📅 Your 25-Day ML Expert Plan

🧱 Phase 1: Python + NumPy Core (Day 1–3)

Learn only what's needed to write ML algorithms manually.

✅ Day 1 – Core Python for ML (No BS)

- Loops, Functions, List comprehensions
- `zip`, `enumerate`, `lambda`, `map`, `filter`
- `random`, `math`, `collections.Counter`
- Practice: Write a `mean`, `median`, `mode` function

✅ Day 2 – NumPy Essentials

- Arrays, shapes, slicing
- Matrix multiplication
- Broadcasting, axis operations
- Practice: Implement dot product manually vs with NumPy

✅ Day 3 – Math for ML (Minimal)

- Mean, Variance, Std Dev
- Linear equations
- Derivatives, Gradients (basic intuition)
- Implement `mean squared error` from scratch

⚙️ Phase 2: ML Algorithms from Scratch (Day 4–14)

✅ Day 4 – What is ML + Your First Model

- Supervised vs Unsupervised
- What is model, training, loss
- Implement: Linear regression manually (Gradient Descent)

✅ Day 5 – Linear Regression Deep Dive

- Cost function, convergence
- Multiple variables (Multivariate Regression)
- Implement from scratch

✅ Day 6 – Classification vs Regression

- Difference between tasks
- Accuracy, Precision, Recall, F1 Score
- Implement: Accuracy manually

✅ Day 7 – Logistic Regression from Scratch

- Sigmoid function, Binary classification
- Cost function, Gradient Descent
- Implement manually

✅ Day 8 – Decision Trees (No scikit)

- Gini Index, Information Gain
- Recursive tree building
- Implement manually

✅ **Day 9 – K-Nearest Neighbors (KNN)**

- Euclidean distance
- Choosing K, Scaling
- Implement from scratch

✅ **Day 10 – Naive Bayes**

- Probabilities, Bayes Theorem
- Text classification (spam vs ham)
- Implement manually

✅ **Day 11 – Support Vector Machine (Basic)**

- Concept of margin and hyperplane
- Linear SVM implementation

✅ **Day 12 – Unsupervised Learning**

- Clustering basics
- Implement: K-means from scratch

✅ **Day 13 – Dimensionality Reduction**

- PCA Intuition + Math (Eigenvectors)
- Implement simple PCA manually (with NumPy)

✅ **Day 14 – Regularization & Overfitting**

- L1 vs L2
- Bias-variance tradeoff

- Update Linear Regression with L2

🚀 Phase 3: Projects & Real-World Work (Day 15–20)

✅ Day 15 – Real Dataset #1 (CSV, Manual Workflow)

- Use a CSV dataset (like housing price or diabetes)
- Manually preprocess (handle missing, normalize)
- Run your regression model

✅ Day 16 – Real Dataset #2 (Text Data)

- Preprocess text manually (tokenize, lowercase)
- Implement a basic spam filter (Naive Bayes)

✅ Day 17 – Real Dataset #3 (Image Basics)

- Load images as arrays (NumPy)
- Flatten, Normalize
- Basic image classifier with KNN

✅ Day 18 – Manual Cross-Validation

- Implement k-fold manually
- Compare models on folds
- Write reusable cross_val_score function

✅ Day 19 – Data Visualization (Only Useful Ones)

- Use matplotlib: scatter, line, hist

- Use seaborn: heatmap, pairplot
- Plot model accuracy, loss curve

✅ **Day 20 – Build a Tiny ML Framework**

- Write reusable functions:
    - fit(), predict(), score()
    - train_test_split(), normalize(), plot_confusion_matrix()

🧠 Phase 4: Advanced Thinking (Day 21–25)

✅ **Day 21 – Intro to Neural Networks**

- Perceptron, activation, forward pass
- Manually build a neural net with 1 hidden layer

✅ **Day 22 – Gradient Descent Variants**

- Batch, Stochastic, Mini-batch
- Update your models to support mini-batch

✅ **Day 23 – Intro to Optimization & Regularization**

- Adam, RMSProp (conceptual only)
- Dropout concept (apply to your own network manually)

✅ **Day 24 – Read & Understand a ML Research Paper**

- Pick one simple research paper
- Break it down, replicate a result

✅ **Day 25 – Expert Mindset + Road to Deployment**

- Learn how to evaluate model robustness
- How to deploy a model (intro only)
- What to learn next: Sklearn, XGBoost, Deep Learning, MLOps

🗃️ You'll Need:

- Jupyter Notebook or Google Colab
- Datasets from: Kaggle, UCI
- Markdown for notes & explanations

✅ Deliverables By End:

- 6+ Models from scratch
- 3 Real Dataset Projects
- Your own mini-ML framework
- Strong theoretical + practical foundation
- Ready to jump into scikit-learn or PyTorch confidently