Statistical Consultation Request

Converting Mean Absolute Error to User-Friendly Accuracy Scores

1. Executive Summary

I'm building an application that evaluates prediction accuracy using Mean Absolute Error (MAE) as the core metric. The MAE values have a wide range (3.6 to 62.7) and I need to convert them to a 0-100 scale that's intuitive for end users. I'm seeking your guidance on selecting the most appropriate transformation function given the data characteristics and design goals outlined below.

2. Problem Statement

I have a dataset of 323 entities where prediction accuracy is measured using Mean Absolute Error (MAE) of rank predictions. I need to transform these MAE values into a 0-100 'accuracy score' that:

- Is intuitive for non-technical users
- Rewards top performers (low MAE) with scores approaching 100
- Provides meaningful differentiation across the performance spectrum
- Handles the wide variance across different subgroups (discussed below)

3. Data Characteristics

3.1 Overall Distribution (n=323)

Statistic	Value	Statistic	Value
Minimum	3.57	Mean	19.93
25th Percentile	11.31	Median	17.00
75th Percentile	25.58	Maximum	62.67
Standard Deviation	11.05	Range	59.10

3.2 Subgroup Characteristics

The dataset contains four distinct subgroups with significantly different MAE distributions:

Subgroup	n	Mean MAE	Median MAE	Range
Group A	36	9.19	8.93	3.6 - 18.0

Subgroup	n	Mean MAE	Median MAE	Range
Group B	89	15.31	14.00	3.9 - 40.3
Group C	66	16.18	14.23	5.3 - 44.0
Group D	132	27.85	26.62	4.7 - 62.7

Note: Group A has ~3× lower MAE than Group D, indicating significantly different prediction difficulty across subgroups.

4. Transformation Function Options

I've identified five candidate transformation functions. All convert MAE → Score where lower MAE yields higher scores:

Option 1: Linear with Floor

`Formula: score = max(10, 100 - MAE × 1.2)`

Characteristics:

- Linear relationship, guaranteed minimum score of 10
- Score range: 25-96 (best to worst)
- Median entity scores 80

Option 2: Power Function

`Formula: score = max(0, 100 - (MAE × 0.5)^1.3)`

Characteristics:

- Non-linear with gentle compression
- Score range: 12-98 (best to worst)
- Median entity scores 84
- Rewards top performers (compresses scores at high end)

Option 3: Exponential Decay

`Formula: score = 100 × exp(-MAE / 15)`

Characteristics:

- Smooth exponential decay
- Score range: 2-79 (best to worst)
- Median entity scores only 32 (may feel too harsh)

Option 4: Square Root Compression

`Formula: score = max(0, 100 - √MAE × 10)`

Characteristics:

- Moderate non-linear compression

- Score range: 21-81 (best to worst)
- Median entity scores 59

Option 5: Percentile-Based

`Formula: score = 100 - percentile_rank(MAE)`

Characteristics:

- Relative ranking within dataset
- Score range: 0-100 (perfectly distributed)
- Median entity scores exactly 50
- Automatically accounts for subgroup differences

5. Design Considerations

5.1 User Psychology

The target users are non-technical but expect scores similar to academic grading (90+ = excellent, 70-80 = average, <60 = poor). This creates a preference for transformations that keep most entities in the 70-90 range.

5.2 Subgroup Fairness

Group D entities are inherently 3× harder to predict than Group A entities. Should the scoring function:

1. **Apply uniformly (Group D entities would score systematically lower)**
2. **Adjust for subgroup difficulty (normalize within each group)**

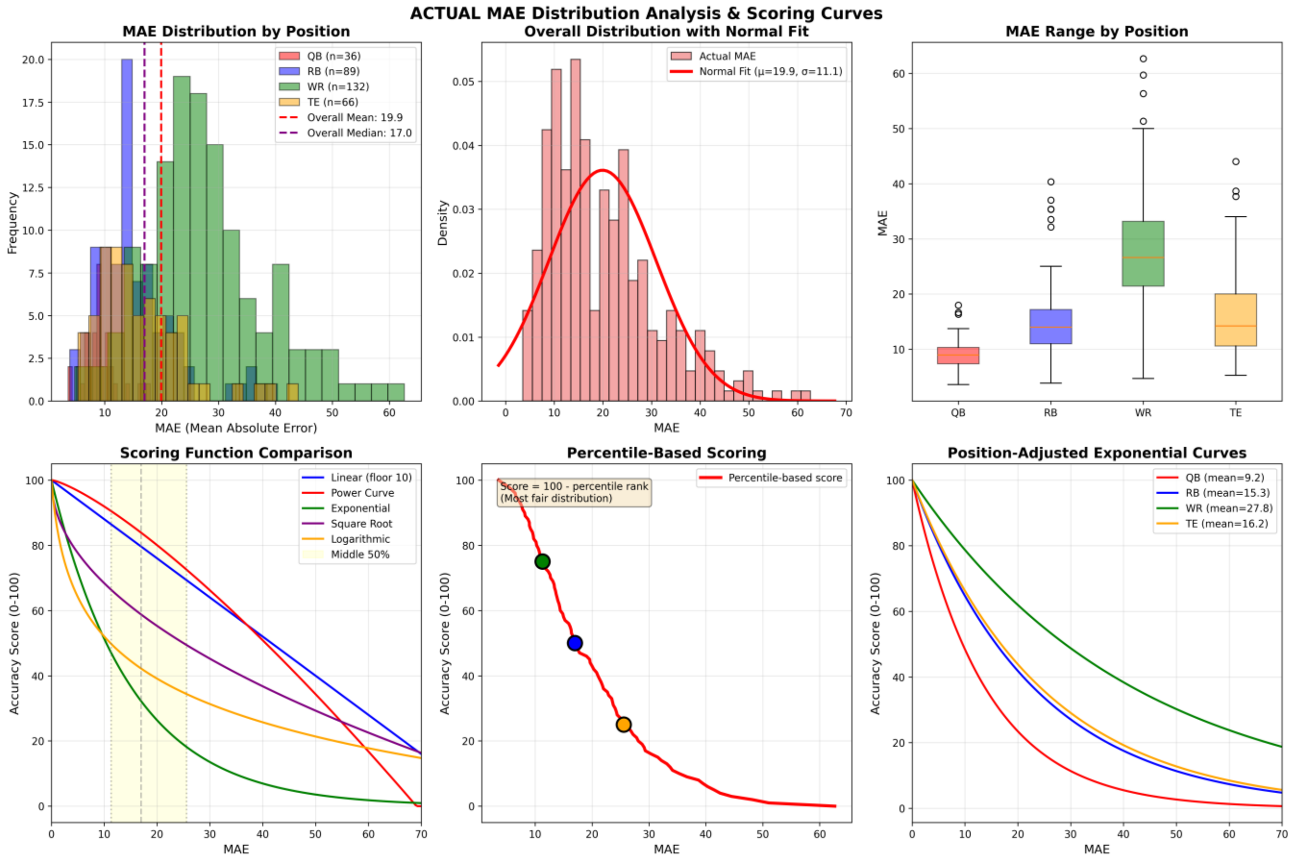
Option (1) is simpler but may feel unfair. Option (2) requires separate scoring per subgroup or a position-adjusted formula.

5.3 Score Interpretability

Users should be able to roughly understand: 'What does a score of 75 mean?' Options 1-2 keep median scores high (80-84), which may inflate perception. Option 3 is potentially too harsh. Option 5 offers clear interpretation (50 = median performer) but loses absolute meaning.

6. Visual Analysis

The attached visualization shows the MAE distribution and how each transformation function maps MAE values to scores. Key observations:



- Top left: MAE distribution by subgroup shows clear separation
- Top middle: Overall distribution approximates normality with right skew
- Bottom left: Scoring curves show dramatic differences in how each function maps MAE to scores
- Bottom middle: Percentile-based provides stepwise but uniform distribution

7. Questions for Expert Input

Given the data characteristics and design goals, I would appreciate your perspective on:

1. **Which transformation function best balances user-friendliness with statistical soundness?**
2. **Should I normalize scores within subgroups given the 3× MAE difference, or accept that some subgroups are inherently harder to predict?**
3. **Are there any statistical considerations I'm overlooking in this transformation problem?**
4. **Would you recommend a different transformation function not listed here?**

Thank you for your time and expertise. I'm happy to provide additional details or clarifications as needed.