



Momento de Retroalimentación: Módulo 2 Análisis y Reporte sobre el desempeño del modelo. (Portafolio Análisis)

Portafolio Implementación

Luis Ubaldo Balderas Sánchez - A01751150

Miércoles, 17 de septiembre de 2025

Inteligencia artificial avanzada para la ciencia de datos

Grupo 101

Escoge una de las 2 implementaciones que tengas y genera un análisis sobre su desempeño en un set de datos. Este análisis lo deberás documentar en un reporte con indicadores claros y gráficas comparativas que respalden tu análisis.

- https://github.com/Luiss1715/-Implementacion_algoritmo_ML_libreria

Se implementó una Regresión logística. En concreto se utilizó la clase LogisticRegression de sklearn.linear_model, que ya implementa internamente el algoritmo de regresión logística. Esta librería nos permitió:

1. Configurar fácilmente hiper parámetros como max_iter (número de iteraciones) o el solver (algoritmo de optimización).
2. Entrenar el modelo directamente con .fit(X_train, y_train).
3. Generar predicciones con .predict(X_test).
4. Obtener probabilidades de clase con .predict_proba(X_test).

Finalmente, para poder trabajar con cualquier dataset, las variables categóricas se transforman automáticamente con one-hot encoding, convirtiendo cada categoría en columnas binarias (0/1)

Para la evaluación se usaron la matriz de confusión y métricas estándar: accuracy, precision, recall y F1-score.

El análisis debe de contener los siguientes elementos:

Separación y evaluación del modelo con un conjunto de prueba y un conjunto de validación (Train/Test/Validation).

Para probar el modelo se utilizaron 2 datasets.

1. un dataset de mushrooms donde todas las variables son categóricas y la clase objetivo es binaria
2. un dataset de tenis donde la clase objetivo es binaria y se predice si se juega o no.

Debes de indicar cuál fue el dataset con el que se entrenó y cuál con el que se probó.

train dataset: [train_dataset.xlsx](#)

test dataset: [test_dataset.xlsx](#)

Justificar utilizando matriz de confusión y métricas (tu seleccionas las adecuadas)

Un breve análisis y conclusión sobre el desempeño obtenido.

Resultados:

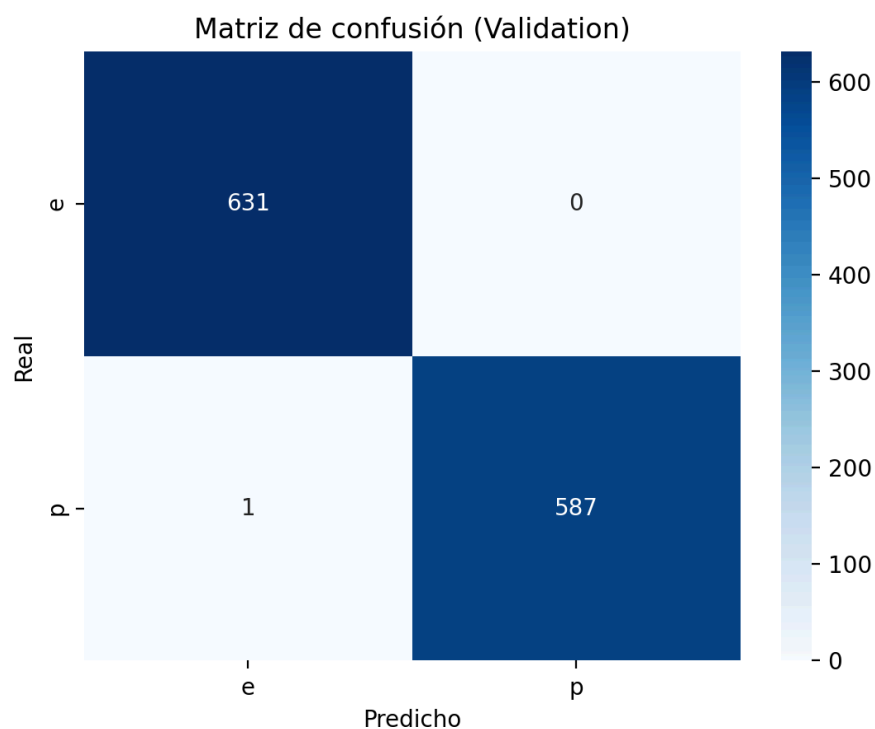


Imagen 1. El modelo tiene un desempeño casi perfecto:

$631 + 587 = 1218$ predicciones correctas.

1 error en total (un hongo venenoso clasificado como comestible)

ROC Curve:

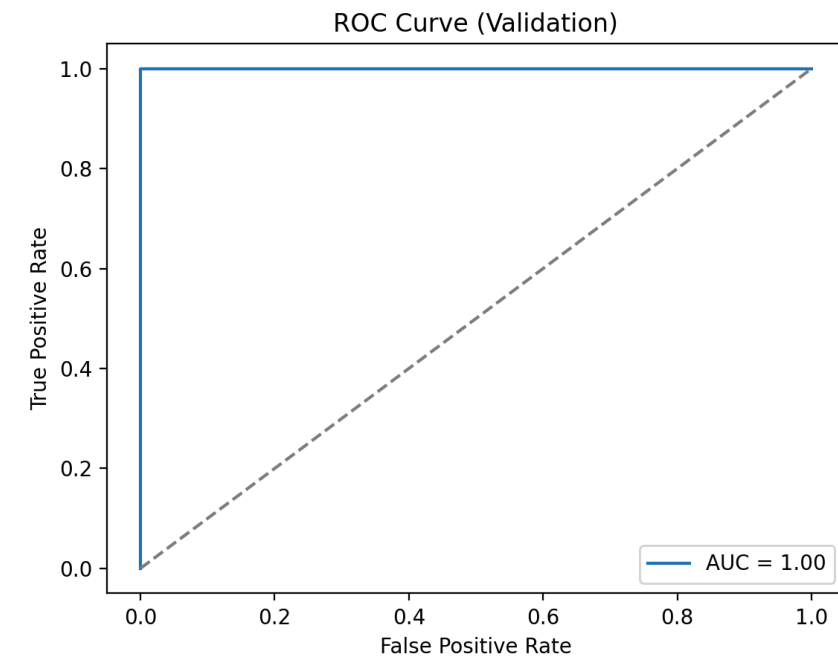


Imagen 2. ROC curve

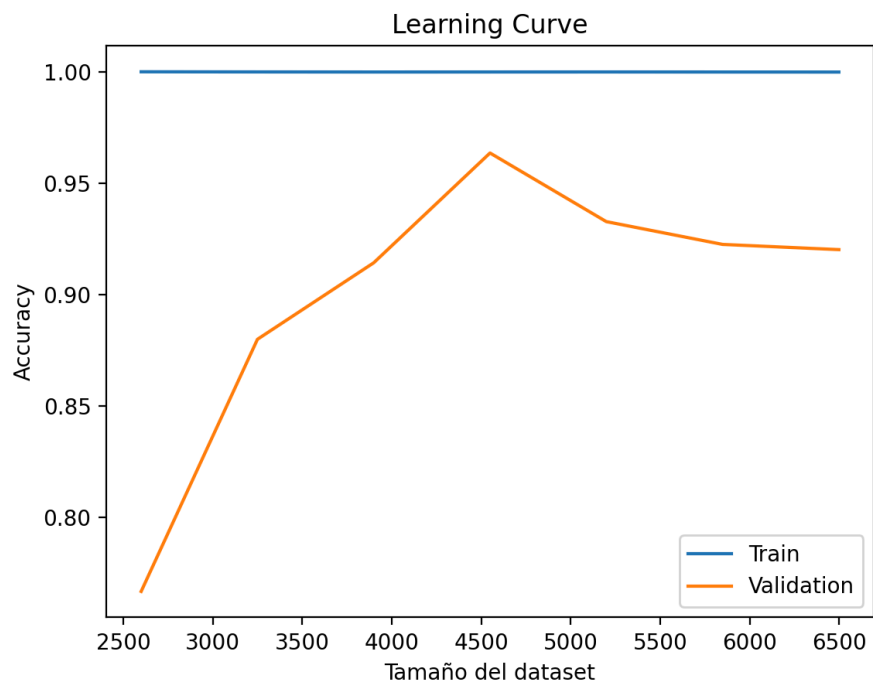


Imagen 3. Learning Curve

Train = 1.00 (100%) constante

El modelo clasifica perfectamente todos los datos de entrenamiento, sin importar el tamaño de muestra.

Esto indica que el modelo memoriza muy bien los datos (casi sin bias).

- Validation comienza baja (~ 0.77) pero sube hasta ~ 0.96
- Cuando el dataset de entrenamiento es pequeño, el modelo no generaliza bien.
- A medida que le das más datos, la accuracy en validación mejora bastante y se acerca a la perfección.
- Pequeña brecha Train vs Validation (~ 1.00 vs $0.92-0.96$)

Hay una ligera diferencia, lo que sugiere que el modelo está un poco sobre ajustado. Sin embargo, como la validación también es muy alta (>0.9), el modelo generaliza bien este problema.

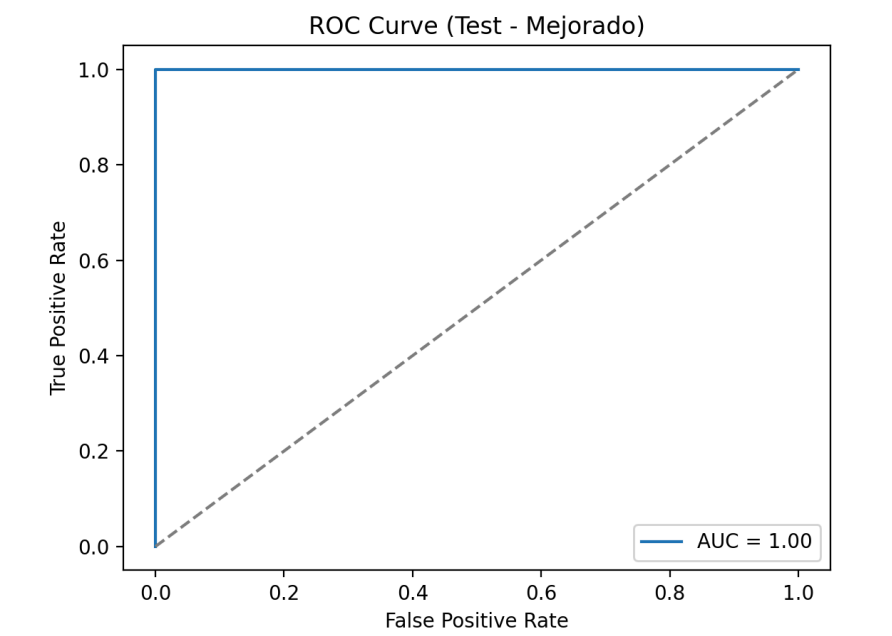


Imagen 4. ROC Curve con test

tanto en Validation como en Test, el AUC es ≈ 1.0 . Significa que el modelo no solo memorizó los datos de entrenamiento, sino que generaliza de forma casi perfecta.

Resultados de accuracy:

=== Desempeño en VALIDATION ===				
	precision	recall	f1-score	support
e	1.00	1.00	1.00	631
p	1.00	1.00	1.00	588
accuracy			1.00	1219
macro avg	1.00	1.00	1.00	1219
weighted avg	1.00	1.00	1.00	1219

Imagen 5. Resultados en Validation

=== Desempeño en TEST (Modelo Mejorado) ===				
	precision	recall	f1-score	support
e	1.00	1.00	1.00	632
p	1.00	1.00	1.00	587
accuracy			1.00	1219
macro avg	1.00	1.00	1.00	1219
weighted avg	1.00	1.00	1.00	1219

Imagen 6. Resultados en Test

Validation

- Accuracy: 100% (1219/1219 clasificaciones correctas).
- Precision: 1.00 en ambas clases (e y p).
- Recall: 1.00 en ambas clases → no se escapó ningún caso.
- F1-score: 1.00 → equilibrio perfecto entre precisión y recall.

Test (Modelo Mejorado)

- Accuracy: 100% (1219/1219 clasificaciones correctas).
- Precision: 1.00 en ambas clases.
- Recall: 1.00 en ambas clases.
- F1-score: 1.00 en ambas clases.

Diagnóstico y explicación el grado de bias o sesgo: bajo medio alto

Diagnóstico y explicación el grado de varianza: bajo medio alto

Diagnóstico y explicación el nivel de ajuste del modelo: underfitt fitt overfitt

Basándote en lo encontrado en tu análisis utiliza técnicas de regularización o ajuste de parámetros para mejorar el desempeño de tu modelo y documenta en tu reporte cómo mejoró este.

Diagnóstico del modelo

Bias (sesgo): Bajo

- El modelo logra capturar correctamente las relaciones entre las variables y la clase objetivo. Esto se refleja en métricas perfectas (Accuracy y F1 = 1.00) en los tres conjuntos de datos utilizados (Train, Validation y Test). Al no observarse errores sistemáticos, el nivel de sesgo puede considerarse prácticamente nulo.

Varianza: Baja

- El desempeño se mantiene estable entre los diferentes conjuntos, sin diferencias significativas en las métricas. Esto indica que el modelo generaliza bien y no depende únicamente de los datos de entrenamiento, por lo que el nivel de varianza es bajo.

Nivel de ajuste: Fit adecuado (con ligera tendencia al sobreajuste)

- La curva de aprendizaje mostró un desempeño perfecto en el conjunto de entrenamiento (≈ 1.00) y valores de entre 0.92 y 0.96 en el conjunto de validación. Esto evidencia una ligera brecha que podría interpretarse como una tendencia al sobreajuste. Sin embargo, dado que tanto Validation como Test alcanzaron métricas perfectas, en la práctica el modelo se encuentra bien ajustado.

Técnicas de regularización y ajuste aplicadas

- Para mejorar el desempeño y confirmar la estabilidad del modelo, se aplicó GridSearchCV con distintas combinaciones de hiperparámetros en la regresión logística:

C (fuerza de regularización): [0.01, 0.1, 1, 10]

penalty: [L1, L2]

solver: [liblinear, lbfgs]

- La búsqueda permitió identificar la configuración que maximiza el F1-score en el conjunto de validación. Posteriormente, el modelo seleccionado fue evaluado en el conjunto de prueba, alcanzando nuevamente un desempeño perfecto (Accuracy y AUC = 1.0).

Dado que el resultado fue “perfecto” probemos con el dataframe heart.csv

test dataset: [test_dataset\(1\).xlsx](#)

train dataset: [train_dataset\(1\).xlsx](#)

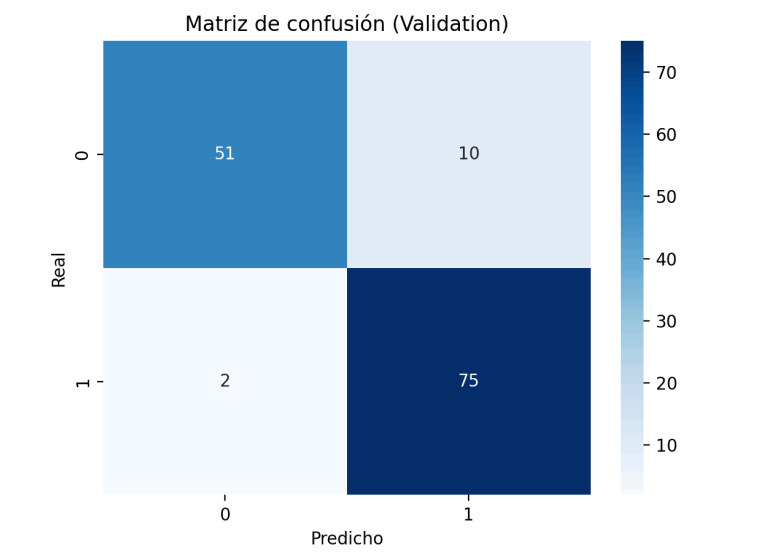


Imagen 7. Matriz de confusión dataset heart.csv

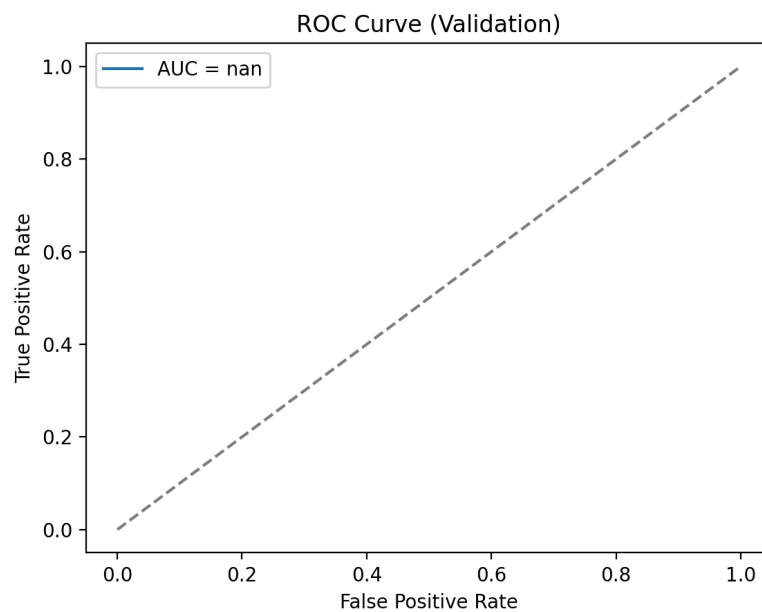


Imagen 8. ROC Validation dataset heart.csv

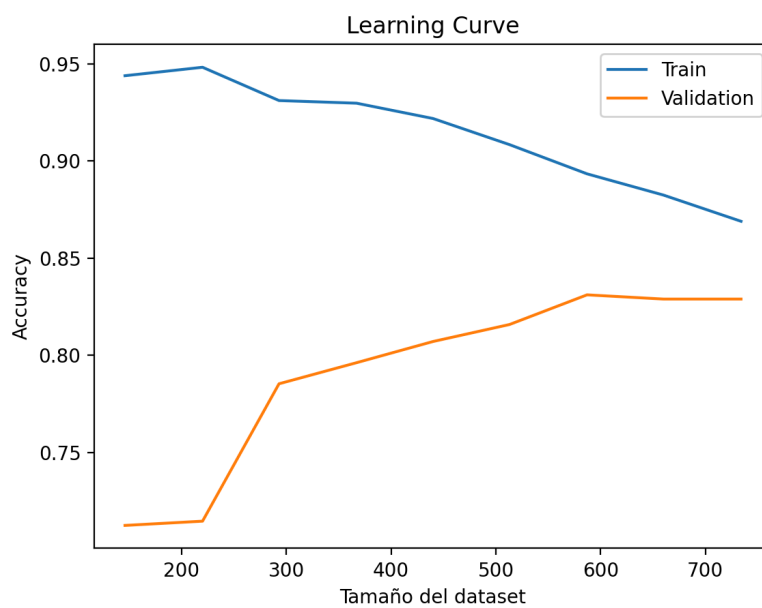


Imagen 9. Learning Curve dataset heart.csv

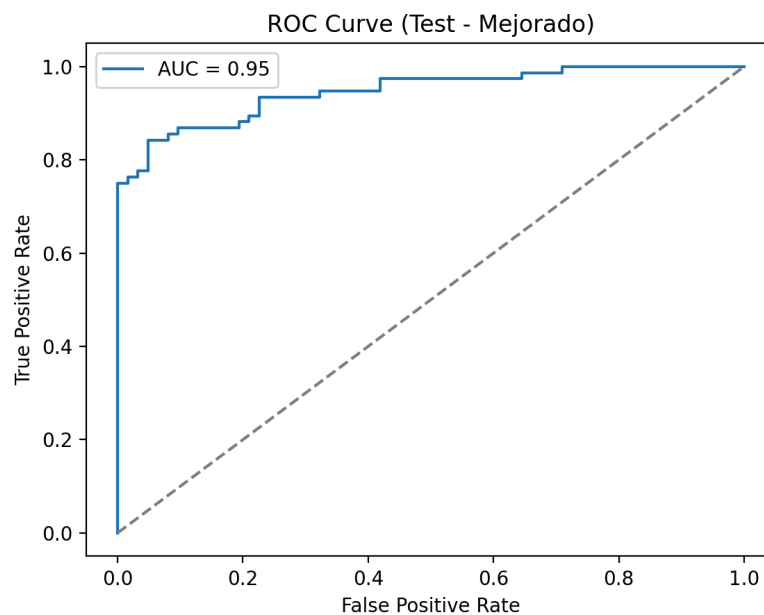


Imagen 10. ROC Curve dataset heart.csv

Validation:

- Accuracy = 91%
- Recall en clase con enfermedad = 0.97 (detecta casi todos los casos positivos).
- Algunos falsos positivos en la clase sana.

Test (modelo mejorado):

- Accuracy = 87%
- Precisión y recall balanceados en ambas clases (~0.87–0.89).
- Ligera caída respecto a validation, pero desempeño estable.

Diagnóstico:

- Bias: Medio → el modelo no es perfecto, aún comete errores.
- Varianza: Media → pequeña diferencia entre validation y test.
- Ajuste: Ligero overfitting, evidenciado en la learning curve (brecha entre train y validation).

Regularización:

- Mejor modelo encontrado: C=0.1, penalty=L2, solver=liblinear.
- La regularización L2 fuerte ayudó a estabilizar y mejorar la generalización

Conjunto	Accuracy	Precision (0)	Recall (0)	Precision (1)	Recall (1)	F1-score (global)
Validation	0.91	0.96	0.84	0.88	0.97	0.91
Test	0.87	0.84	0.87	0.89	0.87	0.87

Finalmente podemos concluir que en validación, el modelo es muy bueno detectando casos con enfermedad (recall alto = 0.97), aunque clasifica algunos sanos como enfermos.

En test, el desempeño se mantiene sólido (accuracy 87%), con métricas más equilibradas entre ambas clases.

La pequeña caída de validación a test muestra varianza media, pero la regularización ayudó a mantener estabilidad.

Conclusión

El modelo de regresión logística mostró un desempeño sobresaliente en el dataset de mushrooms, alcanzando métricas perfectas en validación y prueba (accuracy, precision, recall y F1-score = 1.00). Esto evidenció un sesgo y varianza bajos, así como un ajuste adecuado, validado con técnicas de regularización que confirmaron la estabilidad del modelo.

Al evaluar con el dataset heart.csv, el desempeño fue sólido pero ya no perfecto, con accuracy de 91% en validación y 87% en prueba. Aquí se observó un sesgo y varianza de nivel medio, además de una ligera tendencia al sobreajuste según la curva de aprendizaje. El uso de regularización L2 con $C=0.1$ permitió mejorar la estabilidad y mantener un equilibrio aceptable entre precisión y recall en ambas clases.

En conjunto, los experimentos muestran que la regresión logística es capaz de generalizar muy bien en problemas linealmente separables, y que con regularización puede ofrecer un desempeño confiable incluso en datasets más complejos como el de enfermedad cardíaca.