Learning to Detect Heavy Drinking Episodes Using Smartphone Accelerometer Data

Projet de Python for Data Analysis

par Max CRASTES

Petite paranthèse

Le dataset pris n'est pas le même que celui attribué, en effet, le dataset attribué (Unmanned Aerial Vehicle Intrusion Detection) été vide (https://archive.ics.uci.edu/ml/datasets/Unmanned+Aerial+Vehicle+%28UAV%29+Intrusion+Detection)

Le dataset prit sur le même site est Bar Crawl: Detecting Heavy Drinking (https://archive.ics.uci.edu/ml/datasets/Bar+Crawl%3A+Detecting+Heavy+Drinking)

Article publié sur l'étude avec ce dataset:

http://ceur-ws.org/Vol-2429/paper6.pdf

L’Objectif

- L’idée derrière cette étude est de pouvoir donner des messages préventifs pour l’alcool au bons moments pour limiter les risques et dangers dues à sa consomption
- Site web pour lequel la recherche à été effectué :

https://www.scramsystems.com/scram-international/gb/

La base de données

- La base de donnée a été donné par 3 étudiants et professeurs d'université
- Il a été généré à partir d'observations effectués sur 20 étudiants (21-23 ans) qui consomme de l'alcool régulièrement (en 2017)
- La base de données contient les coordonnées x, y, z d'un téléphone portable obtenues aux travers d'une application mobile mesurées sur une fréquence de 40Hz pendant 18h, les données collectées sur un bracelet mesurant le taux d'alcool dans le sang toutes les 30min
- Les données de 7 participants ont été faussées et donc ne sont pas prises en compte
- Les données d'accéléromètre telles que publiées ont été traité avec un filtre passe-bas
- Les données d'alcoolémie (TAC) ont été filtrées et déplacées de 45min en arrière pour prendre en compte le temps que l'alcool passe dans le sang

Le dataset

Le dataset ce décompose en de multiples fichier csv :

- Le premier contient les données d'accéléromètre, le PID du participant et le temps du relevé
- Le deuxième est un dossier avec des fichiers csv de mesure du TAC (taux d'alcoolémie) et la date du relevé. Un fichier csv est disponible par participant. Ce fichier est présent en non traité et pré-traité.
- Le dernier fichier csv regroupe la catégorie de téléphone utilisé pour les participants

Le traitement de la base de données

- Le premier objectif était de récupérer chaque dataset (sauf celui de la catégorie de téléphone, non utilisé dans l'étude ou dans cette recherche)
- Le deuxième objectif était de lié les datasets (les mesures de TAC ne correspondent pas aux mesure de temps et les données sont dispersées par participants
- Le troisième objectif était de créer des variables qui vont être traités à partir des coordoonées x, y, z. En effet, les coordoonées à elles seules ne sont pas forcément utiles car le référentiel utilisé n'est pas précisé (si celui-ci correspond au référentiel du téléphone, les données sont inutiles en tant que telle, si celui-ci est géocentrique, il y a potentiellement un rapprochement).

Le traitement de la base de données

- Les données crées à partir des coordonnées sont les mouvements déplacements entre 2 mesures, la vitesse de déplacement et l'accélération subie. Dedans, le minimum, le maximum et la moyenne de chaque élément sera prise en compte.
- Le quatrième objectif est de construire des modèles prédictifs
- Le dernier objectif est d'en faire une API simple.

La création des nouvelles variables

- Pour la création de nouvelles variables, j'ai rassemblé toutes les données d'accéléromètre comprises entre 2 mesures de taux d'alcoolémie (en supprimant les valeurs avant la première mesure de TAC car la prise de mesure n'a pas débuté en même temps).

Les modèles

3 modèles ont été effectuées, le premier est une régression, en effet, le taux d'alcoolémie est une valeur continue. Les deux autres sont des modèles de classification

- Régression linéaire simple, on pourrait imaginer une simple corrélation entre quelqu'un d'alcoolisé et les mouvements effectués. Ce modèle s'avère être et ciblé autour d'une valeur d'alcoolémie du notamment à la faiblesse de la base de données (données pas très fiables aussi bien avec la position du téléphone qui peut être n'importe où, les déplacements ou non des participants et le faible nombre de données) et potentiellement un modèle ou des valeurs peu adaptés

Les modèles

Les deux autres modèles sont des classifications pour se mettre en lien avec l'étude. Les résultats sont donc positifs si TAC > 0.08 et négatifs sinon

- Gradient Boosting Classifier, ce modèle de gradient boosting est utilisé aussi bien en régression qu'en classification. Il permet de ciblé les différentes catégories à prédire et d'en faire un ensemble. Il semblé particulièrement adapté à cet étude.
- Decision Tree Classifier, ce modèle permet de séparé en 2 la base de données selon des critères et à multiples reprises. Ce modèle a été choisit aussi bien car il existe potentiellement des valeurs clés qui font la différence, que le fait que le meilleur modèle dans l'étude était un RandomForest, et le decision tree s'en rapprochait.

Les résultats

Les résultats ont été médiocre, malgré le fait qu'on obtient tout de même 80% de prédiction sur le test set (avec le gradient boosting classifier), les résultats peuvent fluctués assez car la base de données n'est malheureusement pas assez riche en données (mesures, participants, répartition des mesures sur l'état d'ébriété ou non) ni assez bien réglementé et dirigé (comment conserver le téléphone, problème de matériel, différents modèles de téléphones, perte de connexion du téléphone).

J'aurai tout de même pu traité les informations différemment afin d'obtenir de meilleur résulta (ex: mettre le TAC dans les mesures de mouvements / vitesses /… en dupliquant la mesure du TAC sur toutes les vitesses correspondantes au lieu de faire l'inverse afin d'enrichir la base de données.

L’API

L’API n’est pas simple d’utilisation, en effet, les valeurs ne sont pas propices à entre en tant que tel.

Soit l’API prend en compte les mesures d’accéléromètre, mais la fréquence et le nombre de mesure doivent être fixes et ceci revient à énormément de données, soit je prends en compte les nouveaux paramètres par soucis de nombres de valeurs, mais conserve les même problèmes.

L’API n’est donc pas propice à ce type d’étude, néanmoins dans le cadre du projet, celui-ci a été réalisé dans API.py et une requête simple est envoyé dans request.py. Celui-ci prend en compte les 9 variables créés (mouvements, vitesses, …) par soucis de simplicité.