Project #1 – Principles of Data Mining – Report

Team ID: **11**

Team members: Daniel Hernandez Vega & Aaliyah Vincent

Answer the following questions. Your answers should be in dark blue. The plots requested should be in this file. Do not submit them as separate files.

Question 0

How did each member of the team contribute to the project?

Daniel led the algorithm development, ensuring the model accuracy and efficiency, while Aaliyah focused on data visualization and compiling the final report.

Cite any sources that you used to complete the project.

We used Pandas, matplotlib, & seaborn libraries for data analysis and visualization.

Question 1

Describe in detail the data structures that you used (how you stored the data) for:

a. the transactions:

the data structure containing transactions is a list of sets with each set representing a transaction.

b. the frequent itemsets:

the data structure containing the frequent itemsets is a dictionary. the dictionary keys are the length of the itemsets, the dictionary values is a frequent itemset dictionary. The frequent itemset dictionary keys are the frequent itemsets and values are the support counts of the respective itemset

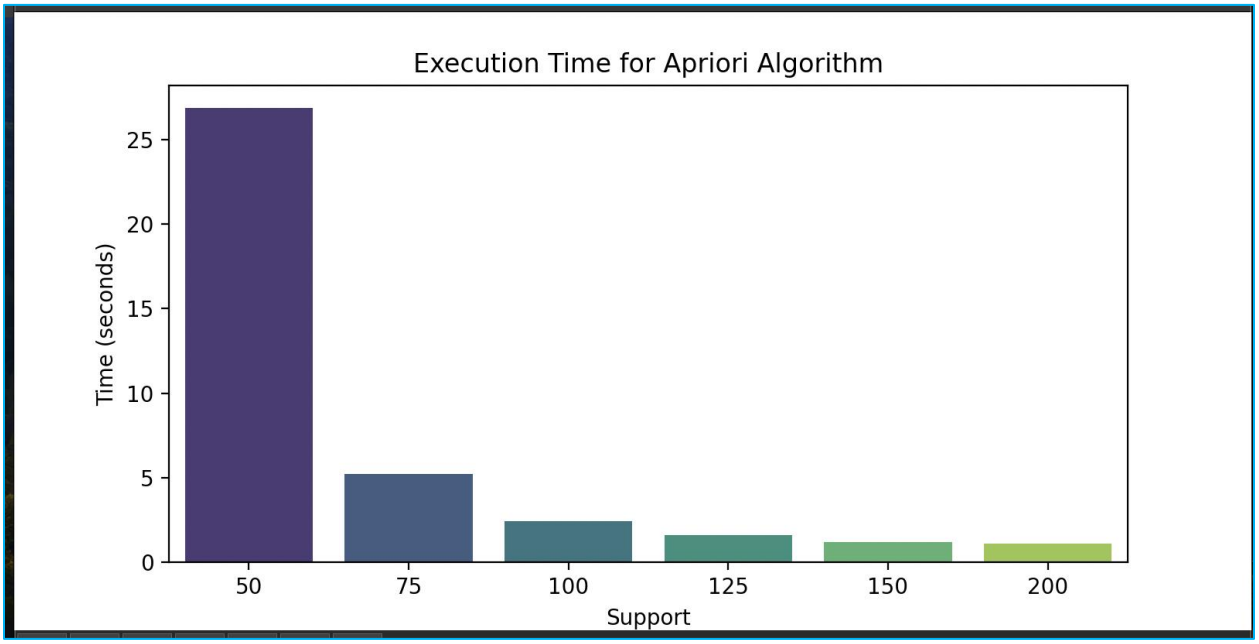
c. rules generated:

the data structure containing the rules is a list of tuples each tuple containing 3 elements. The first two elements of the tuple represents each side of the rule implication respectively, and the last element represents the rules confidence

Question 2

Run your code for *minconf=0.8* and the following values for the *minsup = {50, 75, 100, 125, 150, 200}* (note that this is the support count). Plot the amount of **time** required to generate the frequent itemsets for the different values of minimum support. Make sure that each axis is properly annotated with the quantity that it corresponds to.

<plot>



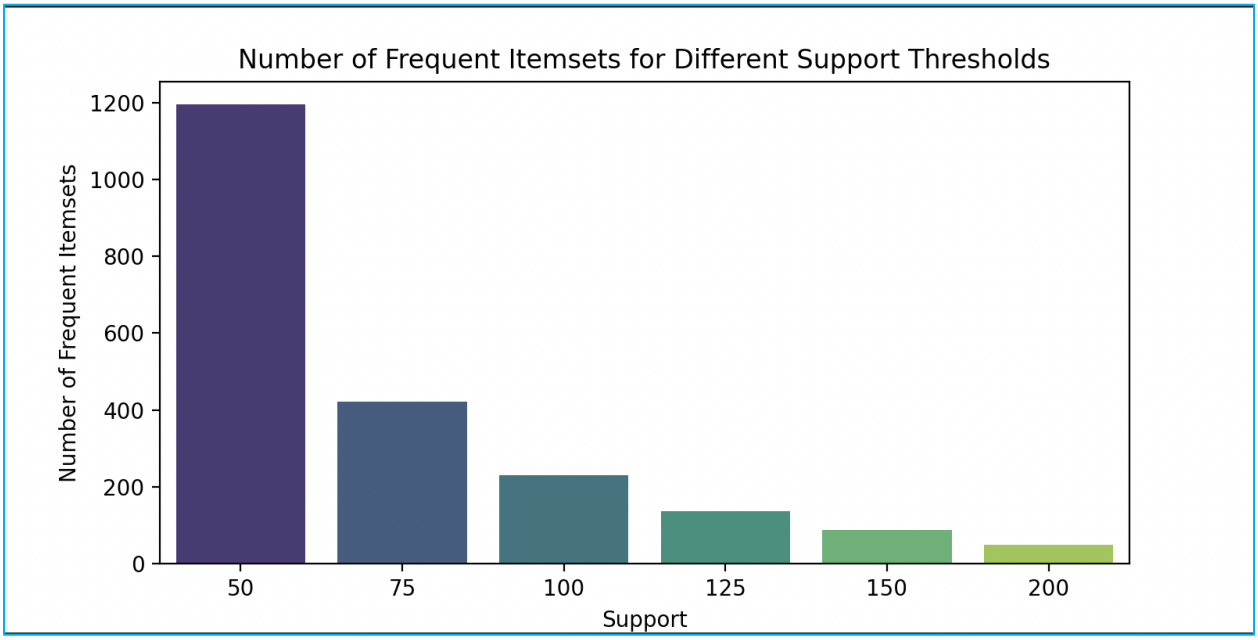
What can you understand from the plot?

From the plot, I understand that the execution time for the Apriori algorithm decreases significantly as the minimum support value increases. At a minimum support of 50, the execution times is the highest, which is over 25 seconds, whereas at minimum support of 200, the time is considerably lower, close to zero. This trend suggests that lower minimum support thresholds require more computation, leading to longer runtimes, while higher minimum support values speed up the algorithm by pruning infrequent itemsets early.

Question 3

Run your code for *minconf=0.8* and the following values for the *minsup = {50, 75, 100, 125, 150, 200}* (same as in Question 2). Plot the **number of frequent itemsets** generated for the different values of minimum support. Make sure that each axis is properly annotated with the quantity that it corresponds to.

<plot>



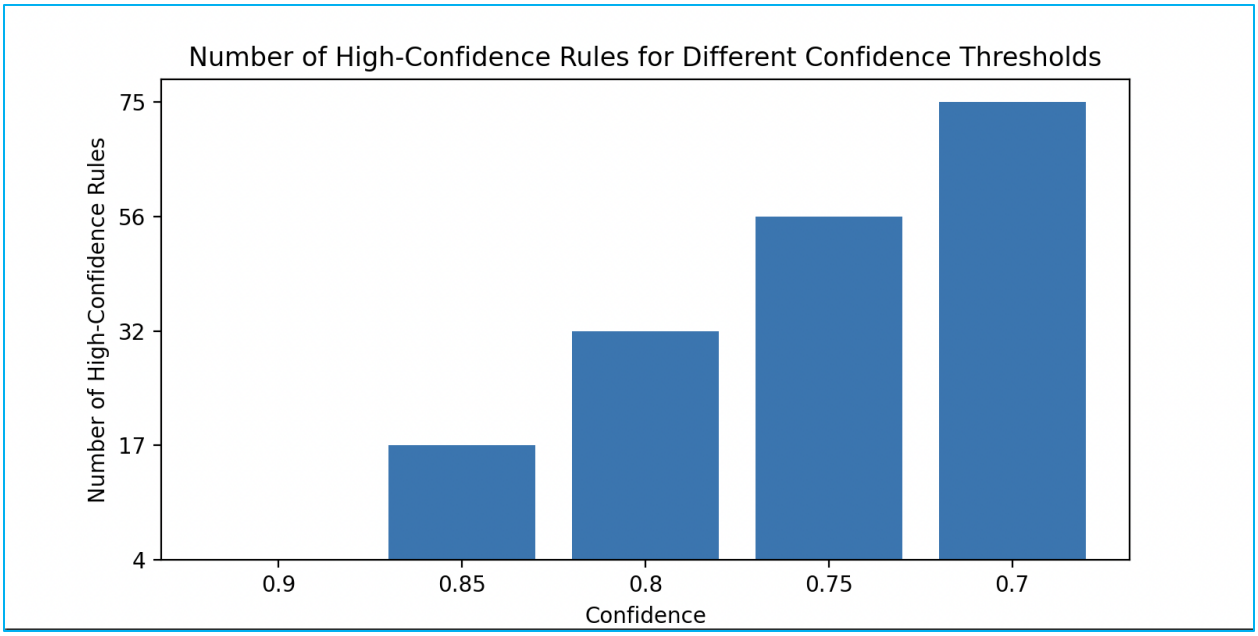
What can you understand from the plot?

From the plot, I understand that the number of frequent itemsets decrease as the minimum support increases. At a minimum support of 50, the number of frequent itemsets is the highest, reaching 1,200. However, as the minimum support increases, the number of frequent itemsets drops significantly. By the time the minimum support reaches 200, we see that very few frequent itemsets remain. The steepest decline occurs between a minimum support of 50 and 75, suggesting that many items exist just above the lower threshold. After minimum support threshold of 125, the number of frequent itemsets almost flattens out, showing only a small core set of highly frequent items that remain.

Question 4

For *minsup=80*, generate the rules at different levels of confidence, i.e., *minconf={0.7, 0.75, 0.8, 0.85, 0.9}*. Plot the **number of rules** that were found for the different values of minconf threshold. Make sure that each axis is properly annotated with the quantity that it corresponds to.

<plot>



What can you understand from the plot?

From the plot, I observe that the number of high-confidence rules decreases as the confidence threshold increases. At a confidence level of 0.7, the highest number of rules is generated, reaching 75. As the confidence threshold rises, fewer rules meet the stricter criteria, leading to a noticeable decline in the number of high-confidence rules. The most significant drop occurs between confidence levels of 0.8 and 0.85, suggesting that many rules exist just below this threshold. By the time the confidence reaches 0.9, there are no rules remaining. This pattern shows the balance between the number and strength of rules, but some may be less reliable. Higher confidence thresholds result in fewer rules, but they are stronger and more reliable.

Question 5

Submit the files produced when minsup=140 and minconf=0.8.