El proceso de desarrollo de un modelo de Machine Learning, enmarcado en el contexto del **Proceso de Descubrimiento de Conocimiento en Bases de Datos (KDD - Knowledge Discovery in Databases)**, comienza con una rigurosa preparación de los datos. En este proyecto, se utilizó el conjunto de datos **Human Activity Recognition Using Smartphones (UCI-HAR)**. Este dataset, fundamental para nuestro análisis, contiene series temporales multivariadas de sensores de aceleración y giroscopio de dispositivos móviles, registradas mientras un grupo de 30 voluntarios realizaba 6 actividades distintas.

**Fases del Proceso KDD aplicadas:**

**Selección de Datos:** Se seleccionó el dataset UCI-HAR, apropiado para el problema de reconocimiento de actividad humana, que incluye datos de acelerómetros y giroscopios de smartphones.

Para este proyecto, el dataset UCI-HAR se obtuvo como un archivo .zip de un repositorio de GitHub. Se implementó un proceso ETL básico para extraer el archivo comprimido del repositorio, Transformar los datos descomprimiendo el archivo, y cargar su contenido en un repositorio local para su posterior análisis y uso como insumo en el modelo.

Gráfico, Gráfico de cajas y bigotes

El contenido generado por IA puede ser incorrecto.

Para la construcción de este proceso se utilizó como lenguaje de programación Python y las siguientes librerías implementadas en el siguiente bloque de codigo

|  |  |
| --- | --- |
| zipfile | Se utiliza para leer, comprimir y descomprimir archivos zip |
| io | Contiene métodos que nos permiten crear objetos tipo archivo en memoria |

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

**Preprocesamiento y Transformación de Datos**

Esta fase es crucial para asegurar la calidad y el formato adecuado de los datos para su posterior modelado.

Características del dataset UCI-HAR:

* Sensores: Acelerómetro y giroscopio triaxiales, capturando señales de aceleración lineal, aceleración angular y gravedad.
* Frecuencia de muestreo: 50 Hz.
* Segmentación: Los datos originales fueron segmentados en ventanas de 2.56 segundos (128 muestras) con una superposición del 50%. Estas ventanas representan la entrada principal para ambos modelos.
* Actividades clasificadas:
  1. Caminando (WALKING)
  2. Subiendo escaleras (WALKING\_UPSTAIRS)
  3. Bajando escaleras (WALKING\_DOWNSTAIRS)
  4. Sentado (SITTING)
  5. De pie (STANDING)
  6. Acostado (LAYING)

Luego de la carga del dataset procedemos a la correspondiente validación de valores nulos, blancos dentro del dataset uci-har

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

Tabla

El contenido generado por IA puede ser incorrecto.

Una captura de pantalla de una computadora

El contenido generado por IA puede ser incorrecto.

Al identificar que el dataset no contiene información nula o vacía procedemos a realizar un análisis exploratorio de datos mostrando las diferentes acciones mediante el uso de una nube de palabras

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

Texto

El contenido generado por IA puede ser incorrecto.

Como se observa, el dataset UCI-HAR contiene seis actividades distintas: LAYING, STANDING, WALKING, WALKING\_UPSTAIRS, y WALKING\_DOWNSTAIRS. Esto delimita el alcance del proyecto a la predicción de estas seis acciones. Una vez identificadas las categorías, se procede a evaluar el balanceo de clases en el dataset de entrenamiento, obteniendo el siguiente resultado

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

Gráfico, Gráfico de barras

El contenido generado por IA puede ser incorrecto.

El gráfico anterior permite una inferencia preliminar sobre el balanceo del dataset de entrenamiento. Aunque a primera vista no parece desbalanceado, para respaldar esta observación con datos precisos, se procederá a calcular el ratio entre las clases.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Como parte del análisis exploratorio, se utilizó la Reducción de Dimensionalidad (PCA) para visualizar la distribución de los puntos en cada clase. PCA permitió reducir las 561 características del dataset a 2 o 3 componentes principales, capturando la variabilidad de los datos. Esto facilitó la identificación de agrupaciones o mezclas entre las clases (visualizadas por colores) y la detección de patrones inusuales o outliers.

Gráfico, Gráfico de dispersión

El contenido generado por IA puede ser incorrecto.

El gráfico muestra un solapamiento entre las clases SITTING y STANDING, indicando una posible dificultad en su clasificación, mientras que LAYING, WALKING\_UPSTAIRS y WALKING\_DOWNSTAIRS están bien diferenciadas. Para mejorar la visualización de la separación entre clases en el dataset, se utilizará una técnica de reducción de dimensionalidad no lineal TSNE.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

Gráfico, Gráfico de dispersión

El contenido generado por IA puede ser incorrecto.

El gráfico muestra una mejor separación de clases en comparación con PCA, gracias a la reducción de dimensionalidad no lineal que preserva las relaciones locales. SITTING y STANDING, antes solapadas, ahora están más separadas, mientras que LAYING, WALKING, WALKING\_UPSTAIRS y WALKING\_DOWNSTAIRS aparecen como clústeres bien diferenciados. Finalmente, se muestra información estadística del dataset y se valida la inexistencia de registros duplicados que pudieran afectar el entrenamiento.

**Transformación para RandomForestClassifier (características extraídas):** Para el modelo RandomForestClassifier, se utilizaron características pre-extraídas proporcionadas con el dataset UCI-HAR. Estas características incluyen estadísticas de dominio de tiempo y frecuencia (por ejemplo media, desviación estándar, energía, coeficientes de transformada de Fourier) calculadas sobre las ventanas de las señales de los sensores. Las etiquetas de las clases fueron codificadas a un formato numérico adecuado para los algoritmos utilizando StandarScaler , presente en la librería de python sklearn

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

**Transformación para CNN1D (datos crudos):** Para el modelo CNN1D, se procesaron los datos crudos directamente. A diferencia de enfoques que utilizan características extraídas, esta aproximación permite a la red aprender y extraer las características relevantes directamente de las series temporales. Para estandarizar las señales y asegurar un entrenamiento óptimo, se aplicó una normalización Z-score a todas las series temporales de cada ventana

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

Este bloque de código organiza el dataset a la forma (n\_samples, 128, 9) para la entrada de la CNN1D, donde 128 son los pasos de tiempo y 9 los canales/sensores. Normaliza cada señal individualmente (media 0, desviación estándar 1) para optimizar el entrenamiento, y hereda de tf.keras.utils.Sequence para un uso eficiente de memoria y paralelización con model.fit() y model.evaluate(). Además, simplifica la gestión de datos al ofrecer una opción integrada para la separación en conjuntos de entrenamiento y validación. Tras esta fase, se procede a construir el modelo CNN1D.

**Transformación para SVM y SVM con GRIDSEARCH**

En SVM para este modelo se esta utilizando escalado de datos mediante el uso de la librería **sklearn.preprocessing.StandardScaler** , el objetivo de la misma es lograr que los datos del dataset uci-har tengan una media igual a cero (0) y una desviación estándar de uno (1) , requisito crucial para muchos algoritmos de aprendizaje automático, incluyendo las máquinas de vectores de soporte (SVM) que son utilizadas en este proyecto.

**Minería de Datos**

En esta fase se aplican los algoritmos de Machine Learning. Para este proyecto, se exploraron tres enfoques de modelado: una Red Neuronal Convolucional Unidimensional (CNN1D), un clasificador de Bosques Aleatorios (RandomForestClassifier) , SVM y su variante SVM GridSearch .Los datos transformados son "cargados" en la memoria para el entrenamiento de los modelos.

**RandomForestClassifier**

El **RandomForestClassifier** es un algoritmo de aprendizaje supervisado versátil que pertenece a la familia de los métodos de ensamblaje. Construye múltiples árboles de decisión durante el entrenamiento y genera la clase que es la moda de las clases (clasificación) o la predicción media (regresión) de los árboles individuales. Es conocido por su robustez, su capacidad para manejar grandes conjuntos de datos con muchas características y su habilidad para reducir el sobreajuste.

**Ventajas para este problema:**

* **Robustez a Ruido:** Al combinar múltiples árboles, el modelo se vuelve más robusto al ruido y a los valores atípicos presentes en las características.
* **Manejo de Dimensionalidad:** Efectivo para conjuntos de datos con un número elevado de características, como es el caso de las características pre-extraídas del UCI-HAR.
* **Menor Sobreajuste:** Tiende a sobreajustarse menos que un único árbol de decisión,.

El RandomForestClassifier se configuró inicialmente con sus parámetros por defecto, que son adecuados para una primera evaluación. Esto incluye un número de estimadores (árboles) predefinido y estrategias de división de nodos.

Texto

El contenido generado por IA puede ser incorrecto.

**Red Neuronal Convolucional Unidimensional (CNN1D)**

Procedemos a construir un modelo de red neuronal convolucional unidimensional (CNN1D) , utilizando la siguiente función

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

La función define una entrada de 128 puntos de tiempo y 9 canales (acelerómetro y giroscopio en 3 ejes). Se establecen tres bloques sucesivos de capas convolucionales y de pooling para detectar secuencias de movimiento, reducir la dimensionalidad y capturar jerarquías de características. Una capa Flatten convierte la salida a un vector, y un Dropout de 0.5 evita el sobreajuste. Finalmente, una capa de salida con activación softmax produce la distribución de probabilidad para las 6 clases de actividad del UCI-HAR

Tabla

El contenido generado por IA puede ser incorrecto.

Procedemos a entrenar el modelo, con la ejecución del siguiente bloque de código

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Luego del entrenamiento vemos que el accuracy asociado al modelo en la última época fue de 88.

Imagen que contiene Patrón de fondo

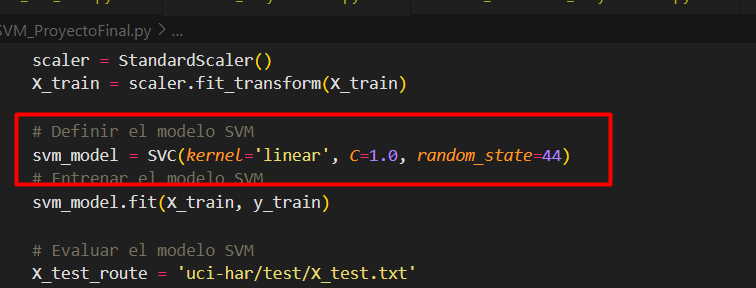
El contenido generado por IA puede ser incorrecto.

**SVM Y SVM GRIDSEARCH**

Uno de los algoritmos aplicados para abordar este proyecto, fue Support Vector Machine (SVM), un modelo de aprendizaje supervisado que busca encontrar el hiperplano óptimo que maximice la separación entre clases en un espacio multidimensional que tiene como objetivo clasificar correctamente la actividad que una persona está realizando en función de las características extraídas de sensores.

Para mejorar el rendimiento del modelo SVM, se emplea GridSearchCV, una técnica de búsqueda exhaustiva sobre un espacio de hiperparámetros. En particular, se optimizan parámetros como:

* C: parámetro de regularización que controla el equilibrio entre maximizar el margen y minimizar el error de clasificación para efectos de este modelo se lo estableció en 1.0
* Random\_state .Aesgura reproducibilidad en ciertas implementaciones subyacentes o para la inicialización de los datos y el valor usado es 44.
* kernel: tipo de función del núcleo (comúnmente 'linear', 'rbf', o 'poly') que transforma los datos para permitir una separación no lineal , para efectos de este proyecto se opto por linear esto implica que el modelo buscara un hiperplano lineal para las clases en el espacio de características.

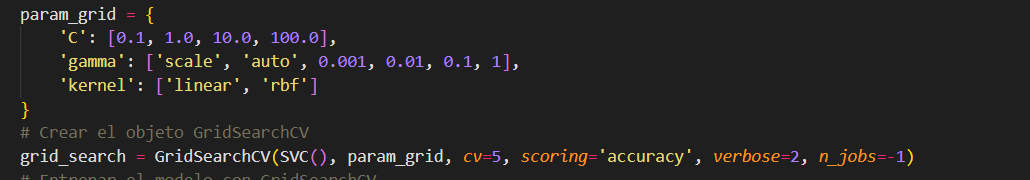


Para el caso de SVM GridSearchCV se implementaron los siguientes parámetros

**C**: Valores probados: [0.1, 1.0, 10.0, 100.0].

**gamma**: Valores probados: ['scale', 'auto', 0.001, 0.01, 0.1, 1]. El parámetro gamma define cuánto influye un solo ejemplo de entrenamiento; valores más grandes significan una influencia más cercana. scale y auto son estrategias predefinidas de scikit-learn.

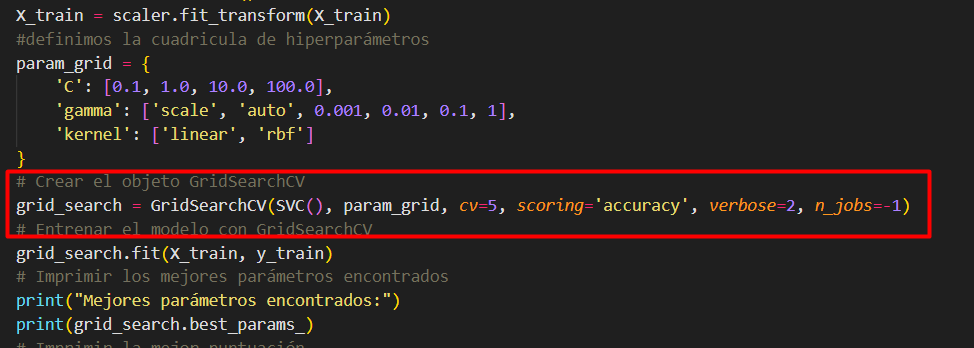
**kernel**: Kernels probados: ['linear', 'rbf']. El kernel rbf (función de base radial) es una opción popular para problemas no lineales, permitiendo al SVM encontrar separaciones no lineales en los datos.



Validación Cruzada: Se utiliza cv=5, lo que significa que la búsqueda de la mejor combinación de hiperparámetros se realiza mediante validación cruzada de 5 pliegues. Esto proporciona una estimación más robusta del rendimiento del modelo.

Métrica de Puntuación: La métrica utilizada para evaluar cada combinación de hiperparámetros es la accuracy (precisión).

Parallelización: n\_jobs=-1 se usa para aprovechar todos los núcleos de la CPU disponibles y acelerar el proceso de búsqueda en la cuadrícula.



Después de la búsqueda en la cuadrícula, se obtiene el best\_estimator\_ (el mejor modelo encontrado). Este modelo es el que tiene la mejor puntuación de precisión según la validación cruzada con los parámetros optimizados.

Texto

El contenido generado por IA puede ser incorrecto.

Finalmente, este best\_model se utiliza para realizar predicciones en el conjunto de prueba (X\_test\_scaled) y se evalúa de la misma manera que en el script anterior, con un reporte de clasificación, matriz de confusión y, adicionalmente, una curva ROC.

Texto

El contenido generado por IA puede ser incorrecto.

**Evaluación del modelo**

**RandomForestClassifier**

Al ser un modelo basado en árboles de decisión, no requiere un proceso iterativo de "épocas" y "lotes" como las redes neuronales. Se entrena en una sola pasada sobre el conjunto de datos

La evaluación se realiza directamente sobre el conjunto de prueba (o validación) una vez que el modelo ha sido ajustado. Las métricas clave incluyen precisión, recall, F1-score y la matriz de confusión.

Gráfico

El contenido generado por IA puede ser incorrecto.

El modelo alcanzó una precisión del 92%, demostrando su efectividad en la predicción de actividades humanas a partir de señales de sensores móviles. Posteriormente, se procederá a identificar la importancia de las variables en estas predicciones

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

El resultado es un gráfico de barras que indica la importancia asignada por el modelo a cada variable predictora al momento de inferir acciones humanas.

Gráfico, Gráfico de barras

El contenido generado por IA puede ser incorrecto.

Como paso final, se evalúa el rendimiento del modelo en la clasificación de las seis actividades humanas presentes en el dataset UCI-HAR mediante el uso de la matriz de confusión.

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

Gráfico, Gráfico de dispersión

El contenido generado por IA puede ser incorrecto.

El modelo RandomForestClassifier demostró un alto desempeño general en la clasificación de actividades humanas del dataset UCI-HAR con características preprocesadas, logrando precisión perfecta en 'LAYING'. No obstante, mostró confusión significativa entre 'SITTING' y 'STANDING', y aunque las actividades dinámicas ('WALKING', 'WALKING\_UPSTAIRS', 'WALKING\_DOWNSTAIRS') tuvieron buena precisión, existieron algunas confusiones cruzadas. Dado su consumo moderado de memoria y el tamaño del dataset, se decidió entrenar un modelo CNN1D como alternativa, utilizando la versión de datos crudos (directorio 'Inertial Signal') y realizando una validación inicial para detectar valores nulos o blancos.

**Red Neuronal Convolucional Unidimensional (CNN1D)**

Los gráficos de pérdida y precisión a lo largo de las épocas son cruciales para entender el comportamiento del modelo CNN1D.

Texto

El contenido generado por IA puede ser incorrecto.

Gráfico, Gráfico de barras

El contenido generado por IA puede ser incorrecto.

 Eje **X (Muestra):** Representa el índice de las muestras de datos en el conjunto de validación o prueba. En este gráfico, parece que se están visualizando alrededor de 3000 muestras.

 Eje **Y (Etiqueta de Actividad):** Muestra las etiquetas de las clases de actividad, que van del 0 al 5.

El gráfico 'Predicciones del Modelo vs. Valores Reales' ofrece una vista detallada del rendimiento del modelo CNN1D. Aunque es exitoso en la clasificación de actividades del dataset UCI-HAR, las principales áreas de mejora se encuentran en el manejo de transiciones y la discriminación precisa entre actividades estáticas, inherentemente más difíciles de diferenciar.

Gráfico, Gráfico de líneas

El contenido generado por IA puede ser incorrecto.

**Eje X (Época):** Representa el número de épocas o iteraciones completas sobre todo el conjunto de datos de entrenamiento. Va de 0 a 30 épocas

**Eje Y (Precisión):** Muestra la precisión del modelo, que es la proporción de predicciones correctas sobre el total de predicciones. Va de 0.0 a 1.0 (o 0% a 100%).

 Rendimiento **en Entrenamiento Excelente:** Un 98.04% de precisión en entrenamiento es muy alto, lo que sugiere que el modelo CNN1D es lo suficientemente potente como para aprender las características de los datos de actividad humana crudos.

 Preocupación **por el Sobreajuste:** La diferencia de casi 10 puntos porcentuales (98.04% vs 88.24%) entre la precisión de entrenamiento y validación al final del entrenamiento es una clara señal de **sobreajuste**. El modelo está memorizando el conjunto de entrenamiento en lugar de aprender patrones generalizables. Si bien un 88.24% es una buena precisión para el dataset UCI-HAR, podría ser mejor si se mitigara el sobreajuste.

Gráfico, Gráfico de líneas, Histograma

El contenido generado por IA puede ser incorrecto.

Este gráfico de pérdida evidencia que el modelo CNN1D está sobreajustándose severamente a los datos de entrenamiento. Aunque el modelo aprende muy bien los datos que ha visto, su capacidad para desempeñarse de manera efectiva en nuevos datos del UCI-HAR es significativamente limitada, lo que subraya la necesidad de implementar estrategias para mitigar el sobreajuste.

El resultado final es la siguiente tabla que resume los indicadores asociados al modelo CNN1D

Tabla

El contenido generado por IA puede ser incorrecto.

Durante la evaluación se genera la siguiente matriz de confusión, que nos permitir identificar que tan bien generaliza el modelo con datos nuevos.

Calendario

El contenido generado por IA puede ser incorrecto.

De la gráfica se desprende que las actividades dinámicas como WALKING (Clase 0), WALKING\_UPSTAIRS (Clase 1) y WALKING\_DOWNSTAIRS (Clase 2) están muy bien clasificadas, con pocos errores. Sin embargo, las actividades estáticas SITTING (Clase 3), STANDING (Clase 4) y LAYING (Clase 5) muestran confusiones significativas entre ellas, lo que indica una dificultad del modelo para distinguirlas debido a la similitud postural en las señales de los sensores.

Tras resultados insatisfactorios al intentar eliminar el sobreajuste con callbacks (lo que redujo la precisión del modelo al 15%), se optó por una estrategia diferente: añadir ruido durante el entrenamiento mediante la inclusión de un bloque de código específico

Imagen que contiene Texto

El contenido generado por IA puede ser incorrecto.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

El contenido generado por IA puede ser incorrecto.

Una vez realizado el ajuste y ejecutado nuevamente el entrenamiento del modelo los nuevos valores arrojados durante el entrenamiento y test del modelo CNN1D fueron los siguientes:

Gráfico, Gráfico de barras

El contenido generado por IA puede ser incorrecto.

Gráfico, Gráfico de líneas

El contenido generado por IA puede ser incorrecto.

Gráfico, Gráfico de líneas, Histograma

El contenido generado por IA puede ser incorrecto.

La reducción de la brecha entre la pérdida de entrenamiento y la pérdida de validación, junto con la disminución general de la pérdida de validación a un nivel mucho más bajo, son indicadores claros de que añadir ruido durante el entrenamiento fue exitoso al mitigar el sobreajuste que era evidente en los gráficos iniciales

Tabla

El contenido generado por IA puede ser incorrecto.

Se constata que el modelo no solo aprende de forma eficiente , sino que también es capaz de aplicar ese conocimiento de forma mas efectiva a datos nuevos y no vistos en el dataset uci-har

Calendario

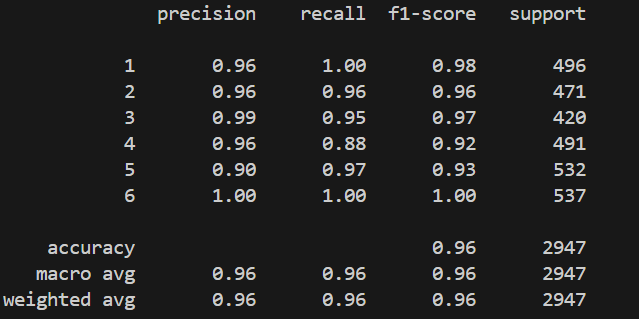
El contenido generado por IA puede ser incorrecto.

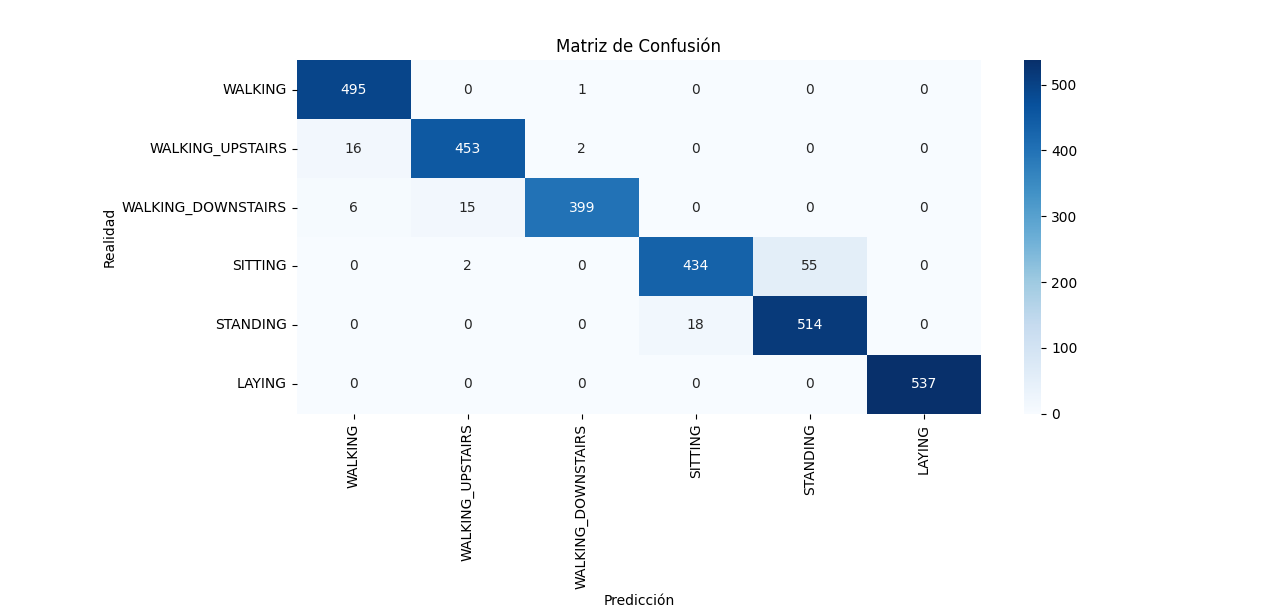
La nueva matriz de confusión confirma que las acciones para mitigar el sobreajuste fueron **exitosas en mejorar la capacidad de generalización del modelo**, especialmente para las actividades dinámicas y, de manera importante, para la clase "SITTING" (Clase 3). Aunque hay algunos compromisos en las precisiones de las clases "STANDING" y "LAYING" (Clases 4 y 5) y una nueva fuente de confusión para la Clase 2 con la Clase 5, la **reducción general en la pérdida de validación y el aumento en la precisión de validación** es un indicador de que el modelo es ahora más robusto y generalizable para el reconocimiento de actividad humana con el dataset UCI-HAR.

Análisis de Resultados para el modelo SVM vs SVM con aplicación de gridsearch:

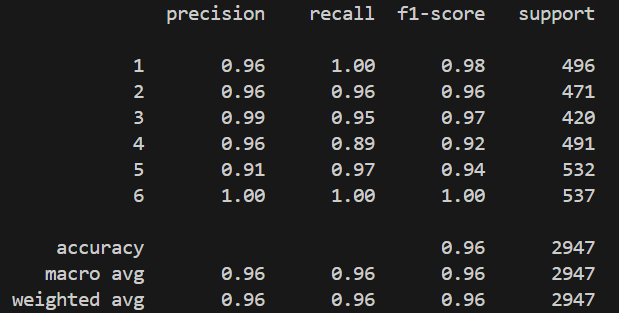
Reportes de clasificación y matrices de confusión:

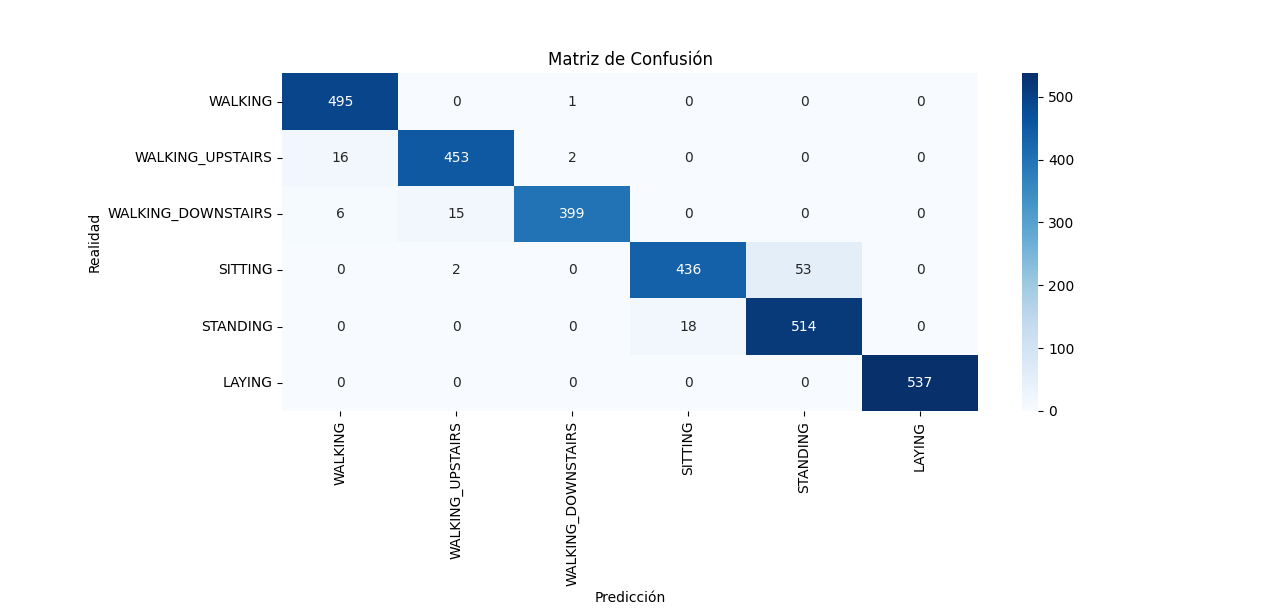
SVM





SVM CON GRIDSEARCH





Hablando de los reportes de clasificación se tiene que los mejores resultados se dan en la categoría 6 que es Laying (tendido), donde en ambos casos se tuvo una precisión y recall del 100% que implica que el modelo clasificó todos los eventos de esta categoría sin error, lo que sugiere un sobreajuste en la misma. Sin embargo, como en las otras categorías no hay coincidencia absoluta en la evaluación de las mismas, tomamos el modelo como válido.

En la categoría 1 también tenemos un recall de 100% pero una precisión menor a uno lo que indica que todos los eventos de esta categoría los clasificamos correctamente , sin embargo, alguno(s) de otra categoría los clasificamos en esta en algún momento, pero sin alarmas ya que la precisión es 0.96.

La categoría que más ha presentado errores en clasificarse es la 4 (siting – sentado), confundiéndose sobre todo con la categoría standing, el 0.04% vienen a ser los falsos positivos para esta categoría en cuanto a accuracy y, en cuanto al recall 11% de las muestras no han sido clasificadas convirtiéndose en falsos negativos para esa categoría.

Todas las categorías logran un buen número de elementos acertados , se nota equilibrado el modelo en este aspecto.

Hablando de costo computacional, SVM aplicando gridsearch va a ser más alto en ese aspecto, pero es porque en cada combinación de hiperparametros el mismo tomara su tiempo en evaluar el modelo en su accuracy y pérdida. Así que es incomparable este aspecto, todo depende de la malla de parámetros que seteamos que a su vez deriva en la cantidad de combinaciones posibles de los mismos para tener que ser evaluadas.

En los resultados finales, la ganancia con la aplicación de Gridsearch ha resultado mínima, pero es una mejora al final, se recuperan algunos datos de los falsos negativos de algunas categorías.

La precisión general de 0.96 denota un modelado consistente en su rendimiento, cabe mencionar que el valor es muy bueno como resultado del modelo.

Hablando de la matriz de confusión, la de SVM con gridsearch, se tienen 22 falsos positivos que el modelo predijo como clase 1 pero son 16 de clase 2 y 6 de clase 3. Así, se tienen 17 falsos positivos en la clase 2 , 3 falsos positivos en clase 3, 18 en clase 4 y 53 en clase 5.

Los falsos negativos en clase 1 son 1 que el modelo clasifico como clase 3 y es en realidad clase 1, 18 en clase 2, 21 en clase 3, 55 en clase 4 y 18 en clase 5.

La parte más comprometida , se puede decir, es las confusiones que el modelo está dando entre la clase 4 y 5.