Instituto Tecnológico y de Estudios Superiores de Monterrey

Campus Estado de México

Uso de framework o biblioteca de aprendizaje máquina para la implementación

Marisol S. Ramírez Herrera A01747396

Inteligencia artificial avanzada para la ciencia de datos I

Dr. Jorge Adolfo Ramírez Uresti

10 de septiembre de 2025

1. Introducción

El aprendizaje supervisado es una de las ramas de *machine learning* (ML) que utiliza conjuntos de datos para entrenar modelos predictivos. A través de etiquetar entradas y salidas, el modelo aprende las relaciones subyacentes entre ellas y aplica los resultados a nuevos datos del mundo real (Belcic & Striker, 2024). Este enfoque se emplea ampliamente en tareas de clasificación y regresión, donde la capacidad de generalizar correctamente determina el valor práctico del modelo.

Dentro de la clasificación supervisada, el algoritmo de *random forest* ocupa un lugar destacado. Se trata de un método de ensamble que combina múltiples árboles de decisión, cada uno entrenado sobre subconjuntos de muestreo bootstrap y evaludado sobre características aleatorias (Shafi, 2024). La diversidad generada por este muestro mejora la estabilidad del modelo y reduce la varianza con respecto a un árbol único. Esta propiedad convierte al *random forest* en un modelo adecuado frente al sobreajuste (*overfitting*), adecuado para trabajar con conjuntos de datos de tamaño reducido.

El presente trabajo implementa un modelo *random forest* para la calisficación del conjunto de datos *Wine*. El objetivo es entrenar y evaluar el clasificador mediante validacion cruzada y búsqueda de hiperparámetros. Asimismo, se incorporan métricas de desemeña para comprender la precisión e interpetabilidad del algoritmo.

2. Metodología

La implementación del algoritmo se realizó con la biblioteca *scikit-learn*. Con el objetivo de entrenar la red, se utilizó el conjunto de datos *Wine* incluido en la biblioteca. Éste se compone de 178 registros, distribuidos en tres clases. La naturaleza multiclase del conjunto de datos representa un escenario idóneo para evaluar la capacidad de generalización del modelo, puesto que exige distinguir patrones complejos en presencia de múltiples categorías

Los datos fueron divididos en dos subconjuntos idenpendientes:

- ***Entrenamiento (60%)***: utilizado para el ajuste de hiperparámetros de la red.
- ***Prueba (40%)***: reservado para la evaluación final del modelo.

Se utilizó la partición estratificada para asegurar que la proporción de clases en el conjunto de entrenamiento y en el de prueba se mantuviese constante. Correspondientemente, se fijó una semilla aleatoria para garantizar la reproducibilidad de los resultados.

En la etapa de optimización, se implementó la técnica de *grid search* con validación cruzada en cinco particiones. Este método permitió evaluar combinaciones de hiperparámetros, incluyendo: profundidad máxima de los árboles, número mínimo de muestras en las hojas y la fracción de características consideradas en cada nodo. La validación cruzada estratificada se adoptó para asegurar que cada *fold* mantuviera la proporción real de clases. Esto con objeto de reducir la varianza y obtener un estimador más estable del desempeño.

El modelo seleccionado se entrenó sobre el conjunto de entrenamiento utilizando los mejores hiperparámetros identificados. Para la evaluación se calcularon las métricas de *accuracy*, *precision*, *recall* y *f1-score*. Estas métricas se complementaron con la matriz de confusión. Correspondientemente, se generaron curvas de aprendizaje para monitorear el overfitting y curvas de aprendizaje. Finalmente, los gráficos de características permiten interpretar el aporte relativo de cada variable en las decisiones del clasificador.

3. Resultados

3.1. Selección de hiperparámetros

Con el objetivo de encontrar la configuración óptima del modelo se utilizó la técnica de *grid search.* En consecuencia, se limitó la profundidad de los árboles a 3 niveles, mínimo de 2 muestras en cada hoja, 4 observaciones para realizar una división y una fracción reducida de características en cada partición (Figura 1). Esta combinación limita la complejidad de los árboles y evita que las reglas de decisión se ajusten de manera excesiva al ruido de los datos.

Figura 1. Parámetros óptimos obtenidos mediante GridSearchCV.

Parámetro	Valor
Profundidad máxima	3
Muestras mínimas en hoja	2
Muestras mínimas por split	4
Fracción de atributos	raíz cuadrada

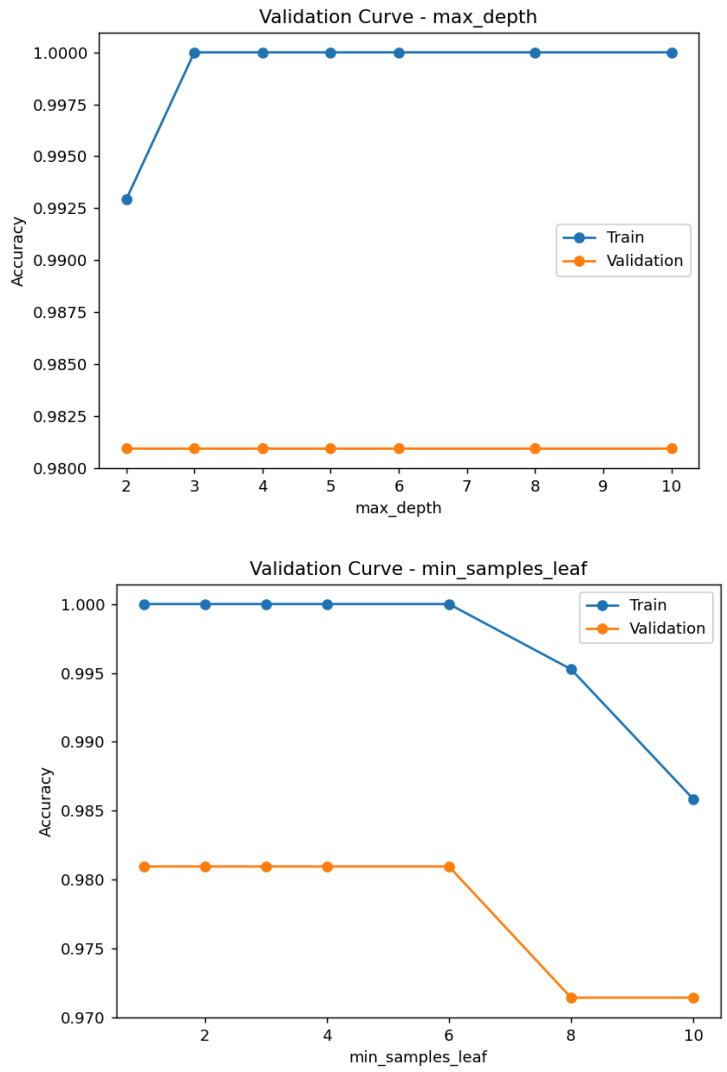
El modelo utilizó el índice de Gini como criterio de impureza para evaluar las divisiones en cada árbol. Este criterio mide la heterogeneidad de las clases en los nodos y favorece divisiones que aumentan la pureza de las ramas.

Con esta configuración, el modelo alcanzó un *accuracy* promedio de validación cruzada de 0.9810 ± 0.0233.

3.2. Curvas de validación

Las curvas de validación de los parámetros de profundidad y muestras mínimas en hojas confirmarn que el modelo mantiene un rendimiento estable bajo las configuraciones seleccionadas (Figura 2). En el caso de profundidad, la validación alcanzó su punto máximo a partir de 3, manteniéndose constante al incrementar la complejidad de los árboles. Por otro lado, el parámetro de muestras mostró que valores bajos (2–4) sostienen la exactitud en validación.

Figura 2. Curvas de validación para profundidad máxima y muestras mínimas en cada hoja.



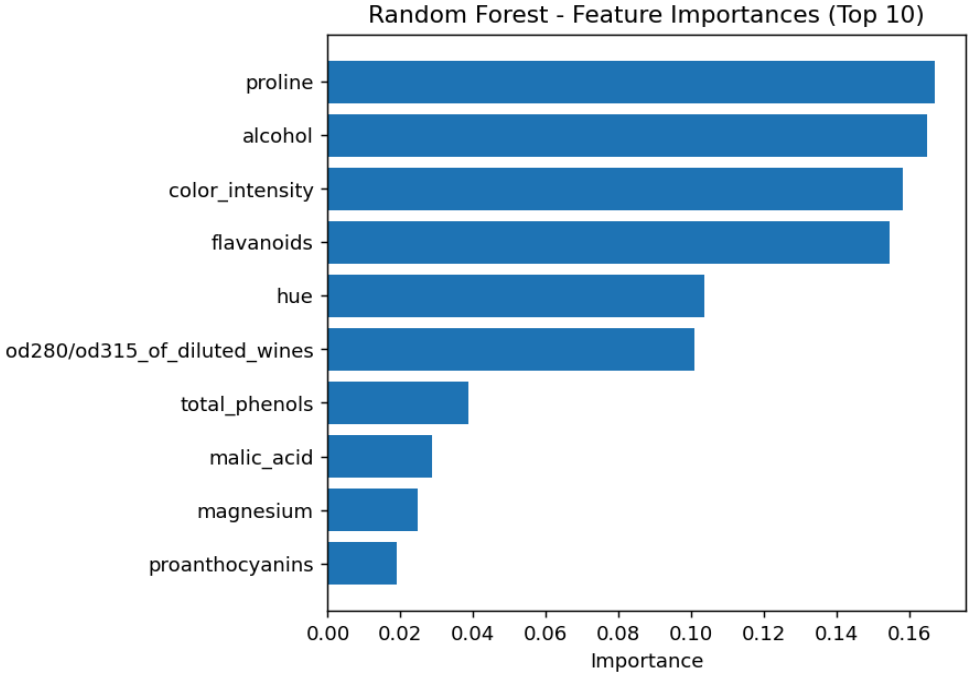
Estos hallazgos justifican la elección de hiperparámetros obtenidos en el punto 3.1.

3.3. Importancia de características.

El análisis de importancia de características permitió identificar las variables más influyentes en la clasificación de los vinos (Figura 3). El modelo seleccionó como atributos principales a *proline*, *alcohol* y *color_intensit.* Le siguen *flavanoids* y *hue*. Estas variables contribuyen en mayor medida a la construcción de reglas de decisión.

El resto de características, como *magnesium* y *proanthocyanins*, presentaron una importancia relativa menor. Esta visualización aporta interpretabilidad al modelo, puesto que ilutsra los atributos con mayor peso durante el proceso de clasificación.

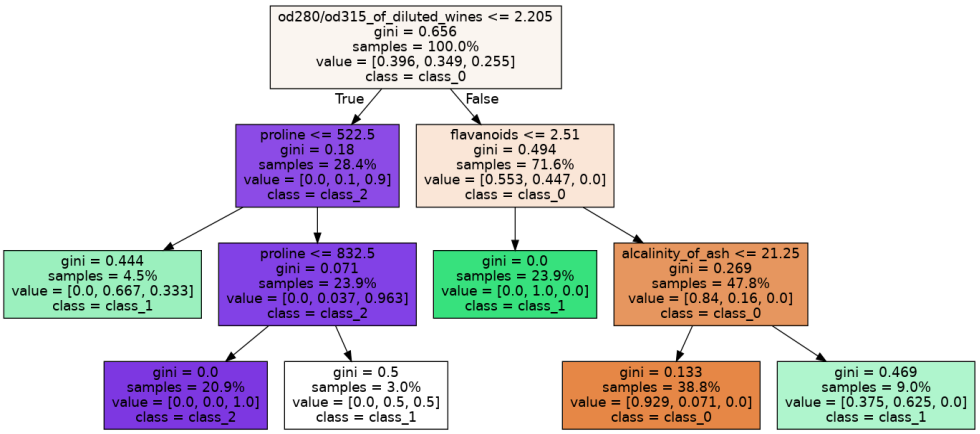
Figura 3. Importancia de características en el modelo aplicado al dataset de vinos.



3.4. Visualización de un árbol individual.

La visualización de uno de los árboles del bosque (Figura 4) mostró reglas de decisión basadas en atributos como: *od280/od315 (alcohol)*, *proline* y *flavanoids*. Estos nodos iniciales reflejan las variables más discriminatorias, confirmando la consistencia con el análisis de importancias.

Figura 4. Árbol de decisión individual.



Aunque un árbol individual no representa la totalidad del bosque, ilustra las divisiones internas.

3.5. Reporte de clasificación y matriz de confusión.

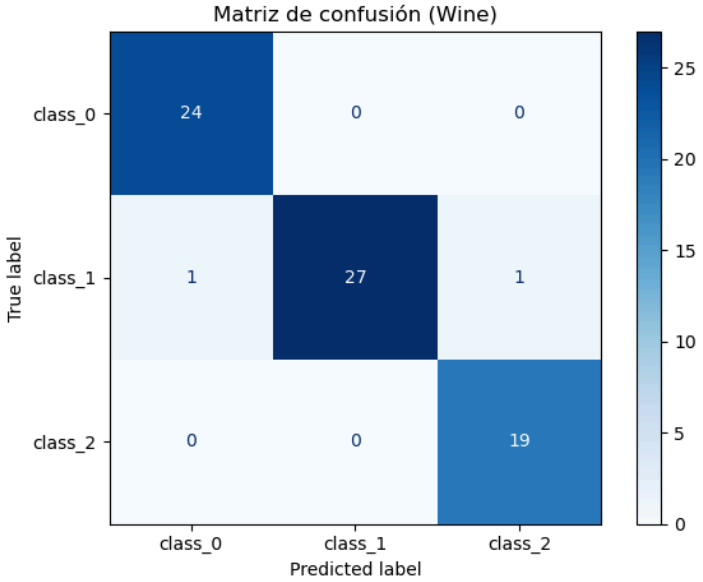
En el conjunto de prueba, el modelo logró un *accuracy* de 0.9722. El resultado es consistente con el valor estimado mediante validación cruzada. El reporte de clasificación evidenció métricas equilibradas: la clase 0 obtuvo un recall perfecto (1.00); la clase 1 presentó un ligero descenso en recall (0.93) asociado a dos errores de clasificación; y la clase 2 alcanzó nuevamente un recall de 1.00. El promedio macro de F1-score fue de 0.97, lo que confirma un desempeño homogéneo entre clases (Figura 5).

Figura 5. Reporte de clasificación con el dataset de prueba.

```
=== Test ===
Accuracy (test): 0.9722

=======Classification Report======
              precision    recall  f1-score   support

     class_0       0.96      1.00      0.98        24
     class_1       1.00      0.93      0.96        29
     class_2       0.95      1.00      0.97        19

    accuracy                           0.97        72
   macro avg       0.97      0.98      0.97        72
weighted avg       0.97      0.97      0.97        72
```

Por su parte, la matriz de confusión ilustra dichos resultados (Figura 6). Como se observa, el modelo clasificó correctamente 70 de 72 observaciones. Este desempeño respalda la capacidad del random forest para distinguir entre vinos con un nivel mínimo de error.

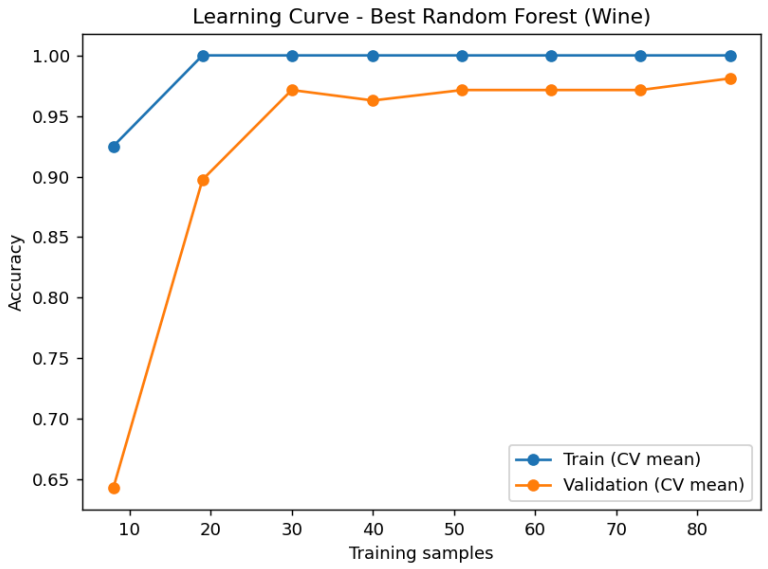
Figura 6. Matriz de confusión.



3.6. Curva de aprendizaje.

La curva de aprendizaje muestra que el *accuracy* en entrenamiento se mantuvo cercano a 1.0. La validación, por su parte, alcanzó valores estables entre 0.96 y 0.98 (Figura 7). Esta diferencia indica un leve *overfitting*, pero dentro de un rango aceptable dado que la validación se mantuvo en un nivel consistente.

Figura 7. Curva de aprendizaje del modelo implementado.



El comportamiento observado demuestra que aumentar el tamaño del conjunto de entrenamiento mejora progresivamente la estabilidad del modelo.. Estos resultados reflejan que el modelo generaliza adecuadamente incluso en presencia de una tendencia natural al ajuste excesivo. Además, confirman la efectividad de las restricciones aplicadas en los hiperparámetros para controlar la complejidad del bosque.

4. Conclusiones.

En conclusión, el algoritmo *random forest* es una herramienta efectiva para abordar problemas de clasificación multiclase. La implementación alcanzó un desempeño adecuado, con métricas de validación cruzada y prueba superiores al 97%. La selección de hiperparámetros mediante *grid search* permite optimizar el modelo. Correspondientemente, la matriz de confusión mostró un número mínimo de errore. Este comportamiento refleja un aspecto esperado en problemas de clasificación real, donde los límites entre categorías pueden ser difusos.

No obstante, el análisis también identificó limitaciones. El leve *overfitting* observado sugiere que la estabilidad del modelo podría reforzarse con un mayor número de datos o con estrategias de regularización adicionales. Asimismo, la confusión parcial en la clase 1 indica que sería beneficioso explorar técnicas de preprocesamiento para mejorar la discriminación entre categorías cercanas

Referencias.

Belcic, I & Stryker, C. (2024). ¿Qué es el aprendizaje supervisado?. *IBM*. Recuperados de https://www.ibm.com/mx-es/think/topics/supervised-learning

GridSearchCV. (s. f.). Scikit-learn. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html#sklearn.model_selection.GridSearchCV.

Shafi, A. (2024). Clasificación de bosques aleatorios con Scikit-Learn. *Datacamp*. Recuperado de https://www.datacamp.com/es/tutorial/random-forests-classifier-python

Uresti, J. (2025). *Tema 7. Aprendizaje supervisado.* Inteligencia artificial para la ciencia de datos I. ITESM CEM.