Universidad ORT Uruguay

Facultad de Ingeniería

Bernard Wand Polak

Ingeniería en Sistemas

Machine Learning para Sistemas Inteligentes

Obligatorio

Mateo Giraz (241195)

Pablo Durán (270956)

Santiago Villar (256345)

Grupo M6C

Docentes: Matías Carrasco

Agustina Disiot

Índice

Introducción

Este proyecto aborda la detección de objetos en imágenes, centrándose en la identificación de rostros. Utilizamos una variedad de modelos de aprendizaje automático, incluyendo técnicas de boosting, bagging y deep learning, mediante las bibliotecas de Python scikit-learn y Keras. El enfoque no se limita a probar únicamente el modelo más eficiente; en cambio, comparamos el rendimiento de varios modelos frente a imágenes reales para evaluar su efectividad y precisión. Este enfoque nos permite no solo identificar el modelo óptimo, sino también comprender las fortalezas y limitaciones de cada enfoque en la práctica, proporcionando una visión detallada y comparativa en el campo de la detección de rostros.

Imágenes y Features

Preprocesamiento

En nuestro proyecto de reconocimiento facial, hemos adoptado un enfoque detallado para el preprocesamiento de los datos, un paso crucial para asegurar la efectividad de nuestros modelos.

El proceso comienza con la carga del conjunto de datos Labeled Faces in the Wild (LFW), el cual es es ampliamente reconocido en el ámbito del procesamiento de imágenes, ya que proporciona una diversa colección de imágenes de rostros, capturadas bajo diferentes condiciones de iluminación y desde múltiples ángulos.Una vez cargado el conjunto de datos, procedemos a dividirlo en subconjuntos de entrenamiento y prueba para evaluar el rendimiento del modelo en datos no vistos, asegurando así que el modelo no solo memorice los datos de entrenamiento. La elección de reservar un 10% del conjunto de datos para pruebas, definida por test_size=0.1, proporciona una muestra suficientemente representativa para la evaluación, mientras que el 90% restante se utiliza para el entrenamiento. El uso del parámetro random_state=42 nos garantiza que los resultados sean reproducibles, un aspecto importante para la validación experimental y la comparación de resultados.

Además de los rostros, incorporamos imágenes de fondo al conjunto de datos para proporcionar ejemplos negativos durante el entrenamiento del modelo. Esto mejora notablemente la capacidad del modelo para distinguir entre imágenes que contienen rostros y las que no. Para ello, seleccionamos diversas imágenes estándar de la biblioteca sklearn, como 'camera', 'text', 'coins', entre otras. Estas imágenes, que varían desde objetos cotidianos hasta escenas astronómicas, aportan una gran variedad de texturas y patrones. Convertimos las imágenes en color (RGB) a escala de grises utilizando la función color.rgb2gray, lo que reduce la complejidad computacional y es coherente con el procesamiento de las imágenes de rostros, dado que la información de color a menudo no es esencial para el reconocimiento facial. A su vez, también agregamos 31 imágenes personalizadas, también convertidas a escala de grises. Estas imágenes adicionales aumentan la diversidad del conjunto de datos con escenarios y contextos únicos, lo que mejora la capacidad del modelo para generalizar a partir de ejemplos negativos variados
Este enriquecimiento del conjunto de datos con una amplia gama de imágenes no faciales es una estrategia clave que nos ayuda a minimizar los falsos positivos, un desafío común en el reconocimiento facial.

Extracción de Features

Esta diversidad en el preprocesamiento de imágenes nos lleva a la siguiente fase crucial de nuestro proyecto: el tratamiento avanzado de las imágenes y la extracción de features utilizando el método Histogram of Oriented Gradients (HOG). Este enfoque es esencial para transformar las imágenes en un formato adecuado para el procesamiento por parte de nuestros algoritmos. Definimos parámetros clave para las HOG y los datos que utilizaremos en el experimento, incluyendo la resolución de las imágenes, las escalas a considerar y las proporciones de entrenamiento y prueba. Al establecer un conjunto de escalas - [0.5,1,2,4,8] - nos aseguramos de capturar características de los rostros a diferentes niveles de detalle. La proporción de parches para entrenamiento y prueba, calculada en función del tamaño del conjunto de datos y la cantidad de imágenes de fondo, garantiza un balance adecuado para una evaluación exhaustiva y efectiva. Luego seguimos con la definición de parámetros como el número de orientaciones, píxeles por celda y celdas por bloque. Estos parámetros son esenciales para determinar cómo se capturan y se cuantifican las orientaciones de los gradientes en la imagen. Optamos por 3 orientaciones y una configuración de (8, 8) píxeles por celda y (3, 3) celdas por bloque.

Acto seguido, procedemos al preprocesamiento de las imágenes de rostros para los conjuntos de entrenamiento y de prueba. Redimensionamos las imágenes de los rostros (positive_patches_train y positive_patches_test) de acuerdo con la resolución definida previamente para estandarizar el tamaño de las imágenes antes de la extracción de características, permitiéndonos un análisis coherente y comparativo. En paralelo, para las imágenes de fondo (negative_patches_train y negative_patches_test), también hicimos una extracción de parches. Seleccionamos aleatoriamente secciones de estas imágenes a diferentes escalas, proporcionándonos ejemplos negativos en una variedad de contextos y escalas. Este proceso enriquece nuestro conjunto de datos y mejora significativamente la capacidad del modelo para diferenciar entre rostros y no rostros.

La etapa final de este proceso implica la construcción de la matriz de características y el vector de etiquetas para los conjuntos de entrenamiento y prueba. Aplicamos el método HOG a cada imagen, tanto positiva como negativa, para extraer características distintivas, aprovechando la eficacia del HOG para capturar información de bordes y gradientes. Estas características extraídas conforman nuestra matriz de características (X_train y X_test), mientras que el vector de etiquetas (y_train y y_test) se crea asignando 1 a las imágenes de rostros y 0 a las de fondo.

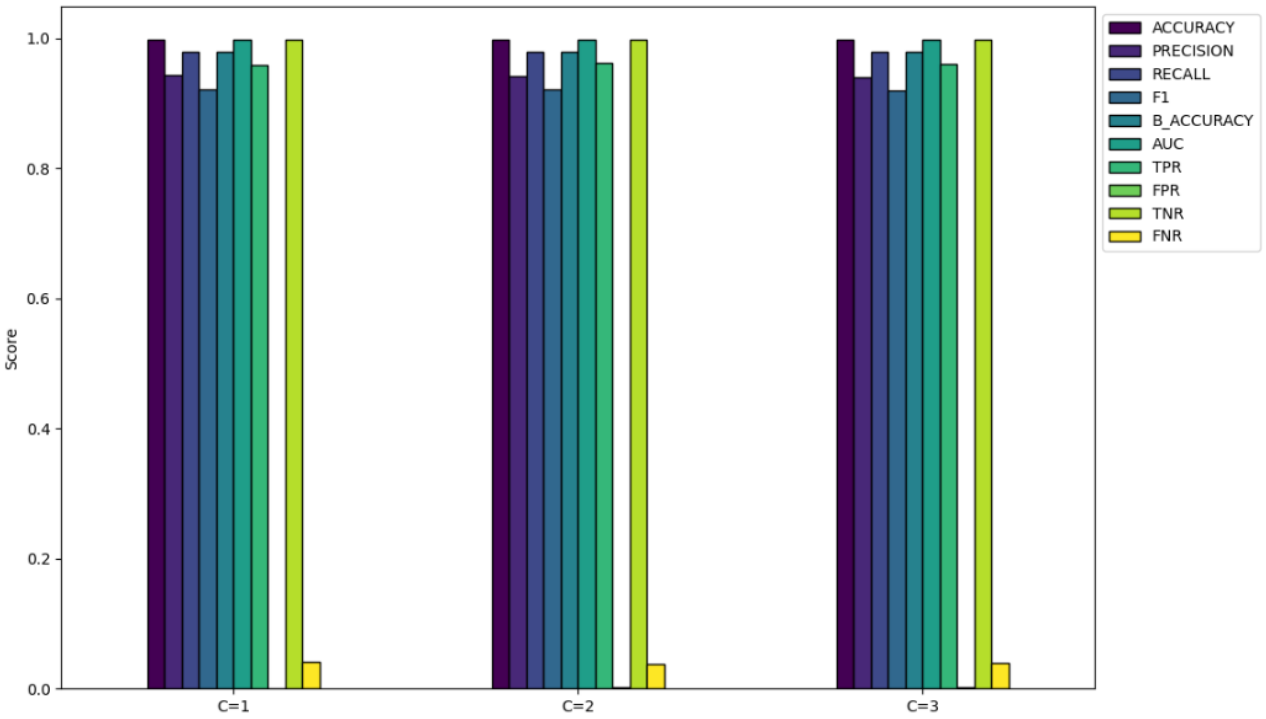
Este meticuloso proceso de preprocesamiento y extracción de características, desde la configuración inicial de los parámetros del HOG hasta la aplicación práctica en las imágenes, es un factor fundamental en nuestro proyecto. Al combinar un enfoque estratégico en la selección y tratamiento de las imágenes con una técnica de extracción de características sofisticada, hemos establecido una base sólida para el desarrollo de un modelo de reconocimiento facial preciso y confiable.

Evaluación de Modelos

Logistic Regression

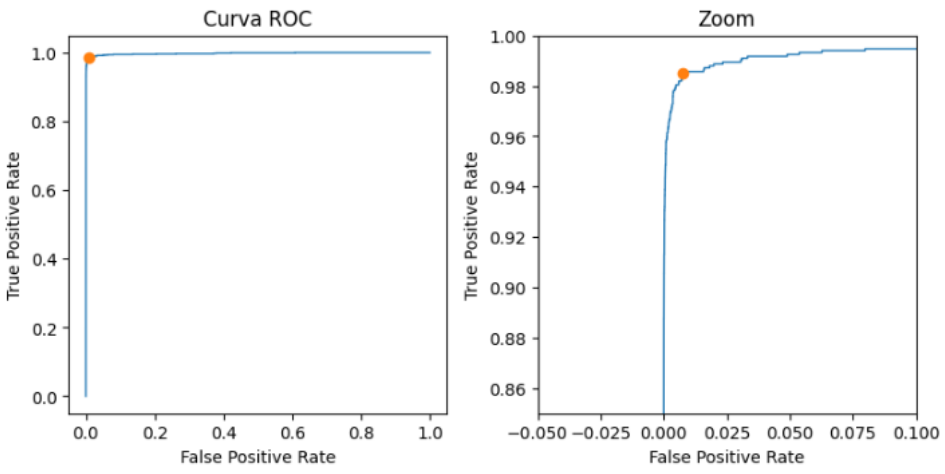
C=1 vs C=2 vs C=3

El primer modelo que entrenamos fue una regresión logística. Para comenzar a experimentar, decidimos evaluar las potenciales diferencias en el rendimiento que encontraríamos al variar el hiperparámetro C. Para poner el foco aquí, tomamos como base los hiperparámetros penalty='l1' y max_iter=1000.



Dado que no encontramos ningún beneficio notorio al incrementar el valor de C, decidimos quedarnos con el valor por defecto (C=1). A continuación se detallan los resultados de este experimento.

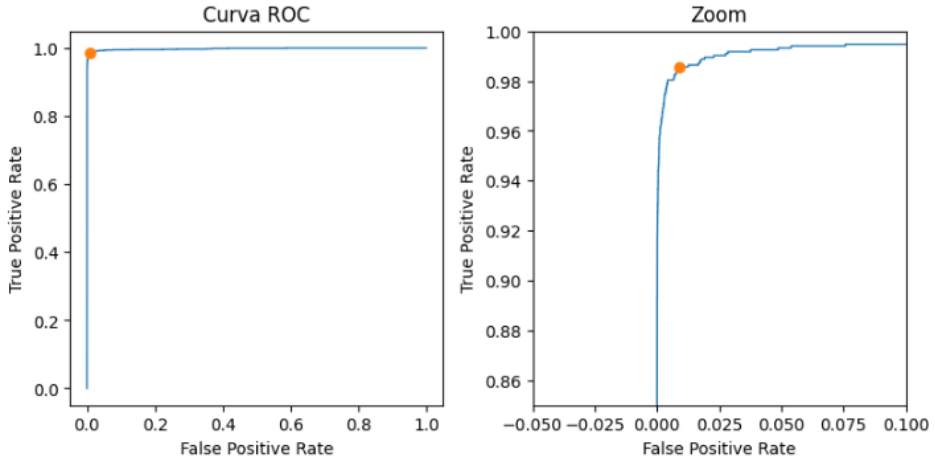
C=1	
ACCURACY	0.998287
PRECISION	0.942911
RECALL	0.978957
F1	0.921291
B_ACCURACY	0.978957
AUC	0.997745
TPR	0.959215
FPR	0.001300
TNR	0.998700
FNR	0.040785



Umbral óptimo: 0.1365
FPR: 0.0077, TPR: 0.9849

C=2

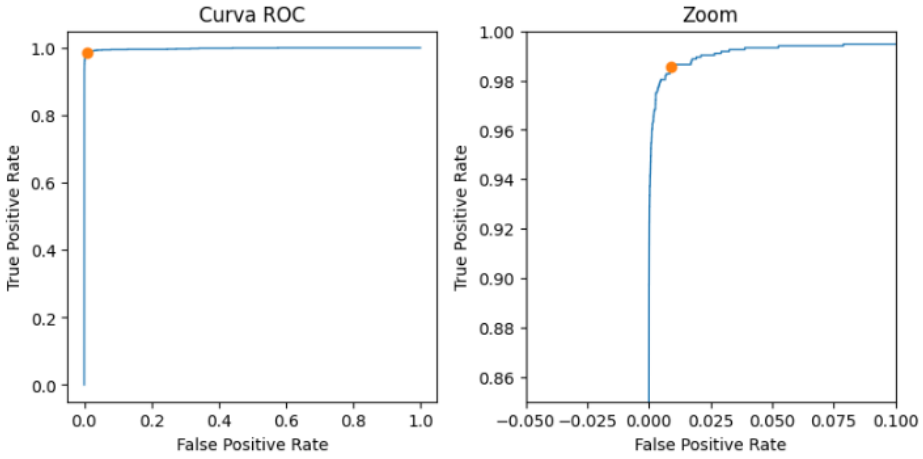
ACCURACY	0.998271
PRECISION	0.941504
RECALL	0.980070
F1	0.920796
B_ACCURACY	0.980070
AUC	0.997825
TPR	0.961480
FPR	0.001340
TNR	0.998660
FNR	0.038520




```
Umbral óptimo: 0.1083
FPR: 0.0091, TPR: 0.9856
```

C=3

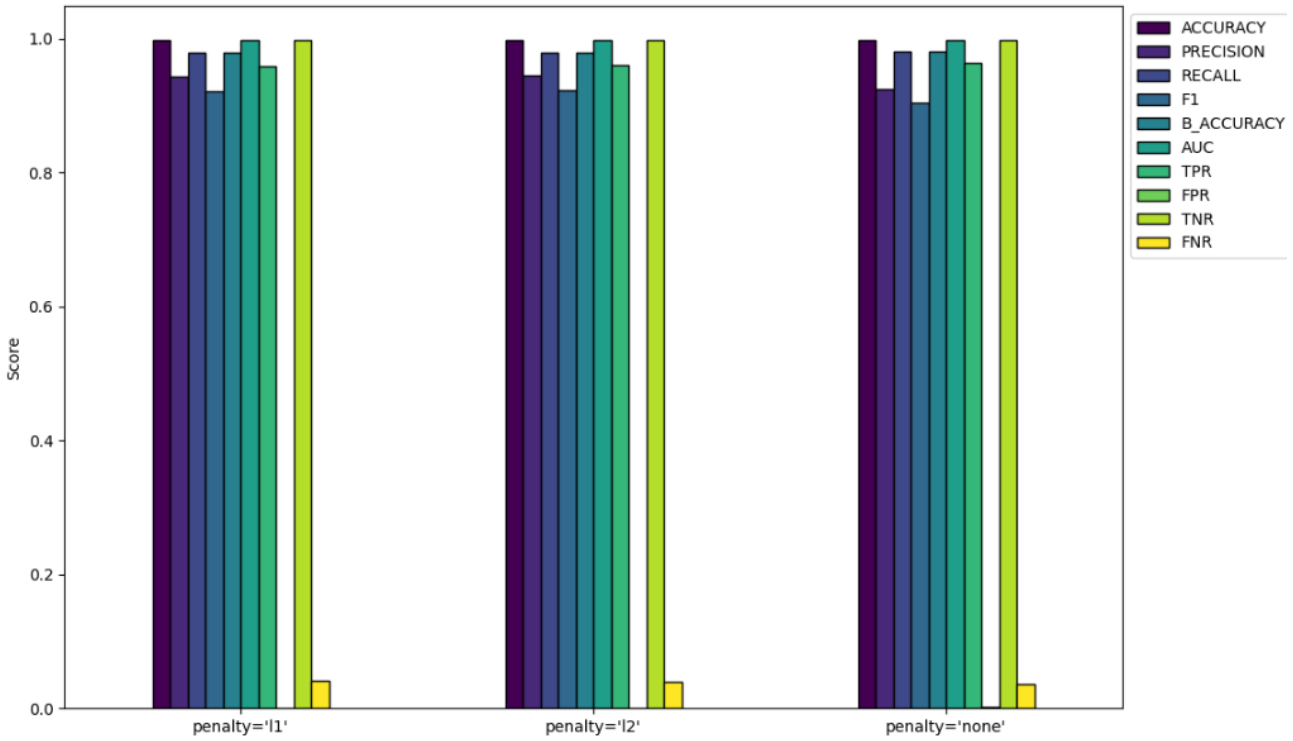
ACCURACY	0.998240
PRECISION	0.940541
RECALL	0.979680
F1	0.919407
B_ACCURACY	0.979680
AUC	0.997850
TPR	0.960725
FPR	0.001364
TNR	0.998636
FNR	0.039275




```
Umbral óptimo: 0.1096
FPR: 0.0089, TPR: 0.9856
```

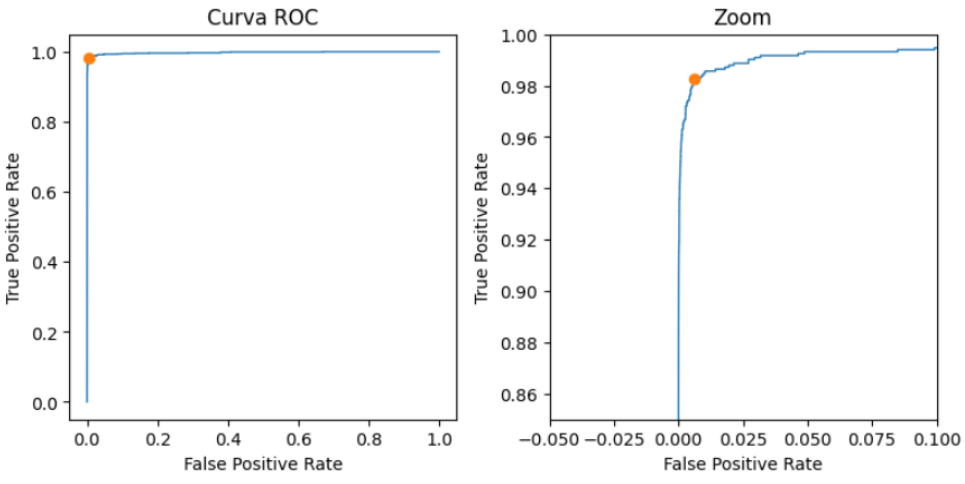
penalty='l1' vs penalty='l2' vs penalty='none'

Ahora que pudimos confirmar que el hiperparámetro C no afecta notablemente a nuestro experimento, vamos a variar los valores de penalty. Nos proponemos comparar el actual mejor experimento de Logistic Regression (C=1, max_iter=1000, penalty='l1', solver='liblinear') con dos nuevos experimentos con pentaly='l1' y pentaly='none'.



Ahora podemos evaluar, que aunque pequeñas, existen diferencias que marcarían primeramente, que los modelos que incluyen penalty son ligeramente más exactos (mayor accuracy) y precisos. Entre estos dos, el modelo entrenado con l2 (Ridge) muestra métricas levemente superiores. Dado esto y, que l2 es la configuración por defecto, decidimos que el experimento de Logistic Regression con hiperparámetros (C=1, max_iter=1000) es el mejor. A continuación se detallan las métricas obtenidas en estos dos nuevos experimentos.

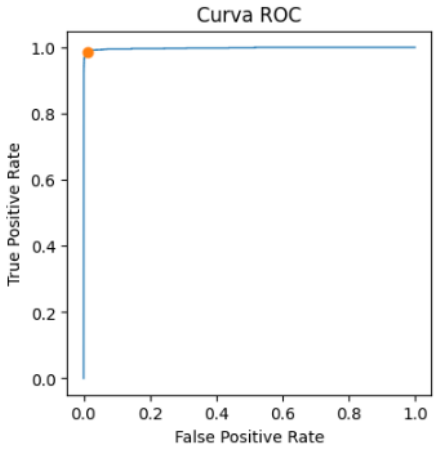
penalty='l2'	
ACCURACY	0.998350
PRECISION	0.945128
RECALL	0.979363
F1	0.924028
B_ACCURACY	0.979363
AUC	0.997673
TPR	0.959970
FPR	0.001244
TNR	0.998756
FNR	0.040030

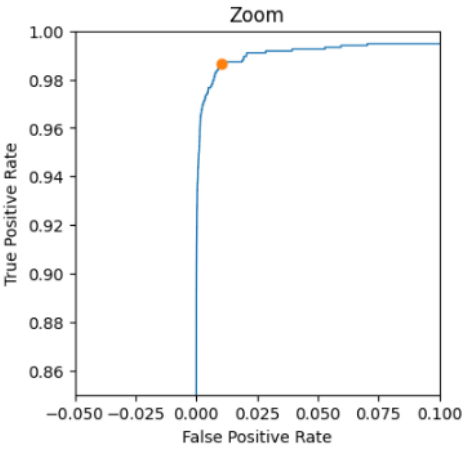



```
Umbral óptimo: 0.1654
FPR: 0.0061, TPR: 0.9826
```

penalty='none'

ACCURACY	0.997853
PRECISION	0.924912
RECALL	0.981353
F1	0.903751
B_ACCURACY	0.981353
AUC	0.997819
TPR	0.964502
FPR	0.001795
TNR	0.998205
FNR	0.035498

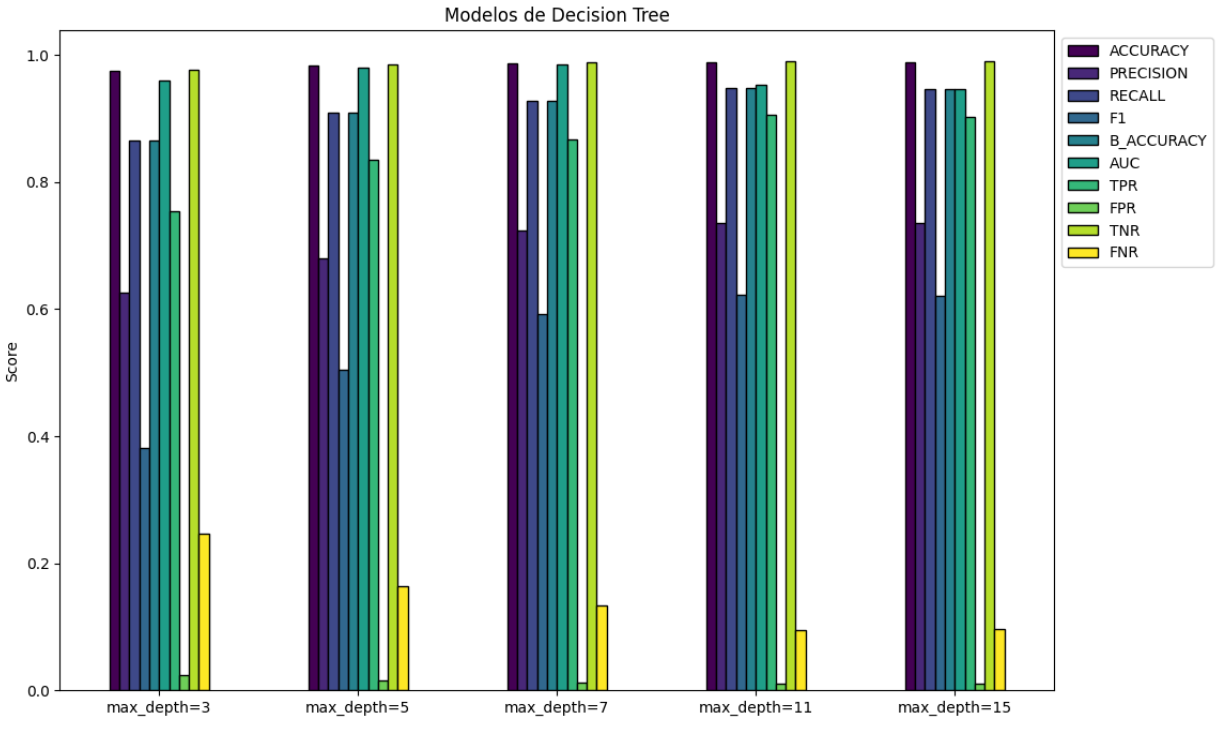





```
Umbral óptimo: 0.074
FPR: 0.0106, TPR: 0.9864
```

Decision Tree

Para el caso del árbol de decisión, decidimos hacer cinco pruebas con distintas alturas máximas para ver si encontrábamos diferencias notables en los resultados. Probamos con alturas máximas de tres, cinco, siete y once. Estos fueron los resultados que obtuvimos.

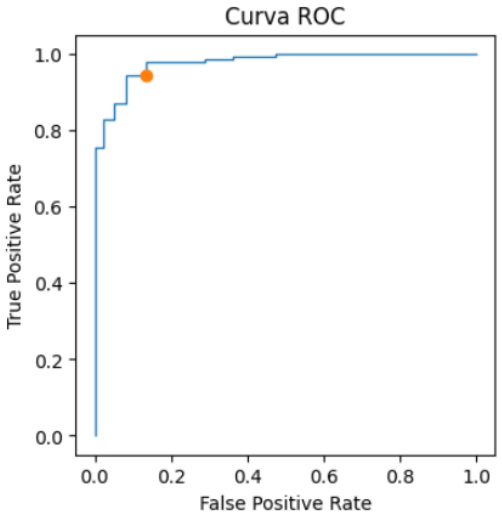


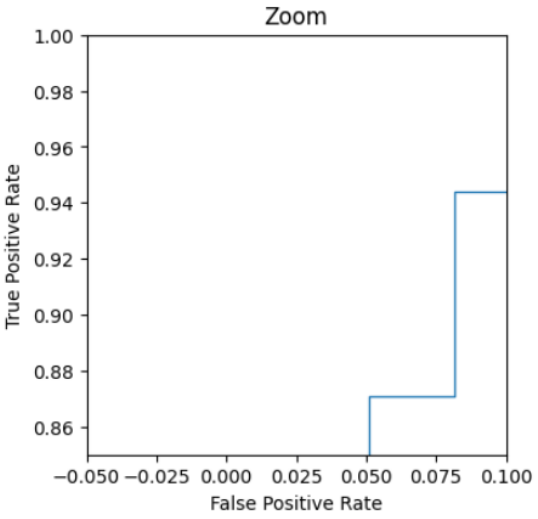
Se observa un ligero aumento en la métrica exactitud (accuracy) a medida que aumenta la profundidad máxima del árbol. De igual manera, la métrica precisión también mejora con la profundidad máxima del árbol, indicando que los modelos más complejos son más correctos a la hora de predecir identificaciones positivas. El Recall también muestra una mejora constante con el aumento de la profundidad máxima del árbol, lo que sugiere que los modelos son mejores identificando positivos reales a medida que se profundiza el árbol. El área bajo la curva ROC aumenta con la profundidad máxima del árbol, indicando un mejor rendimiento general del modelo en la clasificación.

max_depth=3

La precisión es relativamente baja, pero el recall es alto, lo que sugiere que el modelo clasifica correctamente la mayoría de las instancias positivas, pero también puede etiquetar algunas instancias negativas como positivas

max_depth=3	
ACCURACY	0.974495
PRECISION	0.626523
RECALL	0.865301
F1	0.381863
B_ACCURACY	0.865301
AUC	0.959974
TPR	0.753776
FPR	0.023174
TNR	0.976826
FNR	0.246224



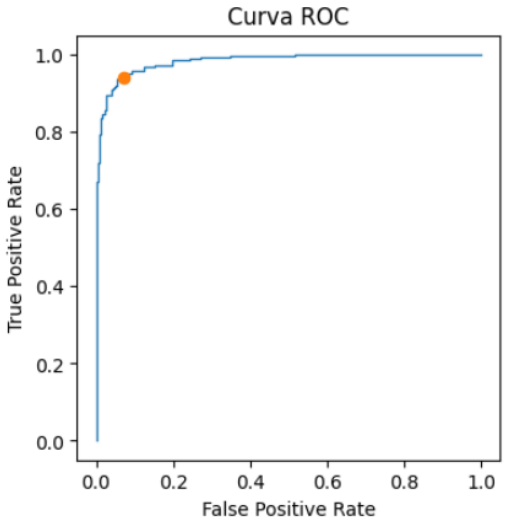


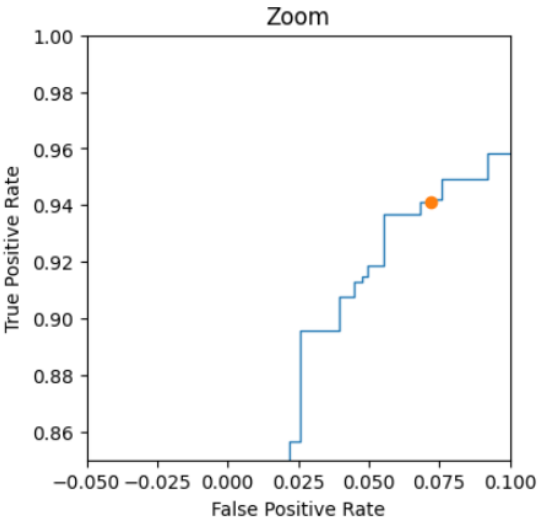

```
Umbral óptimo: 0.1056
FPR: 0.1354, TPR: 0.9441
```

max_depth=5

Presenta mejores valores que el modelo de profundidad 3 en prácticamente todas las métricas. Dado que la precisión y el recall son más altos, podemos esperar una mejora en la capacidad del modelo para clasificar correctamente tanto las instancias positivas como las negativas.

max_depth=5	
ACCURACY	0.982847
PRECISION	0.679778
RECALL	0.909876
F1	0.504447
B_ACCURACY	0.909876
AUC	0.980652
TPR	0.835347
FPR	0.015596
TNR	0.984404
FNR	0.164653



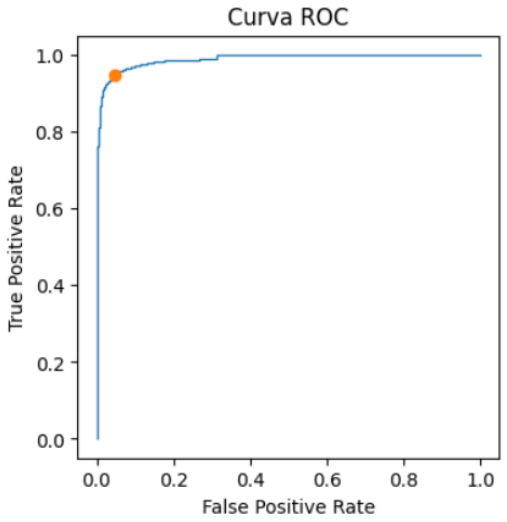


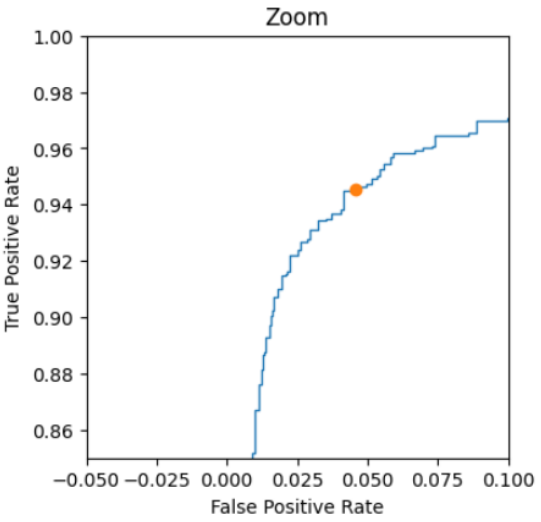

```
Umbral óptimo: 0.0938
FPR: 0.0722, TPR: 0.9411
```

max_depth=7

Continúa mejorando con respecto al modelo de profundidad 5. La precisión y el recall siguen mejorando, indicando un modelo más preciso y sensible.

max_depth=7	
ACCURACY	0.987528
PRECISION	0.724213
RECALL	0.927935
F1	0.592363
B_ACCURACY	0.927935
AUC	0.984948
TPR	0.867069
FPR	0.011200
TNR	0.988800
FNR	0.132931



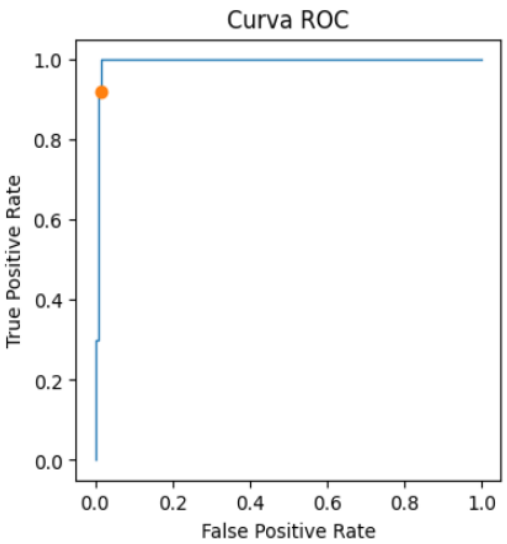


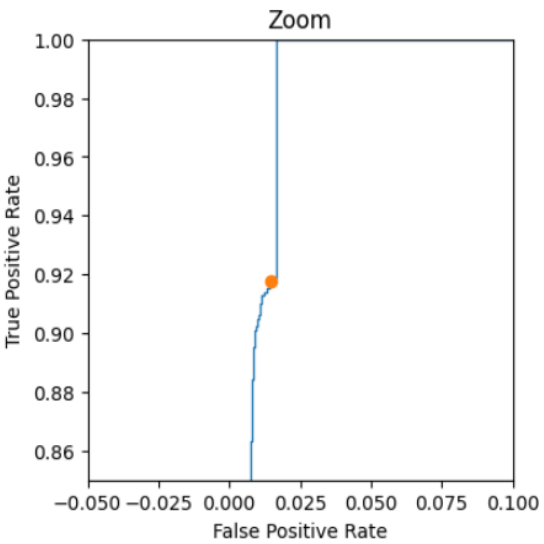

```
Umbral óptimo: 0.0962
FPR: 0.0459, TPR: 0.9456
```

max_depth=11

Este modelo tiene el mejor rendimiento entre los cinco. Comparado con los modelos previos, presenta alta precisión y recall, lo que sugiere la mejor capacidad para clasificar tanto las instancias positivas como las negativas.

max_depth=11	
ACCURACY	0.988491
PRECISION	0.736266
RECALL	0.947478
F1	0.621888
B_ACCURACY	0.947478
AUC	0.952905
TPR	0.905589
FPR	0.010634
TNR	0.989366
FNR	0.094411



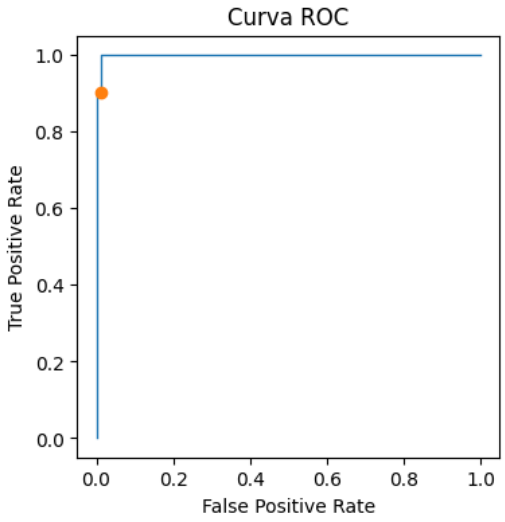


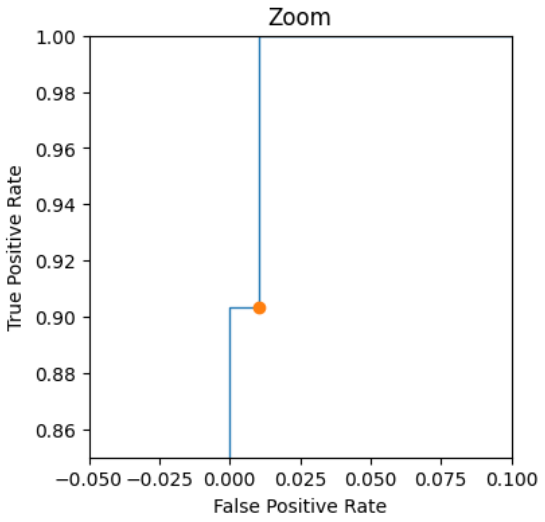

```
Umbral óptimo: 0.0676
FPR: 0.0149, TPR: 0.9177
```

max_depth=15

Este modelo tiene un rendimiento similar al anterior, ligeramente peor. Dado que la complejidad sigue aumentando y no vemos mejoría en las métricas, consideramos que el modelo anterior (altura máxima 11) es el mejor.

max_depth=15	
ACCURACY	0.988483
PRECISION	0.736129
RECALL	0.946353
F1	0.621137
B_ACCURACY	0.946353
AUC	0.946353
TPR	0.903323
FPR	0.010618
TNR	0.989382



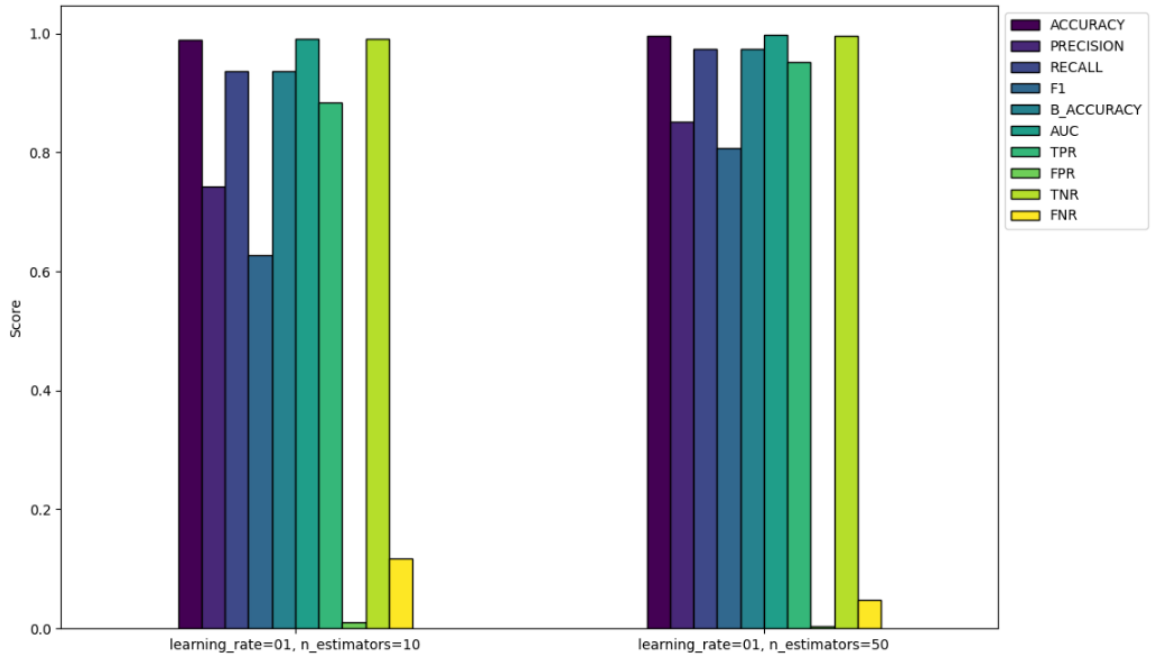



```
Umbral óptimo: 1.0
FPR: 0.0106, TPR: 0.9033
```

Adaptive Boosting

A la hora de entrenar modelos de Adaptive Boosting, decidimos hacer varias pruebas variando los hiperparámetros de learning_rate y n_estimators, a través de los cuáles fuimos desarrollando las siguientes conclusiones.

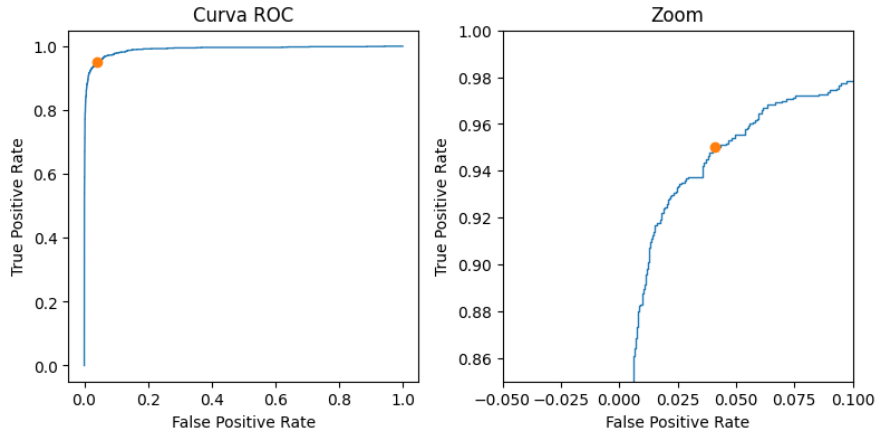
learning_rate=0.1, n_estimators=10 vs n_estimators=50



Para esta primera evaluación, decidimos considerar qué tan bueno sería un modelo con el hiperparámetro n_estimator más bajo que el default (n_estimator=50). En este caso vemos que un número mayor de estimadores es significativamente mejor (presenta mejores métricas a nivel de exactitud, precisión, recall, entre otras).

n_estimators=10

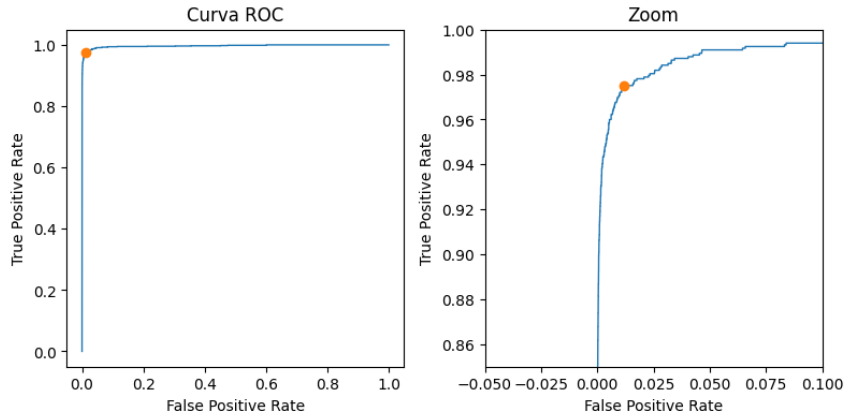
ACCURACY	0.989051
PRECISION	0.742817
RECALL	0.936551
F1	0.627651
B_ACCURACY	0.936551
AUC	0.989822
TPR	0.882931
FPR	0.009828
TNR	0.990172
FNR	0.117069




```
Umbral óptimo: 0.4476
FPR: 0.0412, TPR: 0.9502
```

n_estimators=50

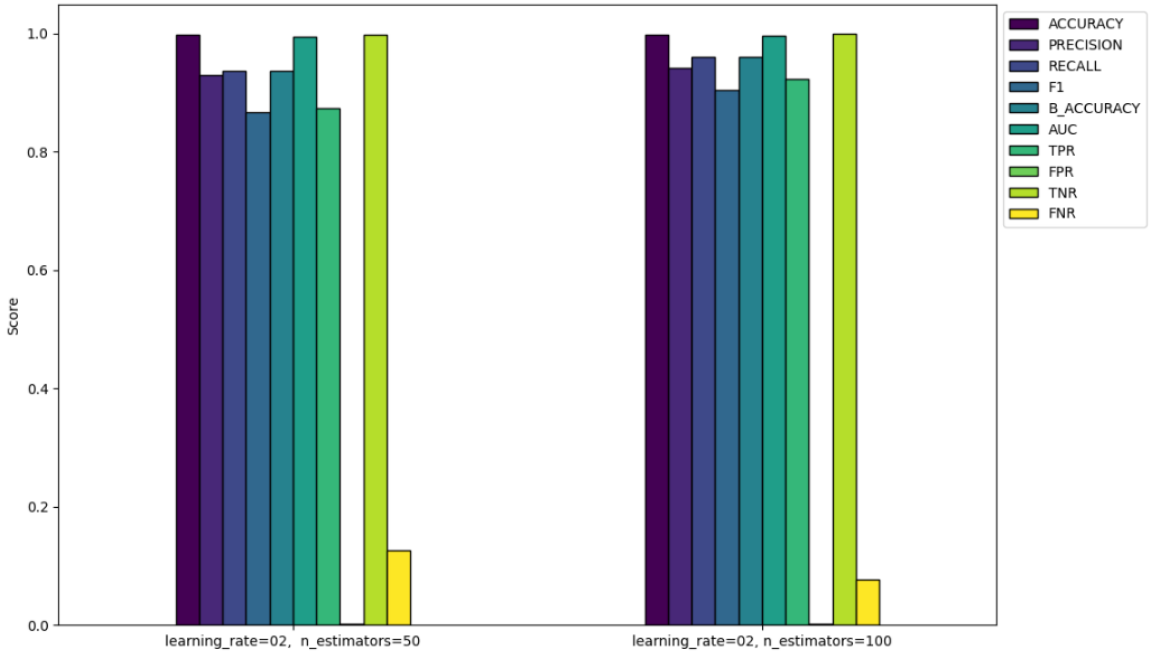
ACCURACY	0.995272
PRECISION	0.850719
RECALL	0.973697
F1	0.807951
B_ACCURACY	0.973697
AUC	0.996664
TPR	0.951662
FPR	0.004268
TNR	0.995732
FNR	0.048338



Umbral óptimo: 0.4905
FPR: 0.0119, TPR: 0.9751

learning_rate=0.2, n_estimators=50 vs n_estimators=100

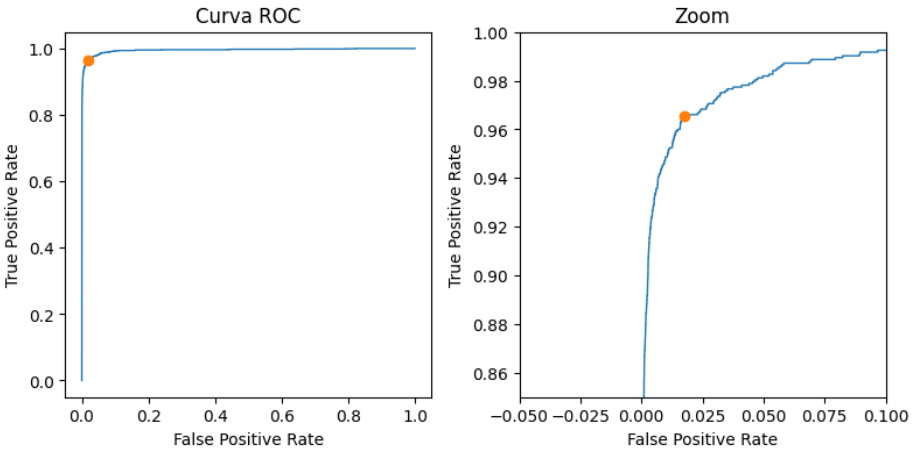
Ahora que confirmamos que el parámetro n_estimators otorga mejores resultados en 50 que en 10, nos interesa corroborar si la tendencia continúa cuando consideramos el valor de n_estimators 100. También evaluamos los resultados al aumentar learning_rate.



Podemos apreciar que aumentando el valor de learning_rate el modelo se vuelve más exacto y preciso. De igual forma, al duplicar n_estimators vemos una ligera mejoría a nivel de métricas, que si bien no es determinante, decidimos considerar como superior.

n_estimators=50

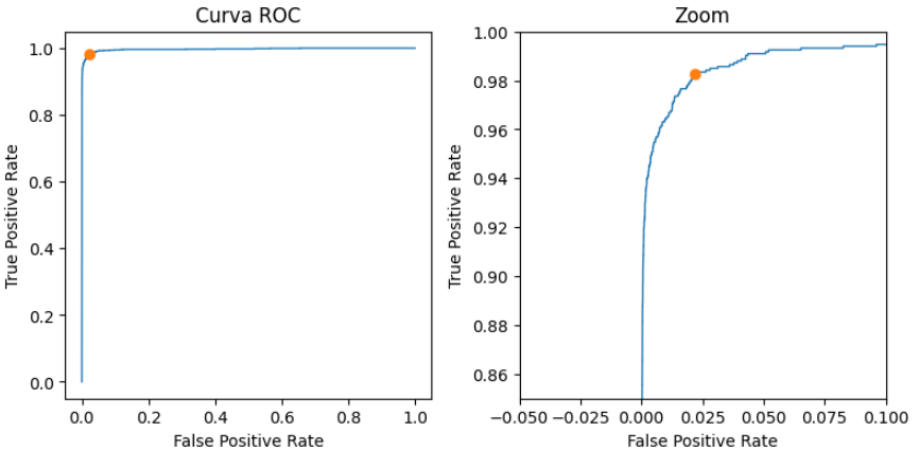
ACCURACY	0.997182
PRECISION	0.929070
RECALL	0.935802
F1	0.866242
B_ACCURACY	0.935802
AUC	0.995134
TPR	0.873112
FPR	0.001508
TNR	0.998492
FNR	0.126888




```
Umbral óptimo: 0.4477
FPR: 0.0174, TPR: 0.9653
```

n_estimators=100

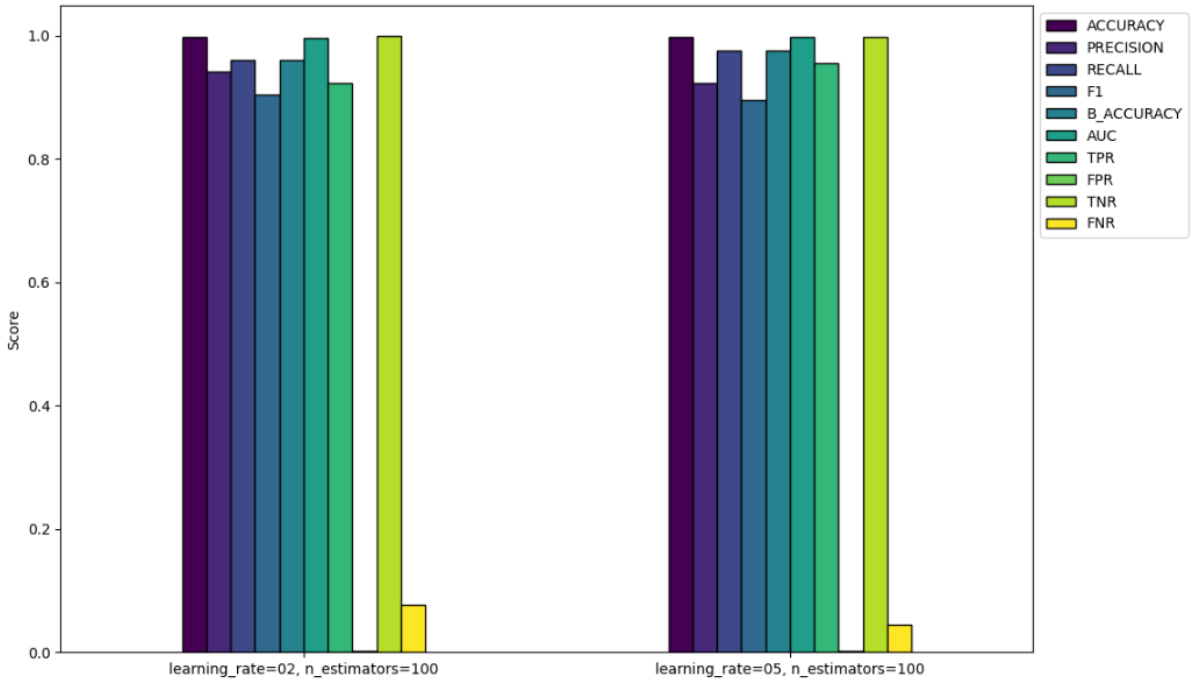
ACCURACY	0.997940
PRECISION	0.941752
RECALL	0.961220
F1	0.903583
B_ACCURACY	0.961220
AUC	0.996922
TPR	0.923716
FPR	0.001276
TNR	0.998724
FNR	0.076284




```
Umbral óptimo: 0.4647
FPR: 0.0218, TPR: 0.9826
```

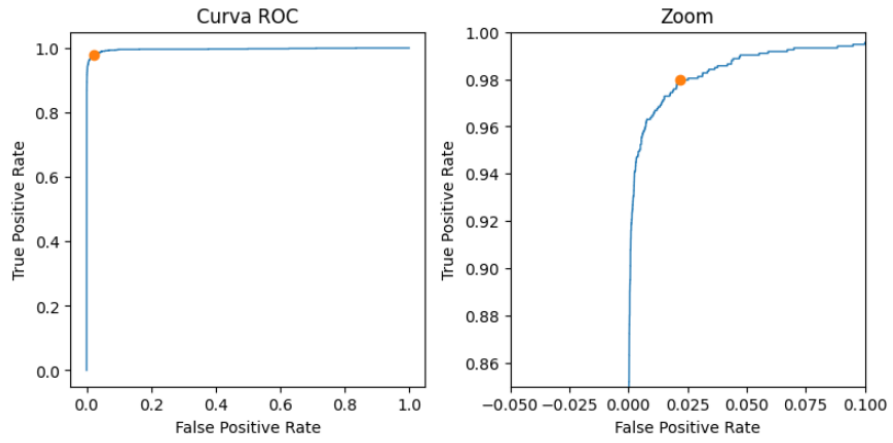
learning_rate=0.5, n_estimators=100

Teniendo en cuenta que elegimos n_estimators=100 como óptimo, ahora queremos saber si el valor que elegimos para learning_rate es correcto o debería ser mayor. Decidimos probar con learning_rate=0.5. Decidimos contrastar este nuevo modelo contra el mejor dentro de los de Adaptive Boosting, que en nuestro caso hasta ahora es learning_rate=0.2, n_estimators=100.



learning_rate=0.5

ACCURACY	0.997695
PRECISION	0.922220
RECALL	0.976416
F1	0.896454
B_ACCURACY	0.976416
AUC	0.997967
TPR	0.954683
FPR	0.001851
TNR	0.998149
FNR	0.045317



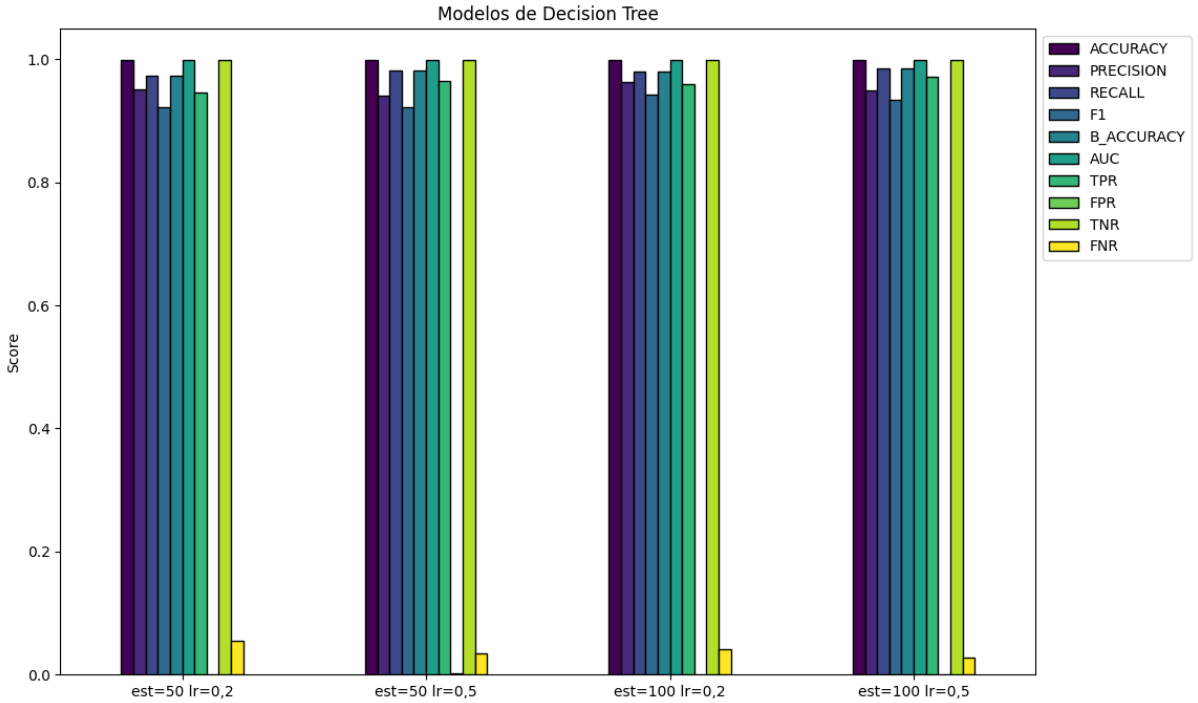
Umbral óptimo: 0.4719
FPR: 0.0217, TPR: 0.9796

Al comparar los resultados del nuevo modelo con el mejor modelo de Adaptive Boosting, vemos que el original, con learning_rate=0.2 es ligeramente superior a nivel de exactitud y precisión. Dado estas consideraciones, entendemos que la mejor combinación para Adaptive Boosting es learning_rate=0.2 y n_estimators=100

Gradient Boosting

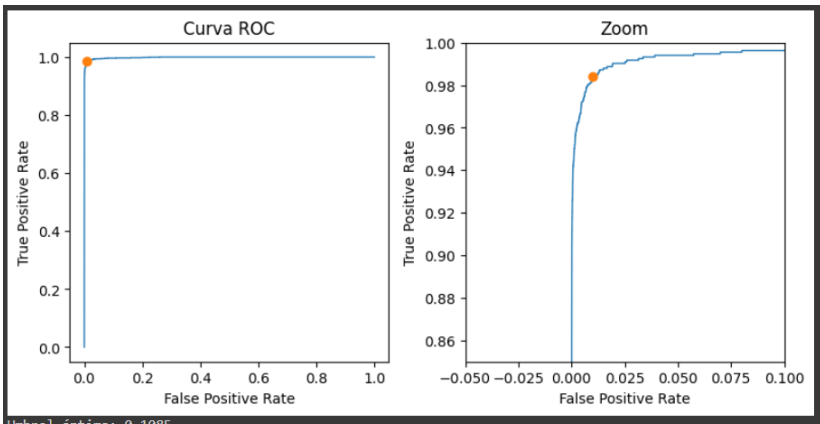
En el momento de comenzar a entrenar modelos con Gradient Boosting, decidimos realizar varios modelos variando la cantidad de estimadores, el valor del learning rate y para cada modelo decidimos entrenarlo con y sin early stopping para poder comparar el impacto de este regulador en nuestro modelos. Decidimos crear 4 modelos variando los estimadores entre 50 y 100 y los valores de learning rate entre 0.2 y 0.5 y finalmente para cada uno de estos entrenamos una variante usando Early Stopping y otra sin. El nivel de paciencia utilizado fue 5, el valor predeterminado.

Modelos sin Early Stopping



Modelo con 50 estimadores, 0.2 de learning rate

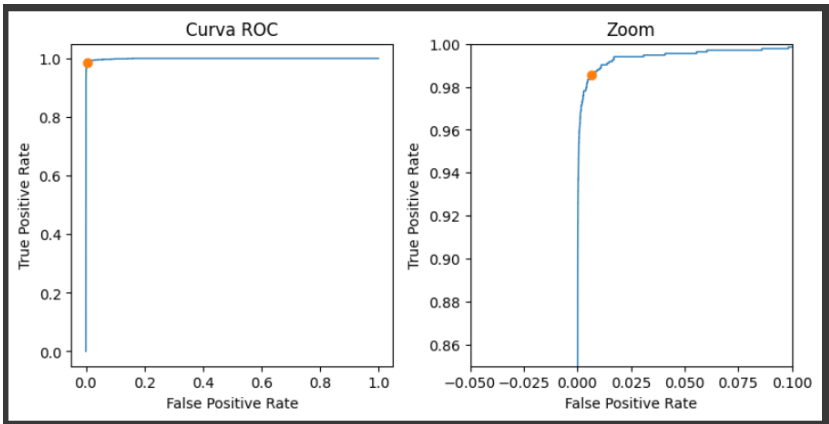
El primer modelo, configurado con 50 estimadores y una tasa de aprendizaje de 0.2, muestra una precisión y recall considerablemente altas. Su valor F1, que equilibra precisión y recall, es robusto, lo que sugiere un rendimiento bien balanceado. La métrica AUC, cercana a la perfección, indica una distinción excepcional entre las clases. Las tasas de verdaderos positivos y verdaderos negativos son altas, resaltando su habilidad para identificar correctamente ambas clases.



est=50 lr=0,2	
ACCURACY	0.998342
PRECISION	0.950072
RECALL	0.972259
F1	0.922623
B_ACCURACY	0.972259
AUC	0.998723
TPR	0.945619
FPR	0.001101
TNR	0.998899
FNR	0.054381

Modelo con 50 estimadores, 0.5 de learning rate

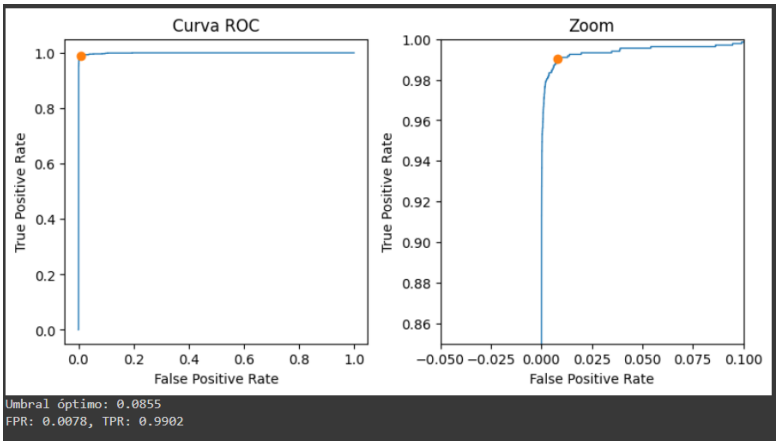
Al aumentar el número de estimadores a 100 y mantener la tasa de aprendizaje en 0.2, observamos una mejora en la precisión y el valor F1 en comparación con los modelos anteriores, lo que implica una mejoría en el balance entre precisión y recall. Este modelo también presenta la puntuación AUC más alta, lo que refleja una excelente capacidad de discriminación entre clases positivas y negativas.



est=50 lr=0,5	
ACCURACY	0.998287
PRECISION	0.940810
RECALL	0.981946
F1	0.921745
B_ACCURACY	0.981946
AUC	0.999196
TPR	0.965257
FPR	0.001364
TNR	0.998636
FNR	0.034743

Modelo con 100 estimadores, 0.2 de learning rate

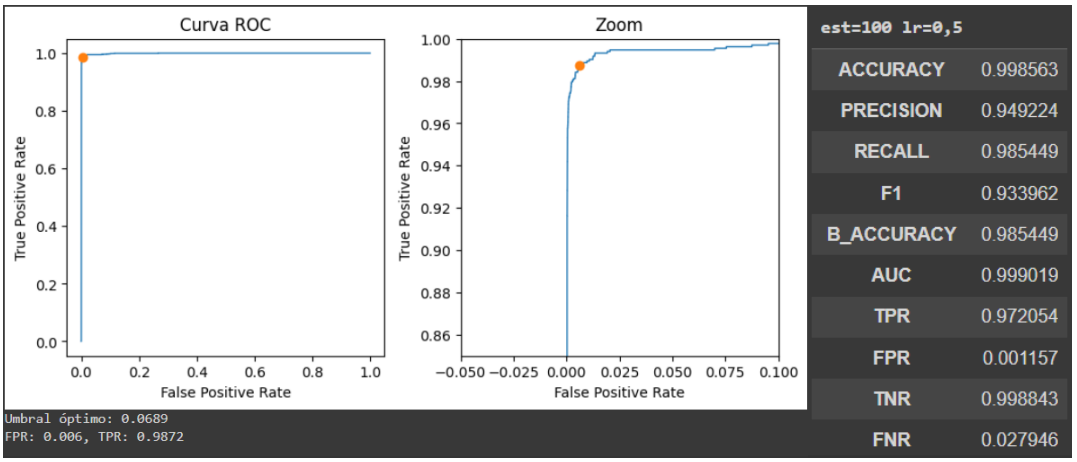
Al aumentar el número de estimadores a 100 y mantener la tasa de aprendizaje en 0.2, observamos una mejora en la precisión y el valor F1 en comparación con los modelos anteriores, lo que implica una mejoría en el balance entre precisión y recall.



est=100 lr=0,2	
ACCURACY	0.998776
PRECISION	0.962950
RECALL	0.979204
F1	0.942486
B_ACCURACY	0.979204
AUC	0.999224
TPR	0.959215
FPR	0.000806
TNR	0.999194
FNR	0.040785

Modelo con 100 estimadores, 0.5 de learning rate

Finalmente, el modelo con 100 estimadores y una tasa de aprendizaje de 0.5 destaca por su excepcional recall, el más alto entre los modelos examinados. Aunque su precisión no es la más alta, el valor F1 y la puntuación AUC son notables. El modelo muestra una habilidad sobresaliente para identificar correctamente los verdaderos positivos, aunque hay una leve incremento en la tasa de falsos positivos.

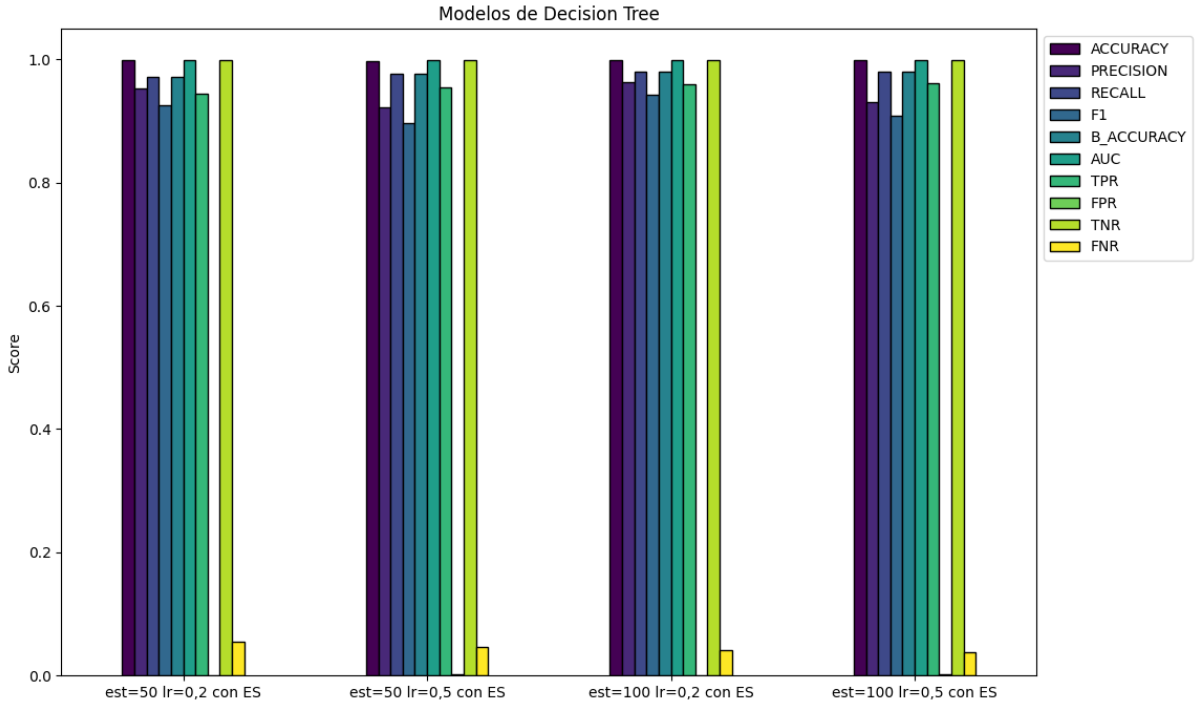


est=100 lr=0,5	
ACCURACY	0.998563
PRECISION	0.949224
RECALL	0.985449
F1	0.933962
B_ACCURACY	0.985449
AUC	0.999019
TPR	0.972054
FPR	0.001157
TNR	0.998843
FNR	0.027946

Observando las métricas, el modelo con 100 estimadores y una tasa de aprendizaje de 0.2 presenta el equilibrio más sólido. Tiene la segunda precisión más alta, un alto recall, el valor F1 más alto y la puntuación AUC más alta, lo que indica una excelente capacidad para distinguir entre las clases de forma correcta y equilibrada. Un valor F1 alto es particularmente deseable en la mayoría de las situaciones, ya que significa que el modelo tiene un buen equilibrio entre la precisión y el recall, siendo efectivo tanto en la identificación de los verdaderos positivos como en la minimización de los falsos positivos y falsos negativos.

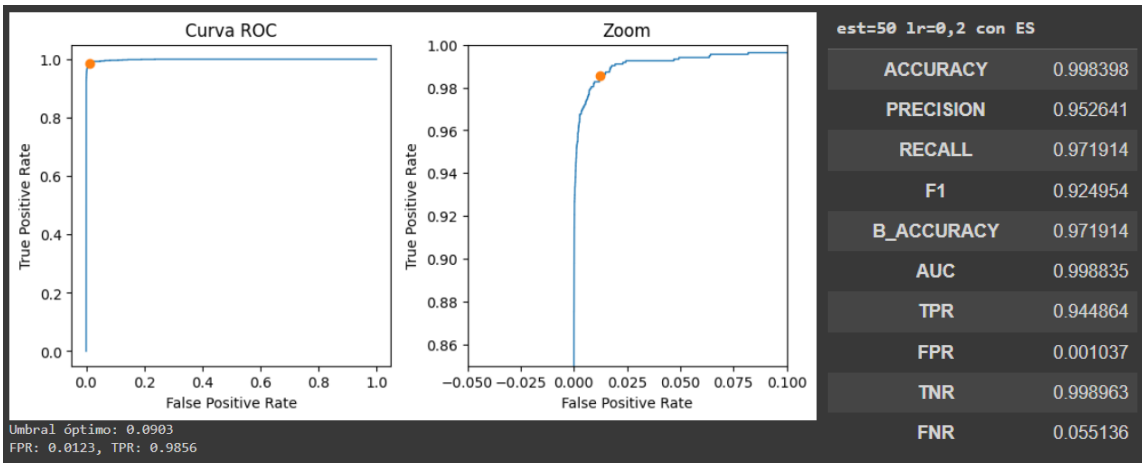
Modelos con Early Stopping

Al abordar el desafío de la clasificación mediante modelos de Gradient Boosting, es crucial evaluar meticulosamente una variedad de métricas para comprender la efectividad del modelo. Los cuatro modelos que hemos implementado con la técnica de Early Stopping difieren en dos parámetros esenciales: la cantidad de estimadores y la tasa de aprendizaje. A continuación, se presenta un análisis detallado de cada modelo, enfocado en su capacidad de predicción y la confiabilidad de sus resultados.



Modelo con 50 estimadores, 0.2 de learning rate y Early Stopping

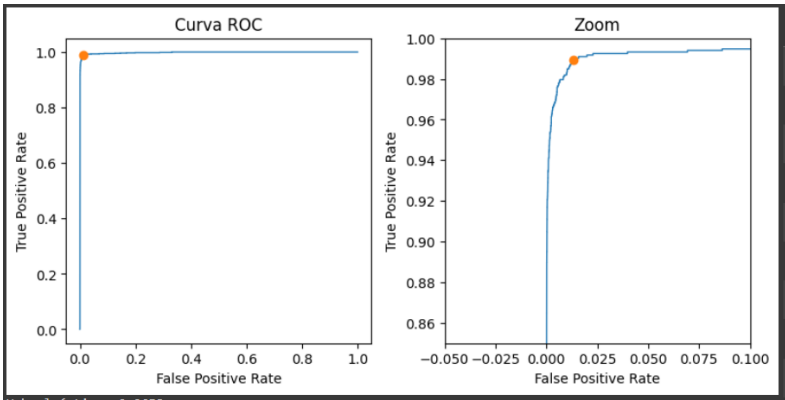
El primer modelo, con 50 estimadores y una tasa de aprendizaje de 0.2, se destaca por su alta precisión. Esto indica que el modelo es excepcionalmente efectivo al identificar las instancias positivas correctamente. Además, este modelo tiene un alto valor de recall, lo que demuestra su habilidad para capturar una amplia mayoría de los casos positivos reales. La métrica F1 Score, que relaciona precisión y recall, es significativamente alta, reflejando un equilibrio robusto entre estas dos métricas críticas.



est=50 lr=0,2 con ES	
ACCURACY	0.998398
PRECISION	0.952641
RECALL	0.971914
F1	0.924954
B_ACCURACY	0.971914
AUC	0.998835
TPR	0.944864
FPR	0.001037
TNR	0.998963
FNR	0.055136

Modelo con 50 estimadores, 0.5 de learning rate y Early Stopping

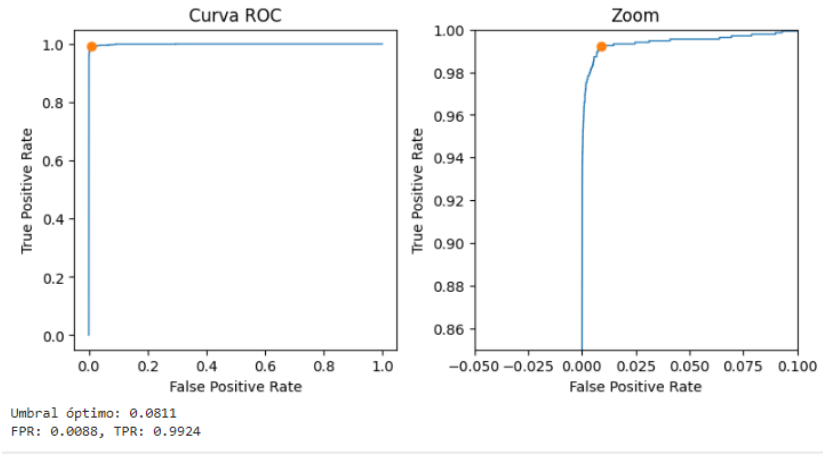
Por otro lado, el segundo modelo con 50 estimadores y una tasa de aprendizaje de 0.5, si bien mejora ligeramente en recall, sufre una disminución en precisión. Esto sugiere una tendencia a incrementar los falsos positivosl. Se observa un aumento en la tasa de falsos positivos, lo que puede ser un punto de preocupación dependiendo del contexto específico de aplicación.



est=50 lr=0,5 con ES	
ACCURACY	0.997679
PRECISION	0.921882
RECALL	0.976034
F1	0.895745
B_ACCURACY	0.976034
AUC	0.998297
TPR	0.953927
FPR	0.001859
TNR	0.998141
FNR	0.046073

Modelo con 100 estimadores, 0.2 de learning rate y Early Stopping

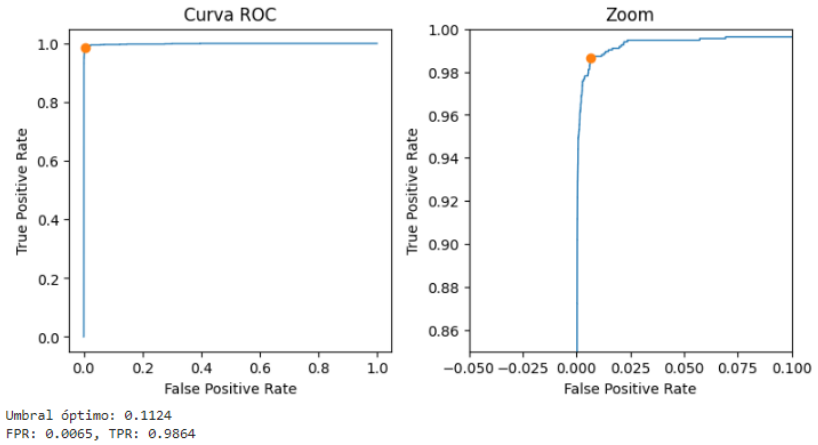
El modelo con 100 estimadores y una tasa de aprendizaje de 0.2 se presenta como el más prometedor. Exhibe la más alta precisión y recall, culminando en el valor F1 más alto, un indicador de un balance ideal entre ambas métricas. Este modelo es extremadamente competente en la clasificación correcta, con las tasas más altas de verdaderos positivos y verdaderos negativos y la tasa más baja de falsos positivos.



est=100 lr=0,2 con ES	
ACCURACY	0.998792
PRECISION	0.963627
RECALL	0.979212
F1	0.943186
B_ACCURACY	0.979212
AUC	0.999198
TPR	0.959215
FPR	0.000790
TNR	0.999210
FNR	0.040785

Modelo con 100 estimadores, 0.5 de learning rate y Early Stopping

Finalmente, el último modelo con 100 estimadores y una tasa de aprendizaje de 0.5 también muestra un alto recall, pero su precisión disminuye ligeramente en comparación con el modelo anterior. Su valor F1 y AUC son altos, pero no alcanzan los del modelo con un menor learning rate.



est=100 lr=0,5 con ES	
ACCURACY	0.997955
PRECISION	0.929574
RECALL	0.979911
F1	0.907665
B_ACCURACY	0.979911
AUC	0.998467
TPR	0.961480
FPR	0.001659
TNR	0.998341
FNR	0.038520

Después de evaluar cuidadosamente las métricas de rendimiento, el modelo con 100 estimadores y una tasa de aprendizaje de 0.2 demuestra ser el más efectivo en general. Ofrece un equilibrio óptimo entre precisión y capacidad de identificar positivos reales, a la vez que minimiza las tasas de error.

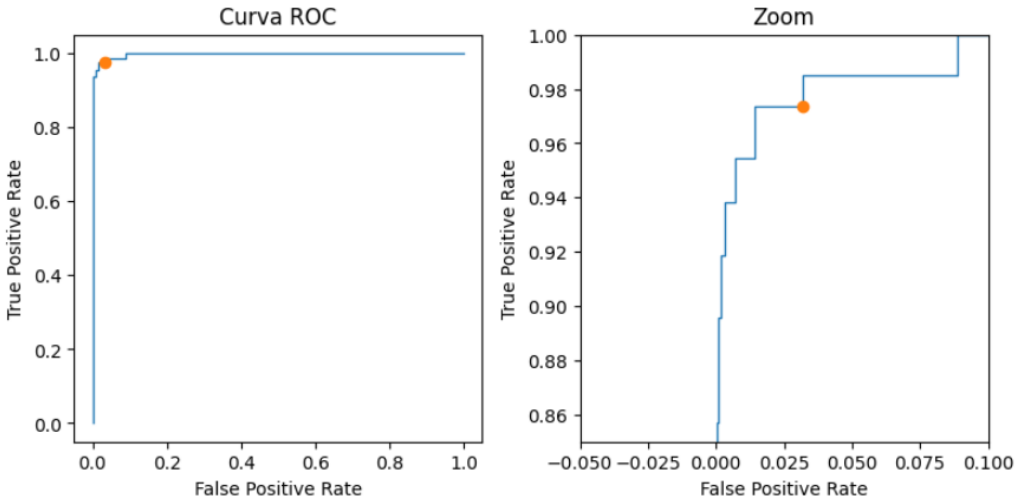
Elección de mejor Modelo

Tras haber analizado todas las variantes, llegamos a que nuestros dos modelos candidatos de Gradient Boosting son los modelos con 100 estimadores, 0.2 de learning rate con y sin Early Stopping. Observando ambos modelos, se puede observar que las métricas son extremadamente similares salvo algunas ligerísimas diferencias en F1 Score y AUC. Sin embargo, la diferencia más notoria que observamos fue en el tiempo de ejecución gracias a la ventaja práctica del Early Stopping que reduce el tiempo de entrenamiento y previene el sobreajuste al detener el entrenamiento cuando el modelo ya no mejora en un conjunto de validación. Por este motivo, tomamos la decisión de seleccionar al modelo de 100 estimadores, 0.2 de learning rate con Early Stopping como el mejor modelo de Gradient Boosting.

Bagging

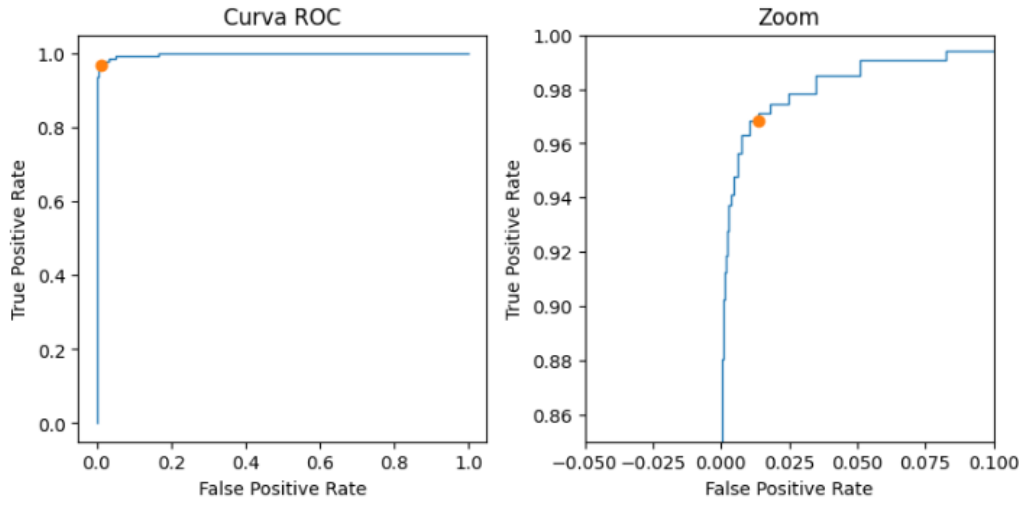
Para nuestro experimento con Bagging decidimos hacer una corrida con los valores por defecto, y otra con el hiperparámetro n_estimators=30. Dado que la diferencia no es notable, y que el segundo experimento es notablemtene más complejo, decidimos elegir el modelo de bagging con n_estimators=10 como el mejor.

n_estimators=10	
ACCURACY	0.997245
PRECISION	0.923930
RECALL	0.947044
F1	0.871738
B_ACCURACY	0.947044
AUC	0.990226
TPR	0.895770
FPR	0.001683
TN0R	0.998317
FNR	0.104230




```
Umbral óptimo: 0.2
FPR: 0.0317, TPR: 0.9736
```

n_estimators=30	
Accuracy	0.997616
Precision	0.936254
Recall	0.950594
F1	0.887816
B_Accuracy	0.950594
AUC	0.994793
TPR	0.902568
FPR	0.001380
TNR	0.998620
FNR	0.097432




```
Umbral óptimo: 0.2333
FPR: 0.0136, TPR: 0.9683
```

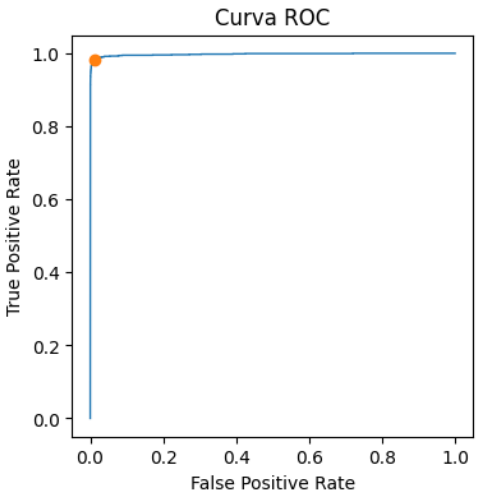
Deep Learning

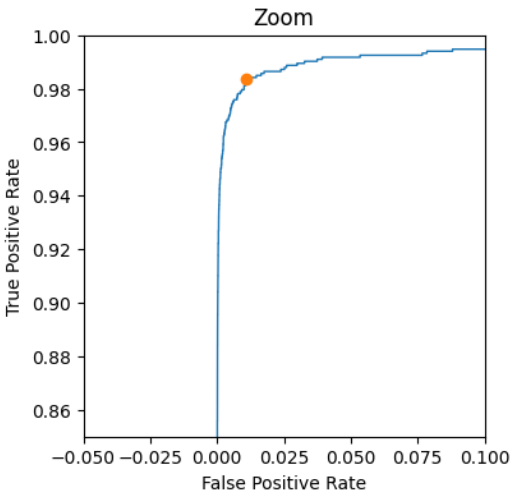
En nuestra estrategia para optimizar modelos de deep learning, variamos sistemáticamente el número de capas y neuronas. Este enfoque nos permite evaluar cómo distintas configuraciones influyen en el aprendizaje y la precisión del modelo, facilitando la identificación de la arquitectura más eficiente para nuestro conjunto de datos específico.

1 - 1

El modelo con una capa de 1 neurona seguido de otra capa de 1 muestra una alta Accuracy (0.995706), indicando una excelente capacidad general de clasificación. A pesar de su simplicidad, logra un AUC sobresaliente aunque su Precision sugiere un margen de mejora en la identificación precisa de instancias positivas. Este modelo básico es un buen punto de partida, pero podría beneficiarse de una mayor complejidad para capturar mejor las sutilezas de los datos.

	Métrica	Valor
0	Accuracy	0.995706
1	Precision	0.859137
2	Recall	0.982137
3	F1	0.824968
4	B_Accuracy	0.982137
5	AUC	0.997466



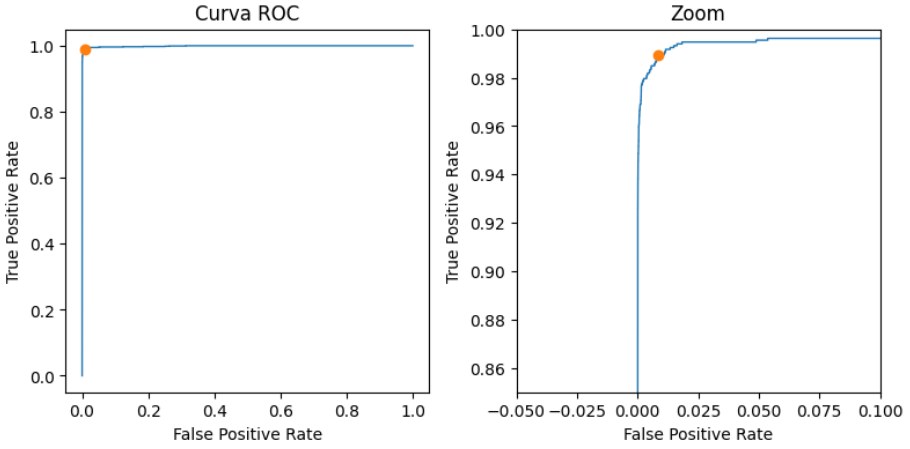



```
Umbral óptimo: 0.22699999809265137
FPR: 0.0109, TPR: 0.9834
```

16 - 1

El modelo mejorado con 16 neuronas en la primera capa muestra un aumento notable en precisión, manteniendo una alta exactitud y AUC, lo que indica una mejor identificación de instancias positivas y una distinción más clara entre las clases. La ligera disminución en F1 sugiere un cambio en el equilibrio entre precisión y sensibilidad, pero en general, el modelo exhibe un rendimiento más equilibrado y robusto.

	Métrica	Valor
0	Accuracy	0.998524
1	Precision	0.948752
2	Recall	0.983934
3	F1	0.932074
4	B_Accuracy	0.983934
5	AUC	0.998884

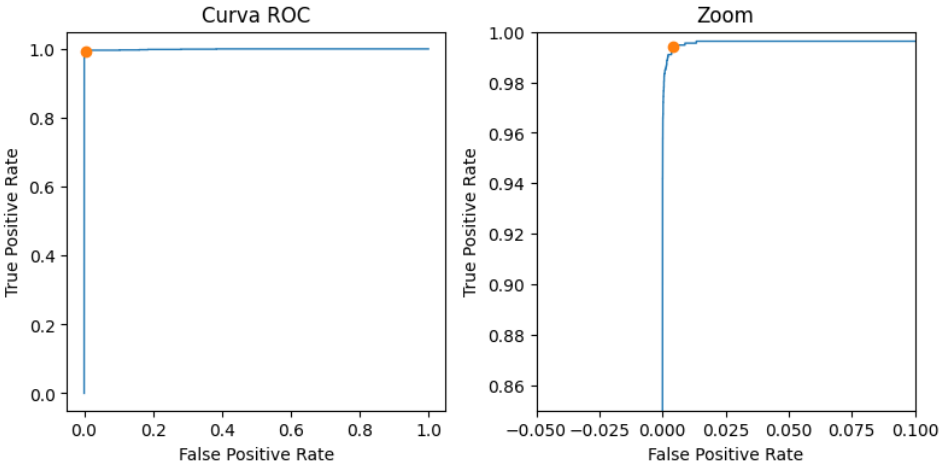



```
Umbral óptimo: 0.04740000143647194
FPR: 0.0086, TPR: 0.9894
```

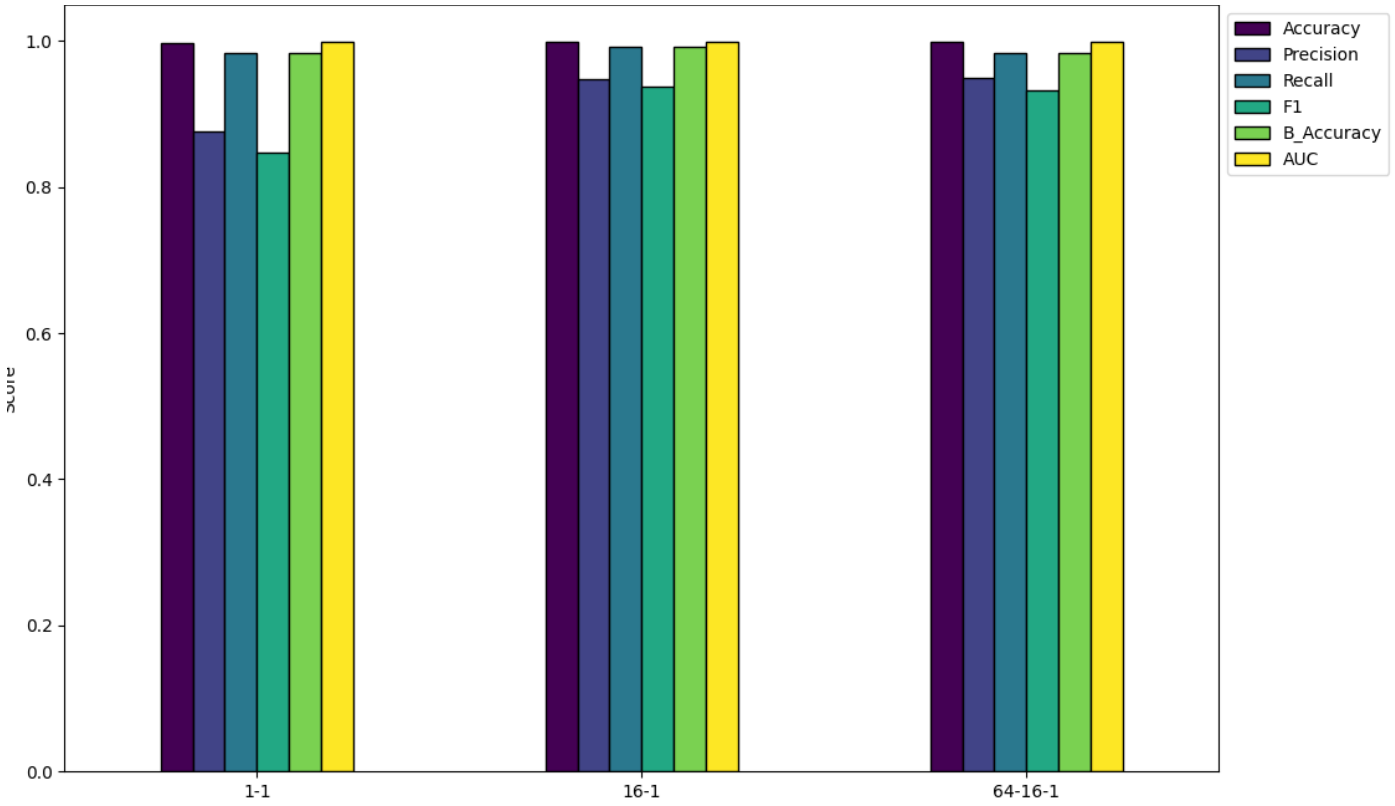
64 - 16 - 1

La precisión no mostró una mejora significativa con la configuración de 64-16-1 neuronas. Esto sugiere que simplemente aumentar la complejidad del modelo no siempre resulta en mejoras sustanciales en todas las métricas, especialmente en un conjunto de datos donde el modelo ya está rindiendo a un nivel muy alto. Por esa razón no seguiremos probando con más modelos neuronales.

	Métrica	Valor
0	Accuracy	0.998626
1	Precision	0.947108
2	Recall	0.991833
3	F1	0.937455
4	B_Accuracy	0.991833
5	AUC	0.999101



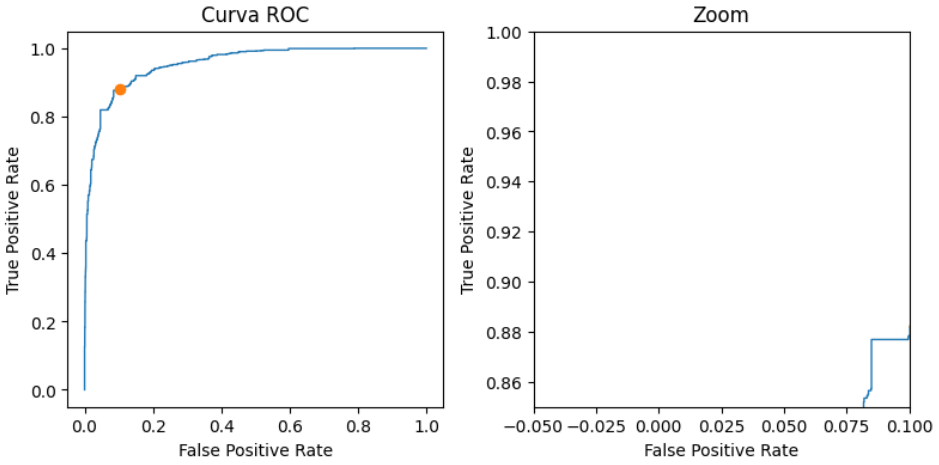

```
Umbral óptimo: 0.09650000184774399
FPR: 0.0042, TPR: 0.994
```


Accuracy
Precision
Recall
F1
B_Accuracy
AUC
1.0
0.8
0.6
0.4
0.2
0.0
1-1
16-1
64-16-1

Random Forest

max_depth=1, n_estimators=10

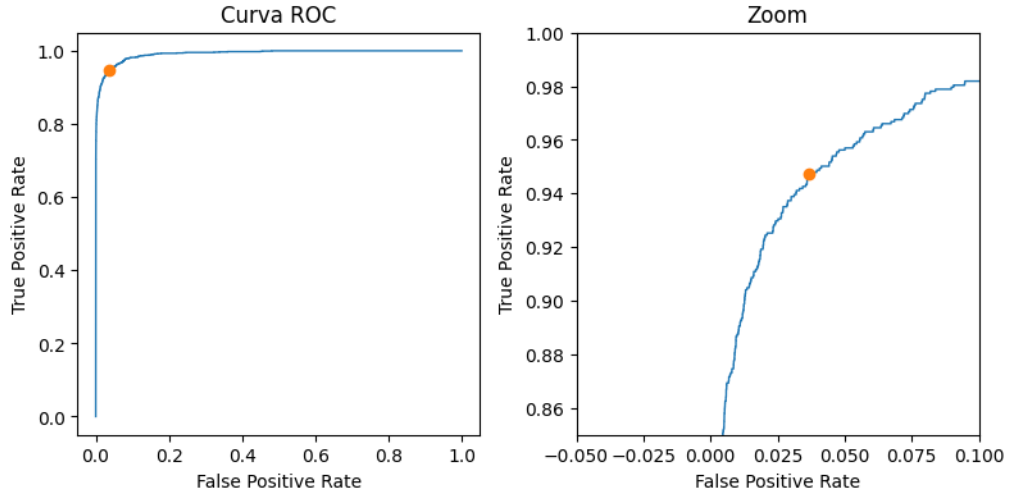
	Métrica	Valor
0	Accuracy	0.989548
1	Precision	0.494774
2	Recall	0.500000
3	F1	0.000000
4	B_Accuracy	0.500000
5	AUC	0.953226
6	TPR	0.000000
7	FPR	0.000000
8	TNR	1.000000
9	FNR	1.000000




```
Umbral óptimo: 0.1509
FPR: 0.1019, TPR: 0.8822
```

max_depth=5, n_estimators=10

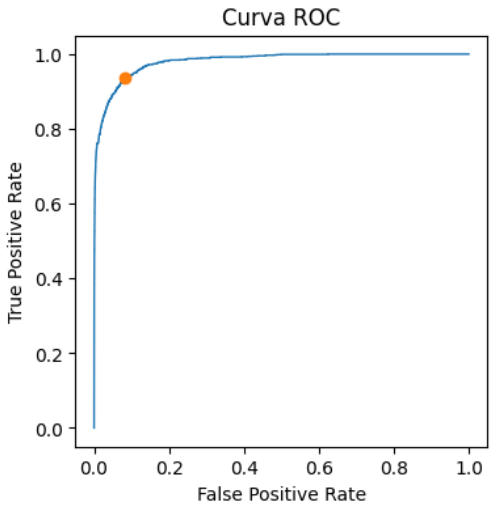
	Métrica	Valor
0	Accuracy	0.996653
1	Precision	0.945816
2	Recall	0.885091
3	F1	0.828062
4	B_Accuracy	0.885091
5	AUC	0.992137
6	TPR	0.771148
7	FPR	0.000965
8	TNR	0.999035
9	FNR	0.228852

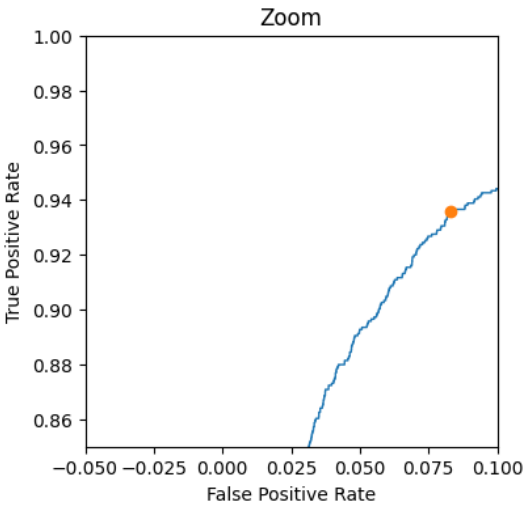



```
Umbral óptimo: 0.1743
FPR: 0.0368, TPR: 0.9471
```

max_depth=1, n_estimators=50

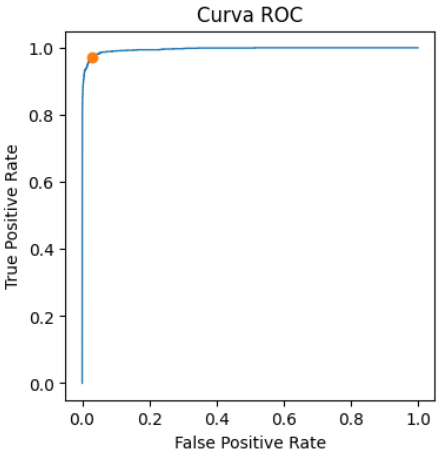
	Métrica	Valor
0	Accuracy	0.989548
1	Precision	0.494774
2	Recall	0.500000
3	F1	0.000000
4	B_Accuracy	0.500000
5	AUC	0.980715
6	TPR	0.000000
7	FPR	0.000000
8	TNR	1.000000
9	FNR	1.000000

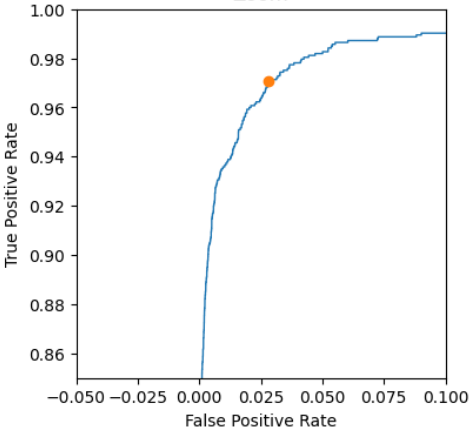




max_depth=5, n_estimators=50

	Métrica	Valor
0	Accuracy	0.997474
1	Precision	0.980859
2	Recall	0.893727
3	F1	0.866999
4	B_Accuracy	0.893727
5	AUC	0.995681
6	TPR	0.787764
7	FPR	0.000311
8	TNR	0.999689
9	FNR	0.212236

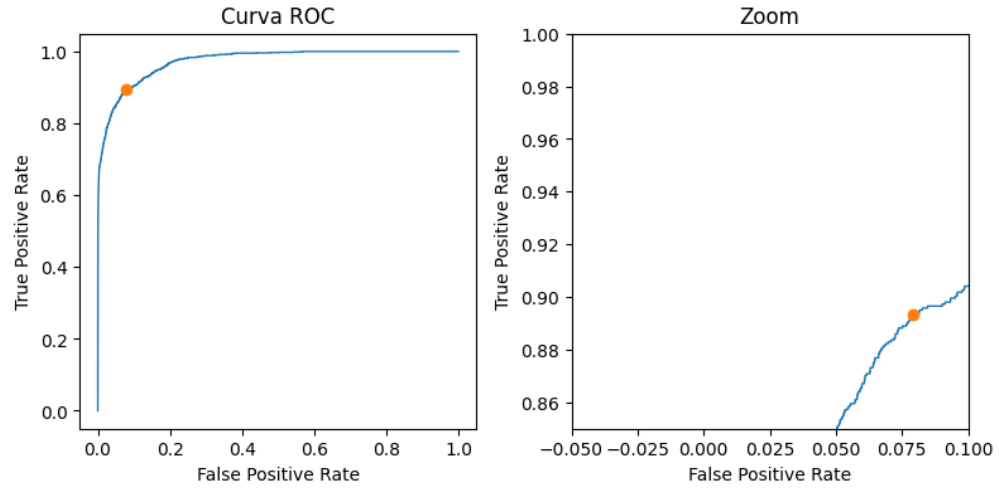





```
Umbral óptimo: 0.1797
FPR: 0.0282, TPR: 0.9705
```

max_depth=1, n_estimators=100

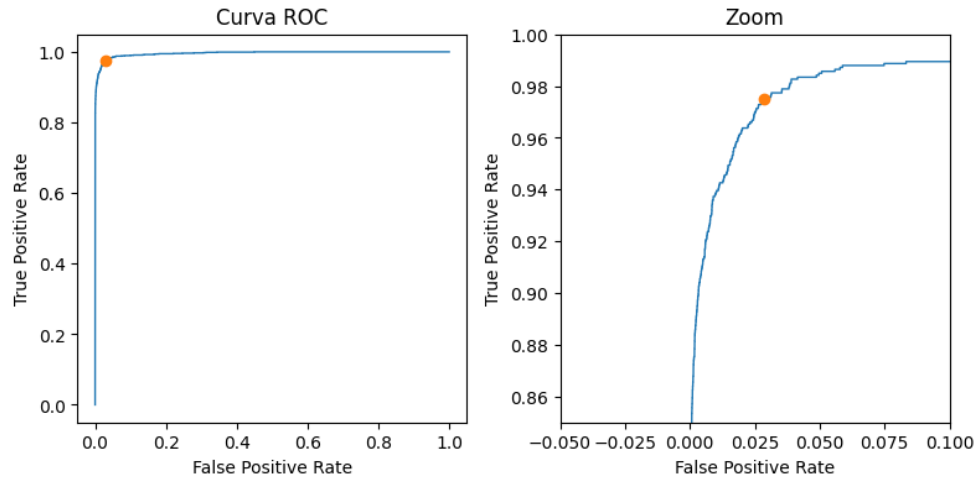
	Métrica	Valor
0	Accuracy	0.989548
1	Precision	0.494774
2	Recall	0.500000
3	F1	0.000000
4	B_Accuracy	0.500000
5	AUC	0.972911
6	TPR	0.000000
7	FPR	0.000000
8	TNR	1.000000
9	FNR	1.000000




```
Umbral óptimo: 0.1555
FPR: 0.079, TPR: 0.8935
```

max_depth=5, n_estimators=100

	Métrica	Valor
0	Accuracy	0.997577
1	Precision	0.986721
2	Recall	0.893778
3	F1	0.871709
4	B_Accuracy	0.893778
5	AUC	0.995918
6	TPR	0.787764
7	FPR	0.000207
8	TNR	0.999793
9	FNR	0.212236



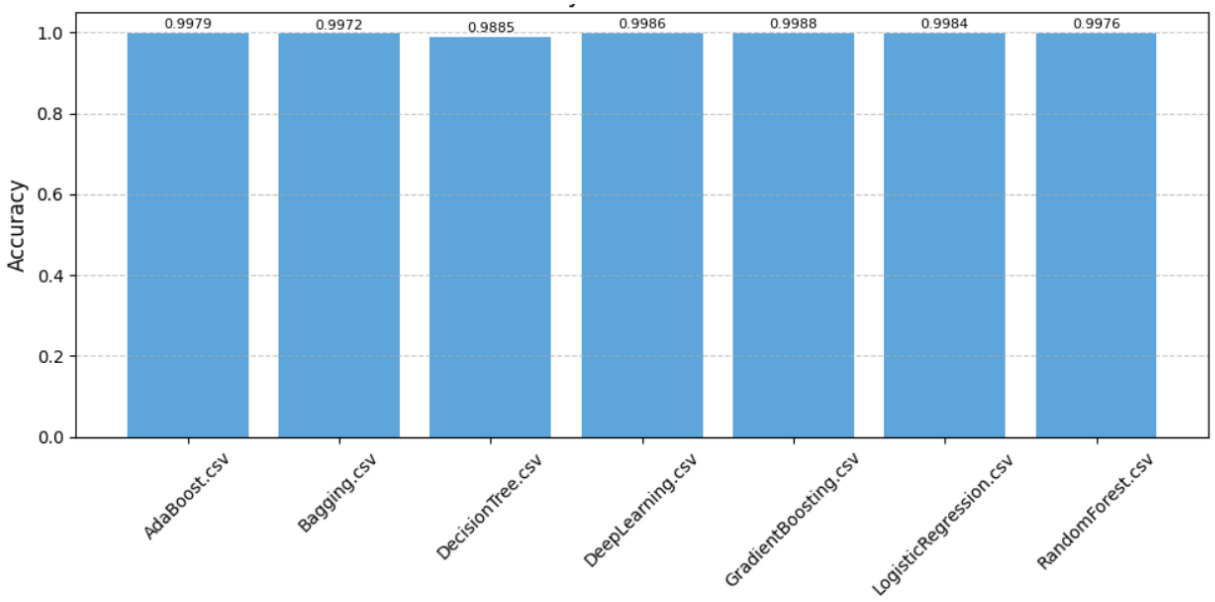

```
Umbral óptimo: 0.1741
FPR: 0.0287, TPR: 0.9751
```

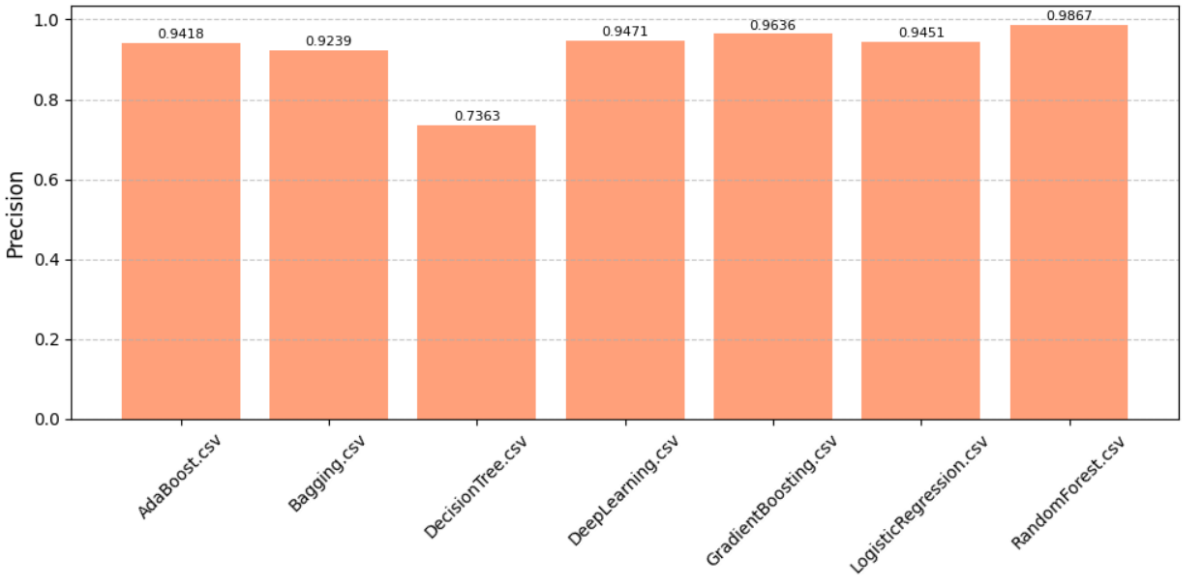
Elección de Modelo Final

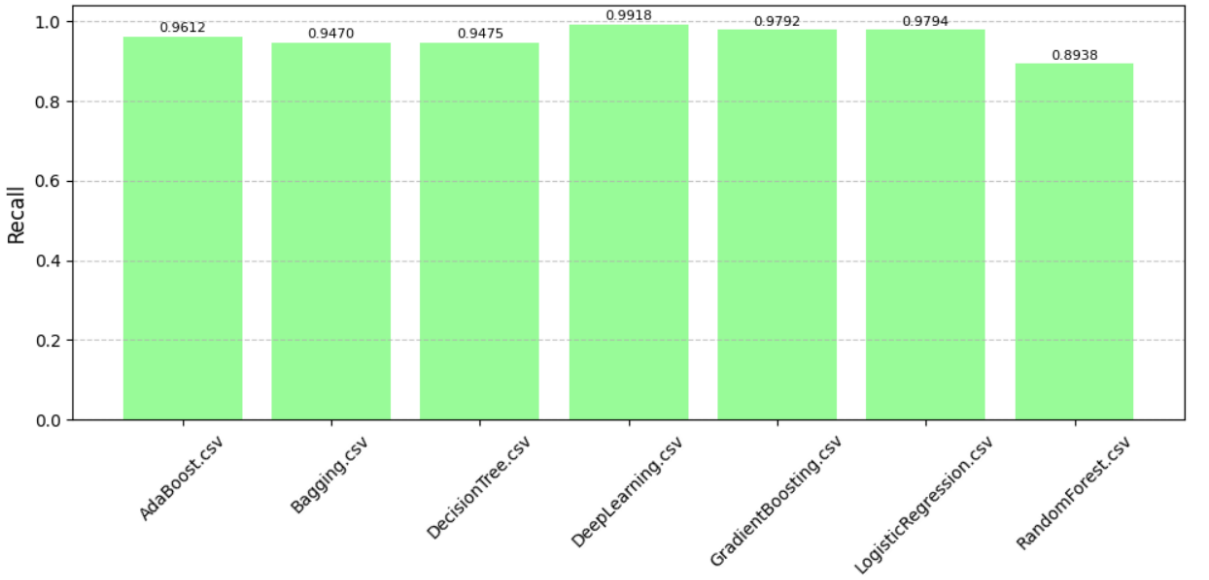
De todos nuestros experimentos, estos son los mejores resultados que ahora vamos a comparar.

- Logistic Regression(C=1, max_iter=1000)
- Decision Tree(max_depth=11)
- Adaptive Boosting(learning_rate=0.2, n_estimators=100)
- Deep Learning(64 - 16 - 1)
- Gradient Boosting(learning_rate=0.2, n_estimators=100 con ES)
- Random Forest(depth = 5, n_estimators = 100)
- Bagging(n_estimators=10)

Para hacer la comparativa, primero desarrollamos las siguientes gráficas:



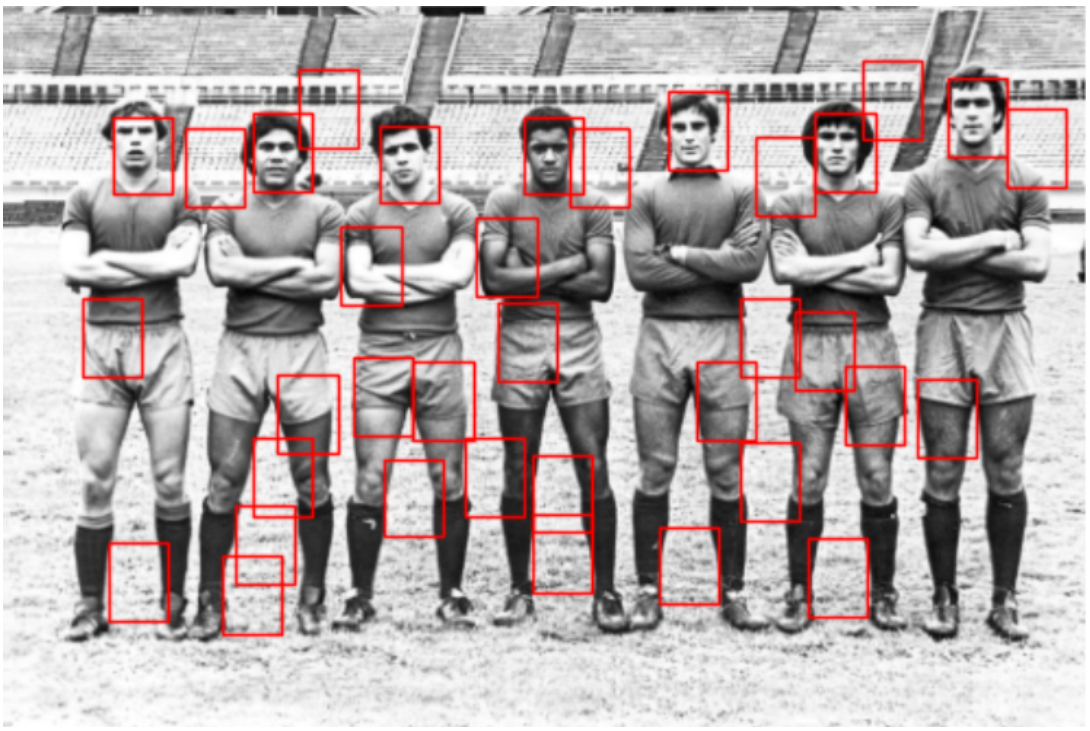




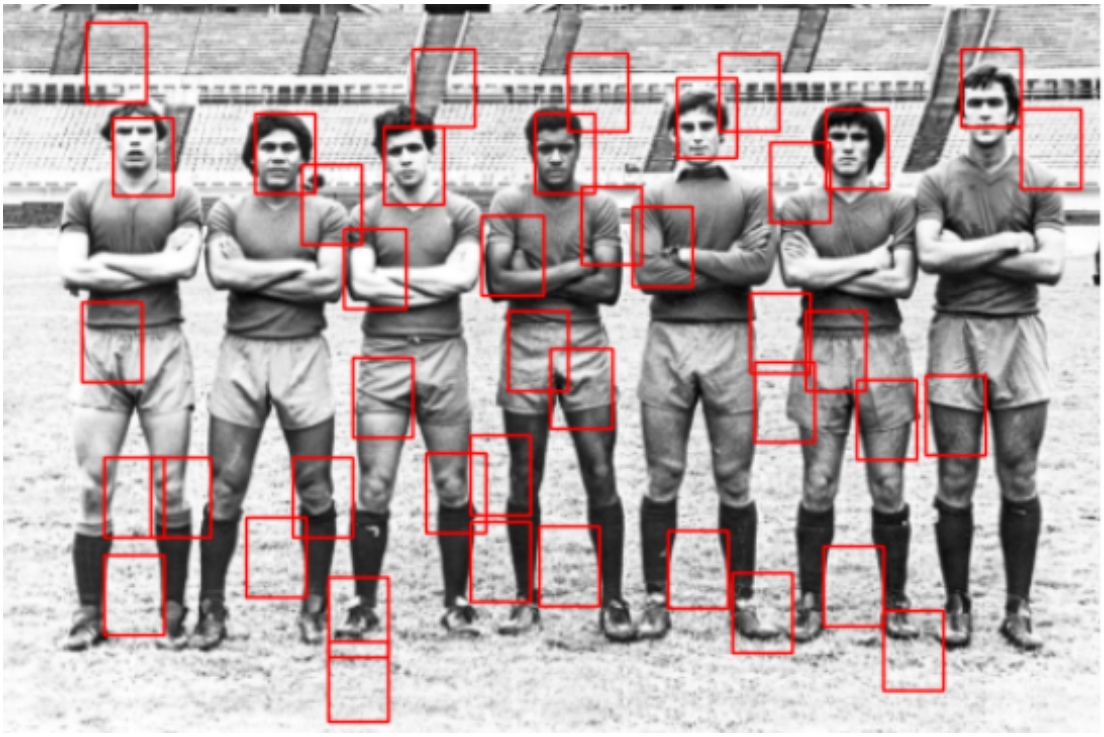
Las gráficas revelan que mientras la mayoría de los modelos exhiben una alta exactitud, el modelo de árbol de decisión se destaca por su precisión significativamente más baja. Esto indica que, aunque el árbol de decisión puede clasificar correctamente la mayoría de las instancias, tiende a generar más falsos positivos en comparación con otros modelos. Al ser tan parecidos los resultados en test, tomamos la decisión de poner a prueba los mejores modelos con distintas imágenes para finalmente inclinarnos por el mejor modelo de todos.

Observando las imágenes, llegamos a la conclusión de que el mejor modelo obtenido es **el de LogisticRegression,** dado que fue el que mejor se desempeñó en los casos de prueba empíricos **:**

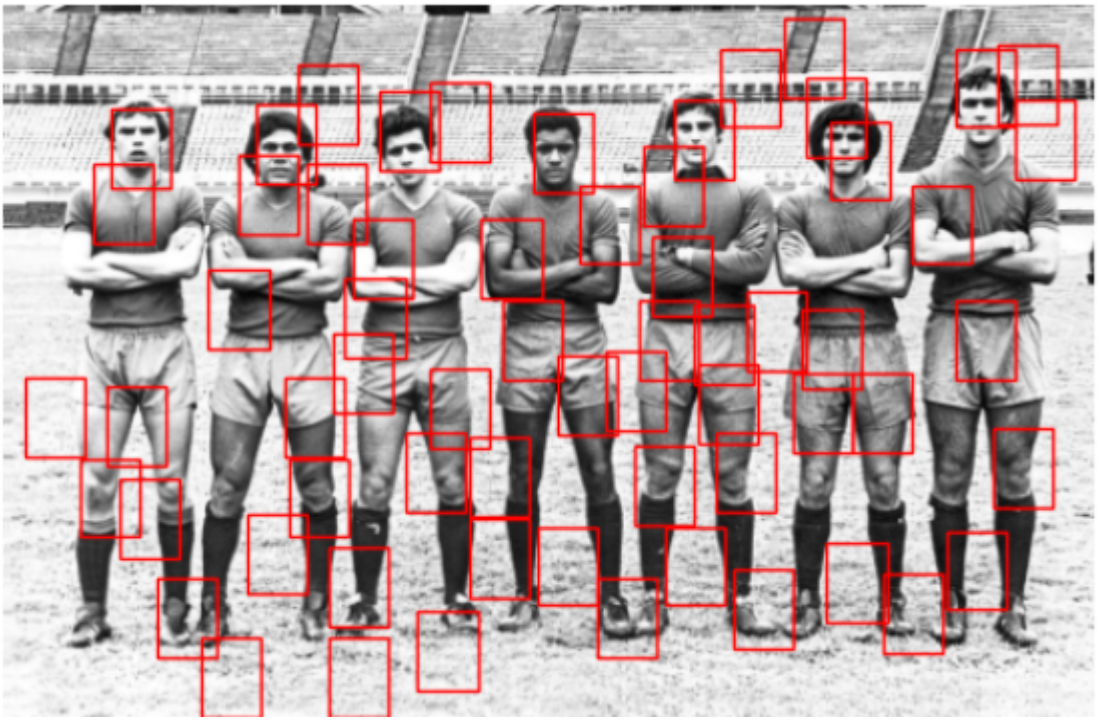
Logistic Regression



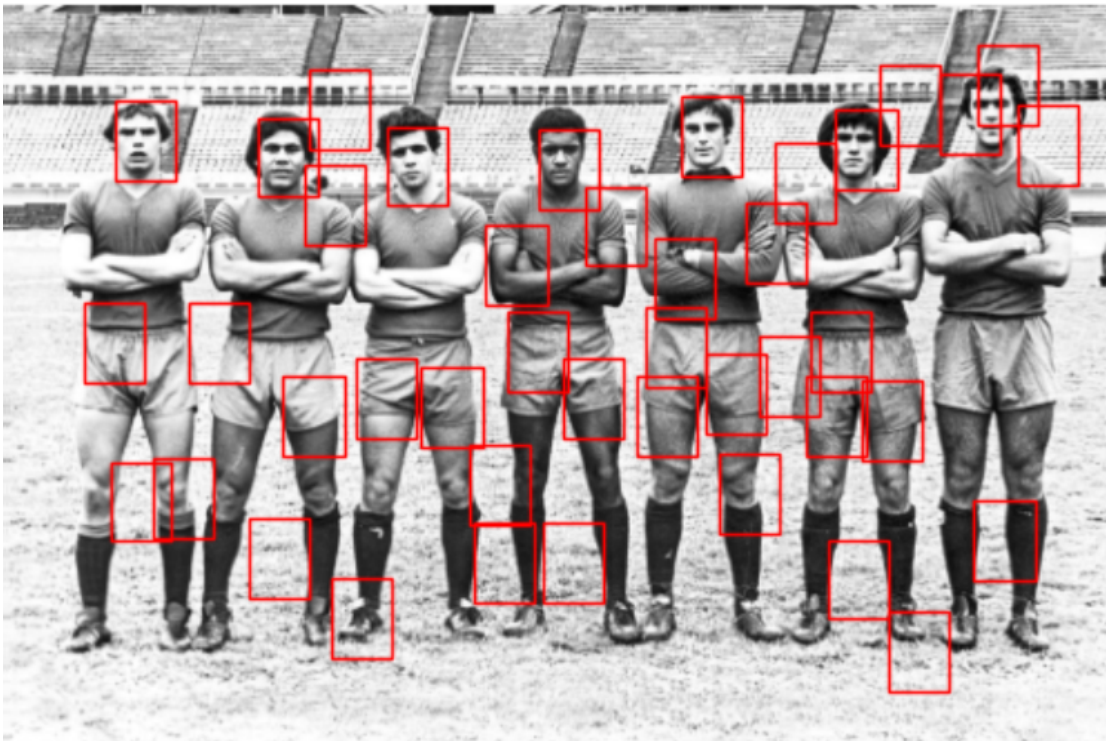
AdaBoost



Bagging



Gradient Boosting



Observaciones finales y Conclusión

Tras haber realizado el obligatorio llegamos a la conclusión de seleccionar al modelo de Logistic Regression como el mejor modelo final dado su desempeño tanto en las pruebas y entrenamiento como en los resultados empíricos.

En nuestro análisis de los modelos, observamos que todos mostraron un rendimiento en las pruebas notablemente superior a lo que esperábamos. Una de las posibles causas podría ser un sobreajuste, donde nuestros modelos se ajustaron demasiado bien a los datos de prueba específicos. A pesar de utilizarlas técnicas necesarias para prevenir esto, es posible que hayamos optimizado inadvertidamente los modelos en función de estos datos de prueba, lo que resultaría en una excelente performance en las pruebas, pero no necesariamente se traduciría en un buen rendimiento en datos no vistos. Esto se ve cuando observamos los resultados empíricos, donde notamos que estos modelos no alcanzaron el rendimiento esperado luego de haber visto las métricas obtenidas.

Mientras que nuestros modelos funcionan excepcionalmente bien en las pruebas, estos resultados pueden no haber capturado completamente las complejidades y variabilidades de los datos en entornos de producción reales, llevando a un rendimiento práctico que no cumplió con nuestras expectativas iniciales.

Nuestro trabajo con los modelos de clasificación ha sido enriquecedor y educativo. A través del preprocesamiento de datos, aprendimos cómo la calidad de los datos impacta directamente en los resultados del modelo. La extracción de características nos mostró la importancia de entender profundamente nuestros datos y elegir las variables más relevantes. Al explorar el funcionamiento de cada modelo y ajustar sus hiperparámetros, comprendimos mejor cómo cada decisión afecta el rendimiento general. En resumen, este obligatorio ha ampliado significativamente nuestra comprensión del área de machine learning y la importancia de una metodología cuidadosa en la misma. Fue, sin duda, una experiencia integral y muy valiosa.