Ministère de l'Enseignement Supérieur et de la Recherche Scientifique

UNIV ALGER 1

RAPPORT

PROJET

Apprentissage Automatique

Partie 1

Objectif : Application des techniques de régression linéaire, de classification et de prétraitement des données

Elaboré par : Akkouche Abderrahmane

Krim Islem

Masdoua Manil

Sassi Kahina

1ère Année Master - ISII

Introduction :

Définition et histoire de l'intelligence artificielle

La première notion d'intelligence artificielle a été abordée en 1950 par le mathématicien Alan Turing. Ce dernier crée alors un test visant à déterminer si une machine peut être considérée comme « consciente ». Le test de Turing est toujours utilisé par les scientifiques de nos jours, mais sa pertinence est régulièrement remise en cause.

Il faudra attendre 1956 pour obtenir une définition de l'IA proposée par Marvin Lee Minsky : « **La construction de programmes informatiques qui s'adonnent à des tâches qui sont, pour l'instant, accomplies de façon plus satisfaisante par des êtres humains car elles demandent des processus mentaux de haut niveau tels que : l'apprentissage perceptuel, l'organisation de la mémoire et le raisonnement critique.** »

L'intelligence artificielle est un vaste domaine qui touche non seulement à l'informatique mais aussi aux mathématiques, à la neuroscience et même à la philosophie. L'IA fascine depuis plus d'un demi-siècle les scientifiques, mais aussi les romanciers et cinéastes. Des cyborgs tueurs de Terminator aux androïdes de Blade Runner en passant par HAL 9000 de 2001, l'Odyssée de l'espace, les humains semblent fascinés par la possibilité de répliquer leur comportement et communiquer avec des machines « **pensantes** ».

Machine learning et Intellligence Artificielle, quel rapport y'a-t-il ?

Machine learning, deep learning, réseaux de neurones, assistants personnels... Ces termes entrés dans nos vies depuis quelques années ont tous trait à des facettes de l'intelligence artificielle. Les progrès scientifiques en la matière sont d'ailleurs époustouflants.

En fait le machine learning est une sous branche de l'Intelligence Artificielle, et qui consiste à utiliser des ordinateurs pour optimiser un modèle de traitement de l'information selon certains critères de performance à partir d'observations, que ce soit des données-exemples ou des expériences passées

Régression : Online Video Characteristics and Transcoding Time Dataset Data Set

Introduction:

Dans l'air de la technologie où l'on vit, nombreuses sont les ressources présentes sur internet, EBOOK, romans, musiques et aussi vidéos.
Des sites web tels que YouTube proposent un service et une base de données de vidéos en ligne en très grande quantité, de toutes catégories allant de loisir et cuisine aux reportages et documentaires. Le contenu vidéo est produit, transporté et consommé dans plus de moyens et de dispositifs que jamais. Pendant ce temps, une interaction sans couture est nécessaire entre la production de contenu vidéo, le transport et dispositifs consommateurs. La différence en ressources de l'appareil, la bande passante du réseau et les types de représentation vidéo entraînent la condition nécessaire pour un mécanisme d'adoption de contenu vidéo. Un tel mécanisme s'appelle **le transcodage vidéo**.

Qu'est-ce qu'une vidéo en ligne ?
C'est une vidéo qu'on peut retrouver sur YouTube ou sur un serveur en ligne (site web en ligne).
Qu'est-ce que Transcoding Time ?
Le transcodage, en vidéo ou en audio, est le fait de convertir le format de codage d'un média (voir aussi codage et codec) utilisé pour comprimer ou encapsuler un média audio ou vidéo dans un fichier ; ou transporter un signal analogique ou numérique. Le but du transcodage est de rendre une vidéo accessible par plusieurs plateformes et dispositifs.

On notera qu'il ne s'agit pas d'un codage au sens strict du terme car le plus souvent la transformation comporte des pertes.
Plus généralement, le terme transcodage est utilisé lorsque l'on change la manière de coder une information.
Les formats de codage employés pour décrire une vidéo doivent tenir compte de nombreux paramètres :

1. Le nombre de lignes et de colonnes formant l'image.
2. Le nombre d'informations nécessaire pour coder un point (profondeur de couleur d'un pixel).
3. Le rapport hauteur/largeur de l'image (4:3, 16:9 par exemple).
4. Le nombre d'images par seconde.
5. La durée de l'enregistrement.

Pourquoi le Transcodage vidéo ?
Le transcodage est actuellement utilisé à des fins telles que : réduction du débit binaire afin de respecter la disponibilité de la bande passante du réseau, réduction de la résolution pour l'adoption de la taille d'affichage, transcodage temporel pour la réduction de la fréquence d'image et finalement le transcodage de la résilience d'erreur pour assurer une haute qualité de service (Qos).
Pourquoi doit-on prévoir le temps d'un Transcodage vidéo ?
Planification à l'exécution des tâches de transcodage en multi cœurs et environnements de cloud est difficile car leurs besoins en ressources peuvent ne pas être connu au préalable. Actuellement, pour les travaux de transcodage vidéo, il faut s'appuyer sur les valeurs les plus défavorables qui conduisent à un sur-approvisionnement des ressources afin de maintenir une qualité de service satisfaisante. Cela est dû au fait que les ressources nécessaires à une tâche de transcodage dépendent fortement des données vidéo à convertir et de leurs paramètres de conversion. Afin de permettre à de tels systèmes distribués et multi cœurs de résoudre le problème de la surproduction, il est nécessaire de disposer d'un procédé permettant de prévoir les besoins en ressources de chaque travail.
Aujourd'hui, les systèmes informatiques varient considérablement les uns des autres et vont de très petit (par exemple, téléphones portables, tablettes, ordinateurs portables), trop grande (serveurs, centres de données, cloud). Cependant, au cœur de chacun de ces systèmes, des composants de gestion des ressources décident de la planification de l'exécution de différentes tâches (en garantissant une utilisation élevée du système ou une utilisation rationnelle de l'énergie) ou en allouant des ressources de programme telles que la mémoire, le stockage et la mise en réseau. En garantissant une longue durée de vie de la batterie ou une allocation équitable des ressources). Ces composants de gestion doivent généralement être en mesure de prédire comment une tâche donnée sera exécutée en fonction de sa taille et de ses autres caractéristiques, de manière à décider de la meilleure façon de planifier pour l'avenir.
Par exemple : en considérant un scénario simple dans un service de transcodage en nuage avec un ensemble de deux types de demandes de transcodage, les travaux de transcodage rapide dans le jeu A et les travaux de transcodage lent dans le jeu B. Un planificateur est souvent confronté à la décision d'exécuter chaque jeu sur différentes ressources de la CPU, ce qui prend potentiellement plus de temps à exécuter ; ou pour s'intercaler entre les deux ensembles et répartir le travail de manière équitable, en exécutant potentiellement les tâches beaucoup plus rapidement. Si le planificateur peut prédire avec précision le temps que chaque travail prendrait sur une plate-forme donnée, il peut prendre une décision optimale en renvoyant les résultats plus rapidement, en minimisant éventuellement l'énergie, le temps d'attente et en maximisant le débit.

Présentation du data set :

Afin de tirer parti de ces opportunités, il est nécessaire d'utiliser un mécanisme de prévision précis. Dans ce rapport nous présentons ce modèle de prévision formé sur la base d'une métadonnée vidéo et paramètres de transcodage sélectionnés et faciles à obtenir (notre dataSet).
En fondant les décisions de planification sur ces connaissances préalables, une meilleure utilisation des ressources peut être obtenue.
Nous possédons 2 parties dans le dataset :

- La première appelée « **youtube_videos.tsv** » contient 10 caractéristiques fondamentales d'une vidéo pour 168 miles vidéo de YouTube, ce fichier va nous être utile pour avoir une idée des caractéristiques de vidéos regardées sur YouTube.

Une présentation simple des attributs de cette dataset :
Id, duration, bitrate (video), height, width, frame rate, frame rate (estimé), codec = même description que pour la 2ème dataset, **category** = la catégorie de la vidéo YouTube, **url** =Le lien directe de la video YouTube.

- La 2eme partie du dataset qui est enregistré dans le deuxième fichier « **transcoding_mesurment.tsv** » contient vingt attributs avec un total de 68000 instances, ils ont été ramassés depuis le premier dataset pour effectuer expérience qui utilise un outil de transcodage de vidéo qui s'appelle ffmpeg4. Cet outil accepte en entrée et sortie des paramètres de transcodage vidéo qui sont les attributs de notre data set.

Les illustrations suivantes nous permettent de bien comprendre le ffmpeg4 (voir un exemple de paramètre de transcodage) :

```
ffmpeg -i input.mkv -c:a copy -s hd720 output.mkv
```

Cette commande nous permet de changer la résolution d'une vidéo.

```
ffmpeg -i input.mkv -c:a copy -s 1280x720 output.mkv
```

Cette commande nous permet aussi de changer la résolution mais nous donne la main de changer également la hauteur et largeur.
Ces vingt attributs on peut les classifier en 3 classes différentes :

1. La première classe qui représente des caractéristiques d'une vidéo avant son transcodage,
2. la deuxième classe qui représente les caractéristiques d'une vidéo après son transcodage.
3. Finalement, la troisième classe qui inclut les attributs qui représentent le temps de transcodage ainsi que la mémoire allouée aux ressources

Une présentation simple des attributs de cette data set :

La classe 1 :

Id = l'identifiant de la vidéo sur YouTube.
Duration = durée de la vidéo.
Codec = codage standard utilisé pour la vidéo, c'est un dispositif matériel ou logiciel permettant de mettre en œuvre l'encodage ou le décodage d'un flux de données numérique, en vue d'une transmission ou d'un stockage
height = hauteur de la vidéo en pixels
width = largeur de la vidéo en pixels
bitrate = le débit ou bitrate (en anglais) d'un flux vidéo ou audio numérique sert à quantifier la quantité de données transmises par seconde. Plus le débit est élevé, meilleure est la qualité de la vidéo. Les unités les plus usuelles de bitrate sont le Kbps et le Mbps.
frame rate = image par seconde (fps).
i = nombre de i frames dans une vidéo.
p = nombre de p frames dans une vidéo.
b = nombre de b frames dans une vidéo.
Frames = nombre de frames dans une vidéo.
i_size = taille totale en octet des i dans une vidéo.
p_size = taille totale en octet des p dans une vidéo.
b_size = taille totale en octet des b dans une vidéo.
Size = taille totale de la vidéo.

La classe 2 :

o_codec = codec de sortie utilisé pour le transcodage.
o_bitrate = débit de sortie utilisé pour le transcodage.
o_framerate = nombre d'images par seconde utilisé pour le transcodage (nouveau fps obtenu).
o_width = largeur de sortie en pixel utilisée pour le transcodage (la nouvelle largeur qu'on veut avoir).
o_height = hauteur de sortie en pixel utilisée pour le transcodage (la nouvelle hauteur qu'on veut avoir).

La classe 3 :

umem = mémoire totale allouée par codec pour le transcodage.
utime = temps total de transcodage.

Nous prenons un tuple quelconque de notre data set pour voir les valeurs de ces différents attributs cités au-dessus.
Nous commençons d'abord par les attributs de **la classe 1** qui sont introduit en entrer :

id	duration	codec	width	height	bitrate	framerate
04t6-jw9czg	130.35667	mpeg4	176	144	54590	12

i	p	b	frames	i_size	p_size	b_size	size
27	1537	0	1564	f(	825054	0	889537

Ensuite, ceux de **la classe 2 :**

o_codec	o_bitrate	o_framerate	o_width	o_height
mpeg4	56000	12	176	144

Enfin, ceux de **la classe 3 :**

umem	utime
22508	0.612

Nous remarquons que dans ce tuple on veut augmenter le débit c'est-à-dire augmenter la quantité de données transmise par seconde, plus généralement cela veut dire augmenter la qualité de la vidéo (la valeur de o_bitrate est différente de la valeur d'entré, et tout le reste est similaire). Ce changement nous a couté 225008 unités de ressource ou plus précisément de mémoire allouée dans un intervalle temporel de 0,612 secondes.

Les erreurs et anomalies existants dans le data set :

Dans un data set aussi énorme que le nôtre, la probabilité de rencontrer des champs vides ou incohérents est très grande, et vérifier les instances une à une, n'est pas une solution envisageable, c'est pourquoi nous allons seulement faire des suppositions des erreurs possibles et existantes dans notre data set.
Par exemple dans le deuxième fichier « **transcoding_mesurment.tsv** » :

Exemple 1 :

04t6-jw9czg	130.35667	mpeg4	176	144	54590	12	27
1537	0	1564	f(	825054	0	889537	mpeg4

Dans la figure ci-dessus nous avons i_size = f(, et cela est une erreur, cette valeur doit être numérique.

Exemple 2 :

04t6-jw9czg	130.35667	.	176	144	5

Dans la figure ci-dessus nous avons codec = '.', et cela est une erreur, cette valeur doit être numérique.

Comment régler ces problèmes ?

Nous allons utiliser des méthodes de pré traitement des données pour corriger ces erreurs-là, notamment dans le cas où le contenu n'est pas représentatif du contenu que doit porter un attribut

1. Retrouver les cases non remplies, ou remplies de manière non conformes à leurs sens.
2. Utiliser une méthode telle que « le binning » (nous avons utilisé d'autres méthodes mais nous ne pouvons pas les citer toutes à cause de nombre de pages limitée).
3. Nous allons maintenant remplacer le contenu de l'attribut avec le résultat engendré de la méthode de prétraitement utilisée.
   Contenu de **i_size** avant le pré traitement :

f(

   Contenu de **i_size** après le pré traitement :

64483

Explication :

Sachant que nous avons utilisé la méthode bininng pour enlever les Noisy-data ainsi que les valeurs vides concernant les instances de l'attribut **i_size**.
La méthode de binning consiste à créer des bins de valeurs égaux (equal-frequency), dans notre cas nous n'avons pas utilisé toutes les valeurs des 68000 instances, nous avons juste exporté 10 instances pour expliquer notre exemple :
remarque :(il faut trier les valeurs d'abord).

Bin 1 : 64482, 644483, « valeur manquante ».
Bin 2 : 64522, 64525, 64526.
Bin 3 : 64542, 64543, 64544.
En appliquant le « smothing by bin means », on remplacera chaque bin par la moyenne des trois valeurs
Bin 1 : 64483, 64483, 64483.
Bin 2 : 64523, 64523, 64523.
Bin 3 : 64544, 64544, 64544.

Conclusion :

Après avoir présenté notre problème avec les différentes caractéristiques de notre data set, nous avons conclu la première partie qui a pour but comprendre notre thème ainsi que les problèmes rencontrés durant le traitement de données.
Nous allons dans la 2ème partie de ce projet de continuer le pré traitement de données qui n'est pas encore complet dans cette partie, et on entamera après l'étape d'implémentation de notre modèle d'apprentissage.

Classification: Activity Recognition system based on Multisensor data fusion (AReM) Data Set

Présentation du problème :

Qu'est-ce qu'un système de reconnaissance d'activité ?
C'est un système de détection qui permet de reconnaitre les différentes activités et mouvements exercées par un utilisateur, à travers des dispositifs pour l'extraction d'un ensemble de données relative au mouvement pratiqué, et qui sont des capteurs. Récemment, la reconnaissance des activités de la vie quotidienne se repose sur la classification en temps quasi-réel des données sensorielles à l'aide d'un ou plusieurs capteurs.
Qu'est-ce qu'un capteur / multi-capteur (d'activité) ?
C'est un outil / composant / dispositif qui peut être filaire ou sans fils, qui est capable de reconnaitre automatiquement le comportement d'une personne, ou de détecter l'interaction entre cette dernière et son environnement.
Dans notre projet, un réseau de capteurs sans fils (WSN [i]) est utilisé, qui s'agit d'un réseau ad hoc composé d'un ensemble de nœuds interconnectés entre eux, placés dans ou devant l'emplacement contrôlé voulu, ou chaque nœud est capable de détecter les mouvements physiques, la chaleur, … et il peut communiquer avec son voisin afin de transférer l'ensemble de données mis en sortie vers la station de surveillance ou l'unité de gestion, cette communication est gérée avec un protocole choisi.
Qu'est-ce qu'une fusion des données ?
C'est la combinaison et l'intégration d'un ensemble de données reçus depuis une ou plusieurs sources, afin de fournir une information résultante de qualité et facile à utiliser qui correspond à l'ensemble des données fusionnées en entrée.
Qu'est-ce qu'un flux RSS [ii] ?
C'est une mesure de la puissance en réception d'un signal reçu d'un émetteur, la valeur RSS peut être utilisée pour estimer la distance entre l'émetteur et le récepteur.

Principe :

La reconnaissance des activités joue un rôle clé dans la fourniture d'une assistance aux activités et de soins aux utilisateurs, et ce projet a comme but final la reconnaissance de l'activité exercée par un utilisateur.
Ce travail présente un système de reconnaissance d'activité qui classe en temps quasi-réel un ensemble d'activités quotidiennes exercée par l'utilisateur qui sont : **pliant (2 types sont également distingués), debout, assis, couché, cyclisme, marche**, en exploitation des données récupérées à partir de l'ensemble de dispositifs de capteurs sans fil usés (WSN), conformément à l'annexe technique du concours EvAAL, ce système utilise les informations provenant du changement et la modification implicite du canal sans fils due aux mouvements de l'utilisateur, afin de parvenir à une classification efficace et réactive.

L'exploitation et la récupération des données se fait à l'aide des nœuds IRIS, qui sont placés sur : **la poitrine** et **les chevilles** de l'utilisateur (Figure 1).

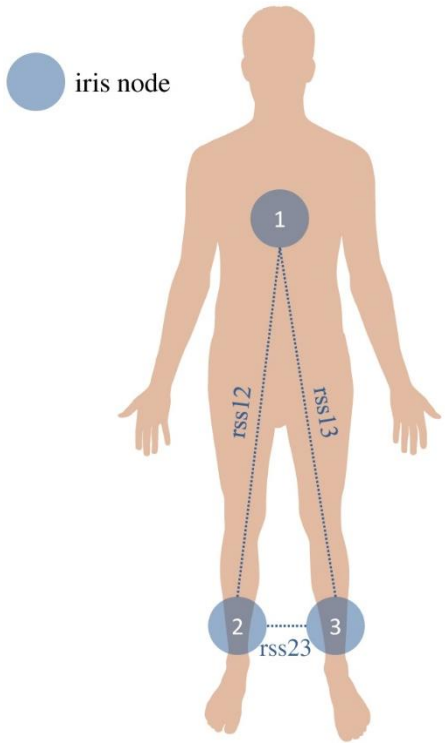

Figure 1.a

Figure 1.b

Figure 1 : Emplacement des capteurs

Ces capteurs / nœuds mesurent le RSS et le communiquent entre eux qui seront utilisé pour la récupération des différentes changements réalisées aux positions / emplacements choisis au début, et qui sont géré à travers un sous-système radio Chipcon AT86RF230 qui implémente le standard IEEE 802.15.4 qui un protocole de communication destiné aux réseaux sans fils, programmé avec un micrologiciel TinyOS conçu pour les réseaux de capteurs sans fil.

Afin d'organiser la communication et l'échange entre les capteurs un simple protocole est utilisé, qui est le protocole de jeton virtuel qui sera donné à un capteur et qui termine son exécution dans délai de 50 millisecondes, et cela veut dire que un seul nœud qui peut transmettre une donnée à la fois, et tous les autres nœuds reçoivent le paquets et effectuent les différentes traitement nécessaires (Figure 2), ce protocole est utilisé pour planifier l'ensemble des paquets et des données échangés entre les capteurs afin de :

- Bien organiser et structurer la communication et l'échange.
- Assurer une forte communication et taux de collecte de donnée élevé.
- Empêcher les collisions ou confusion entre les différentes données échangées.

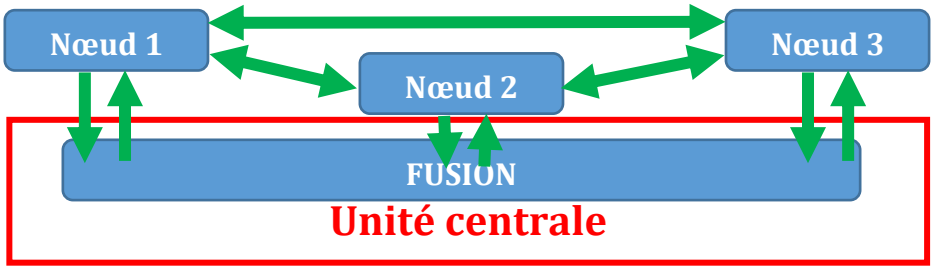

Figure 2 : Fusion et échange de données

La période de temps est fixée à 250 millisecondes selon l'annexe technique de EVAAL, dans un tel créneau horaire, nous élaborons 5 échantillons de RSS (échantillonnés à 20 Hz) pour chacun des trois couples de nœuds WSN, (la cheville droite, cheville gauche), (cheville gauche, poitrine) et (poitrine, cheville droite).

Il y'aura une extraction de caractéristiques de domaine temporelle à partie des données brute afin d'éliminer légèrement le bruit et les corrélations et compresser la série temporelle. Les caractéristiques incluent la valeur moyenne et l'écart type pour chaque lecture de RSS réciproque à partir des capteurs WSN usés.

Objectifs :

L'objectif de ce système est la reconnaissance d'activité des utilisateurs afin de fournir une assistance aux activités et de soins aux utilisateurs des maisons intelligentes, ce système qui classe en temps quasi-réel un ensemble d'activités quotidiennes communes afin de reconnaitre et de détecter l'activité exercée à un temps donné, en exploitant à la fois les données des capteurs réalisées par l'utilisateur, et les valeurs réciproques de l'intensité du signal reçu (RSS) provenant de dispositifs de capteurs sans fil usés (WSN).

Présentation des données :

Nous avons opté à choisir ce jeu de données, car c'est un sujet qui nous permet de bien comprendre et appliquer la tâche de classification, en utilisant l'ensemble de donnée d'entrée fournis, en faisant l'ensemble de traitement nécessaires afin d'avoir les bons résultats, sans oublier de dire que c'est un sujet très intéressant à réaliser qui combine entre les différents domaines.
Ce système de reconnaissance d'activité a comme objectif établissement d'une classification efficace et réactive en temps quasi-réel des activités exercées par l'utilisateur afin de bien classer les activités, notre système va considérer 7 activités différentes (Figure 3) :

1. Debout
2. Assis
3. Couché
4. En marchant
5. Faire du vélo
6. Penché (se pencher) en pliant les jambes
7. Penché (se pencher) sans pliant les jambes

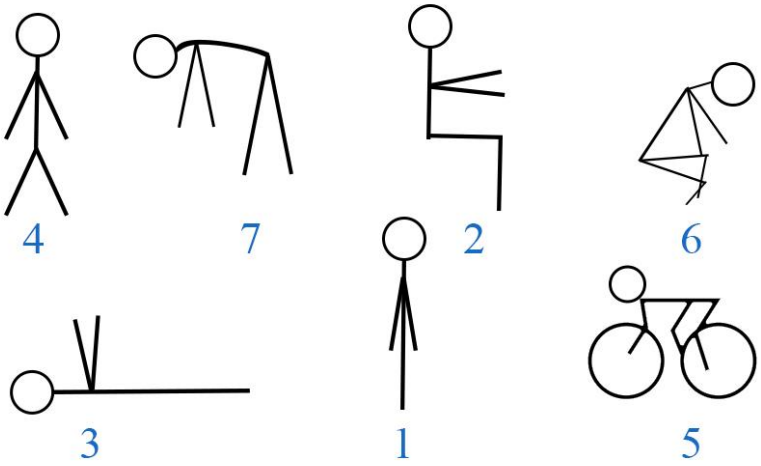

Figure 3 : Les positions reconnues

Notre data set contient des instances pour chaque activité exercée vue, et donc notre système considère pour chaque activité 15 séquences temporelles de donnée RSS d'entrée (15 séquences pour les activités 1,2,3,4 et 5, l'activité 6 contient 6 séquence et l'activité 7 contient 7 séquences), d'où chaque data set d'une activité contient 480 séquences, pour un nombre total de 42240 instances, pour chaque séquence, les données sont fournies au format CSV, et pour chaque activité nous avons les paramètres suivants :

- Task représente l'activité effectuée
- Frequency (Hz) : 20 représente la fréquence d'échantillonnage
- Clock (millisecondes) : 250 représente l'horloge, et c'est la période d'échantillonnage
- Duration (seconde) : 120 représente la durée totale.

Les attributs de data set :

Données d'entrée : Notre data set est composé d'un ensemble de fichier CSV, dont chaque fichier :

- Est nommé datasetID.csv, ou ID est l'ID de séquence numérique progressive pour chaque répétition de l'activité exercée.
- Contient un ensemble d'instances dont chaque instance (ligne) correspond à une mesure de pas de temps dans l'ordre temporelle et contient les informations suivantes :
  - avg_rss12 : correspond à la moyenne de RSS entre le nœud 1 et nœud 2.
  - var_rss12 : correspond à la variance de RSS entre le nœud 1 et nœud 2.
  - avg_rss13 : correspond à la moyenne de RSS entre le nœud 1 et nœud 3.
  - var_rss13 : correspond à la variance de RSS entre le nœud 1 et nœud 3.
  - avg_rss23 : correspond à la moyenne de RSS entre le nœud 2 et nœud 3.
  - var_rss23 : correspond à la variance de RSS entre le nœud 2 et nœud 3.

  Où chaque valeur est une mesure de RSS relatif au paquet échangé entre un nœud A et un nœud B.

Données cibles : sont fournies sous le nom du dossier contenant, qui est le nom de l'activité effectuée, qui contient l'ensemble des fichiers data set relatif à cette activité.

Prétraitement de la base de données :

Notre base de données qui contient 42240 instances doit être passer par un ensemble d'étapes qui consiste à nettoyer les données, structurer les données …, afin d'améliorer la qualité de base de données, et de clarifier sa structure par la suite aux algorithmes de data mining.

Etapes de prétraitement, problèmes et solutions de data set :

1. Certain fichier de data set relatif à l'activité possède des duplications de valeurs d'enregistrement pour des périodes de temps successive, par exemple : dataset1, activité « binding1 », une instance manquante à time = 27250 et time = 27500 :

   27250,42.00,0.00,18.00,0.00,35.25,1.30
   27500,42.00,0.00,18.00,0.00,35.25,1.30

   Trois explications sont associées à cet état :
   - Soit l'utilisateur n'a pas changé sa position par rapport au premier enregistrement.
   - Soit le capteur n'a pas enregistrés la nouvelle valeur
   - Soit les deux mouvements sont très assimilé, de là ils possèdent les mêmes valeurs.

   Avoir tel tuple dans le data set de la mêmes activité (même signification) ne vas pas apporter une amélioration (information non pertinente) pour le model pour la raison que le temps n'est pas pris comme facteur pour l'apprentissage.

   → **La solution proposée** consiste a ignoré (supprimer) tel tuple du data set.
2. La base de données de notre modèle est composée de plusieurs fichiers CSV répartie dans 7 dossiers selon l'activité signifié. Pour entrainer le model sur notre base de connaissance il est impératif de rassembler l'ensemble des fichiers regroupant l'ensemble des données qui constitue le data set pour avoir une seule base de connaissance sur laquelle le model apprendra. A ce niveau un problème de redondance de données peut apparaitre dans certain tuple relatif à la même activité.

   → **La solution proposée** consiste a ignoré (supprimer) tel tuple du data set.
3. Tous les ficher de data set possèdes le même nombre d'attribues avec même nom : donc le problème de nommage qui intervient souvent dans le rassemblement des fichiers de data set n'apparait pas
4. Le data set possède un nombre important d'instance (42240) le flux généré par se nombre important peut mener à un over fitting si toutes les instances seront utilisées pour l'apprentissage.

   → **La solution proposée** consiste à réduire la taille des données ainsi devisées le data set entre les données de training et testing (80% ,20% respectivement), les données seront sélectionnées d'une manière aléatoire en utilisant une fonction Radom.
5. Certain tuples continents des valeurs manquante pour certain attribue (missing data), par exemple : dataset8, activité « sitting », une instance manquante à time = 13500.

→ **La solution proposée** consiste a complété les valeurs manquantes de l'attribue par la moyenne de cet attribue de la même classe.

6. Les classes (targets) que model traité sont présenté dans le nom de dossier pas parmi les attribues de data set. De ce fait le model ne pas apprendre avec l'absence des classe et la base de données avec tel présentation des données n'auront aucun sens (aucun apport à l'apprentissage)

   → **La solution proposée** consiste a ajouté dans notre data set une colonne **label** qui contient les classes représentant les valeurs des activités.

   Ainsi ces classes doivent être représenté d'une manière à rendre l'apprentissage plus facile a effectué, pour mener à ce but les valeurs des classes seront transformé en données numérique ou la valeur de chaque classe (activité) serra prédéfinie par nous-mêmes. Et appliqué pour toute l'instance de mêmes valeurs.

   Le codage choisis et le suivant :

   - Penché (se pencher) sans pliant les jambes → 1
   - Penché (se pencher) en pliant les jambes → 2
   - Faire du vélo → 3
   - Couché → 4
   - Assis → 5
   - Debout → 6
   - En marchant → 7

→ **Proposition pour réduire le nombre d'instance :** Nous avons un nombre important d'instances, avec des petites différences entre les instances qui tend des fois vers 0, par exemple : dataset1, activité bending1 :

26250,42.00,0.00,18.75,0.43,35.00,1.22
26500,42.00,0.00,18.00,0.00,35.00,1.41

On a un nombre important d'instances qui sont presque identiques, et qui se ressemblent. Et comme solution on pourra utiliser clustering la création des clusters, et on remplace les valeurs d'un cluster par la valeur représentante, et comme ça on pourra réduire le nombre d'instances et de combiner entre plusieurs instances afin d'avoir de résultat clair

Répartition des tâches :

Nous avons travaillé en utilisant un système appelé **PEER REVIEW** et **PEER PROGRAMMING**, qui consiste en ce que le travail se code à deux, et aussi à chaque fois qu'un monôme ou binôme effectue sa partie de travail demandé, cette partie-là sera vérifiée et jugé ainsi que rectifié par un autre monôme ou binôme.

KRIM Islam	MASDOUA Manil
➢ Elaboration de la partie conclusion de la partie de régression. ➢ Etude de la data set « transcoding_mesurment ». ➢ Etude de la data set « youtube_videos ». ➢ Description du sujet de régression. ➢ Détection des anomalies des deux data sets, et réflexion sur la méthodologie de neutralisations des anomalies et erreurs présentes. ➢ Vérification du travail de la régression globalement et, plus spécifiquement sur les définitions des attributs ainsi que la détection et corrections des anomalies trouvées.	➢ Elaboration de la partie introduction de la partie de régression. ➢ Etude de la data set « transcoding_mesurment ». ➢ Etude de la data set « youtube_videos ». ➢ Description du sujet de régression. ➢ Détection des anomalies des deux data sets, et réflexion sur la méthodologie de neutralisations des anomalies et erreurs présentes. ➢ Vérification du travail de la régression globalement et, plus spécifiquement sur les définitions des attributs ainsi que la détection et corrections des anomalies trouvées ➢ Elaboration de la conclusion générale

AKKOUCHE Abderrahmane	SASSI Kahina
➢ Elaboration de la partie description du sujet : • Définition : d'un système de reconnaissance d'activité/d'un multi-capteur / flux RSS • Recherche et élaboration du prototype de l'expérience / objectif. ➢ Elaboration de la partie traitement de data Set : • Présentation data set et ces caractéristiques/ présentation de l'ensemble des données. ➢ Elaboration de la partie pré traitement des données : • Recherche des caractéristiques de data set /détection des problèmes • Proposition des solutions associées ➢ Rassemblement des 2 parties ➢ Détection des anomalies des deux data set, et réflexion sur la méthodologie de neutralisations des anomalies et erreurs présentes. ➢ Vérification du travail de la régression globalement et, plus spécifiquement sur les définitions des attributs ainsi que la détection et corrections des anomalies trouvées.	➢ Elaboration de la partie description du sujet : • Définition : d'un système de reconnaissance d'activité/d'un capteur / Fusion de données (signal) multi capteur • Recherche et élaboration du prototype de l'expérience / objectif. ➢ Elaboration de la partie traitement de data Set : • Présentation data set et ces caractéristiques/ présentation de l'ensemble des données. ➢ Elaboration de la partie pré traitement des données : • Recherche des caractéristiques de data set /détection des problèmes • Proposition des solutions associées ➢ Vérification de la cohérence de l'enchainement ➢ Détection des anomalies des deux data set, et réflexion sur la méthodologie de neutralisations des anomalies et erreurs présentes. ➢ Vérification du travail de la régression globalement et, plus spécifiquement sur les définitions des attributs ainsi que la détection et corrections des anomalies trouvées. ➢ Elaboration de l'introduction générale.

Conclusion :

Ce travail a pour objectif l'établissement d'une étude approfondie des bases de connaissance sélectionnées pour l'application de l'apprentissage. Ce vif travail nous as permis d'avoir une vision plus approfondie dans le prétraitement des données de data mining qui représente une phase pertinente dans l'apprentissage automatique, et l'occasion de s'affronté dans l'esprit de travail en équipe.
Dans ce travail nous avons commencé par établir une présentation des sujets et des systèmes choisis, tout en spécifiant les caractéristiques et les détaille des systèmes. Ensuite, nous avons établi une étude détaillée sur le data set de chaque système en élaborant une description générale de la base de connaissance, les caractéristiques des attribues ainsi que les différents problèmes envisagés dans la base pour finir par la proposition des solutions au problème.

Bibliographies:

1. Human Activity Recognition from Accelerometer Data Using a Wearable Device (web : https://www.researchgate.net/publication/221258784_Human_Activity_Recognition_from_Accelerometer_Data_Using_a_Wearable_Device)
2. https://www.researchgate.net/figure/Classification-basee-sur-la-fusion-de-signal-Cependant-cette-approche-a-un-inconvenient_fig2_304929002/download
3. Human activity recognition using multisensor data fusion based on Reservoir Computing, Journal of Ambient Intelligence and Smart Environments, 2016, Volume 8, issue 2
4. https://www.vdocipher.com/blog/what-is-transcoding/?fbclid=IwAR2DH3wfkwwKo_rbzSe3JZ2XTIfohJkIekYVM9cTWkOeSC_RyiENzfwAMQM

[i] Wireless Sensor Network : Réseau de capteurs sans fil
[ii] Received Signal Strength : Puissance du signal reçu