

AI COURSE PLAN

IT PROFESSIONALS (CODERS & NON CODERS)



**ADVAITHA INFORMATION
TECHNOLOGY SERVICES
PRIVATE LIMITED (AITS)**



Week 1

Introduction to Artificial Intelligence & Generative AI Tools

Topics:

1. Foundations of AI

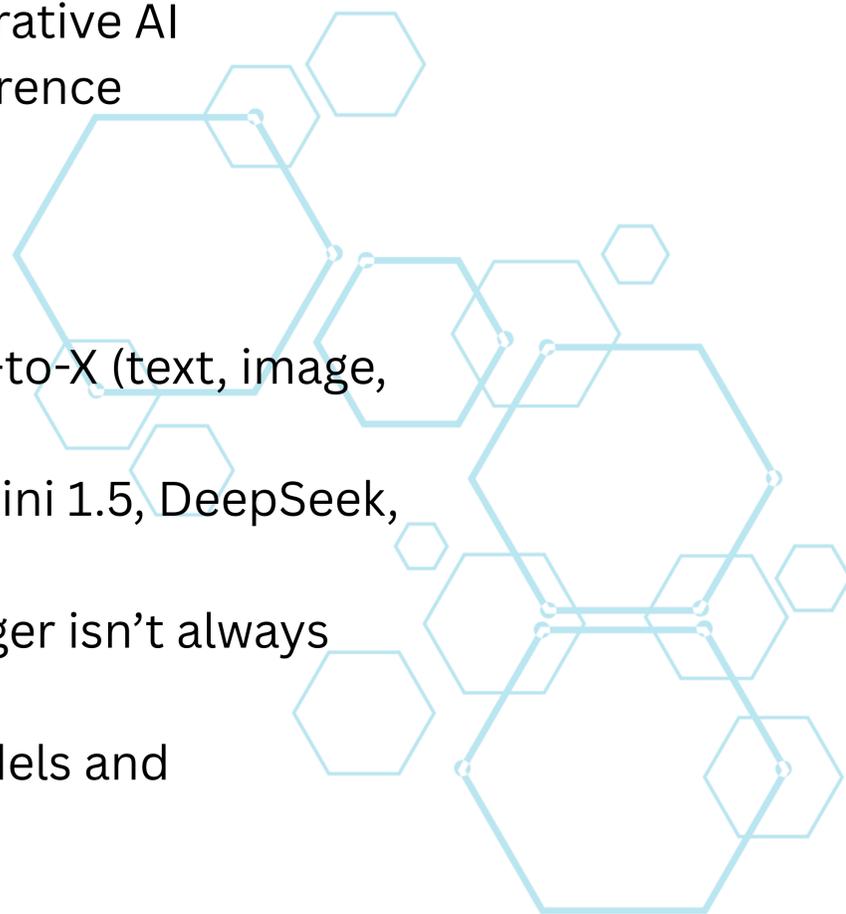
- What is AI, ML, DL, NLP – with practical modern examples
- Evolution from rule-based systems → Generative AI
- Difference between Predictive AI vs Generative AI
- Understanding datasets, models, and inference

2. Generative AI Ecosystem

- What makes GenAI unique: creativity, text-to-X (text, image, code, audio, video)
- Overview of LLMs: GPT-4, Claude 3.5, Gemini 1.5, DeepSeek,
- LLaMA, Mistral
- Understanding “parameters” and why bigger isn’t always better
- Open vs Closed models; open-source models and playgrounds (Ollama, Hugging Face)

3. AI Tools Landscape

- Text Generation: ChatGPT, Claude, Gemini, Copilot Chat
- Image Generation: DALL-E 3, Midjourney, Leonardo AI, Ideogram AI
- Video Generation: Sora, Pika Labs, Runway ML
- Voice & Speech: ElevenLabs, Whisper, PlayHT
- AI Coding Assistants: GitHub Copilot, Replit Ghostwriter, Codeium
- Productivity Copilots: Microsoft 365 Copilot, Google
- Workspace Duet AI, Notion AI



Week 1 (Continued)



4. Understanding LLMs in Action

- How LLMs work (tokenization, embeddings, context window basics)
- Model pipelines and inference explained in plain English
- Understanding temperature, top-p, and reasoning length
- Demonstration: using ChatGPT playground with parameter tuning

5. Introduction to Prompt Engineering

- Prompt structure: system + user + context + examples
- Common prompt styles: role-based, step-by-step, chain-of-thought
- Demonstration: improve outputs using iterative prompting

6. Ethical AI & Responsible Use

- Bias, hallucination, misinformation, and responsible use
- Overview of OpenAI usage policies and safe deployment

Hands-On / Mini Projects

Non-Coders: Create an AI-generated presentation using Notion AI or Gamma.

Coders: Use OpenAI API in Python to build a basic chatbot in Google Colab.

All: Compare responses from ChatGPT, Claude, and Gemini on the same prompt.

Tools Introduced :

ChatGPT Playground · Gemini Studio · Claude AI · Copilot Chat · OpenAI API · Hugging Face Spaces · Notion AI · Gamma App · PowerPoint Copilot · Google Duet AI

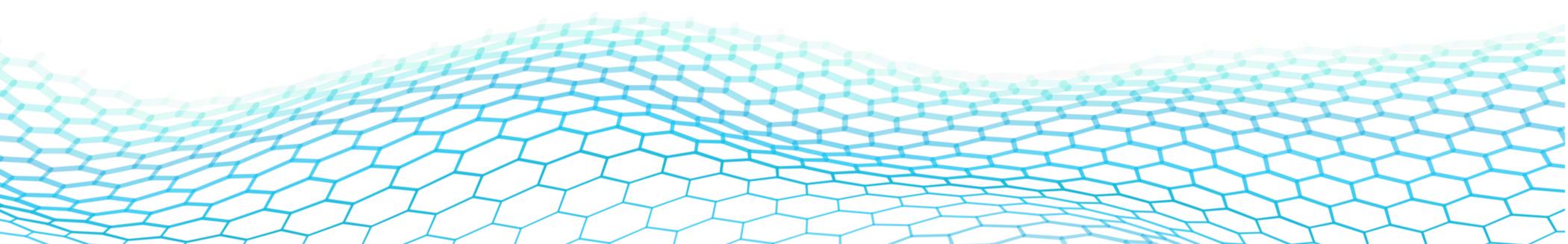
Learning Outcome :

Understand how AI and LLMs generate outputs

Explore major AI tools used in business, education, and coding

Write effective prompts and interact with LLMs

Build confidence using AI for text, code, and media generation





Week 2

Python Fundamentals for GenAI & Data Science

Topics :

1. Python Core Concepts (Foundation)

- Syntax, variables, operators, data types
- Conditional statements (if, elif, else)
- Loops (for, while) and iterations
- Functions, return types, scope, arguments
- Lists, tuples, dictionaries, sets (data structures)
- Basic string handling and f-strings

2. Working in Modern Environments

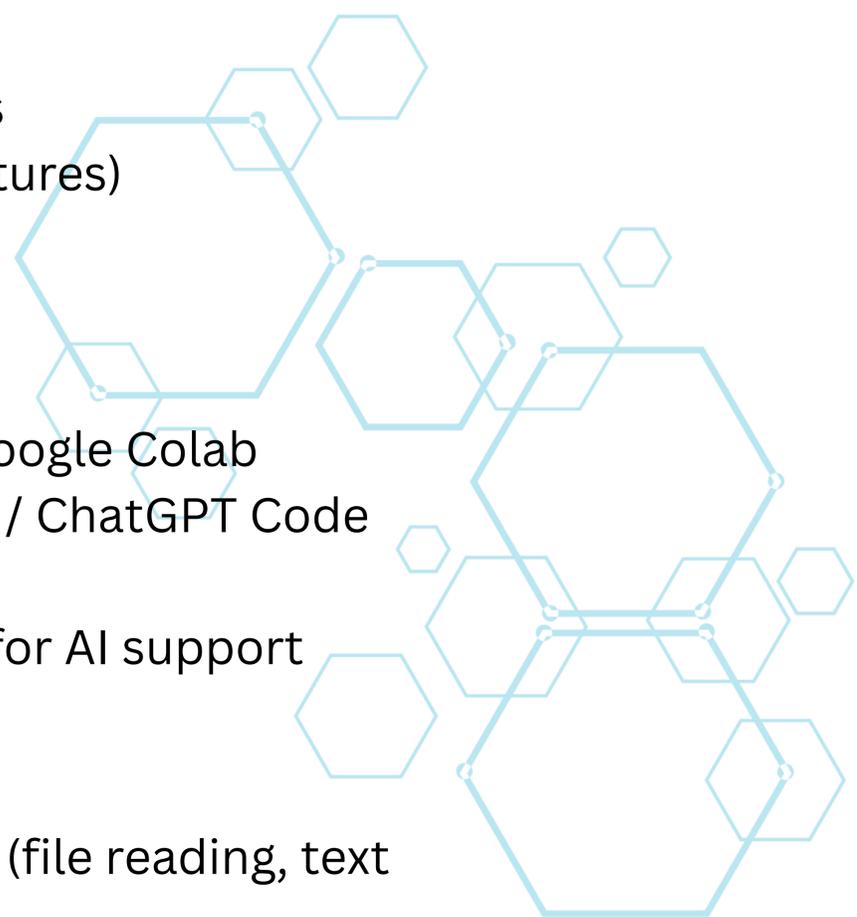
- Installing and using Jupyter Notebook / Google Colab
- AI coding assistance using GitHub Copilot / ChatGPT Code
- Interpreter
- Using Python in VS Code with extensions for AI support

3. Real-World Applications

- Writing your first script to automate tasks (file reading, text summarization)
- Working with CSV and JSON files
- Using ChatGPT API in Python for text generation or translation

Hands - On / Labs

- Create a “Personal AI Assistant” script using the OpenAI API.
- Build a simple “Dataset Explorer” using lists/dicts and loops.





Week 3

Python Intermediate + AI/Data Packages

Topics :

1. Intermediate Python Concepts

- Object-Oriented Programming (Classes, Objects, Inheritance, Polymorphism)
- Exception Handling and Logging
- File I/O and working with external data sources
- Virtual Environments and dependency management

2. Core Data/AI Libraries

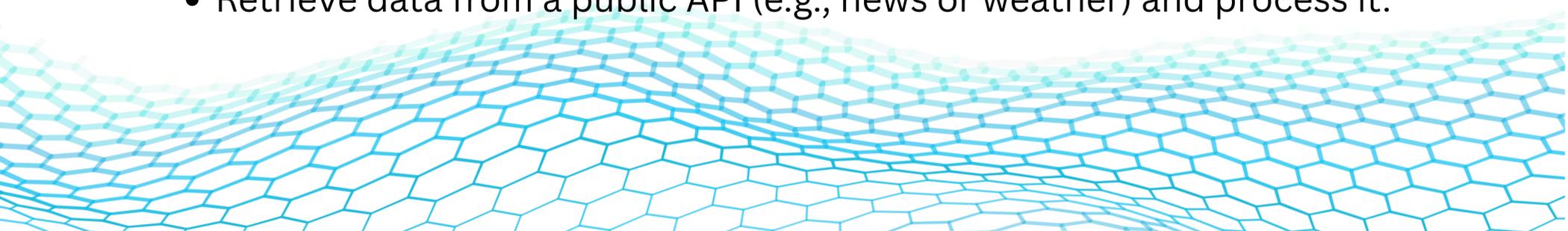
- NumPy: Arrays, broadcasting, numerical operations
- Pandas: Series, DataFrames, filtering, grouping, merging, exporting
- Matplotlib & Seaborn: Basic plotting and visualization
- Regex: Text cleaning and token extraction for NLP tasks
- Requests: Working with APIs and JSON data
- Datetime: Handling timestamps for datasets

3. GenAI-Specific Libraries (Intro Level)

- OpenAI Python SDK → send prompts, handle completions
- LangChain (Preview): Concept of prompt chains
- Hugging Face Transformers: Load and use pre-trained models

Hands-On / Labs

- Load and clean a CSV dataset using Pandas and visualize trends with Seaborn.
- Use OpenAI API to generate summaries for text files.
- Retrieve data from a public API (e.g., news or weather) and process it.





Week 3 (Continued)

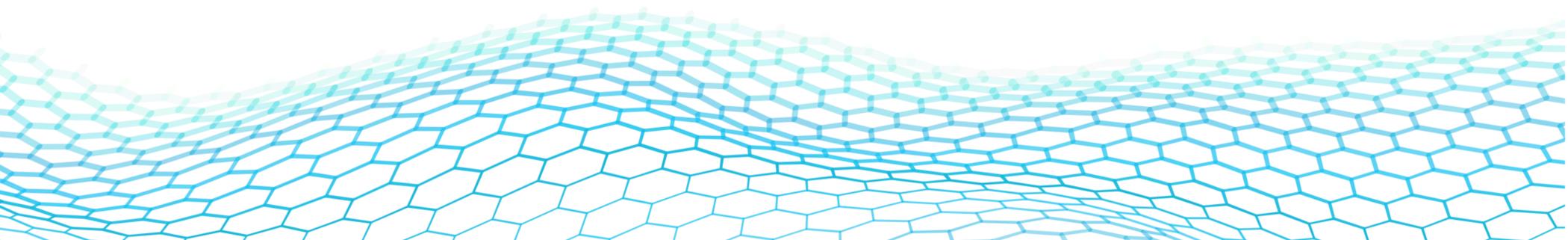
Python Intermediate + AI/Data Packages

Tools & Environment

- Python 3.x | Jupyter Notebook | Google Colab | VS Code | GitHub Copilot | OpenAI SDK | LangChain | Pandas | NumPy Seaborn

Learning Outcomes

- Apply Python for real-world data tasks
- Use core libraries essential for AI & ML projects
- Connect Python scripts with AI APIs
- Be ready for advanced topics like Data Cleaning, Visualization, and Cloud AI (Week 4 onward)





Week 4

Machine Learning And Deep Learning

Topics :

1. What Is Machine Learning?

Definition and categories: Supervised, Unsupervised, Reinforcement

Difference between AI vs ML vs Deep Learning

Real-world ML applications in business, IT, and GenAI systems

ML workflow: data collection → preprocessing → model training → evaluation → deployment

2. Supervised Learning Concepts

Labeled data and target prediction

Algorithms overview:

Linear & Logistic Regression

Decision Trees & Random Forest

Support Vector Machines (SVM)

K-Nearest Neighbors (KNN)

Model evaluation metrics: Accuracy, Precision, Recall, F1, ROC-AUC

Avoiding overfitting and underfitting

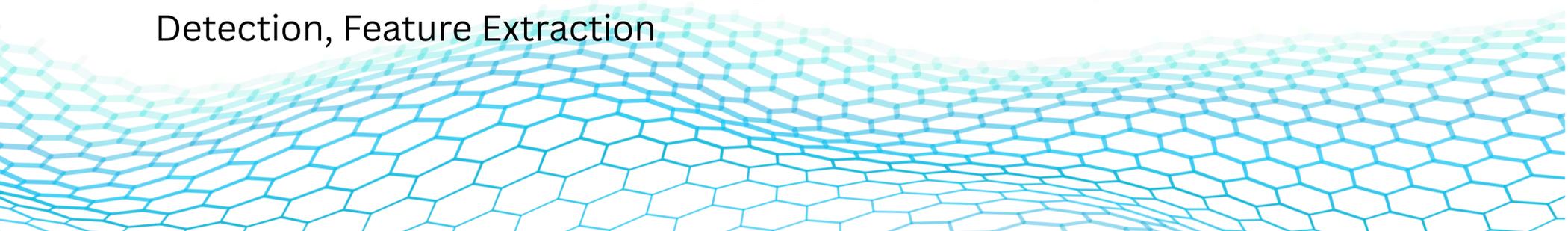
3. Unsupervised Learning Concepts

Understanding unlabeled data

Clustering: K-Means, Hierarchical

Dimensionality Reduction: PCA, t-SNE

Real-world use cases: Customer Segmentation, Anomaly Detection, Feature Extraction





Week 4 (Continued)

Machine Learning And Deep Learning

4. ML Data Handling

Train-test split and cross-validation
Feature scaling (Normalization & Standardization)
Encoding categorical variables
Feature selection techniques

5. Introduction to ML Libraries

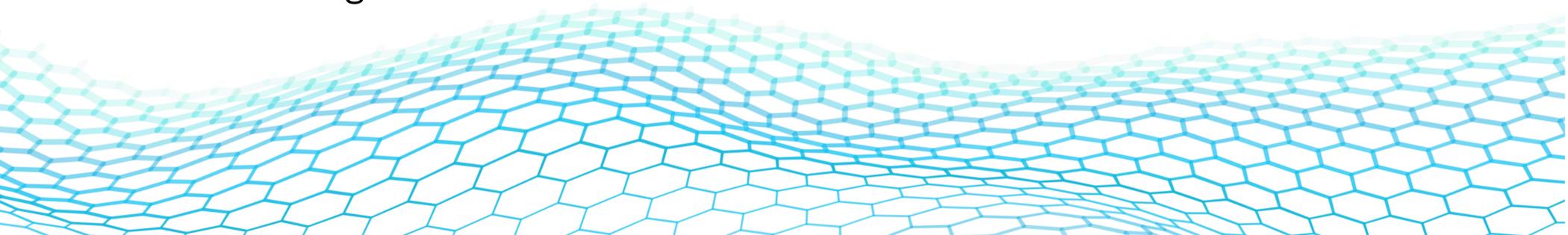
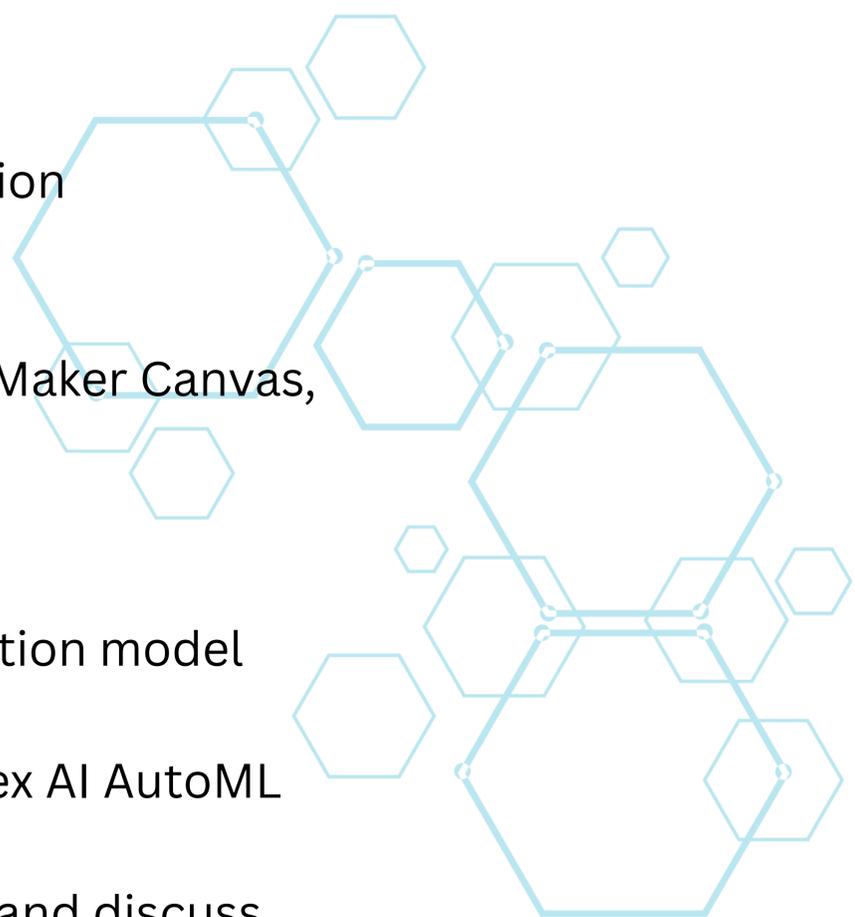
Scikit-Learn: model creation, training, evaluation
Matplotlib / Seaborn: visualize ML results
Joblib / Pickle: saving and loading models
For Non-Coders: AutoML on Azure, AWS SageMaker Canvas, or Google Vertex AI

Hands-On / Mini Projects

Coders: Build and evaluate a simple classification model (e.g., Iris dataset).
Non-Coders: Train a model using Google Vertex AI AutoML and interpret output charts.
Compare model predictions vs actual results and discuss accuracy gaps.

From Machine Learning to Deep Learning

Why traditional ML struggles with high-dimensional data
What is a Neural Network? Neurons, layers, activation functions
Terminology: epochs, batches, weights, bias, loss function
Forward and backpropagation explained simply Role of GPUs and TPUs in accelerating DL





Week 4 (Continued)

Machine Learning And Deep Learning

Deep Learning Architectures

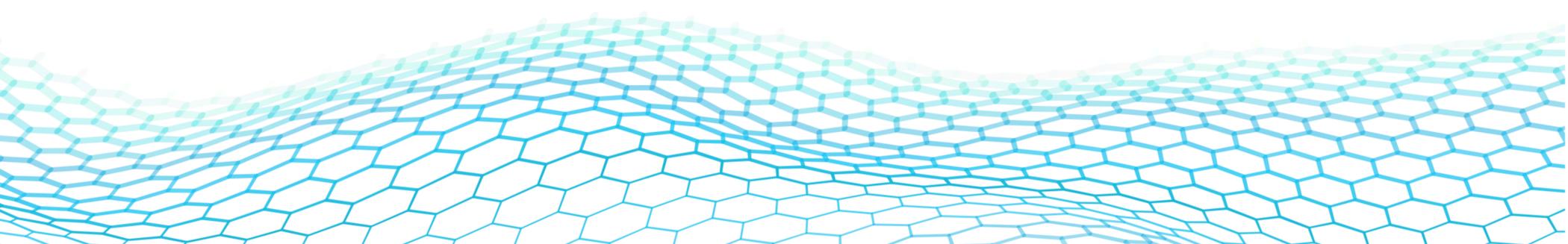
- Feedforward Neural Networks (FNNs) – foundation of DL
- Convolutional Neural Networks (CNNs) – for image and pattern recognition
- Recurrent Neural Networks (RNNs) – for sequence and timeseries data
- LSTM & GRU – memory-based improvements for long dependencies
- Transformers (Preview) – the architecture behind ChatGPT & Gemini

Optimization & Regularization

- Gradient Descent & variants (SGD, Adam, RMSprop)
- Loss functions: MSE, Cross-Entropy, MAE
- Overfitting prevention: Dropout, Early Stopping, Regularization
- Hyperparameter tuning (learning rate, batch size, epochs)

4. Key Frameworks & Libraries

- TensorFlow & Keras: high-level model building
- PyTorch: dynamic graphs and research use
- For Non-Coders: Azure Cognitive Services & Google Vertex AI
- Vision Tools for model creation without code





Week 4 (Continued)

Machine Learning And Deep Learning

Hands-On / Labs

For Coders:

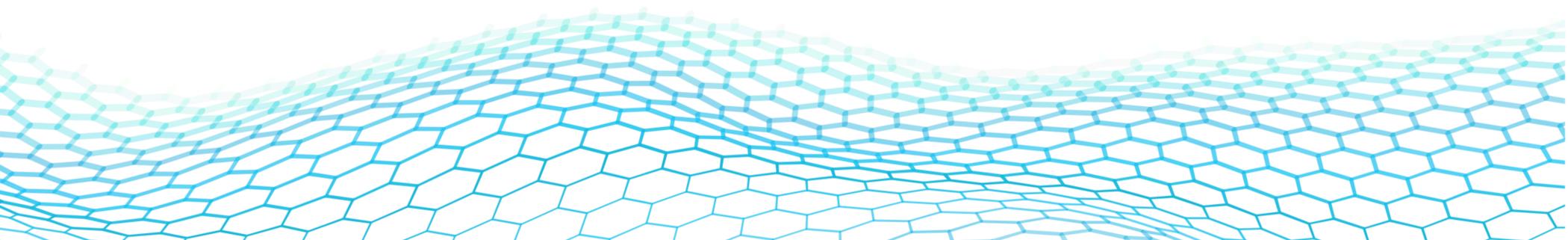
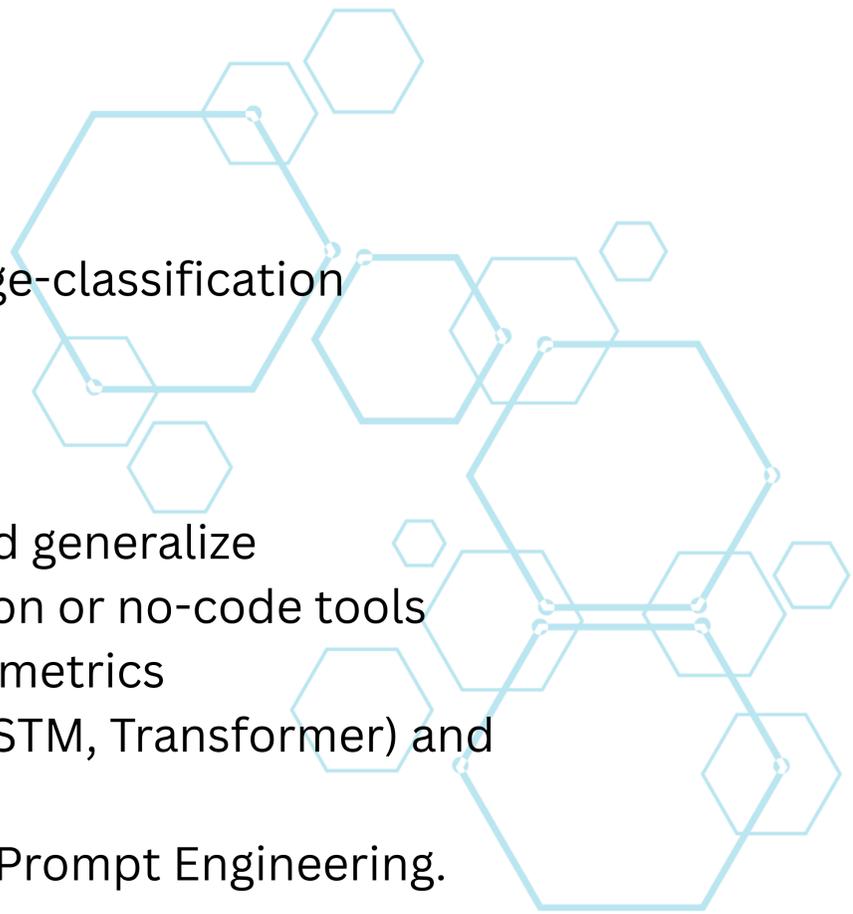
- Build a simple Feedforward NN for handwritten digit recognition (MNIST) using Keras / TensorFlow.
- Visualize training loss and accuracy graphs.

For Non-Coders:

- Use Google Vertex AI Vision to train an image-classification model.

Learning Outcomes

- Explain how deep neural networks learn and generalize
- Build and train basic DL models using Python or no-code tools
- Visualize and evaluate model performance metrics
- Recognize core architectures (CNN, RNN, LSTM, Transformer) and their use cases
- Prepare for advanced topics like LLMs and Prompt Engineering.





Week 5

Natural Language Processing (NLP):

Concepts, Algorithms & Project

1. Introduction to NLP

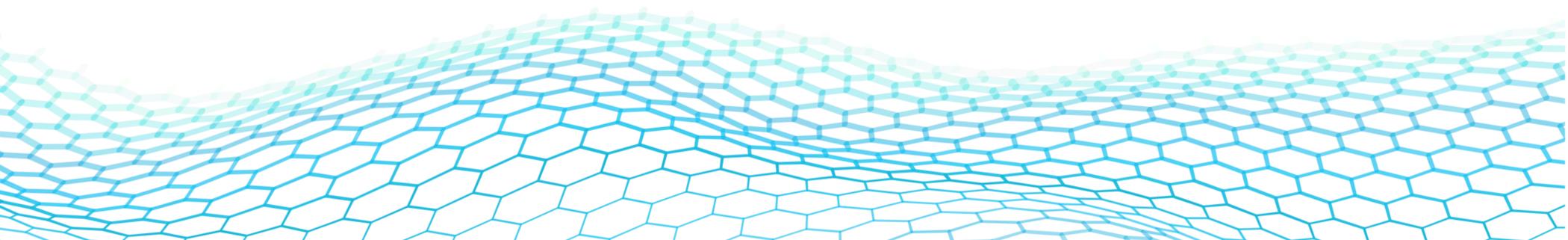
- What is NLP and why it matters in GenAI
- Key NLP tasks: tokenization, stemming, lemmatization, stopword removal
- Text preprocessing pipeline (cleaning → normalization → vectorization)
- Classical NLP vs Neural NLP vs LLM NLP

2. Text Representation Techniques

- Bag-of-Words (BoW) and TF-IDF
- Word embeddings: Word2Vec, GloVe, FastText
- Contextual embeddings: ELMo, BERT concepts
- Tokenization & subword tokenization (BPE, WordPiece)

3. Core NLP Algorithms and Models

- Naïve Bayes Classifier for text classification
- Logistic Regression / SVM for sentiment analysis
- Recurrent Neural Networks (RNNs) for sequence modeling
- LSTM / GRU for contextual text generation
- Transformers Overview (Self-Attention, Encoder-Decoder)
- BERT vs GPT: conceptual differences and use cases





Week 5 (Continued)

Natural Language Processing (NLP): Concepts, Algorithms & Project

4. Applied NLP with Python & APIs

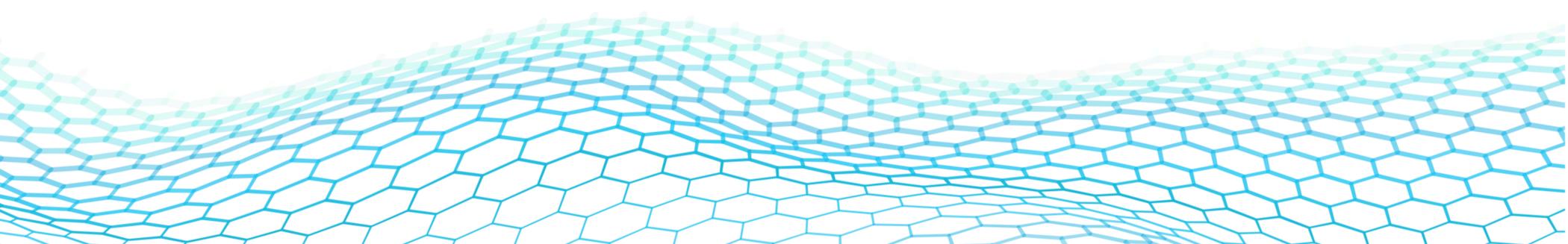
- NLTK and spaCy for text preprocessing and NER
- Hugging Face Transformers Library: load pre-trained models (BERT, DistilBERT, T5)
- OpenAI API / LangChain Intro: prompting and chat flows
- For Non-Coders: Google Vertex AI Text Model and Azure Language Studio

5. Mini Projects / Labs

- Coders: Build a Sentiment Analysis model using TF-IDF + Logistic Regression or fine-tune DistilBERT on custom data.
- Non-Coders: Use Vertex AI Natural Language API for entity and sentiment detection on text datasets.
- All: Create a “Text Summarizer Bot” using OpenAI API or Hugging Face pipeline.

3. Learning Outcomes

- Understand the core building blocks of Natural Language Processing
- Preprocess and represent text data for AI models
- Implement and evaluate classical and neural NLP algorithms
- Apply Transformers and LLMs for summarization, Q&A, and sentiment tasks





Week 6

Prompt Engineering & Applied Projects (With Code and Without Code)

Prompt Engineering Fundamentals

- What is a prompt? Role of context, clarity, and structure
- Anatomy of a Prompt: system message + user instruction + context data + examples
- Zero-shot, One-shot, and Few-shot prompting
- Chain-of-Thought (CoT) reasoning and stepwise explanation
- Output-format control (JSON, tables, markdown)
- Role-based prompting (act as a teacher, analyst, agent etc.)

Advanced Prompt Patterns & Techniques

- Tree-of-Thought and ReAct prompt flows
- Retrieval-Augmented Generation (RAG) concept refresher
- Context window management and token optimization
- Prompt evaluation metrics – relevance, coherence, diversity
- Using embeddings for semantic search and contextual answers

Prompt Engineering with Code

- Using OpenAI API with Python for custom LLM apps
- Basic LangChain project – prompt templates & chains
- Using Hugging Face Transformers pipeline for text generation
- Designing multi-prompt flows (conversational memory demo)
- Integrating LLM output with Flask / FastAPI web app



Week 6 (Continued)

Prompt Engineering & Applied Projects (With Code and Without Code)

Prompt Engineering Without Code

- Using ChatGPT and Gemini interfaces for multi-turn prompt flows
- Creating Notion AI / Gamma AI templates for content automation
- Designing Zapier / Make automations to trigger AI prompts
- Exploring Copilot in PowerPoint and Word for prompt-based content

Hands-On / Mini Projects

For Coders (Choose One):

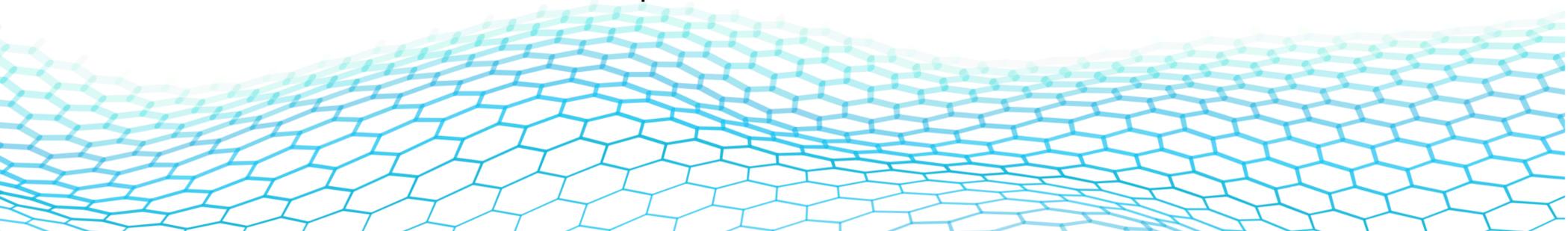
- AI Knowledge Assistant App – Connect OpenAI API with Python to answer PDF or website queries using RAG.
- Prompt Workflow Builder – LangChain multi-step prompt app with memory and user input.

For Non-Coders (Choose One):

- AI Business Proposal Generator – Build a template in Notion
- AI or ChatGPT Custom GPTs.
- AI Interview Coach – No-code prompt chain in Gemini or
- Gamma for mock Q&A simulation.

Tools & Platforms

- OpenAI API · LangChain · Hugging Face · Google Gemini ·
- ChatGPT · Notion AI · Zapier · Gamma AI · VS Code · Colab



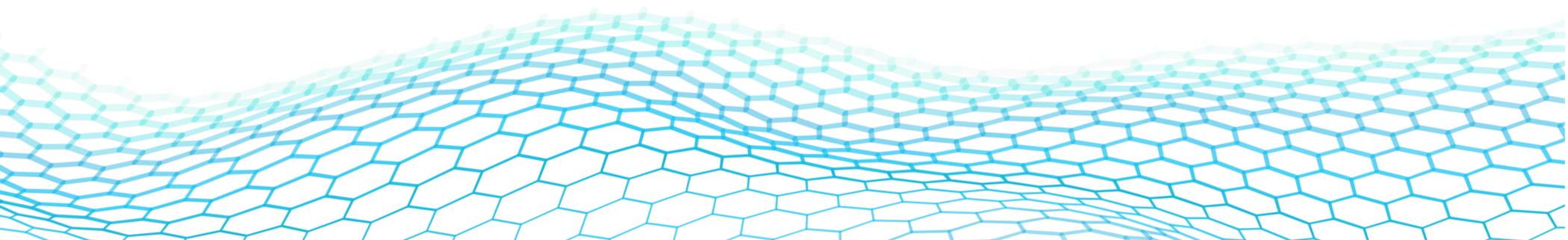


Week 6 (Continued)

Prompt Engineering & Applied Projects (With Code and Without Code)

Learning Outcomes

- Design and optimize prompts for different use cases
- Develop end-to-end prompt-based applications with and without code
- Understand LLM memory, context, and limitations
- Deploy a functional prompt-driven project that solves a real-world problem





Week 7

Large Language Models (LLMs) & GenAI Application Development

Understanding LLM Architecture

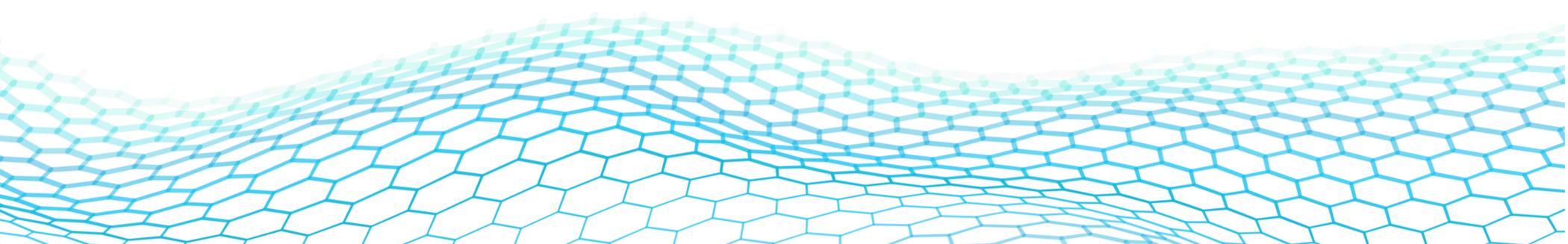
- How transformers work – attention, tokens, embeddings, context window
- Pre-training vs fine-tuning vs RAG vs Agentic flows
- Tokens = the currency of LLMs (why cost and context matter)
- Open-source vs closed-source LLMs (LLaMA, Mistral, Gemma vs GPT / Claude / Gemini)
- Model capabilities: text, image, audio, video, code

Working with LLM APIs and SDKs

- OpenAI API – chat completions, function-calling, JSON mode
- Google Gemini API – multimodal input (image + text) Anthropic Claude API – context and reasoning advantages
- Hugging Face Transformers – load and test open LLMs locally
- Authentication & rate limits (best practices for production apps)

Embeddings & Vector Databases

- What are embeddings and why they matter for RAG and context search
- Using OpenAI Embeddings API and LangChain vector stores
- Integrating FAISS / Pinecone / Chroma DB for semantic search
- Building a “context retriever” for chatbots and knowledge assistants





Week 7 (Continued)

Large Language Models (LLMs) & GenAI Application Development

GenAI App Development Concepts

- Designing end-to-end GenAI apps: Prompt → LLM → Memory
- → UI
- Agent frameworks overview: LangGraph, CrewAI, LlamaIndex
- Connecting LLMs to tools (e.g., browsing, Python, calculators, search)
- Security & Ethics – PII handling, model guardrails, response filters

Hands-On / Mini Projects

For Coders:

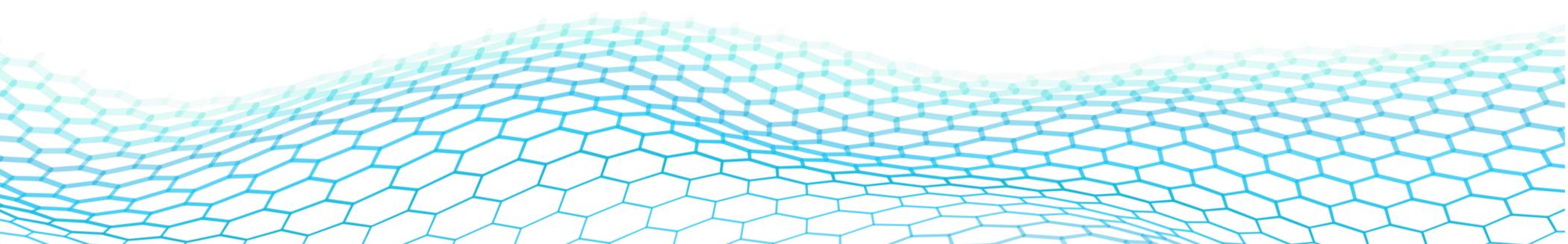
- AI Knowledge Assistant 2.0 – Use LangChain + OpenAI API +
- FAISS to create a document Q&A bot.
- AI Workflow Agent – Build a CrewAI agent that summarizes emails and creates tasks via API calls

For Non-Coders:

- Custom ChatGPT GPT – Design a GPT for business (use Custom GPT Builder interface).
- Gemini Business Advisor – Use Google Gemini Studio to build a prompt-driven assistant for marketing plans.

Tools & Platforms

- OpenAI API · Gemini API · Claude API · Hugging Face ·
- LangChain · LangGraph · CrewAI · FAISS · Chroma DB ·
- Pinecone · Streamlit · Flask · Google Colab · VS Code



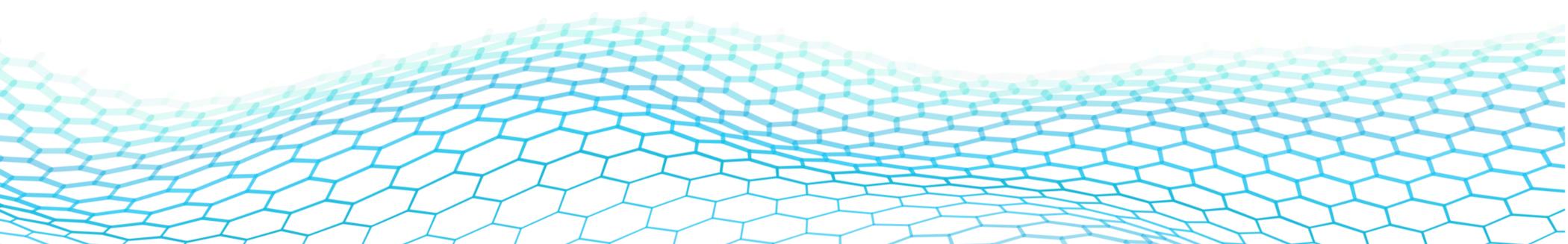
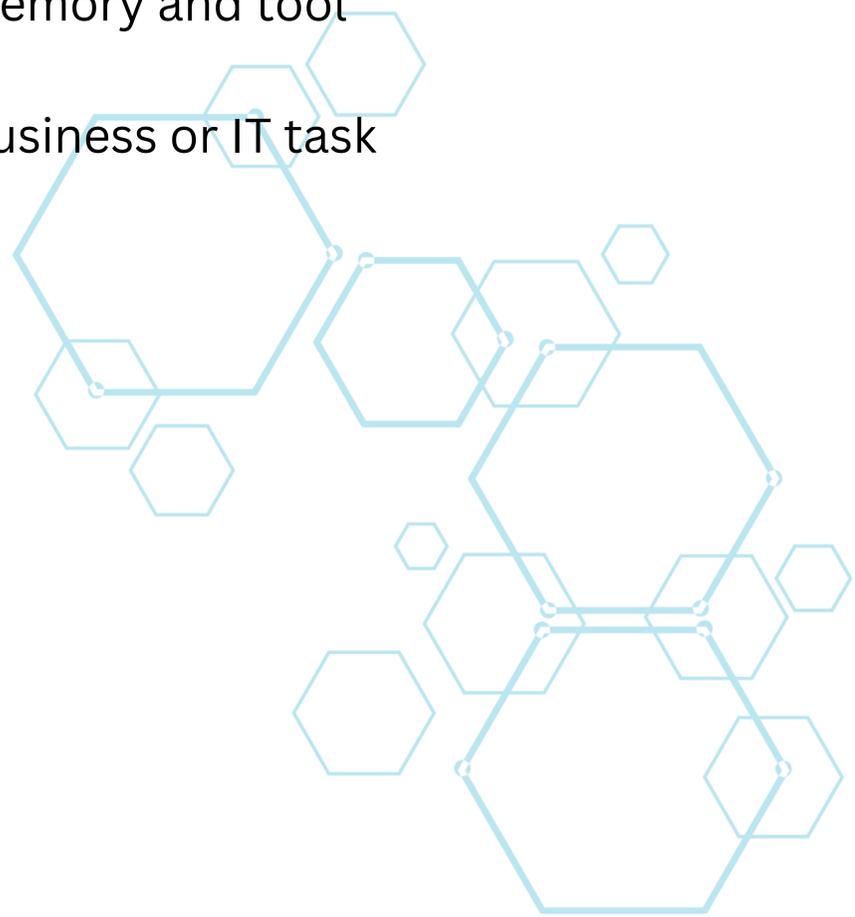


Week 7 (Continued)

Large Language Models (LLMs) & GenAI Application Development

Learning Outcomes

- Understand how LLMs are built and accessed via APIs
- Work with embeddings and vector databases to extend LLMs
- Develop LLM-based apps with context memory and tool usage
- Deploy functional GenAI assistants for business or IT task





Week 8

RAG Applications & AI Agent Frameworks

Understanding RAG (Retrieval-Augmented Generation)

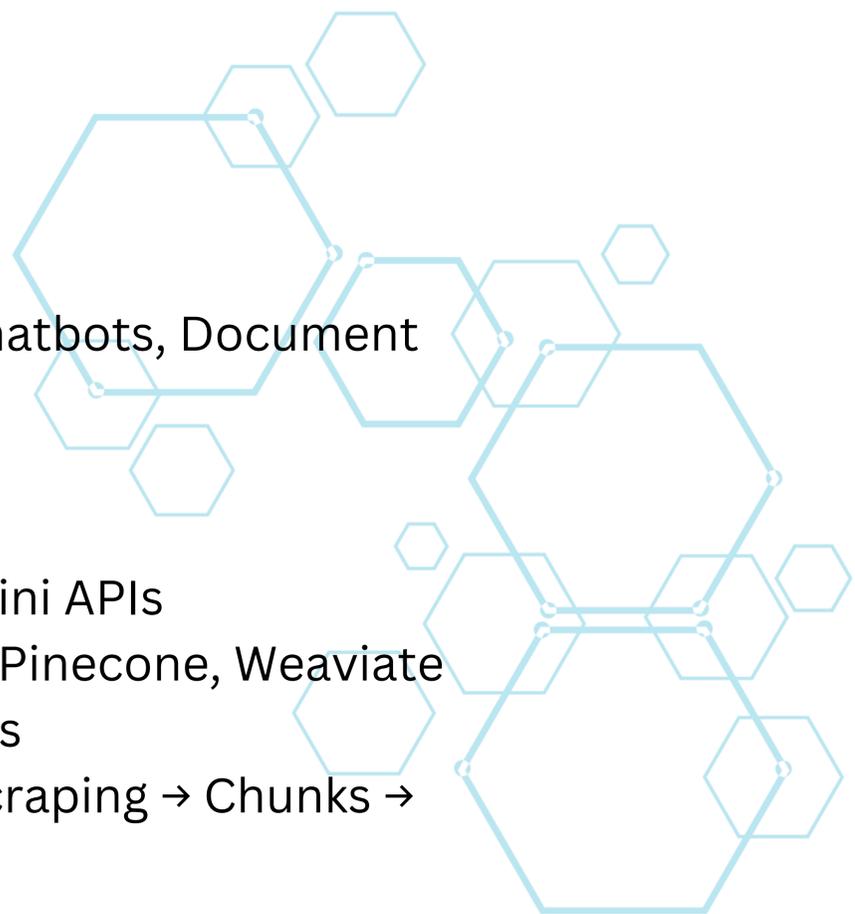
- The problem with LLMs: hallucination & outdated knowledge
- How RAG works: Retrieval + Augmented Context + Generation
- Components of a RAG pipeline:
 - Vectorization (Embeddings)
 - Indexing (Vector Database)
 - Query Retrieval & Re-Ranking
 - LLM Context Response
- RAG vs Fine-Tuning vs Context Injection
- Real-world RAG Use Cases: Enterprise Chatbots, Document Search, Knowledge Assistants

Building a RAG Pipeline

- Creating Embeddings with OpenAI / Gemini APIs
- Using Vector Databases: FAISS, Chroma, Pinecone, Weaviate
- Connecting LangChain Retrievers to LLMs
- Document parsing (PDF, CSV, Website Scraping → Chunks → Vector Store)
- Evaluating RAG accuracy and context relevance

Introduction to AI Agent Frameworks

- What are AI Agents? How they extend LLMs with goals and memory
- Agent architectures: Planner → Executor → Feedback Loop
- LangChain Agents, CrewAI multi-agent systems, AutoGen
- conversation loops
- Tool usage by Agents (e.g., web search, Python, APIs, file I/O)
- Long-term memory and context persistence with Vector Store





Week 8 (Continued)

RAG Applications & AI Agent Frameworks

Agent Workflows and Applications

- AI Research Agent – automatically summarizes and compiles research
- Business Analyst Agent – reads data & creates insights reports
- Personal Assistant Agent – email summarization + calendar planning
- RAG + Agents = self-improving intelligent system

Hands-On / Mini Projects

For Coders:

- Enterprise Document Chatbot – Build a LangChain + FAISS +
- OpenAI RAG chatbot that answers company document queries.
- CrewAI Task Bot – Create two agents (Planner & Executor) to summarize and prioritize emails or reports.

For Non-Coders:

- Custom Knowledge Bot – Use ChatGPT “Knowledge Files” or
- Gemini Memory to build a domain-specific assistant.
- RAG Workflow Automation – Design a Notion AI + Zapier
- workflow that retrieves info and summarizes it automatically.

Tools & Frameworks

- LangChain · LangGraph · CrewAI · AutoGen · FAISS · Chroma ·
- Pinecone · OpenAI Embeddings · Gemini API · Streamlit ·
- Google Colab · Zapier · Notion AI

Learning Outcomes

- Understand and implement RAG architectures to reduce LLM hallucinations
- Use vector databases for context-aware search and retrieval
- Design and deploy AI Agents that can plan, reason, and execute tasks
- Bridge LLM applications into real business automation scenarios



Week 9

Cloud Deployment & Scalability for AI and Agent Applications

Cloud Foundations for AI

- Cloud Computing Models: IaaS, PaaS, SaaS
- Benefits of cloud deployment for AI (scales compute & storage)
- Comparing Azure AI, AWS AI Services, and GCP Vertex AI
- GPU/TPU instances for training and inference

Deploying AI Models as APIs

- Packaging models with Flask / FastAPI
- REST vs GraphQL API structure
- Testing locally before deployment
- Setting up endpoints on Azure App Service, AWS Lambda, or GCP Cloud Run
- Managing environment variables and API keys securely

Containerization & Virtualization

- Introduction to Docker and Container images
- Writing Dockerfiles for AI apps
- Docker Compose for multi-service Agent systems
- Publishing to Azure Container Registry / Amazon ECR / Google Artifact Registry

Scaling and Monitoring

- Auto-scaling with Kubernetes (AKS, EKS, GKE)
- Load balancing and traffic management
- Performance metrics – latency, throughput, cost optimization
- Monitoring with Prometheus & Grafana
- Logging and alerting best practices





Week 9 (Continued)

Cloud Deployment & Scalability for AI and Agent Applications

Integrating Cloud AI Services

- Azure Cognitive Services (text, vision, speech)
- AWS SageMaker Endpoints for deployment
- GCP Vertex AI Pipelines for continuous training & serving
- Using AutoML and Cloud Storage Buckets for model data

Hands-On / Mini Projects

For Coders:

- Deploy Your RAG Bot – Containerize and deploy a LangChain + FAISS chatbot on AWS or GCP.
- AI API Gateway – Create a FastAPI endpoint that wraps OpenAI or Gemini API and host it on Azure.

For Non-Coders:

- No-Code Model Hosting – Deploy a trained model using Vertex AI or Azure Machine Learning Studio.
- Power BI Cloud Dashboard – Connect Power BI to a live AI API for dynamic insight updates.

Tools & Platforms

- Azure Cognitive Services · AWS SageMaker · GCP Vertex AI ·
- Flask · FastAPI · Docker · Kubernetes · Prometheus · Grafana ·
- Power BI · Google Colab · VS Code

Learning Outcomes

- Deploy AI and Agent applications as scalable cloud services
- Containerize and host LLM/RAG apps securely
- Monitor performance and cost efficiency
- Understand CI/CD and production best practices for AI workloads



Week 10

Model Context Protocols (MCPs) & Agent-to-Agent (A2A) Communication

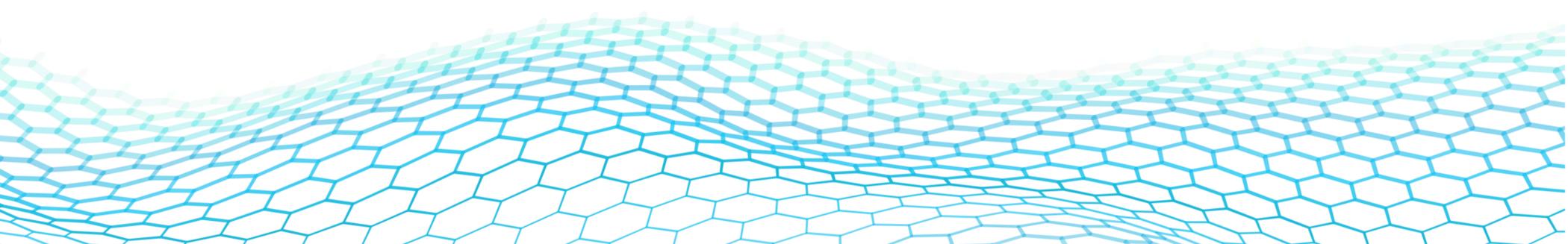
Understanding Model Context Protocol (MCP)

- What is MCP and why it matters in Agentic AI
- Role of context passing between models and tools
- How MCP standardizes communication across agents (OpenAI ↔ Anthropic ↔ Gemini ↔ Local LLMs)
- Structure of an MCP transaction: request, metadata, context, response
- Integration of MCPs in frameworks like LangChain, CrewAI, AutoGen, OpenDevin

A2A (Agent-to-Agent) Communication

Concepts

- What is A2A and how multi-agent systems collaborate
- Planner–Executor–Reviewer pattern for cooperative agents
- Message-passing protocols and memory synchronization
- Negotiation and role-based coordination among agents
- Conflict resolution and feedback loops between agents
- Use Cases:
 - AI Research Teams (Planner Agent + Coder Agent + Reviewer Agent)
 - Customer Service AI (Query Bot ↔ Escalation Bot ↔ Email Bot)
 - Workflow Orchestration (Report Generator ↔ Data Retriever ↔ Validator)





Week 10 (Continued)

Model Context Protocols (MCPs) & Agent-to-Agent (A2A) Communication

Implementing MCP and A2A Frameworks

- Setting up CrewAI for multi-agent collaboration
- Building AutoGen conversational agents that talk to each other
- Defining shared context & task memory between agents
- Multi-Agent communication via LangGraph & OpenDevin
- API bridging: how one LLM triggers another model through MCP

Security, Logging, and Governance

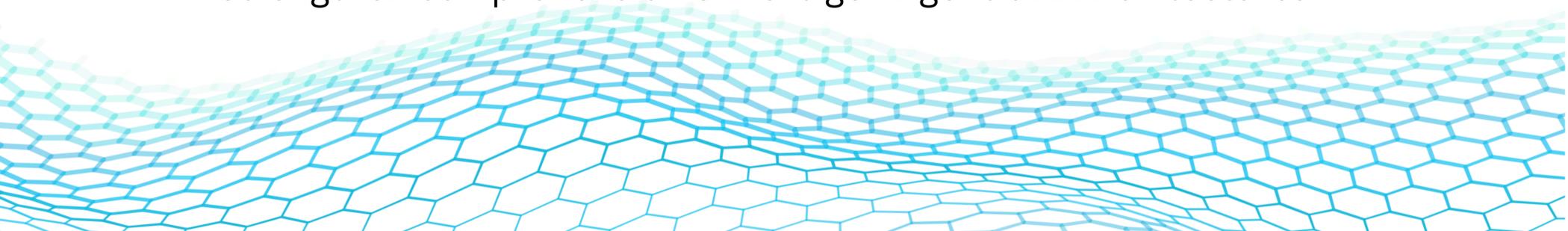
- Handling authentication and authorization between agents
- Preventing context leakage and prompt injection
- Role of observability (logs, metrics, traces) in multi-agent systems
- Governance models for collaborative AI (ethical decisionmaking, auditing)

Tools & Frameworks

CrewAI · AutoGen · LangGraph · OpenDevin · LangChain MCP · Gemini API · OpenAI API · Hugging Face · Zapier · Notion AI

Learning Outcomes

- Understand how MCP standardizes model-to-model and agent communication
- Build multi-agent, multi-model systems that collaborate and share memory
- Apply A2A protocols for real-world autonomous workflows
- Strengthen comprehension of next-gen Agentic AI Architectures





Week 11

Agentic Systems, Orchestration & Deployment (Capstone Phase 2)

Understanding Agentic Systems

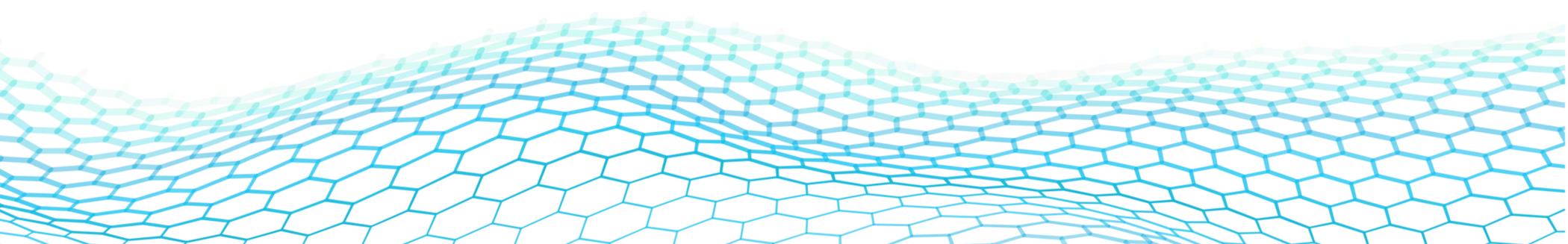
- What makes an AI system Agentic (autonomous, goal-driven, context-aware)
- Roles and architecture of an agentic system:
 - Planner Agent – defines tasks and priorities
 - Executor Agent – performs subtasks and APIs calls
 - Reviewer Agent – validates and refines outputs
- Memory and Context Sharing between agents
- Short-term vs Long-term memory in Agentic design

Orchestration and Multi-Agent Workflows

- Task orchestration frameworks: LangGraph, CrewAI, AutoGen, OpenDevin
- Sequential vs Parallel task management between agents
- Role of Vector Stores in shared context memory
- Communication protocols and event-driven architecture
- Coordinating agents for data → insight → action pipelines
- Logging, tracing, and version control of multi-agent processes

Integration & Deployment

- Integrating LLM + RAG + MCP + Cloud Services into one workflow
- Exposing the orchestrated agents as an API endpoint or web interface
- Using Docker Compose / Kubernetes for distributed orchestration
- Storing task logs in databases or NoSQL stores (MongoDB, Firestore)
- Continuous integration and monitoring for production-grade systems





Week 11 (Continued)

Agentic Systems, Orchestration & Deployment (Capstone Phase 2)

Capstone Phase 2: Team Project Design

- Data ingestion + preprocessing
- RAG knowledge base
- LLM-based reasoning agents
- Agent-to-Agent communication (MCP/A2A)
- Cloud deployment pipeline

Hands-On / Mini Projects

For Coders:

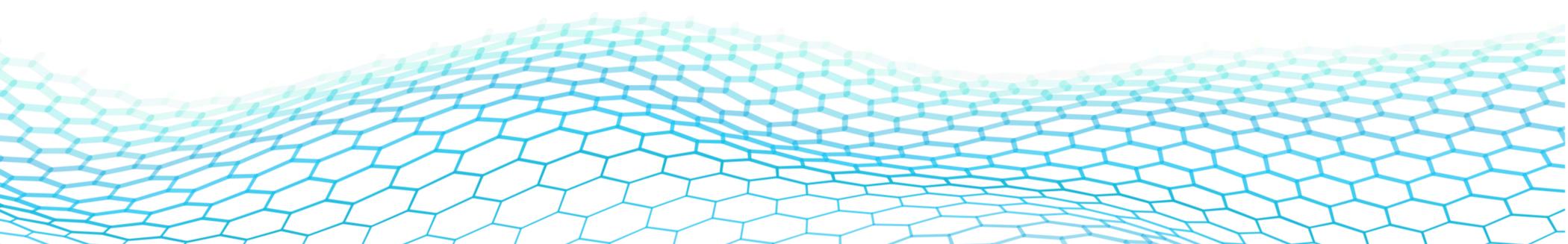
- Multi-Agent Research Assistant – Planner Agent (topic) → Retriever Agent (docs) → Writer Agent (summary) → Reviewer Agent (refinement).
- Agentic Workflow on Cloud – Deploy a CrewAI + LangGraphbased system on AWS or GCP that automates market research or project planning.

For Non-Coders:

- AI Project Coordinator – Create an orchestrated automation using ChatGPT GPTs + Notion AI + Zapier + Power Automate that coordinates reports and updates.
- Cloud Hosted No-Code Agent Flow – Build and publish a multi-tool workflow using Google Vertex AI or Azure AI Builder.

Learning Outcomes

- Understand end-to-end design of multi-agent, clouddeployed systems
- Orchestrate AI workflows using MCP + A2A communication
- Deploy and monitor multi-agent pipelines for business/tech automation
- Complete the Phase 2 of Capstone Project: Agentic AI Deployment





Week 12

No-Code & Low-Code AI Tools and Automation Platforms

Introduction to No-Code / Low-Code AI Ecosystem

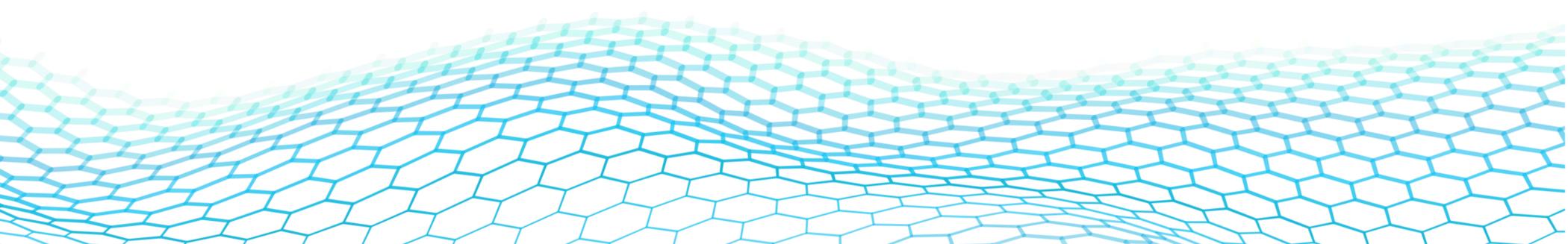
- What are No-Code & Low-Code platforms?
- When to choose no-code over Python-based development
- Understanding APIs, triggers, and connectors
- How no-code tools interact with AI models (LLMs, image, audio, vision APIs)
- Integrating AI services like OpenAI, Gemini, Hugging Face, and Azure AI

Workflow Automation Tools

- n8n: visual node-based automation for multi-step workflows
 - Connecting APIs, LLMs, and databases
 - Running GenAI automations using OpenAI & Gemini nodes
- Zapier & Make (Integromat): automate tasks across apps (Gmail → Sheets → ChatGPT)
- ClickUp Automation: AI-powered task creation, team tracking, and content suggestions
- Notion AI: smart notes, auto-summary, and knowledge dashboards

Building Practical AI Systems Without Code

- Connecting ChatGPT or Gemini to workflow builders (n8n / Zapier)
- Auto-generating reports, summaries, and social posts using Notion + GPT
- Creating an AI-powered CRM or Task Manager using ClickUp + AI automations
- Integrating image or video generation tools (DALL-E, Pika, Runway ML)
- Multi-tool orchestration: chaining AI tools with triggers (e.g., upload → analyze → email → report)





Week 12 (Continued)

No-Code & Low-Code AI Tools and Automation Platforms

Emerging No-Code AI Platforms

- Bolt.new – real-time app builder for AI assistants
- Flowise AI – drag-and-drop LangChain UI for AI pipelines
- Durable AI / Builder.ai – create landing pages and small apps with AI
- Typedream + Softr – building simple frontends for AI tools
- Gamma AI / Canva AI – content and presentation automation

Hands-On / Mini Projects

For Coders:

- Hybrid Automation Project – Use n8n + OpenAI API to automate email classification and response generation.
- ClickUp Task Manager with AI – Build an automated project tracker that generates task summaries and insights using AI.

For Non-Coders:

- No-Code AI Business Assistant – Design a Notion AI or Bolt.new workspace that summarizes daily goals, meetings, and emails.
- Social Media Automation – Use Zapier + ChatGPT to autogenerate and post marketing captions on LinkedIn or Instagram.

Tools & Platforms

- n8n · Bolt.new · ClickUp · Notion AI · Zapier · Make (Integromat) Flowise AI · Gamma AI · Canva AI · Durable AI · Builder.ai OpenAI API · Gemini API

Learning Outcomes

- Understand and apply no-code automation principles
- Build AI-driven workflows using drag-and-drop logic
- Integrate multiple tools to automate repetitive tasks
- Design full AI-powered business apps without writing code Prepare for final Capstone Presentation (Week 14) showcasing
- no-code + coded systems



Week 13

Capstone Presentations, Portfolio & Career Path in AI

Capstone Project Finalization

- Final polishing of projects started in Week 12
- Verifying data pipeline → model → deployment → user interface
- Incorporating AI Agents, MCPs, and No-Code tools into final demos
- Integrating RAG or multi-agent features for intelligent workflows
- Writing technical documentation and user guides

Project Presentation & Peer Review

- Present your AI solution to the class, mentors, or industry evaluators
- Each team demonstrates:
 - Problem statement and objective
 - Tools and technologies used
 - Live demo or video walkthrough
 - Business use case and potential scalability
- Peer & instructor feedback sessions
- Evaluation criteria: innovation, execution, presentation clarity, teamwork

Portfolio Development

- Building a strong AI Project Portfolio (for GitHub & LinkedIn):
 - Uploading code-based projects (Python / LLM Apps / RAG Bots)
 - Showcasing no-code automations (n8n, Bolt, Notion, ClickUp)
 - Documenting capstone projects with README, diagrams, screenshots
- Creating a personal AI Resume Template highlighting:
 - Core Skills: Python, LLMs, RAG, NLP, DL, Cloud AI, Prompt Engineering
 - Tools: OpenAI API, LangChain, Vertex AI, CrewAI, n8n, Notion AI
- Soft Skills: Problem Solving, Critical Thinking, Presentation Skills





Week 13 (Continued)

Capstone Presentations, Portfolio & Career Path in AI

Career Guidance & Future Pathways

- Understanding AI job roles:

For Coders:

- AI Developer, Data Scientist, Prompt Engineer,
- MLOps Engineer, Agent Framework Specialist

For Non-Coders:

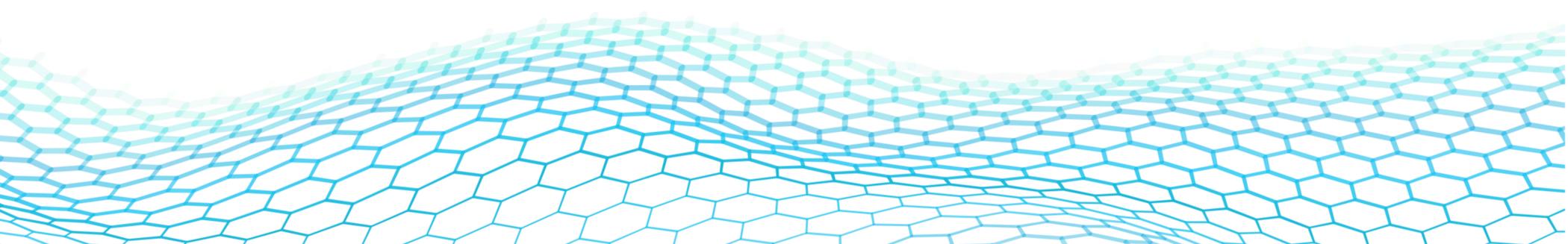
- AI Analyst, AI Project Manager, No-Code
- Automation Specialist, AI Product Designer
- Resume Optimization with AI Tools (Rezi, ChatGPT Resume Builder, Teal)
- LinkedIn Optimization Workshop: writing impactful AI project posts
- Interview preparation for AI roles (technical & behavioral) Pathways to advanced learning: LangGraph, CrewAI, MCP
- SDKs, and cloud certifications (Azure AI Engineer, Google Vertex AI Specialist)

Final Showcases / Activities

- Capstone Showcase Day
 - Live or recorded demo presentations
 - Jury panel evaluation (internal or external mentors)
 - Feedback and Q&A
- “Build Your AI Brand” Activity

Each student builds their personal brand post summarizing their AI journey on LinkedIn with project links & visuals.

- AI Career Launch Roadmap
- Personalized action plan for each student:
 - Resume completion, Portfolio submission
 - LinkedIn optimization, Next project or internship milestone





Week 13 (Continued)

Capstone Presentations, Portfolio & Career Path in AI

Tools & Platforms

- GitHub · LinkedIn · Canva · Notion AI · Bolt.new · n8n ·
- PowerPoint · ChatGPT Resume Builder · Rezi · Google Sites (Portfolio)

Learning Outcomes

- Demonstrate full AI project lifecycle mastery (Data → Model → Agents → No-Code → Cloud)
- Build and publish professional AI portfolios and resumes
- Present projects confidently to mentors, companies, or online communities
- Identify career paths and next-level certifications in AI and GenAI

Week 11A – AI Copilots & Intelligent Productivity Tools

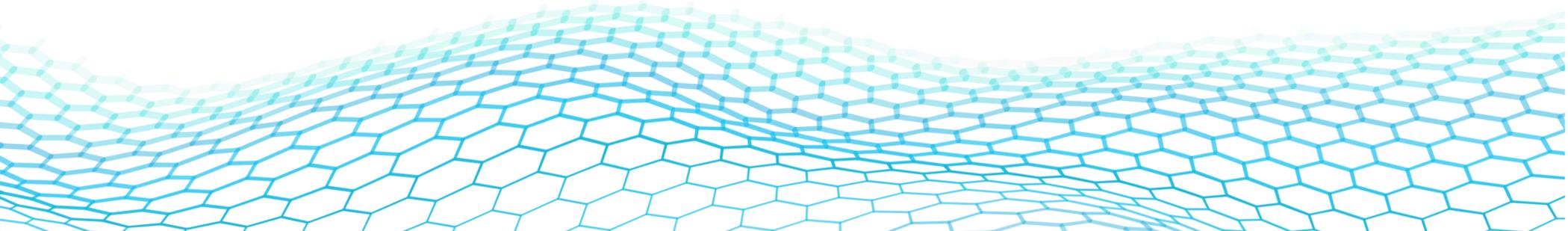
Introduction to AI Copilots

- What is a Copilot? – definition, purpose, and architecture
- How Copilots differ from chatbots or LLM interfaces
- The “AI as a Teammate” concept in software and business
- Integration of Copilots into IDEs, apps, and cloud ecosystems

GitHub Copilot (Developer Productivity)

- Using Copilot in VS Code & JetBrains IDEs
- Auto-completion, docstring generation, and debugging
- Context awareness and inline suggestions
- Building entire functions and classes with minimal typing
- Ethical coding and code privacy considerations

○



Week 13 (Continued)



Week 11A – AI Copilots & Intelligent Productivity Tools

- AI in Excel – formula generation, summarization, trend insights
- AI in PowerPoint – auto-presentation builder from text
- Word Copilot – smart content drafting and rewriting
- Outlook Copilot – email summarization and response automation
- Business automation examples integrating Power Automate + Copilot

Cursor.ai & Codex (Next-Gen Developer Copilots)

- Cursor.ai IDE: full AI-powered coding environment
 - Natural language to code
 - Real-time bug explanations and code refactoring
 - Codebase Q&A and instant documentation
- OpenAI Codex: The foundation behind Copilot
 - Understanding Codex models and use cases
 - Using Codex API for AI coding assistants and IDE extensions

Emerging & Specialized Copilot Tools

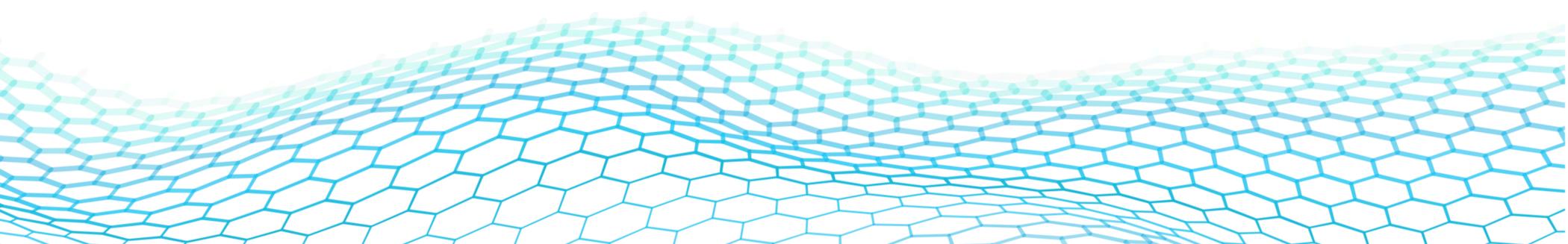
- Amazon CodeWhisperer: AI-powered AWS coding
- Google Duet AI: coding assistance in Google Cloud & Colab
- Replit Ghostwriter: AI for full-stack web apps
- Tabnine: team-based AI coding assistant
- Notion AI / Canva Magic Write: creative copilots for design and documentation

Tools & Platforms

- GitHub Copilot · Microsoft 365 Copilot · Cursor.ai · OpenAI
- Codex · Amazon CodeWhisperer · Google Duet AI · Replit
- Ghostwriter · Tabnine · Notion AI · Canva AI

Learning Outcomes

- Use AI copilots for faster coding, debugging, and documentation
- Automate routine business and creative tasks using integrated AI tools
- Understand how Codex models power coding copilots
- Evaluate and choose the right AI Copilot for specific workflows





Course Concludes

This comprehensive 13-week training caters to both newcomers and upskilling participants, ensuring everyone gains relevant, industry-ready skills and confidence to excel.

**For more details or enrollment,
contact us at:**

Phone: **+91-9866693295/ 9490449316**

Email: **info@advaitaits.com**

Website: **www.advaitaits.com**

