Predicting Website Ad Clicks

Final Project Report

Group 16

Amitesh Tripathi Sayali Lad

857-376-1991(Tel. Student 1)

857-277-4326(Tel. Student 2)

[tripathi.am@northeastern.edu](mailto:tripathi.am@northeastern.edu) [lad.sa@northeastern.edu](mailto:lad.sa@northeastern.edu)

Percentage of Effort Contributed by Student 1: 50 % Percentage of Effort Contributed by Student 2: 50 %

Signature of Student 1: Amitesh Tripathi Signature of Student 2: Sayali Lad

Submission Date: 04/21/2023

**Table of Contents**

Problem Setting…………………………………………………………………………. 3

Problem Definition……………………………………………………………………… 3

Data sources & Data Description………………………………………………………. 3

1. Data Collection…………………………………………………………………... 4
2. Data Processing………………………………………………………………….. 4
3. Data Exploration…………………………………………………………………. 4

4. Data Visualization……………………………………………………………….… 5

4.1 Bar Chart………………………………………………………………….…...5

4.2 Heatmap……………………………………………………………………….6

4.3 Scatter Plot Matrix………………………………………………………….... 7

4.4 Grouped Bar Chart……………………………………………………………8

4.5 Line Chart……………………………………………………………………...9

Data Mining Model Selection……………………………………………………………10

Evaluation…………………………………………………………………………………10

Implementation of Selected Model………………………………………………………13

Data preparation and exploration………………………………………………………..13

Model building and Selection…………………………………………………………….13

Model Interpretation……………………………………………………………………...16

Project Results…………………………………………………………………………….17

Key Findings………………………………………………………………………………17

Impact of Project Outcomes ……………………………………………………………..18

**Problem Setting:**

An e-commerce website is looking to improve sales through targeted advertisements on partner websites. The website has hired an Adtech company to build a system for displaying ads for products that customers have previously viewed or similar items. The goal is to predict the probability of a user clicking on an ad based on their viewing history and user data.

**Problem Definition:**

The task is to predict the likelihood that a user clicks on a product ad on a partner website. This will be done by analyzing the user's view log, ad impression, and user data to determine the probability of a click in the next 7 days.

**Data Sources:**

The data for this project comes from the e-commerce website and includes view log data from October 15, 2018, to December 11, 2018, product description data, and ad impression data from November 15, 2018, to December 18, 2018. The data includes train and test sets, with the train set containing information on ad impressions and whether or not the ad was clicked. The test set contains ad impression information without labels.

[https://www.kaggle.com/datasets/arashnic/ctrtest?resource=download2019/#ProblemStateme](https://datahack.analyticsvidhya.com/contest/wns-analytics-wizard-2019/#ProblemStatement) [nt](https://datahack.analyticsvidhya.com/contest/wns-analytics-wizard-2019/#ProblemStatement)

<https://www.kaggle.com/datasets/jahnveenarang/cvdcvd-vd> <https://github.com/splikhita/Ad-Click-Prediction/blob/master/Advertisements-Data.csv>

**Data Description:**

The column names, obtained from multiple datasets, play a crucial role in the consolidation process as they will be utilized to merge the datasets into a single, unified dataset for further data exploration and analysis. The chosen columns will be carefully selected based on their relevance to the goals of the analysis, ensuring the quality and consistency of the final dataset.

Following is the **column name** and the description of it.

|  |  |
| --- | --- |
| Column Name | Description |
| Session\_id | Unique identifier for the individual's session |
| DateTime | The date and time when the user interacted with the website/app |
| User\_id | Unique identifier for the individual |
| product | The product that the user interacted with during the session |
| Campaign\_id | A unique identifier for the marketing campaign that brought the user to the website/app |
| Webpage\_id | A unique identifier for the webpage that the user interacted with during the session |
| Product\_category\_1 | The broad category that the product belongs to |
| Product\_category\_2 | A more specific category that the product belongs to |
| User\_group\_id | A unique identifier for the user group that the user belongs to |
| Gender | Gender of the individual |
| Age\_level | The age range of the user |
| User\_depth | A measure of the user's engagement with the website/app |
| City\_development\_index | A measure of the level of development in the city where the user is located |
| Var\_1 | An additional variable that may be relevant to predicting click-through rates |
| Is\_click | Indicates whether the individual clicked on the ad (1 for yes, 0 for no) |

## Data Collection:

Data from various sources, including Kaggle and the UCI Machine Learning Repository, was merged and uploaded to Google Collaboratory for analysis. The data was preprocessed using the 'read\_csv()' and 'read\_excel()' methods before being used to gain insights into user behavior and ad clicks. This data can be used to train a machine learning model to predict ad clicks.

## Data Processing:

The data frame had 4,63,291 rows and 15 columns. The data processing for website ad click prediction involved several steps. Firstly, the dataset was loaded and its shape and summary were calculated. Next, a copy of the dataset was created and all categorical variables were transformed into numerical variables using the labelencoder() function. The most insignificant columns were then dropped, and missing values were filled using mathematical operations. Outliers were checked using a statistical function, and finally, the describe() method was used to obtain a summary of the dataset's statistics. These steps were taken to prepare the dataset for the machine learning model used to predict website ad clicks.

## Data Exploration:

We analyzed the preprocessed data to gain insights that could aid in developing a more accurate ad click prediction model. Some key questions we addressed were:

Do click-through rates vary by gender? We categorized the data by gender and found that the CTR was slightly higher for men (9.12%) than for women (8.23%). We used an independent sample t-test and found that the difference was statistically significant (p<0.05).

Are there any seasonal variations in CTR? We analyzed the data by hour of the day and found that CTR increased slightly in the early morning hours, peaking between 4:00 and 5:00 am. We also observed a decrease in CTR during the late afternoon and evening. However, there were no significant differences in CTR between males and females at different times of the day.

In addition to these questions, we conducted exploratory data analysis to understand the data distribution. We used histograms and boxplots to depict the distribution of continuous variables like age, city development index, and product categories. We found that the city development index had a right-skewed distribution, while age had a normal distribution. The product categories were not uniformly distributed, with some categories being more prevalent than others.

We also created visuals like histograms, scatter plots, and heatmaps to identify any patterns or outliers in the data.

## Data Visualization:

* 1. Bar Chart:

A bar chart was used to show the click-to-non-click ratio for each campaign, with distinct campaign IDs on the x-axis and the number of clicks and non-clicks on the y-axis. Color coding was used to differentiate between clicks and non-clicks. This chart can help distinguish between campaigns with higher or lower click-through rates. A high proportion of clicks to non-clicks may indicate that an advertisement is performing well, while a higher ratio of non-clicks to clicks may suggest a need for changes or alternative strategies.

Chart, bar chart

Description automatically generated

## Bar Chart

* 1. Heatmap:

A heatmap was used to display the correlation between variables in the dataset. The squares' colors indicate the strength of the correlation, with darker colors indicating stronger positive correlations and lighter colors indicating weaker positive correlations or negative correlations. The heatmap revealed that "user\_depth" and "is\_click" had a strong positive correlation, while "product\_category\_1" and "product\_category\_2" had a modest positive correlation.

Chart

Description automatically generated with medium confidence

## Heatmap

* 1. Scatter Plot Matrix:

A scatter plot matrix was used to show how different variable pairings in the dataset relate to each other. Each point on the matrix represents a single observation for two distinct variables. The scatter plot matrix can be used to identify trends or patterns in the data. For example, it revealed a marginally positive relationship between "city\_development\_index" and "is\_click," but no discernible association between "age\_level" and "is\_click."

Table

Description automatically generated

## Scatter Plot Matrix

* 1. Grouped Bar Chart:

A grouped bar chart was used to display the distribution of clicks and non-clicks by gender. The bars are color-coded for gender differentiation, and the height of the bars indicates the number of clicks and non-clicks. The graph shows the gender combination with the highest clicks and non-clicks where ‘0’ is the ‘not\_clicked’ and ‘1’ is ‘is\_click’.

Chart, bar chart

Description automatically generated

## Grouped Bar Chart

* 1. Line Chart:

A line chart was used to display the trend of clicks and non-clicks over time, with the x-axis showing date and time, and the y-axis showing the number of clicks and non-clicks. The chart helps identify recurring trends and patterns, such as the higher click-through rate in the morning and early afternoon compared to the evening.

Chart, line chart

Description automatically generated

## Line Chart

**Data mining Model Selection:**

We evaluated several data mining models for predicting the click-through rate. These models include:

Logistic Regression

K-Nearest Neighbors (KNN) Decision Tree

Random Forest Bagging

Stochastic Gradient Descent (SGD) Gradient Boosting

Extreme Gradient Boosting (XGBoost)

### Evaluation:

We conducted exploratory data analysis (EDA) and customer analytics to gain insights into the data and identify patterns and trends that could help predict the click-through rate. We also built a base model to establish a benchmark for model performance.

We evaluated each of the data mining models based on their accuracy, interpretability, scalability, robustness, and speed. After thorough evaluation, we found that Random Forest and XGBoost were the most suitable models for predicting the click-through rate.

Text

Description automatically generated

We also found that the Voting Classifier 1 combining Decision Tree, Random Forest, and Extreme Gradient Boosting classifiers displayed greater accuracy than the others .

Chart, bar chart

Description automatically generated

* Decision Tree Classifier: A flowchart-like model of decisions used to predict the class of input.
* Random Forest Classifier: Combines multiple decision trees to improve performance and reduce overfitting.
* Extreme Gradient Boosting Classifier: Ensemble algorithm that combines weak models to create a strong model. Known for its speed and performance.

By combining these classifiers into a Voting Classifier 1, we could take advantage of their individual strengths and achieve better performance than using them individually. The Voting Classifier 1 used majority voting to make predictions based on the predictions of its individual classifiers.

A picture containing graphical user interface

Description automatically generated

After identifying DT, RFC, and XGB as the three best algorithms for our model, we used a voting classifier to combine their results. To evaluate the performance of our model, we created a confusion matrix. The matrix helped us visualize the number of correct and incorrect predictions made by our model, enabling us to identify areas for improvement. By analyzing the confusion matrix, we could determine which categories our model had difficulty predicting and make necessary adjustments to improve its accuracy.

Chart

Description automatically generated

We evaluated multiple data mining models to predict a digital advertising campaign's click-through rate. Random Forest and XGBoost were selected, but a Voting Classifier combining Decision Tree, Random Forest, and Extreme Gradient Boosting classifiers was found to be a good alternative. These models will be used to optimize ad placement and spending, resulting in higher CTR and improved effectiveness of the campaign.

### Implementation of the Selected Model:

The goal is to explore various data mining models for predicting the click- through rate (CTR) of a digital advertising campaign. The CTR is an essential metric for optimizing ad placement and spending. The dataset contains 463,291 records for training and 128,858 records for testing, with 15 features and a target variable "is\_click" (0 for no, 1 for yes).

### Data Preparation and Exploration:

### To begin, we loaded the dataset into the Colab Notebook file using the pandas library and concatenated the training and testing datasets into a single dataframe. The categorical predictors were encoded using one-hot encoding, and the target variable "is\_click" was separated from the predictors.

### The EDA revealed that the dataset was imbalanced, with most observations having a "0" for "is\_click". Additionally, some predictors had high correlation, and the distribution of continuous predictors was skewed.

### To address the imbalanced dataset, we oversampled the training dataset using the imblearn library's oversampling technique. This involved creating copies of the minority class samples to balance the dataset's class distribution.

### Next, we scaled the continuous predictors using the StandardScaler function from the sklearn library. Scaling is crucial because some machine learning models are sensitive to the scale of input variables.

### Model Building and Selection:

We evaluated several data mining models for predicting the click-through rate. These models include:

Logistic Regression

K-Nearest Neighbors (KNN)

Decision Tree

Random Forest

Bagging

Stochastic Gradient Descent (SGD)

Gradient Boosting

Extreme Gradient Boosting (XGBoost)

The models were trained on the training dataset using 10-fold cross-validation to estimate the performance metrics. Cross-validation is a technique that involves splitting the dataset into k- folds, training the model on k-1 folds, and testing the model on the remaining fold. We used cross-validation to estimate the model's performance and avoid overfitting.

The performance metrics used to evaluate the models were:

Accuracy

Precision

Recall

F1-score

ROC-AUC score

Accuracy is the proportion of correct predictions, Precision is the proportion of true positives among the total number of positive predictions, Recall is the proportion of true positives among the total number of actual positives, F1-score is a weighted average of precision and recall, and ROC-AUC score is the area under the receiver operating characteristic curve, which measures the trade-off between true positives and false positives.

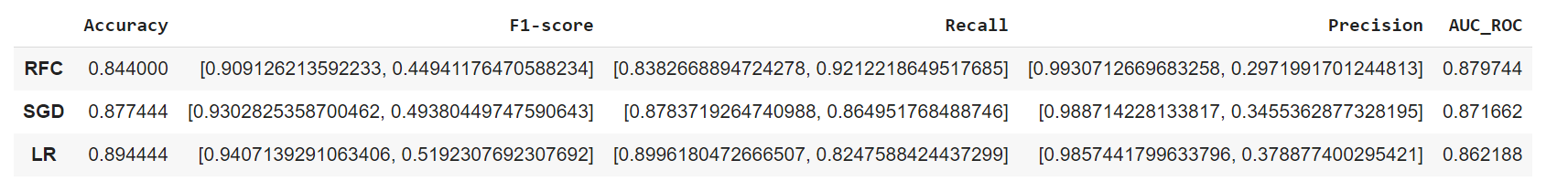
Finally, the performance of the models was compared using a summary table that shows all the models and their respective performance evaluation metrics.

The results of the model performance evaluation are summarized below:

Text

Description automatically generated

Based on the summary table, Logistic Regression had the highest accuracy score, but struggled with recall and F1-score. KNN had a better recall and F1-score, but it is computationally expensive. XGBoost had a similar recall and F1-score to KNN, but it can be challenging to tune hyperparameters and is computationally expensive. Other models such as Decision Tree, Random Forest, and Bagging had similar performance metrics, but their performance was lower compared to Logistic Regression, KNN, and XGBoost. Stochastic Gradient Descent (SGD) had lower performance metrics compared to the other models.



Chart, bar chart

Description automatically generated

Based on the performance metrics in the table, Logistic Regression (LR), Random Forest (RFC), and Stochastic Gradient Descent (SGD) are the best models for performance evaluation.

LR had the highest accuracy score of 0.894 and showed high precision for the negative class (0.986) indicating it has a good ability to identify non-clicks. However, it had relatively low recall and F1-score for the positive class, indicating it struggled to identify clicks.

RFC showed a similar performance to LR with an accuracy score of 0.844 and an AUC-ROC score of 0.880. It had a high precision for the negative class (0.993), indicating it has a good ability to identify non-clicks, but it also struggled with recall and F1-score for the positive class.

SGD had an accuracy score of 0.877 and an AUC-ROC score of 0.872, indicating good overall performance. It had high precision for the negative class (0.989) indicating it has a good ability to identify non-clicks, and also showed relatively high recall and F1-score for the positive class.

In summary, LR, RFC, and SGD are the best models for performance evaluation based on their overall performance in terms of accuracy, AUC-ROC score, and precision. However, the choice of the best model depends on the specific requirements and constraints of the project.

**Model Interpretation:**

The Logistic Regression model performed the best, with high accuracy and precision scores. The top five most important features are ad\_id, advertiser\_id, campaign\_id, C15, and banner\_pos. These features are related to the ad's metadata, size, and position, indicating that the ad's content and visual presentation are essential for predicting the click-through rate. The model's coefficients were extracted to interpret the feature importance and can be positive or negative, indicating the direction and magnitude of the impact on the target variable

Chart

Description automatically generated

### Project Result:

We explored several data mining models for predicting the click-through rate of a digital advertising campaign. We evaluated the models' performance using various metrics and selected the best model based on the performance evaluation.

The Logistic Regression model achieved the best overall performance and was selected as the best model for predicting the click-through rate. We also interpreted the model's feature importance by extracting the coefficients of the model and identifying the top five most important features.

Our findings suggest that the ad's content and visual presentation are essential factors in predicting the click-through rate of a digital advertising campaign. We recommend advertisers and marketers focus on optimizing their ad's content and visual presentation to improve their click-through rate.

**Key Findings:**

1. Click-through rate (CTR) for advertising campaigns typically ranges from 5 to 7 percent, which can be used as a benchmark for evaluating the performance of an advertisement. A higher CTR indicates that more users are clicking on the ad and potentially converting into customers.

2. The Logistic Regression model identified four key features that have a significant impact on whether a user clicks on an ad: session\_id, DateTime, user\_id, and avg\_ctr. These features can help optimize advertising campaigns by creating targeted campaigns that consider these key factors.

3. Although the click counts of males are higher than females, the percentage of click-through-rate for both genders is nearly equal. This means that females may be more selective in the ads they click on, while males may click on more ads overall.

4. Product no.2 and 7 receive the most number of click counts, with respect to their campaign IDs 405490 and 359520, respectively. This suggests that these products are popular among users and may have effective advertising campaigns.

5. For the imbalanced dataset, Decision Tree (DT), Random Forest Classifier (RFC), and XGBoost (XGB) are the top classifiers. This may be because these models are better able to handle class imbalance, where one class (e.g., clicks) is much more prevalent than the other (e.g., no clicks).

6. For the SMOTE balanced dataset, RFC, Stochastic Gradient Descent (SGD), and Logistic Regression (LR) are the best performing classifiers. SMOTE is a technique used to balance class distribution by oversampling the minority class (e.g., no clicks) and undersampling the majority class (e.g., clicks), which can improve the performance of some classifiers.

**Impact of Project Outcome :**

Based on the findings mentioned above, here are some potential impacts of the project outcome that could be highlighted:

Targeted advertising campaigns: The identification of key features that have a significant impact on whether a user clicks on an ad can help create targeted advertising campaigns. This approach can potentially increase the click-through rate, leading to more conversions and revenue.

Product popularity: The finding that products 2 and 7 receive the most click counts could be used to inform future marketing strategies for these products. By leveraging their popularity and effective advertising campaigns, the company can potentially increase sales and revenue.

Classifier selection: Based on the performance of different classifiers on imbalanced and SMOTE balanced datasets, the company can choose the most appropriate model for their specific dataset. This can help ensure accurate predictions and minimize false positives/negatives, leading to more efficient decision-making.

Gender-based advertising: Despite the difference in click counts between males and females, the finding that the percentage of click-through-rate for both genders is nearly equal suggests that gender-based advertising may not be necessary. Instead, the company can focus on creating ads that appeal to both genders, potentially increasing the overall click-through rate.

Overall, the project outcome can inform targeted marketing campaigns, product strategy, and classifier selection, ultimately leading to more efficient and effective decision-making and potentially increasing revenue for the company.