Project Report: Milestone-3 (Tool Development)

Overview

This project implements a conversational agent (chatbot) for answering data science questions using a Retrieval-Augmented Generation (RAG) approach. The system combines semantic search (FAISS), a LightGBM classification model, and a large language model (Google Gemini) to deliver context-aware, high-quality answers and question quality feedback.

Scripts Folder Structure

- **chatbot.py**: Main Streamlit app and chatbot logic.
- **ingest_data.py**: Data ingestion and FAISS index creation.
- **requirements.txt**: Python dependencies.
- **README.md**: Documentation and setup instructions.
- **data/**: Contains the cleaned dataset and FAISS index.
- **model/**: Contains the trained LightGBM model (`lgbm_model.pkl`).

Key Components and Workflow

1. Data Preparation

- Cleaned Stack Exchange data is stored as CSV in `data` Folder.
- The expected columns include `Title`, `Body`, `Tags`, `Score`, `ViewCount`, etc.

2. Data Ingestion & Indexing

- `ingest_data.py` processes the CSV and builds a FAISS index using HuggingFace embeddings (`all-mpnet-base-v2`).
- The index enables efficient semantic retrieval of relevant Q&A posts.

3. Model Training and Usage

- **Model Used:** LightGBM (`lgbm_model.pkl`)
- **Purpose:** Classifies questions into categories (e.g., Data Science, Machine Learning, Statistics) and ranks them.
- **Training:** As detailed in `Milestone-2.ipynb`, the model was trained using feature-engineered and balanced data (SMOTE for class imbalance).
- **Evaluation:** Multiple models (Logistic Regression, Random Forest, SVM, XGBoost, LightGBM) were compared using GridSearchCV and F1 score. LightGBM was selected for its superior performance.
- **Explainability:** SHAP was used to interpret feature importance and model predictions.

4. **Chatbot Application**
   - `chatbot.py` integrates:
     - FAISS-based retrieval for context.
     - LightGBM model for question classification and ranking.
     - Google Gemini LLM for generating conversational answers and question quality assessment.
   - **Features:**
     - Predicts question quality and provides improvement suggestions.
     - Classifies question category.
     - Retrieves and displays relevant Stack Exchange posts.
     - Presents results in an interactive Streamlit UI.

How the Model is Used

- **Classification:** When a user submits a question, the LightGBM model predicts its category and quality.
- **Ranking:** The model helps rank retrieved posts for relevance.
- **Feedback:** The model's output is used to provide actionable suggestions to improve question quality.
- **Explainability:** SHAP plots and feature importances are used to explain model decisions, both during development and (optionally) in the chatbot UI.

Setup and Usage

1. **Install dependencies:**

   ```
   pip install -r requirements.txt
   ```

2. **Prepare data:** Place cleaned CSV in `data`.

3. **Build FAISS index:**

   ```
   python ingest_data.py
   ```

4. **Ensure model is present:** Place `lgbm_model.pkl` in `model/`.

5. **Configure API key:** Add your Google API key to `.streamlit/secrets.toml`.

6. **Run chatbot:**

   ```
   streamlit run chatbot.py
   ```

Customization

- **Dataset:** Replace CSV and rebuild index for new data.
- **Model:** Retrain or swap out LightGBM for other tasks.
- **UI:** Modify `chatbot.py` for new features.

References

- `README.md`
- `chatbot.py`
- `ingest_data.py`
- `Milestone-2.ipynb`