Recommendation Engine and Receipt Analysis Dashboard

Public Project Summary

By Maryam Khan

Submitted to
Central European University
Department of Economics and Business

Master of Science in
Business Analytics

June 2022

Table of Contents

Objective

The main objective of this project is to help the client increase sales and customer satisfaction. I will be achieving this by data mining and drawing associations between items by using the client's data and conducting unsupervised machine learning. By doing this the client can optimize product placement, offer special deals and create new product bundles to further increase sales with these combinations. This will also help in optimizing marketing strategies and campaigns as we will be able to understand the customer better by identifying customer behavior and patterns.

Even though there are general tools available for receipt line analysis for the client, there is no goal-oriented tool for this that focuses on receipt analysis particularly looking at items for the business team to make such decisions.

As a part of this project, I was required to build a recommendation engine for the procurement team. That allowed the user to input an item a number and gave an output in form of a list of products that are frequently bought together by the customers. I was also required to build a dashboard which displayed some insights about the receipt data where the user can filter for item number and analyze trends. As the first phase of the project, I built the recommendation engine once this was approved by the client, I started to work on the second part of the project which consisted of the dashboard.

Data Cleaning

The two main tables I used were the bonpos table and the item meta data table. The bonpos table consists of all the receipt data pertaining to the clients store and the item meta table has the item numbers with item name and item family. The data was stored in the client's database on Databricks, and my entire project was done on it using a combination of Python and SQL. Python

was used for majority of the work and SQL was used for dashboard creation. As part of my project, I filtered the data and conducted my analysis on only a specific number of stores and for a specific time period.

After doing some exploratory data analysis I figured that the receipt ID was not unique and I created a unique ID by concatenating Register ID, Receipt ID, Receipt date and store number.

Transforming the data

The final data frame had to be transformed into a specific data type for me to be able to carry out the FP growth analysis. Each transaction should be in a single row with the items bought in that transaction listed next to it. This was achieved by grouping the data by the unique ID and then removing all duplicates as the same item can be bought multiple times. In this case, we are not concerned with the number of items purchased instead we only care if the item was in the basket or not. The next step was to collect all the items that were purchased into a single list to have a single row for each transaction.

Data mining using FP growth algorithm

The model I used for this is the FP growth algorithm. It is the one of the most sophisticated models for frequent item set mining also known as the Association Rule Mining and basket analysis.

Basket analysis gives recommendations and insights has to what products in the retail stores should be placed alongside or be used together for bundled promotions as they are frequently added into one basket together. Frequent item mining is the frequency of items that are frequently bought together. The main purpose of this algorithm is to help identify products that occur in different combinations together using a fast and efficient algorithm.

Inputs for the model:

The 3 hyper parameters of evaluation for the FP growth algorithm are as follows:

1. Minimum Support: The minimum support for a set of items is the minimum threshold that it requires to be classified as frequent. For instance, if the item appears in 3 out of 5 transactions, then the minimum support would be 3/5 = 0.6
2. Minimum Confidence: The minimum confidence is the parameter that shows how many times the association rule has been found to be true in the data set. For example, an item appears in the dataset 4 times it appears with another item 2 times then the minimum confidence for this association rule is 0.5. The confidence parameter does not affect data mining instead it defines the minimum threshold for generating the association rule. A high value of confidence does not necessarily mean that there is high association between those items for this purpose we look at the lift value.
3. Lift: The lift gives us the strength of association between 2 items. If there is strong association between 2 products, then the lift value is greater than 1. The higher the lift value between 2 products the greater are the chances that a product is already present in the customers cart they will also purchase the second product. This value can help retailers decide the product placement in stores.

For the FP growth input parameters of minimum confidence and minimum support I tried and tested multiple values. The parameters I set for this were as mentioned below:

- Minimum Confidence: 0.001
- Minimum Support: 0.001

The model produces a list antecedent (product that is bought before) and consequents (product that is bought after) along with the support, confidence and lift parameters for each pair.

A recommendation engine is a product was created using the model results that basically helps retailers to suggest products that are frequently bought together by their customers. Using a recommendation engine, the retailer can act in real-time and increase sales by recommending products together. The recommendation engine uses the association rules that are generated from the algorithm and finds the consequent for the antecedent that is selected by the customer.

Dashboard

As the second part of the project, I was expected to create a dashboard that showcased some basic analysis for items along with the recommendation engine. The client requested that I add filters so that the business team can filter for their preferred item and see the analysis of those selected items. The purpose of this dashboard is that the business team can use this to analyze and track the performance of their items and item families. They see the sales trends and the number of sales and plan using this for the upcoming months regarding the future promotions or product placements of those items.

The dashboard was built using the SQL editor in Databricks. A separate query was created for each individual graph with filters so that the user can filter the graph according to their preference. I created separate tables in the data science and engineering section and saved them to the database so that they could be used in the SQL editor to create graphs. I added extra columns like days of the week for in the tables for visualization.

The following visualizations were included:

1. Recommendation Engine – This is a user-friendly table that is generated using the FP growth algorithm the user can filter the table for specific item numbers

2. Sales Trend – This shows the number of times an item was bought everyday throughout the month of March. It is a line graph that the user can filter item number from the drop-down menu and see the sales trend for that item in the month of March.
3. Sales in the day of the week – This is a bar chart that shows the number of times, and an item is bought on a particular day of the week. The user can filter this graph by select an item number and it display sales for that item.
4. Top 20 items by item family – The bar chart shows top 20 items in particular item family for the month of March. The user can filter for item family number and the chart will show the top 20 items sold in that selected item family.
5. Top item family – This is a horizontal bar chart that shows the top 20 item families sold.

Limitations

I faced multiple challenges during the project, some of them have been outlined below:

1. Since I was working with daily receipt data the number of rows were a lot and because of this the computational time for each command to run was at least 30 minutes and this made my work super slow as the cluster size was small.
2. One of the main limitations of using such algorithm is that they do not consider the quantity of the items that bought hence the weight is not considered when running algorithms like FP growth and Apriori.
3. Lack of experience using Databricks
4. Basic python knowledge had to learn pyspark

Way Forward

To further build up on the project we can add more stores as part of the analysis and add a filter for stores as well to see in which areas certain products are popular and in demand. Currently, the analysis is only focused on stores that are in the Budapest city center expanding the analysis to different areas will also help in giving a good comparison as to where the product demand is more. Another feature that we can add is that conduct the data mining exercise for different months of the year to compare the seasonality of the items.

Conclusion

The purpose of this project was to create a recommendation portal with some basic receipt analysis for the business team**.** I created a pipeline by going through a funnel of transformations to make sure that the data was in the correct format for me to implement the FP growth algorithm. The algorithm gives very useful insights regarding item association that the business team can use to make informed decisions when it comes to item placement and promotions.

The data mining insights are very useful and can help the client grow their business in a cost-effective way as it helps them to understand their customers better and as a result it helps them in making strategies according to their customer preferences.