AI-Powered Doctor's Assistant for Risk Analysis

1. Introduction

This project implements a prototype AI-powered physician assistant that analyzes patient data, predicts short-term risk, and generates evidence backed explanations using medical literature. It is a local prototype designed to run on a machine learning Python environment without requiring external databases.

The system combines:
- A classical ML risk model for numerical prediction.
- A vector similarity search over ingested literature.
- A language model to generate natural language explanations.
- FastAPI endpoints to expose functionalities as services.

2. Technology Stack

Component	Tool/Library	Purpose
API Framework	FastAPI	REST API endpoints for prediction, ingestion, feedback
Embedding Model	sentence-transformers (all-MiniLM-L6-v2)	Convert literature text into dense embeddings
Vector Database	FAISS (Facebook AI Similarity Search)	Store and retrieve relevant documents using vector similarity
Language Model (LLM)	Hugging Face google/flan-t5-small	Generate risk explanations referencing evidence
Classical ML Model	Logistic Regression (scikit-learn)	Predict risk score from patient features
Data Storage	JSONL / JSON files	Store ingested literature, feedback, and metadata locally
Serialization	joblib	Save and load trained ML models
Environment	Python ML environment	Run locally without external DB/cloud

3. Data Management

3.1 Literature Database

- Stored locally in data/literature.jsonl (raw documents).
- Metadata maintained in data/metadata.json.
- Vector embeddings stored in data/faiss_index.bin.
- Vector similarity search is powered by FAISS.

3.2 Risk Model Database

- Synthetic patient dataset generated (train_risk_model.py).
- Model stored in models/risk_model.joblib.
- Contains:
  - Scaler (for feature normalization).
  - Logistic Regression classifier.

3.3 Feedback Database

- Feedback stored in append-only file data/feedback.jsonl.
- Each record: {prediction_id, accepted, notes}.

4. Risk Score Prediction Methodology

The risk model is a logistic regression classifier trained on synthetic patient data.

4.1 Input Features

- From patient record (PredictPayload):
- Age
- Systolic Blood Pressure (BP)
- Heart Rate (HR)
- Creatinine (renal marker)
- Hemoglobin
- Diabetes (derived from patient notes)

4.2 Feature Preprocessing

- Features are standardized using StandardScaler.
- Notes are processed to check for keywords ("diabetes", "type 2") → binary feature.

4.3 Risk Probability Formula

The logistic regression estimates probability:

$$P(y = 1 \mid x) = \frac{1}{1 + e^{-z}}$$

where

$$z = \beta_0 + \sum_{i=1}^{n} \beta_i x_i$$

Where:

- (x_i) = input features (age, bp, hr, creatinine, hemoglobin, diabetes).
- (\beta_i) = learned coefficients.
- Output = risk probability between 0 and 1.

4.4 Risk Categorization

- High Risk: Probability ≥ 0.7
- Medium Risk: 0.4 ≤ Probability < 0.7
- Low Risk: Probability < 0.4

5. Explanation & Evidence Generation

5.1 Evidence Retrieval (FAISS)

- Patient notes → embedded vector.
- Query executed against FAISS index.
- Returns top-k most relevant literature snippets.

5.2 Explanation Generation (LLM)

- Patient summary + evidence → fed into Flan-T5.
- Output includes:
  - Risk explanation in natural language.
  - References to supporting literature.
  - Suggested risk category.

6. Methodology Flow (End-to-End)

- Ingest Literature (/ingest/literature) → stored in FAISS.
- Submit Patient Data (/predict-risk) → extract features.
- Logistic regression predicts numerical probability.
- Patient notes embedded → FAISS retrieves evidence.
- Patient summary + retrieved docs → Flan-T5 → explanation.
- Output returned (score + category + explanation + evidence).
- Feedback may be submitted (/feedback) for refinement.

7. System Mindmap

```
Patient Data ───► Feature Extraction ───► Logistic Regression ───► Risk Probability
     │
     ▼
 Patient Notes ───► Embedding ───► FAISS Vector DB ───► Relevant Literature
     │
     ▼
Patient Summary + Literature ───► LLM ───► Explanation + Risk Category
```

8. Example API Output

```
{
  "prediction_id": "abcd-1234",
  "risk_score": 0.68,
  "risk_category": "Medium",
  "model_suggested_category": "High",
  "explanation": "The patient's elevated blood pressure and diabetes history increase cardiovascular risk [doc_12]. Creatinine indicates mild renal dysfunction, further contributing to risk. RISK_CATEGORY: High",
  "evidence": [
    {
      "id": "doc_12",
      "title": "Hypertension and Diabetes Risk",
      "source": "PubMed",
      "snippet": "Patients with type 2 diabetes and high blood pressure are at increased risk..."
    }
  ],
  "timestamp": 1696400000
}
```

9. Conclusion

This prototype demonstrates how AI can augment medical decision-making by:

- Using logistic regression for interpretable risk scoring.
- Leveraging vector search for literature-based evidence retrieval.
- Employing a language model for natural-language risk explanations.
- Providing a modular FastAPI service that integrates all steps.