Stacking Ensemble Strategy for Click Through Rate Prediction

A DISSERTATION

SUBMITTED IN PARTIAL FULFILMENT OF THE

REQUIREMENTS FOR THE AWARD OF THE

DEGREE OF

MASTERS OF

TECHNOLOGY IN

INFORMATION SYSTEMS

Submitted by

Kritarth Bisht 2K20/ISY/11

Under the supervision of

Dr. Seba Susan

DEPARTMENT OF INFORMATION TECHNOLOGY DELHI TECHNOLOGICAL UNIVERSITY

(Formerly Delhi College of Engineering)

Bawana Road, Delhi-110042

MAY 2022

DECLARATION

I hereby certify that the work which is presented in the Project entitled "**Stacking Ensemble Strategy for Click Through Rate Prediction**" in fulfilment of the requirement for the award of the Degree of Masters of Technology in Information Systems and submitted to the Department of Information Technology, Delhi Technological University, Delhi is an authentic record of my own work, carried out during a period from Jan to May 2022, under the supervision of **Dr. Seba Susan.**

(Kritarth Bisht)

SUPERVISOR CERTIFICATE

I hereby certify that the work which is presented in the Project entitled "**Stacking Ensemble Strategy for Click Through Rate Prediction**". To the best of my knowledge, the above work has not been submitted in part or full for any Degree or Diploma to this University or elsewhere. I further certify that the publication and indexing information given by the students is correct.

Dr. Seba Susan
Supervisor Name and Signature

Place: Delhi
Date:

ABSTRACT

In online advertising and recommender systems, CTR prediction, which seeks to forecast the likelihood that a user will click an item, is critical. The two most prominent strategies for CTR prediction are feature interaction modelling and user interest mining. Both have been extensively investigated for many years and have made significant progress. The main focus of this study will be on examining user interest mining-based models using sequence-based CTR models. In current days, most sequential CTR models employ only recent user behaviour and do not take into account a user's past historical data. Directly using historical data can make model training activities complex and time-consuming. The model that uses historical data cannot do so directly and must rely on the user behaviour extraction module to extract the most important user interactions. To address the aforementioned issue, a stacking ensemble based CTR model is suggested, which stacks or combines the prediction results for each of the long sequence and short sequence based CTR models and uses a meta-learner to train over the forecasted data to obtain the final prediction. Tmall, a real-world industrial dataset, is used to conduct extensive experiments. The experimental results reveal that our suggested strategy outperforms current CTR prediction models significantly.

ACKNOWLEDGEMENT

I am grateful to **Dr. Kapil Sharma**, HOD (Department of Information and Technology), Delhi Technological University (Formerly Delhi College of Engineering), New Delhi and all other faculty members of our department for their astute guidance, and constant encouragement and sincere support for this project work.

I would like to take this opportunity to express my profound gratitude and deep regard to my project mentor **Dr. Seba Susan**, for her exemplary guidance, valuable feedback and constant encouragement throughout the project. Her valuable suggestions were of immense help throughout my project work. Her perspective and criticism kept me working to make this project in a much better way. Working under her was an extremely knowledgeable experience for me.

I would also like to give my sincere gratitude to all my friends for their help and support.

(Kritarth Bisht)

CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: Introduction

There has been some major development in the personalized recommender systems on the Web. The ideal use case for the recommender systems is to retrieve items or information from the large repository or database centered on examining the user's preference, behaviour and desire. The personalized recommender system is being used in a variety of information retrieval cases like item recommendations in e-commerce and also the result re-ranking in the Web Search scenario [1]. Since the item/ result click is the major source for representing and extracting the user behaviour. As a result, Click Through Prediction (CTR) is important in establishing the basic functionality of Information Retrieval Systems. In recent times, many Deep, as well as sequential CTR based models, have been developed. Particularly focussing on the sequential based CTR models, most research has been done in analyzing the CTR results for both long and short-term dependency. But not much focus is on combining the performance of both the dependency sets. Yes, there are models that outperform others in terms of performance measures [2], implying that there is no one highest performing CTR-based model for all users. This suggests that the performance of CTR prediction models is influenced by data from the user. As a result, model design based on user level in CTR prediction systems is both fascinating to research and practical to apply.

We tried to explain the user-level oriented design for increasing prediction performance in this project. From the various sequential CTR models, the goal is to combine various models (mainly three) using the meta learner-based ensemble strategy. For both regression and classification, the ensemble technique is most commonly utilised in machine learning problems. For this project, We will specifically use stacking based ensemble methodology as they have proven to give better performance [20]. The method will consist of two modules, First is the base model where all the trained CTR based models are present and the Second module consists of the meta-learner that will discover the optimal way to integrate the outputs of the basis models [3,4]. For the evaluation, Tmall based dataset is used. It contains a user item click session with a timestamp. The main contribution of the project is mentioned below:

- Analyzing the impact of changing the length (increasing) of the user's previous input sequence.
- Analyzing the CTR performance using Stacking based ensemble strategy for combining various sequential CTR models.

CHAPTER 2: Related Works

In this section, We will discuss things relevant to this project, first is different CTR models that being considered and analyzed in recent years, second on the various ensemble strategies being used to combine the result of CTR prediction models.

2.1 CTR models

In the domain of CTR prediction, user behaviour modelling plays an important role as using the raw features will hardly give any fruitful results. The user behaviour and feature modelling literature has been widely researched. [9,27]. The FM (Factorization Machine) was the initial model for CTR prediction. It represents features using a low-dimensional matrix and learns second-order statistics interaction using the inner product of the user's previous features. Many variants based on FM are proposed in recent years [10,11,12]. Apart from the early features interaction model, deep learning models are also being used in CTR prediction. With the application of DNN, the Neural Factorization Machine [13] improves FM, to map higher order and non-linear features. PNN [14] was the first to introduce the model second and higher order feature interactions. The DeepFM [5] and Wide&Deep [16] use the broad part to learn the lower order features and the deep part to learn the higher order feature. Besides modelling the feature interaction user behaviour modelling is also very important. Most of the current CTR prediction model focuses on learning the user interest by analyzing the user's historical interaction behaviour or in other terms user sequential behaviour. DIN [17] is an Attention-based model that assigns scores to objects with which the user has interacted. DIN assumes user interest to be static and never evolving, thus can lead to an issue like user-interest-drifting. DIEN [7] is a modification to the DIN model that assumes dynamic user interest and employs GRU layers and an auxiliary loss function to grasp the user's developing interest. DSIN [18] employs bi-LSTM and self-attention to capture both inter and intra-session user’s interest. HPMN [15] makes use of a very long user input sequence and is based on memory-based networks, though it effectively maps the long term dependencies but is not very effective in industrial use. The above sequential based CTR models are only good enough to handle users’ short term or recent interests. To extract more relevant interaction behaviour from the user’s long history sequence, various search based CTR models are proposed like UBR4CTR [8] and SIM[19].

2.2 Ensemble-based CTR Models

The ensemble techniques are widely used in supervised prediction problems. One of the ensemble techniques that we used in the project is stacking. In stacking the prediction result of the previous model is given as the input to the next level. For understanding the usage of stacking in the prediction domain we have gone through the literature [20,21,26]. Apart from the normal supervised learning, the ensemble strategy is also being used In CTR prediction. ETCF [22] is

proposed with cascades of the GBDT (Gradient Boosting Decision Tree) and geForest for the CTR prediction task and did not need the much hyper-parameter tunning to get the best result. GFM [23] uses the ensemble of FM and GBDT for CTR or recommendations. The FM is used for linear and second order feature modelling, GBDT is used to model raw features to cross-combined features. Various other novel approaches based on creating a hybrid model using FM and Neural Network includes [24]. Distilled CTR [25] uses the ensemble of various top CTR models. The knowledge distillation methodology is used to aggregate them using a gated network and distil them into a DNN. Distilled CTR has been observed to have shown better results and lower latency, thus good to be used in real-time recommendations.

CHAPTER 3: Methodology

In this section, we are format defining our CTR prediction method based on the ensemble machine learning algorithm i.e stacked generalization or Stacking. Stacking, like bagging and boosting techniques, employs a meta-learning algorithm to learn how to optimally integrate the predictions from many well-performing base models on the same dataset. It produces forecasts that outperform any single CTR model. Unlike bagging, stacking uses various models that fit on the same data rather than samples from the training set. Stacking, indiffernce to boosting uses only one model to learn how to optimally combine the outputs of the base methods, rather than a series of models to correct the outcomes of previous models.
The Stacking architecture consists of two or more weak models or learner, known as level:0 models, and a meta-leaner or model that aggregates the predictions of the weak models, it known as a level:1 model.

3.1 Level 0 Models: These models are trained on training data and their predictions are aggregated and supplied into the Metamodel. They are also called Base Models. There are a total of three CTR models that we have used as the base models, namely CASER, DIEN and UBR4CTR. These models are being explained below.

3.1.1 CASER

CASER stands for Convolutional Sequence Embedding Recommendation model [6]. It employs a convolutional neural network (CNN) to record the dynamic pattern in the user's previous interactions as well as the modifications that it introduces in the user's current activities. Caser's key components are a horizontal and a vertical edge based convNetwork. The vertical convolutional network seeks point-level patterns that show the influence of a given object in the previous sequence of the target object. The horizontal convolutional network aims to uncover the union-level patterns, which indicate the impact of several past actions on the target item.
In Caser, each user has a past sequence of item clicks from the item set, which are ordered based on timestamp. Caser's purpose is to suggest future goods based on the user's short-term engagement behaviour. The embedding matrix to represent the previous interaction for time step t may be formed by considering the last n items:

$$E^{(u,t)} = [Q_{t-n}, \ldots, Q_{t-2}, Q_{t-1}] \qquad (1)$$

Where Q_i represents the item embeddings for the i^{th} row. $E^{(u,t)}$ represents user u's transitory interest at time-stamp t In Caser, we see this matrix E as an image that serves as the input to the next two convolutional components. The horizontal layer m horizontal filter H^j, $1 \leq j \leq m$ and vertical layer n V^j, $1 \leq j \leq n$. The result received after executing a sequence of convolution and pool operations is:

$$o = H_{conv}(E^{(u,t)}, H) \quad (2)$$

$$o' = V_{conv}(E^{(u,t)}, V) \quad (3)$$

Where o and o' are the output of horizontal conv.filters and vertical conv. filters respectively. To obtain higher representations, these outputs are integrated and sent to a mutli layer perceptron.

$$z = \phi(W[o, o'] + b) \quad (4)$$

Where W is the weight matrix, b is the bias and z is the output vector denoting the user's short term intent. Finally, general taste is combined with the short term vector using the prediction fumction:

$$y = v_i.[z, p_j] \quad (5)$$

Where, v_i is the i^{th} row of item embedding matrix V, p_j is the j^{th} row of user embedding matrix for user's general taste and z is the user's short term intent.

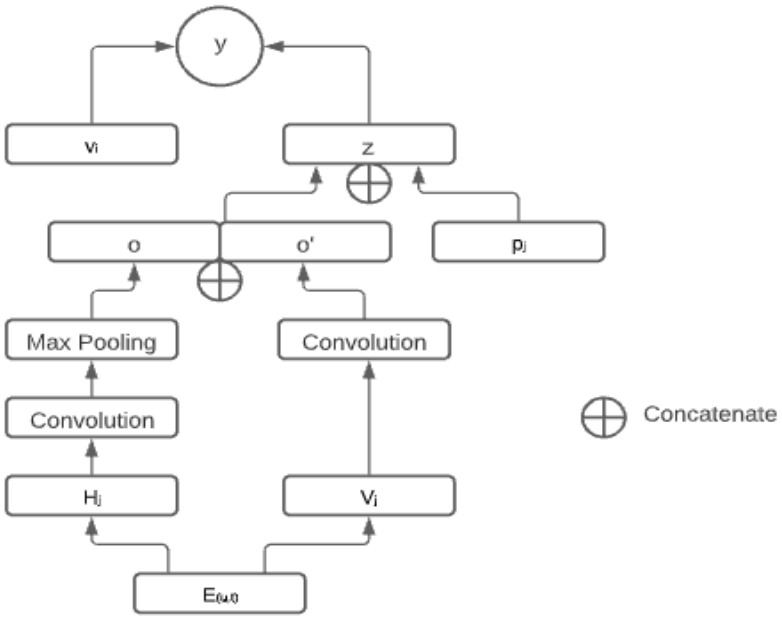

Fig 1 showcase model framework for CASER

3.1.2 DIEN

DIEN stands for Dynamic Interest Evolution Network [7]. Its fundamental architecture is comparable to the DIN model, the only change done is that the processing of the user behaviour sequence is changed and organized into the form of sequence data. DIEN is made up of three levels

or components. The behaviour layer is at the bottom of the stack, and it's used to turn things that the user has previously clicked into embeddings that are sorted by timestamp. The Interest Extractor Layer is the intermediate layer, while the Interest Evolving Layer is the top layer.

Interest Extractor Layer's goal is to extract the interest from embedding data. The user's interest at a given time is not limited to the current behaviour but is also related to the past behaviour. That's Why GRU units are used for the same. The output for GRU units (o_t) is as follows:

$$o' = tanh(W^h i_t + r_t \odot U^h o_{t-1} + b^h) \quad (6)$$

$$o_t = (1 - z_t) \odot o_{t-1} + z_t \odot o_t' \quad (7)$$

Where, z_t and r_t are update gate and reset gate respectively. $\odot$ is an element-wise product, U^h and W^h is hidden size respectively. b stands for user behaviour sequence. I_t means t^{th} user's behaviour that the user took. o_t is the output of GRU and considered the user's extracted initial interest expression. To better extract users' interest, supervised learning is also being introduced. The input to it is embedding vectors based on time steps, e_t and GRU output units o_t and next time step input vector e_{t+1}. Inner-product operation is done with these three inputs to obtain the prediction result and an auxiliary log-loss is introduced.

Interest Evolution Layer's purpose is to comprehend the progression of a user's interest**.** In order to capture the dynamic and developing user's interest, the AUGRU structure utilises an attention mechanism to locally activate the local interest associated to the target item and frame the dependency among these interests.. The attention function can be formulated as:

$$a_t = \frac{exp(o_t W e_i)}{\sum_{j=1}^{T} exp(o_j W e_i)} \quad (8)$$

Where, e_i is the embedding vector for the target item, W is the hidden vector and o_j is the GRU output. a_t denotes the attention score and shows the relationship between the target item and the input o_t. AUGRU is the structure that is used to combine the attention score and GRU. The changes in the GRU gates after incorporating the attention are as follows:

$$z_t' = a_t * z_t \quad (9)$$

$$o_t'' = (1 - z_t') \odot o_{t-1}'' + z_t' \odot o_t' \quad (10)$$

The rest structure of the DIEN is similar to the DIN model

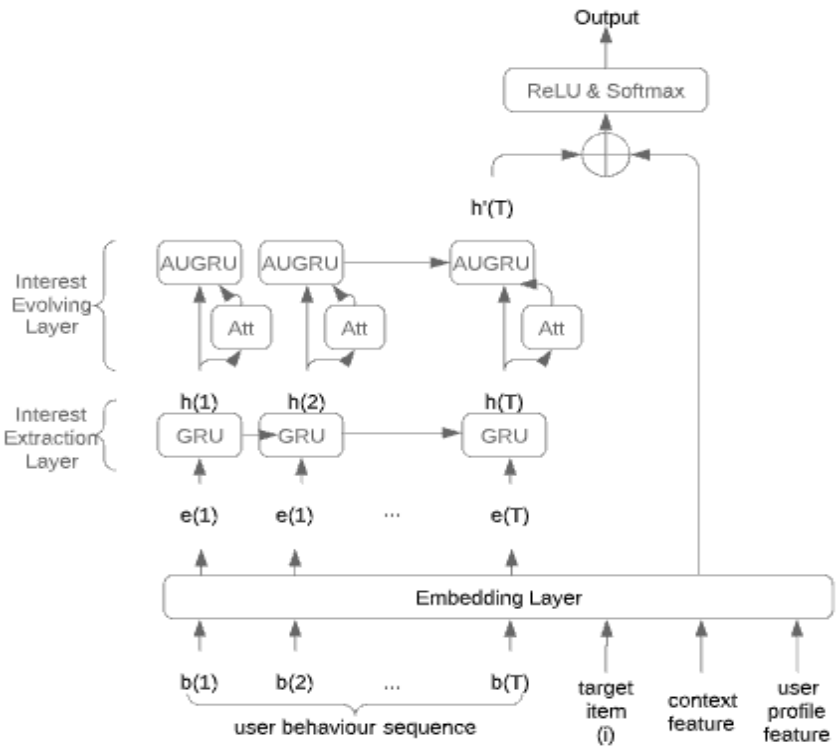

Fig 2 showcase model framework for DIEN

3.1.3 UBR4CTR

UBR4CTR stands for user behaviour retrieval for click through rate prediction. This model is built for mapping long-term user behaviour rather than utilising only N recent behaviours. Instead of modelling complex models to solve long sequence data and prediction, it focuses on the perspective of data. The work consists of first, retrieving a certain number of behaviour sequences from the user's historical data on the target predicted by CTR. The target has 3 parts: target item, target user and context features. Then, another model extract's the most relevant features from the user's historical long sequence data and finally, these extracted features are used to complete the CTR prediction task. The model's main structure is made up of two elements or components: a retrieval module for user behaviour and a prediction module for CTR.

User Behaviour Retrieval module consists of three parts: feature selection model, which is mainly for the features contained in the predicted target such as items features (item id, merchant id, category, time, scenario etc) and user features (gender, location and id) each features selection consider them as the part of features for the search query), search engine client, it is responsible for search algorithms, such as inverted index and BM25 algorithm for scoring) and lastly, user behaviour archive, to store the candidate list of user historical behaviour.

The prediction module uses the multi-layer perceptron as the weight of attention. It adaptively and effectively learns the weight of different historical behaviour for calculating the next item click decision.

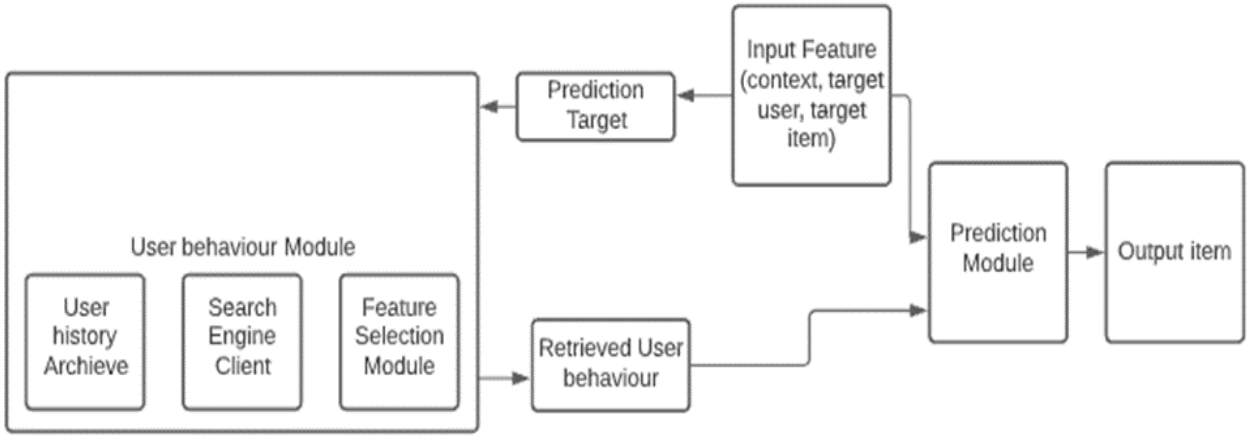

Fig 3 showcase model framework for UBR4CTR

3.2 Level 1 Model: This model learns how to integrate the outputs from the basic models in the most efficient way possible. The Meta-Model is also known as this. The prediction made by the base models on the training data is used to train the meta-model. That is, data is utilised to train the model, predictions are created (in terms of probability values), and these predictions, together with the predicted outputs, are delivered as input-output pairs for fitting in the Meta-Model.
Meta-Model employs a variety of machine learning techniques, including Logistic Regression and Random Forest. The XGBoost Classifier is used as the Meta-Model in this project since it produces superior results. The model's inputs include the Base-Models' predictions as well as training data. Using the training data as the input to the meta-model is beneficial since it gives the meta-model more context.

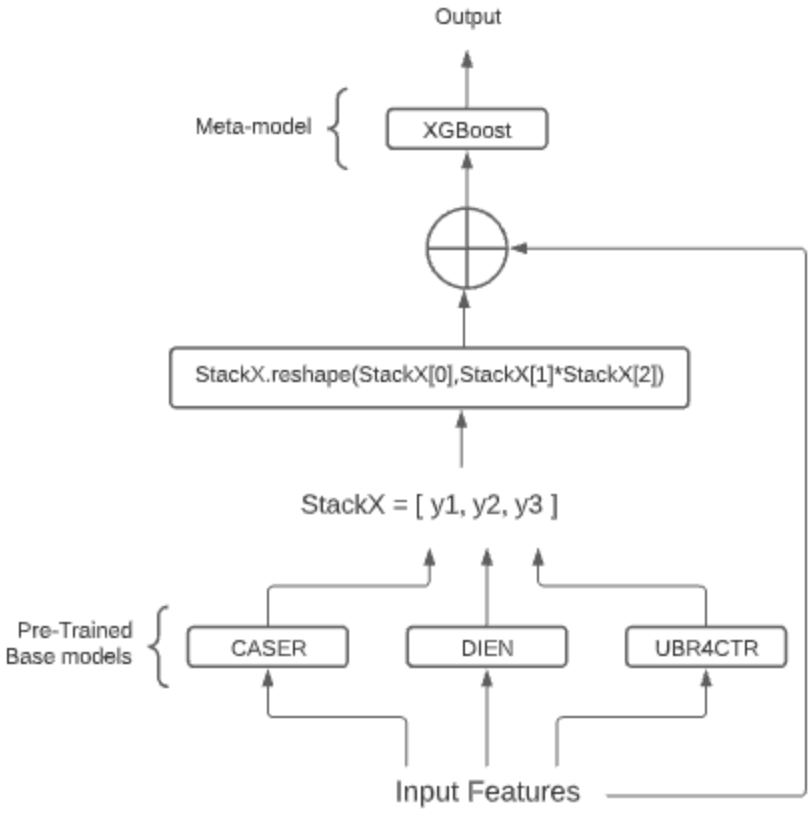

Fig 4 showcase model framework for ensemble based CTR

For the above models, the input features are the user interaction sequence that includes userId, ageId, genderId, itemId, sellerId, categoryId, timestamp etc. These are provided as input to the pre-trained base models. The output of these models is binary probabilities values which are appended to the stack. By feeding the training set into each of the submodels and collecting the predictions, you may create a training dataset for the meta-model. The model provides two predictions in our situation since the result will fall into two categories (0/1). Therefore consider if we have N examples, the result will be 3 arrays of shape [N,2]. If we combine the array the three-dimensional array will be of size [N,3,2]. As there are N example therefore As input to the meta-model, N samples with a certain amount of characteristics. There would be 6(2 x 3) attributes for each case submitted to the meta-model or submodels as the project needed three base models and each model makes two predictions per example, with probabilities of 0 and 1. Thus, the [N,3,2] shaped predictions from the basis models may be transformed into a [N,6] shaped array. After reshaping the stack, the final stack values are the input to the meta-model for training. To further improve the model and add additional context to the meta-model while training over the input features of the base models are also concatenated to the stack values.

CHAPTER 4: Experimental Setup

The experimental setup is discussed in this section. The study questions are first described in section 4.1.. First, the research questions are outlined in §4.1, Next, we describe the dataset that is chosen for the project work and its pre-processing part in §4.2 and §4.3. The evaluation metrics and baseline models used are explained in §4.4 and §4.5 respectively that are being chosen to evaluate the model concerning the other works. At the end of §4.6, we define the parameter setups.

4.1 Research Questions

The following research questions will be addressed in this project:
RQ1: The proposed CTR model outperforms previous CTR based models?
RQ2: How does changing input sequence length for user's histroical behaviour affect models performanace.
RQ3: How does the use of the ensemble strategy to combine different CTR models affect (increase or decrease) the performance of the model.

4.2 Dataset

Tmall is an open log-based session dataset that we used in our research. Tmall.com (a China-based website for B2C), as the database's name implies, is the source of this data and is used in the IJCAI-15 contest. The dataset consists of two tables. The first table comprises information on the user, such as their gender, their age-range and their unique ID. The second, comprises item information such as brand it belong to and also category. The context characteristics such as user timestamp and action type are also included in the dataset. (add-to-cart, add-to-favourite, purchase and click).

Table1: Statistics related to dataset

Dataset	**Tiangong-ST**
#unique users	438,380
#unique items	565,888
#unique instances	4,573,800
#unique features	994,771
#fields	8
collection	Tmall.com (IJCAI-15 contest)

4.3 Dataset Preprocessing

Data preprocessing includes, sorting the items for each user by the timestamp. Following [] the dataset is split for the training and testing. To be specific If there have been N previous user interactions, the behaviour [1, N-3] is utilised for the training set to forecast the target item (N-2nd). Same as for the testing the behaviour [1, N-2] are being used in the validation set for predicting the N-1st target item. The purpose of the validation set is to optimise the model hyperparameters. Lastly, the behaviour [1, N-1] are being used as the testing set for assessing the model performance and to predict the Nth target item.

4.4 Evaluation Metrics

We are comparing the performance of the models using the most traditionally used evaluation metrics, namely Logloss and AUC (Area Under Curve) [5]. The AUC will showcase the pairwise ranking performance between the clicked and not clicked items. In other words, it measures the likeness of assigning the greater scores to the positive sample as compared to the randomly chosen negative sample. AUC with a higher value indicates that the model is doing better.
The next evaluation metric used is Logloss. In the classification task, the log-loss metric is extensively utilised. It displays the test data's total probability. It quantifies the difference between the real and anticipated scores in mathematical terms. A lower Logloss number indicates that the model is doing better.

$$\frac{1}{N}\sum_{i=1}^{N} y_i . log(p(y_i)) + (1 - y_i) . log(1 - p(y_i)) \quad [1]$$

Here, y_i are actual class and $log(p(y_i))$ is the probability of the class

4.5 Baseline Models

The performance of the ensemble-based CTR model was compared to that of different stateof-the-art models, including CASER[6], DIEN[7] and UBR4CTR[8]. The baseline models are implemented the same as being defined in the paper and source code. The parameters related to the dataset (like seed and train-validation-test split) are selected to be the same, rest parameters and hyper-parameters are tweaked and are chosen for the best result.

4.6 Parameter Setups

For Base Model training, the adam optimizer [13] is used to train the basic models with a batch size of 100. For hiding and embedding size the value of 64 is applied. The initial learning rate and regularisation term are both set at 0.0005. Since using the entire interaction behaviour of the user would become hard to train we have used only the last 100 retrieved behaviour of the specific user, In the case of CASER and DIEN the last 100 behaviour are being used and In the case of UBR4CTR the learning module will extract out the most relevant 20 behaviour for a user from

entire user behaviour data. All of the code, training, and testing is done on the Google Colab notebook using PyTorch, and all of the deep learning models are trained on the Nvidia K80 GPU.

CHAPTER 5: Result

We infer the experiment's outcomes in this part to answer the project's proposed research questions. First, we'll illustrate how the implemented model compares to baseline or state-of-the-art models in terms of performance, this will answer the RQ1. For RQ2 used different input sequence length to see how it affect AUC result. Later try to answer the RQ3 by checking various supervised Machine Learning algorithms used in the meta-trainer to combine the result of the base models.

Table2: performance results obtained for all models (up to 4 decimal places)

Model	**AUC**	**Log-Loss**
CASER	0.8907	0.4240
DIEN	0.9102	0.3873
UBR4CTR	0.9189	0.4220
Ensemble Based CTR Model	**0.9327**	**0.3247**

5.1 Performance Comparison

Here, Compare the performance of meta-learner-based CTR models to the performance of several state-of-the-art CTR models. Result for the same can observe in **Table 2** and **Fig 5**. Below is the observation we can derive by inferring the table:

- Ensemble strategy based CTR models yields the best performance on Tmall Dataset in comparison to baseline CTR models. The Ensemble-based CTR outperforms the best performing baseline model i.e UBR4CTR by **1.48%** in terms of AUC and DIEN by **19.27%** in terms of log-loss. The possible reason of Ensemble based CTR model performs best is cause it employs meta-learner models to aggregate the predictions from several base or contributing models, the Ensemble based CTR model may perform well. Furthermore, the meta-learner model is trained on the input to contributing models, their predictions, and the actual output, thereby determining the weightage of each model in the final aggregated predictions for an output.. It also answers ours RQ1 that ensemble based CTR model performing better that the baselines one.
- The CASER based model is the one performing worst among all the baseline models we have used. The reason for that could be cause it is using only the recent user behaviour (like the last 100 interactions for each user) because of which it is only able to grab the short term user behaviour, In contrast, the UBR4CTR is using the user behaviour module to extract out the long term user dependencies (i.e size of retrieved behaviour is 20). This

shows us that the information and interaction behaviour from further history do contain valuable information and pattern about the user's future clicks.

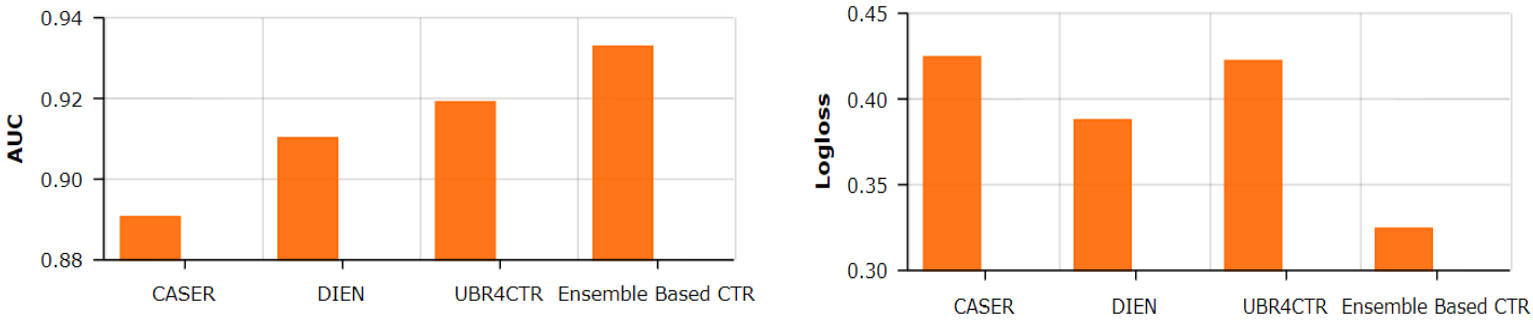

Fig 5: showcase the AUC and Logloss for all models in a Graphical manner

Another variable that can determine the performance of the models is the length of the input sequence of the user used for the models. The ideal length for the input sequence must be the entire history of user behaviour, but if we consider the entire user interaction history the inputs for the model will become too large to handle and train and will make it impossible to use it for industrial purposes, especially in the case when these values are constantly updating. Thus, for baselines models like the CASER and DIEN, I have fixed the length of the recent interaction sequence to consider. To find out which input length to consider, we tried out the various length sequence, and the one which is least costly (training wise) and gives the best result (in terms of AUC) is the one we have chosen. The fig answers our **RQ2** that increasing the input sequence (which means considering further history for the user clicks) does affect the model's CTR performance. Following the **Fig 6** result we have chosen 100 as the input sequence length for every base model.

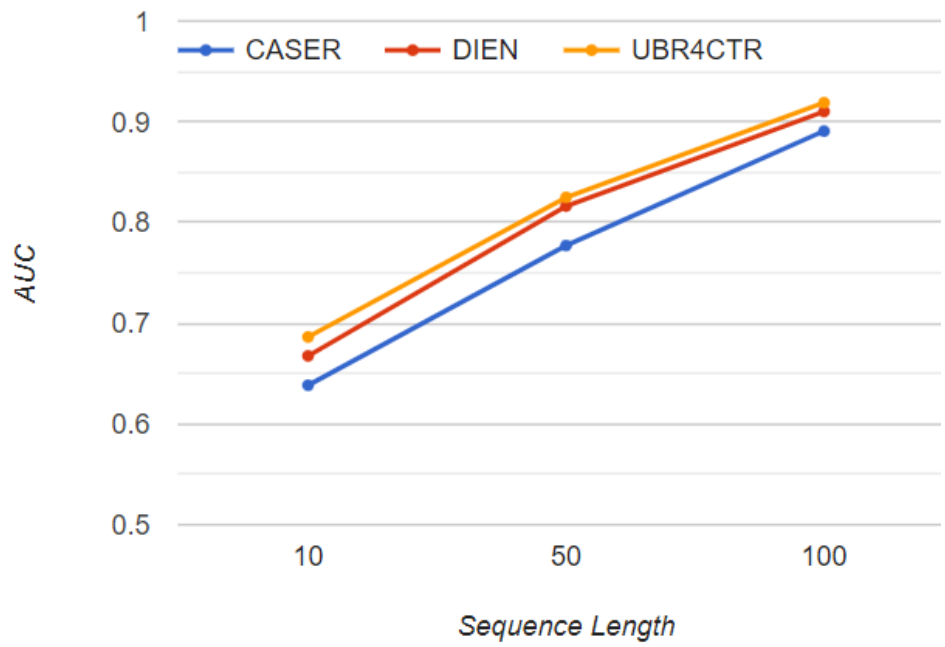

Fig 6 showcase the model performance based on the sequence length of historical behaviour

5.2 Ablation study of Meta-learner model

Apart from using the base CTR models, the machine learning strategy is also used in the meta-learner, one that combines the predictions of the three chosen contributing or base models. Rather than relying just on the base models' predictions as input to the meta-learner, the input supplied to the base models is also used to provide extra context and meaning to the model's input characteristics. For the model supervised machine learning models are being used that include Logistic Regression, XG-Boost and the Random Forest. Considering the result we obtain in **Fig 7**, it can be concluded that the XG-Boost strategy is working best for the usage in the meta-learner and also clarifies our **RQ3**. This is because XG-Boost is based on iterative learning means, it predicts a result initially and then self analyse its mistake and later iteration give more weightage to the input data point which is wrongly predicted. Another reason is cause XG-Boost gives more preference to the functional space for optimising the model. Whereas in the case of random forest and logistic regression more importance is given to hyperparameters for model optimization.

Fig 7 showcases the performance of the CTR model based on different machine learning strategies used in a meta-model

CHAPTER 6: Conclusion and Future Work

The ensemble approach is the focus of this project. Stacking for combining the result of various baseline or contributing models also we tried out various machine learning approaches that we could use in the stacking based meta-learner model. For this project major focus was on implementing baseline CTR models source code and trying to use the most effective machine learning strategy for the meta-learner model to get the improved performance of the CTR predictions. Another finding was that the length of the input sequence of the user's past behaviour is also important in influencing the CTR model's performance. If it cannot be too small, then the model will fail to consider user historical interaction and if it cannot be too large, then the model will take too much time to train and predict and will also not be feasible for the real world use. The overall conclusion is that, based on the experiment and results found in this experiment, the ensemble-based CTR model outperforms the stateof-the-art models.

For future work, more efforts will be put into using the neural network approach for the meta-learner strategy as using a neural network will help to extract the more complex patterns within the predictions of base models and actual output. Another strategy will include varying the size of the input sequence for user behaviour as changing (increasing) the size may improves the performance of the CTR predictions.

References

[1] Fan, Haiyan, and Marshall Scott Poole. "What is personalization? Perspectives on the design and implementation of personalization in information systems." *Journal of Organizational Computing and Electronic Commerce* 16, no. 3-4 (2006): 179-202.

[2] Ekstrand, Michael, and John Riedl. "When recommenders fail: predicting recommender failure for algorithm selection and combination." In *Proceedings of the sixth ACM conference on Recommender systems*, pp. 233-236. 2012.

[3] Andrychowicz, Marcin, Misha Denil, Sergio Gomez, Matthew W. Hoffman, David Pfau, Tom Schaul, Brendan Shillingford, and Nando De Freitas. "Learning to learn by gradient descent by gradient descent." *Advances in neural information processing systems* 29 (2016).

[4] Džeroski, Saso, and Bernard Ženko. "Is combining classifiers with stacking better than selecting the best one?." *Machine learning* 54, no. 3 (2004): 255-273.

[5] Guo, Huifeng, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. "DeepFM: a factorization-machine based neural network for CTR prediction." *arXiv preprint arXiv:1703.04247* (2017).

[6] Tang, Jiaxi, 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*, pp. 565-573. 2018.

[7] Zhou, Guorui, Na Mou, Ying Fan, Qi Pi, Weijie Bian, Chang Zhou, Xiaoqiang Zhu, and Kun Gai. "Deep interest evolution network for click-through rate prediction." In *Proceedings of the AAAI conference on artificial intelligence*, vol. 33, no. 01, pp. 5941-5948. 2019.

[8] Qin, Jiarui, Weinan Zhang, Xin Wu, Jiarui Jin, Yuchen Fang, and Yong Yu. "User behavior retrieval for click-through rate prediction." In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2347-2356. 2020.

[9] Lian, Jianxun, Xiaohuan Zhou, Fuzheng Zhang, Zhongxia Chen, Xing Xie, and Guangzhong Sun. "xdeepfm: Combining explicit and implicit feature interactions for recommender systems." In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 1754-1763. 2018.

[10] Juan, Yuchin, Yong Zhuang, Wei-Sheng Chin, and Chih-Jen Lin. "Field-aware factorization machines for CTR prediction." In *Proceedings of the 10th ACM conference on recommender systems*, pp. 43-50. 2016.

[11] Pan, Junwei, Jian Xu, Alfonso Lobos Ruiz, Wenliang Zhao, Shengjun Pan, Yu Sun, and Quan Lu. "Field-weighted factorization machines for click-through rate prediction in display advertising." In *Proceedings of the 2018 World Wide Web Conference*, pp. 1349-1357. 2018.

[12] Xiao, Jun, Hao Ye, Xiangnan He, Hanwang Zhang, Fei Wu, and Tat-Seng Chua. "Attentional factorization machines: Learning the weight of feature interactions via attention networks." *arXiv preprint arXiv:1708.04617* (2017).

[13] He, Xiangnan, and Tat-Seng Chua. "Neural factorization machines for sparse predictive analytics." In *Proceedings of the 40th International ACM SIGIR conference on Research and Development in Information Retrieval*, pp. 355-364. 2017.
[14] Qu, Yanru, Bohui Fang, Weinan Zhang, Ruiming Tang, Minzhe Niu, Huifeng Guo, Yong Yu, and Xiuqiang He. "Product-based neural networks for user response prediction over multi-field categorical data." *ACM Transactions on Information Systems (TOIS)* 37, no. 1 (2018): 1-35.
[15] Ren, Kan, Jiarui Qin, Yuchen Fang, Weinan Zhang, Lei Zheng, Weijie Bian, Guorui Zhou et al. "Lifelong sequential modeling with personalized memorization for user response prediction." In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 565-574. 2019.
[16] Cheng, Heng-Tze, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson et al. "Wide & deep learning for recommender systems." In *Proceedings of the 1st workshop on deep learning for recommender systems*, pp. 7-10. 2016.
[17] Zhou, Guorui, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. "Deep interest network for click-through rate prediction." In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 1059-1068. 2018.
[18] Feng, Yufei, Fuyu Lv, Weichen Shen, Menghan Wang, Fei Sun, Yu Zhu, and Keping Yang. "Deep session interest network for click-through rate prediction." *arXiv preprint arXiv:1905.06482* (2019).
[19] Pi, Qi, Guorui Zhou, Yujing Zhang, Zhe Wang, Lejian Ren, Ying Fan, Xiaoqiang Zhu, and Kun Gai. "Search-based user interest modeling with lifelong sequential behavior data for click-through rate prediction." In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pp. 2685-2692. 2020.
[20] Pavlyshenko, Bohdan. "Using stacking approaches for machine learning models." In *2018 IEEE Second International Conference on Data Stream Mining & Processing (DSMP)*, pp. 255-258. IEEE, 2018.
[21] Sagi, Omer, and Lior Rokach. "Ensemble learning: A survey." *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery* 8, no. 4 (2018): e1249.
[22] Qiu, Xiaokang, Yuan Zuo, and Guannan Liu. "ETCF: An ensemble model for CTR prediction." In *2018 15th International Conference on Service Systems and Service Management (ICSSSM)*, pp. 1-5. IEEE, 2018.
[23] Wang, Xiaochen, Gang Hu, Haoyang Lin, and Jiayu Sun. "A novel ensemble approach for click-through rate prediction based on factorization machines and gradient boosting decision trees." In *Asia-Pacific Web (APWeb) and Web-Age Information Management (WAIM) Joint International Conference on Web and Big Data*, pp. 152-162. Springer, Cham, 2019.
[24] Toğuç, Hakan, and Rıdvan Salih Kuzu. "Hybrid models of factorization machines with neural networks and their ensembles for click-through rate prediction." In *2020 5th International Conference on Computer Science and Engineering (UBMK)*, pp. 31-36. IEEE, 2020.

[25] Jose, Aljo, and Sujala D. Shetty. "DistilledCTR: Accurate and scalable CTR prediction model through model distillation." *Expert Systems with Applications* (2022): 116474.
[26] Bisht, Kritarth, and Seba Susan. "Weighted Ensemble of Neural and Probabilistic Graphical Models for Click Prediction." In *2021 the 5th International Conference on Information System and Data Mining*, pp. 145-150. 2021.
[27] Bisht, Kritarth, and Seba Susan. "v-TCM: vertical-aware transformer click model for web search." In *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing*, pp. 1917-1920. 2022.