

Bank Marketing Classification Task

Ana Gil, Wenya Zhong, Yassin Ettijani

× × ×



DataSet

01	y
02	Age
03	Job
04	Marital
05	Education
06	Default
07	Balance
08	Housing
09	Loan
10	Contact
11	Poutcome
12	Month



Regresión Logística

Modelo de regresión logística obtenido con step en R

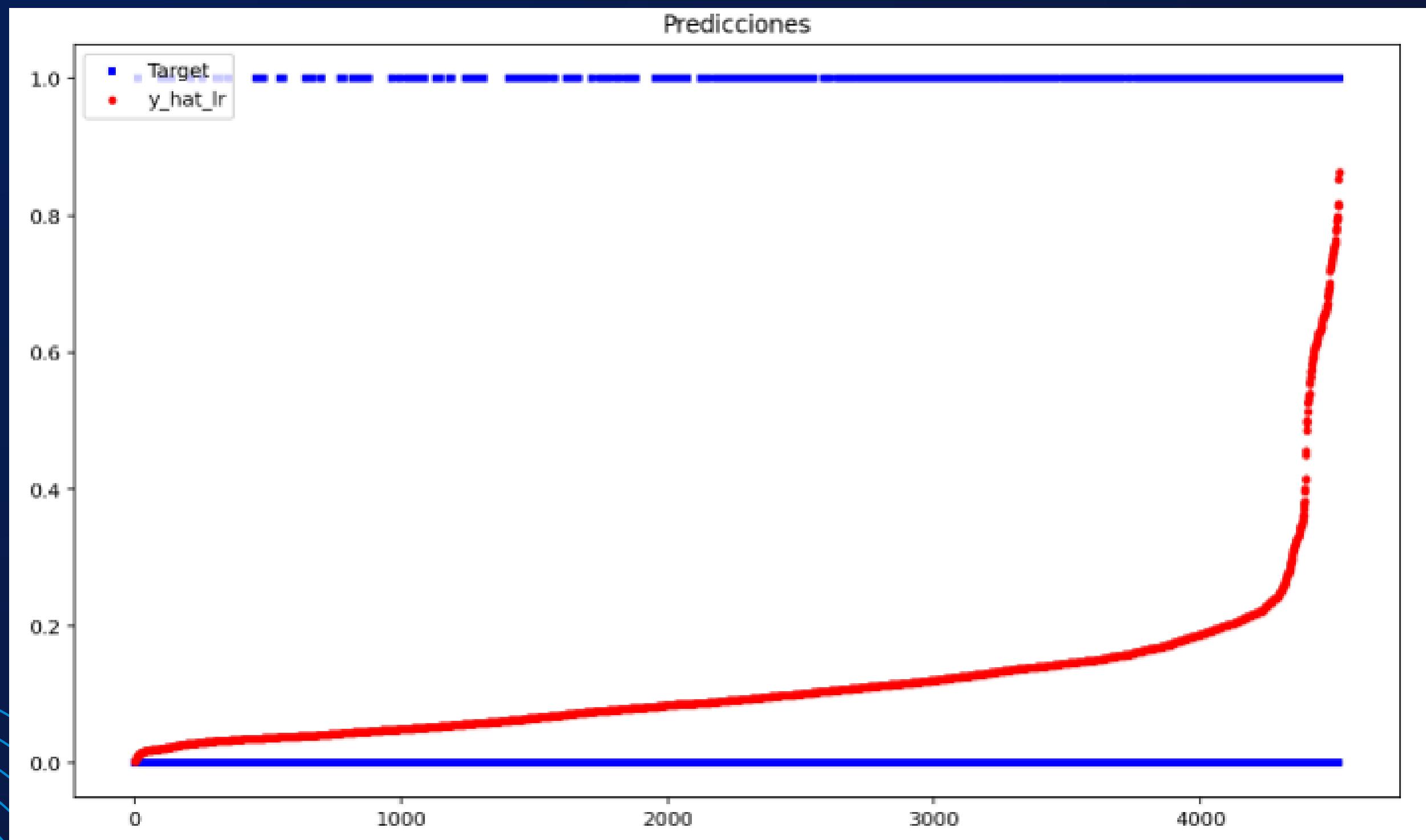
**VARIABLE
DEPENDIENTE**

y

**VARIABLES
INDEPENDIENTES**

- Age
- Marital
- Housing
- Loan
- Contact
- Day
- Month
- Campaign
- Poutcome

Regresión Logística

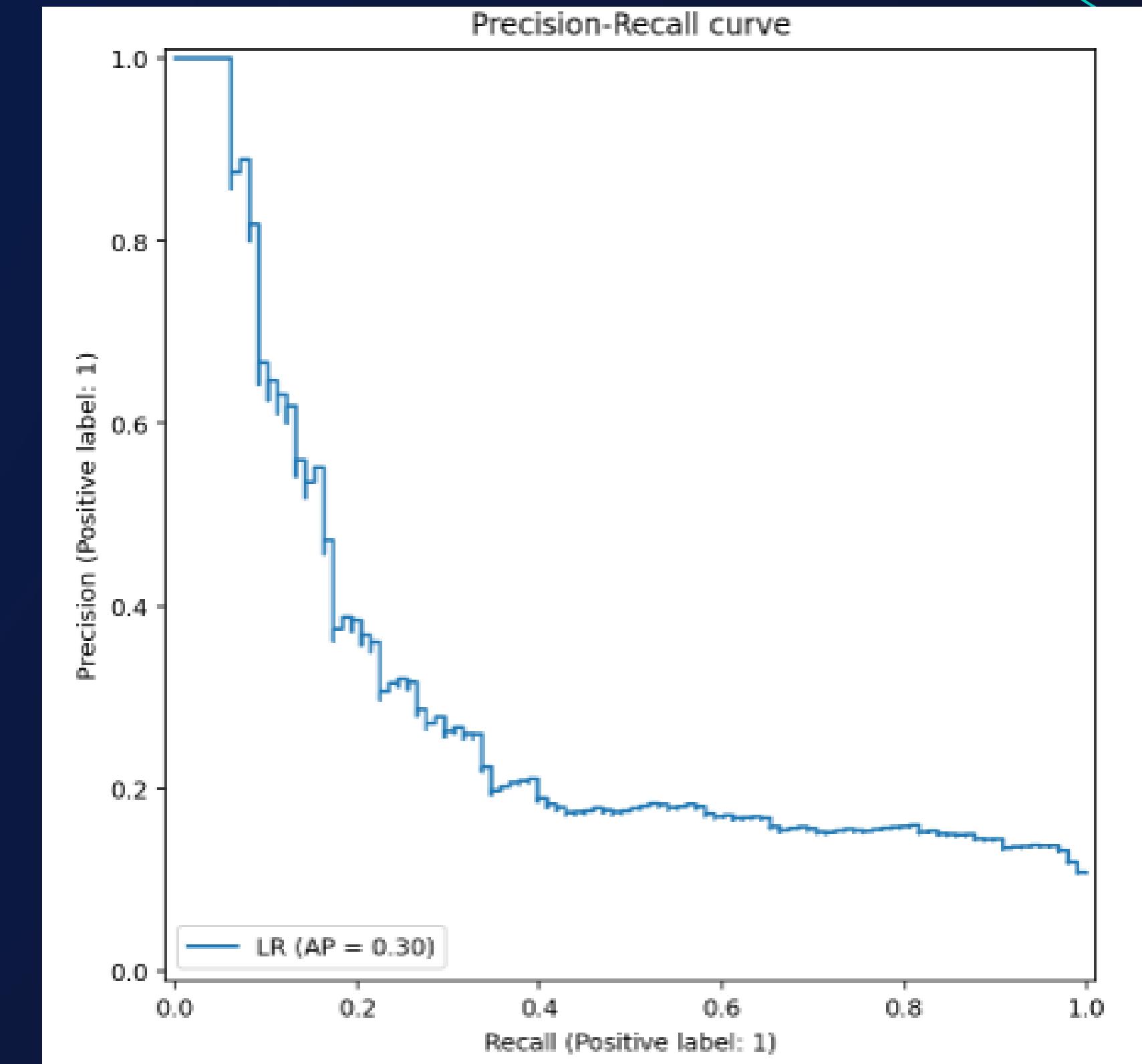


Directamente viendo esta gráfica podemos ver que un buen umbral de clasificación podría ser de 0.2

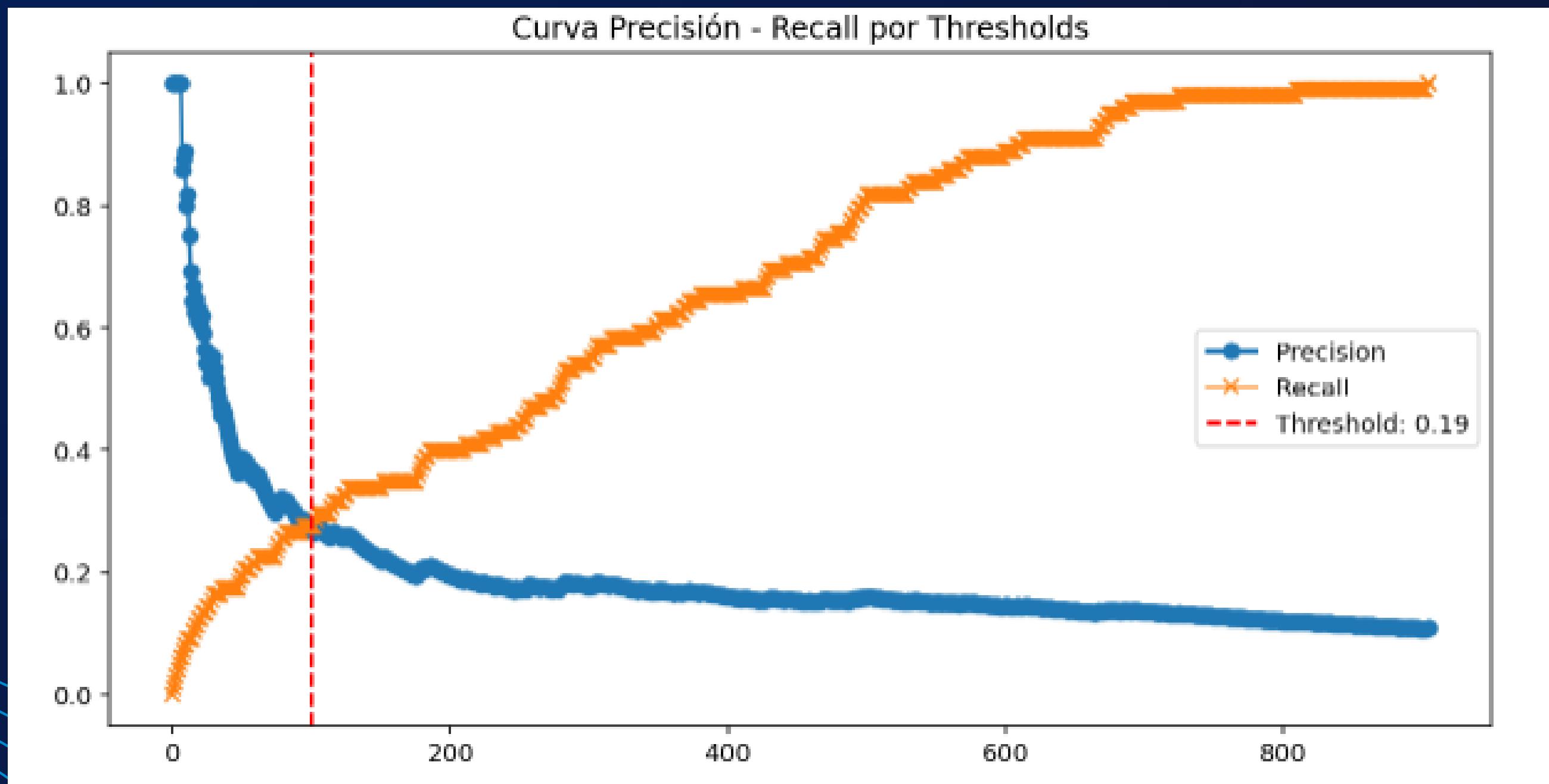
Ranking de variables

	feature	coef	rank
28	poutcome_success	1.551134	1.0
29	poutcome_unknown	-0.980407	2.0
26	poutcome_failure	-0.907449	3.0
25	contact_unknown	-0.761565	4.0
5	loan_num	-0.664873	5.0
13	job_retired	0.587732	6.0
21	married	-0.494578	7.0
4	housing_num	-0.316253	8.0
17	job_technician	-0.280067	9.0
9	job_blue-collar	-0.267814	10.0
27	poutcome_other	-0.237327	11.0
18	job_unemployed	-0.226901	12.0
15	job_services	-0.209791	13.0

Curva Precisión-Recall



Curva Precisión-Recall



Finalmente tomamos un threshold de 0.19

Evaluación del modelo

Accuracy = 0.843094

$$\frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} \mathbf{1}(\hat{y}_i = y_i)$$

Tabla de confusión

	Pred NO	Pred YES	TOTAL
True NO	736	71	807
True YES	71	27	98
TOTAL	807	98	905

Entrenado con
train (80%)
Evaluado con
test (20%)
Threshold: 0.19

Ratio Falsos Positivos:
0.7244897959
Ratio Falsos Negativos:
0.087980173

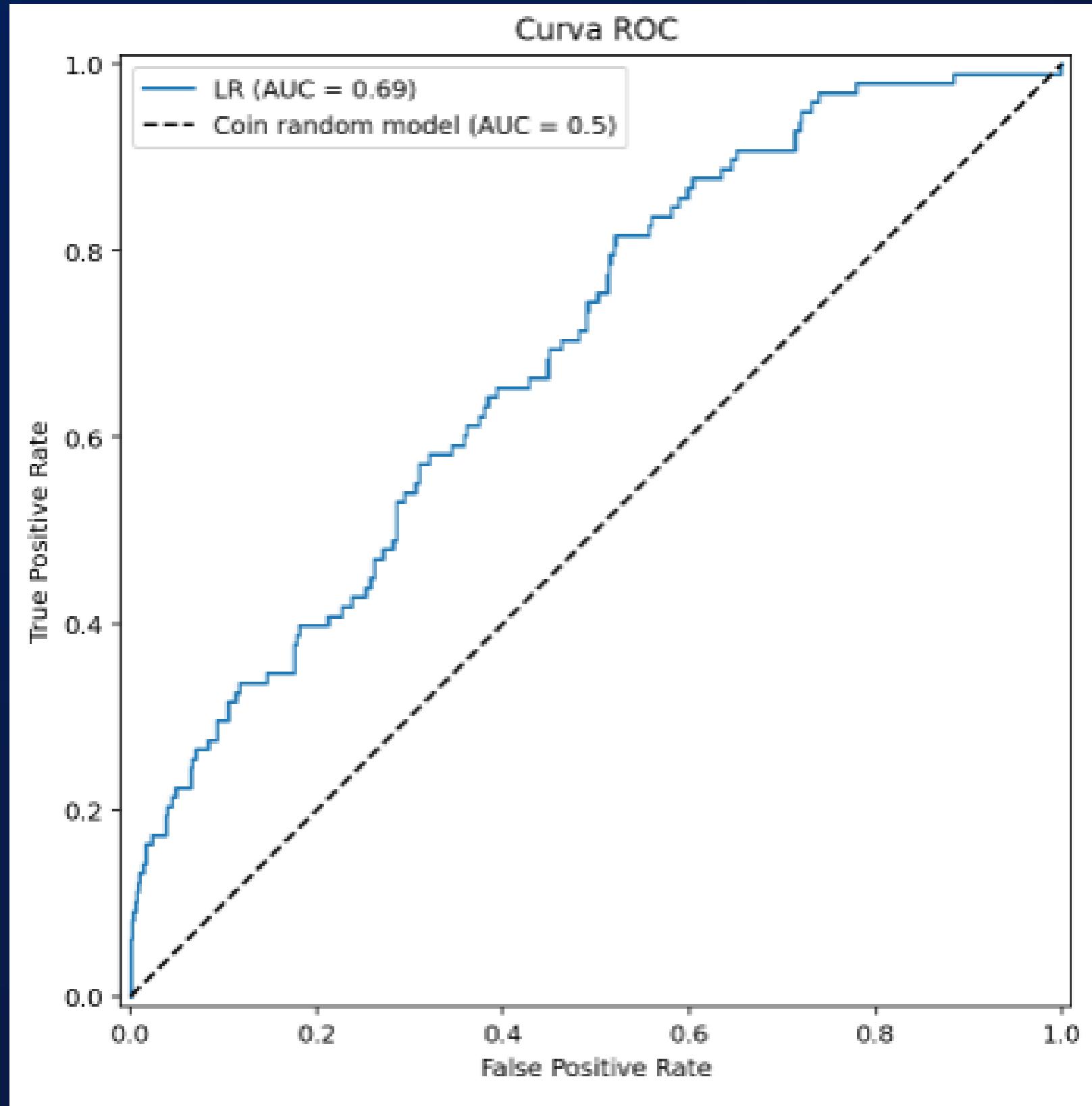
Kappa = 0.18753003

$$\frac{P_o - P_e}{1 - P_e}$$
 Concordancia pobre

Precisión = $\frac{TP}{TP+FP}$

Recall = $\frac{TP}{N_{y=1}}$

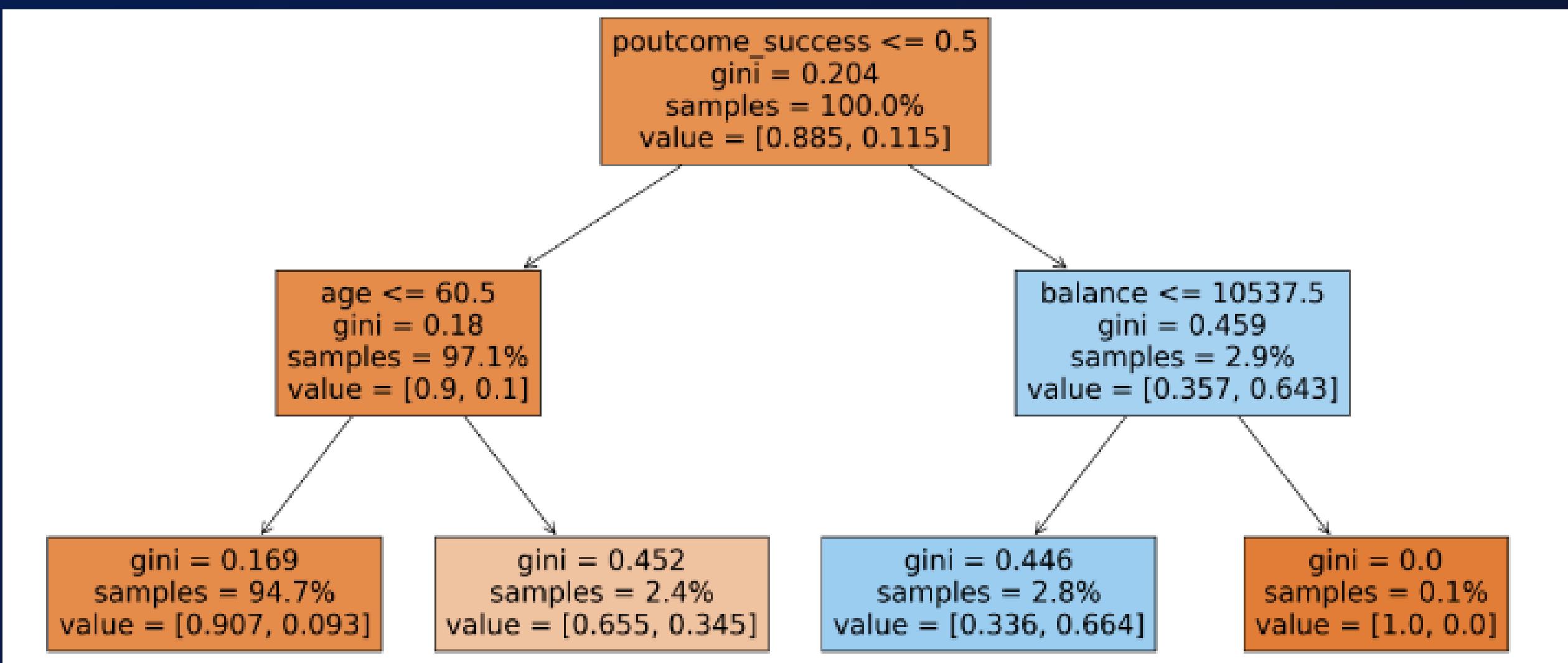
Curva ROC y AUC



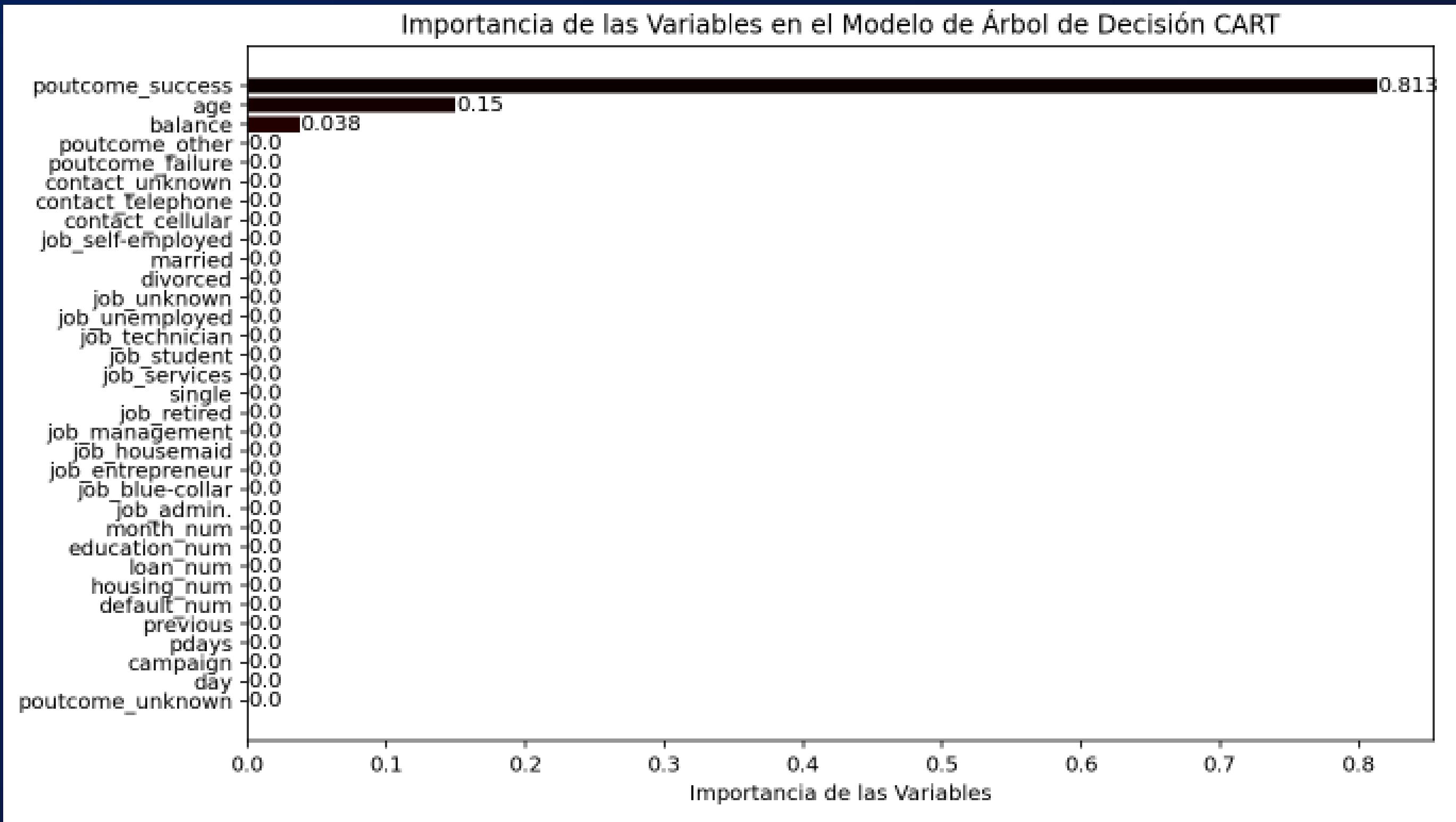
- $AUC = 0.6934097 > 0.5$
- Mejor que el azar

Árbol de Decisión

- Árbol de clasificación binaria: minimizar índice Gini
- Profundidad máxima del árbol: 2 niveles
- Número mínimo de muestras para dividir un nodo: 2 muestras
- Número mínimo de muestras en un nodo hoja: 1 muestra
- Reducción mínima de la impureza: 0



Importancia de las variables



Para cada variable que interviene en la división de un nodo, reducción de Gini que produce acumulada, ponderada por la cantidad de individuos afectados, y normalizada respecto del 100% de la importancia total.

Evaluación del modelo

Accuracy = 0.8961326

$$\frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} \mathbf{1}(\hat{y}_i = y_i)$$

Tabla de confusión

	Pred NO	Pred YES	TOTAL
True NO	798	9	807
True YES	85	13	98
TOTAL	883	22	905

Entrenado con
train (80%)
Evaluado con
test (20%)
Threshold: 0.5

Ratio Falsos Positivos:
0.409090909

Ratio Falsos Negativos:
0.096262741

Kappa = 0.18427815

$$\frac{P_o - P_e}{1 - P_e}$$
 Concordancia pobre

Precisión =
0.59090909

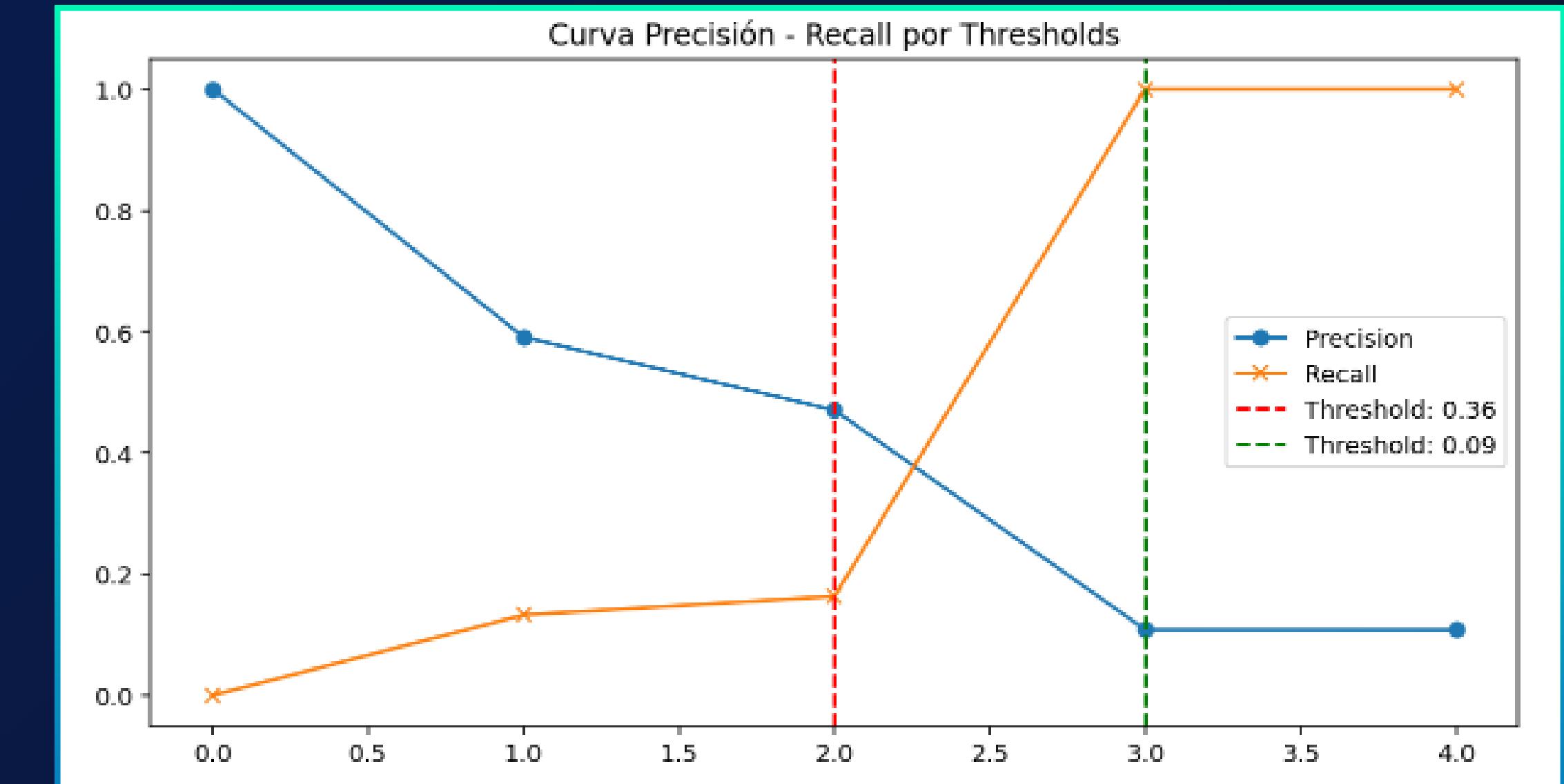
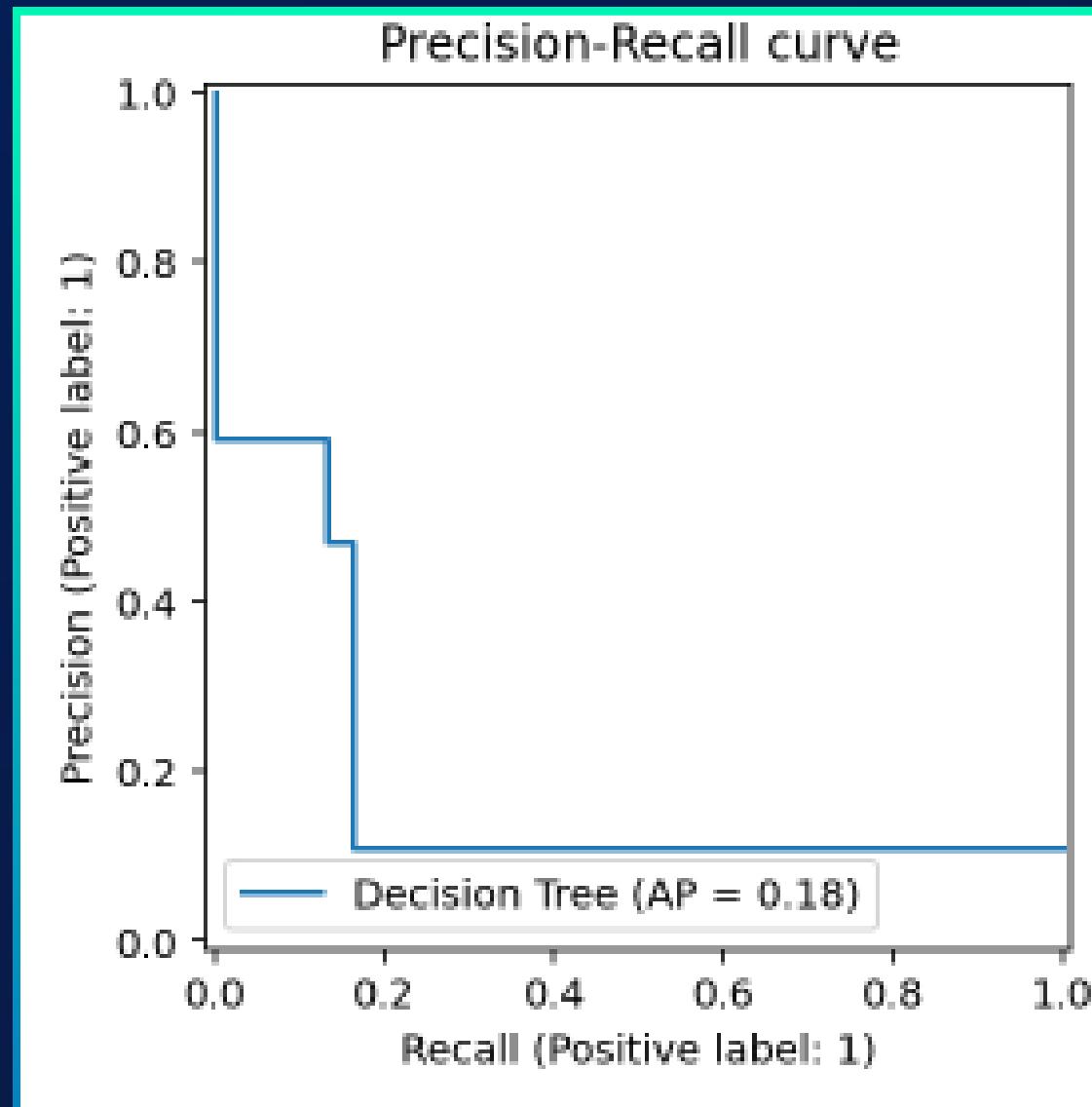
$$\frac{TP}{TP + FP}$$

Recall =
0.13265306

$$\frac{TP}{N_{y=1}}$$

Se pierden demasiados positivos (bajo recall)

Curva Precisión-Recall para buscar el mejor threshold



Nuevo threshold: 0.3

Evaluación del modelo

Accuracy = 0.889503

$$\frac{1}{n_{samples}} \sum_{i=0}^{n_{samples}-1} \mathbf{1}(\hat{y}_i = y_i)$$

Tabla de confusión

	Pred NO	Pred YES	TOTAL
True NO	789	18	807
True YES	82	16	98
TOTAL	871	34	905

Entrenado con
train (80%)
Evaluado con
test (20%)
Threshold: 0.3

Ratio Falsos Positivos:
0.5294117647 (aumenta)
Ratio Falsos Negativos:
0.0941446613 (disminuye)

Kappa = 0.1976666

$$\frac{P_o - P_e}{1 - P_e}$$
 Concordancia pobre

Precisión =
0.47058824

$$\frac{TP}{TP + FP}$$

Recall =
0.16326531

$$\frac{TP}{N_{y=1}}$$

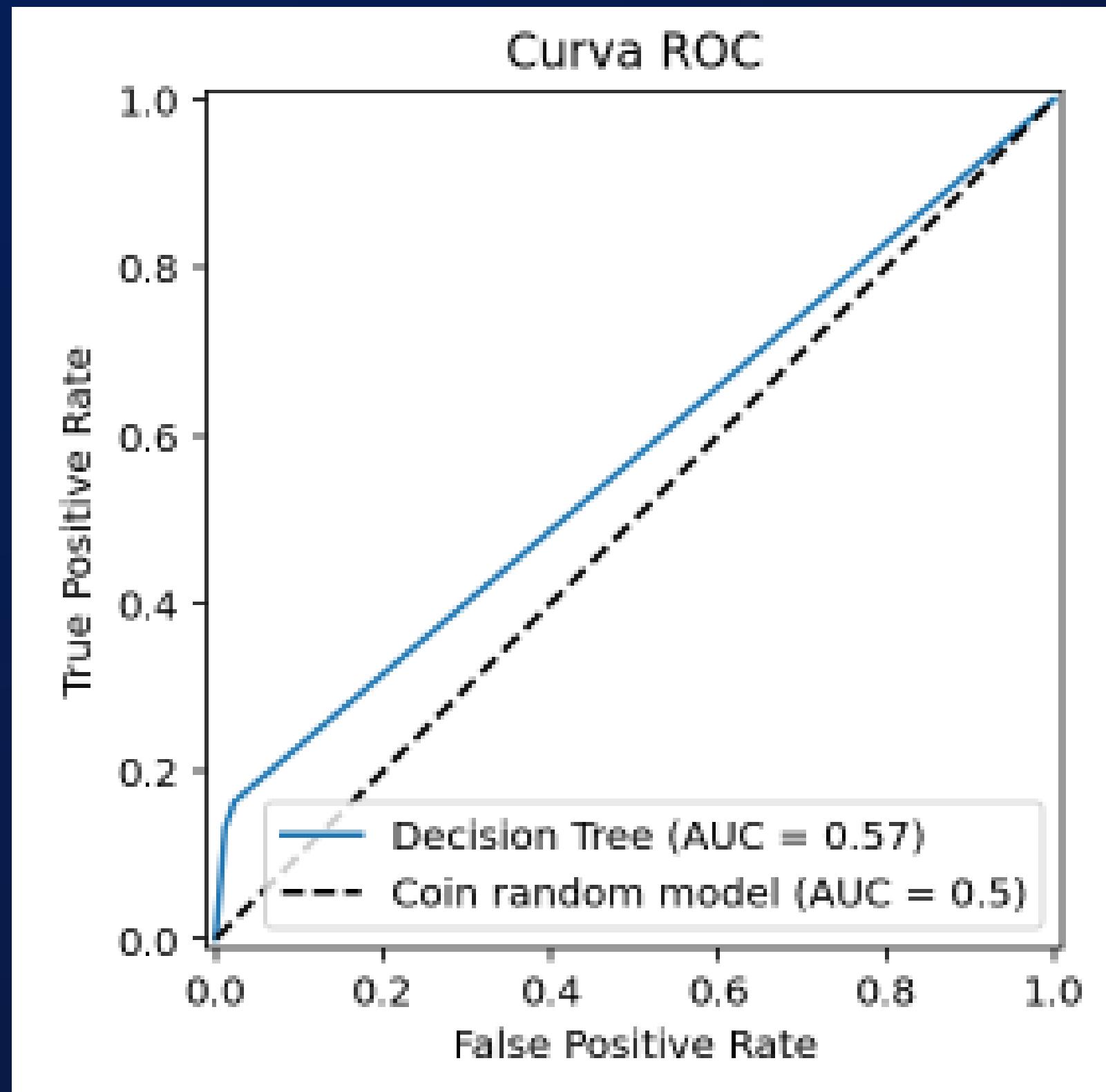
Disminuyen el ratio de acierto (accuracy) y la precisión

Conseguimos aumentar la exhaustividad (recall)

No
disminuyen
demasiado,
siguen
siendo
“altos”

Nos interesa
predecir a
los clientes
que sí se
suscribirán

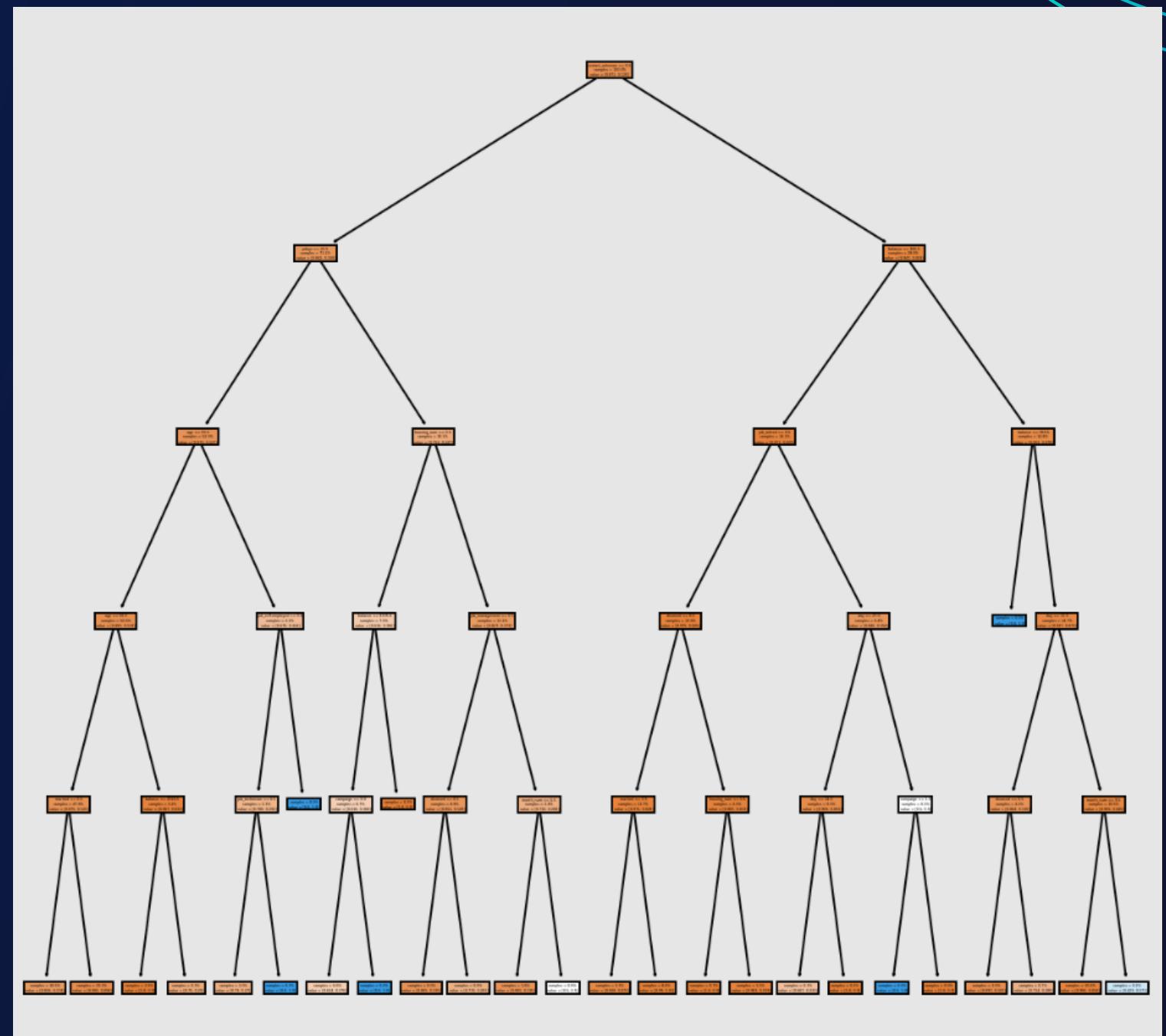
Curva ROC y AUC



- $AUC = 0.571568 > 0.5$
- Mejor que el azar
- No es una gran mejora

Random Forest

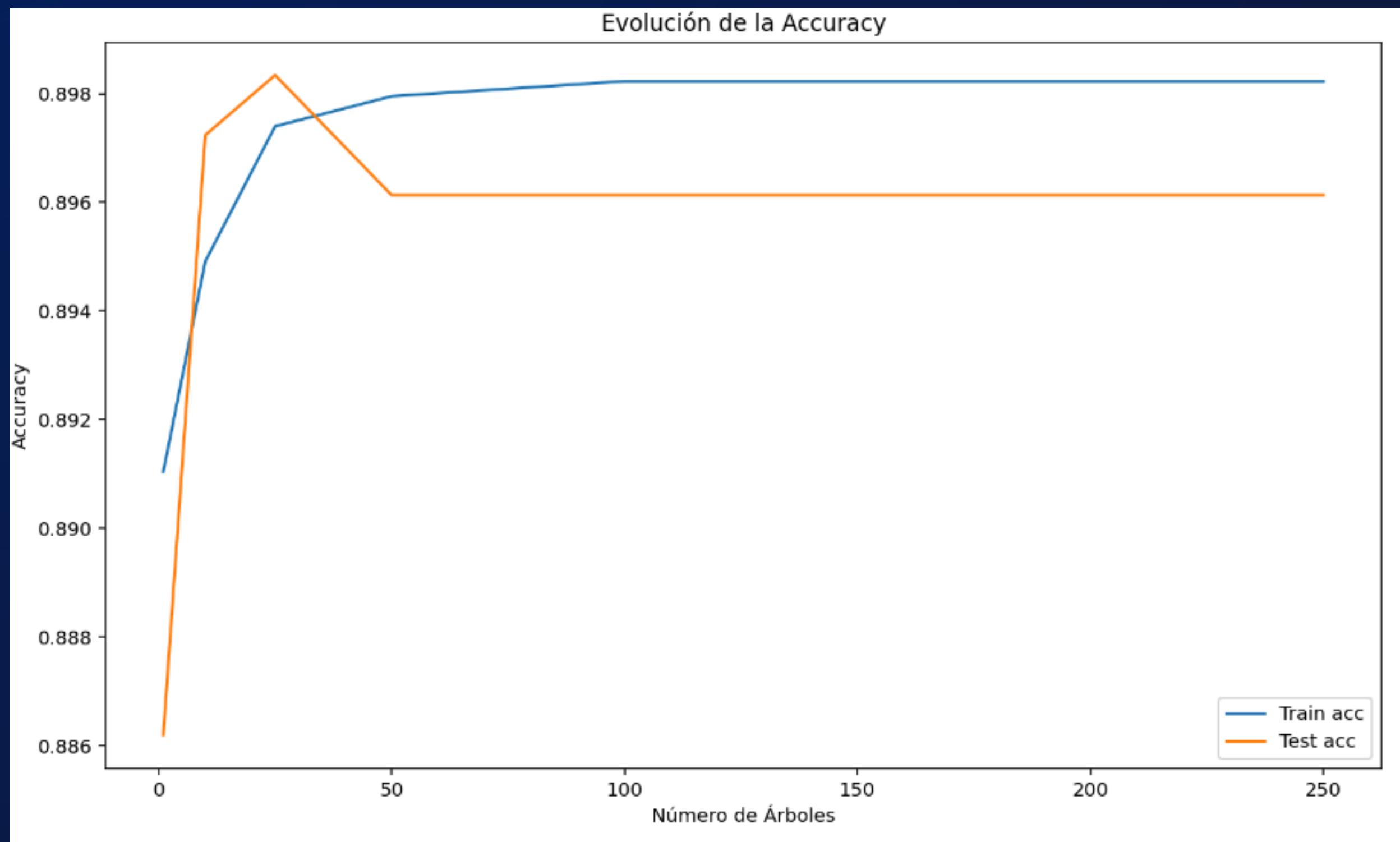
- Número de árboles: **25**
- Profundidad máxima del árboles: **5** niveles
- Número mínimo de muestras para dividir un nodo: **2** muestras
- Número mínimo de muestras en un nodo hoja: **1** muestra
- Número máximo de características: **5** características
- Usar bootstrap para muestrear
- No usar puntuación fuera de bolsa (OOB)
- Número de trabajos en paralelo, usar todos los núcleos disponibles
- Establecer random_state=**42** para asegurar la reproducibilidad



Visualización del árbol

Explore el mejor valor de n_estimators

Entrenado con train (80%)
Evaluado con test (20%)

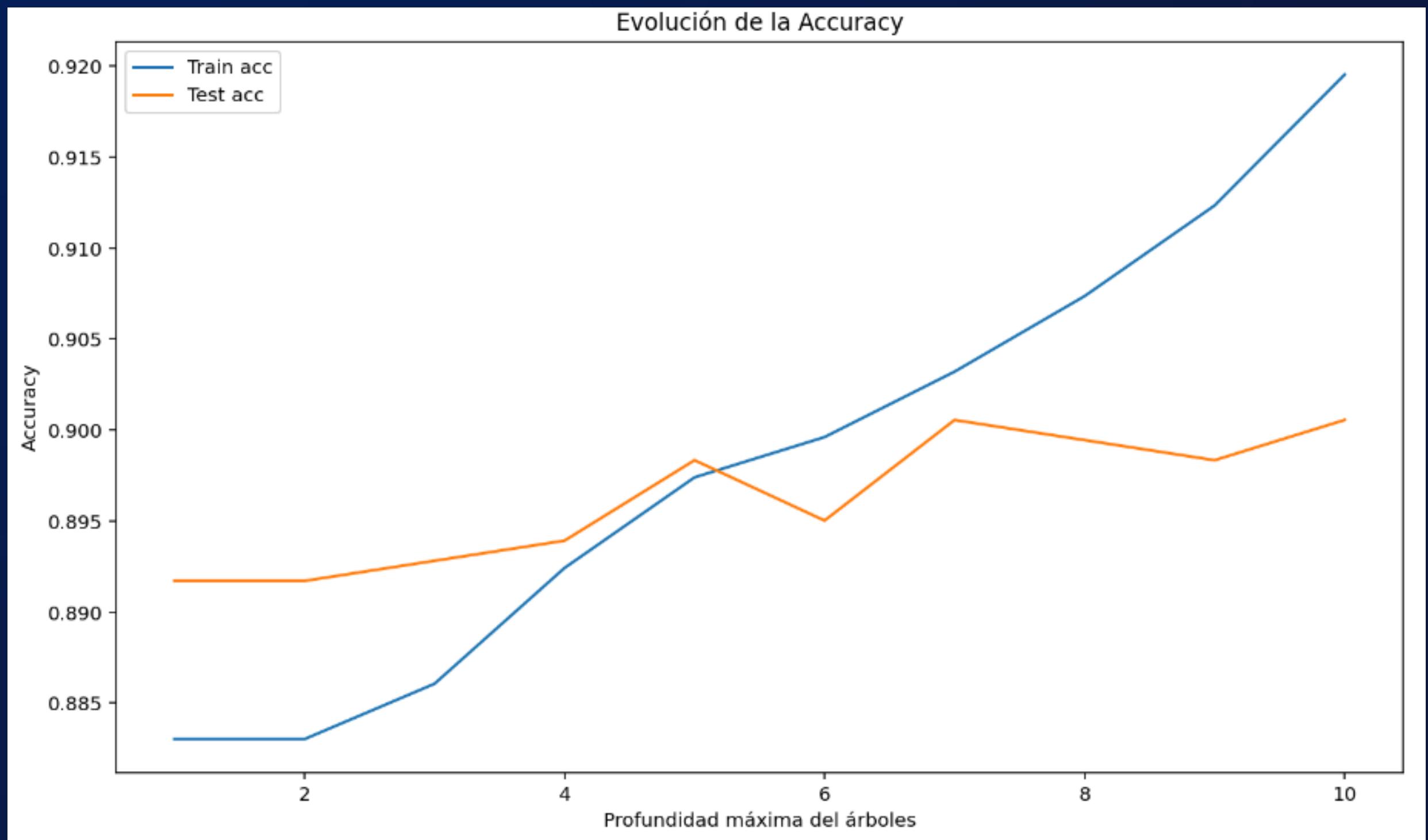


n_trees	acc_train	acc_test
0	1	0.891040
1	10	0.894912
2	25	0.897400
3	50	0.897954
4	100	0.898230
5	250	0.898230

Tomamos n_estimators = 25.

Explore el mejor valor de max_depth

Entrenado con train (80%)
Evaluado con test (20%)

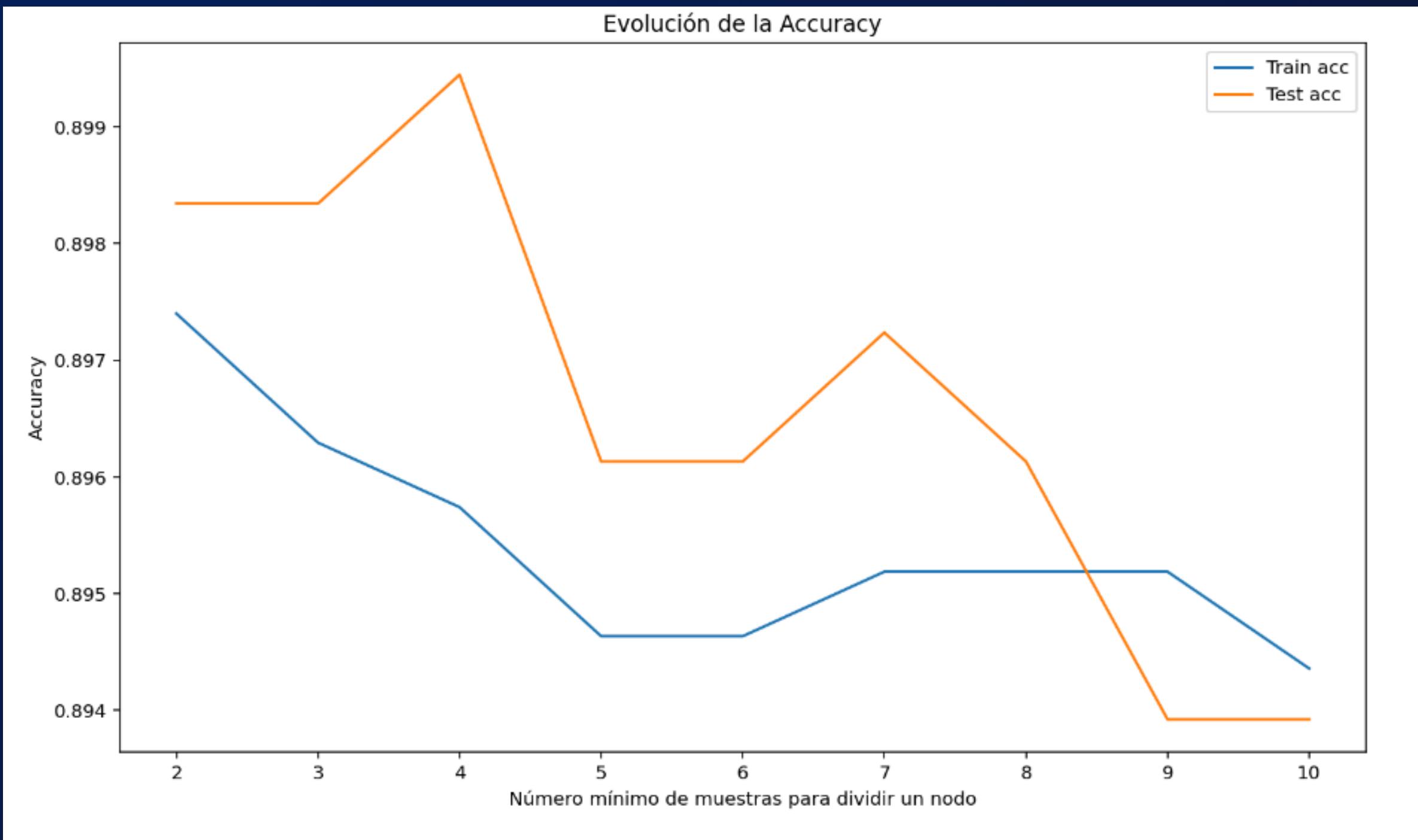


max_depth	acc_train	acc_test
0	0.883020	0.891713
1	0.883020	0.891713
2	0.886062	0.892818
3	0.892423	0.893923
4	0.897400	0.898343
5	0.899613	0.895028
6	0.903208	0.900552
7	0.907356	0.899448
8	0.912334	0.898343
9	0.919524	0.900552

Tomamos max_depth = 5.

Explore el mejor valor de min_samples_split

Entrenado con train (80%)
Evaluado con test (20%)

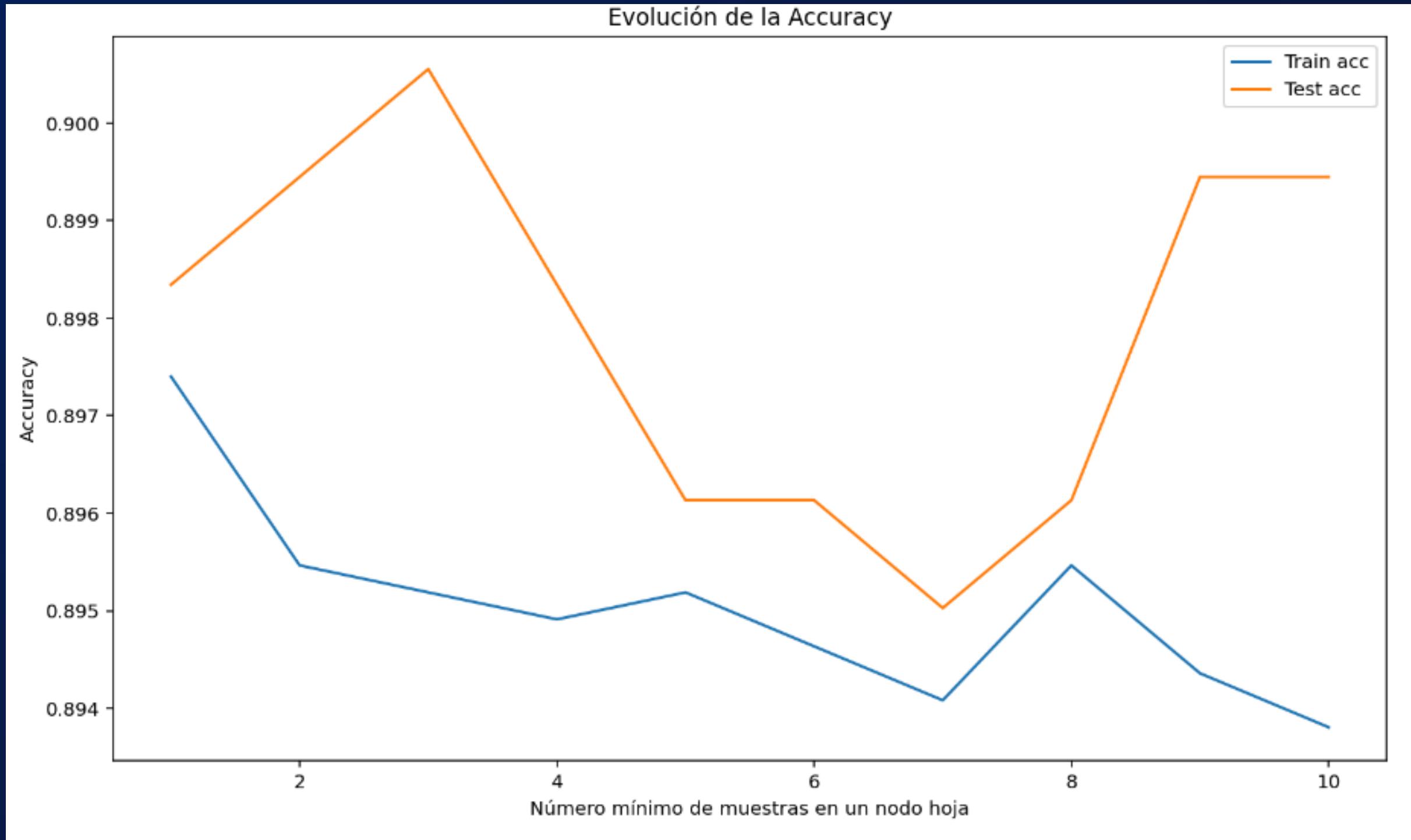


min_samples_split	acc_train	acc_test
0	0.897400	0.898343
1	0.896294	0.898343
2	0.895741	0.899448
3	0.894635	0.896133
4	0.894635	0.896133
5	0.895188	0.897238
6	0.895188	0.896133
7	0.895188	0.893923
8	0.894358	0.893923

Tomamos min_samples_split = 2.

Explore el mejor valor de min_samples_leaf

Entrenado con train (80%)
Evaluado con test (20%)

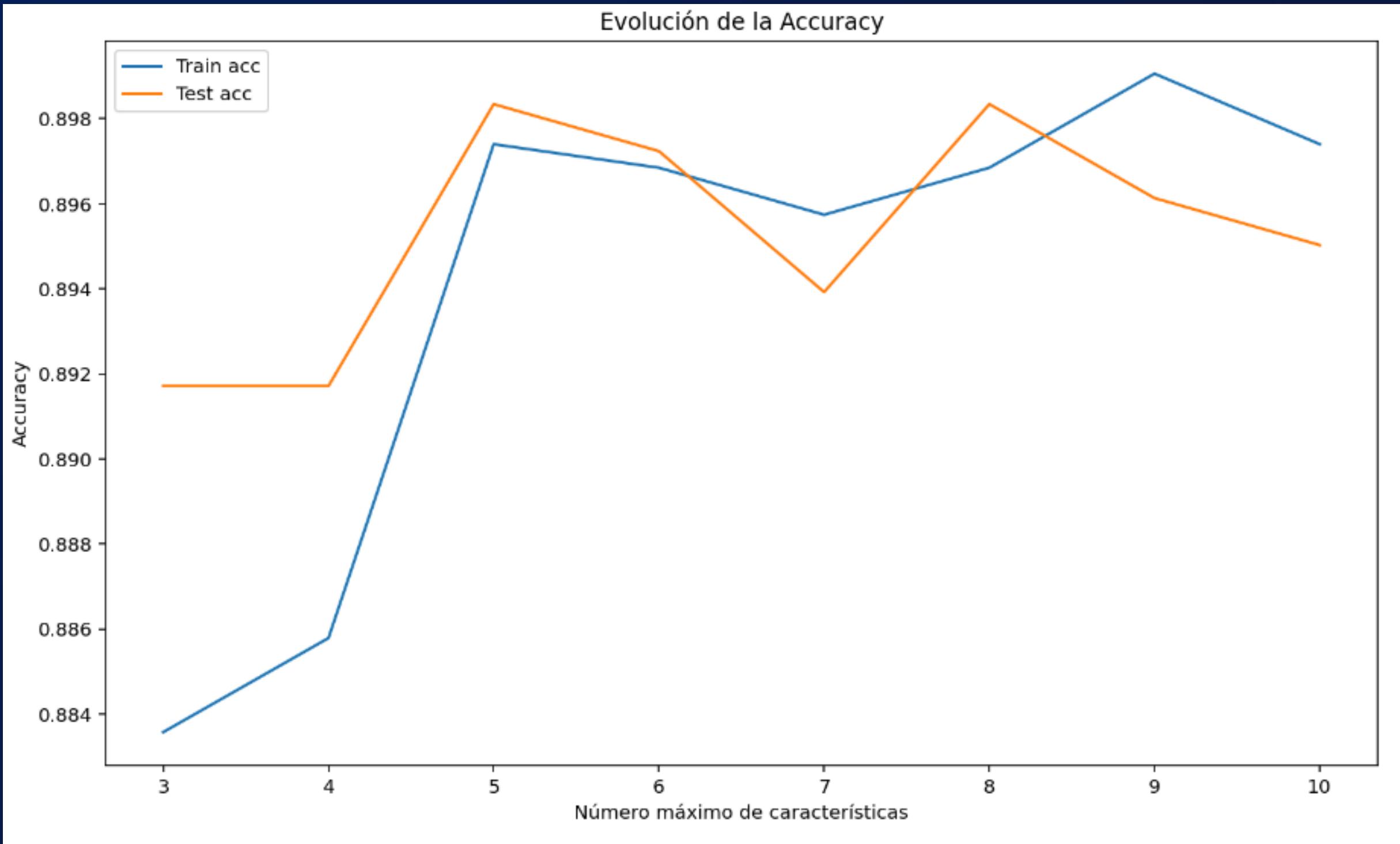


min_samples_leaf	acc_train	acc_test
0	1	0.897400 0.898343
1	2	0.895465 0.899448
2	3	0.895188 0.900552
3	4	0.894912 0.898343
4	5	0.895188 0.896133
5	6	0.894635 0.896133
6	7	0.894082 0.895028
7	8	0.895465 0.896133
8	9	0.894358 0.899448
9	10	0.893805 0.899448

Tomamos min_samples_leaf = 1.

Explore el mejor valor de max_features

Entrenado con train (80%)
Evaluado con test (20%)



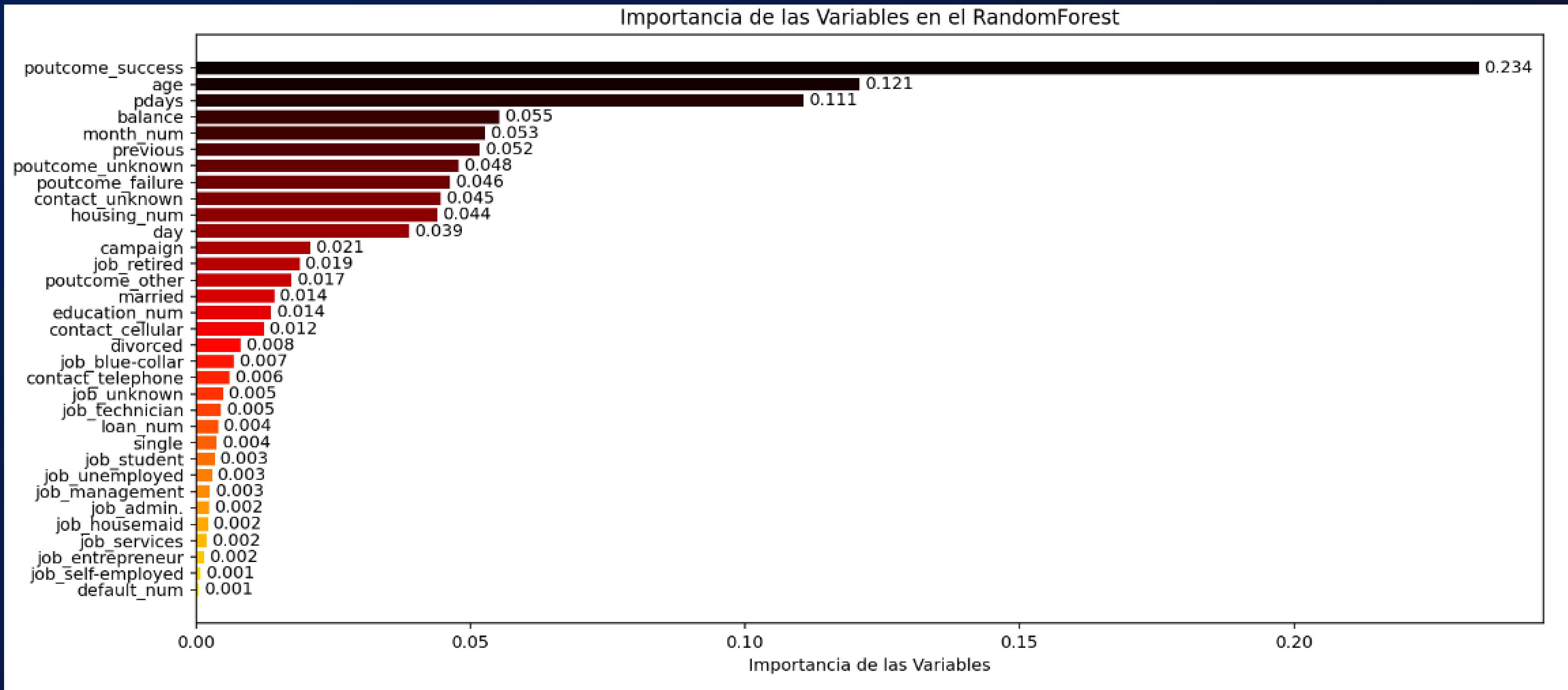
max_features	acc_train	acc_test
0	0.883573	0.891713
1	0.885785	0.891713
2	0.897400	0.898343
3	0.896847	0.897238
4	0.895741	0.893923
5	0.896847	0.898343
6	0.899060	0.896133
7	0.897400	0.895028

Tomamos max_features=5.

Modelo: Random Forest

```
#n_estimators=25, max_depth=5, min_samples_split=2, min_samples_leaf=1, max_features=5
best_model_c = RandomForestClassifier(
    n_estimators=25,
    max_depth=5,
    min_samples_split=2,
    min_samples_leaf=1,
    max_features=5,
    bootstrap=True,
    oob_score=False,
    n_jobs=multiprocessing.cpu_count()
    ,random_state=42 #semilla
)
best_model_c.fit(data[features], data.target)
```

Importancia de las variables



Evaluación del modelo

Accuracy = 0.9

Alta, pero puede ser engañosa en caso de clases desequilibradas.

Tabla de confusión

	Pred NO	Pred YES	TOTAL
True NO	803	4	807
True YES	88	10	98
TOTAL	891	14	905

Entrenado con train (80%)

Evaluado con test (20%)

Threshold: 0.5

Ratio Falsos Positivos:

0.285714286

Ratio Falsos Negativos:

0.098765432

Kappa = 0.16

Concordancia pobre, una mejora pequeña sobre el azar.

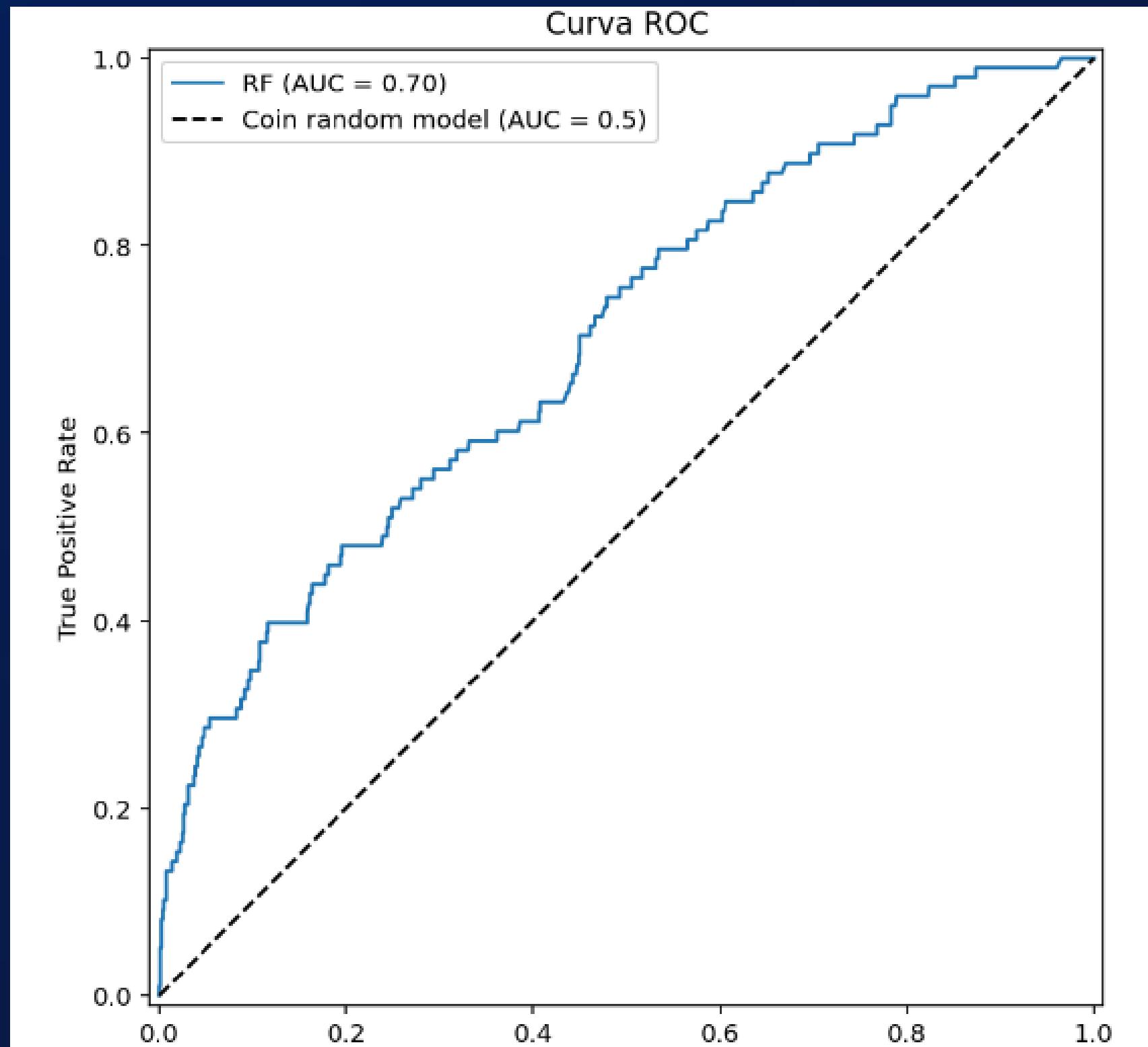
Precisión = 0.71

Recall = 0.10

F1 Score = 0.18

Mal equilibrio entre precisión y recall.

Curva ROC y AUC



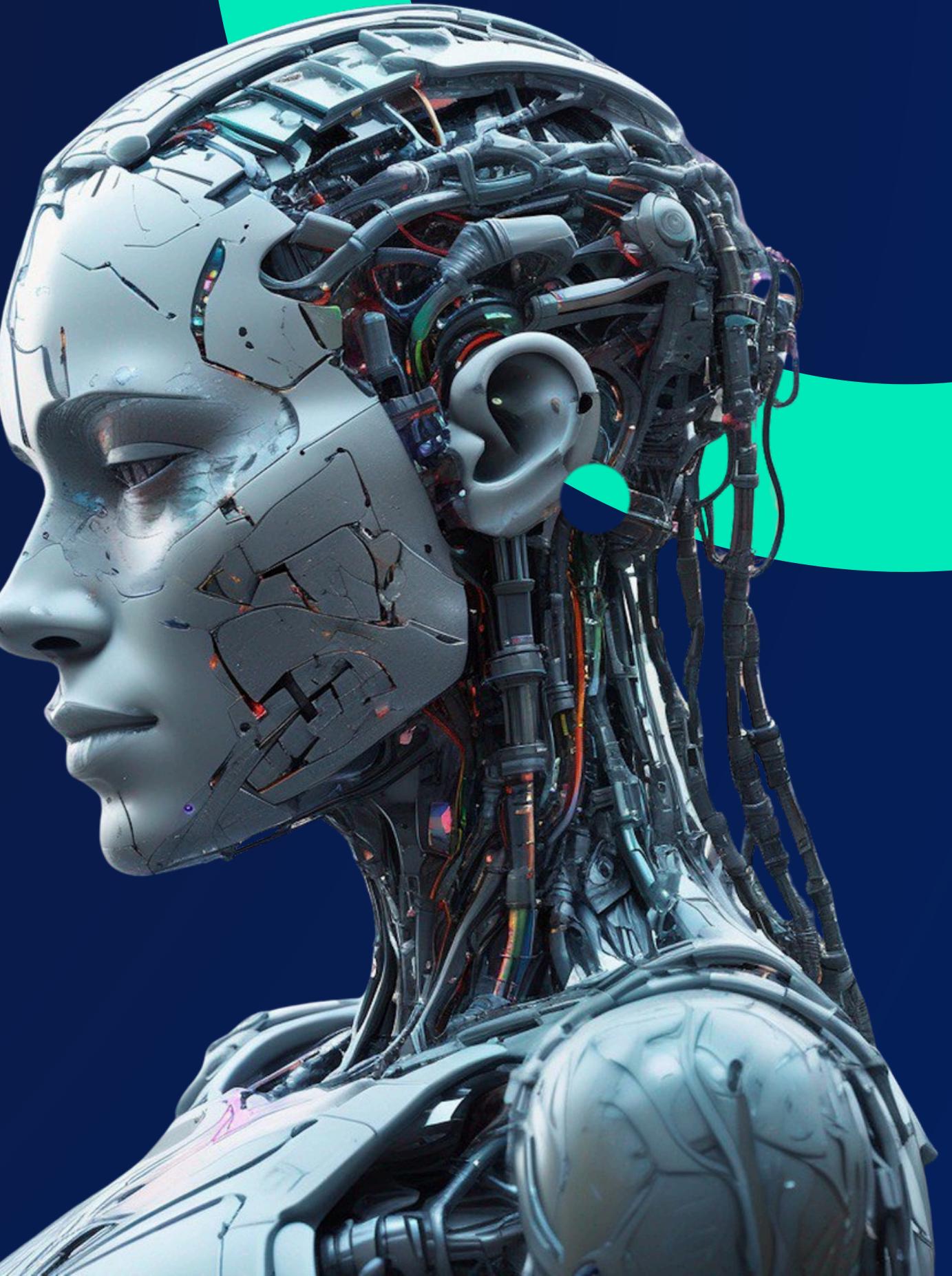
- $AUC = 0.6993526 > 0.5$
- El modelo tiene un rendimiento decente en la distinción entre clases positivas y negativas, pero podría haber margen de mejora.

XGBOOST

Machine Learning

- El número de árboles que tomamos es de 1000 árboles.
- Profundidad máxima de cada árbol: 10 niveles.
- Tomamos una tasa de aprendizaje de 0.001.
- Utilizamos el número máximo de núcleos para ejecutar los trabajos en paralelo para el ajuste y la predicción.

× × ×



Importancia de las variables



Evaluación del modelo

Accuracy
0.90507%

Tabla de confusión

	Pred NO	Pred YES	TOTAL
True NO	404	2	406
True YES	41	6	47
TOTAL	445	8	453

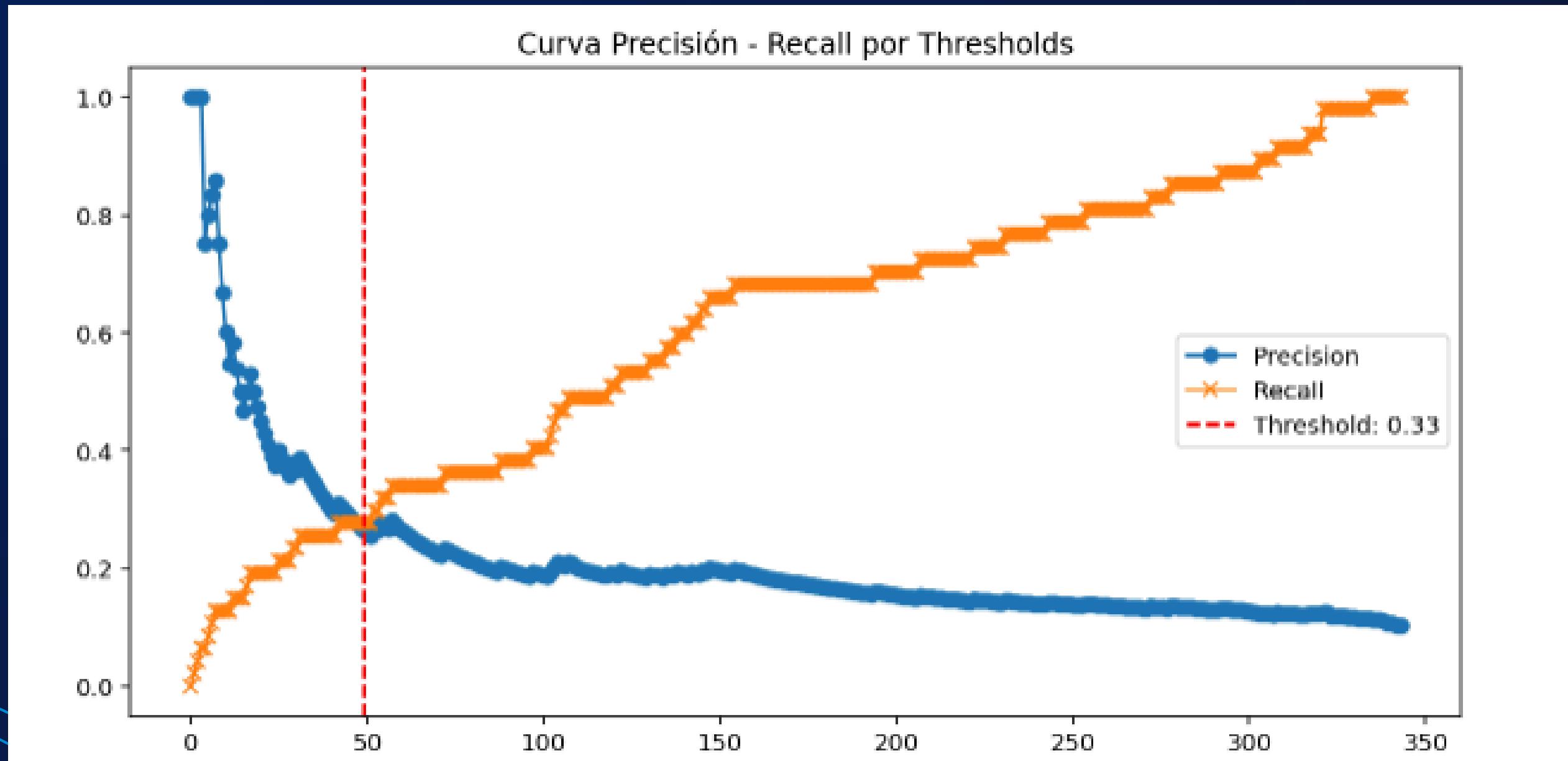
Entrenado con
train (70%)
Evaluado con
test (10%)
Threshold: 0.5

Precision
0.75%
Sensibilidad
0.127659%

F1-Score
0.21818

ROC AUC
0.7%

Curva Precisión-Recall



Finalmente tomamos un threshold de 0.33

Evaluación del modelo

Accuracy
0.849%

Tabla de confusión

	Pred NO	Pred YES	TOTAL
True NO	404	2	406
True YES	41	6	47
TOTAL	445	8	453

Entrenado con
train (70%)
Evaluado con
test (10%)
Threshold: 0.33

Precision
0.27%
Sensibilidad
0.27659%

F1-Score
0.21818

ROC AUC
0.7%

Overfitting

```
# Exhaustividad (recall)
recall_score(target_test, target_hat_test_proba >= 0.33)
```

0.2765957446808511

```
# Exhaustividad (recall)
recall_score(target_train, target_hat_train_proba>=0.33)
```

1.0

- Se observa que el recall utilizando el conjunto de entrenamiento es de 1. Sin embargo, en el conjunto test es de 0.27. Esto es un claro ejemplo de que el modelo se ajusta demasiado bien a los datos de entrenamiento.

Conclusiones



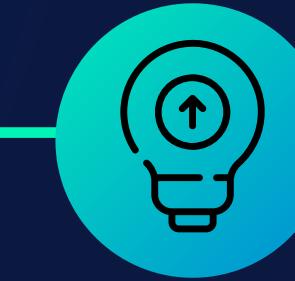
Regresión Logística

- Accuracy = 0.84
- Precisión = 0.2755
- Recall = 0.2755



Decision Tree

- Accuracy = 0.89
- Precisión = 0.4705
- Recall = 0.1632



Random Forest

- Accuracy = 0.9
- Precisión = 0.71
- Recall = 0.1



XGBoost

- Accuracy = 0.85
- Precisión = 0.27
- Recall = 0.27

XXX

XX

NOS QUEDAMOS CON
EL MODELO DE
RANDOM FOREST

Tiene mejor precisión

MÉTRICAS POR PERCENTIL

	TopPercentile	ScoreAvg	TotalCount	Class1Count	Class0Count	TotalCountCumulated	Class1CountCumulated	Class0CountCumulated	PrecisionCumulated	RecallCumulated
0	1	0.599728	45	42	3	45	42	3	0.933333	0.080614
1	2	0.503420	45	29	16	90	71	19	0.788889	0.136276
2	3	0.411826	45	27	18	135	98	37	0.725926	0.188100
3	4	0.328791	45	22	23	180	120	60	0.666667	0.230326
4	5	0.280923	46	23	23	226	143	83	0.632743	0.274472
...
95	96	0.049907	46	0	46	4340	521	3819	0.120046	1.000000
96	97	0.048699	45	0	45	4385	521	3864	0.118814	1.000000
97	98	0.047276	45	0	45	4430	521	3909	0.117607	1.000000
98	99	0.045389	45	0	45	4475	521	3954	0.116425	1.000000

PrecisionCumulated	RecallCumulated	SpecificityCumulated	Class1TotalRatio	Class1TrivialUplift	Class1ModelUplift
0.933333	0.080614	0.00075	0.11524	1.0	8.099040
0.788889	0.136276	0.00475	0.11524	1.0	6.845617
0.725926	0.188100	0.00925	0.11524	1.0	6.299254
0.666667	0.230326	0.01500	0.11524	1.0	5.785029
0.632743	0.274472	0.02075	0.11524	1.0	5.490658
...
0.120046	1.000000	0.95475	0.11524	1.0	1.041705
0.118814	1.000000	0.96600	0.11524	1.0	1.031015

Uplift: ratio de mejora de la Precisión del modelo versus el trivial

En los top N percentil
hay una gran mejora
de la precisión

SIN EMBARGO, EN EL
XGBOOST TENEMOS
MEJOR RECALL

- Accuracy = 0.85
- Precisión = 0.27
- Recall = 0.27

CON ESTE MODELO PREDICIMOS UN 27% DE LOS
CLIENTES QUE SE SUSCRIBEN CORRECTAMENTE

PERFIL DE CLIENTE Y DETALLES DE LA CAMPAÑA

- Cliente cuya anterior campaña haya tenido éxito
- Cliente con educación superior
- Cliente jubilado
- Campaña en marzo u octubre
- Clientes mayores de 60 años

EVITAR SIGUIENTES PERFILES Y DETALLES DE LA CAMPAÑA:

- Contacto desconocido
- Campaña en enero, mayo, julio, noviembre
- Cliente con préstamo hipotecario
- Cliente con préstamo personal
- Cliente cuya anterior campaña haya fracasado
- Cliente casado
- Clientes entre 30 y 50 años



Muchas Gracias!!