# SHL Assessment Recommendation System

## 1 INTRODUCTION

This report details the technical implementation of the **SHL Assessment Recommendation System**, a hybrid AI solution designed to map unstructured job descriptions to precise entries in the SHL assessment catalogue. The core challenge was "Vocabulary Mismatch"—users search for semantic concepts (e.g., "collaborative coder"), while the database contains technical titles (e.g., "Java 8 Standard"). We addressed this using a multi-stage retrieval and ranking pipeline.

## 2 METHODOLOGY: THE HYBRID SCORING MODEL

Our system does not rely on a single retrieval method. Instead, it employs a weighted hybrid approach that progressively filters and ranks candidates.

### 2.1 Phase 1: Retrieval (Weighted Vector Search)

The initial retrieval captures the "Semantic Intent" of the query.

1. **Vectorization:** We use the `BAAI/bge-small-en-v1.5` model to generate 384-dimensional vectors for both queries and assessments.
2. **Input Engineering:** To solve the issue where generic descriptions diluted specific test names, we engineered the input string $D$ for each document:

$$D = (\text{Name} \times 3) + \text{Description} \tag{1}$$

   Tripling the name weight forces the embedding model to prioritize the specific assessment title (e.g., "Java") over generic keywords in the description.
3. **Retrieval:** We calculate the Cosine Similarity $S_{vec}$ between Query vector $Q$ and Document vector $D$. The top 50 candidates are passed to the next phase.

### 2.2 Phase 2: Neural Reranking and Boosting

The raw vector score is often insufficient for precise ranking. We implemented a secondary scoring layer using a Cross-Encoder and deterministic boosts.

**The Scoring Formula:** The final relevance score $S_{final}$ for each candidate is calculated as:

$$S_{final} = S_{cross\_encoder} + (W_{kw} \times K_{overlap}) + (W_{name} \times N_{overlap}) \tag{2}$$

Where:

- $S_{cross\_encoder}$: The output logit from the Cross-Encoder model (hosted on Hugging Face). This measures deep semantic relevance.
- $K_{overlap}$: The ratio of query keywords found in the document text (0.0 to 1.0).
- $N_{overlap}$: The ratio of query keywords found strictly in the *Assessment Name*.
- $W_{kw} = 3.0$: Weight for general keyword boosting.
- $W_{name} = 5.0$: Weight for name-specific boosting.

**Rationale:**

- The **Cross-Encoder** handles the nuance (e.g., distinguishing between "Sales Manager" and "Sales Graduate").

- The **Keyword Boost** ($W_{kw}$) ensures that if a user types a rare specific skill (e.g., "ReactJS"), documents containing that exact word get a significant lift.
- The **Name Boost** ($W_{name}$) is the strongest signal. If the user's query exactly matches words in the title, that document is pushed to the absolute top of the list.

## 3 DATA INTEGRITY OPTIMIZATION

Algorithm accuracy is meaningless without data integrity. We identified that subtle URL mismatches were causing valid predictions to be flagged as incorrect during evaluation.

### 3.1 Manual Verification and URL Normalization

We performed a manual audit of the dataset to map "Crawl URLs" to "Ground Truth URLs." Discrepancies often involved URL encoding (e.g., `%20` vs `-`). We implemented a normalization layer:

- **Input:** `.../shl-test-solutions/verify%20interactive`
- **Output:** `.../shl-test-solutions/verify-interactive-g`

This ensured that 100% of valid algorithmic recommendations were correctly credited in our Recall metrics.

## 4 SYSTEM ARCHITECTURE

The solution is deployed as a scalable microservice architecture:

- **API Layer:** FastAPI (Python) handling async requests.
- **Vector Store:** Supabase (PostgreSQL + pgvector).
- **Reranker Service:** A dedicated Hugging Face Space running 'cross-encoder/ms-marco-MiniLM-L-6-v2'. Offloading this model reduced valid backend memory footprint by ~400MB, preventing OOM errors on the free tier.

## 5 CONCLUSION

By combining a weighted hybrid retrieval strategy with precise deterministic boosting, the SHL Assessment Recommendation System delivers results that are both semantically rich and keyword-accurate. The implementation of the $S_{final}$ scoring formula allows fine-grained control over the recommendation logic, ensuring that hiring managers receive the most relevant assessment options for any given job description.