

- **What kind of algorithms would you explore to solve this issue?**

I will use the supervised learning algorithm to predict the purchase count for each product for a shopper.

Two models will be considered - XGBoost and deep learning. The XGBoost will also need the feature engineering for each intersection

- **What is your preferred model?**

Deep learning model is preferred. This model is composed of three layers,

- a) one layer for shopper embedding to represent the shopper features, (Gender, Age, Location, Weather, etc),
- b) one layer for product feature embedding (Color, Size, Type, Price. )
- c) one layer for intersection embedding (Click Count, Click Process Date, Add To Cart Count, etc )

The three layers will be merged to generate the purchase counts, and the top counts products will be recommended.

- **How would you compare different models, and why? Explain the pros and cons of each of these models.**

I will use cross-validation to compare the two models. Cross-validation can avoid the bias of one-single training-test splitting.

Deep Learning model

Pros:

1. strong ability to explore and generate complex features
2. Fit the data better than shadow models

Cons:

1. Training could be time-consuming and needs GPU
2. Hard to explain how the prediction is produced
3. Sensitive to over-fitting and feature scale

XGBoost Model

Pros:

1. Feature importance can be measured by the model and explain the contribution to the prediction
2. Built-in mechanism to handle missing features, over-fitting
3. No need for feature scaling/normalization

Cons:

1. Heavily relying on the quality of feature engineering
2. Sensitive to outliers

Collaborative Filtering will not be considered since it has the problem of cold-start.

- **Let's assume we have chosen to work with a matrix factorization model.**

- **What are the steps/techniques you use to make sure that you are not over-fitting your model?**

1. Splitting the entire data set to training and test sets by cross-validation
2. Train model over the training set and evaluate over the test set
3. Verify if the training and test performance is significantly different or not. If significant, then over-fitting is confirmed.

- **What techniques would you use to detect outliers?**

1. Calculate the Interquartile Range (IQR) from the feature values,  $IQR = Q3 - Q1$
2. decide the range of the decision range.
  - i. lower bound =  $Q1 - 1.5 \times IQR$
  - ii. upper bound =  $Q3 + 1.5 \times IQR$
3. The values lower than the lower bound or larger than the upper bound are identified as the outliers.

▫ **How would you solve the cold start problem? (i.e., how would you update the algorithm so it not only can make recommendations to the existing users in the recommender, but also to new users that have no prior activities)**

Instead of using matrix factorization algorithm, we can use the content-based recommendation system to resolve the cold start problem. The shopper ID or product ID are not used to present the shopper/product, but only their attributes are embedded and matched to do the recommendation. This solution can be extended to unseen shopper/product by only learning from their attributes.

• **How can we train a model which incorporates both the ratings and the Shoppers and Products attributes (age, gender, location for shoppers, type, size, color for products)? Describe your technique.**

I will build a deep learning model. This model is composed of three layers,

- a) one layer for shopper embedding to represent the shopper features, (Gender, Age, Location, Weather, etc),
- b) one layer for product feature embedding (Color, Size, Type, Price. )
- c) one layer for intersection embedding (Click Count, Click Process Date, Add To Cart Count, etc )

The three layers will be merged to generate the purchase counts, and the top counts products will be recommended.

• **Assume we have 1 instance of our model per store (each store has its own recommender) due to resource (memory and time) limitations. How would you efficiently recommend items from one store to another? These stores can share shoppers. Describe your solution(s).**

Using domain transfer learning algorithms. Each store is a domain, and they share some shoppers/products. We can do fine-tuning of one store's pre-trained model to fit another store.