

Projet tutoré : JadBot

EHPAD AGIR

Rapport de projet tutoré de 5ème année
Septembre 2022 - Décembre 2022

Auteurs :

Imane AOUBIZA (Cheffe de projet)
Julia VILAS
Léandre GARRIGA

Maîtrise d'ouvrage : Piera CLÉMENT

Tutrice universitaire : Imen MEGDICHE

Remerciements

Nous tenons à remercier notre commanditaire, Mme Piera CLÉMENT pour sa bienveillance et pour nous avoir donné l'opportunité de réaliser ce projet dans cet établissement de santé d'envergure.

Nous remercions notre tutrice universitaire, Mme Imen MEGDICHE pour le suivi de nos travaux, ses conseils et encouragements.

Nous souhaitons remercier Rémi BASTIDE pour son suivi.

Introduction

Dans le cadre de notre dernière année de formation d'ingénieur, nous avons été amenés à réaliser un projet tuteuré. L'objectif de ce dernier en tant que maîtrise d'œuvre est de répondre aux besoins de la maîtrise d'ouvrage. Cela passe par l'analyse des besoins, avec une redéfinition du sujet si besoin puis la réalisation d'une solution satisfaisant ces besoins.

Cela fait maintenant quelques années qu'en France nous observons une augmentation des difficultés des EHPAD à prendre soin de leurs résidents. Manque de financement, manque de personnel, maltraitances, . . . Toutes ces problématiques entravent la vie des locataires et le bon fonctionnement des EHPAD.

Afin de pallier ce genre de soucis, l'EHPAD AGIR¹ et Mme. CLÉMENT ont mis en place fin 2021, une application de gestion des besoins des résidents. De plus, ce dispositif appelé Jadbot permet aux ASH de consulter une documentation sur la liste des tâches à effectuer auprès des résidents.

Table des matières

1	Présentation du sujet	1
2	Réalisations	2
2.1	Datasets	2
2.2	Dashboard	3
2.2.1	Mise en place des segments	3
2.2.2	Analyse des données	4
2.3	Modèles de prédiction	7
2.3.1	Théorie	7
2.3.2	Pratique	9
3	Gestion de projet	10
3.1	Scrum	10
3.2	Diagramme de Gantt	11
4	Conclusion	13
	Bibliographie	14
	Liste des figures	15
A	Snippets de code	16
B	Dashboard	17

Glossaire

1. AGIR : Association G rontologique Inter-R gionale
2. AS : Aide Soignante
3. ASH : agent de services hospitaliers
4. Classification : m thode de pr diction avec une cat gorie variable de sortie
5. Dashboard : repr sentation visuelle de donn es sous forme de graphiques, de compteurs
6. Dataset : jeu de donn es, peut se d couper en training (entra nement) et testing (test) sous-parties
7. DAX : langage d'analyse de donn es puissant pr sent sur Power BI
8. EHPAD :  tablissement d'H bergement pour les Personnes  g es D pendantes
9. ELASTIC NET : m thode de r gularisation
10. GIR : Groupe Iso-Ressources, grille d' valuations du niveau de d pendance d'une personne  g e
11. LASSO : m thode de r gularisation
12. Preprocessing : processus de pr paration des donn es avant de les soumettre au mod le d'apprentissage
13. R gularisation : ensemble de m thodes permettant d'optimiser l'apprentissage d'un mod le de machine learning
14. RIDGE : m thode de r gularisation
15. Surapprentissage : situation o  le mod le pr dictif s'adapte tr s fortement au dataset de training

Chapitre 1

Présentation du sujet

De nos jours, les EHPAD (Établissement d'Hébergement pour les Personnes âgées Dépendantes) ont de nombreuses difficultés à gérer l'accompagnement des résidents. Cela comprend l'hébergement comprenant la restauration et le bionettoyage, les soins et l'animation comme la vie sociale et les loisirs.

C'est dans cette optique que l'application JadBot intervient afin de faciliter le quotidien du personnel médico-social de ces établissements.

JadBot est un outil d'assistance vocale pour accompagner le travail des ASH³ et adapter l'organisation du travail en fonction de la réalité et des évolutions du terrain. L'application a été développée par l'entreprise castraise AgirTech.

Madame CLÉMENT s'est tournée vers nous dans le but de réaliser un modèle de classification⁴ des résidents de l'EHPAD AGIR¹, situé à Castres, en fonction des caractéristiques du résident et de son historique d'accompagnement. L'objectif est de définir un temps moyen de bionettoyage en fonction de la typologie de résident à partir du GIR pour répondre à ses besoins.

Dans ce projet, la maîtrise d'ouvrage est Piera CLMENT, la directrice de l'association AGIR¹. Le groupe de projet tuteuré, composé de Imane AOUBIZA (cheffe de projet), Léandre GARRIGA et de Julia VILAS, forme la maîtrise d'œuvre. Il est important de souligner que dans un projet, la maîtrise d'œuvre est chargée de répondre aux besoins de la maîtrise d'ouvrage qui est aussi qualifiée de client du projet.

Nous nous sommes mis d'accord pour que le rendu soit un dashboard⁵ avec des graphiques différents, faciles à comprendre et interactifs selon la classe sélectionnée.

Chapitre 2

Réalisations

Nous avons commencé à travailler lorsque nous avons reçu les données du GIR¹⁰ envoyées par Fabien FARRE le 14 octobre puis les données sur l'accompagnement des résidents par Piera CLÉMENT le 18 octobre. Nous avons réalisé un dashboard qui afin de répondre à l'objectif de ce projet et nous avons eu le temps de mettre en place quelques modèles prédictifs. La première étape a été de comprendre les données. Nous avons donc posé des questions à Piera CLÉMENT par mail et organisé un entretien. Nous avons ensuite traité les données sur Power BI pour qu'elles soient prêtes pour l'analyse. Nous avons déterminé à partir de quelles données les visuels que nous allions créer devaient être filtrés, puis nous avons créé les graphiques adéquats. Une fois le dashboard⁵ terminé, nous avons fait une démonstration de notre dashboard⁵ dynamique à Piera CLÉMENT qui nous a fait des retours positifs. Nous avons ensuite entrepris de créer un modèle permettant de prédire la durée de l'accompagnement du résident selon la typologie de ce dernier. Nous avons testé et comparé plusieurs méthodes dont RIDGE¹⁴, LASSO¹⁰ et ELASTIC NET⁹.

2.1 Datasets

Pour le projet, nous avons reçu de la part de l'EHPAD⁸ deux fichiers Excel.

Le premier fichier est un fichier de logs d'événements issu de l'application Jadbots. Il comprend l'ensemble des accompagnements effectués par les ASH³ de novembre 2021 à octobre 2022 avec quatre colonnes : le type de protocole, la chambre du résident ainsi que le début et la fin de l'accompagnement.

Il existe deux types de protocoles : les entretiens courants et les chambres approfondies dans le cadre de la formation e-learning. Les entretiens courants correspondent au bio-nettoyage de la chambre. Ils sont eux-mêmes divisés en trois catégories : les entretiens courants, les entretiens courants avancés et les entretiens courants expert +. L'ASH³ sélectionne sur l'application le niveau d'aide au protocole qu'elle souhaite avoir. Les niveaux avancés et experts + comprennent moins d'instructions que le niveau classique qui décrit étape par étape le protocole de bio-nettoyage. Le deuxième type de protocole est celui de chambre approfondie qui désigne le bio-nettoyage complet de la chambre suite au départ d'un résident (décès, hospitalisation ou vacances).

Le deuxième fichier contient la liste des patients pseudonymisés par un numéro avec leur chambre et leur GIR¹⁰. Le GIR est une donnée personnelle sensible puisqu'elle touche à la dépendance de la personne et donc à sa santé. Le GIR¹⁰ (Groupe Iso-Ressources) est une échelle de classification de résidents en fonction de leur autonomie s'étalant de 1 à 6. En effet, plus le GIR est important, plus le résident est autonome. Les résidents ont donc été pseudonymisés pour ne pas qu'ils puissent être identifiés.

Les fichiers recensent 11 592 événements et 73 résidents. Les données doivent être nettoyées (étape appelée preprocessing¹²) afin de ne garder que les données cohérentes et ainsi répondre aux exigences du client.

Tout d'abord, le format des chambres du fichier GIR et du fichier Accompagnement étaient différents. Le format du premier fichier indiquait l'étage contrairement au second. Nous avons donc modifié le fichier GIR pour extraire l'étage et le numéro de la chambre pour qu'il corresponde au numéro de chambre du fichier Accompagnement. Pour cela, nous avons utilisé les formules Excel présentées par l'annexe 1.

En regardant le fichier accompagnement, nous avons remarqué que beaucoup d'accompagnements ne duraient que quelques secondes, nous avons donc demandé à Piera CLÉMENT des explications. Elle nous

a indiqué qu'en moyenne les entretiens courants devaient durer 8 minutes environ et que les chambres approfondies devaient durer plus de 30 minutes.

Nous avons continué le traitement des données sur Power BI qui est le logiciel que nous avons choisi d'utiliser pour réaliser le dashboard⁵. Nous avons d'abord chargé les deux fichiers Excel en deux tables que nous avons renommées Accompagnement et GIR pour plus de clarté. Nous avons supprimé les lignes du fichier Accompagnement dont les chambres étaient D01, 120 ou 121. La chambre D01 correspond à une chambre double, or, dans l'autre fichier les chambres doubles ont été dédoublées pour correspondre aux deux résidents. D01 correspond donc à D01F (Fenêtre) et D01P (Porte). Les chambres 120 et 212 étaient des chambres de test de l'application.

Nous avons ensuite calculé la durée en minutes afin de filtrer la table des accompagnements pour ne garder que les accompagnements avec une durée cohérente. Nous avons décidé de supprimer tous les accompagnements inférieurs à 1 minute. En effet, même si le résident est autonome, il paraît incohérent d'avoir des accompagnements aussi courts.

Effectivement, au début de déploiement de l'application, il y a eu des bugs sur certains protocoles.

Selon Piera CLÉMENT, les accompagnements peuvent durer moins de quatre minutes si le résident est autonome, c'est-à-dire, avec un GIR de 5 ou 6, et s'il n'est pas dans une chambre de l'unité protégée, soit pas dans une des chambres 1 à 10. Les accompagnements suivant le protocole de chambre approfondie doivent durer plus de 30 minutes.

Respecter ces conditions revient à créer une table calculée qui filtre la table originelle. Pour filtrer la table Accompagnement avec une condition sur le GIR qui est dans une autre table, nous avons dû lier les deux tables grâce à la vue Modèle. Cette vue permet de voir les différentes tables et leurs liens. Les deux tables ont en commun la colonne Chambre. Plusieurs accompagnements peuvent se faire dans la même chambre et un résident occupe une unique chambre. Les deux tables GIR et Accompagnement sont donc liées par une relation 1 à plusieurs. Nous avons écrit la formule suivante, en DAX⁷, qui est le langage de Power BI.

```
NEW Prises en charge = FILTER('Prises en charge',
('Prises en charge'[Durée en Minutes]>=4 && 'Prises en charge'[Durée en Minutes]<=30) && 'Prises en charge'[Protocole]<>"Chambre Approfondie" ||
('Prises en charge'[Durée en Minutes]>=1 &&
'Prises en charge'[Protocole]<>"Chambre Approfondie" &&
'Prises en charge'[Durée en Minutes]<4 && RELATED(GIR[GIR])>=5 &&
NOT('Prises en charge'[Chambre] IN {"1","2","3","4","5","6","7","8","9","10"})) ||
('Prises en charge'[Durée en Minutes]>30 && 'Prises en charge'[Protocole]=="Chambre Approfondie"))
```

FIGURE 2.1 – Formule en DAX pour filtrer les accompagnements

La nouvelle table des accompagnements respecte maintenant les conditions mais ne contient que 301 lignes seulement sur les 11 592 initiales. Il devait y avoir beaucoup d'erreurs lors de l'utilisation de l'application. En effet, cela reste cohérent car certaines ASH³ ont eu du mal à s'habituer à l'utilisation de l'application au début de la mise en place. En effet, la phase de test était la période où il y a eu des expérimentations, des évolutions de paramétrage ainsi que des difficultés techniques.

Une fois la table des accompagnements créée, nous sommes passés à la création du dashboard⁵.

2.2 Dashboard

Un dashboard⁵ est une page composée de plusieurs graphiques et indicateurs. Il vise à offrir une visualisation du jeu de données. Pour le construire, nous avons utilisé les tables GIR et NEW Accompagnement. Nous devons lier les tables pour utiliser les variables des deux tables sur un même graphique.

2.2.1 Mise en place des segments

D'abord, nous avons jugé que Piera CLEMENT devait pouvoir filtrer les graphiques selon le GIR et le type de protocole. Pour cela, nous avons créé ce qu'on appelle des segments dans Power BI. Les segments permettent de cibler une partie des données observées. Comme le travail des ASH³ est organisé par mois, nous avons ajouté un segment pour les mois et les années.

FIGURE 2.2 – Pluralité de filtres pour le dashboard dynamique

Nous avons aussi ajouté un segment de date relative. Ce segment prend en compte une durée par rapport à la date du jour de consultation du dashboard. Par exemple, nous pouvons choisir d’afficher les données des trois dernières semaines, ou des deux derniers mois. Ce segment n’est pas très utile pour l’instant car les données ne sont pas en temps réel mais sera très utile si le dashboard est intégré à l’application. Ce segment est différent des autres car il ne correspond pas à la liste des valeurs possibles mais est créé par Power BI à partir d’une colonne de dates.

FIGURE 2.3 – Segment de date relative

Il est composé de trois paramètres. Le premier paramètre permet de choisir entre ‘Dernier’, ‘Ce’ et ‘Suivant’. Le deuxième permet d’entrer un nombre pour définir la plage de dates relatives. Le dernier paramètre permet de choisir la mesure de la date. Les possibilités sont les suivantes : ‘Jours’, ‘Semaines’, ‘Semaines (calendaires)’, ‘Mois’, ‘Mois (calendaires)’, ‘Ans’ et ‘Années (calendaires)’. La différence entre une mesure et sa version calendaire est la relativité de la date. Par exemple, si la date du jour est le 21 novembre, en choisissant ‘Mois’ et en entrant ‘2’ au deuxième paramètre, le segment filtrera les données pour ne garder que celles du 22 septembre au 21 novembre. En revanche, en choisissant ‘Mois (calendaires)’, les visuels affichent les données du 1er septembre au 31 octobre. C’est la même chose pour les années et les semaines. Les semaines sont basées sur le calendrier américain automatiquement sur Power BI, bien que les paramètres régionaux soient réglés pour la France. Elles commencent donc le dimanche et finissent le samedi.

A ce stade, nous avons donc sur notre dashboard cinq segments.

2.2.2 Analyse des données

Après avoir créé les segments pour filtrer les visuels, nous avons créé les dits visuels. Piera CLÉMENT souhaite voir la durée moyenne d’un protocole. Nous avons donc créé une carte afin d’afficher le nombre directement.

8,78
Minutes en Moyenne

FIGURE 2.4 – Visuel de durée moyenne d’un protocole

Pour l’ensemble des données, c’est-à-dire, en ne sélectionnant aucun filtre, un protocole dure en moyenne 8,78 minutes ce qui correspond à ce que nous avait précisé Piera CLÉMENT. Ce nombre peut varier selon les segments choisis. En cliquant sur le 6 du segment des GIR, la durée passe à 3,75 minutes en moyenne.



FIGURE 2.5 – Durée moyenne d'un protocole en fonction du GIR 6

Ce résultat est logique car les résidents de GIR 6 n'ont aucun problème dans la réalisation des actes de la vie courante. Ils demandent sûrement aux ASH³ de ne pas faire une partie du travail puisqu'ils peuvent le faire eux-mêmes.

Nous avons ensuite tracé des graphiques pour aider Piera CLÉMENT dans sa prise de décisions. Nous avons commencé par tracer le nombre de résidents par GIR. Ce visuel ne change pas selon les filtres, il sert juste à donner une vision globale de la typologie des résidents.

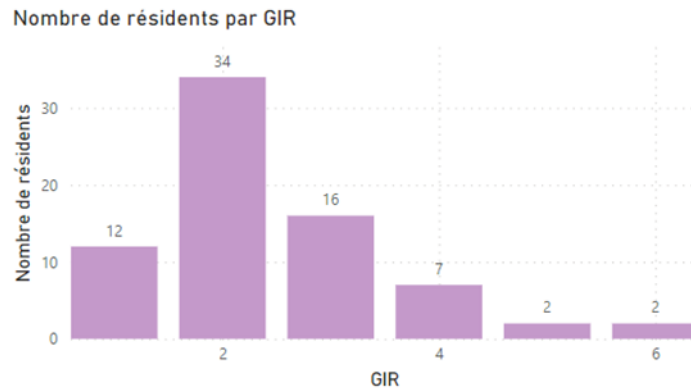


FIGURE 2.6 – Répartition des résidents en fonction de leur GIR

Ce graphique permet d'avoir une vue d'ensemble de la typologie des résidents. Nous pouvons voir qu'il y a peu de résidents autonomes ayant un GIR de 5 ou 6. De manière générale, les ASH³ vont donc passer beaucoup de temps pour s'occuper des chambres puisqu'il y a 69 résidents qui ont un GIR entre 1 et 4. Les résidents de GIR 1 et 2 ne sont pas autonomes. Les résidents de GIR 3 sont autonomes mentalement mais ont besoin d'aide pour les soins corporels. Les résidents de GIR 4 sont autonomes mais ont des difficultés pour certaines tâches de la vie courante. Le niveau d'autonomie peut varier même au sein d'un même GIR. Deux personnes de même GIR peuvent avoir besoin de différentes aides. Nous avons ensuite tracé le nombre de protocoles par mois et par type de protocole.

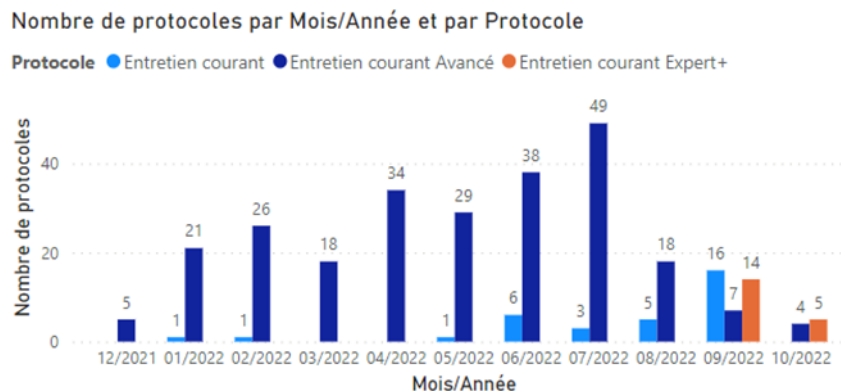


FIGURE 2.7 – Répartition des protocoles en fonction du mois et du type de protocole

La plupart des entretiens choisis par les ASH³ sont les entretiens courants avancés. Ils choisissent le niveau intermédiaire de description du protocole. Ce n'est qu'à partir de septembre 2022 que certains ASH³ utilisent l'option d'entretien courant expert+ sur l'application. Ils doivent s'être habitués et ne nécessitent plus autant d'indications qu'avant avec le protocole d'entretien courant avancé. En traçant la durée moyenne

par GIR et par protocole, nous pouvons déterminer la typologie des résidents qui demandent le plus de temps.

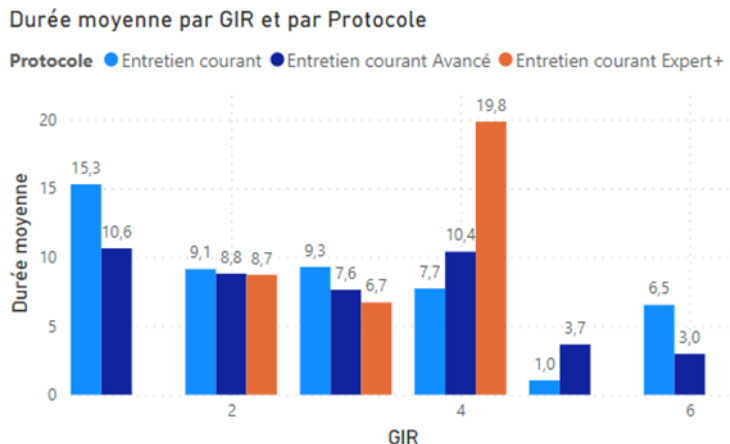


FIGURE 2.8 – Durée moyenne des protocoles en fonction du GIR et du type de protocole

Tout d’abord, on remarque un pic anormal pour les protocoles d’entretiens courants Expert + pour les résidents de GIR 4. A part cette valeur anormale, le reste des données semble cohérent. Les résidents de GIR 1 sont ceux qui requièrent le plus de temps et les résidents de GIR 5 et 6 le moins. Piera CLEMENT nous avait expliqué qu’il n’y avait pas beaucoup de différence entre les GIR 2, 3 et 4 ce qui explique les résultats obtenus.

Nous nous sommes intéressés à la durée totale des protocoles par rapport aux résidents individuellement par le biais de leur numéro de chambre. Le GIR reste visible, il suffit de passer la souris sur la donnée pour le lire.

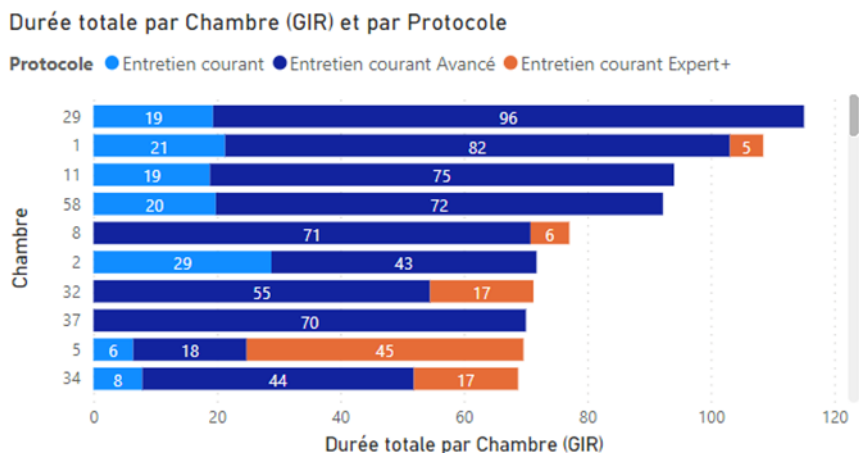


FIGURE 2.9 – Répartition de la durée passée par chambre en fonction du type de protocole

Ce graphique permet d’observer les chambres qui prennent beaucoup de temps et aidera donc Piera CLÉMENT à organiser le travail des ASH³ par chambre.

Une fois tous les visuels tracés, il faut modifier les interactions avec les segments. Tous les segments ne doivent pas s’appliquer à tous les graphiques. Pour le premier graphique présentant le nombre de résidents par GIR, le segment du GIR ne doit pas s’appliquer. Tous ces graphiques et ces segments ont été rassemblés pour produire le dashboard suivant.

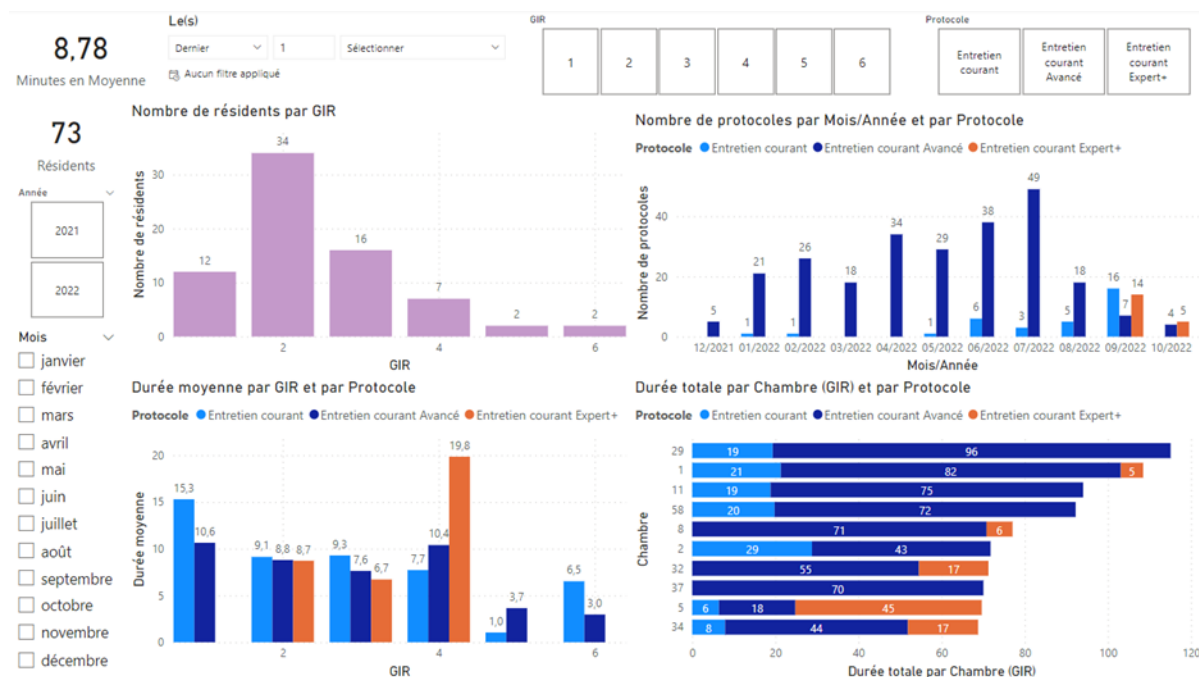


FIGURE 2.10 – Dashboard

2.3 Modèles de prédiction

2.3.1 Théorie

La deuxième tâche à réaliser est l'élaboration d'un modèle de prédiction du temps d'accompagnement suivant différentes variables. Nous avons exporté la table créée sur Power BI pour avoir un dataset⁶ sur Python avec la durée en minutes et les variables indépendantes. Nous avons pensé à faire une régression linéaire multiple. Après avoir réfléchi avec notre tutrice pédagogique, nous avons choisi d'appliquer les méthodes RIDGE¹⁴, LASSO¹⁰, ELASTIC NET⁹ qui sont des méthodes de régression linéaire multiple régularisées. Nous avons décidé d'appliquer ces trois méthodes afin de les comparer et garder la méthode avec les meilleures métriques.

Tout d'abord, afin d'obtenir un modèle de régression linéaire correct, il faut s'assurer de réaliser les étapes suivantes :

1. Construire une fonction de coût : fonction mathématique qui mesure l'erreur commise en faisant une approximation des données
2. Minimiser cette fonction de coût : trouver les meilleurs paramètres possibles pour minimiser l'erreur de modélisation
3. Choisir une méthode de résolution du problème : ici, nous avons choisi la méthode analytique nommée méthode des moindres carrés

Ces méthodes sont des techniques de régularisation¹³, c'est-à-dire que l'on va chercher à simplifier un modèle en pénalisant les valeurs extrêmes ou en cherchant les paramètres les plus explicatifs. La régularisation est un principe fondamental en apprentissage automatique consistant à réduire le sur-apprentissage. Ce dernier correspond à éliminer le phénomène où le modèle a trop appris les particularités de chacun des exemples fournis en exemple. Il présente alors un taux de succès très important sur les données d'entraînement (pouvant atteindre jusqu'à 100%), au détriment de ses performances générales réelles.

Dans notre situation, la meilleure méthode de régularisation correspondrait à celle présentant les meilleures métriques comme le R^2 et l'accuracy.

LASSO

LASSO¹⁰ (Least Absolute Shrinkage and Selection Operator) est une méthode d'analyse de régression qui réalise à la fois une sélection de variables et une régularisation.

Le point fort de cet algorithme est la minimisation des valeurs des données vers un point central.

LASSO met en place la régularisation de type L1, c'est-à-dire qu'on ajoute une pénalité égale à la valeur absolue de la magnitude des coefficients. Cela permet d'obtenir des modèles clairsemés (sparse) avec peu de coefficients. De plus, la norme L1 correspond à la norme de Manhattan, qui est une distance correspondant à un déplacement à angle droit sur un damier.

Réaliser la régression avec LASSO revient à réussir à minimiser le problème quadratique suivant :

$$C(h) = \frac{1}{2p} \sum_{i=1}^p (h(x_i) - y_i)^2 + \lambda \sum_{j=1}^n |\theta_j|$$

FIGURE 2.11 – Formule de coût de LASSO

Somme des résidus quadratiques (RSS) + λ * (somme de la valeur absolue de la magnitude des coefficients)

où :

- λ dénote la quantité de rétrécissement.
- $\lambda = 0$ implique que toutes les caractéristiques sont considérées et c'est équivalent à la régression linéaire où seule la somme résiduelle des carrés est considérée pour construire un modèle prédictif.
- λ qui tend vers l'infini implique qu'aucune caractéristique n'est considérée, c'est-à-dire que lorsque λ se rapproche de l'infini, il élimine de plus en plus de caractéristiques.
- Le biais augmente avec l'augmentation de λ
- la variance augmente avec la diminution de λ

RIDGE

RIDGE est une méthode de régularisation¹³ pénalisante, utilisée pour limiter l'instabilité des prédictions liées à des variables explicatives trop corrélées entre elles. Cette méthode utilise la régularisation de type L2, ce qui correspond à la distance euclidienne. La fonction de coût selon RIDGE est la suivante :

$$C(h) = \frac{1}{2p} \sum_{i=1}^p (h(x_i) - y_i)^2 + \lambda \sum_{j=1}^n \theta_j^2$$

FIGURE 2.12 – Formule de coût de RIDGE

Avec la pénalisation RIDGE, on va diminuer la distance entre les solutions possibles en se basant sur la mesure euclidienne.

Le paramètre important de pénalisation est λ .

De manière générale, en augmentant λ , on augmente le biais de la solution tout en diminuant la variance.

- Quand λ est proche de 0, on s'approche de la solution classique où il n'y a pas de pénalité.
- Quand λ est infini, la pénalisation agit de sorte à ce que tous les paramètres soient tous nuls.

ELASTIC NET

La régularisation ELASTIC NET est une combinaison des deux régularisation précédentes (LASSO et RIDGE). Elle a pour objectif de réaliser une régularisation en évitant une sélection trop forte comme pourrait le faire Lasso ce qui permettrait de conserver des variables qui pourraient être fortement corrélées.

$$C(h) = \frac{1}{2p} \sum_{i=1}^p (h(x_i) - y_i)^2 + \lambda \sum_{j=1}^n \left[\frac{1}{2} (1 - \alpha) \theta_j^2 + \alpha |\theta_j| \right]$$

FIGURE 2.13 – Formule de coût d'ELASTIC NET

Ici, le paramètre Alpha est un paramètre permettant de moduler l'équilibre entre RIDGE et LASSO.

- Si $\alpha = 1$, alors la fonction de coût est celle correspondant à celle de LASSO.
- Si $\alpha = 0$, on retombe sur la régression RIDGE.

2.3.2 Pratique

Avant de commencer à expliquer le choix de nos variables, il nous faut tout d'abord éclaircir ce que l'on souhaite prédire. Afin de gérer au mieux les emplois du temps des ASH³ nous voudrions connaître le temps à allouer à chaque intervention. De ce fait, ce qu'il nous faut prédire, c'est la durée en minutes.

Pour réaliser cela nous devons prendre en compte certains paramètres :

- le GIR, qui nous permettra de généraliser sur ces groupes le temps à allouer.
- le protocole, qui se différencie quant à la longueur de sa liste des tâches et par conséquent, la durée d'intervention.

Nous avons également plusieurs fois modifier la valeur de la variable alpha afin d'essayer de produire de meilleurs résultats.

Le processus d'entraînement de notre modèle a été sensiblement le même pour les trois méthodes de prédiction. Une fois le fichier de données chargé sur python nous les avons tout simplement transformés à l'aide de ce que l'on nomme "Label Encoder" afin qu'elles passent de leurs valeurs initiales à des valeurs numériques afin de les catégoriser. Une fois ces transformations faites, nous sélectionnons les variables de prédiction (rapporté ci-dessus). Nous séparons notre set de données en deux parties, une première pour les données d'apprentissage de notre modèle représentant 80% du total et une seconde pour les données de test représentant les derniers 20%. Suite à ça, nous choisissons le modèle de prédiction que nous souhaitons utiliser, dans notre cas ce fut RIDGE, LASSO et ELASTIC NET⁹ et nous commençons l'entraînement de notre modèle.

L'entraînement de nos modèles de prédiction terminé, nous vérifions leurs résultats via les notions de R^2 et de MAE moyen. Ces informations nous permettent d'évaluer les performances.

- R^2 : Coefficient de détermination, est une mesure de la qualité de la prédiction. Elle se situe sur une intervalle de 0 à 1 et plus la valeur est proche de 1 plus notre modèle est performant.
- MAE : Mean Absolute Error, mesure l'erreur entre deux observations exprimant le même événement.

Lasso	Ridge	ElasticNet
score R^2 : 0.02 Mean MAE: 3.803	score R^2 : 0.01 Mean MAE: 3.569	score R^2 : 0.02 Mean MAE: 3.793

FIGURE 2.14 – Performances des trois différents modèles de régularisation

Comme nous pouvons l'observer nos scores sont loin d'être probants et nous incite à penser qu'aucune de ces 3 méthodes ne peut correspondre à nos besoins.

Nous aurions eu de meilleurs résultats si nous avions plus de variables à exploiter sur la typologie des résidents. Le GIR reste un indicateur très vague sur le niveau d'autonomie d'une personne et donc il est difficile d'en tirer un modèle prédictif de la durée d'un protocole.

Chapitre 3

Gestion de projet

En tant que cheffe de projet, Imane Aoubiza a décidé d'utiliser le cadre Scrum vu plus tôt au cours de l'année. Scrum nous a été présenté en cours d'Architecture Devops / Interopérabilité FHIR par les alumni Ludovic Yol et Alexandre Receveur. Ce framework a été utilisé pour le développement logiciel dans le cadre du cours mais nous avons décidé de l'utiliser pour notre projet.

Cependant, nous n'avons pas pu réellement commencer à travailler le projet en même temps que les autres groupes par souci d'indisponibilité de la part de notre client.

Nous nous sommes rendus à l'EHPAD⁸ le jeudi 20 octobre pour obtenir une meilleure description du périmètre du projet.

Par ailleurs, des réunions de suivi de projet ont été régulièrement organisées avec la maîtrise d'ouvrage et notre tutrice pédagogique.

Afin de gérer l'avancement du projet, nous avons fait le choix de suivre la méthode SCRUM, à l'aide du logiciel JIRA.

3.1 Scrum

Scrum est une des méthodes de gestion de projet agile. C'est un cadre à l'intérieur duquel sont développés des produits complexes dans des environnements complexes. Scrum suit les valeurs agiles telles que la collaboration avec le client et l'acceptation du changement.

Scrum oblige à définir des rôles aux personnes de l'équipe projet appelée Scrum Team. On peut être Product Owner, Scrum Master ou simplement faire partie de la Development Team. Comme nous ne sommes que trois dans notre groupe, il a été décidé que tout le monde ferait partie de la Development Team qui, comme son nom l'indique, est chargée de développer le produit demandé par la maîtrise d'ouvrage. Nous avons décidé que Léandre serait le Product Owner et Julia le Scrum Master.

Le Product Owner a pour rôle de garantir que le projet réponde aux attentes du client et des usagers, il interprète les besoins des clients et les transforme en outils fonctionnels. Il représente l'interface entre le client et l'équipe scrum. De plus, il s'occupe de rédiger les Users Stories et d'alimenter le Product Backlog expliqué ci-après.

Le Scrum Master organise les cérémonies Agile, planifie les Sprints et les différents meetings tel que le Daily Meeting et les Sprint Planning. Il veille à ce que la méthode Scrum soit bien appliquée en accompagnant son équipe et en impliquant ses membres au maximum.

L'équipe de développement est composée d'experts dans leurs domaines qui fournissent un incrément terminé et potentiellement publiable à la fin de chaque Sprint. Les compétences spécifiques dont les développeurs ont besoin sont souvent vastes et varient en fonction du domaine de travail. Cependant, les développeurs sont toujours responsables de créer un plan pour le Sprint et le Sprint Backlog, d'adapter leur plan en fonction des objectifs du Sprint et d'adhérer à la Définition of Done.

Une User Story est une description relativement simple d'une fonctionnalité qui doit être développée, elle se décompose généralement en trois parties sous la forme suivante :

En tant que **<qui>**, je veux **<quoi>**, afin de **<pourquoi>**.

— “Qui” représente l'utilisateur de la fonctionnalité, le point de vue dans lequel nous souhaitons nous placer afin de comprendre ses besoins.

- “Quoi” décrit la fonction qui doit être réalisée et développée de manière relativement simple.
- “Pourquoi” permet de reconnaître l’intérêt de la fonctionnalité et ainsi de justifier son développement.

Dans le cadre de ce projet, où nous ne réalisons pas réellement de développement, nous avons adapté cette technique à nos besoins. Nous avons, pour chaque graphique de notre dashboard réalisé une User Story.

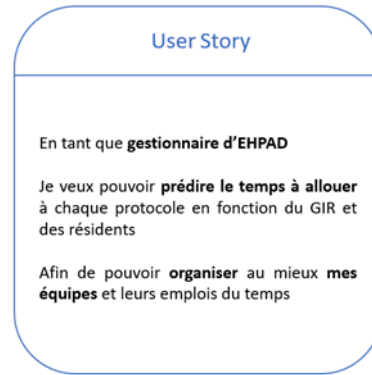


FIGURE 3.1 – User story directrice du projet

Les “sprints” sont des sessions intenses de travail où l’équipe de développement se doit d’implémenter le plus de fonctions possibles. Ces sessions durent de manière générale entre 1 à 4 semaines. Dans le cadre de ce projet nous avons décidé de limiter une séance de sprint à 1 semaine, plus cohérente avec le temps total qui était alloué au projet dans notre cursus.

Les implémentations (User stories) qui doivent être développées pendant le sprint sont disposées dans le sprint backlog. Cette liste est organisée afin d’avoir les fonctions les plus simples à intégrer sur le haut de la pile. Il suffit aux développeurs de venir chercher le travail à développer dans cette liste pendant les sprints. Finalement cette pile alimente sont contenu via le product backlog recensant toutes les fonctionnalités à intégrer.

Afin de connaître la difficulté d’une tâche à développer, il est coutume au début d’un projet de noter celles-ci en se basant sur différents moyens. Nous nous sommes positionnés sur la méthode de poker planning. Chaque membre de l’équipe estime la difficulté de la tâche et dans le cas où les estimations sont différentes, nous en discutons jusqu’à tomber d’accord sur une estimation.

- Les cérémonies sont des réunions qui ont lieu pendant toute la durée d’un sprint. De sa préparation avec le sprint planning à sa fin avec la revue de sprint et la rétrospective et avec des daily meetings pendant le sprint.
- Le sprint planning se déroule au début du sprint. Après l’analyse du product backlog tous les membres échangent et se mettent d’accord sur les fonctionnalités et les outils à développer pendant le sprint.
- Le daily meeting a lieu tous les jours pendant un sprint. C’est une réunion très courte où chaque participant communique sur l’avancée de son travail.
- La revue de sprint prend place à la fin d’un sprint. On y présente, à chaque partie prenantes du projet les fonctions qui ont pu être réalisées pendant le sprint à la manière d’une démonstration.
- La rétrospective est la réunion qui va terminer le sprint. Nous y réalisons un bilan de sprint, ce qui a fonctionné, ce qui n’a pas fonctionné. Et nous faisons des propositions d’amélioration pour les prochains sprints.

3.2 Diagramme de Gantt

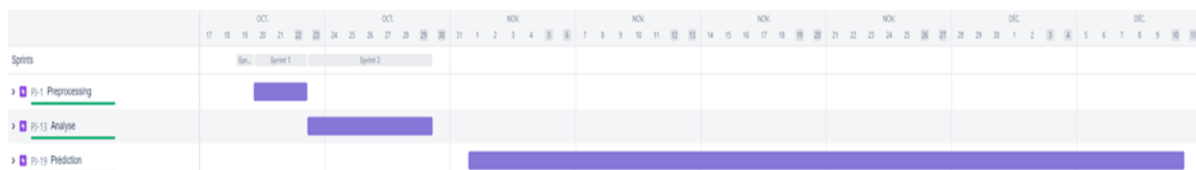


FIGURE 3.2 – Diagramme de Gantt

D'après ce diagramme de Gantt, obtenu avec le logiciel Jira, on peut voir que notre projet s'étend de la mi-octobre à la mi décembre, soit donc 3 mois de travail sur le sujet.

On peut y retrouver 3 grandes étapes que sont le preprocessing, l'analyse et la prédiction. Avec l'analyse, on retrouve le processus de nettoyage du jeu de données, puis on retrouve la mise en place du dashboard avec et enfin l'élaboration d'algorithmes de régression linéaire avec la prédiction.

Chapitre 4

Conclusion

A travers ce projet, nous avons été amené à réaliser un dashboard afin d'offrir une visualisation de l'organisation du bionettoyage au sein de l'établissement d'hébergement pour personnes âgées.

Puis nous avons utilisé plusieurs algorithmes de régression linéaire sur le jeu de données nettoyé selon plusieurs conditions fournies par Madame CLÉMENT afin de prédire le temps à allouer par protocole en fonction du GIR.

Une piste intéressante et pratique pour les EHPAD^s serait d'intégrer le dashboard dans le logiciel même qu'est JadBot sous forme de module.

Bibliographie

- [1] [Régularisation : Ridge, Lasso et Elastic Net | Zeste de Savoir](#)
- [2] [Regularization in Machine Learning | Simplilearn](#)
- [3] [A Complete understanding of LASSO Regression](#)
- [4] [Ridge and Lasso Regression: L1 and L2 Regularization | by Saptashwa Bhattacharyya | Towards Data Science](#)
- [5] [A Complete understanding of LASSO Regression](#)
- [6] [Algorithme N°6 –{} Et si on faisait du LASSO ? ...{} pour comprendre les techniques de régularisation linéaire. - Devoteam France](#)
- [7] [Lasso Regression with Python | Jan Kirenz](#)

Table des figures

2.1	Formule en DAX pour filtrer les accompagnements	3
2.2	Pluralité de filtres pour le dashboard dynamique	4
2.3	Segment de date relative	4
2.4	Visuel de durée moyenne d'un protocole	4
2.5	Durée moyenne d'un protocole en fonction du GIR 6	5
2.6	Répartition des résidents en fonction de leur GIR	5
2.7	Répartition des protocoles en fonction du mois et du type de protocole	5
2.8	Durée moyenne des protocoles en fonction du GIR et du type de protocole	6
2.9	Répartition de la durée passée par chambre en fonction du type de protocole	6
2.10	Dashboard	7
2.11	Formule de coût de LASSO	8
2.12	Formule de coût de RIDGE	8
2.13	Formule de coût d'ELASTIC NET	9
2.14	Performances des trois différents modèles de régularisation	9
3.1	User story directrice du projet	11
3.2	Diagramme de Gantt	11
B.1	Dashboard	17

Annexe A

Snippets de code

```
Etage : =GAUCHE(B2 ; CHERCHE( " " ;B2 ;1))  
Chambre : =DROITE(B2 ;NBCAR(B2)-CHERCHE(" " ;B2 ;1)) =STXT(B2 ;CHERCHE(" " ;B2 ;1)+1 ;CHERCHE(" " ;B2 ;CHERCHE(" " ;B2 ;1)+1)-CHERCHE(" " ;B2 ;1)) =SIERREUR(F2 ; E2)
```

Annexe B

Dashboard

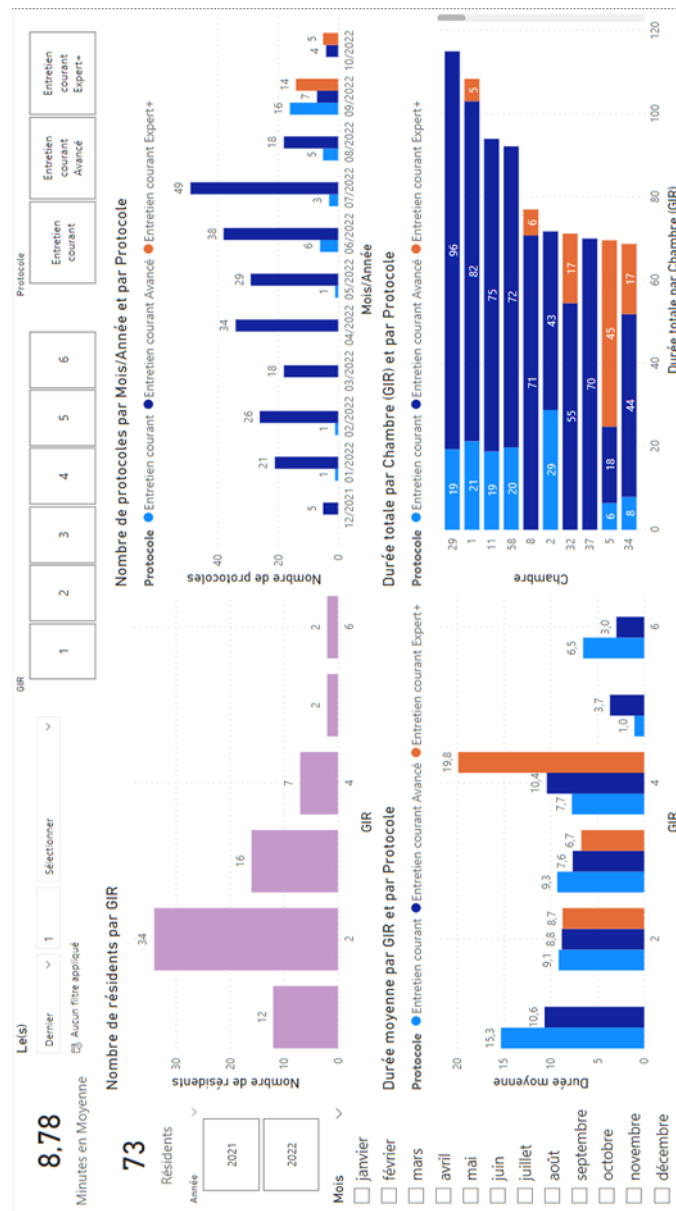


FIGURE B.1 – Dashboard