Capturing Performance and Privacy by Assembling Avengers of Online Advertising

Taehee Kim
taehee@corca.ai
Corca, Inc.
Republic of Korea

Seungyun Baek
seungyun@corca.ai
Corca, Inc.
Republic of Korea

Taehyeon Jeon
taehyeonjeon@corca.ai
Corca, Inc.
Republic of Korea

Hojin Jung
hjjung@corca.ai
Corca, Inc.
Republic of Korea

Joonhong Kim
jhkim@corca.ai
Corca, Inc.
Republic of Korea

Taeho Lee
taeho@corca.ai
Corca, Inc.
Republic of Korea

ABSTRACT

This paper presents our solution for the ACM RecSys Challenge '23 (http://www.recsyschallenge.com/2023/) organized by ShareChat, focusing on the domain of online advertising. The goal was to predict the install probability of advertisements, addressing the problem of deep funnel optimization and user privacy issues. Following the flow of impression → (click, install), the data was composed of subsampled impressions with their click and install results, upon which all feature information was anonymized. Our solution tackles this problem by emphasizing three key aspects: improving feature interaction learning in DNN, modeling relationships between click and install, and integrating diverse models to enhance overall performance. Notably, we designed a model architecture of Deep Cross Attentional Factorization Machine (DCAF) that outperformed other leading deep recommender systems such as DCN and DeepFM. With this approach, our team, Corca, accomplished 7th place on the final leaderboard with an NBCE of 6.015522. The implementation details can be accessed on GitHub at https://github.com/corca-ai/recsys-challenge-2023.

CCS CONCEPTS

• **Information systems** → **Recommender systems**; **Online advertising**; • **Computing methodologies** → Neural networks.

KEYWORDS

ACM RecSys Challenge 2023, Recommender Systems, Online Advertising, Feature Engineering, Gradient Boosting for Decision Trees, Neural Networks, Multi-task Learning, Ensemble

ACM Reference Format:
Taehee Kim, Seungyun Baek, Taehyeon Jeon, Hojin Jung, Joonhong Kim, and Taeho Lee. 2023. Capturing Performance and Privacy by Assembling Avengers of Online Advertising. In *ACM RecSys Challenge 2023 (RecSysChallenge '23), September 19, 2023, Singapore, Singapore.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3626221.3627286

RecSysChallenge '23, September 19, 2023, Singapore, Singapore

ACM ISBN 979-8-4007-1613-3/23/09…$15.00
https://doi.org/10.1145/3626221.3627286

1 INTRODUCTION

Online advertising has become a crucial part of any online medium with a significant user base, creating value by exposing personalized advertisements to users with the goal of eventual sales. In the past, advertisers were charged based on impressions or clicks, but the advertising funnel has evolved to focus on actions (install, purchase) that are closer to sales. The ACM RecSys Challenge 2023, organized by ShareChat, aims to optimize this deep funnel by predicting install probability.

The dataset was composed of advertising serving data for 22 consecutive days. The first 21 days were provided with labels (`is_clicked`, `is_installed`) to use for train, with the 22nd day reserved for test.

Table 1: Label distribution

(is_clicked, is_installed)	Distribution
(0, 0)	67.73%
(1, 0)	14.87%
(0, 1)	10.29%
(1, 1)	7.11%

Predicting user response to advertisements has been a long-standing task in the field. Various methods, such as logistic regression [15], factorization machine (FM) [14], and deep learning models [21], have been used to predict click-through rate (CTR). Recent studies have also explored predicting deeper actions, such as install or purchase, by applying multi-task learning to account for preceding actions [9, 10]. However, most of these studies assume sequential behavior, which differs from the Challenge dataset where 59% of installs occur without a previous click. Table 1 shows the distribution of labels.

Another interesting aspect of the dataset was the anonymization of all features and their contents, highlighting the importance of user privacy. In the dataset, only the day information (`f_1`) was provided explicitly, while remaining columns were described solely by their type: categorical (`f_2-f_32`), binary (`f_33-f_41`), and numerical (`f_42-f_79`). Given this level of obscurity, our objective

was to develop a solution that minimizes the need for domain expertise in feature engineering, thus shifting the responsibility of feature interaction modeling to deep learning models.

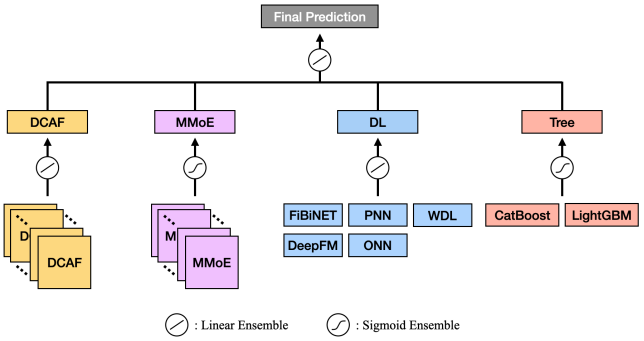

Figure 1: Overall pipeline of the proposed recommender system

To address the challenges above, we introduce a model leveling deep funnel optimization by leveraging deep neural network (DNN) and FM to capture complex feature interactions. Furthermore, we incorporate multi-task learning to model various levels of task relationship between click and install behaviors. We also explore classical approaches using prominent CTR models and Gradient Boosting Decision Tree (GBDT) models. The final result is obtained through a weighted sum of these models.

The Challenge was evaluated based on Normalized Binary Cross Entropy (NBCE), defined as Equation 1. Test data records are defined as N with labels $y_i \in \{0, 1\}$ and estimated probability p_i where $i \in \{1, 2, 3, ...N\}$.

$$NBCE = \frac{-\frac{1}{N}\sum_{i=1}^{N} y_i \log p_i + (1 - y_i) \log (1 - p_i)}{-(p \log p + (1 - p) \log (1 - p))} \quad (1)$$

The remainder of this paper is organized as follows. Section 2 describes the preprocessing and feature engineering techniques applied. Section 3 provides detailed information on the models employed in our solution, followed by ensemble methods in Section 4. Our results and conclusions are shared in Section 5 and 6, respectively.

2 PREPROCESSING AND FEATURE ENGINEERING

2.1 Preprocessing

Null Values. More than a third of the data contained null values, making their imputation crucial. Most of the features were filled with either 0 or mean values. As illustrated in Fig 2, `f_51` in particular showed a differing distribution from others, which we filled with its grouped mean value on another column.

Common Denominator Features. We discovered that majority of numerical features had a common denominator. To ensure consistency in the scale of each data point, each feature was divided by its respective common denominator before applying the logarithmic scale transformation. We have detailed the specific values of common denominator in Appendix A.

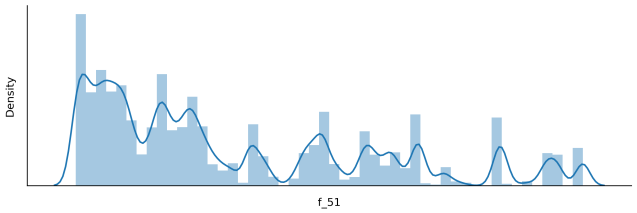

Figure 2: Density distribution for f_51

Outliers. Outliers were treated by applying clipping techniques. For some data points, we employed the Interquartile range (IQR) as a range for clipping. However, in cases where the application of the IQR range resulted in only one remaining value, we opted to use a larger range for a subset of the data.

2.2 Feature Engineering

The unavailability of the feature information posed challenges in creating domain-specific features. Table 2 shows techniques implemented based on data analysis.

Binary Encoding. To tackle the long tail distribution issue [22] in certain numerical features (`f_52` to `f_57`), we applied binary encoding by excluding the most frequent value and setting all others to 1 for each feature. While this approach may cause information loss, it showed promising performance when combined with LightGBM [7].

Feature Grouping. Features of high correlation were grouped for information gain and to discover underlying patterns.

Frequency Encoding. Our dataset showed decrease in the frequency distribution in relation to a metric loss increase. To alleviate this, we applied Inverse Document Frequency (IDF) approach from TF-IDF, which weighs words based on their frequency of occurrence in the bag-of-words [8].

Target Encoding. Target encoding [17] was applied to capture the interaction between categorical features and the target variable. Target encoding and Catboost target encoding [4] were used with smoothing to prevent overfitting. We observed an increase in the NBCE for certain features over time. To capture recency, n recent days were used in encoding.

3 MODELS

3.1 Deep Cross Attentional Factorization Machine

We propose Deep Cross Attentional Factorization Machine (DCAF) as illustrated in Fig 3, which combines a DNN that can automatically learn feature interactions, a cross network that can help understand boundary-degree interactions, and Attentional FM (AFM) [18] that can focus on understanding the relationships between important features. This model effectively captures the essential low-order features and demonstrates substantial improvements in model performance. The leaderboard score improved from 6.148542 to 6.026862.

Table 2: Feature engineering for each model. Each of our models used distinct sets of features based on their characteristics.

	DCAF	MMoE	DL	CatBoost	LightGBM
Binary Encoding	X	X	X	X	O
Feature Grouping	X	f_3-f_4	f_3-f_4	X	f_44-f_47, f_48-f_50
Frequency Encoding	O	O	O	X	X
Target Encoding(TE) (smoothing, recent days)	Mean TE (10, 7)	Catboost TE (5, all)	Catboost TE (5, all)	Mean TE (5, 3)	Mean TE (5, 7) + (5, 3)
Target Encoding - elapsed day	X	O	O	X	X

ReLU activation function and Adam optimizer were used for training with a learning rate of 5e-4 and weight decay of 1e-5. Dropout with a probability 0.1 was set to prevent overfitting.

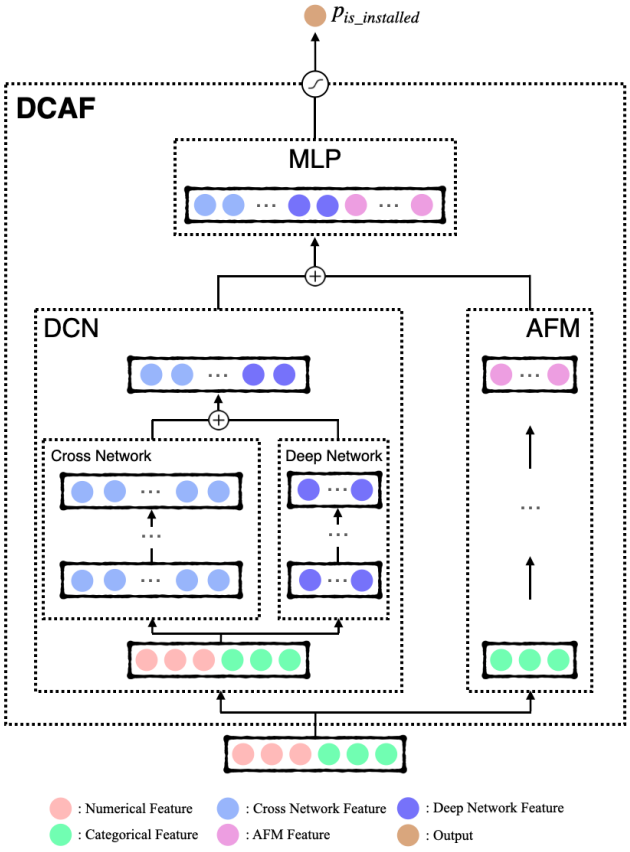

Figure 3: Deep Cross Attentional Factorization Machine architecture

3.2 Multi-Task Model

To leverage both click and install information provided in the dataset, we selected Multi-gate Mixture-of-Experts (MMoE) [9]. MMoE is a model that combines multiple experts using a gate network, enhancing performance with high-correlated tasks. It employs expert networks to learn different aspects of the data and a gating mechanism to determine their influence, improving overall prediction accuracy.

The cases where install events occurred without preceding click events conflicted with prior research [10] assumptions of sequential user actions. For events where installs occurred without any clicks, we modified the data by labeling them as all clicked. This approach enhances the correlation between click and install, leading to improved performance of MMoE.

CatBoost target encoding and inverse frequency encoding were used for feature engineering. MADGRAD [3] was chosen as the optimizer with a learning rate of 1e-4, along with 5-fold cross-validation. It produced a leaderboard score of 6.049729.

3.3 Other Deep Learning Models

We incorporated a range of models, including FiBiNET [6], Product-based Neural Network (PNN) [13], Wide & Deep Learning (WDL) [2], Deep Factorization Machine (DeepFM) [5], and Operation-aware Neural Networks (ONN) [19]. Table 3 shows the individual performance of these models.

As a subsequent step, a simple soft voting technique was employed without assigning any special weights to the models. This average ensemble approach produced a remarkable leaderboard score of 6.049240.

3.4 Tree-Based Models

We tried three GBDT variants: LightGBM, CatBoost [4], and XGBoost [1], Out of the two, LightGBM and CatBoost were used for final prediction due to underperformance of XGBoost. For LightGBM, we applied target encodings based on 3 and 7 recent days, respectively. The CatBoost model was trained separately on each `f_19`, utilizing a day-fold ensemble which is splitting the dataset into K folds by `f_1`(date) and trains K models with K-Fold cross-validation, then combines them into an ensemble. The ensemble of tree-based models achieved a leaderboard score of 6.069782. Table 4 shows the individual performance of tree-based models.

4 ENSEMBLE

Ensemble methods have proven effective numerous times in the past Challenges [8, 16, 20, 23]. During our experimentation, we tested several methods and identified two successful strategies: soft voting mechanism and inverse sigmoid operation. Inverse sigmoid operation is defined as Equation 2, applying σ^{-1} on each logit

Table 3: Individual performance of Deep Learning models

Model	Leaderboard
FiBiNET	6.065765
PNN	6.063737
WDL	6.062937
DeepFM	6.065364
ONN	6.085533
Ensemble	**6.049240**

Table 4: Individual performance of Tree-based models

Model	Leaderboard
LightGBM	6.108842
CatBoost	6.085853
XGBoost	6.153701
Ensemble	**6.069782**

values x_k then soft voting weights w_k and then reapplying sigmoid function σ.

$$\hat{y} = \sigma(\sum w_k \sigma^{-1}(x_k)) \quad (2)$$

Further performance improvement was achieved by combining models with low Pearson correlation coefficient in their outputs, indicating that diversity in model predictions enhances ensemble performance [11, 12]. The pipeline is shown as Fig 1.

Table 5: Performance and weight of model for ensemble

Model	Leaderboard	Weight
DCAF	6.026862	0.55
MMoE	6.049729	0.20
DL	6.049240	0.15
Tree	6.069782	0.10
Ensemble	**6.015522**	

5 RESULTS

Table 5 denotes the performance of each model along with the corresponding weight ratio when ensembled. The DCAF model achieved the highest score as a standalone model, recording 6.026862. By implementing a soft voting technique according to the indicated weights, a score of 6.015522 was attained.

Resource-wise, all models employed in our approach were designed to be lightweight, capable of training on a single NVIDIA RTX 3090 GPU or even CPU for tree-based models. Despite this, for live application cases where facilitating ensemble in production is burdening, taking DCAF as the primary single model is also recommended.

6 CONCLUSION

In this paper, we present our approach for the ACM RecSys Challenge 2023, which focused on predicting install probability from the given advertisements in online advertising. We introduced Deep Cross Attentional Factorization Machine (DCAF) to effectively capture both low- and high-order features even under the circumstance of limited domain knowledge feature engineering due to anonymized data. Also, we analyzed the given task from multiple views, including single and multi-task learning, as well as deep learning models and tree-based models, leading to assembling multiple models. As a result of these strategies, our team, Corca, achieved 7th place on the final leaderboard.

REFERENCES

[1] Tianqi Chen, Tong He, Michael Benesty, Vadim Khotilovich, Yuan Tang, Hyunsu Cho, Kailong Chen, Rory Mitchell, Ignacio Cano, Tianyi Zhou, et al. 2015. Xgboost: extreme gradient boosting. *R package version 0.4-2* 1, 4 (2015), 1–4.

[2] Heng-Tze Cheng, Levent Koc, Jeremiah Harmsen, Tal Shaked, Tushar Chandra, Hrishi Aradhye, Glen Anderson, Greg Corrado, Wei Chai, Mustafa Ispir, et al. 2016. Wide & deep learning for recommender systems. In *Proceedings of the 1st workshop on deep learning for recommender systems*. 7–10.

[3] Aaron Defazio and Samy Jelassi. 2022. Adaptivity without compromise: a momentumized, adaptive, dual averaged gradient method for stochastic optimization. *The Journal of Machine Learning Research* 23, 1 (2022), 6429–6462.

[4] Anna Veronika Dorogush, Vasily Ershov, and Andrey Gulin. 2018. CatBoost: gradient boosting with categorical features support. *arXiv preprint arXiv:1810.11363* (2018).

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

[6] Tongwen Huang, Zhiqi Zhang, and Junlin Zhang. 2019. FiBiNET: combining feature importance and bilinear feature interaction for click-through rate prediction. In *Proceedings of the 13th ACM Conference on Recommender Systems*. 169–177.

[7] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. 2017. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems* 30 (2017).

[8] Hyunsung Lee, Sungwook Yoo, Andrew Yang, Wonjun Jang, and Chiwan Park. 2022. Simple and Efficient Recommendation Strategy for Warm/Cold Sessions for RecSys Challenge 2022. In *Proceedings of the Recommender Systems Challenge 2022*. 50–54.

[9] Jiaqi Ma, Zhe Zhao, Xinyang Yi, Jilin Chen, Lichan Hong, and Ed H Chi. 2018. Modeling task relationships in multi-task learning with multi-gate mixture-of-experts. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*. 1930–1939.

[10] Xiao Ma, Liqin Zhao, Guan Huang, Zhi Wang, Zelin Hu, Xiaoqiang Zhu, and Kun Gai. 2018. Entire space multi-task model: An effective approach for estimating post-click conversion rate. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. 1137–1140.

[11] Giung Nam, Jongmin Yoon, Yoonho Lee, and Juho Lee. 2021. Diversity matters when learning from ensembles. *Advances in neural information processing systems* 34 (2021), 8367–8377.

[12] David Opitz and Jude Shavlik. 1995. Generating accurate and diverse members of a neural-network ensemble. *Advances in neural information processing systems* 8 (1995).

[13] Yanru Qu, Han Cai, Kan Ren, Weinan Zhang, Yong Yu, Ying Wen, and Jun Wang. 2016. Product-based neural networks for user response prediction. In *2016 IEEE 16th international conference on data mining (ICDM)*. IEEE, 1149–1154.

[14] Steffen Rendle. 2010. Factorization machines. In *2010 IEEE International conference on data mining*. IEEE, 995–1000.

[15] Matthew Richardson, Ewa Dominowska, and Robert Ragno. 2007. Predicting clicks: estimating the click-through rate for new ads. In *Proceedings of the 16th international conference on World Wide Web*. 521–530.

[16] Benedikt Schifferer, Jiwei Liu, Sara Rabhi, Gilberto Titericz, Chris Deotte, Gabriel De Souza P. Moreira, Ronay Ak, and Kazuki Onodera. 2022. A Diverse Models Ensemble for Fashion Session-Based Recommendation. In *Proceedings of the Recommender Systems Challenge 2022*. 10–17.

[17] Benedikt Schifferer, Gilberto Titericz, Chris Deotte, Christof Henkel, Kazuki Onodera, Jiwei Liu, Bojan Tunguz, Even Oldridge, Gabriel De Souza Pereira Moreira, and Ahmet Erdem. 2020. GPU Accelerated Feature Engineering and Training for Recommender Systems. In *Proceedings of the Recommender Systems Challenge 2020*. 16–23.

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

[19] Yi Yang, Baile Xu, Shaofeng Shen, Furao Shen, and Jian Zhao. 2020. Operation-aware neural networks for user response prediction. *Neural Networks* 121 (2020), 161–168.

[20] Qi Zhang, Guohao Cai, Wei Guo, Yi Han, Zhenhua Dong, Ruiming Tang, and Liangbi Li. 2022. Fashion Recommendation with a real Recommender System Flow. In *Proceedings of the Recommender Systems Challenge 2022*. 4–9.
[21] Weinan Zhang, Tianming Du, and Jun Wang. 2016. Deep Learning over Multi-field Categorical Data: –A Case Study on User Response Prediction. In *Advances in Information Retrieval: 38th European Conference on IR Research, ECIR 2016, Padua, Italy, March 20–23, 2016. Proceedings 38*. Springer, 45–57.
[22] Yifan Zhang, Bingyi Kang, Bryan Hooi, Shuicheng Yan, and Jiashi Feng. 2023. Deep long-tailed learning: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2023).
[23] Zzh, Wei Zhang, and Wentao. 2022. Industrial Solution in Fashion-domain Recommendation by an Efficient Pipeline using GNN and Lightgbm. In *Proceedings of the Recommender Systems Challenge 2022*. 45–49.

A COMMON DENOMINATOR

Table 6: Common denominator values for each column

Feature	Value
f_42	0.0385640684536896
f_44	0.5711214712545996
f_45	0.5711214712545996
f_46	0.5711214712545996
f_47	0.5711214712545996
f_48	0.5711214712545996
f_49	0.5711214712545996
f_50	0.5711214712545996
f_52	0.0385640684536896
f_53	0.0385640684536896
f_54	0.0385640684536896
f_55	0.0385640684536896
f_56	0.0385640684536896
f_57	0.0385640684536896
f_60	8.07946038858253
f_61	0.1478508992888889
f_62	0.1292997091990755
f_63	0.3552210926047521
f_71	0.5711214712545996
f_72	0.5711214712545996
f_73	0.5711214712545996
f_74	0.0385640684536896
f_75	0.0385640684536896
f_76	0.0385640684536896
f_77	37.38457512430372
f_78	37.38457512430372
f_79	37.38457512430372