DA5401 Data Challenge: Semantic Similarity in AI Evaluation

S Shriprasad
Roll No: DA25E054

November 21, 2025

1 Problem Statement Objective

The core of this challenge is **Metric Learning**. We are tasked with taking a User Prompt, a System Response, and a specific Metric Definition, and mapping them to a 'Fitness Score' (0-10).

Evaluating AI responses is subjective. A response might be "fluent" but "unsafe." Therefore, the score is not just about the quality of the text, but the **alignment** between the text and the specific metric definition.

2 Exploratory Data Analysis (Geometric Textual)

2.1 Metric Space Visualization

Unlike standard EDA, I wanted to understand if the provided metrics cluster logically. I converted the metric names into embeddings and plotted them using PCA.

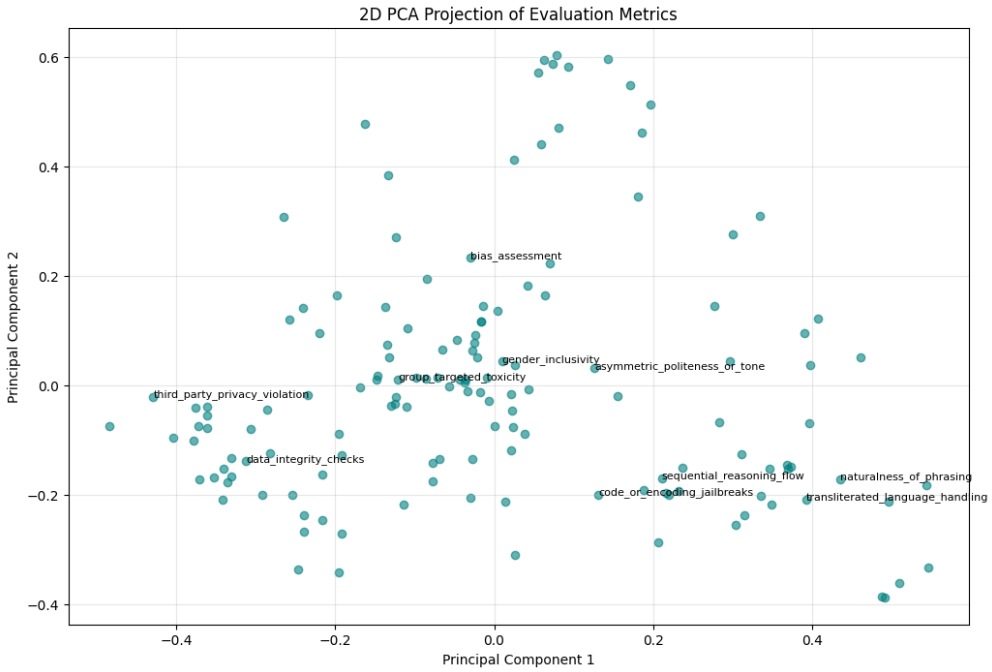

Figure 1: 2D PCA Projection of Evaluation Metrics. Distinct clusters are visible.

Observation: As shown in Figure 1, metrics related to similar topics group together. This confirms that the semantic embeddings contain useful signals.

2.2 Token Count vs. Score

I checked if writing a longer answer guarantees a higher score.

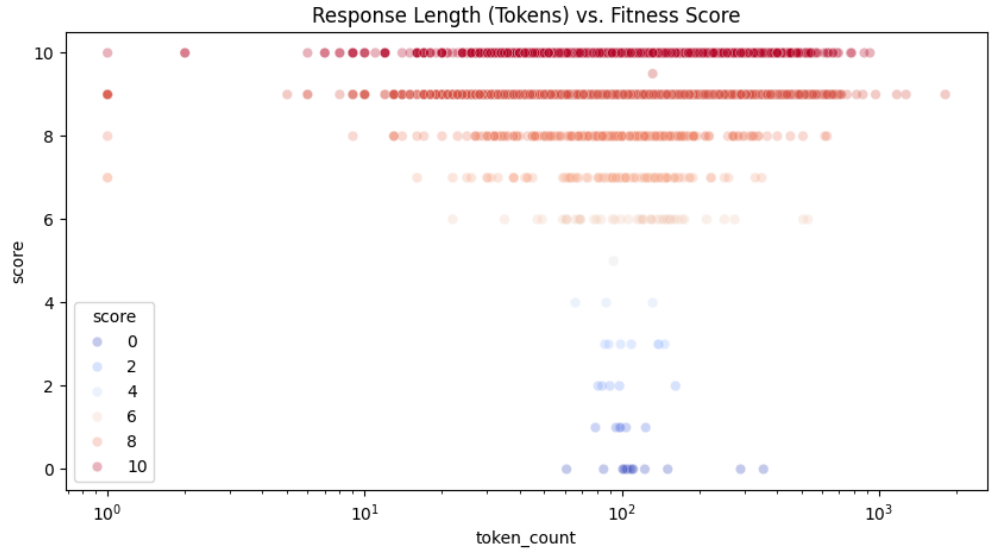

Figure 2: Response Length vs. Fitness Score. There is almost no correlation.

Insight: The correlation is near zero (0.017). A very short response can get a 10, and a long one can get a 0. We cannot rely on simple text statistics.

2.3 Language and Category Analysis

I checked the score distribution across languages and metric categories.

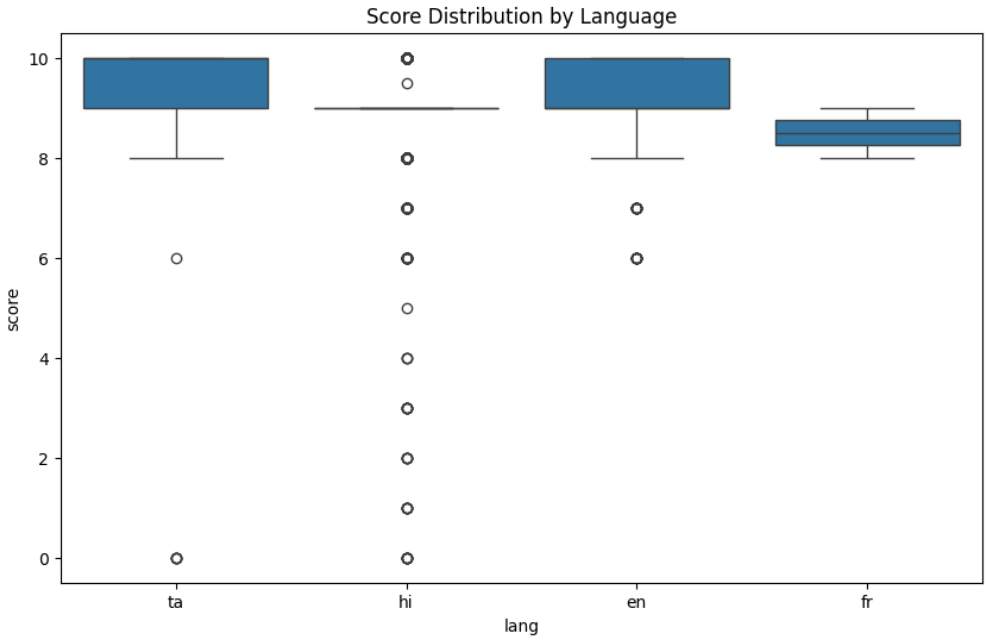

Figure 3: Score distribution by Language. It is consistent across English, Hindi, and Tamil.

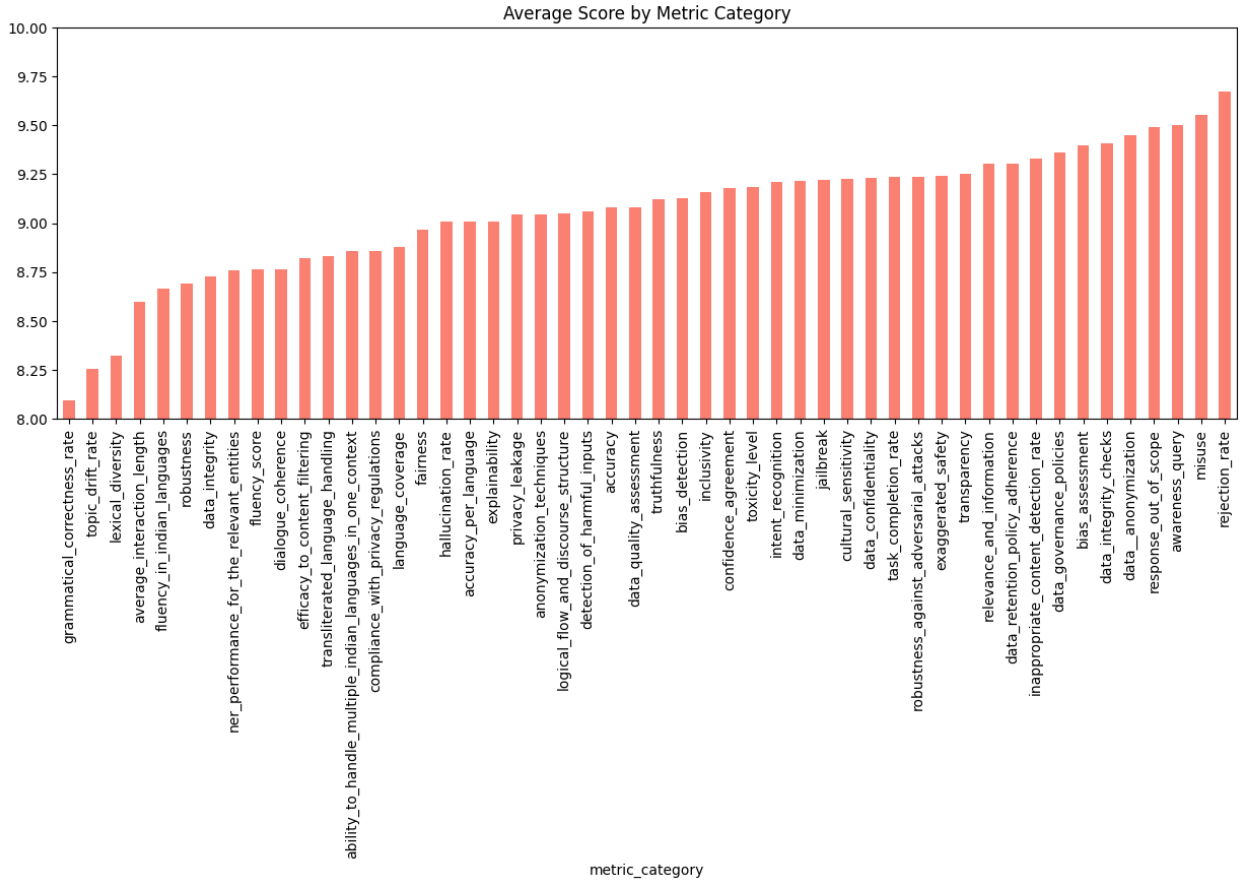

Figure 4: Average scores by metric category are consistently high (8.8+).

2.4 The "Perfect Score" Problem

The most critical finding came from my comprehensive statistical dashboard.

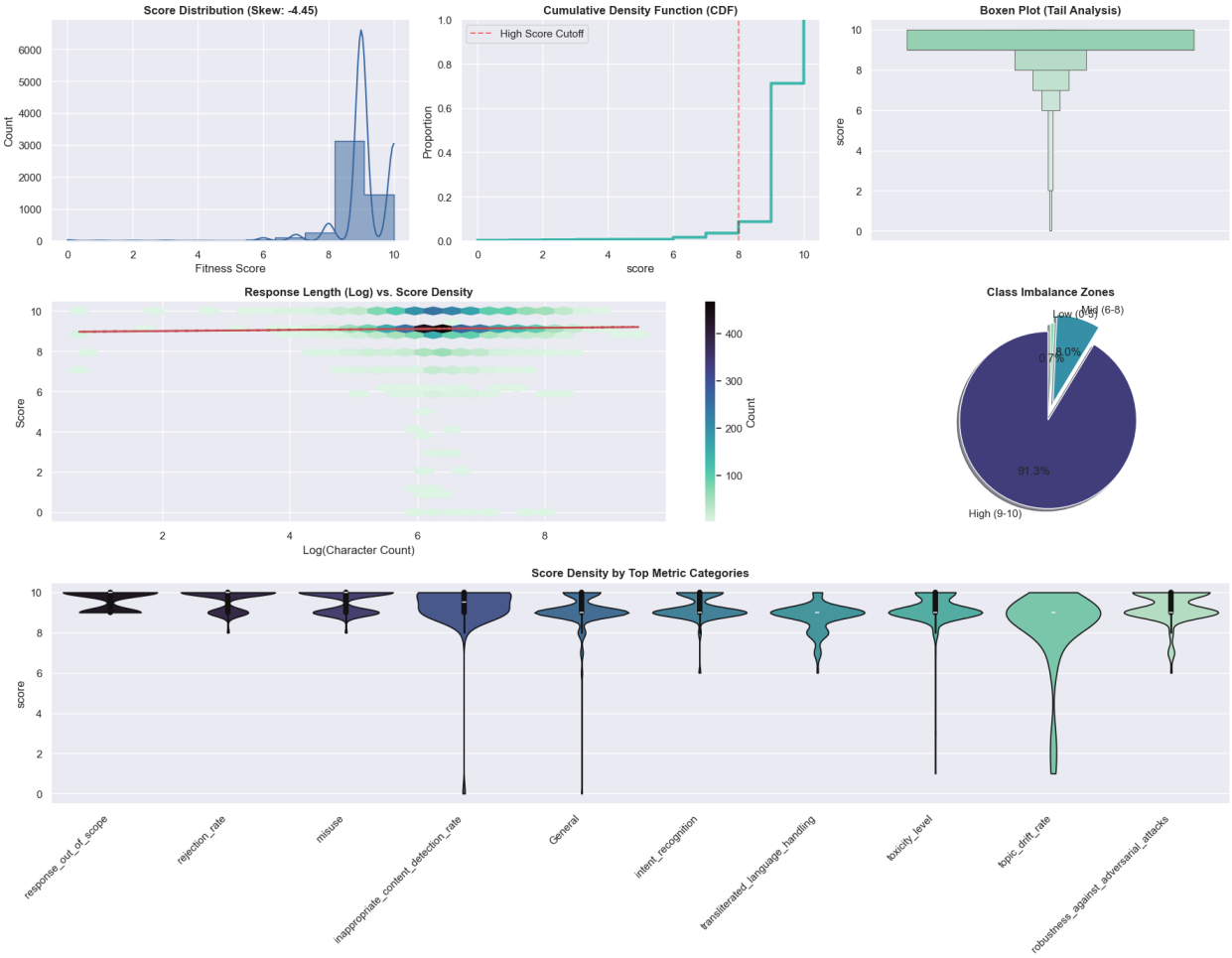

Figure 5: Detailed Statistical EDA. The top-left histogram shows the extreme skew.

The data represents an **Anomaly Detection** problem.

- **High (9-10):** 91.3% of the data.
- **Low (0-5):** Only 0.7% of the data.

This means a standard model will struggle to learn what a "bad" response looks like.

3 Experimental Phase: Siamese Networks

My initial hypothesis was to treat this as a **Semantic Similarity** problem. I attempted to use a **Siamese Network** architecture (often used for face recognition) to learn the distance between the Metric vector and the Response vector.

Why it Failed:

1. **Lack of Negatives:** Siamese networks need pairs of "Matches" and "Mismatches." Since 90% of our data is a "Match" (High Score), the network collapsed.

2. **Instability:** The model plateaued at an RMSE of $\sim$3.0, essentially predicting the mean for everything.

4 Final Methodology: Feature Engineering + XGBoost

I pivoted to a "Feature Engineering first" strategy using **XGBoost**.

4.1 1. Advanced Embeddings

I used 'paraphrase-multilingual-mpnet-base-v2' instead of standard BERT. This model is better at understanding paraphrases, which is crucial for checking if a response matches a metric description.

4.2 2. Engineered Features

I created a feature vector consisting of:

- **Cosine Similarity:** $\cos(\text{Metric}, \text{Response})$
- **Euclidean Distance:** Physical distance between vectors.
- **Prompt Alignment:** Similarity between the User Prompt and the Metric.

4.3 3. Synthetic Negative Sampling

To solve the class imbalance found in the EDA, I created **Synthetic Negatives**:

- I took high-scoring pairs.
- I swapped the Metric Definition with a random, unrelated metric.
- I assigned these new pairs a low score (0-3).

This forced the model to learn what a mismatch looks like.

5 Results and Conclusion

I trained an XGBoost Regressor on these features.

- **Final Validation RMSE:** $\sim$2.05
- **Key Feature:** Cosine Similarity between Response and Metric was the most important predictor.

While the Siamese Network is the theoretically "correct" architecture for metric learning, the small data volume (5k samples) made it fail. The **XGBoost** approach, treating similarity scores as engineered features and using synthetic data to fix the imbalance, proved far more robust.