

## The Huawei DIGIX Advertisement CTR Prediction Competition

Advertisement click-through-rate (CTR) prediction is the key problem in the area of computing advertising. Increasing the accuracy of advertisement CTR prediction is critical to improve the effectiveness of precision marketing. In this competition, big advertising datasets have been released that are anonymized. Based on the datasets, contestants were required to build Advertisement CTR prediction models. Can you set up a competitive prediction model as well?

The datasets (after being masked) contain the advertising behavior data collected from seven consecutive days, including a training dataset and a testing dataset.

### Exercise 1: Data Loading and Exploration

Download the Advertising CTR Dataset from <https://openBIGdata.org> and load the data in R.

- Identify the number of variables and observations in the training and testing datasets.
- Explore the distribution of different variables in the dataset. Identify the most common values for categorical variables and visualize their frequencies.
- Check for missing values in the dataset. Decide on appropriate strategies for handling missing data.
- Compute summary statistics for numerical variables in the dataset.
- Visualize the distribution of user ages and device prices in the dataset.

### Exercise 2: Model Building and Evaluation

This exercise aims to build and evaluate prediction models for Advertisement CTR.

- Split the training dataset into a training set and a validation set (e.g., 80-20 split).
- Train a logistic regression model to predict the likelihood of ad clicks based on relevant features in the dataset. Evaluate the model's performance using appropriate metrics such as accuracy, precision, recall, and F1-score.
- Train a decision tree classifier to predict ad clicks. Experiment with different hyperparameters (e.g., max\_depth) and compare the performance of different models.
- Train a random forest classifier to predict ad clicks. Tune the hyperparameters using techniques like grid search or random search.
- Evaluate the performance of all models on the validation set and select the best-performing model based on chosen evaluation metrics.

### Exercise 3: Feature Engineering and Model Improvement

In this exercise, we'll focus on feature engineering and improving model performance.

- Explore feature importance using techniques like permutation importance or SHAP values. Identify the most important features for predicting ad clicks based on one of the models in exercise 2.
- Engineer new features from existing ones (e.g., interaction terms, polynomial features) and assess their impact on model performance.

- 
- c) Experiment with different data preprocessing techniques such as feature scaling, one-hot encoding, and feature transformation. Evaluate the effect of these techniques on model performance.
  - d) Train an ensemble model (e.g., gradient boosting classifier) using the engineered features. Compare its performance with previous models and analyze any improvements.
  - e) Discuss potential strategies for further improving model performance, such as collecting additional data, feature selection, or trying more advanced modeling techniques like neural networks. You can also use ChatGPT or an alternative AI tool (see e.g., <https://www.gettingstuffdone.ai/>) for further suggestions.

## Exercise 4: Model Deployment

In this exercise, we'll explore how to deploy and interpret our trained models.

- a) Deploy the best-performing model from Exercise 2 or Exercise 3 on the testing dataset and generate predictions for ad clicks.
- b) Evaluate the model's performance on the testing dataset using the same evaluation metrics as before.
- c) Interpret the model's predictions and feature importance to understand which factors influence ad clicks the most.