Kaggle Instacart Solution

challenged Kagglers to predict which grocery products an Instacart consumer will purchase again and when.

This focus on understanding temporal behavior patterns makes the problem fairly different from standard item recommendation, where user needs and preferences are often assumed to be relatively constant across short windows of time.

it's less clear that you'll want to reorder a fresh batch of almond butter or toilet paper if you bought them yesterday.

to understand how he used complex feature engineering, gradient boosted tree models, and special modeling of the competition's F1 evaluation metric to win 2nd place.

how do we take the time since a user last purchased an item into account? Do users have specific purchase patterns, and do they buy different kinds of items at different times of the day? And the competition’s F1 evaluation metric makes sure our models have both high precision and high recall.

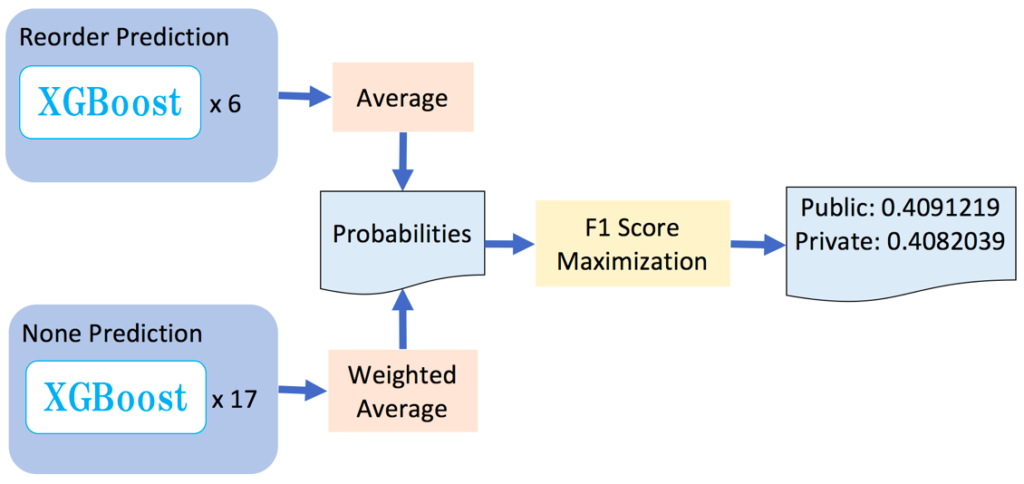
I used XGBoost to create two gradient boosted tree models:

1. **Predicting reorders** - which previously purchased products will be in the next order? This model depends on both the user and product.

**User\_id, product\_id, label**

1. **Predicting None** - will the user’s next order contain any previously purchased products? This model only depends on the user.

User\_id, label



I narrowed down the features to attributes such as order\_day\_of\_week, order\_hour\_of\_day, days\_since\_previous\_order and all my departments (frozen, pantry, breakfast, dairy etc.) converted into numerical variables.

* The reorder prediction model uses XGBoost to create six different gradient boosted tree models (each GBDT uses a different random seed). I average their predictions together to get the probability that User A will repurchase Item B in their next order.
* The None prediction model uses XGBoost to create seventeen different models. 11 of these use an eta parameter (a step size shrinkage) set to 0.01, and the others use an eta parameter set to 0.002. I take a weighted average of these predictions to get the probability that User A won’t repurchase any items in their next order.
* To convert these probabilities into binary Yes/No scores of which items User A will repurchase in their next order, I feed them into a special F1 Score Maximization algorithm that I created, detailed below.