
Proyecto 1: Análisis de datos para aplicaciones médicas

Nubia S. Garcidueñas¹, Mariana León² and Ana A. Rello³

¹ Instituto Tecnológico De Monterrey, Campus Guadalajara

Abstract— El objetivo principal del proyecto es desarrollar un sistema de clasificación de actividades físicas en tiempo real basado en datos de acelerómetro, clasificando las actividades extrayendo características estadísticas de aceleración. Se divide en diversas etapas: recolección de datos, evaluación de modelos de clasificación, optimización de modelos y clasificación en tiempo real. Inicialmente, se evalúan modelos de clasificación. Posteriormente, los modelos son optimizados mediante la selección de los mejores hiperparámetros y características. Finalmente, se implementa un prototipo en línea, donde se realizan predicciones en tiempo real a medida que se recolectan los datos. El resultado esperado es una herramienta capaz de clasificar con precisión las actividades físicas en tiempo real, lo cual podría ser útil para el seguimiento y monitoreo de la salud.

Keywords— Acelerómetro, Clasificación en Tiempo Real, Selección de Características, Optimización de Hiperparámetros, Modelos de Machine Learning, Random Forest, Clasificador MLP, SVM, K-NN, Reconocimiento de Actividades, Adquisición de Datos, Prototipo en Línea

I. INTRODUCCIÓN

El monitoreo y clasificación de actividades físicas a partir de señales de acelerómetros es un campo de creciente interés, especialmente con la expansión de los dispositivos móviles y wearables. Estos dispositivos permiten el acceso a sensores que recogen datos de aceleración en tiempo real, los cuales pueden ser procesados y utilizados para clasificar actividades.

El objetivo principal de este proyecto es desarrollar un sistema de clasificación utilizando modelos de Machine Learning. Para ello, se emplearon datos de acelerómetros recolectados durante diferentes actividades físicas realizadas por un grupo de usuarios. A través de este sistema, se busca identificar y clasificar las actividades.

El presente proyecto se basa en investigaciones previas, pero con un enfoque adicional en la optimización de los modelos y la reducción de características, lo que mejora tanto la precisión de la clasificación como la eficiencia computacional. Además, al implementar una clasificación en línea, este proyecto se acerca más a un sistema funcional que podría utilizarse en aplicaciones de salud o entrenamiento personal en tiempo real.

II. ANTECEDENTES

El campo de la clasificación de actividades físicas ha sido ampliamente explorado en la literatura debido a sus aplicaciones en áreas como la salud y la monitorización del comportamiento humano. Con la popularización de los dispositivos móviles y los sensores portátiles, se ha facilitado la recolección de datos en tiempo real sobre los movimientos y las actividades físicas de las personas.

Google Fit y Apple Health son ejemplos de plataformas comerciales que han integrado los sensores de movimiento para ofrecer un análisis del estado físico del usuario. Estas

aplicaciones rastrean actividades, proporcionando estadísticas detalladas sobre la actividad física diaria. Sin embargo, la clasificación precisa de actividades complejas sigue siendo un desafío, especialmente cuando se requieren clasificaciones en tiempo real.

Investigaciones previas han utilizado algoritmos de Machine Learning, como SVM, K-NN y Random Forest, para la clasificación de actividades físicas. Sin embargo, la mayoría de estos enfoques no abordan la optimización de hiperparámetros ni la reducción de características, lo que puede limitar la eficiencia y precisión del modelo en escenarios con limitaciones de tiempo y recursos computacionales.

III. METODOLOGÍA

Este proyecto se desarrolló en varias etapas interrelacionadas, cada una con su enfoque particular para lograr el objetivo final.

1. Recolección de Datos y Preprocesamiento de los Datos:

Se utilizó la aplicación Phyphox para recolectar las señales con una frecuencia de muestreo de 20 Hz. Los datos fueron almacenados en archivos de texto y divididos en segmentos de 0.5. Una vez recolectados los datos, se eliminaron las filas con valores NaN.

2. Evaluación de Modelos de Clasificación:

Para cada modelo, se realizó un análisis con validación cruzada utilizando 5 particiones, lo que permite estimar el rendimiento del modelo sin que los datos de prueba se filtren en el proceso de entrenamiento.

3. Optimización de Modelos y Selección de Características:

Se optimizaron los hiperparámetros de los modelos

evaluados para reducir el número de características sin perder rendimiento. Encontrando los valores óptimos para parámetros como `n_estimators` en Random Forest o `hidden_layer_sizes` en MLP.

4. Clasificación en Línea:

Finalmente, se implementó un prototipo en línea utilizando uno de los modelos optimizados en las etapas anteriores. El modelo predice la actividad en curso y muestra el resultado al usuario de manera inmediata.

IV. APLICACIÓN IMPLEMENTADA JUNTO CON LOS DATOS RECOLECTADO

a. ¿En qué consiste la aplicación?

La aplicación es un sistema de clasificación de actividades físicas en tiempo real, que utiliza datos de acelerómetros para identificar y clasificar diferentes tipos de movimientos o actividades que una persona está realizando. La aplicación toma dichos datos de sensores de aceleración (por medio de Phyphox), y usa un modelo de Machine Learning para analizar las características de las señales. Los datos obtenidos se procesan para extraer características que permiten clasificar la actividad física, después de realizar la optimización de parámetros o características de forma continua para mejorar el rendimiento y precisión del modelo, la aplicación evoluciona con el fin de detectar automáticamente la actividad que está realizando el usuario, de manera eficiente y precisa.

b. ¿Qué se espera que haga?

Se espera que la aplicación logre varios objetivos clave en tiempo real:

- **Adquisición continua de datos:**

La aplicación debe ser capaz de recibir datos de un dispositivo con acelerómetros, generando datos sobre los movimientos de la persona, que deben ser capturados en el momento.

- **Extracción de características:**

Una vez que los datos son adquiridos, la aplicación debe procesarlos para extraer características relevantes. Esto implica calcular métricas estadísticas y espectrales de las señales de aceleración en los tres ejes (X, Y, Z). Las características que se extraen incluyen la media, desviación estándar, kurtosis (una medida de la "altura" de las distribuciones), skewness (asimetría), y otros parámetros que ayudan a distinguir diferentes tipos de actividades.

- **Clasificación de actividades:**

Usando los datos y las características extraídas, la aplicación debe clasificar la actividad física que se está realizando. Esto se logra utilizando un modelo de Machine Learning entrenado previamente, como Random Forest.

- **Optimización de modelos:**

La aplicación debe poder ajustar los hiperparámetros del modelo utilizando métodos como GridSearchCV

o RandomizedSearchCV. También debe ser capaz de seleccionar las mejores características que realmente aporten información relevante para la clasificación.

- **Mostrar resultados en tiempo real:**

La aplicación debe proporcionar retroalimentación inmediata al usuario, mostrando en pantalla la actividad detectada. Esto permite al usuario conocer en tiempo real qué actividad está realizando, lo que podría ser útil para el seguimiento de su rutina de ejercicios o para monitorear su actividad física a lo largo del día.

En resumen, lo que se espera de esta aplicación es que se convierta en una herramienta precisa y eficiente para clasificar actividades físicas en tiempo real, adaptándose dinámicamente al usuario.

c. ¿Qué ejemplos de aplicaciones hay afuera que hagan algo similar?

- **Aplicaciones industriales:**

Los acelerómetros piezoeléctricos industriales se emplean con frecuencia para supervisar el estado de funcionamiento de maquinaria, pues identificar vibraciones inusuales provocadas por el desgaste de piezas como rodamientos, cojinetes o soportes. Estos dispositivos son esenciales para implementar estrategias de mantenimiento preventivo, ayudando a prolongar la vida útil de las máquinas y evitar averías imprevistas. Se encuentran disponibles en múltiples diseños, niveles de protección y tipos de conexión, lo que permite elegir el modelo más adecuado según las características específicas y las condiciones de trabajo de cada equipo. (Sensing S.L., 2021)

- **Aplicaciones médicas:**

AED (Automated external defibrillator) Plus de Zoll utiliza parches CPR-D⁺ equipados con un acelerómetro para medir en tiempo real la profundidad de las compresiones torácicas durante una reanimación cardiopulmonar.

- **Apple Health:**

Apple Health ofrece una plataforma completa para monitoreo de la salud y el estado físico. Esta app captura información de múltiples fuentes, como el iPhone, Apple Watch y dispositivos de terceros, para rastrear actividades.

- **Garmin connect:**

Garmin usa sus dispositivos de muñeca y otros wearables. Su app también puede ofrecer métricas detalladas sobre el rendimiento físico y proporcionar entrenamientos personalizados.

- **Ámbito deportivo:**

Se ha propuesto incorporar acelerómetros en cascos de fútbol americano para cuantificar la magnitud de los impactos en la cabeza, siendo herramientas valiosas

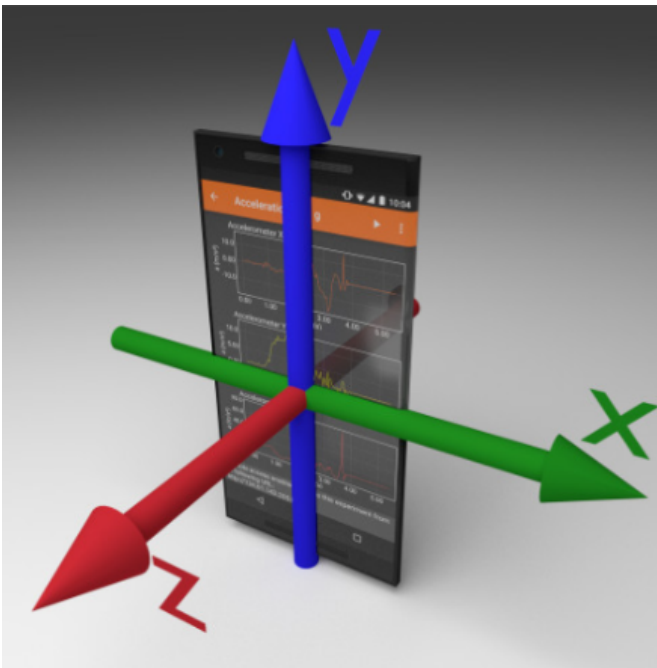
para el monitoreo y evaluación de personas en contextos clínicos y deportivos. (Efe, 2016)

De igual manera, varias empresas han desarrollado relojes deportivos para corredores que incluyen podómetros basados en acelerómetros, permitiendo estimar la velocidad y distancia recorrida. Unos conocidos son: Fitbit, que utiliza acelerómetros en sus dispositivos para reconocer actividad física y etapas de sueño. (Tidd & Tidd, 2023) y Strava, que necesita también datos de un GPS, para detectar automáticamente si el usuario está corriendo o andando en bicicleta. (*Cómo Conectar Tus Actividades a Strava*, n.d.)

d. ¿Qué tipo de información va a monitorear?

En la app phyphox, se manejó un acelerómetro triaxial, donde los ejes están definidos de la siguiente manera:

- El eje z sale perpendicular desde la pantalla, como si "saliera" hacia la cara del usuario.
- El eje x apunta hacia la derecha cuando se sostiene el celular como normalmente se usa (de forma vertical).
- El eje y va hacia arriba, siguiendo el lado largo del teléfono. (Staacks, n.d.)



Para nosotros poder obtener las señales y clasificarlas dependiendo de la actividad que se está realizando, se utilizó el acelerómetro sin g, para obtener la aceleración lineal. La manera en que funciona es, al igual que en otros teléfonos, mediante un sensor MEMS (Sistema Micro Electromecánico) que detecta la aceleración a lo largo de tres ejes: x, y y z. Estos sensores contienen diminutas estructuras móviles que se desplazan cuando el dispositivo experimenta una aceleración. Ese desplazamiento produce una señal eléctrica, que la aplicación Phyphox interpreta para obtener los valores de aceleración en cada dirección. (Dwyer Omega, 2023). Y es esa señal eléctrica la que vamos a monitorear dependiendo la actividad que se realice.

V. DESCRIPCIÓN DE LAS CARACTERÍSTICAS EXTRAÍDAS DE LOS DATOS

Una vez que se obtuvo esta información, se obtuvieron estadísticas temporales, espectrales y la magnitud total del vector de aceleración para que se pudiera decir qué actividad se estaba realizando en tiempo real, al conocer los valores y que estos se alineen a dicha actividad específica. A continuación hay un desglose de las características que calculamos en el archivo de `data_processing`:

Estadísticas temporales:

- `np.mean(sig)`: aceleración promedio
- `np.std(sig)`: variabilidad de la aceleración
- kurtosis: qué tan puntiaguda o plana es la distribución de valores
- skew: qué tan asimétrica está la distribución de valores
- min, max, median, ptp (rango total), percentiles (25 y 75): resumen de la dispersión y tendencia de los datos
- `np.sum(sig**2)`: energía total del movimiento en ese eje

Estadísticas espectrales (frecuencia):

- `fft_vals`: magnitud de la transformada rápida de Fourier
- `np.mean(fft_vals)`, `np.std(fft_vals)`: resumen de la energía en frecuencia
- `fft_freqs[np.argmax(fft_vals)]`: frecuencia dominante, que da una idea de cuántas repeticiones (saltos, sentadillas, etc.) por segundo ocurren

Magnitud total del vector de aceleración:

- `np.linalg.norm(tr[2], axis=1)`: calcula el módulo del vector de aceleración en cada instante

Una vez que se calcularon con ayuda de la feature selection de la librería sklearn (SelectFromModel), éstas fueron las características que nuestro modelo evalúa para determinar cuál actividad se realiza.

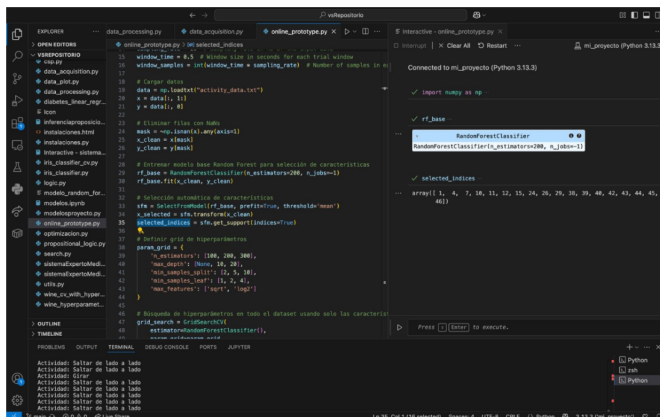
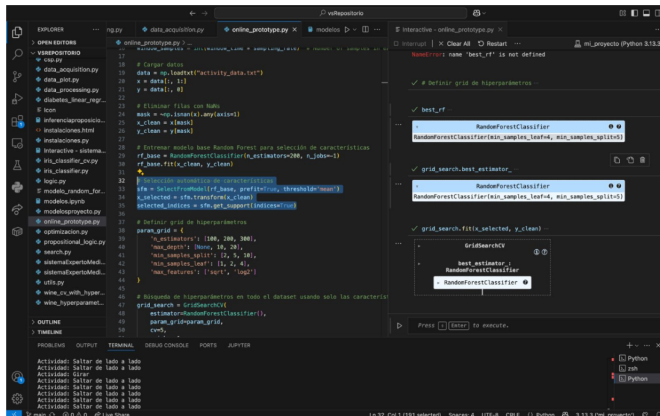
El código nos arroja que las características fueron el siguiente array: `array([1, 4, 7, 8, 10, 11, 12, 15, 21, 24, 26, 29, 38, 39, 40, 42, 43, 44, 45, 46])`. Esas podríamos asumir que se tratan de estas 20 características, en base a cómo las definimos en `data_processing`:

TABLE 1: CARACTERÍSTICAS SELECCIONADAS Y SU EJE/VECTOR CORRESPONDIENTE

Índice	Característica	Eje/Vector
1	Media (mean)	Eje X
4	Mínimo (min)	Eje X
7	Rango (ptp)	Eje X
8	Percentil 25	Eje X
10	Energía (sum(sig ²))	Eje X
11	Media magnitud FFT	Eje X
12	Desviación estándar FFT	Eje X
15	Frecuencia dominante FFT	Eje Y
21	Energía (sum(sig ²))	Eje Y
24	Rango (ptp)	Eje Z
26	Energía (sum(sig ²))	Eje Z
29	Frecuencia dominante FFT	Eje Z
38	Media de magnitud total	Vector total
39	Desviación estándar de magnitud total	Vector total
40	Máximo de magnitud total	Vector total
42	Energía total (sum(norm ²))	Vector total
43	RMS total	Vector total
44	(Posible extra derivada)	Vector total
45	(Posible extra derivada)	Vector total
46	(Posible extra derivada)	Vector total

NOTA: Las posiciones 44, 45 y 46 dependen de cómo concluye la función `extract_features`.

VI. RESULTADOS OBTENIDOS EN LA EVALUACIÓN DE LOS CLASIFICADORES PROBADOS



En un inicio obtuvimos como resultado que en la validación cruzada anidada, Random Forest tuvo un accu-

racy de 0.8561%, mientras que en el MLPClassifier fue de 0.7057%, dándonos cuenta que si es posible reducir el número de características, pues el rendimiento se mantiene o incluso mejora ligeramente si se eligen correctamente, y de la misma manera, la validación cruzada anidada, refuerza esta conclusión. Finalmente fue por ello que Random Forest fue el que se implementó, y una vez entrenado con el GridSearch para los hiperparámetros, su configuración fue la siguiente: RandomForestClassifier(min_samples_leaf=4, min_samples_split=5), donde min_samples_leaf=4 indica que cada hoja del árbol debe tener al menos 4 muestras para poder considerarse válida y min_samples_split=5 significa que para que un nodo se divida en dos ramas, debe haber al menos 5 muestras en ese nodo. El entrenamiento del modelo se llevó de la siguiente manera:

Entrenar modelo base Random Forest para selección de características

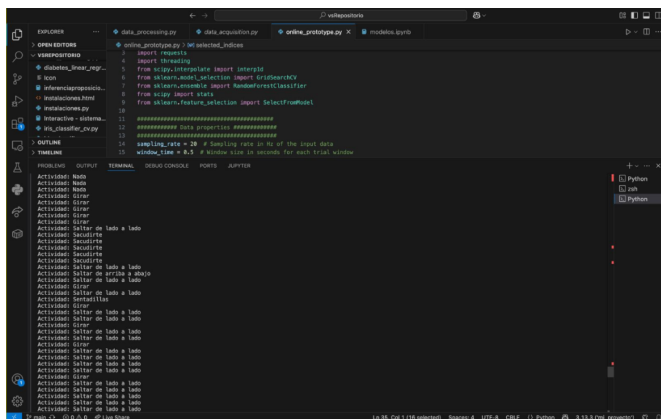
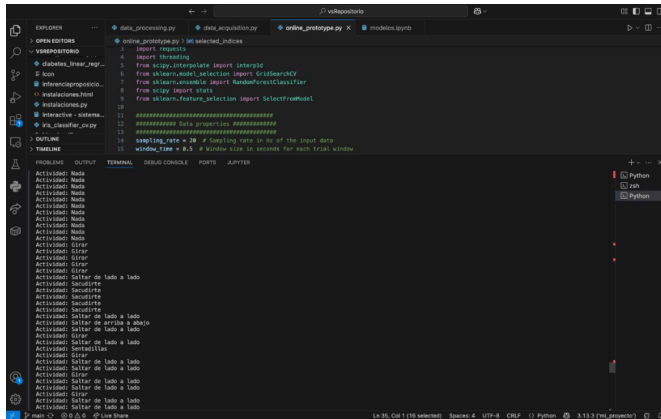
```
rf_base = RandomForestClassifier(n_estimators=200, n_jobs=-1) rf_base.fit(x_clean, y_clean)
```

Se entrenó entonces un modelo Random Forest con 200 árboles, configurado para utilizar todos los núcleos disponibles del procesador y así acelerar su procesamiento. Este modelo se entrena utilizando las variables predictoras `x_clean` y las etiquetas correspondientes `y_clean`. La razón de este entrenamiento inicial es que, durante su proceso, Random Forest calcula de forma automática la importancia de cada característica, utilizando diversos parámetros. Esta información permite identificar cuáles son las variables más relevantes para el modelo, y a partir de ello, seleccionar únicamente las características que realmente aportan valor a la predicción, optimizando así el rendimiento y eficiencia del sistema. Luego se define un diccionario de hiperparámetros (`param_grid`) con diferentes combinaciones de valores para configurar el modelo Random Forest, como el número de árboles (`n_estimators`), la profundidad máxima del árbol (`max_depth`), los criterios mínimos para dividir nodos (`min_samples_split`) y para considerar una hoja como válida (`min_samples_leaf`), y la forma en que se seleccionan las características en cada división (`max_features`). Luego, se usa GridSearchCV, siendo una herramienta que prueba todas las combinaciones posibles de estos hiperparámetros usando validación cruzada de 5 particiones (`cv=5`), con todos los núcleos del procesador (`n_jobs=-1`). Finalmente, se entrena (`fit`) usando sólo las características seleccionadas previamente (`x_selected`) y las etiquetas correspondientes (`y_clean`). En cuanto a las características se utiliza `SelectFromModel` para seleccionar las características más importantes. Esta herramienta toma el modelo Random Forest ya entrenado (`rf_base`) y filtra solo aquellas características cuyo nivel de importancia sea mayor o igual a un umbral. Luego, se transforman los datos con `sfm.transform(x_clean)`, lo que genera un nuevo conjunto de datos `x_selected` compuesto únicamente por las columnas seleccionadas. Posteriormente, con `sfm.get_support(indices=True)` se obtiene un arreglo que indica las posiciones (índices) de las características que fueron seleccionadas. Finalmente, en la feature selection, obtuvimos solo 20 características para considerar. Lo cual puede verse en estas líneas:

```
sfm = SelectFromModel(rf_base, prefit=True,
```

```
threshold='mean')
x_selected = sfm.transform(x_clean)
selected_indices = sfm.get_support(indices=True)
```

VII. APLICACIÓN EN LÍNEA



a. ¿Funciona igual con todos los miembros del equipo?

Después de realizar las pruebas pertinentes, podemos decir que en la mayoría de las pruebas el modelo es generalizable para todos los miembros del equipo, es decir, funciona de manera adecuada para cada uno de los integrantes del equipo, etiquetaba las actividades que efectivamente sí se estaban realizando en el momento de manera sincronizada y eficiente.

b. ¿El rendimiento de la aplicación en línea es el esperado de acuerdo a los resultados de validación cruzada?

Una vez que aplicada la selección automática de atributos, la precisión de las predicciones mejoró notablemente. Esta mejora coincidió con el rendimiento que habíamos anticipado previamente al optimizar el modelo de Random Forest en un Jupyter Notebook, donde habíamos obtenido una precisión del 84.22%. Los resultados se mantienen bastante consistentes siempre y cuando las actividades se realicen de manera similar. Además, una vez que comprendimos cómo el modelo detectaba y clasificaba cada actividad a partir de los patrones de aceleración específicos, nos aseguramos de que se los movimientos de manera uniforme. Esto permitió que los resultados se estabilizaran y fueran coherentes para

todos los participantes, confirmando la efectividad del modelo bajo condiciones controladas.

VIII. CONCLUSIÓN INDIVIDUAL

Nubia Garcidueñas: Este proyecto ha sido una experiencia sumamente enriquecedora tanto a nivel académico como personal. A lo largo de su desarrollo, no solo logré aplicar muchos de los conceptos aprendidos en clase sobre clasificación supervisada, sino que también adquirí nuevos conocimientos técnicos y prácticos que me han dado una visión mucho más clara del potencial que tiene el aprendizaje automático en aplicaciones reales.

Uno de los aspectos que más disfruté fue trabajar con modelos de clasificación. La materia despertó aún más mi interés por este tema, especialmente porque tengo una startup con mis amigos en la que estamos explorando formas de aplicar inteligencia artificial para resolver problemas reales. En ese sentido, este proyecto fue una excelente oportunidad para explorar, experimentar y aprender desde la recolección de datos hasta la clasificación en línea con prototipos funcionales.

Uno de los retos más grandes fue la toma de datos, que resultó mucho más tediosa de lo que inicialmente pensábamos. Tuvimos que repetir varias veces ciertas actividades para garantizar que el dataset fuera consistente, y además enfrentamos errores inesperados como obtener predicciones de “salto lateral” cuando el teléfono estaba completamente estático. Otro reto importante fue investigar cinco nuevos modelos de clasificación que no se habían visto en clase. No solo debíamos comprender su funcionamiento, sino también estudiar sus hiperparámetros, cómo se comportaban con nuestros datos y cómo optimizarlos.

También nos enfrentamos a tiempos de espera prolongados durante la ejecución de algunos scripts, sobre todo al aplicar validación cruzada anidada con búsqueda de hiperparámetros. Esto me obligó a tener más paciencia y a ser estratégica con la selección de parámetros y pruebas. Sin embargo, ver los resultados finales, comprobar que la clasificación en línea funcionaba y entender por qué ciertos modelos eran más efectivos que otros, fue una gran recompensa.

En resumen, este proyecto no solo fortaleció mis habilidades técnicas, sino que también me motivó a seguir explorando el aprendizaje automático, especialmente en el contexto de clasificación de señales en tiempo real.

Mariana León: A lo largo de este proyecto considero que se nos presentaron diversos retos que tuvimos que afrontar como equipo, pero también de manera individual. Uno de ellos fue principalmente en la toma de datos, pues al ser una aplicación completamente desconocida, así como la señal que estábamos utilizando, tuvimos que descubrir cuáles serían las mejores actividades para conseguir resultados óptimos, de qué manera posicionar el teléfono para obtener los mejores datos, entre otros detalles a considerar. Es por ello que mediante cálculo, investigación, prueba y error, pudimos llegar a las condiciones óptimas para un buen resultado. Por otro lado, también fue retador el uso de modelos matemáticos nuevos para evaluar nuestros datos, ya que no solo se trataba de aplicarlos sin más, sino de entender bien cómo funcionaban, qué tipo de datos necesitaban y cómo interpretar los resultados que nos daban. En ese proceso aprendimos a ajustar hiperparámetros, validar nuestros modelos y,

sobre todo, a comparar qué tan bien funcionaban entre sí. Aunque fue algo que al inicio nos costó trabajo, la verdad es que terminó siendo bastante enriquecedor y nos dejó una buena base para proyectos futuros. Además, este trabajo nos permitió entender la importancia de una buena preparación y limpieza de los datos antes de aplicar cualquier modelo, así como la necesidad de hacer pruebas constantemente para asegurarnos de que lo que estamos obteniendo tiene sentido. Finalmente, considero que fue un proyecto bastante interesante, con el que logré aprender mucho acerca de modelos de clasificación, siendo un tema que me interesa bastante y que además me parece fundamental para nuestra carrera. Me gustó poder aplicar todo lo que hemos visto en clase en un caso más práctico, y ver cómo poco a poco se va armando algo real y útil a partir de lo que al inicio parecía solo teoría.

Ana Ana Rello: No voy a negar que, para mí, lo más difícil de este proyecto fue tener todo separado y luego integrarlo en el archivo `online_prototype`. Me preocupaba que, aunque cada parte funcionara bien por sí sola, al juntarlas algo pudiera fallar. Especialmente porque ahora teníamos que recibir datos en tiempo real y probar el modelo que habíamos entrenado previamente, y esa transición me generaba cierta incertidumbre.

Recuerdo que, al inicio, el modelo no lograba distinguir correctamente entre la actividad de no hacer nada y sacudirse. Me puse a revisar las características, a comparar los valores y notar las diferencias que, a simple vista, sí existían. Sin embargo, entendí que una cosa es que yo pudiera ver esas diferencias y otra muy distinta es que el código estuviera correctamente diseñado para que el modelo también pudiera reconocerlas.

Hubo un momento en que decidí rehacer desde cero el `online_prototype`, pero al probarlo, siempre marcaba que se estaba "haciendo nada", aunque el dispositivo estuviera en movimiento. Fue entonces cuando se me ocurrió una idea: tomé una hoja y tracé, desde mi propio entendimiento, el flujo que debía seguir el prototipo en línea. Sin considerar las secciones de código que el profesor había proporcionado (porque en ese momento no dominaba esa parte, solo sabía que debía de tomarse en cuenta `last_data`), identifiqué qué debía leerse, en qué punto analizar los datos y cuándo tomar las mediciones en tiempo real.

Una vez que tuvimos ese pseudocódigo claro, implementar la versión en línea fue mucho más sencillo. Y en ese punto sentí que todo cobró sentido: cada parte del proyecto, que antes parecía aislada, encajó como piezas de un rompecabezas. Fue ahí cuando realmente entendí la importancia de planear, integrar y probar cuidadosamente en proyectos de este tipo. Además de que agradezco eternamente a mi equipo, pues sin ellas, siento que hubiera estado más perdida que nunca, pues me ayudaron muchas veces, implementaron modelos que hasta se me habían olvidado y lograron ayudarme a que la aplicación en línea tuviera como resultado una buena estimación y que cumpliera con todo lo pedido por el profesor.

REFERENCES

- @onlineefe2016, author = Efe, title = Mexicano diseña sistema que registra intensidad de golpes en casco - ESPN, year = 2016, date = 2016-09-25, url = <https://www.espn.com.pa/futbol-americano/nfl/nota/id/2800545/crossDomain,note=ESPN.com.pa>
- @onlinetidd2023, author = Tidd, J., title = How to track your sleep on a fitbit, year = 2023, date = 2023-05-05, url = <https://www.sleepfoundation.org/best-sleep-trackers/how-to-track-your-sleep-on-fitbit>, note = Sleep Foundation
- @onlinestrava, author = Soporte Técnico De Strava, title = Cómo conectar tus actividades a Strava, year = n.d., url = <https://support.strava.com/hc/es-mx/articles/223297187-C>
- @onlinephyphox, author = Staacks, S., title = Supported sensors, year = n.d., url = <https://phyphox.org/sensors/>, note = Phyphox
- @onlinedwyer2023, author = DwyerOmega, title = How to measure acceleration?, year = 2023, date = 2023-07-13, url = <https://www.dwyeromega.com/en-us/resources/accelerometers>, note = DwyerOmega
- @onlinesensing2021, author = Sensing S.L., title = Acelerómetros industriales archivos, year = 2021, date = 2021-06-11, url = <https://sensores-de-medida.es/medicion/sensores-y-transductores/accelerometers-sensores-de-aceleracion/accelerometers-industriales/>
- @onlinedaniel2023, author = Daniel, title = Random Forest: Bosque aleatorio. Definición y funcionamiento, year = 2023, date = 2023-10-30, url = <https://datascientest.com/es/random-forest-bosque-aleatorio-definicion-y-funcionamiento>, note = DataScientest
- @manualmlp2025, title = MLPClassifier, year = 2025, organization = Scikit-learn, url = https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html
- @manualrf2025, title = RandomForestClassifier, year = 2025, organization = Scikit-learn, url = <https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>