🔍 L1 & L2 Regularization: Making Machine Learning Models Robust 🚀

When working with machine learning models, overfitting is a common challenge. 📉 Regularization techniques like L1 and L2 help to combat this by penalizing model complexity. Let's dive in! 🧑‍💻 ✨

💡 What is Regularization?

Regularization is a technique used to prevent overfitting by adding a penalty to the model's loss function. This penalty discourages the model from fitting the noise in the training data, ensuring better generalization to unseen data

🧮 L1 Regularization (Lasso)

Adds a penalty proportional to the absolute value of coefficients.

$Loss = MSE + \lambda \sum |wi|$

Effect:

Shrinks some coefficients to zero, effectively performing feature selection. 🔍

When to Use:

When you suspect many features are irrelevant or unnecessary.

🧮 L2 Regularization (Ridge)

Adds a penalty proportional to the square of the coefficients.

$Loss = MSE + \lambda \sum wi^2$

Effect:

Shrinks coefficients towards zero but doesn't eliminate them entirely.

Ensures smooth solutions and reduces model complexity. 🌊

When to Use:
When all features are important but need regularization to prevent overfitting.

🔑 **Takeaway:**
Use L1 when you want to reduce features and perform feature selection. ☑️
Use L2 when you want a smooth and stable model that doesn't overfit. 🌟
In practice, try ElasticNet, which combines the benefits of both.

GitHub Code: https://github.com/NafisAnsari786/Machine-Learning-Algorithms/blob/main/15%20L1%20%26%20L2%20%20Regularization/L1%20and%20L2%20Regularization.ipynb