SISTEMA MÓVIL PARA LA CLASIFICACIÓN AUTOMÁTICA DE ACTIVIDADES DE LA VIDA DIARIA EMPLEANDO ALGORITMOS DE STREAM LEARNING

Tesis de Trabajo de Grado
Modalidad: Trabajo de Investigación

Jaime Alfonso Pabón Rivas
100616020784
Daniel Gómez Méndez
100616021407

Director: PhD. Diego Mauricio López Gutiérrez

Co-Director. PhD. Jesus David Cerón Bravo

Universidad del Cauca
Facultad de Ingeniería Electrónica y Telecomunicaciones
Departamento de Telemática
Línea de Investigación en e-Salud
Popayán, octubre de 2023

Con la voz quebrada y los dedos temblorosos de esta alegría llena de nostalgia quiero agradecer a mis padres por todo lo que me otorgaron. Ahora de buena fe y sudor entiendo los sacrificios que me hacen estar eternamente agradecido y orgulloso de ellos. A mi tío Alejandro Moncayo por demostrarme cada día la importancia del ingenio y el amor hacia la tecnología. Así mismo agradecer a mis hermanos David Otavo y Samantha Gómez, que fueron la luz y motivación en todos los esfuerzos. A la vida y la poesía por acompañarme en cada momento. Quiero agradecer con demasiado orgullo y amor, a mis amigos de la Universidad Del Cauca, que hicieron este camino lleno de trasnochos, risas, lágrimas, fiestas, aventuras y demás; tan ameno donde ahora son una familia para mí: Cristhian, Realpe, Paula Andrea, Isa, Elmer, Juan Diego, Alfonso, Michael, Vargas, Facundo, Betancourt, Zu, Vero y Andres. Mis momentos más felices de la vida son con ustedes. Agradecer a Jaime, que solo él y yo sabremos todo el esfuerzo que demandó esta tesis donde siempre tuvo buena fe en el proyecto y mi persona. Agradecer a Andrea que fue un pilar en los momentos más difíciles, que le amo y espero que la poesía siempre le acompañe. Por último, esta primera etapa de culminación académica y todo lo positivo que he conseguido en esta vida, se lo dedico y agradezco con todo el amor a mi abuela Lucrecia Cobo, que me acogió en mi niñez y en los momentos más difíciles de mi vida. Abuela esto esto es para usted.

- ***Daniel Gómez Méndez***

Primero quiero agradecerle a Dios por haberme acompañado a lo largo de mi carrera universitaria y permitirme estar aquí en este momento tan importante de mi vida. A mi madre Eloisa quien ha sido mi motor, ejemplo a seguir y motivación para seguir adelante. A mi padre Gerardo por sus valiosos consejos, enseñanzas y ejemplo. De manera general, esto no hubiese sido posible sin el apoyo incondicional de mis padres durante toda mi vida y en especial en este proceso universitario.
Adicionalmente, quiero agradecer a mis hermanos Juan Camilo y Sara que durante diferentes etapas de la carrera estuvieron a mi lado brindándome su amistad y ayuda en momentos difíciles y celebrando los buenos momentos.
No podría olvidarme de mis tías Liliana y Maritza, mi madrina Marina, mis primas Angela e Isabela, mi tío Fredy, mi abuela Ofelia y demás familiares que siempre estuvieron apoyándome y alentandome a seguir adelante con sus esperanzas en mí, sus palabras de apoyo y cariño.
Finalmente, a mis amigos de la Universidad con los cuales vivimos este proceso de la mano tanto en la parte académica como más allá de ella aprendiendo el uno del otro y creciendo en este camino profesional donde quiero resaltar a Juan, Anyi, Chepe, Carvajal, Vero, Andrés y Paula, pero principalmente a mi compañero de Tesis Daniel que a lo largo de este proceso se convirtió en un hermano para mí.

- **Jaime Alfonso Pabón Rivas.**

CONTENIDO

LISTA DE TABLAS

LISTA DE IMÁGENES

Capítulo 1: Introducción

1.1 Planteamiento del problema

Las actividades de la vida diaria ("*Activities of daily living*" ADL) se definen como tareas esenciales y rutinarias que se realizan usualmente en el hogar, tales como: caminar, estar sentado, estar de pie sin ayuda de terceros, entre otros. [1]. Con ello en mente, el uso del concepto de ADL ha ido incrementando en los últimos años en el campo de la salud, principalmente gracias al surgimiento de nuevas tecnologías; de hecho, los profesionales de la salud suelen determinar si un adulto mayor puede vivir de manera independiente por medio de una evaluación de la ejecución de ADL [2]. El concepto de ADL se ha utilizado también en el monitoreo de comportamientos sedentarios [3], seguimiento de terapias de rehabilitación [4], identificación del riesgo de delirio [5], y el monitoreo de enfermedad de obstrucción pulmonar crónica [6,7].

Para evaluar si una persona puede vivir de manera autónoma, los trabajadores de la salud utilizan cuestionarios como el índice de Barthel [8]. Este índice consiste en un conjunto de preguntas acerca de las actividades que el adulto mayor puede realizar en su día a día. Las respuestas a esas preguntas se hacen usualmente de manera subjetiva, por lo tanto, es muy vulnerable a errores de información [9]. El uso de métodos basados en las Tecnologías de la Información y las Comunicaciones (TIC) puede convertir esta evaluación subjetiva en objetiva, por medio del reconocimiento de actividad humana.

El campo de estudio del reconocimiento de la actividad humana (*"Human activities recognition*" HAR), tiene como objetivo reconocer, de manera automática la actividad que está realizando una persona. Diversos sensores y métodos han sido utilizados para HAR, tales como el uso de cámaras [10], sensores de sonda [21], radar de ondas milimétricas [11], acelerómetros [12], sensores infrarrojos [13], entre otros [14]. Debido a que algunas actividades de la vida diaria son más complejas de lo que se puede prever, la recolección de información de contexto como por ejemplo la localización de la persona (en ingles *Indoor Location*) al momento de realizar determinadas actividades es un insumo valioso para realizar la clasificación automática de una ADL [15].

De las diversas investigaciones sobre HAR, en las que se utilizaron acelerómetros, giroscopios, monitores de frecuencia cardíaca, entre otros, se han obtenido conjuntos de datos que permiten realizar y profundizar un estudio de predicción, y/o control a las personas mediante algoritmos de Machine Learning (ML) [16]. Con esto se puede profundizar en el área de HAR que ha sido tratada en la literatura como un típico problema de reconocimiento de patrones (es decir, un problema de clasificación) que intenta identificar la actividad que realiza un individuo en un momento dado. Este problema se ha resuelto habitualmente utilizando algoritmos tradicionales de aprendizaje automático fuera de línea (*Offline Learning*). En este orden de ideas, los modelos fuera de línea (*Offline Models*) son incapaces de adaptarse a los cambios de los datos que se generan continuamente. Es necesario recordar que los datos utilizados para reconocer las actividades humanas cambian constantemente con el tiempo. Esto debido a los cambios en el comportamiento humano, las diferencias en las características físicas propias de cada persona y el estado de salud de los individuos, así como a los cambios en los entornos físicos donde la persona habita [23]. Por otro lado, recientemente se ha

encontrado altamente beneficioso el realizar la clasificación directamente en Smartphones o Wearables debido a que brinda una mayor seguridad hacia los datos recolectados, dado que evita el uso de un servidor externo, que puede llegar a ser vulnerable a ciberataques, más aún cuando el tratamiento de la seguridad de datos de salud es crítico [17]. Sin embargo, hay que tener en cuenta que cuando el reconocimiento de la actividad se realiza a través de Smartphones o wearables, el problema mencionado anteriormente de los modelos offline es aún más crítico; ya que, los datos de los sensores deben ser constantemente recogidos y analizados por modelos de clasificación, y los Smartphones o Wearables tienen limitaciones de memoria y procesamiento [23]. Por otro lado, los modelos en línea (*Online Models*) construidos mediante algoritmos de flujos de datos (en inglés *Stream Learning*), también conocidos como algoritmos de Online Learning, presentan más flexibilidad para datos que evolucionan con el tiempo [22]. De este modo, los Online Models aparecen como una solución para sistemas con bajos recursos de memoria. De este modo, el modelo incorpora nueva información para la segmentación de los datos de las actividades, que evolucionan a gran velocidad, detectando los cambios y adaptando los modelos de actividades a la información más reciente. Además, requieren un menor costo computacional [18]. Según Massive Online Analysis (MOA), un reconocido framework para Data Stream Mining, Stream Learning ha surgido recientemente como una respuesta al desafío de los datos continuos. Estos algoritmos permiten trabajar con conjuntos de datos significativamente más grandes que la capacidad de la memoria disponible en un sistema informático. Además, son capaces de extenderse a aplicaciones en tiempo real que no pueden ser abordadas por enfoques convencionales de Machine Learning o Data Mining [19].

Adicionalmente, se tiene que el HAR basado en Smartphones (SHAR) implica el uso de diferentes sensores móviles integrados en los teléfonos móviles modernos. También implica técnicas de ML para recoger e inferir automáticamente actividades de los usuarios en diferentes ámbitos como el de la salud, la asistencia a personas mayores, el deporte y el bienestar. La selección de un buen algoritmo clasificador tiene un gran impacto en el rendimiento de las aplicaciones SHAR en lo que respecta a los criterios de coste clave como lo son: el consumo energético, procesamiento CPU, el tiempo y la eficiencia temporal. Los objetivos de rendimiento a nivel de sistema incluyen la mejora de la duración de la batería en los dispositivos móviles. Mientras que los objetivos de rendimiento a nivel de aplicación incluyen la mejora de estructuras internas de los modelos de clasificación y su comportamiento de procesamiento en términos de detección de un desequilibrio de clases a partir de flujos de datos inciertos, personalización del modelo y optimización. Por lo anterior las aplicaciones SHAR optan por realizar un aprendizaje y una clasificación online en Stream Learning. [20]

Con base en el estado del arte del anteproyecto del presente trabajo de grado que se puede encontrar en el Anexo K, se evidenció la falta de estudios enfocados en el área de la salud que hagan uso de algoritmos de Stream Learning, específicamente en el área de ADL haciendo uso de HAR, donde se encontraron solamente tres estudios. En cuanto a una comparación entre algoritmos de Offline y Online Learning, se encontraron dos estudios, en lo que respecta a aplicativos móviles que hagan uso de Stream Learning se encontraron únicamente dos, donde uno está enfocado en la salud, pero no en ADL y el otro está enfocado en sistemas computacionales. Existen aplicativos móviles que realizan la clasificación automática de ADL, sin embargo, usan algoritmos de Offline Learning. Por lo anterior y según el conocimiento de los autores, no existen aplicativos móviles que realicen la clasificación automática de actividades de la vida diaria haciendo uso de algoritmos de Stream Learning.

1.2 Pregunta de investigación

Teniendo en cuenta las problemáticas mencionadas anteriormente, se establece la siguiente pregunta de investigación: ¿Cómo implementar algoritmos de Stream Learning para la clasificación de ADL usando dispositivos móviles?

Para responder la pregunta se plantea la hipótesis que establece la factibilidad para desarrollar un Sistema móvil que genere la clasificación automática de actividades de la vida diaria haciendo uso de algoritmos de Stream Learning.

1.3 Motivación

El término "datos", en los últimos años, ha cobrado un alto grado de relevancia en el ámbito de la Informática y las Tecnologías de la Información (TI), designando una representación abstracta procesada por una computadora, la cual se traduce en información comprensible para el usuario. Esta información se ha convertido en el motor que impulsa y transforma diversos sectores, desde las redes sociales hasta campos más especializados como la educación, la informática, la salud y los centros de llamadas, en los que los datos recogidos se utilizan para diferentes propósitos, como mejorar la interacción humano-máquina e implementar estrategias de gestión.

El valor de los datos es especialmente notorio en el sector sanitario, como se evidenció durante la pandemia de COVID-19. Los datos existentes sobre enfermedades respiratorias, manejados de forma experta, permitieron la implementación de estrategias efectivas y el desarrollo de una vacuna, facilitando una transición hacia una nueva normalidad. Además, los datos respaldan las decisiones médicas, tales como determinar cuándo es necesaria una cirugía, cuál medicamento podría ser más efectivo o cómo un procedimiento específico puede influir en la recuperación de un paciente.

Desafortunadamente, la calidad de vida y la tecnología no siempre marchan en paralelo, en particular en el entorno sanitario. A menudo, los pacientes no proporcionan información completa o precisa durante las consultas médicas. Para enfrentar este problema, los profesionales de la salud e ingeniería han comenzado a monitorear de forma más rigurosa estos aspectos. La motivación de esta tesis radica precisamente aquí: en el análisis y procesamiento de estos datos recopilados de actividades de la vida diaria (ADL). Los estudios existentes analizados en el planteamiento del problema han utilizado algoritmos de aprendizaje fuera de línea (Offline Learning), con sus respectivas ventajas y desventajas. Por lo tanto, se propone el uso de algoritmos de aprendizaje en flujo (Stream Learning) para optimizar el procesamiento de la información, teniendo en cuenta también la investigación realizada en el anteproyecto que le precede a esta monografía.

1.4 Objetivos

1.4.1 Objetivo general

Implementar algoritmos de Stream Learning para la clasificación de actividades de la vida diaria usando dispositivos móviles.

1.4.2 Objetivos específicos

- Identificar uno o varios conjuntos de datos para la clasificación de actividades de la vida diaria (ADL) que permita(n) realizar un análisis de rendimiento de algoritmos basados en data stream learning.

- Realizar una comparación del rendimiento de algoritmos basados en Batch Learning contra algoritmos de data Stream Learning usando el o los conjuntos de datos seleccionados en el anterior objetivo.

- Desarrollar una aplicación móvil para la clasificación automática de ADL en la que se despliegue el mejor modelo de clasificación identificado en el objetivo anterior.

1.5 Metodología

Para resolver la pregunta de investigación, se planteó un proyecto de ciencia de datos. En los últimos años, han surgido diferentes enfoques que ofrecen una forma organizada de extraer patrones de datos. Algunos de estos enfoques populares son KDD, SEMMA, CRISP-DM y Catalyst. Solo se consideraron las dos últimas como metodologías de ciencia de datos, ya que describen las etapas específicas, y además proporcionan una guía práctica para llevar a cabo el trabajo [60,67,81]. En concordancia y, para el cumplimiento de los objetivos, se optó por utilizar CRISP-DM, dado que es la metodología más comúnmente aplicada en proyectos de ciencia de datos desde el 2007 [74]. En añadidura, se ha de especificar que esta metodología de código abierto beneficia en gran medida la extracción y aplicación de la misma para el presente proyecto [68].

Se identificó que CRISP-DM es una metodología que se basa en un modelo de proceso jerárquico, el cual se compone de tareas organizadas en cuatro niveles de abstracción: fase, tarea genérica, tarea especializada e instancia de proceso [60].

En el nivel más alto, se dividen las fases del proceso de ciencia de datos en seis etapas; cada fase engloba múltiples tareas genéricas de segundo nivel, que son diseñadas para ser lo más amplias y estables posible. La amplitud implica que cubren todas las posibles situaciones de proyectos de ciencia de datos, mientras que la estabilidad garantiza su aplicabilidad tanto en situaciones normales como en circunstancias imprevistas, como la introducción de nuevas técnicas de modelado.

El tercer nivel, el de tareas especializadas, se encarga de describir cómo realizar cada tarea genérica en situaciones específicas. Por ejemplo, en el segundo nivel puede existir una tarea

genérica llamada "limpieza de datos", y el tercer nivel detalla cómo abordar esta tarea en diferentes contextos, como la limpieza de datos numéricos frente a la limpieza de datos categóricos, o en el contexto de un problema de agrupamiento o modelado predictivo. El cuarto nivel, la instancia de proceso, consiste en un registro detallado de las acciones, decisiones y resultados obtenidos en un proyecto de ciencia de datos real.

Aunque este modelo describe las fases y tareas como pasos secuenciales idealizados, en la práctica, muchas de estas tareas pueden llevarse a cabo en diferentes órdenes, y es necesario revisitar tareas anteriores y repetir ciertas acciones. Según el modelo CRISP-DM, el ciclo de vida de un proyecto de ciencia de datos consta de seis fases, como se ilustra en la Imagen 1.

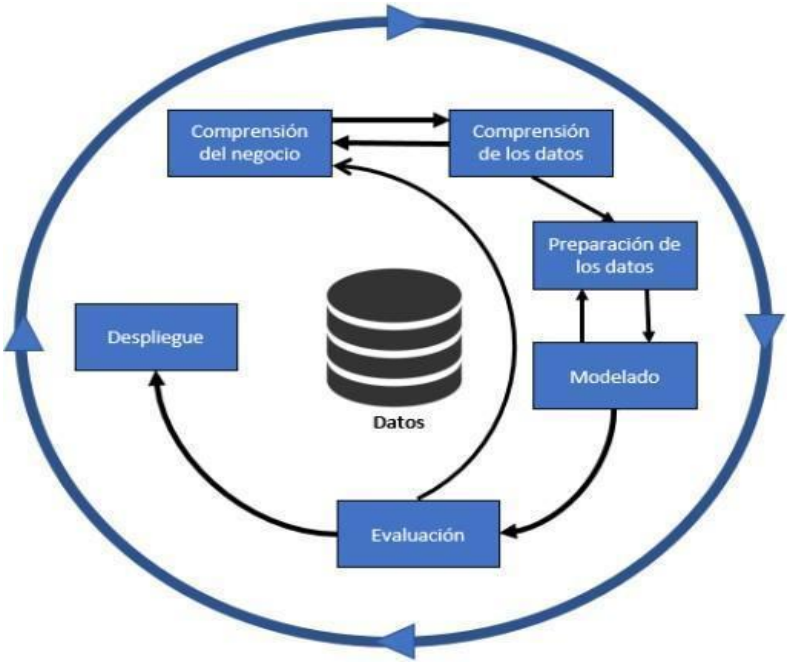

Imagen 1. Metodología CRISP-DM.

1.6 Contenido de la monografía

La presente monografía se realizó con base a la metodología CRISP-DM; la cual, de manera resumida, proporciona una descripción normalizada del ciclo de vida de un proyecto estándar de ciencia de datos [60,62], así como los objetivos específicos del trabajo de grado. Por consiguiente, es necesario enfatizar que a medida que se avanza en la monografía, los objetivos planteados se logran y se cumple así el objetivo general.

Es importante mencionar que en la metodología CRISP-DM la frecuencia de las fases no es rígida; es decir, se permite iterar en las diferentes fases del proyecto, por lo tanto, el pasar de una fase a otra no quiere decir que esta esté terminada. Aquí se presenta lo realizado en cada fase de forma general, esto no quiere decir que cada fase se completó con una única interacción o que el orden en el que se detalla cada actividad dentro de los capítulos sea estrictamente como se realizó el trabajo de grado, dado que el orden puede variar dependiendo de la necesidad planteada.

Capítulo 2

Fase 1: Entendimiento del negocio

La fase inicial de un proyecto de ciencia de datos con CRISP-DM ayuda a comprender los objetivos de negocio y requisitos comerciales que deben lograrse para el final del proyecto, por tal motivo, adquieren un carácter fundamental. Así pues, con el propósito de derivar objetivos más técnicos centrados en la ciencia de datos, al finalizar esta fase, se decidió elaborar un plan del proyecto. Con ello en mente, a continuación, se presenta un listado de las tareas a tener en cuenta en esta primera fase:

- **Entendimiento del negocio.**
    - Determinar el objetivo del negocio.
    - Fondo (background).
    - Objetivos del negocio.
    - Criterios de éxito del negocio.

- **Evaluación de la situación.**
    - Inventario de recursos.
    - Requerimientos, supuestos y restricciones.
    - Riesgos y contingencias.

- **Determinar los objetivos de la ciencia de datos.**
    - Objetivos de la ciencia de datos.
    - Criterios de éxito de la ciencia de datos.

- **Realizar el plan del proyecto.**
    - Plan del proyecto.
    - Evaluación inicial de herramientas y técnicas.

2.1 Determinar los objetivos del negocio

El objetivo principal de esta tarea es identificar los objetivos comerciales y sus indicadores de éxito. Para lograrlo, se creó un contexto que muestre la situación actual del tema central del proyecto: en este caso, los datasets que tengan actividades de la vida diaria, así mismo se buscaron proyectos o artículos alusivos a temáticas de datasets y/o que tengan proyectos que hagan uso de algún algoritmo de aprendizaje en flujo (Stream Learning), también se buscó si éstos tienen un sistema móvil donde se haya desplegado algún modelo específico. En este orden de ideas, en este punto, se realizó lo que en investigación se denomina el estado actual de conocimiento o estado del arte.

2.1.1 Fondo (background)

Esta tarea fue esencial para llevar a cabo el proyecto de ciencia de datos, pues se trabajó cuidadosamente para obtener una búsqueda tanto de datasets de ADL como de conceptos. Con esto en mente, se definieron los objetivos para identificar las brechas existentes. Se eligió una técnica de software denominada análisis de características que está incluida en el método DESMET. [60,32]

2.1.1.1 Estrategia de búsqueda

La búsqueda se realizó sin restricciones temporales, debido a que los algoritmos de Stream Learning aparecieron aproximadamente en el 2012, de este modo, el objetivo primordial fue poder obtener la mayor cantidad de datasets posibles que estuvieran relacionados con las actividades de la vida diaria y cumplieran los criterios de inclusión establecidos. La búsqueda de los datasets se realizó en las siguientes bases de datos: Scopus, UCI, Google y Unicauca, cadena de búsqueda fue construida con las siguientes palabras claves: *"TITLE-ABS-KEY ((\"open dataset\") AND ((\"HAR\") OR (“ADL”)))"*, adicionalmente en los artículos seleccionados están mencionados otros datasets de código abierto que también se tuvieron en cuenta. Para la búsqueda en el repositorio UCI se usaron las cadenas de búsqueda “ADL”, “SHAR” y “HAR”.

2.1.1.2 Criterios de inclusión

Para definir los criterios de inclusión y teniendo en cuenta la técnica que se realizó para la elección de los datasets, se realizó una lista de las características que deben tener los datasets, para poder unificarlos, así como se puede ver a continuación:

- Los datasets tienen que ser del tópico de actividades de la vida diaria.

- La búsqueda de un dataset unificado es necesaria debido a la simulación que se busca del flujo de la data real. Entre más datasets se consigan para la búsqueda más beneficia al proyecto y entrega una mayor credibilidad a las pruebas de ambos algoritmos.

- Las actividades de los datasets tienen que estar categorizadas y/o etiquetadas, para que se puedan combinar en un dataset unificado y garantizar un mejor rendimiento en los algoritmos.

- Los datasets deben contener datos de movimiento angular (giroscopio), y/o aceleración (acelerómetro).

- El tamaño de los datasets no tuvo ninguna restricción, debido a que fue de interés poder tener una cantidad de data considerable con el fin de realizar un análisis sustancioso de los algoritmos de BL y SL.

- El idioma y origen de los datasets es independiente al interés de los autores del presente trabajo de grado.

- El formato de los datasets es independiente al interés de los autores del presente trabajo de grado.

Una vez identificada la lista, fueron definidos los criterios de inclusión de la siguiente manera: (i) Los datos deben provenir de sensores tales como; acelerómetro, giroscopio, magnetómetro, sensores de presión, sensores de suelo y otras fuentes relevantes. (ii) Las actividades dentro del conjunto de datos deben estar debidamente categorizadas y etiquetadas, (iii) El conjunto de datos debe abarcar una variedad de actividades asociadas con la vida cotidiana. De este modo se garantiza una selección coherente y eficiente de los conjuntos de datos para este estudio.

2.1.1.3 Técnica de selección de datasets

Se optó por emplear la técnica de análisis de características (Feature Analysis) del método DESMET para la búsqueda y selección. Para contextualizar, es necesario definir qué es el método DESMET: Es un método de múltiples componentes que comprende pautas para realizar diversas actividades de evaluación, como la selección de métodos de evaluación, estudios de casos cuantitativos, experimentos cuantitativos y análisis de características [32].

Teniendo en claro el método DESMET, se procede a definir qué es la Técnica de Análisis de Características: Es una técnica flexible para evaluar métodos y herramientas de ingeniería de software. Consiste en identificar una lista de características requeridas o deseables en un producto y luego asignar una puntuación a cada producto en función de esas características. La puntuación puede basarse en una respuesta simple de "sí/no" para determinar si una característica específica está presente en un producto, o puede utilizar un sistema de puntuación más detallado para aumentar la confiabilidad del ejercicio de evaluación [32]. Esta técnica facilitó la creación de los criterios de inclusión, así mismo como la elección de los datasets a usar dentro de este proyecto.

2.1.1.4 Proceso de identificación de artículos relevantes

Gracias a la técnica de análisis de características del método DESMET y, después de haber realizado la lista y tener los criterios de inclusión, se implementaron tres requerimientos a tener en cuenta en la elección de los datasets encontrados, debido a que esto ayudó a realizar un análisis cuantitativo y cualitativo, como describe la técnica implementada. Gracias a esto, una vez cumplidos los criterios de inclusión y los requerimientos, se adquirió una mayor certeza de los datasets elegidos para este estudio.

2.1.1.5. Resultados

Los resultados fueron de un total agrado para los autores del presente trabajo de grado. Las dudas iniciales que aparecieron respecto a algunos datasets donde no se tenía un contacto directo y rápido con los escritores de los artículos u otros documentos, al final se lograron solucionar de manera oportuna. De este modo, después del cumplimiento de los criterios de inclusión se identificaron como resultado 29 datasets, los cuales se aprecian en la Tabla 1.

#	Plataformas de Búsqueda	Datasets Encontrados
1	Scopus	11
2	UCI	13
3	Google	2
4	Unicauca	3
TOTAL		29

Tabla 1. Datasets Encontrados en Plataformas de Búsqueda.

Como se puede apreciar en la Tabla 1, se identificaron un total de 29 conjuntos de datos, que cumplen satisfactoriamente con los criterios planteados. En el capítulo 3 de este trabajo se profundiza en el análisis de estos resultados y se termina de aplicar el método de análisis de características. Esto último con el fin de determinar los datasets a usar en esta investigación.

2.1.1.6 Brechas detectadas

Teniendo en cuenta los resultados de la búsqueda de los datasets, se tienen las siguientes brechas de conocimiento que soportan el objetivo principal de este trabajo de investigación:

- Hasta el momento de la búsqueda no se habían realizado estudios en los que se usen algoritmos de Stream Learning orientados a la clasificación de ADL.

- No se ha desarrollado un sistema móvil que despliegue y permita evaluar el rendimiento de un modelo de Stream Learning aplicado a la clasificación de ADL.

- No ha sido posible encontrar un dataset de dominio público el cual provenga de un aplicativo móvil que use un modelo de Stream Learning orientado en el ámbito de las ADL

2.1.2 Objetivos del negocio

Después de la búsqueda de los datasets, con el nivel actual de información y conocimiento de los datasets de actividades de la vida diaria y la ausencia de modelos de Stream Learning relacionado con estos datasets, se planteó el objetivo del negocio:

Construir un modelo de Stream Learning que habilite la clasificación automática de actividades de la vida diaria y desplegarlo en un sistema app/móvil en una versión inicial.

2.1.3 Criterios de éxito del negocio

El único criterio de éxito planteado para el objetivo del negocio es:

Presentación de una primera versión funcional del Sistema móvil para la clasificación automática de ADL empleando algoritmos de Stream Learning.

2.2 Evaluación de la situación

Esta tarea tiene como objetivo la descripción específica sobre todos los recursos disponibles, restricciones relacionadas al proyecto, así mismo como presunciones y otros factores que se deberían tener en consideración en la determinación del objetivo de ciencia de datos.

2.2.1 Inventario de recursos

El inventario de recursos es esencial para una planificación efectiva, ya que proporciona una visión completa de los activos disponibles y ayuda a garantizar que se utilicen de manera eficiente y eficaz a lo largo de todo el proyecto de ciencia de datos. En este orden de ideas se encontraron los siguientes recursos humanos y tecnológicos.

2.2.1.1 Personal

El talento humano es un criterio fundamental para la realización de cualquier proyecto, en esta oportunidad fueron los estudiantes de pregrado, Jaime Alfonso Pabón Rivas y Daniel Gómez Méndez, quienes, soportados por su director de tesis, el Doctor Diego López y su Co-director el Doctor Jesús David Cerón Bravo, y en general por el grupo de ingeniería telemática (GIT) de la Universidad del Cauca.

2.2.1.2 Datos

Una de las brechas detectadas fue la inexistencia de un modelo de Stream Learning para la clasificación de actividades de la vida diaria. Se evidencia entonces la necesidad de obtener un dataset unificado, a partir de los descritos en la Tabla 1 para modelar y evaluar algoritmos de Stream Learning.

Una de las brechas detectadas fue la inexistencia de un modelo de Stream Learning para la clasificación de actividades de la vida diaria. Se evidencia entonces la necesidad de obtener un dataset a partir de los descritos en la Tabla 1 con una gran cantidad de datos para simular un flujo de datos continuo con el fin de modelar y evaluar algoritmos de Stream Learning.

2.2.1.3 Recursos computacionales

Al iniciar este proyecto se dispone de dos computadores portátiles, un computador portátil intel CORE I7 7th Gen 7500U CPU @2.7 GHz 2.90 GHz, memoria RAM de 12 GB. y un Computador portátil AMD Ryzen 5 3500U CPU @2.10 GHz, memoria RAM de 8GB. Lamentablemente para el tratamiento de algunos datasets de gran tamaño, y mucho más en el momento de tener el dataset unificado, que fue la unión de todos los dataset encontrados en la búsqueda de la fase 3. Se requirió utilizar máquinas computacionales de la nube, específicamente en Google Colab, el cual ofrece un plan de pago llamado Google Colab Pro y Google Colab Pro +. donde el uso de la instancia A100 proveyó el rendimiento necesario para la unión de los datasets y el procesamiento final del dataset unificado.

2.2.2 Requisitos, supuestos y restricciones

La planificación y ejecución de cualquier proyecto de ciencia de datos requiere una consideración cuidadosa de los elementos que condicionan su desarrollo. En esta sección se presentan los requisitos, supuestos y restricciones que guiaron y limitaron el curso de esta investigación.

2.2.2.1 Requisitos

A continuación, se listan los requisitos enfocados en el proceso de ciencia de datos:

- Es fundamental que, para el cumplimiento del objetivo del negocio planteado, se obtenga un dataset unificado, así que la recolección de varios datasets de actividades de la vida diaria es esencial.

- El modelo de Stream Learning seleccionado para ser desplegado en el aplicativo móvil, debe de ser el de mejor rendimiento, es por eso que es necesaria una comparación entre diferentes algoritmos de Stream Learning y Batch Learning, para poder comprobar así mismo la hipótesis planteada.

- El aplicativo móvil corresponderá a la versión inicial, por lo tanto, se tendrá el apk disponible para la instalación en un Smartphone y se dejará un demo de sistema móvil en un servidor gratuito de amazon, para poder realizar la prueba.

- El tiempo máximo planeado para la culminación del presente proyecto es de 9 meses a partir de la aprobación del anteproyecto de grado, pero en caso de algún factor externo retrase este proyecto, se estiman 3 meses adicionales de prórroga.

2.2.2.2 Supuestos

De acuerdo con los requisitos previamente descritos, al inicio del proyecto se asumió lo siguiente:

- La recolección de datasets supone que las unidades de medida de los sensores de cada dataset sean acordes a sus valores esperados.

- Los datasets, tienen debidamente documentado su proceso de recolección y sus características.

- La capacidad de procesamiento del computador o de servidores consultados será suficiente para llevar a cabo la unión del dataset y la fase del modelado.

2.2.2.3 Limitaciones

La principal limitación se debe a la escasez de conjuntos de datos de código abierto que registren actividades de la vida diaria utilizando sensores de movimiento. Por lo tanto, la búsqueda desempeña un papel fundamental en la identificación de conjuntos de datos adecuados para este trabajo de grado.

2.2.3 Riesgos y contingencias

A continuación, se presentan los eventos que podrían retardar o hacer fallar el proyecto y sus respectivas acciones de contingencia:

- Escasez de datasets.
    - Riesgo: El número de datasets encontrados es considerablemente pequeño, por lo cual se estimaría que no estaría cumpliendo los valores de filas y columnas esperados en un dataset unificado ya que se pretende tener un dataset de gran tamaño porque así lo requiere el ámbito Stream Learning.
    - Contingencia: Para esto se debe realizar una excelente búsqueda controlada que pueda asegurar el mayor número de datasets encontrados, además de encontrar datasets que combinen datos de diferentes fuentes resulta beneficioso para mitigar el riesgo de tener un dataset muy pequeño.
- Calidad del dataset.
    - Riesgo: La calidad del dataset recolectado no es buena.
    - Contingencia: En puntuación, categorización, preparación y la limpieza de los datos se van a observar los valores atípicos para entrar en consideración el tratamiento de ellos, por lo tanto, no debería existir este riesgo.
- Pérdida de los datos.
    - Riesgo: Se debe asegurar un almacenamiento redundante de los datos recolectados, ya que la capacidad de procesamiento del computador disponible inicialmente no es suficiente para el proceso de modelado.
    - Contingencia: En este caso se hará la inversión en procesadores de Google Colab, para poder asegurar la unión del dataset y las pruebas de los algoritmos, también se tendrá en cuenta la posibilidad de pedir una máquina virtual a los servidores de la Universidad del Cauca.
- Poca mejoría de los modelos de Stream Learning.
    - Riesgo: Los modelos de Stream Learning no muestran mejoría en la clasificación respecto a los modelos offline.
    - Contingencia: En caso de que llegase a pasar algo así, se haría el respectivo análisis de las variables que llevaron a tener este escenario y se dejaría una documentación detallada, que sirva de insumo inicial para futuros proyectos relacionados

- Incapacidad de categorizar actividades.
    - Riesgo: El modelo final no es capaz de categorizar una o varias ADL.
    - Contingencia: En caso de no tener una factibilidad mínima de precisión con este modelo, se dejarán las hipótesis a plantear para el mejoramiento del modelo.

2.2.3.1 Terminología

En esta tarea se deben explicar los términos que fueron empleados a lo largo del proyecto con los que el lector se tiene que familiarizar. Para los efectos del presente trabajo los términos utilizados corresponden a términos relevantes del campo de algoritmos de aprendizaje de flujo, incremental o en línea y actividades de la vida diaria. Anteriormente se han mencionado los términos de Online Learning y Offline Learning, los cuales hacen referencia a Stream Learning y Batch Learning. Además, para simplificar su uso práctico, se proporcionan abreviaciones para estos términos:

- Decision Tree (DT).
- Batch Learning (BL).
- Random Forest (RF).
- Stream Learning (SL).
- Machine Learning (ML).
- K-Nearest Neighbors (KNN).
- Adaptive Random Forest (ARF).
- Modelos preliminares con valores aleatorios (MPCVA).
- Modelos preliminares sin valores aleatorios (MPSVA).

De este punto en adelante, y con el fin de simplificar su uso en este trabajo de grado se emplean estas abreviaturas.

2.3 Determinar objetivos de la ciencia de datos

A diferencia del objetivo del negocio, este objetivo se describe en términos técnicos con un enfoque específico en la ciencia de datos.

2.3.1 Objetivos de la ciencia de datos

Una vez reconocido el tipo de problema de ciencia de datos abordado en la clasificación y recolección, el objetivo se planteó de la siguiente manera:

Conformar un conjunto de datos unificado, a partir de los encontrados, y obtener un modelo para la clasificación de ADL a partir del mismo.

2.3.2 Criterios de éxito de la ciencia de datos

En este proyecto se tomó la precisión y el tamaño del modelo como métricas principales, por lo que el criterio de éxito fue:

Lograr un nivel de rendimiento igual o mayor a 70% en la clasificación de actividades de la vida diaria en un modelo menor a 10 MB consumiendo una RAM inferior a 0.5 GB.

Lograr un rendimiento igual o superior al 70% en la clasificación de actividades de la vida diaria en un modelo de menos de 10 MB es esencial. Esto se debe a la necesidad de mantener el tamaño del modelo en formato .pkl. Debido a que se utilizará en un dispositivo móvil fue necesario garantizar un tamaño de menor tamaño. Esta decisión presenta varias ventajas significativas. En primer lugar, un modelo de menor tamaño requiere menos memoria RAM, consumo energético y procesamiento, lo que mejora la eficiencia de los recursos y permite su ejecución en dispositivos con limitaciones de memoria.[65] Además, la carga es más rápida, lo que es fundamental para su rendimiento en sistemas móviles y para garantizar su escalabilidad. Además, un modelo más pequeño reduce los costos de almacenamiento y requiere menos ancho de banda para su transferencia, lo que resulta esencial al distribuirlo en una red. Finalmente, el porcentaje de precisión igual o superior al 70% se definió en base a ser el valor mínimo aceptable en los proyectos de ciencia de datos enfocados en la clasificación de ADLs, valor referido en los artículos científicos relacionados a los datasets detallados en el Capítulo 3.

2.4 Realizar el plan del proyecto

Finalizando la fase 1, un plan de proyecto debe contener los pasos a ser realizados en las siguientes fases del proyecto basado en CRISP-DM, las cuales corresponden a un plan de proyecto dinámico, pues al final cada fase se deberá de revisar los pendientes, y evaluar si es necesario modificar. Este enfoque adaptativo garantiza la alineación constante del proyecto con sus objetivos y metas, permitiendo ajustes precisos en función de los resultados y hallazgos.

2.4.1 Plan del proyecto

Fase	Pasos	Duración	Recursos	Entradas	Salidas
2	Recolección de datasets	3 meses	Humano: Participantes para reunir los datasets. Computacionales: Computador con acceso a internet.	Datasets encontrados.	Datasets seleccionados.
3	Preparación de los datos y unión del dataset.	3 meses	Computador o servidor. Entornos de programación	Datasets seleccionados	dataset unificado .
4	Generación y evaluación de modelos de clasificación	1 mes	Un computador o un servidor. Entornos de programación.	dataset unificado	Comparación entre modelos de Stream Learning y Batch Learning. El mejor modelo.
5	Análisis de los resultados de la evaluación de los modelos	1 mes	Un computador.	Modelos seleccionados	Análisis del modelo seleccionado
6	Implementación del modelo	1 mes	Computador, servidor	Modelo seleccionado.	Primera versión de un sistema móvil

Tabla 2. Plan general del proyecto.

2.4.2 Evaluación inicial de herramientas y técnicas

Para la elección de la herramienta de trabajo, se hizo un estudio previo de los rendimientos de las mejores herramientas o entornos de desarrollo, a sabiendas que se necesitaba una Unidad Central de Procesamiento (CPU) que tuviera un gran nivel de procesamiento. así como una Unidad de Procesamiento de Gráficos (GPU) que cuentan aún con un mayor procesamiento RAM y una memoria ROM más grande para el almacenamiento temporal de los archivos. Por tal motivo, se optó por utilizar Google Colab [66], que permite un almacenamiento en la nube y además, permite usar instancias con GPU y con mejor comportamiento computacional que los equipos de cómputo de los autores. Para los algoritmos se implementaron los algoritmos clásicos de BL con Scikit-learn y para los algoritmos de SL se utilizó Scikit-Multiflow, ambas bibliotecas Python.

Es relevante considerar que las tecnologías de familias con GPU son innovaciones recientes que ofrecen una alta capacidad de RAM, respaldando tanto los nuevos modelos generativos como el procesamiento de grandes volúmenes de datos. Se exploraron las variantes A100, T4 y V100 en busca del mejor rendimiento proporcionado por Google Colab Pro +.

Es clave tener en cuenta que las tecnologías que incorporan GPU, como las familias A100, T4 y V100 de NVIDIA, son innovaciones recientes con una amplia capacidad de RAM. Se busca alcanzar el mejor rendimiento posible con Google Colab Pro +. A continuación, se reúnen las características de la familia NVIDIA.

Nombre común	Nombre de la instancia	Características de procesamiento	Ventajas y desventajas
T4	Nvidia Tesla T41	16GB de memoria GDDR61	La T4 es una GPU versátil que ofrece un buen equilibrio entre rendimiento y costo. Es adecuada para una amplia gama de tareas, incluyendo aprendizaje profundo y aceleración de gráficos 1. Sin embargo, puede no ser tan potente como las GPUs de gama alta como la V100 y la A100 para tareas intensivas2.
V100	NvidiaTesla V1001	16GB o 32GB de memoria HBM21	La V100 es una GPU de alto rendimiento diseñada para tareas intensivas de aprendizaje profundo y computación de alto rendimiento 1. Ofrece un rendimiento significativamente mayor que la T4, pero también es más costosa 2. Algunos usuarios han informado que no siempre está disponible en Colab Pro+3.
A100	Nvidia A1001	40GB o 80GB de memoria HBM24	La A100 es actualmente la GPU más potente ofrecida por Nvidia y ofrece un rendimiento excepcional para las tareas más exigentes 1. Sin embargo, debido a su alto costo y demanda, puede no estar siempre disponible en Colab Pro+5.

Tabla 3. Características de la familia Nvidia.

Capítulo 3

Fase 2: Entendimiento de los datos

Para facilitar la comprensión de los datos, se dividió esta fase en tres componentes. El primero de ellos es la selección detallada de los datasets haciendo uso del método de análisis de características de DESMET. El segundo presenta una tabla que enumera y ordena todos los datasets encontrados y en un tercer componente, se entrega el análisis y elección preliminar de los datasets con un resumen de sus características individuales.

Todo lo anterior fue necesario debido a que la búsqueda generó gran variedad de datasets y se consideró fundamental un análisis detallado de cada uno de ellos. En la siguiente lista se ven las tareas a cumplir en esta segunda fase.

- Búsqueda de los datasets.
    - Requerimientos de los datasets

- Datasets encontrados
    - Calificación de los datasets

- Categorización de los datasets
    - Reporte de las características del dataset

3.1 Búsqueda de los datasets

Como se mencionó en la sección 2.1.1 (Fondo), se tuvieron en cuenta los criterios de inclusión i, ii y iii en la búsqueda de datasets. Posteriormente, se aplicó la técnica de análisis de características del método DESMET con el propósito de definir requisitos claros. Esto permitió la creación de casos de evaluación, donde aquellos que cumplían con todos los requisitos obtenían la mejor puntuación, tanto cuantitativa como cualitativamente, siendo esenciales para ese trabajo de grado.

3.1.1 Requerimientos de los datasets

En base a los criterios de inclusión se generan tres requerimientos para la selección de los datasets. Los requerimientos se pueden encontrar en la Tabla 4.

Requerimientos	Descripción
Requerimiento 1	El conjunto de datos debe excluir estrictamente cualquier dato proveniente de imágenes o videos. En su lugar, debe enfocarse en datos de sensores, incluyendo acelerómetros, giroscopios, magnetómetros, sensores de presión, sensores de suelo y otras fuentes relevantes.
Requerimiento 2	Las actividades dentro del conjunto de datos deben estar debidamente categorizadas y etiquetadas, permitiendo la identificación de acciones específicas como caminar, sentarse, dormir y más.
Requerimiento 3	El conjunto de datos debe abarcar una variedad de actividades asociadas con la vida cotidiana, cubriendo un conjunto diverso de acciones realizadas de forma regular.

Tabla 4. Requerimientos Datasets.

Considerando los requisitos definidos, se aplicó la técnica de análisis de características, donde se obtuvieron cinco casos distintos. Cada uno de estos escenarios fue evaluado y se le asignó una puntuación de 0 a 3 puntos, que se puede apreciar en la Tabla 5. A continuación, se presentan los casos que se analizaron.

- Caso 1.

  Este escenario representa el caso ideal en el que se satisfacen plenamente todos los requisitos, lo que resulta en una puntuación de 3. Estos conjuntos de datos fueron seleccionados de inmediato, ya que cumplen con todos los criterios necesarios.

- Caso 2.

  En este caso, se cumplen dos de los tres requisitos, lo que da como resultado una puntuación de 2. Esta situación introduce un enfoque cuantitativo junto con un análisis cualitativo. Se analizó cuidadosamente la información del conjunto de datos y sus posibles beneficios para determinar su selección.

- Caso 3.

  Solo se cumple uno de los tres requisitos, lo que resulta en una puntuación de 1. Los conjuntos de datos que entraron en esta categoría fueron descartados.

- Caso 4.

  Los conjuntos de datos que no cumplen ninguno de los requisitos reciben una puntuación de 0 y, en consecuencia, se descartaron.

A continuación, se muestra una plantilla de resumen de las calificaciones que se usó para la calificación de los datasets.

Casos	Requerimiento 1	Requerimiento 2	Requerimiento 3	Calificación
Caso 1	x	x	x	3
Caso 2	x		x	2
	x	x		
		x	x	
Caso 3	x			1
		x		
			x	
Caso 4				0

Tabla 5. Casos de evaluación.

3.2 Datasets encontrados

La calificación de los datasets encontrados requirió un gran esfuerzo, no obstante, se deja aquí el *Anexo A - Búsqueda de los datasets*, donde se listan todos los datasets encontrados. El archivo Excel se conforma de: un indicador numeral, nombre del dataset, lugar donde se encontró, fecha del dataset, requerimiento 1, requerimiento 2, requerimiento 3, calificación y una nota de los autores de algún comentario a resaltar del dataset encontrado.

3.2.1 Clasificación de datasets

Considerando los criterios de búsqueda establecidos, se presentan a continuación los datasets encontrados, junto con sus respectivas puntuaciones. En la Tabla 6 de resultados resumida de la búsqueda de datasets. Esta tabla se puede obtener completa en el *Anexo A*.

#	Referencia	Requerimiento 1	Requerimiento 2	Requerimiento 3	Puntaje
1	[37]	x	x	x	3
2	[38]		x	x	1
3	[39]	x	x	x	3
4	[40]		x	x	2
5	[41]		x	x	2
6	[42]		x	x	2
7	[43]	x	x	x	3
8	[44]		x		1
9	[45]		x		1
10	[46]		x	x	1
11	[47]	x	x	x	3
12	[48]	x	x	x	3
13	[49]	x	x	x	3
14	[50]	x	x	x	3
15	[51]			x	1
16	[52]	x	x	x	3
17	[53]			x	1
18	[54]	x	x	x	3
19	[55]	x	x	x	3
20	[56]	x	x	x	3
21	[57]	x	x	x	3
22	[58]	x	x	x	3
23	[25]	x	x	x	3
24	[26]	x	x	x	3
25	[27]	x	x	x	3
26	[28]	x	x	x	3
27	[29]	x	x	x	3
28	[30]	x	x	x	3
29	[31]	x	x	x	3

Tabla 6. Resumen de Puntuación Datasets Encontrados

En este orden de ideas, se puede apreciar que la cantidad de datasets encontrados fueron 29, en la siguiente Tabla 7 se resumen los resultados:

Calificación	Cantidad de Datasets
3	20
2	3
1	6
0	0
Total	29

Tabla 7. Cantidad de Datasets por Calificación

Es importante destacar que los conjuntos de datos con una puntuación de 3 fueron seleccionados sin ningún análisis cualitativo; los conjuntos de datos con una puntuación de 2 fueron sometidos a un análisis más detallado; sin embargo, se descartaron para preservar la integridad y consistencia del conjunto de datos. Por último, los conjuntos de datos con puntuaciones de 1 o 0 fueron excluidos sin consideración adicional. Es de destacar que gracias a el buen uso de llaves de búsqueda los datasets con calificación 0 fueron nulos.

Finalmente, se obtuvo un conjunto de 20 datasets que tuvieron su proceso de categorización individual, que se puede apreciar en la siguiente sección.

3.3 Categorización de datasets

Una vez fueron seleccionados los 20 datasets en el ítem anterior, se procedió a realizar una categorización que se encuentra completa en el *Anexo B - Categorización de Datasets* que está conformado con los 20 datasets, con la información basada en el número de identificador, nombre del dataset, descripción, actividades (número de actividad, nombre de la actividad, duración de la actividad), personas (edad promedio, estatura promedio, peso promedio, restricciones, total de personas, notas), dispositivos (número de dispositivo, nombre del dispositivo, marca del dispositivo, ubicación del dispositivo, sensor del dispositivo, unidades, marca del sensor, tasa de muestreo, controlado, no controlado, lugar de la actividad), peso del dataset y notas adicionales. Toda esta información fue de suma importancia para la ejecución del siguiente capítulo. En la Tabla 8 se presenta un resumen de la categorización de los datasets.

#	Ref.	Conteo	Actividades	Sensores
1	[26]	11	Sentadilla, bajar escaleras, subir escaleras, saltar, trotar, acostarse y levantarse de la cama, sentarse y levantarse de una silla, caminar normalmente, caída hacia atrás, caída hacia delante y caída lateralmente.	Acelerómetro y giroscopio
2	[58]	4	Inactivo, activo, caminando (caminando, corriendo y trotando) y manejando.	Acelerómetro, magnetómetro, GPS y Giroscopio.
3	[57]	12	Caminando hacia adelante, girando a la izquierda, girando hacia la derecha, subiendo escaleras, bajando escaleras, corriendo hacia adelante, saltando, estar sentado, estar de pie, durmiendo, subiendo por el ascensor y bajando por el ascensor.	Acelerómetro y Giroscopio
4	[56]	6	Caminar, correr, subir escaleras, bajar escaleras, estar sentado y estar de pie.	Acelerómetro
5	[55]	19	Acostado, sentado, de pie, caminando, corriendo, ir en bicicleta, caminata nórdica, ver la televisión, trabajar en el ordenador, conducir el coche, subir escaleras, bajar escaleras, limpiar con la aspiradora, planchar, doblar la ropa, limpiar la casa, jugando futbol, saltando la cuerda, otro (actividades transitorias).	Acelerómetro, magnetómetro, Giroscopio, Temperatura y Ritmo cardíaco
6	[54]	13	Sentado, acostado, de pie, lavar los platos, aspirar, barrer, caminar al aire libre, subir escaleras, bajar escaleras, correr en la cinta de correr (8,3 km/h), bicicleta (50 vatios), bicicleta (100 vatios) y salto con cuerda.	Acelerómetro Giroscopio
7	[52]	3	Caminando en el pasillo, caminando bajando las escaleras y subiendo las escaleras.	Acelerómetro, magnetómetro Giroscopio
8	[50]	5	Caminando, saltando, llamando por teléfono, saludar (ondulando la mano) y escribiendo en el celular.	Acelerómetro, magnetómetro y Giroscopio
9	[49]	12	Caminando, subiendo las escaleras, bajando las escaleras, estar sentado, estar de pie, estar acostado, pasar de estar de pie a estar sentado, pasar de estar sentado a estar de pie, pasar de estar sentado a acostarse, pasar de estar acostado a sentarse, pasar de estar de pie a acostarse, pasar de estar acostado a estar de pie.	Acelerómetro y Giroscopio.
10	[48]	6	Caminando, bajar las escaleras, subir las escaleras, estar sentado, estar de pie y estar acostado.	Acelerómetro y Giroscopio.
11	[47]	9	Caminar, sentarse, pararse, abrir la puerta, cerrar la puerta, verter agua, beber de un vaso, cepillarse los dientes y limpiar la mesa.	Acelerómetro Giroscopio.

#	Ref.	Conteo	Actividades	Sensores
12	[43]	19	Estar sentado, estar de pie, acostado boca arriba, acostado sobre el lado derecho, subir escaleras, bajar escaleras, de pie en un ascensor sin moverse, moviéndose en el ascensor, caminando en un parqueadero, caminando en una caminadora a 4 km/h en posición plana, caminando en una caminadora a 4 km/h con inclinación de 15 grados, corriendo en una caminadora a 8 km/h, ejercitándose en un stepper, ejercitándose en una elíptica, andar en bicicleta estática en posición horizontal, andar en bicicleta estática en posición vertical, remar, saltar y jugar baloncesto.	Acelerómetro, magnetómetro, Giroscopio.
13	[39]	18	Caminar, trotar, escaleras, estar sentado, estar de pie, escribir en el celular, cepillarse los dientes, comiendo sopa, comiendo papas en paquete, comiendo pasta, bebiendo de una copa, comiendo un sándwich, pateando un balón, jugando atrapala con una bola de tenis, driblar (baloncesto), escribir, aplaudir y doblando ropa.	Acelerómetro, Giroscopio.
14	[37]	6	Estar de pie, estar sentado, estar acostado, caminando, subir las escaleras y bajar las escaleras.	Acelerómetro, Giroscopio.
15	[27]	3	Manejando, durmiendo y viendo televisión.	Acelerómetro, magnetómetro, GPS y Giroscopio
16	[28]	116	Acostado, sentado, de pie en un lugar, de pie moviéndose, caminando, corriendo y bicicleta. Adicionalmente, 109 actividades secundarias.	Acelerómetro, Magnetómetro, GPS, Giroscopio, Audio, otros.
17	[29]	7	Caminando, escaleras, estar quieto, usando una jarra, barrer, usar el lavamanos y usar el inodoro.	Acelerómetro, magnetómetro. Giroscopio.
18	[30]	7	Entrar al apartamento (bajar las escaleras, abrir la puerta y entrar), quitarse la chaqueta (entrar al cuarto y dejar la chaqueta ahí), servirse algo para comer e ir al comedor a comer, barrer (tomar la escoba del baño, barrer la sala y devolver la escoba), peinarse (ir al baño, tomar el peine y peinarse), ir al baño (levantar la tapa y sentarse en el baño) y salir del apto (ir por la chaqueta, tomarla y salir).	Acelerómetro, Giroscopio.
19	[31]	3	Caminando, trotando y corriendo.	Acelerómetro. Giroscopio.
20	[25]	10	Caminando, sentarse, acostarse, correr, subir escaleras, bajar escaleras, estar de pie, caer por inconsciencia, caer por ataque al corazón, y caer por resbalar mientras camina.	Acelerómetro.

Tabla 8. Categorización de datasets resumen.

El análisis de los datasets seleccionados que se aprecian en la Tabla 8, se encuentran en el *Capítulo 4*.

Capítulo 4

Fase 3: Análisis exploratorio de los datos (EDA)

El análisis exploratorio de los datos (EDA) es de gran importancia en el proceso de la ciencia de datos, ya que asegura que los datos sean adecuados y estén listos para ser utilizados en las etapas posteriores del proyecto. En esta fase, se emplearon diversas técnicas para transformar, limpiar y estructurar los datos de manera que sean útiles y relevantes para el problema que se está abordando. [24]

Existen varias técnicas disponibles en la preparación de datos, que incluyen la descripción, el resumen de datos, la segmentación, la predicción, la clasificación y la conceptualización. Sin embargo, no todas estas técnicas son aplicables de la misma manera a todos los problemas. La elección de las técnicas de modelado que se utilizaron en la siguiente fase están estrechamente relacionadas con la forma en que se deben preparar los datos. En síntesis, esta fase se encarga del análisis, la preparación de los datos y por último en la obtención del dataset unificado con el que se trabajará en la siguiente etapa del modelado.

Adicionalmente en este proyecto de investigación se encontraron 2 grandes retos:

1) El análisis y depuración individual de datos de cada uno de los datasets seleccionados en el anterior capítulo que permitió la creación de un dataset unificado.
2) La ciencia de datos, específicamente la clasificación, donde se obtuvieron los modelos de ML capaces de clasificar ADL, que permitieron el análisis entre BL y SL.

El análisis mencionado en el reto 2 tomó en consideración diferentes parámetros que se mencionan en el Capítulo 5; de momento, se presenta la lista de tareas correspondientes a este capítulo, las cuales tienen como objetivo principal la obtención del dataset unificado, su preparación, limpieza y acondicionamiento para las etapas posteriores de entrenamiento.

- **Análisis de características de los datasets.**
    - Actividades
    - Sensores
    - Unidades.
    - Frecuencia.
    - Ubicación del sensor.
    - Definición de características (columnas) del dataset unificado.
- **Limpieza y preparación de los datasets.**
    - Preparación de cada Dataset.
    - Técnica de remuestreo.
    - Normalización de unidades.
    - Datasets seleccionados antes de la unión.
    - Depuración datos atípicos.

- ID de actividades.
- Resumen de datasets elegidos (segunda etapa)
- Unión del dataset

- **Normalización del Dataset.**
  - Normalización de rangos.

- **Estructurar, integrar y formatear los datos.**
  - Segmentación del dataset.
  - Exploración de características.

- **Descripción del dataset transformado.**

4.1 Análisis de características de los datasets

El análisis de características es fundamental en la ciencia de datos, debido a que permite descubrir patrones, probar hipótesis y tomar decisiones fundamentadas en los datos [69]. Debido a la cantidad de datasets que se trataron, se eligieron características claras que aseguraron una correcta concatenación de toda la información en un dataset unificado, para ello se eligieron las siguientes variables para la unificación: sensores, unidades, frecuencia, actividades y ubicación del sensor. Ya con esta información analizada se definió la conformación de las columnas finales del dataset unificado.

4.1.1 Actividades

Uno de los análisis más cruciales del presente proyecto es la clasificación de actividades. Un total de 293 actividades están presentes en los 20 datasets. El dataset 16 destaca dado que tiene 7 actividades principales y 109 actividades secundarias. La Imagen 2 muestra una distribución porcentual de las actividades en los datasets.

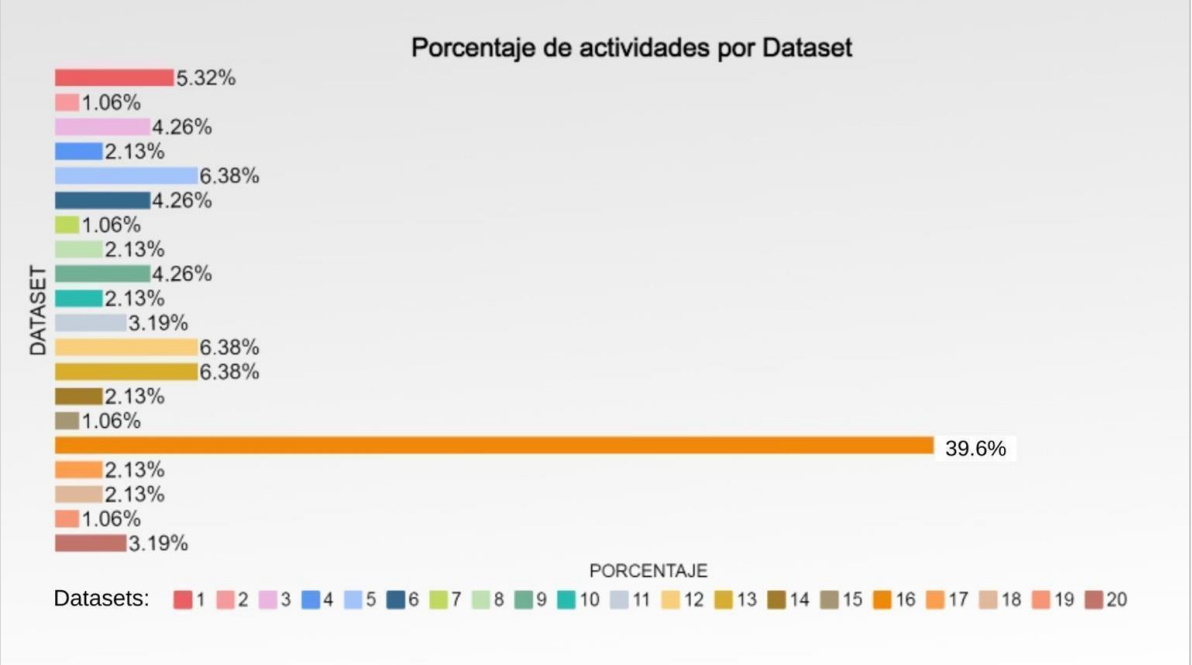

Imagen 2. Cantidad de actividades por dataset.

Como se puede observar, el dataset 16 se destaca debido a la gran variedad de actividades secundarias que contiene, abarcando un 39.6% del total de actividades entre los 20 datasets disponibles. Estos porcentajes tuvieron cambios significativos después de la limpieza de datos, así que esta imagen se deja como una gráfica de referencia de los datos no depurados.

Adicionalmente, se realizó un análisis del número de actividades repetidas, el cual se detalla en el *Anexo C - Actividades Resumen*, donde se presenta la cantidad de actividades repetidas en cada dataset. La Tabla 9 es un resumen del *Anexo C*. Es importante mencionar que para la creación y mejora del modelo en los siguientes capítulos, se tuvo en cuenta esta información de las actividades más repetidas y relevantes de esta sección.

#	Referencia Dataset	Bajar Escaleras	Caminando	Estar Acostado	Estar De Pie	Estar Sentado	Subir Escaleras
1	[26]	X	X				X
2	[58]		X				
3	[57]	X			X	X	X
4	[56]	X	X		X	X	X
5	[55]	X	X	X	X	X	X
6	[54]	X		X	X	X	X
7	[52]						
8	[50]		X				
9	[49]	X	X	X	X	X	X
10	[48]	X	X	X	X	X	X
11	[47]		X				
12	[43]	X			X	X	X
13	[39]		X		X	X	
14	[37]	X	X	X	X	X	X
15	[27]						
16	[28]		X	X		X	
17	[29]		X				
18	[30]						
19	[31]		X				
20	[25]	X	X	X	X	X	X
TOTAL		**10**	**14**	**7**	**10**	**11**	**10**

Tabla 9. Conteo de actividades más comunes por datasets.

4.1.2 Sensores

Para el análisis de los sensores no te tuvo en cuenta la marca del sensor del análisis de características; debido a que no se usó más allá de la verificación de las unidades con el *datasheet* y para tener un mejor entendimiento de la calibración de los sensores en casos en los que en los artículos no se especificaba. En la Tabla 10, se puede apreciar las veces que se repitió un sensor por dataset, donde en los 20 datasets seleccionados destaca en primer lugar el acelerómetro y en segundo lugar el giroscopio. Debido a la diferencia entre el acelerómetro y giroscopio con los demás sensores, se decidió trabajar con los datos de estos dos sensores para evitar un desbalance de datos y mantener la homogeneidad en el dataset unificado.

Sensores/Dataset	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	Total
Acelerómetro	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	20
Giroscopio	x	x	x		x	x	x	x	x	x	x	x	x	x	x	x	x	x	x		18
Magnetómetro		x			x		x	x				x			x		x				7
Temperatura					x																1
Ritmo cardiaco					x																1
Sensor humedad																x					1
Presión del aire																x					1
Sensor proximidad																x					1
Sensor luz																x					1
GPS		x													x						2
Micrófono															x						1
Compas																x					1

Tabla 10. Conteo de Sensores por datasets.

4.1.3 Unidades

Para asegurar la comparabilidad de los datos, se debe obtener un buen análisis de las unidades, en especial verificando que los valores dados sean acordes con las unidades que se mencionan en cada dataset. Para esto se realizó un análisis a profundidad en la normalización de unidades, como se puede observar en la Tabla 11.

Unidades/Dataset	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	Total
Acelerómetro																					
g	x		x		x	x	x		x		x	x				x	x	x	x	x	13
m/s^2		x		x				x		x			x	x	x						7
Giroscopio																					
rad/s		x			x			x	x	x			x			x					9
grados/s	x		x			x	x				x	x		x	x		x	x	x		11

Tabla 11. Conteo de unidades por datasets

4.1.4 Frecuencia

Para facilitar el procesamiento de los datos, es necesario llevar todas las frecuencias de los datasets a una misma frecuencia. Se tiene que la moda de frecuencia de todos los datasets es de 50 Hz (Tabla 12). Se decide entonces llevar un análisis más profundo el cual se puede revisar en la técnica de remuestreo, en la sección 4.2.2.

Frecuencia (Hz)/Dataset	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	Total
200	X					X											X	X	X		5
20	X			X									X								3
0.13		X																			1
4.46		X									X										2
4.66		X																			1
4.66		X																			1
6.24		X																			1
7.91		X																			1
8.16		X																			1
9		X			X																1
9.13		X																			1
11		X																			1
17		X																			1
20		X																			1
25		X																			1
30		X																			1
31.24		X																			1
100			X		X										X						3
40																					1
50							X	X	X	X				X							6
51.16												X									1
100																X					1
204.8																				X	1

Tabla 12. Conteo de frecuencia por dataset.

4.1.5 Ubicación del Sensor

Cada dataset fue recolectado usando sensores ubicados en diversas partes del cuerpo de los usuarios. El resumen de estas ubicaciones se muestra en la Tabla 13, que contiene información del *Anexo B - Categorización de los Datasets*.

Ubicación/Dataset	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	Total
Bolsillo derecho	x												x		x						3
Tobillo	x				x																2
Cintura	x								x	x											3
Muñeca derecha	x		x		x	x	x	x					x	x				x		x	10
Pecho	x				x	x															3
Cadera			x			x							x								3
Bolsillo frontal del pantalón				x																	1
Zapato						x											x	x	x		4
Parte baja del brazo izquierdo											x										1
Parte baja del brazo izquierdo											x										1
											x										1
Parte alta del brazo izquierdo											x										1
											x										1
Parte baja del brazo derecho											x										1
											x	x									2
Parte alta del brazo derecho											x	x									2
Muslo derecho											x										1
Espalda											x										1
Torso											x										1
Pierna derecha												x									1
Pierna izquierda												x									1
Muebles															x						1
Carro															x						1
Mano dominante																x				x	2

Tabla 13. Conteo de ubicación del sensor por dataset.

4.1.6 Definición columnas del dataset unificado

Teniendo en cuenta el análisis de características que se realizó en las anteriores secciones, se definieron las características que debe contener el dataset unificado. En la sección de limpieza de datos, se ajustó cada dataset para que cumpla con la estructura que se muestra en la Tabla 14.

Columnas	Descripción	Tipo de dato
Dataset	Identificador numérico del Dataset	Int
Activity	Identificador de la Actividad	Int/Subject
X_Acc	Medición de aceleración del eje X	Float
Y_Acc	Medición de aceleración del eje Y	Float
Z_Acc	Medición de aceleración del eje Z	Float
X_Gyro	Medición de giroscopio del eje X	Float
Y_Gyro	Medición de giroscopio del eje Y	Float
Z_Gyro	Medición de giroscopio del eje Z	Float
Code	Código	Int
TimeStamp	Marca de tiempo de la muestra	DateTime
Frequency	Frecuencia del sensor	Float
User	Persona que realizó la actividad	Int
Trial	Número de prueba de la actividad	Int

Tabla 14. Características del conjunto de datos unificado: Registros de actividades y mediciones de sensores.

En el análisis de características se contempló la necesidad de tener una columna de código llamada “Code” la cual se compone de 3 dígitos de identificación. Este código facilitó el análisis en la unión del dataset y la etapa de la creación del modelo, debido a que está conformado por: tipo de sensor, lado de la ubicación y ubicación del sensor. En la Tabla 15, se puede apreciar cómo se compone la codificación.

Code		
Tipo de sensor	Giroscopio	1
	Ambos	2
	Acelerómetro	3
Lado de la ubicación	No info	0
	Derecha	1
	Izquierda	2
Ubicación del sensor	No info	0
	Muñeca	1
	Cadera	2
	Pie	3
	Pecho	4
	Espalda	5
	Brazo	6
	Pierna	7
	Cintura	8
	Otros	9

Tabla 15. Code: Conformación de los valores del código.

4.2 Limpieza y preparación de los datasets

4.2.1 Preparación de cada dataset

La sección de limpieza en la ciencia y análisis de datos suele ser una de las más laboriosas debido a su complejidad. En este caso, no fue la excepción, ya que cada dataset tenía su propia distribución de información, lo que implicaba procedimientos específicos de limpieza y estandarización. Algunos datasets estaban disponibles como varios archivos .csv en una misma carpeta, mientras que otros estaban dispersos en diferentes carpetas, subcarpetas y con una variedad de extensiones de archivos (como .txt, .mat, .csv, .aff, .dat, etc.) y sin una documentación clara de los datos brutos. Esta diversidad de formatos y estructuras hizo que la limpieza y unión de los datasets fuese una de las tareas más desafiantes en este proyecto. Lamentablemente, debido a diversas razones, se tuvieron que descartar algunos de los 20 datasets originales, quedando finalmente 15, lo cual se explicará más adelante.

En línea con lo anterior, es esencial proporcionar una explicación detallada del tratamiento aplicado a cada uno de los datasets. Esto es crucial para respaldar de manera sólida y confiable el logro del primer objetivo específico. Todo el proceso de tratamiento de los datasets se llevó a cabo en Google Colab, y se documenta en los cuadernos parte 1, 2 y 3 de "Preparación de los datasets" disponible en el *Anexo G* - *Repositorio Github* de este documento.

Cabe resaltar que para cada uno de los datasets se implementaron los siguientes pasos:

- PASO 1.

  Buscar un artículo donde se describiera cómo se realizó la construcción del dataset. Usualmente se encontraba una descripción de los valores de los sensores, del tipo de unidades, una información más detallada de los sensores y comentarios adicionales que permitían tener un contexto más claro del dataset escogido. Es importante aclarar que no todos los datasets contaban con un artículo y también que no todos los artículos proveían toda esta información requerida.

- PASO 2.

  Si el dataset no tenía un artículo de referencia o el artículo no tenía la suficiente información, se revisaron los "readme" que tenían algunos datasets, o las descripciones de los datos que se anexan dentro de algún archivo con las medidas de los sensores. Si era necesario, también se buscaron otros artículos y/o en las páginas donde potencialmente se pudiera encontrar información del dataset. En ocasiones, la información encontrada fue insuficiente o decía estar en sitios web descontinuados. En estos casos se intentó contactar directamente a los autores.

- PASO 3.

  Teniendo en cuenta la realización de los pasos 1 y 2, y el contexto de los datasets, se tomaba la decisión de conservar o no el dataset para la obtención del dataset unificado.

A continuación, se listan los aspectos más relevantes del tratamiento de cada dataset y su adaptación al formato .csv necesario para la unión de los datasets. La información detallada sobre actividades, sensores y unidades de los datos se encuentra en el *Anexo B - Categorización del Dataset*.

4.2.1.1 UMAFall: Fall Detection Dataset [26]

El dataset estaba distribuido en 747 archivos .csv, donde cada archivo hace referencia a las pruebas realizadas por los usuarios. Cada archivo .csv cuenta con 40 filas de información descriptiva del dataset, al finalizar las 40 filas se encuentra la información de los valores que son: Timestamp, Mediciones del Sensor (X, Y y Z), Sensor Type (tipo del sensor: acelerometro, giroscopio, magnetómetro) y Sensor ID (Identificador del sensor). Para la depuración se realizó lo siguiente:

- En cada uno de los archivos *.csv* fueron eliminadas las primeras 40 filas de información descriptiva. Para esto se desarrolló un código que tomaba cada uno de los archivos *.csv*, eliminando las 40 filas y guardando los 747 archivos en una nueva carpeta donde en la primera fila de cada archivo se apreciaron las columnas de información.

- Los nombres de cada archivo se conformaban con la información de: sujeto, actividad, prueba de la actividad y fecha. Para esto se desarrolló un código que permitió la extracción de estas características, creando cuatro columnas adicionales dentro de los 747 archivos.

- Para la obtención de la columna de Timestamp, se combinó la información del Timestamp que ya venía en la información inicial de los archivos *.csv*, que otorgaba los milisegundos y se incorporó al valor de la fecha, extraída en el ítem anterior, para al final poder obtener una nueva columna del verdadero "TimeStamp".

- Para la creación de la columna "Code" de este dataset, se desarrolló un código que extrajo el tipo de sensor utilizado de la columna "Sensor Type", y la ubicación de la columna "Sensor ID", para posteriormente eliminar estas dos columnas. Además, se eliminaron las filas que poseían el identificador del magnetómetro.

Como resultado fue realizada la división entre los datos con frecuencia de 200 Hz y 20 Hz creando así 2 datasets, *dataset1_1.csv* y *dataset1_2.csv*.

4.2.1.2 A Public Domain Dataset For Real-Life Human Activity Recognition Using Smartphones Sensors [58]

El dataset inicialmente estaba compuesto de 4 archivos *.csv*, cada uno correspondiente a un sensor específico con información esencial: datos del usuario, marcas de tiempo, mediciones del sensor y actividades realizadas. La frecuencia de muestreo en los archivos originales no se proporciona de manera detallada. Sin embargo, el artículo asociado al dataset ofreció rangos de frecuencia para cada actividad, ya que los datos fueron recolectados con los sensores de los teléfonos móviles de los participantes y no se estableció una frecuencia uniforme. Para la depuración se realizó lo siguiente:

- Se desarrolló un código que contabilizó la cantidad de registros por segundo de cada actividad para cada usuario. Se utilizaron estas cantidades para calcular el promedio, la mediana y la moda de la frecuencia en cada caso. Lamentablemente, algunos datos resultaron inconsistentes lo que llevó a eliminar usuarios y actividades con frecuencias irregulares.

- Para facilitar futuros procesos de resampling, se estandarizó una frecuencia de 50 Hz para el acelerómetro y 5 Hz para el giroscopio, respectivamente. Debido a esta diferencia de frecuencias se dividió este dataset en 2 partes.

- El dataset presentó 4 tipos de actividades, "Active, Inactive, Walking y Driving", las tres primeras fueron descartadas ya que al momento de leer las descripciones y revisar el tipo de data la etiqueta agrupaba diferentes tipos de actividades en una misma dando poca confiabilidad a sus datos, dejando únicamente la actividad Driving, ya que las actividades que agrupa sí coinciden completamente con su etiqueta de "Driving".

Como resultado, se obtuvieron dos conjuntos de datos finales: "Dataset_2_part_1_Only_Driving.csv" y "Dataset_2_part_2_Only_Driving.csv".

4.2.1.3 The USC-SIPI Human Activity Dataset [57]

La carpeta original se conforma por 14 carpetas cada una haciendo referencia a un sujeto, dentro de cada una de estas carpetas se encontraban archivos .mat, donde estaban compuestos por un nombre en clave, por ejemplo “a1t1”, que significa actividad uno y prueba 1. De cada actividad se tienen 5 pruebas, y se compone de 12 actividades, dando un total de 60 archivos por persona, es decir un total de 840 archivos. Para la depuración se realizó lo siguiente:

- Se desarrolló un código que permitió recorrer las 14 carpetas y extraer cada archivo .mat, transformándolo a un archivo .csv y guardándolo en otras 14 carpetas, manteniendo su nombre.

- Para la unión de estos archivos *.csv*, y obtener un archivo *.csv* por usuario donde se tuviera toda la información de sus actividades, se desarrolló un código que permitió la unión de todos los archivos de cada uno de los usuarios, generando un archivo *.csv* de las actividades por usuario, es decir 14 archivos, que se guardaron en una nueva carpeta. Donde se unificaron todos los archivos *.csv*, obteniendo así un dataset depurado.

La división del conjunto de datos en tres secciones delimitadas por comas, y se incluyeron las columnas faltantes en el archivo "USC_HAD_Final_Definitivo.csv", que se terminó llamando Dataset_3.csv.

4.2.1.4 WISDM Wireless Sensor Data Mining [56]

Originalmente el dataset estaba distribuido en 4 archivos. Tres archivos de datos en crudo con formato *.txt*: donde los dos primeros hacían referencia a la descripción de los datos sin procesar. El tercero hacía referencia a los datos sin procesar, y finalmente un archivo *.arff*. El archivo de datos sin procesar se conformaba por un ID de usuario, actividad, timestamp y los valores de acelerómetro. Para la depuración se realizó lo siguiente:

- La realización del análisis únicamente al archivo de datos sin procesar, omitiendo los archivos transformados en formato *.arff* y de descripción.

- Se implementó un código que transformó el archivo de formato *.txt* a *.csv*.

- Para terminar el dataset se construyó un código que permitió añadir un encabezado a las columnas existentes y la asignación del código, ID del dataset y frecuencia.

Con lo anterior se obtuvo el archivo final denominado Dataset_4.csv.

4.2.1.5 PAMAP2 Physical Activity Monitoring Dataset [55]

Originalmente se conformaba de dos carpetas llamadas “*Protocol*” y “*Optional*”. *Protocol* tenía actividades obligatorias y *Optional* actividades opcionales. Las actividades eran de cada usuario y tenían el formato *.txt*. Cada archivo *.txt* tenía el formato de nombre "*subject10x*", donde la variable "x" variaba según el sujeto. Estos archivos constaban de 54 columnas, abarcando datos como la marca de tiempo, actividad, frecuencia cardíaca e información de Unidad de Medición Inercial (IMU) para la mano, pecho y tobillo. Los datos IMU incluían temperatura, lecturas de acelerómetro 3D (±16g y ±6g), giroscopio 3D, magnetómetro 3D y orientación. Para la depuración se realizó lo siguiente:

- Haciendo uso de un bucle while, se abrió cada archivo *.txt* como un archivo .csv asignando encabezados correspondientes.

- La asignación de encabezados, la eliminación de columnas superfluas y la agregación de nuevas columnas para identificar al usuario. Posteriormente, se concatenaron los archivos *.csv* individuales.

- La división del dataframe en tres partes según la ubicación de IMU (tobillo, pecho y mano).

- La asignación y fusión horizontal de los códigos específicos de los dataframes creando los *.csv* de *Protocol y Optiona*l.

Finalmente, la fusión vertical entre *Protocol* y *Optiona*l se presenta como Dataset5_Final.csv.

4.2.1.6 DaLiAc Dataset [54]

Desafortunadamente el link de descarga del dataset tenía un error 404. Se contactó con los autores del dataset vía correo electrónico, los cuales otorgaron el dataset satisfactoriamente. En un principio el conjunto de datos constó de 19 archivos *.txt*, cada archivo tiene los datos de un usuario. Los archivos contenían información de los sensores de acelerómetro y giroscopio de las 4 ubicaciones de las IMU lo que resultó en un total de 24 columnas de medidas, además de la columna de etiquetas. Para la depuración se realizó lo siguiente:

- Para poder reducir las 24 columnas de información de acelerómetro y giroscopio en 6 columnas, se diseñó un código que permitió una concatenación vertical de los archivos *.txt*, así mismo la asignación del ID de usuario, y por último la transformación al formato *.csv*.

- El archivo *.csv* fue dividido por ubicación para asignar los respectivos códigos de ubicación, para crear la columna “Code”.

Como resultado de este proceso, se obtuvo el conjunto de datos final, presentado como "Dataset_6.csv".

4.2.1.7 IMU Dataset: Walking Activity Recognition using Inertial Measurement Unit Modules [52]

Debido a la ambigüedad inherente en la información encontrada en este dataset en los primeros acercamientos, surgió la necesidad de revisar, reestructurar repetidamente los datos por efecto de las inconsistencias presentes en el artículo de referencia. Se intentó contactar con los autores, pero no dieron respuesta.

La exclusión de este dataset se debió al número de filas presentes insuficientes para el esfuerzo y el tiempo que estaba requiriendo.

4.2.1.8 HARBox Dataset: Daily Activities Recognition using Smartphones [50]

En un comienzo el dataset constaba de 120 carpetas, cada carpeta correspondiente a un usuario. Las carpetas se conformaban de archivos *.txt* de las actividades realizadas por el usuario, generalmente 5 (aunque algunos usuarios no realizaron todas las actividades). Estos archivos *.txt* contenían 10 columnas que abarcaban información de marca de tiempo y mediciones de sensores (acelerómetro, giroscopio y magnetómetro). Para la depuración se realizó lo siguiente:

- Se realizó un código que permitió recorrer las 120 carpetas, tomando y uniendo los archivos *.txt*, posterior a esto se le asignaron los nombres de columnas faltantes a los datos de los sensores, posterior a ello, de los archivos los nuevos archivos *.txt* por carpeta se convirtió a *.csv*, teniendo 120 archivos *.csv* representando a cada usuario.

- Estos archivos *.csv* también se concatenaron asignando el ID de cada usuario, eliminando la información del magnetómetro con lo cual se obtiene un único archivo *.csv*.

- Para completar el proceso, se incorporó información sobre la frecuencia de muestreo, el ID del conjunto de datos y el código específico.

El resultado final se presentó como el archivo Dataset_8.csv.

4.2.1.9 Smartphone-Based Recognition of Human Activities and Postural Transitions Data Set [49]

En un inicio el dataset constaba de 123 archivos *.txt*. Un archivo, llamado “labels.txt” tenía la información necesaria de las etiquetas, los otros 122 archivos, tenían los valores de las mediciones de los sensores sin procesar, y además se conformaban en diferentes experimentos o pruebas. Cada usuario realizó dos experimentos. Los nombres de estos archivos seguían tenían este formato "acc_exp01_user01.txt". Para la depuración se realizó lo siguiente:

- Se diseñó un código que permitió la relación de las etiquetas del archivo “labels.txt” con los 122 archivos, obteniendo actividades etiquetadas.

- Los segmentos etiquetados se exportaron como archivos individuales y se concatenaron para crear una columna de actividad correspondiente a los archivos *.txt* originales. Además, se combinaron verticalmente los 122 archivos, extrayendo los valores "Trial (exp)" y "User (user)" de cada uno y añadiendolos como columnas separadas.

Se añadió información sobre la frecuencia de muestreo, el número de dataset y el código correspondiente al dataframe resultante. El dataframe final se presentó como "Dataset_9.csv".

4.2.1.10 Human Activity Recognition Using Smartphones Data Set [48]

El análisis del “readme” del dataset 10 y del repositorio UCI donde se encontraba el dataset brindó información a los autores sobre que este dataset es una versión previa del dataset 9 y por lo tanto se tomó la decisión de excluirlo de la depuración de los datasets.

4.2.1.11 A multi-sensory dataset for the activities of daily living [47]

El dataset constaba de 10 carpetas, cada una correspondiente a un usuario. Dentro de cada carpeta se encontraba una subcarpeta y un archivo *.csv* llamado "annotation.csv". Este archivo contiene las actividades registradas junto con marcas de tiempo y en las subcarpetas se encontraban seis archivos *.csv* que contenían datos de las IMUs como mediciones del acelerómetro y giroscopio, con sus propias marcas de tiempo, en el nombre de estos archivos se especificaba la ubicación de las IMU. Para la depuración se realizó lo siguiente:

- Para el etiquetado de los archivos se implementó un código que tomó las marcas de tiempo compartidas entre los archivos y “annotation.csv” para extraer las etiquetas y poder asignarlas a los valores de los sensores en los archivos de IMU.

- Los archivos etiquetados se concatenaron asignando el código correspondiente a la ubicación, con lo cual se crearon 10 archivos *.csv* de cada usuario.

- Finalmente se concatenaron los 10 archivos *.csv* asignando el ID de usuario, con lo cual se obtuvo un único archivo *.csv*, al que se le realizó un último proceso donde se introdujo el ID de dataset, frecuencia y trial.

Gracias a esto se logró generar el archivo del dataset 11 denominado "Dataset_11.csv".

4.2.1.12 Daily And Sports Activities Dataset [43]

Los datos se presentaron en 19 carpetas, cada una representaba una actividad, dentro de estas se encontraban 8 subcarpetas para los sujetos. Cada subcarpeta de sujeto contenía 60 archivos *.txt* que segmentan las muestras y contenían datos sin procesar de 5 IMUs en diferentes posiciones del cuerpo. Para la depuración se realizó lo siguiente:

- Se combinaron los 60 archivos de cada sujeto, para procesar los datos, agregando una columna del número de sujeto, generando así 8 dataframes de cada sujeto dentro de las carpetas de actividades.

- Mediante la fusión de cada uno de los dataframes de cada sujeto, se obtuvieron 19 *.csv* para cada actividad, cada uno con el valor de la actividad correspondiente.

- Los 19 archivos *.csv* de cada actividad se concatenaron generando así un único *.csv*, al cual se le añadió el ID de dataset, frecuencia, código y trial.

Con lo anterior se consiguió un archivo *.csv* que se presenta como Dataset_12.csv.

4.2.1.13 WISDM Smartphone and Smartwatch Activity and Biometrics Dataset [39]

Los datos se presentaban en dos carpetas: "Phone" para los datos capturados por el teléfono celular y "Watch" para los datos del reloj inteligente. Cada carpeta tiene dos subcarpetas: una para los valores de acelerómetro y otra para los valores de giroscopio. Cada subcarpeta consta de 51 archivos *.txt* que representaban a los sujetos y proporcionaban información como el número de sujeto, actividad, marca de tiempo y los valores sin procesar del sensor. Para la depuración se realizó lo siguiente:

- Se fusionaron los 51 archivos dentro de las subcarpetas de acelerómetro y giroscopio asignando el ID de usuario, lo que resultó en dos dataframes.

- La combinación horizontal de los dataframes utilizó la marca de tiempo como referencia para unir filas con valores compartidos y rellenando con NaN donde corresponda.

- Finalmente se asignó frecuencia, ID del dataset y código dependiendo de si la fila tenía uno o ambos valores.

El procedimiento se aplica tanto a las carpetas "Phone" como "Watch". Debido a que son dispositivos diferentes y su frecuencia no es igual se optó por dejar el dataset separado, por lo que se presenta como: Dataset_13_Watch.csv y Dataset_13_Phone.csv.

4.2.1.14 Smartphone Dataset for Human Activity Recognition (HAR) in Ambient Assisted Living (AAL) Data Set [37]

La información disponible se centraba en los datos procesados, sin especificar el número de filas utilizadas en la extracción de características. Las etiquetas solo se asociaban con las muestras procesadas, en la documentación no se encontró información acerca de cuántas filas de los datos en bruto fueron tomadas para la extracción de características, se dividió los datos en bruto entre los procesados para obtener el número de filas, sin embargo, el cociente de la división entre datos procesados y en bruto no coincidía en un número entero, lo cual sugería que hubo un procesamiento adicional que no se mencionaba en la documentación. No se hizo una investigación adicional debido que, aunque se mencionaba la recopilación de datos de 30 sujetos, no se proporcionaba información para identificar a cada usuario. La falta de datos completos condujo a la decisión de no utilizar este dataset en la investigación.

2.4.1.15 Daily Motionless Activities a Dataset With Accelerometer, Magnetometer, Gyroscope, Environment, and GPS Data [27]

El dataset se presentaba en tres carpetas, cada una correspondiente a una actividad monitoreada específica. Dentro de cada carpeta, se encontraron subcarpetas que funcionaban como pruebas o experimentos individuales. Las subcarpetas contenían archivos *.txt* con mediciones de sensores, incluyendo magnetómetro, acelerómetro, giroscopio, sonido y ubicación; así como un archivo *.json* que proporcionaba información adicional, como detalles del usuario, la ubicación de la IMU y la frecuencia de muestreo. Para la depuración se realizó lo siguiente:

- Mediante un código se recorrieron todas las subcarpetas para combinar horizontalmente las mediciones del acelerómetro y el giroscopio, incorporando la información faltante del archivo .json, creando nuevas subcarpetas *.csv*.

- Se concatenaron estos archivos para crear archivos *.csv* correspondientes a las actividades, añadiendo una columna que identificaba la actividad correspondiente.

- Los tres archivos *.csv* resultantes se unieron para formar el *.csv* final.

Finalmente, al último archivo se le asignó el ID de dataset y frecuencia con lo que se obtuvó el "Dataset15.csv".

2.4.1.16 The ExtraSensory Dataset [28]

El dataset "The ExtraSensory" presentó grandes desafíos debido a la distribución de información y al tamaño de los datos registrados. La información se originó en dos dispositivos diferentes: una parte de los datos provenía del celular y la otra del reloj. Después de varios intentos fallidos de procesar el conjunto de datos en su totalidad, debido a su peso se llegó a la conclusión de que no se podía manejar como un único dataset. Esto llevó a la necesidad de dividirlo en seis partes, tres para el reloj y tres para el celular, debido a las limitaciones de recursos computacionales. Para la depuración se realizó lo siguiente:

- Se optó por aumentar el plan de Google Colab de básico a Pro y, finalmente, a Google Colab Pro+ debido a la disponibilidad de instancias de la familia NVIDIA, que proporcionaron la potencia de CPU, GPU y RAM necesarias para una manipulación eficiente de los datos. Sin Google Colab Pro+, trabajar con el dataset dividido en 6 partes habría sido imposible, ya que la instancia estándar se volvía inestable. Se procedió a seleccionar las carpetas "Labels", "raw_acc", "watch_acc", y "proc_gyro" en "The ExtraSensory Dataset". Los datos del celular se conformaron de los archivos "raw_acc" y "proc_gyro" relacionados con el acelerómetro y el giroscopio del celular.

- Los datos del reloj solo presentaban datos de acelerómetro en “watch_acc”. Cada archivo *.csv* contenía información sobre las actividades de los usuarios, y se representaban como una matriz de ceros y unos, donde los unos indicaban las actividades realizadas. Para trabajar con los datos del celular y el reloj, se crearon carpetas "Phone" y "Watch".

- Los archivos con el mismo nombre en "raw_acc" y "proc_gyro" se concatenaron en “Phone”. Sin embargo, debido a limitaciones de recursos, solo se pudieron agrupar 15 sujetos. Los archivos de medidas de sensores se encontraban en las carpetas de cada usuario en "Watch/Phone", y contenían información sobre el timestamp y las mediciones del sensor(es).

- Se relacionaron las carpetas del usuario con los archivos *.csv* de "Labels" utilizando el timestamp para asignar las actividades. Se seleccionaron dos columnas para describir una actividad debido a las limitaciones de recursos y que se consideró que con 2 actividades era suficiente para describir la acción principal del sujeto.

Los archivos *.dat* que estaban en las subcarpetas de usuarios de "Watch/Phone" se unieron después del proceso de etiquetado y también se asignó el ID de usuario y se convirtieron en archivos *.csv.* Finalmente, se agregó la información faltante, como el número de dataset, código y frecuencia, a los resultados finales se presentaron como “phone_1.dat”, “phone_2.dat”, “phone_3.dat”, "watch_1.dat", "watch_2.dat" y "watch_3.dat".

4.2.1.17 Framework for Simultaneous Indoor Localization, Mapping, and Human Activity Recognition in Ambient Assisted Living Scenarios [29]

El dataset estaba compuesto por 23 archivos *.mat*. Cada archivo representaba las mediciones registradas por un usuario e incluía marcas de tiempo y una etiqueta de actividad. Aunque los archivos también contenían información detallada sobre los beacons utilizados en el estudio, esta información se excluyó para centrarse exclusivamente en los datos de los sensores. Para la depuración se realizó lo siguiente:

- Se diseñó un código que permitió transformar los archivos *.mat* a archivos *.csv*, posterior a ello se asignó una columna de usuario y se fusionaron los archivos *.csv* en uno solo.

- Se agregaron datos como el código, el número de dataset, la frecuencia y el trial al dataset resultante.

Como resultado, el dataset se presenta ahora como "Dataset17.csv".

4.2.1.18 Simultaneous indoor pedestrian localization and house mapping based on inertial measurement unit and Bluetooth low-energy beacon data [30]

Cuando se realizó el análisis de la documentación presentada, se observó cierta discrepancia con el objetivo del negocio del presente trabajo de grado. Se contactó con el autor del dataset y se concluyó que este dataset, al estar enfocado en la reconstrucción de trayectoria, no sería adecuado para la clasificación de ADL. Por tanto, se excluyó de la unión.

4.2.1.19 Unicauca Dataset [31]

En la documentación se encontró que el dataset estaba enfocado en la reconstrucción de la trayectoria de diferentes usuarios en la Universidad del Cauca. Donde realizaban actividades tales como correr, caminar y trotar, pero con el enfoque de recrear una trayectoria. Por este motivo su inclusión en el dataset único era irrelevante.

4.2.1.20 The University of Dhaka Mobility Dataset (DU-MD / MD) [32]

Cada usuario tenía su propia carpeta en el dataset, que a su vez contenía subcarpetas para diferentes actividades. Dentro de estas subcarpetas se encontraban los valores de sensores distribuidos en 100 archivos *.txt*. Para la depuración se realizó lo siguiente:

- Se desarrolló un código personalizado para transformar estos archivos *.txt* en archivos *.csv*.

- Estos archivos *.csv* se combinaron en uno solo, incluyendo etiquetas de actividad asociadas a cada archivo *.csv* dentro de cada subcarpeta de actividad. Luego, se unieron todos los archivos de actividad por usuario en un único archivo *.csv*.

Como resultado, el dataset se presenta ahora como "Dataset20.csv".

4.2.2 Análisis de los datasets elegidos

En esta sección, se profundizó en el análisis de los datasets seleccionados.

4.2.2.1 Valores máximos, mínimos y valores nulos

.
Esta sección permitió un análisis que tuvo gran relevancia para el entendimiento preliminar de los datos. Se dividió en dos partes. Todo este análisis y gráficos se puede encontrar en el *repositorio de GitHub* Parte *1, 2 y 3 Preparacion_de_datos* del *Anexo G*, debido al número tan grande de todas las imágenes se decidió presentarlo como un anexo.

i) Las gráficas de pastel de los valores válidos y nulos permitieron entender la distribución de los valores NaN de las actividades y los sensores. Donde se encontraron datos sin etiquetas de actividades, así como registros sin mediciones de acelerómetro y/o giroscopio.

ii) Las gráficas de barras de los valores máximos y mínimos permitieron entender mejor los datos de los datasets debido a que fue un primer acercamiento a observar los valores máximos y mínimos de cada coordenada X, Y y Z del acelerómetro y giroscopio.

4.2.2.2 Resumen de datasets elegidos primera etapa

En el siguiente resumen se presentan los datasets elegidos que se tuvieron en cuenta para la construcción del dataset unificado con la información pertinente. A continuación, se presenta la Tabla 16 que hace referencia a los datasets antes de pasar por la preparación final para poder unificarlos.

Nombre Dataset	Nombre archivo	ID Dataset	Número filas	Número columnas	Peso del archivo	Acelerómetro	Giroscopio	Frecuencia (Hz)
Dataset 1	Dataset_1_part_1.csv	1	2.235.541	13	245,4 MB	g	No aplica	200
	Dataset_1_part_2.csv	2	817.571	13	93,9 MB	g	grados/segundo	20
Dataset 2	Dataset_2_part_1.csv	3	2.825.513	13	1,26 GB	m/s^2	No aplica	50
	Dataset_2_part_2.csv	4	435.207	13	322 MB	No aplica	rad/segundo	5
Dataset 3	Dataset_3.csv	5	2.811.490	13	380,4 MB	g	grados/segundo	100
Dataset 4	Dataset_4.csv	6	1.098.204	13	63,8 MB	m/s^2	No aplica	20
Dataset 5	Dataset_5.csv	7	8.174.859	13	1,29 GB	m/s^2	rad/segundo	100
Dataset 6	Dataset_6.csv	8	18.747.292	13	1,45 GB	g	grados/segundo	204.8
Dataset 8	Dataset_8.csv	9	3.411.500	13	482,6 MB	m/s^2	rad/segundo	50
Dataset 9	Dataset_9.csv	10	815.614	13	143,3 MB	g	rad/s	50
Dataset 11	Dataset_11.csv	11	1.224.408	13	85,2 MB	m/s^2	16 ADC	33
Dataset 12	Dataset_12.csv	12	5.700.000	13	630,5 MB	m/s^2	grados/segundo	25
Dataset 13	dataset13_watch.csv	13	499.980	13	603,7 MB	m/s^2	rad/segundo	20
	dataset13_phone.csv	14	5.746.549	13	63,4 MB	m/s^2	rad/segundo	20
Dataset 15	dataset15.csv	15	2.858.876	13	564,7 MB	m/s^2	rad/segundo	100
Dataset 16	phone1.dat	16	12.786.400	13	4,81 GB	g	rad/segundo	40
	phone2.dat	17	12.283.358	13	4,55 GB	g	rad/segundo	40
	phon3.dat	18	6.173.215	13	2,42 GB	g	rad/segundo	40
	watch1.dat	19	30.461.063	13	2,26 GB	mg	No aplica	25
	watch2.dat	20	24.384.237	13	1,84 GB	mg	No aplica	25
	watch3.dat	21	58.760.504	13	2,24 GB	mg	No aplica	25
Datastet 17	dataset17.csv	22	836.492	13	273,9 MB	g	grados/segundo	204.8
Dataset 20	dataset20.csv	23	556.063	13	16,6 MB	g	No aplica	30

Tabla 16. Resumen de datasets elegidos en su primera etapa.

Aunque se seleccionaron 15 datasets, por las subdivisiones que tuvieron algunos de estos, se tomaron en consideración 23 “datasets” para el tratamiento de la unión. El total del número de filas de datos de los 23 datasets elegidos y representados en la Tabla 16 es de 190,408,454, su peso es de 21.03 GB. La preparación de cada dataset se obtuvo en una nueva tabla con la información actualizada de filas y del peso. También se dejó el *Anexo D -*

Describe de los datasets elegidos, donde se muestra los resultados de la función *.describe()* de Python, a través de la biblioteca Pandas.

La función *.describe()* proporcionó un resumen conciso de las estadísticas clave de un conjunto de datos el cual incluye el recuento de valores no nulos, la media que muestra el valor promedio, la desviación estándar que mide la dispersión de los datos, el valor mínimo y máximo en cada columna numérica, y los cuartiles que indican cómo se distribuyen los datos [72]. Este conjunto de estadísticas permitió el análisis de datos para comprender la distribución y propiedades centrales de los datos, identificar valores atípicos y evaluar la calidad de cada uno de los datasets de la Tabla 16. Con esto se empezó la preparación de los datos en busca de obtener el dataset unificado en su versión “Raw Data”.

4.2.3 Técnica de remuestreo

Antes de hablar de la técnica de remuestreo (resampling), es importante entender por qué es necesario seleccionar una técnica para ello. La problemática con los datasets elegidos es que la mayoría tienen una frecuencia de recolección de datos diferente; en este punto se presenta un desequilibrio de datos, también conocido como desbalance de frecuencia [64].

El desbalance de frecuencia hace referencia a situaciones en las que hay una disparidad significativa en la cantidad de observaciones o eventos registrados en diferentes intervalos de tiempo o momentos temporales. [33] Recurrir al método de resampling se presenta como una solución fundamental. Al estandarizar la frecuencia de todos los datasets en una única, se obtuvo una ventaja clave: la eliminación de preocupaciones relacionadas con discrepancias temporales. Esto simplifica significativamente el proceso de segmentación y mejora la coherencia global de los datos seleccionados.

El remuestreo temporal es un método que implica la modificación de la frecuencia temporal de los datos mediante la adición o eliminación de registros en una clase minoritaria o mayoritaria, respectivamente. [34] Se implementaron dos técnicas de remuestreo conocidas como sobremuestreo (oversampling) y submuestreo (undersampling) [35]. Donde el sobremuestreo implica agregar registros a una clase minoritaria y el submuestreo implica eliminar registros de la clase mayoritaria.

Como se contempló en la Tabla 12, la frecuencia más común en los datasets es la de 50 Hz. Tras el proceso de limpieza y las subdivisiones necesarias de los datasets, como se puede apreciar en la Tabla 16, la moda de la frecuencia continúa siendo 50 Hz. Por lo tanto, se determinó 50 Hz como la frecuencia objetivo para todos los datasets. Para homogeneizar la frecuencia a 50 Hz en los conjuntos de datos seleccionados, se implementó el sobremuestreo en aquellos con una frecuencia menor a 50 Hz y el submuestreo en los conjuntos con una frecuencia mayor a 50 Hz.

Para verificar los datos resultantes después de aplicar los mencionados procesos, se corroboró la efectividad de estas técnicas comparando el *.describe()* antes y después del remuestreo, asegurando que las muestras mantengan valores como la mediana, la desviación estándar, el promedio iguales o similares para que no haya sesgo en los datos.

Lo anterior se puede encontrar más a detalle en el *repositorio de GitHub* Parte *1, 2 y 3 Preparacion_de_datos* del *Anexo G*. Con esto, se obtuvo conjuntos de datos uniformes en frecuencia, el siguiente paso fue tener las mismas unidades para el acelerómetro y para el giroscopio de los datasets.

4.2.4 Normalización de unidades

En cuanto a la normalización de unidades de los datasets, es fundamental garantizar la coherencia de las unidades en un conjunto de datos que aglutina múltiples conjuntos. Para lograr una comparación y análisis efectivos, es imperativo que todos los datos se encuentren en la misma unidad de medida. Esto no solo beneficia la uniformidad en la escala de los datos, sino que también potencia la precisión en el análisis. En el ámbito de algoritmos de aprendizaje automático y otras técnicas de análisis de datos, la suposición general es que los datos se presentan en unidades uniformes. La variación en las unidades podría introducir incertidumbre en los datos, lo que, a su vez, podría llevar a clasificaciones erróneas por parte de los modelos. Por lo anterior, se decidió que las unidades de los datasets convergen en m/s^2 y grados/segundo (o dps).

Al momento de realizar la conversión se tomó en cuenta que una gravedad (1 g) es equivalente a 9.81 m/s^2 para poder realizar las conversiones en los debidos datasets, así mismo se tomó que 1 rad = π/180°.

Un aspecto relevante por destacar fue que en el análisis de unidades de los datasets. Se observaron algunos valores incongruentes a lo que se esperaba según su documentación. Esto conllevo bastante tiempo, debido a que se planteaba al inicio la idea de que se había efectuado algo mal en la limpieza de los datos, no obstante, una vez verificado que el procedimiento realizado anteriormente estaba correcto, se procedió a buscar contingencias para estas anomalías en los siguientes datasets:

- Dataset 11
    - Riesgo: En el artículo del dataset 11 [47, 61], la falta de mención explícita de las unidades representó un riesgo significativo. La información disponible en el artículo se limitaba a la resolución de las unidades, indicando que el acelerómetro tenía una resolución de 0.1 mg y el giroscopio de 0.01 dps. Esto llevó a la suposición inicial de que las unidades eran 'mg' y 'dps'. Como primer acercamiento, se procedió a ajustar los valores del acelerómetro multiplicándose por 9.81/1000, mientras que los valores del giroscopio se mantuvieron sin cambios. Sin embargo, al analizar las estadísticas descriptivas, se observó que la media del acelerómetro era considerablemente alta, y los valores máximos y mínimos eran notablemente amplios, con una desviación estándar de 65. Esto planteaba dudas sobre la idoneidad de los ajustes realizados.
    - Contingencia: Ante la falta de información explícita sobre las unidades en el artículo del dataset 11, se implementó una medida de contingencia

consistente en la suposición inicial de que las unidades eran 'mg' y 'dps'. Sin embargo, al detectar discrepancias notables en las estadísticas en la función *.describe()*, especialmente en la elevada media y los valores máximos y mínimos extensos del acelerómetro, se reconoció la necesidad de revisar y corregir los supuestos sobre las unidades utilizadas en el análisis de los datos. En cuanto al giroscopio, los valores mínimos y máximos sobrepasaban +- 50000, con una desviación estándar de 4000 aproximadamente, por lo que también era un comportamiento bastante extraño. Así que, para el caso del acelerómetro se optó por dividir los datos entre 9.81, es decir que a los datos originales solo debemos dividirlos entre 1000, esto resultó en valores coherentes de acelerómetro, pero el problema continuaba con el giroscopio; sin embargo, después de numerosos intentos los valores se mantenían atípicos. Por consiguiente, se contactó a los autores del dataset y articulo para preguntarles sobre información del giroscopio, se recibió una respuesta bastante rápida donde uno de los autores comentó que los sensores guardaban los datos como 16 ADC, y proporcionó la fórmula para pasar los datos a dps, siendo 65535/500, aunque él mencionó que el valor '500' depende del rango, debido a que en el artículo se mencionan 4 rangos. En resumen, se realizó el cálculo con cada uno de estos valores y se seleccionó el resultado más coherente, siendo este el multiplicar por 65535/2000. Así se logró obtener el dataset con las unidades deseadas.

- Dataset 12

    - Riesgo: En el caso del dataset 12 [43, 63], la documentación carecía de claridad sobre las unidades utilizadas para el acelerómetro y el giroscopio, ya que el repositorio no proporcionaba esta información. Sin embargo, en el artículo publicado por los mismos autores, se mencionaban rangos de ±18g y ±5g para el acelerómetro, y ±1200 dps para el giroscopio. Inicialmente, se asumió que los datos estaban en g y dps, pero esta suposición resultó en resultados incongruentes. Al convertir los valores del acelerómetro de g a m/s^2, los valores resultaron ser excesivamente grandes, y en el caso del giroscopio, resultaron demasiado pequeños. A pesar de los intentos de contacto con los autores, solo uno respondió y remitió a su compañero de autoría, quien no respondió. Esta falta de claridad en las unidades representó un riesgo en el procesamiento de los datos.

    - Contingencia: Ante la falta de claridad en las unidades del dataset 12, se tomó la decisión de teorizar y ajustar los datos para obtener resultados congruentes. Para el acelerómetro, se dividió cada valor entre 9.81, asumiendo que los datos originalmente estaban en m/s^2. En el caso del giroscopio, se multiplicaron los valores por 180°/π para ajustarlos, considerando que los datos originalmente estaban en rad/s. Estos ajustes permitieron obtener resultados coherentes y, por lo tanto, se optó por mantenerlos como valores válidos para su posterior uso.

- Dataset 15

    - Riesgo: El Dataset 15 presentó un inconveniente significativo, ya que en el artículo [75] se mencionaba que la unidad del giroscopio era m/s^2, lo cual no es una unidad válida para un giroscopio. Esto planteaba interrogantes sobre el procesamiento de los valores del giroscopio, ya que se desconocía la forma en que habían sido modificados.
    - Contingencia: Ante la falta de claridad en las unidades del giroscopio en el Dataset 15, se tomó la medida de contactar a los autores. Uno de los autores confirmó que los datos del giroscopio estaban en rad/s. Con esta información, se procedió a realizar la conversión necesaria para expresar los datos del giroscopio en dps, asegurando la coherencia en las unidades utilizadas en el análisis de los datos.

- Dataset 16

    - Riesgo: En la información del dataset encontrada en el artículo [28] no se proporcionaba información clara sobre las unidades del giroscopio. Esto generó preocupación, ya que la carpeta que albergaba las mediciones del giroscopio se denominaba "proc_gyro", lo que inicialmente sugirió un procesamiento de los datos.

    - Contingencia: Para abordar este riesgo, se buscó contactar a uno de los autores, quien mencionó que las unidades podrían haber quedado en los valores por defecto de los sistemas operativos Android e iOS. Estos valores por defecto se identificaron como rad/s.

4.2.5 Depuración datos atípicos

Analizando las gráficas de violín que se encuentran en Github en el cuaderno Gráfica violín de nombre *Parte_4_1_Grafico_de_violin* del *Anexo G*, donde se encuentran las medidas del acelerómetro y giroscopio con los valores atípicos. En los resultados de la función *.describe()* de los datasets se evidencian valores extremos significativos, a pesar de que la desviación estándar no es alta. Se eliminaron los valores atípicos que probablemente eran ruido o resultado del proceso de calibración de los sensores.

Para cada dataset, se aplicaron las funciones mayores y menores que se pueden encontrar en el cuaderno *Parte_4_Union_de_los_datasets* del *Anexo G*. identificando los valores que excedían los rangos esperados. Como se abordó en la sección 4.2.3.

Con las soluciones mencionadas en esta sección, todos los datasets presentaron recuentos relativamente bajos de valores atípicos, lo que permitió eliminar las filas de manera efectiva. Como resultado, se obtuvieron valores máximos y mínimos más representativos, como se muestra en el nuevo gráfico de violín del cuaderno en *Github*.

Adicionalmente, se encontró la existencia de filas sin su respectiva etiqueta, especialmente en los datasets 11 (ID 11) y 16_phone (ID 16, 17 y 18), en donde la cantidad de filas sin etiquetas fue de aproximadamente la mitad. Como se puede apreciar en las gráficas del *repositorio de GitHub* Parte *1*,*2* y *3* *Preparacion_de_datos* del *Anexo G* incluso se tienen estos tipos de valores en la mayoría de datasets sin embargo en escalas diferentes.

En los primeros acercamientos del modelado y pruebas de algoritmos se concluyó que las filas sin etiquetas no aportan información relevante, así que se eliminaron este tipo de registros asegurando una mejor consistencia y homogeneidad de los datos.

4.2.6 ID de Actividades (etiquetado del dataset unificado)

Debido a las actividades repetidas en diferentes datasets fue necesario tener una diferenciación de las actividades por ID del Dataset y por ID de Actividad, esto quiere decir, que, por ejemplo, en todos los datasets "Caminar" pasó a tener el identificador número 6. Este procedimiento se puede apreciar en el *Anexo E* - *Actividades Finales*, donde primero se listan todas las actividades por orden de dataset, y se empiezan a enumerar las actividades con su propio identificador. En el momento de encontrar una actividad repetida, se coloca el mismo ID que se había elegido en una actividad de un dataset anterior. También las actividades que no tenían una información clara en su descripción (como por ejemplo "Activo" del dataset 2, que significaba que la persona estaba en movimiento sin especificar el tipo de actividad) o actividades que no fueran identificadas como ADL, en vez de un ID se colocó una "X" para su análisis y por último su posible eliminación. Finalizando este primer proceso que se encuentra en la pestaña "Actividades Datasets Elegidos" del *Anexo E,* donde se tenían 308 actividades, una vez se eliminaron las primeras actividades etiquetadas con una "X" quedaron 282, esta división de qué actividades fueron eliminadas y qué actividades siguieron en consideración se puede apreciar en la pestaña "*Eliminación X Actividades*", del mismo *Anexo E.*

En relación al mapeo de las actividades, el dataset 16 presentó una particularidad al tener dos o más etiquetas de actividades (Por ejemplo, actividades etiquetadas como Caminando + Usando El Celular). Para evitar posibles problemas de etiquetado múltiple en futuras clasificaciones, se generaron todas las combinaciones posibles a partir de las dos etiquetas de actividades y se creó una actividad única basada en las combinaciones, este procedimiento se muestra en la pestaña "Actividades 16" del *Anexo E.*

Como resultado de este proceso, el número total de actividades se redujo considerablemente, con un total de 92 actividades en lugar de 282. Esto se detalla en el *Anexo E.*. Es importante destacar que, aunque siguen siendo numerosas, se consideró que es una cantidad adecuada para la creación del dataset unificado.

4.2.7 Resumen de datasets elegidos (segunda etapa)

En esta segunda etapa ya se tienen los datasets preparados para la unión. En la preparación de datos de cada dataset, cada dataset pasó a tener el mismo número de columnas. También la frecuencia de todos los datasets se registra en una frecuencia única de 50 Hz. A continuación, se presenta la segunda etapa del resumen de los datasets ya depurados.

Nombre Dataset	Nombre archivo	ID Dataset	Número filas	Peso del archivo
Dataset 1	dataset1_1.csv	1	553.588	52,3 MB
	dataset1_2.csv	2	2.003.738	230,1 MB
Dataset 2	dataset2_1.csv	3	2.825.513	283,1 MB
	dataset2_2.csv	4	4.350.481	422,3 MB
Dataset 3	dataset3.csv	5	1.404.082	173,6 MB
Dataset 4	dataset4.csv	6	2.743.272	111,9 MB
Dataset 5	dataset5.csv	7	4.087.429	382 MB
Dataset 6	dataset6.csv	8	4.545.903	330,8 MB
Dataset 8	dataset8.csv	9	3.411.500	472,7 MB
Dataset 9	dataset9.csv	10	815.614	103,1 MB
Dataset 11	dataset11.csv	11	1.855.118	220,3 MB
Dataset 12	dataset12.csv	12	11.400.000	1,1 GB
Dataset 13	dataset13_1.csv	13	1.249.948	1,48 GB
	dataset13_2.csv	14	14.366.360	158,8 MB
Dataset 15	dataset15.csv	15	1.429.426	128,7 MB
Dataset 16	dataset16_phone_1.csv	16	15.777.000	2,34 GB
	dataset16_phone_2.csv	17	15.354.197	2,28 GB
	dataset16_phone_3.csv	18	7.716.518	1,16 GB
	dataset16_watch_1.csv	19	60.404.428	4,24 GB
	dataset16_watch_2.csv	20	48.070.476	3,41 GB
	dataset16_watch_3.csv	21	29.747.900	4,14 GB
Dataset 17	dataset17.csv	22	204.221	28,4 MB
Dataset 20	dataset20.csv	23	556.063	19,4 MB

Tabla 17. Resumen de datasets elegidos en su segunda etapa.

4.3 Unión del dataset

Para la unión de todos los datasets se realizó un proceso de concatenación vertical obteniendo así un dataset de 263.884.250 filas y 13 columnas, con un peso de 23,54 GB el cual se puede encontrar en el *Anexo J. Datasets finales, en Kaggle.* Bajo el nombre de Dataset_Unico.csv.

Desafortunadamente, por ser un dataset de un gran tamaño no fue posible procesarlo, incluso no fue posible cargarlo en memoria usando la librería de pandas pd.read_csv(). Con esto en mente se inició un análisis con el dataset 16, debido a que era el dataset que ocupaba el mayor porcentaje en el dataset único, especialmente sus medidas de reloj inteligente.

El dataset 16 tiene una peculiaridad y es que sus valores provienen de un ambiente no controlado, es decir, cada usuario podía registrar la actividad que estaba realizando mediante un aplicativo móvil. Por esa razón, en la etapa de modelado, en los primeros acercamientos

con los algoritmos de clasificación, se realizaron pruebas con ese dataset, las cuales no dieron resultados positivos. Esto generó una falta de confiabilidad en este dataset, por lo que se efectuó un plan de contingencia de disminuir la cantidad de usuarios a la mitad. Como resultado se obtuvo un dataset unificado de 183.694.00 filas.

A pesar de lo descrito anteriormente, se continuaba presentando inconvenientes con la capacidad de procesamiento del dataset unificado. En este punto se realizó un análisis más robusto de las actividades, llegando a la conclusión de eliminar actividades no registradas como se detalla la sección de ID Actividades. Además, este análisis resultó en la identificación de actividades que podían fusionarse para simplificar el dataset y hacerlo más conciso. Por ejemplo, actividades como "trotar" y "correr", o "saltar levemente" y "saltar con una cuerda" se combinaron en una sola categoría, lo que contribuyó a reducir la cantidad de actividades en el dataset. Este proceso resultó en un dataset final mucho más consistente y abordó eficazmente el problema de procesamiento mencionado anteriormente.

Este proceso realizado se puede apreciar en el *Anexo E*, donde se listan todas las actividades, se eliminan y fusionan las actividades, resultando un total de 61 actividades que se exponen en el dataset final.

Finalmente, al agrupar estas actividades, se contó con un dataset de 173.241.093 filas, sin embargo, en busca de tener una mejora en la calidad de los datos y en el entrenamiento de los algoritmos así mismo como disminuir la exigencia computacional, se identificó que es posible prescindir de algunas ubicaciones de los sensores. Esto debido a que el objetivo de esta primera versión de sistema móvil es clasificar una actividad en base a data de actividades de manera aleatoria, por ende, se optó por eliminar la ubicación de los códigos con terminación 0, 3, 4, 5, 8 y 9, que hacen referencia a: sin información, pie, pecho, espalda, cintura y otros. Manteniendo las ubicaciones en la muñeca, cadera, brazo y pierna, que son las ubicaciones donde es más probable que se encuentre un celular.

Finalizando, se obtuvo un dataset de 127.766.944 filas con 56 actividades y un peso de 11,88 GB, lo que permitió su procesamiento. El nombre del archivo de esta unión es: datasetFinalGeneral_less_users_activities_and_location.csv, que se puede encontrar en: *Anexo J. Datasets finales, en Kaggle* bajo el nombre de Dataset_Unico_Depurado.csv.

4.4 Normalización del dataset

4.4.1 Normalización de rangos

El dataset presentado en la sección anterior presenta un inconveniente, y es que como se presentó en la gráfica de violín del cuaderno *Parte_4_1_Grafico_de_Violin* del *Anexo G*, los rangos de las medidas de acelerómetro y giroscopio en los diferentes datasets fue desigual, por lo que se planteó realizar un proceso de normalización Con estos sesgos de información evitados, se obtuvieron datos más imparciales y representativos, simulando que los datos provienen de una sola fuente o conjunto de datos uniforme. [36]

El tipo de normalización que se seleccionó se conoce como Z-score, la cual es un tipo de técnica que busca transformar los datos de tal manera que el promedio sea cero y la desviación estándar uno, para lograrlo se hace uso de la fórmula: [36]

$$Z = \frac{x - \mu}{\sigma}$$

Ecuación 1. Normalización Z-Score

Donde:

X: Variable a transformar.
μ: Valor promedio de todas las muestras de la variable a transformar.
σ: Desviación estándar de todas las muestras de la variable a transformar.

Finalizando el proceso de normalización se obtuvo un dataset llamado: DatasetFinalNormalized.csv el cual cuenta con un peso de 14,04 GB.

4.5 Estructurar, integrar y formatear los datos

En proyectos de ciencia de datos es importante realizar un proceso de estructuración de los datos, lo cual es un proceso bastante común en los problemas de clasificación (en el capítulo 5 se aborda a más detalle esta afirmación). En los artículos de los datasets elegidos se evidenció la necesidad de tratar los datos con extracción de características en lugar de hacerlo con los datos en crudo. Esto se debe en gran medida a que ésta técnica reduce los datos redundantes, incrementa la velocidad de aprendizaje y es más eficiente en el uso de recursos computacionales. Adicionalmente, en muchos casos mejora la precisión de los modelos. [76]

Esta técnica consiste en entrenar los modelos con características de los datos en crudo basándose en medidas como la media, mediana, desviación estándar, entre otras. A continuación, se presenta el proceso para la transformación de los datos.

4.5.1 Segmentación del dataset

Las características se calcularon para cada segmento; estos segmentos deben ser uniformes y coherentes para que así mismo lo sean las características. En notación, para los autores de este trabajo fue necesario acordar el tiempo en el cual las personas podrían terminar de

realizar una determinada actividad sin ningún inconveniente. Se determinó tener un tiempo de 10 segundos en base a lo encontrado en los diferentes proyectos de ciencia de datos aplicados en ADLs. A continuación, los pasos a tener en cuenta en el proceso de la segmentación:

- PASO 1.

  El dataset cuenta con una frecuencia de 50 Hz. La segmentación se realizó con una toma de 500 filas (correspondientes a 10 segundos de datos), sin embargo, se evidenció que por la naturaleza de la organización de las actividades del dataset único se deja en consideración que algunos segmentos podrían describir más de una actividad

- PASO 2.

  Para tener una contingencia de lo mencionado anteriormente, se ordenó el dataset por actividad (en lugar de ordenarlo por dataset), y posterior a ello se eliminaron las filas de cada actividad que le impedían ser un múltiplo de 500, con este cambio, se aseguró que al fragmentar los segmentos solo tengan alusión a una determinada actividad. Sin embargo, algunos segmentos contaban con información combinada de datasets.

- PASO 3.

  Se desarrolló un código para la identificación de los segmentos con información combinada de dataset, y se tenían un total de 16.516 segmentos, que fueron eliminados posteriormente.

Una de las consecuencias negativas que trajo consigo la eliminación de los 16.516 segmentos fue la pérdida de 2 datasets (ID's 3,15). Se analizó qué tan significativa era la ausencia de estos 2 datasets y se concluyó lo siguiente:

- El dataset 2 parte 2 (ID 3), era etiquetada por los usuarios, así que se tenía poca confiabilidad de sus valores.

- El dataset 15 (ID 15), era etiquetada también por los usuarios, así que se tenía poca confiabilidad de sus valores.

Gracias a esto se pudo asegurar la confiabilidad de la segmentación en el dataset único. Finalmente, se decidió conservar los 238.990 segmentos que representan 119.495.000 de filas.

4.5.2 Extracción de características

En esta sección se abordará el proceso de extracción de características, la cual se divide en dos partes: la elección de características, donde se explicará cómo se seleccionaron las características y, la aplicación y unión de la extracción de características, en donde se detalla cómo se aplicaron estas características seleccionadas a los segmentos de los datasets.

4.5.2.1 Elección de características

Se seleccionaron características que facilitaron la descripción y diferencia de las actividades. Una buena selección de características es fundamental teniendo en cuenta el entrenamiento que se realizó en los siguientes capítulos. Se buscó no tener redundancia entre las características, asegurando que éstas aporten información relevante de cada actividad. Las características seleccionadas son:

- Entropía: La entropía basada en la teoría de la información de Shannon [59] es una medida de la incertidumbre de una señal. En el contexto de clasificación de actividades como correr, caminar, estar de pie, entre otros, que tienen diferentes patrones de movimiento, estos patrones se ven reflejados en la entropía.

- Promedio: Es una característica básica que proporciona información sobre el valor medio de las medidas en el respectivo segmento

- Desviación estándar: Proporciona información sobre la variabilidad de las medidas en el respectivo segmento, una alta desviación estándar puede expresar movimientos bruscos o variados.

- Máximo: Ayuda a la representación de movimientos de intensa actividad física.

- Mínimo: Permite representar actividades de reposo.

- Media: Proporciona información sobre la tendencia central en los segmentos.

- Promedio absoluto: Mide la distancia promedio de los valores con respecto a su media, lo cual puede ser útil para cuantificar la amplitud de los movimientos.

- Promedio resultante: Es una característica útil para describir la actividad en todas las direcciones (ejes X, Y y Z), se calcula con la Ecuación 4.

- Magnitud: La magnitud de la señal puede proporcionar información sobre la intensidad de las actividades, lo que puede ayudar a identificar actividades que requieren más intensidad física de las que no. Se calcula con la Ecuación 5.

$$H(X) = \sum_{i=0}^{n} P(xi).(P(xi)$$

Ecuación 2. Entropía.

$$MA(X) = \frac{1}{n}\sum_{i=0}^{n} |xi|$$

Ecuación 3. Promedio absoluto.

$$RM = \frac{1}{n}\sum_{i=0}^{n}\sqrt{(xi^2 + yi^2 + zi^2)}$$

Ecuación 4. Promedio resultante.

$$M = \sum_{i=0}^{n}\sqrt{(xi^2 + yi^2 + zi^2)}$$

Ecuación 5. Magnitud de la señal.

Donde:

xi, yi, zi: Es el valor que toma la variable en la instancia i.
i: En el contador para recorrer el segmento, se inicializa en 0.
n:Es la cantidad máxima de muestras, en este caso este valor es de 499
P(xi): Es la probabilidad de encontrar el valor de x en la instancia y en todo el segmento.

4.5.2.2 Aplicación y unión de extracción de características

Con las características definidas, se procedió a realizar el cálculo a cada uno de los segmentos. Este proceso se puede encontrar en el libro de colab *Parte_5_Normalizacion_segmentacion_y_extraccion_de_caracteristicas.* del *Anexo G*.

Se crearon diferentes dataframes para cada característica, los cuales se unieron horizontalmente, conformando así el dataset final, que cuenta con 238.990 filas, 50 columnas y 56 actividades. Se presenta como *Dataset_Final_Caracteristicas.csv*. Tiene un peso de 135,3 MB, con esto se puede apreciar la diferencia de tamaño comparándolo con el dataset antes de extraer características, lo cual fue de gran beneficio para la eficiencia computacional, el tiempo de entrenamiento de los algoritmos y el peso de los modelos.

Para el entrenamiento de los modelos es preferible que el código de la columna de “Code” que en ese momento se encontraba unido. Se separó en 3 columnas diferentes para facilitar el entrenamiento, y la comprensión del mismo. Dejando en claro esto, se pasó de tener la columna ‘Code’ a 'Sensor_Type', 'Left_Right' y 'Ubication'. Este dataset es el que se usó para entrenamiento y prueba de algoritmos, por lo que se lo denominó Dataset definitivo, que se presenta como Dataset_Final_Codigo_Separado.csv con un peso de 135,7 MB.

4.6 Descripción del dataset transformado

En este momento el dataset denominado dataset definitivo cuenta con 238.990 filas, 56 actividades y 52 columnas conformadas por: Segmento, Dataset, Activity, Sensor_Type, Left_Right, Ubication y las 46 columnas restantes son las características definidas anteriormente. Sin embargo, para el entrenamiento del modelo, se eliminó la columna *segmento* y se transformaron los datos NaN a ceros, esto se realizó debido a que los modelos no reciben este tipo de *data*, sin embargo, no se quería eliminar la información que proporcionaban estas filas, pero tampoco se quería afectar el rendimiento por lo que se

decidió en que sean cero. La columna Dataset se mantuvo para que sea una variable para utilizar más adelante con el fin de realizar una distribución uniforme en los datasets de prueba y entrenamiento, una vez realizada esta división se despreciaba esta columna.

Dejando en claro lo anterior, se tiene el dataset listo para el entrenamiento de algoritmos, a continuación, se presenta el gráfico de distribución de datasets en el dataset definitivo.

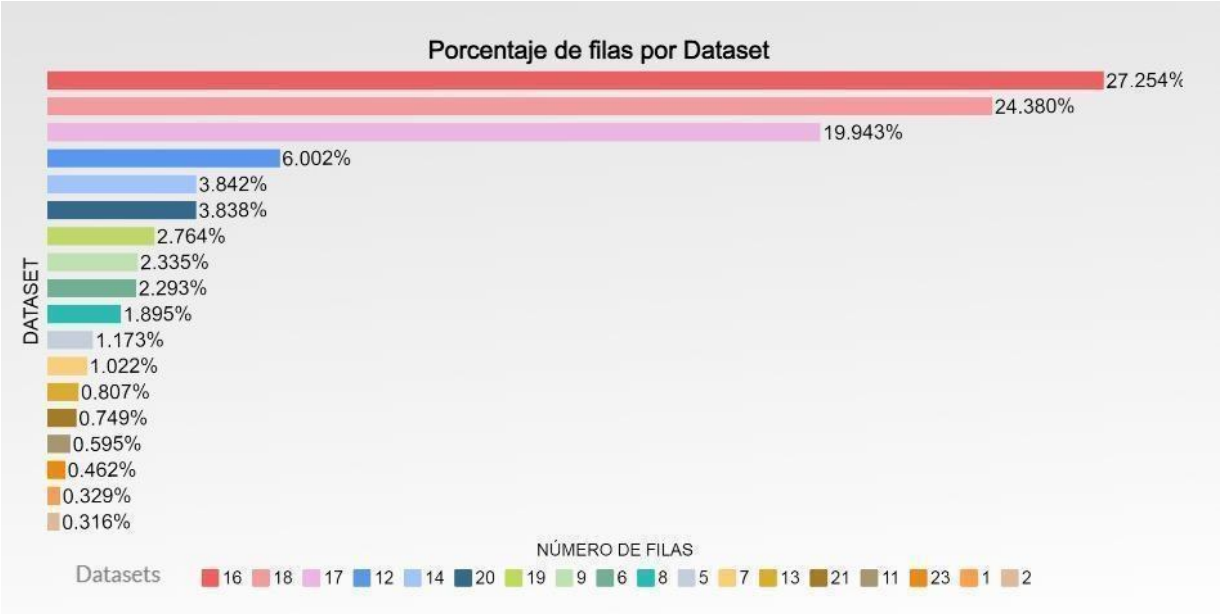

Imagen 3. Porcentaje de filas por dataset en el dataset definitivo.

Como se puede ver en la Imagen 3, el dataset 16 (ID's 16,17,18,19,20,21) representa un total de 78.928%, lo cual es un porcentaje llamativo en el dataset definitivo, todo esto pasará a ser evaluado en los primeros acercamientos del siguiente capítulo.

Capítulo 5

Fase 4: Modelado

En esta fase de CRISP-DM se seleccionan las técnicas de modelado adecuadas para el proyecto, se obtienen los modelos y finalmente se evalúan. Desde la fase 2 del proyecto se eligieron las bibliotecas para ciencia de datos de scikit-learn y scikit-multiflow para Batch Learning (BL) y Stream Learning (SL) respectivamente para realizar el modelado en base al dataset definitivo que se obtuvo en la fase anterior que está configurado específicamente para cumplir con ese propósito. Las tareas propias de la fase de modelado están descritas a continuación.

- **Seleccionar la técnica de modelado.**
    - Seleccionar los algoritmos de BL en la librería scikit-learn.
    - Seleccionar los algoritmos de SL en la librería scikit-multiflow.
    - Acercamiento preliminar de los modelos.
    - Descripción de los nuevos datasets derivados del dataset definitivo.
- **Generación plan de prueba.**
    - Definición de modelos.
    - Plan de prueba.
- **Construir los modelos.**
- **Evaluar los modelos.**
    - Evaluación de modelos preliminares sin valores aleatorios (MPSVA).
    - Evaluación de modelos preliminares con valores aleatorios (MPCVA).
    - Evaluación de modelos depurados.
- **Análisis de los modelos: Primeras diferencias encontradas en el rendimiento de los modelos generados en BL y SL.**

5.1 Seleccionar la técnica de modelado

Las técnicas de modelado se refieren a las estrategias y algoritmos utilizados para crear modelos a partir de datos con el objetivo de hacer clasificaciones, predicciones o tomar decisiones basadas en esos datos. En el contexto de aprendizaje automático y ciencia de datos, las técnicas de modelado incluyen una variedad de algoritmos y métodos que son aplicados a diferentes tipos de datos y problemas.

De acuerdo con el propósito del segundo objetivo específico de *"realizar una comparación del rendimiento de algoritmos basados en Batch Learning frente a algoritmos de Stream Learning utilizando uno o más conjuntos de datos seleccionados"*, y siguiendo las pautas establecidas en esta fase, la principal diferencia entre BL y SL reside en que el modelo BL procesa datos en lotes completos para entrenar modelos, mientras que el modelo SL se enfoca en una pequeña fracción de datos para su entrenamiento mediante un proceso incremental, adaptándose a medida que llegan nuevos datos. [77]

5.1.1 Seleccionar los algoritmos de BL en la librería scikit-learn

Scikit-learn ofrece una extensa gama de herramientas y algoritmos para tareas de clasificación, regresión, agrupación y más. Es una opción popular para desarrolladores y científicos de datos debido a su facilidad de uso y eficiencia. [73] A continuación algunas de las tareas de aprendizaje que tiene Scikit-Learn en la construcción de los modelos.

- Clasificación: Incluye algoritmos como Random Forest (RF), k-Nearest Neighbors
- (k-NN), Support Vector Machine (SVM) y muchos otros para problemas de clasificación binaria y multiclase.
- Aprendizaje no supervisado: Ofrece algoritmos de clustering como k-Means y técnicas de reducción de dimensionalidad como Principal Component Analysis (PCA).
- Regresión lineal: Utilizado para problemas de regresión donde usualmente se busca predecir valores numéricos continuos.

5.1.2 Seleccionar los algoritmos de SL en la librería scikit-multiflow

Scikit-multiflow está diseñada para abordar problemas de aprendizaje automático en flujos de datos, ofrece algoritmos que se adaptan continuamente a la llegada de nuevos datos, lo que lo hace adecuado para aplicaciones como el procesamiento de datos de sensores y análisis de flujos de datos en constante evolución [78]. A continuación, se presentan las tareas de aprendizaje más comunes que tiene scikit-multiflow en la construcción de sus modelos:

- Clasificación de flujo: Incluye algoritmos como Adaptative Random Forest (ARF), k-Nearest Neighbors (k-NN), etc.
- Regresión de flujo: Utilizado para problemas de regresión donde se busca predecir valores numéricos continuos.
- Agrupamiento de flujo: Puede ser útil para identificar patrones emergentes o cambios en grupos de datos en constante evolución, incluye algoritmos como CluStream.

Considerando lo expuesto, se optó por la elección de las tareas de clasificación proporcionadas por scikit-learn y scikit-multiflow. Esta selección permitió comparar directamente los algoritmos de ambos entornos, asegurando una evaluación que cumple con el objetivo principal de identificar el modelo más adecuado en función de los criterios de éxito definidos en la sección 2.3.2.

5.1.3 Algoritmos a tratar para BL y SL

La falta de documentación detallada sobre SL en general se convirtió en una limitante, ya que a pesar de que los algoritmos elegidos estaban disponibles en scikit-multiflow, la escasez de ejemplos y detalles en su documentación significó una problemática. Para esto se contactó con desarrolladores experimentados en SL, que nutren los foros de esta librería, por medio de un canal de Telegram, para su asesoría.

A continuación, se listan con una breve descripción los algoritmos que se trataron en esta creación del modelo.

En síntesis, para todos los algoritmos se usaron los hiper parámetros por defecto exceptuando KNN de Scikit Multiflow.

Batch Learning		
Nombre algoritmo	**Descripción**	**Hiper Parámetros**
DecisionTreeClassifier	Divide datos en ramas, captura patrones. Puede sobreajustar.	*, criterion='gini', splitter='best', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf=0.0, max_features=None, random_state=None)
RandomForestClassifier	Conjunto de árboles, robusto, evita sobreajuste.	-P 100 -I 100 -num-slots 1 -K 0 -M 1.0 -V 0.001 -S 1
NaiveBayes	Simple, supone independencia entre variables. Bueno con datos categóricos.	- -A 0.5
KNN	Clasifica según vecinos cercanos. Sensible a datos ruidosos.	N = 5, leaf_sizeint, default=30
LogisticRegression	Modela probabilidad, lineal. Limitado para datos no lineales.	-R 1.0E-8 -M -1 -num-decimal-places 4
GaussianNB	Naive Bayes con distribución gaussiana. Efectivo con distribuciones normales.	**, priors=None, var_smoothing=1e-09)*
SVM	Separa clases con hiperplanos. Eficaz en datos lineales y no lineales.	Parámetros por defecto
Stream Learning		
Nombre algoritmo	**Descripción**	**Hiper Parámetros**
HoeffdingTreeClassifier	Árbol adaptativo, cambios incrementales. Baja memoria, velocidad.	max_byte_size=33554432, memory_estimate_period=1000000, grace_period=200, split_criterion='info_gain', split_confidence=1e-07, tie_threshold=0.05, binary_split=False)
AdaptiveRandomForestClassifier	Evoluciona con datos. Alto rendimiento en flujo continuo.	n_models=10, max_features="sqrt", lambda_value=6, metric=Accuracy: 0.00%, disable_weighted_vote=False, drift_detector=ADWIN
NaiveBayes	Sencilla estimación de probabilidad. Puede no adaptarse a cambios.	Parámetros por defecto
KNN	Clasifica con vecinos cercanos. Sensible a cambios bruscos.	N = 5, max_window_size=1000, leaf_size=30,

Tabla 18. Algoritmos BL y SL.

5.1.4 Acercamiento preliminar de los modelos

Las pruebas de rendimiento de los algoritmos de BL y SL con el dataset definitivo se pueden apreciar en el *Anexo G* - *Repositorio de Github*, del libro "*Parte_6_Prueba_de_algoritmos*". Los resultados no cumplieron exactamente con los criterios de éxito que se esperaban, no obstante, se generó la inquietud ¿qué tan confiable son los datos que provienen del dataset 16?, debido a que este dataset ocupa el 78.928% del dataset definitivo. Por tanto, se decidió generar 2 datasets a partir del dataset definitivo para enriquecer el análisis, teniendo como resultado 3 datasets a analizar.

- El primero fue el dataset definitivo sin ninguna alteración, donde los modelos generados y/o probados con este pasaron a tener el nombre de Modelo Completo. Este dataset se mantuvo con el nombre de "Dataset Definitivo".

- El segundo fue el dataset definitivo retirando al dataset 16 (ID's 16, 17, 18, 19, 20 y 21) lo que permitió evaluar el rendimiento del modelo sin este conjunto de datos. Cabe mencionar que aun así, en este segundo diagnóstico el dataset permaneció de un tamaño y número de filas considerablemente grande debido a que ronda en el marco de las cincuenta mil (50.000) filas aproximadamente, lo que sigue siendo considerablemente un número bastante alto, dado que según el análisis realizado en este trabajo de grado el número de filas no superan los diez mil (10.000). Este dataset fue denominado *"Dataset Sin 16"*.

- El tercero fue utilizando únicamente el dataset 16, con el fin de apreciar el comportamiento de este dataset individualmente. Este dataset fue denominado *"Dataset solo 16"*.

5.1.4.1 Descripción de los nuevos datasets derivados del dataset definitivo

A continuación, se describe cómo están conformados los dos nuevos datasets.

5.1.4.1.1 Dataset sin 16

Cuenta con 50.360 filas y un peso de 39,7 MB. El dataset describe 42 actividades, este dataset se puede encontrar en el *Anexo J*, con el nombre Dataset_Sin_16.csv

- Manteniendo las columnas de segmento, Dataset y las filas NaN se presenta como: Dataset_Final_para_batch_solo_16.csv.

- Eliminando las columnas de segmento y Dataset y transformando los valores NaN a ceros se presenta como: Dataset_Final_para_stream_sin_16.csv.

5.1.4.1.2 Dataset solo 16

Cuenta con 188.630 filas y un peso de 103 MB. El dataset describe 30 actividades, este dataset se puede encontrar en el *Anexo J*, con el nombre Dataset_Solo_16.csv

- Manteniendo las columnas de segmento, Dataset y las filas NaN se presenta como: Dataset_Final_para_batch_solo_16.csv.

- Eliminando las columnas de segmento y Dataset y transformando los valores NaN a ceros se presenta como: Dataset_Final_para_stream_solo_16.csv.

5.2 Generación plan de prueba

El objetivo de esta tarea fue instaurar cómo se conformó el plan de pruebas para la creación y evaluación de los modelos de BL y SL.

5.2.1 Definición de modelos

Con el fin de cumplir de la mejor manera el objetivo específico número dos, que involucra una comparación entre BL y SL, se crearon dos tipos de modelos, los modelos preliminares y los modelos depurados. A continuación, la explicación y conformación de los modelos preliminares y modelos depurados:

5.2.1.1 Modelos preliminares

Los modelos preliminares se refieren a los modelos creados en el trabajo de grado haciendo uso de algoritmos de BL y SL en los tres datasets definitivos previamente. La principal característica de los modelos preliminares era que tenían todas las actividades. Teniendo en cuenta la finalidad de obtener una excelente comparación entre ambos modelos, se crearon los modelos preliminares sin valores aleatorios (MPSVA) y los modelos preliminares con valores aleatorios (MPCVA). A continuación, las consideraciones tenidas en cuenta en esta etapa inicial de los modelos preliminares:

- Debido a la falta de familiaridad con algoritmos de SL, se realizó un análisis inicial (propio de los autores del presente trabajo de grado), esto con el fin de tener mayor claridad en la diferenciación entre SL y BL. Se desarrolló un código que permitió replicar el proceso de entrenamiento y evaluación de BL en SL, replicando la naturaleza de entrenamiento y evaluación de BL en SL. Esto otorgó una mayor claridad del comportamiento de ambos tipos de algoritmos. Es importante recalcar que este primer análisis fue completamente para el entendimiento y futuras tomas de decisiones así mismo se puede ver a mayor profundidad en los modelos MPSVA.

- Una vez se tuvo claridad del comportamiento de los algoritmos se planteó la pregunta ¿cómo asegurar la imparcialidad de los modelos de SL y BL? La imparcialidad es un objetivo importante de cualquier tipo de modelo de ML, ya que garantiza que estos sistemas tomen decisiones justas y equitativas, ya que, con esto, no exhibe sesgos. Para lograr esto, se planteó un modelo que permitiera este tipo de análisis. Esto se puede ver a detalle en los modelos MPCVA.

A continuación, se explican a detalle los tipos de modelos MPSVA y MPCVA y sus respectivas consideraciones.

i) Modelos preliminares sin valores aleatorios (MPSVA)

Cuando se planteó el capítulo 5, no se tenía un entendimiento preciso del comportamiento de los algoritmos. Por ello se realizaron varias pruebas iniciales con el fin de tener la mejor claridad de los conceptos, haciendo uso de la documentación y recomendaciones de foros, se comenzó un “ensayo y error”. Esto dio origen a los MPSVA.

Los MPSVA, son los modelos entrenados de la manera más “ideal” posible. En el caso de este proyecto, se consideró “ideal” a la falta de modificación de los 3 datasets. Por esto, los datasets para el entrenamiento y evaluación estaban completamente ordenados por actividad, id del dataset, usuario y código, cada muestra da certeza y son valores de mucha confiabilidad (al menos por su orden).

En este punto se buscó que SL se comportara igual que BL a nivel de entrenamiento y evaluación. Hay que aclarar que esta evaluación fue propuesta y realizada por los autores del presente trabajo de grado. La naturaleza de BL es entrenarse con un fragmento del dataset y después evaluarse con el otro fragmento. Mientras que SL se entrena y se evalúa al mismo tiempo en todo el conjunto de datos haciendo uso del aprendizaje incremental de SL. Se tuvieron en cuenta las siguientes consideraciones:

- Para el entrenamiento en BL se entrenó con el 70% y se evaluó con el 30% del dataset como es común. Mientras que para SL se entrenó y evaluó con el 70%, con este resultado se creó un modelo .pkl, el cual se utilizó para volver a entrenar y evaluar con 30% del dataset restante haciendo un paralelismo con la naturaleza de BL.

- Para la división del dataset se hizo uso de la función StratifiedShuffleSplit de scikit-learn debido a que permitió dividir el dataset de manera equitativa, proceso necesario por la diferencia de tamaño de los datasets y actividades que conforman el dataset definitivo. Se aseguró que el entrenamiento tuviese un porcentaje del 70% de cada dataset y actividad, de la misma manera para la evaluación.

- Lo anterior aseguró que al momento del entrenamiento no se perdiera información de ningún dataset ni actividad, con el fin de que el modelo conociera todos los datos a clasificar en el momento de la evaluación.

- Es importante aclarar que la función de aprendizaje incremental de SL consiste en hacer uso de la función .partial_fit() de scikit-learn para retroalimentar el modelo conforme a los nuevos datos del flujo que van llegando.

  Lo mencionado anteriormente se abordó más adelante, específicamente en la sección *5.4. Evaluar los modelos*, donde se presentan resultados, análisis y conclusiones más detalladas de los modelos obtenidos.

ii) Modelos preliminares con valores aleatorios (MPCVA)

Para los MPCVA se tuvo mayor entendimiento respecto a los algoritmos que favorecieron la realización de estos modelos tanto en eficiencia como en tiempo. Los modelos MPCVA tuvieron como objetivo entrenarse y evaluarse en un escenario “poco favorable” para asegurar la imparcialidad de los modelos. Es por esto que se aleatorizaron todos los valores de los 3 datasets en el entrenamiento y evaluación. Para esto se tuvieron las siguientes consideraciones:

- En SL se aleatorizó el dataset, se evaluó y entreno de la manera convencional, es decir se tomó todo el conjunto de datos sabiendo que esto tendría un impacto significativo debido a la naturaleza del aprendizaje incremental de SL. Lo anterior proporcionó resultados más imparciales puesto que no se está limitando al algoritmo a un escenario de datos ordenados sino más bien aproximado a un entorno “real”. Se consideró un entorno “real” al escenario donde se tuvo una aleatorización de todos los datos, asegurando que el modelo entrenado va a ser lo más imparcial posible.

- En BL también se aleatorizó el dataset, sin embargo, mediante pruebas se supo que esto no tiene impacto alguno en BL debido a su naturaleza de tomar todo un lote para su entrenamiento, por esto se planteó el uso de la técnica de validación cruzada (cross-validation), la cual divide en conjunto de datos en subconjuntos y se entrena y evalúa en estos, aclarando que esta vez no se dividieron los datos de manera equitativa. Así, se obtuvieron resultados de manera más precisa sobre el rendimiento de los algoritmos.

Estos tipos de modelos (MPSVA y MPCVA) se probaron en los 3 tipos de modelos que provienen de los 3 datasets creados en la sección 5.1.3.1. No obstante se resalta que gracias a los primeros acercamientos en los MPSVA se seleccionaron los mejores algoritmos de SL y BL, para las evaluaciones de MPCVA.

A continuación, se definen los tipos de modelos que se generan a partir de los 3 datasets que se establecieron en la sección 5.1.4.1, que son: dataset definitivo, dataset sin 16, dataset solo 16.

5.2.1.1.1 Modelo completo

Descripción: El modelo completo ha sido construido a partir del dataset definitivo, donde se encuentran los 23 datasets unidos.

Posibles ventajas y desventajas: Una de las ventajas de este modelo (que también podría considerarse una desventaja) es la cantidad de datos que almacena tanto en cantidad de datasets, como en diversas actividades. No obstante, como se mencionó en la sección 5.1.3, existe un aspecto a considerar que podría convertirse en un inconveniente a futuro, y es que el dataset 16 conlleva una gran relevancia dentro del mismo modelo, lo que puede generar sesgos en la clasificación por la diferencia de tamaño entre los datasets, lo que se conoce como un desbalance de clases.

5.2.1.1.2 Modelo sin el dataset 16

Descripción: El modelo se construye con datos de 17 datasets (donde se excluyen los ID's de datasets 16 al 21 de la Tabla 17) y un total de 50360 filas y 42 actividades, donde las actividades con más filas son: Caminando, comiendo, estar sentado, corriendo, estar acostado, estar sentado, subir escaleras y bajar escaleras.

Posibles ventajas y desventajas: Una de las ventajas a considerar de esta propuesta de modelo sin el dataset 16, es que, entre menos información tanto en número de filas como de actividades, podría mejorar la clasificación de las actividades; no obstante, una desventaja puede ser que el dataset 16, no haya sido la causa de que los primeros acercamientos no dieran los resultados esperados, y por lo tanto, sea un problema de los demás datasets y se deba revisar dataset por dataset.

5.2.1.1.3 Modelo solo dataset 16

Descripción: Este modelo está conformado en su entrenamiento y evaluación únicamente con el dataset 16, el cual tiene 188.630 filas y 30 actividades.

Posibles ventajas y desventajas: Una ventaja radica en su origen mono fuente, lo que podría conllevar a un mejor rendimiento. Para los autores, comprender más a fondo este dataset constituye una ventaja. Por otro lado, una desventaja significativa, crucial para el análisis, sería que el dataset 16 exhibiera un buen rendimiento. Esto indicaría que dicho dataset no es completamente responsable de los resultados iniciales, significando que los demás datasets se han trabajado erróneamente o no se tienen los resultados que se esperaban.

5.2.1.2 Modelos depurados

Los modelos depurados, a diferencia de los modelos preliminares, no tuvieron en cuenta todas las actividades, esto por el interés de poder enriquecer más el análisis de la diferencia del comportamiento de BL y SL con menos actividades.

5.2.1.2.1 Selección de actividades para modelos depurados

En los modelos depurados, debido a que se busca seguir la línea de pruebas de los modelos MPCVA, también son modelos aleatorizados, además se coloca como prioridad la clasificación de actividades teniendo en cuenta el peso que tiene cada actividad en el dataset con mejor rendimiento (que se obtuvo a raíz de los modelos preliminares). Las actividades seleccionadas fueron entre las ADL más comunes (de las actividades que se tienen en el proyecto).

Gracias a la metodología CRISP-DM, que permite tener sus fases y tareas en un análisis en paralelo. Se conoció que el dataset con mejor rendimiento, fue el dataset sin 16. Por temas de tiempo del presente trabajo de grado, los modelos depurados se hicieron en base a las actividades más comunes de este dataset.

A continuación, en la Imagen 4, se puede apreciar el porcentaje de las actividades dentro del dataset sin 16.

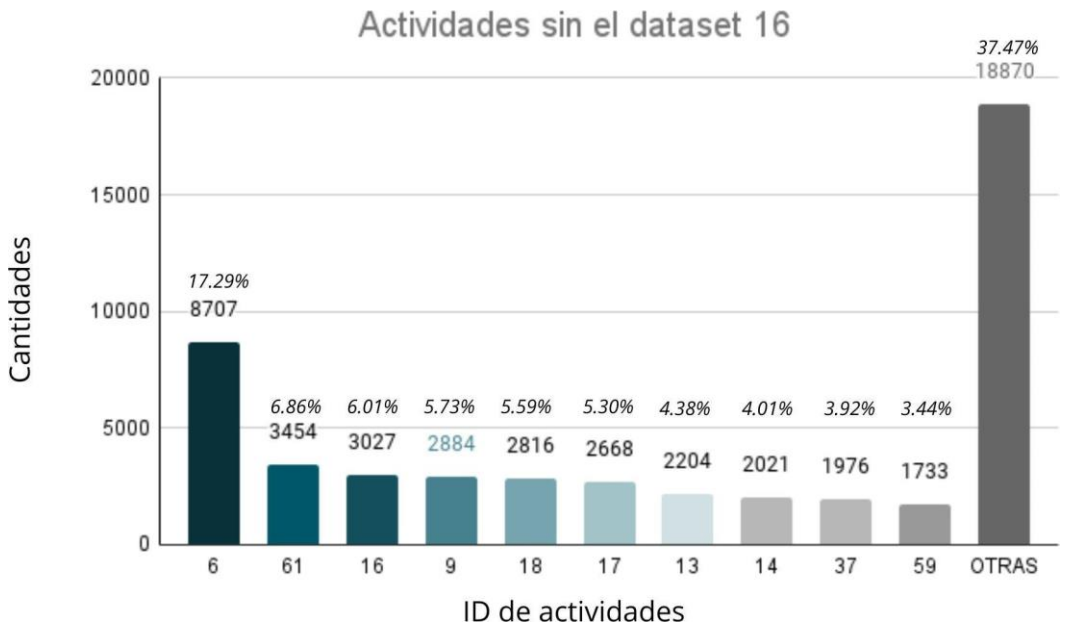

Imagen 4. Actividades sin el dataset 16.

Como se puede apreciar en la imagen 4, las actividades de mayor peso son: caminando (6), comiendo (61), estar sentado (16), corriendo (9), estar dormido (18), estar de pie (17), subir escaleras (13), bajar escaleras (14), escribiendo en el celular (37), jugando baloncesto (59). Las demás actividades se incluyeron en una categoría llamada “otras”, que fue eliminada de inmediato.

Con esto en mente se obtuvo un panorama en referente a la cantidad de datos por actividad, en el anexo Anexo G - Repositorio de Github, del libro “Parte_6_Prueba_de_algoritmos” se encuentra el paso a paso de cómo se obtuvieron estos resultados.

5.2.1.2.2 Conformación de un modelo depurado

Como se puede apreciar en la Imagen 4, se obtuvieron 10 actividades prioritarias, no obstante, por el propósito de este trabajo de grado, se toman en cuenta tan solo 7 de estas actividades para la conformación de los modelos depurados, las cuales son:

- Caminar
- Corriendo
- Estar de pie
- Estar acostado

- Estar sentado
- Subir escaleras
- Bajar escaleras

Se decidió dejar por fuera del análisis las actividades restantes debido a que, como se puede encontrar en el Anexo E, las actividades “comiendo” y “jugando baloncesto” provenían del dataset 13 únicamente, mientras que “escribiendo en el celular” provenía del dataset 8 y 13. Es decir, a diferencia de las demás actividades que se encontraban en varios datasets estas tres solo se encontraban en uno o dos lo cual implicaba que aunque cuentan con un gran número de filas, no así en variedad de fuentes.

Se buscó tener un análisis más riguroso con estas 7 actividades seleccionadas, por lo que se crearon los siguientes modelos depurados:

- Modelo depurado 7 actividades: Se tuvieron en cuenta las 7 actividades seleccionadas.

- Modelo depurado 5 actividades: Para este modelo se excluyeron las actividades de subir y bajar escaleras. Debido a que una vez se tuvieron los resultados del modelo depurado con 7 actividades, se presentó la curiosidad de cómo se trataría el modelo sin las actividades de las escaleras, dado que estas presentaban un menor número de información, así mismo tenían un menor rendimiento de clasificación que las otras. Generando así un problema de rendimiento general del modelo.

En resumen, se obtuvieron modelos preliminares donde el análisis principal era el comportamiento de los algoritmos con todas las actividades y los primeros acercamientos. Mientras que, en los modelos depurados, se centró en seleccionar un número específico de actividades a evaluar. Esto con el fin de tener un enriquecimiento en el análisis de la comparación entre SL y BL.

5.2.2 Plan de prueba

El plan de prueba conlleva los siguientes pasos:

5.2.2.1 Generar los modelos de batch learning

Algoritmos utilizados: Random Forest (RF), Decision Tree (DT), Gaussian Naive Bayes (GNB), K-Nearest Neighbors (KNN) y Logistic Regression (LR).
Números de algoritmos utilizados: 5

Descripción: Se inició con el entrenamiento de modelos BL dado que estos algoritmos son los que tienen más documentación y/o ejemplos. La evaluación se puede apreciar en el cuaderno “Parte_6_Prueba_de_algoritmos” en el repositorio de Github en el *Anexo G*.

5.2.2.2 Generar los modelos de SL

Algoritmos utilizados: Adaptative Random Forest (ADF), K-nearest neighbor (KNN), Hoeffding Tree (HT) y Naive Bayes (NB).
Número de algoritmos utilizados:4

Descripción: El inició de los acercamientos con los algoritmos de SL, se dió en las primeras pruebas en los modelos MPSVA. Gracias a esto se comprendió la naturaleza de SL y se reforzó el entendimiento de BL. La evaluación se puede apreciar en el cuaderno "Parte_6_Prueba_de_algoritmos" en el repositorio de GitHub en el *Anexo G*.

5.2.2.3 Evaluar los modelos obtenidos

En el campo de la ciencia de datos, se dispone de una gran variedad de criterios y métricas para evaluar el desempeño de un modelo. La elección de estas métricas depende en gran medida del problema específico abordado en el proyecto de ciencia de datos. Una de las métricas fundamentales es la Exactitud o Accuracy, la cual compara las clasificaciones del modelo con los valores reales y calcula un porcentaje de aciertos. No obstante, es importante señalar que la exactitud puede generar una falsa sensación de rendimiento cuando existen clases con un desequilibrio significativo en sus tamaños, ya que el modelo puede clasificar eficazmente las clases más numerosas, pero tener un rendimiento deficiente en las clases minoritarias. Además, resulta muy útil cuando se centra en la clasificación de las clases predominantes, que generalmente representan las actividades de la vida diaria más comunes en el dataset generado.

También se consideró la métrica de la puntuación F1 ponderada, especialmente en el contexto de un problema de clasificación multiclase. La puntuación F1 combina información sobre precisión y sensibilidad (recall), ofreciendo así una medida más equilibrada del rendimiento. El término "ponderado" refleja el hecho de que, en este tipo de problemas, se calcula la puntuación F1 para cada clase y se promedian teniendo en cuenta la importancia relativa de las clases predominantes.

Adicionalmente, se evaluó el rendimiento utilizando las curvas ROC (Receiver Operating Characteristic), las cuales contrastan la tasa de falsos positivos con la tasa de falsos negativos. Estas curvas proporcionan información sobre la capacidad del modelo para distinguir las clases positivas cuando la salida real es positiva. Además, se complementó este análisis haciendo uso del Área Bajo la Curva (AUC) como una métrica resumida del rendimiento del modelo. Donde un AUC más alto indica un clasificador más efectivo, donde el valor máximo es 1, correspondiente a un clasificador perfecto, mientras que un clasificador aleatorio tiene un AUC de 0.5. [70,71]

Para realizar un análisis más detallado del rendimiento de los algoritmos, se examinaron la exactitud y el AUC para cada actividad específica.

5.3 Construir los modelos

El primer plan de prueba fue realizado de manera manual. Para dar cumplimiento a los siguientes pasos hay que tener en cuenta que para cada tipo de modelo MPSVA para BL se realizaron pruebas en 5 algoritmos en los 3 tipos de modelos, dando un total de 15 modelos. Mientras que, para SL, en los 3 tipos de modelos se implementaron 4 algoritmos, haciendo que sean 12 modelos donde se tuvo un total de 27 modelos entre SL y BL.

En este punto se tuvo la contemplación de los criterios de éxito de un modelo. El tiempo de entrenamiento del modelo, el tamaño y memoria RAM utilizada. Se entrenaron todos los algoritmos exceptuando el algoritmo SVM, que después de varios intentos con las versiones pagas de Google Colab, y de haber pedido una máquina virtual a la Universidad Del Cauca con 260 GB de RAM y 8 núcleos de CPU, no se logró sacar el cálculo en menos de 10 horas, que fue el tiempo permitido del uso de la máquina al día, así que se decidió descartar la posibilidad de hacer uso de este algoritmo SVM.

Para los tipos de modelo MPCVA, se implementaron 2 algoritmos para BL y 2 algoritmos para SL, aumentando así a 31 modelos.

Inicialmente se planteó la hipótesis que una de las grandes ventajas de los modelos de SL es su eficiencia en recursos de memoria ROM, debido a que no requiere realizar un almacenamiento de todas sus filas, al contrario, va aprendiendo fila por fila y eliminando con el propósito de no tener en cuenta información antigua (o en determinados casos se selecciona la cantidad de información que se considera útil para almacenar).

Se deja un registro del procesamiento de los 31 modelos generados. El tiempo de ejecución total fue de 8 horas, 24 minutos y 57 segundos usando Google Colab. Se hicieron pruebas utilizando la máquina virtual otorgada por la Universidad del Cauca, pero como se mencionó anteriormente, el algoritmo SVM tardó más de 10 horas de procesamiento y no terminó la generación de su modelo. No obstante, se probaron en la máquina virtual algoritmos de RF, donde la diferencia entre la máquina virtual y las instancias T4 de Google Colab, fueron tan mínimas, que se decidió solamente seguir utilizando Colab. Los criterios de evaluación en el *Anexo I - Modelos MPSVA, MPCVA y depurados*, son Accuracy, Precision, F-1, AUC y la curva ROC.

Para las secciones que quedan del capítulo 5, se trabajó con los dos mejores algoritmos para SL y BL, que son Adaptative Random Forest (ARF), KNN y Random Forest (RF), Decision Tree (DT), respectivamente. El *Anexo I* contiene en detalle los resultados de rendimiento de los algoritmos para la generación de los 31 modelos.

5.4 Evaluar los modelos

Para el análisis de la evaluación de los modelos de esta fase se tuvo en cuenta el criterio de éxito: "*Lograr un nivel de precisión igual o mayor a 70%, en la clasificación de actividades de la vida diaria en un modelo menor a 10 MB consumiendo una RAM inferior a 0.5 GB".* Este criterio se aplicó a todos los modelos generados para SL y BL, con el fin de seleccionar los mejores. Así mismo todos los resultados de los modelos completos y con sus respectivas gráficas ROC se encuentran en el *Anexo I - Modelos MPSVA, MPCVA y depurados.*

5.4.1 Evaluación de modelos preliminares sin valores aleatorios (MPSVA)

5.4.1.1 Evaluación modelos MPSVA BL

En la Tabla 19 se pueden observar los algoritmos de BL en los MPSVA generados. Se puede encontrar la información completa en el Anexo I.

Al realizar el primer análisis de la Tabla 19, se percibió un rendimiento notablemente superior en los modelos que prescindieron del dataset 16. Esto confirma las hipótesis planteadas: el dataset 16, al encontrarse en un entorno no controlado, contribuyó a la incertidumbre en sus datos, dado que los usuarios etiquetaban sus propias actividades.

En cuanto al rendimiento de los algoritmos, RF presentó mejores resultados que DT. No obstante, esto fue compensado con las medidas de tiempo, tamaño y consumo de RAM. Este modelo tiene todas las actividades de cada uno de sus datasets con las actividades organizadas en un entorno "ideal"; sin embargo, es de resaltar que los demás algoritmos presentaron métricas irrelevantes, por lo que se concluyó que no están hechos para una tarea de clasificación como la presentada en el proyecto. En general tuvo un buen desempeño BL.

Batch Learning							
Modelos MPSCA	*Algoritmos*	*Accuracy*	*AUC*	*F-1*	*Tiempo (mm:ss)*	*Tamaño*	*RAM*
Modelo Completo	Ramdon Forest	0,7	0,93	0,68	04:00	3,26 GB	1,5 Gigas
	Decision Tree	0,59	0,76	0,59	00:30	39,2 MB	0,07 Gigas
	Gaussian Naive Bayes	0,11	0,73	0,06	00:08	46 KB	2,10 Gigas
	KNN	0,53	0,81	0,51	03:20	63,8 MB	0,05
	Regresión Logística	0,43	0,81	0,34	07:28	24 KB	0,1 Gigas
Modelo sin el dataset 16	Ramdon Forest	0,92	1,0	0,92	00:41	296 KB	0,3 Gigas
	Decision Tree	0,82	0,9	0,82	00:13	2,9 MB	0,01 Gigas
	Gaussian Naive Bayes	0,35	0,9	0,31	00:07	35 KB	0 Gigas
	KNN	0,51	0,85	0,5	00:19	13,4 MB	0 Gigas
	Regresión Logística	0,45	0,9	0,4	01:34	19 KB	0 Gigas
Modelo solo dataset 16	Ramdon Forest	0,64	0,9	0,61	02:07	1,68 GB	1,5 Gigas
	Decision Tree	0,52	0,71	0,52	00:14	20,8 MB	0,2 Gigas
	Gaussian Naive Bayes	0,08	0,68	0,03	00:8	26 KB	1,02 Gigas
	KNN	0,55	0,79	0,53	02:07	50,4 MB	0,04 Gigas
	Regresión Logística	0,48	0,77	0,38	4:20	14 KB	0,04 Gigas

Tabla 19. Evaluación modelos MPSVA BL.

5.4.1.2 Evaluación modelos MPSVA Stream Learning 70%

En la Tabla 20, se puede observar el entrenamiento y evaluación de los algoritmos de SL en el 70% de cada uno de los modelos, el éxito de las métricas es bastante evidente debido a los valores tan cercanos al 100% de éxito mostrando las primeras diferencias del uso del aprendizaje incremental de SL a comparación del aprendizaje clásico de BL. Además, con un tamaño considerablemente menor al de los modelos de BL, pero con un tiempo de entrenamiento más alto.

En estos primeros acercamientos que permitieron los MPSVA, se percató que la variación de la ventana del algoritmo de KNN tenía una gran repercusión en su rendimiento, entre mayor sea la ventana, mayor es la cantidad de datos que se guarda en su memoria. Al final se estableció una ventana de 100, debido a que después de algunas pruebas realizadas en el cuaderno 'Parte_6_Prueba_de_Algoritmos' del repositorio de Github del *Anexo G*, fue la ventana que alcanzó la mejor exactitud.

Al final de este entrenamiento y evaluación, dado a que se recreó un escenario similar al funcionamiento similar a BL, se generaron los archivos .pkl de cada modelo en este 70%,

para evaluarlo en la siguiente sesión del 30%.

Stream Learning (70 %)							
Modelos MPSCA	*Algoritmos*	*Accuracy*	*AUC*	*F-1*	*Tiempo (mm:ss)*	*Tamaño*	*RAM*
Modelo Completo (70 %)	Adaptative Random Forest	0,99	0,98	0,94	16:24	221 KB	0
	Hoeffding Tree	0,74	0,81	0,59	05:31	1.8 MB	0
	Naive Bayes	0,62	0,74	0,44	05:20	385 KB	0
	KNN	0,99	0,98	0,93	01:30	41 KB	0
Modelo sin el dataset 16 (70 %)	Adaptative Random Forest	0,98	0,98	0,93	05:21	331 kB	0
	Hoeffding Tree	0,8	0,87	0,71	02:33	730 kB	0
	Naive Bayes	0,65	0,81	0,6	01:20	291 KB	0
	KNN	0,98	0,97	0,918	00:45	41 KB	0
Modelo solo dataset 16 (70 %)	Adaptative Random Forest	0,99	0,96	0,89	12:10	112 kB	0
	Hoeffding Tree	0,75	0,72	0,4	03:20	1.5 MB	0
	Naive Bayes	0,18	0,54	0,08	02:25	204 KB	0
	KNN	0,99	0,94	0,85	01:12	40 KB	0

Tabla 20. Evaluación modelos MPSVA Stream Learning 70%.

5.4.1.3 Evaluación modelos MPSVA Stream Learning 30%

En la Tabla 21 se pueden apreciar los resultados de la evaluación en el 30% restante del dataset. En este caso, en lugar de usar los algoritmos directamente de la librería de scikit-multiflow se usaron los modelos creados anteriormente.

Hay que resaltar que estos resultados presentados en la Tabla 21, aunque lograban cumplir el criterio de éxito en cuestión de rendimiento. Se está cumpliendo en un escenario “ideal”, no obstante, fue de gran utilidad estos primeros acercamientos a los algoritmos de SL y BL, para el análisis y creación de los demás modelos.

Los algoritmos de ARF y KNN presentaron métricas cercanas al cien por ciento (100%), para este punto los algoritmos de HT decayeron en gran medida y peor aún NB, lo cual fue bastante preocupante recordando que se encontraban en el entorno “ideal”.

Batch Learning (30 %)							
Modelos MPSCA	*Algoritmos*	*Accuracy*	*AUC*	*F-1*	*Tiempo (mm:ss)*	*Tamaño*	*RAM*
Modelo Completo (30 %)	Adaptative Random Forest	0,99	0,975	0,92	07:36	No aplica	0
	Hoeffding Tree	0,4	0,74	0,37	07:17		0
	Naive Bayes	0,27	0,65	0,2	06:27		0
	KNN	0,98	0,94	0,87	03:21		0
Modelo sin el dataset 16 (30 %)	Adaptative Random Forest	0,96	0,98	0,91	01:53		0
	Hoeffding Tree	0,55	0,81	0,53	03:01		0
	Naive Bayes	0,35	0,75	0,38	01:40		0
	KNN	0,94	0,92	0,83	01:10		0
Modelo solo dataset 16 (30 %)	Adaptative Random Forest	0,99	0,93	0,83	05:19		0
	Hoeffding Tree	0,3	0,59	0,13	05:25		0
	Naive Bayes	0,291	0,55	0,05	02:45		0
	KNN	0,98	0,9	0,79	02:13		0

Tabla 21. Evaluación modelos MPSVA Stream Learning 30%.

Finalizando los MPSVA, dado que fueron modelos de “ensayo y error” se optó por la idea de clasificar actividades con estos modelos de RF que rondan el 100% de rendimiento. Sin embargo, una vez se intentó clasificar una actividad, el resultado sorprendió bastante, debido a que fue de 0%. Esto generó una investigación más profunda del comportamiento de los algoritmos de SL, y se concluyó que esto era por la limitada memoria de los algoritmos de SL, debido a que en su memoria solo se quedaba con la última actividad, por tanto, cualquier otra actividad que no fuera la misma, daría 0%. Mientras que en BL, no se presentó este inconveniente, debido a que BL no tiene relevancia si los datos están o no ordenados.

5.4.2 Evaluación de modelos preliminares con valores aleatorios (MPCVA)

Con los resultados de los modelos MPSVA, se puede concluir que, aunque se cumple el criterio de éxito en cuestión de rendimiento en ambos tipos de modelo, sigue siendo un entorno “ideal” y fue un escenario teórico creado por los autores del presente dataset, para el entendimiento de los algoritmos. Con esto en mente se da inició a la evaluación de los MPCVA.

En esta sección se aleatorizaron las actividades de los datasets con el fin de hacer una mayor similitud a cómo podrían comportarse los modelos en un entorno “real”. Además, con esta aleatoriedad de los datos, para este proceso se evaluaron los dos mejores algoritmos de MPSVA. Así mismo se evidenció que a los algoritmos de BL no presentan ningún cambio con la aleatorización y los resultados se mantienen igual que en la Tabla 19, es por eso que, para enriquecer más el análisis, en los MPCVA para BL se utilizó la técnica de validación cruzada.

5.4.2.1 Modelos MPCVA SL

Para los MPCVA se tiene como característica principal que tiene todas las actividades de los modelos y valores aleatorizados. En la Tabla 22, se pueden apreciar que los algoritmos de SL, donde se entrenaron y evaluaron cada uno de los datasets completos, no presenta los mejores resultados. Se concluye que la aleatorización tiene un gran impacto para SL, debido a que BL no sufrió ningún tipo de cambio.

Stream Learning								
Modelos MPCVA	***Algoritmos***	***Accuracy***	***Precision***	***AUC***	***F-1***	***Tiempo (mm:ss)***	***Tamaño***	***RAM***
Modelo Completo	Random Forest	0,38	0,35	0,58	0,20	85:12	2 MB	0 Gigas
	KNN	0,43	0,21	0,57	0,16	49:54	3.8 MB	0 Gigas
Modelo sin el dataset 16	Random Forest	0,40	0,40	0,68	0,37	30:23	1.1 MB	0 Gigas
	KNN	0,41	0,30	0,62	0,26	41:52	3.8 MB	0 Gigas
Modelo solo dataset 16	Random Forest	0,47	0,25	0,53	0,08	48:40	2.6 MB	0 Gigas
	KNN	0,480	0,21	0,56	0,15	50:06	3,8 MB	0 Gigas

Tabla 22. Modelos MPCVA SL.

Debido a la limitada memoria de SL los algoritmos no lograban tener una fuente suficiente de muestras de una misma actividad para aprender correctamente lo cual le impide al algoritmo aprender o reforzar su eficacia al momento de clasificar.

A pesar de que los valores de Accuracy en los 3 modelos no cumple los criterios de éxito, cabe resaltar que en los modelos sin el dataset 16, el AUC general del modelo y el F-1 score es considerablemente superior a los presentes en el Modelo completo y Modelo solo 16 respectivamente.

Se concluye que, en un escenario de todas las actividades, con valores aleatorios SL no cumple el criterio de éxito en cuestión de rendimiento.

5.4.2.2 Modelos MPCVA BL validación cruzada

En el cuaderno de "Parte_6_Prueba_de_algoritmos" en Github del *Anexo G*, se apreció que los algoritmos de BL no se ven afectados con valores aleatorios. No obstante, para enriquecer el análisis se corroboró con otra técnica estos resultados de la Tabla 19.

La validación cruzada [79], es una técnica para evaluar y validar el rendimiento de un modelo de manera robusta. Esta técnica divide el conjunto de datos en subconjuntos de datos (en este caso se dividieron en 10 subconjuntos). Luego, se entrena y evalúa el modelo en múltiples iteraciones, utilizando diferentes combinaciones de los subconjuntos de datos para el entrenamiento y prueba. El propósito de esto es evaluar cómo se comportó el modelo en diferentes secciones del dataset para evitar el sobreajuste o el sesgo. En la tabla 23 se presentan los resultados de esta validación cruzada. Aunque los valores de las métricas disminuyeron, disminuyeron muy poco a comparación de SL.

Batch Learning Validación Cruzada								
Modelos MPCVA	*Algoritmos*	*Accuracy*	*Precision*	*AUC*	*F-m*	*Tiempo (mm:ss)*	*Tamaño*	*RAM*
Modelo Completo	Random Forest	0,70	0,72	0,95	0,68	15:09	No Aplica	3,2 Gigas
	Decision Tree	0,58	0,58	0,78	0,58	02:09		0,35 Gigas
Modelo sin el dataset 16	Random Forest	0,91	0,91	1,0	0,91	03:55		1,01 Gigas
	Decision Tree	0,81	0,81	0,88	0,81	00:30		0,17 Gigas
Modelo solo dataset 16	Random Forest	0,64	0,66	0,85	0,60	08:16		2,2 Gigas
	Decision Tree	0,52	0,52	0,67	0,52	01:17		0,16 Gigas

Tabla 23. Modelos MPCVA validación cruzada.

Se concluye que, en un escenario de todas las actividades, con valores aleatorios o no aleatorios BL cumple el criterio de éxito en cuestión de rendimiento.

5.4.3 Evaluación modelos depurados

Como se aprecia en los resultados de los MPSVA y MPCVA, el dataset que mejor rendimiento tuvo en los modelos en los diferentes escenarios propuestos fue el dataset sin
16. Así que se escogió el “Modelo sin el dataset 16” como el mejor modelo tanto para BL como SL. Ya se observó cómo es el rendimiento con todas las actividades en los modelos preliminares, ahora en esta sección se apreciará cómo es el comportamiento de los algoritmos con pocas actividades.

Lo más importante para un modelo depurado es buscar mejorar la generalización reduciendo la dimensionalidad con el fin de reducir el riesgo de un sobreajuste (overfitting) en los modelos de clasificación ya que se eliminan actividades (como las categorizadas como OTROS de la imagen 4). Se generaron los modelos depurados, uno con 5 actividades y el otro con 7 actividades. [80]

5.4.3.1 Evaluación modelos depurados de BL

En la tabla 24, se puede apreciar que los modelos de BL con 5 y 7 actividades, tienen un rendimiento bastante óptimo en las métricas consideradas. Se obtuvo como mejor algoritmo para BL el algoritmo de RF, aunque el algoritmo DT lo superó en tiempo de entrenamiento, tamaño y consumo de memoria con métricas similares. Además, se cumplió completamente el criterio de éxito planteado, excepto para el algoritmo de RF en el modelo de 7 actividades que no cumple el criterio de éxito para la RAM.

Batch Learning								
Modelos Depurados	***Algoritmos***	***Accuracy***	***Precision***	***AUC***	***F-m***	***Tiempo (mm:ss)***	***Tamaño***	***RAM***
5 Actividades	Random Forest	0,98	0,98	1,0	0,98	01:10	9 MB	0,12 Gigas
	Decision Tree	0,95	0,95	1,0	0,95	00:10	89 KB	0,02 Gigas
7 Actividades	Random Forest	0,95	0,95	0,97	0,95	01:40	26 MB	0,40 Gigas
	Decision Tree	0,89	0,89	0,93	0,89	00:15	257 KB	0,05 Gigas

Tabla 24. Evaluación de modelos depurados de BL.

5.4.3.2 Evaluación modelos depurados de SL

Se puede apreciar en la Tabla 25, que los rendimientos de los algoritmos de SL, no presentaron una mejora a comparación de BL, no obstante SL si cumple todo el criterio de éxito, debido a que en el tamaño del modelo y la RAM, presenta una mejoría significativa a diferencia de BL.

Es notable la mejora del rendimiento de los algoritmos gracias a la reducción de actividades, en comparación a los anteriores modelos, dado que antes las métricas no lograban alcanzar el criterio de éxito. Mientras que ahora los resultados de ambas tablas tanto para SL como para BL superan el umbral del 70%, incluso con datos aleatorizados. Esto creó una mayor confianza para las siguientes pruebas.

El hiper parámetro del tamaño de la ventana del algoritmo de KNN, tuvo diferentes pruebas con valores de 100, 1000, 2000, 10000 y todas las filas, donde se encontró una relación directamente proporcional entre el aumento de filas y el aumento del rendimiento del algoritmo. No obstante, esto también aumentaba de manera considerable el tiempo de procesamiento. Sin embargo, no se encontró una diferencia muy grande en los resultados entre hacer el uso de 10 mil y 20 mil filas, por tanto para evitar el aumento del tiempo de entrenamiento, se tomó la decisión de hacer uso de 10.000 filas, estas pruebas se pueden encontrar en el cuaderno "*Parte_7_Entrenamiento_con_5_y_7_actividades*" del repositorio de Github del *Anexo G.*

Stream Learning								
Modelos Depurados	***Algoritmos***	***Accuracy***	***Precision***	***AUC***	***F-m***	***Tiempo (mm:ss)***	***Tamaño***	***RAM***
5 Actividades	Random Forest	0,88	0,72	0,90	0,71	00:30	1,6 MB	0
	KNN	0,83	0,68	0,87	0,67	19:10	3,8 MB	0
7 Actividades	Random Forest	0,69	0,62	0,76	0,52	00:42	914 KB	0
	KNN	0,70	0,59	0,79	0,56	12:34	9,3 MB	0

Tabla 25. Evaluación modelos depurados de SL.

5.5 Análisis modelos depurados

5.5.1 Área bajo la curva (AUC) de los modelos depurados

Se presentan las gráficas ROC de los modelos depurados, donde se pudo observar con mayor claridad el rendimiento no solo de los modelos, sino también el de cada clase. Las imágenes de BL presentan áreas bajo la curva (AUC) cercana a uno (e incluso uno en algunos casos), notándose mejor en RF, especialmente en las clases 13 y 14, que en DT toman valores bajos en comparación.

En cuanto a las gráficas de SL, el área bajo la curva decae notoriamente, sin embargo, siguen siendo valores bastante buenos, con la excepción de las clases 13 y 14, lo cual es de denotar para este momento y dando un indicio sobre qué esperar de las pruebas realizadas en el siguiente capítulo. Finalmente, se aprecia que en SL el algoritmo de ARF supera al KNN por una diferencia notable, al menos en cuestión de AUC por clase.

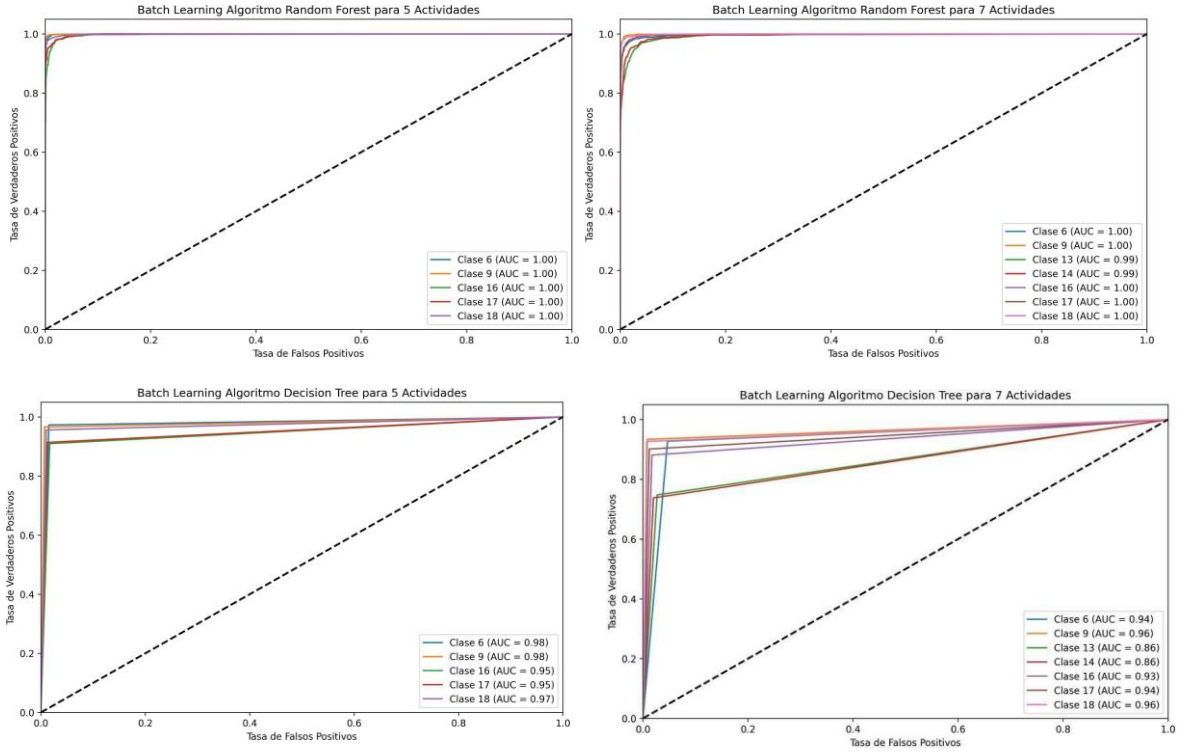

Imagen 5 a, b, c y d: Gráficas ROC de los modelos depurados BL.

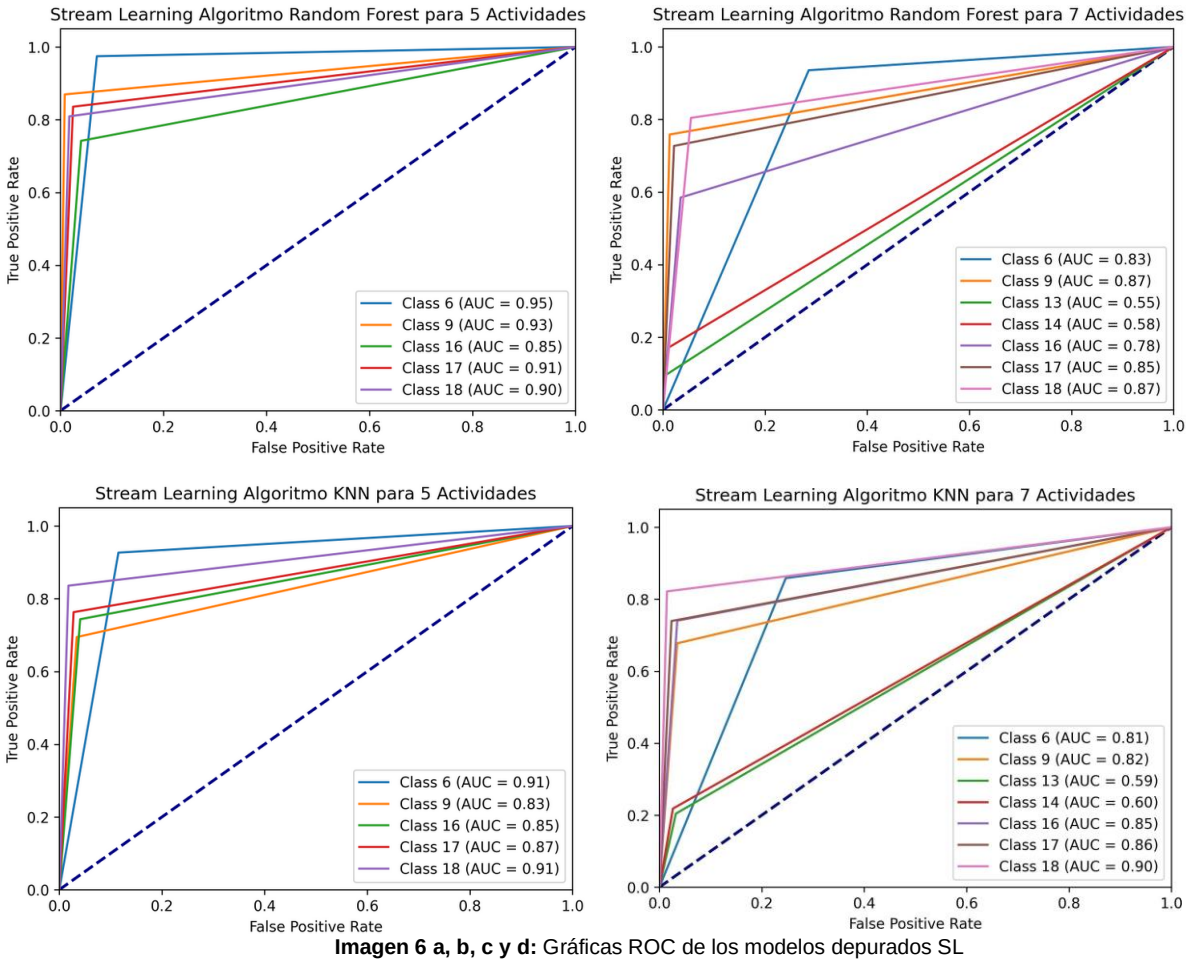

Imagen 6 a, b, c y d: Gráficas ROC de los modelos depurados SL

5.5.2 Accuracy de los modelos depurados

En las siguientes imágenes se puede apreciar el rendimiento de los algoritmos con el paso del entrenamiento y evaluación. Se resalta que las imágenes de curvas de aprendizaje se encuentran para todos los modelos en el *Anexo I*.

En las imágenes se observa claramente que, con solo 5 actividades, el accuracy mejora significativamente tanto en ARF como en KNN. La gráfica también muestra una mejora más rápida en los valores de accuracy, lo que respalda la idea de que, al menos en estos algoritmos de SL, no es aconsejable intentar clasificar muchas actividades aleatorias. Sin embargo, los resultados con 7 actividades siguen siendo notables. El principal desafío radicó en el número de iteraciones necesarias para alcanzar estos valores, posiblemente debido a la dificultad de clasificar las clases 13 y 14, lo que prolongó el proceso de aprendizaje.

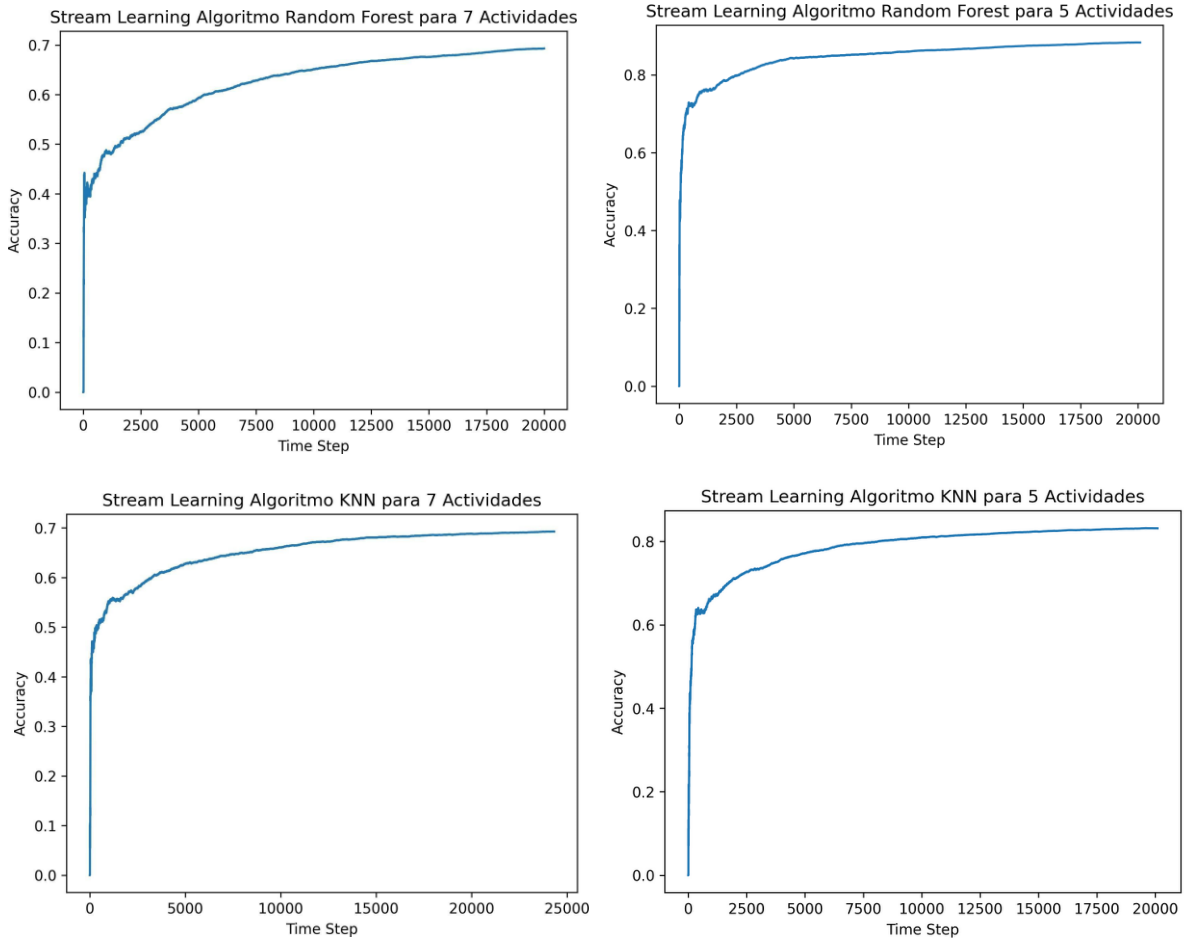

Imagen 7 a, b, c y d: Curvas de aprendizaje de los modelos depurados.

5.6 Análisis del modelado: primeras diferencias encontradas BL y SL

Es relevante recalcar que en la finalización del capítulo 5, se tiene el primer análisis de las diferencias encontradas entre BL y SL, estas conclusiones se completaran y se terminarán en la finalización del siguiente capítulo, pero sí se puede concluir, como se pudieron apreciar en la creación de modelos preliminares y modelos depurados, es que la elección de qué algoritmo es mejor entre BL y SL, es que depende del caso del negocio donde se necesite implementar. Esto se analizará más a detalle en el siguiente capítulo, por lo pronto se dejan las primeras diferencias:

- Los modelos de SL presentaron resultados muy poco favorables cuando no se aleatorizaron los datos para la clasificación de actividades; esto debido a que en su memoria se almacenó información de la última actividad del dataset con el cual se entrenó, lo que causó que esa fuera la única actividad que pudo clasificar, por tanto una vez se intentó clasificar una actividad diferente a ésta, su rendimiento fue nulo.

- Los modelos de BL no tuvieron ningún impacto significativo al momento de ser entrenados con datasets randomizados. Debido a que su entrenamiento distribuido en lotes permite almacenar todo el dataset.

- Los modelos de SL, en un escenario donde fueron entrenados con un dataset desordenado con gran cantidad de clases, aún presentaron cierta dificultad para la predicción de actividades. Esto se debió a que, entre más era la cantidad de clases que se les entregaban, peores fueron sus resultados. Mientras que BL, sigue presentando buenos resultados incluso si la cantidad de clases es grande.

- Los modelos de BL presentaron un mayor peso de sus archivos y un mayor uso de memoria RAM a comparación de los modelos de SL.

- Los modelos depurados de SL tuvieron un mejor rendimiento, debido a que tiene menos clases qué almacenar y así mismo ya no se tiene un desbalance de clases que ocasione inconvenientes en la clasificación, haciendo que tanto BL como SL cumplan el criterio de éxito, para poder ser evaluados con datasets de pruebas.

- Los algoritmos de RF, aunque presentaron una gran mejoría respecto a algoritmos como DT (Para BL) o KNN (Para SL), generaron modelos significativamente más pesados, así mismo consumieron mayor cantidad de RAM para poder entrenar los modelos.

- En las gráficas ROC, los resultados de BL fueron extremadamente cercanos a 1.0 en cada clase, es decir que alcanzaron predicciones casi perfectas, aun así, los modelos de SL se defendieron bastante bien, pero se vieron bastante afectados cuando se agregaron las actividades de subir y bajar escaleras.

Para el siguiente capítulo se usaron los 4 modelos de 5 y 7 actividades, para algoritmos de SL y BL que cumplieron completamente el criterio de éxito. Estos modelos se pueden encontrar en el *Anexo J. Datasets finales, en Kaggle.*

En el capítulo 6, se analizó cómo es el comportamiento de estos datasets para clasificar actividades de datasets con los que fue entrenado (datasets internos) y datasets con los que no fue entrenado (datasets externos). Para finalizar en el siguiente capítulo se determinó cuál fue el mejor modelo que se integró en el SHAR o sistema móvil para clasificar ADL llamado *Jaida*.

Capítulo 6.

Fase 5: Evaluación

En esta fase se evaluaron los modelos obtenidos en el capítulo anterior. La evaluación tuvo como objetivo determinar si los modelos en los escenarios planteados cumplen el criterio de éxito del problema del negocio. Además, se seleccionó el mejor modelo a consideración de los autores para el sistema móvil Jaida. Las tareas propias de esta fase son las siguientes:

- **Escenarios de evaluación**
    - Escenario 1: clasificación de BL
    - Escenario 2: clasificación de SL
    - Escenario 3: clasificación incremental de SL
- **Evaluar los resultados**
    - Valoración de los resultados
- **Análisis final: Comparación SL y BL**
    - Modelos aprobados
- **Revisión del proceso y determinación de próximos pasos**

6.1 Escenarios de evaluación

Para enriquecer más el análisis de resultados de la fase 5 y con el fin de una mejor respuesta al objetivo específico número dos, se plantearon 3 escenarios de evaluación:

6.1.1 Escenario 1: clasificación de BL

Haciendo uso de los modelos de BL obtenidos en la fase anterior se planteó el primer escenario que consistió en una clasificación de actividades de cada una de las filas de cada dataset, calculando las métricas Accuracy, Precision y F1 en base a las clasificaciones realizadas y las actividades originales. Tal y como una evaluación clásica de ML.

6.1.2 Escenario 2: clasificación de SL

El escenario es exactamente igual al anterior con la diferencia de que en este escenario se usaron los modelos de SL.

6.1.3 Escenario 3: clasificación incremental de SL

Este escenario se planteó pensando en el aprendizaje incremental propio de SL, para estas evaluaciones se asumió que los datasets ya están etiquetados de la manera que espera el modelo y por tanto luego de cada clasificación el algoritmo se alimentaba de las etiquetas proporcionadas para reforzar su aprendizaje, y en teoría adaptarse a las muestras proporcionadas.

6.2 Evaluar los resultados

Para la evaluación de los datos en esta fase, se utilizaron los modelos que se entregaron en la fase anterior. Los cuales se evaluaron en los escenarios planteados anteriormente, y cada escenario se puso a prueba con los datasets externos e internos, que se definen a continuación.

6.2.1 Valoración de los resultados

Para los datasets de prueba se escogieron los datasets 5, 9, 16 y 20, que su tamaño e información se pueden encontrar en la Tabla 17, así mismo, los resultados completos de esta diferenciación se encuentran en el *Anexo F - Resultados Stream VS Batch*.

A continuación, la definición de un dataset externo e interno y el por qué se seleccionaron estos 4 datasets.

- Datasets Externos: Se denominaron datasets externos a los datasets con los que no fue entrenado el modelo. Estos datasets recrearon valores de un entorno “real”. Los datasets que conforman esta categorización fueron los datasets 9 y 16 (ID’s 2 y 20, respectivamente de la Tabla 17). Anteriormente se planteó que el dataset 16 tenía valores con bastante incertidumbre, y resultó interesante ver su comportamiento de los modelos tratando de clasificar actividades de este dataset.

- Datasets Internos: Se denominaron datasets internos a los datasets con los que se han entrenado los modelos, estos datasets supusieron pruebas con datos en una simulación más “ideal”. Los datasets seleccionados fueron los datasets 5 y 20 (ID’s 7 y 22 respectivamente de la Tabla 17) se seleccionaron estos dos datasets debido a la diferencia de filas entre los mismos, dado que el dataset 5 cuenta con el doble de filas del dataset 20, además el dataset 20 no cuenta con medidas de giroscopio, a diferencia del dataset 5. Por tanto, se consideraron para las pruebas por resultar de interés su análisis.

6.2.1.1 Evaluación del escenario 1: Clasificación de BL

Para el escenario 1 que consiste en la clasificación de BL se escogieron los dos mejores algoritmos RF y DT. Las métricas que se tuvieron en cuenta fueron Accuracy, Precision y F1, se pueden encontrar los resultados completos en el *Anexo E - Resultados Stream VS Batch*.

Para este análisis se tuvieron en cuenta las métricas más significativas que son Accuracy y Precision. Con esto se evaluó la clasificación y el desequilibrio de clase. A continuación, se presentan las Tablas 26 y 27, que hacen referencia al rendimiento de los modelos con los datasets de prueba y el rendimiento de las clases de los modelos, respectivamente.

Modelos Batch								
Modelos/ Datasets	**Externos**				**Internos**			
	Dataset 9 (4 actividades)		Dataset 16 (5 actividades)		Dataset 20 (5 actividades)		Dataset 5 (5 actividades)	
Modelos 5 actividades	Accuracy	Precision	Accuracy	Precision	Accuracy	Precision	Accuracy	Precision
Random Forest	0,63	0,70	0,59	0,71	0,88	0,89	0,97	0,97
Decision Tree	0,66	0,69	0,39	0,69	0,81	0,81	0,95	0,96
Modelos/ Datasets	Dataset 9 (6 actividades)		Dataset 16 (7 actividades)		Dataset 20 (7 actividades)		Dataset 5 (7 actividades)	
Modelos 7 actividades	Accuracy	Precision	Accuracy	Precision	Accuracy	Precision	Accuracy	Precision
Random Forest	0,44	0,54	0,58	0,70	0,86	0,86	0,98	0,98
Decision Tree	0,37	0,41	0,30	0,65	0,73	0,73	0,94	0,94

Tabla 26. Modelos externos e internos en Batch Learning

Accuracy actividades de Batch Learning									
Dataset		**Random Forest con 7 actividades**				**Decision Tree con 7 actividades**			
ID Actividad	Nombre Actividad	Dataset 9 (6)	Dataset 16	Dataset 20	Dataset 5	Dataset 9 (6)	Dataset 16	Dataset 20	Dataset 5
6	Caminando	0.93	0.37	0.88	0.99	0.71	0.32	0.72	0.95
9	Corriendo	NA	0	0.95	1	NA	0	0.68	1
13	Subir escaleras	0.04	0	0.84	0.97	0.19	0,72	0.75	0.89
14	Bajar escaleras	0.03	0	0.86	0.97	0.03	0	0.76	0.98
16	Estar sentado	0.27	0.85	0.84	0.98	0.23	0.8	0.79	0.95
17	Estar de pie	0.97	0.08	0.84	0.95	0.82	0.07	0.69	0.91
18	Estar acostado	0.27	0.1	0.83	0.99	0.18	0.06	0.7	0.93
Dataset		**Random Forest con 5 actividades**				**Decision Tree con 5 actividades**			
ID Actividad	Nombre Actividad	Dataset 9 (4)	Dataset 16	Dataset 20	Dataset 5	Dataset 9 (4)	Dataset 16	Dataset 20	Dataset 5
6	Caminando	1	0.38	0.23	1	1	0.36	0.87	98
9	Corriendo	NA	0	0.25	1	NA	0,3	0.89	0.99
16	Estar sentado	0.25	0.86	0.17	0.96	0.2	0.84	0.76	0.96
17	Estar de pie	0.26	0.08	0.17	0.95	0.90	0.05	0.72	0.94
18	Estar acostado	1	0.11	0.16	0.98	0.44	0.09	78	0.92

Tabla 27. Accuracy de actividades de los modelos de BL.

- Evaluación de modelos externos en BL

Los modelos de BL, cuando estuvieron a prueba con datasets externos, presentaron una disminución en su Accuracy y Precision bastante considerables. A diferencia de las pruebas con datasets internos. También se resalta que el dataset 16 no tiene una alta confiabilidad, así que las pruebas de clasificación con este dataset, es normal que no se hayan alcanzado valores de rendimiento altos.

Por último, en el análisis por actividades, las actividades de subir y bajar escaleras se clasificaron de manera muy mediocre, hablando de cada dataset, en el dataset 9, las actividades de caminando y estar de pie se clasificaron correctamente, no aspi las demás, y en el dataset 16, la única actividad que se clasificó correctamente fue estar sentado. Probablemente estos resultados tan malos, se debe a que a los modelos de Batch les cuesta clasificar datos desconocidos, tal vez por el sobre ajuste debido a la cantidad de filas.

- Evaluación de modelos internos en BL

Cuando los modelos de BL clasificaron actividades de datasets con los que fueron entrenados los modelos, demostró tener un mejor rendimiento. Actividades como “caminando” o “subir escaleras” que con datasets externos la clasificación tenía una tendencia a 0%, con los datasets ya conocidos vuelve a tener un porcentaje cercano al 90%. También se resalta que, aunque los dos datasets internos presentan excelentes resultados, el dataset 5 obtiene una mejoría considerable, debido a que el dataset 5 si posee información de acelerómetro y giroscopio, mientras que el dataset 20 solo posee valores del acelerómetro.

En la métrica de Accuracy, todos los modelos internos de BL tienen resultados sumamente positivos. Se resalta que con los datos con los que ya fue entrenado el modelo, BL supera las expectativas, incluso en actividades como subir y bajar escaleras, se logran clasificar correctamente.

6.2.1.2 Evaluación del escenario 2: Clasificación de SL

Para el escenario 2, clasificación de SL se encontraron resultados de gran valor. Se recuerda al lector que se escogieron los dos mejores algoritmos de este tipo de modelos SL, los cuales fueron ARF y KNN, las métricas que se tuvieron en cuenta fueron Accuracy, Precision y F1, que se pueden encontrar completas en el *Anexo E - Resultados Stream VS Batch*, para este análisis se tuvieron en cuenta las métricas más significativas para el análisis del presente proyecto que fueron Accuracy y Precision, esto para evaluar correctamente la clasificación y el desbalance de clase, con esto en mente se analizaron los modelos externos e internos *(Tablas 28 y 29*).

Modelos Stream Learning								
Modelos/ Datasets	**Externos**				**Internos**			
	Dataset 9 (4 actividades)		Dataset 16 (5 actividades)		Dataset 20 (5 actividades)		Dataset 5 (5 actividades)	
Modelos 5 actividades	Accuracy	Precision	Accuracy	Precision	Accuracy	Precision	Accuracy	Precision
Random Forest	0,66	0,7	0,49	0,72	0,58	0,65	0,78	0,82
KNN	0,63	0,66	0,29	0,68	0,53	0,55	0,64	0,64
Modelos/ Datasets	Dataset 9 (6 actividades)		Dataset 16 (7 actividades)		Dataset 20 (7 actividades)		Dataset 5 (7 actividades)	
Modelos 7 actividades	Accuracy	Precision	Accuracy	Precision	Accuracy	Precision	Accuracy	Precision
Random Forest	0,63	0,64	0,43	0,68	0,47	0,62	0,58	0,57
KNN	0,42	0,44	0,27	0,66	0,39	0,42	0,55	0,58

Tabla 28. Modelos externos e Internos en SL.

Stream Learning									
Dataset		**Random Forest con 7 actividades**				**KNN con 7 actividades**			
ID Actividad	Nombre Actividad	Dataset 9 (6)	Dataset 16	Dataset 20	Dataset 5	Dataset 9 (6)	Dataset 16	Dataset 20	Dataset 5
6	Caminando	1	0,18	0,88	0,4	0,55	0,41	0,49	0,38
9	Corriendo	NA	0,09	0,81	0,95	NA	0	0,85	0,68
13	Subir escaleras	0	0	0,91	0	0,14	0	0,22	0,27
14	Bajar escaleras	0	0	0,02	0	0,2	0	0,08	0,35
16	Estar sentado	0,1	0,86	0,61	0,81	0,5	0,78	0,59	0,75
17	Estar de pie	0,99	0,09	0,43	0,72	0,75	0,05	0,32	0,38
18	Estar acostado	0,41	0,07	0,5	0,48	0,28	0,17	0,2	0,83
Dataset		**Random Forest con 5 actividades**				**KNN con 5 actividades**			
ID Actividad	Nombre Actividad	Dataset 9 (4)	Dataset 16	Dataset 20	Dataset 5	Dataset 9 (4)	Dataset 16	Dataset 20	Dataset 5
6	Caminando	1	0,33	0,94	0,62	0,99	0,41	0,52	0,54
9	Corriendo	NA	0	0,74	1	NA	0	0,85	0,74
16	Estar sentado	0,29	0,87	0,66	0,81	0,55	0,8	0,64	0,73
17	Estar de pie	1	0,09	0,33	0,89	0,74	0,05	0,38	0,36
18	Estar acostado	0,37	0,11	0,23	0,95	0,27	0,21	0,24	0,77

Tabla 29. Accuracy por actividades de los modelos de SL.

- Evaluación de modelos externos en SL.

Los modelos de SL que clasificaron actividades de datasets y/o información con la que no habían sido previamente entrenados presentaron un mejor rendimiento que los modelos de BL.

Los resultados con el algoritmo ARF superaron a los obtenidos en KNN, aunque no presentaron una diferencia tan grande. La mejor métrica fue Precision, que se alcanzó tanto en el Dataset 9 como en el 16, clasificando 4 y 5 actividades, respectivamente en ARF. No obstante, es importante destacar que el Accuracy en el Dataset 9, al clasificar 4 actividades, aún no alcanza los valores del criterio de éxito.

Respecto al análisis por clases. Se aprecia que las actividades de subir y bajar escaleras tienen un mal rendimiento. Esto se debe a dos factores, el primero, es que las actividades de subir y bajar escaleras no contienen los valores necesarios para estar al nivel de las demás actividades y/o el segundo factor es que los algoritmos de SL con su memoria limitada no logran almacenar suficiente información de estas actividades para la clasificación. De igual manera en los algoritmos de BL presentan bajos rendimientos con estas actividades.

Otro aspecto para considerar es que las ubicaciones del dataset 9 los sensores se encuentran en la cintura, mientras que los modelos están entrenados con la ubicación más cercana a la cadera; lo cual también pudo generar que el rendimiento no fuera el que se esperaba.

- Evaluación de modelos internos en SL.

Los modelos de SL con los datos con los que fue entrenado presentaron un decrecimiento del rendimiento a comparación de cuando no conoce la información a clasificar. Esto se debe a la memoria limitada que tiene SL, que no logra almacenar toda la información de los datasets entrenados previamente.

6.2.1.3 Evaluación del escenario 3: Clasificación incremental de SL

Para el escenario 3, clasificación incremental de SL solo se utilizó el mejor algoritmo de SL que fue ARF, como se puede apreciar en la *Tabla 28*.

En el momento de hacer uso del aprendizaje incremental de SL tanto con los datasets externos e internos; presentaron una mejora bastante considerable a comparación de los resultados anteriores.

La diferencia principal en SL entre clasificación y clasificación incremental es que mientras en la primera forma SL está clasificando qué actividad puede ser teniendo en cuenta una serie de características que recibe. En la segunda forma, además de clasificar, da la opción de aprender, es decir, se va retroalimentando y entrenando constantemente. A continuación, se presentan las pruebas en un entorno de entrenamiento y prueba con los datasets externos e internos.

La diferencia principal en aprendizaje supervisado (SL) entre clasificación y clasificación incremental radica en que, mientras en la primera modalidad el SL se encarga de clasificar las actividades teniendo en cuenta una serie de datos, al igual que en el aprendizaje básico

(BL), la segunda modalidad no solo clasifica, sino que también ofrece la opción de aprender. En otras palabras, se retroalimenta y entrena constantemente mediante el uso del método partial_fit. Este método permite que el algoritmo se retroalimente con las etiquetas que recibe, depurando la información antigua a medida que se ajusta a las características específicas de estos algoritmos. A continuación, se presentan las pruebas realizadas en un entorno de entrenamiento y prueba utilizando conjuntos de datos tanto externos como internos.

Datasets		**5 actividades**			**7 actividades**		
		Accuracy	**Precision**	**F1-score**	**Accuracy**	**Precision**	**F1-score**
Externos	Dataset 9	0,92	0,92	0,92	0,87	0,88	0,87
	Dataset 16	0,81	0,56	0,32	0,79	0,3	0,22
Internos	Dataset 5	0,87	0,89	0,87	0,77	0,8	0,75
	Dataset 20	0,69	0,69	0,68	0,51	0,49	0,47

Tabla 30. SL con su función incremental.

Stream Learning Incremental					
Dataset		**Random Forest con 7 actividades**			
ID Actividad	Nombre Actividad	Dataset 9 (6)	Dataset 16	Dataset 20	Dataset 5
6	Caminando	0,87	0,39	0,87	0,96
9	Corriendo	NA	0	0,85	0,98
13	Subir escaleras	0,85	0	0,09	0,46
14	Bajar escaleras	0,87	0	0,09	0,50
16	Estar sentado	0,74	0,97	0,73	0,81
17	Estar de pie	0,89	0	0,51	0,73
18	Estar acostado	0,95	0,07	0,44	0,71
Dataset		**Random Forest con 5 actividades**			
ID Actividad	Nombre Actividad	Dataset 9 (4)	Dataset 16	Dataset 20	Dataset 5
6	Caminando	1	0,49	0.95	0,97
9	Corriendo	NA	0	0.85	0,97
16	Estar sentado	0,81	0,97	0.62	0,9
17	Estar de pie	0,94	0,02	0.62	0,75
18	Estar acostado	0,95	0,06	0.43	0,73

Tabla 31. Accuracy por actividades de los modelos de SL incremental.

En la Tabla 30, se visualizan resultados positivos en la mayoría de las métricas. Sin embargo, hay algunas apreciaciones que se tuvieron en cuenta.

- Al nivel de actividades al igual que en la clasificación de SL, la clasificación incremental de SL sigue presentando mejores resultados con los datos externos que con los internos. Esto se debe por la memoria limitada que posee SL. Se rectifica nuevamente que el dataset 16, tiene una alta falta de confiabilidad con sus valores, inclusive en la clasificación incremental de SL aún presenta valores bajos a comparación del resto.

Los modelos BL no tienen la funcionalidad de entrenarse incrementalmente, dado que esto es una de las ventajas únicas que tienen los modelos de SL. En la imagen 17 se puede apreciar cómo es el aprendizaje incremental de SL en el dataset externo número 9 con el algoritmo de RF, los demás gráficos se pueden encontrar en el *Anexo F - Resultados Stream VS Batch.*

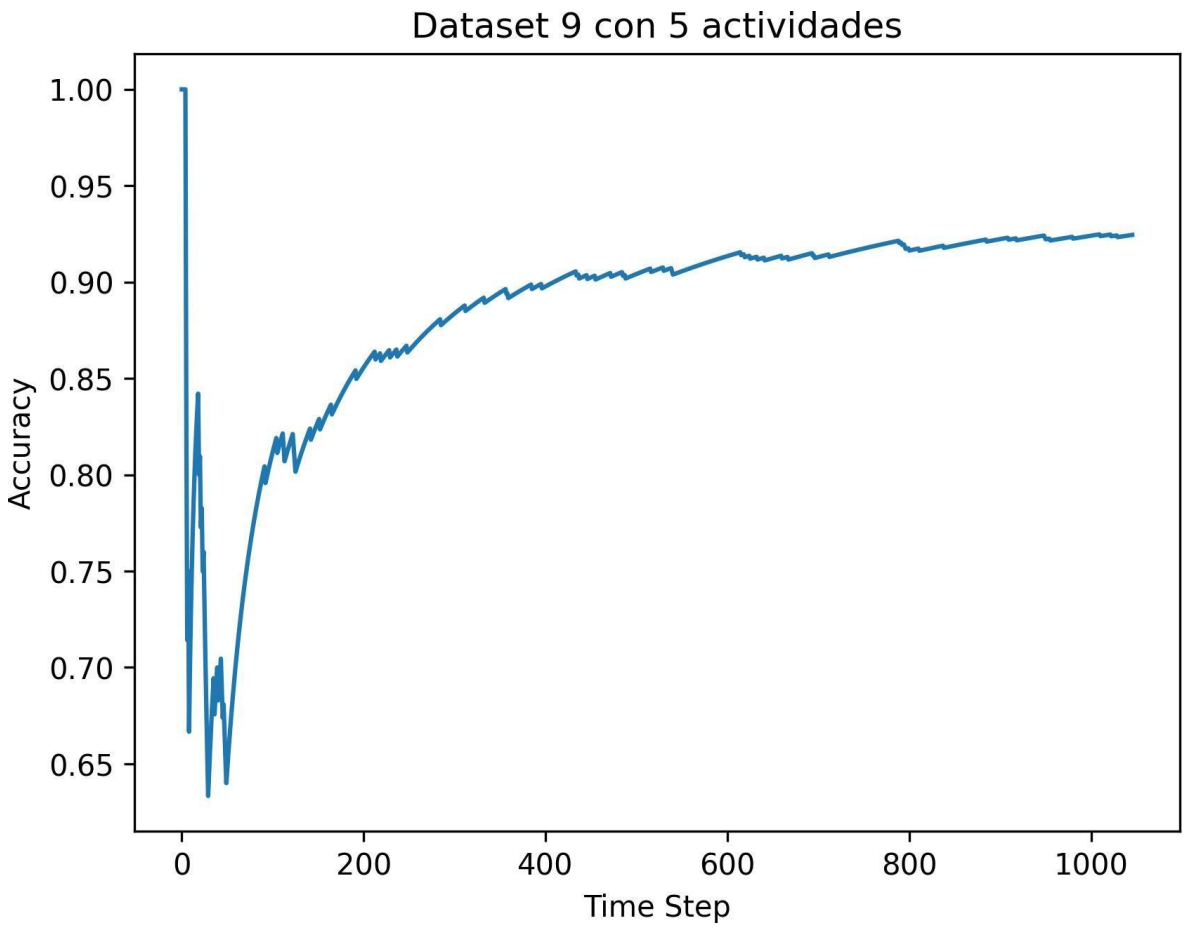

Imagen 8. Aprendizaje incremental Stream Learning.

Debido a que la clasificación incremental de SL está clasificando en cuanto al flujo continuo de datos, es posible tener esta gráfica. Como se puede apreciar en la Imagen 17, el aprendizaje incremental de SL empezó su aprendizaje con grandes picos sin embargo decayó drásticamente debido a encontrarse con datos desconocidos, sin embargo a medida que el modelo va aprendiendo los patrones de estos nuevos datos vuelve a recuperar una exactitud considerable, se observa en la gráfica a partir de aproximadamente las primeras 50 muestras, empieza a tener picos de subida y bajada cada vez más pequeños con una tendencia positiva, esto se debe a encontrarse con diferentes actividades. No obstante, al no ser demasiadas actividades se recupera de estas caídas e incluso supera la exactitud anterior, esto es muy

diferente a como se puede observar en las imágenes 13, 14 y 15 cuando se contaba con muchas actividades debido a que en esos casos no lograba estabilizarse positivamente al cambiar de actividad de manera tan repentina y por lo tanto la tendencia era negativa.

Se resaltó también que en el primer dato desconocido que recibió, la gráfica no bajó a cero esto debido al entrenamiento previo que tenía.

6.3Análisis final: Comparación SL y BL

Continuando el primer análisis realizado al final del capítulo 5, se generaron las conclusiones y aspectos a tener en cuenta en el momento en que se quiera elegir el tipo de algoritmo a utilizar SL o BL. A continuación, la comparación final de SL y BL:

- SL tiene una tendencia a tener modelos más livianos y consumir menos RAM que BL.
- El tiempo de entrenamiento de los modelos de BL es menor al tiempo de SL.
- BL tiene muchísima más documentación que SL.
- En un escenario donde se necesita entrenar el modelo con un gran número de actividades y se tiene la preocupación de que se tenga un desbalance de clases, se recomienda a BL sobre SL. Así mismo, si las actividades a clasificar contienen valores ya familiarizados por el modelo, se sugiere dar preferencia a BL en lugar de SL.
- En el caso de este trabajo de grado, los modelos de SL se les facilitó la clasificación de los modelos externos a diferencia de BL. Los modelos de BL no lograron predecir correctamente valores de actividades con las que no fueron entrenados, debido al sobreajuste, esto no sucede en SL gracias a que no almacena toda la información.
- En un escenario donde las actividades no sean demasiadas, y/o se tenga un balance de clases, por el peso del modelo y la RAM, se recomienda a SL sobre BL.
- La función incremental de SL es un abrebocas a lo que significan los algoritmos de SL en la actualidad. En un escenario donde sea necesario estar retroalimentando el algoritmo constantemente en ADL’s, SL se sobrepone a BL, tanto por rendimiento, peso del modelo, uso de RAM y la escalabilidad que este permite.

6.3.1 Modelos aprobados

Para la selección del modelo aprobado, primero se hizo entrega de consideraciones a tener en cuenta en una correcta selección entre estos dos tipos de algoritmos. A continuación, las consideraciones para saber cuál modelo elegir entre BL y SL:

- Si el modelo quiere seguir siendo utilizado para evaluarse con datos que ya conoce o similar y/o no se tiene un requerimiento con datos desconocidos e incluso clasificar una cantidad considerable de actividades, definitivamente la elección son los modelos de BL. En casos en el que se cuente con buenos recursos computacionales la mejor opción es el modelo de RF, pero si se busca un menor consumo de memoria ROM y RAM sacrificando un poco el rendimiento, la mejor opción es el modelo DT.

- Si el modelo tiene como objetivo la clasificación en un entorno donde se va a enfrentar a datos desconocidos, y se asegura que el modelo fue entrenado correctamente sin una gran cantidad de clases, se recomienda escoger SL con su función incremental.

- Si el modelo tiene requerimientos de ser un modelo ligero, se puede pensar de inmediato en recomendar SL. Sin embargo, si hay una gran cantidad de actividades para entrenar y se van a clasificar actividades del mismo modelo entrenado o similares, se puede considerar BL, la pregunta que surgió aquí es, qué algoritmo para BL, ¿RF o DT? Para esto también entran en consideraciones qué recursos computacionales tiene la persona interesada en este estudio.

    - Si se tienen recursos computacionales bastante buenos y su interés es tener una mejor precisión, se recomendaría en BL el uso de RF.

    - Si no se poseen suficientes recursos computacionales o le interesa más la velocidad de la predicción, se recomendaría en BL el uso de DT.

- BL es una excelente opción para intereses de casos de negocio, donde no vayan a tener muchas variaciones en sus datos a futuro. Mientras que en casos de negocio donde se prevea que la cantidad de datos no va a ser la misma siempre con los datos que fueron entrenados y van a tener cierta cantidad de datos nuevos cada tanto tiempo, se recomienda SL.

Para el interés del presente trabajo de grado se eligió al modelo de clasificación SL, debido al manejo de los recursos de memoria RAM y de su precisión. No se implementó el modelo de clasificación incremental de SL, debido a que traía más consideraciones que no van a fin con los objetivos de este trabajo de grado.

Debido a que se encamina más a dejar un prototipo para los trabajos futuros debido al potencial tan grande de estas tecnologías, se entregan las principales diferencias entre BL y SL con las recomendaciones planteadas en base del presente estudio. Con esto se cumple el objetivo específico dos.

Finalizando este capítulo se define que el mejor modelo bajo los intereses del trabajo de grado va a ser el modelo de SL con el algoritmo de ARF entrenado con 5 actividades, con esto en mente se da como finalizado el objetivo específico número dos.

6.3 Revisión del proceso y determinación de próximos pasos

Cabe recordar que el ciclo de vida de la metodología CRISP-DM no es lineal, así que, aunque no se haya mencionado a lo largo de la monografía todas las iteraciones que se hayan realizado, se da por hecho que el proceso descrito en cada fase de la metodología es al que se ha llegado luego de realizar las iteraciones que hayan sido necesarias. De esta manera, la revisión del proceso involucró reiteradas tareas de modelado y preparación de datos para finalmente describir los resultados obtenidos en la presente monografía. Las revisiones hechas en esta tarea dieron lugar a la ejecución de un paso vital como lo fue la inclusión de la clasificación con enfoque de datasets externos e internos, así como la inclusión del aprendizaje incremental de SL.

Se consideró después de analizar cada una de las fases del presente proyecto, así mismo como el recorrido que se tuvo en el análisis de cada modelo de SL y BL, cada escenario planteado, cada prueba y se llegó a la conclusión de que el presente trabajo de grado permitió obtener un dataset unificado con todos los datasets posibles de ADL's (en el año que se realizó la búsqueda). Se realizó la comparación y se dan recomendaciones de algoritmos de BL y SL en todos los casos que se lograron plantear en el tiempo del trabajo de grado. Por último, como se aprecia en el siguiente capítulo, se demostró que sí es factible la creación de una primera versión del sistema móvil donde se dejó el modelo de clasificación de SL. A continuación, la última fase del proyecto, donde se hizo cumplimiento al objetivo específico número 3 y con él, al objetivo general del presente trabajo de grado.

Capítulo 7.

Fase 6: Despliegue

En esta fase se usaron los resultados obtenidos a lo largo de todas las fases anteriores del proyecto para desplegar la primera versión del sistema móvil denominado "*Jaida*". Las tareas de esta fase son las siguientes:

- Plan de despliegue, monitoreo y mantenimiento.
    - Principios del diseño de aplicaciones móviles.
    - Interacción con el sistema móvil.
- Informe final.
    - Contribuciones.
    - Conclusiones.
    - Trabajos futuros.

7.1 Plan de despliegue, monitoreo y mantenimiento

7.1.1 Principios del diseño de aplicaciones móviles

La primera versión del sistema móvil Jaida se desarrolló en Flutter con un backend en Flask. Además, el sistema móvil se desplegó en una instancia *micro* de la tecnología EC2 de Amazon Web Services (AWS). Este sistema móvil siguió los principios de diseño de una aplicación móvil de Google de manera efectiva y cumplió con los criterios clave para una navegación y exploración exitosas. A continuación, algunos de los criterios considerados:

- Mostrar el valor de la aplicación de inmediato: Jaida presenta claramente sus características principales y nuevas funciones en contextos apropiados, guiando a los usuarios hacia lo que necesitan con un llamado a la acción evidente, como "*Subir CSV*" y "*Generar Data*".
- Navegación intuitiva: La aplicación permite a los usuarios retroceder un paso en lugar de regresar al inicio en caso de necesitar correcciones o cambios, evitando pérdida de datos y reducir posibles frustraciones del usuario al brindar un proceso más fluido
- UX-B1: La app cumple con las pautas de diseño de Android y utiliza patrones e íconos de IU comunes.
- PS-V2: La app muestra texto y bloques de texto de forma aceptable.

Otros criterios importantes que fueron cumplidos a cabalidad fueron: no solicitar permisos intrusivos, ser compatible con la navegación estándar del sistema mediante el botón "Atrás", incluir etiquetas de texto para íconos y la facilidad de uso.

Imagen 9. Imagotipo del sistema móvil.

En la *Imagen 9*, el imagotipo para el aplicativo móvil integró de manera hábil los elementos visuales y el nombre de la marca que se realizó en la primera versión del sistema móvil: Jaida, el cual es la fusión de los nombres de los autores del presente trabajo de grado Jaime y Daniel. Así mismo, al estar orientado a la salud, se diseñó con un corazón, el cual lleva en su interior un pulsador. Esta combinación de elementos crea una conexión emocional de bienestar y vitalidad, la cual invita a permanecer en un entorno acogedor y familiar.

7.1.2 Interacción con el sistema móvil

En la definición de la primera versión del sistema móvil Jaida, se tomaron factores tales como el interés y el alcance conceptual dado por los conocimientos de los autores de este trabajo de grado, así como también, se tuvo en cuenta las limitaciones de tiempo para entregarlo.

Con ello en mente, Jaida representa un aplicativo móvil que con su apk puede ser abierto en cualquier dispositivo móvil. AWS permitió tener las políticas mínimas de seguridad en el servidor donde se encuentra corriendo. También funciona con una interfaz específica que cumple con los criterios básicos del “principio de diseños de aplicativos móviles". [74]

La interacción con un Backend se diseñó en Flask debido a la compatibilidad que tiene con el lenguaje de programación Python y finalmente se implementó en Flutter, para tener la interacción del aplicativo móvil.

Toda la documentación y el código relacionados con el despliegue se pueden encontrar en la carpeta “Aplicativo móvil Jaida” del repositorio de Github, del *Anexo G*. Que se conforma del código de Flutter, el código de Flask y por último el apk.

En profundización a lo anterior Jaida cumple con las siguientes tareas:

- Los usuarios tienen la facilidad de ensayar el sistema móvil con un dataset precargado y/o subir sus propios datasets que cumplan las características necesarias para clasificar actividades mediante su interfaz.

- Los usuarios pueden utilizar una primera versión de prueba y subir un archivo CSV, el cual solamente deberá cumplir con los requisitos para ser procesado. Ya con ello, se podrá simular esos datos para clasificar las actividades de la vida diaria elegidas en el anterior capítulo y/o que pueden generar valores de actividades aleatoriamente con un dataset ya precargado por defecto, haciendo uso de un simple botón.

Es importante aclarar que el formato del archivo CSV, no difiere y/o no tiene importancia el número de filas pero los nombres de las columnas y su posición debe ser la mismo para que el sistema lo reconozca, el cual se puede apreciar en la *Tabla 29*.

#	Nombre columna	Tipo de dato
3	X_Acc	Float
4	Y_Acc	Float
5	Z_Acc	Float
6	X_Gyro	Float
7	Y_Gyro	Float
8	Z_Gyro	Float
9	Code	Int

Tabla 32: Requisitos de Información para datasets de prueba.

Estas columnas se diseñaron para realizar pruebas con el modelo, así mismo, para que el archivo *app_características.py* (que se encuentra en la carpeta de Flask del aplicativo móvil en *Github* del *Anexo G*) pueda procesar y servir como puente para realizar el proceso de extracción de características y procesar 500 muestras en una sola fila de información. Es importante destacar que, para el propósito de modelo, la segmentación de datasets de prueba subidas al sistema móvil Jaida no es necesario, sin embargo, se recomienda que se pueda segmentar previamente a hacer uso del sistema.

El proceso de extracción de características se llevó a cabo para que los archivos CSV obtuvieran la estructura esperada; esta información se puede sintetizar a partir de datos de acelerómetro y giroscopio que contienen actividades etiquetadas. Para un mejor entendimiento de dicha información se proporcionó un dataset de prueba de ejemplo en el repositorio de *Github*. A continuación, se presenta el flujo que se tiene para la transformación de los datos:

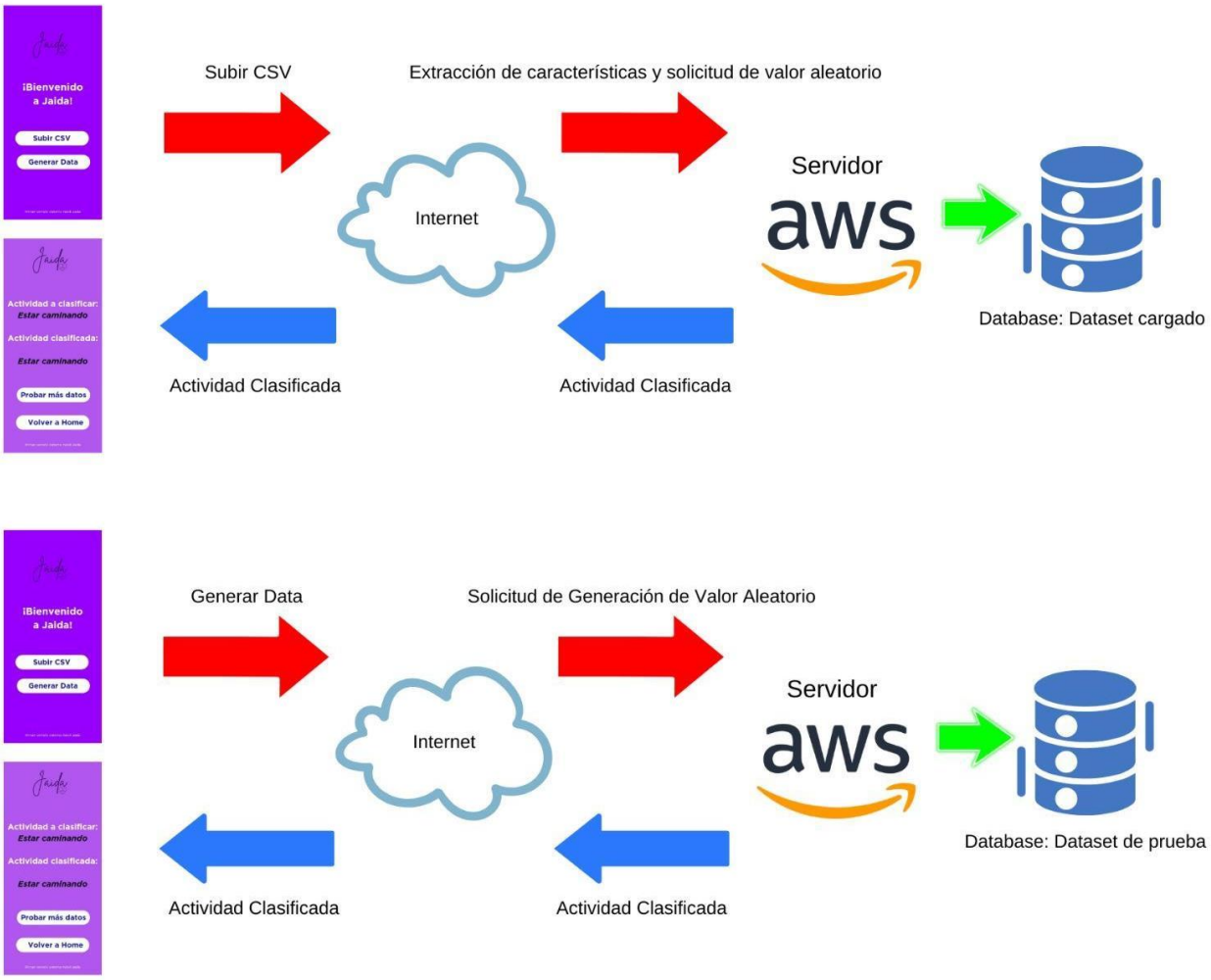

Imagen 10 a y b. Flujo de las funcionalidades del aplicativo móvil Jaida.

Los usuarios pueden identificar fácilmente el nombre del sistema móvil Jaida, así mismo como sus dos opciones habilitadas para subir un archivo CSV, y generar datos aleatorios con los cuales hacer predicciones en un dataset cargado por defecto.

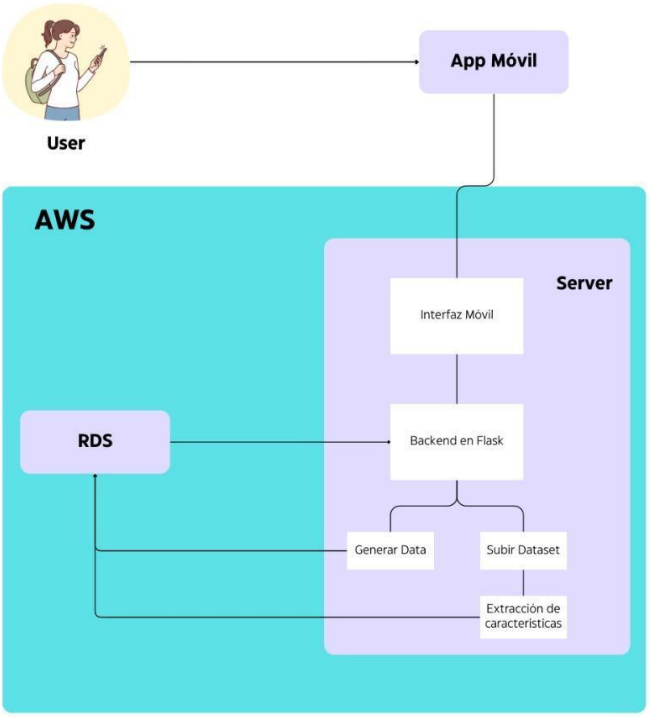

Imagen 11. Arquitectura de despliegue del aplicativo móvil Jaida.

En la Imagen 12, se puede visualizar el proceso que se tiene con la interacción del servidor con el aplicativo móvil en cuanto al almacenamiento de más datasets de prueba que se quieran realizar en el aplicativo móvil, para seguir probando el modelo.

Imagen 12. Vista 1: Home del sistema móvil.

- Los usuarios una vez subidos el dataset, podrán obtener una selección de la actividad a clasificar y la actividad clasificada.

- Que los usuarios puedan identificar fácilmente qué actividad debe clasificar el modelo y qué actividad termina prediciendo en esta primera versión del sistema móvil Jaida.

Con esta segunda vista del sistema móvil el usuario puede identificar fácilmente si la predicción realizada coincide con la esperada, así mismo el usuario tiene la oportunidad de regresar al inicio para seleccionar otro dataset y/o si quiere generar más valores aleatorizados para seguir haciendo uso del modelo y del sistema móvil Jaida.

Imagen 13. Vista 2: Resultados de predicción en el sistema móvil.

Con esta primera versión del sistema móvil Jaida, se cumple el objetivo general y tercer objetivo específico del presente trabajo de grado.

Así mismo el usuario tiene la posibilidad de hacer uso del aplicativo móvil y generar los datos aleatorios que desee del archivo CSV que suba, y analizar de la mejor manera sus resultados, igualmente, este código queda en un repositorio de GitHub, donde se podrá continuar las pruebas y/o mejorar esta primera versión del aplicativo móvil.

7.2 Informe Final

La metodología CRISP-DM brindó flexibilidad para elegir realizar un reporte, una presentación, un resumen y/o artículo, o cualquier documento investigativo para presentar los resultados del proyecto de ciencia de datos. Esta decisión se tomó teniendo en cuenta la audiencia del informe final.

En este orden de ideas, esta monografía aprovecha el espacio para presentar las contribuciones, conclusiones y trabajos futuros.

7.2.1 Contribuciones

Las contribuciones directas del trabajo de grado son las siguientes:

1. La mayor contribución consiste en el cumplimiento de su objetivo general, el cual es la obtención de un sistema móvil para la clasificación de actividades de la vida diaria haciendo uso de algoritmos de Stream Learning.

2. Se proporcionan tres datasets. El primero, denominado Dataset_Unico.csv, incluye la unión de los 15 datasets originales. El segundo, llamado Dataset_Definitivo.csv, comprende la unión de los datasets después de la extracción de características. El tercer dataset, conocido como Dataset_sin_16.csv, se creó excluyendo el dataset 16 y se utilizó en la generación de los modelos descritos en el Capítulo 6. Todos estos datasets están disponibles en la plataforma Kaggle. En el *Anexo J.*

3. Se entregan 5 modelos *.pkl*, los modelos de 5 y 7 actividades para BL, los modelos 5 y 7 actividades para SL y por último el modelo que se utiliza en el sistema móvil para predecir actividades. Todos estos modelos están disponibles también en Kaggle, en el *Anexo J*.

4. Un clasificador basado en algoritmos de SL con enfoque a actividades de la vida diaria, que logra brindar una precisión de predicción del 70%, entrenado con 5 actividades. Además, se provee la flexibilidad de generar más modelos en base a los datasets entregados y las múltiples combinaciones y pruebas que se pueden realizar. Así mismo se dejan otros modelos teniendo en cuenta que el trabajo es escalable para la realización de más pruebas.

5. Una comparación entre algoritmos de BL y SL en el contexto de ADLs lo cual en la primera búsqueda que se efectuó en el planteamiento del estado del arte no se encontró un registro de una comparación de este tipo de algoritmos con un enfoque de ADL. Esto resalta la singularidad y relevancia de esta investigación.

6. Una primera versión de un sistema móvil HAR llamado “Jaida”, que clasifica actividades de la vida diaria haciendo uso de algoritmos de SL. Lo cual en la primera búsqueda que se efectuó en el planteamiento del estado del arte y en el inicio de las dos primeras fases de la metodología CRISP-DM, no se encontró un registro de un sistema móvil que clasificara ADL haciendo uso de algoritmos de SL. Este sistema

móvil se puede encontrar bajo el nombre de “Aplicativo móvil Jaida” del repositorio de GitHub, del *Anexo G*.

7. En referente al artículo comprometido en el Anteproyecto:

   Debido a limitaciones de tiempo y recursos, no se completó el proceso de envío y revisión de dicho artículo a una revista científica o conferencia. Esta holgura fue en gran parte por la complejidad del tema y las restricciones de tiempo en el desarrollo de la tesis.

   Sin embargo, la entrega del artículo posterior a la presentación de esta tesis se plantea como un objetivo adicional. La intención es someterlo a revisión y, potencialmente, buscar su publicación en un futuro cercano. Esto se debe a que se considera que los resultados y hallazgos de esta investigación tienen relevancia y valor significativos para la comunidad académica.

   Una vez publicado, se solicita el favor al comité evaluador de poder editar esta parte de entrega, con la actualización de la publicación del artículo.

8. Cuadernos de Python donde se aprecian diferentes tipos de funciones para la preparación de datos, entre ellos se destaca: el análisis a nivel de código y las automatizaciones en cada uno de los diferentes tipos de datasets de actividades de la vida diaria; dataset con diferentes tipos de unidades, frecuencia, así mismo como diferentes tipos de datos y sobre todo presentaciones; funciones que permiten transformar los datos de un tipo de archivo a otro; búsquedas automatizadas, porque cada dataset cuenta con un proceso independiente para generar un dataset de toda la información recolectada; cuadernos donde se realizó el resampling, normalización, segmentación, extracción de características, y todo el análisis pertinente para poder cumplir la unión del dataset; entrenamientos y pruebas realizadas para los modelos; el uso de algoritmos; la integración de las librerías de Scikit-Learn y Scikit-Multiflow; Y, por último la integración de todo este modelo en código obteniendo la primera versión del sistema móvil Jaida.

9. Se deja también un cuaderno de Python en el repositorio de GitHub, denominado “Parte_12_Prueba_incremental_usando_las_etiquetas”, que son ejemplos diseñados con el fin de poder automatizar el backend en trabajos futuros en el momento que deseen tener un primer acercamiento con SL y su aprendizaje incremental.

Como contribución indirecta se tienen:

1. Se destaca un enfoque que no se encontraba al inicio de esta investigación, que implica el análisis y uso de algoritmos de SL en actividades de la vida diaria. Este enfoque potenciará la aparición de nuevas investigaciones, tanto dentro como fuera de la Universidad del Cauca, relacionadas con la interacción de estos algoritmos de SL en el campo de la salud.

7.2.2 Conclusiones

1. Una de las necesidades de realizar la clasificación de actividades de la vida diaria está soportada en el reciente interés de la comunidad científica que ha ido creciendo en el tópico del impacto que tiene la ciencia de datos en el área de la salud y el comportamiento habitual que tienen las personas en su día a día, preguntas como ¿Qué actividades se realizan con más frecuencia?, ¿Qué herramientas se pueden tener para identificar y/o acompañar a las personas en su diario vivir? resultaron de interés para llevar a cabo la propuesta de una clasificación de actividades más comunes efectuadas por las personas, clasificadas por algoritmos de SL dentro de un sistema móvil. Sin duda alguna, las contribuciones aquí aportadas servirán de base a futuras investigaciones no solamente con algoritmos de SL en actividades de la vida diaria, si no su extrapolación a demás áreas de la salud.

2. La función incremental en Stream Learning es esencial para la adaptación a cambios continuos en situaciones y proyectos a lo largo del tiempo. Su capacidad de permitir una rápida adaptación de algoritmos sin requerir entrenamientos desde cero optimiza la asignación de recursos en organizaciones. Este enfoque es particularmente valioso en situaciones donde se necesita centrarse en tareas estratégicas y donde los modelos previamente solo clasificaban sin retroalimentación. En este trabajo de grado, se ha demostrado que la función incremental mejora significativamente los resultados de los algoritmos de Stream Learning, lo que subraya su importancia en la mejora del rendimiento y la eficacia en contextos de cambio constante. Definitivamente es una línea de investigación con gran futuro.

3. Se concluye que los algoritmos de BL pueden soportar perfectamente una cantidad mínima de datos con pocas o demasiadas clases, no obstante, en el momento que los requerimientos del modelo tengan un cambio a áreas desconocidas o que se incremente el flujo de alimentación de estos datos, van a tener un rendimiento que no será óptimo. Caso contrario sucede con SL, que podrá tener un mejor acoplamiento a los datos nuevos, así mismo el entrenamiento, que puede ser con una gran cantidad de datos. Sin embargo, su limitada memoria donde se guardan las clases representa una posible contraparte cuando no exista una buena comprensión de los objetivos del negocio que planee aplicar estos algoritmos.

4. Para considerar de nuevo la unión de más datasets ya sea en el tópico de ADL o de algún otro campo del área de la salud, se deben de tener en cuenta: El origen de los datos, las unidades, la confiabilidad del dataset, el ordenamiento de los datos, una fecha de publicación reciente, la manera de etiquetado de los datos. Adicionalmente se debe tener presente que para la preparación final se sigan haciendo pruebas referentes a los datos atípicos; esto para evitar algún inconveniente en el momento de tener un dataset unificado y empezar a trabajar con este, debido a que, si no se analiza correctamente la información individual, una vez junta, puede generar resultados no esperados. Sin embargo, si se realiza bien la unión, se pueden dar más valores juntos que separados de un mismo tópico.

5. El seguimiento de la metodología CRISP-DM guió con satisfacción la realización de este trabajo de grado, cada una de las tareas que se formularon en cada una de las fases son coherentes y se sustentan en lo necesario para trabajos de ciencias de datos, por este motivo, la metodología CRISP-DM se sostiene y se verifica que es la mejor metodología para trabajos de ciencia de datos.

7.2.3 Trabajos Futuros

1. Uno de los trabajos más demandantes en cuestión de tiempo fue la unión de todos los datasets encontrados de actividades de la vida diaria, no obstante para trabajos futuros resulta interesante visualizar que en el momento de la segmentación y organización de datos se considere una agrupación y organización dando prioridad al dataset y con esto mantener la organización de la toma que tuvieron estos datos, para poder generar un nuevo modelo, y evaluar si el orden de la toma de los datos puede mejorar el rendimiento de los algoritmos de SL y BL.

2. Es necesario experimentar estos modelos de SL y también de BL en un aplicativo móvil que opere en tiempo real, esto con el fin de seguir esta línea de comparación entre estos dos algoritmos y dar una mejor comprensión del significado de tener un modelo de SL para clasificación de actividades de la vida diaria.

3. Efectuar una continuación de la primera versión del sistema móvil Jaida, en un entorno real, donde el usuario al momento de obtener la predicción de la actividad realizada pueda etiquetarla, de esta manera se permitiría la construcción de un nuevo dataset a raíz de las clasificaciones generadas.

4. Hacer pruebas con diferentes bibliotecas de algoritmos de SL como lo pueden ser *River*, *Creme* y otras bibliotecas que puedan surgir en un futuro.

5. Realizar un análisis aún más exhaustivo de las actividades de cada uno de los datasets, con el fin de que el dataset final presente una cantidad considerablemente menor de valores atípicos, reduciendo su tamaño. Así mismo se pueda evitar un sobreajuste con la hipótesis de que gracias a esto pueden mejorar los resultados finales. Con estas reducciones y análisis más significativos, se podría seleccionar un nuevo catálogo de actividades para modelo de predicción actualizado.

6. Sería muy interesante y tendría un aporte significativo, realizar un estudio en la variación de los hiper parámetros de los algoritmos de SL y BL, con el fin de encontrar modelos más eficaces y eficientes. También se recomienda revisar la posibilidad de editar el código fuente de los algoritmos en caso de ser necesario.

7. En un futuro trabajo es importante incluir el contexto de las actividades como puede ser la ubicación, la hora del día, el tiempo del año, características personales como la altura, edad y diferentes factores que contextualizan de manera más detallada las diferentes actividades con el fin de obtener un dataset más robusto.

8. Evaluar el impacto energético que tiene el uso del sistema móvil Jaida para los Smartphones o wearables, haciendo variaciones de los modelos de SL y BL.

9. Generar modelos nuevos a partir de una revisión minuciosa de qué valor tiene cada dataset dentro del dataset definitivo, es decir, empezar a retirar dataset por dataset y evaluarlo en el dataset unificado, con el fin de entender mejor los datos; para el propósito de este proyecto y por cuestión de tiempo estas pruebas solo se realizaron retirando y evaluando el dataset 16, que era el dataset con mayor impacto a nivel de filas y actividades dentro del dataset definitivo, generando 3 tipos de modelos; no obstante sería interesante hacer estas variaciones para todos los datasets.

10. Hacer uso de modelos Llama2 de meta, que son modelos generativos, sería interesante extender este campo a las nuevas líneas de investigación de inteligencias artificiales, donde se tienen y requieren de instancias GPU más potentes, como recomendación se nombra a los servidores Runpod, para encontrar mejoras en la predicción y en la experiencia de usuario.

11. Creación de aplicación móvil haciendo usó de modelos Llama2, que permitan la interacción entre el usuario y un Chatbot personalizado que esté al tanto de las actividades de la vida diaria del usuario.

12. Para el tipo de algoritmos de SL, se seleccionaron los algoritmos de clasificación, sin embargo, en las búsquedas realizadas se encontraron redes neuronales complejas las cuales se podrían ejecutar y probar su rendimiento.

13. Debido a los recursos computacionales limitados no se lograron realizar pruebas con los algoritmos de SVM, por lo tanto, sería interesante ver los resultados que este puede aportar si se cuenta con los recursos necesarios.

Capítulo 8. Bibliografía

[1] J. Chen, Y. Sun, y S. Sun, «Improving Human Activity Recognition Performance by Data Fusion and Feature Engineering», Sensors, vol. 21, n.º 3, Art. n.º 3, ene. 2021, doi: 10.3390/s21030692.

[2] «What are Activities of Daily Living (ADLs)? - 2022 - Robinhood». https://learn.robinhood.com/articles/1QcOUBlwbkasyMDmIVM0Bt/what-are-activities-of-daily-living-adls/ (accedido 24 de junio de 2022).

[3] R. Viir, A. Veraksitš, Discussion of Letter to the Editor: standardized use of the terms sedentary and sedentary behaviours š Sitting and reclining are different states, Appl. Physiol. Nutr. Metab. 37 (2012) 540–542, doi:http://dx.doi.org/ 10.1139/h2012-123.

[4] E. J et al., «Can the Output of a Learned Classification Model Monitor a Person's Functional Recovery Status Post-Total Knee Arthroplasty?», Sensors (Basel, Switzerland), vol. 22, n.o 10, dic. 2022, doi: 10.3390/s22103698.

[5] Li Q, Zhao Y, Chen Y, Yue J, Xiong Y. Developing a machine learning model to identify delirium risk in geriatric internal medicine inpatients. Eur Geriatr Med. 2022 Feb;13(1):173-183. doi: 10.1007/s41999-021-00562-9. Epub 2021 Sep 23. PMID: 34553310.

[6] C. Deshpande, G. K. Alaparthi, S. Krishnan, K. Chakravarthy Bairapareddy, A. Ramakrishna, y V. Acharya, «Comparison of Londrina activities of daily living protocol and Glittre ADL test on cardio-pulmonary response in patients with COPD: a cross-sectional study», Multidis Res Med, vol. 15, dic. 2020, doi: 10.4081/mrm.2020.694.

[7] A. A. Gulart, A. B. Munari, S. R. Klein, L. Santos da Silveira, y A. F. Mayer, «The Glittre-ADL Test Cut-Off Point to Discriminate Abnormal Functional Capacity in Patients with COPD», COPD: Journal of Chronic Obstructive Pulmonary Disease, vol. 15, n.o 1, pp. 73-78, ene. 2018, doi: 10.1080/15412555.2017.1369505.

[8] J. Cid-Ruzafa and J. Damián-Moreno, "Valoración de la discapacidad física: el indice de Barthel," Revista Española de Salud Pública, vol. 71, no. 2, pp. 127–137, Mar. 1997.

[9]B. Garcia, "¿La mayoría de los pacientes miente a sus médicos?," Saludiario, Apr. 07, 2021. https://www.saludiario.com/la-mayoria-de-los-pacientes-miente-a-sus-medicos/
(accessed Aug. 01, 2022).

[10] S. U. Park, J. H. Park, M. A. Al-masni, M. A. Al-antari, Md. Z. Uddin, and T.-S. Kim, "A Depth Camera-based Human Activity Recognition via Deep Learning Recurrent Neural Network for Health and Social Care Services," Procedia Computer Science, vol. 100, pp. 78–84, 2016, doi: 10.1016/j.procs.2016.09.126.

[11] C. Yu, Z. Xu, K. Yan, Y.-R. Chien, S.-H. Fang, and H.-C. Wu, "Noninvasive Human Activity Recognition Using Millimeter-Wave Radar," IEEE Systems Journal, vol. 16, no. 2, pp. 3036–3047, Jun. 2022, doi: 10.1109/JSYST.2022.3140546.

[12] H.-L. Le, D.-N. Nguyen, T.-H. Nguyen, and H.-N. Nguyen, "A Novel Feature Set Extraction Based on Accelerometer Sensor Data for Improving the Fall Detection System," Electronics, vol. 11, no. 7, p. 1030, Mar. 2022, doi: 10.3390/electronics11071030.

[13] C. Yin, J. Chen, X. Miao, H. Jiang, and D. Chen, “Device-Free Human Activity Recognition with Low-Resolution Infrared Array Sensor Using Long Short-Term Memory Neural Network,” Sensors, vol. 21, no. 10, p. 3551, May 2021, doi: 10.3390/s21103551.

[14] Wang Z, Yang Z, and Dong T. A review of wearable technologies for elferly care that can accurately track indoor position, recognize physical activities and monitor vital signs in real time. Sensors (Switzerland). 2017; 17(2):341.

[15] Ceron, J.D.; Lopez, D.M. Human Activity Recognition Supported on Indoor Localization and viceversa: A. Systematic Review. Stud. Health Technol. Inf. 2018, 249, 93–101.

[16] A. Reiss, D. Stricker, Creating and benchmarking a new dataset for physical activity monitoring, Proc. 5th Int. Conf. PErvasive Technol. Relat. to Assist. Environ. - PETRA ’12. (2012) 1. doi:10.1145/2413097.2413148.

[17] “Almacenamiento en la nube: Ventajas y Desventajas - Fórmate.es,” Fórmate.es, Jun. 27, 2021. https://www.formate.es/blog/consejos/almacenamiento-en-la-nube/ (accessed Aug. 07, 2022).

[18] Bifet, A., Gavalda, R., Holmes, G., Pfahringer, B.: Machine Learning for Data Streams with Practical Examples in MOA. MIT Press, Cambridge (2018).

[19] Bifet, A., Holmes, G., Kirkby, R., Pfahringer, B.: MOA: massive online analysis. J. Mach. Learn. Res. 11, 1601–1604 (2010). http://portal.acm.org/citation.cfm? id=1859903.

[20] I. Amezzane, Y. Fakhri, M. El Aroussi, and M. Bakhouya, “Online Stream Learning for Smartphone-based Human Activity Recognition: An Overview”.

[21] T. Li, Y. Wu, F. Wu, S. Mohammed, R. K. Wong, y K.-L. Ong, «Sleep pattern inference using IoT sonar monitoring and machine learning with Kennard-stone balance algorithm», Computers & Electrical Engineering, vol. 93, p. 107181, jul. 2021, doi: 10.1016/j.compeleceng.2021.107181.

[22] L. Miranda, J. Viterbo, and F. Bernardini, “Impact of Memory Control on Batch Learning in Human Activity Recognition Scenario in Comparison to Data Stream Learning,” in Advances in Soft Computing, vol. 12468, L. Martínez-Villaseñor, O. Herrera-Alcántara, H. Ponce, and F. A. Castro-Espinoza, Eds. Cham: Springer International Publishing, 2020, pp. 145–157. doi: 10.1007/978-3-030-60884-2_11.

[23]W. S. Lima, H. L. S. Bragança, and E. J. P. Souto, “NOHAR - NOvelty discrete data stream for Human Activity Recognition based on smartphones with inertial sensors,” Expert Systems with Applications, vol. 166, p. 114093, Mar. 2021, doi: 10.1016/j.eswa.2020.114093.

[24] [1] «¿Qué es el análisis exploratorio de datos? | IBM». Accedido: 18 de octubre de 2023. [En línea]. Disponible en: https://www.ibm.com/mx-es/topics/exploratory-data-analysis

[25] S. S. Saha, S. Rahman, M. J. Rasna, A. K. M. Mahfuzul Islam and M. A. Rahman Ahad, "DU-MD: An Open-Source Human Action Dataset for Ubiquitous Wearable Sensors," 2018 Joint 7th International Conference on Informatics, Electronics & Vision (ICIEV) and 2018 2nd International Conference on Imaging, Vision & Pattern Recognition (icIVPR), Kitakyushu, Japan, 2018, pp. 567-572, doi: 10.1109/ICIEV.2018.8641051.

[26] E. Casilari, J. A. Santoyo-Ramón, y J. M. Cano-García, «UMAFall: A Multisensor Dataset for the Research on Automatic Fall Detection», Procedia Computer Science, vol. 110, pp. 32-39, 2017, doi: https://doi.org/10.1016/j.procs.2017.06.110.

[27] Pires, Ivan; Garcia, Nuno M. (2022), “Raw dataset with accelerometer, gyroscope, magnetometer, location and environment data for activities without motion”, Mendeley Data, V3, doi: 10.17632/3dc7n482rt.3.

[28] Vaizman, Y., Ellis, K., and Lanckriet, G. "Recognizing Detailed Human Context In-the-Wild from Smartphones and Smartwatches". IEEE Pervasive Computing, vol. 16, no. 4, October-December 2017, pp. 62-74. doi:10.1109/MPRV.2017.3971131.

[29] Ceron, J.D.; López, D.M.; Kluge, F.; Eskofier, B.M. Framework for Simultaneous Indoor Localization, Mapping, and Human Activity Recognition in Ambient Assisted Living Scenarios. Sensors 2022, 22, 3364. https://doi.org/10.3390/s22093364.

[30] Ceron, J.D.; Kluge, F.; Küderle, A.; Eskofier, B.M.; López, D.M. Simultaneous Indoor Pedestrian Localization and House Mapping Based on Inertial Measurement Unit and Bluetooth Low-Energy Beacon Data. Sensors 2020, 20, 4742. https://doi.org/10.3390/s20174742.

[31] Ceron, J.D.; Martindale, C.F.; López, D.M.; Kluge, F.; Eskofier, B.M. Indoor Trajectory Reconstruction of Walking, Jogging, and Running Activities Based on a Foot-Mounted Inertial Pedestrian Dead-Reckoning System. Sensors 2020, 20, 651. https://doi.org/10.3390/s20030651.

[32] B. A. Kitchenham y S. G. Linkman, «DESMET : A method for evaluating Software Engineering methods and tools», 2000. [En línea]. Disponible en: https://api.semanticscholar.org/CorpusID:18828657

[33] S. Lim and J. -W. Park, "Enhancing Frequency Stability for Grid Resilience Based on Effective WPPs Power Curtailment," in IEEE Transactions on Industry Applications, doi: 10.1109/TIA.2023.3322980.

[34] T. Fu, K. Zhang, L. Zhang, S. Wang and S. Ma, "An Efficient Framework of Reference Picture Resampling (RPR) for Video Coding," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 32, no. 10, pp. 7107-7119, Oct. 2022, doi: 10.1109/TCSVT.2022.3176934.

[35] K. Pykes, «Oversampling and Undersampling», Medium. Accedido: 18 de octubre de 2023. [En línea]. Disponible en: https://towardsdatascience.com/oversampling-and-undersampling-5e2bbaf56dcf

[36] «Normalization | Machine Learning», Google for Developers. Accedido: 18 de octubre de 2023. [En línea]. Disponible en: https://developers.google.com/machine-learning/data-prep/transform/normalization

[37] Davis,Kadian and Owusu,Evans. (2016). Smartphone Dataset for Human Activity Recognition (HAR) in Ambient Assisted Living (AAL). UCI Machine Learning Repository. https://doi.org/10.24432/C5P597.

[38] Roggen,Daniel, Calatroni,Alberto, Nguyen-Dinh,Long-Van, Chavarriaga,Ricardo, and Sagha,Hesam. (2012). OPPORTUNITY Activity Recognition. UCI Machine Learning Repository. https://doi.org/10.24432/C5M027.

[39] Weiss,Gary. (2019). WISDM Smartphone and Smartwatch Activity and Biometrics Dataset . UCI Machine Learning Repository. https://doi.org/10.24432/C5HK59.

[40] Ugulino,W., Cardador,D., Vega,K., Velloso,E., Milidiu,R., and Fuks,H.. (2013). Wearable Computing: Classification of Body Postures and Movements (PUC-Rio). UCI Machine Learning Repository. https://doi.org/10.24432/C54W37.

[41] Zdemir,Ahmet and Barshan,Billur. (2018). Simulated Falls and Daily Living Activities Data Set. UCI Machine Learning Repository. https://doi.org/10.24432/C52028.

[42] Wijekoon,Anjana, Wiratunga,Nirmalie, and Cooper,Kay. (2019). MEx. UCI Machine Learning Repository. https://doi.org/10.24432/C59K6T.

[43] Barshan,Billur and Altun,Kerem. (2013). Daily and Sports Activities. UCI Machine Learning Repository. https://doi.org/10.24432/C5C59F.

[44] I. Khosravi and S. K. Alavipanah, "A random forest-based framework for crop mapping using temporal, spectral, textural and polarimetric observations," International Journal of Remote Sensing, vol. 40, no. 18, pp. 7221-7251, Sep. 2019.

[45] Vergara,Alexander. (2012). Gas Sensor Array Drift Dataset. UCI Machine Learning Repository. https://doi.org/10.24432/C5RP6W.

[46] Bruno,Barbara, Mastrogiovanni,Fulvio, and Sgorbissa,Antonio. (2014). Dataset for ADL Recognition with Wrist-worn Accelerometer. UCI Machine Learning Repository. https://doi.org/10.24432/C5PC99.

[47] Ruzzon, Marco; Carfì, Alessandro; Ishikawa, Takahiro; Mastrogiovanni, Fulvio; Murakami, Toshiyuki (2020), "A multi-sensory dataset for the activities of daily living", Mendeley Data, V2, doi: 10.17632/wjpbtgdyzm.2

[48] Reyes-Ortiz,Jorge, Anguita,Davide, Ghio,Alessandro, Oneto,Luca, and Parra,Xavier. (2012). Human Activity Recognition Using Smartphones. UCI Machine Learning Repository. https://doi.org/10.24432/C54S4K.

[49] Reyes-Ortiz,Jorge, Anguita,Davide, Oneto,Luca, and Parra,Xavier. (2015). Smartphone-Based Recognition of Human Activities and Postural Transitions. UCI Machine Learning Repository. https://doi.org/10.24432/C54G7M.

[50] xmouyang, «HARBox Dataset: Daily Activity Recognition using Smartphones,» github.com, 2023. [online]. Available: https://github.com/xmouyang/FL-Datasets-for-HAR/tree/main/datasets/HARBox. (accessed: sep 19 2023).

[51] xmouyang, «UWB Dataset: Human Movement Detection using Ultra Wide Band Modules,» github.com, 2023. [online]. Available: https://github.com/xmouyang/FL-Datasets-for-HAR/tree/main/datasets/UWB.(accessed: sep 19 2023).

[52] xmouyang, «IMU Dataset: Walking Activity Recognition using Inertial Measurement Unit Modules,» github.com, 2023. [online]. Available: https://github.com/xmouyang/FL-Datasets-for-HAR/tree/main/datasets/IMU. (accessed: sep 19 2023).

[53] S. Jeeru, A. Kumar, D. González, J. Gröli, S. Reddy, L. Reddy, «Depth camera based dataset of hand gestures,» Elsevier Inc., vol. 45, nº 108659, 2022.

[54] Leutheuser, H., Schuldhaus, D., Eskofier, B. M. (2013) Hierarchical, multi-sensor based classification of daily life activities: comparison with state-of-the-art algorithms using a benchmark dataset, PLoS ONE, 8(10), e75196.

[55] Reiss,Attila. (2012). PAMAP2 Physical Activity Monitoring. UCI Machine Learning Repository. https://doi.org/10.24432/C5NW2H.

[56] Department of Computer & Information Science, Fordham University, Bronx, NY, «WISDM: WIreless Sensor Data Mining,» 02 Dic. 2012. [online]. Available: https://www.cis.fordham.edu/wisdm/dataset.php.(accessed: sep 19 2023).

[57] M. Zhang, A. Sawchuk, «USC-HAD: A Daily Activity Dataset for Ubiquitous Activity,» Los Angeles, CA 90089-2564 - University of Southern California. [online]. Available: https://sipi.usc.edu/had/mi_ubicomp_sagaware12.pdf.(accessed: sep 19 2023).

[58] Universidad de Coruña - Facultade de Informática, «A Public Domain Dataset For Real-life Human Activity Recognition Using Smartphone Sensors,» 2011. [online]. Available: https://lbd.udc.es/research/real-life-HAR-dataset/.(accessed: sep 19 2023).

[59] Zhao, H., Liu, C., Li, C., & Wang, H. (2010). Feature extraction using wavelet entropy and band powers in brain-computer interface. In 2010 2nd International Conference on Signal Processing

[60] P. Chapman et al., “CRISP-DM 1.0 Step-by-step data mining guide,” 2000.

[61] [kk] M. Ruzzon, A. Carfì, T. Ishikawa, F. Mastrogiovanni, y T. Murakami, «A multi-sensory dataset for the activities of daily living», Data in Brief, vol. 32, p. 106122, 2020, doi: https://doi.org/10.1016/j.dib.2020.106122.

[62] IBM, «Conceptos básicos de ayuda de CRISP-DM,» 17 08 2021. [En línea]. Available: https://www.ibm.com/docs/es/spss-modeler/saas?topic=dm-crisp-help-overview.

[63] Altun, K., Barshan, B. (2010). Human Activity Recognition Using Inertial/Magnetic Sensor Units. In: Salah, A.A., Gevers, T., Sebe, N., Vinciarelli, A. (eds) Human Behavior Understanding. HBU 2010. Lecture Notes in Computer Science, vol 6219. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-14715-9_5.

[64] Khan H., Nurul A., Rafiqul I., «HSDLM: Un muestreo híbrido con un método de aprendizaje profundo para la clasificación de datos desequilibrados,» 2021. [En línea]. Available: DOI: 10.4018/IJCAC.2021100101. [Último acceso: 2023].

[65] G. Boesch, «TensorFlow Lite – Real-Time Computer Vision on Edge Devices,» 2022. [En línea]. Available: https://viso.ai/edge-ai/tensorflow-lite/. [Último acceso: 2023]

[66]R. Díaz, «The Machine Learners: El Mejor editor de código para Python,» 2023. [En línea]. Available: https://www.themachinelearners.com/editor-codigo-python/#Conclusiones. [Último acceso: 2023].

[67] Wirth, R., & Hipp, J. (2000). CRISP-DM: Towards a standard process model for data mining. IV International Conference on the Practical Applications of Knowledge Discovery and Data Mining (pp. 29-39)

[68]M. R. T. Perales, O. N. R. Montalvo, and C. A. C. Mundaca, “MODELO DE CLASIFICACIÓN DE OPINIONES SUBJETIVAS EN REDES SOCIALES,” Rev. Científica Ing. Ciencia, Tecnol. e Innovación, vol. 1, no. 1, p. 77, 2015

[69] «Selección de características en el proceso de ciencia de datos en equipos - Azure Architecture Center | Microsoft Learn». Accedido: 18 de octubre de 2023. [En línea]. Disponible en: https://learn.microsoft.com/es-es/azure/architecture/data-science-process/select-features

[70]L. Torres, «The Machine Learners: Curva ROC y AUC en Python,» 2023. [En línea]. Available: https://www.themachinelearners.com/curva-roc-vs-prec-recall/#%C2%BFQue_es_la_curva_ROC. [Último acceso: 2023].

[71] C. J. van Rijsbergen, "Information Retrieval," 2nd ed., Butterworth-Heinemann, 1979.

[72] Pandas, «pandas.DataFrame.describe,» Created using Sphinx 6.2.1, 2023. [En línea]. Available: https://pandas.pydata.org/pandas-docs/stable/reference/api/pandas.DataFrame.describe.html. [Último acceso: 2023].

[73] Pedregosa, F. and Varoquaux, G. and Gramfort, A. and Michel, V.and Thirion, B. and Grisel, O. and Blondel, M. and Prettenhofer, P. and Weiss, R. and Dubourg, V. and Vanderplas, J. and Passos, A. and Cournapeau, D. and Brucher, M. and Perrot, M. and Duchesnay, E., «Scikit-learn: Machine Learning in Python,» Google Summer of Code project, 2011. [En línea]. Available: https://scikit-learn.org/stable/about.html#citing-scikit-learn. [Último acceso: 2023].

[74] Gregory Piatetsky-Shapiro, “KDnuggets Data Mining, Analytics, Big Data, and Data Science.”

[75] Pires, I.M., Garcia, N.M., Zdravevski, E. et al. Daily motionless activities: A dataset with accelerometer, magnetometer, gyroscope, environment, and GPS data. Sci Data 9, 105 (2022). https://doi.org/10.1038/s41597-022-01213-9.

[76] «Feature Extraction Explained». Accedido: 18 de octubre de 2023. [En línea]. Disponible en: https://www.mathworks.com/discovery/feature-extraction.html.

[77] «Full batch, mini-batch, and online learning». Accedido: 18 de octubre de 2023. [En línea]. Disponible en: https://kaggle.com/code/residentmario/full-batch-mini-batch-and-online-learning.

[78] J. Montiel, J. Read, A. Bifet, y T. Abdessalem, "Scikit-Multiflow: A Multi-output Streaming Framework," Journal of Machine Learning Research, vol. 19, no. 72, pp. 1-5, 2018. [En línea]. Disponible en: http://jmlr.org/papers/v19/18-251.html.

[79] M. Kühn and K. Johnson, "Applied predictive modeling",, 2013. https://doi.org/10.1007/978-1-4614-6849-3.

[80] D. Chicco, "Ten quick tips for machine learning in computational biology", Biodata Mining, vol. 10, no. 1, 2017. https://doi.org/10.1186/s13040-017-0155-3.

[81] J. M. Moine y A. S. Haedo, «Una herramienta para la evaluación y comparación de metodologías de minería de datos», presentado en XXI Congreso Argentino de Ciencias de la Computación (Junín, 2015), 2015. Accedido: 19 de octubre de 2023. [En línea]. Disponible en: http://sedici.unlp.edu.ar/handle/10915/50428