



AI COURSE SYLLABUS

9 Month Program

PROGRAM OVERVIEW

Duration: 9 months

Target Audience: Beginners with no programming experience & professionals transitioning to AI

Course Balance: 50% Hands-on Practice, 50% Theory

Average Starting Salaries: 8 LPA for freshers, 16-22 LPA for 2 year experienced resources, 25-35 LPA for 3 to 5 year experienced resources, 36 to 55 LPA for 6 to 9 year experienced AI Engineers. 60-75 LPA for 10+ YOE candidates



Module 1: Python Programming Foundation

1.1 Python Fundamentals (Weeks 1-2)

Topics:

- Variables, data types, and expressions
- Control structures (if/else, loops)
- Functions and scope
- Basic data structures (lists, tuples, dictionaries, sets)
- File handling and I/O operations
- Error handling with try/except blocks

Libraries:

- Built-in Python libraries (os, sys, json, csv)

Why Important:

- Foundation for all AI/ML programming
- 95% of data science roles require Python
- Enables rapid prototyping and development

1.2 Data Manipulation (Weeks 3-4)

Topics:

- NumPy arrays and vectorized operations
- Pandas DataFrames and Series
- Data cleaning and preprocessing
- Merging and joining datasets
- Handling missing data
- Time series data manipulation

Libraries:

- NumPy - Numerical computing
- Pandas - Data manipulation and analysis
- Openpyxl - Excel file handling

Why Important:

- 80% of AI project time spent on data preparation
- Essential for feature engineering
- Enables efficient data processing at scale

1.3 Data Visualization (Week 5)

Topics:

- Basic plotting with Matplotlib
- Statistical visualizations with Seaborn
- Interactive visualizations
- Dashboard creation basics
- Plot customization and styling

Libraries:

- Matplotlib - Basic plotting
- Seaborn - Statistical visualizations
- Plotly - Interactive web-based visualizations

Why Important:

- Critical for exploratory data analysis
- Communicates insights to stakeholders
- Essential for model result presentation

1.4 Advanced Python for AI (Week 6)

Topics:

- Object-oriented programming
- Decorators and generators
- API integration and web scraping
- Multithreading and multiprocessing
- Package creation and distribution

Libraries:

- Requests - HTTP requests
- BeautifulSoup4 - Web scraping
- Selenium - Web automation
- AsyncIO - Asynchronous programming

Sample Projects:

- Personal Finance Tracker with Pandas
- Weather Data Analysis System
- E-commerce Data Pipeline

Module 2: SQL for Data Management (3-4 weeks | 8-12% of curriculum)

2.1 Database Fundamentals (Week 1)

Topics:

- Relational database concepts
- Table design and normalization
- Primary and foreign keys
- Basic SELECT queries
- WHERE clauses and filtering
- ORDER BY and LIMIT

Tools:

- PostgreSQL - Production database
- SQLite - Lightweight database
- MySQL - Popular open-source database
- BigQuery - Data warehouse

Why Important:

- 80% of data analyst positions require SQL
- Essential for data extraction from enterprise systems
- Foundation for data warehousing

2.2 Intermediate SQL (Week 2)

Topics:

- JOIN operations (INNER, LEFT, RIGHT, FULL)
- Aggregate functions (COUNT, SUM, AVG, MIN, MAX)
- GROUP BY and HAVING clauses
- Subqueries and CTEs
- Window functions
- Data modification (INSERT, UPDATE, DELETE)

Why Important:

- Enables complex data analysis
- Required for business intelligence
- Critical for feature engineering in ML

2.3 Advanced SQL & Python Integration (Week 3)

Topics:

- Advanced window functions (ROW_NUMBER, RANK, LAG/LEAD)
- Query optimization and indexing
- Stored procedures and triggers
- Python-SQL integration
- Big data SQL (Spark SQL basics)

Libraries:

- SQLAlchemy - Database ORM
- Pandas.read_sql() - Direct SQL to DataFrame
- PyODBC - Database connectivity
- PySpark - Distributed SQL processing

Sample Projects:

- Customer Analytics Dashboard
- Sales Performance Analysis System
- Inventory Management Database
- A/B Test Statistical Analysis

Module 3: Statistics for AI (4-6 weeks | 12-18% of weeks | 8-12% of curriculum)

3.1 Descriptive Statistics (Weeks 1-2)

Topics:

- Measures of central tendency
- Variance and standard deviation
- Quartiles and percentiles
- Data distribution shapes
- Correlation analysis
- Outlier detection methods

Libraries:

- NumPy - Statistical functions
- SciPy.stats - Advanced statistics
- Statsmodels - Statistical modeling

Why Important:

- Foundation for understanding ML algorithms
- Essential for data quality assessment
- Required for model evaluation

3.2 Probability Theory (Week 3)

- Topics:
- Basic probability rules
 - Conditional probability
 - Bayes' theorem
 - Probability distributions (Normal, Binomial, Poisson)
 - Central Limit Theorem
 - Sampling distributions

Why Important:

- Core concept for Bayesian ML
- Understanding model uncertainty
- Foundation for statistical inference

3.3 Inferential Statistics (Week 4)

- Topics:
- Hypothesis testing framework
 - t-tests, chi-square tests, ANOVA
 - p-values and significance levels
 - Confidence intervals
 - Type I and Type II errors
 - Multiple testing corrections

Libraries:

- SciPy.stats - Statistical tests
- Pingouin - Statistical analysis
- Statsmodels - Advanced statistical models

Why Important:

- A/B testing for product decisions
- Model performance validation
- Research and experimentation

- Projects:
- A/B Test Analysis Framework
 - Statistical Quality Control System
 - Customer Segmentation Analysis
 - Predictive Analytics with Confidence Intervals

Module 4: Machine Learning Algorithms

4.1 ML Fundamentals & Supervised Learning

- Topics:
- Maximum Likelihood Estimation (MLE)
 - Bayesian statistics basics
 - Cross-validation techniques
 - Bootstrapping methods
 - Feature selection with statistical tests
 - Time series analysis basics

- Topics:
- ML types (supervised, unsupervised, reinforcement)
 - Linear regression and assumptions
 - Logistic regression for classification
 - Regularization (L1/L2, Ridge, Lasso, Elastic Net)
 - Model evaluation metrics
 - Cross-validation strategies
 - Bias-variance tradeoff

Libraries:

- Scikit-learn - Core ML library
- XGBoost - Gradient boosting
- LightGBM - Fast gradient boosting
- CatBoost - Categorical feature handling

Why Important:

- Foundation for all ML applications
- Most common algorithms in industry
- Essential for data scientist roles

4.2 Advanced Supervised Learning

- Topics:
- Decision trees and pruning
 - Random forests
 - Gradient boosting machines
 - Support Vector Machines (SVM)
 - Naive Bayes classifiers
 - K-Nearest Neighbors (KNN)
 - Ensemble methods (bagging, boosting, stacking)

Why Important:

- Industry-standard algorithms
- High performance on structured data
- Wide applicability across domains

4.3 Unsupervised Learning

- Topics:
- K-means clustering
 - Hierarchical clustering
 - DBSCAN for density-based clustering
 - Gaussian Mixture Models
 - Principal Component Analysis (PCA)
 - t-SNE for visualization
 - Anomaly detection methods
 - Association rules (Apriori, FP-Growth)

- Libraries:
- Scikit-learn - Clustering and dimensionality reduction
 - HDBSCAN - Advanced clustering
 - PyOD - Outlier detection
 - MLxtend - Association rules

Why Important:

- Pattern discovery in unlabeled data
- Customer segmentation
- Fraud detection applications

4.4 Deep Learning Foundations

- Topics:
- Perceptron and neural networks
 - Backpropagation algorithm
 - Activation functions
 - Optimization algorithms (SGD, Adam, RMSprop)
 - Regularization (dropout, batch normalization)
 - Convolutional Neural Networks (CNN)
 - Recurrent Neural Networks (RNN, LSTM, GRU)

- Libraries:
- TensorFlow/Keras - Deep learning
 - PyTorch - Dynamic neural networks
 - Weights & Biases - Experiment tracking

Why Important:

- State-of-the-art performance
- Required for computer vision and NLP
- Industry demand for deep learning skills

4.5 Specialized ML Applications

- Topics:
- Natural Language Processing basics
 - Computer Vision fundamentals
 - Time series forecasting
 - Recommender systems
 - Reinforcement learning introduction
 - AutoML concepts

- Libraries:
- NLTK - Natural language toolkit
 - OpenCV - Computer vision
 - Prophet - Time series forecasting
 - Surprise - Recommender systems
 - AutoML libraries (AutoGluon, H2O.ai)

- Projects:
- Customer Churn Prediction System
 - Image Classification with CNNs
 - Sentiment Analysis Tool
 - Recommendation Engine
 - Time Series Sales Forecasting
 - Fraud Detection System
 - Multilabel Classification

Module 5: Deep Learning Frameworks

5.1 TensorFlow & Keras

- Topics:
- Tensor operations and computational graphs
 - Building models with Keras API
 - Custom layers and losses
 - Model checkpointing and callbacks
 - TensorFlow Serving for deployment
 - TensorFlow Lite for mobile
 - Distributed training strategies

Why Important:

- Industry-standard framework
- Production-ready deployments
- Google Cloud integration

5.2 PyTorch Development

Topics:

- Dynamic computation graphs
- Autograd and custom gradients
- Neural network modules
- Data loaders and transforms
- Model optimization techniques
- TorchServe for production
- PyTorch Lightning for research

Libraries:

- PyTorch - Core framework
- TorchVision - Computer vision
- TorchText - NLP utilities
- TorchAudio - Audio processing
- Torchserve - Serving

Why Important:

- Preferred for research
- Flexible architecture
- Growing industry adoption

5.3 NLP with spaCy

Topics:

- Pre-trained language models
- Named Entity Recognition (NER)
- Part-of-speech tagging
- Dependency parsing
- Custom pipeline components
- Training custom models
- Production optimization

Libraries:

- spaCy - Industrial NLP
- Transformers - State-of-the-art models
- Sentence-Transformers - Semantic search

Projects:

- Multi-class Image Classifier
- Real-time Object Detection System
- Text Generation Model
- Neural Machine Translation

Module 6: Transformers & Modern NLP

6.1 Transformer Architecture

Topics:

- Self-attention mechanism
- Multi-head attention
- Positional encoding
- Encoder-decoder architecture
- BERT for understanding
- GPT for generation
- T5 for text-to-text

Libraries:

- Transformers (Hugging Face) - Pre-trained models
- Datasets - NLP datasets
- Tokenizers - Fast tokenization

Why Important:

- Foundation of modern NLP
- Powers ChatGPT, Claude, etc.
- Industry standard for language tasks

6.2 Fine-tuning & Applications

Topics:

- Transfer learning strategies
- Parameter-efficient fine-tuning (LoRA, QLoRA)
- Adapter modules
- Prompt engineering
- Few-shot learning
- Model evaluation metrics

Why Important:

- Cost-effective model customization
- Domain-specific applications
- Rapid prototype development

6.3 Advanced NLP Applications

Topics:

- Question answering systems
- Document summarization
- Named entity recognition

- Sentiment analysis at scale
- Multi-lingual models
- Model distillation

Projects:

- Custom Chatbot with Context
- Document Analysis System
- Sentiment Analysis Dashboard
- Multi-language Translation Service

Module 7: Generative AI & RAG Systems

7.1 Generative Model Foundations

Topics:

- Variational Autoencoders (VAE)
- Generative Adversarial Networks (GAN)
- Diffusion models
- Stable Diffusion basics
- DALL-E and Midjourney concepts
- Evaluation metrics for generation

Libraries:

- Diffusers - Diffusion models
- StyleGAN - Image generation
- OpenAI API - GPT integration

Why Important:

- Fastest growing AI segment
- High commercial value
- Creative AI applications

7.2 Large Language Model Fine-tuning

Topics:

- Full fine-tuning strategies
- Parameter-efficient methods (LoRA, QLoRA, PEFT)
- Instruction tuning
- RLHF concepts
- Data preparation for fine-tuning
- Bias mitigation techniques

Libraries:

- PEFT - Parameter-efficient fine-tuning
- BitsAndBytes - Quantization
- Accelerate - Distributed training

Why Important:

- Customizes models for specific use cases
- Reduces computational costs
- Improves model performance

7.3 RAG (Retrieval Augmented Generation)

Topics:

- Vector database concepts
- Embedding models
- Semantic search implementation
- Document chunking strategies
- Context window management
- Hybrid search approaches
- RAG evaluation metrics

Libraries:

- LangChain - RAG framework
- ChromaDB - Vector database
- Pinecone - Cloud vector database
- FAISS - Similarity search
- Weaviate - Vector search engine

Why Important:

- Reduces hallucinations
- Enables real-time information
- Cost-effective AI solutions

7.4 Advanced Prompt Engineering

Topics:

- Chain-of-thought prompting
- Few-shot learning optimization
- Prompt templates and versioning
- Automatic prompt optimization
- Multi-modal prompting
- Agent-based prompting

Sample Projects:

- E-commerce Chatbot with RAG
- Legal Document Processor
- Code Generation Assistant
- Content Creation Platform

Module 8: Agentic AI

8.1 Agent Development Kit (ADK)

Topics:

- Agent architecture patterns
- Tool integration and management
- State management
- Memory systems
- Planning and reasoning
- Multi-agent orchestration
- Evaluation frameworks

Libraries:

- LangChain Agents - Agent framework
- AutoGen - Multi-agent conversations
- CrewAI - Agent collaboration

Why Important:

- Autonomous problem solving
- Complex workflow automation
- Future of AI applications

8.2 Model Context Protocol (MCP)

Topics:

- MCP architecture
- Server development
- Client integration
- Security implementation
- Resource management
- Tool exposure patterns

Tools:

- MCP SDK - Protocol implementation
- JSON-RPC - Communication protocol

Why Important:

- Industry standard emerging
- Microsoft Copilot integration
- Anthropic Claude support

8.3 Agent-to-Agent Communication

Topics:

- Message passing protocols
- Task distribution
- Shared context management
- Consensus mechanisms
- Error handling
- Recovery strategies

Why Important:

- Scalable AI systems
- Complex problem decomposition
- Enterprise automation

8.4 Production Agent Systems

Topics:

- Agent deployment strategies
- Monitoring and observability
- Performance optimization
- Cost management
- Security considerations
- Compliance frameworks

Projects:

- Multi-Agent Trading System
- Customer Service Orchestra
- Development Team Simulation

Module 9: : MLOps & Cloud Platforms

9.1 Infrastructure & Containerization

Topics:

- Docker for ML applications
- Container optimization
- Kubernetes basics
- Microservices architecture
- Infrastructure as Code
- CI/CD pipelines

Tools:

- Docker - Containerization
- Kubernetes - Orchestration
- Terraform - Infrastructure as Code
- GitHub Actions - CI/CD

Why Important:

- Production deployment essential
- Scalability requirements
- Industry best practices

9.2 Experiment Tracking & Model Management

Topics:

- Experiment tracking systems
- Model versioning
- A/B testing frameworks
- Model registry concepts
- Metadata management
- Reproducibility strategies

Libraries:

- MLflow - Experiment tracking
- DVC - Data version control
- Neptune.ai - Experiment management
- Weights & Biases - ML tracking

Why Important:

- Reproducible research
- Model governance
- Collaboration enablement

9.3 Orchestration Frameworks

Topics:

- Pipeline design patterns
- Workflow orchestration
- Data pipeline integration
- Error handling and retry logic
- Monitoring and alerting
- Resource optimization

Tools:

- Apache Airflow - Workflow orchestration
- Kubeflow - ML pipelines
- Kedro - Data pipeline framework
- Prefect - Modern workflow automation

Why Important:

- Automated ML workflows
- Production reliability
- Operational efficiency

9.4 Google Cloud Platform

Topics:

- Vertex AI end-to-end platform
- BigQuery ML
- Cloud Run deployment
- GKE for ML workloads
- Cloud Storage integration
- Dataflow for streaming

Why Important:

Services:

- Vertex AI - Unified ML platform
- BigQuery ML - SQL-based ML
- Cloud Run - Serverless deployment
- GKE - Kubernetes engine

Why Important:

- Industry-leading platform
- Integrated ML services
- Enterprise scalability

9.5 AWS Machine Learning (Optional)

- Topics:
- SageMaker end-to-end
 - AWS Batch for training
 - Lambda for inference
 - Kinesis for streaming ML
 - S3 for data storage
 - ECR for container registry

- Services:
- SageMaker - Complete ML platform
 - AWS Lambda - Serverless inference
 - Amazon ECR - Container registry
 - AWS Batch - Batch computing

- Why Important:
- Market leader in cloud
 - Comprehensive ML services
 - Wide industry adoption

9.6 Production Monitoring

- Topics:
- Model performance monitoring
 - Data drift detection
 - Concept drift handling
 - A/B testing in production
 - Cost optimization
 - Security best practices

- Tools:
- Evidently AI - ML monitoring
 - Grafana - Visualization
 - Prometheus - Metrics collection
 - WhyLabs - ML observability

- Projects:
- End-to-End ML Pipeline
 - Real-time Recommendation System
 - Computer Vision Production System
 - NLP Service Architecture

Module 10: : Professional Skills & Industry Integration

10.1 AI Patent Filing (Week 1)

- Topics:
- Patent law fundamentals
 - AI patentability criteria
 - Prior art search techniques
 - Claim drafting for AI
 - Patent portfolio strategy
 - International filing considerations

- Tools:
- Google Patents - Prior art search
 - USPTO resources - Filing guidelines
 - PatentScope - International search

- Why Important:
- Intellectual property protection
 - Career differentiation
 - Innovation documentation

10.2 Research Paper Writing

- Topics:
- Academic writing structure
 - Literature review methodology
 - Statistical reporting standards
 - Visualization for papers
 - LaTeX for formatting
 - Peer review process

- Tools:
- LaTeX/Overleaf - Paper formatting
 - Mendeley/Zotero - Reference management
 - ArXiv - Preprint submission

Why Important:

- Knowledge dissemination
- Career advancement
- Industry credibility

10.3 Technical Documentation

Topics:

- API documentation standards
- Model cards creation
- README best practices
- Tutorial writing
- Video documentation
- Documentation automation

Tools:

- Sphinx - Documentation generation
- MkDocs - Project documentation
- Swagger - API documentation

Why Important:

- Professional communication
- Team collaboration
- Open source contribution

10.4 AI Ethics & Governance

Topics:

- Ethical AI principles
- Bias detection and mitigation
- Fairness metrics
- Regulatory compliance (GDPR, CCPA)
- AI governance frameworks
- Responsible AI practices

Certifications Available:

- AI Ethics Specialist
- AI Governance Professional
- Responsible AI Practitioner

Projects:

- Patent Portfolio Analysis
- Research Paper Submission
- Complete API Documentation
- AI Ethics Framework Development

Capstone Project Requirements

Project Scope

- End-to-end AI application
- Real-world problem solving
- Production deployment
- Documentation and presentation

Technical Requirements

- Data collection and preprocessing
- Model development and training
- API development
- Cloud deployment
- Monitoring setup
- Performance optimization

Deliverables

- GitHub repository
- Technical documentation
- API documentation
- Deployment guide
- Video presentation
- Business impact analysis

Capstone Project Requirements

Target Roles

- AI Engineer
- Machine Learning Engineer
- Data Scientist
- Technical Architect

Job Preparation Support

- Resume optimization
- Portfolio development
- Mock interviews
- Salary negotiation
- Industry networking
- Certification guidance - GCP PMLE

Success Metrics

- 80%+ job placement rate
- 6-month placement timeline
- 35% average salary increase
- 90% student satisfaction