School of Informatics

Informatics Project Proposal
Recommender Systems: looking further into the future

S2406121
April 2023

Abstract

Recommender systems are widely used technique which helps to generate a recommendation list to users based on their previous behaviours. Compared with traditional popular-based recommenders, recent recommender systems pay more attention to providing more personalised user experience to increase user satisfaction. However, the widely used prediction method for recommender systems is based on the next-item prediction, which solely focuses on the next single action of the user, and may not fully capture a user's overall preferences. In this project, we introduce the multi-day window predictor to the recommender system that aims to predict all items that the user may be interested in within a time window. In theory, this novel architecture could lead to better recommendation results, reduce the cost of deployments, and be more robust against short-term variations.

Date: Thursday 20$^{\text{th}}$ April, 2023

Tutor: Aurora Constantin
Supervisor: David Wardrope, Timos Korres, Michael la Grange

1 Motivation

The personalised experience is one of the most attractive user experiences in different social media and e-commerce, which has already attracted a lot of attention recently. Recommendation system has become a popular topic [1], which helps users to find their interests and new items based on their historical behaviours sequence. The performance of a recommendation system can directly influence user satisfaction in e-commerce, streaming, and social media platforms. Many different kinds of approaches [2] to improve the performance of recommendation systems have been proposed including traditional models and deep learning models. In recent years, many sequential models, such as recurrent neural network (RNN), have been proven to be very useful in processing sequential data, which is also widely used in recommendation system [3, 4, 5, 6] because users' previous action can also be seen as an input sequence. Due to the great improvement brought by the sequential neural recommender, the performance of the recent recommendation system has been shown very effective. In 2016, Hidasi et al. [3] proposed a GRU-based RNN model for sequential recommendations that highly improves the traditional KNN-based recommender. In 2018, A transformer-based model SASRec proposed by Kang et al. [4] shows better general performance on different datasets compared with GRU4Rec and other traditional models. However, most existing methods still focus on next-item prediction, which aims to correctly recommend the single next item that the user will interact with, and this method may not capture the sufficient scope of user preferences and behaviour. Furthermore, focusing on predicting the next single item can be easily influenced by short-term variations, which also require more frequent deployments to update real-time recommender. Therefore, this project aims to further improve the performance of recommender systems by implementing a window-based sequential recommendation system, which can correctly identify all items the users may interact with in the future within a time window.

1.1 Problem Statement

This project aims to improve the performance of the current recommender system by proposing a new multi-day window prediction method and examining the effectiveness of this architecture.

The next-item prediction method is still mostly used to train the recommender system, but the weaknesses of the next-purchase prediction learning tasks are very obvious. The next-purchase prediction only focuses on predicting the immediate next single action of a user, which means the training progress of this method will only focus on a single user interest. However, the historical behaviours sequence may contain multiple types of user interests, but the next-item prediction will only consider a single interest, which could reduce the diversity of recommendations. Meanwhile, the multi-day window predictor will consider all user interests within the given time window, which will increase the diversity of user interests in theory. Furthermore, next-purchase prediction considers aiming to predict the next single action could be easily influenced by the short-term variation, a sudden interest may make the system ignore the long-term dependency and then generate based on the short-term interest, which will reduce the accuracy of the system. The window-based prediction method will not only aim for predicting a single action but all actions within the time window, which will make the system focused more on long-term user preference and make the system more robust.

1.2 Research Question and Objectives

The aim of the project is to examine whether predicting the sequence of a customer's future purchases is a better learning task for recommender systems than the standard approach of predicting their single next purchase. Therefore, This project will be driven by the following research question: How to design the multi-day window predictor, to outperform the traditional next-item prediction method.

We can further decompose the project into three objectives. In the first subtask, we will implement a sequential neural recommendation system for next-item prediction, specifically based on the transformer-based model SASRec. In the second subtask, we will extend the constructed model to a window-based recommendation system, which aims to predict all items in a given time window. The final subtask will focus on evaluating and analysing the performance of the window-based model with next-item prediction models including different existing models[3, 4, 6] and also some very simple baseline models[2] to prove the effectiveness of the window-based predictor. The evaluation of the proposed model will consider the accuracy and diversity of the recommendation. Overall, this project will explore how to build a window-based sequential recommendation system in detail and whether this window architecture could outperform the next-purchase prediction approach that is most commonly used.

1.3 Timeliness and Novelty

The proposed multi-day window predictor is novel in this field because recent recommender systems still focus on the next-item prediction [2, 3, 4], which pays more attention to single next action but cannot fully capture the overall user preference mentioned above. In our novel approach, All items the user will interact with in the future within the time window will be considered, which may help the system to better capture the user preference and give a more diverse and accurate recommendation. Moreover, the personalised user experience has become more and more popular with the development of the current technology, so it is necessary to improve the performance of the recommendation system to fit the users' requirements. The window-based recommender system PinnerFormer is firstly proposed by Pancha et al. [7] in 2022 to better predict user interest in Pinterest, which shows an improvement compared with the SASRec [4], a transformer-based next-item prediction system. However, PinnerFormer is still a Pinterest-specific task, which only experiments on its own dataset, so we will extend this architecture to other public datasets to evaluate its performance on predictive accuracy and recommendation diversity. Therefore, it is meaningful to examine whether this time window architecture could actually lead to better performance of the sequential recommendation system.

1.4 Significance

The Recommender system is a popular topic in current society, which could affect many areas of our daily life. For instance, a movie search website always needs to provide a movies recommendation based on users' interactive history, an e-commerce website needs to recommend to users their potential interests based on their transaction history, a social media needs to attract users' interest by recommending them some news they may be interested in. Due to the wide usage of the recommender system, it is significant to make some improvements on this field. In recent recommendation systems, the next-item prediction is still the most widely used method, but this method is less robust and hard to capture the large scope of user interests because it only focuses on the next single action. Therefore, it is meaningful to propose some new approaches to improve the model's performance and robustness. This project will introduce a new window prediction approach and examine the effectiveness of the approach, which could potentially improve the current prediction method of sequential recommender systems. If the window prediction has been shown more effective compared with the next-item prediction approach, it will help further development in the field of recommender systems, which may change the research direction from traditional next-item prediction to time window prediction. Furthermore, this project can also incorporate some currently proposed methods [8, 9, 10] for sequential recommendation systems, which can potentially help to achieve better performance when predicting based on user preference.

1.5 Feasibility

The proposed plan of the project consists of implementing a next-item prediction system, extending the time-window architecture and the evaluation. We have found several alternative sequential recommenders from the academic literature[3, 4], which could be used as baseline models. The code implementation of these models [3, 4, 6] could be found on GitHub, which makes the implementation and extension tasks more feasible. Additionally, the data we will use are open-source, which has already been used in other academic literature such as SASRec[4]. For the final evaluation task, we need to compute the performance of different models, which could need a lot of time and computational cost. To address this problem, we will access informatics GPU (MLP cluster), which could provide the sufficient computational ability to address the issue. By separating the whole project into several three subtasks mentioned above, the project could be completed within the 10-week time frame. These steps show the whole progress could be completed on time before the due time.

1.6 Beneficiaries

The research project will benefit all users of the system and may improve the development of recommender systems from next-item prediction to window-based prediction. Specifically, users' overall historical actions may indicate an interest in multiple types of items including cars and houses, but rather than only focus on the next single action which interacts with the car in the next-item prediction, the multi-day window predictor will try to infer all relevant items users may interact with in the future within the given time window, which will give a more reasonable and diverse recommendation list leading to higher user satisfaction. Furthermore, if the proposed next-window prediction approach could be proven to improve the model performance compared with the traditional method, it will significantly influence the field. The performance of recommendation systems in different domains such as e-commerce and movie recommendation may be improved, which will facilitate the development of sequential recommendation systems and also result in higher user satisfaction.

2 Background and Related Work

The development of recommendation systems moves fast, many existing approaches have been proposed and shown effectiveness to some extent. In this section, we introduce different approaches to recommendation systems including traditional models and deep learning models in this section. The strengths and weaknesses of each method are discussed.

Beginning with traditional recommendation systems, traditional recommendation systems often use the first-order Markov Chain for capturing the short-term dependency within a behaviour sequence, which only focuses on the most recent previous single action of the user. Most of the conventional approaches ignore the influence of the sequential dependencies among the user's interactions, which results in an inaccurate recommendation. However, it is worth analysing to analyse the performance of conventional recommendation approaches, which can be used as useful baselines in the project.

PopRec is the simplest model, which can be easily implemented by always recommending the top popular items to the users. However, this model is very commonly used as a baseline in many famous recommendation systems [4, 3].

K-nearest neighbors(KNN) is also used for sequential recommendation. Davidson et al. [11] use the KNN method to produce a YouTube video recommendation system, which has been demonstrated to have an improvement compared with some very basic recommenders. Item-based KNN method, which only considers the last behaviour in a sequence and uses the most recent behaviour to calculate the similarity, is a highly explainable approach. However, the KNN recommendation system cannot capture the sequential dependency in the input sequence, which lead to a bad recommendation. Interestingly, A recent research [12] compared the KNN-based model and multilayer perception(MLP) in 2020, which shows the simple KNN-based matrix factorization model could achieve a better result than a learned similarity using a MLP with optimal hyperparameters.

Matrix Factorization (MF) learns the latent factors for items and users, which is still commonly used as a layer in the current neural recommender. In 2009, Rendle et al. [13] proposed Bayesian Personalized Ranking (BPR), an MF-based recommendation method. BPR learns the user embedding and item embedding from the user-item interaction matrix and uses the specific user embedding to calculate the score of all items in the item embeddings. The difference between this method with standard MF is the optimization objective, BPR aims to rank the relevant positive interaction higher than the negative interaction whereas the standard MF method only considers minimizing the distance between positive items and the user. The implementation based on Netflix Prize shows an improvement of BPR compared with the standard matrix factorization.

Markov chain is commonly applied to build a sequential recommendation system. Especially, many basic recommendation systems are based on the first-order Markov chains, which only pay attention to the last interacted item. Factorized Markov Chain (FMC) uses the matrix factorization method based on the first-order Markov chain to generate the recommendation. Specifically, this approach factorizes the item-transition matrix into two low-dimensional item embeddings, then, use the embedding of the last visited item to calculate a score of all other items. In addition, Rendle et al. [14] in 2010 proposed a more advanced Factorizing Personalized Markov Chains(FPMC) for sequential recommendation extending from the factorized Markov chains. FPMC factorizes the user-item matrix and item-item transition matrix to get the user embedding and the item embedding, the first factorization aims to obtain the long-term user preference and the second factorization aims to capture the short-term dependency as the same as FMC. This method shows an obvious improvement compared with the normal FMC and standard matrix factorization methods. Although all these first order Markov chains-based approaches can generate highly explainable recommendations but due to the limitation of first order Markov chains, they are hard to capture the long-term dependency in the behaviour sequence effectively.

In recent years, with the development of machine learning algorithms, deep learning methods especially some sequential modelling methods including RNNs and transformers are demonstrated to be more effective when processing sequential data. Therefore, current research on sequential recommendation systems focuses on the improvement of sequential modelling methods. Some famous approaches to sequential recommendation will be introduced below.

GRU4Rec is the first model applying recurrent neural networks (RNN) on the recommender systems proposed by Hidasi et al. [3] in 2016, which is a significant milestone in the recommender systems. GRU4Rec use the gated recurrent unit (GRU) architecture to handle the user behaviour sequence to better capture the long dependencies in the interaction sequence, which significantly outperforms the traditional machine learning methods, especially handling long sequence. This model processes the historical sequence of the user and uses the final state of the GRU to represent the user preference, the final recommendation represented by calculating the score of the user preference and all item embeddings. The main advantage of this method is that it can better handle long sequences of user interactions, which is significant because users' history may have a long sequence. GRU4Rec has been shown to outperform other state-of-the-art recommendation algorithms at the time in terms of recommendation accuracy. Although GRU4Rec improve the progress of handling long sequence it is still hard to handle very long sequence and it also lacks the interpretability of its recommendation, the same as many other neural networks, it is hard to know why the model predicts a specific recommendation list. Moreover, GRU4Rec could still suffer from cold-start problems because GRU4Rec relies on users' historical sequence to make a prediction. If the user has no previous interaction or does not have sufficient historical actions, the model will not perform well.

As the self-attention mechanism is proposed by Vaswani [15] in 2017, the transformer becomes a very popular architecture in sequential modelling methods. **SASRec** is the first transformer-based recommender system proposed by Kang et al. [4] shows a better performance in handling very long sequences and enhances the interpretability of the recommender systems. The self-attention mechanism with causal masking computes the user representation by allowing the system to capture the most relevant items in the historical actions of the user regardless of their temporal distance, then, computes the weighted sum of different item embeddings in the previous sequence to build a more reliable user representation. Meanwhile, the self-attention mechanism calculates the weight of different items in the input sequence separately, which gives a clear view of the importance of different items. Therefore, SASRec also has better interpretability. However, SASRec has more learnable parameters, which could cause a higher computational cost. Additionally, Although the self-attention mechanism provides a parallel computational ability to accelerate the training avoiding sequentially pushing through the items, calculating the user representation of the current state still involves all previous items in the sequence, which could cause latency in the online server [10].

PinnerFormer proposed by Pancha et al. [7] in 2022 introduce the window-based prediction method, and shows an improvement in recommender systems. Similar to SASRec, the model handles the sequence by the same transformer architecture to calculate a sequence of user embeddings. Different from the SASRec, which uses only the final user representation is used to generate the recommendation list, PinnerFormer utilises the intermediate user representation, which aims to predict a randomly selected positive action from the set of all positive actions within the time window, this dense all action prediction also improve the signal pass in the network. The time window architecture and dense all action prediction introduced by Pinnerformer improve the performance compared with SASRec. Whereas, the dense all action prediction considers the intermediate user representation, which could increase the computational cost of the system.

3 Programme and Methodology

This project proposes a novel multi-day window prediction method in recommender systems to improve the performance of sequential recommendation. Excluding dissertation writing, this project can be divided into 4 work packages (WPs). First WP focus on data handling, second on building a next-item prediction sequential recommender, third on extending the constructed next-item prediction model to a multi-day window prediction model and the final WP consider the experiment results and results analysis.

Data handling: To build a more robust system, we need to find a sufficient number of training data, MovieLens or Retailrocket datasets could be the potential datasets for our project. Due to the specific requirements of our task, we need to handle the dataset in some different ways for next-item prediction and Multi-day window prediction. In Both methods, to make the training process effective the length of the input sequence should at least reach a pre-defined threshold. Additionally, some interesting and potential experiments could be carried out to figure out the model performance with the threshold setting (minimum sequence length). Further, different train/dev/test splitting strategies [16] could be explored and selected for our task. Standard SASRec [4] chooses the item-based split method while other models may prefer the user-based split strategy. In our project, we should choose an appropriate data-splitting strategy to achieve better performance. Robust augmentation [9] proposed in 2021 introduce many data augmentation approaches to handle the data, which could also be a potential way to address some problems of the dataset.

Next-item prediction model: After handling the dataset, we could obtain suitable training, dev, and test datasets for next-item prediction and multi-day window prediction. The implementation part will start with implementing a transformer-based sequential recommender that can be used to train on our dataset. We already find some related Python code implementations of the recommender system in GitHub and its Pytorch version. Therefore, the main task of this work package focuses on implementing an existing model and doing some modifications to potentially improve the model performance on our dataset. After implementing the sequential recommender, evaluate the model performance on different metrics [2] such as Recall@k, HR@k and nDCG@k. Additionally, We could also try some potential improvement methods to demonstrate whether these approaches could achieve better performance in our model. For instance, Bengio et al. [8] proposed scheduled sampling in 2015, which could mitigate the discrepancy between the training and inference, it introduces a sampling strategy in which the model will not always choose the target input as its input but is more likely to choose its own prediction as the training process. And Mihaylova et al. [17] in 2019 introduced an approach to apply the scheduled sampling in transformer, which may also help to improve the performance of transformer-based recommender systems. We will also evaluate this approach in different evaluation metrics mentioned above to check whether it can improve our recommender system. Moreover, Pi et al. [10] proposed a User Interest Center (UIC) architecture in 2019, which stores fixed-length memory tensor to avoid reloading the user's raw historical sequence, to address the high latency and storage problem caused by the self-attentive mechanism for real-time inference. This strategy can be implemented in our system and evaluated by comparing the memory storage and model updating speed when a previous user has some new actions.

Multi-Day window prediction model: After implementing a next-item prediction sequential recommender, ideally we also have the useful train/dev/test dataset in the first WP for multi-day window predictor. The third work package focus on extending the next-item prediction method to the multi-day window predictor. The difference in the multi-day window predictor is that for each training data, there are multiple ground truth items. Therefore, as a simple approach, we can create a new dataset making the label of the dataset contain all items within the time window. However, PinnerFormer [7], a recently proposed multi-day window predictor, gives another interesting approach by its dense all action loss function, which seems to be more effective to train the model. This method allows the model to better capture a user's longer-term interests rather than still making the loss function only focus on predicting a single next action, this loss function could also be a potential method to improve our multi-day predictor performance. Because of the different aims of the multi-day window predictor, we also need to adapt the evaluation metrics to evaluate this prediction method, which will be mentioned detailed in the evaluation section.

Further Experiments and Analysis: To further evaluate the model performance of the multi-day window predictor, we will analyse the results of the experiment and try different time windows to further demonstrate the performance of the novel predictor. The final work package will figure out whether the multi-day window prediction could improve the current next-item prediction based on our dataset. Moreover, we can also carry out some experiments with different deployment cycles and temporal likelihood, for instance, the product might be likely to be bought but not within the time window. Overall, in the last WP, we should try to optimize the model performance as possible, research some further experiments and evaluate the model performance with other baseline models.

Limitation: Although we will compare many different baseline models with the multi-day window predictor, the generalizability of the multi-window predictor may be limited by the datasets used because we will only focus on a specific dataset to carry out the experiment. In addition, the computational cost of the multi-day window predictor could significantly increase compared with the next-item prediction method, which may not be afforded by some individuals. Lastly, we will inherit the previous hyperparameter settings of the existing model, but other hyperparameter settings may achieve better performance for multi-day window recommenders.

3.1 Risk Assessment

Risk Name	Likelihood	Severity	Control Measures
Insufficient Data	Low	High	Data augmentation
Computational Resource Constraints	Low	High	Apply for high performance computing
Workload Exceeds Time Limit	Low	High	Develop a detailed project plan
Model Complexity	Medium	Medium	Review the model's complexity and select a suitable model architecture

As shown in table 3.1 above, four risks are identified with different degrees of likelihood and severity. Firstly, sufficient training data is essential for training a deep learning sequential model, if our dataset cannot provide enough data after preprocessing and filtering this will cause a problem in our project, which make the model cannot train sufficiently and suffers from the underfitting problem. Therefore, the severity of this problem is high, but these dataset has already been used multiple times in this field so we think the likelihood should be low. If the situation still occurs, data augmentation could be a useful approach to address the problem. Computational Resource constraints could also be a problem when training a large number of data in a complex deep learning model, but we already have permission to access the MLP cluster and have experience using the GPU, so the likelihood for this problem is low, but if MLP cluster cannot mitigate the problem, it will have a high severity. In this case, we can apply for permission for high performance computing in Bayes Center. The likelihood of workload exceeds time limit is low because we already have a detailed project plan but the severity is high. Lastly, the model complexity may be not very suitable for our project such as BERT4Rec [6], so we need to review the complexity of different models and select a suitable model architecture to start our project, the likelihood and severity are both medium in this problem.

3.2 Ethics

There are no exposed ethical issues with this project using the publicly available dataset. Two potential datasets are both widely used for the development of recommendation systems.

4 Evaluation

Following related work [2, 4, 7], three different automatic evaluation metrics will be performed to evaluate the accuracy of the recommendation including Hit Ratio (HR), recall@k and NDCG@k, which are all commonly used evaluation metrics for evaluating the performance of a ranked list. For diversity, we follow the instruction of Pinnerformer, which is evaluated by Interest Entropy@k. The evaluation result of the test set represents the final model performance in our project. Hit Ratio is a binary matrix representing whether the recommendation list generated by the predictor contains at least one ground truth item. A higher HR value indicates the model is more successful to identify at least one relevant item for the user. Recall@k cut off the recommendation list at position k and then calculates the recall in the top-k recommendation list. The value represents how well the recommender system could identify relevant items from the entire set of relevant items, and the higher value indicates the model is better at identifying the relevant items in the recommendation list. NDCG@k also cuts off the recommendation list at the position and compares the real recommendation list with the idea recommendation list. The value represents how close the real recommendation is to the perfect recommendation, and the higher value indicates the high quality of the recommendation system. Interest Entropy@k calculates the entropy of the user's interest distribution over the recommended items. The higher entropy value indicates a more diverse set of recommended items.

Furthermore, these two models are built differently, For the next-item prediction model, which aims to predict the user's next action, the evaluation should focus on the accuracy of predicting the immediate next single action. Multi-day window prediction method aims to predict whole actions within the given time window, which means we need to adapt the evaluation metrics to consider the multiple items predicted within the time window. For instance, we will average the hit ratios of all target items as the hit ratio of the multi-day window predictor. For recall@k, Pinnerformer [7] also introduces how they use recall@k in the multi-day window predictor, which is calculated by how well the system can retrieve the positive actions (in top k) from a random corpus.

5 Expected Outcomes

We expect the multi-day window predictor could implement successfully by extending from the existing sequential recommendation and explore the potential benefits of multi-day window prediction in recommender systems compared to the standard next-item prediction approach. The performance of multi-day window prediction is expected to provide a more diverse and accurate recommendation compared with the next-item prediction method, which indicates that the multi-day window architecture may provide a higher quality recommendation result to users increasing user satisfaction with extending applications. If the model can perform as expected it may contribute to a wide usage of this method in recommendation systems instead of next-item prediction, which is a novel approach in this field. If the multi-day window predictor can achieve higher performance, it will fix some drawbacks of the traditional next-item prediction, which only focuses on the single next user action and may be easily influenced by short-term variations. If the model can successfully extract the temporal likelihood, it could also extend understanding of the temporal aspects of user behaviour and preferences in recommender systems.

6 Research Plan, Milestones and Deliverables

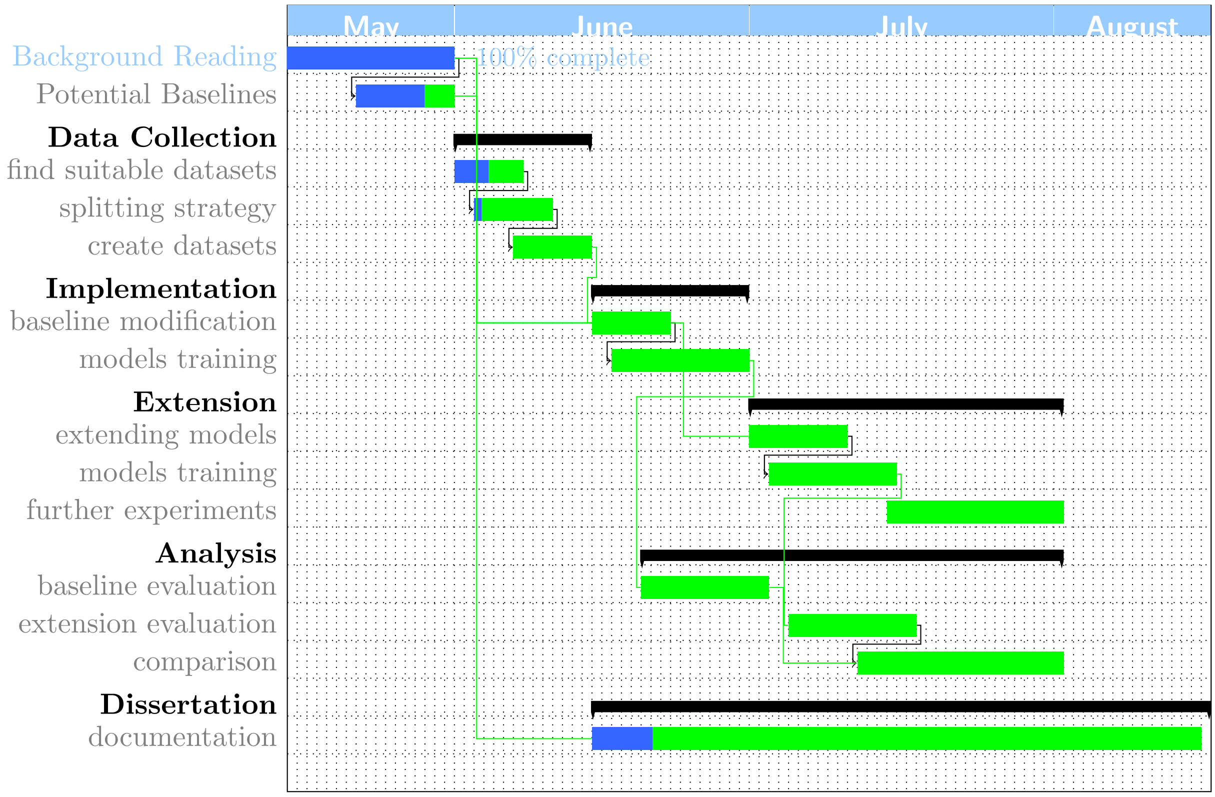

Figure 1: Gantt Chart of the activities defined for this project.

Milestone	Week	Description
M_1	1	Background Study completed
M_2	2	Data collection completed
M_3	4	Baseline model implementation completed
M_4	6	Extension for the time window completed
M_5	8	Evaluation completed
M_6	10	Submission of dissertation

Table 1: Milestones defined in this project.

Deliverable	Week	Description
D_1	2	useful processed dataset
D_2	5	Evaluation report on different baseline models
D_3	7	Evaluation report on multi-day window predictor
D_4	8	Comparison report on baseline models with the proposed model
D_5	10	Dissertation

Table 2: List of deliverables defined in this project.

References

[1] Hyeyoung Ko, Suyeon Lee, Yoonseo Park, and Anna Choi. A survey of recommendation systems: Recommendation models, techniques, and application fields. *Electronics*, 11(1), 2022.

[2] Hui Fang, Danning Zhang, Yiheng Shu, and Guibing Guo. Deep learning for sequential recommendation: Algorithms, influential factors, and evaluations. *ACM Trans. Inf. Syst.*, 39(1), nov 2020.

[3] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk. Session-based recommendations with recurrent neural networks, 2016.

[4] Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation, 2018.

[5] Jiaxi Tang and Ke Wang. Personalized top-n sequential recommendation via convolutional sequence embedding. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, WSDM '18, page 565–573, New York, NY, USA, 2018. Association for Computing Machinery.

[6] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. Bert4rec: Sequential recommendation with bidirectional encoder representations from transformer. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, page 1441–1450, New York, NY, USA, 2019. Association for Computing Machinery.

[7] Nikil Pancha, Andrew Zhai, Jure Leskovec, and Charles Rosenberg. Pinnerformer: Sequence modeling for user representation at pinterest. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, page 3702–3712, New York, NY, USA, 2022. Association for Computing Machinery.

[8] Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. Scheduled sampling for sequence prediction with recurrent neural networks. In C. Cortes, N. Lawrence, D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 28. Curran Associates, Inc., 2015.

[9] Zhiwei Liu, Yongjun Chen, Jia Li, Philip S. Yu, Julian McAuley, and Caiming Xiong. Contrastive self-supervised sequential recommendation with robust augmentation, 2021.

[10] Qi Pi, Weijie Bian, Guorui Zhou, Xiaoqiang Zhu, and Kun Gai. Practice on long sequential user behavior modeling for click-through rate prediction. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery amp; Data Mining*, KDD '19, page 2671–2679, New York, NY, USA, 2019. Association for Computing Machinery.

[11] James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, Mike Lambert, Blake Livingston, and Dasarathi Sampath. The youtube video recommendation system. In *Fourth ACM conference on Recommender systems*, 2010.

[12] Steffen Rendle, Walid Krichene, Li Zhang, and John Anderson. Neural collaborative filtering vs. matrix factorization revisited. In *Proceedings of the 14th ACM Conference on Recommender Systems*, RecSys '20, page 240–248, New York, NY, USA, 2020. Association for Computing Machinery.

[13] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. Bpr: Bayesian personalized ranking from implicit feedback. *arXiv preprint arXiv:1205.2618*, 2012.

[14] Steffen Rendle, Christoph Freudenthaler, and Lars Schmidt-Thieme. Factorizing personalized markov chains for next-basket recommendation. In *Proceedings of the 19th international conference on World wide web*, pages 811–820, 2010.

[15] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. Attention is all you need. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

[16] Zaiqiao Meng, Richard McCreadie, Craig Macdonald, and Iadh Ounis. Exploring data splitting strategies for the evaluation of recommendation models, 2020.

[17] Tsvetomila Mihaylova and André F. T. Martins. Scheduled sampling for transformers, 2019.