Instituto Tecnológico y de Estudios Superiores de Monterrey
Campus Estado de México

TC3006C
Inteligencia Artificial Avanzada para la Ciencia de Datos I

Módulo 2
Aprendizaje automático

Análisis y Reporte sobre el desempeño del modelo

Evidencia del portafolio de análisis

Profesores

Mtro. Alberto Michel Pérez Domínguez

Dra. Andrea Torres Calderón

Mtro. David Higuera Rosales

Dra. Elisabetta Crescio

Dr. Jorge Adolfo Ramirez Uresti

Dr. Julio Guillermo Arriaga Blumenkron

Dr. Victor Adrián Sosa Hernández

Profesor del módulo
Dr. Jorge Adolfo Ramírez Uresti

Grupo 501

Maximiliano De La Cruz Lima A01798048

14 – Septiembre – 2025

1. Introducción

El objetivo de esta evidencia es el análisis y mejora sobre el desempeño de un algoritmo de aprendizaje automático (machine learning), en este caso específico de un clasificador multiclase basado en árboles de decisión mediante el uso de la librería de *scikit-learn*, con la finalidad de que el análisis se realice sobre las implementaciones de calidad que ofrece este popular framework.

A partir de diferentes configuraciones de hipermarámetro y técnicas de regularización, se busca analizar el impacto del sesgo, la varianza y el nivel de ajuste (underfitting, overfitting o fit) sobre la capacidad de generalización del modelo sobre las características de un dataset determinado, específicamente el de Palmer Penguins, un set de datos que busca ser una extensión del popular Iris, ambos muy populares en primera implementaciones de algoritmos de machine learning para tareas de clasificación multiclase.

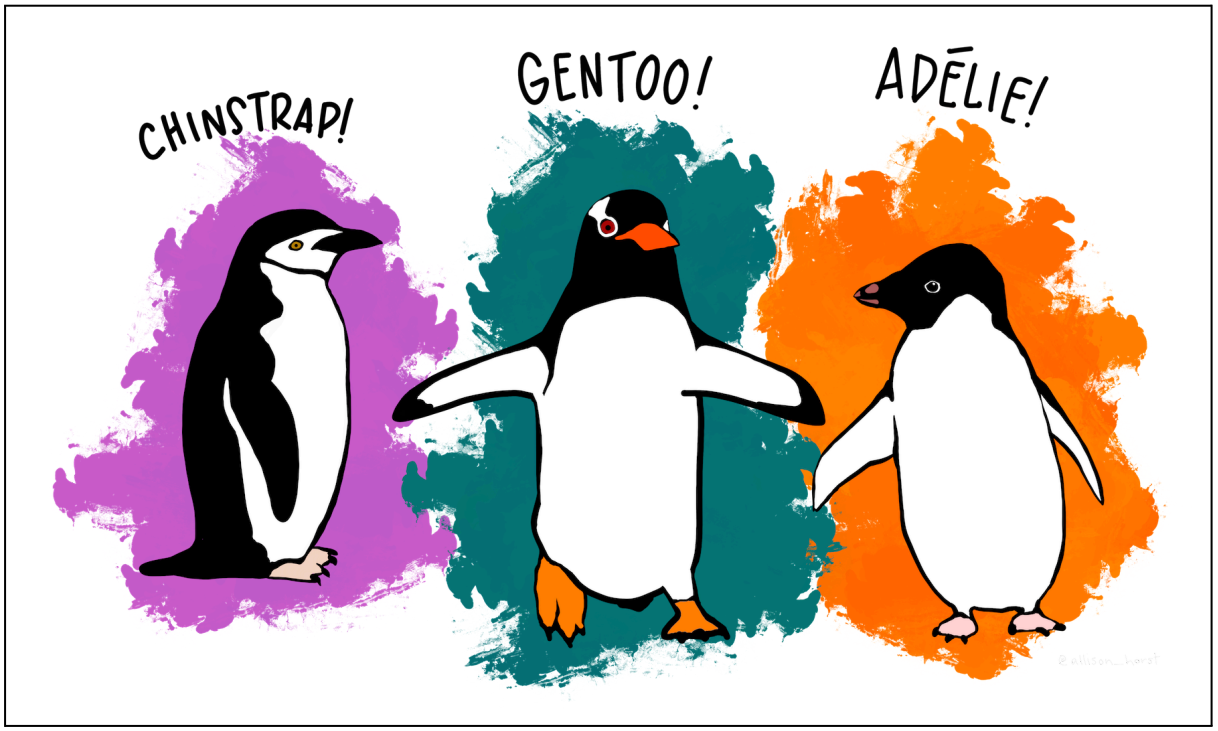
2. Metodología

a) Dataset empleado

El dataset Palmer Penguins consta de 344 muestras distribuidas en tres especies o clases de pingüinos: *Adelie, Gentoo* y *Chinstrap*; con características que describen cada muestra como isla, longitud y profundidad del pico, longitud de la aleta, masa corporal y sexo.

Figura 1

Especies de pingüino del dataset



Nota. Ilustración de las especies de pingüino del dataset obtenida de Análisis Exploratorio de Datos Palmer Penguins [figura], J, Portilla, 2023, Kaggle (https://www.kaggle.com/code/joeportilla/an-lisis-exploratorio-de-datos-palmer-penguins)

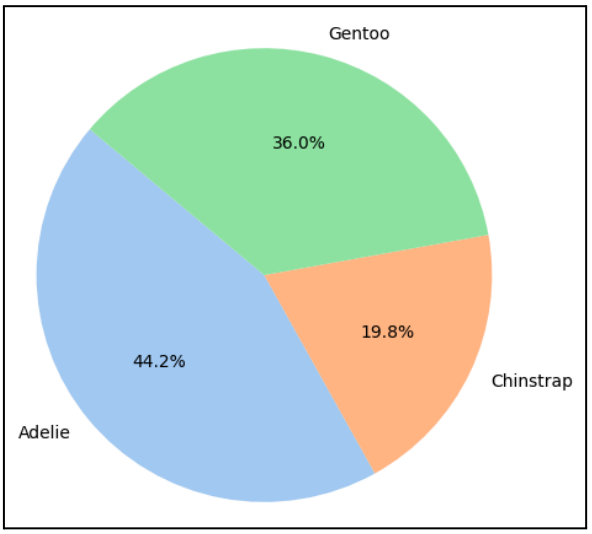
Sin embargo, las clases en este dataset no se encuentran balanceadas, sino que presentan un moderado desbalance entre ellas, contando con una distribución de *44.2 %* para *Adelie*, *36.0 %* para *Gentoo* y *19.8 % para Chinstrap*, teniendo esto un efecto negativo sobre el desempeño del clasificador, particularmente sobre la *accuracy*, bastando con que el modelo clasifique correctamente las clases mayoritarias para obtener un desempeño destacado pero engañoso.

Se realizó una partición del dataset en tres conjuntos, con una proporción de 20% para el conjunto de validación y otro 20% para el de prueba:

- Conjunto de entrenamiento: 220 muestras
- Conjunto de validación: 55 muestras
- Conjunto de pruebas: 69 muestras

Figura 2

Distribución de las clases en el dataset



b) Preprocesamiento

Antes de utilizar el modelo y sus variantes, se implementó una estrategia de preprocesamiento que incluye:

- Imputación de valores faltantes: uso de la mediana en variables numéricas y la moda en variables categóricas.
- Codificación *One-Hot Encoding* en las variables categóricas: island y sex.
- Concatenación de variables numéricas y categóricas transformadas para tener un único conjunto de características.

c) Ajuste de hiperparámetros

Para seleccionar los parámetros iniciales del Modelo 1, se empleó *GridSearchCV con Cross-validation* de 5 pliegues y métrica de optimización basada en el *F1-score macro.*

En este caso, el espacio de búsqueda incluyó los parámetros:

- Criterios: gini, entropy, log_loss
- *max_depth*: None, 3, 4, 5, 6, 8, 10
- *min_sample_split*: 2, 5, 10, 20
- *min_sample_leaf*: 1, 2, 4, 8, 10
- *ccp_alpha*: 0.0, 0.001, 0.01

El mejor árbol de decisión con un resultado $F1\ macro\ =\ 0.9620$ fue un árbol con hiperparámetros por defecto, mostrando el mejor desempeño aun utilizando valores por defecto. No obstante, dado la alta cantidad de recursos computacionales y que incrementa considerablemente el tiempo de ejecución, se optó por eliminarlo del código final, dejando mención al respecto en este reporte.

d) Métricas de evaluación

El desbalance del dataset, en contraste con uno con clases perfectamente balanceadas como el anteriormente mencionado Iris, implica que la métrica *accuracy* no sea suficiente para evaluar el desempeño del modelo por las razones antes descritas en la distribución del dataset. Por ello, se utilizan las métricas *precision, recall* y *F1-score macro*, que ponderan equitativamente el desempeño en cada clase independientemente de su tamaño.

e) Modelos implementados

Trás realizarse varias pruebas ajustando hiperparḿetros y observando los resultados con el junto de validación, se determinó la evaluación cuatro configuraciones a utilizarse:

1. Árbol de decisión con hiperparámetros por defecto trás el resultado de *GridSearchCV*.
2. Árbol restringido mediante con una $min\_impurity\_decrease\ =\ 0.2$.
3. Árbol profundo sin algún tipo de regularización con $max\_depth\ =\ 10$.
4. Bagging con regularización, utilizando árboles de decisión de profundidad limitada y un ensamble de 50 estimadores.

3. Resultados

a) Primer modelo: árbol por defecto

i) Conjunto de validación

Figura 3

Matriz de confusión del primer modelo en el conjunto de validación

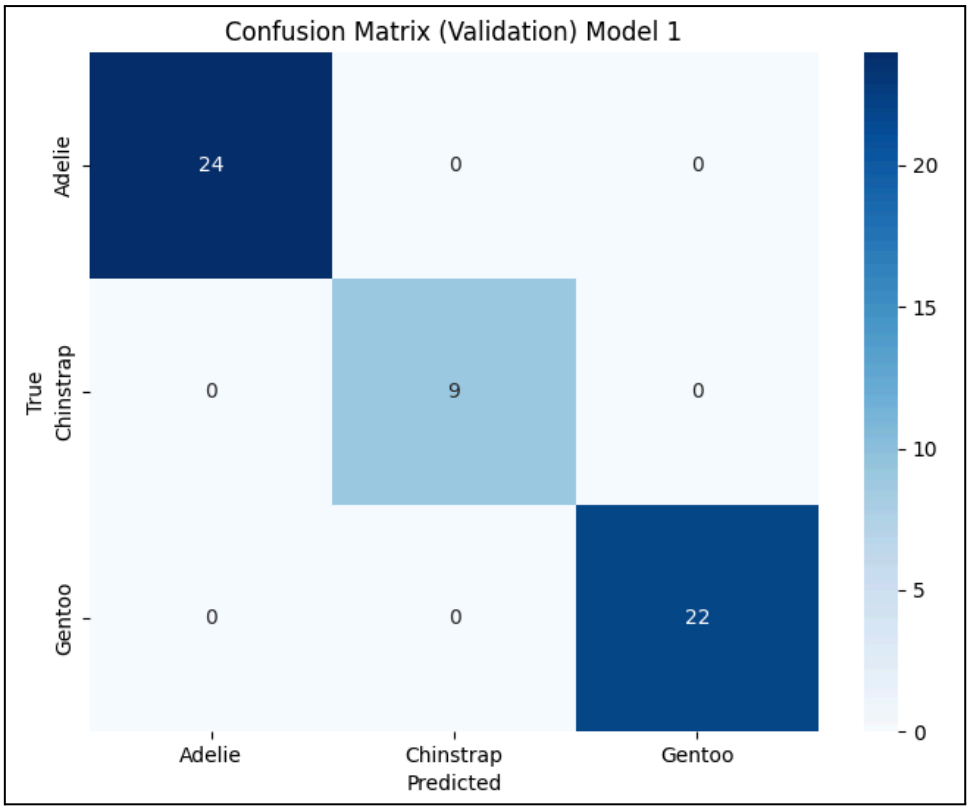


Tabla 1

Reporte de clasificación del primer modelo con el conjunto de validación

	precision	recall	f1-score	support	accuracy
adelie	1.00	1.00	1.00	24	
chinstrap	1.00	1.00	1.00	9	
gentoo	1.00	1.00	1.00	22	
macro avg	1.00	1.00	1.00	55	
					1.00

ii) Conjunto de prueba

Figura 4

Matriz de confusión del primer modelo en el conjunto de prueba

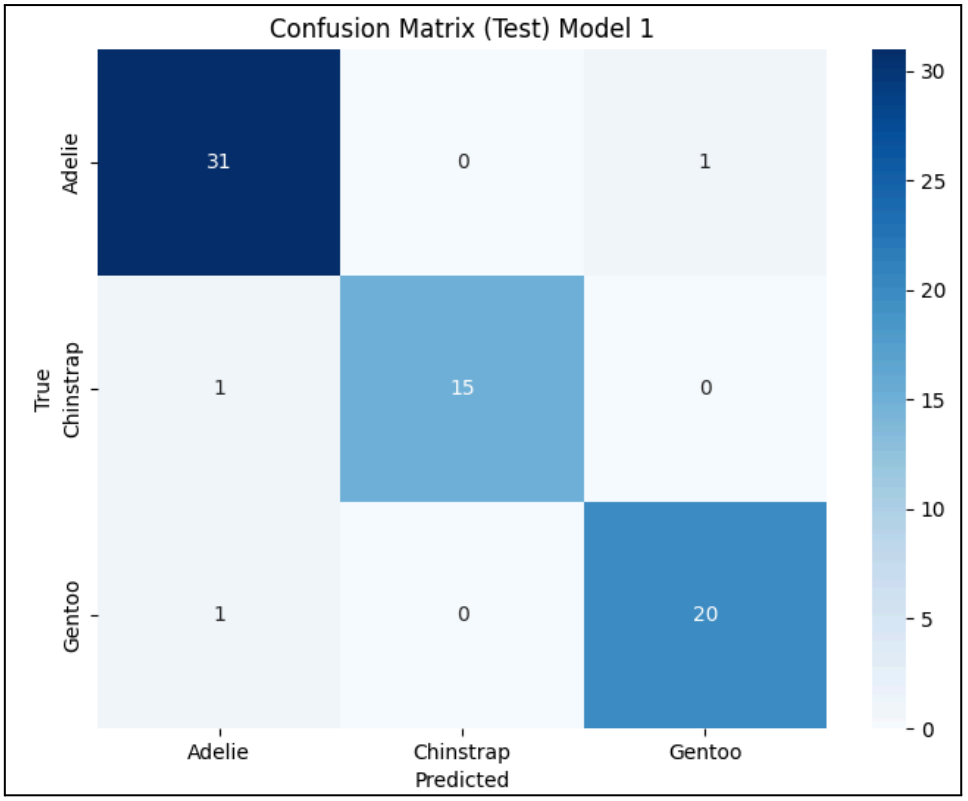


Tabla 2

Reporte de clasificación del primer modelo con el conjunto de prueba

	precision	recall	f1-score	support	accuracy
adelie	0.94	0.97	0.95	32	
chinstrap	1.00	0.94	0.97	16	
gentoo	0.95	0.95	0.95	21	
macro avg	0.96	0.95	0.96	69	
					0.96

b) Segundo modelo: árbol restringido

i) Conjunto de validación

Figura 5

Matriz de confusión del segundo modelo en el conjunto de validación

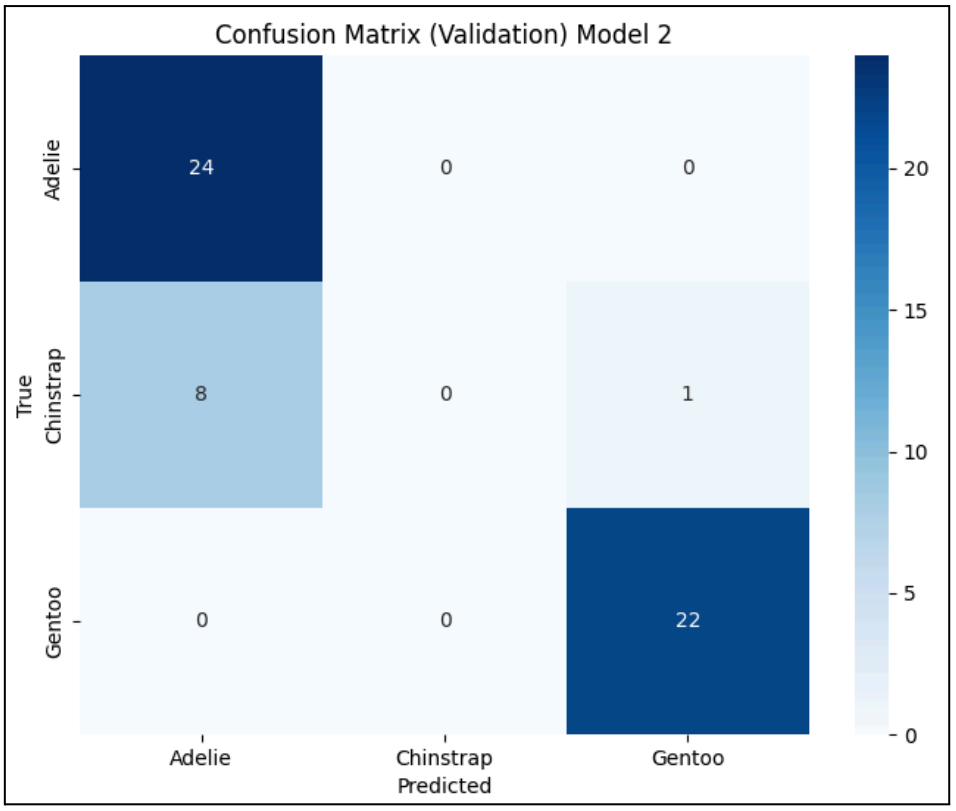


Tabla 3

Reporte de clasificación del segundo modelo con el conjunto de validación

	precision	recall	f1-score	support	accuracy
adelie	0.75	1.00	0.86	24	
chinstrap	0.00	0.00	0.00	9	
gentoo	0.96	1.00	0.98	22	
macro avg	0.57	0.67	0.61	55	
					0.84

ii) Conjunto de prueba

Figura 6

Matriz de confusión del segundo modelo en el conjunto de prueba

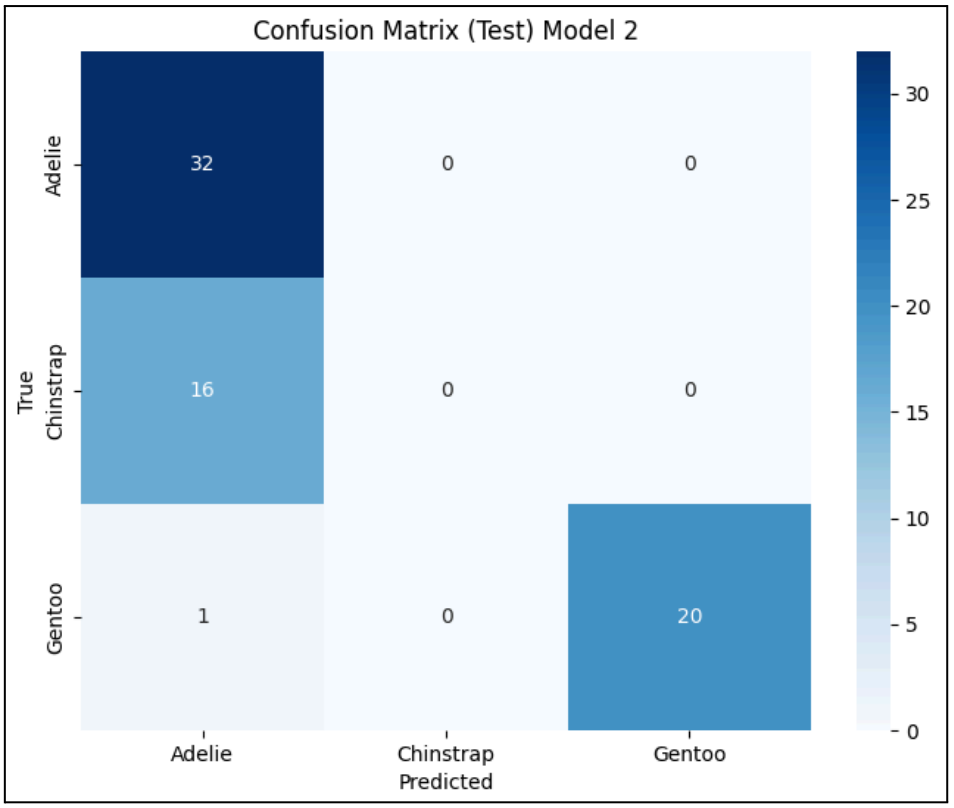


Tabla 4

Reporte de clasificación del segundo modelo con el conjunto de prueba

	precision	recall	f1-score	support	accuracy
adelie	0.65	1.00	0.79	32	
chinstrap	0.00	0.00	0.00	16	
gentoo	1.00	0.95	0.98	21	
macro avg	0.55	0.65	0.59	69	
					0.75

c) Tercer modelo: árbol profundo

i) Conjunto de validación

Figura 7

Matriz de confusión del tercer modelo en el conjunto de validación

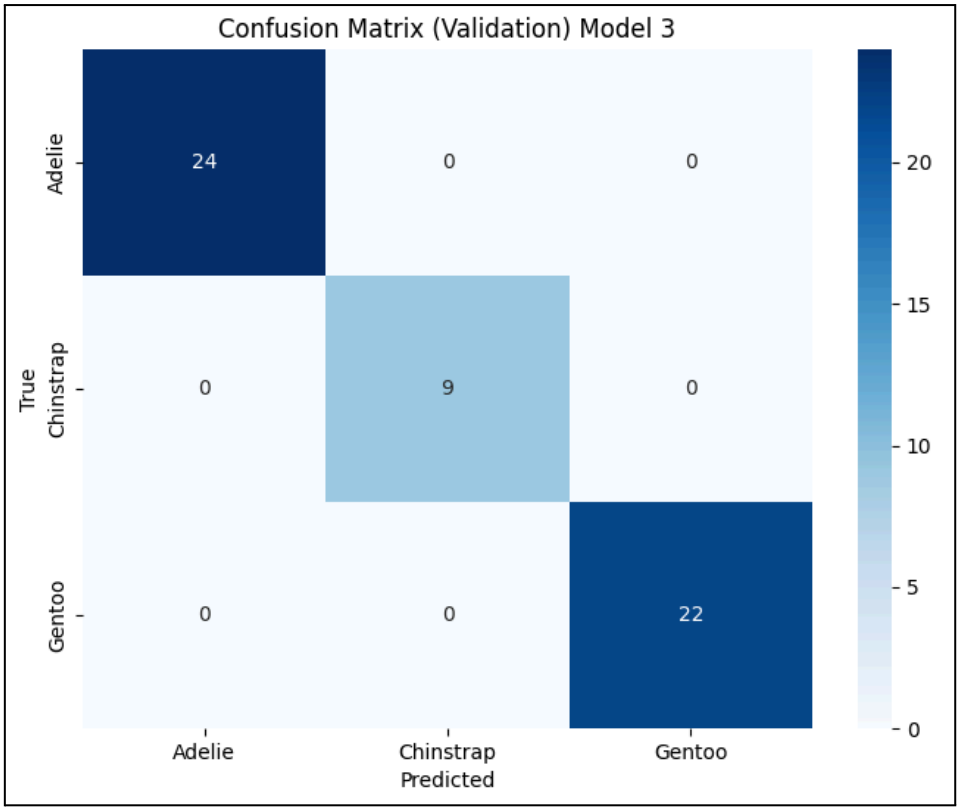


Tabla 5

Reporte de clasificación del tercer modelo con el conjunto de validación

	precision	recall	f1-score	support	accuracy
adelie	1.00	1.00	1.00	24	
chinstrap	1.00	1.00	1.00	9	
gentoo	1.00	1.00	1.00	22	
macro avg	1.00	1.00	1.00	55	
					1.00

ii) Conjunto de prueba

Figura 8

Matriz de confusión del tercer modelo en el conjunto de prueba

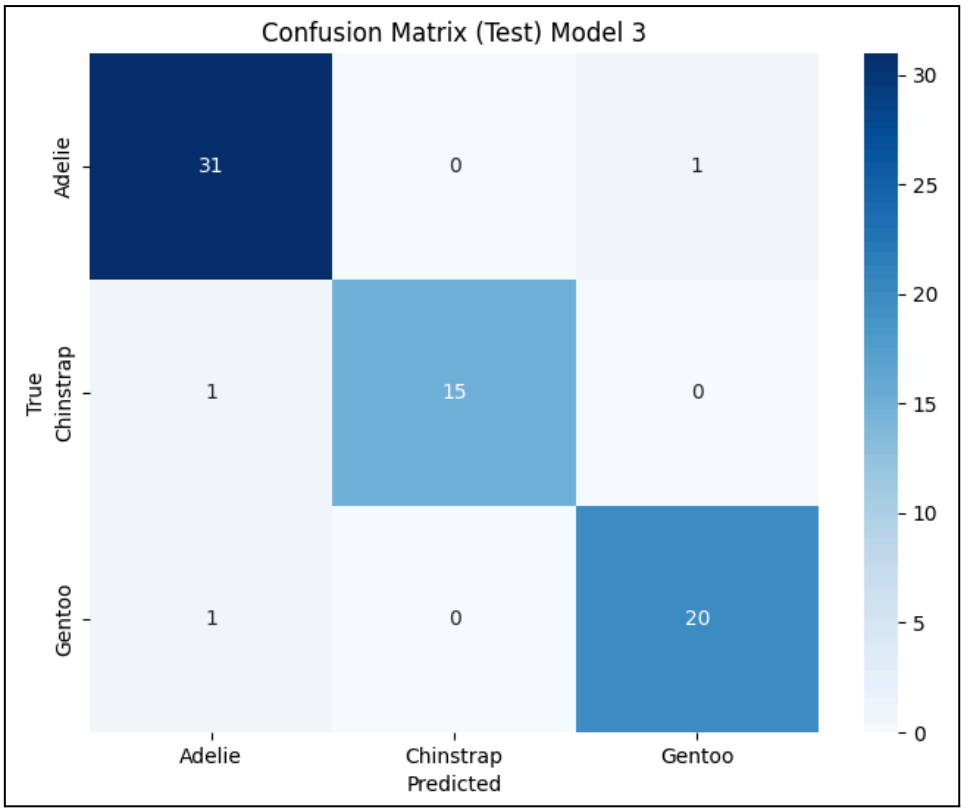


Tabla 6

Reporte de clasificación del tercer modelo con el conjunto de prueba

	precision	recall	f1-score	support	accuracy
adelie	0.94	0.97	0.95	32	
chinstrap	1.00	0.94	0.97	16	
gentoo	0.95	0.95	0.95	21	
macro avg	0.96	0.95	0.96	69	
					0.96

d) Cuarto modelo: bagging con regularización

i) Conjunto de validación

Figura 9

Matriz de confusión del cuarto modelo en el conjunto de validación

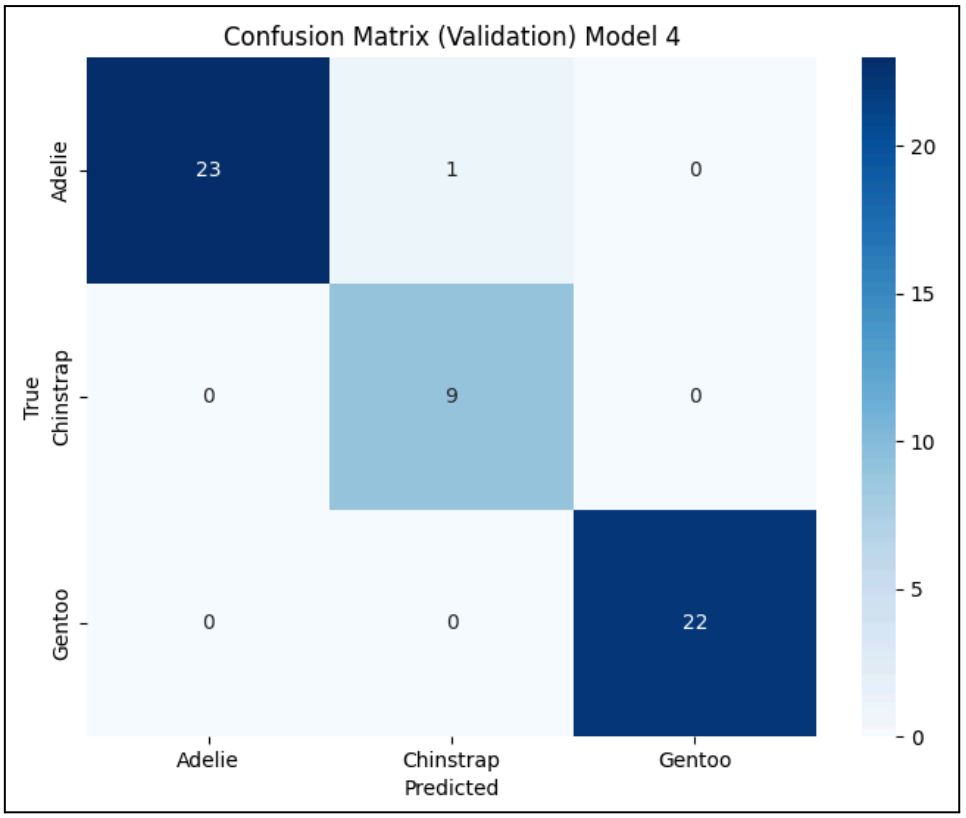


Tabla 7

Reporte de clasificación del cuarto modelo con el conjunto de validación

	precision	recall	f1-score	support	accuracy
adelie	1.00	0.96	0.98	24	
chinstrap	0.90	1.00	0.95	9	
gentoo	1.00	1.00	1.00	22	
macro avg	0.97	0.99	0.98	55	
					0.98

ii) Conjunto de prueba

Figura 10

Matriz de confusión del cuarto modelo en el conjunto de prueba

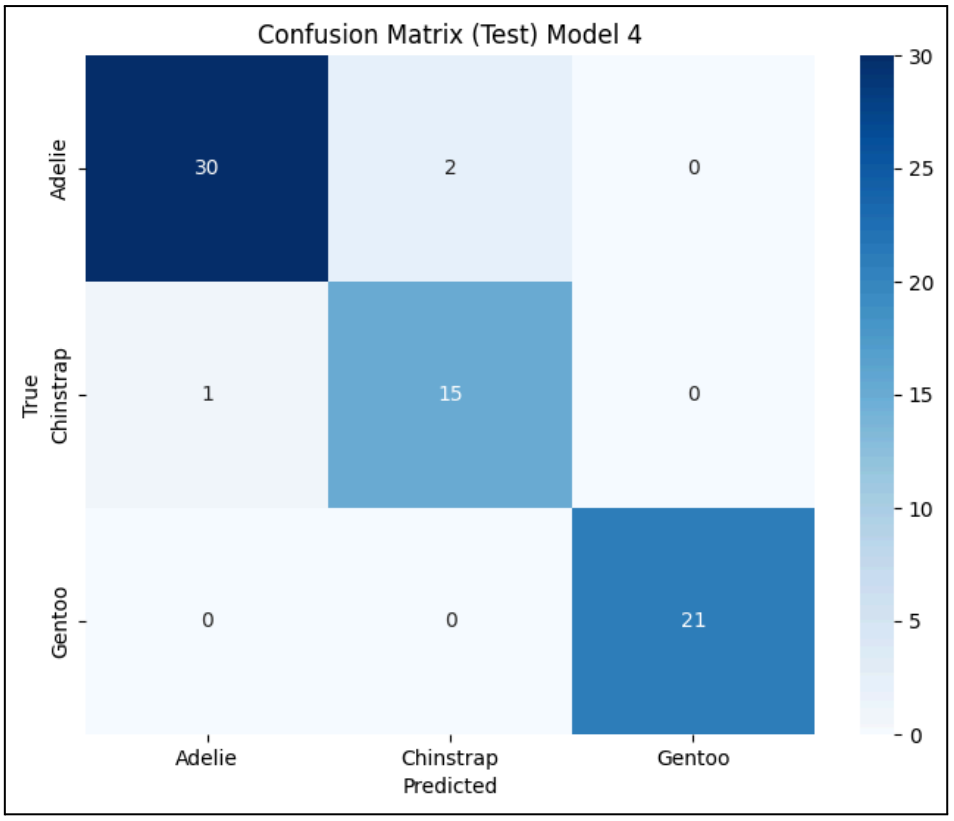


Tabla 8

Reporte de clasificación del cuarto modelo con el conjunto de prueba

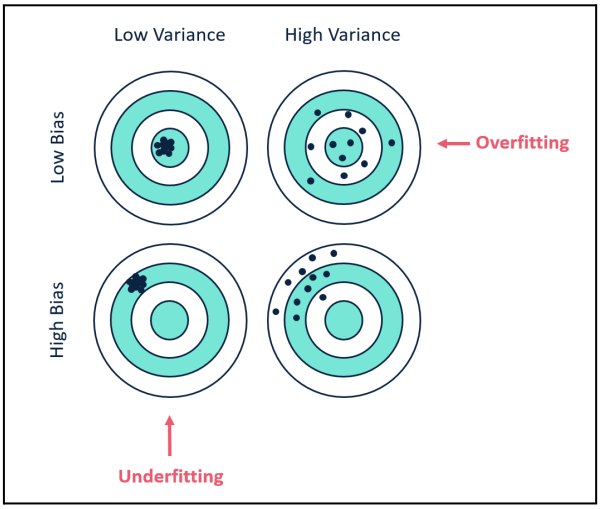
	precision	recall	f1-score	support	accuracy
adelie	0.97	0.94	0.95	32	
chinstrap	0.88	0.94	0.97	16	
gentoo	1.00	1.00	1.00	21	
macro avg	0.95	0.96	0.95	69	
					0.96

4. Análisis de resultados

- El primer modelo (con valores predeterminados) muestra una capacidad para generalizar las características del dataset adecuada, con un sesgo y varianza media que lo hacen susceptible a sobreajuste en sets más pequeños al mostrar diferencias moderadas entre el conjunto de prueba con respecto al de entrenamiento y validación.
- El segundo modelo muestra como un árbol excesivamente restrictivo no logra capturar patrones relevantes de las características del set de datos, con efectos críticos en la clase minoritaria; consecuencia de un claro underfitting debido a un alto sesgo y baja varianza.
- El tercer modelo ilustra como un árbol demasiado profundo llega a memorizar los datos del conjunto de entrenamiento, llegando a ser muy complejo y no mostrar mejoras significativas con el conjunto de prueba. Por consiguiente, es posible afirmar que se trata de un ejemplo evidente de overfitting, consecuencia de un bajo sesgo y una alta varianza.
- El cuarto modelo es un indicativo de las mejoras que se pueden lograr sobre los árboles de decisión estándar mediante la implementación de una estrategia de bagging con regularización, sobresaliendo sobre el resto al ser el único que muestra un desempeño consistente a lo largo de los tres conjuntos, teniendo por lo tanto un fit adecuado debido a sus bajos niveles tanto de sesgo como de varianza.

Figura 11

Relación entre el nivel de sesgo y varianza con el underfitting, fit adecuado u overfitting



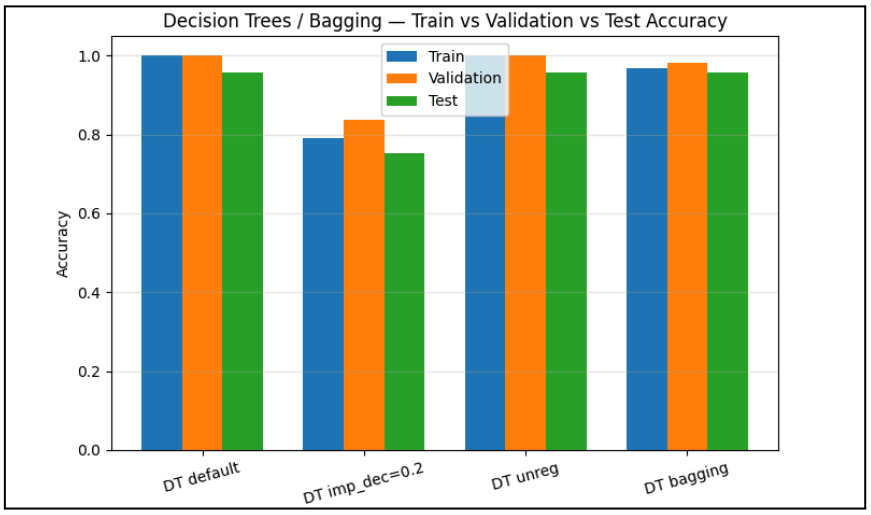
Nota. Ejemplo ilustrativo del sesgo y varianza con juntos sobre un objetivo, obtenida de Bias Versus Variance [figura], S, Firmin, 2019, Alteryx (https://community.alteryx.com/t5/Data-Science/Bias-Versus-Variance/ba-p/351862)

La métrica de *accuracy* fue elevada en la mayoría de modelos con la excepción del segundo modelo, donde el desbalance de clases tuvo un efecto significativo e hizo que fuese incapaz de clasificar correctamente a la clase minoritaria (*Chinstrap*). Por tanto, el uso de *F1-score macro* permite capturar deficiencias como estas, donde un análisis basado enteramente en *accuracy* habría ocultado el problema.

Finalmente, con el objetivo de sintetizar los desempeños analizados, se elaboró la gráfica comparativa vista a continuación:

Figura 12

Comparación de accuracy en entrenamiento, validación y prueba entre los cuatro modelos



5. Conclusiones

- La separación del dataset en los conjuntos *train/validation/test* permitió diagnosticar de forma precisa el grado de sesgo, varianza y ajuste en la configuración de hiperparámetros de los modelos.
- El *Bagging* con regularización (cuarto modelo) ofreció el mejor equilibrio entre sesgo y varianza, mejorando el *recall* sobre la clase minoritaria y manteniendo un *F1-score macro* destacado con un valor de 0.95 en el conjunto de pruebas.
- Se evidencia que la regularización combinada con técnicas de ensamble permite obtener modelos con buen desempeño, evitando tanto el underfitting como el overfitting.
- El árbol de decisión restrictivo (segundo modelo) presentó underfitting por un sesgo demasiado alto, mientras que el árbol de decisión profundo (tercer modelo) mostró un claro overfitting por alta varianza.
- El árbol por defecto (primer modelo) logró un buen desempeño en general, aunque con cierta tendencia al overfitting.
- La métrica *F1-score macro* permite evaluar el verdadero desempeño de modelos donde se presentan circunstancias de desbalance entre clases, donde un análisis enteramente basado en *accuracy* habría resultado engañoso y ocultado los problemas de los modelos.
- De manera general, los árboles de decisión tienden a padecer de overfitting en caso de no ser regularizados al iterar sobre cada una de las características de las instancias y crecer demasiado. Por ello, el uso de técnicas de ensamble como *Bagging* o *Random Forest* resulta altamente recomendable para controlar tanto el sesgo como la varianza y obtener mejores modelos que puedan ser puestos en producción.

6. Referencias

Firmin, S. (2019, Enero 31). *Bias Versus Variance - Data Science*. Alteryx Community. https://community.alteryx.com/t5/Data-Science/Bias-Versus-Variance/ba-p/351862

Gorman, K., Horst, A., & Hill, A. (n.d.). *palmerpenguins*. allisonhorst. https://allisonhorst.github.io/palmerpenguins/

Portilla, J. S. (2023). *Análisis Exploratorio de Datos Palmer Penguins*. Kaggle. https://www.kaggle.com/code/joeportilla/an-lisis-exploratorio-de-datos-palmer-penguins

scikit-learn developers. (n.d.). *DecisionTreeClassifier — scikit-learn 1.7.2 documentation*. Scikit-learn. https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html#sklearn.tree.DecisionTreeClassifier

scikit-learn developers. (n.d.). *GridSearchCV — scikit-learn 1.7.2 documentation*. Scikit-learn. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html