Part 2a.

The Apriori algorithm was designed to provide predictive capabilities through the analysis of k-itemsets, looking for items frequented together to formulate predictions within specific parameters. This method iteratively scans a given set of data and creates k number of itemsets, depending on what is specified. While this method could become computationally expensive depending on the size of the dataset, there are techniques employed to reduce the number of transactions being scanned. For example, the Apriori property will ensure a specified minimum threshold is met by a given itemset, otherwise it will not be scanned. Additionally, antimonotonicity will ensure all supersets of that itemset are not scanned either. Once frequent itemsets are collected, rules can be created based on a few different metrics, namely support, confidence, and lift. The support percentage threshold highlights the frequency of the antecedent itemset. The confidence percentage threshold tells us the frequency percentage of the antecedent and consequent itemsets together. Lift tells us the correlation between both itemsets. If lift is exactly 1, there is no correlation. If lift is above 1, there is a positive correlation, and below 1 means a negative correlation (Han, Kamber, & Pei, 2011).

Part 2div

Confidence, support, and lift are different metrics we can use within the Apriori algorithm to narrow our parameters and discover more interesting predictive patterns in the datasets. The support threshold is a metric for the antecedent itemset, or the left-hand side. In this case the metrics is telling us the percentage of how often k-itemset is seen in the entire dataset. Confidence, on the other hand, tell us how often the antecedent itemset and consequent itemset are seen together. While these can be useful metrics, they do not always tell the whole story. If there happens to just be a lot of the consequent itemset, there is not necessarily a correlation. The lift metric takes into consideration the support metric for both itemsets and determines if there is a positive, negative, or zero correlation between the specified itemsets (Ng, 2016).

Part 2fi

Pruning is an important function for efficient utilization of the Apriori algorithm. This is referring to the process of removing transactions that are not interesting for any association rules. The Apriori property, mentioned in a previous answer, plays a part in this ensuring our minimum thresholds are maintained. Other actions taken to properly prune the dataset includes item merging, sub-itemset pruning, and item skipping. Item merging will form frequent closed itemsets if there is no superset of the consequent itemset. Sub-itemset pruning works by removing descendants of already discovered supersets. Item skipping involves pruning itemsets that have equivalent support at varying levels of different header tables (Han, Kamber, & Pei, 2011).