The background of the slide features a wide-angle shot of a modern office interior. Numerous employees are seated at their desks, each equipped with multiple computer monitors displaying complex data visualizations. The office has floor-to-ceiling windows that offer a panoramic view of a dense urban skyline at dusk or night. The lighting is dramatic, with the warm glow of the city lights reflecting off the glass and illuminating the workers' faces and screens.

Proyecto Machine Learning

Credit Score Classification

Situación de Negocio

CONTEXTO

El banco Pandaland, mediante la información financiera, **clasifica a sus clientes** con diferentes puntuajes para determinar si son aptos para un préstamo o no

PROBLEMA

Se está realizando a través de **métodos manuales** en los se destinan muchos recursos

SOLUCIÓN

Automatizar este proceso para **aumentar la eficiencia y reducir los costes** de la entidad bancaria

ENFOQUE TÉCNICO

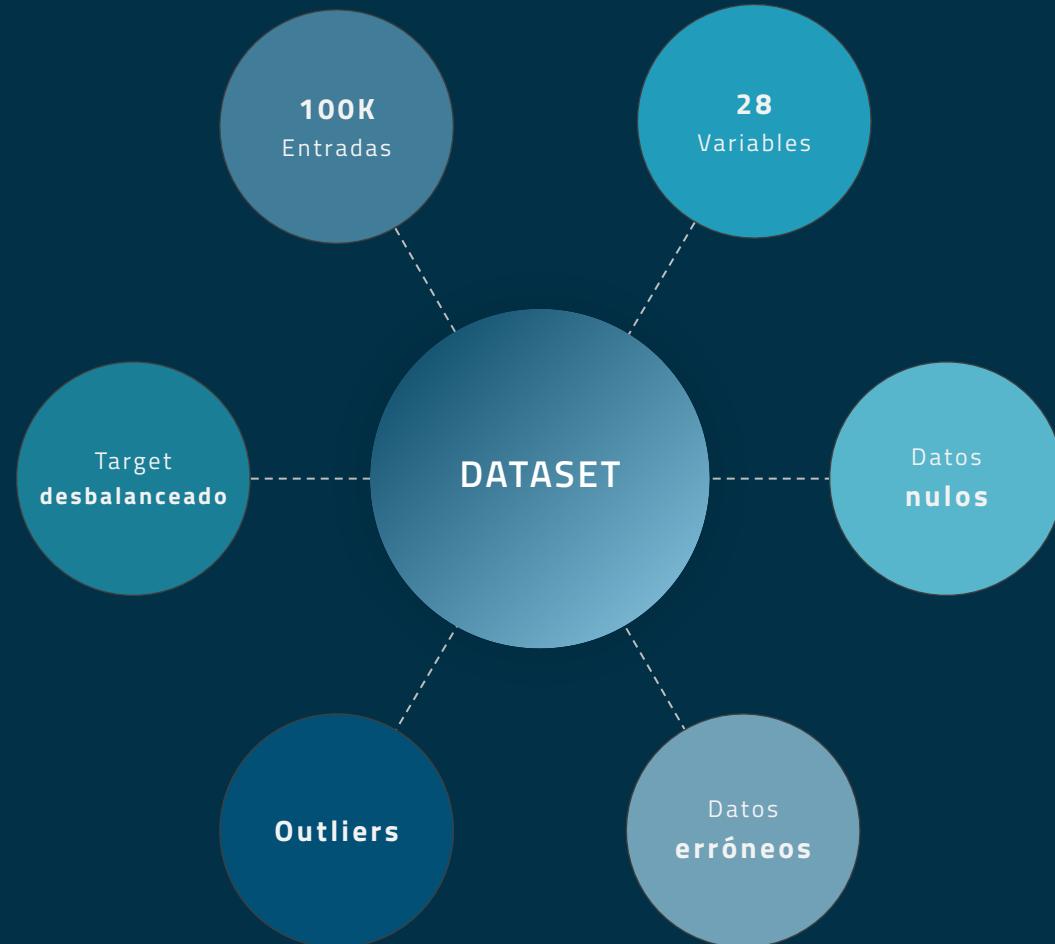
Credit Scoring a través de un **modelo supervisado de clasificación multiclasé**



— Un primer vistazo al dataset

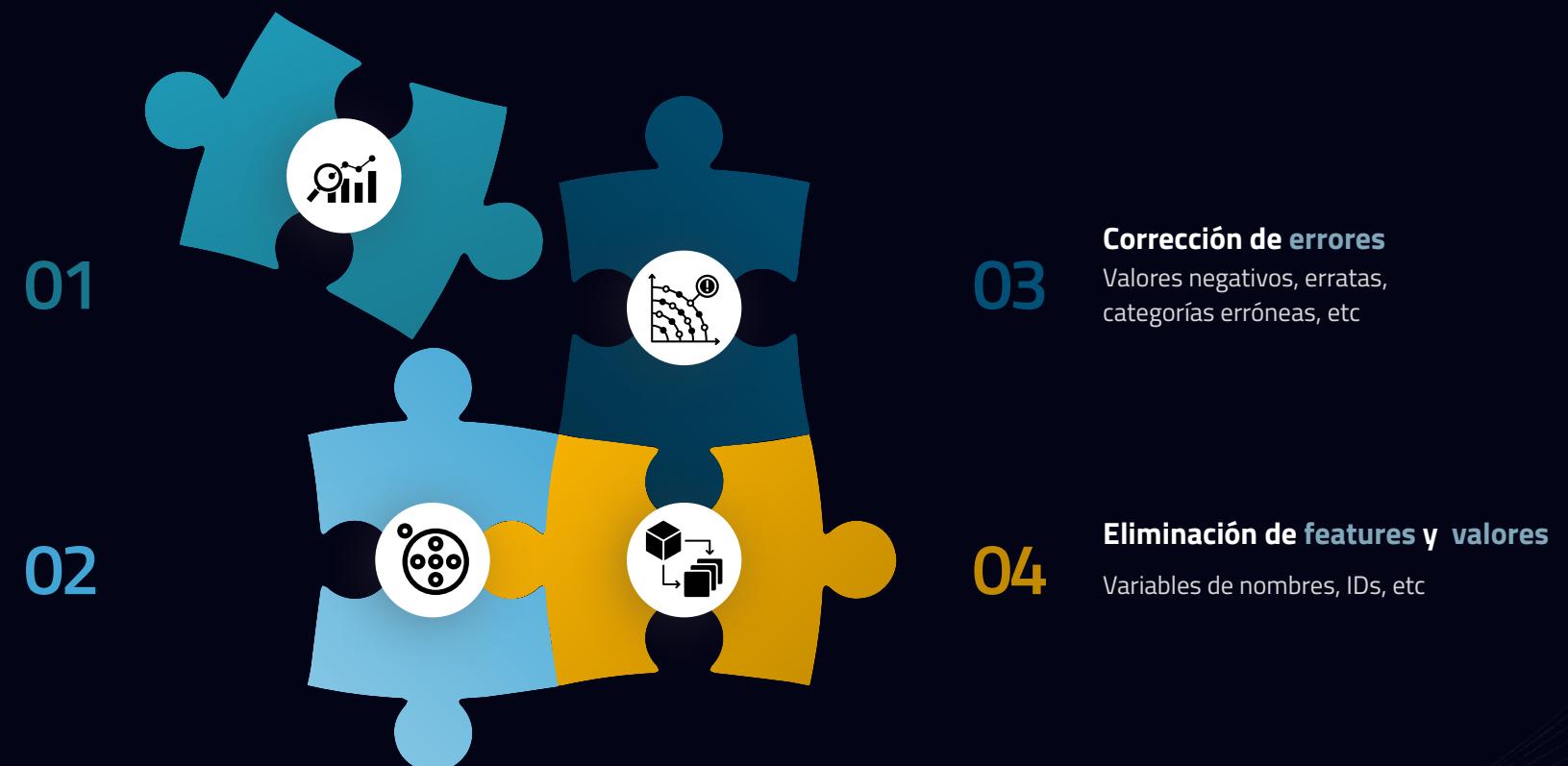
A cerca de los datos

Como primer paso, se ha realizado un análisis previo para entender la **estructura de los datos** y acciones necesarias a realizar para el procesado



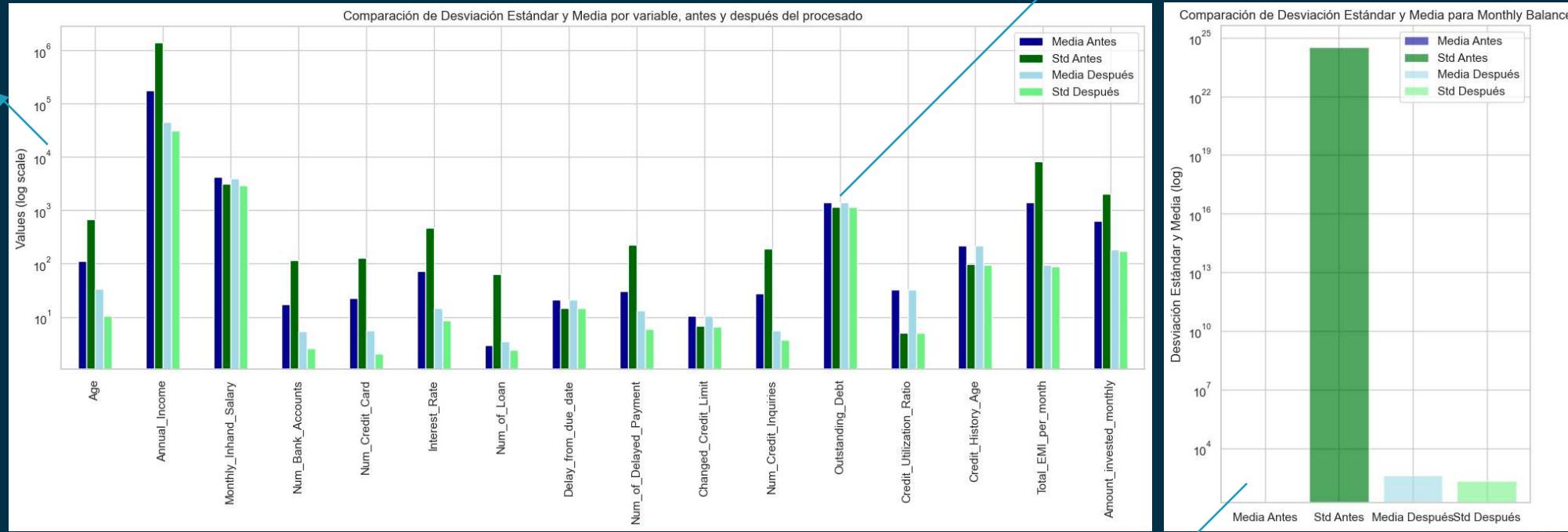
Pasos principales de la limpieza de datos

Resumen **procesado** de información



Actualización estructura de datos

- **Gráficas:** muestran una comparativa del cambio de la estructura de datos antes y después del procesado
- **Métricas:** media y la **desviación estándar**, para entender la estructura de datos y a detectar altos valores de desviación estándar perjudiciales para el modelo
- **'Monthly Balance' corregido:** valor trillonario negativo eliminado, estabilizando las métricas



VARIABLES ESTABLES:
En algunos casos
transformación no necesaria

Media no visible:
En 'Monthly Balance'
debido a valor extremo neg.

Transformaciones esenciales previas al modelado

Acciones principales feature engineering

01

Encoding categóricas

- `Get_dummies` —> variables con pocas categorías
- `Mapping` —> variables ordenadas con pocas categorías
- `Replace` —> binarias
- `Sort_values and select` —> variable con muchas categorías



02

Split de datos

- Train —> 80% (aprox. 80.000 entradas)
- Test —> 20% (aprox. 20.000 entradas)



03

Feature importance

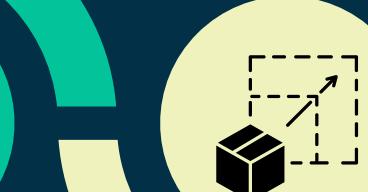
- `RandomForestClassifier` —> mejor herramienta



04

Reducción de features

- Reducir dimensionalidad —> 20 features



05

Escalado de datos

- `StandardScaler` —> distribución de datos normal

Opciones más viables para problema de clasificación multiclas

Selección de Modelos

01

Logistic Regression

- Útil para encontrar relaciones lineales
- Simple y rápido de entrenar

02

Naive Bayes

- Clasificador que aplica el Teorema de Bayes
- Rapidez y eficacia en conjuntos de datos grandes

03

XGBoost

- Implementación optimizada de Gradient Boosting
- Más rápido que GB
- Eficiente para tareas de clasificación

04

Random Forest

- Construcción múltiples árboles de decisión
- Robustez para capturar relaciones complejas
- Múltiples opciones de personalización

Primeras pruebas realizadas entre diferentes modelos

Testing de Modelos

En primer lugar, se han realizado una serie de pruebas utilizando diferentes ajustes y herramientas para la búsqueda de los mejores hiperparámetros

Modelo	Accuracy	Underfitting	Overfitting
Logistic Regression	- Train: 63% - Test: 63%	Alto	No
Naive Bayes	- Train: 60% - Test: 60%	Alto	No
XGBoost	- Train: 80% - Test: 74%	Bajo	Moderado (6%)
Random Forest	- Train: 98% - Test: 79%	No	Alto (19%)

- **Logistic Regression y Naive Bayes:** underfitting alto, precisión test < 80%
- **XGBoost:** mejora respecto a estos, pero no cumple aún con el mínimo deseado
- Seleccionamos **Random Forest** para optimizar y abordar el overfitting detectado

Focus en el modelo elegido para intentar mejorar los resultados

Extensión Random Forest(I)

Tras realizar múltiples pruebas con diferentes tipos de ajustes, se lleva a cabo una selección de los **5 mejores modelos**. A continuación se muestra el favorito

Modelo	Accuracy	F1- Score	Underfitting	Overfitting
Random Forest	<ul style="list-style-type: none"> - Train: 95% - Test: 79% 	<ul style="list-style-type: none"> - Train: 95% - Test: 78% 	No	Moderado (16%)



Optimización en pruebas

- RandomizedSearchCV & HalvingGridSearch:
- Reducción de tiempos = más pruebas
- Enfoques alternativos
- Reducción de coste computacional



Balanceo de clases

- Pipeline SMOTE: creación muestras sintéticas en train
- StratifiedKFold : validación cruzada manteniendo prop. a clases
- Class Weight Balanced: ajusta pesos inversamente prop. a frecuencias



Ajustes de profundidad

- 'max_depth': v.limitados para evitar árboles demasiado complejos
- 'max_leaf_nodes': evitar árboles excesivamente grandes
- 'n_estimators': reducir el número de árboles

Extensión Random Forest(II)

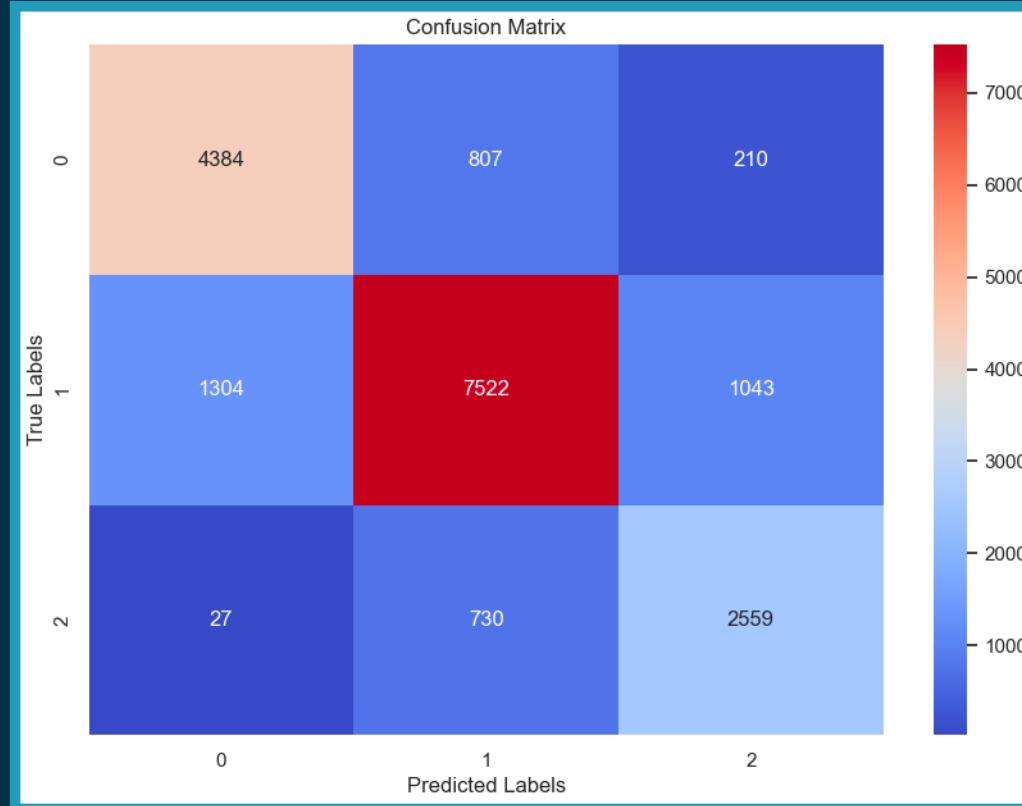
Desempeño del modelo por clase

- Clase 0 ('Poor'):
 - Alta precisión en identificar correctamente su clase
 - Confusión más común con Clase 1 que Clase 2
- Clase 1 ('Standard'):
 - Mejor tasa de aciertos del modelo
 - Se confunde con Clase 0 y Clase 2
- Clase 2 ('Good'):
 - Menor tasa de falsos negativos
 - La confusión con otras clases es infrecuente

Puntos clave

- Clase 1 muestra una identificación precisa, posible sesgo por ser mayoritaria
- La confusión predominante es entre Clase 0 y 1
- FN para la clase 2 son notoriamente bajos, lo que significa que casi nunca se confunde a las clases 0 o 1 como clase 2

Modelo	Accuracy	F1- Score	Underfitting	Overfitting
Random Forest	<ul style="list-style-type: none"> - Train: 95% - Test: 79% 	<ul style="list-style-type: none"> - Train: 95% - Test: 78% 	No	Moderado (16%)



Conclusiones finales

RECAP

- A partir del objetivo de negocio, se ha realizado la limpieza, análisis y preparación de datos para finalmente crear un modelo que pudiera clasificar de forma automática a los clientes a partir de su información crediticia
- Se ha conseguido obtener un modelo, que en términos generales, clasifica correctamente casi un 80% de las veces

ÁREAS DE MEJORA

Overfitting: aunque a través de los diferentes procesos se ha conseguido reducir, todavía queda margen para mejorar:

- Revisión completa de la fase de limpieza
- Análisis detallado de las variables
- Nuevas pruebas con el resto de modelos

¡Muchas gracias por vuestro tiempo!

