Log In Join

Back To Module Home

Machine Learning System Design

0% completed

Search Module

Machine Learning Primer

- [ ] Introduction
- [ ] Feature Selection and Feature Engineering
- [ ] Training Pipeline
- [ ] Inference
- [ ] Metrics Evaluation

Video Recommendation

- [ ]

Feed Ranking

- [ ]

Ad Click Prediction

Rental Search Ranking

Estimate Food Delivery Time

Machine Learning Knowledge

Machine Learning Model Diagnosis

Conclusion

Mark Module as Completed

Introduction

Learn how to approach Machine Learning System Design.

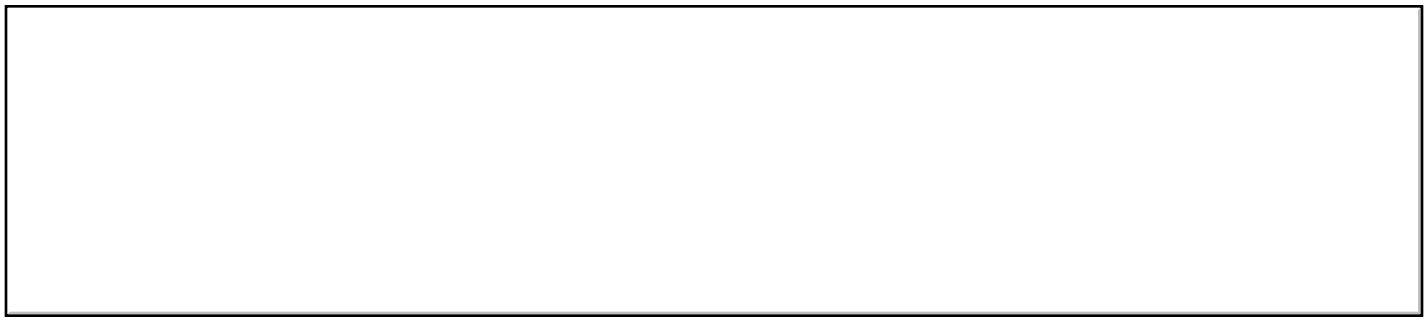
We'll cover the following

1. What should you expect in a machine learning interview?#

- Most major companies, i.e. Facebook, LinkedIn, Google, Amazon, and Snapchat, expect Machine Learning engineers to have solid engineering foundations and hands-on Machine Learning experiences. This is why interviews for Machine Learning positions share similar components with interviews for traditional software engineering positions. The candidates go through a similar method of problem solving (Leetcode style), system design, knowledge of machine learning and machine learning system design.
- The standard development cycle of machine learning includes data collection, problem formulation, model creation, implementation of models, and enhancement of models. It is in the company's best interest throughout the interview to gather as much information as possible about the competence of applicants in these fields. There are plenty of resources on how to train machine learning models and how to deploy models with different tools. However, there are no common guidelines for approaching machine learning system design from end to end. This was one major reason for designing this course.

2. How will this course help you?#

In this course, we will learn how to approach machine learning system design from a top-down view. It's important for candidates to realize the challenges early on and address them at a structural level. Here is one example of the thinking flow.



The 6 basic steps to approach Machine Learning System Design

Problem statement#

It's important to state the correct problems. It is the candidates job to understand the intention of the design and why it is being optimized. It's important to make the right assumptions and discuss them explicitly with interviewers. For example, in a LinkedIn feed design interview, the interviewer might ask broad questions:

Design LinkedIn Feed Ranking.

Asking questions is crucial to filling in any gaps and agreeing on goals. The candidate should begin by asking follow-up questions to clarify the problem statement. For example:

- Is the output of the feed in chronological order?
- How do we want to balance feeds versus sponsored ads, etc.?

If we are clear on the problem statement of designing a Feed Ranking system, we can then start talking about relevant metrics like user agreements.

Identify metrics#

During the development phase, we need to quickly test model performance using offline metrics. You can start with the popular metrics like `logloss` and `AUC` for binary classification, or `RMSE` and `MAPE` for forecast.

Identify requirements#

- Training requirements
    - There are many components required to train a model from end to end. These components include the data collection, feature engineering, feature selection, and loss function. For example, if we want to design a YouTube video recommendations model, it's natural that the user doesn't watch a lot of recommended videos. Because of this, we have a lot of negative examples. The question is asked:

How do we train models to handle an imbalance class?

Once we deploy models in production, we will have feedback in real time.

How do we monitor and make sure models don't go stale?

- Inference requirements

  Once models are deployed, we want to run inference with low latency (<100ms) and scale our system to serve millions of users.

How do we design inference components to provide high availability and low latency?

Train and evaluate model#

- There are usually three components: feature engineering, feature selection, and models. We will use all the modern techniques for each component.
- For example, in Rental Search Ranking, we will discuss if we should use ListingID as embedding features. In Estimate Food Delivery Time, we will discuss how to handle the latitude and longitude features efficiently.

Design high level system#

In this stage, we need to think about the system components and how data flows through each of them. The goal of this section is to identify a minimal, viable design to demonstrate a working system. We need to explain why we decided to have these

components and what their roles are.

- For example, when designing Video Recommendation systems, we would need two separate components: the Video Candidate Generation Service and the Ranking Model Service.

Scale the design#

In this stage, it's crucial to understand system bottlenecks and how to address these bottlenecks. You can start by identifying:

- Which components are likely to be overloaded?
- How can we scale the overloaded components?
- Is the system good enough to serve millions of users?
- How we would handle some components becoming unavailable, etc.
- You can also learn more about how companies scale there design here.

Next

Feature Selection and Feature Engin…

Mark as Completed

Report an Issue