Trabajo 3 - Machine Learning

Miguel Calle [a], Samuel Mazo [a]

[a] Ingeniería Mecatrónica

Envigado, Antioquia, Colombia

22 de noviembre de 2024

Andrés Quintero Zea

Programación

Índice

1. Base de Datos **2**

2. Estrategia de Solución **2**

2.1. DecisionTreeClassifier 2

2.2. RandomForest 5

3. Estrategia de Comparación **7**

4. Conclusiones **9**

1 Base de Datos

La base de datos que se utiliza para este proyecto fue una de las bases de datos del UCI Machine Learning Repository. Esta es la de "Predict Students' Dropout and Academic Success" [Realinho et al., 2021], y cuenta con 4424 instancias y 36 características. Su área de estudio son las ciencias sociales y está orientada a predecir si un estudiante se encuentra en una de tres categorías: desertor (o Dropout), matriculado (o Enrolled) o graduado (o Graduate).

2 Estrategia de Solución

Debido a que la variable objetivo es categórica, se utiliza un modelo de clasificación, y no de regresión, para realizar el machine learning.

En primer lugar, se prepara la base de datos. Para esto, se asegura que la columna "Target", que contiene los datos a analizar, se mapee con un Ordinal Mapping, en el cual se asigna cero a Dropout, uno a Graduate y dos a Enrolled. Por este motivo, durante el presente informe, se refiere a Dropout como la clase 0, a Graduate como la clase 1 y a Enrolled como la clase 2. Luego, se verifica que no haya valores nulos ni atípicos, para seguir con el proceso de clasificación.

Los dos modelos de clasificación seleccionados fueron el DecisionTreeClassifier y el RandomForest, y son estos los que van a ser expuestos en el informe.

2.1. DecisionTreeClassifier

Este modelo de clasificación se basa en un árbol de decisión que contiene nodos y hojas, los cuales representan decisiones y clases, respectivamente. Para comenzar, se realizó una búsqueda exhaustiva con GridSearchCV para encontrar los mejores hiperparámetros para el modelo. En la Tabla 1 se muestran los seleccionados.

Posteriormente, se evaluó el modelo con los hiperparámetros seleccionados, y se analizó con su reporte de clasificación, su matriz de confusión, sus curvas de aprendizaje y sus

Hiperparámetro	Selección
criterion	Gini
max_depth	5
max_features	None
min_samples_leaf	2
min_samples_split	10

Tabla 1: Hiperparámetros seleccionados para DecisionTreeClassifier.

curvas ROC. Estos datos son mostrados con claridad en la Tabla 2, en la Fig. 1, en la Fig. 2 y en la Fig. 3, respectivamente.

En primer lugar, para el reporte de clasificación, se tomaron los datos mostrados en la Tabla 2, los cuales indican un buen rendimiento en la clase 1, reflejado en una alta sensibilidad, así como una falla de clasificación de la clase 2, reflejado en una baja precisión y recall.

	precision	recall	f1-score	support
0	0.88	0.65	0.75	316
1	0.72	0.95	0.82	418
2	0.46	0.31	0.37	151
accuracy			0.74	885
macro avg	0.69	0.64	0.65	885
weighted avg	0.74	0.74	0.72	885

Tabla 2: Reporte de Clasificación para DecisionTreeClassifier.

En segundo lugar, la matriz de confusión es expuesta en la Fig. 1. Esta refuerza la correcta clasificación de la clase 1, con 398 clasficaciones correctas y solo cuatro falsos negativos. Sin embargo, reitera la necesidad de visualizar las clases 0 y 2: la clase 0 cuenta con 206 predicciones correctas, pero 70 y 40 son erradas hacia las clases 1 y 2, respectivamente; mientras que la clase 2 tiene 47 clasificaciones correctas, pero 23 errores hacia la clase 0 y 81 hacia la clase 1.

En tercer lugar, se muestran las curvas de aprendizaje en la Fig. 2, las cuales indican un buen ajuste a los datos en entrenamiento y en validación. No obstannte, el modelo es un poco más preciso en entrenamiento que en validación, lo que demuestra un ligero overfitting en el mismo.

Por último, en la Fig. 3 se exponen las curvas ROC para cada clase. Debido a su cercanía con la esquina superior izquierda de la gráfica, las curvas de las clases 0 y 1 demuestran

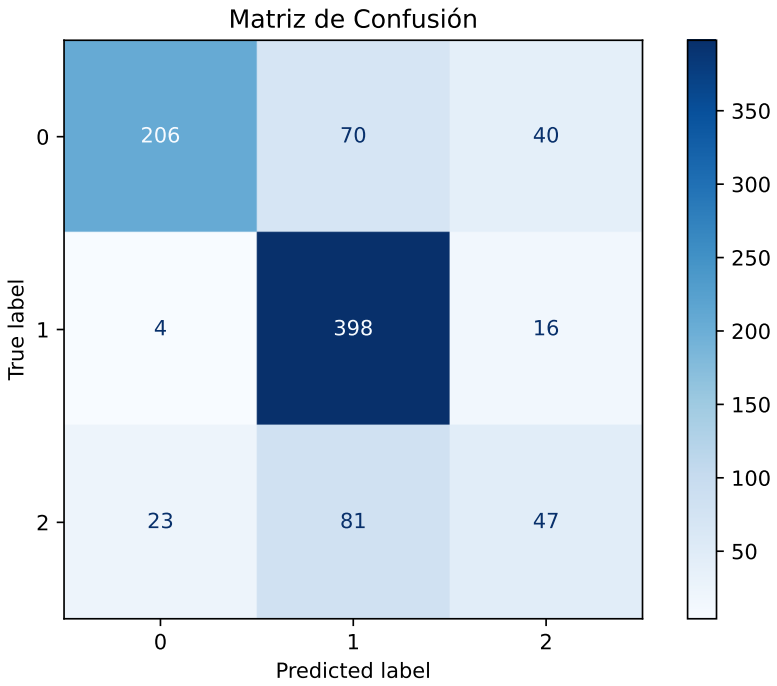

Figura 1: Matriz de confusión para el DecisionTreeClassifier.

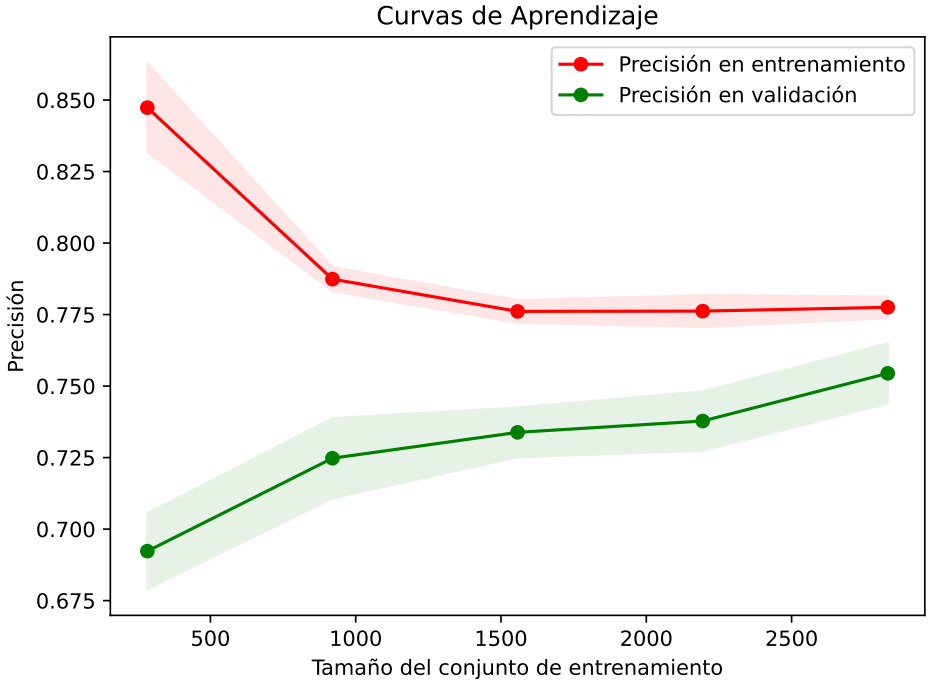

Figura 2: Curvas de aprendizaje para el DecisionTreeClassifier.

un buen desempeño. Sin embargo, al estar la clase 2 tan cerca de la línea de adivinanza aleatoria, se observa un rendimiento limitado para esta clase.

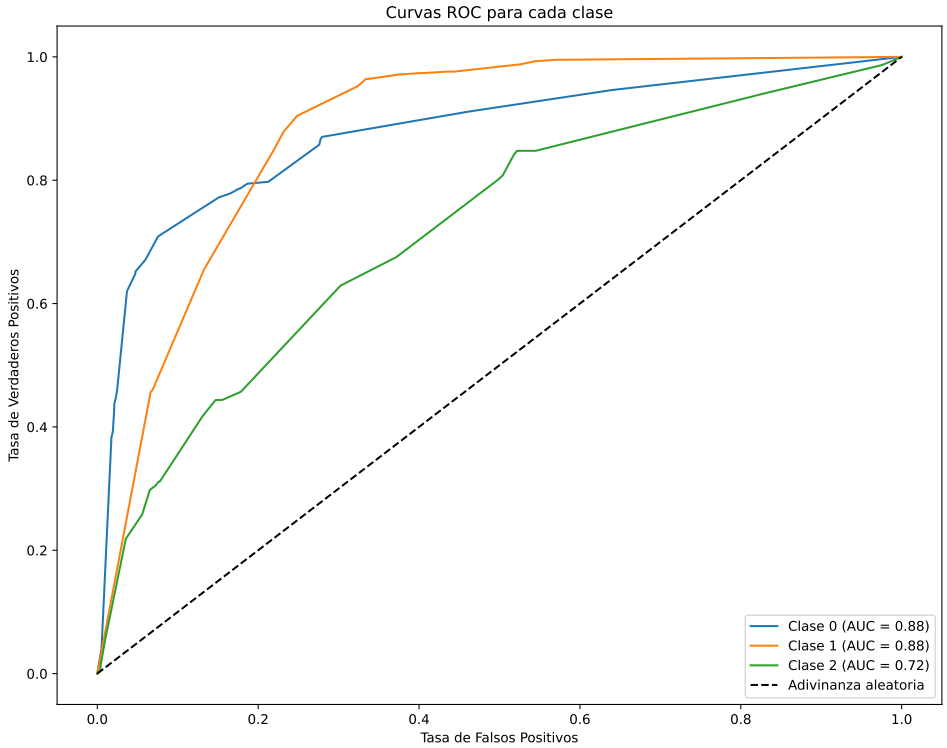

Figura 3: Curvas ROC para el DecisionTreeClassifier.

2.2. RandomForest

El modelo de clasificación de RandomForest funciona como un "bosque"de árboles de decisión entrenados aleatoriamente, tomando una decisión basada en el voto mayoritario de todos los árboles. Por esto, puede verse como una ."extensión"del algoritmo de árboles de decisión. En paralelo a lo aplicado en la Sec. 2.1, se comenzó obteniendo los mejores hiperparámetros para trabajar. Estos se exponen en la Tabla 3.

Hiperparámetro	**Selección**
max_depth	None
max_features	sqrt
min_samples_leaf	2
min_samples_split	5
n_estimators	200
oob_score	True

Tabla 3: Hiperparámetros seleccionados para RandomForest.

El reporte de clasificación mostrado en la Tabla 4 muestra una excelente precisión y recall para las clases 0 y 1, en comparación con el DecisionTreeClassifier. Sin embargo, son claras las dificultades persistentes en la clase 2. La precisión general, por otro lado,

es 3 % mejor (0.77 vs. 0.74) que la del modelo anterior.

	precision	**recall**	**f1-score**	**support**
0	0.85	0.77	0.81	316
1	0.77	0.94	0.85	418
2	0.46	0.31	0.39	151
accuracy			0.77	885
macro avg	0.72	0.67	0.68	885
weighted avg	0.76	0.77	0.75	885

Tabla 4: Reporte de Clasificación para RandomForest.

Asimismo, la matriz de confusión expuesta en la Fig. 4 resalta el buen desempeño en la clase 1 (393 aciertos con solo siete falsos negativos) al unísono que exhibe los grandes errores (51 y 22 y 37 y 67 para las otras clases, respectivamente) que posee el modelo en las clases 0 y 2, con 243 y 47 aciertos, respectivamente. Estos son mayormente notorios en la clase 2, en los que los aciertos son más bajos y las confusiones mayores.

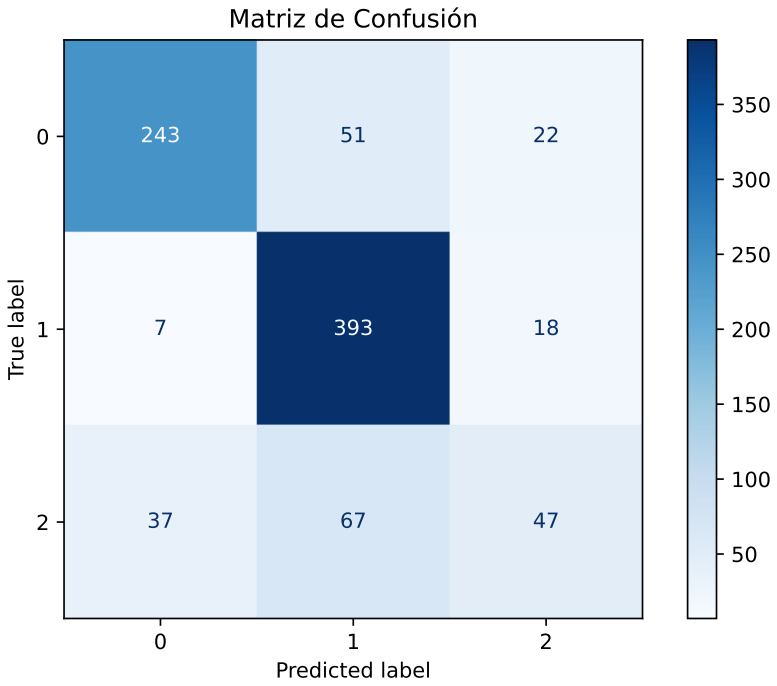

Figura 4: Matriz de confusión para el RandomForest.

La Fig. 5 expone las curvas de aprendizaje del modelo. En estas se percibe una convergencia hacia un rango mayor de valores de las dos curvas; la de entrenamiento, cerca del 95 %, indica un ajuste muy alto, mientras que la de validación está por el 77 %. Esto sugiere un ligero overfitting; no obstante, este modelo cuenta con mejor generalización que el descrito en la sección anterior.

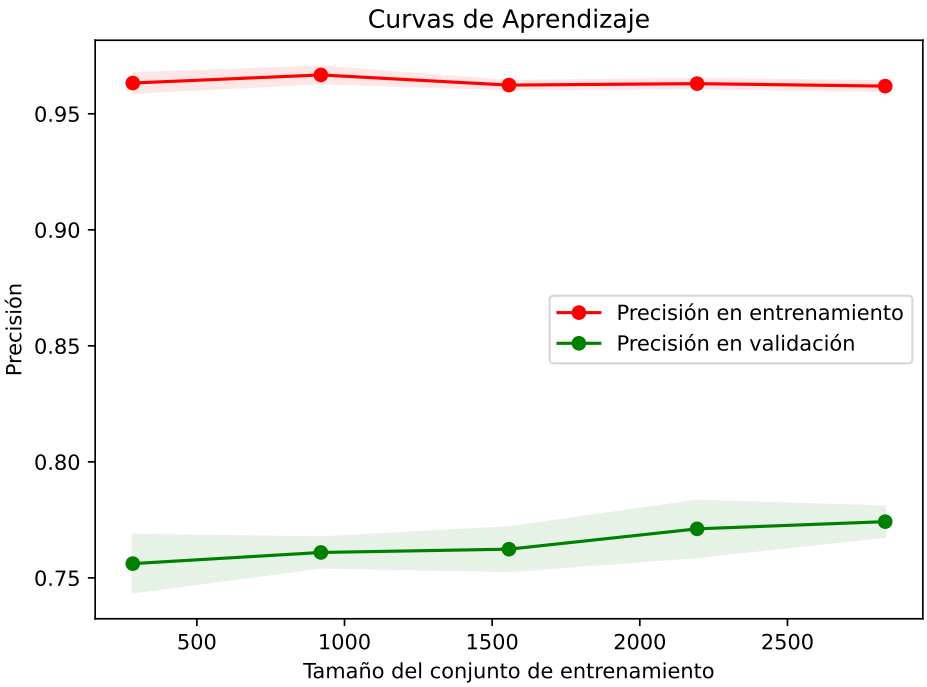

Figura 5: Curvas de aprendizaje para el RandomForest.

Para finalizar, las curvas ROC vistas en la Fig. 6 muestran una mayor capacidad de distinción entre las clases, comparado al DecisionTreeClassifier. Especialmente, la clase 2 muestra un mejor y notable desempeño en este modelo. Igualmente, el AUC es mayor para todas las clases, lo que es buena señal.

3 Estrategia de Comparación

Para la estrategia de comparación utilizada, se llevó a cabo un análisis de concordancia acorde a los modelos obtenidos. Este cuenta con una matriz de confusión comparativa, con el cálculo de la Kappa de Cohen para los modelos y con unas curvas ROC comparativas.

La matriz de confusión comparativa expuesta en la Fig. 7 muestra las dos matrices expuestas en la Sec. 2.1 y en la Sec. 2.2 en una misma gráfica. De estas se visualiza claramente que el RandomForest tiene mayores aciertos para la clase 0 (243 vs. 206) y para la Clase 1 (393 vs. 398). No obstante, los dos modelos presentan inconvenientes al clasificar para la clase 2 (47 aciertos).

La Kappa de Cohen es una métrica que mide el grado de acuerdo entre dos modelos

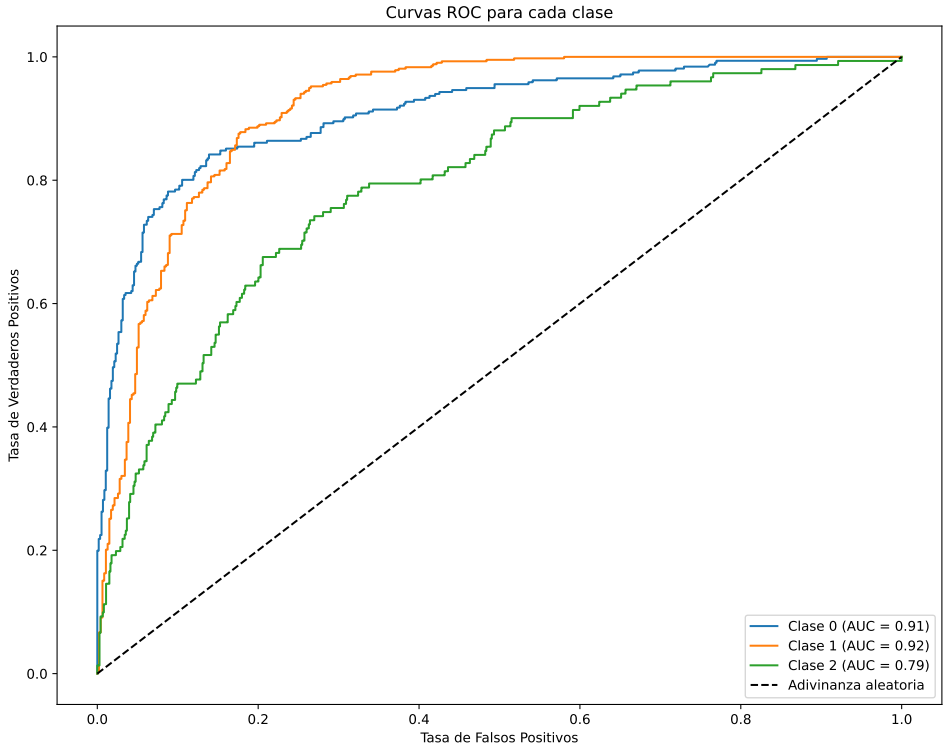

Figura 6: Curvas ROC para el RandomForest.

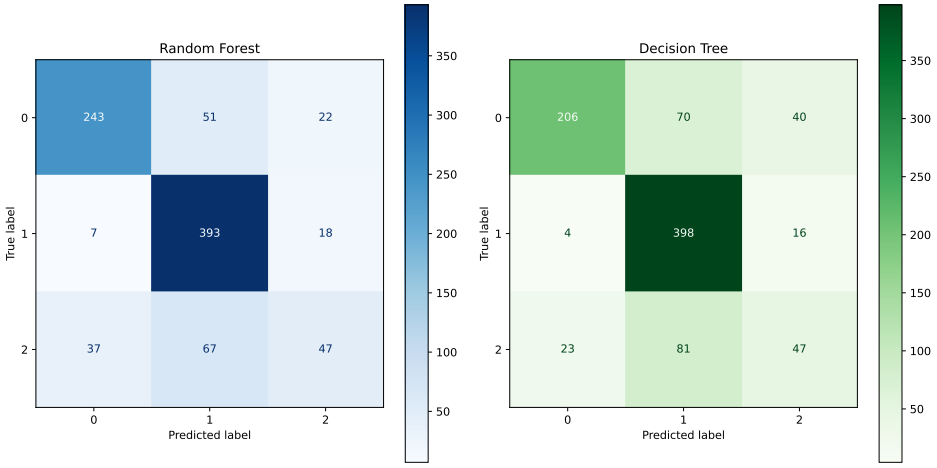

Figura 7: Matriz de confusión comparativa.

seleccionados, ajustándolo por azar. Este tiene la fórmula presentada en (1).

$$\kappa = \frac{P_o - P_e}{1 - P_e} \tag{1}$$

Donde P_o es la proporción de acuerdo observado entre los modelos y P_e es la proporción de acuerdo esperado por azar. Las Kappas de Cohen para los modelos utilizados se exponen

en la Tabla 5. Los cálculos sugieren que el DecisionTreeClassifier posee un valor de kappa aceptable, mientras que el RandomForest es fuerte en esta métrica. Consecuentemente, el segundo modelo muestra mejor concordancia general con las clasificaciones reales.

	DecisionTreeClassifier	**RandomForest**
Cohen's Kappa	0.55	0.62

Tabla 5: Kappa de Cohen para los modelos.

Por último, las curvas ROC comparativas, aunque ya visualizadas en la Sec. 2.1 y en la Sec. 2.2, se muestran en la Fig. 8. Estas demuestran una mejor discriminación de RandomForest en comparación con DecisionTreeClassifier.

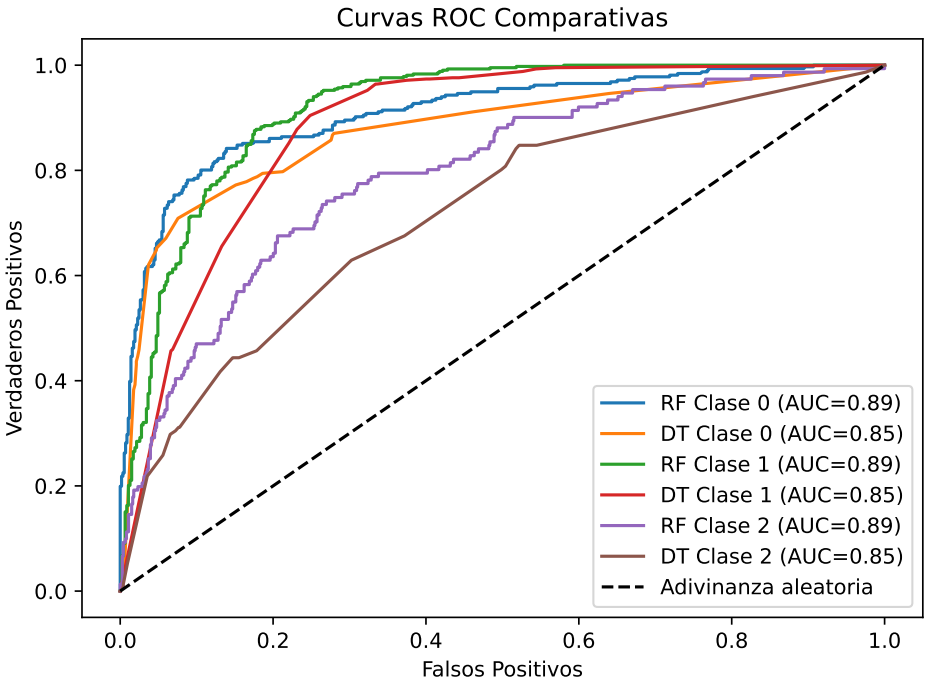

Figura 8: Curvas ROC comparativas.

4 Conclusiones

Aunque el modelo con mejor rendimiento obtuvo un muy buen puntaje en la generalidad de las métricas, es clave aclarar que los dos modelos tuvieron algunos problemas para clasificar la clase 2.

La simplicidad demostrada en el modelo del DecisionTreeClassifier, en conjunto con un

rendimiento decente, vuelven a este modelo una estrategia indicada para problemas simples, con menores requerimientos y que necesiten capacidad de interpretación únicamente.

El RandomForest demostró un mejor rendimiento general en todas las métricas análizadas. Figuras tales como las curvas de aprendizaje ejemplificaron una mejor capacidad de generalización para este modelo. Asimismo, el overfitting visto es menor al del DecisionTreeClassifier, y la Kappa de Cohen confirma una mayor concordancia.

En conclusón, el modelo con mejor rendimiento fue el RandomForest, y, por su robustez, es elegido para problemas en los que se priorice desempeño aantes que simplicidad.

Referencias

[Realinho et al., 2021] Realinho, V., Vieira, M., Machado, J., and Baptista, L. (2021). Predict students' dropout and academic success.