs maximales en magenta)

D.2 Courbes ROC des modèles optimisés

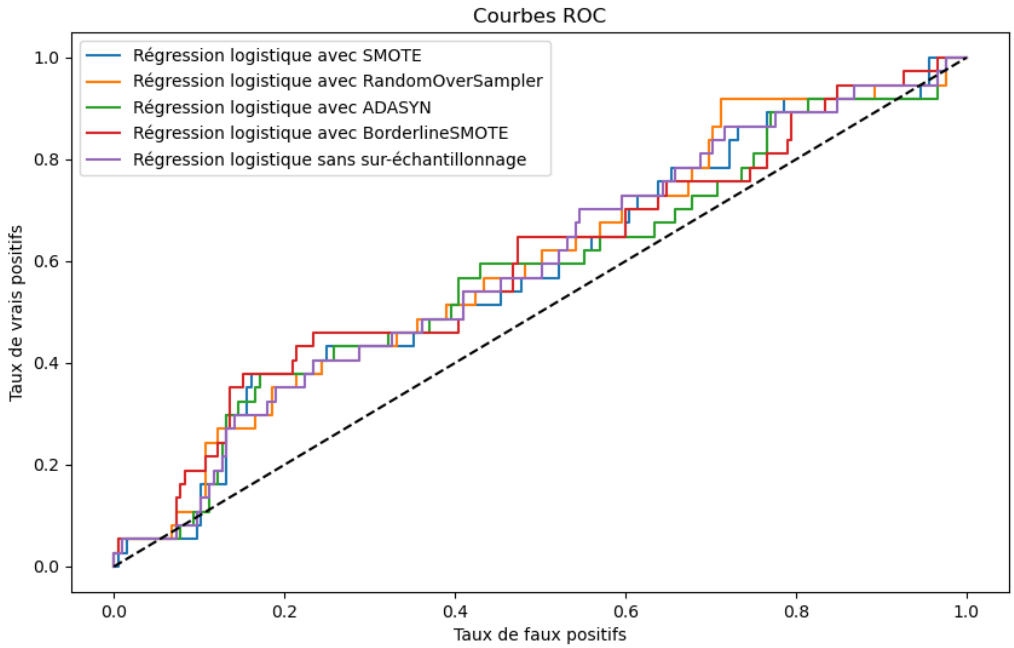


FIGURE 12 – Courbes ROC - Régression logistique (modèles optimisés)

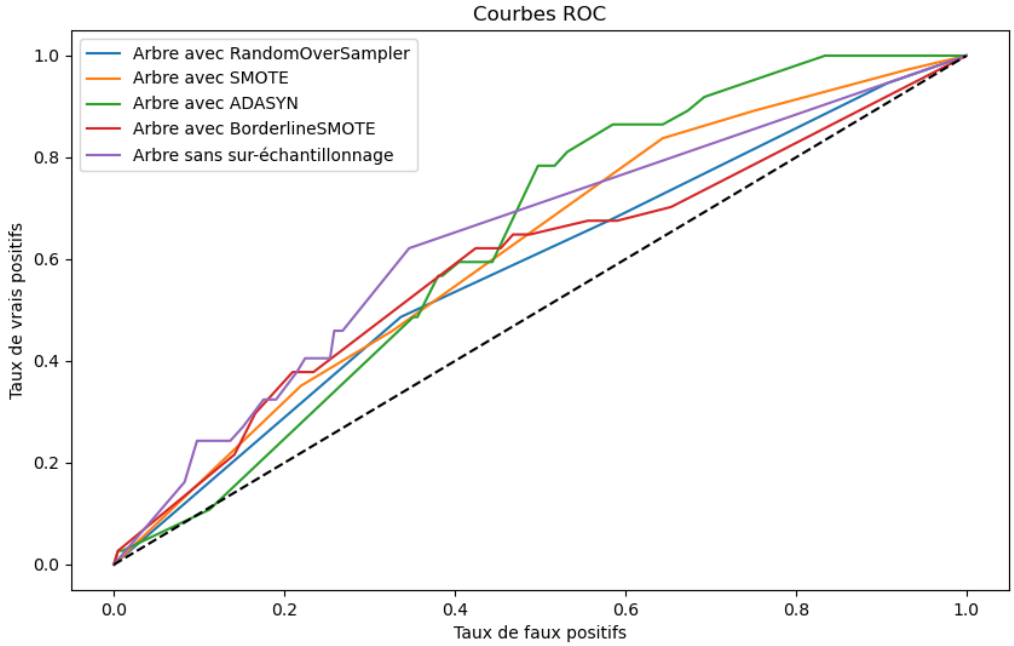


FIGURE 13 – Courbes ROC - Arbres de décision (modèles optimisés)

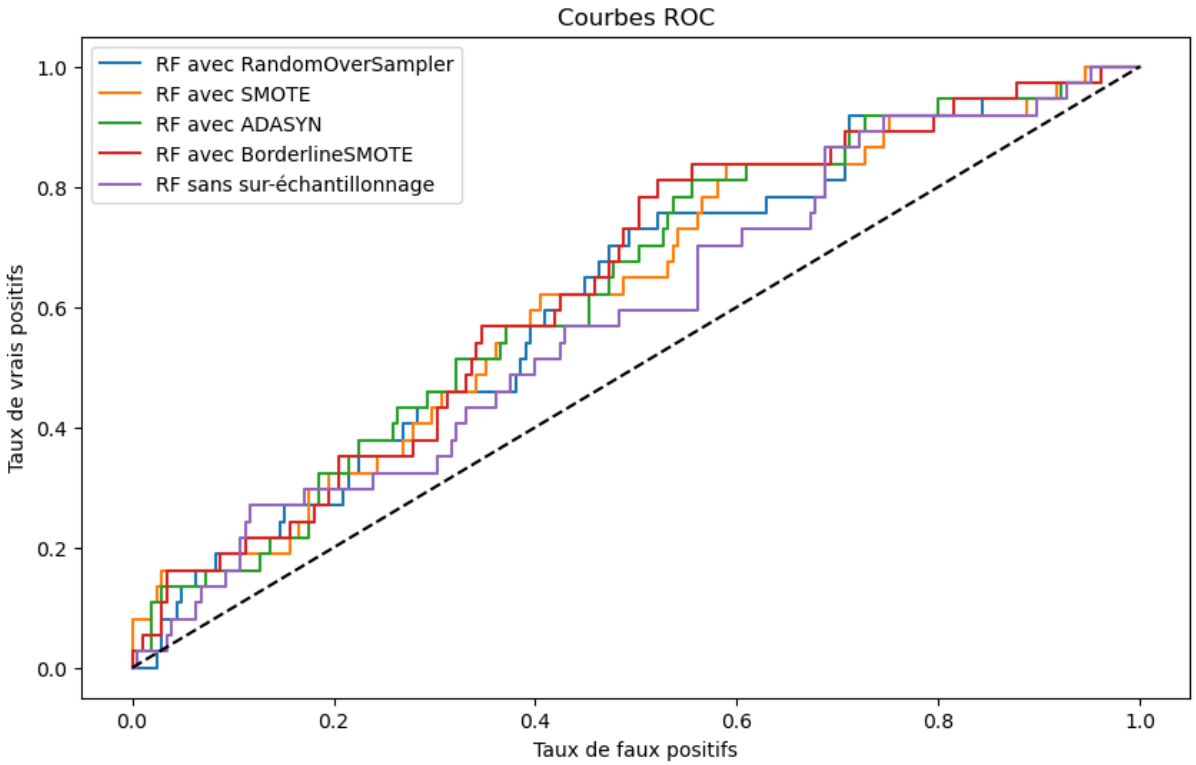


FIGURE 14 – Courbes ROC - Forêts aléatoires (modèles optimisés)

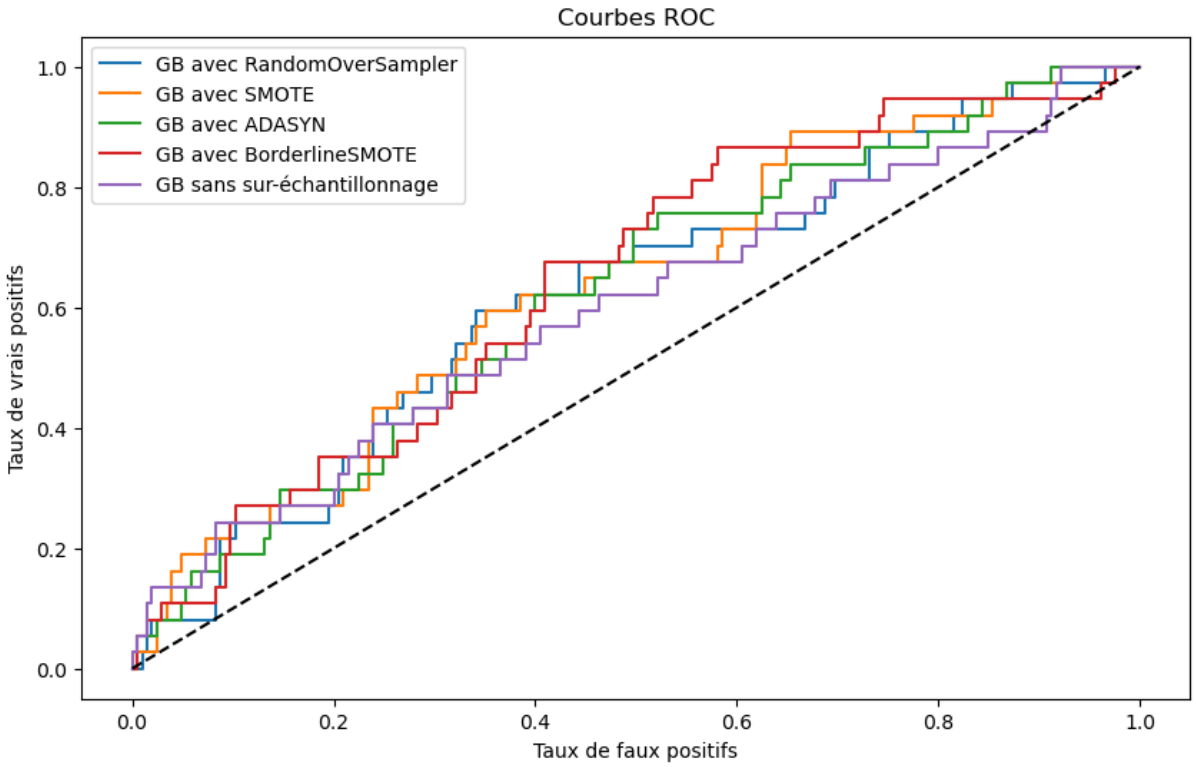


FIGURE 15 – Courbes ROC - Gradient boosting (modèles optimisés)

D.3 Courbes ROC des modèles de forêts aléatoires (2ème optimisation)

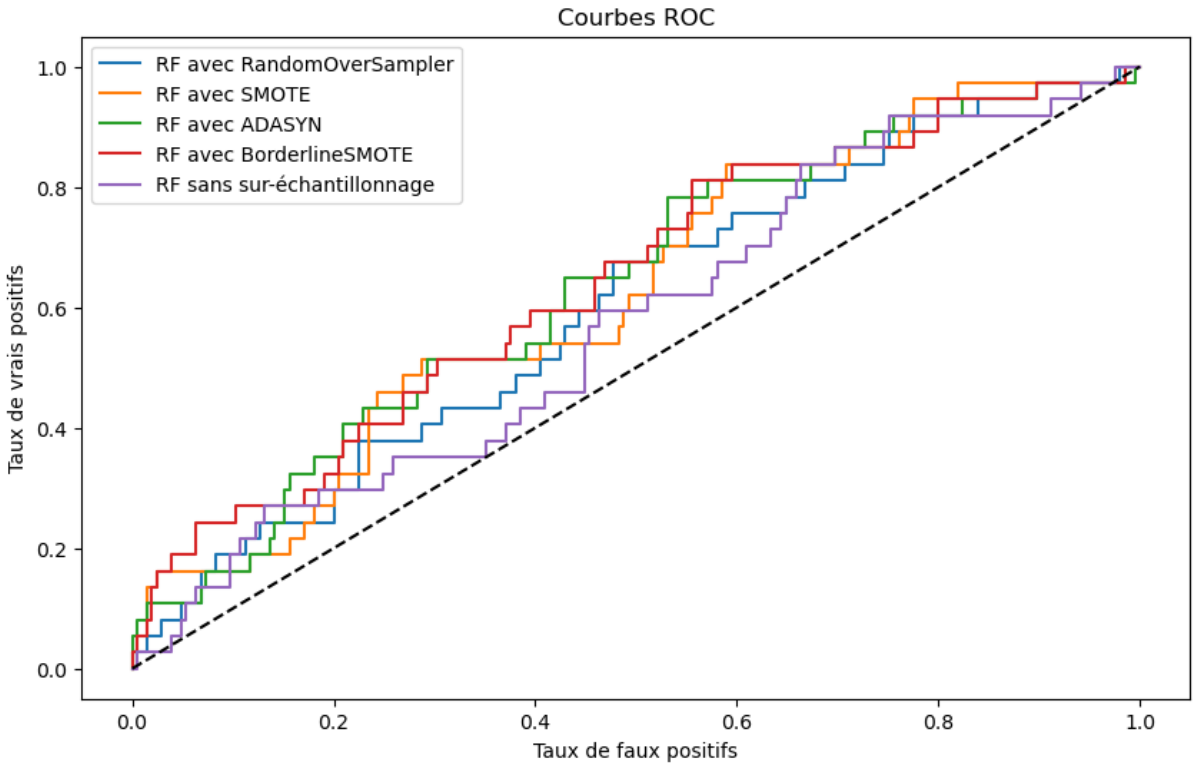


FIGURE 16 – Courbes ROC - Forêts aléatoires (2ème optimisation)

E Annexe 5 : Importance des variables

Variable	Importance
moyenne_conso_indus_hab	0.122 396
p_pop	0.092 284
friche	0.083 144
nb_actes_france_renov	0.082 437
gridens7	0.076 110
departement	0.071 145
total_entreprises	0.069 423
emissions_ges	0.046 926
part_inactifs	0.045 149
part_actifs	0.041 877
superf_choro	0.041 647
part_jeunes_sans_diplome	0.033 485
abstention_municipales	0.024 705
CSP_maire	0.024 642
dependance_eco	0.023 365
part_licencies_sportifs	0.022 776
com_variation_encours_dette_ha_pct	0.017 508
med_disp	0.014 222
moyenne_conso_agri_hab	0.013 412
part_residences_secondaires	0.012 964
moyenne_conso_tertiaire_hab	0.011 699
moyenne_conso_totale_hab	0.007 650
part_trajets_voiture	0.006 460
taux_creation_ent	0.006 229
moyenne_conso_residentiel_hab	0.006 061
climat_Estuaire	0.000 692
beneficiaire_prog	0.000 551
climat_Autre	0.000 551
gare_tgv	0.000 194
ecoquartiers	0.000 150
climat_Mer	0.000 148

TABLE 25 – Importance des variables du modèle final (forêts aléatoires)

Résumé

Ce rapport se concentre sur l'étude approfondie des facteurs susceptibles d'influencer la demande de financements (Etat : fonds vert, ADEME) des communes bretonnes dans le cadre de la transition écologique. Il met en lumière une modélisation statistique, abordant le défi posé des données déséquilibrées. Cet enjeu souligne l'importance du rééquilibrage des données pour améliorer la fiabilité des résultats. L'accent est mis sur l'application de modèles d'apprentissage supervisé avec différentes méthodes de sur-échantillonnage.
Les résultats de cette étude mettent en évidence les communes les moins susceptibles de solliciter des financements écologiques, identifiant ainsi des cibles prioritaires pour les politiques publiques. Ces conclusions offrent des perspectives stratégiques aux décideurs, leur fournissant un outil d'analyse pour renforcer l'engagement écologique à l'échelle locale.
En complément de l'analyse statistique, le rapport explore divers projets de valorisation des données, illustrant l'impact des visualisations de données et des statistiques dans le secteur public. Ces initiatives démontrent comment les visualisations peuvent faciliter la compréhension des politiques publiques par les agents de l'Etat et les décideurs.

Abstract

This report focuses on a study of the factors likely to influence the demand for financing (State : Green Fund, ADEME) from Brittany's communes in the context of the ecological transition. It highlights a statistical modeling, addressing the challenge posed by unbalanced data. This issue highlights the importance of re-balancing data to improve the reliability of results. The focus is on the application of supervised learning models with different oversampling methods.
The results of this study highlight the communes least likely to apply for ecological funding, thus identifying priority targets for public policy. These findings offer strategic insights to decision-makers, providing them with an analytical tool to reinforce ecological engagement at local level.
Complementing the statistical analysis, the report explores various data visualisation projects, illustrating the impact of data visualisations and statistics in the public sector. These initiatives demonstrate how visualisations can make it easier for public officials and decision-makers to understand public policy.