3-Month AI/ML Engineer Roadmap for FAANG Interviews

July 26, 2025

Introduction

This roadmap is designed to prepare you for AI/ML Engineer interviews at top-tier companies like Meta, Google, and Amazon in three months. It assumes basic Python knowledge and high school-level math (calculus, linear algebra, probability). The plan covers AI, Machine Learning (ML), and Generative AI (Gen AI), emphasizing theory, math derivations, and hands-on implementation. Expect to commit 15–20 hours weekly, with daily study (theory, math, coding) and weekend projects. Resources are free, including texts and video lectures, with practice tasks to build skills for coding, ML theory, and system design interviews.

1 Month 1: Foundations

1.1 Week 1: Python for ML and Math Foundations

Goal: Master Python for ML and refresh math prerequisites.
Topics:
- Python: NumPy, Pandas, Matplotlib, Scikit-learn basics.
- Math: Overview of calculus (derivatives, integrals), linear algebra (vectors, matrices), probability basics.

Tasks:
- Learn Python libraries for data science.
- Understand math concepts intuitively for ML.
- Code simple data manipulation tasks (e.g., Kaggle's Titanic dataset).

Resources:
- *Python for ML*:
  - Text: Python Data Science Handbook (NumPy, Pandas, Matplotlib).
  - Video: CS231n Python Tutorial (Stanford, $\sim$1 hour).
- *Math Overview*:
  - Text: Mathematics for Machine Learning (Chapters 1–3).
  - Video: Essence of Linear Algebra by 3Blue1Brown ($\sim$2 hours).

Practice: Implement NumPy matrix multiplication and Pandas data cleaning.

1.2 Week 2: Linear Algebra and Calculus for ML

Goal: Deep dive into linear algebra and calculus with ML applications.
Topics:
- Linear Algebra: Vectors, matrices, eigenvalues/eigenvectors, SVD.
- Calculus: Gradients, partial derivatives, chain rule, gradient descent.

Tasks:

- Understand matrix operations for data representation.
- Derive gradient descent from scratch.
- Code matrix operations and simple optimization.

Resources:
- *Linear Algebra*:
  - Text: Mathematics for Machine Learning (Chapter 4).
  - Video: 3Blue1Brown Linear Algebra Series (~2–3 hours).
- *Calculus*:
  - Text: Mathematics for Machine Learning (Chapter 5).
  - Video: Essence of Calculus by 3Blue1Brown (~3 hours).

Practice: Implement gradient descent for $f(x) = x^2$ using NumPy.

1.3 Week 3: Probability and Statistics for ML

Goal: Master probability and statistics for ML models.
Topics:
- Probability: Random variables, distributions (normal, binomial), Bayes' theorem.
- Statistics: Mean, variance, hypothesis testing, maximum likelihood estimation.

Tasks:
- Understand probabilistic foundations (e.g., Naive Bayes).
- Derive maximum likelihood estimation.
- Simulate distributions in Python.

Resources:
- *Probability and Statistics*:
  - Text: Introduction to Probability by Joseph K. Blitzstein (Chapters 1–5).
  - Video: StatQuest Probability Playlist (~3 hours).

Practice: Simulate a normal distribution and implement a Naive Bayes classifier.

1.4 Week 4: Introduction to Machine Learning

Goal: Understand ML basics and key algorithms.
Topics:
- ML Overview: Supervised vs. unsupervised learning, bias-variance tradeoff.
- Algorithms: Linear regression, logistic regression, k-NN.
- Metrics: MSE, accuracy, precision, recall.

Tasks:
- Derive linear regression's cost function.
- Implement algorithms using Scikit-learn.

Resources:
- *ML Basics*:
  - Text: An Introduction to Statistical Learning (Chapters 1–3).
  - Video: CS229 Machine Learning by Andrew Ng (~5 hours).

Practice: Implement linear regression on Boston Housing dataset (Kaggle), compute MSE.

2 Month 2: Core ML Algorithms and Implementation

2.1 Week 5: Supervised Learning (Regression, Classification)

Goal: Master supervised learning algorithms.
Topics:
- Regression: Linear, polynomial, ridge/lasso.

- Classification: Logistic regression, SVM, decision trees, random forests.
- Math: Derive SVM hinge loss.

Tasks:

- Implement algorithms from scratch and with Scikit-learn.
- Understand overfitting and regularization.

Resources:

- *Supervised Learning*:
  - Text: ISLR (Chapters 4–6).
  - Video: StatQuest Machine Learning Playlist (∼4 hours).

Practice: Build a random forest classifier on Titanic dataset. Derive logistic regression's gradient descent.

2.2 Week 6: Unsupervised Learning and Evaluation Metrics

Goal: Learn unsupervised learning and model evaluation.
Topics:

- Clustering: K-means, hierarchical clustering.
- Dimensionality Reduction: PCA, t-SNE.
- Metrics: Silhouette score, ROC-AUC, confusion matrix.

Tasks:

- Derive PCA's eigenvalue decomposition.
- Implement k-means and PCA in Python.

Resources:

- *Unsupervised Learning*:
  - Text: ISLR (Chapter 10).
  - Video: StatQuest Clustering and PCA (∼2 hours).

Practice: Apply PCA on MNIST dataset and visualize results. Compute ROC-AUC for a classifier.

2.3 Week 7: Neural Networks and Deep Learning Basics

Goal: Understand neural networks and deep learning foundations.
Topics:

- Perceptrons, multilayer perceptrons (MLPs).
- Activation functions (sigmoid, ReLU), backpropagation.
- Math: Derive backpropagation.

Tasks:

- Implement a neural network from scratch.
- Use PyTorch/TensorFlow for basic models.

Resources:

- *Neural Networks*:
  - Text: Deep Learning Book by Goodfellow et al. (Chapters 6–8).
  - Video: CS231n Neural Networks (∼4 hours).

Practice: Code a 2-layer MLP in PyTorch for MNIST. Derive backpropagation equations.

2.4 Week 8: Optimization and Regularization in ML

Goal: Master optimization and regularization techniques.
Topics:

- Optimization: SGD, Adam, learning rate schedules.
- Regularization: L1/L2, dropout, batch normalization.
- Math: Derive SGD updates.

Tasks:

- Experiment with optimizers in PyTorch.
- Compare regularization techniques.

Resources:

- *Optimization and Regularization*:
  - Text: Deep Learning Book (Chapters 7–8).
  - Video: CS231n Optimization ($\sim$2 hours).

Practice: Train a neural network with SGD vs. Adam on CIFAR-10. Compare L2 vs. dropout.

3 Month 3: Advanced Topics, Gen AI, and Interview Prep

3.1 Week 9: Deep Learning Architectures (CNNs, RNNs, Transformers)

Goal: Learn advanced deep learning models.
Topics:

- CNNs: Convolution, pooling, VGG, ResNet.
- RNNs: LSTM, GRU for sequential data.
- Transformers: Attention mechanism, BERT.
- Math: Derive attention mechanism.

Tasks:

- Implement CNN and RNN in PyTorch.
- Fine-tune a pre-trained transformer (Hugging Face).

Resources:

- *Deep Learning Architectures*:
  - Text: Deep Learning Book (Chapters 9–10).
  - Video: CS231n CNNs and CS224n NLP ($\sim$5 hours).

Practice: Build a CNN for CIFAR-10 and an LSTM for IMDb text classification.

3.2 Week 10: Generative AI (GANs, VAEs, Diffusion Models)

Goal: Understand generative models for Gen AI.
Topics:

- GANs: Generator, discriminator, adversarial loss.
- VAEs: Variational inference, reparameterization trick.
- Diffusion Models: Forward/reverse process, denoising.
- Math: Derive GAN loss and VAE ELBO.

Tasks:

- Implement a simple GAN/VAE in PyTorch.
- Explore diffusion model tutorials.

Resources:

- *Generative AI*:
  - Text: Deep Learning Book (Chapter 20).
  - Video: Stanford Generative Models ($\sim$3 hours).

Practice: Train a GAN for MNIST digits. Explore Hugging Face diffusion tutorials.

3.3 Week 11: ML System Design and Practical Implementation

Goal: Learn ML system design and production-level implementation.
Topics:

- ML System Design: Data pipelines, model serving, scalability.
- Tools: TensorFlow Serving, ONNX, Docker.

- Practical Issues: Data drift, model monitoring.

Tasks:

- Design a recommendation system pipeline.
- Deploy a model using Flask or FastAPI.

Resources:

- *ML System Design*:
  - Text: Designing Machine Learning Systems by Chip Huyen (Excerpts online).
  - Video: ML System Design by DeepLearning.AI (∼2 hours).

Practice: Deploy a Scikit-learn model with Flask. Simulate a data pipeline for a Kaggle dataset.

3.4 Week 12: Interview Preparation (Coding, Theory, Projects)

Goal: Prepare for FAANG AI/ML interviews.

Topics:

- Coding: LeetCode (arrays, trees, dynamic programming).
- ML Theory: Explain algorithms, trade-offs, math derivations.
- Projects: Build 2–3 portfolio projects.
- System Design: End-to-end ML system questions.

Tasks:

- Solve 50–70 LeetCode medium/hard problems.
- Practice explaining ML concepts.
- Complete portfolio projects (GitHub).

Resources:

- *Coding*:
  - Text: LeetCode (Free problems).
  - Video: NeetCode (∼5 hours for key problems).
- *ML Theory*:
  - Text: Machine Learning Interviews by Chip Huyen (Free excerpts).
  - Video: AI Interview Prep by DeepLearning.AI (∼2 hours).

Practice: Build a CNN-based image classifier and NLP sentiment analyzer. Practice system design questions.

4 Weekly Schedule and Tips

Daily (2–3 hours):

- 1 hour: Study theory (videos/texts).
- 1 hour: Derive math for key concepts.
- 0.5–1 hour: Code or solve LeetCode problems.

Weekends (4–6 hours):

- Work on Kaggle competitions or portfolio projects.
- Review/debug code, experiment with hyperparameters.

Tips:

- *Math*: Derive equations (e.g., backpropagation) by hand in a notebook.
- *Coding*: Use GitHub for projects. Write clean, modular code.
- *Interviews*: Practice coding (LeetCode medium/hard), ML theory (e.g., explain SVM math), and system design (e.g., scale a model).
- *Portfolio*: Build 2–3 projects (image classification, NLP) and host on GitHub.
- *Community*: Engage on X or Kaggle forums for updates and doubts.

5 Final Notes

This intensive roadmap covers essentials for AI/ML Engineer roles. Adjust the pace if needed, but aim to complete all topics. For xAI's API services, visit x.ai/api. For X Premium subscriptions, check help.x.com. Focus on understanding, practice explaining concepts aloud, and good luck with your interviews!