Mémoire de Master 2 MIAGE SITN

(Méthodes Informatique Appliquées à la Gestion des Entreprises)

La détection des démissions en entreprise par apprentissage automatique

Année universitaire : 2022 / 2023

Auteur : BERTHIER Nicolas

Supervisé par : Michel ZAMFIROIU - Tuteur enseignant

Arnaud PLESSIS - Maître d'apprentissage

Remerciements

Je tiens à remercier toutes les personnes qui m'ont aidé pour la rédaction de ce mémoire.

Tout d'abord, j'adresse mes sincères remerciements à mon maître d'apprentissage de Dassault Systèmes où j'effectue mon apprentissage, M. Arnaud PLESSIS, pour sa bienveillance, et sa disponibilité. Il a toujours été disponible pour moi en cas de difficulté, et a toujours su me guider dans mes missions et pour ce mémoire.

Enfin, je tiens à remercier les professeurs intervenus au cours de ma Licence et de mon Master en MIAGE à l'Université Paris Dauphine, pour leurs apprentissages, qui ont été nécessaires au bon déroulement de mon apprentissage et de la rédaction de ce mémoire. Je tiens à remercier en particulier M. Michel ZAMFIROIU, mon tuteur, qui s'est toujours montré disponible malgré ses obligations, tant lors de nos cours que durant le reste de l'année.

Introduction

Le turnover est un phénomène qui touche toutes les entreprises. Il s'agit du renouvellement des effectifs qui est le résultat des départs des employés : départs à la retraite, démissions, et licenciements[1]. Ce phénomène est un problème central pour les entreprises dont l'objectif est de maximiser leurs rendements. En effet, ce renouvellement d'effectifs a un coût en termes de recrutement. Il peut aussi entraîner une perte de compétences et de connaissances importantes, ce qui peut affecter la productivité et la qualité du travail. De plus, il peut dégrader le tissu social de l'entreprise, ce qui peut avoir comme conséquence d'impacter négativement l'harmonie dans les équipes et la collaboration entre les employés[2]. Enfin, le turnover peut aussi impacter la sécurité informatique, les comptes utilisateurs des anciens employés peuvent rester ouverts et offrir aux hackers des portes d'entrées au système d'information des entreprises[3].

Le principal élément du turnover est le départ volontaire des employés plus communément appelé "démission". En France, entre la fin 2021 et le début 2022, il y a eu près de 520 000 démissions par trimestre, soit un peu moins de 3% des salariés[4]. Il s'agit également du principal levier sur lequel l'entreprise peut agir. Les entreprises peuvent mettre au point des stratégies visant à conserver leurs talents : augmentation des salaires, amélioration des conditions de travail, obtention d'opportunités et possibilités d'évolutions de carrière (...)[5]. Lors de mon apprentissage chez Dassault Systèmes, qui est l'un des principaux éditeurs de logiciels en Europe, j'ai pu constater que cette problématique de rétention des talents y est d'autant plus présente, que le secteur de l'informatique est dynamique et très concurrentiel. Ainsi, détecter ces départs volontaires s'avère un enjeu capital pour les entreprises, pour leur permettre de retenir plus efficacement leurs employés

---

[1] Tout savoir sur la notion de turnover en entreprise, Le Figaro, https://recruteur.lefigaro.fr/article/tout-savoir-sur-la-notion-de-turnover-en-entreprise/ (01/05/2023)
[2] Attrition Issues and Retention Challenges of Employees, Brijesh Kishore Goswami, Sushmita Jha, International Journal of Scientific & Engineering Research Volume 3, Issue 4, April-2012
[3] François Desnoyers, Le turnover des salariés pénalise la sécurité informatique, https://www.lemonde.fr/emploi/article/2019/12/11/le-turnover-des-salaries-penalise-la-securite-informatique_6022404_1698637.html
[4] Adrien Lagouge, Ismael Ramajo, Victor Barry, La France vit-elle une "Grande démission" ?, Dares (Direction de l'Animation de la Recherche et des Statistiques), https://dares.travail-emploi.gouv.fr/publication/la-france-vit-elle-une-grande-demission
[5] Turnover Rates: Factors and Reduction Strategies, Indeed, https://www.indeed.com/career-advice/career-development/turnover-rates (01/05/2023)

et de mettre en place des stratégies pro-actives sur les effectifs les plus aptes à partir.

Problématique

Depuis plusieurs années, le champ d'étude de l'intelligence artificielle s'est particulièrement développé en informatique. L'une de ses composantes, l'apprentissage automatique ou machine learning donne la capacité "d'apprendre" aux ordinateurs[6]. Les entreprises possèdent un grand nombre de données dans leurs SI (systèmes d'informations). Les données contenues dans les SIRH (systèmes d'informations des ressources humaines) pourraient être utilisées pour détecter les démissions des employés. Or, en 2018 seules 22% des services des ressources humaines des entreprises nord-américaines - pourtant à l'avant-garde sur les sujets de l'IA et du machine learning - utilisent des outils d'analyse de données[7].

Cet état de fait, nous pousse à investiguer sur les apports du machine learning pour détecter les départs volontaires des employés. Nous pouvons alors nous demander si le machine learning pourrait aider à détecter ces démissions ? Quels sont les modèles de machine learning les plus efficaces pour les détecter ? Et quelle est la méthodologie la plus appropriée à mettre en place pour détecter les démissions ?

Annonce du plan

Dans un premier temps nous verrons quels sont les principaux facteurs de démissions des employés mis en évidence dans les études sur le sujet. Ensuite, nous présenterons de manière plus exhaustive ce qu'est le machine learning. Nous verrons la différence entre l'apprentissage supervisé et non-supervisé, ou encore entre régression et classification. Enfin, nous présenterons les principaux algorithmes d'apprentissage supervisés utilisés pour détecter les départs volontaires d'employés dans la littérature.

---

[6] Apprentissage automatique, Wikipédia, https://fr.wikipedia.org/wiki/Apprentissage_automatique (01/05/2023)

[7] The Rise of Analytics in HR, Linkedin, https://business.linkedin.com/content/dam/me/business/en-us/talent-solutions/talent-intelligence/workforce/pdfs/Final_v2_NAMER_Rise-of-Analytics-Report.pdf

Dans un second temps, nous mettrons en pratique ces algorithmes sur les données fournies par IBM sur Kaggle[8]. Ce sont sur ces données que s'appuie la majorité des articles de la littérature sur le machine learning dans le cadre de la détection des démissions. Nous présenterons brièvement les données et le caractère "déséquilibré" de celles-ci (le fait d'avoir plus d'employés restants que de démissions). Nous mettrons en évidence notre méthodologie (nos choix en termes de nettoyage de données, de sélections de variables, etc..) et notre protocole d'essai.

Enfin nous testerons les différents algorithmes de machine learning de la littérature. Nous chercherons quels sont les meilleurs modèles, tout en testant deux manières de prendre en charge le déséquilibre des données : par le biais de l'hyperparamètres class_weight, et par une méthode de sur-échantillonnage telles que SMOTE (Synthetic Minority Oversampling TEchnique) ou ADASYN (Adaptive Synthetic Sampling), très utilisées dans le cadre de la détection des fraudes. Nous comparerons l'ensemble des résultats obtenus par ces méthodes et algorithmes.

[8] IBM HR Analytics Employee Attrition & Performance, https://www.kaggle.com/datasets/pavansubhasht/ibm-hr-analytics-attrition-dataset

Les départs volontaires des entreprises

1. Un SIRH en essor contenant des informations clés sur les employés

Au cours des dernières décennies, sous l'influence croissante de la montée des technologies de l'information, les entreprises ont commencé à déployer des systèmes d'information dans différentes fonctions et départements. Les ressources humaines sont l'un des départements qui utilisent le plus fréquemment le système d'information et de gestion. Les informations qui transitent par ce système d'informations couvrent un périmètre de plus en plus large et sont de plus en plus nombreuses, et ce, en particulier dans les grandes organisations. Aujourd'hui, le SIRH permet de gérer et d'automatiser les processus clés des ressources humaines. Parmi ces processus clés nous retrouvons des activités telles que[9] :

- L'identification des employés (création d'un matricule individuel pour chaque employé de l'entreprise permettant une pseudonymisation) ;
- La tenue de dossiers complets sur les employés existants (âge, sexe, adresse du domicile, nom, prénom, état matrimonial légal, éléments de rémunération, numéro de sécurité sociale, numéro de compte) ;
- Le suivi de leurs horaires et de leurs congés (nombre d'heures travaillées, horaires d'arrivée et de départ, prise de congés et dates de congé) ;
- Le suivi des fonctions de l'employé dans l'entreprise (nom du métier, secteur d'activité dans l'entreprise, nom du service).

Le système d'information des ressources humaines aide à déterminer les besoins des employés, et ceux de l'organisation (dotation en personnel, budget lié aux rémunérations, relation avec les employés, etc..). Il permet de réaliser les plans d'action et de mettre en place des objectifs stratégiques à long terme pour l'organisation. Ainsi, le SIRH est une mine d'informations sur les employés et il pourrait contenir des informations précieuses permettant d'étudier ou de prédire quels employés ont une plus forte tendance à démissionner. Ces facteurs de

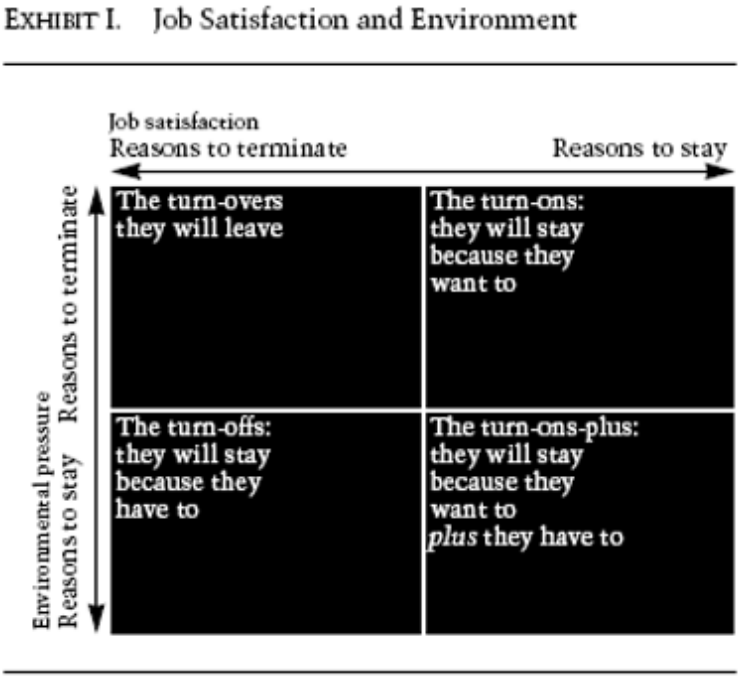
[9] Qu'est-ce qu'un SIRH ?, SAP, https://www.sap.com/france/products/hcm/employee-central-hris/what-is-hris.html

démission ont déjà été étudiés dans la littérature, nous allons maintenant les exposer.

2. Les facteurs de démissions

Plusieurs facteurs, à la fois externes et internes, influencent la volonté des employés à rester dans une entreprise (cf. tableau ci-dessous), notamment les opportunités d'emploi dans le secteur de travail dudit employé. D'autres facteurs externes peuvent être à l'œuvre comme le fait d'avoir des responsabilités familiales, ou encore le fait d'être "trop âgé pour recommencer à zéro"[10]..

Fig. 1 Schéma représentant les facteurs interne et externe aux démissions[11]



Les facteurs internes sont liés à la satisfaction au travail et à l'engagement envers l'organisation. Plusieurs éléments influencent cette satisfaction au travail. Il s'agit par exemple de la sensation d'appartenir à son équipe ou encore à son entreprise[12], de

[10] Vincent S. Flowers and Charles L. Hughes, Why Employees Stay, Employee Retention, Harvard Business Review, https://hbr.org/1973/07/why-employees-stay
[11] Vincent S. Flowers and Charles L. Hughes, Why Employees Stay, Employee Retention, Harvard Business Review, https://hbr.org/1973/07/why-employees-stay
[12] Brooks C. Holtorn, Terence Mitchell, et al., Turnover and Retention Research: A Glance at the Past, a Closer Review of the Present, and a Venture into the Future, The Academy of Management Annals, https://www.researchgate.net/publication/233055591_5Turnover_and_Retention_Research_A_Glance_at_the_Past_a_Closer_Review_of_the_Present_and_a_Venture_into_the_Future

la satisfaction vis-à-vis du niveau de rémunération[13], d'obtenir et/ou de percevoir des opportunités, l'intérêt pour le travail, ou encore d'avoir le sentiment d'avoir réalisé quelque chose[14].

Ainsi, la tension entre facteurs internes et externes peut se schématiser comme sur la figure ci-dessus. Lorsque le niveau de satisfaction au travail et de pression de l'environnement poussent tous deux à partir, l'employé a tendance à démissionner. Dans le cas où seul l'un de ces deux facteurs pousse l'employé à partir, celui-ci aurait tendance à rester. Dans le cas où la pression de l'environnement le pousse à rester, l'employé resterait parce qu'il le doit, tandis que lorsque c'est la satisfaction au travail qui le pousse à rester, celui-ci resterait parce qu'il le veut.

Ces facteurs mis en évidence dans la littérature sont autant d'indices sur les variables, sur les informations qui peuvent être intéressantes à exploiter dans le cadre de notre travail sur la détection automatisée des démissions. Nous pourrions par exemple exploiter des informations comme la situation matrimoniale pour détecter des effets "d'obligation familiale". Néanmoins, certaines autres informations pourraient être absentes de notre base de données et donc du système d'information, ce qui invite les entreprises cherchant à prédire les démissions de leurs employés, à créer/ajouter ces informations à leur système d'informations. Nous allons à présent nous intéresser à la méthode que nous allons utiliser - l'apprentissage automatique - pour tenter de prédire les démissions des employés.

[13] Vermandere C., Impact of salary on job satisfaction, Eurofound, https://www.eurofound.europa.eu/publications/article/2013/impact-of-salary-on-job-satisfaction

[14] Aziri Brikend, Job Satisfaction : A Litterature review, https://mrp.ase.ro/no34/f7.pdf

Le machine learning appliqué aux démissions

1. Généralités sur l'apprentissage automatique

1.1. Historique de l'apprentissage automatique

Historiquement, le terme "machine learning" a été utilisé pour la première fois par Arthur Samuel, un employé d'IBM travaillant dans le domaine de l'intelligence artificielle[15]. Il s'agit d'un ensemble de méthodes permettant aux ordinateurs d'apprendre à partir de données qui leurs sont transmises. Ce champ d'étude a connu un premier essor autour des années 60, notamment avec la création du premier programme capable de jouer au Jeu de dames en 1952[16]. Puis un second essor, plus récent, s'est produit après les années 1990-2000 avec l'augmentation des capacités de calcul des ordinateurs et l'accès aux données qui sont de plus en plus nombreuses[17], grâce notamment à l'avènement d'internet et du Big Data.
Aujourd'hui, l'apprentissage automatique est utilisé dans de nombreux domaines, tels que le secteur industriel (maintenance prédictive des équipements), le commerce (déterminer les produits les plus intéressants pour un client donné), la santé (diagnostics de maladie), le tourisme (tarification en temps réel), les services financiers (analyse et régulation des risques)[18], ou encore les ressources humaines tel que nous le démontrerons au cours de cette étude.

1.2. Objectifs de l'apprentissage automatique

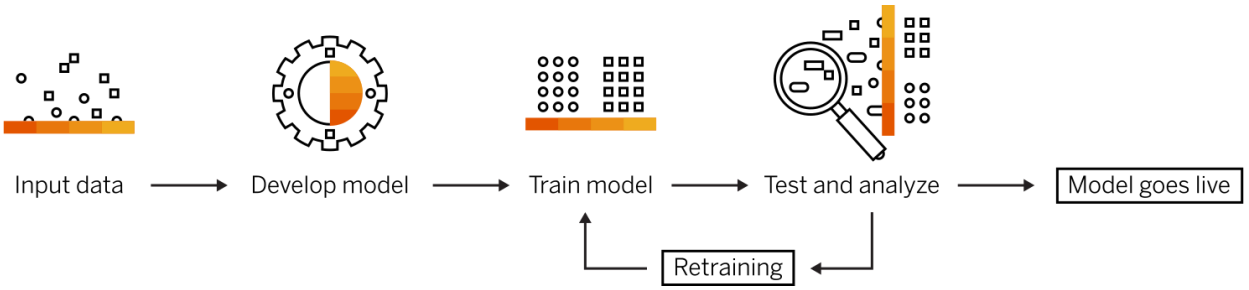
L'apprentissage automatique a pour objectif d'extraire et d'exploiter de manière automatique les informations d'un jeu de données. Il se concentre sur les résultats, sur la prédiction de valeurs futures (*apprentissage supervisé*), et la capacité de

---

[15] Arthur Samuel (computer scientist), Wikipedia, https://en.wikipedia.org/wiki/Arthur_Samuel_(computer_scientist) (05/08/2023)
[16] Timeline of machine learning, Wikipedia, https://en.wikipedia.org/wiki/Timeline_of_machine_learning (28/07/2023)
[17] Quels facteurs ont permis l'essor de l'intelligence artificielle ?, Musée Informatique, https://www.museeinformatique.fr/facteurs-essor-intelligence-artificielle/
[18] Qu'est-ce que le machine learning, NetApp, https://www.netapp.com/fr/artificial-intelligence/what-is-machine-learning/#:~:text=Le%20machine%20learning%2C%20sp%C3%A9cialit%C3%A9%20de,de%20d%C3%A9cision%20sans%20interaction%20humaine.

regrouper des observations en groupes homogènes (*apprentissage non-supervisé*), pour permettre de prendre des décisions à partir d'un grand nombre de données.

Fig. 2 Schéma explicatif du machine learning[19]



L'apprentissage automatique se traduit par une phase d'apprentissage durant laquelle le modèle est directement en contact avec les données et s'adapte à elles. Dans un second temps, l'algorithme est testé, il lui est fourni de nouvelles données qu'il ne connaît pas, et dont il doit prédire les valeurs. Cette phase permet de vérifier si l'apprentissage est correct, et qu'il prédit des valeurs exactes.

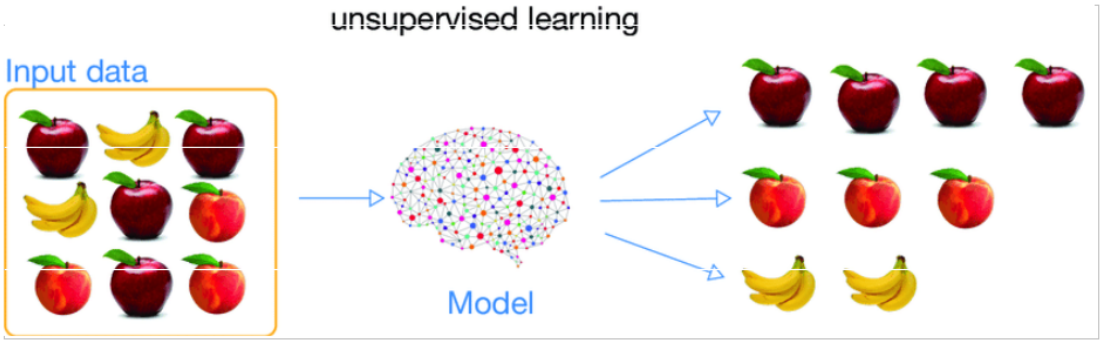
1.2.1. L'apprentissage non-supervisé

L'apprentissage non-supervisé permet à l'ordinateur d'apprendre à partir d'un ensemble de données non étiquetées fournies au début de l'apprentissage. L'objectif de cet apprentissage est de regrouper les observations $(x_1, x_2, ..., x_n)$ en groupes homogènes. Cela permet de découvrir des motifs ou des structures dans les données qui peuvent être utiles pour la prise de décision ou la compréhension des données[20].

Dans l'exemple ci-dessous, des données sont fournies à la machine, sans qu'elle ne sache de quoi il s'agit. Par exemple, il pourrait être fourni les dimensions de chacun des objets du groupe "Input data". La machine, grâce à l'apprentissage non-supervisé, crée des groupes homogènes. Elle met les fruits de la même famille dans les mêmes groupes, sans avoir connaissance qu'il s'agisse de types de fruits ni même de fruits, seulement grâce aux caractéristiques communes de ces éléments trouvées dans les données d'apprentissage (ici par exemple, les dimensions des objets).

[19] Quel est le rapport entre le Machine Learning et l'IA ?, SAP, https://www.sap.com/canada-fr/products/artificial-intelligence/what-is-machine-learning.html
[20] What is supervised learning?, IBM, https://www.ibm.com/topics/supervised-learning (02/05/2023)

Fig. 3 Schéma représentant l'utilisation de l'apprentissage non-supervisé[21]



Ainsi, cette méthode d'apprentissage regroupe un certain nombre d'algorithmes qui ont pour but de faire apparaître de l'information qui n'était pas là au départ. Dans le cadre de notre modélisation des démissions, ce type d'apprentissage ne sera pas utilisé car les données dont nous disposons sont étiquetées. Pour chaque employé nous avons dans nos données l'information concernant sa démission ("Oui/Non").

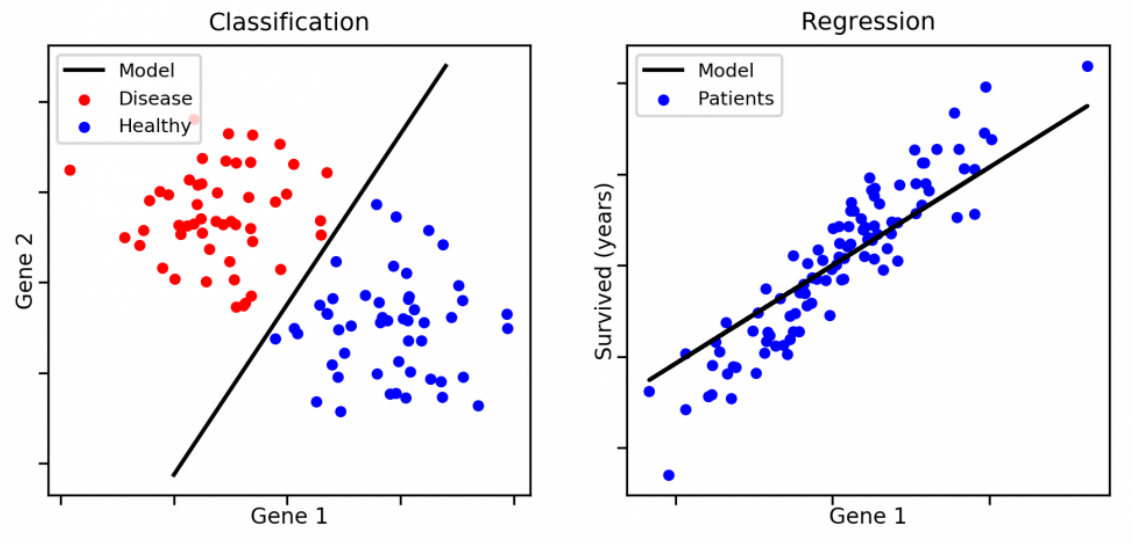
1.2.2. L'apprentissage supervisé

L'apprentissage supervisé utilise un ensemble de données (X) avec n observations $\{x_1, x_2, \ldots, x_n\}$ pour lesquels nous connaissons déjà le résultat attendu (Y, l'étiquette ou label), sans connaître la fonction $f()$ qui permet d'obtenir l'image Y à partir de X. L'objectif de cette méthode est d'approximer la fonction f telle que $f : X \rightarrow Y$ telle que $f(x) \simeq y$.

Ce type d'apprentissage permet de répondre à deux principaux problèmes : les problèmes de classification (où $Y = \{0, 1\}$) et les problèmes de régression (où $Y = \mathbb{R}$). La régression permet de prédire une variable continue qui peut donc prendre n'importe quelle valeur (cf. image de gauche de la figure ci-dessous). Quant à la classification, elle consiste à identifier la classe de nouveaux objets à partir d'exemples antérieurs. La variable de sortie est donc une catégorie telle que "malade" ou "non malade" (cf. image de gauche de la figure ci-dessous).

[21] MACHINE LEARNING VS HUMAN DECISION MAKING (SIMILARITÉS ET DIFFÉRENCES), Je veux être DataScientist, https://www.jeveuxetredatascientist.fr/machine-learning-vs-human-decision-making-similarites-et-differences/ (10/07/2023)

Fig. 4 Représentation graphique de données labellisées dans le cadre d'un problème de classification et de régression[22]



Dans l'exemple de droite, les axes (X et Y) représentent un gène (gène 1, X) et la durée de survie (Y). L'objectif de la régression va être de prédire l'espérance de vie en fonction d'une valeur de gène donnée. Cette prédiction est ici représentée par une droite. Ainsi, pour toute nouvelle valeur (X) de gène, il est possible de retrouver une espérance de vie (Y) correspondante sur la droite.

Dans l'exemple de gauche, les axes (X1 et X2) représentent des gènes (gène 1 et 2). La couleur des points quant à elle représente le fait d'être malade (Y=rouge) ou en bonne santé (Y=bleu). L'objectif de la classification va être de séparer ces deux groupes (dans cet exemple à travers une droite) de manière à pouvoir dans le futur classer un nouveau point. Ainsi, toute nouvelle valeur (X1, X2) apparaissant à la gauche de cette droite sera prédite comme en bonne santé et réciproquement.

Donc, l'apprentissage supervisé permet d'entraîner la machine à partir de données labellisées et de prédire le label des nouvelles données qui lui seront fournies. Les jeux de données utilisés dans la détection des démissions sont labellisés tel que pour chaque occurrence, soit chaque employé, un label indique si l'employé a

[22] Zakariyaa ISMAILI, Aprentissage supervisé vs Non supervisé, https://brightcape.co/apprentissage-supervise-vs-non-supervise/#:~:text=L'objectif%20de%20l'apprentissage%20non%20supervis%C3%A9%20est%20de%20mod%C3%A9liser,r%C3%A9ponse%20correcte%20ni%20d'enseignant.

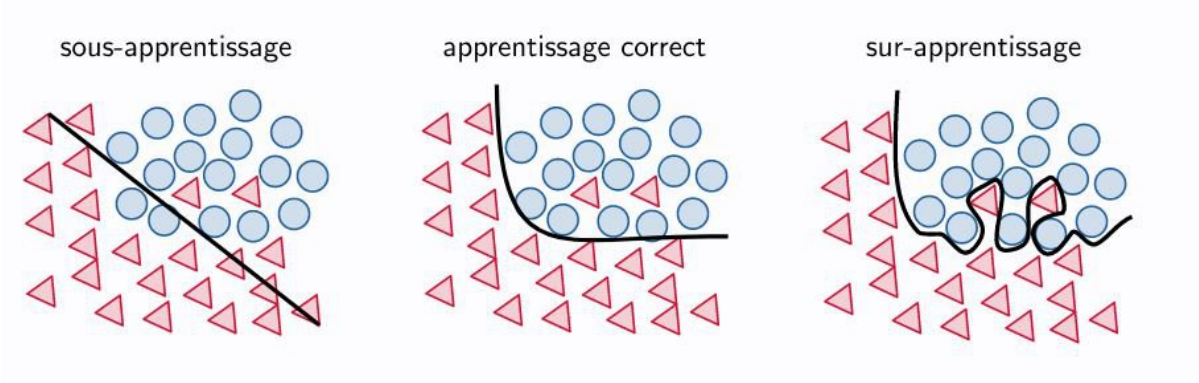
démissionné ou non[23]. Ainsi, la détection des démissions a pour but de classer en deux groupes distincts les employés, il s'agit d'un problème de classification.

1.3. Les problèmes de sous-apprentissage et de sur-apprentissage

L'entraînement de la machine ne se limite pas à la mise en place d'un modèle ou d'un algorithme pour obtenir des résultats. Les algorithmes d'apprentissage automatique ont des hyperparamètres dont le réglage est laissé à la discrétion du data scientist. Ces hyperparamètres "*[sont des] paramètres dont la valeur est utilisée pour contrôler le processus d'apprentissage*"[24]. Ce paramétrage est donc primordial pour éviter deux principaux types d'erreur en machine learning :

- Le sur-apprentissage, il s'agit d'une situation où la *"procédure d'apprentissage qui produit un modèle qui fait de bonnes prédictions sur les données utilisées pour le construire, mais généralise mal*"[25]. En d'autres termes, le modèle entraîné a appris "par coeur". Il est capable de prédire à la perfection des valeurs à partir des données avec lesquelles il s'est entraîné, mais n'est pas capable de prédire de nouvelles valeurs à partir de données qu'il n'a jamais rencontrées auparavant (cf. illustration ci-dessous).

Fig. 5 Schéma illustrant des cas de sous-apprentissage et de sur-apprentissage[26]

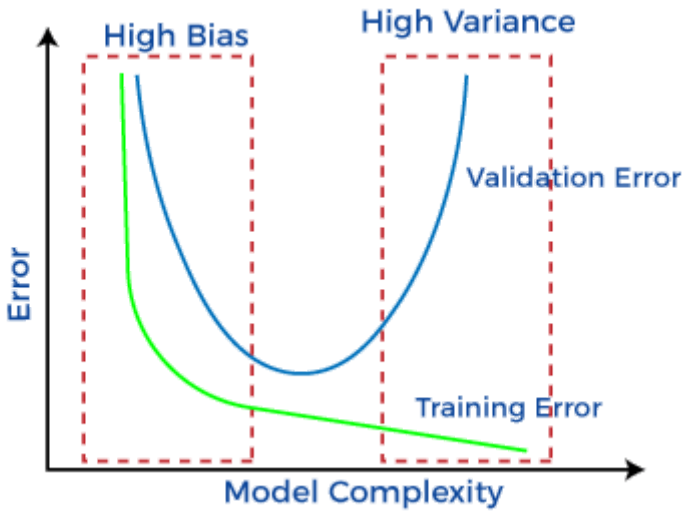


[23] Prediction of Employee Turnover in Organizations using Machine Learning Algorithms - A case for Extreme Gradient Boosting, Rohit Punnoose, Pankaj Ajit, International Journal of Advanced Research in Artificial Intelligence, Vol. 5, No. 9, 2016
https://pdfs.semanticscholar.org/fa49/19810eaee67e851ad13775b78c94217a7908.pdf
[24] Hyperparamètre, Wikipedia, https://fr.wikipedia.org/wiki/Hyperparam%C3%A8tre (05/07/2023)
[25] Chloé-Agathe Azencott, Introduction au Machine Learning,
https://cazencott.info/dotclear/public/lectures/IntroML_Azencott.pdf
[26] Olivier Serpe, Le Coeur de l'intelligence artificielle : l'apprentissage automatique (part 1), EchoSciences Auvergne,
https://www.echosciences-auvergne.fr/articles/le-coeur-de-l-intelligence-articielle-l-apprentissage-automatique-part1

- Le sous-apprentissage signifie que le modèle est trop simple et ne correspond pas bien aux données. Par conséquent, le modèle apprend mal sur les données d'apprentissage et ne peut pas modéliser le phénomène, ce qui empêche la prédiction correcte de valeurs à partir de données fournies à la machine.

Lors de l'entraînement de la machine il est important de trouver un bon compromis "biais-variance". Ce compromis concerne toutes les méthodes algorithmiques d'apprentissage automatique. Le biais est la différence entre la prédiction moyenne du modèle et la valeur correcte qu'il essaie de prédire. Un modèle fortement biaisé accorde peu d'attention aux données d'apprentissage et simplifie à l'extrême le modèle. Cela correspond à un sous-apprentissage et se traduit par des erreurs constamment élevées, que ce soit au niveau de la phase d'apprentissage de la machine ou avec de nouvelles données à prédire.

Fig. 6 Graphique illustrant le compromis biais-variance[27]



Quant à la variance, elle reflète l'erreur due à la sensibilité de l'ensemble d'apprentissage aux petites variations. Une variance élevée peut être causée par des algorithmes qui modélisent un bruit aléatoire dans les données d'apprentissage comme illustré sur le schéma représentant un sur-apprentissage.

[27] Bias and Variance in Machine Learning, Java T point, https://www.javatpoint.com/bias-and-variance-in-machine-learning

Ainsi, les principales erreurs évitables en apprentissage automatique sont le sur-apprentissage et le sous-apprentissage. Cette problématique relève d'un compromis biais-variance. Plus le modèle est complexe, plus il s'accorde aux données d'apprentissage et perd en généralisation, sa variance est élevée (cf. schéma précédent). Moins un modèle est complexe, moins il apprend des données d'apprentissage et commet de nombreuses erreurs (biais élevé).
Ce dosage concerne tous les algorithmes présentés ci-après, en particulier lorsque les données sont déséquilibrées (lorsqu'une classe a un nombre d'observations bien plus important qu'une autre), cela entraîne fréquemment une situation de sur-apprentissage[28]. Nous allons maintenant présenter les algorithmes d'apprentissage les plus souvent employés dans la littérature pour répondre à notre problématique.

2. Les algorithmes utilisés pour prédire les démissions

Le machine learning a une grande diversité d'algorithmes et de modèles qui peuvent résoudre des problèmes de classification, ce qui en fait sa richesse. En se reportant au travail de revue de littérature de l'article de Norsuhada Mansor et al. "Machine Learning for Predicting Employee Attrition" paru dans l'International Journal of Advanced Computer Science and Applications (IJACSA), les principaux algorithmes testés dans le cadre de la détection des démissions sont : le Support Vector Machine (SVM), le Decision Tree (DT), la Régression Logistique (LR) et le Random Forest (RF)[29]. Nous nous focaliserons sur ces algorithmes prédictifs dans notre recherche.

2.1. Les arbres de décision

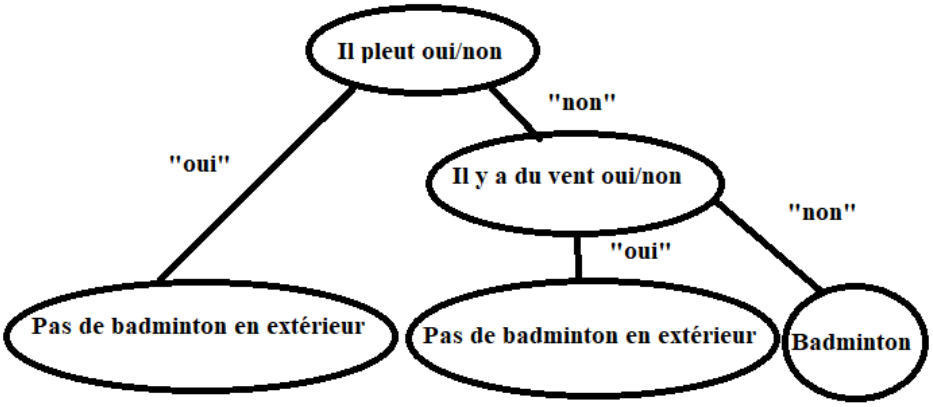
Les arbres de décision sont un algorithme très simple de classification. Ils peuvent être représentés sous forme de graphe où chaque nœud représente une combinaison de variables et de conditions, et où les feuilles représentent le label prédit (cf. exemple ci-dessous).

---

[28] Taha Yassine Ben Ali, Données déséquilibrées, que faire ?, https://blog.octo.com/donnees-desequilibrees-que-faire/
[29] Norsuhada Mansor et al., Journal of Advanced Computer Science and Applications (IJACSA), Machine Learning for Predicting Employee Attrition, https://www.researchgate.net/publication/356809874_Machine_Learning_for_Predicting_Employee_Attrition

Imaginons que nous cherchons à prédire le fait de jouer au badminton ou non en extérieur. Nos données auraient les variables “vent” (oui/non), “pluie” (oui/non), et les labels “badminton en extérieur” (oui/non).

Fig. 7 Graphe représentant un arbre de décision



À partir de nos données nous pourrions créer un arbre de décision comme ci-dessus, chaque nœud/variable séparerait les observations en deux (par exemple). Ainsi, lorsque nous cherchons à prédire le label d’une observation, il suffit de suivre le “bon chemin” en fonction des conditions, ce qui mène vers une feuille, c’est-à-dire la valeur prédite.

Néanmoins cette méthode présente des inconvénients. Elle est faiblement généralisable, et prédit assez mal les valeurs des nouvelles données fournies à l’algorithme. L’algorithme de Random Forest résout ce problème de généralisation.

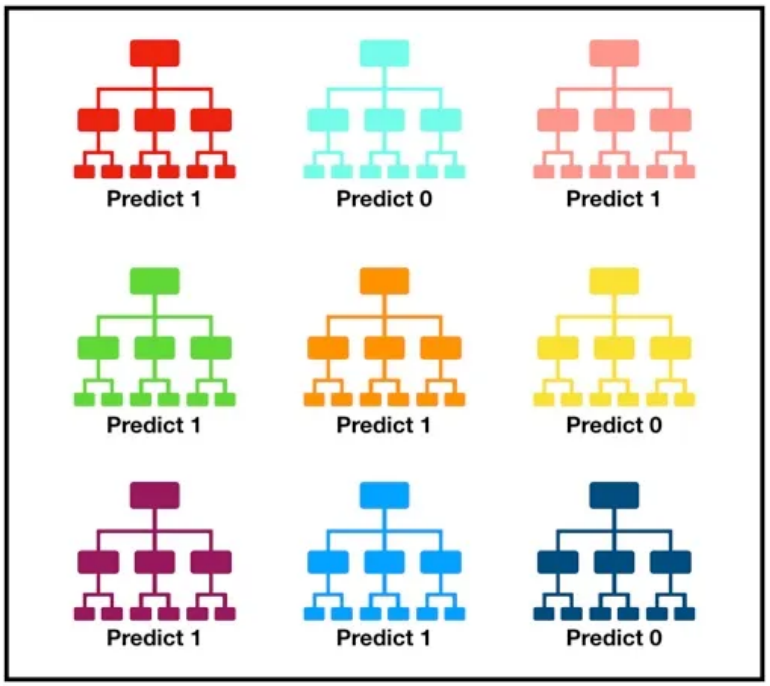
2.2. Random Forest Classifier

Le Random Forest Classifier s’appuie sur le Bagging. Le Bagging est *“un méta-algorithme faisant partie des méthodes ensemblistes : partant d’un algorithme de Machine Learning, il utilise de multiples fois cet algorithme pour obtenir un résultat plus fiable”*[30]. Ainsi, le Random Forest Classifier *(“forêt aléatoire” en français)* s’appuie sur la création de plusieurs arbres de décision, construits sur des parties aléatoirement sélectionnées des données et des variables d’entraînement, pour permettre une meilleure généralisation. Chaque arbre de la forêt fournit une

[30] Le Bagging en Machine learning, de quoi s’agit-il ?, Kobia, https://kobia.fr/le-bagging-en-machine-learning-de-quoi-sagit-il/

prédiction de classe, et la classe qui reçoit le plus de votes devient la prédiction du modèle (voir la figure ci-dessous).

Fig. 8 Schéma du principe de l'algorithme Random Forest[31]



Dans l'exemple ci-dessus, une majorité d'arbres de décision (⅔ des arbres) prédisent à partir d'une donnée fournie la valeur "1". La prédiction de l'algorithme sera donc "1", que nous pourrions imaginer par exemple correspondre au fait de démissionner (et inversement pour la valeur "0").

2.3. La régression logistique

Quant à la régression logistique, il s'agit d'un modèle issu des statistiques et utilisé en machine learning. Elle est utilisée pour modéliser la relation entre une variable binaire (à deux classes) dépendante (variable cible) et un ensemble de variables indépendantes (variables prédictives)[32]. Bien que le nom puisse prêter à confusion,

[31] Tony Yiu, Understanding Random Forest, Towards Data Science, https://towardsdatascience.com/understanding-random-forest-58381e0602d2
[32] Qu'est-ce que la régression logistique ?, IBM, https://www.ibm.com/fr-fr/topics/logistic-regression

la régression logistique est principalement utilisée pour la classification plutôt que pour la régression.

La régression logistique modélise les données grâce à la fonction sigmoïde qui transforme une valeur continue en une valeur comprise entre 0 et 1. Cette fonction est définie par : $\delta(x) = \frac{1}{1+e^{-x}}$.

Le modèle de régression logistique prédit la probabilité que la variable cible appartienne à une classe particulière. La relation entre les variables indépendantes X_i et la probabilité d'obtenir une classe (Y=0 ou Y=1) est modélisée en utilisant la fonction logistique, dérivée de la fonction sigmoïde. Sa forme générale est la suivante : $P(Y = 1 \mid X) = \frac{1}{1+e^{-(\beta_0+B_1x_1+\ldots+\beta_p x_p)}}$[33]. $P(Y = 1 \mid X)$ est la probabilité d'appartenir à la classe "1" ("avoir démissionné" par exemple) sachant les valeurs des variables $x_1, x_2, \ldots, x_p$, soit les données fournies à l'algorithme. Les coefficients β_0, β_1, $\beta_p$doivent être recherchés par la machine à travers un entraînement et sont ajustés de manière à maximiser la cohérence entre les valeurs prédites et les valeurs attendues.

Ainsi, une fois le modèle entraîné, il devient capable de prédire les probabilités d'appartenance à une classe pour de nouvelles observations et donc les classer.

2.4. Support Vector Machine

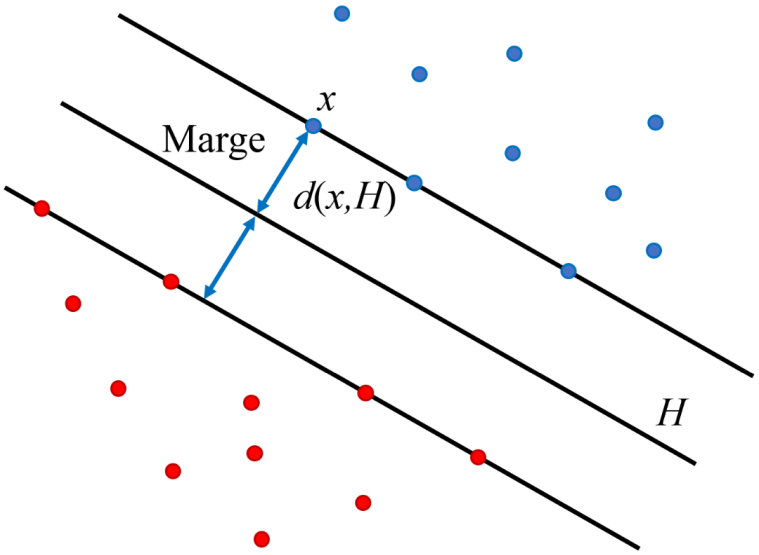
L'algorithme de Support Vector Machine peut être utilisé dans le cadre de la classification et se base sur une approche géométrique du problème. Le but est de trouver l'hyperplan qui maximise la distance entre les deux groupes. Pour un problème de classification binaire comme les démissions (exemple : démissionner = 1, ne pas démissionner = 0), l'hyperplan optimal est celui qui maximise la marge (la distance entre l'hyperplan et les éléments les plus proches pour chaque classe, cf. figure ci-dessous)[34]. Ce choix d'hyperplan permet une meilleure généralisation aux nouvelles données.

---

[33] Régression Logistique, Rappel Théorique, SPSS; https://spss.espaceweb.usherbrooke.ca/regression-logistique/

[34] Rushikesh Pupale, R. Support Vector Machines(SVM) — An Overview. Towards data science, https://towardsdatascience.com/https-medium-com-pupalerushikesh-svm-f4b42800e989 Acquired: 2020-04-28

Fig. 9 Schéma représentant l'hyperplan et ses marges - SVM[35]



Dans cet exemple, les points bleus et rouges représentent réciproquement une classe (démissionner ou non par exemple). L'hyperplan choisi par l'algorithme SVM est représenté par la droite centrale. Cet algorithme est particulièrement performant sur des données d'apprentissage de petite et de moyenne taille[36].

Ainsi, ces différents algorithmes d'apprentissage automatique peuvent résoudre des problèmes de classification telle que la détection des démissions volontaires. Toutefois, tous ces algorithmes n'ont pas nécessairement les mêmes performances et peuvent faire de fausses prédictions.
Ces performances sont également étroitement liées au preprocessing (nettoyage des données), aux choix méthodologiques, et aux sélections d'hyperparamètres. Nous allons dans la suite de notre travail, observer nos données, proposer une méthodologie d'entraînement, entraîner ces différents algorithmes à travers plusieurs méthodes de prises en compte du déséquilibre des données. Enfin nous allons tester ces méthodes de prises en charge et ces algorithmes pour constater quelles sont les méthodes les plus efficaces.

[35] CNAM - UE RCP209, Cours SVM linéaires, CNAM, https://cedric.cnam.fr/vertigo/cours/ml2/coursSVMLineaires.html
[36] Géron, A.: Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems (2017)

La prédiction des départs volontaires

1. Exploration des données

1.1. Découverte des données

L'exploration des données est une phase importante de l'apprentissage automatique. Elle permet de comprendre ce que représentent les données, mais aussi de connaître leur qualité, pour permettre un nettoyage le cas échéant, avant tout entraînement de la machine. En effet, une mauvaise sélection de données, ou des données de mauvaise qualité pourraient perturber l'apprentissage de la machine, ce qui rendrait ses prédictions beaucoup moins précises.

Durant notre étude, nous utiliserons des données fournies sur Kaggle par IBM concernant les départs volontaires d'employés. Ce sont sur ces données que se basent la plus grande partie des travaux de recherche en machine learning sur le sujet. Le jeu de données d'IBM est de taille modeste. Il contient 1470 lignes et 34 colonnes dont 24 colonnes contenant des informations numériques et 10 autres des informations qualitatives (sous forme de chaîne de caractères).

Tab. 1 Liste des variables numériques du jeu de données IBM-attrition

Nom de variable	type	moyenne	écart-type	min	25%	50%	75%	max
Age	Int64	36,92	9,12	18,00	30,00	36,00	43,00	60,00
DailyRate	Int64	802,2	403,2	102,0	465,0	802,0	1157,0	1499,0
Education	Int64	2,9	1,0	1,0	2,0	3,0	4,0	5,0
EmployeeCount	int64	1,0	0,0	1,0	1,0	1,0	1,0	1,0
EnvironmentSatisfaction	Int64	2,7	1,1	1,0	2,0	3,0	4,0	4,0
JobInvolvement	Int64	2,7	0,7	1,0	2,0	3,0	3,0	4,0
JobSatisfaction	Int64	2,7	1,1	1,0	2,0	3,0	4,0	4,0
JobLevel	Int64	2,1	1,1	1,0	1,0	2,0	3,0	5,0
MonthlyRate	Int64	14302,6	7100,3	2094,0	8053,0	14222,0	20460,0	26999,0
MonthlyIncome	Int64	6507,4	4706,4	1009,0	2911,0	4936,0	8380,0	19999,0
NumCompaniesWorked	Int64	2,7	2,5	0,0	1,0	2,0	4,0	9,0
PerformanceRating	Int64	3,2	0,4	3,0	3,0	3,0	3,0	4,0
RelationshipSatisfaction	Int64	2,7	1,1	1,0	2,0	3,0	4,0	4,0
StandardHours	Int64	80,0	0,0	80,0	80,0	80,0	80,0	80,0
StockOptionLevel	Int64	0,8	0,9	0,0	0,0	1,0	1,0	3,0
TotalWorkingYears	Int64	11,3	7,8	0,0	6,0	10,0	15,0	40,0
TrainingTimesLastYear	Int64	2,8	1,3	0,0	2,0	3,0	3,0	6,0
WorkLifeBalance	Int64	2,8	0,7	1,0	2,0	3,0	3,0	4,0
YearsAtCompany	Int64	7,0	6,1	0,0	3,0	5,0	10,0	40,0
YearsInCurrentRole	Int64	4,2	3,6	0,0	2,0	3,0	7,0	18,0
YearsSinceLastPromotion	Int64	2,2	3,2	0,0	0,0	1,0	3,0	15,0
YearsWithCurrManager	Int64	4,1	3,6	0,0	2,0	3,0	7,0	17,0
DistanceFromHome	float64	9,2	8,1	1,0	2,0	7,0	14,0	29,0

A noter que les variables JobInvolvement, JobSatisfaction, JobLevel, StockOptionLevel, WorkLifeBalance, EnvironmentSatisfaction, et RelationshipSatisfaction, bien que numérique, ne sont pas des variables continues, il s'agit de variables ordinales. Elles représentent pour la plupart des niveaux de satisfaction qui peuvent être classés du "plus petit au plus grand", compris entre un minimum et un maximum (par exemple de 1 à 4 par exemple pour Jobsatisfaction).

Tab. 2 Liste des variables catégorielles du jeu de données IBM-attrition

Nom de variable	type	mode	nombre	fréquence
Attrition	string[python]	No	1233	83,9
BusinessTravel	string[python]	Travel_Rarely	1043	71,0
Department	string[python]	Research & Development	961	65,4
EducationField	string[python]	Life Sciences	606	41,2
EmployeeNumber	string[python]	1470	1	0,1
Gender	string[python]	Male	882	60,0
JobRole	string[python]	Sales Executive	326	22,2
MaritalStatus	string[python]	Married	673	45,8
Over18	string[python]	Y	1470	100,0
OverTime	string[python]	No	1054	71,7

Le jeu de données contient aussi des informations sur le niveau de rémunération (MonthlyIncome), sur la situation personnelle (MaritalStatus), sur le niveau d'éducation, ou encore la situation professionnelle de l'employé (département, rôle, ancienneté). Ainsi, nous retrouvons une grande partie des informations liées aux facteurs de démissions mis en évidence dans la littérature, ce qui semble être une bonne chose pour l'entraînement de la machine.

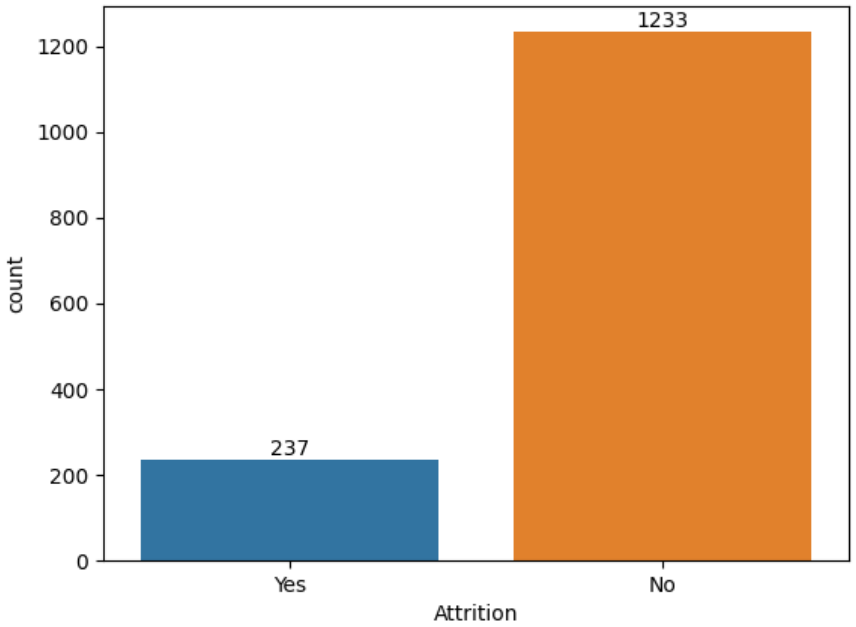
1.2. Etude de la donnée clef : un déséquilibre de données

La variable "Attrition" contient l'information sur le départ volontaire de l'entreprise, il s'agit du label (Y) que nous chercherons à prédire, tandis que toutes les autres variables (X_i) sont de potentielles variables explicatives qui pourront nous aider à prédire ce label.

Avant de mesurer la proportion d'employés qui ont démissionné, nous avons vérifié que chaque ligne de notre jeu de données représente un employé et qu'un employé est bien présent qu'une seule fois. La variable EmployeeNumber est un identifiant

unique pour chaque employé. Elle nous permet de détecter des employés présents plusieurs fois dans les données et nous confirme que chaque employé n'apparaît qu'une seule fois. Nous avons également cherché la présence de données manquantes ou encore de données dupliquées, qui auraient pu entraîner de nombreux biais d'analyse, mais aucune donnée manquante ou dupliquée n'a été trouvée.

Fig. 10 Diagramme représentant les démissions vs "les autres"



Ainsi, dans le jeu de données, les démissions sont minoritaires en comparaison aux effectifs, comme dans la plupart des entreprises. Ce sont 237 employés qui ont démissionné (soit 16,1% du total) contre 1233 restants.

Ces premières informations mettent en évidence plusieurs problématiques. Une première problématique est liée à la taille des données en termes de nombre d'observations. Les algorithmes de machine learning ont besoin d'un grand nombre de données sur lesquelles s'exercer pour pouvoir apprendre correctement. De plus, les données sont déséquilibrées : la classe à détecter, à savoir le départ volontaire d'un employé ($Y = 1$) est largement minoritaire dans les données (<20%). Cela peut entraîner des problématiques de sur-apprentissage comme nous l'avons vu précédemment, mais aussi en termes de mesure de la qualité du modèle.

Notons que ces problématiques sont présentes dans la majorité des jeux de données sur la détection des démissions. En effet, les effectifs d'employés démissionnant sont le plus souvent largement minoritaires au reste de l'effectif des entreprises, et les tailles des jeux de données sont limitées à la taille de l'effectif des entreprises. Cette problématique est ainsi particulièrement prégnante dans les entreprises de taille modérée, et l'apprentissage automatique est impossible à mettre en œuvre dans des entreprises de trop petite dimension. Ainsi, nous soulignons à nouveau la nécessité de répondre à ces problématiques de déséquilibre dans les données et de tailles de données, et ce, à travers une méthodologie adaptée. Nous allons maintenant présenter la méthodologie que nous avons employée durant notre étude.

2. Méthodologie sélectionnée

2.1. Bibliothèques utilisées

Le travail de détection des démissions grâce au machine learning que nous avons mis en place s'appuie sur le langage Python, et plus particulièrement sur plusieurs de ses bibliothèques.

Nous allons nous appuyer sur la bibliothèque Pandas pour la manipulation des données (ajouts ou suppression de lignes ou de colonnes)[37]. La bibliothèque NumPy[38] nous sera utile pour manipuler les données sous forme de matrices issues des Dataframes (structure de données à deux dimensions, comme un tableau excel) de Pandas. Ces matrices seront exploitées grâce aux ressources proposées par la bibliothèque Scikit-Learn[39]. Cette dernière implémente les principaux algorithmes d'apprentissage automatique, des outils de tests d'apprentissage, et de sélection ou d'encodage de variables. Enfin, nous utiliserons les ressources de Imbalanced-Learn pour prendre en charge les déséquilibres de données. Nous allons à présent explorer chacune de ces étapes, en commençant par l'étape clef de nettoyage des données.

[37] Pandas documentation, https://pandas.pydata.org/docs/index.html
[38] NumPy, The fundamental package for scientific computing with Python, https://numpy.org/
[39] Scikit-Learn, Machine Learning in Python, https://scikit-learn.org/stable/

2.2. Nettoyage des données

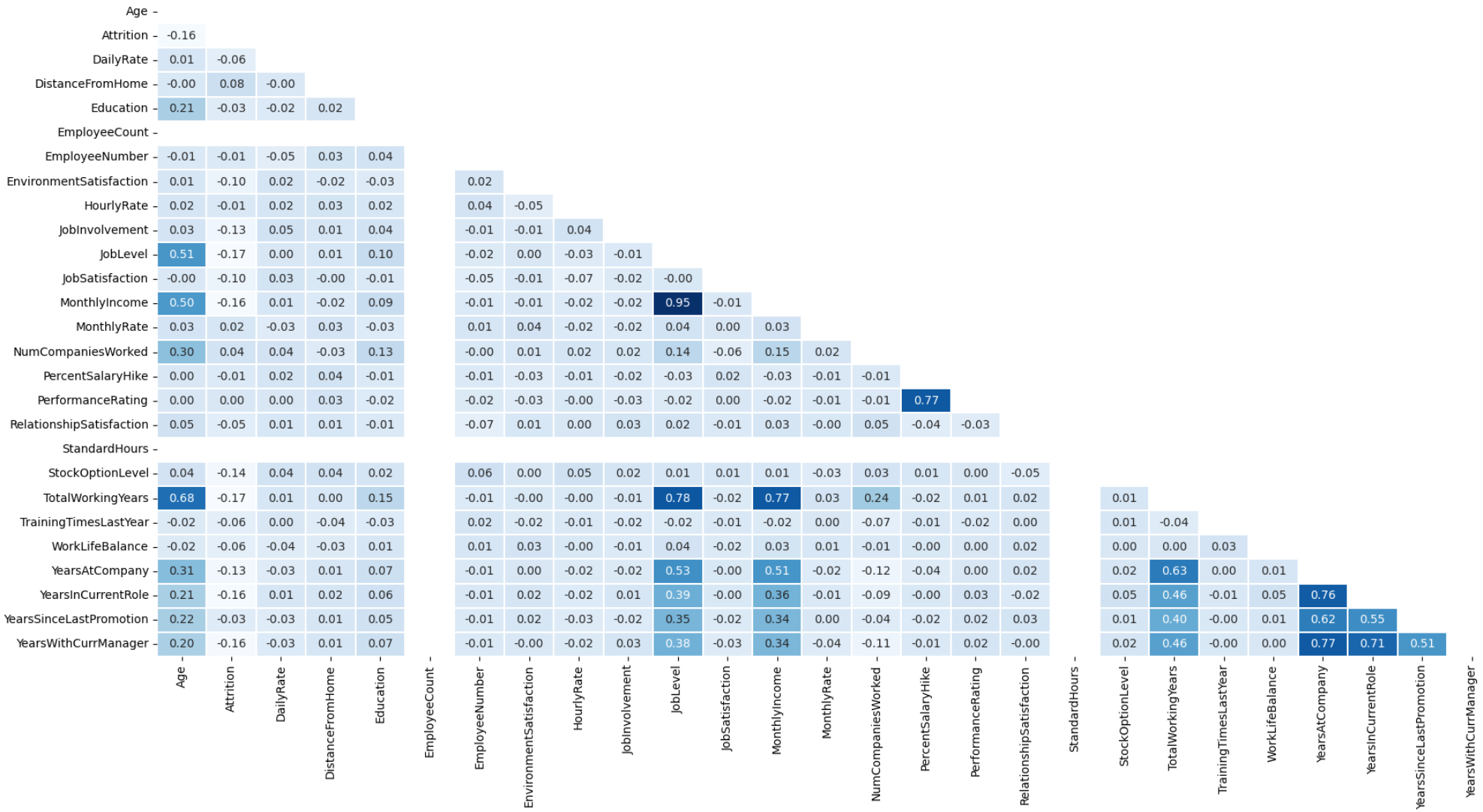
Certaines variables sont redondantes ou ne sont pas porteuses d'information, ce qui peut nuire aux algorithmes de machine learning. Leur suppression en amont, facilite la suite des préparations de données, et assure un meilleur entraînement de la machine. L'étude de la corrélation linéaire au sein de notre jeu de données peut nous permettre de repérer certaines de ces variables (au sein des variables numériques). Le coefficient de corrélations linéaires de Pearson permet de mesurer la liaison entre deux variables, et il est compris entre -1 et 1. Une forte liaison linéaire entre deux variables est située proche de 1 ou de -1 (cf. graphique représentant différent niveau de liaison en page annexe) tandis qu'une liaison nulle est égale à 0.

Ainsi, si une variable (X_i) n'a aucune liaison avec la variable Attrition (Y) ($Cor(X, Y) = 0$), elle est potentiellement peu intéressante pour nos modèles. Cette étude de corrélation nous informe également sur les liaisons entre les variables explicatives X_i elles-mêmes. Deux variables très fortement corrélées sont potentiellement porteuses de la même information. Il est dans ce cas potentiellement préférable de supprimer l'une d'entre elles.

La matrice de corrélation de nos données (cf. page suivante) montre qu'un certain nombre de variables ne sont pas porteuses d'informations pertinentes pour l'entraînement de la machine. Les variables EmployeeCount, et StandardHours ont une corrélation nulle avec l'ensemble des variables du jeu de données. Ces variables contiennent toujours les mêmes valeurs (réciproquement "1" et "80") pour chaque employé. Ces informations ne permettent pas de distinguer les employés entre eux, et donc ne permettront pas à la machine de distinguer leur comportement de démission. Il est donc préférable de supprimer ces deux colonnes. La variable catégorielle Over18 contient également toujours la même information ("Y") et a été supprimée pour les mêmes raisons.

Concernant les variables explicatives entre elles, YearsInCurrentRole, TotalWorkingYears, YearsWithCurrManager et YearsAtCompany ont une corrélation particulièrement élevée entre elles. Le même constat est observable entre TotalWorkingYear et Age. Seules les variables Age et YearsAtCompany ont été conservées.

Fig. 11 Matrice de corrélation des variables du jeu de données IBM



Ainsi, ces suppressions diminuent le travail d'encodage, de standardisation, et les autres étapes de notre méthodologie, que nous allons maintenant présenter.

2.3. Encodage et standardisation des variables

La machine n'opère des calculs qu'à partir de nombres. Elle ne peut comprendre et prédire que des nombres. Les variables qualitatives, sous forme de chaîne de caractères, doivent être traduites sous forme de chiffres. Cela s'appelle l'encodage de variables.

Dans nos données, deux types de variables sous forme de chaîne de caractère sont à encoder : les variables ordinales et les variables catégorielles. Ces deux types de variable nécessitent des encodages distincts. Les variables ordinales sont des variables dont on peut classer les modalités "du plus petit au plus grand", comme nous l'avons vu avec les variables de satisfaction au travail. La variable BusinessTravel a plusieurs modalités ("Non_Travel", "Travel_Rarely", "Travel_Frequentely") qui peuvent être classées de 1 à 3. L'encodage de BusinessTravel, ainsi que de toutes nos variables ordinales sous forme de chaîne de caractères suivent cette logique et ont été modifiées : "Non_Travel" est remplacé par 1, "Travel_Rarely" par 2, et "Travel_Frequentely" par 3.

Quant aux variables qualitatives (ou catégorielles) à l'exemple de la variable Department, ses modalités ("Human Resources", "Research & Development", "Sales") ne peuvent pas être classées. L'encodage utilisé dans ce cas-ci est un codage One Hot. Avec cette technique, chaque catégorie est représentée de manière binaire dans une colonne qui lui est propre (cf. exemple ci-dessous).

Fig. 12 Représentation d'un encodage one hot

Department
Sales
Research & Development
Sales
Human Ressources
Research & Development

One Hot encodage

Sales	Research & Development	Human Ressources
1	0	0
0	1	0
1	0	0
0	0	1
0	1	0

Ainsi, la modalité "Sales" devient une nouvelle variable où il est indiqué de manière binaire (1 = oui, 0 = non) si l'employé appartient oui ou non au département des ventes. L'ensemble de nos variables catégorielles ont été encodée de cette manière.

Donc, l'ensemble de nos variables sont maintenant représentées sous un format numérique.

Enfin, pour que chacune de nos variables soient comparables entre elles et que la machine puisse les utiliser correctement, nous avons centré-réduit nos données[40]. Chaque variable a été soustraite par sa moyenne et divisée par son écart-type tel que : $x = \frac{X - \mu}{\sigma}$ où μ représente la moyenne, X une valeur de la variable, et σ l'écart-type.

2.4. Sélection des variables d'entraînement

Tous les algorithmes d'apprentissage automatique n'ont pas les mêmes réactions aux données. Certaines variables peuvent être utiles pour un algorithme Random Forest, et seront défavorables pour un algorithme Decision Tree pour détecter les démissions. Pour s'ajuster au mieux à chaque modèle, nous nous sommes appuyés sur un algorithme d'élimination récursive de variables (RFE : Recursive Feature Elimination en anglais). Chaque modèle est entraîné sur l'ensemble des données, et l'importance de chaque variable est évaluée. Les variables les plus faibles, celles qui apportent le moins d'informations permettant de détecter une démission sont éliminées au tour par tour. Le processus est répété jusqu'à un nombre optimal de variable pour le classificateur de démissions sélectionné.

2.5. K Fold Cross-validation : la modalité d'entraînement

Pour entraîner nos algorithmes d'apprentissage automatique, et vérifier la qualité de ces entraînements, la machine a besoin de données d'apprentissages et de test. Cette méthode d'apprentissage-test s'appelle Cross Validation ou validation croisée en français. Il existe plusieurs techniques de Cross Validation dont la plus simple est la Hold-out Cross Validation[41].

Dans cette première implémentation de la Cross Validation, les données sont divisées en deux parties : 80% de ces données sont dédiées à l'entraînement de l'algorithme tandis que les 20% suivants servent de données de test (cf. figure

[40] Younes Benzaki, Mr. Mint Machine Learning made easy, https://mrmint.fr/data-preprocessing-feature-scaling-python

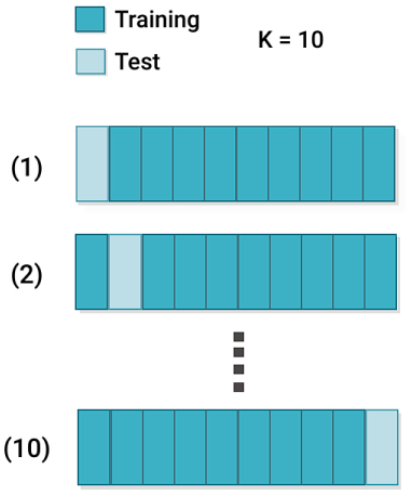
[41] Validimir Lyashenko, Cross-Validation in Machine Learning : How to Do It Right, MLOps Blog, https://neptune.ai/blog/cross-validation-in-machine-learning-how-to-do-it-right

ci-dessous). Cette sélection est aléatoire et sans remise. Elle permet à ces deux échantillons d'être représentatifs de la totalité des données (chaque tuple ayant la même probabilité de sélection pour chaque tirage au sort). L'objectif est de s'assurer que la machine n'est ni en situation de sous-apprentissage ni de sur-apprentissage, et qu'elle est capable de prédire une valeur (une démission ou non) avec des données qu'elle n'a jamais rencontrées (données de test).

Fig. 13 Schéma de sélection de données lors d'une Hold-out Cross Validation[42]

Durant notre étude, nous utiliserons une technique de Cross Validation beaucoup plus robuste, celle de la Stratified K-Fold Cross Validation. En effet, la Stratified K-Fold Cross Validation est une technique qui s'appuie sur la division des données en K groupes ou "fold". L'algorithme est entraîné sur $K - 1$ fold, et est testé sur le dernier fold, et ce, successivement jusqu'à ce que chaque fold ait servi de test. Ces opérations sont reproduites K fois.

Fig. 14 Schéma de sélection des données lors d'une Stratified K-Fold Cross Validation[43]



[42] Validimir Lyashenko, Cross-Validation in Machine Learning : How to Do It Right, MLOps Blog, https://neptune.ai/blog/cross-validation-in-machine-learning-how-to-do-it-right
[43] Validimir Lyashenko, Cross-Validation in Machine Learning : How to Do It Right, MLOps Blog, https://neptune.ai/blog/cross-validation-in-machine-learning-how-to-do-it-right

A la fin, nous mesurons en moyenne sur l'ensemble des folds de données et ces K essais, quelles ont été les performances de l'algorithme. Ainsi, avec cette méthode, l'algorithme est entraîné et testé avec toutes les parties du jeu de données, ce qui est optimal quand elles sont peu nombreuses et déséquilibrées[44].

L'intérêt de cette technique est de s'assurer que les bons ou mauvais résultats obtenus par l'algorithme ne sont pas liés à un tirage au sort favorable ou défavorable. En effet, il se peut avec la méthode Hold-out que lors de la division d'un jeu de données en deux échantillons, que ces sous-jeux de données favorisent fortement l'algorithme, ce qui exagérait ses performances.

Cette méthode permet d'avoir une mesure plus précise des performances d'un algorithme, indépendamment d'un tirage au sort favorable. Elle permet également une meilleure comparaison avec d'autres algorithmes qui pourrait être plus ou moins avantagés sur des sélections d'échantillons d'entraînement et de test différents.

Ainsi, cela nous permettra de mieux appréhender les différences de performance entre les algorithmes que nous utiliserons dans le cadre notre étude. Pour appréhender ces performances, nous utiliserons plusieurs types de mesures (métriques) que nous allons présenter.

2.6. Métriques : évaluer correctement sur des données déséquilibrées

La mesure d'efficacité d'un algorithme d'apprentissage automatique passe par la mesure de ses erreurs. Il existe 4 types de prédictions possibles pour un problème de classification binaire que l'on peut résumer à travers une matrice de confusion (cf . tableau ci-dessous où nous avons représenté les erreurs par les "cases rouges").

Tab. 3 Matrice de confusion

		Prédictions	
		Yes	No
Données	Yes	Vrai Positif	Faux Négatif
	No	Faux Positif	Vrai Négatif

Il est possible de prédire une valeur comme positive ($Y = 1$ ou $Y = \text{'}Démission\text{'}$) et qu'elle s'avère l'être réellement, il s'agit d'un "vrai positif" (ou TP pour "true positive").

[44] Ajitesh Kumar, K-Fold Cross Validation - Python Example, Data Analytics, https://vitalflux.com/k-fold-cross-validation-python-example/

Dans le cas contraire, si l'algorithme prédit une valeur positive et qu'elle est négative dans la réalité ($Y = 0$ ou $Y = 'Rester'$), alors il s'agit d'un faux positif (ou FP). Enfin, il est possible de prédire une valeur comme négative, et qu'elle le soit réellement (vrai négatif ou TN) et réciproquement (faux négatif ou FN).

Avec ces prédictions, il est possible de créer différentes métriques dont la valeur est comprise entre 0 et 1. Plus la valeur est proche de 1, plus l'algorithme est capable de bonnes prédictions. Parmi les principales métriques existantes il y a :

- Le taux de bonne classification (ou Accuracy) : $Acc = \frac{TP+TN}{TP+TN+FP+FN} = 1 - Err$;
- Le rappel ou recall, il s'agit du nombre de démissions réelles et prédites par l'algorithme divisé par l'ensemble des démissions réelles : $R = \frac{TP}{(TP+FN)}$;
- La précision qui est le nombre de démissions réelles et prédites par l'algorithme divisé par l'ensemble des démissions prédites : $P = \frac{TP}{(TP+FP)}$;
- La F-mesure (F1-score) : Il s'agit d'une métrique qui combine la précision et le rappel, en une seule valeur. Ceci est utile pour équilibrer précision et sensibilité. La F-mesure est calculée comme la moyenne harmonique entre la précision et le rappel : $F = 2 * (\frac{Précision*Rappel}{Précision+Rappel})$;
- L'aire sous la courbe ROC (ou Area Under The Curve, AUC, cf. graphique en annexe) : cette métrique mesure la performance d'un classificateur binaire. Elle fait un compromis entre précision et spécificité ($\frac{TN}{TN+FP}$). Elle peut être interprétée comme la probabilité, que parmi deux employés sélectionnés au hasard dont l'un a démissionné et l'autre non, la valeur du marqueur de l'algorithme soit plus élevée pour l'employé ayant démissionné. Un AUC-ROC de 0.5 indique que le marqueur, l'algorithme, est non-informatif et que ses prédictions ont les mêmes performances qu'un tirage au sort aléatoire[45].

Ces mesures sont plus ou moins sensibles aux données déséquilibrées. Dans nos données, nous avons près de 85% de personnes restantes dans l'entreprise tandis que près de 15% ont démissionné. Si la machine prédit toujours le fait de rester dans l'entreprise, elle aurait un score de bonnes classifications (accuracy) tout à fait correct de 85%, alors que ses prédictions sont de piètre qualité.

[45] Tutoriel : Comment lire une courbe ROC et interpréter son AUC, IDBC Groupe A2COM, https://www.idbc.fr/tutoriel-comment-lire-une-courbe-roc-et-interpreter-son-auc/

Ainsi, durant notre étude nous privilégierons l'utilisation de l'aire sous la courbe ROC présentées précédemment, pour sélectionner les bons ajustements (hyperparamètres) de nos algorithmes. Ce métrique offre un bon compromis entre capacité de prédire une démission (vrai positif) et la capacité de prédire qu'un employé reste dans l'entreprise (vrai négatif). Enfin, lors de la lecture de nos résultats, nous nous pencherons davantage sur la précision et le rappel qui sont particulièrement adaptées à l'évaluation des classificateurs travaillant sur des données déséquilibrées.

3. Protocol d'expérimentation

A travers notre travail de préparation des données et la méthodologie que nous avons sélectionnée, nous souhaitons comparer les quatre modèles de machines learning utilisés dans la littérature (Decision Tree, Régression logistique, Random Forest, SVM).

Comme nous l'avons précisé tout au long de notre étude, les données sur les démissions d'employés sont déséquilibrées. Nous souhaitons également comparer les performances de ces modèles selon la mise en place de deux méthodes permettant de prendre en charge ce déséquilibre. Ces deux méthodes sont présentées ci-dessous.

3.1. Les méthodes de prise en charge de données déséquilibrées

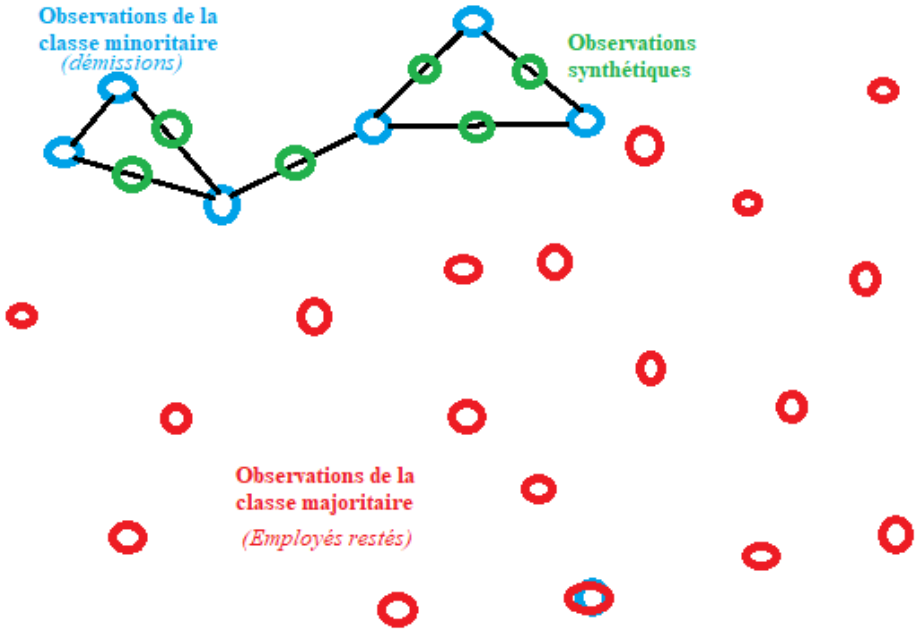
3.1.1. Le sur-échantillonnage des données minoritaires : SMOTE & ADESYN

Il existe plusieurs techniques de sur-échantillonnage des données minoritaires pour prendre en charge des données déséquilibrées. Nous souhaitons tester deux de ces techniques, la première est la technique SMOTE (Synthetic Minority Oversampling TEchnique). C'est une technique récente[46] très utilisée dans les domaines de la

[46] N. V. Chawla, et al. SMOTE: Synthetic Minority Over-sampling Technique, Journal of Artificial Intelligence Research, https://www.jair.org/index.php/jair/article/view/10302

fraude[47][48] et du médical[49] : les fraudes et les personnes malades étant très largement minoritaires par rapport aux autres dans les données de ces domaines.
La technique SMOTE crée de nouvelles observations synthétiques, des employés fictifs ayant démissionné, pour rééquilibrer la répartition des classes du jeu de données. Ces observations synthétiques sont créées grâce à la méthode des K plus proches voisins (cf. schéma en annexe). Une observation peut être représentée géométriquement par un point. Chaque observation de la classe minoritaire (démission) peut être reliée par des vecteurs à ses K plus proches voisins (K=2 dans l'exemple ci-dessous). De nouvelles observations synthétiques appartenant à la classe minoritaire peuvent être ajoutées le long de ces vecteurs représentés par des segments entre les points bleus ci-dessous.

Fig. 16 Représentation graphique de l'algorithme SMOTE



Ces observations synthétiques étant géométriquement proches des observations minoritaires, nous supposons qu'elles se "ressemblent", que ces employés fictifs auront des similitudes importantes avec les employés ayant démissionné. Cela

[47] Cuizhu Meng1, Li Zhou1 and Bisong Liu, A Case Study in Credit Fraud Detection With SMOTE and XGBoost, Journal of Physics: Conference Series, https://iopscience.iop.org/article/10.1088/1742-6596/1601/5/052016
[48] Bassam Kasasbeh et al., EFN-SMOTE — An improved unbalanced data set oversampling based on fuzzy C-means for Credit Cards Fraud detection
[49] Mehdi Naseriparsa,corresponding author1 Ahmed Al-Shammari,1,3 Ming Sheng,2 Yong Zhang,2 and Rui Zhou,RSMOTE: improving classification performance over imbalanced medical datasets National Library of Medicine, https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7292850/

permet d'entraîner l'algorithme sur une population d'employés ayant démissionné plus importante, ce qui permettrait d'améliorer ses prédictions. Cette technique permet également d'augmenter la taille de nos données d'apprentissage.

Enfin, ADASYN (Adaptive Synthetic Sampling) est une seconde méthode de sur-échantillonnage des données minoritaires[50] que nous souhaitons tester. C'est une amélioration de SMOTE. Elle permet de créer des observations synthétiques, plus difficiles à apprendre pour les algorithmes que celles de SMOTE, et qui améliorerait l'apprentissage des algorithmes[51]. En effet, cette méthode s'appuie sur une technique très similaire avec les K plus proches voisins, mais accentue la création d'observations synthétiques de la classe minoritaire, en bordure de la classe majoritaire.

3.1.2. La pondération des observations (Hyperparamètre Class_Weight)

La deuxième méthode de prise en charge des données déséquilibrées que nous souhaitons tester est celle de l'hyperparamètre Class Weight, qui est disponible sur chaque algorithme que nous allons étudier. Cet hyperparamètre permet de pondérer les observations selon leur classe. La bibliothèque Scikit Learn nous permet d'associer un poids plus important aux observations dont le label est minoritaire en affectant la pondération suivante à chaque observation[52] : n_samples / (n_classes * np.bincount(y)) où n_sample correspond au nombre d'observations totales du jeu de données, n_classes au nombre de classes (deux, à savoir "avoir démissionné" et "être resté"), et np.bincount(Y) au nombre de lignes de la classe Y de l'observation.

Ainsi, cette pondération plus importante de la classe minoritaire fait en sorte que l'algorithme accorde plus d'attention à ces observations, ce qui améliorerait la détection de cette classe minoritaire.

---

[50] Haibo He; Yang Bai; Edwardo A. Garcia; Shutao Li, IEEE, ADASYN: Adaptive synthetic sampling approach for imbalanced learning, https://ieeexplore.ieee.org/document/4633969

[51] Jakob Brandt & Emil Lanzén,Department of Statistics Uppsala University, A Comparative Review of SMOTE and ADASYN in Imbalanced Data Classification, https://www.diva-portal.org/smash/get/diva2:1519153/FULLTEXT01.pdf

[52] sklearn.utils.class_weight.compute_class_weight, Scikit Lean, https://scikit-learn.org/stable/modules/generated/sklearn.utils.class_weight.compute_class_weight.html

3.2. Modalités de comparaison

Nous souhaitons mesurer l'efficacité des différents algorithmes utilisés dans la littérature, mais aussi celle de ces techniques. Comme nous l'avons vu au chapitre sur le sous-apprentissage et sur-apprentissage, tous les algorithmes ont des hyperparamètres dont dépendent leurs résultats, et ils doivent être choisis par le Data Scientist.

3.2.1. La recherche des meilleurs hyperparamètres

A la suite du nettoyage des données, et de la sélection des variables que nous avons présentées, pour chacun de nos algorithmes, nous avons utilisé GridSearchCV() de la bibliothèque Scikit Learn (cf. code présenté ci-dessous). Nous nous centrerons dans notre présentation ici, sur un seul modèle de machine learning, la régression logistique. Notre démarche est identique pour les autres algorithmes.

Pour rechercher les meilleurs hyperparamètres d'un algorithme d'apprentissage automatique, plusieurs éléments sont fournis à la fonction GridSearchCV(), parmi ceux-ci :

- Un pipeline qui est un objet Python, qui a pour objectif d'assembler plusieurs éléments qui peuvent faire l'objet d'une cross validation[53]. Cette pipeline (logistic_model, voir capture d'écran ci-dessous) contient l'algorithme de machine learning dont les hyperparamètres optimaux sont recherchés ;
- Un dictionnaire d'hyperparamètres (paramsLogistic) qui comprend l'ensemble des hyperparamètres et des valeurs que nous voulons tester ;
- La méthode de Cross Validation (kf[54]) sélectionnée, à savoir une Stratified Kfold Cross Validation comme présentée dans notre méthodologie ;
- La métrique utilisée (roc_auc[55]) pour mesurer les performances.

L'utilisation de l'AUC ROC permet de nous assurer que les hyperparamètres sélectionnés obtiennent le plus grand score d'AUC ROC, et optimisent la prédiction des observations de la classe minoritaire (Y=1 ou "démission").

---

[53] sklearn.pipeline.Pipeline, Scikit learn, https://scikit-learn.org/stable/modules/generated/sklearn.pipeline.Pipeline.html

[54] La variable kf est définie à la ligne 195 du code présentée. Cette variable correspond à la méthode de Cross Validation indiquée dans notre méthodologie, à savoir une Stratified K-Fold cross Validation.

[55] La chaîne de caractère "roc_auc" fourni à la fonction GridsearchCV à la ligne 209 du code correspond à la métrique AUC ROC définie dans la méthodologie.

Code employé pour rechercher les meilleurs hyperparamètres pour une régression logistique dans le cadre de la détection de démission

```
#Je créé mes jeux de données d'entraînement et de test avec la Stratified KFold Cross Validation méthode
kf = StratifiedKFold(n_splits=NUMBER_KFOLD, random_state=20, shuffle=True)

#Je créé un pipeline contenant un modèle de régression logisitique
logistic_model = make_pipeline(Logistic(max_iter=20000))

#Je créé un dictionnaire d'Hyperparamètres, avec les valeurs d'hyperparamètres à tester
paramLogistic = {
      'solver': ['newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga','newton-cholesky' ],
       'penalty':['none', 'elasticnet', 'l1', 'l2'],
           'C':[0.001, 0.008, 0.01, 0.02, 0.1, 1, 10, 100]}
paramLogit = {'logisticregression__' + key: paramLogistic[key] for key in paramLogistic}


#Grâce à GridSearchCV, je cherche l'hyperparamètre optimal selon l'AUC-ROC
grid_logistic = GridSearchCV(logistic_model, param_grid=paramLogit, cv=kf, scoring='roc_auc',
                             return_train_score=True,
                             n_jobs=5)
grid_logistic.fit(X_train, y_train)

#Les attributs best_params et best_score de GridSearchCV renvoient la liste des hyperparamètres
#optimaux, ainsi que leur score (ici l'AUC-ROC)
print(grid_logistic.best_params_)
print(grid_logistic.best_score_)
```

La fonction GridSearchCV() va à tour de rôle lancer une Stratified Kfold Cross Validation sur les données d'apprentissage, avec l'algorithme d'apprentissage automatique sélectionné, et ce, pour chaque valeur d'hyperparamètre donnée dans le dictionnaire. Les meilleurs hyperparamètres ainsi que leurs résultats nous sont ensuite fournis par les attributs .best_params_ et .best_score_.

Cette procédure de recherche d'hyperparamètres sera utilisée pour chaque algorithme et pour chaque modalité de prise en charge des données déséquilibrées : sans prise en charge, avec SMOTE, avec l'hyperparamètre class weight. Dans le cas de l'utilisation de la méthode SMOTE (cf. code ci-dessous), le sur-échantillonnage des employés ayant démissionné se fait à travers la fonction SMOTE() donnée dans le pipeline[56]. L'utilisation de SMOTE() à travers un pipeline permet de s'assurer que l'apprentissage de la machine se fait avec des données

[56] make_pipeline() retourne un pipeline. Elle est une fonction existantes dans les bibliothèques scikit-learn et imbalanced-learn. La version de imbalanced-learn est utilisée pour prendre en charge SMOTE et ADASYN.

sur-échantillonnées, mais que le test de cet apprentissage se fait avec des données réelles (avec un nombre d'employés ayant démissionné minoritaire).

Lignes de code modifiée dans le cadre de la recherche des meilleurs hyperparamètres avec utilisation du sur-échantillonnage SMOTE

```
197  #Je créé un pipeline contenant un modèle de régression logisitique et
198  #un sur-échantillonneur des données
199  logistic_model = make_pipeline(SMOTE(random_state=42),Logistic(max_iter=20000))
```

La recherche des meilleurs hyperparamètres de l'algorithme avec une pondération des observations se fait en fixant l'hyperparamètre Class_Weight sur "balanced" dans le dictionnaire fourni à GridSearchCV.

Lignes de code modifiée dans le cadre de la recherche des meilleurs hyperparamètres avec l'hyperparamètre class_weight = 'balanced'

```
201  #Je créé un dictionnaire d'Hyperparamètres, avec les valeurs d'hyperparamètres à tester
202  #L'hyperparamètre "class weight" est fixé à "Balanced"
203  paramLogistic = {
204          'solver': ['newton-cg', 'lbfgs', 'liblinear', 'sag', 'saga','newton-cholesky' ],
205            'penalty':['none', 'elasticnet', 'l1', 'l2'],
206                'C':[0.001, 0.008, 0.01, 0.02, 0.1, 1, 10, 100],
207                'class_weight':['balanced']}
208  paramLogit = {'logisticregression__' + key: paramLogistic[key] for key in paramLogistic}
```

Ainsi, les hyperparamètres optimaux de nos algorithmes seront recherchés selon les trois modalités que nous souhaitons comparer : avec la méthode SMOTE, avec la pondération des effectifs de la classe minoritaire, et sans aucun traitement.

3.2.2. La comparaison

Après avoir recherché ces hyperparamètres, nous comparerons les résultats de nos algorithmes à travers les métriques que nous avons présentées dans notre méthodologie. La fonction cross_validate() de Scikit-Learn (cf. code ci-dessus) permet de lancer une Cross Validation et de retourner un ensemble de métriques sélectionnées.

Code utilisé pour lancer les cross validation et obtenir les métriques de performance

```
231
232  #Cross-validate permet d'obtenir l'ensemble des métriques fournis dans un tableau, résultat de la cross validation
233  scores = cross_validate(logistic_pipeline, X_train, y_train, cv=kf, scoring=['accuracy','recall','roc_auc','precision','f1'])
234
```

Nous allons maintenant présenter nos résultats, issus de Stratified K-fold Cross Validation. Ces résultats sont présentés avec des matrices de confusion pour voir le détail des détections de démissions, ainsi qu'avec un tableau récapitulatif des performances des différents algorithmes selon les métriques que nous avons exposées, et ce, pour chacune des prises en charge des données déséquilibrées.

4. Résultats

4.1. L'entraînement sans prise en charge du déséquilibre des données

L'algorithme de Random Forest a un taux de bonnes classifications de 85% et une précision de 69% sans prise en charge du déséquilibre dans les données. Ces taux exagèrent les performances réelles de l'algorithme qui est sous-apprentissage et ne détecte qu'une très faible partie des employés ayant démissionnées (<10%), et ce, malgré une recherche des hyperparamètres optimaux.

Tab. 4 Matrice de confusion de l'algorithme Random Forest sans prise en charge du déséquilibre des données

		Valeurs prédites	
		Rester	Démission
Valeurs réelles	Rester	1222	11
	Démission	216	21

Hyperparamètres sélectionnés : n_estimators=220, max_depth=10, min_samples_split=9, min_samples_leaf=1, graine aléatoire utilisée : random_state=42

Dans le cas de la régression logistique et du support vector machine, ce constat est d'autant plus flagrant. Ces deux algorithmes ont tous les deux un taux de bonnes classifications de 84%, mais ne détectent que très peu d'employés ayant démissionné.

Tab. 5 Matrice de confusion de la Régression Logistique sans prise en charge du déséquilibre des données

		Valeurs prédites	
		Rester	Démission
Valeurs réelles	Rester	1232	1
	Démission	232	5

Hyperparamètres sélectionnés : max_iter=20000, C=0.01, penalty=l2, solver=lbfgs, graine aléatoire utilisée : random_state=42

En effet, la régression logistique prédit 6 démissions et le support vector machine n'en prédit aucune, alors que nos données en comptabilise 237. Les valeurs de la métrique recall est ainsi particulièrement faible pour ces deux algorithmes (2% pour la régression logistique et 0% pour le support vector machine).

Tab. 6 Matrice de confusion du Support Vector Machine, sans prise en charge du déséquilibre des données

		Valeurs prédites	
		Rester	Démission
Valeurs réelles	Rester	1233	0
	Démission	237	0

Hyperparamètres sélectionnés : C=0.1, gamme=scale, kernel=rbf, graine aléatoire utilisée : random_state=42

Tab. 7 Matrice de confusion de l'arbre de décision, sans prise en charge du déséquilibre des données

		Valeurs prédites	
		Rester	Démission
Valeurs réelles	Rester	1185	48
	Démission	200	37

Hyperparamètres sélectionnés : criterion=log_loss, max_depth=8, min_samples_leaf=13, min_samples_split=2, splitter=best, graine aléatoire utilisée : random_state=42

Enfin, l'arbre de décision a un taux de bonne classification similaire aux autres algorithmes (84%) et semble avoir un sous-apprentissage moins important que les autres modèles.
L'algorithme prédit 85 démissions dont 37 sont réelles (soit une précision de 44%) sur les 237 démissions du jeu de données (soit une valeur de recall de 18%).

Tab. 8 Récapitulatif des performances des différents algorithmes d'apprentissage sans méthode de prise en charge des données déséquilibrées

	Random Forest	Régression Logistique	Décision Tree	Support Vector Machine
Accuracy	0.85	0.84	0.84	0.84
AUC ROC	0.77	0.76	0.65	0.75
F1	0.16	0.04	0.22	0.00
Précision	0.69	0.70	0.44	0.00
Recall	0.09	0.02	0.18	0.00

Dans l'ensemble, sans prise en charge du déséquilibre des données, les algorithmes sont en sous-apprentissage. Leurs taux de bonnes classifications plutôt élevés est dû au déséquilibre des données. Toutefois, ils ne sont pas capables de détecter les démissions des employés, ou dans des proportions négligeables. Seules 7% des démissions sont prédites en moyenne sur les 4 algorithmes.

4.2. L'utilisation de la méthode SMOTE

Avec la méthode de sur-échantillonnage SMOTE présentée précédemment, l'algorithme Random Forest garde un taux de bonnes classification important (86%). L'algorithme prédit 120 départs volontaires d'employés, dont plus de la moitiés (61%) sont réels.
Toutefois, bien que l'algorithme détecte près de 3 fois plus de démissions que sans SMOTE, il continue d'en détecter relativement peu parmi celles présentes dans le jeu de données (30%). Cette caractéristique est partagée avec son algorithme voisin, l'arbre de décision.

Tab. 9 Matrice de confusion du Random Forest, avec la technique de sur-échantillonnage SMOTE

		Valeurs prédites	
		Rester	Démission
Valeurs réelles	Rester	1185	48
	Démission	165	72

Hyperparamètres sélectionnés : n_estimators=1180, max_depth=12, min_samples_split=11, min_samples_leaf=3, graine aléatoire utilisée : random_state=42

Tab. 10 Matrice de confusion de l'arbre de décision avec la technique de sur-échantillonnage SMOTE

		Valeurs prédites	
		Rester	Démission
Valeurs réelles	Rester	985	248
	Démission	148	89

Hyperparamètres sélectionnés : criterion=gini, max_depth=13, min_samples_leaf=13, min_samples_split=3, splitter=best, graine aléatoire utilisée : random_state=42

L'arbre de décision ne détecte que 89 démissions sur les 237, soit 36% d'entre elles. A noter que pour un nombre semblable de détection de démissions, l'arbre de décision fait beaucoup plus d'erreurs avec une précision de 30% (vs 61% pour le Random Forest)..

Tab. 11 Matrice de confusion de la Régression Logistique avec la technique de sur-échantillonnage SMOTE

		Valeurs prédites	
		Rester	Démission
Valeurs réelles	Rester	884	349
	Démission	86	151

Hyperparamètres sélectionnés : max_iter=20000, C=0.01, penalty=l2, solver=sag, graine aléatoire utilisée : random_state=42

La régression logistique et le support vector machine ont tous les deux les taux de bonnes classifications les plus bas avec réciproquement 70% et 73% de taux de bonnes classifications. Contrairement aux deux précédents algorithmes, ils prédisent un grand nombre de démissions.

Tab. 12 Matrice de confusion du Support Vector Machine, avec la technique de sur-échantillonnage SMOTE

		Valeurs prédites	
		Rester	Démission
Valeurs réelles	Rester	907	326
	Démission	78	159

Hyperparamètres sélectionnés : C=0.1, gamme=scale, kernel=rbf, graine aléatoire utilisée : random_state=42

La régression logistique en prédit un total de 500 et le support vector machine de 485, un nombre beaucoup plus important que le nombre de départs volontaires d'employés réels, ce qui explique ce taux de bonnes classifications plus bas.
La méthode SMOTE semble améliorer les résultats de nos algorithmes, le Random Forest se distingue par une précision (61%) particulièrement élevée au détriment d'une détection importante des démissions (rappel à 30%). Ses résultats s'opposent à ceux de la régression logistique et du support vector machine qui ont des taux de rappel élevés au détriment de la précision. Quant à l'arbre de décision, ses résultats sont en retrait par rapport à ces autres algorithmes.

Tab. 13 Récapitulatif des performances des différents algorithmes d'apprentissage avec la méthode de sur-échantillonnage SMOTE

	Random Forest	Régression Logistique	Decision Tree	Support Vector Machine
Accuracy	0.86	0.70	0.75	0.73
AUC ROC	0.76	0.75	0.64	0.76
F1	0.40	0.41	0.35	0.44
Précision	0.61	0.30	0.30	0.33
Recall/Rappel	0.30	0.64	0.36	0.67

4.3 L'utilisation de la méthode ADASYN

Avec l'utilisation de notre seconde méthode de sur-échantillonnage de la classe minoritaire, ADASYN, l'algorithme Random Forest conserve un taux de bonnes classifications de 85%. En termes de métriques, ses prédictions sont similaires à celles avec SMOTE.

Tab. 14 Matrice de confusion du Random Forest, avec la technique de sur-échantillonnage ADASYN

		Valeurs prédites	
		Rester	Démission
Valeurs réelles	Rester	1189	44
	Démission	172	65

Hyperparamètres sélectionnés : n_estimators=220, max_depth=13, min_samples_split=11, min_samples_leaf=1, graine aléatoire utilisée : random_state=42

L'algorithme prédit un faible nombre de démissions (109 démissions vs 120 avec SMOTE) avec une précision relativement élevée de 61%. Quant à la régression logistique et au support vector machine, tout comme avec SMOTE, ils enregistrent avec ADASYN une prédiction élevée de démissions (respectivement 505 démissions et 512 démissions) au dépend de leur précision (respectivement à 30% et 31%).

Tab. 15 Matrice de confusion de la Régression Logistique avec la technique de sur-échantillonnage ADASYN

		Valeurs prédites	
		Rester	Démission
Valeurs réelles	Rester	877	356
	Démission	88	149

Hyperparamètres sélectionnés : max_iter=20000, C=0.008, penalty=l2, solver=saga, graine aléatoire utilisée : random_state=42

C’est une nouvelle fois, cette précision plus faible et ces prédictions de taux de démissions élevés, qui font baisser le taux de bonnes prédictions de ces deux algorithmes comparativement à l’algorithme Random Forest.

Tab. 16 Matrice de confusion du Support Vector Machine, avec la technique de sur-échantillonnage ADASYN

		Valeurs prédites	
		Rester	Démission
Valeurs réelles	Rester	881	352
	Démission	77	160

Hyperparamètres sélectionnés : C=0.1, gamme=500, kernel=linear, graine aléatoire utilisée : random_state=42

L’arbre de décisions a lui aussi un taux de bonnes classifications plus bas que celui du Random Forest (66%).

Tab. 17 Matrice de confusion de l’arbre de décision avec la technique de sur-échantillonnage ADASYN

		Valeurs prédites	
		Rester	Démission
Valeurs réelles	Rester	892	341
	Démission	128	109

Hyperparamètres sélectionnés : ccp_alpha=0.001, criterion=entropy, max_depth=9, min_samples_leaf=15, min_samples_split=3, splitter=random, graine aléatoire utilisée : random_state=42

Cela s’explique également par une prédiction importante de démissions (450 démissions prédites). Toutefois ses résultats sont en deçà de la régression logistique et du support vector machine, avec une précision et une valeur de recall moins bonnes.

Ainsi, l’utilisation d’ADASYN semble avoir des résultats similaires qu’avec SMOTE et améliore la détection de chacun des algorithmes comparativement à notre essai sans prise en charge des données déséquilibrées. Seul l’arbre de décision semble avoir des résultats légèrement meilleurs avec ADASYN qu’avec SMOTE.

Néanmoins, une nouvelle fois, cet algorithme a des performances en dessous des autres.

Tab. 18 Récapitulatif des performances des différents algorithmes d'apprentissage avec la méthode de sur-échantillonnage ADASYN

	Random Forest	Régression Logistique	Decision Tree	Support Vector Machine
Accuracy	0.85	0.70	0.66	0.71
AUC ROC	0.77	0.75	0.66	0.75
F1	0.37	0.40	0.31	0.43
Précision	0.61	0.30	0.25	0.31
Recall	0.27	0.63	0.54	0.68

4.4. La pondération des observations

En affectant un poids plus élevé aux observations de la classe minoritaire grâce à la configuration de l'hyperparamètre class_weight sur "balanced", l'algorithme Random Forest conserve un taux de bonnes classification élevé (84%). L'algorithme prédit une nouvelle fois un faible nombre de départs volontaires des employés (111 démissions vs 247 dans la réalité). La précision de l'algorithme avec la pondération des observations, comparativement aux deux précédentes méthodes, est en déclin (49% vs 61% avec ADASYN et 61% avec SMOTE).

Tab. 19 Matrice de confusion du Random Forest, avec l'utilisation de l'hyperparamètre class weight réglé sur "balanced"

		Valeurs prédites	
		Rester	Démission
Valeurs réelles	Rester	1176	57
	Démission	183	54

Hyperparamètres sélectionnés : n_estimators=220, max_depth=10, min_samples_split=9, min_samples_leaf=1, class_weight=balanced, graine aléatoire utilisée : random_state=42

La régression logistique et le support vector machine continuent d'avoir des résultats similaires. Ils ont des taux de bonnes classifications relativement élevés et proches : 72% pour le support vector machine et 69% pour la régression logistique.

Tab. 20 Matrice de confusion de la Régression Logistique avec l'utilisation de l'hyperparamètre class weight réglé sur "balanced"

		Valeurs prédites	
		Rester	Démission
Valeurs réelles	Rester	839	394
	Démission	67	170

Hyperparamètres sélectionnés : max_iter=20000, C=0.01, penalty=l2, solver=lbfgs, class_weight=balanced, graine aléatoire utilisée : random_state=42

Ces deux algorithmes ont toujours tendance à prédire un grand nombre de démissions, en particulier la régression logistique avec 564 départs volontaires prédits, en dépit de la précision. Celle-ci reste relativement faible pour ces deux algorithmes avec respectivement des précisions de 30% pour la régression logistique et de 32% pour le support vector machine.

Tab. 21 Matrice de confusion du Support Vector Machine, avec l'utilisation de l'hyperparamètre class weight réglé sur "balanced"

		Valeurs prédites	
		Rester	Démission
Valeurs réelles	Rester	904	329
	Démission	83	154

Hyperparamètres sélectionnés : C=0.1, gamme=scale, kernel=rbf, class_weight=balanced, graine aléatoire utilisée : random_state=42

Une nouvelle fois, l'arbre de décision a les performances les moins importantes avec un taux de bonnes classifications de 63% dû à une prédiction très importante de démissions (519 démissions) et une précision très faible (25%).
Ainsi, la régression logistique et le support vector machine ont des résultats similaires qu'avec ADASYN et SMOTE. Quant au Random Forest, ses performances

ont diminué avec cette pondération par rapport aux deux méthodes de sur-échantillonnage. Cela est visible à travers la métrique F1 qui prend en compte la précision et le rappel.

Tab. 22 Matrice de confusion de l'arbre de décision, avec l'utilisation de l'hyperparamètre class weight réglé sur "balanced"

		Valeurs prédites	
		Rester	Démission
Valeurs réelles	Rester	842	391
	Démission	109	128

Hyperparamètres sélectionnés : criterion=log_loss, max_depth=8, min_samples_leaf=13, min_samples_split=2, splitter=best, class_weight=balanced, graine aléatoire utilisée : random_state=42

Le Random Forest est passé d'une valeur de F1 d'à peu près 40% à 30% avec la pondération. Enfin, l'arbre de décision reste ici aussi en retrait en termes de performances par rapport aux autres algorithmes.

Tab. 23 Récapitulatif des performances des différents algorithmes d'apprentissage avec l'utilisation de l'hyperparamètre class_weight réglé sur "balanced"

	Random Forest	Régression Logistique	Decision Tree	Support Vector Machine
Accuracy	0.84	0.69	0.63	0.72
AUC ROC	0.77	0.75	0.65	0.76
F1	0.31	0.42	0.35	0.43
Précision	0.49	0.30	0.25	0.32
Recall	0.23	0.72	0.55	0.65

4.5. Discussion

Les algorithmes utilisés sur le jeu de données d'IBM ont des performances moyennes pour détecter les démissions. Ces performances moyennes

correspondent aux performances constatées dans la littérature[57] sur ces mêmes données. Ces performances sont largement améliorées avec la prise en charge du déséquilibre de données entre les observations d'employés ayant démissionné et les autres. Les différentes méthodes que nous avons utilisées pour prendre en charge ce déséquilibre ont particulièrement accentué la capacité de prédire des démissions des modèles (Recall à 7% sans méthode vs. 49% minimum en moyenne pour les autres). En moyenne les trois méthodes de prise en charge du déséquilibre de données ont des résultats voisins, mais elles impactent de manière différente les algorithmes testés.

Tab. 24 Moyennes des performances des modèles pour chacune des méthodes de prise en charge du déséquilibre des données

	Sans	SMOTE	ADESYN	Pondération
Accuracy	0.84*	0.76	0.73	0.72
AUC ROC	0.73	0.73	0.73	0.73
F1	0.11	0.40	0.38	0.38
Précision	0.46	0.39	0.37	0.34
Recall	0.07	0.49	0.53	0.54

Il s'agit des moyennes de l'ensemble des algorithmes d'apprentissage, pour chaque métrique, et pour chaque méthode de prise en charge du déséquilibre des données. *Note de lecture : Sans méthode d'équilibrage des données, les algorithmes d'apprentissage automatique obtiennent en moyenne un taux de bonnes classifications de 84%.

L'arbre de décision a des performances plus importantes que les autres algorithmes lorsque le déséquilibre dans les données n'est pas pris en charge. A l'inverse, avec l'ensemble des méthodes de prise en charge de ce déséquilibre, ses résultats sont largement en deçà des autres algorithmes. Quant à l'algorithme Random Forest, il obtient des résultats satisfaisants en particulier avec le sur-échantillonnage SMOTE et ADESYN. Ses performances sont en déclin avec la pondération des observations. Cet algorithme est particulièrement précis mais a pour faiblesse une capacité de prédiction relativement faible avec des valeurs de recall de l'ordre de 30% avec

[57] İrem ERSÖZ KAYA, Oya KORKMAZ, Machine Learning Approach for Predicting Employee Attrition and Factors Leading to Attrition, Cukurova University Journal of the Faculty of Engineering, 36(4), pp. 913-928, December 2021, https://dergipark.org.tr/en/download/article-file/2147635

ADESYN et SMOTE. Au contraire, la régression logistique et le support vector machine conservent des performances similaires avec les trois modalités de prise en charge des données déséquilibrées. Ces deux algorithmes sont aussi plus enclins à détecter les démissions, mais au prix d'une précision moindre par rapport au Random Forest.

Ainsi, nous pouvons choisir de prédire moins de démissions mais avec plus de chance qu'elles se produisent, avec le Random Forest, ou d'en prédire davantage mais avec une précision moindre. Ce choix relève davantage d'un problème métier. En effet, il peut être préférable de viser "trop large" dans le cas où les mesures prises pour éviter les démissions ne seraient pas trop coûteuses. Dans ce cas, le choix du support vector machine ou de la régression logistique (avec une prise en charge du déséquilibre des données) serait ici plus judicieux. A l'inverse, nous pourrions privilégier la précision avec le Random Forest utilisé avec une méthode de sur-échantillonnage.

En outre, les performances plutôt moyennes des algorithmes s'expliquent principalement par le nombre d'observations. Nous avons mené une étude complémentaires sur un second jeu de données d'environ 20.000 observations, avec des caractéristiques similaires (cf. tableaux en annexe), fourni par IBM sur Kaggle[58]. Sur ces données, l'algorithme Random Forest s'est particulièrement démarqué avec un taux de bonnes prédictions de l'ordre de 95% sans prise en charge du déséquilibre de données et 98% avec SMOTE. Ainsi, les méthodes de sur-échantillonnage ont un impact d'autant plus important que le jeu de données contient un faible nombre d'observations, ou que le déséquilibre qui le caractérise est important, ce qui est également observé dans la littérature[59]. Ces méthodes de sur-échantillonnages sont donc particulièrement propices pour les entreprises de taille modeste dans le cadre de la détection des démissions.

[58] RUshikesh Ghate, Capstone Project-IBM Employee Attrition Prediction, https://www.kaggle.com/datasets/rushikeshghate/capstone-projectibm-employee-attrition-prediction?select=IBM+HR+Data+new.csv

[59] Jakob Brandt & Emil Lanzén, A Comparative Review of SMOTE and ADASYN in Imbalanced Data Classification, Department of Statistics Uppsala University, https://www.diva-portal.org/smash/get/diva2:1519153/FULLTEXT01.pdf

Conclusion

Ainsi, l'apprentissage automatique peut être une méthode fiable pour détecter les démissions d'employés dans les entreprises de tailles relativement grandes (> 1500 employés). En effet, notre étude montre que l'algorithme Random Forest souffre d'une relative faible détection des démissions sur un jeu de données de 1470 employés, tandis que le support vector machine et la régression logistique manquent de précisions. Une exploitation de données aux caractéristiques similaires a montré que sur des données de taille plus importante, l'apprentissage automatique, et en particulier l'algorithme de Random Forest obtient des résultats très satisfaisants.

La prise en charge du déséquilibre des données est un élément important pour détecter correctement la démission d'employés. En particulier, celle-ci est d'autant plus pertinente que les données sont en faible nombre, et que le déséquilibre entre employés partis et ceux restés dans l'entreprise est important. Les méthodes SMOTE et ADESYN ont amélioré de manière générale le caractère prédictif de tous les algorithmes, faisant de ces méthodes des outils particulièrement précieux, en particulier pour les entreprises de taille modeste ou avec un fort déséquilibre. Quant à la pondération des observations, elle apporte également de bons résultats, mais les performances du Random Forest déclinent avec celle-ci. Notre étude démontre donc qu'il peut être utile de tester chacune de ces méthodes, ce qui peut se faire rapidement avec les bibliothèques Scikit-Learn et Imbalanced-Learn de Python, puisqu'elles améliorent le caractère prédictif des algorithmes de manière différente selon l'algorithme (et sans doute selon la structure du jeu de données).

Enfin, l'utilisation du machine learning est possible à condition d'avoir les données pour le mettre en œuvre. Les données telles que l'âge, le genre, la situation matrimoniale, sont courantes dans les SIRH. Néanmoins, les données concernant la satisfaction le sont beaucoup moins, alors qu'il s'agit d'un facteur important dans le cadre des départs volontaires d'employés. Beaucoup d'entreprises, à l'instar de Dassault Systèmes, mettent en place annuellement des rendez-vous employés-manager. A l'issue de ces entretiens, les protagonistes indiquent sur des logiciels dédiés (qui communiquent avec le SI de l'entreprise), leurs ressentis, aspirations, ou encore un avis sur leurs performances. Ces informations sont importantes, mais ne sont exploitables ou difficilement exploitables de manière quantitative. Ce type de logiciels pourrait intégrer des systèmes de notation allant de

1 à 4 sur diverses thématiques comme la satisfaction au travail, pour permettre une exploitation quantitative des données, et donc permettre une utilisation efficace du machine learning.

Bibliographie - Webographie

Qu’est-ce qu'un SIRH ?, SAP,
https://www.sap.com/france/products/hcm/employee-central-hris/what-is-hris.html

Tout savoir sur la notion de turnover en entreprise, Le Figaro,
https://recruteur.lefigaro.fr/article/tout-savoir-sur-la-notion-de-turnover-en-entreprise/
(01/05/2023)

Attrition Issues and Retention Challenges of Employees, Brijesh Kishore Goswami, Sushmita Jha, International Journal of Scientific & Engineering Research Volume 3, Issue 4, April-2012

François Desnoyers, Le turnover des salariés pénalise la sécurité informatique,
https://www.lemonde.fr/emploi/article/2019/12/11/le-turnover-des-salaries-penalise-la-securite-informatique_6022404_1698637.html

Adrien Lagouge, Ismael Ramajo, Victor Barry, La France vit-elle une “Grande démission” ?, Dares (Direction de l’Animation de la Recherche et des Statistiques),
https://dares.travail-emploi.gouv.fr/publication/la-france-vit-elle-une-grande-demission

Turnover Rates: Factors and Reduction Strategies, Indeed,
https://www.indeed.com/career-advice/career-development/turnover-rates
(01/05/2023)

Apprentissage automatique, Wikipédia,
https://fr.wikipedia.org/wiki/Apprentissage_automatique (01/05/2023)

The Rise of Analytics in HR, Linkedin,
https://business.linkedin.com/content/dam/me/business/en-us/talent-solutions/talent-intelligence/workforce/pdfs/Final_v2_NAMER_Rise-of-Analytics-Report.pdf

Vincent S. Flowers and Charles L. Hughes, Why Employees Stay, Employee Retention, Harvard Business Review, https://hbr.org/1973/07/why-employees-stay

Brooks C. Holtorn, Terence Mitchell, et al., Turnover and Retention Research: A Glance at the Past, a Closer Review of the Present, and a Venture into the Future, The Academy of Management Annals, https://www.researchgate.net/publication/233055591_5Turnover_and_Retention_Research_A_Glance_at_the_Past_a_Closer_Review_of_the_Present_and_a_Venture_into_the_Future

Vermandere C., Impact of salary on job satisfaction, Eurofound, https://www.eurofound.europa.eu/publications/article/2013/impact-of-salary-on-job-satisfaction

Aziri Brikend, Job Satisfaction : A Litterature review, https://mrp.ase.ro/no34/f7.pdf

Arthur Samuel (computer scientist), Wikipedia, https://en.wikipedia.org/wiki/Arthur_Samuel_(computer_scientist) (05/08/2023)

Timeline of machine learning, Wikipedia, https://en.wikipedia.org/wiki/Timeline_of_machine_learning (28/07/2023)

Quels facteurs ont permis l'essor de l'intelligence artificielle ?, Musée Informatique, https://www.museeinformatique.fr/facteurs-essor-intelligence-artificielle/

Qu'est-ce que le machine learning, NetApp, https://www.netapp.com/fr/artificial-intelligence/what-is-machine-learning/#:~:text=Le%20machine%20learning%2C%20sp%C3%A9cialit%C3%A9%20de,de%20d%C3%A9cision%20sans%20interaction%20humaine .

Quel est le rapport entre le Machine Learning et l'IA ?, SAP, https://www.sap.com/canada-fr/products/artificial-intelligence/what-is-machine-learning.html

What is supervised learning?, IBM, https://www.ibm.com/topics/supervised-learning (02/05/2023)

MACHINE LEARNING VS HUMAN DECISION MAKING (SIMILARITÉS ET DIFFÉRENCES), Je veux être DataScientist, https://www.jeveuxetredatascientist.fr/machine-learning-vs-human-decision-making-similarites-et-differences/ (10/07/2023)

Zakariyaa ISMAILI, Aprentissage supervisé vs Non supervisé, https://brightcape.co/apprentissage-supervise-vs-non-supervise/#:~:text=L'objectif%20de%20l'apprentissage%20non%20supervis%C3%A9%20est%20de%20mod%C3%A9liser,r%C3%A9ponse%20correcte%20ni%20d'enseignant.

Prediction of Employee Turnover in Organizations using Machine Learning Algorithms - A case for Extreme Gradient Boosting, Rohit Punnoose, Pankaj Ajit, International Journal of Advanced Research

in Artificial Intelligence, Vol. 5, No. 9, 2016 https://pdfs.semanticscholar.org/fa49/19810eaee67e851ad13775b78c94217a7908.pdf

Hyperparamètre, Wikipedia, https://fr.wikipedia.org/wiki/Hyperparam%C3%A8tre (05/07/2023)

Chloé-Agathe Azencott, Introduction au Machine Learning, https://cazencott.info/dotclear/public/lectures/IntroML_Azencott.pdf

Norsuhada Mansor et al., Journal of Advanced Computer Science and Applications (IJACSA), Machine Learning for Predicting Employee Attrition, https://www.researchgate.net/publication/356809874_Machine_Learning_for_Predicting_Employee_Attrition

Taha Yassine Ben Ali, Données déséquilibrées, que faire ?, https://blog.octo.com/donnees-desequilibrees-que-faire/

Olivier Serpe, Le Coeur de l'intelligence artificielle : l'apprentissage automatique (part 1), EchoSciences Auvergne, https://www.echosciences-auvergne.fr/articles/le-coeur-de-l-intelligence-articielle-l-apprentissage-automatique-part1

Bias and Variance in Machine Learning, Java T point, https://www.javatpoint.com/bias-and-variance-in-machine-learning

Tony Yiu, Understanding Random Forest, Towards Data Science, https://towardsdatascience.com/understanding-random-forest-58381e0602d2

Qu'est-ce que la régression logistique ?, IBM, https://www.ibm.com/fr-fr/topics/logistic-regression

Le Bagging en Machine learning, de quoi s'agit-il ?, Kobia, https://kobia.fr/le-bagging-en-machine-learning-de-quoi-sagit-il/

Régression Logistique, Rappel Théorique, SPSS; https://spss.espaceweb.usherbrooke.ca/regression-logistique/

CNAM - UE RCP209, Cours SVM linéaires, CNAM, https://cedric.cnam.fr/vertigo/cours/ml2/coursSVMLineaires.html

Rushikesh Pupale, R. Support Vector Machines(SVM) — An Overview. Towards data science, https://towardsdatascience.com/https-medium-com-pupalerushikesh-svm-f4b42800e989

Géron, A.: Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems (2017)

Younes Benzaki, Mr. Mint Machine Learning made easy, https://mrmint.fr/data-preprocessing-feature-scaling-python

Validimir Lyashenko, Cross-Validation in Machine Learning : How to Do It Right, MLOps Blog, https://neptune.ai/blog/cross-validation-in-machine-learning-how-to-do-it-right

Ajitesh Kumar, K-Fold Cross Validation - Python Example, Data Analytics, https://vitalflux.com/k-fold-cross-validation-python-example/

N. V. Chawla, et al. SMOTE: Synthetic Minority Over-sampling Technique, Journal of Artificial Intelligence Research, https://www.jair.org/index.php/jair/article/view/10302

Cuizhu Meng1, Li Zhou1 and Bisong Liu, A Case Study in Credit Fraud Detection With SMOTE and XGBoost, Journal of Physics: Conference Series, https://iopscience.iop.org/article/10.1088/1742-6596/1601/5/052016

Bassam Kasasbeh et al., EFN-SMOTE — An improved unbalanced data set oversampling based on fuzzy C-means for Credit Cards Fraud detection

A. Mary Sowjanya, Owk Mrudula, Effective treatment of imbalanced datasets in health care using modified SMOTE coupled with stacked deep learning algorithms. Appl Nanosci 13, 1829–1840 (2023). https://doi.org/10.1007/s13204-021-02063-4

İrem ERSÖZ KAYA, Oya KORKMAZ, Machine Learning Approach for Predicting Employee Attrition and Factors Leading to Attrition, Cukurova University Journal of the Faculty of Engineering, 36(4), pp. 913-928, December 2021, https://dergipark.org.tr/en/download/article-file/2147635

Jakob Brandt & Emil Lanzén,Department of Statistics Uppsala University, A Comparative Review of SMOTE and ADASYN in Imbalanced Data Classification, https://www.diva-portal.org/smash/get/diva2:1519153/FULLTEXT01.pdf

Annexe

Fig. Représentation graphique en nuage de points, de deux variables, selon des coefficient de corrélation différents[60]

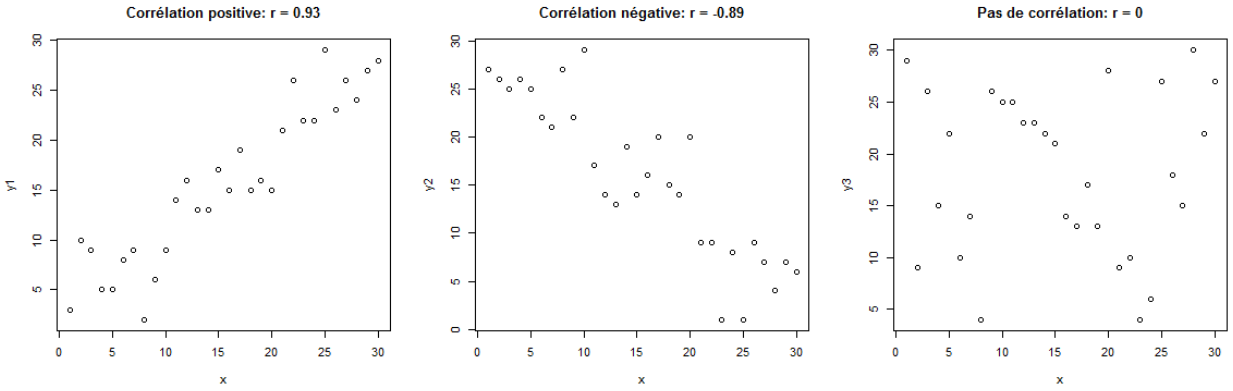
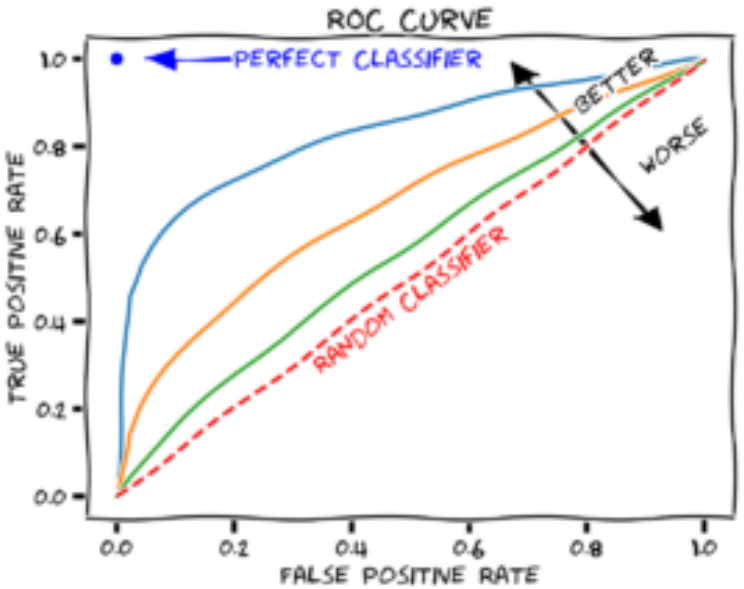


Fig. Exemple de courbe ROC[61]



[60] Corrélation de Pearson, Biostat, http://www.biostat.ulg.ac.be/pages/Site_r/corr_pearson.html#:~:text=Corr%C3%A9lation%20de%20Pearson&text=La%20valeur%20de%20r%20obtenue,sont%20r%C3%A9ellement%20corr%C3%A9l%C3%A9es%20ou%20pas.

[61] Tutoriel : Comment lire une courbe ROC et interpréter son AUC ?, IDB Groupe A2COM, https://www.idbc.fr/tutoriel-comment-lire-une-courbe-roc-et-interpreter-son-auc/

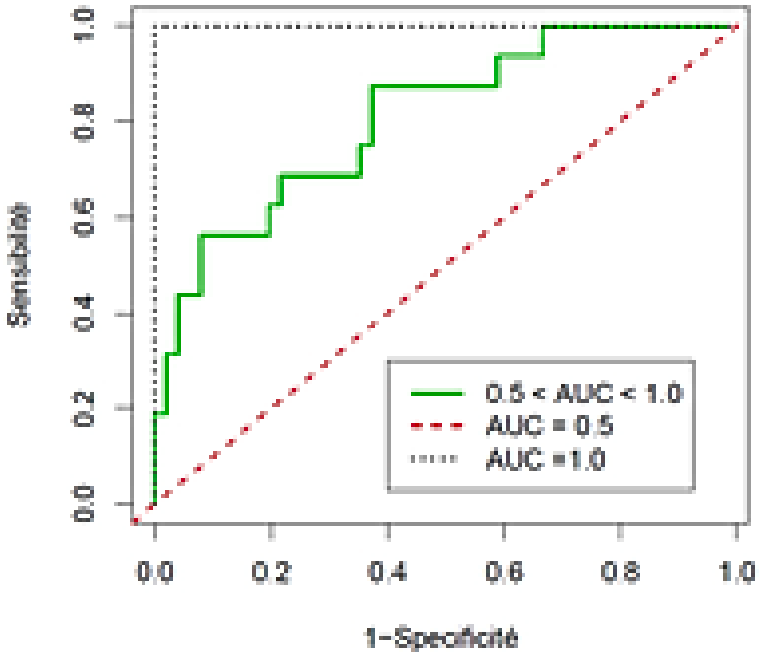
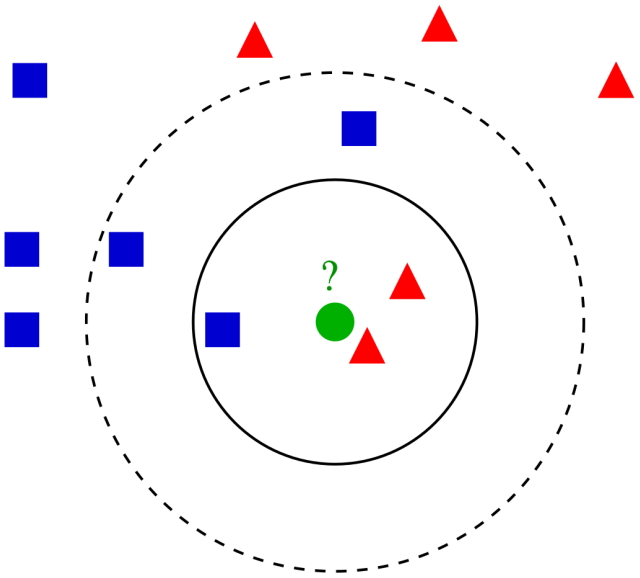


Fig. Schéma explicatif de la méthode des K plus proches voisins[62]



L'objectif de l'algorithme des K plus proches voisins est de classer un point en fonction de ses K voisins voisins les plus proches, K étant un paramètre sélectionnable. Dans cet exemple, en considérant le point vert dont la classe est indéterminée, si l'on choisit K=3, les trois plus proches voisins sont dans le premier cercle. Deux de ces voisins sont rouges tandis qu'un est bleu, les rouges étant majoritaires, pour K=3 la classe prédite est "rouge".

[62] Méthode des K plus proches voisins, Wikipedia, https://fr.wikipedia.org/wiki/M%C3%A9thode_des_k_plus_proches_voisins (18/08/2023)

Pour K=5, l'ensemble des voisins sont situés dans le cercle en pointillés. Cette fois, il y a trois voisins bleus contre deux rouges. Les bleus étant majoritaires pour K=5, la classe prédite est "bleu".

Tab. Liste des variables en chaîne de caractère du jeu de données IBM-attrition avec 22 000 observations

Nom de variable	type	mode	nombre	fréquence
Attrition	string[python]	Current employee	19525	84,2
BusinessTravel	string[python]	Travel_Rarely	16441	70,9
Department	string[python]	Research & Development	15122	65,2
EducationField	string[python]	Life Sciences	9569	41,3
EmployeeNumber	string[python]	23244	6	0,0
Gender	string[python]	Male	13904	60,0
JobRole	string[python]	Sales Executive	5055	21,8
MaritalStatus	string[python]	Married	10612	45,8
Over18	string[python]	Y	23186	100,0
OverTime	string[python]	No	16623	71,7

Tab. Liste des variables numériques du jeu de données IBM-attrition avec 22 000 observations

Nom de variable	type	moyenne	écart-type	min	25%	50%	75%	max
Age	Int64	36,92	9,12	18,00	30,00	36,00	43,00	60,00
Application ID	int64	135202,7	6776,1	123456,0	129337,3	135204,5	141061,8	146972,0
DailyRate	Int64	802,2	403,2	102,0	465,0	802,0	1157,0	1499,0
Education	Int64	2,9	1,0	1,0	2,0	3,0	4,0	5,0
EmployeeCount	int64	1,0	0,0	1,0	1,0	1,0	1,0	1,0
EnvironmentSatisfaction	Int64	2,7	1,1	1,0	2,0	3,0	4,0	4,0
JobInvolvement	Int64	2,7	0,7	1,0	2,0	3,0	3,0	4,0
JobSatisfaction	Int64	2,7	1,1	1,0	2,0	3,0	4,0	4,0
JobLevel	Int64	2,1	1,1	1,0	1,0	2,0	3,0	5,0
MonthlyRate	Int64	14302,6	7100,3	2094,0	8053,0	14222,0	20460,0	26999,0
MonthlyIncome	Int64	6507,4	4706,4	1009,0	2911,0	4936,0	8380,0	19999,0
NumCompaniesWorked	Int64	2,7	2,5	0,0	1,0	2,0	4,0	9,0
PerformanceRating	Int64	3,2	0,4	3,0	3,0	3,0	3,0	4,0
RelationshipSatisfaction	Int64	2,7	1,1	1,0	2,0	3,0	4,0	4,0
StandardHours	Int64	80,0	0,0	80,0	80,0	80,0	80,0	80,0
StockOptionLevel	Int64	0,8	0,9	0,0	0,0	1,0	1,0	3,0
TotalWorkingYears	Int64	11,3	7,8	0,0	6,0	10,0	15,0	40,0
TrainingTimesLastYear	Int64	2,8	1,3	0,0	2,0	3,0	3,0	6,0
WorkLifeBalance	Int64	2,8	0,7	1,0	2,0	3,0	3,0	4,0
YearsAtCompany	Int64	7,0	6,1	0,0	3,0	5,0	10,0	40,0
YearsInCurrentRole	Int64	4,2	3,6	0,0	2,0	3,0	7,0	18,0
YearsSinceLastPromotion	Int64	2,2	3,2	0,0	0,0	1,0	3,0	15,0
YearsWithCurrManager	Int64	4,1	3,6	0,0	2,0	3,0	7,0	17,0
DistanceFromHome	float64	9,2	8,1	1,0	2,0	7,0	14,0	29,0