Steps in Apriori Algorithm:

1. Apply appropriate data cleaning to make data in unified format.
2. Check for duplicate data and remove if required.
3. Remove null or bad data.
4. Reset index
5. Gather all items of each transaction into numpy array
6. Create new table to get frequency count of each item
7. Visualize top10/top30 most bought items using the above table.
8. Using transaction encoder from mlxtend library, encode the data
9. Use apriori library from mlxtend library on the encoded dataset, with suitable threshold/min\_support.
10. Use the above dataset from apriori to create association rules with threshold = min\_lift, which should be greater than 1.
11. Sort this association rules dataset by lift.
12. Predict for a given antecedent:
    1. Extract rule for the particular antecedent
    2. Get the top few consequents for this antecedent from the association rules

<https://www.kdnuggets.com/2016/04/association-rules-apriori-algorithm-tutorial.html>

Steps in Hybrid Recommendation:

* Collaborative Filtering:

1. Do some basic data cleaning on text based columns like converting to lowercase, removing numbers or other formatting, so that all the data is in the same format.
2. Filter out the common data(remove rare items : those items that have been least frequent)
3. Create new data frame with index as user id and columns as title of data with values of the given data(it is a sort of encoding based on different titles)
4. Select a random user from this new dataframe. We will recommend items to this user based on his data and similar activity users.
5. Extract the items that he has interacted with, and create a new dataframe with only these items as columns. We will filter out users who have interacted with these items only.
6. Extract all the users who have interacted with at least one of these items.
7. Take a count of all items these users have interacted with(items that the given user had interacted with).
8. We only need users who have interacted with at least 60% items. So we will filter them out.
9. Combine the given user with these users in a new dataframe.
10. Calculate correlation between each user and store it in a dataframe.
11. From this dataframe extract all users who were correlated with our given user by at least 60%. These are the users whose items we will recommend to the given user.
12. Calculate weighted data for each user and his items(Weighted rating = Correlation \* Data)
13. Take the sum of correlation and sum of weighted data for each of the items.
14. Calculate weighted average recommendation score = Sum of weighted data / Sum of correlation for each of these.
15. Sort this data in descending order and recommend top 5-10 items with the highest score.

* Content Based Filtering:

1. Using item-metadata to recommend similar items who have similar metadata
2. Combine all the metadata of the items into one column
3. Make this column into a unified format, i.e. keep all strings in lowercase, check for number formatting, etc.
4. Create a countvectorizer for this dataframe
5. Pass this countvectorizer to cosine similarity and create a similarity matrix
6. Here, look for a particular item in the rows and for each column check which has max similarity, take the top 10 elements having the highest similarity and recommend these items.