Mentor: Srdjan Santic *By Anushree Srinivas*

Data Wrangling for Instacart Market Basket Analysis

There are **six** data sets in this project which I worked on:

1) **Aisles**.csv: has **2** columns aisle_id and aisles
2) **Departments**.csv: has **2** columns again department_id and department
3) **Order_products_prior**.csv: has **4** columns, order_id,product_id,add_to_cart_order and reordered.
4) **Order_products_train**.csv: has **4** columns same as order_products_prior
5) **Orders**.csv: has **8** columns, order_id, user_id, eval_set, order_number ,order_dow, order_hour_of_day, days_since_prior_order.
6) **Products**.csv : has **4** columns, product_id,product_name,aisle_id,department_id.

The first thing I did was to check for missing values in all the data sets. I used the **isnull()** function in Pandas to check for null values in the data sets and also tried to get the percentage of the null values in each data set.

To find the missing values, I first found the **tota**l for each data set using the **.isnull() and .sum()** functions. This gave me the number of null values for each column in each data set. Then to get a better idea of how significant is the null values I calculated the **percentage** of null values for each column in all the 6 data sets using the **total** (which I had calculated earlier) and then dividing it by using the **.isnull() and .count()** functions in Pandas.

Then, I made a missing values tables for better visualization using the Pandas **pd.concat** function to put together the total and percentage of each dataset called the missing_values_table for each.

There were no null values for aisles, deparments, order_product_prior, order_product_train and products dataset. Only the **orders** dataset has some null values in the days_since_prior_order column. The percentage for those null values was around 6%. Since, the null values were around 206209 out of a total of 3421083 records, I decided to drop the null values from the days_since_prior_column in order to clean the data as it would still leave us with around 3.2 million records for our data analysis. I then made a new data set for orders using the **notnull()** function for the days_since_prior_orders columns. The syntax to which is: **orders_new=orders_df[orders_df['days_since_prior_order'].notnull()]**