Regresión con RandomForestRegressor

Yury Patricia Basto Figueroa

10 de septiembre de 2025

Introducción a la Regresión

- La regresión es una técnica estadística y de machine learning que modela la relación entre variables independientes X y una variable dependiente continua Y.
- Objetivo: encontrar una función $f(X)$ tal que

$$Y = f(X) + \varepsilon,$$

donde ε representa ruido aleatorio con media cero.

- Ejemplo: predecir el precio de una casa a partir de variables como tamaño, ubicación y número de habitaciones.

Características de la Regresión

- Puede ser lineal (relación simple) o no lineal (patrones más complejos).
- Variables explicativas X pueden ser numéricas o categóricas.
- Se usa principalmente para:
  - **Predicción**: estimar valores de Y dado X.
  - **Inferencia**: comprender el efecto de X sobre Y.

¿Qué es Random Forest?

- Algoritmo de aprendizaje de conjunto (ensemble).
- Construye múltiples árboles de decisión a partir de subconjuntos de datos y variables.
- Para regresión, combina los resultados tomando el promedio de las predicciones de todos los árboles.

Ventajas de Random Forest

- Captura relaciones no lineales y complejas.
- Reduce el sobreajuste gracias al promediado de múltiples árboles.
- Es robusto ante ruido y valores atípicos.
- Escalable y aplicable a grandes volúmenes de datos.
- Permite estimar la importancia de las variables predictoras.

Desventajas de Random Forest

- **Complejidad computacional:** Entrenar muchos árboles puede ser costoso en tiempo y memoria.
- **Menor interpretabilidad:** Difícil de explicar en comparación con modelos simples (ej. regresión lineal).
- **Predicciones menos suaves:** Tiende a generar funciones de predicción con saltos o escalones.
- **Dependencia de hiperparámetros:** El desempeño depende de la correcta elección de parámetros como número de árboles y profundidad.
- **No siempre el mejor modelo:** En datasets muy pequeños o con relaciones lineales simples, puede no superar a métodos más básicos.

Generación de Datos Sintéticos

- Se generan 100 puntos en el rango $[0, 5]$.
- La variable objetivo sigue una función seno con ruido gaussiano.

```
import numpy as np

np.random.seed(42)
X = np.sort(5 * np.random.rand(100, 1), axis=0)
y = np.sin(X).ravel() + np.random.normal(0, 0.2, X.
    shape[0])
```

Listing 1: Generación de datos

Entrenamiento del Modelo

- Se divide el conjunto en entrenamiento (80 %) y prueba (20 %).
- Se entrena un RandomForestRegressor con 100 árboles.

```
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.2, random_state=42)

rf = RandomForestRegressor(n_estimators=100,
   random_state=42)
rf.fit(X_train, y_train)
```

Listing 2: Entrenamiento del modelo

Predicción y Visualización

- Se predice para un rango continuo de valores.
- Se grafican datos de entrenamiento, prueba y la predicción.

```
import matplotlib.pyplot as plt

X_plot = np.linspace(0, 5, 500).reshape(-1, 1)
y_pred = rf.predict(X_plot)

plt.scatter(X_train, y_train, c='blue', label='
    Entrenamiento')
plt.scatter(X_test, y_test, c='green', label='Prueba')
plt.plot(X_plot, y_pred, c='red', label='Predicci n
    RF')
plt.xlabel('X')
plt.ylabel('y')
plt.legend()
plt.show()
```

Listing 3: Predicción y gráfico

Grafico

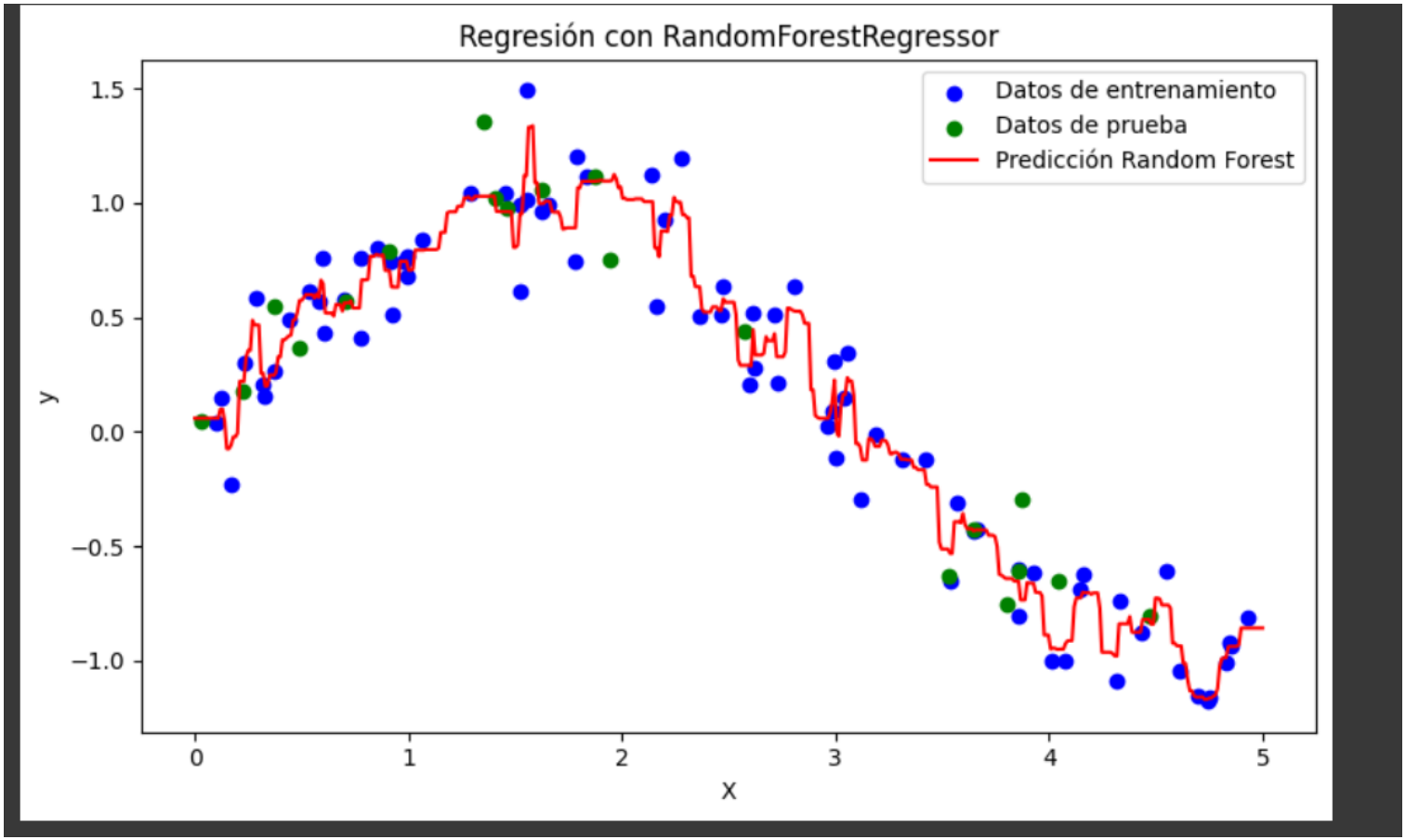

Figura: Predicción de RandomForestRegressor sobre datos sintéticos

Explicación del Resultado

- **Puntos azules:** datos usados para entrenar el modelo.
- **Puntos verdes:** datos de prueba no vistos durante el entrenamiento.
- **Línea roja:** predicción promedio de los 100 árboles.
- El modelo sigue la forma oscilante de $\sin(x)$.
- Captura máximos y mínimos a pesar del ruido, mostrando capacidad para aprender patrones no lineales.
- La predicción presenta pequeños escalones, típicos de los árboles de decisión.
- Gracias al ensamble, se suavizan los saltos y se reduce la varianza.

Conclusiones

- Random Forest es un modelo robusto para problemas de regresión con ruido.
- Permite capturar relaciones no lineales de manera efectiva.
- Reduce el riesgo de sobreajuste al promediar múltiples árboles.
- Generaliza bien y es aplicable a problemas reales como predicción de precios, análisis de riesgo y series temporales.