Stimul

CentraleSupélec

Projet Innovation S8 - 2019

Projet Stimul x Latitudes

Client
Etienne Dormeuil

Authors
Hugo Cohen
Ariane Dalens
Jianxiang Gao
Guillaume Kunsch
Félix Motte

28 mai 2019

Table des matières

1 Présentation du projet.

1.1 Présentation de Stimul.

Stimul est une start-up d'accompagnement numérique de patients créée en 2016.

Elle est née suite à la constatation du peu de suivi médical en dehors des institutions de santé ; or pour empêcher le développement d'une pathologie, agir sur le mode de vie (l'alimentation, l'activité sportive, la connaissance des risques et plus généralement une culture médicale) est une méthode de bon sens, qui a de plus été appuyée par plusieurs études.

En pratique l'accompagnement réalisé par Stimul prend la forme suivante. Chaque participant s'engage dans le programme au sein d'une cohorte d'une dizaine de personnes atteintes, ou bien susceptible, de développer la même pathologie. Chaque participant reçoit 2 outils : une montre connectée qui permet de compter les pas réalisés durant une journée, et une balance connectée qui mesure leur masse. En outre les participants sont invités à effectuer des activités physiques variées et à regarder des leçons portant sur l'amélioration de leur mode de vie. En plus de cela, le contact direct entre un participant et Stimul est assuré par un coach qui fait des recommandations, encourage et motive les patients à effectuer leurs activités.

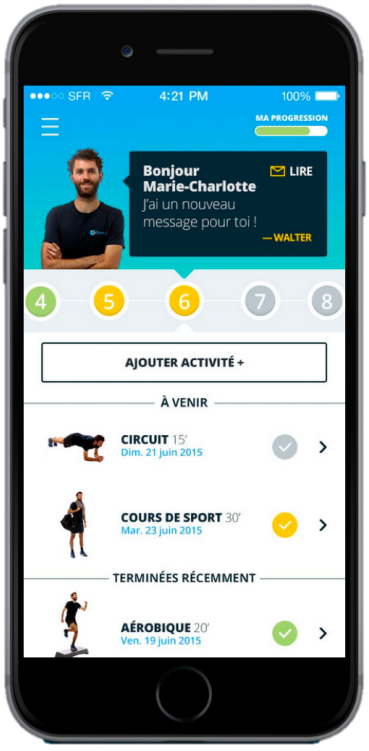

FIGURE 1 – Présentation du dispositif sur téléphone mobile

Ainsi le suivi de Stimul passe par la collecte de données de ses utilisateurs, certaines sensibles car liées à l'identité et d'autres moins spécifiques mais plus utiles pour le programme (nombre de pas, leçons faites, ..).

1.2 Leurs besoins.

De la présentation générale précédente, on peut identifier le besoin du projet. Stimul n'a pas les moyens d'analyser eux-mêmes leurs données : nous devrons donc voir dans quelle mesure les données nous permettent de dire si la méthode opératoire de Stimul est efficace. Cela se décline en plusieurs sous-questions : le regroupement par cohorte est-il utile ? Les participants réussissent-ils le programme en général ? Les liens avec l'éducateur jouent-t-il un rôle ?

1.3 But du projet.

On peut décomposer le but du projet en 2 grandes parties.

D'une part, la réalisation des besoins identifiés à la partie précédente. D'autre part, la transmission de nos compétences à Stimul pour leur permettre d'effectuer eux-mêmes leurs analyses dans le futur. Cela passera à la fois par la connaissance des outils utilisés (Python, pandas, numpy, ...) ainsi que les codes utilisés.

2 Mise en place

2.1 Exploration des données

2.1.1 Étude de la bibliographie

Avant de commencer le projet en lui-même nous nous sommes intéressés à la "validation du programme de thérapie digitale". Pour cela Etienne nous a donné quelques articles de presse et articles scientifique pour nous aider à cerner les enjeux autour des programmes de thérapie digitale. Par la suite nous nous sommes intéressés à des articles scientifiques et études cliniques [1] pour observer le type de traitement statistique [2] mis en place. On a pu observer l'importance de cohortes et phénomènes de groupes, ainsi que le type de graphiques et visualisation des données utilisés.

2.1.2 Les fichiers .csv

Le premier jeux de données obtenus était un ensemble de 7 documents .csv (*comma separated value*) donnant des indications par participant, en effet il s'agit d'extraits de la base de donnée de Stimul dont les données ont été anonymisées. Un nettoyage préalable se révéla nécessaire pour enlever les caractères spéciaux, les valeurs aberrantes, et d'une façon générale assurer la cohérence du fichier.

A cela fut ensuite ajouté un fichier excel contenant des extraits d'échanges entre un coach et un participant (un travail d'anonymisation avait été réalisé par Stimul au préalable) ; ainsi qu'un fichier .csv spécifiant les cohortes.

Il y a eu un réel travail sur le compréhension de la signification réelle de chaque tableau de donnée qui nous a été donné. En première partie il a fallu s'interroger sur le sens de chaque colonne des tables des .csv afin de pouvoir réfléchir aux jointures pertinentes à réaliser par la suite entre les différents fichiers.

On voit que dans cet exemple la jointure est assez naturelle et se fait entre l'id de la première table et le customer_id de la seconde.

id	user_id	gender	height	weight	deletedAt	createdAt	disease_id
1	6	2	180	71.0	NULL	09/05/2016 14 :47	1
4	9	2	186	82.0	NULL	10/05/2016 18 :01	1
5	12	1	165	62.0	NULL	20/05/2016 20 :16	1
6	13	2	175	29.6	NULL	26/05/2016 10 :21	1
7	14	1	169	66.0	NULL	01/06/2016 15 :35	1
16	25	2	187	80.0	NULL	30/06/2016 12 :45	1
18	27	1	172	55.0	NULL	30/06/2016 21 :43	3
22	31	1	174	65.0	NULL	04/07/2016 16 :22	3
42	59	2	175	65.0	NULL	12/09/2016 12 :23	5
43	60	1	154	55.0	NULL	14/09/2016 21 :48	2

TABLE 1 – Extrait du .csv Customer un des 7 premiers .csv

id	lesson_id	customer_id	state	date
1	1	1	done	08/05/2016 10 :00
2	2	1	done	15/05/2016 10 :00
3	3	1	done	22/05/2016 10 :00
4	4	1	new	29/05/2016 10 :00
5	5	1	done	05/06/2016 10 :00
6	6	1	new	12/06/2016 10 :00
7	7	1	done	19/06/2016 10 :00
8	8	1	new	26/06/2016 10 :00

TABLE 2 – Extrait du .csv Customer_lesson un des 7 premiers .csv qui indique les lessons réalisées

2.1.3 Pandas.

Pandas [3] est une librairie python qui permet de manipuler facilement des données à analyser :

- manipuler des tableaux de données avec des étiquettes de variables (colonnes) et d'individus (lignes).
- ces tableaux sont appelés DataFrames, similaires aux dataframes sous R.
- on peut facilement lire et écrire ces dataframes à partir ou vers un fichier tabulé.
- on peut facilement tracer des graphes à partir de ces DataFrames grâce à matplotlib.

C'est donc l'outil idéal pour notre projet, il correspond parfaitement à nos besoins et est par ailleurs documenté et en open source [4].

2.1.4 Google analytics - .JSON

JavaScript Object Notation (JSON) est un format de données textuelles dérivé de la notation des objets du langage JavaScript. Il permet de représenter de l'information structurée sous forme de dictionnaire. C'est le langage utilisée par Google Analytics, plate-forme elle-même utilisée par Stimul pour voir l'activité de son site.

Google Analytics est un service gratuit d'analyse d'audience d'un site Web ou d'applications utilisé par plus de 10 millions de sites, soit plus de 80 % du marché mondial.

Nous avons recus des fichiers .JSON récapitulant les connexions (nombre de pages visitées et temps passé sur le site) de 3 utilisateurs.

2.2 Organisation du travail.

Pour lancer le projet nous sommes allés plusieurs fois à Paris dans les locaux de Stimul ou Latitudes pour nous imprégner des enjeux. Par la suite nous avons travaillé chaque mardi sur le campus. Pour collaborer entre nous, nous avons utilisés Trello et Google Drive, pour se répartir les tâches et se transmettre les documents, Jupyter pour réaliser les codes en collaboration, et WhatsApp pour communiquer. Nous avions un appel par semaine avec Etienne (notre client, co-fondateur de Stimul) pour rendre compte de nos avancées et se mettre d'accord sur la suite, ainsi qu'avec Quentin (notre mentor) qui nous guidait sur des questions de fond et nous donnait des conseils techniques afin de perfectionner chaque semaine notre code par itération.

3 Analyse des données.

3.1 Définition des KPI.

Stimul a défini des indicateurs clés pour étudier l'efficacité médicale de son programme (*Key Performance Indicator*) :

1. Un KPI pour le nombre de pas réalisé par chaque utilisateur. Il est fixé à 840000 pas sur 120 jours.
2. Un KPI associé à la perte de poids : il est fixé à 7% de perte de poids sur 120 jours
3. Un KPI associé aux MET (activité physique) : 9000 MET sur 120 jours. L'équivalent métabolique (MET) est une méthode permettant de mesurer l'intensité d'une activité physique et la dépense énergétique. On définit le MET comme le rapport de l'activité sur la demande du métabolisme de base. L'échelle d'équivalence métabolique va de 0,9 MET (sommeil) à 18 MET (course à 17,5 km/h).

De plus nous avons de notre côté fixé deux autres KPI liés à l'implication des utilisateurs dans le programme. Tout d'abord un KPI associé aux leçons réalisées par les utilisateurs. Ce KPI compte le pourcentage moyen de leçons faites sur 120 jours (on compare le nombre de leçons faites au nombre de leçons disponibles). Il vaut 1 si l'utilisateur a suivi au moins 66,7% des leçons, 0 sinon.

Le deuxième KPI que nous avons ajouté est associé à l'implication des utilisateurs pendant les trois premières semaines de programme. Il compte le nombre d'actions volontairement réalisées (leçons, activités, ajout de données).

3.2 Présentation et analyse du notebook KPI.

Le notebook Notebook_KPI_com.ipynb contient des fonctions permettant de calculer les KPI pour chaque utilisateur, et les compile dans un tableau. L'importation des données et leur manipulation se font grâce au module `pandas` présenté précédement.

3.2.1 Nettoyage préalable des données.

Les fichiers sont importés en utilisant la commande `pandas.read_csv`. La première étape du nettoyage est de convertir les dates. En effet les dataframes créés contiennent des colonnes de dates, qui sont importées au format string (i.e comme des chaînes de caractères). On veut les convertir au format `datetime`, qui permet d'effectuer des manipulations sur les dates comme des soustractions pour obtenir une durée.

On utilise la commande `date_parser = pd.to_datetime`. On doit donc spécifier à l'importation quelles colonnes de chaque tableau on veut convertir :

- Dans les fichiers 1_Activity.csv, 6_Metric.csv et 4_Customer_Lesson.csv, on veut convertir la colonne `date`.
- Dans les fichiers 11_Customer.csv et cohortes.csv on veut convertir la colonne `created_at`.

Néanmoins la commande `date_parser` ne convertit que les dates qui sont écrites au format $jj/mm/aahh : min$. Ceci pose un problème car dans le dataframe `user` les dates de naissances des utilisateurs sont au format $jj/mm/aa$. On créé donc une liste `birthday` qui contient les dates au bon format, et on la met dans le tableau user à la place de la colonne `birthday` déjà présente :

```
birthday = pd.to_datetime(user['birthday'], format='%d/%m/%Y', errors = 'coerce')
user = user.drop(['birthday'], axis = 1)

user['birthday'] = birthday
```

Notons que l'exécution du programme laisse intacts les fichiers csv, ces derniers sont importés sous forme de tableaux (des dataframes) et ce sont ces dernier qui sont ensuite modifiés.

Les données qui nous été envoyées par Stimul sont donc restées intactes.

3.2.2 Résultats du programme.

Le notebook KPI renvoie un tableau récapitulatif des calcul de KPI et une matrice de corrélation des colonnes les plus importantes de ce tableau.

Les tableau final contient les colonnes suivantes :

— `customer_id`

— `user_id`

— `gender.`

— `age.`

— `heigth.`

— `weight` : qui est le poids à la fin du programme.

— `disease_id`

— `poids_ini`

— `poids_fin`

— `perte_de_poids`

— `Periode de perte de poids` : cette donnée est le temps écoulé entre la première et la dernière entrée du poids d'un utilisateur.

— `KPI_pas_data` : le nombre de lignes correspondant à des entrées de nombre de pas pour calculer la valeur du KPI sur 120 jours.

— `KPI_pas_done` : valeur binaire valant 1 si l'objectif du KPI des pas est validé, 0 sinon.

— `pas_data_total` : le nombre de ligne total correspondant à des entrées de nombre de pas.

— `pas_total` : nombre total de pas réalisé par l'utilisateur depuis qu'il est dans le programme.

— `KPI_pas_duree` : la période sur laquelle l'utilisateur a rentré ses nombres de pas (durée entre l'entrée la plus récente et l'entrée la plus ancienne). Cette donnée sert trouver le nombre de pas ramené sur 120 jours.

— `KPI_pas` : le KPI des pas (sur 120 jours).

— `KPI_MET_duree` : la période sur laquelle l'utilisateur a effectué des activités lui faisant augmenter son score de MET (cette valeur est limitée à 120 jours).

— `KPI_MET_data` : le nombre d'activités réalisées par l'utilisateur sur 120 jours.

— `KPI_MET` : le nombre de MET réalisés par l'utilisateur sur une durée 120 jours.

— `KPI_MET_done` : indique si le KPI a été réalisé (1) ou non (0). La valeur seuil est de réaliser 9000 MET en 120 jours.

— `MET_data_total` : le nombre d'activités réalisées par l'utilisateur sur toute sa durée de participation au programme.

— `MET_total` : le nombre de MET réalisés par l'utilisateur sur toute sa durée de participation au programme.

— `MET_duree_total` : La durée totale sur laquelle l'utilisateur a réalisé des activités lui faisant augmenter son score de MET.

— `KPI_lesson` : pourcentage de leçons effectuées.

— `KPI_lesson_done` : 1 si KPI lesson $>$ 0,66 - 0 sinon.

— `nbr_lecon_faites` : nombre de leçons faites par les utilisateurs sur les 3 premières semaines.

— `nbr_acti_faites` : il s'agit du nombre d'activités faites pendant les 3 premières semaines de programme.

— `nbr_metrics` : il s'agit du nombre d'entrées dans le tableau metrics, associées à des actions volontaires(l'utilisateur a lui même rentré la donnée), sur les 3 premières semaines.

La matrice de corrélation en valeur absolue est la suivante :

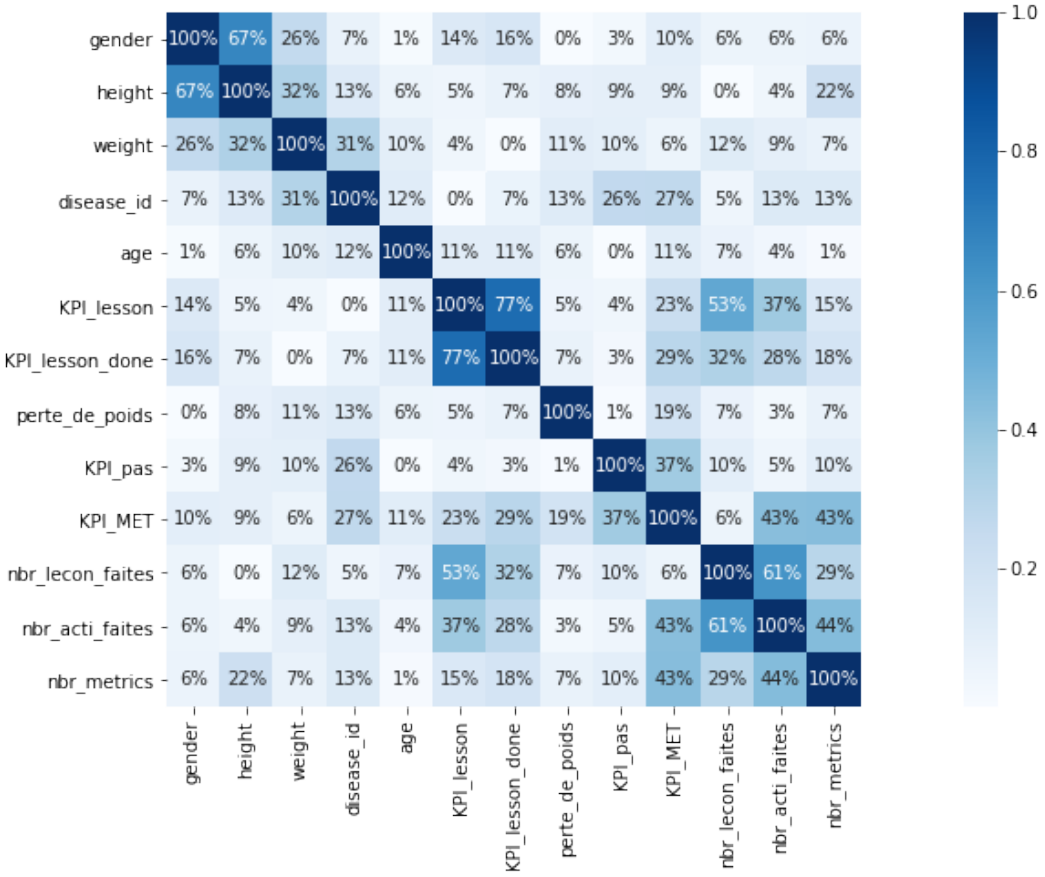

FIGURE 2 – Matrice de corrélation du tableau KPI.

On remarque tout d'abord des corrélations logiques : gender et taille, KPI_lesson et nbr_lesson_faites. On remarque de plus une corrélation entre les colonnes KPI_MET et KPI_pas, qui indique que les utilisateurs ayant la meilleure activité physique ont tendance aussi à marcher le plus (c'est aussi un résultat qui semble logique). Le disease_id est corrélé au KPI des pas et a celui des MET, ce qui montre qu'il y a bien un dynamique de cohorte. Le nombre d'activités réalisées est corrélé au nombre de leçons faites, ce qui montre que les utilisateurs assidus sur l'un le sont aussi sur l'autre.

Ce premier Notebook renvoie enfin la moyenne de chacuns de ces indicateurs moyennés par cohortes. On y observe une grande variabilité selon la cohorte du profil des utilisateurs, et cela peut permettre parfois d'avoir une vue d'ensemble sur les données observées.

disease_id	gender	height	weight	age	KPI_lesson	weight_loss	KPI_pas	KPI_MET	user_nb
1	1.5	173	81	36	0.44	2.08	827090	3676	28
2	1.2	166	75	42	0.42	16.51	3079351	1387	9
3	1.2	166	67	46	0.33	1.44	1836686	2751	9
5	1.0	165	60	47	0.32	1.64	4436734	621	34
8	1.0	164	74	49	0.53	1.56	1065984	2371	13
10	1.4	169	71	37	0.35	6.05	1254993	3538	7
11	1.0	163	74	54	0.49	3.38	877707	1834	8
12	1.5	168	73	47	0.44	1.43	631355	1420	4
14	2.0	191	112	45	0.20	2.69	1225967		1
18	1.6	173	72	49	0.50	-1.09	960275	1299	8
19	1.4	168	76	44	0.34	2.62	719054	464	12
20	1.0	166	122	41	0.20	-0.03	788212	1157	5
21	1.0	167	85	48	0.65	-6.61	568838	655	6
22	1.4	158	60	22	0.44		1478263	1226	7
23	1.3	167	120	34	0.25	-0.51	432051	454	10
24	1.3	165	78	45	0.31	0.94		1151	12
25	1.5	164	86	53	0.15	1.41	1361763	2060	13

TABLE 3 – Résultat de moyenne des KPI par cohorte

3.3 Présentation et analyse du notebook temporalité

Dans la partie précédente nous nous sommes concentrés uniquement sur les KPI, c'est à dire la valeur des paramètres (MET, pas, leçons) prises à 120 jours. Pour compléter notre analyse nous allons nous intéresser à l'évolution temporelle de ces paramètres au cours de toute la durée du programme.

Pour cela, nous avons réalisé le Notebook Temporalité. Des commentaires accompagnent le Notebook pour sa prise en main et la compréhension de sa structure, cependant l'on peut rappeler ici l'idée fondamentale du code. A chaque fois qu'un utilisateur réalise une activité qui lui fait monter la valeur du paramètre (que ce soit le nombre de pas, le nombre de leçons ou les MET), cela sa traduira par un nouveau point dont l'abscisse représente l'instant où l'activité a été faite et l'ordonnée représente la valeur cumulée du paramètre jusqu'à cet instant.

Le Notebook donne l'évolution temporelle des 3 paramètres pour toutes les cohortes. On va ici extraire quelques courbes et les analyser.

On observe plusieurs points :

• Malgré des échelles différentes, ce qui est normal puisque les MET et les pas sont deux choses différentes, les courbes ont globalement la même tendance. Cela tend à prouver une corrélation entre ces deux paramètres : plus une cohorte réalise de MET, plus il y aura de chance qu'elle réalise de pas, et inversement. Ce résultat peut s'expliquer en considérant que les deux paramètres sont fondamentalement liés à une activité physique.

• De plus, on observe à environ 200 jours, une augmentation significative des MET et des pas. Cela tend à montrer un "effet de groupe". En effet, cette cohorte comporte une vingtaine de personne,

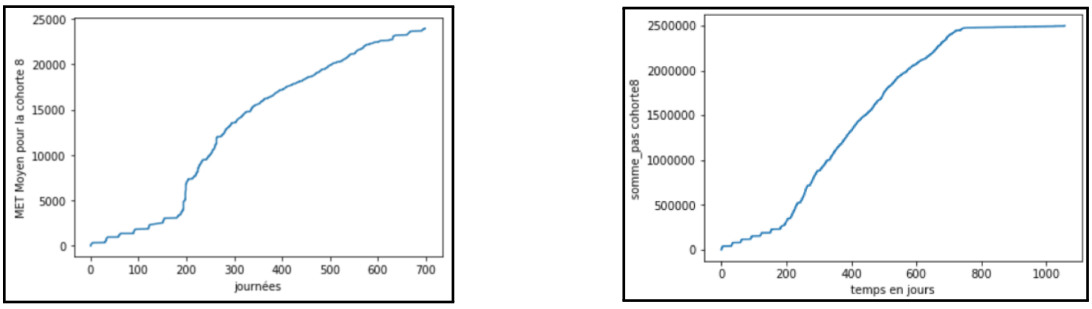

FIGURE 3 – Comparaison de l'évolution des MET et du nombre de pas pour la cohorte 8

et l'on peut donc penser qu'une personne motivée est susceptible de motiver d'autres personnes de la cohorte pour réaliser les exercices. De même, un groupe peu motivée ne sera que tirée vers le bas. Un point complémentaire qu'il faudrait traiter est : quelle est la raison de cet engouement soudain ?

Ajoutons à présent l'évolution des leçons.

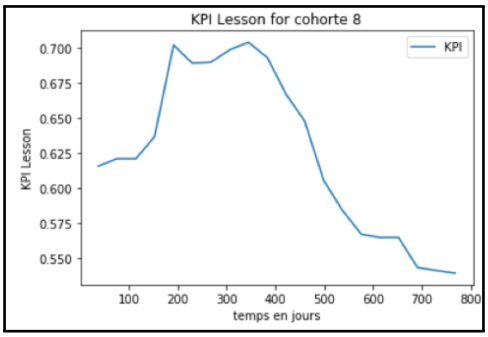

FIGURE 4 – Évolution temporelle du pourcentage de leçons réalisées par la cohorte 8

On observe cette fois ci que l'évolution des leçons ne suit pas du tout la même dynamique que les pas et les MET. Cela est compréhensible dans la mesure où la réalisation des leçons n'est pas liée à une activité physique, contrairement aux deux autres paramètres. Il ne semble donc pas y avoir de corrélation. On notera qu'on a ici représenté l'évolution cumulée du pourcentage de leçons réalisées, on observe quand même des diminutions car des leçons deviennent disponibles au fur et à mesure du programme.

Les remarques précédentes s'appliquent aussi à la cohorte 23 (voir les courbes en dessous) et à la plupart des autres cohortes (voir le Notebook).

3.4 Présentation et analyse du notebook implication

Dans cette partie, on explore des fichiers .json de Google Analytics et mise en forme de données exploitables pour obtenir des informations sur l'implication de l'utilisateur.

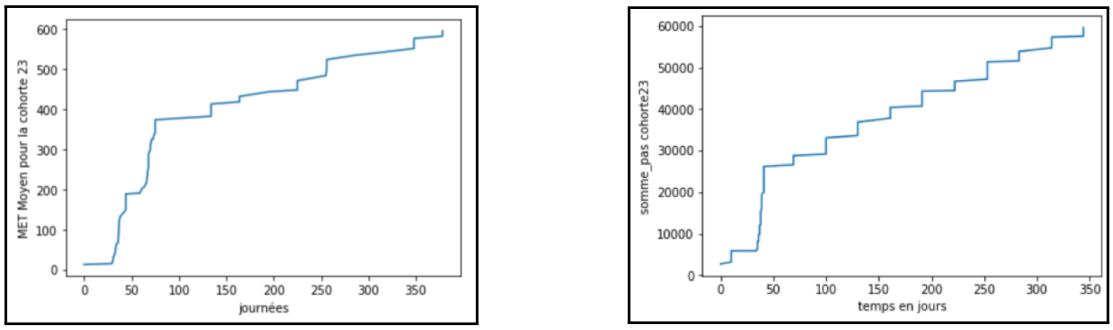

FIGURE 5 – Comparaison de l'évolution des MET et du nombre de pas pour la cohorte 23

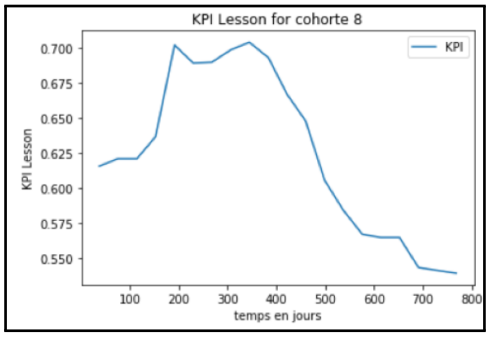

FIGURE 6 – Évolution temporelle du pourcentage de leçons réalisées par la cohorte 23

Le notebook Traitement des fichiers JSON.ipynb contient des fonctions permettant d'analyser des fichiers JSON et de présenter le temps passé sur le site et le nombre de pages visitées par jour des utilisateurs (cf 3.5).

On définit des fonctions intermédiaires qui permettent de traiter les fichiers .json.

— `importation` : La fonction `importation` permet d'importer un fichier .json dans python.

— `minute` : La fonction `minute` prend en argumment un string de la forme 'mm :ss' et renvoie le nombre de secondes correspondant. Cette fonction va permettre de convertir les colonnes 'duration' des fichiers importés.

— `normalize` : La fonction `normalize` prend en argument un fichier .json et l'aplatit en un dataframe contenant : les dates de connections des users, le nombre de pages visitées à chaque connection, et le temps passé en secondes.

— `concat` : La fonction `concat` prend en argument une liste de fichier .json, leur applique la fonction `normalize` et concatène les dataframes obtenus.

De plus, on définit une fonction `evol_temporelle` qui regroupe les fonctions intermédiaires et renvoie l'evolution temporelle du temps passé sur le site et du nombre de pageviews sur le site.

D'après ces données, on peut trouver les pics du temps passé sur le site et du nombre de pageviews par jour des utilisateurs. On peut relier ces données à des événements se déroulant dans le monde réel et analyser les raisons de l'activité de l'utilisateur.

3.5 Etude détaillée des users 578, 579 et 588

On va s'intéresser de manière plus précise aux users 578, 579 et 588. Ils font partie de la cohorte 21, qui correspond aux multi-pathologies chroniques. Il s'agit de 3 hommes.

On extrait pour ces users les lignes correspondantes du tableau KPI_df (avec la commande KPI_df[KPI_df['user_id'].isin([578,579,588])]). Les données KPI_pas et KPI_MET sont des valeurs sur 120 jours.

customer_id	user_id	KPI_pas	height	disease_id	age	KPI_lesson	KPI_lesson_done
443	578	268105	166.0	21	37.0	0.263158	0.0
444	579	41901	164.0	21	45.0	0.857143	1.0
452	588	45622	175.0	21	48.0	0.947368	1.0

perte_de_poids(%)	poids_ini	poids_fin	KPI_pas_done	KPI_MET	nbr_acti_faites
2.671148	80.757143	78.6	0.0	270.0	7.0
-1.923077	78.000000	79.5	0.0	466.	25.0
-0.383877	104.200000	104.6	0.0	NaN	1.0

FIGURE 7 – tableau récapitulatif des KPI pour les users 578, 588 et 579

On s'intéresse aussi aux données des fichiers JSON qui sont regroupées dans les graphiques ci-dessous. Il y'a trois graphiques par user. Le premier (en vert) trace le temps passé sur le site chaque jour, le deuxième en bleu le nombre de pages visitées par jour, et le dernier le nombre cumulé de pages visitées en fonction du temps ; la première barre pointillée bleue marque les trois premières semaines du programme, la deuxième les 120 jours.

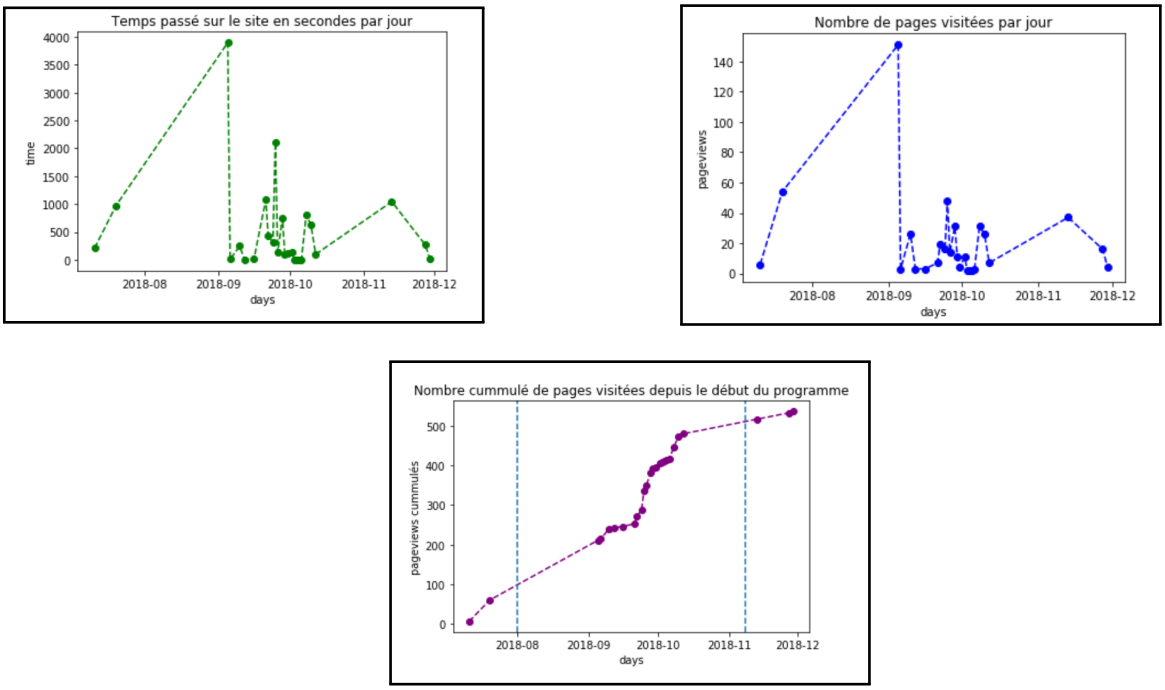

FIGURE 8 – tableaux récapitulatifs pour le user 578

Étudions les critères de réussite du programme pour ces users. Le numéro 578 n'a pas satisfait le KPI des leçons, ni celui du nombre de pas, ni celui des MET (il n'a fait que 270 MET sur les 120 premiers jours, alors que le KPI est fixé à 9000 MET). Il n'a fait que 7 activités, mais par contre il est le seul à avoir perdu du poids (il a perdu 2,67% de son poids sur toute la durée du programme).

Le user 579 a réalisé 86% des leçons, on considère donc qu'il a réussi le KPI leçons ; il a réalisé 25 activités mais que 466 MET sur 120 jours, ce qui est très loin de l'objectif de 9000. Il a légèrement gagné du poids, mais seulement 1,5 kg, ce qui n'est pas une valeur très fiable étant donné toutes les incertitudes qui entrent en jeu lors du pesage d'une personne.

Le user 588 a fait 95% des leçons mais n'a fait qu'une seule activité et son poids est resté stable. Il n'a pas satisfait le KPI des pas et n'a pas rentré d'information sur les MET, ce qui laisse supposer qu'il n'a pas réalisé le KPI des MET.

Pour ces 3 utilisateurs on remarque que leur activité sur le site/application de Stimul suit une dynamique particulière : si on s'intéresse aux graphiques du nombre cumulé de pages visitées, les users réalisent la majorité de leur activité entre les 3 premières semaines et les 3 premiers mois, puis l'activité ralentit.

De plus, les courbes de nombre de pages visitées par jour et du temps passé par jour sur le site ont les mêmes allures, ce qui semble logique (plus le user visite de pages, plus il passe de temps sur le site), sauf pour le user 588. A partir de novembre 2018 il passe beaucoup de temps sur le site mais visite assez peu de pages. Ceci peut signifier qu'il ne se concentre que sur un aspect du programme (les leçons en l'occurrence pour cet utilisateur) et délaisse les autres (il n'a jamais rentré d'information sur les MET) , où alors qu'il laisse l'application ouverte pendant qu'il fait autre chose.

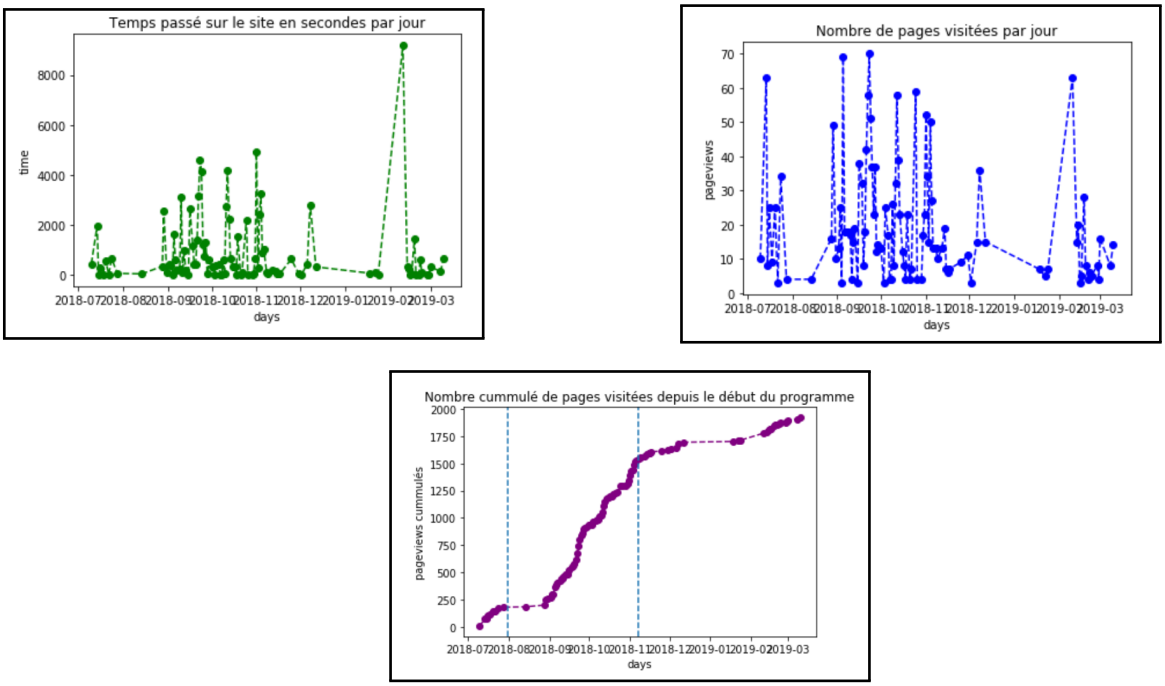

FIGURE 9 – tableaux récapitulatifs pour le user 579

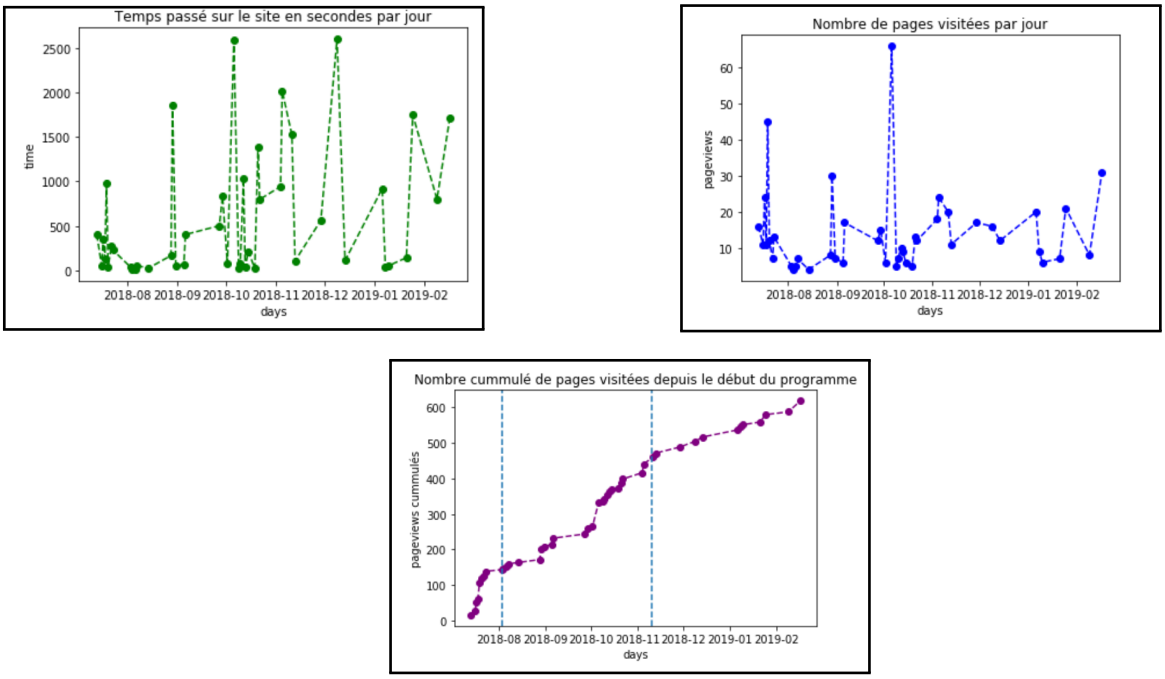

FIGURE 10 – tableaux récapitulatifs pour le user 588

Regardons maintenant les résultats de la cohorte 21 :

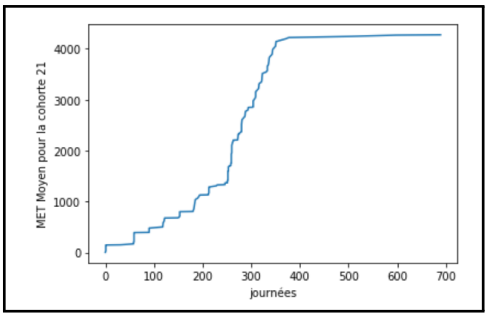

FIGURE 11 – Nombre de MET moyen cumulé en fonction du temps

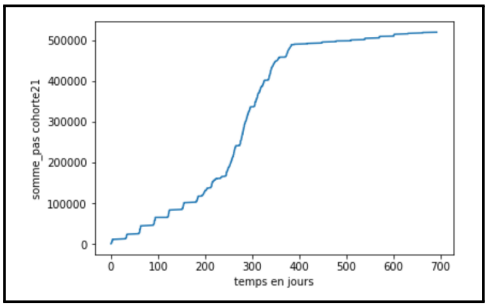

FIGURE 12 – Nombre de pas moyen cumulé en fonction du temps

On voit qu'en moyenne un utilisateur de la cohorte 21 a fait environ 600 MET et 75000 pas en 120 jours. Les 3 utilisateurs que nous étudions sont largement en dessous de ces moyennes, sauf pour le user 578 qui a fait plus de pas que la moyenne.

Les résultats moyens de cette cohorte sont largement en dessous des objectifs de 9000 MET et 840000 pas en 120 jours.

4 Conclusions

Pour conclure, au travers de notre étude nous avons mis en évidence certaines corrélations entre différents features grâce à la matrice de corrélation, exploré les données par cohortes pour analyser la variance qui existe entre les cohortes et grâce à une étude de la temporalité observer un phénomène de "progression de groupe" aussi bien au niveau des KPIs que de l'implication dans le programme observé dans les fichier .json.

Grâce à ce projet, nous avons sommes montés en compétences en analyse de donnée, avons découvert de nouvelles bibliothèques python comme Pandas, et nous sommes familiarisé à l'utilisation de Jupyter Notebook. Ce projet nous a par ailleurs permis de trouver un sens à ce que nos compétences d'ingénieurs peuvent apporter au sein de la "Tech for Good" en particulier dans le cas présent appliqué au domaine médical. Il a donc été pour nous très porteur et nous encourage à continuer de chercher dans nos futurs métiers d'ingénieur un sens appliqué, concret et pour le bien commun.

Nous avons cherché à rendre toutes nos études et nos codes commentées, génériques et exploitables afin que de futures données et projets supplémentaires soient faciles à mettre en place.

Nous espérons que ce projet aura une suite ; coté Stimul, afin d'améliorer leur application et leur programme grâce aux analyses et aux insights apportés par ce projet et qu'il donnera lieu à de futurs projets dans la continuité de celui-ci : comme le développement d'un dashboard, qui permettrait à partir des KPIs calculés dans nos notebooks de pouvoir suivre en temps réel l'évolution de ceux-ci sur une cohorte afin de pousser le groupe à se dépasser, ou bien l'exploration des données sémantiques afin de chiffrer l'influence du rôle des coachs dans le programme.

Nous voulions enfin remercier Latitudes et Stimul de nous avoir proposé ce projet si riche, et en particulier Etienne pour sa présence afin de nous guider dans le projet et en comprendre le sens ainsi que Quentin pour son accompagnement technique et ses nombreux conseils pour mener le projet à bien.

5 Bibliographie

Références

[1] Lisa Affengrubera Viktoria Titschera Isolde Sommera Nina Matyasa Gernot Wagnera Christina Kiena Irma Kleringsa Gerald Gartlehnera Anna Glechnera, Lina Keuchelb. Effects of lifestyle changes on adults with prediabetes : A systematic review and meta-analysis. *Primary care diabetes*, 12 :393–408, 2018.

[2] Eliza Gibson Erica N. Madero Barbara Rubino Janina Morrison Debra Rosen Wendy Imberge Michael R. Cousineau Sue E. Kim, Cynthia M. Castro Sweet. Evaluation of a digital diabetes prevention program adapted for the medicaid population : Study design and methods for a non-randomized, controlled trial. *Contemporary Clinical Trials Communications*, 2018.

[3] Wes McKinney. pandas : a python data analysis library.

[4] Wes McKinney. Data structures for statistical computing in python. In Stéfan van der Walt and Jarrod Millman, editors, *Proceedings of the 9th Python in Science Conference*, pages 51 – 56, 2010.

[5] Daniel Muñoze Justin Bachmanne Ashton Stahlc Ryan Case Cardella Leak Russell Rothmana Sunil Kripalania Christianne L. Roumiea, Niral J. Patela. Design and outcomes of the patient centered outcomes research institute coronary heart disease cohort study. *Contemporary Clinical Trials Communications*, pages 42–48, 2018.

[6] Grace A. Rowan Julie Gray Thomas R. Blue Roxanne Muiruri Kevin Knight Wayne E.K. Lehman, Jennifer Pankow. Staysafe : A self-administered android tablet application for helping individuals on probation make better decisions pertaining to health risk behaviors. *Contemporary Clinical Trials Communications*, 10 :86–93, 2018.