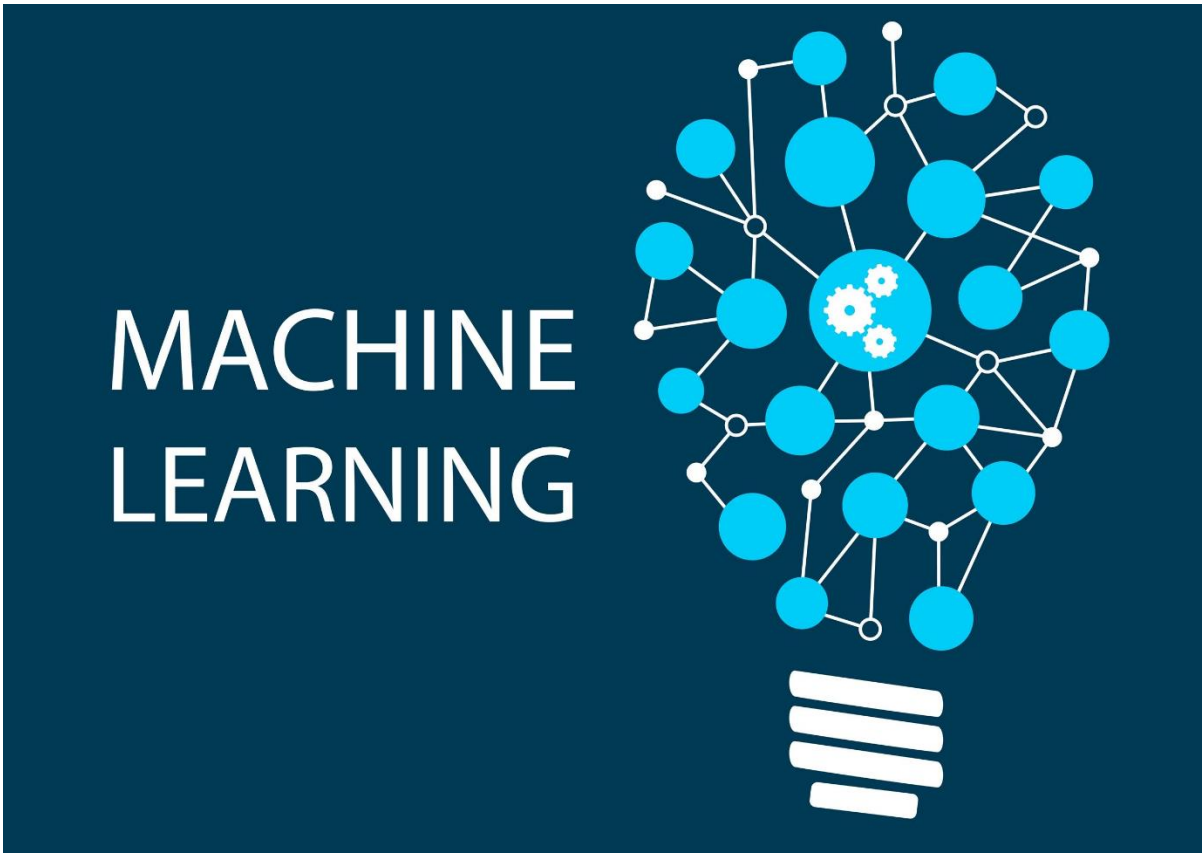
13 AVRIL 2022



ANALYSE DE DONNEES
PROJET - TURNOVER

USSI4X - ENTREPOSAGE ET FOUILLE DE DONNEES

NINON QUESNEL / LEONIE BERTON / PAUL REVERSAC / NICOLAS PERICHON
EICNAM NIORT

SOMMAIRE

1. Présentation du projet

Nous avons pour objectif de déterminer, à partir d'un fichier de données de type csv, quels employés sont susceptibles de quitter ou non l'entreprise. Pour cela, nous allons créer plusieurs modèles afin d'identifier lequel est le plus fiable pour prédire des données de source externe. Une fois aboutis, nous aurons pour objectif d'améliorer, d'optimiser notre apprentissage.

1.1. Présentation des données

Libellé	Description	Type
Satisfaction	Note sur 1 représentant le taux de satisfaction du salarié dans son métier	Float
Derniere_evaluation	Dernière note attribuée au collaborateur	Float
Nombre_de_projets	Nombre de missions sur lesquelles le salarié travaille	int
Nombre_heures_mensuelles_moyenne	Nombre d'heures travaillées en moyenne par mois pour un employé	int
Temps_passe_dans_entreprise	Nombre d'années passées dans l'entreprise par le salarié	int
Accident_du travail	L'employé a déjà eu un accident ou non ?	int

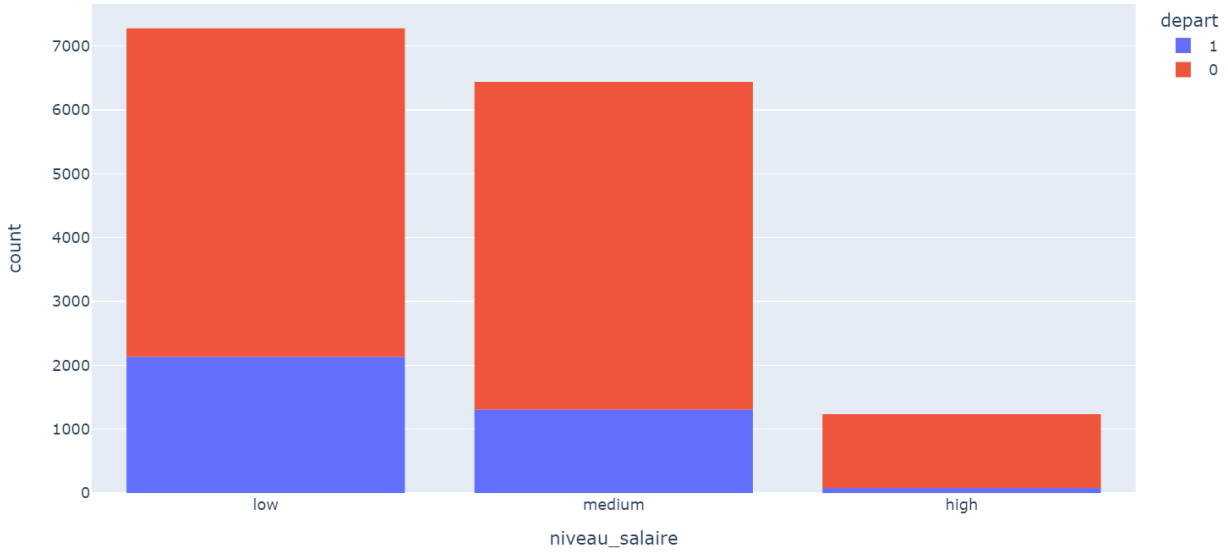
Depart	L'employé a-t-il quitté l'entreprise ? 1 : Oui / 0 : Non	int
Promotion_5_dernieres_annees	Le salarié a-t-il eu une promotion ces 5 dernières années ? 1 : Oui / 0 : Non	int
Service	Service dans lequel l'employé travaille (sales …)	object
Niveau_salaire	Niveau de salaire de l'employé (low, medium, high)	object

Le jeu de données contient 14 950 enregistrements ainsi que 10 variables. Notre variable cible est la variable ***depart***. Nous allons à partir de modèles, déterminer si le salarié va partir ou non de l'entreprise.

Nous avons réalisé quelques analyses de bivariées afin de mieux connaître notre jeu de données.

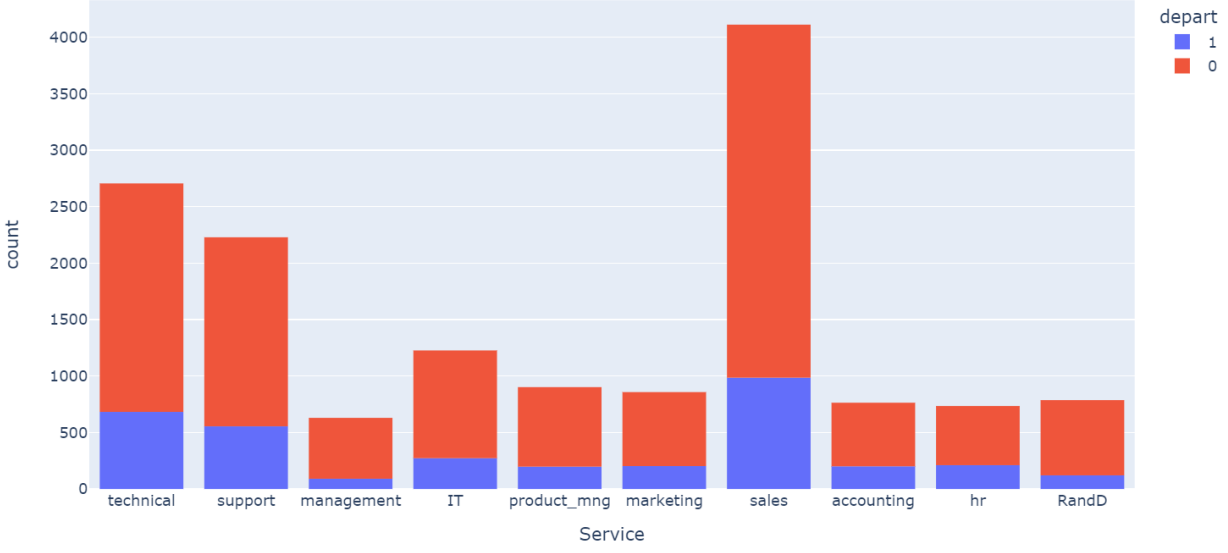
1.2. Data Viz

Répartition des départs en fonction des niveaux de salaire



Ici, nous avons représenté le niveau de salaire des salariés en fonction des variables de départ. Nous pouvons remarquer que plus le salaire augmente moins les employés vont avoir tendance à partir. 29% des employés ayant un salaire bas vont partir, 20% avec un salaire moyen vont partir et seulement 6% des employés avec un salaire haut vont partir de l'entreprise. Nous pouvons donc penser qu'il y a une corrélation entre la variable salaire et la variable départ.

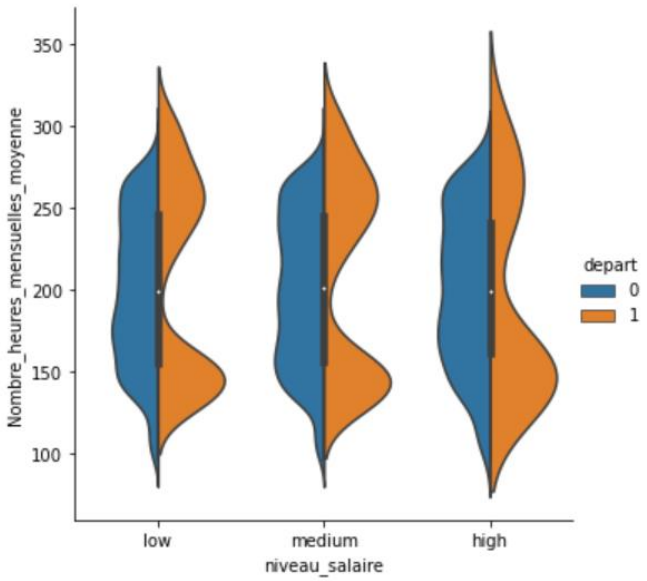
Répartition des départs en fonction des services



Nous avons ensuite été voir les départs en fonction des services où les employés travaillent. Pour le service, on retrouve environ entre 25 et 30% des employés qui partent sauf pour le service management et le service RandD où 15% des employés partent.
Il n'y a donc pas une forte corrélation entre le type de service et le départ des employés.

Ensuite nous avons regardé la relation entre le niveau de salaire et le nombre d'heures mensuelles moyenne travailler en regardant également la variable départ.
Nous avons réalisé un *catplot* car ce module permet de travailler efficacement avec des données catégorielles.

Répartition des départs en fonction des heures mensuelles travaillées et du niveau de salaire associé



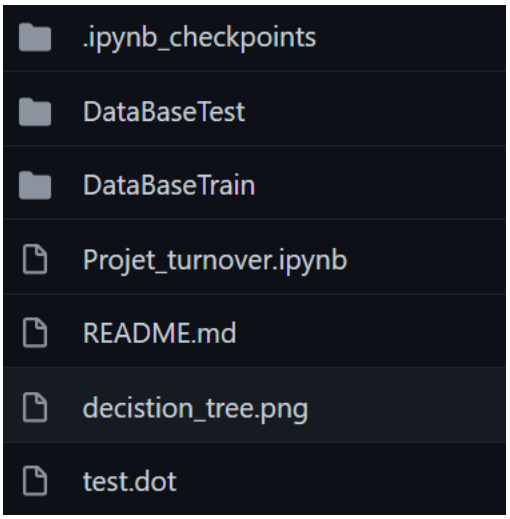
Nous pouvons remarquer que le nombre d'heures mensuelles n'influence pas sur le salaire, nous voyons que la répartition des heures est assez équivalente pour les 3 catégories de salaire. De même, pour la répartition des départs elle est à peu près similaire sur les 3 figures. Les variables ***Nombres_heures_mensuelles_moyenne*** et ***Niveau_salaire*** ne sont donc pas fortement corrélées.

2. Description de la méthodologie et des modèles

2.1. Architecture du projet

Nous avons créé un répertoire *DataBaseTrain* dans lequel se situe le fichier csv de base. Il sera utilisé pour l'entraînement de nos modèles.

Nous avons également créé un répertoire *DataBaseTest* dans lequel nous situons un fichier "externe" (reprenant le schéma de la base de données de *train*) pour tester nos modèles. Pour exécuter notre code nous avons créé un fichier *jupyter notebook* sur lequel nous réalisons notre développement à la racine du projet GIT.



2.2. Plan de développement

Pour répondre à notre problématique nous avons créé un code que nous avons organisé comme ceci :

En haut de page de code on retrouve tous les imports qui seront nécessaires au fonctionnement de notre code.

En premier lieu se trouve notre fonction d'affichage qui nous permettra de retourner la valeur, ici le départ ou non de l'employé, pour un enregistrement, soit un salarié de notre base de données.

On importe ensuite les données à partir du fichier csv donné. On affiche les différents types des variables puisqu'il sera nécessaire de n'avoir que des nombres pour la suite. Cela permet donc d'identifier les variables à traiter.

Ensuite on affiche différentes informations concernant la base de données, notamment à l'aide de la fonction .describe(). Elle permet d'afficher pour nos 10 variables, le nombre de fois où on les retrouve, leur moyenne respective, leur minimum… Cela permet notamment de savoir sur combien est évaluée la satisfaction ou bien la dernière évaluation…

On réalise ensuite différents graphiques pour constater si la proportion de départ ou non a un rapport avec différentes variables.

On réalise également une matrice de corrélation pour remarquer quelles variables sont plus importantes que d'autres dans la corrélation avec le départ du salarié, cela permet d'avoir un premier aperçu.

On fait également un test du V de Cramer qui sera intéressant pour comparer les variables que le modèle choisit automatiquement avec les résultats de la matrice précédente. Cela nous donne deux analyses pour confirmer notre choix ou non de certaines variables.

On compare les taux de fiabilité de chaque modèle et on affiche les matrices de confusion qui permettent de voir les faux positifs et les faux négatifs.

On en conclut le meilleur modèle parmi les 4.

On réitère ensuite en réalisant une feature engineering. On crée 3 nouvelles variables en fonction des autres :

- ***heures_totales*** : c'est la multiplication du *Nombre_heures_mensuelles_moyenne* par 12 (une année) et par le *Temps_passe_dans_entreprise* (en années).
- ***projets_par_années*** : c'est le *Nombre_de_projets* par rapport au *Temps_passe_dans_entreprise*.
- ***satisf_eval*** : c'est la Satisfaction divisée par *Deniere_evalutation*.

On refait une matrice de corrélation suivie d'un VCramer. On constate avec la matrice de corrélation que nos 3 nouvelles variables sont bien parmi les plus corrélées. On réalise cette fois nos 4 modèles précédents sur les nouvelles meilleures features.

On compare les résultats et on conclut.

2.3. Modèles utilisés

On entraîne ensuite notre jeu de données sur 4 modèles :

- Random Forest
- Logistic Regression
- KNN
- SVM

La random forest est composée 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. Il s'agit donc de s'inspirer de différents avis, traitant un même problème, pour mieux l'appréhender.

La régression logistique est un modèle statistique permettant d'étudier les relations entre un ensemble de variables qualitatives Xi et une variable qualitative Y. Un modèle de régression logistique permet aussi de prédire la probabilité qu'un événement arrive (valeur de 1) ou non (valeur de 0) à partir de l'optimisation des coefficients de régression. Ce résultat varie toujours entre 0 et 1.

Le KNN est la méthode des k plus proches voisins (ou k-nearest neighbors en anglais), c'est une méthode d'apprentissage supervisé. Elle permet de classifier des points dans des classes connus en fonction de leur distance. Cette méthode est très utile pour résoudre des problèmes de classification et de régression.

Les SVM ou Support Vector Machine, sont des algorithmes d'apprentissage automatique. Ils servent à résoudre des problèmes de classification, de régression ou encore de détection d'anomalie. On peut les utiliser pour résoudre des problèmes de discrimination, donc décider à quelle classe appartient un échantillon, ou de régression, prédire la valeur numérique d'une variable.

3. Description des résultats obtenus

Considérant que l'on travaille principalement avec des variables qualitatives, l'un des tests que l'on peut envisager est celui du Khi-2, pour tester l'indépendance. À titre d'exemple, il est possible de tester les hypothèses suivantes :

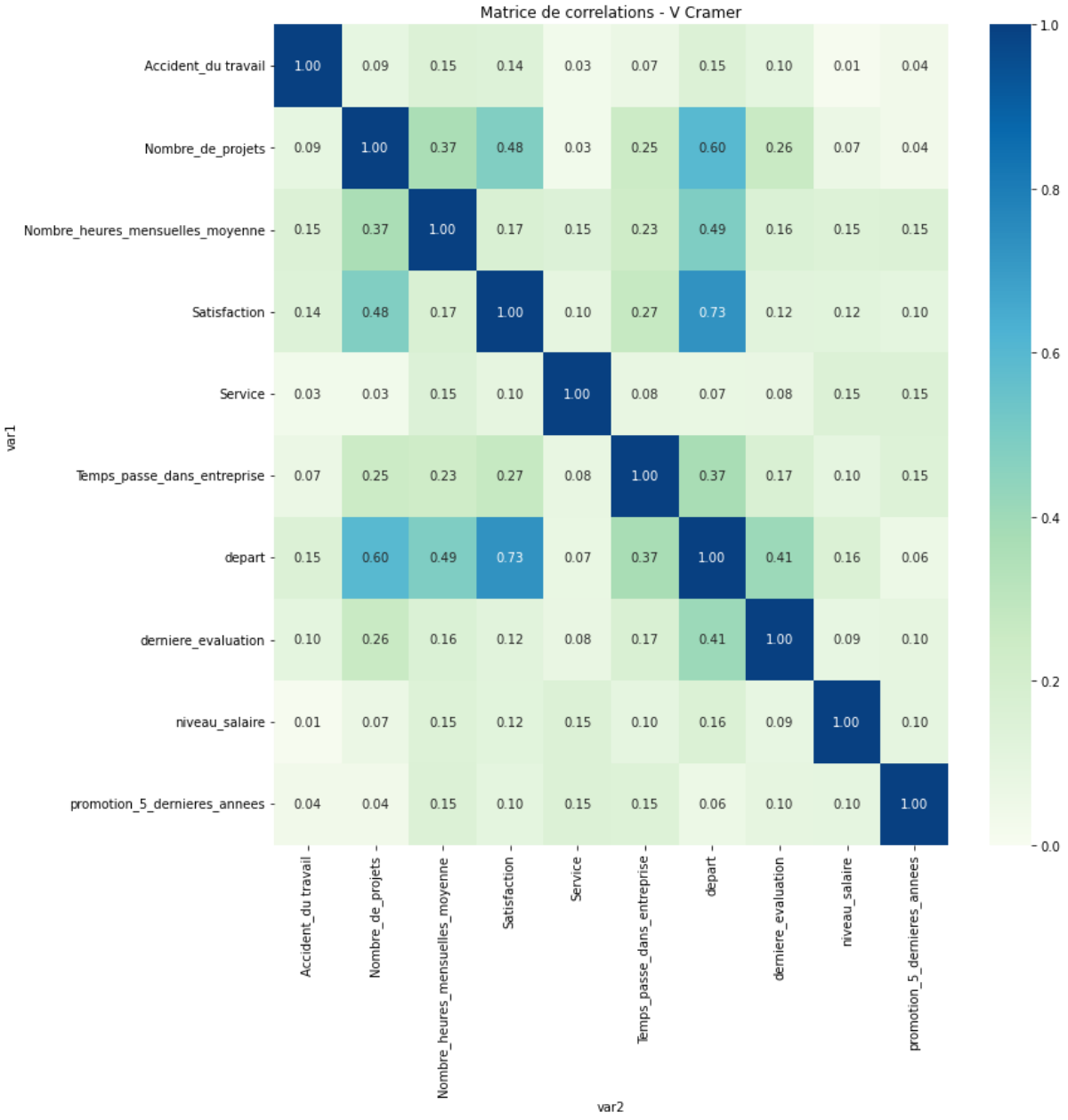
- H0 : le départ des employés est indépendant de leur service dans l'entreprise
- H1 : le départ des employés est dépendant de leur service dans l'entreprise

On se donne un risque alpha à 5%.

La fonction *stats.chi2_contingency* de la librairie *scipy* permet d'effectuer le test du Khi-2 et d'obtenir la valeur du test, la p-value, les degrés de libertés, ainsi que les effectifs théoriques. Cette dernière information est difficilement interprétable à elle seule, il est préférable de se focaliser sur la p-value pour conclure le test d'indépendance.

Dans le cas des variables *depart* et *Service,* la p-value retournée est égale à 3.9e-14 (environ). La p-value étant largement inférieur à notre risque alpha, on peut rejeter notre hypothèse H0. À priori, il y a dépendance entre le départ des employés et le service auquel ils appartiennent dans l'entreprise.

Pour faire lien avec le test du Khi-2, il est possible de calculer le V de Cramer. On obtient alors la matrice suivante :

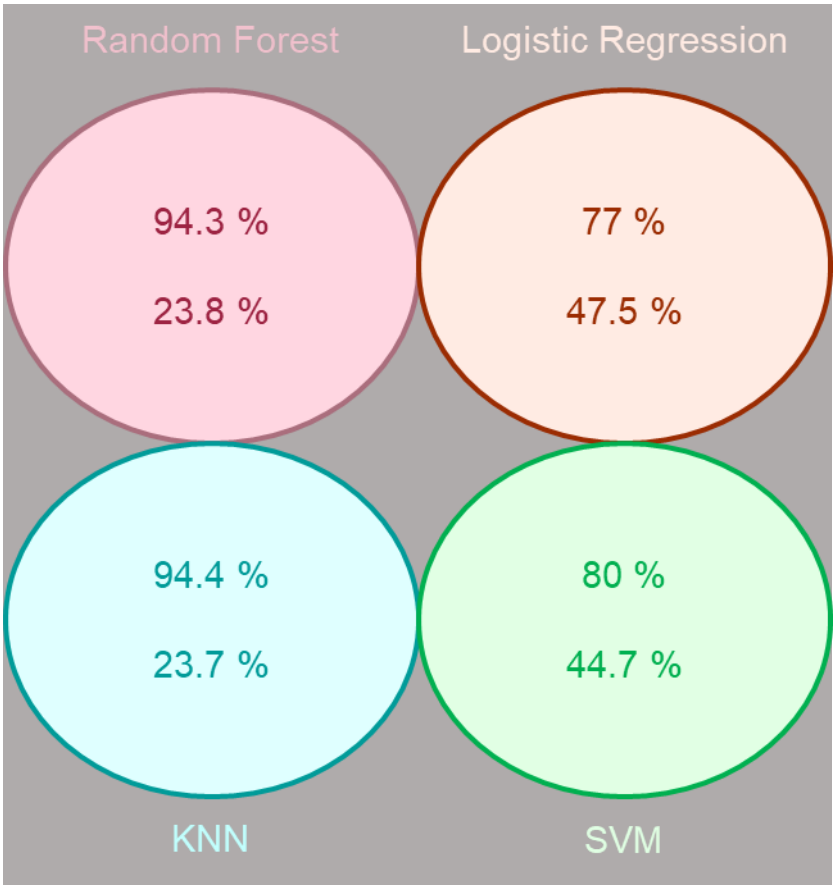


Les variables les plus fortements associées à la variable cible *depart* sont :

- *Satisfaction* (0.73 : association forte)
- *Nombre_de_projets* (0.60 : association relativement forte)
- *Nombre_heures_mensuelles_moyenne* (0.49 : association relativement forte)
- *derniere_evaluation* (0.41 : association relativement forte)
- *Temps_passe_dans_entreprise* (0.37 : association modérée)

À l'inverse, les variables les moins associées à la variable cible *depart* sont :

- *promotion_5_dernières_annees* (0.06 : association négligeable)
- *Service* (0.07 : association négligeable)
- *niveau_salaire* (0.16 : association faible)
- *accident_travail* (0.15 : association faible)

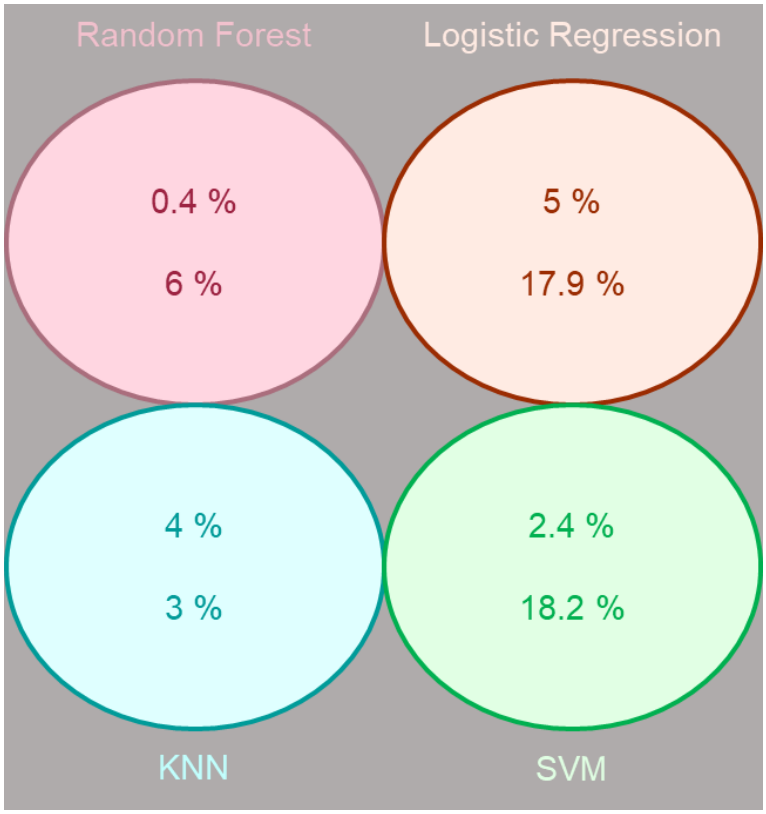


Le premier élément est le taux de fiabilité sur les données train. Le second élément correspond au RSME. On constate donc que le modèle KNN est le plus fiable. En effet, il faut que le RSME soit le plus proche de 0 pour éviter les erreurs de prédiction.

Satisfaction	derniere_eva	Nombre_de_	Nombre_heu	Temps_passé	Accident_du	promotion_5	Service	niveau_salai	random fore	logistic regre	knn	svm	expected
0.45	0.54	2	135	3	0	0	1	1	Left	Stay	Left	Stay	Left
0.11	0.81	6	305	4	0	0	1	1	Left	Left	Left	Left	Left
0.84	0.92	4	234	5	0	0	1	1	Stay	Stay	Left	Stay	Left
0.41	0.55	2	148	3	0	0	1	1	Left	Stay	Left	Stay	Left
0.36	0.56	2	137	3	0	0	1	1	Left	Stay	Left	Stay	Left
0.38	0.54	2	143	3	0	0	1	1	Left	Stay	Left	Stay	Left
0.45	0.47	2	160	3	0	0	1	1	Left	Stay	Left	Stay	Left
0.78	0.99	4	255	6	0	0	1	1	Stay	Stay	Stay	Stay	Left
0.45	0.51	2	160	3	1	1	1	1	Left	Stay	Left	Stay	Left
0.76	0.89	5	262	5	0	0	1	1	Left	Stay	Left	Stay	Left
0.44	0.51	2	156	3	0	0	9	3	Left	Stay	Left	Stay	Left
0.09	0.8	7	283	5	0	0	9	1	Left	Left	Left	Left	Left
0.92	0.87	4	226	6	1	0	9	2	Stay	Stay	Left	Stay	Left
0.74	0.91	4	232	5	0	0	9	2	Stay	Stay	Left	Stay	Left
0.09	0.82	6	249	4	0	0	9	2	Left	Left	Left	Left	Left
0.89	0.95	4	275	5	0	0	9	2	Stay	Stay	Left	Stay	Left
0.1	0.86	6	278	4	0	0	9	3	Left	Left	Left	Left	Left
0.81	1	4	253	5	0	0	9	1	Stay	Stay	Left	Stay	Left
0.11	0.8	6	282	4	0	0	9	2	Left	Left	Left	Left	Left
0.11	0.84	7	264	4	0	0	9	2	Left	Left	Left	Left	Left

On a créé un fichier externe avec quelques enregistrements de la base de données existantes qu'on lui a enlevé. On a supprimé la colonne cible et on a ajouté à ce fichier prédit les résultats des 4 modèles ainsi que le vrai résultat attendu : « *expected* ». On constate que le modèle KNN est bien le meilleur.

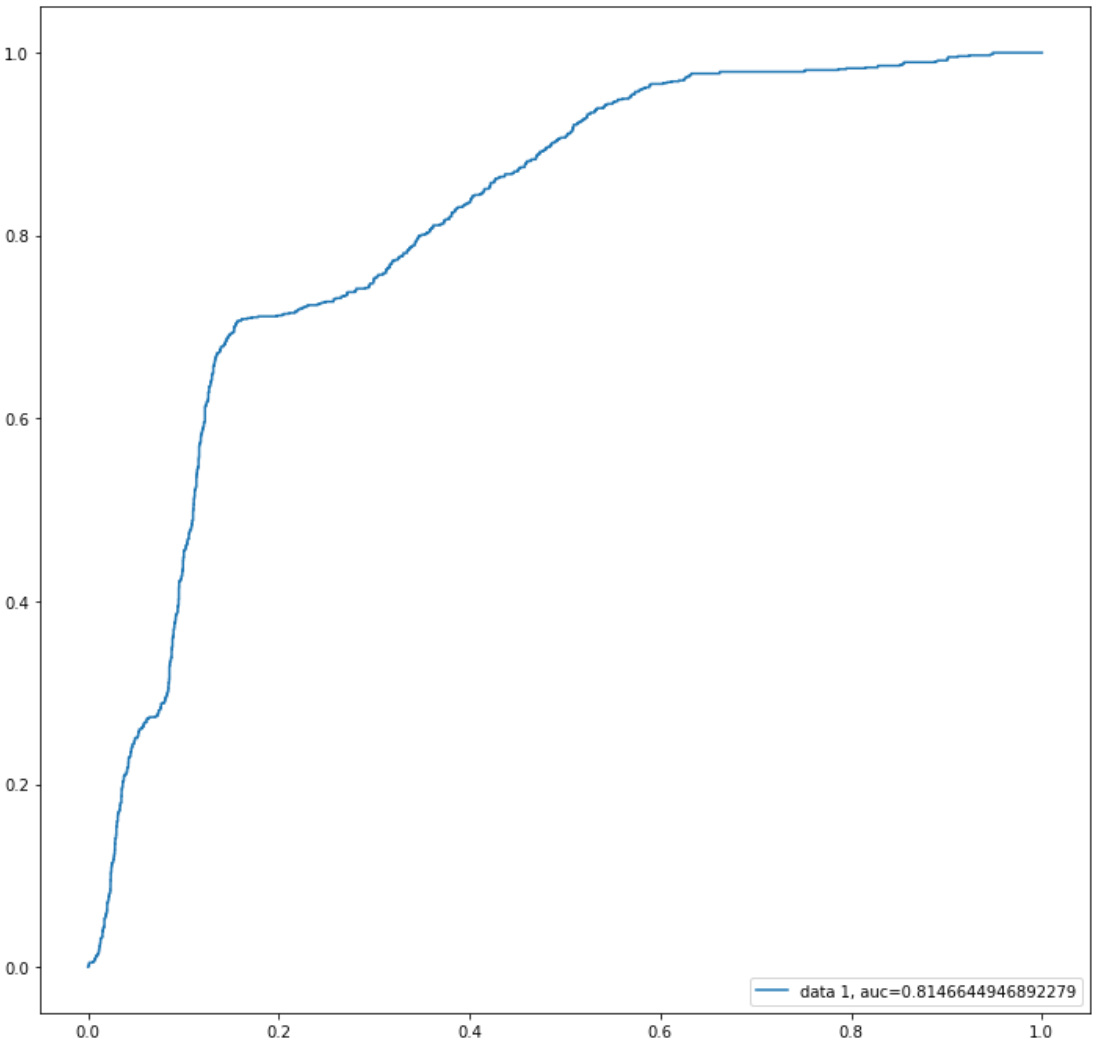
On peut également établir la matrice de confusion suivante qui permet de repérer les faux négatifs et positifs. Les résultats sont cohérents avec les précédents.



4. Critique des résultats

Les résultats semblent tous corrects, on observe simplement différents niveaux de fiabilité : la régression logistique semble être le modèle le moins efficace, tandis que la méthode Random Forest semble donner les meilleurs résultats. Mais pour s'assurer de la validité de chaque modèle, il est possible de détailler leurs résultats.

Dans le cadre de la régression logistique, la courbe de ROC est un autre indicateur possible :



La dispersion dans le modèle est relativement élevée (la qualité de la prédiction est assez variable). On confirme donc que la régression logistique n'est pas le modèle le plus adapté pour notre sujet.

Si on se pose du possible point de vue d'un service des ressources humaines, il faut savoir si l'on préfère une erreur où l'on prédit un départ quand il n'y en a pas, ou une erreur où l'on prédit une absence de départ quand il y en a une. L'entreprise préfère-t-elle engager des mesures (potentiellement financières) pour garder un employé qui, au final, n'a jamais eu l'intention de partir, ou bien préfère-t-elle passer à côté des employés qui souhaitent partir, et ainsi préserver ses ressources (autres que humaines). Les matrices de confusion fournissent une base d'aide à la décision sur cette question.

Si la priorité est mise pour économiser les ressources, les modèles du Random Forest, de la régression logistique et du SVM sont les plus adaptés, car ayant un plus faible nombre de faux positifs (départ estimé alors que ce n'est pas le cas) que de faux négatifs (pas de départ estimé alors que c'est le cas).

La méthode du Random Forest donne les meilleurs résultats, avec les plus faibles nombres de faux positifs et de faux négatifs parmi les trois modèles. On minimise le taux de fausses découvertes au détriment du taux de fausses omissions. On obtient :

- sensibilité = 94.6%
- spécificité = 98%
- précision = 99.5%
- taux de fausses découvertes = 0.5%
- taux de fausses omissions = 18.7%

Si, à l'inverse, la priorité est de conserver les employés à tout prix, le modèle KNN est le plus adapté, son nombre de faux positifs étant plus élevé que son nombre de faux négatifs. On minimise le taux de fausses omissions au détriment du taux de fausses découvertes. On obtient :

- sensibilité = 96.6%
- spécificité = 83.9%
- précision = 94.9%
- taux de fausses découvertes = 5.1%
- taux de fausses omissions = 11.1%

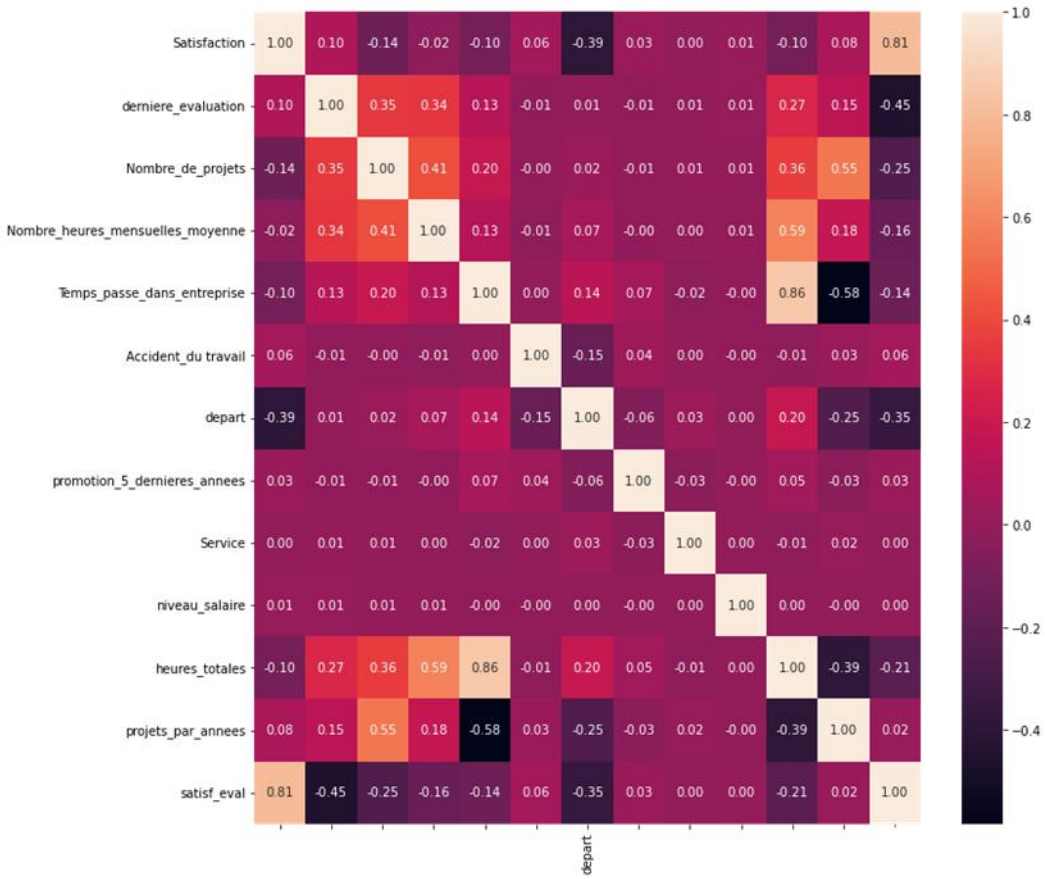
5. Perspectives d'amélioration

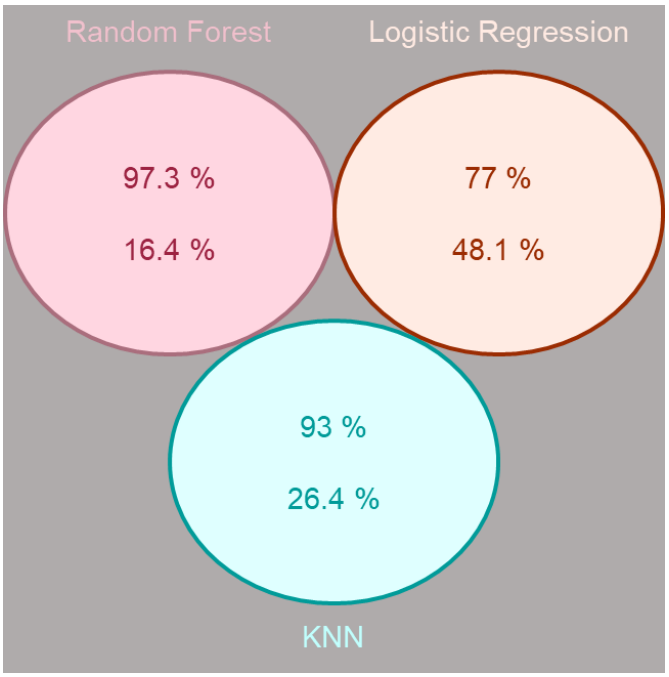
5.1. Mises en place

On a créé 3 nouvelles variables basées sur les features afin de voir si du feature engineering améliorerait nos résultats déjà très bons pour le modèle KNN mais également pour les autres.

- 'heures_totales' = 'Nombre_heures_mensuelles_moyenne' x 12 x 'Temps_passe_dans_entreprise'
- 'projets_par_annees' = 'Nombre de projets' / 'Temps_passe_dans_entreprise'
- 'satisf_eval' = 'Satisfaction' / 'derniere_evaluation'

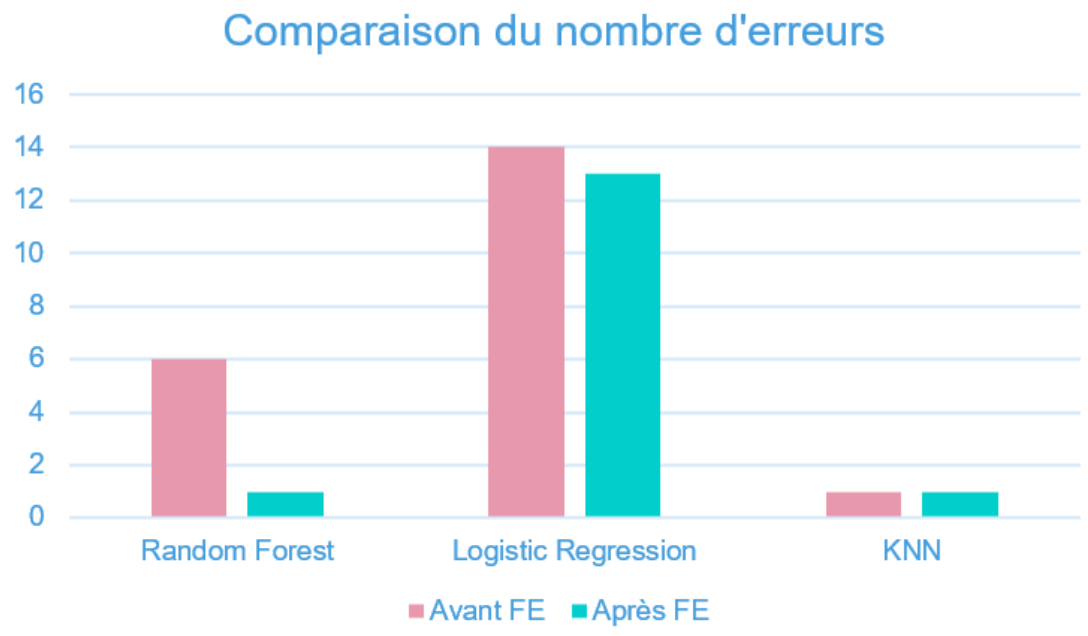
On constate d'ailleurs que ces 3 nouvelles variables sont fortement corrélées sur notre matrice de corrélation ce qui confirme notre choix dans ces variables.


Satisfaction 1.00 0.10 -0.14 -0.02 -0.10 0.06 -0.39 0.03 0.00 0.01 -0.10 0.08 0.81
derniere_evaluation 0.10 1.00 0.35 0.34 0.13 -0.01 0.01 -0.01 0.01 0.01 0.27 0.15 -0.45
Nombre_de_projets -0.14 0.35 1.00 0.41 0.20 -0.00 0.02 -0.01 0.01 0.01 0.36 0.55 -0.25
Nombre_heures_mensuelles_moyenne -0.02 0.34 0.41 1.00 0.13 -0.01 0.07 -0.00 0.00 0.01 0.59 0.18 -0.16
Temps_passe_dans_entreprise -0.10 0.13 0.20 0.13 1.00 0.00 0.14 0.07 -0.02 -0.00 0.86 -0.58 -0.14
Accident_du travail 0.06 -0.01 -0.00 -0.01 0.00 1.00 -0.15 0.04 0.00 -0.00 -0.01 0.03 0.06
depart -0.39 0.01 0.02 0.07 0.14 -0.15 1.00 -0.06 0.03 0.00 0.20 -0.25 -0.35
promotion_5_dernieres_annees 0.03 -0.01 -0.01 -0.00 0.07 0.04 -0.06 1.00 -0.03 -0.00 0.05 -0.03 0.03
Service 0.00 0.01 0.01 0.00 -0.02 0.00 0.03 -0.03 1.00 0.00 -0.01 0.02 0.00
niveau_salaire 0.01 0.01 0.01 0.01 -0.00 -0.00 0.00 -0.00 0.00 1.00 0.00 -0.00 0.00
heures_totales -0.10 0.27 0.36 0.59 0.86 -0.01 0.20 0.05 -0.01 0.00 1.00 -0.39 -0.21
projets_par_annees 0.08 0.15 0.55 0.18 -0.58 0.03 -0.25 -0.03 0.02 -0.00 -0.39 1.00 0.02
satisf_eval 0.81 -0.45 -0.25 -0.16 -0.14 0.06 -0.35 0.03 0.00 0.00 -0.21 0.02 1.00
depart
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4


Random Forest
Logistic Regression
97.3 %
16.4 %
77 %
48.1 %
93 %
26.4 %
KNN

Satisfaction	Accident_du	heures_total	projets_par_	satisf_eval	random fore	logistic regre	knn	expected
0.45	0	4860	0.66	0.83	Left	Stay	Left	Left
0.11	0	14640	1.5	0.13	Left	Left	Left	Left
0.84	0	14040	0.8	0.91	Left	Stay	Left	Left
0.41	0	5328	0.66	0.74	Left	Stay	Left	Left
0.36	0	4932	0.66	0.64	Left	Left	Left	Left
0.38	0	5148	0.66	0.7	Left	Left	Left	Left
0.45	0	5760	0.66	0.95	Left	Stay	Left	Left
0.78	0	18360	0.66	0.78	Left	Stay	Stay	Left
0.45	1	5760	0.66	0.88	Left	Stay	Left	Left
0.76	0	15720	1	0.85	Left	Stay	Left	Left
0.44	0	5616	0.66	0.86	Left	Stay	Left	Left
0.09	0	16980	1.4	0.11	Left	Left	Left	Left
0.92	1	16272	0.66	1.05	Stay	Stay	Left	Left
0.74	0	13920	0.8	0.81	Left	Stay	Left	Left
0.09	0	11952	1.5	0.11	Left	Left	Left	Left
0.89	0	16500	0.8	0.93	Left	Stay	Left	Left
0.1	0	13344	1.5	0.11	Left	Left	Left	Left
0.81	0	15180	0.8	0.81	Left	Stay	Left	Left
0.11	0	13536	1.5	0.13	Left	Left	Left	Left
0.11	0	12672	1.75	0.13	Left	Stay	Left	Left

On constate maintenant avec l'apprentissage sur un modèle avec seulement ces features donne de meilleurs résultats notamment pour le random forest qui fait beaucoup moins d'erreurs.



5.2. Envisagées

Pour aller plus loin dans l'analyse du lien entre les variables qualitatives, une autre solution est de réaliser une Analyse Factorielle des Correspondances (AFC).

Dans les modèles de prédictions, nous n'avons également pas eu l'occasion de tester un réseau de neurones. La librairie scikit-learn propose des solutions dans ce sens, avec les classes *MLPClassifier* et *MLPRegressor*.

Concernant la partie preprocessing, la variable Satisfaction peut être réduite en variable qualitative en créant des classes.

Dans le cadre de la régression logistique, il est possible de limiter les variables du modèle en se basant sur la méthode Recursive Feature Elimination (RFE).

6. Conclusion

On conclut donc que le meilleur modèle dans notre cas est le modèle KNN. En effet, malgré l'utilisation du feature engineering, on remarque que ce modèle reste le meilleur même s'il est intéressant de constater qu'il a amélioré les résultats de prédiction des autres modèles, notamment la random forest.