Advanced Data Science

1

Foundations

Python programming, NumPy, Pandas, and database management fundamentals

Analysis & Visualization

Statistical analysis, dashboarding, and data preprocessing workflows

Machine Learning

ML algorithms, model evaluation, and feature engineering techniques

Deep Learning

Neural networks, NLP, and transfer learning approaches

Generative AI

Generative frameworks, diffusion models, and large language models

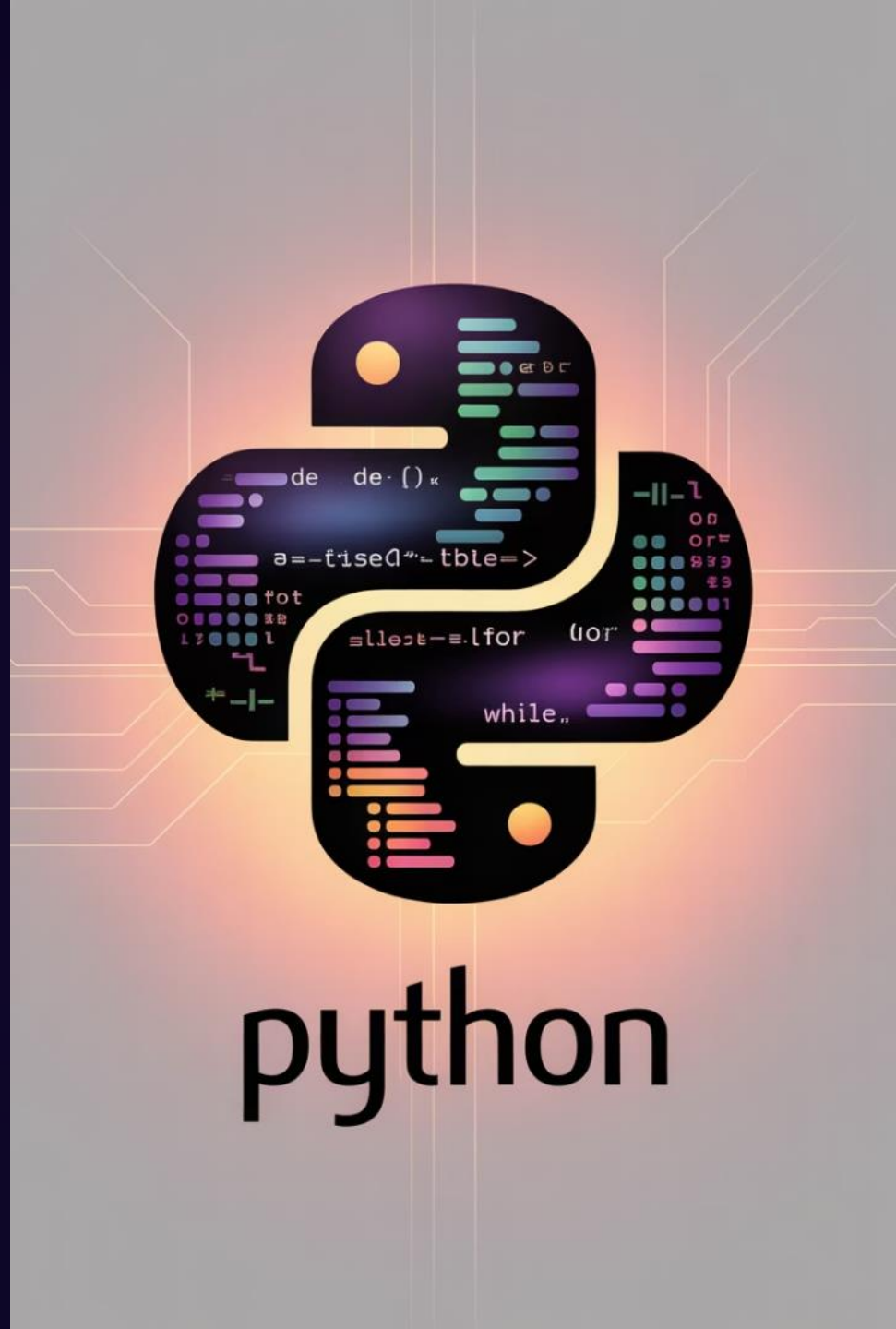
Cloud Deployment

AWS deployment, MLOps practices, and ML pipelines

L&L AI Solutions

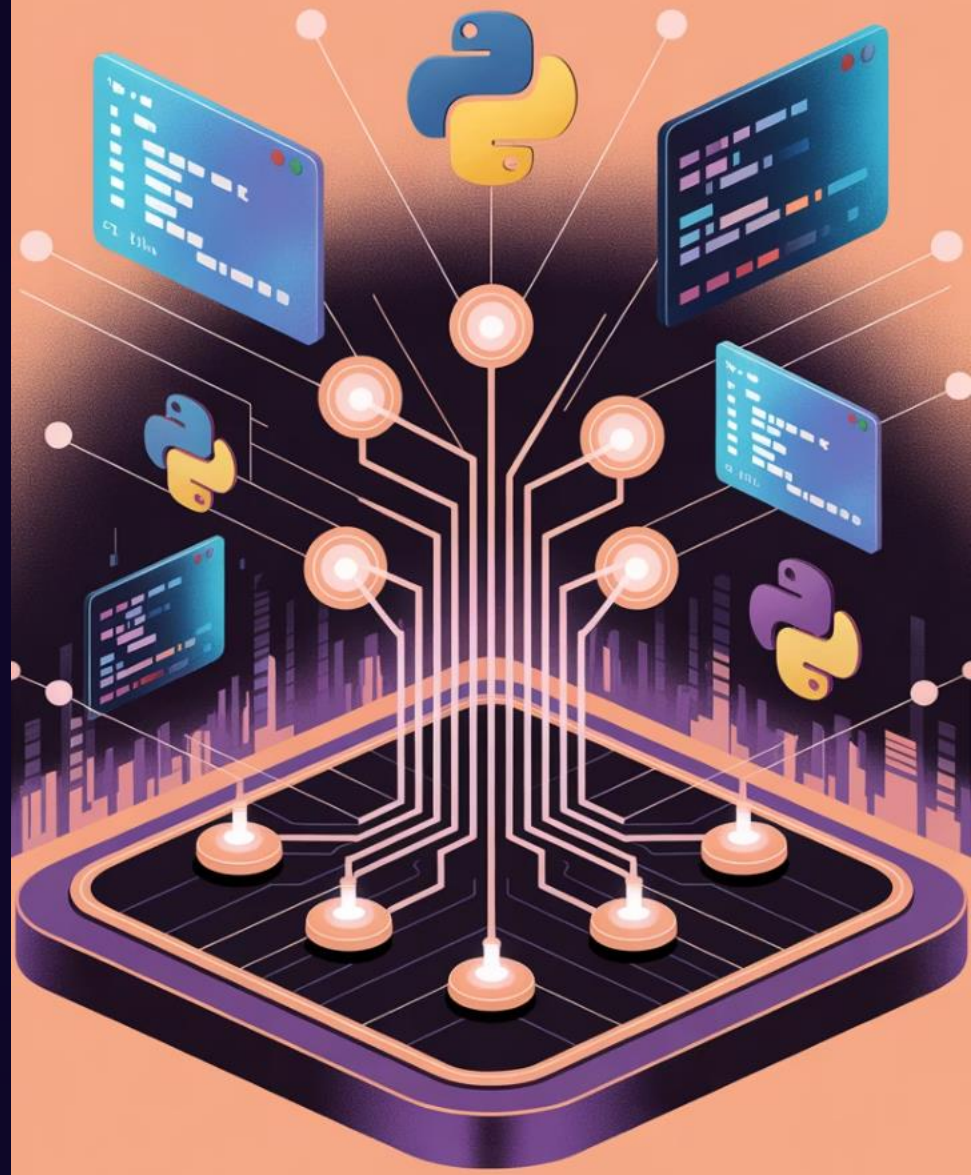
Python

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

Data Analytics Using Python

Module: Introduction to Numerical Computing with NumPy

28. Introduction to NumPy

- What is NumPy?
- Benefits of using NumPy
- Installing and importing NumPy

29. Core Operations in NumPy

- Array creation techniques (arrays, arange, linspace, zeros, ones, etc.)
- Indexing, slicing, and iterating
- Mathematical and statistical operations
- Copies vs. views in NumPy
- String operations in NumPy
- Reshaping and broadcasting

Module 2: Data Handling with Pandas

30. Introduction to Pandas

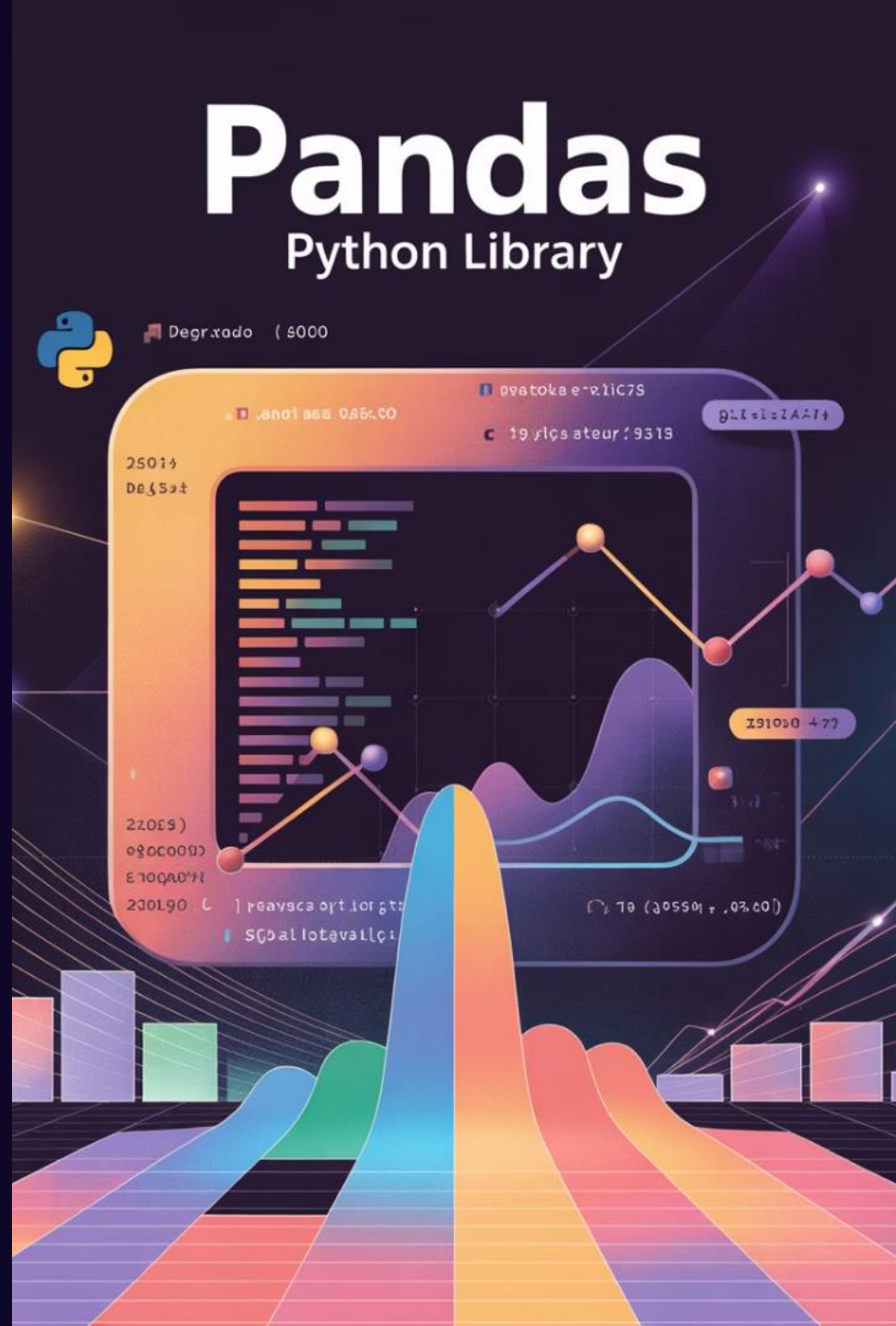
- What is Pandas and why is it important?
- Understanding Series and DataFrames
- Creating and manipulating DataFrames and Series

31. Data Wrangling and Manipulation

- Indexing and Selecting Data
  - .loc[], .iloc[], Boolean indexing
- Data Aggregation
  - Using groupby() for grouping and aggregation
  - Aggregation functions: mean(), sum(), count(), etc.
- Data Merging and Combining
  - merge(), concat(), join()

32. Grouping Operations

- Hierarchical indexing
- Split-apply-combine strategy using groupby()



Module 3: Data Cleaning and Preprocessing

34. Data Cleaning Techniques

- Handling missing values (isna(), fillna(), dropna())
- Removing duplicates
- Renaming columns
- Converting data types
- Handling outliers (z-score, IQR)

35. Data Transformation

- Sorting and ordering
- Merging and concatenating datasets
- Pivot tables and crosstabs
- Melting and reshaping datasets

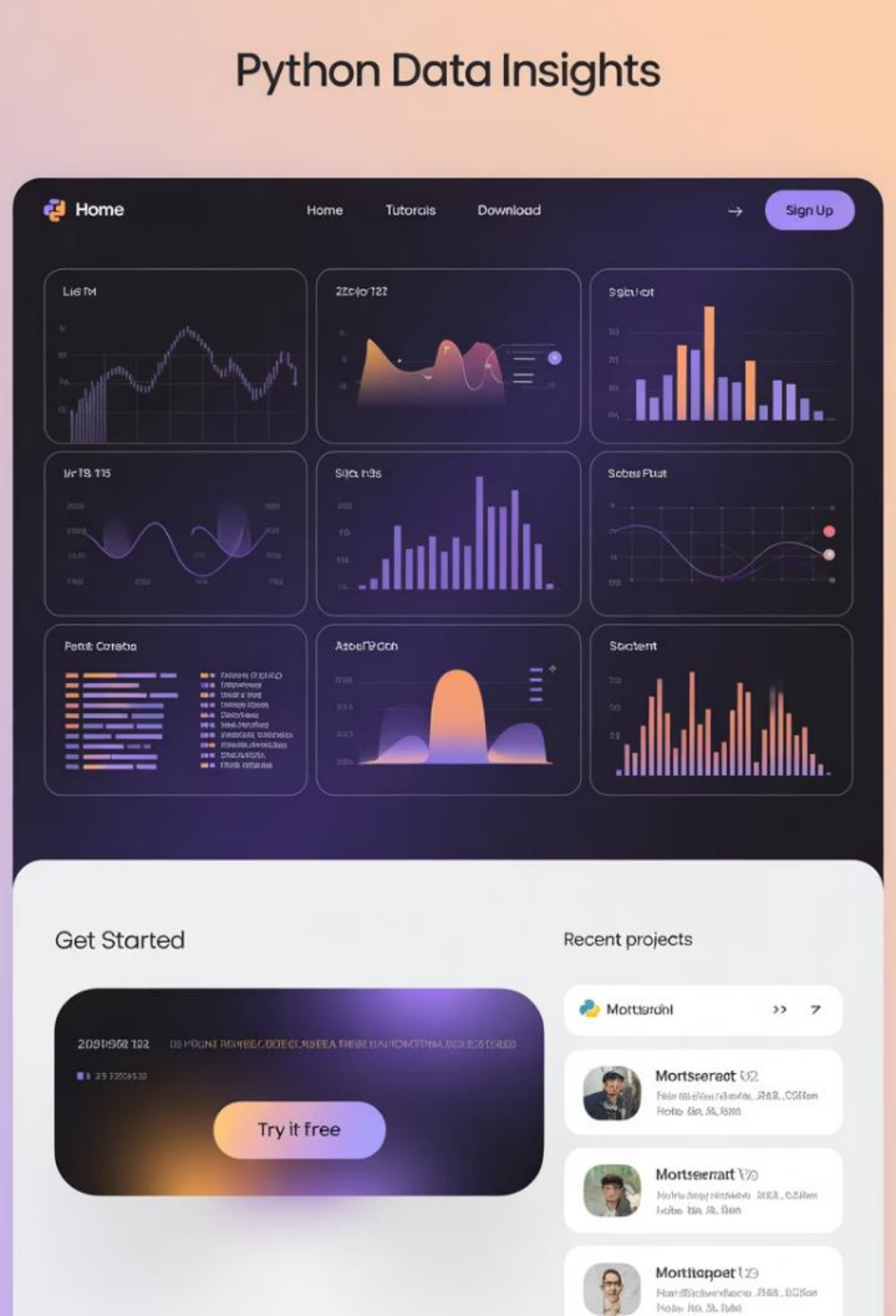
Module 4: Data Visualization

36. Data Visualization Using Python Libraries

- **Matplotlib Basics**
  - Line plots, bar charts, histograms, scatter plots
- **Seaborn for Statistical Visualizations**
  - Distribution plots, box plots, violin plots, pair plots
- **Plotly for Interactive Visuals**
  - Introduction to interactive dashboards and charts

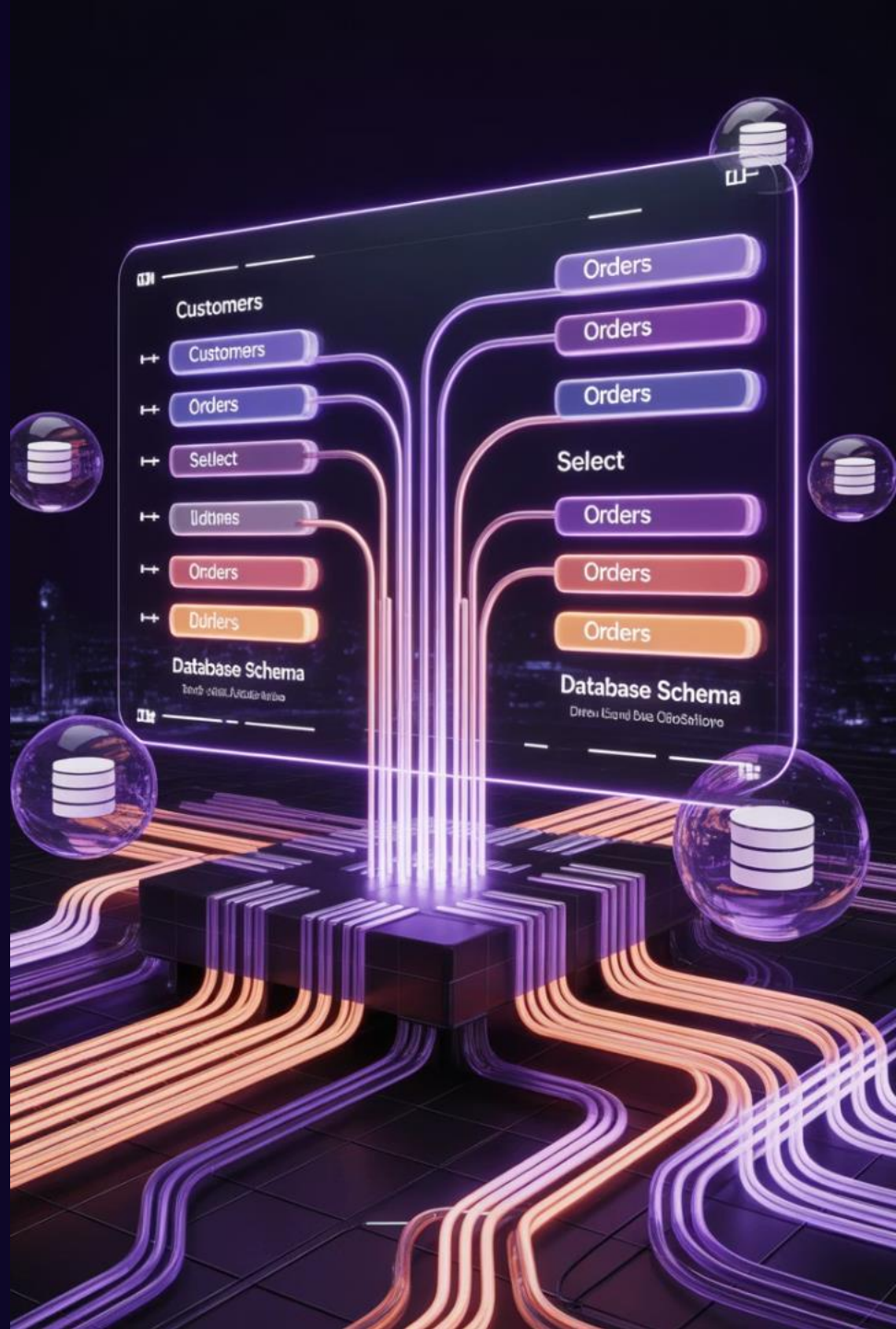
Types of Plots Covered:

- Univariate Analysis (histograms, box plots, etc.)
- Bivariate Analysis (scatter plots, correlation heatmaps)
- Multivariate Analysis (pair plots, facet grids)



MySQL

1. **Introduction to Databases and SQL**
    - Definition of databases and SQL (Structured Query Language).
    - Importance of databases in storing and managing data.
    - Overview of SQL as a standard language for interacting with relational databases.
2. **Data Types and SQL Commands**
    - **Data Definition Language (DDL)**
        - Creating and modifying database objects (tables, views, indexes).
        - Common data types: INTEGER, VARCHAR, DATE, etc.
    - **Data Manipulation Language (DML)**
        - Inserting, updating, and deleting data from tables.
    - **Data Control Language (DCL)**
    - **Transaction Control Language (TCL)**
    - **Data Query Language (DQL)**
3. **Basic SQL Commands**
    - SELECT statement: Retrieving data from tables.
    - FROM clause: Specifying the source tables.
    - WHERE clause: Filtering rows based on conditions.
    - Logical operators: AND, OR, NOT.
    - CASE statement: Conditional logic.
    - NULL functions: Handling NULL values.
    - Comments: Adding comments in SQL code.



Advanced SQL Topics

1 Filtering and Sorting

- Advanced filtering using IN, OR, and NOT operators.
- Sorting data with GROUP BY and ORDER BY clauses.

2 SQL Joins

- Inner Join: Retrieving matching rows from multiple tables.
- Left/Right Outer Join: Retrieving all rows from one table and matching rows from the other.
- Full Outer Join: Retrieving all rows from both tables.
- Self Join: Joining a table with itself.
- Cross Join: Cartesian product of two tables.

3 SQL Aggregations

- Common aggregations: COUNT, SUM, MIN, MAX, AVG.
- Using aggregate functions with GROUP BY clause.

4 Subqueries

- Using subqueries to nest one SELECT statement within another.
- Correlated subqueries and non-correlated subqueries.

5 Window Functions

- ROW_NUMBER(): Assigning a unique sequential integer to each row.
- RANK(): Assigning a rank to each row based on specified criteria.
- DENSE_RANK(): Similar to RANK(), but without gaps in ranking.
- LAG and LEAD: Accessing data from previous or next rows in a result set.
- Using SUM, COUNT, AVG with window functions.
- Creating Views.

Tableau

1. **Introduction to Tableau**
   - Understanding Tableau and its role in data visualization.
   - Installation of Tableau Desktop 10.
   - Connecting Tableau to various datasets.
2. **Inserting Data into Tableau**
   - Importing different types of data into Tableau, including Excel sheets, CSV files, databases, etc.
3. **Visualizing Data Using Tableau**
   - Utilizing appropriate plots, charts, and maps to visualize the data effectively.
   - Creating various types of visualizations such as bar charts, line charts, scatter plots, pie charts, geographic maps, etc.
4. **Tableau for Data Science**
   - Exploring Tableau's capabilities for data analysis in the context of data science.
   - Analyzing, blending, joining, and calculating data within Tableau.
5. **Data Manipulation in Tableau**
   - Working with data hierarchies to organize and structure data effectively.
   - Understanding data blending and its applications in Tableau.
6. **Enhancing Visualizations with Tableau Features**
   - Utilizing parameters to add interactivity and flexibility to visualizations.
   - Creating calculated fields to perform custom calculations and derive insights from the data.

7) Filtering and Dashboard Creation

- Creating interactive dashboards
- Adding actions to dashboards
- Implementing filters and quick filters to refine and focus visualizations
- Designing interactive dashboards to present insights and analysis effectively

8) Advanced Features in Tableau

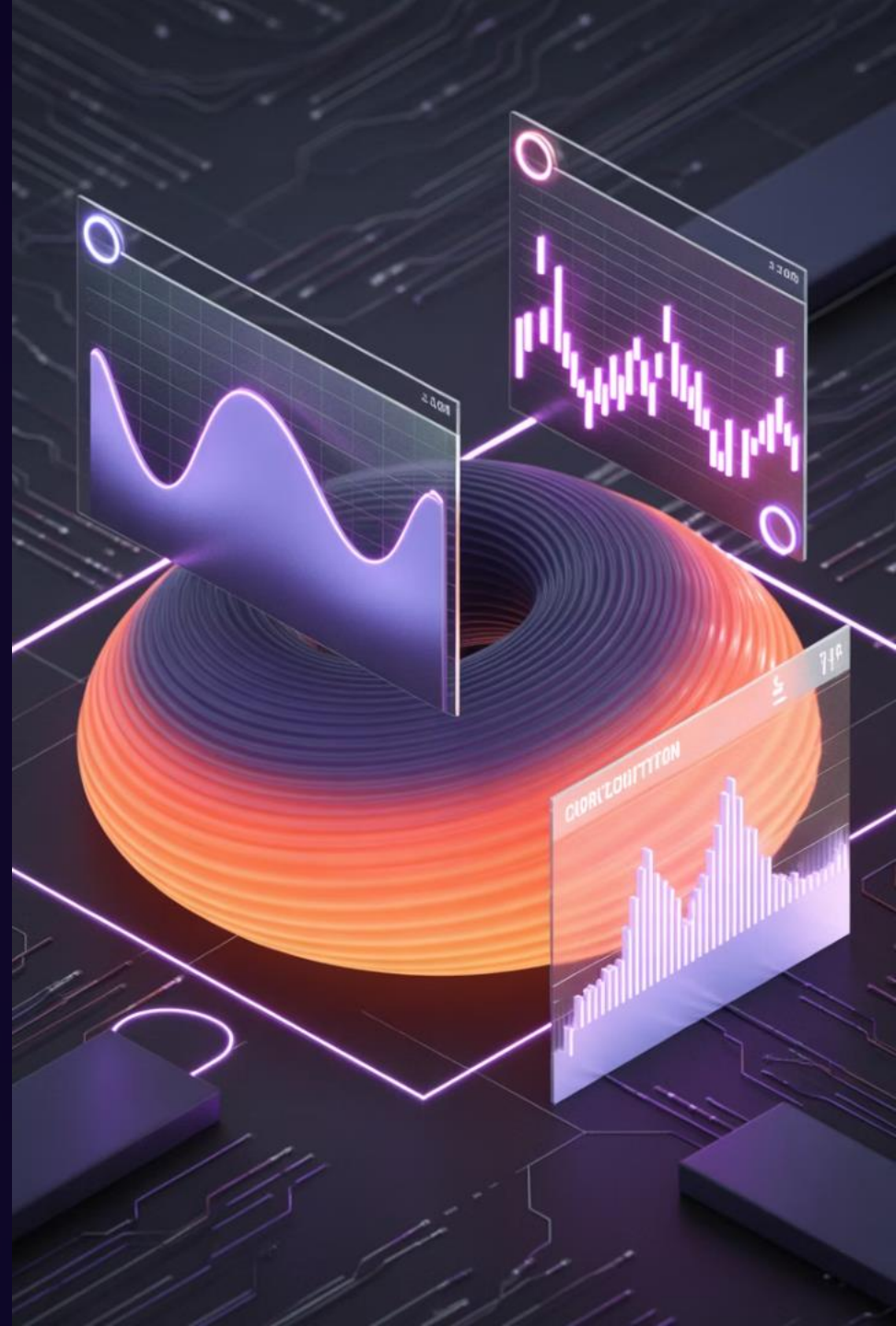
- Incorporating actions to enhance the interactivity and user experience of dashboards
- Exploring advanced features such as forecasting, trend lines, and statistical functions

Math for Data Science

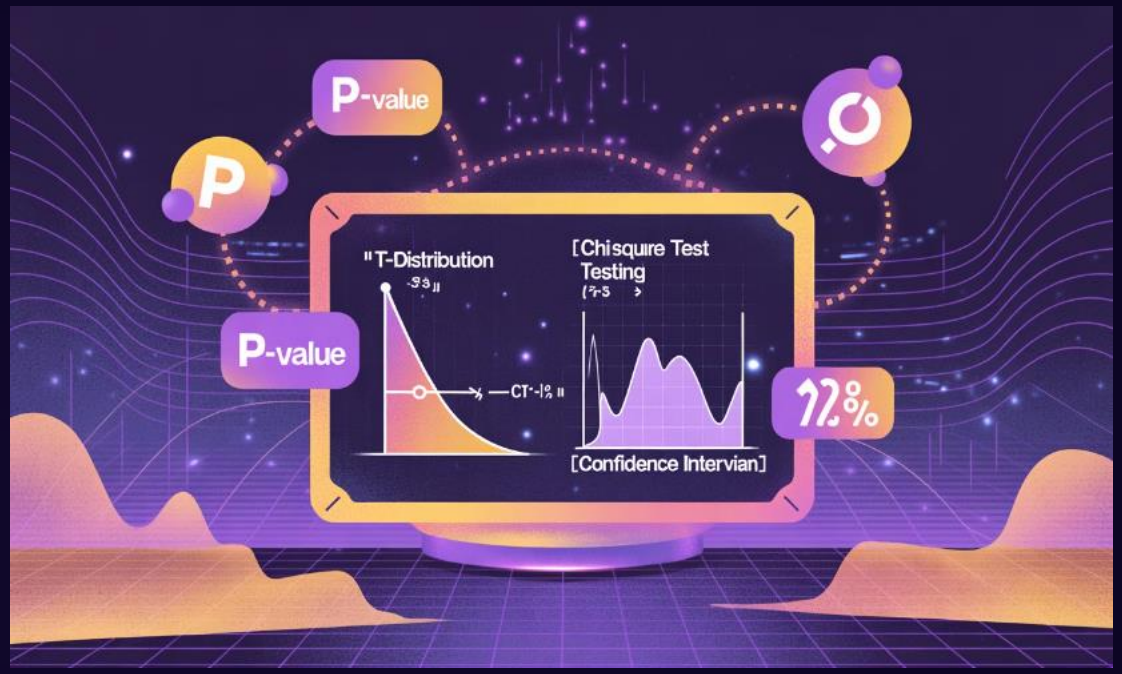
1. **Introduction to Basic Statistics Terms**
   - Overview of fundamental statistical concepts and terminology.
2. **Types of Statistics**
   - Descriptive statistics: Summarizing and describing data.
   - Inferential statistics: Making inferences and predictions about populations based on sample data.
3. **Types of Data**
   - Categorical data: Data that falls into categories or groups.
   - Numerical data: Data represented by numbers.
4. **Measures of Central Tendency**
   - Mean
   - Median
   - Mode
5. **Measures of Dispersion**
   - Range
   - Variance
   - Standard Deviation
6. **Random Variables**
   - Variables whose possible values are outcomes of a random phenomenon.

Statistical Concepts

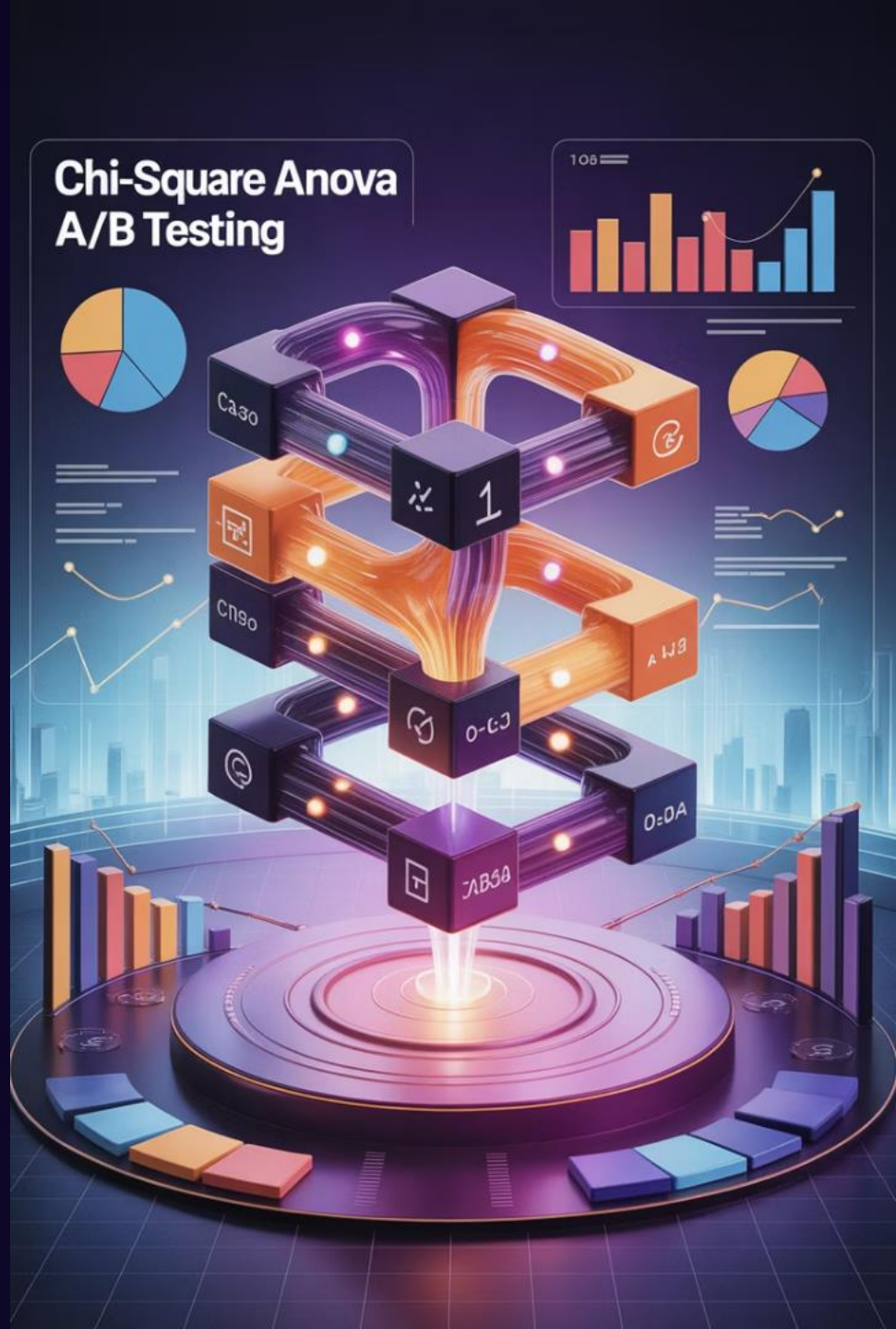
1. **Outliers**
    - What is an outlier
    - Detection of outliers
    - Removing outliers
2. **Skewness**
    - What is skewness
    - Types of skewness
3. **Covariance and Correlation**
    - Covariance
    - Correlation
4. **Probability Density Function and Mass Function**
    - What is probability
    - Sample, population, types of sampling techniques
    - Probability distributions for continuous and discrete random variables
5. **Types of Probability Distribution**
    - What is distribution
    - Types of probability distribution
        - Binomial Distribution
        - Poisson Distribution
        - Normal Distribution (Gaussian Distribution)
        - Bernoulli Distribution
        - Uniform Distribution



1. T-stats, F-test, T-distribution
2. T-stats vs. Z-stats: Overview
3. When to Use a T-test vs. Z-test
4. **Chi-square Test**
5. **Central Limit Theorem and Applications**
6. **Confidence Intervals**
    - Confidence Interval (CI)
    - Confidence Intervals and the Margin of Error
    - Interpreting Confidence Levels and Confidence Intervals
7. **Estimation and Hypothesis Testing**
    - What is Hypothesis
    - Types of Hypothesis
        - P-value
        - Null Hypothesis (H_0)
        - Alternative Hypothesis (H_a)
        - One-tailed Alternative Hypothesis
        - Two-tailed Alternative Hypothesis
        - Type 1 & Type 2 Error
8. Bayes' Theorem



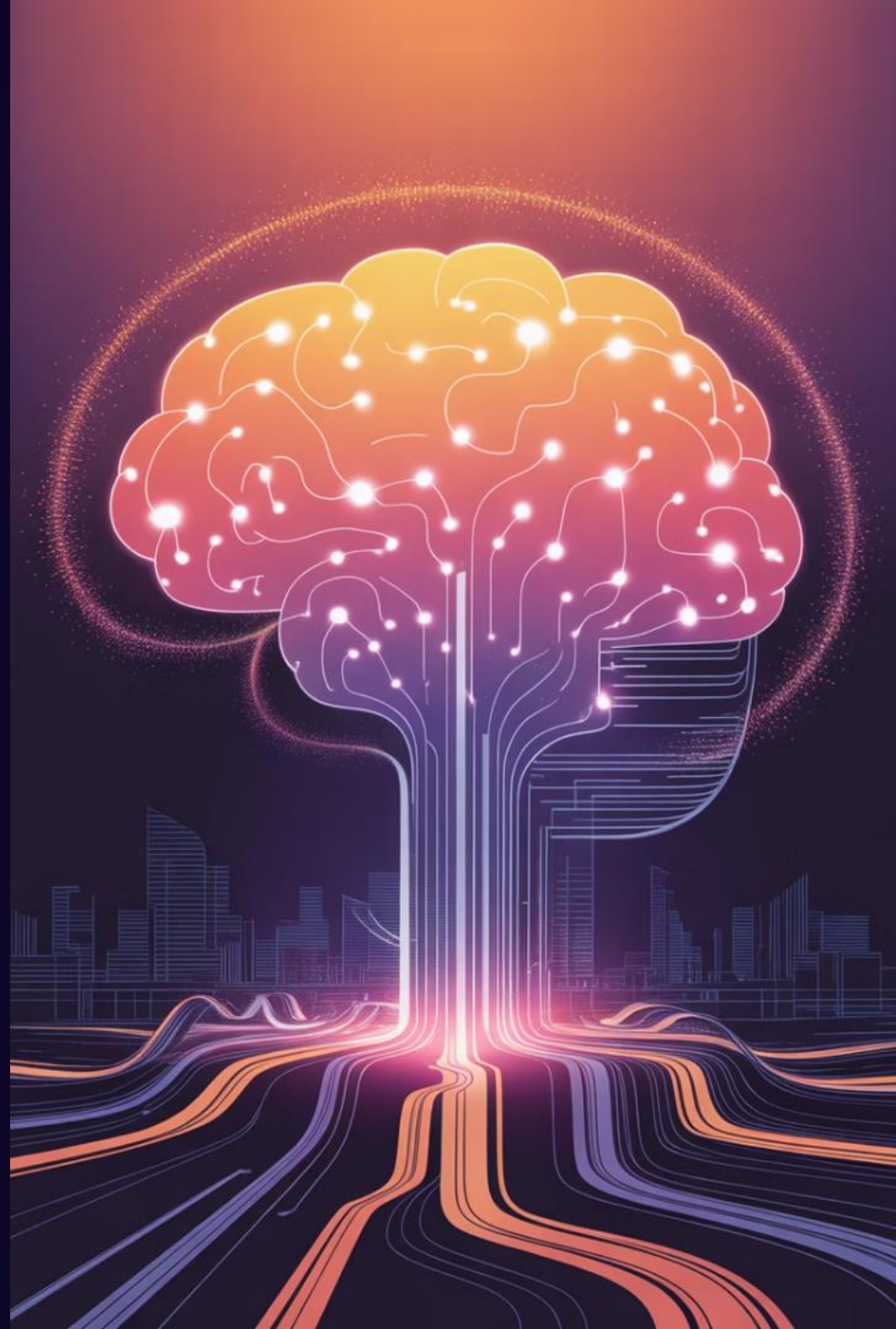
- **Chi-Square Test**
  - Chi-Square Distribution Using Python
  - Chi-Square For Goodness Of Fit Test
- **A/B Testing**
- **ANOVA Test**



Machine Learning

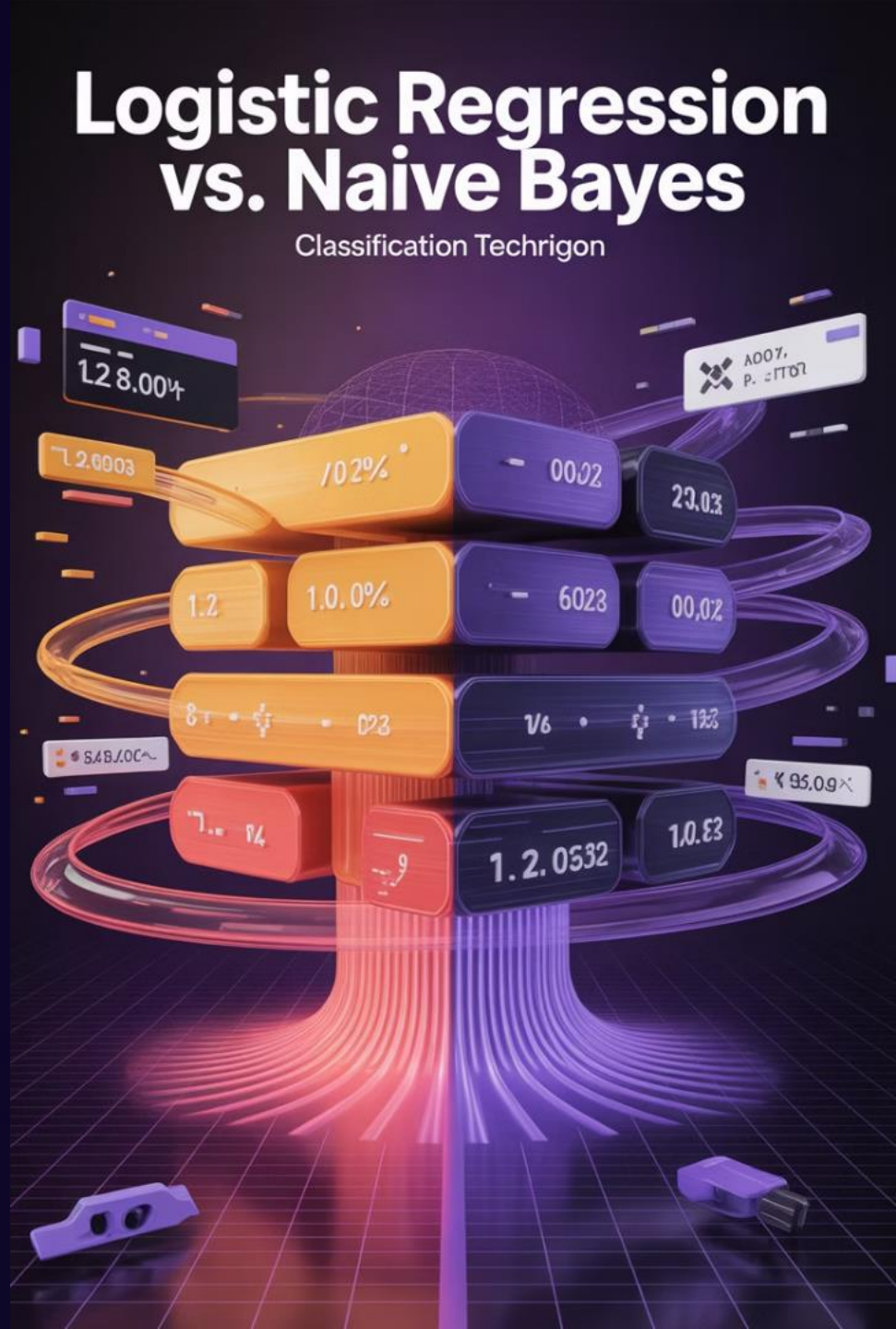
1. Intro to ML and how machine learns
2. Linear Regression(OLS)
    1. Practical Linear Regression
        1. types of regression
        2. Assumptions of linear regression
    2. Bias - Variance Trade-off
    3. Gradient descent
        1. how gradient descent works
        2. Types of gradient descent
    4. Regularized Linear Regression [ridge,lasso,elastic]
    5. Cross validation and Hyperparameter Tuning
    6. Regression metrics(MSE,RMSE,R2,adjusted r2)

Challenge - Supervised Learning Algorithms



Classification techniques

1. Logistic Regression
   - classification metrics(precision, recall, f1 score, roc, auc)
   - project
2. Naive Bayes Classifier



Feature Engineering

1. What is feature engineering, feature extraction
2. Feature selection
   - Types of feature selection
3. Handling the data
   - Handling Missing Data
   - Handling Imbalanced Data
4. Handling Outliers
5. Feature Scaling (normalization, standardisation, Robust scallingS)
6. Covariance And Correlation
7. Vif (variable Influence factor)
8. One hot encoding, label encoding, Dummy encoding, power transformation

L&L AI Solutions

4) Both classification and regression regression Algorithms

- i) Decision Tree
- ii) Ensemble methods
  - a) bagging
    - 1) Random forest
  - b) Boosting
    - 1) Grdient boost
    - 2) xg-boost
    - 3) ada boost
    - 4) cat-boost
  - c) Stacking
- iii) Support Vector Machines(SVM)
- iv) KNN

Challenge - Supervised Learning Algorithms – II

5) Detecting anomalies

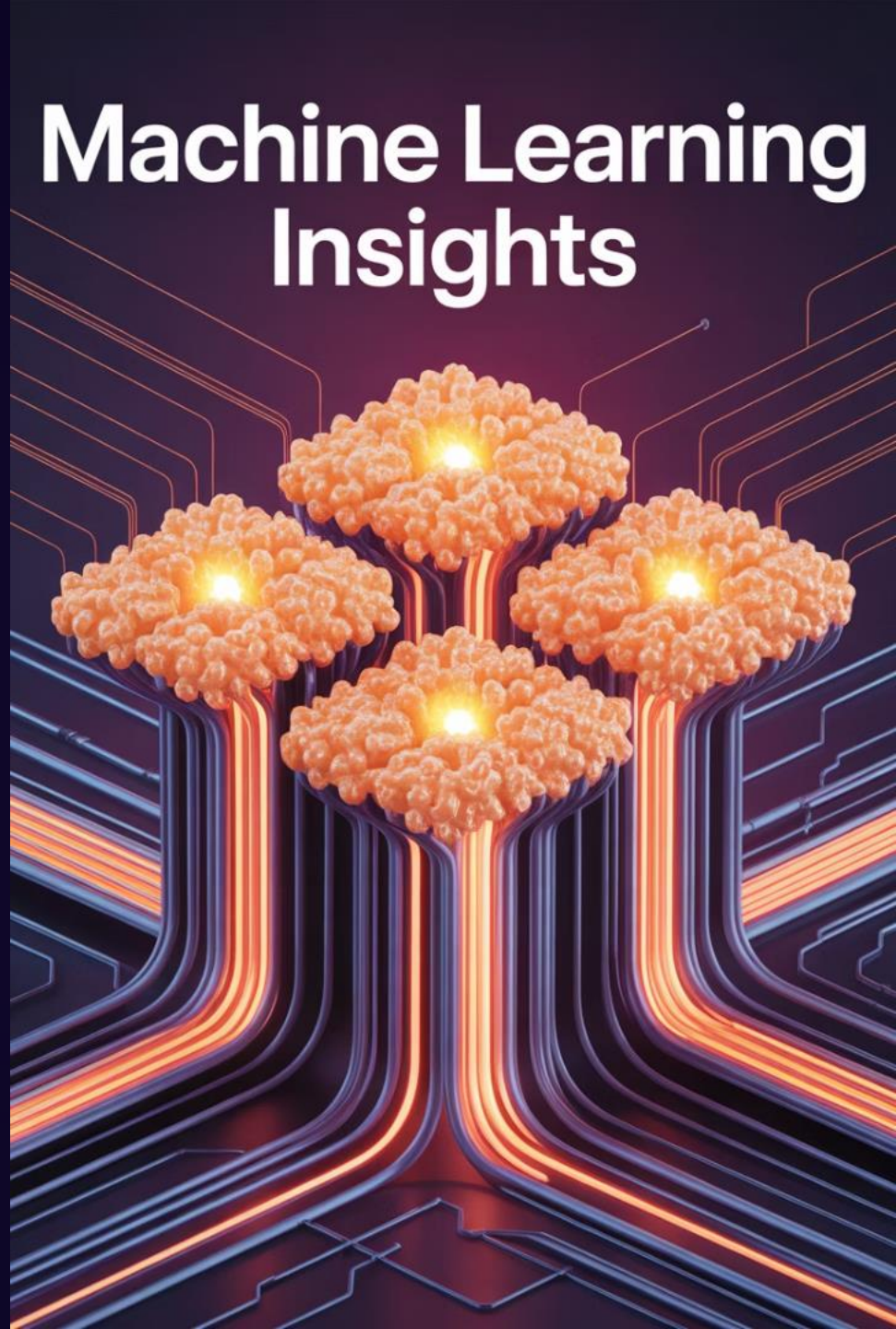
- i) Handling Class Imbalance
- ii) Anomaly Detection

6) Dimensionality reduction techniques

- i) Principal Component Analysis(PCA)
- ii) Linear Discriminant Analysis (LDA)
- iii) t-Distributed Stochastic Neighbor Embedding (t-SNE)

7) Unsupervised Learning Algorithms

- i) K-Means Clustering,k-median
- ii) Hierarchical Clustering
- iii) DBSCAN



8) Recommender Systems

Module 1: Collaborative Filtering

- User-based Collaborative Filtering
- Similarity between users
- Neighborhood-based recommendations
- Item-based Collaborative Filtering
- Similarity between items
- Sparse matrices and data structures
- Limitations: cold start, sparsity
- Hands-on project: MovieLens dataset recommender

📍 Module 2: Matrix Factorization Techniques

- Latent factor models and intuition
- Singular Value Decomposition (SVD)
- SVD++ and Baseline models
- ALS (Alternating Least Squares)
- Libraries: Surprise, LightFM, Implicit
- Model tuning and hyperparameters
- Evaluation: RMSE, MAE, Precision@k, Recall@k

Module 3: Content-Based Filtering

- Concept of item profiles and user profiles
- Feature extraction (e.g., genre, tags, text data)
- Text vectorization: CountVectorizer, TF-IDF
- Similarity measures: Cosine similarity
- Building and evaluating a content-based recommender
- Hands-on project: Course or Movie Recommender using metadata

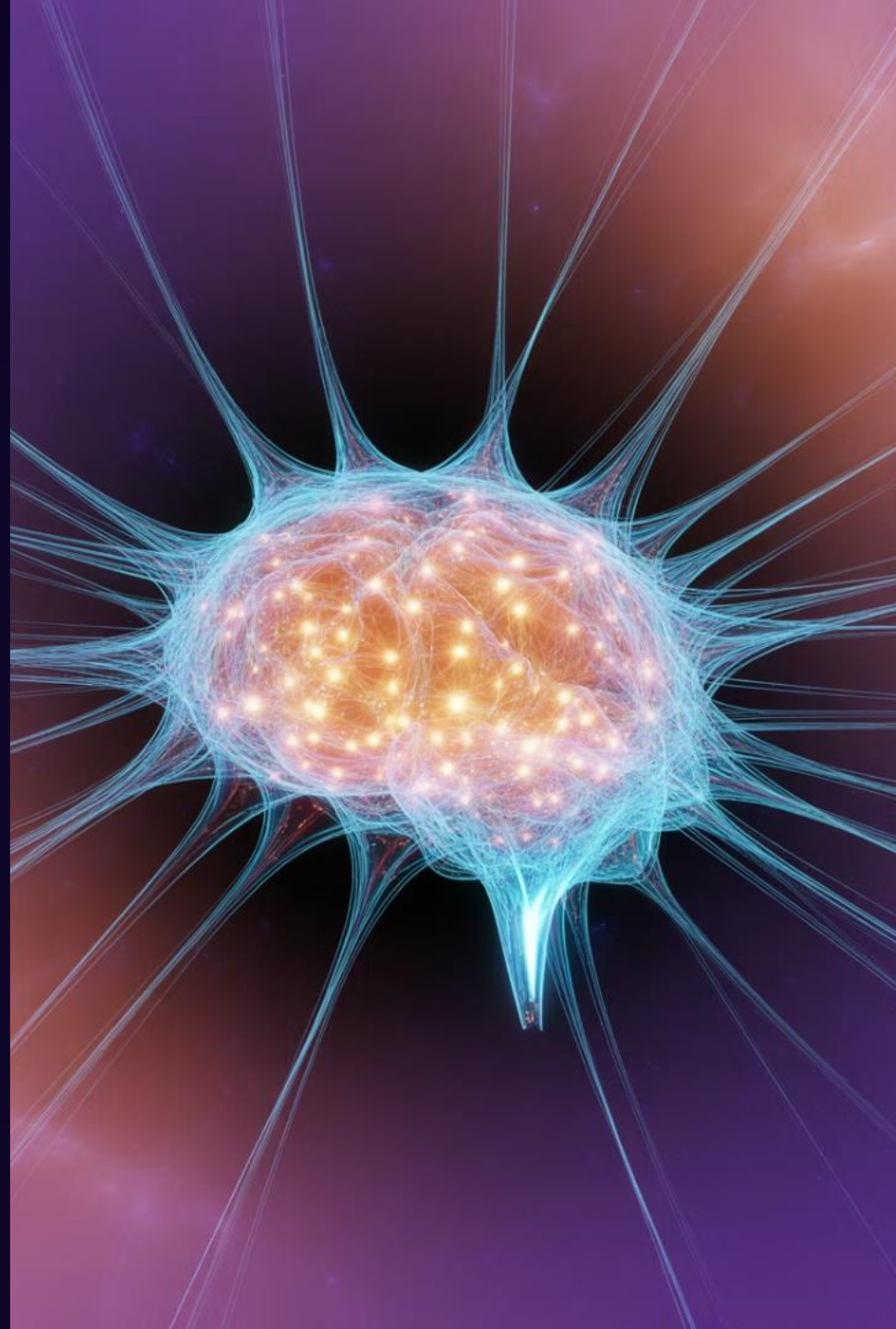
9) Time Series Analysis

- i) Introduction to Time Series Analysis
- ii) Modeling a Time Series Problem

10) Introduction to Model Explainability

Deep learning

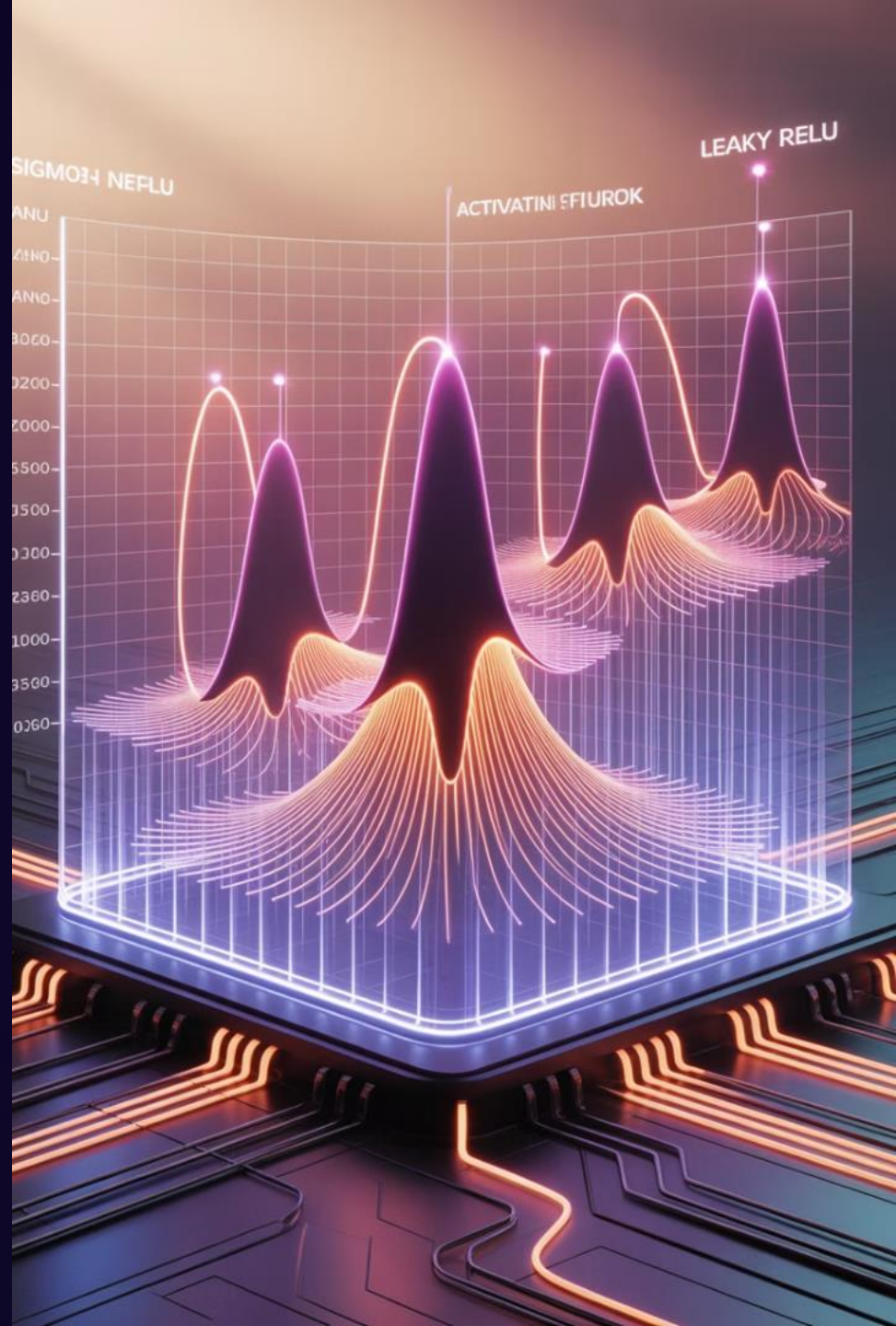
1. **Introduction to Neural Networks**
    1. History of Neural Networks
    2. Comparison with Biological Neurons
    3. How Human Brain Trains by Neurons
    4. How Perceptron Mimics Human Brain
    5. Neuron
    6. Perceptron
2. **Types of Neural Networks**
    1. Introduction to various types of neural networks:
        1. Single-layer Neural Networks
        2. Multilayer Neural Networks



Activation Functions

Types of Activation Functions

- Binary
- linear
- Non linear
  - Sigmoid
  - ReLU (Rectified Linear Unit)
  - Softmax
  - Leaky ReLU
  - Tanh
  - Swish



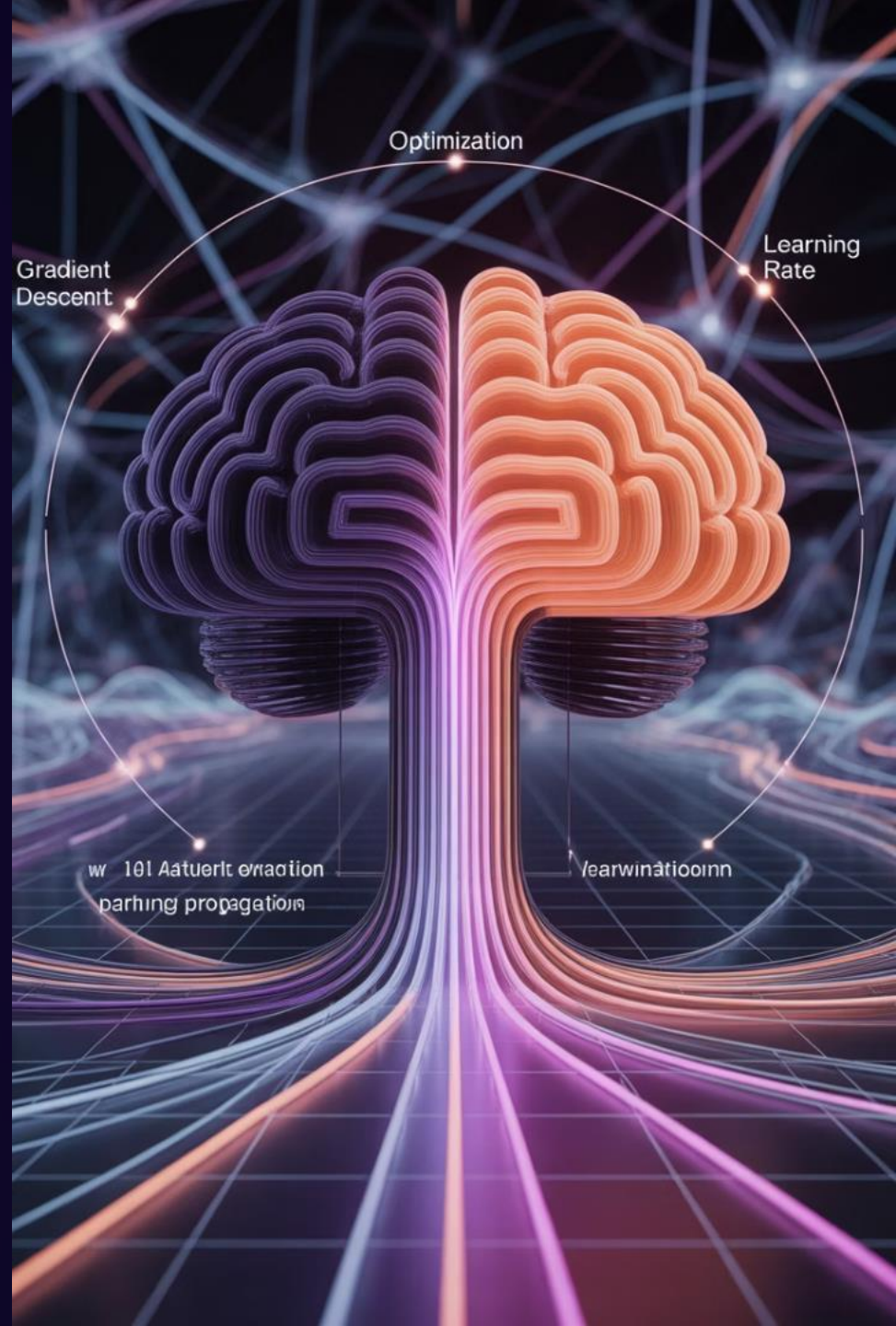
3) Learning Rate and Tuning

- Explanation of learning rate and its significance in training neural networks.
- Techniques for tuning the learning rate for optimal performance.

4) Gradient Descent and Problems

- **Parameters and Gradient Descent:** How parameters (weights) are updated using gradient descent.
- **Vanishing Gradient Problem:** Explanation of the issue where gradients become extremely small during backpropagation.
- **Exploding Gradient Problem:** Explanation of the issue where gradients become excessively large during backpropagation.

5) Forward propagation, Backward propagation. propagation.



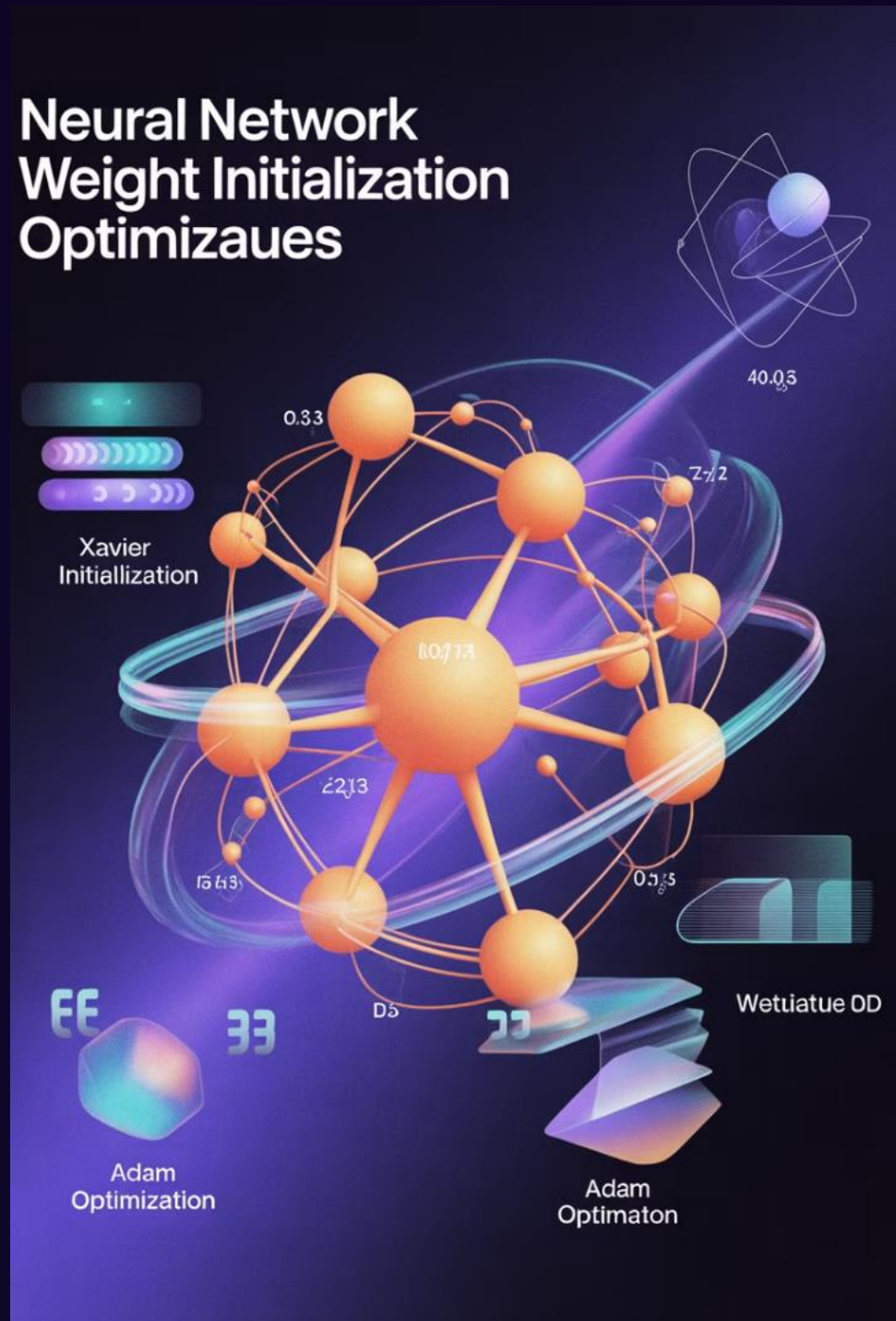
6) Weight initialization techniques

- uniform distribution
- xavier/glorot initialization techniqu
- kaimming he initialization

7) Epochs, Batch

8) Optimizers

- **Definition and Functioning:** Explanation of optimizers and their role in minimizing loss during training.
- **Types of Optimizers**
  - Gradient Descent
  - Stochastic Gradient
  - Descent
  - Adagrad
  - Adadelta
  - RMSprop
  - Adam



iii) Loss functions

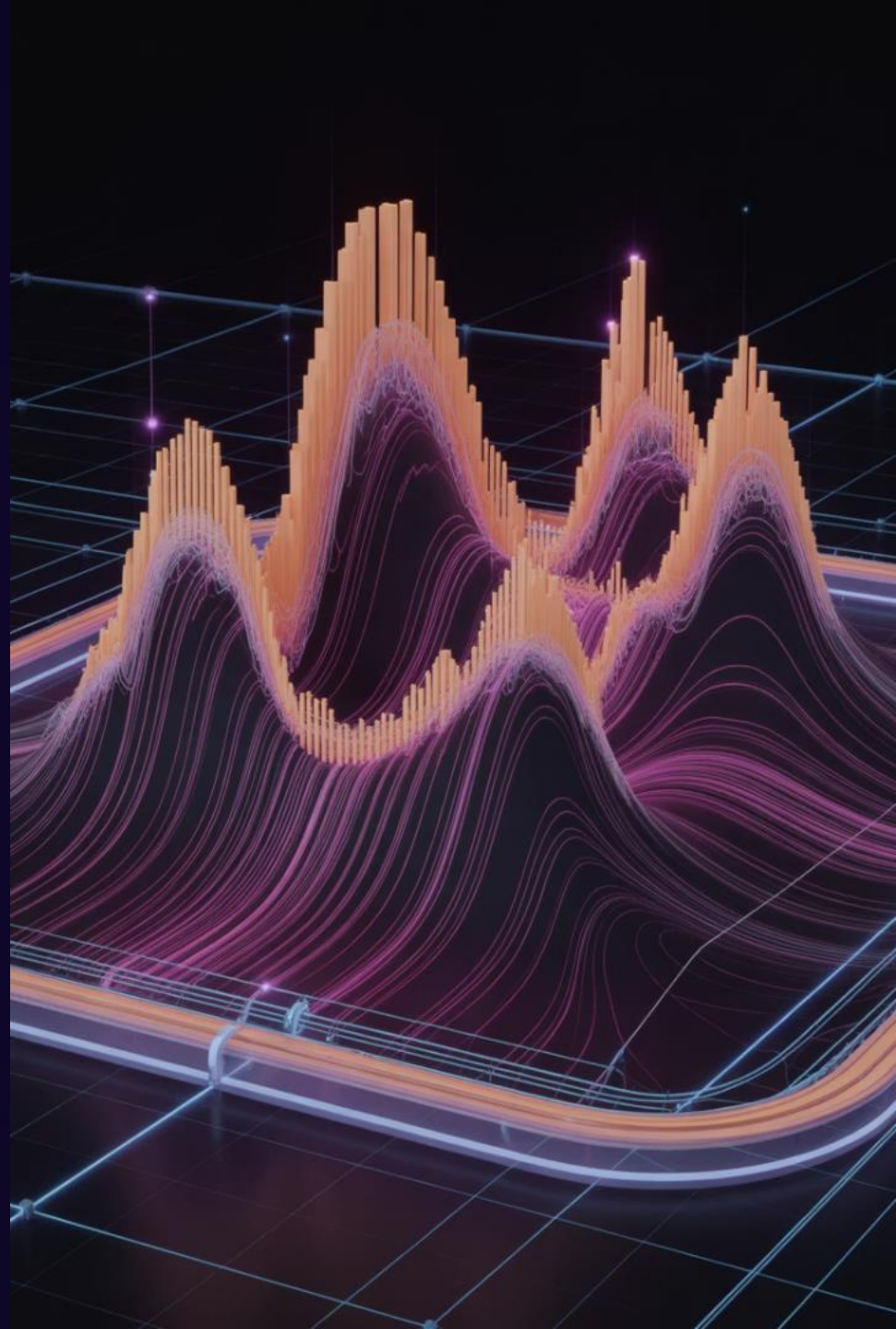
- a)For regression(MSE ,MAE ,Huber loss)
- b)Classification (Binary cross entropy ,categorical cross entropy ,sparse categorical cross entropy,hinge loss , KL divergence

6) Weight Initialization, Regularization, Drop-out Drop-out Layers

- i)Explanation of techniques to initialize weights, prevent overfitting through regularization, and improve generalization using dropout layers.

8) i)Training Steps and Flow:

Basic steps involved in training single-layer and multilayer Neural Networks



9) Architecture Required for Deep Deep Learning (GPU, CPU):

- Explanation of the hardware requirements for training deep learning models, including GPUs and CPUs.

10) Types of Artificial Neural Networks:

- **i)ANN:** Artificial Neural Networks
- **ii)RNN:** Recurrent Neural Networks
- **iii)CNN:** Convolutional Neural Networks
  - **a)OpenCV:** Introduction to the library for computer vision tasks.

Libraries in Deep Learning:

- Overview of popular deep learning libraries:
  - TensorFlow
  - Keras
  - PyTorch
  - Jax
  - maxnet

ANN Projects

Real vs. Fake News Classifier

Goal: Build an ANN to classify whether a news article is real or fake using NLP.

Key Concepts: Text classification, word embeddings, binary classification, overfitting handling

Handwritten Signature Verification (Genuine (Genuine or Forged)

Goal: Classify whether a handwritten signature is genuine or fake using an ANN.

Key Concepts: Image flattening, ANN over image vectors, binary classification

House Price Prediction Using Deep ANN

Goal: Predict house prices based on features like size, location, and number of rooms.

Key Concepts: Feature scaling, normalization, regression metrics, model tuning

Student Performance Predictor

Goal: Predict final exam scores based on hours studied, attendance, past grades, etc.

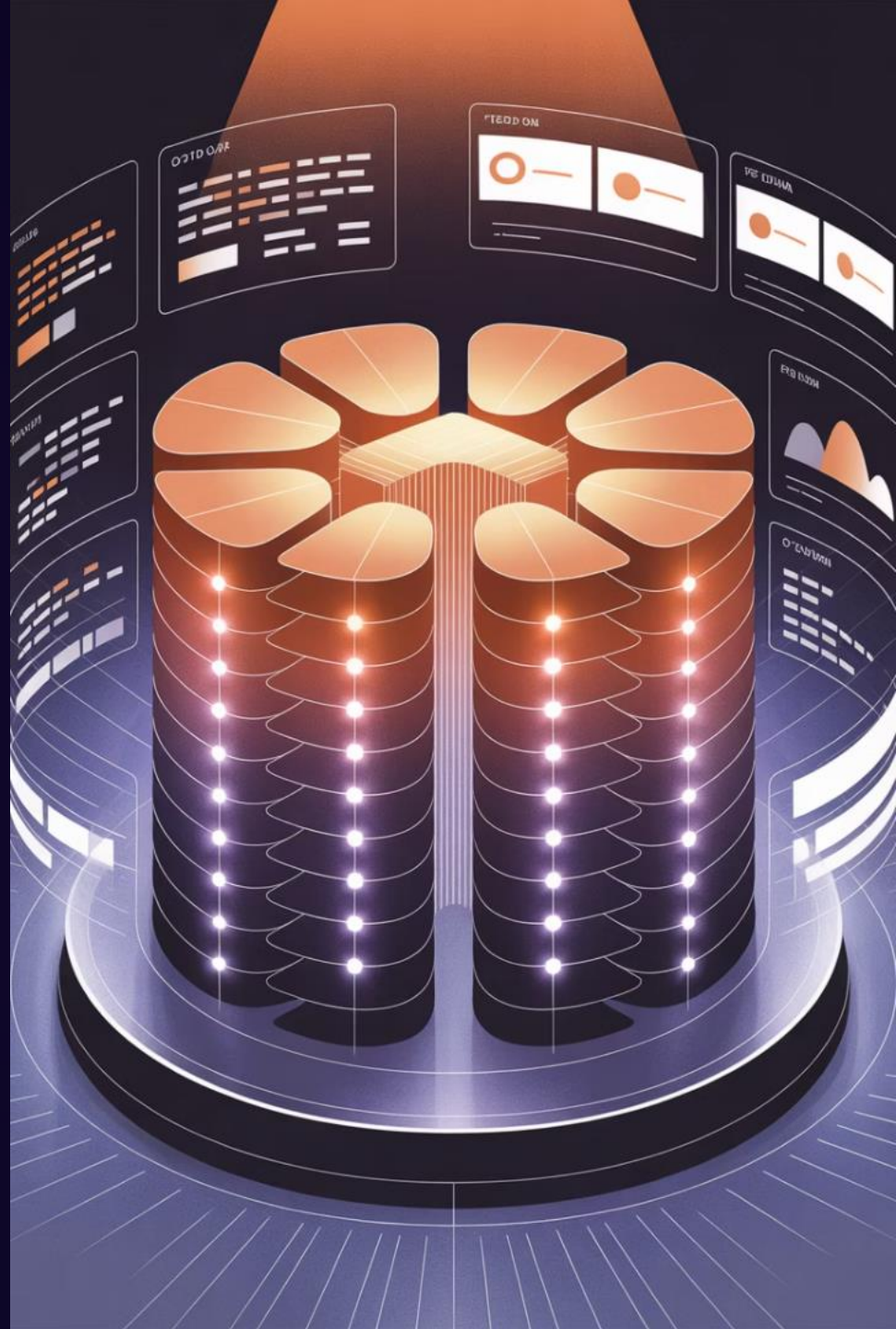
CNN

Module: Computer Vision - The World of Images Images

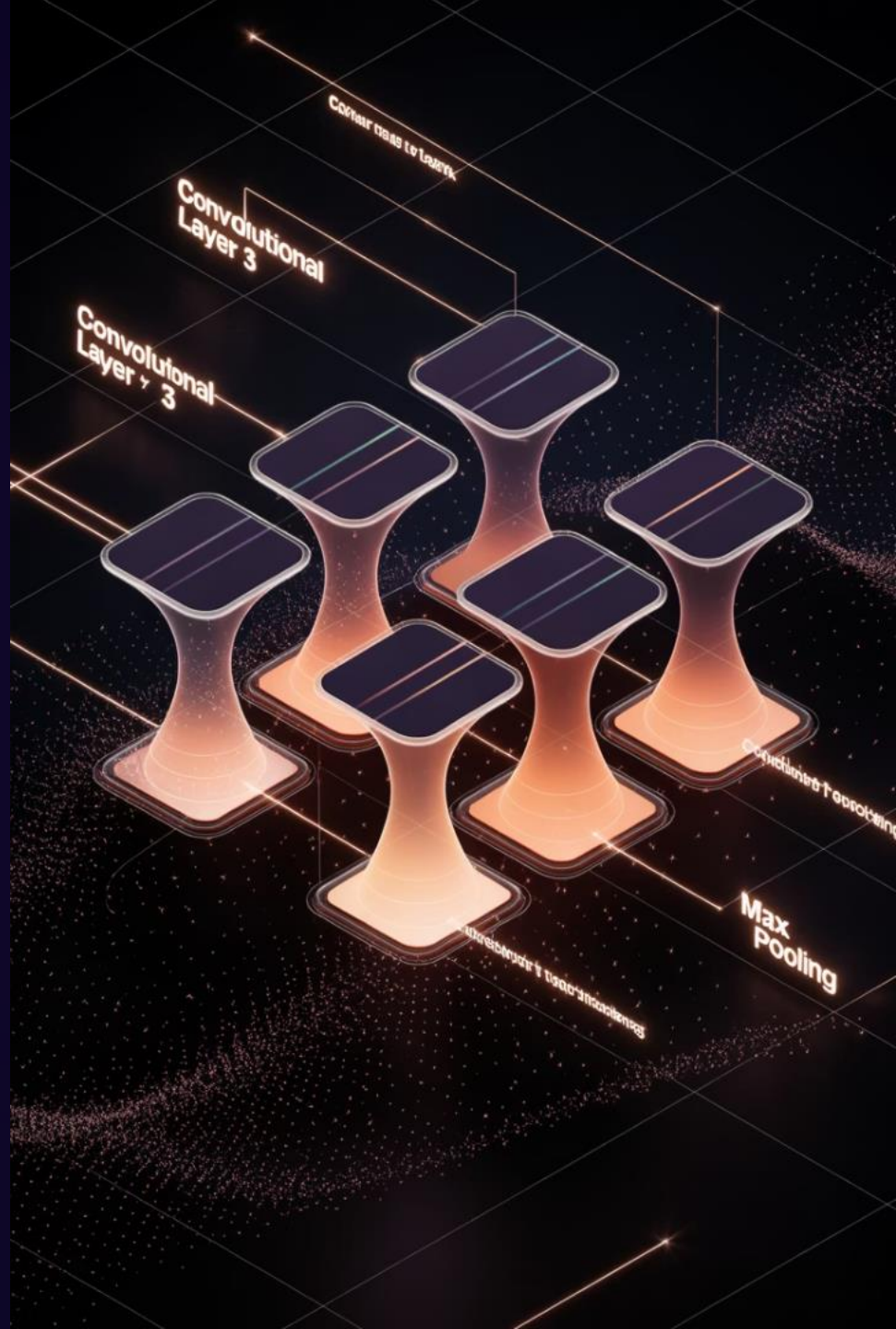
- **Introduction to Computer Vision:**
  - What is Computer Vision? Applications (image classification, object detection, segmentation, facial recognition).
  - How computers "see" images (pixels, channels, resolutions).
- **Image Processing Basics (Briefly):**
  - Image representation (arrays, matrices).
  - Basic operations (resizing, cropping, grayscale conversion).

Module: Convolutional Neural Networks (CNNs) - The Core

- **Why CNNs for Images?**
  - Limitations of traditional ANNs for image data (too many parameters, loss of spatial information).
  - Motivation for convolution (feature sharing, locality, spatial hierarchies).



- **The Convolutional Layer:**
  - **Convolutional Filters/Kernels:** What they are, how they work (sliding window operation).
  - **Feature Maps:** Output of convolutional layers.
  - **Stride:** How filters move across the image.
  - **Padding:** Preserving spatial dimensions.
  - **Activation Functions in CNNs:** Primarily ReLU and its variants.
  - **Practical Example:** Building a simple 1-layer CNN in TensorFlow/Keras or PyTorch.
- **Pooling Layers (Subsampling Layers):**
  - **Max Pooling, Average Pooling:** Reducing dimensionality and achieving translation invariance.
  - How pooling contributes to feature robustness.
- **Fully Connected Layers:**
  - Transition from convolutional/pooling layers to traditional ANN layers for classification.
  - Flattening the output of convolutional blocks.



- **The Complete CNN Architecture:**
- Connecting convolutional, pooling, and fully connected layers.
- Typical CNN architecture patterns (e.g., Conv -> ReLU -> Pool -> Conv -> ReLU -> Pool -> Flatten -> Dense).
- **Key Concepts in CNN Training:**
- **Data Augmentation:** Techniques to artificially expand dataset size and improve generalization (rotation, flipping, zooming, shifting).
- **Batch Normalization:** Accelerating training and stabilizing gradients.

- **Regularization for CNNs:** Dropout revisited.

Module: Advanced CNN Architectures (Leveraging Pre-Pre-trained Models)

Evolution of CNN Architectures (as we discussed discussed earlier):

- **LeNet:** Historical significance, foundational concepts.
- **AlexNet:** Breakthrough, ReLU, dropout, GPU utilization.
- **ZFNet:** Visualization techniques.
- **VGG16/VGG19:** Simplicity, depth, small kernels.
- **GoogleNet (Inception):** Inception modules, computational efficiency.
- **ResNet:** Residual connections, overcoming vanishing gradients in very deep networks.
- *Possibly:* DenseNet, MobileNet (for efficiency).

Transfer Learning with Pre-trained Models:

Concept:

Reusing models trained on huge datasets (like ImageNet).

Why it's powerful:

Saves computation, requires less data, faster training.

Feature Extraction:

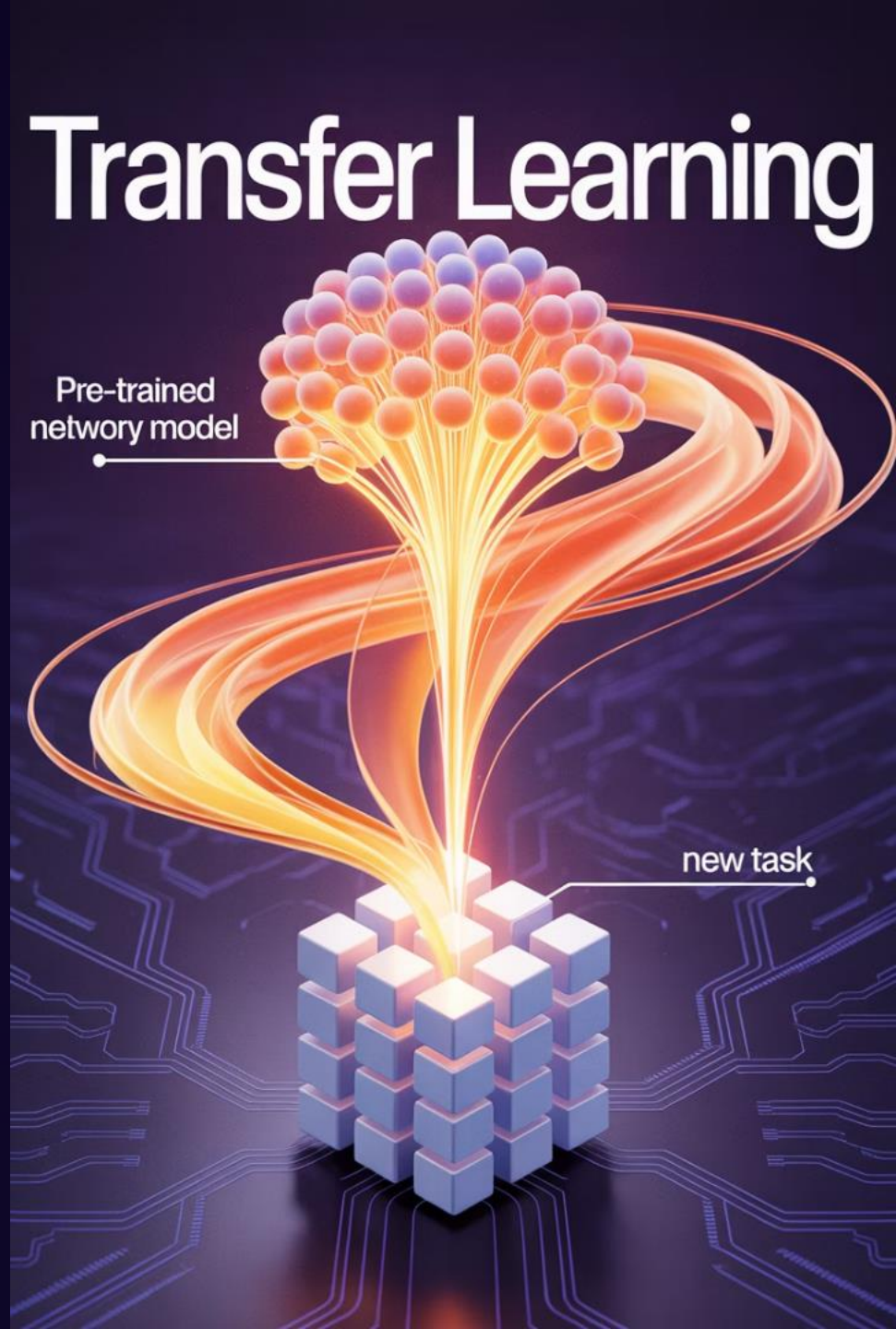
Using the pre-trained model's convolutional base as a fixed feature extractor.

Fine-tuning:

Unfreezing layers and retraining with a small learning rate.

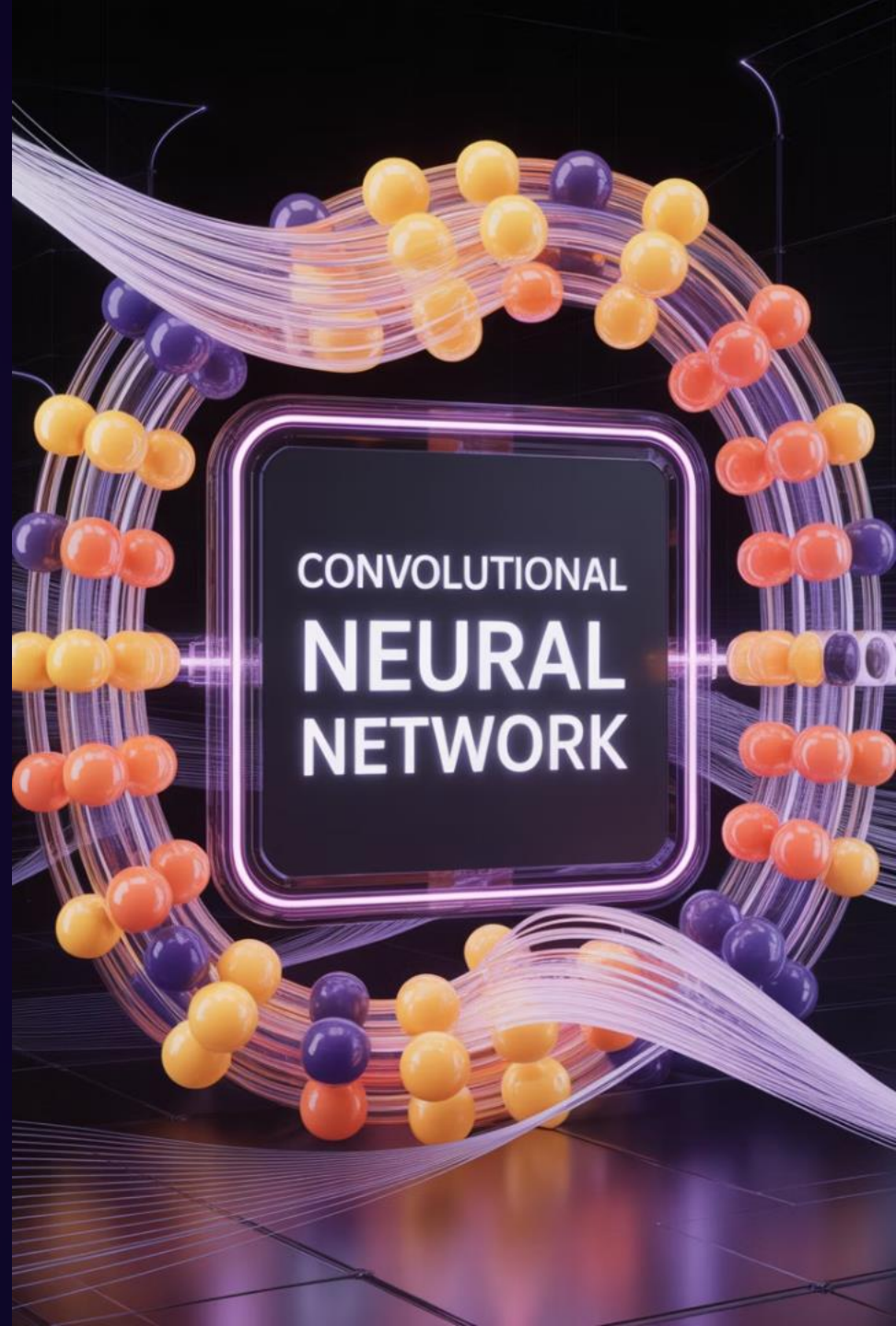
Practical Implementation:

Loading pre-trained models (e.g., tf.keras.applications or torchvision.models), modifying the top layers for a custom classification task.



Module: Practical CNN Projects & Applications

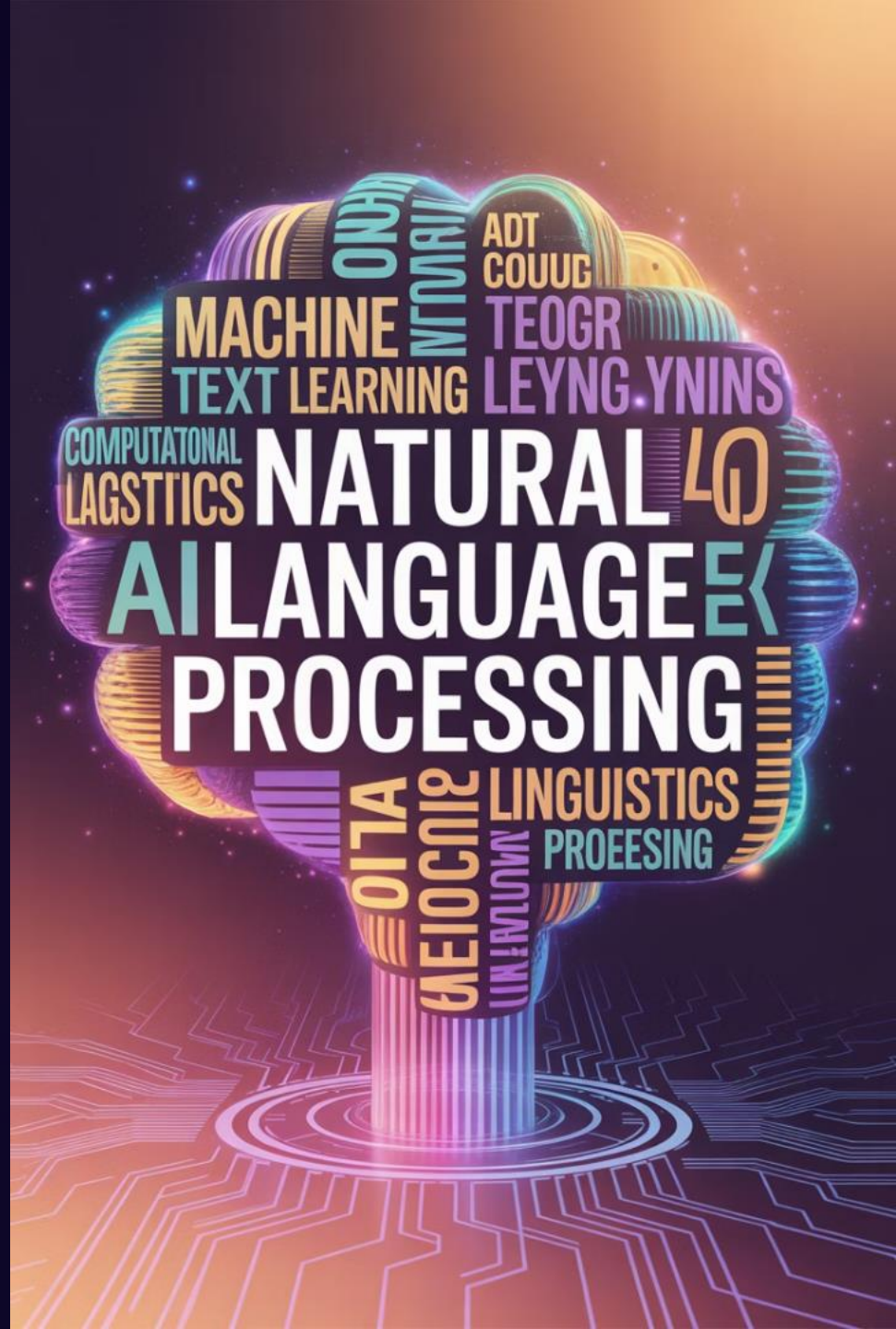
- **Image Classification Project:**
  - End-to-end project: data loading, preprocessing, model building, training, evaluation, prediction.
  - Common datasets (e.g., CIFAR-10, Fashion MNIST, custom small datasets).
- **Introduction to Object Detection (Optional, but common in full-stack DS):**
  - Bounding boxes, IoU (Intersection over Union).
  - Brief overview of algorithms like YOLO, SSD, Faster R-CNN (the course might touch upon practical implementation using pre-trained detection models or libraries like Detectron2, YOLOv5/v8).
- **Image Segmentation (Brief Introduction):**
  - Semantic vs. Instance Segmentation.
  - U-Net (common architecture for segmentation).
- **Deployment of CNN Models:**
  - Saving and loading models.
  - Creating simple web applications (e.g., with Flask/Streamlit) to showcase an image classification model.



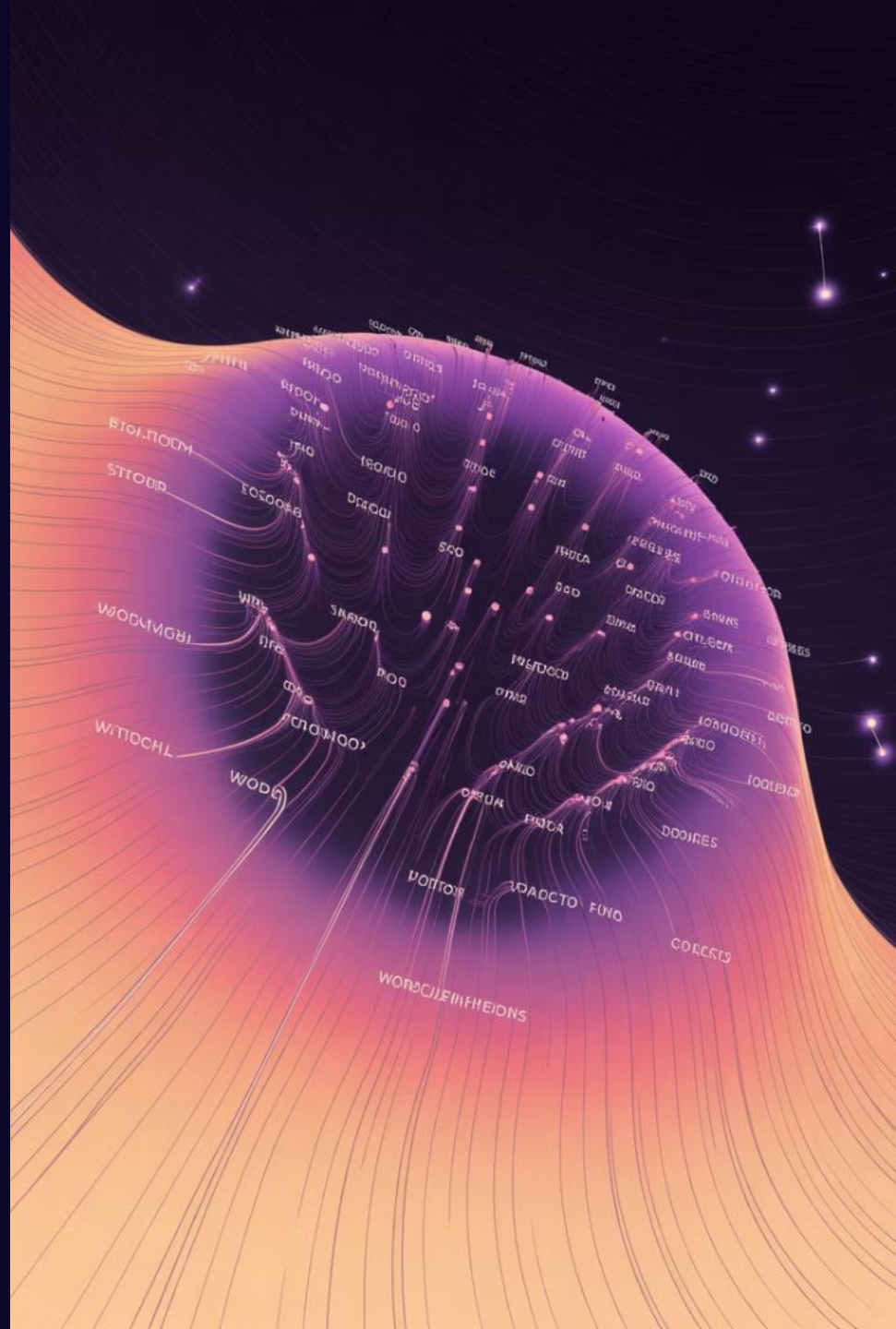
NLP

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

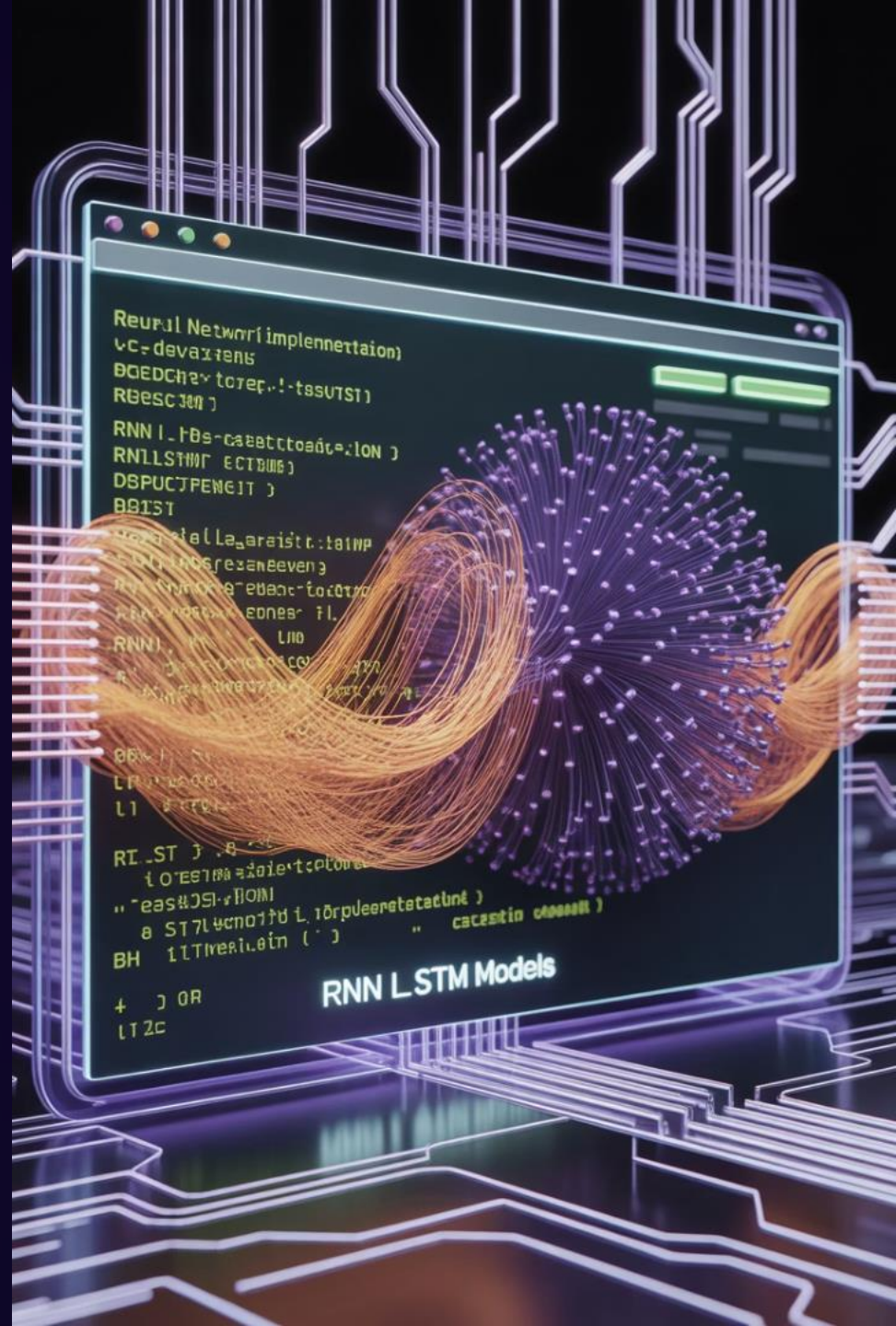
- LSTM architecture and gates.
- Advantages over traditional RNNs.

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

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

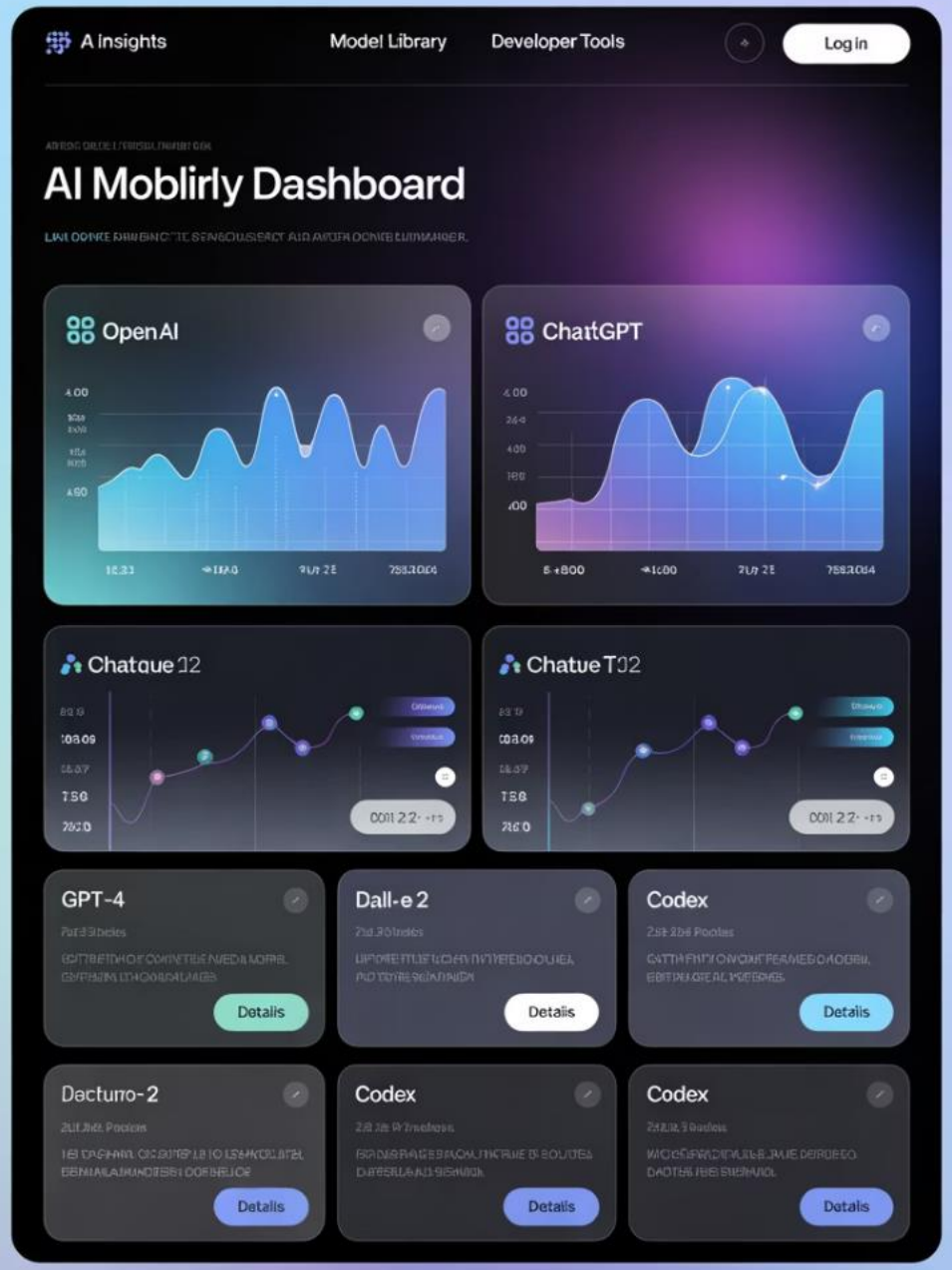
- text completion (blog creation)
- restaurant chat bot (taking orders from customer)

8) Groq

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

Project

- Text Summarization



RAG Architecture with Langchain

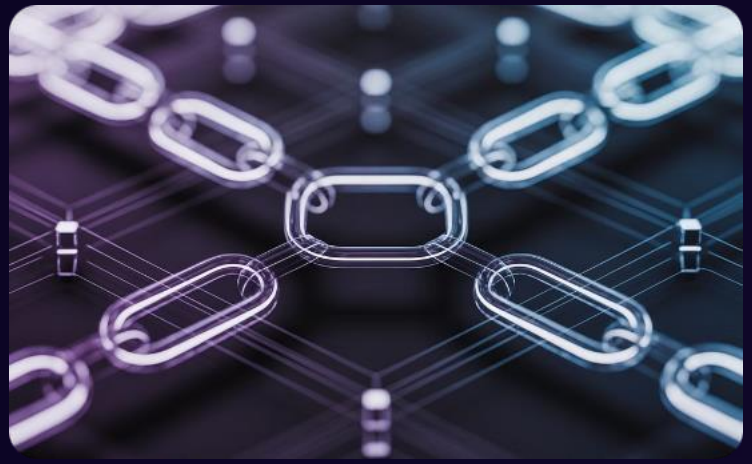
- What is langchain?
- Exploring the langchain documentation
- Flow of RAG:
  - Data Loaders(different types)
    - Analyzing CSV, PDF, and JSON Files using LangChain
  - Splitters
  - Prompt and prompt engineering
    1. What & Why of Prompt Engineering
    2. Prompt Engineering with ChatGPT Custom 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

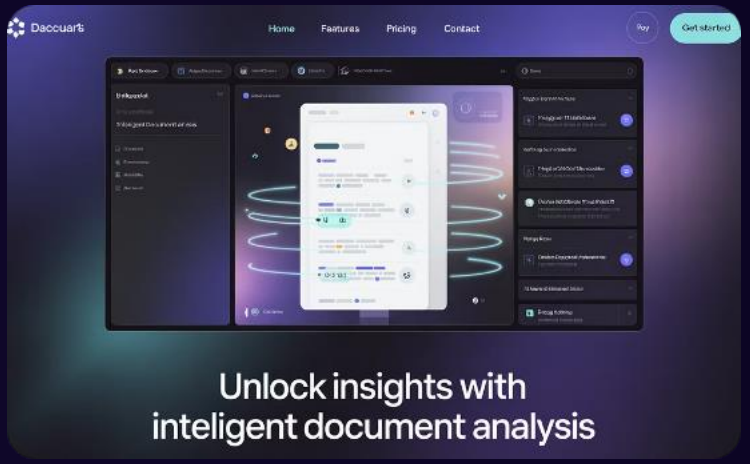
Built-in Runnables and Functions in LCEL

Combining LCEL Chains for Complex Workflows

Practical RAG Demo

Implement RAG with LCEL components

Memory integration for context retention



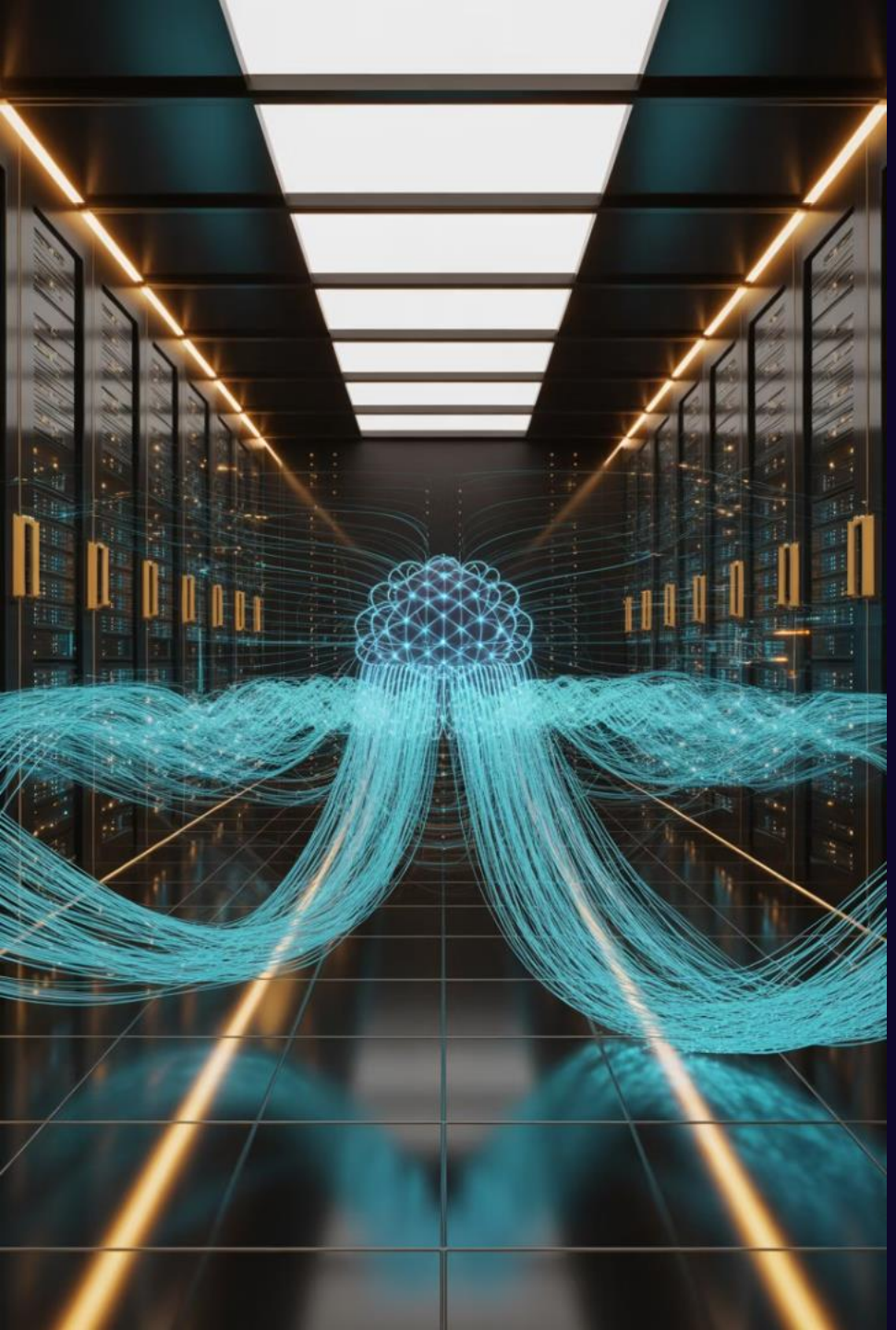
LLM Integration

Get access to any one LLM

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

Types of RAG implementations

L&L AI Solutions



AWS for Machine Learning & Deep Learning

A comprehensive 10-week curriculum to master AWS services for machine learning and deep learning workflows. From cloud fundamentals to advanced MLOps.

LS

L&L AI Solutions

AWS Foundations

Core Concepts

Learn AWS basics, account setup, and cloud computing models (IaaS, PaaS, SaaS).

Security Essentials

Master IAM for secure access with users, groups, roles, and policies.

Storage & Compute

Set up S3 buckets for datasets and EC2 instances for training.

Networking & Tools

Configure VPCs and learn AWS CLI with Boto3 SDK.

L&L AI Solutions

AWS AI Services

Pre-built AI Solutions

Explore AWS services for vision, speech, language, forecasting, and personalization.

Computer Vision

Use Amazon Rekognition for image analysis, object detection, and facial recognition.

Natural Language

Implement Amazon Comprehend for sentiment analysis and entity recognition.

Speech Services

Convert speech to text with Transcribe and text to speech with Polly.

Amazon SageMaker Basics

Environment Setup

Launch SageMaker Studio and create Jupyter notebooks for development.

Data Preparation

Process datasets using SageMaker Data Wrangler and Processing Jobs.

Model Training

Train models with built-in algorithms like Linear Learner and XGBoost.

Deployment

Deploy trained models as endpoints for real-time inference.

Deep Learning on AWS

GPU Instances Instances

Select P-series and G-series EC2 instances optimized for deep learning.

Framework Support

Use pre-configured Deep Learning AMIs with TensorFlow and PyTorch.

Custom Training Training

Bring your own code to SageMaker with framework-specific containers.

Optimization

Implement distributed training and managed spot instances for cost savings.

Advanced SageMaker Features

Hyperparameter Optimization Optimization

Find optimal model parameters automatically with SageMaker HPO.

Model Versioning

Track different model runs and configurations with experiments.

Elastic Inference

Attach GPU acceleration to CPU instances for cost-effective inference.

Distributed Training

Scale training across multiple instances using data and model parallelism.

MLOps Pipelines

1

Data Preparation

Ingest and process data with SageMaker Feature Store.

Model Training

Automate training with SageMaker Pipelines.

Model Registry

Catalog and version models for production deployment.

Monitoring

Track model performance with SageMaker Model Monitor and CloudWatch.

Real-World Projects

Fake News Detection

Build an NLP pipeline using BERT on SageMaker from data ingestion to deployment.

Image Classifier

Create a CNN-based vision solution with auto-scaling endpoints for high traffic.

Automated ML Pipeline

Set up complete MLOps workflow with SageMaker Pipelines for continuous deployment.

L&L AI Solutions