

[Log In](#)

[Join](#)

[Back To Module Home](#)

Machine Learning System Design

0% completed

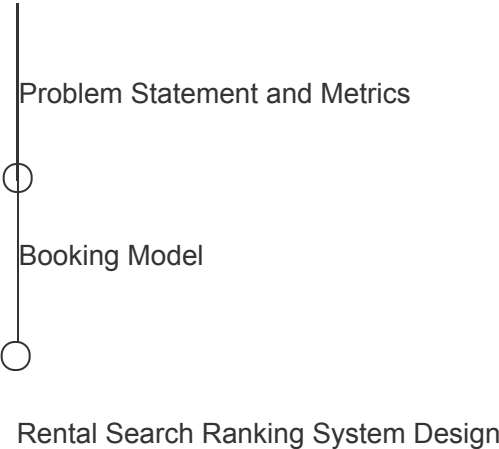
Machine Learning Primer

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

Problem Statement and Metrics

Let’s dive into the problem statement and metrics required for the Airbnb rental search ranking application.

We'll cover the following

- Airbnb rental search ranking
- 1. Problem statement
- 2. Metrics design and requirements
 - Metrics
 - Offline metrics
 - Online metrics
 - Requirements
 - Training
 - Inference
 - Summary

Airbnb rental search ranking#

1. Problem statement#

Airbnb users search for available homes at a particular location. The system should sort stays from multiple homes in the search result so that the most frequently booked homes appear on top.

Search ranking system for Airbnb

- The naive approach would be to craft a custom score ranking function. For example, a score based on text similarity given a query. This wouldn't work well because similarity doesn't guarantee a booking.
- The better approach would be to sort results based on the likelihood of booking. We can build a supervised ML model to predict booking likelihood. This is a binary classification model, i.e., classify booking and not-booking.

2. Metrics design and requirements#

Metrics#

Offline metrics#

- Discounted Cumulative Gain $DCG_p = \sum_{i=1}^p \frac{rel_i}{\log_2(i+1)}$
 - where rel_i stands for relevance of result at position i .
- Normalized discounted Cumulative Gain:

$$nDCG_p = \frac{DCG_p}{IDCG_p}$$

- IDCG is ideal discounted cumulative gain:

$$IDCG_p = \sum_{i=1}^{|REL_p|} \frac{2^{rel_i} - 1}{\log_2(i + 1)}$$

Online metrics#

- Conversion rate and revenue lift: This measures the number of bookings per number of search results in a user session.

$$conversion_rate = \frac{number_of_bookings}{number_of_search_results}$$

Requirements#

Training#

- Imbalanced data and clear-cut session: An average user might do extensive research before deciding on a booking. As a result, the number of non-booking labels has a higher magnitude than booking labels.
- Train/validation data split: Split data by time to mimic production traffic, for example, we can select one specific date to split training and validation data. We then select a few weeks of data before that date as training data and a few days of data after that date as validation data.

Inference#

- Serving: Low latency (50ms - 100ms) for search ranking
- Under-predicting for new listings: Brand new listings might not have enough data for the model to estimate likelihood. As a result, the model might end up under-predicting for new listings.

Summary#

Type	Desired goals
Metrics	Achieve high normalized discounted Cumulative Gain metric
Training	Ability to handle imbalance data Split training data and validation data by time
Inference	Latency from 50ms to 100ms Ability to avoid under-predicting for new listings

Back

Ads Recommendation System Design

Next

Booking Model

Mark as Completed

Report an Issue