

[Sign in](#)[Get started](#)[Follow](#)

602K Followers

Editors' Picks

Features

Deep Dives

Gi

You have **2** free member-only stories left this month. [Sign up for Medium and get an extra one](#)

# How To Get A Machine Learning Job In 6 Months

Follow this study guide to be successful in a Machine Learning interview at top tech companies.



Stefan Hosein Mar 8 · 7 min read ★

I have given and took many Machine Learning Engineering (MLE) interviews at companies like Google, Twitter, Lyft, Snapchat and others. Since I have been on both sides of the interviewing process for MLE, I can tell you that unlike software engineering, the interview process for MLE is chaotic. By chaotic, I mean every company has their idea of what skills a candidate needs to have; thus, every company has a different hiring process. However, I will give you a template to follow so that by the end you will have

the knowledge to pass almost any MLE interview at FAANG and other tech companies.



Photo by [Christina @ wocintechchat.com](#) on [Unsplash](#)

One good thing that is happening within the last year or so in the field now is that most companies are (slowly) converging on a somewhat similar process:

*[Some people consider the call with the recruiter the first step, but in all honesty, unless you lied on your resume or say something offensive, this is just a chat about the next steps]*

1. Phone Interview — This usually would take either two forms: a complete software engineering interview or a mix of software engineering and a few ML questions

2. Onsite Interview:

- Software Engineering/ML Coding Interview
- ML Theory Questions — I have seen this interview split into two parts: ML theory and ML coding
- ML System Design
- Company Specific — I put this here because most companies tend to have one of the above interviews repeated or something specific based on the job (e.g. Facebook has a System Design interview, note this is not an ML System Design interview)
- Hiring Manager (behavioral) — I failed an interview for a company by being unprepared for this round. I aced the technical interviews and was confident I would get an offer. However, when I did not, the feedback was that I did not do well on the behavioral interview. After that feedback, I spent a lot of time preparing for these questions.

So we can see that there are three main rounds that you will encounter in almost any MLE interview:

1. Software Engineering/ML Coding
2. ML Theory
3. ML System Design

We will go through each of these rounds and I will provide you with the best (IMO) websites/documents that you can use to succeed in your next MLE interview in 6 months. Since we have 6 months we can build a strong baseline of knowledge and then branch out.

## Software Engineering & ML Coding

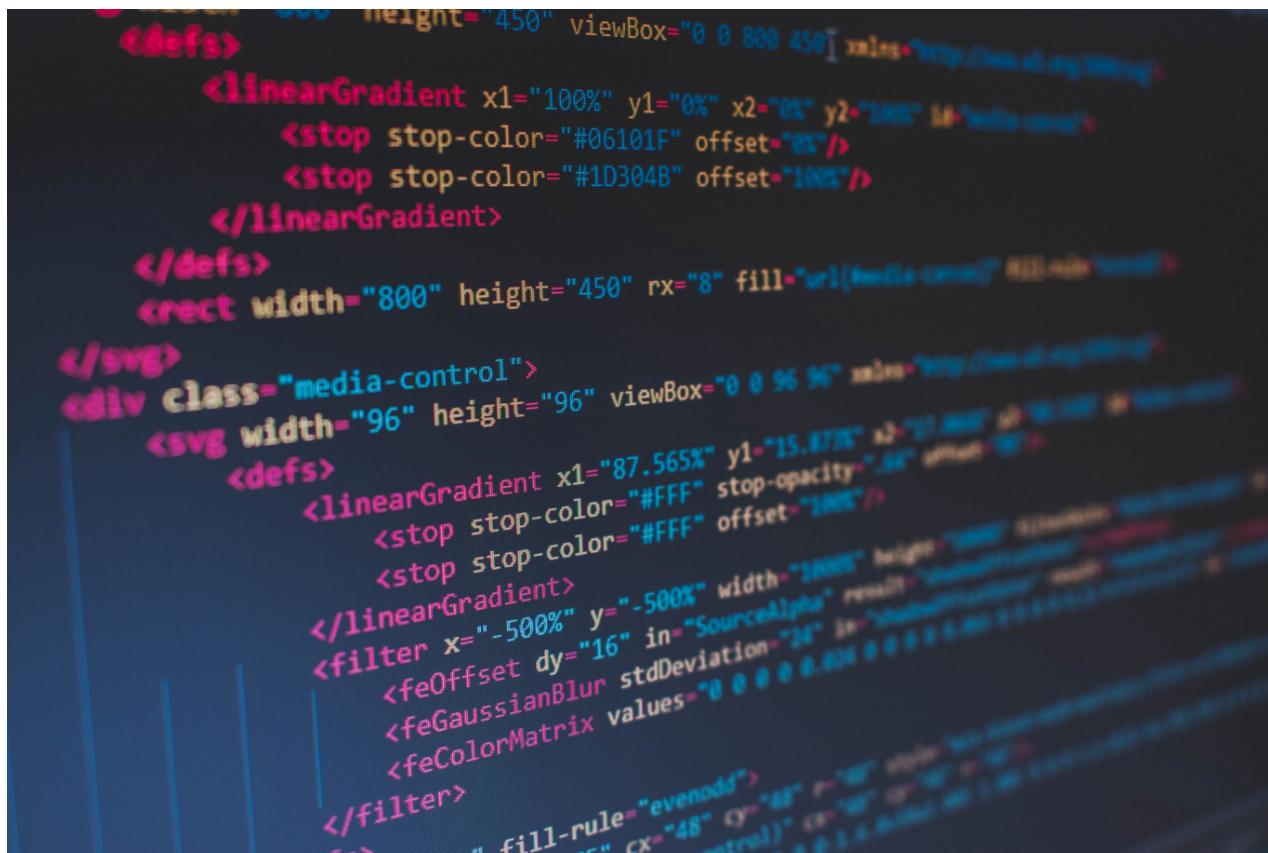


Photo by [Florian Olivo](#) on [Unsplash](#)

We can split this into two sections: pure software engineering (SWE) coding and then ML coding.

For the SWE coding, the plan would be:

1. **Leetcode**: This should not be a strange inclusion as it is ubiquitous for SWE interviews and for good reason! I would recommend following the below steps in order as they build on each other:
  - [2 weeks] <https://leetcode.com/explore/learn/> — this is good to refresh your memory on data structures and

algorithms

- [2 weeks] <https://seanprashad.com/leetcode-patterns/> — this is a great resource containing many patterns that you would face in a coding interview
- [2 weeks] <https://leetcode.com/problemset/top-interview-questions/> — you should be aware of the popular questions that tend to come up in an interview
- N.B: While answering these coding questions, you should create a spreadsheet of difficult questions and the algorithm (not the code) that is used. Then go back to the spreadsheet later on and try to solve the questions a second time. This will help you understand your weaknesses. Once you complete the three links above, you should answer at least one leetcode question every day (preferably medium and sometimes hard). When it is closer to your interview date, I recommend buying leetcode premium and doing the company questions.

2. [1 1/2 months — optional but recommended] [Elements of Programming Interviews](#): Many people recommend [Cracking the Coding Interview](#) but, I would recommend you buy [Elements of Programming Interviews](#) (EPI). I make this recommendation because you can choose between Java, C++ or Python as your programming language and can use their coding [judge system](#) to test your code. This book was hard for me to get through at first and I could not solve most of the questions. However, as I

continued, I started writing python code efficiently and changed how I think about solving problems.

## For ML Coding, the plan would be:

1. [1 week] ML Coding Algorithms: I would recommend this site because getting proper ML coding resources became an issue for me when studying for interviews. Most websites are lacking in two ways: the code is far too complex than what is required in an interview or does not contain time/space complexity. This website solves both problems.
2. [1 week] Github ML Algorithms: This resource contains almost all of the possible ML algorithms you will probably encounter in an interview. The code is written in a very modular way, making it is easy to understand the code.
3. [2 weeks — optional] Machine Learning Mastery: A popular website containing the most popular ML algorithms that the author builds from scratch. It is optional because there are parts of the code that may not be relevant in an interview (e.g. data ingestion). If you have extra time, you can browse the code.

## ML Theory

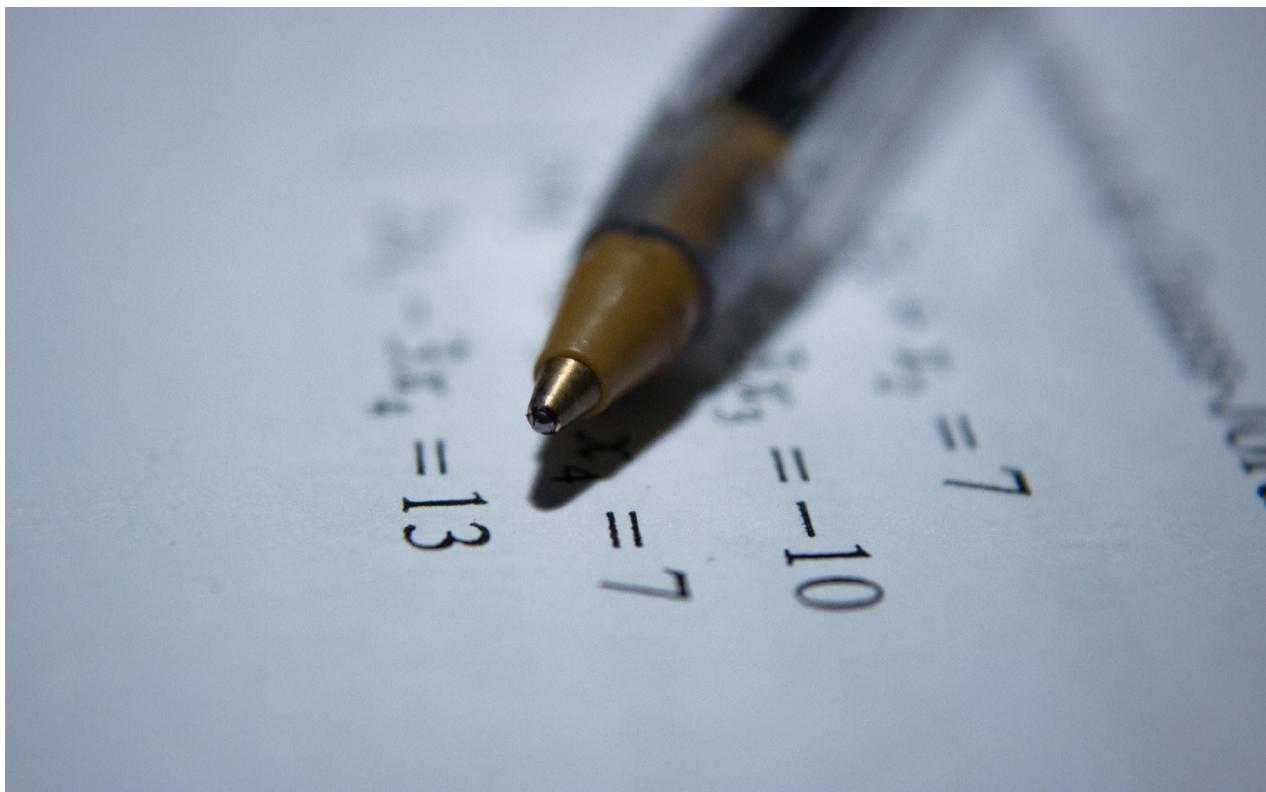


Photo by [Antoine Dautry](#) on [Unsplash](#)

I found myself spending hours upon hours trying to understand what topics I should know for these theory interviews. After countless hours of research, these are the topics you should know:

### **“Foundational” ML Topics (topics you HAVE to know):**

- Linear Regression
- Logistic Regression
- KNN
- Tree-based models- decision trees, random forests, bagged/ boosted trees, KL Divergence, Entropy)

- SVM
- Clustering (basic ones like K-means, Hierarchical) & EM
- PCA (SVD & Eigen Values)
- Naive Bayes
- Maximum Likelihood Estimation, Maximum A Priori Estimation

**Specialized ML Topics (you should know at least Neural Network Architectures; for the other topics, they can be reviewed based on your resume and what the job specifies):**

- Neural Network Architectures: FFNNs, RNNs, LSTMs, CNNs
- GANs
- NLP
- Computer Vision
- Recommendation Algorithms
- Information Retrieval
- Reinforcement Learning

**Misc (these are VERY popular questions which you should know):**

- Activation Functions
- Optimization schemes SGD, Adam, RMSProp...
- Losses — log-loss, hinge loss, Huber loss, L1/2 loss...
- Metrics — Precision, Recall, Intersection over Union, F1, RMSE...
- Feature Selection
- Feature Engineering

For studying, I will give two options based on your preferred method of studying (note, you can mix and match if you would like):

## Books:

1. **[3/4 weeks]** [Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow](#): This book is an excellent book that covers most of the topics above (other than some specialized topics). It is an easy read with engaging diagrams and code to help you understand each topic.
2. **[1/2 weeks]** [The 100 Page ML Book](#): This is another great book that covers most of the topics above. It goes into more mathematical detail than the Hands-On ML book but, the author does an excellent job of being concise while still being an easy read. Also, it is free :)

## (OR) Video Courses:

1. [1 week] [Udacity's Machine Learning Course](#): An easy but comprehensive course on ML. The instructors give you an overview of most of all the foundational topics and some of the misc topics. It is an excellent refresher if you are looking for somewhere to start.
2. [2/3 weeks] [Coursera's Machine Learning Course](#): This has to be the most popular course on the internet for machine learning and for good reason! It goes into more of the mathematical foundations of many of the traditional ML algorithms while still being accessible to anyone. Note, if you want a summary of the course rather than watching all the videos you can check out [these notes](#).
3. [4/5 weeks] [Coursera's Deep Learning Specialization](#): These courses cover most of the neural network architectures above and go into sufficient detail for an interview. I highly recommend this as it has almost all the information you need for succeeding in questions on Neural Networks and deep learning. Again, if you would like a summary of the course you can read [these notes](#).

Once you are finished with the above resources, you probably need questions to practice. Here is a collection of question and answer docs:

1. <https://github.com/andrewekhalel/MLQuestions>
2. [https://github.com/IndyNYU/Machine-Learning-Interview/blob/master/Algorithms %26 Theory](https://github.com/IndyNYU/Machine-Learning-Interview/blob/master/Algorithms%26Theory)
3. <https://www.mle-interviews.com/ml-questions>

## ML System Design

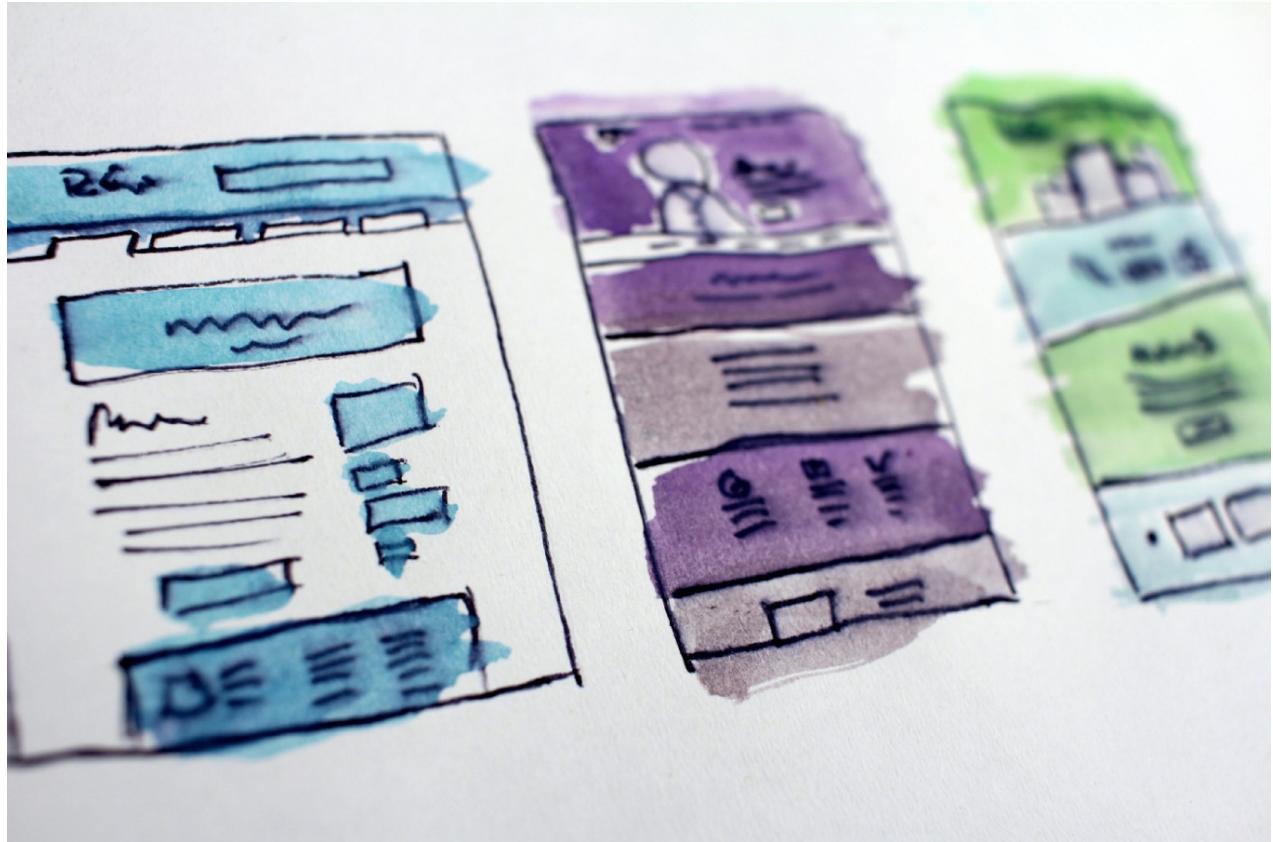


Photo by [HalGatewood.com](#) on [Unsplash](#)

An ML system design question is (almost) a given for any MLE onsite interview. Since there are no courses or books on this as of this writing, we should use blogs (we can complete all the below

in 2 weeks)

1. ML System Design Template For MLE Interviews: A general framework that can be used for most ML System Design interview questions. There are specific questions and answers that most companies will ask.
2. Architecting a Machine Learning Pipeline: A blog for understanding the engineering side for ML System Design
3. Machine Learning System Design Draft PDF: A good overview of themes that you should include in your response to ML System Design
4. Machine Learning System Design Blog: A great resource of links to many company blogs which explain how they build their ML systems.

Wow, that was a lot! I never said studying for MLE position would be easy but, if you enjoy ML, this should be enjoyable. A final recommendation is to take notes, whether it be online or using the old fashioned pen and paper and don't forget those leetcode questions once a day!

---

## Sign up for The Variable

By Towards Data Science

Every Thursday, the Variable delivers the very best of Towards Data Science: from hands-on tutorials and cutting-edge research to original features you don't want to miss. [Take a look.](#)

[Get this newsletter](#)

You'll need to sign in or create an account to receive this newsletter.

Machine Learning

Computer Science

Programming

Interview

Editors Pick

About

Help

Legal