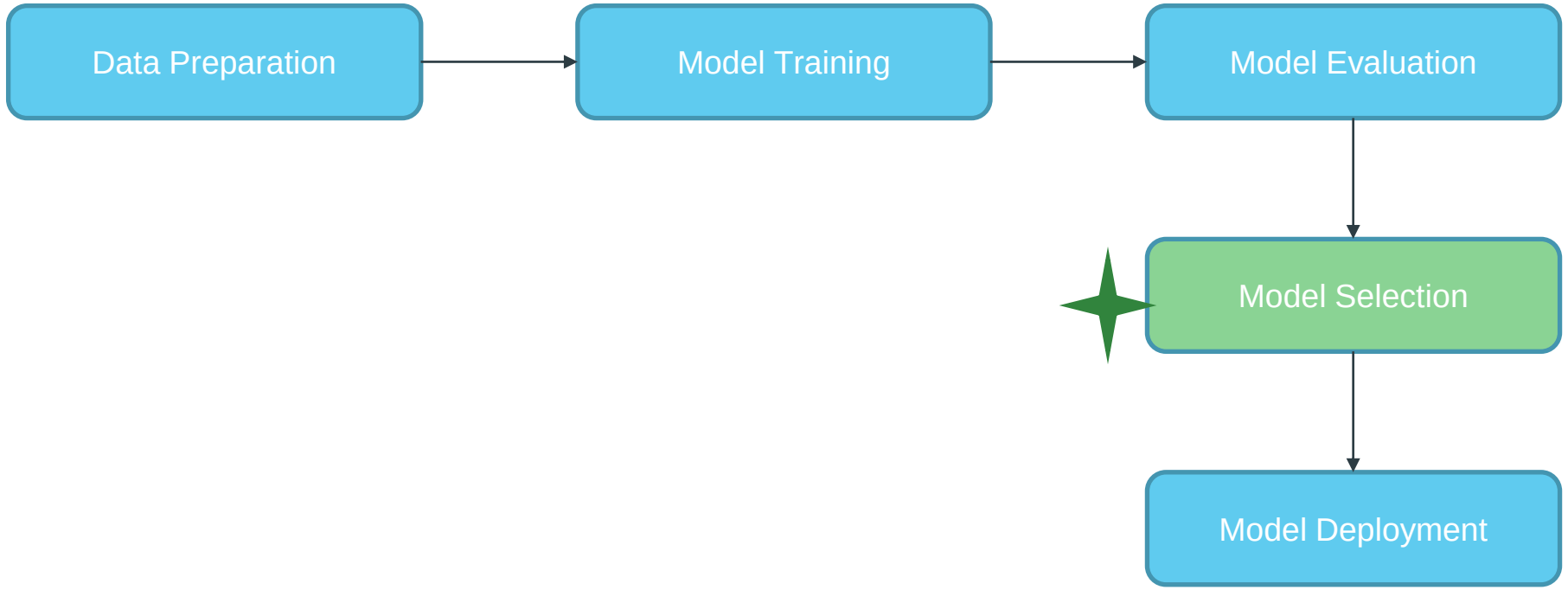
Azure Databricks ML Flow

Agenda

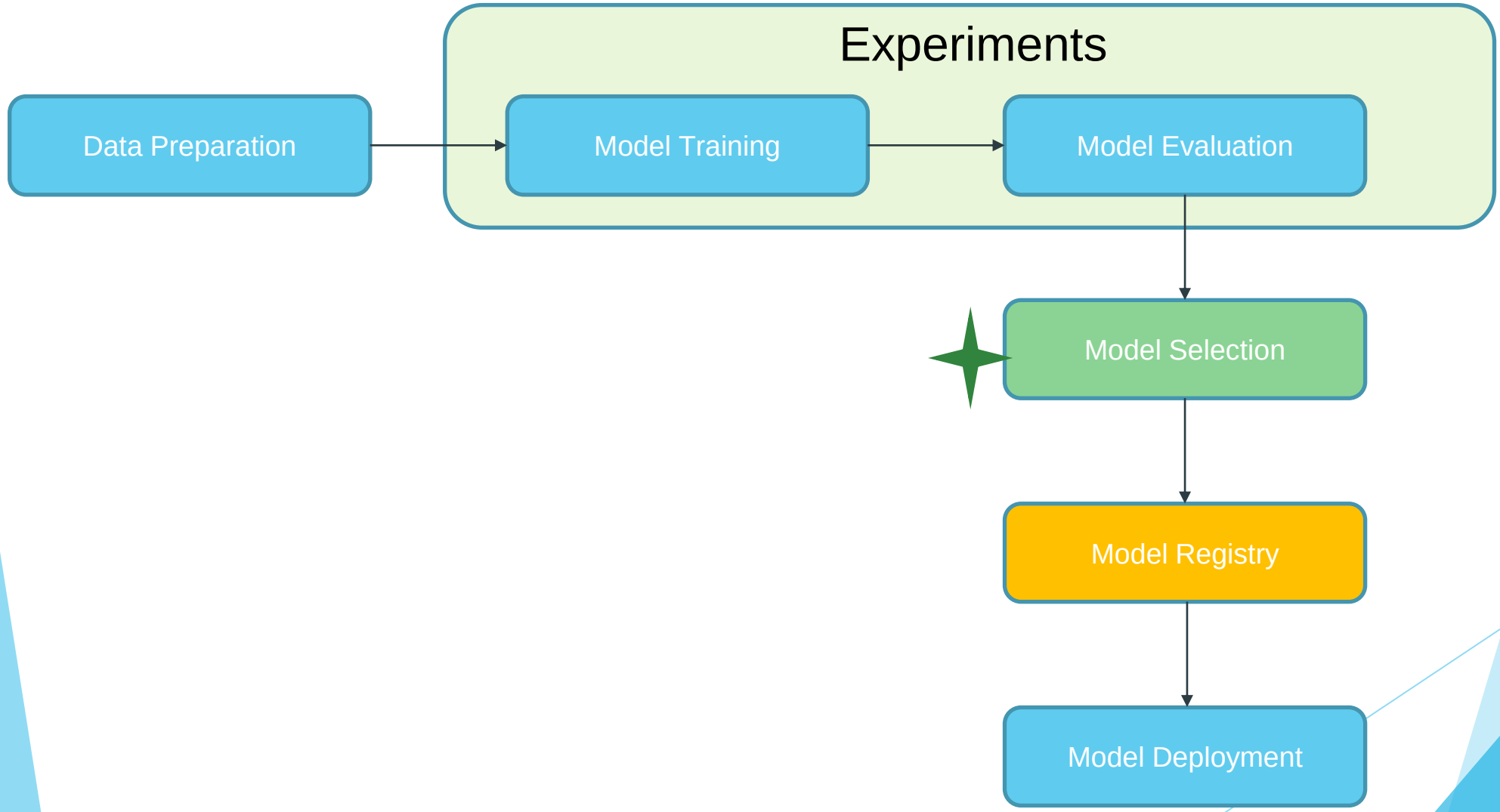
- ML Experiment
  - Notebook Experiment
  - Workspace Experiment
- Log, Load Models
- ML Model Registry
- Deploy ML Models
- ML Model API

ML Workflow


Data Preparation
Model Training
Model Evaluation
Model Selection
Model Deployment

Experimentation with ADB - MLFlow

Experiments can have multiple runs inside it. Each run can belong to same algorithm or different algorithms



Types of Experiments

There are two types of Experiments available in Azure Databricks

- ## Notebook Experiments

A notebook experiment is associated with a specific notebook. Azure Databricks automatically creates a notebook experiment if there is no active experiment when you start a run using mlflow.start_run()

- ## Workspace Experiments

You can create a workspace experiment from the Databricks Machine Learning UI or the MLFlow API. Workspace experiments are not associated with any notebook, and any notebook can log a run to these experiments by using the experiment ID or the experiment name.

ML Flow Runs

An MLFlow *run* corresponds to a single execution of model code. Each run records the following information

Source: Name of the notebook that launched the run or the project name and entry point for the run.

Version: Notebook revision if run from a notebook or Git commit hash if run from an MLflow Project.

Start & end time: Start and end time of the run.

Parameters: Model parameters saved as key-value pairs. Both keys and values are strings.

Metrics: Model evaluation metrics saved as key-value pairs. The value is numeric. Each metric can be updated throughout the course of the run (for example, to track how your model's loss function is converging), and MLFlow records and lets you visualize the metric's history.

Tags: Run metadata saved as key-value pairs. You can update tags during and after a run completes. Both keys and values are strings.

Artifacts: Output files in any format. For example, you can record images, models (for example, a pickled scikit-learn model), and data files (for example, a Parquet file) as an artifact.

Model Management

Azure Databricks provides a hosted version of MLFlow Model Registry to help you to manage the full lifecycle of MLflow Models.

Model Registry provides **chronological model lineage** (which MLFlow experiment and run produced the model at a given time), model versioning, stage transitions (for example, from staging to production or archived), and email notifications of model events. You can also create and view model descriptions and leave comments.

Reference:
https://docs.microsoft.com/en-us/azure/databricks/applications/machine-learning/manage-model-lifecycle/

Practical Demo:
Registering the Model

ML Staging

Transition a model stage

A model version has one of the following stages: **None**, **Staging**, **Production**, or **Archived**.

The **Staging** stage is meant for model testing and validating

Production stage is for model versions that have completed the testing or review processes and have been deployed to applications for live scoring.

An **Archived** model version is assumed to be inactive, at which point you can consider deleting it.

Different versions of a model can be in different stages.

A user with appropriate permission can transition a model version between stages. If you have permission to transition a model version to a particular stage, you can make the transition directly. If you do not have permission, you can request a stage transition and a user that has permission to transition model versions can approve, reject, or cancel the request.

Reference: https://docs.microsoft.com/en-us/azure/databricks/applications/machine-learning/manage-model-lifecycle/#str

Deploy and Serve Models

Serve models with MLflow

Azure Databricks provides MLflow Model Serving, which allows you to host machine learning models from the Model Registry as REST endpoints that are updated automatically based on the availability of model versions and their stages. MLflow Model Serving is available for Python MLflow models.

Reference: https://docs.microsoft.com/en-us/azure/databricks/applications/machine-learning/model-deploy/

Practical Demo:

Deploy a model

Scoring new Data using REST endpoint URI

Caveats

Model API can work only if the data types of the input data in model signature matches with the data we are passing to API