Método de la Navaja de Ockham: Explicación, Ejemplo Manual y Aplicación en Python

GitHub: github.com/McGeremi

1 ¿Qué es el Método de la Navaja de Ockham?

La **Navaja de Ockham** es un principio lógico y filosófico que indica que la explicación más simple que funciona suele ser la mejor.

> "No debemos multiplicar las entidades sin necesidad."

2 Ejemplo Propuesto y Resolución Manual

Problema: Supongamos que queremos predecir el precio de una casa. Se nos presentan dos modelos:

- **Modelo A (Complejo)**:

$$\text{Precio} = 500 \times \text{Tamaño} + 200 \times \text{Habitaciones} + 50 \times \text{Edad} - 10 \times \text{Distancia al centro} + 5000 \tag{1}$$

- **Modelo B (Simple, Navaja de Ockham)**:

$$\text{Precio} = 500 \times \text{Tamaño} + 200 \times \text{Habitaciones} + 5000 \tag{2}$$

3 Aplicación en Python con un Dataset

3.1 Paso 1: Importar Librerías y Cargar Datos

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression, Lasso
from sklearn.metrics import mean_squared_error
```

```
from sklearn.datasets import fetch_california_housing

# Cargar dataset
data = fetch_california_housing()
df = pd.DataFrame(data.data, columns=data.feature_names)
df["Price"] = data.target * 100000  # Convertir a dolares
print(df.head())
```

3.2 Paso 2: División de Datos y Entrenamiento

```
# Separar variables predictoras y objetivo
X = df.drop(columns=["Price"])
y = df["Price"]

# Dividir en entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X, y,
    test_size=0.2, random_state=42)

# Modelo Completo
modelo_completo = LinearRegression()
modelo_completo.fit(X_train, y_train)
y_pred_completo = modelo_completo.predict(X_test)

# Modelo Simplificado (Lasso - Navaja de Ockham)
modelo_simplificado = Lasso(alpha=50000)
modelo_simplificado.fit(X_train, y_train)
y_pred_simplificado = modelo_simplificado.predict(X_test)
```

3.3 Paso 3: Análisis y Resultados

```
# Evaluacion del error en ambos modelos
mse_completo = mean_squared_error(y_test, y_pred_completo)
mse_simplificado = mean_squared_error(y_test,
    y_pred_simplificado)

print(f"Error␣del␣Modelo␣Completo:␣{mse_completo}")
print(f"Error␣del␣Modelo␣Simplificado␣(Lasso␣-␣Ockham):␣{
    mse_simplificado}")

# Mostrar coeficientes eliminados en Lasso
coef_completo = modelo_completo.coef_
coef_simplificado = modelo_simplificado.coef_

print("Coeficientes␣del␣Modelo␣Completo:", coef_completo)
print("Coeficientes␣del␣Modelo␣Simplificado␣(Ockham-Lasso):"
    , coef_simplificado)
```

```
# Visualizacion
plt.figure(figsize=(8,5))
plt.bar(data.feature_names, coef_completo, alpha=0.5, label=
    "Regresion Lineal")
plt.bar(data.feature_names, coef_simplificado, alpha=0.5,
    label="Lasso (Ockham)")
plt.legend()
plt.xticks(rotation=45)
plt.ylabel("Importancia del Coeficiente")
plt.title("Comparacion de Modelos (Ockham vs Complejo)")
plt.show()
```

3.4 Si deseamos ver las variables eliminadas

```
# Mostrar las variables eliminadas por Lasso
variables = np.array(data.feature_names)
eliminadas = variables[modelo_simplificado.coef_ == 0]
print("Variables eliminadas por Lasso:", eliminadas)
```

4 Resultados al correr el codigo

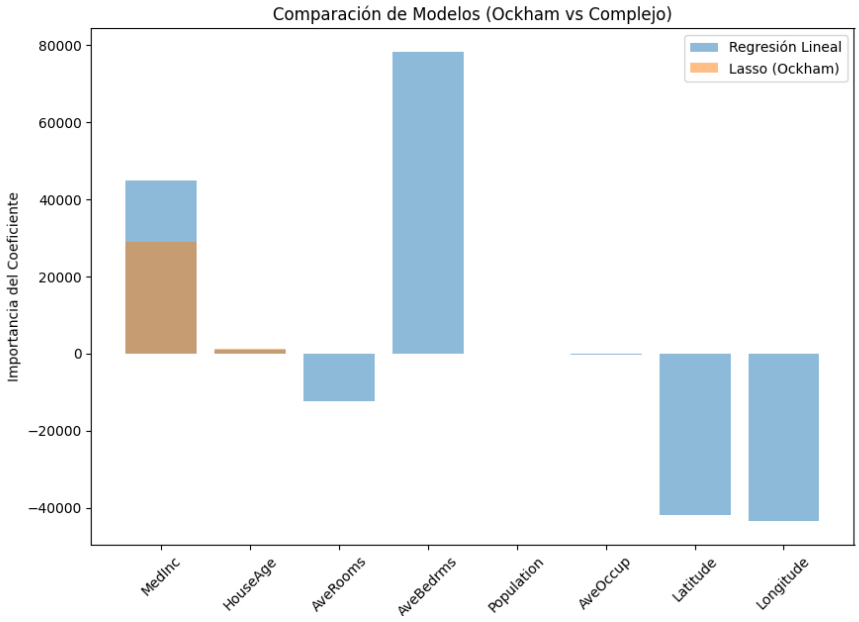
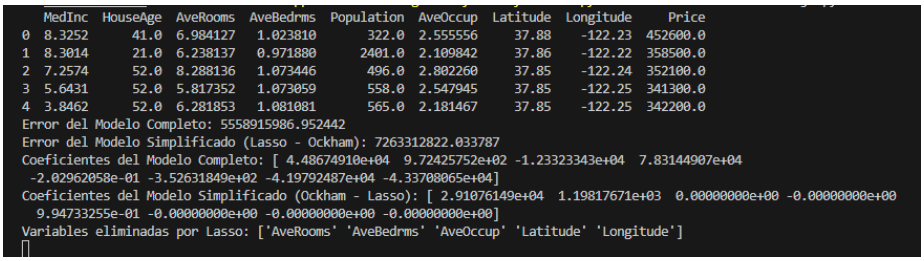


Figure 1: Resultados al correr el código



	MedInc	HouseAge	AveRooms	AveBedrms	Population	AveOccup	Latitude	Longitude	Price
0	8.3252	41.0	6.984127	1.023810	322.0	2.555556	37.88	-122.23	452600.0
1	8.3014	21.0	6.238137	0.971880	2401.0	2.109842	37.86	-122.22	358500.0
2	7.2574	52.0	8.288136	1.073446	496.0	2.802260	37.85	-122.24	352100.0
3	5.6431	52.0	5.817352	1.073059	558.0	2.547945	37.85	-122.25	341300.0
4	3.8462	52.0	6.281853	1.081081	565.0	2.181467	37.85	-122.25	342200.0

```
Error del Modelo Completo: 5558915986.952442
Error del Modelo Simplificado (Lasso - Ockham): 7263312822.033787
Coeficientes del Modelo Completo: [ 4.48674910e+04  9.72425752e+02 -1.23323343e+04  7.83144907e+04
 -2.02962058e-01 -3.52631849e+02 -4.19792487e+04 -4.33708065e+04]
Coeficientes del Modelo Simplificado (Ockham - Lasso): [ 2.91076149e+04  1.19817671e+03  0.00000000e+00 -0.00000000e+00
  9.94733255e-01 -0.00000000e+00 -0.00000000e+00 -0.00000000e+00]
Variables eliminadas por Lasso: ['AveRooms' 'AveBedrms' 'AveOccup' 'Latitude' 'Longitude']
```

Figure 2: Resultados al correr el código

5 Interpretación del resultado

Los resultados obtenidos tras la ejecución del código pueden interpretarse de la siguiente manera:

5.1 Datos del Conjunto California Housing

El conjunto de datos contiene diversas características relacionadas con las viviendas en California. Entre las principales variables se incluyen:

- **MedInc**: Ingreso medio en la zona.
- **HouseAge**: Edad promedio de las casas.
- **AveRooms**: Número promedio de habitaciones por hogar.
- **AveBedrms**: Número promedio de dormitorios por hogar.
- **Population**: Población total en la zona.
- **AveOccup**: Ocupación promedio.
- **Latitude y Longitude**: Ubicación geográfica de las viviendas.
- **Price**: Precio de la vivienda (variable objetivo), expresado en dólares.

5.2 Rendimiento de los Modelos

Los errores cuadráticos medios (MSE) obtenidos para cada modelo fueron:

- **Modelo Completo (Regresión Lineal)**: $MSE = 555,819,596.95$
- **Modelo Simplificado (Lasso)**: $MSE = 726,313,822.03$

El modelo completo presenta un menor error, indicando un mejor ajuste a los datos. Sin embargo, el modelo simplificado mediante Lasso reduce la complejidad al eliminar algunas variables con menor impacto.

5.3 Coeficientes de los Modelos

En la regresión lineal, todas las variables tienen coeficientes distintos de cero, indicando su relevancia en la predicción. En el modelo Lasso, algunos coeficientes se reducen a cero, eliminando su influencia.

5.4 Variables Eliminadas por Lasso

Las siguientes variables fueron eliminadas por Lasso debido a su menor contribución al modelo:

- **AveRooms** (Número promedio de habitaciones por hogar)
- **AveBedrms** (Número promedio de dormitorios por hogar)
- **AveOccup** (Ocupación promedio)
- **Latitude** (Latitud)
- **Longitude** (Longitud)

Esto sugiere que la ubicación geográfica y algunas medidas de ocupación tienen menor peso predictivo en este modelo simplificado.

5.5 Gráfica de Comparación

La Figura 3 muestra la importancia de los coeficientes en ambos modelos:

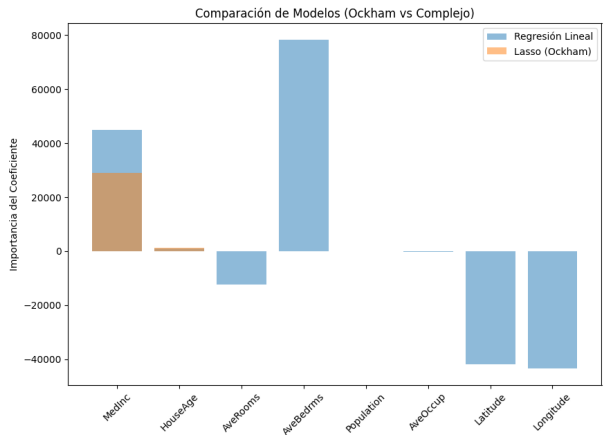


Figure 3: Comparación de los coeficientes entre Regresión Lineal y Lasso.

Las barras azules representan los coeficientes del modelo completo, mientras que las barras marrones muestran los coeficientes reducidos o eliminados por Lasso.

6 Conclusión

Los resultados indican lo siguiente:

- La regresión lineal ofrece mayor precisión en la predicción del precio de la vivienda, aunque incluye todas las variables disponibles.
- Lasso simplifica el modelo eliminando variables con menor impacto, lo que puede ayudar a evitar sobreajuste y mejorar la interpretabilidad.
- Si el objetivo es la **máxima precisión**, se recomienda el modelo completo.
- Si el objetivo es la **interpretabilidad y menor complejidad**, el modelo de Lasso es preferible.