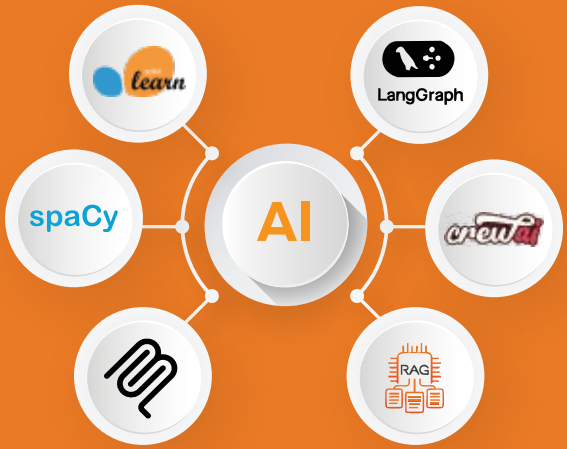
VECTOR INDIA

DRIVES YOU TO INDUSTRY



GENERATIVE AI & AGENTIC SYSTEMS

FROM MACHINE LEARNING TO PRODUCTION-GRADE AGENTIC AI SYSTEMS

Machine Learning - Core Algorithms

Supervised Learning

- Linear & Logistic Regression
- Ridge & Lasso (Regularization)
- K-Nearest Neighbors (KNN)
- Decision Trees & Random Forests
- Gradient Boosting (XGBoost)
- LightGBM
- CatBoost
- Support Vector Machines (SVM)
- Naive Bayes classifier
- Ensemble methods

Unsupervised Learning

- K-Means Clustering
- Hierarchical Clustering
- DBSCAN (density-based)
- PCA – Dimensionality Reduction
- t-SNE visualization
- Anomaly Detection methods

Model Engineering

- Bias-Variance tradeoff
- Cross-validation strategies
- Feature selection & importance
- Hyperparameter tuning (GridSearch)
- Evaluation metrics
- Model interpretability (SHAP, LIME)
- Imbalanced datasets (SMOTE)
- Pipeline building
- Saving & loading models
- Hands-on ML Mini Projects

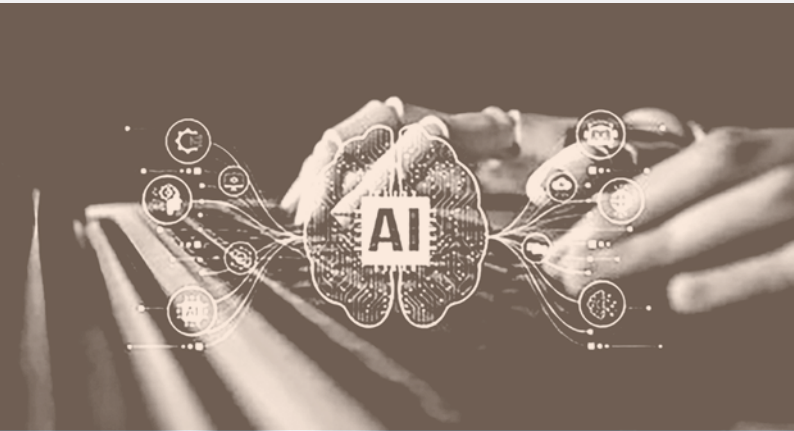
Deep Learning & Computer Vision

Neural Network Foundations

- Neurons & network architecture design
- Activation & loss functions
- Optimizers — Adam, SGD, RMSProp
- Backpropagation & gradient flow
- Overfitting, dropout & regularization
- Batch normalization

Computer Vision

- CNNs — Conv, Pool, Flatten layers
- Transfer learning (ResNet, EfficientNet)
- Object detection with YOLO



NLP, Transformers & Large Language Models

NLP Fundamentals

- Text preprocessing & tokenization
- Bag-of-Words & TF-IDF
- Word2Vec embeddings
- Sentiment analysis
- Named Entity Recognition (NER)
- Text classification

Sequence Models

- RNN, LSTM, GRU architectures
- Bidirectional RNNs
- Encoder-Decoder architecture
- Attention mechanism deep dive

Transformers & LLMs

- Transformer architecture — Self-attention, Multi-head attention, Positional encoding, Encoder, Decoder, Residual connections, Layer normalization, Feed forward networks, Masking, Tokenization Context windows .
- BERT
- GPT series — architecture & usage
- HuggingFace pipelines & models
- Prompt Engineering strategies
- Embeddings & vector search
- Gemini / Claude / GPT API integration
- Chatbot development with LLMs

Generative AI, RAG & Agentic Systems

LLM Application Foundations

- LangChain fundamentals
- Prompt Templates
- Chains & LCEL (LangChain Expression Language)
- Output Parsers & Structured Outputs
- Streaming
- LangSmith (tracing & debugging)
- Model selection strategies
- Token counting & cost optimization

Context Engineering

- Data ingestion & preprocessing
- Document chunking strategies
- Metadata enrichment
- Embeddings & semantic search
- Vector databases (FAISS, Chroma, Pinecone)
- Context window management
- LLM application architecture

Retrieval-Augmented Generation (RAG)

- RAG system design & architecture
- Retrieval techniques & hybrid search strategies
- Query understanding & query rewriting
- Prompt engineering for RAG
- Memory & conversational RAG
- Self-RAG & Graph RAG
- Agentic RAG
- RAG evaluation metrics

Agentic Systems - Core Concepts

- Foundations of Agentic AI
- Core Agent Architecture
- Tool Use / Function Calling — external APIs, Python tools, database tools, web search tools, calculator tools, file system tools
- Planning & Reasoning
- Memory Systems
- Multi-Agent Systems

Orchestration Frameworks

- LangGraph — graph architecture, state management, nodes, edges, conditional routing, cycles, checkpointing
- LangChain Agents
- CrewAI

Agent Communication Protocols & Standards

- MCP (Model Context Protocol)
- A2A (Agent-to-Agent protocol)
- Function/tool schema standards
- Agent discovery mechanisms

Autonomous Agent Design Patterns

- Supervisor pattern
- Planner–Executor pattern
- Hierarchical agents
- ReAct
- Tree of Thoughts
- Reflection & self-consistency
- Planning loops
- Deep Research agents
- Human-in-the-Loop
- Agent streaming

Evaluation, Safety & Observability

- Prompt evaluation
- Agent evaluation
- RAG evaluation
- Tracing, logging & monitoring
- AI Guardrails & Safety
- Responsible AI

MLOps & Model Deployment

Version Control & Tracking

- Git & GitHub
- Branching & pull requests
- Model registry & versioning

Model Serving & APIs

- FastAPI endpoint design
- REST API design principles
- API documentation with Swagger
- Authentication & security basics (JWT)

Frontend for AI Apps

- JavaScript fundamentals
- Linux Fundamentals
- React basics for AI-powered UIs

Containerisation & CI/CD

- **Docker fundamentals**
- **Dockerizing ML application**
- **GitHub Actions (CI/CD pipelines)**

Cloud Deployment

- **AWS (EC2, S3, RDS)**
- **Cloud deployment strategies**

Monitoring & Observability

- **Application & model logging**
- **Model performance monitoring & drift detection**
- **Basic observability tooling (Prometheus/Grafana concepts)**

Capstone Project — Industry Grade

Project Execution

- Real-world problem statement selection
- Data collection & pipeline design
- EDA & feature engineering
- Model building & evaluation
- Generative AI or Agentic component
- FastAPI backend development
- Docker containerisation
- Deployment to cloud platform

Career Preparation

- GitHub portfolio setup
- Resume preparation guidance
- Technical interview preparation
- Mock interviews with feedback

Capstone Deliverables

- A GitHub repository with clean code, README, architecture diagram, and demo instructions
- A deployed web application or API accessible via a public URL