telecom

Predicción de Default Crediticio

Yanina Iberra

CASO DE USO

DATOS

En el archivo se encuentran distintas variables, cuyo diccionario de datos se detalla en el archivo del dataset en formato Excel.

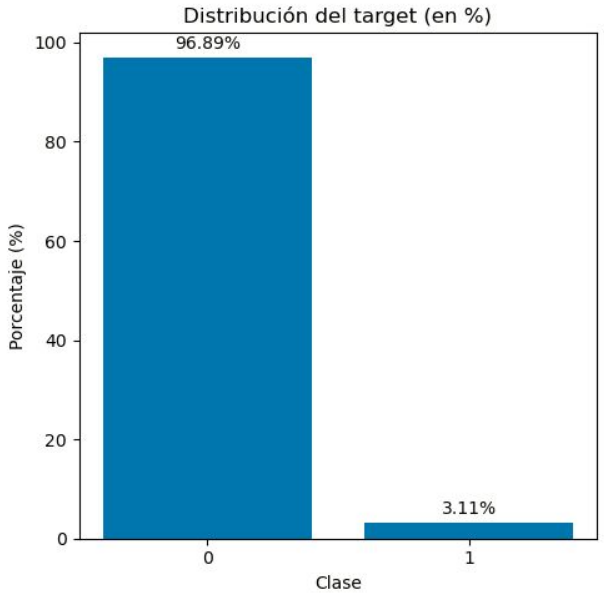
- El archivo contiene un registro por cliente por mes.
- El campo **ID** es un identificador único por cada persona (como el documento, cuil, o el número de línea).

OBJETIVO

El dataset cuenta con un campo **Target**, el cuál es el objetivo del modelo (lo que hay que predecir), en este caso si el cliente va a ser default crediticio o no.

DATASET

Conjunto de datos de clientes, con un registro por cliente por mes. Y el campo Target a predecir, en este caso si el cliente va a ser default crediticio o no.



El dataset se encuentra desbalanceado, donde la clase minoritaria es la de si tener Default Crediticio (target=1). Por lo cual trabajaremos una técnica de balanceo de clases, para aumentar la clase minoritaria.

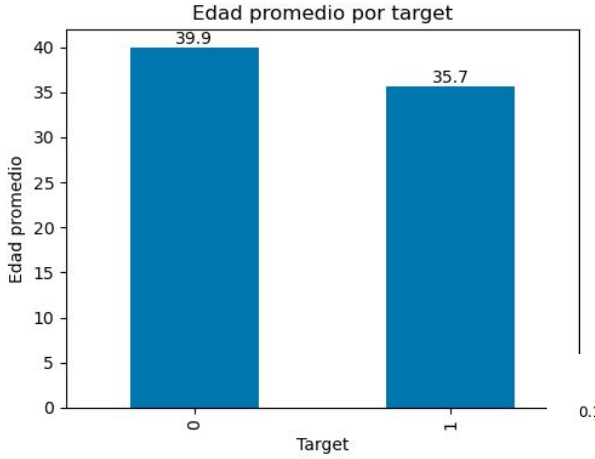
VALORES NULOS

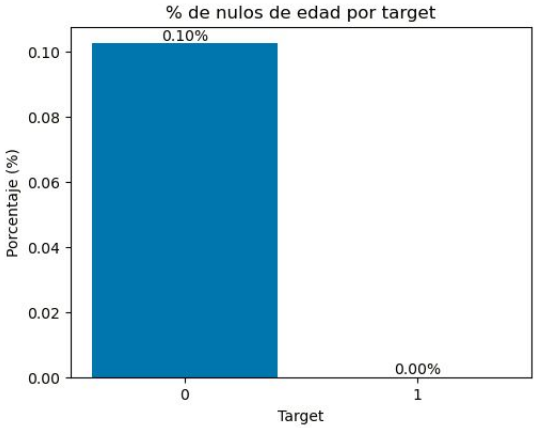
Insights

En las variables numéricas, se observa que existe un 3,42% de nulos en `**sco_fin**` que representa el Score financiero.

También existen nulos en la variable `**edad**`, un 0.1%.

columna	# faltantes	% faltantes
sco_fin	275	3.42
edad	8	0.10





- Se observa que quienes NO hacen default tienen mayor **edad** promedio (~39.9) que quienes sí (target=1, ~35.7) (~10–12% menor en el grupo con default).
- Pareciera que la edad se asocia inversamente con el default (más jóvenes, más riesgo), al menos en promedio.
- Imputaremos por la mediana, tanto en train como en test.

VALORES NULOS

target	# registros	# missing sco_fin	% missing sco_fin
0	7797	245	3.14
1	250	30	12.00

Score Financiero

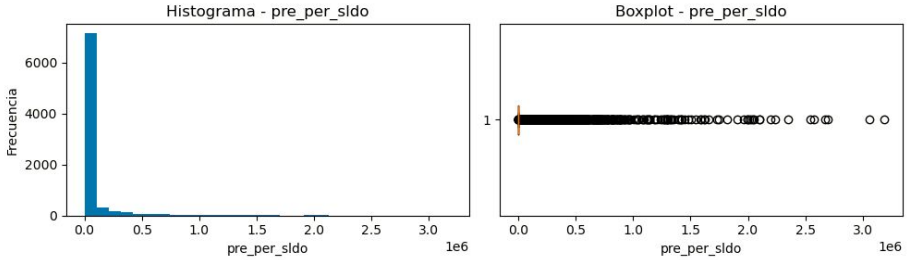
Como la tasa de faltantes cambia por clase (3.14% en target=0 vs 12% en target=1), la "falta" es informativa, por lo cual imputaremos la variable score con la mediana (para evitar "Data leakage"), tanto en train como en test.

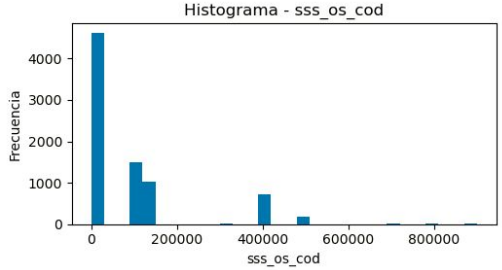
target	promedio sco_fin
0	470.08
1	343.66

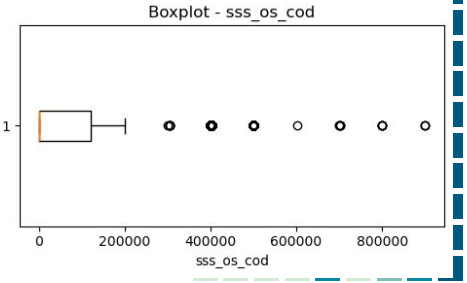
- También observamos que el promedio de "sco_fin" difiere entre targets (470 en target=0 vs 343 en target=1) -> es una variable importante como input del modelo.
- Se decide agregar también una columna binaria "**sco_fin_missing**" (=1 si falta el score y =0 si tiene score).

Análisis univariado: Distribuciones y outliers

ANÁLISIS UNIVARIADO DISTRIBUCIONES Y OUTLIERS







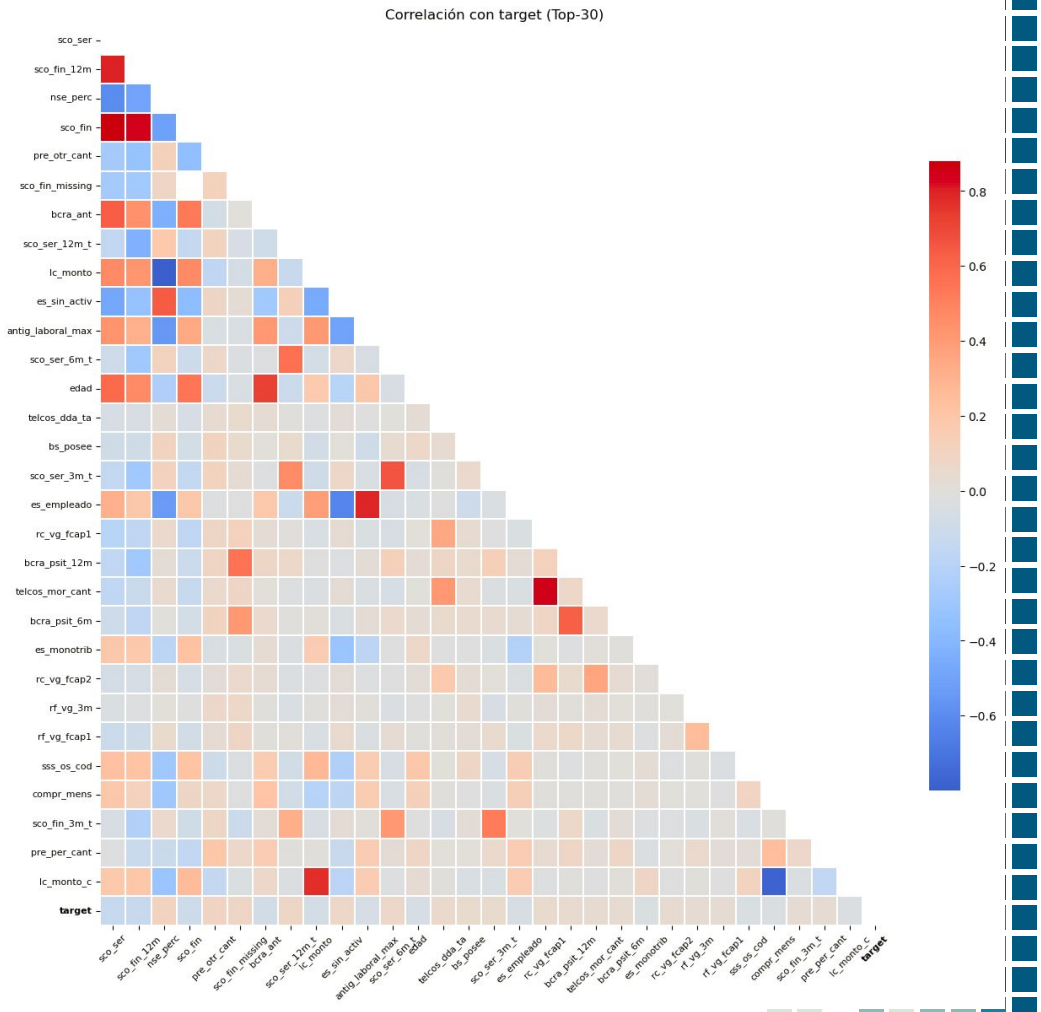
- Se observa diversidad de outliers, con diferentes proporciones y distribuciones, pero existen columnas con un porcentaje de outliers superior al 20%.
- Se decide tratar dichos outliers limitando sus valores a los percentiles p5–p95. Este paso se aplica tanto a los datos de entrenamiento como de test.
- Variables numéricas con gran amplitud serán escaladas.

columna	pct_outliers	n_outliers	n
sco_fin_6m_t	27.64	2224	8047
tc_sdot	24.54	1975	8047
tc_cant	24.54	1975	8047
sco_fin_3m_t	20.63	1660	8047
pre_per_sldo	19.96	1606	8047
pre_per_cant	19.96	1606	8047
sco_ser_12m_t	15.96	1284	8047
pre_otr_sldo	15.11	1216	8047
sss_os_cod	11.32	911	8047
sco_ser_6m_t	10.17	818	8047

Análisis multivariado con respecto al target

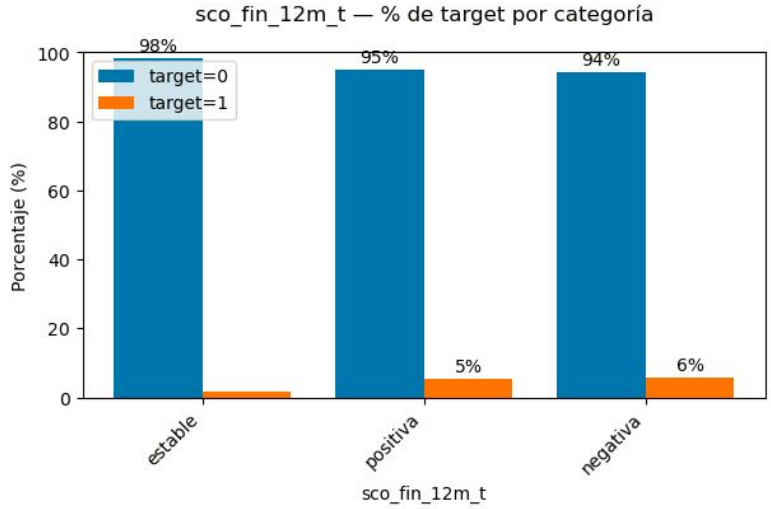
ANÁLISIS MULTIVARIADO CON RESPECTO AL TARGET

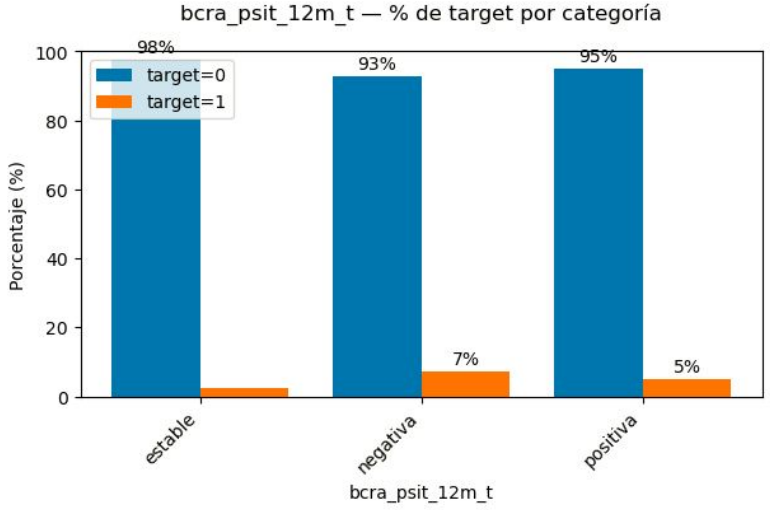
- No se tendrán en cuenta para el modelo las variables con valores constantes (**varianza cero**), ya que no aportan información al modelo.
- Eliminación de la **multicolinealidad**: si dos variables están altamente correlacionadas (p. ej., corr > 0,9), solo dejar una de ellas como input del modelo.
- Se observa una baja **correlación** absoluta con el target (< 0.2 tanto Pearson como Spearman) => no hay relación lineal fuerte univariada, pero pueden aportar otro tipo de relación/información.
- Esas variables, por sí solas, no muestran un patrón simple (recta o tendencia única) con el target. Pueden ser útiles en conjunto o realizar transformaciones.



ANÁLISIS MULTIVARIADO CATEGÓRICAS

- Para la variable `**sco_fin_12m_t**` vemos tasas de default crediticio distintas por categoría (~2% estable vs ~5–6% positiva/negativa), por lo cual vamos a dejar ésta variable ya que aporta señal.
- Para la variable `**bcra_psit_12m_t**` podemos ver diferencias claras de tasa de default crediticio por categoría (p.ej., ~2% “estable”, ~5% “positiva”, ~7% “negativa”). Esa separación indica poder diferencial útil para el modelo.
- Se convierten dichas variables categóricas en numéricas con el método One-hot encoder.





FEATURE ENGINEERING

Nuevas variables

NUEVAS VARIABLES & TRANSFORMACIONES

Se generan nuevas variables que puedan sumar información para la determinación de Default Crediticio:

- **Peor score actual (score_min)**: si cualquiera de los dos scores es bajo, aumenta la probabilidad de default.

  Cálculo: score_min = min(sco_fin, sco_ser) (por fila).

- **Tendencia 12m del financiero (sco_fin_delta_12m)**: deterioro reciente (delta negativo) es una alerta temprana de morosidad.

  Cálculo: sco_fin_delta_6m = sco_fin - sco_fin_12m.

  En riesgo crediticio, los datos más cercanos en el tiempo suelen predecir mejor que los muy viejos.

- **Imputamos los nulos** en las variables `sco_fin` y `edad` con sus correspondientes medianas.
- Para las variables numéricas cuya cantidad de **outliers** supere el umbral del 20%, se acotan sus valores dentro de los percentiles p5–p95

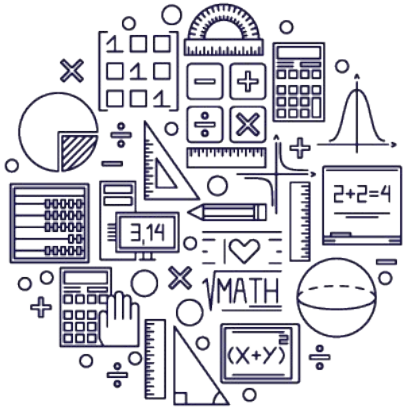
MODELOS

MODELOS - MÉTRICA

Métrica del modelo:

- Utilizamos como métrica: **PR-AUC (Average Precision)** que nos dice qué tan bien encontramos morosos sin llenarnos de "falsas alarmas".
  - Recall (sensibilidad) = cuántos morosos reales (positivos) identificamos.
  - Precision = de los que marcamos como morosos, qué porcentaje realmente lo son (evita falsos positivos = rechazar buenos).
- PR-AUC resume la precisión promedio que el modelo mantiene mientras aumentamos el recall. Con el siguiente objetivo: ¿podemos capturar más morosos sin disparar el número de buenos rechazados?

- En default suele haber pocos positivos. La métrica **ROC-AUC** se centra justo donde duele el negocio: morosos detectados (recall) vs buenos mal rechazados (falsos positivos implica baja precisión). Así, mantenemos buena precisión a medida que capturamos más morosos, reduciendo incobrables sin rechazar de más.



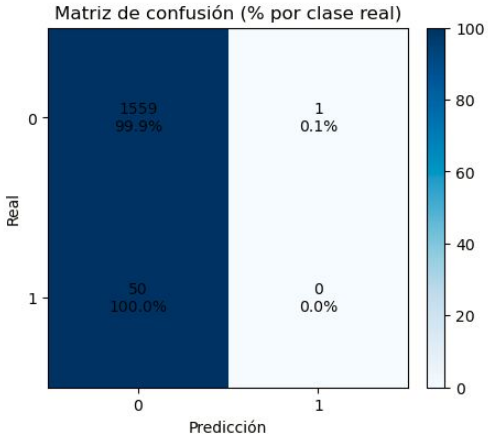
MODELO BASELINE - RANDOM FOREST

```
PR-AUC (test): 0.0605  | baseline (prev): 0.0311  | lift: 1.95x
ROC-AUC (test): 0.6697

Classification report (umbral=0.5):
              precision    recall  f1-score   support

           0      0.969     0.999     0.984      1560
           1      0.000     0.000     0.000        50

    accuracy                          0.968      1610
   macro avg      0.484     0.500     0.492      1610
weighted avg      0.939     0.968     0.953      1610
```



Insights:

- **PR-AUC: 6.25%** implica que con el modelo duplicamos (**lift**=1,95) la precisión al buscar morosos con respecto al aza r(baseline: 3.1%)
- El modelo aporta (≈2x sobre el azar en precisión-recall) pero, con una **ROC-AUC ~0.67** , su capacidad discriminante es moderada.
- El modelo es hiperconservador: no marca ningún default crediticio (clase 1 con precision/recall/F1 = 0).
- Accuracy 0.968 es engañosa (la clase 1 es ~3%). Con solo predecir "todo 0" ya se logra algo parecido.
- Clase 0 sale muy bien (recall 0.999), pero a costa de perder todos los morosos (recall 0.0 en clase 1). **Para negocio, inaceptable.**
- **Este modelo sirve para priorizar y filtrar (p.ej., a quién revisar primero), no para automatizar decisiones duras sin controles adicionales.**

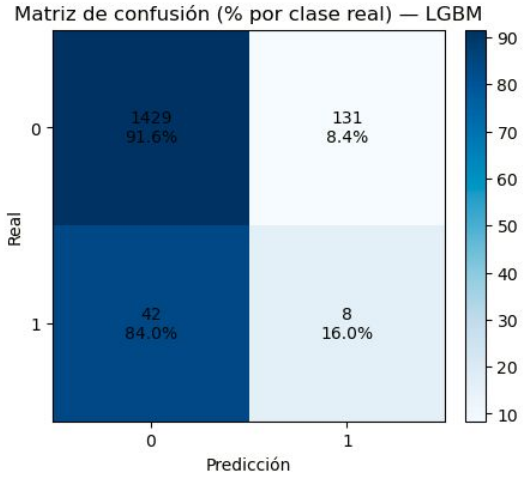
MODELO - LGBM

```
[LGBM] PR-AUC(test)=0.0551 | baseline(prev)=0.0311 | lift=1.77x
[LGBM] ROC-AUC(test)=0.6222
Prevalencia train=0.0311 | val=0.0311 | test=0.0311
Umbral elegido por F1.5 (validación): 0.019

Classification report (test):
              precision    recall  f1-score   support

           0      0.971     0.916     0.943      1560
           1      0.058     0.160     0.085        50

    accuracy                          0.893      1610
   macro avg      0.515     0.538     0.514      1610
weighted avg      0.943     0.893     0.916      1610
```



Insights:

- **PR-AUC = 0.0551** vs baseline (prevalencia) = 0.0311 ⟶ lift ≈ 1.77×*, la ganancia aún es modesta.
- **ROC-AUC = 0.622**, separación moderada (hay señal, pero lejos de un modelo fuerte).
- **Recall** de morosos = 16% ⇒ de 50 morosos reales, el modelo detecta ~8.
- **Precisión** = 5.8% ⇒ de cada 100 casos marcados, ~6 son morosos y ~94 son falsos positivos.
- **El modelo sirve para priorizar (mejor que azar), no para automatizar rechazos: captura poco y genera muchos falsos positivos con este corte.**
- **Puede usarse como filtro inicial / ranking combinado con reglas de negocio o revisión manual.**

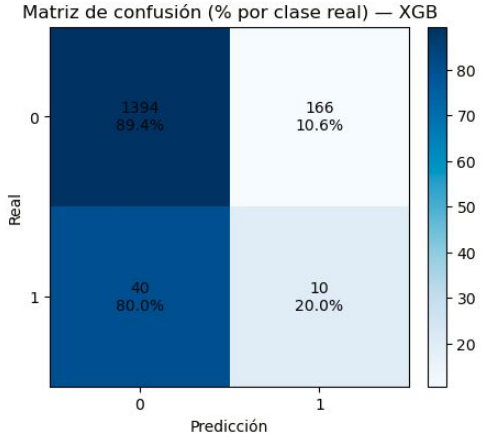
MODELO - XGBOOST

```
[XGB] PR-AUC(test)=0.0505 | baseline(prev)=0.0311 | lift=1.62x
[XGB] ROC-AUC(test)=0.6313
Prevalencia train=0.0311 | val=0.0311 | test=0.0311
Umbral elegido por F1.0 (validación): 0.079

Classification report (test):
              precision    recall  f1-score   support

           0      0.972     0.894     0.931      1560
           1      0.057     0.200     0.088        50

    accuracy                          0.872      1610
   macro avg      0.514     0.547     0.510      1610
weighted avg      0.944     0.872     0.905      1610
```



Insights:

- **PR-AUC = 0.0505** vs baseline = 0.0311 ⟶ lift ≈ 1.62×: hay señal, pero modesta.
- **ROC-AUC = 0.631**: separación moderada (mejor que azar, lejos de un modelo fuerte).
- **Recall** de morosos = 20% ⟶ de 50 morosos reales, detecta ~10.
- **Precisión** = 5.7% ⟶ de cada 100 marcados, ~6 son morosos y ~94 son falsos positivos.
- **Accuracy** 0.872 no es buen criterio aquí (la clase 1 es ~3.1%): puede ser alta aunque el modelo falle en morosos.
- **El modelo sirve para priorizar (mejor que azar), no para decisiones automáticas: captura pocos morosos y genera muchos falsos positivos con este umbral.**

POSIBLES MEJORAS

- Continuar con el análisis de features para sumar variables, revisar variables con poco aporte y redundancias.
- Prueba de otros modelos como CatBoost o incluso aplicar ajuste por hiperparámetros, para intentar obtener mejoras.
- Análisis de feature importances para tratar de reducir la cantidad de variables que están tomando los modelos.
- Comparar las métricas de train y de test para verificar que no haya overfitting (altas en train, pero más bajas en test).
- Probar si los modelos mejoran usando técnicas de reducción de dimensionalidad, como PCA. Para reducir la cantidad de variables que toma el modelo.
- Análisis de interpretabilidad de modelos con librerías como SHAP, ésto nos permite analizar las predicciones individuales. Por ejemplo responde a la pregunta: "¿Para este cliente en particular, qué variables hicieron que lo clasifiquemos como default si /default no?" o para explicar por ej: por qué un cliente fue mal clasificado.

THANKS!

telecom