Generative AI Crash Course Agenda Agenda

Python Essentials

Core libraries for AI development

NLP Fundamentals

Tokenization, embeddings, transformers

Generative AI Models

Architecture, training, inference techniques

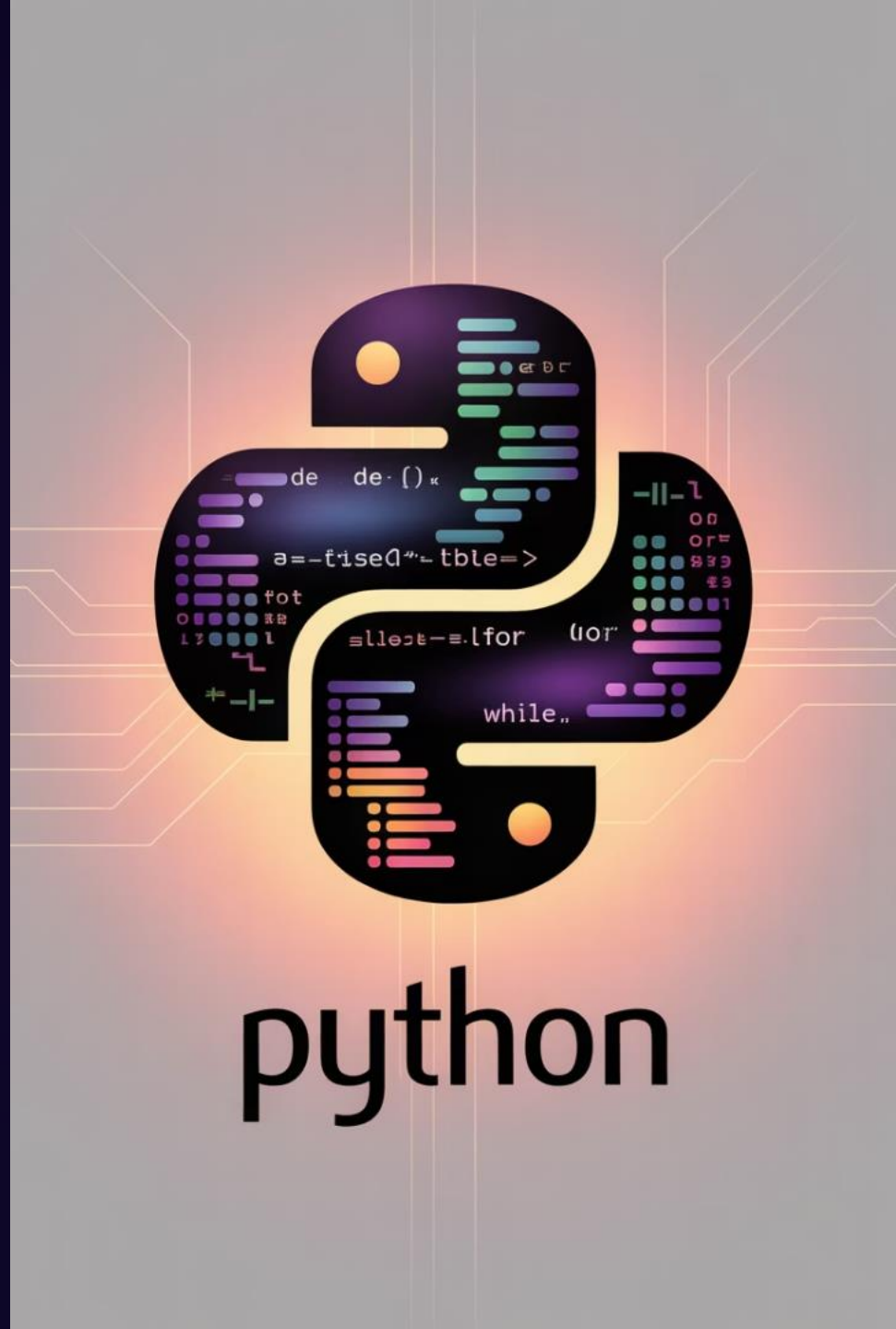
LLMOps

Deployment, monitoring, evaluation frameworks

LS

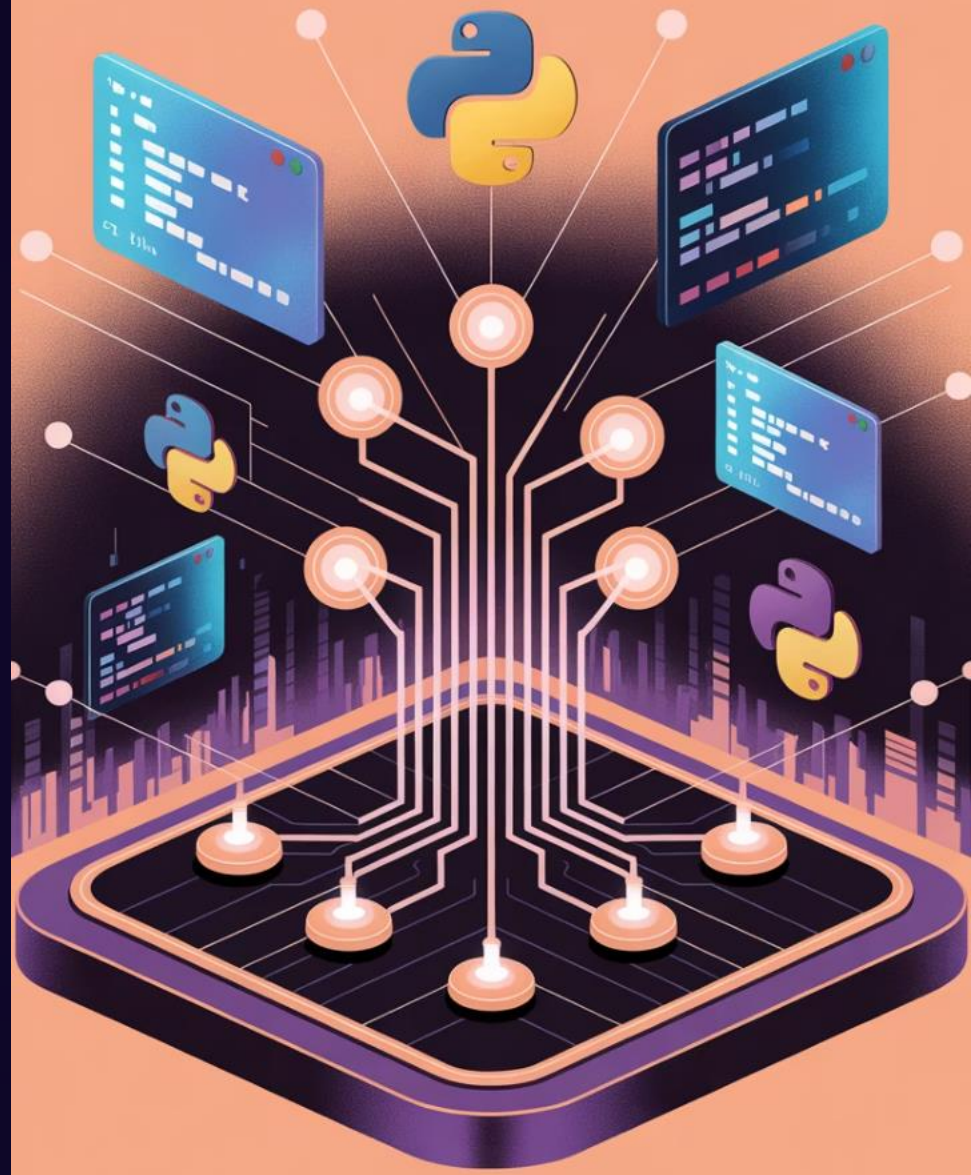
Python

1. What is Python
2. Installation of Python – VS Code, PyCharm, Anaconda, Google Colab
3. 100 Python main function names
4. Integer, float, Boolean operations (like math calculations), arithmetic (+-*/===''==''), logical operators (and, or, xor)
5. What is variable, types, about print and format combinational operations, basics of strings
6. Strings and methods (user input)
7. List and tuple
8. Sets, Dictionaries (enumerate, zip)
9. Conditionals
    1. if, elif, else
10. Match, case
11. For loops, for–else, nested loops
12. Break, continue, pass, list comprehension, dictionary comprehension
13. While loops, while-else
14. Comments, docstrings, about functions and modules
15. Types of functions, scope of function working and creating functions
    1. Parameters, arguments, *args, **kwargs
    2. Global variable, local variable
16. Creating module, import module help(), dir() aliasing, renaming



Python Topics

1. Lambda function, map, reduce, filter
2. Iterator, generator, decorator
3. Modules & packages
    - Math
    - Random
    - Date time
    - OS
    - Sys
    - Re module
    - if __name__ == "__main__" in Python
4. File handling
5. Logging
6. Error and exceptions handling
7. What is object, class, __init__ method, instance variables
8. Types of methods in Python
    - Instance method
    - Class method
    - Static method
9. Public, private, protected members and methods
10. Inheritance and Types of Inheritance
11. Polymorphism, encapsulation, abstract method



Python Troubleshooting

1. Errors in Python
2. How to search on Google to find error corrections using Google, Stack Overflow, and documentation

UI Frameworks

1. Flask / Fast API / Streamlit

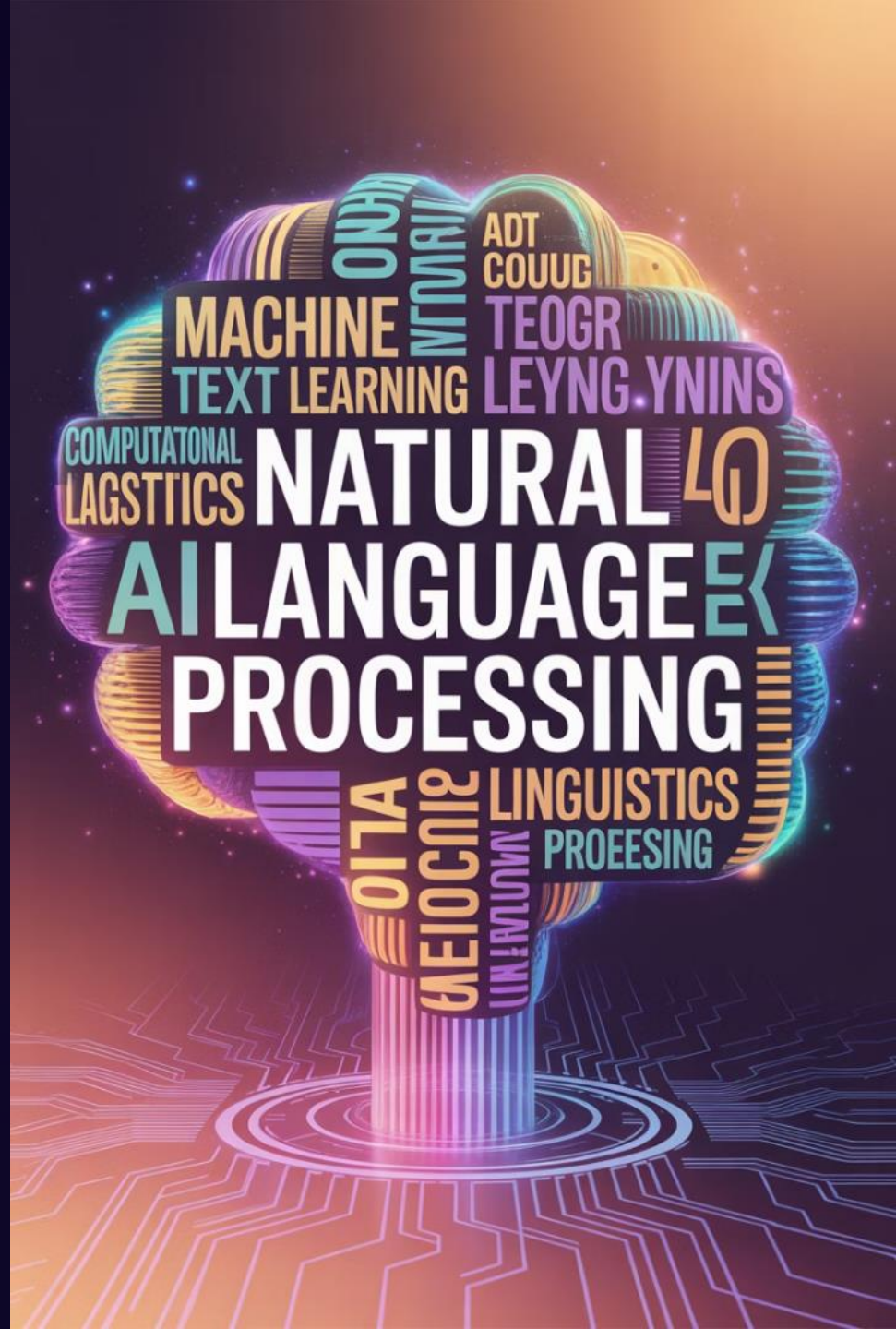
Projects

1. Calculator
2. Snake game

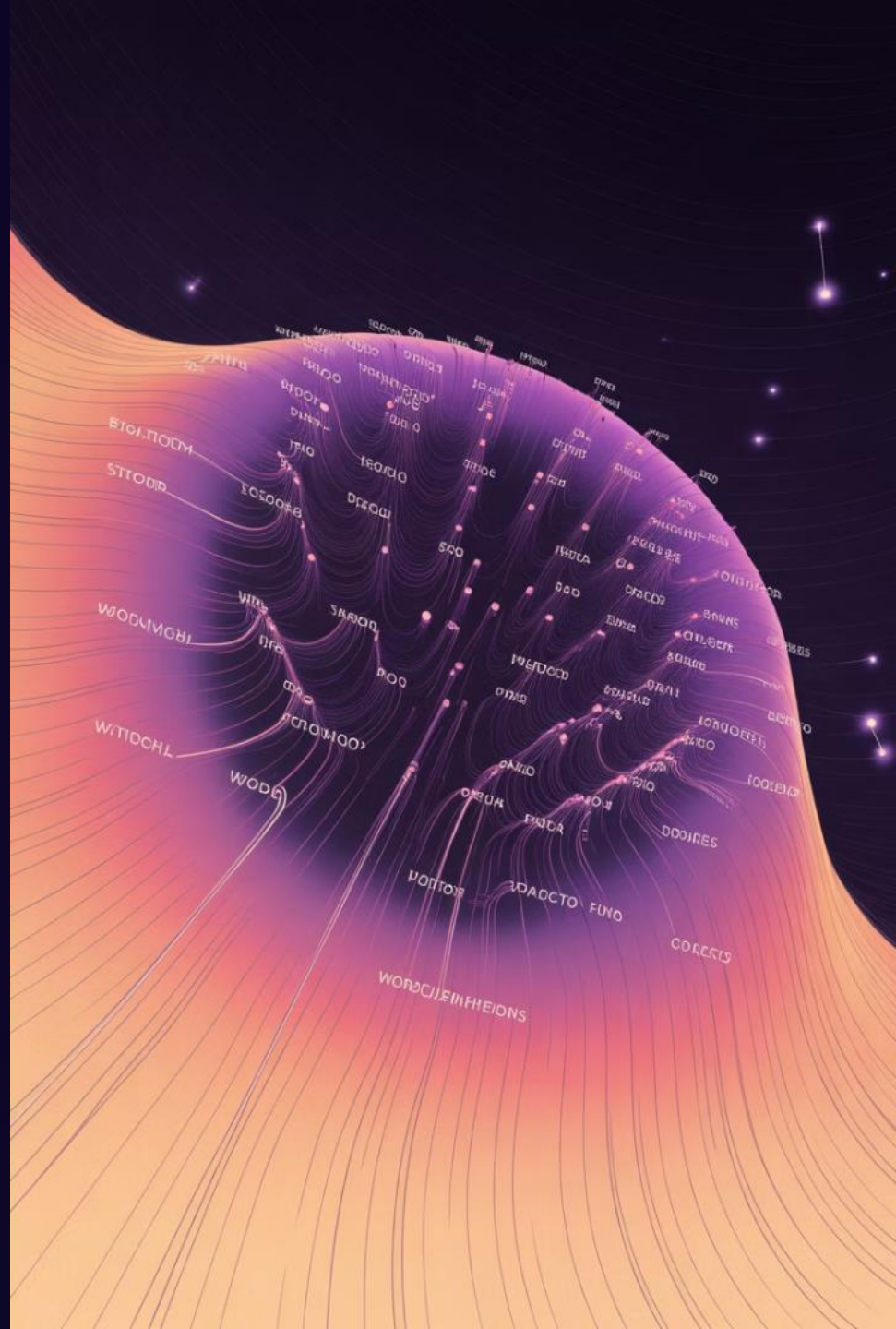
NLP

Module 1: Foundations of NLP

- Introduction to NLP
  - Understanding the scope and applications of NLP. Key terminologies: words, sentences, documents, corpus.
- Text Preprocessing Techniques
  - Segmentation and Tokenization.
  - Text normalization: case conversion, spell correction, one gram, bi grams, n-grams.
  - Cleaning text: removing stop words, punctuations, and white spaces.
  - Stemming and Lemmatization.
  - Part-of-Speech (POS) Tagging.
  - Rephrasing text for clarity.



- Word Embeddings
  - One-hot encoding.
  - Bag of Words (BoW).
  - TF-IDF.
  - Word2Vec.
  - GloVe.



NLP Libraries Overview

- NLTK.
- SpaCy.
- Gensim.
- fastText.
- Stanford NLP Toolkit.

Module 2: Sequential Models in NLP

Introduction to Sequential Models

- Understanding the need for sequential models in NLP.

Recurrent Neural Networks (RNNs)

- Architecture and working.
- Challenges: vanishing and exploding gradients.

Long Short-Term Memory (LSTM) Networks

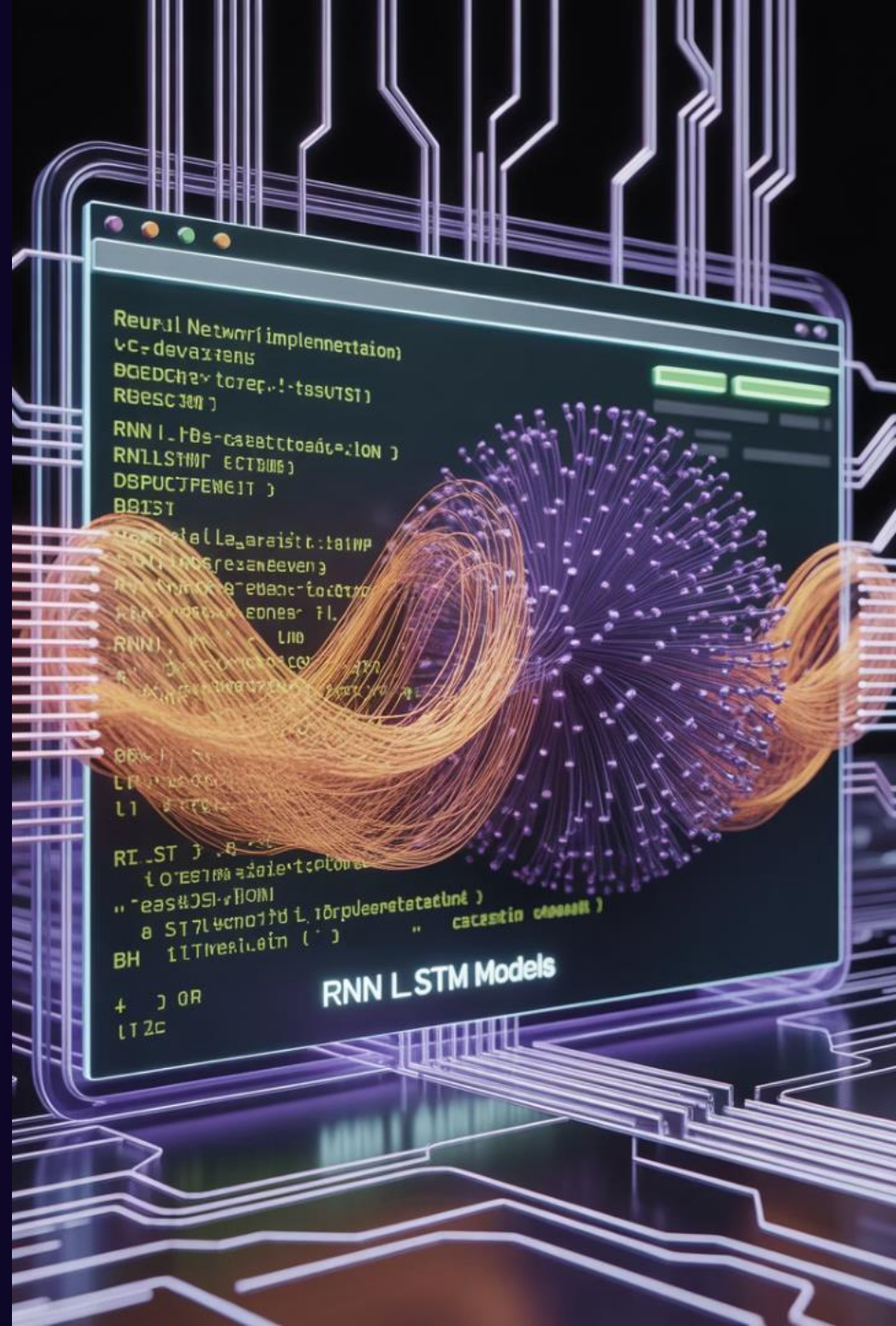
- LSTM architecture and gates.
- Advantages over traditional RNNs.

Gated Recurrent Units (GRUs)

- GRU architecture.
- Comparison with LSTMs.

Hands-on Implementations

- Building RNN, LSTM, and GRU models using TensorFlow/Keras.
- Projects:
  - Sentiment analysis on the IMDB dataset.
  - Text generation tasks.
  - Named Entity Recognition (NER).



Module 3: Advanced NLP Applications Applications

Web Scraping and Data Collection

- Techniques for extracting textual data from websites.

Text Visualization

- Creating Word Clouds.
- Exploratory Data Analysis (EDA) for text data.

Text Similarity Measures

- Cosine Similarity.
- Jaccard Similarity.

Building NLP Applications

- Developing a spam classifier.
- Creating a basic chatbot using rule-based approaches.

Projects

1. Text Summarization for News Articles

- **Objective**: Generate concise summaries of lengthy news articles.
- **Techniques**: Extractive summarization using TextRank.
- **Tools**: Gensim, spaCy.
- **Dataset**: CNN/Daily Mail Dataset.Guvi+5Fynd Academy+5The Knowledge Academy+5

2. Chatbot for Customer Service

- **Objective**: Develop a chatbot to handle customer inquiries.
- **Techniques**: Rule-based responses, intent classification.
- **Tools**: Rasa, Dialogflow.
- **Dataset**: Custom intents and responses.ProjectPro+2Fynd Academy+2Guvi+2

3. Language Translation System

- **Objective**: Translate text from one language to another.
- **Techniques**: Sequence-to-sequence models with attention mechanisms.
- **Tools**: TensorFlow, Keras.
- **Dataset**: Multi30k Dataset.

Generative AI

Module 1: Foundations of AI & Generative Models

1. Introduction to AI
2. AI vs ML vs DL
3. Types of Learning
    1. Supervised
    2. Unsupervised
    3. Reinforcement
4. Core Difference between ML and DL

Module 2: Introduction to Natural Language Processing (NLP)

1. History of NLP
2. Intro to RNN, LSTM, GRU, BERT
3. Problems with RNN, LSTM, GRU
4. Shift from RNNs to Transformers

Module 3: Understanding Generative AI

1. What is Generative AI?
2. Why are Generative Models Required?
3. Understanding Generative Models and Their Significance
4. Generative AI vs Discriminative Models
5. Recent Advancements and Research in Generative AI

L&L AI Solutions

Module 4: Transformers – The Foundation of Modern Generative AI

1. In-Depth Intuition of the Transformer Architecture *(Attention is All You Need Paper)*
2. Transformer Variants:

Encoder-only (e.g., BERT)

BERT Models- Google

1. BERT(Bidirectional Encoder Representations from Transformers)
2. RoBERTa (Robustly Optimized BERT Approach)
3. DistilBERT
4. ALBERT
5. XLNet

ii)Decoder-only (e.g., GPT)

iii)Encoder-Decoder (e.g., T5, BART)

1. When to Use Which Transformer Architecture
2. Generative AI End-to-End Project Lifecycle
3. Key Applications of Generative Models
4. Real-world Use Cases of Large Language Models (LLMs)

Module 5: Introduction to Large Language Models (LLMs)

1. What is llm
2. Types of llm
3. LLM providers
    1. hugging face
    2. open ai
    3. groq

Module 6:

1. Hugging Face Overview:
    1. What is Hugging Face?
    2. How to Use Hugging Face Models
    3. API Key Generation
2. Selecting Models & Tokenizers
3. Pre-trained Models in HF:
    1. Text-to-Text
    2. Text-to-Image
    3. Text-to-Speech
    4. Text to video
    5. Speech-to-Text
    6. Speech to speech
    7. Image to text

Projects:

1. Project on using hugging face (making the conversation with hugging face model)
2. Project (image to text)

7. Open ai

- exploring the open ai play ground
- accesing the models, and api key
- How ChatGPT is Trained – Behind the Scenes

Project

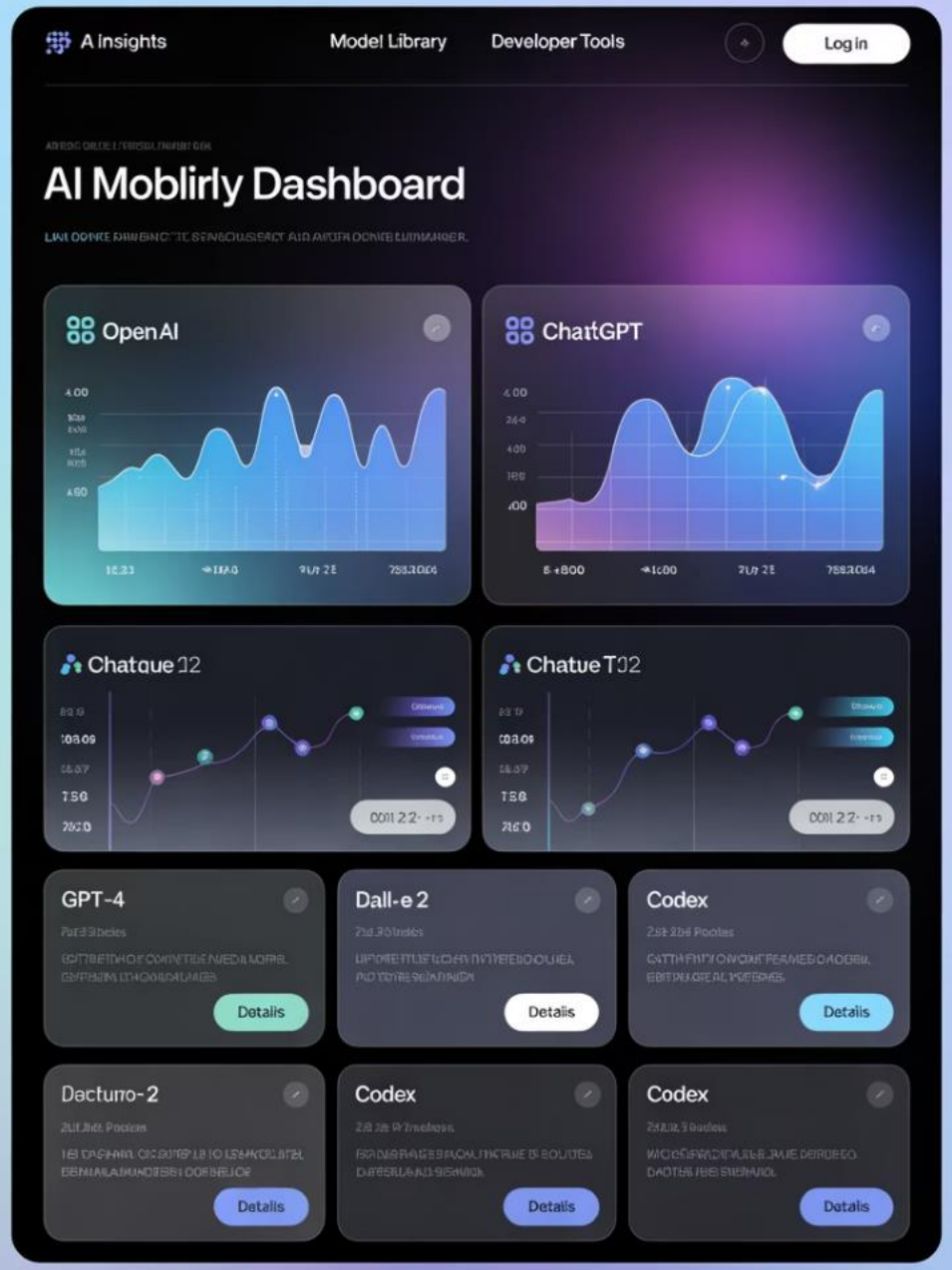
- text completion (blog creation)
- restaurant chat bot (taking orders from customer)

8) Groq

- Explaring the playground in groq
- get the api key and access the models for projects

Project

- Text Summarization



Module 8: Types of LLMs & Project Applications

1. Text Completion Models(level – projects)

- 🔧 Projects:
- Redaction Improver
- Blog Post Generator
- Text Summarization

RAG Architecture with Langchain

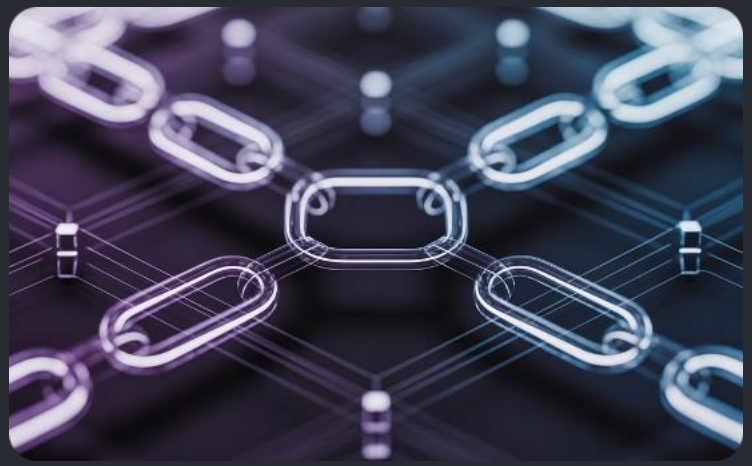
- What is langchain?
- Exploring the langchain documentation
- Flow of RAG:
  - Data Loaders(different types)
    - Analyzing CSV, PDF, and JSON Files using LangChain
  - Splitters
  - Prompt and prompt engineering
    1. What & Why of Prompt Engineering
    2. Prompt Engineering with ChatGPT Custom Instructions
    3. Deciding What Context to Add
    4. Zero-, One-, and Few-Shot Prompting, chain of thought, Tree of thought, React prompting
    5. Providing Effective Prompts to LLMs
  - Embedding Techniques (types)

> This section explores vector databases, their types, and applications in similarity search, with a focus on popular implementations like ChromaDB, FAISS, and Pinecone, concluding with an overview of retrievers in chains.

- Vector Databases (types)
    1. Introduction to Vector Databases
    2. Vector DB Use Cases
    3. Text Embedding & Similarity Search
    4. Types of Vector Databases:
        1. ChromaDB
        2. FAISS
        3. Pinecone
        4. Milvus (with Attu UI)
        5. Weaviate
        6. Neo4j for Graph + Vector Search
- Retrievers with Chains (types)

LangChain Expression Language (LCEL)

A visual overview of key LCEL concepts and applications



Chains and Runnables

Built-in Runnables and Functions in LCEL

Combining LCEL Chains for Complex Workflows

Practical RAG Demo

Implement RAG with LCEL components

Memory integration for context retention

LLM Integration

Get access to any one LLM

🔧 **Project: Q&A on Your Own Documents**

Types of RAG implementations

L&L AI Solutions

RAG Architecture with Llamaindex

- What is Llamaindex
- Key Differences: LlamaIndex vs. LangChain vs. Llama (the model)
- i)exoring the Llamaindex documentation
- ii) Flow of RAG:
- Data Loaders
- Splitters
- Embedding Techniques
- Build Index

i)Vector Databases

- Retrievers with Chains
- Setup Memory Context
- LLM Integration
- 🔧 **Project: Q&A on Your Own Documents(csv file)**
- 2. Types of RAG s

- **LangChain Ecosystem Overview: LangServe, LangGraph, LangSmith**
  - Hands-on Demo: LangServe for Deployment
  - Hands-on Demo: LangGraph for Building Agentic Workflows
  - Hands-on Demo: LangSmith for Debugging, Testing, and Monitoring

Level 1 Application Development

1. Advanced Chatbot with Memory
2. Key Data Extraction
3. Sentiment Analysis Tool
4. SQL-based Question Answering Application
5. PDF-based Question Answering
6. Basic Retriever Applications
7. RAG Application

Level 2 Application Development

1. Application for Converting Slang to Formal English
2. Blog Post Generation Application
3. Text Summarization with Split
4. Text Summarization Tools
5. Key Data Extraction from Product Reviews
6. Interview Questions Creator Application
7. Medical Chatbot Project

Level 2 Application Deployment

1. Multimodal Gen AI Applications
    1. Steps to implement Multimodal LLM Applications
    2. Building Multimodal LLM Applications with LangChain & GPT 4o Vision

Level 3 (Professional) Application Development

Introduction to Level 3 Application

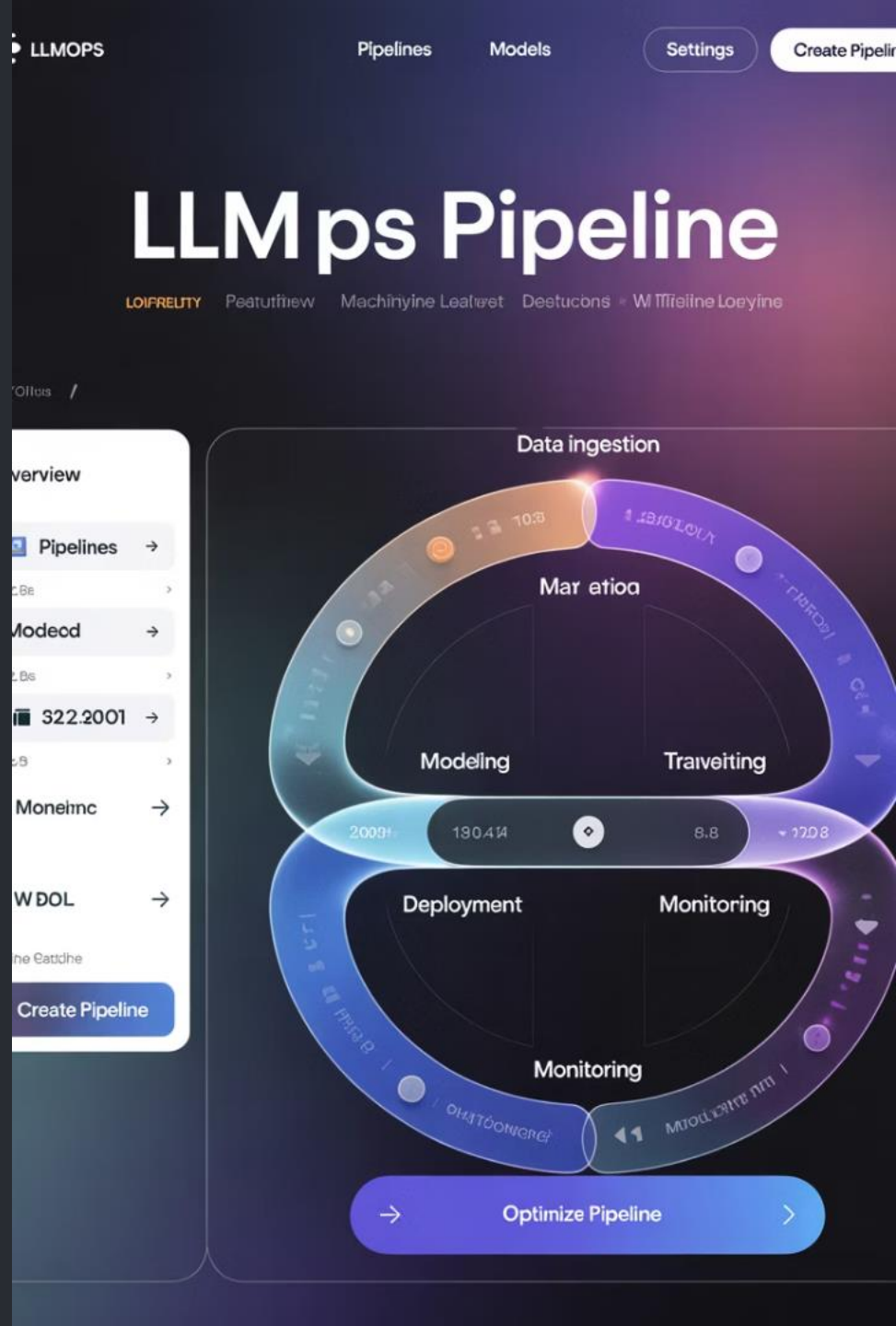
- Project 1: Advanced RAG-Based Knowledge Management System
- Project 2: Medical Diagnostics Support Application
- Project 3: Image generation (DALL·E, Midjourney)
- Project 4: Youtube video summarizer and youtube script writing

L&L AI Solutions

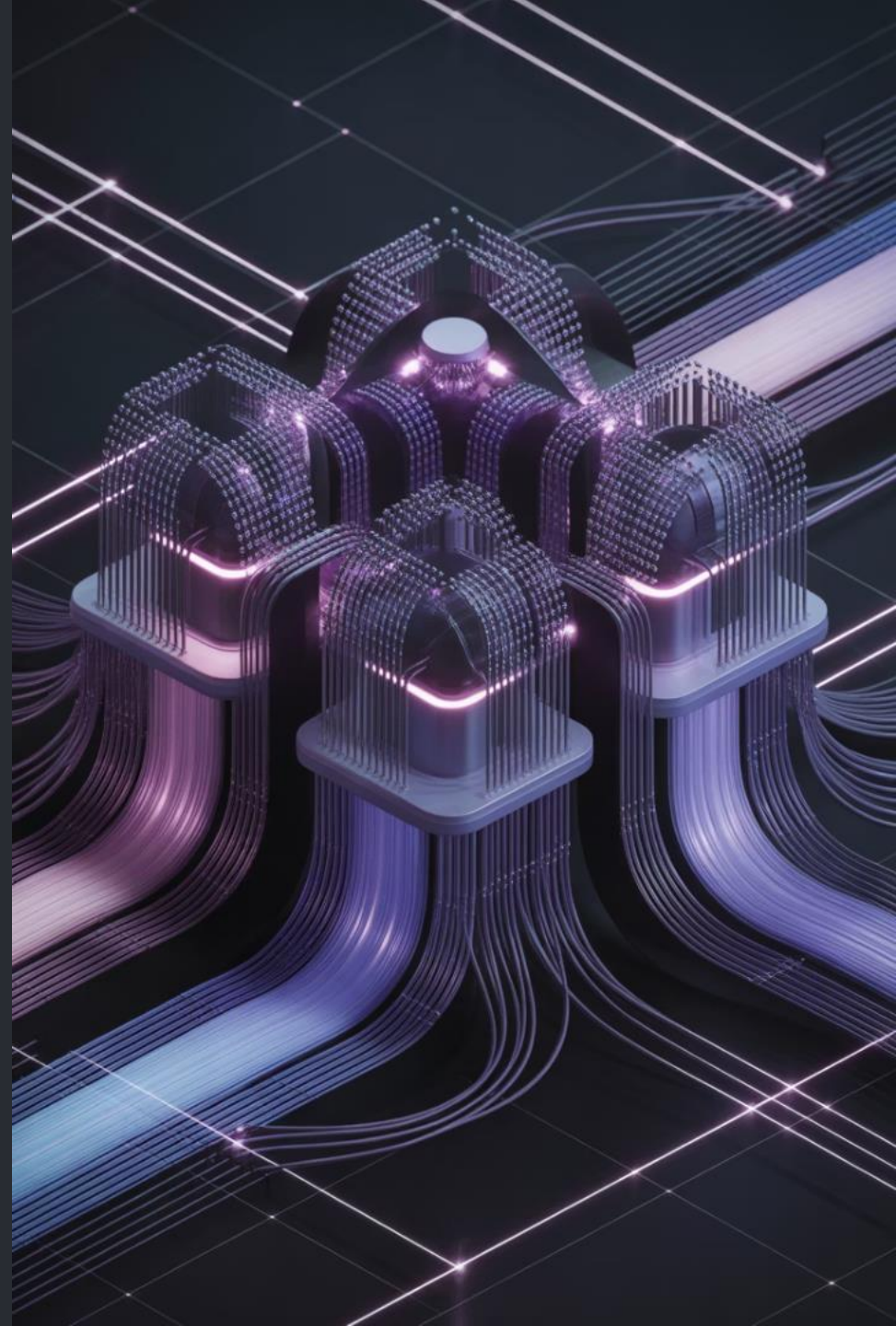
Introduction to LLMOps

LLMOps

1. What is LLMOps?
2. Why LLMOps is Different from Traditional MLOps
3. The Evolution from MLOps to LLMOps
4. Key Challenges in LLMOps
5. Overview of the LLM Lifecycle
6. Core Components of an LLMOps Pipeline
7. Tools and Frameworks Shaping LLMOps
8. Real-World Use Cases of LLMOps
9. Skills Required to Become an LLMOps Engineer
10. Course Overview and How to Navigate This Series



1. Foundation of LLMs
2. Understanding Transformer Architectures
3. Anatomy of a Large Language Model
4. Tokenization and embeddings
5. How Self-Attention Mechanism Works
6. Understanding Context Window in LLMs
7. Prompt Engineering: Basic Concepts
8. Prompt Engineering: User & System Prompts in LLMs
9. Open Source vs Proprietary LLMs
10. Comparison: LLaMA, GPT, Claude, Mistral, Gemini
11. Pretraining vs Fine-tuning vs Instruction-tuning



LLMOps Tooling Landscape

- Introduction to LangChain
- Introduction to LlamaIndex
- Introduction to Haystack
- LangChain vs LlamaIndex vs Haystack
- Introduction to Vector Databases
- Introduction to Pinecone
- Introduction to Weaviate
- Introduction to FAISS
- Introduction to ChromaDB
- Introduction to LangSmith
- Introduction to LangServe

LLM Development Topics

- Introduction to HuggingFace Inference Endpoints
- Introduction to FastAPI for LLM Inference
- Setup MLflow on AWS for LLMOps
- Training Models with MLflow: A Hands-On Guide
- MLflow for Model Inference: Techniques and Practices
- Building CI/CD Pipelines with GitHub Actions
- Data Management for LLMs
- Data Collection Strategies for LLMs
- Scrapping Web for LLM Datasets
- Cleaning & Preprocessing Raw text data
- Chunking Strategies
- Embedding Data for Retrieval

Building a Private Knowledge Base

- Using LLMs to Generate Synthetic Data
- Training and Fine-tuning LLMs
  - Introduction to Fine-tuning
  - RAG vs Fine-tuning
  - Introduction to PEFT
  - Introduction to LoRA & QLoRA
  - Fine-tuning LLMs using PEFT
- Retrieval-Augmented Generation (RAG)
  - What is Retrieval-Augmented Generation (RAG)?

- Working with Custom Data (Data Loaders) & RAG Basic Concepts
- Different RAG Components like (Splitters, Embeddings, Vector Stores, Retrievers, Top k)
- RAG Implementation with LCEL
- Model Serving and Inference
- Introduction to Model Inference
- Serving LLMs with FastAPI
- Dockerizing LLM Inference Services
- Serving LLMs with LangServe
- Evaluation & Monitoring

Course Outline

- LLM Evaluation With MLFlow And Dagshub
- LLM Monitor and Tracing with LangSmith
- LLMOps Platforms
- Why we need LLMOps Platform
- Generative AI with Google Cloud (Vertex AI) a LLMOps Platform
- Vertex AI Hands-On on Google Cloud

Vertex AI Local Setup – Run Gemini Pro on Local Machine

- RAG on Vertex AI with Vector Search and Gemini Pro
- LLM powered application on Vertex AI
- Fine tuning Foundation Model VertexAI
- Introduction to AWS Bedrock
- Hands-on AWS Bedrock
- Capstone Projects
- Project Walkthrough
- Project Setup & Template
- Data Ingestion: Chunking, Embedding and Vector store
- RAG Pipeline and User App
- Project Deploy: AWS CICD