Back To Module Home

Machine Learning System Design

0% completed

Machine Learning Primer

Video Recommendation

Feed Ranking

- Problem Statement and Metrics
- Feed Ranking Model

Feed Ranking System Design

Ad Click Prediction

Rental Search Ranking

Estimate Food Delivery Time

Machine Learning Knowledge

Machine Learning Model Diagnosis

Conclusion

Mark Module as Completed

Feed Ranking System Design

Learn about the Feed Ranking system design for the LinkedIn application.

We'll cover the following

4. Calculation & estimation#

Assumptions#

- 300 million monthly active users

- On average, a user sees 40 activities per visit. Each user visits 10 times per month.
- We have $12 * 10^{10}$ or 120 billion observations/samples.

Data size#

- Assume the click through rate is about 1% for 1 month. We collected 1 billion positive labels and about 110 billion negative labels. This is a huge dataset.
- Generally, we can assume that for every data point, we collect hundreds of features. For simplicity, each row takes 500 bytes to store.
- In one month, we need 120 billion rows. Total size: $500 * 120 * 10^9 = 60 * 10^{12}$ bytes = 60 Terabytes. To save costs we can keep the last 6 months or 1 year of data in the data lake and archive old data in cold storage.

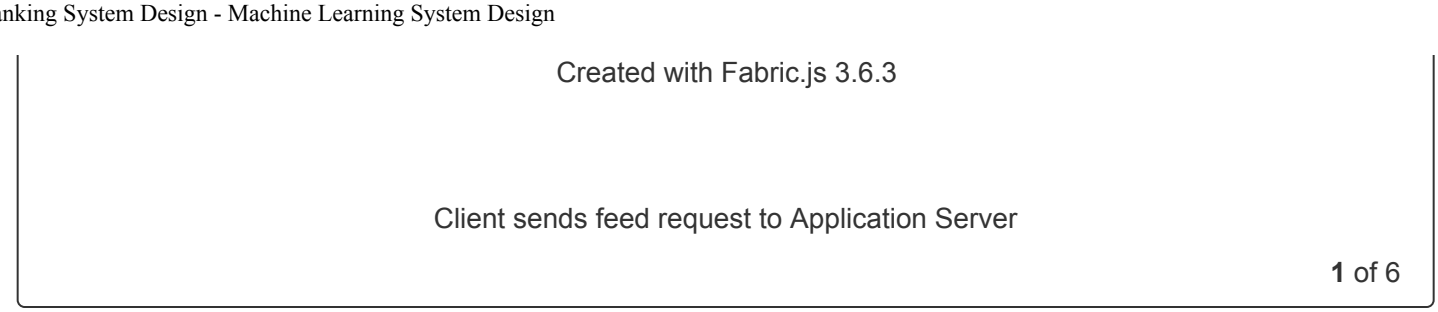
Scale#

- Supports 300 million users

5. High-level design#

- **Feature store** is feature values storage. During inference, we need low latency (<10ms) to access features before scoring. Examples of feature stores include MySQL Cluster, Redis, and DynamoDB.
- **Item store** stores all activities generated by users. It also stores models for the corresponding users. One goal is to maintain the consistent user experience, i.e., to use the same feed ranking method for any particular user. Item store provides the correct model for the corresponding users.

Let’s examine the flow of the system:



- A user visits the LinkedIn homepage and requests an Application Server for feeds. The Application Server sends feed requests to the Feed Service.
- Feed Service gets the latest model from Model Repos, gets the correct features from the Feature Store, and all the feeds from the ItemStore. Feed Service will provide features for the Model to get predictions.
- The Model returns recommended feeds sorted by click through rate likelihood.

6. Scale the design#

- Scale out the Feed Service module as it represents both Retrieval Service and Ranking Service. This provides better visualization.
- Scale out the Application Server and put the Load Balancer in front of the Application Server to balance load.

7. Summary#

- We learned how to build Machine Learning models to rank feeds. The binary classification model with custom loss function helps the model be less sensitive to background click through rate.
- We learned how to create the process to generate training data for the Machine Learning Model.
- We learned how to scale training and inference by scaling out the Application Server

and Feed Services.

- You can also learn more about how companies scale there design here.

Feed Ranking Model

Next

Problem Statement and Metrics

Mark as Completed

Report an Issue