

Proyecto final - Algoritmos y Programación III

Caso de Estudio: Sistema de Anotación de video

Entrega 2

Análisis de impactos de la solución en el contexto del problema

Nuestro sistema de anotación de video busca ser útil en la rehabilitación física, el monitoreo ergonómico o análisis deportivo. Quiere facilitar el seguimiento de posturas y movimientos sin necesidad de sensores físicos, ayudando y simplificando procesos complicados y lentos para que las personas logren alcanzar sus objetivos y cumplir sus ejercicios con la respuesta inmediata del programa para correcciones y ayuda. Sin embargo, es prioritario mantener la privacidad de los datos obtenidos y las responsabilidades éticas previamente definidas, además de la precisión del modelo ante variaciones de entorno y persona.

Estrategia implementada para la obtención de nuevos datos

Para esta entrega, se consideró que no era necesaria la recolección de nuevos datos y se continuó utilizando los datos obtenidos en la entrega 1, ya que estos fueron grabados con un mismo dispositivo, a una misma velocidad, luz, altura, calidad y con una gran cantidad de personas. Sin embargo, como estrategias a implementar en caso de que se necesiten nuevos y más datos se tendría como prioridad ampliar el tipo de persona documentada, es decir, ampliar la cantidad de voluntarios pero manteniendo una variedad en sexo, altura, peso, color de piel y edad para evitar que los modelos desarrollen ‘bias’ o no detecten a un tipo específico de persona. Al grabar nuevas muestras, es recomendado hacerlo en un lugar donde se pueda controlar la iluminación (salón de clase, estudio de fotografía) para mantener la uniformidad en la luz de los videos, al igual que tener un espacio amplio y adecuado (sin mascotas, objetos ni cuadros alrededor) para evitar el ruido en los videos. También se deben implementar variaciones en los tiempos en los que las personas realizan las actividades a realizar para el entrenamiento.

Preparación de los datos

Para el preprocessing se transformaron los datos crudos extraídos de los videos utilizando ‘MediaPipe Pose’ en un conjunto de características numéricas que los modelos pudieran interpretar. Con los 33 puntos de referencia del cuerpo (landmarks), se normalizaron las coordenadas espaciales centrando la pose en el punto medio de las caderas y escalando respecto a la distancia entre los hombros, para así eliminar la variabilidad causada por el tamaño o la posición del sujeto. Estas características se combinan en un vector plano que representa cada frame del video, obteniendo como resultado un dataset limpio, numérico y adecuado para el entrenamiento de los modelos.

Entrenamiento de los modelos

En el entrenamiento de los modelos, se cargaron las características definidas durante el preprocesamiento y se dividieron en conjuntos de entrenamiento y prueba para asegurar una

evaluación imparcial del desempeño. Se utilizan dos modelos supervisados (*Random Forest* y *XGBoost*) integrados en pipelines que incluyen estandarización ('*StandardScaler*') y reducción de dimensión con *PCA*, para mejorar la estabilidad y eficiencia del aprendizaje. Utilizando '*GridSearchCV*' para encontrar los hiperparametros se optimizaron las configuraciones de cada modelo utilizando validación cruzada.

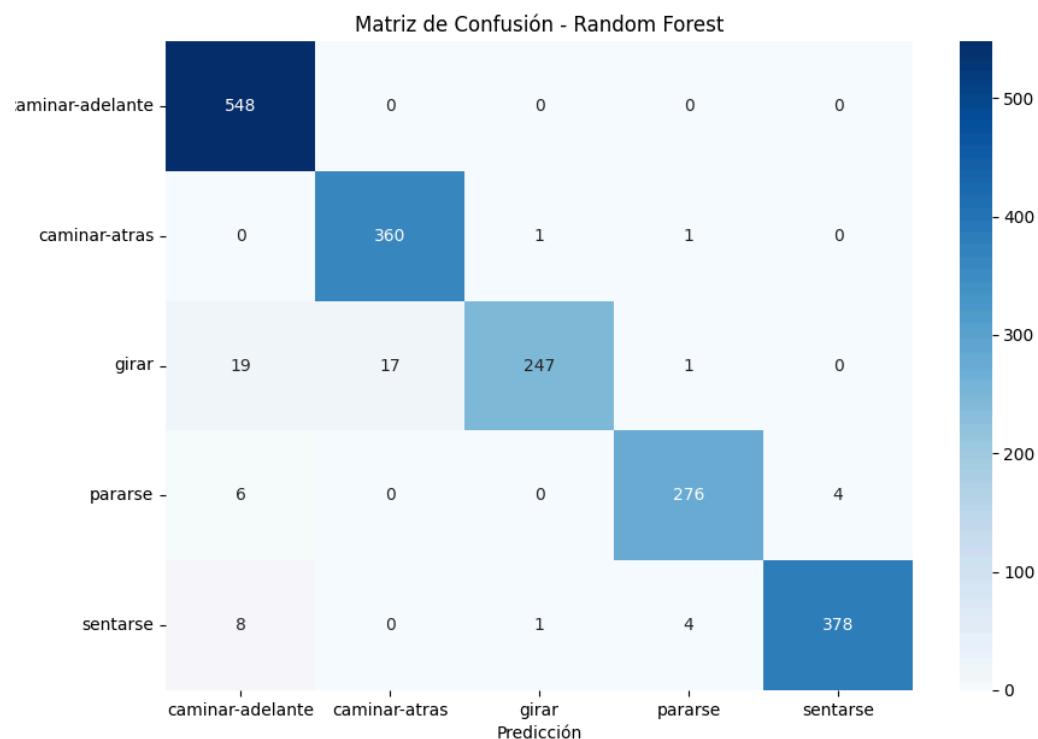
Resultados obtenidos

Después del entrenamiento, se calcularon los resultados obtenidos por cada modelo mediante el uso de las métricas Accuracy, precision y recall f1-score:

Random Forest alcanzó un buen resultado en el conjunto de prueba, con un *accuracy* del 96.69%. Las métricas por clase mostraron un buen equilibrio, con valores de *precision* y *recall* mayores al 94% en todas las categorías. Las tags “caminar-adelante” y “caminar-atras” obtuvieron un *recall* cercano al 100%, lo que muestra que el modelo identifica correctamente casi todos los ejemplos de estas clases. Al contrario, la tag “girar” presentó un *recall* de 0.87, mostrando una mayor dificultad para reconocer algunos casos, lo que puede ser debido a la variabilidad en las trayectorias y el parecido de poses con movimientos de caminar.

Por otro lado, el modelo XGBoost, obtuvo un rendimiento muy similar, con una *accuracy* de 96.15%. Su comportamiento por clase también es equilibrado, con métricas de *precision* y *recall* superiores al 95% en casi todas las categorías. Al igual que en el caso anterior, la tag “girar” fue la más desafiante, con un *recall* de 0.88

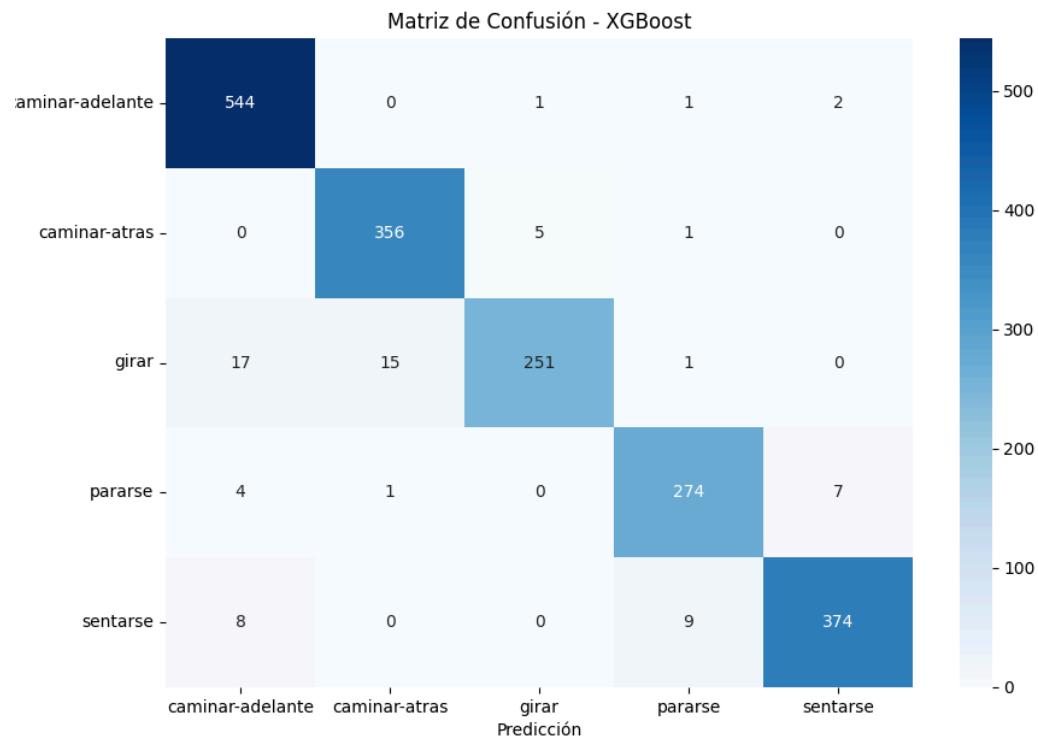
Obtuvimos también matrices de confusión mostrando el desempeño de los 2 modelos



Random Forest muestra alta confusión en las clases de girar, al compartir varios frames que el modelo interpreta como similares con otros movimientos.

Tiene muy buenos resultados en las clases de “pararse” y “sentarse”

Las clases de caminar, presentan leves confusiones entre si



XGBoost muestra un desempeño alto en general, con las confusiones que presenta concentradas entre “girar” y las clases de movimiento (caminar), ya que comparten frames que el modelo interpreta como similares.

Las clases estáticas (“pararse”, “sentarse”) también presentan leve confusión entre sí, al igual que el problema anterior, al ser movimientos que comparten frames similares, el modelo las confunde.

Plan de despliegue

Para el desarrollo del prototipo funcional a desplegar tenemos como objetivo utilizar los modelos realizados en esta etapa para crear la aplicación en tiempo real.

Con un scrip de implementación que:

1. Cargará la cámara web con OpenCV.
2. Cargará el pipeline .joblib (ej. best_random_forest_model.joblib).

3. Capturará frames, aplicará la misma función extract_features y usará el pipeline para predecir la acción.
4. La aplicación mostrará en pantalla mediante una ui fácil de entender la clasificación de los movimientos realizados por la persona, mientras captura los frames.