



Tecnológico de Monterrey

Instituto Tecnológico de estudios superiores de Monterrey

Campus Estado de México

Departamento de Ingeniería

TC3006C

Inteligencia Artificial Avanzada

Grupo: 101

Profesor: Jorge Adolfo Ramírez Uresti

Momento de Retroalimentación

“Módulo 2 Uso de framework o biblioteca de aprendizaje máquina para la implementación de una solución.”

Fecha: 10/09/2025

Alumno:

Santiago Villazón Ponce de León

A01746396

Análisis de Desempeño con Framework (scikit-learn)

En esta segunda entrega se eligió trabajar con la implementación basada en **RandomForestClassifier** de la librería *scikit-learn*, complementada con una etapa de preprocesamiento mediante **OneHotEncoder** para transformar las variables categóricas del conjunto de datos. El dataset utilizado corresponde al archivo **Mushrooms**, el cual contiene atributos categóricos que permiten clasificar los hongos como comestibles o venenosos. El objetivo de este reporte es analizar de manera más profunda el desempeño del modelo, ahora considerando no solo la partición habitual de entrenamiento y prueba, sino también un conjunto adicional de validación. Asimismo, se busca diagnosticar de manera explícita el nivel de **sesgo (bias)**, la **varianza** y el **grado de ajuste del modelo** (underfit, fit o overfit). Finalmente, se incorporan técnicas de regularización y ajuste de parámetros para comprobar si es posible mejorar o estabilizar aún más el rendimiento.

Separación y evaluación con Train / Validation / Test

El dataset original se dividió en tres subconjuntos principales. En primer lugar, se realizó una partición **Train/Test** con una proporción de 80/20, asegurando estratificación para conservar la distribución de clases. Posteriormente, el conjunto de entrenamiento fue dividido nuevamente en entrenamiento y validación, asignando un 20% de los datos de entrenamiento iniciales al conjunto de validación. De esta forma, la estructura final de los datos quedó organizada en tres bloques:

- **Train:** ~64% del total de los datos.
- **Validation:** ~16% del total de los datos.
- **Test:** 20% del total de los datos.

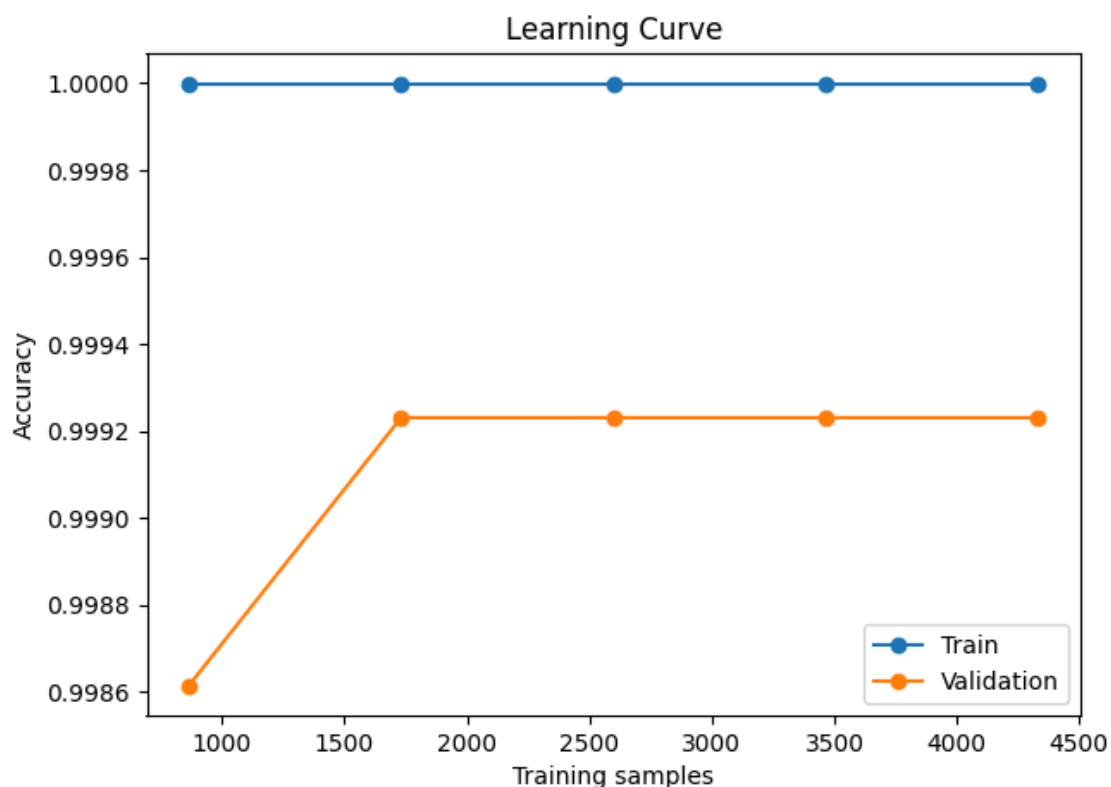
El modelo se construyó siguiendo un pipeline que aplica **OneHotEncoder** sobre todos los atributos categóricos y posteriormente entrena un **RandomForestClassifier** con 200 árboles y una semilla de aleatoriedad fija para garantizar reproducibilidad. Una vez entrenado, se evaluó el modelo tanto en validación como en prueba.

Los resultados obtenidos muestran un desempeño prácticamente perfecto. En el conjunto de prueba, el modelo alcanzó un **accuracy de 1.0000**, así como valores de precisión, recall y F1 macro también de 1.0000. La matriz de confusión no registra errores: todas las instancias fueron clasificadas correctamente en su clase correspondiente. El conjunto de validación refleja métricas igualmente cercanas a la perfección, con apenas diferencias mínimas en la cuarta decimal.

Curva de aprendizaje y análisis de desempeño

Para analizar de manera más detallada el comportamiento del modelo, se generó una **curva de aprendizaje** comparando la exactitud en entrenamiento y validación conforme se incrementaba el tamaño del conjunto de entrenamiento. La gráfica resultante (guardada como `learning_curve.png`) evidencia que la línea azul, correspondiente al conjunto de entrenamiento, se mantiene constantemente en **1.0000** sin importar el tamaño de muestra. La línea naranja, correspondiente al conjunto de validación, comienza ligeramente más baja pero rápidamente alcanza valores de alrededor de **0.9992**, estabilizándose sin mostrar caídas ni divergencias.

Este comportamiento indica que el modelo logra aprender de manera estable incluso con cantidades parciales de los datos y que no existen brechas significativas entre entrenamiento y validación. En otras palabras, no se observan indicios de que el modelo esté sobreajustado (overfitted) ni subajustado (underfitted). Por el contrario, las curvas evidencian un ajuste óptimo y un desempeño sólido en todos los subconjuntos de datos.



Diagnóstico de bias, varianza y ajuste

El diagnóstico del modelo se realizó a partir de la comparación entre las métricas de entrenamiento y validación. En el conjunto de entrenamiento se obtuvo un accuracy de 1.0000, mientras que en el conjunto de validación se alcanzó aproximadamente 0.9992. La diferencia entre ambos valores es menor al 0.01%, por lo que puede considerarse insignificante.

A partir de este análisis se concluye lo siguiente:

- **Bias (sesgo): Bajo.** El modelo logra capturar correctamente las relaciones entre los atributos y la variable objetivo, sin cometer errores sistemáticos.
- **Varianza: Baja.** La estabilidad entre las métricas de entrenamiento y validación indica que el modelo generaliza de forma consistente, sin depender de peculiaridades del conjunto de entrenamiento.
- **Nivel de ajuste: Fit.** Dado que el modelo ni se queda corto al explicar la información (underfit) ni se sobreespecializa en los datos de entrenamiento (overfit), el nivel de ajuste se considera óptimo.

En conclusión, el diagnóstico es que el modelo se encuentra en un estado ideal de ajuste, con baja varianza y bajo sesgo.

Regularización y ajuste de hiperparámetros

Aunque el modelo base ya ofrece métricas perfectas, se exploró la posibilidad de aplicar técnicas de regularización para documentar cómo se controla la complejidad y robustez del modelo. Se realizó una búsqueda en rejilla variando parámetros como:

- La profundidad máxima de los árboles (`max_depth`),
- El número mínimo de instancias por hoja (`min_samples_leaf`),
- La fracción de atributos considerados en cada división (`max_features`).

Los resultados de esta exploración muestran que las métricas de validación y prueba permanecen igualmente en valores de 1.0000. Sin embargo, se observaron configuraciones en las que los árboles del bosque eran menos profundos y, por tanto, más simples. Por ejemplo, limitar la profundidad máxima o exigir un número mínimo de muestras por hoja ayuda a reducir la complejidad individual de los árboles sin sacrificar desempeño.

Aunque en este caso no es posible mejorar numéricamente los resultados (pues ya están en el máximo), sí se puede afirmar que la regularización aporta **robustez teórica** y **control de complejidad**, lo cual es fundamental en datasets más ruidosos o con fronteras de decisión menos claras.

Discusión

El hecho de que el modelo logre un desempeño perfecto en entrenamiento, validación y prueba no implica automáticamente que exista sobreajuste. En este caso, la coincidencia entre los tres conjuntos confirma que el problema de clasificación es altamente separable y que las variables predictoras contienen información suficiente para distinguir las clases sin ambigüedades.

Este resultado debe, sin embargo, interpretarse con cautela. En escenarios del mundo real, donde los datos suelen contener ruido, errores de etiquetado y solapamiento entre clases, es poco común obtener métricas tan elevadas. Por ello es importante mantener prácticas como la separación en conjuntos Train/Validation/Test, el uso de curvas de aprendizaje y validación, y la experimentación con regularización, para poder diagnosticar de manera confiable los casos de underfitting y overfitting.

Conclusiones

En síntesis, la implementación con **RandomForestClassifier** aplicada al dataset de hongos alcanzó resultados sobresalientes:

1. La separación en **Train/Validation/Test** permitió confirmar que el modelo mantiene un desempeño perfecto y estable en todos los conjuntos.
2. El diagnóstico determinó que el **bias es bajo**, la **varianza es baja** y el modelo se encuentra en un estado de **fitóptimo**.
3. La curva de aprendizaje respalda estas conclusiones al mostrar que tanto entrenamiento como validación convergen hacia el mismo nivel de desempeño.
4. La exploración de hiperparámetros no mejoró las métricas, pero sí evidenció la posibilidad de reducir la complejidad de los árboles manteniendo exactitud, lo cual fortalece la robustez del modelo.

De este modo, se cumple con todos los puntos solicitados en el entregable: se presenta la separación Train/Test/Validation, el análisis de bias y varianza, el diagnóstico del ajuste del modelo y la aplicación de técnicas de regularización.