

# AI/ML Roadmap

This is a roadmap to learn AI/ML and land an internship in around 6 months. The timeline may vary depending on your current knowledge, learning pace, and consistency.

## Stage 1: Foundations

1. Learn the basics of Python programming. You don't need to master it, but you should feel confident writing code and debugging simple errors. Cover variables, constants, lists, dictionaries, loops, conditionals, functions, class and objects, basic OOP concepts, exception handling, file handling, working with modules, and basic understanding of standard libraries.
2. Also, learn to use Jupyter Notebooks for experimentation, and get comfortable with Git and GitHub for version control and collaboration.
3. Basic math and statistics (Course content is enough, for revision you can check vectors, metrics, dot product, matrix multiplication, mean, variance, std, normal distribution, conditional probability, partial derivative)

## Stage 2: Core Libraries

1. Learn NumPy basics. A 1-hour crash course is enough to understand arrays, indexing, broadcasting, and vectorized operations.
2. Dive deeper into pandas – learn dataframes, indexing, filtering, grouping, merging, missing data handling, etc.
3. Learn matplotlib basics for data visualization: line plots, bar plots, histograms, scatter plots.
4. (Optional) Learn seaborn for better-looking plots and statistical visualizations.

## Stage 3: Dealing with Data/EDA

1. Learn data preprocessing and cleaning techniques – handling missing values, encoding categorical variables, normalizing data, dealing with outliers, and feature scaling.
2. Understand data visualization principles and tools to explore patterns and distributions.
3. Learn how to split data for training and testing.
4. Practice working with real-world datasets from Kaggle, UCI, or open data portals.

## Stage 4: Core Machine Learning Algorithms and Evaluation (Sklearn library)

1. Learn supervised ML algorithms – linear regression, logistic regression, K-nearest neighbors, decision trees, random forest, support vector machines.
2. Learn ensemble methods – bagging, boosting, voting classifiers (Random Forest, XGBoost, AdaBoost).
3. Explore unsupervised learning – k-means clustering, hierarchical clustering, PCA.
4. Understand model evaluation metrics – accuracy, precision, recall, F1-score, confusion matrix, ROC-AUC.
5. Learn concepts like cross-validation, bias-variance tradeoff, overfitting, underfitting, regularization (L1, L2).
6. Make small projects using these algorithms, build UIs using Streamlit or Gradio, and publish them on GitHub with proper README files.

[Note: Mastering this stage could take quite long time as there are simply too many topics to learn, but you don't have to master all algorithms, make projects using 2-3 algorithms, and have introductory knowledge of them and move on, but since you'll be using sklearn library throughout this stage, you must get comfortable using it in any scenario)

## **Stage 5: Deep Learning**

1. Understand the basics of neural networks – architecture, activation functions, forward propagation, backward propagation, weight initialization, loss functions, gradient descent.
2. Learn PyTorch or TensorFlow – focus on defining custom models, training loops, loss and optimizer usage, and evaluation.
3. Get a high-level understanding of CNNs and RNNs. (You don't need to train heavy CNN or RNN models; give 2–3 days to understand how they work, their components, use cases, and limitations.)

### **4.1. Computer Vision (CV path):**

If you want to focus on computer vision, learn image processing concepts and go deeper into CNNs (convolutions, pooling, feature maps, common architectures).

### **4.2. Natural Language Processing (NLP path):**

Explore transformers and deeply understand the attention mechanism. Read and break down the “Attention Is All You Need” paper (must for NLP). Also understand RNN/LSTM at a conceptual level.

5. Optionally train and fine-tune basic models based on your chosen path:

CV: small CNN models or transfer learning with pre-trained CNNs

NLP: fine-tune transformer models using the HuggingFace Transformers library

## **Stage 6: Applied AI**

1. Learn about Retrieval-Augmented Generation (RAG) and vector databases like FAISS or Chroma.
2. Understand LangChain or LlamaIndex for building document-aware chatbots.
3. Explore agentic workflows (e.g., ReAct, tool-using LLM agents).
4. Learn to work with HuggingFace – use pre-trained models for NLP, vision, and more.
5. Explore embedding models like sentence-transformers and OpenAI embeddings.
6. Build full-stack AI projects like PDF Q&A bots, resume analyzers, or content recommenders.
7. Deploy apps using Streamlit, FastAPI, Docker, and if possible, cloud platforms like AWS or GCP.
8. Polish GitHub projects, write blog posts or documentation, create a strong AI-focused resume, and apply for internships on LinkedIn, Internshala, AI-focused Discord communities, or company pages.

## **Additional Skills (Other than AI/ML)**

1. Git and Github (Must)
2. FastAPI
3. virtualenvs
4. Basic SQL
5. Docker basics