Análisis y Selección de Modelo Predictivo para Churn de Clientes

- Autor: Miguel Ángel Toro Romo

- Fecha: 08-08-2025

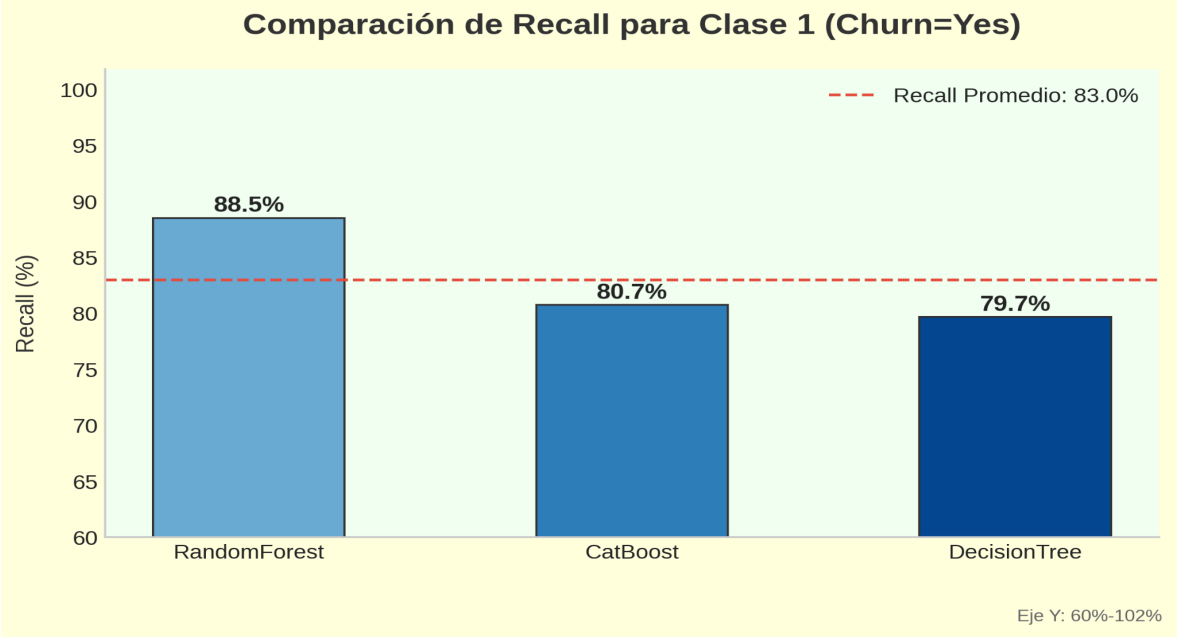
ÍNDICE

Resumen Ejecutivo

- **Objetivo**: Este proyecto tiene como objetivo construir un modelo de clasificación que prediga si un cliente abandonará o no los servicios de la compañía, a partir de sus características sociodemográficas y de los servicios contratados.
- La empresa quiere anticiparse al problema de la cancelación, y debemos construir un pipeline robusto para esta etapa inicial de modelado.
- **Conclusiones clave**: El modelo ganador fue RandomForest optimizado, su **recall** para clase 1 es de 88.5% y un **accuracy** de 68.5%.
- **Beneficio esperado**: Una reducción del 25% en la tasa de abandono podría significar un incremento de ingresos de $950.000 anuales.
- **Métricas claves:**

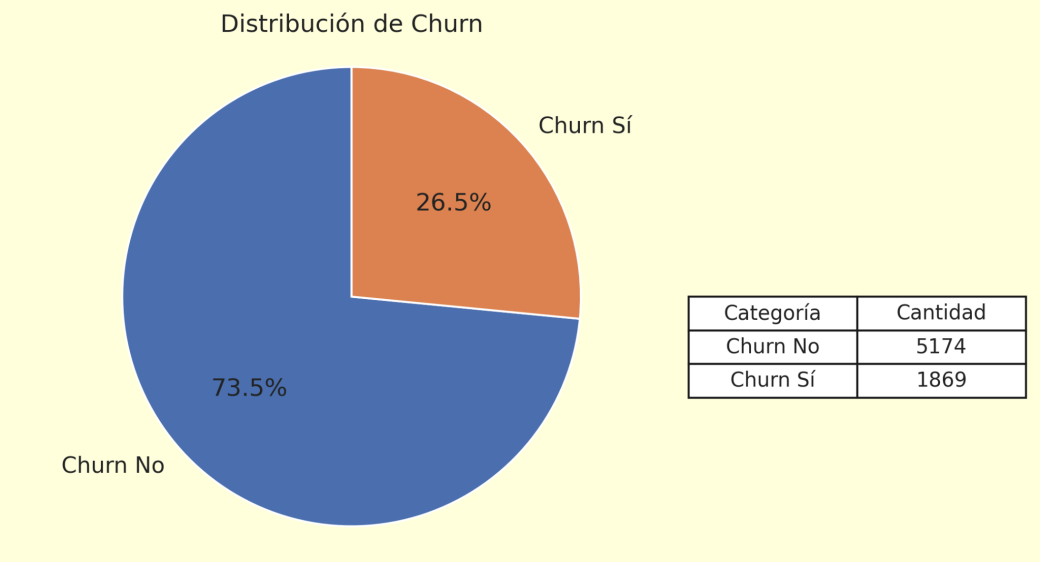
Tabla Comparativa de Modelos:

Modelo	Accuracy	Precision_Clase_1	Recall_Clase_1	F1-Score_Clase_1
DecisionTree	72.69%	49.12%	79.68%	60.77%
CatBoost	75.44%	52.43%	80.75%	63.58%
RandomForest	68.49%	45.22%	88.50%	59.86%



Contexto y Alcance

- Telecom X desea mejorar la retención de clientes:
    - El tema del abandono de clientes - lo que llamamos 'churn' - es un indicador clave de la salud del negocio.
    - Cuando los clientes se van, no solo se pierden ingresos estables mes a mes. Además tiene un costo financiero, porque conseguir un cliente nuevo puede costar bastante más que mantener a uno que ya se tiene.
    - La verdadera ventaja está en adelantarse a la decisión de abandono del cliente. Si se entiende por qué y cuándo se van, se pueden crear estrategias personalizadas de retención antes de que decidan irse.
- Descripción del dataset:
    - Para realizar este análisis, Telecom X nos ha entregado un dataset con 7042 registros.
    - La distribución de clientes que han abandonado (Churn Sí) y los que permanecen (Churn No) es como muestra el siguiente gráfico:

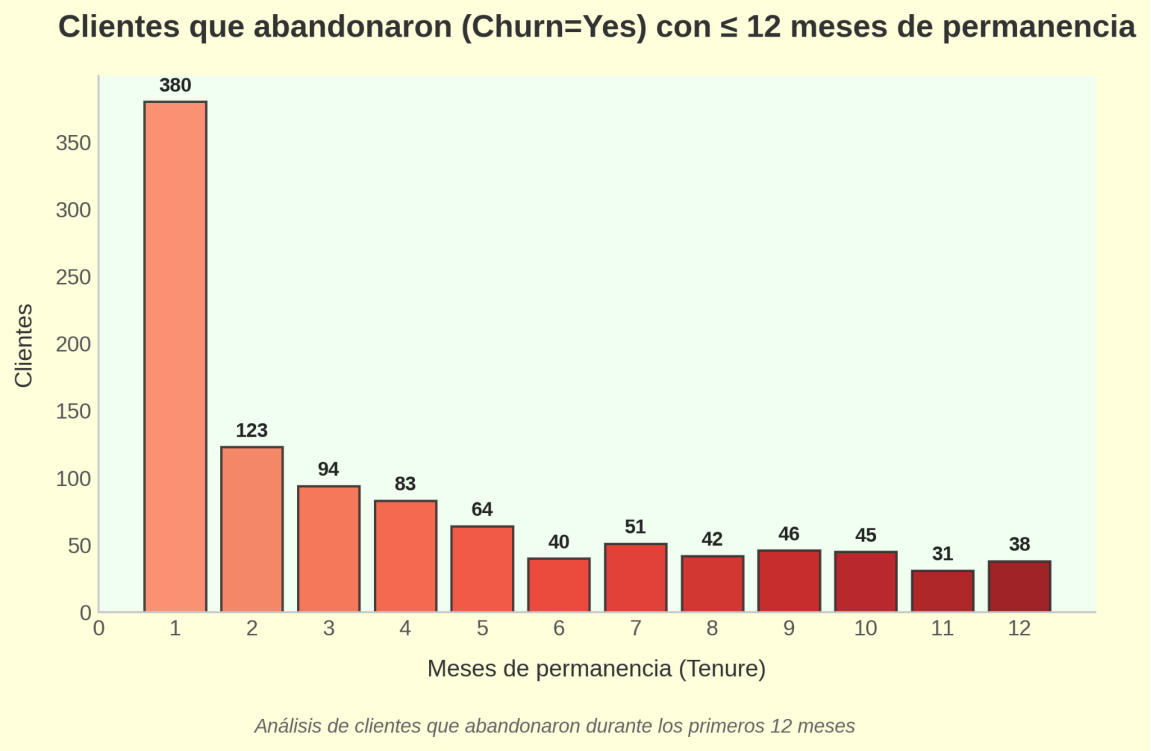


Categoría	Cantidad
Churn No	5174
Churn Sí	1869

    - Como se puede apreciar, existe un desbalance importante entre las clases Churn Sí y Churn No.

- Las variables del set de datos y sus valores posibles son:
    - customerID: identificación alfanumérica del cliente
    - Churn: ['No' 'Yes'] (Yes si el cliente abandonó)
    - gender: ['Female' 'Male']
    - SeniorCitizen: [0 1]
    - Partner: ['Yes' 'No']
    - Dependents: ['Yes' 'No']
    - tenure: cantidad de meses que tiene o tuvo en la compañía
    - PhoneService: ['Yes' 'No']
    - MultipleLines: ['No' 'Yes' 'No phone service']
    - InternetService: ['DSL' 'Fiber optic' 'No']
    - OnlineSecurity: ['No' 'Yes' 'No internet service']
    - OnlineBackup: ['Yes' 'No' 'No internet service']
    - DeviceProtection: ['No' 'Yes' 'No internet service']
    - TechSupport: ['Yes' 'No' 'No internet service']
    - StreamingTV: ['Yes' 'No' 'No internet service']
    - StreamingMovies: ['No' 'Yes' 'No internet service']
    - Contract: ['One year' 'Month-to-month' 'Two year']
    - PaperlessBilling: ['Yes' 'No']
    - PaymentMethod: ['Mailed check' 'Electronic check' 'Credit card (automatic)'
    - 'Bank transfer (automatic)']
    - Charges.Monthly: facturación mensual del cliente
    - Charges.Total: facturación acumulada del cliente

- El abandono según los meses de permanencia en el primer año es el siguiente:



  - Se puede ver que la mayor fuga de clientes se produce en los primeros 5 meses de permanencia, concentrándose en el primer mes una cantidad que más que triplica a la de los meses siguientes.

Metodología

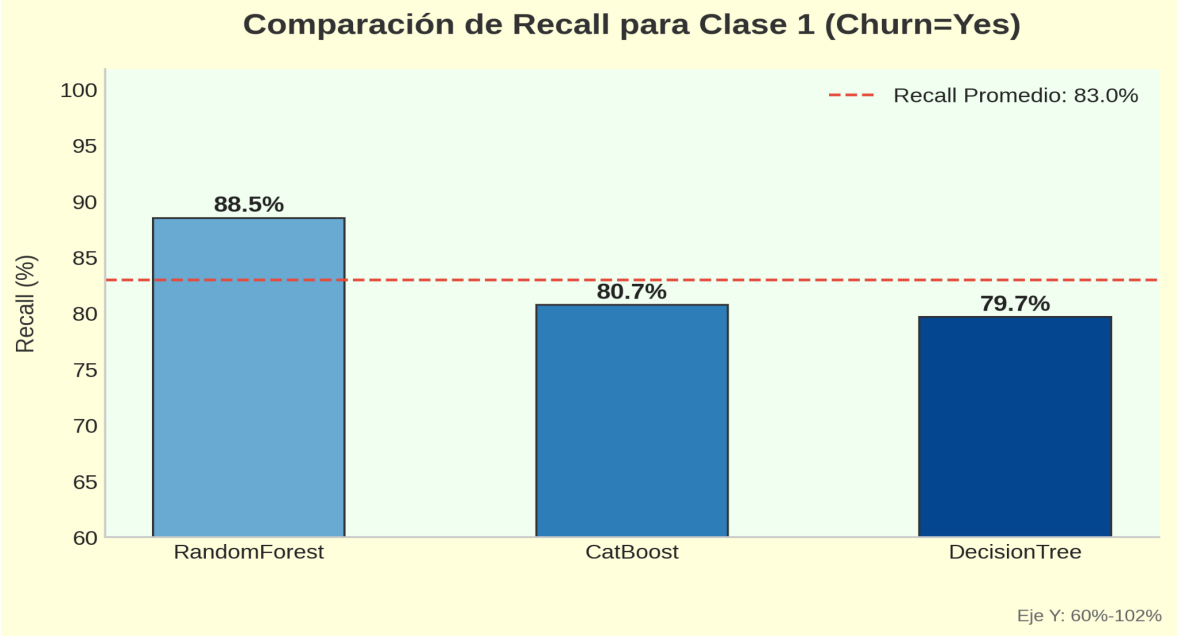
- Preprocesamiento:
    - Eliminación de variables innecesarias:
        - customerID porque es solo un identificador del clientes
        - Charges.Monthly porque su información ya está implícita en Charges.Total.
    - Codificación de variables:
        - Se codificaron las variables binarias ‘Yes’ y ‘No’ a 1 y 0 respectivamente.
        - Se codificaron las columnas 'gender', 'InternetService', 'Contract', 'PaymentMethod' con técnica de recomendada eficiencia.
    - Tratamiento del desbalanceo:
        - Se probaron varias técnicas de balanceo y se optó por RandomOverSampler, por tener los mejores resultados para todos los algoritmos.
- Algoritmos evaluados:
    - DecisionTreeClassifier.
    - CatBoostClassifier.
    - RandomForestClassifier.
- Optimización de hiperparámetros:
    - Método usado (GridSearchCV, RandomizedSearchCV).
    - Métrica objetivo (recall clase 1).
- Validación:
    - División train/test.

Resultados

Comparativa de modelos

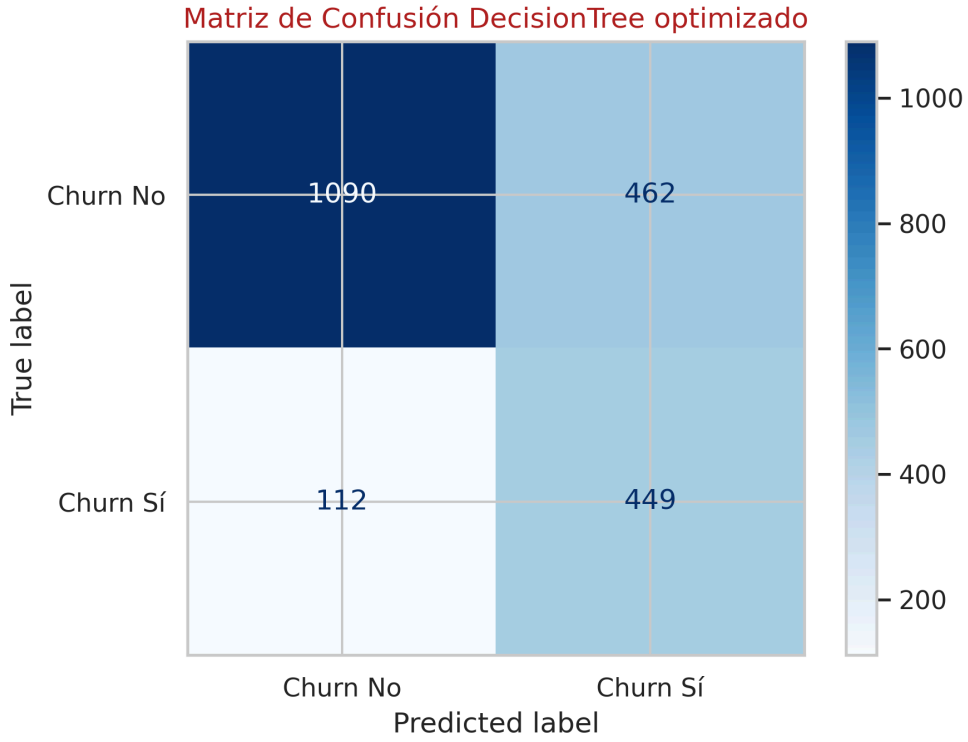
Tabla Comparativa de Modelos:

Modelo	Accuracy	Precision_Clase_1	Recall_Clase_1	F1-Score_Clase_1
DecisionTree	72.69%	49.12%	79.68%	60.77%
CatBoost	75.44%	52.43%	80.75%	63.58%
RandomForest	68.49%	45.22%	88.50%	59.86%

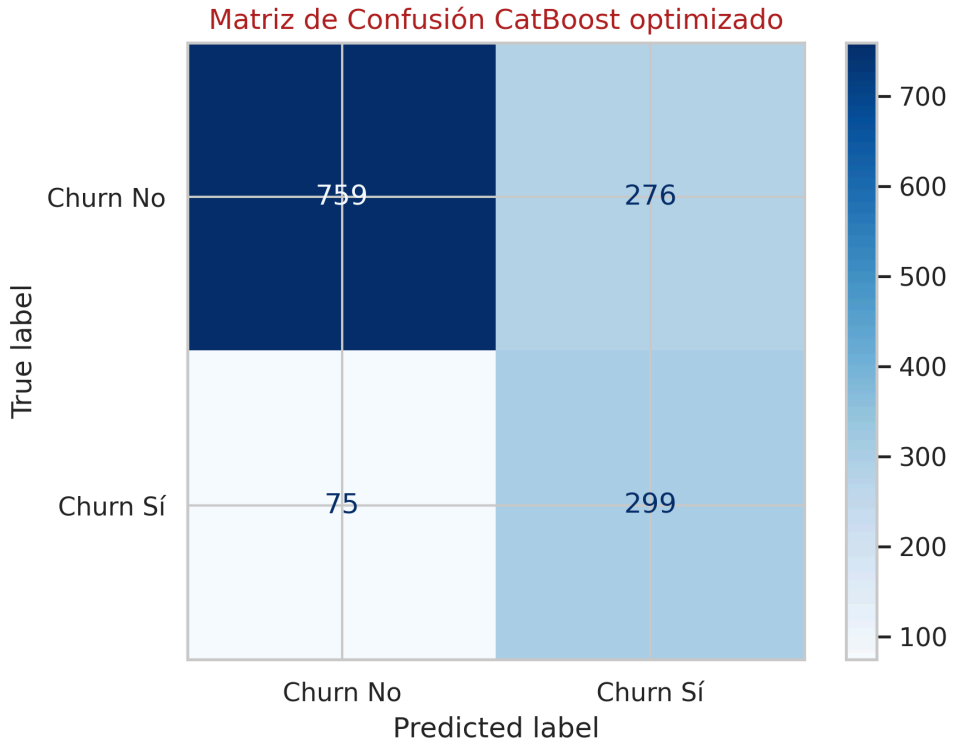


Análisis visual con matrices de confusión de los tres modelos

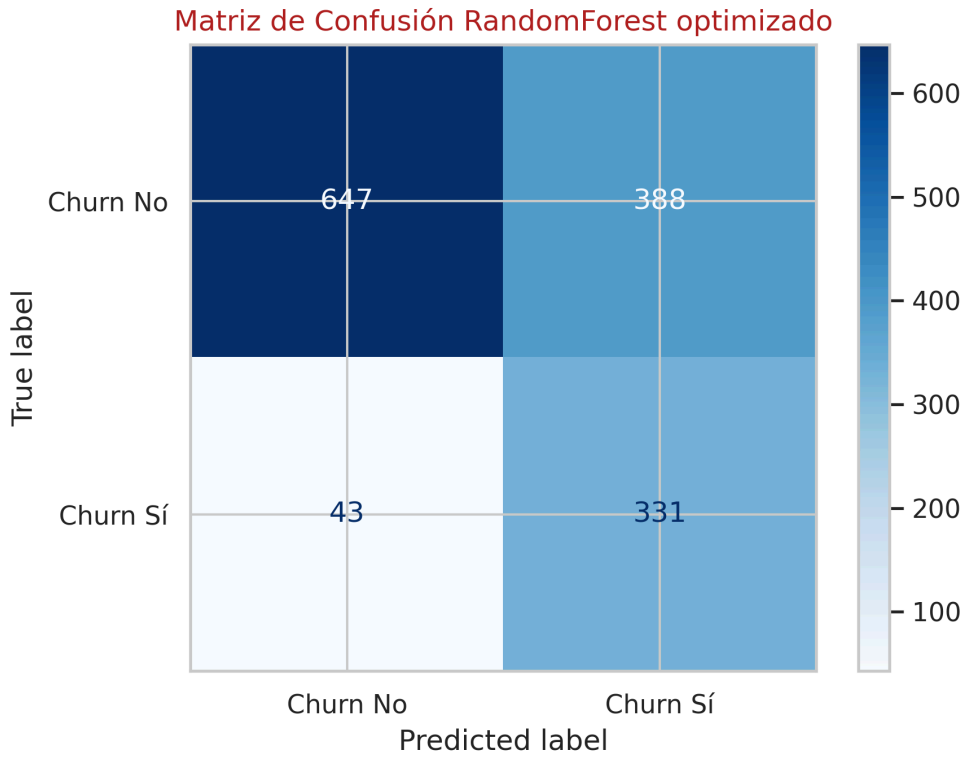
- DecisionTreeClassifier optimizado:



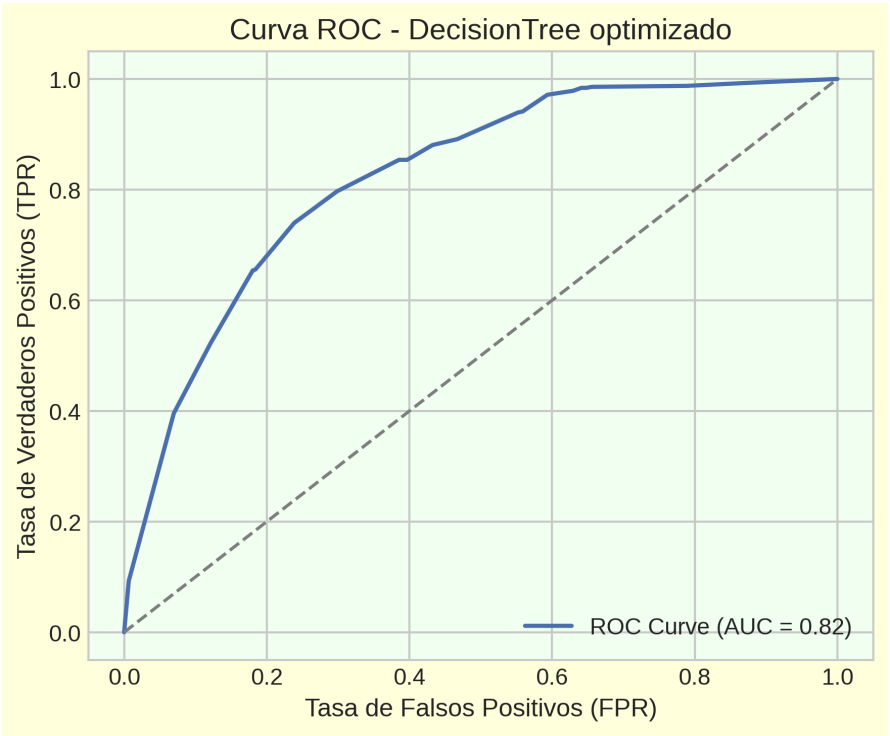
- CatBoost optimizado:

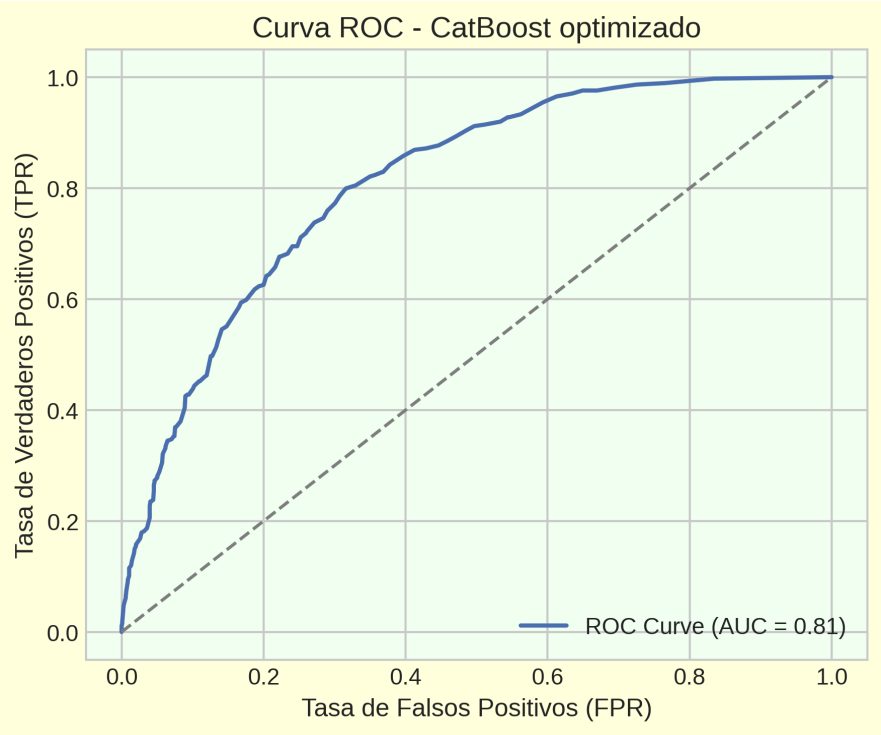


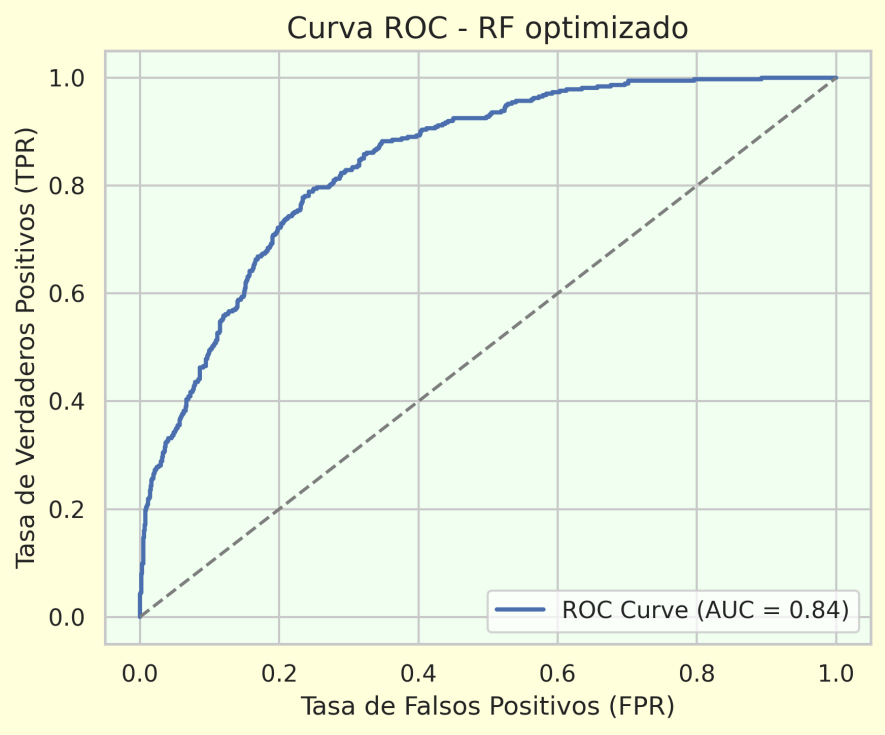
- RandomForest optimizado:



- Curvas ROC:



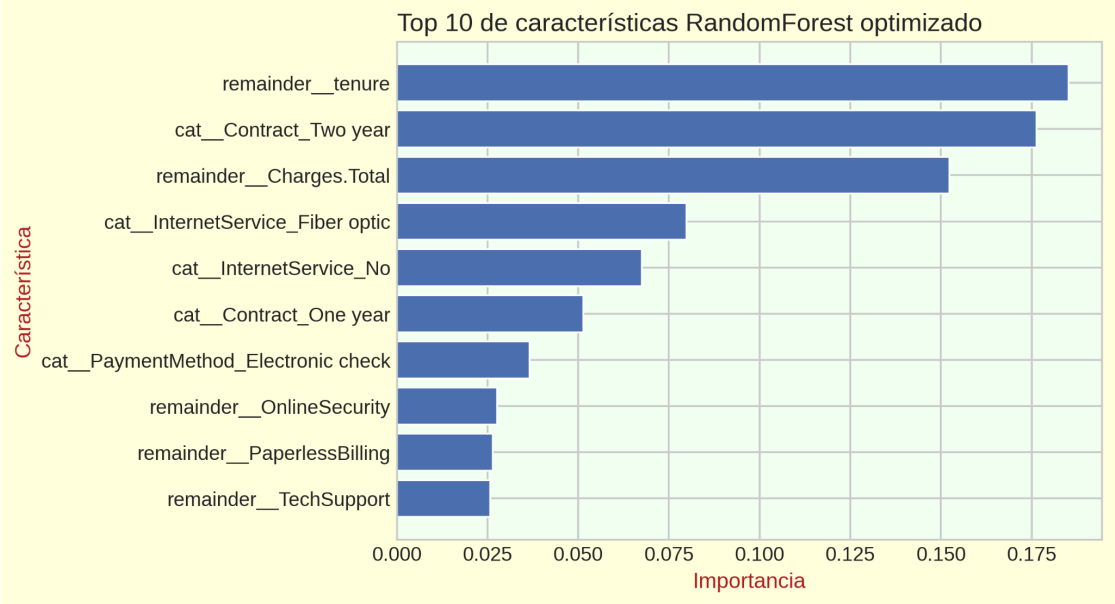

Curva ROC - CatBoost optimizado
Tasa de Verdaderos Positivos (TPR)
Tasa de Falsos Positivos (FPR)
ROC Curve (AUC = 0.81)
0.0
0.2
0.4
0.6
0.8
1.0


Curva ROC - RF optimizado
Tasa de Verdaderos Positivos (TPR)
Tasa de Falsos Positivos (FPR)
ROC Curve (AUC = 0.84)
0.0
0.2
0.4
0.6
0.8
1.0

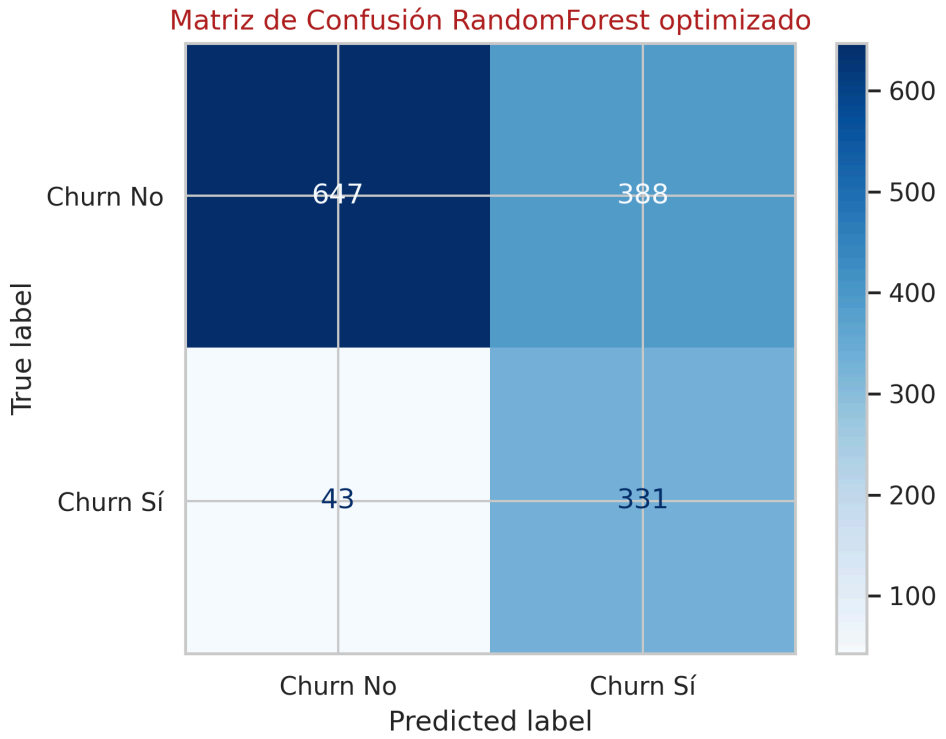
Modelo seleccionado

- Por tener las mejores métricas para el Recall de la clase 1 (Churn Sí) se eligió el modelo generado con **RandomForestClassifier**.

- Importancia de variables (*feature importance*) de este modelo:



- Matriz de confusión final (sobre test).



- Classification report final.

Reporte de clasificación usando el pipeline cargado:				
	precisión	**recall**	**f1-score**	**support**
0	0.94	0.61	0.74	1035
1	0.45	0.89	0.60	374
accuracy			0.69	1409
macro avg	0.70	0.75	0.67	1409
weighted avg	0.81	0.69	0.70	1409

Conclusiones

- El modelo **RandomForest** optimizado tuvo las mejores métricas para el recall de la clase 1 (89%). Sus parámetros de optimización son:

```
parametros_optimizados = {
    'n_estimators': 100,
    'max_depth': 10,
    'min_samples_split': 5,
    'min_samples_leaf': 4,
    'max_features': 'sqrt',
    'bootstrap': True,
    'class_weight': {0: 1, 1: 3},
    'random_state': 42
}
```

- Esto significa que puede predecir correctamente en promedio a un 89% de los clientes susceptibles de abandonar la compañía. Esto es muy importante porque permite enfocar los esfuerzos de retención en el conjunto de interés.

- Principales Limitaciones del Modelo de Churn:

1. **Desbalance de clases**

   A pesar de aplicar técnicas para mitigar la desproporción entre clientes que abandonan y los que permanecen, el sesgo natural en los datos puede afectar la capacidad del modelo para identificar con precisión la clase minoritaria.

2. **Dependencia de la calidad de los datos**

   El modelo está limitado por la exactitud y completitud de la información histórica disponible.

3. **Alcance de las variables**

   El conjunto de variables utilizado refleja principalmente información interna (contratos, servicios, facturación). Factores externos como cambios de mercado, competencia o coyuntura económica no están modelados, lo que podría limitar la capacidad predictiva.

4. **Posible sobreajuste a los datos de entrenamiento**

   Aunque se aplicaron técnicas para reducir el *overfitting*, siempre existe el riesgo de que el modelo se ajuste demasiado a patrones específicos del conjunto de entrenamiento, perdiendo generalización.

5. **Horizonte temporal de predicción**

   El modelo predice la probabilidad de abandono en base al comportamiento histórico, pero no define con exactitud el plazo en que este abandono podría ocurrir.

Consideraciones para producción

- Arquitectura del pipeline de producción:

```
A[Datos nuevos] --> B[Drop Columns]

B --> C[Transform Yes/No]

C --> D[One-Hot Encoding]

D --> E[Modelo RF]

E --> F[Predicción]
```

- Datos nuevos:
  - El pipeline de producción recibe registros con la misma estructura y valores del dataset original. No es necesario hacer transformaciones previas.
- Drop Columns:
  - El pipeline elimina las categorías que se ignoraron para el modelado.
- Transform Yes/No:
  - El pipeline transforma las categorías binarias en 1 y 0.
- One-Hot Encoding:
  - Con esta técnica el pipeline transforma las categorías restantes para obtener un dataset con todas las categorías binarias.
- Modelo RF:
  - Con los datos ya transformados y codificados el pipeline ejecuta el modelo sobre ellos.
- Predicción:
  - Por último, el pipeline entrega la predicción para el cliente o clientes ingresados. Indicando cuál es la probabilidad para que abandone o permanezca en la compañía.