

Days 130 & 131: Data Science Learning Update

Over the past couple of days, I delved into critical machine learning concepts, with a focus on **Gradient Descent** and its variations. Here's what I covered:

What is Gradient Descent?

Gradient Descent is an optimization algorithm used to minimize the cost function in machine learning models. By iteratively adjusting model parameters, it aims to find the lowest point on the error surface, leading to better predictions over time.

Types of Gradient Descent:

1. **Batch Gradient Descent (BGD):**

This method processes the entire dataset in each iteration to compute gradients. It's stable but can be slow on large datasets.

2. **Stochastic Gradient Descent (SGD):**

SGD updates model parameters using a single training example at a time, speeding up training but introducing fluctuations.

3. **Mini-batch Gradient Descent:**


A balanced approach that processes small data batches, combining speed and stability.



Mathematical Formulation of BGD:

I delved into the math behind Batch Gradient Descent, understanding how to calculate gradients of the cost function and update parameters to minimize error.

GDRegressor Class:

Excitingly, I implemented my own Gradient Descent-based regressor—the **GDRegressor**. Translating theory into Python code, I created a custom class for linear regression using the BGD algorithm. Witnessing optimization algorithms in action was enlightening!

The learning journey continues, and I eagerly anticipate applying these concepts in real-world projects. 

Keep up the momentum, and feel free to reach out if you have any questions or need further guidance!   .