**Experiment 4**

**Name: SAP ID:**

|  |  |
| --- | --- |
| **Date:** | **04/10/2023** |
| Aim | **Build recommender system using association rule mining.** |
| Software | Colab |
| Pre-requisite | Internet and required dataset |
| Theory | **Introduction**  While browsing any eCommerce platform, you must have encountered the frequently bought together section. Many businesses add this section to improve sales, which leads to increased income.  Market Basket Analysis, also known as Association analysis, is a method for understanding client purchase trends based on historical data. In other words, Market Basket Analysis enables merchants to find links between the products that customers purchase  **Dataset Description**  The dataset is a transnational data collection covering all transactions made by a UK-based and registered non-store internet retailer between 2010 and 2011. The firm primarily distributes one-of-a-kind all-occasion presents to wholesalers.  The dataset includes information on 500K clients across eight attributes. You can download the dataset here.  The eight columns or features are:  InvoiceNo: The invoice number of a particular transaction.  StockCode: The unique code for an item.  Description: The description of a specific item.  Quantity: The quantity of an item bought by the customer.  InvoiceDate: The date and time when the transaction was made.  UnitPrice: The price of 1 unit of an item.  CustomerID: The unique id of the customer who bought the item.  Country: The country or region of the customer.  **Let's install the dependencies:**              We can observe the top 5 loyal customers with the most orders.                   Free Items and Sales This section will show how free products affect order volume and how discounts and incentives affect sales:                          plt.ylabel('Item')           Frequently Bought Together Market basket analysis can be defined as a data mining approach. Businesses employ it to enhance sales by understanding client purchase habits better. It includes evaluating massive data sets, such as purchase history, to uncover product groups and products likely to be purchased together.  This section will answer the question:   * Which items are frequently bought together?   We will use the groupby() function to create a basket that specifies if an item is present in a particular invoice number.  We will get the quantity present in the specific invoice number, which must be fixed.  Note that this is for all items and invoices:     Apriori Algorithm Concepts Following this [link](https://www.kdnuggets.com/2019/12/market-basket-analysis.html), we can read this: Apriori algorithm assumes that any subset of a frequent itemset must be frequent. It's the algorithm behind Market Basket Analysis. Say, a transaction containing {Grapes, Apple, Mango} also has {Grapes, Mango}. So, according to the principle of Apriori, if {Grapes, Apple, Mango} is frequent, then {Grapes, Mango} must also be frequent. What is Support, Confidence, Lift, and Association Rules **Support** is the ratio of A-related transactions to all transactions. If out of 100 users, 10 purchase bananas, then support for bananas will be 10/100 = 10%. In other words: Support(bananas) = (Transactions involving banana)/(Total transaction).  **Confidence** divides the number of A and B transactions by the number of B transactions. Suppose we are looking to build a relation between bananas and tomatoes. So, if out of 40 bananas buyers, 7 buy tomatoes along with it, then confidence = 7/40 = 17.5%.  **Lift** is an increased sales of A when selling B; it is simply the confidence divided by the support: Lift = confidence/support. So, here lift is 17.5/10 = 1.75  **Association rule** mining finds interesting associations and relationships among large sets of data items. This rule shows how frequently an item set occurs in a transaction. Based on those rules created from the dataset, we perform Market Basket Analysis.  In the below program:   * apriori() function returns a list of items with at least 15% support. * association\_rules() function returns frequent itemsets only if the level of lift score > 1(min\_threshold=1). * sort\_values() function sorts the data frame in descending order of passed columns (lift and support).     If we have a rule "B ->D", B stands for antecedent, and D stands for consequent.  **Leverage** is the difference between the observed frequency of B and D occurring together and the frequency that would be expected if B and D were independent.  **Antecedent support** computes the fraction of transactions that include the antecedent B.  **Consequent support** computes the support for the itemset of the consequent C.  **Conviction**: A high conviction value indicates that the consequent strongly depends on the antecedent.  Now, we will create the function in which we pass an item name, and it returns the items that are frequently bought together:      We can have a view of the items which are commonly purchased together. |
| Dataset | Link: <https://archive.ics.uci.edu/dataset/352/online+retail>  Link: <https://thepythoncode.com/article/build-a-recommender-system-with-association-rule-mining-in-python> |
| Conclusion | Your understanding about experiment. |