**Restaurant Take-Home Case Study**

**Pre-requisite: To re-run this notebook, one will only need libraries that are being imported to be installed on their system**

Please find below the approach to the solution to the given problem.

1. Load the given data in 2 separate data frames.
2. Perform fundamental analysis to check if there is any missing data before merging data frames.
3. After merging the 2 data frames into 1 based on restaurant\_id, I started with analysis which included checking the presence of null values, checking the data type
4. Since there were no missing values, I proceeded with data type conversion. Otherwise, null values need to be handled
5. During data conversion, I thought about multiple derived features such as time\_of\_day to get a better understanding of during which part of the day restaurant is receiving the max orders
6. The next step was to perform a univariate analysis to understand the data better. Thus plotted different graphs to get a better understanding
7. The next step was to perform a bi-variate analysis to understand the relationship between the 2 features
8. As for the next step, I performed outlier detection and handled them with mean. Although there is a possibility that the outliers might be coming due to other reasons about which better understanding can be acquired post discussing with business people.
9. Performed data encoding using Label encoders for ordinal features and BaseNEncoder for others. Didn’t use one hot encoding since the number of distinct values was very high.
10. Once the data was free from issues, I plotted a heatmap to understand the correlation better.
11. The next step was to build an ML model. Thus instead of trying multiple algorithms, which could have been done if more time was available, I used XGBoostRegressor to build the ML model.
12. Performed data split into 70:30 ratio between train & test.
13. Also used is a sequential feature selector to identify the prominent features.
14. Used GridSearchCV to find the best hyper-parameters
15. Used XGboot model with those parameters to perform training and prediction
16. Used root\_mean\_squared\_log\_error to check model performance. It came at 0.46, which can be treated as a baseline model, and further improvements can be made.

**Other things that can be implemented if more time was given:**

* Adding order\_queue\_number to check how many orders are currently listed for a particular restaurant on a given day and with 40-45 mins intervals
* Performing statistical analysis using methodologies like hypo-thesis testing, Chi-squared, and Annova to identify prominent features
* Using different ML algorithms
* Performing hyper-parameter tuning up to a better extent using Optuna
* Handling categorical variables in a better way