

Predicción supervisada de *churn* bancario: metodología, resultados y discusión

Jose Morales Rendón

Facultad de Ciencias Físico-Matemáticas, UANL

Keywords: Churn, Clasificación binaria, Regresión Logística, Banca, Aprendizaje supervisado

Contexto

Se aborda la predicción de abandono de clientes (*churn*) en banca como un problema de **clasificación binaria** con clases desbalanceadas. Se emplea *Regresión Logística* por su interpretabilidad y capacidad para entregar probabilidades calibradas, integrando un flujo de preprocesamiento (escalado y codificación *one-hot*) en un *pipeline* reproducible. En el conjunto **BankChurners** (10,127 observaciones; 18 variables; 16.07 % de churn) el modelo alcanza **ROC-AUC = 0.90**, con **recall** de 0.82 para la clase positiva, lo que resulta adecuado para escenarios donde el costo de no detectar a un cliente que se irá es mayor que el de intervenir a uno que permanecería.

1. Metodología

Problema y datos.. El objetivo es estimar la probabilidad de abandono de cada cliente (*Churn*=1 si abandona; 0 en caso contrario) a partir de variables demográficas y transaccionales. El dataset **BankChurners.csv** contiene 10,127 registros y 18 variables; la proporción de la clase positiva es 16.07 % (desbalance moderado).

Modelo.. Se emplea **Regresión Logística** como línea base interpretable. Dado un vector de predictores $x \in \mathbb{R}^p$, el modelo estima:

$$P(y = 1 | x) = \sigma(\beta_0 + \beta^\top x) = \frac{1}{1 + e^{-(\beta_0 + \beta^\top x)}}, \quad (1)$$

ajustando β por máxima verosimilitud. Para mitigar el desbalance se utiliza `class_weight=balanced`.

Preprocesamiento y validación.. Las variables numéricas se escalan con *StandardScaler* y las categóricas se codifican con *One-Hot Encoder*, todo dentro de un *ColumnTransformer*. Se realiza partición entrenamiento/prueba 75/25 con estratificación para preservar la proporción de clases. El *pipeline* (*preprocesador + modelo*) garantiza reproducibilidad y facilita la futura incorporación de otros algoritmos.

2. Métricas de evaluación

Al ser un problema de clasificación binaria, se emplean métricas derivadas de la matriz de confusión y de curvas basadas en probabilidades:

- **Accuracy:** proporción global de aciertos (puede ser engañosa con desbalance).
- **Precision y Recall** de la clase positiva (churn): calidad de alertas y cobertura de verdaderos abandonos.
- **F1-score:** media armónica entre *Precision* y *Recall*.
- **ROC-AUC:** capacidad de discriminación independiente del umbral.
- **PR-AUC:** área bajo la curva *Precision-Recall*, relevante en casos de desbalance severo.

3. Resultados

En el conjunto de prueba, el modelo logra:

- **Accuracy:** 0.84
- **Recall (clase churn):** 0.82
- **Precision (clase churn):** 0.49
- **F1-score (clase churn):** 0.62
- **ROC-AUC:** 0.90

La matriz de confusión mostró una alta detección de la clase positiva, con falsos positivos moderados, consistente con una política que prioriza cobertura sobre pureza en alertas.

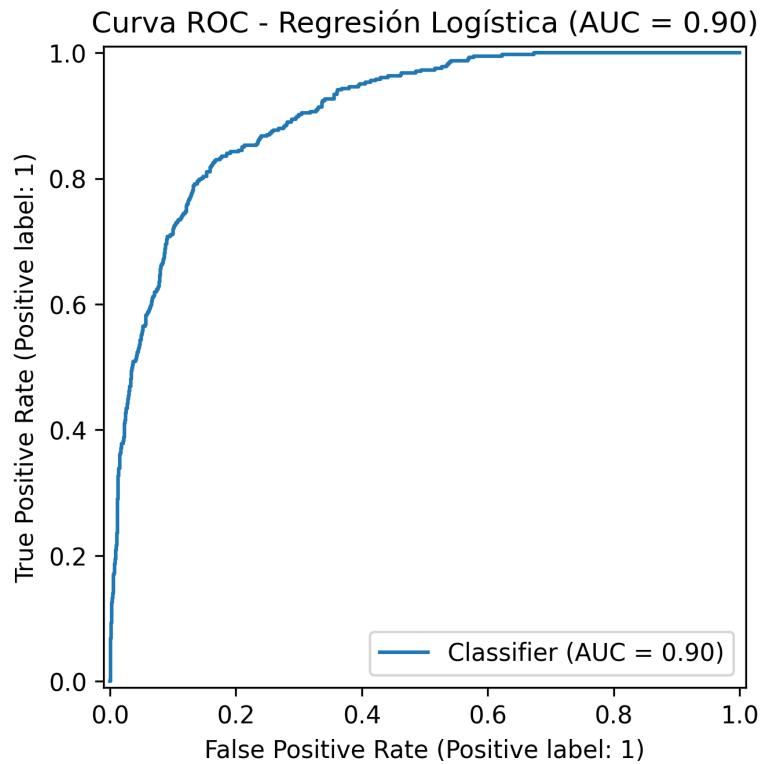


Figura 1: Curva ROC del modelo de Regresión Logística. El área bajo la curva (AUC = 0.90) evidencia una alta capacidad de discriminación.

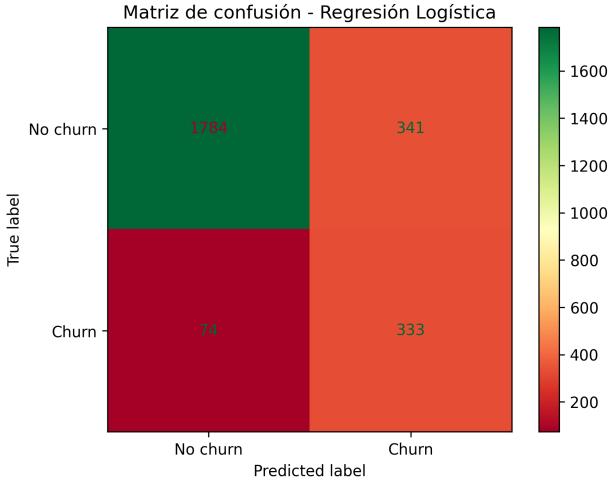


Figura 2: Matriz de confusión del modelo. Se observa un alto número de verdaderos negativos y un *recall* del 82 % para la clase *churn*.

4. Discusión

El rendimiento (AUC cercano a 0.90) confirma que una línea base interpretable puede ser competitiva para priorización de campañas de retención. El **recall elevado** es deseable cuando el costo de un falso negativo (no intervenir a un cliente que se va) supera el de un falso positivo (intervenir a un cliente que permanecería).

Para mejorar la *precision* sin sacrificar demasiado *recall*, se sugiere: (i) ajuste de umbral según una matriz de costos, (ii) comparación con *Random Forest* o *XGBoost*, y (iii) análisis de explicabilidad (p.ej., valores SHAP) para identificar impulsores del *churn* y diseñar intervenciones específicas.

5. Conclusión

El modelo de Regresión Logística demostró ser una herramienta sólida, equilibrando precisión y sensibilidad en un contexto de datos desbalanceados. Su simplicidad y transparencia lo convierten en un excelente punto de partida para estrategias analíticas en banca.

Los resultados ($AUC = 0.90$, $Recall = 0.82$) validan que una implementación bien estructurada puede alcanzar desempeño de nivel profesional sin recurrir a modelos complejos. Además, la metodología propuesta —preprocesamiento, validación estratificada y evaluación mediante múltiples métricas—

sienta una base robusta para la extensión hacia modelos más avanzados y explicables.

En síntesis: este proyecto demuestra que los datos no solo describen el pasado; bien analizados, anticipan el futuro. Cada cliente detectado a tiempo representa una oportunidad ganada y un paso más hacia una banca más inteligente y centrada en las personas.

Reproducibilidad

El flujo se implementó en Python ([scikit-learn](#)) con un *pipeline* que integra preprocesamiento y modelado. La partición fue 75/25 con estratificación, y se calcularon métricas de desempeño junto con las curvas ROC y PR. Los gráficos fueron generados automáticamente y exportados como `roc_curve.png` y `conf_matrix.png` para su documentación visual.