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

Estimated Delivery Model

Learn how to build Estimate Delivery model for the food delivery app.

We'll cover the following

3. Model#

Features engineering#

Features	Feature engineering	Description
Order features: subtotal,		

cuisine Item features: price and type Order type: group, catering			
Merchant details			
Store ID	Store Embedding		
Realtime feature	Number of orders, number of dashers, traffic, travel estimates		
Time feature	Time of day (lunch/dinner), day of week, weekend, holiday		
Historical Aggregates	Past X weeks average delivery time for: Store/City/market/Time OfDay		
Similarity	Average parking times, variance in historical times		
Latitude/longitude	Measure estimated driving time between delivery of order(to		

	consumer) & restaurants	

Training data#

- We can use historical deliveries for the last 6 months as training data. Historical deliveries include delivery data and actual total delivery time, store data, order data, customers data, location, and parking data.

Model#

Gradient Boosted Decision Tree#

- Gradient Boosted Decision Tree sample

- How do Gradient Boosted Decision Trees work?
    - Step 1: Given historical delivery, the model first calculates the average delivery time. This value will be used as a baseline.
    - Step 2: The model measures the residual (error) between prediction and actual delivery time.

      $$Error = ActualDeliveryTime - EstimatedDeliveryTime$$

    - Step 3: Next, we build the decision tree to predict the residuals. In other words, every leaf will contain a prediction for residual values.
    - Step 4: Next we predict using all the trees. The new predictions will be used to construct predictions for delivery time using this formula:

      $$EstimatedDeliveryTime = Average\_delivery\_time + learning\_rate * residuals$$

    - Step 5: Given the new estimated delivery time, the model then computes the

new residuals. The new values will then be used to build new decision trees in step 3.

- Step 6: Repeat steps 3-5 until we reach the number of iterations that we defined in our hyperparameter.

- One problem with optimizing RMSE is that it penalizes similarly between under-estimate prediction and over-estimate prediction. Have a look at the table below. Note that both models use boosted decision trees.

Actual	Model 1 Prediction	Model 1 square error	Model 2 Prediction	Model 2 square error
30	34	16	26	16
35	37	4	33	4

- Although Model 1 and Model 2 have the same RMSE error, model1 overestimates delivery time which prevents customers from making orders. Model2 underestimates the delivery time and might cause customers to be unhappy.

Actual	Model 1 Prediction	Model 1 square error	Model 2 Prediction	Model 2 square error
30	34	16	26	16
35	37	4	33	4

Q

We trained 2 boosted decision tree models to predict delivery time: Model1 and Model2. In this table, we have an example of sample data and model predictions.

Which model should we choose to deploy?

Problem Statement and Metrics

Next

Estimate Food Delivery System Design

Mark as Completed

Report an Issue