Zepto Recommendation Case Study

Submitted by -

Name - Kumar Anant
Roll - 22MA10030

Link to my Jupyter Notebook: Click Here

1. Understanding the Problem

The goal of this case study is to develop a recommendation engine that suggests more products to customers based on the items currently in their shopping cart. In the fast-paced world of quick commerce, enhancing the cart with relevant recommendations can significantly improve the shopping experience.

The objective is to :-

1. **Provide Faster cart creation**
2. **Enhance the Net Value of the Cart**

Description of Data Provided: The data consists of two columns: **session_id** & **product_name**. session_id column is a string data type representing the session ID of each product purchased. product_name is also a string data type representing the name of the product purchased.

Approach Overview: Since the data available with us has only transaction records, I used a recommendation system based on **Association Rule Mining**, specifically utilizing the **Apriori algorithm**. The transactional data is used to identify frequent itemsets. (sets of products bought together) and extract strong association rules. These rules will form the basis for recommending complementary products to customers.

2. Data Exploration and Preprocessing (EDA)

The CSV file is loaded into DataFrame.
Data Cleaning: Checked for NULL values if present in the data following are the results :-
session_id : 0 Null Values
product_name : 0 Null Values
There are no NULL values, so there is no need to handle it.

transactions list is created to store all purchased items of each session.

example:

Data given

Session Id	Product
1	t1
1	t2
2	t1

transaction list

Session Id	Product
1	t1, t2
2	t1

The following are some useful values fetched from data :-
Total number of transactions (sessions) settled: len(**transactions**) = **165335**
Total number of distinct items purchased in all transactions given in data: 396

A histogram of the number of purchases per item is plotted. Click here to view
Some of the **Most Bought items** are obtained from the Histogram plot:-

1. onion
2. cucumber
3. fresh cow milk
4. beetroot
5. curd
6. gourds

etc.

So it seems beneficial for the Business to try to keep above items always available in Stock.

The average no. of products people buy from the store is **~2**
The maximum number of product ever buyed in a session is **9**

3.Recommendation Engine Design: Association Rule Mining with Apriori

Association Rule Mining: The vast amount of transactional data is analyzed, looking for clues about which products tend to appear together. This "togetherness" is crucial, as it suggests a potential complementary relationship.

Apriori: Apriori is one of the leading algorithms for association rule mining. It's like a sophisticated search engine that efficiently scans through our data, systematically identifying frequent itemsets. The idea is to operate candidate itemsets of a given size and then scan the data to find the most frequent ones.

Support: This is a metric used to represent the frequency of itemsets that are bought together in the data. This also **tunable** according to the results we obtain.

For Example, let **min_support = 2**

transaction list

Session Id	Product
1	t2, t3
2	t1, t3, t5
3	t1, t2, t4
4	t2, t3

itemsets of size 1

itemset	Support
{t1}	2
{t2}	3
{t3}	3
{t4}	1
{t5}	1

itemset {t5} & {t4} is discarded because **min_support = 2**

itemsets of size 2

itemset	Support
{t1, t2}	1
{t1, t3}	1
{t2, t3}	2

itemset {t1, t2} & {t1, t3} is discarded because **min_support = 2**

One thing to note is that support for each itemset is calculated as the no. of occurrences of that itemset in transaction list

Rules: A rule is the transition map from A to B.
In above example, {t2, t3} is a frequent itemset so the rules genrated as follows.

Rules will be :

$$\begin{tikzcd} & \{t2, t3\} \arrow[dl] \arrow[dr] & \\ \{t2\} \to \{t3\} & & \{t3\} \to \{t2\} \end{tikzcd}$$

This suggests that if someone has bought t2, then it should be good to suggest t3 as well.

This suggests that if someone has bought t3, then it should be good to suggest t2 as well.

But NOT all rules are important enough to be suggested that why we calculate the
The **confidence score** for each rule tells us how likely someone is to buy a specific product given that they already have other products in their cart.
it is the calculated by using formula below:

$$\textbf{Confidence score of A -> B} = \frac{\text{Support}(A \cup B)}{\text{Support}(A)}$$

One another useful metric used to evaluate how powerful the rules are is the **lift score**.
Lift: it measures how much stronger the association between two products is compared to what we'd expect if they were independent. It essentially tells us if the products are being bought together more often than we would expect based on their individual popularity.
it is calculated as follows:

$$\textbf{Lift score of A -> B} = \frac{(\text{Support}(A \cup B) \ / \ N)}{(\text{Support}(A) \ / \ N) * (\text{Support}(B) \ / \ N)}$$

N : no. of transaction in the transaction list
Support(A ∪ B) : Support of A union B

- **Lift > 1**: Indicates a positive association. The products are bought together more often than expected by chance. This suggests a strong complementary relationship.
- **Lift = 1**: Indicates no association. The products are bought together at a rate expected based on their individual popularity.
- **Lift < 1**: Indicates a negative association. The products are bought together less often than expected by chance. This might indicate that one product actually *decreases* the likelihood of buying the other.

4.Evaluation and Results

Overall, 3 Metrics are used to Build and Tune the performace of model :
Support, Confidence & Lift

In Jupyter Notebook,
I have kept my **minimum support** threshold = 0.005*165335 = **82**
Threshold for **Lift = 1**

Total Rules generated = **390**
(while this can be increased or decreased, using the above 3 tunable parameters, based on the **performance**, **outcomes** & **effects** of the recommendation model on the business or store)

Following are some recommendation that are generated by the model:-

Cart Items	**Reccomendations genarated by model**
Match box	agarbatti & incense sticks
cucumber	garlic
	curd
	beetroot
	gourds
	watermelon

Cart Items	**Reccomendations genarated by model**
almond	cashew
	raisins & kismis

Another thing that can be done is that the no. of recommendations can be limited to a number using another max recommendations threshold.

5.Discussion and Further Improvements

- **Scope for further improvement:** Since the given data field is limited to only two columns, if we would have given with more fields like:-
  - **Time for each transaction**: Getting the Time for each transaction can help more in performing EDA and in analyzing & keeping track of the performance of Business or Store, such as average speed of transactions settled by the business or store, Busy hours of Bussiness or store, and more.
  - **Customer Demographics**: Understanding customer age, location, and preferences can help us personalize recommendations and target specific product bundles.
  - **Product Categories**: Categorizing products can help us identify broader complementary relationships, like suggesting spices when a customer buys a main ingredient.
- **Other algoithms:** If we would have been given with more data fields then we can also analyze the performance of other algorithms like :-
  - **Collaborative Filtering**
  - **Advanced Apriori Variants and more**

-- end --