216364 ITMGT 25 - B

Tobi Jasper Wang Mr. Joe Ilagan

Ad Hoc Analysis Approach

I did the 'ItemBreakdown' by loading the json file to the dataframe (boom). I then created functions that will split multiple transactions from one buyer (splititemlist) and split the individual items (brandanditem, qty). I vectorize the three functions. Afterwards, I used the explode function on the vectorized 'split multiple transactions from one buyer'. I created a new dataframe (transitemsdf) to store data by vectorizing the previous exploded function. I then extract the number part from 'qty' column using .str.replace then converting it into int data type. I created another dataframe (boomdf) to store products from dataframe transitemsdf's brand: item. I created another function (transdateitem) that adds up the transaction dates per item sold. It will then return the 'transaction_date_list' which will be exploded and added to 'transitemsdf'. There, I would split the year, month, and date to extract the value of the month and changing their data types to int. Then, I'll get the individual prices of the items and storing it in a new dataframe (adf). Using the pd.merge method, I joined the two columns of 'brand:item' and 'indiv. price' into dataframe 'boomdf'. Dataframe 'boomdf' shows the product brand and item and their respective prices. Afterwards, using for loop, I got the count of each item per month, and using a function the multiplies the total quantity to its respective price, I got the total sales per month. I created two pivot tables and two bar graphs to illustrate the count of each item per month and total sales of each item per month. I added another two pivot tables and two bar graphs to show the trend of quantity sold of each item for six months and the sale value trend of each item for six months, which shows how good the product performs through six months.

For the 'Customer', I removed the day in the transaction dates after loading the json file into dataframe 'boom'. I then created a new column in 'boom' called 'order_id' to keep track of which transactions are together before splitting them later. I got the unique values from the transaction dates and made them into an array. I created a pivot table for the count of each customer's transactions per month (bduserdf). I created a statement to return the top customers with highest transaction total through .sort_values. I created a while loop with nested if statements inside to categorize whether the customers are new (first time buyers), repeaters, inactive, or engaged. I then store them into 'frequency_df'. I outputted three pivot tables: a. each

customer’s transactions for every month; b. the top ten customers with the highest total transactions; and c. the customer frequency metrics per month. Also, I made a bar graph to illustrate the top ten customers with the highest total transactions. Lastly, I made five line graphs with four of them being repeaters, inactive, engaged, and new and the last one is the combination of the four customer metrics.