Predicting User Click-Through Rate (CTR) on an E-commerce Platform

A Machine Learning Approach

Travis Hughes
Applied Computer Science
The University of Colorado at Boulder
Boulder CO USA
trhu5358@colorado.edu

ABSTRACT

This report details the end-to-end development of a machine learning model for predicting user Click-Through Rate (CTR) on a large-scale e-commerce platform. Our primary goal was to create a robust predictive model that could accurately forecast user engagement with online advertisements, thereby enabling more efficient ad delivery, enhancing the user experience, and improving overall platform revenue. The project followed a comprehensive data science lifecycle, from data acquisition and rigorous preprocessing to advanced feature engineering and the development of a deep learning model. We utilized a comprehensive public dataset from Taobao, a major e-commerce platform, which provided a rich foundation of user, ad, and behavioral data.

The deep learning model, built with Keras and TensorFlow, demonstrated strong learning capabilities during the training phase, with the Area Under the Curve (AUC) score steadily increasing and loss decreasing. However, a significant performance gap emerged when the model was evaluated on a held-out test set, indicating a clear case of overfitting. We have since applied regularization techniques, including Dropout layers and L1/L2 regularization, to mitigate this issue and improve the model's ability to generalize to new, unseen data. The knowledge gained from this project can be directly applied to optimize ad targeting, refine business strategy, and ultimately improve the digital advertising ecosystem for both advertisers and end-users.

CCS CONCEPTS

Computing methodologies ~ Machine learning ~ Supervised learning ~ Classification ~ Applied computing ~ Electronic commerce ~ Information systems ~ Data mining ~ Information systems ~ Information retrieval ~ Click-through prediction

KEYWORDS

Click-Through Rate, CTR Prediction, E-commerce, Machine Learning, Gradient Boosting, LightGBM, XGBoost, Feature Engineering, User Behavior, Online Advertising

1. Introduction

1.1 The Central Role of Click-Through Rate in the Digital Economy

In the modern digital landscape, online advertising has evolved into a multi-billion dollar industry that serves as the lifeblood of e-commerce platforms, social media networks, and content publishers. At the heart of this industry lies a fundamental metric: the Click-Through Rate (CTR). Defined as the ratio of ad clicks to the number of times an ad is displayed (impressions), CTR is more than just a simple statistic; it is a direct measure of an advertisement's effectiveness and relevance. It quantifies the degree to which an ad successfully captures a user's attention and prompts them to take action.

A high CTR is a win-win scenario for all parties involved. For advertisers, it signifies that their ad campaigns are reaching the right audience, leading to a higher return on investment (ROI). For e-commerce platforms, it translates directly into increased revenue from ad sales, which are often priced on a per-click basis (Cost-Per-Click or CPC). Most importantly, for the end-user, a high CTR suggests that the ads being displayed are personalized, relevant, and not overly intrusive, thus enhancing the overall user experience. Conversely, a low CTR indicates a breakdown in this ecosystem. It represents wasted ad spend, diminished platform revenue, and a frustrating user experience characterized by irrelevant ads that contribute to ad fatigue and the growing use of ad-blocking software.

The critical importance of CTR has elevated the task of predicting it from a simple data analysis problem to a central challenge in computational advertising and machine learning. A model that can accurately predict a user's likelihood of clicking on an ad is not merely a tool for forecasting; it is a strategic asset that can intelligently optimize the entire ad delivery pipeline in real-time. This project is motivated by a deep-seated need to move beyond historical campaign data and build a predictive system that can anticipate user behavior with high fidelity.

1.2 The Core Problem: Prediction in a Large-Scale and Highly Imbalanced Environment

The task of CTR prediction is deceptively complex due to several inherent characteristics of the data. First, the scale of ad impressions is immense. A major e-commerce platform can generate hundreds of millions of impression events in a single day, leading to a dataset that is both massive in size and high in dimensionality. Managing and processing this volume of data presents a significant computational challenge.

Second, and perhaps most critically, the data is characterized by extreme class imbalance. The number of users who click on an ad is dwarfed by the number of users who see an ad and do not click. Typically, CTRs range from 1% to 5%, meaning that for every click event, there are 20 to 100 non-click events. This imbalance poses a major obstacle for standard machine learning models, as they can easily achieve a high accuracy score (e.g., 99%) by simply predicting the majority class ("no click") for every impression. A truly effective model must be able to overcome this bias and accurately identify the rare positive events.

Finally, a robust CTR prediction model must be able to capture the complex, non-linear interactions between a multitude of features that

define the context of an ad impression. These features include:

- **User Features**: Demographics (age_level, final_gender_code), past behavior (shopping_level, historical interactions), and user-specific IDs (user_id).
- **Ad Features**: Characteristics of the ad itself, such as its campaign_id, brand, and category_id.
- **Contextual Features**: The time of day, day of the week, or the type of device the user is on.

The challenge lies in the fact that the predictive power is not in any single feature, but in the intricate ways these features interact. For example, a user in a specific age group might be highly likely to click on an ad for a certain product brand, but only if they are viewing the ad during a particular time of day. Building a model that can automatically discover and model these subtle, high-order interactions is at the core of this project.

1.3 Project Goals and Motivations

This project was initiated with a clear set of goals and motivations designed to tackle the challenges outlined above. Our work is not simply an academic exercise but a practical endeavor to build a system with a tangible, positive impact. The primary goals are as follows:

- **Develop a Robust Predictive Mod**el: Our foremost goal is to create a machine learning model that can consistently achieve a high predictive performance on unseen data. This involves moving beyond basic models and exploring advanced architectures, including deep neural networks with embedding layers, which are particularly well-suited for the task.
- **Identify Key Influential Factors**: Beyond just prediction, we aim to gain a deeper understanding of the factors that drive user engagement. By analyzing the features that our model deems most important, we can provide valuable insights to marketing and product teams. This knowledge can inform decisions on ad content creation, targeting strategies, and overall platform design to maximize user interest.
- **Optimize Ad Delivery and Revenue Generation**: The practical application of our model lies in ad ranking. A high-performing CTR model can be integrated into an ad-serving system to dynamically rank ads based on their predicted click probability. This intelligent ranking mechanism ensures that the most relevant ads are shown to the most receptive users, leading to an increase in ad impression efficiency and, consequently, higher revenue for the platform.
- **Enhance User Experience**: The project's success is also measured by its impact on the end-user. By improving the relevance of the ads a user sees, we can create a more personalized and less intrusive experience. This fosters a positive relationship with the platform, reduces ad fatigue, and encourages user loyalty.

This project, therefore, represents a comprehensive effort to leverage state-of-the-art machine learning techniques to solve a critical

business problem. By navigating the complexities of large-scale, imbalanced data, we aim to build a solution that is not only accurate but also provides actionable insights that can drive strategic business decisions. The following sections of this report detail the methodologies employed, the results achieved, and the key findings that emerged from this end-to-end machine learning pipeline.

2. Literature Survey

The field of CTR prediction has a rich history, with a continuous evolution of techniques driven by the exponential growth of data. Early efforts primarily focused on linear models, such as Logistic Regression[1] and Factorization Machines (FMs)[2]. These models served as strong baselines due to their interpretability and computational efficiency. However, their ability to capture complex, non-linear feature interactions was limited.

The rise of tree-based models marked a significant leap forward. Algorithms like Gradient Boosting Machines (GBMs), including LightGBM[3] and XGBoost[4], have become the industry standard for tabular data due to their high performance and ability to automatically discover intricate feature interactions. The advent of deep learning has also introduced models with embedding layers that can handle vast numbers of categorical features, and deep neural networks (DNNs) are now commonly used to model the complex, non-linear relationships in CTR data[5], [6]. Our work builds on this modern approach to push the boundaries of predictive accuracy.

3. Proposed Work

Our project will systematically cover data acquisition, preprocessing, feature engineering, model training, and evaluation.

3.1 Data Collection, Preprocessing, and Derived Data

Initial steps involve ensuring data quality and readiness for training:

- **Data Acquisition & Initial Inspection:** Loaded raw_sample, ad_feature, and user_profile datasets. Performed sanity checks (data types, statistics, unique IDs). Transformed the noclk column to a click (1 for click, 0 for no click) target variable.
- **Data Cleaning:** Handled missing values by imputation with -1 for numerical ID-like columns and 'unknown' for other categorical columns.
- **Data Integration:** Merge raw_sample with ad_feature (on adgroup_id), then merge the result with user_profile (on user/userid) to create a consolidated DataFrame.

3.2 Feature Engineering

This crucial phase transforms raw data into patterns for CTR prediction:

- **Time-based Features:** Extracted hour, day of week, day number from time_stamp. Explore interactions with user/ad attributes.
- **Count/Frequency Features:** Calculated historical click/impression counts and CTRs for users, ads, categories, brands, and campaigns. Compute dynamically to prevent data leakage.
- **Behavioral Features:** Aggregated historical user interactions (e.g., distinct

categories visited, recency, average time between visits/clicks).

- **Interaction Features:** Created cross-features (e.g., user_age_level with ad_category_id).
- **Encoding Categorical Features:** Use One-Hot Encoding for low cardinality; target encoding or embeddings for high cardinality.

3.3 Data Splitting

We adhered to a natural temporal split to prevent data leakage and ensure our model generalizes to future data. The training set consisted of data from the first seven days, while the test set was created from the final day's data.

3.4 Model Training

We explored deep learning models with architectures (e.g., feed-forward with embeddings) to handle the dataset's mixed data types and complexity. The final layer was a single **Dense layer with a sigmoid activation function** to output the probability of a click.

3.5 Hyperparameter Tuning

Systematic tuning will optimize performance using GridSearchCV/RandomizedSearchCV. Bayesian optimization (Optuna, Hyperopt) may be considered for efficiency.

4. Dataset

We utilized a comprehensive dataset provided to Kaggle by Alimama, sourced from Taobao, one of Alibaba's largest e-commerce platforms.

- **Source:** All datasets are available on Kaggle: https://www.kaggle.com/datasets/pavansanagapati/ad-displayclick-data-on-taobaocom[7]
- **Download Status:** All necessary data files have been downloaded to the local machine.

The dataset has three interconnected tables:

1. **raw_sample.csv:** Ad display/click logs for 1.14M users over 8 days (approx. 26M records). Key fields: user_id, time_stamp, adgroup_id, pid, noclk.
2. **ad_feature.csv:** Categorical ad information. Key fields: adgroup_id, category_id, campaign_id, brand, customer_id.
3. **user_profile.csv:** Demographic/behavioral info for 1.06M users. Key fields: userid, cms_segid, cms_group_id, final_gender_code, age_level, pvalue_level, shopping_level, occupation, new_user_class_level.

These datasets provide a rich foundation for exploring CTR factors.

5. Evaluation Methods

Evaluation focused on ranking accuracy and probability prediction:

- **Area Under the Receiver Operating Characteristic Curve (AUC-ROC):** Standard for binary classification, measuring the model's discrimination power and is robust for imbalanced datasets.
- **Log Loss (Cross-Entropy Loss):** Measures probability prediction performance; lower log loss indicates

better calibrated predictions, vital for ranking/bidding.

- **Precision, Recall, F1-Score:** Provide insights at specific thresholds, but AUC-ROC and Log Loss are preferred for overall probability quality.

Evaluation Process: The deep learning model was evaluated exclusively on the test set (2017-05-13), ensuring generalization to future data. Comparison will be made against baselines and literature.

6. Tools

- **Programming Language:** Python
- **Data Manipulation:** Pandas, NumPy
- **Machine Learning Libraries:** Scikit-learn, LightGBM, XGBoost, Keras / TensorFlow
- **Visualization:** Matplotlib, Seaborn
- **Environment:** Jupyter Notebooks

7. Milestones Completed

Milestone 1: Data Acquisition & Initial Preprocessing (July 18) - COMPLETED

This milestone has been successfully completed. The following tasks were achieved:

- **Data Access and Loading**: Confirmed access to raw_sample.csv, ad_feature.csv, and user_profile.csv. All datasets were successfully loaded into Pandas DataFrames.
- **Initial Inspection**: Basic sanity checks were performed, including inspecting data types, initial statistics, and unique IDs.
- **nonclk to click Transformation**: The nonclk column in raw_sample.csv was successfully transformed into a click column, where 1 indicates a click and 0 indicates no click. The original nonclk column was dropped.
- **Data Cleaning (Missing Values)**: Missing values in ad_feature.csv and user_profile.csv were handled. For numerical ID-like columns, missing values were imputed with -1. For other columns, they were filled with 'unknown'.
- **Data Integration**: The three datasets were successfully merged into a single consolidated DataFrame (df_final). This involved:
    - Merging raw_sample with ad_feature on adgroup_id.
    - Renaming the user column in raw_sample to user_id to ensure consistency.
    - Renaming the userid column in user_profile to user_id to ensure consistency.
    - Merging the result with user_profile on the unified user_id column.

The final df_final DataFrame has 26,557,961 entries and 19 columns, with the appropriate data types and initial handling of missing values, ready for feature engineering.

Milestone 2: Core Feature Engineering & Baseline Model (July 25) - COMPLETED

This milestone has been successfully completed. The following tasks were achieved:

- Implemented time-based and initial count/frequency features. Apply temporal split. Train/evaluate Logistic Regression (AUC-ROC, Log Loss).

Milestone 3: Gradient Boosting Model & Initial Optimization (July 30) - COMPLETED

This milestone has been successfully completed. The following tasks were achieved:

- Train LightGBM/XGBoost. Evaluate performance. Conduct initial hyperparameter tuning (RandomizedSearchCV). Analyze feature importance.

Milestone 4: Advanced Feature Engineering & Refined Model (August 4) - COMPLETED

This milestone has been successfully completed. The following tasks were achieved:

- Implement advanced behavioral/interaction features. Integrate into GBM. Refine hyperparameter tuning. Finalize evaluation.

Milestone 5: Documentation & Deep Learning Exploration (Optional) (August 8) - COMPLETED

This milestone has been successfully completed. The following tasks were achieved:

- Explore deep learning models.

8. Remaining Milestones:

- **None**

9. Discussion

Model	AUC-ROC Score	Log Loss Score
LightGBM (Advanced)	0.6944	0.1852
LightGBM (Original)	0.6935	0.1853
Deep Learning Model (Non-Optimized)	0.6916	0.1857
Logistic Regression	0.6802	0.1918
Deep Learning Model (Optimized, Sampled)	0.6397	0.6286

The results from training our deep learning model on the full dataset mark a pivotal moment in this project. Our optimized deep learning model suffered from severe overfitting when trained on a sampled dataset, a common issue for complex models with insufficient data. The non-optimized model, however, has fundamentally changed our findings. Its performance effectively closed the gap with the state-of-the-art LightGBM models, transforming our deep learning exploration from a diagnostic exercise into a successful and highly competitive path.

The most critical insight from this run is the undeniable importance of data volume for deep learning models. The sampled model's failure to generalize (with a test AUC of 0.6397) was a direct consequence of a limited data pool, which forced the complex architecture to memorize noise instead of learning fundamental patterns. By training on the full 26-million-record dataset, the non-optimized model achieved a test AUC-ROC score of 0.6916 and a Log Loss of 0.1857. These scores are nearly identical to the performance of our best LightGBM model (AUC 0.6944, Log Loss 0.1852), demonstrating that deep learning is not only a viable but also a

highly effective solution for this specific problem.

Furthermore, this model exhibits very little overfitting. Its training AUC of 0.7019 is just a small step above its test AUC, indicating that it has learned to generalize well to unseen data. This stability, combined with its high performance, makes it an excellent candidate for the final solution. The deep learning model's ability to learn rich, low-dimensional representations through its embedding layers likely gives it a slight advantage in capturing the complex interactions between high-cardinality categorical features.

The key takeaway is that the perceived superiority of LightGBM was likely a result of its efficiency on smaller datasets and its robust handling of overfitting. With enough data, our deep learning architecture proved its capability to perform on par with—and potentially even surpass—the best tree-based models. This finding validates our initial architectural design and provides a solid foundation for the final phase of the project.

10. Conclusion

The project has now successfully moved beyond all initial milestones, culminating in a highly competitive deep learning model for CTR prediction. The journey from a simple logistic regression baseline to a sophisticated deep learning architecture has provided critical insights into the nature of the data and the capabilities of various modeling techniques. The core finding—that with sufficient data volume, our deep learning model can achieve performance on par with best-in-class gradient boosting models—is the most significant result of this project.

The path forward is no longer about diagnostics, but about building on a proven success. The project will now focus on final refinement. The future work for this project is centered on three key areas:

1. **Advanced Hyperparameter Tuning**: Now that we have a stable, high-performing model, the next step is to apply the regularization and dropout enhancements we designed to this full-dataset model. We will conduct a more extensive and automated hyperparameter search using tools like **Optuna** or **Keras Tuner** to find the optimal combination of regularization parameters, dropout rates, and learning rates. This process is essential for pushing the model's test AUC score higher and extracting every last bit of performance from the architecture.

2. **Ensemble Modeling and Comparison**: The final step before deployment will involve building an ensemble model. The deep learning model and the advanced LightGBM model have demonstrated unique strengths. We can leverage these by combining their predictions in a final ensemble. An ensemble, such as a **stacking classifier**, will produce a more accurate and stable final prediction by leveraging the diverse predictive power of each model. This strategy will provide the

best possible performance and add an extra layer of resilience to the system.

3. **Deployment and Actionable Insights**: The final objective is to move from experimentation to a deployable solution. The focus will shift to serializing the final model and integrating it into a mock ad-serving environment. We will also produce a detailed analysis of feature importance to provide actionable insights for business stakeholders, helping them understand which user and ad characteristics are most influential in driving clicks.

In conclusion, this project has established a strong, end-to-end foundation for CTR prediction. We have successfully navigated the complexities of data preparation and model development, and our final deep learning model has proven its viability. The insights and framework developed here serve as a robust blueprint for creating a production-ready machine learning system that can provide a tangible impact on e-commerce advertising revenue and user experience.

ACKNOWLEDGMENTS

Thanks to my wife, Adele Rehm, and her constant support. Thanks to my mother in law, Pam Rehm, for stepping in to help out when things got tough

REFERENCES

[1] H. B. McMahan 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*, Chicago, IL, USA, 2013. Available: https://static.googleusercontent.com/media/research.google.com/en//pubs/archive/41159.pdf

[2] S. Rendle, "Factorization Machines," in *2010 IEEE International Conference on Data Mining*, Sydney, NSW, Australia, 2010, pp. 995-1000. Available: https://www.ismll.uni-hildesheim.de/pub/pdfs/Rendle2010FM.pdf

[3] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, San Francisco, CA, USA, 2016, pp. 785-794. Available: https://dl.acm.org/doi/10.1145/2939672.2939785

[4] G. Ke et al., "LightGBM: A Highly Efficient Gradient Boosting Decision Tree," in *Advances in Neural Information Processing Systems*, Long Beach, CA, USA, 2017. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf

[5] H.-T. Cheng et al., "Wide & Deep Learning for Recommender Systems," arXiv:1606.07792 [cs.LG], Jun. 2016. Available: https://arxiv.org/abs/1606.07792

[6] H. Guo et al., "DeepFM: A Factorization-Machine based Neural Network for CTR Prediction," arXiv:1703.04247 [cs.LG], Mar. 2017. Available: https://arxiv.org/abs/1703.04247

[7] P. Sanagapati, "Ad Display-Click Data on Taobao.com," Kaggle, [Online]. Available: https://www.kaggle.com/datasets/pavansanagapati/ad-displayclick-data-on-taobaocom. Accessed: 7/4/2025