1. Esquema arquitectura modelo de machine learning

De modo esquemático, se presenta en la Figura 1 el flujo de trabajo esperado a lo largo del PT3 previamente mencionado.

Diagrama

Descripción generada automáticamente

Figura 1. Esquema del flujo de trabajo del proceso inteligente esperado

En ésta, se pueden apreciar como punto de partida los datos del experimento recogidos a través del sensor seleccionado, así como información complementaria de los datos personales de cada sujeto. Con esto, se buscará, tal y como se menciona previamente, la forma de detectar las distintas fases de la marcha, de tal modo que, junto a los datos personales iniciales, se proceda con los distintos pasos relativos al modelado inteligente. Todo esto, requerirá una fase previa exhaustiva de pre-procesado y preparación de los datos.

Dicho modelado se realizará haciendo uso de distintos modelos posibles y parametrizaciones de algoritmos de aprendizaje supervisado de clasificación, y será validado utilizando las técnicas que se consideren adecuadas mediante la selección de la métrica o métricas de relevancia.

Así, dicho modelo será la base para la generación de un sistema experto de prevención de caídas que, junto con todo lo previo, será incluido en un sistema de visualización que presente la información relevante detectada.

En cuanto a las herramientas de programación a utilizar, se basará el desarrollo en el lenguaje Python, haciendo uso a priori de la versión 3.7 de éste, y utilizando librerías asociadas al análisis de datos e inteligencia artificial, como *numpy*, *pandas*, *scipy*, *sklearn* o *xgboost*.

1. Infraestructura del sistema experto

El sistema experto queda cimentado dentro de una arquitectura de microservicios que se puede ver esquematizada en Figura 2. De forma abreviada, existe un *frontend* accesible por los usuarios que, dependiendo de su rol, podrán acceder a diversas características sobre visualización y ejecución de modelos para las pruebas de los sujetos que se lanzarán contra un sistema de colas de trabajo. El *backend* son una serie de aplicaciones desplegadas en contenedores que realizarán diversas funciones para el desarrollo del flujo de operaciones de aprendizaje automático. A groso modo los modelos pueden aprender de los datos de los que se vaya alimentando vía pruebas de sujetos y poder ofrecer resultados más óptimos.

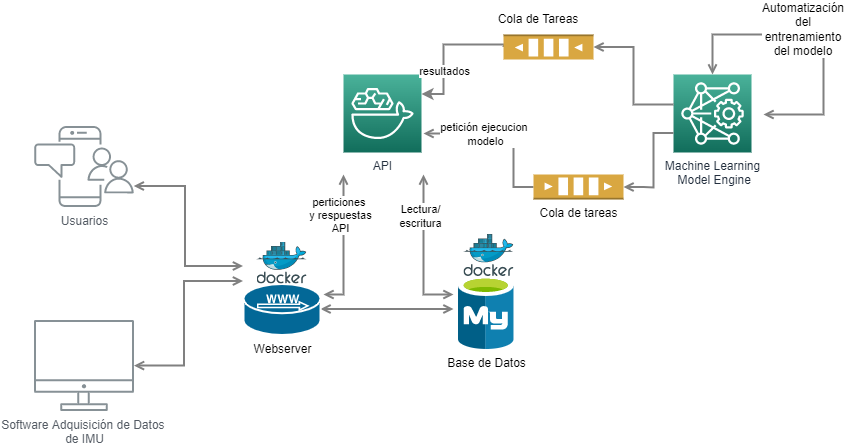
**

Figura 2. Arquitectura de microservicios para PreFall

1. Explicación de algunas herramientas de aprendizaje automático utilizadas.

En la Figura 3 pueden se muestran algunos algoritmos que se pueden utilizar con la herramienta scikit-learn donde se implementan clasificación, regresión y algorimos de *clustering* como Support-Vector Machines (SVM), Random Rorests, Gradient Boosting, K-means y DBSCAN. Estas herramientas funcionan en conjunto perfectamente con librerías numéricas como NumPy y SciPy.

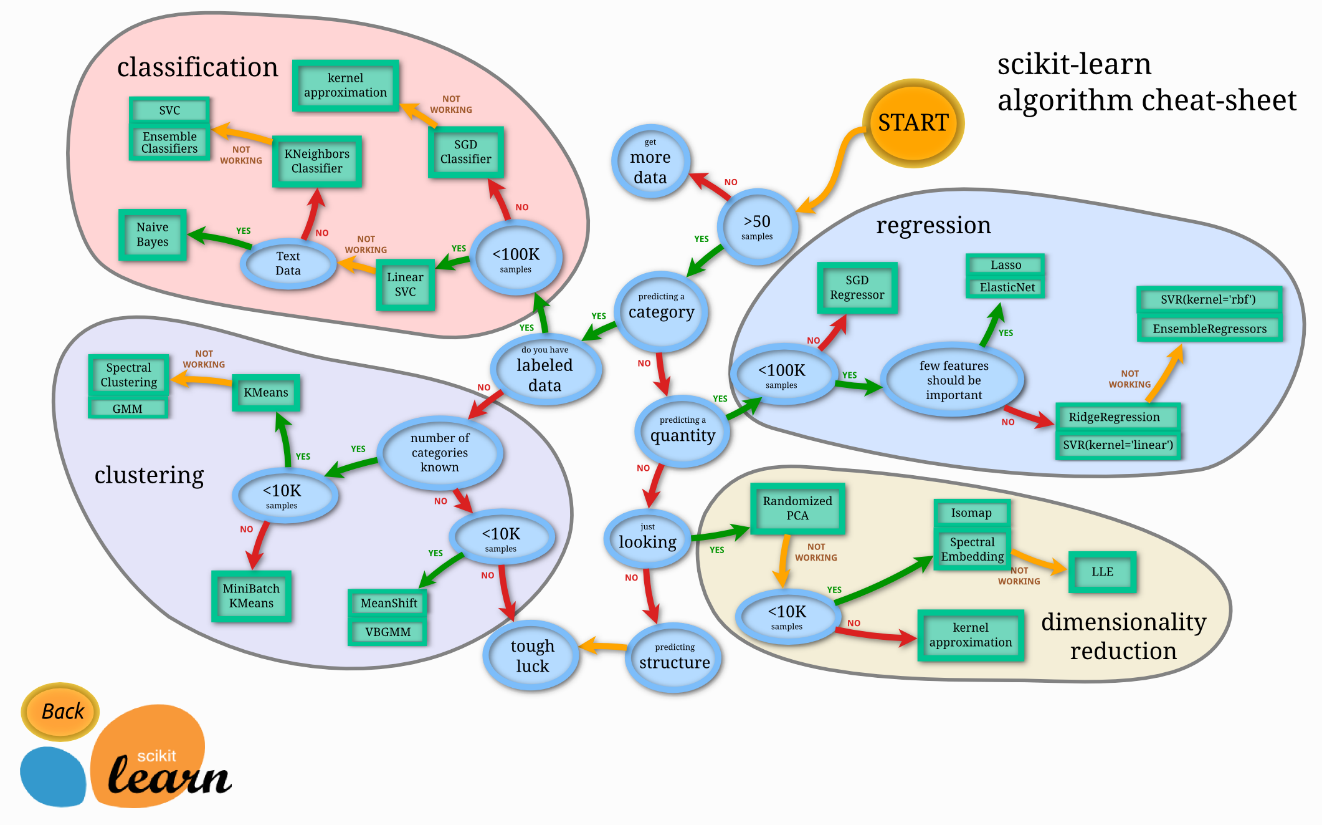


Figura 3. Tipos de algorimos