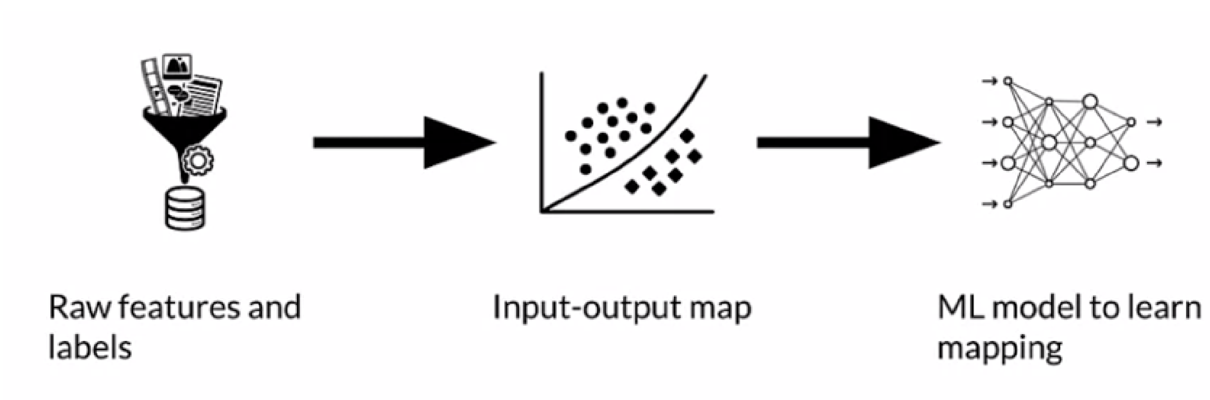
Week3: Data Journey, Data Storage, Evolving Data, Enterprise Data Storage.

Data Journey:



Data Transformation:

- Data transforms as it flows through the process
- Interpreting model results requires understanding data transformation

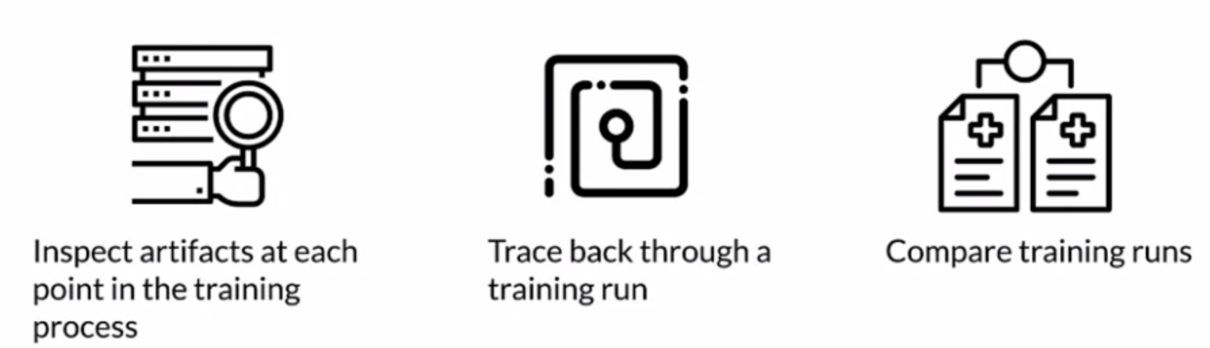
Artifacts and the ML pipeline:

- Artifacts are created as the components of the ML pipeline execute
- Artifacts include all of the data and objects which are produced by the pipeline components
- This includes the data, in different stages of transformation, the schema, the model itself, metrics, etc.

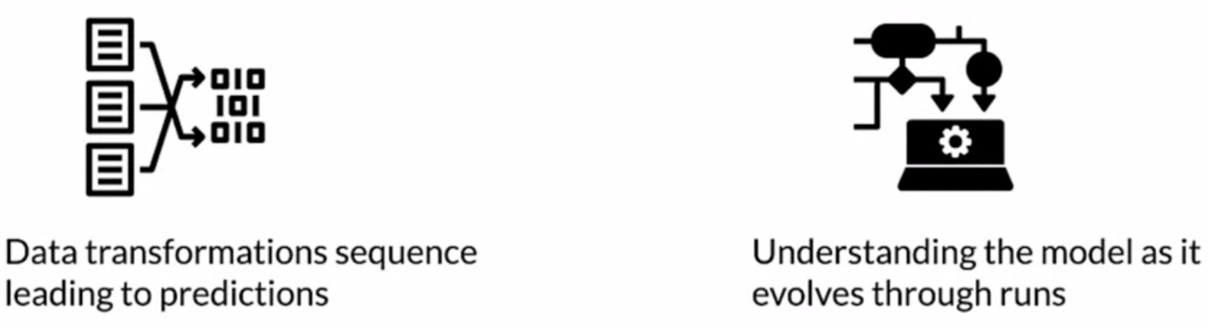
Data provenance and lineage:

- The chain of transformations that led to the creation of a particular artifact
- Important for debugging and reproducibility

This helps with debugging and understanding the ML pipeline



- Organizations must closely track and organize personal data
- Data lineage is extremely important for regulatory compliance
- It is key for understanding model results

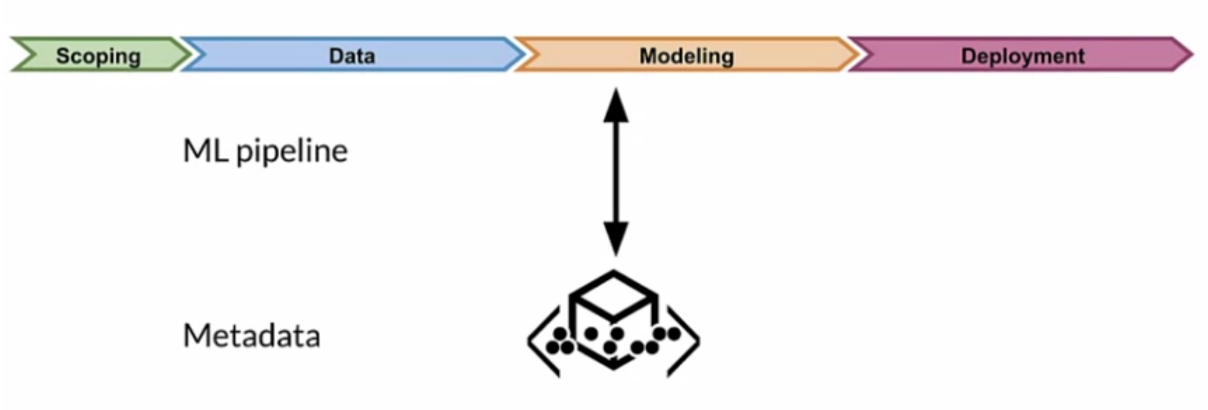


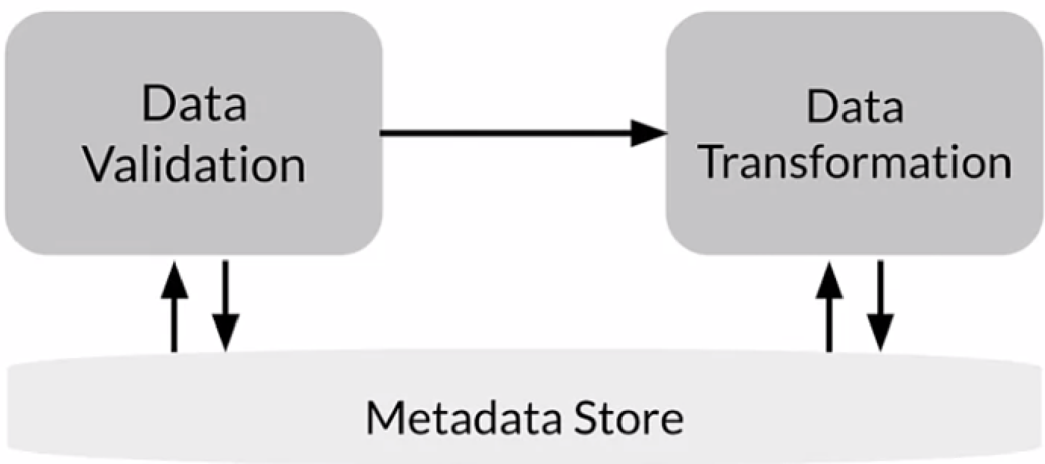
Data Versioning:

- Data pipeline management is a major challenge
- Machine learning requires reproducibility
- Code versioning: GitHub and similar code repositories
- Environment versioning: Docker, Terraform, and similar.
- Data versioning:
    - Version control of datasets.
    - Examples: DVC, Git-LFS (Tools for data versioning)

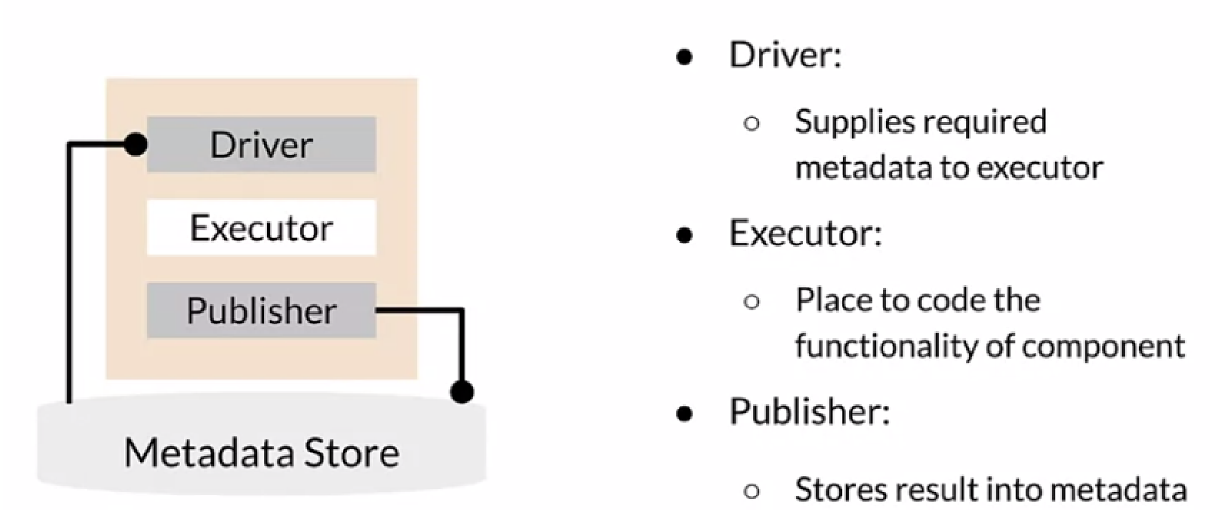
Metadata: Tracking artifacts and pipeline changes

Metadata is like logging in software engineer





Metadata: TFX component architecture



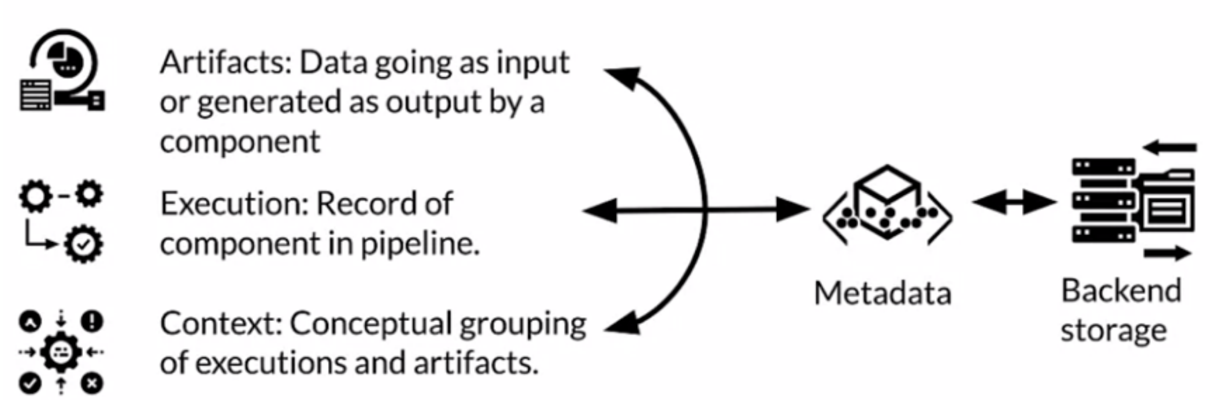
ML metadata Library:

- ML MD
- Tracks metadata flowing between components in pipeline
- Supports multiple storage backends

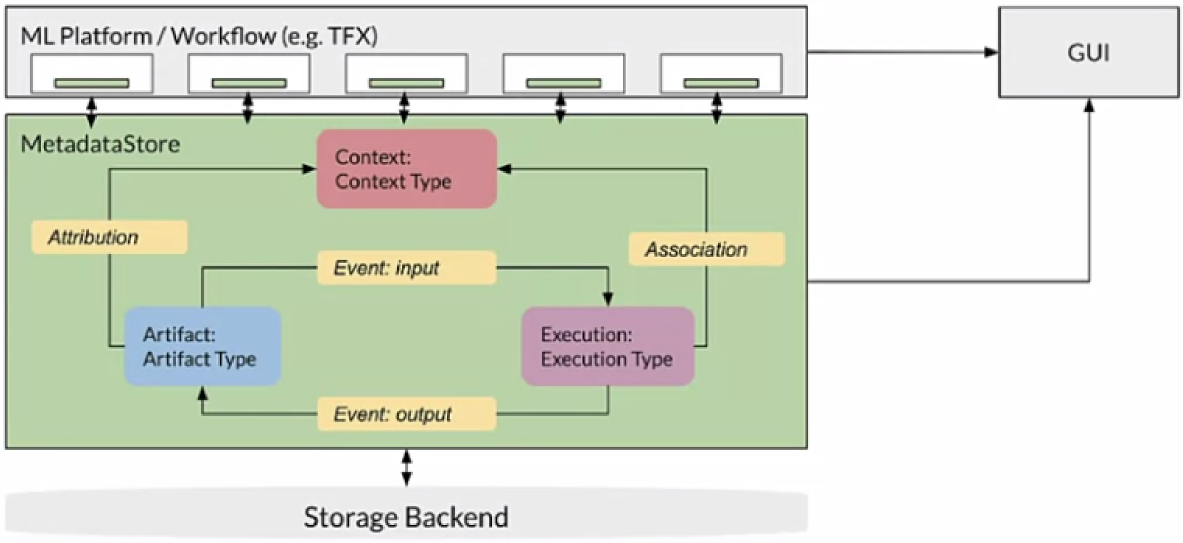
ML metadata terminology:

Units	Types	Relationships
Artifact	ArtifactType	Event
Execution	ExecutionType	Attribution
Context	ContextType	Association

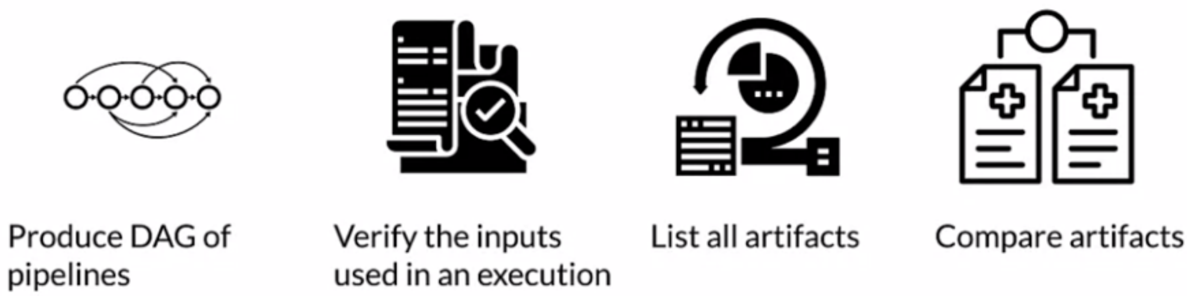
Metadata Stored:



Inside MetadataStore:



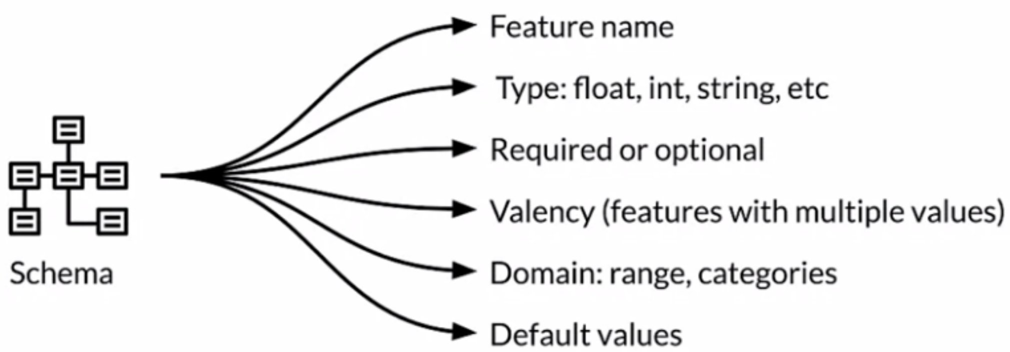
Benefits of ML Metadata:



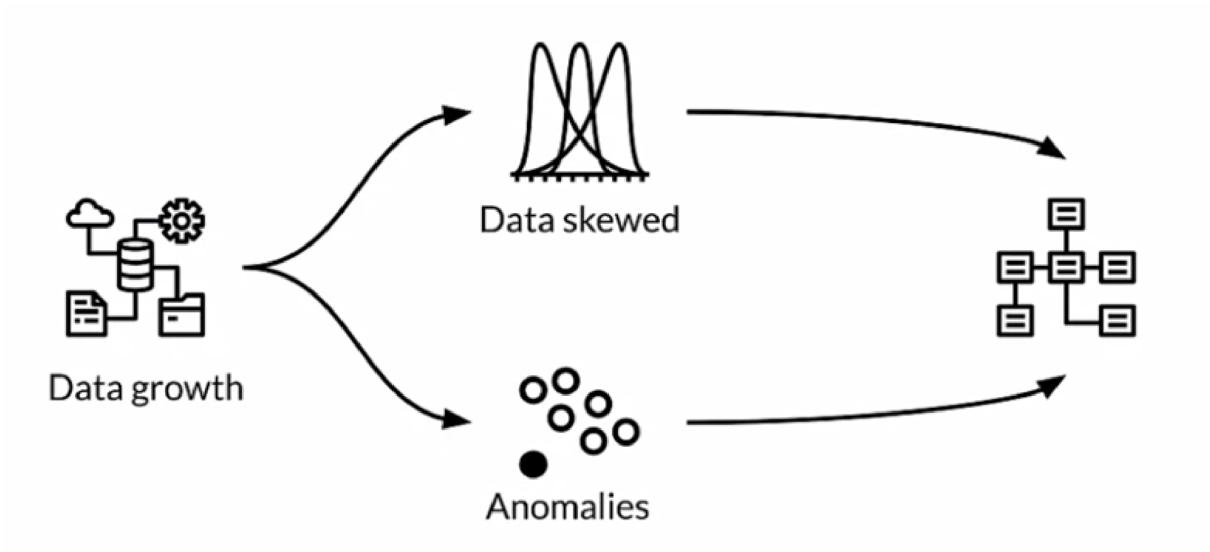
- ML metadata registers metadata in a database called Metadata Store
- APIs to record and retrieve metadata to and from the storage backend:
    - Fake database: in-memory for fast experimentation/prototyping
    - SQLite: in-memory and disk
    - MySQL: server based
    - Block storage: File system, storage area network, or cloud based.

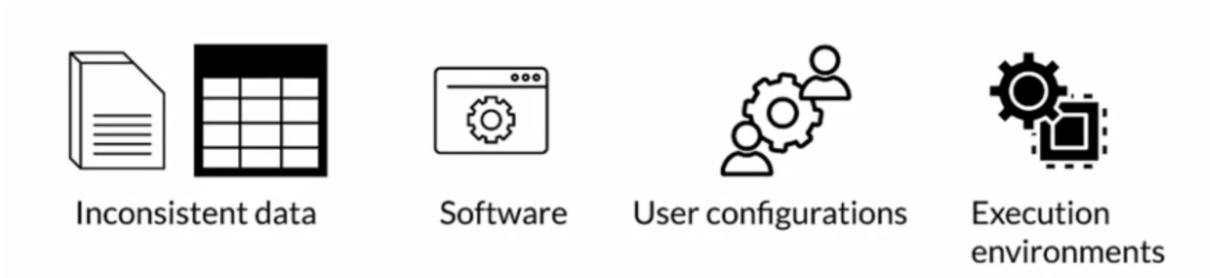
Evolving Data:

Schema: Relational objects summarizing the features in a given dataset

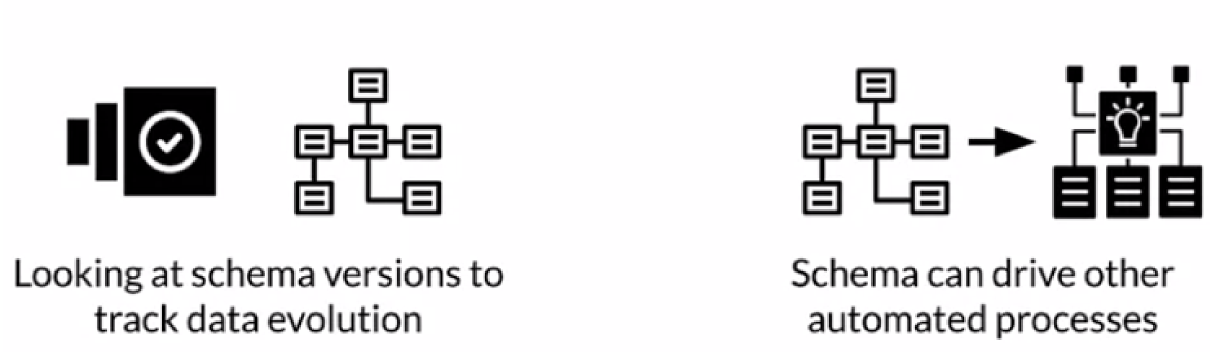


Iterative schema development & evolution

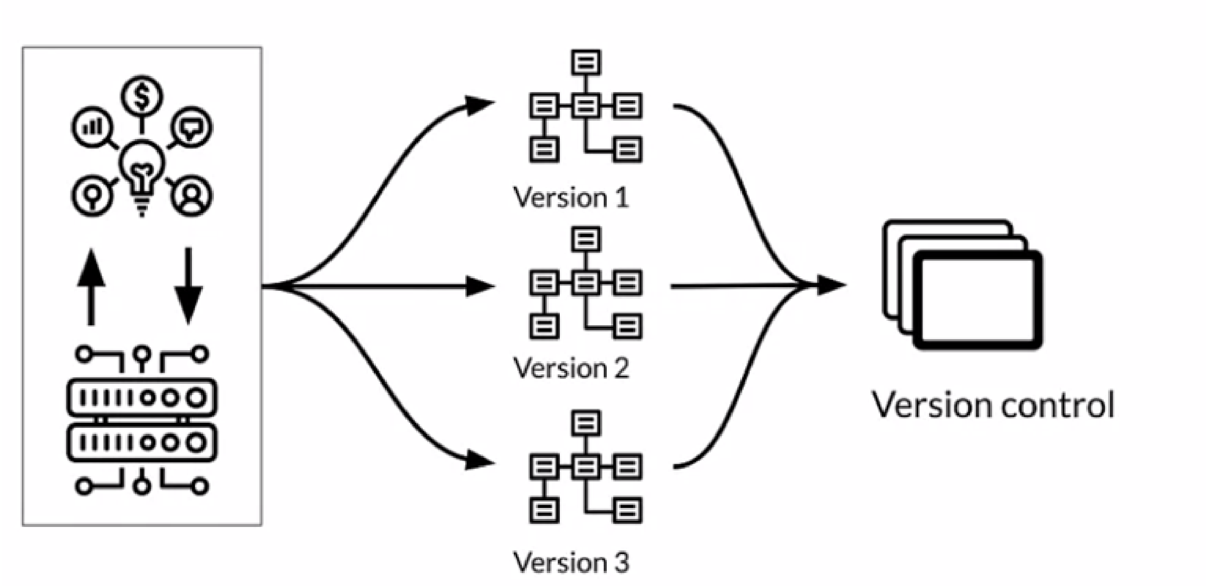




Scalability:



Multiple schema versions:



Feature storages:

Week 3: Data Journey and Data Storage

If you wish to dive more deeply into the topics covered this week, feel free to check out these optional references. You won't have to read these to complete this week's practice quizzes.

Data Versioning:

1. https://dvc.org/
2. https://git-lfs.github.com/

ML Metadata:

1. https://www.tensorflow.org/tfx/guide/mlmd#data_model
2. https://www.tensorflow.org/tfx/guide/understanding_custom_components

Chicago taxi trips data set:

1. https://data.cityofchicago.org/Transportation/Taxi-Trips/wrvz-psew/data
2. https://archive.ics.uci.edu/ml/datasets/covertype

Feast:

1. https://cloud.google.com/blog/products/ai-machine-learning/introducing-feast-an-open-source-feature-store-for-machine-learning
2. https://github.com/feast-dev/feast
3. https://www.gojek.io/blog/feast-bridging-ml-models-and-data