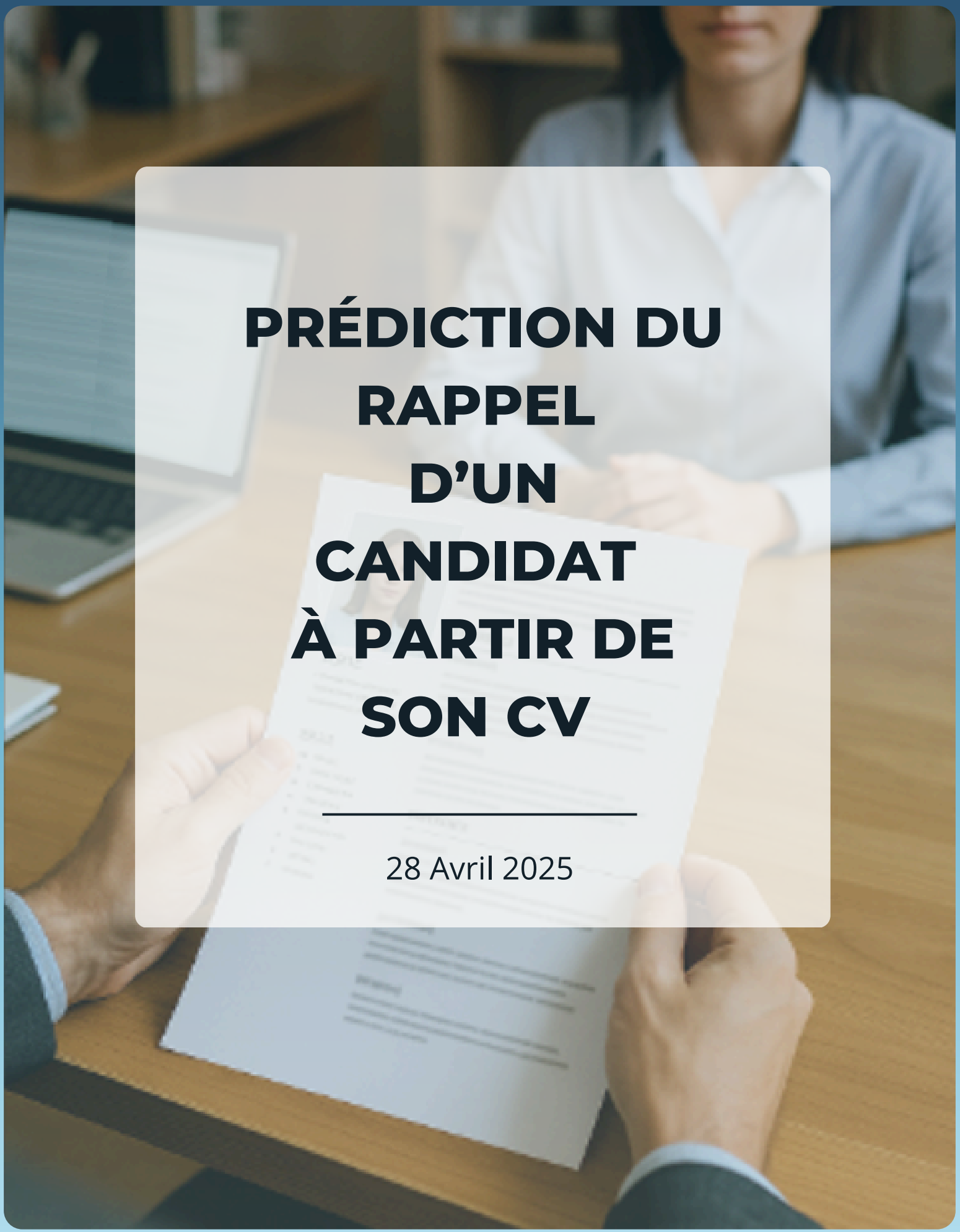


PRÉDICTION DU RAPPEL D'UN CANDIDAT À PARTIR DE SON CV

28 Avril 2025

Rousseau Nora
Boudamous Lyna

Sommaire

1 Introduction

Dans un marché de l'emploi de plus en plus compétitif, le processus de recrutement constitue un enjeu majeur pour les entreprises. La sélection des candidats repose sur de nombreux critères, souvent subjectifs, qui influencent leur probabilité d'être rappelés après l'envoi de leur CV. Comprendre ces critères et prédire cette sélection peut permettre d'optimiser les processus de recrutement et d'assurer une meilleure adéquation entre les candidats et les attentes des employeurs.

L'objectif de ce projet est de développer un modèle capable de prédire si un candidat sera rappelé ou non en fonction des informations contenues dans son CV. Pour cela, nous testerons et comparerons divers modèles de classification supervisée, tels que l'analyse discriminante linéaire (LDA), l'analyse discriminante quadratique (QDA), les arbres de décision et plusieurs autres algorithmes. L'enjeu principal sera d'identifier le modèle offrant la meilleure performance prédictive, afin de mieux comprendre les éléments clés qui influencent la décision de rappel des candidats.

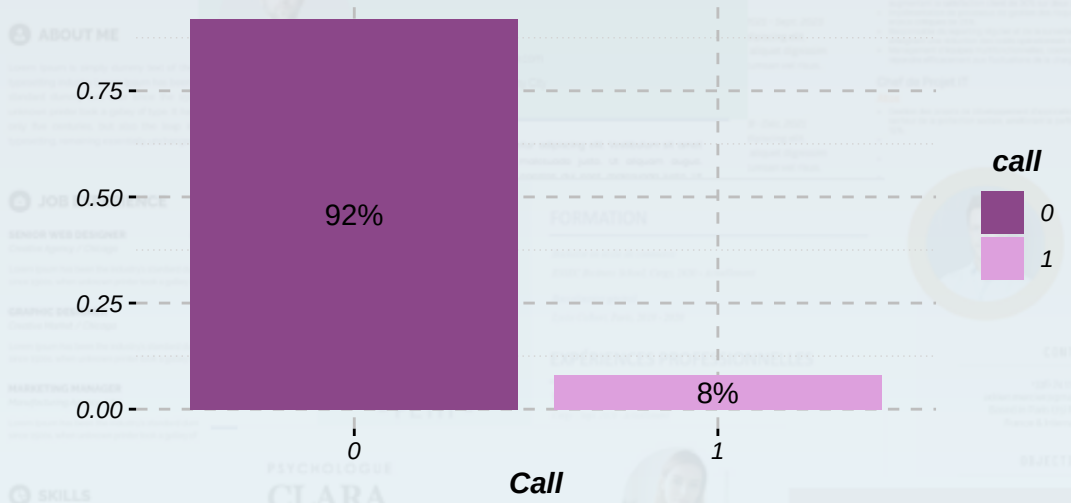
Pour mener à bien cette étude, nous utilisons une base de données contenant 4 870 CV fictifs envoyés en réponse à des offres d'emploi à Chicago et Boston en 2001. Cette base de données comporte 26 variables, permettant d'examiner l'influence des informations individuelles et des compétences perçues sur les chances d'être rappelé par un recruteur. L'absence de valeurs manquantes garantit une exploitation optimale des données, facilitant ainsi l'application et l'évaluation des différents modèles de classification.

	Nom de la variable	Type	Description
1	gender	factor	Sexe du candidat
2	ethnicity	factor	Origine ethnique (prénom à consonance caucasienne ou afro-américaine)
3	quality	factor	Qualité du CV
4	call	factor	Le candidat a-t-il été rappelé (1:Oui, 0:Non)
5	city	factor	Ville concernée (Boston ou Chicago)
6	honors	factor	Le CV mentionnait-il des distinctions ?
7	volunteer	factor	Le CV mentionnait-il une expérience de bénévolat ?
8	military	factor	Le candidat a-t-il une expérience militaire ?
9	holes	factor	Le CV comporte-t-il des périodes d'inactivité ?
10	school	factor	Le CV mentionne-t-il une expérience professionnelle pendant les études ?
11	email	factor	L'adresse e-mail figurait-elle sur le CV du candidat ?
12	computer	factor	Le CV mentionne-t-il des compétences en informatique ?
13	special	factor	Le CV mentionne-t-il des compétences particulières ?
14	college	factor	Le candidat a-t-il un diplôme universitaire ou plus ?
15	equal	factor	L'employeur est-il pour l'égalité des chances en matière d'emploi ?
16	wanted	factor	Type de poste recherché par l'employeur
17	requirements	factor	L'annonce mentionne-t-elle des exigences pour le poste ?
18	reqexp	factor	L'annonce mentionne-t-elle des exigences d'expérience ?
19	reqcomm	factor	L'annonce mentionne-t-elle des compétences en communication ?
20	reqeduc	factor	L'annonce mentionne-t-elle des exigences en matière de diplôme ?
21	reqcomp	factor	L'annonce mentionne-t-elle des compétences informatiques requises ?
22	reqorg	factor	L'annonce mentionne-t-elle des compétences organisationnelles requises ?
23	industry	factor	Secteur d'activité de l'employeur
24	jobs	integer	Nombre d'emplois répertoriés sur le CV
25	experience	integer	Nombre d'années d'expérience de travail sur le CV
26	minimum	numeric	Expérience minimale exigée de l'employeur

Avant de construire les modèles prédictifs, une exploration des données s'impose afin d'en cerner les spécificités et guider les choix méthodologiques à venir.

2 Exploration des données

2.1 Visulisation de la fréquence d'appel des candidats rappelés et non rappelés



On observe un déséquilibre marqué dans la distribution de la variable à prédire call, avec 92 % des candidats non rappelés contre seulement 8 % qui l'ont été.

Cette disparité peut poser problème lors de l'application de méthodes de prédiction, car elle favorise la classe majoritaire, augmentant ainsi le risque de sur-ajustement. En conséquence, le modèle pourrait avoir des performances réduites pour identifier correctement la classe minoritaire.

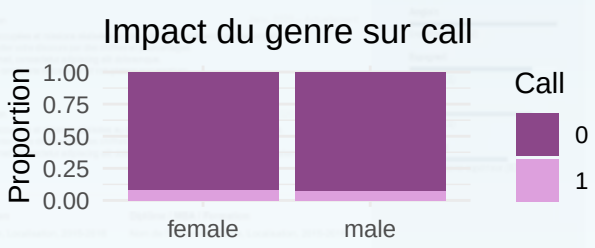
Il est donc crucial d'adopter des stratégies adaptées, comme le rééquilibrage des classes, afin d'améliorer la fiabilité des prédictions.

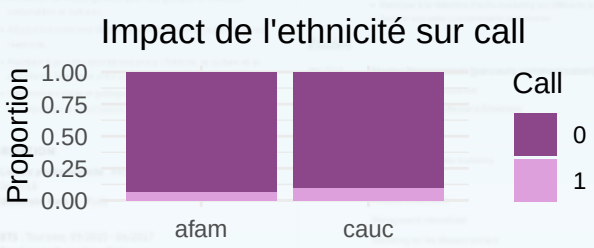
2.2 Analyse des variables qualitatives

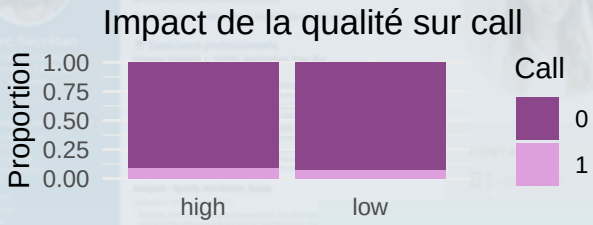
2.2.1 Proportions

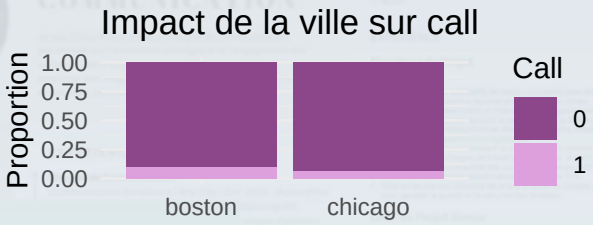
Afin de déterminer la relation entre nos variables qualitatives et la variable à prédire "call" (la probabilité d'être rappelé), nous analyserons les proportions de rappel pour chacune de leurs modalités. Les graphiques suivants illustrent cette analyse en montrant les proportions de rappel en fonction de différentes informations issues du CV des candidats, telles que le genre, l'origine ethnique ou la qualité du CV, ainsi que les exigences du poste auquel le candidat a postulé.

Ces visualisations permettent d'observer comment chaque modalité influence la probabilité d'être rappelé (Call), offrant ainsi une compréhension plus claire des facteurs liés au rappel des candidats.

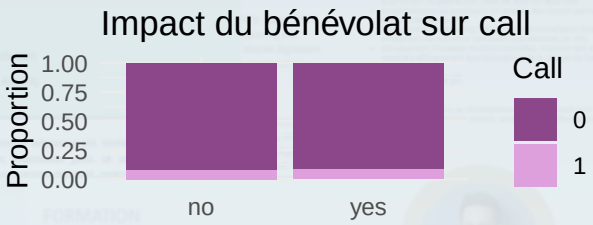


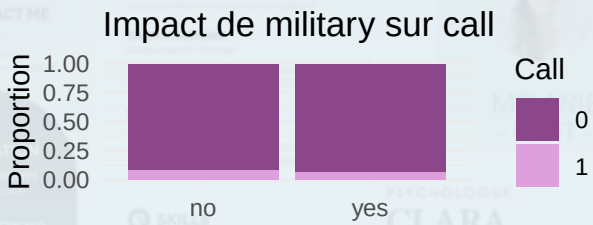



Impact de la qualité sur call
Proportion
1.00
0.75
0.50
0.25
0.00
high
low
Call
0
1

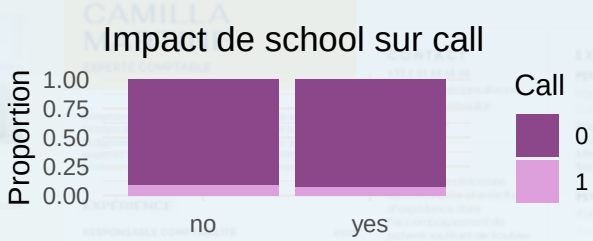

Impact de la ville sur call
Proportion
1.00
0.75
0.50
0.25
0.00
boston
chicago
Call
0
1

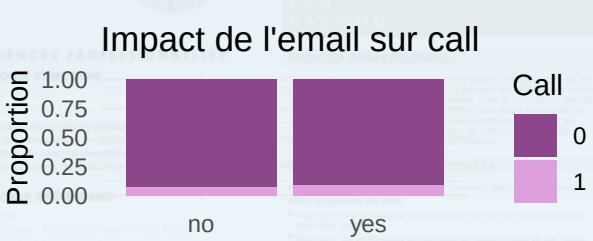

Impact des honneurs sur call
Proportion
1.00
0.75
0.50
0.25
0.00
no
yes
Call
0
1

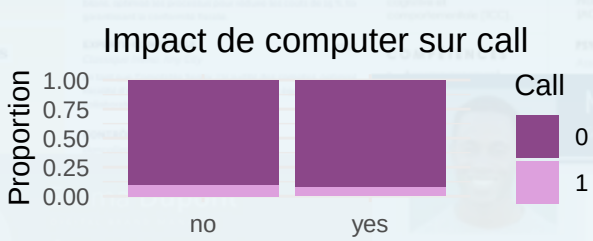

Impact du bénévolat sur call
Proportion
1.00
0.75
0.50
0.25
0.00
no
yes
Call
0
1

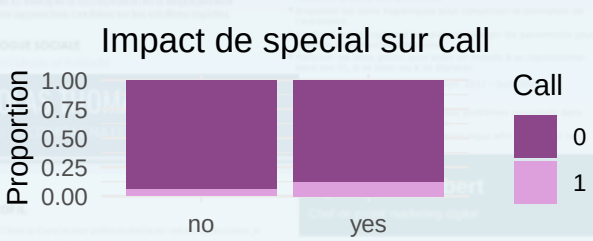

Impact de military sur call
Proportion
1.00
0.75
0.50
0.25
0.00
no
yes
Call
0
1


Impact des holes sur call
Proportion
1.00
0.75
0.50
0.25
0.00
no
yes
Call
0
1

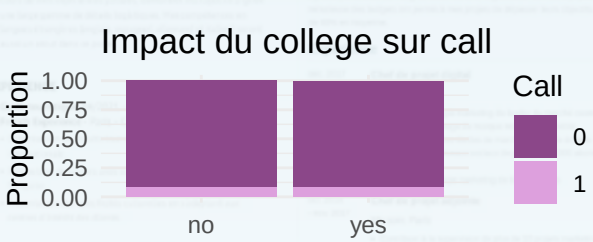

Impact de school sur call
Proportion
1.00
0.75
0.50
0.25
0.00
no
yes
Call
0
1

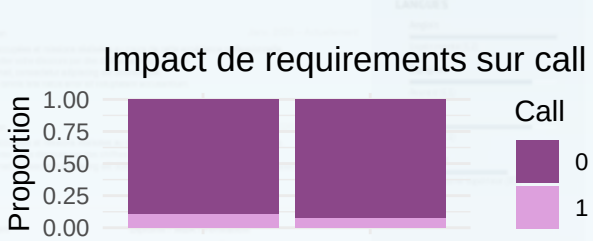

Impact de l'email sur call
Proportion
1.00
0.75
0.50
0.25
0.00
no
yes
Call
0
1

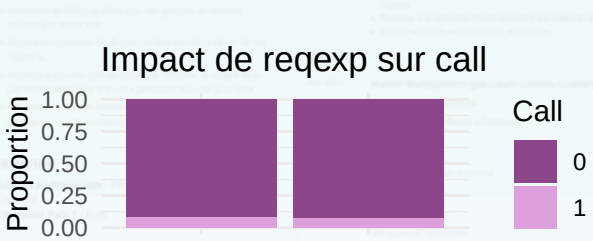

Impact de computer sur call
Proportion
1.00
0.75
0.50
0.25
0.00
no
yes
Call
0
1

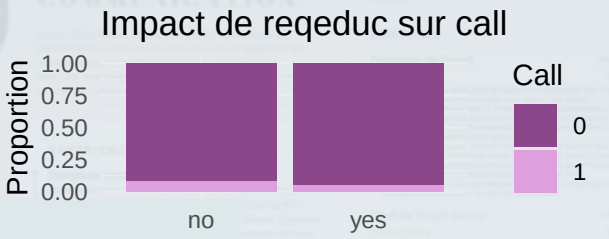

Impact de special sur call
Proportion
1.00
0.75
0.50
0.25
0.00
no
yes
Call
0
1

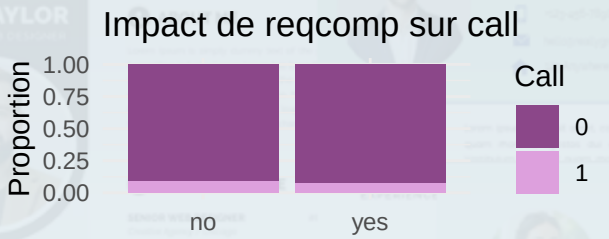

Impact de equal sur call
Proportion
1.00
0.75
0.50
0.25
0.00
no
yes
Call
0
1

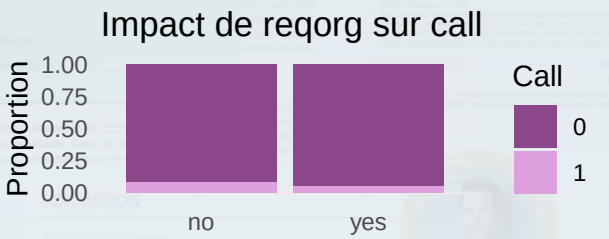

Impact du college sur call
Proportion
1.00
0.75
0.50
0.25
0.00
no
yes
Call
0
1

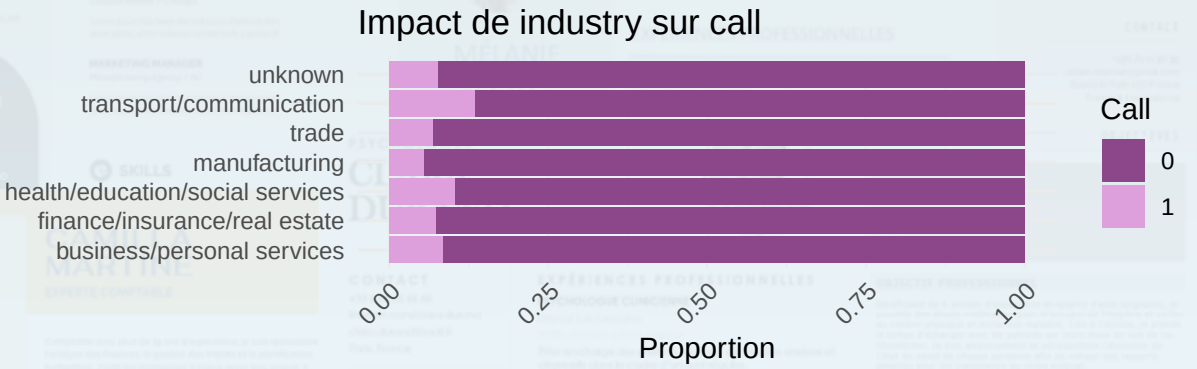

Impact de requirements sur call
Proportion
1.00
0.75
0.50
0.25
0.00
no
yes
Call
0
1

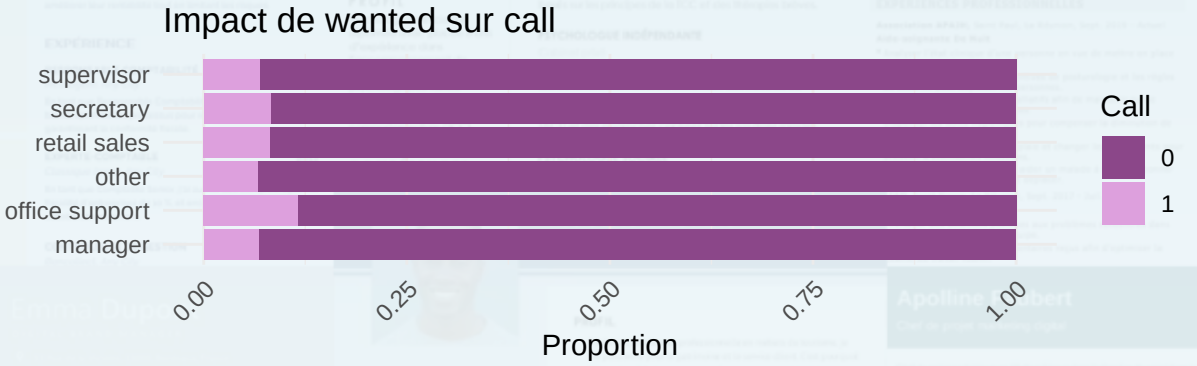

Impact de reqexp sur call
Proportion
1.00
0.75
0.50
0.25
0.00
no
yes
Call
0
1











Les graphiques ci-dessus mettent en évidence le déséquilibre de classe entre les candidats rappelés et ceux qui ne l'ont pas été, ce qui complique l'interprétation des proportions de call pour chaque modalité de nos variables qualitatives.

Dans l'ensemble, la plupart des variables ne présentent quasiment aucune différence visible entre les candidats rappelés et ceux qui ne l'ont pas été. On observe néanmoins de légères différences pour les variables "honors", "industry" et "wanted". Les candidats avec des distinctions (honors) semblent légèrement plus souvent rappelés. Les employeurs dans les secteurs d'activité "transport/communication" (industry) rappellent davantage de candidats que ceux d'autres secteurs. Enfin, les candidats postulant pour les postes "office support" (wanted) sont plus fréquemment rappelés que les autres, probablement en raison d'une demande plus forte ou de critères de sélection moins stricts dans ce domaine.

2.2.2 Tests du χ^2

Afin de compléter l'analyse des proportions, nous allons appliquer un test du χ^2 pour chacune des variables qualitatives pour évaluer si certaines sont statistiquement liées à la variable call, c'est-à-dire si des informations présentes dans le CV influencent la probabilité d'être rappelé. Les hypothèses testées sont les suivantes :

$$\begin{cases} H_0 : \text{les variables sont indépendantes} \\ H_1 : \text{les variables sont dépendantes} \end{cases}$$

Ces tests nous permettront de déterminer si les différences observées dans les proportions sont statistiquement significatives, ou simplement dues au hasard.

Table 2: Résultats des tests

	t-stat	p-value
Gender	7.60e-01	3.83e-01
Ethnicity	1.64e+01	5.00e-05
Quality	3.06e+00	8.01e-02
City	1.39e+01	1.95e-04
Honors	2.40e+01	9.42e-07
Volunteer	2.01e-01	6.54e-01
Military	1.81e+00	1.78e-01
Holes	2.46e+01	6.91e-07
School	3.58e+00	5.83e-02
Email	3.07e+00	7.95e-02
Computer	3.77e+00	5.21e-02
Special	5.92e+01	1.41e-14
College	2.84e-01	5.94e-01
Equal	2.49e-02	8.75e-01
Wanted	1.42e+01	1.47e-02
Requirements	8.10e+00	4.43e-03
Reqexp	1.66e+00	1.97e-01
Reqcomm	7.97e-03	9.29e-01
Reqeduc	5.19e+00	2.27e-02
Reqcomp	2.84e+00	9.20e-02
Reqorg	4.98e+00	2.57e-02
Industry	1.76e+01	7.23e-03

Les résultats des tests nous révèlent que les variables Ethnicity, City, Honors, Holes, Special, Wanted, Requirements, Reqeduc, Reqorg et Industry sont statistiquement liées à la probabilité d'être rappelé par le recruteur, les p-values associées étant inférieures à 0.05, ce qui n'est pas le cas des autres variables.

Ces résultats confirment nos observations concernant les variables Honors, Wanted et Industry, pour lesquelles des différences visibles dans les proportions de candidats rappelés et non rappelés ont été notées. Par ailleurs, bien que certaines autres variables significatives ne montrent pas de différences évidentes dans les graphiques de proportions, elles sont néanmoins statistiquement liées à la probabilité d'être rappelé, ce qui suggère qu'elles peuvent également jouer un rôle, même minime, dans le processus de sélection des candidats.

2.3 Analyse des variables quantitatives

2.3.1 Tests de Student pour échantillons indépendants

Après avoir analysé les variables qualitatives, nous allons maintenant nous concentrer sur les variables quantitatives. Pour cela, nous allons réaliser des tests de Student. Ces tests nous permettront de vérifier si les moyennes des variables quantitatives sont significativement différentes entre les candidats rappelés et ceux non rappelés, ce qui pourrait indiquer que certaines de ces variables influencent la décision du recruteur de rappeler un candidat.
Voici les hypothèses des tests :

$$\begin{cases} H_0 : \mu_1 = \mu_2, \text{il n'y a pas de différence significative entre les moyennes des deux groupes} \\ H_1 : \mu_1 \neq \mu_2, \text{il y a une différence significative entre les moyennes des deux groupes} \end{cases}$$

Table 3: Résultats des tests de Student

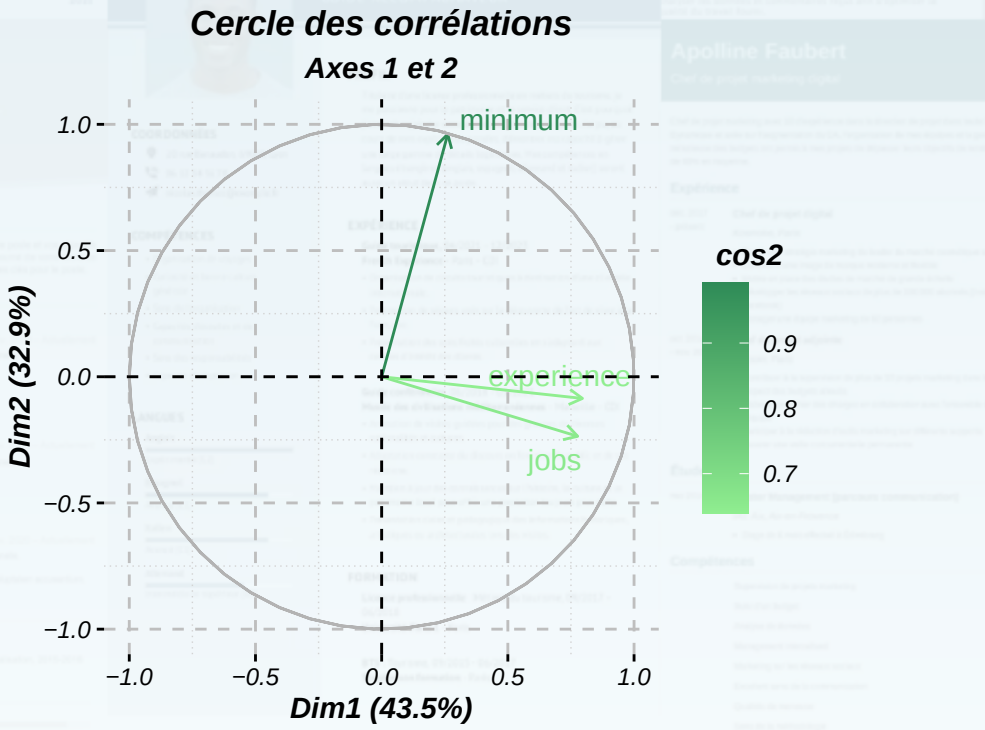
	Variable	Estimate	μ1	μ2	p-value	IC inf	IC sup
1	jobs	-0.01	3.66	3.67	8.80e-01	-0.15	0.12
2	experience	-1.14	7.75	8.89	9.57e-05	-1.71	-0.57
3	minimum	0.09	0.97	0.88	2.06e-01	-0.05	0.24

On remarque une différence de moyenne pour la variable Expérience entre le groupe des rappelés et celui des non rappelés. De plus, cette différence est statistiquement significative, car sa p-value est inférieure à 0.05. Cela suggère que l'expérience influence probablement la probabilité d'être rappelé.

Pour les variables Jobs et Minimum, les moyennes des deux groupes sont très proches, et les p-values supérieures à 0.05 indiquent qu'il n'y a pas de différence significative. Ainsi, ces variables ne semblent pas être liées à la probabilité d'être rappelé.

2.3.2 ACP

Nous allons faire une ACP afin de déterminer les corrélations entre nos variables quantitatives et d'identifier d'éventuelles redondances.



Nous pouvons observons sur le graphique, que la variable Minimum est bien représentée sur le plan factoriel formé par les axes F1 (43,5%) et F2 (32,9%), comme l'indique son cos^2 élevé (vert foncée).

Cette variable contribue principalement à l'axe F2 et partiellement à l'axe F1, jouant un rôle majeur dans l'explication de la variance des données.

On remarque que les variables Experience et Jobs sont relativement proches sur le plan F1, F2 ce qui indique qu'elles sont corrélés bien qu'elles soient moins bien représenté sur ce plan que Minimum.

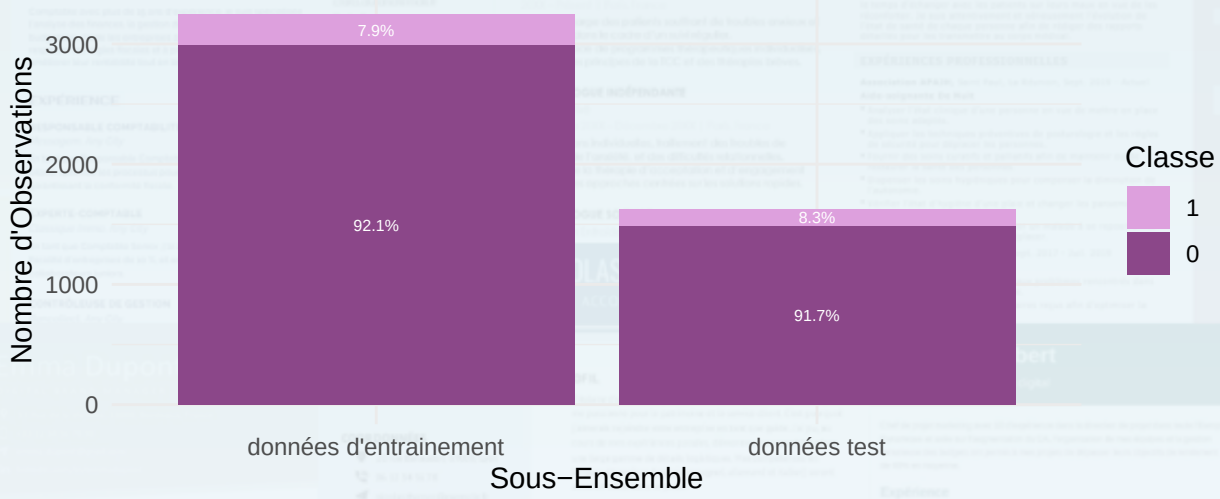
Nous utiliserons donc step_corr pour tenter d'atténuer ce problème de colinéarité, qui fera en sorte de sélectionner un groupe de variables dont le coefficient de corrélation maximal n'excédera pas un seuil spécifié, garantissant ainsi une indépendance suffisante entre elles.

3 Préparation des données

3.1 Découpage train/test

Nous avons décider de découper notre jeu de données en deux parties : 2/3 des observations seront utilisées pour l'apprentissage et 1/3 pour le test du modèle entrainé

3.1.1 Graphique de répartition des classes dans les sous-Ensembles



Dans notre ensemble d'entraînement, 92,1 % des échantillons appartiennent à la classe 0, contre 7,9 % pour la classe 1. De manière similaire, l'ensemble de test contient 91,7 % d'échantillons de la classe 0 et 8,3 % de la classe 1.

3.2 Optimisation des hyperparamètres

En raison du déséquilibre prononcé dans la distribution de la variable cible call, nous avons mis en place une stratégie de rééquilibrage des classes pour améliorer la capacité prédictive de nos modèles. Pour cela, nous utiliserons dans les recettes de nos modèles la technique de sur-échantillonnage synthétique SMOTENC, une version de SMOTE (Synthetic Minority Over-sampling Technique) adaptée aux données mixtes (numériques et catégorielles) , qui permet de générer artificiellement de nouvelles observations pour la classe minoritaire (call=1). Cette méthode vise à améliorer la détection de ces cas rares mais essentiels pour notre analyse.

Les paramètres retenus pour la méthode SMOTE sont over_ratio et neighbors. Over_ratio permet de fixer la proportion d'observations synthétiques à générer relativement à la classe majoritaire, afin de rééquilibrer efficacement les classes. Quant à Neighbors, il détermine le nombre de voisins les plus proches utilisés lors de la création des exemples synthétiques, garantissant ainsi que ces nouvelles observations restent cohérentes avec la structure locale des données existantes.

Les autres étapes de prétraitement appliquées aux différents modèles, avec des variations selon les spécificités algorithmiques, incluent : la transformation des variables qualitatives en variables binaires (step_dummy), la suppression des variables à variance quasi-nulle (step_zv), la normalisation, le centrage et la réduction des variables numériques (step_normalize, step_center, step_scale), la réduction de la dimensionnalité par ACP (step_pca), la suppression des variables fortement corrélées (step_corr), ainsi que le sous-échantillonnage de la classe majoritaire (step_downsample). Les recettes de chaque modèle seront mis en annexe.

Pour évaluer et optimiser la performance des modèles, nous nous baserons principalement sur le F1-score :

$$\text{F1-score} = \frac{2 \times (\text{Précision} \times \text{Sensibilité})}{\text{Précision} + \text{Sensibilité}}$$

avec Sensibilité :

$$\text{Sensibilité} = \frac{VP}{VP + FN}$$

et Précision :

$$\text{Précision} = \frac{VP}{VP + FP}$$

Nous examinerons également l'accuracy (exactitude) définie comme suit :

$$\text{Accuracy} = \frac{VP + VN}{VP + VN + FP + FN}$$

avec

VP : Vrais Positifs (candidats correctement prédits comme rappelés)

VN : Vrais Négatifs (candidats correctement prédits comme non rappelés)

FP : Faux Positifs (candidats prédits comme rappelés alors qu'ils ne l'ont pas été.)

FN : Faux Négatifs (candidats prédits comme non rappelés alors qu'ils l'ont été.)

Matrice de confusion : prédiction des candidats rappelés

Réalité / Prédiction	n'ont pas été rappelés(0)	ont été rappelés (1)
n'ont pas été rappelés(0)	VN	FP
ont été rappelés (1)	FN	VP

Enfin, nous utiliserons également la courbe ROC et l'indicateur AUC afin de sélectionner le meilleur modèle possible, c'est-à-dire celui qui minimisera simultanément les erreurs de classification des deux classes.

$$AUC \approx 1 \Rightarrow \text{Excellent modèle}, \quad AUC \approx 0.5 : \text{modèle aléatoire}$$

Nous estimons qu'un modèle avec une AUC proche de 1 possède une excellente capacité de discrimination et constitue donc le meilleur choix. Notre objectif est d'optimiser les performances afin d'identifier avec précision les candidats rappelés tout en réduisant le nombre d'erreurs.

Pour garantir une évaluation des modèles robuste, nous avons utilisé une validation croisée à 10 plis avec 3 répétitions.

4 Modèles

Dans cette partie, nous allons tester et comparer plusieurs modèles de classification supervisée afin de prédire si un candidat est rappelé ou non par le recruteur. Pour cela, nous appliquerons les modèles suivants :

- Analyse Discriminante Linéaire (LDA)
- Analyse Discriminante Quadratique (QDA)
- k-plus proches voisins (k-NN)
- Support Vector Machine à noyau radial (SVM Radial)
- Support Vector Machine linéaire (SVM Linéaire)
- Régression logistique (Logit)
- Arbre de décision
- Forêt Aléatoire
- Boosting

4.1 Analyse discriminante linéaire (LDA)

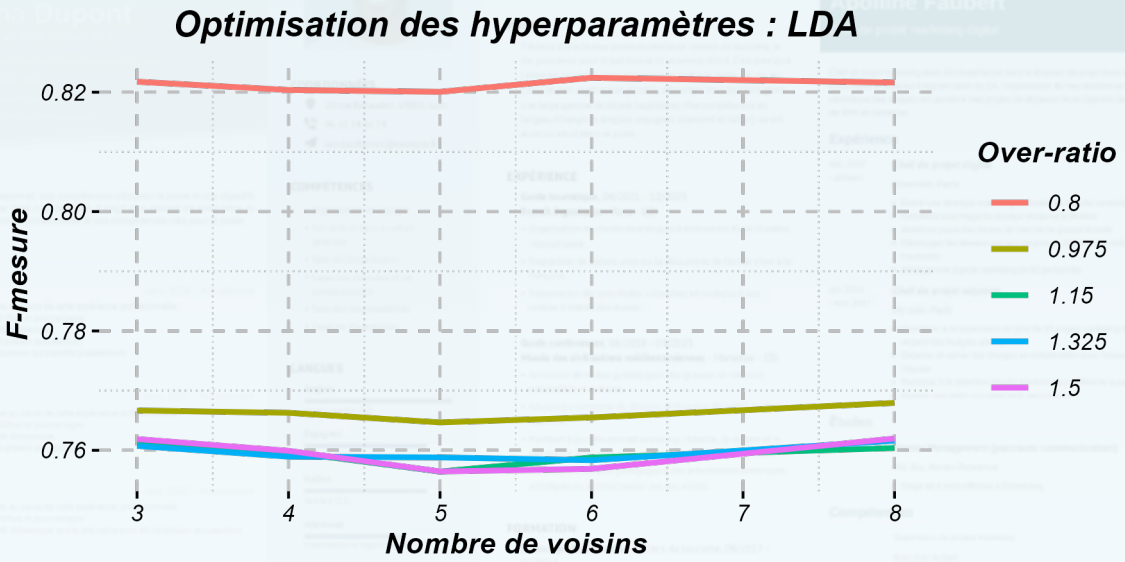
La LDA représente une approche simple mais souvent efficace, ce qui en fait un bon choix pour initier notre étude. Lorsqu'une structure linéaire domine les relations entre variables, ce modèle peut offrir des performances intéressantes. Malgré sa simplicité, il arrive qu'il surpasse des méthodes plus complexes, d'où l'intérêt de ne pas l'écarter trop tôt.

Optimisation des hyperparamètres :

Le modèle LDA a été optimisé en utilisant la métrique F1-score.

Table 4: Hyperparamètres retenus - LDA

Neighbors	Over-ratio
6	0.8



Performances du modèle :

Table 5: Matrice de confusion : LDA

		Prédiction		
		0	1	Total
Réalité	0	1076	70	**1146**
	1	413	65	**478**
	Total	**1489**	**135**	**1624**

Table 6: Performances du modèle : LDA

Précision	48.15%
Spécificité	93.89%
Taux de faux positifis	6.11%
Taux de faux négatifs	86.40%
Sensibilité	13.60%
F-score	21.21%
Taux d'erreur test	29.74%
Taux d'erreur train	28.65%
AUC	61.74%
Accuracy	70.26%

[1] Taux de faux positifs : Proportion des candidats prédits comme rappelés alors qu'ils ne l'ont pas été.

[2] Taux de faux négatifs : Proportion des candidats prédits comme non rappelés alors qu'ils l'ont été.

Malgré une recette incluant la méthode SMOTENC pour traiter le déséquilibre des classes, le modèle LDA se révèle inadapté pour repérer efficacement les candidats rappelés. Sa sensibilité très basse (13.60%) et son F-score limité (21.21%) traduisent une incapacité à identifier une part significative des profils rappelés. Face à ces résultats peu satisfaisants, nous poursuivons l'analyse avec un autre modèle dans l'espoir d'obtenir de meilleures performances.

4.2 Analyse Discriminante Quadratique (QDA)

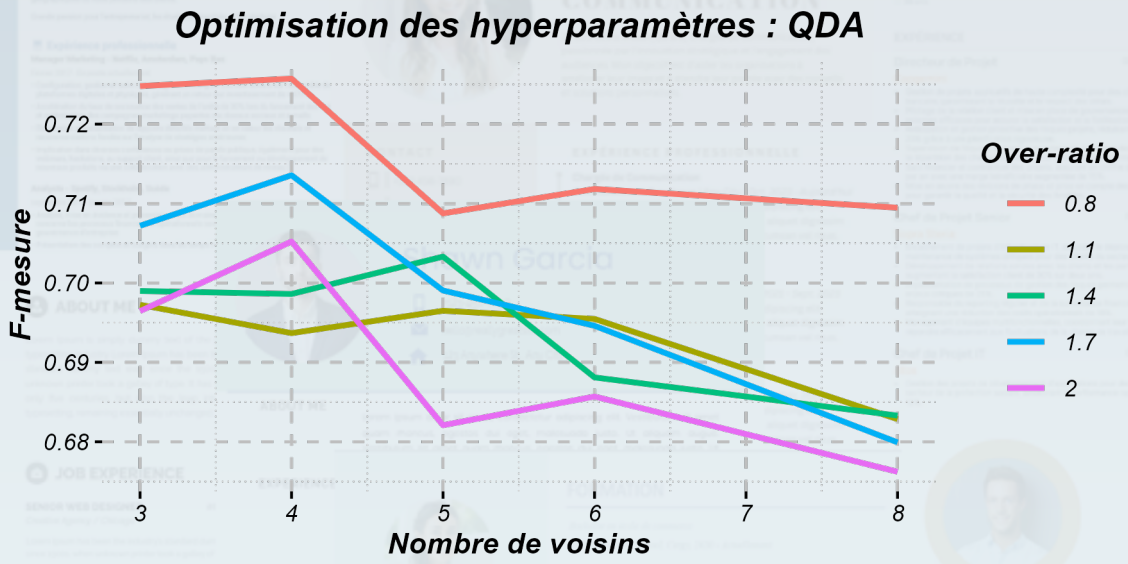
La QDA constitue une version plus flexible de la LDA, capable de modéliser des relations non linéaires entre les variables. Ce modèle peut être pertinent lorsque les groupes à prédire présentent des structures de variance différentes. Bien qu'il soit plus complexe, il peut dans certains cas mieux s'adapter aux données, ce qui justifie son exploration dans le cadre de notre étude.

Optimisation des hyperparamètres :

Le modèle QDA a été optimisé en utilisant la métrique F1-score.

Table 7: Hyperparamètres retenus - QDA

Neighbors	Over-ratio
4	0.8



Performances du modèle :

Table 8: Matrice de confusion : QDA

		Prédiction		
		0	1	Total
Réalité	0	832	76	**908**
	1	657	59	**716**
	Total	**1489**	**135**	**1624**

Table 9: Performances du modèle : QDA

Précision	43.70%
Spécificité	91.63%
Taux de faux positifs	8.37%
Taux de faux négatifs	91.76%
Sensibilité	8.24%
F-score	13.87%
Taux d'erreur test	45.14%
Taux d'erreur train	42.17%
AUC	47.15%
Accuracy	54.86%

[1] Taux de faux positifs : Proportion des candidats prédits comme rappelés alors qu'ils ne l'ont pas été.

[2] Taux de faux négatifs : Proportion des candidats prédits comme non rappelés alors qu'ils l'ont été.

Le modèle QDA, malgré l'optimisation des hyperparamètres, présente des résultats globalement moins bons que le modèle LDA. Sa précision de 43.70% et sa faible sensibilité (8.24%) montrent qu'il a des difficultés à identifier les candidats rappelés. Le faible F-score de 13.87% et le taux d'erreur élevé en test et train confirme que ce modèle n'est pas optimal pour prédire les candidats rappelés, et ne répond donc pas efficacement à l'objectif de l'étude.

4.3 k-plus proches voisins (k-NN)

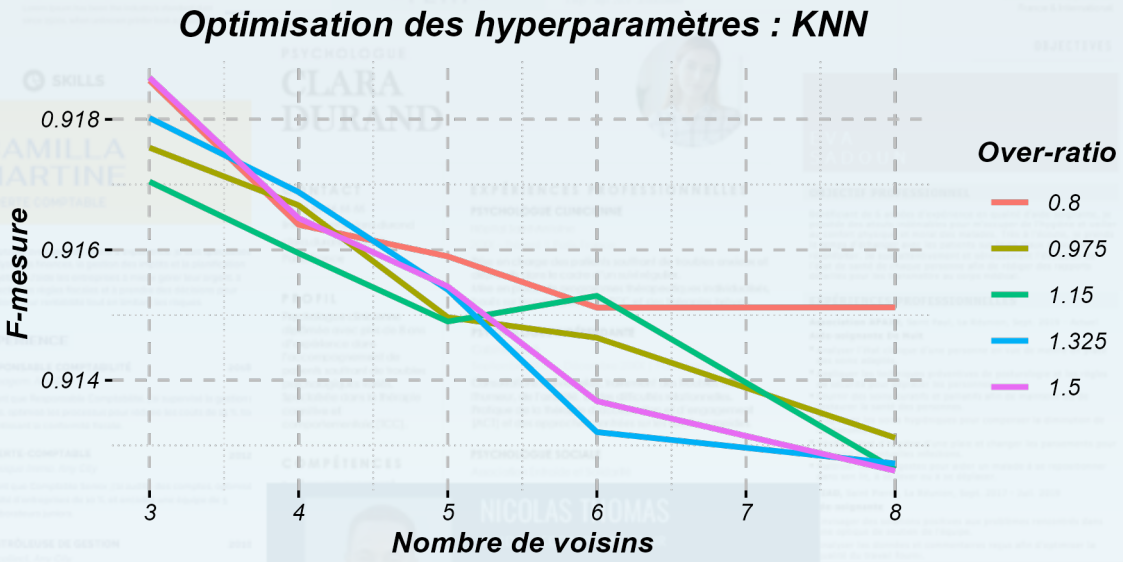
Le modèle KNN est un modèle simple et intuitif basé sur la notion de voisinage : il prédit en se référant aux observations les plus proches dans l'espace des données. Sa flexibilité face à des relations complexes fait de lui un modèle pertinent à tester, bien que ses performances dépendent fortement du choix du nombre de voisins et du prétraitement des variables.

Optimisation des hyperparamètres :

Le modèle KNN a été optimisé en utilisant la métrique F1-score.

Table 10: Hyperparamètres retenus - KNN

Neighbors	Over-ratio
3	1.5



Performances du modèle :

Table 11: Matrice de confusion : KNN

		Prédiction		
		0	1	Total
Réalité	0	1362	112	**1474**
	1	127	23	**150**
	Total	**1489**	**135**	**1624**

Table 12: Performances du modèle : KNN

Précision	17.04%
Spécificité	92.40%
Taux de faux positifs	7.60%
Taux de faux négatifs	84.67%
Sensibilité	15.33%
F-score	16.14%
Taux d'erreur test	14.72%
Taux d'erreur train	0.71%
AUC	60.69%
Accuracy	85.28%

[1] Taux de faux positifs : Proportion des candidats prédits comme rappelés alors qu'ils ne l'ont pas été.
[2] Taux de faux négatifs : Proportion des candidats prédits comme non rappelés alors qu'ils l'ont été.

Le modèle KNN présente une faible précision de 17.04% et une sensibilité de 15.33%, ce qui indique qu'il peine à identifier les candidats rappelés. Le F-score de 16.14% montre un compromis médiocre entre précision et rappel. La performance globale du modèle KNN reste insuffisante pour une prédiction fiable des candidats rappelés. En comparaison, le modèle LDA semble plus adapté pour prédire la classe 1, avec une précision et un F-score plus élevés.

4.4 SVM Radial

Le SVM à noyau radial est un modèle performant lorsqu'il s'agit de capturer des relations non linéaires entre les variables. Grâce à son noyau, il projette les données dans un espace de dimensions supérieures pour mieux les séparer. Ce modèle est particulièrement utile quand les frontières entre classes sont complexes. Toutefois, son efficacité repose sur le réglage de deux hyperparamètres : le coût, qui gère le compromis entre marge et erreurs de classification, et le noyau rbf_sigma, qui détermine la portée de l'influence d'un point sur la frontière. Un ajustement approprié de ces paramètres permet d'éviter le surapprentissage ou un modèle trop simpliste.

Optimisation des hyperparamètres :

Le modèle SVM Radial a été optimisé en utilisant la métrique ROC-AUC.

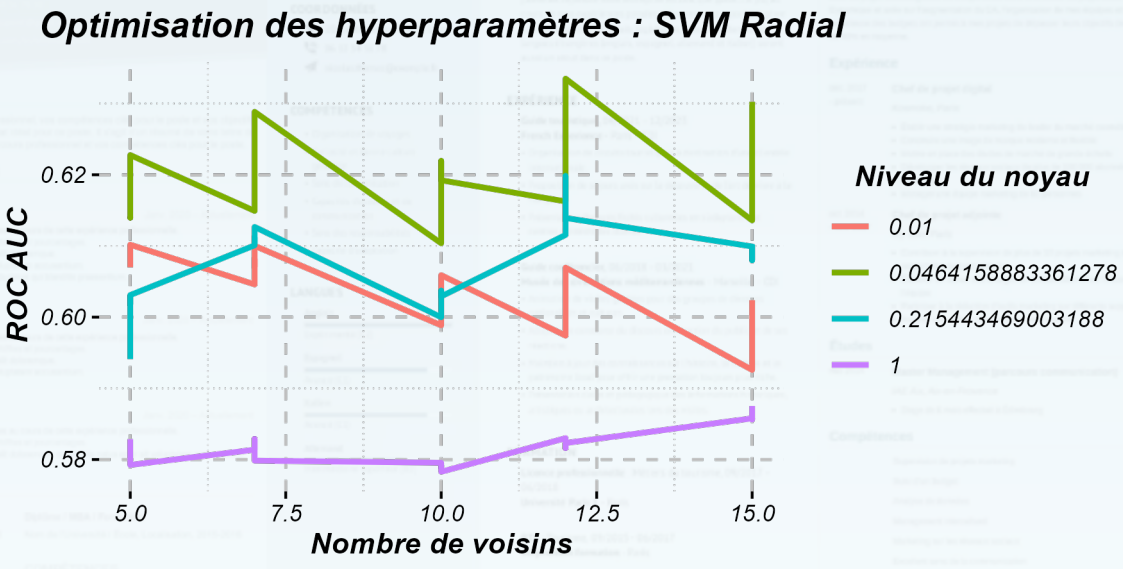


Table 13: Hyperparamètres retenus - SVM Radial

Cout	Niveau du Noyau	Taux de re-échantillonnage	Nombres de voisins utilisé
2	0.05	0.875	12

Performances du modèle :

Table 14: Matrice de confusion : SVM Radial

		Prédiction		
		0	1	Total
Réalité	0	1331	112	**1443**
	1	158	23	**181**
	Total	**1489**	**135**	**1624**

Table 15: Performances du modèle : SVM Radial

Précision	17.04%
Spécificité	92.24%
Taux de faux positifs	7.76%
Taux de faux négatifs	87.29%
Sensibilité	12.71%
F-score	14.56%
Taux d'erreur test	16.63%
Taux d'erreur train	8.01%
AUC	63.23%
Accuracy	83.37%

[1] Taux de faux positifs : Proportion des candidats prédits comme rappelés alors qu'ils ne l'ont pas été.

[2] Taux de faux négatifs : Proportion des candidats prédits comme non rappelés alors qu'ils l'ont été.

Bien que la SVM radial affiche une bonne accuracy (83.37 %) et une forte spécificité (92.24 %), sa capacité à identifier les candidats rappelés reste très limitée comme le montre la sensibilité médiocre et le faible F-score. Le taux élevé de faux négatifs (87.29 %) indique que le modèle passe à côté de la majorité des cas positifs, malgré les tentatives de correction du déséquilibre. Ces résultats suggèrent que le modèle reste largement biaisé en faveur de la classe majoritaire, il répond donc pas à à l'objectif de prédire efficacement les rappelés.

4.5 SVM Linéaire

La SVM linéaire est un modèle qui cherche à séparer les classes par une droite ou un plan, en maximisant la marge entre elles. Elle est efficace lorsque les données sont séparables de façon linéaire, mais moins performante si la relation entre les variables est complexe. Le principal paramètre à optimiser ici est cost, qui gère l'équilibre entre une séparation stricte des classes et l'autorisation d'erreurs.

Optimisation des hyperparamètres :

Le modèle SVM Linéaire a été optimisé en utilisant la métrique F1-score.

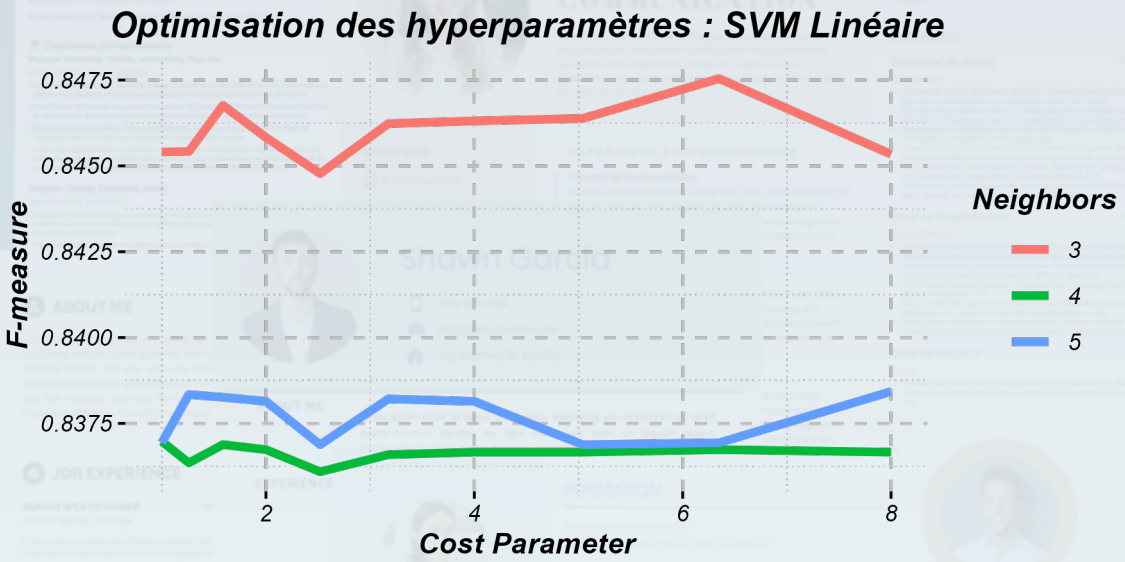


Table 16: Hyperparamètres retenus - SVM Linéaire

	Cout	Taux de re-échantillonnage	Nombres de voisins utilisé
1	6.3496	0.7	3

Performances du modèle :

Table 17: Matrice de confusion : SVM Linéaire

		Prédiction		
		0	1	Total
Réalité	0	1134	87	**1221**
	1	355	48	**403**
	Total	**1489**	**135**	**1624**

Table 18: Performances du modèle : SVM Linéaire

Précision	35.56%
Spécificité	92.87%
Taux de faux positifs	7.13%
Taux de faux négatifs	88.09%
Sensibilité	11.91%
F-score	17.84%
Taux d'erreur test	27.22%
Taux d'erreur train	23.63%
AUC	57.41%
Accuracy	72.78%

[1] Taux de faux positifs : Proportion des candidats prédits comme rappelés alors qu'ils ne l'ont pas été.

[2] Taux de faux négatifs : Proportion des candidats prédits comme non rappelés alors qu'ils l'ont été.

Bien que la spécificité atteigne un solide 92,87 %, la sensibilité reste désastreusement basse à 11,91 %, traduisant un taux de faux négatifs de 88,09 %, tandis que le F-score plafonne à seulement 17,84 %. Cette disparité met en évidence le fait que, malgré une bonne détection des candidats non rappelés, le modèle SVM linéaire manque la quasi-totalité des profils pertinents à rappeler.

4.6 Régression Logistique (Logit)

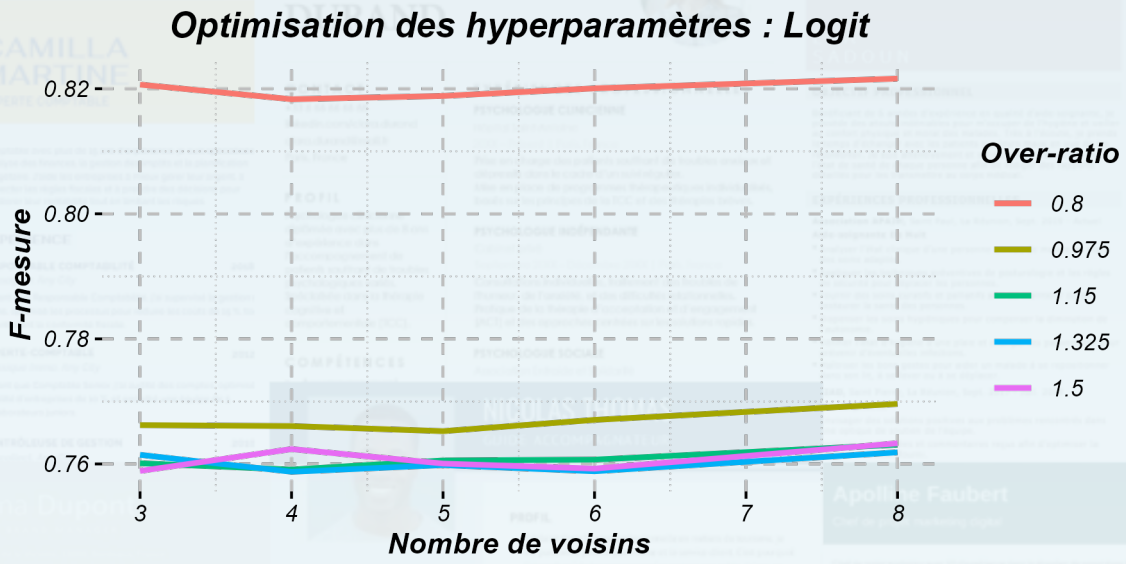
Le modèle Logit, ou régression logistique, est un modèle linéaire utilisé pour la classification binaire. Il estime la probabilité d'appartenance à une classe à partir des variables explicatives, offrant ainsi simplicité et interprétabilité.

Optimisation des hyperparamètres :

Le modèle Logit a été optimisé en utilisant la métrique F1-score.

Table 19: Hyperparamètres retenus - Logit

Neighbors	Over-ratio
8	0.8



Performances du modèle :

Table 20: Matrice de confusion : Logit

		Prédiction		
		0	1	Total
Réalité	0	1075	72	**1147**
	1	414	63	**477**
	Total	**1489**	**135**	**1624**

Table 21: Performances du modèle : Logit

Précision	46.67%
Spécificité	93.72%
Taux de faux positifis	6.28%
Taux de faux négatifs	86.79%
Sensibilité	13.21%
F-score	20.59%
Taux d'erreur test	29.93%
Taux d'erreur train	28.13%
AUC	60.73%
Accuracy	70.07%

[1] Taux de faux positifs : Proportion des candidats prédits comme rappelés alors qu'ils ne l'ont pas été.
[2] Taux de faux négatifs : Proportion des candidats prédits comme non rappelés alors qu'ils l'ont été.

Le modèle Logit présente une bonne spécificité (93.72%), ce qui signifie qu'il est efficace pour éviter les faux positifs. Cependant, sa faible sensibilité de 13.21% montre qu'il a du mal à identifier les candidats rappelés. Cela indique que, bien qu'il soit fiable pour prédire les cas négatifs, il reste insuffisant pour prédire efficacement les candidats rappeler. En comparaison, les résultats de LDA surpassent légèrement ceux du modèle Logit dans l'ensemble, bien que la différence reste modeste.

4.7 Arbre de décision

L'arbre de décision est un modèle intuitif qui segmente les données en suivant des règles simples basées sur les variables explicatives. Il s'adapte bien aux variables qualitatives et résiste aux valeurs aberrantes. Cependant, il est sensible au surapprentissage, ce qui nécessite d'optimiser le paramètre de complexité, afin de limiter la croissance excessive de l'arbre et d'améliorer sa généralisation.

Optimisation des hyperparamètres :

Le modèle Arbre de décision a été optimisé en utilisant la métrique Accuracy.

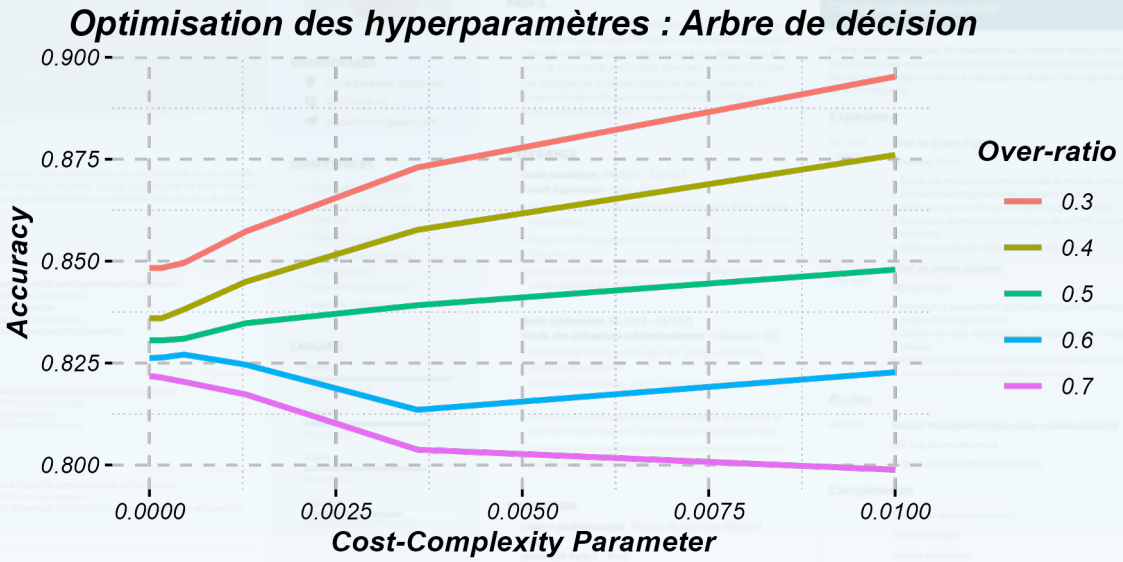
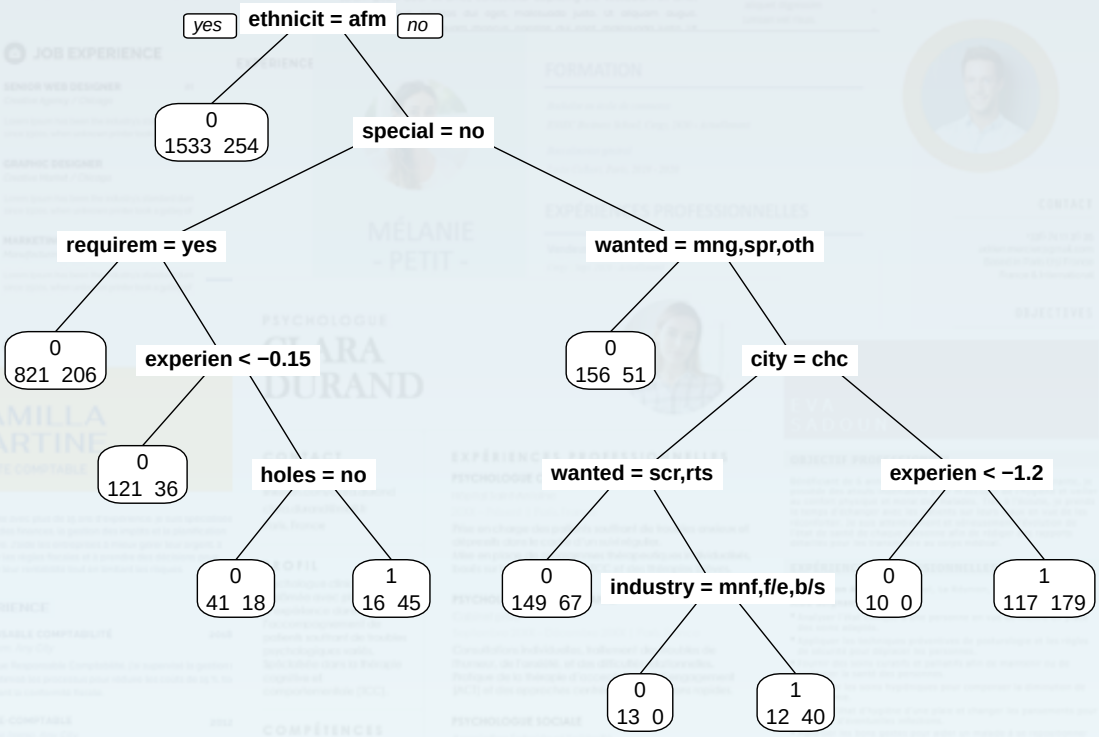


Table 22: Hyperparamètres retenus - Arbre de décision

Coût de complexité	Taux de re-échantillonnage	Nombres de voisins utilisé
1.00e-02	0.3	5

Arbre de décision :



L'arbre de décision met en évidence les variables les plus discriminantes dans la prédiction des candidats rappelés. La racine de l'arbre, ethnicity = afam, montre que l'origine ethnique des candidats joue un rôle central dans la décision du modèle. Le fait que l'arbre ne poursuive aucune division pour ethnicity = afam "yes" et prédit systématiquement l'absence de rappel pour les candidats qui ont un prénom à consonnance afro-américaine révèle un biais du modèle, influencé par un déséquilibre de classes. Bien que 254 candidats afro-américains aient été rappelés, l'arbre les classe tous à tort comme "non rappelés", illustrant ainsi comment le déséquilibre entre les classes (rappelés vs non rappelés) affecte la performance du modèle, conduisant à une généralisation erronée pour cette catégorie.

En revanche, pour les candidats non afro-américains, l'arbre considère des critères supplémentaires : l'absence de compétences particulières sur le CV (special = no), la correspondance avec les exigences du poste (requirem = yes), et enfin le type de poste recherché, à savoir management, office support ou supervisor (wanted). Ces variables sont utilisées pour affiner la décision du modèle et mieux prédire les candidats rappelés.

Performances du modèle :

Table 23: Matrice de confusion : Arbre de décision

		Prédiction		
		0	1	Total
Réalité	0	1408	109	**1517**
	1	81	26	**107**
	Total	**1489**	**135**	**1624**

Table 24: Performances du modèle : Arbre de décision

Précision	19.26%
Spécificité	92.81%
Taux de faux positifs	7.19%
Taux de faux négatifs	75.70%
Sensibilité	24.30%
F-score	21.49%
Taux d'erreur test	11.70%
Taux d'erreur train	10.66%
AUC	56.00%
Accuracy	88.30%

[1] Taux de faux positifs : Proportion des candidats prédits comme rappelés alors qu'ils ne l'ont pas été.

[2] Taux de faux négatifs : Proportion des candidats prédits comme non rappelés alors qu'ils l'ont été.

Le modèle arbre de décision présente une sensibilité de 24.30%, la meilleure parmi les modèles testés jusqu'à présent, ce qui indique une meilleure capacité à identifier les candidats rappelés. Son F-score de 21.49% et son accuracy de 88.30% montrent qu'il parvient à classifier une majorité de cas, mais la performance pour prédire les candidats rappelés reste insuffisante, surtout comparée à ce qu'on pourrait espérer pour une prédiction plus fiable. Bien que l'accuracy soit élevée, elle est biaisée par le déséquilibre des classes, car le modèle classe facilement les candidats non rappelés.

4.8 Forêt Aléatoire

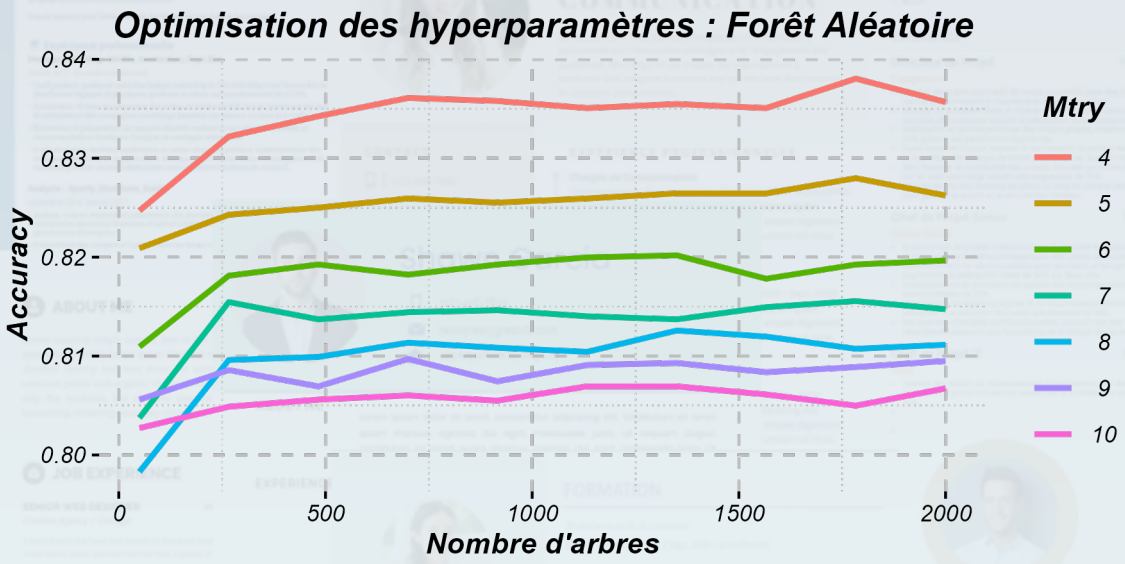
La forêt aléatoire est un modèle basé sur un ensemble d'arbres de décision, chacun construit à partir de sous-ensembles aléatoires de données. Ce modèle est robuste et efficace pour traiter des données complexes. Ces performances dépendent principalement de deux paramètres à optimiser : le nombre d'arbres (trees) et le nombre de variables à utiliser pour chaque division (mtry).

Optimisation des hyperparamètres :

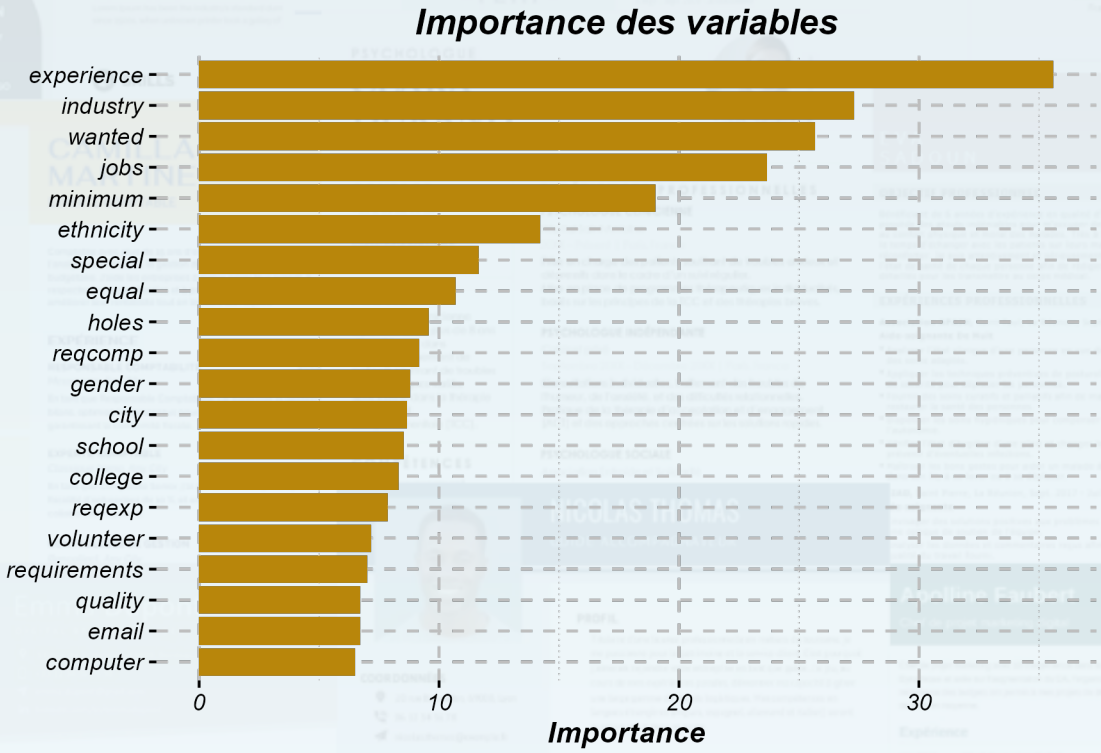
Le modèle Forêt Aléatoire a été optimisé en utilisant la métrique Accuracy.

Table 25: Hyperparamètres retenus - Forêt Aléatoire

Mtry	Nombre d'arbres	Min nodesize	Over-ratio	Neighbors
4	1783	1	1	3



Importance des variables :



Le graphique d'importance des variables montre la contribution de chaque variable aux prédictions du modèle. On constate que le nombre d'années d'expérience professionnelle est le facteur le plus déterminant dans la décision de rappel des recruteurs, suivi du secteur d'activité de l'employeur (industry), du type de poste recherché (wanted) et du nombre d'emplois mentionnés sur le CV (jobs). Ces quatre variables jouent un rôle central dans la décision du recruteur de recontacter ou non un candidat.

Performances du modèle :

Table 26: Matrice de confusion : Forêt Aléatoire

		Prédiction		
		0	1	Total
Réalité	0	1292	97	**1389**
	1	197	38	**235**
	Total	**1489**	**135**	**1624**

Table 27: Performances du modèle : Random Forest

Précision	28.15%
Spécificité	93.02%
Taux de faux positifs	6.98%
Taux de faux négatifs	83.83%
Sensibilité	16.17%
F-score	20.54%
Erreurs OOB	20.20%
Taux d'erreur en test	18.10%
Taux d'erreur en entraînement	10.72%
AUC	63.58%
Accuracy	81.90%

[1] Taux de faux positifs : Proportion des candidats prédits comme rappelés alors qu'ils ne l'ont pas été.
[2] Taux de faux négatifs : Proportion des candidats prédits comme non rappelés alors qu'ils l'ont pas été.

La forêt aléatoire présente une meilleure précision (28.15%) par rapport à l'arbre de décision, mais ce dernier affiche une sensibilité plus élevée (24.30%) et un F-score de 21.49%, ce qui montre qu'il est légèrement plus efficace pour identifier les candidats rappelés. L'erreur OOB (20.20%) de la forêt aléatoire suggère qu'elle généralise bien sur des données non vues, bien que sa capacité à prédire correctement les candidats rappelés reste limitée. On s'attendait à ce que la forêt aléatoire surpasse l'arbre de décision en raison de sa capacité à combiner plusieurs arbres, mais le déséquilibre des classes semble avoir un impact plus important, réduisant ainsi son efficacité à prédire les candidats rappelés.

4.9 Boosting

Place maintenant au dernier modèle étudié, le boosting, un modèle d'ensemble qui combine plusieurs modèles faibles, généralement des arbres de décision, afin de créer une prédiction plus robuste et précise. Ce modèle est particulièrement efficace pour améliorer les performances en se concentrant sur les erreurs commises lors des itérations précédentes. Le boosting peut être sensible au bruit et aux valeurs aberrantes, car il met fortement l'accent sur les erreurs précédemment commises, ce qui peut entraîner un surapprentissage si le modèle n'est pas bien régularisé. Les paramètres clés à optimiser pour ce modèle incluent le nombre d'arbres (trees), la profondeur des arbres (depth), le taux d'apprentissage (learning_rate) et le nombre de variables à utiliser pour chaque division (mtry).

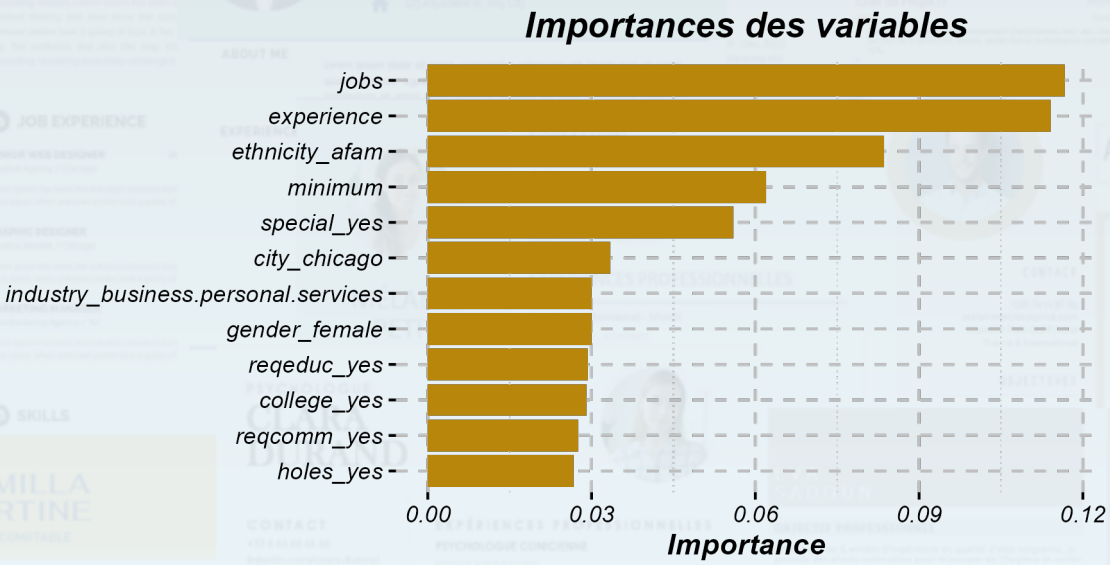
Optimisation des hyperparamètres :

Le modèle Boosting a été optimisé en utilisant la métrique ROC-AUC.

Table 28: Hyperparamètres retenus - Boosting

Mtry	Nombre d'arbres	Profondeur d'arbre	Taux d'apprentissage
4	525	15	0.001

Importances des variables :



On remarque que les variables qui influencent le plus la décision de rappel du recruteur sont le nombre d'emplois listés sur le CV (jobs), le nombre d'années d'expérience (experience) et l'origine perçue du candidat à travers son prénom, notamment lorsqu'il est à consonance afro-américaine (ethnicity = afam).

Performances du modèle :

Table 29: Matrice de confusion : Boosting

		Prédiction		
		0	1	Total
Réalité	0	1361	110	**1471**
	1	128	25	**153**
	Total	**1489**	**135**	**1624**

Table 30: Performances du modèle : Boosting

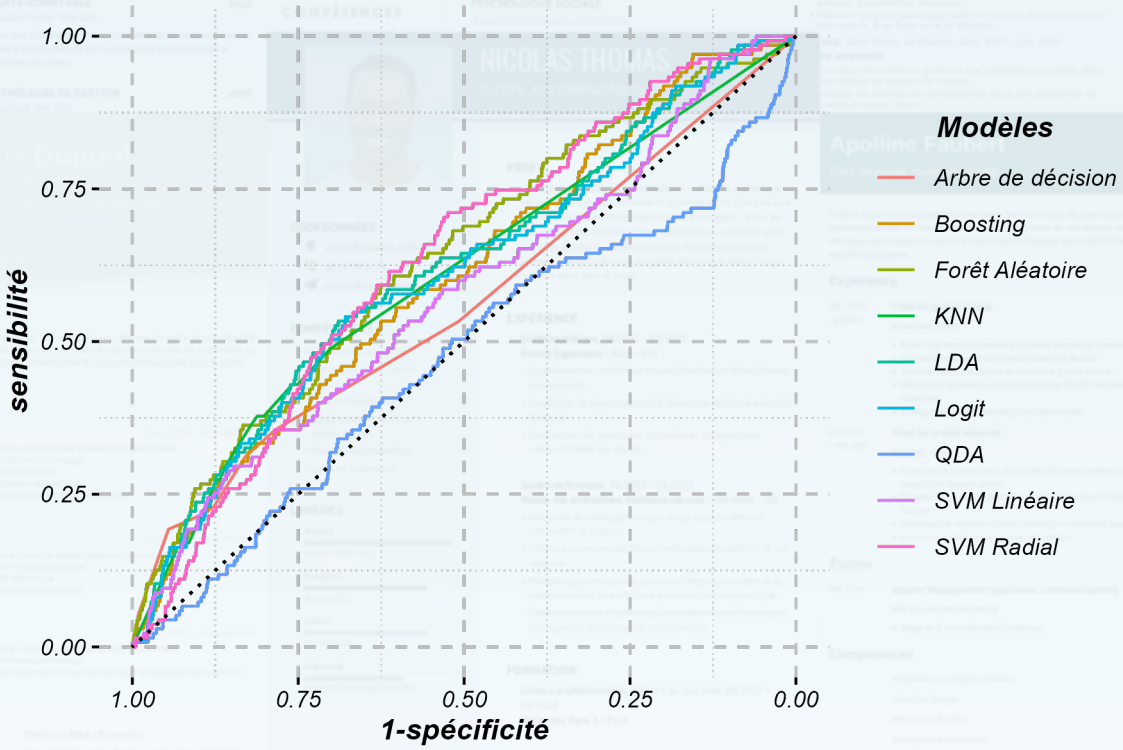
Précision	18.52%
Spécificité	92.52%
Taux de faux positifis	7.48%
Taux de faux négatifs	83.66%
Sensibilité	16.34%
F-score	17.36%
Taux d'erreur test	14.66%
Taux d'erreur train	9.15%
AUC	60.47%
Accuracy	85.34%

[1] Taux de faux positifs : Proportion des candidats prédits comme rappelés alors qu'ils ne l'ont pas été.
[2] Taux de faux négatifs : Proportion des candidats prédits comme non rappelés alors qu'ils l'ont été.

Le modèle de boosting montre de faibles performances pour prédire les candidats rappelés, avec une sensibilité de 16.34% et un F-score de 17.36%, indiquant qu'il en identifie peu correctement. En comparaison, l'arbre de décision, bien que limité, se révèle plus performant pour cet objectif, ce qui en fait un choix relativement plus adapté à la détection des rappels dans ce contexte.

5 Comparaison des modèles étudiés

5.1 Courbes ROC des différents modèles



5.2 Comparaison des performances des modèles

	Précision	Sensibilité	F_score	AUC	Accuracy
LDA	0.4815	0.1360	0.2121	0.6174	0.7026
QDA	0.4370	0.0824	0.1387	0.4715	0.5486
KNN	0.1704	0.1533	0.1614	0.6069	0.8528
SVM Linéaire	0.3556	0.1191	0.1784	0.5741	0.7278
SVM Radial	0.1704	0.1271	0.1456	0.6323	0.8337
Logit	0.4667	0.1321	0.2059	0.6073	0.7007
Arbre de décision	0.1926	0.2430	0.2149	0.5600	0.8830
Forêt Aléatoire	0.2815	0.1617	0.2054	0.6358	0.8190
Boosting	0.1634	0.1852	0.1736	0.6047	0.8534

Les modèles linéaires comme LDA et logistique offrent un bon compromis entre précision, F-score et AUC, mais restent limités en sensibilité. Parmi les approches non linéaires, l'arbre de décision se distingue avec les meilleures performances, notamment en termes de F-score et de sensibilité. Bien qu'il n'offre pas des résultats exceptionnels, l'arbre de décision reste le modèle le plus efficace pour prédire les candidats rappelés, avec la meilleure accuracy.

6 Optimisation de l'Arbre de décision

6.1 Ajustement du seuil de classification

L'objectif du modèle est de prédire si un candidat sera rappelé à partir des informations contenues dans son CV.

Ce problème de classification binaire nécessite un compromis entre deux objectifs importants :

- **Maximiser la sensibilité (Recall)** : détecter un maximum de candidats qui seront réellement rappelés.
- **Maximiser la précision (Precision)** : éviter de prédire à tort qu'un candidat sera rappelé, afin de limiter les coûts associés à un traitement inutile de candidatures non retenues.

Pour atteindre cet équilibre, une méthode efficace consiste à ajuster le seuil de classification.

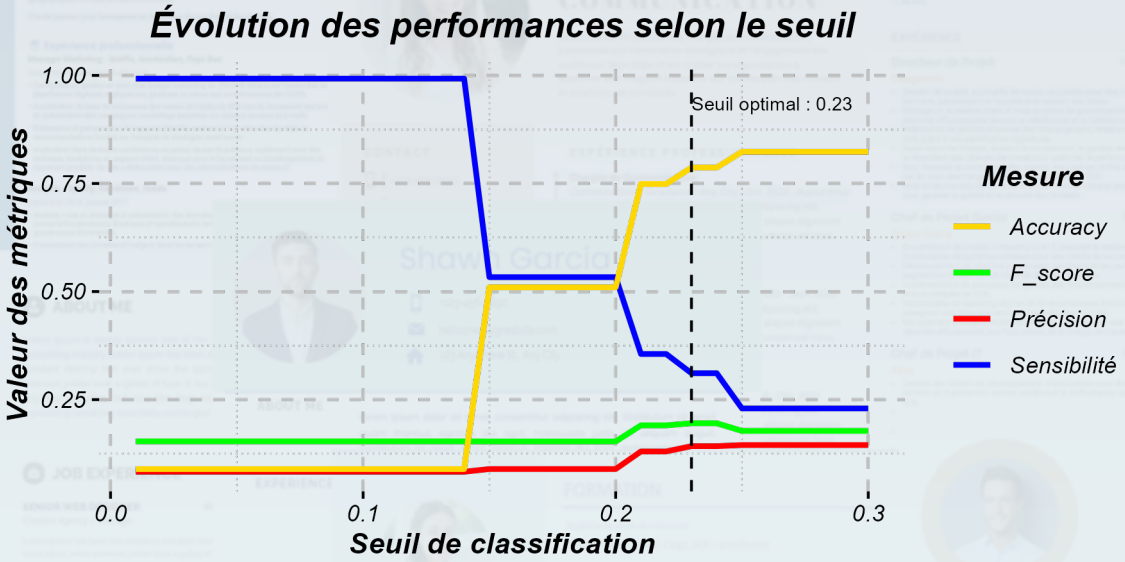
En pratique, le modèle assigne à chaque candidat une probabilité prédite d'être rappelé. Par défaut, un candidat est classé comme "rappelé" si cette probabilité dépasse un certain seuil θ. Toutefois, ce seuil peut être ajusté pour mieux répondre aux priorités opérationnelles :

- Un seuil plus bas augmente la **sensibilité**, mais peut faire baisser la **précision**.
- Un seuil plus élevé améliore la **précision**, mais peut diminuer la **sensibilité**.

La règle de décision devient donc :

$$\text{Si } P(\text{call} = 1 \mid X_i = x) > \text{seuil optimisé, alors le candidat est prédit comme rappelé.}$$

L'ajustement du seuil permet ainsi de contrôler finement le comportement du modèle, en fonction du compromis souhaité entre rappel et précision, en lien direct avec les enjeux du recruteur.



6.2 Seuil de classification optimal

Table 32: Meilleurs seuils de classification (Arbre de décision)

Précision	Sensibilité	F_score	Accuracy	threshold	Moyenne
0.1424	0.3111	0.1953	0.7869	0.23	0.3589
0.1424	0.3111	0.1953	0.7869	0.24	0.3589
0.1301	0.3556	0.1905	0.7488	0.21	0.3562
0.1301	0.3556	0.1905	0.7488	0.22	0.3562
0.1449	0.2296	0.1777	0.8233	0.25	0.3439

Le seuil de 0.23 et 0.24 semblent être les meilleurs choix qui combine le mieux les performances que nous souhaitons optimiser, car ils offrent les meilleures valeurs de moyenne (0.3589), avec un bon compromis entre précision, sensibilité, F-score et accuracy. Ces seuils optimisent globalement les performances tout en maintenant une bonne sensibilité et un F-score raisonnable.

7 Conclusion

Parmi tous les modèles testés, l'arbre de décision s'est distingué comme étant le plus performant pour prédire les candidats susceptibles d'être rappelés après la soumission de leur CV à une offre d'emploi. Ce modèle offre la meilleure sensibilité (0,2430), le meilleur F1-score (0,2149) et l'accuracy la plus élevée (0,8830), ce qui en fait le choix optimal pour notre étude. Cependant, malgré cette performance relative, il est important de noter que tous les modèles, y compris l'arbre de décision, présentent des résultats globalement insatisfaisants, ce qui peut être attribué au déséquilibre important des classes entre les candidats rappelés et non rappelés. Ce déséquilibre rend difficile la prédiction fiable de la classe minoritaire (les candidats rappelés), malgré l'optimisation du modèle.

Table 33: Hyperparamètres retenus : Arbre de décision

Coût de complexité	Taux de re-échantillonnage	Nombres de voisins utilisé
1.00e-02	0.3	5

Table 34: Performances du modèle : Arbre de décision

Précision	19.26%
Spécificité	92.81%
Taux de faux positifs	7.19%
Taux de faux négatifs	75.70%
Sensibilité	24.30%
F-score	21.49%
Taux d'erreur test	11.70%
Taux d'erreur train	10.66%
AUC	56.00%
Accuracy	88.30%

[1] Taux de faux positifs : Proportion des candidats prédits comme rappelés alors qu'ils ne l'ont pas été.

[2] Taux de faux négatifs : Proportion des candidats prédits comme non rappelés alors qu'ils l'ont été.

8 Annexes

8.1 Coefficients Obtenus (Logit)

Table 35: Coefficients significatifs

Variables	Coef	Odds Ratios	P-value
(Intercept)	-0.273	0.761	8.5e-20
experience	0.127	1.135	4.9e-04
gender_female	0.123	1.131	4.9e-04
ethnicity_afam	-0.354	0.702	1.1e-31
quality_high	0.198	1.218	3.6e-04
city_chicago	-0.138	0.871	3.5e-04
honors_yes	0.117	1.124	2.1e-04
volunteer_yes	-0.194	0.824	1.8e-04
military_yes	-0.101	0.904	4.6e-03
holes_yes	0.18	1.198	3.0e-06
school_yes	0.084	1.087	3.0e-02
special_yes	0.357	1.43	2.8e-27
college_yes	0.169	1.184	2.8e-07
wanted_supervisor	-0.155	0.857	2.1e-04
wanted_retail.sales	-0.126	0.882	1.6e-03
requirements_yes	-0.139	0.87	2.4e-03
reqexp_yes	0.206	1.229	3.7e-04
reqcomm_yes	-0.114	0.892	9.0e-04
reqeduc_yes	-0.291	0.748	5.5e-16
reqorg_yes	-0.213	0.808	1.4e-08
industry_health.education.social.services	0.149	1.16	3.4e-04

Les résultats obtenues montrent que les principaux facteurs augmentant les chances d'être rappelé par le recruteur sont le nombre d'années d'expérience de travail (experience), être une femme (gender_femmale), avoir un CV de qualité (quality_high), des distinctions (honors_yes), des compétences particulières (special_yes), au minimum un diplôme universitaire (college_yes), avoir eu une péridode d'inactivité (holes_yes), une experience professionnelle pendant les études (school_yes), la correspondance avec les exigences d'expérience du poste (reqexp_yes), ainsi que le fait que l'employeur appartienne aux secteurs de la santé, de l'éducation ou des services sociaux (industry).

En revanche les autres varaibles mentionnés dans le tableau présentant un coefficient négatif comme avoir un prénom à cononance afor-américaine (ethnicity_afam) diminue les chances d'un candidat d'être rappelé.

8.2 Recette des Modèles

LDA

```
lda_recipe <- recipe(call ~ ., data = call_train) %>%
  step_dummy(all_nominal_predictors(), one_hot = TRUE) %>%
  step_zv(all_predictors()) %>%
  step_normalize(all_numeric_predictors()) %>%
  step_smotenc(call, over_ratio = tune(), neighbors = tune()) %>%
  step_corr(all_numeric_predictors(), threshold = 0.9)
```

QDA

```
qda_recipe <- recipe(call ~ ., data = call_train) %>%
  step_smotenc(call, over_ratio = tune(), neighbors = tune()) %>%
  step_dummy(all_nominal_predictors(), one_hot = TRUE) %>%
  step_zv(all_predictors()) %>%
  step_normalize(all_numeric_predictors()) %>%
  step_pca(all_numeric_predictors(), threshold = 0.9)
```

KNN

```
knn_recipe <- recipe(call ~ ., data = call_train) %>%
  step_dummy(all_nominal_predictors(), one_hot = TRUE) %>%
  step_zv(all_predictors()) %>%
  step_normalize(all_numeric_predictors()) %>%
  step_smotenc(call, over_ratio = tune(), neighbors = tune()) %>%
  step_corr(all_numeric_predictors(), threshold = 0.9)
```

SVM Radial

```
svmr_rec <- recipe(call ~ ., data = call_train) |>
  step_center(all_numeric_predictors()) |>
  step_scale(all_numeric_predictors()) |>
  step_smotenc(call, over_ratio = tune(), neighbors = tune()) |>
  step_downsample(call, under_ratio = 1) |>
  step_dummy(all_nominal_predictors())|>
  step_corr(all_numeric_predictors(), threshold = 0.9, method = "spearman")
```

SVM Linéaire

```
svmlin_rec <- recipe(call ~ ., data = call_train) |>
  step_center(all_numeric_predictors()) |>
  step_scale(all_numeric_predictors()) |>
  step_smotenc(call, over_ratio = tune(), neighbors = tune()) |>
  step_dummy(all_nominal_predictors())  |>
  step_corr(all_numeric_predictors(), threshold = 0.9, method = "spearman")
```

Régression Logistique

```
logit_recipe <- recipe(call ~ ., data = call_train) %>%
  step_dummy(all_nominal_predictors(), one_hot = TRUE) %>%
  step_zv(all_predictors()) %>%
  step_smotenc(call, over_ratio = tune(), neighbors = tune()) %>%
  step_normalize(all_numeric_predictors()) %>%
  step_corr(all_numeric_predictors(), threshold = 0.9)
```

Arbre de décsion

```
data_rec <- call_train |>
  recipe(call ~ ., strata = call) |>
  step_center(all_numeric_predictors()) |>
  step_scale(all_numeric_predictors()) |>
  step_smotenc(call, over_ratio = tune(), neighbors =tune() ) |>
  step_corr(all_numeric_predictors(), threshold = 0.9, method = "spearman")
```

Forêt Aléatoire

```
rec <- recipe(call ~ ., data = call_train) |>
  step_center(all_numeric_predictors()) |>
  step_scale(all_numeric_predictors()) |>
  step_smotenc(over_ratio = tune(), skip = TRUE, neighbors = tune()) |>
  step_downsample(call, under_ratio = 2) |>
  step_corr(all_numeric_predictors(), threshold = 0.9, method = "spearman")
```

Boosting

```
boost_rec <- recipe(call ~ ., data = call_train) %>%
  step_center(all_numeric_predictors()) %>%
  step_scale(all_numeric_predictors()) %>%
  step_smotenc(call, over_ratio = tune(), neighbors = tune()) %>%
  step_dummy(all_nominal_predictors())
```