SHL Assessment Recommendation System

Approach Summary

Objective

The goal of this project is to develop an intelligent recommendation system that can suggest the most relevant SHL assessments for any job description or natural language query. The solution leverages natural language understanding, semantic search, and keyword-aware re-ranking to ensure both precision and balance across behavioral and technical assessments.

High-Level Architecture

1. Catalog Ingestion: Web scraping of SHL's product catalog (excluding pre-packaged job solutions). The scraper captures name, URL, and product description.
2. Data Augmentation: Augment the scraped data with provided labeled training and test datasets to ensure coverage.
3. Embedding Representation: SentenceTransformer (MiniLM-L6-v2) encodes product text into embeddings.
4. Indexing: FAISS vector index built for fast similarity retrieval.
5. Query Handling: Input query processed via the same embedding model; relevant tokens (e.g., python, sql, leadership) extracted.
6. Retrieval: FAISS returns the top semantically similar items.
7. Re-ranking: Final ranking combines semantic similarity and keyword overlap signals for balanced recommendations.

Data Pipeline

The pipeline includes scraping, cleaning, encoding, indexing, and recommendation generation. The system automatically loads SHL product details, stores them locally (df_products.pkl and embeddings.npy), and exposes them via an API for real-time recommendation.

Evaluation Methodology

Performance was evaluated using Mean Recall@10 over the labeled training set. This measures how many correct recommendations were retrieved in the top-10 for each query. A combination of semantic similarity and keyword weighting significantly improved recall scores compared to a purely embedding-based baseline.

System Components

- Flask API exposing /health and /recommend endpoints.
- Streamlit frontend allowing recruiters to paste job descriptions or queries.
- Dual output: Human-readable (UI CSV) and SHL submission-ready CSV.

Technologies Used

- Python (Flask, Streamlit, FAISS, SentenceTransformers, Pandas, BeautifulSoup)
- Model: all-MiniLM-L6-v2 (384-dimensional embeddings)
- Libraries: FAISS for similarity search, Flask for serving, Streamlit for UI.

Key Learnings and Optimizations

1. Token-aware scoring improved relevance for technical roles (Python, SQL, JavaScript).
2. Data augmentation from labeled dataset expanded coverage and improved balance.
3. Balanced recommendation strategy ensured inclusion of both knowledge-based and personality-based assessments.
4. Automated data ingestion pipeline reduced manual overhead for future catalog updates.

Deployment and Usage

The Flask API can be hosted locally or deployed to cloud platforms (Render, Railway, etc.). Streamlit serves as an interactive frontend. Both systems work seamlessly for real-time recommendations.

Artifacts Submitted

- API endpoint URL (Flask + ngrok or local)
- Streamlit web application URL
- GitHub repository with all code and experiments
- SHL_Assessment_Approach.pdf (this document)
- recommendations_final.csv (evaluation file)

Conclusion

This project demonstrates a robust, reproducible, and deployable AI-powered recommendation engine for SHL assessments, leveraging modern LLM embeddings and hybrid ranking for optimal recall and relevance.

Author: Moinuddin Navalur

Date: November 2025