**Predicting Instacart Customers Purchasing Behaviors**

**(When they will make their next purchase and what products to expect in that purchase)**

# Approaches to achieving project goals:

# This project deals with two predictions namely when the user will make the next purchase (prediction-1) and what are all the possible products in the next purchase order (prediction-2). In the case of prediction 1, with the findings from the data exploration, I decided to classify the days since prior purchase into four categories (Class 0 – less than 10 days, Class 1 – 11 to 20 days, Class 2 – 21 to 30 days, Class 3 – greater than 30 days). I made this decision, as difference by one or two days will not defeat the stock management purpose. For prediction 2, in order to simplify the multi-label classification problem, I reframed the question as ‘will this product be in the user’s next order or not?’ turning it into a binary classification problem.

# Feature Engineering:

# With been defined the approach for modeling, I extracted various features including user-product features such as purchase frequency, reorder rate, average add to cart order and so on from the history of the users’ orders based on data exploration findings. Further, to increase the complexity, have also added user’s features such as overall purchase frequency, average cart size, etc. Similarly, I extracted overall product features to add as the input to the model.

# Data Preprocessing:

# After creating the features for both the models, I encountered missing values in the prediction-2’s feature frequency. The missing values belong to the first purchase of the user, i.e., the product is present only in the first order. Hence filled it with the 999 to depict less probability of purchase of that product. Following data cleaning, identified the categorical columns and indexed using string-indexer and converted to the corresponding numerical category using One-Hot-Encoder-Estimator. Then, all the features are assembled into a vector data frame using the Vector assembler. Finally, the labels and features are fitting in the pipeline for easy processing.

# Modeling overview:

# For prediction 1, several multi-class classification algorithms were trained and tested. Among those, the logistic regression model seems to perform better compared to others producing a test accuracy and f-measure of 75% each. Though it is not the best model, I am working on tuning the model to achieve greater accuracy and f-measure.

# On the other hand, for prediction-2 also Logistic regression seems to output better results compared to other algorithms. However, it outputs a poor f-measure of 0.49. I hope hyper-parameter tuning and increasing the complexity will produce better f-measure upon which I am working now.

# Further steps:

# In upcoming days, I will be working on to improve the performance of the model by implementing various optimization techniques. Also, I will check whether passing prediction 1 data to prediction 2 will be helpful.