Algorithm	Apriori	Eclat
type	BFS (breadth first search)	DFS(depth first search)
Data representation	Horizontal (transactions)	Vertical (item -> transaction IDs)
Support calc	Count itemsets by scanning the database multiple times	Intersect transaction ID lists
Memory Usage	Moderate to high for large datasets	Often lower
Efficiency	Slower on large/dense datasets due to multiple scans and many candidates	Faster on dense datasets due to fewer database scans
Complexity	Generates lots of candidates, pruning needed	No candidate generation, recursively intersects tid-lists
Output	Frequent itemsets and association rules	Frequent itemsets and association rules
Ease of Implementation	Easy with libraries	More complex due to recursive implementation
Best For	Sparse datasets, smaller databases	Dense datasets, larger databases
Candidate Explosion Problem	have	Haven't
Execution Time to Produce Rules	Slower than eclat	Fast