Gradient Boosted Tree-based Model for Deep Funnel Optimization Systems

SHARAD AWASTHI*, Delhi Technological University, India
SAMAR PRATAP SINGH SHEKHAWAT*, Delhi Technological University, India
AMIT PANDEY*, International Institute of Information Technology Hyderabad, India
VIKRAM PUDI, International Institute of Information Technology Hyderabad, India

This paper by Team ApheliaAI attempts the task of predicting installation rates based on a user profile. The task was posed as a challenge at RecSys 2023 and aims to improve the deep funnel optimization system for online advertising. The task becomes particularly interesting because it focuses on user privacy, consists of anonymized features, and does not provide the semantics of the individual features. We qualitatively analyze the performance of lightweight gradient-boosted tree-based algorithms for this task and further compare it with the performance of shallow neural networks. We also study the importance of various features for the purpose of feature selection. Our lightweight model requires significantly less computational resources, and we rank 22nd on the leaderboard with a score of 6.665839.

Additional Key Words and Phrases: Deep Funnel Optimization, Random Forest, XGBoost, Feature Selection

ACM Reference Format:
Sharad Awasthi*, Samar Pratap Singh Shekhawat*, Amit Pandey*, and Vikram Pudi. 2025. Gradient Boosted Tree-based Model for Deep Funnel Optimization Systems. 1, 1 (April 2025), 7 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

Online advertising has been a thriving industry that has contributed to the development of the internet. Unlike traditional mass advertising, online advertising can tailor its messages to individual users, making advertising more accessible and inclusive for businesses of all sizes, and offering clear evidence of the return on investment for the advertisers. The task of predicting installation rates plays a crucial role in the field of recommendation systems, enabling platforms to understand and anticipate user behavior. In this paper, we investigate this task using a dataset provided by RecSys 2023, which contains valuable information about user interactions with various items and their corresponding installation rates. The prediction of installation rates is a non-trivial problem due to the complexity of the dataset and the rarity of such studies in the prevalent literature. To address this challenge, we conduct a qualitative analysis of multiple model architectures, aiming to enhance our understanding of their performance characteristics. Notably, our models demonstrate a significant improvement in learning difficult patterns as the experimentation progresses. Furthermore, this paper presents a comprehensive analysis of our methodology, starting with a detailed description

*These authors contributed equally to this work

Authors' addresses: Sharad Awasthi*, Delhi Technological University, Delhi, India, contactsharadon@gmail.com; Samar Pratap Singh Shekhawat*, Delhi Technological University, Delhi, India, samarshekhawat1603@gmail.com; Amit Pandey*, International Institute of Information Technology Hyderabad, Hyderabad, India, amit.pandey@research.iiit.ac.in; Vikram Pudi, International Institute of Information Technology Hyderabad, Hyderabad, India, vikram@iiit.ac.in.

of our data preprocessing steps, including feature selection and normalization. We then delve into the architecture of our models and discuss our process of model selection and hyperparameter tuning. The best-performing model achieves an impressive score of 6.665839 (the metric is explained in section 4) in the installation rate prediction task. To provide further insights, we conduct a qualitative examination of the results, investigating feature importance and model interpretability, which contribute to our understanding of the underlying factors influencing installation rates. The implications of our research extend beyond the current study, providing valuable directions for future research in recommendation systems. The analysis of feature importance sheds light on the most relevant factors affecting installation rates, enabling platforms to prioritize and optimize these features. Lastly, our work emphasizes the significance of model interpretability, addressing the need for transparency and understanding of recommendation system operations. we provide a comprehensive evaluation of the strengths and weaknesses of each method, enabling researchers and practitioners to make informed choices based on their specific requirements and constraints. This paper contributes to the advancement of recommendation systems by investigating the task of predicting installation rates using XGBoost [4]and Random Forest algorithms on complex datasets, such as those provided by RecSys 2023. Our research provides valuable insights into the relevance of different features in predicting installation rates and underscores the importance of model interpretability in recommendation systems. The standings in the Leaderboard for the competition suggest satisfactory achievements, positioning it as a promising solution to address the challenges of predicting installation rates in recommendation systems. The code repository [1] contains all the experiment files.

2 RELATED WORK

We give a succinct overview of the research on deep learning techniques for click-through rate (CTR) prediction in this section. To increase CTR prediction using deep learning techniques, several methods have been developed, including wide and deep learning (WDL) [5], Deep Factorization Machine(DeepFM) [6], eXtreme Deep Factorization Machine (XDeepFM) [9], and Deep and Cross networks [13]. The sequential pattern of user activity in practical recommendation settings are not taken into account by current methods, which instead primarily concentrate on combining features or using various neural network topologies.

To manage users' behavior sequences, Deep Interest Networks (DIN) [14] added an attention mechanism to identify multiple connections between the target item and previously clicked things. [7, 11] use Transformer-based [12] models to address sequential recommendation problems in a sequence-to-sequence fashion.

[10] finds the performance of XGBoost to be better than deep models for some tabular datasets. It also shows that an ensemble of XGBoost and deep neural models outperform the baseline (just XGBoost), which aids in directing the testing model flow in our research.

3 TASK DESCRIPTION

The Sharechat RecSys Challenge 2023 dataset is provided for the task of predicting "is_clicked" and "is_installed" labels in the test set. The dataset includes 30 training files and 1 test file. The training data consists of records that capture user and ad features along with labels indicating click and install events. The test data follows the same format but lacks the label columns. The training data spans 21 consecutive days, while the test data represents the 22nd day. The dataset consists of about 10 million random users who accessed the ShareChat + Moj app during a three-month span. The activity of each user has been sampled, resulting in 10 impressions per user. The dataset covers various features, such as demographic information (anonymized), content preference embeddings, app affinity embeddings, ad categorical features (anonymized), ad embeddings, and count features capturing historical user-ad interactions. The dataset does

not contain the semantics of individual features. Each line in the files is tab-separated, with columns representing features such as categorical, binary, and numerical attributes. Some features may have null values denoted by empty strings. The training dataset contains a total of 3,485,852 instances, providing a diverse set of records for modeling the prediction of clicks and installs based on user and ad characteristics. A Description of the dataset is given below which is extracted directly from the sharechat provided dataset[2]

- The first row is the header row that contains names $f0$ to $f79$ followed by "is_clicked" and "is_installed".
- Each line consists of different columns that are tab separated.
- The data types of different columns are:
  a. RowId ($f0$)
  b. Date ($f1$)
  c. Categorical features ($f2$ to $f32$)
  d. Binary features ($f33$ to $f41$)
  e. Numerical features ($f42$ to $f79$)
  f. Labels ("is_clicked", "is_installed")

4 EVALUATION METRIC

The performance of the recommender system in this research paper is assessed by the Log Loss metric. Log Loss is a metric for binary classification problems. It measures the performance of a classifier by calculating the difference between predicted probabilities and actual binary labels. Let N be the total number of records in the dataset. For each record i, we have the true binary labels y_i ("is_installed") and the predicted probabilities p_i.

$$\text{LogLoss} = -\frac{1}{N}\sum_{i=1}^{N}(y_i \log(p_i) + (1 - y_i)\log(1 - p_i))$$

This metric helps the researchers assess the performance and accuracy of their recommender system objectively and compare it with other methods or models. The Log Loss metric is based on the binary cross-entropy loss function that is commonly used in logistic regression and classification problems and is used to evaluate scores.

5 DATA PREPROCESSING

In the data preprocessing phase for the Sharechat RecSys Challenge 2023 dataset, several important steps are undertaken to ensure the quality and integrity of the data. Firstly, missing values are addressed by dropping rows that contain any NaN values. This is necessary to maintain the purity and consistency of the dataset. As a result, the dataset is reduced to approximately 1.7 million instances from the original 3.4 million.

To further enhance the dataset, a correlation analysis is performed to identify highly correlated columns. The presence of highly correlated columns can lead to multicollinearity, which can negatively impact model performance. Therefore, to mitigate this issue and prevent redundant information, certain columns are dropped from the dataset. By removing these highly correlated columns, the dataset is streamlined and optimized for subsequent modeling and analysis.

By performing these preprocessing steps, the dataset is cleaned and prepared for subsequent model training and evaluation. The resulting dataset contains a subset of impression records with non-null values and is ready for use in developing predictive models for the Sharechat RecSys Challenge 2023. The XGBoost model and Random Forest (TensorFlow) models are given data without dropping the rows of missing values.

Table 1. The Table below shows various models with different configurations where 4 classes in Target Labels mean a linear Combination of Target Labels, i.e., a*("is_installed") + b*("is_clicked") where a and b are real numbers.

Model	Features	Target Labels	Accuracy (%)
MLP	Categorical Features + Numerical Features	4-classes	77
MLP	All Features	4-classes	56
MLP	All Features	"is_installed"	77
MLP	All Features	"is_clicked" + "is_installed"	56
Random Forest	All Features	"is_installed"	85
XGBoost	All Features	"is_installed"	87
XGBoost	Numerical Features+ Binary Features	"is_installed"	85
XGBoost	Categorical Features + Binary Features	"is_installed"	84

6 EXPERIMENTATION AND RESULTS

6.1 Mulltilayer Perceptron

Our approach for the Sharechat RecSys Challenge 2023 involves experimenting with different machine learning and deep learning models to predict user behavior on the platform. We use a shallow Multilayer Perceptron (MLP) model having six fully connected layers. We further experiment with different activation functions, such as Relu, LeakyRelu, and Sigmoid, with no significant difference in the overall performance. The model employs dropout regularization to prevent overfitting, which randomly drops out some neurons during training. The model uses Adam optimizer and Xavier initialization[3], and trains for 50 epochs with a batch size of 64. The results are shown in Figure 1.

The MLP model trains on Sharechat's dataset and tries to predict two binary labels for each interaction: one being "is_clicked," which indicates whether the user clicks on the content, and "is_installed," which indicates whether the user installs an app after seeing an ad.

The MLP model achieves 55% accuracy in predicting the two target labels "is_clicked" and "is_installed" and 78% accuracy in predicting the one target label "is_installed." The accuracy obtained suggests that though a shallow MLP model can capture some underlying patterns and relationships in the data, it would require a careful selection of features and a more complex model to achieve better results.

6.2 Random Forest

We explore alternative models beyond MLP by experimenting with decision tree models. We evaluate their performance on a subset of the dataset to assess their accuracy. Surprisingly, the results reveal that the Random Forest algorithm outperforms the MLP model. This intriguing finding motivates us to delve deeper into the potential of decision trees for our task. By focusing on Random Forest, we aim to leverage its unique ensemble approach and investigate its ability to capture and exploit the underlying patterns within the data.

The Random Forest algorithm achieves 85% accuracy when trained and tested on a 70/30 split of the given training dataset. This is a significant improvement compared to the MLP model, which only achieves 55% accuracy.

Considering the time-consuming nature of these models, the logical next step is to switch to gradient-based methods such as gradient boosting trees, which offer faster learning and utilize GPU capabilities. Libraries like XGBoost and LightGBM [8] provide efficient implementations of these methods.

After experimenting with Random Forest, the focus shifts to tinkering with Gradient Boosting methods like XGBoost, seeking to further enhance the model's performance.

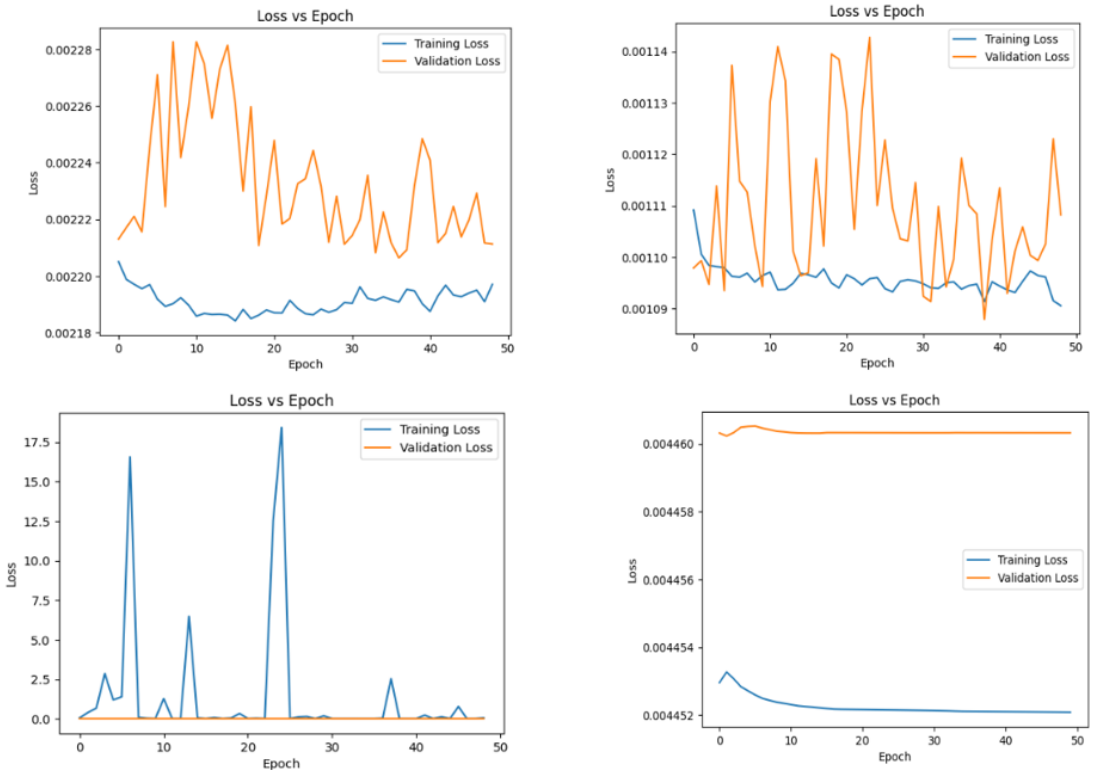

Fig. 1. MLP Plots of various configurations of models, the first plot is for 2 target features "is_clicked" and "is_installed" with only categorical and numerical features, the second takes all features, i.e., categorical, numerical and binary features and modifies the task for multiclass classification by making a linear combination of both target features, the third plot is for the model which only predicts one target feature and the fourth being for multi-label binary classification.

6.3 XGBoost

XGBoost is a gradient-boosted ensemble learning-based model that trains multiple weak learners, and after Random Forest models, this is the next model to evaluate. Since there are many features (80 features), some insights are also important, whether they are about which features are important or which features have more significance. XGBoost can provide such information as well as which features are important in terms of gain and weight. XGBoost has achieved an accuracy of 88% which is better than both MLP and Random Forest. To Know more about the dataset and more insights on features significance, weight importance, and gain importance are extracted from the model and have been shown in figure 2. Both gain importance and weight importance, derived from XGBoost, are crucial for the advancement of models. Gain importance measures the improvement in the model's loss function by incorporating specific features, helping prioritize the most influential ones, reducing dimensionality, and gaining insights into feature relationships. Weight importance, on the other hand, reveals the frequency of feature usage in splitting the data across all trees, assisting in identifying relevant features in imbalanced datasets, assessing feature stability, and effectively handling missing data. By leveraging these metrics, model developers can enhance model performance, streamline feature selection, and improve interpretability, leading to more robust and accurate models with enhanced predictive capabilities.

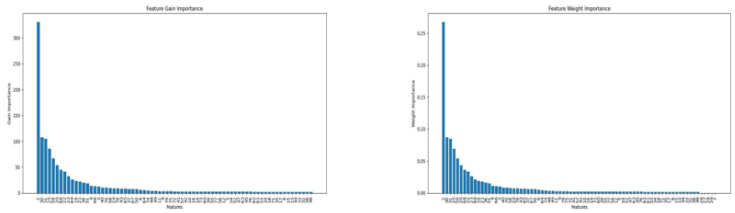

Fig. 2. Two plots depicting Gain Importance and feature importance have similar trends of features. X-axis is the feature name to map these features' names to the features named in the original files just omit f in the name for example- "f-32" is "32" In the plot.

6.4 Feature Engineering

To investigate further, features such as categorical features and numerical features have been individually tested. All features return an accuracy of 87% in the 70/30 split in the given training set. A model with only numerical features along with binary features where categorical features were omitted performed on a similar level, 85%, and when the same is done with categorical features with binary features dropping numerical features, a similar level of accuracy 84% is observed. Since both the plots above are showing similar trends in feature importance, a subset of the features could be used for training more sophisticated models that require higher computational resources, and since there is already a plethora of features for resource-intensive models, only a subset of features that contribute to learning should be used for training. Trends in figure 1 are used for the MLP model, which gives similar accuracy to that already observed with other features but is unable to capture intricate learning patterns.

Table 2. Feature variations with XGBoost

Features	Accuracy
All features	87%
Numerical Features + Binary Features	85%
Categorical Features + Binary Features	84%

7 CONCLUSION

The paper compares three machine learning models: MLP, Random Forest, and XGBoost, and evaluates their accuracy on the test data. We find that lighweight XGBoost performs well, achieving an accuracy of 88% on the test subset from the given dataset for training, and the score achieved on the leaderboard is 6.665839. XGBoost also offers more insights into the importance of different features for user behavior prediction. The work shows the potential of XGBoost for improving recommendations and user experiences on platforms like Sharechat. Due to computational resource constraints, we could not experiment with larger models but it would be interesting to further analyze how attention scores in Transfomer-based models correlate with the feature importances derived from XGBoost.

REFERENCES

[1] [n. d.]. repo. https://github.com/SharadAwasthi369/Recsys-2023.
[2] [n. d.]. Website Title. https://sharechat.com/recsys2023.
[3] Wadii Boulila, Maha Driss, Mohamed Al-Sarem, Faisal Saeed, and Moez Krichen. 2021. Weight initialization techniques for deep learning algorithms in remote sensing: Recent trends and future perspectives. In *2021 IEEE International Geoscience and Remote Sensing Symposium (IGARSS)*. IEEE, 1–4.

[4] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. *arXiv preprint arXiv:1603.02754* (2016).

[5] 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.

[6] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. 2017. DeepFM: a factorization-machine based neural network for CTR prediction. In *IJCAI*. 1725–1731.

[7] Wang-Cheng Kang and Julian McAuley. 2018. Self-attentive sequential recommendation. In *ICDM*. 197–206.

[8] 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. In *Advances in Neural Information Processing Systems*. 3146–3154. https://proceedings.neurips.cc/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf

[9] Jianxun Lian, Xiaohuan Zhou, Fuzheng Zhang, Zhongxia Chen, Xing Xie, and Guangzhong Sun. 2018. xDeepFM: Combining explicit and implicit feature interactions for recommender systems. In *KDD*. 1754–1763.

[10] Amitai Armon Ravid Shwartz-Ziv. 2021. TABULAR DATA: DEEP LEARNING IS NOT ALL YOU NEED. *arXiv:2106.03253v2 [cs.LG] 23 Nov 2021* (2021).

[11] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, and Peng Jiang. 2019. BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer. In *arXiv preprint arXiv:1904.06690*.

[12] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*. 5998–6008.

[13] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. 2017. Deep & Cross Network for Ad Click Predictions. In *Proceedings of the ADKDD'17*.

[14] Guorui Zhou, Xiaoqiang Zhu, Chenru Song, Ying Fan, Han Zhu, Xiao Ma, Yanghui Yan, Junqi Jin, Han Li, and Kun Gai. 2018. Deep interest network for click-through rate prediction. In *KDD*. 1059–1068.