Instituto Politécnico Nacional
Escuela Superior de Cómputo

Ingeniería en Inteligencia Artificial
Grupo: 5BM1
Unidad de aprendizaje: Aprendizaje de Máquina
Práctica #7 Modelos de validación
Profesor: Abdiel Reyes Vera
Alumnos:
Rodríguez Juárez Héctor Sajoni
Velázquez Arrieta Eduardo Uriel
Fecha de entrega:
13 de Octubre del 2025

índice de contenido

Índice de imágenes

Introducción

Al momento de entrenar un modelo de Machine Learning tenemos dos tipos fundamentales de modelos, los supervisados y no supervisados; la principal diferencia entre estos es que los supervisados ya cuentan con datos etiquetados, mientras que los no supervisados se encargan de agrupar y etiquetar los datos tomando en cuenta las características con las que cuenta el conjunto de datos a trabajar.

En esta práctica nos centraremos en el tratamiento de la información y el cómo dividir de varias maneras el conjunto de datos (dataset) puede influir demasiado en la calidad de predicciones de los modelos.

En el ámbito del Machine Learning, la correcta evaluación del rendimiento de un modelo es tan crucial como su desarrollo. Esta práctica se enfoca en la exploración de diversos métodos de validación, herramientas esenciales para garantizar que nuestros modelos no solo aprendan de los datos de entrenamiento, sino que también sean capaces de generalizar eficazmente a datos no vistos.

A lo largo de esta práctica, utilizaremos un conjunto de herramientas de la biblioteca scikit-learn, incluyendo cross_validate, KFold, train_test_split, y LeaveOneOut, para implementar y comparar diferentes estrategias de validación. Trabajaremos con datasets clásicos como iris, breast_cancer y wine, y evaluaremos el desempeño de modelos como RandomForestRegressor y LinearRegression mediante métricas clave como el Error Cuadrático Medio (MSE) y el Coeficiente de Determinación (R^2).

El objetivo principal es comprender cómo la división y el tratamiento de los conjuntos de datos pueden influir significativamente en la calidad y fiabilidad de las predicciones de los modelos, así como identificar las ventajas y desventajas de cada método de validación.
En el ámbito del Machine Learning, la correcta evaluación del rendimiento de un modelo es tan crucial como su desarrollo. Esta práctica se enfoca en la exploración de diversos métodos de validación, herramientas esenciales para garantizar que nuestros modelos no solo aprendan de los datos de entrenamiento, sino que también sean capaces de generalizar eficazmente a datos no vistos.

A lo largo de esta práctica, utilizaremos un conjunto de herramientas de la biblioteca scikit-learn, incluyendo cross_validate, KFold, train_test_split, y LeaveOneOut, para implementar y comparar diferentes estrategias de validación. Trabajaremos con datasets clásicos como iris, breast_cancer y wine, y evaluaremos el desempeño de modelos como RandomForestRegressor y LinearRegression mediante métricas clave como el Error Cuadrático Medio (MSE) y el Coeficiente de Determinación (R^2).

El objetivo principal es comprender cómo la división y el tratamiento de los conjuntos de datos pueden influir significativamente en la calidad y fiabilidad de las predicciones de los modelos, así como identificar las ventajas y desventajas de cada método de validación.

Desarrollo

Como ya hemos trabajado en otras prácticas, el cómo particionar el dataset es algo que define mucho el rumbo de la práctica, trabajaremos con 3 métodos de validación; los cuales a su vez tienen varios métodos que parten de una idea principal y dependiendo del modelo, dataset y si queremos hacer una regresión o clasificación cada cada uno puede tener mejores o peores resultados.

Cross Validation

Con el fin de evaluar el rendimiento de un modelo de Machine Learning, hay que probarlo con nuevos datos. En función del rendimiento del modelo con datos desconocidos, se puede determinar si aún falta por ajustarlo, se ha ajustado de más o está "bien generalizado".

Una de las técnicas más empleadas para probar la eficacia de un modelo de Machine Learning es la "cross-validation" o validación cruzada. Este método también es un procedimiento de "re-sampling" (remuestreo) que permite evaluar un modelo incluso con datos limitados.

Para efectuar una "CV" (cross-validation), hace falta apartar de antemano una parte de los datos de la serie de datos de entrenamiento. Esos datos no se utilizarán para entrenar el modelo, sino más tarde para probarlo y validarlo.

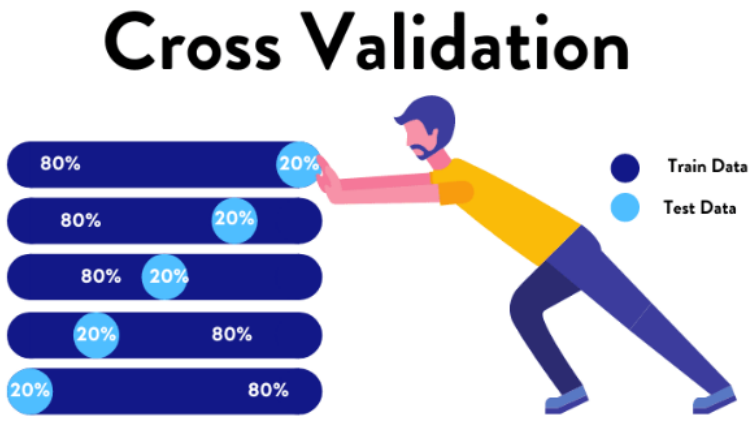

Imagen 1. Cross Validation

Los primeros métodos que implementamos fueron los de cross validation, para ser específicos KFolds y LeaveOneOut.

KFolds

En Kfolds partimos el dataset en varias n partes tal que ,n = 1,2,3,....,n $\forall n \, \varepsilon \, \Re$; aunque generalmente se suele partir entre un 10% a un 20%; estas divisiones son llamadas pliegues o folds.

En cada iteración se selecciona un fold diferente para entrenamiento, por lo que los parámetros que se producen pueden diferir un poco entre cada uno. Los parámetros obtenidos en cada modelo son promediados para obtener un modelo final.

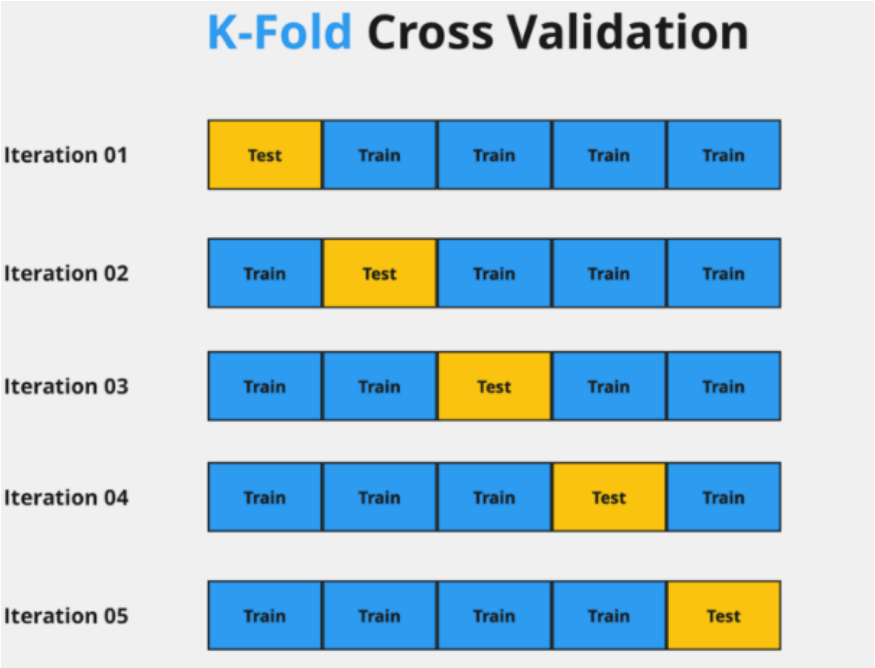

Imagen 2. KFolds

Leave-One-Out

Esta variante de la validación cruzada excluye un punto de datos de los datos de entrenamiento. Por ejemplo, si hay n puntos de datos en la muestra original, las piezas utilizadas para entrenar el modelo son n-1, y p puntos se utilizarán como conjunto de validación.

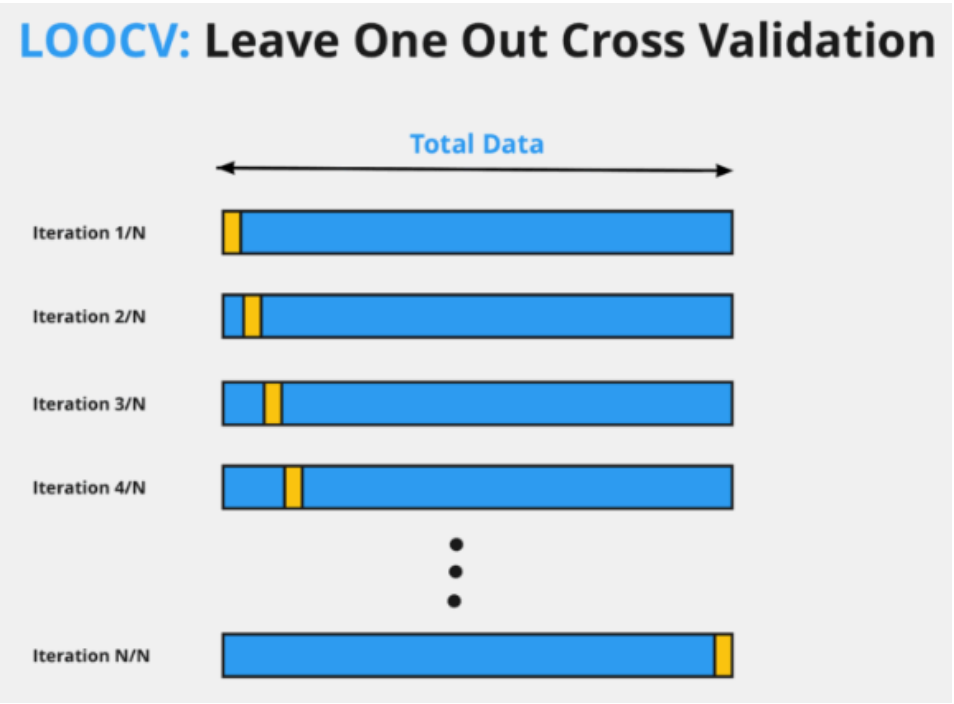

Imagen 3. LeaveOneOut

Forma Tradicional

La forma “tradicional” de entrenar un modelo de machine learning es simplemente segmentando el dataset en una proporción 70-30 hasta un 80-20, siendo la mayor parte para el entrenamiento y la restante para la validación y pruebas para saber qué tan acertadas son las predicciones.

Si bien esta manera suele tener resultados buenos, puede que por falta de información o por el contrario exceso de la misma se generan dos problemas Overfitting y Underfitting, como ya lo hemos mencionado en prácticas anteriores esto representa un gran problema, por lo que tratamos de evitarlo.

Para esta parte de la práctica se implementaron dos casos, uno donde se parte en 70-30 y otra donde se ocupa el 100 del dataset para entrenamiento y validación.

Resultados obtenidos

Para la realización de la práctica ocupamos los datasets de iris, breast_cancer y wine; esto por que ya los hemos trabajado en el pasado y conocemos bien los resultados esperados; para los modelos se usó RandomForestRegressor y LinearRegression, esto por la misma razón de los datasets, conocemos su funcionamiento y las predicciones esperadas.

Cross Validation KFolds

Para KFolds dividimos el dataset en 5 como indica la documentación de scikit-learn y nos dio los siguientes resultados.

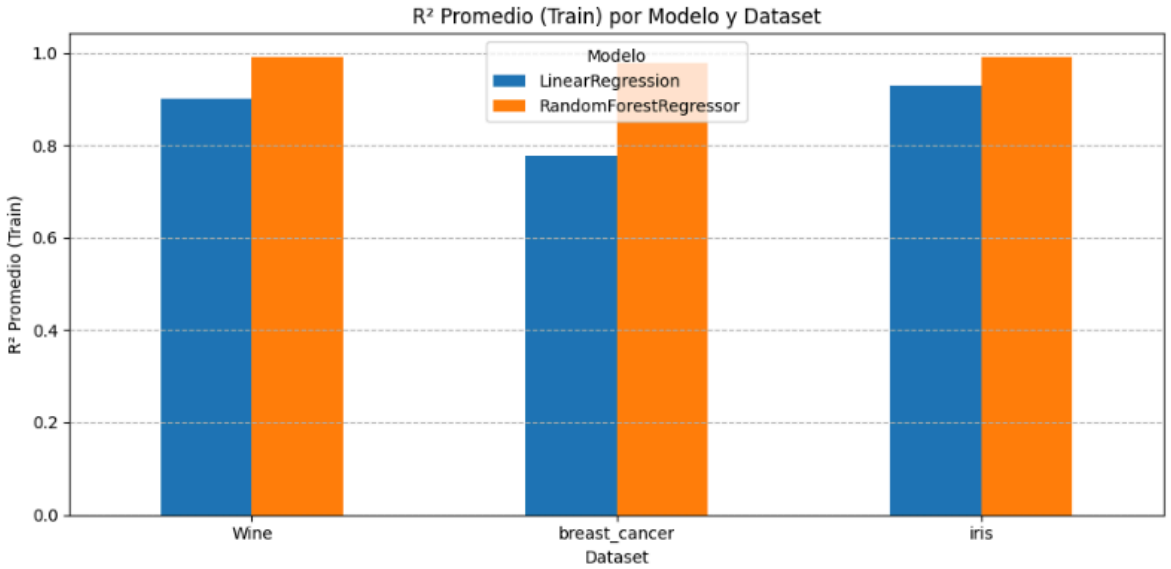

Imagen 4. Resultados KFolds

Cross validation Leave-One-Out

Para ejecutar cross validation se tomaron los mismos modelos y datasets que para KFolds, también nos percatamos de que este modo de separar el dataset es significativamente más pesado debido a todas las iteraciones que realiza.

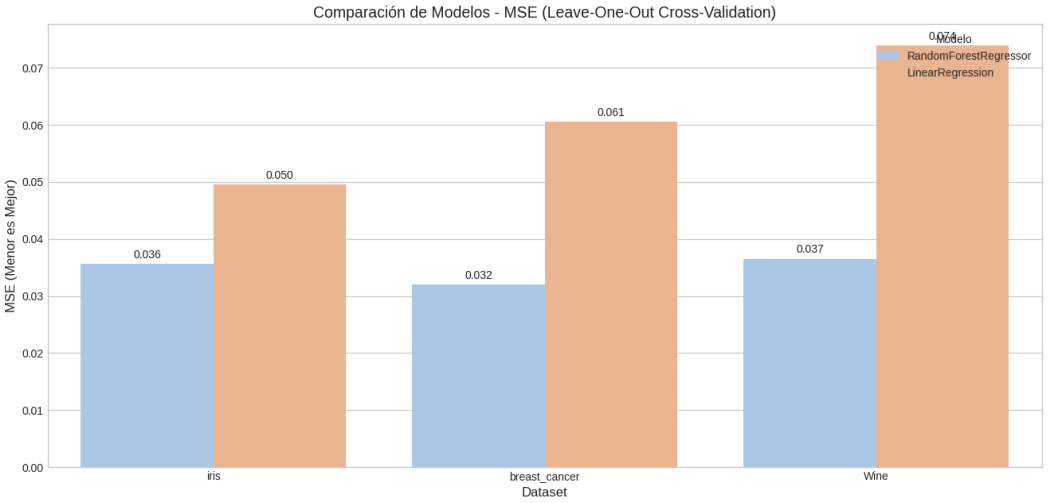

Imagen 5. Resultados Leave-One-Out

Modelo tradicional 70-30

Para este caso simplemente usamos un train_test_split para dividir en 70/30 los datos, donde obtuvimos métricas que ya habíamos visto en prácticas pasadas.

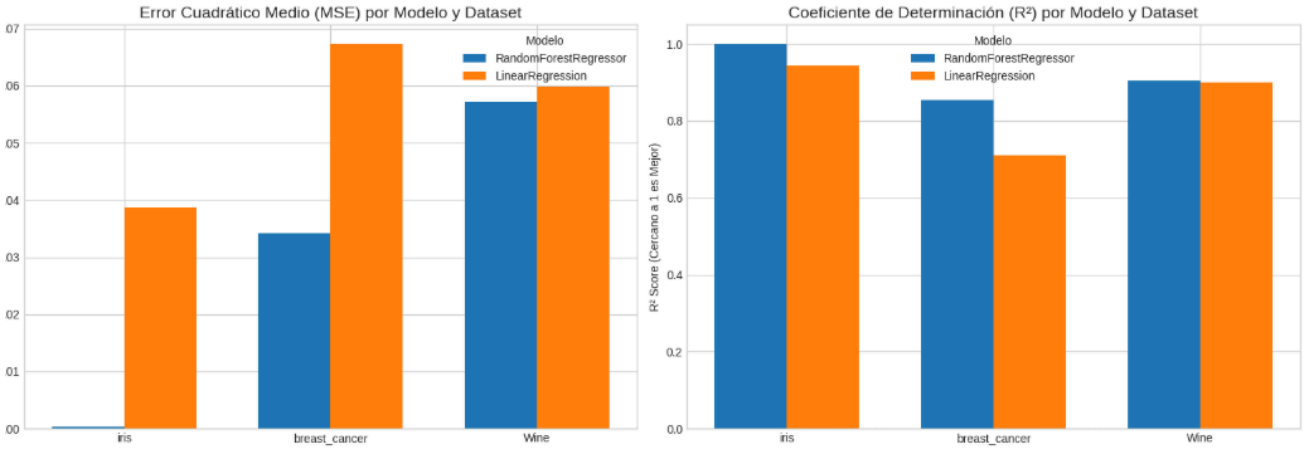

Imagen 6. Resultados de división 70-30

Modelo tradicional 100

Para el caso de 100 simplemente entrenamos y evaluamos el modelo con todo el dataset, el resultado esperado es tener un overfitting.

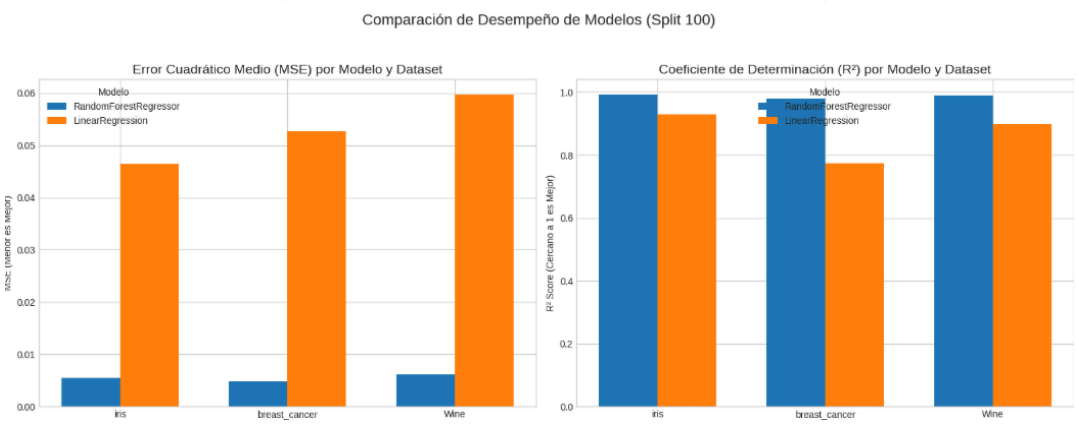

Imagen 7. Resultados de usar el 100 del dataset

Conclusiones

Las diversas estrategias de validación exploradas en esta práctica, como Cross Validation (KFolds y Leave-One-Out) y la división tradicional (70-30 y 100% del dataset), demuestran la importancia crítica de cómo se gestionan los conjuntos de datos en el aprendizaje automático. Cada método tiene sus propias fortalezas y debilidades, que se manifiestan en la fiabilidad y la capacidad de generalización de los modelos.

El Cross Validation, particularmente KFolds, se presenta como una técnica robusta para evaluar el rendimiento del modelo. Al dividir el dataset en múltiples pliegues y rotar los conjuntos de entrenamiento y validación, se reduce el riesgo de que el rendimiento del modelo dependa de una partición específica de los datos. Nuestros resultados con KFolds, utilizando datasets como iris, breast_cancer y wine con modelos RandomForestRegressor y LinearRegression, proporcionaron una visión más equilibrada de la capacidad predictiva. Esta metodología ayuda a mitigar el sobreajuste (overfitting) y el subajuste (underfitting), ofreciendo una estimación más precisa de cómo se comportará el modelo con datos no vistos. La documentación de scikit-learn sugiere que dividir el dataset en 5 pliegues es un buen punto de partida, lo cual se reflejó en los resultados obtenidos.

Por otro lado, Leave-One-Out, aunque también es una forma de validación cruzada, mostró ser computacionalmente más intensiva debido al gran número de iteraciones que realiza. Si bien puede ofrecer una estimación de sesgo muy baja, su alto costo computacional la hace menos práctica para datasets grandes, a pesar de usar los mismos modelos y datasets. La elección entre KFolds y Leave-One-Out, por lo tanto, no solo se basa en la precisión, sino también en los recursos computacionales disponibles y el tamaño del dataset.

En contraste, la "forma tradicional" de dividir el dataset, ya sea en proporciones 70-30 u 80-20, es más sencilla de implementar. Para el caso de 70-30, obtuvimos métricas que confirmaron un rendimiento razonable, similar a lo que habíamos observado en prácticas anteriores. Sin embargo, este método puede ser más susceptible a la variabilidad si la división no es representativa del conjunto de datos completo, lo que podría llevar a estimaciones sesgadas del rendimiento del modelo.

En resumen, la práctica ha reforzado la idea de que la selección adecuada del método de validación es tan crucial como la elección del algoritmo de aprendizaje automático y el preprocesamiento de los datos. Los modelos de validación no solo nos permiten estimar el rendimiento de un modelo, sino que también nos ayudan a detectar problemas como el overfitting y el underfitting. La biblioteca scikit-learn ofrece herramientas esenciales para implementar estas estrategias, permitiendo a los ingenieros de IA construir modelos más robustos y confiables. La comprensión de las ventajas y desventajas de cada método, junto con la experiencia práctica en su aplicación, es indispensable para desarrollar modelos de Machine Learning que sean verdaderamente efectivos en escenarios del mundo real.

Referencias

3.1. Cross-validation: evaluating estimator performance. (s. f.). Scikit-learn

https://scikit-learn.org/stable/modules/cross_validation.html

Arif, A. (2023, 19 octubre). How Cross-Validation works in machine Learning.

Dataaspirant. https://dataaspirant.com/cross-validation/