Enhancing Retail Store Recommendations through Deep Neural Networks

Karol Ziolo
Master of DS and AI
Univeristy of Antwerp
Email: karol.ziolo@student.uantwerpen.be
Student Nr: 20224449

***Abstract*—Within this research report, the efficacy of Deep Neural Networks (DNNs) for retail store recommender systems is explored. The study investigates the comparative performance of DNN-based Matrix Factorisation against Multi-Layer Perceptron (MLP) models. Additionally, it evaluates the impact of feature engineering on the TwoTower model and examines the potential of personalized models in enhancing recommender system performance. Findings reveal insights into model performance, the importance of feature augmentation, and the potential for personalized models in retail recommender systems.**

1. Introduction

In the realm of fashion retail, recommendation systems stand as essential pillars, revolutionising how consumers discover and engage with products. These sophisticated tools have undergone significant development, harnessing advanced algorithms and data analytics to anticipate and cater to individual preferences. Their evolution is paramount for the industry, shaping a landscape where personalised experiences reign supreme. By understanding and adapting to changing consumer behaviours, these recommendation systems not only drive sales but also elevate customer satisfaction. As such, their continual enhancement remains a critical strategy for fashion brands to ensure relevance, resonating deeply with their audience's ever-evolving desires.

This report outlines my endeavour to develop a recommender system for the H&M Personalised Fashion Recommendations Kaggle competition. The competition provided datasets detailing customer profiles, articles, and transactions. Utilising this data, I aimed to construct a recommender system leveraging Machine Learning techniques to predict customer actions. Initially, we were introduced to a foundational model, Radek's LGBMRanker, during our classes. This model primarily relied on article popularity and bestsellers. However, in my approach, I sought to create diverse models utilising Deep Learning methods for a more personalised recommendation system. I believed this strategy would better capture and interpret customers' shopping behaviours.

In the project, the decision was made to evaluate two distinct deep learning models. The initial model involved employing a Multi Layer Perceptron (MLP) to analyse customers' baskets and forecast their subsequent basket choices. The second model utilised a Two Tower approach to construct customer embeddings and article embeddings, utilising these embeddings to estimate the likelihood of a particular customer purchasing a specific item. Both models underwent rigorous testing with various configurations, allowing for a comprehensive comparison of their performances.

2. Research Questions

Deep Learning has demonstrated its ability to comprehend intricate data patterns. There's a belief that in recommendation systems, this capability could contribute to generating more tailored suggestions. As we delve into improving these systems within the realm of fashion, three significant research queries have surfaced:

1) Which DNN model performs better, MPL for Recommendation or DNN for Matrix Factorisation?

2) Can the enhancement of customer and article features improve the performance of the TwoTower model?

3) Can the creation of personalised models enhance the performance of recommender system?

Together, these research questions seek to expand the understanding of recommendation systems in the fashion retail domain. They provide valuable insights into strategies for improving user experiences and optimising these systems specifically for enhancing recommendations at H&M and similar retailers.

3. Data

This section will concentrate on the datasets provided for analysis. These datasets encompass three categories: the initial dataset contains comprehensive information about all customers and their respective attributes. The second dataset comprises detailed information about various articles and their features. Finally, the third dataset catalogs transactional

records along with their specific details. Within this section, I aim to explore and delineate the characteristics of these datasets to offer a comprehensive overview of the context we are navigating.

3.1. Data Exploration

The purpose of this subsection was to familiarise ourselves with the data by generating statistical insights and visualising patterns and observations. The following table displays descriptive statistics derived from the datasets:

TABLE 1. DESCRIPTIVE STATISTICS

Dataset	# of observations	# of features with NAs
Articles	105,542	1
Customers	1,371,980	5
Transactions	31,788,324	0

The information within the table indicates a substantial volume of data, potentially valuable for pattern recognition, yet also posing potential memory-related challenges. Thus, efficient data handling becomes crucial for the models. Additionally, both the Article and Customer datasets contain variables with missing values.

During this phase, I conducted a detailed analysis of specific features to assess whether any preprocessing steps were necessary and to identify fundamental patterns. An initial observation revealed that nearly all article features are categorised or grouped. Figure 1. illustrates the distribution of assortment concerning the index, delineating the subgroups within each index. This pattern is prevalent across most features within the dataset and can be depicted similarly for various other attributes.

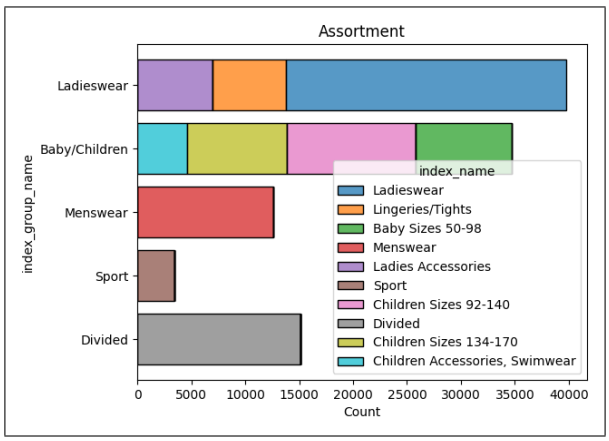

Figure 1. Assortment Distribution

Upon scrutinising the features, certain discernible patterns emerged. Figure 2. illustrates the presence of discernible preferences in the choice of selling channels. This statistic presents an opportunity to explore whether distinguishable patterns exist within these distinct groups.

I opted to delve deeper into the age variable, considering its potential significance in identifying distinct patterns. Notably, variations in age distribution across index groups were

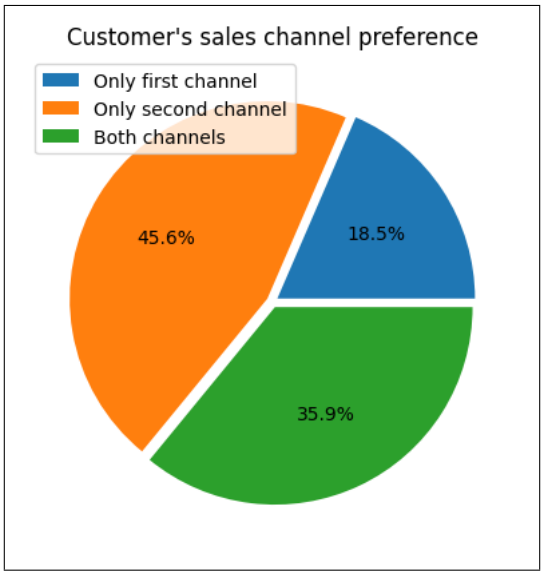

Figure 2. Sales Channel Distribution

observed. Consequently, a decision was made to generate a violin plot (Figure 3.) to visually represent these differences. The visualisation distinctly portrays that most groups exhibit a comparable age distribution, except for the Baby/Children group. Unlike the characteristic double hills observed in other groups, this group showcases a single hill, indicating a shift towards older customers. This observation holds potential significance for implementation in the modeling phase.

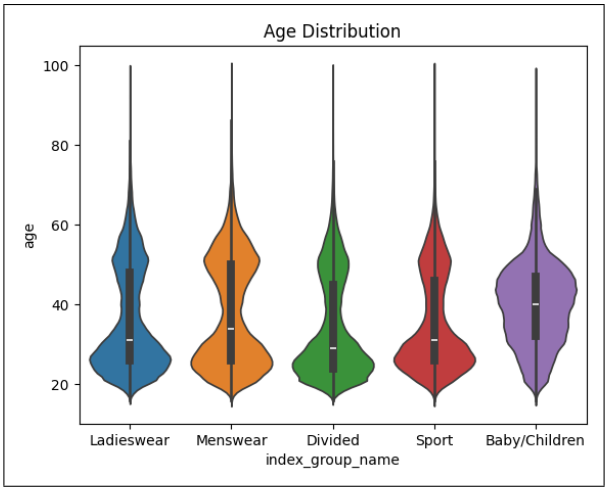

Figure 3. Age Distribution

Furthermore, an exploration into the sales proportion across index groups within the product categories was conducted and is depicted in Figure 4. This cross-analysis of features highlighted variations, indicating that certain product categories exhibit higher popularity within specific index groups. The plot on the left showcases highly sold products, whereas the right plot exhibits less popular products. It's

apparent that the majority of products are predominantly favoured within the Ladieswear segment. Nonetheless, valuable insights can be derived from these visualisations. For instance, only a few product groups are observed within the Menswear index group. Similarly, specific groups are exclusively sold within the Children or Ladieswear segments.

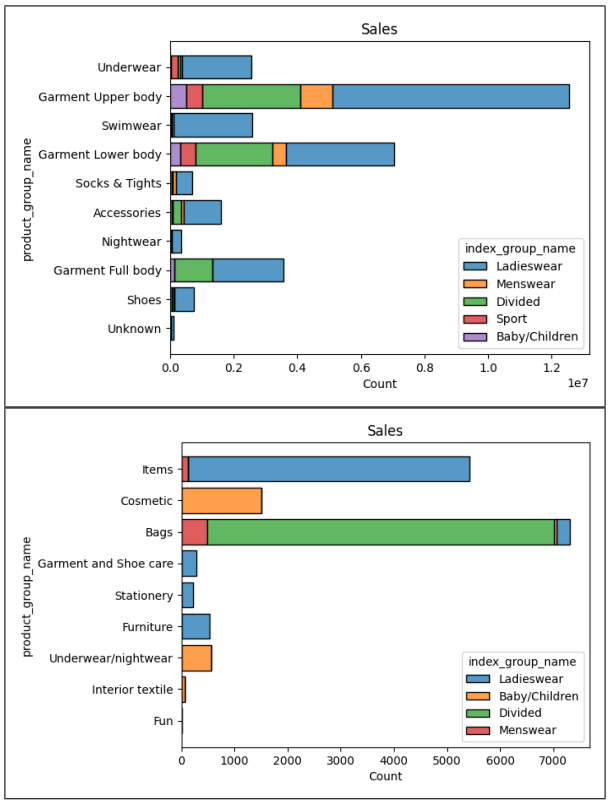

Figure 4. Sales Distribution

3.2. Data Preprocessing

From the previous subsection we can conclude that, the data included vast amount of information. However, it needed to be properly preprocessed to capture some patterns. Firstly, it was noted that data presented in the article dataset might be strongly colinear as most features are grouped. This might raise some problems related with the model's stability. Moreover, I also tried not focus too much on the detailed features as they might lead to loosing the important patterns. Therefore it was decided to leave the following features: product type, graphical appearance, perceived colour master, department dame, index name, section name and garment group. In terms of the customers dataset there was different issue. There were some features that consist missing values. For FN and Active features it was decided to add -1. However, for Age it was decided to fill it with a median value. For both these datasets I decided to encode their categorical variables to reduce their memory space which handling will be a critical point in the modelling part. Therefore the encoder and decoder dictionaries have been created. For transaction dataset it was decided to transform the date feature to date format and then apply encodings from customers and articles.

From the prior section, it's evident that the data holds a wealth of information, but preprocessing was necessary to uncover specific patterns. Initially, the article dataset raised concerns about strong collinearity due to feature grouping, potentially impacting the model's stability. Also, I steered away from focusing too much on intricate features, as this might obscure essential patterns. Consequently, I opted to retain certain key features—such as product type, appearance, colour, department, index name, section name, and garment group.

In the customer dataset, there were missing values in some features. To address this, I used "-1" for FN and Active features, while Age was filled with the median value. Additionally, I encoded categorical variables in both datasets to conserve memory space, which will be crucial during modelling. This led to the creation of encoder and decoder dictionaries.

Regarding the transaction dataset, I converted the date feature into date format and applied encodings derived from the customer and article datasets. This comprehensive approach aims to optimise data handling and lay a solid foundation for subsequent modelling steps.

4. Model Creation and Development

In this section of the report, I'll explore the architecture of the models, their training process, and how data was managed and supplied. Our attention will be directed towards two specific models, namely the MLP and TwoTower. This segment plays a pivotal role in addressing the initial research question. It serves as a foundational component for subsequent analyses and advancements. Additionally, it holds significance in handling the considerable volume of data, which stands as a primary challenge within this project.

4.1. Multi Linear Perceptron

The initial model, the Multi Linear Perceptron, utilized a straightforward neural network architecture to forecast forthcoming articles. Specifically, it operated by processing basket vectors for customers, predicting the likelihood of a customer purchasing a particular product based on these vectors, with the intention of recommending articles with the highest probabilities.

The primary challenge encountered revolved around the extensive data, considering the vast amount of articles, exceeding 100,000, leading to sparsity. To address this, sparse matrices were employed. I developed a collate function to transform the scipy sparse matrix into a pytorch sparse tensor for application within pytorch, subsequently used by the dataloader. This was instrumental in efficiently managing memory by loading sparse matrices in batches via the dataloader.

The MLP model's initial architecture was relatively straightforward, featuring a sole hidden layer comprising 100 neurons. Activation functions were incorporated—ReLU for the initial layer and sigmoid for the classification layer. Opting for sigmoid over softmax was deliberate, considering the model's focus on addressing multi-classification challenges and assuming that the likelihood of purchasing an item remains independent of other purchase decisions.

For model training, a specific function was devised, employing the Adam optimiser. Adam's adaptive learning rate aids quicker convergence, particularly in scenarios with sparse gradients or noisy data. It dynamically adjusts learning rates for each parameter, potentially leading to faster convergence and enhanced performance compared to traditional optimisers like SGD. Additionally, the BCEWithLogitsLoss function was utillized, optimising predictions for each class independently, advantageous when classes aren't mutually exclusive. The model underwent training for 10 epochs, with a learning rate set at 0.001.

However, despite successful training without crashes, the model exhibited underfitting, reflected by identical train and validation accuracy across all epochs. This common issue in recommendation models hindered the model's learning beyond the initial epoch.

To address this, a deeper model with two hidden layers was developed—one with 500 neurons and the other with 100. The training parameters remained unaltered. The deeper architecture exhibited slightly improved performance, albeit the differences weren't significantly impactful.

4.2. Two Tower Model

The subsequent model developed was the Two Tower model. In essence, this model is structured around two distinct blocks (towers) that leverage deep neural networks to compute embeddings for articles and customers. These embeddings are subsequently utilised to estimate the probability of a customer purchasing a specific article.

4.2.1. Base Model.

In contrast to the MLP, the Two Tower model necessitated different data handling procedures. Firstly, I implemented negative sampling, opting for random selection of negative samples. This action resulted in doubling the dataset size to achieve a balanced representation of positive and negative samples. Secondly, the dataloader required a different approach. This time, it needed to manage customer IDs and article IDs as both inputs and targets.

The architecture of the Two Tower model involves two distinct models: the Customer Tower and the Article Tower. These models accept customer feature vectors and article feature vectors, respectively. They generate embeddings, compute the dot product, and subsequently employ the sigmoid function to estimate probabilities.

The training process is more intricate, involving the extraction of specific customer and article IDs from dataloader batches. Dense tensors of corresponding features are gathered and fed through the model. For all categorical features, I employed one-hot encoding. Moreover, the training utilized the Adam optimizer and MSE loss function. This training method proves efficient in terms of memory usage, and the model exhibits better learning between epochs compared to the MLP model.

To enhance the Two Tower model, I experimented with deeper architectures for both the customer and article towers. Additionally, I tested embedding layers, excluding one-hot encoding, in place of linear layers.

4.2.2. Customer and Articles Diversification.

While training the base models, it became apparent that both customer and article features wield substantial influence over model performance. The extensive range of these features could potentially facilitate the generation of precise embeddings that comprehensively characterize customers and articles. As a result, the decision was made to implement feature engineering and generate new features derived from the provided data.

The selected features to be created within the customer dataset include: 1) Sales channel preference, 2) Favourite colour for current season, 3) Favourite garment group, 4) Average price, 5) Amount of recent purchases, 6) Sex prediction, 7) Kid prediction. However, for the article dataset it was decided to create: 1) Proportion of seasonal sales, 2) Seasonal bestsellers, 3) Age group preference, 4) Average price, 5) Sales channel preference.

The description how these features have been obtained are presented in the appendix alongside with their distribution

5. Personalised Models

This part of the research focuses on developing models that cater to specific user preferences and behaviours. The aim is to understand how personalized methods improve the accuracy of recommendations. This investigation seeks to uncover how personalized models enhance the effectiveness of recommendation systems, providing valuable insights for strategic improvements aligned with the preferences of H&M's customers and other similar retailers.

The primary approach involved identifying customers based on their distinct shopping behaviors. Following a thorough analysis, several discernible groups were identified and are outlined below:

1) Customers following bestseller trends based on they age group,

2) Customers buying only clothes from a specific index section,

3) Customers buying only discounted products,

4) Seasonal customers.

To assemble these customer groups, I had to compute relevant indices reflecting their respective behaviours. Utilising these indices, I determined the distribution among these customers and identified thresholds indicating membership in specific groups. The complete analysis is available in the appendix.

However, one potential concern emerged that could affect the recommender systems. Specifically, certain customers were found to be associated with multiple groups, as illustrated by the heatmap showcased in the Figure 5.

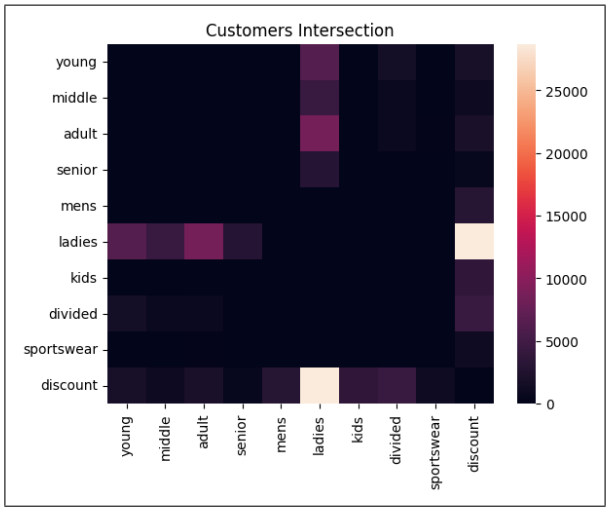

Figure 5. Shared Customers

6. Results

To generate recommendations across all these models, distinct recommendation functions were crafted, each handling the MLP and Two Tower models differently. For MLP models, the process was relatively straightforward. However, the recommenders for the Two Tower models required more intricate memory management. Initially, embeddings for all customers and articles were generated, followed by the creation of customer batches. Vectorization methods were employed to derive probabilities and select the top probable articles for each customer within the batch, forming a separate recommendation matrix.

To assess the recommenders, a validation group of customers was formed, utilizing their last purchased baskets as targets. Subsequently, for the remaining baskets, the top-k recommendations were generated. By comparing these recommendations to the targets, precision and recall scores were calculated. The initial comparison was conducted for all base MLP and Two Tower models, depicted in the Figure 6. Both comparisons highlighted the substantial superiority of the Two Tower model over the MLP model, which displayed notably poor performance. This conclusion effectively addresses the first research question. Furthermore, it was observed that the developed Two Tower model with embedding layers performed less effectively than the model utilizing One Hot Encoded categorical variables and employing linear layers. Additionally, the model featuring deeper towers showcased improved performance over the basic version. However, the most noteworthy enhancement was the significant improvement in model performance achieved through both customer and article diversification.

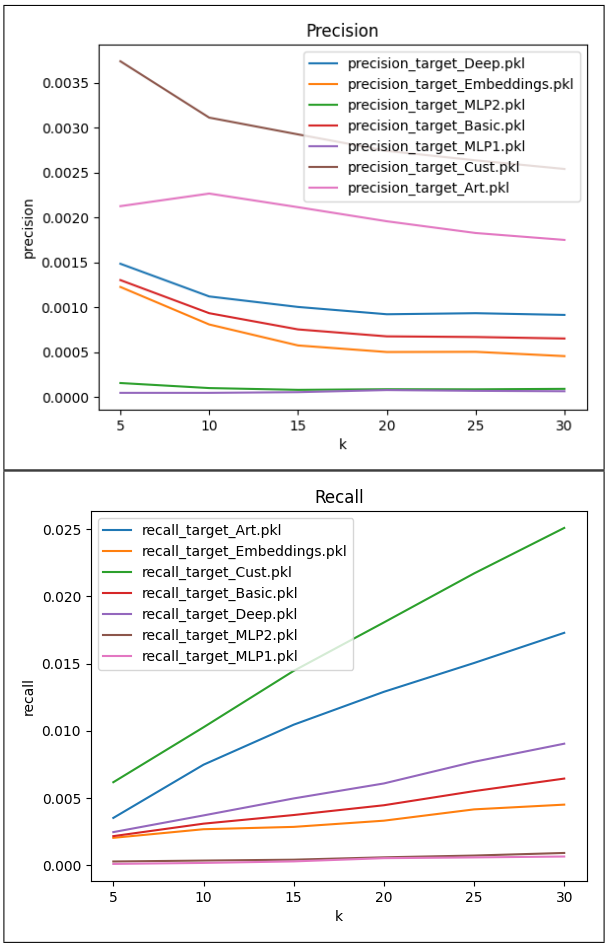

Figure 6. Performance Results for Base Models

This observation provides a positive response to the second research question.

Given the superior performance of the Two Tower model with a deep architecture and diversified article and customer inputs, it was chosen as the foundation for addressing the third research question. This model served as the basis for training personalised models. However, an observation emerged indicating that, in certain customer scenarios, the shallow customer tower exhibited better performance. Consequently, a decision was made to train two models for each customer group.

Furthermore, the recommenders were now capable of considering only specific article candidates. Hence, recommendations were generated for each model, encompassing scenarios with and without the inclusion of article candidates. This process yielded a plots, presented in the Figure 7., illustrating the precision and recall scores of the recommenders based on these criteria.

From these plots, it's evident that performance strongly correlates with customer groups. Additionally, the effectiveness of specific settings varies across these groups. Interestingly, customers categorised as kids exhibits slightly poorer performance compared to others. This suggests a potential

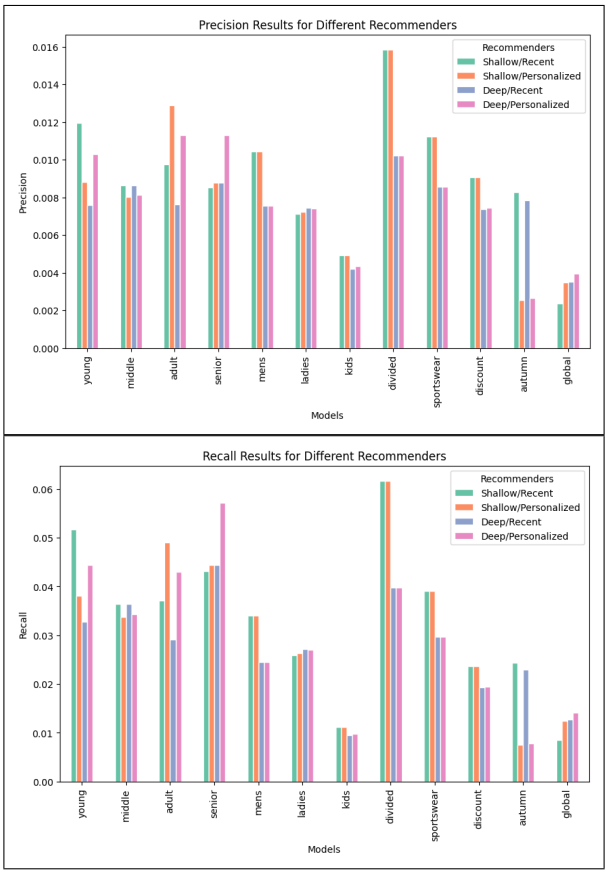

Figure 7. Performance Results for Personalised Models

need for adjustments in the indices defining these behaviours or for stricter thresholds. Consequently, the decision was made to select the best-performing model within each group.

To generate final recommendations using personalised models, a pipeline was constructed to produce recommendations based on the models assigned to individual customers. Addressing the issue of customers assigned to multiple groups, the solution involved prioritising models based on their precision scores. Hence, the pipeline recommends products starting from customers associated with models performing less effectively and progresses to those with better-performing models. This approach overwrites recommendations for shared customers with models exhibiting superior performance.

In evaluating this personalised recommender system, the top-k precision and recall scores were computed again, and presented in the Figure 8. Additionally, the Average MAP score has been calculated and the results are presented in the Figure 9. Ultimately, a comparison was made among four models: the personalised model, the personalised model excluding already purchased products, the basic Two Tower model (lacking article and customer diversification), and the global model, encompassing all developments but lacking personalisation.

Once again, it's evident that through further enhance-

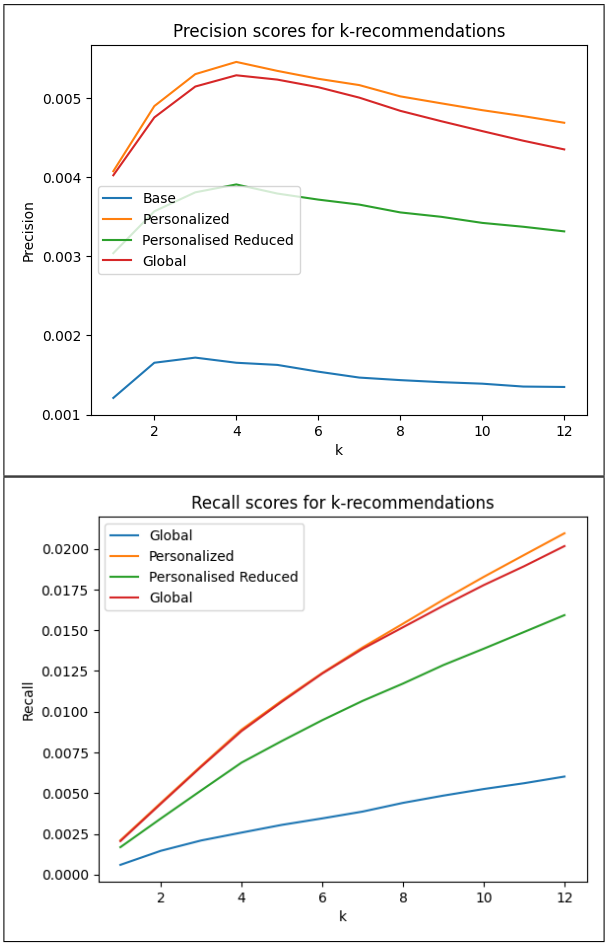

Figure 8. Performance Results for Personalised Models

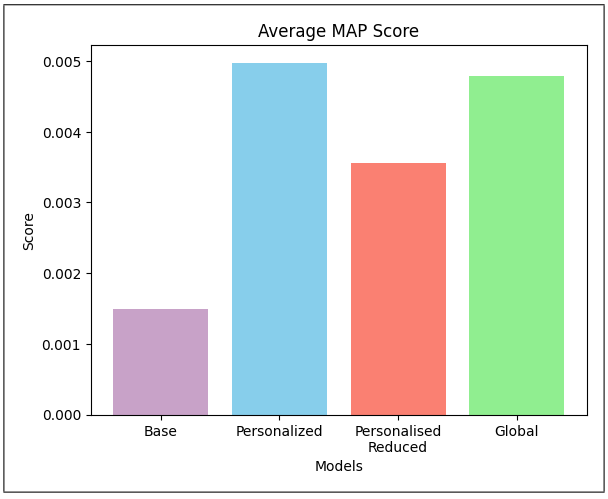

Figure 9. Performance Results for Personalised Models

ments, the performance of the recommender system has markedly improved. In all plots, the Base model is notably surpassed by the other models. Additionally, it is observable that the personalised model marginally enhances the results, except in scenarios where already purchased products are restricted.

Finally, all the recommendations have been submitted to the Kaggle competition website, and their scores are outlined in the Table2.

TABLE 2. FINAL SCORES

Model	Private Score	Public Score
Basic	0.00084	0.00091
Personalised	0.00293	0.00306
Global	0.00332	0.00322

Additionally, I'd like to touch upon Radek's model, which significantly outperformed the model I presented. Its superior performance largely stems from its reliance on repurchased articles as candidates. Hence, an attempt was made to adopt a similar approach for both the personalised and global recommenders. In this instance, previously purchased articles were utilised as candidates, but from the last month, in contrast to Radek's approach, which utilised those from the last week. The private scores for this modified approach were 0.01061 and 0.01302, while the public scores were 0.01073 and 0.01323. These results provide a glimmer of hope indicating that the Two Tower model is beginning to chase the LGBMRanker.

7. Conclusion

In conclusion, I successfully addressed all three initial research questions outlined in this report. The superiority of the DNN for Matrix Factorisation over MLP models in constructing recommender systems was confirmed. Moreover, the significance of feature engineering became evident, showcasing that an increased array of customer and article features notably enhances the Two Tower model's performance. However, when it comes to the personalised Two Tower model, the outcomes were not as conclusive as in prior cases. While the evaluation part based on the validation dataset suggested a slightly superior performance of the personalised model over the global one, the Kaggle results imply the contrary.

Towards the end, I opted to assess both the personalised and global models using repurchased candidates, resulting in a substantial improvement in the Kaggle results. I do regret not initiating this approach earlier during the creation of personalised models. I believe that merging these candidates with the Two Tower model could effectively differentiate customers, potentially elevating the model's performance. In essence, this study underscores the potential within the Two Tower model but emphasises the need for an extensive investment in time and data to gather additional features. Moreover, it confirms the critical nature of handling article candidates and diverse customer groups.

References

[1] Christian Bracher, Sebastian Heinz, and Roland Vollgraf. Fashion dna: Merging content and sales data for recommendation and article mapping, 2016.

[2] Miguel Campo, JJ Espinoza, Julie Rieger, and Abhinav Taliyan. Collaborative metric learning recommendation system: Application to theatrical movie releases, 2018.

[3] Google Developers. Softmax function (tensorflow). Retrieved from https://developers.google.com/machine-learning/recommendation/dnn/softmax.

[4] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering, 2017.

[5] Sebastian Heinz, Christian Bracher, and Roland Vollgraf. An lstm-based dynamic customer model for fashion recommendation, 2017.

[6] Asit Mishra, Jorge Albericio Latorre, Jeff Pool, Darko Stosic, Dusan Stosic, Ganesh Venkatesh, Chong Yu, and Paulius Micikevicius. Accelerating sparse deep neural networks, 2021.

[7] PyTorch Contributors. PyTorch Documentation on Sparse Tensors. https://pytorch.org/docs/stable/sparse.html, 2023. Accessed: December 2023.

[8] Yanru Qu, Han Cai, Kan Ren, Weinan Zhang, Yong Yu, Ying Wen, and Jun Wang. Product-based neural networks for user response prediction, 2016.

[9] Florian Strub and Jérémie Mary. Collaborative Filtering with Stacked Denoising AutoEncoders and Sparse Inputs. In *NIPS Workshop on Machine Learning for eCommerce*, Montreal, Canada, December 2015.

[10] Shuai Zhang, Lina Yao, Aixin Sun, and Yi Tay. Deep learning based recommender system: A survey and new perspectives. *ACM Computing Surveys*, 52(1):1–38, February 2019.

[11] Yuhan Zhao, Rui Chen, Riwei Lai, Qilong Han, Hongtao Song, and Li Chen. Augmented negative sampling for collaborative filtering. In *Proceedings of the 17th ACM Conference on Recommender Systems*, RecSys '23. ACM, September 2023.

[12] Guorui Zhou, Chengru Song, Xiaoqiang Zhu, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. Deep interest network for click-through rate prediction, 2018.