Object storage is used to hold these objects, due to its scalable nature, having no predefined structure. Ensuring reproducibility requires version control and tracking lineage for all artifacts used and produced throughout the data and model development processes.

### Versioning Binary Artifacts

Reproducibility in ML requires tracking how data is combined with code and any artifacts produced in the process. Most artifacts used and produced in machine learning are binary in nature, having no predefined structure. Ensuring reproducibility requires version control and tracking lineage for all artifacts used and produced throughout the data and model development processes. Versioning binary files, requires tracking chunks of binary data and detecting when pieces of it changes. To do this files are split into chunks, each versioned with a hashing function to ensure uniqueness. Object storage is used to hold these objects, due to its scalable nature.

Dependency management is crucial for reproducible software, especially with ML code. Containerization is frequently the best way to capture all dependencies because it allows users to create containerized definitions for PyTorch applications, making them reproducible and scalable.

### Reproducibility: Data Versioning with Pipelines

Reproducibility can be achieved by combining versioned data with reproducible pipelines. Pachyderm, a system built specifically for data versioning and data triggered pipelines, can be used in conjunction with TorchX (inside versioned containers) to initiate distributed training workload, data processing pipelines, and other PyTorch components. The system provides a full lineage and history of any changes in the system, whether to the code or data.

---

**References**
