1. Introducción

El análisis automático del movimiento humano es una tarea clave en aplicaciones como la rehabilitación física, el monitoreo de personas y la ergonomía. Este proyecto propone el desarrollo de una herramienta basada en inteligencia artificial que permite detectar y clasificar, en tiempo real, actividades corporales específicas como caminar, girar, sentarse y levantarse, a partir de video capturado por cámara. Usando técnicas de estimación de pose y modelos de aprendizaje supervisado, el sistema identifica acciones y analiza patrones posturales mediante el seguimiento de articulaciones clave, proporcionando información relevante sobre ángulos e inclinaciones del cuerpo.

2. Recolección y Análisis de Datos

2.1 Captura de Movimiento

Con el fin de generar un conjunto de datos personalizado para el entrenamiento y evaluación de modelos de clasificación de gestos posturales, se realizaron grabaciones de video en las que distintas personas ejecutaron acciones como caminar, sentarse, girar, inclinarse, acercarse y alejarse. Las grabaciones se realizaron desde una cámara fija para facilitar la extracción coherente de información espacial y temporal. Estos videos constituyen la base para construir un dataset orientado al análisis postural en tiempo real.

2.2 Extracción de Características con MediaPipe

Los videos capturados fueron procesados mediante Media Pipe, un framework de aprendizaje automático (AA) desarrollado por Google para tareas de visión por computadora. En particular, se utilizó el módulo de Pose Landmarker, que permite identificar diversos puntos de referencia anatómicos en cada fotograma, incluyendo cabeza, extremidades y articulaciones principales. El sistema genera tanto coordenadas normalizadas como coordenadas tridimensionales relativas al entorno, permitiendo un análisis detallado del movimiento.

Este proceso convierte cada video en una secuencia estructurada de datos numéricos, que describe con precisión la postura del cuerpo en cada instante, y que posteriormente se utiliza como entrada para los modelos de clasificación.

https://ai.google.dev/edge/mediapipe/solutions/guide?hl=es-419

Figura 1. *Ejemplo de detección de puntos de referencia con MediaPipe.*

La imagen muestra cómo MediaPipe identifica automáticamente los puntos claves del cuerpo humano a partir de un video o imagen. Estos puntos —representados por coordenadas en el plano de la imagen— permiten capturar con precisión la postura y posición de cada articulación, lo que facilita el análisis del movimiento y la clasificación de actividades.

2.3 Organización del Dataset

Una vez obtenidas las coordenadas espaciales de los puntos clave del cuerpo, se procedió a organizar y transformar esta información en una estructura adecuada para el aprendizaje automático. En lugar de considerar cada fotograma como una muestra independiente —lo cual limitaría gravemente la capacidad del modelo para interpretar movimiento— se diseñó una estrategia basada en secuencias temporales.

Cada ejemplo del dataset se construyó a partir de un segmento compuesto por frames consecutivos debidamente etiquetados. Esta ventana temporal fue seleccionada con el objetivo de capturar el desplazamiento de las articulaciones durante cortos intervalos de tiempo, lo que proporciona una mejor descripción del gesto o actividad en curso. A través de este método, se logró representar no solo la postura en un instante, sino también su transición y continuidad, aspectos esenciales para una correcta clasificación de acciones humanas.

Por cada secuencia de fotogramas, se concatenaron las coordenadas tridimensionales de cada punto clave detectado, generando un vector de características de alta dimensionalidad. Esta representación permite que cada muestra contenga una descripción completa del movimiento observado, facilitando el análisis temporal y la detección de patrones dinámicos por parte del modelo.

Desde el punto de vista computacional, este formato vectorial presenta múltiples ventajas. Por un lado, reduce la redundancia en los datos, ya que evita almacenar información estática repetida frame a frame. Por otro lado, ofrece una entrada uniforme y compacta al

modelo, optimizando tanto el procesamiento como la generalización durante el entrenamiento. Además, este enfoque ayuda a mitigar el impacto del ruido generado por pequeñas variaciones o errores de detección en fotogramas individuales. Al considerar secuencias completas, el modelo puede enfocarse en las tendencias generales del movimiento, lo que mejora su robustez frente a fluctuaciones aleatorias y datos atípicos.

4. Ampliación del Dataset

Se incorporaron nuevas sesiones de captura en las que participaron distintas personas, realizando los mismos movimientos clave en condiciones variables. Estas grabaciones incluyeron diferencias en iluminación, tipo de entorno (interior y exterior), ropa utilizada y características individuales de los participantes. Esta variabilidad permite que el modelo aprenda a generalizar mejor y no dependa exclusivamente de condiciones particulares del entorno o de un grupo limitado de sujetos.

El procesamiento de estos nuevos registros se llevó a cabo utilizando el mismo flujo de trabajo aplicado anteriormente, con MediaPipe para la detección de puntos de referencia corporales y OpenCV para la lectura y manejo de los videos. Así, se obtuvieron nuevamente vectores de coordenadas tridimensionales que complementan el dataset original, aportando mayor variedad y robustez al conjunto final utilizado para el entrenamiento y la validación del modelo.

5. Elección de Modelos de Clasificación

Con el objetivo de evaluar distintos enfoques en la clasificación de movimientos humanos, se optó por probar tres algoritmos de aprendizaje supervisado: Support Vector Machine (SVM), Random Forest y XGBoost. Cada uno fue seleccionado por sus ventajas particulares frente al tipo de datos utilizados y las necesidades del proyecto.

Random Forest: fue elegido por su capacidad para adaptarse bien a datos complejos y poco estructurados. En contextos donde las posiciones corporales pueden variar significativamente entre personas o condiciones ambientales, este modelo ofrece una solución estable. Su estructura basada en múltiples árboles independientes permite reducir el impacto del ruido o de valores atípicos que podrían sesgar la predicción si se usara un único árbol. Además, al tratarse de un método de ensamblado, su entrenamiento es relativamente rápido y su rendimiento suele mantenerse alto sin necesidad de ajustes finos.

XGBoost, por otro lado, representa una opción más sofisticada que prioriza la precisión. Su ventaja principal radica en cómo construye los árboles de decisión: cada uno se entrena para corregir los errores del anterior, lo que permite un aprendizaje más refinado. Esto es particularmente útil en tareas donde existen pequeñas diferencias entre clases, como distinguir entre caminar o girar el torso, acciones que pueden compartir coordenadas similares durante ciertos momentos. Su diseño también incluye mecanismos internos de regularización, lo cual ayuda a mantener un buen balance entre precisión y generalización.

SVM fue considerado como un tercer enfoque, especialmente útil cuando se necesita tomar decisiones claras entre clases que pueden estar muy próximas entre sí en el espacio de características. Aunque no es un modelo secuencial como los anteriores, su fortaleza está

en encontrar un límite óptimo de separación entre grupos, incluso en espacios no lineales. Esta propiedad lo convierte en una buena alternativa cuando las actividades se diferencian por pequeños cambios de posición en una o pocas articulaciones, ya que puede capturar esas sutilezas mediante el uso de funciones kernel.

En conjunto, estos tres modelos permitieron comparar distintos enfoques de clasificación: desde la robustez y simplicidad de Random Forest, hasta la precisión progresiva de XGBoost y la separación clara que ofrece SVM. Esta diversidad metodológica fue clave para identificar cuál modelo se adapta mejor a las particularidades del conjunto de datos y al comportamiento dinámico del cuerpo humano en movimiento.

6. Refinamiento del Dataset

Antes de entrenar un modelo de aprendizaje automático, es fundamental asegurarse de que los datos estén en condiciones óptimas. En este proyecto, se trabajó con un conjunto de datos compuesto por coordenadas tridimensionales (x, y, z) de las articulaciones del cuerpo humano, extraídas mediante MediaPipe. Aunque esta información es muy rica, también puede presentar problemas como ruido, duplicados o valores faltantes que deben ser tratados cuidadosamente.

El primer paso consistió en limpiar el conjunto de datos. Se eliminaron registros duplicados y se trataron los casos en los que algunas coordenadas estaban incompletas o presentaban valores atípicos. Esto ayudó a mantener la coherencia de la información y evitó que errores puntuales influyeran en el rendimiento del modelo.

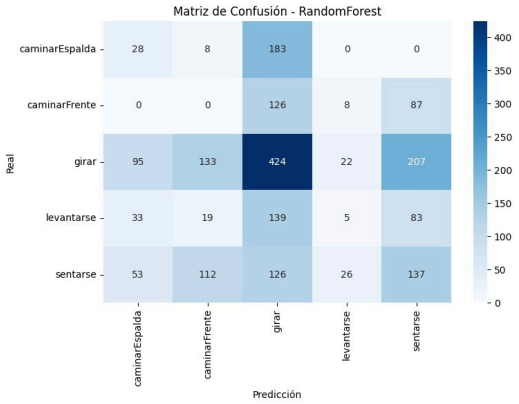
Luego, se realizó un proceso de estandarización utilizando la técnica **StandardScaler**, que permite llevar todas las coordenadas a una misma escala. Dado que los modelos de clasificación pueden verse afectados por la magnitud de los valores, esta transformación evitó que alguna dimensión tuviera un peso mayor solo por tener un rango más amplio.

También se revisó el balance de clases en el dataset. Algunas actividades, como caminar o girar, estaban más representadas que otras como sentarse, lo que podría provocar que el modelo se incline a favor de las clases mayoritarias. Por esta razón, se consideró aplicar técnicas de balanceo para asegurar una representación justa de todas las categorías de movimiento.

Finalmente, se exploró la posibilidad de reducir la dimensionalidad del conjunto de datos. Aunque se disponía de información detallada para cada articulación, no todos los puntos contribuían de forma significativa a la tarea de clasificación. Con el fin de mejorar la eficiencia del entrenamiento y reducir el riesgo de sobreajuste, se consideró aplicar PCA (Análisis de Componentes Principales) y realizar una selección más afinada de las características más relevantes.

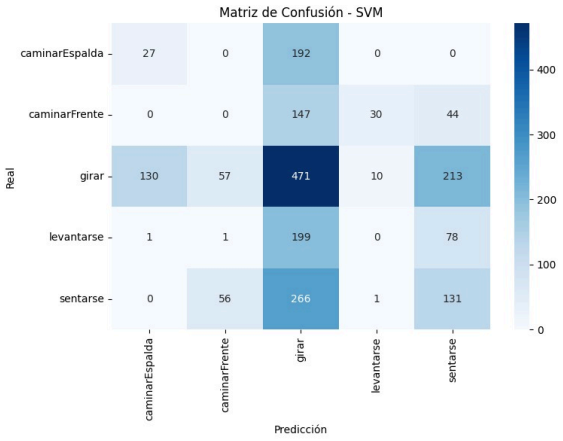
7. Análisis resultados

Random Forest



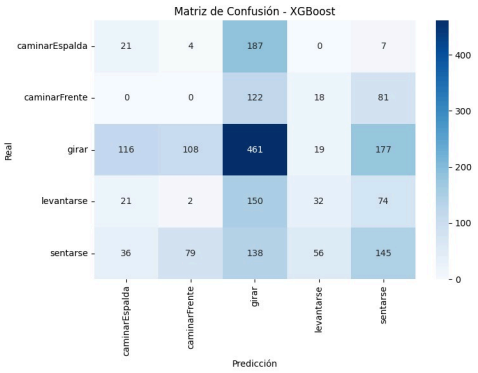
El modelo de Random Forest mostró un accuracy de prueba del 29%, con un desempeño particularmente bajo en las clases minoritarias ("caminarFrente" con F1-score de 0.00 y "levantarse" con 0.03). Su única fortaleza relativa se observó en la clase mayoritaria "girar" (F1-score: 0.45), lo que sugiere que el modelo está sesgado hacia las clases con más muestras. La validación cruzada reveló una alta variabilidad (media 0.317 ± 0.129), indicando inconsistencia en su capacidad de generalización. Estos resultados reflejan las limitaciones del algoritmo para manejar datos desbalanceados sin ajustes específicos.

Support Vector Machine (SVM)



El modelo SVM alcanzó un accuracy ligeramente superior (31%), destacándose en el recall de la clase "girar" (0.53), pero mostrando los mismos problemas graves con "caminarFrente" y "levantarse". Su performance en validación cruzada (media 0.306 ± 0.126) fue similar a Random Forest, confirmando la dificultad intrínseca del problema. El hecho de que mantenga un rendimiento aceptable en "girar" y "sentarse" sugiere que se podría estar captando algunas características discriminativas, pero no suficientes para las clases problemáticas.

XGBoost



XGBoost obtuvo el mejor rendimiento global, mostrando el balance más equilibrado entre métricas. Aunque mantuvo problemas con "caminarFrente", logró mejoras significativas en "levantarse" (F1-score: 0.16) y "sentarse" (0.31). La validación cruzada (media 0.328 ± 0.151) confirmó su ventaja relativa, aunque la alta desviación estándar indica que sigue siendo sensible a particiones específicas del dataset. Su capacidad para manejar mejor el desbalance de clases lo posiciona como el mejor modelo para el proyecto.

8. PLAN DE DESPLIEGUE

Tenemos pensado empaquetar todo el pipeline en un contenedor Docker que incluya las dependencias clave como OpenCV, MediaPipe, scikit-learn, XGBoost e imbalanced-learn. Se definirá un Dockerfile que instale Python 3.10, clona el repositorio del proyecto y prepare un entorno listo para ejecutar. Al igual, tendrá una interfaz web ligera para que el usuario pueda elegir el modelo que desee. Esta imagen contendrá además el modelo entrenado y el escalador, de modo que con un simple docker run se pueda iniciar la captura de video y la inferencia en vivo sin necesidad de configuración adicional.

Aunque para esto también tenemos en cuenta la adopción por parte de usuarios no técnicos, entonces tendremos un manual de usuario que describa paso a paso cómo instalar Docker, descargar la imagen y lanzar el servicio. Así como los parámetros configurables puerto de servicio, resolución de video, rutas de archivos, mejor dicho todo desde 0. Además, se producirá un video demo de menos de cinco minutos que muestre la interfaz en funcionamiento desde la selección de la cámara, el inicio de la detección de pose, hasta la visualización de la actividad clasificada y las métricas en tiempo real. Finalmente, se proporcionará un repositorio público con plantillas de configuración, ejemplos de captura y una sección para resolver diversas preguntas comunes al desplegar el sistema en entornos de laboratorio o de producción.