Machine Learning

Louzi Idriss[a], Mahdioui Mohamed Amine[a], and Théophile Schmutz[a]

[a]*Université Paris-Dauphine PSL.*

December 22, 2024

Abstract

The code is available in this repo.

Contents

1 Exploration des Données

Dans cette section, nous explorons, nettoyons et préparons les données à notre disposition. À noter que nous n'utiliserons pas la variable `duration`, qui correspond à la durée de l'appel, car elle n'est pas connue à l'avance. Ainsi, cette donnée doit être écartée si l'on souhaite obtenir un modèle prédictif réaliste.

1.1 Statistiques descriptives

Le jeu de données contient un total de 41,188 individus, répartis entre la classe positive et la classe négative de la manière suivante :

Variable cible `y` répartis entre la classe positive et la classe négative de la manière suivante :

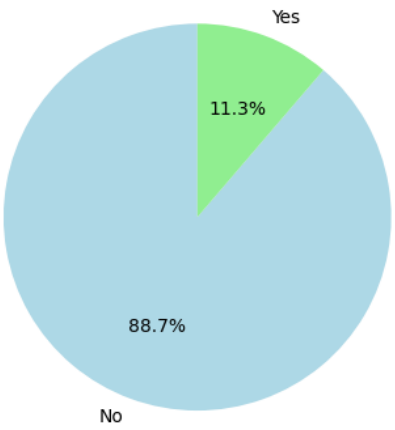

Figure 1.1: *Répartition de variable cible* ***y***

- Classe positive (True) : **11.3%** du total, soit 4,640 individus.
- Classe négative (False) : **88.7%** du total, soit 36,548 individus.

Ce déséquilibre marqué entre les classes peut influencer les résultats des modèles de machine learning, et des méthodes telles que l'oversampling, l'undersampling ou l'utilisation de poids de classe peuvent être envisagées pour y remédier.

Variables numériques Le jeu de données comprend plusieurs variables numériques, telles que `Age`, `Campaign`, `Pdays`, `Previous`, `Emp.var.rate`, `Cons.price.idx`, `Cons.conf.idx`, `Euribor3m`, et `Nr.employed`. Ces variables ont des distributions variées, avec certaines présentant des

asymétries ou des valeurs aberrantes. Les valeurs manquantes dans ces variables sont également codées comme *Inconnu*. Une vue d'ensemble de ces variables est présentée dans le tableau suivant :

Variable	Mean	Std	Min	25%	50%	75%	Max
Age	40.02	10.42	17.00	32.00	38.00	47.00	98.00
Campaign	2.57	2.77	1.00	1.00	2.00	3.00	56.00
Pdays	962.48	186.91	0.00	999.00	999.00	999.00	999.00
Previous	0.17	0.49	0.00	0.00	0.00	0.00	7.00
Emp.var.rate	0.08	1.57	-3.40	-1.80	1.10	1.40	1.40
Cons.price.idx	93.58	0.58	92.20	93.08	93.75	93.99	94.77
Cons.conf.idx	-40.50	4.63	-50.80	-42.70	-41.80	-36.40	-26.90
Euribor3m	3.62	1.73	0.63	1.34	4.86	4.96	5.05
Nr.employed	5167.04	72.25	4963.60	5099.10	5191.00	5228.10	5228.10

Table 1.1: *Résumé des statistiques descriptives des variables numériques*

Ce tableau présente des statistiques descriptives clés pour chaque variable numérique. Les moyennes et les médianes des variables telles que Age, Campaign, et Euribor3m montrent des tendances centrées, tandis que des valeurs aberrantes sont observées dans certaines colonnes comme Pdays, Emp.var.rate, et Cons.conf.idx.Pdays est fréquemment égale à 999, indiquant l'absence de démarchage. Les valeurs minimales et maximales de Age et Campaign indiquent une grande diversité dans les participants du jeu de données.

Variables catégorielles Le jeu de données contient plusieurs variables catégorielles, telles que Job, Marital, Education, Default, Housing, Loan, Contact, Month, Day_of_week, Poutcome, et y. Ces variables comportent des modalités variées, avec certaines catégories présentant des fréquences faibles. Un résumé des modalités et de leurs fréquences est présenté dans le tableau suivant :

	unique	top	freq
job	12	admin.	10422
marital	4	married	24928
education	8	university.degree	12168
default	3	no	32588
housing	3	yes	21576
loan	3	no	33950
contact	2	cellular	26144
month	10	may	13769
day_of_week	5	thu	8623
poutcome	3	nonexistent	35563
y	2	no	36548

Table 1.2: *Résumé des modalités des variables catégorielles*

Ce tableau présente les modalités des variables catégorielles ainsi que les fréquences des modalités les plus fréquentes. Par exemple, pour la variable `Job`, la catégorie la plus fréquente est `admin.`, apparaissant 10,422 fois, tandis que pour la variable `Marital`, la modalité `married` est la plus fréquente avec 24,928 occurrences. Les variables telles que `Default`, `Housing`, et `Loan` présentent des catégories majoritaires qui sont souvent `no` ou `yes`, et certaines variables ont des catégories moins fréquentes, comme `Poutcome` où la modalité `nonexistent` est largement prédominante.

Valeurs manquantes Une proportion importante de données manquantes est présente dans les variables catégorielles, comme vue dans la figure ci-dessous:

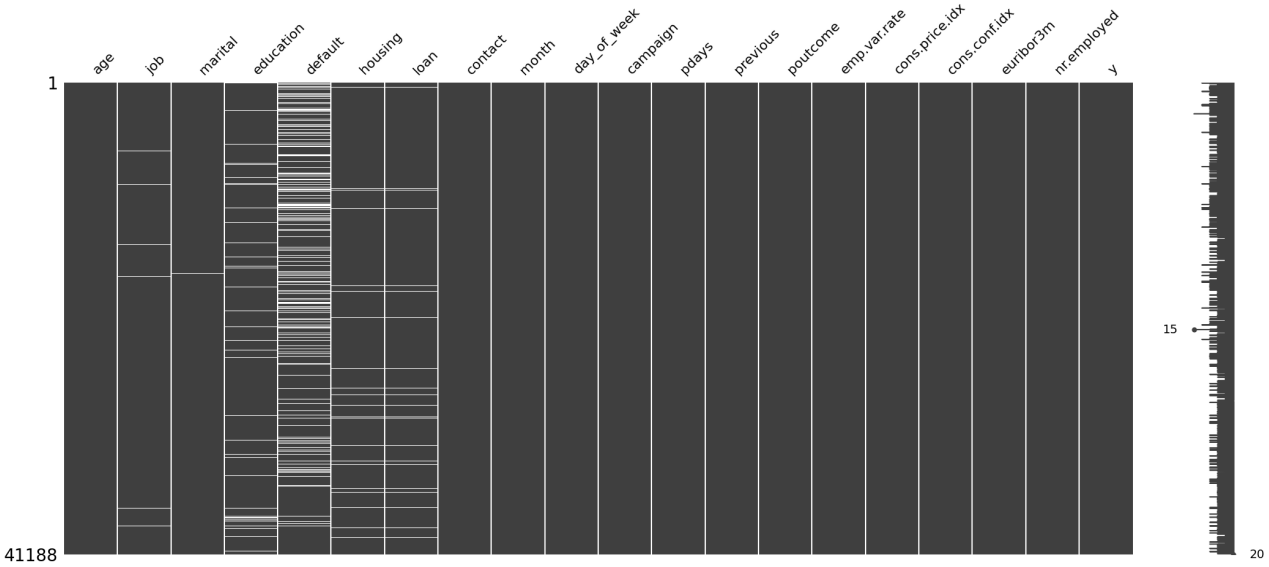

Figure 1.2: *Matrice des données manquantes.*

Ces valeurs manquantes sont représentées par *unknown* et nécessitent une méthode d'imputation ou d'élimination avant toute analyse. Un résumé des valeurs manquantes pour chaque variable est présenté dans la table 1.3.

Variable	`Age`	`Job`	`Marital`	`Education`	`Default`	`Housing`	`Loan`	`Other Variables`
Valeurs manquantes	0	330	80	1731	8597	990	990	0

Table 1.3: *Résumé des valeurs manquantes par variable*

Ce tableau montre que certaines variables présentent un nombre non négligeable de valeurs manquantes, notamment `Job` (330 valeurs manquantes), `Education` (1731 valeurs manquantes), et `Default` (8597 valeurs manquantes). D'autres variables, comme `Age`, `Contact`, `Month`, et `Day_of_week`, ne présentent aucune valeur manquante. Pour les variables avec des valeurs manquantes importantes, une méthode d'imputation ou d'élimination devra être considérée.

1.2 Analyse univariée des variables

Dans cette section, nous explorons plus en détail certaines des variables numériques et catégorielles afin de mieux comprendre la structure des données et de dégager des observations importantes avant de procéder à l'analyse de corrélation.

1.2.1 Variables numériques

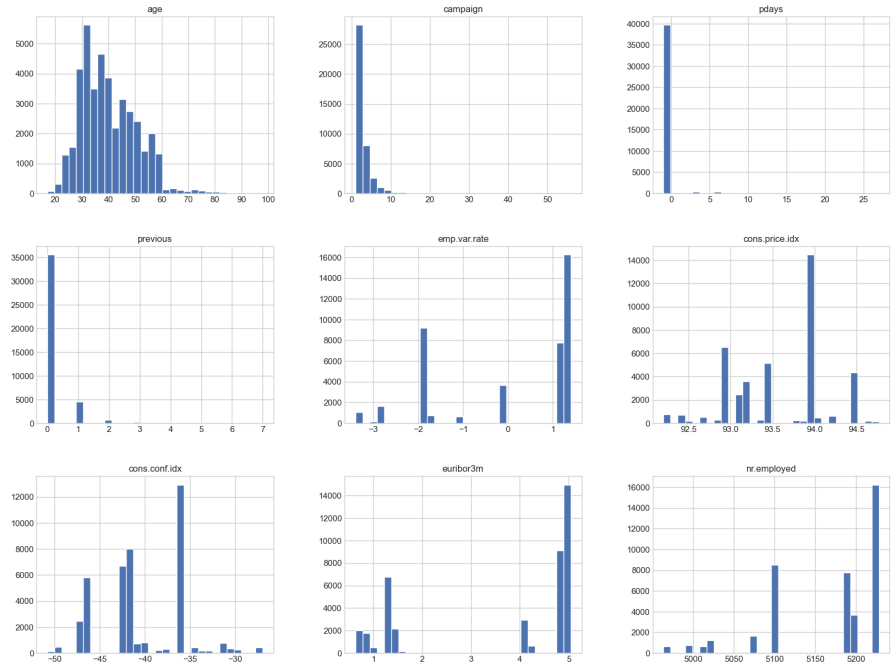

Figure 1.3: *Distribution des variable numériques*

- **Âge** : La distribution de l'âge est légèrement asymétrique à droite, la majorité des clients étant âgés entre 20 et 60 ans, avec un pic autour de 30-40 ans.
- **Campagne** : La plupart des clients ont été contactés une seule fois lors de la campagne en cours, et ce nombre diminue de manière significative au fur et à mesure que le nombre de contacts augmente.
- **Pdays** : La majorité des clients n'ont pas été contactés lors de campagnes précédentes (valeur -1 indiquant l'absence de contact antérieur). Une très petite portion montre des recontacts récents.
- **Previous** : La plupart des clients n'ont pas été contactés lors de campagnes précédentes (valeur 0), et seuls quelques-uns ont été contactés plus d'une fois.

Les variables comme **campagne**, **pdays** et **previous** suggèrent que la banque cible fréquemment de nouveaux clients et suit rarement les contacts antérieurs. Les variables fortement asymétriques, telles que **pdays** et **campagne**, peuvent nécessiter une transformation pour les modélisations afin de réduire les biais dans les modèles.

1.2.2 Données catégorielles

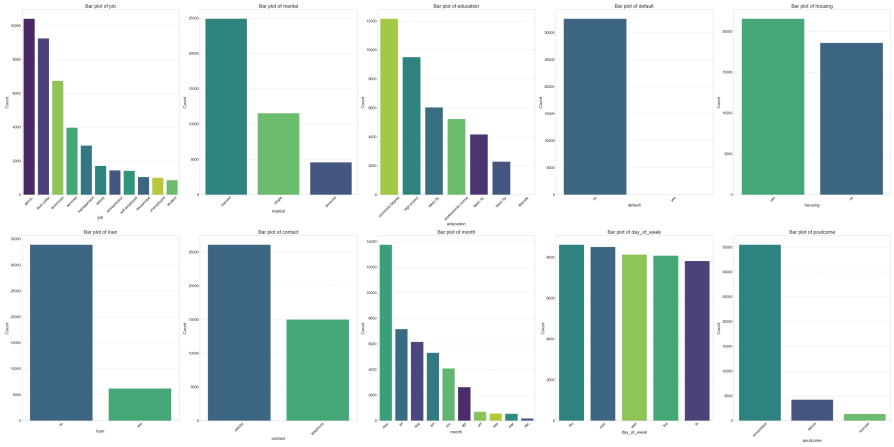

Figure 1.4: *Distribution des Données catégorielles*

- **Job, Marital, et Education** : La majorité des clients sont employés dans des postes administratifs ou ouvriers, avec une proportion significative dans des rôles de techniciens. Une grande majorité est mariée, suivie des célibataires. En termes d'éducation, les diplômés universitaires dominent, bien que les catégories de niveau secondaire et de formation de base soient également présentes.
- **Default, Housing, et Loan** : Très peu de clients ont eu un défaut de paiement, mais les prêts immobiliers sont courants. Les prêts personnels sont moins fréquents. Ces tendances suggèrent la stabilité financière de la majorité de la clientèle, ce qui pourrait influencer la probabilité de souscrire à de nouveaux produits financiers.
- **Contact et Month** : Le téléphone portable est le mode de contact dominant, ce qui met en évidence la tendance à une approche plus directe et personnalisée. L'activité des campagnes atteint son pic en mai, suivie d'une diminution progressive dans les mois suivants, ce qui suggère une poussée stratégique durant une période spécifique, peut-être liée à des cycles budgétaires ou à des facteurs saisonniers.
- **Jour de la semaine** : Les appels sont distribués de manière homogène au cours des jours de la semaine, sans préférence marquée pour un jour particulier.
- **Poutcome** montre que la majorité des clients ont un résultat "inexistant" pour les campagnes précédentes, ce qui suggère que la banque cherche principalement de nouveaux contacts.

Les graphiques montrent la répartition des clients selon leur statut professionnel, marital, éducatif, et les types de contact utilisés par la banque. Ces informations sont cruciales pour mieux comprendre le profil des clients et les priorités stratégiques de la banque.

1.3 Analyse Bivariee : Numerique vs Categorielle

Dans cette sous-section, nous analysons les relations entre les variables numériques et la variable cible `y` (abonnement : `yes` ou `no`) à l'aide de boxplots. Ces visualisations mettent en évidence les différences entre abonnés et non-abonnés, clarifiant l'impact potentiel des variables sur la probabilité d'abonnement.

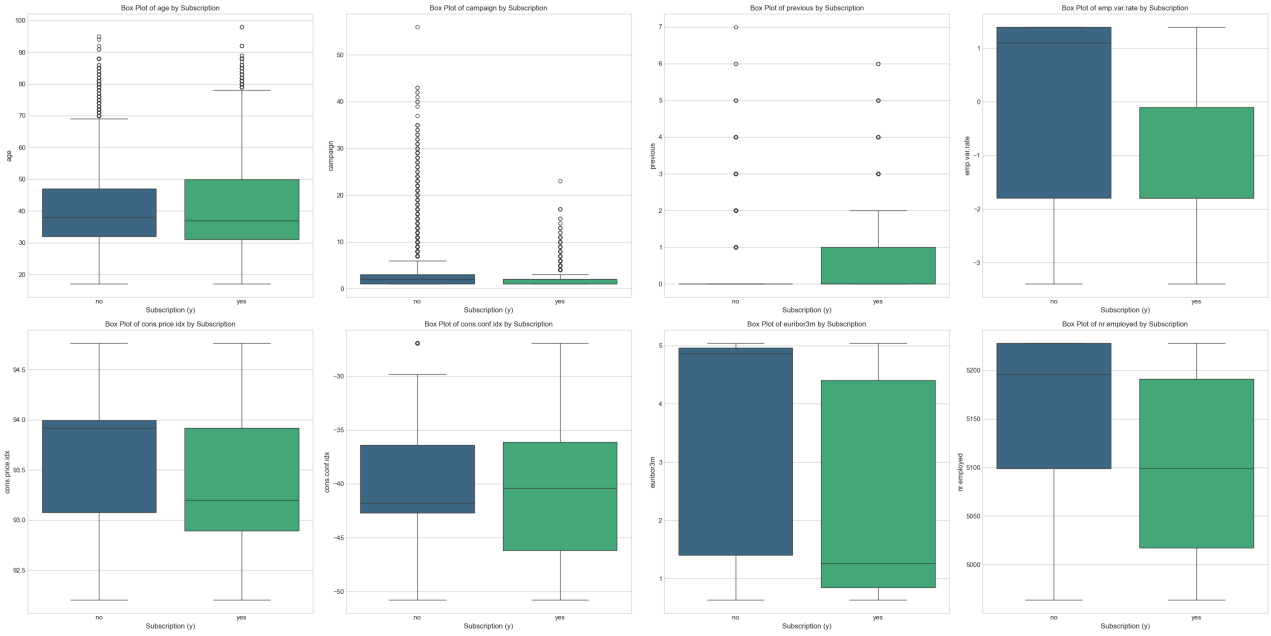

Figure 1.5: *Boxplot des variables par* `y`

- **Age** : Les abonnes presentent une plage plus large, ce qui pourrait indiquer une probabilite legerement plus elevee pour les individus plus ages de s'abonner. Cependant, les medianes etant proches, cet effet semble mineur.
- **Campagne** : Les non-abonnes ont, en moyenne, ete contactes plus souvent. Cela suggere que limiter le nombre de contacts pourrait etre une strategie plus efficace pour convertir les clients.
- **Interactions Precedentes (`previous`)** : Les abonnes ont montre une frequence d'interaction anterieure plus elevee, soulignant l'importance de maintenir des relations regulieres avec les clients.
- **Taux de Variation de l'Emploi (`emp.var.rate`)** : Les non-abonnes affichent un taux moyen de variation de l'emploi plus eleve, ce qui laisse entendre que des conditions de marche defavorables pourraient decourager l'abonnement.
- **Indice des Prix a la Consommation (`cons.price.idx`)** : Bien que les non-abonnes aient des moyennes et medianes legerement plus elevees, l'impact de cette variable semble negligeable.
- **Indice de Confiance des Consommateurs (`cons.conf.idx`)** : Les abonnes ont une

confiance des consommateurs legerement meilleure, mais aussi plus variable, suggerant un role mineur de l'optimisme dans les decisions d'abonnement.

- **Taux Euribor sur 3 Mois (`euribor3m`)** : Les non-abonnes ont des taux Euribor moyens plus eleves, ce qui indique que des taux d'interet plus faibles pourraient favoriser l'abonnement.
- **Nombre d'Employes (`nr.employed`)** : Les non-abonnes ont un nombre moyen d'employes plus eleve, ce qui pourrait suggerer une correlation entre des niveaux d'emploi plus faibles et une probabilite accrue d'abonnement.
- **Indicateurs Economiques** : Les abonnes sont associes a des taux de variation de l'emploi plus faibles, a des taux Euribor plus bas et a un nombre d'employes moindre, indiquant que des conditions economiques plus faibles pourraient encourager l'abonnement.
- **Interactions Clients** : Les interactions anterieures (`previous`) et les recents contacts (`pdays`) sont des facteurs significatifs pour predire l'abonnement.
- **Strategie de Campagne** : Reduire le nombre d'appels de campagne semble etre une approche plus efficace pour convertir les clients en abonnes.

1.4 Analyse des corrélations

Analyser les corrélations entre variables, catégoriques et numériques, guide la sélection des caractéristiques pertinentes pour notre modèle.

1.4.1 Corrélation entre les variable

Pour mesurer la relation entre ces types de variables, nous avons utilisé le coefficient de corrélation bisérial ponctuel (*Point-Biserial Correlation*). Cette méthode est particulièrement adaptée pour analyser la corrélation entre une variable numérique et une variable catégorique binaire, en quantifiant la force et la direction de leur association. Cette analyse nous a permis d'identifier les interactions significatives et d'orienter notre sélection de variables pour le modèle final.

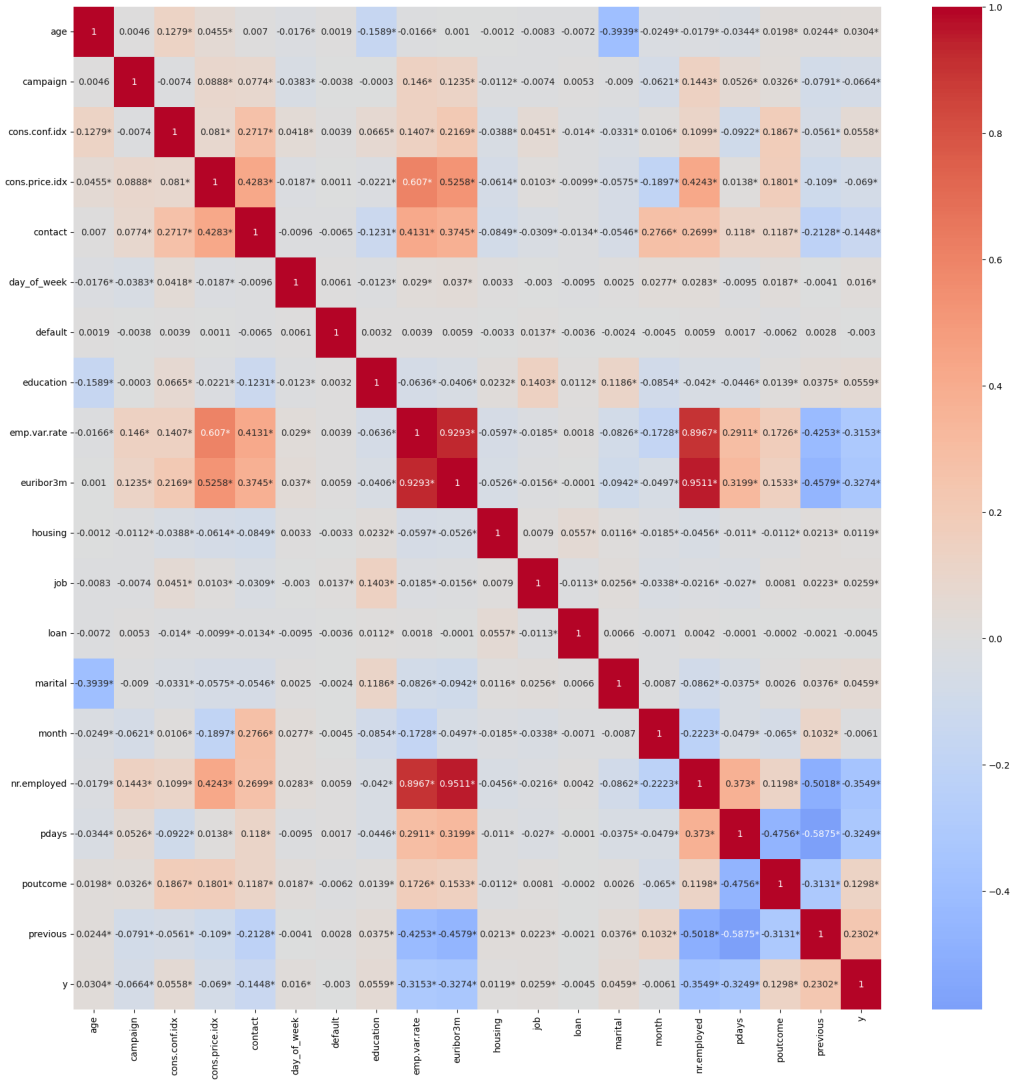

Figure 1.6: *Carte thermique des corrélations entre les variables.*

Variables client

- `age` : présente des corrélations minimales avec la plupart des variables, la plus élevée étant avec `default` à **0,1650**.
- `education` : montre une faible corrélation avec d'autres variables, par exemple avec `y` à **0,0578**.

Variables financières

- `default` : inclut les corrélations mentionnées indirectement liées aux variables démographiques comme `age`.
- `housing` et `loan` : présentent une corrélation négligeable avec la cible (`y`).

Variables liées au démarchage

- `campaign` et `y` : faible corrélation négative (**-0,0664**), indiquant qu'un nombre plus élevé de contacts n'améliore pas nécessairement le succès.
- `pdays` et `previous` : corrélation positive modérée (**0,5065**), montrant que des contacts récents dans les campagnes précédentes sont liés à un nombre plus élevé de contacts passés.
- `pdays` et `y` : légère corrélation positive (**0,2790**), indiquant qu'une période plus courte depuis les derniers contacts pourrait influencer positivement les souscriptions.
- `previous` et `y` : faible corrélation positive (**0,2302**), suggérant qu'un nombre accru de contacts passés est modestement associé au succès.
- `poutcome` et `y` : corrélation modérée (**0,1298**), montrant que le succès des campagnes passées est lié à celui des campagnes actuelles.
- `month`, `day_of_week` : non mentionnées explicitement mais incluses comme variables liées au démarchage.

Indicateurs marketing et économiques

- `contact` et `euribor3m` : corrélation modérée (**0,3998**), reliant les types de contact aux taux Euribor.
- `contact` et `cons.price.idx` : corrélation modérée (**0,5915**), montrant une influence des stratégies marketing par l'indice des prix à la consommation.
- `cons.price.idx` et `emp.var.rate` : corrélation forte (**0,7753**), alignant l'indice des prix à la consommation avec la variation de l'emploi.

Variables économiques

- `emp.var.rate` et `euribor3m` : tres forte correlation (0,9722), refletant les conditions economiques globales.
- `emp.var.rate` et `nr.employed` : forte correlation (0,9070), indiquant que des taux d'emploi plus eleves s'accompagnent d'une augmentation des personnes employees.
- `euribor3m` et `nr.employed` : correlation significative (0,9452), suggerant un alignement 'etroit entre les taux d'interet et les niveaux d'emploi.

2 Feature Engineering

Dans cette section, nous présentons les nouvelles caractéristiques créées grâce à la transformation des données et discutons de l'impact de ces changements sur la préparation des données avant l'entraînement du modèle.

2.1 Réduction de la Dimensionnalité : Analyse en Composantes Principales (ACP)

Pour réduire la redondance causée par la forte interdépendance des variables économiques, une Analyse en Composantes Principales (ACP) a été appliquée.

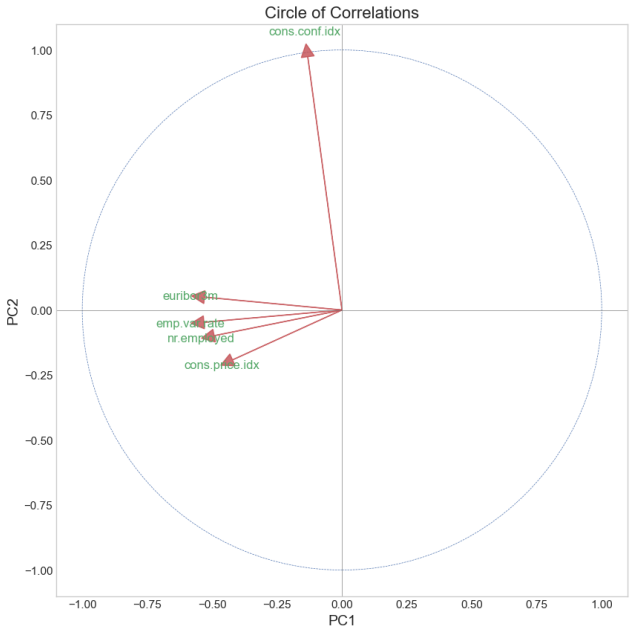

Figure 2.1: *Biplot des Composantes Principales (PC1 et PC2)*

- **Composante Principale 1 (PC1)** :
  - Represente la *stabilite ou croissance macroeconomique*.
  - Des valeurs elevees de PC1 peuvent indiquer des periodes de contraction economique (par exemple, faible emploi, baisse des taux d'interet).
  - Des valeurs faibles de PC1 refletent une economie stable ou en croissance.
- **Composante Principale 2 (PC2)** :
  - Represente le *sentiment des consommateurs*.
  - Des valeurs elevees de PC2 sont associees a un optimisme dans la confiance des consommateurs.
  - Des valeurs faibles de PC2 peuvent refleter un pessimisme ou une incertitude chez les consommateurs.

En conséquence, nous pouvons réduire la complexité du modèle en nous concentrant sur les facteurs principaux qui décrivent les conditions économiques. Les variables secondaires peuvent être supprimées en se basant sur les informations capturées par la première composante principale (PC1).

2.2 Création des nouvelles caractéristiques

Le processus de feature engineering inclut plusieurs étapes de transformation qui ajoutent de nouvelles colonnes pertinentes à l'ensemble de données. Voici les principales transformations effectuées sur les caractéristiques brutes :

1. **Colonne** `prev_contact` : Cette colonne est créée à partir de la variable `pdays`. Elle catégorise l'intervalle de temps écoulé depuis le dernier contact en quatre groupes : "Never" (jamais contacté), "[0-7]" (contacté récemment), "[7-15]" et "15<" (contacté il y a plus de 15 jours). Cela permet de mieux comprendre la relation entre la durée depuis le dernier contact et la probabilité de succès de la campagne.
2. **Colonne** `age_cat` : L'âge des individus est catégorisé en cinq tranches d'âge distinctes. Cette transformation permet de capturer les tendances comportementales en fonction des groupes d'âge. Ces tranches d'âge sont : "[0-23[", "[23-30[", "[30-40[", "[40-60[" et "60<".
3. **Colonne** `campaign_efficiency` : Cette nouvelle caractéristique mesure l'efficacité de la campagne marketing en fonction du ratio entre le nombre de contacts précédents et le nombre total de contacts dans la campagne, mais uniquement si le résultat de la campagne (`poutcome`) est un succès. Cela peut offrir une vision plus précise de l'impact de la campagne en fonction des antécédents.

2.3 Résumé du Pipeline de Préparation des Données

Le pipeline d'ingénierie des caractéristiques est conçu pour prétraiter les données brutes et générer de nouvelles caractéristiques pour l'entraînement du modèle. Voici les étapes de transformation, décrites pas à pas :

- Supprimer la colonne `duration`.
- Remplacer 999 dans la colonne `pdays` par -1.
- Convertir les colonnes de type `object` en type `catégoriel`.
- Créer une nouvelle colonne `prev_contact` basée sur `pdays`, en utilisant des intervalles (*bins*) et des étiquettes (*labels*) spécifiés.
- Créer une nouvelle colonne `age_cat` basée sur `age`, en utilisant des intervalles (*bins*) et des étiquettes (*labels*) spécifiés.
- Créer une nouvelle colonne `campaign_efficiency` basée sur les colonnes `previous` et `campaign`.
- Ajouter la première composante principale des facteurs économiques sous forme de colonne `EconStabSentPCA`.
- Supprimer les colonnes spécifiées et retourner le DataFrame transformé.

Une représentation graphique du pipeline d'ingénierie des caractéristiques peut aider à visualiser les transformations. Voici une recommandation pour la structure d'un tel

graphique :

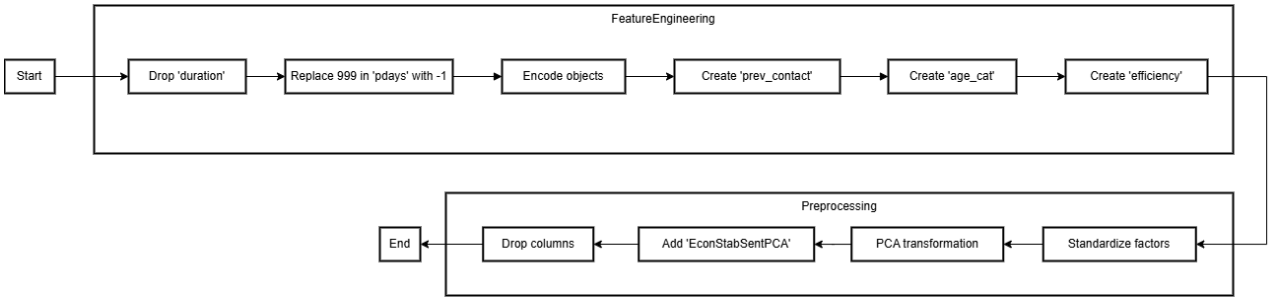

Figure 2.2: *Diagramme du pipeline d'ingénierie des caractéristiques*

3 Modèles A: Gradient Boosting et XGBoost

Dans cette section, nous introduisons les **modèles A**, qui sert de base à la dérivation de six variantes:

1. Deux variantes de *Gradient Boosting Classifier* de la bibliothèque python *sklearn*:
   (a) Une où la *precision* est maximisée,
   (b) Une où le *F score* est maximisée
2. Quatre variantes de *XGBoost* de la bibliothèque python *xgboost*:
   (a) Une où la *precision* est maximisée sans poids de classe,
   (b) Une où la *precision* est maximisée avec poids de classe,
   (c) Une où le *F score* est maximisée sans poids de classe,
   (d) Une où le *F score* est maximisée avec poids de classe,

3.1 Sampling

Le dataset est d'abord divisé en deux sous-ensembles distincts : 80% pour l'entraînement et 20% pour le test. Un échantillonnage stratifié est appliqué sur la variable cible afin de préserver la distribution des classes dans ces sous-ensembles. Ensuite, au sein des 80% dédiés à l'entraînement/validation, une nouvelle division est effectuée pour réaliser la validation croisée (80%) et l'ajustement du seuil de décision[1] (20%). Cette seconde étape garantit que le test final reste indépendant et n'est pas utilisé lors des phases de validation ou d'optimisation. Pour pallier le déséquilibre des classes dans le dataset, un poids pourra être attribué aux instances positives pour certains modèles (deux modèles *XGBoost*):

$$\texttt{scale\_pos\_weight} = \frac{\texttt{\# instances négatives}}{\texttt{\# instances positives}}.$$

3.2 Hyperparamétrage

Pour chacun des six modèles, les hyperparamètres suivants sont ajustés : la profondeur maximale des arbres (`max_depth`), le nombre d'arbres (`n_estimators`), le taux d'apprentissage (`learning_rate`) et le ratio de sous-échantillonnage (`subsample`). Les valeurs explorées pour chaque hyperparamètre sont présentées dans le tableau ci-dessous :

Hyperparamètre	**Valeurs explorées**
`max_depth`	$[4, 6, 8, 10]$
`n_estimators`	$[25, 50, 75, 100]$
`learning_rate`	$[0.05, 0.1, 1]$
`subsample`	$[0.3, 0.5, 0.7]$

Table 3.1: *Hyperparamètres explorés lors de la validation croisée pour les six modèles.*

[1] *threshold tuning*

3.3 Cross-Validation

Une validation croisée stratifiée à 5 plis est utilisée pour évaluer les performances des modèles. L'option `shuffle=True` est activée afin de mélanger les échantillons de chaque classe avant leur répartition dans les plis, garantissant ainsi une distribution équilibrée des classes.

Les modèles sont évalués sur la grille d'hyperparamètres décrite dans table 3.1. L'optimisation des hyperparamètres est réalisée en deux étapes : une première optimisation ciblant la `precision`, et une seconde ciblant le `F score`. Ainsi, chaque modèle sera disponible en deux versions : une version maximisant la `precision`, et une autre maximisant le `F score`. Cette approche permet de répondre aux besoins spécifiques de l'entreprise, en fonction de sa stratégie de démarchage, qu'elle soit plus ou moins agressive.

Une fois l'hyperparamétrage terminé, les modèles sont entraînés sur le *train*. Ensuite, les *thresholds* sont ajustés pour optimiser une métrique personnalisée, appelée `business_metric`.

Il est important de noter que l'ajustement du *threshold* doit être réalisé avec prudence. Par exemple, maximiser uniquement la précision peut conduire à des modèles très conservateurs, offrant une précision élevée (proche de 100%), mais au détriment d'un rappel très faible.

Dans le code, nous avons également prévu la possibilité d'optimiser le seuil en maximisant la métrique initiale du modèle, afin de visualiser l'impact de cette approche sur la prudence des prédictions. Cette méthode a été écartée, car les modèles se révélaient alors trop conservateurs. Nous avons également tenté d'optimiser les seuils en maximisant l'autre métrique (Si le modèle maximisait initialement la précision, alors le seuil était ajusté pour maximiser le F-score, afin de trouver un certain équilibre prudence-agressivité), mais les résultats n'étaient pas satisfaisants.

3.3.1 Business metric

En effet, la métrique dite "business" traduit la performance d'un modèle de classification en termes économiques, en tenant compte des bénéfices et des coûts spécifiques des démarchages. L'objectif principal de cette métrique est d'aligner l'évaluation du modèle sur les priorités et la stratégie de la banque.

La métrique calcule la valeur nette[2] des démarchages. Elle est calculée en additionnant les bénéfices des vrais positifs (*benefit*) et en soustrayant les coûts associés aux faux positifs (*cost*) et aux faux négatifs (*missed oportunity*). Cela permet d'évaluer directement l'impact économique global du modèle en tenant compte à la fois des gains et des pertes liés aux prédictions.

$$\texttt{net\_value} = (\#\texttt{tp} \times \texttt{benefit}) - (\#\texttt{fp} \times \texttt{cost}) - (\#\texttt{fn} \times \texttt{missed\_opportunity})$$

[2] *net value*

En ajustant les valeurs de *cost*, *missed opportunity*, et *benefit*, il est possible de moduler les priorités du modèle. Par exemple, augmenter *cost* par rapport à *benefit* (exemple: *cost* = 1.5 *benefit*) pénalisera davantage les faux positifs, favorisant ainsi la précision en donnant une importance 50% fois supérieurs aux faux positifs, ce qui réduira le nombre de clients non pertinents contactés par le modèle. De même, en augmentant *missed opportunity* par rapport à *benefit*, le modèle tendra à privilégier le recall. L'équilibre entre ces trois valeurs doit être défini en collaboration avec la banque, afin qu'il reflète au mieux sa stratégie marketing et ses priorités commerciales.

$$\texttt{benefit} = 1, \quad \texttt{cost} = 1.5, \quad \texttt{missed\_opportunity} = 0.5$$

Nous n'avons pas souhaité utiliser cette métrique personnalisée lors de l'hyperparamétrage, car nous ne disposions pas d'estimations des coûts et bénéfices de la banque. De plus, les hyperparamètres peuvent être sensibles à ces estimations, ce qui aurait pu introduire de l'incertitude dans le processus d'optimisation. Dans le code nous avons fixé,

3.3.2 Processus

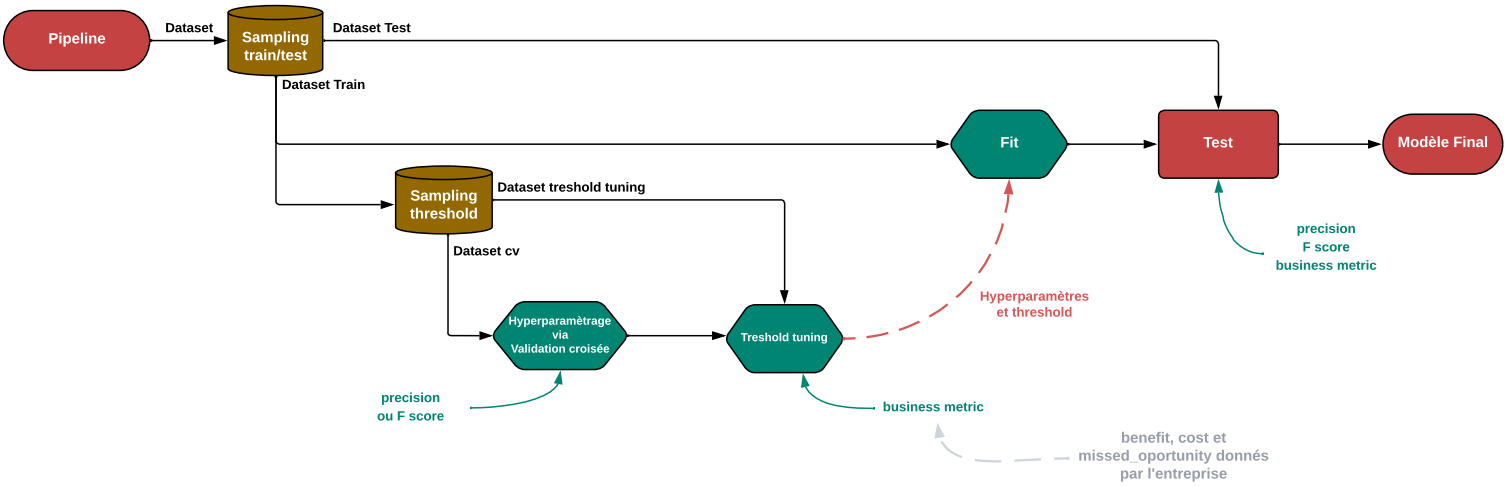

Figure 3.1: *Résumés des six modèles A (GBoost, XGBoost, avec et sans poids, optimisé en validation croisée avec la precision ou la F score). La version verticale de ce schéma (un peu plus visible) est disponible en annexe fig. A.1.*

3.4 Résultats

Type du modèle	Modalités	Metric			
		Precision	Recall	F1	Business
GBoost	precision	0.79	0.15	0.25	-316
	f1	0.61	0.30	0.40	-311
XGBoost	precision no weights	0.77	0.13	0.22	-336
	precision weights	0.45	0.59	0.51	-638
	f1 no weights	0.61	0.30	0.41	-308
	f1 weights	0.44	0.61	0.51	-692

Table 3.2: *Résultats des tests des modèles A avec un threshold par défaut de 0.5. Les modalités du modèle correspondent à la métrique utilisée lors de l'hyperparamétrage (precision ou F score), ainsi qu'à l'utilisation de poids pour compenser le déséquilibre des classes.*

Type du modèle	Modalités	Optimal Threshold	Metric			
			Precision	Recall	F1	Business
GBoost	precision	0.39	0.70	0.22	0.33	-292
	f1	0.51	0.62	0.29	0.40	-311
XGBoost	precision no weights	0.39	0.71	0.22	0.34	-284
	precision weights	0.73	0.62	0.27	0.38	-314
	f1 no weights	0.71	0.73	0.16	0.26	-326
	f1 weights	0.78	0.68	0.20	0.31	-318

Table 3.3: *Résultats des tests des modèles A avec un threshold optimisé en maximisant la business_metric. Les modalités du modèle correspondent à la métrique utilisée lors de l'hyperparamétrage (precision ou F score), ainsi qu'à l'utilisation de poids pour compenser le déséquilibre des classes.*

Les tableaux 3.2 et 3.3 comparent les performances des modèles GBoost et XGBoost respectivement aux seuils par défaut (0.5) et aux seuils optimaux maximisant la métrique métier. À noter que ce n'est pas grave si la métrique de business est négative dans les tableaux table 3.3 et table 3.2, car elle ne signifie pas grand-chose concrètement dans notre cas puisque nous ne l'avons pas utilisée avec les réelles estimations de la banque.

3.5 Analyse des résultats

3.5.1 Performances aux seuils par défaut

Pour GBoost, optimiser la precision produit une précision élevée (0.79) mais un rappel très faible (0.15), menant à un F1-score médiocre (0.25). L'optimisation du F1-score améliore légèrement le rappel (0.30) et le F1 (0.40). Pour XGBoost, les modèles non pondérés montrent une meilleure balance entre précision et rappel. Par exemple, le modèle optimisé pour le F1-score atteint un F1 de 0.41 avec une métrique métier plus compétitive (-308) comparée au modèle pondéré (-692).

3.5.2 Performances aux seuils optimaux

Pour GBoost, l'optimisation du seuil améliore légèrement la métrique métier (-292 pour la précision optimisée, -311 pour le F1 optimisé). Toutefois, le rappel reste bas (0.22-0.29), ce qui limite l'impact global. XGBoost montre une meilleure adaptation à l'optimisation de seuils. Par exemple, pour le modèle non pondéré avec optimisation du F1-score, le seuil optimal réduit la perte métier (-284) tout en équilibrant précision (0.71) et rappel (0.22).

Les deux tableaux mettent en évidence plusieurs points importants :

- Impact de l'optimisation du seuil: L'optimisation des seuils réduit légèrement la perte sur la métrique métier dans tous les cas.
- Comparaison entre GBoost et XGBoost: XGBoost montre une meilleure capacité d'adaptation avec des seuils optimisés, en particulier pour les modèles non pondérés. Cela reflète une capacité plus robuste à équilibrer précision et rappel.
- Rôle des pondérations des classes: Les modèles pondérés améliorent généralement le rappel, mais cela se fait au détriment de la précision et d'une perte accrue sur la métrique métier.

3.5.3 Analyse comparative des seuils pour GBoost et XGBoost (*F1-score*)

Lorsque les seuils sont optimisés pour maximiser la métrique métier, les résultats montrent des changements notables (Figure 3.2) : Pour GBoost, l'optimisation du seuil améliore légèrement la métrique métier (-292 contre -311 initialement). Le rappel reste faible (0.22-0.29), suggérant des difficultés à capturer suffisamment de cas positifs. Avec XGBoost, des seuils optimaux comme 0.73 et 0.71 augmentent la métrique métier, mais celle-ci reste négative. Le compromis entre précision et rappel est légèrement meilleur pour les modèles non pondérés.

Les figures 3.2 illustrent l'optimisation des seuils pour les modèles GBoost et XGBoost, avec *F1-score* comme métrique d'hyperparamétrage. Les courbes montrent que le modèle XGBoost optimisé pour le F1-score sans pondération atteint une AUC plus élevée (0.93) comparée au modèle pondéré (0.88). Cela reflète une meilleure capacité à équilibrer les

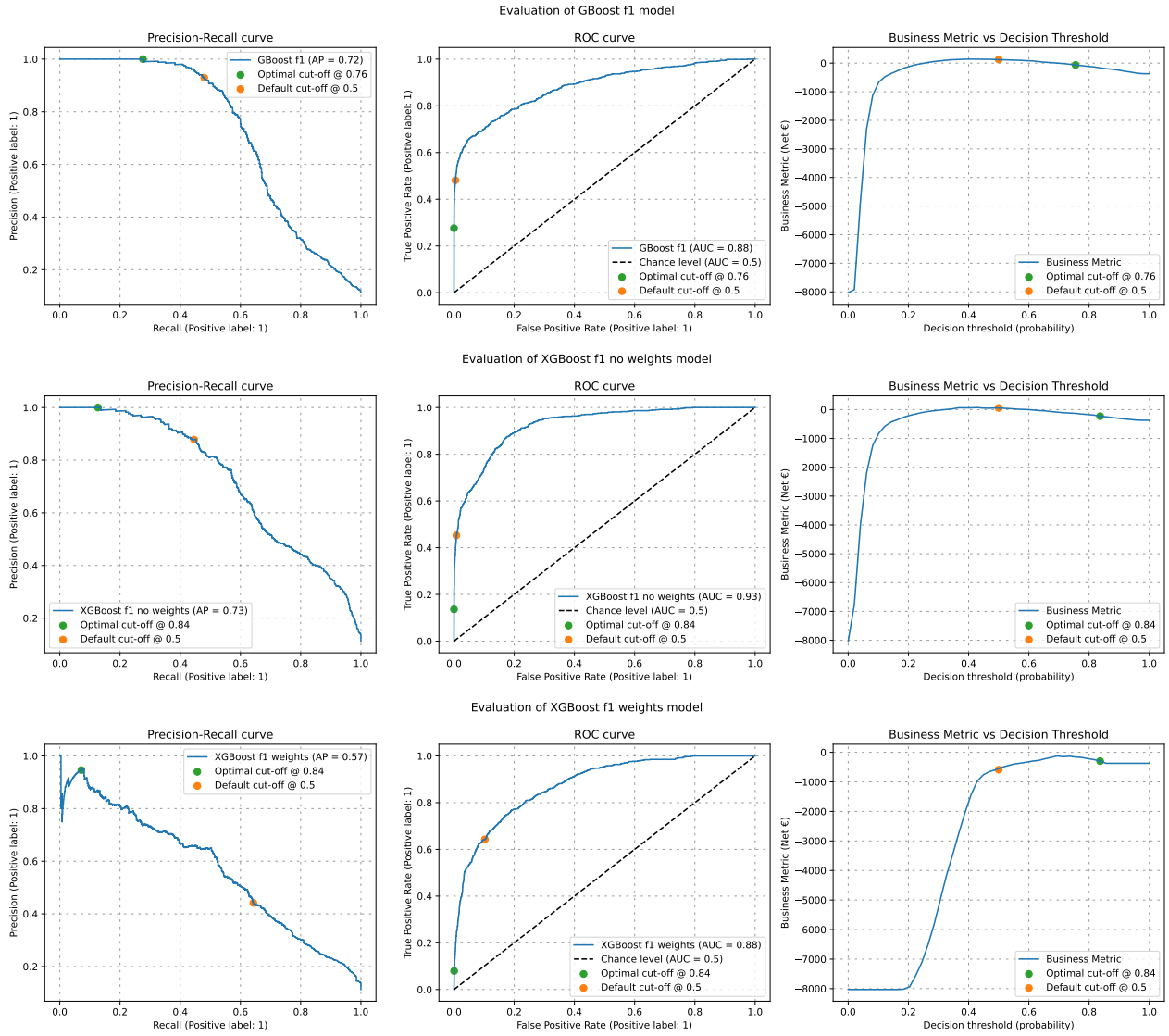

Figure 3.2: *Tresholds tuning des modèles où le F score est maximisé.*

taux de faux positifs et de vrais positifs. Pour GBoost, une AUC de 0.88 et une précision moyenne (AP) de 0.72 révèlent une performance correcte mais insuffisante pour une application pratique.

Conclusion. Les deux modèles présentent des avantages distincts. GBoost semble mieux équilibré et moins dépendant des ajustements du seuil pour atteindre des performances optimales. XGBoost, en revanche, nécessite un seuil plus élevé pour maximiser la métrique *business*, mais offre une discrimination équivalente selon la courbe ROC. Le choix entre ces deux modèles dépendra des priorités spécifiques, telles que la simplicité d'ajustement ou la sensibilité aux seuils optimisés.

4 Modèle B

Nous pouvons nous demander si nous pouvons obtenir des performances proches aux modèles précédents à l'aide d'un modèle plus simple, à savoir le modèle de régression logistique.
Nous proposons ici de mettre en œuvre un autre formatage des données.

4.1 Formatage des données alternatif

Le formatage des données que l'on va réaliser ici est le suivant:

- Suppression de `Default` car contient énormément de 'No' et uniquement trois 'Yes', comme vu lors de l'exploration. Il n'est donc pas pertinent de la garder
- Remplacement des 'unknown' par nan
- Suppression des individus avec nan, excepté les individus qui présentent des nan dans `education` car, comme vu lors de l'exploration des données, `education` contient un grand pourcentage de données manquantes.
- Encodage des données
- Remplacement des données manquantes par algorithme RandomForest dans `education` (la target étant `education`) afin d'affecter aux individus présentant des nan la valeur de la classe que le modèle RandomForest juge optimale (les hyperparamètres optimaux pour le modèle RandomForest sont ceux par défaut dans la fonction, trouvés par Cross validation)
- Sélection des variables quantitatives (Drop de `emp.var.rate` car très corrélée à `cons.price.idx` , `euribor3m` , `nr.employed`)
- Sélection des variables catégorielles selon un threshold de 0.75 (aucune n'est enlevée)
- Normalisation des données

4.2 Sampling

Le dataset est d'abord divisé en deux sous-ensembles distincts : 80% pour l'entraînement et 20% pour le test. Un échantillonnage stratifié est appliqué sur la variable cible afin de préserver la distribution des classes dans ces sous-ensembles. L'ensemble train nous permettra alors de réaliser une cross-validation et de trouver le threshold maximisant la métrique dédiée. Egalement, de l'OverSampling utilisant la technique SMOTE (Synthetic Minority Oversampling) sera effectué dans certains cas pour évaluer son impact sur les performances prédictives du modèle.

4.3 Hyperparamétrage

Nous utilisons les hyperparamètres suivants pour le modèle:

Hyperparamètre	Valeurs explorées
`penalty`	$['l_1', 'l_2']$
`C`	$[0.01, 0.1, 1, 10]$
`solver`	$['liblinear', 'saga']$

Table 4.1: *Hyperparamètres explorés lors de la validation croisée.*

'penalty' indique le type de régularisation (l1 ou l2), 'C' l'inverse du terme de pénalisation et 'solver' l'algorithme d'optimisation ('liblinear' pour une descente de gradient et 'saga' pour une descente de gradient stochastique).

4.4 Cross-Validation

Une validation croisée stratifiée à 5 plis est utilisée pour évaluer les performances des modèles. L'option shuffle=True est activée afin de mélanger les échantillons de chaque classe avant leur répartition dans les plis, garantissant ainsi une distribution équilibrée des classes. Nous proposons ici une cross-validation ciblant la précision, le f-1 score puis le f-beta score. Il est à noter qu'ici le f-beta score peut jouer le rôle de business metric; en effet, pour un paramètre $\beta < 1$, on donne plus de poids à la précision qu'au recall alors que pour un $\beta > 1$, c'est au recall qu'on donne plus de poids. Ici nous partons du principe que $\beta = 0.3$, c-à-d que l'on accorde trois fois plus d'importance à la maximisation de la précision que du recall.

4.5 Résultats

Métrique Optimisée	SMOTE Utilisé	Type de Seuil Utilisé	Seuil Optimal	Précision	Rappel	F1 Score	F-beta Score	Métrique Métier
precision	Oui	Seuil Optimal	0.95	0.755906	0.107023	0.187500	0.187500	-351.0
precision	Oui	Seuil Par Défaut (0.5)	0.50	0.289524	0.677815	0.405739	0.405739	-1774.5
precision	Non	Seuil Optimal	0.80	1.000000	0.001115	0.002227	0.002227	-447.0
precision	Non	Seuil Par Défaut (0.5)	0.50	0.697959	0.190635	0.299475	0.299475	-303.0
f1	Oui	Seuil Optimal	0.70	0.492380	0.468227	0.480000	0.480000	-468.0
f1	Oui	Seuil Par Défaut (0.5)	0.50	0.288703	0.692308	0.407480	0.407480	-1812.0
f1	Non	Seuil Optimal	0.20	0.432065	0.531773	0.476762	0.476762	-673.5
f1	Non	Seuil Par Défaut (0.5)	0.50	0.703448	0.227425	0.343724	0.343724	-271.5
fbeta	Oui	Seuil Optimal	0.90	0.706767	0.209588	0.323302	0.591007	-283.5
fbeta	Oui	Seuil Par Défaut (0.5)	0.50	0.289524	0.677815	0.405739	0.303898	-1774.5
fbeta	Non	Seuil Optimal	0.50	0.706294	0.225195	0.341505	0.600387	-271.5
fbeta	Non	Seuil Par Défaut (0.5)	0.50	0.706294	0.225195	0.341505	0.600387	-271.5

Table 4.2: *Tableau récapitulatif des métriques optimisées et configurations associées.*

Ici nous avons testé notre modèle sur notre ensemble de Test selon plusieurs configurations, en fonction de la métrique optimisée lors de la validation croisée, de la réalisation ou non du SMOTE, du type de seuil utilisé (si on utilise le seuil par défaut de 0.5 ou si on optimise notre seuil pour maximiser la métrique cible), les différentes valeurs des métriques, puis l'impact de chaque configuration sur la Business Metric comme définie précédem-

ment.

4.6 Analyse des résultats

Nous remarquons premièrement que l'oversampling diminue la performance de généralisation de notre modèle en règle générale, lorsque le seuil par défaut est pris. En effet, nous pouvons expliquer cela par le fait que réaliser de l'oversampling sur les données Train fausse la distribution des données réelles: en général, les personnes qui décrochent les appels sont rares et réaliser de l'oversampling enlève cette caractéristique d'événement rare, ce qui fait que quand on évalue notre modèle sur les données de Test (qui conservent la distribution réelle de l'événement décrocher à l'appel), les résultats sont plus dégradés.

Nous remarquons également que lorsque le seuil optimal est très élevé, (0.8 ou plus), la métrique de précision devient extrêmement élevée (étant égale à 1 parfois) au détriment des autres métriques (surtout du Recall, voir lignes 1 et 3) , signe que le modèle est extrêmement prudent: il préfère répondre Non très souvent que de se tromper (effet d'autant plus mis en exergue par un faible F-1 score signe d'un modèle très déséquilibré), ce qui n'est pas très intéressant.

Nous pouvons alors nous demander si optimiser le F-1 score ne nous permettrait pas d'atteindre un certain équilibre?
En effet, lorsque nous optimisons le F-1 Score par cross-validation et que l'on cherche le seuil optimal maximisant le F-1 score, on se retrouve avec des indicateurs tous proches de 0.5 (lignes 5 et 7). Toutefois, lorsque l'on utilise le seuil par défaut (sans SMOTE), nous perdons cet équilibre mais obtenons une configuration avec une business metric maximale.

En essayant de maximiser le F-β (en supposant $\beta = 0.3$,c-à-d que l'on donne trois fois plus d'importance à la précision qu'au Recall), nous obtenons deux résultats intéressants sur la Business Metric (ligne 9, avec SMOTE et seuil optimal de 0.9 avec une business metric de -283.3 et ligne 12, sans SMOTE, avec seuil par défaut (qui est ici égal au seuil optmial) avec une business metric maximale de -271.5). Le maximum de la business metric est également atteint pour une optimisation du F-1 score sans SMOTE et avec seuil par défaut. Lorsque la business metric est maximale, nous obtenons des valeurs de précision, de recall et de F-1 score très similaires (précision = 0.7 , recall = 0.22 , F-1 score = 0.34 , F-β score = 0.60). Nous obtenons une précision satisfaisante, un F-β score = 0.60 (qui peut aussi être vu comme une business metric) relativement haut, avec un recall plus élevé que lorsque l'on essaie de maximiser la précision (sans compter le SMOTE car il nous fait perdre trop d'argent dans le cas de la maximisation de la précision avec seuil par défaut). Ainsi nous avons réussi à obtenir un modèle plus audacieux et qui maximise notre business metric, que ce soit par la

méthode de l'optimisation du F-1 score ou du F-β score.

4.7 Conclusion

Le modèle de régression logistique obtenu avec ce formatage des données permet d'obtenir une meilleure performance que les modèles précédents (dans le sens des métriques définies ainsi que de la business metric), en étant moins complexe. Toutefois, de meilleures performances peuvent sûrement être atteintes avec plus de variables explicatives pertinentes afin de capturer une meilleure relation features-target, surtout dans notre cas où les classes dans la target ne sont pas du tout homogènes.

5 L'intégration des modèles dans le cadre d'une banque

Une autre question qui se pose à présent est comment la banque peut mettre en œuvre le modèle sélectionné (ici de régression logistique étant donnée qu'il est plus rapide à entraîner que les forêts et légèrement plus performant) de manière efficace pour automatiser le processus de démarchage ? La banque est un environnement par lequel transitent plusieurs données et en temps réel.

- Une première recommandation serait de développer des Pipelines de déploiement de modèle, selon une architecture bien définie et des business metrics pertinentes, et y intégrer des Pipeline ETL (Extract Transform Load) des données pour automatiser l'ensemble du processus depuis la collecte des données à l'enregistrement des résultats prédictifs.
- Il faudrait également mettre en place des systèmes de monitoring en temps réel afin de surveiller les données qui passent en temps réel par notre modèle. En effet, nous ne sommes pas à l'abri de changement progressif des données par rapport aux données sur lesquelles le modèle a été entraîné (data drift) et il faudrait définir un ensemble de graphes qui suivent cette évolution et pouvoir déclencher une alerte si un changement trop significatif a lieu (par exemple la différence entre les données sur lesquelles le modèle s'est entraîné et les données en temps réel dépasse un seuil à définir), donnant lieu ensuite à un réentrainement du modèle sur ces nouvelles données.
- On peut aussi penser à développer des tests automatisés pour vérifier l'intégrité des modèles (test d'équité, biais, performance sur des données nouvelles) et intégrer des frameworks de tests pour valider les transformations et les prédictions.

6 Conclusion

Ce projet a été très formateur quant aux différents aspects qui interviennent dans la démarche de Data Science. Il a mis en exergue l'importance du traitement des données et son impact sur les performances du modèle, le caractère itératif de l'entraînement et du test des modèles, l'importance du choix de la métrique convenable et l'importance de réfléchir à comment déployer le modèle convenablement. Nous sommes assez satisfaits des performances de nos modèles, toutefois, nous pensons qu'avec plus de variables explicatives, nous pouvons arriver à de meilleures performances.

A Schéma

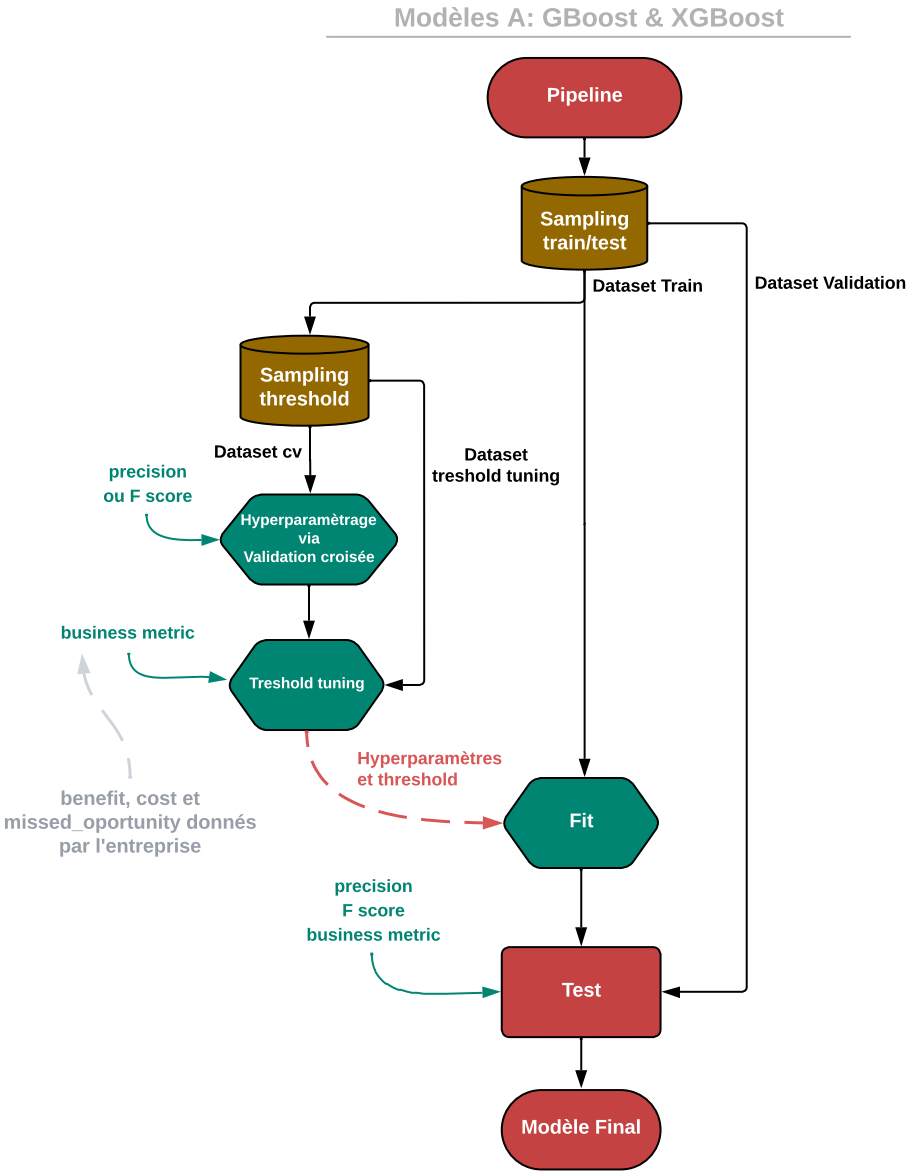

Figure A.1: *Version verticale du schéma résumant des six modèles A (GBoost, XGBoost, avec et sans poids, optimisé en validation croisée avec la precision ou la F score).*