Log In Join

Back To Module Home

Machine Learning System Design

0% completed

Search Module

Machine Learning Primer

Video Recommendation

Feed Ranking

Ad Click Prediction

- Problem Statement and Metrics
- Ad Click Prediction Model
- Ads Recommendation System Design

Rental Search Ranking

Estimate Food Delivery Time

Machine Learning Knowledge

Machine Learning Model Diagnosis

Conclusion

Mark Module as Completed

Ad Click Prediction Model

Learn about the Ad prediction model architecture.

We'll cover the following

3. Model#

Feature engineering#

Features	Feature engineering	Description

AdvertiserID	Use Embedding or feature hashing	It's easy to have millions of advertisers
User's historical behavior, i.e., numbers of clicks on ads over a period of time.	Feature scaling, i.e., normalization	
Temporal: time_of_day, day_of_week etc	One hot encoding	
Cross features	Combine multiple features	See example in the Machine Learning System Design Primer

Training data#

Before building any ML models we need to collect training data. The goal here is to collect data across different types of posts while simultaneously improving the user experience. As you recall from the previous lesson about the waterfall model, we can collect a lot of data about ad clicks. We can use this data for training the Ad Click model.

We can start to use data for training by selecting a period of data: last month, last six months, etc. In practice, we want to find a balance between training time and model accuracy. We also downsample the negative data to handle the imbalanced data.

Model#

Selection#

- We can use deep learning in distributed settings. We can start with fully connected layers with the Sigmoid activation function applied to the final layer. Because the

CTR is usually very small (less than 1%), we would need to resample the training data set to make the data less imbalanced. It's important to leave the validation and test sets intact to have accurate estimations about model performance.

Evaluation#

- One approach is to split the data into training data and validation data. Another approach is to replay evaluation to avoid biased offline evaluation. Assume the training data we have up until time t. We use test data from time $t+1$ and reorder their ranking based on our model during inference. If there is an accurate click prediction, we record a match. The total match will be considered as total clicks.
- During evaluation we will also evaluate how big our training data set should be and how frequently we need to retrain the model among many other hyperparameters.

Problem Statement and Metrics

Next

Ads Recommendation System Design

Mark as Completed

Report an Issue