THE UNITED REPUBLIC OF TANZANIA

MINISTRY OF
EDUCATION, SCIENCE AND TECHNOLOGY

DAR ES SALAAM INSTITUTE OF TECHNOLOGY

DATA MINING AND ANALYTICS (MODULE COU 08104)

ASSIGNMENT 4 (BENG22COE & BENG22ETE)

8th February 2026

INSTRUCTIONS

- **This is an individual assignment.** Students are required to complete it independently.
- **Submissions** should be made via GitHub. I will compute a grade based on the state of your repo ***as of the submission deadline***, so feel free to make changes to it up to that point.
- Submit a Python code file (. ipynb), [Do not Submit checkpoints for .ipynb] to **GitHub**
- Submit a single zip file containing all csv dataset files used and Python code file (.py) to **Gradescope**.
- Do not commit inoperative codes. You should document or comment on your codes professionally, this demonstrates that the code is unique and owned by the student, but avoid unnecessary comments, you should not write comments on obvious codes.
- **All import statements** must appear together in a single cell, or in consecutive cells at the top of the ipynb file**, before any other executable code**. ***[Note that in this assignment we will be using python Surprise package]***
- Submitting answers directly from generative AI is not allowed as it could lead to being flagged for plagiarism and you will earn 0 score for the assignment.
- You should create a directory named ***Assignment04*** in your GitHub repository and your solution file should reside in this directory. The file MUST be named the same as your registration number, e.g., “230454545669.ipynb”.

The submission deadline is Sunday 22nd, February 2026 11:59 PM EAT.

1. Phone Faceplate Transactions - (9 points)

Association rules, or affinity analysis, constitute a study of "what goes with what." This method is also called market basket analysis because it originated with the study of customer transactions databases to determine dependencies between purchases of different items. A store that sells accessories for cellular phones runs a promotion on faceplates. Customers who purchase multiple faceplates from a choice of six different colors get a discount. The store managers would like to know what colors of faceplates customers are likely to purchase together. Support of a rule is the number of transactions that include both the antecedent, and consequent itemsets. The confidence of a rule compares the co-occurrence of the antecedent and consequent itemsets in the database to the occurrence of the antecedent itemsets.

1.1. Load the phone faceplate dataset (Faceplate.csv) and display the first 10 transactions.

1.2. What is the support of the itemset {red, white} in the faceplate dataset?

2. Apriori Algorithm Application (20 points)

Several algorithms have been proposed for generating frequent itemsets, but the classic algorithm is the Apriori algorithm of Agrawal et al. (1993). The key idea of the algorithm is to begin by generating frequent itemsets with just one item (one-itemsets) and to recursively generate frequent itemsets with two items, then with three items, and so on, until we have generated frequent itemsets of all sizes. It is easy to generate frequent one-itemsets. All we need to do is to count, for each item, how many transactions in the database include the item. These transaction counts are the supports for the one-itemsets. We drop one-itemsets that have support below the desired minimum support to create a list of the frequent one-itemsets. To generate frequent two-itemsets, we use the frequent one-itemsets. The reasoning is that if a certain one-itemset did not exceed the minimum support, any larger size itemset that includes it will not exceed the minimum support. In general, generating k-itemsets uses the frequent $(k - 1)$-itemsets that were generated in the preceding step. Each step requires a single run through the database, and therefore the Apriori algorithm is very fast even for a large number of unique items in a database.

2.1 Using the phone faceplate dataset, apply Apriori algorithm with a minimum support of 0.2 (20%) for frequent itemsets creation. Display all frequent itemsets with their support values.

2.2 From the frequent itemsets you generated above, generate association rules using a minimum confidence threshold of 0.5 (50%). You can use the association_rules function from mlxtend.frequent_patterns. Display the rules sorted by lift ratio in descending order.

2.3 For each of top 6 rules (by lift ratio), display the following, antecedents, consequents, support, confidence, lift and leverage. Drop the following columns 'antecedent support', 'consequent support', and 'conviction' from the output.

2.4 Translate rule with the highest lift ratio into an a sentence that can well be understood. For example: "If [antecedent items] are purchased, then with confidence X% [consequent items] will also be purchased. This rule has a lift ratio of Y."

3. Book Purchase Association Rules (15 points)

The Charles Book Club database contains 2000 transactions of books purchases. With 11 different types of books. This database is in the form of binary incidence matrix, where each row represents a transaction and each column represents a book type, with 1 indicating purchase and 0 indicating no purchase.

3.1 Load Charles book club dataset (CharlesBookClub.csv) into Python then create binary incidence matrix by selecting only the book-related columns, (ignores 'Seq#', 'ID#', 'Gender', 'M', 'R', 'F', 'FirstPurch', 'Related Purchase', and any code columns). Subsequently, convert all values which are greater than 0 to 1. Display the first 10 rows of the binary matrix.

3.2 Apply the Apriori algorithm to the book purchase data, with minimum support of 200 transactions (5% of 4000 total transactions), Generate frequent itemsets and display the number of frequent itemsets found.

3.3 From the frequent itemsets, generate association rules with a minimum confidence of 0.5 (50%). Display the 25 rules with the highest lift ratios, showing: antecedents, consequents, support, confidence, lift, and leverage.

4. Interpreting Association Rules Results (21 points)

In interpreting association rules results, data miners use to look at various measures. The support for the rule indicates its impact in terms of overall size. Questions like, how many transactions are affected? may arise. If only a small number of transactions are affected, the rule may be of little use (unless the consequent is very valuable and/or the rule is very efficient in finding it). The lift ratio indicates how efficient the rule is in finding consequents, compared to random selection. The confidence tells us at what rate consequents will be found, and is useful in determining the business or operational usefulness of a rule.

4.1 Using book purchase association rules from Question 3, select the rule with the highest support value. what are the antecedents, consequents, support, confidence, and lift ratio for this rule?

4.2 From the above question, select the rule with the highest lift ratio from the book purchase rules. Compare its support value to the rule with highest support. Discuss the trade-off between a very efficient rule (high lift) that has very low support versus a less efficient rule with much greater support.

4.3 Identify the top 10 rules by lift ration. Among this top 10 rules by lift ratio, identify rule with the lowest confidence value.

5. Association rules and chance effects (15 points)

When assessing possible chiselling, or false causation from the coincidence,when evaluating possible assocation or correlation rules we need to concentrate on whether or not the rules are qualitatively sound/Interesting strong rules, or are they just a product of chance. There are two principles that will assist in evaluating the quality of rules when determining if they are possible chiseling or coincidence as a result of sampling error: (1) The greater the number of instances upon which the item associations or rules are derived, the stronger the conclusions will be , and (2) The more distinct associations being involved in rule creation, the greater the probability that there will be a certain portion of those rules which arise from a sampling error.

5.1 Create a synthetic dataset with 50 transactions; 9 items per transaction and randomly assigned items per transaction, use random.seed(0) for repeatability. To create this dataset create a binary incidence matrix where rows represent each transaction and columns represent the items (1-9).

5.2 Run the Apriori algorithm on the randomly generated data with Minimum Support (MS) set to 2 transactions or 4% of total transactions (50) Find the frequent itemsets. Create the association rules from the frequent items at a minimum of confidence (CI) of 0.7 (70%).

5.3 Find the 6 rules with the best uplift opportunities when sorting by the uplift ratio (descending). Identify and display: antecedents , consequents , support , confidence and uplift. Even though the data you created was generated through a random process, did you find any rules which had an exceptionally high uplift ratio?

6. Item-Based Collaborative Filtering (20 points)

When the number of users is much larger than the number of items, it is computationally cheaper (and faster) to find similar items rather than similar users. Specifically, when a user expresses interest in a particular item, the item-based collaborative filtering algorithm has two steps: (1) Find the items that were co-rated, or co-purchased, (by any user) with the item of interest. (2) Recommend the most popular or correlated item(s) among the similar items.

Similarity is now computed between items, instead of users. This can be done offline. In real time, for a user who rates a certain movie highly, we can look up the movie correlation table and recommend the movie with the highest positive correlation to the user's newly rated movie. According to an industry report by researchers who developed the Amazon item-to-item recommendation system, "[The item-based] algorithm produces recommendations in real time, scales to massive data sets, and generates high-quality recommendations." The disadvantage of item-based recommendations is that there is less diversity between items (compared to users' taste), and therefore, the recommendations are often obvious.

6.1 Using random seed=0, create synthetic ratings dataset with 5000 ratings of the following attributes, itemID (ranging from 0-99), userID (ranging from 0-999), and rating (ranging from 1-5). Print first 10 rows of the synthetic dataset you created.

6.2 Convert the data set into the format required by the surprise package. The columns must correspond to user id, item id, and ratings (in that order) What are the dimensions of the training and test sets?

6.3 Compute cosine similarity between users. Build an item based collaborative filtering model using cosine similarity.

6.4 Predict ratings for all pairs (u, i) that are NOT in the training set. Print the recommended items for each user