**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.

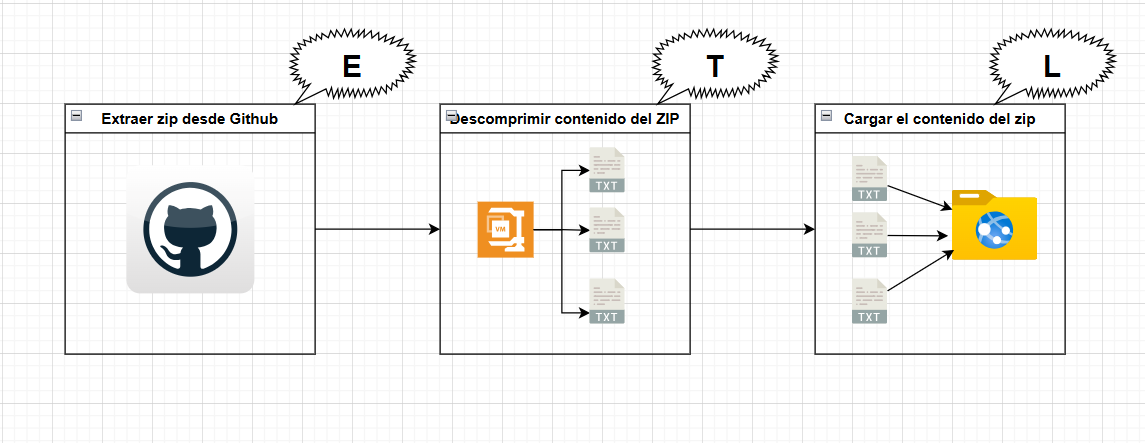
* 1. Proceso ETL para carga del dataset desde GITHUB

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

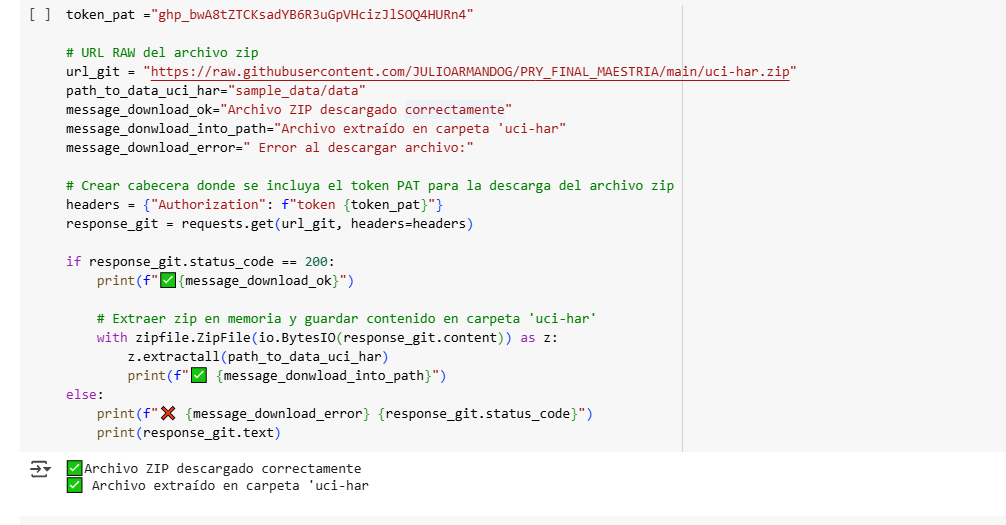
Cargar el contenido del archivo zip en un repositorio local con el propósito de que sirva como insumo para el análisis de datos.



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

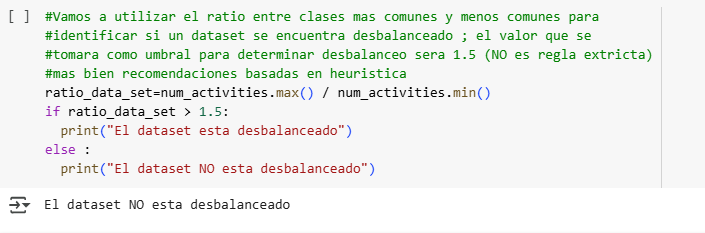
El contenido generado por IA puede ser incorrecto.

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

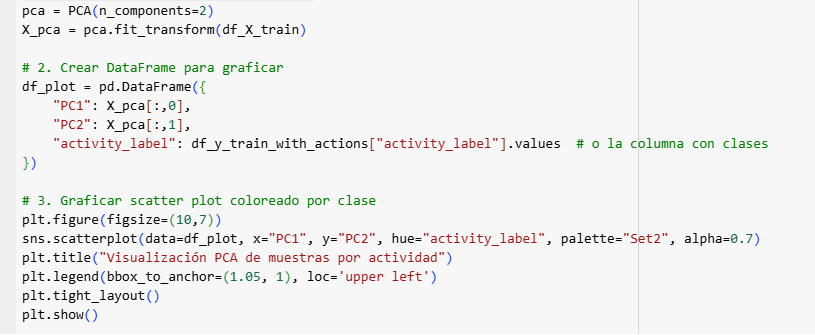
Reducción de dimensionalidad a 2D usando PCA permitiendo bajar de 561 caracteristicas que posee el dataset

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



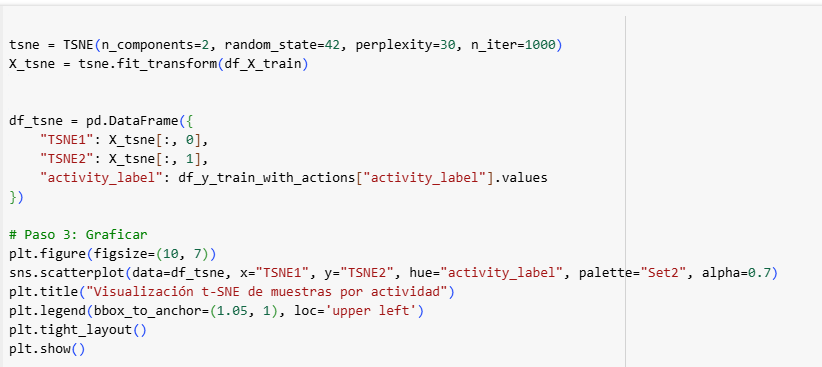
De la ejecución del anterior código obetenemos el siguiente grafico

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

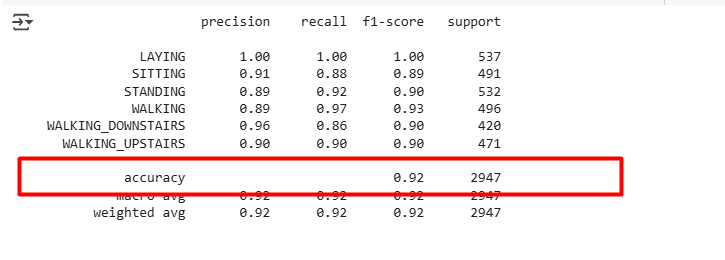
El contenido generado por IA puede ser incorrecto.

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

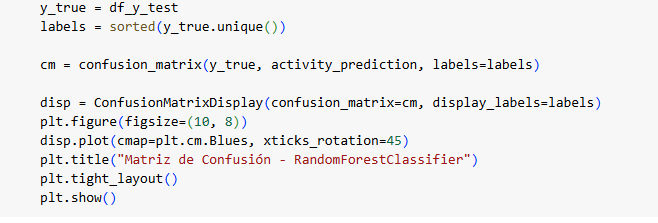
El contenido generado por IA puede ser incorrecto.

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).

En general, el modelo muestra un buen equilibrio entre sensibilidad y especificidad, aunque la diferenciación entre ciertas clases puede beneficiarse del uso de modelos que capturen mejor la información temporal (como LSTM o CNN-1D).