Conrad take home assignment

Shreyas Gururaj

March 10, 2024

1 Bundle recommendation module

1.1 Training design decisions (Neural Collaborative Filtering)

- Neural Collaborative Filtering is an arbitrary non-linear function representation of Collaborative Filtering, which involves training a neural network to project users and items into a shared latent space, using a vector of latent features to represent a user or an item.

- Traditional Matrix Factorization techniques involve decomposing the user-item interaction matrix into the product of two lower dimensionality rectangular matrices. The input User and Item matrix are usually very sparse and the decomposition (eg: SVD) grows in a cubic way ($\mathcal{O}(n^3)$) with the size of the (User X Item) matrix. The final representation is based on a fixed linear function (eg: inner product) and has a cold start problem for each User/Item addition.

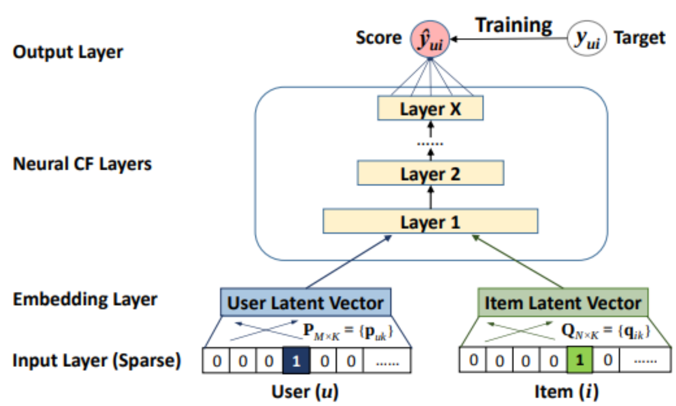

Figure 1: Backbone NCF architecture

- The above backbone architecture was used to learn the implicit feedback of each (User X Item) pair. 4 negative interactions were created for each positive interaction during the training. Each sparse input is projected to dense vectors using the embedding layer which is also trained. In our pre-processed training dataset, the total number of unique customers was **4334** and unique items were **3659**. The user and item embeddings after the embedding layer are of the size **128** each.

- Train-test split: for each StockCode, their most recent transaction is used as the test set(i.e. leave one out) and the rest as the training set. This train-test split strategy is used when training and evaluating recommender systems. Doing a random split (80/20) would not be fair, introducing data leakage with a look-ahead bias, and the performance of the trained model would not be generalizable to real-world performance.

- NCF can be customized to support a wide range of modelling of users and items input embeddings, such as context-aware, content-based, and neighbour-based with the input feature (eg: product description embeddings can be combined)

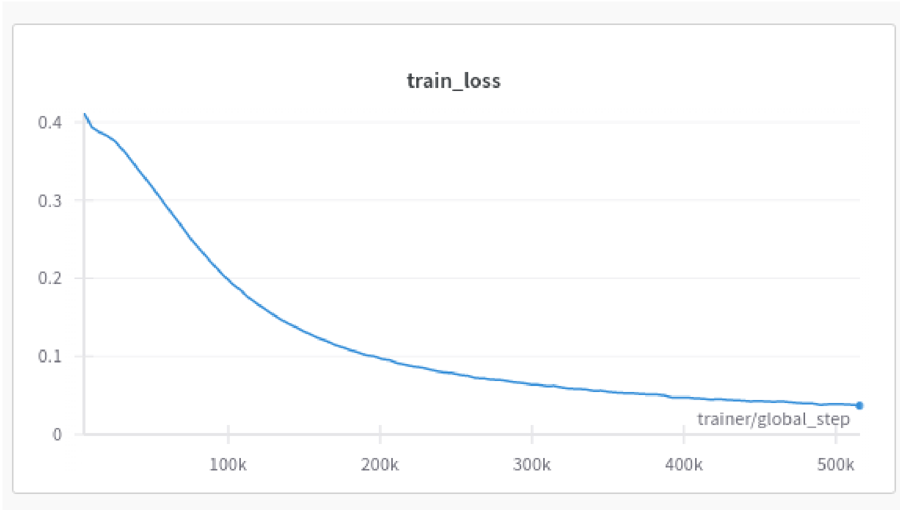

Figure 2: Training loss of the NCF model for 100 epochs

1.2 Evaluation and FastAPI further details

- **HR@10** metric was used to evaluate the model performance. It calculates the share of users for which at least one relevant item is present in the top 10 items with the highest probabilities of items according to the prediction of the trained model. I have used a model with just **2 million parameters** and the HR@10 value was already **0.68** on the test dataset.
- It can be further improved with a deeper NN and by adding other context-based information to the input embedding (eg: Embedding can be generated for the product description (text) feature using a pre-trained transformer style model (BERT) and combined with the input).
- FastAPI is used to create a local web server to process the input request of the user/item pair from the test dataset. The input is sent through the trained model and the top 5 personalized items which the user has not purchased and the respective bundle price is returned as a JSON object.

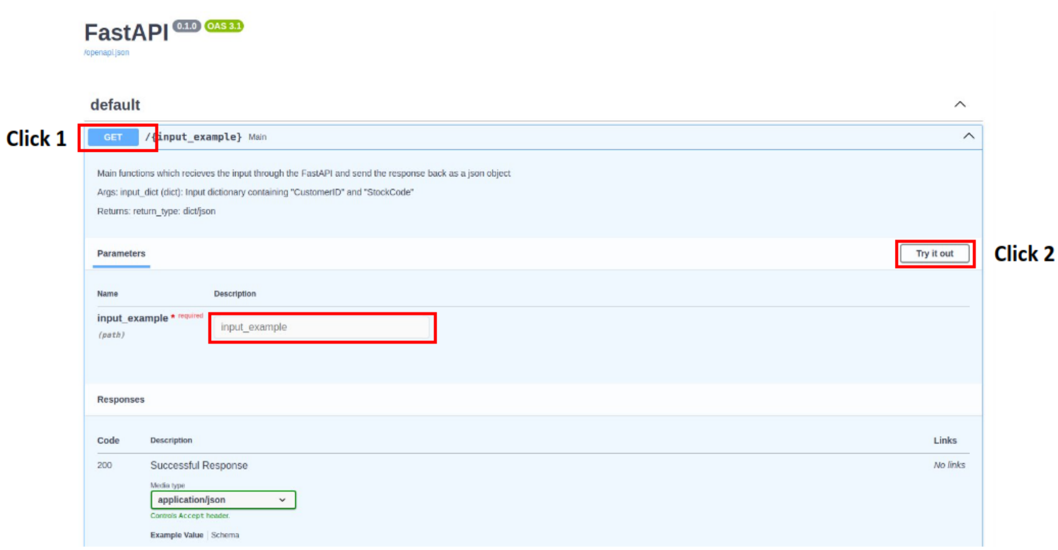

Figure 3: Steps for providing input request in FastAPI webserver

2 Regression 'UnitPrice' prediction

2.1 Methods used

- New and important features (such as "days since last purchase", "monetary", and "item frequency") were engineered for the price prediction task.
- Due to the scarcity of training data (around 2500), after random train/test split (80/20), XGBoost and a very small NN were used for the price prediction task. "Root Mean Squared Error (RMSE)" was used to validate the performance of both the XGBoost and the NN model.

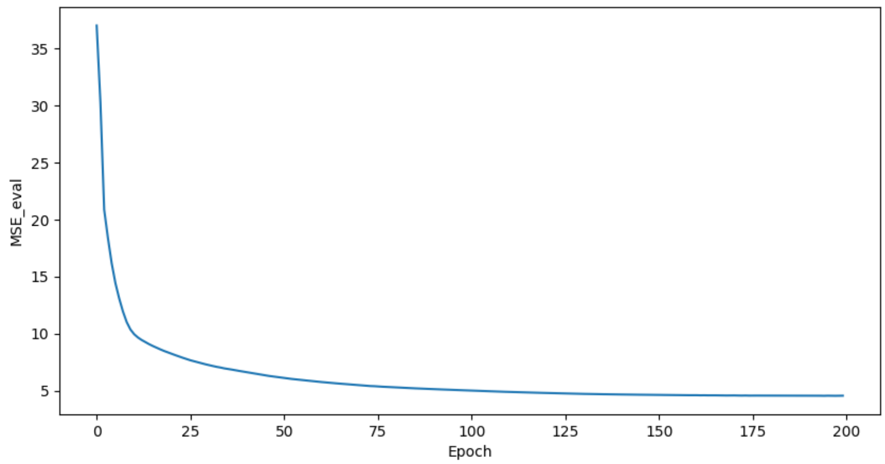

Figure 4: MSE (training loss) of the simple Neural Network used for UnitPrice prediction

2.2 Evaluation

- The range of 'UnitPrices' after removing outliers and zeros was (0.1 Euros to 40 Euros). On the test dataset, the **RMSE of XGBoost: 2.26 and that of the Neural Network: 2.13**.

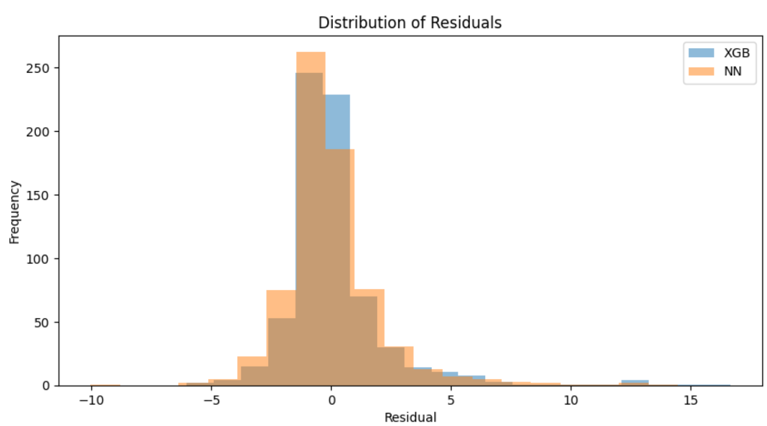

Figure 5: residuals, **abs(model predicted price - actual price)** on the test dataset

- The above residual plot peaks at the centre (Residual = 0), indicating most of the UnitPrice predicted by both the XGBoost and the NN model is close to the actual UnitPrice of the item.

3 Business impact of the proposed recommendation system

- Based on the evaluation metric HR@10 used, it is evident that 68% of the users find at least 1 relevant item in the recommended bundle. Further online metrics like Click-through rate (CTR), Conversion rate, and Average revenue per user (ARPU) are required to finalize the best model to be deployed using A/B testing.

- The size of the top-k recommendation list (currently k=5), can be decided by using a factor of the UnitPrice of the current item. This factor can be decided by using the previous purchase values and purchase trends for each user.