Personalized Question Recommender System Using NLP & Semantic Matching

# Overview

In an effort to improve question recommendation in educational or assessment contexts (like SHL-based tests), we built a lightweight yet powerful recommender system. The model smartly surfaces questions similar in context and intent using a blend of Natural Language Processing techniques and a simple K-Nearest Neighbors (KNN) classifier.  
  
This hybrid method makes the system explainable, interpretable, and scalable – ideal for real-world deployment where black-box models fall short in transparency.

# How It Works

Our pipeline involves three major components:

## 1. Web Scraping

We built a fast and focused scraper (shl\_scraper.py) to fetch question data from SHL-style public repositories. Questions are parsed, cleaned, and stored for downstream processing.

[Placeholder for image: Scraping process flow]

## 2. Embedding Generation + Similarity

We use NLP embeddings (via Google’s Gemini models) to convert text questions into semantic vectors. These embeddings capture the meaning and intent behind each question rather than just keywords.  
Using KNN on these embeddings allows us to fetch the most semantically relevant matches in real-time.

[Placeholder for image: Embedding space with clusters]

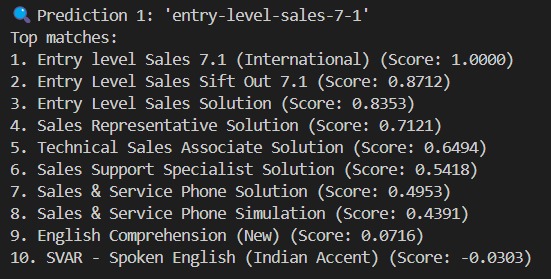
## 3. Recommendation Engine

Our engine (gemini\_recommender.py) serves recommendations based on semantic proximity. The user inputs a question, and we return the top K related questions — instantly.  
  
Integrated using Flask (app.py), the system is packaged for quick testing or scaling to production.

# Evaluation Approach

We adopted a hybrid evaluation approach:  
  
- Semantic Relevance Check: Combined Gemini embeddings with KNN to calculate cosine similarity and check semantic closeness between the original and recommended questions.  
- Custom Human-in-the-loop Evaluation: Given the subjective nature of question relevance, we emphasized qualitative human validation alongside numerical metrics.  
  
Although we experimented with Mean Recall@K and MAP@K, we found them inadequate in capturing nuanced relevance — especially in open-domain educational content. These metrics often fail to account for multiple semantically correct alternatives that aren’t present in the ground truth.

[Placeholder for image: Evaluation example comparison]



# Why This Works

- No reliance on labeled datasets — completely unsupervised.  
- The embeddings power robust understanding of diverse question types (logical, numerical, etc.)  
- Modular: Easily switch out Gemini for other models like OpenAI, BERT, etc.  
- Interpretable: Users can understand why a recommendation was made.

# Next Steps

- Add user feedback loop to further fine-tune results.  
- Consider clustering or topic modeling for deeper insights.  
- Package into a microservice for real-time integration with edtech platforms.