**Final Project: Implementing a Domain-Specific AI Assistant**

This final project presents the complete implementation of a domain-specific AI educational assistant originally conceptualized in the midterm project plan. The assistant is designed to support students in introductory mathematics by offering personalized guidance, adaptive feedback, and example-based problem assistance through a simulated few-shot learning approach. While the midterm focused on high-level design, this final implementation translates those ideas into a fully functional prototype developed in Google Colab using Python, TF-IDF similarity methods, and structured feature engineering.

The assistant operates within the domain of intelligent education and targets the need for accessible tutoring tools capable of adjusting to individual learners' needs. The dataset utilized in this implementation consists of synthetic algebra and geometry questions, each containing a problem statement, solution, topic label, and difficulty indicator. Although modest in size, the dataset enables a complete demonstration of preprocessing workflows, engineered feature construction, retrieval-based inference, interactive usage, and evaluation under realistic constraints. This controlled setup makes it possible to implement the full workflow within Colab's CPU-only environment while preserving conceptual alignment with our midterm proposal for a few-shot learning system.

Data preprocessing followed the structured pipeline outlined in the midterm plan. The raw text was transformed through lowercasing, regular expression filtering to remove non-alphanumeric characters, whitespace standardization, and stopword removal using NLTK. Exploratory data analysis verified that the dataset was free of missing values and included balanced topic and difficulty labels. Visualization of token lengths before and after preprocessing showed consistent reductions in noise and improved textual structure.

Although the dataset is synthetic and relatively clean, these preprocessing steps are essential to ensure reliable similarity-based retrieval by minimizing irrelevant textual variation that could distort semantic matching.

Feature engineering was a critical step in enabling the assistant to interpret user questions. Several pedagogically meaningful features were constructed, including question length, topic indicators, difficulty encoding, and TF-IDF semantic vectors. These features reflect aspects of mathematical question structure such as problem type and cognitive complexity. To examine the importance of the manually engineered features, a small RandomForestClassifier was trained to predict whether a question belonged to the algebra category. Despite the limited sample size, the analysis indicated that both question length and difficulty encoding were informative for the classification task. This reinforces the pedagogical relevance of lightweight engineered features in environments with limited data.

The core functionality of the assistant was implemented using a retrieval-based simulation of few-shot reasoning. When a user submits a question, the system preprocesses the input, computes its TF-IDF vector representation, and calculates cosine similarity between the query and all stored examples. It then selects the most semantically similar question and returns the corresponding answer along with topic and difficulty information. A general explanation template provides additional guidance by framing the retrieved example as a model for solving the user's query. This methodology approximates few-shot reasoning by identifying the best instructional analog from the available data. A simple text-based interface was implemented to allow users to interact directly with the system. Practical tests demonstrate that the assistant effectively retrieves relevant instructional examples in response to algebraic and geometric queries, showing that even a small dataset can support meaningful retrieval performance when coupled with appropriate preprocessing and feature engineering.

Evaluation adhered to the criteria established in the midterm's evaluation framework. We prepared a set of unseen test queries covering core mathematical topics and computed cosine similarity scores between these queries and their retrieved matches. The resulting similarity scores—0.91, 0.88, and 0.72—produced an average of 0.84, which aligns well with the midterm expectation of approximately 0.85 for adequate retrieval quality. These outcomes demonstrate that the implemented preprocessing pipeline and TF-IDF retrieval mechanism are effective at identifying semantically relevant examples even with a relatively small dataset.

During development, several challenges emerged. As the sole member of the project team, I assumed all responsibilities that would normally be distributed among multiple contributors. In a typical group setting, work might be divided into categories such as data preprocessing and exploratory analysis, feature engineering and model development, system implementation and testing, and evaluation with final report writing. Because no group members were available, I independently completed each of these components, including coding the entire notebook, performing all analysis, writing all documentation, and integrating results into a cohesive final report. This required careful planning and time management but also provided a valuable opportunity to engage deeply with each part of the data science workflow.

Several adjustments were made from the original midterm plan to reflect the practical limitations of the implementation environment. The midterm proposal initially envisioned integrating a true few-shot prompting method using large language models. However, to remain within Google Colab's CPU constraints and avoid external API usage costs, I replaced the true few-shot mechanism with a retrieval-based approach. This adaptation retained the spirit of example-driven learning while ensuring the system remained lightweight and

executable within the provided constraints. The dataset was also scaled down to a synthetic sample that preserved diversity of mathematical question types while supporting efficient experimentation. These adjustments provided a tractable and fully implemented prototype consistent with the assignment's emphasis on data science techniques over complex model architectures.

Reflecting on the implementation, several key insights emerged. High-quality preprocessing significantly improves the consistency of similarity-based retrieval systems. Even in small datasets, well-designed engineered features can meaningfully enhance interpretability and prototype performance. Retrieval-based assistants can serve as practical stand-ins for few-shot reasoning when computational resources are minimal. At the same time, the limitations of the system highlight avenues for future improvement. Expanding the dataset to include hundreds or thousands of questions would increase coverage and retrieval accuracy. Replacing TF-IDF with modern embedding models such as Sentence-BERT would improve semantic matching, particularly for questions phrased in unfamiliar ways. Integrating true few-shot prompting via transformer-based language models would enable adaptive, step-by-step explanations tailored to each student's reasoning. A more advanced system could also incorporate performance tracking and difficulty adaptation to model learner progression over time.

**References**

Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). *BERT: Pre-training of deep bidirectional transformers for language understanding.* Proceedings of NAACL-HLT. https://doi.org/10.48550/arXiv.1810.04805

Hugging Face. (n.d.). *Transformers documentation.* https://huggingface.co/docs/transformers

Kaggle. (n.d.). *MathQA dataset.* https://www.kaggle.com/datasets

OpenAI. (n.d.). *OpenAI API documentation.* https://platform.openai.com/docs

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). *Attention is all you need.* Advances in Neural Information Processing Systems. https://doi.org/10.48550/arXiv.1706.03762