Predicting Advertisement Click-Through Rate

Maria Haralampopoulos
Department of Computer and Information Science
Fordham University
New York, NY, USA
mharalampopoulos2@fordham.edu

Abstract*—Click-Through Rate (abbreviated CTR) is a measure of the percentage of users that see a promoted message or advertisement and interact with it, mainly by clicking on it. This measure is a very crucial task within marketing, especially online advertising as it informs decisions about placement and promotion of advertisement campaigns as well as which users or user-bases to aim these campaigns towards. Despite the importance of this statistic, predicting user CTR is particularly challenging due to noise in user behavior data, weak prediction signal, and an extremely strong class imbalance, due to the fact that most users will scroll past an advertisement instead of clicking on it. In this project, we will evaluate the application of varying machine learning models in predicting CTR using a large-scale dataset containing categorical and numerical features. We will compare two logistic regression models (L2 and L1 regularization respectively) and an MLP neural network, all using the same preprocessed dataset and with two different manners of re-weighting applied to account for class imbalances within the data. In this case, we explored the efficacy of applying Synthetic Minority Oversampling Technique (SMOTE) to our dataset compared to applying traditional class re-weighting. All model performance is assessed and analyzed through metrics such as ROC AUC, Log Loss and confusion matrices. Our results found that all models analyzed performed nearly negligibly better than random, and that increased complexity of the model did not have a substantial impact on ranking performance, which was expected and consistent with the findings of prior CTR studies. Regarding re-weighting, we found that SMOTE improved recall, but introduced tradeoffs in calibration and false positive rates. The findings provided in this paper highlight the difficulties raised by noisy and severely imbalanced data when predicting CTR in a practical setting.

I. Introduction

Since the boom of Internet usage throughout the 2000s and onward, online advertising has become a major player in reaching large numbers of potential customers, regardless of where they may be. Where in the past billboards and print advertisements like magazine spreads would only reach smaller-spanning groups, meaning companies would have to relegate their advertisements to a specific geographic area or magazine reader-base, online advertising campaigns can now target individuals based on their Internet presence and search activity regardless of their location. For example, in the past an ad about Singer sewing machines would probably appear in a women's fashion or crafts magazine, targeting those that have demonstrated a specific interest in the hobby to boot, however a single father looking to sew his child's Halloween costume would not be reached by this campaign despite being a prospective buyer. Nowadays though, if he made a simple Google search about DIY Halloween costume tutorials, his search behavior could trigger the advertisement to be shown to him, thus reaching another prospective buyer and facilitating another sale. It is important to note that placing advertisements online is a very expensive venture, meaning that companies must ensure the placement of their advertising campaigns is effective and gets them their money's worth. As a result, companies use varying metrics to evaluate the success of a campaign, one of which being click-through rate (CTR). Defined as the proportion or percent of advertisement impressions that result in users clicking said advertisement, an accurate prediction of user CTR can aid in determining and optimizing not only where to place an advertisement online, but also which users to show it to. However, despite its importance, proper working and accurate predictions of user CTR are notably hard to achieve, making it an extremely challenging machine learning problem. User activity, which serves as a key predictor for interactions with advertisements, is influenced by a number of unobserved or unobservable factors, including time of day, external context of website or search use, and even if the user is utilizing an adblocker or VPN. As a result of these unobserved factors, the features we are able to observe and use as predictors usually have a notably weak predictive power. Another key issue faced within CTR datasets is the occurrence of severe class imbalance, as most individuals tend to close out of, scroll past, or skip an advertisement instead of clicking. Consequently, the imbalance heavily skews datasets and results in problems consisting of model training and evaluation, as naive classification models will provide inaccurate and deceptively high accuracy rates due to always predicting in favor of the majority.

Early predictive models for CTR utilized and relied on linear models like logistic regression with typically extensive feature engineering, whereas more recent studies have looked into nonlinear models like neural networks and deep learning models that can examine more complex relationships and interactions between studied features. It is important to note, however, that increased model complexity may not exactly lead to improved performance and accuracy of predictions, especially if feature signal remains limited.

This project aims to examine user CTR prediction from an applied and difficulty-oriented perspective, emphasizing model comparison, evaluation strategy and handling of class imbalance. Rather than trying to achieve objectively correct results, this project will provide insight into challenges faced

when predicting CTR, using analysis of results from our models as well as our methods process to provide a practical examination of CTR model performance.

The rest of this paper is organized as follows. Section 2 serves as a literature review and describes previous studies on CTR prediction and their effects on our methods. Section 3 provides a description of our utilized dataset containing features, imbalance and preprocessing decisions. Section 4 gives a discussion of preprocessing, model definitions and weight applications, as well as setup of our tests. Section 5 provides an analysis of our findings and a discussion of their significance. Section 6 gives study limitations and preliminary ideas for future research. Section 7 concludes this study.

II. Background

To ideally and effectively place an advertisement somewhere online, it is crucial to be able to accurately predict CTR (probability of the user clicking) for the advertisement in question as CTR estimates hold a direct influence on advertisement ranking, placement and pricing in varying sponsored advertising platforms such as search engine results [1], [2]. CTR is typically measured as a binary classification (no click = 0, click = 1) or estimation task ratio (general probability or percent that a user will click on advertisement), with the rate estimated given a set of observed features describing user and advertisement qualities. Whenever an advertisement is displayed on a search page, it has a chance of being viewed by the user, therefore making the probability of a click depending on two factors, those being the probability that it is viewed and the probability that it is clicked given the advertisement was viewed [1]. In these cases, more linear statistical models are used, particularly logistic regression as it is ideally suited for probabilities, as well as able to be easily interpreted and scaled [1]. These regularized logistic regressions can provide reasonable initial CTR estimates when combined with proper feature selection, even if advertisements are new (and thus having little to no historical click data) [1].

Later studies also note and suggest that despite deeper models having been developed and utilized for the same predictions, linear models seem to hold more competence, especially in cases where predictive power is weaker and the provided dataset is extremely imbalanced [2].

If the size and dimensions of a CTR dataset are larger, manual feature selection becomes a hassle and it makes more sense to begin automatically capturing feature relationships. As a result, more complex models such as factorization machines have been utilized, particularly in modeling lower-order interactions to reduce or outright replace manual feature selection [2]. Regardless of this push forward, however, these models have provided negligible differences in their expressiveness, as quality of analysis remains heavily dependent on the quality of features selected and used.

As of late, there has remained an interest in use of deep learning models in predicting user CTR particularly to model feature relationships more explicitly, with those models ranging from multilayer perceptrons (MLPs) to more advanced and sophisticated models like cross networks [3]. However, results of these studies seem to report that despite the complexity of a model, performance increases, particularly ROC AUC and log loss, remain negligible, if they exist at all [2], [3]. In light of these findings, we can posit that complexity of a model is not strong enough to overcome any limitations that are present as a result of the weak and noisy nature of CTR dataset features.

An extremely prevalent and previously iterated problem in CTR data collection and analysis is that of severe class imbalance, resulting from the fact that most advertisement views do not exactly translate to clicks on those advertisements at a 1:1 ratio [1], [2], [3]. As a result of this imbalance, any naive analysis and prediction models will provide a high accuracy level due to the models only predicting the majority "no-click" category and giving a large number of false negatives. To combat these problems, there are a few options present, such as applying instance re-weighting and data resampling to the training dataset prior to model construction and usage. Re-weighting can be used in an event where the overall data distribution requires preservation, whereas resampling methods like SMOTE can be used to improve overall recall for minority classes. However, findings from prior research raise issues with oversampling, noting calibration issues and higher rates of false positives [2], [3].

These existing results suggest overall that performance of CTR prediction models are often more heavily constricted by the quality of data characteristics compared to the quality of the models used. It is important to note that despite these previous findings and warnings, we ignored them anyway and opted to reinvent the wheel and present a comparative and applied view of CTR prediction models.

III. Data Description

We used a dataset taken from Kaggle, providing information on the online advertising data of an unnamed company, with the training and test datasets consisting of 463,291 observations and 15 variables and 128,858 observations and 14 variables respectively. Pre-data cleaning, each observation from the training set contains the following information:

- **session_id: The unique ID number of a user's online session.**
- **DateTime: The date and time stamp of the logged online session.**
- **user_id:** The unique user ID number, it does not differ by session if the same user is logged.
- **product:** The product being advertised, each individual product being represented by a certain letter.
- **campaign_id:** The unique ID number for each advertising campaign.
- **webpage_id:** The unique ID number for each webpage the advertisement was found on.
- **product_category_1:** The category of each product, categorized from 1 to 5.
- **product_category_2:** A secondary categorizer value for products.

- **user_group_id:** User ID groupings for broader categorization.
- **gender:** User gender for broader categorization.
- **age_level:** User age ranges, separated by categories 0-6 for broader categorization.
- **user_depth:** Depth of user interaction on the platform where advertisement was found (activity levels of the user).
- **city_development_index:** Unknown, presumed to hold some sort of geographical indicator reference.
- **var_1:** An unknown binary variable.
- **is_click:** Our target variable, lists whether or not the user in the current instance clicked on the advertisement (0 signifies no click, 1 signifies click).

The test dataset is nearly identical to the training dataset, only differing by omitting is_click from the variables.

A. Initial Cleaning and Feature Removal

Exploratory analysis (top 10 missingness ranking) found extremely high missingness values for a couple of variables, namely product_category_2 and city_development_index, which sat at about 79% and 27% of their values missing, respectively. As a result of the high percentage of missing values and resulting limited interpretability of those variables, both features were dropped from the dataset. After dropping those features, we then dropped any remaining observations with missing values from our datasets in order to have a dataset with exclusively complete observations.

TABLE I
FEATURES WITH MISSINGNESS SCORES

Feature Name	*Missingness Score*
product_category_2	0.789685
city_development_index	0.270087
gender	0.039377
user_group_id	0.039377
age_level	0.039377
user_depth	0.039377

B. Feature Preparation

Following initial dataset cleaning, we separated features into numerical and categorical groups based on their data types. Numerical features included continuous or ordinal variables including "age_level, user_depth and user_id," whereas categorical features included discrete and nominal fields including "product" and "gender". In total, our cleaned dataset now contained nine numerical and three categorical features.

We converted our target "is_click" to a binary integer to ensure consistency during model training, and also accounted for any potential string representations (such as "yes/no, true/false") by mapping them to their corresponding integer values. Following this step, our target variable was split from the feature matrix prior to splitting our train and validation sets.

C. Train-Validation Split

We split our training dataset into an 80/20 train-validation split to use for our original models for purposes of maintaining insights such as ROC AUC and log loss following model application. Despite having a separate testing dataset, the test dataset was reserved exclusively for final model predictions and not used during preprocessing, initial model training and performance evaluations due to a lack of the target variable "is_click" making it unsuitable for some comparisons. All preprocessing steps that were applied to our training data were subsequently applied to the test set to generate predicted click probabilities for the dataset once our modeling comparisons were complete.

D. Transformation Pipelines

A column-wise pipeline structure for both numerical and categorical features was implemented, with numerical features being processed through median imputation and standardization to zero mean, and categorical features being processed using frequent-value imputation and one-hot encoding.

E. Class Imbalance

TABLE II
CLASS BALANCE

is_click	Proportion
0	0.932373
1	0.067627

Due to the severely imbalanced nature of our data, we applied class weighting directly within the models using balanced class weights from the training data and followed up with application of SMOTE to the training set in order to create synthetic minority samples and result in a balanced distribution before training.

TABLE III
BALANCE WITH SMOTE APPLICATION

SMOTE Applied?	Negative Hits	Positive Hits
No	331,992	24,046
Yes	331,992	331,992

IV. MODELS

We applied three supervised learning models to our data in order of increasing model complexity: L2-regularized linear regression, L1-regularized linear regression, and a multilayer perceptron (MLP) neural network. All models utilized the same preprocessed features previously outlined, and were evaluated on a common validation set. Additionally of note, a support vector machine (SVM) model was also attempted on this dataset, but was removed due to the model's extensive runtime.

A. L2-Regularized Linear Regression

This model was selected as our standard baseline for our user CTR predictions due to its interpretability and probabilistic output, as well as for its aid in stabilizing optimization and reducing overfitting by penalizing large coefficients. Our model was trained with a maximum iteration count of 500 and performed on our resampled training dataset, whereas the model performance was then evaluated on our validation set using predicted click probabilities.

B. L1-Regularized Linear Regression

This model was selected in order to examine effects of sparsity-creating penalties in a high-dimensional feature space, as the model encourages coefficients to shrink to zero, effectively serving as an implicit form of feature selection. Due to the higher computational cost of L1 regression, the model was trained with a maximum iteration count of 200, and, similarly to the L2 model, was performed on our resampled training set.

C. MLP Neural Network

We implemented this model to serve as our choice of non-linear approach with a more advanced structure. We applied two hidden layers with 128 and 64 units respectively as well as employed the ReLU activation function, and the model was tested on a stochastic gradient descent with an adaptive learning rate and initial learning rate of 0.001. Early stopping was applied to avoid overfitting and unnecessary computation, and mini-batch training with batch sizes defined as smaller fractions of the training set size were used. As with all other models, our MLP neural network was trained on our resampled training data and evaluated on our untouched validation set.

D. Predictions Utilizing Test Dataset

Following model training and validation, we applied our final trained models to our unused test dataset to create predicted CTR probabilities. We applied the same preprocessing pipelines that we fit onto our training set to our test features, and each model created a probability estimate for the positive hit class (click class).

V. Results

Model performance was evaluated on our validation set using various ranking-based and threshold-dependent metrics to examine the effects of severe class imbalance on these models' abilities to predict user CTR. More specifically, we examine ROC AUC, log loss, accuracy, precision, recall and F1 scores in order to discuss and quantify our findings.

A. AUC and Log Loss by Model

TABLE IV
ROC AUC and Log Loss

Model Type	ROC AUC	Log Loss
L2	0.53091185499508	0.6647553713215875
L1	0.5313993406494452	0.6673517226513335
MLP	0.5281974231874311	1.112887403030634

Both logistic regression models ended up with nearly identical ROC AUCs with both of them standing at around 0.531, while our MLP neural network ended up with an ROC AUC around 0.528, performing worse than our linear regressions even if by a borderline negligible amount. All models' AUCs exceeded that of the random selection (value of 0.5), but the tiny improvement size seems to point towards limited ranking separability between clicks and non-clicks.

Examining log loss, we found that our linear regressions led the pack, with L2 regularization beating L1 for the top spot by 0.002, standing at around 0.665 and 0.665 respectively. Meanwhile, our MLP reported a notably higher log loss of 1.112, a nearly 0.448-unit gap between best and worst log loss, which is suggestive of worse calibration of predicted probabilities despite very similar AUC values. Therefore, it is safe to conclude that while MLP ranking capabilities were on par with linear models, the resulting probability values were inconsistent with observed outcomes.

B. ROC Graph

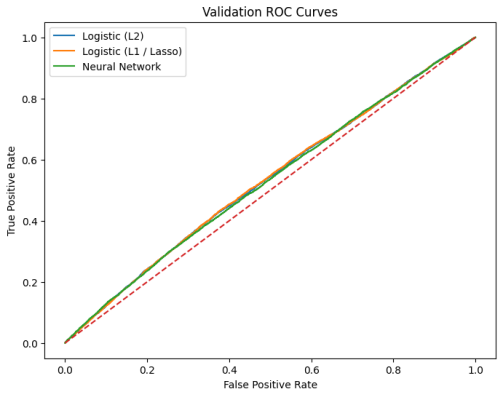

Fig. 1. Validation ROC Graph, SMOTE applied.

All three curves are only slightly above the dotted diagonal indicator for random performance (previously stated 0.5 value) and overlap with each other to a very notable extent, suggesting that ranking performance was not particularly affected by model choice. Regardless of whether or not SMOTE was applied, the graphs remained extremely similar, suggesting that at least regarding ROC AUC, the effects of resampling are quite limited and unremarkable.

C. Confusion Matrices

We made a point to include a set of pre-SMOTE confusion matrices in order to show the effects of horribly imbalanced data on prediction models, as these confusion matrices show exactly what was detailed in the introduction- every prediction was listed as negative, indicating model bias towards the larger set of data (the 93% of data that was listed as no-clicks).

Following SMOTE application, both logistic regressions showed similar behavior, with accuracies for both standing at approximately 0.586. Not the best, but still decent and demonstrating moderate overall model correctness likely due

to correct classification of the majority non-click class values. Looking at precision values, they topped out at the extremely low value of about 0.075, suggesting a large number of false positives for predicted clicks. Recall values on the other hand were relatively very high (standing at around 0.45), indicating that almost half of all the true positives were correctly identified.

Our MLP neural network behaved notably differently, with overall accuracy being about 0.3 units higher at approximately 0.89, but this could be mainly attributed to a near-exclusive rate of predicting non-click values. Precision for the click class was higher than that of our linear models by about 0.1 but recall had a sharp drop to a value of 0.66 which led to the MLP model having the lowest F1 score of all three models and indicates that MLP must have adopted some sort of decision strategy that minimized false positives but traded off with missing true positives.

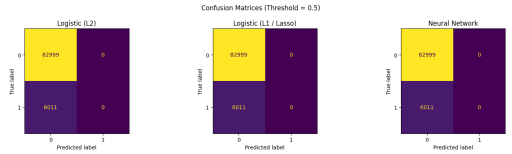

Fig. 2. Confusion Matrices (No SMOTE applied).

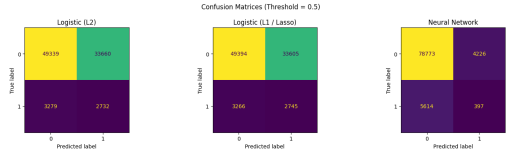

Fig. 3. Confusion Matrices (SMOTE applied).

Discussion

Across all three models tested, prediction performance remained barely higher than random selection despite application of preprocessing strategies, handling of imbalanced data and use of both linear and nonlinear models. These findings remain consistent with previous user CTR studies, indicating that the inherently weak feature signal and noisy, imbalanced user behavior present in CTR data limit overall achievable performance for those models. Increasing model complexity from a logistic regression to a neural network did little to nothing to improve performance, with our MLP failing to outperform the two logistic regression models in terms of AUC, and showed notably worse probability calibration rates as demonstrated by its higher log loss level, suggesting that despite the neural network having higher flexibility, it did not lead to any significant improvements. In a similar vein, the practically identical performance stats of our logistic regression models suggest that L1 regularization did very little in the way of improving predictions and indicating that in the case of user CTR, predictive signal is not concentrated in a small set of features and instead manifests as something more diffuse.

Looking at class imbalances, SMOTE application improved recall for the minority "click" class within our logistic regression models but in tandem reduced precision, which culminated in a higher number of false positives. Inversely, the MLP neural network's approach of predominantly selecting to predict the majority "no click" class increased its accuracy but culminated in very few true positives being selected.

Overall, our findings seem to indicate that model selection is not a driving force in CTR predictions, but it is instead the characteristics of the CTR data being analyzed, as without more detailed signals, data and flags, even the most detailed models can only perform so well.

Conclusion

To conclude, in this study we compared varying linear and nonlinear predictive models on applied CTR data with consistent preprocessing and evaluation procedures in an attempt to see which factors or models worked the best for our dataset. Through these comparisons, our study showed that increased complexity of a model does not necessarily equate to improvements in predictive performance across CTR analysis. Our findings came as a result of the heavy hand of the inherent data limitations that plague CTR data, particularly weak feature signal and strong class imbalance. Despite all we did to even out these limitations (SMOTE application, etc.), we still saw drawbacks in terms of model precision and calibration, indicating weakness in our dataset and metric selections.

Looking to the future, we would like to see more of these analyses done on more advanced and detailed data, as the dataset provided for this study was weaker than expected, leading to all analyses not providing particularly good results. Looking at this study from a more applied point of view, this work serves as an example of the limitations of standard model approaches and analyses in real-world settings.

References

[1] M. Richardson, E. Dominowska, and R. Ragno, "Predicting clicks: Estimating the click-through rate for new ads," in *Proceedings of the 16th International World Wide Web Conference (WWW)*, Banff, AB, Canada, 2007, pp. 521–530

[2] Y. Yang and X. Zhai, "Click-through rate prediction in online advertising: A literature review," *Information Processing & Management*, vol. 59, no. 1, Art. no. 102853, Jan. 2022.

[3] W. Zhang, T. Du, and J. Wang, "Deep learning for click-through rate estimation," *arXiv preprint* arXiv:2104.10584, 2021.

[4] Kaggle, "CTR in Advertisement dataset," 2025. [Online]. Available: https://www.kaggle.com/datasets/arashnic/ctr-in-advertisement/data?select=Ad_click_prediction_train+