**ZEPTO ASSIGNMENT**

### **ABSTRACT**

This project focuses on creating a model to predict Click-Through Rate (CTR) using data from the Zepto platform over a 90-day period, with the goal of ranking products by their CTR. The dataset contains information such as search queries, city identifiers, product details, and historical performance metrics. We began by conducting an Exploratory Data Analysis (EDA) to gain insights into the data's distribution and relationships, which informed our feature engineering process. This involved refining and improving features to enhance predictive accuracy. After cleaning and preprocessing the data, we developed a model to rank products at the city level based on predicted CTR, using various metrics to evaluate and monitor performance. This comprehensive methodology aims to optimize sponsored search results through advanced data analysis and feature engineering techniques.

### **INTRODUCTION**

In the realm of digital advertising, the Click-Through Rate (CTR) is a vital metric for measuring the effectiveness of online ads. It represents the percentage of users who click on an ad out of those who view it, providing valuable insights into ad performance and user engagement. Accurate prediction of CTR is crucial for optimizing advertising strategies and maximizing return on investment.

This project utilizes 90 days of data from the Zepto platform to develop a CTR prediction model. The dataset includes a diverse set of features, such as search queries, city identifiers, product details, and historical metrics related to clicks and views. By analyzing and leveraging this data, our objective is to build a predictive model capable of ranking products based on their likelihood of being clicked.

The project is divided into several key phases: Exploratory Data Analysis (EDA) to understand the data's structure and relationships, feature engineering to create relevant predictors for the model, and model development to rank products effectively.

### **DATASET**

The dataset encompasses 90 days of detailed information on user interactions and product performance within the Zepto platform. Key fields include search\_term, which represents the search queries entered by users, and city\_id, a unique identifier for different cities. Each product is uniquely identified by product\_variant\_id, with the target variable is\_clicked indicating whether a product was clicked (1) or not (0). Additional features include metrics such as total\_clicks, session\_views, and various click-through rates (CTR) over different time periods (e.g., CTR\_last\_30\_days, CTR\_product\_30\_days). The dataset also includes platform-wide metrics like query\_product\_plt\_clicks\_60\_days and query\_product\_plt\_ctr\_60\_days, as well as product-specific details such as Product\_name, Brand\_name, latest\_margin, and ad\_revenue. This comprehensive dataset provides an in-depth view of user interactions and product performance, facilitating the development of a robust CTR prediction model.

**APPROACH:**

[**Code**](https://github.com/gvritesh/zepto.git)

### **APPROACH**

1. Data Preparation:
   * Data Collection: Utilized a dataset spanning 90 days from Zepto, containing features such as search queries, city identifiers, product details, and historical performance metrics.
   * Feature Selection: Conducted an initial assessment to identify and drop columns deemed unimportant for the prediction model. This helped streamline the dataset and focus on relevant features.
   * Data Splitting: Divided the dataset into training and testing sets using an 80-20 split to ensure the model's ability to generalize to unseen data. This was done with train\_test\_split from sklearn.model\_selection.
2. Model Building:
   * Model Choice: Chose XGBRegressor from the XGBoost library for its effectiveness in handling complex datasets and its ability to improve performance through boosting.
   * Parameter Tuning: Initialized the model with the following parameters:
     + objective='reg:squarederror': Specifies the regression objective with squared error loss.
     + n\_estimators=100: Number of boosting rounds or trees in the model.
     + learning\_rate=0.1: Step size used for each boosting round to reduce overfitting.
     + max\_depth=3: Maximum depth of the trees to control model complexity.
   * Model Training: Trained the model on the training data using the fit method.
3. Model Evaluation:
   * Predictions: Generated predictions on the test set using the trained model.
   * Metrics Calculation: Evaluated model performance with the following metrics:
     + Mean Squared Error (MSE): Measures the average squared difference between predicted and actual values, providing an indication of prediction accuracy.
     + R² Score: Indicates the proportion of variance in the target variable that is predictable from the features, reflecting the model’s goodness of fit.
4. Results Reporting:
   * Presented the Mean Squared Error and R² Score to assess the model’s performance and its ability to make accurate predictions.

This approach ensures that the model is well-prepared for prediction tasks, effectively utilizing data preprocessing, feature selection, and performance evaluation to deliver accurate and actionable results.