Log In Join

Back To Module Home

Machine Learning System Design

0% completed

Search Module

Estimate Food Delivery System Design

Machine Learning Knowledge

Machine Learning Model Diagnosis

Conclusion

□

Mark Module as Completed

Problem Statement and Metrics

Let’s dive deeper into the problem statement and metrics required for the food delivery system.

We'll cover the following □

Estimate Delivery Time#

1. Problem statement#

Build a model to estimate the total delivery time given order details, market conditions, and traffic status.

> To keep it simple, we do not consider batching (group multiple orders at restaurants) in this exercise.

$$DeliveryTime = PickupTime + Point\_to\_PointTime + Drop\_off\_Time$$

2. Metrics design and requirements#

Metrics#

- Offline metrics: Use Root Mean Squared Error (RMSE)

  $$\sum_{k=1}^{n} \frac{(predict-y)^2}{n}$$

  where,

  n is the total number of samples,

  $predict$ is Estimated wait time,

  y is the actual wait time.

- Online metrics: Use A/B testing and monitor RMSE, customer engagement, customer retention, etc.

Requirements#

Training#

- During training, we need to handle a large amount of data. For this, the training pipeline should have a high throughput. To achieve this purpose, data can be organized in Parquet files

- The model should undergo retraining every few hours. Delivery operations are under a dynamic environment with a lot of external factors: traffic, weather conditions, etc. So, it is important for the model to learn and adapt to the new environment. For example, on game day, traffic conditions can get worse in certain areas. Without a retraining model, the current model will consistently underestimate delivery time. Schedulers are responsible for retraining models many times throughout the day.

- Balance between overestimation and under-estimation. To help with this, retrain multiple times per day to adapt to market dynamic and traffic conditions.

Inference#

- For every delivery, the system needs to make real-time estimations as frequently as possible. For simplicity, we can assume we need to make 30 predictions per delivery.
- Near real-time update, any changes on status need to go through model scoring as fast as possible, i.e., the restaurant starts preparing meals, the driver starts driving to customers.
- Whenever there are changes in delivery, the model runs a new estimate and sends an update to the customer.
- Capture near real-time aggregated statistics, i.e., feature pipeline aggregates data from multiple sources (Kafka, database) to reduce latency.
- Latency from 100ms to 200ms

Summary#

Type	Desired goals
Metrics	Optimized for low RMSE. Estimation should be less than 10-15 minutes. If we overestimate, customers are less likely

	to make orders. Underestimation can cause customers upset.
Training	High throughput with the ability to retrain many times per day
Inference	Latency from 100ms to 200ms

Rental Search Ranking System Design

Next

Estimated Delivery Model

Mark as Completed

Report an Issue