

Learning roadmap for AI/ML



- Languages & Frameworks: Python, LangChain, HuggingFace
- AI/ML Technologies: RAG, Prompt Engineering, Generative AI
- Cloud & Tools: Azure AI Services, Azure OpenAI, Databricks, Git, DBeaver
- Data Processing: PDF/Word/Excel parsing, ETL pipelines

with in these which one should i start learning with give me a roadmap to complete all the things

Got it 👍 Before I sketch out a roadmap, let me quickly check in:

Are you aiming to learn these for **career growth (like AI/ML engineer, data engineer, cloud developer)**, or for **projects/personal learning**?

That will help me fine-tune how deep we go into each.

For now, I'll assume you're looking at **career-ready learning in AI/ML with practical projects**. Here's a step-by-step roadmap to cover everything you listed:

Roadmap (Beginner → Advanced)

Step 1: Core Programming & Data Handling

- **Python** → Foundation. Master syntax, OOP, data structures, libraries (NumPy, Pandas).
- **Data Processing** → Learn how to parse **PDF/Word/Excel** (e.g., PyPDF, docx, openpyxl) and build small ETL pipelines.
 - 🖱️ *Mini Project*: Extract text from PDFs, clean it, and save to a database.

Step 2: AI/ML Fundamentals

- **ML Basics** → Scikit-learn (classification, regression, clustering).
- **Generative AI Concepts** → Understand LLMs, embeddings, tokenization.
- **Prompt Engineering** → Practice crafting prompts for ChatGPT & open-source LLMs.

Step 3: Frameworks & Libraries

- **LangChain** → Learn chaining, memory, tools, agents.
- **HuggingFace** → Explore pre-trained models, transformers, and fine-tuning.
- **RAG (Retrieval Augmented Generation)** → Combine embeddings + vector DBs (e.g., FAISS, Chroma).

- 🖱️ *Mini Project:* Build a Q&A chatbot over your PDF collection.

Step 4: Cloud & Tools

- **Azure AI Services & Azure OpenAI** → Learn deploying LLMs, integrating cognitive search.
- **Databricks** → Learn scalable data processing, ML pipelines.
- **Git & DBeaver** → Version control & database management.
 - 🖱️ *Mini Project:* Deploy your RAG chatbot on Azure, connect it to a database, and manage with Git.

Step 5: Integration & Advanced Projects

- End-to-end **AI application**:
 - ETL pipeline to collect data
 - Use LangChain + HuggingFace for RAG
 - Deploy on Azure with Databricks pipelines
 - Manage codebase with Git

Suggested Timeline

- **Month 1–2:** Python + Data Processing
- **Month 3:** ML basics + Prompt Engineering
- **Month 4–5:** LangChain + HuggingFace + RAG
- **Month 6–7:** Azure + Databricks + Deployment
- **Month 8+:** Build full-stack AI projects

💡 Question for you:

Do you prefer a **project-based roadmap** (learn by building small apps each step) or a **theory-first roadmap** (cover concepts deeply, then projects)?



I'm planning to learn these for my career growth in AI Engineer - Generative AI Solutions

Perfect — that makes it clear!

If your goal is **AI Engineer (Generative AI Solutions)**, then the roadmap should be **project-driven**, because employers care about seeing end-to-end solutions you can build, not just theory.

Here's how I'd structure it for you:

Career Roadmap for AI Engineer – Generative AI Solutions

Phase 1: Strong Foundation (Month 1-2)

- **Python (must be solid!)**
 - Focus on: OOP, NumPy, Pandas, API requests, file handling.
 - Build mini-projects: data wrangling scripts, PDF/Excel parsers.
- **Git & DBeaver**
 - Get comfortable with version control & basic database queries.

👉 *Mini project:* Build a pipeline that extracts text from PDF → cleans → stores in a DB.

Phase 2: AI/ML Core + Prompt Engineering (Month 3)

- **ML basics:** Scikit-learn → regression, classification, clustering.
- **Prompt Engineering:** Learn few-shot, chain-of-thought, role prompting.
- **Generative AI basics:** Understand embeddings, tokenization, fine-tuning.

👉 *Mini project:* Prompt-based chatbot using OpenAI API.

Phase 3: RAG & Frameworks (Month 4-5)

- **LangChain:** Learn chains, agents, memory.
- **HuggingFace:** Use LLMs (GPT, LLaMA, Falcon) & embeddings.
- **RAG:** Combine vector DBs (FAISS/Chroma) + LLMs for contextual answers.

👉 *Mini project:* "Ask my documents" RAG chatbot using LangChain + HuggingFace.

Phase 4: Cloud & Scalability (Month 6-7)

- **Azure AI Services:** Cognitive search, Azure OpenAI, Vector search.
- **Databricks:** Data pipelines, model training at scale.
- **Deployments:** Containerization (Docker) + CI/CD.

👉 *Mini project:* Deploy your RAG chatbot on Azure, scale using Databricks pipelines.

Phase 5: Career-Ready Capstones (Month 8+)

- End-to-end **Generative AI solution** projects:
 1. **Enterprise Chatbot** → PDF/Excel/DB knowledge base.
 2. **Generative AI API** → Offer text summarization / Q&A as a service.
 3. **Hybrid AI app** → Combine structured data (SQL/Excel) with LLM for insights.

👉 Showcase on GitHub + write a blog/LinkedIn post → recruiters love visible projects.

Key Learning Strategy

1. **Always tie learning to projects** → theory alone won't stick.
2. **Publish your work** → GitHub + LinkedIn portfolio.
3. **Focus on RAG + Azure deployment** → hot skills right now for AI Engineers.

Now, to make this roadmap concrete:

Do you want me to **design your first 2 projects (step-by-step tasks + resources)** so you can start building while learning Python & data processing?



I want to complete it in like about 2 months period. and i'm well versed in java so i'm sure i can manage python as well and i know how to use git

That's great — since you already know Java and Git, you can **compress the timeline**. Instead of 8 months, we can make a **focused 2-month (8-week) roadmap** with projects baked in.

Here's how you can do it:

2-Month Roadmap for Generative AI Engineer

Week 1: Python for AI + Data Processing

- Brush up only the Python bits you don't know → list/dict comprehensions, decorators, async, NumPy, Pandas.
- Learn to parse PDFs/Word/Excel.
- *Mini Project*: Extract text from PDFs → clean → load into SQLite/DB (using DBeaver for queries).

Week 2: Prompt Engineering + Generative AI Basics

- Study prompting techniques: zero-shot, few-shot, chain-of-thought, role prompting.
- Explore OpenAI API and HuggingFace Transformers at a basic level.
- *Mini Project*: Build a small chatbot using OpenAI API with prompt templates.

Week 3-4: RAG + LangChain + HuggingFace

- Deep dive into LangChain: chains, tools, memory, agents.
- HuggingFace: embeddings, using models locally (LLaMA, Falcon, Mistral).
- RAG: combine embeddings + vector DB (FAISS/Chroma).
- *Mini Project*: Q&A chatbot over PDFs (retrieves answers from docs).

Week 5: Azure AI & Databricks

- Learn Azure OpenAI, Cognitive Search, and vector search.
- Get familiar with Databricks → simple ETL pipelines, ML model training.
- *Mini Project*: Deploy your RAG chatbot on Azure (serve via REST API).

Week 6: Integration + Deployment

- Dockerize your chatbot.
- Learn CI/CD basics (GitHub Actions or Azure DevOps).
- *Mini Project*: End-to-end deployment of your chatbot with a web UI (Streamlit/Flask + Azure).

Week 7-8: Capstone Projects (Portfolio-Ready)

Choose **2 projects** to polish and showcase:

1. **Enterprise Document Assistant** → upload docs (PDF/Excel/Word), ask questions, get answers.
2. **Generative Insights API** → an API that summarizes, extracts insights, and answers queries from business data.

👉 Push to GitHub + write short LinkedIn posts about what you learned → this is *gold* for recruiters.

Study Rhythm (per week)

- **15-20 hrs total** → 2 hrs on weekdays, 5 hrs on weekends.
- Balance: 40% coding projects, 40% reading/docs, 20% videos/tutorials.

⚠️ Key advantage: Since you know Java + Git, you can skip a lot of basics and dive directly into **LangChain, HuggingFace, and Azure deployment** — those are the skills employers want to see.

Do you want me to **draft a detailed weekly learning plan (resources + project outline for each week)** so you know exactly what to study and build each step of the way?



yeah give me a weekly planning and make it into a pdf for me to download both roadmap and weekly planning

Here's your **2-Month Generative AI Engineer Roadmap & Weekly Plan** in PDF format:

 [Download PDF](#)

It includes both the roadmap overview and a detailed weekly breakdown with mini-projects.

Do you want me to also add a **list of resources (docs, tutorials, GitHub repos)** for each week inside the PDF, so you don't have to hunt them down separately?