Introduction to AI Engineering with LLMs

The leap from algorithms to reality

Pascal Sun

UWA NLP-TLP Group

The University of Western Australia

August 7, 2025

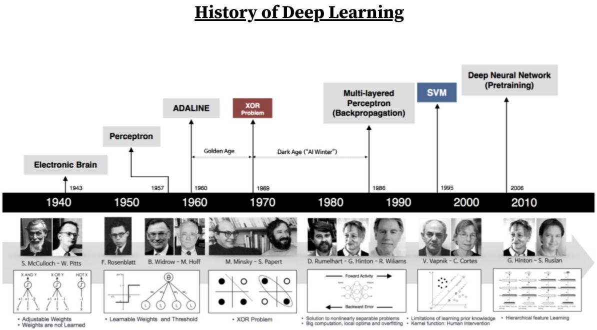
Table of Contents

Introduction

Several questions to address today

- First things first - what is **AI** really?
- Who is this *AI Engineer* and what do they do?
- Why isn't regular programming enough anymore?
- How do you actually make AI work in the real world?
- What skills and knowledge do **YOU** need to join this revolution?

Why AI? Why Now?



Title: Evolution of Deep Learning from 1940-2010
Source: *Deep Learning blog by Beamlab*

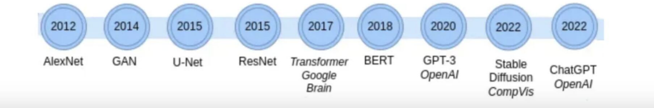


Figure: Where we from?

The Developer Identity

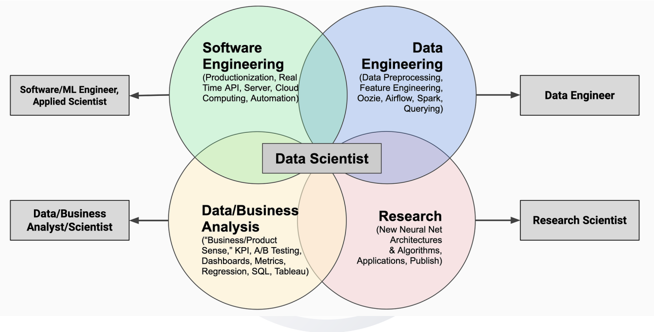


Figure: The Landscape of Modern Data Roles[1]

[1] https://www.geeksforgeeks.org/artificial-intelligence/data-scientist-vs-ai-engineer-which-is-better/

The Developer Identity

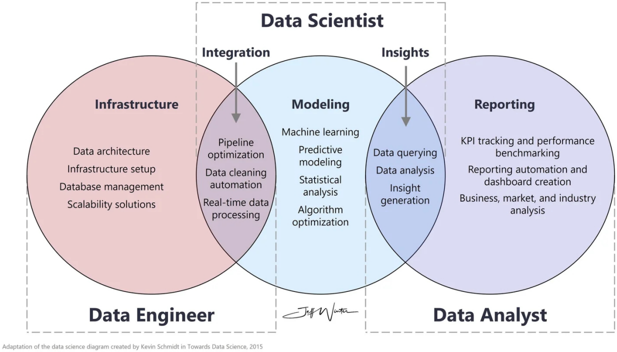


Figure: Responsibilities Breakdown[2]

[2] https://www.datacamp.com/blog/data-scientist-vs-data-engineer

AI Development Lifecycle

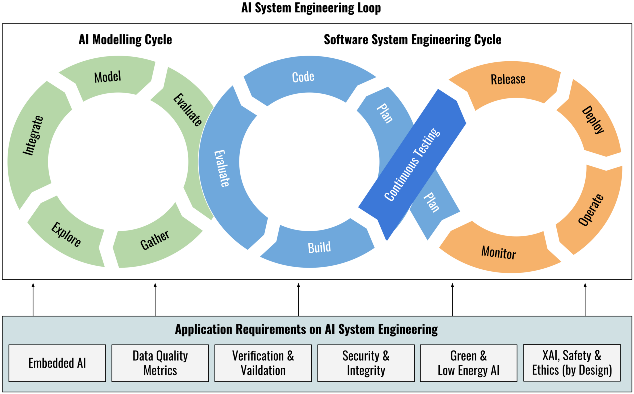


Figure: The life cycle for AI product development[3]

[3] https://speakerdeck.com/redhatlivestreaming/apac-hybrid-cloud-kopi-hour-e6-ai

What is AI?

Three AI Pillars

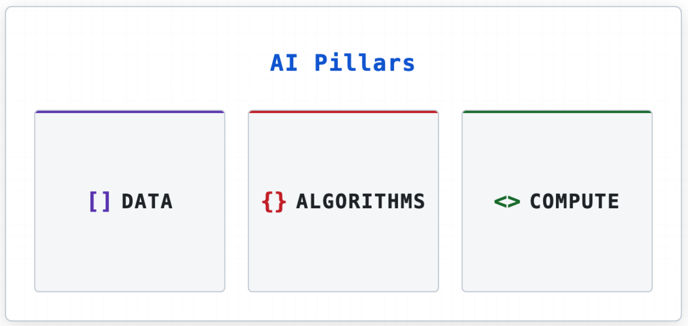


Figure: Data, Algorithms, and Computational Resources

Data: Seven Key Data Dimensions

Dimension	**Low / Simple End**	**High / Complex End & Engineering Impact**
Modality	Tabular numbers	Text → Image → Audio → Video → **Multimodal**
Structure	Fully structured (SQL)	JSON / Graph → Unstructured Texts
Volume	MB – GB	TB – PB lakes; drives cost & distributed training
Velocity	Batch loads	Real-time streams; calls for Spark/Flink
Veracity	Clean, labeled	Noisy, biased; heavier data-ops, active learning
Sensitivity	Public	PII / trade secrets; encryption, privacy tech
Lifecycle	Raw capture	versioned assets; governs MLOps rollback

Data: Data Types → Model Families

Data Type	**Typical Model Families**	**Representative Use-cases**
Tabular	Linear / Logistic Reg	Credit scoring, sales forecast
Text	RNN; Transformer / LLM; Word2Vec/BERT	Chatbots, sentiment analysis
Image	CNN; Vision Transformer; Diffusion	Defect detection, medical imaging
Audio	1-D CNN; Spectrogram CNN	ASR, machine health diagnostics
Video	3-D CNN; Temporal Transformer	Surveillance, action recognition
Time series	TCN; LSTM; Prophet	Predictive maintenance, trading
Multimodal	Cross-modal Transformer; CLIP	Image–text search, VQA

Data: How Models “Push Back” on Data

- **Supervision:** Classification/detection needs precise labels (bounding boxes, token spans).
- **Resolution & Sampling:** CNNs depend on image resolution; audio nets need 16 kHz; TS models like uniform spacing.
- **Context Length:** LLMs want long windows; video Transformers require frame chunking.
- **Class Balance:** Imbalance $\rightarrow$ resampling or cost-sensitive loss, or metrics lie.
- **Continuous Feed:** Online systems need fresh data for incremental fine-tuning.
- **Privacy & Compliance:** Medical/finance data may mandate de-identification, FL, or DP.

Algorithms: Evolution Timeline

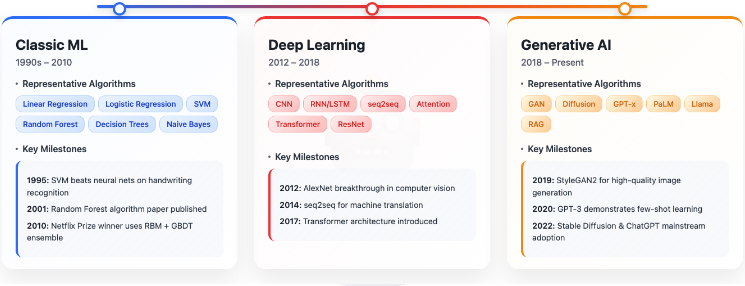


Figure: From classic ML models to generative AI

Algorithms: Several core concepts

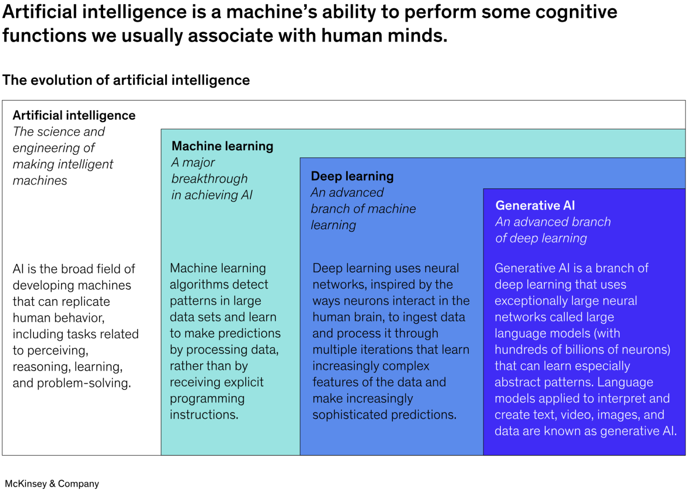


Figure: Explain about several core concepts from McKinsey[4]

[4] https://www.mckinsey.com/featured-insights/mckinsey-explainers/

Algorithms: Classic ML – Quick Hits

Task	**Common Alg.**	**Production Use-cases**
Classification	Logistic Reg.; Random Forest	Fraud detection; email spam filter.
Regression	Gradient Boosting (XG-Boost)	Ride-hailing demand forecast; price-elasticity modeling.
Ranking / Retrieval	SVM-Rank; LambdaMART	Search result ordering; recommendation systems v1.
Clustering	k-Means; DBSCAN	Customer segmentation; anomaly hunt in sensor fleets.

- Features **hand-crafted**: tf-idf n-grams (NLP), SIFT/HOG (CV), domain KPIs (tabular).
- Still a strong baseline for tabular & small-data regimes—cheap to train, easy to explain.

Algorithms: Deep Learning Revolution, End-to-End Learning

Computer Vision

↓

CNN → ResNet → Vision Transformer

↓

Medical imaging ▪ Radiology diagnosis ▪ Pathology analysis

Natural Language Processing

↓

RNN → Attention → Transformer

↓

Clinical notes ▪ Drug discovery ▪ Medical QA systems

Key Breakthrough: Eliminates manual feature engineering ▪ Scales with data + compute

Algorithms: Generative AI — What & Why?

Two Types of AI Models

Discriminative Models

- Learn $P(y \mid x)$ — decision boundaries
- Examples: SVM, ResNet, BERT

Generative Models

- Learn $P(x)$ or $P(x \mid c)$ — data distribution
- Examples: VAE, GAN, GPT

Key Difference

Discriminative: tells you **what** something is
Generative: shows you **what else** could exist

Key Milestones

2014 — GANs
Adversarial training → realistic images

2017 — Transformers
Self-attention → scalable pre-training

2019–2024 — LLMs & Diffusion
GPT-3/4, Stable Diffusion, Sora

If a model ***explicitly samples*** *new data that looks plausible, we call it* ***generative***.

Compute: From Hobby to Hyperscale

Era	**TFLOPS**	**\$/T-FLOP**	**AI Research Impact**
CPU Clusters	0.1	\$10k+	Hand-crafted features, SVMs
(Pre-2010)			Weeks for MNIST training
Gaming GPUs	1-3	\$1k	Deep CNNs, ImageNet breakthrough
(2012 AlexNet)			Backprop at scale
Data Center GPUs	15-30	\$100	Large NLP, ResNets, Transformers
(V100 2017-20)			BERT-scale pretraining
AI ASICs	100-300	\$10	LLMs (GPT-3+), diffusion models
(H100, TPU v4+)			100B+ param models
Cloud Elastic	Variable	\$1-5	Democratized access
(2023+)			Pay-per-second training

Key Insight: Every 2-3 years → 10× compute + 2× efficiency → qualitatively new AI capabilities

Compute: The Software Stack: Write Once, Run Anywhere

Core Frameworks

- **PyTorch**: Research-friendly
- **JAX**: Fast compilation
- **TensorFlow**: Production-ready

Scale-Up Tools

- **DeepSpeed**: Memory-efficient training
- **Ray**: Distributed computing

Hardware Layer

- **CUDA**: GPU programming
- **Docker**: Reproducibility

Why It Matters

- **Flexibility**: Same code, different hardware
- **Cost Control**: Switch providers easily
- **Speed**: Less infrastructure setup
- **Collaboration**: Reproducible experiments

Key Principle

Build portable first, optimize later.

Research Tip: Use cloud notebooks (Colab) to test ideas quickly.

Compute: Market Dynamics: The NVIDIA Gold Rush

The Stock Story

- **2012**: AlexNet breakthrough → \$4B market cap
- **2016-18**: Deep RL boom → \$30-100B
- **2020-22**: COVID + enterprise AI → \$300-500B
- **2023-24**: ChatGPT frenzy → \$1-3T (world's most valuable)

What This Means

- GPU shortages → higher cloud costs
- New competitors emerge (Groq, Cerebras)
- Geopolitical chip competition intensifies

Key Lesson

AI breakthroughs → market boom → more R&D funding → better hardware → new breakthroughs

For Researchers

- Follow compute trends for opportunity
- Diversify hardware dependencies
- Leverage cloud democratization

Disclaimer: Yacht cooling systems remain fictional.

What is AI Engineer?

AI Engineer: Unified Role Definition

AI Engineer =
Research + Engineering + Operations

RESEARCH & BUILD

- Algorithm Design
- Model Architecture
- Experiment Framework
- Performance Analysis
- Literature Review

ENGINEER & SHIP

- System Architecture
- API Development
- Scalability Design
- Security Implementation

OPERATE & MONITOR

- Performance Monitoring
- Model Drift Detection
- Incident Management
- Reliability Engineering

Technical Competency Matrix

Skill Domain	Research	Engineering	Operations
Programming	Python, R, Julia	Python, Go, Rust	Python, Bash, SQL
ML/DL Frameworks	PyTorch, JAX	PyTorch, ONNX	TensorRT, Serve
Data Processing	Pandas, Dask	Spark	Kafka
Infrastructure	Jupyter, Colab	Docker, K8s	Prometheus, Grafana
Cloud Platforms	SageMaker	Vertex AI	CloudWatch
Version Control	Git, DVC	Git, GitOps	Git, IaC

Proficiency Requirement: Expert in one area, proficient in others, conversational across all

Daily Workflow: Research-Heavy Days

Deep Work (4-6h)

- Literature review & analysis
- Model architecture design
- Ablation studies & experiments
- Paper writing & reading
- Mathematical formulation
- Data exploration

Collaboration (2-3h)

- Research meetings
- Peer review sessions
- Knowledge sharing
- Mentoring

Infrastructure (1-2h)

- Experiment tracking
- Data pipeline maintenance

Focus: Innovation, discovery, and advancing the state-of-the-art

Daily Workflow: Engineering-Heavy Days

Development (5-6 hours)

- Production code implementation
- API design & development
- Performance optimization & profiling
- Integration & unit testing
- Code review & refactoring
- Database schema design

Architecture (2-3 hours)

- System design reviews
- Infrastructure planning
- Security and scalability implementation
- Technical documentation

Deployment (1-2 hours)

- CI/CD pipeline updates
- Staging environment testing
- Production deployments

Focus: Building robust, scalable, and maintainable ML systems

Daily Workflow: Operations-Heavy Days

Monitoring (3-4h)

- Dashboard analysis & KPI tracking
- Alert investigation & triage
- Performance troubleshooting
- Incident response & resolution
- Root cause analysis
- System health assessments

Maintenance (2-3h)

- System updates & patches
- Data quality checks
- Resource optimization

Planning (1-2h)

- Capacity planning & forecasting
- Process improvements
- Cost optimization analysis

Focus: Ensuring reliability, performance, and operational excellence

How?

Full Stack Architecture

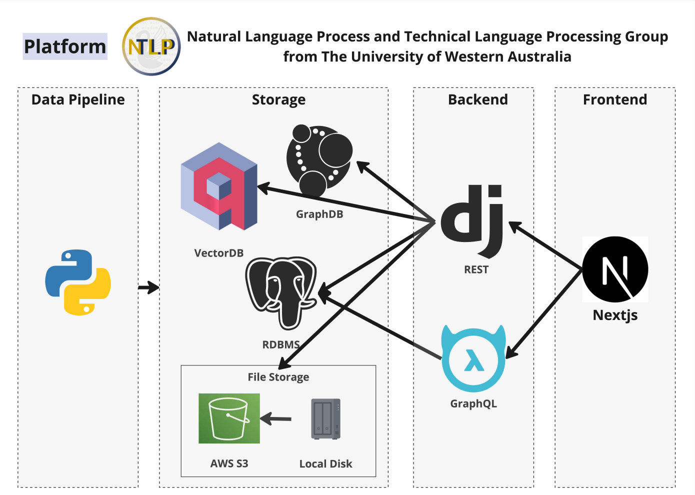


Figure: A fully in control full stack architecture

Fully Cloud Azure Architecture

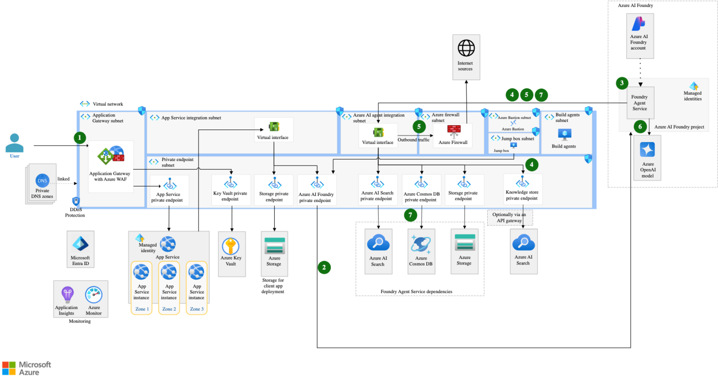


Figure: A more complex version of production deployment[5]

[5] https://learn.microsoft.com/en-us/azure/architecture/ai-ml

RAG High Level Concept Explained

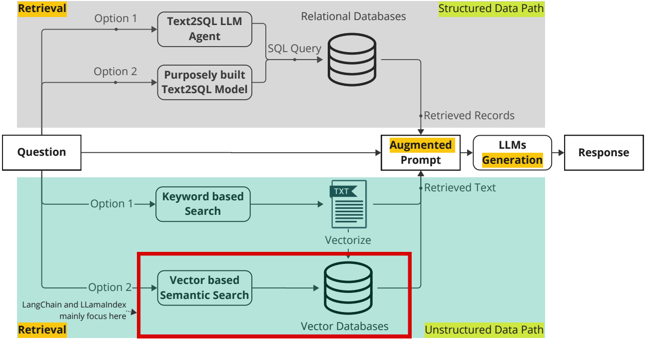


Figure: Typical RAG workflow for structured and unstructured data sources

Most work in RAG predominantly focuses on the **Unstructured Data Path (Option 2)**. LlamaIndex[6] and LangChain[7] start their journey here.

[6] https://www.llamaindex.ai/

[7] https://www.langchain.com/

Model finetuning

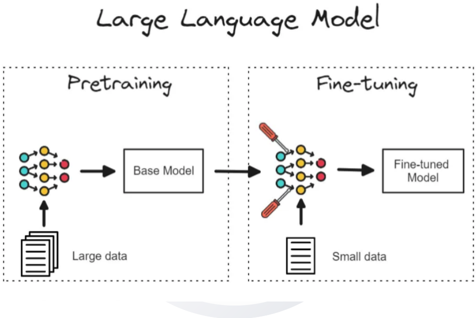


Figure: LLM Finetuning[8]

[8] https://medium.com/@prasadmahamulkar/fine-tuning-phi-2-a-step-by-step-guide-e672e7f1d009

AI Agent

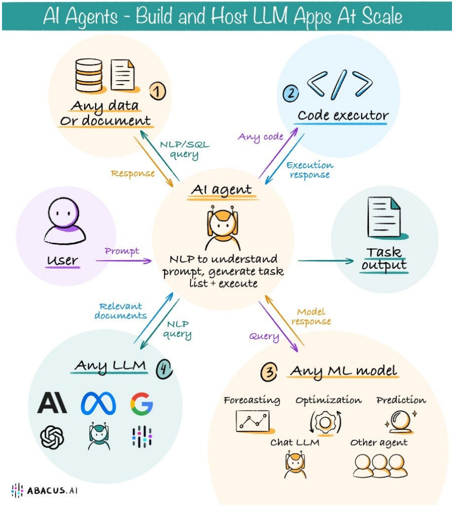
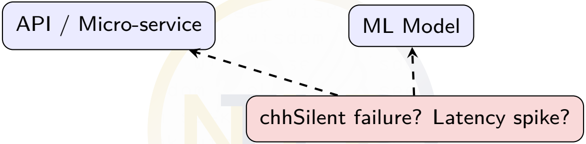


Figure: AI Agent concept illustration

Without Observability — You're Flying Blind



- **Symptoms**: Users see weird outputs or 5-second lags before you do.
- **Cost**: revenue hits, user churn, lost trust, firefighting at 2 a.m.
- **Moral**: if you can't see latency, errors, drift, you can't fix them.

Layered Observability — From Cloud to Model

Layer	**What we watch**
Infra / Arch	CPU, GPU, memory, autoscaling, network errors
App / User	p95 latency, HTTP 5xx, feature flags, user journeys
ML / AI	Accuracy, drift, bias, hallucination rate, cost per 1k calls

Key Takeaway

Instrument *every* layer so signals cascade into **action**—scale pods, cut traffic, trigger fine-tune—**before** users notice.

AI Agent Stack

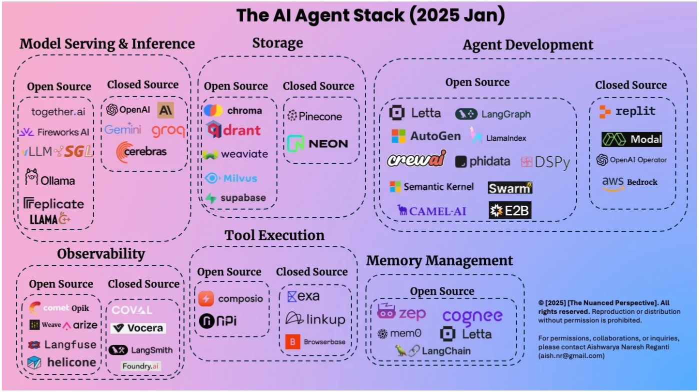


Figure: AI Agent Stack[9]

[9] https://thenuancedperspective.substack.com/p/the-ai-agent-stack-in-2025-how-its

Cursor

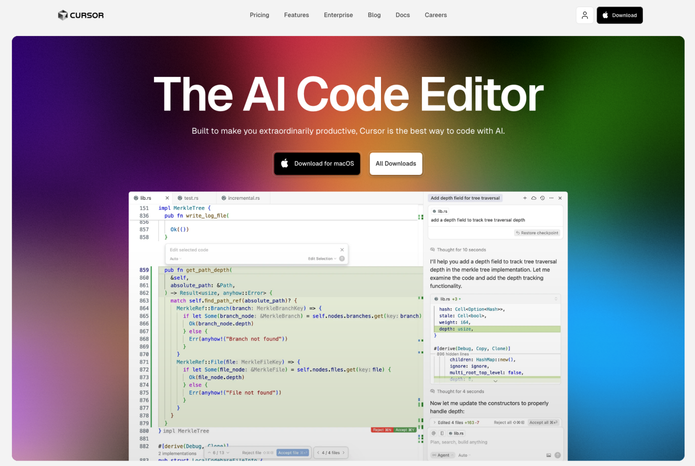


Figure: Your AI Buddy

Q&A

Q&A

Questions?

Feel free to ask anything