

Document Explaining My Approach To The Problem

For both the monthly sales and loyalty metric, which were solved using two different notebooks, I first loaded the JSON file into Python and store it as a Pandas DataFrame. I looked into the contents of the data first and how it would look like in the DataFrame.

For the monthly sales per item, I first parsed the 'transaction_items' column of the DataFrame to separate the items into a list. Then, I used the `.explode()` method to turn the list into rows, which made it easier for me to extract the details of the items sold. The brand and the item type were grouped together, and the quantity sold for each item was separated and converted to an integer using regex. These were stored in a different DataFrame to make it easier to manipulate later. Each of the items was then attributed to a transaction date. To identify the unit price of each item, I simply checked transactions with single item purchases with a quantity of one. I then used the `.unique()` function to only get the unique items and it's unit prices. These were solved using two DataFrames and were subsequently merged using SQL methods. Each item purchased from a specific month were then added to the DataFrame that contained the unique items and their unit prices. To get the total sale per item per month, the unit price was simply multiplied to the quantity sold of each item per month. This resulted in a table that consolidated both the quantity sold and sales value per month. They were then separated using pivot tables, wherein graphs were made using both Matplotlib and Seaborn. Seaborn was used to add a different color palette to the graphs.

For the loyalty metric, I made a new DataFrame to store the customer names and the transaction dates. The transaction dates were converted into DateTime to make it easier to work with. I then added a new column in the same DataFrame called 'transaction_count' which basically had integer values of 1. This made it easier for me to count for the number of times a customer bought in a month since there was no count function in the pivot table. The month of transactions were also extracted to make the pivot table. From the pivot table, I created a truth table in a different DataFrame, and then I added a column that consolidates the Boolean values into a list, which was easier to compare using a vectorised function. After applying the conditions for each metric, the numbers were summed up for each of the three metrics and placed in a DataFrame, which was then used to create a bar graph.