**Qn1: Softprob gives probabilities, so the final output can still be close to the true class.**

And that’s **true** in terms of the **expected prediction** — but **the loss function itself** doesn’t account for the **distance** between the classes.

|  |  |  |
| --- | --- | --- |
| **Aspect** | **Cross-Entropy Loss** | **Earth Mover’s Distance (EMD) Loss** |
| **What it measures** | How well predicted probabilities match truth | How much “effort” is needed to transform prediction into truth |
| **Treats classes as** | **Unrelated categories** (categorical) | **Ordered positions** (ordinal) |
| **Sensitive to distance?** | ❌ No — class 1 vs 5 is same as 1 vs 2 | ✅ Yes — wrong by 4 bins is worse than by 1 |
| **Use case fit** | Good for pure classification | Ideal for **ordinal prediction** with meaningful class order |
| **Loss penalty style** | Hard penalty for any wrong class | Gradual penalty based on how far off |

If you predict bin 1 instead of bin 2, EMD gives a small penalty. If you predict bin 4 instead of bin 2, EMD gives a larger penalty. Cross-entropy would penalize both equally if they assign the same probability to bin 2.

**Earth Mover’s Distance is inspired by the idea of how much "effort" it takes to transform one distribution into another.**

**Imagine your model’s predicted probabilities are like piles of sand spread across the bins, and the correct answer is a single pile sitting entirely on the true bin.  
EMD measures how far and how much of that sand you’d need to shift to perfectly match the true label — the farther you move it, the higher the cost.**

I used fit\_transform on the training set, so the scaler learned the scaling parameters only from training data. Then I used transform on the validation and test sets using those same learned parameters. This prevents data leakage and ensures a fair evaluation.

 To **simplify the modeling** task, especially in noisy or hard-to-predict settings.

 To **capture ordinal structure** — for example, predicting whether utilization is *low, medium, or high* is often sufficient for operational decision-making.

My RMSE is unitless, since the target is a ratio. So a value of 0.22 means the model is off by 0.22 on average in absolute terms, or about 22 percentage points.