Operationalizing ML Models

October 26, 2019

Operationalizing ML Models built with Azure Databricks through Azure ML

John Hoff

Machine Learning Architect

jhoff@productiveedge.com

What Do I Mean by Operationalize?

To operationalize machine learning models is to provide open access to models for use by personnel and applications in a systematic manner that provides the ability to monitor the usage and relevance of the insights provided by the models.

Why Would You Want to Operationalize?

Bring a modern DevOps strategy to your machine learning team and how they interact with the rest of your organization.

- Automated
  - Reduce risks of mistakes and faster delivery
- Open
  - Can be used by any other technical personnel or application in the organization
- Monitored
  - Is it being used properly?

Operationalizing is a Challenge

There are multiple different ways to operationalize – What works for one company may not work for another

Best practices and standards are still emerging – You will be expected to fill the gaps as you find them

Many powerful tools are available – Almost too many! How do you choose the right tools and platforms for your needs?

A wide range of skills are required for success – This will be a team effort

Who is this session for?

Folks who want to know how to take their investment in using Azure for machine learning further.

This is not for people who want to know what machine learning is or how to do it.

What Am I Covering Today

Define a possible strategy to operationalize ML models

Demonstrate a sample implementation

- → Azure Databricks
- → Azure Machine Learning Services

Share this implementation so you can learn from it

- → This presentation and all source code is available on github: https://git.io/Je05T

Answer your questions

- → Find me at our booth for a conversation or email me: jhoff@productiveedge.com

What is an ML Model?

A mathematical model created using machine learning algorithms to make decisions or predictions without being explicitly directed how to complete the task.

For most people this it is a black box that you pour data into and get an answer back out.

How do we make that block box easier to use for roles outside of data science such as analysts, designers, engineers, or product managers?

Azure Databricks

One cloud platform for massive scale data engineering and collaborative data science

- Collaborative data science workspace for scientists, engineers, and analysts
- Unified data service that combines reliable data lakes with powerful data pipelines
- Simple, scalable, and secured enterprise cloud service

Azure Machine Learning Services

Build models rapidly and operationalize at scale from cloud to edge.

- → Simplified machine learning
- → Robust DevOps
- → Scale on demand
- → Azure enterprise ready

Why Combine These Two Platforms?

Azure Machine Learning Services

Deliver those insights to your users

- → Clear focus on the operations side of machine learning.
- → Ready to integrate to your current Azure infrastructure
- → You deploy and execute here

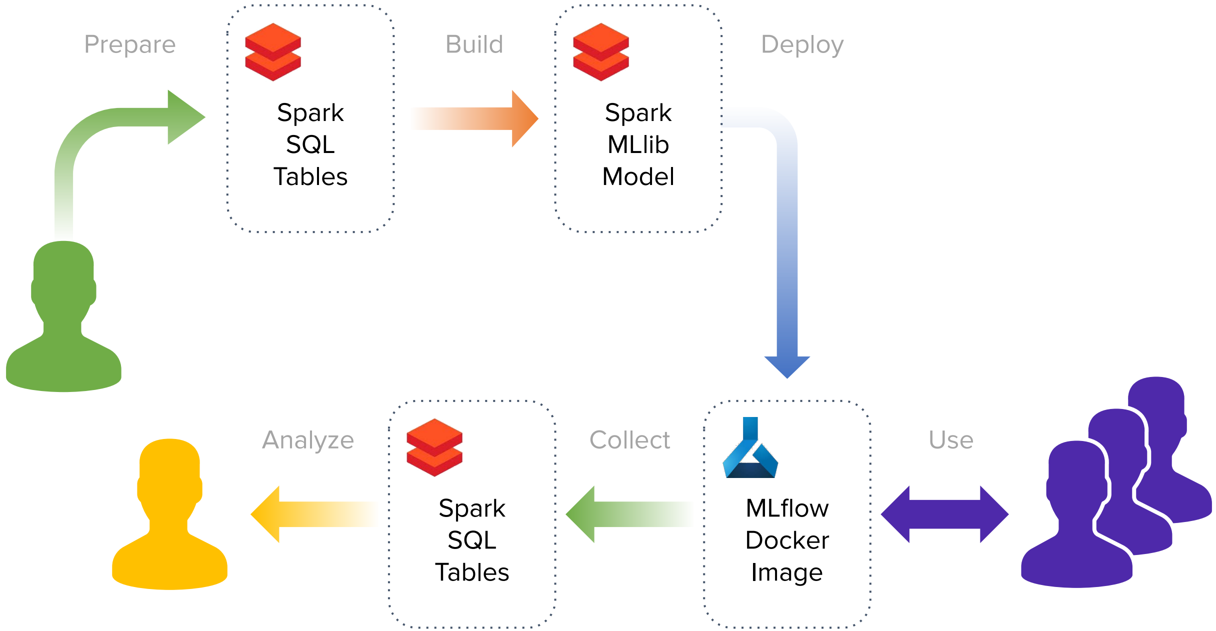
Azure Databricks

Create insights from your data

- → Leading big-data analytics platform.
- → You aren't going to outscale it.
- → You create models here.

Everything I'm about to show can be done using only Azure Machine Learning Services…so why am I talking about Azure Databricks as well? The capabilities of Azure Databricks are far superior in the realm of big-data and modern data lakes. If you aren't in that realm now, you likely will be soon.

Define Our Operationalizing Strategy



Define Our Operationalizing Model

Prepare Data:

- → Human-readable datasets without confusing machine learning transformations

Build and Deploy a Model

- → Data transformation automatically performed on training data and production inputs
- → Self contained Docker image containing the model

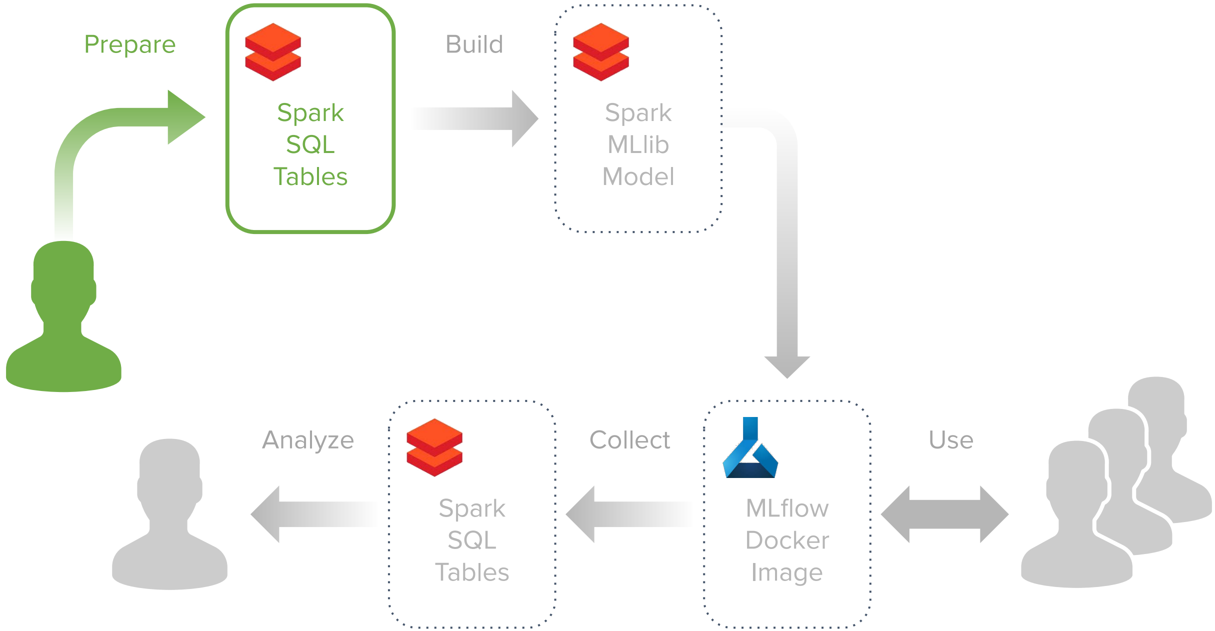
Use a model

- → Easy to consume REST endpoint

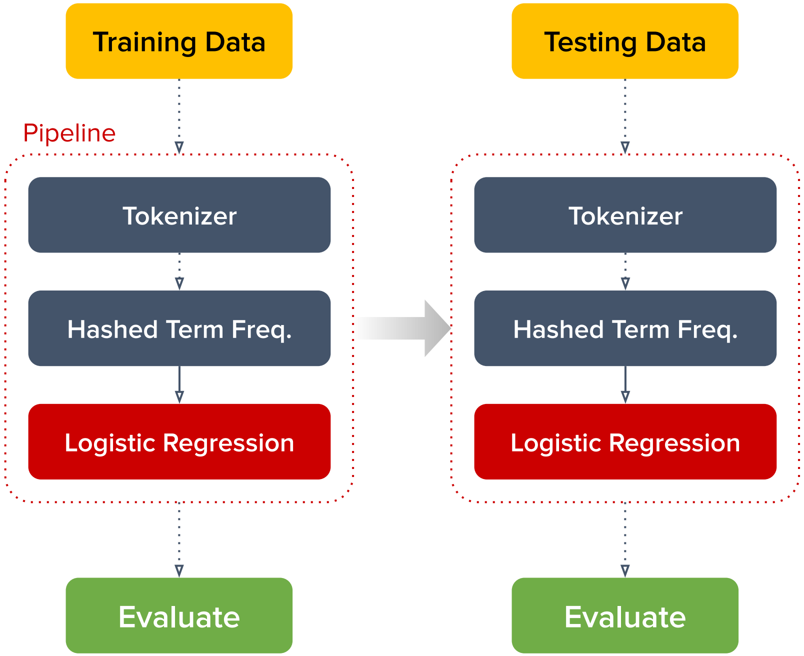
Collect and Analyze usage data:

- → Verify users are receiving relevant and accurate predictions and decisions

Preparing the Data



What is an ML Pipeline?



A set of steps that can be connected together to perform complex operations on data for machine learning.

Transformer: Inbound data is analyzed and transformed and delivered as inbound data to subsequent pipeline steps.

Estimator: The last step of a pipeline. Inbound data is consumed by a machine learning algorithm to train a model.

Why Use Pipelines?

Once a pipeline has been defined, it can be reused in other machine learning processes.

A defined pipeline is much like source code. It can be easily shared and extended to define new pipelines.

Once a pipeline has been trained, it can be can be executed by another process.

A trained pipeline is much like an executable application. It cannot be modified, but can be used to transform data by other applications.

What happens when you use an ML Pipeline

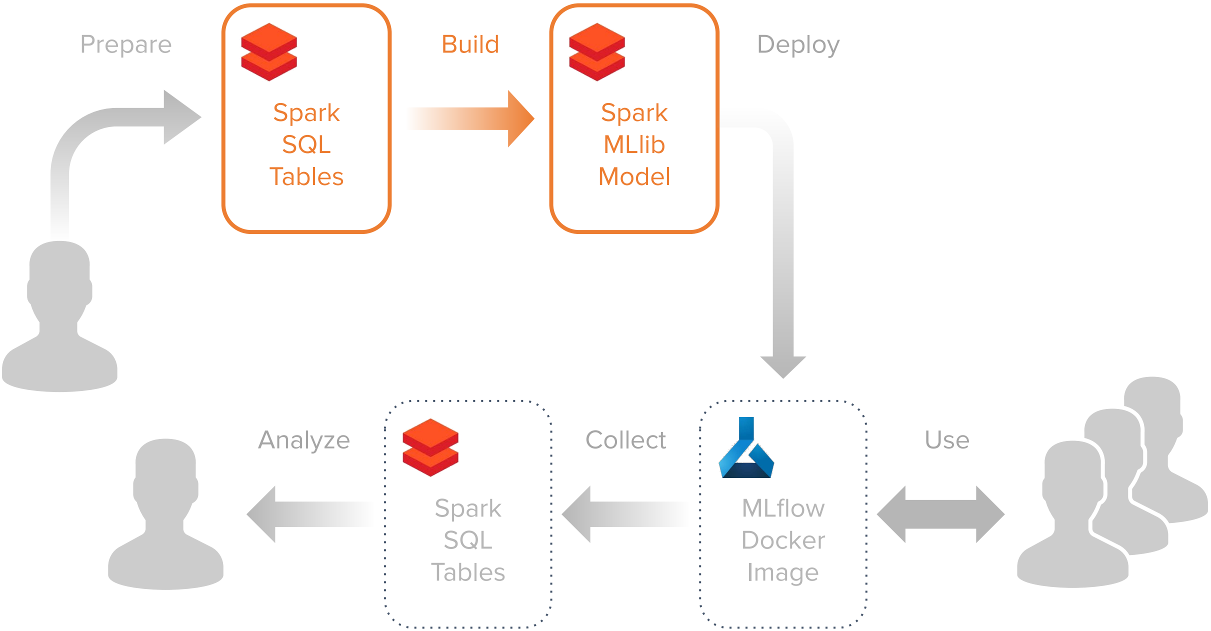
Instead of focusing on optimizing for machine learning, we will focus on human-accessible data.

Traditional pre-processing tasks are being moved into ML Pipelines.

The machine learning libraries in Azure Databricks provide robust ML pipelines that scale to datasets of almost any size.

age	job	marital	education	default
58	admin.	single	university.degree	no
49	admin.	married	high.school	no
44	blue-collar	married	basic.9y	no
39	blue-collar	married	basic.9y	no
49	blue-collar	married	professional.course	no
38	blue-collar	divorced	basic.9y	no
51	admin.	married	basic.4y	unknown
41	blue-collar	married	basic.9y	no

Building the Model



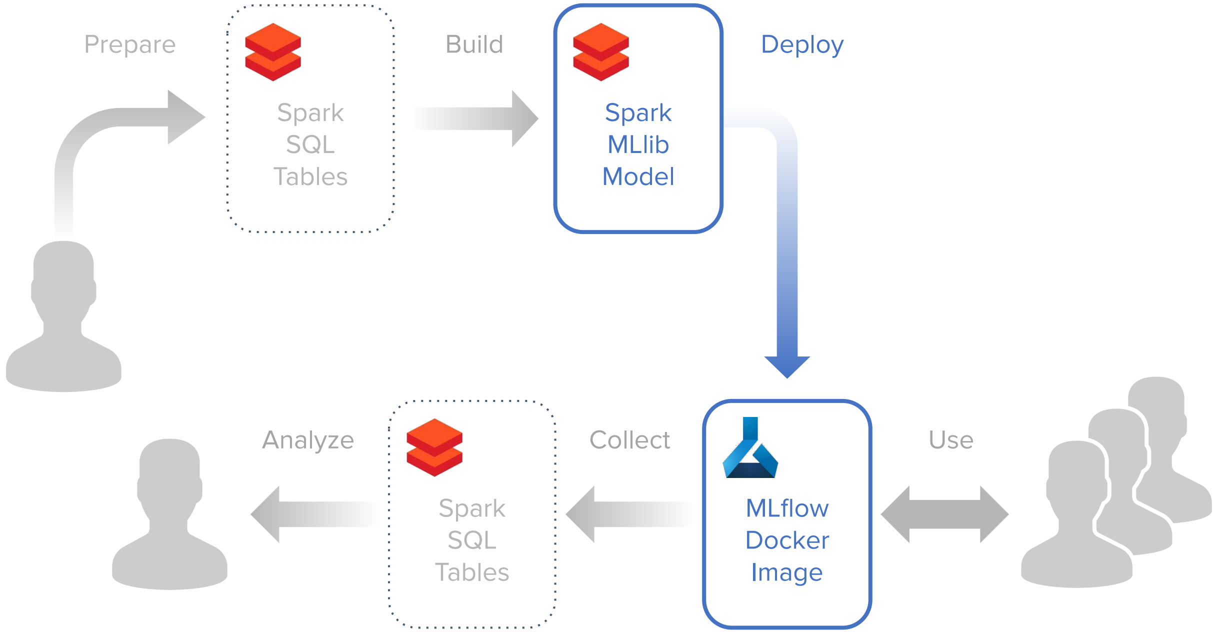
Building the Model

This is where we build the pipeline.

We take the Spark SQL data we have prepared, run it through the ML Pipeline and save the resulting train MLlib model.

What’s the reason for doing this?

Deploying the Model



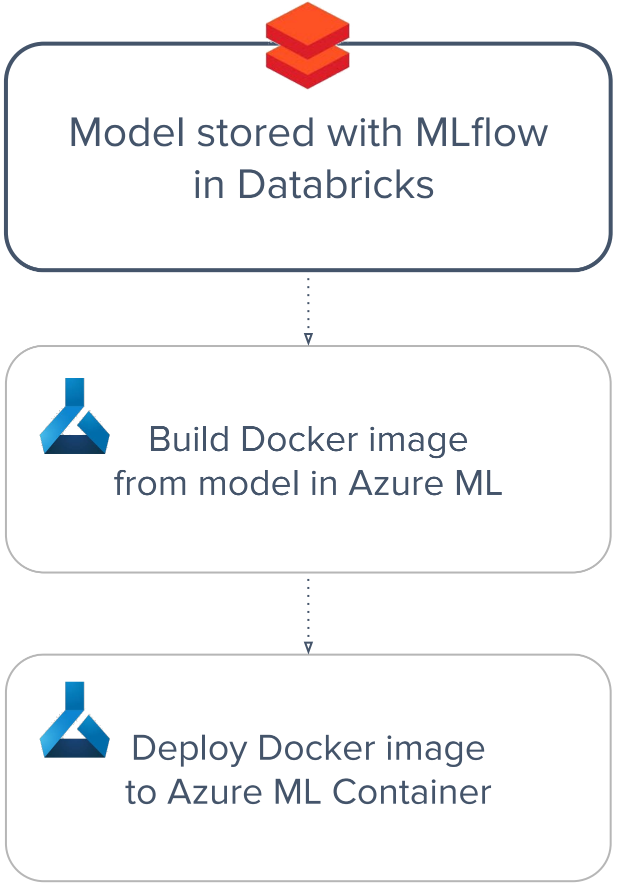
Deploying the Model

From a saved Spark MLlib model, we can create a docker image for deployment.

- Automatically manage dependencies
- Track multiple versions of a given model

Once a image has been created, it can be quickly deployed as a REST service

- Azure Container Instance for testing and validating models
- Azure Kubernetes Services for deploying auto-scaling production models
- No changes required to deploy a model to either service



Why Deploy as a REST Service?

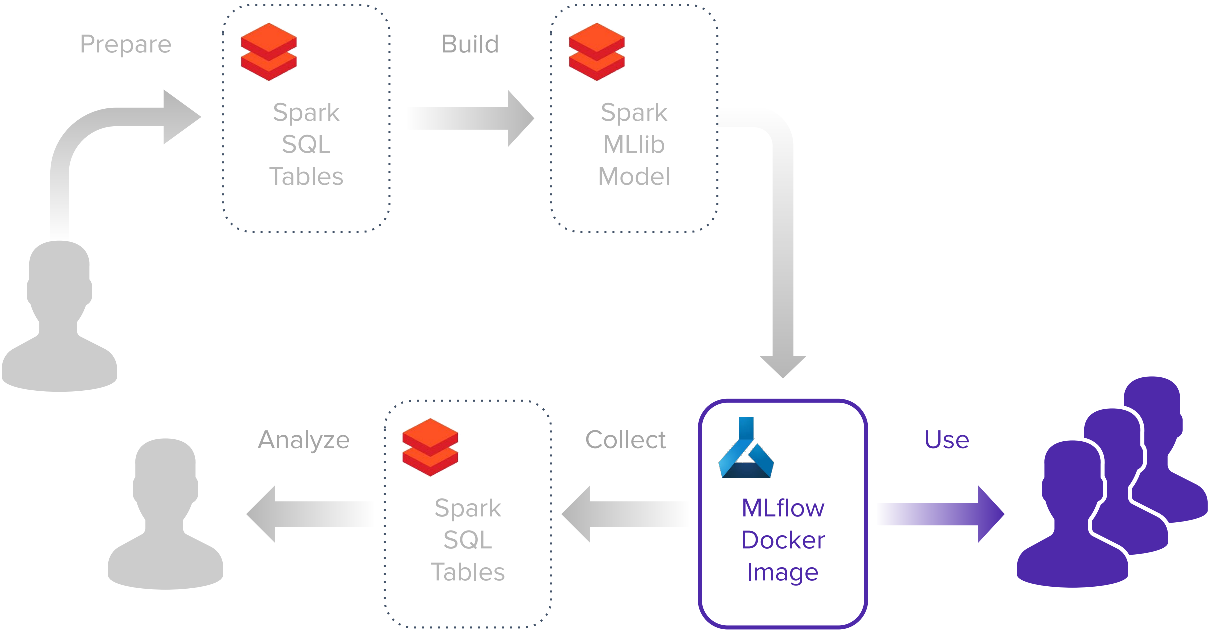
Allows for the model to be easily integrated into other software systems.

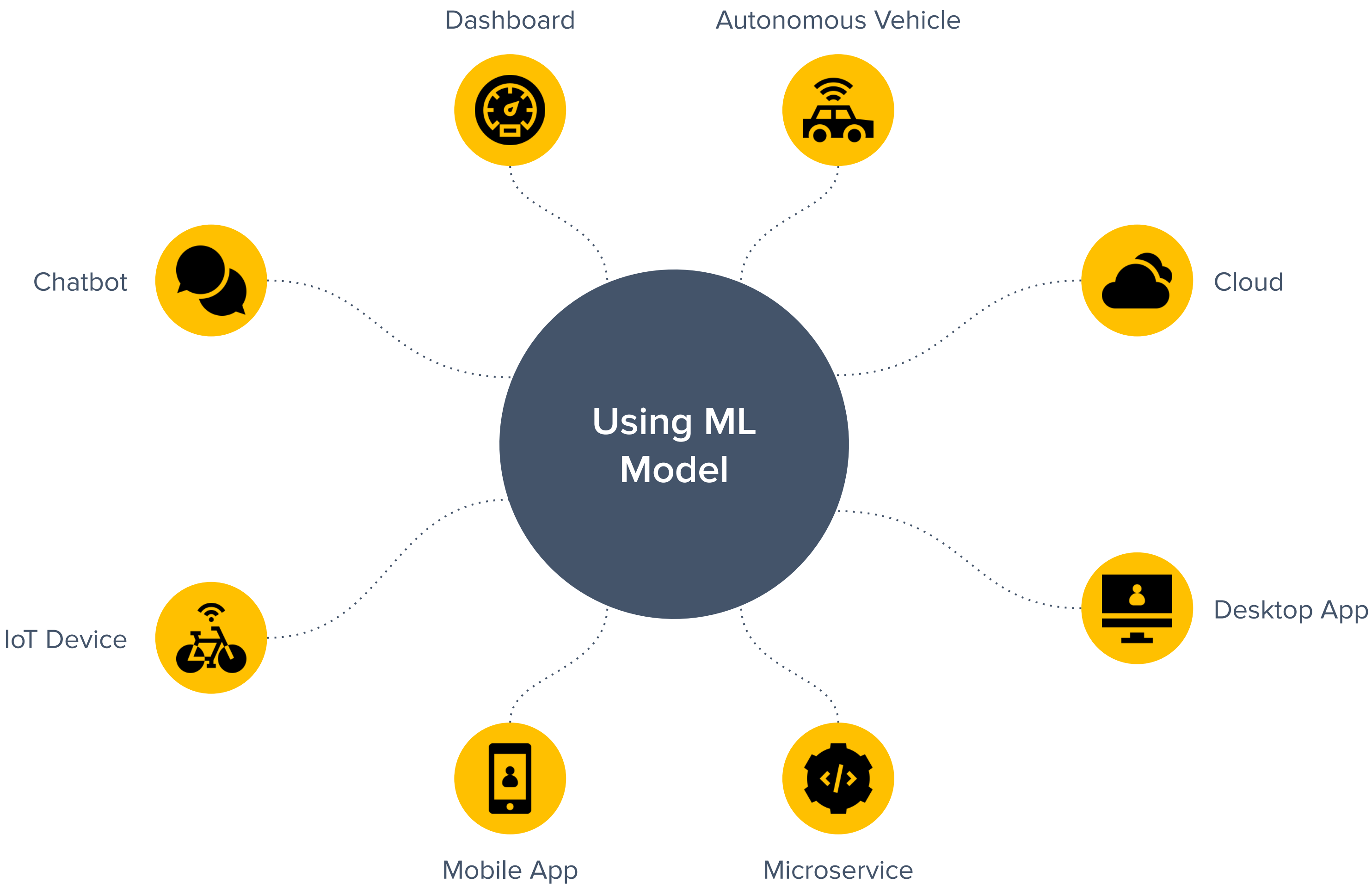
Model can be used by any language and platform.

Minimal set of dependencies required to interact with the model.

Capable of working in both batch and individual modes.

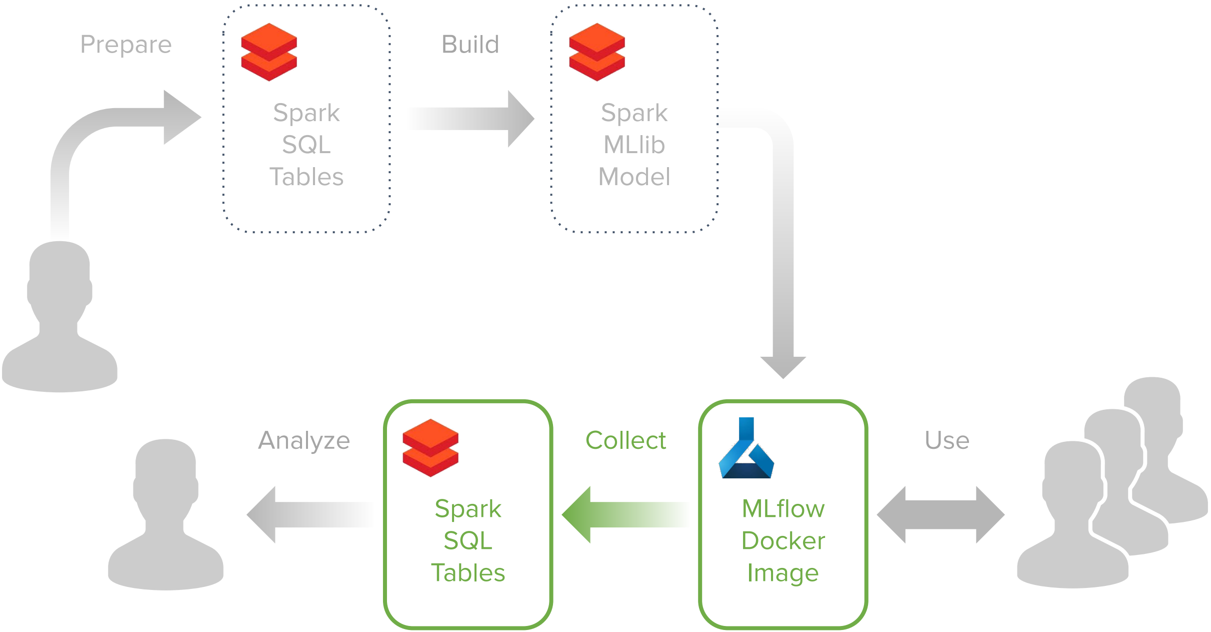
Using the Model





Dashboard
Autonomous Vehicle
Chatbot
Cloud
Using ML Model
IoT Device
Desktop App
Mobile App
Microservice

Collecting Model Usage



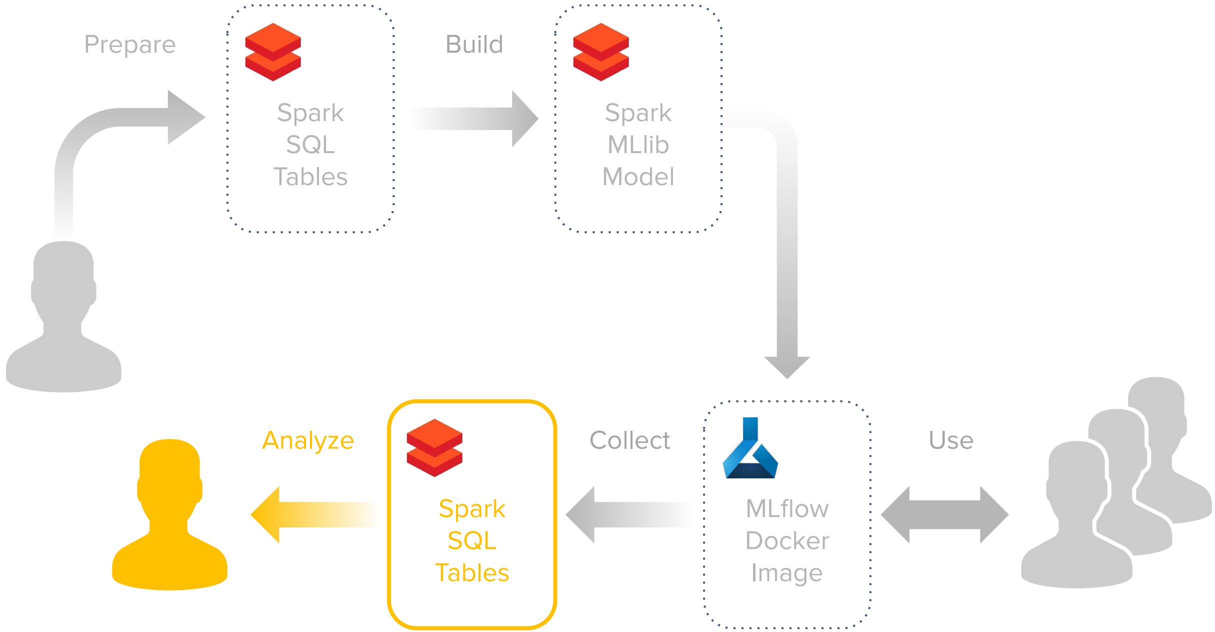
Collecting Model Usage

Collect logs of deployed images

Extract recorded model inputs

Save recorded model inputs into Spark SQL tables

Analyzing Model Usage



Analyzing Model Usage

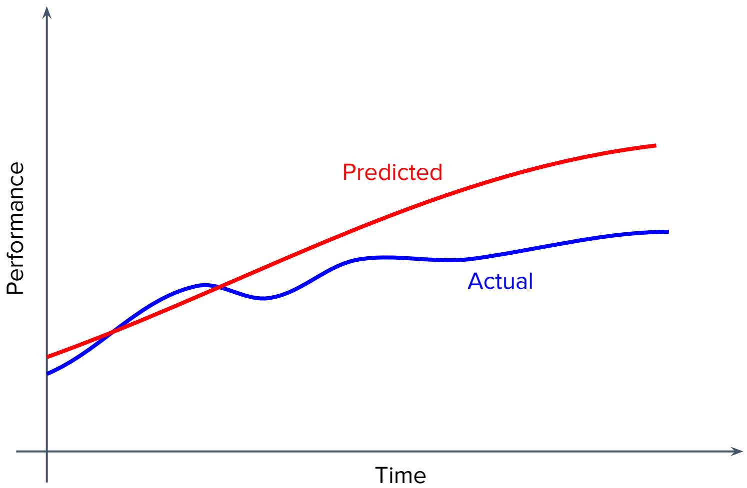
What specifically will we be looking for?

- Has the model performance been changing over time?
- Are the model inputs still representative of the training data for the model?

What is the core question that needs to be answered?

- Is the deployed model being used properly?

Measure Changes in Model Performance



Simple to Implement

Depends on model error

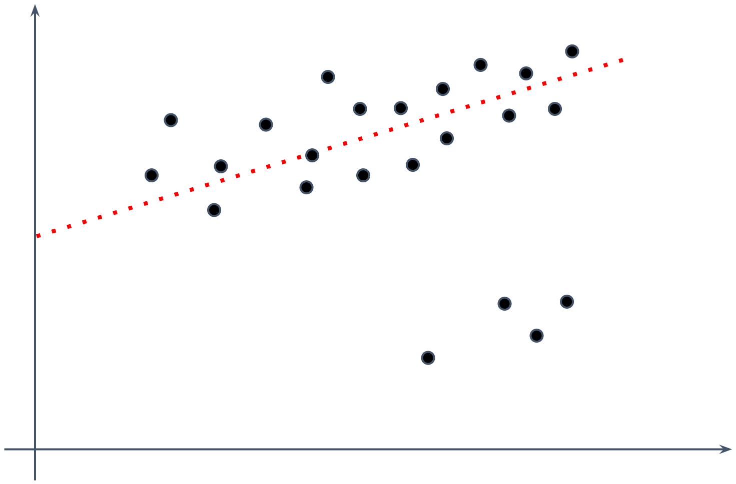
Usually a passive process performed by incorporating fresh data.

Measure Changes in Model Inputs

Hard to Implement

Does not depend on error

An active process done by actual analyzing model inputs



Example Implementation

https://git.io/Je05T

Self-contained set of Databricks notebooks

Includes all source code required for the implementation…just import and run

Data provided by UCI Machine Learning Repository

Data Source

Standard DSv2 Family vCPUs

Why this dataset?

- Mix of category and numeric attributes to showcase ML pipelines
- Binary classification to keep things simple
- Large enough size to use model with unseen data

Bank Marketing Data Set
UCI Machine Learning Repository

45k records with 21 attributes.

Classification Problem: Will the client subscribe to a term deposit?

Implementation Requirements

Azure Subscription Quotas

Standard DSv2 Family vCPUs and Total Regional vCPUs

- 10 for ACI model deployment
- 20 for AKS model deployment

Azure Machine Learning Services

- Active Azure ML Workspace
- No special configuration required

Azure Databricks

Active Azure Databricks Workspace

Running interactive cluster

- Smallest possible cluster works fine
  - single driver node
  - single worker node
- 5.5 TLS Cluster (NOT ML version)
- Python modules installed
  - mlflow==1.3.0
  - azureml-sdk[databricks]

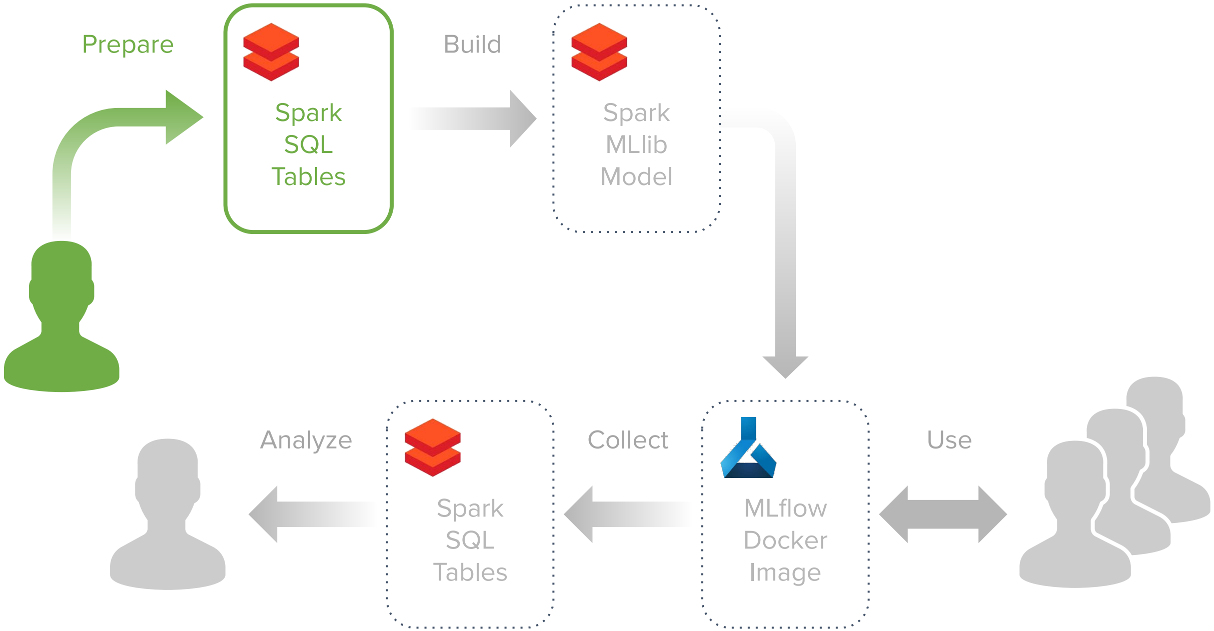
Preparing The Data

Download open dataset

Break Into Two Sets:

- → Training Set Used to Construct Model
- → Skewed Set Used as Model Inputs

Ensure Drift is Present



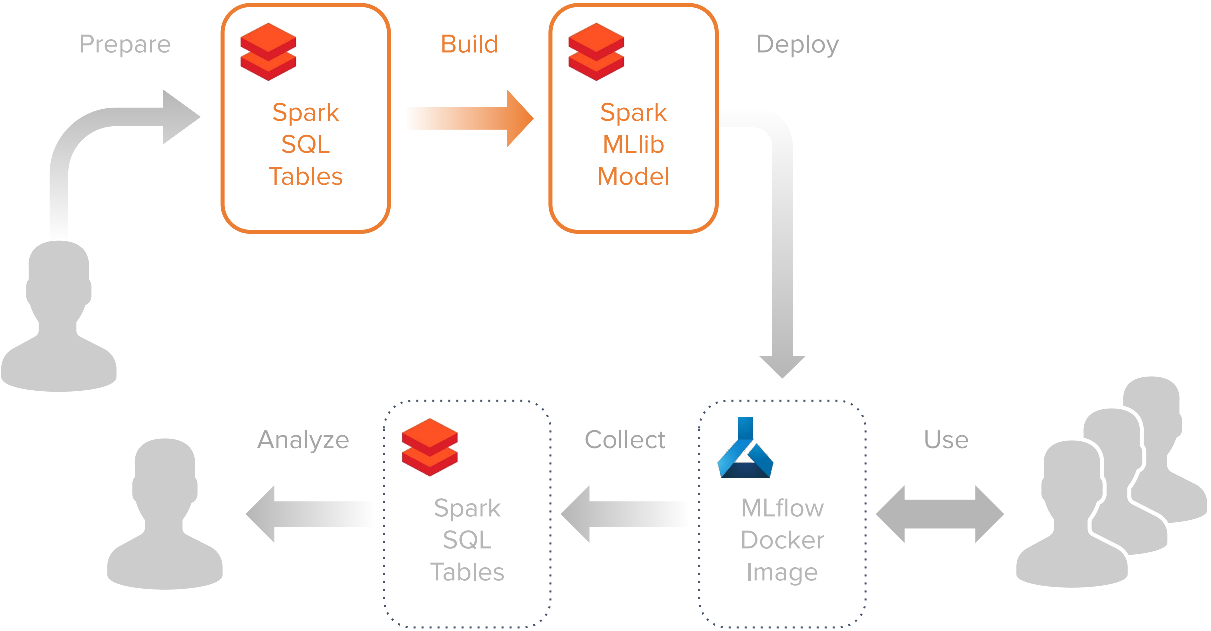
Building the Model

Create a Reusable Preprocessing Pipeline

Train a Random Forest Model

Collect Model Performance

Track and Store Model with MLflow

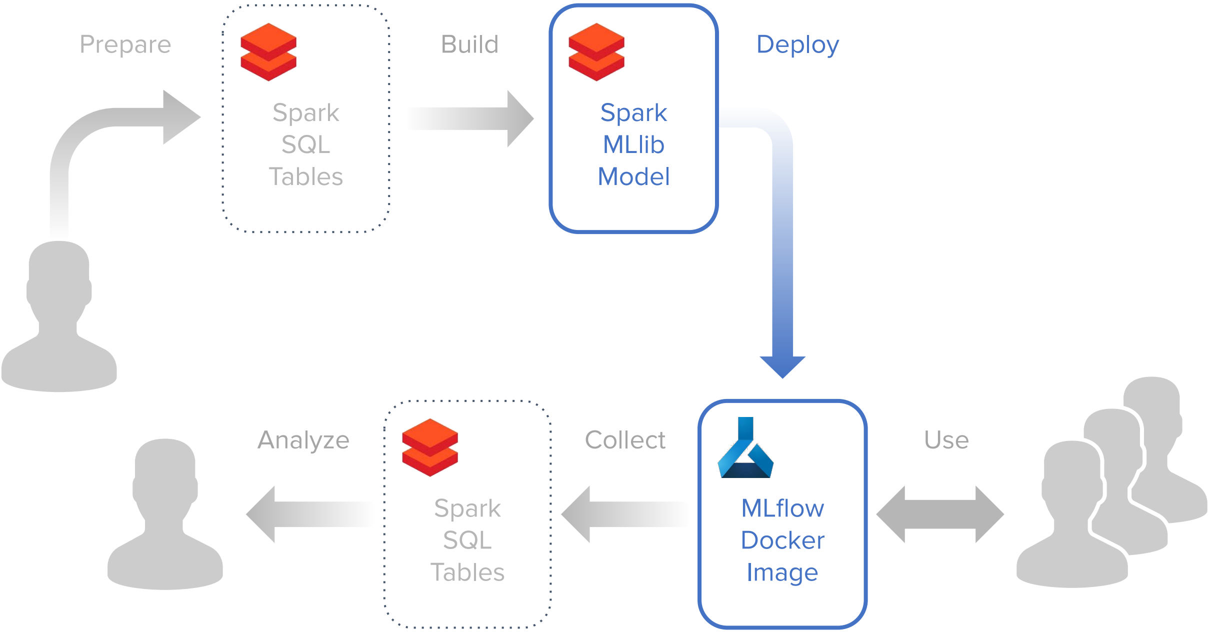


Deploying The Model

Connect to Azure ML Workspace

Build MLflow docker model image

Deploy docker model image to ACI/AKS

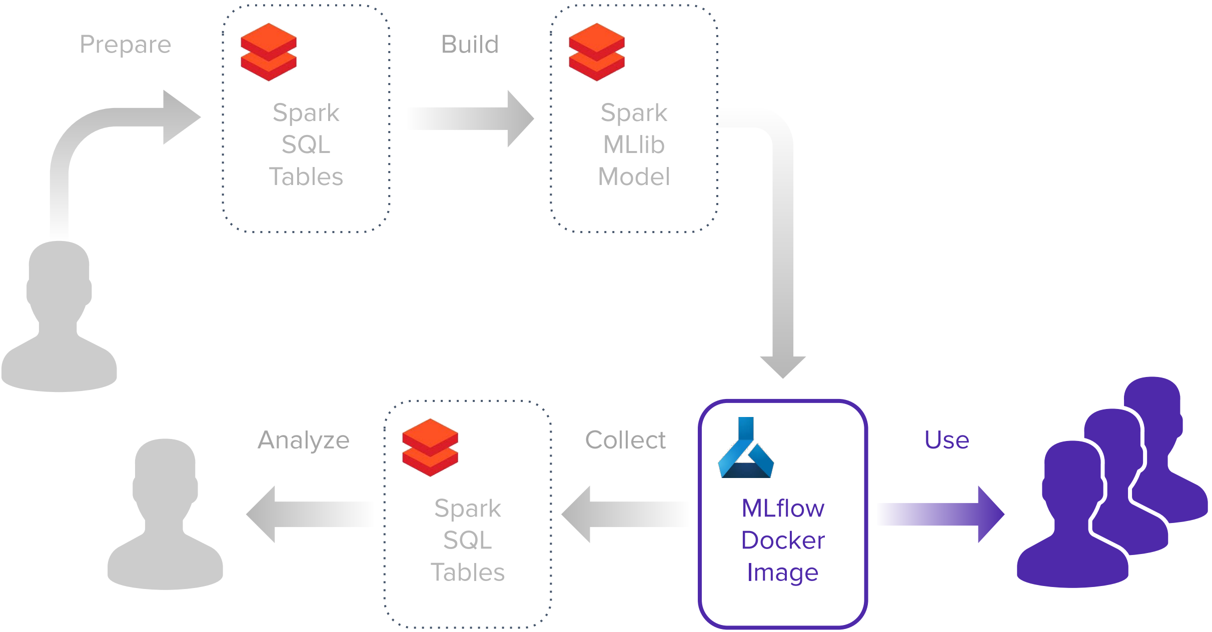


Using The Model

Load operations dataset

Call locally saved model

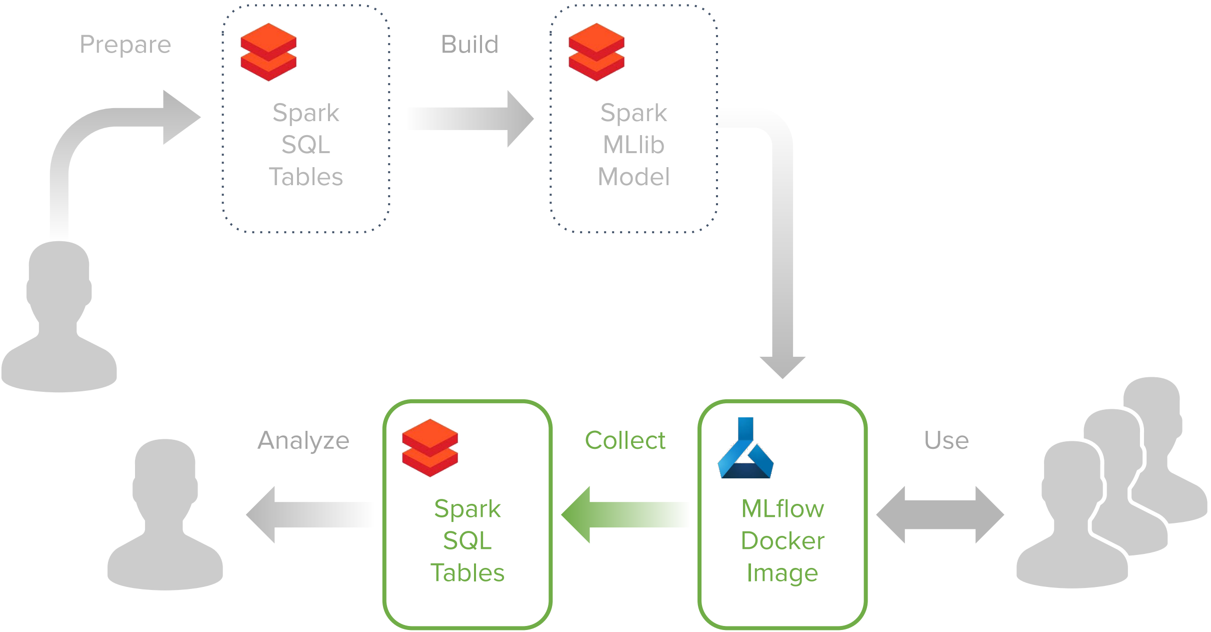
Call model deployed to ACI/AKS



Collecting Model Usage

Load input data from deployed model

Save to Spark SQL Table

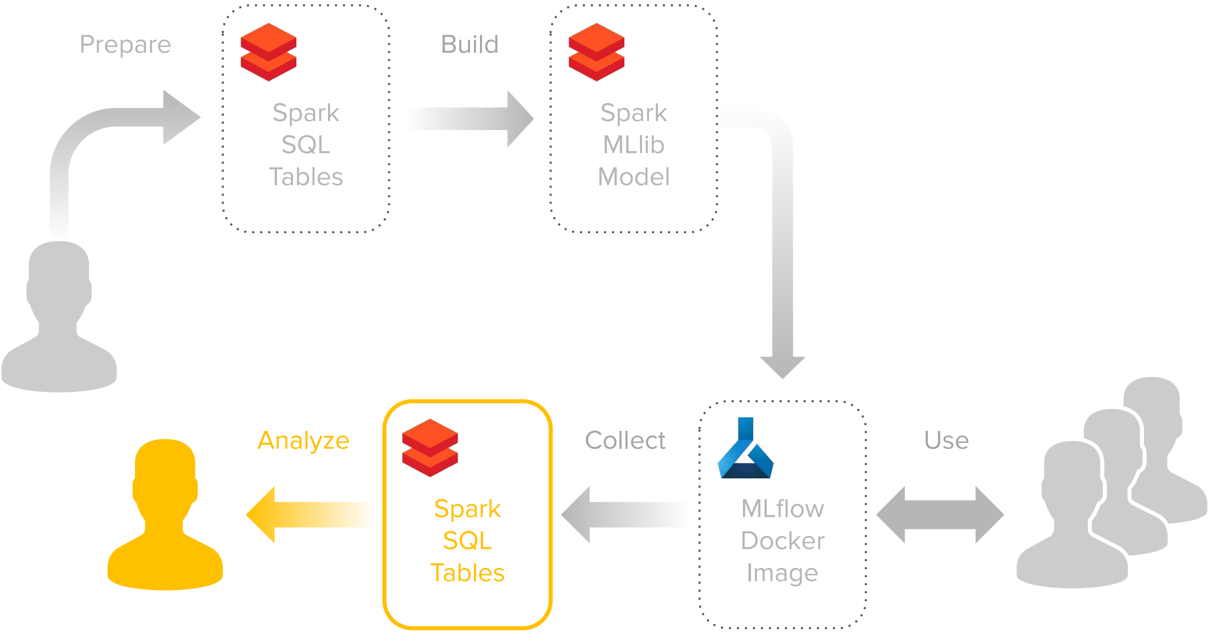


Analyzing Model Usage

Load data from model inputs

Extend preprocessing pipeline to construct drift classification

Measure model drift and identify features contributing to drift



Closing Remarks

What I showed you today

- → Experiment with my examples: https://git.io/Je05T
- → Try developing an operationalized model of your own.
- → Reach out for assistance: jhoff@productiveedge.com

Operationalizing is difficult!

- → No silver bullet solution
- → Any solution will have a lot of moving parts
- → Diversity of skills are required to succeed

11 E Illinois St Suite 200
Chicago, IL 60611, USA