

Universidad Nacional del Altiplano

Facultad de Ingeniería Estadística e Informática

Docente: Fred Torres Cruz

Autor : Ronald Junior Pilco Nuñez

Repositorio : [Regularizacion](#)

Regularización de la Navaja de Occam

1. Introducción

La **Regularización de la Navaja de Occam** es un concepto fundamental en el aprendizaje automático y la estadística, basado en el principio filosófico de la **navaja de Occam**. Este principio establece que, entre varias explicaciones posibles para un fenómeno, la explicación más simple (es decir, la que tiene menos suposiciones) tiende a ser la correcta. En el contexto de los modelos predictivos, esto se traduce en preferir modelos más simples sobre modelos más complejos, siempre que ambos expliquen los datos de manera similar.

En este informe, se explorará en detalle la relación entre la navaja de Occam y los métodos de regularización en el aprendizaje automático, así como sus aplicaciones prácticas.

2. El Principio de la Navaja de Occam

El principio de la navaja de Occam, atribuido al filósofo medieval Guillermo de Ockham, sugiere que, en igualdad de condiciones, la explicación más simple es preferible. En el contexto de los modelos estadísticos y de aprendizaje automático, esto implica que:

- **Modelos simples** (con menos parámetros o características) son preferibles a **modelos complejos** (con muchos parámetros o características).
- Un modelo simple es menos propenso al **sobreajuste** (overfitting), es decir, a adaptarse demasiado a los datos de entrenamiento y perder capacidad de generalización.

3. Regularización en Aprendizaje Automático

La regularización es una técnica que introduce una penalización en la función de pérdida de un modelo para controlar su complejidad. Esto ayuda a prevenir el sobreajuste y a mejorar la generalización del modelo. La **Regularización de la Navaja de Occam** se implementa a través de métodos como L1 (Lasso) y L2 (Ridge), que penalizan los coeficientes del modelo de diferentes maneras.

3.1. Regularización L1 (Lasso)

La regularización L1 añade una penalización proporcional al valor absoluto de los coeficientes del modelo. Esto tiende a producir modelos **dispersos**, es decir, modelos en los que algunos coeficientes son exactamente cero. Matemáticamente, se expresa como:

$$L_{\text{Lasso}}(\theta) = \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^p |\theta_j|$$

■ **Ventajas:**

- Realiza selección de características, eliminando aquellas que no son relevantes.
- Es útil cuando se tienen muchas características y se sospecha que solo unas pocas son importantes.

3.2. Regularización L2 (Ridge)

La regularización L2 añade una penalización proporcional al cuadrado de la magnitud de los coeficientes. Esto reduce la magnitud de los coeficientes sin eliminarlos por completo. Matemáticamente, se expresa como:

$$L_{\text{Ridge}}(\theta) = \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^p \theta_j^2$$

■ **Ventajas:**

- Es útil cuando todas las características son relevantes, pero algunas tienen un impacto menor.
- Ayuda a controlar la multicolinealidad (correlación entre características).

3.3. Regularización Elástica (Elastic Net)

La regularización elástica combina las penalizaciones L1 y L2, lo que permite aprovechar las ventajas de ambos métodos. Matemáticamente, se expresa como:

$$L_{\text{Elastic Net}}(\theta) = \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \lambda_1 \sum_{j=1}^p |\theta_j| + \lambda_2 \sum_{j=1}^p \theta_j^2$$

- **Ventajas:**

- Combina la selección de características de L1 con la estabilidad de L2.
- Es útil cuando se tienen muchas características correlacionadas.

4. Relación con la Navaja de Occam

La **Regularización de la Navaja de Occam** se manifiesta en estos métodos de regularización al penalizar la complejidad del modelo. Al añadir una penalización a los coeficientes, se favorecen modelos más simples que:

- Tienen menos parámetros no nulos (en el caso de L1).
- Tienen coeficientes de menor magnitud (en el caso de L2).

Esto está en línea con el principio de la navaja de Occam, ya que se busca la explicación más simple (modelo más simple) que sea capaz de explicar los datos de manera adecuada.

5. Regularización L1 (Lasso) aplicada a optimización de costos de transporte

Descripción de los datos

El conjunto de datos utilizado simula escenarios de transporte e incluye las siguientes características:

- **Distancia_km**: Distancia en kilómetros.
- **Volumen_toneladas**: Volumen de carga transportada en toneladas.
- **Costo_por_km**: Costo por kilómetro en dólares.
- **Tiempo_horas**: Duración del trayecto en horas.
- **Demanda_destino**: Nivel de demanda en el destino.
- **Frecuencia_uso**: Frecuencia con la que se utiliza la ruta.
- **Estado_carretera**: Calidad del estado de la carretera (escala numérica).
- **Peajes_dolares**: Costo de peajes en dólares.

La variable objetivo es **Costo_total_dolares**, que representa el costo total del transporte.

Técnica aplicada: Regularización L1

La regularización L1 agrega una penalización proporcional a la suma de los valores absolutos de los coeficientes del modelo. Esto tiene dos efectos principales:

- Reduce la magnitud de los coeficientes menos importantes.
- Puede establecer algunos coeficientes exactamente en cero, lo que equivale a eliminar esas características del modelo.

El modelo utilizado es una regresión lineal penalizada con el parámetro de regularización $\alpha = 100$, ajustado experimentalmente.

Código utilizado

A continuación se presenta el código utilizado para implementar la técnica:

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import Lasso
from sklearn.metrics import mean_squared_error
import matplotlib.pyplot as plt
import seaborn as sns

# 1. Cargar la base de datos
file_path = "datos_transporte_optim.csv"
data = pd.read_csv(file_path)

# Separar características y variable objetivo
X = data.drop(columns=["Costo_total_dolares"])
y = data["Costo_total_dolares"]

# 2. Dividir en conjuntos de entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)

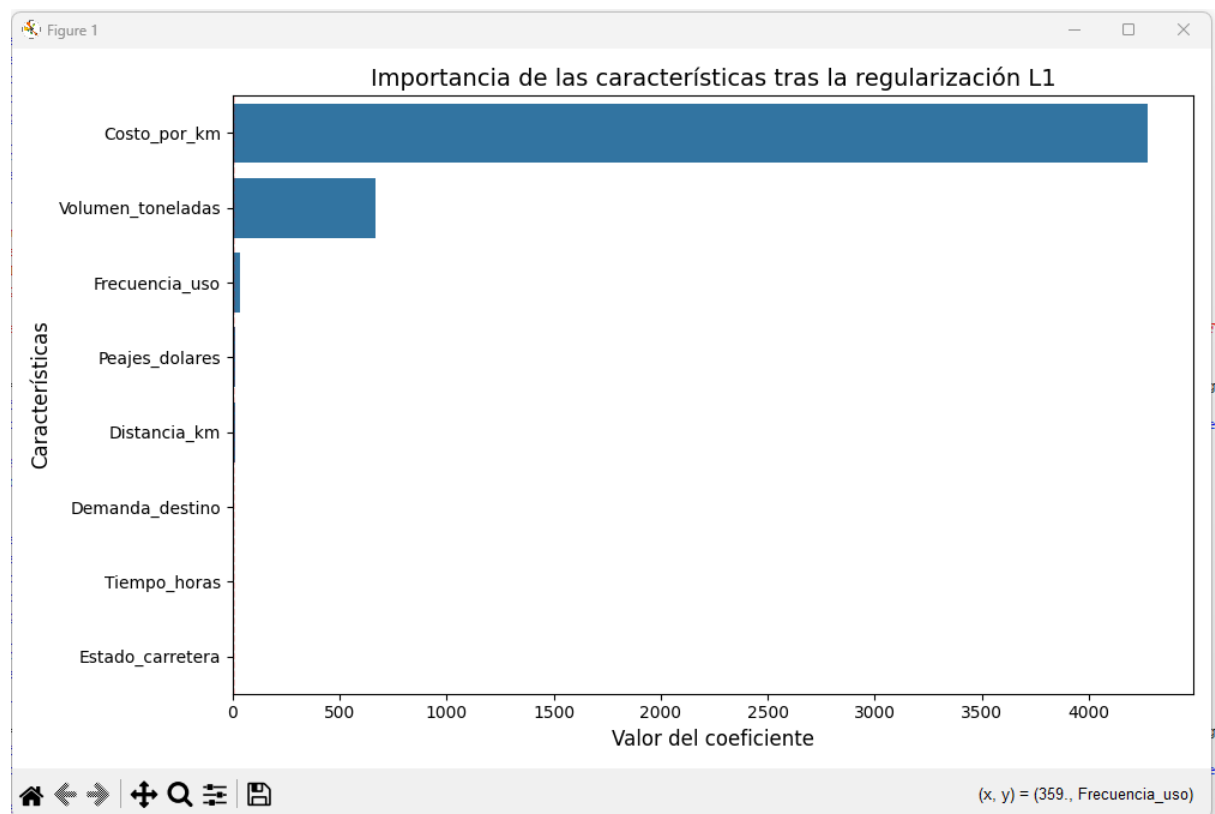
# 3. Entrenar modelo Lasso
alpha = 100
lasso = Lasso(alpha=alpha, random_state=42, max_iter=10000)
lasso.fit(X_train, y_train)

# 4. Identificar características seleccionadas
coef = lasso.coef_
selected_features = X.columns[coef != 0]

# 5. Evaluar el modelo
y_pred = lasso.predict(X_test)
mse = mean_squared_error(y_test, y_pred)

# 6. Visualizar coeficientes
coef_df = pd.DataFrame({"Característica": X.columns, "Coeficiente": coef})
plt.figure(figsize=(10, 6))
sns.barplot(data=coef_df, x="Coeficiente", y="Característica", dodge=False)
plt.title("Importancia de las características tras la regularización L1")
plt.xlabel("Valor del coeficiente")
plt.ylabel("Características")
plt.axvline(0, color="red", linestyle="--")
plt.tight_layout()
plt.show()
```

Resultados



Los resultados obtenidos tras entrenar el modelo y evaluar su rendimiento son los siguientes:

- **Características seleccionadas por Lasso:** Distancia_km, Volumen_toneladas, Costo_por_km, Demanda_destino, Frecuencia_uso, Peajes_dolares.

- **Coefficientes del modelo:**

Característica	Coefficiente
Distancia_km	13.60
Volumen_toneladas	666.48
Costo_por_km	4273.64
Tiempo_horas	0.00
Demanda_destino	7.56
Frecuencia_uso	33.64
Estado_carretera	0.00
Peajes_dolares	13.93

- **Error cuadrático medio (MSE):** 6,538,092.28

Conclusion de resultados

- **Características clave:**
 - *Costo_por_km* (4273.64) y *Volumen_toneladas* (666.48) son las variables que más impactan en el costo total del transporte.
 - *Frecuencia_uso*, *Peajes_dolares*, *Distancia_km* y *Demanda_destino* también influyen, aunque en menor medida.
- **Variables descartadas:** *Tiempo_horas* y *Estado_carretera* no aportan significativamente y fueron eliminadas por Lasso.
- **Modelo:** El modelo es más simple y generalizable, enfocándose solo en las características relevantes. El *MSE* obtenido es de 6538092.28, lo que refleja el nivel de error promedio del modelo, el cual podría optimizarse ajustando el parámetro de regularización.
- **Conclusión:** La regularización L1 identificó las variables más importantes, simplificando el análisis y mejorando la interpretabilidad del modelo.

Referencias

- [1] Tibshirani, R. (1996). Regression shrinkage and selection via the Lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1), 267-288. Disponible en: <https://doi.org/10.1111/j.2517-6161.1996.tb02080.x>
- [2] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12, 2825-2830. Disponible en: <https://scikit-learn.org/stable/>
- [3] Ballou, R. H. (2004). *Business Logistics/Supply Chain Management: Planning, Organizing, and Controlling the Supply Chain*. Pearson Education. Un enfoque profundo sobre los costos logísticos y su optimización en sistemas de transporte.
- [4] Han, J., Kamber, M., Pei, J. (2012). *Data Mining: Concepts and Techniques*. Elsevier. Incluye métodos de preprocesamiento de datos y selección de características.