



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

Curso Académico:

A Pascual Ibor Medina.

AGRADECIMIENTOS

En primer lugar, deseo agradecer toda la ayuda prestada durante la realización del trabajo a mi tutor, Juan Manuel Belda Lois. Sin su dedicación y esfuerzo este trabajo no hubiera sido posible. Mis ganas de aprender y superarme a diario ante los retos que presentaba el trabajo han sido fruto de la confianza y motivación que he recibido por su parte. Ha sido una experiencia satisfactoria poder llevar a cabo el trabajo en un centro como el Instituto de Biomecánica de Valencia (IBV).

También me gustaría agradecer a todos los niños, y sus padres, que aceptaron a participar en el ensayo que me permitió obtener datos para la realización del trabajo.

Me gustaría mencionar a Rosa, mi profesora de piano, a la que debo agradecer toda la paciencia y comprensión que ha tenido conmigo durante estos años, los importantes consejos que me ha dado y la disciplina que me ha inculcado. Ella ha sido fundamental para que pudiera compaginar mi hobby y mi carrera.

No puedo dejar escapar la oportunidad de dar las gracias a mis amigos, a los de siempre y a aquellos con los que he compartido estos cuatro años de carrera. Sus ánimos, cariño y compañía han sido imprescindibles durante estos años. Espero poder seguir aprendiendo y disfrutando con y de ellos durante muchos años.

En especial quisiera darle las gracias a Pablo por el enorme apoyo que he recibido por su parte, y el de su familia, por escucharme y por estar ahí cuando lo he necesitado. Ha vivido conmigo el trabajo como si fuera suyo y ha conseguido que diera lo mejor de mí en cada momento sin pedir nada a cambio, pero sobretodo, quisiera agradecerle el bonito equipo que hemos formado juntos.

Mis últimos agradecimientos son para mi familia: mis abuelos, mis tíos, mis primos, mis padres y Kitty. La energía que transmiten para que consigas cada meta propuesta y la alegría e ilusión que muestran ante cada logro conseguido son un aliento para seguir adelante. Pero a los que de verdad les tengo que agradecer el estar escribiendo estas líneas es a las personas que más admiro, a mis padres. Gracias a ellos he aprendido a que, con constancia, esfuerzo, respeto y humildad puedes conseguir todo aquello que te propongas. Son mis referentes en la vida y ojalá algún día pueda parecerme a ellos.

RESUMEN

El incremento de la obesidad infantil ha suscitado, en los últimos años, el auge de técnicas de valoración de la actividad física, que es uno de los factores determinantes en la misma. Una de las técnicas punteras, en este campo, es la acelerometría. Este tipo de técnicas permiten la obtención de datos de intensidad, sin embargo, no permiten distinguir entre tipos de actividad. Los acelerómetros MEMS utilizados en estas técnicas, son capaces de recoger la información registrada y transmitirla a otro dispositivo, como una computadora. Con estos datos se pueden crear algoritmos de clasificación, que analizando los datos de entradas (aceleraciones), consiguen que a su salida se prediga el tipo de actividad que se está llevando a cabo.

Bajo estas premisas, el objetivo del trabajo ha sido desarrollar un algoritmo capaz de clasificar las actividades infantiles de gateo, marcha y carrera, mediante las señales obtenidas a través de un acelerómetro triaxial.

Para su creación se han empleado técnicas estadísticas cuya principal característica es la extracción de información de las señales obtenidas a través del acelerómetro. Una vez obtenida la información, estas técnicas también son capaces de averiguar la relación existente entre los datos. Para la realización del algoritmo, la técnica seleccionada ha sido la discriminante lineal. La misma predice como influyen cada una de las variables que diferencian a los distintos grupos que se han determinado para la clasificación. Durante la realización del trabajo se han extraído la media y la desviación típica en las tres direcciones del espacio (X, Y, Z), y después de examinar todos los datos, el modelo ha determinado los coeficientes de influencia de cada una de ellas en la marcha, gateo y carrera.

Tras la creación del algoritmo, se ha utilizado una validación cruzada. En este tipo de validación los datos se dividen en dos grupos de datos, uno de ellos se utiliza para ajustar el modelo, y el otro para realizar un testeo. Considerando que el número de datos obtenido en este estudio no es elevado, el método de validación cruzada seleccionado para el mismo ha sido "Leave-one-out". Su funcionamiento está basado en seleccionar todos los datos excepto uno para ajustar el modelo, y posteriormente el dato excluido se utiliza para realizar el testeo. El testeo consiste en realizar la predicción para el valor excluido y una vez obtenida la misma se compara con el valor real de este dato. Este proceso se repite hasta que todos los datos hayan pasado por testeo. De esta forma, se consigue que haya un mayor número de ensayos y que la capacidad del algoritmo de generalizar aumente. Para evaluar la capacidad de generalización y por tanto su calidad, una vez validado, se han obtenido los siguientes indicadores: la matriz de confusión o error y el parámetro de exactitud.

Los resultados obtenidos en la evaluación del algoritmo creado muestran que su poder discriminatorio es óptimo en comparación a resultados obtenidos con algoritmos de cualidades similares.

Palabras clave: IMUs, clasificador, niños, marcha, carrera, gateo

RESUM

L'increment de l'obesitat infantil ha suscitat, en els últims anys, l'auge de tècniques de valoració de l'activitat física, que és un dels seus factors determinants. Una de les tècniques capdavanteres, en este camp, és l'accelerometria. Este tipus de tècniques permeten l'obtenció de dades d'intensitat, no obstant això, no permeten distingir entre els tipus d'activitat. Els acceleròmetres MEMS utilitzats en estes tècniques, són capaços de recollir la informació registrada i transmetre-la a un altre dispositiu, com una computadora. Amb estes dades es poden crear algorismes de classificació, que analitzant les dades d'entrades (acceleracions), aconseguen que a la seua eixida es prediga el tipus d'activitat que s'està duent a terme.

Sota estes premisses, l'objectiu del treball ha sigut desenvolupar un algorisme capaç de classificar les activitats infantils del gateig, marxa i carrera, mitjançant els senyals obtinguts a través d'un acceleròmetre triaxial.

Per a la seua creació s'han emprat tècniques estadístiques que tenen com a característica principal l'extracció d'informació dels senyals obtinguts a través de l'acceleròmetre. Una vegada obtinguda la informació, estes tècniques també són capaces d'esbrinar la relació existent entre les dades. Per a la realització de l'algorisme, la tècnica seleccionada ha sigut la discriminació lineal. Esta prediu com influeixen cadascuna de les variables que diferencien als diferents grups, que s'han determinat per a la classificació. Durant la realització del treball s'han extret la mitjana i la desviació típica en les tres adreces de l'espai (X, Y, Z), i després d'examinar totes les dades, el model ha determinat els coeficients d'influència de cadascuna d'elles en la marxa, gateig i carrera.

Després de la creació de l'algorisme, s'ha utilitzat una validació creuada. En este tipus de validació les dades es divideixen en dos grups de dades, un d'ells s'utilitza per ajustar el model, i l'altre per realitzar un testeig. Considerant que el nombre de dades obtingut en este estudi no és elevat, el mètode de validació creuada seleccionat ha sigut "Leave-one-out". El seu funcionament està basat a seleccionar totes les dades excepte una per ajustar el model, i posteriorment la dada exclosa s'utilitza per realitzar el testeig. Este consisteix a realitzar la predicció per al valor exclòs i una vegada obtinguda es compara amb el valor real. Este procés es repeteix fins que totes les dades s'hagen passat per el teste. D'esta forma, s'aconsegueix que hi haja un major nombre d'assajos i que la capacitat de l'algorisme per generalitzar augmente. Per avaluar la capacitat de generalització i per tant la seua qualitat, una vegada validat, s'han obtingut els següents indicadors: la matriu de confusió o error i el paràmetre d'exactitud.

Els resultats obtinguts en l'avaluació de l'algorisme creat mostren que el seu poder discriminatori és òptim, en comparació a resultats obtinguts amb algorismes de qualitats similars.

Paraules clau: IMUs, classificadors, xiquets, marxa, carrera, gateig

ABSTRACT

In the recent years, the increasement in childhood obesity has cause the rise of techniques for the assessment of physical activity. This is due to the fact that physical activity is one of the determining factors in obesity. One of the leading techniques in this field is accelerometry. Although this technic allows obtaining intensity data, it does not allow to distinguish between types of activity. The MEMS accelerometers are used to solve this issue. These ones have the capacity of collecting the recorded information and transmitting it to another device, such as a computer. With this data, classification algorithms can be created. After its creation, they can analyze the input data (accelerations) and afterwards get a prediction, at its output, of the type of activity that is being carried out.

Under these premises, the objective of these work has been to develop an algorithm capable of classifying children's activities, through the signals obtained through a triaxial accelerometer. The activities that were able to distinguish were: crawling, running and gait.

For its creation, statistical techniques have been used. Their main characteristic is the capability of extracting information from the signals obtained through the accelerometer and then finding out the relationship between data. The selected technique for the work has been the linear discriminant analysis. This supervised method predicts how the variables selected for the classification can differentiate between groups and which is its influence. During the work, the mean and the standard deviation were extracted in the three directions of the space (X, Y, Z), and after examining all the data, the model was able to determine the coefficients of influence of each of three activities studied.

After the creation of the algorithm, a cross-validation was used. In this type of validation, the data is divided into two groups, one of them is used to adjust the model, and the other is used for testing. Considering that the number of data obtained in this study is not high, the method of cross validation selected for it has been "Leave-one-out". The way it proceeds is: it takes all data except one to adjust the model, and subsequently the excluded data is used to perform the test. The test consists of making the prediction for the excluded value and once it is obtained, it is possible to compare it with the real value. This process is repeated until all the data has gone through testing. In this way, there is a larger number of iterations and the ability of the generalization of the algorithm is increased. Once it is validated, the confusion or error matrix and the accuracy parameter were obtained. These indicators make possible the evaluation of the algorithm and it capacity of generalizing.

The results obtained in the evaluation of the algorithm created show that its discriminatory power is optimal compared to results obtained with algorithms with similar qualities.

Key words: IMUs, classifier, children, gait, running, crawling

Tabla de contenido

CAPÍTULO I: INTRODUCCIÓN	14
1.1 MOTIVACIÓN	14
1.2 OBJETIVOS	15
1.4 PLAN DE TRABAJO	16
CAPÍTULO II: ESTADO DEL ARTE	17
2.1 Clasificadores	17
2.1.1 Aprendizaje automático	17
2.1.2 Algoritmos de clasificación	17
2.1.3 Técnicas de clasificación supervisada.....	18
2.2 Clasificación de la actividad	20
2.2.1 Instrumentos usados	20
2.2.2 Clasificación de la actividad en adultos	21
2.2.3 Clasificación de la actividad en niños	24
CAPÍTULO III: MATERIAL Y MÉTODOS	26
3.1 Diseño experimental	28
3.1.1 Elección de la muestra.....	29
3.1.2 Realización de los experimentos.....	29
3.2 Tratamiento de los datos	33
3.2.1 Registro de los datos	34
3.2.2 Procesamiento	35
3.2.3 Análisis realizado.....	38
CAPÍTULO IV: RESULTADOS Y DISCUSIÓN	46
4.1 Descripción de los datos obtenidos	46
4.1.1 Número de señales.....	46
4.1.2 Longitud de las señales.....	49
4.1.3 Medidas perdidas.....	49
4.2. Clasificación de la actividad	50
4.2.1 Descripción de los clasificadores.....	50
4.2.2 Validación de los clasificadores.....	57
CAPÍTULO V: CONCLUSIONES	59
CAPÍTULO VI: PRESUPUESTO	60
5.1 PRESUPUESTO	60
5.1.1 MANO DE OBRA	60
5.1.2 MATERIAL	61
5.1.2.1 Material para el ensayo	61
5.1.2.2 Material para el estudio y desarrollo del proyecto	62
CAPÍTULO VII: BIBLIOGRAFÍA	64
CAPÍTULO VIII: ANEXOS	70

CAPÍTULO I: INTRODUCCIÓN

En este primer capítulo se expone una introducción para situar en contexto el desarrollo del trabajo.

1.1 MOTIVACIÓN

La obesidad y sobrepeso infantil son uno de los problemas de salud pública más importantes del siglo XXI. Según la OMS (Organización Mundial de la Salud), se prevé que el número de lactantes y niños pequeños con sobrepeso aumentará de 41 millones establecidos en 2016, a 70 millones para 2025. Además, existe una clara tendencia por parte de estos niños a continuar con los malos hábitos a medida que crecen, manteniendo estos problemas hasta llegar a la edad adulta. Esto provoca que tengan mayores riesgos de padecer enfermedades cardiovasculares o diabetes. Por este motivo son relevantes las estrategias para su prevención.

La realización de actividad física diaria, es un factor importante para mejorar esta situación. Debido a ello y gracias al gran avance tecnológico de los últimos años, se ha incrementado el uso de métodos objetivos para el estudio de la actividad física en niños. La herramienta más empleada para su registro y almacenamiento es la acelerometría, ya que es considerada como una de las opciones más fiables (Trost et al. 2002)(Rowl et al. 1999).

Gran parte de los estudios revisados, están centrados en detectar la realización de la actividad y el consecuente gasto energético. Esto permite conocer los requisitos energéticos diarios y posibilita el diseño de estrategias nutricionales correctas. Sin embargo, debido a la falta de material estudiado en niños, resulta más costoso poder caracterizar los patrones de movimiento básicos que les rigen e influyen en su desarrollo. Además, extrapolar los resultados obtenidos en adultos en niños no es posible ya que están regidos por patrones distintos.

Los patrones de movimiento son secuencias fijas, organizadas y estructuradas que se producen al llevar a cabo un determinado movimiento. Los patrones básicos se adquieren y desarrollan desde que nacen hasta los 5 años. Posteriormente, se adquieren patrones más complejos a medida que su aparato locomotor y sus cambios a nivel neuromuscular maduran. Un correcto desarrollo motor es significativo para conseguir una total independencia y un adecuado desarrollo psicomotriz.

En consecuencia, la creación de algoritmos de clasificación es una manera de poder distinguir entre las diferentes actividades infantiles y poder comprender así, el concepto de desarrollo motor y su influencia a lo largo de la vida. Con esta intención surge este trabajo, que pretende realizar un algoritmo de clasificación capaz de detectar, a través de medidas de acelerometría, cuando un niño está gateando, andando o corriendo.

1.2 OBJETIVOS

El objetivo general del trabajo es desarrollar a través de un análisis discriminante lineal un algoritmo capaz de clasificar las actividades infantiles de gateo, marcha y carrera, mediante las señales obtenidas a través de un acelerómetro triaxial y el uso del lenguaje de programación Python.

Para conseguirlo es necesario cumplir una serie de objetivos específicos que se pueden dividir en:

Previos al desarrollo del algoritmo:

- Estudiar el aprendizaje automático y las diferentes técnicas de clasificación supervisada.
- Realizar una revisión del estado del arte de las herramientas empleadas para la clasificación de la actividad física.
- Medir en un ensayo la actividad física de niños de 1 a 5 años a través de unos sensores inerciales, IMUs, adheridos al calzado de los participantes mientras realizan movimientos de gateo, marcha y carrera.
- Realizar un estudio estadístico, a través de la media y la desviación típica, de los datos obtenidos. Para ello, emplearemos la herramienta Python.

Posteriores al desarrollo del algoritmo:

- Validar a través del método “leave-one-out” la capacidad de clasificación del algoritmo.
- Discutir los resultados obtenidos y compararlos con los de otros estudios.
- Extraer conclusiones del trabajo.

1.4 PLAN DE TRABAJO

A la hora de realizar un proyecto es necesario marcar los tiempos en los que se va a desarrollar. Por ello, en un primer momento se planificó como iba a organizarse, los pasos que se debían seguir y el tiempo necesario para cada uno de ellos.

En concreto, para este proyecto se tuvo en cuenta que se disponía del curso académico 2017/18 y que había que pasar por 4 fases diferentes que debían de ser llevadas a cabo en diferentes momentos. Estas son:

1. Fase del ensayo. La segunda quincena de noviembre de 2017 fue la fecha establecida para realizar la parte experimental del proyecto.
2. Fase de documentación. Durante febrero de 2018, se buscó toda la información necesaria sobre el desarrollo de clasificadores. Además, se realizó una búsqueda bibliográfica sobre el tema a tratar haciendo hincapié en los estudios y trabajos que habían sido realizados hasta el momento. Esta fase sirvió para establecer y seleccionar los métodos que se iban a emplear en el desarrollo del trabajo. Con ellos claros, se pasó a la tercera fase.
3. Fase de desarrollo. La parte técnica del proyecto se llevo a acabo durante los meses de abril, mayo y junio de 2018. Esta fase se subdividió de acuerdo con los objetivos específicos establecidos. Así pues, primero se programó la selección de características; tras ello, se obtuvo el ajuste del modelo de clasificación y finalmente, se llevo a cabo la validación y evaluación del modelo. El tutor y la alumna concertaban dos días a la semana para ir revisando, mejorando y desarrollando el proyecto.
4. Fase de redacción. Una vez satisfechos con los resultados obtenidos se procedió en junio de 2018 a la redacción de la memoria del trabajo y se llevó a cabo un proceso de revisión y mejoras por parte del tutor para que la alumna consiguiera crear un trabajo adecuado.

La planificación temporal que se acaba de explicar, se resume en el siguiente diagrama de Gantt:

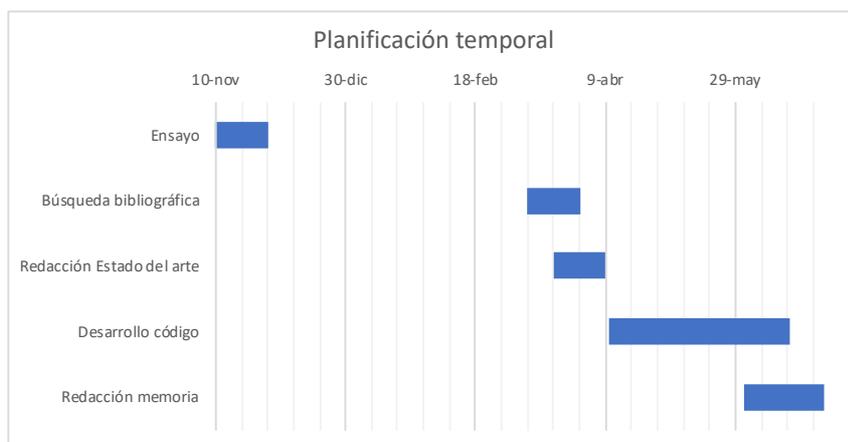


Gráfico 1. Diagrama de Gantt

CAPÍTULO II: ESTADO DEL ARTE

En el segundo capítulo, se va a realizar un análisis sobre las bases teóricas y los estudios que se han llevado a cabo hasta el momento sobre la actividad física y como clasificarla y cuantificarla.

2.1 Clasificadores

El término clasificador presenta varias acepciones, pero en el contexto de este trabajo hace referencia a aquel algoritmo que lleva implementado en su estructura la acción de discriminar. A continuación, se explica este concepto con más detalle.

2.1.1 Aprendizaje automático

El aprendizaje es un proceso que incluye la adquisición y organización de nuevo conocimiento para convertirlo en conceptos generales, el desarrollo de habilidades motoras y cognitivas a través de la práctica o del cumplimiento de instrucciones y el descubrimiento de nuevos factores y teorías a través de la observación y la experimentación. Los investigadores han intentado dotar de estas cualidades a los ordenadores gracias al desarrollo y la complejidad que han experimentado en las últimas décadas. La inteligencia artificial ha intentado y sigue intentando abordar este tema (Michalski, Carbonell, y Mitchell 2013).

El aprendizaje automático es una de las ramas de la Inteligencia Artificial basada en la construcción de algoritmos guiados por los datos («Neural Networks: Machine Learning Inspired by the Brain» 2017) con el fin de que estos aprendan por sí mismos gracias a la identificación de los patrones que se ‘esconden’ tras los datos. Es decir, trata de crear sistemas “inteligentes” capaces de aprender de su entorno a través de la observación del mismo.

Cundo uno de estos algoritmos lleva a cabo un proceso de clasificación se le conoce como clasificador. Actualmente, se usa mucho para este fin ya que, analizando los datos de estudio, el algoritmo de clasificación creado extrae relaciones entre ellos, descubriendo conocimiento y siendo capaz de discriminar en distintas clases estos datos de forma más rápida de lo que podría hacer un humano.

2.1.2 Algoritmos de clasificación

Los algoritmos de clasificación o clasificadores se encargan, mediante el uso de modelos matemáticos adecuados y técnicas de aprendizaje automático, de permitir que la computadora generalice comportamientos y reconozca patrones a partir de la información dada en forma de ejemplos. Así, para un nuevo dato desconocido este algoritmo, este será capaz de determinar a que clase corresponde. En resumen, el acto de clasificar implica transformar los datos obtenidos en conceptos.

La taxonomía de los algoritmos de clasificación se puede construir en base a varios criterios (Mannini y Sabatini 2010).

Un criterio es el conocimiento previo. En base a él, existe el aprendizaje supervisado y no supervisado. La principal diferencia entre ambas técnicas es el uso de etiquetas.

En la clasificación supervisada, el conjunto de datos disponible estará etiquetado. Así pues, las clases a las que pertenecen los datos se conocen y el objetivo será extraer una función que asigne a cada nuevo dato su clase. Para ello, el conjunto de datos con su correspondiente etiqueta será empleado por el algoritmo a modo de ejemplo para poder aprender a clasificar correctamente. Tras el entrenamiento, si se introduce un nuevo dato no etiquetado este debería ser capaz de etiquetarlo de forma correcta.

En la clasificación no supervisada, no hay un conocimiento previo, y la separación entre clases está basada en ir encontrando grupos que tengan características similares. Es útil en casos en los que el objetivo principal es descubrir relaciones implícitas en un conjunto de datos para agruparlos.

Otro criterio es por la aproximación que define la clasificación, es decir, cuales son los fundamentos teóricos por los que están regidos.

Los algoritmos pueden estar basados en procedimientos estadísticos, donde la máxima probabilidad es la clave para la clasificación o en procedimientos geométricos, en los cuales la clasificación se realiza en función de la construcción de fronteras de decisión que separarán cada clase. Los límites se construyen a través de procedimientos iterativos o consideraciones geométricas. También existen técnicas basadas en umbrales, seleccionados cuidadosamente. Estos umbrales, pueden ser características relacionada como la energía, o simplemente la variación de datos, que ayudan a discriminar entre la presencia o la ausencia de movimiento. La información contextual, también es empleada para crear algoritmos de clasificación. Finalmente, existen métodos jerárquicos en los que las decisiones se crea un método que de forma ordenada y a través de umbrales va subdividiendo los datos hasta que ya no puede realizar ninguna subdivisión.

2.1.3 Técnicas de clasificación supervisada

Los datos que va a recibir el clasificador creado en el presente trabajo van a ir acompañados de información que indica a que clase pertenecen, es decir, van a ir etiquetados. Por este motivo, la clasificación se llevará a por mecanismos supervisados. De entre la búsqueda bibliográfica sobre estos métodos se ha encontrado que los que se utilizan principalmente para la clasificación del movimiento son los siguientes:

- Análisis discriminante (Linear Discriminant Analysis)

Se trata de una técnica estadística que ayuda a identificar de entre todas las variables y muestras disponibles de los grupos, clases, a estudiar cuales son las que mejor diferencian a las distintas clases. Una vez localizadas, crea una función lineal combinación de estas variables que será la que mejor discrimine a las diferentes clases o grupos. Su fin es emplear la función como mecanismo para la clasificación de una muestra desconocida de la que se conocerá el valor de las variables, pero no, a qué clase pertenece.

- Clasificador Bayesiano simple (Naïve bayesian)

Se trata de un clasificador probabilístico que asume que los datos de aceleración obtenidos por los sensores siguen una distribución gaussiana o normal cuya media y varianza dependen del conjunto de clases. Con esa condición y basándose en las probabilidades condicionadas, la probabilidad de que un nuevo dato desconocido haya sido generado por una actividad específica puede estimarse de forma directa y sencilla. Siempre se escogerá como actividad a aquella que tenga una probabilidad máxima, sobre la base del Teorema de Bayes.

- Modelo de mezclas gaussianas (Gaussian Mixture Model – GMM)

Este método, es similar al anterior pues también está basado en buscar la probabilidad máxima, pero en este caso, usa las funciones de densidad de probabilidad de las clases y las expresa como combinaciones lineales de distribuciones Gaussianas. Una vez obtenidas estas mezclas se obtendrán las medias y las varianzas y se podrá proceder a calcular las probabilidades condicionadas.

- Máquinas de soporte vectorial (Support Vector Machines - SVM)

Estos clasificadores geométricos son empleados para distinguir entre clases que no se pueden separar linealmente. Su objetivo principal es buscar un hiperplano de decisión que separe de forma óptima las clases, encontrándose estas a su vez, lo más alejadas posible.

- K-NN vecinos próximos (KNN)

Se trata de un método que no realiza suposiciones sobre la distribución que siguen los datos ni tampoco crea un modelo tras haber aprendido. Lo que realiza este algoritmo es una clasificación de cada dato nuevo que tiene en el grupo que corresponda, según tenga K vecinos más cerca de un grupo o de otro. Es decir, calcula la distancia del elemento nuevo a cada uno de los existentes, y ordena dichas distancias de menor a mayor para ir seleccionando el grupo al que pertenecer. Este grupo será, por tanto, el de mayor frecuencia con menores distancias.

- Decisiones de árboles (Decision tree clasifier)

Esta técnica está basada en obtener un conjunto de reglas que permitan ir clasificando de forma ordenada los datos según las características que presenten. Para ello, se crea un modelo con nodos de decisión y nodos hojas. De cada nodo de decisión, salen más ramas, y cada dato irá por una rama dependiendo de sus características, hasta que no pueda dividirse más y llegué al nodo hoja que dará la clasificación final.

- Bosques aleatorios (Random Trees)

La construcción de este método está fundamentada en la combinación de árboles de decisión que analizan conjuntos de variables. De forma sencilla su manera de actuar es la siguiente: se selecciona un vector aleatoriamente, con sus valores se construyen árboles de decisión

independientes y se obtiene una predicción de cada uno de ellos, después, se lleva a cabo un proceso de ponderación a través del cálculo de la clase más votada de los árboles que se generaron para finalmente poder hacer la predicción.

- Redes neuronales artificiales (ANN)

Para el uso de este tipo de métodos se requiere de una función matemática flexible que presente relaciones complejas entre sus entradas (variables independientes) y salidas (variables dependientes). Se configura inicialmente con datos de entrenamiento y se emplea algún proceso de optimización que permita predecir salidas conocidas para un conjunto de datos de entrada. Las entradas corresponden a los datos del sensor, y las salidas a las diferentes clases de actividades.

2.2 Clasificación de la actividad

Para poder clasificar la actividad de forma objetiva es necesario contar con instrumentos tecnológicos que faciliten la captación de las intensidades con las que se llevan a cabo las mismas.

A continuación, se hace un análisis de estos instrumentos y de las técnicas empleadas con mayor frecuencia para el estudio de la clasificación de la actividad en adultos y en niños en base a los clasificadores explicados en el apartado anterior.

2.2.1 Instrumentos usados

La principal herramienta de valoración de la actividad física es la acelerometría. Esta técnica está presente en una amplia variedad de campos, sobretodo, aquellos centrados en el estudio del movimiento humano ya que ofrece la posibilidad de llevar a cabo una monitorización de la actividad física de forma objetiva a través del estudio de la aceleración.

Los acelerómetros son instrumentos que miden intensidades aplicadas a lo largo de un cuerpo, en otras palabras, estos dispositivos son capaces de medir la fuerza de inercia generada cuando una masa se ve afectada por un cambio de velocidad.

Según el contexto en el que vayan a ser utilizados, existen diferentes tipos de acelerómetros:

- Acelerómetros mecánicos. En este tipo de acelerómetros, los cambios producidos se obtienen gracias a la existencia de galgas extensométricas, que aprovechan el efecto piezo¹-resistivo para medir deformaciones.
- Acelerómetros piezoeléctricos. Como su nombre indica, estos dispositivos los cambios son notables por el uso de cristales con propiedades piezo-eléctricas. Estos cristales producen señales eléctricas al variar su estructura cristalina. Por tanto, al aplicar una fuerza sobre una masa, esta la trasladará al cristal provocando una deformación en él que se traducirá en una corriente que se podrá medir y que permitirá obtener medidas de aceleración.

¹Piezo: apretar

- Acelerómetros piezoresistivos. Este tipo de instrumentos hacen uso de un sustrato que cambia su capacidad resistiva cuando se ejerce una fuerza sobre él. Así, se pueden medir intensidades que después se pueden transformar en la obtención de aceleraciones.
- Acelerómetros capacitivos. Para la obtención de la aceleración en este tipo de instrumentos se usan dos placas condensadoras unidas a un punto fijo. Ante una aceleración aplicada, estas se desplazan cambiando la capacidad de conducción entre ellas. Esta variación de la capacidad es la que se mide.

Todos estos tipos de acelerómetros, a su vez, se pueden clasificar por las componentes de dirección en que se miden las aceleraciones. Así, existen acelerómetros mono-axiales, cuando solo miden en una dirección, bi-axiales, cuando las aceleraciones se miden a lo largo de dos ejes perpendiculares o triaxiales, cuando se miden en las tres direcciones del espacio. Este último caso resulta de gran utilidad en procesos como la medición de la actividad física, ya que las intensidades obtenidas durante su realización cambian en las tres direcciones.

Los avances tecnológicos han permitido el desarrollo de nuevos sistemas de medición de aceleración conocidos como acelerómetros micromecánicos o MEMS. Estos están formados por estructuras miniaturizadas con características mecánicas y electromecánicas diseñadas para obtener cambios en el ambiente gracias a que llevan en su interior sensores de movimiento e inerciales, además de otros tipos. Es por ello, que los MEMS son cada vez más empleados en técnicas de análisis de movimiento.

2.2.2 Clasificación de la actividad en adultos

Para el uso de los métodos de clasificación supervisada, previamente explicados, se requiere de una función matemática flexible que represente las relaciones complejas entre sus entradas (variables independientes) y salidas (variables dependientes). Inicialmente se reconfiguran con datos de entrenamiento y se emplea algún proceso de optimización que permita predecir salidas conocidas para un conjunto de datos de entrada. Las entradas corresponden a los datos del sensor, y las salidas a las diferentes clases de actividades.

A continuación, se citan (Tabla 1) y exponen diversos artículos en los que se emplean las técnicas empleadas para la clasificación de actividades en adultos en diferentes contextos. También se hace hincapié en la evaluación obtenida de estos.

Tabla 1. Cuadro comparativo de técnicas de clasificación supervisada empleadas en la predicción de actividades diarias. Se indica tanto el número de actividades clasificadas como la exactitud o proporción de clasificaciones correctas que se obtuvo.

Clasificador	Actividades clasificadas	Exactitud (%)	Referencias
GMM	8	91.3	(Allen et al. 2006)
Naive Bayesian	10	67	(Lester et al. 2005)
Naive Bayesian	8	64	(Ravi,N et al. 2012)

Naive Bayesian	6	87	(Maurer et al. 2006)
Naive Bayesian	20	52	(Bao et al. 2004)
ANN	4	83-90	(Mantjarvi,J et al 2001)
Umbral	3	90.8	(Sekine et al. 2002)
Umbral adaptativo	10	95.1	(S.H. Lee et al. 2003)
Decisión binaria	8	54	(Ravi,N et al. 2012)
Decisión binaria	10	90.8	(Karantonis et al. 2006)
KNN	7	96.2	(Lei Gao et al. 2012)
KNN	8	55	(Ravi,N et al. 2012)
KNN	6	87	(Maurer et al. 2006)
KNN	17	90	(Pirttikangas et al. 2006)
Árbol de decisión	6	87	(Pirttikangas et al. 2006)
Árbol de decisión	7	96.4	(Lei Gao et al. 2012)
Árbol de decisión	8	57	(Ravi et al. 2005)
Árbol de decisión	16	86	(Parkka et al. 2006)
Jerarquización	8	71	(Allen et al. 2006)
Jerarquización	8	82	(Parkka et al. 2006)
Jerarquización	9	83	(Ravi et al. 2005)
Red neuronal	8	82	(Ravi,N et al. 2012)
Red neuronal	17	93	(Pirttikangas et al. 2006)
SVM	7	92.7	(Lei Gao et al. 2012)
SVM	8	63	(Ravi,N et al. 2012)

El uso de un umbral permitió crear un algoritmo capaz de clasificar entre 5 diferentes actividades estáticas y 4 dinámicas (Kern, Schiele, y Schmidt 2003). Mediante el uso de un acelerómetro triaxial fijado al cuerpo en la parte trasera, se extrajeron datos de 12 sujetos. En el estudio se

comparó el uso del umbral con o sin adaptabilidad, resultando que con el último caso se obtenía una mayor exactitud, con un valor de 95.1%.

Los árboles de decisión binarios fueron empleados (Mathie et al. 2004) en un estudio con el fin de clasificar movimientos realizados en la vida diaria a través de acelerómetros triaxiales. En él, los movimientos eran divididos en clases y subclases en diferentes niveles jerárquicos. La clasificación general se daba en los niveles más altos mientras que los detalles de la clasificación se daba a niveles más bajos. Por tanto, primero se dividía en movimiento y reposo para luego más tarde distinguir entre caídas, andar, transiciones posturales o estar sentado o de pie. Con estos acelerómetros triaxiales se monitorizaba a los pacientes en un laboratorio. Teniendo en cuenta cada clasificación, la sensibilidad obtenida superó el 87% y la especificidad un 94%. Con un total de 1309 de movimientos recogidos, la exactitud general del sistema, medida como el número de clasificaciones correctas, fue de un 97.7% y la especificidad de un 98.7%.

Los árboles de decisión también han sido empleados para estudiar la posibilidad del uso de teléfonos móviles para la detección de pasos y el reconocimiento del movimiento (Susi Melania et al. 2013). Obteniendo los datos de un sensor instaurado en el calzado y en la mano, crearon un algoritmo capaz de reconocer actividades diarias como andar, nadar o ir hablando por el teléfono móvil. Este era capaz de distinguir las actividades con una precisión de un 94%.

Los métodos SVM y redes neuronales artificiales han obtenido resultados satisfactorios en estudios realizados con pacientes con derrame cerebral. Por una parte, con la intención de detectar diferentes actividades realizadas por estos pacientes, con pie caído, se hizo un comparativa (Lau et al. 2007) con distintos algoritmos como SVM, redes artificiales neuronales o decisión bayesiana. SVM fue el que mejor exactitud consiguió, 84%, llegando a clasificar con un 100% de exactitud las tareas de subir o bajar escaleras. Otras actividades detectadas fueron: caminar y descender una cuesta o mantenerse de pie. Por otra parte, en otro estudio (Lei Gao et al. 2012) se compararon 5 algoritmos de clasificación para instaurar un sistema de reconocimiento de la actividad basado en sensores con múltiples acelerómetros. Los algoritmos comparados fueron: clasificador bayesiano, árbol de decisión, K-vecinos próximos, redes neuronales artificiales y SVM. Los resultados obtenidos dieron como ganador a la red neuronal artificial con una exactitud en la clasificación de un 96.8%. Detrás suyo, se situó el árbol de decisión y K-vecinos próximos con un 96.3%. El que peor exactitud obtuvo fue el clasificador bayesiano.

Un sistema mixto gaussiano (Felicity et al. 2006) fue comparado con un sistema heurístico para el reconocimiento de cinco tipos de movimientos y tres tipos de posturas. Ambos algoritmos dieron mejores resultados a la hora de comparar movimientos que posturas y el sistema GMM, en general, superó al heurístico en todas las clasificaciones.

Para la detección de caídas de personas de la tercera edad, se crearon 7 algoritmos diferentes, y se hizo un estudio comparativo de ellos (Armando Collado et al. 2016) a través de la implantación de acelerómetro triaxial ubicado en la muñeca. Estos algoritmos eran un árbol de decisión binario (C4.5), K-vecinos próximos(K-NN) (para K =1 y K=2), regresión logística, Random Forest, Naive Bayesian, y PART. Tras compararlos, se observó que C4.5, 1-NN y PART eran los que mayor precisión presentaban. Además, para aumentar su rendimiento se hacía uso de los atributos de media y desviación típica. El tamaño de las ventanas también era un factor clave a la hora de obtener una clasificación con exactitud. Teniendo en cuenta estos factores añadidos,

el mejor rendimiento se dió en 1-NN, después en PART y en último lugar C4.5, aún así, los datos se situaron por encima de un 95%.

Los algoritmos de clasificación K-vecinos más cercanos y Redes Neuronales Artificiales fueron comparados en un estudio en el que se pretendía crear un sistema capaz de reconocer en tiempo real actividades físicas como correr, saltar, caminar o realizar abdominales o lagartijas (Lara Geenkel et al. 2016). En caso de obtener una correcta validación de los mismos, estos dos algoritmos irían instaurados en un dispositivo, que contaba con un acelerómetro y un giroscopio que, a su vez, estaría colocado en la muñeca del usuario. Comparando su rendimiento se pudo concluir que, pese a que no hay grandes diferencias, las redes neuronales tienen un mayor rendimiento. En ambos casos, la exactitud en la clasificación es realmente óptima pero sus miembros afirman que para validarlos totalmente sería necesario contar con un mayor número de datos de entrenamientos.

2.2.3 Clasificación de la actividad en niños

El número de referencias bibliográficas que intentan la clasificación de la actividad en niños es menor que las referencias de clasificación de la actividad en adultos. Al introducir las palabras clave “physical activity clasiffication” en google académico el número de resultados se aproxima a los 4,5 millones, mientras que al añadir “children”, el número de resultado disminuye en 3 millones. A su vez, entre los artículos relacionados con estas palabras clave abundan aquellos en los que a través de los datos obtenido hacen únicamente una correlación con el tipo de actividad y no un algoritmo de clasificación. Esto se puede mejorar con la búsqueda avanzada y más si tienes en cuenta las edades de los niños.

A continuación, se detallan artículos destacables para la clasificación de actividad infantil y que métodos llevan a cabo.

Basado en técnicas de correlación, se llevó a cabo un estudio (SG Trost et al., 2001) que tenía el objetivo de demostrar que la inactividad física era un factor crucial para el aumento de la obesidad infantil. Para conseguirlo, compararon los patrones de actividad física y los determinantes psicosociales y ambientales hipotéticos de esta en una muestra variada de niños obesos y no obesos.

Para caracterizar las actividades infantiles, hay un estudio (Maurice et al., 2004) en el que se empleó como dato discriminatorio la predicción del gasto de energía. Este parámetro se obtenía a través de la acelerometría. En el estudio se quería validar el uso de acelerómetros con este fin utilizando para ello umbrales que categorizaban el nivel de intensidad de las actividades. Se comprobó que los acelerómetros si eran válidos. Pese a emplear técnicas método de clasificación supervisado no se creó un algoritmo de decisión.

Un algoritmo de discriminación de clasificación de actividad física (Ohkawara K et al. 2011) por eliminación de gravedad (GRPACA) fue creado con dos objetivos principales. El primero, examinar si este era capaz de discriminar entre actividades locomotoras y no locomotoras para varias actividades físicas de niños. El segundo, demostrar que este método de discriminación era útil para la estimación inmediata de las intensidades diarias de la actividad física utilizando un acelerómetro triaxial. El resultado fue exitoso para ambos objetivos.

El uso de correlaciones y determinantes demográficos, biológicos, sociales y culturales fue determinante para llevar a cabo un estudio (Daniel D. Bingham et al. 2016) en el que se pretendía relacionarlos con la actividad física infantil. Consideraban que obteniéndolos sería posible establecer estrategias de actuación y prevención para evitar un desarrollo no sano en los primeros años de vida por falta de actividad física. En este estudio, la palabra “correlacionar” hacía referencia a la situación donde una variable independiente se encontraba que estaba asociada con la actividad física en estudios transversales y, por consiguiente, su causalidad no se podía determinar; y el término “determinante” se empleaba al encontrar una asociación entre una variable independiente y la actividad física en estudios longitudinales donde se podía observar asociaciones temporales a lo largo del tiempo, aunque todavía existía un riesgo de vías de causalidad bidireccional o inversa.

Frente a la falta de bibliografía sobre algoritmos de clasificación y teniendo en cuenta que está validada la correlación entre la un desarrollo sano infantil y un estilo de vida acompañado de actividad física diaria, el trabajo presente pretende crear un algoritmo que sea capaz de clasificar las siguientes actividades físicas en niños: gateo, marcha y carrera.

CAPÍTULO III: MATERIAL Y MÉTODOS

Para el correcto desarrollo de un sistema de clasificación es necesario llevar a cabo una serie de pasos secuenciales (Gráfico 2) que se pueden dividir en dos partes principales: el diseño experimental, en el que se produce la adquisición de los datos y el tratamiento de los datos, que está destinado al registro, procesamiento y análisis de los mismos.

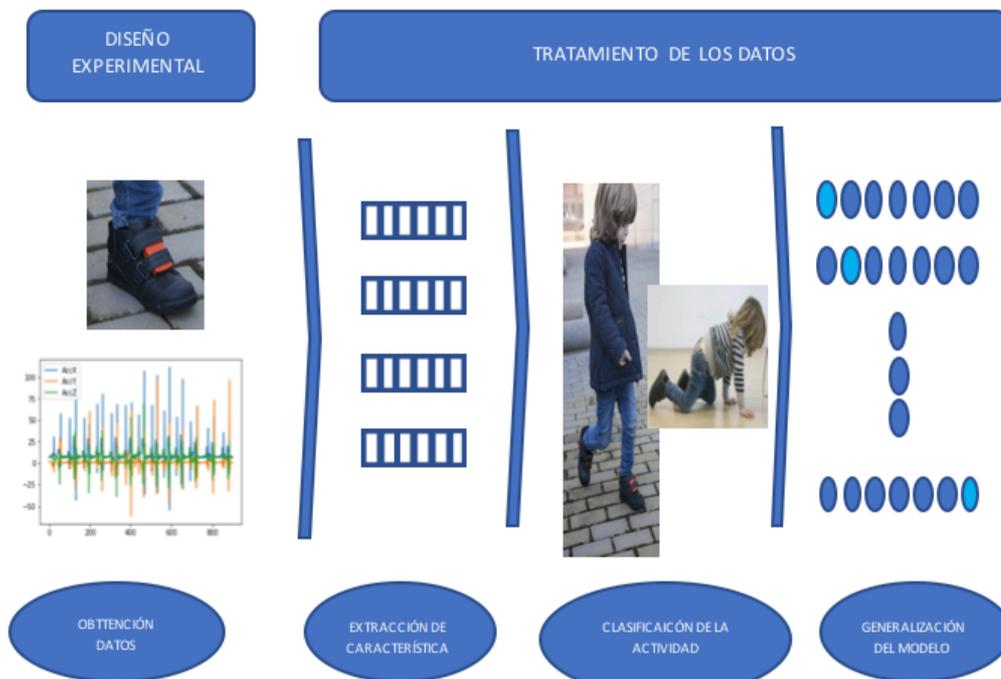


Gráfico 2. Esquema de los pasos a seguir en el desarrollo de un clasificador de actividad y los resultados obtenidos de cada uno de ellos.

A continuación, se definen cada uno de los pasos que se siguieron durante el desarrollo del trabajo:

Diseño experimental

Adquisición de datos

Gracias al uso de sensores triaxiales, se obtuvo la aceleración en los ejes X, Y, Z, para cada uno de los pies y para la espalda durante la ejecución de las tres actividades a clasificar. Este proceso se llevó a cabo a través de un ensayo en el que participaron 7 niños de entre 1 y 5 años.

Tratamiento de los datos

Registro de los datos

Una vez se obtuvieron los datos, y se pre procesaron para quedarse con la información de utilidad, se registraron en el programa que iba a permitir su posterior análisis.

Procesamiento

Con los datos de aceleración totalmente disponibles y correctamente almacenados, se procedió a extraer las características que mejor los definieran. Para ello, se hizo uso del análisis estadístico, ya que este permite obtener relaciones entre los valores obtenidos. Las variables seleccionadas con este fin fueron la media y la desviación típica.

Debido a que los datos de aceleración obtenidos presentan tanto valores positivos como negativos, y la aceleración no puede ser negativa, previo al cálculo de los parámetros estadísticos fue necesario el cálculo del valor eficaz de cada una de las señales obtenidas. De esta forma, se aseguró que los valores obtenidos eran positivos.

Tras la realización de estos pasos, se obtuvo la media y la desviación típica de la aceleración para cada una de las direcciones del espacio, para cada usuario involucrado en el experimento, cada sensor (IMU) y cada actividad realizada.

Análisis realizado

Una vez conocidas las variables capaces de discriminar a los datos y relacionarlos entre sí, el siguiente paso era obtener la función que indicara cuáles son las variables de las que más dependían cada una de las actividades. De esta forma el gateo, la marcha y la carrera, aunque vendrá definida por: la media y la desviación típica de cada uno de los ejes y el IMU, tendrán mayor influencia por unas características que por otras.

Para poder realizar este proceso y poder crear un modelo a través de la función obtenida se hizo uso del método análisis discriminante lineal. Así, se obtuvo el algoritmo inicial de clasificación. Tras pasar por el ajuste, cuando aparezcan un dato este será capaz de predecir la actividad que se está realizando en función de los valores de las variables sea extraído que depende.

Tras haber realizado el modelo era necesario que este aprendiera para que pudiera generalizar y realizar clasificaciones correctas con las llegadas de datos desconocidos para lo que se llevó a cabo un proceso de validación a través del método "leave-one-out". Su funcionamiento es el siguiente: coge los datos de todos los usuarios excepto uno y se entrena con ellos sirviendo el

restante de testeo. Este proceso se repite tantas veces como usuarios haya, en este caso, 7. Con ello, todos los datos habrán pasado por el entrenamiento y la validación.

Tras la realización de estos pasos ya se obtuvo el clasificador deseado.

A continuación, se explican con detalle y detenimiento, los pasos previamente mencionados, haciendo hincapié en los materiales y métodos que fueron necesarios para que se llevará a cabo.

3.1 Diseño experimental

El diseño experimental es una parte importante de las investigaciones científicas ya que tras su realización se obtienen valores de diferentes variables o factores que, tras su análisis, van a permitir establecer la existencia de relaciones entre estas características. Se obtienen diferentes valores por que las variables varían en función de la situación y el contexto.

Para el presente trabajo, este apartado fue fundamental puesto que el interés era obtener datos de medidas inerciales a través de la realización de las actividades de gateo, marcha y carrera para poder después analizarlos, ver si con ellos era suficiente para discriminar y clasificar entre actividades y permitir la creación de un algoritmo de clasificación.

En todo experimento es necesario establecer cuál va a ser la selección de la muestra y como se va a llevar a cabo el ensayo.

Antes de pasar a establecer estos dos puntos, fue necesario establecer las variables a investigar que se resumieron en una tabla de planificación (Tabla 2).

Tabla 2. Determinación de las diferentes variables que se tuvieron en cuenta en el desarrollo del experimento.

TABLA DE PLANIFICACIÓN		
VARIABLES INDEPENDIENTE	VARIABLE DEPENDIENTE	VARIABLES DE CONTROL
<u>Factores cambiantes</u>	<u>Factor a medir</u>	<u>Factores que no se alteran</u>
Actividad (Gateo, Marcha, Carrera)	Aceleración	Colocación de los sensores, IMUs
Paciente (Talla, Edad)		Recorrido realizado durante las actividades

Una vez conocidas las variables influyentes, se redactó un protocolo de ensayo en el que se especificaban los pasos a seguir para la obtención de la aceleración.

3.1.1 Elección de la muestra

Debido a la imposibilidad de estudiar a toda la población, se seleccionó a una población de interés, que fuera representativa, de la cuál se iban a recolectar datos. Para seleccionarla se creó un anuncio para que todos aquellos participantes interesados y que cumplieran los criterios de inclusión pudieran participar. Estos criterios, establecidos en el protocolo de actuación, se pueden resumir en los siguientes puntos:

- Tener entre dos y seis años
- Utilizar una talla de calzado en el rango 22-28
- Índice de Masa Corporal acorde con la edad del participante, dentro de los límites correctos
- Capacidad para seguir instrucciones
- Capacidad para caminar, gatear y correr

Tras el periodo de inscripción, se obtuvieron 7 participantes, 3 niñas y 4 niños, con diferente talla de calzado y edad (Tabla 3).

Tabla 3. Resumen de la edad, talla y género de los participantes en el estudio.

PARTICIPANTE	GÉNERO	EDAD	TALLA
1	CHICA	1	22
2	CHICO	2	24
3	CHICA	4	28
4	CHICO	4	28
5	CHICA	2	24
6	CHICO	2	22
7	CHICO	3	26

Una vez se conocieron los participantes, se citaron durante la última semana de noviembre de 2017 en el IBV, y allí se procedió a la realización de los experimentos necesarios.

3.1.2 Realización de los experimentos

El primer paso para la realización de los experimentos fue la obtención de los datos con los que después se va a trabajar. Para ello se llevó a cabo en el Instituto de Biomecánica de Valencia (IBV) un ensayo con niños de edades de entre 1 a 5 años. En él, se adhirieron al calzado unos sensores inerciales, IMUs, que se encargaban de recoger las intensidades, en forma de señales de aceleración, con las que los niños realizaban tres actividades, el gateo, la marcha y la carrera. De esta forma, fue posible la generación de medidas inerciales.

Para la realización de los experimentos fue necesario contar con el siguiente material:

- Protocolo de ensayo diseñado previamente.

Este estaba dividido en cuatro apartados, que se siguieron ordenadamente durante la realización del experimento:

- o Pre-ensayo
- o Recepción de participantes
- o Instrumentación de participantes
- o Realización de pruebas biomecánicas (Ensayo y obtención de los datos)
- o Despedida

- Cuatro pares de zapatos de las tallas 22, 24, 26 y 28.

Estos números fueron elegidos acorde al rango de tallas que calzaban los participantes en el estudio.

- Sensores corporales.

Se utilizaron sensores corporales que eran capaces de medir una variable como la presión, aceleración u otros fenómenos de forma automática y enviar sus datos a otro dispositivo capaz de registrarlos.

Para el trabajo se contó con el uso de los sensores XSENS (Figuras 1-2). Se trata de unos sensores inalámbricos con un software gráfico que permite la exportación, el registro y la visualización de datos en tiempo real. Cuenta con sensores inerciales micro-mecanizados electromecánicos o MEMS altamente sensibles que son capaces de medir la aceleración lineal. Estos llevan incorporados un sistema de resorte de masa, que reside en el vacío. Un cambio en el acelerómetro provoca un desplazamiento del mismo que es recogido y leído a través de un sistema capacitivo.

- Software de adquisición de datos de los sensores y algoritmo de lectura de los mismos.

Ambos iban incluidos con los sensores con lo que no fue necesario crear ningún tipo de programa nuevo, si no simplemente cerciorarse de que estaban bien conectados.

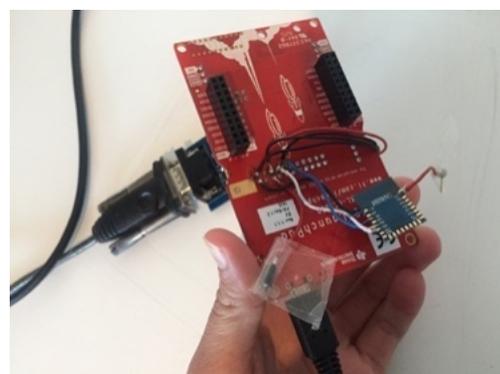


Figura 1. Sensores Xsens incorporados en su base para cargarse.

Figura 2. Placa basa para el registro y comunicación de datos entre los sensores y el programa de lectura.

Los pasos seguidos durante la realización del ensayo, en función del protocolo establecido, fueron:

- Pre-ensayo: Con anterioridad a la llegada del participante, era importante poner a punto los sensores. Para ello, se comprobaba que estuvieran cargados y conectados con el equipo que contenía el programa de lectura de datos. Una manera de comprobar esta conexión era mediante la aparición de gráficas de aceleración y velocidad angular en la pantalla de registro.
- Recepción del participante: Una vez llegado el niño a las instalaciones, se llevaba a cabo una pequeña presentación y se establecía una primera toma de contacto con él. A continuación, mientras su acompañante firmaba y rellenaba la documentación necesaria, se le explicaba de forma entretenida en que iba a consistir la actividad que iba realizar y lo divertida que era para que le resultara atractiva, aumentando de esta forma su interés en participar.
- Instrumentación del participante: Una vez que eran conscientes de la actividad a realizar, se procedía a la colocación de los zapatos. En el velcro de cada uno de ellos se enganchaban los sensores Xsens (Figura 4). El sensor restante, iba colocado en la cintura. En este último, de cara a amenizar la actividad y conseguir que disfrutaran de la misma, se adjuntó un logo de Batman o de muñecas (Figura 3).



Figura 3. Introducción de los sensores en el calzado.



Figura 4. Uso de cinturones atractivos.

- Ensayo: Constaba de la realización de 3 actividades diferentes: primero llevaban a cabo la marcha, posteriormente corrían durante un cierto tiempo y finalmente gateaban. En todas ellas, el proceso a seguir era el mismo: primero mostrar al participante un ejemplo práctico, facilitándole su posterior reproducción y después, se generaba una especie de circuito en la sala mediante la utilización de sillas, para marcar el recorrido a seguir y facilitar la realización de la actividad. Por otro lado, a modo de ayuda en la actividad de carrera, se solía participaba de forma conjunta con el participante, a modo de competición, lo que hacía que el niño mostrará mayor interés y menor vergüenza a la hora de realizar la prueba.
- Obtención de los datos: Para la obtención de los datos, los niños debían repetir el circuito establecido para cada actividad 3 veces, siendo la primera de prueba y las otras dos para leer cada vez uno de los sensores de los zapatos. En muchas de las ocasiones, hubo que repetir la actividad más de 3 veces, por falta de datos o por que el participante no estaba el tiempo suficiente realizando la actividad.

Los datos que se iban recogiendo, se asignaron con un código con la siguiente estructura:

BSS2_PZZ_TYY_aaZ_IMUX, donde:

BSS2, hacía referencia al calzado.

P, era el espacio correspondiente para indicar el género del participante, y se marcaba con V o M, para niño o niña respectivamente.

ZZ, indicaba el código relativo al usuario, independientemente de su género. En este caso al haber obtenido la participación de 7 niños, este código iba de 01 a 07.

YY, era para especificar la talla del zapato (T). Para este estudio 22,24,26 o 28.

aa, indicaba el tipo de prueba. En este caso, se empleó “ma” para indicar marcha, “ga”, para indicar gateo y “ca”, para indicar carrera.

Z, indicaba el número de repetición. Las variaciones que se establecieron fueron del 1 al 6.

X, indicaba el IMU correspondiente. Este número, por tanto, variaba entre 1, 2 o 3.

No sólo los datos estaban numerados, si no que los sensores también llevaban una numeración para poder ser identificados con claridad. En la siguiente tabla, se muestra la numeración de los sensores:

Tabla 4. Numeración sensores.

	Pie derecho	Pie izquierdo	Lumbar
IMUS	1 → 00B41DD2	2 → 00B414D0	3 → 00B4169D

- Final del ensayo/Despedida. Una vez obtenidos todos los datos, se felicitaba al participante por su colaboración y se le obsequiaba con un pequeño premio, como una

piruleta, en agradecimiento por su colaboración. Sus padres también recibían un detalle, en forma de bono regalo, por su participación.

Una vez terminados todos estos pasos, ya se tenían todas las muestras necesarias para pasar a realizar un análisis de los datos obtenidos.

3.2 Tratamiento de los datos

Una vez se obtuvieron los datos que se habían generado a través del ensayo experimental, era necesario llevar a cabo una etapa de preprocesado.

El principal objetivo de la preparación de los datos era manipularlos para que la información obtenida fuese de acceso más fácil y preciso. Los datos reales podían ser impuros y conducir a la extracción de patrones poco útiles debido a la existencia de inconsistencias o datos incompletos. Con el preprocesamiento, se obtuvo un conjunto de datos más pequeño que el original, pero más útil lo que mejoró el eficacia y eficiencia del proceso de minería, obteniendo datos de calidad.

En el caso de este estudio, el preprocesado de los datos consistió en recortar las señales que no proporcionaban información y quedarse únicamente con aquellos ejemplos que aportaran información interesante para el estudio.

Una vez finalizado este proceso se pasó al tratamiento específico de los datos. Se realizó un proceso de manipulación de los datos obtenidos que permitió extraer información útil y que el ordenador aprendiera en base a los ejemplos obtenidos.

Una vez claros los datos y seleccionados, convenía analizarlos para obtener los patrones que tras ellos se escondían y que iban a permitir llevar a cabo una clasificación correcta de la actividad cuando uno nuevo dato se introdujera. Este proceso se llevo a cabo en dos etapas, explicadas en los siguientes puntos: registro de los datos y análisis realizado.

Para su realización fue necesario contar con la ayuda de un ordenador y de un lenguaje de programación capaz de ejecutar órdenes, a través de las líneas de código que se establecían en él. En el trabajo se hizo uso de la herramienta de programación Python por las características que ofrecía: un propósito general, libre y de código abierto que contaba con estructuras de datos eficientes y de alto nivel; un enfoque simple pero efectivo para la programación orientada a objeto; una sintaxis clara y sencilla, con un lenguaje de alto nivel y tipado dinámico², que hacían de él una herramienta de desarrollo rápido de código y de mantenimiento sencillo. Además, contaba con un gran número de librerías fácilmente importables y con conexiones con otros lenguajes de programación.

² Tipado dinámico: Una misma variable puede tomar distintos valores dependiendo del contexto en el que se empleen.

3.2.1 Registro de los datos

Los datos obtenidos, en forma de señales de aceleración, se guardaron en 7 carpetas diferentes, cada una de ellas correspondiente a un usuario. El formato empleado para ello fue "csv", un formato de archivo sencillo donde se almacenaban los datos tabulares en formato de texto. Cada archivo contenía el nombre del dato, y tres columnas correspondientes a las aceleraciones en los ejes X, Y, Z, respectivamente.

Con todos los datos organizados, ya se pudo proceder a trabajar con Python y a llevar a cabo el análisis estadístico que permitió el paso al análisis posterior.

El primer punto del análisis posterior para conseguir el objetivo era leer los datos.

La herramienta de Python que se utilizó para ello fue la librería Pandas que poseía un amplio abanico de prestaciones para el análisis de datos estructurados y que contaba con la función "read_csv ()" que permitía leer datos en dicha extensión. Además, se decidió crear un diccionario que guardara todas las señales, para que así fuera más fácil acceder a ellas en el momento que se requirieran. Entre los distintos tipos de estructuras de datos que el programa ofrecía, se encontraban los diccionarios. En ellos, se podía almacenar cualquier tipo de valor e identificarlo a través de una clave de la siguiente forma:

Diccionario ["clave"] = "valor".

En el trabajo se guardaron todos los datos poniendo como clave el código "NUaaZIMUX" donde:

U, era el usuario (de 1 a 7)

aaZ, era la actividad a realizar y el ensayo

X, número de IMU correspondiente

Y como valor: cada una de las señales.

Para obtener este valor se usó la función "read_csv ()", que como se ha mencionado, era capaz de leer datos y exportarlos para poder trabajar con ellos. En el paréntesis de la función se le indicó el directorio de almacenamiento de los datos para que el programa acudiera a él y de esta forma pudiera extraerlos. En el proceso de guardado no se emplearon cabeceras para indicar cual era cada una de las aceleraciones (Tabla 6). Por suerte, esta función permitía introducir como input el nombre de las cabeceras que se desean, y así se hizo. Las cabeceras introducidas fueron en este caso X, Y, Z para aclarar de que dirección provenían las intensidades.

```
diccionario ["N6ga4IMU2"] = pd.read_csv(ruta+"/06/BSS2_V06_T22_ga4_IMU2.csv",  
names=["X","Y","Z"])
```

Ecuación 1. Código para guardar datos en un diccionario

Una vez realizado el proceso para todos los datos se guardó el script generado que se adjunta en los anexos del trabajo.

3.2.2 Procesamiento

Durante el procesamiento, con los datos leídos y archivados en Python, ya se pudo proceder al procesamiento puro de los datos que iba a permitir extraer las mejores características para definir los datos a través del análisis estadístico.

Como ya se ha mencionado en la definición del problema, antes de calcular los parámetros estadísticos de media y desviación típica, era necesario calcular el valor eficaz de cada una de las muestras. Este parámetro se definió como la raíz cuadrada del sumatorio de todos los valores de una señal al cuadrado dividido entre el número de valores sumados.

$$\text{Valor eficaz (RMS)} = \sqrt{\frac{1}{T} [f(t)]^2}$$

Ecuación 2. Fórmula del valor eficaz de una señal

Para la obtención del valor eficaz con la herramienta de Python fue necesario cargar previamente las librerías Pandas (ya explicada), Numpy y Matplotlib. La librería Numpy proporcionaba herramientas para la generación y manipulación de datos (arrays), así como para la computación numérica y Matplotlib, permitía el manejo de gráficos y herramientas de visualización 2D y 3D. Esta última, contaba además con funciones que permitían generar documentos PDF con las gráficas obtenidas. A parte de las librerías también era fundamental cargar el diccionario que contenía los datos.

La obtención del valor eficaz requirió una secuencia de operaciones que hacía recurrir al uso de funciones. En la declaración de las funciones se introdujeron los datos de entrada o input y en su cuerpo se definieron una sucesión de sentencias para que se ejecutara la operación a llevar a cabo.

La sucesión de sentencias para obtener el valor eficaz se llevó a cabo por medio de bucles "for" que eran estructuras de control que repetían un bloque de instrucciones. Específicamente, el bucle "for" elegido permitía iterar sobre una variable compleja un número predeterminado de veces establecidos por el usuario. También había que tener en cuenta que la indentación era clave para su correcta ejecución ya que en Python había que respetarla pues de ella dependía la estructura del código.

En el desarrollo de esta sección fue necesario la creación y empleo de 4 bucles "for".

El primero de ellos, que afectaba a los siguientes, se empleó para indicar que se requería la repetición de su cuerpo tantas veces (k) como datos ("NUaaZIMUX") hubiera en el diccionario. En el cuerpo de este bucle las sentencias que aparecían fueron las siguientes:

- Creación de una variable que almacenaba los datos.
- Cálculo de la longitud o cantidad de valores de cada uno de los datos. Python cuenta con una función, len (), que realiza de forma automática este cálculo.
- Extracción del nombre de cada dato ("NUaaZIMUX"), el usuario (U), la actividad (aa) y el IMU (X), correspondiente.

Una vez cumplidos estos requisitos, y a su mismo nivel de indentación, aparecían dos bucles:

En uno de ellos, se indicaba que su ejecución debía ser para cada valor (i) dentro del rango de la longitud (len ()) calculado con antelación. A la longitud se le restaba el tamaño de ventana (longitud-N) para asegurarse de que solamente se obtenía la información deseada. En su cuerpo aparecía otro bucle que indicaba que a parte de para cada dato y cada i, también se debía repetir para cada una de las componentes (comp) de aceleración X, Y, Z. Dentro del cuerpo de este último bucle es donde se daba la instrucción de la obtención del valor eficaz. Para definir la fórmula, se tuvo que hacer uso de las funciones media (mean) y raíz cuadrada (sqrt) que aporta la librería Numpy. Teniendo esto en mente, la fórmula, quedaba definida de la siguiente manera: “`vefcomps = np.mean(np.sqrt(valores[comp][i:i+N]**2))`”.

Todos los valores calculados se almacenaban en un DataFrame que es un tipo de objeto que existe en Pandas, basado a su vez en Numpy, y que es capaz de generar tablas en las que se pueden especificar tanto el index (nombre de las filas) como las columns (nombre de las columnas). El DataFrame (vefcomps) se creó con las columnas de interés: una para cada valor eficaz de cada una de las aceleraciones X, Y, Z; y luego, cuatro más para indicar el nombre del dato, usuario, IMU y actividad, obteniéndose un total de 7 columnas (X, Y, Z, Dato, Usuario, IMU, Actividad). Al inicio, este DataFrame estaba vacío, pero a medida que se iban ejecutando todos y cada uno de los pasos y tras la obtención de todos los valores y de la función “append”, que DataFrame ofrece, se iban añadiendo los valores y la tabla se iba completando.

En el cuarto bucle se generaban las gráficas del valor eficaz obtenido. Resultó interesante la obtención de las gráficas ya que estas iban a permitir sacar resultados y conclusiones de los valores obtenidos. Conocer las distribuciones de las intensidades obtenidas para cada participante era conveniente para realizar un análisis gráfico y visual de los datos y poder hacerse una idea de la tendencia seguida en los valores para cada una de las actividades. En la definición del bucle se establecía su repetición para cada una de las componentes X, Y, Z. En el cuerpo del bucle, se pedía que se generaran las gráficas del DataFrame (vefcomps) a través de la función “plot” de la librería Matplotlib. Cabe resaltar que para un dibujo adecuado de estas se tuvo que crear un filtro que equiparara la columna Dato del DataFrame con los nombres de los datos (k) del diccionario. La librería Matplotlib, también permite a través de las funciones “label”, “title” y “legend”, el etiquetado de las gráficas y de sus ejes. Como resultaba de gran utilidad guardar en un mismo documento todas las gráficas, para tenerlas bien almacenadas y poder disponer de ellas en cualquier momento, se emplearon de las funciones “PDFPages” y “savefig ()” que esta misma librería ofrecía.

Finalmente, y gracias a la posibilidad que ofrecían las funciones de devolver valores, se pidió a la función que devolviera el DataFrame creado y lo asignara a una variable para que el programa lo almacenara en su base de datos, lo que dió la posibilidad de trabajar con él en etapas posteriores.

Antes de continuar con la obtención de los valores de media y desviación típica, se procedió a guardar todos los datos en un Excel. Para ello hubo que emplear la función “ExcelWriter ()” de Pandas donde en el input (paréntesis) había que indicarle el nombre con el que se deseaba guardar el archivo. Después emplear la función “to_excel” de DataFrame para que esta realizara la conversión y finalmente usar la función “save ()” para guardarlo.

El resultado obtenido fue el mismo que el DataFrame a excepción del formato. Aunque debido a la semejanza entre ambas este paso parecía prescindible en la realidad es que era verdaderamente útil pues permitió ahorrar tiempos de espera, es decir, agilizó el proceso de obtención de la información al no tener que ejecutarse cada vez que se abría el programa de nuevo. Simplemente, leyendo el Excel creado, al igual que se hacía para leer archivos “csv”, pero ahora con archivos “Excel”, se podía tener toda la información de forma rápida con la función de Pandas “read_excel (‘nombre del archivo’)”.

De todos los parámetros disponibles para el análisis estadístico, se procedió a obtener la media y la desviación típica puesto que eran las más significativas para obtener las medidas de centralización y de dispersión, respectivamente.

La media aritmética indicaba el centro de gravedad de la distribución estadística y la desviación típica permitía saber si los datos estudiados se encontraban más concentrados o más dispersos.

$$\bar{x} = \frac{\sum x_i * f_i}{N}$$

Ecuación 3. Fórmula de la media de los datos

$$\sigma (\text{Desviación típica}) = \sqrt{\bar{x}^2 - \overline{x^2}}$$

Ecuación 4. Fórmula de la desviación típica de los datos

Conocer la forma en la que se distribuían los datos obtenidos era la manera más eficiente de poder encontrar similitudes y diferencias entre datos pertenecientes a las mismas actividades y diferentes al resto de actividades.

Python, presentaba una opción para trabajar con los DataFrames conocida como “groupby”. Esta función permitía agrupar los datos en función de los parámetros indicados. La implantación de esta función en el trabajo fue de la siguiente forma: Por una parte, se agruparon los datos en función de la actividad, el usuario y el IMU. Y, por otra parte, para esa agrupación y para cada componente X, Y, Z, se calcularon los parámetros con las funciones de Numpy “mean”, para la media, y “std”, para la desviación estándar (Gráfico 3). En función de los usuarios e IMUs que se obtuvieron para cada actividad, se extrajeron mayor número de medias y desviaciones.

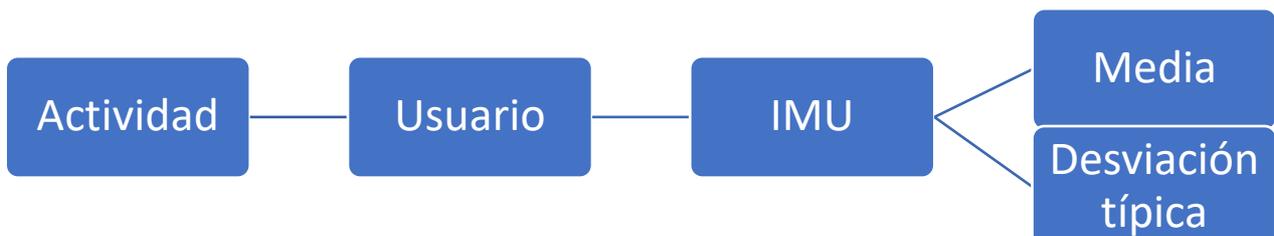


Gráfico 2. Esquema de la agrupación de datos llevada a cabo

La siguiente figura muestra cómo se llevó a cabo el proceso de separación solamente para los usuarios 2 y 3 de la actividad de carrera:

```
In [4]: d.head()
Out[4]:
```

Actividad	Usuario	IMU	X		Y		Z \
			mvef	dvef	mvef	dvef	mvef
ca	2	1	17.356427	2.046546	5.517604	0.765392	12.296189
		2	17.850638	1.596160	5.459697	0.571308	13.580289
		3	10.752574	0.541129	3.256112	0.743653	4.202144
	3	1	20.142914	3.202721	12.576431	1.920923	17.176020
		2	21.556373	2.965518	11.722245	1.288384	16.561353

Actividad	Usuario	IMU	dvef
ca	2	1	1.649601
		2	1.239737
		3	1.322188
	3	1	1.479357
		2	2.016981

Figura 5. División de datos y posterior obtención de la media y desviación típica.

Este punto permitió reducir la cantidad de datos de los que se partían, seleccionando sólo aquellos que eran verdaderamente útiles para proseguir con el desarrollo del algoritmo de clasificación.

La siguiente actuación fue crear una variable para cada media y desviación típica obtenida en cada una de las direcciones X, Y, Z. Esto fue posible mediante el uso del método “values”, que fueron capaces de devolver los valores demandados. También era imprescindible extraer los valores de la actividad, IMU e usuario. En este caso, no se empleó el método “values” si no que se hizo uso de “index.get_level_values (‘índice a extraer’)” ya que lo que se requería era que se retornará el valor del índice. Además, había que tener en cuenta que en el caso del IMU y del usuario, a pesar de que este se mostraba como un valor numérico realmente se trataba de un valor categórico, puesto que como su nombre indica hacen referencia a una categoría. Es decir, pese a que la columna de los usuarios se dividía en números del 1 a al 7 estos eran solo para nombrar a cada uno de ellos y en ningún momento indicaban un valor correspondiente a dicha variable. Esto ocurría de la misma forma para los IMUs. Para que el lenguaje de programación lo detectara bastaba con indicar que el tipo de elemento a extraer era un objeto (dtype = object).

Con los datos resultantes, se creó un diccionario que los almacenaba. Posteriormente y para finalizar el análisis estadístico se creó un nuevo DataFrame (datos) con los datos del diccionario que iba a facilitar el uso de los datos obtenidos en las siguientes etapas.

3.2.3 Análisis realizado

Una vez conocidas las variables definitorias de los datos era posible la creación del algoritmo. Teniendo en mente que los datos obtenidos estaban etiquetados, que la cantidad de muestras

no era muy elevada y haciendo uso de la ayuda ofrecida por “scikit-learn” (Figura 6), se dedujo que la clasificación supervisada sería la opción más adecuada. De entre sus posibles opciones, el uso de un clasificador probabilístico sería el más óptimo. El modelo seleccionado fue, el análisis discriminante lineal, más conocido como LDA.

Es un clasificador clásico con una superficie de decisión lineal, tal y como su nombre lo indica. Y como tal, asumía que los datos de cada clase seguían una distribución gaussiana y que las matrices de covarianza, que determinaban la forma de la densidad de probabilidad, eran idénticas para los datos de las distintas clases. Si eran iguales, las distribuciones de las probabilidades para cada clase tendrían la misma forma, y simplemente estarían desplazadas por la diferencia entre medias y escaladas por la probabilidad a priori.

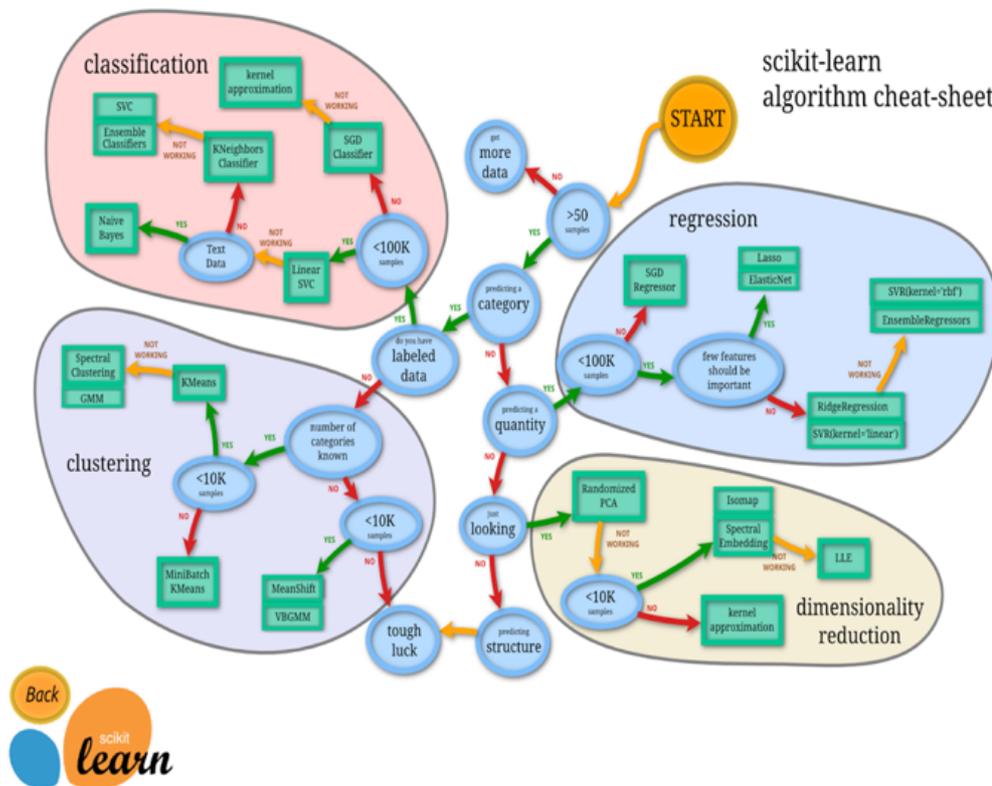


Figura 6. Ayuda para la selección del modelo

Las funciones principales eran:

- Crear una función capaz de distinguir con la mayor precisión posible a los miembros de una u otra clase.
- Con la función creada, crear y ajustar un modelo que cuando recibiera un dato fuera capaz de clasificarlo en cada una de las clases establecidas.

En resumidas cuentas, con esta técnica lo que se pretendía era conocer que variables caracterizan mejor a las actividades de marcha, gateo y carrera. Además, se buscaba que, al recibir un nuevo dato, fuera capaz de indicar cual de las actividades anteriores se estaba realizando.

Para que Python creara este modelo y lo llevara a cabo, los pasos que se siguieron fueron:

1. Diseño a través de la selección de variables dependientes e independientes.

Una variable dependiente representaba un elemento cuyo valor dependía de cómo se modificara la variable independiente. La finalidad de este clasificador era que, a partir de las variables independientes, este fuera capaz de predecir que tipo de actividad se estaba llevando a cabo. Por ello, mientras la variable dependiente iba a ser la actividad, las independientes eran los valores de la media, de la desviación típica de cada componente de la aceleración y del IMU. Para crearlas, se hizo uso de listas. En el lenguaje de programación empleado, las listas eran conjuntos de elementos ordenados que se podían manipular con facilidad. La primera lista creada, "X", tenía en su interior las variables independientes y la segunda, "y", la variable dependiente. Todos estos datos se extrajeron del DataFrame creado en el apartado anterior, "datos", con el método "values".

2. Estimación del modelo.

Tras haber diferenciado las diversas variables, se procedió a obtener la función discriminante y ajustar el modelo. Era necesario pues para este punto, haber importado con antelación la función `LinearDiscriminantAnalysis` que la librería `Scikit-Learn` presenta. Esta librería de propósito general estaba indicada para su uso en 'machine learning'. Poseía integrados muchos algoritmos y módulos extendidos para las operaciones de pre-procesado, validación, etc.

Seguidamente, se llamó a la función a través del código "`LinearDiscriminantAnalysis()`" y se le asignó a una variable. Después, se ajustó el modelo a través del método "`fit()`", pasándole como input las listas "X" e "y" creadas.

Tras la ejecución de estos pasos ya se tuvo el modelo de clasificación y el siguiente paso era ponerlo en acción. En este punto se obtuvieron los coeficientes de la función que permitieron ver como afectaban cada una de las variables independiente a la variable dependiente, actividad. En Python se llevó a cabo este proceso con el uso de los atributos `coef_` (para extraer los coeficientes), `priors_` (porcentaje de prioridad de cada una de las actividades).

3. Método de estimación.

El siguiente paso era obtener las predicciones que el modelo realizaba con los datos dados y compararlas con el valor real. Para poder hacer esta comparación, se crearon, antes que nada, dos nuevas listas: una para que almacenara los datos predichos (`Pred = []`), y otra para que almacenara los reales (`Real = []`).

El siguiente paso era la creación de un bucle "for", en el que se requería que el cuerpo de instrucciones se repitiera tantas veces como actividades fueron registradas, siendo en este caso 49. En cada una de las iteraciones, se llevaron a cabo dos operaciones:

- 3.1 Obtención de los valores predichos por el modelo y su posterior almacenamiento en un variable. Para realizar la predicción se llamó al método “predict ()” que ofrecía la función LDA y se puso como input la variable dependiente “X”.
- 3.2 Obtención tanto del valor predicho como del real, que también se almacenaron en una variable. Para obtener el valor predicho se llamó a la variable que los tenía almacenados y para obtener el valor real se acudió al DataFrame (datos) a través del método “iloc []”, que devolvía el índice (“ma”, “ga” o “ca”) al indicarle donde se encontraba ubicado (datos.iloc[ind][“Actividad”]).

El último paso era introducir en las listas creadas al inicio, Pred y Real, cada uno de los datos correspondientes por medio del método “append” para después imprimir por pantalla los resultados obtenidos y poder compararlos.

Trascurridas estas secuencias de operaciones, se obtuvo ya el clasificador, pero se requería de un último paso para que se pudiera dar por finalizado. Este era darle la capacidad de que este fuera capaz de generalizar y poder ser utilizado para nuevos casos desconocidos.

Así, una vez creado el modelo, fue necesario hacerlo pasar por una fase de validación en la que se ejercitaba su capacidad de clasificación. Se eligió la validación cruzada que se utilizaba en los procesos de aprendizaje automático para juzgar como de bien se ajustaba al modelo. El conjunto de datos se dividió en dos partes. Una de ellas, se utilizó para entrenar el modelo mientras que la otra sirvió para testarlo. Con esto, se obtuvo una idea acerca de cuál era la capacidad a la que aspiraba el modelo a la hora de generalizar datos independientes.

El método seleccionado para la validación de este trabajo fue “Leave-one-out”. Este método, en vez de dividir el conjunto de datos entero, empleaba todos los datos excepto uno para ajustarlo y luego hacía una predicción que posteriormente se comparaba con el valor actual. Esto se repitió para todos los datos disponibles. Debido a que se tenían pocos datos resultó muy útil puesto que se crearon mayor cantidad de iteraciones y el entrenamiento resultó más eficaz.

En este trabajo, se decidió que los datos se separaran en función de las variables “X” e “y”, previamente calculadas y, además, para cada participante. Así, en cada iteración se iban extrayendo los datos de uno de los participantes para que funcionara como dato de testeo y el resto actuaban como datos de entrenamiento (Gráfico 4).

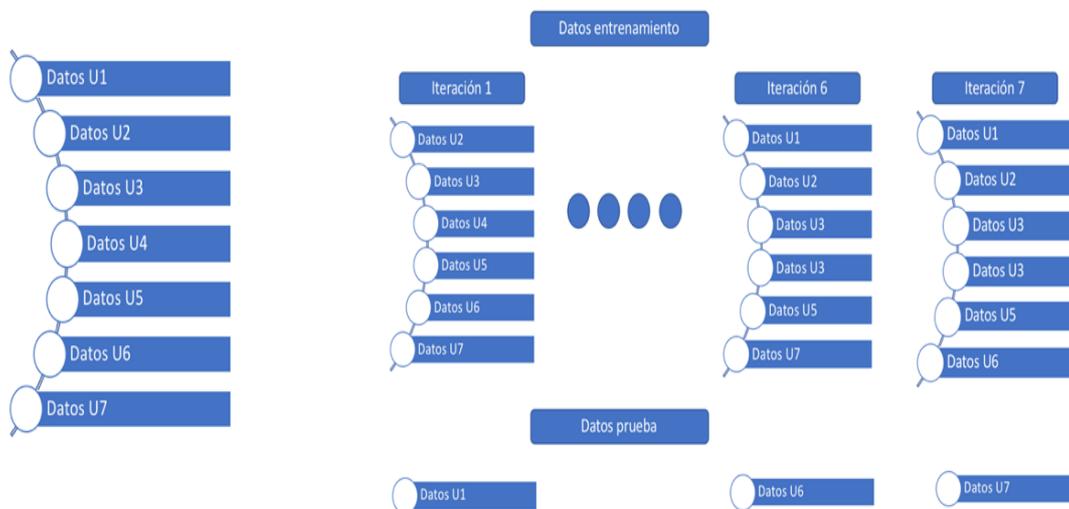


Gráfico 3. Esquema del método de validación Leave-one-out

La implantación de este proceso en Python requirió de los pasos:

- Importación de la función "LeaveOneOut", que sklearn ofrecía, que sería la encargada de llevar a cabo el proceso de validación.
- Obtención de un array con la columna Usuario del DataFrame datos. Para crear arrays, que eran un tipo de dato estructurado de Python que permitía almacenar elementos de todo tipo y que requirió el uso de la librería Numpy. Para extraer el valor de la columna, se volvió a recurrir al método "values".
- Llamar a la función "LeaveOneOut()" asignándole a esta llamada una variable e indicarle a través del método "get_n_splits()", la divisiones que se deseaban realizar. En el input del método (paréntesis), se introdujeron las variables "X" e "y". Con la finalidad de dividir esos datos por grupo de usuario se indicó también en el input que los grupos ("groups"), serán los datos del array creado en el paso anterior.
- Creación de un bucle "for". En su cabecera se indicó que debían realizarse las repeticiones para cada uno de los índices de entrenamiento y test obtenidos antes en la división de los datos. En el cuerpo, se almacenaron gracias al uso de listas. Por una parte, se crearon dos listas: una con los índices de entrenamiento de las variables independientes "X" y otra con los índices de testeo de esas mismas variables. De la misma forma, se obtuvieron otras dos, pero en este caso para la variable dependiente "y". Por otra parte, se llamó a la función LDA, y se volvió a ajustar el modelo, tal y como se detalla en el apartado anterior. En cambio, en este caso, los datos para el ajuste eran las listas de entrenamiento creadas previamente.
- Creación de tres nuevas variables. En una de ellas, se almacenaron los datos de la predicción de la actividad obtenida con el modelo ajustado. Para ello, se empleó el método "predict ()" y se le pasó como input el conjunto de datos de test de la variable

“X”. En otra, se tuvieron los valores de testeo de la variable “y” y en la última, los de testeo de la variable “X”. Para introducir los valores en cada una de ellas, se empleó el método “append” y se crearon vacías antes de comenzar el bucle.

Tras la sucesión de estos pasos, ya se pudo pasar a la obtención de la validación y los resultados del clasificador creado pues este ya se había entrenado con todas las muestras disponibles.

Uno de los pasos fundamentales tras la creación del clasificador, fue evaluar su capacidad de predicción. Los métodos escogidos para dicha realización fueron: la matriz de confusión y el cálculo de la exactitud.

a) Obtención de la matriz de confusión.

La matriz de confusión o de error mostraba la relación entre las dos series de medidas correspondientes al área de estudio. La primera correspondía a los datos de referencia o valores reales y la segunda, a los extraídos por el clasificador o valores predichos, formando así una matriz $n \times n$, donde n era el número de clases, siendo n para este trabajo, el número de actividades a clasificar.

Su funcionamiento, encuadrado en el trabajo, fue el siguiente: se obtuvo una matriz de 3×3 , con las tres actividades a clasificar: gateo (ga), carrera (ca) y marcha (ma), como la que aparece a continuación:

Tabla 5. Generación de la matriz de confusión

		Valor Predicho		
		Carrera	Gateo	Marcha
Valor real	Carrera			
	Gateo			
	Marcha			

Para cada conjunto valor predicho – valor real, se insertó un valor 1 en la casilla correspondiente. Es decir, si se había predicho carrera y el real era carrera, se introducía un 1 en la primera casilla de la primera columna y de la primera fila. En cambio, si la combinación la actividad predicha era gateo y la real, marcha, el 1 iría en la segunda columna, tercera fila. Y así, se iría rellenando la matriz hasta que se acabaran los pares, sumando una unidad a cada caso repetido. Lo ideal hubiera sido que solamente existiesen valores en la diagonal, ya que de esta forma el clasificador predeciría de forma totalmente correcta y sin error alguno, pero esto era difícil de obtener.

La obtención de la matriz con Python se llevó a cabo con la función “confusión_matrix ()” que la librería sklearn ofrecía. Los parámetros que se introdujeron como inputs fueron: los valores reales y predichos y las etiquetas correspondientes, "ca", "ga", "ma".

b) Obtención de la exactitud

La exactitud se definió como el grado de concordancia entre las clases asignadas por el clasificador y sus ubicaciones correctas.

Para comparar la exactitud con la que el clasificador predecía correctamente el tipo de actividad era necesario importar la sub-librería “metrics” de la librería de sklearn.

Hasta el momento los datos se encontraban almacenados en tres listas, pero para poder obtener este parámetro era necesario pasar estas listas a arrays. Tras ello y empleando los métodos “.score()” que llevaba implementado la función LDA y el método “.accuracy_score()” de la sub-librería “metrics” se obtuvo la exactitud. En el primer método, se introdujeron como inputs los datos de test tanto de la variable dependiente como de la independiente mientras que el segundo método se introdujeron los datos de la variable “y” tanto de test como predichos.

Con estos pasos, ya se pudo proceder a examinar los resultados y extraer conclusiones del trabajo realizado.

Paralelamente al trabajo realizado para la creación del clasificador, también se obtuvieron los picos máximos, mínimos y pasos por cero que en estudios futuros podrían ser utilizados para la obtención del número de pasos realizados durante las actividades de ganeo, marcha y carrera.

Para su obtención a través de Python se importaron las librerías Pandas, Numpy, DATOSXYZ y Matplotlib. Además, se contó con la función “clasifica_puntos”.

El primer paso fue la creación de un bucle “for” en la que, para cada uno de los datos del diccionario, al igual que se hizo en la creación del valor eficaz, se obtuvieron los estos valores y se almacenaran en una nueva variable.

Después, se crearon dos diccionarios: uno diccionario vacío, data, que iba a almacenar en su interior cada uno de los datos de coordenadas obtenidos y otro, picoscomp, que estaba formado por cada uno de los valores de cada una de las aceleraciones en forma de array.

Tras su creación, se creó otro bucle que requería su repetición para cada dato del segundo diccionario creado, picoscom. En él, se llevarían a cabo varias secuencias de sentencias:

- Primero, se almacenaban esos datos en una variable nueva
- Luego, se llamaba a la función “clasifica_puntos()” para obtener el valor máximo, mínimo y los pasos por cero. En el input de la función se introducía la variable que se acababa de crear, el error permitido, y la intención o no de generar las gráficas obtenidas. Para el trabajo, resultó útil extraer las gráficas así que se demandó su extracción por pantalla y la generación de un archivo PDF. Para ello, se llevó el mismo proceso que para la generación de gráficas cuando se extrajo el valor eficaz.

- Una vez obtenidos los datos, se crearon tres nuevas variables dentro del primer diccionario creado, data, teniendo como valor los máximos, mínimos y pasos por ceros respectivamente.
- Finalmente, gracias al uso de la función “.append ()”, se introdujeron en un DataFrame previamente definido con tres columnas, para cada una de las variables de interés. En el input se introdujo el diccionario data.

Los datos obtenidos eran las coordenadas de donde se encontraban cada uno de los máximos, mínimos y pasos por ceros de las señales obtenidas. Con ellos, se podían extraer su valor y realizar el proceso de cuantificación de la actividad.

CAPÍTULO IV: RESULTADOS Y DISCUSIÓN

En el siguiente capítulo, se van a exponer los resultados obtenidos durante la ejecución del programa haciendo hincapié en como se presentaron y cuáles fueron los datos obtenidos, los coeficientes establecidos a través del modelo y su validación.

4.1 Descripción de los datos obtenidos

A continuación, se va a realizar un análisis detallado de los datos que se obtuvieron tras el ensayo realizado.

4.1.1 Número de señales

Una vez se adquirieron los datos, se guardaron numerados en una tabla para clarificar cuales eran los ejemplos y que medidas se obtuvieron de cada uno de los niños que participaron en el estudio (Tabla 3).

Tabla 6. Ejemplo del guardado de los datos tras su adquisición. Estos se guardan en función de la actividad, el ensayo y el lugar en el que se encuentra el sensor.

			Pie derecho	Pie izquierdo	Lumbar
IMUs	Marcha	Ensayo 1	BSS2_M01_T22_ma1_IMU1	BSS1_M01_T22_ma1_IMU2	BSS1_M01_T22_ma1_IMU3
		Ensayo 2	BSS2_M01_T22_ma2_IMU1	BSS1_M01_T22_ma2_IMU2	BSS1_M01_T22_ma2_IMU3
		Ensayo 3	BSS2_M01_T22_ma3_IMU1	BSS1_M01_T22_ma3_IMU2	BSS1_M01_T22_ma3_IMU3
	Carrera	Ensayo 1	BSS1_M01_T22_ca1_IMU1	BSS1_M01_T22_ca1_IMU2	BSS1_M01_T22_ca1_IMU3
		Ensayo 2	BSS1_M01_T22_ca2_IMU1	BSS1_M01_T22_ca2_IMU2	BSS1_M01_T22_ca2_IMU3
		Ensayo 3	BSS1_M01_T22_ca3_IMU1	BSS1_M01_T22_ca3_IMU2	BSS1_M01_T22_ca3_IMU3
	Gateo	Ensayo 1	BSS1_M01_T22_ga1_IMU1	BSS1_M01_T22_ga1_IMU2	BSS1_M01_T22_ga1_IMU3
		Ensayo 2	BSS1_M01_T22_ga2_IMU1	BSS1_M01_T22_ga2_IMU2	BSS1_M01_T22_ga2_IMU3
		Ensayo 3	BSS1_M01_T22_ga3_IMU1	BSS1_M01_T22_ga3_IMU2	BSS1_M01_T22_ga3_IMU3

Con todos los datos almacenados, se procedió a analizar cual de ellos eran válidos e iban a aportar información relevante. Estos serían el punto de inicio para la creación y validación del clasificador (Tabla 6). En total se obtuvieron 178 ficheros de los ensayos realizados teniendo en cuenta cada una de las actividades, cada IMU y cada usuario.

Los datos que se obtuvieron para cada usuario fueron los siguientes:

Usuario 1

- Pruebas de gateo y marcha. Se realizaron 3 ensayos para cada ejercicio, de los cuales sólo resultaron válidos 1 y 3.
- Sólo se pudieron recoger datos de los IMUs 1 y 2.

Usuario 2

- Pruebas de carrera y marcha. Se realizaron 3 ensayos para la marcha y sólo se pudo obtener un ensayo de carrera válido.
- Se pudieron recoger datos de los tres IMUs.

Usuario 3

- Pruebas de gateo y carrera. Se realizaron 3 ensayos, de los cuales sólo resultó válido el tercero para carrera y en el caso del gateo, los tres fueron válidos.
- Se pudieron recoger datos de los tres IMUs.

Usuario 4

- Pruebas de carrera, gateo y marcha. Se realizaron 4 ensayos para actividad, pero en el caso del gateo hubo que realizar un 5 ensayo pues el 4 no resultó efectivo.
- Se pudieron recoger datos de los tres IMUs.

Usuario 5

- Pruebas de carrera, gateo y marcha. Se realizaron 5 ensayos para actividad, pero en el caso de la carrera hubo que realizar un 6 ensayo pues el 5 no resultó efectivo.
- Se pudieron recoger datos de los tres IMUs.

Usuario 6

- Pruebas de carrera, gateo y marcha. Se realizaron 4 ensayos para actividad, pero en el caso de la marcha se realizó uno más. Se pudieron recoger datos de los IMUs 1 y 2.

Usuario 7

- Pruebas de carrera, gateo y marcha. Se realizaron 4 ensayos para actividad.
- Se pudieron recoger datos de los tres IMUs.

Tabla 7. Citación de los datos válidos en función del participante. Con ellos se realizarán las posteriores etapas.

	U1	U2	U3	U4	U5	U6	U7
Pruebas	Gateo Marcha	Marcha Carrera	Gateo Carrera	Gateo Marcha Carrera	Gateo Marcha Carrera	Gateo Marcha Carrera	Gateo Marcha Carrera
Ensayos Válidos	1 3	1 2 3 *Carrera sólo 1 válido	1 2 3 *Carrera sólo 3 válido	1 2 3 4 *Gateo hubo que realizar un 5 ensayo (4 no válido)	1 2 3 4 5 *Gateo hubo que realizar un 6 ensayo (5 no válido)	1 2 3 4 *Marcha hubo que realizar un 5 ensayo	1 2 3 4
IMUs	1,2	1,2,3	1,2,3	1,2,3	1,2,3	1,2	1,2,3
Total ensayos	8	12	12	36	48	26	36

Una vez los datos fueron guardados en cada uno de los ficheros, se obtuvo una tabla (Tabla 6) que contenía las aceleraciones para cada ensayo, cada IMU, cada actividad y cada usuario. En total, se obtuvieron 126445 filas de datos.

Tabla 8. Primeras 7 filas de las aceleraciones obtenidas (X, Y, Z) del IMU 1 del usuario 1 mientras gatea.

BSS2_M01_T22_ga1_IMU

5.6064	1.3322	8.5246
6.4923	1.8701	7.7658
6.7315	1.8161	7.4736
6.5389	1.5128	7.4359
6.6196	1.5776	7.5205
6.9302	1.5723	7.3869
7.0815	1.8588	7.0015

4.1.2 Longitud de las señales

Cada una de las señales obtenidas contenía una gran cantidad de datos. A través de Python se pudo determinar la longitud de cada una de ellas. La variación de la longitud fue función del tiempo que se empleó en llevar a cabo cada una de las actividades propuestas.

Un ejemplo de los datos que se obtuvieron es el siguiente:

- N5ma4IMU3: 1199 datos
- N5ma4IMU2: 1197 datos
- N5ma4IMU1: 1197 datos
- N5ca1IMU3: 892 datos
- N5ca1IMU2: 892 datos
- N5ca1IMU1: 892 datos
- N5ga5MU3: 693 datos
- N5ga5MU2: 693 datos
- N5ga5MU1: 693 datos

Como se puede observar de los ejemplos obtenidos, para cada uno de las actividades y ensayos el número de datos era distinto, pero no se apreciaron diferencias entre IMUs. Esto se debió a que los ensayos se realizaron con los tres IMUs colocados a la vez.

4.1.3 Medidas perdidas

Durante la realización de un ensayo, aparecieron diversos factores que no estaban previstos y que hicieron que los datos obtenidos no fueran los deseados. Sin duda, trabajar con niños fue una causa de que existieran variables o se produjesen errores con los que no se contaban. En el momento en que los diferentes participantes estaban realizando las actividades, algunos de ellos decidían parar por que estaban cansados o simplemente no tenían ganas de seguir con el ejercicio. Además, había casos en los que los participantes más mayores no estaban por la labor de realizar la actividad de gateo. También hubo casos en los que la carrera la realizaban a modo de marcha y no de carrera.

Por todo ello, hubo ensayos que no se pudieron tomar como válidos (Tabla 9), en total 27.

Tabla 9. Datos perdidos en el ensayo

	U1	U2	U3	U4	U5	U6	U7
Pruebas	Carrera Gateo Marcha	Carrera	Carrera	Gateo	Gateo	Marcha	

Ensayos NO Válidos	2 *Para la carrera no se pudo obtener ningún ensayo	1	1 2	4	5	4	
IMUs	1,2,3	1,2,3	1,2,3	1,2,3	1,2,3	1,2,3	
Total ensayos	9	3	6	3	3	3	0

4.2. Clasificación de la actividad

Una vez conocidas todas las muestras disponibles y con la creación del código de programación se pudieron extraer sus resultados y con ellos los del trabajo presente que se va a proceder a explicar.

4.2.1 Descripción de los clasificadores

Como se ha comentado en la metodología, antes de obtener las características de la señal se introdujeron todos los datos en un diccionario.

```
In [3]: diccionario ["N6ga4IMU2"]
Out[3]:
```

	X	Y	Z
0	6.1558	0.032828	7.638900
1	6.0672	0.404840	7.577700
2	5.9055	0.073668	7.575700
3	6.0283	0.206190	7.506700

Figura 7. Ejemplo valor diccionario en el que solo aparecen tres filas.

El diccionario tenía almacenado los datos del mismo modo que se dividieron los ficheros que se habían creados tras su obtención (Tabla 6) con la adición de las cabeceras (Figura 7). Estas permitieron que el acceso a las aceleraciones para cada eje fuera más rápido y, por tanto, que el código resultara más eficiente.

Con estos datos disponibles, el primer paso seguido en la metodología fue la extracción del valor eficaz de las señales de aceleración. Una vez calculado se obtuvo un DataFrame, vefcomps, con cada uno de los valores obtenidos para este parámetro donde además se indicaba a que dato, usuario, IMU y actividad correspondía. El resultado fue el siguiente:

```
In [4]: vefcomps
Out[4]:
```

	X	Y	Z	Dato	Usuario	IMU	Actividad
0	9.754790	1.523618	1.490361	N5ma4IMU3	5	3	ma
1	9.792290	1.525078	1.478928	N5ma4IMU3	5	3	ma
2	9.818362	1.515146	1.462855	N5ma4IMU3	5	3	ma
3	9.821542	1.502852	1.443677	N5ma4IMU3	5	3	ma
4	9.814470	1.498236	1.434438	N5ma4IMU3	5	3	ma
5	9.814604	1.502026	1.432979	N5ma4IMU3	5	3	ma
6	9.829223	1.507705	1.428633	N5ma4IMU3	5	3	ma
7	9.843006	1.502937	1.423238	N5ma4IMU3	5	3	ma
8	9.853526	1.488271	1.420630	N5ma4IMU3	5	3	ma
9	9.849186	1.487585	1.440448	N5ma4IMU3	5	3	ma
10	9.826566	1.500364	1.450386	N5ma4IMU3	5	3	ma

Figura 8. 10 primeras filas del DataFrame creado

En la Figura 8 solo se observan 10 filas y todas para un mismo caso, pero, en realidad se extrajeron un total de 126446 filas de datos, tal y como se había descrito anteriormente, solo que esta vez, con los datos de valor eficaz. Este DataFrame se trasladó a un Excel siguiendo el mismo formato que el presentado en la Figura superior.

Un ejemplo de las gráficas obtenidas tras el cálculo de esta variable es el que aparece en la figura 9. Las tres imágenes que aparecen en esta figura pertenecían al usuario 5, al quinto ensayo y a los datos obtenidos del IMU1, que se correspondían con el pie derecho. La diferencia entre ellas radicaba en la actividad que se estaba realizando en cada momento. Mientras en la primera imagen se mostraba la marcha; en la segunda, el gateo; y en la última, la carrera.

Como era de esperar, la que presentaba los valores más elevados de las tres gráficas fue la carrera puesto que realizar dicha acción implicaba ir a mayor velocidad, y por tanto establecer una mayor aceleración; seguida de ella, se encontraba la marcha y finalmente el gateo, que era la actividad que menor intensidad demandaba durante su ejecución.

Tras la realización de un análisis de las mismas se extrajo que:

Durante la realización de la marcha la dirección que mayores valores obtuvo fue la del eje “x”, debido a que era la dirección en la que se desplazaban los participantes. Seguidamente, se encontraba la del “z”, en la que había que tener en cuenta que durante la realización de la marcha no se seguía una dirección recta y, por tanto, de ahí salieron los valores de esta. Finalmente, el eje “y”, era el que menores valores presentaba puesto que era la dirección vertical y la espalda no sufrió grandes cambios durante la ejecución de la actividad.

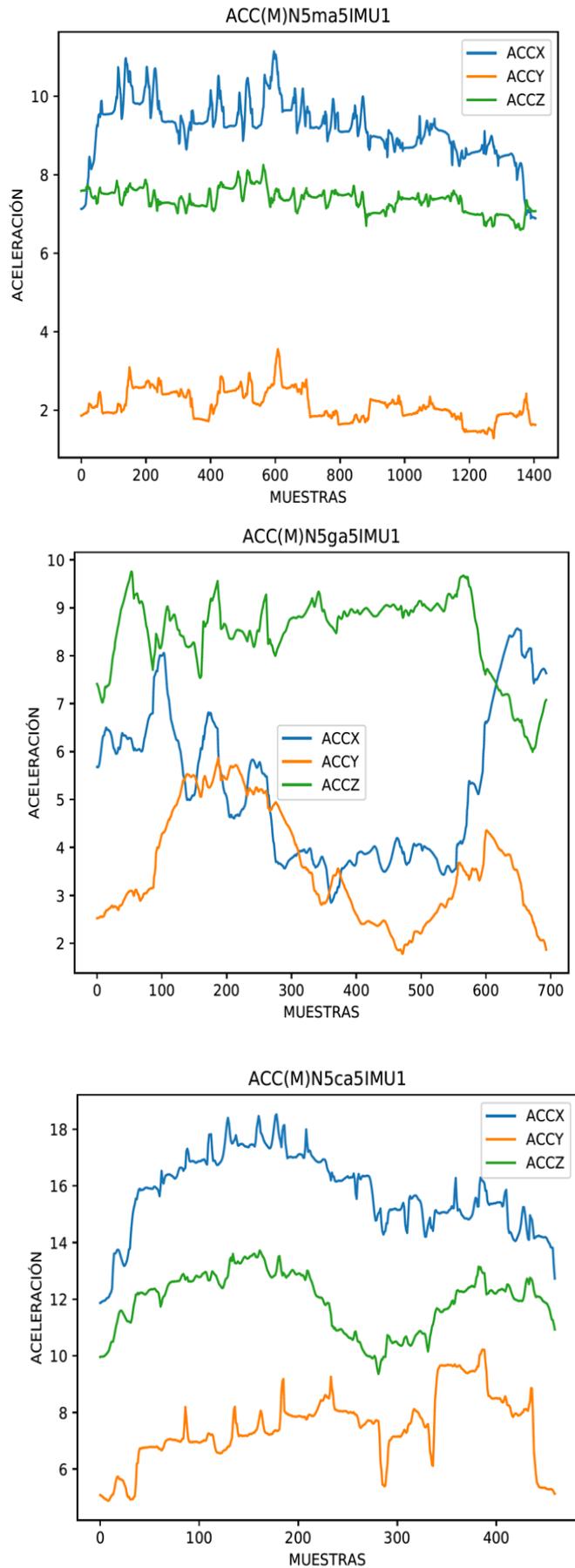


Figura 9. Gráficas del valor eficaz del usuario 5 en el ejemplo 5 e IMU 1 para las tres diferentes actividades.

Durante la realización del gateo la dirección que mayores valores obtuvo fue la del eje “z”, tras ella el eje “x”, y finalmente, el eje “y”. Estos cambios respecto al anterior se produjeron por la posición en la que se llevó a cabo. Al tener las cuatro extremidades apoyadas, el desplazamiento provocaba que los pies y las manos fueran hacía dentro y hacía a fuera. Además, los valores de aceleración alcanzados y, por tanto, de valor eficaz fueron menores que en las otras dos actividades.

En el caso de la carrera, se siguió la misma distribución que en la marcha, con varios cambios que afectaban a los valores. En general, para los tres ejes, los valores eran superiores y en este caso, los del eje “y” eran más próximos a los de los otros dos ejes que en el caso de la marcha. Esto era debido a que el movimiento de la espalda durante la carrera era mayor.

Es importante resaltar que, aunque solo se mencione este ejemplo, la tendencia y la distribución fue muy similar para todas las señales, variando únicamente los valores.

Tras la obtención del valor eficaz el próximo paso fue la obtención de los parámetros estadísticos de media y desviación típica para cada una de las 49 agrupaciones en función de los ejemplos de los que se partían. Cabe resaltar que se pasaron de 178 datos a 49 agrupaciones ya que al realizar las medias y las desviaciones por grupos de datos estos se unieron y así se disminuyó el número de datos.

Los resultados obtenidos se muestran en la Figura 10 y el análisis extraído fue el siguiente:

En términos generales, las desviaciones se encontraban en un intervalo de 0 a 2 para las tres direcciones del espacio. Esto indicaba que no había una gran dispersión de los datos obtenidos, encontrándose todos en un mismo rango de valores y próximos al valor medio. Solamente se encontró una excepción en la desviación del eje “x” en el usuario 4, en concreto, en la realización de la actividad de carrera. Este valor más elevado de desviación (Figura 10) indicaba que los valores obtenidos distaban de los valores medios y que entre los propios datos existían cambios. Teniendo en cuenta que fue el usuario que mayores medias había obtenido para las tres actividades, es lógico que se produjera este resultado, ya que la rapidez con la que las había realizado superaba al resto de compañeros. Esto se veía afectado por el hecho de que fuera el paciente con más edad (Tabla 3) y su control de los movimientos le hubiera permitido establecer una mayor aceleración en la realización de las actividades. Además, al realizar varios ensayos, este pudo haber realizado la misma actividad con distinta intensidad y eso haber influido en el resultado obtenido. Estos datos fueron realmente útiles para establecer probabilidades de que ocurriera una actividad u otra.

Las medias de los datos en los tres ejes, seguían distribuciones similares a las que se han explicado de las gráficas obtenidas para el valor eficaz. Esto resultó así por que al final la media de los datos no fue más que la suma de todos dividido entre el total de datos y, por tanto, se respetó lo obtenido para el valor eficaz. Así, se obtuvo que los valores de aceleración tanto para la marcha como para la carrera eran superiores que para el gateo, y a su vez los datos de carrera eran superiores que los de marcha. Estos datos fueron lógicos debido a las intensidades que se alcanzaron al realizar cada una de las actividades.

	DX	DY	DZ	IMU	MX	MY	MZ	Usuario	\ activ
0	2.046546	0.765392	1.649601	1	17.356427	5.517604	12.296189	2	ca
1	1.596160	0.571308	1.239737	2	17.850638	5.459697	13.580289	2	ca
2	0.541129	0.743653	1.322188	3	10.752574	3.256112	4.202144	2	ca
3	3.202721	1.920923	1.479357	1	20.142914	12.576431	17.176020	3	ca
4	2.965518	1.288384	2.016981	2	21.556373	11.722245	16.561353	3	ca
5	0.532246	0.879582	0.612314	3	12.335744	6.935871	5.595337	3	ca
6	6.292578	2.373972	4.103499	1	22.556414	10.126668	17.471033	4	ca
7	7.029090	2.931636	4.153746	2	22.376548	10.580067	18.076213	4	ca
8	1.319887	1.423205	1.869882	3	11.992140	6.116281	6.778724	4	ca
9	2.822113	2.241331	1.977197	1	13.939991	5.311296	10.535342	5	ca
10	2.307859	1.899579	1.815023	2	13.888656	5.103437	9.385013	5	ca
11	0.777770	1.521566	2.111069	3	10.416513	3.244395	4.190544	5	ca
12	2.788416	1.545208	1.527579	1	17.870206	7.009479	11.297521	6	ca
13	2.365429	1.545615	1.148989	2	17.431566	6.911141	10.476106	6	ca
14	4.652942	1.948851	2.611305	1	25.408434	9.954269	17.972185	7	ca
15	3.803208	2.639015	2.951677	2	23.422075	8.371299	17.319415	7	ca
16	0.759933	1.326239	1.289952	3	11.593816	5.555396	5.722791	7	ca
17	3.072632	1.582522	1.624114	1	4.864384	3.077160	8.371789	1	ga
18	3.223695	2.147618	1.844953	2	4.293424	3.524144	8.392460	1	ga
19	3.629337	1.035545	1.017071	1	5.516198	3.690242	8.419226	3	ga
20	4.117378	1.014211	0.964389	2	5.178253	3.159054	8.830453	3	ga
21	2.917089	0.602278	2.796490	3	5.023975	2.188653	7.173344	3	ga
22	1.653752	1.270504	1.013802	1	7.578336	3.926390	8.536208	4	ga
23	2.100920	1.441418	1.121998	2	7.859851	5.034333	8.485593	4	ga
24	2.506563	1.334722	2.318420	3	4.513988	4.471895	7.188943	4	ga
25	3.292280	1.539704	1.197761	1	5.591636	3.215238	9.113440	5	ga
26	3.276268	1.580157	1.123049	2	5.099214	3.760466	9.071136	5	ga
27	2.288260	1.098633	1.242047	3	4.010948	2.909341	8.621859	5	ga
28	2.642010	1.416741	1.258131	1	4.330747	3.463801	8.232583	6	ga
29	3.109598	1.679202	1.551426	2	4.577384	3.759971	7.764254	6	ga
30	1.847670	1.281624	0.879671	1	8.021659	4.827019	8.634229	7	ga
31	1.790389	1.803330	1.404453	2	9.075316	4.998132	7.982369	7	ga
32	1.629358	0.701448	1.013579	3	3.515231	3.671347	8.654584	7	ga
33	1.853881	1.573692	0.790372	1	11.250232	4.175759	7.133014	1	ma
34	1.742769	1.259003	1.054423	2	11.684745	3.431800	7.173569	1	ma
35	1.827325	1.223484	0.858890	1	10.225800	3.016731	6.965882	2	ma
36	1.538422	1.197330	0.971884	2	10.250475	2.888986	7.262725	2	ma
37	0.262504	0.511412	0.738313	3	9.829690	1.491099	1.827476	2	ma
38	1.591885	1.344600	1.246275	1	12.827918	4.133591	7.971243	4	ma
39	1.929314	1.425663	1.408150	2	13.016376	3.824108	7.933409	4	ma
40	0.553899	0.639449	0.917805	3	9.261708	2.107688	3.783073	4	ma
41	0.847350	0.551716	0.373940	1	8.883636	2.046869	7.139409	5	ma
42	0.684767	0.580687	0.381198	2	9.061380	1.926821	6.545638	5	ma
43	0.092779	0.200590	0.231324	3	9.826012	1.362211	1.175396	5	ma
44	2.190077	1.453982	1.257536	1	12.177859	4.329566	7.323995	6	ma
45	2.314218	1.395867	1.116543	2	12.416324	4.076998	7.371055	6	ma
46	1.629924	1.061555	1.081974	1	11.284197	3.291363	7.593753	7	ma
47	1.652751	0.868235	0.888114	2	11.122587	3.456955	7.411113	7	ma
48	1.216187	0.457694	1.869450	3	9.365154	1.840996	2.693216	7	ma

Figura 10. Media y desviación típica de los ejemplos obtenidos. DX = desviación típica en el eje X; DY = desviación típica en el eje Y; DZ = desviación típica en el eje Z; MX = media en el eje X; MY = media en el eje Y; MZ = media en el eje Z.

Teniendo en cuenta que el valor en los IMU 1 y 2 correspondía al pie derecho e izquierdo respectivamente, y que el IMU 3 correspondía al sensor colocado en la espalda, se observó como en los dos primeros los valores eran más elevados que en el último en las actividades de carrera y marcha. La posición en la que se realizaban estas actividades era la culpable de estos resultados. En la actividad de gateo, la diferencia entre IMUS era menos pronunciada. Además, en las actividades de carrera y marcha también eran más elevados los valores obtenidos, lo cuál concordaba con que ambas actividades alentaban a llevar una velocidad superior y, por tanto, una aceleración también superior que se veía reflejada en la media de los datos.

Con estos datos procesados el siguiente paso fue la creación y el ajusta del modelo a través del método de análisis discriminante lineal se podían extraer los coeficientes que permitían saber como influían cada una de las variables independientes: MX, MY, MZ, DX, DY, DZ E IMU en las tres diferentes actividades: carrera, gateo y marcha.

Los resultados obtenidos fueron los siguientes:

```
In [14]: clf.coef_
Out [14]:
array([[ 1.32581843,  0.31658655, -0.6439652 , -1.45496462,  0.45413018,
         1.24646919,  0.52338222],
       [-2.43311462,  0.61617583,  1.4175573 ,  1.0267334 , -0.59553518,
         0.17295183, -0.09168682],
       [ 1.09272805, -1.01605231, -0.78223389,  0.55377761,  0.12055666,
        -1.59714703, -0.49536724]])
```

Figura 11. Coeficientes del modelo ajustado obtenidos a través de Python.

Se apreciaba que en mayor o menor medida todas las variables influían en las diferentes actividades, ya que todos los valores se encuentran por encima del 0. Aunque existen valores negativos, estos hacían referencia a la dirección.

Tabla 10. Coeficientes de la función discriminatoria de la actividad obtenidos a través de la función LDA

VARIABLE/ACTIVIDAD	CARRERA	GATEO	MARCHA
MEDIA EJE X	1.326	-2.433	1.093
MEDIA EJE Y	0.317	0.616	-1.016
MEDIA EJE Z	-0.644	1.42	-0.782
DESVIACIÓN EJE X	-1.455	1.027	0.554
DESVIACIÓN EJE Y	0.45	-0.596	0.121
DESVIACIÓN EJE Z	1.246	0.173	-1.597
IMU (sensor)	0.523	-0.191	-0.495

Analizando los resultados (Tabla 8), se extrajo que:

La colocación del IMU no fue muy influyente en ninguna de las actividades propuestas.

En la actividad de carrera, las variables más influyentes y, por tanto, que determinaban esta actividad eran la media en eje "x" y las desviaciones de ese mismo eje y el "z". En la de marcha, los valores de los coeficientes indicaban que las variables más influyentes eran la media en eje "x" e "y" y la desviación típica en el eje "z". Finalmente, en la del gateo eran la media en el eje "x" y "z" y la desviación en el eje "x", los más influyentes.

El avance de estas actividades se realizaba en la dirección "x", como ya se ha comentado, era por este motivo, por el cual en los tres casos la media en el eje "x" fue verdaderamente interesante. El que se obtuvieran, además, otras variables determinantes, estuvo ocasionado por la posición en la que se realizan las diversas acciones. En cuanto a la carrera, los movimientos realizados con el cuerpo para llevar a cabo la acción implicaban un mayor movimiento angular en los tobillos, rodillas y cadera, y por ello fueron elevadas las desviaciones en el eje "x" y "z". En cuanto a la marcha, aunque también se realizaba en posición erguida, el movimiento era más estático y los apoyos producidos eran diferentes. En las dos actividades, la desviación en el eje "z" se produjo por que no se llevaban a cabo estas acciones en línea recta y esto provocaba cambios en esa dirección. En cuanto al gateo, su posición fue totalmente distinta puesto que las cuatro extremidades estaban apoyadas durante su ejecución. La dirección "z" marcaba la profundidad y, al realizar el movimiento, se tendía a mover los pies de dentro hacia a fuera seguidamente, a la vez que se producía desplazamiento. Es por ello, que este movimiento se convirtió en un dato importante, aunque en menor medida que el de desplazamiento hacia delante y hacia atrás. La media en el eje "y" también influía, puesto que la espalda se encontraba inclinada hacia delante y también se movía al llevar a cabo la acción.

Con el modelo ajustado, también fue posible obtener las probabilidades a priori de cada una de las actividades. Estas fueron:

```
In [9]: print(clf.priors_)  
[ 0.35416667  0.33333333  0.3125   ]
```

Figura 12. Probabilidades a priori de cada una de las actividades de carrera, gateo y marcha, respectivamente.

Para ser 100% equitativos, los porcentajes debían ser de 0.333 en todos los casos. Sin embargo, para la marcha y la carrera esto no ocurría, aunque la diferencia era poca. Esto fue así porque había menos medidas de marcha que de las demás actividades tal y como se ha expuesto en la obtención de los datos.

4.2.2 Validación de los clasificadores

Tras el proceso de validación, se pudo pasar a la evaluación de la calidad del clasificador y su capacidad para clasificar correctamente cada una de las actividades. Para ello, se extrajo tanto la matriz de confusión como el valor de la exactitud.

La matriz de confusión, como ya se ha matizado en el apartado anterior, se creó cuando se obtuvieron los valores predichos y reales y se observaron las coincidencias y discrepancias entre los valores. Un ejemplo de esta comparativa es el que se muestra en la Figura 13, donde se remarca en azul una mala predicción del clasificador. En ella se observa que se confundió la marcha con la carrera.

```

('predicho:', u'ca', 'real:', u'ca')
('predicho:', u'ca', 'real:', u'ca')
('predicho:', u'ma', 'real:', u'ca')
('predicho:', u'ca', 'real:', u'ca')
    
```

Figura 13. Comparación entre valores predichos y reales

La suma de ese ejemplo y la matriz de confusión que se obtuvo (Figura 14), permitió observar que, de las observaciones mal predichas, 5, 4 se producían por dicho motivo. Es decir, en prácticamente todos los casos, se había confundido la actividad de carrera con la de marcha. Esto pudo venir suscitado por la velocidad a la que se producía la actividad. Los niños de entre 1 y 5 años, realizaban, en general, las acciones de carrera y marcha de forma muy similares y esto pudo suscitar fácilmente confusiones. El otro caso encontrado, se produjo entre el gateo y carrera. Esta confusión pudo estar determinada por que algún participante dominara a la perfección el gateo y lo hiciera a una velocidad mayor a la esperada llegando a aproximarse a valores dentro de la desviación aceptada para los valores de carrera.

	ma	ga	ca
ma	13	0	4
ga	0	15	1
ca	0	0	16

Figura 14. Matriz de confusión. Las columnas son los valores reales mientras que las filas son los predichos. ma= marcha; ga = gateo; ca = carrera

En cuanto al valor de exactitud, el porcentaje obtenido fue de un 89.79 %. Se consideró un valor óptimo pues indicaba que solamente iba a fallar en un 10% de las veces, porcentaje que se podía asumir y que cumplía con las expectativas presentadas.

Si se compara este valor obtenido con los estudios previamente obtenidos (Tabla 1), se encuentra que:

Cuando se emplean técnicas de características similares a las del algoritmo creado, es decir, estadísticas, el porcentaje de exactitud obtenido varían entre un 52 a un 87%, quedando el clasificador creado por encima de estos valores. Hay que añadir que en todos esos casos el número de actividades a clasificar era más elevado que en este trabajo, de ahí probablemente que el porcentaje sea un poco superior. El que mayor porcentaje de exactitud ha obtenido, 87%, clasificaba 6 actividades mientras que el trabajo presente sólo se clasifican tres obteniendo un parámetro de exactitud casi tres puntos superior. Esto permite afirmar que el clasificador creado lleva a cabo correctamente su objetivo, encontrándose su porcentaje de acierto de actividad dentro de un rango óptimo.

Teniendo en cuenta el número de actividades clasificadas y comparándolo con otro estudio que clasificaba el mismo número de actividades, la exactitud obtenida es ligeramente inferior, pasando de un 90.8 % del estudio revisado a un 89.79% en que se describe, pero la técnica empleada es distinta pues en el primero, se utilizaron para clasificar técnicas basadas en umbrales, con lo cual los datos podrían ser equiparables.

Finalmente, en comparación con otros clasificadores que emplean técnicas más complejas y sofisticadas, el valor de exactitud obtenido con este clasificador desciende alguna unidad. No obstante, esto puede deberse a que el número de actividades clasificadas en los complejos es mayor, lo que provoca que se obtengan más medidas, permitiendo un periodo de entrenamiento mayor y un aumento de su capacidad de generalización. También puede verse influido por la complejidad del modelo que permite un análisis más minucioso de los datos.

Aún así el valor obtenido es elevado e indica que la probabilidad de que ante un nuevo caso la clasificación se realice correctamente sea elevada.

CAPÍTULO V: CONCLUSIONES

En el siguiente apartado, se exponen las conclusiones del trabajo que cierran con los objetivos especificados en el apartado 1.3:

- Tras el estudio de las técnicas de clasificación, la revisión de su aplicación en la clasificación de actividades, el uso de la acelerometría y considerando la forma en la que se presentan los datos obtenidos en el ensayo, se puede concluir que las técnicas estadísticas son la mejor opción para la creación de un clasificador.
- Los parámetros de media y desviación típica de las aceleraciones obtenidas a través de los sensores triaxiales han permitido la creación de una función discriminadora lineal que ha sido ajustada haciendo uso de un análisis discriminante lineal. A través de él, es posible caracterizar el gateo, la marcha y la carrera en niños de 1 a 5 años. Las medias y desviaciones que más influyen en cada actividad son distintas y esto es consecuencia de la postura y los movimientos que se llevan a cabo durante sus realizaciones.
- La matriz de confusión obtenida muestra que el clasificador creado discrimina correctamente entre clases. También se ha observado que los casos que pueden llevar a confusión son cuando la carrera se ha llevado a una velocidad más lenta de la que cabría esperar y por ello se ha podido confundir con la marcha.
- La exactitud obtenida indica que el ajuste llevado a cabo y la fidelidad de los resultados obtenidos es alta y correcta si se compara con los resultados obtenidos en clasificadores con las mismas características que el creado en este trabajo.
- Tras un proceso de validación, el algoritmo creado ha sido capaz de extraer los patrones escondidos en los datos. De esta forma ante un nuevo dato no etiquetado, pero del que se conocen los valores de media y desviación típica, este es capaz de asignar la actividad que se está llevando a cabo: gateo, marcha o carrera y poder evaluarse.

CAPÍTULO VI: PRESUPUESTO

En esta sección del trabajo se pretende estimar el coste que hubiera supuesto el proyecto si se hubiera realizado de forma profesional.

5.1 PRESUPUESTO

La división del presupuesto consta de tres partes: la mano de obra, el material del ensayo y el material del estudio. Cada uno de ellos, va a ser detallado a continuación.

5.1.1 MANO DE OBRA

Para la realización del proyecto, se ha requerido del trabajo y ayuda de dos profesionales y una alumna.

El desarrollo del trabajo ha supuesto 300 horas de trabajo por parte de la alumna. Estas horas han sido repartidas en función de la demanda de tareas y necesidades, es decir, no ha habido un horario de trabajo predeterminado, aunque sí, una serie de horas que cumplir. Pensando que se trata de una estudiante en práctica, se ha supuesto que el precio a pagar por cada hora trabajada es de 10 euros que multiplicado por las 300 horas de trabajo hacen un total de 3000 euros.

La alumna ha contado con el apoyo y la ayuda de un tutor. Se trata de un profesional del sector, que conoce a la perfección el tema a tratar y que ha ido guiando a la alumna para que el desarrollo del trabajo fuera correcto y favorable. Las horas que ha empleado se estima que han sido aproximadamente 150 que a 35 euros cada una de ellas (debido a su categoría y su puesto de trabajo) suponen un gasto total de 5250 euros.

Durante la realización del estudio previo, se contó con la presencia de una profesional. Ella organizó y planificó los pasos a seguir, para que la extracción de datos fuera exitosa y después fue la encargada de explicar a alumna como se debía llevar a cabo el proceso estando con ella en todos los ensayos realizados con los participantes. Se ha calculado que las horas en que su presencia ha sido requerida ha sido 20 y que el precio a pagar por cada una de ellas es de 25 euros.

La siguiente tabla, resume todos los gastos asociados a la mano de obra y el subtotal obtenido.

Tabla 11. Costes mano de obra

Mano Obra	Cantidad (horas)	Coste unitario (€/h)	Coste total (€)
<i>Estudiante en prácticas</i>	300	10	3000.00
<i>Tutor</i>	150	35	5250.00
<i>Técnico</i>	20	25	500.00
<i>Subtotal</i>			8750.00

5.1.2 MATERIAL

El material asociado al coste proyecto, se puede dividir en dos partes. Una primera parte, tiene en cuenta el material necesario para la obtención de los datos; y una segunda, el del desarrollo del proyecto.

Cabe resaltar que para la obtención de esta parte del presupuesto se ha tenido en cuenta el precio de venta al público al que el fabricante suministra el producto empleado.

5.1.2.1 Material para el ensayo

El material que se necesitó para la realización del ensayo fue:

- Sensores corporales Xsens, capaces de recoger datos de aceleración en las tres direcciones del espacio (X, Y, Z). Se necesitaron 3: uno para cada pie y otro para la cintura. A su vez, para poder transmitir los datos y cargar estos sensores, se requirió de una base y de un pen drive para poder trasladarlos a otras computadoras.
- Calzado infantil. Los pares de zapatos empleados, fueron de la marca Biomechanics, puesto que es con la que colabora el IBV. Debido a la diferencia de tallas de los participantes hubo que tener cuatro pares de diferentes tallas disponibles. Además, por el hecho de emplear su tiempo en asistir al ensayo y como agradecimiento por su colaboración, se les obsequió al final del ensayo con un detalle.

La tabla 12, recoge todos los materiales, su coste unitario y el coste total que han supuesto para el proyecto.

Tabla 12. Costes asociados el ensayo para la obtención de datos

Material ensayo	Concepto	Cantidad	Coste unitario (€)	Coste total (€)
<i>Xsens</i>	Sensor Corporal	3 u.	400.00 []	1200.00
<i>Base sensores</i>	Carga sensores y transmisión de datos	1 u.	900.00[]	900.00
<i>Biomechanics</i>	Calzado infantil	4 u.	47.00 []	188.00
<i>USB 64 Gb</i>	Almacén datos	1 u.	14.50 []	14.50
<i>Detalle a los participantes</i>	Agradecimiento	7 u.	15.00 []	105.00
<i>Subtotal</i>				2407.50

5.1.2.2 Material para el estudio y desarrollo del proyecto

Para poder conseguir el objetivo y desarrollar un sistema de clasificación, se ha necesitado el uso de diferentes softwares y de una computadora.

Durante la obtención de los datos, se requería de un software de lectura. Este no ha supuesto un gasto extra ya que iba incluido en el precio de los sensores.

El proyecto se ha desarrollado con el software Python, que es gratuito y por tanto tampoco ha supuesto un coste adicional. Pero si que lo ha supuesto la compra de la licencia de office necesaria para el proceso de escritura del trabajo.

La tabla 9 resume el precio de esta sección.

Tabla 13. Costes asociados al desarrollo del clasificador

Material estudio	Concepto	Cantidad	Coste unitario (€)	Coste total (€)
<i>MacBook Pro</i>	Ordenador personal	1 u.	1300.00 []	1300.00
<i>Paquete Office</i>	Licencia 1 año	1 u.	129.00[]	129.00
<i>Subtotal</i>				1429.00

Tras el desglosamiento de cada uno de los precios, la Tabla 14 recoge, el subtotal de cada una de las tres secciones y precio total del proyecto. Este asciende a 12586.50 euros.

Tabla 14. Coste total del proyecto

Tipo de coste	Coste total (€)
<i>Mano de obra</i>	8750.00
<i>Material ensayo</i>	2407.50
<i>Material estudio</i>	1429.00
TOTAL	12586.50

CAPÍTULO VII: BIBLIOGRAFÍA

«1. Supervised learning — scikit-learn 0.19.1 documentation». s. f. Accedido 3 de julio de 2018. http://scikit-learn.org/stable/supervised_learning.html#supervised-learning.

«71014023_RTfY9Vz.pdf». s. f. Accedido 3 de julio de 2018. https://www.sanzytorres.es/static/pdf/71014023_RTfY9Vz.pdf.

«A study on the activity classification using a triaxial accelerometer - IEEE Conference Publication». s. f. Accedido 3 de julio de 2018. <https://ieeexplore.ieee.org/document/1280534/>.

«Activity recognition from accelerometer data». s. f. Accedido 3 de julio de 2018. <https://dl.acm.org/citation.cfm?id=1620107>.

«alimentacion_escolar.pdf». s. f. Accedido 3 de julio de 2018. https://www.aeped.es/sites/default/files/documentos/alimentacion_escolar.pdf.

Allen, Felicity R., Eliathamby Ambikairajah, Nigel H. Lovell, y Branko G. Celler. 2006a. «An Adapted Gaussian Mixture Model Approach to Accelerometry-Based Movement Classification Using Time-Domain Features». *Conference Proceedings: ... Annual International Conference of the IEEE Engineering in Medicine and Biology Society. IEEE Engineering in Medicine and Biology Society. Annual Conference 1*: 3600-3603. <https://doi.org/10.1109/IEMBS.2006.259613>.

Allen, Felicity R, Eliathamby Ambikairajah, Nigel H Lovell, y Branko G Celler. 2006b. «Classification of a Known Sequence of Motions and Postures from Accelerometry Data Using Adapted Gaussian Mixture Models». *Physiological Measurement* 27 (10): 935-51. <https://doi.org/10.1088/0967-3334/27/10/001>.

Alvis-Chirinos, Katherine, Lucio Huamán-Espino, Jenny Pillaca, y Juan Pablo Aparco. 2017. «Medición de la actividad física mediante acelerómetros triaxiales en escolares de tres ciudades del Perú». *Revista Peruana de Medicina Experimental y Salud Pública* 34 (1): 28. <https://doi.org/10.17843/rpmesp.2017.341.2764>.

«API Reference — scikit-learn 0.19.1 documentation». s. f. Accedido 3 de julio de 2018. http://scikit-learn.org/stable/modules/classes.html#module-sklearn.discriminant_analysis.

«Assessment of a Smart Sensing Shoe for Gait Phase Detection in LevelWalking - Buscar con Google». s. f. Accedido 3 de julio de 2018. <https://www.google.es/search?q=Assessment+of+a+Smart+Sensing+Shoe+for+Gait+Phase+Detection+in+LevelWalking&oq=Assessment+of+a+Smart+Sensing+Shoe+for+Gait+Phase+Detection+in+LevelWalking&aqs=chrome..69i57.532j0j4&sourceid=chrome&ie=UTF-8>.

Bao, Ling, y Stephen S. Intille. 2004. «Activity Recognition from User-Annotated Acceleration Data». En *Pervasive Computing*, editado por Alois Ferscha y Friedemann Mattern, 3001:1-17. Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-24646-6_1.

Barber, Sally E., Cath Jackson, Shaheen Akhtar, Daniel D. Bingham, Hannah Ainsworth, Catherine Hewitt, Gerry Richardson, et al. 2013. «“Pre-Schoolers in the Playground” an Outdoor Physical Activity Intervention for Children Aged 18 Months to 4 Years Old: Study Protocol for a Pilot Cluster Randomised Controlled Trial». *Trials* 14 (octubre): 326. <https://doi.org/10.1186/1745-6215-14-326>.

Bassett, David R., Lindsay P. Toth, Samuel R. LaMunion, y Scott E. Crouter. 2017. «Step Counting: A Review of Measurement Considerations and Health-Related Applications». *Sports Medicine (Auckland, N.z.)* 47 (7): 1303-15. <https://doi.org/10.1007/s40279-016-0663-1>.

Bilgin, Suleyman. 2017. «The Comparison of Neurodegenerative Diseases and Healthy Subjects Using Discrete Wavelet Transform in Gait Dynamics». *Journal of Medical and Bioengineering* 6 (1): 4.

Bingham, Daniel D., Silvia Costa, Stacy A. Clemes, Ash C. Routen, Helen J. Moore, y Sally E. Barber. 2016. «Accelerometer Data Requirements for Reliable Estimation of Habitual Physical Activity and Sedentary Time of Children during the Early Years - a Worked Example Following a Stepped Approach». *Journal of Sports Sciences* 34 (20): 2005-10. <https://doi.org/10.1080/02640414.2016.1149605>.

Bingham, Daniel D., Silvia Costa, Trina Hinkley, Katy A. Shire, Stacy A. Clemes, y Sally E. Barber. 2016a. «Physical Activity During the Early Years: A Systematic Review of Correlates and Determinants». *American Journal of Preventive Medicine* 51 (3): 384-402. <https://doi.org/10.1016/j.amepre.2016.04.022>.

———. 2016b. «Physical Activity During the Early Years: A Systematic Review of Correlates and Determinants». *American Journal of Preventive Medicine* 51 (3): 384-402. <https://doi.org/10.1016/j.amepre.2016.04.022>.

«Cap 7.pdf». s. f. Accedido 3 de julio de 2018. <http://amexcomp.mx/files/libro/Cap%207.pdf>.

«CLIR-002-16.pdf». s. f. Accedido 3 de julio de 2018a. <https://www.clir-lab.org/repository/Documents/CLIR-002-16.pdf>.

———. «CLIR-002-16.pdf». s. f. Accedido 3 de julio de 2018b. <https://www.clir-lab.org/repository/Documents/CLIR-002-16.pdf>.

Del Rosario, Michael B, Kejia Wang, Jingjing Wang, Ying Liu, Matthew Brodie, Kim Delbaere, Nigel H Lovell, Stephen R Lord, y Stephen J Redmond. 2014. «A comparison of activity classification in younger and older cohorts using a smartphone». *Physiological Measurement* 35 (11): 2269-86. <https://doi.org/10.1088/0967-3334/35/11/2269>.

Dyson, M., S. J. Duckett, y F. C. Allen. 2000. «A Therapy-Relevant Casemix Classification System for School-Age Children with Disabilities». *Archives of Physical Medicine and Rehabilitation* 81 (5): 634-43.

«Electronics | Free Full-Text | Assessment of a Smart Sensing Shoe for Gait Phase Detection in Level Walking». s. f. Accedido 3 de julio de 2018. <http://www.mdpi.com/2079-9292/5/4/78>.

Ermes, M., J. Pärkkä, J. Mäntyjärvi, y I. Korhonen. 2008. «Detection of Daily Activities and Sports With Wearable Sensors in Controlled and Uncontrolled Conditions». *IEEE Transactions on*

Information Technology in Biomedicine 12 (1): 20-26.
<https://doi.org/10.1109/TITB.2007.899496>.

Ermes, M., J. Parkka, J. Mantyjarvi, y I. Korhonen. 2008. «Detection of Daily Activities and Sports With Wearable Sensors in Controlled and Uncontrolled Conditions». *IEEE Transactions on Information Technology in Biomedicine* 12 (1): 20-26.
<https://doi.org/10.1109/TITB.2007.899496>.

«Evaluation of accelerometer based multi-sensor versus single-sensor activity recognition systems - ScienceDirect». s. f. Accedido 3 de julio de 2018.
<https://www.sciencedirect.com/science/article/pii/S1350453314000344>.

Gao, L., A. K. Bourke, y J. Nelson. 2011. «A system for activity recognition using multi-sensor fusion». En *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 7869-72. <https://doi.org/10.1109/IEMBS.2011.6091939>.

———. 2012a. «A comparison of classifiers for activity recognition using multiple accelerometer-based sensors». En *2012 IEEE 11th International Conference on Cybernetic Intelligent Systems (CIS)*, 149-53. <https://doi.org/10.1109/CIS.2013.6782169>.

———. 2012b. «Activity recognition using dynamic multiple sensor fusion in body sensor networks». En *2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 1077-80. <https://doi.org/10.1109/EMBC.2012.6346121>.

Hallems, Ann, Lin Dhanis, Dirk De Clercq, y Peter Aerts. 2007. «Changes in Mechanical Control of Movement during the First 5 Months of Independent Walking: A Longitudinal Study». *Journal of Motor Behavior* 39 (3): 227-38. <https://doi.org/10.3200/JMBR.39.3.227-240>.

Henderson, Carol J., Daniel J. Lovell, Bonny L. Specker, y Barbara N. Campaigne. s. f. «Physical Activity in Children with Juvenile Rheumatoid Arthritis: Quantification and Evaluation». *Arthritis & Rheumatism* 8 (2): 114-19. <https://doi.org/10.1002/art.1790080210>.

«IB_T4_OCW.pdf». s. f. Accedido 3 de julio de 2018. http://ocw.uv.es/ingenieria-y-arquitectura/1-5/ib_material/IB_T4_OCW.pdf.

«Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring - IEEE Journals & Magazine». s. f. Accedido 3 de julio de 2018a. <https://ieeexplore.ieee.org/document/1573717/>.

———. «Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring - IEEE Journals & Magazine». s. f. Accedido 3 de julio de 2018b. <https://ieeexplore.ieee.org/document/1573717/>.

Jeng, S. F., K. I. Yau, L. C. Chen, y S. F. Hsiao. 2000. «Alberta Infant Motor Scale: Reliability and Validity When Used on Preterm Infants in Taiwan». *Physical Therapy* 80 (2): 168-78.

Karantonis, D.M., M.R. Narayanan, M. Mathie, N.H. Lovell, y B.G. Celler. 2006. «Implementation of a Real-Time Human Movement Classifier Using a Triaxial Accelerometer for Ambulatory Monitoring». *IEEE Transactions on Information Technology in Biomedicine* 10 (1): 156-67. <https://doi.org/10.1109/TITB.2005.856864>.

Kern, Nicky, Bernt Schiele, y Albrecht Schmidt. 2003. «Multi-Sensor Activity Context Detection for Wearable Computing». En *Ambient Intelligence*, editado por Emile Aarts, René W. Collier,

Evert van Loenen, y Boris de Ruyter, 2875:220-32. Berlin, Heidelberg: Springer Berlin Heidelberg. https://doi.org/10.1007/978-3-540-39863-9_17.

Lester, J., T. Choudhury, N. Kern, G. Borriello, y B. Hannaford. 2005. «A Hybrid Discriminative/Generative Approach for Modeling Human Activities». En , 766-722. Edinburgh, Scotland. http://brl.ee.washington.edu/BRL_Pubs/Pdfs/Rep198.pdf.

Lötters, J. C., J. Schipper, P. H. Veltink, W. Olthuis, y P. Bergveld. 1998. «Procedure for in-use calibration of triaxial accelerometers in medical applications». *Sensors and Actuators A: Physical*, Eurosensors XI, 68 (1): 221-28. [https://doi.org/10.1016/S0924-4247\(98\)00049-1](https://doi.org/10.1016/S0924-4247(98)00049-1).

Mannini, Andrea, y Angelo Maria Sabatini. 2010. «Machine Learning Methods for Classifying Human Physical Activity from On-Body Accelerometers». *Sensors (Basel, Switzerland)* 10 (2): 1154-75. <https://doi.org/10.3390/s100201154>.

Mathie, M. J., B. G. Celler, N. H. Lovell, y A. C. F. Coster. 2004. «Classification of Basic Daily Movements Using a Triaxial Accelerometer». *Medical & Biological Engineering & Computing* 42 (5): 679-87.

Mathie, M. J., A. C. F. Coster, N. H. Lovell, y B. G. Celler. 2003a. «Detection of Daily Physical Activities Using a Triaxial Accelerometer». *Medical & Biological Engineering & Computing* 41 (3): 296-301. <https://doi.org/10.1007/BF02348434>.

———. 2003b. «Detection of Daily Physical Activities Using a Triaxial Accelerometer». *Medical & Biological Engineering & Computing* 41 (3): 296-301. <https://doi.org/10.1007/BF02348434>.

Mathie, Merryn J., Adelle C. F. Coster, Nigel H. Lovell, y Branko G. Celler. 2004a. «Accelerometry: Providing an Integrated, Practical Method for Long-Term, Ambulatory Monitoring of Human Movement». *Physiological Measurement* 25 (2): R1-20.

———. 2004b. «Accelerometry: Providing an Integrated, Practical Method for Long-Term, Ambulatory Monitoring of Human Movement». *Physiological Measurement* 25 (2): R1-20.

Mathie, Merryn J., Adelle C. F. Coster, Nigel H. Lovell, Branko G. Celler, Stephen R. Lord, y Anne Tiedemann. 2004. «A Pilot Study of Long-Term Monitoring of Human Movements in the Home Using Accelerometry». *Journal of Telemedicine and Telecare* 10 (3): 144-51. <https://doi.org/10.1258/135763304323070788>.

«Maturation of gait dynamics: stride-to-stride variability and its temporal organization in children. - PubMed - NCBI». s. f. Accedido 3 de julio de 2018. <https://www.ncbi.nlm.nih.gov/pubmed/10066721>.

Maurer, Uwe, Anthony Rowe, Asim Smailagic, y Daniel Siewiorek. 2006. «Location and Activity Recognition Using EWatch: A Wearable Sensor Platform». En *Ambient Intelligence in Everyday Life*, 86-102. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11825890_4.

Mayagoitia, Ruth E., Joost C. Lötters, Peter H. Veltink, y Hermie Hermens. 2002. «Standing Balance Evaluation Using a Triaxial Accelerometer». *Gait & Posture* 16 (1): 55-59.

Michalski, R. S., J. G. Carbonell, y T. M. Mitchell. 2013. *Machine Learning: An Artificial Intelligence Approach*. Springer Science & Business Media.

«ml_map.png (1600×998)». s. f. Accedido 3 de julio de 2018. https://2.bp.blogspot.com/-RChN0img2WM/Wgq0O1XnSKI/AAAAAAAAAG0s/joleEQEB6cA-Ni0GPMr3bRAWXb7CGVn8QCLcBGAs/s1600/ml_map.png.

«Monkey: Wearable sensor that keeps kids from wandering too far from their parents – BetaList». s. f. Accedido 3 de julio de 2018. <https://betalist.com/startups/themonkeyapp>.

«Motion Mode Recognition and Step Detection Algorithms for Mobile Phone Users». s. f. Accedido 3 de julio de 2018. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3649428/>.

«Neural Networks: Machine Learning Inspired by the Brain». 2017. Hiring | Upwork. 10 de abril de 2017. <https://www.upwork.com/hiring/data/neural-networks-demystified/>.

«Organización Mundial de la Salud». s. f. Accedido 3 de julio de 2018. <http://www.who.int/es>.

Pärkkä, Juha, Miikka Ermes, Panu Korpiää, Jani Mäntyjärvi, Johannes Peltola, y Ilkka Korhonen. s. f. «Activity classification using realistic data from wearable sensors». En *Cornell University*, 2007–2008. Press.

———. s. f. «Activity classification using realistic data from wearable sensors». En *Cornell University*, 2007–2008. Press.

«(PDF) Activity Identification Using Body-Mounted Sensors — A Review of Classification Techniques». s. f. Accedido 3 de julio de 2018. https://www.researchgate.net/publication/24254330_Activity_Identification_Using_Body-Mounted_Sensors_-_A_Review_of_Classification_Techniques.

«(PDF) Análisis de la marcha humana con plataformas dinamométricas. Influencia del transporte de carga.» s. f. Accedido 3 de julio de 2018. https://www.researchgate.net/publication/236626705_Analisis_de_la_marcha_humana_con_plataformas_dinamometricas_Influencia_del_transprte_de_carga.

«(PDF) Motion Mode Recognition and Step Detection Algorithms for Mobile Phone Users». s. f. Accedido 3 de julio de 2018. https://www.researchgate.net/publication/235102065_Motion_Mode_Recognition_and_Step_Detection_Algorithms_for_Mobile_Phone_Users.

Pirttikangas, Susanna, Kaori Fujinami, y Tatsuo Nakajima. 2006. «Feature Selection and Activity Recognition from Wearable Sensors». En *Ubiquitous Computing Systems*, 516-27. Lecture Notes in Computer Science. Springer, Berlin, Heidelberg. https://doi.org/10.1007/11890348_39.

Ravi, Nishkam, Nikhil Dandekar, Preetham Mysore, y Michael L. Littman. 2005. «Activity Recognition from Accelerometer Data». En *Proceedings of the 17th Conference on Innovative Applications of Artificial Intelligence - Volume 3*, 1541–1546. IAAI'05. Pittsburgh, Pennsylvania: AAAI Press. <http://dl.acm.org/citation.cfm?id=1620092.1620107>.

«Recognizing human motion with multiple acceleration sensors - IEEE Conference Publication». s. f. Accedido 3 de julio de 2018. <https://ieeexplore.ieee.org/document/973004/>.

Rowl, Ann V., Sarah M. Powell, Rhiannon Humphries, y Roger G. Eston. s. f. *Original Article THE EFFECT OF ACCELEROMETER EPOCH ON PHYSICAL ACTIVITY OUTPUT MEASURES*.

Rowlands, Alex, y Roger Eston. 2007. *The Measurement and Interpretation of Children's Physical Activity*. Vol. 6.

Sekine, M., T. Tamura, T. Togawa, y Y. Fukui. 2000. «Classification of Waist-Acceleration Signals in a Continuous Walking Record». *Medical Engineering & Physics* 22 (4): 285-91.

Sekine, Masaki, Toshiyo Tamura, Metin Akay, Toshiro Fujimoto, Tatsuo Togawa, y Yasuhiro Fukui. 2002. «Discrimination of Walking Patterns Using Wavelet-Based Fractal Analysis». *IEEE Transactions on Neural Systems and Rehabilitation Engineering: A Publication of the IEEE Engineering in Medicine and Biology Society* 10 (3): 188-96. <https://doi.org/10.1109/TNSRE.2002.802879>.

«Sensors | Free Full-Text | Classification of Sporting Activities Using Smartphone Accelerometers | HTML». s. f. Accedido 3 de julio de 2018. <http://www.mdpi.com/1424-8220/13/4/5317/htm>.

Steins, Dax, Helen Dawes, Patrick Esser, y Johnny Collett. 2014. «Wearable Accelerometry-Based Technology Capable of Assessing Functional Activities in Neurological Populations in Community Settings: A Systematic Review». *Journal of NeuroEngineering and Rehabilitation* 11 (1): 36. <https://doi.org/10.1186/1743-0003-11-36>.

Steins, Dax, Ian Sheret, Helen Dawes, Patrick Esser, y Johnny Collett. 2014. «A Smart Device Inertial-Sensing Method for Gait Analysis». *Journal of Biomechanics* 47 (15): 3780-85. <https://doi.org/10.1016/j.jbiomech.2014.06.014>.

Trost, Stewart G., Paul D. Loprinzi, Rebecca Moore, y Karin A. Pfeiffer. 2011. «Comparison of accelerometer cut points for predicting activity intensity in youth.» *Medicine and science in sports and exercise* 43 (7): 1360–1368.

Trost, Stewart G., Russell R. Pate, James F. Sallis, Patty S. Freedson, Wendell C. Taylor, Marsha Dowda, y John Sirard. s. f. *and gender differences in objectively measured physical activity in youth*.

Trost, Stewart, Neville Owen, Adrian Bauman, James F Sallis, y Wendy Brown. 2003. *Correlates of adults' participation in physical activity: Review and update*. Vol. 34. <https://doi.org/10.1249/01.MSS.0000038974.76900.92>.

Tseng, Tzu-Liang (Bill), Kalyan Reddy Aleti, Zhonghua Hu, y Yongjin (James) Kwon. 2016. «E-quality control: A support vector machines approach». *Journal of Computational Design and Engineering* 3 (2): 91-101. <https://doi.org/10.1016/j.jcde.2015.06.010>.

Villaverde, Armando Collado, y Maria D R-Moreno. s. f. «Detección de caídas mediante un acelerómetro de tres ejes ubicado en la muñeca en personas de tercera edad», 10.

«Wearable accelerometry-based technology capable of assessing functional activities in neurological populations in community settings: a systematic review | Journal of NeuroEngineering and Rehabilitation | Full Text». s. f. Accedido 3 de julio de 2018. <https://jneuroengrehab.biomedcentral.com/articles/10.1186/1743-0003-11-36>.

Wixted, A. J., D. V. Thiel, A. G. Hahn, C. J. Gore, D. B. Pyne, y D. A. James. 2007. «Measurement of Energy Expenditure in Elite Athletes Using MEMS-Based Triaxial Accelerometers». *IEEE Sensors Journal* 7 (4): 481-88. <https://doi.org/10.1109/JSEN.2007.891947>.

CAPÍTULO VIII: ANEXOS

En el siguiente capítulo se muestran los códigos generados en la realización del trabajo siguiendo los apartados del trabajo.

CLASIFICACIÓN DE LA ACTIVIDAD

```
- RECOGIDA DE DATOS
# -*- coding: utf-8 -*-
"""
Spyder Editor

This is a temporary script file.
"""

from numpy import *
from matplotlib.pyplot import *
import pandas as pd
import os
import glob

#%%
ruta = "/Users/martagarciaibor/Documents/Ingenieriããa
Biomeããdica/CUARTO CURSO/TFG/01_data_XSENS_recortadas"

contenido = os.listdir(ruta)

dirs = [d for d in contenido if os.path.isdir(ruta+"/"+d)]
dirs

for d in dirs:
    fich = glob.glob(ruta+"/"+d+"/*.csv")
    fich

# Vamos a crear un diccionario con los datos (mã°s cabecera)
diccionario = {}
diccionario["N1ga1IMU1"] =
pd.read_csv(ruta+"/01/BSS2_M01_T22_ga1_IMU1.csv",
names=["X","Y","Z"])
diccionario["N1ga1IMU2"] =
pd.read_csv(ruta+"/01/BSS2_M01_T22_ga1_IMU2.csv",
names=["X","Y","Z"])
```

```

diccionario ["N1ga3IMU1"] =
pd.read_csv(ruta+"/01/BSS2_M01_T22_ga3_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N1ga3IMU2"] =
pd.read_csv(ruta+"/01/BSS2_M01_T22_ga3_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N1ma1IMU1"] =
pd.read_csv(ruta+"/01/BSS2_M01_T22_ma1_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N1ma1IMU2"] =
pd.read_csv(ruta+"/01/BSS2_M01_T22_ma1_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N1ma3IMU1"] =
pd.read_csv(ruta+"/01/BSS2_M01_T22_ma3_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N1ma3IMU2"] =
pd.read_csv(ruta+"/01/BSS2_M01_T22_ma3_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N2ca1IMU1"] =
pd.read_csv(ruta+"/02/BSS2_V02_T24_ca1_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N2ca1IMU2"] =
pd.read_csv(ruta+"/02/BSS2_V02_T24_ca1_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N2ca1IMU3"] =
pd.read_csv(ruta+"/02/BSS2_V02_T24_ca1_IMU3.csv",
names=["X","Y","Z"])
diccionario ["N2ma1IMU1"] =
pd.read_csv(ruta+"/02/BSS2_V02_T24_ma1_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N2ma1IMU2"] =
pd.read_csv(ruta+"/02/BSS2_V02_T24_ma1_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N2ma1IMU3"] =
pd.read_csv(ruta+"/02/BSS2_V02_T24_ma1_IMU3.csv",
names=["X","Y","Z"])
diccionario ["N2ma2IMU1"] =
pd.read_csv(ruta+"/02/BSS2_V02_T24_ma2_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N2ma2IMU2"] =
pd.read_csv(ruta+"/02/BSS2_V02_T24_ma2_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N2ma2IMU3"] =
pd.read_csv(ruta+"/02/BSS2_V02_T24_ma2_IMU3.csv",
names=["X","Y","Z"])
diccionario ["N2ma3IMU1"] =
pd.read_csv(ruta+"/02/BSS2_V02_T24_ma3_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N2ma3IMU2"] =
pd.read_csv(ruta+"/02/BSS2_V02_T24_ma3_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N2ma3IMU3"] =
pd.read_csv(ruta+"/02/BSS2_V02_T24_ma3_IMU3.csv",
names=["X","Y","Z"])
diccionario ["N3ca3IMU1"] =
pd.read_csv(ruta+"/03/BSS2_M03_T28_ca3_IMU1.csv",
names=["X","Y","Z"])

```

```

diccionario ["N3ca3IMU2"] =
pd.read_csv(ruta+"/03/BSS2_M03_T28_ca3_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N3ca3IMU3"] =
pd.read_csv(ruta+"/03/BSS2_M03_T28_ca3_IMU3.csv",
names=["X","Y","Z"])
diccionario ["N3ga1IMU1"] =
pd.read_csv(ruta+"/03/BSS2_M03_T28_ga1_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N3ga1IMU2"] =
pd.read_csv(ruta+"/03/BSS2_M03_T28_ga1_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N3ga1IMU3"] =
pd.read_csv(ruta+"/03/BSS2_M03_T28_ga1_IMU3.csv",
names=["X","Y","Z"])
diccionario ["N3ga2IMU1"] =
pd.read_csv(ruta+"/03/BSS2_M03_T28_ga2_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N3ga2IMU2"] =
pd.read_csv(ruta+"/03/BSS2_M03_T28_ga2_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N3ga2IMU3"] =
pd.read_csv(ruta+"/03/BSS2_M03_T28_ga2_IMU3.csv",
names=["X","Y","Z"])
diccionario ["N3ga3IMU1"] =
pd.read_csv(ruta+"/03/BSS2_M03_T28_ga3_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N3ga3IMU2"] =
pd.read_csv(ruta+"/03/BSS2_M03_T28_ga3_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N3ga3IMU3"] =
pd.read_csv(ruta+"/03/BSS2_M03_T28_ga3_IMU3.csv",
names=["X","Y","Z"])
diccionario ["N4ca1IMU1"] =
pd.read_csv(ruta+"/04/BSS2_V04_T28_ca1_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N4ca1IMU2"] =
pd.read_csv(ruta+"/04/BSS2_V04_T28_ca1_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N4ca1IMU3"] =
pd.read_csv(ruta+"/04/BSS2_V04_T28_ca1_IMU3.csv",
names=["X","Y","Z"])
diccionario ["N4ca2IMU1"] =
pd.read_csv(ruta+"/04/BSS2_V04_T28_ca2_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N4ca2IMU2"] =
pd.read_csv(ruta+"/04/BSS2_V04_T28_ca2_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N4ca2IMU3"] =
pd.read_csv(ruta+"/04/BSS2_V04_T28_ca2_IMU3.csv",
names=["X","Y","Z"])
diccionario ["N4ca3IMU1"] =
pd.read_csv(ruta+"/04/BSS2_V04_T28_ca3_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N4ca3IMU2"] =
pd.read_csv(ruta+"/04/BSS2_V04_T28_ca3_IMU2.csv",
names=["X","Y","Z"])

```

```

diccionario ["N4ca3IMU3"] =
pd.read_csv(ruta+"/04/BSS2_V04_T28_ca3_IMU3.csv",
names=["X","Y","Z"])
diccionario ["N4ca4IMU1"] =
pd.read_csv(ruta+"/04/BSS2_V04_T28_ca4_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N4ca4IMU3"] =
pd.read_csv(ruta+"/04/BSS2_V04_T28_ca4_IMU3.csv",
names=["X","Y","Z"])
diccionario ["N4ga1IMU1"] =
pd.read_csv(ruta+"/04/BSS2_V04_T28_ga1_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N4ga1IMU2"] =
pd.read_csv(ruta+"/04/BSS2_V04_T28_ga1_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N4ga1IMU3"] =
pd.read_csv(ruta+"/04/BSS2_V04_T28_ga1_IMU3.csv",
names=["X","Y","Z"])
diccionario ["N4ga2IMU1"] =
pd.read_csv(ruta+"/04/BSS2_V04_T28_ga2_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N4ga2IMU2"] =
pd.read_csv(ruta+"/04/BSS2_V04_T28_ga2_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N4ga2IMU3"] =
pd.read_csv(ruta+"/04/BSS2_V04_T28_ga2_IMU3.csv",
names=["X","Y","Z"])
diccionario ["N4ga3IMU1"] =
pd.read_csv(ruta+"/04/BSS2_V04_T28_ga3_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N4ga3IMU2"] =
pd.read_csv(ruta+"/04/BSS2_V04_T28_ga3_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N4ga3IMU3"] =
pd.read_csv(ruta+"/04/BSS2_V04_T28_ga3_IMU3.csv",
names=["X","Y","Z"])
diccionario ["N4ga5IMU1"] =
pd.read_csv(ruta+"/04/BSS2_V04_T28_ga5_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N4ga5IMU2"] =
pd.read_csv(ruta+"/04/BSS2_V04_T28_ga5_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N4ga5IMU3"] =
pd.read_csv(ruta+"/04/BSS2_V04_T28_ga5_IMU3.csv",
names=["X","Y","Z"])
diccionario ["N4ma1IMU1"] =
pd.read_csv(ruta+"/04/BSS2_V04_T28_ma1_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N4ma1IMU2"] =
pd.read_csv(ruta+"/04/BSS2_V04_T28_ma1_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N4ma1IMU3"] =
pd.read_csv(ruta+"/04/BSS2_V04_T28_ma1_IMU3.csv",
names=["X","Y","Z"])
diccionario ["N4ma2IMU1"] =
pd.read_csv(ruta+"/04/BSS2_V04_T28_ma2_IMU1.csv",
names=["X","Y","Z"])

```

```

diccionario ["N4ma2IMU2"] =
pd.read_csv(ruta+"/04/BSS2_V04_T28_ma2_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N4ma2IMU3"] =
pd.read_csv(ruta+"/04/BSS2_V04_T28_ma2_IMU3.csv",
names=["X","Y","Z"])
diccionario ["N4ma3IMU1"] =
pd.read_csv(ruta+"/04/BSS2_V04_T28_ma3_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N4ma3IMU2"] =
pd.read_csv(ruta+"/04/BSS2_V04_T28_ma3_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N4ma3IMU3"] =
pd.read_csv(ruta+"/04/BSS2_V04_T28_ma3_IMU3.csv",
names=["X","Y","Z"])
diccionario ["N4ma4IMU1"] =
pd.read_csv(ruta+"/04/BSS2_V04_T28_ma4_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N4ma4IMU2"] =
pd.read_csv(ruta+"/04/BSS2_V04_T28_ma4_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N4ma4IMU3"] =
pd.read_csv(ruta+"/04/BSS2_V04_T28_ma4_IMU3.csv",
names=["X","Y","Z"])
diccionario ["N5ca1IMU1"] =
pd.read_csv(ruta+"/05/BSS2_M05_T24_ca1_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N5ca1IMU2"] =
pd.read_csv(ruta+"/05/BSS2_M05_T24_ca1_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N5ca1IMU3"] =
pd.read_csv(ruta+"/05/BSS2_M05_T24_ca1_IMU3.csv",
names=["X","Y","Z"])
diccionario ["N5ca2IMU1"] =
pd.read_csv(ruta+"/05/BSS2_M05_T24_ca2_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N5ca2IMU2"] =
pd.read_csv(ruta+"/05/BSS2_M05_T24_ca2_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N5ca2IMU3"] =
pd.read_csv(ruta+"/05/BSS2_M05_T24_ca2_IMU3.csv",
names=["X","Y","Z"])
diccionario ["N5ca3IMU1"] =
pd.read_csv(ruta+"/05/BSS2_M05_T24_ca3_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N5ca3IMU2"] =
pd.read_csv(ruta+"/05/BSS2_M05_T24_ca3_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N5ca3IMU3"] =
pd.read_csv(ruta+"/05/BSS2_M05_T24_ca3_IMU3.csv",
names=["X","Y","Z"])
diccionario ["N5ca4IMU1"] =
pd.read_csv(ruta+"/05/BSS2_M05_T24_ca4_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N5ca4IMU2"] =
pd.read_csv(ruta+"/05/BSS2_M05_T24_ca4_IMU2.csv",
names=["X","Y","Z"])

```

```

diccionario ["N5ca4IMU3"] =
pd.read_csv(ruta+"/05/BSS2_M05_T24_ca4_IMU3.csv",
names=["X","Y","Z"])
diccionario ["N5ca5IMU1"] =
pd.read_csv(ruta+"/05/BSS2_M05_T24_ca5_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N5ca5IMU2"] =
pd.read_csv(ruta+"/05/BSS2_M05_T24_ca5_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N5ca5IMU3"] =
pd.read_csv(ruta+"/05/BSS2_M05_T24_ca5_IMU3.csv",
names=["X","Y","Z"])
diccionario ["N5ca6IMU1"] =
pd.read_csv(ruta+"/05/BSS2_M05_T24_ca6_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N5ca6IMU2"] =
pd.read_csv(ruta+"/05/BSS2_M05_T24_ca6_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N5ca6IMU3"] =
pd.read_csv(ruta+"/05/BSS2_M05_T24_ca6_IMU3.csv",
names=["X","Y","Z"])
diccionario ["N5ga1IMU1"] =
pd.read_csv(ruta+"/05/BSS2_M05_T24_ga1_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N5ga1IMU2"] =
pd.read_csv(ruta+"/05/BSS2_M05_T24_ga1_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N5ga1IMU3"] =
pd.read_csv(ruta+"/05/BSS2_M05_T24_ga1_IMU3.csv",
names=["X","Y","Z"])
diccionario ["N5ga2IMU1"] =
pd.read_csv(ruta+"/05/BSS2_M05_T24_ga2_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N5ga2IMU2"] =
pd.read_csv(ruta+"/05/BSS2_M05_T24_ga2_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N5ga2IMU3"] =
pd.read_csv(ruta+"/05/BSS2_M05_T24_ga2_IMU3.csv",
names=["X","Y","Z"])
diccionario ["N5ga3IMU1"] =
pd.read_csv(ruta+"/05/BSS2_M05_T24_ga3_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N5ga3IMU2"] =
pd.read_csv(ruta+"/05/BSS2_M05_T24_ga3_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N5ga3IMU3"] =
pd.read_csv(ruta+"/05/BSS2_M05_T24_ga3_IMU3.csv",
names=["X","Y","Z"])
diccionario ["N5ga4IMU1"] =
pd.read_csv(ruta+"/05/BSS2_M05_T24_ga4_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N5ga4IMU2"] =
pd.read_csv(ruta+"/05/BSS2_M05_T24_ga4_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N5ga4IMU3"] =
pd.read_csv(ruta+"/05/BSS2_M05_T24_ga4_IMU3.csv",
names=["X","Y","Z"])

```

```

diccionario ["N5ga5IMU1"] =
pd.read_csv(ruta+"/05/BSS2_M05_T24_ga5_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N5ga5IMU2"] =
pd.read_csv(ruta+"/05/BSS2_M05_T24_ga5_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N5ga5IMU3"] =
pd.read_csv(ruta+"/05/BSS2_M05_T24_ga5_IMU3.csv",
names=["X","Y","Z"])
diccionario ["N5ma1IMU1"] =
pd.read_csv(ruta+"/05/BSS2_M05_T24_ma1_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N5ma1IMU2"] =
pd.read_csv(ruta+"/05/BSS2_M05_T24_ma1_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N5ma1IMU3"] =
pd.read_csv(ruta+"/05/BSS2_M05_T24_ma1_IMU3.csv",
names=["X","Y","Z"])
diccionario ["N5ma2IMU1"] =
pd.read_csv(ruta+"/05/BSS2_M05_T24_ma2_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N5ma2IMU2"] =
pd.read_csv(ruta+"/05/BSS2_M05_T24_ma2_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N5ma2IMU3"] =
pd.read_csv(ruta+"/05/BSS2_M05_T24_ma2_IMU3.csv",
names=["X","Y","Z"])
diccionario ["N5ma3IMU1"] =
pd.read_csv(ruta+"/05/BSS2_M05_T24_ma3_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N5ma3IMU2"] =
pd.read_csv(ruta+"/05/BSS2_M05_T24_ma3_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N5ma3IMU3"] =
pd.read_csv(ruta+"/05/BSS2_M05_T24_ma3_IMU3.csv",
names=["X","Y","Z"])
diccionario ["N5ma4IMU1"] =
pd.read_csv(ruta+"/05/BSS2_M05_T24_ma4_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N5ma4IMU2"] =
pd.read_csv(ruta+"/05/BSS2_M05_T24_ma4_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N5ma4IMU3"] =
pd.read_csv(ruta+"/05/BSS2_M05_T24_ma4_IMU3.csv",
names=["X","Y","Z"])
diccionario ["N5ma5IMU1"] =
pd.read_csv(ruta+"/05/BSS2_M05_T24_ma5_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N5ma5IMU2"] =
pd.read_csv(ruta+"/05/BSS2_M05_T24_ma5_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N5ma5IMU3"] =
pd.read_csv(ruta+"/05/BSS2_M05_T24_ma5_IMU3.csv",
names=["X","Y","Z"])
diccionario ["N6ca1IMU1"] =
pd.read_csv(ruta+"/06/BSS2_V06_T22_ca1_IMU1.csv",
names=["X","Y","Z"])

```

```

diccionario ["N6ca1IMU2"] =
pd.read_csv(ruta+"/06/BSS2_V06_T22_ca1_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N6ca2IMU1"] =
pd.read_csv(ruta+"/06/BSS2_V06_T22_ca2_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N6ca2IMU2"] =
pd.read_csv(ruta+"/06/BSS2_V06_T22_ca2_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N6ca3IMU1"] =
pd.read_csv(ruta+"/06/BSS2_V06_T22_ca3_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N6ca3IMU2"] =
pd.read_csv(ruta+"/06/BSS2_V06_T22_ca3_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N6ca4IMU1"] =
pd.read_csv(ruta+"/06/BSS2_V06_T22_ca4_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N6ca4IMU2"] =
pd.read_csv(ruta+"/06/BSS2_V06_T22_ca4_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N6ga1IMU1"] =
pd.read_csv(ruta+"/06/BSS2_V06_T22_ga1_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N6ga1IMU2"] =
pd.read_csv(ruta+"/06/BSS2_V06_T22_ga1_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N6ga2IMU1"] =
pd.read_csv(ruta+"/06/BSS2_V06_T22_ga2_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N6ga2IMU2"] =
pd.read_csv(ruta+"/06/BSS2_V06_T22_ga2_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N6ga3IMU1"] =
pd.read_csv(ruta+"/06/BSS2_V06_T22_ga3_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N6ga3IMU2"] =
pd.read_csv(ruta+"/06/BSS2_V06_T22_ga3_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N6ga4IMU1"] =
pd.read_csv(ruta+"/06/BSS2_V06_T22_ga4_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N6ga4IMU2"] =
pd.read_csv(ruta+"/06/BSS2_V06_T22_ga4_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N6ma1IMU1"] =
pd.read_csv(ruta+"/06/BSS2_V06_T22_ma1_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N6ma1IMU2"] =
pd.read_csv(ruta+"/06/BSS2_V06_T22_ma1_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N6ma2IMU1"] =
pd.read_csv(ruta+"/06/BSS2_V06_T22_ma2_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N6ma2IMU2"] =
pd.read_csv(ruta+"/06/BSS2_V06_T22_ma2_IMU2.csv",
names=["X","Y","Z"])

```

```

diccionario ["N6ma3IMU1"] =
pd.read_csv(ruta+"/06/BSS2_V06_T22_ma3_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N6ma3IMU2"] =
pd.read_csv(ruta+"/06/BSS2_V06_T22_ma3_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N6ma4IMU1"] =
pd.read_csv(ruta+"/06/BSS2_V06_T22_ma4_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N6ma4IMU2"] =
pd.read_csv(ruta+"/06/BSS2_V06_T22_ma4_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N6ma5IMU1"] =
pd.read_csv(ruta+"/06/BSS2_V06_T22_ma5_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N6ma5IMU2"] =
pd.read_csv(ruta+"/06/BSS2_V06_T22_ma5_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N7ca1IMU1"] =
pd.read_csv(ruta+"/07/BSS2_V07_T26_ca1_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N7ca1IMU2"] =
pd.read_csv(ruta+"/07/BSS2_V07_T26_ca1_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N7ca1IMU3"] =
pd.read_csv(ruta+"/07/BSS2_V07_T26_ca1_IMU3.csv",
names=["X","Y","Z"])
diccionario ["N7ca2IMU1"] =
pd.read_csv(ruta+"/07/BSS2_V07_T26_ca2_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N7ca2IMU2"] =
pd.read_csv(ruta+"/07/BSS2_V07_T26_ca2_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N7ca2IMU3"] =
pd.read_csv(ruta+"/07/BSS2_V07_T26_ca2_IMU3.csv",
names=["X","Y","Z"])
diccionario ["N7ca3IMU1"] =
pd.read_csv(ruta+"/07/BSS2_V07_T26_ca3_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N7ca3IMU2"] =
pd.read_csv(ruta+"/07/BSS2_V07_T26_ca3_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N7ca3IMU3"] =
pd.read_csv(ruta+"/07/BSS2_V07_T26_ca3_IMU3.csv",
names=["X","Y","Z"])
diccionario ["N7ca4IMU1"] =
pd.read_csv(ruta+"/07/BSS2_V07_T26_ca4_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N7ca4IMU2"] =
pd.read_csv(ruta+"/07/BSS2_V07_T26_ca4_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N7ca4IMU3"] =
pd.read_csv(ruta+"/07/BSS2_V07_T26_ca4_IMU3.csv",
names=["X","Y","Z"])
diccionario ["N7ga1IMU1"] =
pd.read_csv(ruta+"/07/BSS2_V07_T26_ga1_IMU1.csv",
names=["X","Y","Z"])

```

```

diccionario ["N7ga1IMU2"] =
pd.read_csv(ruta+"/07/BSS2_V07_T26_ga1_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N7ga1IMU3"] =
pd.read_csv(ruta+"/07/BSS2_V07_T26_ga1_IMU3.csv",
names=["X","Y","Z"])
diccionario ["N7ga2IMU1"] =
pd.read_csv(ruta+"/07/BSS2_V07_T26_ga2_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N7ga2IMU2"] =
pd.read_csv(ruta+"/07/BSS2_V07_T26_ga2_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N7ga2IMU3"] =
pd.read_csv(ruta+"/07/BSS2_V07_T26_ga2_IMU3.csv",
names=["X","Y","Z"])
diccionario ["N7ga3IMU1"] =
pd.read_csv(ruta+"/07/BSS2_V07_T26_ga3_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N7ga3IMU2"] =
pd.read_csv(ruta+"/07/BSS2_V07_T26_ga3_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N7ga3IMU3"] =
pd.read_csv(ruta+"/07/BSS2_V07_T26_ga3_IMU3.csv",
names=["X","Y","Z"])
diccionario ["N7ga4IMU1"] =
pd.read_csv(ruta+"/07/BSS2_V07_T26_ga4_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N7ga4IMU2"] =
pd.read_csv(ruta+"/07/BSS2_V07_T26_ga4_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N7ga4IMU3"] =
pd.read_csv(ruta+"/07/BSS2_V07_T26_ga4_IMU3.csv",
names=["X","Y","Z"])
diccionario ["N7ma1IMU1"] =
pd.read_csv(ruta+"/07/BSS2_V07_T26_ma1_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N7ma1IMU2"] =
pd.read_csv(ruta+"/07/BSS2_V07_T26_ma1_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N7ma1IMU3"] =
pd.read_csv(ruta+"/07/BSS2_V07_T26_ma1_IMU3.csv",
names=["X","Y","Z"])
diccionario ["N7ma2IMU1"] =
pd.read_csv(ruta+"/07/BSS2_V07_T26_ma2_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N7ma2IMU2"] =
pd.read_csv(ruta+"/07/BSS2_V07_T26_ma2_IMU2.csv",
names=["X","Y","Z"])
diccionario ["N7ma2IMU3"] =
pd.read_csv(ruta+"/07/BSS2_V07_T26_ma2_IMU3.csv",
names=["X","Y","Z"])
diccionario ["N7ma3IMU1"] =
pd.read_csv(ruta+"/07/BSS2_V07_T26_ma3_IMU1.csv",
names=["X","Y","Z"])
diccionario ["N7ma3IMU2"] =
pd.read_csv(ruta+"/07/BSS2_V07_T26_ma3_IMU2.csv",
names=["X","Y","Z"])

```

```

    diccionario          ["N7ma3IMU3"]          =
pd.read_csv(ruta+"/07/BSS2_V07_T26_ma3_IMU3.csv",
names=["X","Y","Z"])
    diccionario          ["N7ma4IMU1"]          =
pd.read_csv(ruta+"/07/BSS2_V07_T26_ma4_IMU1.csv",
names=["X","Y","Z"])
    diccionario          ["N7ma4IMU2"]          =
pd.read_csv(ruta+"/07/BSS2_V07_T26_ma4_IMU2.csv",
names=["X","Y","Z"])
    diccionario          ["N7ma4IMU3"]          =
pd.read_csv(ruta+"/07/BSS2_V07_T26_ma4_IMU3.csv",
names=["X","Y","Z"])

```

- ANÁLISIS DE DATOS

```

#!/usr/bin/env python2
# -*- coding: utf-8 -*-
"""
Created on Sat May 12 08:10:13 2018

@author: martagarciabor
"""

#IMPORTAR LAS LIBRERÍAS NECESARIAS PARA PODER EJECUTAR EL PROGRAMA
import pandas as pd
import matplotlib.pyplot as plt
import numpy as np
import DATOSXYZ
from DATOSXYZ import diccionario, ruta
from matplotlib.backends.backend_pdf import PdfPages

#%% OBTENCIÓN DEL VALOR EFICAZ SEVEAL

#DEFINIR LA FUNCIÓN VALOREFICAZ

# INPUT: N --> MUESTRAS VENTANAS
def valoreficaz(N):

#CREACIÓN LISTA CON LAS COLUMNAS DE INTERÉS
    columns = ["X", "Y", "Z", "Dato","Usuario","IMU", "Actividad"]

#A PARTIR DE ELLAS, OBTENER UN DATAFRAME
    vefcomps = pd.DataFrame(columns=columns)

#MARCAR QUE SE GUARDE UN PDF Y DONDE
    with PdfPages(ruta+'/multipage_pdfDVE.pdf') as pdf:

#B1. PARA CADA DATO DEL DICcionario:

# A) IMPRIMIR CADA DATO (K) Y EXTRAER EL USUARIO, ACTIVIDAD E IMU
    for k in diccionario:
        print(k)
        usuario = k[1]
        actividad = k[2:4]
        IMU = k[8]

```

```

# B) ALMACENAR EN UNA VARIABLE TODOS LOS DATOS
    valores = diccionario[k]

# C) OBTENER LA LONGITUD (CANTIDAD DE VALORES) DE CADA DATO
    lon= len(valores)

#B2.1. PARA CADA K, PARA CADA VENTANA (I):
# VENTANA (DEFINIDA DESDE I HASTA LON-N)
# RESTAR N PARA NO OBTENER DATOS INNECESARIOS
    for i in range(lon-N):

# D) CREAR UN DICCIONARIO AL QUE LUEGO AÑADIRLE LOS DATOS (TEMP)
    temp = {}

#B3. PARA CADA K Y CADA I, PARA CADA COMPONENTE X,Y,Z (ACELERACIONES):
    for comp in ["X","Y","Z"]:

# E) OBTENER EL VALOR EFICAZ
# VALOR EFICAZ = MEDIA RAÍZ DEL CUADRADO DE LOS DATOS.
# F) ALMACENAR EL RESULTADO, EL DATO, USUARIO, IMU Y ACTIVIDAD EN TEMP
        temp[comp] =
np.mean(np.sqrt(valores[comp][i:i+N]**2))
        temp["Dato"]=k #nombre del dato completo
        temp["Usuario"]= usuario #usuario(arriba definido)
        temp["IMU"] = IMU #IMU(arriba definido)
        temp["Actividad"] = actividad #actividad(arriba
definido)

# G) ADICIÓN DE TODOS LOS VALORES EN EL DATAFRAME CREADO
    vefcomps = vefcomps.append(temp, ignore_index=True)

# H) CREACIÓN DE UN FILTRO QUE PERMITA OBTENER LAS GRÁFICAS
    filtro = vefcomps["Dato"]==k

#B2.2. PARA CADA K, Y CADA COMPONENTE DE LA ACELERACIÓN
    for comp in ["X","Y","Z"]:

# I) DIBUJAR LAS GRÁFICAS Y ETIQUETARLAS
        plt.plot(vefcomps[comp][filtro])
        plt.xlabel("MUESTRAS")
        plt.ylabel("ACELERACIÓN")
        plt.title("ACC(M)" + k)
        plt.legend(['ACCX', 'ACCY', 'ACCZ'])

# J) GUARDAR PDF CON TODAS LAS FIGURAS
        pdf.savefig()

# K) NO IMPRIMIR POR PANTALLA CADA DATO
        plt.close()

# L) OBTENER EL DATAFRAME CREADO
    return vefcomps

# M) DEFINIR EL DATAFRAME CON LA FUNCIÓN
vefcomps = valoreficaz(100)

%% GUARDAR DATOS

```

```
#IMPORTAMOS LA FUNCIÓN QUE PERMITE ESCRIBIR EN UN EXCEL
from pandas import ExcelWriter

#CREAMOS UN EXCEL QUE ALMACENE LA INFORMACIÓN OBTENIDA
writer = ExcelWriter('dveficazpu.xlsx')
vefcomps.to_excel(writer, 'DataFrame')

#GUARDAMOS EL EXCEL
writer.save()

### CARGAR DATOS

#ABRIMOS EL EXCEL CREADO ANTERIORMENTE PARA NO TENER QUE ESTAR
#EJECUTÁNDOLO CADA VEZ
vefcomps = pd.read_excel('dveficazpu.xlsx')

### OBTENCIÓN DE ANÁLISIS ESTADÍSTICO

#AGRUPAMOS LA INFORMACIÓN EN FUNCIÓN DE LA ACTIVIDAD, USUARIO E
#IMU
dagrup = vefcomps.groupby(['Actividad', 'Usuario', 'IMU'])

#PARA ESA AGRUPACIÓN, OBTENEMOS LA MEDIA Y LA DESVIACIÓN TÍPICA
d = dagrup[['X', 'Y', 'Z']].agg([np.mean, np.std])

#RENOMBAMOS LAS COLUMNAS A NUESTRO GUSTO
d = d.rename(columns={'std' : 'dvef', 'mean' : 'mvef'})

### CREACIÓN DE UN DATAFRAME CON LOS DATOS DE INTERÉS PARA PODER
#HACER EL ANÁLISIS

#MEDIAS Y DESVIACIONES TÍPICAS
MX = d['X']['mvef'].values
MY = d['Y']['mvef'].values
MZ = d['Z']['mvef'].values
DX = d['X']['dvef'].values
DY = d['Y']['dvef'].values
DZ = d['Z']['dvef'].values

#VALOR DE LA ACTIVIDAD E IMU Y DIVISIÓN EN USUARIOS
a = np.array(d.index.get_level_values('Actividad'))
i = np.array(d.index.get_level_values('IMU'), dtype= object)
u = np.array(d.index.get_level_values('Usuario'), dtype= object)

#CREACIÓN DE UN DICCIONARIO CON LOS DATOS OBTENIDOS
ddatos = {}
ddatos["activ"] = a
ddatos["IMU"] = i
ddatos["MX"] = MX
ddatos["MY"] = MY
ddatos["MZ"] = MZ
ddatos["DX"] = DX
ddatos["DY"] = DY
ddatos["DZ"] = DZ
ddatos["Usuario"] = u

#PASAR DE DICCIONARIO A DATAFRAME PARA PODER EMPLEARLO EN LDA
```

```
datos = pd.DataFrame(ddatos)

#%% ANÁLISIS DISCRIMINANTE LINEAL

#IMPORTAMOS LA FUNCIÓN QUE PERMITA LLEVAR A CABO LA CLASIFICACIÓN
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis

#DIVIDIMOS LOS DATOS EN DOS LISTAS

#X PARA LAS VARIABLES QUE DETERMINARÁN EL TIPO DE ACTIVIDAD
X = datos[["MX", "MY", "MZ", "DX", "DY", "DZ","IMU"]].values

#Y CON EL VALOR(EN ESTE CASO, CATEGÓRICO NO NUMÉRICO) DE LAS
ACTIVIDADES POSIBLE
y = datos["activ"].values

#LLAMAMOS A LA FUNCIÓN
clf = LinearDiscriminantAnalysis()

#AJUSTAMOS EL MODELO
clf.fit(X, y)

print(clf.coef_)
#CREACIÓN DE LISTAS PARA GUARDAR LOS DATOS PREDICHOS Y REALES
Pred = []
Real = []

#CREACIÓN DE UN BUCLE QUE SE REPITA PARA LAS 49 ACTIVIDADES
REGISTRADAS
for ind in range(49):

#PREDICCIÓN DEL RESULTADO
    sal = clf.predict(X)

#IMPRESIÓN DEL VALOR PREDICHO Y EL REAL
    Valor = "predicho:", sal[ind],"real:", datos.iloc[ind]["activ"]
    print(Valor)

#INSERCIÓN DE CADA VALOR EN EL ARRAY CREADO AL INICIO DEL ANÁLISIS
    Pred.append(Valor[1])
    Real.append(Valor[3])

#OBTENCIÓN DEL VALOR
#print(Pred)
#print(Real)

#%% VALIDACIÓN A

#IMPORTAR FUNCIÓN PARA REALIZAR MATRICES DE EVALUACIÓN
from sklearn.metrics import confusion_matrix

#CREACIÓN DE LA MATRIZ DE CONFUSIÓN
CM = confusion_matrix(Real, Pred, labels=["ca","ga","ma"])
print(CM)
```

```

### VALIDACIÓN B

#OBTENEMOS UN ARRAY CON LOS DATOS DE USUARIO PARA PODER IR ELIMINANDO
CADA VEZ
#UN USUARIO EN EL MÉTODO DE VALIDACIÓN
P = np.array(datos["Usuario"].values)

#IMPORTAR FUNCIONES PARA OBTENER LA EXACTITUD Y PODER VALIDAR
from sklearn import metrics
from sklearn.model_selection import LeaveOneOut

#LLAMAMOS A LA FUNCIÓN, Y LE PASAMOS LOS PARÁMETROS PARA QUE DIVIDA
EN GRUPOS
loo = LeaveOneOut()
loo.get_n_splits(X,y, groups=P)

#LLAMAR AL LDA
clf = LinearDiscriminantAnalysis()

#CREAR LISTAS CON LAS VARIABLES QUE SE VAN A EMPLEAR EN LA VALIDACIÓN
total_y_pred = []
total_y_test = []
total_x_test = []

#BUCLE PARA DIVIDIR LOS DATOS DE ENTRENAMIENTO Y TESTEO
for train_index, test_index in loo.split(X,y, groups=P):

    X_train, X_test = X[train_index], X[test_index]
    y_train, y_test = y[train_index], y[test_index]

#AJUSTE DEL MODELO PARA CADA CONJUNTO DE ENTRENAMIENTO
    clf.fit(X_train, y_train)

#PREDICCIÓN DE LOS VALORES E INSERCIÓN DE LOS MISMOS EN LAS LISTAS
    total_y_pred.append(clf.predict(X_test)[0])
    total_y_test.append(y_test[0])
    total_x_test.append(list(X_test[0]))

print(clf.coef_)
print(clf.priors_)
#print(clf.intercept_)
#print(clf.means_)
### OBTENCIÓN DE MATRICES
from sklearn.metrics import confusion_matrix

#CREACIÓN DE ARRAYS
total_y_pred = np.array(total_y_pred)
total_y_test = np.array(total_y_test)
total_x_test = np.array(total_x_test)

#OBTENCIÓN DE LA EXACTITUD
#CTe = clf.score(X, y)
#CTr = clf.score(total_x_test, total_y_test)
CTl = clf.score(X_train, y_train)
print (CTl)
AS = metrics.accuracy_score(total_y_test, total_y_pred)
print (AS)

```

```
CM2 = confusion_matrix(total_y_test, total_y_pred,
labels=["ca", "ga", "ma"])
print(CM2)
```

CUANTIFICACIÓN DE LA ACTIVIDAD

```
#!/usr/bin/env python2
# -*- coding: utf-8 -*-
"""
Created on Mon May 7 15:19:56 2018

@author: martagarciaibor
"""

#%%

import pandas as pd
import numpy as np
from clasifica_puntos import clasifica_puntos
from DATOSXYZ import diccionario, ruta
import matplotlib.pyplot as plt
from matplotlib.backends.backend_pdf import PdfPages

maxmin0 = pd.DataFrame (columns = ["MAX", "MIN", "PASOCERO"])

with PdfPages(ruta+'/deteccionpicos2.pdf') as pdf:
    for k in diccionario:
        valores = diccionario[k]
        data = {}
        picoscom = {"accX":np.array(valores["X"]),
"accY":np.array(valores["Y"]), "accZ":np.array(valores["Z"])}
        for t in picoscom:
            detecpic = picoscom[t]
            M,m,z = clasifica_puntos(detecpic, 0.2, draw = False)
            # plt.xlabel("MUESTRAS")
            # plt.ylabel(U"ACELERACION")
            # plt.title("ACC(T)" + k)
            # pdf.savefig()
            # plt.close()
            data["MAX"] = M
            data["MIN"] = m
            data["PASOCERO"] = z
        maxmin0 = maxmin0.append(data, ignore_index=True)

#%%
from sympy import integrate

MAXIMO = np.array(maxmin0["MAX"])
MIN = np.array(maxmin0["MIN"])

#def resta (a):
```

```
# for i in range(len(a)-1):  
#     r = r.append(a[i +1]- a[i])  
#     return r  
#r = resta(maxmin0["MAX"])
```