Priming The Aggregate System With Demand-Defined Aggregates

You can cause the AtScale engine to generate demand-defined aggregates before any analysts start querying your data. By priming your cubes with aggregates, you might be able to improve the performance of the initial queries that analysts issue from their data-analytics software. Such priming is done by enabling Training Mode on the User Profile page and running queries representative of the queries that analysts will issue.

When you publish an AtScale cube, the AtScale engine defines and builds instances of all-member and dimension-only aggregates. (See Types of Aggregate Tables in AtScale.) However, the engine creates demand-defined aggregates only as queries are issued against the cube. This means that queries that a) the engine has not yet seen, b) do not resemble other initial queries on the cube, or c) cannot be satisfied by prediction-defined aggregates will run against raw data. As the engine builds, maintains, and revises a repertoire of aggregate tables, queries start running against the aggregate tables rather than against the raw data, and query performance starts greatly improving.

If you know the types of queries that analysts are going to use, you can run such queries in training mode to cause the AtScale engine to generate demand-defined aggregates. Training mode runs queries, but does not return results. After you publish a project, you enable **Training Mode** on the User Profile page and then issue the queries from data-analytics software. When you are finished, disable training mode and let analysts know that they can start querying the cubes in the project.