



Pre-Fall – Sistema inteligente para la prevención y predicción de caídas

E3.3 – Validación de los modelos de aprendizaje automático

Proyecto	Pre-Fall – Sistema inteligente para la prevención y predicción de caídas
Entregable	E3.3 – Validación de los modelos de aprendizaje automático

Contenido

Contenido	1
Índice de tablas	2
Índice de figuras	3
1 Introducción	4
2 Actualización del conjunto de datos inicial y selección de variables	5
3 Modelos de aprendizaje automático	6
3.1 Regresión logística.....	6
3.2 Naive Bayes	7
3.3 Árboles de decisión	7
3.4 Random Forest	8
3.5 Redes neuronales	9
3.6 K-vecinos	10
3.7 XGBoost.....	11
3.8 Máquinas Vector Soporte	12
4 Validación de los modelos de aprendizaje automático supervisado	14
4.1 Métricas de clasificación	14
4.2 Resultados comparativos de clasificación.....	15
5 Sistema experto de prevención de caídas y evaluación de la eficacia de rehabilitación....	17
6 Conclusiones.....	19

Índice de tablas

Tabla 1. Resumen de los datos generado a partir de los originales.....	5
Tabla 2. Resumen de la distribución de la clase de la variable objetivo en los datos sintéticos. .	5
Tabla 3. Resumen de los resultados de los modelos de clasificación.	16

Índice de figuras

Figura 1. Representación gráfica de la función sigmoidea.	6
Figura 2 Ejemplo de árbol de decisión. Imagen obtenida de: https://www.xoriant.com/blog/decision-trees-for-classification-a-machine-learning-algorithm7	
Figura 3. Esquema de una neurona artificial.....	9
Figura 4. Esquema del modelo K-vecinos para una clasificación binaria. Imagen obtenida de: https://medium.com/machine-learning-researcher/k-nearest-neighbors-in-machine-learning-e794014abd2a	11
Figura 5. Ejemplo gráfico de una máquina vector soporte	12
Figura 6. Ciclo de funcionamiento del sistema experto de recomendación.....	18

1 Introducción

Este entregable está enmarcado en la tarea “T3.3: Validación de los modelos”, perteneciente al paquete de trabajo “PT3 – Sistema experto de prevención de caídas” dentro del proyecto PRE-FALL. A lo largo de este documento se presentarán los resultados de la validación de los modelos diseñados y desarrollados a lo largo del paquete de trabajo.

2 Actualización del conjunto de datos inicial y selección de variables

Todos los desarrollos aquí presentados se han realizado basándose en los datos obtenidos y procesados tal y como se indica en el entregable “E3.1 – Procedimiento de depuración y preprocesado de los datos”. Tras aplicar las técnicas comentadas en dicho entregable, se han generado 200 ficheros sintéticos por cada fichero real, generando un total de 400 ficheros.

ID	Tamaño medio(KB)	Duración media (segundos)	Número de ficheros
Synthetic USER1	41,5	5,03	200
Synthetic USER2	41,6	5,30	200
TOTAL	16620	2066	400

Tabla 1. Resumen de los datos generado a partir de los originales.

Como se comentó en dicho entregable, estos constan de valores interpolados a partir de los ficheros originales con características similares. Cada uno de estos ficheros es procesado siguiendo los pasos del entregable E3.1 de manera que a cada fichero se le asocian la duración y valores medios de los datos en los 3 ejes de las 4 fases de la marcha identificadas, así como la variable objetivo que indica si la persona asociada a dicho fichero tiene riesgo o no de caída.

La distribución de esta variable objetivo se ha realizado de forma aleatoria, pero asegurando que cada mitad de los datos corresponda a cada clase. Esto se ha hecho así para crear un modelo equilibrado dado que trabajamos con poca información y la mayor parte es sintética.

ID	Riesgo de caída	Número de ficheros
Synthetic USER1	Si	100
Synthetic USER1	No	100
Synthetic USER2	Si	100
Synthetic USER2	No	100
TOTAL		400

Tabla 2. Resumen de la distribución de la clase de la variable objetivo en los datos sintéticos.

3 Modelos de aprendizaje automático

En la literatura existen multitud de modelos diferentes para tareas de clasificación por lo que se han seleccionado un subconjunto de ellos junto con una serie de valores para sus respectivos hiperparámetros de manera que se pueda determinar el que mejor funcione para este problema en específico. En concreto, se han considerado los casos presentados en los siguientes subapartados.

3.1 Regresión logística

El modelo de regresión logística utiliza una función sigmoide para transformar una variable continua en un valor entre 0 y 1, que se interpreta como la probabilidad de que la instancia pertenezca a la clase positiva. El modelo utiliza un conjunto de coeficientes para ponderar las características o variables de entrada y ajustar la curva sigmoide para que se ajuste a los datos de entrenamiento. La función sigmoide se representa:

$$g(z) = \frac{1}{1 + e^{-z}}$$

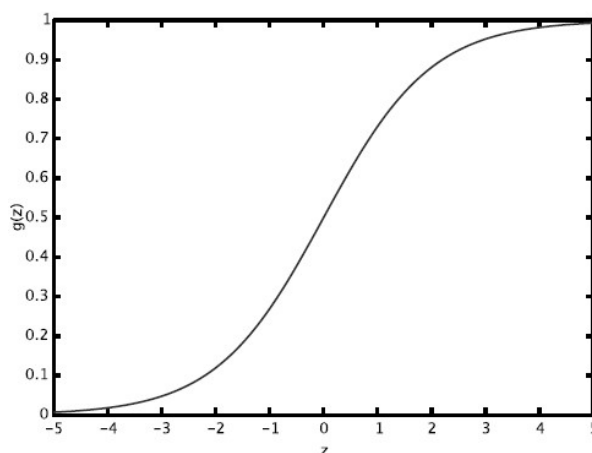


Figura 1. Representación gráfica de la función sigmoidea.

En esta función a medida que z crece, la función tiende a 1, es decir, $g(x) \rightarrow 1$ si $z \rightarrow \infty$ y al contrario, a medida que z decrece, la función tiende a 0, es decir, $g(x) \rightarrow 0$ si $z \rightarrow -\infty$.

Durante el entrenamiento, el modelo ajusta los coeficientes utilizando un método de optimización como el descenso de gradiente estocástico (SGD). Una vez que se han ajustado los coeficientes, se pueden utilizar para hacer predicciones sobre nuevas instancias.

Los hiperparámetros elegidos para este modelo dentro de la librería de scikit-learn son “C” y “penalty”, ambos encargados de controlar la fuerza y el tipo de regularización del modelo, es decir, de evitar el sobreajuste del mismo.

3.2 Naive Bayes

El modelo de Naive Bayes se basa en el teorema de Bayes utilizando la probabilidad condicional para calcular la probabilidad de que una instancia pertenezca a una clase determinada. Para cada clase hace uso de una distribución de probabilidad diferente, que se estima a partir de los datos de entrenamiento. Durante el entrenamiento, el modelo calcula la probabilidad de cada característica dado cada clase y utiliza estas probabilidades para calcular la probabilidad de cada clase dado un conjunto de características.

Durante la predicción, el modelo calcula la probabilidad de que una instancia pertenezca a cada clase y selecciona la clase con la probabilidad más alta como la predicción.

Para este caso se ha considerado el hiperparámetro “var_smoothing” que es un valor positivo que se agrega a la varianza de las características en la estimación de la densidad de probabilidad. Un valor mayor de “var_smoothing” significa una mayor suavidad en la estimación de la densidad de probabilidad, lo que puede ayudar a evitar problemas de división por cero, pero también puede disminuir la precisión del modelo.

3.3 Árboles de decisión

Los modelos de clasificación con árboles de decisión se basan en la construcción de un conjunto de nodos y ramas que los conectan que divide el conjunto de datos en ramas basadas en las características de los datos. Cada nodo interno representa una decisión del algoritmo y una partición del árbol hacia la siguiente decisión hasta llegar a los nodos finales u hojas que representan la predicción final sobre la clase de la variable objetivo. Cada una de las ramas puede entenderse como una expresión lógica que dependiendo de si el resultado de la misma es verdadero o falso, el algoritmo irá en la dirección de uno de los nodos hijos de manera que el árbol en su conjunto se trate de una serie de expresiones lógicas unidas mediante un operador “and”.

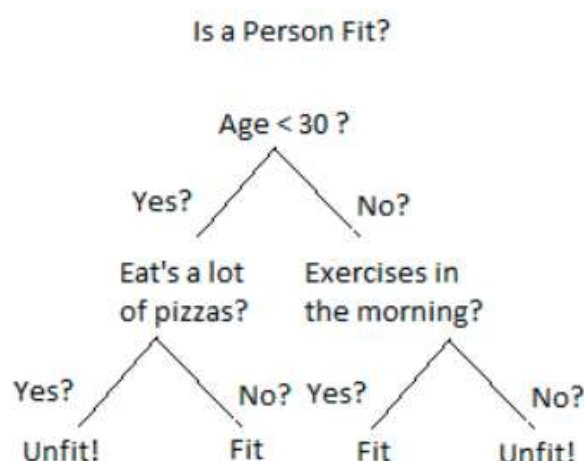


Figura 2 Ejemplo de árbol de decisión. Imagen obtenida de: <https://www.xoriant.com/blog/decision-trees-for-classification-a-machine-learning-algorithm>

Para la construcción del árbol de decisión, se han elegido los siguientes hiperparámetros:

- El hiperparámetro "max_depth" controla la profundidad máxima del árbol de decisión. Un valor más alto puede llevar a un árbol más complejo y propenso al sobreajuste, mientras que un valor más bajo puede llevar a un árbol menos complejo y con un rendimiento inferior.
- El hiperparámetro "min_samples_leaf" establece el número mínimo de ejemplos requeridos en cada hoja del árbol de decisión. Un valor más alto puede llevar a un árbol más generalizado, mientras que un valor más bajo puede llevar a un árbol más específico.
- El hiperparámetro "min_samples_split" establece el número mínimo de ejemplos requeridos para dividir un nodo interno del árbol de decisión. Un valor más alto puede llevar a un árbol más generalizado, mientras que un valor más bajo puede llevar a un árbol más específico.
- El hiperparámetro "criterion" establece la función de medición de calidad utilizada para evaluar la calidad de una división en el árbol de decisión. Las opciones disponibles en scikit-learn son "gini" y "entropy". "gini" se utiliza para la impureza Gini, mientras que "entropy" se utiliza para la ganancia de información. En general, no hay una opción mejor que la otra y se recomienda probar ambas opciones para ver cuál funciona mejor para el conjunto de datos específico.

3.4 Random Forest

El modelo de clasificación Random Forest se basa en la combinación de múltiples árboles de decisión para construir un modelo más robusto y generalizable.

Cada árbol de decisión en el modelo de Random Forest se construye a partir de una muestra aleatoria de los datos de entrenamiento y utiliza un subconjunto aleatorio de las características. De esta manera, cada árbol es ligeramente diferente y tiene una mayor diversidad. Durante la predicción, el modelo de Random Forest combina las predicciones de todos los árboles de decisión para producir una predicción final.

Para definir el Random Forest se ha decidido considerar los siguientes hiperparámetros:

- El hiperparámetro "n_estimators" establece el número de árboles de decisión que se construyen en el modelo. Cuanto mayor sea el número de árboles, más preciso será el modelo, pero también aumentará el tiempo de entrenamiento y la complejidad del modelo.
- El hiperparámetro "max_features" establece el número máximo de características que se utilizan para dividir cada nodo interno del árbol de decisión en cada árbol de decisión. Cuanto menor sea el número de características, mayor será la diversidad entre los árboles y mejor será la generalización del modelo, pero también puede disminuir el rendimiento.

- El hiperparámetro "max_depth" controla la profundidad máxima de cada árbol de decisión en el modelo. Si se establece en "None", los árboles se expandirán hasta que todas las hojas contengan menos de "min_samples_split" muestras. Un valor más alto puede llevar a un modelo más complejo y propenso al sobreajuste, mientras que un valor más bajo puede llevar a un modelo menos complejo y con un rendimiento inferior.
- El hiperparámetro "bootstrap" establece si se utiliza o no el muestreo de arranque para construir cada árbol de decisión. El muestreo de arranque puede mejorar la precisión y la estabilidad del modelo, pero también puede aumentar la complejidad.

3.5 Redes neuronales

Las redes neuronales son modelos estadísticos adaptativos que se asemejan a la estructura de las neuronas del cerebro humano. Estas reciben una señal de entrada desde unas conexiones denominadas dendritas y generan una señal de salida que envían a través de otra conexión llamada axón que a su vez interactúa con las dendritas de otras neuronas mediante pesos sinápticos.

La estructura de las redes neuronales se basa en el uso de nodos que son núcleos computacionales denominados neuronas unidos entre sí en forma de un grafo que puede ser dirigido o no dirigido. De manera similar al proceso biológico, cada una de estas neuronas recibe datos de entrada de un conjunto de unidades de entrada y genera una señal de salida que envía a través de un conjunto de unidades de salida habiendo otro conjunto de unidades intermedias encargadas de procesar la lógica de las operaciones.

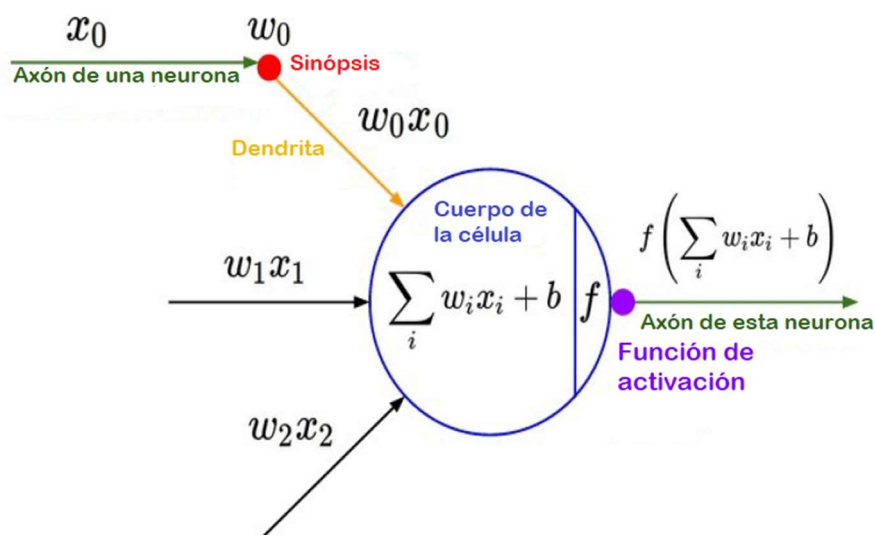


Figura 3. Esquema de una neurona artificial.

De cara a entrenar la red neuronal se han tenido en cuenta los siguientes hiperparámetros:

- "hidden_layer_sizes": este parámetro determina el número y el tamaño de las capas ocultas en la red neuronal. El valor predeterminado es una sola capa oculta con 100

neuronas, pero se pueden especificar diferentes números de capas y diferentes tamaños de neuronas para personalizar la arquitectura de la red.

- "activation": este parámetro determina la función de activación que se utiliza en las neuronas de la red. Las opciones incluyen "identity" (sin función de activación), "logistic" (función de activación sigmoidea), "tanh" (tangente hiperbólica) y "relu" (unidad lineal rectificada).
- "solver": este parámetro determina el algoritmo utilizado para optimizar los pesos de las conexiones entre las neuronas. Las opciones incluyen "lbfgs" (método de optimización de Broyden-Fletcher-Goldfarb-Shanno), "sgd" (descenso de gradiente estocástico) y "adam" (adaptive moment estimation).
- "alpha": este parámetro controla la fuerza de regularización de los pesos de la red neuronal para evitar el sobreajuste. Un valor mayor de alfa dará como resultado una mayor regularización y una red neuronal más simple.
- "learning_rate": este parámetro determina la tasa de aprendizaje utilizada por el algoritmo de optimización para ajustar los pesos de la red neuronal. Un valor más pequeño de "learning_rate" resultará en una optimización más lenta pero posiblemente más precisa.

3.6 K-vecinos

Este algoritmo se basa en la idea de que los elementos similares están cerca en el espacio de características. El objetivo es clasificar una nueva instancia en una de las clases existentes en función de sus vecinos más cercanos en el conjunto de entrenamiento. En otras palabras, el algoritmo busca los k ejemplos más cercanos (vecinos) en el espacio de características, donde "k" es un número entero positivo, y determina la clase de la nueva instancia en función de la clase predominante en esos k vecinos.

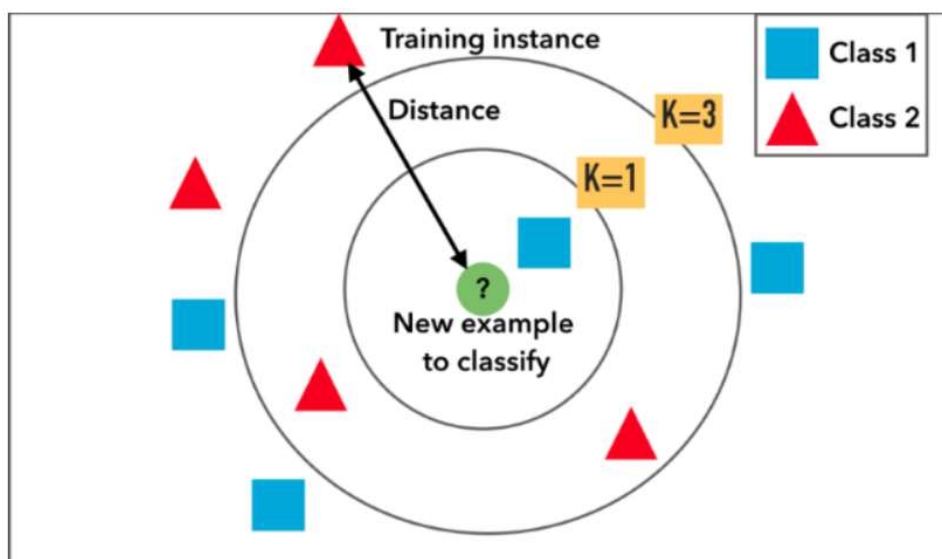


Figura 4. Esquema del modelo K-vecinos para una clasificación binaria. Imagen obtenida de: <https://medium.com/machine-learning-researcher/k-nearest-neighbors-in-machine-learning-e794014abd2a>

El hiperparámetro principal de este modelo es la variable “k”. Un valor k más grande puede ser más preciso, pero también puede ser más costoso computacionalmente.

3.7 XGBoost

El algoritmo XGBoost (*Extreme Gradient Boosting*) se basa en la técnica de *boosting*, que combina varios modelos débiles en un modelo fuerte. Para ello se entrenan una serie de árboles de decisión y se combinan para formar un modelo predictivo. Cada árbol se entrena utilizando una muestra aleatoria de los datos de entrenamiento y se utiliza para predecir la clase de una nueva instancia. Los árboles se construyen secuencialmente, y cada nuevo árbol se enfoca en los errores del modelo anterior.

El algoritmo XGBoost se diferencia de otros algoritmos de *boosting* al utilizar un enfoque de gradiente estocástico (SGD) para optimizar la función de pérdida. Esto significa que XGBoost utiliza una función de pérdida personalizada para cada problema de clasificación y ajusta los pesos de las muestras de entrenamiento de forma iterativa para minimizar la función de pérdida.

Los hiperparámetros que se han considerado para optimizar este modelo son:

- “booster”: este hiperparámetro se utiliza para especificar el tipo de modelo que se va a entrenar con XGBoost. Puede ser “gbtree” para árboles de decisión o “gblinear” para modelos lineales.
- “eta”: la tasa de aprendizaje, también conocida como “learning rate”, controla el tamaño de los ajustes que se realizan en cada iteración. Un valor más bajo significa que se realizarán ajustes más pequeños, lo que puede ayudar a evitar el sobreajuste.
- “gamma”: el parámetro de regularización gamma controla cuánto se requiere que una hoja de un árbol de decisión reduzca la función de pérdida. Un valor más alto significa

que se requerirá una reducción de pérdida más significativa antes de agregar una nueva hoja al árbol.

- “max_depth”: este hiperparámetro controla la profundidad máxima de cada árbol de decisión. Un árbol más profundo puede capturar relaciones más complejas en los datos, pero también puede ser más propenso al sobreajuste.
- “n_estimators”: el número de árboles de decisión que se van a entrenar. Un mayor número de árboles puede mejorar el rendimiento del modelo, pero también puede aumentar el tiempo de entrenamiento.

3.8 Máquinas Vector Soporte

Las máquinas de vector soporte se basan en el concepto de espacios linealmente divisibles mediante hiperplanos. El objetivo es encontrar la mejor separación posible entre dos clases mediante la búsqueda de un hiperplano óptimo en un espacio de características de alta dimensión. Este hiperplano se define como el que maximiza la distancia entre los puntos más cercanos de cada clase, también conocido como el margen de separación.

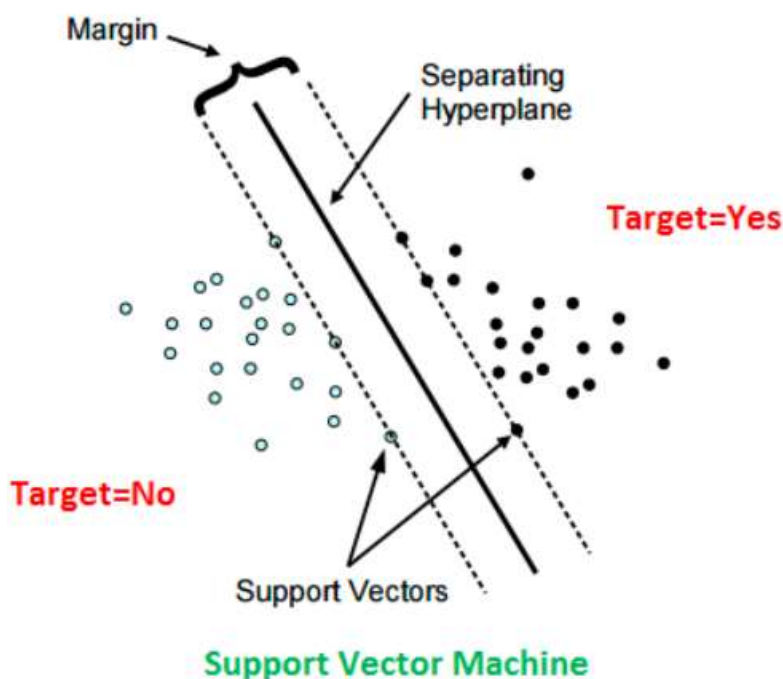


Figura 5. Ejemplo gráfico de una máquina vector soporte

Se han considerado los siguientes hiperparámetros para optimizar el modelo SVM:

- “C”: Es un parámetro de regularización que controla el límite de la violación del margen permitido. Un valor más alto de C indica que se permiten más violaciones al margen, lo que resulta en un modelo más complejo. Un valor más bajo de C da como resultado un modelo más simple.
- “kernel”: Es la función utilizada para transformar los datos de entrada en un espacio de características de mayor dimensión.

- “gamma”: Es un parámetro que controla la forma de la función de kernel utilizada para transformar los datos de entrada en un espacio de características de mayor dimensión. Un valor más alto de gamma significa que la función de kernel tendrá un efecto más localizado, lo que puede llevar a un ajuste demasiado ajustado de los datos de entrenamiento y aumentar el riesgo de sobreajuste. Un valor más bajo de gamma permite un efecto más global de la función de kernel, lo que resulta en un modelo menos complejo.

4 Validación de los modelos de aprendizaje automático supervisado

En esta segunda fase, se ha procedido a presentar los resultados asociados al modelado de aprendizaje automático supervisado de clasificación binaria. De nuevo, se ha basado dicho desarrollo en los datos obtenidos a lo largo del proyecto tal y como se indica en el entregable “E3.1 – Procedimiento de depuración y preprocesado de los datos”.

Por otro lado, tal y como se ha denotado en el apartado previo, se han seleccionado diferentes modelos de clasificación con diferentes parametrizaciones. En concreto, se han considerado los siguientes casos:

- Regresión Logística: con 7 valores "C" entre -3 y 3 en escala logarítmica y "penalty" con valores 'lasso' o 'ridge'.
- Naive Bayes Gausiano: con 100 valores "var_smoothing" entre 0 y -9.
- Random forest: con el hiperparámetro "n_estimators" entre 1 y 7, "max_features" entre 1 y 15, "max_depth" con valores 10, 20, 30, 40, 50 y 'None' y "bootstrap" con valores True o False.
- Neural Network: con valores "hidden_layer_sizes": '(50,50,50)', '(50,100,50)' y '(100,)', "activation" valores 'tanh' o 'relu', "solver" valores 'sgd' o 'adam', "alpha" con posibles valores '0.0001' o '0.05' y "learning_rate" 'constant' o 'adaptive'
- K-Vecinos: con número de vecinos de 1 a 100.
- Árbol de decisión: con "max_depth" entre 1 y 15, "min_samples_leaf" con 5 valores entre 10 y 50, "min_samples_split" con 5 valores entre 10 y 50 y "criterion" con valores 'gini' o 'entropy'.
- XGBoost: con valores de "booster" 'gbtree', 'gblinear' o 'dart', "eta" con valores 0.3, 0.6 o 1, "gamma" valores 25 o 50, "max_depth" valores 1, 5 o 10 y "n_estimators" con valores 50 o 100.
- Máquina Vector Soporte: con valores "C" 0.1, 1, 10 o 100, "gamma" con posibles valores 1, 0.1, 0.01 o 0.001 y "kernel" valores 'rbf', 'poly' o 'sigmoid'.

4.1 Métricas de clasificación

Existen varias métricas para medir el rendimiento de los modelos de *machine learning* de clasificación. De entre ellas destacan: *accuracy*, *recall*, especificidad y precisión.

El *accuracy* mide la proporción de instancias clasificadas correctamente por el modelo en relación al total de instancias. Su fórmula es la siguiente:

$$Accuracy = \frac{\text{Número de instancias clasificadas correctamente}}{\text{Total de instancias}}$$

El *recall* o tasa de verdaderos positivos es la proporción de casos verdaderamente positivos (VP) de la clase que fueron identificados correctamente, respecto al total de casos positivos. Su fórmula es la siguiente:

$$\text{Recall} = \frac{\text{Verdaderos positivos}}{(\text{Verdaderos positivos} + \text{Falsos negativos})}$$

La especificidad es la proporción de casos verdaderamente negativos (VN) de la clase que fueron identificados correctamente, respecto al total de casos negativos. Su fórmula es la siguiente:

$$\text{Especificidad} = \frac{\text{Verdaderos negativos}}{(\text{Verdaderos negativos} + \text{Falsos positivos})}$$

La precisión es la proporción de verdaderos positivos (VP) de la clase entre los casos clasificados como positivos, es decir, cuántos casos realmente positivos fueron identificados correctamente. Su fórmula es:

$$\text{Precisión} = \frac{\text{Verdaderos positivos}}{(\text{Verdaderos positivos} + \text{Falsos positivos})}$$

Dado que actualmente el modelado se ha realizado sobre datos sintéticos y el proceso de entrenamiento de modelos del punto anterior requiere el uso de multitud de modelos y parámetros, se ha seleccionado únicamente el *accuracy* como métrica para establecer el mejor modelo para los datos.

4.2 Resultados comparativos de clasificación

A continuación, se presentan los resultados después de realizar el proceso de *Grid Search* con los diferentes modelos y parámetros mencionados anteriormente en formato de tabla indicando el modelo, la mejor combinación de los hiperparámetros propuestos y el valor de la métrica *accuracy* obtenido.

Modelo	Mejor set de hiperparámetros	Accuracy
Regresión logística	'C': 1.0, 'penalty': 'l2'	0.520
Naive Bayes	'var_smoothing': 1.0	0.550
<i>Random Forest</i>	'bootstrap': True, 'max_depth': 10, 'max_features': 1, 'n_estimators': 1	0.530
<i>Neural Network</i>	'activation': 'relu', 'alpha': 0.05, 'hidden_layer_sizes': (50, 100, 50), 'learning_rate': 'constant', 'solver': 'adam'	0.505
<i>K-vecinos</i>	'n_neighbors': 29	0.530
<i>Árbol de decisión</i>	'criterion': 'gini', 'max_depth': 2, 'min_samples_leaf': 40, 'min_samples_split': 10	0.595
XGBoost	'booster': 'gblinear', 'eta': 1, 'gamma': 25, 'max_depth': 1, 'n_estimators': 100	0.550
Máquina vector soporte	'C': 100, 'gamma': 1, 'kernel': 'rbf'	0.525

Tabla 3. Resumen de los resultados de los modelos de clasificación.

Como se puede observar en la tabla anterior, el rendimiento de los diferentes modelos con su mejor parametrización es bastante similar. De todos ellos, el modelo que ha obtenido la mejor tasa de acierto es el modelo basado en árboles de decisión con la parametrización mencionada en la tabla y rozando el 60% de precisión.

Nota: dado que hasta el momento se está trabajando con datos sintéticos, se procederá a futuro a generar el entrenamiento de nuevo con los datos reales que se proporcionen para obtener modelos con una mayor fiabilidad.

5 Sistema experto de prevención de caídas y evaluación de la eficacia de rehabilitación

Una vez construido un modelo que sea capaz de realizar una clasificación binaria de riesgo o no de caída para un paciente en función de una medición, se ha construido un sistema experto que permita servir de apoyo al personal médico.

Dicho sistema recibe como entrada uno o varios ficheros de entrada que serán mediciones del movimiento del paciente con estructura similar a la explicada a lo largo de este paquete de trabajo. A partir de estos ficheros el sistema utilizará el modelo construido anteriormente para calcular la probabilidad de caída del paciente que se corresponde con la probabilidad de que una medición pertenezca a la clase riesgo de caída que será complementaria a la probabilidad de la clase no riesgo de caída.

En función del valor de la probabilidad devuelta el sistema es capaz de proporcionar diferentes protocolos a seguir como sugerencia al personal médico. Un ejemplo de este tipo de sugerencias podría ser:

- > 25%: trabajo de la marcha en circuito o caminar con obstáculos para mejorar el equilibrio en bipedestación. Rehabilitación 2 días por semana.
- > 50%: potenciación de miembros inferiores con ejercicios de musculación en las piernas y columna lumbar, practicar subidas y bajadas de escaleras, marcha con ayudas técnicas (andador, dos/una muleta). Rehabilitación 3 días por semana.
- > 75%: ejercicios de potenciación de miembros inferiores en espaderas (agarrado/a a una barra para mantener el equilibrio y con silla detrás), practicar la marcha en barras paralelas y con andador siempre con supervisión y acompañado con la silla detrás. Rehabilitación 3 o 4 días por semana a discreción del personal médico.

Una vez proporcionado el protocolo experto a seguir, este sistema se modificará para dar una respuesta médica apropiada.

Además, si se le ha proporcionado más de un fichero, el sistema proporciona una indicación de si el proceso de rehabilitación ha reducido o no la probabilidad de caída del paciente, es decir, si la rehabilitación está funcionando o no. Esto se realiza analizando la evolución de las probabilidades obtenidas en cada una de las mediciones proporcionadas al sistema.

A continuación, se muestra un esquema a modo de resumen del funcionamiento del sistema experto de recomendación:

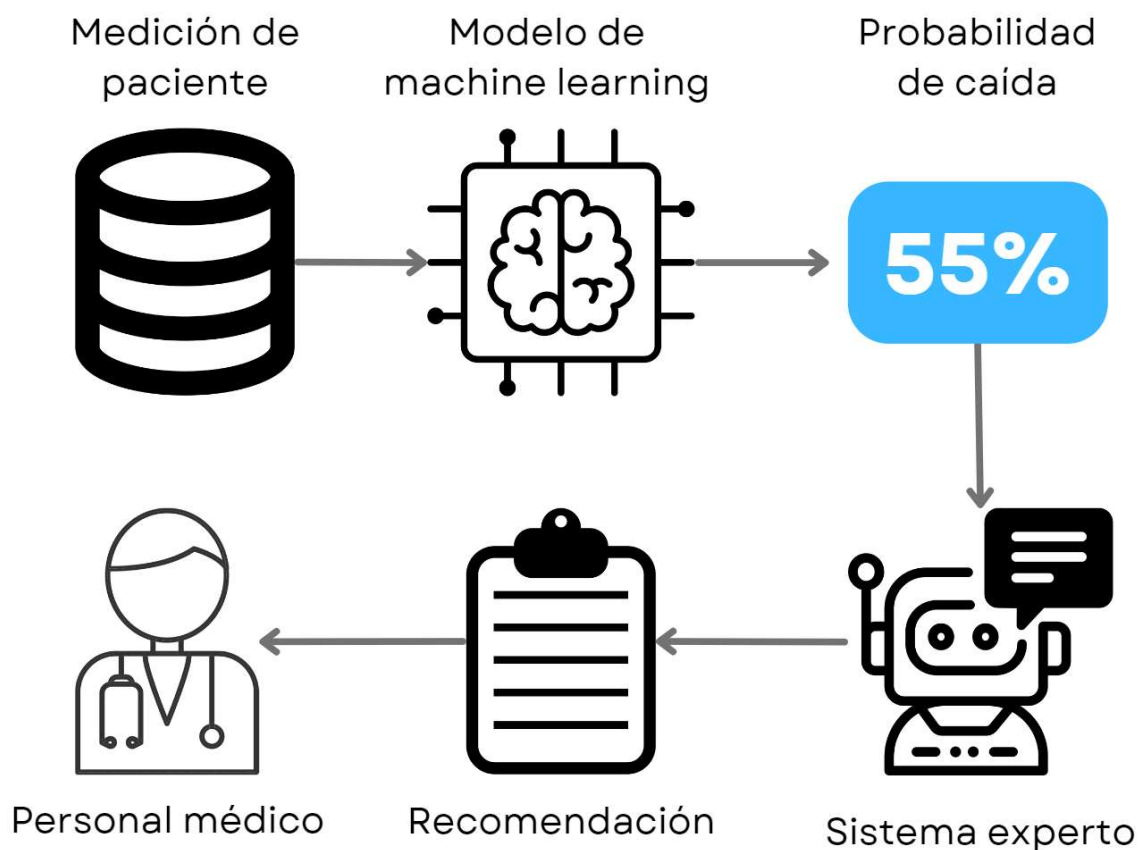


Figura 6. Ciclo de funcionamiento del sistema experto de recomendación.

6 Conclusiones

Este entregable corresponde al paquete de trabajo “PT3 – Sistema experto de prevención de caídas”, mostrando los resultados obtenidos a través del entrenamiento de múltiples modelos de aprendizaje automático supervisado.

Se ha abordado el modelado de aprendizaje automático supervisado de clasificación, donde la variable objetivo binaria identifique si el individuo al que se le ha realizado la medición tiene riesgo o no caída. Se han aplicado una gran variedad de parametrizaciones para 9 modelos diferentes.

De entre los diferentes modelos, el que mejor resultado ha dado es el modelo basado en árboles de decisión con una precisión del 59% aunque hay que recordar que este entrenamiento se ha realizado sobre datos sintéticos generados a partir de mediciones reales. Además, al no existir una variable objetivo de clasificación ya definida, se han generado valores aleatorios de esta. En un futuro se pretende aplicar la estructura de modelado ya implementada sobre datos reales ya etiquetados.

Finalmente se ha implementado un sistema experto de prevención de caídas y evaluación de la eficacia de rehabilitación que sirve como apoyo para personal médico que proporciona la probabilidad de caída del paciente, una descripción de del posible tratamiento a partir de esta y en caso de proporcionar múltiples ficheros, la evolución del paciente con la rehabilitación.