Report: Time Savings and Reproducibility Using Dagster ML Pipeline

Conventional workflows for **machine learning in Jupyter notebooks** frequently have **problems with reproducibility**, as code blocks can be rewritten, executed in a non-linear way, or run with varying **data versions** over time. These issues can create **inconsistent results** and make **debugging complicated**. This project introduces an alternative approach to create a more reproducible and dependable machine learning pipeline using the new **data orchestration framework** known as **Dagster**.

The machine learning process for this project uses the **Superstore dataset** and includes all the major **steps for creating a machine learning model**, such as **importing data** from a source, **preparing data** for analysis, using **EDA** to explore the data, **splitting the data into a training set and test set**, **training a model**, and producing a **visual representation of results**. Three machine learning models were trained and evaluated using different types of algorithms: a **Decision Tree**, **Random Forest**, and **Logistic Regression**. **Dagster manages the dependencies** between each of the components of the machine learning process and **ensures the correct path for data** to flow through the various segments of the process so that **each step executes in the same manner every time**.

We've demonstrated **reproducibility** by running the pipeline through two iterations: one with the **original dataset** and the other with a **modified version of that same dataset**. The **pipeline code was the same in both situations**; only the **dataset is what was different**. Dagster automatically created **two distinct runs** and captured all **execution information** for those runs, demonstrating that the same pipeline can be **reused safely on different datasets** without requiring the user to **rerun all notebook cells manually**.

Traditionally, when you change your data **using a Jupyter notebook**, you must **rerun all cells** (pre-processing, EDA, and modeling), which could take **several minutes**. In contrast, using **Dagster to run the pipeline was completed** in approximately **40 seconds** for the original data and **44 seconds** for the modified data. When you consider that it could take **3–5 minutes** to rerun an entire Jupyter notebook, using Dagster to **automate execution order** and support **controlled reruns** leads to a dramatic **increase in efficiency**.

Overall, this project provides evidence that **Dagster allows for reproducible, efficient, and well-structured machine-learning workflows**. When an **unstructured Jupyter notebook workflow** is converted into a **structured pipeline workflow**, execution occurs **faster**, is much **easier to manage**, and can be **more reliably completed**. Thus, **Dagster is an ideal technology for real-world machine-learning applications**.