Click-Through Rate Analysis using Machine Learning

Aishwarya Sen (as6718), Chantal Ye (xy2541), Hannah Portes (hsp2122), Henrik Tseng (ht2600), and Kunul Kundu (kk3585)

ABSTRACT

In online advertising, click-through rate (CTR) is a very important metric for evaluating ad performance. As a result, click prediction systems are essential and widely used for sponsored search and real-time bidding. This project aims at building a classification model to predict click/no-click of Avazu banners in web pages and applications. The analysis includes exploratory data analysis of the various features affecting CTR, data preprocessing and finally implementation of several machine learning techniques aimed at identifying the most efficient classification model.

1. Dataset Description

The dataset is obtained from Kaggle, a popular data science platform, on users' view and click through behaviors in Avazu web pages and apps through 11 days. It contains 40,428,968 rows and 23 features which are a mix of numeric and categorical variables, including the target feature click: 0/1 for no click/click. The remaining features can be categorized as: site features (site_id, site_domain, site_category), app features (app_id, app_domain, app_category), device features (device_id, device_ip, device_model, device_type, device_conn_type), anonymized categorical features (C1, C14-C21), id: ad identifier, banner_pos: banner position and hour: which gives the date and hour of click.

2. Exploratory Data Analysis

Various analyses and visualizations were carried out in order to get a basic idea of the dataset, the distribution of its features along with their effect on the target variable (click). Some of the key observations are:

1. The distribution of number of clicks versus no clicks shows a high imbalance of 82.91% of the data for when the user has not clicked and just 17.09% of the data for when the user has clicked, indicating the need of data imbalance treatment.

2. Feature engineering is used to study the date-time features. One such example (Figure 1) is first extracting the hour from the 'hour' feature which is in the format YYMMDDHH. It is observed that the maximum number of clicks occur at the 13th or 14th hour i.e., 1 or 2pm. A plot is then created for the click through rate (CTR) by dividing the number of clicks by number of observations for each hour. Here we observe that highest CTR occurs at midnight hours (1, 7 and 15).

3. An arbitrarily picked variable - Device Type, has been analysed (Figure 1). Even though initial visualization shows that device type 1 has most clicks, the CTR chart proves that device type 0 has amassed most clicks.

Thus, the above experiments show that mere observations of number of clicks do not show the correct picture and further analysis and exploration helps find useful insights and patterns. Several other charts and graphs analyse similar distributions and characteristics of the features and are included in the code.

3. Data Sampling and Preprocessing

In this section several raw data issues are addressed:

- **Large Unmanageable Dataset**: The dataset includes 40,428,967 observations altogether, loading which as a whole lead to storage overflow and thus, the dataset is sampled while loading it.
- **Dropped Columns**: Using a correlation matrix, the correlation of features is studied. It was found that, C1 and device_type have a correlation coefficient of 0.9 and C14 and C17 have a correlation coefficient of 0.98. Arbitrarily, C1 and C14 are dropped to avoid multicollinearity-related issues. Additionally, some columns do not provide possible information for the prediction, and could lead to overfitting. These columns are user_id, device_id and user_ip. Theoretically, CTR should be predicted upon website features or group features of users, but not unique features of users and devices, otherwise the model might overfit on those uniqueness. These columns are thus dropped.
- **Imbalanced Dataset**: The ratio of negative and positive values in the target is about 5 : 1. To address problem 1 and 2, undersampling technique is applied to simultaneously decrease the data size and balance it. Specifically, 50,000 samples are randomly drawn from positive and negative classes separately, and then merged back to be a manageable dataset.

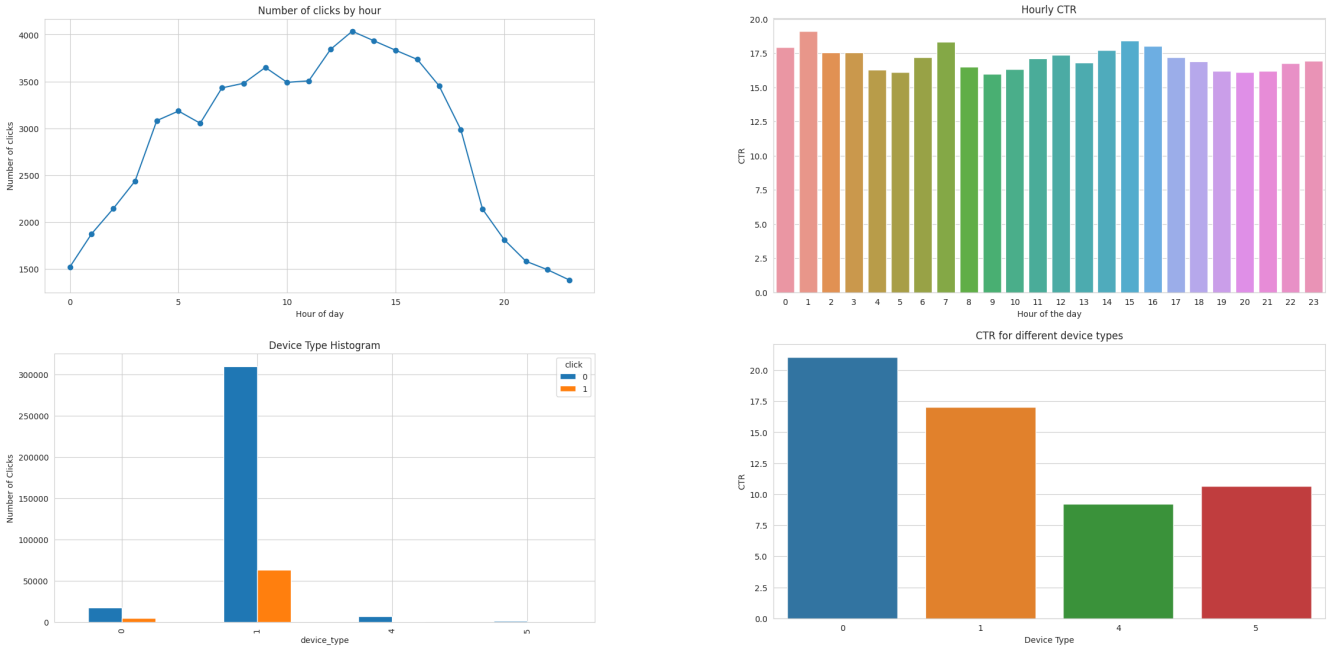

Figure 1. Top: Line graph showing average number of clicks per hour of the day (left) and bar chart showing hourly CTR (right); Bottom: Distribution of number of clicks for different banner positions (left) and bar chart showing CTR for different banner positions (right)

- **Categorical Features**: Target encoding technique is leveraged to handle the categorical variables, considering that most features bear a considerable number of classes.
- **Feature Range Variation**: Standard Scalar is applied to reformulate the data to the same scale.

4. Methodology

Several classification models are experimented with, in order to choose the most efficient one. Different techniques for improvement of evaluation scores are used including hyperparameter tuning, cross validation, calibration of models and ensemble modelling.

4.1. Machine Learning Models

- **Basic Classification Models**: Logistic regression models are used as a simple trial of classification, which are widely used for their interpretability and effectiveness in high-dimensional feature spaces. For better performance, regularization and feature selection techniques are applied to get a slight improvement in accuracy. Also, through the coefficients of logistic regression, it can be found that app_id, site_id and device_model are redeemed as the most impactful features.
- **Tree-Based Models**: Tree models, including decision tree and random forest, provide an intuitive and efficient way to model nonlinear relationships between the features and the target variable. Since there are multiple hyperparameters to tune at a time, grid search is specifically applied to search for the best hyperparameter set. To try and further raise the performance, calibration and cross-validation techniques was applied to the random forest classifier. Feature importance scores are then tracked to investigate the influence of all features on the click through rate, among which site_domain, app_id, site_id and device_model are ranked the top features (Figure 2).
- **Neural Network**: A simple 3-layer convolutional neural network (CNN) is used. Hyperparameters are varied through random search to observe a small improvement in performance.
- **Ensemble Modelling**: Finally, ensemble techniques are used to combine the predictions of the different models and improve their overall performance. Bagging, boosting (AdaBoost) and gradient boosting as well as stacking techniques are utilized, which are popular ensemble methods that can be used with a wide range of machine learning models. The bagging classifier shows a significant improvement in training accuracy.

4.2. Explainable AI

In addition to the above implemented ML techniques, in order to further elucidate our results, LIME or Local Interpretable Model-Agnostic Explanations is used to explain a random prediction by demonstrating the top features and their range of values that contribute to the correct prediction (Figure 2). The provided LIME explanation highlights the top 5 features and their impact on the predicted label, where higher values of 'site_id' and 'site_domain' and lower values of 'app_id', 'device_model', and 'device_conn_type' contribute to a higher probability of predicting label 1.

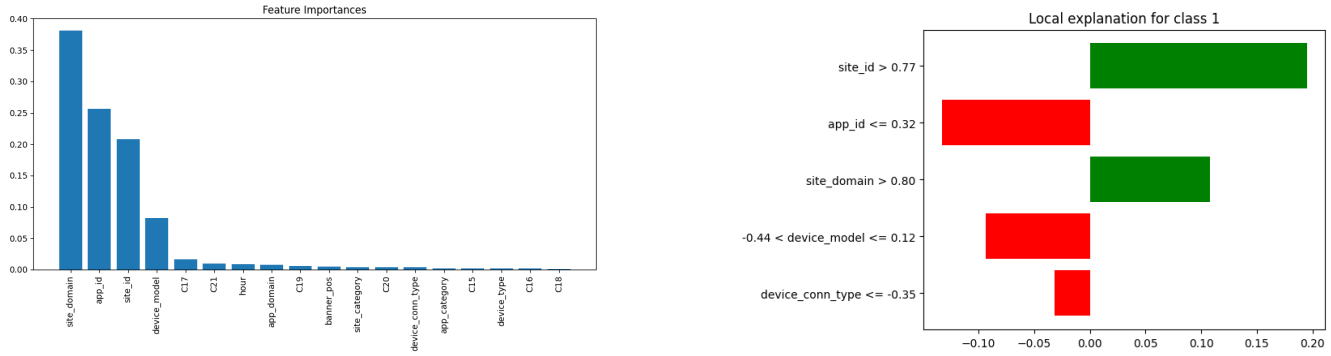

Figure 2. Left: Feature Importance Plot for Decision Tree; Right: Local Interpretation plot by LIME

5. Observations and Results

Through the whole model tuning and selection process, accuracy score and f-1 score are used as the two main evaluation metrics. The outcomes are summarized as below:

Model	Training accuracy	Training F-1	Validation accuracy	Validation F-1	Testing accuracy	Testing F-1
Logistic regression	0.7	0.72	0.65	0.68	0.66	0.68
Decision tree	0.70	0.70	0.66	0.65	0.66	0.66
Random forest	0.72	0.72	0.67	0.67	0.67	0.67
Neural Network	0.70	0.70	0.66	0.66	0.66	0.66
Gradient boosting	0.70	0.70	0.70	0.70	0.67	0.67
Bagging Classifier	0.81	0.81	0.65	0.65	0.65	0.65
AdaBoost Classifier	0.69	0.68	0.67	0.67	0.66	0.66
Stacking Classifier	0.73	0.73	0.67	0.67	0.66	0.66

Table 1. Model Performances

While all models have demonstrated similar accuracy and F1-scores across train, test and validation sets, we observe that the ensemble models have achieved better scores, with the gradient boosting and stacking classifiers achieving consistent high accuracy scores and the bagging classifier achieving the highest training score.

Additionally, it is observed that 'site_id', 'app_id' and 'device_model' are all three among the most important features as per the logistic regression model and the decision tree and also used by LIME to explain a correct prediction. It can thus be concluded that these three features, along with 'site_domain' contribute significantly to model prediction.

6. Conclusion

In conclusion, machine learning models can be extremely useful by providing predictions of click. These models utilize complex algorithms and statistical techniques to analyze collected data and generate insights that can help direct advertising campaigns for optimal performance. While all the models explored are suitable for click prediction, choosing the right model and optimizing the performance can be challenging. This project thus demonstrates how various ML techniques can be leveraged to improve the models and additionally how explainable AI can be used to elucidate the results.