AI/ML Engineer Roadmap

A Practical, Step-by-Step Guide to Becoming a Job-Ready AI/ML Engineer

- ***By Akber Shaikh*** 📺 *YouTube* 💼 *LinkedIn*

Important Advice

Before you start this roadmap, make one decision: **You're not here to 'try' — you're here to transform**. Most people watch, feel inspired, then quit when it gets uncomfortable. But you're not most people. **You've already chosen to learn a skill that 99% only dream about**. So write this down and say it out loud: '**I finish what I start. I don't chase motivation — I build momentum**. Every step I take, even the hard ones, is proof that I'm becoming the person I promised myself I'd be.' Now take a breath… and take the first step.

Divide AI/ML Learning into 5 Stages

1. **Core CS**
2. **Data Handling**
3. **Foundational Math**
4. **Core ML**
5. **Advanced ML**

1️⃣ Core CS

Python

Why: Best language for AI/ML.

What:

- Basic syntax, indentation, comments
- Variables, loops, if-else
- Data types: strings, lists, dictionaries
- Functions, classes, modules

Time: 3–4 weeks (with 3–4 hrs daily)

Resource: https://www.coursera.org/specializations/python

DSA (Data Structures & Algorithms)

Why: Must for interviews + efficient coding.

What:

- Arrays, Strings, Hashmaps
- Recursion, Sorting, Searching
- Trees, Graphs, Heaps
- Dynamic Programming
- Time & Space Complexity

Where: Solve 100–150 problems on LeetCode

Time: 6–9 weeks

Resource: https://youtu.be/pkYVOmU3MgA?si=GENkXsAYiuG-l55b

Git

Why: Version control + teamwork.

What: Git basics, GitHub workflows, branching, pull requests

Time: 1 week

Resource: https://git-scm.com/book/en/v2

APIs

Why: ML systems interact via APIs.

What:

- API basics (REST, GraphQL, Webhooks)
- API calls, authentication, error handling
- Build your APIs using **FastAPI/Flask**

Time: 2–4 weeks

Resources:
https://www.youtube.com/watch?v=Z1RJmh_OqeA&ab_channel=freeCodeCamp.org

https://www.youtube.com/watch?v=WXsD0ZgxjRw&ab_channel=freeCodeCamp.org

---

Cloud Platforms

Why: Most AI/ML jobs demand cloud experience. AWS is the best.

What:

- IaaS vs PaaS vs SaaS

- EC2, Lambda, S3
- SageMaker, Vertex AI
- Deploying ML models on the cloud

Time: 3–4 weeks

Resource: https://skillbuilder.aws/

Practise CS skills with Mini Project: Weather Info App

- Use OpenWeatherMap API
- Basic search, conversions
- Expose your API
- Git for version control
- Deploy on AWS (EC2 or Lambda)

2️⃣ Data Handling Skills

SQL

Why: Work with real-world data.

What: SELECT, JOIN, GROUP BY, CTEs, Indexing

Time: 3–4 weeks

Resource: https://www.khanacademy.org/computing/computer-programming/sql/sql-basics/v/welcome-to-sql

---

Pandas

Why: Clean, explore, transform data.

Focus: DataFrames, groupby, aggregations, missing values

Time: 2–3 weeks

Resource: https://www.youtube.com/watch?v=vmEHCJofslg&ab_channel=KeithGalli

---

NumPy

Why: Fast, efficient math operations.

Focus: Arrays, vectorized ops, reshaping, random gen

Time: 2–3 weeks

Resource: https://www.youtube.com/watch?v=QUT1VHiLmmI&ab_channel=freeCodeCamp.org

---

Data Visualization

Why? Understand messy data through visuals.

Tools: Matplotlib, Seaborn

Time: 1–2 weeks

Resource:

Matplotlib**:** https://www.youtube.com/watch?v=wB9C0Mz9gSo&ab_channel=DerekBanas

Seaborne: https://www.youtube.com/watch?v=6GUZXDef2U0&ab_channel=DerekBanas

---

Data Handling Project

Tips:

- Choose an interesting dataset
- Clean & transform with Pandas
- Visualize correlations
- Share findings via graphs/slides

3️⃣ Maths (Algebra, Stats, Probability, Calculus)

Why: Understand how models learn + evaluate.

What:

- Statistics & Probability
- Linear Algebra
- Calculus

Time: 6–8 weeks

Resource:

Statistics & Probability : https://www.khanacademy.org/math/statistics-probability

Linear Algebra: https://www.khanacademy.org/math/linear-algebra

Calculus: https://www.khanacademy.org/math/calculus-1

All in one: https://mml-book.github.io/

4️⃣ Core ML Concepts & Algorithms

Why: Heart of ML

What:

- Linear/Logistic Regression
- Decision Trees, Random Forest
- KNN, Naive Bayes
- SVM, Clustering (K-Means)
- PCA
- Cross Validation, Overfitting/Underfitting
- Gradient Descent
- Evaluation Metrics: Accuracy, Precision, F1, ROC
- Scikit-Learn

Time: 4–6 weeks

Resource:
ML: https://developers.google.com/machine-learning/crash-course

Scikit-Learn:
https://www.youtube.com/watch?v=pqNCD_5r0IU&ab_channel=freeCodeCamp.org

---

Machine Learning Project

Steps:

1. EDA with Pandas
2. Hypothesize with domain knowledge
3. Try multiple algorithms
4. Use validation/test split
5. Submit on Kaggle
6. Analyze → Improve → Repeat

5️⃣ Advanced Topics (Optional)

Only when needed.

What:

- Deep Learning: CNNs, RNNs, Transformers
- Optimization: Regularization, Hyperparams
- Deployment: Scaling, APIs
- Research Reading

Resource: https://developers.google.com/machine-learning/advanced-courses

---

Mistakes to Avoid

Don’t Memorize

Understand **why** and **when** to use algorithms.

Don’t Get Stuck in Tutorial Hell

- Do **1–2 tutorials per topic**
- Then build **real projects**

Collaborate & Share

- Coding buddies
- Share on communities
- Get feedback

How to Land a High-Paying Job

Top 5 Recruiter Priorities (Based on Data):

1. **Real-World Projects & Portfolio**
2. **Problem Solving & DSA Skills**
3. **Internships or Practical Experience**
4. **Communication & Business Understanding**
5. **Certifications & Education** (Optional but helpful)

Make the most of the Roadmap

To fully understand how to use this roadmap effectively, you **MUST** watch my video:

👉 [AI/ML Engineer Roadmap]

In the video, I explain:

- Why I chose these specific skills
- What common mistakes do beginners make?
- How to avoid wasting time or burning out

Share the Knowledge

If this guide helps you, **please share it with anyone** trying to break into the tech industry. You might change someone's life.
