**Zepto Assignment**

**CTR Prediction model**

[**code**](https://github.com/Abhinavchad/zepto_assignment.git)

**ABSTRACT**

This project develops a Click-Through Rate (CTR) prediction model using 90 days of Zepto data, aiming to rank products based on their CTR. The dataset comprises search queries, city identifiers, product details, and historical performance metrics. We first conducted an Exploratory Data Analysis (EDA) to understand data distributions and relationships.This calculation guided our feature engineering process, where we refined and enhanced features for better prediction. After preprocessing and cleaning the raw features, I built a model to rank products at the city level based on predicted CTR, employing various metrics to evaluate and track performance. This comprehensive approach optimizes sponsored search results through advanced data analysis and feature engineering techniques.

**INTRODUCTION**

In the digital advertising landscape, the Click-Through Rate (CTR) is a critical metric for evaluating the effectiveness of online advertisements. CTR measures the ratio of users who click on an ad to those who view it, providing insights into ad performance and user engagement. Accurate prediction of CTR is essential for optimizing advertising strategies and maximizing return on investment.

This project uses 90 days of Zepto data to develop a CTR prediction model. The dataset includes a range of features such as search queries, city identifiers, product details, and historical click and view metrics. By analyzing and utilizing this data, the goal is to build a predictive model that can rank products based on their likelihood to be clicked.

The project encompasses 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 building to rank products effectively.

**DATASET**

The dataset provided for this project spans 90 days and includes comprehensive 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, and the target variable is\_clicked indicates 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 uses platform-wide metrics like query\_product\_plt\_clicks\_60\_days and query\_product\_plt\_ctr\_60\_days, as well as product-specific details including Product\_name, Brand\_name, latest\_margin, and ad\_revenue. This rich dataset provides a detailed view of user interactions and product performance, enabling the development of a robust CTR prediction model.

**APPROACH**

To build an effective CTR prediction model, we began by addressing missing values in the dataset. Specifically, columns such as ‘product\_name’, ‘brand\_name’, ‘category\_name’, and ‘subcategory\_name’ contained NaN values. Given that these were categorical variables, filling them with the mode was considered. However, this approach was not suitable as it would have led to recommending the same product or brand for any search query, which could undermine the model’s effectiveness. Therefore, we opted to drop these columns to ensure the model’s recommendations remain relevant and diverse.

Next, I aggregated the data by grouping it based on ‘city\_id’, further grouping on ‘search\_term’ and finally on the ‘product\_name’. The purpose of using this multiple group-by was to rank products based on search terms and city level. The final result obtained through this grouping grouped a product based in multiple hierarchies - City -> Search Term -> Product. This aggregation was crucial for ranking products based on search terms and city level. We then calculated the Click-Through Rate (CTR) using the formula CTR=total\_clicks/session\_views.

After that I performed an inner join operation between two dataframes: final and df\_cleaned. The pd.merge function is used to combine these data frames based on common columns.

final: This dataframe contains aggregated data with columns like total\_clicks, session\_views, and the computed CTR (Click-Through Rate) based on the grouping of city\_id, search\_term, and product\_name.

df\_cleaned: This dataframe holds the original dataset with additional features and columns that were cleaned and prepared earlier in the process.

Following this, we examined the final dataframe for duplicate rows and missing values to ensure data quality. Categorical columns were then encoded using LabelEncoder to prepare the data for modeling.

For feature selection, I used ‘mutual\_info\_regression’ from the sklearn.feature\_selection module. This method was preferred over correlation analysis, as mutual information captures both linear and non-linear relationships between features and the target variable. Features with low mutual information scores were removed as they contributed minimally to the model's performance.

Subsequently, the data was scaled to improve model performance. Given the remaining number of features, we applied Principal Component Analysis (PCA) to reduce the dimensionality to 15 features.

For modeling, we utilized the **XGBRegressor** with the following parameters:

squared error loss, a maximum tree depth of 3, a learning rate of 0.3, and 200 boosting rounds.

We evaluated the model’s performance using metrics such as Mean Squared Error (MSE), Mean Absolute Error (MAE), and R-squared (R^2):

* Mean Squared Error: 3.39
* Mean Absolute Error: 0.51
* R^2 Score: 0.78

These metrics indicate the model's effectiveness in predicting CTR and its ability to rank products accurately based on search terms and city level.