



Tecnológico de Monterrey

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

Uso de framework o biblioteca de aprendizaje máquina para la implementación de una solución

Evidencia del portafolio de implementación

Profesores

Mtro. Alberto Michel Pérez Domínguez

Dra. Andrea Torres Calderón

Mtro. David Higuera Rosales

Dra. Elisabetta Crescio

Dr. Jorge Adolfo Ramírez 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

10 – Septiembre – 2025

1. Introducción

El objetivo de esta evidencia es la implementación de un algoritmo de aprendizaje automático, específicamente de un clasificador mediante árboles de decisión, haciendo uso de la librería *scikit-learn*, con la finalidad de evaluar su desempeño sobre un dataset con clases desbalanceadas y datos faltantes mediante el análisis de cómo diferentes configuraciones de hiperparámetros tienen un efecto sobre la capacidad de generalización de un modelo.

El dataset seleccionado para la evaluación del modelo fue el de Palmer Penguins, ampliamente utilizado en ámbitos de aprendizaje de algoritmos de aprendizaje automático para tareas de clasificación multiclase. Asimismo, se aplicaron técnicas de procesamiento como imputación y codificación; ajuste de hiperparámetros y evaluación del desempeño con métricas estándar en aprendizaje automático supervisado para clasificación.

2. Metodología

a) Dataset empleado

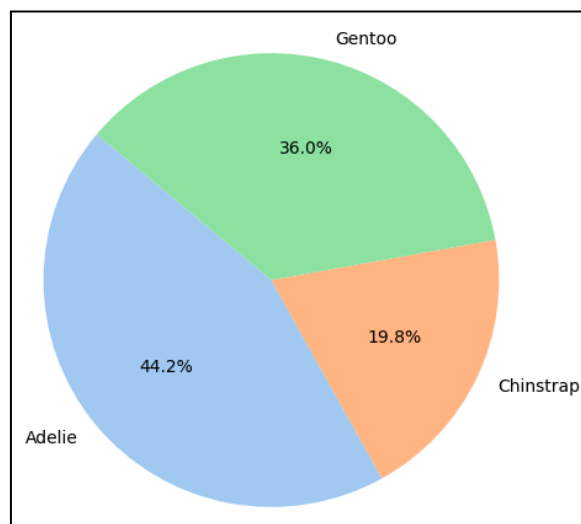
El dataset Palmer Penguins consta de 344 muestras distribuidas en tres especies (clases) de pingüinos: *Adelie*, *Gentoo* y *Chinstrap*; con cada muestra descrita por las características: isla, longitud y profundidad del pico, longitud de la aleta, peso corporal y sexo. No obstante, el dataset presenta un desbalance moderado en sus clases con una distribución de *Adelie* (44.2 %), *Gentoo* (36.0 %) y *Chinstrap* (19.8 %), pudiendo afectar el desempeño del clasificador.

Se dividió el dataset con una proporción de 20% para el conjunto de validación y otro 20% para el de prueba, teniendo la siguiente cantidad de muestras por conjunto:

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

Figura 1

Distribución de las clases en el dataset



b) Preprocesamiento

Antes de implementar el modelo, se construyó un pipeline de preprocesamiento con los siguientes pasos:

- 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.
- Separación entre columnas numéricas y categóricas mediante *ColumnTransformer*.

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

Con lo cual se obtuvo como mejor modelo un árbol de decisión con hiperparámetros por defecto tras tener un resultado *F1 macro* = 0.9620, confirmando que se tiene un buen desempeño utilizando los valores por defecto. Sin embargo, dado que este proceso incrementa considerablemente el tiempo de ejecución, se optó por comentar las líneas referentes a *GridSearchCV* en el código para agilizar la ejecución de las pruebas comparativas.

d) Métricas de evaluación

En contraste con un dataset balanceado como lo es *Iris*, este conjunto de datos (por su desbalance moderado) exige prestar mayor atención en métricas distintas a la exactitud (*accuracy*). Por lo tanto, se analizaron prioritariamente *precision*, *recall* y *F1-score*, con énfasis especial en el promedio macro debido a su visión equilibrada del desempeño del modelo en todas las clases, independientemente del tamaño.

e) Modelos implementados

Después de algunos ajustes de hiperparámetros y pruebas de evaluación, se establecieron los siguientes modelos a utilizar:

- Primer modelo: valores por defecto tras el resultado de *GridSearchCV*.
- Segundo modelo: árbol restringido mediante *min_impurity_decrease* = 0.2.
- Tercer modelo: árbol profundo.
- Cuarto modelo: bagging con regularización.

3. Resultados

a) Primer modelo

i) Conjunto de validación

Figura 2

Matriz de confusión del primer modelo en el conjunto de validación

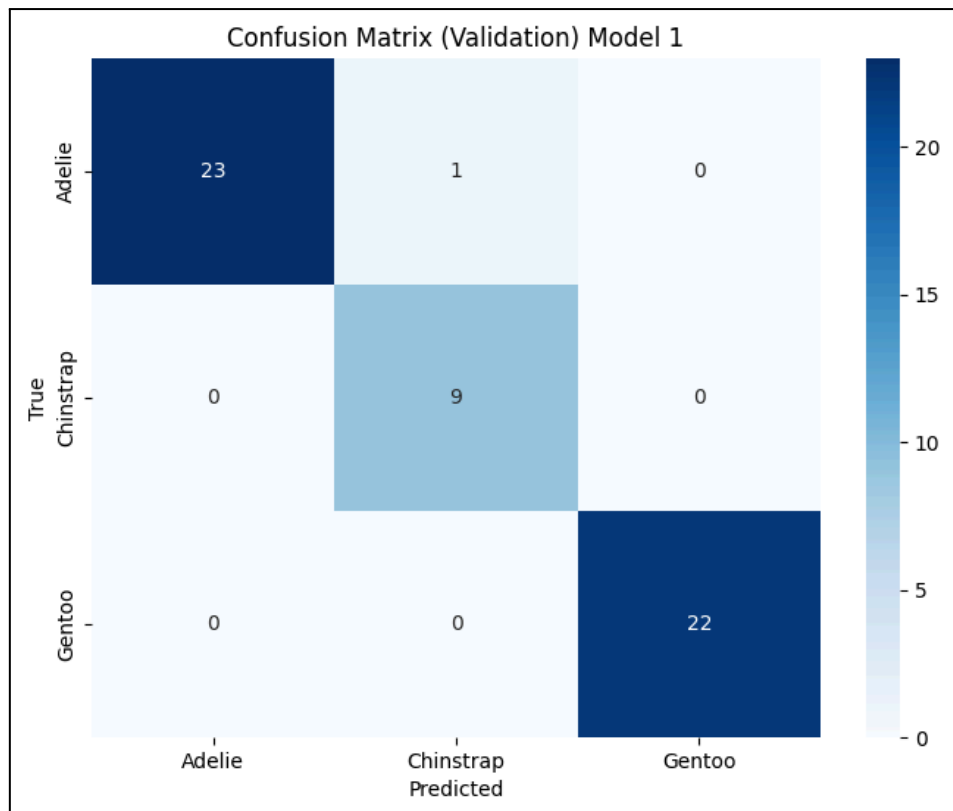


Tabla 1

Reporte de clasificación del primer modelo con el conjunto de validación

	precision	recall	f1-score	support	accuracy
adelie	1.00	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 3

Matriz de confusión del primer modelo en el conjunto de prueba

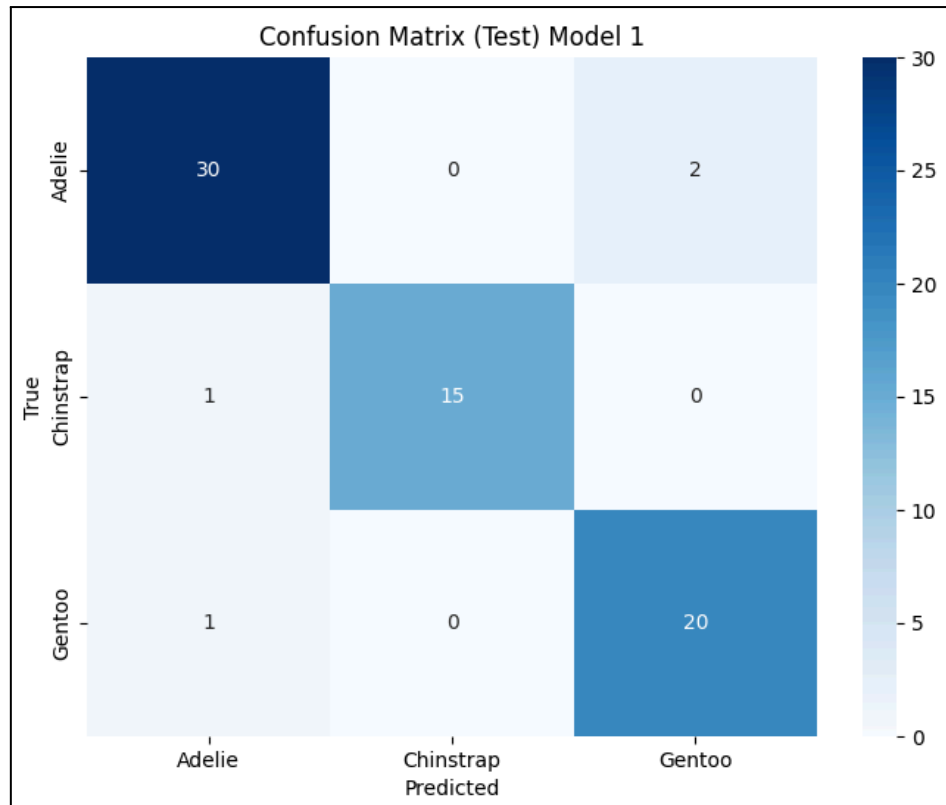


Tabla 2

Reporte de clasificación del primer modelo con el conjunto de prueba

	precision	recall	f1-score	support	accuracy
adelie	0.94	0.94	0.94	32	
chinstrap	1.00	0.94	0.97	16	
gentoo	0.91	0.95	0.93	21	
macro avg	0.95	0.94	0.95	69	
					0.94

b) Segundo modelo

i) Conjunto de validación

Figura 4

Matriz de confusión del segundo modelo en el conjunto de validación

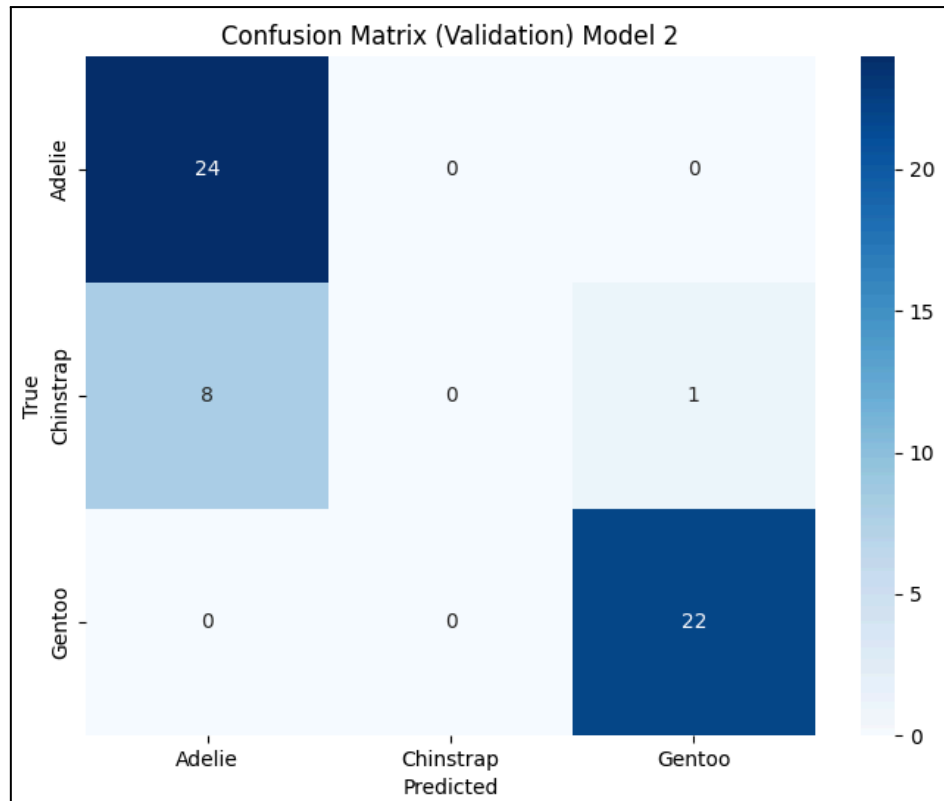


Tabla 3

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 5

Matriz de confusión del segundo modelo en el conjunto de prueba

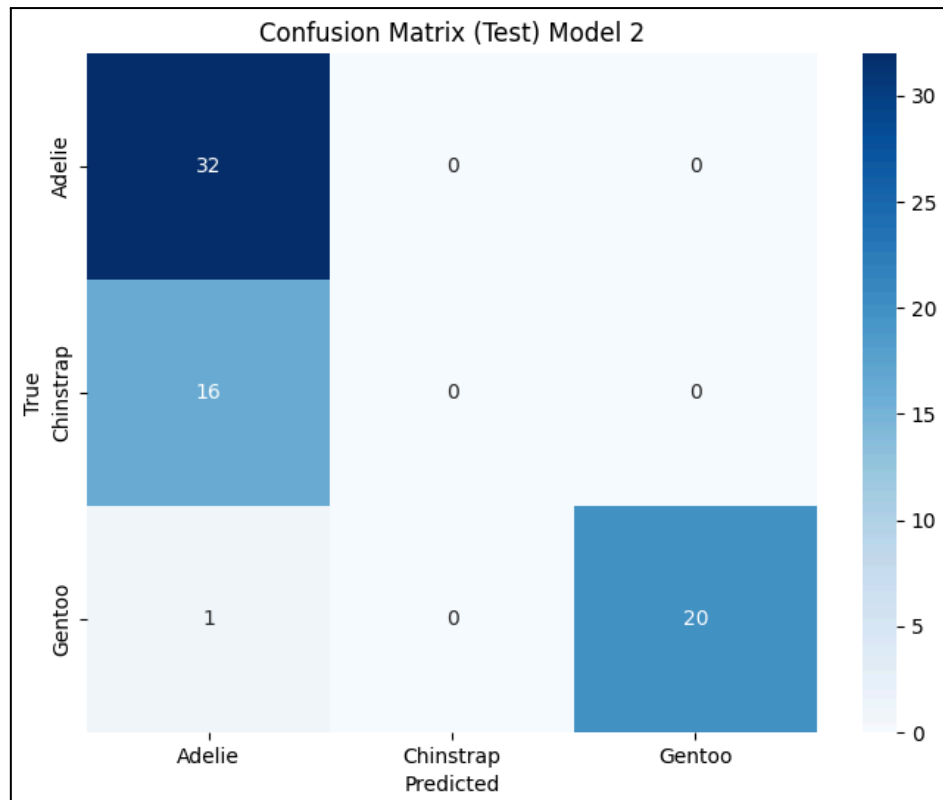


Tabla 4

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

i) Conjunto de validación

Figura 6

Matriz de confusión del tercer modelo en el conjunto de validación

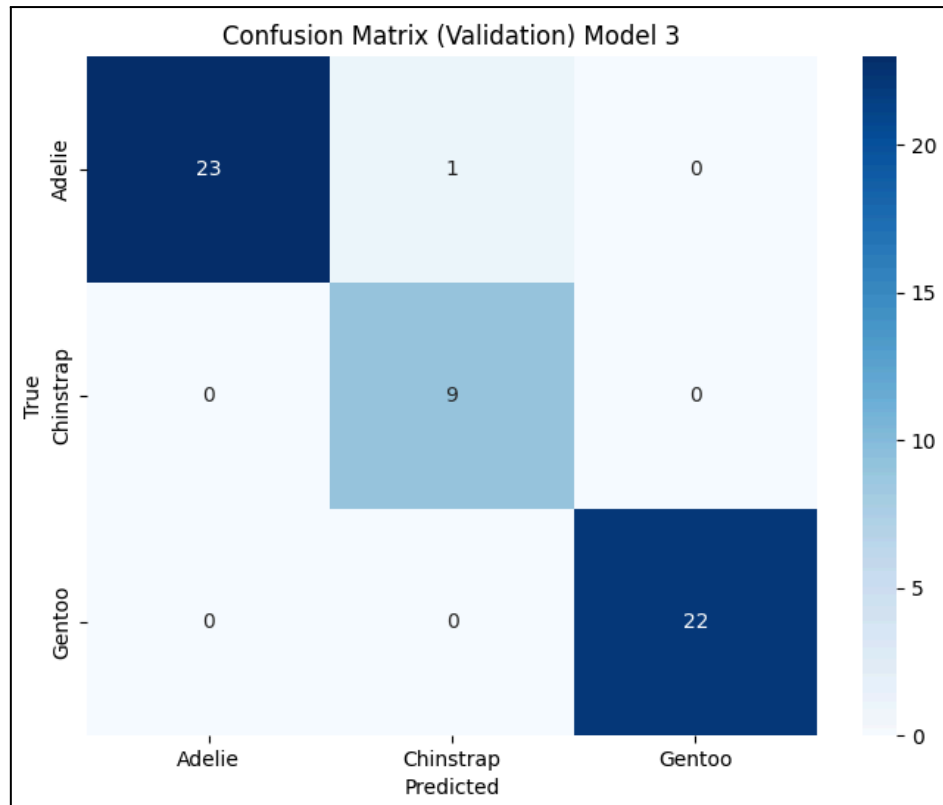


Tabla 5

Reporte de clasificación del tercer modelo con el conjunto de validación

	precision	recall	f1-score	support	accuracy
adelie	1.00	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 7

Matriz de confusión del tercer modelo en el conjunto de prueba

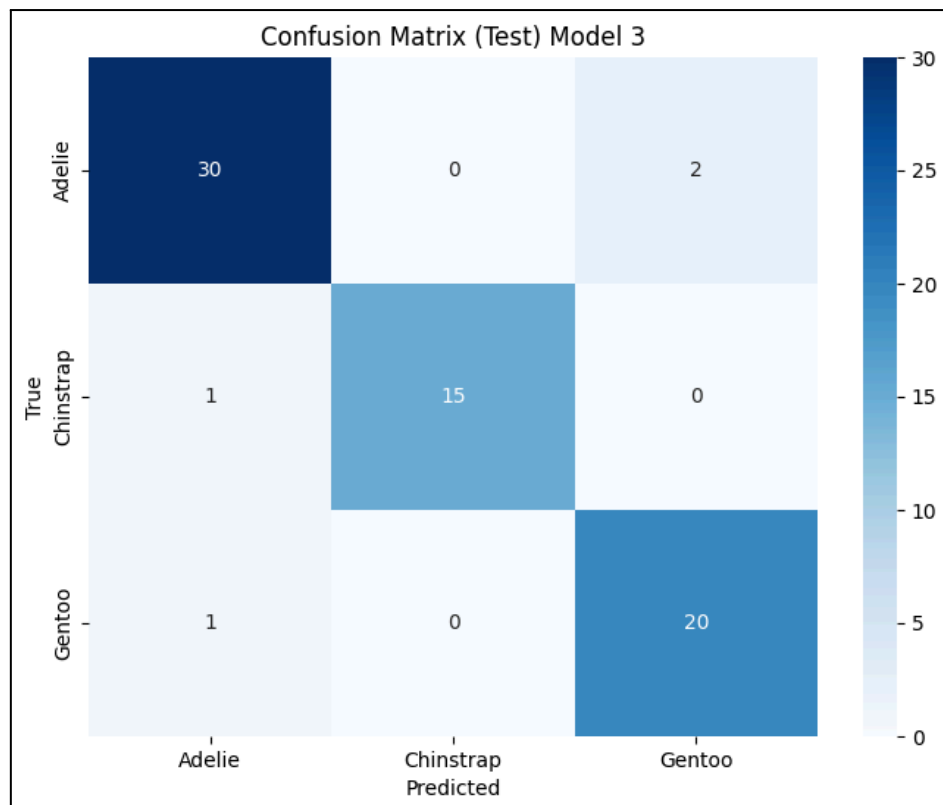


Tabla 6

Reporte de clasificación del tercer modelo con el conjunto de prueba

	precision	recall	f1-score	support	accuracy
adelie	0.94	0.94	0.94	32	
chinstrap	1.00	0.94	0.97	16	
gentoo	0.91	0.95	0.93	21	
macro avg	0.95	0.94	0.95	69	
					0.94

d) Cuarto modelo

i) Conjunto de validación

Figura 8

Matriz de confusión del cuarto modelo en el conjunto de validación

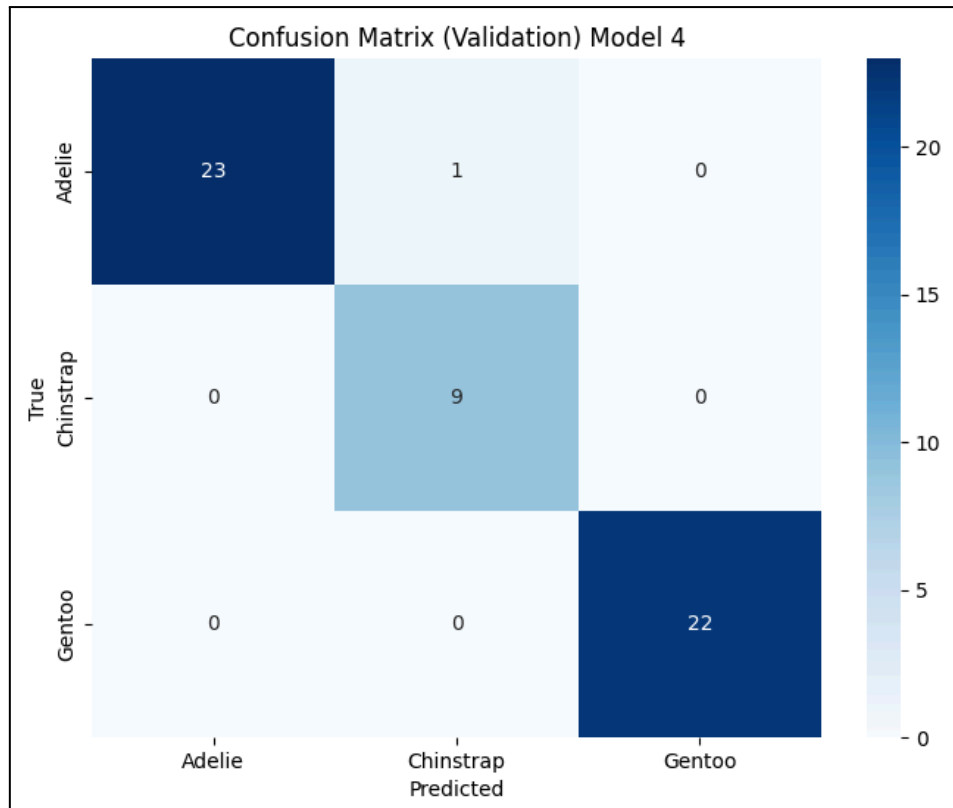


Tabla 7

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 9

Matriz de confusión del cuarto modelo en el conjunto de prueba

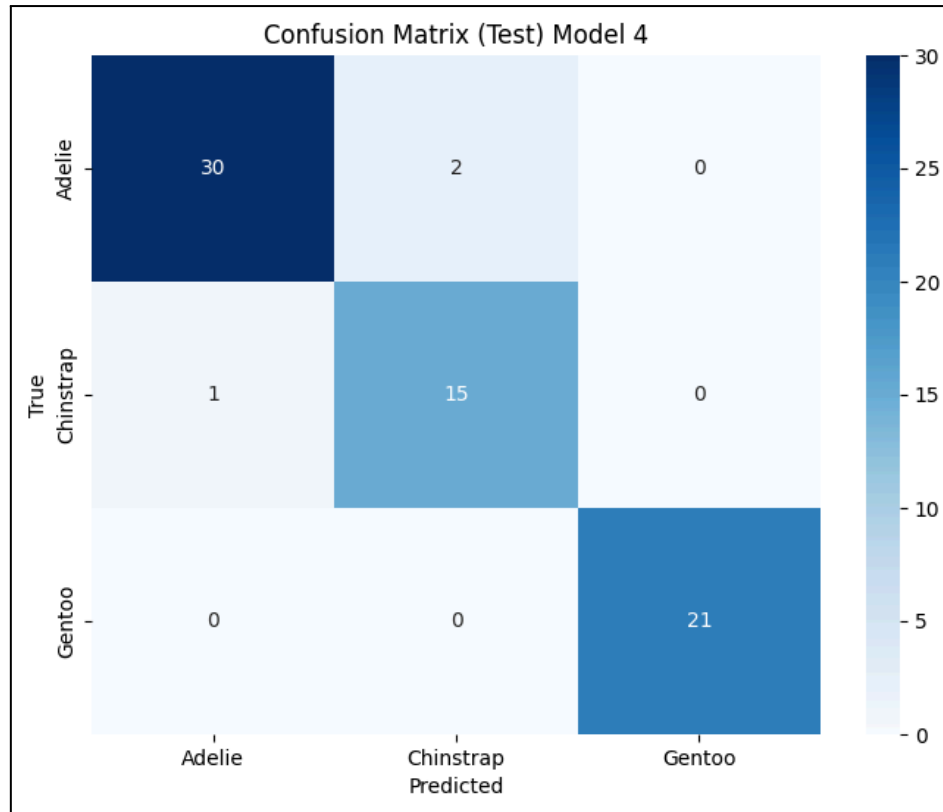


Tabla 8

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

- La métrica de *accuracy* fue elevada en la mayoría de modelos con la excepción del segundo modelo, pero no refleja el verdadero desempeño de estos en la clase minoritaria (*Chinstrap*).
- El segundo modelo ilustra como un árbol excesivamente restrictivo no logra capturar patrones de la clase minoritaria (underfitting), consecuencia de un sesgo alto y varianza baja.
- El tercer modelo presentó un leve nivel de overfitting, al crecer demasiado en su complejidad sin lograr mejoras significativas en las métricas obtenidas con el conjunto de prueba, pudiendo inferir un sesgo bajo y una varianza en un nivel medio.
- El primer modelo (con valores predeterminados) muestra una capacidad para generalizar adecuada, logrando por tanto un equilibrio gracias a un sesgo bajo y una varianza media que lo lleva a tener diferencias moderadas entre los conjuntos de validación y de prueba.
- El cuarto modelo (bagging) sobresalió sobre el resto al mejorar el recall de la clase minoritaria (*Chinstrap*) con 0.94 sin sacrificar el desempeño en Adelie y Gentoo, teniendo por tanto bajos niveles de sesgo y varianza que le otorgan un fit adecuado.

5. Conclusiones

- La métrica *F1-score macro* más apropiada para evaluar el desempeño del modelo dadas las circunstancias del desbalance entre clases, mostrando de esta manera que una *accuracy* similar tiene comportamientos muy distintos respecto a la clase minoritaria.
- El *Bagging* (cuarto modelo) ofreció el mejor balance F1 macro, con un valor de 0.95 con el conjunto de prueba, destacando por su gran desempeño general pero particularmente por su recall alto respecto a la clase minoritaria (*Chinstrap*).
- El uso de *GridSearchCV* confirma que los parámetros por defecto de un árbol de decisión son adecuados para un dataset pequeño y simple como Penguins o Iris, con *F1 Macro* de 0.962 en validación cruzada.
- Con respecto al sesgo, varianza y ajuste del modelo:
 - El segundo modelo presenta underfitting: alto sesgo y baja varianza.
 - El tercer modelo presenta ligero overfitting: bajo sesgo y alta varianza.
 - El primer y cuarto modelo presentan fit: bajos sesgo y baja varianza.
- En términos generales, los árboles de decisión tienden a sobre ajustarse (overfitting) ya que, de no ser regularizados, tienden a crecer demasiado y memorizar los datos de entrenamiento, formando reglas que resultan muy específicas y que no generalizan bien.

6. Referencias

Gorman, K., Horst, A., & Hill, A. (n.d.). *palmerpenguins*. allisonhorst.

<https://allisonhorst.github.io/palmerpenguins/>

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. Retrieved a, from

https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html

scikit-learn developers. (n.d.). *7.1. Pipelines and composite estimators* — *scikit-learn 1.7.2 documentation*. Scikit-learn. <https://scikit-learn.org/stable/modules/compose.html>