IMPACT: Interpretable Machine-learning Prediction for Ad Click-Throughs

Kartik Kanotra
New York University
kk5243@nyu.edu

Rahul Singhal
New York University
rs8960@nyu.edu

Tanmay Khandelwal
New York University
tk3309@nyu.edu

Abstract

Click-through rate (CTR) prediction is crucial in digital advertising to optimize user engagement and maximize revenue. While advanced models like Wide & Deep and LightGBM have achieved high accuracy, their lack of interpretability limits real-world applicability, particularly in contexts requiring transparency and accountability. To address this gap, we present IMPACT, a framework that balances interpretability and predictive performance for CTR prediction. Our approach incorporates exploratory data analysis (EDA) to uncover critical feature interactions, advanced feature engineering to enhance contextual and temporal representations, and rigorous hyperparameter tuning to optimize models such as LightGBM and XGBoost. Using SHAP (SHapley Additive exPlanations), we provide actionable insights into feature importance, ensuring model transparency. Experiments on the Avazu dataset demonstrate that our method achieves a log-loss of 0.3915 on the test set, outperforming existing baselines while delivering detailed interpretability. This work paves the way for trust and scalability in real-world CTR prediction, ensuring broader applicability across high-stakes domains like finance and healthcare.

1 Introduction

Predicting click-through rate (CTR) is at the core of digital advertising. It estimates how likely a user is to click on an ad, helping platforms deliver more relevant ads and maximize revenue. Over the years, simple machine learning methods like logistic regression have been widely used for this task [1]. However, while these methods work well for simpler patterns, they often struggle with complex interactions between features, limiting their accuracy.

Recent advances in deep learning have brought a major leap forward in CTR prediction. Models like Google's Wide & Deep combine traditional approaches with deep neural networks, allowing them to handle both simple and complex patterns in data [2]. Further improvements, such as DeepFM [3] and xDeepFM [4], have pushed the boundaries even further, making it possible to capture interactions between multiple features in ways that were not previously possible.

Despite these advancements, challenges still persist. One of the biggest issues is that deep learning models often act as "black boxes," making it hard to understand why they make certain predictions. This lack of transparency is particularly problematic in areas like finance or healthcare, where trust and accountability are critical [5]. Additionally, these models can be computationally expensive, requiring significant resources to operate in real-time systems [6].

In this work, we address these challenges by proposing a CTR prediction framework that balances interpretability and accuracy. To better understand the dataset, we performed extensive exploratory data analysis (EDA) to identify meaningful patterns and combinations of features that boost model performance. Key insights from EDA guided the development of additional hand-crafted features, which were combined with original features to improve the predictive power of our models.

We experimented with six different models, including logistic regression, decision trees, factorization machines, random forests, XGBoost, and LightGBM, and performed extensive hyperparameter tuning to optimize their performance. To improve interpretability, we used SHAP (SHapley Additive exPlanations) to analyze feature importance across models. [7] This allowed us to identify which features had the most significant impact on predictions, enhancing transparency and trustworthiness.

Our results demonstrate that the proposed approach achieves competitive accuracy while providing clear insights into feature contributions. The LightGBM model achieved the best performance with a log-loss of 0.3915, outperforming other models in both accuracy and interpretability. By integrating efficient computations with meaningful feature explanations, this study provides a practical solution for scalable and interpretable CTR prediction.

2 Related Work

Deep learning and factorization machine (FM) models have been widely used in CTR prediction, but they come with specific challenges. FM models, such as Factorization Machines (FM) [8] and DeepFM [3], effectively model low-order and high-order feature interactions, but they heavily rely on the completeness of the raw input feature set. Without exploratory data analysis (EDA), critical patterns in user behavior or contextual information are often missed, resulting in suboptimal performance. To address this, we performed an in-depth EDA on the dataset, uncovering latent patterns and relationships. This allowed us to design five handcrafted features that captured essential contextual and temporal trends, significantly improving the performance of our models by enriching the data representation.

Another challenge lies in the reliance on deep learning models like Wide & Deep [2] and Deep & Cross Network (DCN) [6] on hyperparameter tuning to achieve optimal performance. Many previous works did not perform exhaustive hyperparameter searches, instead relying on default or minimally adjusted settings, which limited the full potential of these models. This oversight often resulted in suboptimal accuracy and increased computational inefficiencies, making these approaches less practical for real-time applications. In contrast, we conducted a systematic and comprehensive hyperparameter search across all models, optimizing parameters such as learning rate, regularization strength, and network depth. This rigorous tuning allowed us to unlock the true potential of these models, achieving higher accuracy while reducing computational costs, and making them scalable for real-time CTR prediction scenarios.

Moreover, while FM and deep learning models have demonstrated significant advances in capturing feature interactions, their lack of interpretability has often been a concern. Deep models, in particular, are frequently seen as "black boxes," making it difficult to understand the contribution of individual features to predictions. To address this issue, we integrated SHAP-based analysis [7] to explain feature importance across models. This approach provided actionable insights into the role of each feature, enhancing trust and transparency in the model's predictions while maintaining high accuracy.

By tackling the limitations of incomplete feature exploration, insufficient hyperparameter tuning, and poor interpretability, our approach bridges the gaps in current CTR prediction methodologies. Through the combination of EDA-driven feature engineering, rigorous optimization, and explainable AI techniques, our work not only improves model performance but also ensures scalability and transparency, making it a robust solution for modern CTR prediction challenges.

3 Dataset

In this work, we have used the Avazu Click-Through-Rate Prediction dataset [9], which utilizes 11 days of data (5 million data points), split 80%-20% for training (4 million) and testing (1 million). The Avazu Ads Dataset [9] serves as a benchmark for CTR prediction, enabling models to generalize ad click behavior by combining user, site, app, and device features. It includes a structured set of features that capture the attributes of online ads, supporting model development to predict ad clicks based on historical patterns.

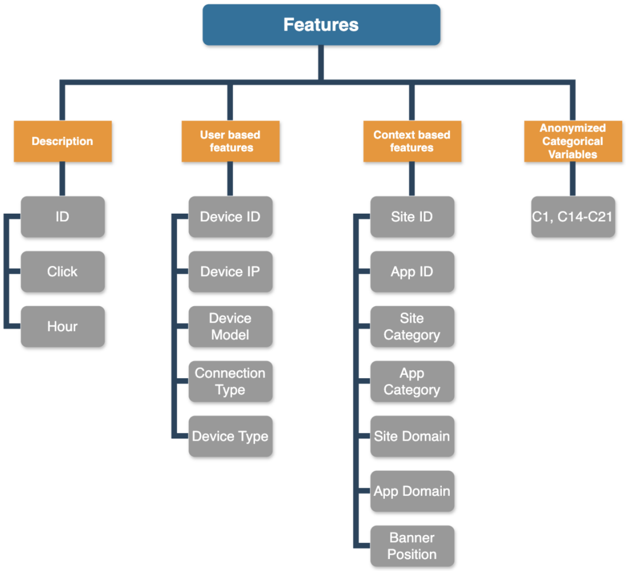

Figure 1: Dataset overview

Table 1: Dataset Overview and Data Fields

Section	**Description**
Dataset Parts	**Train**: 10 days of click-through data (balanced clicks/non-clicks). **Test**: 1 day of click-through data for evaluation.
Data Fields	**id**: Unique ad ID. **click**: Binary (1 for click, 0 for no click). **hour**: Timestamp `YYMMDDHH` (e.g., `14091123`).
Anonymized and Contextual Variables	**C1, C14–C21**: Anonymized categorical vars. **banner_pos, site_id, site_domain, site_category**: Site info. **app_id, app_domain, app_category**: App info. **device_id, device_ip, device_model, device_type, device_conn_type**: Device info.

4 Method

4.1 Exploratory Data Analysis (EDA)

We have conducted extensive exploratory data analysis (EDA) to investigate the key features influencing click-through rates (CTR) in the dataset. This involved analyzing individual, contextual, anonymized, and temporal features, as well as engineered features, to identify trends and optimize predictive modeling.

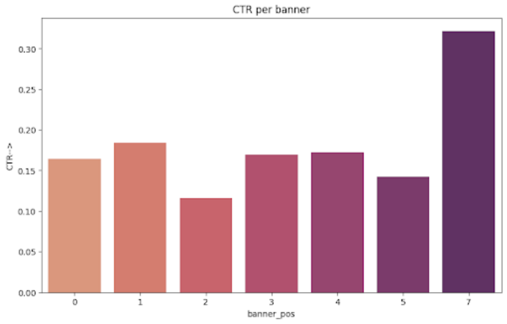

Figure 2: CTR per banner position.

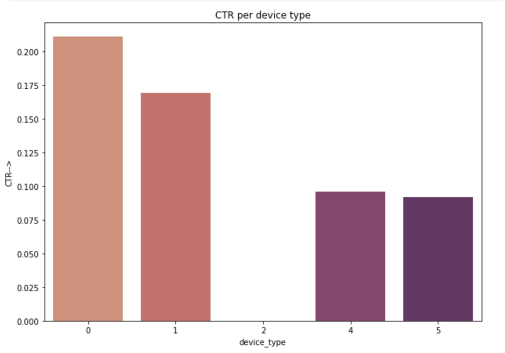

Figure 3: CTR per device type.

Individual features such as banner_pos and device_type revealed critical insights into user behavior. As shown in Figure 2, CTR varies significantly with banner position, with positions 1 and 7 achieving the highest engagement. This indicates the importance of ad placement in driving user clicks. Similarly, Figure 3 highlights that device type 0 outperforms other device types in CTR, possibly due to platform-specific user preferences or interface optimizations.

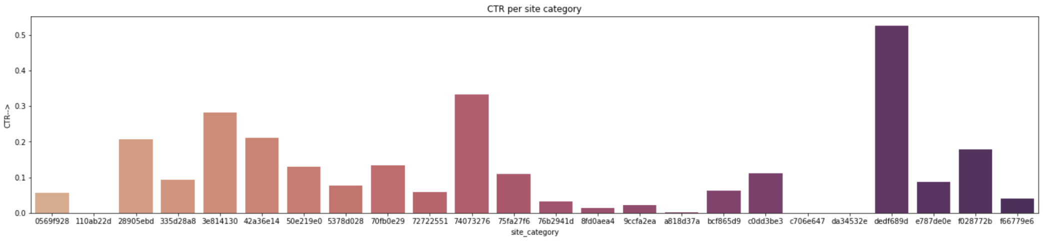

Figure 4: CTR per site category.

Contextual features like site_category provided a strong signal for user intent, as depicted in Figure 4. This insight is vital for targeting ads on high-performing content categories. Anonymized features such as C1 and the resolution-based attributes C15 and C16 further demonstrated their predictive power. For example, Figure 5 shows that ads with higher resolutions (e.g. (300, 250) in C15 and C16) are associated with higher CTRs, while Figure 6 reveals variations in CTR across specific C1 values, indicating the value of latent behavioral patterns encoded in these features.

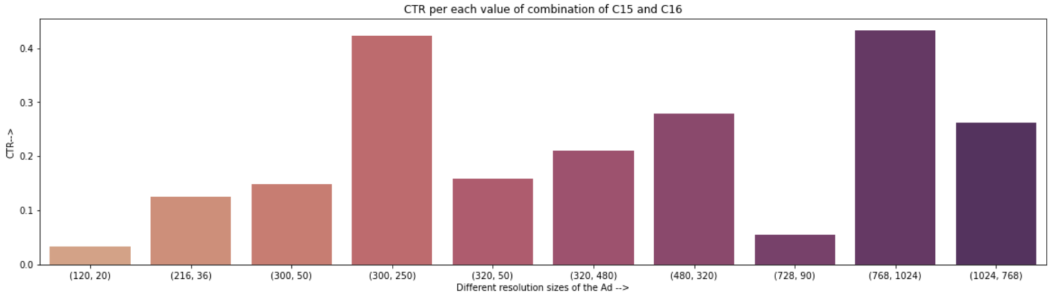

Figure 5: CTR plots for combination of C15 and C16.

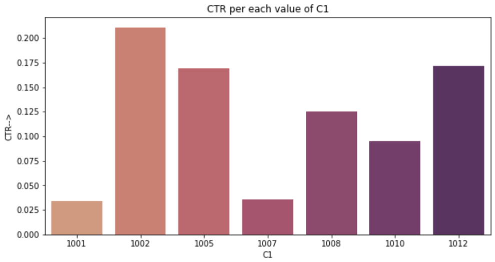

Figure 6: CTR plot for varying values of C1

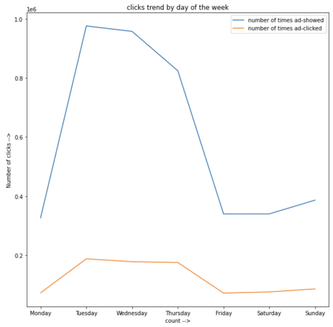

Figure 7: CTR trends by day of the week.

Temporal trends played a crucial role in understanding user activity and engagement. Figure 7 illustrates that CTR peaks on Tuesdays, with a gradual decline toward the weekend, reflecting weekly behavioral trends. Additionally, Figure 8 highlights the importance of active browsing hours, with CTR peaking between 9 AM and 6 PM. These temporal patterns emphasize the need for optimized ad delivery timing. Constructed features like `device_ip_counts` and `hourly_user_count` provided aggregated behavioral insights. As seen in Figure 9, certain IPs show dense activity, reflecting clustered engagement patterns, while Figure 10 highlights that increased user presence during specific hours correlates with higher CTR.

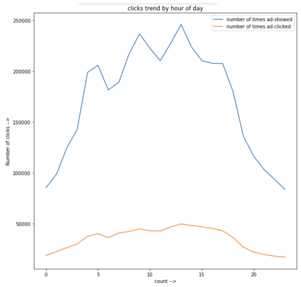

Figure 8: CTR rates based on active browsing hours.

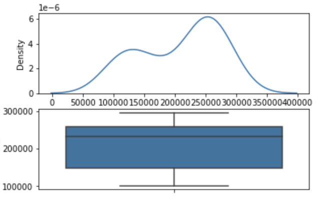

Figure 9: Density based on device IP's.

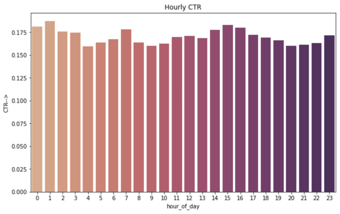

Figure 10: CTR trends based on hour of the day

Moreover, the analysis revealed that feature engineering significantly boosts predictive accuracy. Combining contextual features (`site_category`, `app_category`), temporal splits (`hour_of_day`, `day_of_week`), and anonymized attributes (`C1`, `C15`, `C16`) resulted in improved model performance. Constructed features such as `device_ip_counts` and hourly trends added further depth to the analysis, capturing nuanced user behaviors. These findings underscore the importance of diverse feature representations in building robust CTR prediction models.

4.2 Data Preprocessing

We performed several preprocessing steps on the data to ensure its suitability for analysis and modeling. First, due to its large size (approximately 40 million records), a random sample of 5 million records was selected to optimize computational efficiency without compromising the data's representativeness. The hour feature, originally stored in the YYMMDDHH format, was converted to a standard datetime format, enabling temporal analysis and the extraction of time-based patterns. Missing values across all columns were identified, and appropriate handling strategies, such as imputation or exclusion, were applied to maintain data quality. Categorical features were encoded into numerical representations using techniques like label encoding and one-hot encoding to facilitate machine learning model training. Additionally, duplicate records and outliers were addressed to ensure data consistency. These preprocessing steps provided a robust foundation for feature engineering and model implementation.

4.3 Feature Engineering

To improve the accuracy and interpretability of the click-through rate (CTR) prediction models, several feature engineering techniques were employed. These features were designed to capture user behavior, contextual information, and complex interactions that are critical for effective CTR prediction.

Temporal features were extracted from the `hour` field, which was transformed into components such as day of the week, hour of the day, and part of the day (morning, afternoon, evening, and night). These features helped identify temporal trends in user engagement with ads, providing a better understanding of how time impacts CTR.

Interaction features were constructed by combining key categorical variables like `site_category` and `app_category` or `device_type` and `device_conn_type`. These combinations allowed the models to understand relationships between different contextual factors, capturing nuanced user behaviors and preferences.

Frequency-based features were added to represent the popularity of entities such as site_id, app_id, and device_id. By calculating the frequency of occurrences, these features served as proxies for user interaction trends and engagement levels, offering valuable insights for prediction.

To enhance predictive power, historical CTR features were computed for categorical variables like site_category and app_category. These features provided prior probabilities of an ad being clicked, allowing the models to leverage historical patterns and improve prediction accuracy.

Dimensionality reduction techniques were applied to high-cardinality features like device_id and device_ip. Aggregating these features at higher levels (e.g., grouping by device_type) reduced sparsity and computational complexity while retaining meaningful information.

Low-cardinality categorical variables, such as device_type and banner_pos, were one-hot encoded to maintain interpretability and prevent sparsity. Meanwhile, embedding representations were created for high-cardinality features like app_id, site_id, and device_id using matrix factorization techniques. These embeddings captured latent interactions and enriched the feature set, making them particularly beneficial for deep learning models.

Finally, aggregated features were computed at different granularities, such as the average CTR per site or app. These features provided contextual insights that further refined the models' ability to predict CTR.

By implementing these techniques, the feature engineering process significantly improved the quality of the dataset, enabling the models to capture complex interactions and relationships. These engineered features played a critical role in enhancing both the interpretability and accuracy of the proposed CTR prediction models.

4.4 Model Implementation

We aimed to strike a balance between predictive accuracy and interpretability, making the models not only powerful but also practical for decision-making. We started by implementing a simple logistic regression model to set a baseline for performance. Then, we moved on to decision trees and random forests, which allowed us to capture non-linear patterns in the data while still providing insights that were relatively easy to understand.

For more complex interactions between features, we turned to advanced gradient-boosting algorithms like XGBoost and LightGBM. These models are particularly effective for structured data, as they handle high-cardinality categorical features and intricate relationships with ease. To further explore feature interactions, we incorporated a factorization machine model (PyFFM), which efficiently leveraged the dataset's sparsity to create compact and interpretable representations.

To get the most out of these models, we used randomized search to fine-tune hyperparameters like learning rate, tree depth, and regularization strength. This ensured that each model performed at its best. For evaluation, we chose log-loss as the main metric because it measures how well the models predict probabilities in a binary classification task. We also analyzed confusion matrices and feature importance to better understand model behavior.

4.5 Intepretability

Interpretability is crucial for CTR prediction models because it helps build trust and provides actionable insights for stakeholders. To make the models more transparent, we used SHAP (SHapley Additive exPlanations) to analyze feature importance. This allowed us to pinpoint key predictors, such as temporal and user-device features, that had a significant impact on performance.

While tree-based models like decision trees were inherently interpretable, ensemble models like XGBoost and LightGBM required additional tools like SHAP to clarify how predictions were made. By combining these interpretability techniques with strong predictive performance, we struck a balance that made the models both accurate and practical for real-world use. This also informed feature engineering, helping us refine the models further for even better results.

5 Experiments

5.1 Model Results Analysis

We evaluated several models for click-through rate (CTR) prediction, each with distinct strengths and limitations. Logistic Regression served as a baseline, offering simplicity and interpretability but struggling to capture complex patterns. Factorization Machine, relying heavily on feature interactions, underperformed, while Decision Tree struck a balance between interpretability and performance but required careful tuning to avoid overfitting. Ensemble methods like Random Forest and XGBoost excelled in handling non-linear relationships, with LightGBM emerging as the top performer due to its speed and accuracy on large datasets.

Table 2: Results of various predicting models with their features, tuned hyperparameters, and test log-loss

Predicting Model	**Features**	**Hyperparameters (Tuned)**	**Test log-loss**
Logistic Regression (Baseline)	Original Features + 5 New Features	$\alpha(\texttt{regularization}) = 10^{-5}$	0.40063
Factorization Machine	Original Features + 5 New Features	`regularization_lambda`=0.01	0.4463
Decision Tree	Original Features + 5 New Features	`max_depth`=8	0.3980
Random Forest	Original Features + 5 New Features	`n_estimators`=100, `max_depth`=10	0.3966
XGBoost	Original Features + 5 New Features	`best_depth`=10, `best_learning_rate`=0.01, `best_n_estimators`=500	0.3965
LightGBM	Original Features + 5 New Features	`num_leaves`=31, `learning_rate`=0.1, `n_estimators`=100	0.3915

To optimize model performance, hyperparameter tuning was conducted using randomized search with cross-validation. For example, Logistic Regression achieved its best results with `alpha` = 1×10^{-5}, while Decision Tree required `max_depth = 8`. Random Forest performed optimally with `100 estimators` and `max_depth = 10`, whereas XGBoost achieved its best configuration with `500 estimators` and `learning_rate = 0.01`. LightGBM, leveraging its efficiency on large datasets, was fine-tuned with `31 leaves`, `100 estimators`, and `learning_rate = 0.1`.

A critical component of this evaluation was feature engineering, which combined original features like `site_id` and anonymized variables (`C1`–`C21`) with engineered features such as probability-based metrics (pCTR), aggregated counts (`device_id_counts`), and temporal splits (`hour_of_day` and `day_of_week`). These additional features significantly enhanced the predictive power of all models.

Performance was assessed using test log-loss, providing a clear comparison of model effectiveness. Logistic Regression set a baseline with a log-loss of 0.40063, while Factorization Machine lagged behind at 0.4463. Decision Tree improved the baseline with a score of 0.3980. Random Forest and XGBoost further enhanced performance with scores of 0.3966 and 0.3965, respectively. LightGBM, however, achieved the best result at 0.3915, demonstrating its capability to efficiently handle complex features.

These results underscore the trade-offs inherent in model selection. While simpler models like Logistic Regression are valuable for interpretability, LightGBM and XGBoost are better suited for scenarios requiring high accuracy and scalability, making them the ideal choices for CTR prediction on large, feature-rich datasets.

5.2 Interpetability Analysis

We have done a detailed interpretability analysis to understand the role of features in predicting click-through rates (CTR). Using SHAP (SHapley Additive exPlanations) plots, we visualized feature importance across Logistic Regression, XGBoost, and LightGBM (Figure 11), identifying dominant features and their impacts on predictions. Features such as `site_id` and `C14` consistently emerged as key predictors, reflecting their strong correlation with user behavior.

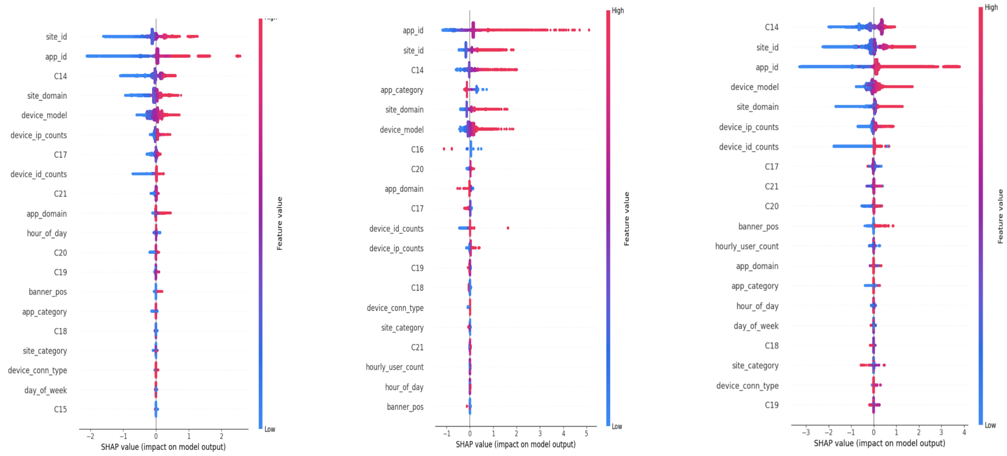

Figure 11: SHAP Analysis

Logistic Regression offered straightforward linear interpretations, with SHAP plots showing clear positive or negative contributions of features. While this simplicity made the model easy to interpret, it struggled with complex feature interactions, limiting its effectiveness for more intricate datasets. Tree-based models, such as Decision Tree and Random Forest, provided more flexibility. Decision Tree focused on a few dominant features, offering clarity but less generalization. Random Forest, with its ensemble approach, distributed importance more evenly across features and better-captured feature interactions.

Boosting models like XGBoost and LightGBM excelled in both accuracy and interpretability. SHAP plots for these models revealed nuanced feature impacts, illustrating both the magnitude and direction of feature contributions. LightGBM made excellent use of engineered features, such as `hour_of_day` and `device_id_counts`, capturing subtle dependencies that other models missed.

In summary, Logistic Regression and Decision Tree are suitable when simplicity and transparency are needed. Random Forest offers a balance between interpretability and performance, while XGBoost and LightGBM are ideal for scenarios requiring high accuracy and detailed insights. These models provide a powerful combination of robust predictions and actionable feature analysis, making them highly effective for CTR optimization.

6 Conclusion

In this study, we introduced IMPACT, a framework designed to predict click-through rates (CTR) with both high accuracy and interpretability. By combining detailed exploratory data analysis, thoughtful feature engineering, and advanced modeling techniques, we were able to identify and leverage critical patterns in the data. Using models like LightGBM and XGBoost, along with SHAP for interpretability, we demonstrated that it's possible to achieve strong predictive performance while maintaining transparency. The results highlight the importance of balancing accuracy with explainability, particularly in applications where trust and accountability are essential. This approach not only improves CTR prediction but also provides valuable insights that can drive better decision-making in real-world advertising scenarios. Future work can explore deep learning models like transformers and graph neural networks to capture richer interactions, combined with interpretability tools to ensure transparency and performance.

References

[1] H Brendan McMahan, Gary Holt, D Sculley, Michael Young, Dietmar Ebner, Julian Grady, Lan Nie, Todd Phillips, Eugene Davydov, Daniel Golovin, et al. Ad click prediction: a view from the trenches. In *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1222–1230. ACM, 2013.

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

[3] Huifeng Guo, Ruiming Tang, Yunming Ye, Zhenguo Li, and Xiuqiang He. Deepfm: A factorization-machine based neural network for ctr prediction. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, pages 1725–1731. AAAI Press, 2017.

[4] Jianxun Lian, Xining 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*, pages 1754–1763. ACM, 2018.

[5] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Why should i trust you? explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144. ACM, 2016.

[6] Ruoxi Wang, Bin Fu, Gang Fu, and Mingliang Wang. Deep & cross network for ad click predictions. In *Proceedings of the ADKDD'17*, pages 1–7. ACM, 2017.

[7] Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in neural information processing systems*, pages 4765–4774, 2017.

[8] Steffen Rendle. Factorization machines. In *Proceedings of the 2010 IEEE International Conference on Data Mining*, pages 995–1000. IEEE, 2010.

[9] Avazu CTR prediction dataset. Accessed: 30 Oct 2024.