ABP – Módulo 6
Proyecto de Aplicación Ciencia de Datos

Hans Jorge Contreras Robledo

Fecha: 30/08/2025

1. Introducción

En este proyecto final se aborda la simulación de un dataset, la preparación de datos, el entrenamiento de modelos de clasificación y regresión, la evaluación de métricas de desempeño y la implementación de una API de despliegue utilizando Flask.

1. Definición del problema y dataset

Se simuló un dataset de evaluación crediticia con 600 filas y 9 variables predictoras, incluyendo edad, ingresos, razón de deuda, morosidades, historial crediticio, hipoteca, dependientes y años de empleo. Los objetivos del análisis fueron dos: clasificar solicitudes de crédito aprobadas/rechazadas y estimar un límite de crédito.

2. Metodología

El pipeline incluyó las siguientes etapas:

- Preprocesamiento: escalamiento de variables numéricas mediante StandardScaler y codificación OneHotEncoder para variables categóricas.
- Modelos: se probaron Logistic Regression y RandomForest para clasificación; LinearRegression y RandomForestRegressor para regresión.
- Validación: se empleó validación cruzada estratificada 5-Fold para clasificación, seleccionando el mejor modelo por ROC-AUC.
- Métricas: Accuracy, Precision, Recall, F1, ROC-AUC para clasificación; MAE, RMSE y R^2 para regresión.
- Despliegue: se implementó una API con Flask para consumo del modelo en producción.

3. Resultados (clasificación)

Modelo seleccionado: LogisticRegression.

Accuracy: 0.758 | Precision: 0.940 | Recall: 0.767 | F1: 0.845 | ROC-AUC: 0.835

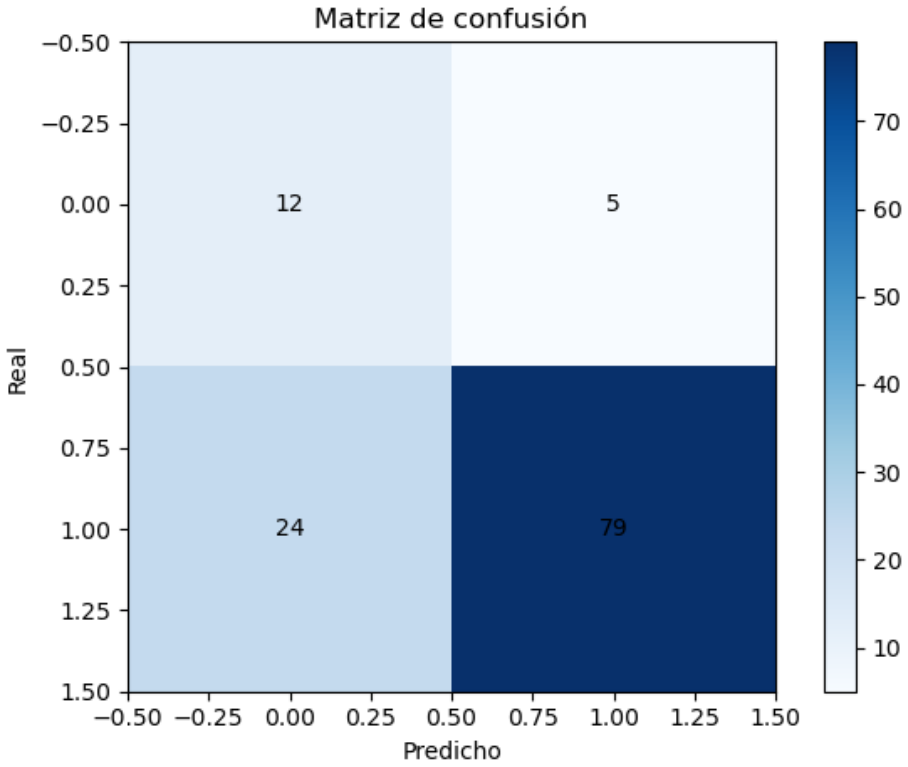

Figura 1. Matriz de confusión.

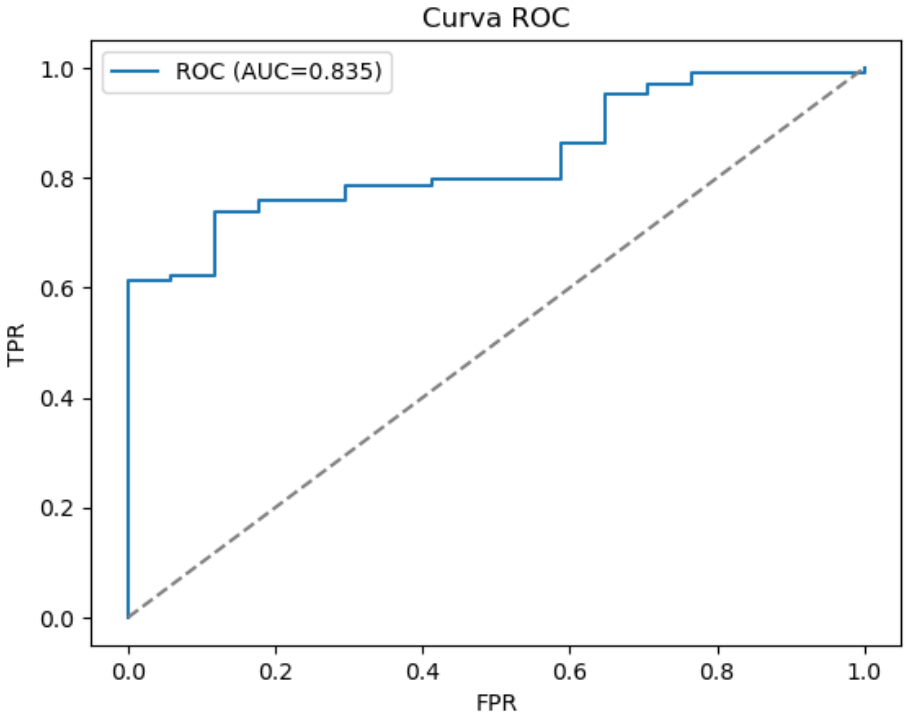

Figura 2. Curva ROC.

4. Resultados (regresión)

LinearRegression → MAE: 106,542 | RMSE: 134,130 | R^2: 0.913

RandomForestRegressor → MAE: 167,344 | RMSE: 209,402 | R^2: 0.787

Modelo seleccionado: LinearRegression.

5. API Flask y despliegue

Se serializó el pipeline del mejor modelo de clasificación y se creó una API RESTful con Flask. La API expone un endpoint /predict que recibe un JSON con las características de un solicitante y devuelve la probabilidad de aprobación y la clase predicha. Se incluyó un endpoint /health para verificación del estado del servicio.

6. Conclusiones

El proyecto integra todo el flujo de trabajo de un modelo de Machine Learning: simulación de datos, preprocesamiento, entrenamiento, evaluación y despliegue. Se logró construir un pipeline robusto, obteniendo un modelo con buen rendimiento y una API funcional para consumo en producción. Como mejoras futuras, se podría aumentar el tamaño del dataset, ajustar hiperparámetros y revisar periódicamente el modelo para evitar la degradación del desempeño.