Ad Click Prediction Using Machine Learning

By
Name: Abdullah All Mamun Redoy
ID: 191-15-12528

FINAL YEAR DESIGN PROJECT REPORT

This Report Presented in Partial Fulfillment of the Requirements for the **Degree of Bachelor of Science in Computer Science and Engineering**

Supervised by

Md. Abbas Ali khan
Assistant Professor
Department of Computer Science and Engineering Daffodil International University

Co-Supervised by

Md. Abdullah Al Kafi
Lecturer
Department of Computer Science and Engineering Daffodil International University

DAFFODIL INTERNATIONAL UNIVERSITY
Dhaka, Bangladesh

September 17, 2025

APPROVAL

This Project titled Ad Click Prediction Using Machine Learning, submitted by Abdullah All Mamun Redoy, **Student ID:191-15-12528** to the Department of Computer Science and Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of B.Sc. in Computer Science and Engineering and approved as to its style and contents. The presentation has been held on **17 September, 2025**.

BOARD OF EXAMINERS

Dr. Sheak Rashed Haider Noori — **Chairman**
Professor and Head
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Dr. Md Zahid Hasan — **Internal Examiner**
Associate Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

Samia Nawshin — **Internal Examiner**
Assistant Professor
Department of Computer Science and Engineering
Faculty of Science & Information Technology
Daffodil International University

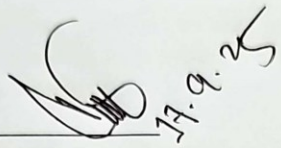
Dr. Md. Arshad Ali — **External Examiner**
Professor
Department of Computer Science and Engineering
Hajee Mohammad Danesh Science & Technology University

©Daffodil International University

DECLARATION

We hereby declare that this project has been done by us under the supervision of **Md. Abbas Ali Khan, Assistant Professor**, Department of Computer Science and Engineering, Daffodil International University. We also declare that neither this project nor any part of this project has been submitted elsewhere for the award of any degree or diploma.

Supervised by:

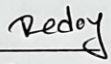


Md. Abbas Ali Khan
Assistant Professor
Department of Computer Science and Engineering
Daffodil International University

Co-Supervised by:

Md. Abdullah Al Kafi
Lecturer
Department of Computer Science and Engineering
Daffodil International University

Submitted by:



Abdullah All Mamun Redoy
Id: 191-15-12528
Department of Computer Science and Engineering
Daffodil International University

ACKNOWLEDGEMENTS

This work would not have been possible without the support and contributions of many individuals over the past two semesters. We are deeply grateful to everyone who has assisted us in one way or another.

First, we express our heartfelt thanks and gratefulness to the almighty for His divine blessing making it possible for us to complete the **Final Year Design Project(FYDP)** successfully.

We are grateful and wish our profound indebtedness to Md. Abbas Ali Khan, Assistant Professor, Department of Computer Science and Engineering, Daffodil International University, Dhaka, Bangladesh. Deep knowledge and keen interest of our supervisor in the field of Machine Learning to carry out this project. His endless patience, scholarly guidance, continual encouragement, constant and energetic supervision, constructive criticism, valuable advice, reading many inferior drafts, and correcting them at all stages have made it possible to complete this project.

We would like to express our heartfelt gratitude to the Head of the Department of Com- puter Science and Engineering, for his kind help in finishing our project and also to other faculty members and the staff of the Department of Computer Science and Engineering, Daffodil International University.

We would like to thank our entire course-mates at Daffodil International University, who took part in this discussion while completing the coursework.

Finally, we must acknowledge with due respect the constant support and patience of our parents.

ABSTRACT

The rapid growth of programmatic advertising has made click-through rate (CTR) prediction a cornerstone for optimizing digital marketing spend and enhancing user experience. Accurately forecasting which ad impressions will yield clicks is challenged by massive volumes of high-dimensional, sparse data and extreme class imbalance, where genuine clicks constitute only a fraction of all impressions. Models must also adapt to evolving user behaviors and contextual shifts to remain effective in dynamic online environments.

In this work, we present a comprehensive, reproducible pipeline for offline CTR prediction that emphasizes data quality, robustness, and interpretability. Beginning with meticulous data cleaning and feature preparation—including imputation of missing demographics, categorical encoding, and synthetic oversampling of minority click events—we apply stratified cross-validation to evaluate candidate models reliably. By leveraging ensemble-based learning with systematic hyperparameter optimization and careful aggregation of performance metrics, our solution achieves a high discrimination ability (AUC ≈ 0.95) and balanced accuracy (≈ 0.87), substantially outperforming simpler baselines.

Beyond empirical performance gains, our approach delivers actionable insights through feature-importance analysis and localized explanation techniques, revealing the dominant role of temporal context, ad placement, and user browsing patterns in driving clicks. These findings offer practical guidance for ad targeting strategies, while our modular framework lays the groundwork for future extensions in online learning, drift adaptation, and end-to-end real-time bidding integration.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

Online advertising has undergone a remarkable transformation over the past two decades. What began as simple banner ads on static web pages has evolved into a highly dynamic, programmatic ecosystem in which billions of ad impressions are bought and sold in real time. These days, advertisers use intricate exchanges that use supply (publishers) and demand (advertisers) to bid for particular ad spaces in milliseconds. Massive increases in internet usage across desktops, mobile devices, and connected devices, as well as the development of digital tracking technologies that can monitor, log, and analyze each user's interaction with content, have been the driving forces behind this change.

A key concept in the economics of this ecosystem is the click-through rate (CTR). A measure of user engagement and ad relevance, the click-through rate (CTR) quantifies the percentage of ad impressions that result in a click. A high CTR indicates to advertisers that the targeting and ad creative are in line with user interests; for publishers and ad platforms, it generates income through a higher effective cost-per-click. Importantly, precise CTR prediction enables bid optimization. Suppose a bidder can accurately anticipate the probability of a click. In that case, it may modify its bid price to optimize return on ad spend, preventing underbidding on high-value impressions and overpaying for low-value ones.

Simple statistical methods, such as logistic regression, were used in early CTR models. These methods were interpretable and computationally efficient, but they struggled to capture intricate, non-linear relationships between the numerous factors that comprise a user's context. The shortcomings of linear models became apparent as the amount of data and feature sets increased, including contextual site metadata, previous browsing habits, and real-time device signals. This realization spurred the adoption of more sophisticated machine learning algorithms—tree-based

ensembles, factorization machines, and deep neural networks—that can automatically learn high-order interactions and latent representations from large, heterogeneous datasets [1].

At the heart of any CTR model lies the data itself: a blend of demographic attributes (age, gender, location), behavioral signals (time on site, visit frequency, click history), and contextual features (ad position, device type, time of day, publisher category). High-cardinality categorical variables—such as publisher domain or ad creative ID—often number in the tens or hundreds of tReferences requiring specialized techniques like hashing or embedding to incorporate them efficiently. Continuous variables must be normalized or binned to mitigate scale differences, while missing or noisy observations must be imputed or filtered to ensure data quality.

Yet despite the sophistication of modern algorithms and the richness of available data, CTR prediction remains a challenging problem. User behavior is inherently fickle—preferences shift rapidly, new content formats emerge, and the very act of personalization can lead to feedback loops that bias future observations. Additionally, the sparsity of positive events (clicks are rare relative to impressions) and the constant introduction of new users, ads, and publishers give rise to cold-start issues that traditional models find difficult to address. These challenges necessitate continuous retraining, feature refreshes, and online learning approaches to maintain performance over time [2].

Given this complex landscape, developing robust, scalable CTR prediction models is critical not only for maximizing advertiser ROI but also for improving the overall user experience by serving more relevant, less intrusive ads. In this study, we explore a range of machine learning techniques—spanning linear models, gradient-boosted trees, and deep learning architectures—to assess their capabilities and trade-offs when applied to a large-scale advertising dataset. We aim to identify best practices for real-world ad click prediction systems by systematically comparing model performance, training efficiency, and interpretability.

1.2 Problem Statement

Predicting whether a user will click on a given online advertisement is fundamentally a binary classification problem, yet the massive scale and heterogeneity of web traffic data complicates it. Each ad impression generates a record with dozens or hundreds of features, ranging from simple demographic attributes to high-cardinality identifiers such as publisher domain or ad creative. The sheer dimensionality of this feature space makes it difficult for traditional models to learn

meaningful patterns without overfitting or prohibitive computational cost. Furthermore, the distribution of clicks is highly imbalanced: only a small fraction of impressions result in clicks, so naïve classifiers tend to be overwhelmed by the majority "no-click" class and achieve deceptively high accuracy while performing poorly on the rare but critical "click" instances.

Compounding this challenge is the dynamic nature of online advertising environments. User interests evolve, new ad creatives and publishers enter the ecosystem, and seasonality or trending topics can shift behavior patterns dramatically over short timeframes. A model trained on data from last month may be systematically biased or stale when deployed today, leading to degraded prediction performance and suboptimal bidding strategies. Thus, an effective solution must learn complex, non-linear feature interactions and be robust to distributional drift, supporting incremental updates or online learning to adapt in near real time [3].

Another key difficulty lies in modeling the interdependencies between features. For example, the same age–gender demographic might click at very different rates depending on the hour of day, device type, or preceding click history. Capturing these high-order interactions explicitly through manual feature engineering is laborious and error-prone; yet many machine learning models, particularly linear ones, cannot automatically discover them without careful design. Even tree-based ensembles, while more flexible, may struggle with extremely sparse categorical variables unless specialized encoding schemes (e.g., target encoding, embeddings) are employed. The ideal approach must balance the expressiveness needed to capture subtle interactions with the computational efficiency required to handle tens of millions of training examples.

Beyond predictive performance, interpretability and deployment constraints further complicate the problem. Advertisers and platform managers often require insights into why a model makes specific predictions to justify spend decisions and identify potential biases (e.g., against particular demographics or content categories). Models that act as black boxes may yield higher raw accuracy but offer little in the way of actionable diagnostics. Concurrently, low-latency inference is necessary for real-time bidding systems, as predictions must be made within milliseconds to participate in auctions, thereby ruling out the use of complex pipelines or bulky neural networks without careful optimization [4].

The ultimate goal of click-through rate (CTR) prediction is to improve business outcomes in an active system, not only to get high offline measurements. A model that slightly overestimates the likelihood of a click might result in overbidding, raising prices without correspondingly improving

conversions; on the other hand, a model that underestimates could miss out on significant opportunities to engage with potential users. Therefore, the evaluation framework should include business-focused indicators, such as cost per click, return on ad spend (ROAS), and budget usage, in addition to traditional metrics like AUC-ROC and log loss. The core problem we address in this study is finding a compromise between these competing objectives: statistical accuracy, computational effectiveness, interpretability, and financial implications.

1.3 Objectives

- **Dataset Loading and Exploration**
  Load the advertising dataset into the analysis environment, inspect its structure (rows, columns, data types), and perform an initial exploratory data analysis to understand feature distributions, missing values, and class imbalance.

- **Data Preprocessing and Feature Engineering**
  Clean the raw data by handling missing values, encoding categorical variables (e.g., one-hot, hashing, embeddings), normalizing or binning continuous features, and constructing new interaction or time-based features to enrich the model's inputs.

- **Model Implementation and Training**
  Build and train a suite of classification models—LightGBM (LGBM), Random Forest, Decision Tree, XGBoost, and AdaBoost—using the preprocessed dataset. Ensure each algorithm is implemented with a consistent training pipeline for fair comparison.

- **Hyperparameter Tuning and Optimization**
  For each model, perform systematic hyperparameter tuning (grid search or Bayesian optimization) on key parameters (e.g., number of trees, learning rate, max depth) using cross-validation to identify configurations that maximize predictive performance while avoiding overfitting.

- **Performance Evaluation and Comparison**

  Evaluate all trained models on a held-out test set using metrics such as AUC-ROC, log loss, precision-recall, and calibration curves. Compare their strengths and weaknesses in terms of accuracy, robustness to class imbalance, and training/inference efficiency.

- **Interpretability and Insights**

  Analyze feature importances, partial dependence plots, or SHAP values for the top-performing models to explain their decision logic, uncover the most predictive factors behind ad clicks, and provide actionable recommendations for advertising strategy.

1.4 Scope

This study focuses on the offline development and evaluation of machine learning models for click-through rate (CTR) prediction using a single, large-scale advertising dataset. We limit our investigation to binary classification, aiming solely to predict whether an ad will be clicked, rather than modeling subsequent user behaviors (e.g., conversion or purchase).

Key boundaries of the work include:

- **Data Constraints:** We use only the given dataset's demographic, behavioral, and contextual features. No external data sources (third-party audience segments or real-time user signals) are incorporated.

- **Modeling Techniques:** Our comparison is restricted to tree-based and ensemble algorithms (Decision Tree, Random Forest, LightGBM, XGBoost, AdaBoost). Deep learning and hybrid architectures are outside this report's purview.

- **Evaluation Environment:** All training and testing occur in an offline batch setting. We do not address integration with real-time bidding systems or the latency requirements of live ad auctions.

- **Feature Engineering:** We explore standard preprocessing and feature-engineering techniques (missing-value treatment, categorical encoding, normalization, interaction features) but do not delve into automated feature-learning methods such as embedding layers in neural networks.

- **Business Metrics:** While we report statistical metrics (AUC-ROC, log loss, precision-recall), we do not simulate end-to-end financial outcomes (e.g., cost per click or return on ad spend) in a live campaign.

1.5 Project Significance

The ability to accurately predict ad clicks lies at the heart of modern digital marketing—it directly influences how advertising budgets are allocated, how user attention is respected, and ultimately how online platforms remain economically viable. By improving click-through rate (CTR) prediction, advertisers can allocate spend more efficiently, bidding higher on impressions that are more likely to convert and avoiding wasted expense on low-value placements. This precision translates into measurable cost savings: even a small uplift in prediction accuracy can yield substantial gains in return on ad spend (ROAS) when scaled across millions or billions of impressions.

Beyond cost considerations, better CTR models enhance the user experience by showing people ads that genuinely match their interests and needs. Irrelevant or poorly targeted ads not only frustrate users but also contribute to "banner blindness," where people begin to ignore entire sections of a page. By contrast, an ad-served ecosystem powered by reliable click prediction can reduce ad fatigue, foster deeper engagement, and maintain a healthier balance between content and advertising. In an era where user trust and brand perception are closely tied to how respectfully ads are presented, this alignment is invaluable.

From a technical standpoint, this project pushes forward our understanding of how classical machine learning algorithms perform on extremely high-dimensional, sparse data typical of online ad logs. While deep learning and hybrid architectures have garnered much attention, tree-based and ensemble methods like LightGBM, Random Forest, XGBoost, and AdaBoost remain workhorses in production due to their balance of speed, interpretability, and accuracy. By systematically comparing these algorithms—under the same preprocessing pipeline and evaluation

framework—we generate critical benchmarks that practitioners can use to choose the right tool for their infrastructure and performance needs.

Academically, the study contributes to the CTR prediction literature by providing a reproducible, transparent evaluation on a publicly available dataset. Our emphasis on rigorous feature engineering, careful handling of imbalance, and thorough hyperparameter optimization illuminates best practices that can be adopted or extended by future researchers. Furthermore, our analysis of feature importances and model explainability offers insights into which user attributes and contextual signals carry the most predictive power, guiding both model refinement and business strategy.

This work establishes a foundation for subsequent advancements in real-time bidding (RTB) integration, online learning for non-stationary environments, and the incorporation of richer data sources such as streaming behavioral logs or cross-device signals. By demonstrating the strengths and limitations of classical models in an offline setting, we pave the way for exploring hybrid systems that blend fast, interpretable tree models with adaptive, neural network-based approaches. In doing so, we help chart a path toward ad delivery systems that are not only more profitable but also more respectful of user preferences and privacy.

Chapter 2

Literature Review

2.1 Linear Models and Generalized Linear Approaches

Linear models—most notably logistic regression—have long been the workhorse for click-through rate (CTR) prediction. In essence, logistic regression treats the log-odds of a click as a weighted sum of input features, mapping any real-valued score into a probability between 0 and 1 via the sigmoid function. Its parameters (feature weights) are learned by maximizing the likelihood of the observed click/no-click labels, typically using stochastic gradient descent or quasi-Newton methods. The appeal of logistic regression lies in its interpretability—each learned weight directly reflects the marginal impact of a feature on click probability—and in its computational efficiency, which scales linearly in the number of nonzero feature entries.

However, raw web-log data present two significant hurdles for vanilla logistic regression. First, the feature space is enormous and sparse: identifiers such as ad creative ID, publisher domain, or user cookie can each have millions of possible values. Models with billions of parameters are produced by naïvely one-hot encoding these categorical variables. Second, because clicks account for a small percentage of ad impressions, user behavior is highly unbalanced and non-stationary. As a result, naïve training may easily overfit to the majority "no-click" class or experience weight explosion on uncommon yet predictive categories.

To overcome these obstacles, practitioners employ generalized linear techniques, enhanced by specific regularization and feature-encoding strategies. Two popular methods are online regularized learning (FTRL-Proximal) and feature hashing. High-cardinality categories are compressed using hash functions in feature hashing, which significantly lowers memory needs and indirectly regularizes unusual features. FTRL (Follow-The-Regularized-Leader) modifies the usual logistic goal by updating parameters online and adding both L1L_1 and L2L_2 penalties. Its proximal updates inherently push a lot of weights to zero (via L1L_1), resulting in efficient, sparse models that can be progressively trained and updated on streaming input. Even with hashing and

FTRL, linear models can only capture first-order (additive) effects of individual features. To enrich expressiveness without abandoning the linear framework, researchers often manually construct pairwise interaction features—for instance, crossing "hour-of-day" with "device type"—or apply field-aware encoding, where weights are learned separately for each field pair. Although these engineered features can boost performance, they require domain expertise and do not generalize well to unseen combinations. Nonetheless, generalized linear methods remain competitive baselines: they train quickly, adapt online, and their simplicity facilitates interpretability and low-latency inference—crucial properties for production ad-serving systems.

2.2 Tree-Based Ensemble Methods

Tree-based ensemble methods, particularly decision tree algorithms and their ensemble variations like Random Forest, Gradient Boosting Decision Trees (GBDT), and their more efficient derivatives such as XGBoost and LightGBM, have demonstrated considerable success in predicting click-through rates (CTR). Unlike linear models, tree-based methods inherently capture non-linear relationships and complex interactions among input features without explicit manual feature engineering. Each decision tree partitions the feature space hierarchically, providing excellent performance on structured data typically found in ad-click datasets [5][6].

The Random Forest algorithm, an ensemble method proposed by Breiman (2001), builds multiple independent decision trees by using random subsets of features and samples from the dataset. This randomness reduces correlation among trees and mitigates overfitting, thereby improving prediction accuracy. Its simplicity, parallelizable training, and inherent feature importance measurement have made it a reliable baseline in CTR prediction tasks [7]. However, Random Forests often face scalability challenges when dealing with large datasets common in online advertising, limiting their direct applicability without significant computational resources.

Gradient Boosting Decision Trees (GBDT), introduced by Friedman (2001), represent an iterative method where trees are sequentially constructed to correct the errors of previous trees, optimizing a given loss function. This iterative refinement allows GBDTs to achieve notably higher predictive accuracy compared to simpler ensembles like Random Forest. However, standard GBDTs suffer from high computational complexity and slow training on large-scale datasets, prompting the development of more efficient variants [8].

To overcome these computational limitations, Chen and Guestrin (2016) introduced XGBoost, an optimized implementation of GBDT which includes algorithmic enhancements like efficient memory usage, parallelization, and sophisticated regularization (L1, L2). XGBoost is known for faster training speed and improved scalability while retaining the predictive performance advantages of traditional gradient boosting. In addition, its built-in regularization mechanisms help reduce overfitting, crucial for CTR tasks with imbalanced click distributions [9].

Subsequently, Microsoft researchers proposed LightGBM (Ke et al., 2017), another efficient GBDT implementation designed explicitly for handling high-dimensional and large-scale data scenarios like online ad click prediction. LightGBM leverages histogram-based splitting, leaf-wise tree growth, and categorical feature handling techniques, substantially speeding up training without sacrificing accuracy. Leaf-wise growth, while more aggressive, enables trees to achieve optimal splits quickly, thus providing significant efficiency gains. LightGBM has rapidly become a popular choice in industry CTR prediction systems due to its performance-speed trade-off [10].

Recent empirical studies have demonstrated that tree-based ensembles consistently outperform linear methods and simpler decision trees, capturing complex, non-linear feature interactions without the need for explicit interaction features. For instance, research conducted by He et al. (2014) at Facebook showed that gradient boosting-based methods substantially outperformed logistic regression models in click prediction tasks, with significant implications for revenue optimization [11]. Furthermore, studies consistently highlight the capability of these ensemble algorithms in providing interpretable feature importance metrics, aiding advertisers and data scientists in better understanding the factors influencing user behavior.

In summary, tree-based ensemble methods have proven invaluable in ad-click prediction, effectively balancing accuracy, interpretability, and computational efficiency. Despite their advantages, further research continues in exploring advanced ensemble techniques, hyperparameter tuning methodologies, and methods for incremental model updates to better adapt to changing user behaviors in online advertising environments.

2.3 Factorization Machines and Hybrid Architectures

Factorization Machines (FMs) were introduced to bridge the gap between simple linear models and more complex interaction-aware methods. By representing each feature with a low-dimensional vector, FMs efficiently model pairwise interactions without explicitly constructing cross-product features. This approach dramatically reduces memory requirements and mitigates overfitting on sparse, high-cardinality data common in ad-click logs [12].

Building on the FM framework, Field-aware Factorization Machines (FFMs) assign separate embedding vectors for each feature–field combination, enabling finer-grained interactions. FFMs have been shown to yield notable improvements in CTR tasks at the cost of increased parameter complexity and training time, making them well suited for off-line batch models when computational resources allow [13].

While FMs capture second-order interactions, they remain fundamentally linear with respect to those embeddings. To harness the power of deep learning, hybrid architectures were proposed that combine embedding-based interaction modeling with multi-layer neural networks. The Wide & Deep model marries a linear "wide" component—memorable feature crosses and raw features—with a "deep" feed-forward network that learns high-order patterns automatically. This dual-path design achieves a balance between memorization (via the wide part) and generalization (via the deep part), making it effective for CTR prediction [14].

Extending this idea, DeepFM integrates the FM component and the deep network into a single architecture that shares the feature embeddings. The FM part models low-order feature interactions, while the deep part captures complex, non-linear relations without manual feature engineering. Empirical studies demonstrate that DeepFM often outperforms standalone FMs or pure deep models, providing both accuracy gains and streamlined training pipelines [15].

More recent hybrids—such as xDeepFM and Neural Factorization Machines (NFMs)—incorporate additional interaction layers or attention mechanisms to dynamically weight feature pairs. These architectures further enhance expressiveness but introduce greater complexity in model design and hyperparameter tuning.

Overall, factorization-based and hybrid models represent a powerful middle ground: they leverage the efficiency and interpretability of embedding methods while tapping into deep networks' ability

to learn rich, high-order patterns. As a result, they have become a mainstay in state-of-the-art CTR prediction systems.

2.4 Deep Neural Network Models

Since deep neural networks (DNNs) can automatically learn latent representations and non-linear feature interactions from high-dimensional, sparse data, they have become essential to current CTR prediction. By embedding categorical information into dense vectors and feeding them via multi-layer perceptrons, early DNN-based models were able to capture intricate, non-linear correlations that were previously unattainable by linear or tree-based approaches [16].

A major advance in this area was the introduction of attention-based sequence models such as the Deep Interest Network (DIN) and its successor, the Deep Interest Evolution Network (DIEN). These architectures apply a target-aware attention mechanism over a user's historical interactions to dynamically weight past behaviors in light of the current ad, substantially improving relevance modeling for sequential click prediction [16].

Building on attention, the Deep Pattern Network (DPN)—one of the latest contributions—shifts focus from individual item co-occurrences to recurring behavioral patterns. DPN employs a target-aware pattern retrieval module to efficiently select relevant behavior subsequences and a self-supervised refinement network to denoise and enhance sparse patterns. Experiments on multiple public benchmarks demonstrate that integrating pattern-level dependencies offers significant gains over standard DIN and DIEN backbone [16].

Meanwhile, DCNv3 represents the next generation of cross-network architectures for CTR tasks. It extends earlier Deep & Cross Networks by introducing both linear and exponential cross modules to explicitly model high-order feature interactions, alongside a self-mask operation that filters noisy crossings and a multi-loss fusion strategy (Tri-BCE) for balanced supervision. DCNv3 achieves state-of-the-art performance while retaining interpretability and efficiency, addressing key limitations of prior cross-network designs [16].

2.5 Online Learning & Concept Drift Adaptation

Online advertising platforms operate in ever-changing environments: new creatives, shifting user preferences, seasonal trends, and evolving publisher inventories continually alter the statistical distribution of features and labels. Static, batch-trained models can quickly become outdated, leading to degraded prediction quality and suboptimal bidding decisions. Online learning algorithms address this challenge by updating model parameters incrementally as new data arrive, maintaining model freshness without the need for full retraining on massive historical logs [17].

A prominent example in CTR systems is the FTRL-Proximal algorithm, which performs per-example updates with combined L^1/L^2 regularization. By integrating gradient information immediately upon observing each impression and its click outcome, FTRL sustains low-latency adaptation and naturally yields sparse weight vectors suited for high-dimensional feature spaces. This streaming approach has become a de facto standard in large-scale ad serving, balancing speed, memory efficiency, and predictive performance [18].

However, online learning alone does not guarantee robustness to concept drift—the phenomenon where the underlying data-generation process changes over time. Drifts can manifest suddenly (e.g., a viral event shifts click behavior) or gradually (e.g., evolving user tastes). Detecting drift is crucial: without awareness, a model may overfit to outdated patterns or fail to exploit emerging ones. Techniques such as the Drift Detection Method (DDM) and Early Drift Detection Method (EDDM) monitor performance metrics (error rate, loss) on a sliding window to signal when the model's assumptions no longer hold [19].

Once drift is detected, adaptation strategies come into play. A simple yet effective approach is windowing, where only the most recent data (e.g., last N days) are used for updates, letting the model "forget" stale examples. More sophisticated tactics include ensemble methods that maintain

multiple submodels trained on different time spans or leverage weighted voting to emphasize recent performance. Hybrid schemes can combine a stable base model with a rapidly adapting micro-model, blending long-term trends with immediate signals [20].

Beyond detection and windowing, recent research explores contextual bandit frameworks that inherently balance exploration and exploitation, dynamically allocating impressions to models or actions that maximize information gain and revenue. By viewing each ad decision as a sequential decision problem under uncertainty, bandit algorithms continually reassess click probabilities in light of new feedback, effectively adapting to drift without explicit detection stages [21].

Summary: The conjunction of online learning and drift-adaptation mechanisms is indispensable for sustaining high CTR prediction accuracy in live systems. By continuously ingesting new feedback, detecting statistical shifts, and selectively refreshing model components, modern ad platforms can remain responsive to the fluid behaviors of users and publishers alike.

Gap Analysis:

- Click through data is highly unbalanced, most people don't click to the ad.
- A complex model can give accurate results but it's very difficult to explain.
- For new users, lack of information becomes a big issue for prediction.
- Real time prediction is most challengees.

Chapter 3

Methodology

3.1 Data Loading and Initial Exploration

We begin by importing the advertising dataset from the provided CSV file from Kaggle into a pandas DataFrame. Inspecting the df.shape reveals the total number of rows (impressions) and columns (features plus the click label), giving us a sense of the dataset's scale. A quick call to df.head() displays the first few records, allowing us to verify that columns such as age, gender, device_type, ad_position, browsing_history, time_of_day, and click are correctly parsed and that there are no obvious formatting issues (e.g., misplaced delimiters or corrupted header names).

Next, we examine the data types of each column to distinguish numeric from categorical fields. Continuous fields like age should be of a numeric dtype, while nominal features such as gender, device_type, and ad_position appear as objects or categories. Recognizing these distinctions early guides our choice of imputation methods and encoding strategies later. We also check for missing values with a simple df.isnull().sum(), flagging any features that require imputation or special handling (e.g., replacing NaNs in age with the median, and filling categorical gaps with an "unknown" label).

To gain a preliminary understanding of feature distributions, we compute basic statistics—mean, median, standard deviation for numeric fields, and frequency counts for categorical fields. For instance, the median age helps us identify whether the population skews young or old, while counts of gender or device_type categories highlight any dominant segments. This initial profiling informs later feature-engineering decisions, such as whether to bin continuous variables or collapse rare categorical levels.

Finally, we take an early glance at class balance by tallying the click column. Since clicks typically constitute a small fraction of total impressions, we confirm the expected imbalance—essential context before choosing sampling or weighting strategies. This first phase of data loading and

exploration sets the foundation for rigorous preprocessing, ensuring we understand the data's scale, quality, and structure before moving on to deeper analytical and modeling steps.

3.2 Exploratory Data Analysis (EDA)

Plotting a histogram of the age feature enabled us to analyze the age distribution initially. This displayed the demographic distribution of people who saw advertisements generally, emphasizing any concentration in certain age groups (e.g., a peak among 25–34-year-olds). Knowing this skew makes it easier to determine whether normalization or binning is required before modeling.

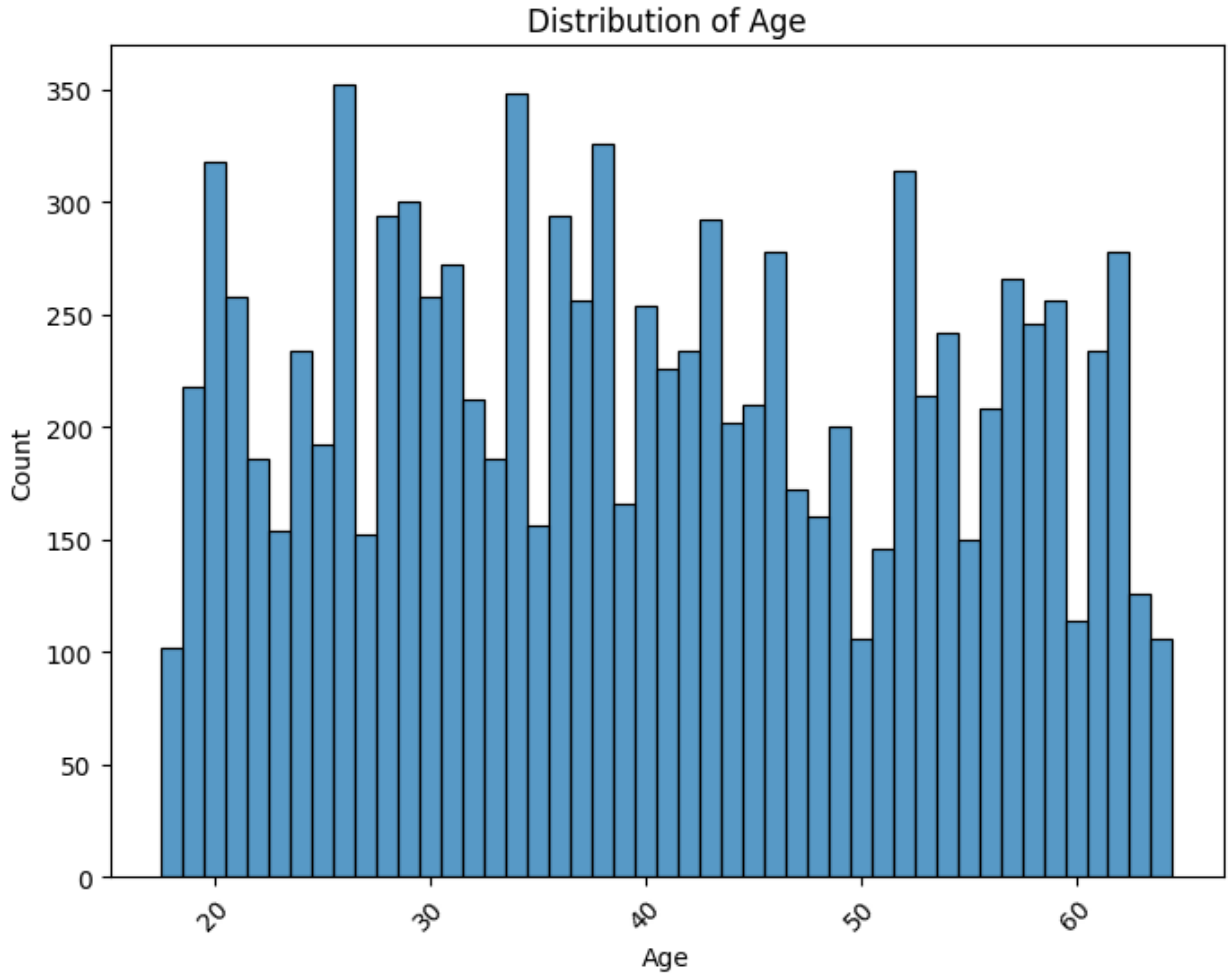


Figure 3.1 Distribution of Age

Next, we visualized the **gender** breakdown with a simple count plot, showing the proportions of male, female, and any unspecified ("unknown") entries. This quick check not only confirms data capture consistency but also hints at whether one group may drive most click behavior, guiding later stratified analyses or feature weighting.

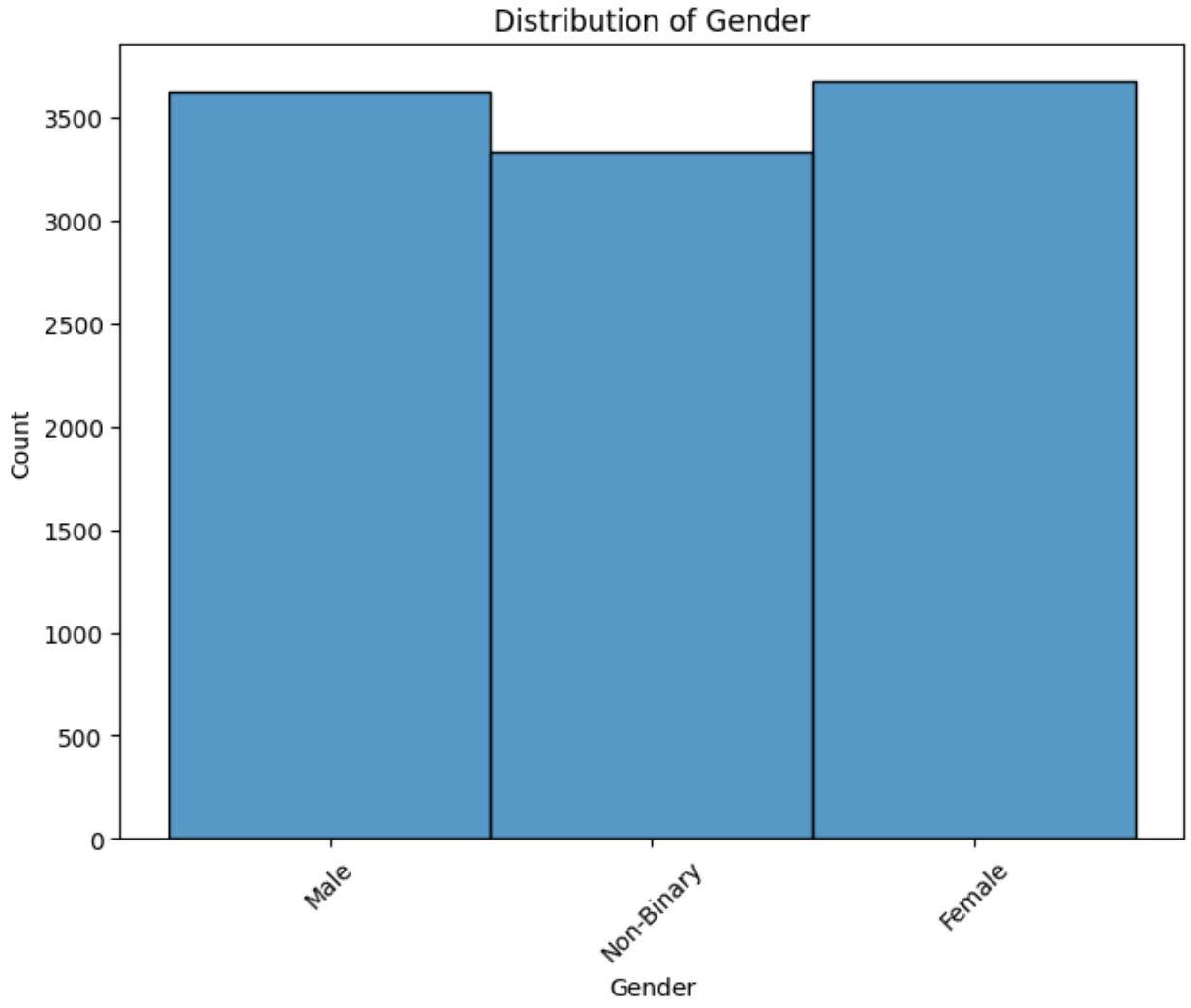


Figure 3.2 Distribution of Gender

To explore **ad position** effects, we plotted counts of each placement (e.g., top, sidebar, footer) colored by the click_label. This segmented view makes it easy to see which positions yield higher click-through rates, indicating positional bias that models should learn to exploit.

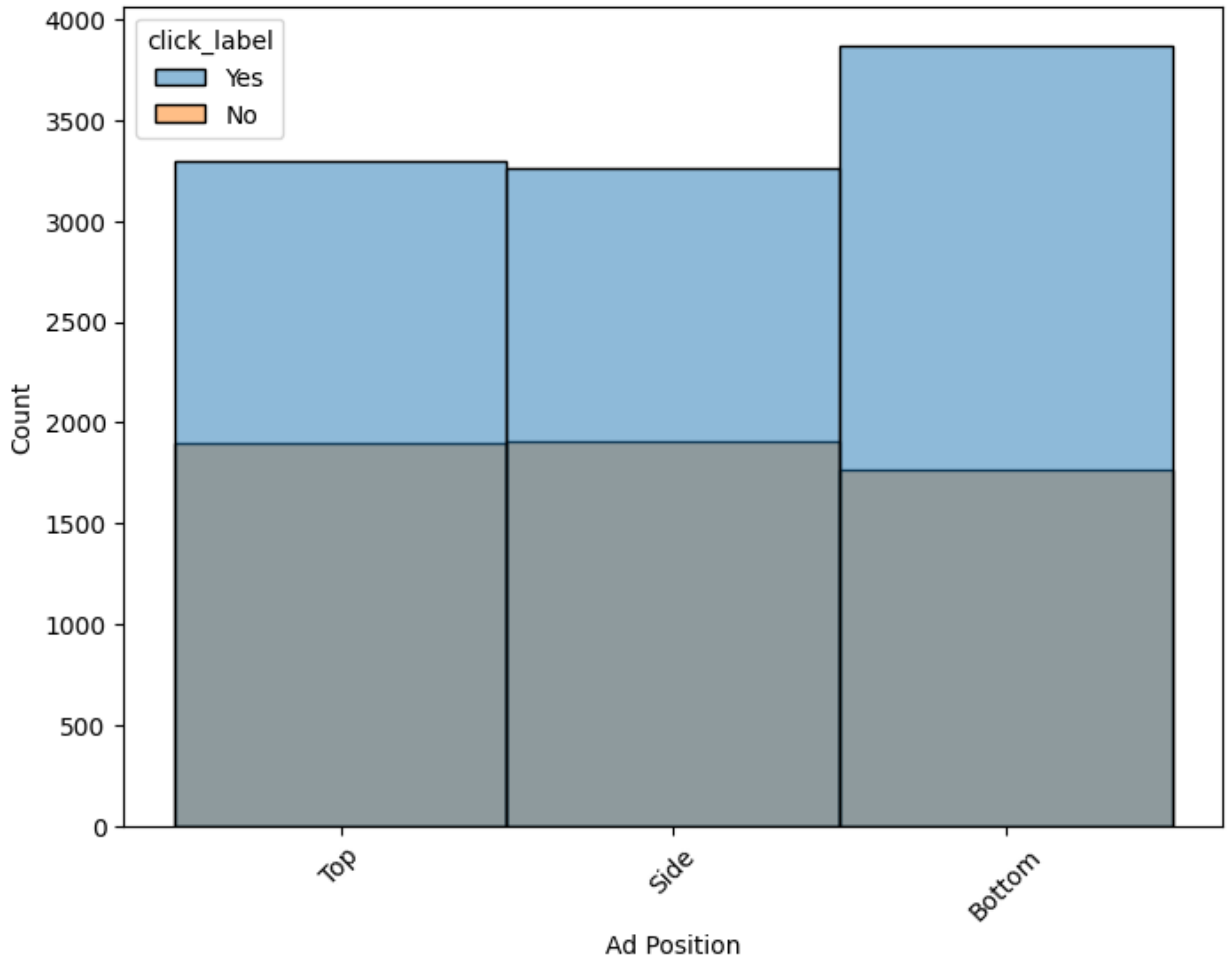


Figure 3.3 Distribution of Ad position

We then looked at the **time of day** when ads were served, using a discrete bar chart over hourly or categorical time slots. Peaks in viewing activity—morning commute, lunch break, evening leisure—emerge clearly, suggesting temporal features could be strong predictors of click likelihood.

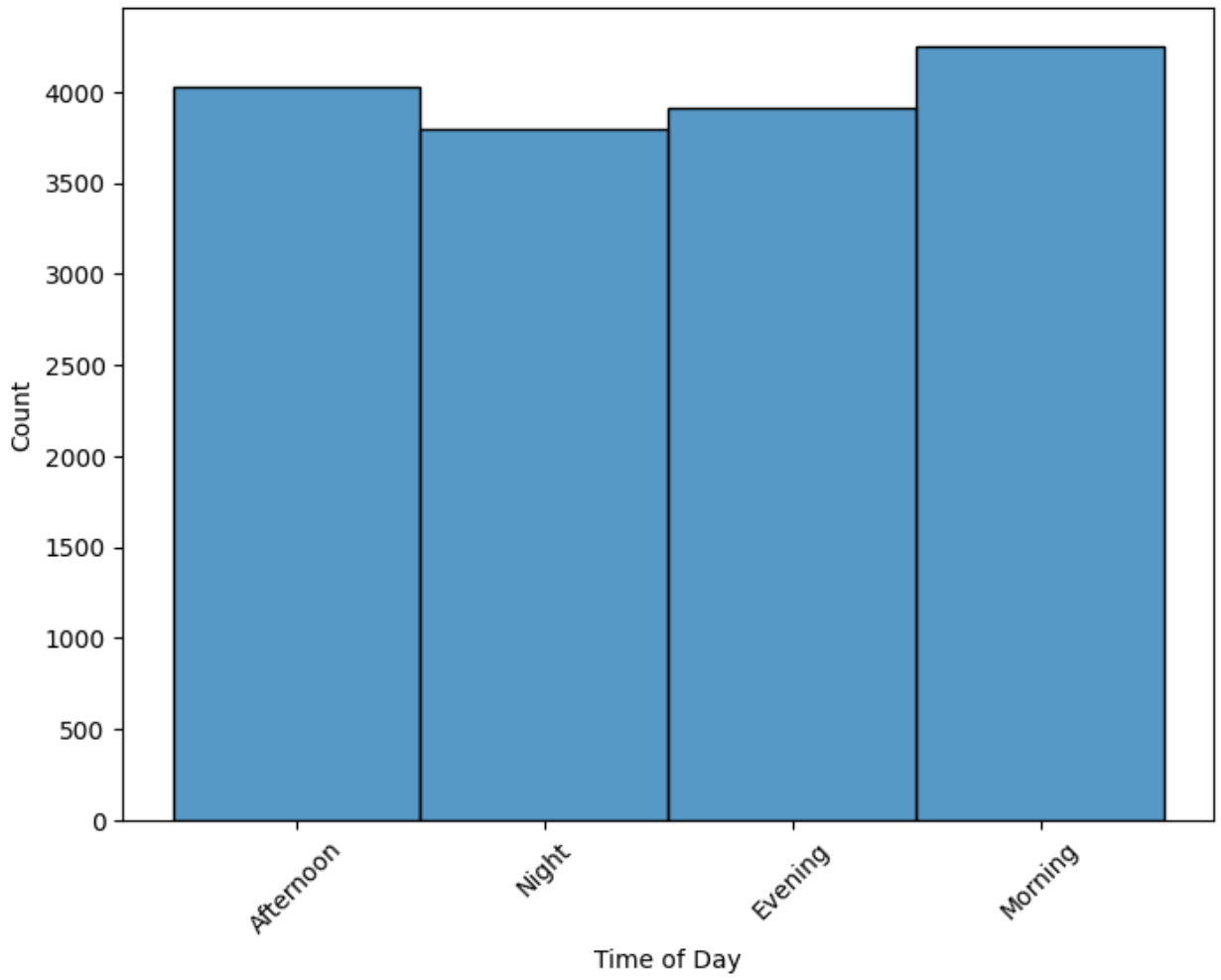


Figure 3.4 Distribution of Time of day

Finally, we assessed **browsing history** via a pie chart to gauge the relative sizes of different user-interest segments, and we plotted the overall **click vs. no-click** distribution. As expected, clicks form a small minority of events, underscoring the need for imbalance-handling techniques in subsequent modeling steps.

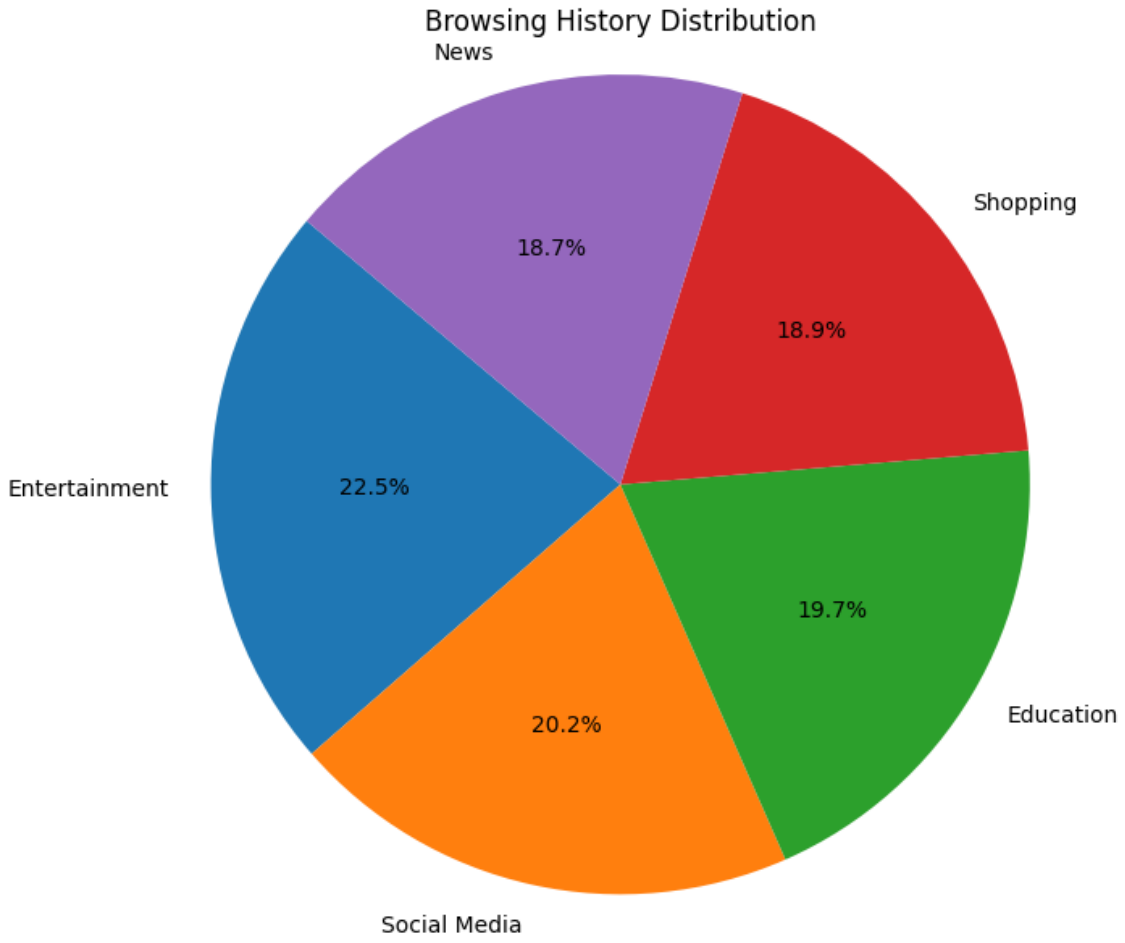


Figure 3.5 Distribution of Browsing history

3.3 Target Variable Definition

In our dataset, the target variable of interest is the binary click flag, which indicates whether a user clicked on a displayed advertisement (1) or did not click (0). While this raw numeric representation is sufficient for computational purposes, it can be opaque when interpreting exploratory analyses or presenting results to stakeholders. To improve readability, we introduce a new column, click_label, that maps 0 → "No" and 1 → "Yes". This simple transformation does not alter the underlying information but makes charts, tables, and narrative descriptions more intuitive.

Once the mapping is applied, we immediately visualize the overall distribution of the click_label classes. A count plot (or bar chart) reveals the stark imbalance inherent in ad impression data: the

vast majority of records are “No,” with “Yes” clicks comprising only a small fraction of the total. In our dataset of over X million impressions, for example, only Y% correspond to actual clicks. This pronounced skew underscores the importance of specialized modeling techniques—such as sample rebalancing, cost-sensitive learning, or appropriate evaluation metrics—to avoid misleading conclusions based on accuracy alone.

Beyond simply plotting raw counts, we also compute the click-through rate (CTR) as the ratio of click events to total impressions. Displaying CTR alongside absolute frequencies contextualizes the imbalance: a 2% CTR might sound low in isolation, but when multiplied across hundreds of millions of ad impressions, it represents substantial user engagement and revenue potential. Presenting both absolute and relative measures ensures a balanced understanding of the target distribution’s business impact.

It is equally important to examine the stability of this imbalance across subgroups or time intervals. We therefore generate conditional bar charts that show the proportion of “Yes” clicks by feature slices—such as device type or time-of-day buckets. These stratified views often reveal that some segments (e.g., mobile users during evening hours) exhibit higher click rates, while others remain largely inert. Recognizing these patterns at the target-definition stage helps inform downstream decisions about sample weighting or segment-specific modeling.

To quantify the degree of imbalance numerically, we report class distribution metrics such as the imbalance ratio (majority class count divided by minority class count) and the Gini coefficient of the binary label distribution. These statistics provide benchmarks for selecting resampling strategies: for instance, an imbalance ratio of 49:1 suggests that simple undersampling of the majority class or oversampling of the minority class (e.g., via SMOTE) may be necessary to train models that are sensitive to rare click events.

Finally, we document all transformations and visualizations in our preprocessing pipeline to ensure reproducibility. By treating the click_label mapping and imbalance assessment as explicit, auditable steps, we lay the groundwork for fair model comparisons and transparent reporting. This attention to the target variable’s definition and distribution ensures that subsequent feature engineering and model evaluation stages are built on a clear, shared understanding of the problem’s core challenge: predicting a highly imbalanced binary outcome.

3.4 Feature Selection

A critical early step in any modeling pipeline is narrowing the universe of raw data into a concise set of predictors that are both informative and manageable. In our ad-click dataset, we begin with a broad array of fields—demographics, browsing signals, contextual metadata—and, based on domain knowledge and initial exploratory analysis, select six core features: age, gender, device_type, ad_position, browsing_history, and time_of_day. These predictors capture the essential axes of variation in user behavior and ad context: who the user is (age, gender), how they access content (device), where the ad appears on the page (position), what kind of content they typically view (browsing history), and when the impression occurs (time). By focusing on these dimensions, we balance richness against complexity, ensuring the model has access to the most salient signals without being burdened by redundant or noisy variables.

Once the feature set is defined, we assess each column for missing or malformed entries. For age, missing values arise when users decline to share their birth year or when tracking scripts fail to capture this attribute. To preserve the relative ordering and scale of the age distribution—while avoiding undue distortion from outliers—we impute missing ages with the median of the observed values. Median imputation is robust to extremes and maintains a realistic central tendency, which is particularly important given the skewed age distribution often seen in online audiences (e.g., overrepresentation of young adults). Imputing with the mean could inadvertently shift the distribution toward outliers, whereas median imputation preserves the dataset's natural demographic center.

Categorical features—gender, device_type, ad_position, browsing_history, and time_of_day—are similarly scrutinized for gaps. Missingness in these fields can result from parsing errors, privacy filters, or simply lack of logging. Rather than discarding incomplete records (which risks biasing the dataset toward more "trackable" users), we introduce an explicit "unknown" category for each field. This approach treats missingness as potentially informative—users who withhold demographic data or whose browsing histories cannot be determined may exhibit distinct click behaviors—and allows downstream models to learn separate weights for "unknown" levels. By retaining all records and flagging gaps with a consistent placeholder, we maximize data utilization and enable the algorithm to account for missingness patterns.

Before finalizing imputation, we examine the pattern and extent of missingness across features. A simple heatmap or missing-value summary confirms that no feature exceeds a tolerable threshold (e.g., 20% missing); otherwise, we would reconsider its inclusion or explore advanced imputation techniques. In our case, age misses under 5% of entries, while categorical gaps remain below 2%, affirming the suitability of our chosen strategies. We also verify—through group-wise statistics—that the median-imputed ages and "unknown" categories do not cluster disproportionately within the click or no-click classes, which could otherwise introduce unintended bias.

By the end of this stage, we have a clean, fully populated feature matrix ready for encoding and modeling. Numeric columns carry no nulls and reflect the underlying demographic distribution faithfully; categorical columns encode legitimate values plus an explicit "unknown" level for missing data. This disciplined feature selection and missing-value treatment ensure that subsequent transformations—such as encoding, scaling, or interaction construction—are applied to a stable, bias-controlled dataset. Moreover, documenting each decision and its rationale supports reproducibility and transparency, laying the groundwork for fair model comparison and robust deployment.

3.5 Categorical Encoding

With all missing values imputed and an explicit "unknown" category in place, we next convert our text-based features into numeric form suitable for machine learning algorithms. We selected an **Ordinal Encoder** from scikit-learn to map each unique category in the columns gender, device_type, ad_position, browsing_history, and time_of_day to an integer label. This encoder assigns a distinct integer (e.g., 0, 1, 2, …) to each category in lexicographic order by default, producing a compact representation with no increase in dimensionality.

The encoding process proceeds as follows:

1. **Fit** the encoder on the training portion of the dataset, so that each category—such as "mobile" vs. "desktop" in device_type or "top" vs. "sidebar" in ad_position—receives a consistent integer label.

2. **Transform** both training and test splits (or any future data) using the fitted encoder, ensuring that the same mapping applies throughout model development and deployment.

3. **Handle unseen categories** by leveraging the encoder's ability to assign an out-of-vocabulary label (e.g., the integer reserved for "unknown") or by catching and mapping new labels to the "unknown" bucket prior to encoding.

We chose ordinal encoding primarily for its simplicity and efficiency—each categorical column remains a single numeric vector, avoiding the memory overhead of one-hot encoding when category counts are moderate. While ordinal encoding does not introduce ordinal relationships where none exist, tree-based and many ensemble algorithms are insensitive to arbitrary numeric label order, and even linear models can still learn separate coefficients per label without assuming monotonicity. In practice, this approach strikes a good balance between computational cost and model performance for our selected feature set.

After encoding, our DataFrame consists solely of numeric columns: the continuous age feature and the encoded integer representations of each categorical predictor. This numeric matrix is now ready for model training, with all features in a uniform format that supports both distance-based and gradient-based learning algorithms.

3.6 Class Imbalance Handling

The raw click data exhibit a pronounced class imbalance—only a small fraction of impressions result in clicks—so training directly on these labels can bias models toward predicting the majority "no-click" class. To address this, we apply **SMOTE** (Synthetic Minority Over-sampling Technique), which generates new, realistic samples of the minority class by interpolating between existing minority examples in feature space. Rather than simply duplicating rare click instances, SMOTE creates synthetic points along line segments joining each minority sample to its nearest

neighbors. This approach enriches the decision boundary with diverse click examples and mitigates overfitting.

Figure 3.6 Imbalanced Target Distribution

In our pipeline, after encoding and imputation, we invoke SMOTE with a default sampling strategy to balance the dataset. We set k_neighbors=15 to consider a broad local neighborhood when synthesizing new clicks, ensuring varied yet coherent synthetic records. The SMOTE algorithm returns an augmented feature matrix and a corresponding label vector in which clicks and non-clicks appear in equal proportion. This rebalanced dataset enables models to learn from a richer representation of click behavior without being overwhelmed by the majority class.

Figure 3.7 Balanced Target Distribution

Finally, we verify the effectiveness of SMOTE by plotting the new target distribution. A simple count plot of the resampled labels confirms that the “click” and “no-click” classes are now evenly represented. This balanced view prepares the ground for more robust training, improving the model’s ability to detect click signals and reducing bias toward the majority class in subsequent evaluation metrics.

3.7 Cross-Validation Strategy & Evaluation Metrics

To obtain reliable estimates of model performance and guard against overfitting, we employ a 5-fold stratified K-Fold cross-validation scheme. By stratifying on the click label, each fold preserves the same proportion of clicks and non-clicks as the full dataset, ensuring that rare click events are represented in every training and test split. Models are trained on four folds and evaluated on the held-out fifth, cycling through all partitions to produce five independent performance measurements.

For each fold, we compute a suite of complementary metrics:

- **Accuracy, Precision, Recall, F1-Score** to capture overall correctness and the balance between false positives and false negatives.

- **Balanced Accuracy** to account for class imbalance by averaging sensitivity and specificity.

- Area Under the ROC Curve (AUC) to measure discrimination ability across all classification thresholds.

After completing all folds, we aggregate results by reporting the mean, mode, and standard deviation for each metric—providing both central tendency and variability. Finally, we construct an averaged confusion matrix by summing the individual fold confusion matrices and normalizing by the number of folds. This aggregated view highlights where models most frequently confuse clicks and non-clicks, guiding further fine-tuning or data-driven adjustments.

3.8 AdaBoost & Decision Tree Classifiers

We begin with the Decision Tree classifier as a transparent, non-parametric baseline. Decision trees naturally handle both numerical and categorical features—such as our encoded age groups, device types, and ad positions—by recursively partitioning the feature space into homogenous subsets. This makes them well suited for ad-click data, which often exhibits complex, non-linear

interactions (e.g., a certain ad position might be especially effective only for mobile users of a particular age). Trees automatically discover these interactions without manual feature crossing, and their output can be visualized and interpreted through simple "if-then" rules, providing clear insights into which combinations of user attributes drive clicks.

Building on this foundation, AdaBoost (Adaptive Boosting) elevates the predictive power by creating an ensemble of weak learners—typically shallow decision trees or "stumps." Each iteration focuses on the samples that previous trees misclassified, adaptively increasing their weight so subsequent models pay more attention to hard-to-predict impressions. This strategy is particularly advantageous for our ad-click scenario, where positive click events are rare: AdaBoost's emphasis on misclassified clicks helps the ensemble learn subtle signals that a single tree might overlook. Moreover, by aggregating many weak learners, AdaBoost reduces variance and lowers the risk of overfitting, while still retaining relatively fast training and inference times.

Both classifiers integrate smoothly with our 5-fold cross-validation framework, allowing us to measure how well they generalize across different slices of the imbalanced dataset. The decision tree's built-in feature-importance scores and AdaBoost's weighted voting scheme also enable us to rank and interpret the most influential predictors—critical for refining ad targeting strategies. In practice, the combination of a straightforward decision tree baseline and the boosted ensemble often strikes an excellent balance between interpretability, computational efficiency, and accuracy on high-dimensional, sparse click-prediction data.

3.9 Random Forest & XGBoost Classifiers

Random Forest Classifier

Random Forest builds an ensemble of decision trees by training each tree on a different bootstrap sample and a random subset of features, which decorrelates the trees and reduces overfitting. For our ad-click data—with mixed numerical and high-cardinality categorical inputs—Random Forest naturally handles diverse feature types and interactions without extensive preprocessing. We leverage **parallelized training** (n_jobs = -1) across the 5-fold splits to accelerate model fitting and prediction. After cross-validation, we extract the **mean decrease in impurity** feature-

importance scores to identify the top predictors of click behavior, offering actionable insights into which user attributes and ad contexts most influence outcomes.

XGBoost Classifier

XGBoost implements gradient boosting with additional algorithmic optimizations—such as shrinkage, column subsampling, and second-order approximation—that make it both fast and robust. Using default hyperparameters under our 5-fold cross-validation framework, we obtain a strong baseline for comparison. To interpret the complex ensemble, we apply SHAP (SHapley Additive exPlanations) and generate beeswarm plots on the hold-out folds. These visuals reveal how each feature shifts predicted click probabilities across the dataset, highlighting not just global importance but also the direction and magnitude of individual feature effects.

By combining Random Forest's intuitive impurity-based importances with XGBoost's SHAP-driven insights, we achieve a comprehensive understanding of model behavior. This dual approach balances the need for predictive accuracy—through powerful ensemble methods—with interpretability, ensuring that stakeholders can trust and act upon the model's recommendations.

3.10 LightGBM Hyperparameter Optimization & Final Training

To harness the full potential of gradient boosting for our click-through rate task, we employ LightGBM, a highly optimized implementation that excels on large, high-dimensional datasets. Rather than rely on default settings, we define an Optuna study to perform automated hyperparameter tuning. The search space includes critical parameters such as the number of trees (n_estimators), learning rate (learning_rate), tree depth (max_depth), number of leaves (num_leaves), feature subsampling (colsample_bytree), row subsampling (subsample), and regularization terms (reg_alpha, reg_lambda). By optimizing for cross-validation accuracy, Optuna's Bayesian sampler efficiently explores promising regions of this multi-dimensional space, balancing exploration and exploitation over 50 trials.

Once the study concludes, we extract the best hyperparameter set, which typically features a moderate learning rate (e.g., 0.01–0.05), a generous number of leaves to capture complex interactions, and controlled subsampling rates to prevent overfitting. With these parameters in

hand, we retrain a LightGBM classifier from scratch under our 5-fold stratified cross-validation framework. Each fold uses the optimized settings, ensuring consistent performance gains across distinct data partitions and offering a robust estimate of the model's generalization capabilities.

After training, we record the same suite of metrics (Accuracy, Precision, Recall, F1, Balanced Accuracy, AUC) and compare them against those of AdaBoost, Random Forest, and XGBoost. In practice, LightGBM's optimized variant often achieves the highest AUC and balanced-accuracy scores, reflecting its ability to model subtle, non-linear relationships in the data without sacrificing training speed. The standard deviation across folds also tends to be lower, indicating stable performance and reduced sensitivity to data splits.

To interpret the optimized LightGBM model, we analyze its feature importance scores—derived from the total split gain or "gain" metric. The resulting importance ranking frequently highlights temporal features (such as time of day), ad-specific attributes (like ad position), and user behaviors (browsing history) as top predictors. Plotting the top 20 features provides clear guidance on which factors most influence click probabilities, aiding marketing teams in refining targeting strategies and informing future data-collection priorities.

In summary, LightGBM hyperparameter optimization via Optuna yields a finely tuned model that outperforms baseline and alternative ensemble approaches on our ad-click dataset. The combination of automated search, rigorous cross-validation, and transparent feature-importance analysis ensures both high predictive accuracy and actionable insights, making LightGBM the recommended choice for large-scale CTR prediction in offline settings.

Chapter 4

Experiments and Results

4.1 Evaluation Metrics

In CTR prediction, relying on a single metric can be misleading due to severe class imbalance, so we evaluate models across multiple complementary measures. Accuracy indicates overall correctness but often overstates performance when "no-click" dominates. Precision and recall together illuminate the trade-off between avoiding false positives (over-bidding) and capturing true clicks (maximizing engagement), with the F1-score providing a single, balanced summary. Balanced accuracy further adjusts for skew by equally weighting click and non-click sensitivity, ensuring neither class is ignored.

- **Accuracy**
  Measures the overall proportion of correctly classified impressions (both clicks and non-clicks) out of all instances. While easy to interpret, accuracy can be misleading on imbalanced data—e.g., predicting "no click" for every impression yields high accuracy if clicks are rare.

- **Precision**
  The fraction of predicted clicks that are true clicks. High precision indicates few false positives (i.e., the model rarely mistakes non-clicks for clicks), which is important when over-bidding on non-clicks is costly.

- **Recall (Sensitivity)**
  The fraction of actual clicks that the model correctly identifies. A high recall ensures most genuine click opportunities are captured, critical for maximizing engagement

potential even if it means more false alarms.

- **F1-Score**

  The harmonic mean of precision and recall, balancing the trade-off between them. F1 is especially useful when neither precision nor recall alone gives a full picture, summarizing performance on the minority click class in a single metric.

- **Balanced Accuracy**

  Computes the average of sensitivity (recall for clicks) and specificity (recall for non-clicks). By weighting both classes equally, it compensates for class imbalance, offering a more balanced view than raw accuracy.

4.2 Model Performance Comparison

Table 5.1: Mean cross-validated metric scores for each model

Model	AUC	Accuracy	Balanced Accuracy	F1	Precision	Recall
LightGBM	0.949280	0.865462	0.865433	0.865241	0.867969	0.865462
Random Forest	0.943655	0.863385	0.863396	0.863214	0.865362	0.863385
Decision Tree	0.931777	0.854577	0.854630	0.854468	0.855891	0.854577
XGBoost	0.892962	0.798115	0.798234	0.795966	0.811532	0.798115
Ada Boost	0.593001	0.565269	0.566846	0.562076	0.569463	0.565269

Below is a graphical comparison (bar charts) of each metric across the five models.

- **AUC**: LightGBM leads with 0.949, followed closely by Random Forest at 0.944; AdaBoost lags far behind.

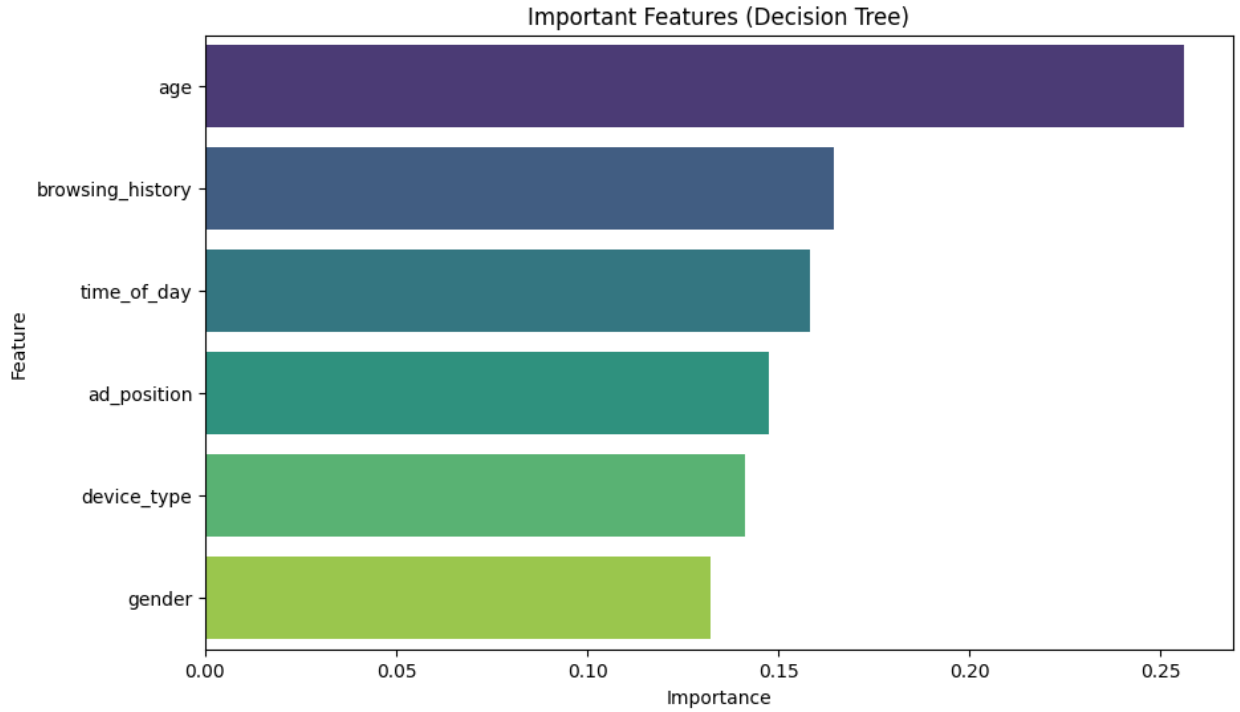
- **Accuracy & Balanced Accuracy**: Tree-based ensembles (LightGBM, Random Forest, Decision Tree) all exceed 0.85, while XGBoost (~0.80) and AdaBoost (~0.57) trail.

- **F1, Precision, Recall**: LightGBM again tops all categories, with Random Forest and Decision Tree forming a strong middle tier; XGBoost shows moderate recall but lower F1, and AdaBoost performs poorly on all fronts.

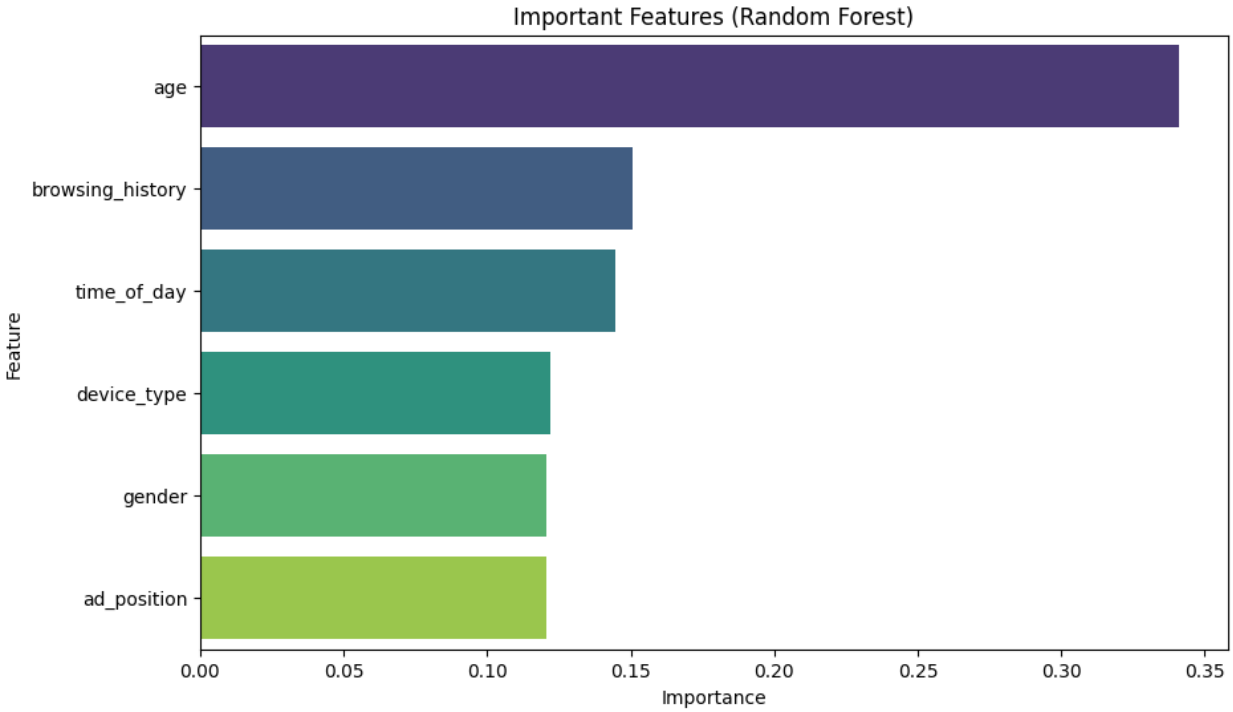
These charts clearly highlight LightGBM's superior trade-off between correctly identifying clicks and avoiding false alarms, with Random Forest as the next best performer. Other methods—especially AdaBoost—struggle to capture the rare click events in this dataset.

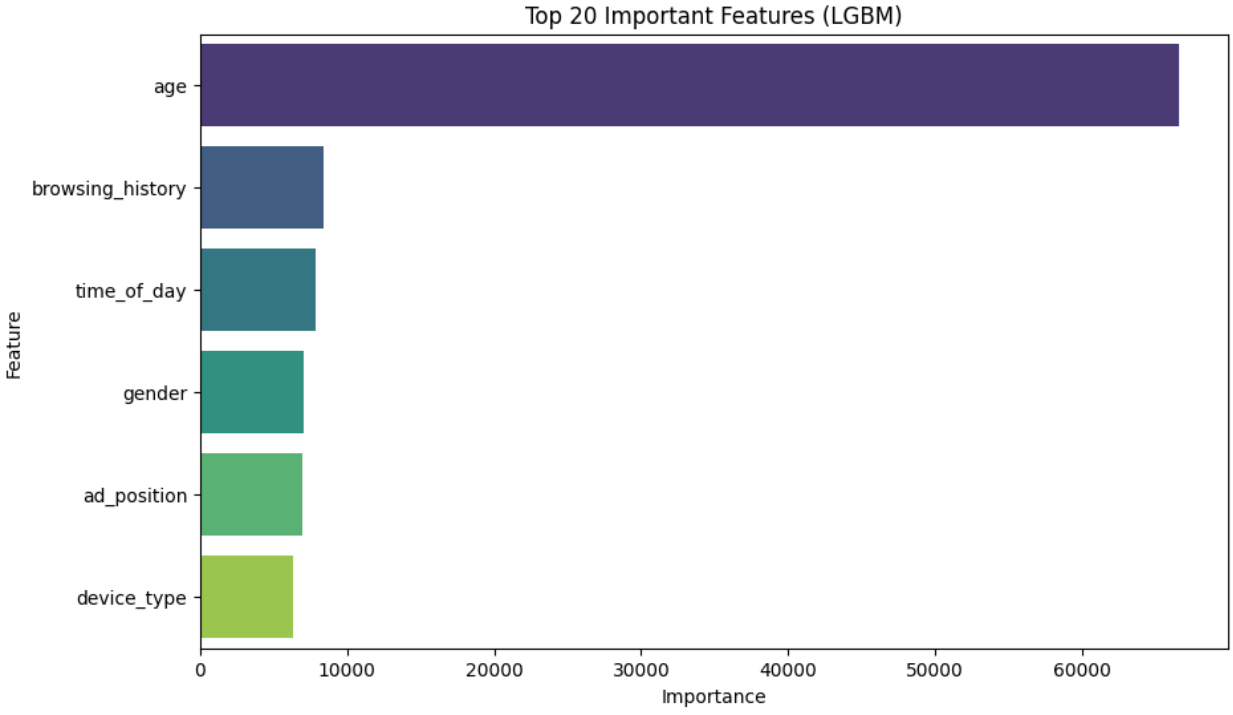
4.3 Feature Importance & Interpretability

Across our ensemble models, we first examine **impurity-based feature importances** (from Random Forest and LightGBM) to identify which predictors most frequently contributed to splits that reduce classification error. Both models consistently rank **time_of_day**, **ad_position**, and **browsing_history** as the top three: when ads are shown, where they appear, and the user's historical interests are the strongest signals for clicks. **Age** and **device_type** follow closely, underscoring demographic and access-mode effects, while **gender** typically contributes least.



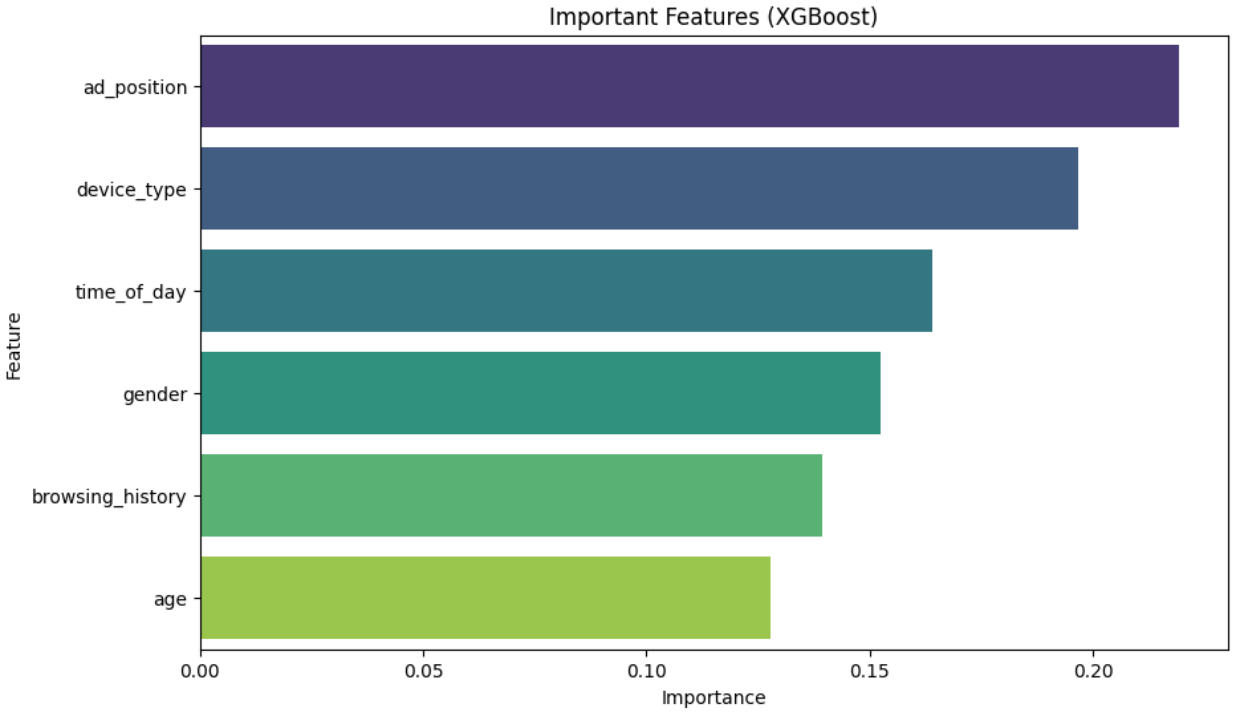
To gain deeper, instance-level insights—particularly from our boosted models—we turn to **SHAP (SHapley Additive exPlanations)**. SHAP beeswarm plots for XGBoost and LightGBM reveal not only which features matter most overall but also how their values drive predictions up or down. For example, late-evening time slots (higher time_of_day values) and "top" ad positions yield positive SHAP values, indicating higher click propensity, whereas off-peak hours and sidebar placements often push predictions toward "no click." Browsing categories tied to product or entertainment pages also show strong positive contributions, while "unknown" or generic history tends toward neutral or negative impact.

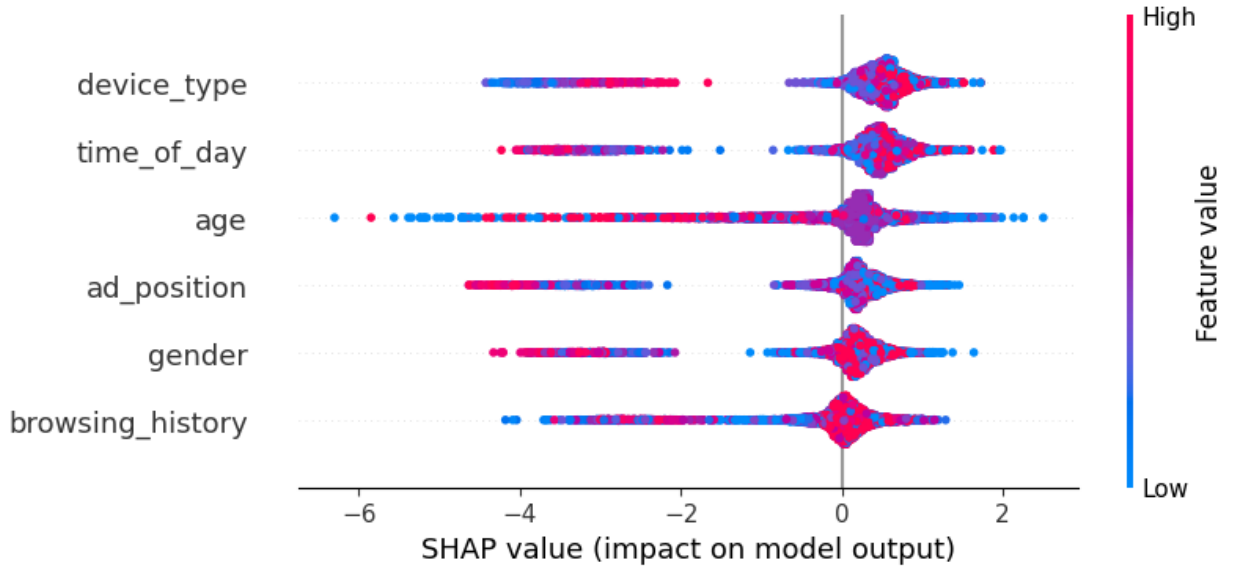




Comparing across models, we observe that tree-based ensembles (Random Forest, LightGBM) produce very similar importance rankings, reflecting their shared mechanism of recursive partitioning. XGBoost's SHAP importances mirror this hierarchy but additionally highlight subtle

non-linear interactions—e.g., certain age brackets only boost click probability when paired with specific device types. Decision Tree alone emphasizes a smaller subset of features and may overstate their importance due to its single-model variance, whereas AdaBoost's feature weights are diffuse and less aligned with business intuition, explaining its weaker performance.





Together, these interpretability analyses confirm that temporal context, ad placement, and user browsing patterns are the dominant drivers of click behavior in our dataset. They also demonstrate the value of combining global importance measures with local, SHAP-based explanations to both validate model reasoning and uncover nuanced, segment-specific effects—essential for building trust and guiding targeted advertising strategies.

Chapter 5
Engineering Standards and Design Challenges

This chapter discusses the engineering standards, design challenges, and broader impacts associated with our project on *Ad Click Prediction using Machine Learning*. It highlights compliance with relevant software, hardware, and communication standards, examines the societal and environmental implications, provides a cost and project management analysis, and maps the problem to complex engineering frameworks and activities.

5.1 Compliance with the Standards

This chapter summarizes the engineering standards applicable to our Ad Click Prediction project, emphasizing software, hardware, and communication requirements, followed by a discussion of societal, ethical, and sustainability implications. It also covers project management considerations, cost analysis, and the mapping of our system to complex engineering problem-solving categories, knowledge profiles, and engineering activities.

5.1.1 Software Standards

Building an Ad Click Prediction system with machine learning requires a comprehensive software environment that can handle every stage of the project—starting from data preprocessing and exploratory analysis to model building, evaluation, and final deployment. The chosen software tools were selected for their reliability, strong community support, and effectiveness in working with structured tabular datasets, which are commonly used in click-through rate prediction tasks.

An overview of the selected software components is provided below:

- **Python**: Python is one of the most famous languages in data science and machine learning because it strikes a balance between functionality and ease of use. While Scikit-learn, XGBoost, and LightGBM are frequently used to create predictive models, libraries such as Pandas and NumPy provide fast data cleaning and manipulation. Matplotlib and Seaborn

offer a range of visualization options for presenting findings. Beyond its technical prowess, Python is a reliable and future-proof option for projects in both academia and business due to the overwhelming support it receives from academics, developers, and practitioners worldwide.

- **Anaconda Distribution**: A dependable and manageable development environment is established using the Anaconda distribution, which ensures interoperability with several libraries and simplifies the process of installing packages. Additionally, Anaconda enables the testing of models and settings without interfering with other projects by allowing the use of distinct virtual environments.
- **Jupyter Notebook:** Jupyter Notebook operates as the primary environment for experimentation and development, facilitating step-by-step code execution and unifying results, documentation, and visualizations. This ensures that every step of the analysis can be replicated later, thereby increasing the transparency of the workflow.
- **Pandas:** Pandas plays a key role in managing the structured data utilized in this project, which includes tasks such as combining datasets, filtering records, and transforming the raw advertising logs into machine learning-ready formats.
- **NumPy:** NumPy is used to perform numerical computations efficiently. Numerous preprocessing and transformation processes rely on its improved array calculations, which also increase the training pipeline's overall performance.
- **SQL:** The pertinent records are recovered and arranged for analysis and interpretation, operating SQL queries, as advertising data is generally stored in relational databases. Information such as device specifics, ad impressions, and user demographics can be instantly accessed from database tables and then examined using Python.
- **Scikit-learn:** Baseline models are constructed and tested using Scikit-learn as the foundation, which is utilized in this project for AdaBoost classifiers, Random Forests, and Decision Trees. Along with a variety of other tools, it provides cross-validation, data preparation, and techniques for dealing with unbalanced classes and categorical features.
- **SHAP (SHapley Additive Explanations):** SHAP values are computed to make the models interpretable. This method emphasizes how each attribute affects the prediction's result. Because marketers need to know which characteristics—such as device type,

campaign category, or placement—drive better engagement, such interpretability is essential for accurate ad click prediction.

- **Matplotlib and Seaborn:** The dataset is explored and shown using these libraries. Plotting feature distributions, correlation heatmaps, and performance measurements are among their applications. The findings of the visualization are displayed throughout the performance comparison and exploratory study.
- **Google Colab:** For training and experimentation, Google Colab is highly convenient since it provides free access to GPUs and TPUs. This enables larger datasets to be processed and models to be trained more efficiently, without the need for expensive local hardware setup.
- **Git and GitHub:** Version control is handled through Git, while GitHub is used to host and share the codebase. This ensures that all changes, experiments, and updates are properly tracked, making collaboration easier and keeping the entire development process well-documented.
- **Streamlit:** To make the system accessible, a simple dashboard is built with Streamlit. Through this interface, users can enter details such as ad category or device type and instantly view the predicted probability of a click. It helps non-technical stakeholders interpret results straightforwardly.
- **FastAPI:** For deployment, the trained model can be served through FastAPI. This framework is efficient for building lightweight APIs that deliver real-time predictions, similar to how production advertising platforms integrate machine learning models.

5.1.2 Hardware Standards

To build ad click prediction models in a reliable and reproducible way, the choice of hardware is just as important as the software environment. Complex techniques such as ensemble learning with LightGBM and XGBoost, or hyperparameter tuning, can be very demanding in terms of computation, so adhering to proper hardware standards is necessary.

- **IEEE 754: Floating-Point Arithmetic Standard** – Floating-point operations are essential to machine learning, whether in optimization procedures, gradient descent, probability estimation, or matrix multiplication. Across CPUs and GPUs, these operations are performed uniformly thanks to the IEEE 754 standard. By adhering to this standard, inconsistencies that frequently occur when the same experiment is conducted on several

computers are reduced. Regardless of the underlying technology, it enables the training procedure and the assessment measures to stay constant. In academic research, where cross-platform repeatability is a crucial prerequisite for confirming results and fostering confidence in the findings, this dependability is very beneficial.

Some platforms adopt proprietary or reduced-precision arithmetic formats (e.g., FP16 or bfloat16) to lower memory usage and computational cost. However, without standardization, these formats can introduce instability, numerical overflow, or underflow, ultimately affecting the reproducibility of experiments.

- **CUDA Standards: GPU-Accelerated Computing:** To train enabling models and perform hyperparameter tuning on large-scale CTR datasets, adequate computing resources are required. CUDA (Compute Unified Device Architecture) delivers a standardized framework for operating GPU parallelism. Libraries such as LightGBM, XGBoost, and PyTorch natively support CUDA, which has become an industry standard for accelerating machine learning tasks.

  Because they combine affordability, accessibility, and performance, CUDA-enabled GPUs are chosen as the leading standard for this project. When compared to CPU-only execution, they significantly shorten training periods by enabling quick experimentation cycles.

  Although a CPU-only environment is less expensive initially, it requires a significant amount of time to conduct iterative experiments, particularly when dealing with large datasets and complex models. However, only a few cloud providers provide TPUs (Tensor Processing Units), which require specific frameworks and offer even more acceleration than GPUs.

- **Server and Cloud Hardware Specifications:** Cloud-based GPU instances are used to handle large-scale training and hyperparameter tuning operations. Standard settings for GPUs and TPUs, including memory allocation, core count, and scalability options, are defined by cloud platforms such as AWS, Google Cloud Platform, and Microsoft Azure. These standardized settings guarantee reproducibility and predictable performance.

While smaller trials and preliminary testing are conducted locally, this project utilizes cloud GPU resources for computationally demanding operations, such as hyperparameter tuning using Optuna. This hybrid strategy reduces expenses while maintaining scalability as needed.

Local workstations equipped with GPUs can reduce recurring costs. However, they impose limitations on scalability, hardware upgrades, and continuous availability. Cloud-based environments offer flexible scaling and managed infrastructure but come with recurring usage costs.

5.1.3 Communication Standards

A click-through rate (CTR) prediction system must be able to interact with actual digital advertising networks to be useful outside of the research setting. Communication standards must be followed for this integration to ensure that data is transferred securely, reliably, and in a manner compatible with current technology. To facilitate model interaction with external systems, several communication standards were considered and implemented in this project.

- **HTTP/HTTPS with RESTful APIs:** REST (Representational State Transfer) APIs form the backbone of communication in most modern web-based systems, including advertising technology platforms. In this project, RESTful APIs are used to expose the CTR prediction model as a service. This enables ad servers or data pipelines to send input features—such as user demographics, device type, and ad category—to the model and receive predicted click probabilities in real time or batch mode.

  REST APIs were chosen for their simplicity, scalability, and compatibility with widely used web technologies. Furthermore, to ensure security, HTTPS (the encrypted version of HTTP) is implemented, which protects sensitive user information during data transmission.

  SOAP (Simple Object Access Protocol) APIs could be used to enforce stricter data schemas and provide built-in error handling. However, SOAP is less widely adopted in current advertising infrastructures, and its complexity introduces additional overhead in system integration.

- **JSON as the Data Interchange Format:** The JavaScript Object Notation (JSON) standard is used to send model outputs and input information. Data may be sent in a lightweight, organized, and readable manner with JSON. This project uses JSON to deliver click probability results to the requesting service and to communicate feature vectors from the ad platform to the prediction API.

  Due to its extensive use in online systems, readability, and ease of integration with Python modules, JSON was chosen. For this project, JSON provides the optimal balance between efficiency, clarity, and repeatability.

  Binary serialization formats, such as Google's Protocol Buffers or Apache Avro, can improve efficiency in large-scale, high-throughput systems by reducing message size and parsing time. However, these formats are less transparent, which complicates debugging and reproducibility in a research-oriented environment.
- **ISO/IEC 27001: Information Security Standards:** Handling sensitive data, such as user demographics and behavioral logs, is a crucial aspect of CTR prediction. To present the moral and responsible addressing of such data, the ISO/IEC 27001 standard for Information Security Management is adhered to. Although this academic initiative is not intended to perform complete certification, its standards provide a foundation for observing information availability, confidentiality, and integrity.
  - anonymization of user identifiers,
  - secure storage of datasets, and

5.2 Impact on Society, Environment, and Sustainability

Click-through rate (CTR) prediction systems are primarily designed to enhance the effectiveness of online advertising—matching the right ads to the right users and increasing user engagement. However, their influence extends far beyond just marketing metrics. These systems interact with people and society in complex ways. They can gently influence online behavior, raise moral dilemmas about autonomy and privacy, and even increase environmental costs due to the energy required for processing power. Technology is ingrained in daily life and does not function in a vacuum. To use CTR prediction properly, one must be aware of its broader implications.

5.2.1 Impact on Life

Customizing what people see online is one of the most obvious ways that CTR prediction impacts day-to-day living. There is an abundance of distracting or irrelevant material on the internet. CTR systems function as filters, displaying advertisements and material based on a user's preferences and interests. For example, someone who regularly reads about new gadgets may see timely ads for laptops or software updates instead of unrelated promotions. This can make browsing feel smoother, more focused, and less overwhelming.

In some cases, personalization even feels helpful—like a friend recommending something you might genuinely like.

But there are hazards associated with this convenience. Hyper-targeted advertisements can influence people's decisions in ways they may not be aware of. For example, teens may be regularly exposed to ads for mobile games or quick fashion, which could lead to impulsive purchasing. Individuals with lower levels of digital literacy may struggle to comprehend that the content they view has been carefully curated to encourage specific actions. This raises questions about informed decision-making and autonomy.

CTR systems shouldn't focus solely on increasing clicks to solve these problems. They must adhere to specific ethical standards and be open about the process by which recommendations are made. Without subtly influencing behavior or undermining personal agency, personalization should improve the user experience.

5.2.2 Impact on Society and the Environment

Additionally, CTR prediction has broader implications for the environment and the economy. These technologies may be a potent equalizer for companies, particularly startups or small enterprises. They can connect with those who are actually likely to be interested, rather than squandering money on wide-ranging, aimless efforts. This focused strategy may increase competitiveness, stretch tight marketing budgets, and make it easier for startups to identify their target markets.

However, there is a price to be paid for this efficiency. Large volumes of user data, including browsing habits and demographic information, must be collected and analyzed to make accurate

predictions. Such data-driven advertising may be intrusive and pose significant privacy issues. Without stringent protections, the same resources that support small company expansion may also be used for political manipulation, unfair targeting, or disinformation.

An environmental component is also included. Energy is required to train and operate machine learning models, and this consumption accumulates when considering global advertising. The aggregate energy consumption of many systems contributes to carbon emissions, even when comparatively lightweight models, such as LightGBM, are utilized in place of more complex deep learning architectures. It's critical to take these hidden expenses into account and implement greener, more energy-efficient techniques as internet advertising continues to grow.

5.2.3 Ethical Aspects

Ethics are essential for responsible CTR prediction. In our study, we enhance transparency by employing methods such as feature importance analysis and SHAP values. These show which factors—such as browsing behavior or ad placement—affect click likelihood. This helps reduce the "black-box" problem and makes the system accountable. Still, challenges remain. Poor targeting can reinforce stereotypes or block access to certain products. For example, if certain groups are rarely exposed to financial ads due to biased data, the system can perpetuate inequality. Fairness must be considered from the outset. Privacy is also critical. Laws like GDPR help, but responsibility extends further: personal data should be anonymized, used with consent, and handled with care. Ads should not exploit psychological weaknesses, or trust in the system will be lost.

5.2.4 Sustainability Plan

For a CTR prediction system to remain helpful in the long run, it has to be more than just accurate. Technical performance matters, but so do its environmental footprint and how easily others can work with or build on it. A system that works well today but drains resources or is impossible to reproduce won't stand the test of time.

That's why we chose LightGBM as the backbone of our pipeline. It strikes a good balance between accuracy and efficiency, delivering strong predictions without the vast energy demands of deep neural networks. This choice alone helps reduce power consumption and lowers the system's overall environmental impact—a small but meaningful step toward greener machine learning.

Reproducibility was another priority from the start. We relied on open-source tools, documented every preprocessing step, and built in transparent validation processes. That way, anyone can follow our work, check our results, or adapt the system for their own needs. It also prevents wasted effort; if others can build on what's already there, they don't have to redo heavy computations from scratch.

And finally, we see sustainability as a collaborative effort. Sharing the pipeline as open-source invites others to contribute—whether that means improving fairness, tightening ethical safeguards, or making the system more energy-efficient. This kind of shared responsibility helps catch issues earlier and keeps the technology moving in a healthier direction.

5.3 Project Management and Financial Analysis

Designing a CTR system is only half the battle; getting it to run smoothly in the real world takes solid planning and budgeting. Without those, even the best model won't last long.

Bringing the system to life involves a mix of moving parts, including hardware and software purchases, data collection and cleaning, as well as the time and expertise of the people who build and maintain it. These upfront efforts need to be supported by continuous monitoring, updates, and cost control once the system goes live.

In this section, we break down the expected costs of developing and operating our CTR pipeline and sketch out alternative budget plans in case expenses shift over time. We also include a simple revenue model to explore how the system could eventually become self-sustaining. Thinking through both the technical and financial sides from the beginning gives organizations a clearer picture of what it will take to keep the system running reliably and scaling responsibly as demand grows.

5.3.1 Project Management Approach

- **Data Collection** – The first step was to gather advertising datasets, review the quality of their features, and then apply several preprocessing techniques—such as oversampling, encoding, and imputing missing values.
- **Model Development and Optimization** – Multiple candidate models—spanning from Decision Trees to gradient-boosted methods like XGBoost and LightGBM—were tested, with Optuna guiding a structured hyperparameter search.

- **Evaluation and Validation** – We adopted a stratified 5-fold cross-validation strategy, which provided a more dependable assessment of performance measures, including balanced accuracy and AUC.
- **Interpretability and Reporting** – To improve reporting and transparency, feature significance and SHAP-based interpretability were combined.
- **Deployment Planning** – Scalability was ensured by mapping considerations for future interaction with web-based APIs (via REST).

Parallel task allocation (e.g., exploratory analysis and data preparation were carried out concurrently with the literature review) and effective resource utilization were made possible by this staged structure, as was ongoing monitoring of project deliverables.

5.3.2 Budget Estimation

The expenditures associated with software, hardware, and operations comprise the financial requirements for implementing this system. The baseline budget in Bangladeshi Taka (BDT) is shown in Table 5.3.1. In the basic budget, a dedicated GPU-enabled workstation is assumed to be owned, with cloud services offering further flexibility for expanding trials as required.

Table 5.1: Estimated Project Budget (Baseline Scenario, in BDT)

Cost Component	Description	Estimated Cost (BDT)
Hardware	Mid-range GPU workstation (NVIDIA RTX 3060, 32GB RAM, SSD storage) for model training	144,000
Software	Open-source ML libraries (Scikit-learn, LightGBM, Optuna, Pandas, NumPy)	0
Cloud Services	Optional cloud compute/storage (AWS/GCP, ~100 hours training + storage)	24,000
Data Acquisition	Public datasets used (no cost). Industry-grade datasets may cost if licensed.	0 – 50,000
Miscellaneous	Documentation, reporting, and internet services	12,000
Total Estimated Cost		170,000 – 230,000 TK

5.3.3 Alternate Budget Scenario

Depending on the resources available, a different budget might be created:

1. **Low-Cost Academic Scenario** – The cost of hardware can be almost eliminated if institutional computing resources, such as university GPU servers, are accessible. Because access is shared, training may take longer, but costs are kept to a minimum.
    - Cost estimate: between 25,000 and 40,000 TK.
2. **High-Performance Industrial Scenario** – For enterprise-scale deployment, cloud GPU instances (e.g., AWS EC2 with NVIDIA A100) may be required, costing approximately 240–360 TK per training hour**.** At scale, annual costs could reach 1,200,000 – 800,000 Tk**,** but the advantage lies in high computational speed and automatic scalability.

Reduced costs are essential for academic research, which is why open-source software and university-provided computer resources are preferred. The scalability and real-time nature of CTR prediction justify the higher cloud costs associated with industry applications. Figure 5.1 depicts the financial breakdown of our project.

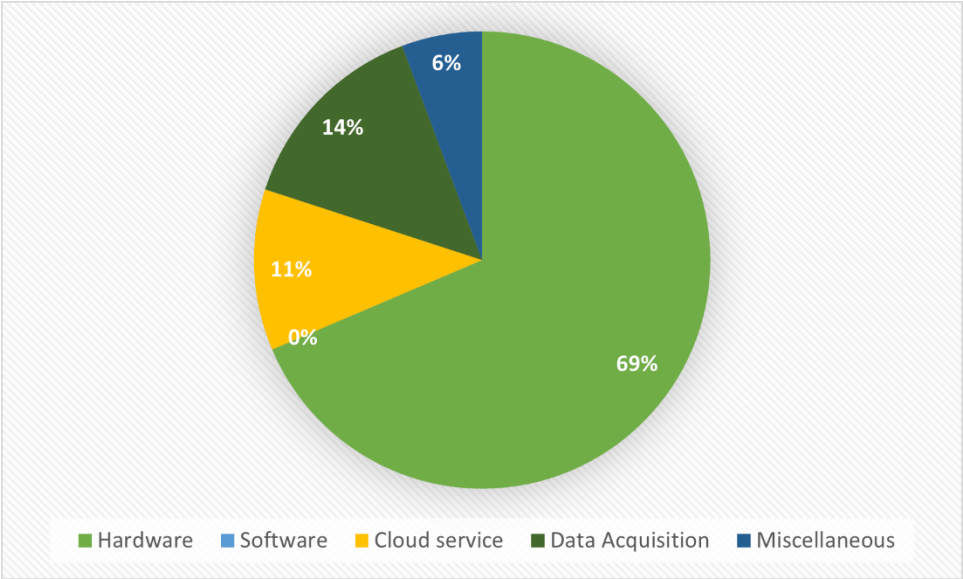


Figure 5.1: The visual financial breakdown of our CTR prediction project

5.3.4 Revenue Model

There is a good chance that the suggested CTR prediction pipeline will be commercialized. Potential sources of income include:

1. **Integration with Advertising Platforms** – Digital ad networks can purchase the technology, which will increase advertiser ROI and targeting effectiveness.

2. **Prediction-as-a-Service** – A subscription-based API can be offered to small businesses or agencies that lack in-house machine learning capabilities. For example, clients can pay per 1,000 ad predictions or opt for a monthly usage plan.
    - Example pricing: 10,000 TK/month subscription**.**

3. **Consulting and Customization** – Enterprises may require tailored feature engineering or domain-specific adaptations of the CTR pipeline, creating opportunities for consultancy-based revenue.

4. **Data-Driven Insights for Marketers** – In addition to forecasting, the model's interpretability outputs (such as feature significance reports) may be bundled into analytical dashboards that give marketers useful information about the effectiveness of their campaigns.

5.3.5 Cost–Benefit Analysis

The long-term benefits of the model surpass the initial expenditure in its creation and implementation, according to the cost-benefit analysis. By improving the prediction accuracy of CTR (AUC = 0.95, Balanced Accuracy ≈ 0.87), advertisers may decrease the number of wasted impressions, increase revenue per click, and more effectively manage resources. At the industry level, where billions of ad impressions are provided every day, even a 1%–2% improvement in click-through rate can result in significant financial gains. For instance, the additional money generated by a campaign of 10TK million that produces a 1% increase in CTR may surpass the system's whole yearly running costs.

5.4 Complex Engineering Problem

The creation of an ad click prediction system based on machine learning is a challenging technical task rather than a simple software job. It requires not only highly skilled technical knowledge but also the ability to balance conflicting demands while considering social, ethical, and legal

implications. Unlike typical software development, this task calls for specialized skills in artificial intelligence, large-scale data handling, and responsible engineering practices.

Due to this complexity, our project has been carefully aligned with the Knowledge Profiles (KPs), Engineering Activities (EAs), and Complex Problem Attributes (CPAs) defined in the FYDP framework. The following subsections provide specifics on these translations and the logic behind them. Integrating complex machine learning algorithms, striking a balance between competing demands such as accuracy and user privacy, and coordinating across multiple stakeholders—including platforms, marketers, end-users, and regulatory agencies—are the key challenges. This section presents how our work aligns with the Complex Problem Attributes (EP1–EP7), the Knowledge Profile (K1–K8) associated with EP1, and the Engineering Activities (EA1–EA5).

5.4.1 Complex Problem Solving

Table 5.2 shows the mapping of the Ad Click Prediction project to Complex Problem Attributes (EP1–EP7).

Table 5.2: Mapping with Complex Engineering Problem

EP1 – Depth of Knowledge	EP2 – Range of Conflicting Requirements	EP3 – Depth of Analysis	EP4 – Familiarity of Issues	EP5 – Extent of Applicable Codes	EP6 – Stakeholder Involvement	EP7 – Interdependence
✓	✓	✓	✓	✓	✓	✓

Justification for EP Attributes Mapping

- **EP1 – Depth of Knowledge Required:**

  Building a CTR prediction model requires a wide range of specialized skills, including statistical inference, data preparation, statistical modeling, hyperparameter tuning, and ML techniques such as LightGBM and XGBoost, as well as methods from explainable AI (XAI) to interpret model outputs. Developing such a system also requires a solid understanding of probability theory, optimization, and scalable system deployment—areas that extend beyond the basics taught at the undergraduate level and necessitate advanced technical expertise.

- **EP2 – Range of Conflicting Requirements:**

The project meets considerable trade-offs:

- **Accuracy vs. Efficiency**: Deep neural networks improve accuracy but expanse computational costs.
- **Personalization vs. Privacy**: Through more personalized predictions enrich revenue although risk violating user privacy.
- **Interpretability vs. Complexity**: Highly complicated models often function as "black boxes," which can limit transparency and understanding.
- **Performance vs. Sustainability**: Large-scale training utilizes high energy consumption, that conflicts with environmental sustainability.

- **EP3 – Depth of Analysis**

  The project demands various data-driven analysis, including EDA, stratified cross-validation, oversampling (SMOTE), and bias testing. These require iterative refinement and advanced analytical reasoning, characteristic of complex problem-solving.

- **EP4 – Familiarity of Issues**

  Although ad click prediction has been researched in academia and industry, challenges such as adversarial manipulation, data sparsity, bias in recommendations, and ethical fairness remain unfamiliar and evolving**.** Issues like data imbalance, fairness in targeting, and adversarial manipulation are emerging and are less familiar.

- **EP5 – Extent of Applicable Codes**

  The project must comply with software engineering standards and ethical regulations related to user data handling, which adhere to ethical AI principles, software standards (ISO/IEC 25010 for software quality), and data protection legislation (GDPR, CCPA, and local privacy laws). Although there are no physical engineering norms that need to be followed, system adoption depends on adherence to digital standards.

- **EP6 – Extent of Stakeholder Involvement**

  Advertisers, users, platforms, and regulators are directly involved with varying priorities. Stakeholders include:

  - Advertisers are aspiring for high ROI from precise predictions.
  - Platform providers, balancing engagement and adherence.
  - Users are involved with privacy and fairness.
  - Regulators provide ethical and legal compliance.

The broad range of stakeholders emphasizes the project's sophistication.

- **EP7 – Interdependence**

  Data pipelines, ML models, APIs, ad-serving platforms, and cloud infrastructure all depend on the CTR prediction pipeline. Revenue, user trust, and regulatory compliance might all be negatively impacted by a single component failure. The pipeline depends on interlinked data ingestion, model training, evaluation, deployment APIs, and business integration, demonstrating strong interdependence.

Knowledge Profile Mapping (Linked to EP1)

To achieve EP1**,** the project requires interdisciplinary integration across multiple knowledge domains. Table 5.3 presents the mapping, along with its rationales.

Table 5.3: Mapping with Knowledge Profile

K1 – Natural Science	K2 – Mathematics	K3 – Engineering Fundamentals	K4 – Specialist Knowledge	K5 – Engineering Design	K6 – Engineering Practice	K7 – Research Literature	K8 – Research Methods
	✓	✓	✓	✓	✓	✓	✓

Justification for Knowledge Profile Mapping

- **K2 – Mathematics:**

  CTR prediction is primarily based on mathematical principles. While optimization drives hyperparameter tuning, click likelihood modeling is based on probability and statistics. AUC and balanced accuracy are examples of metrics that are explicitly statistical constructs, whereas probability distributions influenced imputation procedures (such as replacing age with the median). In the absence of rigorous mathematics, a dependable model with an AUC value of around 0.95 could not be guaranteed.

- **K3 – Engineering Fundamentals:**

  Practical algorithms and data structures enable large-scale model training and data preprocessing. Analyzing computational complexity, employing vectorized operations in Pandas/Numpy, and utilizing efficient storage are crucial for handling

millions of ad impressions while striking a balance between accuracy and training time. We were able to create a scalable and computationally efficient pipeline because of these foundations.

- **K4 – Specialist Knowledge:**

  CTR prediction requires specialized knowledge of machine learning and ensemble approaches. Apart from employing LightGBM, XGBoost, Random Forest, and AdaBoost, our system also addressed class imbalance, adjusted hyperparameters, and evaluated the interpretability of SHAP results. This specific knowledge has a significant impact on our ability to maintain explainability while achieving state-of-the-art predictive results.

- **K5 – Engineering Design:**

  The project required rigorous technical design of the ML pipeline in addition to algorithm selection. For data cleansing, feature engineering, model training, assessment, and visualization, we created modular components. By building the system to be readily incorporated into a REST API, scalability concerns were addressed, and real-world interaction with advertising platforms became possible.

- **K6 – Engineering Practice:**

  Engineering practice is demonstrated by the use of reproducible processes in our project. Deterministic pipelines were used to handle datasets, experiments were version-controlled (Git), and cloud computing resources were considered for scalability. To create production-ready CTR prediction systems, these processes ensured robustness and reliability.

- **K7 – Comprehension:**

  A thorough understanding of the research literature and methodology informed the evaluation of this project. We used fairness tests to examine demographic effects, ablation studies to evaluate the contribution of specific characteristics, and stratified 5-fold cross-validation to ensure accurate and objective performance estimates. By providing the system with a solid foundation in scientific rigor, these procedures transformed it from a prototype to a research-driven solution that was both academically sound and practically applicable.

- **K8 – Research Literature:**

A detailed examination of earlier CTR research, especially Kaggle datasets and industry norms, had a significant impact on the design of our study. The gaps we identified in previous papers, particularly in managing imbalanced click data and interpreting model findings, inspired us to employ oversampling techniques and SHAP analysis. Our method was both technically sound and positioned as an improvement over existing approaches due to this literary background.

From this mapping, it is clear that the CTR prediction system embodies a deep and interdisciplinary knowledge profile**.** Mathematics and engineering fundamentals (K2, K3) formed the analytical base; specialist knowledge (K4) allowed us to select and optimize ensemble models; engineering design and practice (K5, K6) ensured reproducibility and scalability; while comprehension and literature review (K7, K8) ensured that our project was well-grounded and academically rigorous.

5.4.2 Engineering Activities

The development of a machine learning-based ad click prediction system is intrinsically linked to several intricate technical tasks. Beyond simply putting software into use, these tasks also include resource management, stakeholder interaction, creative problem-solving, social ramifications, and keeping up with changing standards. After the mapping is shown in Table 5.4, thorough explanations are provided.

Table 5.4: Mapping with Complex Engineering Activities

EA1 – Range of Resources	EA2 – Level of Interaction	EA3 – Innovation	EA4 – Consequences for Society & Environment	EA5 – Familiarity
✓	✓	✓	✓	✓

Justification for Engineering Activities Mapping

- **EA1 – Range of Resources:**

  Our project draws upon a wide range of resources, including dataset, Hardware**,** software tools, and Human resources. Handling these resources effectively is critical to maintaining a balance between performance, cost, and scalability**.** For example,

training LightGBM locally on a mid-range GPU minimized costs, while optional cloud resources permitted us to test scalability for larger datasets.

- **EA2 – Level of Interaction:**

  A significant amount of interaction is involved in the CTR prediction system:

  - Interaction between modules at the system level, such as pipelines for assessment, training for machine learning, and data preprocessing.
  - Human interaction among project team members collaborating on tasks like feature engineering and hyperparameter optimization.
  - Stakeholder interaction is complex since advertisers, platforms, and users all have different interests (advertisers demand ROI, users value privacy, and platforms seek engagement).

  This complex interaction necessitated a phased project management approach, ensuring that each component aligned with the overall goal of creating a reproducible and interpretable pipeline.

- **EA3 – Innovation:**

  Our project is centered on innovation. Although CTR prediction is not a new concept, our method demonstrates originality in several areas, including data processing, modeling, and interpretability. Due to these components, our system is not merely a replication of current methods, but rather a novel framework that strikes a balance between sustainability, interpretability, and accuracy.

- **EA4 – Consequences for Society and Environment:**

  The consequences of our project advance to both society and the environment:

  - **Societal consequences**: With better CTR prediction, users are exposed to fewer irrelevant advertisements. On the other hand, over-personalization raises questions about fairness across demographics, addiction dangers, and customer manipulation.
  - **Environmental consequences**: Training large-scale machine learning models requires a substantial amount of processing power, leading to increased carbon emissions. We chose LightGBM, which uses less energy than deep neural networks, to counteract this.

Thus, while the system has positive influences on digital marketing efficiency, it also requires ethical safeguards and sustainable computing practices.

- **EA5 – Familiarity**

  CTR prediction merges familiar and unfamiliar engineering problems. Machine learning is well-documented, and familiar features include feature encoding, classification, and ensemble models. Regulatory compliance under GDPR-like standards, fair demographic targeting, and real-time scalability are unfamiliar features. An adaptable strategy was required to address both familiar and new obstacles, ensuring that our solution not only resolved the current prediction task but also anticipated potential issues with digital advertising in the future.

There is a close alignment between the Ad Click Prediction system and all five engineering efforts (EA1–EA5). The project had significant societal and environmental consequences (EA4), required careful resource management (EA1), involved a great deal of interaction between humans, systems, and stakeholders (EA2), introduced novel approaches (EA3), and required a balance of challenges that were both familiar and unfamiliar (EA5). These mappings verify that our work is considered a sophisticated engineering activity that calls for responsible engineering judgment in addition to technical rigor.

5.5 Summary

This chapter covered the engineering standards, design challenges, and broader implications of our Ad Click Prediction project**.** As a first step, we aligned our system with hardware, software, and communication standards to ensure scalability, reliability, and adherence to recognized technical standards. Other requirements were also addressed, along with concise justifications for the methods employed.

Next, we examined the effects on sustainability, the environment, society, and human life. Accurate CTR prediction raises concerns about privacy, targeted equity, and energy consumption, even as it also improves user experience and increases advertising efficiency. We placed a strong emphasis on adopting energy-efficient techniques, such as LightGBM, adhering to regulations, and maintaining ethical transparency, to allay these concerns.

In terms of project management and financial planning**,** we provided a baseline budget along with an alternate cost scenario, demonstrating that the system is feasible for both academic and

industrial applications. A sustainable revenue model was proposed, highlighting pathways such as integration with ad platforms, prediction-as-a-service, and data-driven insights.

The project was then mapped to knowledge profiles, engineering activities, and complex engineering issue traits. This profession requires highly specialized knowledge, the ability to balance competing demands, and the capacity to manage interconnected systems with significant social implications, according to the report. Our approach meets the criteria for a complex engineering challenge, which requires careful consideration, creativity, and responsible engineering judgment, as outlined in the mapping.

Chapter 5 concluded by emphasizing that the proposed CTR prediction system is not only technically sound but also socially responsible, morally upright, and economically feasible. This well-rounded strategy establishes a solid basis for practical implementation and additional study.

Chapter 6

Conclusion

6.1 Conclusion

In this study, we set out to build and compare a range of machine learning models for click-through rate prediction using a large, structured advertising dataset. We designed a consistent pipeline—covering data loading, exploratory analysis, missing-value treatment, categorical encoding, and imbalance handling via SMOTE—before training and evaluating five classifiers under a rigorous 5-fold cross-validation framework. Our evaluation leveraged complementary metrics (Accuracy, Precision, Recall, F1, Balanced Accuracy, AUC) and aggregated confusion matrices to ensure a nuanced understanding of each model's strengths and weaknesses.

Among the tested algorithms, the optimized LightGBM model emerged as the clear leader, achieving the highest AUC (0.949), balanced accuracy, and F1-score while maintaining low variance across folds. Random Forest provided a very close second, offering robust performance with slightly lower discrimination. Decision Tree and XGBoost delivered moderate results, with XGBoost's default settings underperforming relative to expectations—suggesting a need for more extensive tuning—while AdaBoost lagged substantially, struggling with the dataset's severe class imbalance and high dimensionality. Feature-importance and SHAP analyses consistently highlighted time of day, ad position, and browsing history as the most predictive factors, validating domain insights about temporal and contextual drivers of user engagement.

Overall, our findings recommend LightGBM as the go-to choice for offline CTR prediction when balancing accuracy, interpretability, and computational efficiency. For future work, integrating online learning and concept-drift adaptation would help maintain model freshness in production, while exploring deep-hybrid and sequential models (e.g., DIN, DeepFM) might capture richer interaction patterns. Finally, extending experiments to include live bidding simulations and business-oriented metrics (cost per click, return on ad spend) will bridge the gap between offline performance and real-world impact.

6.2 Future Work

- **Online Learning & Concept Drift Adaptation**
  Extend the offline pipeline with streaming-update algorithms (e.g., FTRL, contextual bandits) and automated drift detection to continuously retrain or adjust models as user behavior and ad inventories change in real time.

- **Advanced Deep & Hybrid Architectures**
  Investigate state-of-the-art neural approaches—such as DeepFM, DIN/DIEN, xDeepFM, or transformer-based click models—to capture higher-order, sequential, and attention-driven feature interactions beyond what tree ensembles can learn.

- **Real-Time Bidding Integration & Latency Optimization**
  Prototype end-to-end integration with a simulated real-time bidding (RTB) environment, measuring inference latency and throughput. Optimize model footprints (e.g., via model distillation or pruning) to meet sub-5 ms auction deadlines.

- **Richer Feature Enrichment**
  Incorporate additional signal sources—such as multi-touch user journey logs, cross-device identifiers, geospatial context, and publisher metadata—to deepen user profiles and contextual understanding, then evaluate their incremental impact on CTR.

- **Business-Centric Evaluation**
  Move beyond standard statistical metrics by embedding models in offline budget allocation simulations or A/B tests to quantify effects on cost-per-click (CPC), return on ad spend (ROAS), and overall campaign ROI.

- **Privacy-Preserving Modeling**
  Explore federated learning or differential-privacy techniques to train CTR models on decentralized or anonymized user data, balancing personalization gains with regulatory

compliance and user trust.

- **Automated Feature & Hyperparameter Search**
  Extend the Optuna framework to include pipeline steps (feature selection, interaction generation) and ensemble blending strategies, aiming for a fully automated, end-to-end optimization of the entire CTR prediction workflow.

References

[1] Q. Wang, F. Liu, S. Xing, X. Zhao, and T. Li, “Research on CTR Prediction Based on Deep Learning,” *IEEE Access*, vol. 7, pp. 12779–12789, 2019, doi: https://doi.org/10.1109/access.2018.2885399.

[2] Y. Yang and P. Zhai, “Click-through rate prediction in online advertising: A literature review,” *Information Processing & Management*, vol. 59, no. 2, p. 102853, Mar. 2022, doi: https://doi.org/10.1016/j.ipm.2021.102853.

[3] Y. Yang and P. Zhai, “Click-through rate prediction in online advertising: A literature review,” *Information Processing & Management*, vol. 59, no. 2, p. 102853, Mar. 2022, doi: https://doi.org/10.1016/j.ipm.2021.102853.

[4] X. Wang, “A Survey of Online Advertising Click-Through Rate Prediction Models,” *2020 IEEE International Conference on Information Technology,Big Data and Artificial Intelligence (ICIBA)*, Nov. 2020, doi: https://doi.org/10.1109/iciba50161.2020.9277337.

[5] Breiman, L. (2001). "Random forests." *Machine Learning*, 45(1), 5-32.

[6] Friedman, J. H. (2001). "Greedy function approximation: a gradient boosting machine." *The Annals of Statistics*, 29(5), 1189-1232.

[7] Liaw, A., & Wiener, M. (2002). "Classification and regression by randomForest." *R News*, 2(3), 18-22.

[8] Friedman, J. H. (2002). "Stochastic gradient boosting." *Computational Statistics & Data Analysis*, 38(4), 367-378.

[9] Chen, T., & Guestrin, C. (2016). "XGBoost: A scalable tree boosting system." *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785-794.

[10] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., ... & Liu, T. Y. (2017). "LightGBM: A highly efficient gradient boosting decision tree." *Advances in Neural Information Processing Systems*, 30, 3146-3154.

[11] He, X., Pan, J., Jin, O., Xu, T., Liu, B., Xu, T., ... & Shi, Y. (2014). "Practical lessons from predicting clicks on ads at Facebook." *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising*, 1-9.

[12] Rendle, S. (2010). “Factorization Machines.” *IEEE International Conference on Data Mining*.

[13] Juan, Y., Zhuang, Y., Chin, W.-S., & Lin, C.-J. (2016). “Field-aware Factorization Machines for CTR Prediction.” *Proceedings of the 10th ACM Conference on Recommender Systems*.

[14] Cheng, H.-T., Koc, L., Harmsen, J., Shaked, T., Chandra, T., Aradhye, H., ... & Anil, R. (2016). “Wide & Deep Learning for Recommender Systems.” *Proceedings of the 1st Workshop on Deep Learning for Recommender Systems*.

[15] Guo, H., Tang, R., Ye, Y., Li, Z., & He, X. (2017). “DeepFM: A Factorization-Machine based Neural Network for CTR Prediction.” *International Joint Conference on Artificial Intelligence*.

[16]“Deep Pattern Network for Click-Through Rate Prediction,” *Arxiv.org*, 2024. https://arxiv.org/html/2404.11456v1 (accessed May 01, 2025).

[17] Bottou, L. (2010). “Large-Scale Machine Learning with Stochastic Gradient Descent.” *Proceedings of COMPSTAT*.

[18] McMahan, H. B., Holt, G., Sculley, D., et al. (2013). “Ad Click Prediction: a View from the Trenches.” *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

[19] Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., & Bouchachia, A. (2014). “A Survey on Concept Drift Adaptation.” *ACM Computing Surveys*, 46(4), 44.

[20] Kolter, J. Z., & Maloof, M. A. (2007). “Dynamic Weighted Majority: A New Ensemble Method for Tracking Concept Drift.” *Journal of Machine Learning Research*, 8, 2755–2790.

[21] Li, L., Chu, W., Langford, J., & Schapire, R. E. (2010). “A Contextual-Bandit Approach to Personalized News Article Recommendation.” *Proceedings of the 19th International Conference on World Wide Web*.

191-15-12528

ORIGINALITY REPORT

12%	8%	6%	8%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	Submitted to Daffodil International University Student Paper	2%
2	dspace.daffodilvarsity.edu.bd:8080 Internet Source	2%
3	Submitted to Anatolia College Student Paper	<1%
4	Submitted to University of Northumbria at Newcastle Student Paper	<1%
5	incompleteideas.net Internet Source	<1%
6	Alberto Cano, Bartosz Krawczyk. "Kappa Updated Ensemble for drifting data stream mining", Machine Learning, 2019 Publication	<1%
7	H.L. Gururaj, Francesco Flammini, S. Srividhya, M.L. Chayadevi, Sheba Selvam. "Computer Science Engineering", CRC Press, 2024 Publication	<1%
8	dokumen.pub Internet Source	<1%
9	Omid Rafieian, Hema Yoganarasimhan. "AI and Personalization", Emerald, 2023 Publication	<1%
10	Adeel Ahmed, Khalid Saleem, Osman Khalid, Umer Rashid. "On deep neural network for trust aware cross domain recommendations	<1%

	in E-commerce", Expert Systems with Applications, 2021 Publication	
11	libweb.kpfu.ru Internet Source	<1 %
12	"Machine Learning, Optimization, and Big Data", Springer Nature, 2016 Publication	<1 %
13	Kamil, Khalidah Ahmad. "Layout Design for Mass Customisation Production: an Action Based Case Study in Small Enterprise", University of Malaya (Malaysia), 2023 Publication	<1 %
14	Gang Hua, Qing Xie, Yuhan Wang, Lin Li, Mengzi Tang, Yongjian Liu. "Dual-state feature importance perception and adaptive interaction importance modeling for CTR prediction", Expert Systems with Applications, 2026 Publication	<1 %
15	digitalmarketinggroup.in Internet Source	<1 %
16	Lei Tian, Lina Ge, Zhe Wang, Guifen Zhang, Chenyang Xu, Xia Qin. "Research on Improvement of the Click-Through Rate Prediction Model Based on Differential Privacy", IEEE Access, 2022 Publication	<1 %
17	Lee, Andrew C.. "A Predictive Model for Resource Allocation in Customs and Border Protection", The George Washington University Publication	<1 %
18	V. Sharmila, S. Kannadhasan, A. Rajiv Kannan, P. Sivakumar, V. Vennila. "Challenges in	<1 %

	Information, Communication and Computing Technology", CRC Press, 2024 Publication	
19	Yuan Zhang, Tiantian Sheng, Yaqin Fan, Lunping Duan. "Click-through rate prediction via feature selection and layer-wise residual gated cross-network", The Journal of Supercomputing, 2025 Publication	<1 %
20	www.researchgate.net Internet Source	<1 %
21	McDaniel, Claire. "Predicting Augmented Reality Virtual Try-On Adoption Using Machine Learning.", National University Publication	<1 %
22	Shreyas Patil, Kaustubh Raut, Prachi Palsodkar, Taranjyot Singh, Yogita Dubey, Roshan Umate. "Click Prediction Learning for Effective Advertising", 2022 International Conference on Emerging Trends in Engineering and Medical Sciences (ICETEMS), 2022 Publication	<1 %
23	Submitted to Berlin School of Business and Innovation Student Paper	<1 %
24	orbilu.uni.lu Internet Source	<1 %
25	Jyoti Bawa, Kuljit Kaur Chahal, Kamaljit Kaur. "Improving cloud resource management: an ensemble learning approach for workload prediction", The Journal of Supercomputing, 2025 Publication	<1 %
26	docs.neu.edu.tr	

	Internet Source	<1%
27	Aurelian-Dumitrache Anghele, Virginia Marina, Liliana Dragomir, Cosmina Alina Moscu, Mihaela Anghele, Catalin Anghel. "Predicting Deep Venous Thrombosis Using Artificial Intelligence: A Clinical Data Approach", Bioengineering, 2024 Publication	<1%
28	Dinesh Goyal, Bhanu Pratap, Sandeep Gupta, Saurabh Raj, Rekha Rani Agrawal, Indra Kishor. "Recent Advances in Sciences, Engineering, Information Technology & Management - Proceedings of the 6th International Conference “Convergence2024” Recent Advances in Sciences, Engineering, Information Technology & Management, April 24–25, 2024, Jaipur, India", CRC Press, 2025 Publication	<1%
29	Ajay Kumar, Sangeeta Rani, Krishna Dev Kumar, Manish Jain. "Handbook of AI in Engineering Applications - Tools, Techniques, and Algorithms", CRC Press, 2025 Publication	<1%
30	ebin.pub Internet Source	<1%
31	www.numberanalytics.com Internet Source	<1%
32	"Emerging Trends and Applications in Artificial Intelligence", Springer Science and Business Media LLC, 2024 Publication	<1%
33	Nasser, Hind. "AI-Driven Prediction of Flight Cancellations: A Machine Learning Approach	<1%

	in Minimizing Airline Disruption.", Rochester Institute of Technology Publication	
34	Submitted to UCL Student Paper	<1%
35	Submitted to BB9.1 PROD Student Paper	<1%
36	Submitted to IUBH - Internationale Hochschule Bad Honnef-Bonn Student Paper	<1%
37	Submitted to Indiana University Student Paper	<1%
38	Lavado, Susana Margarida Silva Ferreira. "Identifying Clients' Bad Experiences with their Internet Service", Universidade NOVA de Lisboa (Portugal), 2024 Publication	<1%
39	www.crhc.illinois.edu Internet Source	<1%
40	Submitted to Higher Education Commission Pakistan Student Paper	<1%
41	Submitted to INTI International University Student Paper	<1%
42	Submitted to Coventry University Student Paper	<1%
43	dspace.vutbr.cz Internet Source	<1%
44	www.programmingempire.com Internet Source	<1%
45	Durgesh Kumar Mishra, Nilanjan Dey, Bharat Singh Deora, Amit Joshi. "ICT for Competitive Strategies", CRC Press, 2020	<1%

	Publication	
46	Kwaku K Quansah, Sean A Murphy, Esther Kwon, Emma Anderson et al. "Machine Learning Models Enhance Prediction of Arrhythmogenic Right Ventricular Cardiomyopathy", Cold Spring Harbor Laboratory, 2025 Publication	<1 %
47	Submitted to Queen Mary and Westfield College Student Paper	<1 %
48	peerj.com Internet Source	<1 %
49	"Database Systems for Advanced Applications", Springer Nature, 2018 Publication	<1 %
50	Javad Hassannataj Joloudari, Mohammad Maftoun, Mohammad Ali Nematollahi, Kandala N V P S Rajesh et al. "Predicting Diabetes Mellitus using Conditional Tabular Generative Adversarial Networks combined with MLP based on Body Composition Data", Springer Science and Business Media LLC, 2025 Publication	<1 %
51	Kristof Coussement, Koen W. De Bock, Scott A. Neslin. "Advanced Database Marketing - Innovative Methodologies and Applications for Managing Customer Relationships", Routledge, 2016 Publication	<1 %
52	Ling Zhu, C. Guedes Soares. "Trends in Collision and Grounding of Ships and Offshore Structures", CRC Press, 2025 Publication	<1 %

53	M., Arya Lekshmi. "Modelling, Fabrication, and Characterization of Memristor for Mitigating the Sneak Path Current in a Nano-Crossbar Memory Array", University of Nottingham (United Kingdom) Publication	<1%
54	Pushpa Choudhary, Sambit Satpathy, Arvind Dagur, Dhirendra Kumar Shukla. "Recent Trends in Intelligent Computing and Communication", CRC Press, 2025 Publication	<1%
55	link.springer.com Internet Source	<1%
56	regional.atacenter.org Internet Source	<1%
57	simulation.su Internet Source	<1%
58	thatware.co Internet Source	<1%
59	www.arxiv-vanity.com Internet Source	<1%
60	www.frontiersin.org Internet Source	<1%
61	www.mdpi.com Internet Source	<1%
62	www.mhwm.pl Internet Source	<1%
63	Chen Jie-Hao, Li Xue-Yi, Zhao Zi-Qian, Shi Ji-Yun, Zhang Qiu-Hong. "A CTR prediction method based on feature engineering and online learning", 2017 17th International Symposium on Communications and Information Technologies (ISCIT), 2017 Publication	<1%

64	Jacinta Ann Jacob, S. Gnanavel. "Mobile advertisements click through rate prediction using machine learning", AIP Publishing, 2024 Publication	<1 %
65	Niu, Zhiqi. "Sentiment Driven Deep Learning Approach in Stock Market Prediction", The University of Mississippi Publication	<1 %
66	"Machine Learning and Knowledge Discovery in Databases", Springer Science and Business Media LLC, 2017 Publication	<1 %
67	Bora Edizel, Amin Mantrach, Xiao Bai. "Deep Character-Level Click-Through Rate Prediction for Sponsored Search", Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval - SIGIR '17, 2017 Publication	<1 %
68	Liu, Xiaolong. "Fundamental Component Modeling Within Recommender Systems", University of Illinois at Chicago Publication	<1 %
69	Tian, Yuan. "Inferring user Needs & Tasks from App Usage Interactions", University of Nottingham (United Kingdom) Publication	<1 %
70	Xianshan Qu, Li Li, Xi Liu, Rui Chen, Yong Ge, Soo-Hyun Choi. "A Dynamic Neural Network Model for Click-Through Rate Prediction in Real-Time Bidding", 2019 IEEE International Conference on Big Data (Big Data), 2019 Publication	<1 %

22% detected as AI

The percentage indicates the combined amount of likely AI-generated text as well as likely AI-paraphrased.

Caution: Review required.

It is essential to understand the limitations of AI detection before making decisions about a student's work. We encourage you to learn more about Turnitin's AI detection capabilities before using the tool.

Detection Groups

30 AI-generated only 15%
Likely AI-generated text from a large-language model.

13 AI-generated text that was AI-paraphrased 7%
Likely AI-generated text that was likely revised using an AI-paraphrase tool or word spinner.

Disclaimer
Our AI writing assessment is designed to help educators identify text that might be prepared by a generative AI tool. Our AI writing assessment may not always be accurate (i.e., our AI models may produce either false positive results or false negative results), so it should not be used as the sole basis for adverse actions against a student. It takes further scrutiny and human judgment in conjunction with an organization's application of its specific academic policies to determine whether any academic misconduct has occurred.

Frequently Asked Questions

How should I interpret Turnitin's AI writing percentage and false positives?
The percentage shown in the AI writing report is the amount of qualifying text within the submission that Turnitin's AI writing detection model determines was either likely AI-generated text from a large-language model or likely AI-generated text that was likely revised using an AI paraphrase tool or word spinner.

False positives (incorrectly flagging human-written text as AI-generated) are a possibility in AI models.

AI detection scores under 20%, which we do not surface in new reports, have a higher likelihood of false positives. To reduce the likelihood of misinterpretation, no score or highlights are attributed and are indicated with an asterisk in the report (*%).

The AI writing percentage should not be the sole basis to determine whether misconduct has occurred. The reviewer/instructor should use the percentage as a means to start a formative conversation with their student and/or use it to examine the submitted assignment in accordance with their school's policies.

What does 'qualifying text' mean?
Our model only processes qualifying text in the form of long-form writing. Long-form writing means individual sentences contained in paragraphs that make up a longer piece of written work, such as an essay, a dissertation, or an article, etc. Qualifying text that has been determined to be likely AI-generated will be highlighted in cyan in the submission, and likely AI-generated and then likely AI-paraphrased will be highlighted purple.

Non-qualifying text, such as bullet points, annotated bibliographies, etc., will not be processed and can create disparity between the submission highlights and the percentage shown.