SHL GenAI Assessment Recommendation System

Approach & Solution Overview

Meka . Durga sai vardhan reddy | December 2025

1. Problem Understanding

The objective is to design a system that recommends the most suitable SHL assessments for hiring requirements expressed as free-form text. Traditional keyword-based matching is insufficient because job descriptions and assessment names often use different terminology. The core challenge is to understand the **semantic intent** of hiring requirements, match them accurately against a large catalog, and provide **relevant, explainable, and reproducible** results.

2. Solution Overview

I implemented a **semantic recommendation system** using vector embeddings and similarity search, following a **Retrieval-Augmented Recommendation (RAR)** approach where recommendations are driven by semantic retrieval rather than keyword-based logic.

> **Pipeline:** Text Embedding → FAISS Vector Search → Metadata Enrichment → Deterministic Explanation → UI Display

3. Data Preparation & Representation

Each SHL assessment is represented using structured metadata: **Assessment Name, Description, Duration, Test Type, Remote Support, and Adaptive Support**. To enable semantic comparison, assessment text is converted into **384-dimensional dense vectors** using `SentenceTransformer (all-MiniLM-L6-v2)`. All embeddings are stored in a **FAISS IndexFlatL2** for fast nearest-neighbor search.

4. Semantic Retrieval Using FAISS

When a user provides a hiring requirement, the text is encoded into a vector and queried against the FAISS index to retrieve the **top-K semantically similar** assessments. FAISS was chosen because it provides sub-millisecond retrieval, scales well with large catalogs, and produces deterministic, reproducible results.

5. Recommendation & Explanation Logic

Once top-K assessments are retrieved, the system enriches results with metadata. Instead of relying on an external LLM at inference time, I implemented a **deterministic explanation layer** that generates structured explanations for why each assessment is suitable. This ensures **reproducibility, auditability, and deployment stability** without external API dependencies.

6. System Architecture & Deployment Strategy

Local / Offline Mode (API + UI)

- Flask REST API (`api/app.py`) with `/recommend` endpoint
- Streamlit UI (`ui/app.py`) connects to API
- Ideal for modular development and testing

Cloud Mode (Single Streamlit App)

- Unified Streamlit app (`app.py`) embeds retrieval logic
- Complies with Streamlit Cloud constraints
- No backend server required

🔗 **Live Demo:** https://mdsvr-shl-genai-assessment-recommendation-system-app-d5f5f2.streamlit.app/

7. CI/CD and Engineering Practices

A **GitHub Actions CI pipeline** (`.github/workflows/ci.yml`) was implemented to ensure code reliability:

- Installs dependencies from `requirements.txt`
- Validates Python syntax using `compileall`
- Verifies FAISS index and metadata files exist
- Performs smoke test on core components (FAISS, SentenceTransformer)

```
# Smoke test excerpt from ci.yml python - <
```

8. Evaluation & Prediction Submission

For the final submission, predictions are generated by running the trained FAISS retrieval system on the provided test dataset. For each query, the **top-ranked assessment** is selected as the model's prediction and exported to vardhan_reddy.csv. This ensures no training data leakage, fully automated and reproducible predictions, and compliance with the submission format.

9. Key Design Decisions & Justification

Decision	Justification
Semantic search over keyword matching	Improves accuracy and handles varied terminology
FAISS-based retrieval	Enables scalable and fast similarity search
Deterministic explanations	Ensures transparency, auditability, reproducibility
Single-process cloud deployment	Improves stability on Streamlit Cloud
CI/CD integration	Demonstrates engineering maturity, prevents regressions

10. Conclusion

> The final solution delivers a **robust, explainable, and deployment-ready** assessment recommendation system. By combining semantic embeddings with efficient vector search and clean architectural design, the system effectively maps hiring requirements to relevant SHL assessments while maintaining transparency and operational reliability. This approach aligns with real-world enterprise hiring needs and demonstrates both **applied machine learning skills** and **strong software engineering practices**.

SHL GenAI Assessment Recommendation System | Vardhan Reddy | December 2025