

Grant Thornton
Applications & Projets
29 Rue du Pont
92200 Neuilly-sur-Seine

DAILLE Thomas
Université Paris Dauphine
M2 MIAGE ID en Apprentissage
Année 2021-2022

Comment l'analyse de données peut-elle permettre d'anticiper le départ d'un collaborateur afin de réduire le turn-over d'une organisation ?

Année universitaire 2021 / 2022

Nom et prénom de l'étudiant : DAILLE Thomas

Formation : M2 MIAGE spécialité Informatique Décisionnelle

Entreprise d'accueil : Grant Thornton

Tuteur enseignant : MAYAG Brice

Maître d'apprentissage : BELHASSEN Bruno / ZERZERI Hichem

Résumé :

Face au problème de la grande démission, les organisations tentent aujourd'hui de mettre en place des stratégies de rétention de leurs collaborateurs. En effet, un turn-over élevé peut être dangereux pour l'entreprise qui verra son image ternie, ses coûts explosés et sa productivité diminuée (engendre également l'usure des parties présentes, managers par la nécessité de recruter, former à nouveau, évaluer, etc.) Grâce à la mise en place de projets d'analyses descriptives, les organisations tentent d'expliquer les facteurs de démission qu'elles rencontrent. Loin d'être suffisante, cette analyse descriptive peut aujourd'hui être complétée par une analyse prédictive visant à comprendre les collaborateurs en les analysant par le biais des variables de suivi. Grâce à l'intelligence artificielle et notamment au Machine Learning, les analystes sont aujourd'hui capables d'estimer la probabilité de départ d'un collaborateur. Grâce à ces probabilités, l'organisation peut alors mettre en place des actions ciblées de rétention et ainsi tenter plus efficacement de conserver ses talents.

Table des matières

Liste des figures	5
Remerciements	7
I. Introduction générale.....	8
1.1 Mise en contexte	8
1.2 Problématique et intérêts	9
1.3 Plan du document.....	10
II. Anticiper le départ d'un collaborateur.....	11
2.1 Le turn-over, un indicateur clé	11
2.1.1 Définition.....	11
2.1.2 L'analyse du taux de turnover	12
2.1.3 Les causes et les conséquences.....	14
2.2 Les solutions RH.....	15
2.2.1 Les métriques	15
2.2.2 Les systèmes d'informations des ressources humaines.....	17
2.3 Un problème de classification	20
2.3.1 Le Machine Learning	20
2.3.2 Les problèmes de classification	27
2.3.3 Les outils informatiques	46
III. Mise en application : le cas d'IBM	50
3.1 Introduction.....	50
3.1.1 Identification du besoin.....	50
3.1.2 Choix de l'outil et collecte des données.....	51
3.2 EDA (Exploratory Data Analysis).....	51
3.2.1 Découverte du jeu de données	52
3.2.2 Exploration des données	55
3.3 Choix des modèles et préparation des données	62
3.3.1 Les modèles	62
3.3.2 Data processing	62
3.4 Implémentation des modèles.....	64
3.4.1 Etablir une Baseline	64
3.4.2 Régression Logistique	64
3.4.3 Random Forest	66
3.4.4 XGBoost	70

3.5	Mise en production	72
3.5.1	Quel modèle choisir ?.....	72
3.5.2	Quel sont les gains ?.....	73
3.6	Pour aller plus loin	75
3.6.1	Utilisation et alternatives	76
3.6.2	Proposition d'une méthode pour l'organisation	76
	Conclusion générale	78
	Bibliographie	80

Liste des figures

Figure 1 - Durée moyenne dans un poste selon l'âge du collaborateur	8
Figure 2 - Turn-over par secteur d'activité en 2019	13
Figure 3 - Découpage du turn-over classique.....	17
Figure 4 - Machine Learning et DataScience	21
Figure 5 - Processus projet Machine Learning	23
Figure 6 - Exemple de validation croisé à 3 partitionnements.....	26
Figure 7 - Exemple de stratification	26
Figure 8 - Matrice de confusion	28
Figure 9 - Courbe ROC lambda	32
Figure 10 - Courbe ROC théorique	32
Figure 11 - PR curve lambda.....	33
Figure 12 - Fonction sigmoïde	36
Figure 13 - Schéma arbre de décision	38
Figure 14 - Exemple arbre de décision	38
Figure 15 - Schéma forêt aléatoire.....	40
Figure 16 - Schéma XGBoost	42
Figure 17 - Schéma KNN	43
Figure 18 - KNN et comparaison d'erreurs.....	43
Figure 19 - SVM exemple étape 1	44
Figure 20 - SVM exemple étape 2	45
Figure 21 - SVM exemple étape 3	45
Figure 22 - Schéma réseau de neurones	46
Figure 23 - Magic Quadrant Gartner des plateformes de Machine Learning	47
Figure 24 - Forest Wave Forrester des plateformes de Machine Learning.....	48
Figure 25 - Taille du jeu de données	52
Figure 26 - Echantillon du jeu de données.....	52
Figure 27 - Description statistique du jeu de données.....	53
Figure 28 - Recherche de doublons.....	53
Figure 29 - Recherche de valeurs manquantes	54
Figure 30 - Découverte des valeurs distincts.....	54
Figure 31 - Suppression des colonnes sans informations	55
Figure 32 - Découverte de l'âge	55
Figure 33 - Découverte du revenu mensuel.....	56
Figure 34 - Découverte des années avec le manager actuel.....	56
Figure 35 - Découverte de la satisfaction des relations	57
Figure 36 - Découverte du domaine d'études.....	57
Figure 37 - Découverte du statut marital.....	58
Figure 38 - Découverte de la fréquence de voyage.....	58
Figure 39 - Découverte de l'emploi	59
Figure 40 - Découverte du département.....	60
Figure 41 - Matrice de corrélation	61
Figure 42 - Taille de l'échantillon	63
Figure 43 - Définition du jeu de test et d'entraînement	63
Figure 44 - Taille du jeu de test et d'entraînement.....	63

Figure 45 - Crédit à la création du modèle de régression logistique	64
Figure 46 - Entrainement et prédition (Régression logistique)	65
Figure 47 - Matrice de confusion (Régression logistique).....	65
Figure 48 - Métriques d'évaluation (Régression logistique).....	65
Figure 49 - Crédit à la création et entraînement du modèle (Random forest)	66
Figure 50 - Matrice de confusion (Random forest)	66
Figure 51 - Métriques d'évaluation (Random forest).....	67
Figure 52 - Paramètres du modèle Random forest.....	67
Figure 53 - Grille de nouveaux paramètres.....	68
Figure 54 - Recherche de la combinaison de paramètre la plus performante dans la grille.....	68
Figure 55 - Matrice de confusion (Random forest amélioré).....	69
Figure 56 - Métriques d'évaluation (Random forest amélioré).....	69
Figure 57 - Importance des variables selon Random forest.....	70
Figure 58 - Crédit à la création, entraînement et prédictions du modèle (XGBoost)	70
Figure 59 - Matrice de confusion et métriques d'évaluations (XGBoost)	71
Figure 60 - Importance des variables (XGBoost)	71
Figure 61 - Comparaison des différents modèles	72
Figure 62 - Performances du modèle choisi.....	73
Figure 63 - Processus de sélection des employés bénéficiant de l'effort de rétention	74

Remerciements

Je tiens tout d'abord à remercier M. BELHASSEN, pour m'avoir donné l'opportunité de travailler au cours de ces 2 ans au sein du pôle Applications & Projets chez Grant Thornton.

Aussi, j'adresse mes remerciements particuliers à M. NAUDIN et M. ZERZERI pour m'avoir accompagné tout au long de mon alternance.

Je tiens également à remercier chaleureusement Mme GUEDDA, M. NADRE et M. NADIH pour avoir pris le temps de m'accompagner dans les différentes missions que j'ai pu effectuer. Je les remercie d'autant plus pour leur patience et leur gentillesse à mon égard compte tenu des nombreuses questions que j'ai pu leur poser.

Par ailleurs, je remercie toute l'équipe pour l'expérience enrichissante et pleine d'intérêt qu'ils me font vivre au quotidien. Leur accompagnement m'a permis d'évoluer et d'acquérir de nouvelles compétences tant techniques que relationnelles.

Enfin, je tiens à remercier M. MAYAG enseignant à l'Université Paris Dauphine pour ses conseils et son suivi tout au long de cette année.

I. Introduction générale

Sommaire

I. Introduction générale.....	8
1.1 Mise en contexte	8
1.2 Problématique et intérêts	9
1.3 Plan du document.....	10

1.1 Mise en contexte

Les entreprises de tous secteurs et de toutes tailles font aujourd’hui face à un problème majeur : retenir leurs talents. Le schéma traditionnel qui destinait un employé à effectuer l’ensemble de sa carrière au sein de la même organisation ne semble plus être la norme. En moyenne, les cadres restent aujourd’hui 4 ans au même poste. Cela signifie qu’un travailleur lambda qui commence sa carrière professionnelle à 18 ans et qui la termine à ses 65 ans occupera environ 10 emplois différents au cours de sa vie. La réalité est encore plus frappante, puisque si on s’intéresse uniquement aux jeunes travailleurs, on s’attend à ce que ceux-ci occupent jusqu’à 15 emplois au cours de leur vie active [1].

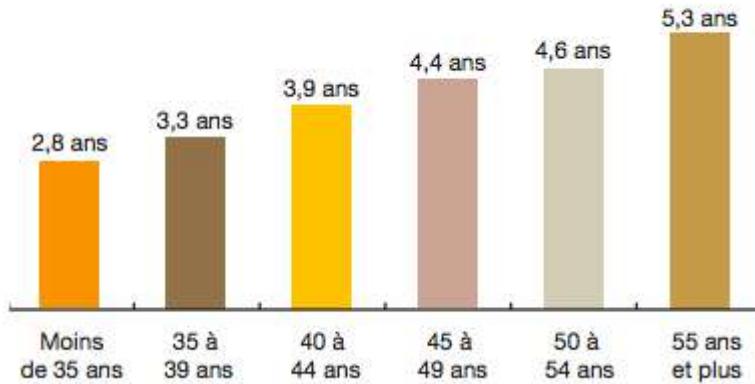


Figure 1 - Durée moyenne dans un poste selon l'âge du collaborateur

Les raisons de ces mouvements sont multiples : augmentation de salaire, perspectives d’évolution, meilleur équilibre entre la vie professionnelle et personnelle, bien-être au travail, pluralité des expériences, etc. Le phénomène touche l’ensemble des secteurs, même si certains domaines comme l’informatique apparaissent comme particulièrement sensibles. A titre d’exemple, Capgemini, une société leader dans le domaine du service numérique, a déclaré cette année avoir perdu près de 23,5% de son effectif en 1 an. Ce chiffre record ne traduit pourtant pas une mauvaise santé de l’entreprise puisque son bénéfice est lui aussi record avec une hausse de 21%. Toutefois, la société se retrouve aujourd’hui avec un tiers de ses collaborateurs disposant de moins d’un an d’ancienneté dans l’entreprise [4]. Cette situation entraîne une rupture entre l’organisation et sa culture et instaure ainsi une certaine distance vis-à-vis des salariés. Il devient difficile de fidéliser les collaborateurs et on cultive une incertitude sur les ressources disponibles à

moyen/long terme pour l'entreprise. Le lancement de gros projets devient plus difficile puisqu'on ne peut plus compter sur une stabilité totale des équipes.

Ces flux ne sont donc pas une bonne nouvelle pour les organisations. En effet, recruter représente un investissement important pour l'organisation. Bien que difficilement quantifiable, on estime que le coût de remplacement d'un salarié est d'en moyenne de 6 à 9 mois du salaire annuel de l'employé. Cette estimation peut également varier en fonction du rôle du salarié dans l'organisation. Ainsi on peut considérer que le remplacement pourrait coûter :

- 35% du salaire annuel pour un collaborateur nouvellement diplômé
- 150% du salaire annuel pour un collaborateur plus expérimenté
- Jusqu'à 300% voire 400% du salaire annuel pour le remplacement d'un collaborateur ultra-qualifié

Cette somme, très variable d'un profil à l'autre, comprend des coûts directs comme le salaire des recruteurs, les intermédiaires, la diffusion d'annonces...etc. mais également des coûts indirects comme la perte de performance, d'image, la formation du nouveau collaborateur, la passation de connaissance, l'impact sur le chiffre d'affaires de la société (dans le cas du remplacement d'une fonction dirigeante)...etc. [2]. On peut par ailleurs noter que lorsque qu'un recrutement échoue, c'est-à-dire que le collaborateur nouvellement recruté quitte l'organisation moins d'un an après son arrivé, le coût du recrutement s'envole. D'après le cabinet de recrutement Hays, selon les niveaux de responsabilité, la nature du poste ou la rémunération, le recrutement d'un collaborateur qui quitte l'organisation moins de 12 mois avant son arrivée engendre entre 45 000 et 100 000 euros de pertes [3]. Vous l'aurez compris, le processus de remplacement d'un collaborateur peut coûter beaucoup. L'entreprise a d'ailleurs tout intérêt à être capable de chiffrer avec précision à combien lui revient ce changement. En fonction de ce montant, elle peut ajuster sa stratégie en ayant recours à la rétention, à l'externalisation de certains processus ou à l'utilisation des logiciels spécialisés. Finalement, que peut faire l'entreprise pour réduire ses coûts ? Deux grands leviers sont à sa disposition :

- Recruter plus efficacement pour avoir des collaborateurs en phase avec l'entreprise
- Retenir ses collaborateurs afin de ne pas subir de départs

La combinaison de ces leviers doit permettre à l'entreprise de réduire ses dépenses tout en lui assurant de grandir. Dans la suite du document, nous nous intéresserons plus particulièrement à la problématique de rétention des collaborateurs et notamment à la manière de prévenir les départs.

1.2 Problématique et intérêts

Au regard de cet état des lieux qui touche particulièrement le secteur dans lequel je vais évoluer, j'ai cherché à comprendre comment l'organisation pouvait s'adapter à ces nouvelles tendances. Du point de vue de l'entreprise, la situation qui est la plus redoutée est celle du départ volontaire et imprévu d'un collaborateur. En effet, lors d'une démission, l'organisation perd sans prévenir une ressource opérationnelle. Elle se retrouve à devoir dénicher un remplaçant qualifié dans un délai défini (3 mois), à le former et à assurer le transfert de connaissance entre le salarié sortant et entrant. Cette période se traduit par une baisse de productivité qui coûte et retarde l'organisation. Dans certains secteurs, cette tâche peut s'avérer très ardue, puisqu'en cas de pénurie de ressources, il peut être très difficile de recruter un remplaçant qualifié. De plus, une fois la nouvelle ressource trouvée, cette dernière ne sera pas opérationnelle immédiatement. Alors pourquoi ne pas s'intéresser à faire en sorte de conserver les collaborateurs déjà dans l'organisation ? Plutôt que de recruter un nouveau collaborateur, pourquoi ne pas

faire en sorte de garder ceux-que l'on possède déjà ? Dans ce mémoire, j'ai voulu m'intéresser à la problématique du turn-over en entreprise et plus particulièrement à la manière d'estimer la probabilité d'un collaborateur à quitter l'organisation. L'objectif d'une telle analyse est de détecter les profils à risque afin de prendre des mesures pour s'assurer leur fidélité. Bien souvent, il est plus rentable d'écouter un collaborateur et de faire un pas vers lui plutôt que de risquer de le perdre.

Ainsi dans la suite de ce document, nous nous intéresserons à la manière dont l'analyse de données peut permettre d'anticiper le départ d'un collaborateur. L'objectif est d'être en mesure de mettre en place des actions ciblées afin de réduire le turn-over d'une organisation. Cette problématique fait écho dans de nombreuses organisations qui font aujourd'hui face à la « grande démission ». L'analyse des effectifs et de leur flux est devenue une composante essentielle des systèmes d'informations des ressources humaines. Elle peut permettre à l'organisation de mieux comprendre ses salariés afin de répondre au mieux à leurs attentes. Au sein de cette problématique, je vais pouvoir mettre en pratique les outils et méthodes étudiés à l'université afin d'avoir un réel impact dans la prise de décision d'une société. Alternant au sein d'un cabinet d'audit, le turn-over est au cœur des problématiques RH. Comprendre ces flux et les anticiper pourrait apporter un second souffle aux équipes sans cesse impactées par le départ d'un employé.

1.3 Plan du document

Dans une première partie de ce document, nous réaliserons un état de l'art des solutions aujourd'hui utilisées pour travailler l'anticipation de départs de collaborateurs. Nous étudierons les outils et métriques utilisés par les ressources humaines avant de nous intéresser au domaine de l'analyse prédictive. Ainsi dans un second temps de l'état de l'art, nous étudierons ensemble comment le Machine Learning peut apporter aux ressources humaines les clefs du problème. Nous étudierons les principes de fonctionnement du ML¹ ainsi que les manières dont ont été pensés les différents algorithmes. L'objectif est de donner au lecteur une vision générale des solutions qui existent à ce jour. Dans la seconde partie du document, nous tenterons ensemble de répondre à notre problématique par le scope d'une entreprise fictive. Grâce à des données fournies par les data scientist d'IBM, nous réaliserons un projet d'analyse prédictive sur le départ des employés.

¹ Machine Learning

II. Anticiper le départ d'un collaborateur

Sommaire

II. Etat de l'art.....	Erreur ! Signet non défini.
2.1 Le turn-over, un indicateur clé	11
2.1.1 Définition.....	11
2.1.2 L'analyse du taux de turnover.....	12
2.1.3 Les causes et les conséquences.....	14
2.2 Les solutions RH	15
2.2.1 Les métriques	15
2.2.2 Les système d'informations des ressources humaines	17
2.3 Anticiper les départs : un problème de classification	20
2.3.1 Le Machine Learning	20
2.3.2 Les problèmes de classification.....	27
2.3.3 Les outils informatiques	46

2.1 Le turn-over, un indicateur clé

2.1.1 Définition

Intéressons-nous tout d'abord au taux de turnover. Aussi appelé taux de rotation du personnel, le taux de turnover est un indicateur clé de l'organisation. Selon une étude réalisée en par M. Andrée Laforgue en 2014, 94% des entreprises québécoises reconnaissent le calculer [5]. La société SBA Compta déclare qu'il est même "de loin l'un des meilleurs indicateurs de la réussite à long terme" d'une l'entreprise [6]. Il permet entre autres, d'avoir une idée sur le climat social de l'organisation et est généralement représentatif de l'ambiance au sein d'un groupe de travail ou d'une entreprise. Indicateur phare, le turn-over permet finalement d'analyser les rotations du personnel :

$$\text{Taux de turn - over} = \frac{(\text{Nombre de départs sur la période} + \text{Nombre d'arrivées sur la période})/2}{\text{Nombre d'employés au 1er jour de la période}} * 100$$

Afin de le calculer et de correctement l'interpréter, il est nécessaire d'en comprendre au préalable les différentes composantes :

- La période : Il est important de définir clairement l'intervalle de temps sur lequel on souhaite calculer son taux de turn-over. Est-ce qu'il s'agit d'un turnover mensuel ? Trimestrielle ? Annuel ? Est-ce que l'on parle de l'année civile ou de l'année fiscale ?
- Le nombre d'employés au 1er jour de la période : Une fois la période explicitement définie, il nous faut calculer l'effectif au 1er jour de la période. Il faut alors se demander quelle population

souhaite-t-on suivre ? S'agit-il uniquement des employés en CDI ? Désire-t-on suivre l'effectif d'un bureau en particulier ? Les prestataires font-ils partie de l'effectif ?

- Le nombre de départs sur la période : En fonction de la population identifiée lors du calcul de l'effectif, il nous faut à présent définir quelles sont les sorties que nous souhaitons considérer. L'organisation peut choisir de s'intéresser aux départs volontaires, involontaires ou les deux
- Le nombre d'arrivées sur la période : De la même manière que pour les départs, il nous faut clarifier la définition d'arrivée.

Ainsi, si l'on prend l'exemple d'une organisation disposant au 1er janvier de 1000 salariés et ayant eu 100 départs et 150 arrivées au cours de l'année, celle-ci se verra dotée d'un turnover de 12,5%. Cela signifie que 12,5% de l'effectif de l'entreprise a donc été renouvelé sur l'année. Ce taux peut être calculé à différentes échelles comme celui de l'organisation, du pôle, du métier, de l'équipe, du type de contrat... etc. On peut également considérer différentes périodes. On peut vouloir s'attarder sur le turn-over volontaire traitant des départs subis par l'entreprise ou encore le turn-over involontaire concernant les départs provoqués par l'organisation. Ainsi, on obtient différentes métriques permettant d'ajuster les stratégies RH² dans l'organisation. Ces stratégies peuvent s'appliquer au niveau global ou bien comporter des composantes spécifiques au sein des différentes directions métiers.

2.1.2 L'analyse du taux de turnover

Bien que le calcul soit relativement simple, l'analyse du taux de turnover requiert un peu plus d'attention. Il est communément admis qu'un turnover idéal n'existe pas. Il est important de comprendre qu'en fonction du secteur d'activité ou de la typologie de l'entreprise, le taux de turn-over peut sembler anormalement élevé sans pour autant traduire un mauvais climat social.

Ainsi on ne peut pas analyser de la même manière un turn-over de 15% dans une entreprise de service numérique que dans une administration publique. Pareillement, au sein de la même organisation, un turnover de 15% peut se révéler très élevé pour une équipe mais tout à fait acceptable pour une autre. Pour que l'analyse soit la plus fine et la plus juste possible, il est donc nécessaire de contextualiser la situation. Pour ce faire, il faut comparer son turnover au secteur d'activité, et parfois même au poste. Une bonne connaissance du marché est finalement primordiale pour correctement interpréter un taux de turn-over :

² Ressources Humaines

		POPULATION ET EMPLOI			CHIFFRES CLÉS 2019										
		Mouvements de main d'œuvre par secteur d'activité en 2017													
		Taux de rotation	Taux d'entrée	Taux d'entrée en CDD	Taux d'entrée en CDI	Part des CDD dans les embauches	Taux de sortie	Taux de fin de CDD	Taux de démission	Taux de licenciement économique	Taux de licenciement non économique	Taux de ruptures conventionnelles	Taux de fin de période d'essai	Taux de départs en retraite	Taux de sortie pour autres motifs (décès, accident...)
4	Fabrication de denrées alimentaires, de boissons et de produits à base de tabac	76,1%	74,8%	47,3%	27,5%	63,3%	77,3%	41,5%	14,2%	0,6%	5,9%	3,1%	9,9%	1,6%	0,6%
	Cokéfaction et raffinage	ns	ns	ns	ns	ns	ns	ns	ns	ns	ns	ns	ns	ns	ns
	Fabrication d'équipements électriques, électroniques, informatiques ; fabrication de machines	14,8%	15,0%	6,0%	8,9%	40,2%	14,6%	4,2%	3,6%	0,6%	1,3%	1,3%	0,9%	1,8%	0,9%
	Fabrication de matériels de transport	15,8%	15,5%	3,8%	11,7%	24,5%	16,1%	2,7%	6,9%	0,7%	1,9%	0,5%	0,4%	2,3%	0,7%
	Fabrication d'autres produits industriels	26,3%	26,6%	15,1%	11,5%	56,8%	25,9%	11,7%	5,3%	0,7%	2,4%	1,9%	1,7%	1,7%	0,5%
	Industries extractives, énergie, eau, gestion des déchets et dépollution	17,3%	16,8%	7,8%	9,0%	46,6%	17,8%	5,7%	3,8%	0,2%	2,5%	1,3%	1,1%	2,6%	0,8%
	Construction	28,7%	30,2%	12,2%	18,0%	40,3%	27,3%	7,7%	8,6%	0,3%	3,9%	2,2%	2,8%	1,2%	0,4%
	Commerce ; réparation d'automobiles et de motocycles	50,1%	50,7%	29,7%	21,0%	58,6%	49,6%	23,6%	10,2%	0,6%	5,1%	2,4%	6,2%	0,9%	0,7%
	Transports et entreposage	38,4%	39,4%	26,4%	13,0%	66,9%	37,4%	22,3%	4,6%	0,2%	4,0%	1,1%	2,9%	1,8%	0,4%
	Hébergement et restauration	297,2%	296,7%	248,4%	48,3%	83,7%	297,8%	243,3%	24,1%	0,2%	8,5%	2,9%	16,8%	0,8%	1,1%
	Information et communication	205,9%	208,3%	186,6%	21,7%	89,6%	203,4%	182,9%	10,0%	0,4%	2,6%	2,5%	3,8%	0,6%	0,6%
	Activités financières et d'assurance	23,8%	24,3%	14,2%	10,1%	58,5%	23,3%	11,6%	4,7%	0,2%	1,7%	1,3%	1,7%	1,8%	0,4%
	Activités immobilières	51,3%	53,0%	35,8%	17,1%	67,6%	49,5%	32,1%	7,2%	0,1%	2,6%	2,5%	2,9%	1,7%	0,4%
	Activités scientifiques et techniques ; services administratifs et de soutien	132,7%	134,4%	113,8%	20,5%	84,7%	131,1%	108,4%	8,8%	0,3%	4,6%	2,6%	4,5%	1,0%	0,9%
	Administration publique, enseignement, santé humaine et action sociale	257,5%	258,3%	241,3%	17,0%	93,4%	256,8%	234,8%	10,1%	0,2%	3,7%	2,0%	3,9%	1,7%	0,5%
	Autres activités de services	304,0%	304,9%	289,3%	15,6%	94,9%	303,0%	282,9%	8,8%	0,2%	3,4%	2,5%	3,7%	1,1%	0,4%
	Ensemble	122,3%	123,3%	104,2%	19,1%	84,5%	121,3%	99,4%	9,1%	0,4%	4,0%	2,1%	4,5%	1,3%	0,6%

Source : Dares, MMO, données rétropolées avant 2016. Champ : île-de-France, établissements de 10 salariés ou plus du secteur privé (hors agriculture, hors intérim), champ MMO.
Lecture : pour 100 salariés employés dans les établissements franciliens du secteur de l'hébergement et de la restauration de 10 salariés ou plus, 248,4 embauches se font en CDD.

Figure 2 - Turn-over par secteur d'activité en 2019

La figure 2 met en évidence que le turn-over ne s'analyse pas de la même manière dans le secteur de l'hôtellerie / restauration que dans la fabrication de matériel de transport.

De manière générale, on s'accorde à dire qu'un taux de turnover faible est synonyme d'efficacité. La fidélisation des collaborateurs permet de bénéficier sur le long terme d'un effet d'apprentissage. Les opérations sont réalisées mieux et plus vite puisque le collaborateur connaît parfaitement son travail. On assiste alors à un gain de productivité. La fidélisation conduit l'entreprise à développer sa culture d'entreprise et lui permet de mener des projets sur le long terme avec des salariés investis [7]. Il peut être intéressant d'évaluer le succès de sa stratégie de fidélisation par rapport à ses concurrents en comparant son taux de turnover aux normes du secteur. On peut ainsi avoir une opinion plus objective sur l'organisation et ses performances.

Attention toutefois, même si à première vue un turnover bas semble être l'objectif, cela ne peut toujours s'avérer totalement vrai. Prenons l'exemple de deux sociétés ayant les entrées et sorties suivantes :

Société	Effectif au 1er janvier	Départ sur la période	Arrivées sur la période	Taux de turn-over
A	1000	250	0	$\frac{(250 + 0)}{1000} * 100 = 12,5\%$
B	1000	0	250	$\frac{(0 + 250)}{1000} * 100 = 12,5\%$

Chacune de ses sociétés dispose d'un turn-over de 12,5%. Or leurs situations ne sont pas du tout les mêmes. En effet la société « A » semble être dans une posture délicate : un quart de son effectif a quitté l'entreprise durant l'année. Il serait par exemple intéressant d'affiner notre analyse en discrétilisant ces départs et en essayant de comprendre s'il s'agit de départs volontaires ou non. De l'autre côté, la société « B » semble en pleine croissance.

Ainsi on comprend que l'indicateur du turn-over n'est pas parfait et qu'il est primordial de l'analyser en fonction du contexte de l'organisation. L'indicateur seul peut nous mener à tirer des conclusions hâtives, il est donc primordial de l'analyser au regard d'autres métriques. Le turn-over n'est finalement pas un indicateur parfait, et il mérite d'être analysé avec précaution.

2.1.3 Les causes et les conséquences

Dans cette section nous allons rapidement nous pencher sur les causes et les conséquences d'un turn-over élevé. L'objectif est de mieux comprendre le contexte dans lequel se pose la problématique afin d'être plus à même d'interpréter nos résultats. Il est également intéressant d'entrevoir ici les données qui pourraient être suivies dans le cadre d'une collecte par l'entreprise. Vous l'avez compris, les salariés sont aujourd'hui mobiles et n'hésitent pas à changer d'organisation. Mais quelles en sont les causes ? Qu'est ce qui incite un collaborateur à quitter son poste ? Le journal Le Figaro [9], l'entreprise Wuro [10] (spécialisé dans la gestion d'entreprise) ainsi que QualtricsXM [8] (entreprise d'analyse d'expérience client et collaborateurs) nous apportent quelques réponses. Parmi les causes les plus importantes, on peut citer :

- Le mal-être au travail dû à un mauvais climat social (mésentente entre les collaborateurs, avec la direction, mauvaise gestion et/ou mauvaise communication au sein de l'établissement...etc.)
- Le déséquilibre entre vie professionnelle et vie personnelle (peu de flexibilité, gros volume horaire...etc.)
- Le manque de perspectives d'évolution
- Le salaire et des avantages jugés trop faibles

On peut également retrouver le manque de reconnaissance professionnelle vis-à-vis de la hiérarchie ou les mauvaises conditions de travail (absence de télétravail, horaires fixes, surcharge, stress...). Il est intéressant de remarquer que l'entreprise a un véritable pouvoir d'action sur ces causes de départ. Elle a la capacité de prendre des initiatives afin de remédier à ces éléments. Il ne s'agit pas d'événements exceptionnels vis à vis desquelles elle serait démunie. Elle possède donc les cartes pour lutter contre ces démissions, encore faut-il qu'elle en ait conscience. On remarque que ces causes d'insatisfaction n'apparaissent généralement pas du jour au lendemain. Elles sont le produit d'une longue frustration qui n'aura pas été détectée par les encadrants. Il est donc important pour l'entreprise d'être au courant de l'atmosphère et des conditions de travail au sein de ses équipes afin d'éviter le roulement répété et non maîtrisé de ses effectifs.

On peut noter que dans certains secteurs comme l'informatique, le juridique et comptable, et l'ingénierie-R&D, les profils sont extrêmement convoités. Le turnover risque donc d'être plus important pour ces secteurs et métiers en tension, puisque ces candidats seront particulièrement courtisés pour être débauchés.

Les conséquences d'un turn-over élevé au sein d'une organisation sont alors multiples. En plus des coûts directs que nous avons évoqués en introduction, il existe des coûts indirects qui vont atteindre la santé de l'entreprise. Parmi les coûts directs on retrouve notamment :

- Les coûts liés directement aux départs : charges administratives, entretiens, pot de départ
- Les dépenses liées au recrutement d'un nouveau collaborateur : publication de l'offre d'emploi, communication, négociations de salaire
- Charges administratives relatives à l'intégration du nouvel employé
- Coût de l'éventuelle formation du personnel nouvellement recruté [8]

L'ensemble de ces éléments se chiffre et en fonction de l'établissement et de la fonction, cela peut coûter très cher. En plus de ces frais, il existe de nombreux coûts dissimulés auxquels l'entreprise devra faire face. Plus difficiles à estimer financièrement, on peut par exemple citer :

- La perte de productivité avant le départ du salarié due à sa baisse de motivation
- La perte de productivité à l'arrivée du nouveau collaborateur : le temps d'adaptation de l'employé au poste et de l'équipe au nouvel arrivant
- La perte de temps pour la branche RH et le manager due à l'élaboration de la fiche de poste, aux recherches de candidats et tri entre ces derniers, aux entretiens d'embauche et à la sélection du candidat retenu
- L'impact sur le moral et l'environnement de travail de l'équipe
- La perte de motivation si, par exemple, les anciens collègues apprennent que leur confrère les a quittés pour un poste plus rémunéré et avec plus de responsabilités et d'avantages [8]

On peut également noter qu'un turn-over élevé peut affecter la culture et l'image de l'entreprise. Il peut donc lui porter préjudice pour des recrutements futurs. On peut même parler d'effet « boule de neige » puisque plus une entreprise voit ses employés la quitter, moins les autres ont envie d'y postuler. En effet il est courant de demander en entretien pourquoi le poste est ouvert et une démission n'est jamais bon signe pour un candidat.

Il faut garder à l'esprit que ces coûts sont à multiplier par le nombre d'individus qui quittent l'organisation. L'accumulation de ces frais peut être préjudiciable pour l'ambiance de travail ainsi que pour la santé financière d'une société. Il est donc primordial pour les dirigeants de garder le ratio de turnover à l'œil afin d'être en mesure de réagir vite en cas d'alerte.

Pour conclure, nous avons vu dans ce chapitre que le turn-over est un indicateur largement calculé dans les organisations mais qu'il faut l'analyser avec soin. Il est important de l'étudier au regard du contexte de l'organisation et des objectifs de l'entreprise. Nous avons vu que les causes du turn-over identifiées ne sont pas insurmontables. Il s'agit d'une balance sur laquelle on peut mettre le consentement du salarié d'un côté et les efforts pour y répondre de l'entreprise de l'autre. Enfin nous avons vu qu'un turn-over élevé pouvait coûter cher à l'entreprise aussi bien en termes d'argent et de productivité qu'en termes de réputation et de rayonnement. Attention toutefois, un turn-over raisonnable n'est pas mauvais pour l'entreprise, il garantit l'internalisation de nouvelles compétences et de nouvelles visions du monde qui ne peuvent qu'enrichir l'organisation. Il ne faut pas chercher à retenir tout le monde mais tout mettre en œuvre pour garder les collaborateurs qui ont le plus de valeur pour l'entreprise.

2.2 Les solutions RH

2.2.1 Les métriques

Dans le chapitre précédent nous avons largement étudié le turn-over. Bien qu'extrêmement répandu, il ne constitue pas l'unique manière d'analyser la stratégie de rétention de collaborateurs d'une organisation. Dans ce chapitre, nous verrons d'autres métriques qui peuvent venir compléter le taux de turn-over dans l'optique d'obtenir une vision plus large du contexte. L'objectif n'est pas de fournir une liste exhaustive de tous les KPI³ utilisés dans les ressources humaines mais bien de choisir ceux qui sont liés aux objectifs que nous avons définis, à savoir réduire le taux de roulement du personnel.

Les indicateurs RH sont plus que de simples chiffres. Leur but est de quantifier les objectifs et de mesurer le progrès. Ils montrent à quel point nous sommes proches de nos objectifs, nous permettant de modifier notre stratégie ou d'envisager de nouvelles actions. Les KPI nous fournissent finalement des données à interpréter dont l'information nous permet de prendre des décisions et d'ajuster notre stratégie. George T. Doran, dans son livre "There's a S.M.A.R.T. way to write management's goals and objectives" propose un

³ Key Performance Indicator ou indicateurs de performance

moyen mnémotechnique pour décrire ses objectifs. Cette méthode a par la suite été reprise et appliquée aux indicateurs de performances. Ainsi un indicateur se doit d'être :

- **Spécifique** : par exemple, un KPI RH doit se référer à une action ou à une tâche très spécifique comme économiser le temps passé sur chaque recrutement.
- **Mesurable**
- **Atteignable** : les objectifs fixés pour chaque indicateur de performance doivent être réalistes et adaptés aux conditions dont on dispose.
- **Réaliste (ou pertinent)**: le défi relevé devra motiver le plus grand nombre de participants afin d'éviter au mieux l'abandon
- **Temporel** : doit porter sur une période spécifique et être revu périodiquement [11]

Dans le chapitre précédent nous avons étudié le turn-over de manière générale. Cependant au regard de notre problématique nous nous intéresserons à des turn-over plus affinés. Si on s'intéresse de plus près aux départs de salariés, on peut les classer en 2 grandes catégories :

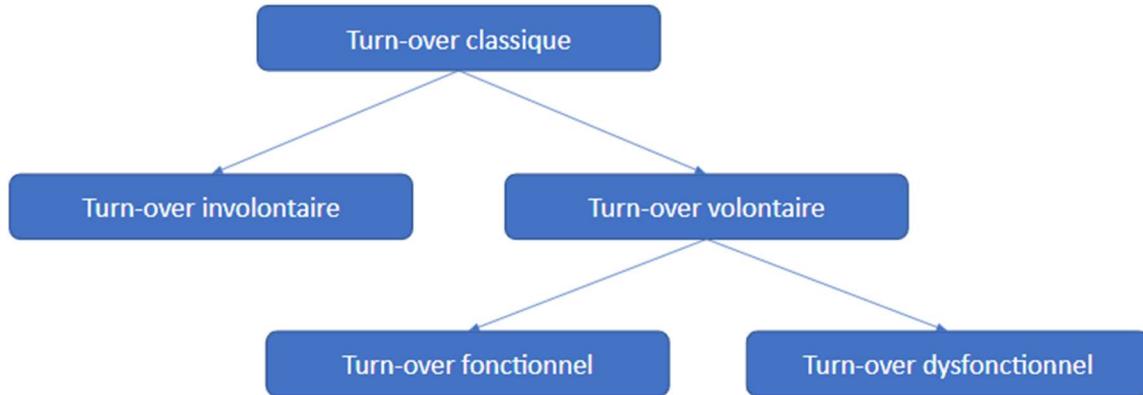
- Les départs volontaires, c'est à dire ceux dont la décision a été prise par le salarié
- Les départs involontaires sont ceux dont la décision est prise par l'employeur. Ces départs comprennent des actions telles que le licenciement, la mise à pied, le retranchement, la retraite.

Pour l'organisation, les départs involontaires ne sont pas subis. Ils résultent d'une prise de décision réfléchie pour lesquelles l'entreprise a déjà pris ses prédispositions. En revanche, dans le cas d'un départ volontaire, l'organisation peut se retrouver prise de cours. Elle va devoir gérer cet événement qu'elle n'avait pas prévu. Il s'agit donc d'une charge de travail supplémentaire non planifiée. L'organisation a donc tout intérêt à éviter ce genre de départ. Ce sont donc ceux-ci que nous chercherons à prévoir en priorité. On cherche donc à minimiser notre turn-over volontaire.

D'autre part, comme nous l'avons évoqué plus haut, tout départ n'est pas mauvais en soi. Parfois il est bénéfique pour le salarié comme pour l'organisation de cesser de travailler ensemble. Ainsi, du point de vue de l'organisation, on distingue deux turn-over volontaires distincts :

- **Le turnover fonctionnel** qui s'intéresse aux employés dit "peu performants et qui quittent l'organisation". Exemple : Un employé qui n'atteint pas toujours les objectifs, un employé qui arrive toujours en retard aux réunions ou encore un employé qui met les autres maux à l'aise et sape le moral de l'équipe. Lorsque ces employés démissionnent volontairement, il s'agit de roulement fonctionnel qui est bénéfique pour l'entreprise
- A l'inverse, **le turnover dysfonctionnel** s'illustre lorsque les employés les plus performants quittent l'entreprise. [12]

Finalement, on comprend assez facilement que l'entreprise va chercher à encourager le turnover fonctionnel et décourager le turnover dysfonctionnel. Son objectif est donc double, elle cherche à minimiser les départs volontaires dysfonctionnels tout en maximisant les départs volontaires fonctionnels.



Notre objet d'étude porte finalement sur le **turn-over dysfonctionnel lié aux départs volontaires**. Les actions RH devront alors se concentrer sur les départs évitables de ces ressources critiques.

Une seconde métrique intéressante à étudier dans notre problème est le taux de rétention employé. Cet indicateur a pour objectif de mesurer la manière qu'a l'entreprise de conserver les employés qu'elle recrute. Il est un premier indicateur de l'engagement et la fidélisation de vos collaborateurs. Il se calcule en divisant le nombre de salariés toujours en poste à N+1 par le nombre d'embauches à N.

$$\text{Taux de rétention} = \frac{\text{Nombre de salariés toujours en poste à } N+1}{\text{Nombre de personnes recrutées à } N} * 100$$

Il est donc le pendant du turn-over puisqu'au lieu de mesurer les départs, on mesure les employés qui restent. Ainsi on peut être capable de découvrir des profils de collaborateurs qui seraient plus à même de rester longtemps dans l'entreprise. Ces collaborateurs pourraient alors être privilégiés dans les phases de recrutement. On pourrait mettre l'accent sur ces derniers afin de minimiser les risques qu'ils partent de l'entreprise.

Enfin la performance à N+1 ainsi que le degré de satisfaction du manager [13] sont deux indicateurs qui pourraient être très intéressants pour repérer ces fameuses ressources critiques qui nous font défaut dans le turn-over dysfonctionnel. De la même manière que pour le taux de rétention, on pourrait ici aussi tenter de déterminer le profil des employés qui travaillent bien afin d'en brosser le portrait. Ce profil pourrait être transmis aux recruteurs afin de les aider dans leurs recherches de candidats.

2.2.2 Les systèmes d'informations des ressources humaines

Au regard des métriques RH qui sont à visée descriptive, nous allons dans ce chapitre, nous intéresser à savoir comment les ressources humaines tentent aujourd'hui de prévoir le départ d'un collaborateur. Nous nous intéresserons aux différents niveaux qui peuvent exister et comment l'informatique décisionnelle permet d'améliorer la prise de décision.

2.2.2.1 La technique du "doigt mouillé"

Afin de déterminer la probabilité de départ d'un collaborateur, une première approche naturelle consisterait simplement à le "pressentir". Il s'agirait, par le biais d'indices, de ressentis, de déceler la volonté du salarié à quitter l'entreprise. Cette approche de bas niveau peut par exemple être adoptée dans les microstructures où les collaborateurs se connaissent bien ou dans les organisations ayant moins de budget

comme les associations ou les écoles. En effet, le problème de départ volontaire peut être étendu à toute type d'organisation. Toutefois, concentrons-nous un instant sur les petites entreprises. Du fait du faible nombre de salariés, on peut constater que les ressources humaines peuvent alors être plus proches des salariés et plus facilement déceler les risques de départ de l'un d'eux. Cependant, développer un climat de confiance propice au partage par les collaborateurs prend du temps. Finalement cette solution a l'avantage de ne coûter que la vigilance et la bienveillance du responsable des ressources humaines. Elle a en revanche l'énorme inconvénient d'être très subjective. De plus, si l'employé n'est pas suivi d'assez près, il serait quasiment impossible pour le responsable des ressources humaines de déceler une potentielle démission. Il est donc nécessaire que ce dernier soit proche des salariés. Ce genre d'approche ne peut pas être adopté dans les grandes entreprises du fait du nombre de salariés à suivre.

2.2.2.2 Le SIRH, un premier pas vers l'aide à la décision

Nous l'avons vu, simplement suivre ses salariés n'est pas une option envisageable pour les moyennes et grandes entreprises. Ces dernières ont donc investi dans l'acquisition de système d'information dédié aux ressources humaines. Ces SIRH⁴ sont des logiciels destinés au stockage et au traitement des données relatives aux employés d'une entreprise permettant de fluidifier le travail des ressources humaines et d'améliorer leur productivité. Le SIRH remplit des fonctions telles que la gestion du recrutement, le suivi des candidats, la gestion du temps et l'évaluation des performances. En bref, il englobe tout ce dont le groupe de gestion des ressources humaines a besoin pour mener à bien ses missions. Cette solution constitue un excellent moyen de construire une base de données intelligente, facilitant grandement le travail des services de ressources humaines. Grâce à de tels outils, nous sommes dorénavant capables de suivre de plus grandes masses de collaborateurs. L'acquisition d'un tel outil s'inscrit dans la stratégie globale de numérisation des organisations. Il est recommandé d'utiliser ces solutions dès lors que l'organisation dépasse 25 collaborateurs. En effet, au-delà il devient difficile de réunir, actualiser et suivre les données de chaque salarié [14].

Ces solutions sont donc un premier pas vers l'aide à la décision. Ils permettent de centraliser les informations d'un collaborateur au sein d'un même outil. Attention toutefois, le SIRH n'a pas vocation à remplacer le rôle des responsables des ressources humaines. L'objectif est plutôt de faciliter leur travail, tout en leur permettant de se concentrer sur d'autres tâches plus valorisantes. Ces solutions permettent de réaliser des analyses descriptives de l'environnement. Elles permettent de connaître le taux de turnover ou le taux de rétention au sein de l'organisation. Toutefois, cette vision globale n'est généralement qu'une partie de la réponse. En effet, les résultats ne sont pas d'une grande aide pour anticiper les comportements au cas par cas. Vous savez que votre taux de rétention est bas, vous ne savez pas pourquoi. Il s'agit d'un excellent outil descriptif de la situation au sein de votre organisation. Elles laissent les décideurs faire leurs propres analyses et leurs propres conclusions. L'emploi des SIRH est aujourd'hui incontournable. Il existe une multitude d'acteurs sur ce marché tel Cegid, Taleo par Oracle, Altays ou TalentSoft. La mise en place de tels systèmes coûte de l'argent et il revient donc à l'organisation de décider d'investir ou non dans ces fonctions supports à l'activité.

Finalement le SIRH n'est qu'une étape sur notre chemin nous permettant de prévoir le départ d'un salarié. Il donne des indications précieuses sur le contexte de l'organisation. Il permet de structurer la collecte et le stockage de données pour les mettre à disposition des décideurs. Nous devons dorénavant passer d'une analyse descriptive à une analyse prédictive.

2.2.2.3 L'analytique RH et le Machine Learning

Avec le SIRH, l'entreprise est dans une dimension descriptive. Cherchons comment la faire passer dans une dimension prédictive. L'analytique RH est un processus d'analyse des données des ressources humaines permettant d'améliorer la performance de l'organisation. Il s'agit d'une démarche statistique qui consiste à extraire des informations utiles à partir de différentes données de l'organisation. L'idée est de mobiliser

⁴ Systèmes d'Information des Ressources Humaines

les chiffres pour mieux connaître l'entreprise. Cette démarche proche des données est une réelle cassure avec la culture qualitative des ressources humaines et leur goût des relations humaines. Pour ces derniers, il est difficile de considérer un employé comme un chiffre et de prendre des décisions en conséquence. L'humain est important et il ne faut pas le négliger. Toutefois avec l'explosion des volumes et de la complexité des données, il devient difficile de s'en passer. Les bénéfices qu'offrent ces solutions ne sont pas négligeables et c'est en partie pourquoi on assiste au développement de cette branche. En effet, selon Mercer plus de 77% des entreprises interrogées prévoient de développer leurs compétences en analytique RH [16]. Comme le secteur du marketing l'a fait avant lui pour analyser les besoins client, les ressources humaines travaillent aujourd'hui à analyser les besoins salariés. On cherche donc à développer des modèles statistiques capables de répondre à des questions complexes telles que "l'absentéisme affiche-t-il des variations saisonnières?", "pourquoi le taux de démission non désiré est-il plus important dans telle ou telle population" ou encore "les salariés embauchés avec un certain niveau de qualification ont-ils de meilleures performances?". L'analytique RH semble être alors la solution capable de nous livrer, sur un plateau d'argent, la réponse à toutes nos interrogations. Toutefois, pour qu'une analyse soit pertinente, elle doit réunir 3 éléments :

- Tout d'abord, **les données doivent exister.** Cela paraît évident mais la réalité ne l'est pas toujours autant. Il arrive que l'on désire ardemment faire une analyse à partir de données qui ne sont actuellement pas collectées ou pas accessibles. De plus, elles doivent être propres, à jour et disponibles, ce qui n'est pas systématiquement le cas. Ces dernières ne sont pas toujours correctement structurées dans de beaux SIRH et l'on doit parfois partir à leur recherche dans l'entreprise. Il arrive même que l'information que l'on recherche n'existe pas puisque l'on n'a pas pensé à la collecter ou que l'on n'en a pas le droit.
- Ensuite, **les modèles statistiques doivent exister.** Ce point ne pose généralement pas de problème tant la littérature statistique est riche.
- Enfin, **les ressources humaines doivent avoir les compétences** pour gérer ces données et les modèles d'analyse qui les exploitent. L'intervention d'un professionnel RH compétent est la clé. Il y a donc un enjeu pour ces structures de développer des compétences et une culture de la collecte et de l'analyse de données. On assiste trop souvent dans ces métiers à une incompréhension des problématiques data conduisant à produire de la donnée de faible valeur ajoutée.

L'analytique de données est une indication supplémentaire pour le décideur. Il faut toutefois faire attention à ne pas succomber à l'hypnose du chiffre et garder son esprit critique vis-à-vis de l'analyse. Parfois les données ne représentent pas le monde de manière exhaustive, parfois le modèle ne correspond pas bien, parfois le résultat est mal interprété. Il faut donc se souvenir que l'analytique RH n'est pas une formule magique qui va nous permettre de guérir tous nos maux mais bien d'un outil supplémentaire à notre disposition. Il est important de garder un esprit critique et de toujours se demander si le résultat que nous obtenons correspond bel et bien à ce que nous cherchions.

Si on revient sur la problématique de prédiction de départ d'un collaborateur, l'analytique RH semble tout à fait indiqué. Brook Holtom, professeur de management à l'université de Georgetown, et David Allen, professeur à Warwick business school ont développé grâce à l'analytique RH et notamment grâce au Machine Learning un indice permettant de prédire en temps réel la probabilité qu'une personne considère une offre extérieure et quitte éventuellement l'entreprise. Cet indicateur appelé le « turnover propensity index for individuals (TPI) » permet de classer graduellement ces profils selon leur propension à accepter une nouvelle opportunité d'emploi [18]. Les professeurs ont travaillé avec une entreprise spécialisée dans le recrutement de talent afin de collecter un large échantillon sur lequel travailler. Ils ont ensuite collecté des données sur des employés comme leur nombre d'emplois antérieurs, leur niveau d'éducation, leur sexe, leur lieu de vie, leurs compétences, etc. Grâce au Machine Learning, ils ont réussi à déterminer dans

quelles mesure un employé serait réceptif à de nouvelles opportunités. On pourrait donc supposer que plus le TPI de l'individu est élevé, plus il a de chances de considérer une offre d'embauche et donc plus il a de chance de démissionner.

Afin de tester leur modèle, ils ont décidé d'envoyer des invitations par courriel à un échantillon de 2 000 personnes. Ces personnes ont été identifiées par l'algorithme en fonction de leur probabilité d'ouvrir une invitation à consulter des emplois disponibles, adaptés à leurs compétences et intérêts spécifiques. Ils ont ensuite mesuré le nombre de clics et se sont aperçus que, parmi les personnes qui avaient ouvert le courriel, celles qui étaient considérées comme les plus susceptibles d'être réceptives par l'algorithme, l'étaient bel et bien. Ils ont finalement réalisé que les personnes ayant un score TPI élevé étaient « 63% plus susceptibles de changer d'emploi » et « 40% plus susceptibles de démissionner », par rapport aux personnes ayant un score TPI faible [19].

Ces travaux académiques ont donc montré la puissance de l'analytique RH et du Machine Learning. On peut d'ailleurs noter que les organisations ont un avantage énorme sur les chercheurs externes pour développer leur propre TPI puisqu'elles peuvent utiliser des gros volumes de données internes. Elles disposent notamment de facteurs comme l'intégration au travail ou les opportunités de carrière qui sont souvent difficiles d'accès dans les travaux académiques. Finalement, selon Brook Holtom et David Allen, les entreprises qui investissent dans la collecte et l'analyse de ces données pourront engager de manière proactive les employés les plus précieux. Elles pourront ensuite réaliser des entretiens réguliers avec ces derniers et voir comment il est possible de les faire rester. [18]

2.3 Un problème de classification

2.3.1 Le Machine Learning

Nous l'avons vu avec les travaux de Holtom et Allen, le Machine Learning pourrait être la clé de la prédiction du turn-over. Dans ce chapitre, nous allons nous intéresser à savoir ce qu'est réellement le Machine Learning et comment l'utiliser afin de répondre à des problématiques métiers. Nous nous intéresserons dans un premier temps à définir le Machine Learning puis nous tenterons de comprendre les logiques qui se cachent derrière les différents algorithmes.

2.3.1.1 Que signifie « Machine Learning » ?

« Machine Learning », deux mots magiques que l'on entend aujourd'hui beaucoup et qui sont censés répondre à toute nos interrogations, reste pour une beaucoup une boîte noire. Utilisé dans de nombreux domaines, on distingue deux grands types d'applications :

- Tout d'abord, la **recherche théorique** qui consiste à rendre une machine autonome. On cherche à permettre à un ordinateur de réaliser des tâches à la place de l'Homme tout en améliorant la performance. On peut par exemple citer la reconnaissance d'image.
- Ensuite, les **applications pratiques** qui consistent quant à elles à créer de nouveaux services comme permettre de trouver comment augmenter le rendement de l'entreprise ou baisser les coûts de production de nos produits.

Il existe une multitude d'applications au Machine Learning si bien qu'il fait partie intégrante de nos vies sans même que nous nous en rendions compte. En médecine, il permet de diagnostiquer des maladies et d'analyser des radios. En marketing, il détermine comment améliorer les campagnes et aide à mieux comprendre les besoins clients. Dans l'automobile, il s'illustre dans le développement de voitures autonomes. Enfin dans le divertissement, il permet par exemple aux plateformes de streaming ou aux réseaux sociaux de nous recommander du contenu pertinent.

Vous l'aurez compris, les entreprises du monde entier ont adopté le Machine Learning. Cette technologie, à l'origine d'innovations, leur permet de mettre en place des processus plus intelligents grâce à leurs

capacités d'apprentissage. Selon une étude McKinsey, 49 % des entreprises explorent ou prévoient d'utiliser le Machine Learning. Selon une autre étude, d'Accenture cette fois, 40% des sociétés sondées estiment avoir connu une amélioration de leur productivité grâce à l'utilisation de l'intelligence artificielle et du Machine Learning [20]. Le développement de cette technologie est donc au cœur des préoccupations des entreprises. Gardons toutefois à l'esprit que le déploiement de solutions de Machine Learning s'accompagne de défis, qui incluent l'accès aux données, leur qualité et la pénurie de personnes qualifiées pour résoudre les problèmes.

Arthur Samuel, un informaticien américain et l'un des pères du Machine Learning, définit que le propre de cette technique est de « donner la capacité à une machine d'apprendre sans qu'elle soit programmée explicitement pour ». Cette définition, bien qu'encore un peu floue, place le Machine Learning au croisement de plusieurs domaines.

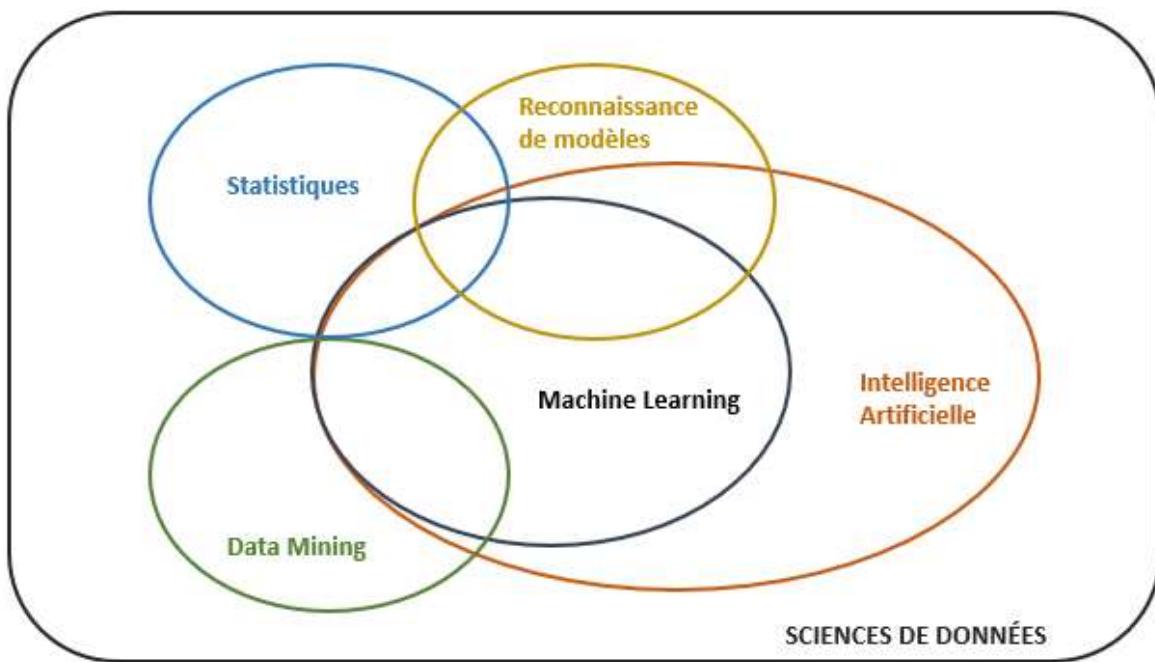


Figure 4 - Machine Learning et DataScience

Sous-catégorie de l'intelligence artificielle, le Machine Learning est une technique utilisant des algorithmes mathématiques basés sur des techniques de statistique et de sciences des données. Ce qu'il faut comprendre, c'est que ces algorithmes mathématiques sont capables d'apprendre et de s'améliorer en fonction d'un objectif défini. Mais comment apprennent-ils ? Comme lors d'un cours dispensé par un professeur à ses élèves, l'algorithme d'apprentissage va recevoir en entrée des données qui lui serviront de base d'apprentissage. Plus le volume de données auxquelles il aura accès est important, plus il deviendra précis. Ces sources de données sont multiples mais on peut toutefois les regrouper en deux grandes catégories :

- Les **données structurées** d'une part qui sont prédéfinies et formatées selon une structure précise. Une base de données relationnelle est un bon exemple de données structurées : les données ont été formatées dans des champs précisément définis, comme le numéro de carte de crédit ou l'adresse. Ces données ont l'avantage d'être relativement faciles à utiliser par les algorithmes ainsi que par les professionnels du secteur. Cependant, la rigidité résultant du formatage et des options de stockage en fait des données peu flexibles.
- Les **données non structurées** quant à elles sont stockées dans leur format d'origine et non traitées avant utilisation. Elles se présentent sous une multitude de formats de fichiers, comme des courriels, des posts sur les réseaux sociaux, des présentations, des images ou encore des audios.

Elles ont l'avantage de disposer d'un taux d'accumulation plus rapide: les données ne devant pas être prédéfinies, elles peuvent ainsi être collectées rapidement et facilement. La difficulté réside dans la possession d'expertise et d'outils spéciaux pour en exploiter pleinement le potentiel [21].

En fonction du type de données dont on dispose, nous serons en mesure d'utiliser un algorithme plutôt qu'un autre. De manière générale, les données structurées sont plus facilement utilisables.

Lorsque l'on s'intéresse à la méthode couramment utilisée lors des projets de Machine Learning, on s'attarde principalement sur six grandes étapes :

1. Identifier les besoins et les objectifs :

Aux prémisses d'un projet de Machine Learning, il est primordial de définir précisément le but et les intérêts de la solution. Développer un projet constitue un processus coûteux et laborieux. Fixer les objectifs permet d'établir le cadre et les limites tout en fournissant une métrique d'évaluation afin de juger la réussite du projet. L'objectif est de poser clairement la problématique métier que l'on souhaite résoudre afin de définir quels types de données recueillir, quels résultats (données de sortie) attendre, et même le type de modèle à utiliser.

2. Collecter les données :

Une fois le cadre clairement défini, il est alors temps de récolter les données nécessaires à l'apprentissage de l'algorithme. La qualité et la quantité des données récoltées ont un impact direct sur l'efficacité du modèle. Plus celles-ci sont nombreuses et fiables, plus le résultat obtenu sera précis et adapté aux besoins de l'entreprise. Il est donc essentiel de réunir des données en fonction des objectifs définis à l'étape précédente. Les sources de données peuvent être multiples, il peut s'agir de données internes comme externes à l'organisation.

3. Préparer les données :

L'adage "Garbage in, garbage out" résume parfaitement la situation à ce stade. Si les données collectées ne sont pas de bonne qualité, les résultats en sortie du projet ne seront pas satisfaisants. En effet, un modèle d'apprentissage réussi passe avant tout par des données de qualité. Il est donc nécessaire d'analyser et de traiter les données recueillies afin d'en découvrir la pertinence. Lors de l'intégration de données, on attache une attention particulière aux données mal annotées, aux données non disponibles, aux doublons et informations incohérentes ou superflues. Il est également important de préparer les données en effectuant parfois des transformations. On veut pouvoir retirer les valeurs aberrantes, standardiser ou encore encoder certaines variables. On peut également s'intéresser à savoir si les données représentent bien la réalité de l'entreprise et si elles ne sont pas biaisées d'une quelconque manière. Cette étape peut être fastidieuse si les données entrées ne sont pas de qualité suffisante. Toutefois, elle constitue la clef de voûte d'un modèle réussi et il est alors essentiel de bien la travailler.

4. Choisir le modèle d'apprentissage :

Une fois les données nettoyées et prêtées à l'emploi, il nous faut choisir le ou les bon(s) algorithme(s) pour traiter notre problématique initiale. Il en existe en effet plusieurs que nous détaillerons au chapitre suivant. L'enjeu consiste à jouer sur les hyperparamètres des algorithmes (variables d'ajustements) permettant de contrôler le processus d'entraînement du modèle afin d'obtenir les meilleurs résultats possibles. Il est commun d'essayer plusieurs algorithmes et de choisir ensuite celui qui fonctionne le mieux.

5. Entraîner et évaluer :

Une fois nos modèles choisis, il nous faut les entraîner. Alimenté en données, le modèle est entraîné sur la durée afin d'améliorer de façon progressive sa capacité à réagir face à une situation donnée, à résoudre un problème complexe ou à effectuer une tâche. Il est donc conseillé de séparer le jeu de données en 2 à 3 ensembles: un jeu d'apprentissage, un jeu de test et un jeu de validation. Nous utilisons l'ensemble d'apprentissage pour enseigner à notre modèle. Attention

toutefois à ce que l'ensemble d'entraînement soit un échantillon représentatif du jeu de données, auquel cas on risquerait d'induire des biais.

6. Tester et déployer :

Une fois le modèle entraîné, il est temps de le confronter à la réalité. On teste l'algorithme sur le jeu de test (qui n'a pas servi à l'apprentissage). À partir de là, nous pouvons modifier le modèle et son entraînement jusqu'à ce que nous obtenions une précision acceptable. Ensuite, nous utilisons le jeu de validation pour nous assurer que nous n'avons pas sûr-adapté le modèle aux données existantes. On évalue ensuite la performance de l'algorithme grâce à diverses métriques que nous étudierons plus loin. L'objectif est de vérifier à quel taux l'algorithme prédit correctement [22].

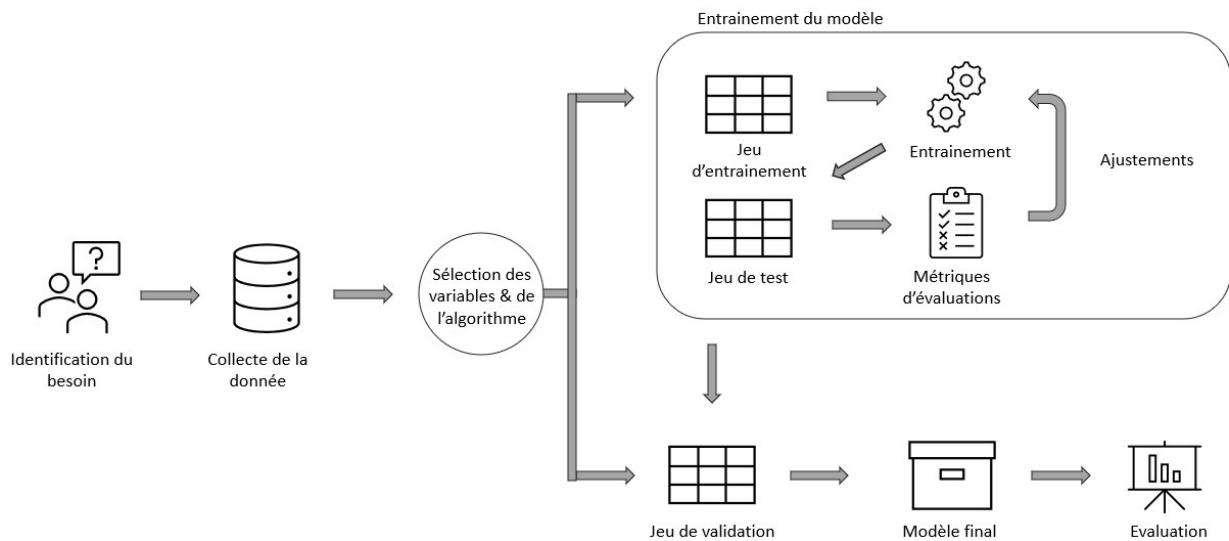


Figure 5 - Processus projet Machine Learning

2.3.1.2 Les algorithmes d'apprentissage

Nous l'avons vu dans le chapitre précédent, les algorithmes d'apprentissage sont le cœur du Machine Learning. Avant de rentrer dans les détails de leurs spécificités respectives, il est important de comprendre qu'en fonction du problème que l'on désire résoudre, nous devons orienter notre choix vers une famille d'algorithme plutôt qu'une autre. Ainsi on distingue 3 grandes familles de problèmes :

- Les problèmes de **régression** se caractérisent par une variable de sortie qui est une valeur réelle ou continue. On veut par exemple prédire l'âge d'une personne, le cours d'une action ou le prix d'un produit. Finalement dans un problème de régression, on cherche des relations entre nos variables.
- Les problèmes de **classification** consistent à catégoriser les observations dans une ou plusieurs classes connues. Le modèle tente de tirer une conclusion des valeurs observées et sa valeur de sortie est donc catégorielle. On peut par exemple chercher à savoir si un mail est un spam ou non.
- Enfin, les problèmes de **clustering** ont pour vocation de regrouper dans des ensemble les observations qui se ressemblent. Contrairement aux problèmes de classification, on ne connaît pas initialement les caractéristiques de chaque groupe. L'algorithme va lui-même proposer des regroupements en fonction des similitudes entre les données. Le clustering est particulièrement utile lorsque vous ne voulez pas étiqueter les choses à la main, que vous ne savez pas quelles sont

les étiquettes à l'avance ou lorsque celles-ci sont trop difficiles à comprendre. On peut par exemple chercher à identifier les différents profils clients de notre entreprise.

En fonction de la problématique, nous sommes désormais à même de nous diriger vers l'une ou l'autre des familles de problèmes. Toutefois, au sein de ses grands ensembles, on retrouve une multitude d'algorithmes d'apprentissage qui peuvent une nouvelle fois être discrétilisés en fonction de leurs rapports à l'apprentissage :

- Avec **l'apprentissage supervisé**, la machine apprend par l'exemple. L'objectif est défini à l'avance, c'est-à-dire que l'on connaît la réponse que doit nous donner l'algorithme. L'algorithme n'a pas la liberté de fournir une autre réponse. Le modèle se compose de paires de données d'entrée (x) et de sortie (y), dans lesquelles la sortie est étiquetée avec la valeur souhaitée. L'algorithme va essayer d'apprendre la fonction de mappage de l'entrée à la sortie $y = f(x)$. Supposons que le but de la machine soit de faire la différence entre un chien et un chat. Notre paire de données d'entrée comprendra l'image d'un chien et l'image d'un chat chacune étiquetée. Lors de l'apprentissage, l'algorithme a donc accès à la réponse. Si nous souhaitons que la machine choisisse le chien. L'image du chien sera donc identifiée au préalable comme étant le résultat attendu. Grâce à l'algorithme, le système compile l'ensemble des données d'entraînement et les met en corrélation. Il identifie alors des similarités, des différences et d'autres points de logique, jusqu'à ce qu'il puisse donner par lui-même la réponse à la question. On l'entraîne donc sur un ensemble connu pour ensuite lui demander de trouver la réponse sur un ensemble que l'on ne connaît pas. En général, on utilise l'apprentissage supervisé pour résoudre des problèmes de régressions ou de classification.
- Au contraire, dans **l'apprentissage non supervisé**, la réponse à la question ne se trouve pas dans le jeu de données. On ne dispose pas de labels permettant d'identifier clairement les enregistrements et c'est à l'algorithme de proposer une réponse. Si on reprend l'exemple de classifier un ensemble d'animaux, cette fois-ci l'algorithme n'a aucune idée des notions de chat et de chien. Il va discriminer les observations en différents ensembles selon leurs caractéristiques. Ainsi il est possible que l'on retrouve au sein d'un même groupe des chats et des chiens. Cela ne signifie pas que l'apprentissage non supervisé est moins efficace que le supervisé. Encore une fois, tout dépend de la question initiale qui a motivé la mise en place du projet. Si la question de départ était « Est-ce un chien ou chat ? », alors le supervisé s'impose. En revanche, si la question était « Quels animaux se ressemblent le plus ? » alors le non-supervisé est de mise. De manière générale, les problèmes de clustering utilisent des algorithmes d'apprentissage non supervisés.
- Entre les deux, on trouve **l'apprentissage semi-supervisé**. Il s'agit d'une vision plus réaliste du monde dans laquelle coexistent des données étiquetées et d'autres non. Ces algorithmes fonctionnent en insérant de petits volumes de données étiquetées afin d'enrichir des ensembles de données non étiquetés. Ainsi, les données étiquetées permettent au système d'avoir une longueur d'avance, ce qui peut améliorer significativement la vitesse et la précision de l'apprentissage. Un algorithme d'apprentissage semi-supervisé invite la machine à analyser les données étiquetées afin de déterminer des propriétés corrélatives qui pourraient être appliquées aux données non étiquetées.
- Enfin, **l'apprentissage par renforcement** est la dernière famille d'algorithmes. Ces algorithmes prennent place lorsque le résultat souhaité n'est pas fixe ou binaire mais peut varier. Ils sont utiles lorsque l'on souhaite par exemple réaliser une IA⁵ qui puisse jouer à des jeux comme les échecs ou le jeu de Go. Il diffère des autres classes d'algorithme par l'interaction agent/environnement qu'il suggère. En fonction de ce qu'il se passe autour de lui, il réagit. Lorsque l'agent prend une décision il passe donc d'un état à un autre. L'idée est de récompenser de manière positive ou négative la décision prise. L'algorithme va ainsi apprendre au fil de ses expériences. Il s'agit pour lui de trouver

⁵ Intelligence Artificielle

un compromis entre essayer de nouvelles choses pour apprendre et refaire l'action qui procure la récompense (équilibre entre exploration et exploitation) [23].

En fonction du type de problème que l'on cherche à résoudre et des données dont on dispose, il existe finalement une multitude d'algorithmes qui nous permettent d'avancer. Toutefois ces algorithmes ne se valent pas tous et certains sont plus adaptés que d'autres en fonction des situations. Il s'agit alors d'évaluer les différents modèles en cherchant à améliorer leurs performances.

2.3.1.3 Evaluation des modèles

« Rome ne s'est pas faite en un jour ». Cet adage nous rappelle qu'il faut du temps pour accomplir un projet important. Il en va de même en Machine Learning. Le premier modèle que nous implémenterons ne sera pas parfait et nécessitera des améliorations au fil du temps. L'objectif est d'améliorer ses performances afin qu'il soit à chaque fois plus précis. Il est également important de garder à l'esprit qu'un temps de calcul convenable et une utilisation des ressources en mémoire raisonnable est plus que souhaitable. Dans ce chapitre, nous allons étudier les différentes techniques d'évaluation de modèles de Machine Learning.

2.3.1.3.1 Baseline et overfitting

Dans tout projet, il est intéressant de disposer d'une Baseline, une référence. Il s'agit d'un élément pouvant nous permettre de critiquer les résultats de notre modèle au regard de quelque chose que l'on connaît. On distingue deux grandes familles de Baseline :

- Tout d'abord, dans certains cas, nous pouvons déjà disposer d'un modèle de Machine Learning répondant à la question. Nous pouvons alors comparer les performances de notre nouveau modèle avec les performances de celui-ci.
- Dans d'autres cas, il peut être intéressant de comparer les performances de notre modèle avec les connaissances métier d'experts de l'organisation. Si on cherche par exemple à prévoir le prix d'un bien immobilier, il peut être judicieux de demander l'avis d'un expert du secteur et de confronter ces estimations à celles du modèle.

Un autre enjeu d'évaluation de modèle est de déterminer si l'algorithme n'a pas sûr-appris. On parle de surapprentissage (overfitting en anglais) lorsque notre modèle est incapable de généraliser ses prédictions sur des données de test, car il a appris par cœur les données d'entrainements. En somme, le modèle a trop bien appris. Pour repérer l'overfitting on peut s'intéresser à savoir si les performances sur les jeux d'entraînement sont largement supérieures à celles sur les données de test. Si c'est le cas, alors l'algorithme a sur appris : cela signifie que face à de nouvelles données, la machine est incapable d'élaborer des modèles prédictifs. Si l'écart est faible, on peut considérer qu'il n'y a pas eu d'overfitting.

2.3.1.3.2 Validation croisée et stratification

Afin d'entraîner notre modèle, nous avons vu plus haut que nous devions partitionner notre jeu de données en plusieurs ensembles. L'objectif de cette manipulation est d'entraîner le modèle sur un échantillon de données représentatif, le tester ensuite sur un autre échantillon et l'ajuster. Une fois les ajustements terminés, on le lance avec le dernier échantillon de validation (afin de vérifier que l'on n'a pas un peu trop ajusté le modèle). L'idée de la validation croisée est de comparer les performances de l'algorithme en faisant varier ces différents échantillons. On va effectuer ces opérations à plusieurs reprises de telle sorte que les données soient à tour de rôle utilisées comme données d'apprentissage, de test et de validation.

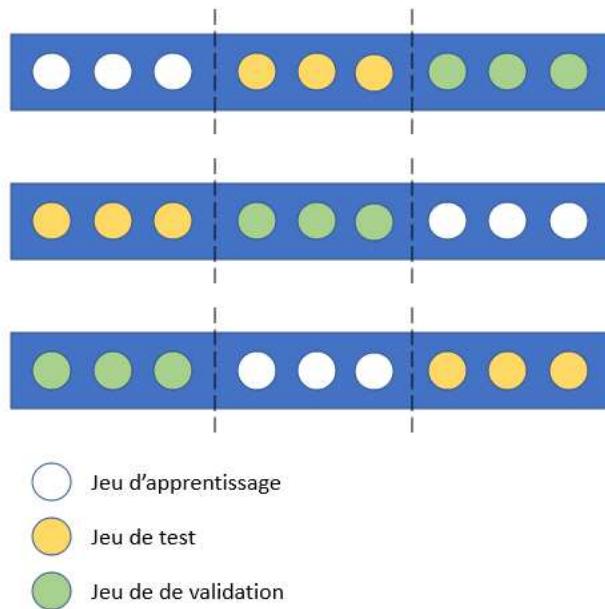


Figure 6 - Exemple de validation croisé à 3 partitionnements

La validation croisée permet donc d'évaluer un modèle de Machine Learning en ayant la moyenne des performances et l'erreur type sur chacun des jeux [24]. Pour des raisons de temps de calcul, on utilise généralement entre cinq et dix partitionnements différents.

Attention toutefois à ne pas découper n'importe comment nos échantillons. Il est important de répartir des observations de façon homogène avant l'échantillonnage, c'est-à-dire répartir les étiquettes pour que chaque échantillon ressemble au maximum à un petit jeu de données connues [24]. C'est ce qu'on appelle la stratification :

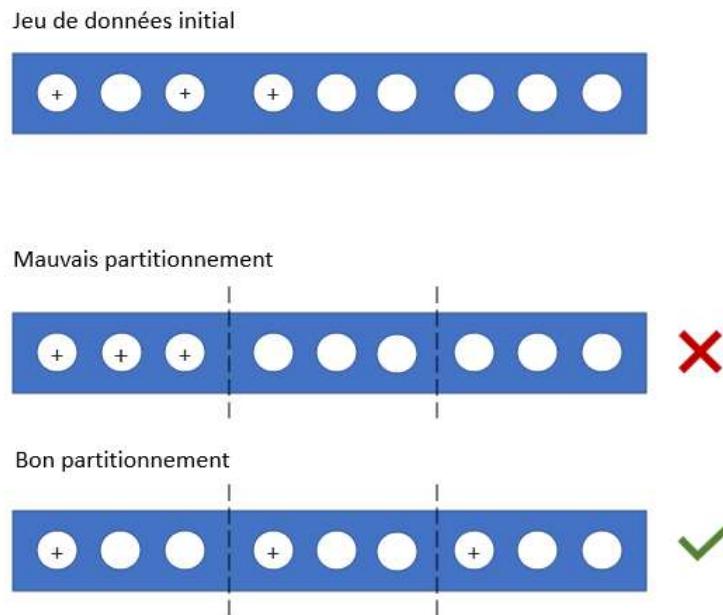


Figure 7 - Exemple de stratification

Outre la validation croisée, il existe une multitude de critères d'évaluations plus spécifiques en fonction du type de problème que l'on cherche à résoudre. Ci-dessous une liste non exhaustive d'entre eux :

- Pour les problèmes de Régression: score R², MAE (Mean Absolute Error), MSE (Mean Square Error), RMSE (Root Mean Square Error), AIC (Akaike Information Criterion), BIC (Bayesian Information Criterion), Goodness-of-fit test
- Pour les problèmes de Classification : Accuracy, Confusion matrix, Sensitivity and specificity, ROC (Receiver Operating Characteristic), score F1
- Pour les problèmes de Clustering : silhouette coefficient, stability

2.3.2 Les problèmes de classification

Chercher à prédire si un employé va quitter l'entreprise revient à travailler sur un problème de classification. On cherche à déterminer la classe de chaque observation comme : « le collaborateur souhaite quitter l'entreprise » ou « le collaborateur ne souhaite pas quitter l'entreprise ». Nous l'avons vu, les départs peuvent être volontaires ou involontaires. Dans notre cas, nous étudions particulièrement les départs volontaires car ce sont ceux qui impactent le plus l'entreprise.

Différents travaux de recherche [25] ont permis de mettre en évidence les variables prédictives les plus importantes dans les problématiques de roulement de personnels volontaires. On y retrouve notamment :

- L'âge
- L'ancienneté dans l'organisation
- La rémunération
- La satisfaction au travail
- La perception de l'équité

Les variables personnelles telles que le sexe, l'origine ethnique, l'éducation et l'état civil du collaborateur constituent également des facteurs importants dans la prédiction de rotation volontaire du personnel mais sont plus difficiles à utiliser en raison des réglementations strictes encadrant l'utilisation des données personnelles.

On peut également citer les conditions de travail, la supervision, la reconnaissance, le potentiel de développement du collaborateur et l'épuisement professionnel comme étant des facteurs significatifs pouvant impacter la prise de décision du collaborateur. Finalement, ces études nous montrent que l'environnement de notre problème est extrêmement vaste. De nombreuses variables peuvent expliquer, ou conduire au départ d'un employé. Ces dernières peuvent varier d'un secteur d'activité à l'autre, d'une équipe à l'autre et d'un collaborateur à l'autre. Il est donc primordial pour les entreprises de personnaliser leurs approches en fonction de leur contexte et d'essayer de comprendre les raisons qui poussent leurs employés à vouloir les quitter.

L'enjeu pour l'entreprise est donc de détecter, grâce à ces variables les collaborateurs « à risque », c'est-à-dire ceux qui seraient potentiellement à même de quitter l'organisation. Dans ce type de problème (classification) on cherche avant tout un score, une probabilité qu'a l'observation d'appartenir à une classe. Nous étudierons par la suite différents modèles statistiques permettant de déterminer cette fameuse probabilité. Pour les problèmes de classification binaire tel que celui du turn-over, nous devons définir un seuil de confiance à partir duquel on va considérer que l'observation appartient bel et bien à la classe. Imaginons par exemple que notre seuil soit à 0.5, toutes les observations ayant obtenu une note supérieure au seuil se verront attribuer à cette classe. Ce seuil n'est pas fixe, il peut changer en fonction des objectifs que doit remplir l'algorithme. Il est à déterminer par le décideur en fonction de ces objectifs. Souvent ce dernier devra faire des compromis afin de choisir celui qui correspond le mieux à ses besoins et des métriques qu'il cherche à satisfaire.

2.3.2.1 Evaluer les problèmes de classification

Avant d'étudier en détail les différents algorithmes de classification, arrêtons-nous un instant sur leur méthode d'évaluation. Nous l'avons dit, différents modèles peuvent être utilisés pour répondre à la même question. L'enjeu étant de choisir celui qui fonctionne le mieux au vu de notre contexte. La validation croisée, déjà évoquée dans le chapitre précédent peut être une première manière d'évaluer nos différents modèles. Dans la suite de ce chapitre nous allons nous intéresser aux métriques spécifiques aux problèmes de classification qui peuvent être utilisés pour comparer différents modèles.

2.3.2.1.1 La matrice de confusion

Les modèles de classification permettent d'étiqueter de nouvelles observations non étiquetées. Pour créer ces modèles, nous utilisons les observations déjà étiquetées du jeu d'entraînement afin que l'algorithme apprenne. Pour tester la qualité du modèle, nous utilisons le jeu de test dans lequel les étiquettes sont cachées. Ainsi, on va classer ces individus « tests » en faisant comme si on ne connaissait pas leur étiquette. On va alors pouvoir comparer ces nouvelles étiquettes avec les vraies afin de connaître le taux de bonne classification et donc la qualité du modèle. Ces différentes comparaisons nous permettent finalement d'obtenir la matrice de confusion. Dans le cas d'une classification binaire, la matrice de confusion est un tableau à 4 valeurs représentant les différentes combinaisons de valeurs réelles et valeurs prédites.

		Predicted Class		
		Positive	Negative	
Actual Class	Positive	True Positive (TP)	False Negative (FN) Type II Error	Sensitivity $\frac{TP}{(TP + FN)}$
	Negative	False Positive (FP) Type I Error	True Negative (TN)	Specificity $\frac{TN}{(TN + FP)}$
	Precision $\frac{TP}{(TP + FP)}$	Negative Predictive Value $\frac{TN}{(TN + FN)}$	Accuracy $\frac{TP + TN}{(TP + TN + FP + FN)}$	

Figure 8 - Matrice de confusion

De cette matrice de confusion, on peut tirer différentes métriques que nous allons expliquer par la suite nous permettant d'appréhender un peu plus notre modèle. Dans la réalité, il est rare d'obtenir un excellent score pour chaque métrique. Ainsi, en fonction de la nature du problème, on va donner plus d'importance à un indicateur plutôt qu'à un autre, l'enjeu étant de trouver l'équilibre qui répond le mieux au contexte du problème ainsi qu'aux besoins du décideur :

- La « Précision » ou « Positive Predictive Value » (PPV), est le taux de prédictions correctes parmi les prédictions positives. C'est la capacité du modèle à prédire vrai lorsque ça l'est bel et bien.

$$\text{Precision} = \frac{TP}{TP+FP}$$

La précision mesure donc la capacité du modèle à ne pas faire d'erreur lors d'une prédiction positive. Elle permet de savoir combien d'individus sélectionnés par mon modèle sont réellement pertinents. Dans un contexte métier, si on imagine qu'un décideur cherche à savoir quels clients désirent résilier leurs abonnements afin de leur proposer une réduction pour les faire rester. Ce dernier peut vouloir choisir une stratégie visant une précision élevée afin de ne pas proposer de réduction aux clients qui ne souhaitent pas partir. On détectera peut-être moins de résiliations mais on évitera une perte de marge. Attention toutefois on peut relativement facilement avoir une très bonne précision en prédisant très peu de positifs puisque l'on ne prendra pas beaucoup de risque pour se tromper [27].

- La « Négative Préditive Value » (NPV) est le pendant de la PPV mais pour les valeurs négatives. Il s'agit de connaître le taux de prédictions correctes parmi les prédictions négatives. Le modèle prédit faux et ça l'est bel et bien.

$$\text{Negative Predictive Value} = \frac{TN}{TN+FN}$$

- La « Sensitivity » (True Positive Rate TPR) aussi appelé Recall mesure la capacité du modèle à détecter l'ensemble des individus positifs. C'est-à-dire la proportion de positifs que l'on a correctement identifiés parmi l'ensemble.

$$\text{Sensitivity} = \frac{TP}{TP+FN}$$

Il permet de savoir combien d'individus pertinents sont identifiés par le modèle parmi tous les éléments pertinents. Si on reprend l'exemple métier exposé dans la section sur la PPV. Cette fois le décideur ne souhaite plus éviter la perte de marge (il accepte de proposer des réductions à des clients qui ne seraient peut-être pas partis), en revanche il ne veut absolument pas perdre de client. Il va donc chercher à avoir une Sensitivity élevé afin d'identifier le maximum de clients susceptibles de partir en prenant le risque de proposer des réductions à des clients qui ne seraient pas partis. Encore une fois, on peut avoir un très bon rappel en prédisant systématiquement « positif ». On ne ratera aucun individu, mais notre modèle ne servira pas à grand-chose. L'enjeu est donc de trouver une balance entre les différentes métriques [27].

- La « Specificity » (True Négative Rate TNR) indique, quant à elle, la proportion d'individu identifiés comme n'appartenant pas à la classe parmi l'ensemble de ceux n'y appartenant pas.

$$\text{Specificity} = \frac{TN}{TN+FP}$$

Une Specificity de 90% signifie par exemple que 9 individus sur 10 n'appartenant pas à la classe sont correctement identifiés par le modèle comme n'y appartenant pas.

- Enfin, l'« Accuracy » constitue un indicateur plus global de la performance de notre modèle. Il permet de mesurer la fréquence à laquelle l'algorithme classe correctement un point de données (que l'observation soit positive ou négative).

$$\text{Accuracy} = \frac{TP+TN}{TP+FP+FN}$$

Ces différentes mesures permettent de juger la qualité d'un modèle de classification. Elles sont cependant à utiliser relativement prudemment. Elles peuvent parfois montrer leurs limites comme lorsque les données sont fortement déséquilibrées et qu'une classe est surreprésentée par rapport à l'autre. Dans ce genre de contexte, l'Accuracy peut par exemple conduire à des erreurs de jugement. Si on considère un jeu de données avec 10% d'individus positifs et que notre modèle nous prédit l'ensemble des individus comme négatifs, nous obtiendrons une Accuracy de 90% (un bon score) alors que notre modèle ne prédit rien. Il faut donc garder à l'esprit que la mesure est à utiliser au regard du contexte des données et du problème.

Afin de remédier au contexte de données déséquilibrées, on peut alors utiliser une variante de l'Accuracy appelé la Balanced Accuracy. Cette métrique considère qu'au lieu de pondérer chaque rappel (Sensitivity et Specificity) par la proportion de sa classe, la pondération est la même pour chaque classe. Dans le cas d'une classification binaire comme dans la problématique du turn-over, chaque rappel est donc pondéré par $\frac{1}{2}$ [28].

$$\text{Balanced Accuracy} = \frac{1}{2} * \frac{TP}{TP+FN} + \frac{1}{2} * \frac{TN}{TN+FP}$$

2.3.2.1.2 Le score F1

Le F1-score est une métrique particulièrement utilisée pour les problèmes de classification utilisant des données déséquilibrées comme la détection de fraudes, la prédition d'incidents graves ou dans notre cas, la prédition du turn-over. Il permet de résumer les valeurs de la précision et de la sensibilité en une seule métrique. Mathématiquement, le F1-score se définit comme étant la moyenne harmonique de ces deux métriques [29] :

$$\text{Score F1} = \frac{2}{\frac{1}{\text{Precision}} + \frac{1}{\text{Sensitivity}}}$$

L'équation peut également s'écrire à partir des coefficients de la matrice de confusion étudié plus haut :

$$\text{Score F1} = \frac{TP}{TP + \frac{1}{2}(FN+FP)}$$

L'enjeu est donc de comparer les prédictions positives correctes aux erreurs faites par le modèle. Ainsi un score F1 de 50% signifie que pour une prédition positive correcte, le modèle fasse deux erreurs (faux négatif ou faux positif).

Le F1-score considère que la précision et la sensibilité ont le même poids dans le calcul de notre score. En fonction du contexte du problème et de la problématique elle-même, on peut vouloir donner plus d'importance à l'un de ses deux aspects. Afin de faire varier la pondération de ces termes, on utilise la formule du F-beta score [29] :

$$F_{\beta} - \text{score} = (1 + \beta^2) * \frac{\text{Precision} * \text{Sensitivity}}{(\beta^2 * \text{Precision}) + \text{Sensitivity}}$$

Elle peut également s'écrire :

$$F_{\beta} - score = \frac{TP}{TP + \frac{1}{1+\beta^2}(\beta^2 FN + FP)}$$

Ainsi pour la variation de la valeur de beta va permettre de pondérer nos mesures :

- Pour $\beta \geq 1$, on accorde plus d'importance à la sensibilité (autrement dit aux faux négatifs).
- Pour $\beta \leq 1$, on accorde plus d'importance à la précision (autrement dit aux faux positifs).
- Pour $\beta = 1$, on retrouve le F1-score, qui accorde autant d'importance à la précision qu'à la sensibilité.

2.3.2.1.3 Courbe ROC (Receiver Operating Characteristic) et AUC (Area Under the Curve)

La courbe ROC permet de visualiser l'évolution de la Spécificité et de la Sensibilité d'un modèle en fonction du seuil de classification. Elle trace donc l'ensemble des valeurs du couple (1-Specificity, Sensitivity) selon le seuil. Utiliser l'anti-spécificité (1-Specificity) revient à calculer le taux de faux positifs :

$$1 - Specificity = 1 - \frac{TN}{TN+FP}$$

Dans le cas d'une classification binaire, on peut par exemple considérer le jeu de données et le score attribué par le modèle suivant [30] :

Etiquette réelle	+	-	+	+	-	-
Score de l'algorithme	0.99	0.95	0.51	0.45	0.10	0.01

Un score élevé signifie que l'algorithme estime que l'observation à de grandes chances d'appartenir à la classe. Si on choisit un seuil supérieur à 0.99, toutes nos observations seront négatives. Ainsi notre sensibilité (taux de vrai positif TP / P) sera égale à 0. L'anti-spécificité quant à elle vaudra également 0 (taux de faux positifs FP/P).

On obtient la matrice de confusion suivante :

		Prédiction			
		1	0		
Réalité	1	0	3		
	0	0	3		

$$\text{Sensibilité} = TP / (TP + FP) = 0/3 = 0$$

$$1 - \text{Spécificité} = 1 - (3/3) = 0$$

En fonction du seuil de classification choisi, on obtient le tableau suivant :

Seuil	> 0.99	0.95 – 0.99	0.51 – 0.95	0.49 – 0.51	0.10 – 0.45	0.01 – 0.10	< 0.01
TP / P	0	1/3	1/3	2/3	1	1	1
FP / P	0	0	1/3	1/3	1/3	2/3	1

On constate par exemple que, si on considère un seuil supérieur à 0.95, notre sensibilité sera de $\frac{1}{3}$ (on prédit 1 observation positive sur les 3) mais notre anti-spécificité reste à 0.

Finalement on obtient la courbe ROC suivante :

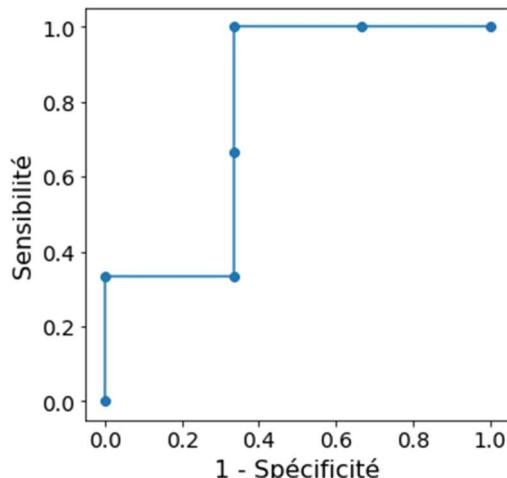


Figure 9 - Courbe ROC lambda

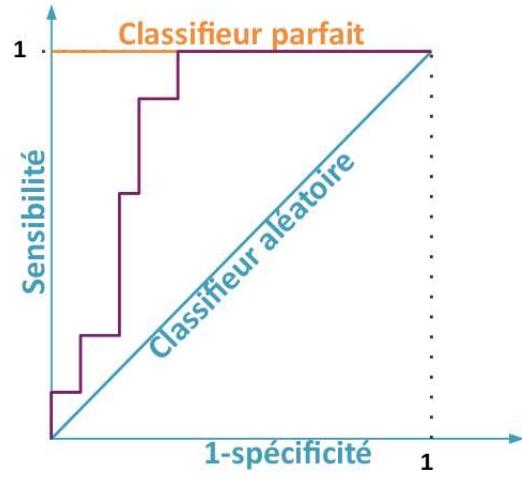


Figure 10 - Courbe ROC théorique

En bas à gauche de la courbe ROC, on considère la plus grande valeur prédite pour seuil de classification. On prédit donc que tous les points sont négatifs puisqu'aucun ne dépasse le seuil. A l'inverse, en haut à droite, on prend la plus petite valeur prédite comme seuil. Tous les points sont au-dessus et donc prédits positifs. Le reste de la courbe décrit toutes les situations intermédiaires pour lesquelles le seuil est entre ces valeurs.

Un modèle parfait va systématiquement associer des scores plus faibles aux exemples négatifs qu'aux exemples positifs. La courbe ROC correspondante dessine donc le coin supérieur gauche du carré. Un modèle aléatoire, par contraste, va dessiner la diagonale du carré : quel que soit le seuil utilisé, comme le modèle est aléatoire, on aura la même proportion de prédictions positives correctes que de prédictions positives incorrectes [30].

Afin de comparer différents classificateurs, on peut résumer la courbe ROC de chacun des modèles à son aire sous la courbe (AUC). Le classifieur parfait aura donc une AUC de 1 tandis que le classifieur aléatoire se contentera d'une AUC de 0,5. On aura tendance à préférer le classifieur qui possède la plus grande AUC.

2.3.2.1.4 Courbe précision-rappel (“PR curve”)

Un bon algorithme de classification devrait avoir une précision élevée (on prédit vrai lorsque ça l'est vraiment) ainsi qu'une sensibilité élevée (les individus pertinents identifiés sur les individus pertinents réels). Toutefois, en général, utiliser des algorithmes d'apprentissage signifie trouver un compromis entre les deux. La courbe précision-rappel a la valeur de sensibilité (rappel TPR) sur l'axe des abscisses et la précision sur l'axe des ordonnées. La précision aide ainsi à mettre en évidence la pertinence des résultats récupérés.

Si on reprend l'exemple étudié dans la section précédente, on obtient le tableau suivant :

Seuil	> 0.99	0.95 – 0.99	0.51 – 0.95	0.49 – 0.51	0.10 – 0.45	0.01 – 0.10	< 0.01
TP / P	0	1/3	1/3	2/3	1	1	1
Précision	-	1	1/2	2/3	3/4	3/5	3/6

La courbe précision-rappel permet de visualiser le compromis entre la précision et le rappel pour différents seuils :

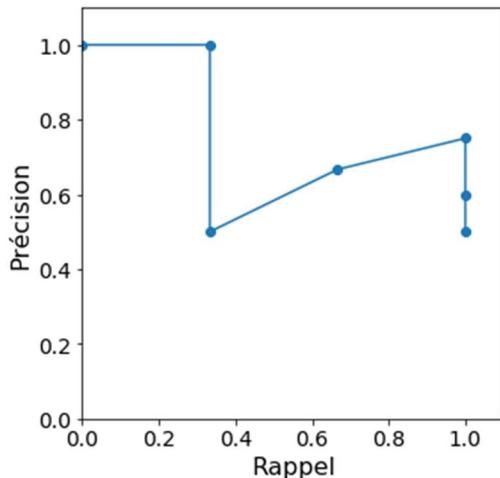


Figure 11 - PR curve lambda

La courbe PR est particulièrement pertinente lorsque l'on est dans une situation de recherche d'information. Dans ces contextes, nous sommes généralement en présence de jeux de données déséquilibrés dans lequel la classe à trouver est sous représentée. Dans ce type de problèmes, le nombre de vrais négatifs est souvent important ce qui rend la courbe ROC moins pertinente que la courbe PR. Ici, on se concentre uniquement sur les observations prédictives positives. On peut noter qu'afin de comparer deux modèles, on utilise une nouvelle fois l'aire sous la courbe [31].

2.3.2.1.5 Le choix du seuil

Le problème d'anticipation de départ des salariés nous renvoie à deux classes : « est susceptible de partir » ou « n'est pas susceptible de partir ». Nous sommes donc dans un contexte binomial et devons donc introduire un seuil de classification [26] :

$$y = \begin{cases} 1 & \text{si } h(x) \geq \text{seuil} \\ 0 & \text{si } h(x) < \text{seuil} \end{cases}$$

La fixation de la valeur seuil dépend du problème de classification lui-même. La décision concernant la valeur du seuil est fortement influencée par les valeurs des métriques de précision et de rappel que nous avons vu précédemment. Idéalement, nous voulons que la précision et le rappel soient de 1, mais c'est rarement le cas. Nous allons donc devoir jouer entre les métriques en fonction de nos besoins :

- Faible précision / rappel élevé :** Dans les applications où nous voulons réduire le nombre de faux négatifs sans nécessairement réduire le nombre de faux positifs, nous choisissons une valeur de seuil qui fournit une valeur faible de précision ou une valeur élevée de rappel. Par exemple, dans une application de diagnostic de cancer, nous ne voulons pas qu'un patient affecté soit classé comme non affecté. Il est vital de détecter la maladie chez le patient réellement atteint. On préfère donc avoir un peu plus de faux positifs tant que l'on détecte l'intégralité des malades [26].
- Haute précision / Faible rappel :** Dans les applications où nous voulons réduire le nombre de faux positifs sans nécessairement réduire le nombre de faux négatifs, nous choisissons une valeur de décision qui donne une précision élevée ou une faible valeur de rappel. Par exemple, si nous classons les clients qui réagiront positivement ou négativement à une publicité personnalisée. Nous voulons être absolument sûrs que le client réagira positivement à la publicité, sinon, une réaction négative peut entraîner une perte de ventes potentielles du client. En revanche on accorde moins d'importance à trouver l'ensemble des clients qui auraient été réceptifs [26].

Finalement, en fonction de la nature du problème, on va privilégier certains indicateurs plutôt que d'autres. Ainsi la valeur du seuil sera choisie de manière à maximiser l'une ou l'autre des métriques.

2.3.2.2 Les algorithmes de classification

Afin de prédire le départ d'un collaborateur, nous cherchons finalement à résoudre un problème de classification. Il existe aujourd'hui de nombreux algorithmes, plus ou moins performants, pouvant être utilisés afin de résoudre ce type de problème. Différents chercheurs ont travaillé à comparer ces algorithmes en fonction de contexte et d'objectifs d'études. Dans son article du journal « International Journal of Advanced Research in Artificial Intelligence » [25], Rohit Punnoose recense les travaux relatifs à notre problématique :

Auteur	Problème	Algorithmes étudiés	Recommandation
Jantan, Hamdan and Othman	Techniques de Machine Learning pour la prédition de la performance des employés	decision tree, Random Forest, Multilayer Perceptron (MLP) and Radial Basic Function Network	C4.5 decision tree
Nagadevara , Srinivasan and Valk	Relation entre les comportements de retrait tels que les retards et l'absentéisme, le contenu du travail, l'ancienneté et les données démographiques, et la rotation du personnel.	Artificial neural networks, logistic regression, classification and regression trees (CART), classification trees (C5.0), and discriminant analysis	Classification and regression trees (CART)
Hong, Wei and Chen	Faisabilité de l'application des modèles Logit et Probit aux prédictions de rotation volontaire des employés.	Logistic regression model (logit), probability regression model (probit)	Logistic regression model (logit)
Marjorie Laura KaneSellers	Explorer les diverses variables personnelles et professionnelles ayant un impact sur le roulement volontaire des employés.	Binomial logit regression	Binomial logit regression
Alao and Adeyemo	Analyse de l'attrition des employés à l'aide d'algorithmes d'arbres de décision multiples	C4.5, C5, REPTree, CART	C5 decision tree
Saradhi and Palshikar	Comparer les techniques d'exploration de données pour prédire le taux de rotation des employés.	Naïve Bayes, Support Vector Machines, Logistic Regression, Decision Trees and Random Forests	Support Vector Machines
Rohit Punnoose	Prédiction de turn-over avec un contexte de bruit dans les données	Logistic Regression, Naïve Bayesian, Random Forest (Depth controlled), SVM (RBF kernel), LDA, KNN (Euclidean distance), XGBoost	XGBoost

M. Punnoose met finalement en évidence dans son article que les problèmes de classification dans les ressources humaines sont particulièrement sujets à la qualité des données du SIRH. Il nous explique que la plupart des organisations n'ont pas cherché, dans la mise en place de leur SIRH, à se doter d'un outil capable de réellement récolter et stocker les données de leurs employés. La compréhension limitée des avantages et des coûts de la mise en place de tels systèmes a conduit les organisations à se retrouver avec des données RH peu fiables, mal renseignées et peu exhaustives. A titre d'exemple, Grant Thornton utilise la solution SIRH Talentsoft citée précédemment dans ce document. Bien que la solution soit reconnue, il s'avère que les données contenues au sein de l'outil ne sont pas toujours fiables ou cohérentes. Les saisies ne sont pas effectuées avec rigueur et certaines informations ne sont tout bonnement pas collectées. Certaines données sont même stockées sur d'autres supports que le SI⁶ tels que des fichiers Excel au sein des équipes. Ces lacunes sont le résultat d'une incompréhension et d'un manque de vision de ces métiers non-initiés à l'informatique et aux problématiques décisionnelles. La volonté de faire évoluer la société vers un mode d'administration piloté par les données doit alors être comprise et partagée par tous. Il appartient à l'organisation de développer une réelle culture de la donnée dans l'intégralité de la structure y compris dans les ressources humaines.

Ce constat n'est pas une bonne nouvelle pour nos algorithmes de classification puisque plus la donnée en entrée est peu fiable et aride, moins la prédiction que notre modèle nous fournira sera exploitable. Les algorithmes vont donc devoir composer avec un certain bruit dans les données. M.Punnoose montre dans ses travaux que les classifieurs à base d'arbres sont les plus performants pour ce type de problème. Il montre entre autres que grâce à son mécanisme de correction itérative, l'algorithme du gradient boosting résiste mieux au bruit dans les données que les autres modèles. Ci-dessous ses résultats en utilisant un jeu de données de 73 115 observations avec 33 variables (27 numériques et 6 catégorielles) [25] :

Algorithme	AUC	Temps d'exécution	Maximum de mémoire utilisée (sur 16GB)
XGBoost	0.88	16 min 12 sec	12%
Logistic Regression	0.66	52 sec	20%
Naïve Bayesian	0.64	59 sec	20%
Random Forest (Depth controlled)	0.79	23 min 10 sec	29%
SVM (RBF kernel)	0.68	105 min 30 sec	21%
LDA	0.74	6 min 51 sec	35%
KNN (Euclidean distance)	0.52	180 min 12 sec	35%

Dans la suite de ce chapitre, nous allons nous attarder sur la théorie derrière les différents algorithmes de classification ainsi que les différentes techniques d'évaluations de ces modèles. L'objectif n'est pas d'expliquer chaque algorithme dans ces moindres détails mais de comprendre la manière dont ils fonctionnent, leurs forces et leurs faiblesses. L'objectif est d'obtenir une vision globale de ces algorithmes afin d'avoir en tête leur fonctionnement.

2.3.2.2.1 Logistic Regression Algorithm

La régression logistique est le premier algorithme que nous étudierons. Il s'agit d'un algorithme classique pour la résolution de problèmes de classification supervisée. Dans les problèmes de classification, la

⁶ Système d'information

variable cible y (la classe à prévoir), ne peut prendre que des valeurs discrètes pour un ensemble donné de caractéristiques X . La régression logistique estime la probabilité qu'un événement se produise, tel que partir ou ne pas partir, sur la base d'un ensemble donné de variables indépendantes. Comme le résultat est une probabilité, la variable dépendante est bornée entre 0 et 1. La régression logistique modélise les données à l'aide de la fonction sigmoïde. On cherche à savoir quelle est la probabilité que l'observation appartienne à la classe A ou à la classe B [26].

$$f(x) = \frac{1}{1+e^{-x}}$$

Graphiquement, celle-ci correspond à une courbe qui a pour limites 0 et 1 lorsque x tend respectivement vers $-\infty$ et $+\infty$.

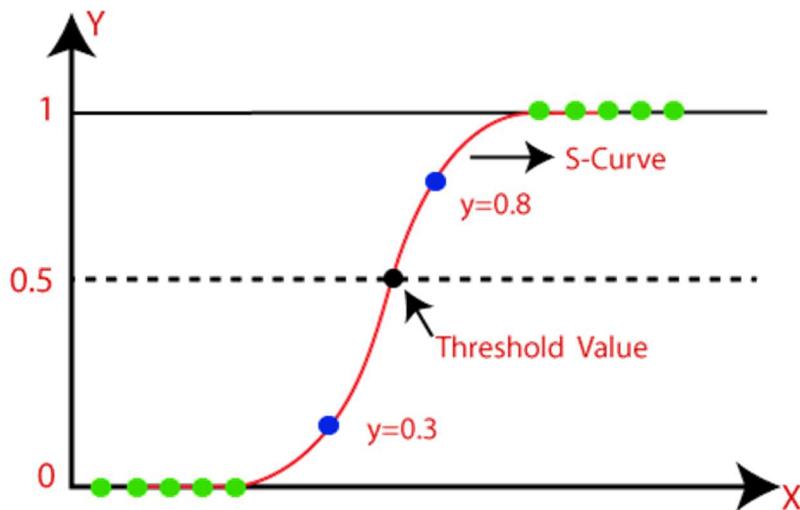


Figure 12 - Fonction sigmoïde

En fonction du nombre de classe possibles, la régression logistique peut être de différents types :

- **Binomial** : la variable cible ne peut avoir que 2 types possibles : « 0 » ou « 1 » qui peuvent représenter « gagné » contre « perdu », « reste » contre « part », « vivant » contre « mort », ...etc.
- **Multinomial**: la variable cible peut avoir plus de 2 types qui ne sont pas ordonnés (c'est-à-dire que les types n'ont pas de signification quantitative) comme « maladie A » contre « maladie B » contre « maladie C ».
- **Ordinal** : il traite des variables cibles avec des catégories ordonnées. Par exemple, un score de test peut être catégorisé comme: « très mauvais », « médiocre », « bon », « très bon ». Ici, chaque catégorie peut recevoir un score comme 0, 1, 2, 3.

Dans notre contexte, nous utiliserons donc une régression Logistique de type binomiale. Plus généralement, la régression logistique peut être utilisée pour la détection de fraudes. Elle permet d'aider les équipes à identifier les anomalies de données, les comportements caractéristiques peuvent être davantage associés à des activités frauduleuses. Dans le même registre, elle peut être utilisée pour éliminer les faux comptes d'utilisateurs des bases. Elle est également utilisée en médecine pour la prédiction de maladie sur des populations données ou encore pour évaluer les départs au sein d'une organisation.

2.3.2.2.2 Naïve Bayes Algorithm

Les classifieurs naïfs Bayésiens sont une collection d'algorithmes de classification basés sur le théorème de Bayes. Ces algorithmes sont parmi les plus simples et les plus efficaces du Machine Learning et permettent de faire des prédictions rapides. Ils peuvent être utilisés à la fois pour les classifications binaires mais également pour les classifications multi-classe. Ils sont notamment utilisés dans la classification de texte telle que le filtrage de spam, l'analyse des sentiments ou le Credit Scoring (évaluer la capacité d'un demandeur de crédit bancaire à bénéficier et solder un financement).

Les algorithmes naïfs Bayésiens sont un compromis de deux notions :

- Tout d'abord ils sont « **naïfs** ». Cela signifie que l'on considère d'une part que chaque paire de caractéristiques est indépendante l'une de l'autre, et d'autre part, chaque caractéristique dispose du même poids. Tous les attributs sont pertinents et supposés contribuer de manière égale au résultat. On remarque que les hypothèses faites ici ne sont généralement pas correctes dans des situations réelles. L'hypothèse d'indépendance notamment est rarement correcte mais l'algorithme fonctionne quand même souvent bien dans la pratique [32].
- Ensuite ils sont « **Bayésiens** ». En effet, ces algorithmes se basent sur le théorème de Bayes. Ce dernier permet de connaître la probabilité qu'un événement se produise compte tenu de la probabilité qu'un autre événement se soit déjà produit [33]. Concrètement, dans un jeu de données, la probabilité d'une observation d'appartenir à la classe A se définit comme suit :

$$P(A/X) = \frac{P(X|A)*P(A)}{P(X)}$$

Où, A est une variable de classe et X est un vecteur de caractéristiques (de taille n) tel que :

$$X = (\text{âge}, \text{sexe}, \text{nombre d'enfants}, \dots, x_n)$$

2.3.2.2.3 Arbres de décisions

Les arbres de décisions sont des algorithmes d'apprentissage supervisé qui peuvent être utilisés à la fois pour des problèmes de classification mais également pour les problèmes de régressions. Il s'agit de classificateurs arborescents, où les nœuds internes représentent les caractéristiques d'un ensemble de données, les branches représentent les règles de décision et chaque nœud feuille représente le résultat.

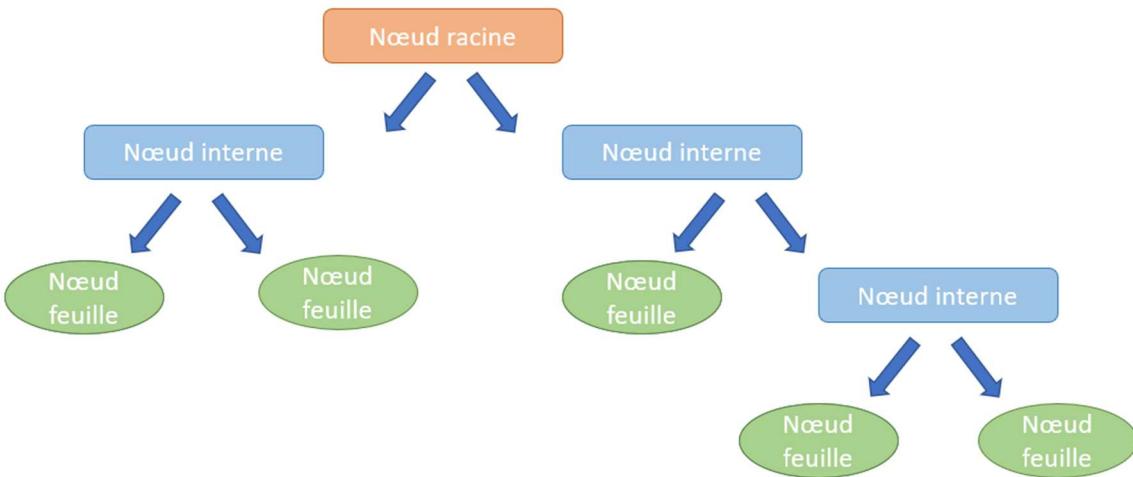


Figure 13 - Schéma arbre de décision

Au sein de l'arbre, chaque nœud interne désigne un test sur un attribut, chaque branche représente un résultat du test et chaque nœud feuille (nœud terminal) contient une étiquette de classe. Afin de progresser dans la construction de l'arbre, on divise l'ensemble source en sous-ensembles grâce à un test de valeur d'attribut. L'objectif est de déterminer l'attribut qui va nous permettre de séparer au mieux notre ensemble de données. Cette opération est effectuée récursivement jusqu'à atteindre un nœud feuille de l'arbre (tous les individus du sous-ensemble appartiennent à la même classe) [34].

En considérant le jeu de données suivant :

Individu	Toux	Fièvre	Poids	Douleur
Marie	Non	Oui	Normal	Gorge
Fred	Non	Oui	Normal	Abdomen
Julie	Oui	Oui	Maignre	Aucune
Elvis	Oui	Non	Obese	Poitrine

Un arbre de décision pourrait être :

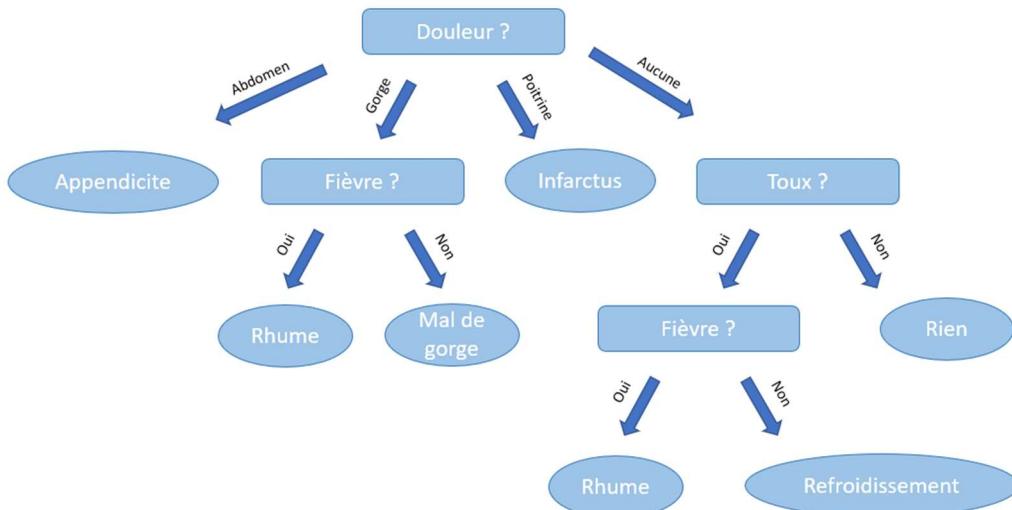


Figure 14 - Exemple arbre de décision

Mais alors comment l'algorithme détermine-t-il quel est le meilleur attribut pour le nœud-racine et les nœuds internes suivants ? La mesure de sélection d'attribut (ASM) se divise en deux techniques distinctes :

- Tout d'abord, le **gain d'information** qui se définit comme la mesure des changements d'entropie après la segmentation d'un ensemble de données en fonction d'un attribut. L'entropie permet de mesurer l'incertitude d'une variable aléatoire. Elle caractérise l'impureté des valeurs échantillonnées. Par exemple, si tous les échantillons du jeu de données, S , appartiennent à une même classe, alors l'entropie sera égale à zéro. En revanche, si seulement la moitié des échantillons sont classifiés dans une classe et l'autre moitié dans une autre classe, l'entropie sera à son maximum de 1. L'entropie calcule donc la quantité d'informations qu'une fonctionnalité nous fournit sur une classe. Afin de sélectionner la meilleure variable pour effectuer le fractionnement et trouver l'arbre de décisions optimal, il convient d'utiliser l'attribut avec le plus petit total d'entropie [35].

$$\text{Entropie } (S) = \sum_{i=1}^c - p_i \log_2 p_i$$

tel que :

- S le jeu de données avec lequel l'entropie est calculée
- c les classes du jeu S
- p_i la proportion de points de données appartenant à la classe c par rapport au nombre total de données du jeu S

L'arbre de décision essaie toujours de maximiser la valeur du gain d'informations. Le nœud/attribut ayant le gain d'informations le plus élevé est divisé en premier. On peut calculer le gain à l'aide de la formule ci-dessous :

$$\text{Gain d'information } (S, a) = \text{Entropie } (S) - \sum_{v \in \text{Valeurs}(a)} \frac{|S_v|}{|S|} \text{Entropie } (S_v)$$

tel que :

- Valeurs(a) est l'ensemble des valeurs v possibles pour a
- a un attribut
- $|S_v| / |S|$ représente la proportion des valeurs dans S_v par rapport au nombre de valeurs dans le jeu de données S

- L'autre technique consiste à calculer l'**indice de Gini** afin de mesurer la fréquence à laquelle un élément choisi au hasard serait identifié de manière incorrecte.

$$\text{Indice de Gini} = 1 - \sum_i (p_i^2)$$

On préfère sélectionner l'attribut qui a un indice de Gini le plus petit possible. On peut noter que cet indice ne crée que des fractionnements binaires [34].

Une fois la sélection d'attribut effectuée, on peut vouloir contrôler la complexité du nombre de branches et de feuilles de l'arbre afin de trouver le nombre optimal de nœud. Cette optimisation est appelée **l'élagage** et permet d'obtenir un arbre optimal. Un arbre trop grand augmente le risque de surapprentissage, et un arbre trop petit peut ne pas capturer toutes les caractéristiques importantes de l'ensemble de données. Encore une fois, deux techniques sont possibles :

- **L'élagage de la complexité des coûts** ou pré-élagage, consiste à arrêter de diviser un nœud quand la pureté des points qui domine est non parfaite mais suffisante. C'est à dire arrêter de diviser

quand il y a une classe dominante dans le nœud. Il nous faut alors utiliser un seuil pour détecter les classes dominantes.

- **La réduction des erreurs d'élagage** ou post-élagage intervient une fois l'arbre terminé. On va simplifier l'arbre en remontant des feuilles vers la racine pour trouver ou élaguer. On utilise des critères de qualité qui mesurent l'erreur à chaque nœud.

Finalement, la construction d'un classificateur d'arbre de décision ne nécessite aucune connaissance du domaine métier ou réglage de paramètres, et est donc appropriée pour la découverte de connaissances exploratoires. Il permet de générer des règles de décision compréhensibles sans nécessiter beaucoup de calculs. Il permet de traiter des variables continues et catégorielles et fournit une indication claire des champs les plus importants pour la prédiction ou la classification. Toutefois, il montre ses limites lorsque le jeu de données contient de nombreuses classes et un nombre relativement faible d'exemples d'apprentissage.

2.3.2.2.4 Random Forest Algorithm

Comme pour les arbres de décision, l'algorithme Random Forest peut être utilisé à la fois pour les problèmes de classification et de régression. Il s'agit d'un algorithme supervisé ensembliste, c'est-à-dire qu'il combine plusieurs classifieurs afin de résoudre un problème complexe.

L'algorithme du Random Forest contient plusieurs arbres de décisions, entraînés sur divers sous-ensembles du jeu d'entraînement. Au lieu de s'appuyer sur un seul arbre de décision, Random Forest étudie la prédiction de chacun des arbres qu'il contient et prédit la sortie finale sur la base de votes majoritaires des prédictions. Le plus grand nombre d'arbres dans la forêt conduit à une plus grande précision et évite le problème de sur-apprentissage [36].

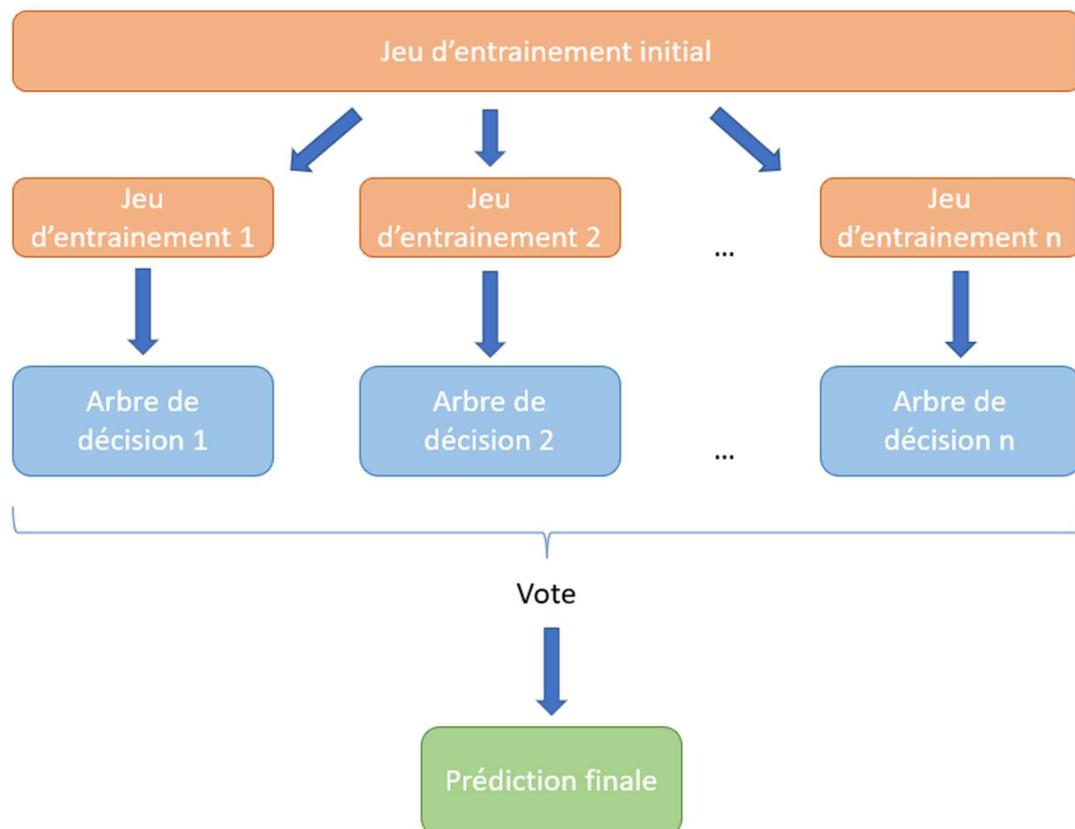


Figure 15 - Schéma forêt aléatoire

Finalement, Random Forest réalise ce que l'on appelle du "Bagging", c'est-à-dire que l'on assemble un grand nombre d'algorithmes avec de faibles performances individuelles pour en créer un beaucoup plus efficace. Les algorithmes de faible performance sont appelés les « weak learners » et le résultat obtenu « strong learner ». Dans un Bagging, il est important de comprendre que chaque algorithme est indépendant l'un vis-à-vis des autres. L'idée est de dire que plusieurs petits algorithmes peuvent être plus performants qu'un seul grand algorithme.

L'avantage de Random Forest est d'être capable de gérer de grands ensembles de données avec une grande dimensionnalité. Il permet d'améliorer la précision du modèle par rapport à un arbre seul et évite le sur-apprentissage.

2.3.2.2.5 Extreme Gradient boosting (XGBoost)

XGBoost est un algorithme relativement récent puisqu'il est le résultat d'un projet de recherche organisé par l'Université de Washington en 2016. Il se base sur les arbres de décision que nous avons vus plus haut et fonctionne sur un principe similaire au Bagging du Random Forest. En effet, il s'agit également d'un algorithme ensembliste qui va utiliser plusieurs autres algorithmes pour parvenir au résultat final. Cependant, là où, lors du Random Forest, les différents algorithmes sont indépendants, le boosting rend les différents weak learners dépendant les uns des autres. Il ne s'agit plus de lancer un ensemble d'algorithmes simultanément mais plutôt de les exécuter successivement. Ainsi, chaque « weak learner » est entraîné pour corriger les erreurs des « weak learner » précédents [39].

XGBoost est une version particulière de l'algorithme du Gradient Boosting. Tout comme lui, l'ensemble des weak learners cherchent à prédire les erreurs, c'est-à-dire les écarts entre les prédictions et la réalité. Chaque weak learner est ainsi entraîné pour prévoir l'erreur de son prédecesseur. Dans le cadre d'une classification, chaque individu dispose d'un poids qui est le même au départ, et qui, si un modèle se trompe, sera augmenté avant d'estimer le modèle suivant (qui prendra donc en compte ces poids). Les prédictions d'erreur sont multipliées par un facteur inférieur à 1 pour réduire la taille des "pas" et augmenter la précision. L'idée derrière cette multiplication est que plusieurs petits pas sont plus précis que quelques grands pas. Par ce procédé, XGBoost cherche à écarter petit à petit les prédictions du modèle de la moyenne afin de les rapprocher de la réalité.

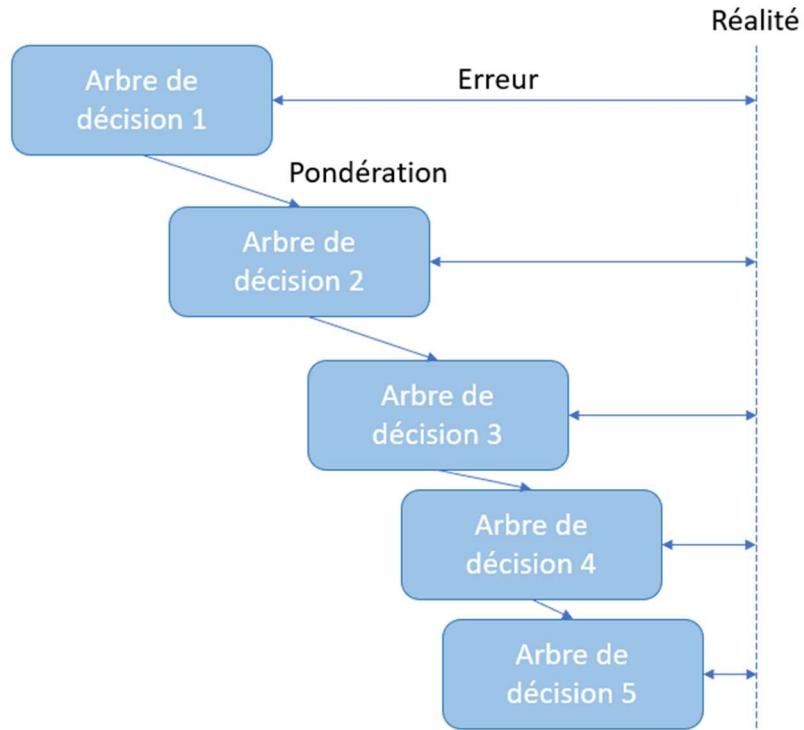


Figure 16 - Schéma XGBoost

La force d'XGBoost par rapport à un Gradient Boosting classique est que celui-ci utilise des arbres décisionnels qu'il punit sévèrement en cas de mauvaise performance. En effet, les arbres qui ne sont pas assez bons sont "élagués". Nous l'avons vu plus haut, cela signifie que l'on va leur couper des branches. XGBoost peut même aller jusqu'à supprimer un arbre complet. Ainsi l'algorithme donne son résultat final en se basant uniquement sur les prédictions des bon weak learners.

2.3.2.2.6 K-Nearest Neighbors (KNN) Algorithm

L'algorithme du KNN est un algorithme supervisé pouvant également être utilisé pour la classification et la régression. Il est toutefois particulièrement efficace dans les problèmes de classifications. Cet algorithme est non paramétrique, ce qui signifie qu'il ne fait aucune hypothèse sur les données sous-jacentes. L'idée de l'algorithme est de considérer que la nouvelle observation appartient à la classe majoritaire parmi ces K plus proches voisins déjà connus. On considère les observations voisines comme étant celles avec la similarité la plus forte avec la nouvelle observation. KNN fait partie de ces algorithmes dit paresseux qui n'apprennent pas immédiatement à partir de l'ensemble d'apprentissage. L'algorithme stocke dans un premier temps les données d'apprentissage puis, au moment de la classification d'une nouvelle observation, va effectuer une action [37].

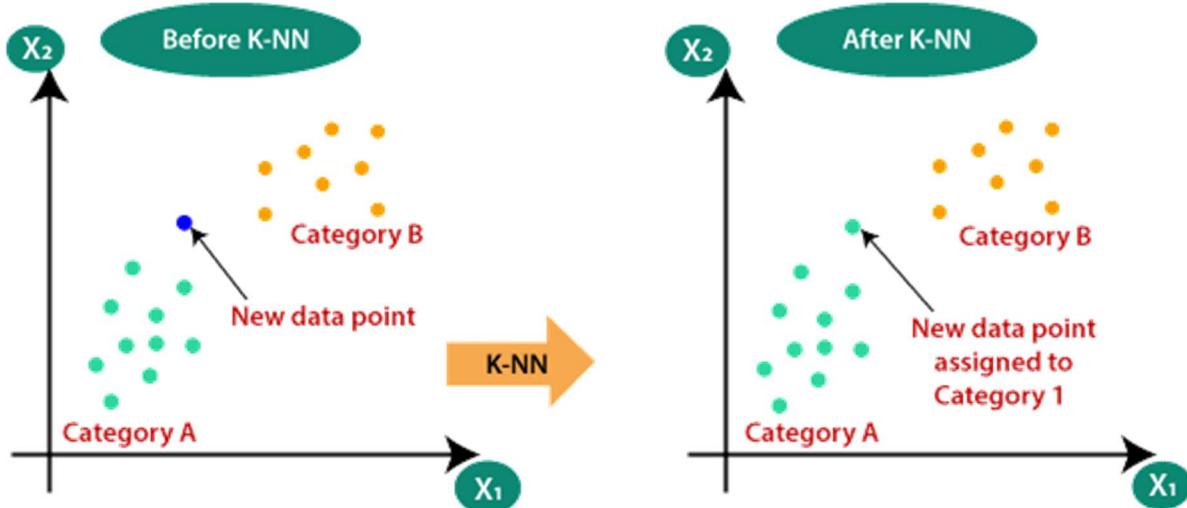


Figure 17 - Schéma KNN

L'algorithme fonctionne de manière simple :

1. Sélectionner le nombre K de voisins
2. Calculer la distance du nombre K de voisins
3. Prendre les K voisins les plus proches de l'observation à classer
4. Parmi ces K voisins, compter le nombre d'observations par classe
5. Attribuer la classe majoritaire à la nouvelle observation

La similarité entre deux observations peut être définie de diverses manières : on peut en effet choisir une distance euclidienne, de Manhattan...etc. La valeur de k n'est pas toujours évidente à trouver puisqu'il n'existe pas de méthode particulière pour déterminer sa valeur optimale. La meilleure approche consiste à tester différentes valeurs et de comparer les erreurs du modèle [37].

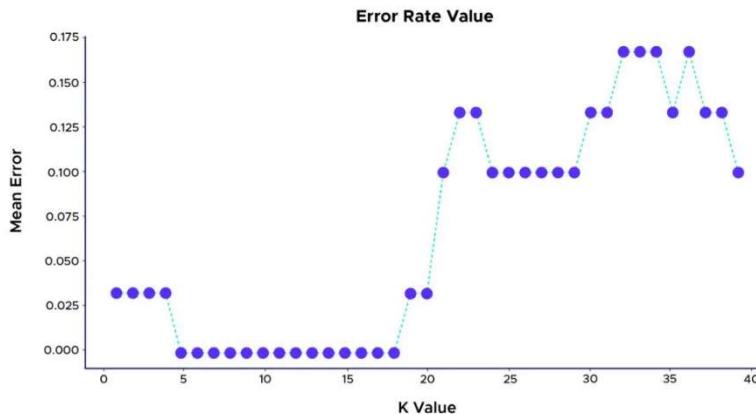


Figure 18 - KNN et comparaison d'erreurs

Ici les meilleures prédictions sont obtenues pour un K entre 5 et 18. Au-delà on peut constater un phénomène de surapprentissage dans lequel l'algorithme n'arrive plus à généraliser le modèle.

L'algorithme du KNN a l'avantage d'être simple à implémenter et robuste au bruit dans les données. Comme d'autres algorithmes il est particulièrement efficace lorsqu'il dispose d'un large jeu de données d'entraînements. Toutefois il nécessite de déterminer la valeur de K, ce qui n'est pas toujours évident et

possède un coût de calcul important puisque pour chaque nouvelle observation, l'algorithme calcul la distance entre l'observation et toutes les autres données du jeu.

2.3.2.2.7 Support Vector Machine (SVM) Algorithm

L'algorithme SVM est un algorithme de classification ayant également quelques applications en régression. Il s'agit donc d'un algorithme supervisé dont l'objectif est de trouver un hyperplan dans un espace à N dimensions qui classe distinctement les points de données. Ces dimensions sont les classes possibles pour notre observation. Ainsi si l'il s'agit d'une classification binaire, l'hyperplan n'est qu'une ligne. En revanche, si le nombre de classes en entrée est de trois, l'hyperplan devient un plan 2D et ainsi de suite [38].

Il existe deux types de SVN :

- Le **SVN linéaire** qui est utilisé lorsque les données sont séparables linéairement. Cela signifie que l'ensemble de données peut être classé en deux classes en utilisant une seule ligne droite. C'est le cas de notre classification pour le turn-over.
- Le **SVN non linéaire** qui à l'inverse intervient lorsque l'ensemble de données ne peut pas être classé à l'aide d'une ligne droite

Dans le cas d'une classification binaire, il peut y avoir plusieurs lignes de décision pour séparer les classes dans un espace à n dimensions. L'enjeu est alors de trouver la meilleure, celle qui aide à classer au mieux les points de données. Cette frontière est appelée l'hyperplan.

Supposons que nous ayons un jeu de données qui a deux balises (vert et bleu) et que le jeu de données à deux caractéristiques x_1 et x_2 . Nous cherchons un modèle capable de classer la paire (x_1, x_2) de coordonnées en vert ou en bleu. Considérez l'image ci-dessous :

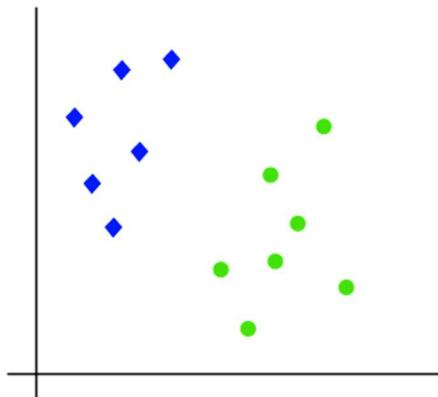


Figure 19 - SVM exemple étape 1

Dans le cas de la classification binaire, on peut séparer le jeu de données en utilisant simplement une ligne droite. Cependant plusieurs lignes peuvent le permettre :

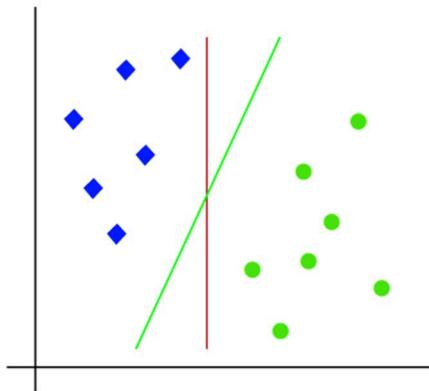


Figure 20 - SVM exemple étape 2

L'algorithme SVM permet donc de trouver la « meilleure ligne » aussi appelé « limite de décision » pour le jeu de données. Afin de trouver ce fameux hyper plan, SVM s'appuie sur le point le plus proche de la ligne pour chacune des deux classes. Ces points permettent de construire des vecteurs supports. La distance entre les vecteurs et l'hyperplan est appelée marge . Finalement le but de SVM est de maximiser cette marge. Il choisira donc la ligne qui maximise la distance entre cette dernière et les vecteurs supports.

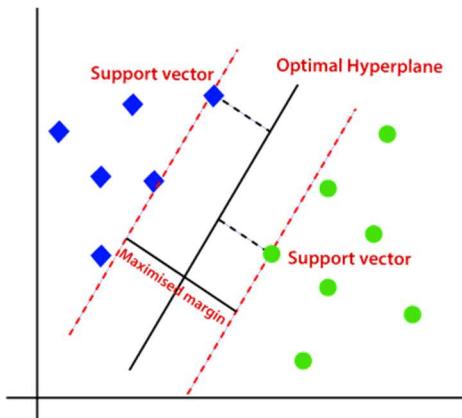


Figure 21 - SVM exemple étape 3

Une fois l'hyperplan déterminé à l'aide du jeu d'apprentissage, les observations sont classées dans l'une ou l'autre des classes.

2.3.2.2.8 Neural Networks (NN)

Le réseau de neurones constitue le dernier algorithme que nous étudierons ici. Il est inspiré du fonctionnement du cerveau humain imitant la façon dont nos neurones biologiques se signalent les uns aux autres. L'un des réseaux de neurones les plus connus est l'algorithme de recherche de Google. Bien que l'on cherche à imiter le fonctionnement de notre cerveau, il ne s'agit en aucune manière de créer une machine totalement autonome capable de se substituer à l'homme. Terminator et autres super-robots peuvent encore attendre un peu.

Les réseaux de neurones sont composés d'une multitude de couches de nœuds (un nœud étant symboliquement un neurone). On y retrouve une couche d'entrée, une ou plusieurs couches cachées intermédiaires et enfin une couche de sortie. Chaque nœud, ou neurone artificiel, dispose d'un poids et d'un seuil ainsi que d'une connexion à un autre nœud. Si la sortie d'un nœud est supérieure à la valeur du seuil, le nœud s'active et envoie les données à la couche suivante du réseau. Sinon, aucune donnée n'est transmise à la couche suivante du réseau [40].

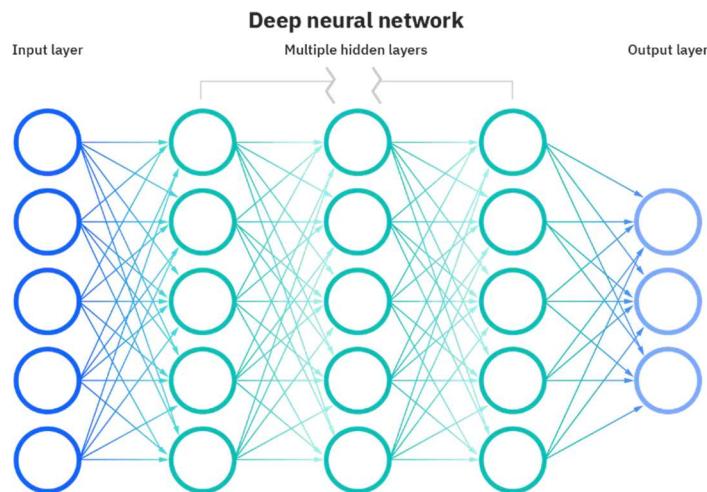


Figure 22 - Schéma réseau de neurones

Le principe d'un neurone est simple, il prend un groupe d'entrées pondérées, applique une fonction d'activation et renvoie une sortie. On peut considérer chaque neurone comme un mini-modèle disposant d'un seuil lui permettant de déterminer si sa sortie doit passer au nœud suivant. Afin de créer cette toile les données entre les nœuds (neurones) et la sortie, on utilise des synapses qui sont comme les routes de notre réseau neuronal. Ils connectent les entrées aux neurones, les neurones aux neurones et les neurones aux sorties. Pour passer d'un neurone à l'autre, on doit donc voyager le long de la synapse en payant un coup (poids) en cours de route. Chaque connexion entre deux neurones a une synapse unique avec un poids unique qui lui est attaché.

Les réseaux de neurones sont entraînés de manière itérative à l'aide de techniques d'optimisation telles que la descente de gradient. Après chaque cycle d'entraînement, une métrique d'erreur est calculée en fonction de la différence entre la prédiction et la cible. Les dérivées de cette métrique d'erreur sont calculées et propagées à travers le réseau par rétropropagation. Les coefficients (poids) de chaque neurone sont ensuite ajustés par rapport à la contribution de l'erreur totale. Ce processus est répété de manière itérative jusqu'à ce que l'erreur réseau tombe en dessous d'un seuil acceptable. Finalement, la couche de sortie reçoit l'entrée de la couche cachée précédente, applique éventuellement une fonction d'activation et renvoie une sortie représentant la prédiction de votre modèle [41].

2.3.3 Les outils informatiques

Afin d'implémenter ces fameux algorithmes, il revient à tout organisme de mettre en place des solutions informatiques lui permettant d'effectuer les calculs et de présenter ces résultats. Dans cette section, nous allons rapidement évoquer les solutions qui s'offrent aux organisations souhaitant développer une expertise en Machine Learning.

2.3.3.1 Notebooks & Languages

Une première option consiste à utiliser ce qu'on appelle des notebooks. Très utilisés dans le milieu universitaire, ces outils permettent d'organiser différents éléments tels que le texte, le code, les images, la sortie, etc. Ils permettent d'enregistrer le processus de réflexion lors de la conception du processus de recherche. Ces solutions ont l'immense avantage d'être très libres, on peut écrire ce que l'on veut et où on le désire. On accède depuis un simple navigateur, sont généralement gratuits et collaboratifs. Parmi les plus utilisés pour les travaux de Machine Learning, on peut citer le Notebook Jupyter, Google Collab ou Kaggle [42] .

Toutefois ces outils ne sont pas à la portée de n'importe qui. Afin d'implémenter des solutions de Machine Learning sur ces notebooks, il est primordial d'être un professionnel de l'analyse de données, et notamment de connaître certains langages de programmations tels que :

- Python
- C/C++
- Java
- R

Il existe d'autres langages plus marginaux utilisés pour le Machine Learning comme Julia, Scala, Ruby, Octave, MATLAB ou SAS.

2.3.3.2 *Les plateformes de Machine Learning*

Bien que les Notebooks soient très appréciés dans le milieu de la recherche, ils ne permettent pas toujours d'obtenir une application industrialisable dans un contexte d'entreprise. Certaines plateformes ont alors développé des logiciels spécialisés permettant d'accompagner les organisations dans la réalisation de leurs projets. Ces applications proposent généralement des packages complets de fonctionnalités allant au-delà des simples besoins d'implémentation d'algorithme. On peut, par exemple, retrouver une brique de visualisation de données et une brique d'intégration. Ces logiciels plus complets permettent de cadrer un peu plus les projets de Machine Learning en fournissant de l'aide à la réalisation pour les développeurs. En revanche, elles ne sont pas gratuites et demandent des coûts d'installation et de maintenance pour les organisations. Vous trouverez ci-dessous les analyses Forrester et Gartner de ces solutions :

Figure 1. Magic Quadrant for Data Science and Machine Learning Platforms



Figure 23 - Magic Quadrant Gartner des plateformes de Machine Learning

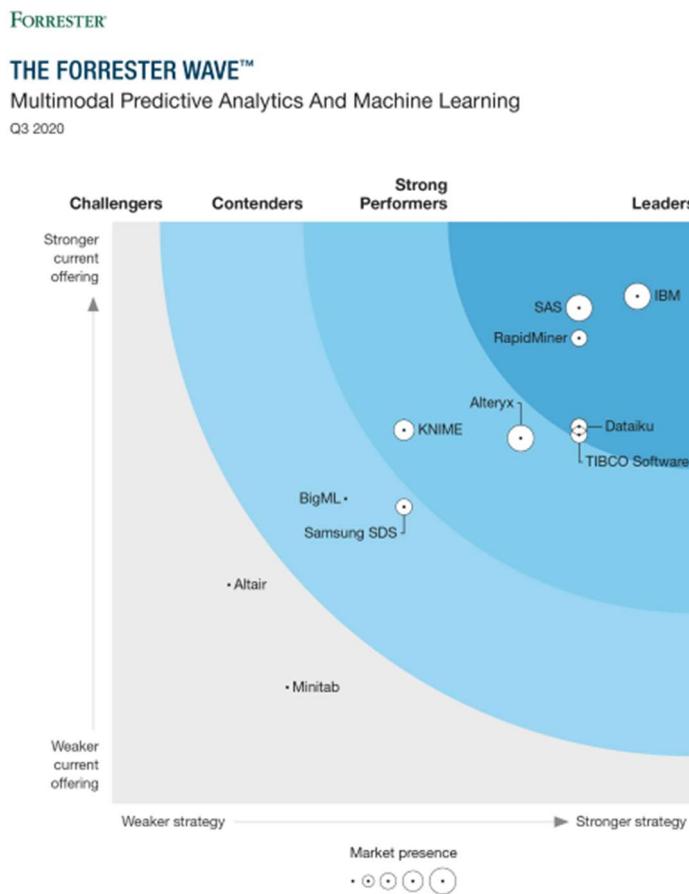


Figure 24 - Forest Wave Forrester des plateformes de Machine Learning

Ces évaluations sont à prendre avec précaution mais traduisent tout de même une certaine tendance sur le marché. Ainsi en fonction des besoins et des moyens de l'organisation, on peut se tourner vers une solution de ce type afin d'implémenter des solutions de Machine Learning. L'avantage de ce type de solution réside dans leur approche simplifiée des problèmes de Machine Learning. Au sein de ces plateformes, les data scientists peuvent être accompagnés et guidés dans leur tâche, ce qui n'est généralement pas le cas dans les notebooks. Cet aspect rend alors le développement de ces solutions plus accessible pour les organisations ne disposant pas forcément d'expert en statistiques.

2.3.3.3 Les solutions self-service

Bien que les plateformes de Machine Learning fassent un premier pas vers les utilisateurs, leur prise en main reste réservée à des professionnels du secteur. Fort de ce constat, certains développeurs de solution informatique ont décidé de proposer des outils permettant de profiter de la puissance du Machine Learning sans forcément disposer de compétences techniques. Ainsi, les solutions Self-services permettent à des utilisateurs ne possédant pas de compétence technique d'implémenter des solutions de Machine Learning [43]. Parmi ces solutions self-service, on retrouve notamment :

- BigML
- H2O Driverless AI
- DataRPM
- DataRobot
- Google Cloud AutoML

L'avantage de ce type de solution est de permettre à des non-initiés de manipuler des solutions de Machine Learning de manière totalement autonome. En revanche, il devient alors difficile d'ajuster les modèles et d'expliquer les résultats. Le professionnel sera capable d'apporter au processus des nuances, des intuitions et une résolution créative de problèmes, là où l'autoML sera beaucoup plus procédurale. Le risque pour ces outils est de les voir devenir des boîtes magiques qui permettent de prédire tout et n'importe quoi sans vraiment savoir de quoi il retourne.

III. Mise en application : le cas d'IBM

Sommaire

III.	Mise en application : le cas d'IBM	50
3.1	Introduction.....	50
3.1.1	Identification du besoin.....	50
3.1.2	Choix de l'outil et collecte des données.....	51
3.2	EDA (Exploratory Data Analysis).....	51
3.2.1	Découverte du jeu de données	52
3.2.2	Exploration des données	55
3.3	Choix des modèles et préparation des données	62
3.3.1	Les modèles.....	62
3.3.2	Data processing	62
3.4	Implémentation des modèles.....	64
3.4.1	Etablir une Baseline.....	64
3.4.2	Régression Logistique.....	64
3.4.3	Random Forest	66
3.4.4	XGBoost	70
3.5	Mise en production	72
3.5.1	Quel modèle choisir ?	72
3.5.2	Quel sont les gains ?	73
3.6	Pour aller plus loin.....	75
3.6.1	Utilisation et alternatives	76
3.6.2	Proposition d'une méthode pour l'organisation	76

3.1 Introduction

Fort de cet état de l'art, nous avons désormais toutes les clefs en main pour réaliser une analyse prédictive sur les données RH de notre organisation. Nous sommes en effet capables de construire un modèle de Machine Learning nous permettant d'anticiper le départ d'un collaborateur et ce de manière autonome. Dans cette partie, nous allons étudier une solution envisageable pour la mise en place d'un tel modèle. Cette solution ne se veut pas suprême et absolue. Il s'agit d'une manière de réaliser l'implémentation parmi d'autres. Il est même probable qu'un statisticien ou un data scientiste aguerri puisse pousser l'implémentation encore plus loin. Cette approche se veut plutôt comme une méthode, un guide permettant de structurer la création d'une telle solution au sein d'une équipe métier.

3.1.1 Identification du besoin

Nous l'avons vu plus haut dans ce document, la première étape d'un projet d'analyse prédictive consiste à correctement identifier le besoin auquel il répond. Dans notre cas, nous sommes dans un contexte de

rétention d'employé au sein d'une organisation. Les employés étant l'épine dorsale de l'organisation, les performances de cette dernière dépendent fortement de la qualité de ses employés. Les défis qu'une organisation doit relever en raison du départ d'employés peuvent être les suivants :

- Perte d'employés expérimentés
- Impact sur la productivité
- Impact sur les bénéfices
- Coût de recrutements
- Coût en temps et en argent de la formation de nouveaux employés

Ce projet vise donc à améliorer la fidélisation des employés en réduisant les coûts liés au turn-over. Bien que ce ne soit pas toujours évident, il peut donc être intéressant de chiffrer l'ensemble de ces éléments afin d'obtenir un moyen d'évaluer la plus-value du projet.

Notre modèle doit permettre aux collaborateurs des ressources humaines de répondre à des questions métiers telles que :

- Quels sont les employés les plus susceptibles de partir dans l'année ?
- Quels sont les employés les moins fidèles ?
- Quelles sont les caractéristiques qui poussent les employés à quitter l'organisation ?
- Quels sont les facteurs déterminants d'une démission ?

3.1.2 Choix de l'outil et collecte des données

Une fois le contexte clairement énoncé, il nous faut choisir l'outil avec lequel nous tenterons de répondre au problème. Comme vu plus haut, différentes plateformes s'offrent à nous. Dans cet ouvrage, nous travaillerons avec une solution de notebook Python sur Google Collab gratuite nous permettant beaucoup de libertés sur l'implémentation du code ainsi que peu d'apprentissage. Il suffit de connaître le langage pour commencer à développer.

On peut finalement passer à la collecte des données. Nous n'insisterons jamais assez sur l'importance de disposer de données propres et qualifiées pour les analyses. Bien que la plupart des organisations disposent déjà de données collectées de façon routinière, les données brutes ne peuvent généralement pas à elles seules fournir des informations utiles. Il faut dans un premier temps s'assurer de la qualité des données disponibles. On peut par exemple s'intéresser à leur complétude, leur exhaustivité, leur fraîcheur ou encore leur disponibilité. Disposer d'un jeu de données propre et fiable permet d'assurer plus sereinement dans l'analyse prédictive.

Dans la suite de ce document, nous allons utiliser un jeu de données fictif créé par les data scientists d'IBM et disponible gratuitement sur la plateforme en ligne Kaggle. Une organisation se doit d'utiliser ses propres données récupérées depuis son datawarehouse, son datalake ou bien même directement depuis son système d'information des ressources humaines. Il est important de disposer d'un jeu de données relativement important et d'une représentation des classes correcte.

3.2 EDA (Exploratory Data Analysis)

Dans cette section, nous explorons l'ensemble de données du jeu en examinant les types de données, les distributions des caractéristiques, les corrélations entre les caractéristiques ou encore les anomalies.

L'objectif est de se familiariser avec le jeu de données afin de commencer à comprendre les tendances qu'il peut nous révéler.

3.2.1 Découverte du jeu de données

Nous partons du principe que l'analyste qui réalise le projet n'est pas forcément issu du métier d'où provient la demande. Ce dernier peut tout à fait débarquer sur un jeu de données dont il ignore absolument tout. Il doit donc se faire une idée de la ressource dont il dispose afin de commencer à entrevoir les possibilités qui s'offrent à lui. Les premières analyses que nous pouvons donc réaliser concernent la structure de notre dataset. De combien de données dispose-t-on, de quoi s'agit-il et à quoi ressemble-t-elles ?

```
# taille de l'échantillon
data.shape
```

↳ (1470, 35)

Figure 25 - Taille du jeu de données

Nous disposons donc de 1470 lignes et de 35 colonnes. Chacune de ces colonnes représente une variable tracée par l'organisation telle que le salaire, le statut marital ou le nom du poste. On peut d'ores et déjà regarder un aperçu du jeu de données :

# échantillon de données data.head()												
	Age	Attrition	BusinessTravel	DailyRate	Department	DistanceFromHome	Education	EducationField	EmployeeCount	EmployeeNumber	...	
0	41	Yes	Travel_Rarely	1102	Sales	1	2	Life Sciences	1	1	...	
1	49	No	Travel_Frequently	279	Research & Development	8	1	Life Sciences	1	2	...	
2	37	Yes	Travel_Rarely	1373	Research & Development	2	2	Other	1	4	...	
3	33	No	Travel_Frequently	1392	Research & Development	3	4	Life Sciences	1	5	...	
4	27	No	Travel_Rarely	591	Research & Development	2	1	Medical	1	7	...	

5 rows × 35 columns

Figure 26 - Echantillon du jeu de données

Cet extrait donne à l'analyste une vision des données brutes dont il dispose. Notre variable cible est identifiée comme étant « Attrition ». Elle permet de déterminer si un employé a quitté l'organisation ou non. Au sein du dataset que nous étudions, nous retrouvons les variables suivantes :

- AGE
- ATTRITION
- BUSINESS TRAVEL
- DAILY RATE
- DEPARTMENT
- DISTANCE FROM HOME
- EDUCATION
- EDUCATION FIELD
- EMPLOYEE COUNT
- EMPLOYEE NUMBER
- ENVIROMENT SATISFACTION
- MONTHLY INCOME
- MONTHLY RATE
- NUMCOMPANIES WORKED
- OVER 18
- OVERTIME
- PERCENT SALARY HIKE
- PERFORMANCE RATING
- RELATIONS SATISFACTION
- STANDARD HOURS
- STOCK OPTIONS LEVEL
- TOTAL WORKING YEARS

- GENDER
- HOURLY RATE
- JOB INVOLVEMENT
- JOB LEVEL
- JOB ROLE
- JOB SATISFACTION
- MARITAL STATUS
- TRAINING TIMES LAST YEAR
- WORK LIFE BALANCE
- YEARS AT COMPANY
- YEARS IN CURRENT ROLE
- YEARS SINCE LAST PROMOTION
- YEARS WITH CURRENT MANAGER

Chaque organisation dispose de ses propres variables qui s'adaptent à son fonctionnement. On peut ici évoquer que plus nous disposons de variables, plus les analyses peuvent être affinées. Il ne s'agit toutefois pas de collecter sans réfléchir, mais de s'interroger sur les variables qu'il serait intéressant de suivre. Nous pouvons remarquer ici que nous ne disposons pas d'information sur le motif de départ du collaborateur ni sur la valeur que celui-ci a pour l'entreprise.

Nous pouvons finalement chercher à afficher divers statistiques concernant ces variables de manière à commencer à cerner notre population.

```
[ ] # statistiques de base
data.describe()
```

	Age	DailyRate	DistanceFromHome	Education	EmployeeCount	EmployeeNumber	EnvironmentSatisfaction	HourlyRate	JobInvolvement	JobLevel	...
count	1470.000000	1470.000000	1470.000000	1470.000000	1470.0	1470.000000	1470.000000	1470.000000	1470.000000	1470.000000	...
mean	36.923810	802.485714	9.192517	2.912925	1.0	1024.865306	2.721769	65.891156	2.729932	2.063946	...
std	9.135373	403.509100	8.106864	1.024165	0.0	602.024335	1.093082	20.329428	0.711561	1.106940	...
min	18.000000	102.000000	1.000000	1.000000	1.0	1.000000	1.000000	30.000000	1.000000	1.000000	...
25%	30.000000	465.000000	2.000000	2.000000	1.0	491.250000	2.000000	48.000000	2.000000	1.000000	...
50%	36.000000	802.000000	7.000000	3.000000	1.0	1020.500000	3.000000	66.000000	3.000000	2.000000	...
75%	43.000000	1157.000000	14.000000	4.000000	1.0	1555.750000	4.000000	83.750000	3.000000	3.000000	...
max	60.000000	1499.000000	29.000000	5.000000	1.0	2068.000000	4.000000	100.000000	4.000000	5.000000	...

8 rows × 26 columns

Figure 27 - Description statistique du jeu de données

Ces premières analyses nous apprennent par exemple que la moyenne d'âge au sein de l'organisation est de 37 ans et que la satisfaction va de 1 à 4.

Nous pouvons pousser les analyses un peu plus loin en cherchant par exemple à qualifier la qualité des données. On peut s'intéresser à savoir combien de valeurs distinctes chaque variable admet, si le jeu contient des doublons ou bien des valeurs nulles.

```
▶ # découverte des doublons
data.duplicated().sum()
```

0

Figure 28 - Recherche de doublons

Le jeu de données ne souffre pas de la présence de doublons.

```
[ ] # découverte des valeurs manquantes
data.isnull().sum()
```

Age	0
Attrition	0
BusinessTravel	0
DailyRate	0
Department	0

Figure 29 - Recherche de valeurs manquantes

Il ne subit pas non plus la présence de valeurs manquantes. La réalité est souvent bien différente. L'informaticien doit alors aviser et, en concertation avec le propriétaire de la donnée, trouver une manière de pallier au problème. Doit-on remplacer les valeurs manquantes par des estimations (ex : la moyenne) ? Doit-on simplement supprimer les données ?

Intéressons-nous maintenant au nombre de valeurs distincts par variables :

```
# découverte du nombre de valeurs distincts par variables
uniqueValue_df = pd.DataFrame({'Column':[], 'Unique value number':[]})

for column in data.columns:
    new_row = {'Column':column, 'Unique value number':data[column].nunique()}
    uniqueValue_df = uniqueValue_df.append(new_row, ignore_index=True)

display(uniqueValue_df)
```

	Column	Unique value number
0	Age	43.0
1	Attrition	2.0
2	BusinessTravel	3.0
3	DailyRate	0.0
8	EmployeeCount	1.0
9	EmployeeNumber	1470.0
21	Over18	1.0
26	StandardHours	1.0

Figure 30 - Découverte des valeurs distincts

On peut remarquer ici que les variables 'EmployeeCount', 'Over18', 'StandardHours' ont seulement une valeur unique. 'EmployeeNumber' quant à lui dispose 1470 valeurs uniques soit une valeur unique par ligne. Ainsi ces caractéristiques ne semblent pas apporter d'informations particulières et nous pouvons donc les écarter du dataset.

```
[ ] # drop columns
data.drop(['EmployeeCount', 'EmployeeNumber', 'Over18', 'StandardHours'], axis="columns", inplace=True)
```

Figure 31 - Suppression des colonnes sans informations

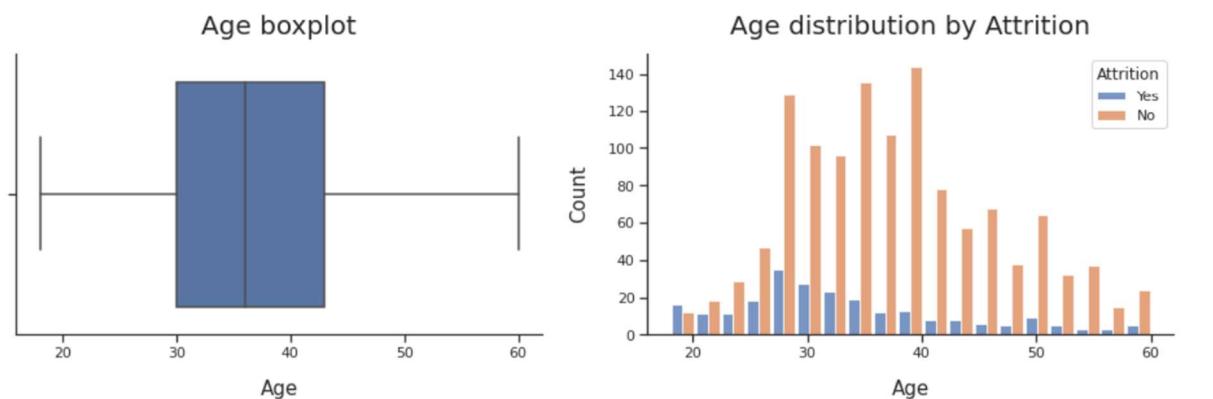
Cette première étape de découverte nous aura finalement permis de définir la structure de notre jeu de données. Nous disposons de 1470 lignes et 35 colonnes sans valeurs manquantes ni doublons. Nous disposons de données qualitatives et quantitatives dont certaines n'apportent pas d'informations.

3.2.2 Exploration des données

Nous avons découvert plus haut que la variable « Attrition » était notre valeur cible. Nous allons dans l'exploration chercher à déterminer comment cette dernière se comporte vis-à-vis de l'ensemble de nos variables et quelles sont les corrélations que l'on peut d'ores et déjà relever.

On constate que 16,1% des individus du jeu de données sont considérés comme ayant quitté l'organisation. Nous sommes donc en présence d'un contexte de classes déséquilibrées dans lequel l'une d'elle est sur représentée par rapport à l'autre. On peut par la suite s'intéresser à cette distribution en fonction des variables quantitatives de notre modèle. Parmi ces dernières, on retient par exemple les analyses descriptives suivantes :

```
=====
Age:
Minimum: 18, Maximum: 60, Mean: 36.92, Std: 9.14
```



L'organisation dispose d'un effectif allant de 18 à 60 ans avec une forte concentration entre 30 et 40 ans. Les jeunes trentenaires semblent par ailleurs les plus sujets à l'attrition. Ces profils sont donc à surveiller de près.

```
=====
MonthlyIncome:
Minimum: 1009, Maximum: 19999, Mean: 6502.93, Std: 4707.96
```

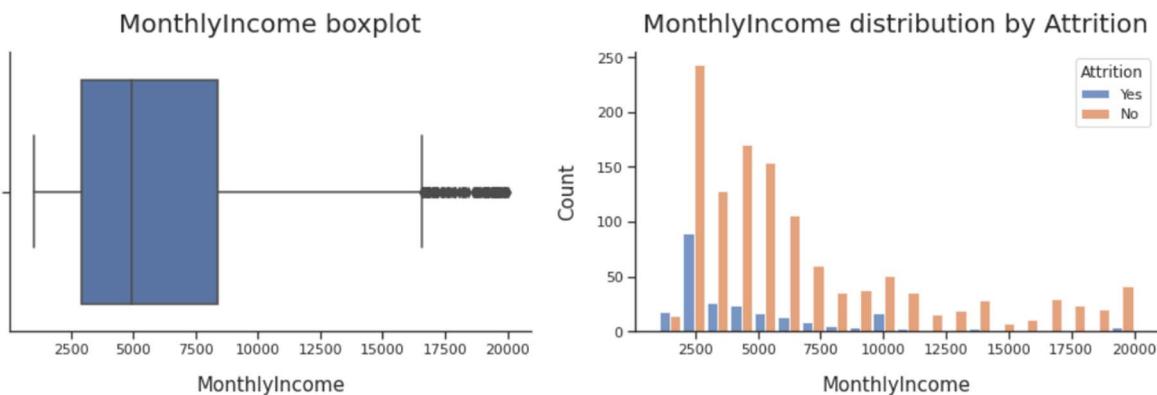


Figure 33 - Découverte du revenu mensuel

En analysant les salaires mensuels, on se rend d'abord compte grâce à la boîte à moustaches de la présence de valeurs aberrantes. En effet, certains salaires semblent anormalement élevés par rapport aux autres. Quand on s'intéresse à l'attrition, on peut noter que celle-ci intervient particulièrement sur les employés disposant des salaires les plus faibles.

```
=====
YearsWithCurrManager:
Minimum: 0, Maximum: 17, Mean: 4.12, Std: 3.57
```

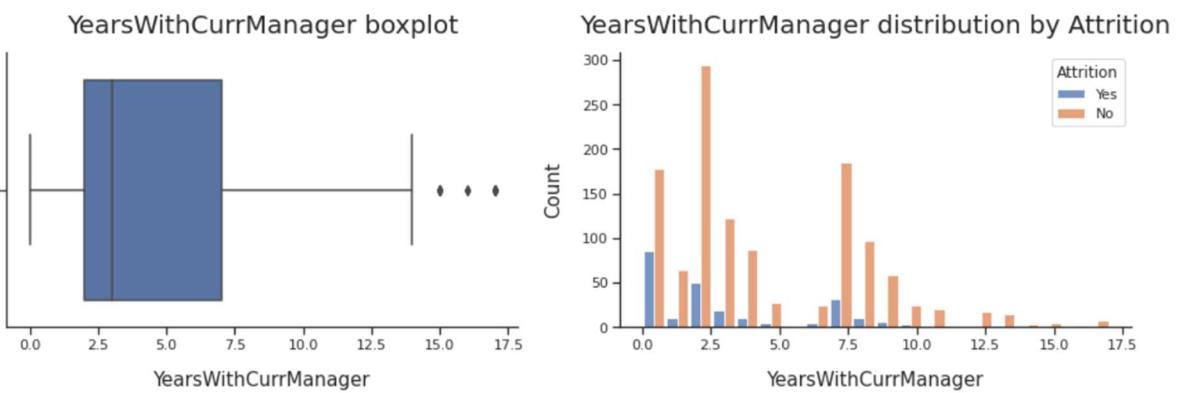


Figure 34 - Découverte des années avec le manager actuel

On remarque que certains collaborateurs semblent rester avec le même manager plus longtemps qu'à l'accoutumé (plus de 15 ans). L'attrition quant-à-elle est particulièrement élevée chez les employés disposant d'un nouveau manager. Il peut s'agir, soit de nouveaux collaborateurs fraîchement recrutés, soit de collaborateurs déjà en place pour qui le management a changé. Deux autres piques d'attrition à 2 et 7 ans peuvent être révélateurs d'une volonté d'évolution de la part du collaborateur.

=====
RelationshipSatisfaction:
Minimum: 1, Maximum: 4, Mean: 2.71, Std: 1.08

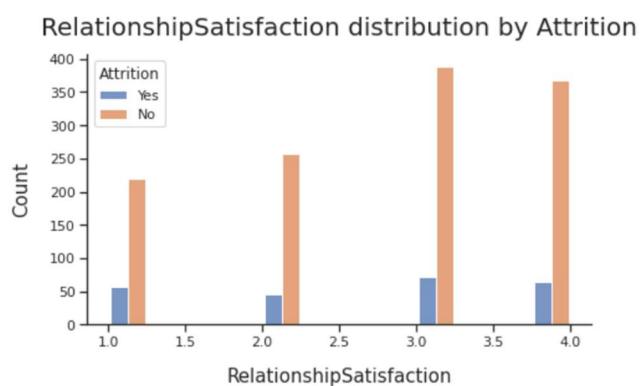
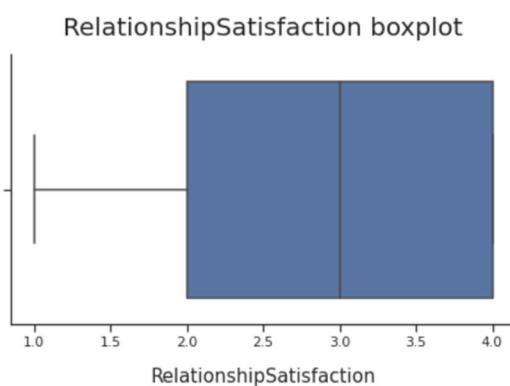


Figure 35 - Découverte de la satisfaction des relations

Enfin, certaines variables quantitatives comme la satisfaction des relations au travail ne semble pas ou très peu influencer l'attrition.

Intéressons-nous à présent aux variables catégorielles :

=====
EducationField:

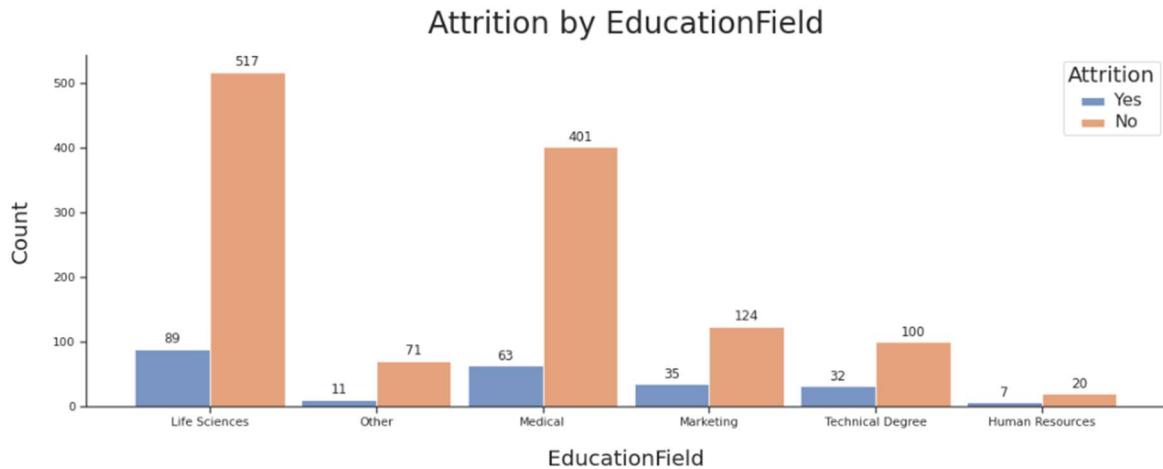


Figure 36 - Découverte du domaine d'études

La variable « EducationField » nous permet de comprendre que l'organisation est majoritairement constituée de personnel médical ou issus d'un cursus en sciences de la vie. Lorsque l'on s'intéresse au taux d'attrition au sein de ces différentes catégories, on constate que celui-ci est plus élevé chez les collaborateurs issus des fonctions supports de l'organisation.

frequency of attrition (%)	
Human Resources	25.93
Technical Degree	24.24
Marketing	22.01
Life Sciences	14.69
Medical	13.58
Other	13.41

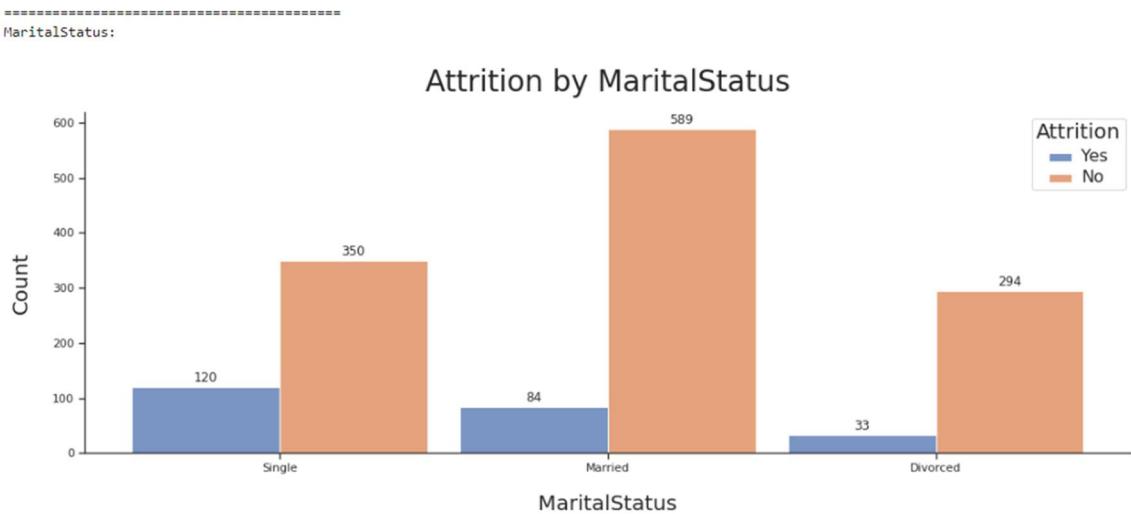


Figure 37 - Découverte du statut marital

Le statut marital des collaborateurs nous apprend que les employés qui sont seuls sont plus enclins à quitter l'organisation que les autres. En effet, près de 25% des collaborateurs classés « Single » ont quitté l'organisation.

	frequency of attrition (%)
Single	25.53
Married	12.48
Divorced	10.09

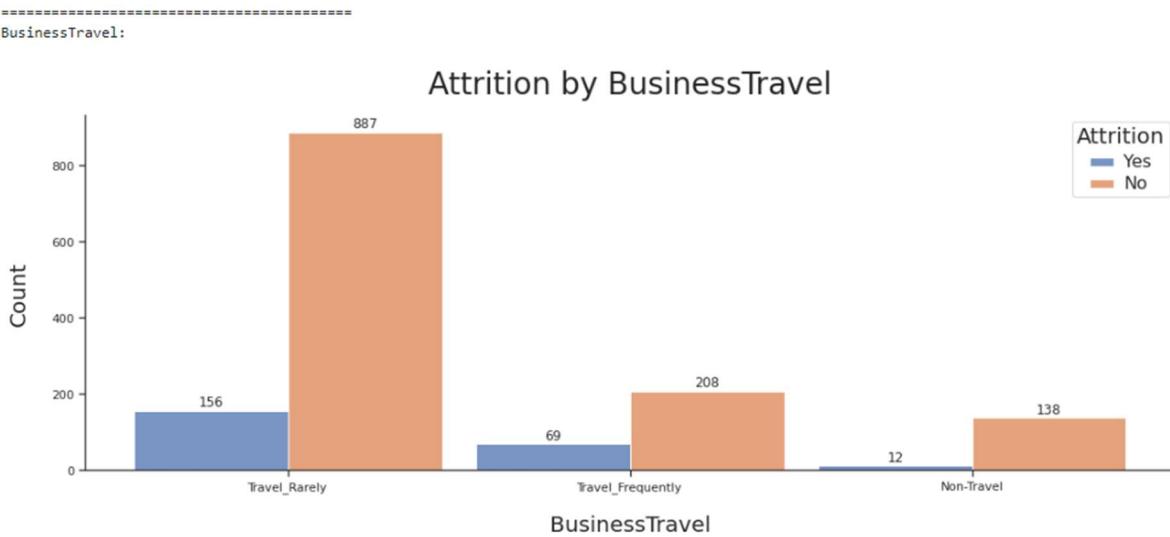


Figure 38 - Découverte de la fréquence de voyage

Une analyse descriptive de l'attribut « BusinessTravel » nous apprend que la majorité des employés est amené à voyager. On remarque toutefois que plus la fréquence de voyage est élevée, plus le taux d'attrition augmente.

	frequency of attrition (%)
Travel_Frequently	24.91
Travel_Rarely	14.96
Non-Travel	8.00

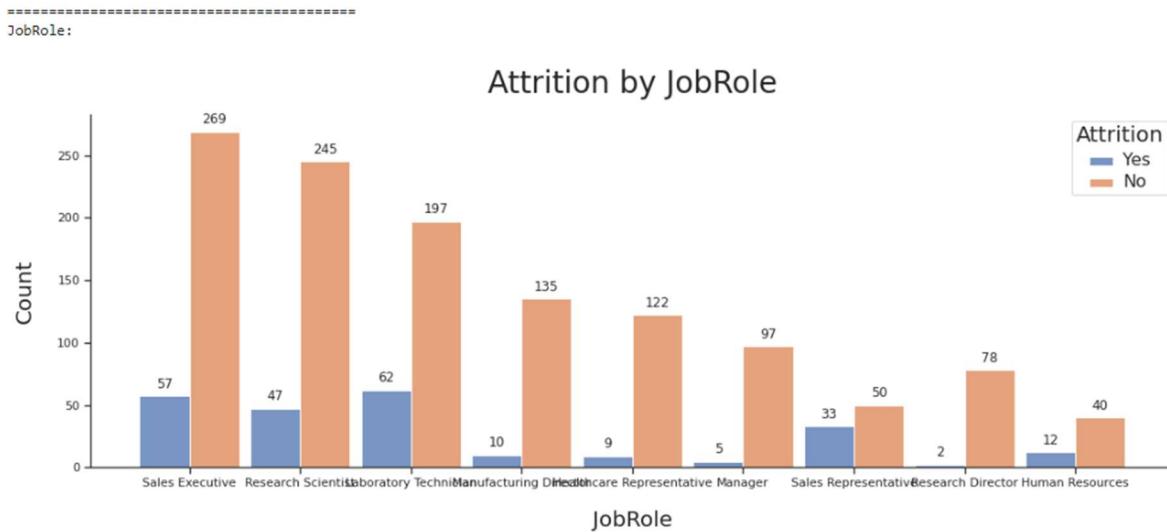


Figure 39 - Découverte de l'emploi

Au sein de l'organisation, divers rôles cohabitent correspondant aux différents postes que l'on peut retrouver. Le métier le plus présent est celui de chargé de ventes suivis de près par les chercheurs en sciences. Quand on s'intéresse à l'attrition, on remarque que certains types d'emploi ne sont quasiment pas touchés tels que les directeurs de recherche, les managers, les représentants de santé et les directeurs de la fabrication. Dans l'ensemble il s'agit de postes à responsabilités, piliers de l'organisation. Toutefois, les représentants des ventes échappent à cette tendance avec près de 40% de l'effectif qui a quitté l'organisation.

	frequency of attrition (%)
Sales Representative	39.76
Laboratory Technician	23.94
Human Resources	23.08
Sales Executive	17.48
Research Scientist	16.10
Manufacturing Director	6.90
Healthcare Representative	6.87
Manager	4.90
Research Director	2.50

On peut également effectuer des analyses en croisant plusieurs variables. On peut par exemple se rendre compte ici que la moyenne des salaires des collaborateurs ayant quitté l'organisation est nettement inférieure à celle des collaborateurs qui sont restés.

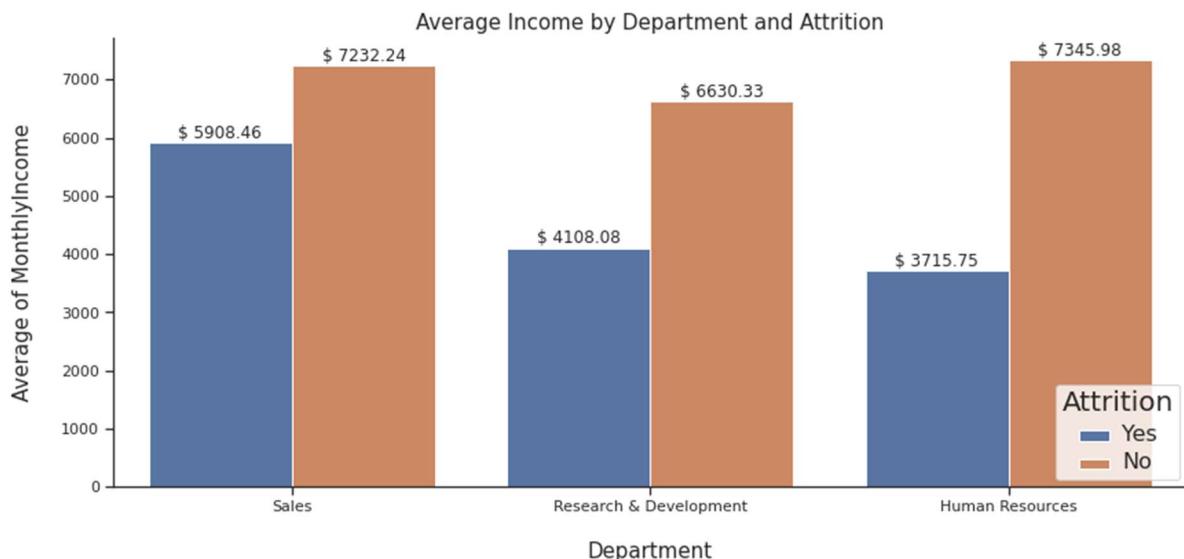


Figure 40 - Découverte du département

Toutes ces informations sont finalement à visée descriptive. On cherche ici les premiers éléments de réponses permettant d'expliquer le départ des employés. Nous avons étudié ensemble 8 des 34 variables explicatives de l'attrition dont nous disposons. Ces représentations graphiques du jeu de données peuvent tout à fait constituer un rapport de monitoring de l'effectif dans lequel les responsables des ressources humaines peuvent détecter des tendances au sein de l'entreprise. La mise en place d'une telle solution d'analyse descriptive est un premier pas vers une activité pilotée par les données. Finalement, grâce à une analyse descriptive complète du jeu de données, on peut tirer les informations suivantes :

- Les jeunes générations sont plus enclines à changer d'employeurs.
- Les collaborateurs disposant d'un niveau d'emploi faible, un revenu mensuel faible, un nombre d'années dans de la société faible ou un nombre total d'années de travail faibles sont plus susceptibles de quitter l'organisation.
- Les travailleurs qui voyagent beaucoup sont plus susceptibles de démissionner que les autres employés.
- Les employés du département « Recherche et Développement » ont plus tendance à rester que les travailleurs des autres départements.
- Les travailleurs ayant un diplôme en ressources humaines ou un diplôme technique sont plus susceptibles de démissionner que les employés des autres domaines de formation.
- Les travailleurs des catégories « Technicien de laboratoire », « Représentant des ventes » et « Ressources humaines » sont plus susceptibles de démissionner que les travailleurs des autres catégories.
- Les travailleurs qui ont un statut marital de célibataire sont plus susceptibles de démissionner que ceux qui sont mariés ou divorcés.

- Les employés qui travaillent plus d'heures ont plus tendance à démissionner que les autres.
- Les employés qui travaillent avec un nouveau manager sont plus enclins à quitter l'organisation que les autres.
- Les représentants des ventes sont très touchés par le turnover.
- Il semble que les variables liées à la satisfaction à l'égard de l'environnement, la satisfaction à l'égard du travail, l'évaluation du rendement et la satisfaction à l'égard des relations n'aient pas un grand impact sur la détermination de l'attrition des employés.

En plus de ces analyses descriptives, on peut s'intéresser à dresser la matrice de corrélation des variables de notre modèle. Cette matrice mesure le degré de relation linéaire qui existe entre chaque paire de variables. Les valeurs de corrélation peuvent être comprises entre -1 et +1. Si les deux variables ont tendance à augmenter et à diminuer en même temps, alors la valeur de corrélation est positive. Si lorsqu'une valeur évolue positivement l'autre évolue négativement alors la valeur de corrélation est négative.

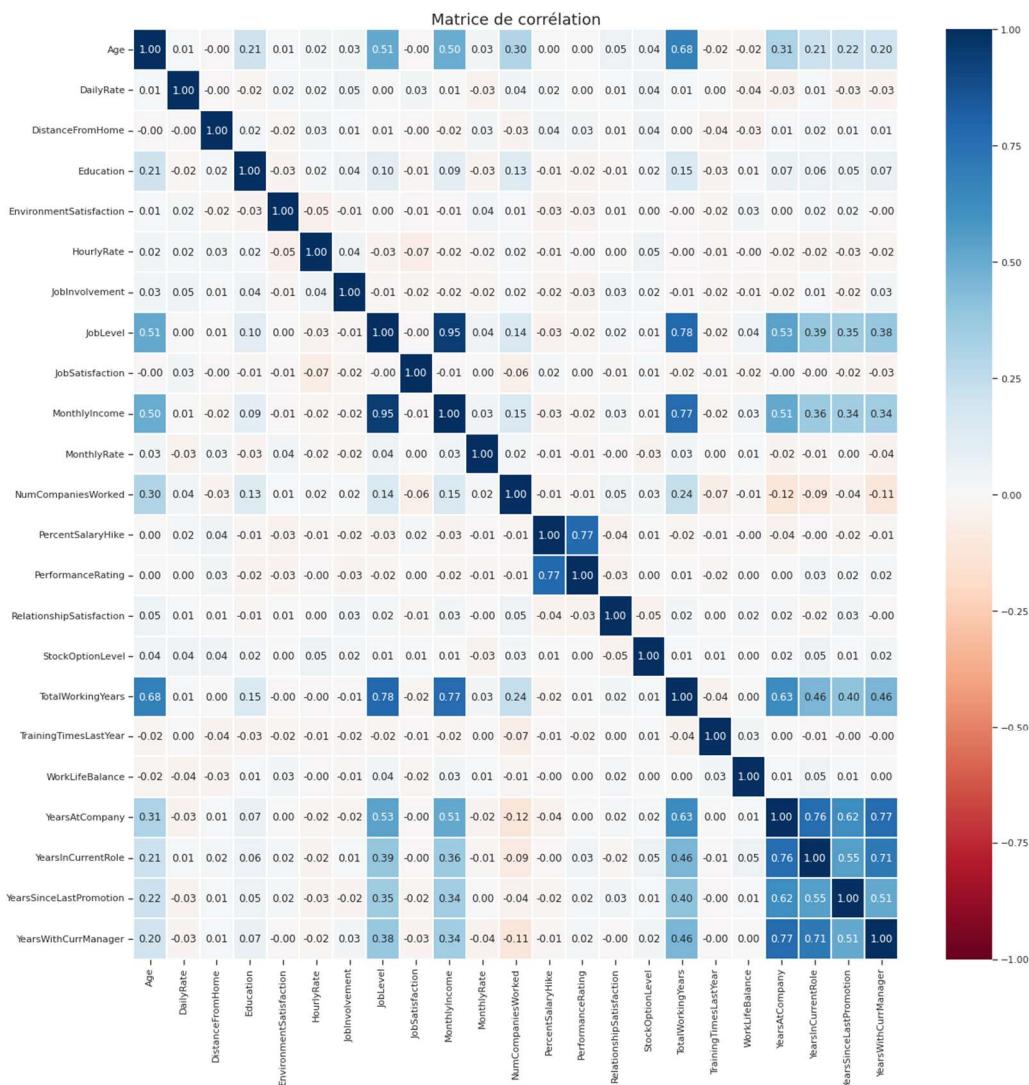


Figure 41 - Matrice de corrélation

Cette matrice de corrélation peut finalement nous permettre de comprendre que :

- Plus le nombre total d'années de travail est élevé, plus le revenu mensuel d'un employé est élevé.
- Plus le pourcentage d'augmentation de salaire est élevé, plus la note de performance est élevée.
- Plus le nombre d'années passées avec le manager actuel est élevé, plus le nombre d'années écoulées depuis la dernière promotion est élevé.
- Plus l'âge est élevé, plus le revenu mensuel est élevé.

De manière globale, on constate qu'un grand nombre de nos variables semblent être faiblement corrélées entre elles. C'est une bonne nouvelle puisque, lors de la création d'un modèle prédictif, il est préférable d'entraîner un modèle avec des caractéristiques qui ne sont pas ou peu corrélées entre elles afin de ne pas avoir à traiter des caractéristiques redondantes

3.3 Choix des modèles et préparation des données

3.3.1 Les modèles

Nous avons vu dans l'état de l'art qu'il était possible d'utiliser de nombreux algorithmes afin de résoudre les problèmes de classification. Au sein de cette solution, j'ai décidé d'en implémenter 3 spécifiquement :

- La **Régression Logistique** car c'est un algorithme classique de classification, couramment utilisé dans les problèmes de prédiction de turn-over. Elle utilise une hypothèse d'indépendance des variables ce qui semble être relativement le cas ici.
- Le **Random Forest** afin d'essayer un algorithme à base d'arbres de décision. De plus son caractère ensembliste est censé nous apporter robustesse et scalabilité.
- Enfin **XGBoost** en raison de sa forte popularité au sein des défis et concours de Machine Learning.

L'objectif est de montrer les différents résultats que l'on peut obtenir et d'implémenter un premier code de comparaison d'algorithme. En fonction des résultats, nous finirons par choisir l'un des 3 algorithmes qui sera mis en production au sein de l'organisation.

Nous aurions tout à fait pu en implémenter d'autres afin d'avoir une plus grande vision des possibles. J'ai ici simplement choisi d'en implémenter 3 comme nous aurions pu en implémenter 5 ou 10. Plus on teste de modèles, plus on s'assure qu'il n'existe pas une solution plus pertinente pour notre problème. Toutefois, implémenter d'autres algorithmes demande du temps et de l'investissement.

3.3.2 Data processing

Une fois nos modèles choisis, il nous faut préparer nos données. En effet, nous ne sommes pas encore capables de simplement alimenter les algorithmes avec des données brutes et leur demander de renvoyer une réponse. Nous devrons apporter quelques modifications mineures pour mettre nos données en termes compréhensibles par machine.

3.3.2.1 One Hot Encoding

La première manipulation que nous allons effectuer consiste à encoder les variables catégorielles en une représentation numérique sans ordres arbitraires. Ce que les ordinateurs connaissent le mieux, ce sont les chiffres et pour le Machine Learning, nous devons les prendre en compte. Ainsi, grâce à la technique du

« One Hot Encoding », nous allons par exemple passer d'une variable comme « MaritalStatus » à 3 variables distinct :

MaritalStatus	MaritalStatus_Single	MaritalStatus_Married	MaritalStatus_Divorced
Single	1	0	0
Married	0	1	0
Divorced	0	0	1

Grâce à ce procédé, nous démultiplions le nombre de variables et les rendons utilisable par nos différents algorithmes.

```
[62] data.shape
(1470, 136)
```

Figure 42 - Taille de l'échantillon

3.3.2.2 Scalarisation

La seconde manipulation que nous effectuons consiste à scalariser nos données à l'aide du `standardScaler()` de Sklean. Lors de cette transformation, on retire la moyenne et on divise par l'écart-type de l'échantillon, permettant d'avoir une variance unitaire. Ce procédé permet de ramener les données sur une même échelle de grandeur sans perdre d'information.

3.3.2.3 Découpage du jeu de données

Une fois nos données encodées et scalarisées, il est maintenant temps de les découper en jeu d'entraînement et de test. Ici, nous avons décidé d'utiliser 80% du jeu de données pour l'entraînement et 20% pour les tests.

```
# définition du jeu de test et du jeu d'entraînement
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42, stratify=y)
```

Figure 43 - Définition du jeu de test et d'entraînement

Comme nous sommes dans un contexte de données déséquilibrés, nous spécifions le paramètre « `stratify` » à vrai. Cela permet de répartir les observations selon le principe de stratification que nous avons évoqué plus haut.

```
Training Features Shape: (1176, 135)
Training Labels Shape: (1176,)

Testing Features Shape: (294, 135)
Testing Labels Shape: (294,)
```

Figure 44 - Taille du jeu de test et d'entraînement

Finalement, notre jeu d'entraînement contient 1176 observations et tandis que notre jeu de test en dispose de 294.

3.4 Implémentation des modèles

Une fois nos données prêtes pour nos algorithmes, nous pouvons créer nos modèles, les entraîner et les tester. Pendant l'entraînement, nous laissons le modèle « voir » les réponses, c'est-à-dire si l'employé part réellement ou non, afin qu'il puisse apprendre à prédire l'attrition depuis ces caractéristiques. On s'attend à ce qu'il y ait une certaine relation entre toutes les caractéristiques et la valeur cible, et le travail du modèle consiste à apprendre cette relation pendant la formation.

Une fois entraîné, on demande au modèle de prédire les classes du jeu de test. Ayant les réponses réelles de l'ensemble de test (mais pas l'algorithme) nous pouvons alors comparer les prédictions avec la réalité et juger de la précision du modèle.

3.4.1 Etablir une Baseline

Comme nous l'avons étudié plus haut, il peut être intéressant au début de notre projet de machine learning, d'établir une Baseline nous servant de repère pour évaluer nos modèles. Dans notre cas, nous ne disposons pas d'experts métier nous informant sur les performances réelles de la prédiction de turn-over par les ressources humaines, ni même si elles existent. Ainsi pour établir notre Baseline, nous pouvons considérer le fait de ne prédire aucun départ. De cette manière nous pouvons considérer les scores suivants :

```
=====
Jeu d'entraînement :
=====

Taux de restants: 83.84%
Taux de sortants: 16.16%

=====
Jeu de test :
=====

Taux de restants: 84.01%
Taux de sortants: 15.99%
```

Ne prédire aucun départ revient à ne pas utiliser de solution d'analyse prédictive. Notre objectif pourrait donc être de battre les scores que ce genre de pratiques implique. En considérant que l'ensemble des employés restent dans l'organisation, nous obtiendrons un score d'Accuracy pour le jeu de test de 84%. Essayons de voir si nous arrivons à faire mieux.

3.4.2 Régression Logistique

Nous allons dans un premier temps implémenter un modèle de régression logistique. Pour ce faire nous utilisons la classe LogisticRegression de Sklearn :

```
# create the model
logistic_model = LogisticRegression(solver='liblinear')
```

Figure 45 - Création du modèle de régression logistique

Selon la documentation Sklearn, le solver liblinear est préféré pour les petits jeux de données. Nous entraînons ensuite le modèle sur le jeu d'entraînement avant de le tester :

```
# train the model (we use standardized data)
logistic_model.fit(X_train_std, y_train)

# predict
y_train_pred = logistic_model.predict(X_train_std)
y_test_pred = logistic_model.predict(X_test_std)
```

Figure 46 - Entrainement et prédiction (Régression logistique)

Nous le testons à la fois sur les données d'entraînement afin de vérifier que le modèle n'a pas surappris puis sur les données test pour s'assurer qu'il soit capable de généraliser.

```
=====
Régression logistic :
=====

Nombre d'observations du jeu d'entraînement : 1176
dont 190 collaborateurs qui sont partis.

Nombre d'observations du jeu de test: 294
dont 47 collaborateurs qui sont partis.
```

Matrice de confusion :

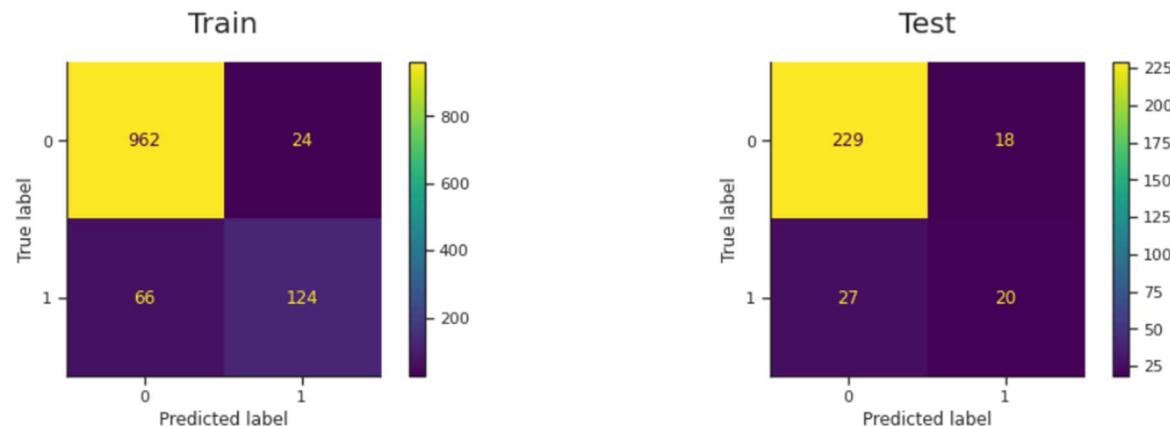


Figure 47 - Matrice de confusion (Régression logistique)

Métriques d'évaluation :

	model	jeu de données	precision	recall	accuracy	f1	AUC	ROC
0	LogisticRegression	test	0.53	0.43	0.85	0.47	0.68	
1	LogisticRegression	train	0.84	0.65	0.92	0.73	0.81	

Figure 48 - Métriques d'évaluation (Régression logistique)

Le modèle ne semble pas avoir sûr-appris. On constate sur le jeu de test que le modèle identifie correctement 20 des 47 collaborateurs qui sont partis soit un Recall de 0,43. On identifie donc correctement 43% des collaborateurs qui compte quitter l'organisation. Toutefois on identifie à tort 18 collaborateurs ce qui nous donne une précision de 53%. De manière générale, on identifie correctement 85% des observations. Toutefois rappelons-nous que l'Accuracy n'est pas un bon indicateur en présence de données déséquilibrées et préférons donc le score f1.

Ainsi sur 10 collaborateurs qui souhaitent réellement quitter l'organisation, ce modèle va permettre d'en identifier 4 parmi ceux-là et 5 autres ne désirant pas réellement la quitter. Le responsable des ressources humaines peut alors être averti au sujet de ces 9 collaborateurs et les rencontrer pour tenter de déceler leur volonté réelle de démissionner. Si besoin, il pourra alors entamer des démarches pour tenter de les faire rester. Il ne s'agit pas de proposer une augmentation aux neufs collaborateurs mais de proposer leur liste aux responsables des ressources humaines afin d'investiguer précisément sur leurs cas.

3.4.3 Random Forest

Essayons maintenant d'implémenter un algorithme de forêt aléatoire sur notre jeu de données. Rappelons-le, il s'agit d'exécuter un grand nombre d'arbres simultanément puis de donner la réponse qui revient le plus. Nous utilisons la classe RandomForestClassifier de Sklearn :

```
# create the model
random_forest_model = RandomForestClassifier(n_estimators=1000, random_state = 42, bootstrap=False)

# train the model
random_forest_model.fit(X_train, y_train)

# predict
y_pred = random_forest_model.predict(X_test)
```

Figure 49 - Création et entraînement du modèle (Random forest)

Ici, nous utilisons une forêt de 1000 arbres avec l'ensemble du jeu de données utilisé pour construire chaque arbre. Intéressons-nous à la matrice de confusion :

```
=====
Random Forest :
=====

Nombre d'observations du jeu d'entraînement : 1176
dont 190 collaborateurs qui sont partis.

Nombre d'observations du jeu de test: 294
dont 47 collaborateurs qui sont partis.
```

Matrice de confusion :

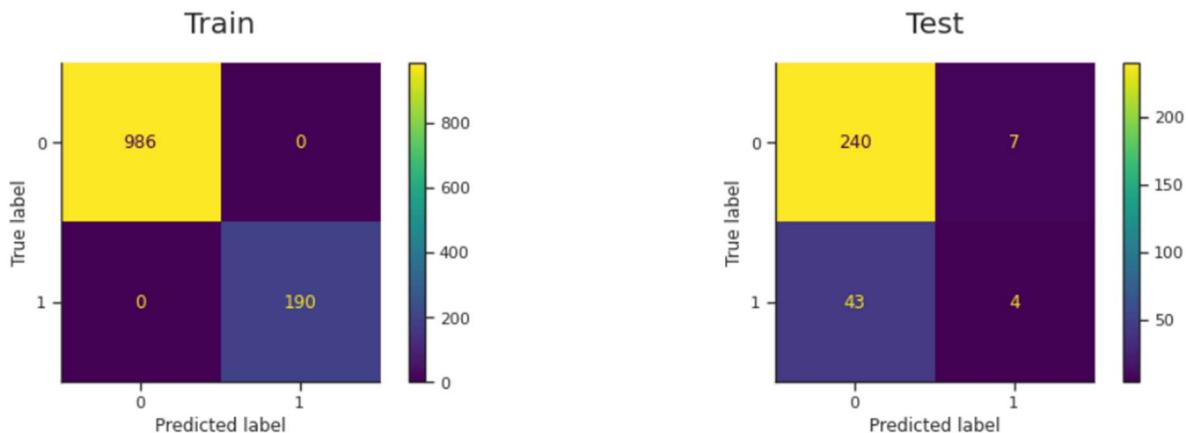


Figure 50 - Matrice de confusion (Random forest)

	model	jeu de données	precision	recall	accuracy	f1	AUC	ROC
0	RandomForestClassifier	test	0.36	0.09	0.83	0.14	0.53	
1	RandomForestClassifier	train	1.00	1.00	1.00	1.00	1.00	

Figure 51 - Métriques d'évaluation (Random forest)

Sur le jeu d'entraînement, l'Accuracy du modèle est surprenant, chaque observation est classée à la perfection. Le modèle ne commet aucune erreur. Cette observation est une première alerte. Si on s'intéresse au jeu de test en revanche, on constate que le modèle n'est pas très performant. Il s'agit de toute évidence d'un phénomène de sur-apprentissage. Notre modèle obtient de très bons résultats sur l'ensemble d'apprentissage, mais n'est pas en mesure de généraliser à de nouvelles données. Afin de remédier à ce problème, on peut optimiser le choix des paramètres de l'algorithme afin d'en améliorer les performances.

Dans le cas d'une forêt aléatoire, les hyperparamètres comprennent le nombre d'arbres de décision dans la forêt et le nombre de caractéristiques prises en compte par chaque arbre lors de la division d'un nœud. Scikit-Learn met en œuvre un ensemble d'hyper paramètres par défaut judicieux pour tous les modèles, mais il n'est pas garanti qu'ils soient optimaux pour un problème. Jetons un œil aux hyperparamètres de notre problème :

```
# Look at parameters used by our current forest
print('Parameters currently in use:\n')
print(random_forest_model.get_params())

Parameters currently in use:

{'bootstrap': False, 'ccp_alpha': 0.0, 'class_weight': None, 'criterion': 'gini', 'max_depth': None, 'max_features': 'auto',
```

Figure 52 - Paramètres du modèle Random forest

```
'max_leaf_nodes': None, 'max_samples': None, 'min_impurity_decrease': 0.0, 'min_samples_leaf': 1, 'min_samples_split': 2,
'min_weight_fraction_leaf': 0.0, 'n_estimators': 1000, 'n_jobs': None, 'oob_score': False, 'random_state': 42, 'verbose': 0, 'warm_start': False}
```

- **Bootstrap = False** : l'ensemble du jeu de données utilisé pour construire chaque arbre
- **Ccp_alpha = 0.0** : aucun élagage n'est effectué sur les arbres
- **Class_weight = None** : Toutes les classes sont supposés avoir le même poids
- **Criterion = Gini** : Utilise le critère de Gini pour mesurer la qualité du choix d'un nœud
- **Max_depth = None** : Les nœuds sont développés jusqu'à ce que toutes les feuilles soient pures ou jusqu'à ce que toutes les feuilles contiennent moins de min_samples_split échantillons.
- **Max_features = auto** : Le nombre de caractéristiques à prendre en compte lors de la recherche du meilleur split est $\sqrt{n_features}$
- **Max_leaf_nodes = None** : le nombre de nœuds feuilles est illimité
- **Max_samples = None** : le nombre d'échantillons à tirer de X pour entraîner chaque estimateur de base (si Bootstrap est vrai)
- **Min_impurity_decrease = 0.0** : Un nœud sera divisé si cette division induit une diminution de l'impureté supérieure ou égale à 0.0
- **Min_sample_leaf = 1** : Le nombre minimum d'observations requis pour être à un nœud feuille
- **Min_sample_split = 2** : Le nombre minimum d'observations requis pour diviser un nœud interne
- **Min_weight_fraction_leaf = 0.0** : Les échantillons ont un poids égal pour être nœuds feuille

- N_estimators = 1000 : Le nombre d'arbres dans la forêt
- N_jobs = None :
- Oob_score = False :
- Random_state = 42 :
- Verbose = 0 :
- Warm_start = False :

Cette longue liste d'hyper paramètres ne semble pas facile à prendre en main. Il revient cependant au data scientiste de choisir précisément chacune de leur valeur. L'idée est d'effectuer de nombreuses itérations de l'ensemble d'un processus de cross-validation, en utilisant à chaque fois différents paramètres de modèle. Chaque fois que nous voulons évaluer un ensemble différent d'hyper paramètres, nous devons diviser nos données d'entraînement en K ensemble et effectuer K entraînements et évaluations. Si nous disposons par exemple de 10 ensembles d'hyper paramètres et que nous utilisons une cross-validation à 5 ensembles, cela représente 50 boucles d'entraînement. Cette étape peut donc s'avérer fastidieuse compte tenu du nombre d'hyper paramètre et de leur domaine. Heureusement pour nous, Sklearn nous propose la classe GridSearchCV permettant de trouver de manière automatique le meilleur jeu de paramètre pour notre problème. Il suffit de lui donner le domaine de chacun d'entre eux :

```
# grille de paramètres
param_grid = dict(
    n_estimators= [100, 500, 900],
    max_features= ['auto', 'sqrt'],
    max_depth= [2, 3, 5, 10, 15, None],
    min_samples_split= [2, 5, 10],
    min_samples_leaf= [1, 2, 4],
    bootstrap= [True, False]
)
```

Figure 53 - Grille de nouveaux paramètres

Nous ne définissons pas le domaine de chacun des hyperparamètres vus plus haut mais uniquement ceux qui, selon la documentation Sklearn, sont les plus importants. L'objectif est de trouver la combinaison qui nous donne les meilleures performances. Dans cet exemple, on ne spécifie pas beaucoup de valeurs possibles afin de réduire le temps de calcul. Plus on agrandit les domaines de définition, plus le temps de calcul est long. Cette grille de paramètre est par exemple testée par la GridSearchCV en 45 minutes. Lors de mes tests, j'ai travaillé une grille plus complexe qui ne m'a pas donné de résultats au bout de 6h. En fonction de la puissance de calcul disponible, on peut chercher à être plus ou moins précis.

```
# find the best hyper parameters
best_random_forest_model = RandomForestClassifier(random_state=42)
search = GridSearchCV(best_random_forest_model, param_grid=param_grid, scoring='roc_auc', cv=5, verbose=1, n_jobs=-1)
search.fit(X_train, y_train)
```

Figure 54 - Recherche de la combinaison de paramètre la plus performante dans la grille

Nombre d'observations du jeu d'entraînement : 1176
dont 190 collaborateurs qui sont partis.

Nombre d'observations du jeu de test: 294
dont 47 collaborateurs qui sont partis.

Matrice de confusion :

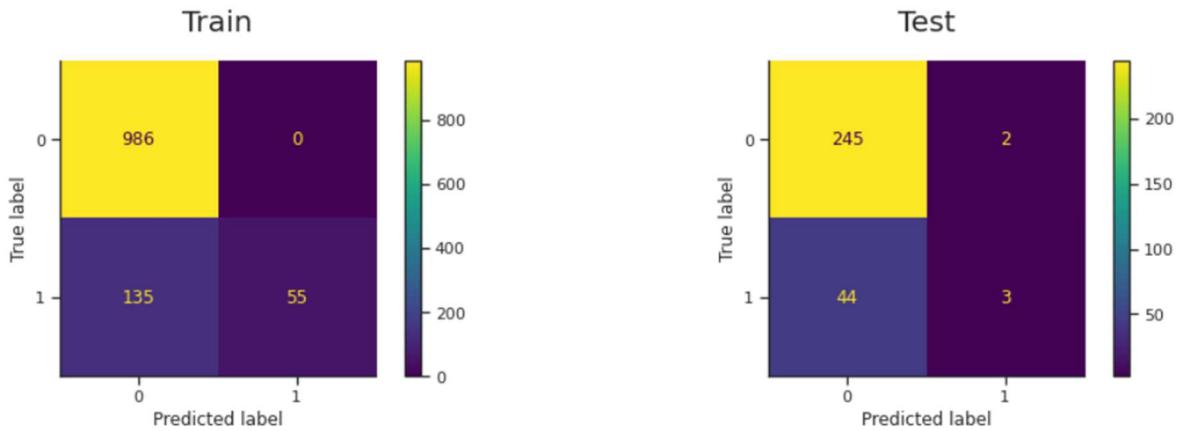


Figure 55 - Matrice de confusion (Random forest amélioré)

	model	jeu de données	precision	recall	accuracy	f1	AUC	ROC
0	RandomForestClassifier	test	0.6	0.06	0.84	0.12	0.53	
1	RandomForestClassifier	train	1.0	0.29	0.89	0.45	0.64	

Figure 56 - Métriques d'évaluation (Random forest amélioré)

Finalement notre modèle de RandomForest amélioré ne semble plus faire de sur-apprentissage. Le modèle présenté ici obtient une bonne précision mais un Recall très faible ce qui conduit à un score f1 faible également. Lorsqu'il identifie un individu, il le fait bien, en revanche il ne les détermine pas tous. Ce genre de performance peut être utile dans le cas où l'on désire sélectionner un petit nombre de candidats de manière certaine (ex : le recrutement).

Sklearn propose un attribut `feature_importances_` pour les algorithmes de forêts aléatoires permettant de connaître l'importance des variables au sein des arbres.

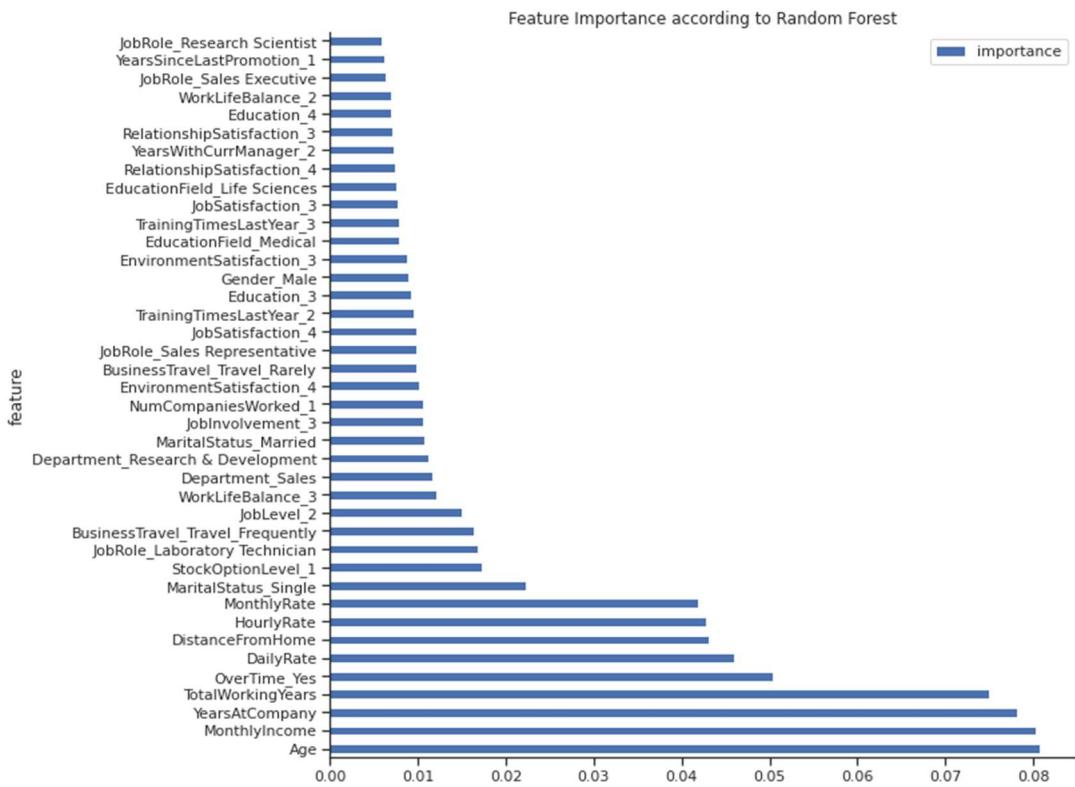


Figure 57 - Importance des variables selon Random forest

Selon notre forêt, les variables les plus importantes dans l'explication de l'attrition sont l'âge du collaborateur, son revenu mensuel et son ancienneté dans l'entreprise.

3.4.4 XGBoost

Essayons ensemble un dernier algorithme relativement célèbre dans les concours de Machine Learning : XGBoost.

```
# create the model
xgb_modele = XGBClassifier(learning_rate=0.01, n_estimators=2000, use_label_encoder=False, random_state=420)

# train the model
xgb_modele.fit(X_train, y_train)

# predict
y_train_pred = logistic_model.predict(X_train)
y_test_pred = logistic_model.predict(X_test)
```

Figure 58 - Création, entraînement et prédictions du modèle (XGBoost)

Une fois le modèle entraîné et testé, intéressons-nous à ces performances :

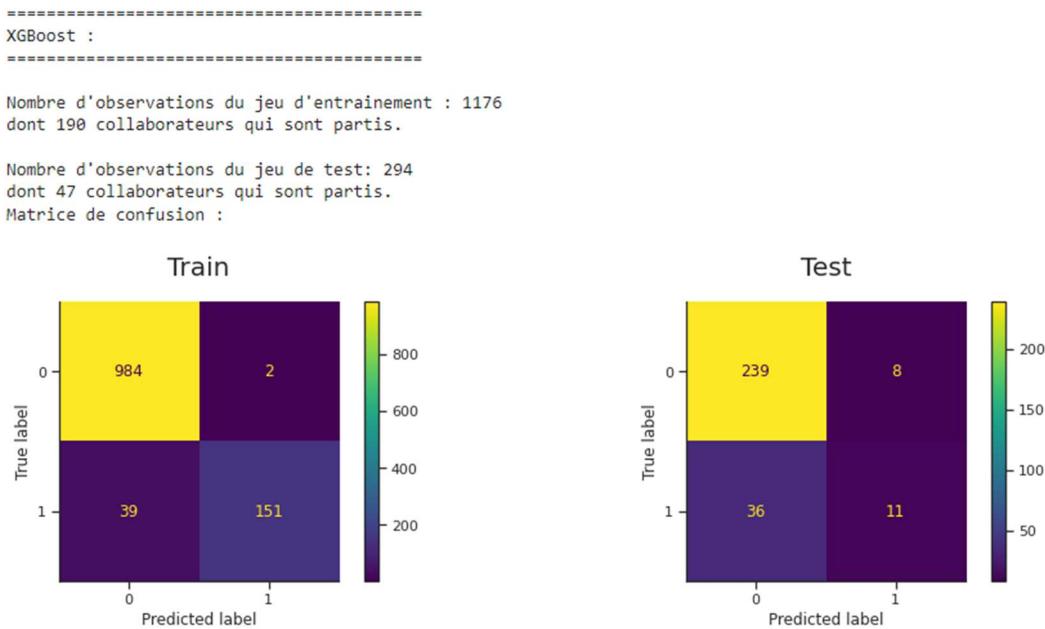


Figure 59 - Matrice de confusion et métriques d'évaluations (XGBoost)

Avec une Accuracy de 0.97, XGBoost est bon sur le jeu d'entraînement sans pour autant perdre ses moyens sur le jeu de test. Les résultats sont satisfaisants avec notamment une bonne précision. Toutefois, comme avec la forêt aléatoire, il ne dispose pas d'un excellent Recall.

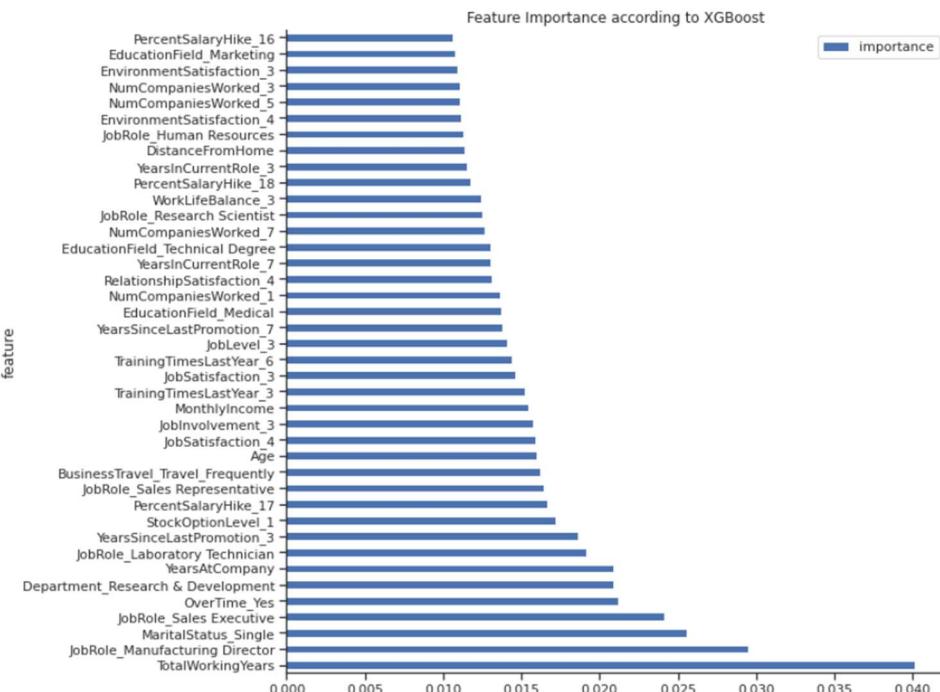


Figure 60 - Importance des variables (XGBoost)

Pour XGBosst, l'expérience du collaborateur, le rôle de « Director Manufacturing » et le fait d'être en situation maritale « seul » sont les caractéristiques les plus importantes pour expliquer l'attrition.

3.5 Mise en production

A la vue de ces 3 algorithmes, nous sommes en mesure de choisir l'un d'entre eux, en fonction de nos exigences, afin de l'industrialiser. L'objectif est d'utiliser le modèle choisi au quotidien dans la vie de l'entreprise. Nous allons dans cette section étudier ensemble quel algorithme peut être le plus intéressant et quels en seraient les bénéfices.

3.5.1 Quel modèle choisir ?

Nous arrivons à un point clef de notre projet de Machine Learning. Il est temps de décider, aux regards de nos exigences quel serait le modèle le plus à même de convenir à notre organisation. Si on s'intéresse à l'Accuracy, chacun des 3 algorithmes semble être un bon candidat.

	model	jeu de données	precision	recall	accuracy	f1	AUC	ROC
0	LogisticRegression	test	0.53	0.43	0.85	0.47	0.68	
0	RandomForestClassifier	test	0.60	0.06	0.84	0.12	0.53	
0	XGBClassifier	test	0.58	0.23	0.85	0.33	0.60	

Figure 61 - Comparaison des différents modèles

Chacun des algorithmes obtient un score légèrement supérieur ou égal à notre Baseline ce qui nous conforte dans notre idée de pouvoir proposer une solution avec une réelle plus-value. Si on considère l'aire sous la courbe ROC, la régression Logistique semble prendre les devants. Pourtant, pouvons-nous uniquement nous baser sur ces métriques générales ? Bien qu'elles reflètent les performances globales de nos modèles, nous avons pu constater leurs limites au sein de l'état de l'art. Rappelons-nous ce que désire réellement l'organisation : détecter les employés désirant quitter l'organisation afin de les retenir. Grâce à l'analyse descriptive, nous avons constaté qu'uniquement 16% des employés se trouvaient dans cette position. Nous sommes donc dans un contexte de données déséquilibrées nous demandant des analyses plus précises que celles de l'Accuracy ou de l'AUC ROC. Nous nous intéresserons donc plutôt à la précision et au Recall qui sont résumés dans le score f1, particulièrement utilisés pour les problèmes de classification utilisant des données déséquilibrées. Deux stratégies s'offrent alors à nous :

La stratégie Précision consiste à accepter de sacrifier notre indice de Recall au profil de la précision. On préfère détecter moins d'individus mais être certain de leur appartenance à la classe cible. Dans le cadre de notre problème, cette stratégie peut être adoptée lorsque l'on dispose de peu de cartes à jouer pour retenir nos collaborateurs. Un manager peut par exemple n'avoir à sa disposition qu'un budget limité pour distribuer des augmentations et préférer se concentrer sur les collaborateurs qui désirent le quitter. Ainsi avec une stratégie Précision il identifiera peut-être moins de collaborateurs mais son budget sera alloué de manière plus efficace. Au sein de nos algorithmes, la solution XGBoost semble la plus en adéquation avec cette stratégie. En effet, la forêt aléatoire dispose ici d'un Recall tellement bas que sa bonne précision ne la rend ici pas compétitive.

La stratégie Recall au contraire consiste à accepter de prédire avec moins de précision, c'est-à-dire que parmi les individus identifiés par le modèle, une plus grande part d'entre eux n'aurait probablement pas réellement quitté l'organisation. En revanche, on cherche ici à identifier le maximum d'individus du groupe

de démissionnaires. Dans un contexte de rétention important des collaborateurs, il peut être intéressant de choisir cette stratégie. Ainsi, on optera ici pour la solution se basant sur la régression logistique.

Bien que plusieurs approches soient possibles, gardons à l'esprit que les prédictions réalisées par nos algorithmes seront ensuite complétées par une analyse terrain effectuée par les RH. Si le modèle identifie 10 individus à risques, il appartient alors aux ressources humaines d'investiguer auprès de ses collaborateurs afin d'identifier clairement s'ils désirent partir ou non. Au regard de ces résultats, il semblerait toutefois que la régression Logistique soit le bon choix. Grâce à une qualité de données irréprochable, ce modèle a su s'imposer face aux arbres de décisions qui eux, montrent leurs forces surtout en présence de bruit.

3.5.2 Quel sont les gains ?

Au bout d'un processus complexe, nous avons finalement sélectionné notre algorithme. Mais quels sont les gains réels pour l'entreprise ? Tentons de chiffrer les économies que pourrait effectuer notre organisation.

Nous allons ici réaliser une mise en situation pratique et considérer le jeu de données testé précédemment comme une représentation réaliste d'une organisation. Ainsi considérons que l'entreprise IBM dispose de 1470 employés dont 16,1% sont partis et ont été remplacés au cours de l'année. Cela signifie donc que l'entreprise a dû procéder au renouvellement de :

$$1470 * 0,161 = 237 \text{ collaborateurs}$$

En moyenne, le salaire mensuel de ces 237 employés s'élève à \$4787, ce qui équivaudrait à un salaire annuel de \$ 57 444. Nous avons vu en introduction, que le coût du renouvellement d'un employé peut valoir à l'entreprise entre 6 et 9 mois de salaire. On peut alors chiffrer le coût du renouvellement des 237 collaborateurs à :

$$237 * \left(\frac{6}{12} * 57\,444 \right) = \$ 6\,807\,114$$

Forte de ce constat, l'entreprise IBM a décidé de mettre en place un projet de Machine Learning afin de l'aider à détecter les collaborateurs souhaitant quitter l'organisation. Cette dernière a décidé d'utiliser un modèle de régression Logistique lui assurant un bon score de Recall :

Métriques d'évaluation :

	model	jeu de données	precision	recall	accuracy	f1	AUC	ROC
0	LogisticRegression	test	0.53	0.43	0.85	0.47	0.68	

Figure 62 - Performances du modèle choisi

Au cours de l'année, l'entreprise a testé régulièrement ses effectifs afin de déterminer les profils à risque. Au terme de l'exercice, l'entreprise cherche à calculer le retour sur investissement de son projet en quantifiant notamment les économies que la solution lui a permis de réaliser. Ne disposant pas de plusieurs exercices au sein des données, on considère ici que les tendances sont les mêmes d'une année sur l'autre. Ainsi, la solution de Machine Learning a permis à l'organisation de détecter 192 profils qui auraient potentiellement pu démissionner :

$$\text{Grâce aux données de test, on sait que recall} = \frac{TP}{TP + FN} = 0,43$$

On sait également que $TP + FN = 237$, ce qui nous permet d'écrire que $\frac{TP}{237} = 0,43$

Ainsi $TP = 0,43 * 237 = 102$

On peut estimer que parmi les 192 profils identifiés par le modèle, 102 comptaient réellement quitter l'organisation.

D'autre part, comme $precision = \frac{TP}{TP + FP} = 0,53$ et $TP = 102$

$On\ a\ FP = \frac{1}{0,53} * 102 - 102 = 90$

Ainsi, le modèle a donc désigné 90 profils qui ne comptaient vraisemblablement pas quitter l'organisation. Au moment des prédictions, l'organisation ne sait pas qui souhaite réellement démissionner. Ainsi les 192 employés ont finalement été rencontrés par les collaborateurs des ressources humaines avant leur départ afin de chercher à identifier les causes du mécontentement et chercher à les résoudre.

Parmi ces 192 alertes, les collaborateurs des ressources humaines ont réussi à identifier 70% des FP soit environ 63 personnes. Parmi les 102 TP, on estime que 20% d'entre eux quittent l'organisation quoi qu'il arrive et que pour diverses raisons l'entreprise ne peut rien faire pour les retenir ou ne désire simplement pas le faire. Les 80% restant, soit 82 personnes ont pu être retenues à temps grâce à des dispositions prises par l'entreprise.

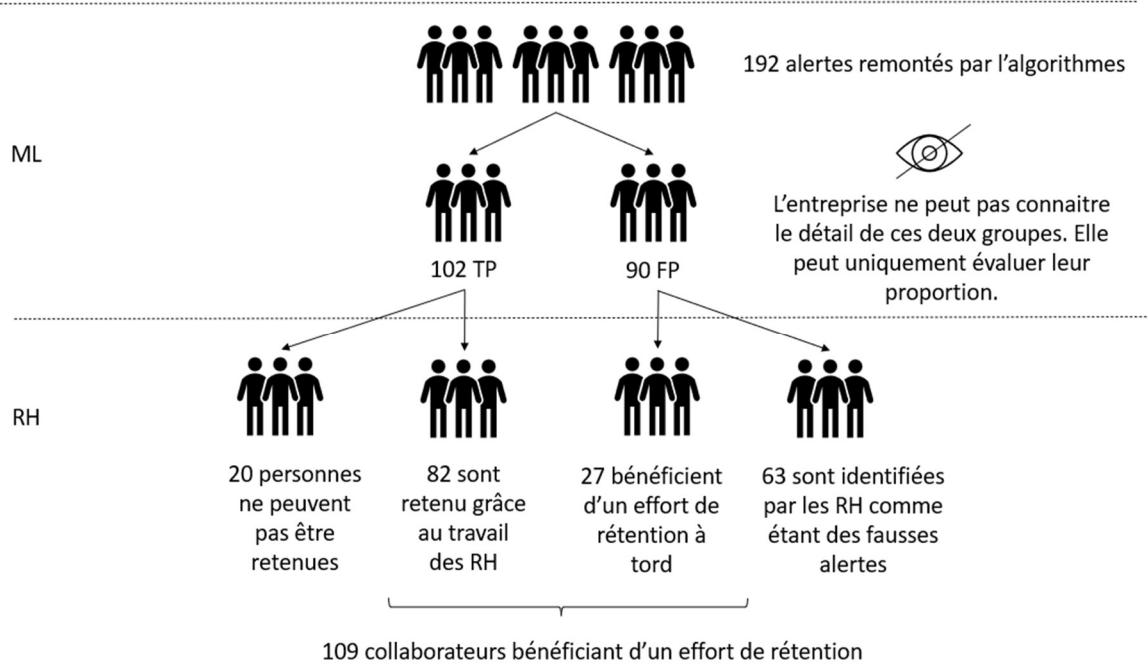


Figure 63 - Processus de sélection des employés bénéficiant de l'effort de rétention

Nous avons finalement pu, grâce à l'analyse prédictive faire économiser à l'entreprise :

$$82 * \left(\frac{6}{12} * 57\,444 \right) = \$ 2\,355\,204$$

Toutefois, afin de retenir les collaborateurs, les ressources humaines ont dû accorder au 82 TP + 27 FP = 109 salariés suspectés de départ, divers avantages comme le télétravail, un changement de management, des horaires flexibles ou des changements de statuts. Ces « concessions » ne coûtent rien à l'entreprise et résultent simplement d'un processus d'écoute du collaborateur. Pour certains employés, l'entreprise a cependant dû proposer au salarié une prime, des formations ou une revalorisation de salaire. On peut ici faire l'hypothèse que cela concerne la moitié des employés et qu'une augmentation de salaire de 10% a permis de les retenir. Finalement, l'entreprise aura déboursé :

$$0,1 * 57\,444 * \frac{109}{2} = \$ 313\,069$$

L'augmentation des avantages et des salaires revient finalement moins chère à l'entreprise qu'une démission. Au regard de l'ensemble de ces éléments, on peut estimer que la mise en place du projet d'analyse prédictive a finalement permis à l'organisation d'économiser :

$$2\,355\,204 - 313\,069 = \$ 2\,042\,135$$

Si on s'intéresse à l'évolution du taux de turn-over et que l'on considère que les sorties sont remplacées par de nouveaux collaborateurs, on peut constater que sa valeur a diminué :

Année N (avant la mise en place du projet)	$\text{taux de turn - over} = \frac{(237 + 237) / 2}{1470} * 100 = 16,1\%$
Année N+1 (après la mise en place du projet)	<p>Grâce à l'analyse prédictive, nous avons retenu 82 employés sur les 237 qui comptaient partir.</p> $237 - 82 = 155$ $\text{taux de turn - over} = \frac{(155 + 155) / 2}{1470} * 100 = 10,5\%$

Grâce à l'analyse prédictive et à la mise en place d'une solution de Machine Learning, nous avons pu mettre en place des actions ciblées afin de réaliser de la rétention d'employé et donc de réduire le taux de turn-over.

Finalement les retours de l'implémentation d'une telle solution ne sont pas négligeables. Il s'agit toutefois de chiffrer sa mise en place afin d'évaluer la réelle plus-value du projet. En fonction de la disponibilité de ressources internes qualifiées, de la charge de travail à consacrer au recueil et à la qualification des données, on peut ici difficilement estimer le coût du projet. En revanche, une organisation souhaitant réaliser ce genre de travaux à tout intérêt à le faire.

3.6 Pour aller plus loin

La solution présentée ici est loin d'être parfaite mais permet de tracer la voie pour des analyses plus poussées. On pourrait, par exemple, chercher à retirer les valeurs aberrantes dans les données comme

celles vues lors de l'analyse de la variable « MonthlyIncome » ou « YearsWithCurrManager ». On pourrait également chercher, grâce à une puissance de calcul plus grande, à trouver les paramètres optimaux pour chacun des algorithmes ou encore à développer d'autres modèles.

3.6.1 Utilisation et alternatives

Pour l'utilisation courante de la solution, on pourrait imaginer qu'à chaque mouvement dans l'entreprise, chaque changement de variable pour un employé, l'algorithme teste l'individu et retourne un résultat. Ainsi en fonction de la vie de l'individu au sein de l'organisation, les employés des ressources humaines pourraient tracer ses probabilités de départ. La décision de démissionner n'étant pas immédiate après l'entrée dans une situation, l'organisation aurait le temps de préparer une réponse à proposer au collaborateur.

On pourrait par ailleurs imaginer réentraîner l'algorithme périodiquement avec un historique de données plus frais. Ainsi le modèle pourrait prendre en compte les évolutions sociales et les envies des collaborateurs. En effet, ce qui constituait un motif de démission hier ne l'est plus forcément aujourd'hui. Il serait donc intéressant de réentraîner périodiquement l'algorithme pour suivre l'évolution de nos sociétés.

Enfin, on pourrait imaginer implémenter un nouvel algorithme. Cette fois, on pourrait chercher, non plus à connaître la classe d'un employé, mais plutôt sa durée de vie dans l'entreprise. Il s'agirait de calculer en fonction des mêmes variables la durée pendant laquelle le collaborateur souhaite rester dans l'organisation. On pourrait par exemple prévoir qu'un individu souhaitera quitter l'organisation dans 3 mois, 1 an, 6 ans. Il s'agirait donc plus d'un modèle de classification mais bien de régression. Ce type de modèle nécessiterait tout de même un peu plus de données temporelles que celles disponibles aujourd'hui dans notre jeu de données.

3.6.2 Proposition d'une méthode pour l'organisation

Afin de réaliser un projet de création d'analyse prédictive pour le suivi et la prédiction de turn-over, on peut au regard de ce mémoire proposer une méthode, qu'il convient d'adapter en fonction des contextes :

1^{ère} étape: Chiffrer la perte d'un collaborateur et identifier les ressources critiques afin de mieux contextualiser l'environnement. On sait alors combien nous coûte le départ d'un employé, ce qui aide à envisager les différents leviers possibles pour le faire rester. On sait alors combien on est prêt à mettre et pourquoi. Grâce à l'identification des ressources critiques, on connaît dans chaque équipe les éléments qu'il ne faut surtout pas perdre de vue et laisser partir. Ces deux éléments sont applicables même hors d'un contexte de mise en place de projet d'analyse prédictive.

2^{ème} étape : Mettre en place une collecte des données efficace. Collecter de la donnée c'est bien, la collecter efficacement c'est mieux. Si la collecte n'est pas rigoureuse, les analystes vont parfois devoir passer un temps fou à tenter de les corriger. Pire, ils pourraient être tentés de ne pas les utiliser, rendant leur collecte inutile. Il est donc essentiel que l'organisation dispose d'une méthode de collecte claire et précise de la donnée. On peut imaginer qu'une donnée incomplète soit refusée et retournée à l'opérateur pour correction tant que celle-ci n'est pas complète. Ce processus n'est pas à prendre à la légère, les gains potentiels pour l'entreprise sont énormes. En parallèle, l'organisation peut, au regard de ce qu'il est possible de faire, se poser la question de ce qu'elle veut collecter. Doit-elle par exemple enregistrer le fait qu'un collaborateur a droit au télétravail ? Souvent négligée, elle est la base et le fondement des analyses qui suivront.

3^{ème} étape : Mettre en place un projet d'analyse descriptive. Cet élément est généralement en place dans la plupart des organisations. Fortement dépendante de la qualité des données qu'elle utilise, l'analyse descriptive permet de poser le contexte de l'organisation. Elle révèle les tendances et les relations entre les différentes variables de l'organisation. Elle accompagne les décideurs et permet d'avoir un regard complet sur l'historique de l'organisation.

4^{ème} étape : Réaliser un projet d'analyse prédictive. Dans la lignée de l'exemple que nous avons traité ensemble, l'organisation est alors assez mature pour réaliser un projet de ML. Elle dispose de données de qualités et des informations pertinentes lui permettant d'attendre une réelle plus-value du projet.

Conclusion générale

Pour conclure, nous avons vu qu'un taux de turn-over élevé coûte cher à l'entreprise. Plutôt que de chercher à le réduire à tout prix, nous avons compris qu'il était plus intéressant pour l'organisation de se concentrer principalement sur les départs involontaires de ses ressources critiques. Avoir du turn-over n'est pas forcément mauvais en soi, en revanche, perdre ses éléments clés est un problème. Il devient donc primordial pour les organisations d'arriver à identifier ces employés. L'analyse descriptive, aujourd'hui pratiquée dans la plupart des SIRH, ne permet pas d'anticiper efficacement le départ d'un collaborateur. Certaines métriques comme le taux turn-over ou le taux de rétention permettent d'avoir un aperçu de la stratégie mise en place mais ne permettent pas de se projeter. Ils sont révélateurs du fonctionnement de l'organisation mais sont utilisés comme alertes plutôt que comme guides. Face à ce constat, l'analytique RH et notamment le Machine Learning propose des analyses prédictives sur les données collaborateurs permettant d'anticiper leur départ.

L'enjeux de ces solutions réside dans un premier lieu dans la disponibilité de données qualifiées et de confiance. L'organisation doit être mature dans son utilisation de la donnée et prendre conscience des gains qu'elle pourrait en tirer. Plus cette dernière collectera de façon précise et juste, plus les retombées pour l'organisation seront grandes. En effet, nous avons vu comment la mise en place d'un projet de Machine Learning appliqué au domaine des ressources humaines peut faire économiser à l'entreprise, tout en faisant diminuer son taux de turn-over. Ce genre de performance peut alors donner confiance en l'entreprise et envoyer des signaux positifs aux collaborateurs.

Bien que la mise en place de telles solutions soit encore réservée à des professionnels, on assiste à une libéralisation des solutions rendant accessible la réalisation d'analyse au plus grand nombre. Attention toutefois car l'essor des solutions AutoML pourrait engendrer toute sorte d'abus. N'importe qui pourrait alors prédire n'importe quoi et le sens critique de l'opérateur pourrait être effacé au profit de la machine savante.

Bibliographie

- [1] <https://www.capital.fr/votre-carriere/les-cadres-prets-a-changer-de-job-mais-sous-conditions-1044792>
- [2] [Turn Over : combien coûte la perte d'un employé ? | MOMEN \(cabinet-management-transition.com\)](#)
- [3] <https://recruteur.lefigaro.fr/article/quel-est-le-cout-dun-recrutement-rate/>
- [4] <https://www.franceinter.fr/emissions/histoires-economiques/histoires-economiques-du-lundi-21-fevrier-2022>
- [5] [taux de roulement: la vérité, toute la vérité, rien que la vérité! | Mesurer le capital humain \(wordpress.com\)](#)
- [6] [Taux Turnover : Comment le calculer et l'analyser ? \(smallbusinessact.com\)](#)
- [7] [Calculer et analyser son taux de turnover - Le Blog GERESO](#)
- [8] [Turnover : définition, causes et conséquences | Qualtrics](#)
- [9] [Turnover : définition, calcul et conseils pour le réduire \(lefigaro.fr\)](#)
- [10] [Causes et conséquences du Turnover \(wuro.fr\)](#)
- [11] [KPI RH et indicateurs RH | +100 exemples pour votre compagnie \(bizneo.com\)](#)
- [12] [Les différents types de turnover du personnel - Yumani](#)
- [13] [Recrutement : quels sont les KPI's RH à suivre ? - La Boite à Outils des RH \(laboiteaoutilsdesrh.fr\)](#)
- [14] [Qu'est-ce qu'un SIRH et à quoi sert-il ? | CIEFA Paris](#)
- [15] [Prix Logiciel RH, Recrutement, Paie et Ressources Humaines | CELGE](#)
- [16] [Comment prédire le départ de vos collaborateurs \(et 15 signes avant-coureurs pour l'anticiper\) \(sparkbay.com\)](#)
- [17] [C'est quoi l'analytique RH ? - Bing video](#)
- [18] [Comment prédire le départ de ses collaborateurs ? \(helloworldplace.fr\)](#)
- [19] [Let's Predict Who's Going to Quit | by Andy Chan | The Human Business | Medium](#)
- [20] [60 Notable Machine Learning Statistics: 2021/2022 Market Share & Data Analysis -inancesonline.com](#)
- [21] [Donnée structurée et non structurée : définition | Talend | Talend](#)
- [22] [Etapes Machine Learning : comprendre le processus | Talend | Talend](#)
- [23] [Qu'est-ce que le Machine Learning ? | Définition, types et exemples | SAP Insights](#)
- [24] [Machine learning : comment évaluer vos modèles ? Analyses et métriques! \(saagie.com\)](#)
- [25] [Prediction of Employee Turnover in Organizations using Machine Learning Algorithms \(semanticscholar.org\)](#)
- [26] [Understanding Logistic Regression - GeeksforGeeks](#)
- [27] [Matrice de confusion / Confusion matrix pour le Machine Learning \(kobia.fr\)](#)
- [28] [Balanced Accuracy Weighted, Accuracy and imbalanced data \(kobia.fr\)](#)
- [29] [F1-score & F-beta score, compromis entre Precision et Recall en classification \(kobia.fr\)](#)

- [30] [Évaluez un algorithme de classification qui retourne des scores - Évaluez les performances d'un modèle de machine learning - OpenClassrooms](#)
- [31] [Courbe de rappel de précision | ML – Acervo Lima](#)
- [32] <https://www.geeksforgeeks.org/naive-bayes-classifiers/>
- [33] <https://www.javatpoint.com/machine-learning-naive-bayes-classifier>
- [34] <https://www.javatpoint.com/machine-learning-decision-tree-classification-algorithm>
- [35] <https://www.geeksforgeeks.org/decision-tree-implementation-python/>
- [36] <https://www.javatpoint.com/machine-learning-random-forest-algorithm>
- [37] <https://www.javatpoint.com/k-nearest-neighbor-algorithm-for-machine-learning>
- [38] <https://www.geeksforgeeks.org/support-vector-machine-algorithm/>
- [39] [XGBoost — Programmation Informatique — DATA SCIENCE](#)
- [40] [What are Neural Networks? | IBM](#)
- [41] [Concepts — ML Glossary documentation \(ml-cheatsheet.readthedocs.io\)](#)
- [42] [Les meilleures notebooks Python pour l'machine learning – Acervo Lima](#)
- [43] [Machine Learning : 5 outils à utiliser si vous n'êtes pas Data Scientist \(lebigdata.fr\)](#)

Grant Thornton
 Applications & Projets
 29 Rue du Pont
 92200 Neuilly-sur-Seine

DAILLE Thomas
 Université Paris Dauphine
 M2 MIAGE ID en Apprentissage
 Année 2021-2022

RÉSUMÉ

Face au problème de la grande démission, les organisations tentent aujourd’hui de mettre en place des stratégies de rétention de leurs collaborateurs. En effet, un turn-over élevé peut être dangereux pour l’entreprise qui verra son image ternie, ses coûts explosés et sa productivité diminuée (engendre également l’usure des parties présentes, managers par la nécessité de recruter, former à nouveau, évaluer, etc.) Grâce à la mise en place de projets d’analyses descriptives, les organisations tentent d’expliquer les facteurs de démission qu’elles rencontrent. Loin d’être suffisante, cette analyse descriptive peut aujourd’hui être complétée par une analyse prédictive visant à comprendre les collaborateurs en les analysant par le biais des variables de suivi. Grâce à l’intelligence artificielle et notamment au Machine Learning, les analystes sont aujourd’hui capables d’estimer la probabilité de départ d’un collaborateur. Grâce à ces probabilités, l’organisation peut alors mettre en place des actions ciblées de rétention et ainsi tenter plus efficacement de conserver ses talents.

MOTS CLÉS

Informatique décisionnel - Ressources humaines – Turn-over – Machine Learning – Apprentissage supervisé - Classification – Analyse descriptive – Analyse prédictive – Régression logistique – Random Forest - XGBoost

ABSTRACT

Business Intelligence - Human Resources - Turnover - Machine Learning - Supervised Learning - Classification - Descriptive Analysis - Predictive Analysis - Logistic Regression - Random Forest - XGBoost

KEYWORDS

Faced with the problem of high turnover, organizations are now trying to implement strategies to retain their employees. Indeed, a high turnover can be dangerous for the organization which will see its image tarnished, its costs explode and its productivity decrease. Thanks to the implementation of descriptive analysis projects, they try to explain the resignation factors they encounter. Far from being sufficient, this descriptive analysis can now be completed by a predictive analysis aiming at understanding the employees by analyzing them thanks to follow-up variables. Thanks to artificial intelligence and especially Machine Learning, analysts are now able to estimate the probability of an employee's departure. Thanks to these probabilities, the organization can then set up targeted retention actions and thus try to retain its talents more efficiently.