Telecom Churn Predictor

Modelo de detección del churn para empresa de telecomunicaciones

// Technical Presentation //

Objetivo

El objetivo principal de este proyecto es desarrollar un modelo predictivo para identificar la probabilidad de churn o tasa de abandono de clientes, en una empresa de telecomunicaciones.

Un modelo que permita a la empresa implementar estrategias proactivas para retener a los clientes en riesgo de churn, optimizando así la gestión de relaciones con los clientes y mejorando la rentabilidad.

Churn	
0	5163
1	1869

Datos

El punto de partida es un dataset obtenido de Kaggle que contiene datos de una empresa americana de servicios de internet.

El dataset se encontraba en bastante buen estado, con muy pocos NaN o campos vacíos. Sin embargo sí que se observaba un ligero desbalanceo en las clases objetivo.

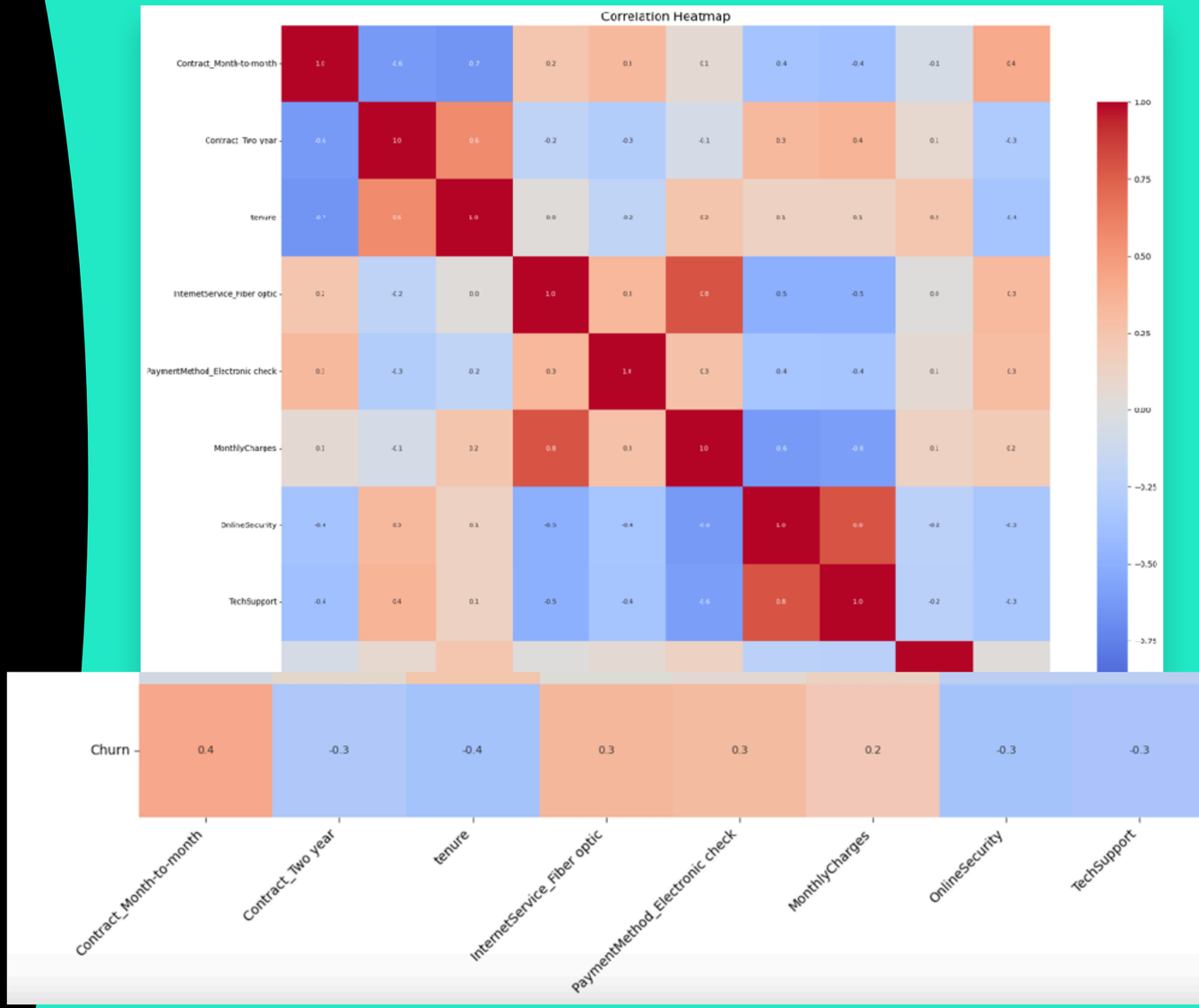
Con 21 variables de origen, tras la conversión de variables categóricas y binarias a formato numérico, el total de variables se amplió a 27.

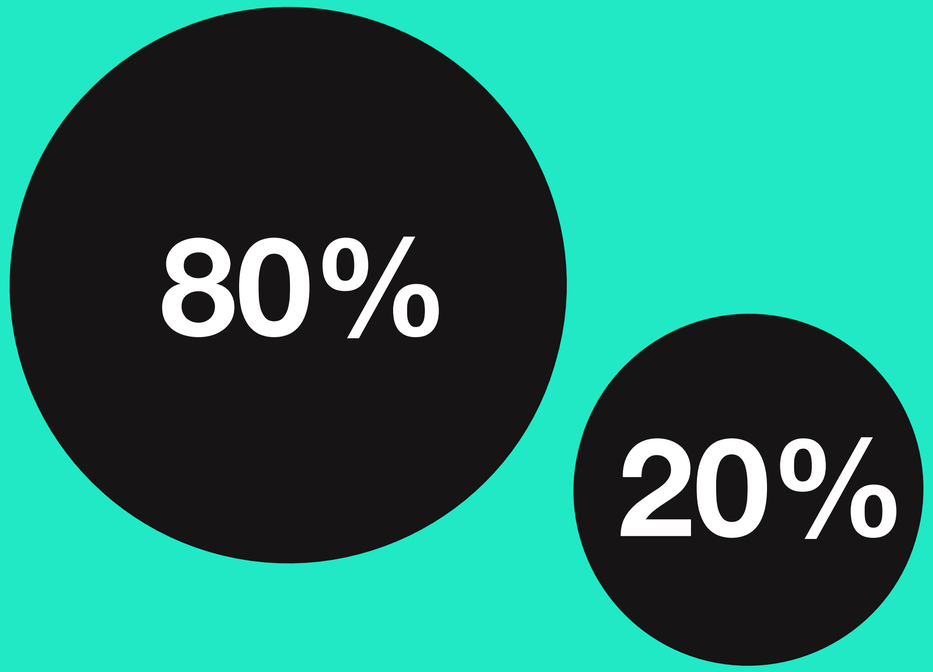
Análisis y feature engineering

Mediante un mapa de correlaciones para analizar las relaciones entre las diferentes variables, decidí qué variables serían incluidas en las pruebas con modelos.

Por otra parte tras percibir posible riesgo de multicolinealidad entre algunas de ellas como ‘tenure’, ‘TotalCharges’ y ‘MonthlyCharges’, decidí eliminar ‘TotalCharges’ debido a su alta correlación con las otras.

Con el resto de variables estuve haciendo pruebas con varios conjuntos diferentes para quedarme finalmente con el conjunto de variables que mejor rendimiento mostraba.





Split de datos

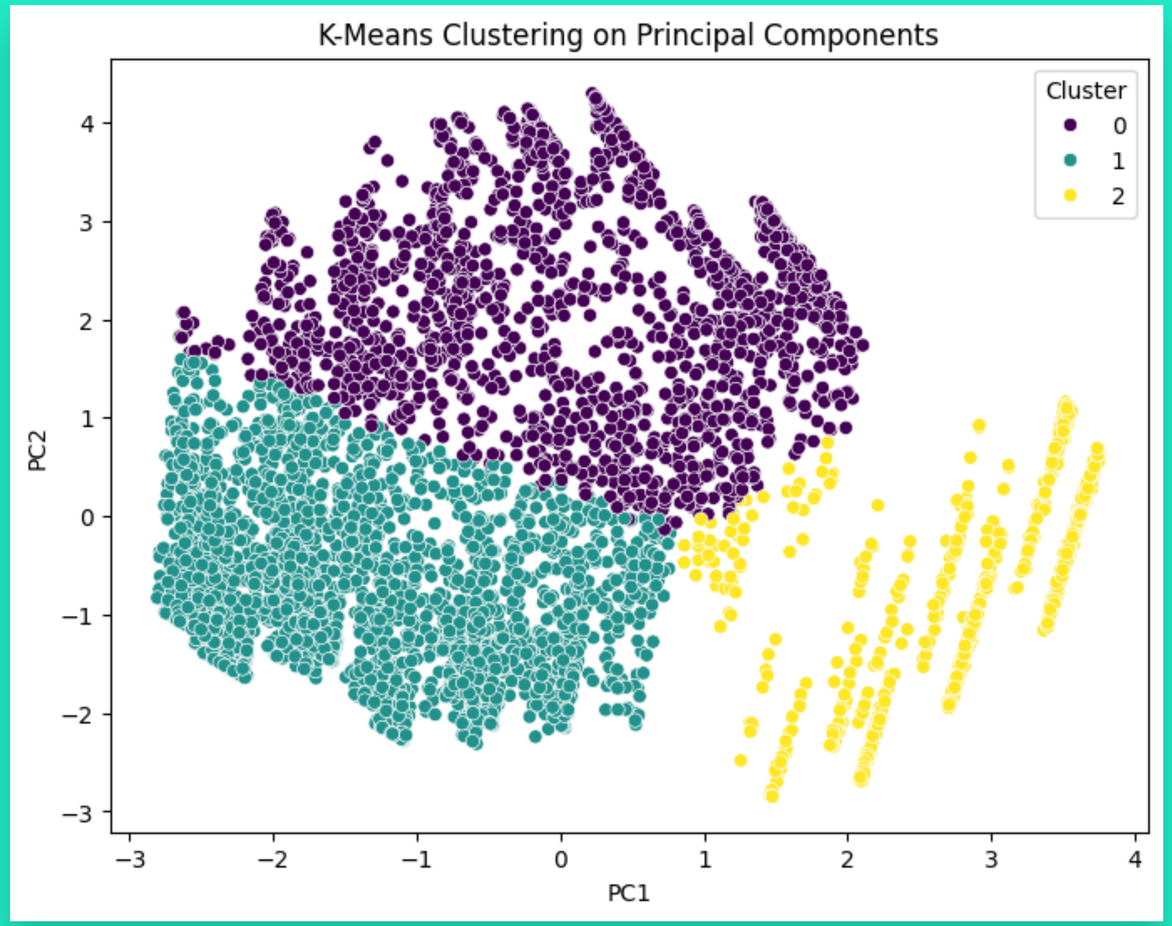
Para el desarrollo y validación del modelo, dividí los datos en conjunto de entrenamiento y conjunto de prueba como es habitual.

De esta manera me aseguro de que el modelo sea entrenado y evaluado de manera que permita medir su capacidad de generalización, evitando problemas de sobreajuste y proporcionando métricas de rendimiento más precisas.

No Supervisado, K-Means

Durante el desarrollo del proyecto, estuve considerando el uso de un modelo no supervisado como K-Means para identificar patrones y segmentar a los clientes en diferentes grupos.

Sin embargo, al evaluar los resultados y la utilidad de los clústeres para la predicción directa del churn, vi que K-Means no proporcionaba la capacidad predictiva necesaria, al menos no sin dedicarle bastante más tiempo, algo de lo que no disponía.



Pruebas con modelos supervisados

Debido a la necesidad de predicciones claras y usables, decidí centrarme en los modelos supervisados.

Modelos que permitieran una evaluación directa del churn mediante métricas específicas que midieran las probabilidades de churn para cada cliente y focalizándome en lograr el mejor recall posible para maximizar la capacidad del modelo de identificar los casos positivos.

Adicionalmente usé GridSearch para ajustar los hiperparámetros en algunos de estos modelos y así tratar de hacerlos más eficientes.

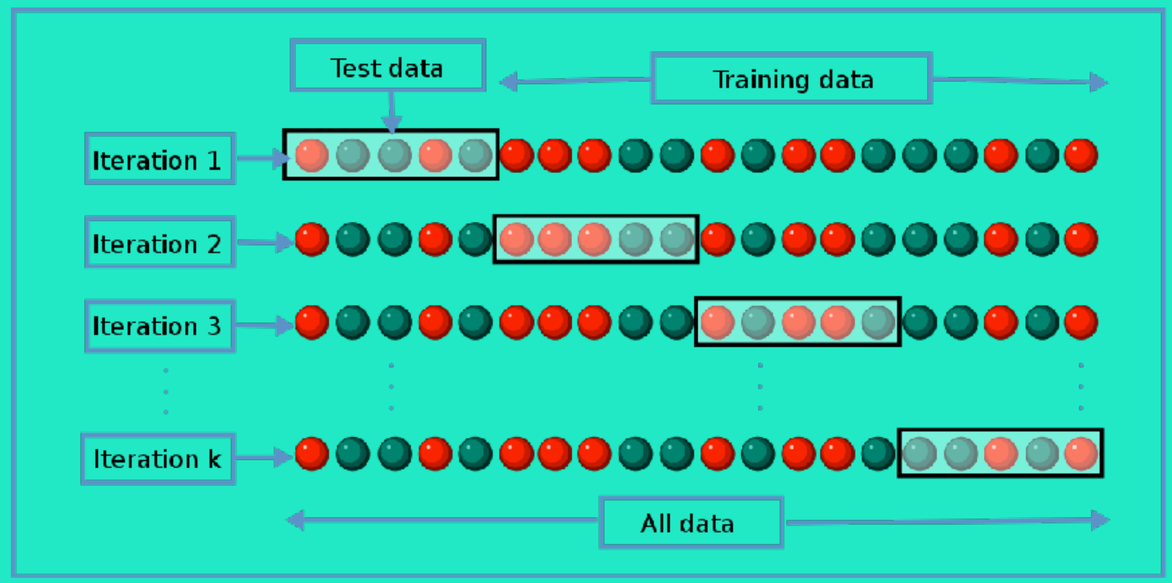
Por otra parte usé SMOTE para el balanceo de clases y StandardScaler para la normalización de las características.

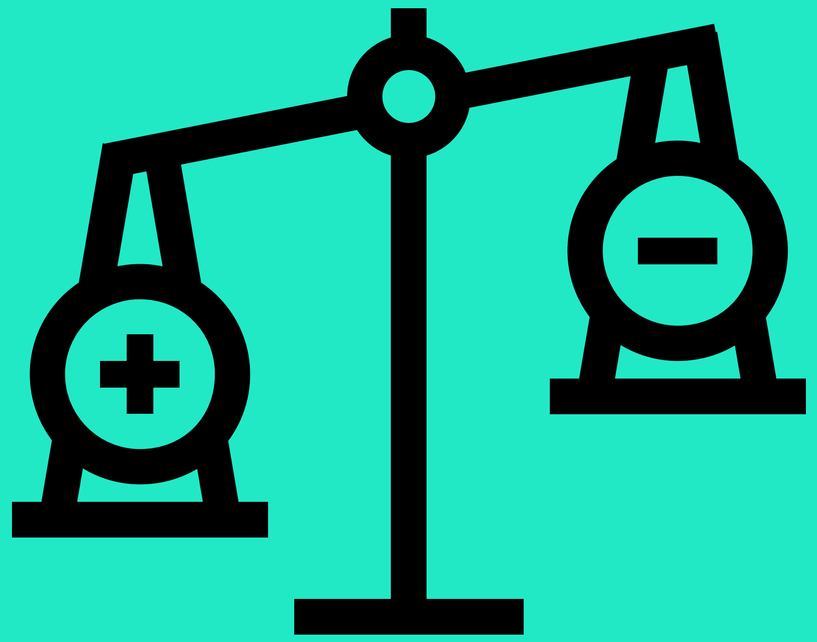
Validación cruzada

Para asegurarme de que los resultados del modelo eran robustos y generalizables, utilicé la validación cruzada para evaluar su capacidad de controlar el overfitting.

En lugar de dejar los datos divididos en un solo conjunto de entrenamiento y otro de prueba, la validación cruzada permite dividir los datos en varios subconjuntos (folds).

De esta manera, el modelo se entrena y evalúa múltiples veces, utilizando diferentes combinaciones de estos subconjuntos. Este enfoque ayuda a evitar el sobreajuste, asegurando que el modelo generalice bien a datos nuevos.





La importancia de detectar positivos

Hay sectores como la salud, donde no detectar un caso positivo puede tener consecuencias graves.

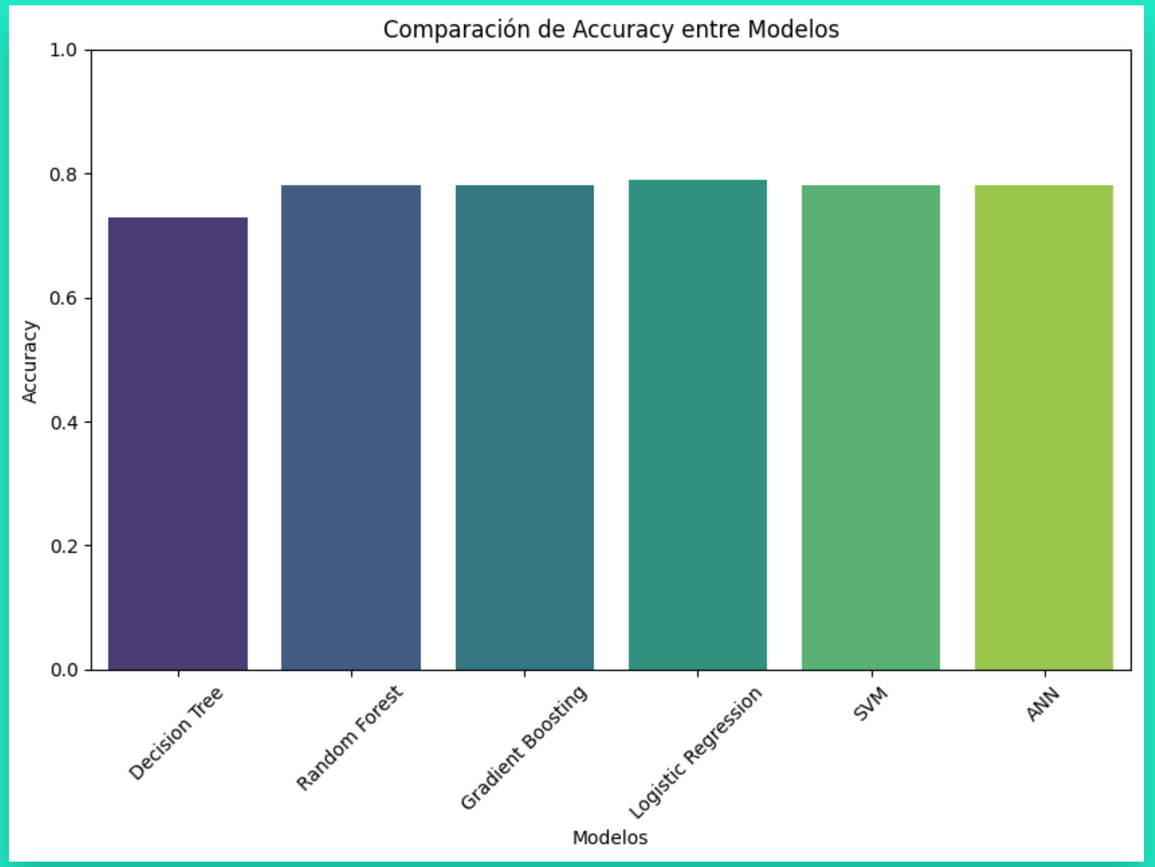
O la seguridad en sistemas de detección de fraude o intrusión, donde es crucial detectar todas las amenazas posibles.

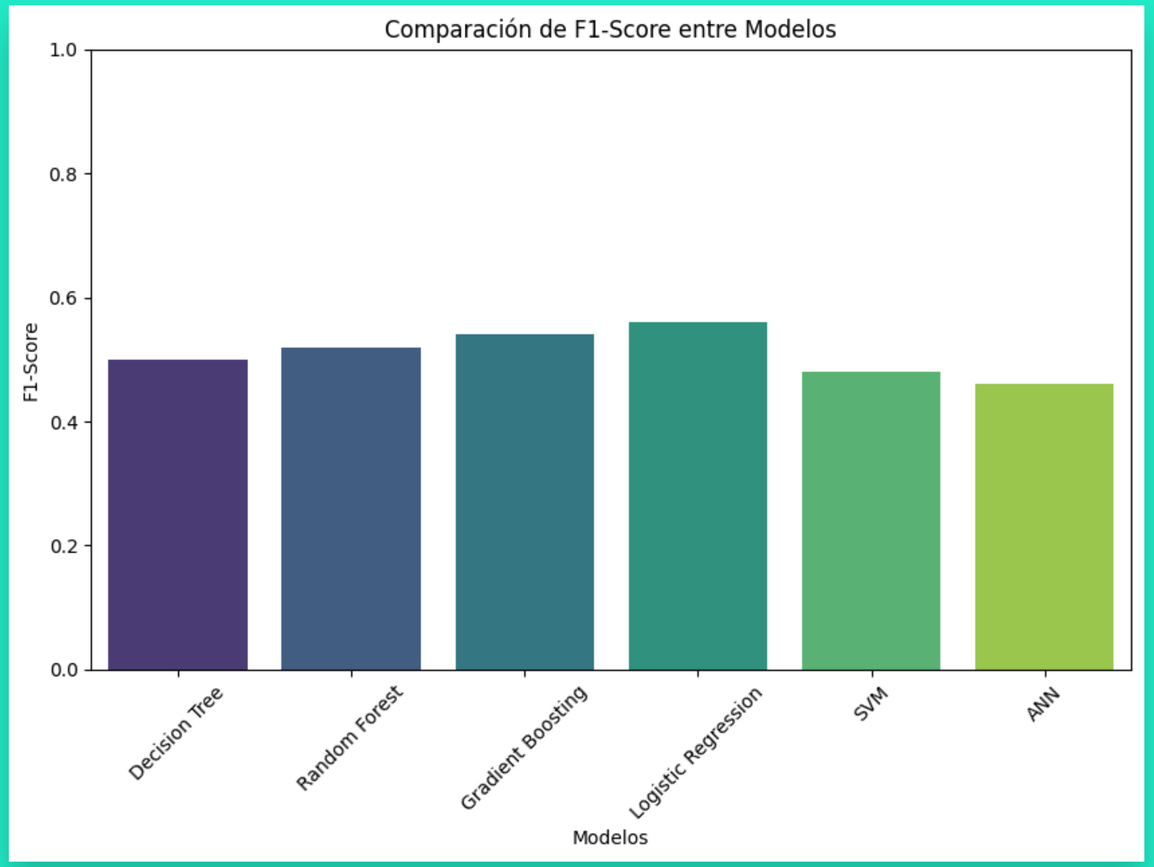
Sin embargo también está el marketing y más concretamente las campañas de retención de clientes (churn), donde es crucial identificar todos los clientes que puedan abandonar el servicio para poder intervenir.

Este es el motivo por el que decidí centrarme en el Recall.

Accuracy

Aunque la métrica más importante es el Recall, el Accuracy en regresión logística muestra una precisión similar al resto de modelos, cercana al 0.80.
Esto indica que, a pesar de ser un modelo lineal, la regresión logística está desempeñándose bien en este problema.





F1-Score

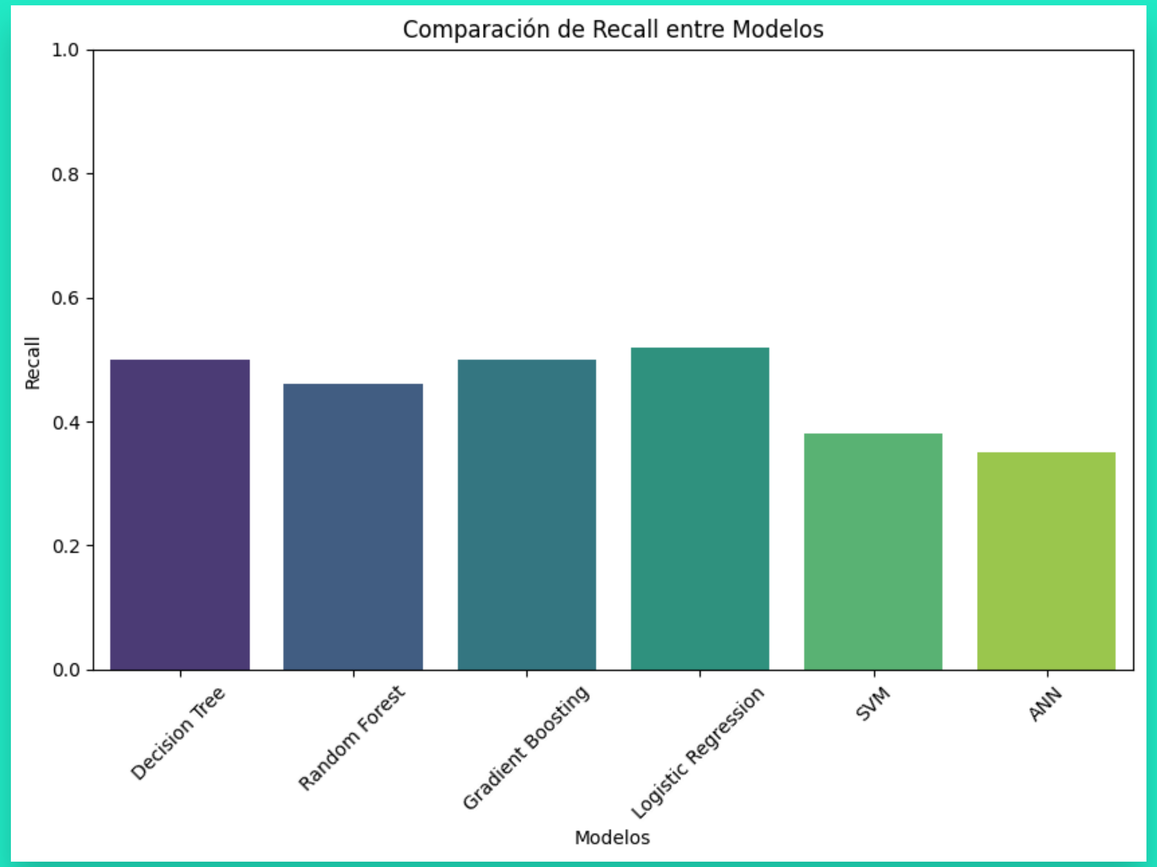
La regresión logística tiene el F1-Score más alto entre los modelos evaluados, cerca de 0.60. Esto sugiere que. Este modelo también maneja bien el balance entre precisión y recall para este problema.

Recall antes de balanceo, normalización y ajuste de hiperparámetros

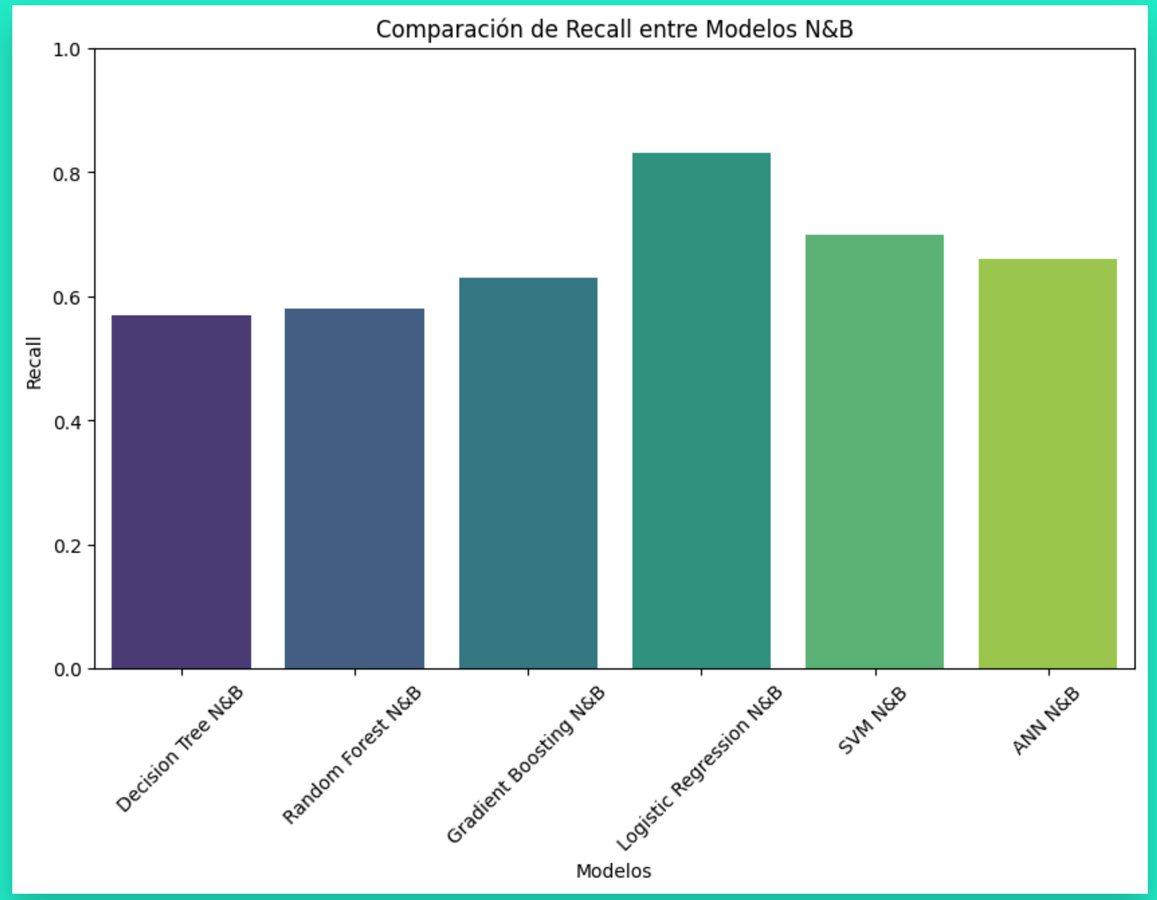
En este gráfico, se compara el Recall en los diferentes modelos antes de aplicar balanceo, normalización y ajuste de hiperparámetros.

Se puede observar que el modelo de **Regresión Logística** muestra un rendimiento ligeramente superior comparado con otros modelos como Decision Tree, Random Forest, Gradient Boosting, SVM y ANN.

Aunque la diferencia no es extremadamente marcada, es claro que la regresión logística ya comienza a destacarse en su capacidad para identificar correctamente los casos positivos.



Recall después de balanceo, normalización y ajuste de hiperparámetros



En este segundo gráfico, después de aplicar técnicas de balanceo, normalización y ajuste de hiperparámetros, podemos ver que el modelo de **Regresión Logística** no solo sigue siendo el que mejor rendimiento tiene, sino que aumenta significativamente su ventaja.

La diferencia de rendimiento entre el modelo de Regresión Logística y los otros modelos se hace más notable, destacando la efectividad de las técnicas aplicadas para mejorar su capacidad de detección de clientes en riesgo de churn.

Matriz de confusión: evaluación del modelo final

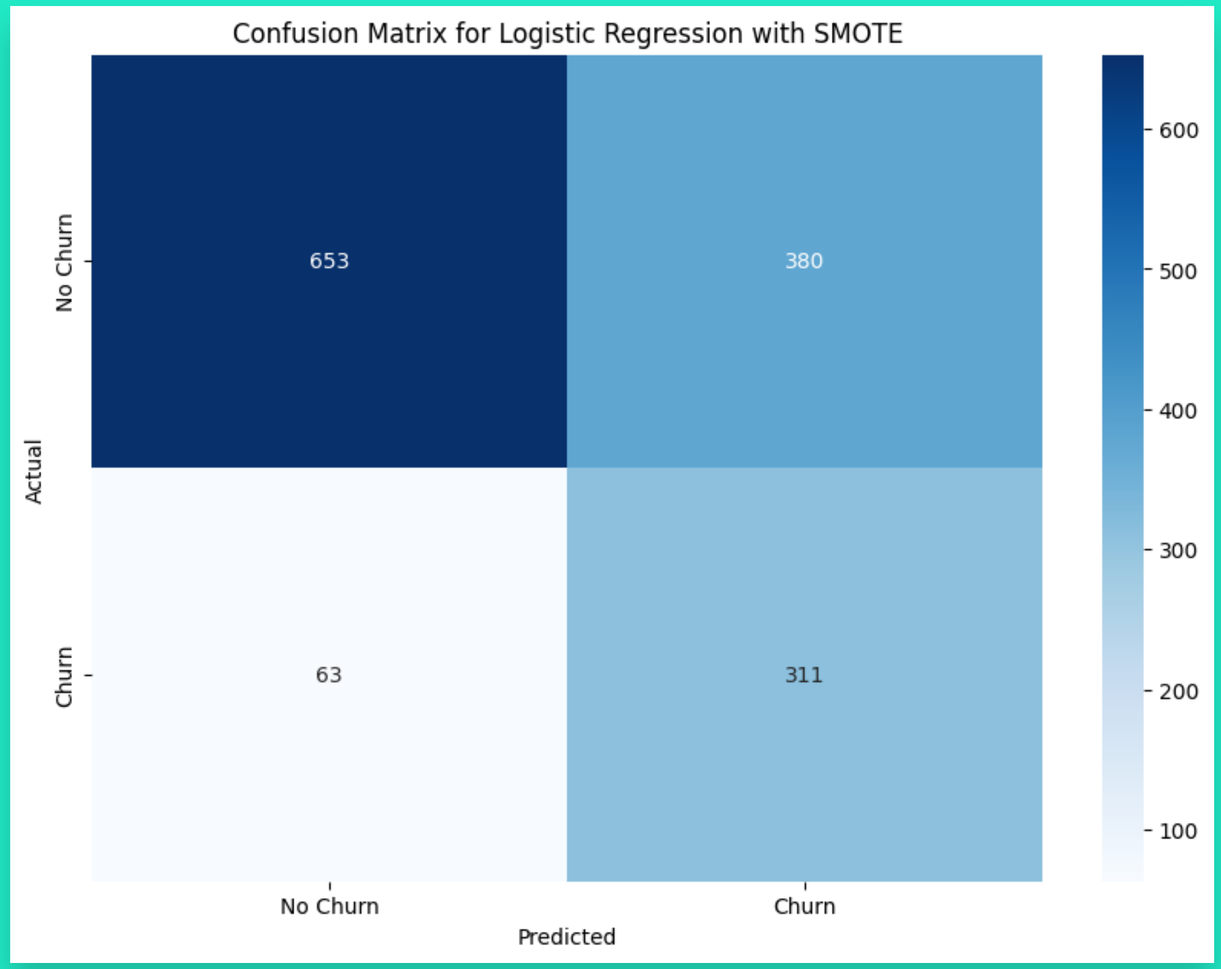
Con este modelo el recall es de 0.83, lo que significa que se identifica correctamente la mayoría de los clientes que harán churn.

True Negatives (TN = 653): Los clientes que el modelo identificó correctamente como no probables a hacer churn.

False Positives (FP = 380): Los clientes que el modelo predijo que harían churn, pero que en realidad no lo hicieron.

False Negatives (FN = 63): Estos son los clientes que el modelo predijo que no harían churn, pero que en realidad sí lo hicieron.

True Positives (TP = 311): Los clientes que el modelo identificó correctamente como probables a hacer churn.



Conclusión, siguientes pasos

El recall de 0.83 refleja un alto nivel de éxito en la identificación de clientes en riesgo de churn, lo cual es crucial para implementar estrategias de retención efectivas.

Sin embargo, es fundamental continuar optimizando el modelo, monitoreando su rendimiento en producción una vez desplegado y afinar las estrategias de retención de clientes con acciones personalizadas basadas en las predicciones.

¡Gracias por la atención!

GitHub: https://github.com/nacjacds/Telecom_Churn_Predictor

Web: https://nachojacquot.com

LinkedIn: https://www.linkedin.com/in/jacquot/

Probar el modelo:
https://telechurnpredictor.streamlit.app/