Reporte de la solución

Fernando Cerriteño Magaña - A01702790 and Leonardo Millán Velázquez A01639823

Instituto de Estudios Superiores de Monterrey, Av. Gral Ramón Corona No 2514, Colonia Nuevo México, 45201 Zapopan, Jal. https://tec.mx/es

Abstract. El propósito de este documento es demostrar la solución a un problema relevante de inteligencia artificial con impacto en la industria, sociedad, economía o salud descrito dentro de una competencia internacional. Más específicamente se planea encontrar un modelo de clasificación para generar una solución, la cual, utilice herramientas computacionales de vanguardia, para poder identificar la etiqueta correspondiente dentro del set de datos.

Keywords: Clasificación · Muestreo · Predicciones · Solución.

1 Introducción

El propósito de este documento es demostrar la solución a un problema relevante de inteligencia artificial con impacto en la industria, sociedad, economía o salud descrito dentro de una competencia internacional. Más específicamente se planea encontrar un modelo de clasificación para generar una solución, la cual, utilice herramientas computacionales de vanguardia, para poder identificar la etiqueta correspondiente dentro del set de datos que se puede encontrar en la página: https://www.kaggle.com/datasets/anshtanwar/adult-subjects-70-95-years-activity-recognition.

Se usará el conjunto de datos The Human Activity Recognition 70+ el cual está profesionalmente anotado y contiene información crucial sobre la actividad humana de 18 adultos mayores, cuyas edades oscilan entre 70 y 95 años. Los participantes usaron dos acelerómetros tridireccionales durante aproximadamente 40 minutos en un entorno de vida libre semiestructurado, con el objetivo de registrar y clasificar sus actividades cotidianas.

Este documento se centra en el desafío de generar un modelo de clasificación robusto utilizando el conjunto de datos HAR70+. Cada sujeto proporciona grabaciones individuales, las cuales están disponibles en archivos .csv separados. Estos archivos contienen información detallada sobre la aceleración en las direcciones x, y, y z de dos sensores: uno ubicado en el muslo derecho y otro en la parte baja de la espalda. Además, se incluye una columna de etiquetas (label) que representa el código de actividad anotada para cada registro.

Las actividades anotadas en el conjunto de datos abarcan una variedad de situaciones cotidianas, desde caminar y estar de pie hasta ascender y descender

escaleras. Cada actividad se identifica mediante un código específico, lo que proporciona un marco de referencia para la clasificación.

El objetivo principal de este documento es desarrollar y evaluar modelos de clasificación de alta precisión que puedan reconocer automáticamente las actividades humanas a partir de los datos de acelerómetros proporcionados. Para abordar este desafío, se utilizarán técnicas avanzadas de aprendizaje automático y se explorarán diferentes enfoques para la preparación y procesamiento de datos.

2 Metodología

En esta sección se platica sobre los diferentes métodos que se utilizaron y una breve descripción del cómo funcionan. La obtención de información se obtuvo de algunas páginas en internet y del libro Hands-On Machine Learning with Scikit-Learn and TensorFlow [1]

2.1 EDA

El Análisis Exploratorio de Datos, o EDA por sus siglas en inglés, es una metodología usada para obtener y entender la información de los conjuntos de datos antes de realizar algún tipo de análisis más avanzado o algún tipo de modelado predictivo.

El EDA consiste en una serie de técnicas y enfoques simples que permiten examinar y comprender las características de los datos. Esto incluye, más no se limita a:

- Resumen estadístico inicial
- Visualización de datos
- Manejo de valores nulos
- Detección de valores atípicos
- Análisis de correlación entre los datos

El EDA es una metodología de vital importancia en la ciencia de datos debido a que proporciona información fundamental para la construcción de modelos de aprendizaje, al mismo tiempo que permite identificar si existen problemas con los datos que se quieren evaluar, ayudando a construir modelos más precisos y significativos.

2.2 Método de balanceo de datos "under-sampling"

El método de undersampling es una técnica de balanceo utilizada para ajustar la distribución de clases en un conjunto de datos.

El undersampling funciona reduciendo aleatoriamente la clase mayoritaria para equilibrar la proporción de cada clase. Esto se hace normalmente durante la fase de entrenamiento del modelo para asegurar que el modelo se entrena en un conjunto de datos equilibrado.

2.3 Método de clasificación SVM

El método de clasificación SVM (Máquina de Vectores de Soporte) funciona correlacionando datos a un espacio de características de grandes dimensiones, de forma que los puntos de datos se puedan categorizar, incluso si los datos no se pueden separar linealmente de otro modo.

La función matemática utilizada para la transformación se conoce como función kernel. SVM admite los siguientes tipos de kernel: Lineal, Polinómico, Función de base radial (RBF), Sigmoide. Deberá experimentar con las diferentes funciones para obtener el mejor modelo en cada caso, ya que utilizan algoritmos y parámetros diferentes.

En este método, se detecta un separador entre las categorías y los datos se transforman de forma que el separador se puede extraer como un hiperplano. Tras ello, las características de los nuevos datos se pueden utilizar para predecir el grupo al que pertenece el nuevo registro.

2.4 Método de clasificación "random forest"

El método de clasificación Random Forest, o Bosque Aleatorio, es un algoritmo de aprendizaje automático que combina la salida de múltiples árboles de decisión para alcanzar un solo resultado.

Su nombre proviene de la aleatoriedad incorporada en la construcción de árboles y la selección de características, lo que lo hace efectivo en una variedad de aplicaciones de aprendizaje automático.

Una de sus ventajas clave es su capacidad para manejar conjuntos de datos grandes y complejos, así como para lidiar con características irrelevantes o ruidosas. Además, proporciona estimaciones de la importancia de las características, lo que puede ayudar en la selección de variables y la comprensión de los factores que influyen en las predicciones.

2.5 Método de clasificación k-vecinos

El método de clasificación k-vecinos, también conocido como KNN o k-NN, es un algoritmo de aprendizaje supervisado no paramétrico. Este algoritmo utiliza la proximidad para hacer clasificaciones o predicciones sobre la agrupación de un punto de datos individual.

El k-NN es un algoritmo de clasificación no paramétrico, lo que significa que no asume ninguna distribución específica de los datos y puede utilizarse en una variedad de situaciones.

El valor de "k" es crucial y debe elegirse con cuidado según el problema. El k-NN es un algoritmo de clasificación que se basa en la similitud de características y es simple pero efectivo en muchos casos.

2.6 Método de obtención de hiperparámetros "filter"

Es una técnica utilizada en el preprocesamiento de datos para la selección de características. Este método se basa en las características numéricas de las variables,

como la correlación con la variable objetiva, en lugar de cualquier resultado del modelo.

Para hacer este método primero se calcula la correlación, luego establece un umbral y por último la selección de características. Este método es simple y rápido, pero no tiene en cuenta las interacciones entre las características, lo que puede ser una limitación en algunos casos.

Adicionalmente, para cada método, se dividió la base de datos en datos de entrenamiento y de testeo, para poder tener una exactitud más precisa con cada modelo.

3 Experimentación

Todos los siguientes puntos fueron elaborados y probados como un archivo .ipynb dentro de la página de desarrollo de Google Colab, esto con el propósito de trabajar en conjunto en un mismo archivo y evitar problemas de compatibilidad. Cabe destacar que en todos los modelos de clasificación se realizó el método de validación cruzada k-folds para mejorar el modelo. Finalmente, todos los conjuntos de datos fueron recuperados de la página de kaggle [2]

3.1 EDA

Al inicio del reto se realizó diferentes métodos para poder identificar diferentes puntos claves de la base de datos, esto incluye; Se checó si existían valores nulos dentro de los datos, Se analizaron diferentes medidas estadísticas en los datos, como el promedio, da desviación estándar, entre otros, Se analizó los tipos de datos que se tenían y se revisó si existía correlación en los datos, finalmente se graficó los diferentes puntos con respecto al tiempo que venía dentro de los datos. Todo esto se realizó con el propósito de poder determinar si los datos eran óptimos para su uso, y en el caso de que no lo fueran, modificarlos, eliminarlos o adaptarlos, para poder usarlos sin ningún inconveniente.

3.2 Balanceo de datos

Al graficar los "labels", identificamos una desigualdad significativa en la distribución de nuestras etiquetas de datos. Observamos que el "label" 1 poseía una cantidad excesiva de datos en comparación con las demás. Esta disparidad en los datos puede llevar a un sesgo en los resultados del modelo, lo que nos llevó a la conclusión de que era necesario implementar un método de balanceo de datos.

Después de considerar varias técnicas, optamos por el método de "undersampling". Esta técnica fue seleccionada principalmente por su eficiencia en el manejo de conjuntos de datos grandes. El método de "undersampling" funciona reduciendo el tamaño del conjunto de datos más grande para que coincida con el tamaño del conjunto de datos más pequeño. En nuestro caso, esto implicó reducir la cantidad de datos correspondientes al "label" 1.

Este enfoque nos permitió equilibrar nuestros datos sin comprometer significativamente la eficiencia del proceso. Sin embargo, es importante destacar que, aunque el método de "undersampling" es eficaz para equilibrar los datos, también puede llevar a la pérdida de información si no se aplica correctamente. Por lo tanto, se debe tener cuidado al implementar esta técnica para asegurar que la información valiosa no se pierda en el proceso.

3.3 Métodos de clasificación

Para los de clasificación se decidió evaluar tanto con datos balanceados como sin balancear. El objetivo de este enfoque dual era evaluar el impacto del balanceo de datos en el rendimiento de cada modelo y, por ende, determinar el modelo más óptimo para nuestro conjunto de datos.

Para facilitar una evaluación exhaustiva y precisa del rendimiento de cada modelo, se generó una matriz de confusión. Esta matriz proporciona una representación visual del rendimiento del modelo, permitiendo una fácil identificación de las verdaderas y falsas predicciones positivas y negativas.

Además, se calculó el recall (sensibilidad), la precisión y el puntaje F1 para cada modelo. El recall es la proporción de verdaderos positivos que se identificaron correctamente. La precisión es la proporción de identificaciones positivas que fueron realmente correctas. El puntaje F1 es una medida de la precisión del modelo que considera tanto la precisión como el recall.

Estas métricas nos permitieron evaluar y comparar el rendimiento de los diferentes modelos con un alto grado de precisión, facilitando así la selección del modelo más adecuado para nuestro conjunto de datos.

3.4 Obtención de hiperparámetros

La optimización de características es un componente crucial en el desarrollo de modelos de clasificación eficaces. Esta técnica nos permite identificar y eliminar las características que no contribuyen significativamente a la capacidad predictiva del modelo, lo que resulta en una mejora de la precisión de las estimaciones del modelo.

Para determinar la relevancia de cada característica, se calcula un promedio basado en el número de características. Este promedio proporciona una medida cuantitativa de la importancia de cada característica, lo que facilita la identificación de las características más y menos significativas.

Además, se generará una lista de las características seleccionadas. Esta lista proporcionará una visión clara de las características que se han identificado como las más relevantes para el modelo.

4 Resultados

Para la obtención de los diferentes resultados se está suponiendo que cada archivo .csv que se tiene, se trata de diferentes personas.

4.1 Balanceo de datos

Como se había relatado previamente en el punto 3.2, la clase 1 de la base de datos tenía una prevalencia de casi el 60%, así mismo se contaban con clases las cuales tenían muy poca representación, como lo muestra la figura 1, en valores numéricos, la distribución se puede representar como se muestra en la tabla 1.

Table 1. Distribución de las clases en porcentaje

Clase	Porcentaje
1	—**60.535336%**—
3	—**01.806278%**—
4	—**00.088581%**—
5	—**00.485269%**—
6	—**13.841710%**—
7	—**14.020797%**—
8	—**09.222030%**—

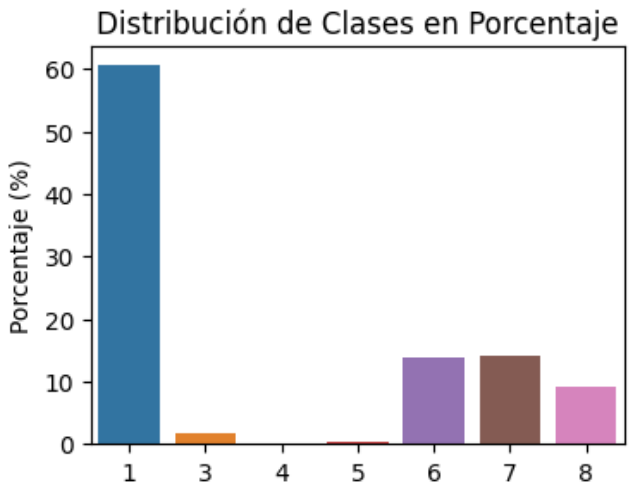

Fig. 1. Gráfico de barras representando la distribución de los datos.

Por lo tanto, se procedió a utilizar el método de balanceo under-sampling como se había mencionado previamente.

4.2 Método de clasificación SVM

Para esta sección se implementó el método de soporte de máquinas vectoriales con la librería de sklearn para analizar sus resultados por clase y de forma

general, se realizaron dos pruebas, una aplicando balanceo de datos, del cual se puede analizar los resultados obtenidos en la figura 2, y otra aplicando el balanceo de datos previamente mencionado, del cual se pueden analizar los resultados en la figura 4. Así mismo, se realizó la matriz de confusión para ambos casos, las cuales se pueden visualizar en las figuras 3 y 5 respectivamente.

	precision	recall	f1-score	support
1	0.87	0.98	0.92	62872
3	0.00	0.00	0.00	1876
4	0.00	0.00	0.00	92
5	0.00	0.00	0.00	504
6	0.85	0.52	0.64	14376
7	1.00	1.00	1.00	14562
8	1.00	1.00	1.00	9578
accuracy			0.90	103860
macro avg	0.53	0.50	0.51	103860
weighted avg	0.88	0.90	0.88	103860

Fig. 2. Resultados del modelo SVM sin balanceo de datos.

Analizando los resultados obtenidos, se puede interpretar a simple vista que el método cuando no se aplicaba un balanceo era mejor, no obstante, esto puede ser refutado cuando se evalúan los resultados de cada una de las clases, ya que se puede observar que las clases 3, 4 y 5 no se están contando, teniendo una calificación de 0 en cada prueba, esto se debe al desbalance mencionado previamente, esto se puede verificar mirando a la matriz de confusión, ya que analizando los valores en las posiciones 3-3, 4-4 y 5-5, podemos observar que el modelo nunca pudo predecir el valor de forma correcta, así mismo se puede observar como predice que dichos valores eran 1, mostrando de forma indirecta que esta clase cuenta con mayor relevancia en el set de datos.

4.3 Método de clasificación Random Forest

El segundo modelo que se evaluó fue Random Forest, igual que con SVM, se uso la libreria sklearn para generar el modelo y posteriormente evaluar los resultados del mismo, de igual manera se realizaron dos pruebas, una sin balanceo de datos figura 6 y otra con under-sampling figura 8, así como las matrices de confusión de cada uno, figura 7 y 9.

Analizando los resultados se puede notar que se tiene el mismo caso que se presentó en el modelo SVM con datos desbalanceados pero a una menor escala, ya que ahora la principal clase afectada fue la 4, analizando la matriz de confusión podemos ver el problema mencionado, ya que en la casilla 4-4 no cuenta con ningún dato acertado, dando entender que al modelo aún le cuesta clasificar esta clase, si se hace una comparación con la matriz de confusión con

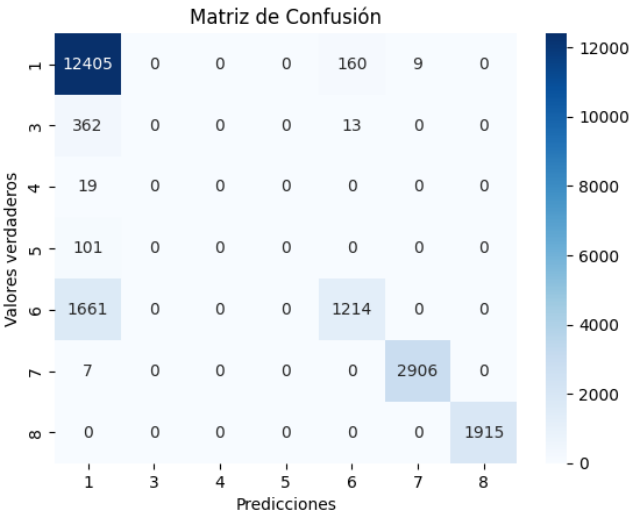

Fig. 3. Matriz de confusión del modelo SVM sin balanceo de datos.

	precision	recall	f1-score	support
1	0.96	0.32	0.48	62872
3	0.04	0.25	0.06	1876
4	0.01	0.57	0.01	92
5	0.02	0.46	0.03	504
6	0.56	0.89	0.69	14376
7	0.95	1.00	0.97	14562
8	1.00	1.00	1.00	9578
accuracy			0.56	103860
macro avg	0.50	0.64	0.46	103860
weighted avg	0.89	0.56	0.62	103860

Fig. 4. Resultados del modelo SVM con under-sampling.

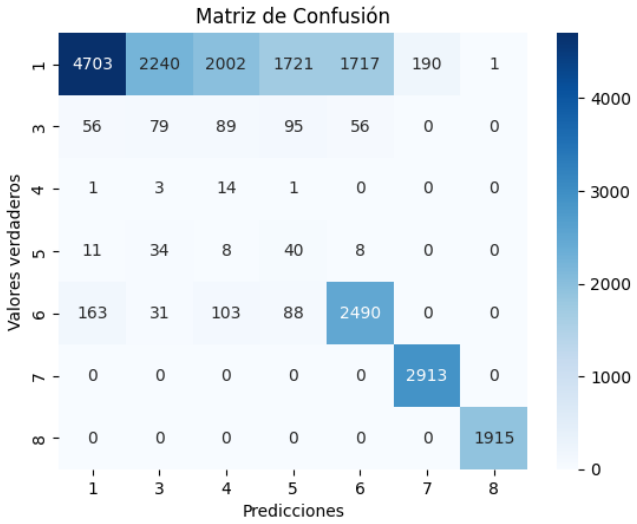

Fig. 5. Matriz de confusión del modelo SVM con under-sampling.

	precision	recall	f1-score	support
1	0.97	1.00	0.98	62872
3	0.84	0.25	0.38	1876
4	0.00	0.00	0.00	92
5	0.87	0.08	0.15	504
6	0.97	0.97	0.97	14376
7	1.00	1.00	1.00	14562
8	1.00	1.00	1.00	9578
accuracy			0.97	103860
macro avg	0.81	0.61	0.64	103860
weighted avg	0.97	0.97	0.97	103860

Fig. 6. Resultados del modelo Random Forest sin balanceo de datos.

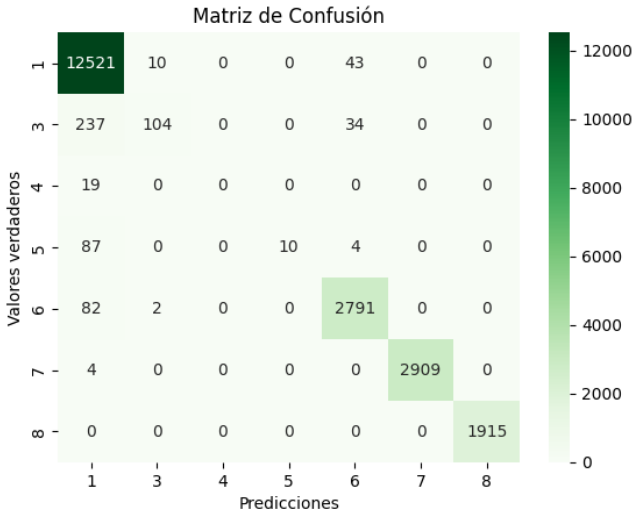

Fig. 7. Matriz de confusión del modelo Random Forest sin balanceo de datos.

	precision	recall	f1-score	support
1	0.99	0.67	0.80	62872
3	0.14	0.54	0.22	1876
4	0.01	0.74	0.02	92
5	0.04	0.68	0.07	504
6	0.90	0.87	0.88	14376
7	0.98	1.00	0.99	14562
8	1.00	1.00	1.00	9578
accuracy			0.77	103860
macro avg	0.58	0.79	0.57	103860
weighted avg	0.96	0.77	0.84	103860

Fig. 8. Resultados del modelo Random Forest con under-sampling.

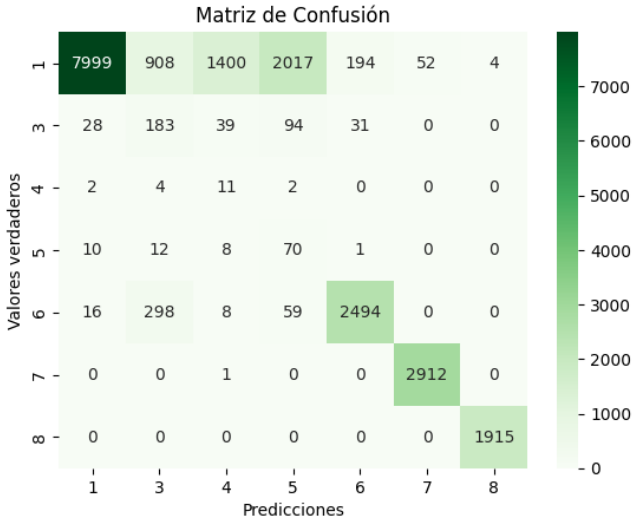

Fig. 9. Matriz de confusión del modelo Random Forest con under-sampling.

datos balanceados, podemos ver como cuenta con más aciertos en las clases 3, 4, 5 y 7, y mientras que disminuye en las demás clases, es preferente tener este modelo a uno en donde solo sirve con la clase 1.

4.4 Método de clasificación KNN

El último modelo a evaluar fue el KNN, igual que con los otros dos modelos se generaron dos casos, uno con los datos desbalanceados y otro con under-sampling, y con ayuda de la librería sklearn, se generaron los modelos con sus respectivos datos y se analizaron los resultados, figuras 10 y 12, al mismo tiempo que se generaron sus matrices de confusión, figuras 11 y 13.

Analizando las matrices de confusión se puede notar que el modelo con los datos balanceados es mejor, al tener un número mayor de aciertos en las clases con menos representación que en la matriz de confusión con los datos desbalanceados, no obstante analizando los resultados en las figuras 10 y 12 se puede ver que se tiene un nivel mayor de precisión en las clases con menor representación en el conjunto de datos desbalanceados que en el conjunto de datos balanceados, esto puede ser explicado con el parámetro de recall, la cual determina él cuan bien puede el modelo detectar a la clase correspondiente, y si se analiza el recall entre las figuras 10 y 12, se puede determinar que la figura 12 cuenta con un mejor recall en dichas clases.

4.5 Selección del modelo

Al analizar los resultados entre los modelos realizados se determinó que el mejor modelo para este set de datos es Random Forest con under samplimg, debido

	precision	recall	f1-score	support
1	0.98	0.98	0.98	62872
3	0.63	0.47	0.54	1876
4	0.58	0.21	0.30	92
5	0.65	0.36	0.46	504
6	0.96	0.98	0.97	14376
7	1.00	1.00	1.00	14562
8	1.00	1.00	1.00	9578
accuracy			0.97	103860
macro avg	0.83	0.71	0.75	103860
weighted avg	0.97	0.97	0.97	103860

Fig. 10. Resultados del modelo KNN sin balanceo de datos.

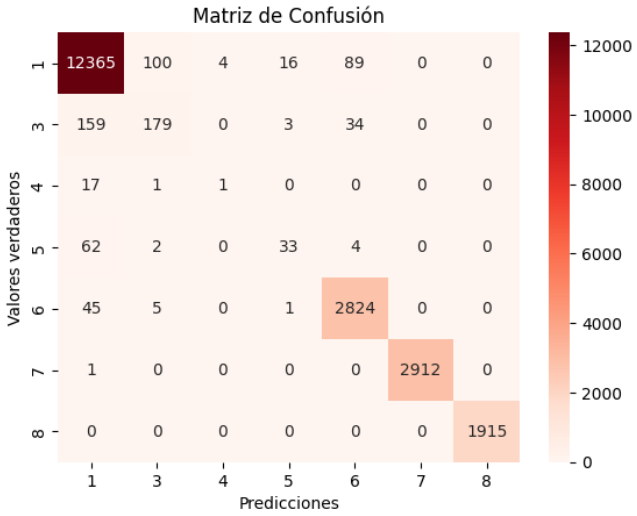

Fig. 11. Matriz de confusión del modelo KNN sin balanceo de datos.

	precision	recall	f1-score	support
1	0.99	0.64	0.78	62872
3	0.12	0.53	0.20	1876
4	0.01	0.73	0.02	92
5	0.04	0.64	0.08	504
6	0.88	0.89	0.89	14376
7	0.96	1.00	0.98	14562
8	1.00	1.00	1.00	9578
accuracy			0.76	103860
macro avg	0.57	0.78	0.56	103860
weighted avg	0.95	0.76	0.83	103860

Fig. 12. Resultados del modelo KNN con under-sampling.

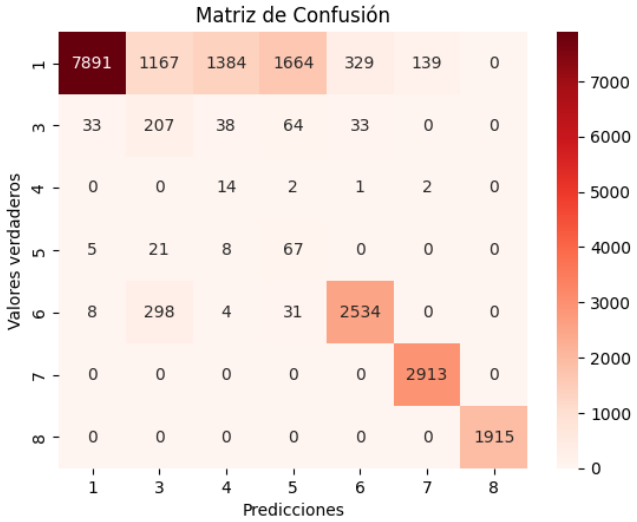

Fig. 13. Matriz de confusión del modelo KNN con under-sampling.

a que cuenta con los mejores valores de recall y obtuvo una exactitud de 78% siendo la más alta entre los diferentes métodos evaluados, aparte de que cuenta con las precisiones individuales más altas entre los otros dos métodos, el único inconveniente que se tiene es que cuenta con precisiones muy bajas en algunos datos, como 0.01 cuando la clase es igual a 4 o 0.04 cuando es igual a 5, mientras que se puede justificar con los valores mencionados de recall, esto aún puede generar inconvenientes al momento de clasificar.

4.6 Optimización de características

Al analizar los el resultado de la obtención de características e hiperparámetros se puede llegar a entender que las características a utilizar serían 4, siendo back x, back z, thigh x y thigh z por lo que podemos llegar al entendimiento que el eje y tanto para el sensor de la espalda como el de la pierna no termina siendo un valor tan determinante, como lo podría ser el eje x o z, a la hora de estimar un valor y así saber en qué posición se encuentra la persona.

4.7 Selección de hiperparámetros

Para haber llegado al resultado anterior se obtuvieron, primero se tuvieron que determinar los parametros para el metodo de arboles de busqueda que fueron ”n-estimators”, que son el numero de arboles de busqueda, ”max-depth”, que determina la profundidad de los arboles, ”min-samples-split”, que es el número mínimo de muestras requeridas para dividir un nodo, ”min-samples-leaf”, que es número mínimo de muestras requeridas en cada hoja. Ya haciendo esto los

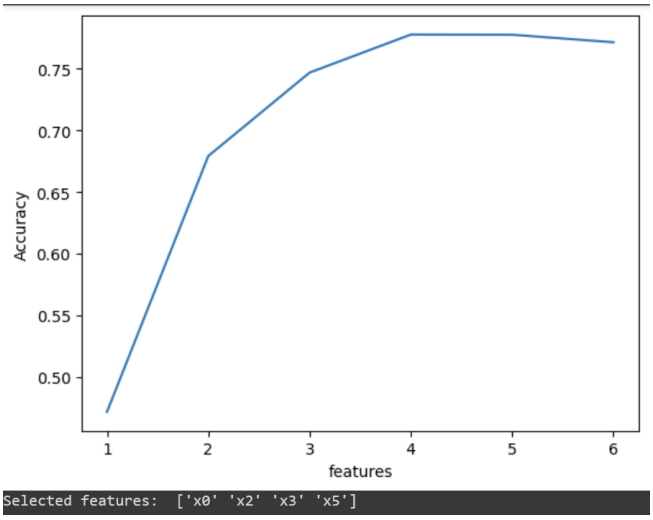

Fig. 14. Características seleccionadas.

mejores 5 resultados se obtuvo, fueron que para el "max-depth" fueron 20,30 y none respectivamente, para "min-samples-leaf" fueron 4, 2 y 1 respectivamente, "min-samples-split" fueron 5, 2 y 10 respectivamente y por ultimo "n-estimators" fueron 300, 200 y 100 respectivamente. (Fig.15)

```
Hiperparámetros Óptimos Seleccionados para Cada Partición:
Partición 1: {'max_depth': 30, 'min_samples_leaf': 2, 'min_samples_split': 5, 'n_estimators': 100}
Partición 2: {'max_depth': 20, 'min_samples_leaf': 4, 'min_samples_split': 10, 'n_estimators': 300}
Partición 3: {'max_depth': 30, 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 200}
Partición 4: {'max_depth': 20, 'min_samples_leaf': 4, 'min_samples_split': 5, 'n_estimators': 200}
Partición 5: {'max_depth': None, 'min_samples_leaf': 2, 'min_samples_split': 5, 'n_estimators': 300}

              precision    recall  f1-score   support

           1       0.99      0.69      0.81     62872
           3       0.15      0.45      0.22      1876
           4       0.01      0.62      0.02        92
           5       0.03      0.61      0.06       504
           6       0.89      0.86      0.87     14376
           7       0.95      1.00      0.98     14562
           8       1.00      1.00      1.00      9578

    accuracy                           0.78    103860
   macro avg       0.57      0.75      0.57    103860
weighted avg       0.95      0.78      0.85    103860
```

Fig. 15. Hiperparámetros seleccionados.

4.8 Interfaz web para utilizar el modelo

Una vez que se obtuvo el modelo con los mejores hiperparámetros, se utilizó la librería pickle en Python para poder exportar ese modelo como un archivo .sav, una vez se tiene este modelo se creó un código compatible con flask, el cual carga

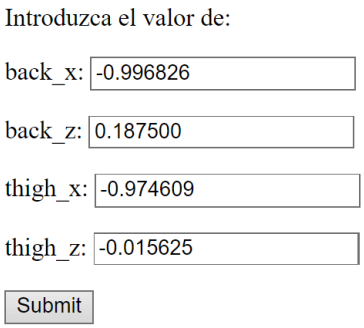

Fig. 16. Solicitud de los datos

Fig. 17. Resultado con la clase

el modelo, y con base en los parámetros que el usuario seleccione, imprime un resultado

4.9 Evaluación del modelo con otros conjuntos de datos

Una vez que se obtuvo el mejor modelo posible, se decidió probarlo con una base de datos perteneciente a otra persona (persona 14), esto con el fin de evaluar la efectividad del modelo con diferentes bases de datos. No obstante, los resultados obtenidos, figura 18, demuestran que el modelo no sirve para este otro tipo de datos, analizando la diferencia de los datos predichos correctamente con el número de predicciones en total, podemos obtener una exactitud del 26%.

Con estos resultados podemos llegar a la conclusión que el modelo necesita ser adaptado o entrenado con el conjunto de datos de la persona 14, esto es debido a que los datos son diferentes, para justificar este punto, se generó una gráfica comparativa entre las personas 1, 2 y 14, esto para comparar la actividad que se estaba elaborando, para una mejor visualización, se tomaron 10 puntos aleatorios para comparar que actividad se estaba realizando, esto con el propósito de revisar los comportamientos similares de las personas, figura 19, así mismo, se puede atribuir el problema a la forma en la que diferentes personas realizan diferentes actividades. La leyenda de los colores se puede encontrar en la figura 20.

5 Conclusiones

5.1 General

Como aprendizaje general podemos destacar la importancia del balanceo de datos, la selección adecuada de características y la adaptabilidad del modelo para lograr una clasificación precisa en un contexto de reconocimiento de actividades humanas. Sin embargo, se debe tener precaución al aplicar el modelo a

	Predicción	Dato Real	Resultado
66412	3	6	False
77853	3	1	False
87289	4	1	False
22621	3	6	False
20994	7	7	True
91334	3	6	False
13420	7	7	True
57993	3	1	False
81525	3	1	False
14102	7	7	True
14582	7	7	True
48325	8	8	True
4852	1	1	True
93054	3	1	False
58906	3	6	False
1244	3	6	False
13678	7	7	True
23140	1	1	True
47595	8	8	True
18307	7	7	True

Fig. 18. Tabla con las predicciones y valores reales.

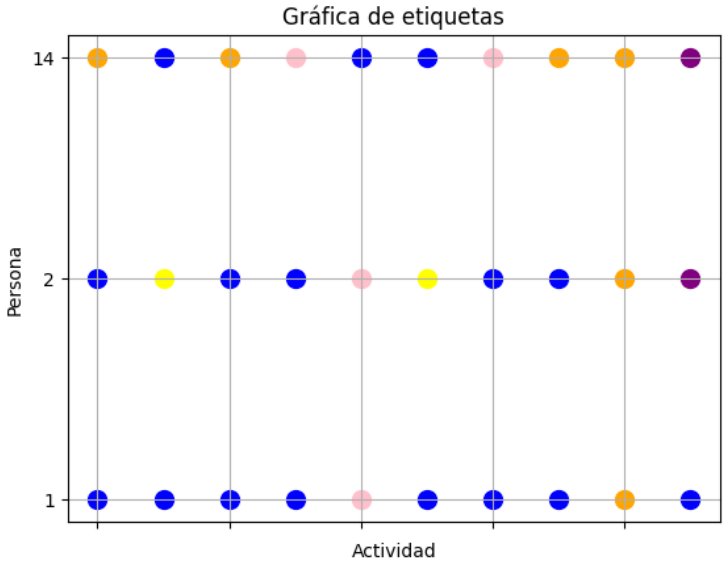

Fig. 19. Gráfica con los labels de diferentes personas en el mismo tiempo.

Fig. 20. Leyenda de los colores para la figura 16

Label	Color
Caminando (1)	**Azul**
Gatear (3)	**Amarillo**
Subiendo (4)	**Verde**
Bajando (5)	**Rojo**
Parado (6)	**Morado**
Sentado (7)	**Naranja**
Acostado (8)	**Rosa**

conjuntos de datos diferentes, ya que puede requerir ajustes y reentrenamiento para garantizar su eficacia.

En cuanto a los resultados, podemos concluir que el modelo de Random Forest con las características e hiperparámetros óptimos, nos puede servir para predecir las clases de una persona, no obstante, si se trata de predecir con los datos de otra, este puede fallar.

Una forma para incrementar podría ser probando con un set de datos conjunto, es decir, combinar todos los datos de las diferentes personas en uno solo para poder contar con más datos para analizar, otra técnica que se podría haber utilizado podría ser la de sobre muestreo para balancear los datos, ya que solo se utilizó sub muestreo, ya que era más rápido y no compromete tanto los datos.

También se pudo haber probado diferentes métodos de validación cruzada para analizar las muestras de entrenamiento y testeo, ya que solo se implementó K-folds.

Así mismo, cabe destacar que el modelo seleccionado puede no ser el más óptimo, ya que solo se analizaron los tres modelos presentados, por lo que un modelo como un perceptron multicapa o Regresión lineal pudieron haber otorgado mejores resultados.

5.2 Fernando

Esta actividad me ayudo mucho con modelos de clasificación, ya que pude reforzar la implementación de algunas de las diferentes técnicas vistas en clase, así mismo logre tener una mejor idea sobre los diferentes puntos tocados a lo largo del documento, por ejemplo, aprendí la importancia de los EDA, ya que previamente realizaba algunas de las técnicas que se implementan en este, sin embargo, no les prestaba tanta atención o no las consideraba tan útiles, no obstante, una vez que las realicé de forma más completa y las analicé de forma más detallada, me di cuenta de lo útiles que son al momento de visualizar o entender la información de los datos. Otro tema en el cual mejore fue al momento de comprender los resultados de los modelos obtenidos con la función .score(), ya que solo me fijaba en los valores de precisión y exactitud, pero gracias a que se tuvo casos como los vistos en el KNN ahora puedo comprender mejor datos como el recall y el f1-score, así como su importancia al momento de evaluar un modelo, finalmente uno de los puntos con los cuales no contaba con ningún tipo de experiencia fue con el desarrollo del entorno web, ya que mientras que fue algo muy simple, creo que resulta de gran valor y me pareció bastante entretenido.

5.3 Leonardo

En el transcurso de este desafío, la implementación de modelos de clasificación ha demostrado ser de gran importancia. La necesidad de construir múltiples modelos para determinar el más adecuado ha reforzado mi comprensión de estos algoritmos. Además, he adquirido un conocimiento más profundo sobre las mejores estimaciones y los valores clave que deben considerarse para identificar el modelo óptimo. El Análisis Exploratorio de Datos (EDA) ha demostrado ser

una herramienta esencial en los métodos de clasificación. Este proceso permite una comprensión más profunda de los datos y facilita la identificación de patrones, tendencias y relaciones que pueden no ser evidentes a primera vista. Además, he adquirido una valiosa experiencia en la comprensión y aplicación de hiperparámetros y características. La elección correcta del número de estos elementos puede mejorar significativamente la precisión de los procesos. Este conocimiento es crucial para optimizar los resultados y garantizar la eficacia del modelo. En resumen, este desafío ha proporcionado una oportunidad invaluable para profundizar en el uso de modelos de clasificación, el análisis exploratorio de datos y la optimización de hiperparámetros y características. Estas habilidades son fundamentales para el desarrollo y la implementación exitosa de soluciones basadas en datos.

References

1. Aurélien, G: Hands-on machine learning with Scikit-learn and TensorFlow. 2nd edn. O'Reilly, Location (2019)
2. Kaggle Human Activity Recognition with sensors (HAR70+) https://www.kaggle.com/datasets/anshtanwar/adult-subjects-70-95-years-activity-recognition. Last accessed 14/09/2023