le cnam
Nouvelle-Aquitaine

Rapport final – 26 mars 2024

Prédiction du départ d'un employé

Entreposage et fouilles de données

Github projet : https://github.com/ndesbrosse/Projet-final

Théo CHARRÉ
Léo JUBIEN
Djessy NGAKA
Julien FÉVRIER
Nathan DESBROSSE

Table des matières

Introduction

Dans un environnement professionnel en constante évolution, la rétention des employés constitue un enjeu majeur pour les entreprises. La perte de talents qualifiés peut entraîner des coûts significatifs, perturber la continuité opérationnelle et compromettre la compétitivité sur le marché. Pourtant, anticiper les départs et identifier les employés à risque de quitter l'entreprise demeure un défi complexe. L'émergence de stratégies et d'outils visant à prévoir les mouvements de personnel s'avère essentielle pour les départements des ressources humaines et les gestionnaires. Cette tâche ne se résume pas uniquement à reconnaître les signaux évidents de désengagement, elle implique une compréhension profonde des facteurs individuels et organisationnels qui influencent les décisions des employés. Dans cette optique, cette étude se penche sur les différentes approches et méthodes permettant d'identifier les employés susceptibles de quitter l'entreprise. Nous explorerons les indicateurs comportementaux, les modèles analytiques avancés et les technologies émergentes qui facilitent cette démarche. En outre, nous mettrons en lumière l'importance de la rétention proactive et des pratiques de gestion axées sur l'engagement pour favoriser la fidélisation des talents. Ainsi, cette recherche vise à offrir aux décideurs des perspectives éclairées et des solutions pragmatiques pour anticiper les départs au sein de leur organisation, afin de promouvoir un environnement de travail stable, productif et propice à l'épanouissement professionnel des employés.

Problématique

Comment les modèles analytiques avancés, tel que l'intelligence artificielle, peut-il être exploité pour prédire avec précision les départs des employés et aider les entreprises à prendre des mesures préventives ?

Présentation des données, contexte et position du problème

Dans ce projet, notre objectif est de réussir à identifier les employés qui seraient à risque de quitter l'entreprise. Pour ce faire, nous avons notre dataset qui est un fichier csv contenant des informations sur ces employés. Voici une description des données que nous avons :

id	type	na_value	unique_value	value_1	value_2	value_3
satisfaction	float64	0	92	0.38	0.8	0.11
derniere_evaluation	float64	0	65	0.53	0.86	0.88
nombre_de_projets	int64	0	6	2.0	5.0	7.0
nombre_heures_mensuelles_moyenne	int64	0	215	157.0	262.0	272.0
temps_passe_dans_entreprise	int64	0	8	3.0	6.0	4.0
accident_du travail	int64	0	2	0.0	0.0	0.0
depart	int64	0	2	1.0	1.0	1.0
promotion_5_dernieres_annees	int64	0	2	0.0	0.0	0.0
service	object	0	10	sales	sales	sales
niveau_salaire	object	0	3	low	medium	medium

Dans ces données nous avons donc :

- **Satisfaction** : Un nombre à virgules flottantes compris entre 0 et 1 qui représente le taux de satisfaction du salarié au sein de son entreprise
- **Dernière évaluation** : Un nombre à virgules flottantes compris entre 0 et 1 qui représente la dernière note attribuée à l'employé.
- **Nombre de projets** : Un entier qui représente le nombre de projets sur lequel le salarié a travaillé dans cette entreprise
- **Nombre heure mensuelles moyenne** : Un entier qui représente le nombre d'heure en moyenne travaillé par l'employé
- **Temps passé dans l'entreprise** : Un entier qui représente le nombre d'années travaillées par le salarié au sein de l'entreprise
- **Accident du travail** : Un entier qui représente si un salarié a déjà eu un accident du travail au sein de l'entreprise ou non. (0 si non, 1 si oui)
- **Départ** : Un entier qui représente si un salarié est parti de l'entreprise (0 si non, 1 si oui)
- **Promotion 5 dernières années** : Un entier qui représente si un salarié a obtenu une promotion dans les cinq dernières années au sein de l'entreprise (0 si non, 1 si oui)
- **Service** : Une valeur qui indique le service dans lequel l'employé travaille
- **Niveau salaire** : Une valeur qui indique le niveau de salaire de l'employé selon trois catégories : low, medium, high

Grâce à ce tableau nous pouvons remarquer qu'il n'y a pas de valeurs vides dans notre dataset, et que nous avons plusieurs valeurs booléennes, c'est-à-dire des valeurs ou le résultat est soit vrai soit faux, 0 ou 1, oui ou non. Ces valeurs sont Accident du travail, Départ et Promotion dans les 5 dernières années.

En parcourant les données un peu plus en détail, nous obtenons ceci :

	Satisfaction	derniere_evaluation	Nombre_de_projets	Nombre_heures_mensuelles_moyenne	Temps_passe_dans_entreprise	Accident_du travail	depart	promotion_5_dernieres_annees
count	14999.000000	14999.000000	14999.000000	14999.000000	14999.000000	14999.000000	14999.000000	14999.000000
mean	0.612834	0.716102	3.803054	201.050337	3.498233	0.144610	0.238083	0.021268
std	0.248631	0.171169	1.232592	49.943099	1.460136	0.351719	0.425924	0.144281
min	0.090000	0.360000	2.000000	96.000000	2.000000	0.000000	0.000000	0.000000
25%	0.440000	0.560000	3.000000	156.000000	3.000000	0.000000	0.000000	0.000000
50%	0.640000	0.720000	4.000000	200.000000	3.000000	0.000000	0.000000	0.000000
75%	0.820000	0.870000	5.000000	245.000000	4.000000	0.000000	0.000000	0.000000
max	1.000000	1.000000	7.000000	310.000000	10.000000	1.000000	1.000000	1.000000

Ce tableau permet d'avoir plus d'informations sur les données tel que la moyenne, les quartiles, la valeur minimale et maximale. Nous pouvons par exemple remarquer que nous avons un dataset qui comporte des informations sur **14999 employés**. La moyenne permet également d'avoir des tendances sur notre dataset tels que :

- **Satisfaction** : **0.61** Cette moyenne nous permet de remarquer que les employés sont dans l'ensemble plutôt satisfait dans l'entreprise.
- **Dernière évaluation** : **0.71** Cette moyenne nous permet de remarquer que les employés sont dans l'ensemble plutôt bien noté sur leur travail
- **Nombre de projets** : **3.80** Pas d'observation particulière

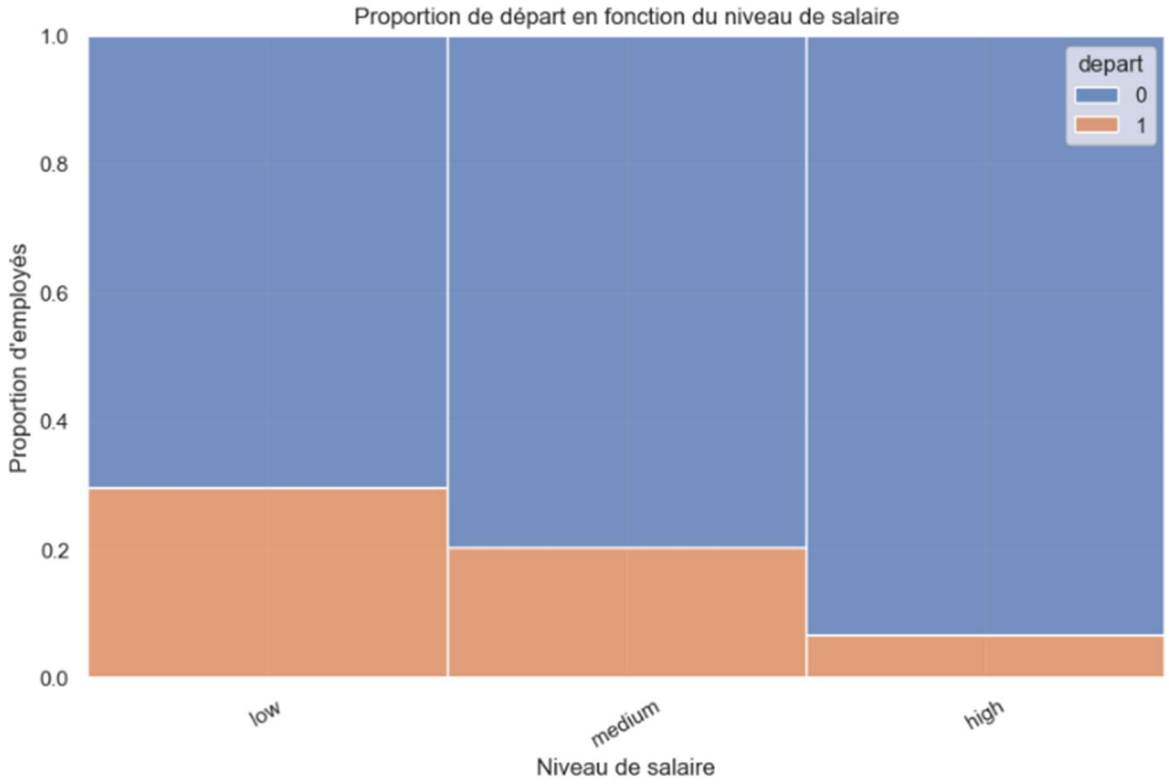
- **Nombre heure mensuelles moyenne** : **201** Pas d'observation particulière
- **Temps passé dans l'entreprise** : **3.49** Pas d'observation particulière
- **Accident du travail** : **0.14** Pas d'observation particulière
- **Départ** : **0.23** Cette moyenne nous permet de remarquer qu'une partie des employés, environ un quart ne sont plus dans l'entreprise.
- **Promotion 5 dernières années** : **0.02** Cette moyenne nous permet de remarquer que très peu d'employés ont obtenus une promotion récemment.

Nous avons également **3008 doublons** dans notre dataset. Les doublons peuvent engendrer des résultats faussés car les données ne seraient donc pas réalistes. Cependant dans notre jeu de données, nous n'avons pas d'identifiant pour différencier les lignes. Nous avons donc fait le choix de supposer que les lignes en doublons pouvaient être des personnes ayant les mêmes caractéristiques et nous avons décider de garder toutes les lignes.

depart	satisfaction	derniere_evaluation	nombre_de_projets	nombre_heures_mensuelles_moyenne	temps_passe_dans_entreprise
0	0.666810	0.715473	3.786664	199.060203	3.380032
1	0.440098	0.718113	3.855503	207.419210	3.876505

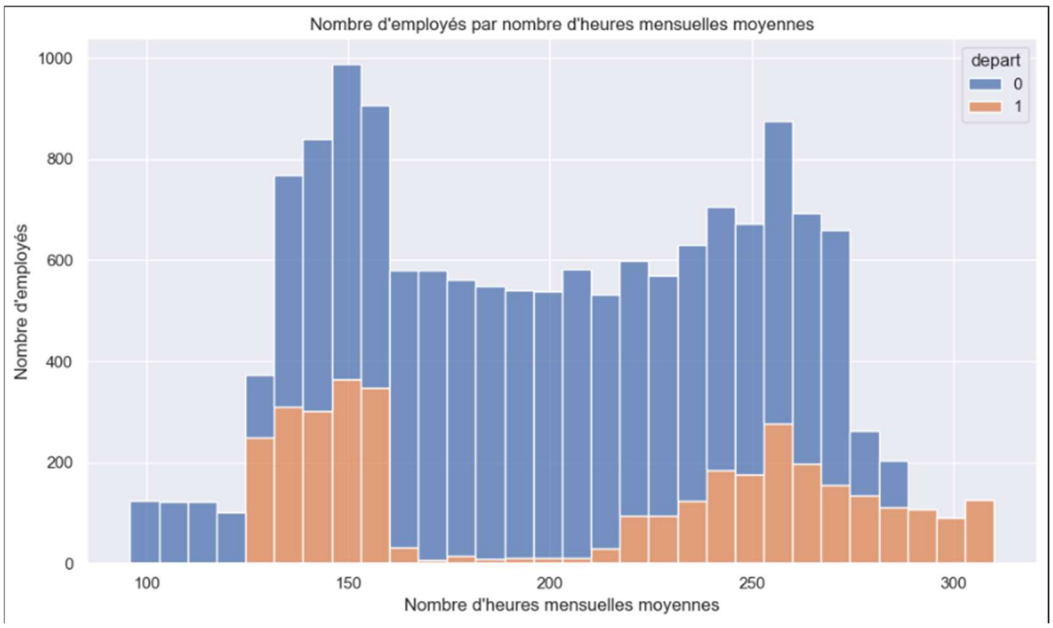
Dans ce graphique, nous pouvons observer la moyenne de différentes variables qui nous semblent judicieuses (variables quantitatives) en fonction des présences ou des départs des employés. Nous pouvons remarquer plusieurs choses :

- La **satisfaction** est **bien supérieure** sur les employés encore présents dans l'entreprise.
- **L'évaluation** des employés est **similaire** peu importe le statut de l'employé
- Le **nombre de projets** sur lesquels travaillent les employés est **similaire** peu importe le statut de l'employé
- Les employés qui sont partis de l'entreprise **travaillaient plus** d'heure en moyenne que les employés encore présents dans l'entreprise
- Les employés partis de l'entreprise ont en moyenne **travaillé plus** dans l'entreprise que les personnes encore actuellement présentes.



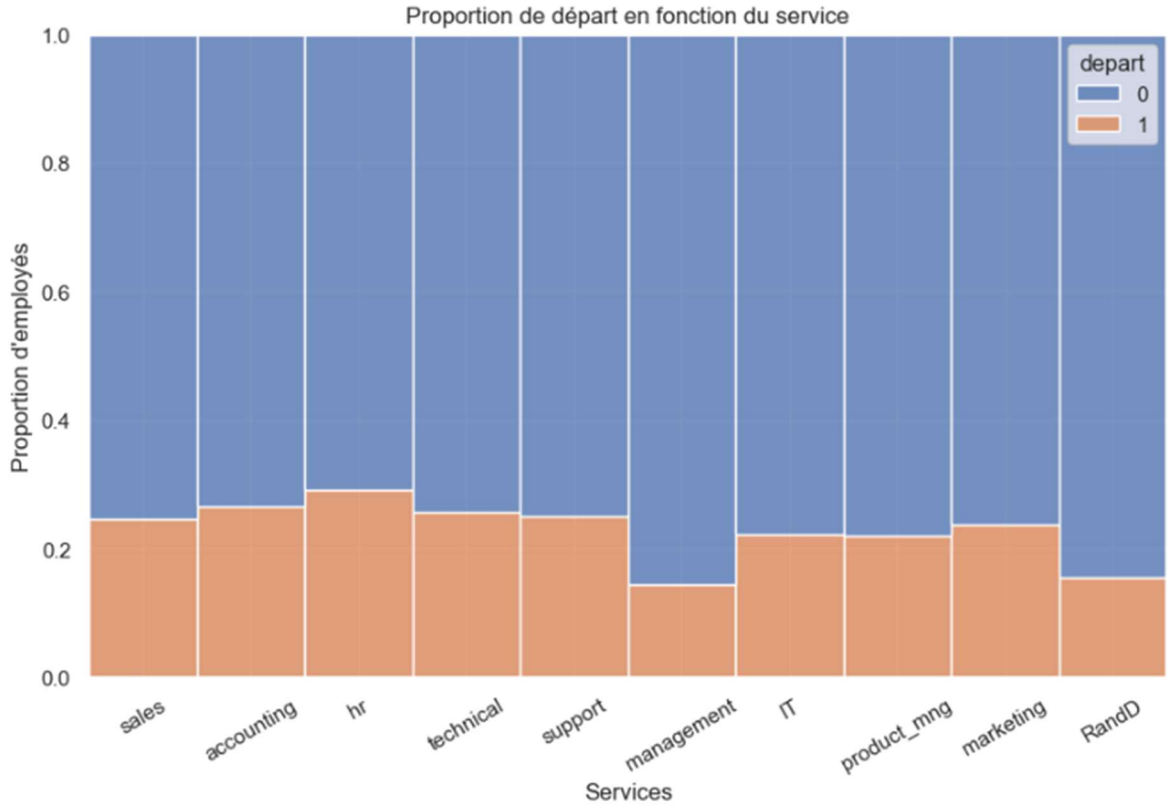
Ce graphique est intéressant car il permet de comparer le niveau des salaires des employés en fonction de s'ils sont partis ou non.
Ce graphique nous permet de remarquer qu'une majorité des employés partis étaient dans le niveau de salaire « low » ou faible en français.

Hypothèse : Plus le niveau de salaire d'un employé est bas, plus il y a de chances que le salarié parte.



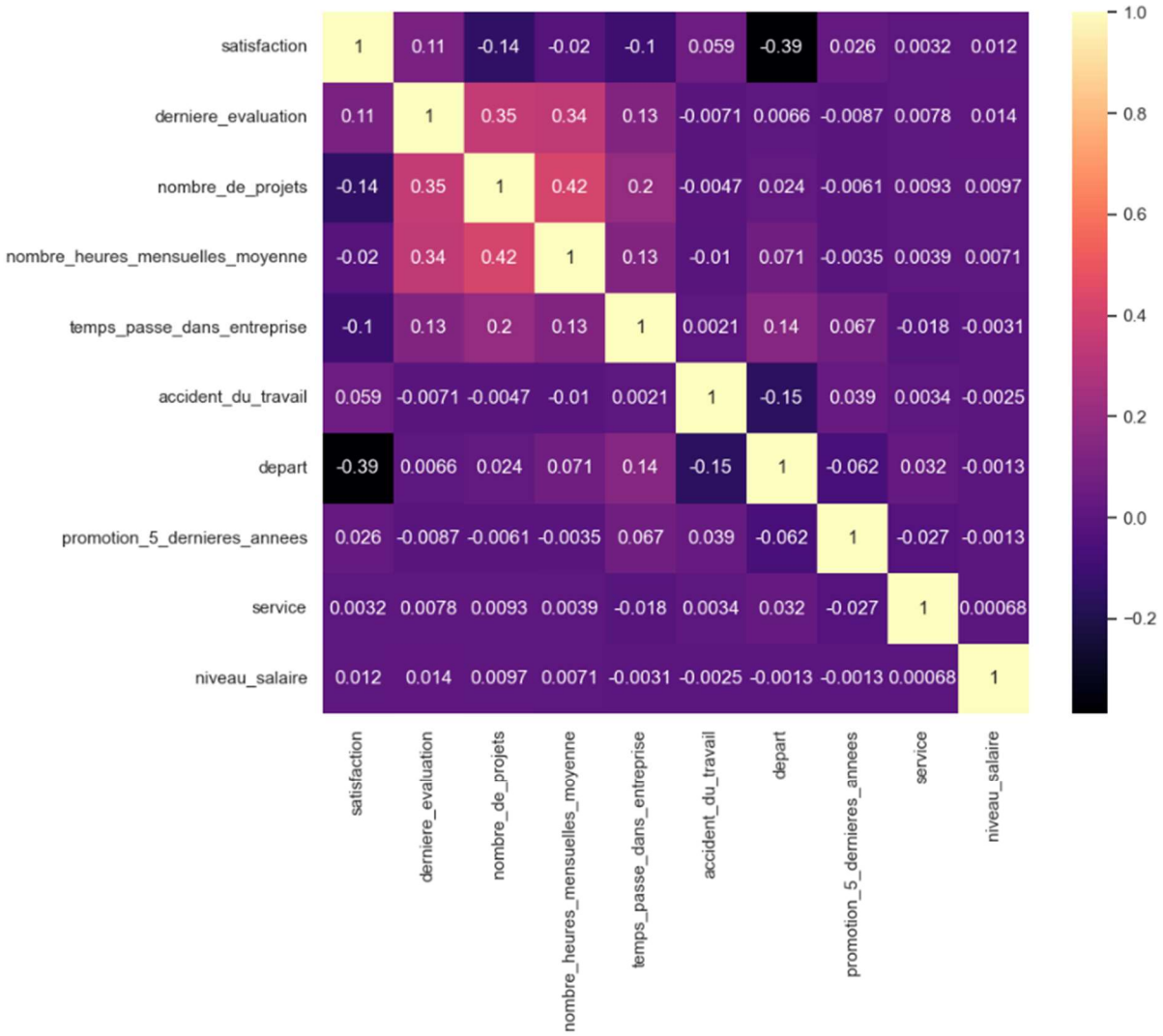
Ce graphique permet de comparer le nombre d'heure mensuelles qui sont travaillées par les employés en fonction de s'ils sont partis ou non.
Ce graphique nous permet de remarquer qu'une majorité des employés travaillant beaucoup (+ 285h/mois) sont partis.

Hypothèse : Plus un employé travaille un nombre d'heures élevée en moyenne par mois, plus il y a de chances que le salarié parte.



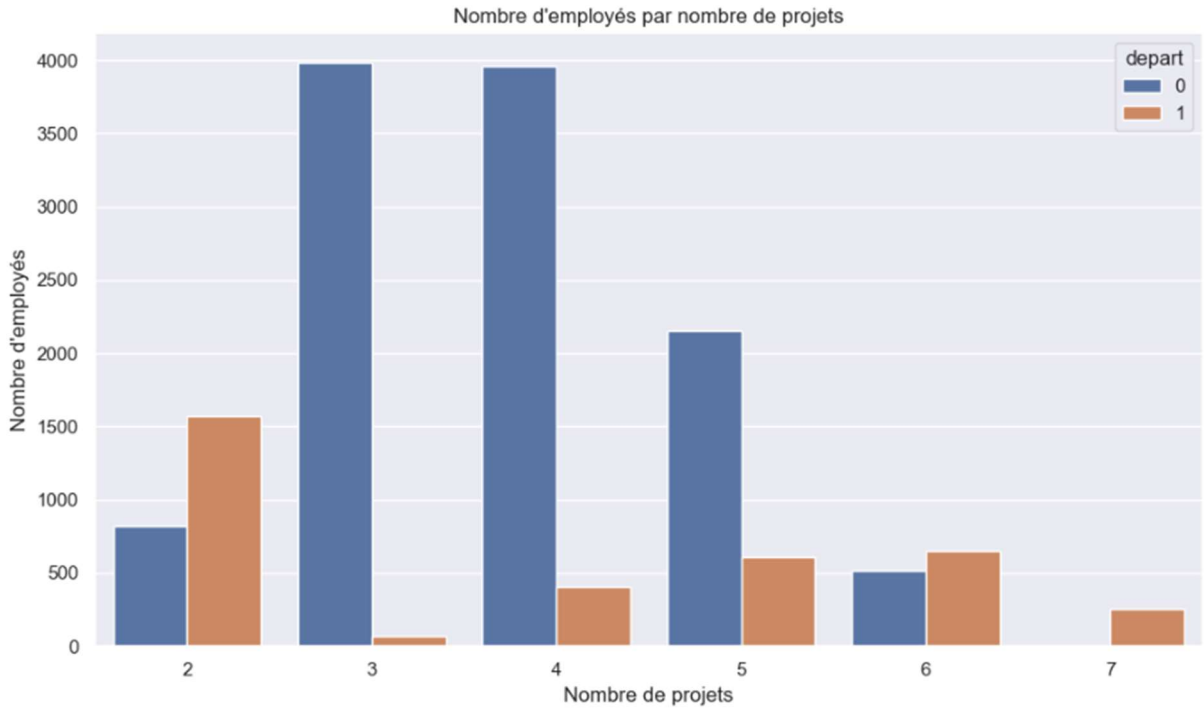
Ce graphique permet de comparer le service dans lequel les employés travaillent en fonction de s'ils sont partis ou non.
Ce graphique nous permet de remarquer qu'il n'y a pas de réel service ou les employés partent plus que d'autres.

Hypothèse : Le service dans lequel travaille l'employé n'a pas de grande incidence sur son départ.



La matrice de corrélation nous permet de voir les variables qui sont les plus corrélés.
Cela nous permet de remarquer plusieurs choses :

- La **satisfaction** des employés a une **corrélation** avec le **départ**
- La **dernière évaluation** des employés a une **corrélation** avec le **nombre de projets travaillés**
- Le **nombre de projets travaillés** a une **corrélation** avec le **nombre d'heures travaillées en moyenne** par mois
- La **dernière évaluation** a une **corrélation** avec le **nombre d'heures travaillées en moyenne** par mois

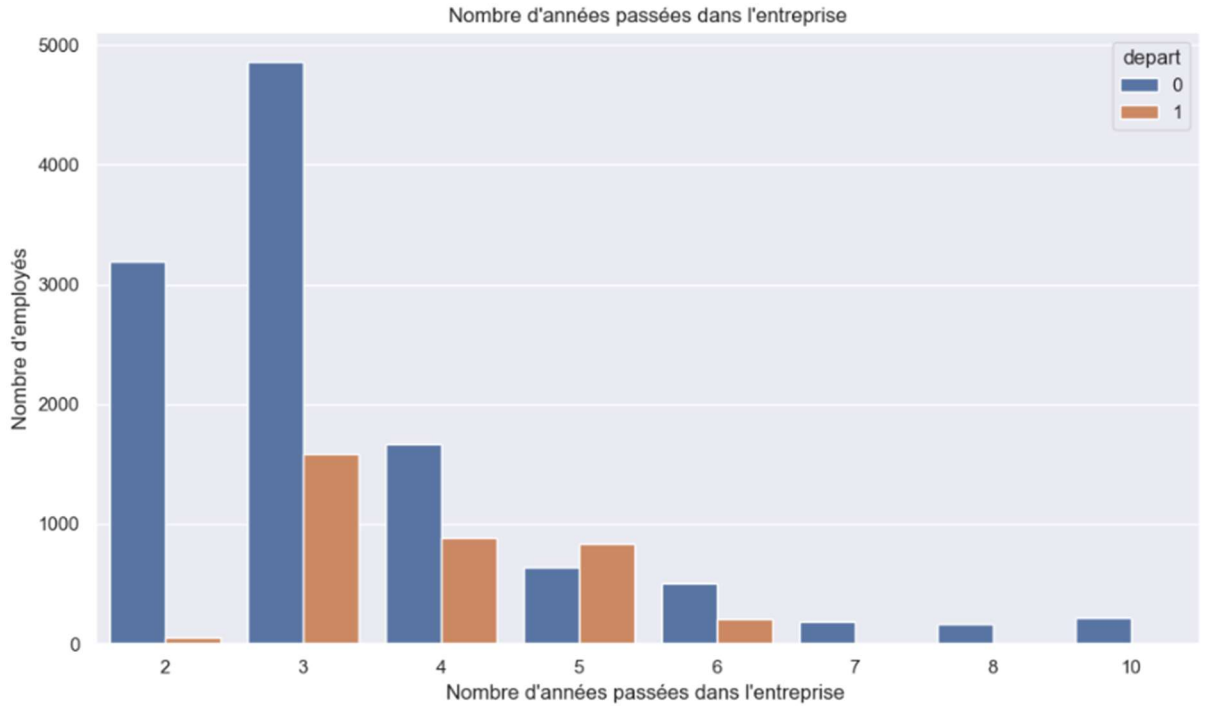


Ce graphique permet de comparer le nombre de projets travaillés par les employés en fonction de s'ils sont partis ou non.

Rappel : La moyenne globale des projets travaillés par les employés est de 3.8

Ce graphique nous permet de remarquer que les employés encore actuellement dans l'entreprise sont en grande majorité entre 3 et 5 projets, tandis que les employés partis sont majoritairement aux extrêmes (2 ou 5+).

Hypothèse : Un nombre de projets trop faible ou trop élevé pourrait entrainer un départ de l'employé.



Ce graphique permet de comparer le nombre d'années travaillés par les employés en fonction de s'ils sont partis ou non.

Rappel : La moyenne globale des années travaillées par les employés est de 3.49

Ce graphique nous permet de remarquer que les employés encore actuellement dans l'entreprise sont en grande majorité récents (4 ans ou moins), tandis que les employés partis sont compris entre 3 et 6 ans. Aux extrêmes (2 ans et 7+), il n'y a pas de départ des employés.

Hypothèse : Les employés récemment entrés dans l'entreprise sont moins à même de partir, ainsi que ceux étant dans l'entreprise depuis plus de 7 a

Descriptif de la méthodologie mise en place, différentes étapes (modèles choisis)

Méthodologie

Recherche Individuelle : Chaque membre de l'équipe a entrepris des recherches individuelles approfondies sur les différentes approches et méthodes utilisées pour identifier les employés à risque de quitter une entreprise. Cette phase préliminaire a permis de collecter une diversité d'informations et d'approches disponibles.

Mise en Commun des Résultats : À la suite de la recherche individuelle, une session de mise en commun a été organisée. Durant cette étape, les membres de l'équipe ont partagé leurs

découvertes et ont identifié quelques modèles distincts pour prédire les départs des employés. Nous présenterons ces modèles dans une partie ultérieure.

Test des Modèles : Chaque modèle a ensuite été testé individuellement par les membres de l'équipe. Cette étape a impliqué l'utilisation des données présentées précédemment pour évaluer l'efficacité de chaque modèle dans la prédiction des départs.

Analyse Comparative : À la suite des tests individuels, une comparaison des résultats a été effectuée lors d'une nouvelle session de mise en commun. Les performances de chaque modèle ont été examinées en termes de précision, de sensibilité et d'autres métriques pertinentes.

Sélection des Meilleurs Modèles : Après une analyse approfondie, les trois modèles présentant les performances les plus prometteuses ont été sélectionnés pour une utilisation ultérieure. Cette décision a été prise de manière collective, en tenant compte des avantages et des limites de chaque modèle.

Analyses Complémentaires et Optimisation : Les trois modèles retenus ont fait l'objet d'analyses complémentaires et d'une phase d'optimisation. Cette étape a inclus des ajustements des paramètres, des validations croisées et d'autres techniques visant à améliorer la robustesse et la précision des modèles sélectionnés.

En suivant cette méthodologie rigoureuse, l'équipe a pu identifier, tester et optimiser les modèles les plus pertinents pour prédire les départs des employés, fournissant ainsi une base solide pour les prochaines étapes de l'étude.

Modélisation

Préparation des données

Avant de tester les différents modèles pour faire notre prédiction, nous sommes passés par une phase de modification des données afin de faciliter l'apprentissage de nos modèles et d'obtenir de meilleurs résultats. Nous avons testé plusieurs modifications et nous avons gardé celles que l'on jugeait les plus pertinentes.

Dans un premier temps, nous avons traité les variables qualitatives (niveau de salaire et service). En effet, nous avons voulu transformer ces variables contenant des chaines de caractères afin que l'on obtienne des valeurs numériques. Pour ce faire, nous avons encodé les variables avec la méthode OrdinalEncoder. OrdinalEncoder transforme chaque variable catégorielle en une séquence de nombres ordinaux. Pour chaque variable catégorielle, les valeurs uniques sont assignées à des entiers ordonnés de 0 à n-1 (n étant le nombre de catégories distinctes).

Voici un exemple d'OrdinalEncoder sur la variable du niveau de salaire :

niveau_salaire
1.0
2.0
2.0
1.0
1.0

Il est également possible de de dichotomiser les variables, c'est-à-dire séparer une variable en autant de variables qu'il y a de valeurs distinctes.

Voici un exemple de dichotomisation sur la variable du service :

Service_IT	Service_RandD	...	Service_hr	Service_management	Service_marketing	Service_product_mng	Service_sales	Service_support	Service_technical
False	False	...	False	False	False	False	True	False	False
False	False	...	False	False	False	False	True	False	False
False	False	...	False	False	False	False	True	False	False
False	False	...	False	False	False	False	True	False	False
False	False	...	False	False	False	False	True	False	False
...	...	...	...	...	...	...	...	...	...
False	False	...	False	False	False	False	False	True	False
False	False	...	False	False	False	False	False	True	False
False	False	...	False	False	False	False	False	True	False
False	False	...	False	False	False	False	False	True	False
False	False	...	False	False	False	False	False	True	False

Les différents classifieurs

Arbre de décision

Les arbres de classification et de régression, aussi connus sous le nom d'arbres de décision, peuvent être représentés sous la forme d'arbres binaires. Un arbre de décision peut être décrit comme une représentation visuelle d'un algorithme de classification. Il s'agit de suivre différents critères de décision.

L'arbre de décision permet, après entraînement sur un ensemble de données, d'effectuer facilement des prédictions sous la forme de règles logiques successives de classification. Les résultats sont ainsi facilement interprétables et donc exploitables, la communication autour de la modélisation plus aisée.

Random Forest

Le random forest est composé de plusieurs arbres de décision, travaillant de manière indépendante sur une vision d'un problème. Chacun produit une estimation, et c'est l'assemblage des arbres de décision et de leurs analyses, qui va donner une estimation globale. En somme, il s'agit de s'inspirer de différents avis, traitant un même problème, pour mieux l'appréhender. Chaque modèle est distribué de façon aléatoire aux sous-ensembles d'arbres décisionnels.

Un random forest fonctionne sur le principe du bagging. Il s'agit en effet d'un cas de bagging spécialement appliqué sur l'algorithme des arbres de décision. La première étape consiste à découper un dataset en sous-ensembles (arbres de décision), puis de proposer un modèle d'entraînement à chacun de ses groupes. Enfin, on combine les résultats de ces arbres afin d'obtenir la prévision la plus solide.

Régression Logistique

La régression logistique en machine learning est une technique utilisée pour prédire des résultats binaires. Elle modélise la probabilité qu'un résultat soit vrai en fonction des variables d'entrée. En d'autres termes, elle cherche à trouver la relation entre les caractéristiques des données et la probabilité d'un résultat spécifique. C'est un outil utile pour la classification binaire, comme la prédiction de fraudes, la détection de spam ou la prédiction de l'issue d'un événement.

KNN

Le modèle KNN (K plus proches voisins) en machine learning est une méthode qui repose sur le principe que des choses similaires sont proches les unes des autres. Pour prédire la classe ou la valeur d'une nouvelle donnée, KNN regarde les K exemples d'entraînement les plus proches dans l'espace des caractéristiques et prend une décision en fonction de la majorité de leurs étiquettes ou valeurs. C'est comme demander à vos voisins les plus proches leur opinion pour prendre une décision.

SVM

Le modèle SVM (Support Vector Machine) est un outil de machine learning qui cherche à trouver la meilleure ligne ou le meilleur plan pour séparer des points de données dans différents groupes. Son objectif est de trouver la "frontière de décision" qui maximise la distance entre les points des différents groupes, appelés vecteurs de support. En d'autres

termes, c'est comme dessiner la meilleure ligne possible entre deux groupes de données pour les séparer au mieux.

Réseau de Neurones

Un réseau de neurones est un modèle informatique inspiré du cerveau humain. Il est composé de nombreux nœuds interconnectés, appelés neurones artificiels, qui travaillent ensemble pour résoudre des problèmes complexes. Chaque neurone prend des entrées, effectue des calculs et transmet des signaux à d'autres neurones, contribuant ainsi à la prise de décision ou à la prédiction de résultats.

Optimisation

Afin d'obtenir les meilleures prédictions possibles, nous avons testé les 6 modèles présentés précédemment.

Pour chaque modèle, nous avons utilisé la fonction GridSearch. Cette fonction est utilisée pour rechercher les meilleurs hyperparamètres pour un modèle d'apprentissage automatique. Elle explore de manière systématique différentes combinaisons d'hyperparamètres spécifiées dans une grille prédéfinie, puis évalue chaque combinaison à l'aide d'une technique de validation croisée pour déterminer celle qui donne les meilleurs résultats en termes de performance du modèle.

Cependant, cette fonction n'est pas compatible avec certains modèles comme le modèle SVM et la régression logistique. Pour ces deux modèles, nous avons donc choisi les paramètres qui nous semblaient les mieux adaptés à notre problématique.

Comparaison

Rapport de classification

	Decision Tree	Random Forest	Logistic Regression	KNN	SVM	Neural Network
Train Accuracy	0.987	0.987	0.837	0.955	0.834	0.826
recall	0.979	0.980	0.835	0.940	0.833	0.833
precision	0.979	0.980	0.798	0.942	0.694	0.694
f1-score	0.978	0.980	0.801	0.941	0.757	0.757

Les modèles d'arbre de décision et de random forest affichent les meilleures performances, avec des scores de précision, de recall et de F1-score supérieurs à 97%, ce qui indique une capacité élevée à classifier correctement les données.

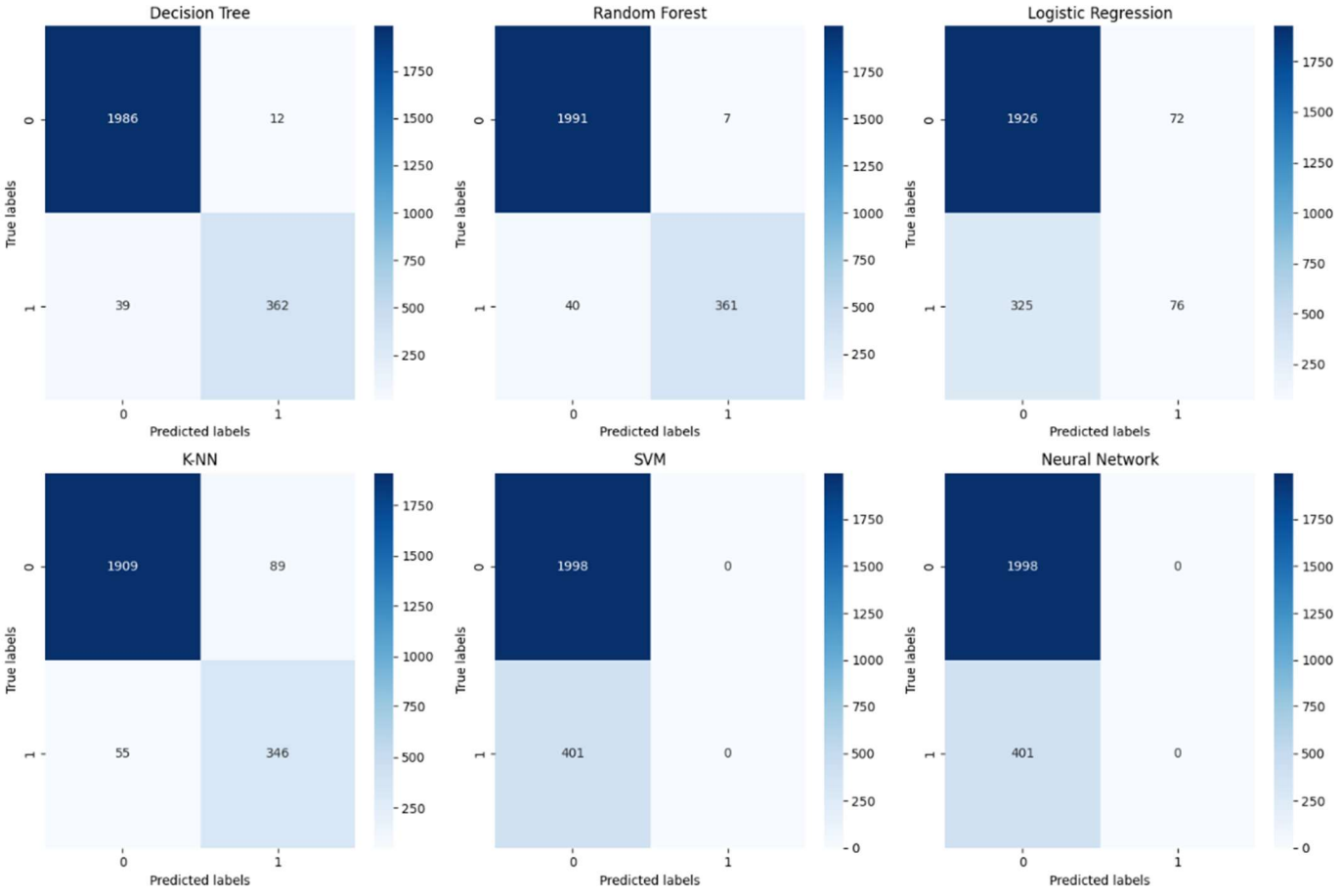
En revanche, la régression logistique montre des performances relativement inférieures avec des scores de précision, de recall et de F1-score autour de 80%.

Le modèle KNN obtient de bons scores, mais légèrement inférieurs à ceux des arbres de décision et du random forest, avec des scores autour de 94%.

Les modèles SVM et réseaux neuronaux affichent des performances en dessous des autres modèles, avec des scores de précision, de recall et de F1-score autour de 70%. Cela suggère qu'ils pourraient nécessiter une optimisation plus poussée des hyperparamètres ou que ces types de modèles ne sont pas vraiment adaptés à notre problématique.

En résumé, les arbres de décision et le random forest se démarquent comme les meilleurs modèles dans cette comparaison, suivis du modèle KNN. La régression logistique, les SVM et les réseaux neuronaux montrent des performances légèrement inférieures, mais pourraient bénéficier d'un ajustement supplémentaire pour améliorer leurs résultats.

Matrice de confusion



Les matrices de confusions nous permettent de voir plus précisément les valeurs qui ont été prédites par nos modèle. On peut voir plus précisément le nombre de valeurs qu'il a réussi à prédire et s'il a mieux prédit les départs ou les non-départs.

Pour nos deux meilleurs modèles, à savoir l'arbre de décision et le random forest, on voit qu'ils ont relativement la même répartition des résultats. En effet, les erreurs qu'ils ont obtenues sont principalement sur des départs qui ont été prédits en non-départs.

A l'inverse, le modèle KNN obtient moins d'erreurs sur les faux négatifs que sur les faux positifs (55 contre 89). C'est une bonne chose car il est plus embêtant de prédire des faux négatifs que des faux positifs.

Pour la régression logistique, notre modèle a plus d'erreurs que de bonnes prédictions quand il s'agit de prédire les départs. On confirme donc grâce à cette matrice de confusion que la régression logistique n'est pas le modèle le plus adapté.

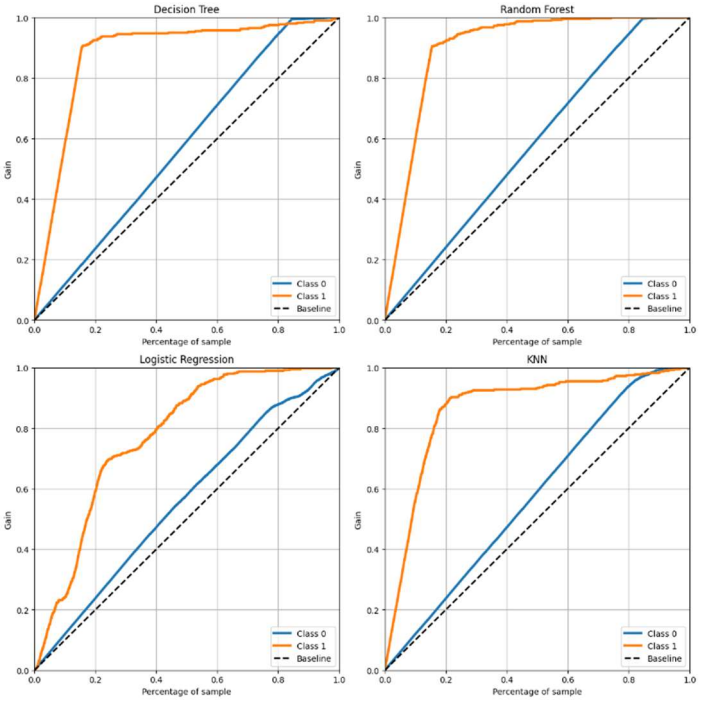
En ce qui concerne le modèle SVM et le réseau de neurones, on observe que ces deux modèles n'ont prédit que des non-départs. Cela explique les mauvais scores observés précédemment et confirme que ces modèles ne répondent pas à notre problématique, vu qu'ils ne nous donnent aucune prédiction de départ.

Courbe de gain cumulé

Les courbes de gain cumulé nous permettent de confirmer les résultats précédents. En effet sur ces graphiques, plus la courbe orange monte vite vers 1 et plus le modèle est performant.

Ici on observe une fois de plus que l'arbre de décision et le random forest sont les meilleurs modèles.

On voit également que la régression logistique n'a pas de très bons résultats.



Pour la suite de notre analyse, nous avons donc décidé de ne garder que les modèles d'arbre de décision, de Random Forest et de KNN pour une analyse plus rapide et plus pertinente et évitant de consacrée du temps sur des modèles peu performants.

Dichotomisation

Après avoir tester les 6 modèles précédents, nous avons tenté de modifier nos données initiales voir si cela avait un impact sur nos résultats et s'il y avait une possibilité de les améliorer. Pour ce faire, nous avons tout d'abord dichotomiser nos variables catégorielles.

	Decision Tree	Random Forest	KNN
Train Accuracy	0.982	0.982	0.989
recall	0.976	0.978	0.949
precision	0.976	0.978	0.948
f1-score	0.975	0.978	0.948

Après avoir réentraîné nos modèles sur nos données dichotomisées, nous obtenons les résultats ci-dessus.

On observe donc que la dichotomisation a légèrement fait baisser les performances de nos modèles d'arbre de décision et de random forest. Cependant, cela a augmenté les résultats de notre modèle KNN. La différence avec ou sans dichotomisation n'étant pas significative, nous avons décidé de ne pas garder cette modification sur nos variables.

Normalisation

De même que pour la dichotomisation, nous avons ensuite normaliser nos données pour tenter de voir si cela aurait un impact sur nos données.

	Decision Tree	Random Forest	KNN
Train Accuracy	0.982	0.985	0.991
recall	0.975	0.980	0.965
precision	0.975	0.980	0.964
f1-score	0.975	0.979	0.965

Cependant, on remarque très peu de changement. En effet, cela peu d'une part s'expliquer par le fait que nos variables aient des échelles globalement comparables. De plus, les algorithmes d'arbre de décision et de random forest sont peu sensible à l'échelle des variables. C'est donc pour cette raison que les scores sont quasi identiques pour eux. Néanmoins, on remarque une petite augmentation pour le modèle KNN qui est passé d'un recall de 0,949 à 0,965.

De ce fait, nous avons également décider de garder notre jeu de donnée de base et de ne pas normaliser nos données.

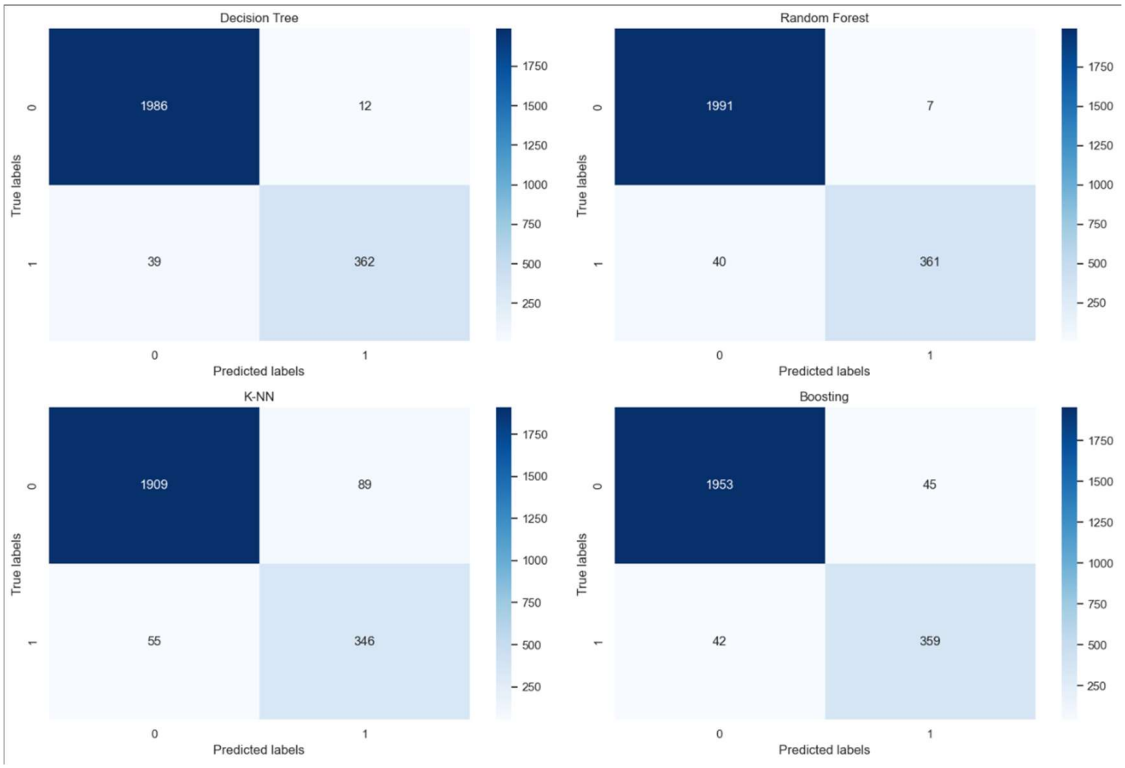
Gradient Boosting

Au vu des résultats de l'arbre de décision et du random forest, nous avons trouvé judicieux de réaliser un boosting.

Le principe du boosting est de combiner les sorties de plusieurs classifieurs faibles pour obtenir un résultat plus fort. Chaque classifieur faible est pondéré par la qualité de sa classification : mieux il classe, plus il sera important. Les exemples mal classés auront un poids plus important vis-à-vis de l'apprenant faible au prochain tour, afin qu'il pallie le manque.

Comme précédemment, nous avons également réaliser une optimisation des hyper paramètres. Après cela, les résultats étaient plus qu'encourageant.

	Decision Tree	Random Forest	KNN	Boosting
Train Accuracy	0.987	0.987	0.955	1.000
recall	0.979	0.980	0.940	0.988
precision	0.979	0.980	0.942	0.988
f1-score	0.978	0.980	0.941	0.988

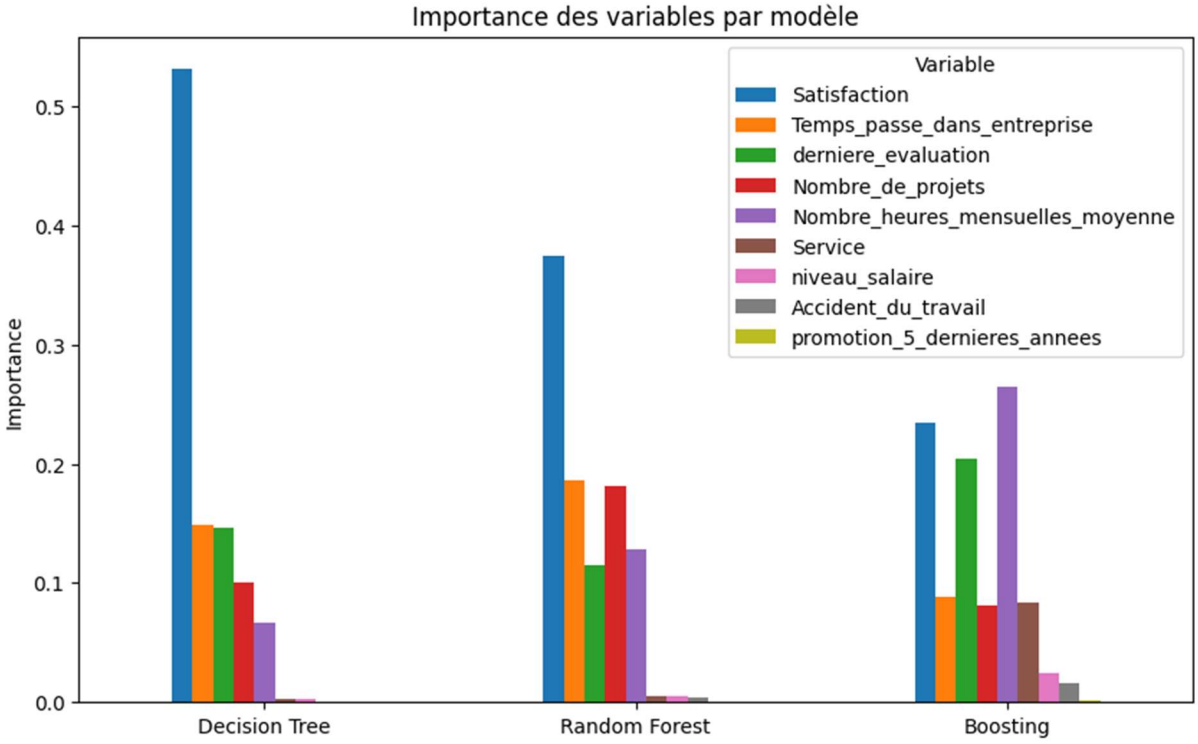


En effet, on peut voir que pour le jeu de donnée d'entraînement, le modèle à atteint **100%** de précision. Pour ce qui est du jeu de test, la précision reste tout de même à **98,8%** ce qui est quand même extrêmement bon et supérieur aux autres classifieurs.

On peut voir grâce aux matrices de confusion, que le modèle prédit mieux que les autres les vrais positifs, plus précisément les employés étant réellement partis.

Importance des variables

Pour comparer nos modèles, nous avons ensuite décidé d'analyser l'importance des variables pour chaque modèle. Nous avons utilisé l'attribut **feature_importances_** présent dans scikit-learn. Ce dernier contient l'importance normalisée des variables dans la prise de décision (les choix réalisés par l'arbre). Il n'est donc pas disponible pour le KNN.



Grâce à ce graphique, on peut donc remarquer qu'en général, la **satisfaction au travail** ressort comme étant le facteur prédominant pour tous les modèles, affichant des valeurs se situant entre environ 0,23 et 0,53. Le **temps passé au sein de l'entreprise** et la **dernière évaluation** semblent également avoir une certaine importance, bien que cette importance varie d'un modèle à l'autre. Par exemple, le modèle de boosting accorde apparemment plus d'importance au **nombre d'heures mensuelles moyennes** que les deux autres modèles. Les autres caractéristiques semblent présenter une importance relativement faible, avec des valeurs proches de zéro pour certaines d'entre elles.

On remarque également une certaine différence entre les modèles. Cette variabilité des résultats entre les modèles peut être due à la nature même des algorithmes. Par exemple, les arbres de décision peuvent avoir tendance à favoriser certaines features par rapport à d'autres, tandis que les méthodes ensemblistes comme Random Forest et Boosting peuvent prendre en compte une plus grande diversité de features.

Synthèses des résultats

Grâce à toutes ces analyses, nous avons pu obtenir de nombreux résultats et retenir le plus précis. En effet, la classification de variable binaire est une des choses les plus simples dans le domaine du machine learning. Nous avons alors pu tester de nombreux modèles rapidement avec de nombreux paramètres différents nous permettant de retenir les meilleurs.

Nous avons donc constaté que pour ce projet, l'utilisation d'arbre de décision et de random forest était de loin le plus performant. Cela nous a alors indiqué à utiliser le boosting pour améliorer notre modèle.

Ce choix à été très utile étant donné que nous avons réussi à créer un modèle le modèle le plus performant de tous ceux essayer précédemment (~99%).

Nous avons également remarqué l'importance de la satisfaction au travail, du temps passé au sein de l'entreprise et de la dernière évaluation. En effet, ces variables ont un réel impact sur le départ d'un salarié de l'entreprise.

Difficultés rencontrées

L'un des défis majeurs rencontrés lors de ce projet a été la difficulté à trouver des moments propices pour collaborer et échanger sur notre avancement. Les disparités dans nos emplois du temps, nos lieux de résidence et nos contraintes personnelles ont rendu la coordination complexe. Cette expérience nous a permis de réaliser l'importance de planifier et segmenter efficacement le travail dès le début du projet afin de minimiser les interférences et d'optimiser notre productivité.

Un autre point bloquant a été le temps de calcul de certains hyperparamètres. Dans l'objectif d'avoir les meilleurs résultats possibles nous avons optimisé au maximum les paramètres de chaque modèle mais pour cela il fallait tous les tester pour en retirer les meilleurs.

Perspectives d'améliorations

Afin d'améliorer nos résultats, nous aurions pu tester d'utiliser l'ACP pour réduire la dimension de notre jeu de données. L'ACP est une méthode de réduction de données s'appuyant sur l'algèbre linéaire afin de compresser un jeu de données.

Comme nous l'avons fait en cours, nous aurions pu ajouter une partie dans le pre-processing pour faire une sélection des variables avec un wrapper. Le principe est de générer des sous-ensembles candidats et de les évaluer grâce à un algorithme de classification. Cette évaluation est faite par le calcul d'un score, par exemple un score d'un ensemble sera un compromis entre le nombre de variables éliminées et le taux de réussite de la classification sur un fichier de test. Un mécanisme supplémentaire de validation croisée est fréquemment utilisé. Le principe de wrappers est de générer un sous ensemble bien adapté à l'algorithme de classification. Les taux de reconnaissance sont élevés car la sélection prend en compte le biais intrinsèque de l'algorithme de classification. Un autre avantage est sa simplicité conceptuelle ; nous n'avons pas besoin de comprendre comment l'induction est affectée par la sélection des variables, il suffit de générer et de tester.

Bien que le réseau de neurones de que nous avons testé ne nous ait pas donné de bons résultats, nous pensons qu'il aurait été possible de l'améliorer. En effet, nous avons fait un réseau de neurones relativement simple mais en ajoutant des couches et changeant certains paramètres, nous aurions pu obtenir d'autres résultats.

Nous aurions également pu tester un modèle XGBoost qui est un modèle souvent utilisé dans les problèmes de classifications.

Bilan final

Dans le cadre de cette étude, nous avons exploré divers modèles analytiques avancés pour prédire avec précision, les départs des employés et pour aider les entreprises à prendre des mesures préventives. A travers une méthodologie rigoureuse, nous avons analysé les données disponibles et testé plusieurs modèles notamment le K-Nearest Neighbors (KNN), le Support Vector Machine SVM, Random Forest, la régression logistique et des réseaux de neurones.

Les résultats obtenus ont révélé le potentiel de ces modèles à fournir des insights significatifs pour anticiper les départs des employés. Nous avons observé des corrélations importantes entre des facteurs tels que la satisfaction des employés, leurs évaluations, leur charge de travail et leur ancienneté dans l'entreprise avec leurs départs éventuels. En outre, l'utilisation de techniques avancées comme le gradient boosting a permis de capturer des relations complexes dans les données, conduisant ainsi à des prédictions plus précises.

Cependant, notre étude a également mis en lumière des défis importants. La coordination des horaires de travail et la gestion des contraintes géographiques ont entravé la collaboration au sein de l'équipe.

Malgré ces défis, notre approche méthodologique rigoureuse et notre sélection diversifiée de modèles ont permis de développer une compréhension approfondie des facteurs influençant les départs des employés.

En conclusion, l'intégration de modèles analytiques avancés, tels que l'intelligence artificielle, dans les processus de gestion des ressources humaines peut jouer un rôle crucial dans la rétention des talents et la stabilité organisationnelle. En associant ces modèles à une approche centrée sur l'humain et à une gestion proactive de l'engagement des employés, les entreprises peuvent prendre des mesures préventives plus efficaces pour favoriser un environnement de travail positif et durable.