AI Augmented Programming

Umang Chaudhry, Senior Data Scientist

Vanderbilt Data Science Institute

Introduction to AI-Augmented Programming

- What is AI-Augmented Programming?
  + AI-assisted tools that enhance coding workflows
  + Benefits: Speed, efficiency, and error reduction
- Why it Matters for Data Scientists
  + Automates repetitive tasks (data wrangling, exploratory analysis)
  + Debugs and optimizes code faster
  + Faster iteration cycles for experiments
  + Greater focus on solving problems, not syntax

Setup

- Cursor: https://www.cursor.com
- OpenAI API Keys: https://platform.openai.com/docs/overview
- Anthropic API Keys: http://console.anthropic.com/

AI for Data Manipulation

- Pain Points in Data Manipulation
  - + Inconsistent formats
  - + Missing values
  - + Generating quick insights
- How AI Helps
  - + Auto-generates scripts
  - + Suggest EDA plots
  - + Simplifies data cleaning workflows

- Example: iris Dataset but its dirty

https://www.kaggle.com/datasets/bharathkumarkathula/iris-with-missing-data/code

Debugging with AI

- How can AI assist?
  - Identifies syntax errors and logical bugs
  - Refactors inefficient code

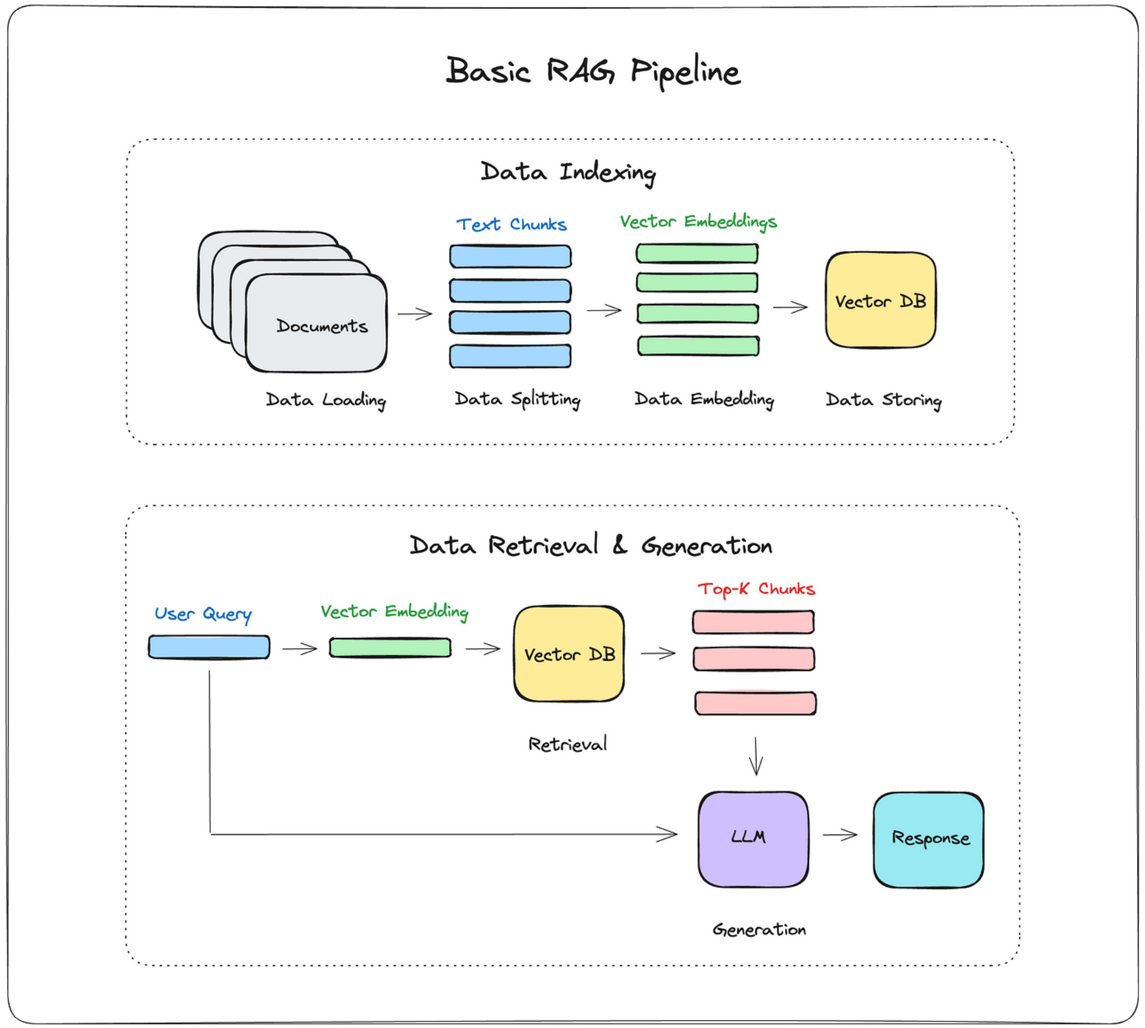
- Example: A buggy function that returns Fibonacci numbers
- Expected results:
  - 0
  - 1
  - 1
  - 2
  - 3
  - 5

Interfacing with LLMs: RAG Chatbot with LangChain

- What is RAG?
  - + Retrieval Augmented Generation combines retrieval of relevant information (from structured or unstructured data) with generative AI for robust and context-aware chatbots
- What is LangChain?
  - + A Python framework for building applications powered by LLMs, with integrated tools for retrieval, embeddings, and pipelines
- Necessary Libraries
  - + pip install langchain openai faiss-cpu pypdf tiktoken

Basic RAG Steps:

1. Load
2. Split
3. Embed
4. Retrieve
5. Query



Limitations and Best Practices

- Limitations:
  - Output dependent on prompt quality for complex tasks
  - Ethical concerns – working with PHI, HIPPA, FERPA or any other form of protected data stored in your codebase?

Don't use AI!!!

- Best practices:
  - Always review AI generated code – it might not do exactly what you think it does, especially when working with APIs (avoid unexpected costs due to unexpected recursion)
  - AI does not eliminate the need to understand your code. Being unfamiliar with your working code will make debugging down the road a nightmare
  - Use AI as a tool, not a decision maker