SHL Assessment Recommendation System – Approach Document

Candidate: Aakanksha Sai

Date: 2025-05-06

1. Problem Overview

Hiring managers often struggle to find the right SHL assessments for their open roles due to inefficient keyword-based search. The goal of this project is to build an intelligent recommendation system that, given a natural language query or job description, returns the most relevant SHL assessments along with key attributes (name, URL, remote/adaptive support, duration, test type).

2. Solution Approach

A. Data Preparation

- Source: SHL product catalog

- Format: Compiled a CSV with columns: Assessment Name, URL, Remote Testing Support, Adaptive/IRT Support, Duration, Test Type

- Processing: Combined assessment name and test type into a search_text field for improved matching.

B. Recommendation Logic

- Vectorization: Used TfidfVectorizer (scikit-learn) to convert search_text into TF-IDF vectors.

- Similarity: Computed cosine similarity between the query and all assessment vectors.

- Ranking: Returned the top 10 most similar assessments.

C. Web Application

- UI: Built with Streamlit for rapid prototyping and user-friendly interaction.

- Features:
  - Input box for job description or query
  - Results displayed in a table with clickable assessment links and all required attributes

D. API

- Framework: FastAPI for RESTful endpoints.

- Endpoints:
  - /health – API health check
  - /recommend – Accepts a POST request with a query, returns up to 10 recommended assessments in the required JSON format

3. Tools & Libraries

- Python 3.10+
- pandas
- scikit-learn
- Streamlit
- FastAPI
- Uvicorn
- Docker (for Hugging Face Spaces deployment)

4. Evaluation

Metrics:
- Mean Recall@3
- MAP@3

Procedure:
- Used provided test queries and ground truth.
- For each query, compared top-3 recommendations to the ground truth.
- Calculated Recall@3 and MAP@3, then averaged over all queries.

Results:
- [Insert your computed Recall@3 and MAP@3 scores here, e.g., Recall@3: 0.80, MAP@3: 0.75]

5. Optimization & Future Work

Tried:
- Adding more fields (e.g., description) to search_text for richer matching.
- Tuning TF-IDF parameters (ngram range, stop words).

Future:
- Integrate semantic search (e.g., Sentence Transformers)
- Use LLMs for more context-aware recommendations
- Deploy on scalable cloud infrastructure

6. Submission Links

- Webapp: https://shl-assessment-recommendation-sys.streamlit.app/

- API: https://Aakanksha-Sai-shl-assessment-api.hf.space/recommend

- GitHub: https://github.com/Aakanksha-Sai/-SHL-Assessment-Recommendation-System-

Contact: aakankshasai04@gmail.com

This document summarizes the approach, tools, evaluation, and deployment for the SHL Assessment Recommendation System submission.