# Kaggle Competition: CTR Prediction

## 1 Introduction

This project involves predicting whether a user will click on an online advertisement ("is_click"), given various features related to the user, campaign, webpage, and session characteristics. The target variable is binary, indicating whether a user clicked on an advertisement (1) or not (0). Our analysis focuses on developing robust feature engineering techniques to capture complex user behaviors and temporal patterns, ultimately improving the accuracy of CTR prediction.

## 2 Method

### 2.1 Data Handling

- **Missing Data Treatment:**
  We have treated missing values differently, depending on the severity of the completeness rate and the underlying meaning of the features at hand.
  - For demographic features with 4.7% missing values (gender, age level, user depth), we first attempted to fill gaps using other sessions of the same user, falling back to mode imputation only when user-level data was unavailable. This preserved the consistency of user characteristics while ensuring completeness.
  - "Product Category 2" had a significant amount of missing values (79.16%), so we added a separate feature indicating its "missingnes", while leaving the original values intact to avoid introducing potentially misleading data.
  - The City Development Index, missing in 27. 60% of the cases, was treated with a hybrid approach that combined mode imputation and a "missingness" indicator. Features with less than 1% missing values were handled using straightforward type-dependent imputation: mode for categorical variables and median for numerical ones.

  The above method was applied equally to the training and testing data sets. In addition, we have validated the correctness and validity of these choices by looking at the feature distributions, which confirmed that key statistical properties remained stable post-imputation.

- **Data Cleaning:**
  Our initial data cleaning process focused on removing noisy data while preserving valuable information.
  - We removed empty rows and full row duplicates, reducing the dataset from 389,163 to 369,652 rows (a reduction of 19,511 rows or approximately 5%).
  - We deliberately chose not to remove statistical outliers, particularly in user behavior metrics like session counts and engagement levels. While we identified some extreme values (e.g., users with up to 206 sessions), these represent genuine user patterns rather than data errors and could be valuable predictors for our model.

- **Feature Type Conversion:**
  - DateTime values were standardized to datetime format for consistent temporal analysis.
  - ID-like features that behaved as categories (campaign_id, webpage_id, product_category_1) were converted to integer type while being treated as categorical in our modeling.
  - User engagement metrics (age_level, user_depth, city_development_index) were converted to integers to reflect their ordinal nature.
  - Binary indicators were converted to integer type (0/1) for computational efficiency.

### 2.2 Feature Engineering

We have introduced new feature groups, while carefully preventing any data leakage (future sampling/feature sampling):

- *Temporal Features:* Created time-based features using only past data points, including hour, day of week, business hours flags, and holiday (or holiday proximity) indicators. Each temporal aggregate was calculated using expanding windows to ensure no future information leaked to our computation.
- *User Engagement Metrics:* Developed features tracking user behavior (e.g., historical CTR, session counts, time since last click) using only past sessions for each calculation. We applied log transformations to handle the high skewness in metrics like session counts.
- *Campaign Performance:* Generated campaign-level metrics (historical CTR, webpage performance) using cumulative calculations to maintain temporal integrity.
  - For the test datasets and prediction tasks, we have pre-computed global empirical params to emulate this data, which by definition relies on our target feature.
- *Interaction Features:* Created cross-features between temporal, demographic, and behavioral variables (e.g., age-weekend interaction, user-depth-time combinations) to capture complex patterns.

### 2.3 Feature Selection

Since we have introduced so many new features, we had to employ a multi-stage approach to identify the most relevant features:

- *Variance Analysis:* Removed low-variance features (variance $< 0.01$) that showed minimal variation and thus limited predictive potential.
- *Correlation Analysis:* Removed highly correlated features (correlation $> 0.95$), keeping the feature with stronger correlation to the target variable when pairs were highly correlated.
- *Information Value:* Removed features with low mutual information scores ($< 0.001$) with respect to the target variable, indicating weak predictive power.
- *Statistical Significance:* Removed features that showed no statistically significant relationship with the target variable ($p > 0.05$ in uni-variate testing).

This process reduced our feature set drastically, with key predictors including campaign success percentile, webpage performance, historical user CTR, and temporal patterns.

### 2.4 Cross-Validation

- *Feature Selection CV:* Used a 3-fold cross-validation during forward feature selection to evaluate the impact of each feature (in which, features were only kept if they improved the F1 score consistently across folds).
- *Model Evaluation CV:* Used a 5-fold cross-validation for model comparison, using positive F1-score as the scoring metric.
- *Hyper-parameter Optimization:* During grid search, we used a 2-fold cross-validation with early stopping for each parameter combination, again optimizing for positive F1 score.

Our CVs consistently used stratification to handle class imbalance, ensuring each fold maintained the same proportion of positive clicks (approximately 7%) as the full dataset.

## 3 Evaluation

### 3.1 Evaluation Metrics

- *F1 Score:* Primary metric, chosen due to class imbalance (approximately 7% positive rate) and business need to balance precision and recall.
- *ROC AUC:* Secondary metric to evaluate ranking quality across different threshold choices.

| Model | F1 Score | ROC AUC | Avg Precision |
|---|---|---|---|
| Logistic Regression | 0.139 | 0.569 | 0.087 |
| Random Forest | 0.262 | 0.714 | 0.238 |
| Gradient Boosting | 0.261 | 0.752 | 0.257 |
| XGBoost (Final) | **0.275** | **0.741** | **0.257** |

Table 1: Model Performance Comparison

### 3.2 Model Performance

We evaluated several models, with XGBoost showing the strongest performance:

### 3.3 Hyper-parameter Tuning

We used Grid search with 2-fold cross-validation, to find the following optimal parameters for XGBoost: [max_depth: 5, learning_rate: 0.2, min_child_weight: 3, subsample: 0.9, colsample_bytree: 0.9, scale_pos_weight: 10].
The final model achieved early stopping at 67 rounds, which seems to suggest good convergence without over-fitting.

### 3.4 Distribution Analysis

Post-training analysis revealed interesting probability distributions:

| Metric | Training Set | Test Set |
|---|---|---|
| Probability Range | [0.003, 0.989] | [0.257, 0.465] |
| Mean | 0.367 | 0.363 |
| Standard Deviation | 0.211 | 0.053 |

Table 2: Probability Distribution Comparison

The compressed probability range in the test set suggests potential dataset drift, though the preservation of mean probabilities indicates the model maintains relative ranking capability.

To address this for the test data prediction, we implemented a percentile-based threshold adjustment targeting the training set's positive rate (6.71%), which achieved a 6.09% positive rate in test predictions, suggesting effective calibration despite the compressed probability range.

In the next tuning iterations, we will use better probability calibration techniques, and possibly retrain the model with techniques that improve generalization.

## 4 Conclusion Next Steps

Our analysis revealed several key insights about click-through rate prediction in this context. Campaign-level features, particularly the campaign success percentile and historical performance metrics, proved to be the strongest predictors. This suggests that campaign optimization might be more impactful than user targeting for improving CTR.
The observed distribution shift between training and test sets suggests an opportunity for model improvement. For the next phase, we will look into:

- Developing more robust feature scaling methods to handle distribution shifts.
- Exploring ensemble methods that combine campaign-level and user-level predictions.
- Investigating time-series aspects of the data to better capture temporal patterns (depending on the next batches of test datasets and when they were sampled).