**Compaction Optimization Strategy**

**Goal:**

Efficiently select which partitions to compact under a limited compute budget, balancing performance gains and compaction costs.

**Summary of Approach:**

We designed a data-driven model to decide which partitions to compact by:

1. **Defining a Performance Penalty Function**

Penalty increases with the number of small files, delete files, and low access frequency.

**Formula:**

Penalty = ( w1 \* small\_file\_count + w2 \* delete\_file\_count \* avg\_delete\_file\_size\_mb ) \* partition\_access\_frequency

w1 = 2 , w2 = 0.3

1. **Defining a Cost Model**

Cost increases with total file count and estimated size.

Formula:

Cost = file\_count \*avg\_file\_size\_mb \* avg\_delete\_file\_size\_mb \* w3

1. **Selection Logic / Scoring Algorithm**

We compute:

Score = Penalty / ( Cost + 1e-6)

Partitions are **ranked by score**, and we compact top-ranked partitions until the **total cost reaches the compute budget**.

**Trade-Offs Considered:**

* **Performance vs. Cost:**  
  Compacting high-penalty partitions isn't always optimal if they are too costly.
* **Greedy selection:**  
  We use a greedy algorithm that maximizes penalty reduction per unit cost.
* **Access Frequency Impact:**  
  Frequently accessed partitions are penalized more for inefficiency.

This approach can easily adapt to evolving workload patterns by updating penalty weights or learning them from real system feedback.

**Simulation Results (Sample Dataset)**

* **Total Partitions:** 20
* **Compute Budget:** 3000 mins
* **Partitions Selected:** 18
* **Total Cost Used:** 2926.13
* **Estimated Performance Gain (Penalty Avoided):** 4682571.39
* **Strategy:** Score-based greedy selection

**Conclusion**

By scoring each partition using a performance-to-cost ratio and selecting under budget constraints, we optimize compaction for maximum system performance improvement with minimal compute usage. Like cleaning the busiest rooms in a house first — we prioritize messy, high-traffic partitions to improve overall efficiency.