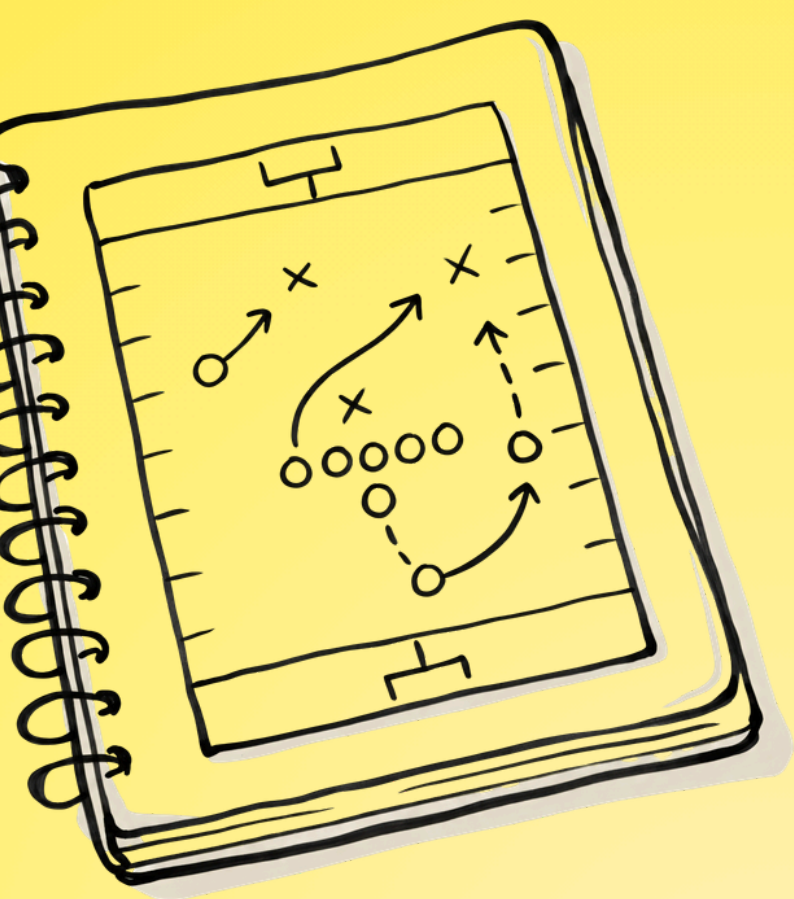
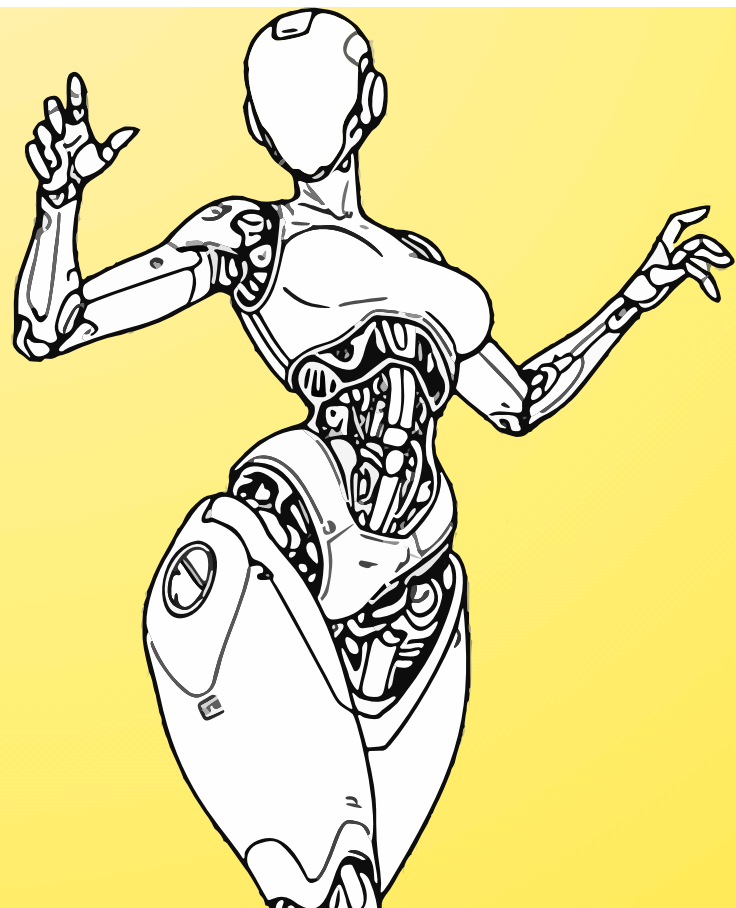




The AI Engineering Handbook

A curated collection of books, courses, projects, and resources that helped me learn AI Engineering.

By Riwano Fariz



Why I Created This Handbook ?

When I started learning AI Engineering, I found hundreds of roadmaps, tutorials, and courses—but no single place that connected everything together.

So I started organizing the resources that helped me understand AI Engineering while building projects. This handbook is the result.

It's not meant to be the only way to learn AI Engineering. It's a curated guide to the books, courses, documentation, projects, and communities that helped me build a strong foundation.

My hope is that it saves you time, helps you avoid information overload, and gives you a clearer path than I had when I started.

Since AI evolves quickly, this handbook will continue to be updated over time.



Before You Start

AI Engineering is one of the fastest-moving fields in technology.

This handbook isn't meant to be the only way to learn AI Engineering, nor is it a shortcut. Instead, it's a curated collection of the books, courses, documentation, projects, and resources that helped me understand AI Engineering while building projects.

I'm continuously learning too, so think of this as a living handbook rather than a finished one. Whenever better resources or new technologies emerge, I'll update future versions.

Wherever possible, I've prioritised official documentation and primary sources so you can learn directly from the people and organisations building these technologies.

My goal isn't to teach you everything. It's to help you spend less time wondering what to learn next and more time actually building.



**This is a living handbook and
will be updated as I discover
better resources.**

How to Use This Handbook

This handbook is designed to be followed from top to bottom, but each chapter can also be read independently if you're looking for a specific topic.

Every chapter includes:

- 📖 **What you'll learn**
- 🎯 **Why it matters**
- 📚 **Recommended books, courses, blogs, videos, or official documentation**
- ✅ **A learning checklist**
- ➡️ **What to learn next**

This handbook isn't about collecting resources—it's about learning in the right order.

My advice is simple:

Don't try to learn everything at once.

Focus on understanding one topic before moving to the next. AI Engineering is built layer by layer, and every chapter in this handbook is designed to build on the previous one.

Take your time, stay consistent, and most importantly, keep building as you learn.



Table of Contents

Part 1 — Getting Started

1. What is an AI Engineer?
2. The AI Engineering Roadmap

Part 2 — Foundations

3. Mathematics
4. Python
5. SQL
6. Data Analysis

Part 3 — Machine Learning

7. Machine Learning
8. Neural Networks

Part 4 — AI Engineering

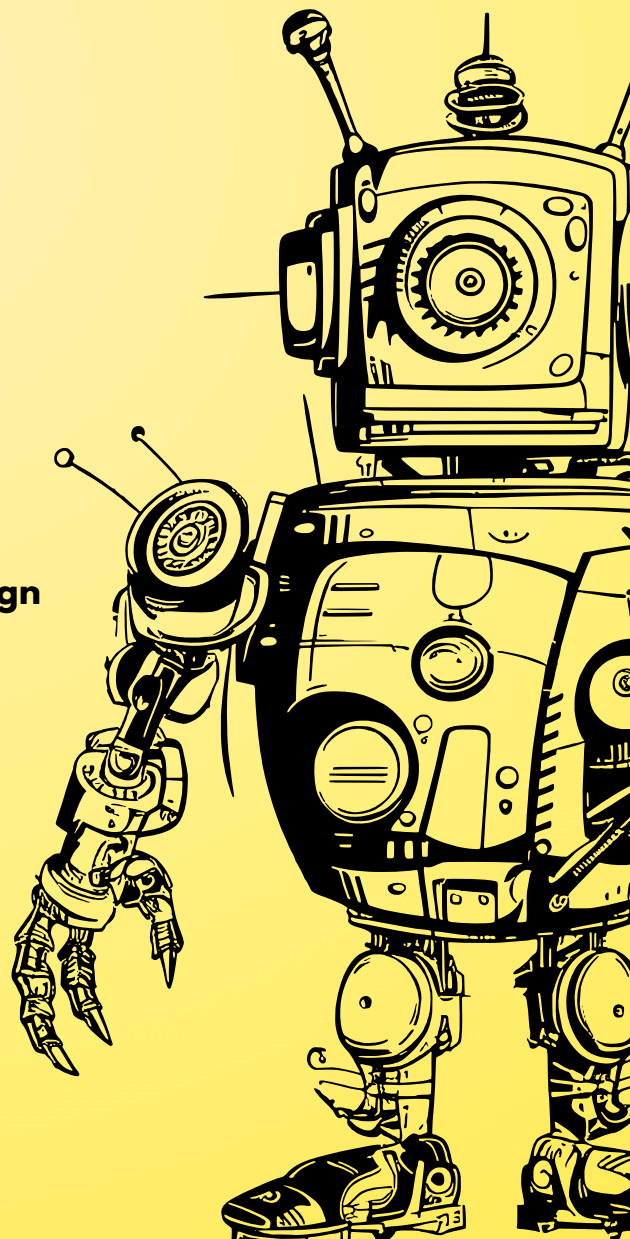
9. Large Language Models
10. RAG
11. AI Agents

Part 5 — Engineering

12. Software Engineering & System Design

Appendix

- AI Glossary
- Final Advice



Chapter 1

What is an AI Engineer?

Brief Overview

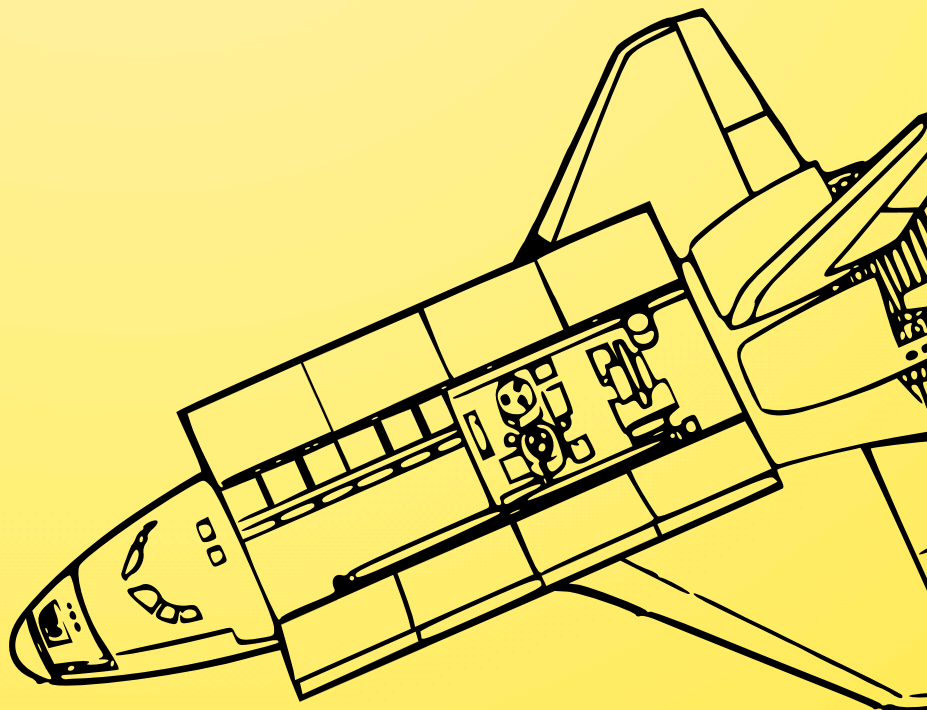
An AI Engineer builds software powered by Artificial Intelligence. Instead of creating AI models from scratch, they use existing models like Large Language Models (LLMs) to build applications such as chatbots, AI assistants, search systems, and workflow automations. Think of an AI Engineer as the bridge between AI models and real-world products.

What They Actually Do

An AI Engineer typically:

- **Integrates AI into applications**
- **Builds AI assistants and chatbots**
- **Implements RAG and AI agents**
- **Connects AI with databases and APIs**
- **Deploys and monitors AI systems**
- **Improves reliability, speed, and user experience**

Their goal is to build useful AI-powered products—not train foundation models.



Chapter 1

What is an AI Engineer?

AI Engineer	Machine Learning Engineer
Builds AI applications	Builds and optimizes ML models
Uses LLMs, RAG, and AI agents	Trains and fine-tunes models
Focuses on product integration	Focuses on model performance

AI Engineer	Data Scientist
Builds AI products	Analyzes data and finds insights
Works on deployment	Works on analysis and experiments
Focuses on production systems	Focuses on business decisions

AI Engineer	Software Engineer
Builds AI-powered software	Builds software systems
Understands LLMs and AI workflows	Understands software architecture
Combines AI with engineering	Focuses on application development

Chapter 1

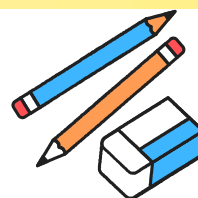
What is an AI Engineer?

Why This Role Exists

Modern AI models are already incredibly powerful. Most companies don't need to build new models—they need engineers who can integrate them into real products.

That's where AI Engineers come in. They combine AI knowledge with software engineering to build reliable, production-ready applications.

Recommended Resources



Books

- **AI Engineering — Chip Huyen**

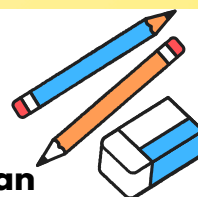
A practical introduction to building production-ready AI systems. Great for understanding modern AI Engineering beyond just machine learning.

YouTube

- **Andrej Karpathy**

One of the best resources for understanding how LLMs, neural networks, and modern AI actually work.

Learning Checklist



Before moving to the next chapter, make sure you can answer these questions:

- **What is an AI Engineer?**
- **How is an AI Engineer different from a Machine Learning Engineer?**
- **How is an AI Engineer different from a Data Scientist?**
- **Why has AI Engineering become an important role?**
- **Can you explain what an AI Engineer builds in simple terms?**

Key Takeaway

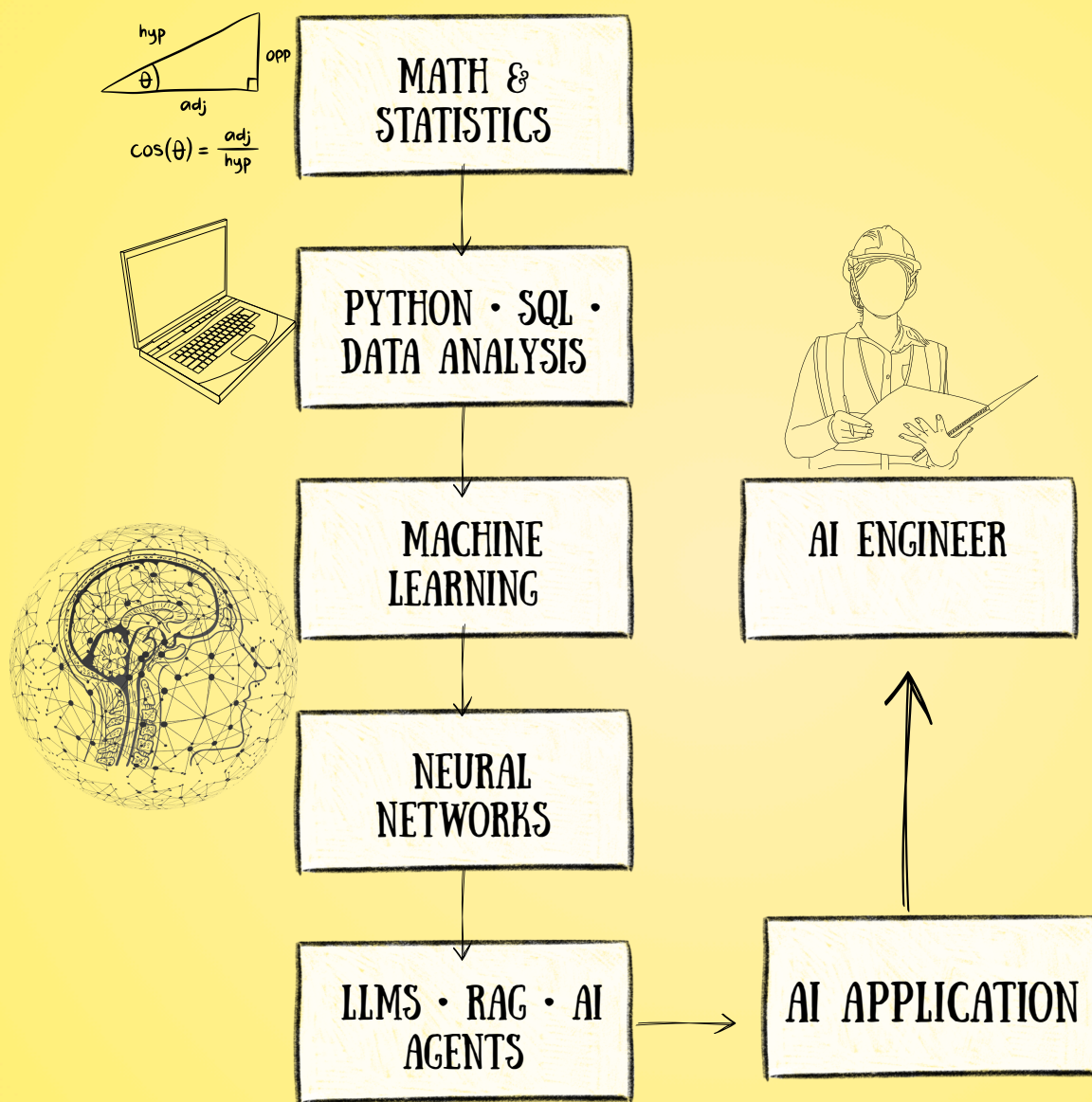
**AI Engineers don't build foundation models like GPT.
They build reliable, production-ready applications that
use those models to solve real-world problems.**

Chapter 2

The AI Engineer Roadmap

The Roadmap

AI Engineering isn't a single skill—it's a combination of multiple disciplines that build on top of each other.



Along with this:

Software Engineering + System Design

Chapter 2

The AI Engineer Roadmap

Why This Order?

Many people start by learning Python or watching tutorials on ChatGPT. Instead, start by understanding what an AI Engineer actually builds.

Once you know the destination, it's much easier to understand why each layer below it matters.

Each layer in this roadmap supports the one above it.

What Each Layer Represents:

- **Math & Statistics**

The foundation for understanding how machine learning models work.

- **Python, SQL & Data Analysis**

The core tools you'll use to work with data and build AI applications.

- **Machine Learning**

Learn how machines identify patterns, make predictions, and learn from data.

- **Neural Networks**

Understand the models that power modern AI systems, including Large Language Models.

- **LLMs, RAG & AI Agents**

Learn how modern AI applications are built using foundation models and external knowledge.

- **AI Applications**

Bring everything together to build chatbots, AI assistants, automation tools, and other real-world AI products.

- **Software Engineering & System Design**

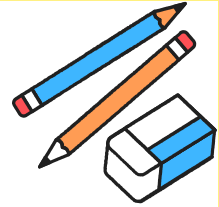
AI doesn't run on its own. Every AI application needs APIs, databases, authentication, deployment, scalability, and reliable system design. These skills support every layer of the roadmap.

Chapter 2

The AI Engineer Roadmap

Learning Checklist

- ☐ I understand the overall AI Engineering roadmap.
- ☐ I know why each layer exists.
- ☐ I understand why Software Engineering and System Design are important.
- ☐ I know the order I'll follow throughout this handbook.



Key Takeaway

Don't memorise the roadmap. Understand why each layer exists and how it supports the next one.

Who Is This Handbook For?

- **You're a student interested in AI.**
- **You're a software engineer moving into AI.**
- **You're a data professional exploring LLMs.**
- **You're building AI projects and want a structured learning path.**



Chapter 3

Mathematics

Brief Overview

You don't need a PhD in Mathematics to become an AI Engineer. However, understanding a few core mathematical concepts will help you understand how machine learning models, neural networks, and LLMs actually work.

The goal isn't to solve complex equations—it's to build intuition.

Why Learn Mathematics?

Mathematics helps you understand:

- **How machine learning models learn from data**
- **How vectors represent text and images**
- **How embeddings and vector databases work**
- **Why neural networks learn through optimisation**
- **Why models make predictions the way they do**

What to Learn

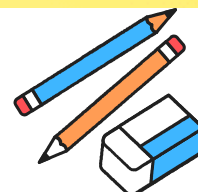
- **Statistics:** Helps you understand data, distributions, evaluation metrics, and model performance.
- **Probability:** Most machine learning models make predictions based on probability.
- **Linear Algebra:** The foundation of vectors, matrices, embeddings, and neural networks.
- **Vectors & Matrices:** Modern AI represents text, images, and other data as vectors. Understanding them is essential for embeddings and vector databases.
- **Calculus:** Helps you understand optimisation, gradient descent, and backpropagation.

Don't worry about mastering every formula—focus on the intuition.

Chapter 3

Mathematics

Recommended Resources



Books

- **Mathematics for Machine Learning**
Marc Peter Deisenroth, A. Aldo Faisal & Cheng Soon Ong
A beginner-friendly introduction to the mathematical concepts behind machine learning, including linear algebra, calculus, and probability.

- **Practical Statistics for Data Scientists**
Peter Bruce, Andrew Bruce & Peter Gedeck
Focuses on the statistics you'll actually use in data science and machine learning, with practical examples in Python.

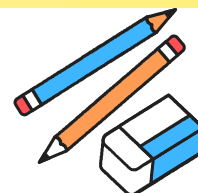
Courses or Learning Platforms

- **Khan Academy**
Excellent for strengthening fundamentals in algebra, probability, statistics, linear algebra, and calculus—especially if you're starting from scratch.

YouTube

- **3Blue1Brown**
One of the best channels for building mathematical intuition through visual explanations.
- **StatQuest with Josh Starmer**
Outstanding explanations of statistics, probability, machine learning algorithms, and evaluation metrics. Perfect for beginners.

Learning Checklist



- ☐ I understand why statistics is important.
- ☐ I understand the basics of probability.
- ☐ I know what vectors and matrices are.
- ☐ I understand the intuition behind linear algebra.
- ☐ I know why calculus is used in neural networks.

Key Takeaway

You don't need to become a mathematician—you need enough mathematical intuition to understand how AI models work.

Chapter 4

Python



Brief Overview

Python is the most widely used programming language in AI, Machine Learning, and Data Science. Its simple syntax and vast ecosystem make it the preferred language for building AI applications.

Most AI libraries, frameworks, and tools you'll use are built around Python.

Why Learn Python?

Python enables you to:

- **Build AI and Machine Learning applications**
- **Work with data efficiently**
- **Use libraries like NumPy, Pandas, PyTorch, and TensorFlow**
- **Build APIs using FastAPI**
- **Automate repetitive tasks**
- **Integrate AI models into real-world applications**

What to Learn

- **Python Basics**

Variables, data types, loops, functions, modules, and file handling.

- **Object-Oriented Programming (OOP)**

Classes, objects, inheritance, and encapsulation.

- **APIs**

Making API requests and working with JSON data.

- **Virtual Environments & Package Management**

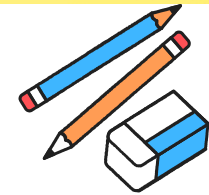
Using venv, pip, and managing project dependencies.

- **Async Programming (Optional)**

Understanding asynchronous programming with async and await for faster AI applications.

Chapter 4

Python



Recommended Resources

Books

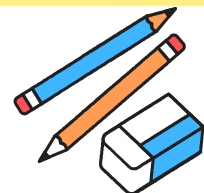
- **Python Crash Course — Eric Matthes**
A beginner-friendly introduction to Python with hands-on projects.
- **Fluent Python — Luciano Ramalho**
Best for intermediate developers who want to write clean, Pythonic code.

Courses

- **CS50's Introduction to Programming with Python (Harvard)**
A fantastic beginner course that teaches Python through practical exercises.

YouTube

- **Corey Schafer**
One of the best channels for learning Python fundamentals, OOP, virtual environments, and best practices.
- **freeCodeCamp.org**
Offers complete Python courses ranging from beginner to advanced.
- **Official Documentation**
Python Documentation
The official reference for Python syntax, standard libraries, and language features.



Learning Checklist

- ☐ I understand Python fundamentals.
- ☐ I can write functions and work with modules.
- ☐ I understand Object-Oriented Programming.
- ☐ I know how to create and manage virtual environments.
- ☐ I know how to work with APIs and JSON.
- ☐ I'm comfortable reading Python documentation.

Key Takeaway

Python isn't just another programming language—it's the foundation of the modern AI ecosystem. Master it once, and you'll use it throughout your AI Engineering journey.

Chapter 5

SQL



Brief Overview

SQL (Structured Query Language) is used to store, retrieve, and manipulate data from relational databases.

As an AI Engineer, you'll often work with user data, application data, logs, and metadata stored in databases. Knowing SQL allows you to efficiently access and manage that data.

Why Learn SQL?

SQL helps you:

- Query and filter data
- Join data from multiple tables
- Analyze datasets
- Prepare data for machine learning
- Power backend applications
- Work with databases like PostgreSQL and MySQL

What to Learn

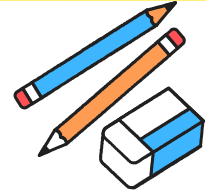
Topic	What to Learn	Why It Matters
SQL Basics	SELECT, WHERE, ORDER BY, LIMIT	Retrieve and filter data from databases.
Joins	INNER JOIN, LEFT JOIN, RIGHT JOIN	Combine data stored across multiple tables.
Aggregations	COUNT(), SUM(), AVG(), MIN(), MAX(), GROUP BY, HAVING	Summarize and analyze datasets efficiently.
Window Functions	ROW_NUMBER(), RANK(), DENSE_RANK(), PARTITION BY	Perform advanced analytics without modifying the original data.
PostgreSQL Basics	Tables, Indexes, Constraints, Views	Understand how relational databases are structured and optimized.

Chapter 5

SQL



Recommended Resources



Books

- **Learning SQL — Alan Beaulieu**

A beginner-friendly introduction to SQL that covers the fundamentals with practical examples.

YouTube

- **freeCodeCamp.org**

Comprehensive SQL tutorials for beginners.

- **Programming with Mosh**

Clear explanations of SQL fundamentals and database concepts.

Practice Platforms

- **SQLBolt**

Interactive lessons that teach SQL step by step.

- **LeetCode (Database Section)**

Practice SQL problems commonly asked in interviews.

Backend Platform

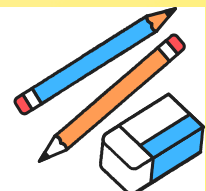
- **Supabase**

Learn how PostgreSQL is used in real applications through an open-source backend platform. Great for understanding databases, authentication, storage, APIs, and vector support in AI applications.

Official Documentation

- **PostgreSQL Documentation**
- **Supabase Documentation**

Learning Checklist



- ☐ I can write basic SQL queries.
- ☐ I understand different types of JOINS.
- ☐ I can aggregate and summarize data.
- ☐ I know how to use basic window functions.
- ☐ I understand the fundamentals of PostgreSQL.
- ☐ I know how to use Supabase to build and interact with a PostgreSQL database.
- ☐ I understand how databases fit into modern AI applications.

Key Takeaway

Data powers AI. SQL is the language that helps you access, understand, and work with that data.

Chapter 6

Data Analysis



Brief Overview

Before building Machine Learning models, you need to understand your data.

Data Analysis is the process of collecting, cleaning, exploring, and visualising data to uncover patterns and prepare it for machine learning.

Almost every AI or Machine Learning project begins with data analysis.

Why Learn Data Analysis?

Data Analysis helps you:

- Understand your dataset
- Find missing or incorrect values
- Discover patterns and trends
- Prepare data for Machine Learning
- Create meaningful visualisations
- Make better decisions using data



Note: Start by learning Pandas since it's the industry standard and supported by most tutorials. Once you're comfortable, learn Polars to work more efficiently with larger datasets and modern data pipelines.

What to Learn

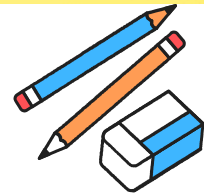
Topic	What to Learn	Why It Matters
NumPy	Arrays, Mathematical Operations, Broadcasting	The foundation for numerical computing in Python.
Pandas	DataFrames, Indexing, Filtering, GroupBy, Merge	The most widely used library for data cleaning, manipulation, and analysis.
Polars	DataFrames, Lazy Execution, Expressions	A modern, high-performance DataFrame library designed for speed and large datasets.
Data Visualization	Matplotlib, Seaborn, Plotly	Helps understand data through graphs and charts.
Exploratory Data Analysis (EDA)	Missing Values, Outliers, Correlation, Feature Distribution	Understand your data before building models.
Data Cleaning & Preprocessing	Null Values, Duplicates, Encoding, Scaling	Clean and prepare data for machine learning models.

Chapter 6

Data Analysis



Recommended Resources



Books

Python for Data Analysis — Wes McKinney

Written by the creator of Pandas, this remains one of the best books for learning data analysis in Python.

YouTube

Keith Galli

Excellent tutorials on Pandas, NumPy, and practical data analysis.

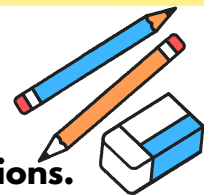
Alex The Analyst

Great for learning complete data analysis workflows, SQL, Python, and Power BI.

Official Documentation

- **NumPy Documentation**
- **Pandas Documentation**
- **Polars Documentation**
- **Matplotlib Documentation**
- **Plotly Documentation**

Learning Checklist



- ☐ I understand NumPy arrays and numerical operations.
- ☐ I can manipulate data using Pandas DataFrames.
- ☐ I understand when to use Polars and how it differs from Pandas.
- ☐ I can clean and preprocess datasets.
- ☐ I know how to perform Exploratory Data Analysis (EDA).
- ☐ I can visualize data using Matplotlib or Plotly.
- ☐ I understand how to prepare data before training a machine learning model.

Key Takeaway

Data is the foundation of every AI system. Learning how to clean, analyze, and understand it is just as important as building the model itself.

Chapter 7

Machine Learning



Brief Overview

Machine Learning is a branch of Artificial Intelligence that enables computers to learn patterns from data instead of being explicitly programmed.

Rather than writing rules for every scenario, we train models using data so they can make predictions, classifications, or decisions.

Machine Learning forms the foundation of modern AI, including neural networks and Large Language Models.

Why Learn Machine Learning?

Machine Learning helps you:

- **Understand how AI models learn from data**
- **Build predictive models**
- **Prepare for Deep Learning and LLMs**
- **Understand model evaluation and performance**
- **Develop intuition for how modern AI works**

What to Learn

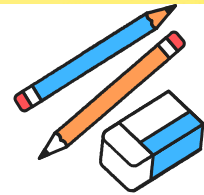
Topic	What to Learn	Why It Matters
Supervised Learning	Regression, Classification	Learn how models make predictions using labeled data.
Unsupervised Learning	Clustering, Dimensionality Reduction	Discover hidden patterns in unlabeled data.
Reinforcement Learning	Agents, Rewards, Environment	Understand how agents learn through trial and error.
Model Evaluation	Train/Test Split, Cross Validation, Accuracy, Precision, Recall, F1 Score	Measure how well a model performs.
Overfitting & Underfitting	Bias, Variance, Regularization	Learn why models fail to generalize.
🔧 Feature Engineering	Feature Selection, Scaling, Encoding	Improve model performance using better input data.

Chapter 7

Machine Learning



Recommended Resources



Books

Hands-On Machine Learning with Scikit-Learn, Keras & TensorFlow — Aurélien Géron

One of the best practical books for learning Machine Learning through Python.

YouTube

StatQuest with Josh Starmer

Excellent explanations of Machine Learning algorithms, evaluation metrics, and statistics.

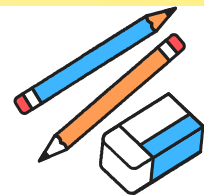
freeCodeCamp.org

Complete Machine Learning courses with hands-on coding examples.

Official Documentation

- **scikit-learn Documentation**
- **XGBoost Documentation**
- **LightGBM Documentation**

Learning Checklist



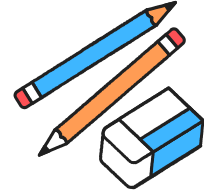
- ☐ **I understand the difference between Supervised, Unsupervised, and Reinforcement Learning.**
- ☐ **I know when to use Regression vs Classification.**
- ☐ **I understand model evaluation metrics.**
- ☐ **I know what overfitting and underfitting are.**
- ☐ **I understand the importance of feature engineering.**
- ☐ **I'm comfortable using scikit-learn to build basic Machine Learning models.**

Chapter 7

Machine Learning



Don't try to learn every algorithm at once. A practical order is:



- 1. Linear Regression**
- 2. Logistic Regression**
- 3. Decision Trees**
- 4. Random Forest**
- 5. Support Vector Machines (SVM)**
- 6. K-Nearest Neighbors (KNN)**
- 7. Naive Bayes**
- 8. K-Means Clustering**
- 9. Principal Component Analysis (PCA)**
- 10. Ensemble Methods (XGBoost, LightGBM)**

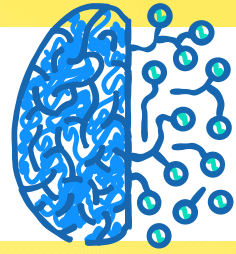
This progression moves from simple, interpretable models to more advanced techniques and provides a strong foundation before you move on to Neural Networks, which is the next chapter.

Key Takeaway

Machine Learning isn't about memorising algorithms—it's about understanding how models learn from data and how to evaluate them effectively.

Chapter 8

Neural Network



Brief Overview

Neural Networks are a type of Machine Learning model inspired by the human brain. They are designed to recognize patterns in data and form the foundation of modern AI systems, including image recognition, speech recognition, and Large Language Models (LLMs).

Almost every breakthrough in modern AI is powered by neural networks.

Why Learn Neural Networks?

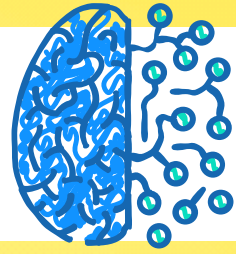
Learning Neural Networks helps you:

- **Understand how modern AI models work**
- **Build a strong foundation for Deep Learning**
- **Understand how LLMs are trained**
- **Learn the concepts behind image, speech, and text models**
- **Prepare for learning Transformers and LLMs**

Topic	What to Learn	Why It Matters
 Perceptron	Inputs, Weights, Bias, Activation	The basic building block of every neural network.
 Activation Functions	ReLU, Sigmoid, Tanh, Softmax	Help neural networks learn complex patterns.
 Loss Functions	MSE, Cross Entropy	Measure how well a model is performing during training.
 Gradient Descent	Learning Rate, Optimization	The algorithm used to minimize errors during training.
 Backpropagation	Error Propagation	Allows neural networks to improve by updating their weights.
 Deep Neural Networks	Hidden Layers, Forward Pass	Learn how multiple layers improve learning capability.
 CNNs	Convolution, Pooling	Used for computer vision and image-related tasks.
 RNNs & LSTMs	Sequential Data	Understand how earlier sequence models worked before Transformers.
 Transformers (Introduction)	Self-Attention, Encoder, Decoder	The architecture behind modern LLMs like GPT and Claude.

Chapter 8

Neural Network



Recommended Resources

Books

Deep Learning — Ian Goodfellow, Yoshua Bengio & Aaron Courville

Often called the definitive book on Deep Learning. Best used as a reference while learning.

YouTube

3Blue1Brown

Watch the "Neural Networks" series for one of the clearest visual explanations of how they work.

StatQuest with Josh Starmer

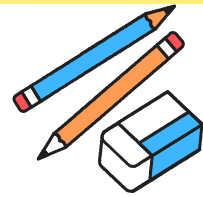
Excellent explanations of activation functions, gradient descent, backpropagation, and neural network concepts.

Andrej Karpathy

Provides practical insights into how neural networks and modern AI models are built.

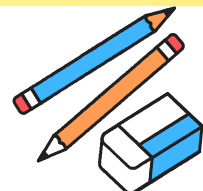
 **Official Documentation**
PyTorch Documentation
TensorFlow Documentation

- **Keras Documentation**



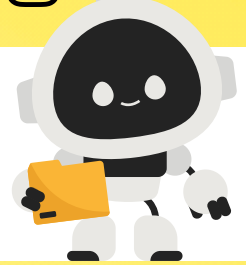
Learning Checklist

- ☐ I understand what a neural network is.
- ☐ I know how a perceptron works.
- ☐ I understand activation functions.
- ☐ I understand gradient descent and backpropagation conceptually.
- ☐ I know the difference between CNNs, RNNs, and Transformers.
- ☐ I understand why Transformers became the foundation of modern AI.



Key Takeaway

Neural Networks transformed AI by enabling computers to learn complex patterns from data. Today's LLMs are built on this foundation, making neural networks one of the most important concepts to understand before moving into Generative AI.



Chapter 9

LLMs

Brief Overview

Large Language Models (LLMs) are deep learning models trained on massive amounts of text to understand and generate human language. They power applications like ChatGPT, Claude, Gemini, GitHub Copilot, and many AI assistants used today.

For an AI Engineer, understanding how LLMs work is essential because they're the foundation of most modern AI applications.

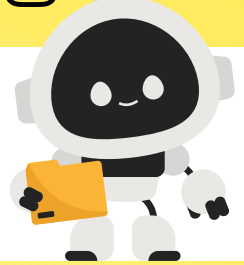
Why Learn LLMs?

Learning LLMs helps you:

- Understand how modern AI assistants work
- Build AI-powered applications
- Work with APIs from OpenAI, Anthropic, Google, and others
- Learn Prompt Engineering, RAG, and AI Agents
- Build production-ready AI systems

What to Learn

Topic	What to Learn	Why It Matters
 Tokens & Tokenization	Tokens, Context Window	Understand how LLMs process text.
 Embeddings	Vector Representations	Learn how text is converted into numerical vectors.
 Transformers	Self-Attention, Attention Mechanism	The architecture behind modern LLMs.
 Prompt Engineering	Prompt Structure, Context, Examples	Improve the quality of LLM responses.
 Inference Parameters	Temperature, Top-p, Max Tokens	Control how models generate responses.
 Fine-tuning	Fine-tuning vs Prompting	Learn when model customization is necessary.
 RAG (Introduction)	External Knowledge	Understand why LLMs need access to your own data.
 Model APIs	OpenAI, Anthropic, Gemini, OpenRouter	Learn how to integrate LLMs into applications.



Chapter 9

LLMs

Recommended Resources

Books

AI Engineering — Chip Huyen

One of the best books for understanding how modern LLM applications are built and deployed.

YouTube

Andrej Karpathy

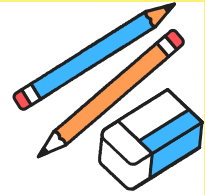
Excellent explanations of Transformers, tokenisation, LLMs, and modern AI.

DeepLearning.AI

High-quality tutorials covering LLMs, prompt engineering, RAG, and AI agents.

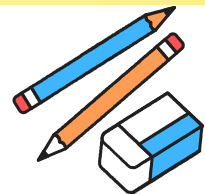
Official Documentation

- OpenAI Documentation
- Anthropic Documentation
- Google AI (Gemini) Documentation
- Hugging Face Documentation



Learning Checklist

- ☐ I understand what an LLM is.
- ☐ I know how tokenisation works.
- ☐ I understand embeddings at a high level.
- ☐ I know what a Transformer is and why it matters.
- ☐ I understand prompt engineering fundamentals.
- ☐ I know the difference between prompting, RAG, and fine-tuning.
- ☐ I can integrate an LLM using an API.



Key Takeaway

As an AI Engineer, you don't need to build an LLM from scratch—but you should understand how it works and how to use it effectively in real-world applications.

Before moving on, try building a simple AI application using an LLM API. Even a basic chatbot will help connect the concepts you've learned so far and make the next chapters on RAG and AI Agents much easier to understand.



Chapter 10

RAG

Brief Overview

Retrieval-Augmented Generation (RAG) is a technique that allows **Large Language Models** to retrieve information from external knowledge sources before generating a response.

Instead of relying only on what the model learned during training, RAG lets the model access documents, databases, PDFs, websites, or company knowledge in real time.

This makes responses more accurate, up-to-date, and relevant.

Why Learn RAG?

Learning RAG helps you:

- Build AI chatbots on your own data
- Reduce hallucinations
- Keep AI responses up to date
- Build enterprise AI applications
- Understand how most production AI systems work

What to Learn

Topic	What to Learn	Why It Matters
 Document Loading	PDFs, CSVs, Websites, Databases	Learn how to ingest different data sources.
 Chunking	Chunk Size, Overlap, Recursive Splitting	Break large documents into searchable pieces.
 Embeddings	Text Embeddings, Embedding Models	Convert text into vectors for semantic search.
 Vector Databases	Chroma, Pinecone, Weaviate, Qdrant, FAISS,	Store and search embeddings efficiently.
 Retrieval	Similarity Search, Top-K Retrieval	Retrieve the most relevant information.
 Hybrid Search	Keyword + Semantic Search	Improve retrieval quality.
 Re-ranking	Cross Encoders, Re-rankers	Improve the relevance of retrieved documents.
 RAG Evaluation	Precision, Recall, Groundedness	Measure the quality of your RAG system.

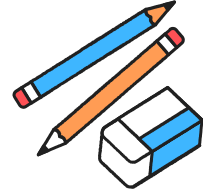


Chapter 10

RAG

Recommended Resources

Books



AI Engineering — Chip Huyen

Excellent coverage of production AI systems, including RAG architectures.

YouTube

DeepLearning.AI

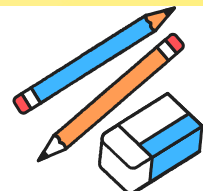
Excellent videos on RAG, embeddings, vector databases, and production AI.

LangChain

Official tutorials covering RAG pipelines and real-world implementations.

- #### Official Documentation
- **LangChain Documentation**
 - **LangGraph Documentation**
 - **LlamaIndex Documentation**
 - **Chroma Documentation**
 - **Pinecone Documentation**
 - **Qdrant Documentation**
 - **Weaviate Documentation**
 - **pgvector Documentation**

Learning Checklist



- ☐ **I understand what RAG is.**
- ☐ **I know why RAG is used instead of only prompting.**
- ☐ **I understand chunking and why it's important.**
- ☐ **I know how embeddings work at a high level.**
- ☐ **I understand how vector databases store embeddings.**
- ☐ **I can build a basic RAG pipeline.**
- ☐ **I understand how to evaluate a RAG system.**

Key Takeaway

LLMs know general knowledge. RAG gives them access to your knowledge.

Chapter 11

AI Agents

Brief Overview

AI Agents are systems that use Large Language Models (LLMs) to reason, plan, use tools, and complete tasks autonomously.

Unlike a traditional chatbot that only responds to prompts, an AI Agent can interact with APIs, search the web, access databases, execute code, and perform multi-step workflows.

AI Agents are becoming a key part of modern AI applications.

Why Learn AI Agents?

Learning AI Agents helps you:

- **Build autonomous AI applications**
- **Automate repetitive workflows**
- **Connect AI with external tools and APIs**
- **Create multi-step reasoning systems**
- **Build production-ready AI assistants**

What to Learn

Topic	What to Learn	Why It Matters
 AI Agent Basics	What an AI Agent is, Agent Architecture	Understand how agents differ from chatbots.
 Tool Calling	Function Calling, APIs, External Tools	Enable agents to interact with the outside world.
 Memory	Short-term & Long-term Memory	Help agents remember conversations and context.
 Planning	Task Decomposition, Multi-step Reasoning	Allow agents to solve complex problems step by step.
 Workflows vs Agents	Fixed Workflows vs Autonomous Agents	Learn when to use each approach.
 Model Context Protocol (MCP)	MCP Basics	Standardize how AI models connect with tools and data sources.
 Guardrails	Validation, Safety, Output Constraints	Build reliable and secure AI systems.
 Multi-Agent Systems	Agent Collaboration	Coordinate multiple agents to solve larger tasks.
 Agent Frameworks	LangGraph, OpenAI Agents SDK, PydanticAI, CrewAI	Learn the tools commonly used to build and orchestrate AI agents.

Chapter 11

AI Agents



Recommended Resources

Books

AI Engineering — Chip Huyen

Provides practical insights into building AI systems and modern agentic applications.

YouTube

LangChain

Official tutorials on AI Agents, LangGraph, and production AI applications.

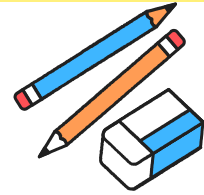
DeepLearning.AI

High-quality content covering AI Agents, MCP, workflows, and LLM applications.



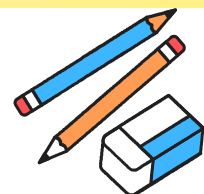
Official Documentation

- [LangChain Documentation](#)
- [LangGraph Documentation](#)
- [OpenAI Documentation \(Function Calling\)](#)
- [Anthropic Documentation \(Tool Use\)](#)
- [Model Context Protocol \(MCP\) Documentation](#)



Learning Checklist

- ☐ I understand what an AI Agent is.
- ☐ I know the difference between a chatbot and an AI Agent.
- ☐ I understand tool calling and function calling.
- ☐ I know when to use workflows instead of agents.
- ☐ I understand memory in AI Agents.
- ☐ I understand the basics of MCP.
- ☐ I know why guardrails are important.
- ☐ I can build a simple AI Agent.



Key Takeaway

LLMs know general knowledge. RAG gives them access to your knowledge.

Chapter 12

Software Eng. & System Design

Brief Overview

Building an AI model is only part of the job.

As an AI Engineer, you'll also need to build APIs, manage databases, design scalable systems, deploy applications, and ensure your AI products are reliable in production.

Software Engineering helps you build the application, while System Design helps it scale.

Why Learn Software Engineering & System Design?

These skills help you:

- **Build production-ready AI applications**
- **Connect AI models with users and databases**
- **Design scalable and reliable systems**
- **Improve application performance**
- **Deploy and maintain AI services**
- **Collaborate effectively with engineering teams**

Chapter 12

Software Eng. & System Design

What to Learn :

Category	Topics	Why It Matters
 APIs	REST APIs, FastAPI, JSON	Connect AI models with applications.
 Authentication	JWT, OAuth, API Keys	Secure AI applications and user access.
 Docker	Containers, Docker Compose	Package and deploy applications consistently.
 Git & GitHub	Version Control, Branching, Pull Requests	Collaborate and manage code efficiently.
 Testing	Unit Testing, Integration Testing	Build reliable and maintainable applications.
 Caching	Redis, Cache Strategies	Reduce latency and improve application performance.
 Load Balancing	Reverse Proxy, Load Balancers	Handle increased traffic and improve reliability.
 Queues	RabbitMQ, Kafka, Background Jobs	Process long-running AI tasks asynchronously.
 Databases	PostgreSQL, Supabase, Redis	Store application and AI-related data.
 Architecture	Monoliths, Microservices	Design maintainable applications.
 Scalability	Horizontal & Vertical Scaling	Support growing numbers of users.
 Deployment	CI/CD, Cloud Platforms	Deploy AI applications to production.
 Monitoring	Logging, Metrics, Observability	Detect issues and monitor production systems.

Chapter 12

Software Eng. & System Design

Recommended Resources

Books

Designing Data-Intensive Applications — Martin Kleppmann

One of the best books for understanding scalable software systems, databases, distributed systems, and system design.

System Design Interview – Volume 1 — Alex Xu

A beginner-friendly introduction to system design concepts through real-world examples.

Github Repo
[system-design-primer](#)

YouTube

ByteByteGo

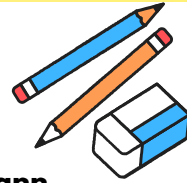
Excellent explanations of distributed systems, databases, caching, load balancers, and scalability.

Hussein Nasser

Deep dives into backend engineering, networking, databases, APIs, and system architecture.

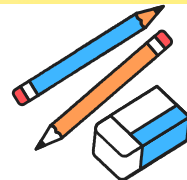
Official Documentation

- **FastAPI Documentation**
- **Docker Documentation**
- **Git Documentation**
- **PostgreSQL Documentation**
- **Supabase Documentation**
- **Redis Documentation**



Learning Checklist

- ☐ I understand REST APIs and FastAPI.
- ☐ I know how authentication works.
- ☐ I can use Git and GitHub effectively.
- ☐ I understand Docker basics.
- ☐ I know when to use PostgreSQL, Supabase, and Redis.
- ☐ I understand caching and why it improves performance.
- ☐ I understand background jobs and message queues.
- ☐ I understand monoliths and microservices.
- ☐ I know the basics of load balancing and scalability.
- ☐ I understand the fundamentals of system design.
- ☐ I know how AI applications are deployed and monitored.



Key Takeaway

A great AI model isn't enough. AI Engineers build complete, scalable, and reliable systems that people can actually use.

 [@riwano.py](https://www.instagram.com/riwano.py)

Appendix

AI Glossary

A

API (Application Programming Interface)

A way for different software applications to communicate with each other. AI Engineers use APIs to integrate AI models into applications.

Agent

An AI system that can reason, plan, use tools, and perform tasks autonomously.

Attention

A mechanism that helps Transformer models focus on the most relevant parts of the input.

B

Backpropagation

The process used by neural networks to learn by adjusting weights based on prediction errors.

Bias

A value added to a neuron that helps improve a model's ability to learn patterns.

C

Chunking

The process of splitting large documents into smaller sections before storing them in a vector database.

Context Window

The maximum amount of information an LLM can process at one time.

Cross Validation

A technique used to evaluate how well a machine learning model performs on unseen data.

D

Data Analysis

The process of cleaning, exploring, and understanding data before building machine learning models.

Deep Learning

A branch of Machine Learning that uses neural networks with multiple layers.

Appendix

AI Glossary

E

Embedding

A numerical vector representation of text, images, or other data that captures meaning.

Epoch

One complete pass of the training dataset through a machine learning model.

F

Feature

An individual input variable used by a machine learning model.

Feature Engineering

The process of creating or selecting better input features to improve model performance.

Fine-tuning

Training an already pre-trained model on a specific dataset for a particular task.

G

Gradient Descent

An optimization algorithm used to minimize a model's error during training.

Guardrails

Rules and constraints added to AI systems to improve safety and reliability.

H

Hallucination

When an AI model generates incorrect or fabricated information that sounds believable.

Hyperparameter

A setting chosen before training a model, such as learning rate or batch size.

I

Inference

The process of using a trained AI model to generate predictions or responses.

Appendix

AI Glossary

L

LLM (Large Language Model)

A deep learning model trained on massive amounts of text to understand and generate language.

Linear Algebra

A branch of mathematics involving vectors and matrices, essential for machine learning.

Loss Function

A function that measures how far a model's predictions are from the correct answers.

M

Machine Learning

A field of AI where computers learn patterns from data instead of being explicitly programmed.

Matrix

A rectangular collection of numbers used to represent data in machine learning.

Model

A trained system that makes predictions or generates outputs from input data.

Model Context Protocol (MCP)

An open standard that allows AI models to securely connect with external tools, data sources, and applications.

N

Neural Network

A machine learning model inspired by the human brain, consisting of interconnected layers of neurons.

Node (Neuron)

A single computational unit inside a neural network.

O

Overfitting

When a model memorizes training data and performs poorly on new, unseen data.

Appendix

AI Glossary

P

Parameter

A value learned by a machine learning model during training.

Perceptron

The simplest form of a neural network and the foundation of modern deep learning.

Probability

The likelihood of an event occurring, used extensively in machine learning.

Prompt

The input given to a Large Language Model.

Prompt Engineering

The practice of designing prompts that produce better AI responses.

R

RAG (Retrieval-Augmented Generation)

A technique that allows an LLM to retrieve external information before generating a response.

Regression

A machine learning task used to predict continuous numerical values.

Reinforcement Learning

A type of machine learning where agents learn through rewards and penalties.

Retrieval

The process of finding relevant information from a knowledge base.

S

Scikit-learn

A popular Python library for classical machine learning.

SQL

A language used to query and manage relational databases.

Appendix

AI Glossary

S

Scikit-learn

A popular Python library for classical machine learning.

SQL

A language used to query and manage relational databases.

Supervised Learning

Machine learning using labeled training data.

System Design

The process of designing scalable, reliable, and maintainable software systems.

T

Temperature

A parameter that controls how random or creative an LLM's responses are.

Token

The basic unit of text processed by an LLM.

Tokenization

The process of breaking text into tokens.

Transformer

The neural network architecture behind modern LLMs like GPT and Claude.

U

Underfitting

When a model is too simple to learn meaningful patterns from the data.

Unsupervised Learning

Machine learning that finds patterns in unlabeled data.

Appendix

AI Glossary

V

Vector

A list of numbers representing data mathematically.

Vector Database

A database designed to store and search vector embeddings efficiently.

W

Weight

A value inside a neural network that determines how strongly one neuron influences another.

This glossary isn't meant to replace detailed learning resources. It's a quick reference to help you remember the most common AI Engineering terms you'll encounter throughout this handbook and your learning journey.

Final Advice

Learning AI Engineering can feel overwhelming.

**Every week there's a new model, framework, or tool claiming to change everything.
Don't try to learn everything.**

Focus on understanding the fundamentals. Technologies will change, but concepts like Python, Machine Learning, Neural Networks, Software Engineering, and System Design will continue to be valuable.

**Build as you learn.
Reading books and watching tutorials will help you understand concepts, but building projects is what turns knowledge into skill.**

**Don't wait until you feel "ready."
You probably never will.**

Start with a simple project, make mistakes, debug problems, and keep improving. Every project will teach you something that no tutorial can.

**Remember, this handbook isn't the finish line—it's your starting point.
Stay curious.
Keep building.
And most importantly, enjoy the journey.**

**Thank you for reading.
— Riwano Fariz**

**You don't become an AI Engineer by finishing this handbook.
You become one by consistently building, learning, failing, and improving.**

If this handbook helped you, consider sharing it with someone who's starting their AI Engineering journey.