Projet SY09: Analyse des données: "An Open Dataset for Human Activity"

ABOUD Solene, GALTIER Anthony, ZEROUATI Wassim

10 juin 2019

Résumé

Ce rapport présente une analyse du jeu de données "An Open Data Set for Human Activity". Ce jeu de données est constitué de mesures d'activité humaine réalisées sur un individu. Dans ce projet, nous avons tenté d'utiliser ces données pour prédire l'activité pratiquée par le sujet de l'expérience au moment de chaque mesure. Nous nous sommes basés pour cela sur les connaissances acquises dans l'UV SY09 - Analyse de données et Data Mining.

1 Introduction

Aujourd'hui, les appareils connectés, omniprésents dans notre quotidien, sont devenus une source exceptionnelle d'information brute pour l'étude de l'activité humaine. Ces appareils sont munis de capteurs relativement précis permettant de collecter des données de nature diverse sur l'activité physique d'un individu, ses mouvements, son environnement, ses émotions et autre. Dans ce projet, nous avions à notre disposition trois ensembles de données typiques de ce domaine, collectés respectivement par une smartwatch, un smartphone et une paire de lunettes connectées. Nous avions également à notre disposition des informations relatives à l'activité réalisée par l'individu au moment de la collecte des données. Le sujet que nous nous sommes proposé a donc été de construire un modèle prédictif de l'activité réalisée par un individu en fonction des données collectées par ses appareils connectés du quotidien.

Nous présenterons en partie 2 les ensembles de données étudiés avant d'en faire une analyse plus approfondie en 3. Nous expliquerons alors les pré-traitement effectués sur les ensembles pour se ramener à une forme exploitable avec les méthodes étudiées dans le cadre de ce cours. Après une analyse exploratoire des données pré-traitées en 4.2, nous expliquerons nos choix de modèles et de paramètres pour la classification supervisée partie 5. Enfin, nous présenterons et analyserons nos résultats dans la section 6 avant de conclure.

2 Présentation des données

Les données que nous étudions dans ce projet sont issues de trois sources différentes : une smartwatch, un smartphone et une paire de smartglasses. Pendant 15 jours consécutifs en Juillet 2017, un individu équipé de ces dispositifs a collecté des données lorsqu'il effectuait certaines activités du quotidien [1]. L'intérêt premier de la smartwatch est de collecter des informations sur le rythme cardiaque de l'individu. Le smartphone permet de collecter des informations contextuelles, en particulier relatives à l'environnement informatique de l'individu à travers les interfaces Wifi et Bluetooth de l'appareil. La smartwatch et le smartphone permettent aussi de collecter certaines informations relatives aux mouvements de l'individu issues de l'accéléromètre de ces appareils. Enfin, les lunettes collectent des informations sur l'activité oculaire de l'individu (clignement et orientation des yeux) et sur les mouvements de l'individu par un accéléromètre et un gyroscope (mouvements de la tête).

En plus de ces mesures, pendant le temps de l'étude, l'individu a enregistré les activités qu'il réalisait par le biais de l'application TimeLogger [1].

Nous disposons de ce fait de quatre ensembles de données au format CSV :

- smartwatch.csv
- smartphone.csv
- glasses.csv
- report.csv

2.1 Données issues de la Smartwatch

L'ensemble de données collecté par la smartwatch est un fichier au format CSV de 200 471 lignes. Chaque ligne correspond à une mesure effectuée par la smartwatch pour lesquelles on dispose de l'horodatage de la mesure (colonne "timestamp"), du type de mesure en

1. http ://www.atimelogger.com/

question (colonne "source" prenant une valeur parmi 'heart rate', 'step detector', 'accelerometer'...) et enfin des valeurs de ces mesures sous forme d'une chaîne de caractères représentant la valeur ou la liste des valeurs en question (colonne "values").

Sous cette forme, semblable à un dictionnaire, l'ensemble de données n'est pas directement exploitable. Nous avons donc procédé à une analyse préliminaire et un pré-traitement quelque peu fastidieux de cet ensemble qui seront détaillés par la suite.

2.2 Données issues du Smartphone

Nous avons ici un smartphone qui est utilisé afin de collecter des données quant à l'activité de l'utilisateur. Pour le smartphone, les données sont essentiellement contextuelles, c'est-à-dire quelles permettent de recueillir des informations relatives à l'environnement de l'utilisateur à un moment de la journée.

L'ensemble de données est donc un fichier csv de 1 528 218 mesures pour lesquelles nous avons comme précédemment le "timestamp", la "source" et enfin la "value".

Ces différentes mesures proviennent de 17 sources différentes dans notre jeu de données :

- Principalement, nous avons une API qui permet de reconnaître l'activité de l'utilisateur (Immobile, Debout, Dans un véhicule, etc.) qui liste ses activités de la plus probable à la moins probable.
- Une source permettant de mesurer le son ambiant.
- Des sources permettant de savoir si l'individu a effectué un pas et compter son nombres de pas.
- Des sources permettant de reporter les appareils bluetooth ainsi que les points d'accès Wi-Fi à proximité.
- Diverses sources relatives aux mouvements smartphone (accélération, gravité, etc.)
- Enfin, une dernière source relevant le niveau de batterie

Pendant la plupart des mesures, le téléphone était dans la poche de l'utilisateur.

Cet ensemble de données requiert de changer le format des données afin de pouvoir être traité. En effet, nous devons transformer notre colonne "source" en de multiples colonnes pour chaque type de capteurs afin de pouvoir analyser ce jeu de données. Cette analyse préliminaire et ce pré-traitement sera détaillé dans la suite du rapport.

2.3 Données issues des Smartglasses

Les données "*smartglasses*" ont été collectées grâce au port de lunettes Jins Meme [2]. Il s'agit d'une paire de lunettes connectées dotées de 9 capteurs : des capteurs de gyroscope et d'accélération ainsi que 3 voltmètres. Ces derniers mesurent la tension en trois points sur la peau de l'individu : de chaque côté des yeux, ainsi qu'entre les deux yeux, au niveau du nez. Ces mesures sont utilisées pour retourner les 4 mesures d'electro-oculographie [3] que nous récupérons dans le dataset. Le schéma en annexe **??** illustre le fonctionnement de ces capteurs et le calcul des données transmises.

Nous avons donc 3 types de données dans le jeu de données *smartglasses* :

- des données d'accélération sur 3 axes X,Y,Z : on y obtient une information quand aux déplacements de l'individu
- des données de gyroscope donnant des informations sur l'inclinaison et les mouvements de la tête de l'individu
- des données d'électro-oculographie (EOG) symbolisant le mouvement des yeux via des différences de tension.

Les lignes de ce jeu de données correspondent à des relevés effectués toutes les 20 ms. Pour chaque ligne, on dispose d'un horodatage (timestamp) et des 10 mesures expliquées précédemment, toutes sous forme d'un float.

L'ensemble des données collectées par la smartwatch est un fichier au format CSV de 3 940 000 lignes. Pour autant, le temps de port des lunettes ne correspond qu'à 6.31% du temps de l'expérience. Pour de nombreuses activités, l'individu ne porte pas ses lunettes. On remarque en effet que le jeu de données report comprend à l'origine 24 activités, alors que l'individu ne porte les lunettes que pendant 12 de ces activités. Les lunettes sont si peu portées qu'elles n'apportent pas assez de données pour être traitées avec les jeux de données "*smartwatch*" et "*smartphone*". Nous les étudierons donc à part.

2.4 Données de Report

Le jeu de données report est le retour de l'application *TimeLogger*. Le sujet de l'expérience a pu y entrer quotidiennement ses activités.

Ce jeu de données consiste en 6 colonnes d'informations :

- **index :** Le rang de l'activité dans la journée ;

2. https ://jins-meme.com/en/
3. L'electro-oculographie est une technique normalement utilisée en médecine pour analyser le sommeil ou confirmer un diagnostique. Elle permet de mettre en valeur les mouvements des yeux.

— **duration :** La durée de pratique de l'activité ;
— **from :** Le moment auquel la pratique de l'activité a débutée, précis à ... ;
— **to :** Le moment auquel la pratique de l'activité s'est terminée, précis à la minute près ;
— **activity_type :** Le nom de l'activité réalisée par le sujet de l'expérience pendant cet intervalle de temps ;
— **comment :** L'individu a la possibilité de laisser ici un commentaire pour détailler sont activité. Très peu de commentaire ont été laissés.

Nous nous intéresserons tout particulièrement aux instants de début et fin des activités ainsi qu'aux types d'activités. Les deux premières données permettront d'associer les données des autres jeux de données aux types d'activités. Nous ne gardons donc que ces trois types de données pour le jeu de données "rapport".

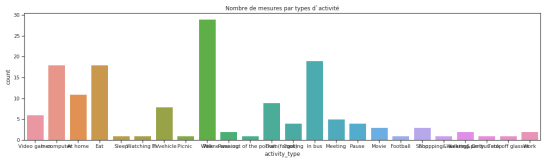

FIGURE 1 – Types d'activités dans le jeu de données report

Le variable *activity_type* est composée de 24 modalités. Pour autant, nous remarquons que le nombre de données par modalité est très inégalement réparti, comme nous pouvons le voir sur la figure 1. Cette variation risque de changer lorsque le jeu de données "*report*" sera joint aux autres jeux de données. Nous adapterons donc les données prises pour la classification après avoir joint les jeux de données. Nous devrons bien sûr faire en sorte que l'ensemble des données que nous obtenons se répartisse de manière homogène sur toutes les activités.

3 Analyse préliminaire des données et pré-traitement

3.1 Smartwatch et Smartphone

Afin de pouvoir exploiter les ensembles de données issus de la smartwatch et du smartphone, nous avons cherché à nous ramener à un tableau individus-variables pertinent. Nous avons donc commencé par pivoter les tableaux de données sur la colonne source de manière à obtenir un premier tableau individus-variables dans lequel chaque individu (ligne) correspond à une mesure différente et chaque variable (colonne) correspond à un type de mesure issue de la colonne 'source' précédente. La colonne 'timestamp' est conservée et indexe les mesures (lignes) de l'ensemble de données.

En s'intéressant au nombre de mesures réalisées pour chaque source différente, nous avons observé dans l'ensemble de données issu du smartphone que le nombre de mesures relevées pour les sources 'bluetooth' et 'light' sont très inférieures aux autres. Afin d'éviter des problèmes liés à un trop grand nombre de valeurs nulles dans la suite de notre étude, nous avons choisi de mettre de côté les mesures de ces deux sources.

Dans l'ensemble de données issu de la smartwatch, le nombre de mesures de 'heart rate' prédomine très largement. Puisqu'il est de plus spécifié que l'intérêt de la smartwatch est exclusivement de mesurer l'activité cardiaque de l'individu [1] et pour les même raisons qu'avec les données smartphone, nous avons choisi de mettre de côté les mesures d'autres sources.

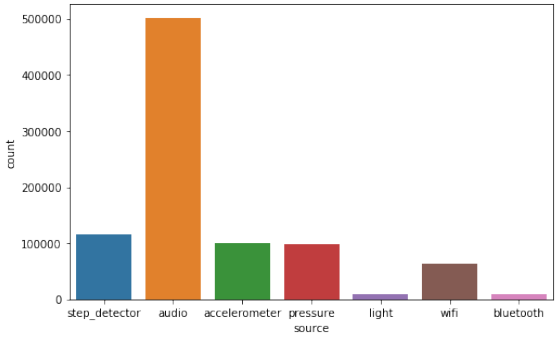

FIGURE 2 – Diagramme en barre du nombre de mesures relevées pour chaque source dans l'ensemble de données issu du smartphone.

Nous ne retenons pas la mesure principale du smartphone (i.e. l'activité de l'utilisateur reconnue par l'API). En effet, nous avons effectué une analyse indépendante des autres variables de cette mesure. En séparant chacune des 8 activités en colonnes nous avons fait un boxplot.

On remarque que la majorité des activités reconnues (et les plus probables) sont celles où il est immobile. Dans un premier temps, cela nous indique que ces données ne seront pas totalement utiles par rapport aux réelles activités indiquées dans le jeu de données 'report'.

En recoupant avec les variables retenues du jeu de données smartphone, nous retrouvons le même problème que pour les sources 'bluetooth' et 'light' car il y aurait donc eu un trop grand nombre de valeurs nulles issues des activités de l'utilisateur.

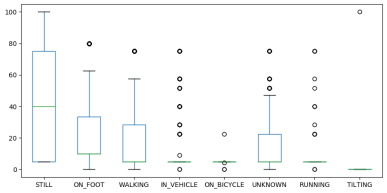

FIGURE 3 – Diagramme en boîte des activités reconnues les plus probables

Finalement nous avons décidé de ne pas prendre en compte la source 'activity'.

Les ensembles de données sont alors des tableaux individus-variables de mesures indexées par l'horodatage de celles-ci. Les variables retenues pour le smartphone sont l'amplitude audio maximale mesurée par le microphone du téléphone (colonne 'audio'), le nombre de borne wifi détectées à proximité du smartphone (colonne 'wifi'), les pas détectés (colonne 'steps' qui prend la valeur 1 si un pas est détecté et 0 sinon). Une mesure d'accéléromètre étant un vecteur tri-dimensionnel de réels représentant l'accélération linéaire selon trois axes orthogonaux. Nous avons choisi de représenter cette information (colonne 'accelerometer') en calculant la norme euclidienne de ce vecteur :

$$\|\vec{a}\| = \sqrt{X^2 + Y^2 + Z^2} \tag{1}$$

La seule variable retenue pour l'ensemble de données issu de la smartwatch est le rythme cardiaque en battement par minute (colonne 'heart rate').

A ce stade, chaque ligne contient des valeurs nulles puisqu'une ligne correspond à la mesure d'une seule source. En partant de l'hypothèse que des mesures sur un intervalle d'une minute sont représentatives de l'activité d'un individu, nous avons donc décidé d'agréger les valeurs par minute. L'agrégation par minute permet en particulier d'obtenir des lignes pleines, sans valeur nulles. Pour le rythme cardiaque, la norme du vecteur d'accélération, l'amplitude audio et le nombre de bornes wifi, nous calculons la moyenne des valeurs observées sur une minute. Pour les pas, nous en retenons la somme. L'index est alors l'horodatage à la minute près de ces mesures.

Dans l'objectif de classifier les mesures selon le type d'activité, nous avons pensé qu'il était pertinent de considérer la dimension temporelle. Nous avons tout d'abord ajouté une variable explicative 'hour', qui pour chaque ligne va retenir l'heure de la journée correspondante.

L'heure peut toutefois être considérée comme un variabnle qualitative ordinale représentant la période de la journée. Nous avons donc choisi à posteriori de l'encoder ('Category Encoders') ; ici nous utilisons le 'one-hot encoding' qui va simplement binariser la variable que nous avons introduite (en utilisant des dummy variables).

3.2 Smartglasses

Pour exploiter l'ensemble de données "*smartglasses*", nous avons d'abord cherché à limiter le nombre de variables afin d'éviter les données répétitives. De fait, par exemple, les données d'accélération sur les axes X,Y,Z sont très fortement liées. De même, généralement, lorsque l'oeil droit bouge, l'oeil gauche en fait autant, alors les données EOG sont elles aussi répétitives.

Pour les trois types de données (accélération, gyromètre et EOG) nous souhaitons quantifier un mouvement, un "niveau d'activité". Nous avons donc décidé d'agréger les données pour chaque type en calculant leur norme euclidienne.

Ainsi, pour les données d'accéléromètre et de gyroscope, une mesure est un vecteur tri-dimensionnel de réel représentant l'accélération ou l'orientation selon trois axes orthogonaux. Nous avons donc représenté cette information en calculant la norme euclidienne de ce vecteur.

$$\left\|A\vec{C}C\right\| = \sqrt{X^2 + Y^2 + Z^2} \tag{2}$$

$$\left\|GY\vec{R}O\right\| = \sqrt{X^2 + Y^2 + Z^2} \tag{3}$$

Pour l'electro-oculographe, la mesure est répartie sur un vecteur à 4 dimensions.

$$\left\|E\vec{O}G\right\| = \sqrt{L^2 + R^2 + V^2 + H^2} \tag{4}$$

Nous remarquons également que des relevés toutes les 20 ms fournissent beaucoup trop de données, ce qui rend le traitement difficile. En émettant l'hypothèse que des mesures sur un intervalle d'une minute sont représentatives de l'activité d'un individu, nous avons décidé d'agréger les valeurs par minute. Pour cela, nous avons réalisé la moyenne sur chacune des variables des valeurs observées sur une minute. L'index du jeu de données est maintenant l'horodatage à la minute près.

Nous avons ensuite procédé à l'étiquetage du jeu de données. Pour ce faire, nous nous sommes appuyés sur le jeu de données report. Lorsque les l'horodatage d'une donnée de *glasses* est comprise dans l'intervalle

de temps d'une activité de *report*, cette activité y est associé dans la nouvelle colonne "activity_type".

Nous obtenons alors un tableau individu-variable de 874 lignes pour 12 classes. Mais les données sont inégalement réparties dans chaque classe. Nous avons, par exemple 17 données pour l'activité "cooking" contre 347 données pour l'activité "at home". Nous supprimons donc les activités extrêmes pour lesquelles nous avons trop ou trop peu de données, ainsi que les classification trop larges. Ces dernières sont par exemple les classes "walking and party" ou "at home" qui peuvent correspondre à des activités très différentes, et même inclure d'autres classes.

Une fois ce tri effectué, nous obtenons un tableau individu-variable de 467 lignes avec 7 classificateurs ayant entre 32 (pour le classificateur "In bus") et 75 (pour le classificateur "Walk") lignes.

3.3 Jointure des jeux de données

Une fois l'agrégation par minute réalisée, les tableaux individus-variables smartphone et smartwatch sont unifiés par une opération de jointure interne sur l'index des mesures (colonne 'timestamp'). Nous avons ensuite étiqueté les lignes du tableau résultant en s'appuyant sur les données du fichier 'report.csv'. Lorsque les lignes dont l'index (l'horodatage) correspondent à une activité annoté dans l'ensemble de données 'report', le nom de l'activité réalisée est ajouté dans la nouvelle colonne dédiée 'activity type'.

Dans le cadre de notre étude, nous nous intéressons essentiellement aux méthodes d'apprentissage supervisée pour classifier les données observées selon le type d'activité réalisé. Nous conservons donc exclusivement les individus étiquetés, c'est à dire les lignes dont on connaît le type d'activité correspondant. En s'intéressant ensuite à la distribution des mesures selon l'activité, on observe que certaines sont très peu représentées. Les activités "Shopping" et "Video Games" par exemples sont représentées par moins de 20 lignes chacune contre 60 pour la classe médiane et 82 en moyenne. Nous avons donc choisi de mettre de côté les activités les moins représentées afin, d'une part, d'éviter des problèmes lié au déséquilibre des classes rencontrés dans certains modèles de classification comme les K-plus proche voisins et, d'autre part, de limiter le nombre de classes.

4 Analyse exploratoire des données

L'ensemble de données est un tableau individu variable de 920 lignes. Les variables sont le rythme cardiaque moyen par minute (colonne 'heart_rate'), la norme moyenne par minute du vecteur d'accélération (colonne 'accelerometer'), le nombre de pas détectés par minute (colonne 'steps'), l'amplitude audio moyenne par minute (colonne 'audio') et le nombre moyen par minute de bornes wifi détectées à proximité (colonne 'wifi').

La colonne 'timestamp' indexe les mesures à la minute près. Chaque ligne est associée à une unique classe parmi 11 au total (colonne 'activity type'). Chaque classe est représentée par un nombre de ligne entre 52 et 125 lignes, 81 pour la classe médiane et 83,6 en moyenne.

4.1 Analyses mono-dimensionnelles

On s'intéresse tout d'abord à la distribution des mesures selon le type d'activité réalisée pour chaque variable. Pour visualiser ceci, pour chaque variable, nous avons aligné les diagrammes en boite des valeurs selon le type d'activité. A travers ces graphiques, nous pouvons faire de premières observations quant aux différences de distributions entre les types d'activité. On observe par exemple dans la figure 4 que l'environnement semble plus bruyant lors d'un picnic que lors des autres activités et au contraire, que l'environnement est plus silencieux pendant le sommeil.

En ce qui concerne le rythme cardiaque, la médiane varie peu d'une activité à l'autre, toutefois l'étendue et l'écart inter-quartile change. La figure 5 suggère par exemple que le rythme cardiaque est plus variable lorsque l'utilisateur est sur son ordinateur que lorsqu'il est en réunion.

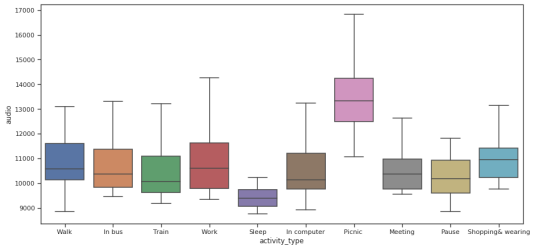

FIGURE 4 – Diagrammes en boite de l'amplitude audio moyenne par minute mesurée selon le type d'activité

Puisque certains modèles d'apprentissage pour la classification supposent que les données sont distribuées

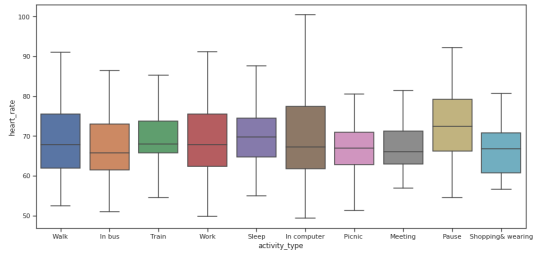

FIGURE 5 – Diagrammes en boite du rythme cardiaque moyen par minute mesuré selon le type d'activité

normalement, nous avons soumis les différentes variables à un test d'adéquation à une loi normale : le test de Shapiro-Wilk. Pour chaque variable et chaque classe, la valeur p de ce test est inférieure à 0.01. On rejette donc l'hypothèse que les valeurs suivent une loi normale au niveau de signification de 1% pour chaque variable et chaque classe. On peut donc s'attendre à ce que les modèles basés sur des hypothèses de normalité comme les analyses discriminantes ne fonctionnent pas bien sur nos données.

4.2 Analyses multi-dimensionnelles

Nous nous intéressons ici au lien qu'il peut exister entre les différentes variables. Pour ce faire nous avons calculé et représenté les coefficients de corrélation empirique entre chacune des variables. Les valeurs de ces coefficients ne dépassent pas 0.3 ce qui semble montrer que les variables ne sont pas ou du moins peu linéairement liées entre elles.

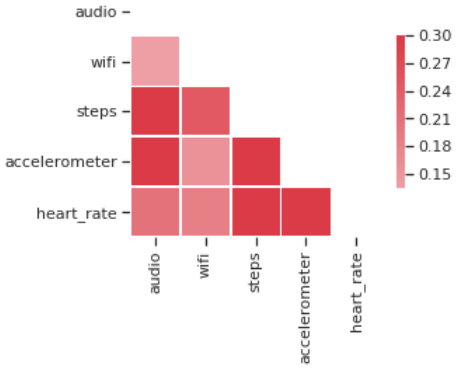

FIGURE 6 – Distribution et dispersions des points dans les espaces des variables les plus corrélées

On pourrait toutefois s'attendre à ce qu'il existe un lien entre certaines variables au vu de leur nature. En particulier, il pourrait exister un lien entre les valeurs de l'accéléromètre, le ryhtme cardiaque et le nombre de pas. C'est d'ailleurs entre ces trois variables que le coefficient de corrélation est le moins négligeable. Afin de visualiser avec plus de détail ces liens, nous avons produit un graphique matriciel (multiplot) représentant les points dans les plans correspondants au différents couples de variables. En diagonale de ce multiplot sont représentées les distribution mono-variées de chaque variable.

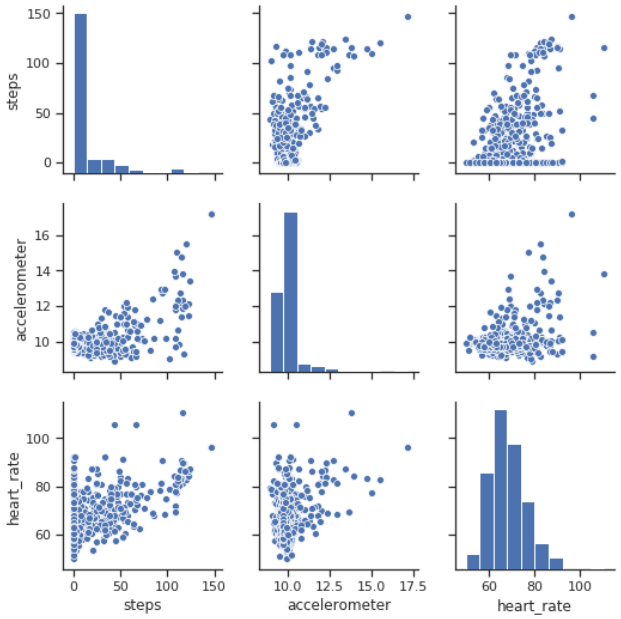

FIGURE 7 –

On observe ainsi qu'en effet, il n'y pas de lien linéaire significatif entre ces variables. Toutefois, au vu de la dispersion des points observés, on pourrait penser qu'il existe une relation entre la variance de l'accéléromètre et le nombre de pas et de même entre la variance du nombre de pas et le rythme cardiaque.

4.3 Analyses exploratoire des données smart-glasses

Nous avons maintenant un tableau individu-variable *glasses* de 467 lignes. Ses variables sont les normes moyennes par minute du vecteur d'accélération (colonne "ACC"), du mouvement gyroscopique (colonne "GYRO") et du mouvement des yeux (colonne "EOG"). La colonne 'DATE' indexe les mesures à la minute près. Chaque ligne est associée à une unique classe parmi 7 au total (colonne 'activity type'). Chaque classe est

représentée par un nombre de ligne entre 30 et 80 lignes. Ce tableau est indéxé par la date, un timestamp avec un intervalle à la minute.

En traçant sur un intervalle de temps les trois données, elles semblent coïncider, comme on peut le voir sur la figure 8.

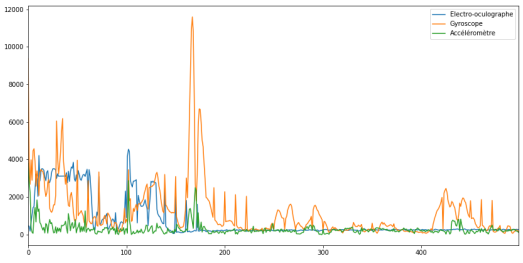

FIGURE 8 – Représentation graphique des données d'électro-oculographe, et gyroscope et d'accéléromètre.

On peut alors se poser la question de l'existence d'une corrélation linéaire entre les différentes variables. Nous nous sommes donc intéressés plus particulièrement au lien entre ces variables et avons calculé les coefficients de corrélation empirique entre ces différentes variables. Nous obtenons la matrice de corrélation suivante (figure 9) sur laquelle on peut remarquer des coefficients de corrélations toujours inférieurs à 0.3. On peut en déduire que ces variables sont peu linéairement liées entre elles.

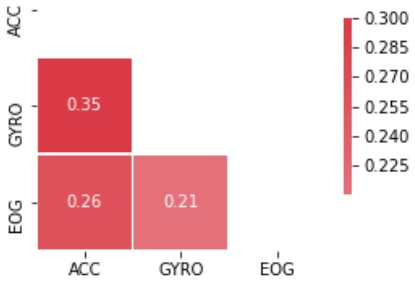

FIGURE 9 – Matrice de corrélation pour les données glasses

5 Classification supervisée

Nous avons pour objectif dans cette partie d'apprendre un modèle qui permettra de déterminer le type d'activité à partir de nouvelles mesures observées des variables explicatives. La variable à prédire, le type d'activité est une variable qualitative à 11 modalités. Les variables explicatives sont le rythme cardiaque moyen par minute, la norme moyenne par minute du vecteur d'accélération, le nombre de pas détectés par minute, l'amplitude audio moyenne par minute et le nombre moyen par minute de bornes wifi détectées à proximité. Nous disposons donc de 920 observations de 5 variables réparties de manière assez équilibrée dans les 11 différentes classes.

5.1 Analyse factorielle

Le nombre de variables est faible et au vu de l'analyse exploratoire précédente, les valeurs de celle-ci ne sont pas significativement liées. De plus, le nombre d'observations par classe est relativement grand (entre 60 et 120 observations par classe). On peut donc penser qu'on ne rencontrera pas de problèmes lié au fléau de la dimension. L'intérêt des méthodes d'analyse factorielle semble ainsi limité. Nous avons tout de même effectué une analyse par composante principale afin de visualiser les points dans un espace réduit.

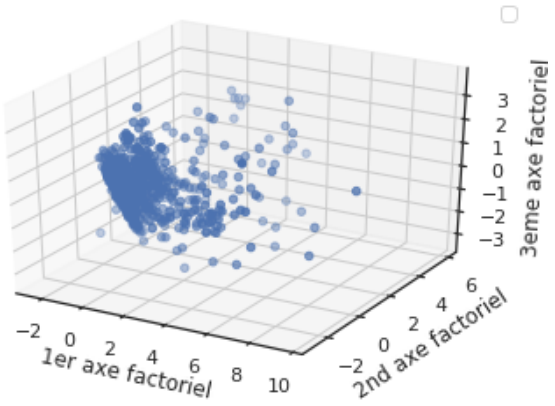

FIGURE 10 – Représentation du nuage de points selon les trois premiers axes factoriel de l'ACP

En effectuant l'analyse par composante principale, on est capable d'expliquer 82% de l'inertie totale à partir des trois premiers axes factoriels. Nous pouvons visualiser les points dans l'espace tri-dimensionnel défini par ces axes factoriels.

Si nous souhaitions obtenir une représentation fidèle

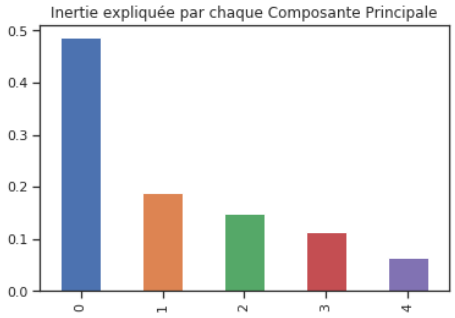

FIGURE 11 – Inertie expliquée par les composantes principales de l'ACP

de nos données dans un espace de dimension inférieure à 5, il serait judicieux de retenir au moins les deux premiers axes factoriels. Le second axe factoriel marque en effet une rupture dans la décroissances des inerties expliquées par chaque axe factoriel (méthode du coude) et avec au premier axe, permet d'expliquer environ 70% de l'inertie totale.

5.2 K Plus Proches Voisins

Dans notre objectif de classification supervisée, nous avons choisi dans un premier temps d'appliquer l'algorithme des K Plus Proches Voisins (KPPV). L'algorithme des KPPV consiste à affecter le vecteur de valeurs observées à la classe (le type d'activité) la plus représentée parmi celles des K plus proches voisins au sens d'une distance donnée. Nous avons choisi ici par défaut la distance euclidienne.

Comme nous l'avons vu dans l'analyse exploratoire de l'ensemble de données, nous n'avons pas pu justifier l'hypothèse que nos données suivent une distribution normale. Contrairement à d'autres modèles que nous avons écartés qui reposaient sur des hypothèses de normalité des variables, l'algorithme des KPPV a l'avantage de ne pas faire d'hypothèse sur la distribution sous-jacente des données observées. La simplicité de l'algorithme est aussi un avantage. Il est plus facile de comprendre et d'interpréter les résultat des KPPV qu'un algorithme plus complexe avec un grand nombre de paramètre. Bien que l'algorithme des KPPV puisse devenir coûteux en terme de mémoire et de complexité de calcul pour des grands ensembles de données et des grandes valeurs de K, pour notre ensemble de 920 lignes, cet algorithme reste un choix raisonnable.

Puisque l'algorithme des KPPV, tel que nous l'avons implémenté, repose sur le calcul des distances euclidiennes entre les différents points observés, il est pertinent de centrer et de réduire nos données au préalable. En effet, des différences d'échelle significatives entre les mesures des différentes variables déséquilibreraient l'influence de celles-ci pour la classification.

Nous nous sommes enfin penché sur la question du paramètre K (le nombre de voisins) à retenir. Dans l'objectif d'obtenir un modèle généralisable, nous cherchons à ce que les frontières de décision entre les classes soient assez régulières. Nous avons donc cherché à choisir une valeur de K qui ne soit pas trop petite. Pour choisir la valeur de K, nous avons évalué les performances de l'algorithme pour différents nombres de voisins. L'ensemble de données a été séparé en un sous-ensemble d'apprentissage et un sous-ensemble de validation. Le sous-ensemble d'apprentissage contient 70% des points tirés au hasard dans l'ensemble de données et l'ensemble de validation contient le reste. Le critère d'évaluation que nous avons regardé est la proportion d'individus correctement classifiés.

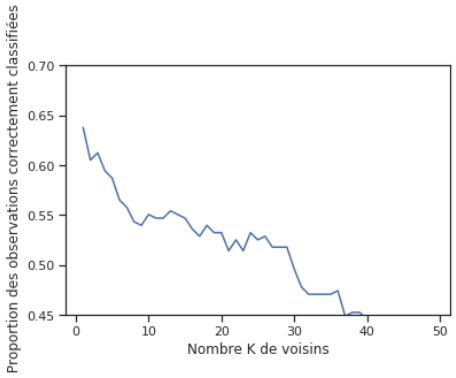

FIGURE 12 – Performance des KPPV selon la valeur de K choisie

On peut supposer qu'une valeur de K inférieure à 10 résulte en des frontières de décision irrégulières au vu du fait que nous avons presque autant de classes à prédire. D'après l'évaluation des performances du modèle, on peut conclure qu'une valeur de k choisie entre 11 et 18 est convenable. Une telle valeur semble être un bon équilibre entre performance du modèle et capacité de généralisation.

5.3 Arbre de Décision

Nous avons décidé dans un second temps d'utiliser les arbres de décision pour notre classification supervisée. En effet, cet algorithme est bien adapté à ce type d'apprentissage. L'avantage d'utiliser les arbres est qu'ils ne font, à priori, aucune hypothèse sur la distribution de

nos variables, dû à la nature non-paramétrique de notre algorithme. En effet, le test de Shapiro-Wilk montre qu'elles ne suivent pas une distribution normale et dans ces cas les arbres de décision ont de bonnes performances.

De plus comme nous l'avons vu, notre jeu de données est assez équilibré avec des mesures bien réparties dans les 11 différentes classes.

Cela permet d'éviter les potentiels biais produits avec des jeux de données déséquilibrés.

Enfin, à la fin de notre pré-traitement nous avions peu de bruit dans nos données malgré qu'il n'y ait aucun lien linéaire significatif entre les données ; notre modèle aura donc de meilleures performances dans ces conditions. En addition, le manque de lien linéaire avantage l'utilisation des arbres de décision car ils prennent facilement des formes non-linéaire. Un dernier avantage important des arbres de décision est leur facilité d'interprétation.

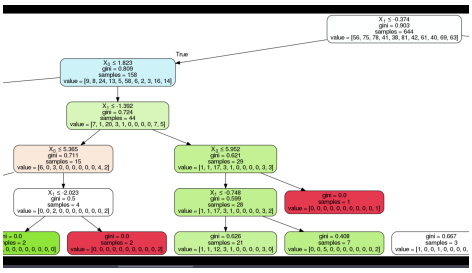
.png

FIGURE 13 – Extrait d'une représentation graphique d'un arbre de décision généré en Python

Nous apercevons dans chacun de nos noeuds le terme 'Gini'. Ce terme Gini fait référence à un quotient qui mesure la pureté du noeud. C'est-à-dire, simplement l'homogénéité des classes, nous pouvons par exemple dire d'un noeud qu'il est pur quand toutes ses mesures appartiennent à la même classe (ces noeuds font référence aux feuilles de l'arbre).

Enfin dans nos paramètres nous devions ajuster le paramètre de profondeur maximale de l'arbre. L'enjeu est important, car si notre profondeur est trop faible nous aurons un problème d'underfitting, au contraire, si notre arbre a une profondeur trop élevée nous aurons un problème d'overfitting.

Ainsi pour obtenir une profondeur optimale nous aurions pu considérer celle-ci comme un hyper-paramètre et tester une série de paramètres et de comparer les performances pour en déduire le meilleur paramétrage. Cependant nous avons décidé de fixer ce paramètre à 5, car il s'agit d'un bon compromis et qu'il s'agit également d'une valeur souvent utilisée pour les arbres de décision, par défaut permettant d'éviter les 2 problèmes cités préalablement.

Puis, à la suite des arbre de décision classiques, nous avons utilisé les forêts aléatoires pour classifier nos données [2]. Une forêt aléatoire est un regroupement d'arbres de décision où les résultats sont agrégés en un seul résultat final. L'avantage des forêts aléatoires est donc leur robustesse par rapport à un simple algorithme d'arbre de décision. L'agrégation de multiples arbres de décisions permet de limiter l'overfitting et donc d'avoir globalement de meilleurs performances.

5.4 Classification binaire répétée

Nous avons d'abord voulu tenter une classification binaire sur un unique classificateur pour voir comment nous arrivions à classifier un élément comme correspondant ou non à une activité spécifique. Pour cela, nous avons utilisé un arbre de décision de profondeur 50 pour le classificateur "Work" et avons obtenu un score d'environ 91.1% de bonne classification.

Nous avons alors testé cette classification binaire pour tous les classificateur et avons obtenu des scores allant de 89.4% (pour le classificateur "In Computer") à 97.4% (pour les classificateurs "Picnic" et "Meeting").

Face à ces résultats encourageants, nous avons donc imaginé une classification binaire répétée : nous pourrions réaliser successivement une classification binaire pour chaque classificateur. En théorie, ces classifications successives devraient apporter un score général de environ $0.93^{13} = 0.39$ de réussite.

Nous avons décidé de tenter une telle méthode de classification dont voici le procédé. D'abord, nous avons décidé d'utiliser le classificateur KPPV qui est celui avec lequel nous obtenons les meilleurs scores de classification binaire. Pour chaque classe, nous avons récupéré des tableaux de prédiction affichant 1 si la donnée était prédite comme faisant partie de cette classe, 0 sinon. Nous avons ensuite agrégé ces prédictions en un unique tableau de résultats contenant les noms des classes prédites, et 'Unknown' pour les données qui nous n'avions pas su prédire. Nous récupérons finalement les données 'Unknown' et les classons selon la méthode classique des KPPV.

Cette méthode permet de classer au mieux chaque classe, puis de classer légèrement moins bien les données non classées.

6 Résultats obtenus

Pour analyser les résultats des différents modèles que nous appliquons à notre jeu de données, nous utilisons principalement deux métriques.

Premièrement nous utilisons le 'score' qui va tout simplement évaluer la performance de notre estimateur par la proportion des individus correctement classifiés.

Puis, pour un résultat plus solide, nous utilisons 'cross_val_score' pour évaluer le résultat de précision de nos différents modèle par cross validation (avec 15 scores différents, dont on donne la moyenne dans le tableau ci-dessous). La cross-validation (ou validation croisée) est simplement une démarche de ré-échantillonnage permettant d'obtenir un meilleur estimateur de l'erreur commise par le modèle que nous avons construit sur l'ensemble de nos données. On vient simplement découper en K différents segments notre base de données de façon aléatoire, on en utilise alors $K-1$ pour l'apprentissage et 1 pour tester. On ré-applique notre modèle autant de fois que nous avons produit de segments, ainsi si le modèle appliqué à notre jeu de données est sensiblement robuste, nous retrouvons lors des 15 différents test des scores de justesses relativement égaux.

Modèle	Score	Précision avec CV
KPPV	0.58	49.643
Arbre de décision	0.47	39.580
Forêts aléatoires	0.55	45.535
ADL	0.31	27.029

Ainsi nous voyons que la méthode donnant le meilleur score est la méthode des K Plus Proches Voisins suivie de près par la méthode des forêts aléatoires. Ces deux méthodes ont des points commun pour donner des résultats assez proche. En effet, la simplicité algorithmique des deux modèles ainsi que le fait de ne pas supposer une quelconque distribution des données préalables font qu'ils performent bien sur notre jeu de données. Les arbres de décisions évaluent moins bien les données que les forêts aléatoires car, comme nous l'avons vu, ils sont moins robustes que l'agrégation que génèrent les forêts aléatoires.

La méthode de l'ADL (Analyse Discriminante Linéaire) est ici simplement à titre comparatif, nous permettant de mieux nous rendre compte des résultats des autres méthodes. En effet, sachant que les données n'étaient pas distribuées normalement, il était d'ores et déjà évident que l'ADL ne fonctionnerait pas correctement. Ainsi il était également peu intéressant d'appliquer une ADQ (Analyse Discriminante Quadratique), qui représente un modèle encore plus complexe que l'ADL. Celle-ci a un pouvoir de généralisation supérieur sans compter les hypothèses de normalité des données. Cela nous donnerait donc un résultat bien inférieur et ne donnerait aucune information supplémentaire pour notre étude.

6.1 Classification binaire répétée

Avec la classification binaire kppv répétée nous obtenons un score général de 59.74% de bonne classification.

Ce score est détaillé dans la matrice de confiance figure 14.

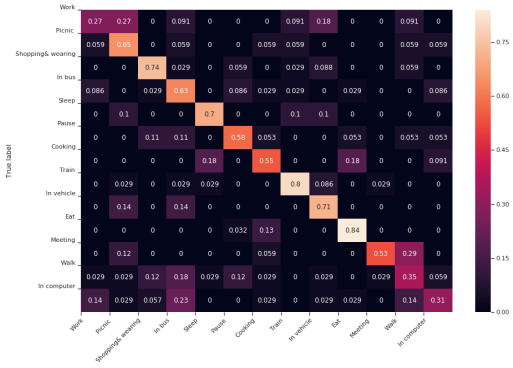

FIGURE 14 – Matrice de confiance de la classification binaire répétée

Finalement, le résultat de cette méthode de classification n'apporte pas une grande amélioration à la classification, par rapport à la méthode kppv. Notons surtout que pour un très léger pourcentage de meilleure classification, l'opération de kppv est répétée 14 fois.

7 Conclusion

Finalement, pour notre analyse du jeu de données "An Open Data Set for Human Activity" nous avons retenu la méthode des K Plus Proches Voisins comme meilleur classifieur avec un coût relativement faible, contrairement à la classification binaire répétée qui est bien plus *time-intensive*.

Cependant, bien que cela dépasse les outils vus en cours, nous aurions pu faire des paramétrages plus poussés pour nos arbres de décisions. Ce plus particulièrement pour les forêts aléatoires possédant différents hyper-paramètres à régler, nous permettant d'obtenir des résultats bien plus significatifs et

d'améliorer la robustesse du modèle.

Chaque ligne semble liée à la ligne précédente puisqu'elles représentent un mouvement. Réaliser des séries temporelles pourrait donc être la méthode la plus représentative d'étudier ces données.

Enfin, quant à nos données en elles-mêmes, nous avons fait le choix lors de notre pré-traitement de ne retenir que très peu de variables. En effectuant un prétraitement plus solide, nous aurions pu garder plus de variables, et perdre moins d'information. Cela est d'autant plus vrai pour les lunettes connectées, dont les données n'ont pas été pleinement exploitées. Ne connaissant pas non plus le type d'activité pour toutes les mesures dont nous disposions, des méthodes d'apprentissage semi-supervisées auraient aussi pu être envisagées.

Références

[1] Sébastien Faye et al. "An Open Dataset for Human Activity Analysis using Smart Devices". In : *Archive ouverte HAL* (sept. 2017). URL : https://hal.archives-ouvertes.fr/hal-01586802.

[2] *Sklearn Ensemble Random Forest Classifier*¶. URL : https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html.

A Annexes

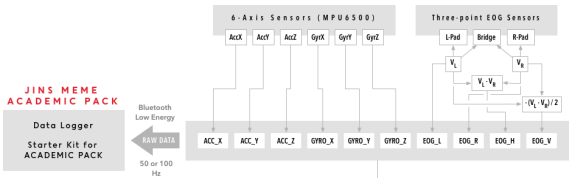

Figure 15 – Origine des données des lunettes connectées