In the past two weeks, I mainly focus on three things improving overall recommendation system performance.

First, I designed query feature that will be used in search project. Search scenario is slightly different from standard recommendation. In recommendation, when user send request for service, only related items that user would be interested in need to be returned. Hence, there are only user-item two parities involved in the recommendation service. However, when user conducts search action, the item returned should first be related to the query user key in, rather than items that user may be interested in. Consider a simple example that for a guy who is only interested in electronic products, decide to purchase a gift for his mother on mother’s day. He may utilize search function to find proper gifts. Under such circumstance, items related to mother’s day should be returned to user although user himself may not interested in them base on his past behaviors. To fulfill such functionality, we need to construct query features to accurately match user key-in query and items.

In my proposed query features, there are four domains, namely, query feature, query-item feature, query-category feature and query-user feature. Most domain are constructed to represent of query information and query-user feature is focused on personalization under query matched items.

Second, due to high recall latency, I transferred the recall platform from ElasticSearch to RediSearch. After one of the projects I am responsible went online, we found that the overall service latency is much higher than we expected. By further breaking down the component of service, we realize the high latency is due to the slow response from ElasticSearch. To solve this problem, one proposed solution is to transfer from ElasticSearch to RediSearch, which in our practice has better latency performance. In the past two weeks, I provided mock data to support the transformation, and monitored the latency performance after new recall service is online. By testing, the new recall service performed well just as we expected, and all traffic has been directed to RediSearch.

Third, I developed algorithms to detect trending keyword using internal resources. One of the newly initiated project is auto-collection, which means to collect a group of items subject to a certain topic, all by algorithm without human interference. To do so, the first challenge we face is how to determine the theme of the collection. We want the theme to be as popular as possible, so that more customers will be willing to step in and purchase items. So I have developed a algorithm, that uses previous 3 days’ search data, to predict what keywords are trending now and have potential to contribute order to platform. Currently I have already developed and submitted the first draft of the algorithm, which only utilizes internal source from company. This will inevitably cause latency in trending detection. In next step, I will try to combine external information like trending topic on Instagram and Twitter to better detect trending topics.