**Para el desarrollo de este proyecto nos apoyaremos del proceso KDD Knowledge Discovery in Databases que básicamente contiene fases ya definidas que nos van a permitir descubrir conocimiento a partir de una fuente de datos**

**1 Fase selección de fuente de datos**

Para el desarrollo del proyecto se uso como fuente de datos el data set uci-har que se encuentra en el repositorio <https://archive.ics.uci.edu/dataset/240/human+activity+recognition+using+smartphones> , el cual contiene informacion de registros de 30 sujetos de prueba que realizaban actividades diarias mientras llevaban un teléfono inteligente montado en la cintura con sensores inerciales incorporados a saber : Acelerómetro y Giroscopio , el registro de las señales emitidas por estos dispositivos se dio en las dimensiones o ejes X;Y;Z.

Se enfoco básicamente en 6 actividades que debian realizar las personas : CAMINAR, SUBIR, BAJAR, SENTAR, DE PIE, TUMBADO , acciones que eran medidas mediante el acelerómetro y giroscopio incorporado en el dispositivo móvil , se registraron aceleración lineal y velocidad angular triaxiales a una frecuencia constante de 50HZ , los datos obtenidos se dividieron en dos conjuntos de datos de forma aleatoria ,70% para datos de entrenamiento y 30% para datos de prueba.

Las señales de los sensores se preprocesaron mediante la aplicación de filtros de ruido y se muestrearon en ventanas deslizantes de ancho fijo de 2.56 segundos y un solapamiento de 128 lecturas por ventana .La señal de aceleración del sensor que incluye componentes gravitacionales y de movimiento corporal se separo mediante un filtro de paso bajo Butterworth

De lo anteriormente expuesto se desprende que el dataset utilizado para este proyecto ya se encuentra dividido en conjuntos de entrenamiento y test ; también podemos constatar que dicho data set contiene dos versiones de dataset , el primero corresponde a los registros preprocesados , donde cada columna es una feauture procesada y contiene 561 columnas por fila , normalizadas y listas la aplicación de modelos como RandomForest , SVM, KNN y el segundo contiene las señales crudas de los sensores antes de ser transformadas en características , que contiene series temporales de Aceleración del cuerpo, Aceleración por gravedad y Velocidad angular .

Dado que la primera versión contiene los datos ya procesados y separados para entrenamiento y test , utilizaremos esta primera versión para entrenar un modelo de clasificación Random Forest Classifier , para el resto de modelos a ser abordados en este proyecto se hará uso de la segunda versión del dataset , la cual contiene señales crudas de los dispositivos.

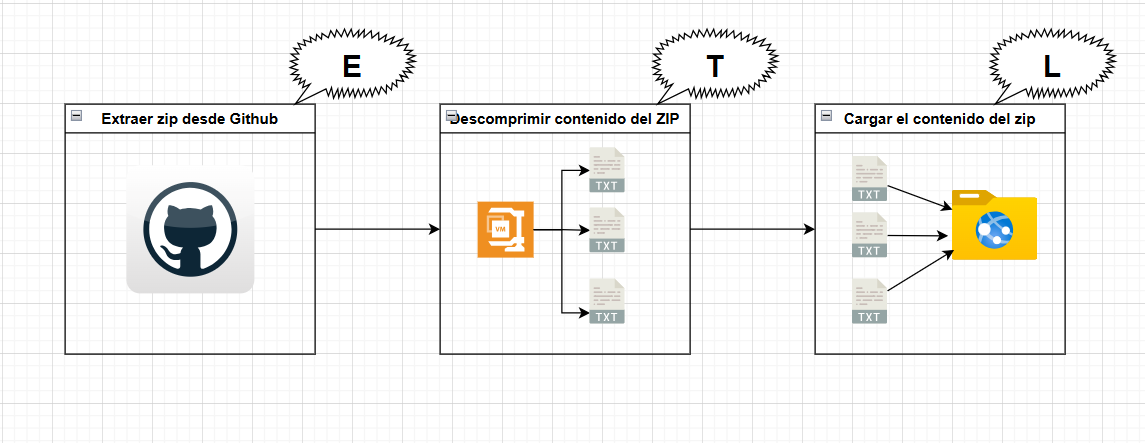
* 1. Proceso ETL para carga del dataset desde GITHUB

Para este proyecto el dataset de uci-har se encuentra comprimido en formato zip , dentro del cual se encuentran las dos versiones de los datase a ser utilizadas(procesados, datos crudos) , en virtud de esto se procede a generar un proceso ETL básico que :

Extraiga el comprimido uci-har.zip que se encuentra en un repositorio github

Descomprima el archivo zip para poder utilizarlo

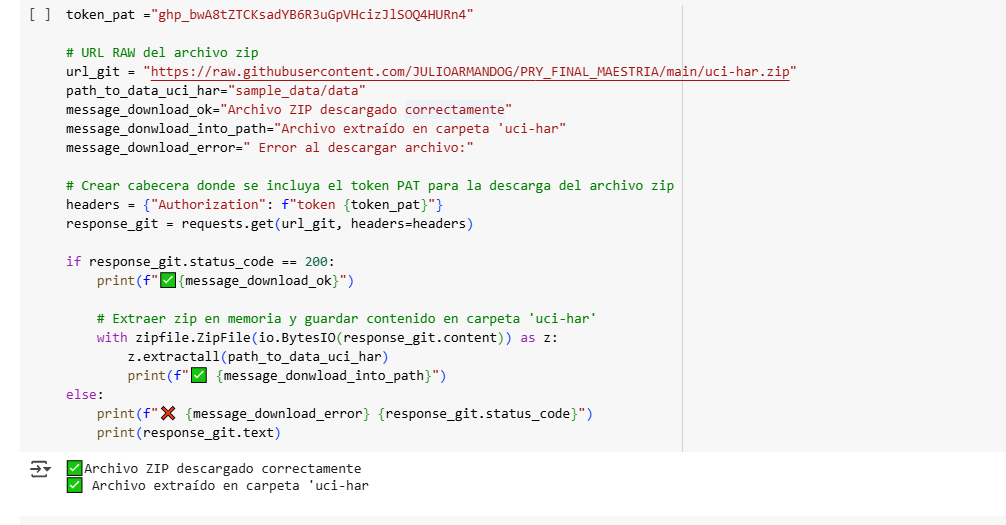
Cargar el contenido del archivo zip en un repositorio local con el propósito de que sirva como insumo para el análisis de datos.



Para la construcción de este ETL se utilizo como lenguaje de programación Python y las siguientes librerías :

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

El siguiente bloque de código muestra la implementación del proceso ETL básico dentro del proyecto



**Fase de preprocesamiento**

Una vez identificado el insumo de datos con el cual se va a entrenar el modelo procedemos a ejecutar el análisis exploratorio de datos EDA , como punto de partida para el análisis de la información contenida en el data set con el propósito de entender su contenido , detectar patrones , valores , faltantes, anomalías en los datos y relacione s entre variables antes de construir los modelos , así como el posible desbalance de clases predictoras . Si bien es cierto la versión del dataset que contiene los datos procesados implicaría no tener la necesidad de validar datos faltantes o anomalías en los datos , es importantes que no obviemos esta fase asumiendo como verdad absoluta que el data set se encuentra sin errores, mas bien como buena practica vamos a ejecutar las fase del análisis exploratorio de datos sin saltarnos ningún paso.

En lo referente a esta fase , cabe acotar que las tareas ejecutadas van a diferir en función de la naturaleza de los datos con los cuales vamos a entrenar el modelo y del modelo propiamente dicho.En virtud de lo expuesto anteriormente el presente proyecto de titulación hará uso de los seguimiento modelos a saber:

CNN1D , SVM\_Gridsearch,SVM, RandomForestClassifier, ya que durante los entrenamientos estas arrojaron los mejores índices de acuracy

RandomForestClassifier .- Es uno de los algoritmos de clasificación supervisado presentes en la librería scikit-learn , es muy popular debido a su facilidad de uso, robustes y precisión.Cada árbol utiliza un subconjunto diferente del dataset para ser entrenado , al final se hace la predicción en función del voto mayoritario entre los diferentes arboles entrenados.

Para este algoritmo de entrenamiento , según lo dicta la documentación vamos a utilizar la versión del dataset uci-har que contiene los datos procesados estadísticamente y divididos en subconjuntos de entrenamiento y prueba , que que como lo indica la documentación esta es la versión que se debe utilizar cuando se pretende utilizar modelos de entrenamiento como RandomForestClassifier , que fácilmente pueden alcanzar un acuracy elevado “Anguita, D., Ghio, A., Oneto, L., Parra, X., & Reyes-Ortiz, J. L. (2013). A Public Domain Dataset for Human Activity Recognition Using Smartphones. ESANN 2013 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges, Belgium. <https://www.esann.org/sites/default/files/proceedings/legacy/es2013-84.pdf>”

Procedemos con la carga de datos de entrenamiento y prueba desde los repositorios donde nuestro proceso ETL dejo cargada la información del dataser uci-har



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



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 vacia procedemos a realizar un análisis exploratorio de datos mostrando las diferentes acciones presentes dentro del dataset mediante el uso de una nube de palabras:



Texto

El contenido generado por IA puede ser incorrecto.

Como vemos en la figura anterior los diferentes estados presentes dentro del dataset uci-har son :

LAYIN , STANDING,WALKING,WALKING\_UPSTAIRS,WALKING\_DOWNSTAIRS

A partir de este momento ya podemos delimitar el alcance del proyecto a la predicción de estas seis acciones presentes dentro del dataset utilizado para este proyecto de titulación.

Una vez identificadas las posibles categorías que se pueden predecir dentro del dataset , procedemos a identificar el balanceo de clases dentro del dataset de entrenamiento mediante el uso del siguiente código.

Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

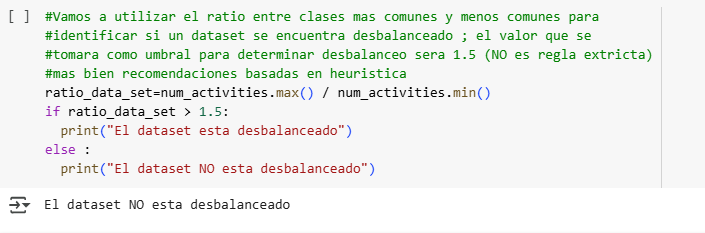
El contenido generado por IA puede ser incorrecto.

Obeteniendo el siguiente resultado

Gráfico, Gráfico de barras

El contenido generado por IA puede ser incorrecto.

El grafico anterior nos permite inferir de forma preliminar si el dataset de entrenamiento se encuentra desbalanceado ; a primera vista esto no sucede , pero con el proposito de basar nuestra decision en numeros que respalden dicha aserveración vamor a proceder a identificar el mismo mediante el calculo del ratio entre clases



Como parte del analisis exploratorio procedemos a visualizar

como estan distribuidos cada uno de los puntos dentro de cada

clase a predecir mediante el uso de PCA

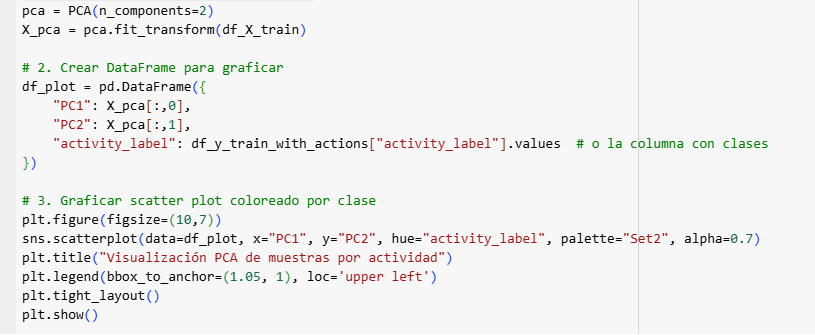
Reducción de dimensionalidad a 2D usando PCA permitiendo bajar de 561 caracteristicas que posee el dataset

a 2 o 3 componente principales que capturen de mejor forma la variabilidad entre los datos

permitiendo identificar de mejor forma agrupacion o mezclas entre las diferentes clases

mostrar el separacion entre las diferentes clases , ya que las identificamos por diferentes colores.

Detectar outlier o patrones inusuales



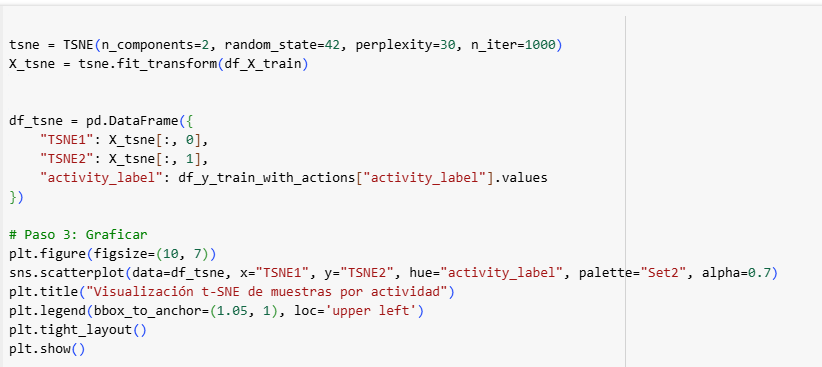
De la ejecución del anterior código obetenemos el siguiente grafico

Gráfico, Gráfico de dispersión

El contenido generado por IA puede ser incorrecto.

En funcion del grafico mostrado en la figura anterior se puede indentificar un solapamiento entre las clases SITTING Y STANDING , lo cual podria indicar una dificultar para clasificarlas, LAYING, WALKING\_UPSTAIRS y WALKING\_DOWNSTAIRS estan bastante diferenciadas

Procedemos a la utilizacion de una tecnica de reducción de dimensionalidad no lineal con el proposito de mejorar la visualización de la separación entre clases dentro del data set



De la ejecución del código anterior obtenemos el siguiente resultado

Gráfico, Gráfico de dispersión

El contenido generado por IA puede ser incorrecto.

El grafico anterior muestra una mejor separacion entre clases que aquella basada en el uso de PCA , dado el uso de reducción no lineal que preserva la relaciones locales , menos solapamiento mejor separacion entre clases

Sittgin y Standing aunque aparecian solapadas en PCA , ahora esta mas separadas e identificables pero cercanas lo cual es correcto , laying se encuenta bien separado , mientras que WALKING, WALKING\_UPSTAIRS, WALKING\_DOWNSTAIRS aparecen como cluster bien separados

Como parte final procedemos a mostrar información estadística del dataset y validamos la existencia de registros duplicados que puedan alterar el resultado del entrenamiento



Fase de Trasnformacion

En esta fase procedemos al escalamiento de la información con el propósito de hacerlos aptos para el uso de los algoritmos de entrenamiento.

Interfaz de usuario gráfica, Texto, Aplicación

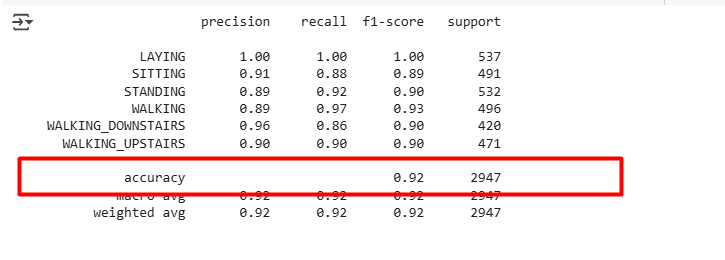
El contenido generado por IA puede ser incorrecto.

Ejecutamos el entrenamiento del modelo , las predicciones , y ek reporte de clasificación que forman parte de la fase de minieria de datos donde alpicamos los diferentes algoritmos de entrenamiento para este caso RandomFOrestClassifier con el propósito de encontrar pratrones que nos permitan hacer predicciones efetivas de las acciones humanas basados en las señales emitidas por el giroscoipi y acelerometeo de los smarthphone

Texto

El contenido generado por IA puede ser incorrecto.

El resultado de esta fase es un reporte de clasficiacion donde el índice del acuracy nos indicara que tan efectivo es nuestro modelo a la hora de inferir acciones humanas basadas en datos diferentes a los de entrenamiento



Observamos un acuracy del 92% que indica que el modelo es efectivo a la hora de predecir actividades humanas basadas en las señales emitidas por los sensores de los dispositivos móviles.

Como parte de la fase de interpretación procedemos a identificar la importancia que tuvieron las variables a la hora de predecir las acciones humadas mediante el siguiente bloque de código

Interfaz de usuario gráfica, Texto

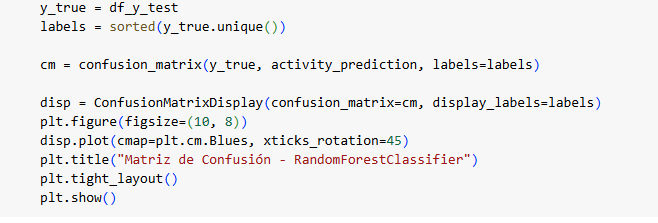
El contenido generado por IA puede ser incorrecto.

El resultado es una grafica de barras que indica la importancia que el modelo le da a las variables predictoras al momento de predecir acciones humanas como el que se muestra a continuación.

Gráfico, Gráfico de barras

El contenido generado por IA puede ser incorrecto.

Como paso final procedmos a el rendimiento del modelo cuando intenta clsificar las seis actividades humanas contenidas dentro del dataset uci-har



Gráfico, Gráfico de dispersión

El contenido generado por IA puede ser incorrecto.

.El modelo RandomForestClassifier alcanzó un desempeño general alto al clasificar las actividades humanas en el dataset UCI-HAR utilizando características previamente procesadas.

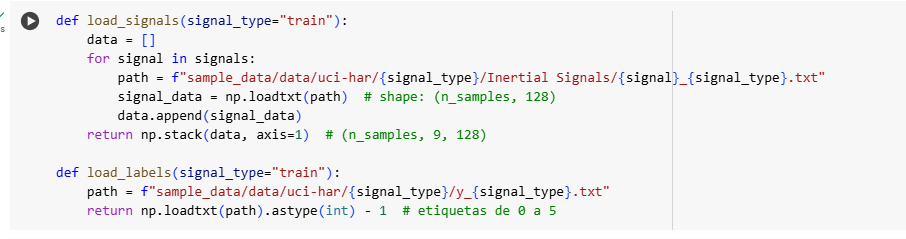
Precisión perfecta en la clasificación de la actividad LAYING, lo que sugiere que esta clase presenta un patrón distintivo fácilmente reconocible.

Se observa confusión significativa entre las actividades SITTING y STANDING, probablemente debido a similitudes en las posturas y señales captadas por los sensores durante estas acciones estáticas.

Las actividades dinámicas como WALKING, WALKING\_UPSTAIRS y WALKING\_DOWNSTAIRS presentan un buen nivel de precisión, aunque con algunas confusiones cruzadas entre movimientos verticales (UPSTAIRS vs. DOWNSTAIRS).

Dado que el modelo RandomForestClassifier tiene un consumo de memoria moderado ya que el dataset utilizado posee menos de 10.000 registros , se opto por entrenar otros modelos que nos permitan tener varias opciones sobre las cuales decidir como es el caso de CNN1D ,

Al igual que el primer algoritmo utilizado , después de la carga procedemos a una validación inicial del dataset para identificar posibles valores nulos o blancos , con la diferencia que para este algoritmo se trabaja con la segunda versión de data que contiene el data set uci-har ; aquellos que contienen los datos crudos de los sensores depositados dentro del directorio Inertial Signal



Interfaz de usuario gráfica, Aplicación

El contenido generado por IA puede ser incorrecto.

|  |  |
| --- | --- |
| Identificador | Actividad asociada |
| 0 | WALKING |
| 1 | WALKING\_UPSTAIRS |
| 2 | WALKING\_DOWNSTAIRS |
| 3 | SITTING |
| 4 | STANDING |
| 5 | LAYING |

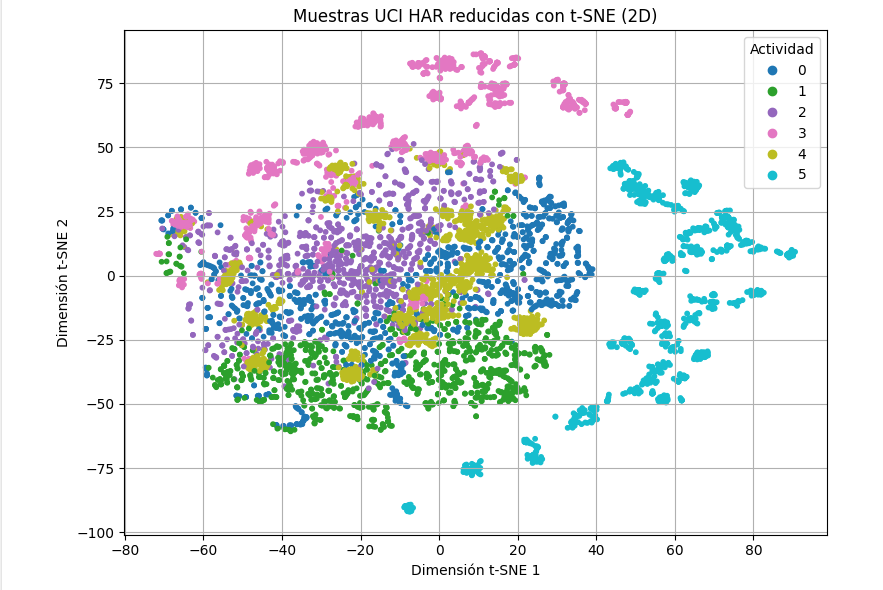
Una vez realizada la validación se procede con una exploración inicial de los datos mediante el uso PCA para poder identificar como están agrupadas las diferentes actividades humanas dentro de los datos crudos del dataset , tomand como referencia la tabla previamente mostrada ,obteniendo el siguiente resultado.

Gráfico, Gráfico de dispersión

El contenido generado por IA puede ser incorrecto.

Como se puede ver en el grafico anterior una exploración inicial de los datos crudos el dataset uci-har ha logrado separar de forma parcial las diferentes actividades , y tal cual como ocurre con los datos procesados , los datos crudos del dataset uci-har muestran un solapamiento entre las clases SITTING Y STANDING , lo cual podria indicar una dificultar para clasificarlas, LAYING, WALKING\_UPSTAIRS y WALKING\_DOWNSTAIRS estan bastante diferenciadas

Al igual que lo hicimos con el primer algoritmo mejoramos la exploración inicial haciendo uso de una técnica no lineal TNSE obteniendo el siguiente resultado



Como podemos observar se observar de mejor manera la separación de grupos de actividades en el dataset uci-har de aplicado a los datos crudos.

Una vez realizada la exploración inicial sobre los datos crudos procedemos a evaluar si existe desbalanceo a nivel de actividades a predecir mediante el siguiente código

Interfaz de usuario gráfica, Texto, Correo electrónico

El contenido generado por IA puede ser incorrecto.

El resultado de la ejecución de esta función es el siguiente grafico

Gráfico, Gráfico de barras

El contenido generado por IA puede ser incorrecto.

Según el diagrama de barras , aparentemente no existe desbalanceo de clases dentro del dataset asociado a los datos en crudo; Sin embrago nuevamente

vamos a utilizar el ratio entre clases más comunes y menos comunes para

identificar si un dataset se encuentra desbalanceado , el valor que se

tomara como umbral para determinar desbalanceo será 1.5 (NO es regla extricta) mas bien recomendaciones basadas en heurística mediante la ejecución del siguiente código.

Interfaz de usuario gráfica, Texto, Aplicación

El contenido generado por IA puede ser incorrecto.

Como se puede observar en la grafica anterior , el dataset no se encuentra desbalanceado.

Luego de haber realizado las validaciones pertinentes procedemos a normalizar y formatear los datos en crudo del dataset para que estos puedan ser utilizados en el entrenamiento de redes nuronales convolucionales de 1 dimension mediante la ejecución del siguiente bloque de código.

Interfaz de usuario gráfica, Texto

El contenido generado por IA puede ser incorrecto.

En forma general , el bloque anterior de código realiza lo siguiente:

Organiza los datos para que dentro del dataset se encuentra bajo la forma (n\_samples, 9, 128) a la forma aceptada como entrada para una Conv1D (n\_samples, 128, 9) donde :

128 = pasos de tiempo

9 = canales/sensores (acelerómetro y giroscopio en 3 ejes)

Normaliza cada una de las señales individuales a media 0 y desviación estándar 1, mejorando la estabilidad del entrenamiento y el rendimiento del modelo

Hereda de tf.keras.utils.Sequence, lo cual permite usar esta clase directamente en model.fit() y model.evaluate(), optimizando el uso de memoria y permitiendo paralelización.

Ofrece una opción integrada para separar los datos en entrenamiento y validación mediante el parámetro validation\_split, simplificando la gestión de datos durante el entrenamiento.

AL culminar esta fase , procedemos a construir un modelo de red neuronal convolucional unidimensional (CNN1D) , utilizando la siguiente función:

Interfaz de usuario gráfica, Texto

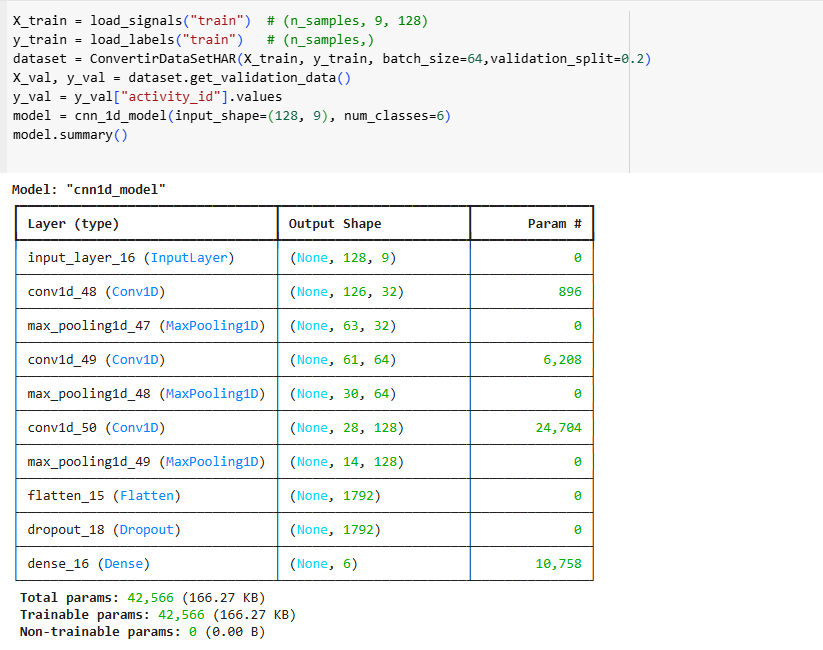
El contenido generado por IA puede ser incorrecto.

En la función anterior se define como entrada 128 puntos de tiempo por muestra y 9 canales (3 ejes para acelerómetro y giroscopio) .

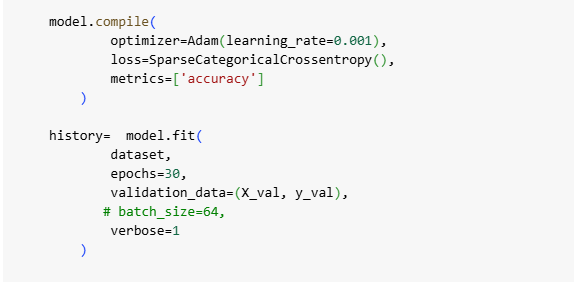
Definimos capas convolucionales y de pooling en 3 bloques sucesivos para detectar secuencias de movimiento , reducir dimensionalidad temporal y capturar jerarquías de características.

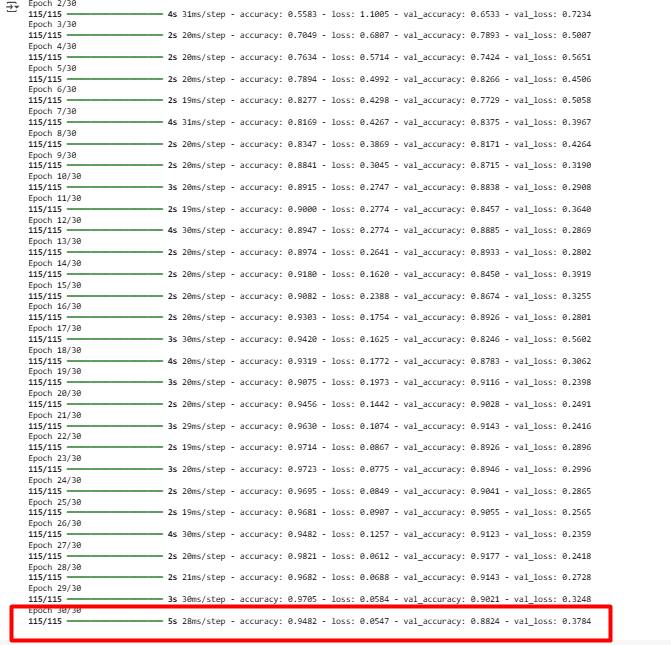
Una capa de regulación que convierte la salida final en un vector y un dropout de 0.5 que evita el sobreajuste apagando aleatoriamente neuronas durante el entrenamiento.

Una capa de salida con activación softmax, que produce una distribución de probabilidad sobre las clases de actividad (6 en total en UCI-HAR)



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





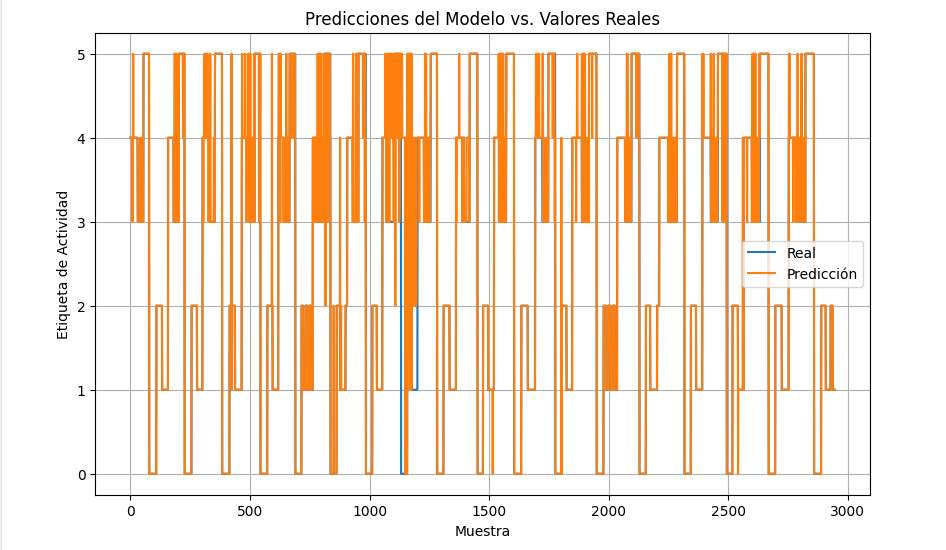
Luego del entrenamiento vemos que el accuracy asociado al modelo en la ultima época fue de 88

Luego del entrenamiento procedemos a evaluar nuestro modelo mediante la ejecución del siguiente bloque de código.

Texto

El contenido generado por IA puede ser incorrecto.

El resultado de la ejecución del anterior bloque de código son las siguientes graficas asociadas a los siguientes indicadores:



 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 perspectiva granular del rendimiento del modelo. Muestra que, si bien el CNN1D es en gran parte exitoso en la clasificación de actividades en el dataset UCI-HAR, las áreas clave para la mejora residen en el manejo de transiciones y la discriminación más fina entre las actividades estáticas que son 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 grafica anterior se resume lo siguiente:

Clase 0 (WALKING): Muy bien clasificada (474). Pocos errores, la mayoría (22) confundidos con la clase 3.

Clase 1 (WALKING\_UPSTAIRS): Excelente clasificación (433). Algunos errores menores con las clases 2 (24) y 3 (14).

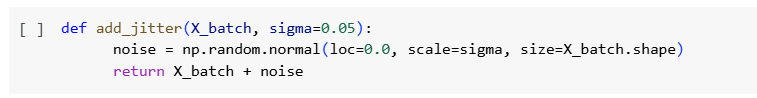
Clase 2 (WALKING\_DOWNSTAIRS): Bastante bien clasificada (390). Algunas confusiones con la clase 0 (10), 1 (8), 3 (4), 4 (6) y 5 (2).

Clase 3 (SITTING): Buena clasificación (368), pero presenta una cantidad significativa de errores. Es confundida notablemente con la clase 4 (80) y la clase 5 (43). Esto sugiere que el modelo tiene dificultades para distinguir "SITTING" de "STANDING" (si la clase 4 es STANDING) y "LAYING" (si la clase 5 es LAYING), lo cual es un problema común dado la similitud postural.

Clase 4 (STANDING): Muy bien clasificada (478). Es confundida principalmente con la clase 3 (38) y la clase 5 (16). Esta confusión con la clase 3 (SITTING) es esperable.

Clase 5 (LAYING): Muy bien clasificada (441). Presenta confusiones significativas con la clase 3 (51) y la clase 4 (42). Nuevamente, esto tiene sentido ya que "LAYING", "SITTING" y "STANDING" son actividades estáticas que pueden tener señales de sensor similares

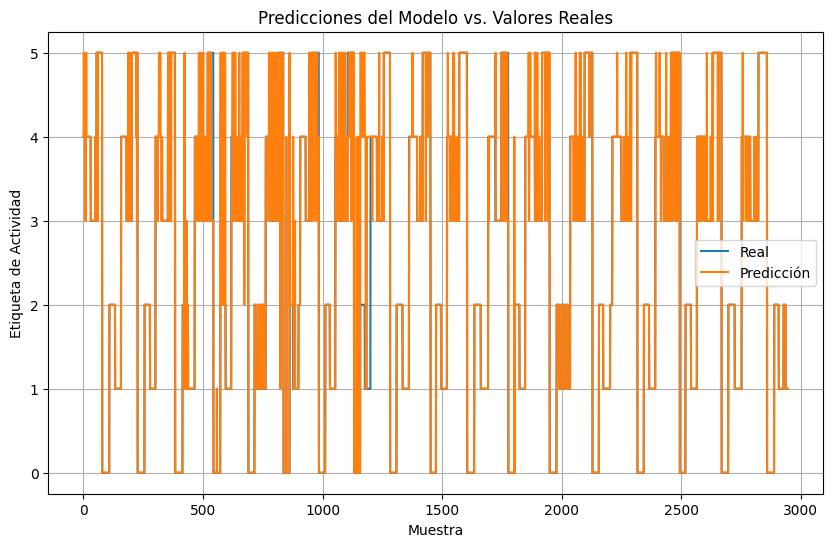
Dados los resultados anteriores se procedió a eliminar el sobreajuste utilizando callbacks durante el entrenamiento , lo cual ocasionó que el modelo se degradara y su accuracy descendiera a un 15% , lo cual fue evidencia que ese no era el camino para eliminarlo . En virtud de estos últimos resultados se opto por añadir ruido durante el entrenamiento mediante la adicción del siguiente bloque de código.

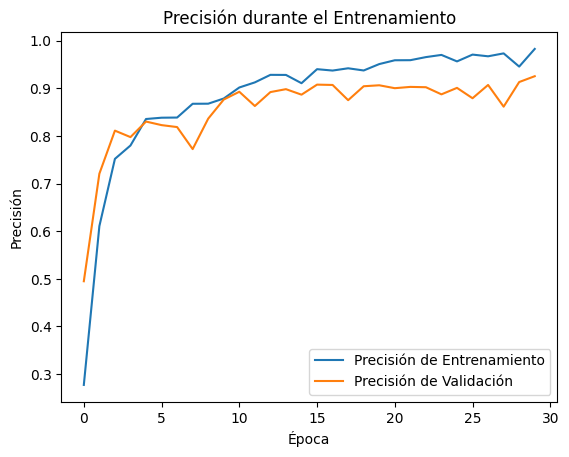


Interfaz de usuario gráfica, Texto, Aplicación, Correo electrónico

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 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.