Predicting Click-through Rate of an Advertisement Using Machine Learning

Divyasha Priyadarshini (2022180), Kirti Jain (2022250)
Mahi Mann (2022272), Tanish Verma (2022532)

November 29, 2024

Abstract

Advertisement industry is a multi-billion dollar enterprise. From the user's perspective, some advertisements are appealing, while others are not, making engagement a key factor. Understanding the factors that drive user engagement can help designers and advertisers make more informed decisions about the type and frequency of advertising. The project includes pre processing on the chosen dataset, employing different models and doing hyper parameter tuning for to increase click-through rate prediction accuracy. For each model, changes in the preprocessing step are done to increase relevant metric. GitHub Link to project

1 Introduction

This project aims to develop a machine learning model that predicts the CTR of advertisements by analyzing user patterns and key features from available data. By improving the prediction accuracy, the model will provide insights for advertisers to enhance ad design and investment decisions, ultimately increasing user engagement and maximizing the return on marketing investments.

2 Literature Survey

- **Click-Through Rate Prediction in Online Advertising: A Literature Review** - Yanwu Yang , Panyu Zhai (2022) conducted a comprehensive literature review on click-through rate (CTR) prediction, focusing on modeling frameworks and their evolution. They emphasize the significance of understanding user behavior and advertising relevance in predicting CTR. The authors categorize various CTR prediction models, including multivariate statistical models, factorization machines, and deep learning models, discussing their strengths and limitations. They note that factorization machines, for instance, handle sparse data more effectively by considering feature interactions, while deep learning models can capture complex, high-order interactions. The review also outlines key challenges in the field, such as model reproducibility, the lack of standardized evaluation protocols, and the need for further exploration in areas like feature interaction and model interpretability.

- **Ad Click Prediction: A Comparative Evaluation of Logistic Regression and Performance Metrics** by Niharika Namdev and Nandini Tomar (2023). This paper explores the use of logistic regression for predicting ad click-through rates. They highlight the model's simplicity and efficiency, particularly for large-scale applications. Logistic regression is noted for its interpretability, making it a popular choice in real-world applications. However, the authors also point out that the model struggles with capturing non-linear interactions between features, which limits its predictive performance in complex environments. Their work includes a comparative analysis of logistic regression against more complex models, focusing on key performance metrics such as precision, recall, and accuracy.

- **BARS-CTR: Open Benchmarking for Click-Through Rate Prediction** by Jieming Zhu, Jinyang Liu and Shuai Yang (2023). The paper addresses the issue of inconsistent evaluation methods in CTR prediction research. Their paper introduces the BARS-CTR framework, which provides an open benchmarking system for evaluating CTR models. By standardizing datasets and evaluation metrics, BARS-CTR aims to improve the reproducibility and comparability of results across studies. The authors argue that without a consistent benchmarking framework, comparing model performance becomes difficult, leading to skewed results and limited real-world applicability. They propose that future research should focus on improving model consistency and addressing data imbalance issues that can affect CTR predictions.

3 Dataset

- **Dataset Link**: Link

- **Dataset Description**:
  - id: ad identifier
  - click: 0/1 for non-click/click
  - hour: format is YYYYMMDDHH, so 14091123 means 23:00 on Sept. 11, 2014 UTC.
  - C1 – anonymized categorical variable
  - banner_pos
  - site_id
  - site_domain
  - site_category
  - app_id
  - app_domain
  - app_category
  - device_id
  - device_ip
  - device_model
  - device_type
  - device_conn_type
  - Column 14- Column C21 - anonymized categorical variables
- **Data Description**: The dataset consists of click-through rate (CTR) data, with records related to online advertisements. The dataset is split into two parts: a training set that spans 10 days of advertisement interactions and a test set containing 1 day of interactions for evaluating model performance. The key features in the data include both numerical and categorical columns such as advertisement attributes, device type, connection type, and timestamp information.
  - **Train Set**: Contains 10 days of click-through data. Non-clicks and clicks are subsampled based on different strategies. The dataset contains approximately 40 million records.
  - **Test Set**: Consists of 1 day of advertisement interactions, used to validate model predictions.
  - **Key Columns**:
    - **Categorical Columns**: Include device types, connection types, and other encoded features like C15, C16, C19, C21, etc.
    - **Numerical Columns**: Include features such as click, hour, banner_pos, device_type, etc.
    - **Target Variable**:click – a binary variable indicating whether an advertisement was clicked (1) or not clicked (0).

4 Data Preprocessing Techniques

- **Random Sampling:** : Given the large dataset size (around 40 million records), a random sample of 5 million records is selected for training. This ensures computational efficiency without compromising data diversity. Random rows were skipped using Python's random.sample method to balance the dataset size
- **Datetime Conversion**:The hour column, which contains timestamp information, is converted into a datetime format (%y%m%d%H). This transformation is crucial for extracting features like the hour of the day, day of the week, or month from the timestamp.
- **Handling Missing Values**:After loading the data, a check for missing values is conducted using the isnull() function. Columns with missing values can negatively affect model performance, so detecting and handling them is an essential step in preprocessing.
- **Outlier Capping**: Outliers in numerical columns (C15, C16, C19, C21) are capped at the 98th percentile. This approach prevents extreme values from skewing the model's predictions, a common technique for managing skewed distributions in machine learning.
- **Categorization of Columns**: The dataset is separated into numerical and categorical columns for more targeted preprocessing. Numerical columns include id, click, hour, device_type, etc., while categorical columns include encoded features like C14, C15, C16, etc.
- **Feature Encoding**: A Target Encoder is applied to categorical variables. Target encoding replaces categorical values with the mean of the target variable (in this case, click) for each category. This technique is particularly useful for high-cardinality categorical features, where one-hot encoding may introduce too many dimensions.
- **Feature Reduction**: Some unnecessary columns such as month and C20 are dropped from the dataset. Additionally, click is renamed to y to clarify its role as the target variable in modeling, and hour_time is renamed to hour for consistency.
- **Data Sampling for Efficiency**: To see the effectivenes of the model, only 10% of the test data is used initially due to the original size if the data being huge. This downsampling is

achieved with the sample() method, ensuring that the sample is representative of the overall data distribution.

– **Exploratory Data Analysis (EDA)**:
  * Histograms are plotted for all numerical columns to understand the data distribution and spot potential anomalies or skewed data.
  * A Pearson correlation matrix is generated to visualize the relationships between numerical features, helping to identify multicollinearity or highly correlated variables that could affect the model's performance.
  * **Removing Correlated columns**: Highly correlated columns like C14 and C17 are removed as those columns don't give much information/input.

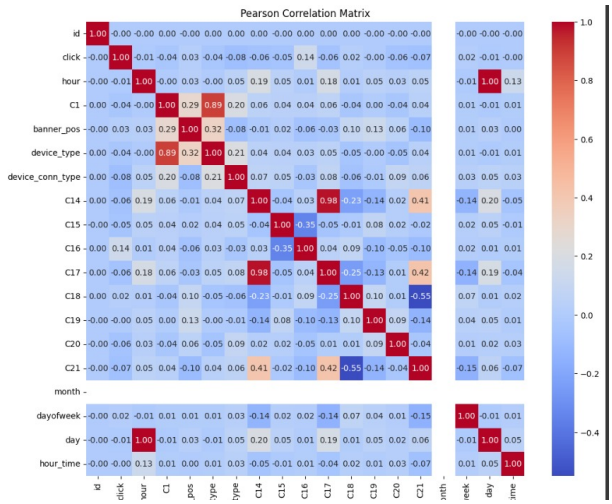

Figure 1: Correlation matrix

5 Methodology and Model Details

This project aims to develop a machine learning model that predicts click-through rate of advertisements with high accuracy. The model will offer valuable insights for advertisers and investors, aiding in designing and investment decisions and increasing engagement from the users.

– **Logistic Regression**:The first model applied is Logistic Regression, a simple yet effective classification model often used for binary classification problems like click-through prediction. The model is trained on the balanced training data, and predictions are made on both the training and test sets.
  * **Evaluation Metrics for Logistic Regression:**
    · Accuracy: Measures the proportion of correct predictions.
    · Recall: Measures the ability to capture true positive clicks out of all actual clicks.
    · Precision: Measures how many of the predicted clicks were actually correct.
    · Confusion Matrix: Provides counts of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN)
  * **ROC Curve**: To further evaluate the model, the Receiver Operating Characteristic (ROC) curve is plotted, which shows the trade-off between the true positive rate (TPR) and false positive rate (FPR). The area under the ROC curve (AUC) is also computed to assess overall performance.

– **Decision Tree**: A Decision Tree classifier is trained next. Decision trees are non-parametric models that split the data into subsets based on feature values, making decisions at each node.
  * **Evaluation Metrics for Decision Tree**: The same evaluation metrics (accuracy, recall, precision) are computed for both the training and test sets. Like logistic regression, the confusion matrix is used to calculate true positives, false positives, etc.
  * ROC curve is also plotted similarly to logistic regression model.

– **Random Forest Classifier**:The Random Forest classifier is an ensemble model that combines multiple decision trees to improve accuracy and reduce overfitting. The random forest aggregates predictions from each tree to make the final prediction.
Hyper parameter tuning on RF parameters like **maximum depth** and **n estimators** increase the accuracy of the model.
5-fold Cross Validation is also applied in order to increase the training and testing metrics.
  * **Evaluation Metrics for Random Forest**: The same evaluation metrics (accuracy, recall, precision) are computed for both the training and test sets. Like logistic regression, the confusion matrix is used to calculate true positives, false positives, etc.

    - ROC curve is also plotted similarly to logistic regression and decision tree model.

– **Perceptron**: It is a supervised learning algorithm used for binary classification tasks. Here, the binary classification is whether or not a particular ad is clicked. **Hyper Parameter Tuning** in parameters like learning rate and number of nodes etc. was applied to get a model with good accuracy.
    - **Evaluation Metrics for Perceptron**: he same evaluation metrics (ac- curacy, recall, precision) are computed for both the training and test sets. The confusion matrix is used to calculate true positives, false positives, etc.
    - ROC curve is also plotted.

– **XGBoost Model**: An efficient implementation of gradient boosting, designed for supervised learning tasks like regression, classification, and ranking. It uses a combination of a loss function and a regularization term to prevent overfitting.
    - **Grid Search**: It is a way of Hyper parameter tuning in XGBoost Model to find the combinations that yield the best results.
    - **Evaluation Metrics for XGBoost**: he same evaluation metrics (ac- curacy, recall, precision) are computed for both the training and test sets. The confusion matrix is used to calculate true positives, false positives, etc.
    - ROC curve is also plotted.

6 Results and Analysis

- **Logistic Regression**: Logistic Regression was the first model applied, offering a straightforward approach to binary classification. Logistic regression performed reasonably well given the simplicity of the model, showing moderate to good predictive power. However, it may not capture complex interactions between features.
    - **Training Accuracy**: 89.2117% (High)
    - **Testing Accuracy**: 89.3121% (High)
- **Decision Tree**:The decision tree model tended to overfit the training data. This indicates that the model was memorizing the training data rather than generalizing well to unseen data. But after futher preprocessing and removal of correlated columns, the accuracy increased.
    - **Train Accuracy earlier**: 99.7138% (High)
    - **Test Accuracy earlier**: 65.5123% (Low, compared to the training accuracy)
    - **Train Accuracy later**: 99.68% (High)
    - **Test Accuracy later**: 96.31%
- **Random Forest Classifier**:Like the decision tree, the random forest model overfitted the training data, achieving near-perfect training accuracy. But after hyperparamter tuning and Cross validation, the test accuracy also increased.
    - RF with and without parameter tuning:
    - **Train Accuracy earlier**: 99.7135% (High)
    - **Test Accuracy earlier**: 50.6561% (Low, compared to the training accuracy)
    - **Train Accuracy later**: 92.43% (High)
    - **Test Accuracy later**: 91.90% (High)
    - RF with Cross Valdation:
    - **Train Accuracy**: 99.68% (High)
    - **Test Accuracy**: 96.39% (High)
- **Perceptron**: Hyper Param- eter Tuning in parameters like learning rate and number of nodes etc. was applied to get a model with good accuracy and prevent any overfitting).
    - **Train Accuracy**: 88.09%
    - **Test Accuracy** : 87.98% (similar to train accuracy, no overfitting)
- **XGBoost Model**: The initial XGBoost model is simulated with and without Grid Search (hyperparameter tuning). Grid search tuning gave better accuracy.
    - **Train Accuracy without Grid Search**: 91.25% (High)
    - **Test Accuracy without Grid Search**: 91.11% (Low, compared to the training accuracy)
    - **Train Accuracy with Grid Search**: 95.21% (High)
    - **Test Accuracy with Grid Search**: 93.52% (no overfitting)

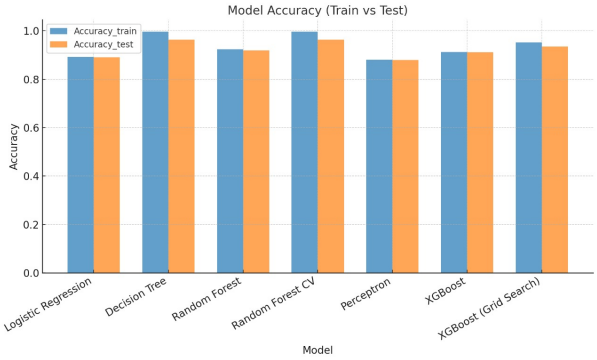

Figure 2: Accuracy for different models

7 Conclusion

* **The key takeaways from the project done are:**
  - **Imbalance in Clicks:** The dataset is heavily imbalanced, with far fewer clicks than non-clicks. Techniques like random oversampling is used to tackle this issue.
  - **Importance of Feature Engineering:** Converting timestamp features like the hour column into usable date time features highlights the importance of extracting useful information from raw data.
  - **Model simplicity can outperform complex models when overfitting is an issue,** as evidenced by logistic regression's moderate success compared to overfitting in decision trees and random forests.
  - **Target encoding** is an effective way to handle high-cardinality categorical features, improving model performance by reducing dimensionality.
  - **Parameter Tuning**: Hyperparameter Tuning enable us to control how a model learns and generalizes, and their values significantly affect the model's accuracy, speed, and ability to handle unseen data.
* **Challenges**
  - **Data size**: Due to large data size, the computational resources at hand like RAM, processing power and storage etc. were limited.
  - **Trade off between model complexity and resources**: more complex models demand greater resources, while simpler models are more resource-efficient but may compromise performance. Therefore, using models more complex than XGBoost wasn't feasible.
  - **Correlated Features**: The correlated features played a major role in the decreasign the accuracy.
* **Contribution**:
  - **Divyasha Priyadarshini:** Model development and Exploratory data analysis
  - **Mahi Mann:** Preprocessing, feature engineering and hyperparameter tuning
  - **Kirti Jain:** Data collection, comparative analysis with baselines, documentation and report preparation.
  - **Tanish Verma:** Model Development and evaluating model performance

8 Bibliography

* Dataset and Preliminary Information
* Click-Through Rate Prediction in Online Ad- vertising: A Literature Review - Yanwu Yang , Panyu Zhai (2022)
* Ad Click Prediction: A Comparative Evalu- ation of Logistic Regression and Performance Metrics by Niharika Namdev and Nandini Tomar (2023).
* BARS-CTR: Open Benchmarking for Click- Through Rate Prediction by Jieming Zhu, Jinyang Liu and Shuai Yang (2023)
* Lecture slides for topic understanding