In the fields of heuristic optimization and machine learning, experimentation is the way to assess the performance of an algorithm setup and the hardness of problems. Good experimentation is complicated. Most algorithms in the domain are anytime algorithms, meaning they can improve their approximation quality over time. This means that one algorithm may initially perform better than another one but converge to worse solutions in the end. Instead of single final results, the whole runtime behavior of algorithms needs to be compared (and runtime may be measured in multiple ways). We do not just want to know which algorithm performs best and which problem is the hardest — a researcher wants to know why. We introduce a process which can 1) automatically model the progress of algorithm setups on different problem instances based on data collected in experiments, 2) use these models to discover clusters of algorithm (or problem instance) behaviors, and 3) propose reasons why a certain algorithm setup (or problem instance) belongs to a certain algorithm (or problem instance) behavior cluster. These high-level conclusions are presented in form of decision trees relating algorithm parameters (or instance features) to cluster ids. We emphasize the duality of analyzing algorithm setups and problem instances. Our process is implemented as open source software and tested in two case studies, on the Maximum Satisfiability Problem and the Traveling Salesman Problem. Besides its basic application to raw experimental data, yielding clusters and explanations of “quantitative” algorithm behavior, our process also allows for “qualitative” conclusions by feeding it with data which is normalized with problem features or algorithm parameters. It can also be applied recursively, e.g., to further investigate the behavior of the algorithms in the cluster with the best-performing setups on the problem instances belonging to the cluster of hardest instances. Both use cases are investigated in the case studies.