### **Detailed Summary: Problem Statement and Metrics - Machine Learning System Design**

#### **Introduction**

The document provides a comprehensive framework for designing a Machine Learning (ML) system for ranking Airbnb rental search results. This involves defining the problem, determining relevant metrics, and outlining requirements for training and inference phases.

#### **Problem Statement**

**Objective:** To optimize the ranking of Airbnb rental search results by the likelihood of booking rather than merely text similarity.

* **Current Challenge:** A naive ranking function based on text similarity does not effectively predict bookings.
* **Proposed Solution:** Implement a supervised ML model for binary classification to predict booking likelihood.

#### **Metrics Design and Requirements**

Metrics are crucial for evaluating the model's performance. The document divides these into offline and online metrics:

**Offline Metrics:**

* **Discounted Cumulative Gain (DCG):** Measures the relevance of results based on their positions in the search results.
  + Formula: DCGp=∑i=1prelilog⁡2(i+1)DCGp​=∑i=1p​log2​(i+1)reli​​
* **Normalized Discounted Cumulative Gain (nDCG):** Normalizes DCG by the ideal DCG (IDCG).
  + Formula: nDCGp=DCGpIDCGpnDCGp​=IDCGp​DCGp​​

**Online Metrics:**

* **Conversion Rate:** Measures the number of bookings relative to the number of search results shown.
  + Formula: Conversion Rate=Number of BookingsNumber of Search ResultsConversion Rate=Number of Search ResultsNumber of Bookings​
* **Revenue Lift:** Evaluates the increase in revenue attributed to improved search ranking.

#### **Requirements**

**Training:**

* **Imbalanced Data Handling:** Due to more non-booking sessions than booking ones, special techniques are required to handle this imbalance.
* **Train/Validation Data Split:** Splitting data by time to mimic production scenarios, e.g., using a specific date to separate training and validation datasets.

**Inference:**

* **Latency:** Ensuring low latency (50ms - 100ms) for real-time search ranking.
* **New Listings Challenge:** Addressing under-prediction for new listings with insufficient data.

#### **Summary**

The summary emphasizes the following:

* **Goals:** Achieve high nDCG and optimize the likelihood of booking predictions.
* **Training Requirements:** Handle data imbalance and time-based data splitting.
* **Inference Requirements:** Maintain low latency and accurate predictions for new listings.

#### **Modules Covered**

The document lists other related modules, such as:

* Machine Learning Primer
* Video Recommendation
* Feed Ranking
* Ad Click Prediction
* Booking Model
* Estimate Food Delivery Time
* Machine Learning Knowledge
* Machine Learning Model Diagnosis

This detailed framework aims to guide the design and implementation of an effective ML system for rental search ranking, ensuring high performance and relevance of search results to enhance user experience and increase booking rates on Airbnb.

### **Detailed Summary: Booking Model - Machine Learning System Design**

#### **Introduction**

The document outlines the design of a machine learning model specifically tailored for predicting booking behavior on Airbnb's rental search platform. This includes feature engineering, training data preparation, and model architecture.

#### **Model Overview**

**Objective:** To predict whether a user will book a rental property, focusing on creating a binary classification model.

#### **Feature Engineering**

**Key Features:**

1. **Geolocation of Listing (Latitude/Longitude):**
   * Challenge: Raw latitude and longitude are difficult to model due to uneven feature distribution.
   * Solution: Use the logarithm of the distance from the map center for latitude and longitude.
2. **Favorite Place Encoding:**
   * Users’ favorite places are stored in a 2D grid and encoded into specific cells, then embedded before training and serving.

**Feature Types and Engineering Techniques:**

* **Listing ID:** Embedded for better feature representation.
* **Listing Features:** Includes number of bedrooms, list of amenities, and listing city.
* **Location:** Latitude and longitude are measured from the center of the user’s map and normalized.
* **Historical Search Queries:** Text embeddings are used.
* **User Features:** Includes age, gender, previous bookings, and previous lengths of stay, all normalized or standardized.
* **Time-Related Features:** Includes month, week of year, holiday, day of week, and hour of day, appropriately normalized or standardized.

#### **Training Data**

**Data Collection:**

* User search history, view history, and bookings are considered.
* Data periods such as the last month or last six months are evaluated to balance training time and model accuracy.

**Data Preparation:**

* Multiple experiments are conducted to determine the optimal length of training data, balancing model accuracy and training time.

#### **Model Architecture**

**Input Data:**

* User data, search queries, and listing data serve as inputs.

**Output:**

* A binary classification output indicating the likelihood of a user booking a rental.

**Initial Model:**

* A deep learning model with fully connected layers is used as a baseline.
* The model outputs a probability between 0 and 1, representing booking likelihood.

**Advanced Model Options:**

* Variational Autoencoder (VAE) and Denoising Autoencoder are considered for further model improvements.
  + **Variational Autoencoder:** Provides a probabilistic framework to model complex data distributions and generate new data points.
  + **Denoising Autoencoder:** Trains the model to reconstruct the original input from a corrupted version, improving feature extraction and robustness.

### **Summary**

The booking model for Airbnb’s rental search is designed to predict booking behavior with high accuracy. The process includes sophisticated feature engineering, careful training data selection, and advanced model architectures. This system aims to enhance user experience by providing highly relevant search results, ultimately increasing booking rates on the platform.

### **Detailed Summary: Rental Search Ranking System Design - Machine Learning System Design**

#### **Introduction**

This document outlines the design and scaling considerations for an Airbnb rental search ranking system using machine learning. It covers estimation, high-level design, scaling strategies, and follow-up considerations.

#### **Calculation & Estimation**

**Assumptions:**

* **User Base:** 100 million monthly active users.
* **Booking Frequency:** On average, users book rental homes 5 times per year.
* **Search Interactions:** Users see about 30 rentals per booking.
* **Data Size:** 15 billion observations annually or 1.25 billion samples per month.
* **Storage Requirements:** Assuming each sample takes 500 bytes, total data size is approximately 625 GB per month.

**Scaling:**

* **Future Support:** Plan to support 150 million users.

#### **High-Level Design**

**Feature Pipeline:**

* Processes online features and stores them in key-value storage for low-latency downstream processing.

**Feature Store:**

* Stores feature values for inference with <10ms access latency. Examples include MySQL Cluster, Redis, and DynamoDB.

**Model Store:**

* Uses distributed storage, such as Amazon S3, to store models.

**System Flow:**

1. **User Query:** The user searches for rentals, and the application server receives the query.
2. **Search Service:** Looks up the indexing database and retrieves rental candidates.
3. **Ranking Service:** Scores each candidate using an ML model based on booking likelihood.
4. **Result Sorting:** Search Service sorts candidates based on booking probability and returns the sorted list to the application server.
5. **User Display:** The application server returns the sorted list to the user.

#### **Scaling the Design**

**Scale-Out Strategies:**

* **Application Servers:** Scale out with load balancers to distribute load.
* **Search and Ranking Services:** Scale out to handle increased request volumes.
* **Logging:** Log recommended candidates as training data using cloud storage or Kafka clusters.

#### **Follow-Up Questions**

**Embedding Issues:**

* **Listing ID Embedding:** Limited unique IDs and bookings per listing might not provide sufficient data for training embeddings.

**Network Redesign:**

* **Correlation with Viewing Time:** If viewing time correlates with booking likelihood, redesign the network to have multiple outputs for view\_time and booking.

**Model Retraining Frequency:**

* **Retraining Triggers:** Monitor online metrics and trigger retraining when metrics degrade.

#### **Summary**

**Key Learnings:**

* **Formulating Search Ranking:** Using booking likelihood as the target.
* **Metrics:** Discounted Cumulative Gain (DCG) for model training.
* **Scaling Systems:** Strategies to handle millions of requests per second.

The document provides insights into formulating and scaling a search ranking system to improve user experience and booking rates on Airbnb. It emphasizes the importance of feature engineering, efficient data handling, and scalable infrastructure to meet growing demands.