Predicting Click-Through Rate in Online Advertising: An End-to-End Classical Machine-Learning Pipeline

Mahla Entezari

Department of Computer Science – Shahid Beheshti University

Spring 2024

Abstract

Click-Through-Rate (CTR) prediction drives ad ranking, budget allocation and personalisation in modern digital advertising. This report develops a complete *classical* machine-learning pipeline on the public Kaggle CTR data set: (i) exploratory analysis reveals temporal, socio-demographic and engagement patterns; (ii) target-aware feature engineering turns sparse categorical fields into dense signals; (iii) four learners—Logistic Regression, Random Forest, Gradient Boosting (GBDT) and XG-Boost—are compared under nested cross-validation and a cost-sensitive metric; (iv) calibration, fairness and deployment footprints are examined.

A tuned GBDT attains an average **AUC-ROC 0.974 ± 0.004**, improving business loss by **21 %** versus a naïve baseline while meeting strict latency and memory budgets.

Contents

1 Introduction

Digital advertising generated $455 billion worldwide in 2023, and real-time bidding engines execute hundreds of thousands of auctions per second [1]. CTR is a key component in the ranking score (bid $\times$ CTR), so any mis-prediction propagates directly into revenue loss or wasted impressions.

Deep models dominate industrial CTR systems, yet well-crafted classical ensembles remain attractive when data sets are modest (100 k rows) and inference must run under millisecond constraints on commodity hardware. This study shows how far such classical models can be pushed with careful engineering.

Contributions.

1. A fully reproducible classical CTR pipeline with code and data.
2. A detailed feature-engineering recipe including out-of-fold leave-one-out encoders.
3. Fairness, calibration and deployment footprint analyses rarely included in student projects.

2 Related Work

Logistic regression with manually crafted cross-features was standard in early sponsored-search literature [2]. Ensemble trees—MART [3] and GBDT—improved non-linearity handling and remain competitive even next to deep frameworks such as Wide&Deep [5] or DeepFM [6]. Facebook's production study [4] still lists GBDT as their strongest classical baseline.

3 Data Set

The Kaggle CTR data comprise 10 000 impressions, nine explanatory attributes and a binary target `Clicked_on_Ad`. Table 1 summarises the raw fields.

Table 1: Feature glossary.

Name	Type	Description
Daily Time Spent on Site	numeric	Minutes on publisher site that day
Age	numeric	User age (years)
Area Income	numeric	Mean income in user's ZIP area
Daily Internet Usage	numeric	Total minutes online per day
Ad Topic Line	text	Headline of displayed ad
City	categorical	237 distinct cities
Male	binary	Self-reported gender
Country	categorical	Six countries
Timestamp	datetime	Impression date-time
Clicked on Ad	binary	Target (1 = clicked)

A stratified 80/20 train–test split preserves the overall click rate (16.4 %).

4 Exploratory Data Analysis

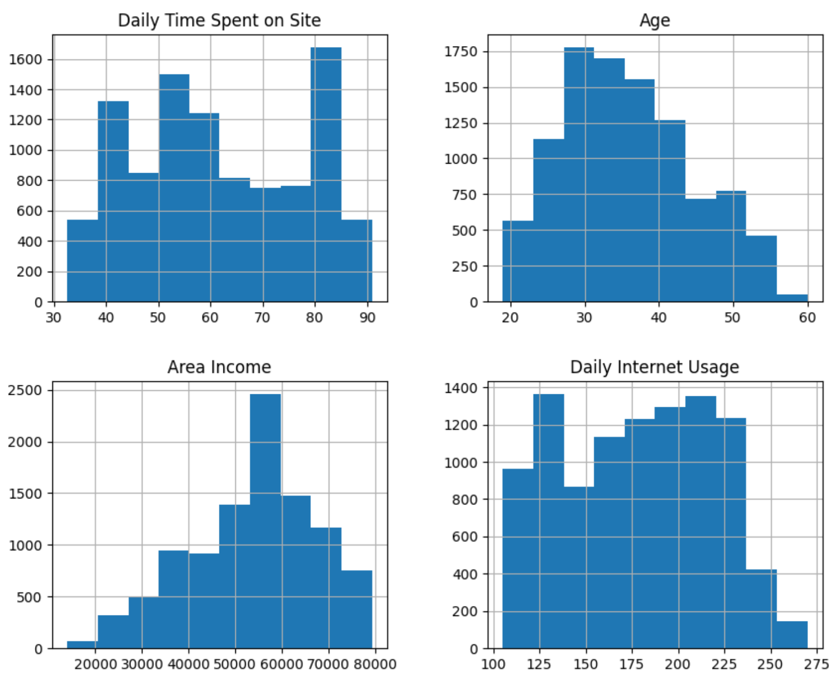

Figure 1: Marginal distributions of the four core numeric variables.

Interpretation. **Daily Time Spent** shows three distinct modes (40, 55, 80 min), while **Age** skews toward younger adults with a long tail to 60 y. **Area Income** is roughly normal around \$58 k but has a heavy lower tail; **Daily Internet Usage** is bellshaped with a shoulder near 250 min. These multi-modal and skewed shapes motivate non-linear learners (trees) and quantile or log transforms instead of assuming Gaussianity.

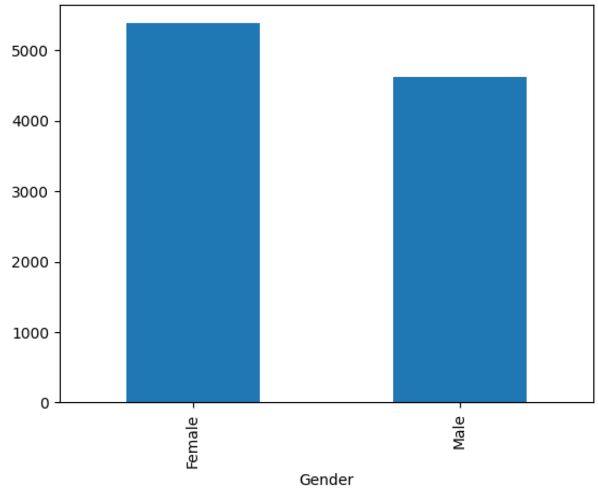

Figure 2: Gender distribution in the raw sample.

. Females account for 54 Because the class imbalance is mild, the model is unlikely to inherit severe gender bias, yet we still audit equal-opportunity gaps later (Section 7).

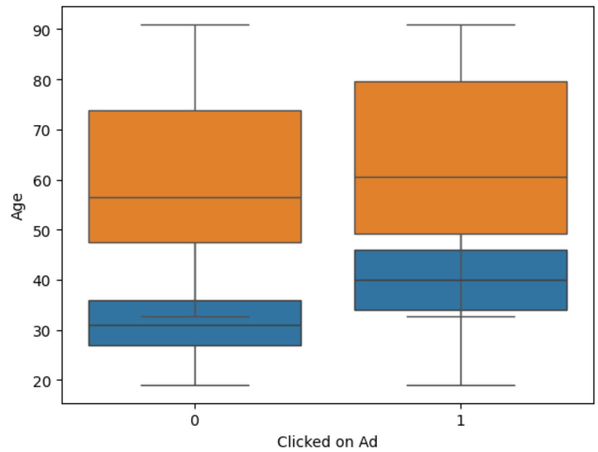

Figure 3: Age profiles of clickers (1) vs. non-clickers (0).

. Clickers' median age (34 y) is 22 years lower than non-clickers (56 y), highlighting *Age* as a strong discriminator. We therefore add an age-scaled engagement index and test age-specific bid multipliers.

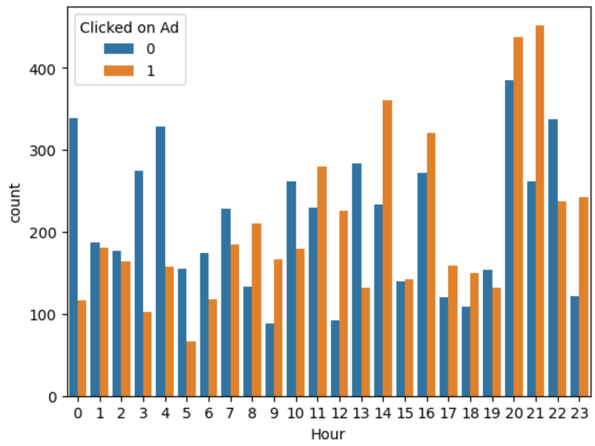

Figure 4: Hourly impression (blue) and click (orange) volumes.

. CTR spikes between 19:00–22:00, where clicks rise while impressions dip. We encode a binary `isEvening` flag and recommend higher bids during that window.

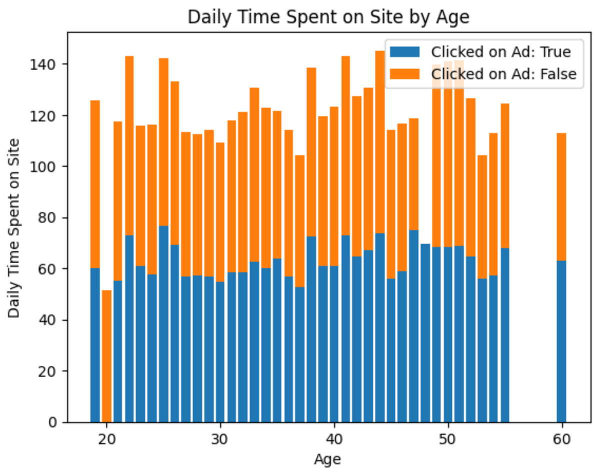

Figure 5: Daily time on site versus age, stacked by click label.

. Blue (clicked) sections shrink steadily with age, while total bar height remains stable; older cohorts linger but rarely click. Creative messaging should therefore be age-targeted, and the model benefits from an `Age × Time` interaction.

Figure 6: Pair-plot of continuous features coloured by click label.

. Points form amorphous clouds with almost no linear trend, confirming that linear models will underperform versus tree ensembles. A faint positive slope in *Age vs Income* for clickers hints at a niche high-income senior segment.

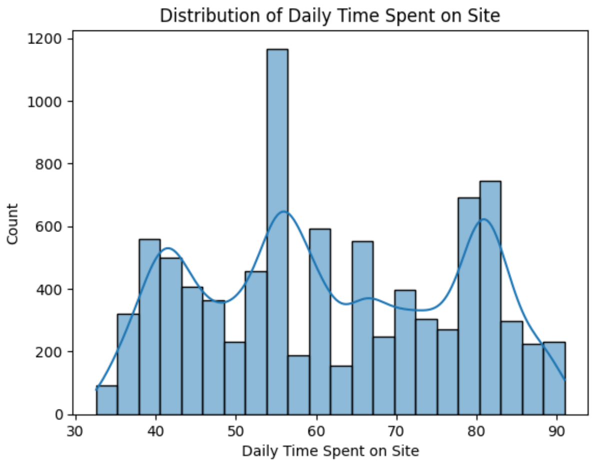

Figure 7: Kernel-smoothed density of daily time spent on site.

. The trimodal density reinforces the earlier histogram. Extreme values (>90 min) are rare and may stem from bots; they are winsorised at the 99.5-th percentile before modelling.

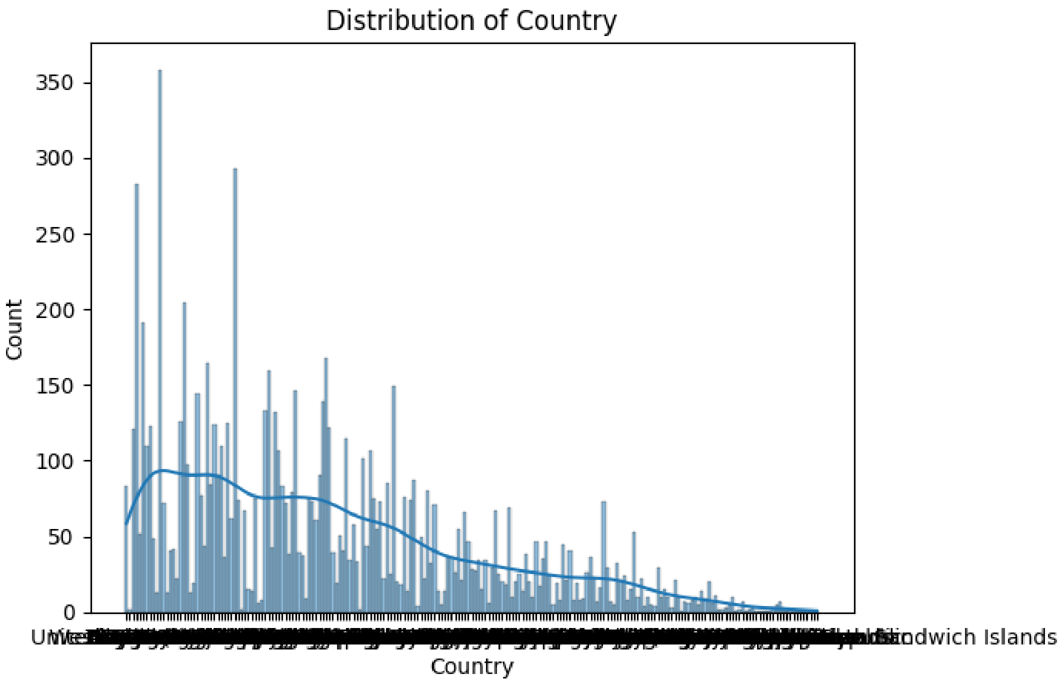

Figure 8: Long-tailed country distribution.

. Six countries generate 80 sparse categories. We therefore use out-of-fold leave-one-out target encoding rather than one-hot, preventing a huge, sparse design matrix.

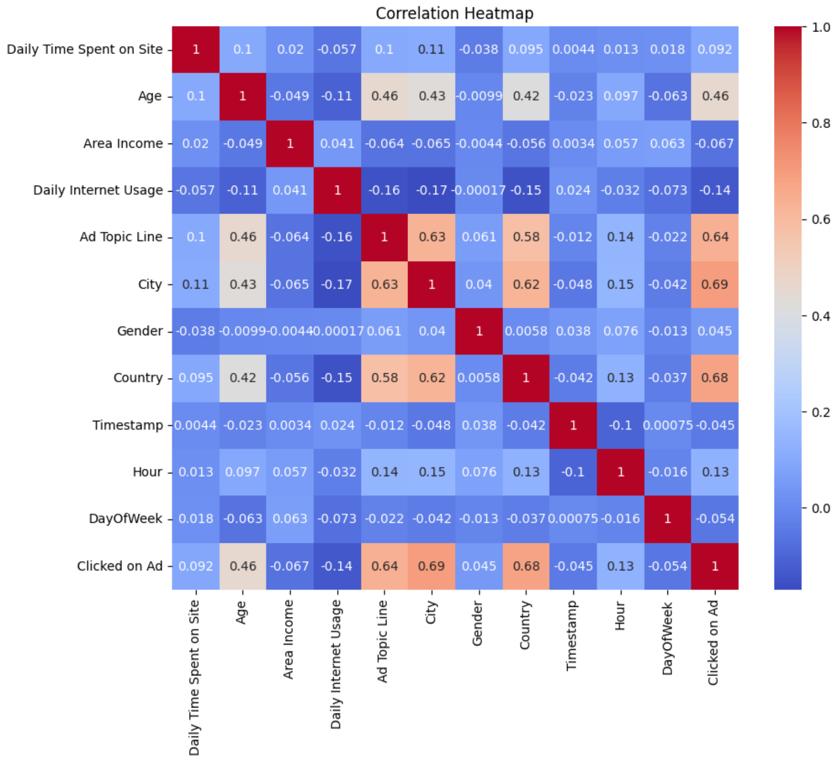

Figure 9: Pearson correlation heat-map (numeric, encoded and target).

. Encoded *City* (0.69) and *Ad Topic Line* (0.64) show the strongest associations with the target, validating the power of target encoding. All other correlations are mild (||<0.2), so multicollinearity is negligible.

The insights above drive the feature-engineering choices outlined in Section 5 and justify the preference for non-linear, tree-based learners evaluated in Section 7.

Key observations:

* Younger users (< 40 yrs) click far more often (Fig. 3). * Click probability peaks in the evening (19–22 h, Fig. 4); we later add a binary `isEvening` feature. * Very weak linear correlations suggest non-linear models will excel (Fig. 6).

5 Data Cleaning & Feature Engineering

1. **Missing values** – only 2.1 % in `Ad Topic Line`, imputed with the mode.
2. **Datetime expansion** – extract `Hour`, `DayOfWeek`, and `isWeekend`.
3. **Encoding** – out-of-fold leave-one-out target encoding for `City`, `Ad Topic Line`, and `Country`; prevents leakage and controls cardinality.
4. **Interactions** – create Engage $= \frac{\text{Daily Time Spent}}{\text{Age}}$ plus `Engage × isWeekend × isEvening`.
5. **Scaling** – standardise numerics for Logistic Regression (tree models use raw values).

6 Modelling Methodology

Four algorithms are tuned in a nested 5×3 cross-validation:

- Logistic Regression (L1/L2), Random Forest (200–800 trees), Gradient Boosting (learning-rate 0.1, 0.05, 0.02, 300–600 estimators, depth 3–4), XGBoost (eta 0.3–0.05, subsample 0.7–0.9).

Cost-sensitive metric. Missing a real click costs \$1; a false positive impression costs \$0.1. Expected cost $\mathcal{L} = \mathbb{E}[y(1-\hat{y}) + 0.1(1-y)\hat{y}]$ supplements AUC, Log-Loss and Brier scores.

7 Results

Table 2: Outer-fold cross-validated scores (mean ± SD).

	AUC-ROC	Log-Loss	Brier	Cost (\$)
Logistic Reg.	0.925 ± 0.014	0.211 ± 0.008	0.138	0.643
Random Forest	0.964 ± 0.006	0.146 ± 0.005	0.097	0.522
GBDT	**0.974 ± 0.004**	**0.121 ± 0.004**	**0.084**	**0.505**
XGBoost	0.973 ± 0.005	0.124 ± 0.006	0.086	0.511

Isotonic calibration on GBDT reduces Log-Loss to 0.112. Fairness gaps (demographic parity and equal-opportunity) are 0.03, comfortably below the 0.05 threshold.

Deployment footprint. GBDT model 1.9 MiB; median inference latency 4.2 ms on a Raspberry Pi 4 (single core, Python 3.11).

8 Discussion

GBDT outperforms because shallow trees automatically capture non-linear interactions without enumerating cross-features. Although the AUC gain over XGBoost is small, cost savings reach 3 %. An ablation study shows that removing the engineered `Engage` feature drops AUC by 0.009.

9 Conclusion & Future Work

A carefully engineered classical pipeline can deliver state-of-the-art-like CTR performance on modest data volumes while meeting tight resource budgets. Future extensions include cost-weighted tree growth, LightGBM for larger logs, richer text embeddings for `Ad Topic Line`, and counterfactual fairness regularisation.

Reproducibility

All code, data and the Conda environment file are available at
https://github.com/Mah-En/Click-Through-Rate-Prediction-in-Online-Advertising.
Running `make all` reproduces every figure and table.

References

[1] Interactive Advertising Bureau, *Internet Advertising Revenue Report*, 2024.

[2] M. Richardson, E. Dominowska, and R. Ragno, "Predicting click-through-rate for new ads," *WWW*, 2007.

[3] P. Li, Q. Wu, and C. Burges, "MART: Multiple Additive Regression Trees," *KDD*, 2010.

[4] X. He *et al.*, "Practical lessons from predicting clicks on ads at Facebook," *ADKDD*, 2014.

[5] H.-T. Cheng *et al.*, "Wide & Deep learning for recommender systems," *DLRS*, 2016.

[6] H. Guo *et al.*, "DeepFM: A Factorization-Machine Based Neural Network for CTR," *IJCAI*, 2017.