

DSCI-272: Predicting with Cloudera Machine Learning



Introduction

Course Chapters

- **Introduction to CML**
 - **Introduction to AMPs and the Workbench**
 - **Data Access and Lineage**
 - **Data Visualization in CML**
 - **Experiments**
 - **Running a Spark Applications**
 - **Inspecting a Spark DataFrame**
 - **Transforming DataFrames**
 - **Transforming DataFrame Columns**
 - **User-Defined Functions**
-
- **Reading and Writing DataFrames**
 - **Combining and Splitting DataFrames**
 - **Summarizing and Visualizing DataFrames**
 - **Exploring and Visualizing DataFrames**
 - **Monitoring, Tuning and Configuring Spark Applications**
 - **Filtering and Evaluating Classification Models**
 - **Tuning Algorithm Hyperparameters Using Grid Search**
 - **Applying a Scikit-Learn Model to a Spark DataFrame**

Course Chapters

- **Working with Machine Learning Pipelines**
- **Deploying a Machine Learning Model as a REST API**
- **Autoscaling, Performance, and GPU Settings**
- **Model Metrics and Monitoring**
- **Appendix: Workspace Provisioning**

Trademark Information

- The names and logos of Apache products mentioned in Cloudera training courses, including those listed below, are trademarks of the Apache Software Foundation

Apache Accumulo	Apache Hadoop	Apache Kudu	Apache Ranger
Apache Avro	Apache HBase	Apache NiFi	Apache Solr
Apache Airflow	Apache Hive	Apache Oozie	Apache Spark
Apache Atlas	Apache Iceberg	Apache ORC	Apache Sqoop
Apache Bigtop	Apache Impala	Apache Ozone	Apache Tez
Apache Druid	Apache Kafka	Apache Parquet	Apache Zeppelin
Apache Flink	Apache Knox	Apache Phoenix	Apache ZooKeeper

- All other product names, logos, and brands cited herein are the property of their respective owners

Chapter Topics

Introduction

- **About This Course**
- **Introductions**
- **About Cloudera**
- **About Cloudera Educational Services**
- **Course Logistics**

Course Objectives

During this course, you will learn how to:

- Provision a Machine Learning workspace
- Utilize Cloudera SDX and other components of the Cloudera Data Platform to locate data for machine learning experiments
- Use an Applied ML Prototype (AMP)
- Manage machine learning experiments
- Connect to various data sources
- Visualize and explore data for ML
- Utilize Apache Spark Dataframes, user-defined functions, and Spark ML
- Deploy an ML model as a REST API
- Manage and monitor deployed ML model

Chapter Topics

Introduction

- About This Course
- **Introductions**
- About Cloudera
- About Cloudera Educational Services
- Course Logistics

Course Objectives

- **About your instructor**
- **About you**
 - Currently, what do you do at your workplace?
 - What is your experience with database technologies, programming, and query languages?
 - How much experience do you have with UNIX or Linux?
 - What is your experience with big data?
 - What do you expect to gain from this course?
 - What would you like to be able to do at the end that you cannot do now?

Chapter Topics

Introduction

- **About This Course**
- **Introductions**
- **About Cloudera**
- **About Cloudera Educational Services**
- **Course Logistics**



We believe that **data** can make what is impossible today, possible tomorrow

We empower people to transform complex **data anywhere** into actionable insights faster and easier

We deliver a hybrid data platform with secure **data management** and portable cloud-native **data analytics**

“The **future data ecosystem** should leverage distributed data management components — which may **run on multiple clouds and/or on-premises** — but are **treated as a cohesive whole** with a high degree of automation. **Integration, metadata and governance capabilities** glue the individual components together.”

Gartner®

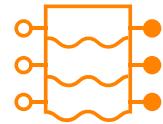
Strategic Roadmap for Migrating Data Management to the Cloud
Published 21 March 2022 - ID G00746011, Analyst(s): Robert Thanaraj, Adam Ronthal, Donald Feinberg

A More Specific Definition of Hybrid

Hybrid = Freedom to move existing and future applications, data, and users bi-directionally between the data center and multiple public clouds



Applications include streams, pipelines, workloads, workspaces, and other data products



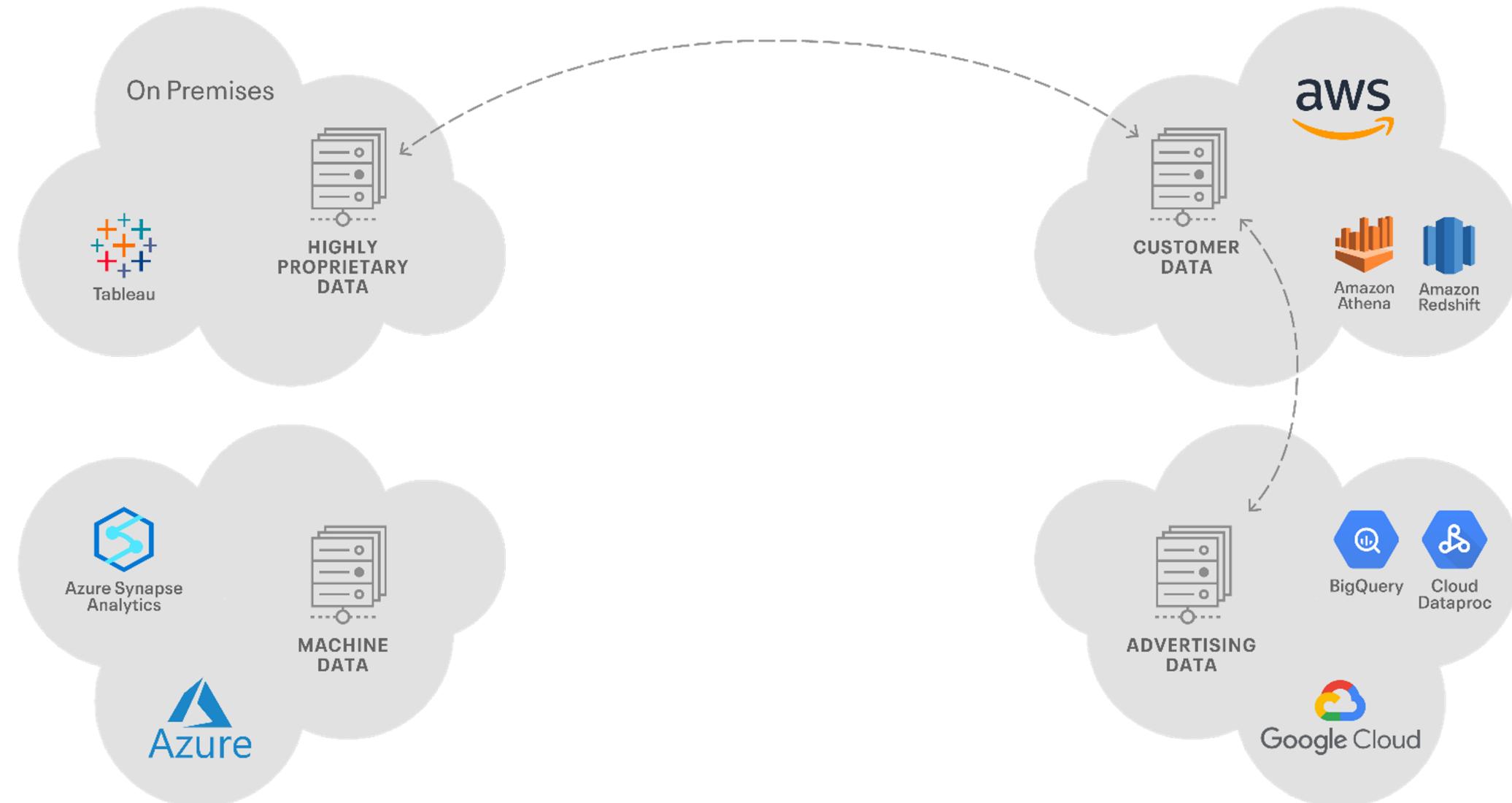
Data considerations include formats, policies, metadata, replication etc.



Users need to leverage existing skills and processes, while continuously modernizing

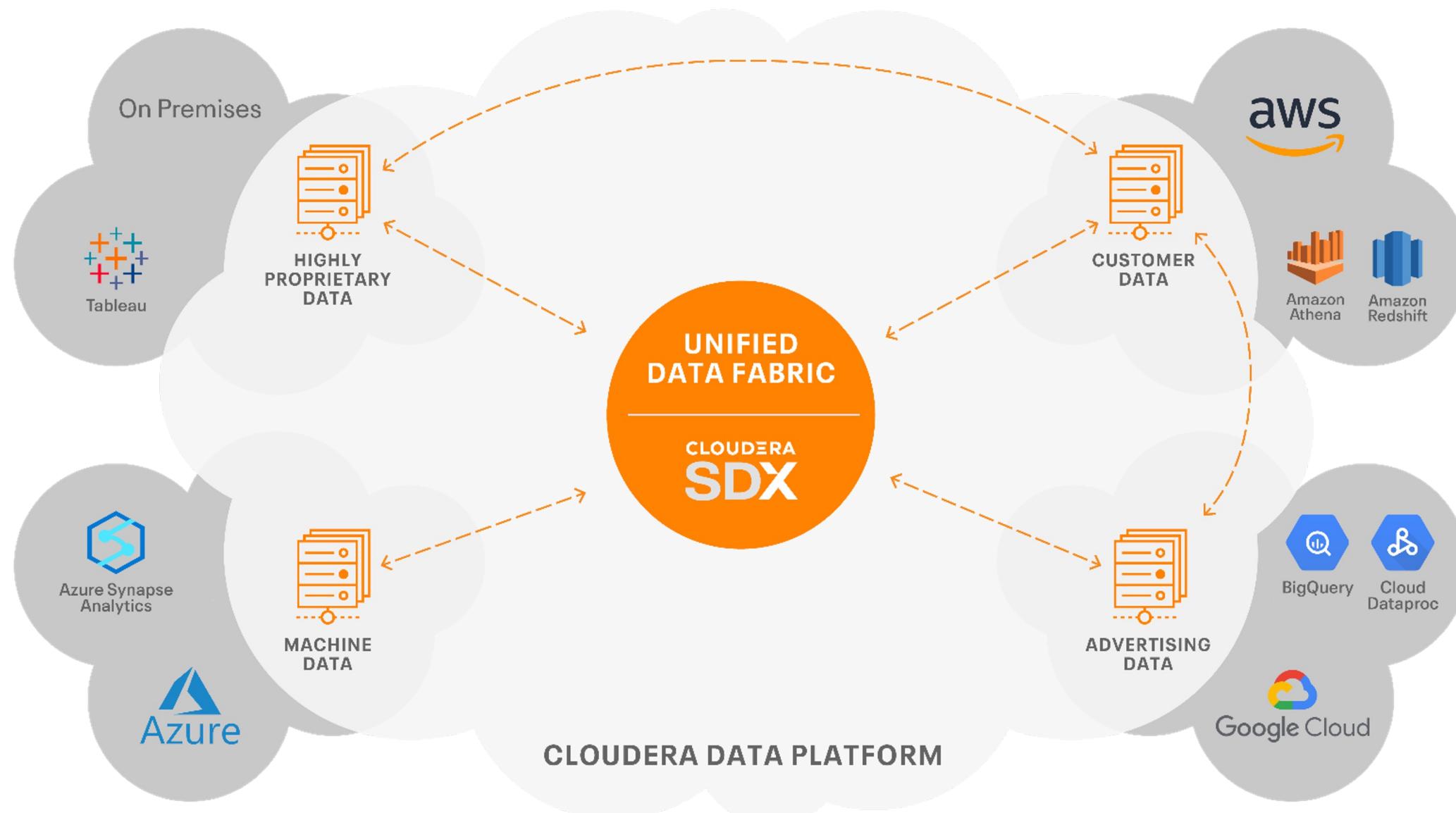
Cloudera's Hybrid Data Platform

Portable – Interoperable – Open – Secure



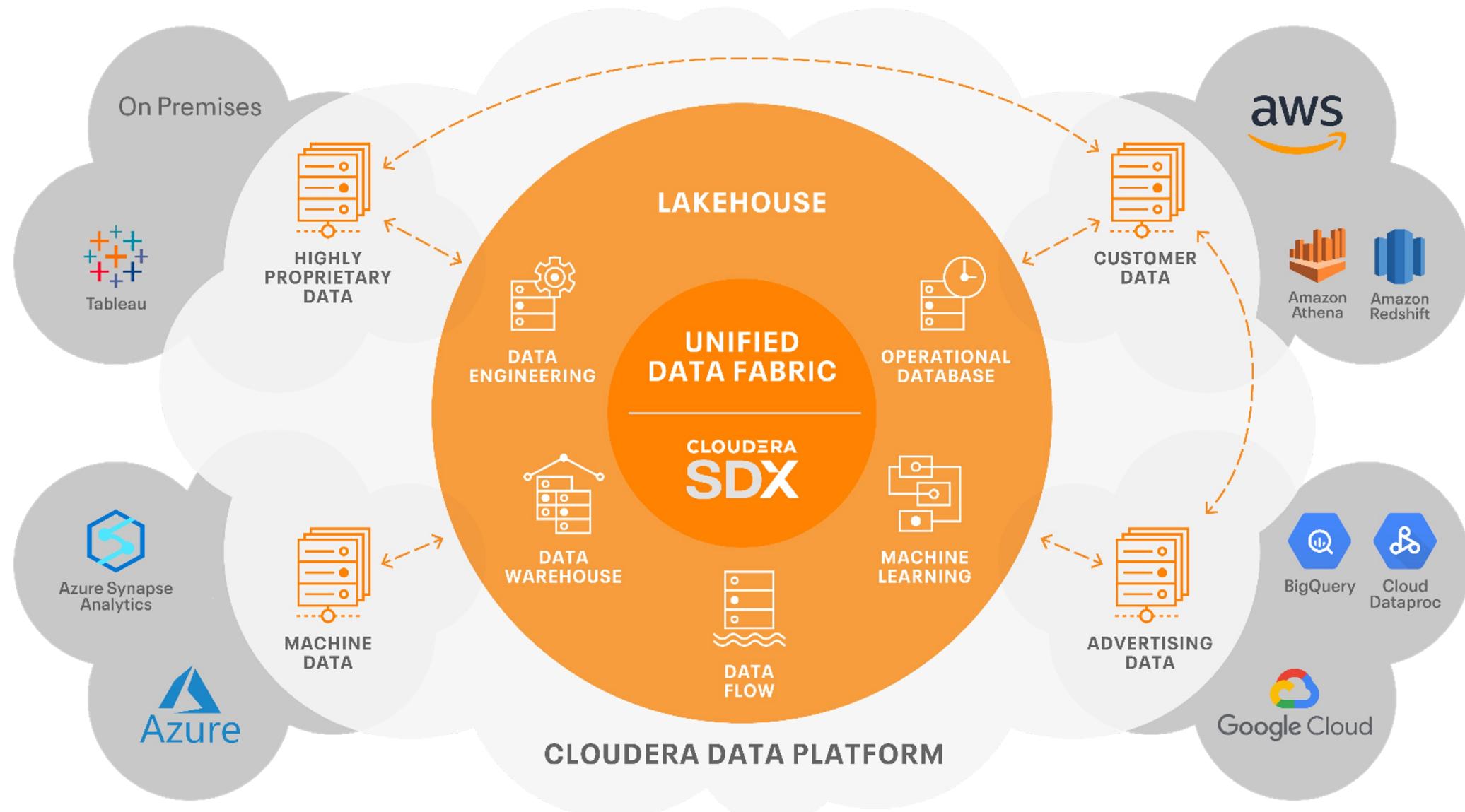
Cloudera's Hybrid Data Platform (2)

Portable – Interoperable – Open – Secure



Cloudera's Hybrid Data Platform (3)

Portable – Interoperable – Open – Secure

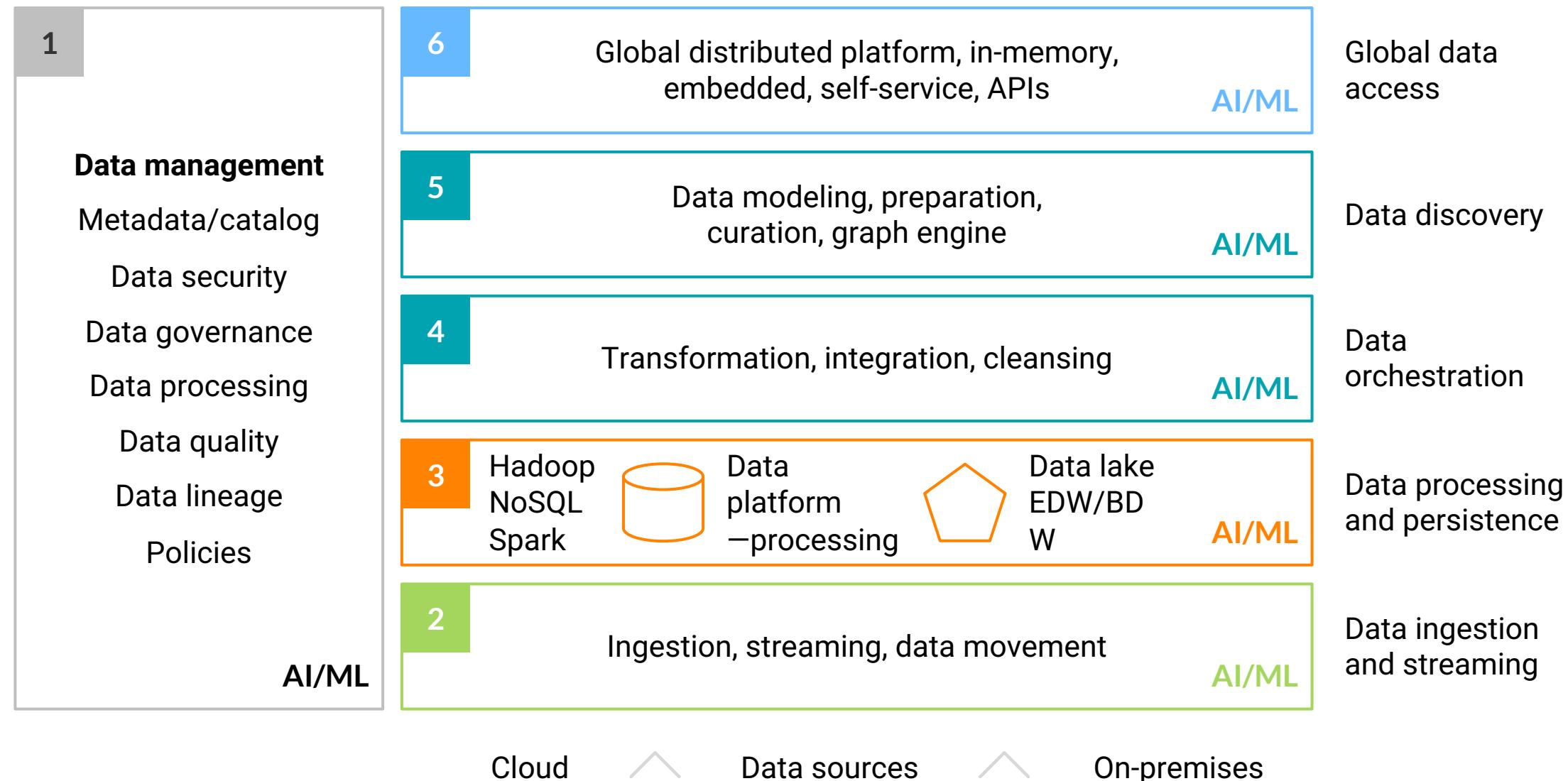


Forrester's Definition

A **Data Fabric** orchestrates disparate data sources intelligently and securely in a self-service manner, leveraging data platforms such as data lakes, data warehouses, NoSQL, translytical, and others to deliver a unified, trusted, and comprehensive view of customer and business data across the enterprise to support applications and insights.

CDP Hybrid Data Platform – Enabling Data Fabric (2)

Forrester's Definition



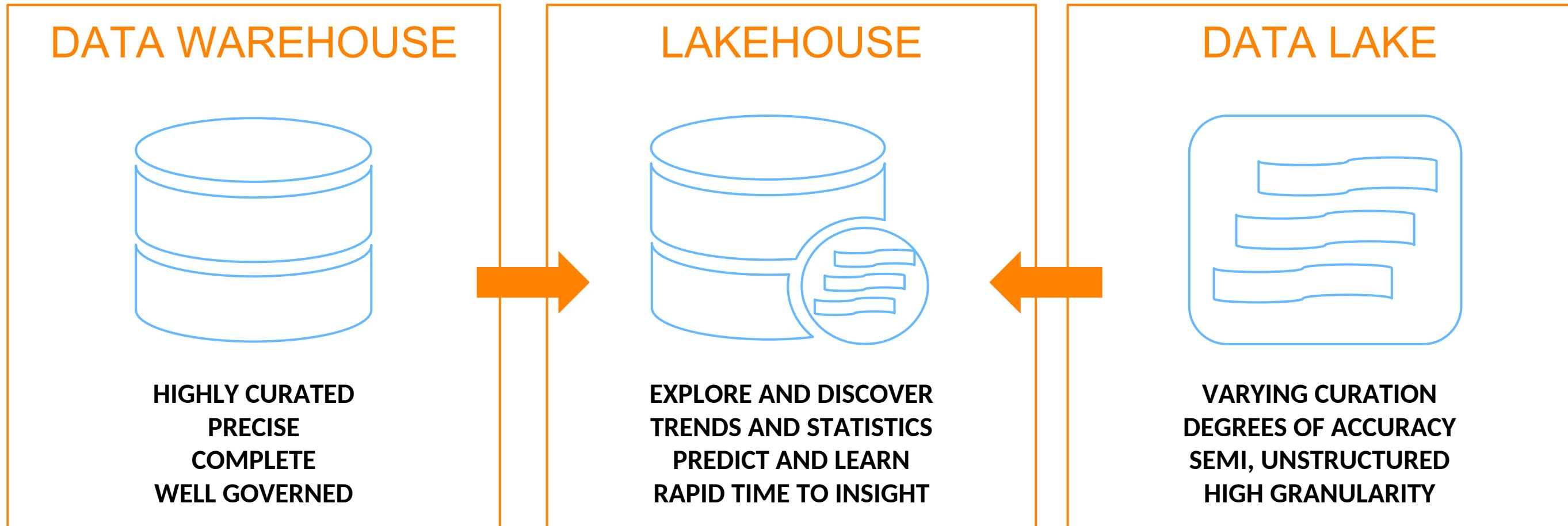
CDP Hybrid Data Platform – Enabling Data Lakehouse

Gartner's Definition

Data lakehouses integrate and unify the capabilities of data warehouses and data lakes, aiming to support AI, BI, ML and data engineering (“mulfunction analytics”) on a single platform.

The “Lakehouse” – Best of Both Worlds

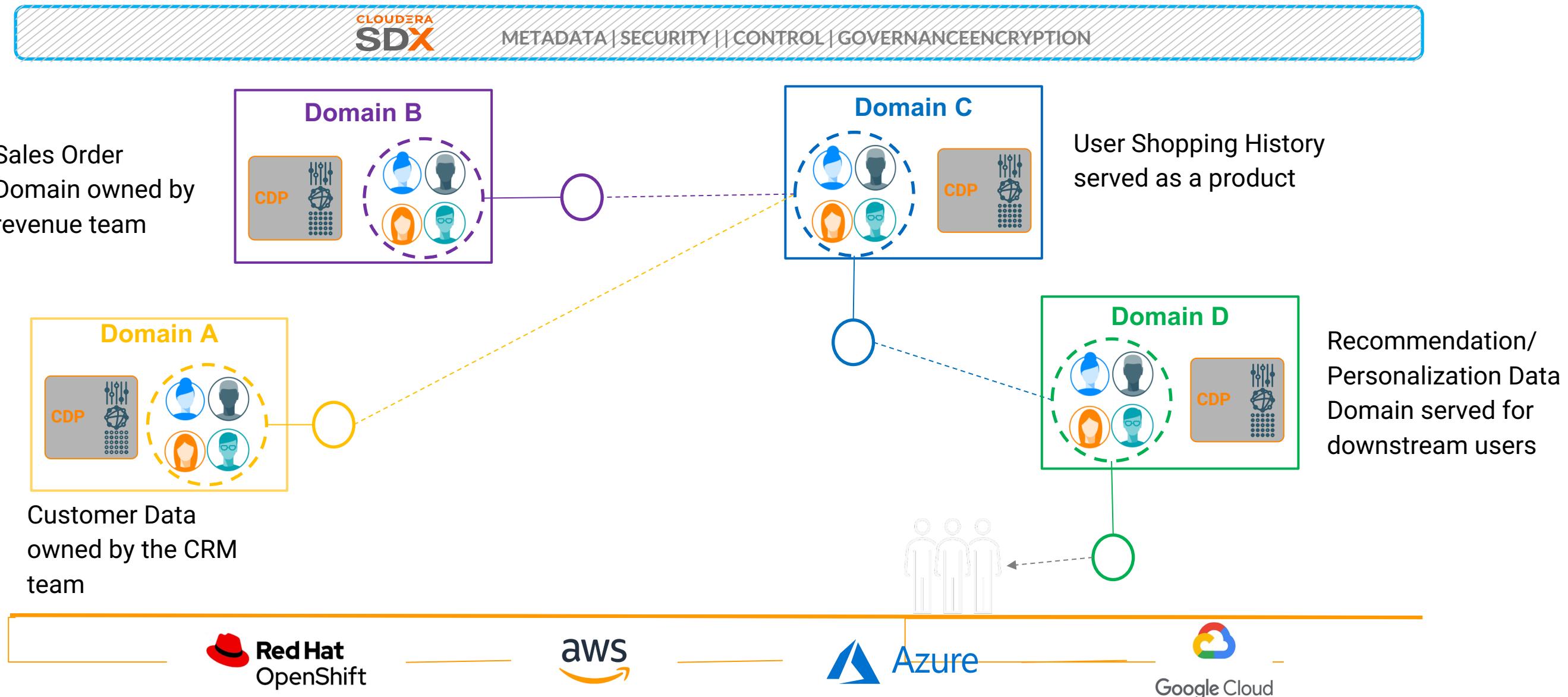
When ALL data needs to come together to answer critical business challenges



Data as a Product

Distributed data products oriented around domains and owned by independent cross-functional teams who have embedded data engineers and data product owners, using common data infrastructure as a platform to host, prep and serve their data assets.

CDP Hybrid Data Platform – Enabling Data Mesh (2)



Chapter Topics

Introduction

- About This Course
- Introductions
- About Cloudera
- **About Cloudera Educational Services**
- Course Logistics

Cloudera Educational Services

- **We offer a variety of ways to take our courses**
 - Instructor-led, both in physical and virtual classrooms
 - Private and customized courses also available
 - Self-paced, through Cloudera OnDemand
- **Courses for all kinds of data professionals**
 - Executives and managers
 - Data scientists and machine learning specialists
 - Data analysts
 - Developers and data engineers
 - System administrators
 - Security professionals

Cloudera Education Catalog

- A broad portfolio across multiple platforms
 - Not all courses shown here
- See [our website](#) for the complete catalog

Administrator	Administrator	Administrator Private Cloud Basic	Administrator CDP Public Cloud	Security Administration	Upgrading HDP/CDH to CDP							
Data Steward	Data Governance											
Data Analyst	Analyst Hive/Impala	CDP Data Visualization										
Developer Data Engineer	Spark Development	Spark Performance Tuning	NiFi Flow Management	Kafka/Stream Processing	Architecture	HBase	Flink					
Data Scientist	CDSW	Data Science	CML									
General	OnDemand Library CDP CDF	“CDP” Essentials, AWS Fundamentals	“Just Enough” Python, GIT, Scala	Tech Overviews	Live	OnDemand						

Cloudera OnDemand

- **Our OnDemand catalog includes**
 - Courses for developers, data analysts, administrators, and data scientists
 - Exclusive OnDemand-only courses, such as those covering CDP Public Cloud Administration and Cloudera Data Science Workbench
 - Free courses
- **Features include**
 - Video lectures and demonstrations
 - Hands-on exercises through a browser-based virtual environment
- **Purchase access to a library of courses or individual courses**
 - [Full OnDemand Library subscription](#)

Accessing Cloudera OnDemand

- Cloudera OnDemand subscribers can access their courses online through a web browser

The screenshot shows the Cloudera OnDemand web interface. At the top, there's a navigation bar with the Cloudera logo, a search bar, and various user icons. Below the header, a large orange banner says "Welcome to Cloudera University". It contains a brief description of the service and a "Read More" button. To the right of the banner is a video thumbnail featuring a person in a white jumpsuit. The main content area has two sections: "Total Number of Courses" (9 Enrolled Courses, 5 Completed Courses, 1 Learning Paths) and "Recent Activity" (three entries showing course visits). To the right, there are two course cards: "Just Enough Git" (In Progress, 6 Modules, 33% completed) and "Developer Training for Apache Spark and Hadoop" (In Progress, 20 Modules, 0% completed).

CDP Certification Program

- **New role-based certifications**
 - CDP Certified Generalist
 - CDP Certified Administrator
 - Private Cloud
 - Public Cloud
 - CDP Certified Developer
 - CDP Certified Analyst
- **Convenient online, proctored exams**
- **Digital badges**
- **www.cloudera.com/about/training/cdp-certification.html**



Chapter Topics

Introduction

- **About This Course**
- **Introductions**
- **About Cloudera**
- **About Cloudera Educational Services**
- **Course Logistics**

Logistics

- Class start and finish time
- Lunch
- Breaks
- Restrooms
- Wi-Fi access
- Virtual machines

Downloading the Course Materials

1. Log in using <https://education.cloudera.com/>

- Click **Sign In** on the top right
 - If necessary, create an account
 - If you have forgotten your password, use the **Forgot your password** link

2. Locate the course

- Find the course in your list of enrollments, or enter the course title in the search bar
- Click on the course title

3. Access the materials

- Click on **Content** across the top
- Click on the module to view or download the contents
- If there is more than one module, use the Course Contents panel on the left to navigate

The screenshot shows the 'Course Contents' page for the course 'ILT - Cloudera DE: Developing Applications with Apache Spark - 2764504'. At the top, there are icons for a grid, a list, and an envelope. Below that is a 'Return to Dashboard' button. The course title and ID are displayed prominently. A 'My Progress' bar shows 33% completion. The main content area lists three modules: 'Developing Applications Notebooks (210831)', 'Developing Applications Student Slides (210831)', and 'ILT - Cloudera DE: Developing Applications with Apache Spark - 2764504'. The third module is currently selected, indicated by a checkmark icon.



Introduction to CML

Introduction to CML

By the end of this chapter, you will be able to

- **Describe the benefits of CML**
- **Discuss the differences between CML and CDSW**
- **Create a project within an ML workspace**
- **Set user roles**
- **Work in projects and teams**
- **Understand the flexibility of ML Runtimes**
- **Provision a workspace**

Chapter Topics

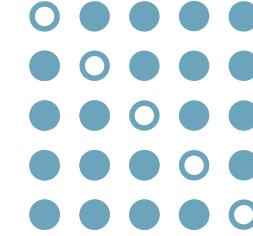
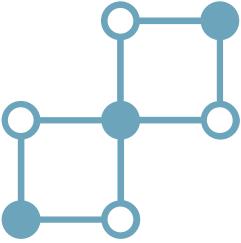
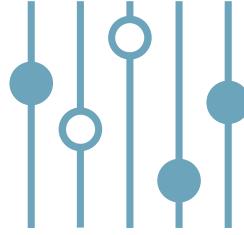
Introduction to CML

- **Overview**
- **CML Versus CDSW**
- **ML Workspaces**
- **Workspace Roles**
- **Projects and Teams**
- **Settings**
- **Runtimes/Legacy Engines**
- **Exercise Overview**
- **Essential Points**
- **Hands-On Exercise: Getting Started with CML**

Cloudera Machine Learning is a comprehensive platform to collaboratively build and deploy machine learning capabilities at scale

- **Cloudera Machine Learning (CML) is part of the Cloudera Data Platform (CDP)**
 - Enables enterprise data science teams to collaborate across the full data lifecycle
 - Provides immediate access to enterprise data pipelines, scalable compute resources, and access to preferred tools.
 - Optimizes ML workflows across your business with native and robust tools for deploying, serving, and monitoring models
- **To use CML, open a web browser and sign into CDP**
 - Use your organization's single sign-on (SSO) system

Accelerate Machine Learning from Research to Production



ANALYZE DATA

Explore data securely and share data **insights** with the team

TRAIN MODELS

Run, track, and compare reproducible **experiments**

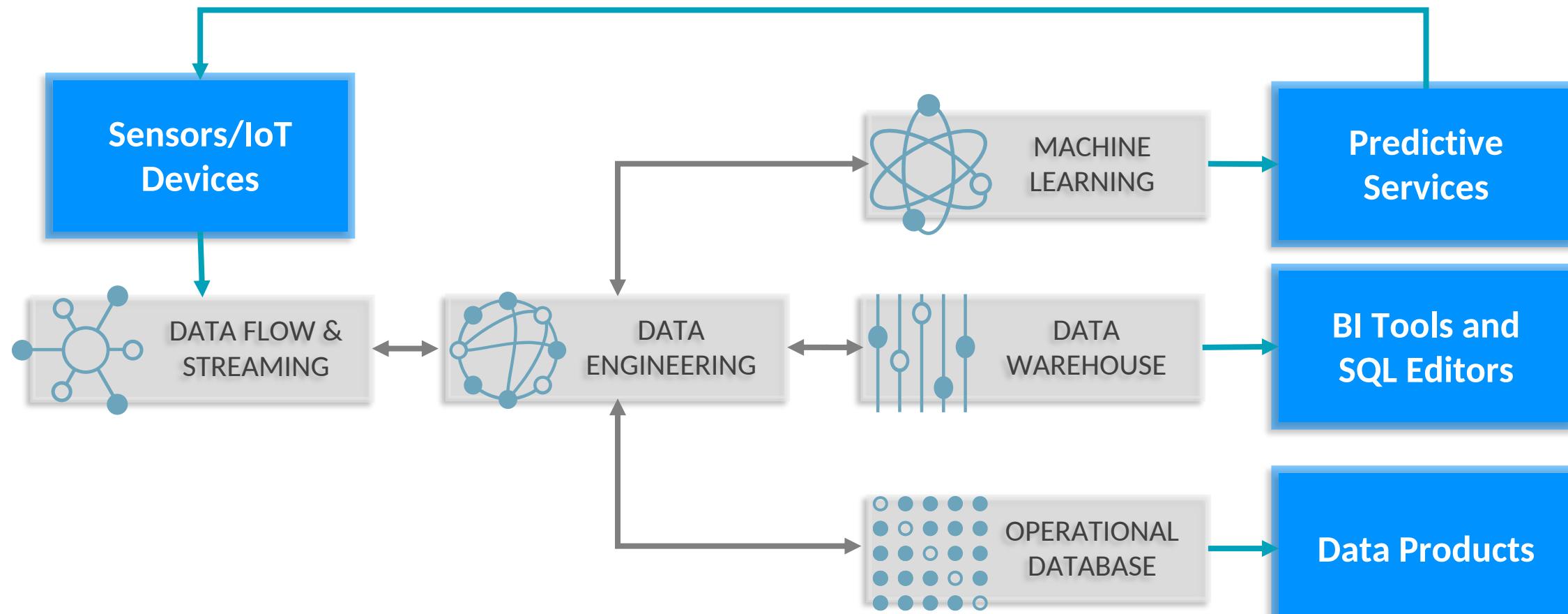
DEPLOY APIs

Deploy and monitor models as APIs to **serve predictions**

MANAGE SHARED RESOURCES

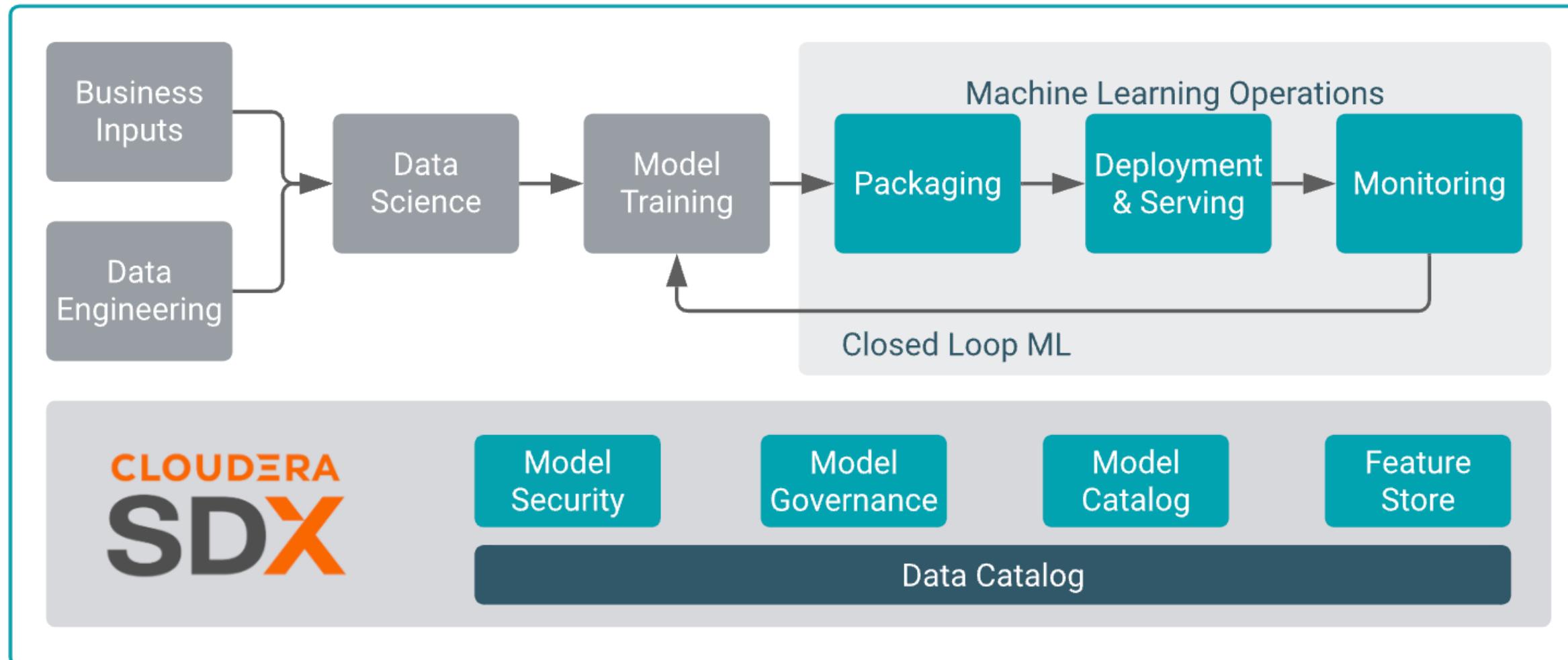
Provide a secure, collaborative, **self-service platform** for data science teams

Useful Enterprise Machine Learning is More Than ML

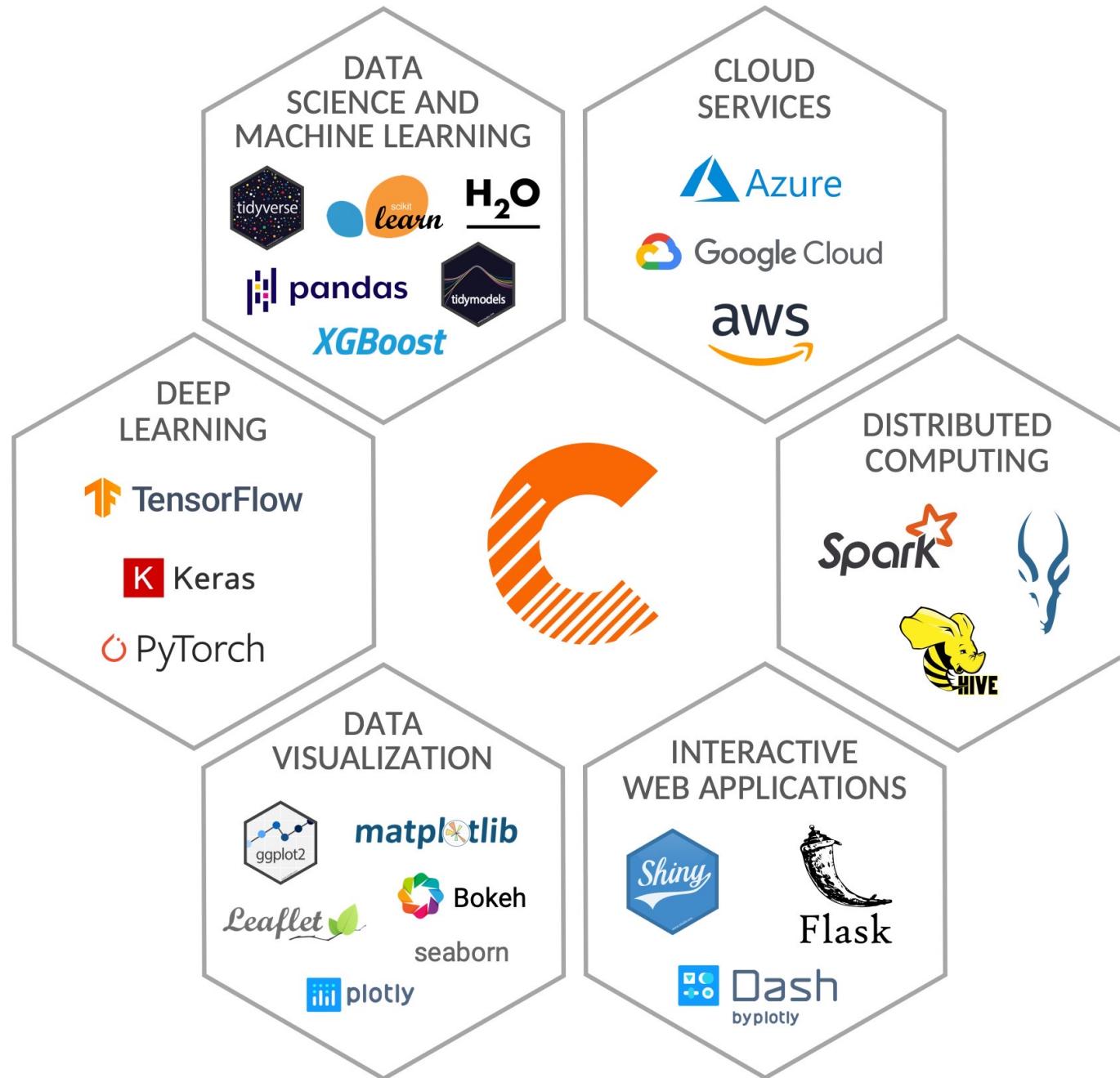


DATA, METADATA, SECURITY, GOVERNANCE, WORKLOAD MANAGEMENT

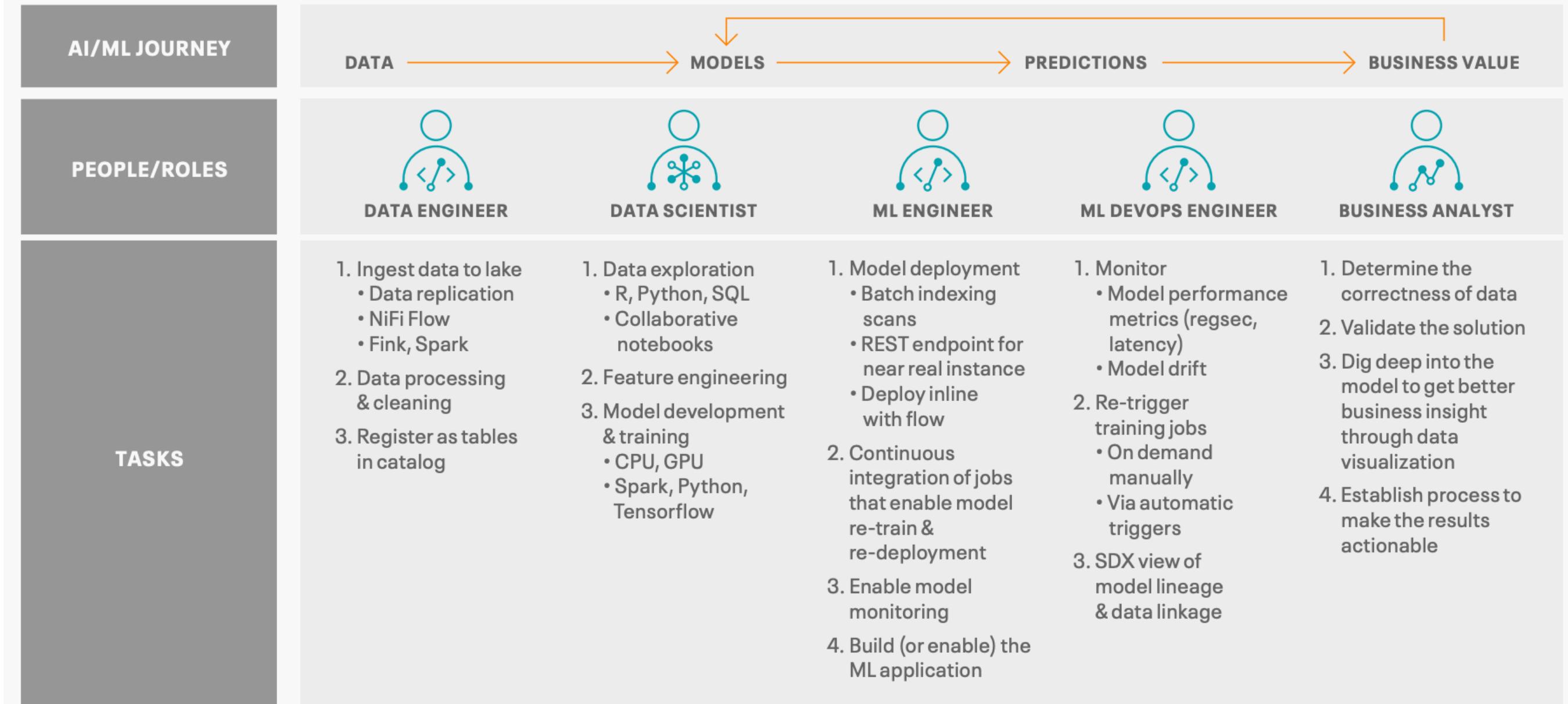
ML Workflow and Operations



What CML Can Help You Do



AI/ML Journey



Chapter Topics

Introduction to CML

- Overview
- **CML Versus CDSW**
- ML Workspaces
- Workspace Roles
- Projects and Teams
- Settings
- Runtimes/Legacy Engines
- Exercise Overview
- Essential Points
- Hands-On Exercise: Getting Started with CML

How is Cloudera Machine Learning (CML) related to Cloudera Data Science Workbench (CDSW)?

- **CML expands the end-to-end workflow of Cloudera Data Science Workbench (CDSW) with cloud-native benefits such as**
 - Rapid provisioning
 - Elastic autoscaling
 - Distributed dependency isolation
 - Distributed GPU training
- **Both products help data engineers and data science teams be more productive on shared data and compute, with strong security and governance**

Differences Between CML and CDSW

CDSW	CML Private Cloud	CML Public Cloud
Operates in a Kubernetes sidecar connected to CDH/HDP	Operates in the customer's own Kubernetes-based private cloud (e.g. OpenShift)	Operates in the customer's public cloud, using EKS in AWS and AKS in Azure
No autoscaling; resources are dedicated to CDSW	Shared resource pool; resources limited by the private cloud environment	Cloud-based autoscaling
Distributed compute workloads are pushed to the CDH/HDP cluster (Spark-on-YARN)	Workloads run on the dedicated Kubernetes cluster separate from CDP-CD (Spark-on-K8S)	Workloads run on the dedicated Kubernetes cluster separate from CDP (Spark-on-K8S)
<i>Close to End-of-Life</i>	Where all the R&D goes	Where all the R&D goes

Chapter Topics

Introduction to CML

- Product Overview
- CML Versus CDSW
- **ML Workspaces**
- Workspace Roles
- Projects and Teams
- Settings
- Runtimes/Legacy Engines
- Exercise Overview
- Essential Points
- Hands-On Exercise: Getting Started with CML

ML Workspaces

- Each ML workspace enables collaborative teams of data scientists to
 - Develop, test, train, and deploy machine learning models
 - Build predictive applications that can be used within the organization

The screenshot shows the Cloudera Management Console interface. On the left is a dark sidebar with navigation links: Dashboard, Environments, Data Lakes, User Management, Data Hub Clusters, Data Warehouses, **ML Workspaces** (which is currently selected and highlighted in blue), and Classic Clusters. The main content area is titled "Machine Learning Workspaces". It features a search bar labeled "Search Workspaces", a dropdown menu for "Environment" set to "All", and a large blue button labeled "Provision Workspace". Below these are two tables. The first table lists workspaces with columns for Status (green checkmark or red exclamation mark), Workspace Name, Environment, Creation Date, and Actions (three vertical dots). The second table lists environments with columns for Environment Name, Status (green checkmark or red exclamation mark), Creation Date, and Actions (three vertical dots). At the bottom right, there is a pagination indicator showing "Displaying 1 - 10 of 10" with a page number "1" in a blue box.

Status	Workspace Name	Environment	Creation Date	Actions
✓	irbcm1	ml-demo-env-6jan	01/13/2020 12:38 PM PST	⋮
✓	regtest	mlx-dev-prod-env	01/13/2020 10:24 AM PST	⋮
!	ram-CB-4990	sense-prod-env	01/10/2020 10:40 AM PST	⋮
✓	ml-demo-wksp-7jan	ml-demo-env-6jan	01/07/2020 6:29 AM PST	⋮
✓	eng-cml-cluster	mlx-dev-prod-env	01/06/2020 2:21 PM PST	⋮
✓	OASC-test	ml-demo-env-3dec	12/10/2019 12:03 PM PST	⋮
!	nallen-demo-workspace	testenv	12/06/2019 10:24 AM PST	⋮
✓	bigdatum-MLW	bigdatum-environment	12/03/2019 10:43 PM PST	⋮
!	ml-demo-wksp-3dec	ml-demo-env-3dec	12/03/2019 11:04 AM PST	⋮
✓	dnarain-ml-aps1	lord-of-the-rules-aps1	08/27/2019 11:17 AM PDT	⋮

Environment Name	Status	Creation Date	Actions
mlx-dev-prod-env	✓	01/13/2020 10:24 AM PST	⋮
sense-prod-env	!	01/10/2020 10:40 AM PST	⋮
ml-demo-env-6jan	✓	01/13/2020 12:38 PM PST	⋮
mlx-dev-prod-env	✓	01/06/2020 2:21 PM PST	⋮
ml-demo-env-3dec	✓	12/10/2019 12:03 PM PST	⋮
testenv	!	12/06/2019 10:24 AM PST	⋮
bigdatum-environment	✓	12/03/2019 10:43 PM PST	⋮
ml-demo-env-3dec	!	12/03/2019 11:04 AM PST	⋮
lord-of-the-rules-aps1	✓	08/27/2019 11:17 AM PDT	⋮

Elastic Kubernetes Cluster

- Each ML Workspace is an elastic Kubernetes cluster

Advanced Options

CPU Settings

* Instance Type

Select an Environment to see options.

Autoscale Range

 0 5 10

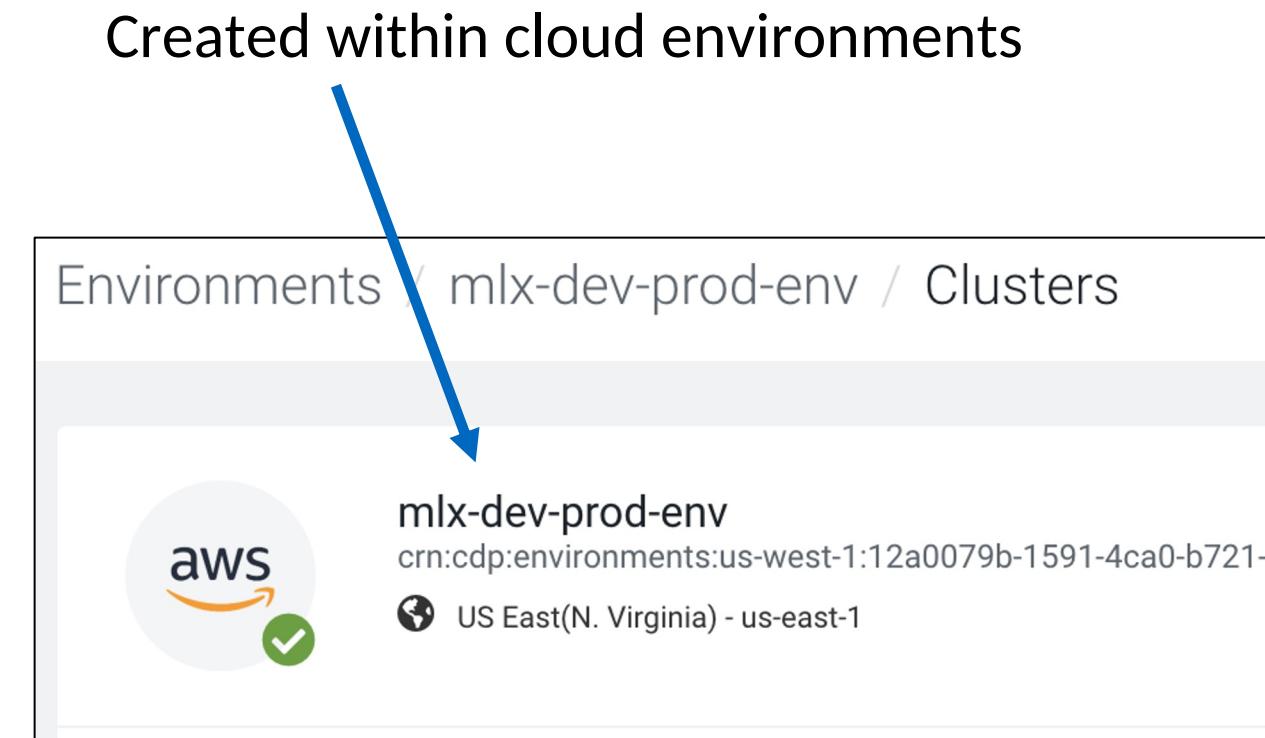
GPU Settings

Instance Type

Select an Environment to see options.

Autoscale Range

 0 10



Accessing a Workspace

The screenshot shows the CDP web interface with the following components:

- Header:** "Your Enterprise Data Cloud"
- Top navigation icons:** Data Hub Clusters (pink), DataFlow (purple), Data Engineering (blue), Data Warehouse (light blue), Operational Database (teal), and Machine Learning (green). The Machine Learning icon is highlighted with a red box.
- Sub-navigation:** "Machine Learning Workspaces" with a green sidebar icon.
- Search and filters:** "Search Workspaces" input, "Environment" dropdown set to "All", and "Provision Workspace" button.
- Table:** A list of workspaces with columns: Status, Workspace, Environment, Region, Creation Date, Cloud Provider, and Actions. One row is shown:

Status	Workspace	Environment	Region	Creation Date	Cloud Provider	Actions
Ready	edu-cml	bshimmel-dev-us-east-2	us-east-2	06/20/2022 12:52 PM EDT	AWS	⋮
- Pagination:** "Displaying 1 - 1 of 1" and "25 / page" dropdown.

- Log in to the CDP web interface <https://console.cdp.cloudera.com>
- Click **Machine Learning**
- Click the link for desired workspace

Projects

- One or more projects are created within an ML Workspace

The screenshot shows the Cloudera Machine Learning interface with the 'Projects' page selected. The left sidebar includes options like Projects, Sessions, Experiments, Models, Jobs, Applications, User Settings, AMPs, Runtime Catalog, Site Administration, and Learning Hub. The main area displays 'Active Workloads' with counts: SESSIONS 0, EXPERIMENTS 0, MODELS 1, JOBS 0, and APPLICATIONS 7. It also shows 'User Resources' and 'Workspace Resources' for CPU, Memory, and GPU. Below this, a grid of project cards is shown:

Project Name	Created by	Last worked on
CDP on CML	dev_20_22829	17 hours ago
Churn Analysis Refact...	dev_20_22829	a day ago
Continuous Model Mo...	dev_20_22829	3 days ago
Evidently Test	dev_20_22829	14 days ago
Deep Learning with GP...	dev_20_22829	15 days ago
Duocar 20	dev_20_22829	15 days ago
Student 20	dev_20_22829	15 days ago

Pagination controls at the bottom right indicate page 1 of 25.

A Project Contains...

- A project is a directory structure containing code, data, and assets
- Project code is commonly written in Python, R, or Java

```
Seaborn Analysis
analysis.py
File Edit View Navigate Run
analysis.py

1 # Setup
2 # -----
3
4 import pandas as pd
5 import seaborn as sns
6
7 # Basic Data Manipulation
8 # -----
9 #
10 # Use the seaborn tips dataset to generate a best fitting linear regression line
11
12 tips = sns.load_dataset("tips")
13 sns.set(font="DejaVu Sans")
14 sns.jointplot("total_bill", "tip", tips, kind='reg').fig.suptitle("Tips Regression", y=1.01)
15
16 # Examine the difference between smokers and non smokers
17 sns.lmplot("total_bill", "tip", tips, col="smoker").fig.suptitle("Tips Regression - categorized by smoker", y=1.05)
18
19 # Explore the dataframe
20 tips.head()
21
22 # Using IPython's Rich Display System
23 # -----
24 #
25 # IPython has a [rich display system](bit.ly/HHP0ac) for
26 # interactive widgets.
27
28 from IPython.display import IFrame
29 from IPython.core.display import display
30
31 # Define a google maps function.
32 def gmaps(query):
33     url = "https://maps.google.com/maps?q={0}&output=embed".format(query)
34     display(IFrame(url, '700px', '450px'))
35
36 gmaps("Golden Gate Bridge")
37
38 # Worker Engines
39 # -----
40 #
41 # You can launch worker engines to distribute your work across a cluster.
42 # Uncomment the following to launch two workers with 2 cpu cores and 0.5GB
43 # memory each.
44
45 # import cdsweb
46 # workers = cdsweb.launch_workers(n=2, cpu=0.2, memory=0.5, code="print('Hello from a CDSW Worker')")
```

Line 1, Column 1

★ 47 Lines Python Spaces 2

Managing Project Files

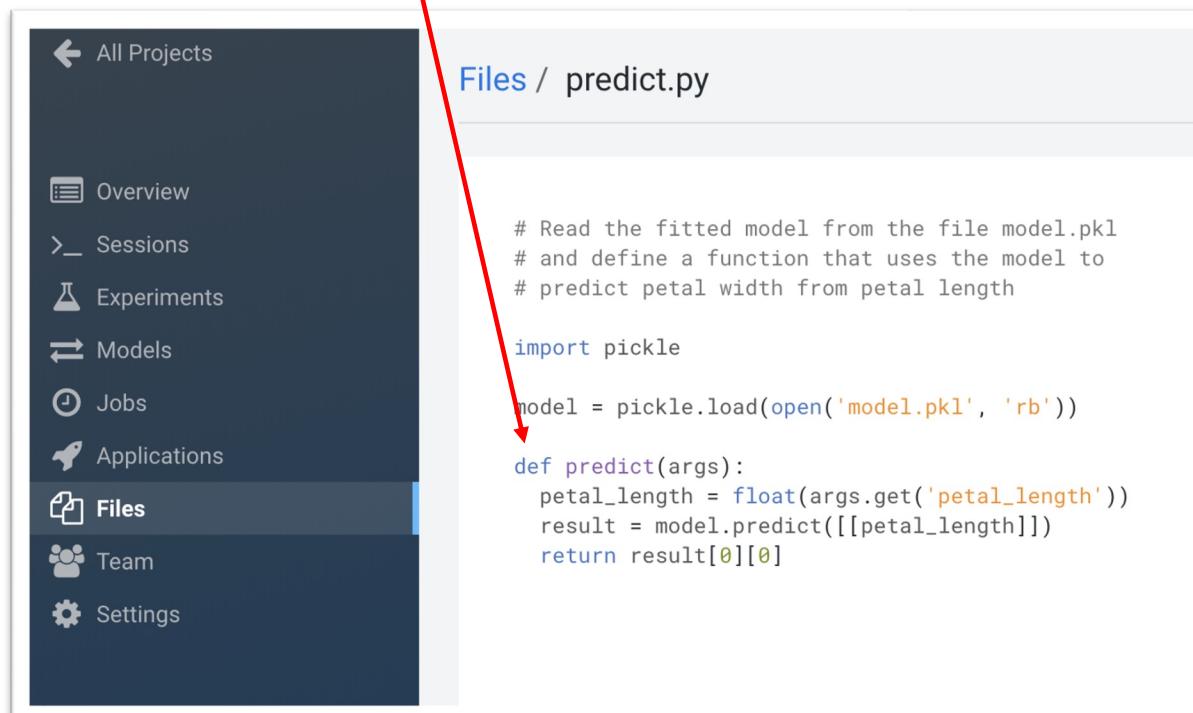
The screenshot shows a dark-themed user interface for managing project files. On the left, a vertical sidebar lists various project management sections: All Projects, Overview, Sessions, Data, Experiments, Models, Jobs, Applications, Files (which is currently selected and highlighted in green), Collaborators, Project Settings, and Help. The main area is titled 'Files' and displays a list of files and directories. The list includes:

- seaborn-data
- analysis.ipynb
- analysis.py
- cdsw-build.sh
- config.yml
- entry.py
- fit.py
- lineage.yaml
- pi.py
- predict.py
- predict_with_metrics.py
- README.md
- requirements.txt
- use_model_metrics.py

- **Move, rename, copy, and delete files within the scope of the project**
 - Upload new files to a project
 - Download project files
- **Be careful about deleting files and directories in CML!**
 - There is no way to undo the deletion

Models Are Deployed Within a Project

Write the function that makes the prediction...



All Projects

- Overview
- Sessions
- Experiments
- Models
- Jobs
- Applications
- Files
- Team
- Settings

Files / predict.py

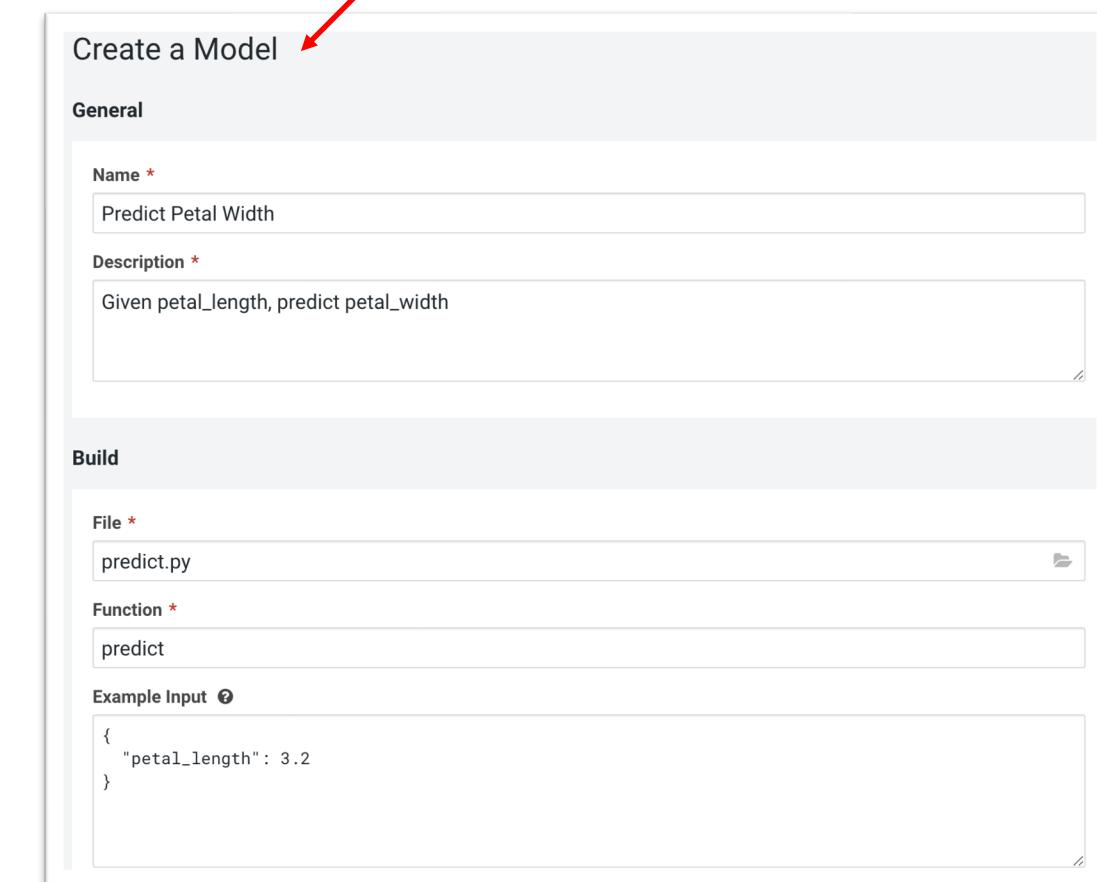
```
# Read the fitted model from the file model.pkl
# and define a function that uses the model to
# predict petal width from petal length

import pickle

model = pickle.load(open('model.pkl', 'rb'))

def predict(args):
    petal_length = float(args.get('petal_length'))
    result = model.predict([[petal_length]])
    return result[0][0]
```

...and then use CML to serve it up



Create a Model

General

Name *

Description *

Build

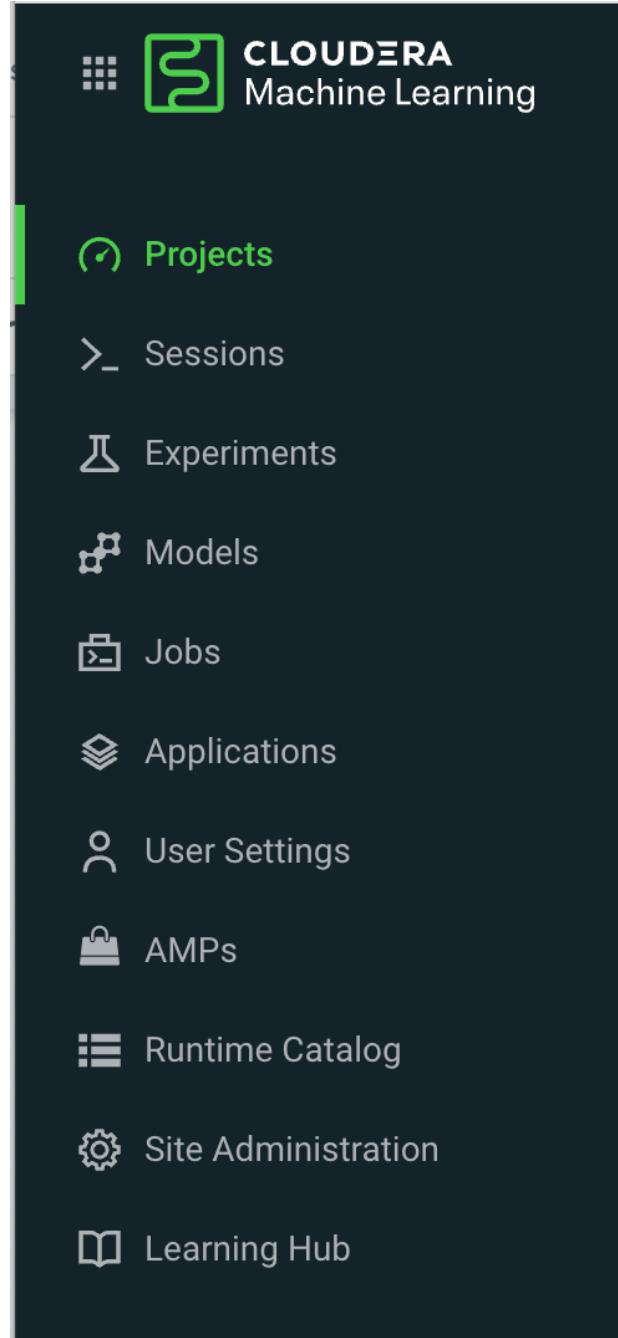
File *

Function *

Example Input ?

```
{ "petal_length": 3.2 }
```

Project Components



- **Sessions**
 - Directly leverage the CPU, memory, and GPU compute available across the workspace
- **Experiments**
 - Run multiple variations of model training workloads in order to train the best possible Model
- **Models**
 - Implementation of machine learning algorithms based on sample data
- **Jobs**
 - Orchestrate an entire end-to-end automated pipeline
- **Applications**
 - Deliver interactive experiences for business users

Chapter Topics

Introduction to CML

- Overview
- CML Versus CDSW
- ML Workspaces
- **Workspace Roles**
- Projects and Teams
- Settings
- Runtimes/Legacy Engines
- Exercise Overview
- Essential Points
- Hands-On Exercise: Getting Started with CML

User Roles

Environment

Provides access to CML within a given CDP environment

Workspace

Provides access to a single workspace

- **If a user has more than one role**
 - Role with highest level of permissions takes precedence
- **If a user is a member of more than one group**
 - Can gain additional roles

Environment Roles

■ Access to CML workspaces within a given CDP environment

MLAdmin

- Create and delete ML workspaces
- Administrator-level access to all workspaces in the environment
 - Run workloads
 - Monitor and manage user activity
 - Also needs the account-level role of IAMViewer in order to list users or assign resource roles

MLUser

- List ML workspaces provisioned
- Run workloads on all workspaces provisioned

MLBusinessUser

- View applications deployed under the projects they have been added to

Workspace Roles

- **Access to a specific CML workspace within a given CDP environment**

MLWorkspaceAdmin

- Manage all machine learning workloads and settings inside a specific workspace.
- Also needs the account-level role of IAMViewer in order to list users or assign resource roles.

MLWorkspace
BusinessUser

- View shared machine learning applications insides a specific workspace

MLWorkspaceUser

- Run machine learning workloads inside a specific workspace

Configuring User Access

- In the workspace, select Manage Access from the Actions menu

The screenshot shows the Cloudera Machine Learning workspace interface for a workspace named 'cml-edu'. The left sidebar has 'Workspaces' selected. The main area displays workspace details: Name (cml-edu), Environment / Region (ps-sandbox-aws / us-east-2), Filesystem ID (fs-00129cad4de821fa2), NFS Protocol version (unknown), CRN (crn:cdp:ml:us-west-1:558bc1d2-8867-4357-8524-311d51259233:workspace...), Workspace ID (ml-8250e328-99f), Cluster Name (liftie-jyr14lyr), Monitoring (Enabled), and TLS (Enabled). A red arrow points to the 'Actions' dropdown menu on the right, which includes options: Manage Access (highlighted), Manage Remote Access, Download Kubeconfig, Open Grafana, Upgrade Workspace, Backup Workspace, and Remove Workspace.

Machine Learning Workspaces / cml-edu

Status ✓ Ready

Details Events & Logs

Actions ▾

Name	cml-edu
Environment / Region	ps-sandbox-aws / us-east-2
Filesystem ID	fs-00129cad4de821fa2
NFS Protocol version	unknown
CRN	crn:cdp:ml:us-west-1:558bc1d2-8867-4357-8524-311d51259233:workspace...
Workspace ID	ml-8250e328-99f
Cluster Name	liftie-jyr14lyr
Monitoring	Enabled
TLS	Enabled

Manage Access

Manage Remote Access

Download Kubeconfig

Open Grafana

Upgrade Workspace

Backup Workspace

Remove Workspace

Updating a User's Roles

- Select the user to add or remove a role

Update Resource Roles for Sheryl Sarokas

X

Resource Roles		
<input type="button" value="-"/>	Role	Description
<input type="checkbox"/>	MLWorkspaceAdmin	Grants permission to manage all machine learning workloads and settings inside a specific workspace.
<input type="checkbox"/>	MLWorkspaceBusinessUser	Grants permission to view shared machine learning applications inside a specific workspace.
<input checked="" type="checkbox"/>	MLWorkspaceUser	Grants permission to run machine learning workloads inside a specific workspace.
<input type="checkbox"/>	Owner	Grants all permissions on the resource.

Chapter Topics

Introduction to CML

- Overview
- CML Versus CDSW
- ML Workspaces
- Workspace Roles
- **Projects and Teams**
- Settings
- Runtimes/Legacy Engines
- Exercise Overview
- Essential Points
- Hands-On Exercise: Getting Started with CML

Projects Are Independent

- **Users can work freely without interfering with one another or breaking existing workloads**
- **Allows for collaboration with other users**
 - Teams can be created to access the project
 - Individual users can be added to private projects

The screenshot shows the Cloudera Machine Learning interface. At the top left is the 'CLOUDERA Machine Learning' logo. To its right is a navigation bar with the following items: 'All Projects' (with a back arrow), 'Overview' (highlighted in green), 'Sessions', 'Data', 'Experiments', 'Models', 'Jobs', 'Applications', 'Files', 'Collaborators', and 'Project Settings'. On the right side of the interface, the path 'bshimeldevuseast2_3 / ML_Project' is displayed. The main area is titled 'ML_Project' and includes a lock icon. Below it is the sub-header 'Sample project'. There are three sections: 'Models' (with a note: 'This project has no models yet. Create a [new model](#).'), 'Jobs' (with a note: 'This project has no jobs yet. Create a [new job](#) to document your analytics pipelines.'), and 'Files'. The 'Files' section lists the following items:

	Name
<input type="checkbox"/>	seaborn-data
<input type="checkbox"/>	analysis.ipynb
<input type="checkbox"/>	analysis.py
<input type="checkbox"/>	cdsw-build.sh

New Project

- Name
- Description
- Visibility
 - Private
 - Public
- Initial Setup
 - Blank
 - Template
 - AMPs
 - Local file
 - Git

* Project Name
Customer Churn

Project Description
This primary goal of this project is to build a logistic regression classification model to predict the probability that a group of customers will churn from a telecommunications company.

Project Visibility
 Private - Only added collaborators can view the project
 Public - All authenticated users can view this project.

Initial Setup
[Blank](#) [Template](#) [AMPs](#) [Local Files](#) [Git](#)

Templates include example code to help you get started.
[Python](#) ▾

New Project Runtime Setup

Runtime setup

[Basic](#) [Advanced](#)

Basic configuration adds the most commonly used Editors for the Kernel of your choice. To fine-tune the Editors available in the project, choose the Advanced tab.

Kernel

Python 3.7

Add GPU enabled Runtime variant

These runtimes will be added to the project:

- JupyterLab - Python 3.7 - Standard - 2022.04
- PBJ Workbench - Python 3.7 - Tech Preview - 2022.04
- Workbench - Python 3.7 - Standard - 2022.04

Project Visibility

- When you create a project in your personal context, specify one of the following visibility levels to the project
 - **Public**: grant read-level access to everyone with access to the Cloudera Machine Learning application
 - **Private**: you must explicitly add someone as a project collaborator to grant them access
- To work with colleagues on a project, add them to the project as a collaborator

Collaborators

This project is **private**. Only collaborators can view and edit this project. [Change Settings](#).

Add Collaborator

Search by name, username, or email...

Collaborator	Permission	Actions
bshimeldevuseast2_3	Owner	

Granting Admin or Contributor permission to other users may have security impact since it gives them full access to your project files and running sessions.

Collaborator Access Levels

Viewer

- Read-only access to team projects

Operator

- Read-only access to team projects
- Start and stop existing jobs in team projects they have access to

Contributor

- Write-level access to team projects

Admin

- Complete access to team projects and associated members

Team Accounts

Members	Type	Actions
C Char	Contributor	change delete
N Na	Contributor	change delete
W William	Contributor	change delete

- **Team projects are owned by the team**
 - Specify users who work together on more than one project
 - Provide streamlined administration
- **Team administrators can create and manage teams**
 - Modify account profile
 - Manage members
 - Add or remove team members
 - Manage each member's permissions (access level)
 - Modify Team SSH Keys

Chapter Topics

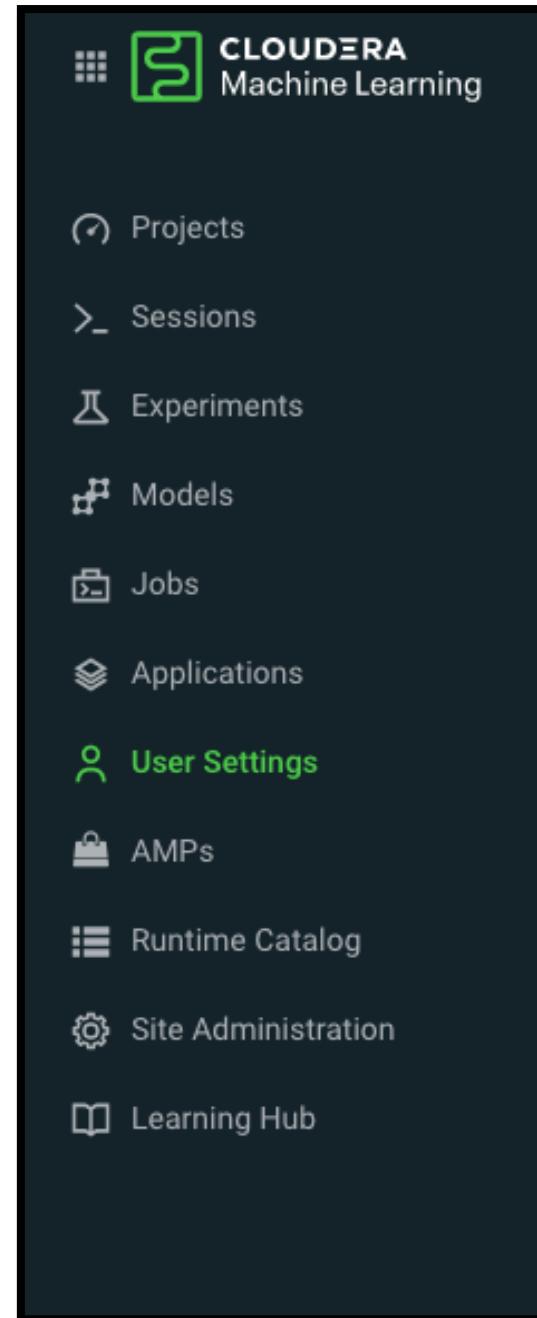
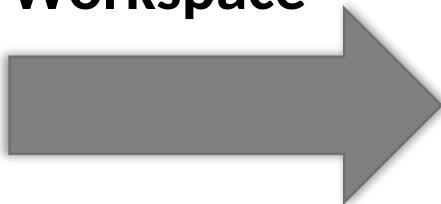
Introduction to CML

- Overview
- CML Versus CDSW
- ML Workspaces
- Workspace Roles
- Projects and Teams
- **Settings**
- Runtimes/Legacy Engines
- Exercise Overview
- Essential Points
- Hands-On Exercise: Getting Started with CML

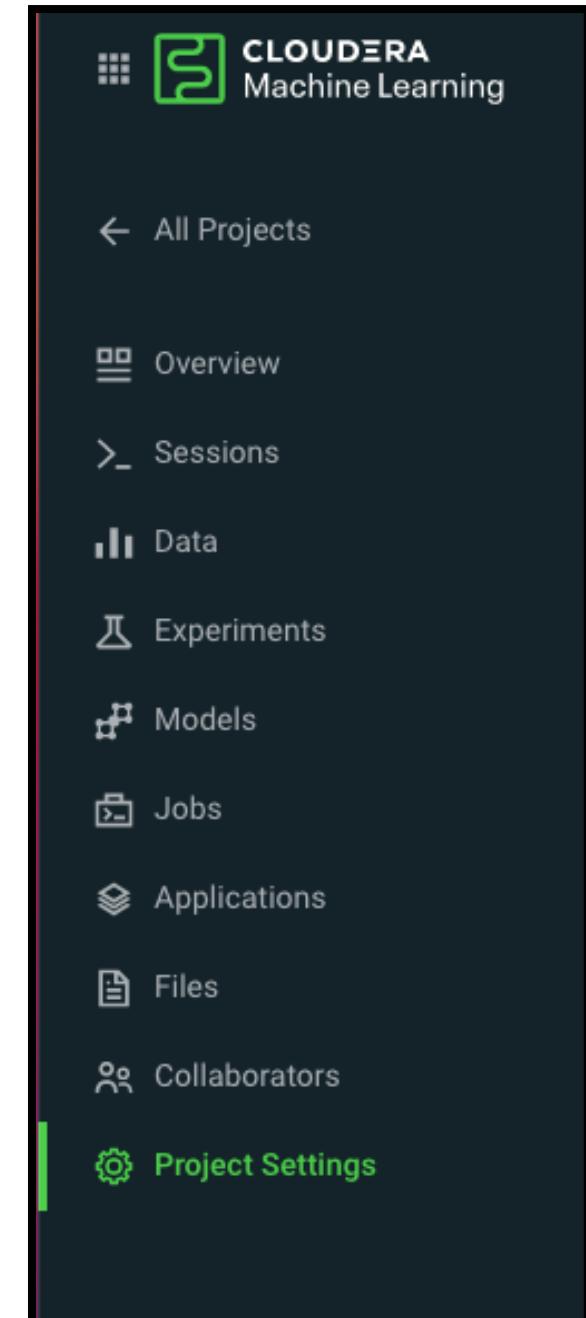
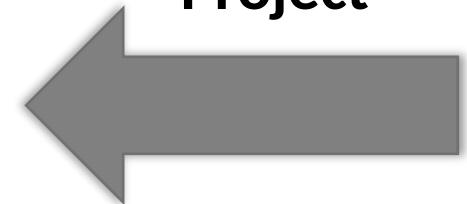
Settings

- Located in CML menu
- Are context sensitive

Workspace



Project



User Settings

■ Specific to the User within the Workspace

User Settings

Profile Teams Outbound SSH API Keys Remote Editing Environment Variables

Full Name
Bruce Shimel

Email
bshimel@cloudera.com

Bio
Bio

Update Account

User Settings - Tabs

Tab	Description
Profile	<ul style="list-style-type: none">• Modify the project name and privacy settings
Teams	<ul style="list-style-type: none">• Lists the teams
Outbound SSH	<ul style="list-style-type: none">• Displays user public SSH key• Reset SSH key
API Keys	<ul style="list-style-type: none">• Create API Keys
Remote Editing	<ul style="list-style-type: none">• Use SSH public keys to authenticate SSH sessions that you initiate with the CLI client
Environment Variables	<ul style="list-style-type: none">• Set your user's environment variables that can be accessed from your scripts

Project Settings in CML

Project Settings

Options Runtime/Engine Advanced SSH Tunnels Data Connections Prototype Delete Project

Default Engine: ML Runtime ⓘ Legacy Engine ⓘ

Available Runtimes

Sessions and other workloads in this Project can use one of the Runtime variants configured below.

Editor	Kernel	Edition	Version	Jobs / Apps / Models using Runtime	
JupyterLab	Python 3.9	Standard	2022.04	0 / 0 / 0	X
Workbench	Python 3.9	Standard	2022.04	0 / 1 / 0	

Displaying 1 - 2 of 2 < 1 > 25 / page ▾

Project Settings - Tabs

Tab	Description
Options	<ul style="list-style-type: none">• Modify the project name and privacy settings
Runtime/Engine	<ul style="list-style-type: none">• Select engine version and add third-party editors
Advanced	<ul style="list-style-type: none">• Set environment variables used for all engines in project• Specify additional shared memory available to running sessions
Tunnels	<ul style="list-style-type: none">• Connect to resources using SSH key
Delete Project	<ul style="list-style-type: none">• Irreversible action in which all files, data, sessions and jobs are removed• Only accessed by project admins

Team Settings

Team Settings

Profile Members Outbound SSH

Description

Team description

Update Account

Team Settings - Tabs

Tab	Description
Profile	<ul style="list-style-type: none">• Team description
Members	<ul style="list-style-type: none">• Manage team members and access level
Outbound SSH	<ul style="list-style-type: none">• Displays user public SSH key• Reset SSH key

Team Members

TestTeam / Team Settings / Members

Team Settings

Profile Members Outbound SSH

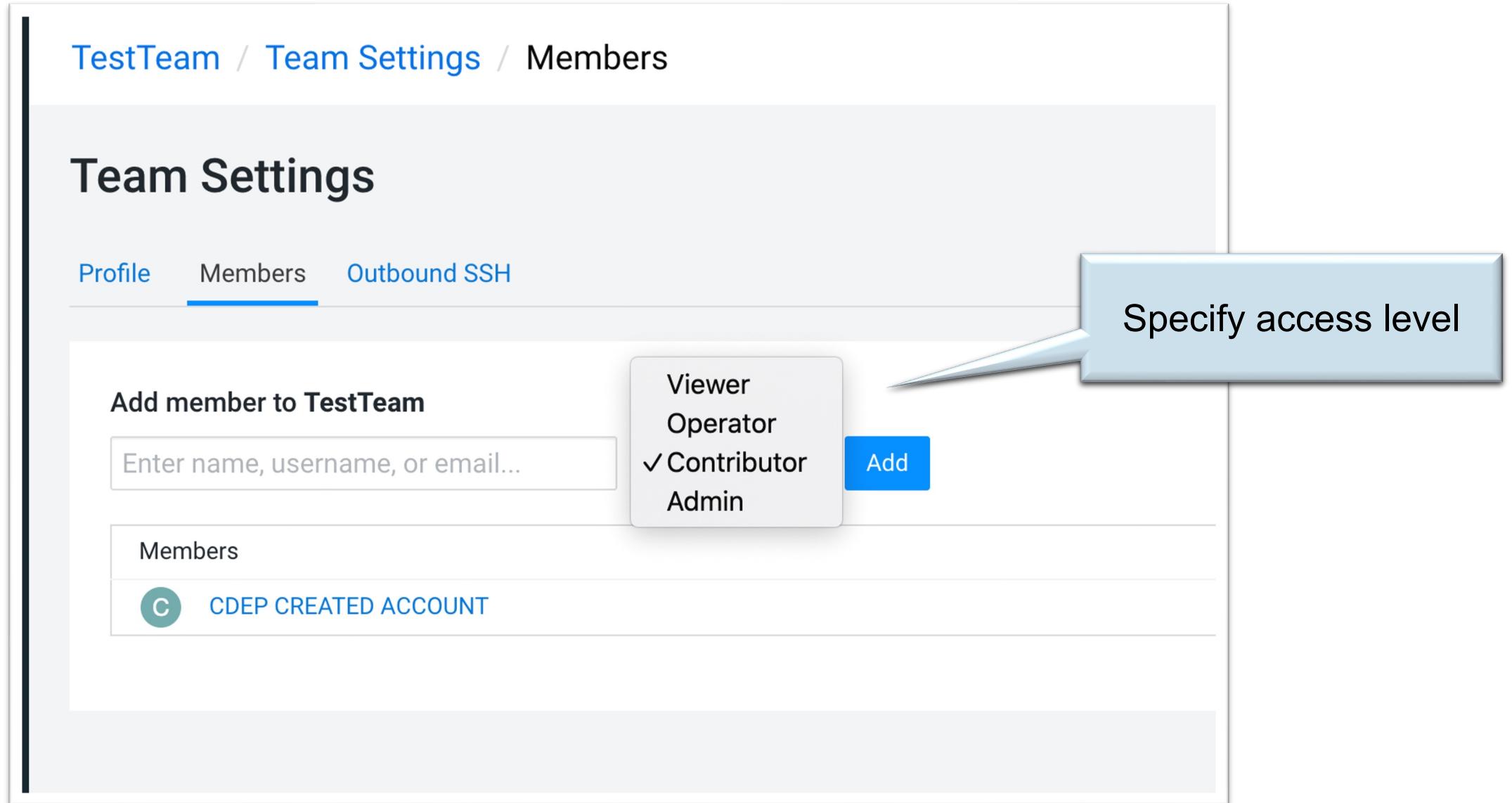
Add member to TestTeam

Enter name, username, or email...

Viewer
Operator
 Contributor
Admin

Add

Specify access level



The screenshot shows the 'Members' tab selected in the 'Team Settings' section of the 'TestTeam' interface. A tooltip labeled 'Specify access level' points to a dropdown menu where 'Contributor' is selected. Other options in the menu include 'Viewer', 'Operator', and 'Admin'. A button labeled 'Add' is visible next to the dropdown. Below the dropdown, there is a list of members starting with 'CDEP CREATED ACCOUNT'.

Site Administration

- Site administrators can manage the entire workspace
 - Monitor and manage all user activity across a workspace
 - Add new custom engines
 - Configure certain security settings

The screenshot shows the 'Site Administration / Overview' page. At the top, there is a navigation bar with tabs: Overview (underlined), Users, Teams, Usage, Quotas, Models, Runtime/Engine, Data Connections, Security, AMPs, Settings, and Support. To the right of the tabs is a search bar labeled 'Project quick find'. Below the navigation bar, the main title 'Site Administration' is displayed. Underneath the title, there is a section titled 'Cluster Monitoring' with the sub-instruction 'View cluster usage metrics and trends in custom built Grafana dashboards.' followed by a link 'Grafana Dashboard' with a blue arrow icon. Further down, there is a section titled 'Cluster Metrics Snapshot' containing a table with two rows. The first row has 'Release' in the first column and 'dev' in the second column. The second row has 'Domain' in the first column and 'ml-7a767539-390.bshimel.kfjr-x0dh.cloudera.site' in the second column.

Release	dev
Domain	ml-7a767539-390.bshimel.kfjr-x0dh.cloudera.site

Site Administration - Tabs

Tab	Description
Overview	<ul style="list-style-type: none">View cluster usage metrics and trends in custom built Grafana dashboards
Users	<ul style="list-style-type: none">Manage users, see which users are currently active, and when a user last logged on to CML
Teams	<ul style="list-style-type: none">Manage teams and team members
Usage	<ul style="list-style-type: none">Monitor activity on an ML workspace
Quotas	<ul style="list-style-type: none">Configure CPU, GPU, and memory quotas for users of an ML workspace
Models	<ul style="list-style-type: none">Monitor all active models currently deployed on your workspace.

Site Administration – Tabs (2)

Tab	Description
Runtime/Engine	<ul style="list-style-type: none">• Select engine version and add third-party editors
Data Connections	<ul style="list-style-type: none">• Manage data connection name, type, and availability in project
Security	<ul style="list-style-type: none">• Restrict or hide specific functionality that non-Site Administrator users have access to in the UI• Set web sessions timeout
AMPs	<ul style="list-style-type: none">• Manage AMP entries
Settings	<ul style="list-style-type: none">• General settings such as access control, project templates, SMTP server settings, and garbage collection
Support	<ul style="list-style-type: none">• Download diagnostic bundles for an ML workspace

Chapter Topics

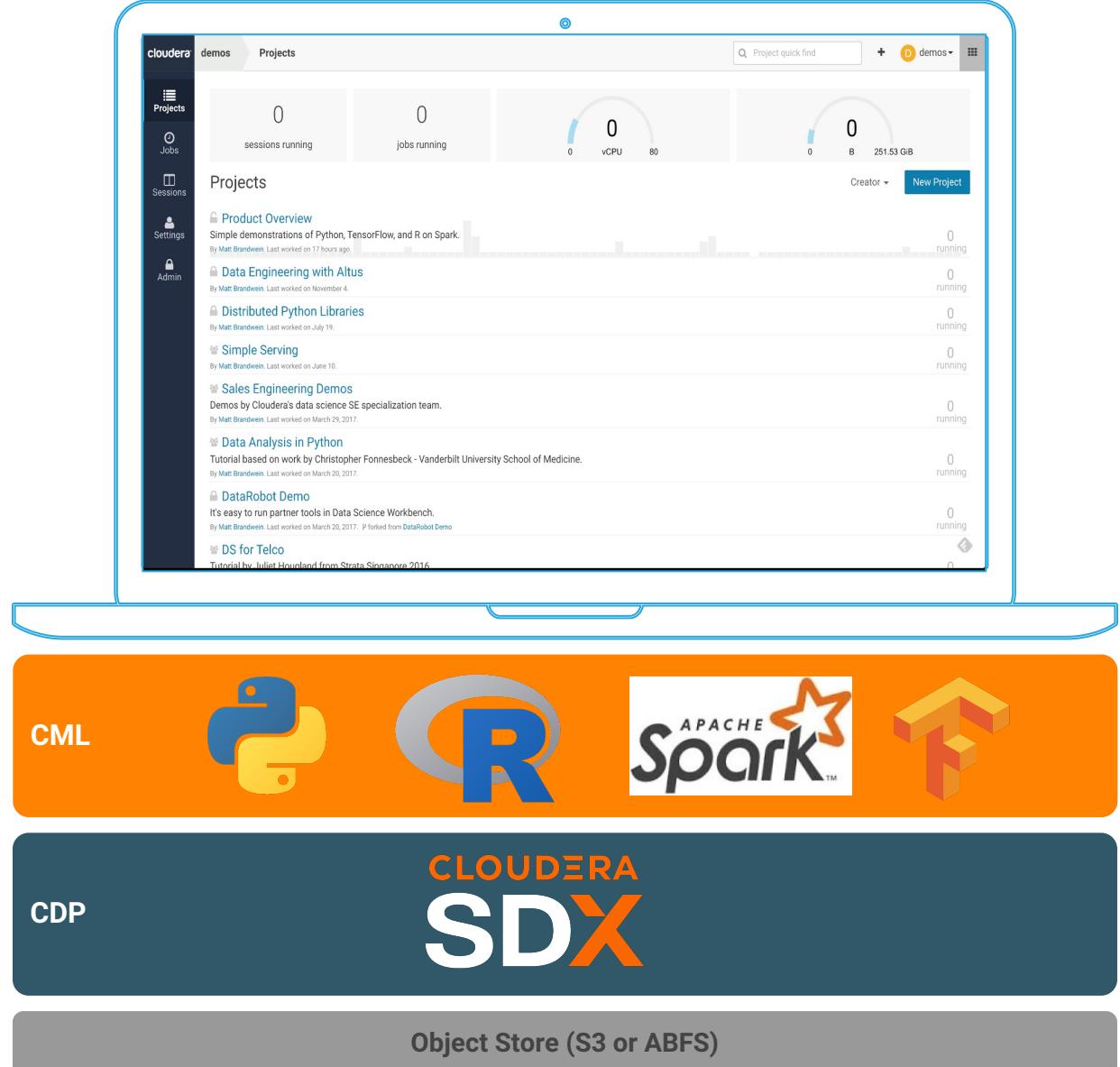
Introduction to CML

- Overview
- CML Versus CDSW
- ML Workspaces
- Workspace Roles
- Projects and Teams
- Settings
- **Runtimes/Legacy Engines**
- Exercise Overview
- Essential Points
- Hands-On Exercise: Getting Started with CML

ML Runtimes

- Responsible for running the code written by users and intermediating access to the data
- Keeps the images small and improves performance, maintenance, and security
- Similar to a virtual machine
 - Container images that contain the Linux OS, interpreter(s), and libraries
 - Customized to have all the necessary dependencies to access the computing cluster
 - Keeps each project's environment entirely isolated

Cloudera ML Runtimes



- **Isolated, containerized** working environment for our end users
- Core feature to enable self-service data science
 - Direct access to data
 - On-demand resources
 - Ability to install and use any library, framework without IT assistance.
- Out of the box support
 - Editors: Workbench, JupyterLab
 - Kernels: Python 3.7-3.9, R 3.6, 4.0-4.1, Scala 2.11
 - Editions: Standard, NVIDIA GPU

Runtime Environments Enable Flexibility

Start A New Session

Session Name
Exploration

Runtime

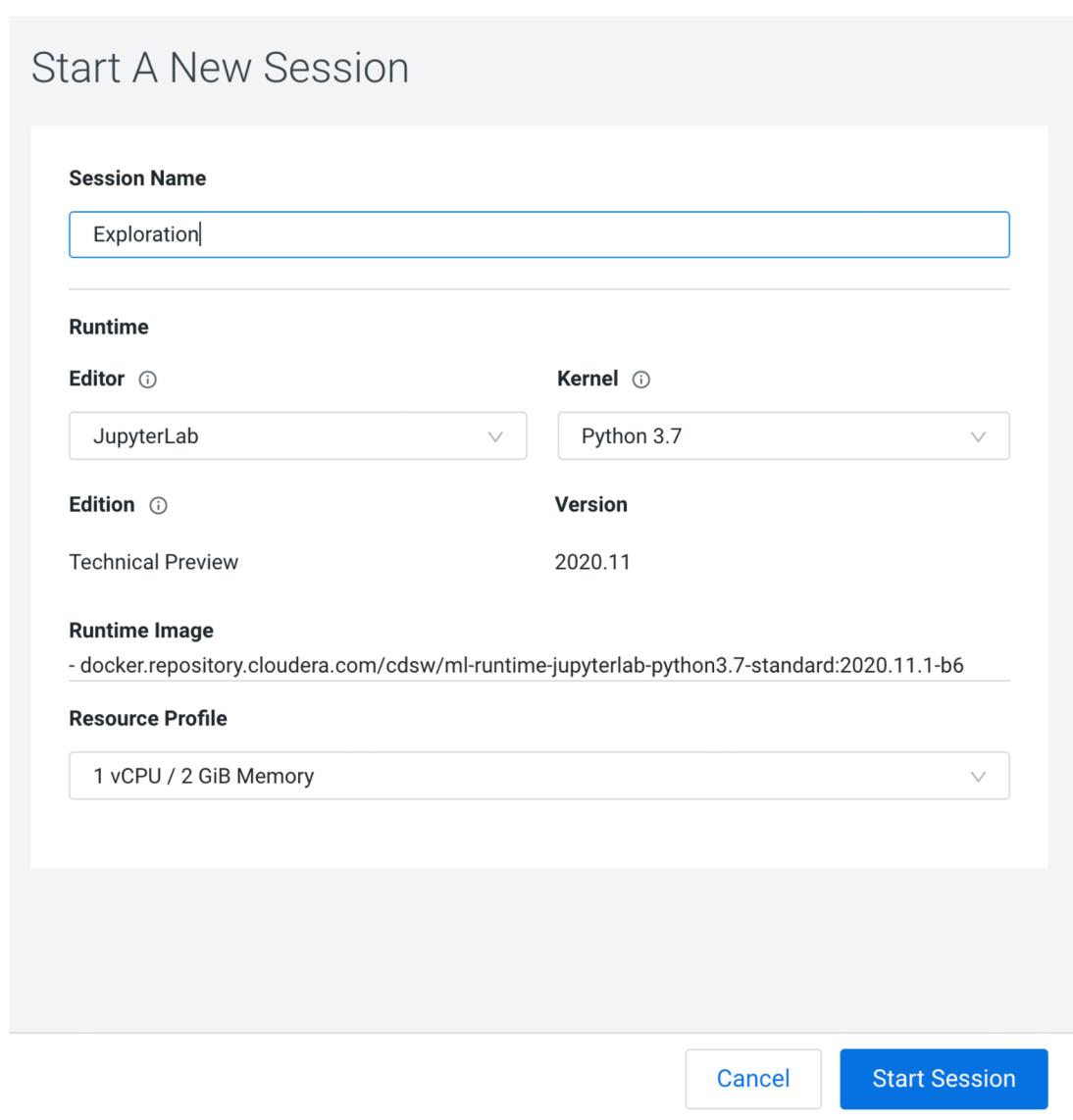
Editor (i) JupyterLab Kernel (i) Python 3.7

Edition (i) Technical Preview Version 2020.11

Runtime Image
- docker.repository.cloudera.com/cdsw/ml-runtime-jupyterlab-python3.7-standard:2020.11.1-b6

Resource Profile
1 vCPU / 2 GiB Memory

Cancel Start Session



Editor

Development interface to write and execute code

Examples: Workbench, JupyterLab

Kernel

Execution engine for the session of work

Examples: Python 3.7, Python 3.8, R3.6

Edition

Set of capabilities (tools/libraries) available for the run

Examples: Standard, RAPIDS

Version

Major version of the runtime

Example: 2022-04

ML Runtime Catalog

 CLOUDERA
Machine Learning

- Projects
- Sessions
- Experiments
- Models
- Jobs
- Applications
- User Settings
- AMPs
- Runtime Catalog**
- Site Administration
- Learning Hub

Runtime Catalog

Search Runtime Editor All Kernel All Edition All

Status	Type	Editor	Kernel	Edition
✓	CLOUDERA	Workbench	Cloudera Data Visualization	CDV 6.3.7
✓	CLOUDERA	Workbench	Cloudera Data Visualization	CDV 6.4.1
✓	CLOUDERA	JupyterLab	Python 3.7	Nvidia GPU
✓	CLOUDERA	JupyterLab	Python 3.7	Standard
✓	CLOUDERA	JupyterLab	Python 3.8	Nvidia GPU
✓	CLOUDERA	JupyterLab	Python 3.8	Standard
✓	CLOUDERA	JupyterLab	Python 3.9	Nvidia GPU
✓	CLOUDERA	JupyterLab	Python 3.9	Standard
✓	CLOUDERA	PBJ Workbench	Python 3.7	Tech Preview
✓	CLOUDERA	Workbench	Python 3.7	Nvidia GPU
✓	CLOUDERA	Workbench	Python 3.7	Standard
✓	CLOUDERA	Workbench	Python 3.8	Nvidia GPU

ML Runtimes Versus Legacy Engines

- **Runtimes and the Legacy Engine have the same purpose**
 - Container images that contain the Linux OS, interpreter(s), and libraries
 - The environment in which your code runs
- **ML Runtimes**
 - Small image
 - Contains a single interpreter and UNIX tools
 - Improved performance, maintenance, and security
 - Recommended to use for all new projects
- **Legacy Engine**
 - Huge image
 - Contains four Engine interpreters (Python 2, Python 3, R, Scala), and UNIX tools
 - Existing Engine-based projects can be migrated to ML Runtimes

Chapter Topics

Introduction to CML

- Overview
- CML Versus CDSW
- ML Workspaces
- Workspace Roles
- Projects and Teams
- Settings
- Runtimes/Legacy Engines
- **Exercise Overview**
- Essential Points
- Hands-On Exercise: Getting Started with CML

Scenario Explanation

- **Le DuoCar is a fictitious French ridesharing company**
- **DuoCar is its American subsidiary**
 - The company expanded to the United States because the regulatory environment and the taxi lobby in France limit growth there
 - DuoCar is a new entrant in the market dominated by Uber and Lyft
- **Fargo, North Dakota is the first American city where DuoCar is operating**
 - Driver and rider sign-ups began on January 1, 2017
 - Rides began on February 1, 2017
 - The company plans to expand to other American cities
- **You have been hired as a data scientist at DuoCar**

duocar

How DuoCar Works

- **DuoCar works basically the same way as Uber and Lyft**
 - Drivers are independent contractors
 - They set their own schedules
 - They drive their own vehicles
 - Riders use a mobile app to request rides
- **DuoCar offers four levels of service based on vehicle attributes**

Service	Vehicle type	Year	Color
DuoCar	Sedan or larger	≥ 2002	Any
DuoGrand	Large vehicle with room for six riders	≥ 2002	Any
DuoNoir	Luxurious sedan or larger	≥ 2012	Black
DuoElite	Luxurious large SUV with room for six riders	≥ 2012	Black

DuoCar Datasets

- **DuoCar generates and stores several datasets:**

- Drivers
- Riders
- Rides
- Ride routes
- Ride reviews

- **DuoCar also uses some external datasets:**

- Weather
- Demographics

- **Your job is to use these datasets:**

- To discover and communicate insights
- To help develop data products

DuoCar Business Goals

- **DuoCar wants to better understand patterns of driver and rider behavior**
 - What factors affect driver and rider sign-ups?
 - How do different types of riders use DuoCar?
 - What are the reasons for rider dissatisfaction?
- **DuoCar wants to develop its general analytic capability**
 - To use data to answer questions accurately as they arise
 - To predict driver and rider behavior
 - To develop methods that scale well as data sizes increase
 - To build a culture of data-driven decision making

DuoCar Machine Learning Platform

- **DuoCar's data science team uses a mix of Python and R**
 - In the past, they have typically worked on laptops using data extracts
 - They use Jupyter Notebooks and RStudio
 - They use many packages from PyPI and CRAN
- **DuoCar's leadership has recognized the need for a more modern data environment**
 - That enables use of larger datasets, and scales as the data grows
 - That lets the team use their preferred tools and packages—without security risks or IT burden
 - That facilitates collaboration and sharing
- **The company deployed Cloudera Data Platform (CDP) Public Cloud**
 - A permanent cluster running on Amazon EC2 instances
- **The company recently installed Cloudera Machine Learning (CML)**
 - To provide data scientists with secure self-service access to the cluster

DuoCar CDP Cluster

- **Data is ingested from DuoCar's operational systems and stored in the cluster for analytic use**
 - Most data is stored in S3
- **The cluster has three main tools for processing analytic workloads**
 - Apache Spark, for a wide range of data processing jobs
 - Apache Hive, for batch SQL queries
 - Apache Impala, for interactive SQL queries
- **The cluster has two web-based workbench tools**
 - Hue, for interfacing with S3, Hive, and Impala
 - CML, for interfacing with Spark through Python and R code

Chapter Topics

Introduction to CML

- Overview
- CML Versus CDSW
- ML Workspaces
- Workspace Roles
- Projects and Teams
- Settings
- Runtimes/Legacy Engines
- Exercise Overview
- **Essential Points**
- Hands-On Exercise: Getting Started with CML

Essential Points

- **Cloudera Machine Learning (CML) is part of the Cloudera Data Platform (CDP)**
 - Analyze data
 - Train models
 - Deploy APIs
- **An ML Project**
 - Contains all the code, configuration, and libraries needed to reproducibly run analyses
 - Is independent, ensuring users can work freely without interfering with one another or breaking existing workloads
 - Allows for collaboration with other users
- **The Settings link in the left menu is context-sensitive and provides configuration for a user, a project, or a team**
- **ML Runtimes are responsible for running the code and intermediating access to the data**

Chapter Topics

Introduction to CML

- Overview
- CML Versus CDSW
- ML Workspaces
- Workspace Roles
- Projects and Teams
- Settings
- Runtimes/Legacy Engines
- Exercise Overview
- Essential Points
- **Hands-On Exercise: Getting Started with CML**

Hands-On Exercise: Getting Started with CML

- **In this exercise, you will**
 - Login to the Cloudera Data Platform exercise environment,
 - View the Cloudera Machine Learning workspace
 - Create a new CML project
 - Create a new session
 - Delete a CML project
- **Please refer to the Hands-On Exercise Manual for instructions**

Introduction to AMPs and the Workbench

Introduction to AMPs and the Workbench

By the end of this chapter, you will be able to

- Run code from within a session
- Use Git for version control
- Create ML web applications/dashboards and easily share them with other business stakeholders
- Install and run AMPs that provide reference example machine learning projects in CML

Chapter Topics

Introduction to AMPs and the Workbench

- Editors and IDE
- Git
- Embedded Web Applications
- AMPs
- Essential Points
- Hands-On Exercise: (AMP) Streamlit on CML

Native Workbench Console and Editor

- Interactive environment tailored specifically for data science
- UI includes
 - An editor where you can edit your scripts.
 - A console where you can track the results of your analysis
 - A command prompt where you can enter commands interactively
 - A terminal where you can use a shell

Working in Workbench

1. Add code to your project

- Create and edit your code
- Upload existing code files
- Specify what compute resources you need

2. Launch a session using Python, R, or Scala

3. Run code

- From your code files
- At the session prompt
- Or run commands in the terminal

4. Stop a session

- Or have it timeout

New Session

- **Editor**
 - Selects the Editor, for example JupyterLab, Workbench
- **Kernel**
 - Selects the Kernel, for example Python 3.7, R4.0
- **Edition**
 - Selects the Runtime Edition
 - Initially only Standard variants are supported
- **Version**
 - Selects the ML Runtimes version

Start A New Session

Session Name
Test Session

Runtime

Editor	Kernel	Edition	Version
JupyterLab	Python 3.9	Standard	2022.04

Configure additional runtime options in [Project Settings](#).

Enable Spark [Spark 3.2.0 - CDE 1.15 - HOTFIX-2](#)

Runtime Image
- docker.repository.cloudera.com/cloudera/cdsw/ml-runtime-jupyterlab-python3.9-standard:2022.04.1-b6

Resource Profile
1 vCPU / 2 GiB Memory

[Cancel](#) [Start Session](#)

- **To run code from the editor**
 - Select a script from the project files on the left sidebar
 - Click the arrow to run the entire script or
Highlight the code you want to run and Ctrl+Enter (Windows/Linux) or cmd+Enter (macOS)
- **Code autocomplete**
 - Python and R kernels include support for automatic code completion
 - Single tab to display suggestions
 - Double tab for autocomplete
- **Project code files**
 - All project files are stored to persistent storage at
`/var/lib/cdsw/current/projects/<project name>`

Third-Party Editors

- CML can be configured to work with third-party editors
 - Browser-based IDEs such as Jupyter
 - Local IDEs that run on your machine
- The configuration for an IDE depends on which type of editor you want to use
- You can only edit and run code interactively with the IDEs
- Tasks such as creating a project or deploying a model require the
 - CML web UI
 - API v2

Session Output

Install Data  Success

By Bruce Shimel — Session — 1 vCPU / 2 GiB Memory — 2 days ago

Session Logs   

Getting Started

This is your **session**. Your **editor** is on the left and your **input prompt** is on the bottom.

To execute code from the editor, select the code and execute it with **Command-Enter** on Mac or **Ctrl-Enter** on Windows. You can also enter code at the prompt below.

Use **?command** to get help on a particular command.

```
> !chmod 755 cml/setup-data.sh
> !cml/setup-data.sh

S3_ROOT = s3a://cdp-storage-bshimel-456-class-2283/datalake-bshimel-456-class-2283
HIVE_EXT = s3a://cdp-storage-bshimel-456-class-2283/datalake-bshimel-456-class-2283/warehouse/tablespace/external/hive
PRINCIPAL = adm_bshimel_2283
DATALAKE = datalake-bshimel-456-class-2283
Aug 09, 2022 8:01:46 AM org.apache.knox.gateway.shell.KnoxSession createClient
INFO - Using default IAAS configuration
```

"!" executes commands in the shell, outside of the Python interpreter

Session Logs

Install Data  Success

By Bruce Shimel – Session – 1 vCPU / 2 GiB Memory – 2 days ago

Session Logs   

Name	Status
engine	unknown
spark executor 1	failed
spark executor 2	failed
spark executor 3	failed

Completed:

2022-08-09 08:01:39.257 1	INFO	Engine.Init.Root	r8p752wod8bl1hqy	Initial engine startup as UID data = {"uid":8536,"user":"cdsw"}
2022-08-09 08:01:39.257 1	INFO	EngineInit.LiveLog	r8p752wod8bl1hqy	Start creating LiveLog client data = {"user":"cdsw"}
2022-08-09 08:01:39.263 1	INFO	EngineInit.LiveLog	r8p752wod8bl1hqy	Finish creating LiveLog client data = {"user":"cdsw"}
2022-08-09 08:01:39.263 1	INFO	Engine.Init.Root	r8p752wod8bl1hqy	Not chowning /cdn and /output because we're not root data = {"u
2022-08-09 08:01:39.264 1	INFO	EngineInit.AddonSetup	r8p752wod8bl1hqy	Start setting HadoopCLI runtime addon on the engine data = {"user"
2022-08-09 08:01:39.273 1	INFO	EngineInit.Utils	r8p752wod8bl1hqy	Created symlink data = {"dest":"/usr/lib/hadoop","src":"/runtime-a
2022-08-09 08:01:39.276 1	INFO	EngineInit.Utils	r8p752wod8bl1hqy	Created symlink data = {"dest":"/usr/lib/hadoop-mapreduce","src":'
2022-08-09 08:01:39.279 1	INFO	EngineInit.Utils	r8p752wod8bl1hqy	Created symlink data = {"dest":"/usr/lib/hadoop-hdfs","src":"/runi
2022-08-09 08:01:39.284 1	INFO	EngineInit.Utils	r8p752wod8bl1hqy	Created symlink data = {"dest":"/usr/lib/jvm/java-8-openjdk-amd64"
2022-08-09 08:01:39.286 1	INFO	EngineInit.Utils	r8p752wod8bl1hqy	Created symlink data = {"dest":"/usr/lib/libhdfs.so","src":"/runi
2022-08-09 08:01:39.292 1	INFO	EngineInit.Utils	r8p752wod8bl1hqy	Created symlink data = {"dest":"/etc/java-8-openjdk","src":"/runi
2022-08-09 08:01:39.292 1	INFO	EngineInit.AddonSetup	r8p752wod8bl1hqy	Setting environment variables on the path data = {"envs":{ "CDH_M
2022-08-09 08:01:39.292 1	INFO	EngineInit.AddonSetup	r8p752wod8bl1hqy	EI_

Session List

adm_bshimel_2283 / Test_01 / Sessions

Project quick find + B adm_bshimel_2283 ▾

Creator All Show Running Only

Stop Selected Delete Selected

<input type="checkbox"/>	Status	Session	Kernel	Creator	Created At	Duration	
<input type="checkbox"/>	Timeout	Test	(Python 3.9 Workbench Standard)	Bruce Shimel	08/09/2022 4:12 AM	1h 8m 16s	<input type="button" value="Delete"/>
<input type="checkbox"/>	Success	Install Data	(Python 3.9 Workbench Standard)	Bruce Shimel	08/09/2022 4:01 AM	6m 11s	<input type="button" value="Delete"/>
<input type="checkbox"/>	Success	Install Dependencies	(Python 3.9 Workbench Standard)	Bruce Shimel	08/09/2022 3:58 AM	2m 48s	<input type="button" value="Delete"/>

Displaying 1 - 3 < 1 > 25 / page

Chapter Topics

Introduction to AMPs and the Workbench

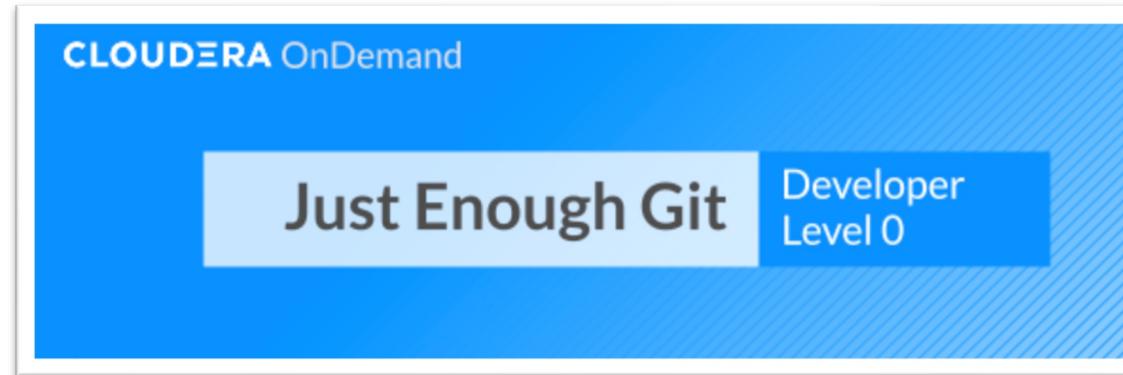
- Editors and IDE
- **Git**
- Embedded Web Applications
- AMPs
- Essential Points
- Hands-On Exercise: (AMP) Streamlit on CML

Using Git with CML

- **CML is designed to be used with Git**
 - Create a project by cloning a Git repository
 - Perform Git operations to sync a project with a remote repository
- **CML provides full access to Git on the command line**
 - It does not provide a graphical user interface for Git
- **CML can be used with repositories on any Git server or service, including:**
 - Cloud-based Git hosting services such as GitHub, GitLab, Bitbucket Private Git servers (if accessible)
 - Other version control systems with Git interfaces



Learning Git



- **To prepare for using Git in CML, consider taking the course *Just Enough Git* on Cloudera OnDemand:**
 - Teaches fundamentals of using Git on the command line
 - Includes video demonstrations and hands-on exercises

Authorizing CML to Access Git

- In CML, you can create a project by cloning a Git repository
 - For read-only access to public repos, no extra setup is required
 - For write access or private repo access, authorization is required
 - Add the public SSH key from your CML account to your Git provider account
 - This is a self-service task in the CML user interface

Adding SSH Key to GitHub

User Settings

Profile Outbound SSH API Keys Remote Editing Environment Variables

User Public SSH Key

```
ssh-rsa AAAAB3NzaC1yc2EAAAQABAAQDVIsKNgMCObS2ZcLN2rv5qsoRN2HcnKtiy2/NYWrTWaSCIPr4BKxE7A0qMrhiXfq0auqQVD8xrWuBNemD60++FfuZ/inw/A1LjzH8okxjQqdmF/1T7gicY5K0v3Xek4Sg31T/ws1HUn5vJjc7bsKZLUwNiaVKoHcgFrE9MtZFRQYL03CHbBYipZ04qXzM1ZDK+jCi3RQjGv51J5DIV0ugcX1IkNANV8Lqx8rgaha6qN4hLhBW7mH/8SU05UWr1xSV7/DeurcUokTrJx12pJ5EXWvKyXb9TC4B/I0eaZI7Fe3lJmd0D30Z9VuhXJbQpLEAFvJz6JrY7ynXEluMniSx cdsw
```

Reset SSH key

1. Sign in to CML
2. Go to User Settings
3. Go to the Outbound SSH tab and copy your public SSH key
4. Sign in to your GitHub account and add the CML key copied in the previous step to your GitHub account
5. For instructions, refer the GitHub documentation on [adding SSH keys to GitHub](#)

Cloning a Git Repository in CML

- In CML, you can create a new project by cloning a Git repository
- CML accepts two types of Git URLs

HTTPS URLs

- Begin with https://
- Use for read-only access to public repos
- No extra setup required before using

SSH URLs

- Begin with username@hostname:
- Use for write access or private repo access
- Before using, authorize your CML account to access the Git repo

Linking an Existing Project to a Git Remote

- If you did not create your project from a Git repository, you can link an existing project to a Git remote

```
$ git init  
$ git add *  
$ git commit -a -m 'Initial commit'  
$ git remote add origin git@github.com:username/repo.git
```

- Run **git status** after **git init** to make sure your **.gitignore** includes a folder for libraries and other non-code artifacts.

Contributing to a Git Repository from CML

- You can keep a CML project and a remote Git repo in sync
 - By pulling and pushing commits
- To push commits to a repo from CML
 - Ensure you have write access to the repo
 - Use the SSH URL to clone the repo into CML

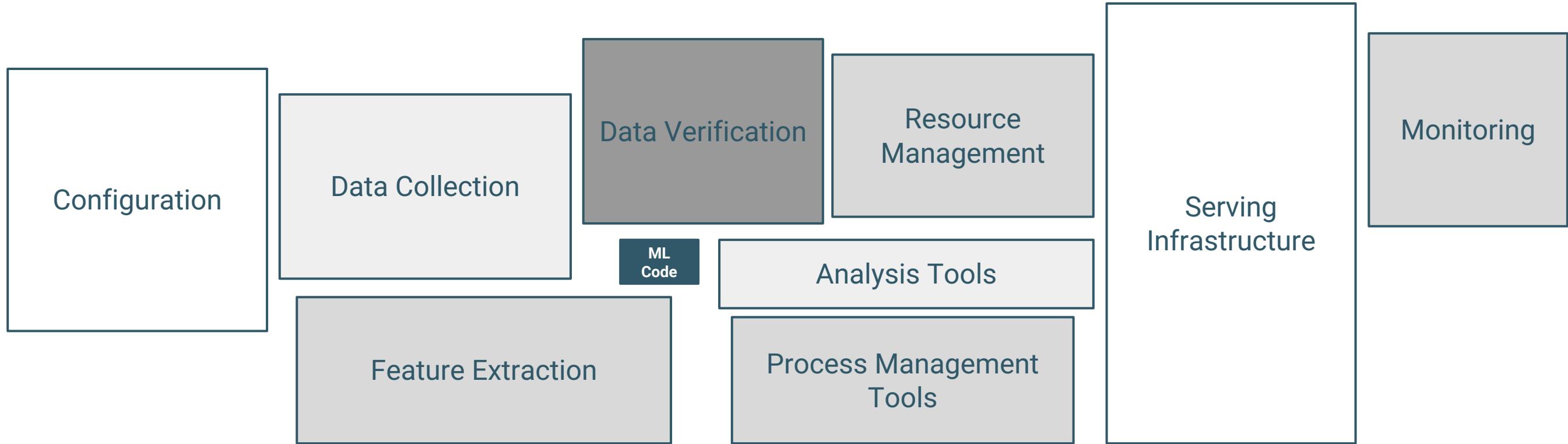
Chapter Topics

Introduction to AMPs and the Workbench

- Editors and IDE
- Git
- **Embedded Web Applications**
- AMPs
- Essential Points
- Hands-On Exercise: (AMP) Streamlit on CML

Hidden Technical Debt of ML Systems

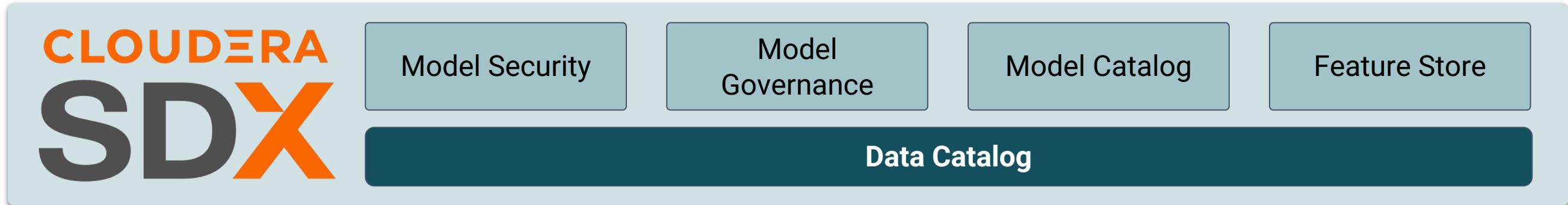
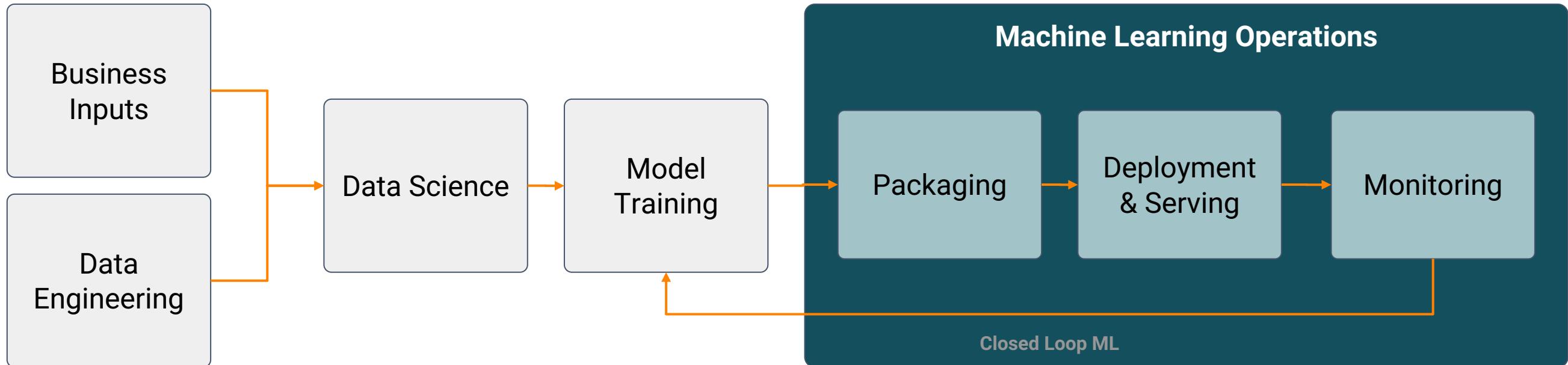
Google Paper



Source: <https://papers.nips.cc/paper/5656-hidden-technical-debt-in-machine-learning-systems.pdf>

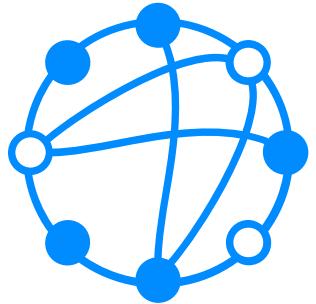
Only a small fraction of real-world ML systems is composed of the ML code, as shown by the small black box in the middle.
The required surrounding infrastructure is vast and complex.

ML Operations Closes the Loop

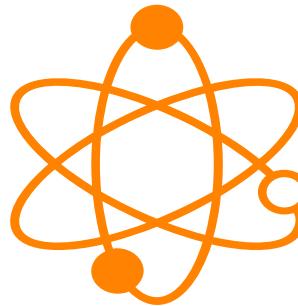


AI Applications

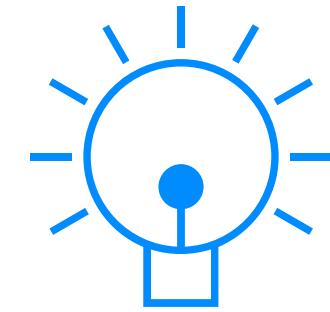
- Power more data science and AI use cases faster across the enterprise



Build predictive data applications: exploration, model development, training, and deployment, hosted Applications



Tools for the end-to-end workflow: SQL, Visuals, Python/R Code, Experiments, Models and Applications



Goals: Insights & Data Products
At the hand of the **Business**

Applications Defined

- **Applications are long-running web applications that make calls to deployed models that**
 - Give data scientists a way to create ML web applications/dashboards and easily share them with other business stakeholders
 - Can range from single visualizations embedded in reports, to rich dashboard solutions such as Tableau
 - Can be interactive or non-interactive
 - Stand alongside other existing forms of workloads in CML (sessions, jobs, experiments, models)
 - Must be created within the scope of a project like all other workloads
 - Are launched within their own isolated runtime
- **Additionally, like models, engines launched for applications do not time out automatically**
 - Will run as long as the web application needs to be accessible by any users
 - Must be stopped manually when needed

Test Your Application

- Before you deploy an application, make sure your application has been thoroughly tested
 - Use sessions to develop, test, and debug your applications
 - Test web apps by embedding them in sessions as described here:

<https://docs.cloudera.com/machine-learning/cloud/projects/topics/ml-embedded-web-apps.html>

Create Your Application

Field	Description
Name	<i>Unique name for the application</i>
Subdomain	Subdomain that will be used to construct the URL for the web application. For example, if you use test-app as the subdomain, the application will be accessible at test-app.<ml-workspace-domain-name>.
Description	Description for the application
Script	Script that hosts a web application on either CDSW_READONLY_PORT or CDSW_APP_PORT. Applications running on either of these ports are available to any users with at least read access to the project. The Python template project includes an entry.py script to test
Engine Kernel and Resource Profile	Kernel and computing resources needed for this application
Set Environment Variables	Specify the name and value for application variables

Application-level environment variables override the project-level environment variables if there is a conflict

Secure Your Application

- You can provide access to Applications via either the **CDSW_APP_PORT** or the **CDSW_READONLY_PORT**
- Any user with read or higher permissions to the project can access an application served through either port
 - CML applications are accessible by any user with read-only or higher permissions to the project. The creator of the application is responsible for managing the level of permissions the application users have on the project through the application
 - CML does not actively prevent you from running an application that allows a read-only user (i.e. Viewers) to modify files belonging to the project
 - By default, authentication for applications is enforced on all ports and users cannot create public applications. If desired, the Admin user can allow users to create public applications that can be accessed by unauthenticated users (see next slide)
 - For Transparent Authentication, CML can pass user authentication to an Application, if the Application expects an authorized request. The REMOTE-USER field is used for this task

Public Applications

- **To allow users to create public applications on an ML workspace:**
 1. As an Admin user, turn on the feature flag in Admin > Security by selecting Allow applications to be configured with unauthenticated access
 2. When creating a new application, select Enable Unauthenticated Access
 3. For an existing application, in Settings select Enable Unauthenticated Access
- **To prevent all users from creating public applications**
 - Go to Admin > Security and deselect Allow applications to be configured with unauthenticated access
 - After one minute, all existing public applications stop being publicly accessible.

Chapter Topics

Introduction to AMPs and the Workbench

- Editors and IDE
- Git
- Embedded Web Applications
- **AMPs**
- Essential Points
- Hands-On Exercise: (AMP) Streamlit on CML

Powering ML Use Cases at Scale is Hard

Enterprises need to overcome the barriers in development and production

20%

Of ML models in the enterprise
making it into production
Environments –*From 3073
C-Level executives surveyed.*



ginablaber @ginablaber

The story of enterprise Machine Learning: "It took me 3 weeks to develop the model. It's been >11 months, and it's still not deployed."
@DineshNirmalIBM #StrataData #strataconf

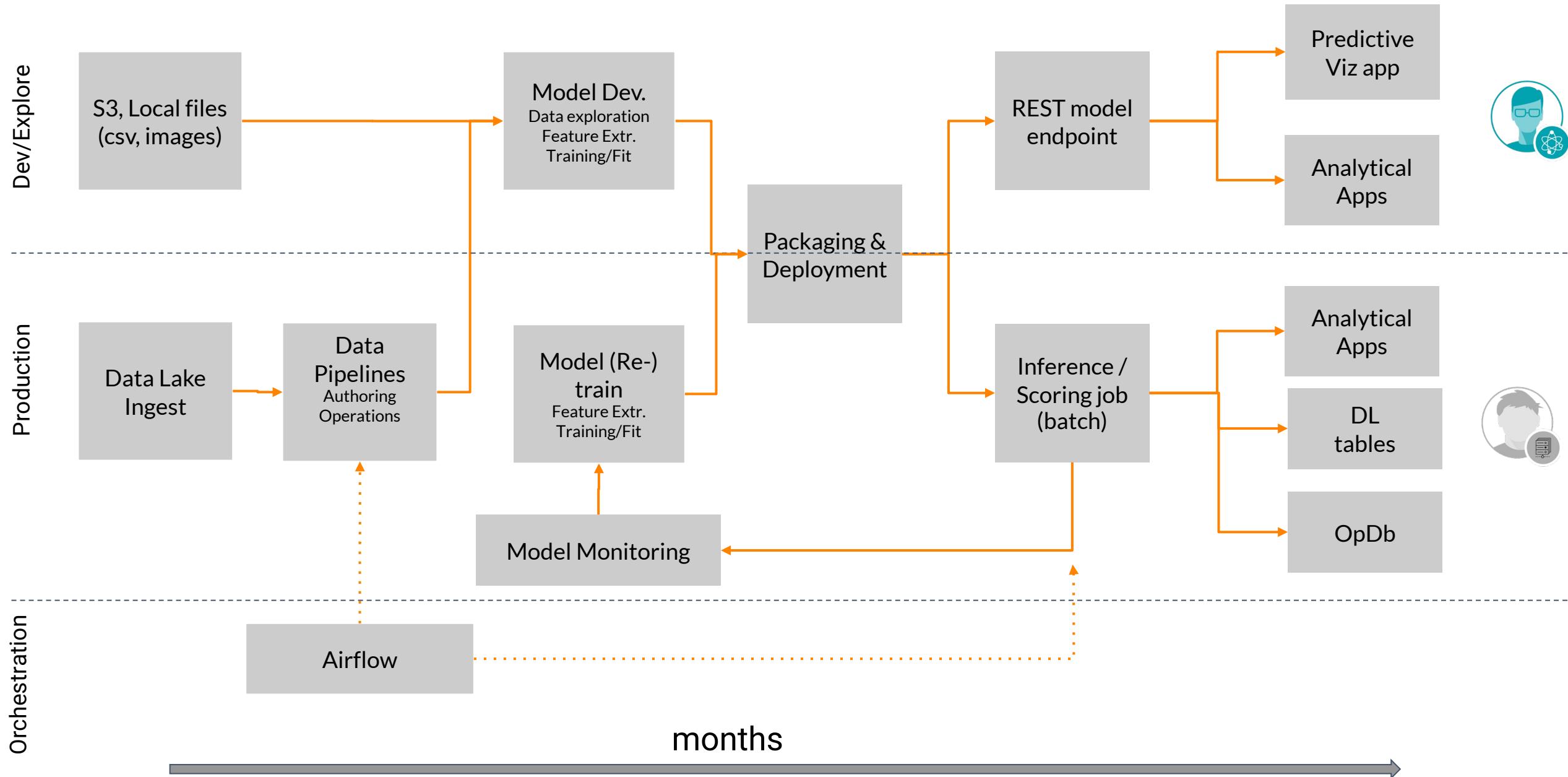


7

22



Building ML Applications - Timeline



Building ML Use Cases Faster



AMPs Defined

- AMPs stands for Applied Machine learning Prototypes
- AMPs are Cloudera Machine Learning projects and applications packaged for easy delivery
- AMPs accelerate machine learning projects and kickstart AI use cases by providing example workflows and applications that leverage the power of CML
 - Make it easy for novice CML users to be successful on our platform
 - Decrease time to value by providing high quality reference examples
 - Leverage Fast Forward research to showcase cutting edge ML techniques

Dozens of Use Cases Ready to Plug and Play

	Phone	Device	Device	Phone	Device	TV	Smart	Smart	Smart	Phone	Phone	TV	Churn
3870	Yes	Fiber optic	No	No	No	No	No	No	No	Mobile check	Electron. check	Yes	\$88.10
6380	Yes	Fiber optic	Yes	No	No	No	No	No	No	Mobile check	Electron. check	Yes	\$88.00
2909	Yes	Fiber optic	Yes	No	No	No	No	No	No	Mobile check	Electron. check	Yes	\$74.25
131	Yes	Fiber optic	Yes	Yes	No	No	No	No	No	Mobile check	Mobile check	Yes	\$88.00
5814	Yes	Fiber optic	No	Yes	No	No	No	No	No	Mobile check	Electron. check	Yes	\$88.25
2197	Yes	Fiber optic	No	No	No	No	No	No	No	Mobile check	Mobile check	Yes	\$70.10
3131	Yes	Fiber optic	No	No	No	No	No	No	No	Mobile check	Electron. check	Yes	\$7144.50
813	Yes	Fiber optic	Yes	No	No	No	No	No	No	Mobile check	Mobile check	Yes	\$84.00
3055	Yes	Fiber optic	Yes	No	No	No	No	No	No	Mobile check	Electron. check	Yes	\$88.00
2080	Yes	Fiber optic	Yes	No	No	No	No	No	No	Mobile check	Electron. check	Yes	\$88.25
4047	Yes	Fiber optic	Yes	No	No	No	No	No	No	Mobile check	Electron. check	Yes	\$70.00
2988	Yes	Fiber optic	No	No	No	No	No	No	No	Mobile check	Electron. check	Yes	\$2180.00

Churn Modeling with XGBoost

EXPLAINABILITY XGBOOST

Select Dataset: INCEPTIONV3
Select Model: INCEPTIONV3
Select Layer: LAYER 310
Distance Metric: COSINE

Visualization of Embeddings (UMAP) for Extracted Features
Top 15 results based on your search configuration: MODEL: INCEPTION (LAYER: 310), DISTANCE METRIC: COSINE

Search result score: 78.3%
What is this? 1.00 0.99 0.98 0.97 0.96 0.95 0.94 0.93 0.92 0.91 0.90 0.89 0.88 0.87 0.86 0.85 0.84 0.83 0.82 0.81 0.80



Deep Learning for Image Analysis

COMPUTER VISION IMAGES



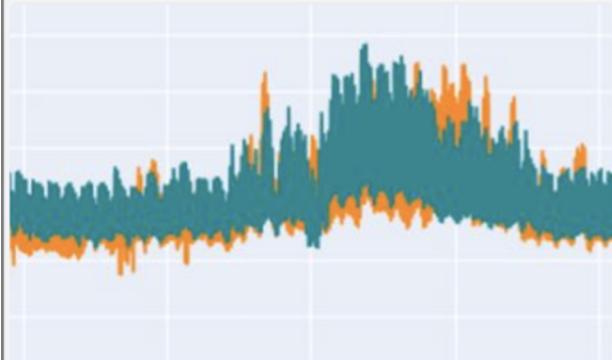
Neural Question Answering

NLP BERT



Airline Delay Prediction

XGBOOST



Structural Time Series

TIME SERIES

Selection on Network Intrusion Data

The `pk_in intrusion` dataset is a dataset of TCP connections that have been labeled as normal or representative of network attacks. Each TCP connection is represented as a set of attributes on domain knowledge pertaining to each connection such as the number of failed logins, connection duration, data bytes from source to destination, etc. The table below is a random sample of rows.

label	id	Server Count	Server Error Rate	Server S Error Rate	R Error Rate	Server R Error Rate	Same Server Error Rate	Different Server Error Rate	Server Different Host Rate	External Host Count
1	0	1.001%	0	0	0	0	1	0	0	1
1	1	1.001%	0	0	0	0	1	0	0	1
2	2	1.001%	0	0	0	0	1	0	0	1
3	3	0.001%	0	0	2	0.5	0.22	1	1	2
4	4	0.001%	0	0	0	0	1	0	0	0.0117%
5	5	0.001%	0	0	1	0.5	0.22	1	1	1
6	6	0.001%	1	1	0	0	0.01	0.09	0	1
7	7	0.025%	0	0	0	0	1	0	0.55	1
8	8	0.057%	0	0	1	0.47	0.5	1	1	1

Deep Learning for Anomaly Detection

ANOMALY DETECTION DEEP NEURAL NETS



Fraud Detection

FRAUD DETECTION ANOMALY DETECTION

Model Result Output
The Test Sentence will be sent to the selected model and the response displayed below

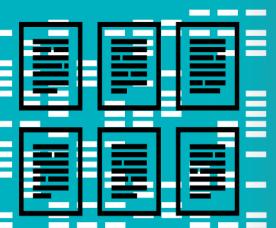
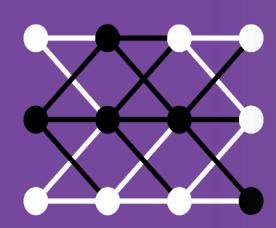
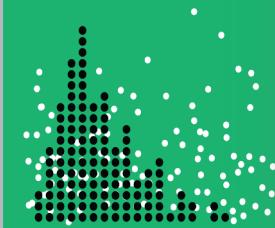
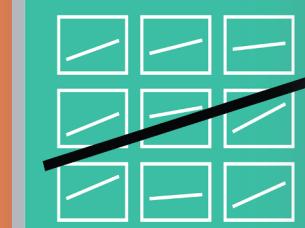
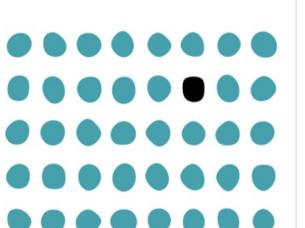
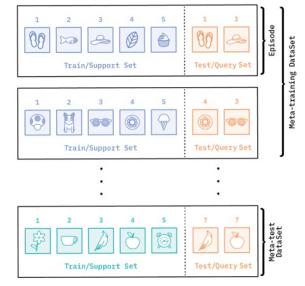
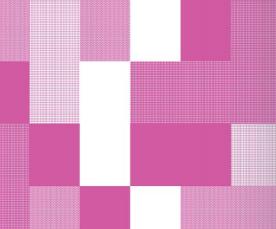
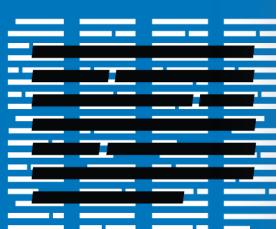
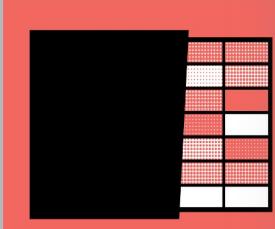
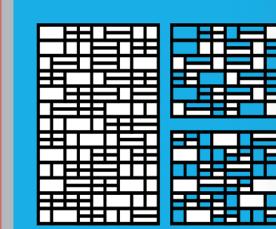
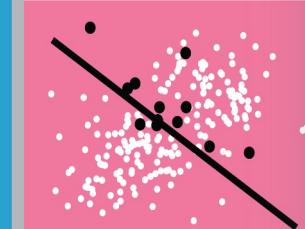
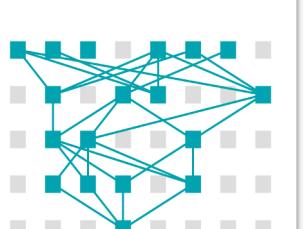
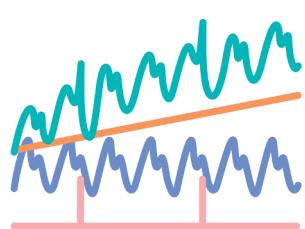
day

The model is 99.87 % confident that is Negative

Sentiment Analysis with Sparklyr and Tensorflow

SENTIMENT ANALYSIS SPARK

Powered by Cloudera Fast Forward Labs Research

<p>Natural Language Generation</p> 	<p>Deep Learning: Image Analysis</p> 	<p>Probabilistic Programming</p> 	<p>Semantic Recommendations</p> 	<p>Federated Learning</p> 	<p>Transfer Learning for Natural Language Processing</p> 	<p>Deep Learning for Anomaly Detection</p> 	<p>Meta-Learning</p> 
<p>Probabilistic Methods for Realtime Streams</p> 	<p>Summarization</p> 	<p>Interpretability</p> 	<p>Multi-Task Learning</p> 	<p>Learning with Limited Labeled Data</p> 	<p>Deep Learning for Image Analysis 2019 Edition</p> 	<p>Causality for Machine Learning</p> 	<p>Structural Time Series</p> 

[Preview all of our research here](#)

How to Use AMPs?

- Just pick one...

The screenshot displays the Cloudera Machine Learning interface. On the left, a dark sidebar menu lists various ML components: Projects, Sessions, Experiments, Model Deployments, Model Registry, Jobs, Applications, User Settings, **AMPs** (which is highlighted with a red box), Runtime Catalog, Site Administration, and Learning Hub. At the bottom of the sidebar, it shows version 2.0.38-b125. The main area is titled "Applied ML Prototypes" and contains a grid of 18 preview cards, each representing a different pre-built ML project:

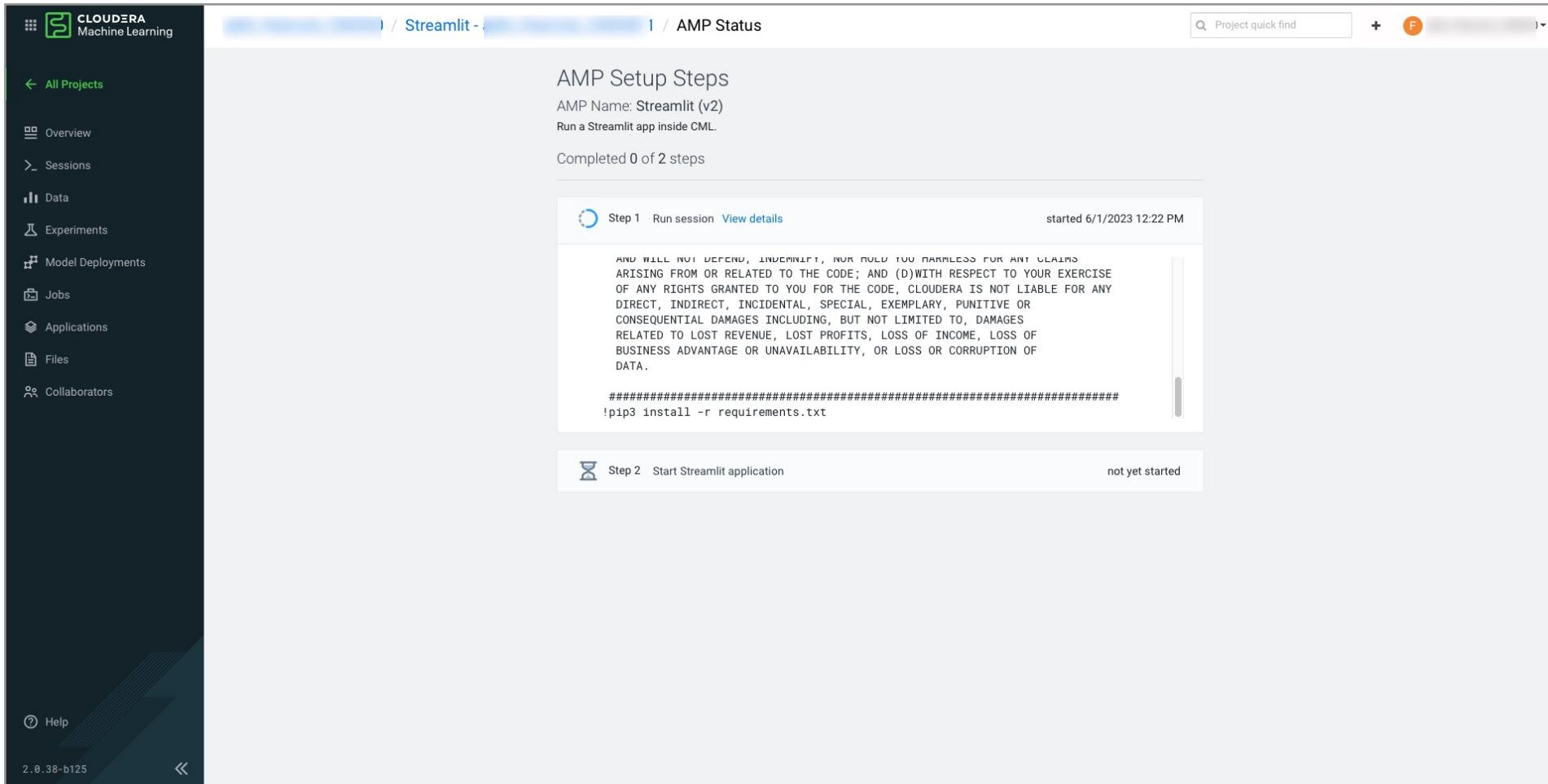
- LLM Chatbot Augmented with Enterprise Data**: CHATBOT, LLM
- Churn Modeling with scikit-learn**: CHURN PREDICTION, LOGISTIC REGRESSION
- Deep Learning for Image Analysis**: COMPUTER VISION, IMAGE ANALYSIS
- Deep Learning for Anomaly Detection**: ANOMALY DETECTION, TENSORFLOW
- NeuralQA**: QUESTION ANSWERING, BERT
- Structural Time Series**: TIME SERIES, PROPHET
- Analyzing News Headlines with SpaCy**: SPACY, NLP
- Question Answering with Wikipedia**: WIKIPEDIA, The Free Encyclopedia
- Deep Learning for Question Answering**: AUTOMATED QUESTION ANSWERING, EXTRACTIVE QUESTI
- Explaining Models with LIME and SHAP**: INTERPRETABILITY, EXPLAINABILITY
- Active Learning**: ACTIVE LEARNING, LEARNING WITH LIMITED LABELED DATA
- MLFlow Tracking**: EXPERIMENT TRACKING
- Few-Shot Text Classification**: NLP, FEW-SHOT LEARNING
- Canceled Flight Prediction**: BINARY CLASSIFICATION, XGBOOST
- Streamlit**: STREAMLIT, APPLICATIONS
- Object Detection Inference Visualized**: COMPUTER VISION, OBJECT DETECTION

A large watermark for "Stream" is visible across the bottom of the main content area.

Cloudera Machine Learning: AMP Catalog – Available from left menu

Click On It

- Sit back and enjoy the show!



Creating and Deploying New AMPs

■ Build new AMPs

- Once a project has been built in Cloudera Machine Learning, you can package it as an AMP
- The ML course exercises are an example of an AMP

■ AMPs can be deployed from

- AMP catalog
- Zip file
- Git repository

New Project

* Project Name
Example AMP

Project Description

Project Visibility
 Private - Only added collaborators can view the project.
 Public - All authenticated users can view this project.

Initial Setup
Blank Template AMPs Local Files Git

Applied ML Prototypes provide components to create a complete project. They may include jobs, models and experiments.

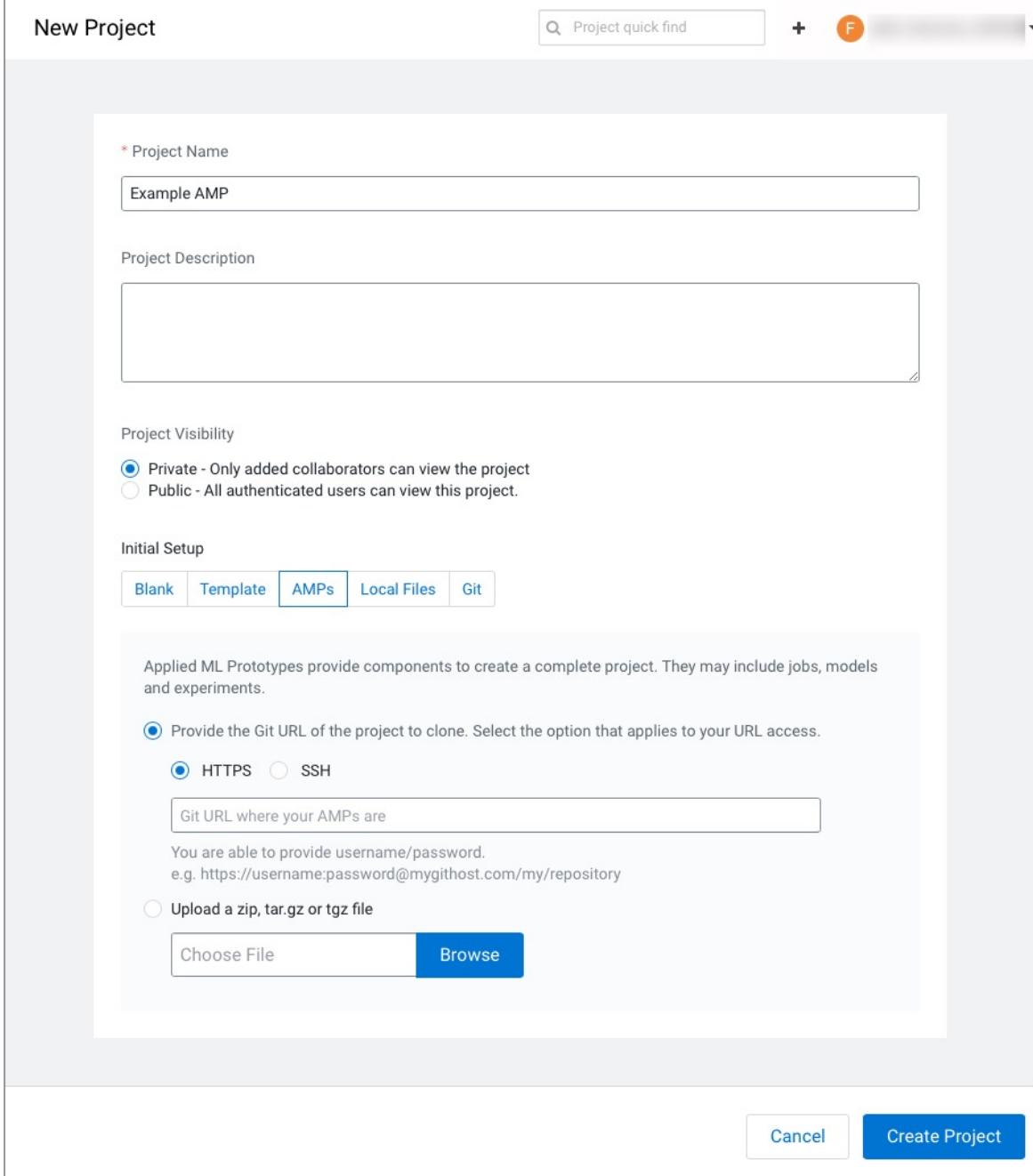
Provide the Git URL of the project to clone. Select the option that applies to your URL access.
 HTTPS SSH

Git URL where your AMPs are
e.g. https://username:password@mygithost.com/my/repository

Upload a zip, tar.gz or tgz file

Choose File Browse

Cancel Create Project



Example AMP Structure: Streamlit AMP

```
.  
├── cml                      # This folder contains scripts that facilitate the project launch on CML.  
├── docs/images                # Storage for the images in this README.  
├── .project-metadata.yaml     # Declarative specification of this project  
├── app.py                     # The Streamlit app script.  
├── LICENSE                    # This code has an Apache 2.0 License  
└── README.md                  # Information about the AMP  
└── requirements.txt           # Python 3 package requirements.
```

- **The `.project-metadata.yaml` file is the only requirement.**
 - The metadata file defines the environmental resources needed by the AMP and the tasks to install the AMP.
- **The `cml` directory typically contains the deployment scripts.**
- See [Creating New AMPs](#) for detailed information.

Chapter Topics

Introduction to AMPs and the Workbench

- Editors and IDE
- Git
- Embedded Web Applications
- AMPs
- **Essential Points**
- Hands-On Exercise: (AMP) Streamlit on CML

Essential Points

- **Run your code from within a session**
 - Can have multiple sessions running in a project
 - Use standard or third-party editor
- **CML provides full access to Git for version control**
- **Create ML web applications/dashboards and easily share them with other business stakeholders**
- **AMPs provide reference example machine learning projects in Cloudera Machine Learning**
 - AMPs are available to install and run from the Cloudera Machine Learning user interface
 - As new AMPs are developed, they will become available in the catalog
 - You can create your own AMPs

Chapter Topics

Introduction to AMPs and the Workbench

- Editors and IDE
- Git
- Embedded Web Applications
- AMPs
- Essential Points
- **Hands-On Exercise: (AMP) Streamlit on CML**

Hands-On Exercise: Streamlit on CML

- **In this exercise, you will**
 - Use an AMP (Applied ML Prototype) to deploy a simple Streamlit application using CML
 - Deploy an AMP to understand how applications work in CML
- **Please refer to the Hands-On Exercise Manual for instructions**



Data Access and Lineage

Data Access and Lineage

- **By the end of this chapter, you will be able to**

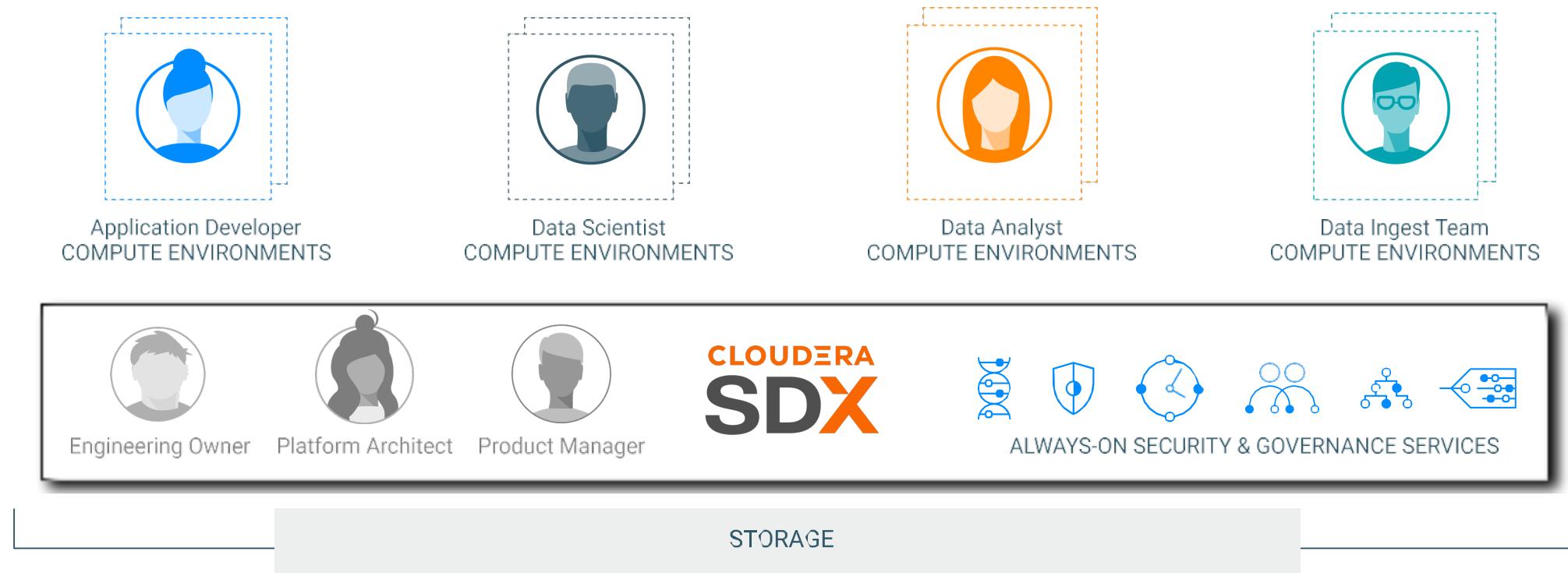
- Identify which tools in Cloudera Data Platform (CDP) to use for key data governance activities
- View and search for entities using the Data Catalog
- Describe how Apache Ranger applies policies to allow or deny access
- View and interpret entity's lineage using Apache Atlas

Chapter Topics

Data Access and Lineage

- **SDX Overview**
- Data Catalog
- Authorization
- Lineage
- Essential Points
- Hands-On Exercise: Data Access

Shared Data Experience (SDX)



- Shared catalog holds the state (structure and business context) of all data
- Unified security model with a consistent set of controls
- Consistent governance model for secure access to all data
- Flexible data ingest and replication to help preserve data integrity

SDX Benefits For a Data Scientist

- A key component within Cloudera SDX is a shared data catalog
 - Enables self-service access to business data
 - Provides consistent security, governance, and management functions that can be leveraged for analytics applications
- For a Data Scientist, you want know
 - What data is available
 - How recent is the data
 - When the data was last modified
 - Is the data clean
 - If the data was transformed
 - Who touched the data
 - How to get access to the data (for example, who to ask and what to ask)

Data Governance in Cloudera Data Platform

- **Cloudera's Shared Data Experience (SDX) applies security and governance services across all workloads**
 - Apache Ranger
 - Apache Atlas
 - Data Catalog
 - Replication Manager
 - Workload Manager

Apache Ranger

Apache Ranger is an open source application to define, administer, and manage security policies

- **Helps manage policies for files, folders, databases, tables, or columns**
 - You can set policies for individual users or groups
 - *Policies can control full access, or partial access using data masking and row level filtering*
 - Policies are enforced consistently across workloads for a data lake
- **Provides a centralized audit location**
 - Tracks all access requests in real time



Apache Ranger

Apache Atlas is an open source application for managing metadata

- **Exchanges metadata with other tools and processes**
 - Captures lineage across components
 - Import existing metadata and models from current tools
 - Export metadata to downstream systems
- **Allows modeling of assets with complex attributes and relationships**
 - Custom metadata structures in a hierarchy taxonomy
 - Classification of assets for the needs of the enterprise
 - Classifications: PII, PHI, PCI, PRIVATE, PUBLIC, CONFIDENTIAL
- **Enables search for assets using classification and attributes**
- **Includes REST API for flexible access**



Apache **Atlas**

Data Catalog

Data Catalog is a service in CDP for managing, securing, and governing data assets

- **A layer over Ranger and Atlas, providing easier access to their information**
- **Features**
 - Search function for quick access to data objects
 - Quick links to Apache Atlas and Apache Ranger
 - Profilers enable automatic gathering and quick viewing of information
 - Cluster Sensitivity Profiler: Finds and tags potentially sensitive data
 - Ranger Audit Profiler: Summarizes audit logs provided by Apache Ranger
 - Hive Column Statistical Profiler: Provides summary statistics for columns in Hive tables
 - Collections of data assets called datasets facilitate collaboration
 - Bookmarking
 - Discussions and commenting



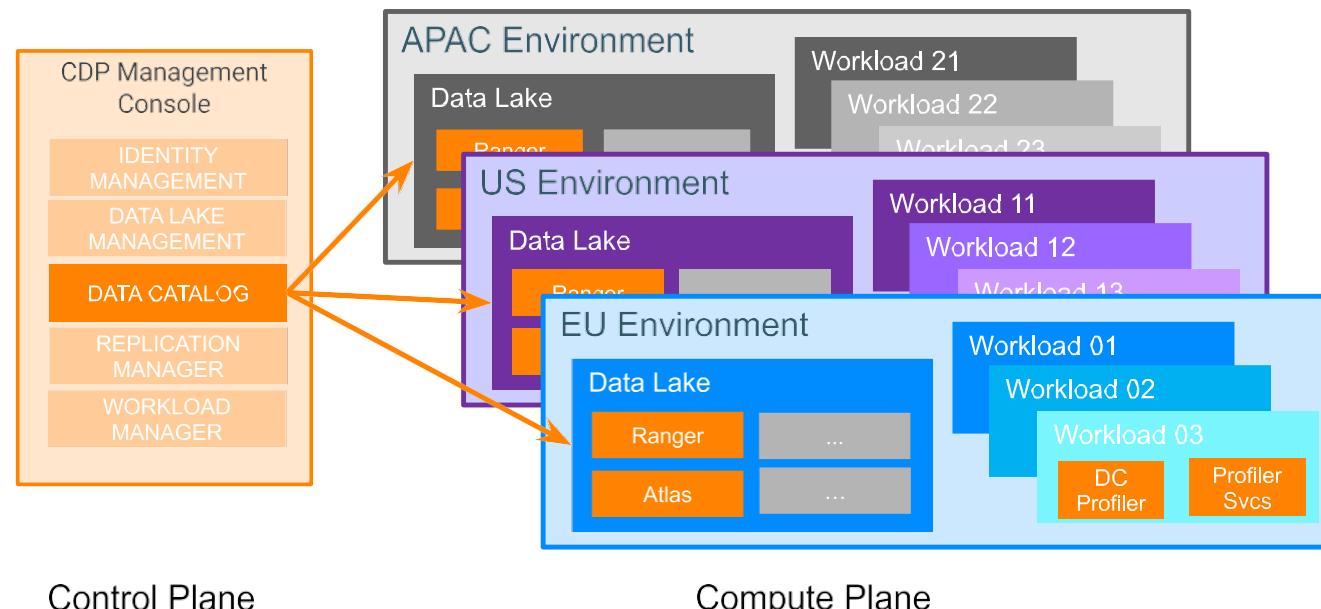
Chapter Topics

Data Access and Lineage

- SDX Overview
- **Data Catalog**
- Authorization
- Lineage
- Essential Points
- Hands-On Exercise: Data Access

Data Catalog Overview

- Data Catalog provides access to items in a particular **Environment**
- Within each Environment are
 - A Data Lake (storage)
 - Atlas and Ranger monitor and control access to the Data Lake
 - Workloads (compute)
 - Profilers and profiler services use dedicated workload resources



Accessing Data Catalog

- Click the Data Catalog icon in the Control Plane on the CDP home page *



Navigating Data Catalog

The screenshot shows the Cloudera Data Catalog interface. On the left, there's a sidebar with links for Search, Datasets, Bookmarks, Profilers, and Atlas Tags. Below that are Get Started, Help, and a search bar. The main area is titled 'Data Catalog / Search' and shows a table of 'Data Lakes'. The table has columns for Type, Name, Location, Owner, and Source. There are also checkboxes for Type and Name. A red box highlights the 'Data Lakes' section, another highlights the search bar, and a third highlights the 'Tool links' in the top right.

Type	Name	Location	Owner	Source
Spark Process	execution-2	/application_1596831586309_0...	-NA-	spark
Spark Process	execution-2	/application_1596831586309_0...	-NA-	spark
HDFS Path	/user/hdfs/uszips-target.csv	/hdfs://case702725-sparktmp...	-NA-	hdfs
Spark Application	Case 702725 application_1...	/application_1596831586309_0...	-NA-	spark
Spark Application	Case 702725 application_1...	/application_1596831586309_0...	-NA-	spark
Spark Process	execution-2	/application_1596831586309_0...	-NA-	spark
AWS S3 Pseudo Dir	/case702725/	/s3://repro-aws-sup-cdp-bucket...	-NA-	aws
Spark Process	execution-2	/application_1597076277511_0...	-NA-	spark
Spark Application	Case 702725 application_1...	/application_1596831586309_0...	-NA-	spark
Spark Application	Case 702725 application_1...	/application_1596831586309_0...	-NA-	spark

- Data Catalog presents information for a specific Data Lake
- Page links provide different features within Data Catalog
- Tool links take you to Atlas and Ranger UI pages for the same Data Lake

Searching in Data Catalog

- Opens to the Search page
 - Fast access to Entities
 - Sufficient for most search needs
- Uses faceted filtering
- Includes filters for specific attributes, for example
 - Owner
 - Database
 - When created

Data Catalog / Search		
<input type="text"/> Search		
Data Lakes	Type	Name
<input type="checkbox"/>	Hive Table	us_customers
<input type="checkbox"/>	Hive Table	txn_stg
<input type="checkbox"/>	Hive Table	txn_final
<input type="checkbox"/>	Hive Table	txn_hist
<input type="checkbox"/>	Hive Table	kudu_txn_final
<input type="checkbox"/>	Hive Table	ext_hist
<input type="checkbox"/>	Hive Table	prov_view
<input type="checkbox"/>	Hive Table	prov_view2
<input type="checkbox"/>	Hive Table	provider_summary
<input type="checkbox"/>	Hive Table	claims_view
<input type="checkbox"/>	Hive Table	claim_savings
<input type="checkbox"/>	Hive Table	consent_data
<input type="checkbox"/>	Hive Table	eu_countries
<input type="checkbox"/>	Hive Table	eventbatchop
<input type="checkbox"/>	Hive Table	telco_churn
<input type="checkbox"/>	Hive Table	employees_masked
<input type="checkbox"/>	Hive Table	employees
<input type="checkbox"/>	Hive Table	uk_employees
<input type="checkbox"/>	Hive Table	eu_employees
<input type="checkbox"/>	Hive Table	tax_2009
<hr/>		
Filters		
TYPE	Clear	
<input checked="" type="checkbox"/>	Hive Table	
<input type="checkbox"/>	HBase Table	
+ Add New Value		
OWNERS	Clear	
DATABASE	Clear	
ENTITY TAG	Clear	
+ Add New Value		
COLUMN TAG	Clear	
+ Add New Value		
CREATED WITHIN	Clear	
<input type="radio"/>	Last 7 days	
<input type="radio"/>	Last 15 days	

Asset Details: Schema

- Hive column statistical profiler provides statistics for each column of a table

Unique values*

Minimum value

Maximum value

Null values

Mean of values

- Use the Schema tab for the Hive table page

Chart Type	Column Name	Type	Unique Values *	Null Values	Max	Min	Mean
bar	age	int	33	NA	84	19	46.34
dot	dateofbirth	date	0	NA			
bar	email	string	52	NA			
bar	id	int	53	NA	979,607	134,841	476,579.68
bar	name	string	49	NA			
bar	phone	string	52	NA			
dot	region	string	4	NA			
bar	salary	int	49	NA	197,537	42,005	114,979.2

*Approximate values as being computed using HLL algorithm.

* Approximation, not actual count

What About the Rest of Your Data?

- Data includes how the entities have changed
 - **Atlas**: Data lineage
- In addition, data also includes
 - **Ranger**: Metadata
 - Who created the table
 - How it was created
 - ...
 - **Atlas**: Audit Logs
 - Details of access to data and metadata

Asset Details: Audit

- Data Catalog provides quick assess to audit information
 - Operational metadata information from Atlas
 - Access information from Ranger

Data Catalog / Asset Details

Name	Type	Data Lake	Dataset	Atlas
ww_customers	HIVE TABLE	demo-gov-ekar-datalake	0	

Overview Schema Policy Audit

Access Type: ALL Result: ALL

Policy ID	Event Time	User	Resource Type	Access Type	Result	Access Enforcer	Client IP
8	09/15/2020 13:43:30 GMT	hive	@table	METADATA OPERATION	ALLOWED	ranger-acl	10.10.1.131
8	09/14/2020 20:42:59 GMT	hive	@table	METADATA OPERATION	ALLOWED	ranger-acl	10.10.1.131
13	09/14/2020 20:14:57 GMT	ekarnowski	@url	READ	ALLOWED	ranger-acl	10.10.1.131
8	09/14/2020 20:14:57 GMT	ekarnowski	@table	CREATE	ALLOWED	ranger-acl	10.10.1.131

Chapter Topics

Data Access and Lineage

- SDX Overview
- Data Catalog
- **Authorization**
- Lineage
- Essential Points
- Hands-On Exercise: Data Access

Authorization

- **Controlling access to a resource**
 - Limit the scope of tools or resources available to a user
 - Limit the scope to data available to a user
- **Depends on authentication**
- **Multiple strategies including**
 - Role-based access control (RBAC)
 - Attribute-based access control (ABAC)



Data Access Using Apache Ranger

- The comprehensive policy management system across CDP and CDF
- Managed using
 - Browser-based UI
 - REST API

A screenshot of the Apache Ranger Service Manager interface. The top navigation bar includes 'Ranger', 'Access Manager', 'Audit', 'Security Zone', and 'Settings'. A user dropdown shows 'adm_bshimel_22829'. The main area is titled 'Service Manager' and displays 14 service configurations in a grid:

- HDFS: cm_hdfs
- YARN: cm_yarn, dh_cml_edu_22829_yarn
- KAFKA: cm_kafka
- ATLAS: cm_atlas
- OZONE: cm_ozone
- S3: cm_s3
- HBASE: cm_hbase
- KNOX: cm_knox
- NIFI: cm_nifi
- ADLS: cm_adls
- SCHEMA-REGISTRY: cm_schema_registry
- HADOOP SQL: Hadoop SQL
- SOLR: cm_solr
- NIFI-REGISTRY: cm_nifi_registry
- KUDU: cm_kudu
- KAFKA-CONNECT: cm_kafka_connect

Each service entry has a '+' button, a checkmark icon, a refresh icon, and a trash bin icon. On the right side, there is a 'Security Zone' dropdown set to 'Select Zone Name' and buttons for 'Import' and 'Export'. The status bar at the bottom right shows 'Last Response Time : 09/22/2022 09:46:46 AM'.

Policies in Apache Ranger

- Policies provide the rules to allow or deny access to an entity based on a
 - Role
 - Group
 - User
- Resource-based policies are associated with a particular service
 - Identify who can use the resource
 - Perform specific actions
 - Access specific assets
 - Create or edit through the plugin for the service
- Attribute or tag based policies
 - Restrict access using classifications and other attributes

Resource-Based Policies

- Policies specific to the tool being used (Hive, HDFS, Kafka, and so on)

The screenshot shows the Ranger UI interface for managing Hadoop SQL Policies. The top navigation bar includes links for Access Manager, Audit, Security Zone, and Settings, along with a user profile for 'adm_bshimel_22829'. The current page is 'Hadoop SQL Policies' under the 'Service Manager' section. A search bar at the top allows users to search for policies. Below the search bar is a table listing 17 policies, each with columns for Policy ID, Policy Name, Policy Labels, Status, Audit Logging, Roles, Groups, Users, and Action.

Policy ID	Policy Name	Policy Labels	Status	Audit Logging	Roles	Groups	Users	Action
8	all - global	--	Enabled	Enabled	--	_c_ranger_admins_52763591	hive beacon dpprofiler hue + More..	<input type="button"/> <input type="button"/> <input type="button"/>
9	all - database, table, column	--	Enabled	Enabled	--	_c_ranger_admins_52763591	hive beacon dpprofiler hue + More..	<input type="button"/> <input type="button"/> <input type="button"/>
10	all - database, table	--	Enabled	Enabled	--	_c_ranger_admins_52763591	hive beacon dpprofiler hue + More..	<input type="button"/> <input type="button"/> <input type="button"/>
11	all - storage-type, storage-url	--	Enabled	Enabled	--	_c_ranger_admins_52763591	hive beacon dpprofiler hue + More..	<input type="button"/> <input type="button"/> <input type="button"/>
12	all - database	--	Enabled	Enabled	--	_c_ranger_admins_52763591 public	hive beacon dpprofiler hue + More..	<input type="button"/> <input type="button"/> <input type="button"/>
13	all - hiveservice	--	Enabled	Enabled	--	_c_ranger_admins_52763591	hive beacon dpprofiler hue + More..	<input type="button"/> <input type="button"/> <input type="button"/>
14	all - database, udf	--	Enabled	Enabled	--	_c_ranger_admins_52763591	hive beacon dpprofiler hue + More..	<input type="button"/> <input type="button"/> <input type="button"/>
15	all - url	--	Enabled	Enabled	--	_c_ranger_admins_52763591	hive beacon dpprofiler hue + More..	<input type="button"/> <input type="button"/> <input type="button"/>
16	default database tables columns	--	Enabled	Enabled	--	public	--	<input type="button"/> <input type="button"/> <input type="button"/>
17	Information_schema database tables columns	--	Enabled	Enabled	--	public	--	<input type="button"/> <input type="button"/> <input type="button"/>

Policy Conditions

- Policies are defined using allow and deny conditions

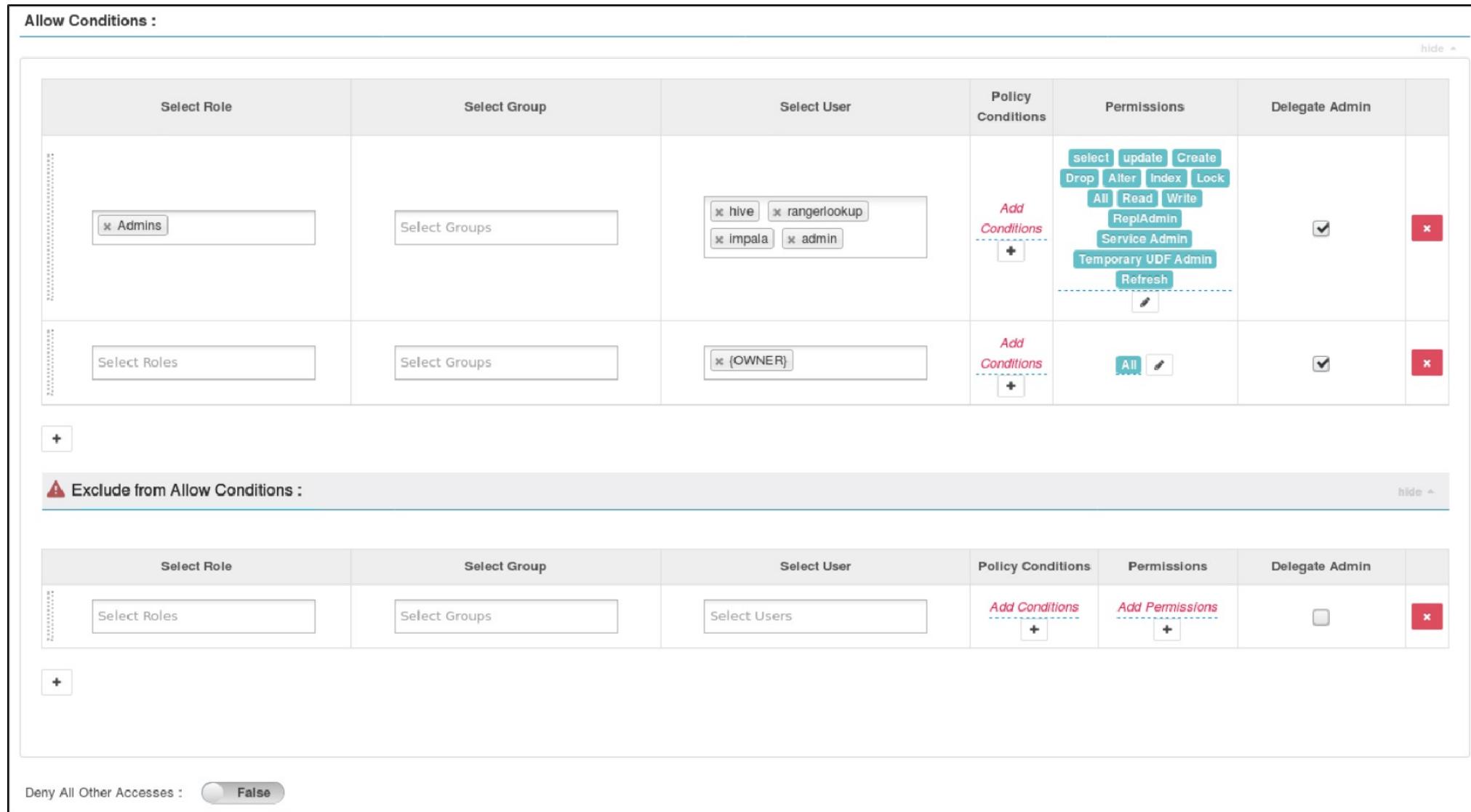
Allow Conditions :

Select Role	Select Group	Select User	Policy Conditions	Permissions	Delegate Admin	
<input type="checkbox"/> Admins	Select Groups	<input type="checkbox"/> hive <input type="checkbox"/> rangerlookup <input type="checkbox"/> impala <input type="checkbox"/> admin	Add Conditions +	<input type="checkbox"/> select <input type="checkbox"/> update <input type="checkbox"/> Create <input type="checkbox"/> Drop <input type="checkbox"/> Alter <input type="checkbox"/> Index <input type="checkbox"/> Lock <input type="checkbox"/> All <input type="checkbox"/> Read <input type="checkbox"/> Write <input type="checkbox"/> ReplAdmin <input type="checkbox"/> Service Admin <input type="checkbox"/> Temporary UDF Admin <input type="checkbox"/> Refresh	<input checked="" type="checkbox"/>	<input type="checkbox"/>
Select Roles	Select Groups	<input type="checkbox"/> {OWNER}	Add Conditions +	<input type="checkbox"/> All	<input checked="" type="checkbox"/>	<input type="checkbox"/>
+ <input type="button" value="Add"/>						

Exclude from Allow Conditions :

Select Role	Select Group	Select User	Policy Conditions	Permissions	Delegate Admin	
Select Roles	Select Groups	Select Users	Add Conditions +	Add Permissions +	<input type="checkbox"/>	<input type="checkbox"/>
+ <input type="button" value="Add"/>						

Deny All Other Accesses : False



View Policies

- List of policies provides overview

The screenshot shows a table with the following data:

Policy ID	Policy Name	Policy Labels	Status	Audit Logging	Roles	Groups	Users	Action
75	all - database, table, column	--	Enabled	Enabled	Admins	--	hive rangerlookup impala admin + More...	
76	all - database, udf	--	Enabled	Enabled	Admins	--	hive rangerlookup impala {OWNER}	
77	access: us_customers_table	--	Enabled	Enabled	Admins	us_employee dpo public	hive	
78	access: ww_customers	--	Enabled	Enabled	--	us_employee eu_employee etl	hive etl_user impala	
79	access: eu_countries	--	Enabled	Enabled	--	public eu_employee	--	
80	prohibit zipcode, insuranceId, bl...	--	Enabled	Enabled	--	analyst	--	
81	prevent UDF create/drop	--	Enabled	Enabled	--	us_employee	--	
90	access: Information Schema pol...	--	Disabled	Enabled	--	us_employee etl	hive	

- Policy ID links to policy details
- Labels for organizing and easy search
- Status (enabled or disabled)
- Roles, groups, and users mentioned in the policy

Policies that Mask Data

- **Masking data provides results without values**
 - Possibly showing the data exists without exposing it
 - Supported for Hive tables (accessed with Hive or Impala)

The screenshot shows a database interface with a query editor at the top containing the following SQL code:

```
1 SELECT surname, streetaddress, country,
2    | age, password, nationalid, ccnumber, mrn, birthday
3 FROM worldwidebank.us_customers
4 LIMIT 50
```

Below the query, a status bar indicates "Query 9041a33259515e1b:e8e7fa0e00000000: 0% Complete (0 out of 1)". Two specific columns are highlighted with orange boxes and arrows pointing to them in the results table below:

- Street address masked (redacted)**: Points to the "streetaddress" column in the results table.
- Password masked (hashed)**: Points to the "password" column in the results table.

The results table has a header row with columns: Query History, Saved Queries, Results (50), surname, streetaddress, country, age, and password. Below the header, there are five data rows:

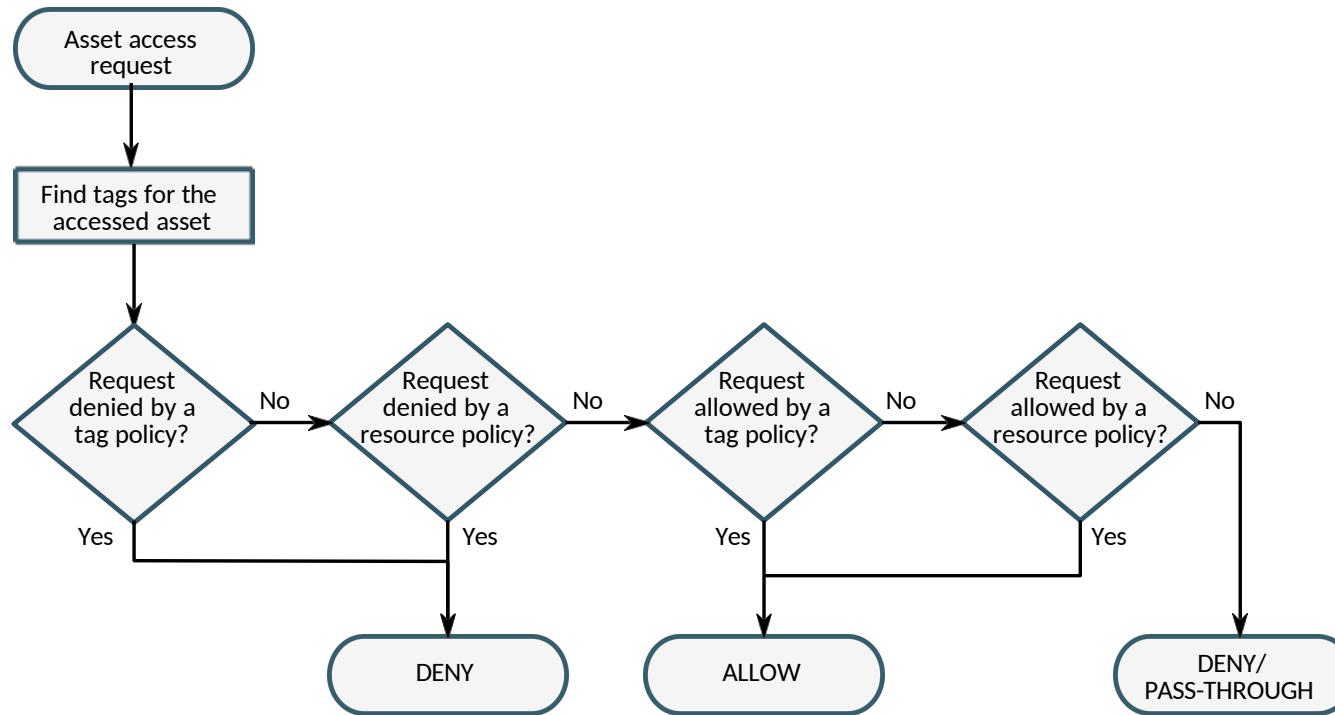
	Query History	Saved Queries	Results (50)	surname	streetaddress	country	age	password
1	Powers	nnnn Xxxxxxx Xxxxx	US	52	76a3fe33eb676cb12b99f00400772e7f5e5abc00950f432a7428d8e356			
2	Whitman	nnnn Xxxxxxx Xxxxx	US	55	6103bc600fc877e5cdf250521e422fc738544bff75165f335e5d2437e67			
3	Marone	nnn Xxxxx Xxxxx	US	47	da01407d5922624b1dd67c22e849cb9ee0ba0fbf3b25ab6e3f78ed234a			
4	Harp	nnnn Xxx Xxxx Xxxx	US	26	717051c13b6bda179f3f1fca42d415524c7e373dcfec265c0973badf28			
5	Pereira	nnnn Xxxxxxx Xxxxxx	US	80	47d2ebe1a1c1146698e22d126282356ac2ccbe0ca60a9d4d6d4f757cd			

Masking Options

Masking Option	Description	Example
Redact	Replace alphabetic characters with “x” and numeric characters with “n”	nnn Xxx Xxxxx
Show last 4	Show only the last four characters	xxx-xx-8366
Show first 4	Show only the first four characters	5.20xxxx+xx
Hash	Replace all characters with a hash of entire cell value	6103bc600fc877
Nullify	Replace all characters with NULL	NULL
Show only year	Show only the year portion of a date string and default the month and day to 01/01	01/01/2022
Custom	Use a valid Hive expression to specify custom value (must return the same data type)	varies

Policy Evaluation Flow

- Deny conditions are checked first, then allow conditions
 - This is opposite of the order they are presented on the policy page in Ranger



Audits in Apache Ranger

- **Use the Audit link in the Ranger menu bar**
- **From this page you can get information about several aspects of the system, for example:**
 - Access: Which data was accessed, when, and by whom
 - Admin: Any administrative tasks, such as creation of policies and assignment of user roles
 - Login Sessions: Login attempts to Ranger Administration; when, by whom, and whether successful
 - User Sync: Service activity data for all usersync processes

Access Audits in Ranger

The screenshot shows the 'Access' tab in the Cloudera Manager interface. A search bar at the top contains the query 'USER: joe_analyst'. Below it, a checkbox labeled 'Exclude Service Users' is unchecked. The main table displays two rows of access logs:

Policy ID	Policy Version	Event Time	Application	User	Service	Name / Type	Permission	Result	Action
80	1	09/09/2020 05:47:42 AM	hiveServer2	joe_analyst	Hadoop SQL Hadoop SQL	worldwidebank/ww_cust... @column	SELECT	select Denied	ra...
89	1	09/09/2020 05:47:42 AM	hiveServer2	joe_analyst	Hadoop SQL Hadoop SQL	worldwidebank/ww_cust... @table	ROW_FILTER	select Allowed	ra...

A modal window titled 'Hive Query' is open over the second row, showing the executed SQL query:

```
select zipcode, insuranceid, bloodtype
from worldwidebank.ww_customers
limit 10
```

- Use to get details of access including
 - Applicable policy ID, with link to policy details
 - Time of access event
 - User
 - Resource accessed, with a pop-up showing the query
 - Result (access allowed or denied)

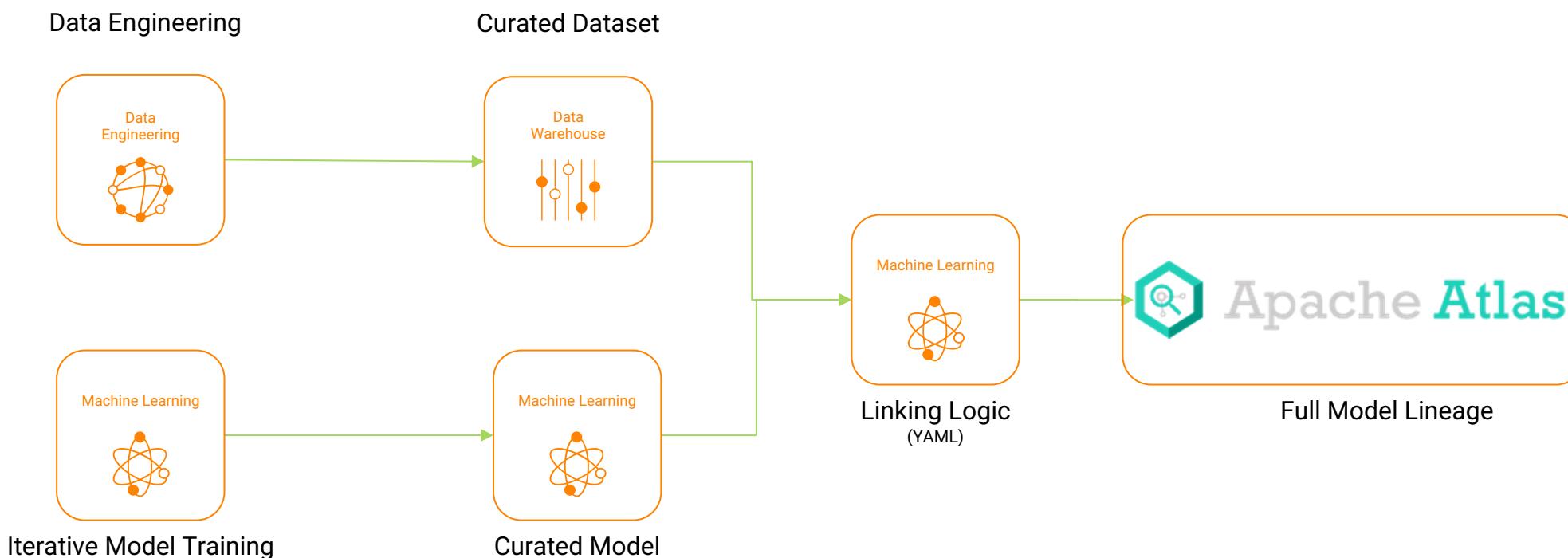
Chapter Topics

Data Access and Lineage

- SDX Overview
- Data Catalog
- Authorization
- **Lineage**
- Essential Points
- Hands-On Exercise: Data Access

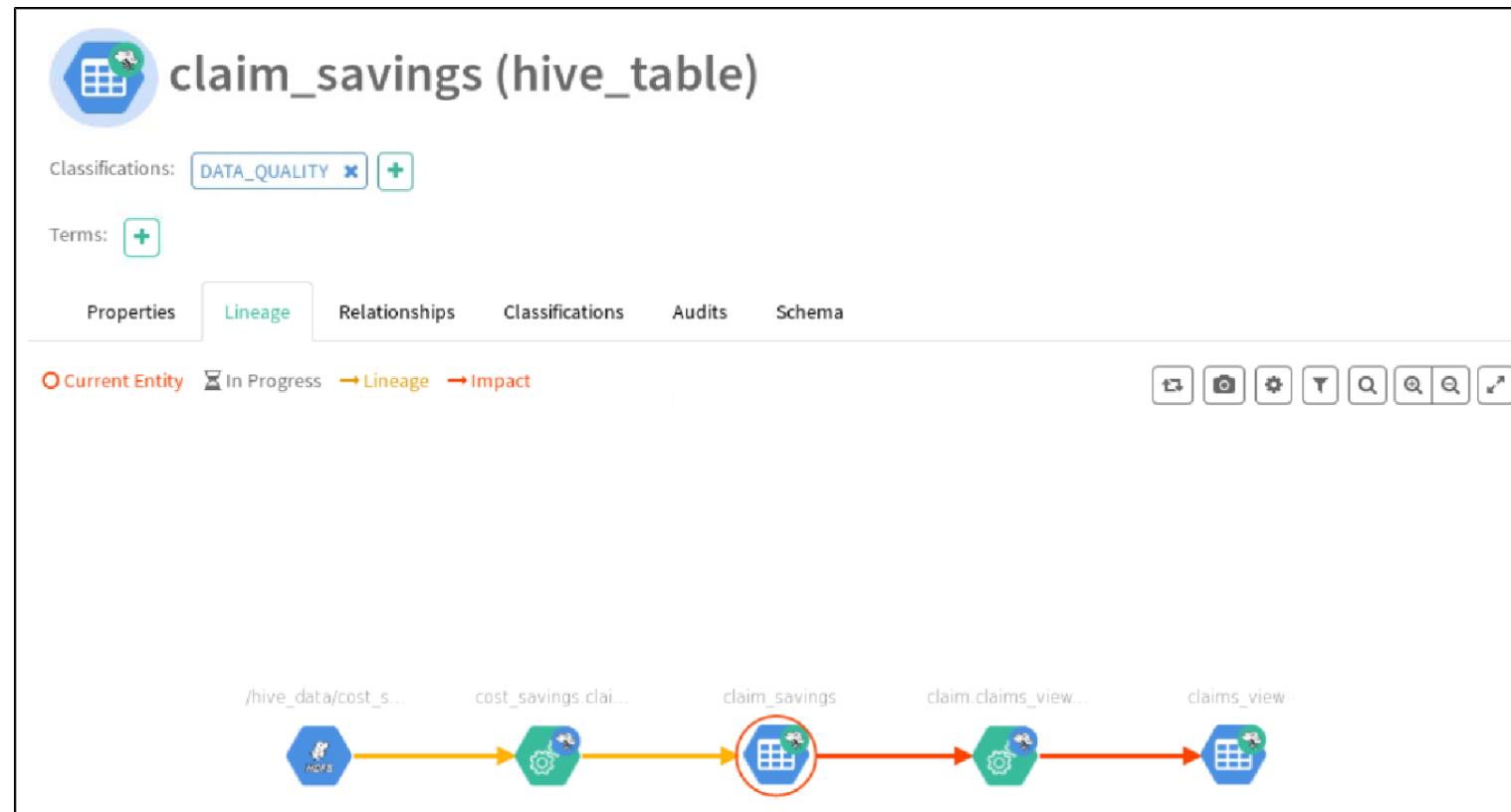
Inspecting Data using Apache Atlas

- View and interpret the lineage for a table or other data object
 - Find when errors were introduced
 - Consider impact a change in a table might have
 - Assess the suitability of a table's data for a given purpose
 - Reduce duplication of assets
 - Informs future project decisions



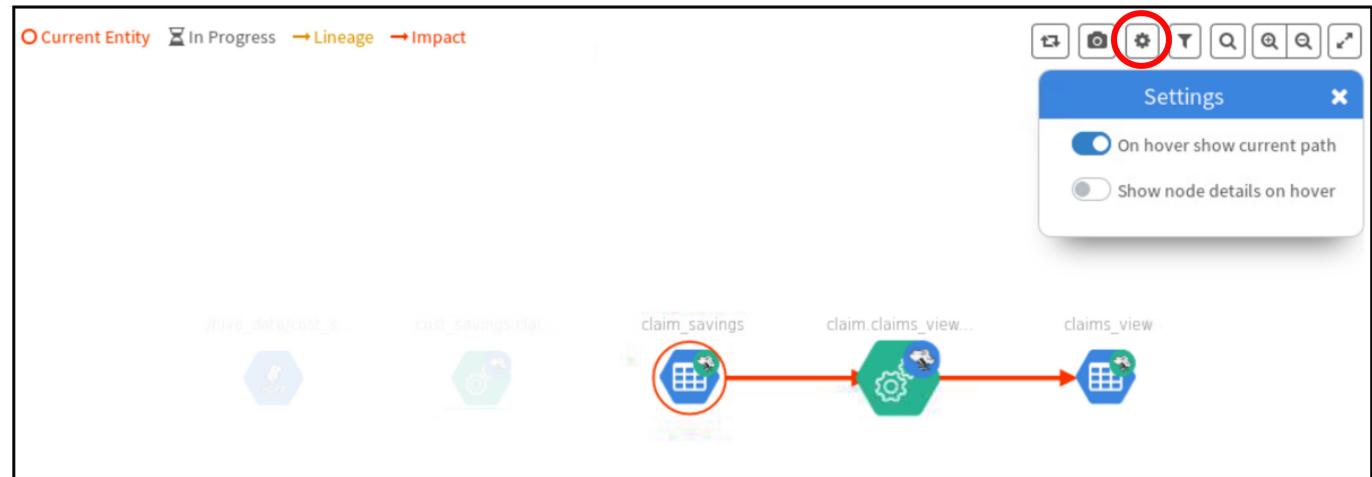
Viewing an Object's Lineage in Apache Atlas

- Use the Lineage tab on an entity's details page
 - Appears as a directed graph
 - Differentiates between lineage (upstream) and impact (downstream)
- Provides information such as who created the entity, and when
 - Does not provide information about access queries



Lineage Nodes

- **Nodes represent data objects**
 - Tables, views, processes, ...
- **Settings control what you see when hovering over a node**
 - Current path (immediate previous and next nodes)
 - Details (entity name and type)



Viewing Entity Details from Lineage

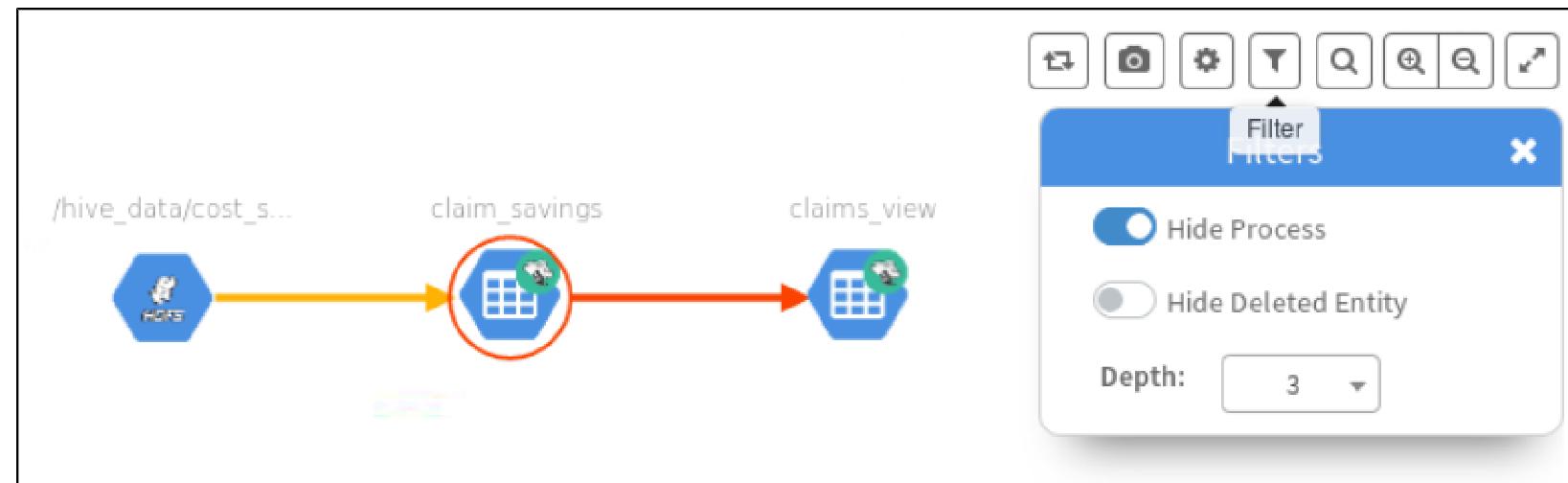
- Click on a node for more detail
- Includes the guid as a link to that node's entity page
- Also shows classifications and terms attached to the entity, if any

Key	Value
guid	e80267cc-8590-4c72-99f9 -bd28ddd1e87a
typeName	hive_process
name	cost_savings.claim_savi ngs@cm:1598027871000
qualifiedName	cost_savings.claim_savi ngs@cm:1598027871000
status	ACTIVE
classifications	N/A
term	N/A

Filtering

■ Show or hide

- Process nodes
- Deleted entities
- Depth determines how many assets (excluding processes) before and after will be shown



Ranger Policies for Apache Atlas

- **Ranger supplies access control for Apache Atlas**
- **Some example policies are provided by default to the following users and groups:**
 - **admin**: the initial Atlas administrator user has full access to all Atlas actions
 - **rangertagsync**: the TagSync service user has read access to entity metadata
 - **rangerlookup**: the Ranger lookup service user has read access to entity metadata
 - **public**: all users are granted access to read Atlas entity metadata
 - **{USER}**: users can save searches for later Atlas sessions

Chapter Topics

Data Access and Lineage

- SDX Overview
- Data Catalog
- Authorization
- Lineage
- **Essential Points**
- Hands-On Exercise: Data Access

Essential Points

- SDX holds the state (structure and business context) of all data
- Data Catalog provides access to a Data Lake and Workloads
- Policies are created in Ranger to provide the rules to allow or deny access to an entity
- Use Altas to display lineage
 - Provides the history of entities and processes

Chapter Topics

Data Access and Lineage

- SDX Overview
- Data Catalog
- Authorization
- Lineage
- Essential Points
- **Hands-On Exercise: Data Access**

Hands-On Exercise: Data Access

- **In this exercise, you will**
 - View policies and identify users that have access to specific data or services
 - Inspect the lineage details of data assets
- **Please refer to the Hands-On Exercise Manual for instructions**



Data Visualization in CML

Data Visualization in CML

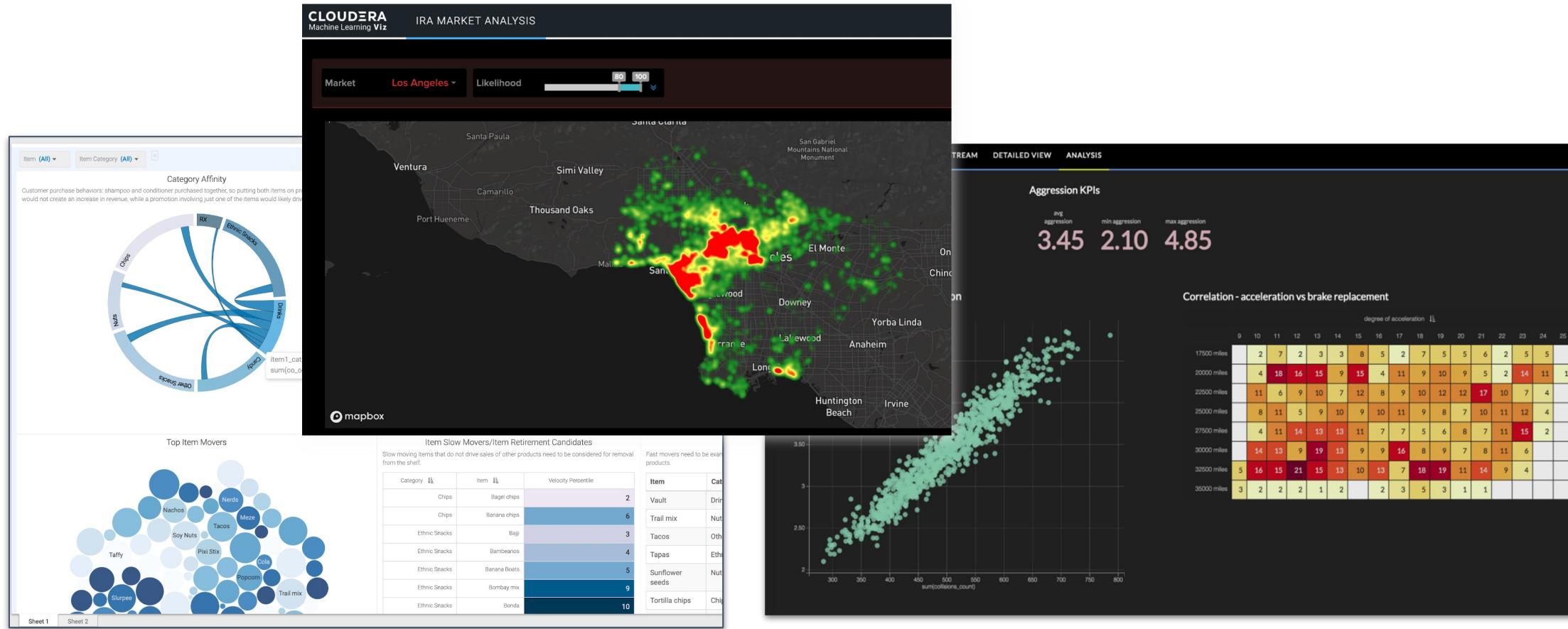
- **By the end of this chapter, you will be able to**
 - Understand the importance of data visualization in the context of data science
 - List the concepts used in the CML Data Visualization application
 - Create your own dashboard in CML Data Visualization

Chapter Topics

Data Visualization in CML

- **Data Visualization Overview**
- CDP Data Visualization Concepts
- Using Data Visualization in CML
- Essential Points
- Hands-On Exercise: Build a Visualization Application

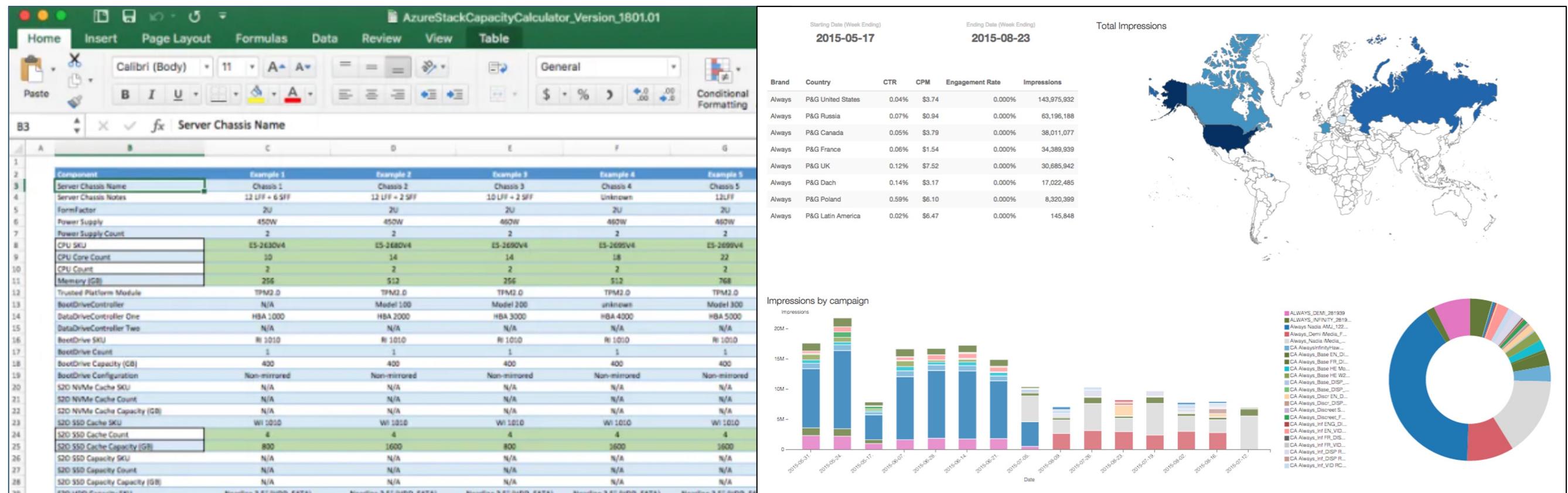
What is CDP Data Visualization?



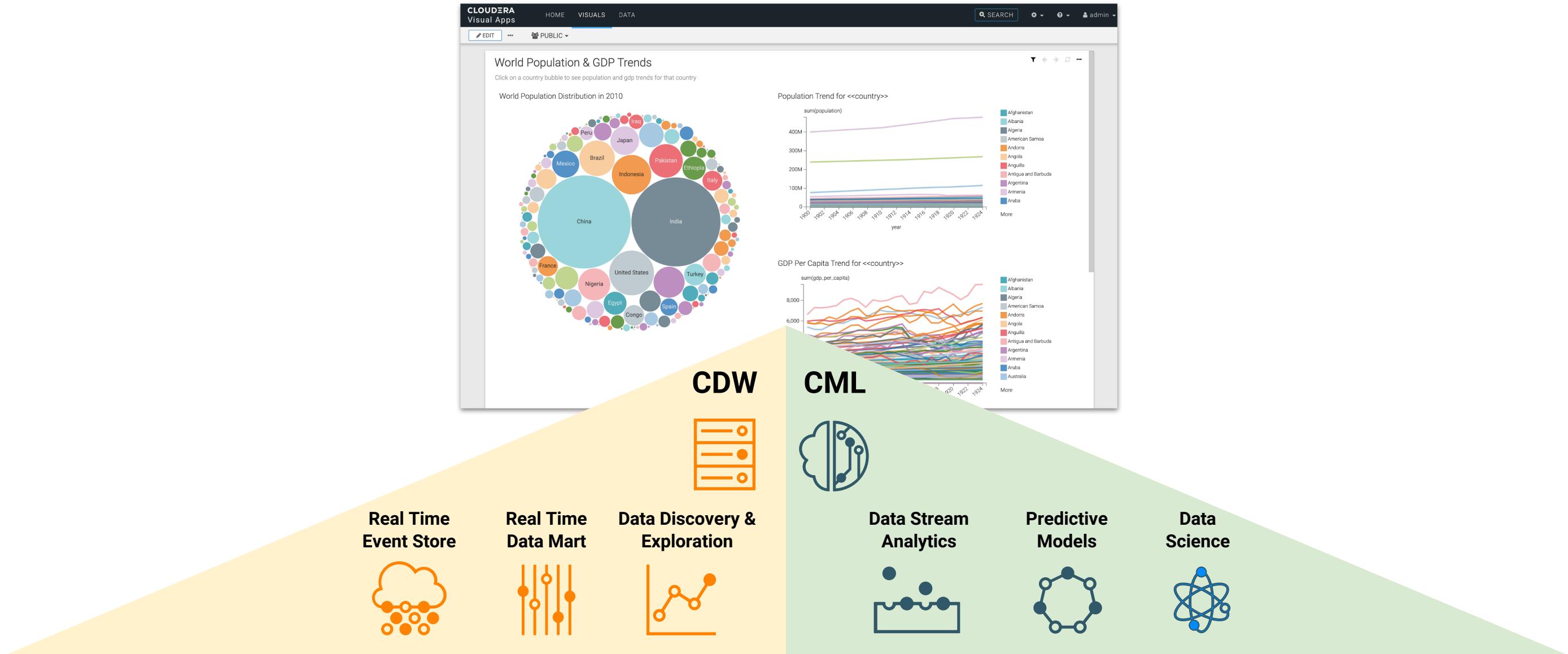
CDP Data Visualization (CDV) enables data engineers, business analysts, and data scientists to **quickly and easily explore data, collaborate, and communicate explainable insights across the data lifecycle.**

Visualizations Are Essential

- Table-based data is great for calculation and organization, but hard to use for decision making when working with large sets of data
- Data visualizations enable humans to make inferences and draw conclusions about large sets of data based on visual input alone



Bring All Your Data Together



Native Data Visualizations in CML

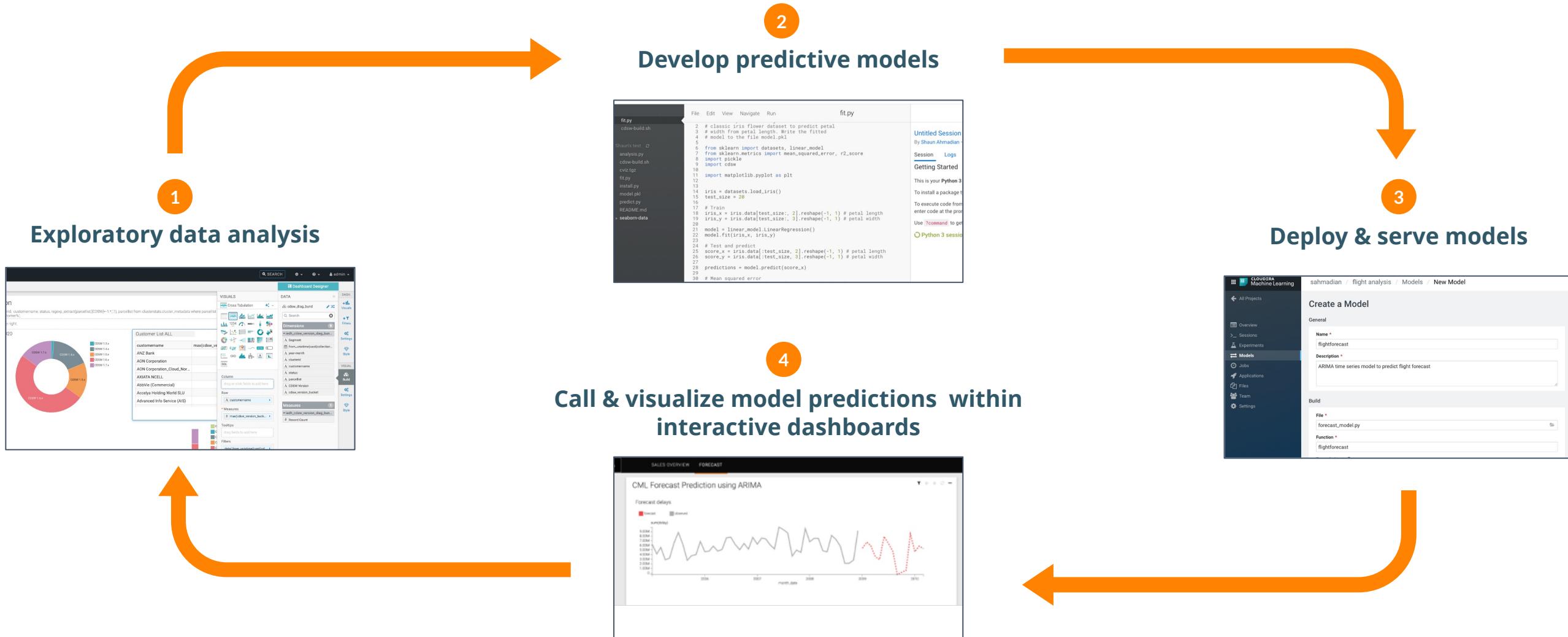
The screenshot displays the Cloudera Machine Learning (CML) interface, specifically the 'Data' section. On the left, a sidebar lists navigation options: All Projects, Overview, Sessions, Data (which is selected and highlighted in green), Experiments, Models, Jobs, Applications, Files, Collaborators, and Project Settings. The main area shows a grid of 18 data visualization cards, each with a thumbnail and a title. The cards are arranged in three rows of six. The titles of the visualizations are:

- Ride Dashboard
- Deficiency Details: <<county:Queens>>
- State of NYC
- Sample App
- Store Details<<owner_name:>>
- Cereal Comparisons
- Earthquakes Around the World
- Life Expectancy Dashboard
- World Population & GDP Trends
- Animated world population - GDP vs life
- US State Population Trends
- Census Dashboard
- Global Threats
- Time & Industry Threat View
- Inspector View
- Consumer View
- Iris species w/ images
- Taxi rides application

At the top of the main area, there are buttons for 'NEW DASHBOARD' and 'NEW APP', a search bar, and action buttons for 'EXPORT' and 'DELETE'. The top right corner shows a user profile and the identifier 'dev_20_22829'.

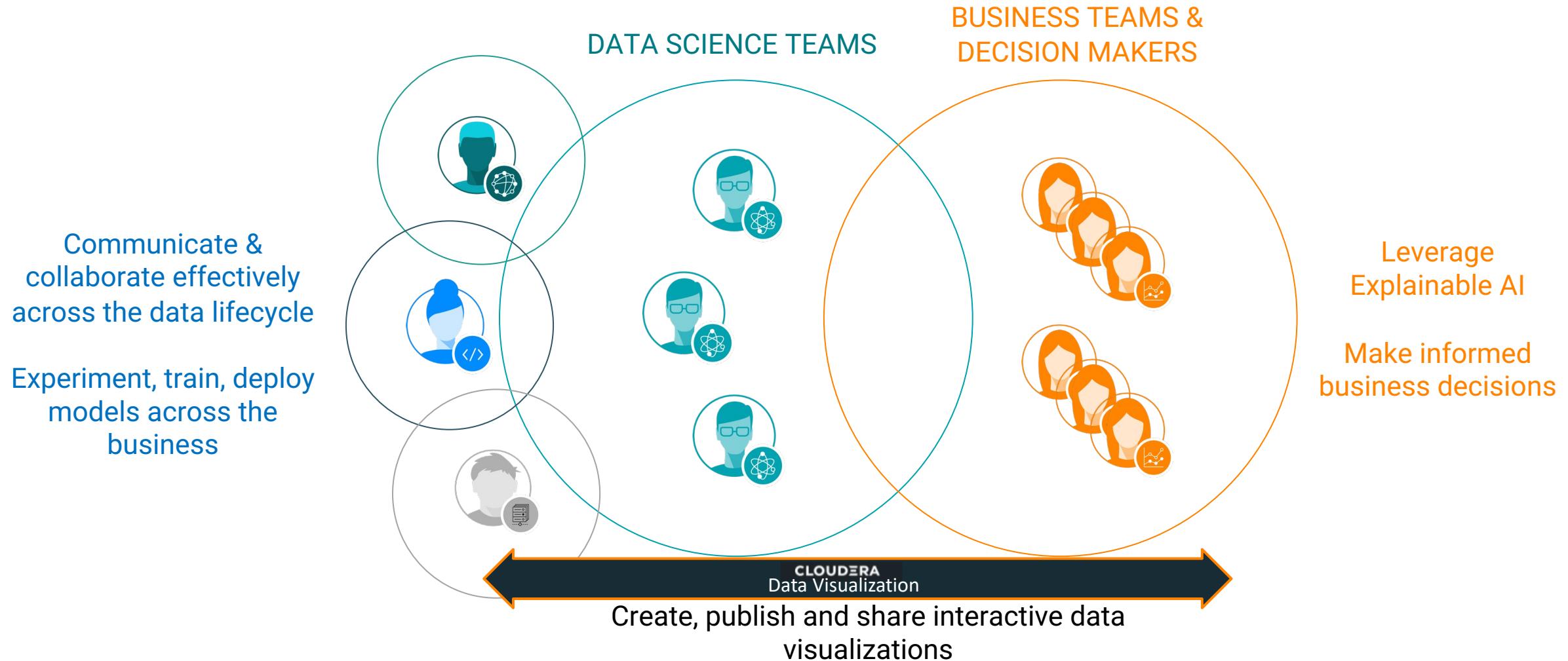
Integrated Data Visualizations for ML Workflows

Create fast, explainable insights from ML models



Visual Collaboration for Getting to Production

Accelerating Production ML Workflows from Raw Data to Business Impact



Visualize, Share, and Collaborate



FAST, SELF-SERVICE DATA EXPLORATION

- Intuitive drag & drop UI
- Integrated with CDP, no moving data or data silos
- Inherently secure with SDX - no data extraction needed



EXPEDITE CROSS-TEAM COLLABORATION

- Self-service for everyone
- Fast insight sharing across the lifecycle
- Same sharable experience everywhere



POWER ANALYTICAL AUTOMATION

- Real time custom dashboards natively in CDP
- Visualize across the data lifecycle to discover optimization opportunities

Visualization Benefits

- Work within the **workflow of ML**
- Simple drag & drop interface
- Interactive, dynamic dashboard
- Completely **web based** for integration through links, embedding, HTML/JS
- Many ways to **share and collaborate**
 - Bookmark link
 - Emailed reports
 - Customized applications
- Access to all of CDP sources (**Impala, Hive, ML models, etc.**)

Chapter Topics

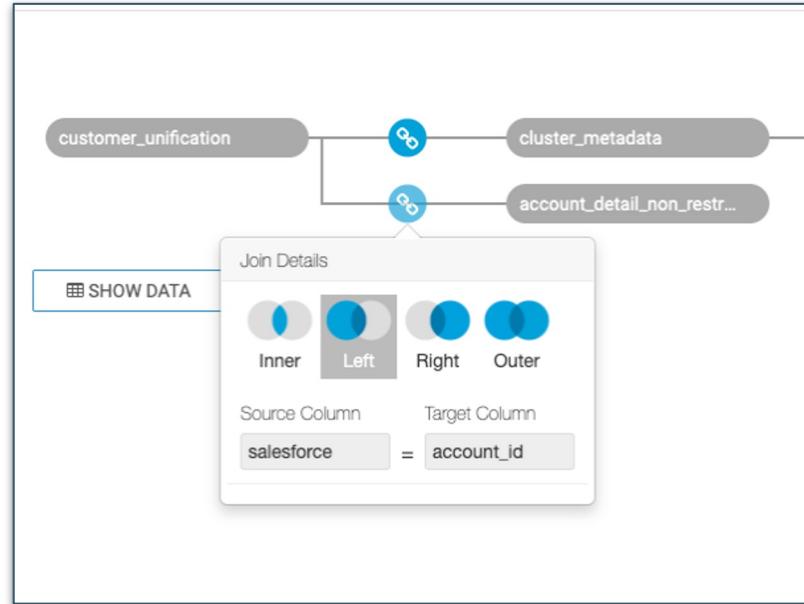
Data Visualization in CML

- Data Visualization Overview
- **CDP Data Visualization Concepts**
- Using Data Visualization in CML
- Essential Points
- Hands-On Exercise: Build a Visualization Application

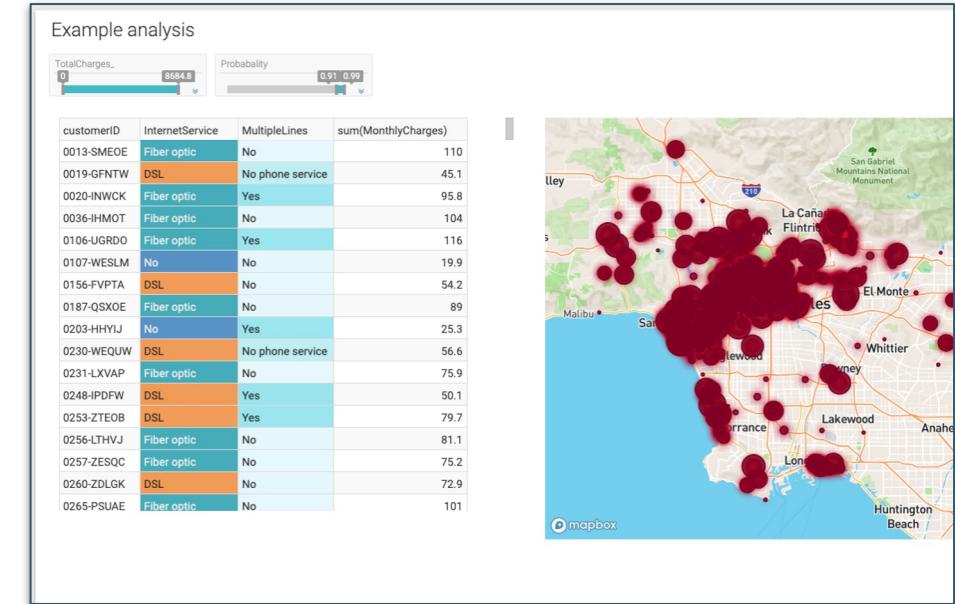
Build, Publish, and Share



**Visual Types &
Custom Extensions**



**Data Modeling for sharable
logical semantic layer**



**UI-powered custom interactive
dashboards & applications**

Concepts

- **Connections**
- **Datasets**
- **Visuals**
- **Dashboards**
- **Applications**

Connections

- **Create and manage connections to many types of external data sources such as**
 - SQL (Impala, Hive, MySQL)
 - Events and time series data (Impala over Kudu)
 - Unstructured data (Solr)
 - ML workloads (Spark)
- **Using Cloudera Machine Learning (CML), you can**
 - Connect to an Impala or a Hive data warehouse
 - Tie in data from predictive CML models
- **Using CDP Public Cloud with Cloudera Data Warehouse (CDW)**
 - The data connection is automatically set up, but you can connect to other data sources as well

Supported Types

- Impala
- Hive
- Druid
- MariaDB
- MySQL
- PostgreSQL
- Solr
- Spark SQL
- SQL Stream Builder
- SQLite

Datasets

- **The foundation and starting point for visualizing your data**
 - The ***semantic layer on top of your data tables*** and views in the data store
 - Allows you to model it into what you need for your visual application without changing underlying data or tables
- **Represents a single data table or data matrix from several tables on the same connection**
- **Can be modeled using**
 - Table joins
 - Calculated fields
 - Modification of data types, dataset fields, and default aggregation of fields

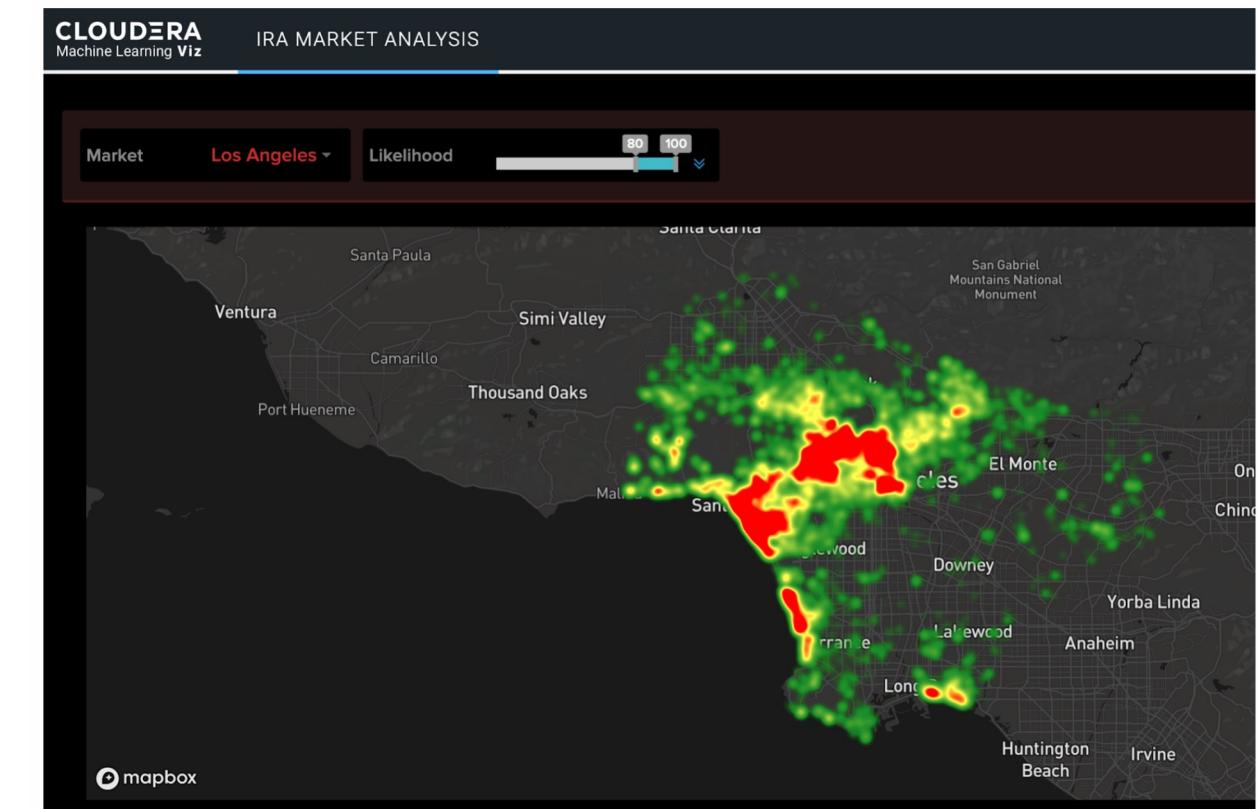
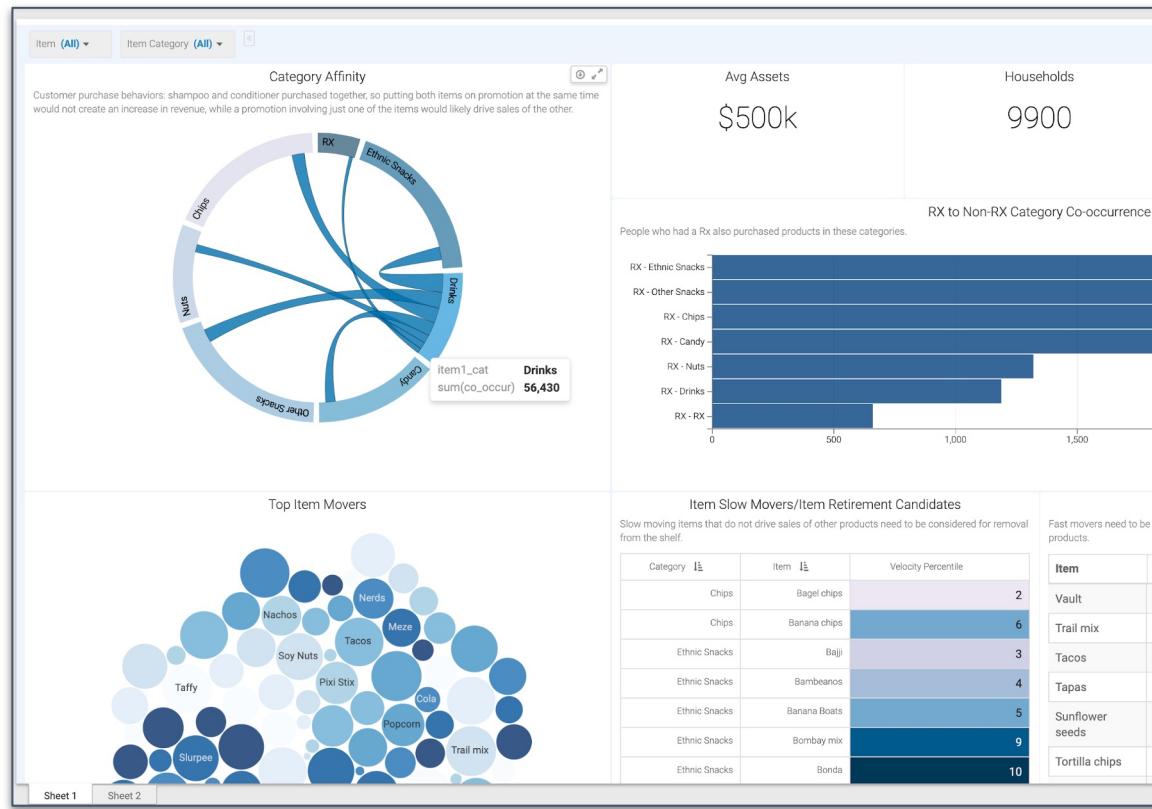
Visuals

- A visual is a *single piece of visualized data*,
for example
 - Pie charts
 - Histograms
 - Heatmaps
- It translates large data sets and metrics into a visual representations
 - Easier to identify insights about the information represented in the data
- CDP Data Visualization has a rich offering of different types of visualization to assist you in analyzing your data



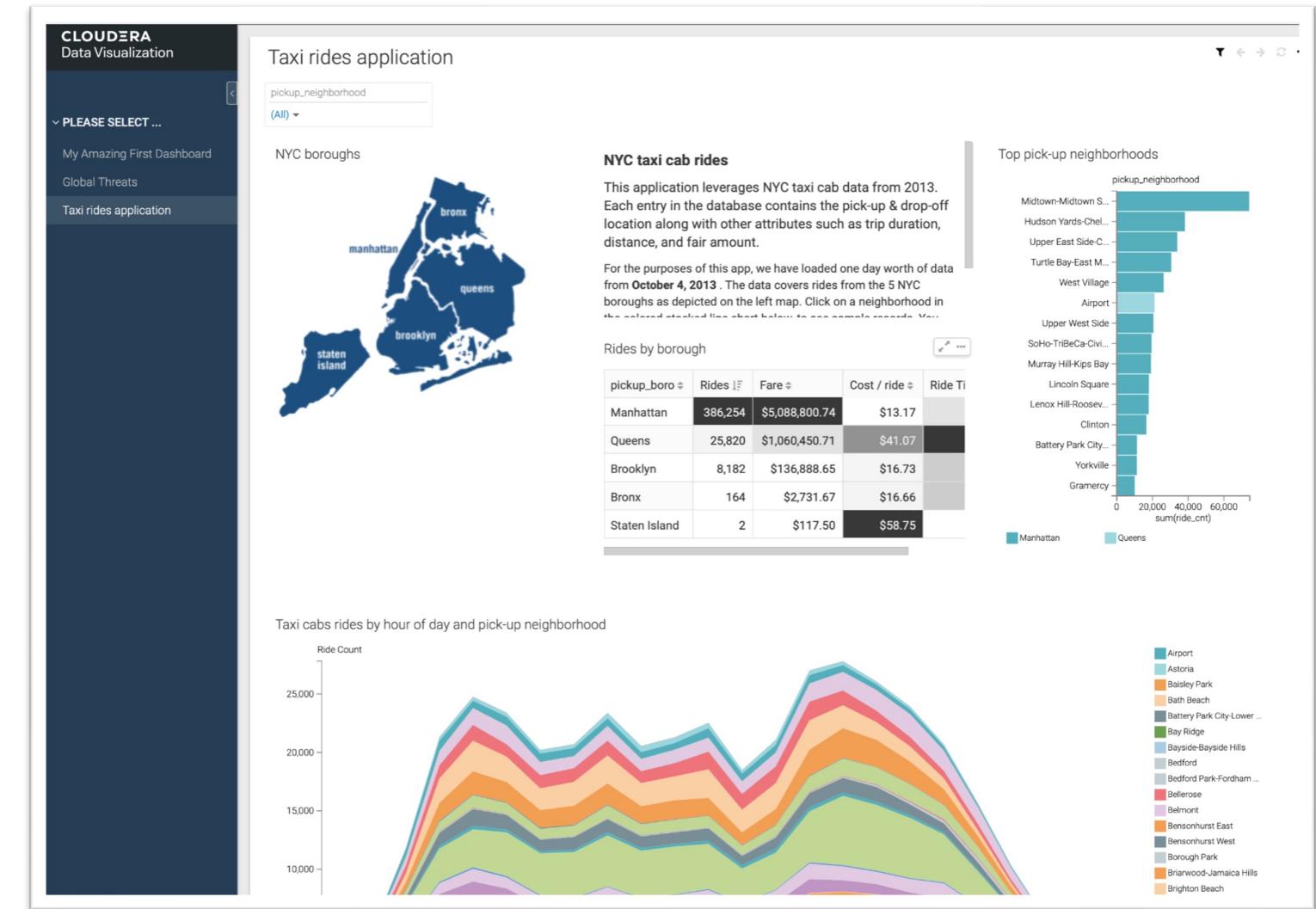
Dashboards

- CDP Data Visualization dashboards consolidate related visualizations
 - Display and link visuals that are based on different datasets across different connections
 - Provide optional run-time filtering on all referenced information



Applications (Apps)

- A collection of dashboards tied together that can be launched as a standalone, branded data visualization tool
- The App Designer interface enables you to
 - Build applications
 - Customize the tabs of the apps
 - Populate the tabs with relevant dashboards



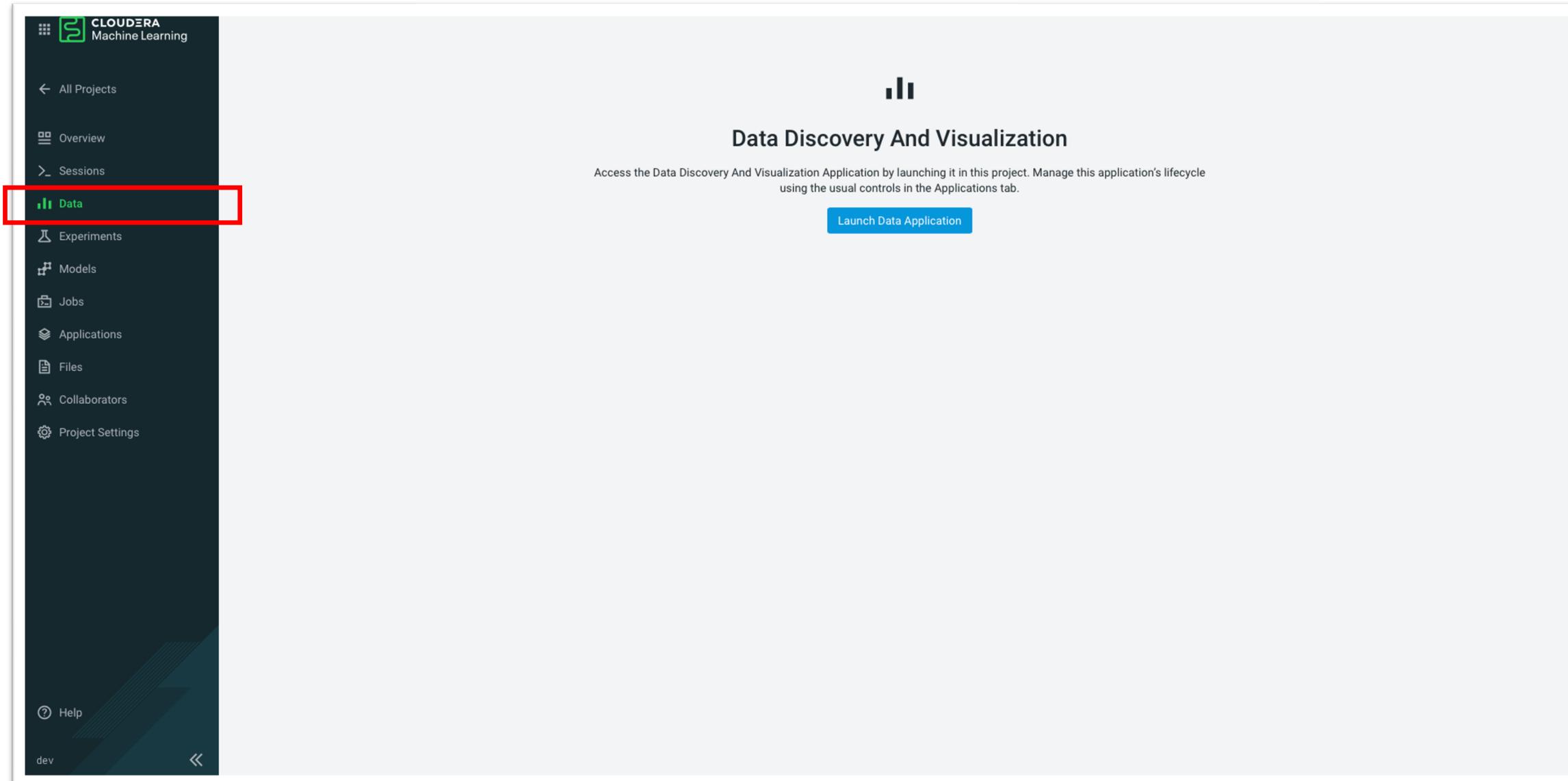
Chapter Topics

Data Visualization in CML

- Data Visualization Overview
- CDP Data Visualization Concepts
- **Using Data Visualization in CML**
- Essential Points
- Hands-On Exercise: Build a Visualization Application

Creating Data Visualization Application

- Create a new application that hosts visualization by selecting Data on the left sidebar



CDV Home Page

The screenshot shows the Cloudera Machine Learning (CDV) Home Page. The left sidebar includes links for All Projects, Overview, Sessions, Data (highlighted with a red box), Experiments, Models, Jobs, Applications, Files, Collaborators, and Project Settings. The main content area displays the "Get Started" section with five numbered steps: 1. Sync Connections, 2. Explore with SQL, 3. Create a Dashboard, 4. CML Notebook, and 5. What's Next?. Step 1 shows a comparison between the Cloudera Management Console and the CDV interface, with numbered callouts 1 through 5 pointing to specific UI elements. Step 1 also includes a sub-section titled "User Settings" showing environment variables. The right side of the page contains instructions for setting up data connections and a list of recent queries and metrics.

HOME SQL VISUALS DATASETS

adm_bshimel_22829

Get Started

1 Sync Connections

2 Explore with SQL

3 Create a Dashboard

4 CML Notebook

5 What's Next?

Recent Queries

17 DASHBOARDS 1 APPS 13 DATASETS 2 CONNECTIONS

Data Connections may already be setup for you to use on the [datasets](#) page.
If they are not, you will need to set up your workspace password:

1. On the Cloudera Management Console, select your username
2. View your profile, and set a workload password using the "Set Workload Password" link
3. In your ML Workspace, go to "User Settings" page
4. Under "Environmental Variables" tab, set your "WORKLOAD_PASSWORD" variable to the same password from your Cloudera Management Console profile
5. Restart the "Data Discovery and Visualization" application in CML

Home Page Views

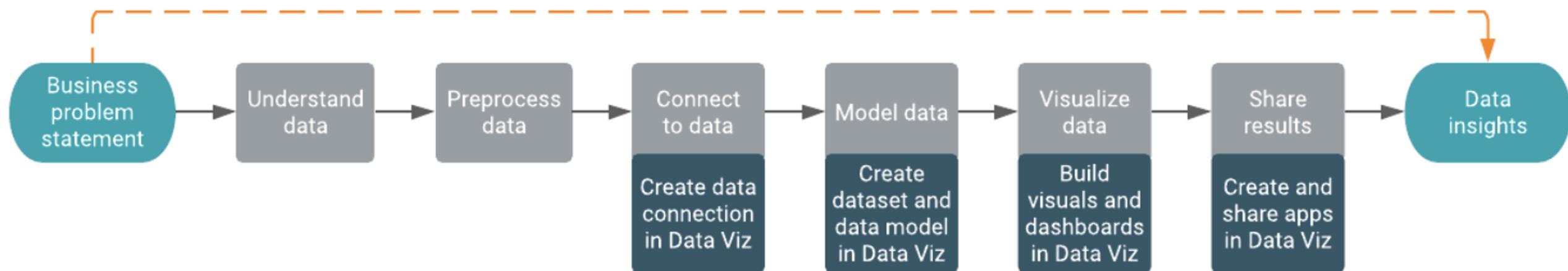
- The interface of Data Visualization has three views:

- HOME
- VISUALS
- DATA

The screenshot shows the Cloudera Data Visualization interface. At the top, there is a navigation bar with three tabs: HOME (highlighted with a red box), VISUALS, and DATA. A yellow callout bubble labeled "Views" points to this navigation bar. Below the navigation bar is a statistics banner displaying the following information: 16 DASHBOARDS, 1 APPS, 12 DATASETS, 0 QUERIES, and 0 TOTAL VIEWS. A yellow callout bubble points to this banner with the text: "Statistics banner shows the number of Dashboards, Apps, Datasets, Queries and Total Views that you can access". The main content area is divided into sections: "Last Viewed by You" and "Recently Created by You", each showing a grid of visual preview cards. A yellow callout bubble points to this area with the text: "Visuals preview area provides quick access to the existing visuals and dashboards". On the right side, there is a sidebar with a user profile (vizeapps_admin) and sections for "NEW DASHBOARD", "NEW APP", and "LEARN". The "LEARN" section includes links for "Get Started", "What's New in 6.3.6", and "Documentation".

General Workflow

1. Create a data connection
2. Create a dataset using your data connection
3. Create a dashboard based on your new dataset
4. Add visuals to your dashboard
5. Create an application to share your dashboard with business users



Chapter Topics

Data Visualization in CML

- Data Visualization Overview
- CDP Data Visualization Concepts
- Using Data Visualization in CML
- **Essential Points**
- Hands-On Exercise: Build a Visualization Application

Essential Points

- **CDP Data Visualization provides out of the box functionality without additional integration efforts, moving data, or creating security issues**
 - Easily and quickly build interactive dashboards and instantly share insights across your business
 - Fully integrated data visualization from within ML Workflows
 - More than 34 visual types are available which help in representing the data in the most suitable format rather than using rows and columns to present the information
 - In addition to out-of-the-box visuals, you can also build applications using custom extensions

Bibliography

- Cloudera Data Visualization Documentation

Chapter Topics

Data Visualization in CML

- Data Visualization Overview
- CDP Data Visualization Concepts
- Using Data Visualization in CML
- Essential Points
- **Hands-On Exercise: Build a Visualization Application**

Hands-On Exercise: Build a Visualization Application

- In this exercise, you will use the DuoCar data to
 - Connect to the data
 - Create a dataset
 - Display data using different visuals
 - Create an application
- Please refer to the Hands-On Exercise Manual for instructions



Experiments

Data Visualization in CML

- **By the end of this chapter, you will be able to**
 - Understand how you can run Experiments with CML
 - Understand how you can monitor your Experiments when they are running
 - Understand how you can track metrics and files in Experiments

Chapter Topics

Experiments

- **Experiments in CML**
- Essential Points
- Hands-On Exercise: Running an Experiment

Experiments in CML

- Machine Learning requires experimenting with a wide range of datasets and algorithms to build a model that maximizes a target metric.
- The Experiment tracking feature allows tracking of:
 - Parameters,
 - Code versions,
 - Metrics, and
 - Output files.
- CML Experiments are compatible with [MLflow™ tracking API](#).



The functionality described in this document is for the new version of the Experiments feature. For information on the legacy feature, see the [documentation](#).

MLflow is a trademark of LF Projects, LLC.

Running Experiments

- **To run an experiment**
 - Create a new session using ML Runtimes,
 - Experiment runs cannot be created from sessions using Legacy Engine.
 - Create a new Job, or
 - Create a new Model
- **No need to install the MLflow library**
 - The MLflow client library is installed in the ML Runtime by default

Best practice: It's useful to display two windows while creating runs for your experiments: one window displays the Experiments tab and another displays the MLflow Session.

Using the MLflow API

- **First import the MLflow in your code**

```
import mlflow
```

- **Set the active experiment**

- If the experiment does not exist, it will be created.

```
mlflow.set_experiment("My Grand Experiment")
```

Using the MLflow API

■ Start a run

```
mlflow.start_run()
```

- Starts a new run
- You do not need to call `start_run()` explicitly, calling one of the logging functions with no active run automatically starts a new one.

Using the MLflow API

- **Track parameters, metrics, tags, and artifacts**

```
# Log a parameter  
mlflow.log_param("input", 5)  
  
# Log a metric  
mlflow.log_metric("score", 100)  
  
# Set a tag  
mlflow.set_tag("release.version", "1.0")  
  
# Log an artifact  
with open("data/features.txt", 'w') as f:  
    f.write(features)  
mlflow.log_artifact("features.txt")
```

- For more information on MLflow API commands used for tracking see [MLflow Tracking](#)

Using the MLflow API

- End a run

```
mlflow.end_run()
```

- Ends the active run

Complete Example

- CML allows you to track metrics and artifacts in your experiments
- First import the MLflow in your code

```
import mlflow

mlflow.set_experiment("My Grand Experiment")
mlflow.start_run()

mlflow.log_param("input", 5)

mlflow.log_metric("score", 100)

with open("data/features.txt", 'w') as f:
    f.write(features)
mlflow.log_artifact("features.txt")

mlflow.end_run()
```

Monitoring Experiments

The Experiment tab

sci_3_23727 / Experiments - Student 3 / Experiments / Add It Up

Project quick find + sci_3_23727

Experiment BETA

Experiment Name: Add It Up

Experiment ID: hb22-uyvm-n668-t5yg

Artifact Location: /home/cdsweb/experiments/hb22-uyvm-n668-t5yg

Notes

Runs (3)

Search: metrics.rmse < 1 and params.model = "true" and tags.miflow.source.type="LOCAL"

Status	Start Time	Run Name	Duration	User	Source	Version	Models
<input type="checkbox"/>	2023-07-31 12:04:58	gupu-uzj0-06w...	51ms	sci_3_23727	ipython3	a5e808	-
<input type="checkbox"/>	2023-07-31 12:07:01	vah3-ajsa-mrwl...	131ms	sci_3_23727	ipython3	a5e808	-
<input type="checkbox"/>	2023-07-31 12:09:23	46uk-e47d-k59...	137ms	sci_3_23727	ipython3	a5e808	-

Parameters

Input	Count	Sum	Tags
[20, 30]	50	6cy763owsy2aif...	
[20, 30]	2	pj9nzm3qq48u...	
[20, 30, 40]	90	o5vtzzi3mm67c...	

Metrics

Tags	Count	Sum
engineID	50	6cy763owsy2aif...
	50	pj9nzm3qq48u...
	90	o5vtzzi3mm67c...

Tags

Specific Run

Metrics

Parameters

Known Issues and Limitations

- Currently, CML only supports Python for experiment tracking.
- The version column in the runs table is empty for every run. In a future release, this will show a git commit SHA for projects using git.
- There is currently no mechanism for registering a model to a Model Registry. In a future release, you will be able to register models to a Model Registry and then deploy Model REST APIs with those models.
- Browsing an empty experiment will display a spinner that doesn't go away.
- Running an experiment from the workbench (from the dropdown menu) refers to legacy experiments and should not be used going forward.

Limitations (cont.)

- Tag/Metrics/Parameter columns that were previously hidden on the runs table will be remembered, but CML won't remember hiding any of the other columns (date, version, user, etc.)
- Admins can not browse all experiments. They can only see their experiments on the global Experiment page.
- Performance issues may arise when browsing the run details of a run with a lot of metric results, or when comparing a lot of runs.
- Runs can not be deleted or archived.

Chapter Topics

Experiments

- Experiments in CML
- **Essential Points**
- Hands-On Exercise: Running an Experiment

Essential Points

- CML uses MLflow for Experiments
- Experiments tracks parameters, code versions, metrics, and output files.
- Only Python is supported

Chapter Topics

Experiments

- Experiments in CML
- Essential Points
- **Hands-On Exercise: Running an Experiment**

Hands-On Exercise: Running an Experiment

- **In this exercise, you will**
 - Use Git repository to create your project
 - Track an experiment using the CML experiments feature
- **Please refer to the Hands-On Exercise Manual for instructions**



Spark Overview

Spark Overview

- **By the end of this chapter, you will learn**
 - How Apache Spark works and what capabilities it offers
 - Which popular file formats Spark can use for data storage
 - Which programming languages you can use to work with Spark
 - How to get started using PySpark
 - How a Spark job is made up of a sequence of transformations followed by an action
 - How Spark uses lazy execution
 - How Spark splits input data into partitions
 - How Spark executes narrow and wide operations
 - How Spark executes a job in tasks and stages

Chapter Topics

Spark Overview

- **How Spark Works**
- The Spark Stack
- File Formats in Spark
- Spark Interface Languages
- Introduction to PySpark
- How DataFrame Operations Become Spark Jobs
- How Spark Executes a Job
- Essential Points
- Demonstration and Exercises: Using PySpark to Prepare Data for ML

Apache Spark

- **Spark is a fast, general-purpose, large-scale data processing engine**
- **Spark can run a wide range of different data processing workloads**
 - Includes several different libraries and APIs
- **Spark can run on several types of clusters**
 - Apache Hadoop YARN
 - Apache Mesos
 - Spark Standalone
 - Kubernetes
- **Spark can also run locally instead of on a cluster**



How Spark Works

- **When a Spark application starts, it launches:**
 - A master process called the *driver* that manages the application
 - Multiple worker processes called *executors* that process data
 - On a YARN cluster, an Application Master process starts to manage the application's executors
 - Under YARN's *dynamic resource allocation*, executors can be launched and terminated during the life of an application as demand for work grows and shrinks
- **When the Spark application stops, these processes are terminated**

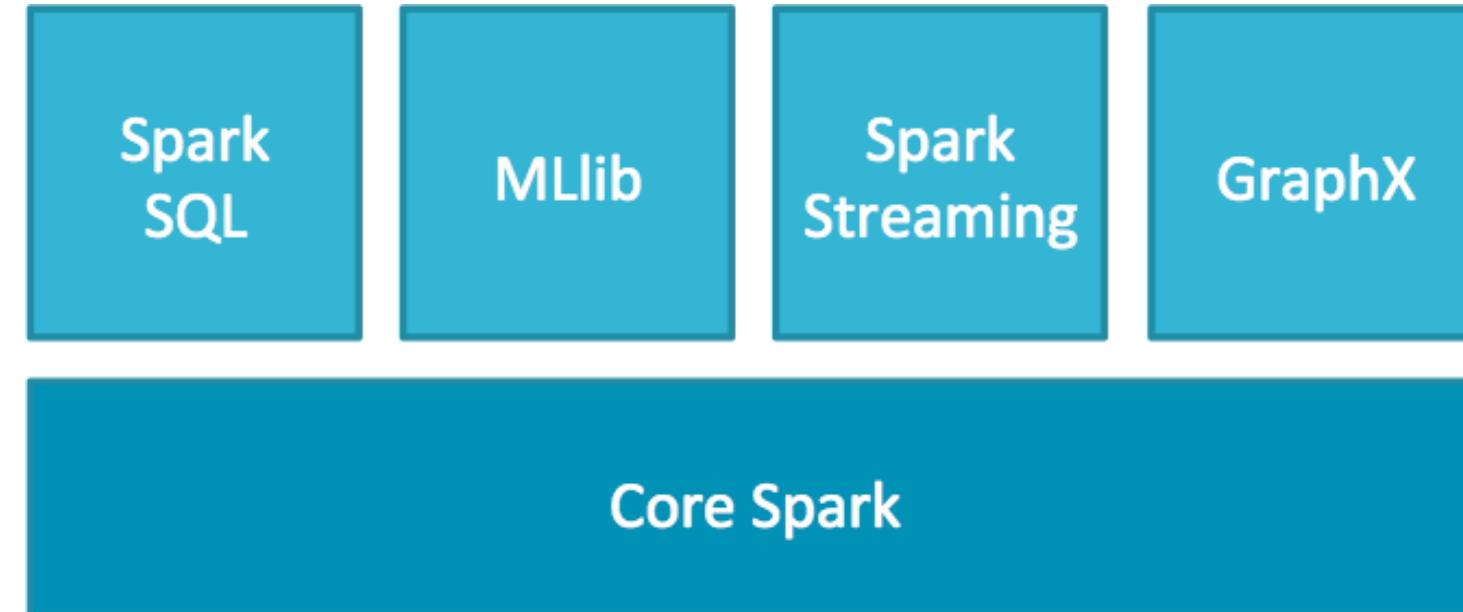
Chapter Topics

Spark Overview

- How Spark Works
- **The Spark Stack**
- File Formats in Spark
- Spark Interface Languages
- Introduction to PySpark
- How DataFrame Operations Become Spark Jobs
- How Spark Executes a Job
- Essential Points
- Demonstration and Exercises: Using PySpark to Prepare Data for ML

The Spark Stack

- **Spark provides a stack of libraries built on core Spark:**
 - Spark SQL works with structured data
 - Spark MLlib supports scalable machine learning
 - Spark Streaming applications process data in real time
 - GraphX works with graphs and graph-parallel computation
- **This course covers the Spark libraries that are most relevant to data scientists**
 - Spark SQL
 - Spark MLlib



Spark SQL

- **Spark SQL is a Spark library for working with structured data**
- **Spark SQL provides:**
 - The DataFrame API
 - A SQL query engine
 - Catalyst optimizer-a n extensible query optimization framework
- **Spark versions before 2.0 did not include the full Spark SQL functionality as presented here**
 - This course covers Spark 3.0 and later

DataFrames

- **DataFrames are the primary representation of data in Spark SQL**
- **DataFrames represent structured data in a tabular form**
 - They model data similar to tables in a relational database
 - They consist of a collection of loosely typed row objects
 - Row objects are organized into columns described by a schema
- **DataFrames are immutable**
 - You cannot modify a DataFrame in place
 - You can *transform* a DataFrame, producing a new DataFrame

Chapter Topics

Spark Overview

- How Spark Works
- The Spark Stack
- **File Formats in Spark**
- Spark Interface Languages
- Introduction to PySpark
- How DataFrame Operations Become Spark Jobs
- How Spark Executes a Job
- Essential Points
- Demonstration and Exercises: Using PySpark to Prepare Data for ML

File Formats in Apache Spark

- **Spark supports several different file formats for data storage**
 - You can convert between formats as needed
- **Selecting the best format for a dataset involves several considerations**
 - How the dataset is created or ingested
 - Size and performance requirements
 - Which other tools are used to work with the dataset
- **Two popular file formats supported by Spark and many other tools are**
 - Text
 - Parquet

Text File Formats

- **Text files are the most basic file type**
 - Can be read or written from virtually any programming language
 - Are compatible with many applications
- **Common examples of text file formats include**
 - Comma-separated values (CSV)
 - Tab-separated values (TSV)
 - JSON
- **Text files are human-readable**
 - All values are represented as strings
 - Useful when debugging
- **Text files are inefficient at large scale**
 - Representing numeric values as strings wastes storage space
 - Conversion to and from native types adds performance penalty

Parquet File Format

- **Apache Parquet is an open source columnar file format**
 - Originally developed by engineers at Cloudera and Twitter
 - Now an Apache Software Foundation project
 - Supported by Spark, Hive, Impala, and other tools
 - Schema information is embedded in the file
- **Uses advanced optimizations**
 - To reduce storage space
 - To increase performance
- **Most efficient when adding many records at once**
 - Some optimizations rely on identifying repeated patterns

Chapter Topics

Spark Overview

- How Spark Works
- The Spark Stack
- File Formats in Spark
- **Spark Interface Languages**
- Introduction to PySpark
- How DataFrame Operations Become Spark Jobs
- How Spark Executes a Job
- Essential Points
- Demonstration and Exercises: Using PySpark to Prepare Data for ML

Spark Interface Languages

- **Spark provides APIs for four programming languages**
 - Scala
 - Java
 - Python (PySpark)
 - R (SparkR)
- **Of these, SparkR is the least mature**
- **sparklyr is an alternative R interface to Spark**
 - Developed by Rstudio
 - Compatible with the popular R package dplyr
- **This course covers**
 - Python: PySpark

Chapter Topics

Spark Overview

- How Spark Works
- The Spark Stack
- File Formats in Spark
- Spark Interface Languages
- **Introduction to PySpark**
- How DataFrame Operations Become Spark Jobs
- How Spark Executes a Job
- Essential Points
- Demonstration and Exercises: Using PySpark to Prepare Data for ML

PySpark

- **PySpark is Spark's Python API**
 - Developed and distributed as a part of the Apache Spark project
 - Provides access to the full Spark API
- **PySpark includes an interactive shell application**
 - Start it from the operating system shell using the command `pyspark3`
- **You can also use PySpark with other Python applications and scripts**
 - By using the `pyspark` Python package
- **See documentation and examples on the Apache Spark website**
 - <https://spark.apache.org>

Data Science with PySpark

- **PySpark exposes the Spark SQL library for working with DataFrames**
 - Includes methods for loading and saving DataFrames
 - Includes methods for transforming DataFrames, such as
 - `select()` to select one or more columns
 - `filter()` or `where()` to filter rows by some condition
 - `orderBy()` or `sort()` to sort rows
 - `withColumn()` to add or replace columns
 - `agg()` to compute aggregates
 - `groupBy()` to define groups of rows
- **PySpark exposes the Spark MLlib library for machine learning, which provides**
 - Feature transformers
 - Machine learning algorithms
- **PySpark integrates with the SciPy ecosystem of tools, including pandas**

Chapter Topics

Spark Overview

- How Spark Works
- The Spark Stack
- File Formats in Spark
- Spark Interface Languages
- Introduction to PySpark
- **How DataFrame Operations Become Spark Jobs**
- How Spark Executes a Job
- Essential Points
- Demonstration and Exercises: Using PySpark to Prepare Data for ML

DataFrame Operations

- **There are two main types of DataFrame operations**
 1. *Transformations* create a new DataFrame based on existing one(s)
 - Transformations are executed by the Spark application's executors
 2. *Actions* output data values from the DataFrame
 - Output is typically returned from the executors to the Spark driver or saved to a file
- **A sequence of transformations followed by an action is a *job***
 - Transformations in a job can be chained together

Enable Spark Job (1)

- Consider this example Spark job:

```
1 a = spark.read.csv(...)  
2 b = a.select(...)  
3 c = b.filter(...)  
4 d = c.groupBy(...)  
5 e = d.filter(...)  
6 f = e.select(...)  
7 g = f.withColumn(...)  
8 h = g.orderBy(...)  
9 h.write.csv(...)
```

- The operations on Lines 1-8 are all ***transformations***
 - Each one creates a new Spark DataFrame
- The operation on Line 9 is an ***action***
 - It does *not* create a new Spark DataFrame
 - It outputs a result

- Only ***actions*** cause Spark to perform work

- In this example, Spark does no data processing until Line 9
- This is *lazy execution*; no work is performed until the result is needed

Enable Spark Job (2)

- The example on the previous slide could be rewritten:

```
1 spark.read.csv(...)\n2   .select(...)\n3   .filter(...)\n4   .groupBy(...)\n5   .filter(...)\n6   .select(...)\n7   .withColumn(...)\n8   .orderBy(...)\n9   .write.csv(...)
```

- This demonstrates how a sequence of transformations can be chained together
 - It is not necessary to assign intermediate DataFrames to variables

- But you may choose to assign intermediate DataFrames to variables
 - To allow reuse
 - To control caching
 - To improve code clarity

Enable Spark Job (3)

- The example on the previous slides could be rewritten in two jobs:

```
1 h = spark.read.csv(...)\n2   .select(...)\n3   .filter(...)\n4   .groupBy(...)\n5   .filter(...)\n6   .select(...)\n7   .withColumn(...)\n8   .orderBy(...)\n9 h.persist(...)\n10 h.write.csv(...)\n11 h.collect(...)
```

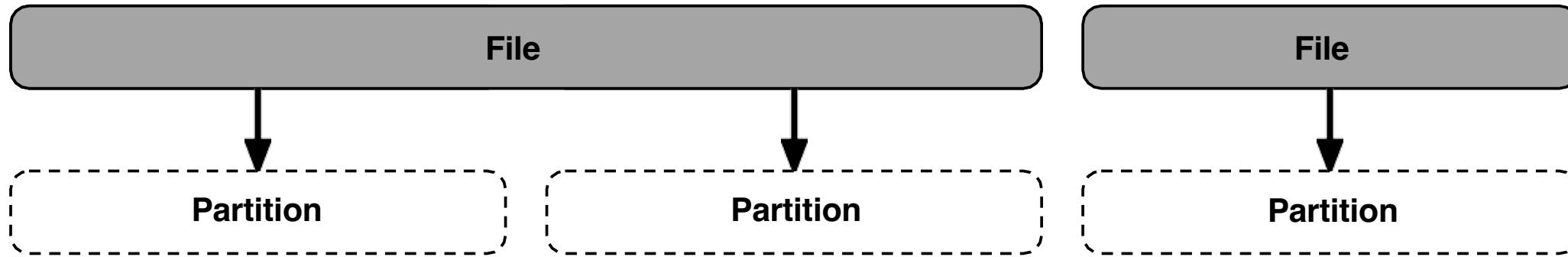
- Note that Lines 10 and 11 perform separate actions**
 - As a result, this code executes two separate Spark jobs

Chapter Topics

Spark Overview

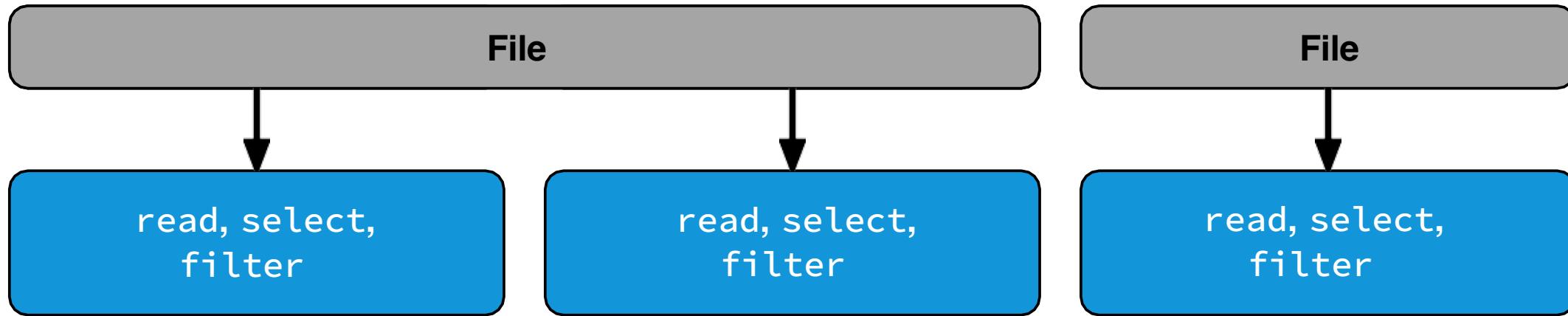
- How Spark Works
- The Spark Stack
- File Formats in Spark
- Spark Interface Languages
- Introduction to PySpark
- How DataFrame Operations Become Spark Jobs
- **How Spark Executes a Job**
- Essential Points
- Demonstration and Exercises: Using PySpark to Prepare Data for ML

Input Splits



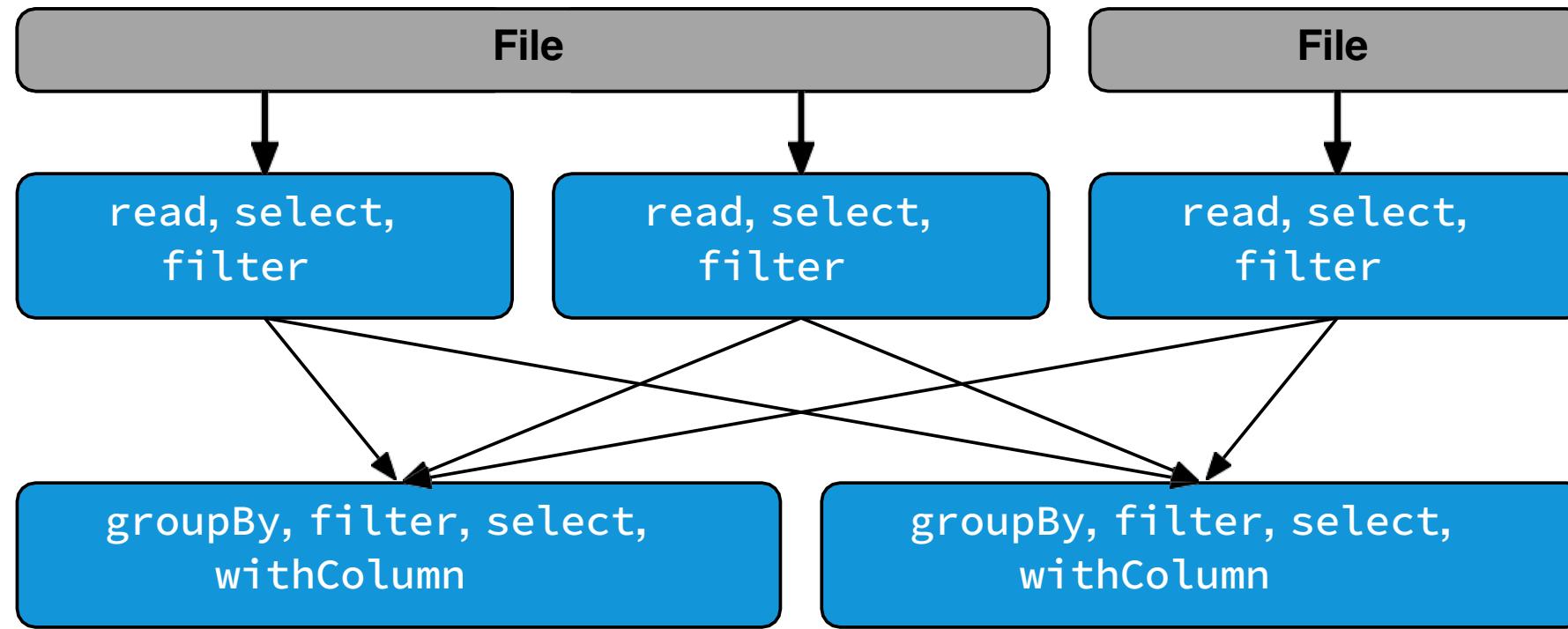
- **Read operations define the input to a Spark job**
 - At runtime, Spark computes *input splits* on the data
 - These correspond to *partitions* in the first DataFrame
- **Spark's Catalyst optimizer automatically chooses input splits**
 - Based on the number and size of files
- **Spark also allows manual control of input splits**

Narrow Operations



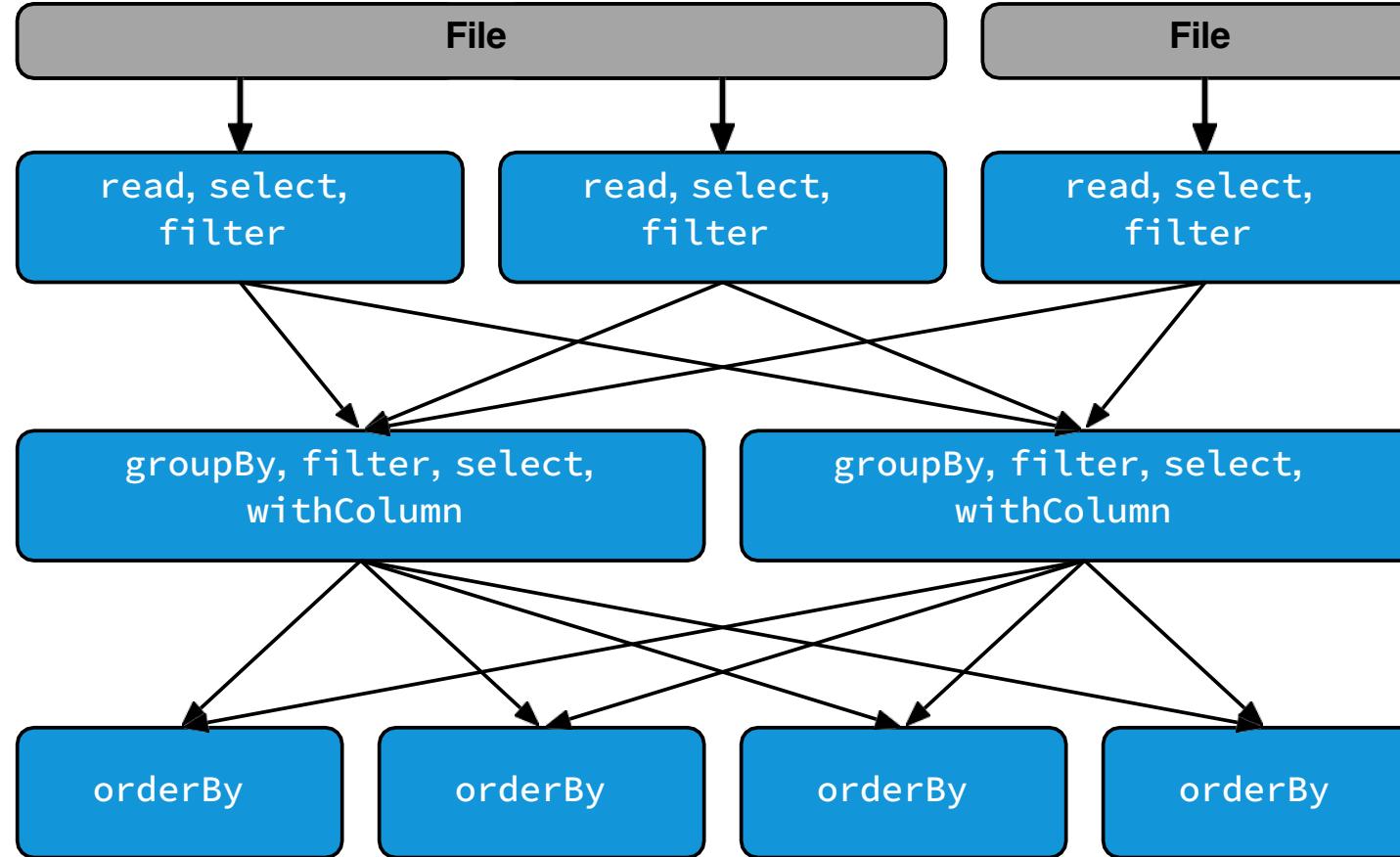
- **Some Spark transformations are *narrow***
 - They essentially affect only one record at a time
- **Spark combines sequences of narrow operations**
 - In this example, Spark has combined **read, select, and filter**

Wide Operations



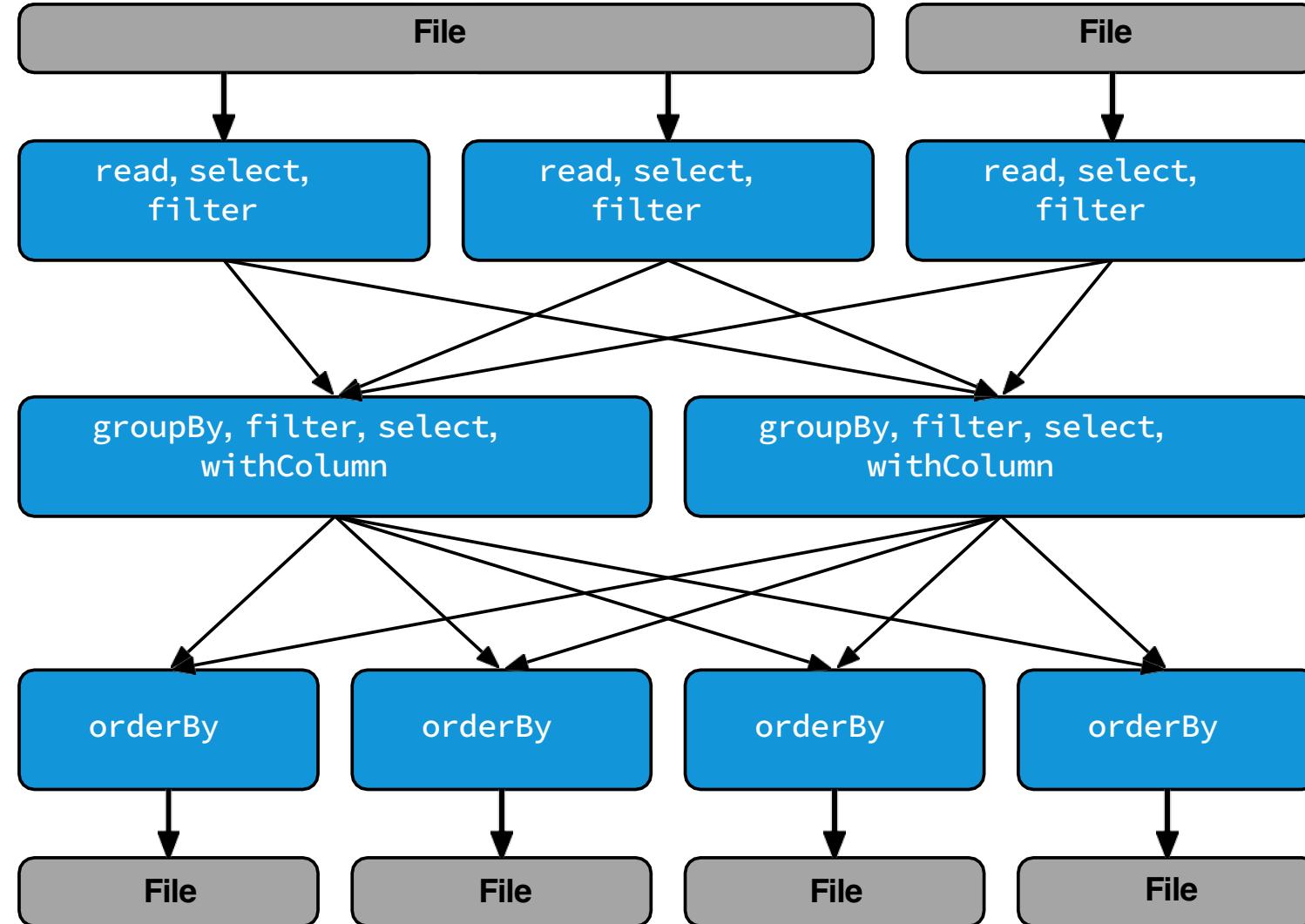
- **Some Spark operations are wide**
 - They force redistribution of records
- **In wide operations, the number of partitions may change**
- **Subsequent narrow operations are combined with the wide operation**
 - In this example, Spark has combined the subsequent filter, select, and withColumn (narrow operations) with groupBy (a wide operation)

Stages and Tasks



- **Each new wide operation in a job forces a new *stage***
- **In each stage, a task is performed on each partition**
 - A task can perform a single operation or a combined sequence of operations
- **Each task runs in parallel with the other tasks in its stage**

Shuffle



- Wide operations force Spark to perform a *shuffle*
 - An expensive but necessary redistribution of records into new partitions

Chapter Topics

Spark Overview

- How Spark Works
- The Spark Stack
- File Formats in Spark
- Spark Interface Languages
- Introduction to PySpark
- How DataFrame Operations Become Spark Jobs
- How Spark Executes a Job
- **Essential Points**
- Demonstration and Exercises: Using PySpark to Prepare Data for ML

Essential Points

- When a Spark application starts, it launches a driver and executors
- Spark SQL works on structured data
- DataFrames are the primary representation of structured data in a tabular form
- You cannot modify a DataFrame in place, however you can *transform* a DataFrame
- PySpark exposes
 - The Spark SQL library for working with DataFrames
 - The Spark MLlib library for machine learning
- A job is a sequence of transformations followed by an action

Chapter Topics

Spark Overview

- How Spark Works
- The Spark Stack
- File Formats in Spark
- Spark Interface Languages
- Introduction to PySpark
- How DataFrame Operations Become Spark Jobs
- How Spark Executes a Job
- Essential Points
- **Demonstration and Exercises: Using PySpark to Prepare Data from ML**

Demonstration and Exercises: Using PySpark to Prepare Data for ML

- Your instructor will now demonstrate how to use PySpark to prepare data for machine learning



Machine Learning Overview

Machine Learning Overview

- **By the end of this chapter, you will learn**
 - What machine learning is
 - Some important terms and concepts in machine learning
 - Different types of machine learning algorithms
 - Which libraries are used for machine learning

Chapter Topics

Machine Learning Overview

- **Introduction to Machine Learning**
- Machine Learning Tools
- Essential Points

A Definition of Machine Learning

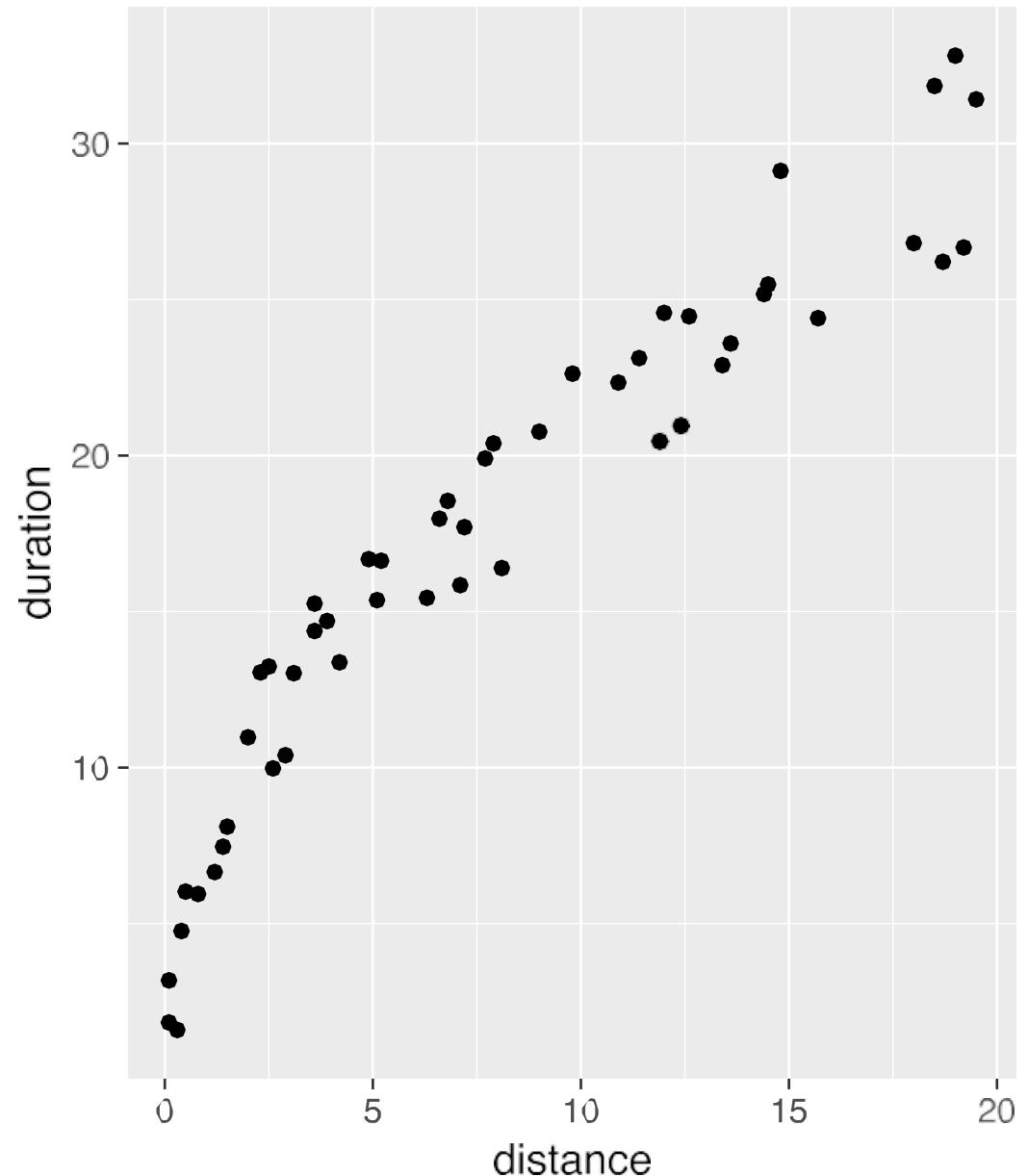
**Machine learning is the use of algorithms
to automatically discover relationships or patterns in
historical data that generalize to future data**

Machine Learning Versus Traditional Programming

- In traditional programming:
 - A programmer creates a program with instructions for the computer
 - The program *contains* the logic mapping inputs to outputs
- In machine learning:
 - A program creates instructions on its own by examining *training data*
 - The program *learns* the logic to map inputs to outputs
 - This learning process is called *training*
 - The result is a *model*
 - The model is applied to new data to make *predictions*

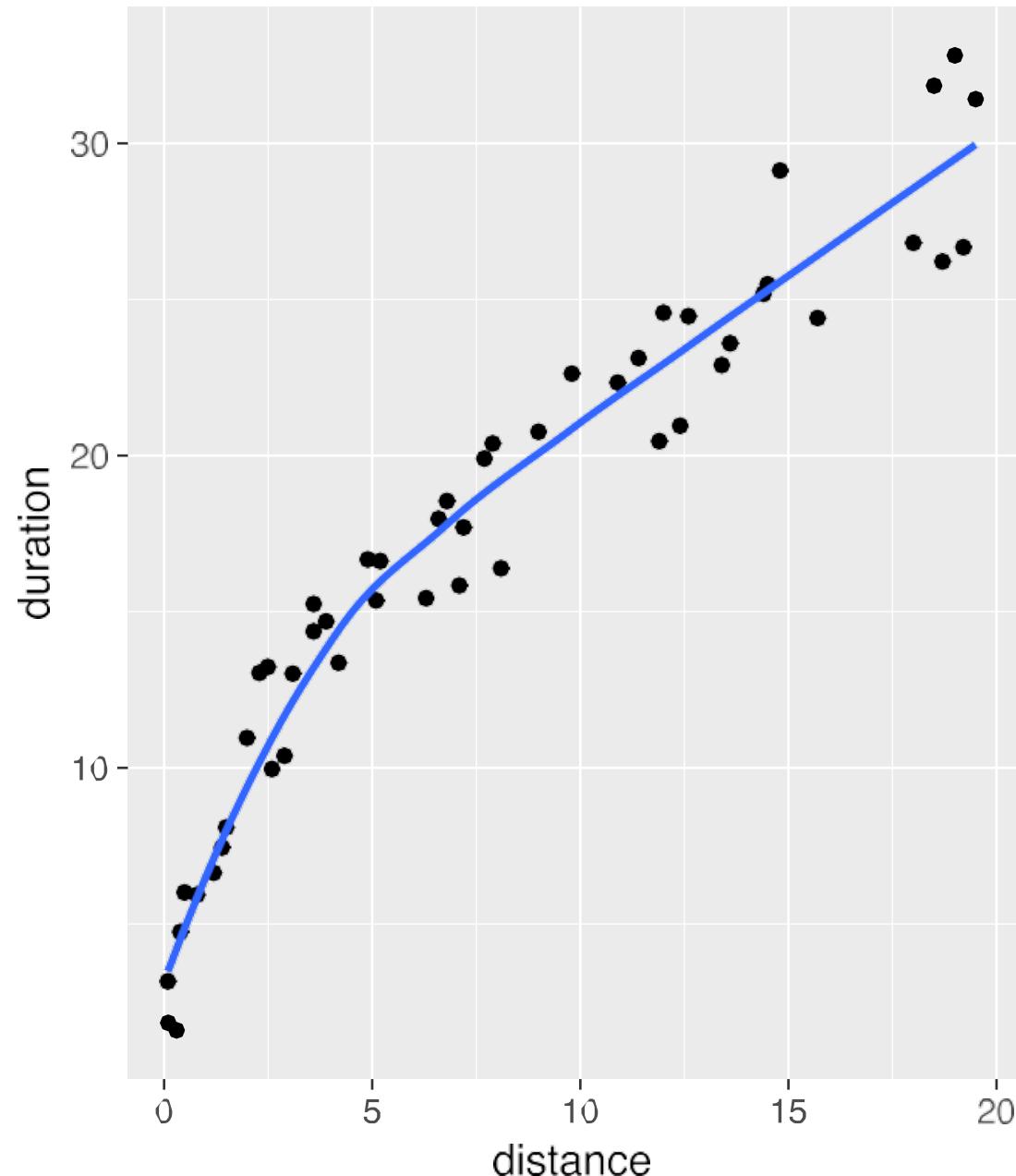
Machine Learning Example (1)

- The scatterplot visualizes a small set of training data with two variables
 - Ride distance (in miles) and duration (in minutes)
 - These two variables are positively associated
- A machine learning model could predict duration based on distance
 - Distance is the *input variable*
 - Also called *predictor variable* or *feature*
 - A model can have more than one input variable
 - Duration is the *output variable*
 - Also called *response variable* or *label*
 - A model typically has only one output variable



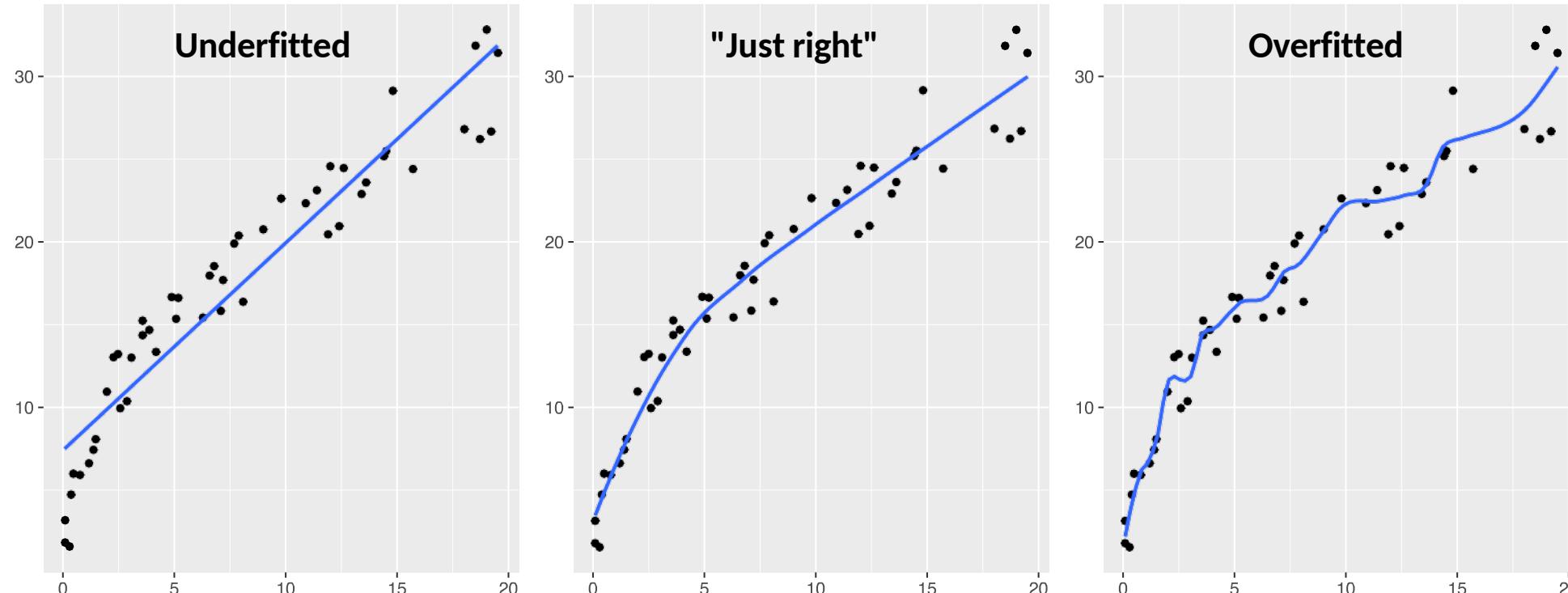
Machine Learning Example (2)

- The overlaid line visualizes a machine learning model
 - In this example, a local polynomial regression model
 - Shows a steeper, concave relationship for shorter rides
 - Shows a linear relationship for longer rides
- Given a value for ride distance, the model predicts ride duration
 - The prediction will include some random error
 - Due to expected variation
 - The prediction should be free of systematic error
 - If not, the model is *biased*



Underfitting and Overfitting

- A useful machine learning model should explain a large proportion of the variation of the output variable
 - Underfitting is when a model does not explain enough of the systematic variation in the training data
- A model can also explain too much variation
 - Overfitting is when a model explains *random* variation in the training data



Model Validation

- **How can you evaluate whether a model is underfitting or overfitting?**

- Validate the model using a separate dataset

- Use the model to generate predictions on separate data

- Typically a random sample held out from the training data

- Compare the predictions to the actual values of the output variable

- Compute measures such as R^2 , the proportion of the variance in the data that is explained by the model

- **Validation helps you select a model that explains as much systematic variation as possible**

- And does not explain random variation in the training data

Hyperparameters

- **How can you control whether a model is underfitting or overfitting?**
 - By choosing appropriate values of *hyperparameters*
- **Hyperparameters are algorithm parameters specified by the user before training a model**
 - In the previous example, the smoothness of the local polynomial regression curve is specified as a hyperparameter
 - Larger values yield smoother curves
 - Smaller values yield wigglier curves
 - Hyperparameters are passed as arguments to machine learning functions
- **The process of choosing suitable hyperparameters is called *tuning* a model**
 - Some machine learning libraries include tools to automate this process

Supervised and Unsupervised Learning

- There are two main types of machine learning

Supervised learning

Used when the training data includes an output variable

- In other words, when the training data is *labeled*

Predicts values of the output variable based on values of the input variable(s)

Unsupervised learning

Used when the training data does not include an output variable

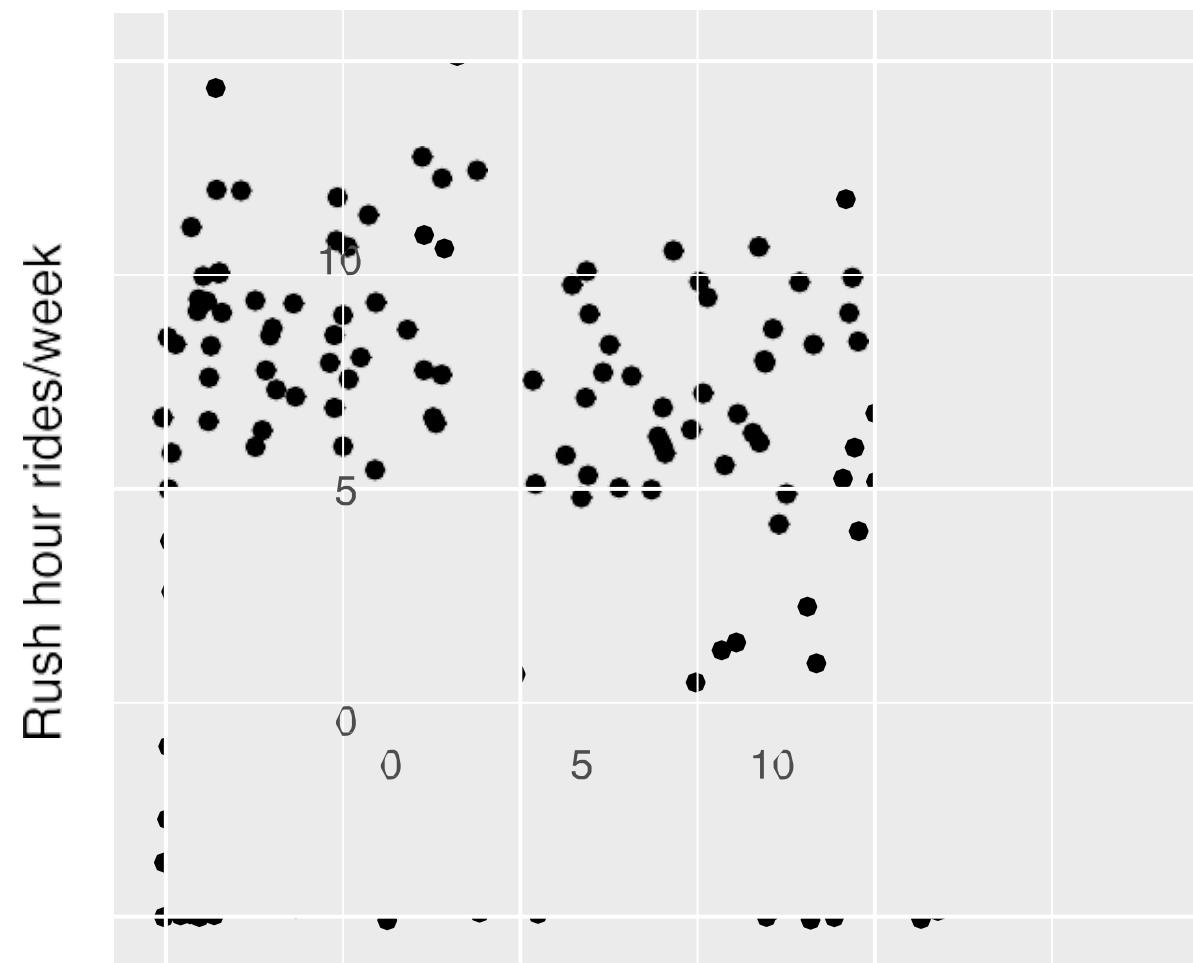
- In other words, when the training data is *unlabeled*

Describes structure or patterns in the data

- The previous example demonstrated supervised learning
- The next example demonstrates unsupervised learning

Unsupervised Learning Example (1)

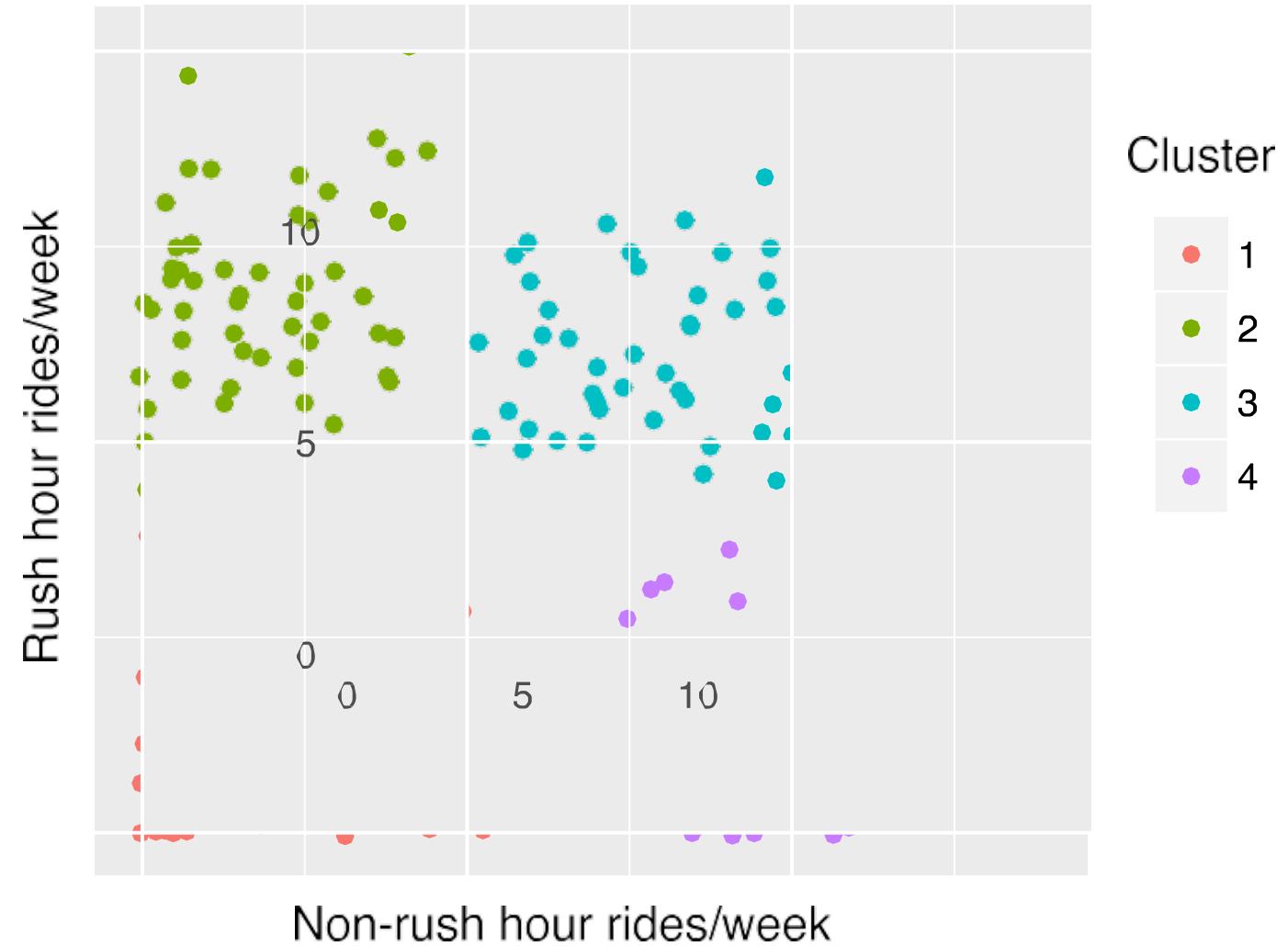
- The scatterplot visualizes a small set of training data with two variables
 - Each point represents one rider
 - On the vertical axis: Average number of rides per week during rush hour periods
 - On the horizontal axis: Average number of rides per week outside of rush hour periods
- This data could be partitioned into clusters to identify different categories of riders
 - But the training data does not include an output variable giving the category of each rider
 - It is *unlabeled*
 - Only an *unsupervised* learning algorithm can be used



Non-rush hour rides/week

Unsupervised Learning Example (2)

- ***k-means clustering*** is a popular unsupervised learning algorithm
- A ***k-means clustering model*** is fitted to the training data
 - To partition the data into four clusters
 - The number of clusters is a *hyperparameter*
 - The clusters represent different categories of riders
 1. Infrequent riders
 2. Commuters
 3. Frequent riders
 4. Non-commuters



Chapter Topics

Machine Learning Overview

- Introduction to Machine Learning
- **Machine Learning Tools**
- Essential Points

Machine Learning Algorithms

- The examples in the previous section demonstrated

- A supervised learning algorithm (local polynomial regression)
- An unsupervised learning algorithm (k -means clustering)

- There are many other machine learning algorithms

Supervised learning algorithms

For a continuous output variable (*regression* problems)

[Linear regression](#)

[Generalized linear regression](#)

For a binary or categorical output variable (*classification* problems)

[Logistic regression](#)

[Naive Bayes classifiers](#)

For either type of problem

[Decision trees](#)

[Random forest](#)

[Survival regression](#)

[Isotonic regression](#)

[Support vector machines](#)

[Linear discriminant analysis](#)

[Gradient-boosted trees](#)

[Neural networks](#)

Unsupervised learning algorithms

[Gaussian mixture models](#)

[Latent Dirichlet allocation](#)

Machine Learning Libraries

- Machine learning algorithms are implemented in libraries and packages for Python and R
 - For Python: The **scikit-learn** library
 - For R: Many different packages, unified by the **tidymodels** packages
- Apache Spark provides Mllib
 - A scalable machine learning library
 - Can be used with Java, Scala, Python (PySpark), and R (SparkR and sparklyr)
- Many other libraries focus on neural networks and deep learning

These include

Apache MXNet	Keras
BigDL	PyTorch
Caffe	TensorFlow
CNTK	Theano
DL4J	Torch

Coverage of deep learning is beyond the scope of this course

Chapter Topics

Machine Learning Overview

- Introduction to Machine Learning
- Machine Learning Tools
- **Essential Points**

Essential Points

- In machine learning, a program creates instructions on its own by examining the training data
- A useful machine learning model should explain a large proportion of the variation of the output variable
 - Tuning a model is the process of choosing suitable hyperparameters
 - Allows you to control whether a model is underfitting or overfitting
- Types of machine learning include:
 - Supervised learning – includes an output variable
 - Unsupervised learning – does not include an output variable
- There are many machine learning algorithms for you to use in a model

CLOUDERA

Apache Spark MLlib

Apache Spark MLLib

- **By the end of this chapter, you will learn**
 - Which machine learning capabilities MLLib provides
 - How to build, validate, and use machine learning models with MLLib

Chapter Topics

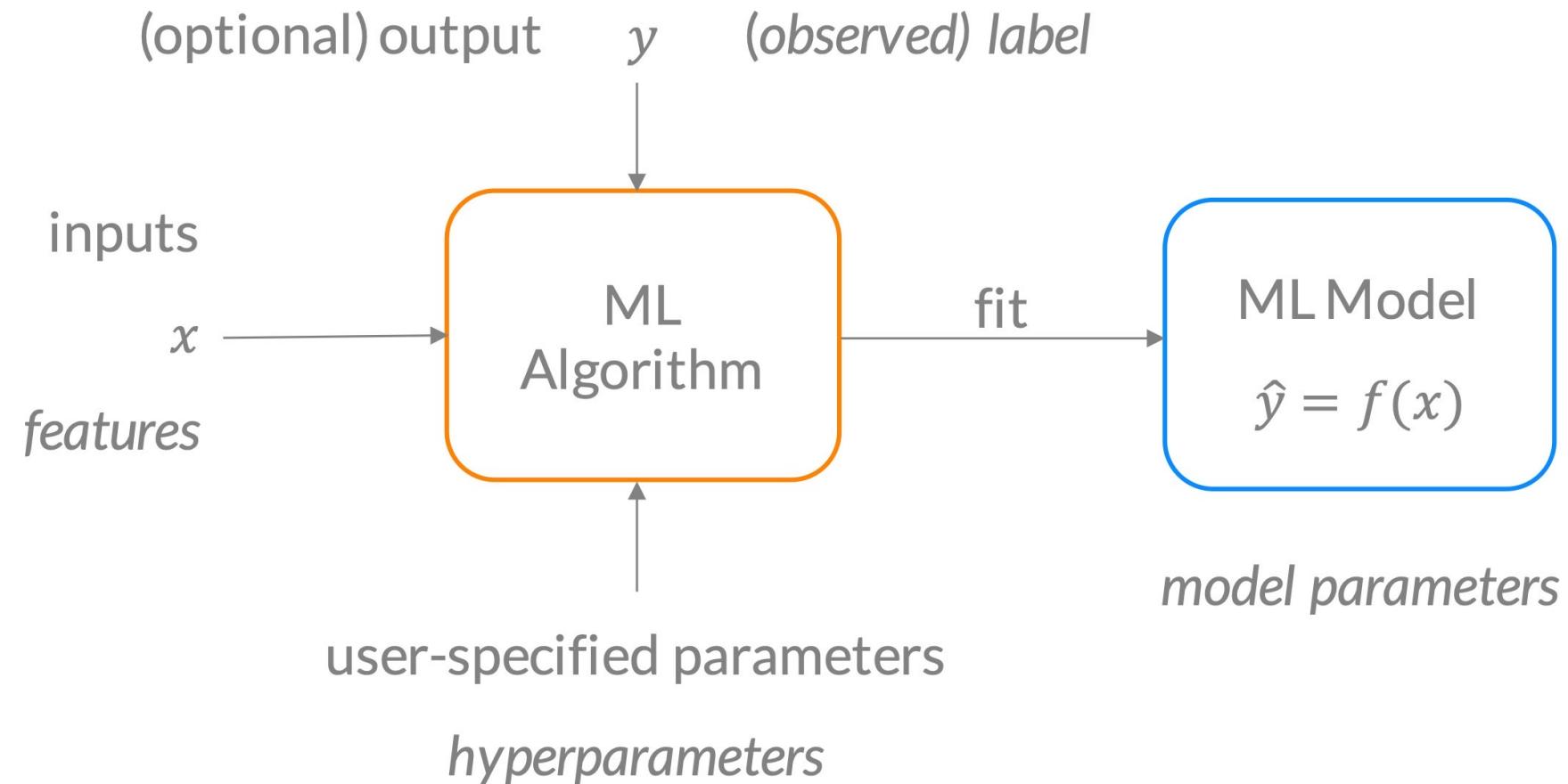
Apache Spark MLlib

- **Introduction to Apache Spark MLlib**
- Essential Points
- Demonstrations and Exercises: Using MLlib

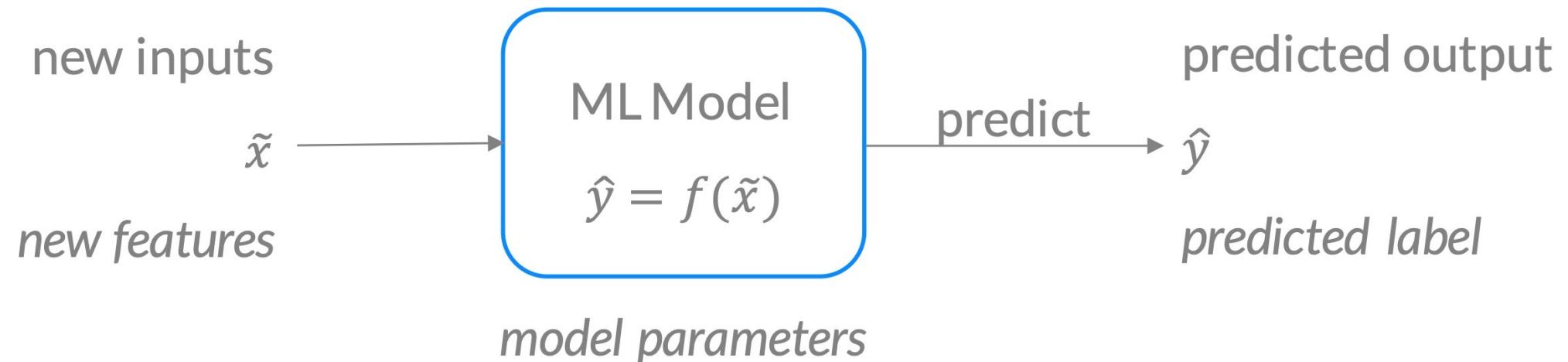
Apache Spark MLLib

- **MLlib is Spark's scalable machine learning library**
 - Designed to work on very large datasets
 - Interoperates with Spark SQL and the DataFrames API
- **MLlib includes implementations of many popular machine learning algorithms, such as**
 - Supervised learning algorithms for regression and classification problems
 - Unsupervised learning algorithms
- **MLlib provides utilities to prepare data before building machine learning models**
 - *Feature extractors* to extract features from data
 - *Feature transformers* to process data into a form suitable for modeling
 - *Feature selectors* to select a subset of features
- **MLlib also includes machine learning workflow utilities, for**
 - Tuning hyperparameters
 - Creating pipelines

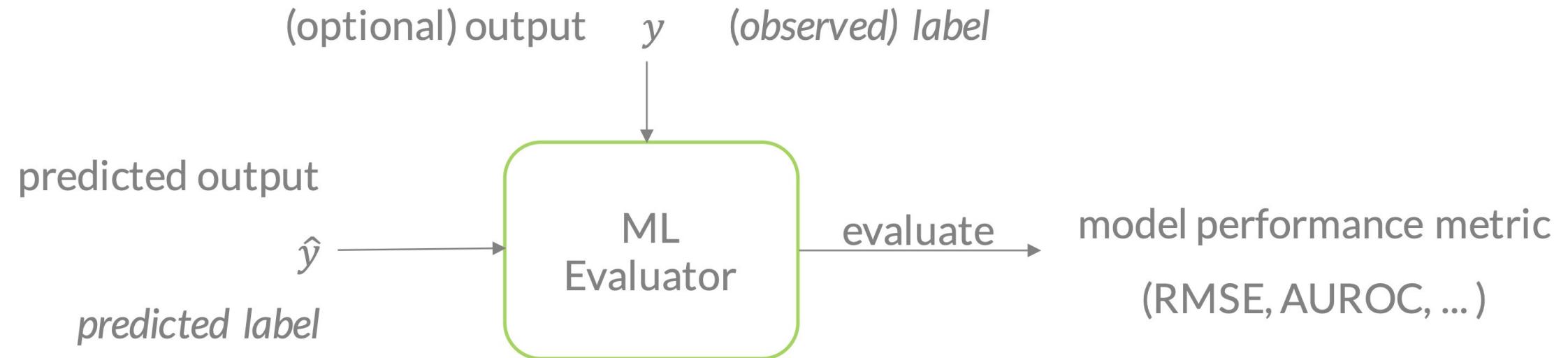
Model Fitting Phase



Model Prediction Phase



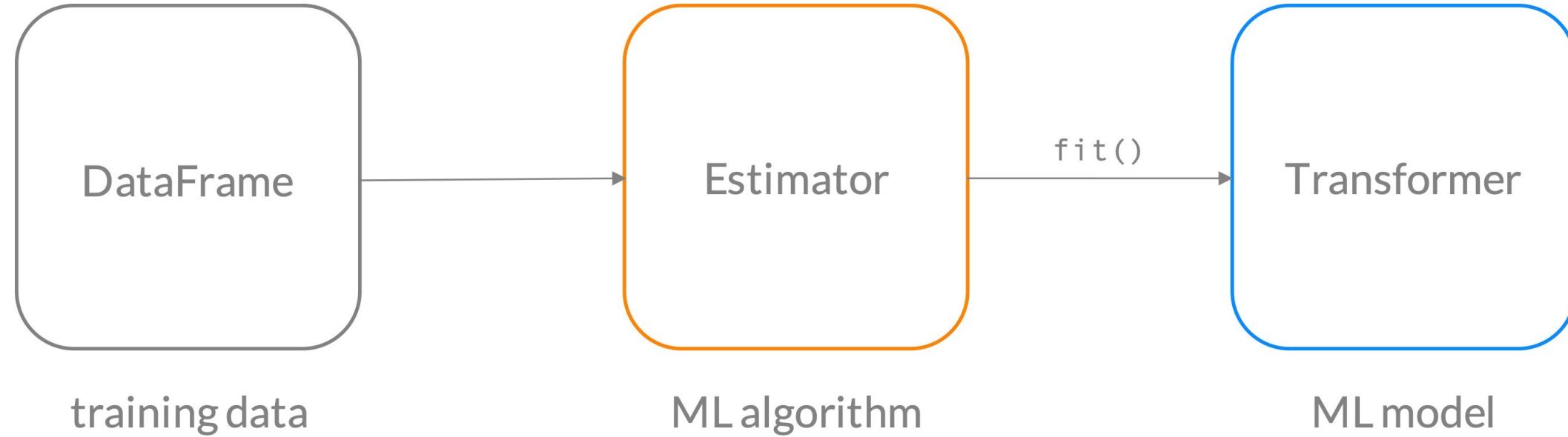
Model Evaluation Phase



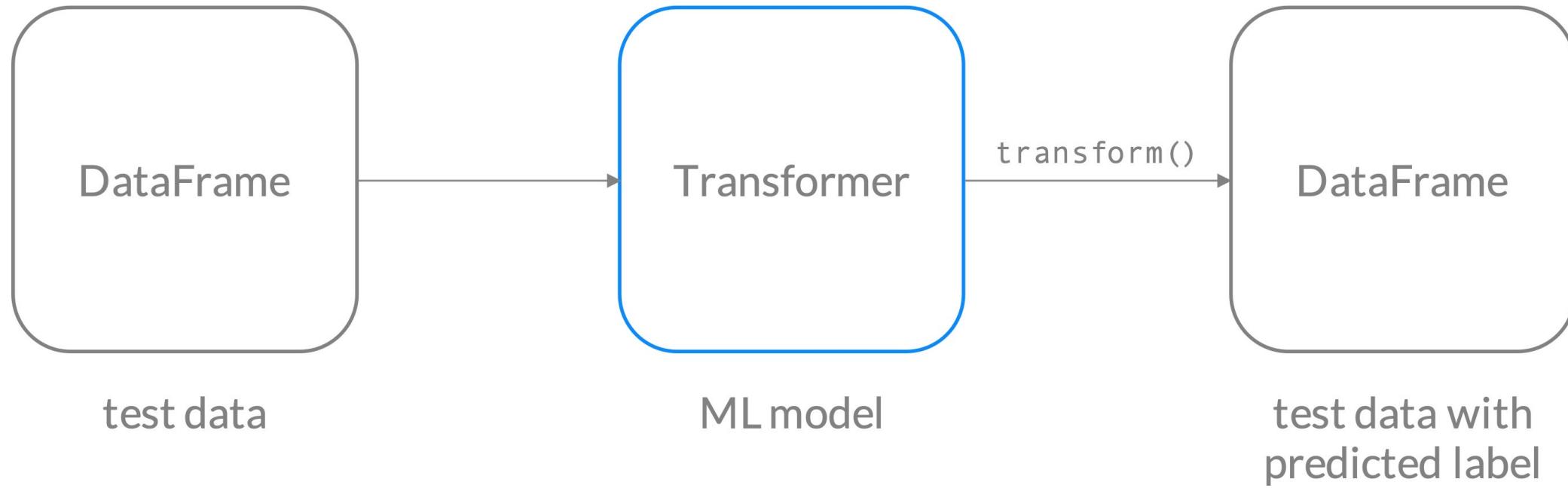
Primary Abstractions in MLlib

- ***Estimator*** class
 - Typically represents a machine learning algorithm
 - For example, `LinearRegression()`
- ***Transformer*** class
 - Typically represents a machine learning model
 - For example, `LinearRegressionModel()`
- ***Evaluator*** class
 - Represents a collection of model performance metrics
 - For example, `RegressionEvaluator()`

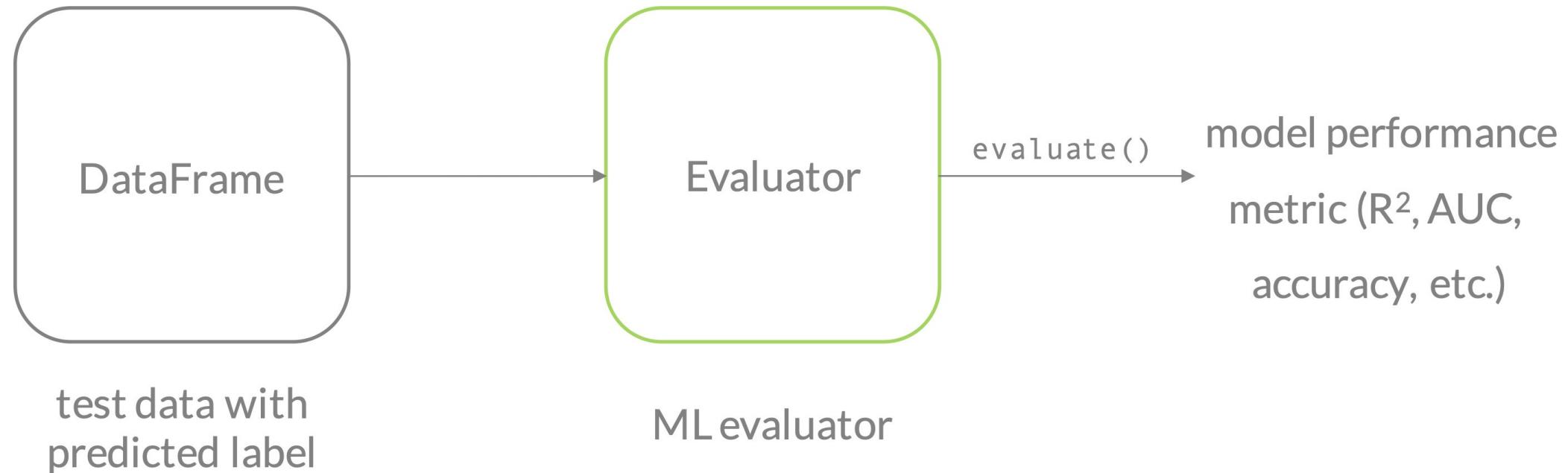
Estimator Class



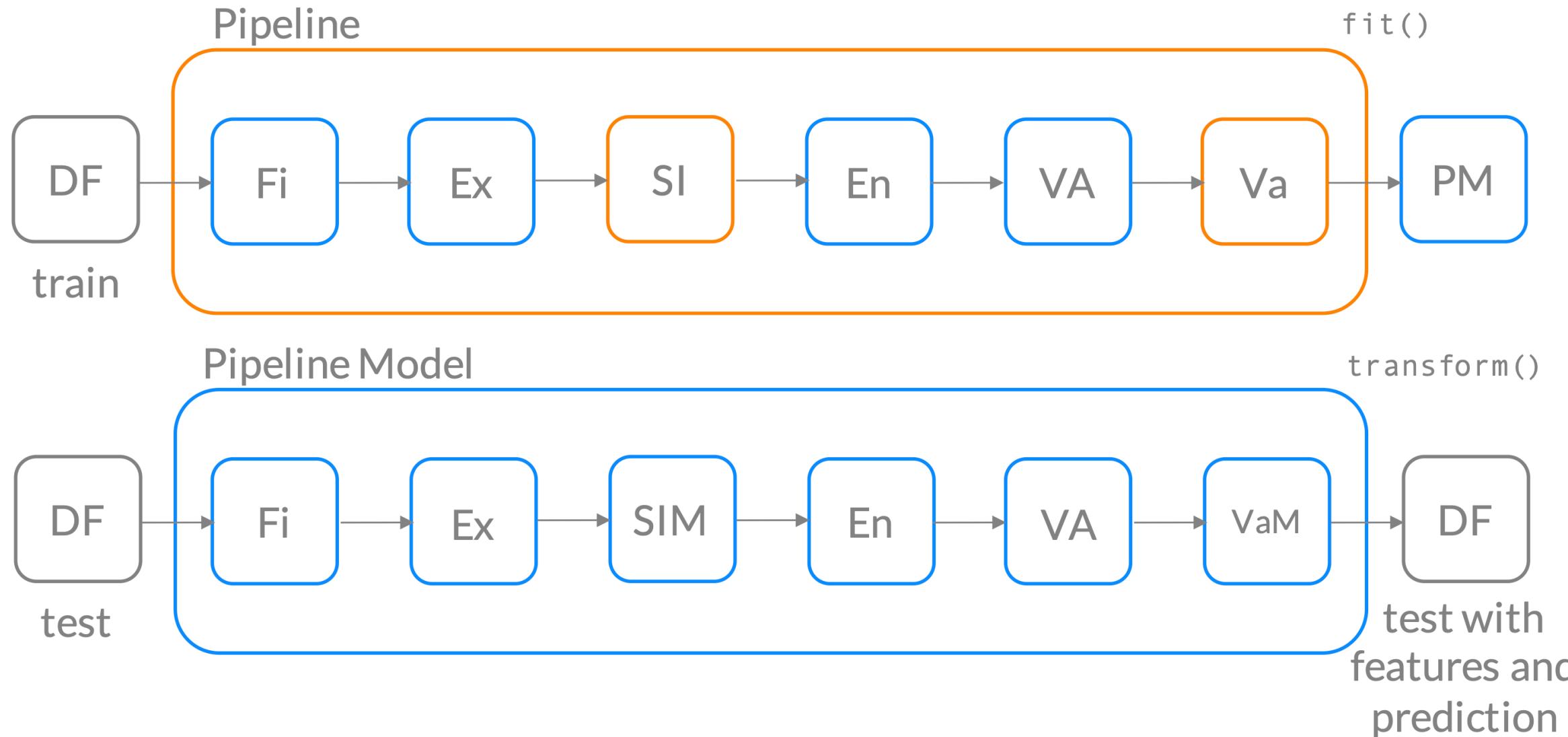
Transformer Class



Evaluator Class



MLlib Pipeline Example



Chapter Topics

Apache Spark MLlib

- Introduction to Apache Spark MLlib
- **Essential Points**
- Demonstrations and Exercises: Using MLlib

Essential Points

- **MLlib includes implementations of both supervised and unsupervised algorithms**
- **MLlib utilities to prepare data include:**
 - Extractors
 - Transformers
 - Selectors
- **MLlib workflow utilities include:**
 - Tuning hyperparameters
 - Creating pipelines

Chapter Topics

Apache Spark MLlib

- Introduction to Apache Spark MLlib
- Essential Points
- **Demonstrations and Exercises: Using MLlib**

Demonstrations and Exercises: Using MLlib

- Your instructor will now demonstrate how to use featurization utilities, machine learning algorithms, and workflow utilities in MLlib



Autoscaling, Performance, and GPU Settings

Autoscaling, Performance, and GPU Settings

- **By the end of this chapter, you will be able to**
 - Understand how CML automatically scales up and down
 - Provision a workspace with GPUs
 - Set quotas on GPUs per user
 - Add GPUs to an existing workspace
 - Understand the financial implications of GPUs
 - Add GPUs to a session, experiment or job
 - Write hardware aware code
 - Understand the impact of GPUs on the performance of your workloads

Chapter Topics

Autoscaling, Performance, and GPU Settings

- **Autoscaling Workloads**
- Working with GPUs
- Essential Points
- Hands-On Exercise: Deep Learning with GPUs in CML

Autoscale Groups

- A CML workspace or cluster consists of three different autoscaling groups: “**infra**”, “**cpu**” and “**gpu**”
- These groups scale independently of one another
 - CPU and GPU can be set when provisioning the workspace or when editing the workspace details
 - The Infra autoscaling group is created automatically when a user provisions a CML cluster, and is not configurable from the UI

Workspace Instances							
Name	Instance Type	CPU	GPU	Memory	Count	Autoscale Range Min - Max	
CML CPU Workers	m5.4xlarge	16	-	64 GiB	1	1 - 5	
CML GPU Workers	p2.8xlarge	32	8	488 GiB	0	0 - 3	
CML Infra	m5.2xlarge	8	-	32 GiB	2	2 - 3	
Platform Infra	m5.large	2	-	8 GiB	2	2 - 4	

 [Delete GPU](#)

Scaling Up

- The primary trigger for scaling up (or expanding) an autoscaling group is failure by the Kubernetes pod scheduler to find a node that meets the pod's resource requirements
- In CML, if the scheduler cannot find a node to schedule an engine pod because of insufficient CPU or memory, the engine pod will be in “pending” state
- When the autoscaler notices this situation, it will change the desired capacity of the autoscaling group (CPU or GPU) to provision a new node in the cluster
- As soon as the new node is ready, the scheduler will place the session or engine pod there

Scaling Down

- A cluster is scaled down by the autoscaler by removing a node, when the resource utilization on the given node is less than a predefined threshold, provided the node does not have any non-evictable pods running on it
- This threshold is currently set to **20% CPU utilization** the engine pod will be in “pending” state
- That is, a node is removed if the following criteria are met:
 - ✓ The node does not have non-evictable pods
 - ✓ The node's CPU utilization is less than 20%
 - ✓ The number of active nodes in the autoscaling group is more than the configured minimum capacity



By setting the minimum range of the GPU workers to 0 you will only incur their cost when you use them

Chapter Topics

Autoscaling, Performance, and GPU Settings

- Autoscaling Workloads
- **Working with GPUs**
- Essential Points
- Hands-On Exercise: Deep Learning with GPUs in CML

Provision a Workspace with GPUs

GPU Settings

Instance Type

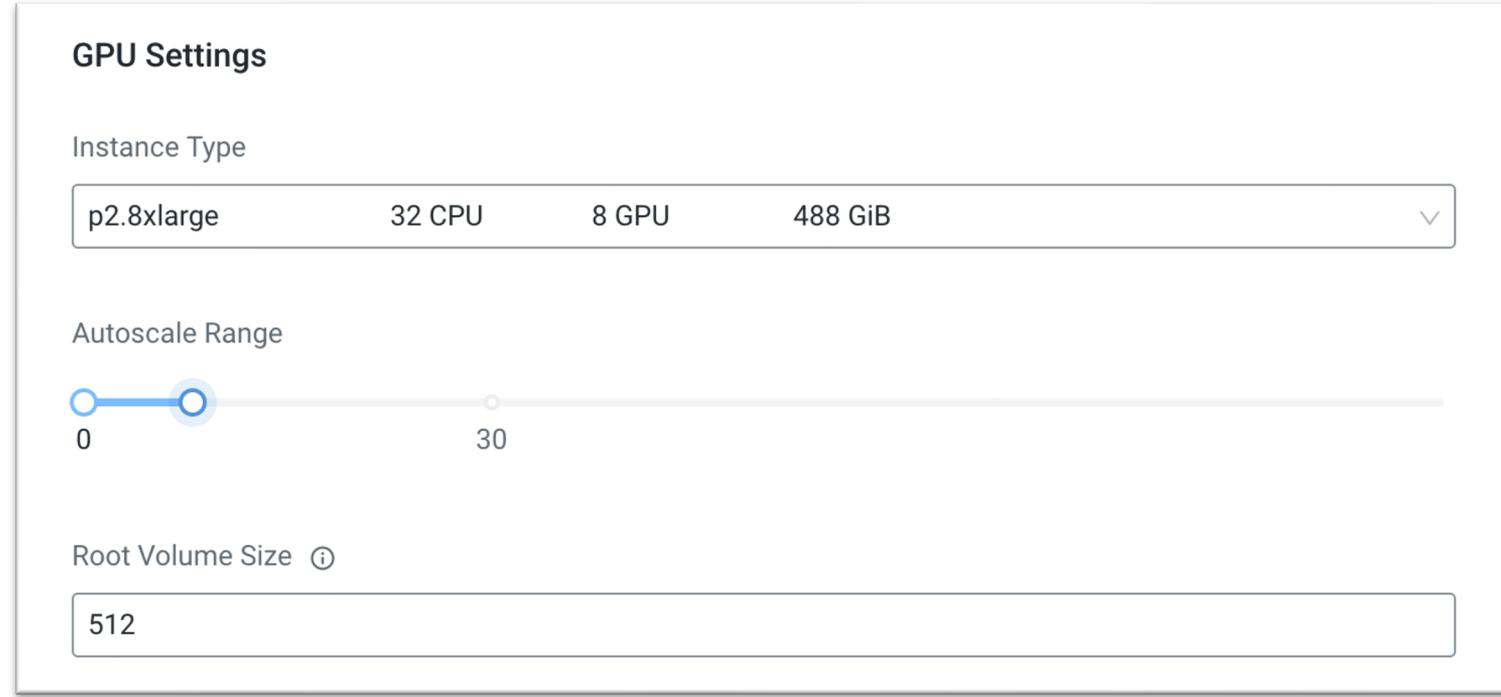
p2.8xlarge 32 CPU 8 GPU 488 GiB

Autoscale Range

0 30

Root Volume Size ⓘ

512



- The cloud makes provisioning GPU enabled machines frictionless
- With the **MLAdmin** role, switch on the **Advanced Options** in the **Provision Machine Learning Workspace** form: the **GPU Settings** section will appear
 - Choose your Instance Type
 - Specify the Autoscale Range
 - Set the Root Volume Size

Set Quotas on GPUs Per User

- As an **MLAdmin** you can set quotas per user on GPUs similar to any other resource in the Site Administration page

The screenshot shows the Cloudera Site Administration interface. The top navigation bar includes links for Overview, Users, Teams, Usage, **Quotas**, Models, Runtime/Engine, Data Connections, Security, AMPs, Settings, and Support. The Quotas section is active, indicated by a blue underline. A toggle switch labeled "ON" is followed by a question mark icon and the text "Default Quota". Below this, there are two sections: "Quotas" and "Default (per user)". Under "Default (per user)", it shows "4 Users on this cluster" with resource allocations: CPU (vCPU) 2, Memory (GiB) 8, and GPU 0. An "Add User" button is visible. A modal window titled "Edit default quota" is open in the center. It displays "4 Users on this cluster" and allows setting resource limits: CPU (vCPU) 2, Memory (GiB) 16, and GPU 8. There are "Cancel" and "Update" buttons at the bottom of the modal. At the bottom of the main page, it says "Workspace: edu_cml_gpu" and "Cloud Provider: aws (AWS)".

Add GPUs to an Existing Workspace

- With the **MLUser** role, you can add GPUs to an existing workspace by editing its details and clicking on **+Add GPU**

Workspace Instances

Name	Instance Type	CPU	GPU	Memory	Count	Autoscale Range Min - Max
CML CPU Workers	m5.4xlarge	16	-	64 GiB	1	1 - 5
CML Infra	m5.2xlarge	8	-	32 GiB	2	2 - 3
Platform Infra	m5.large	2	-	8 GiB	2	2 - 4
CML GPU Workers	p2.8xlarge	32 CPU	8 GPU	488 GiB	<input type="button" value="0"/> - <input type="button" value="1"/>	<input type="button" value="Save"/> <input type="button" value="Cancel"/>

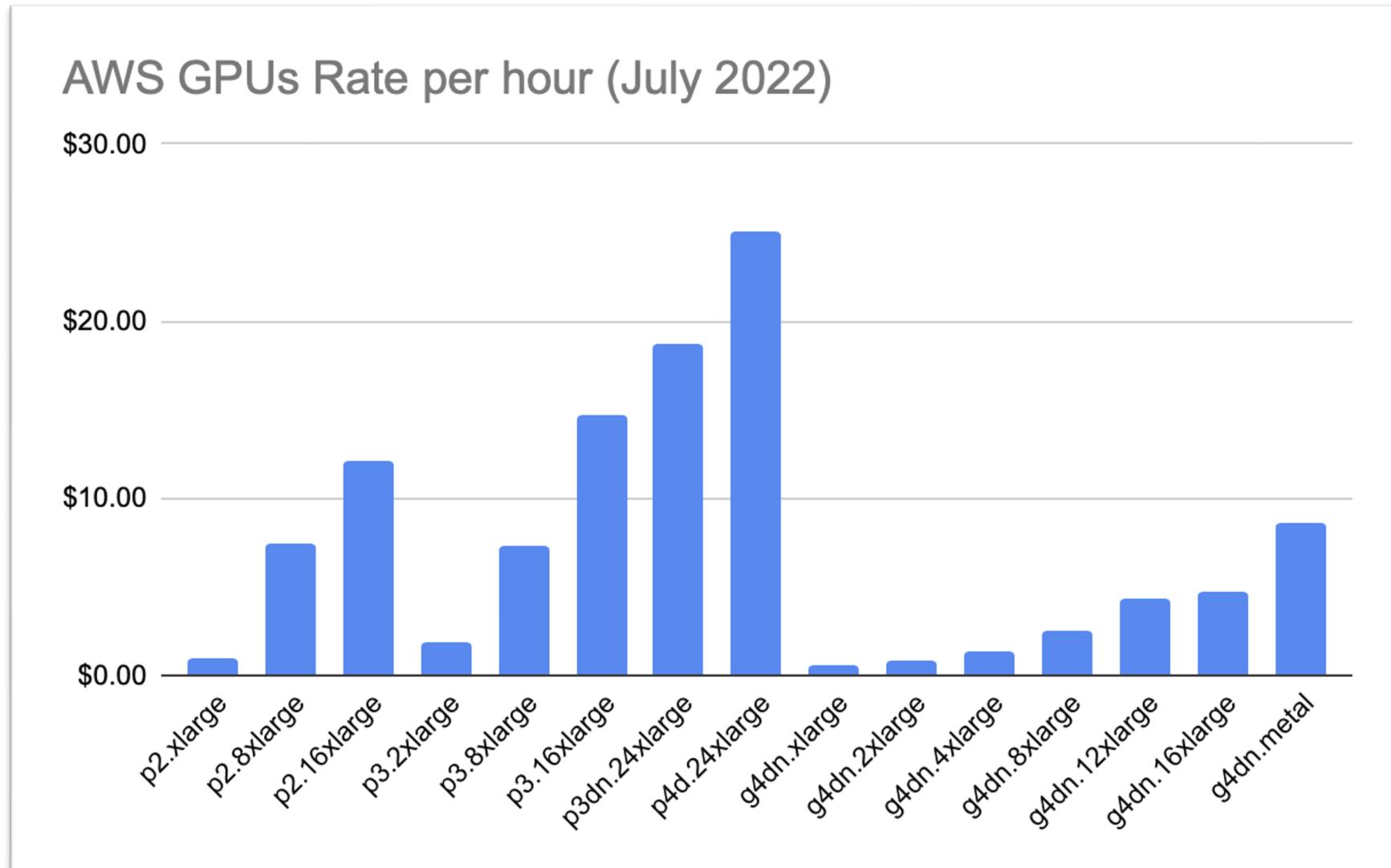
+ Add GPU

Subnets for Worker Nodes

Subnet Id	CIDR		
p3.16xlarge	64 CPU	8 GPU	488 GiB
p3.2xlarge	8 CPU	1 GPU	61 GiB
p3.8xlarge	32 CPU	4 GPU	244 GiB
g4dn.12xlarge	48 CPU	4 GPU	192 GiB

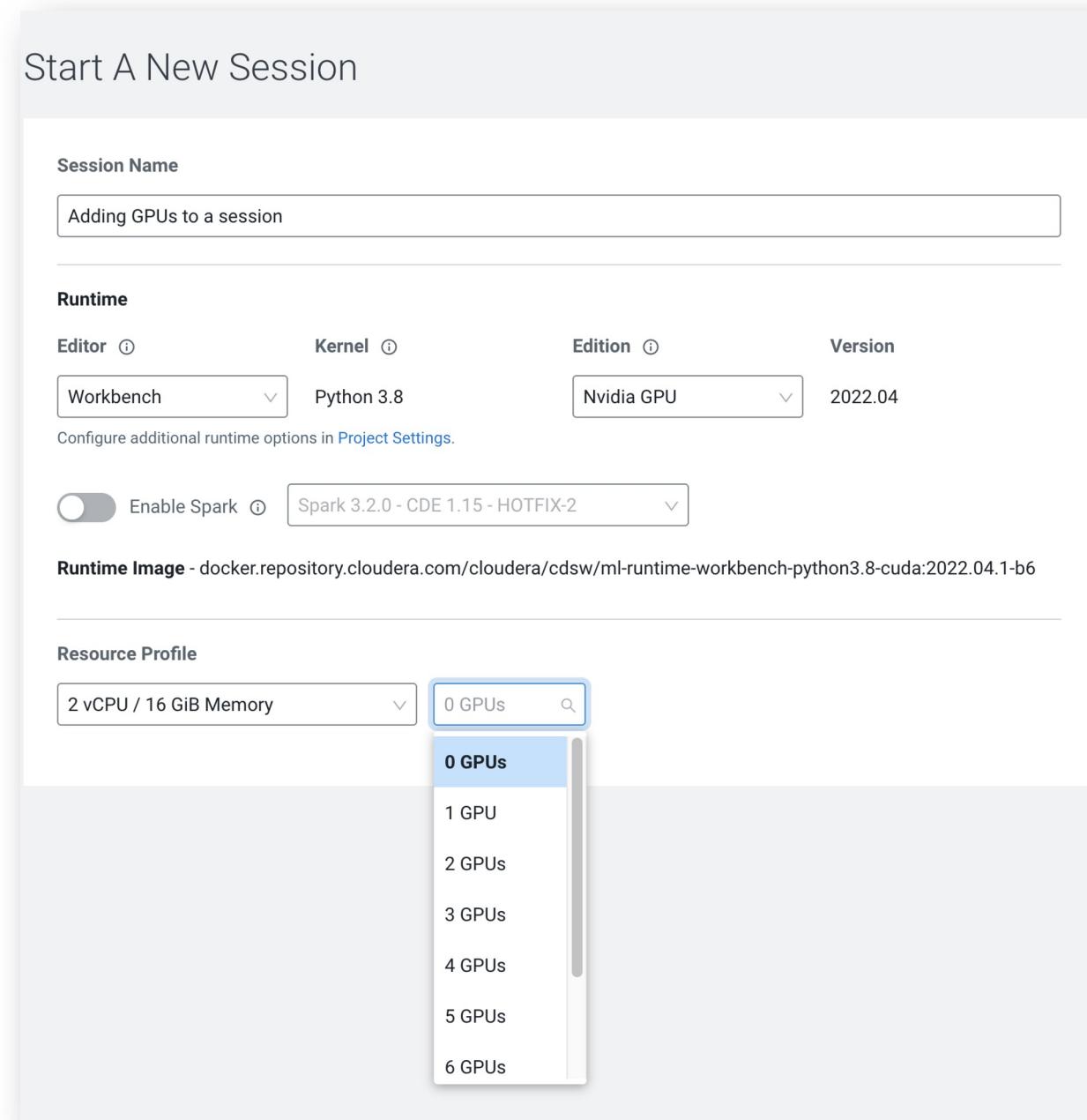
Available GPUs and Their Prices

- When selecting your instance type, you need to be aware of the financial implications



Add GPUs to a Session, Experiment, or Job

- Once your workspace has been provisioned with GPUs, you can leverage them in your session, experiment, or job by
 - Selecting the **Nvidia GPU** Edition and
 - Choosing the **number of GPUs** you want to use in the Resource Profile section



Write Hardware-Aware Code

- For instance, with TensorFlow use the `MirroredStrategy` class to distribute the processing on a single node with 0 or multiple GPUs

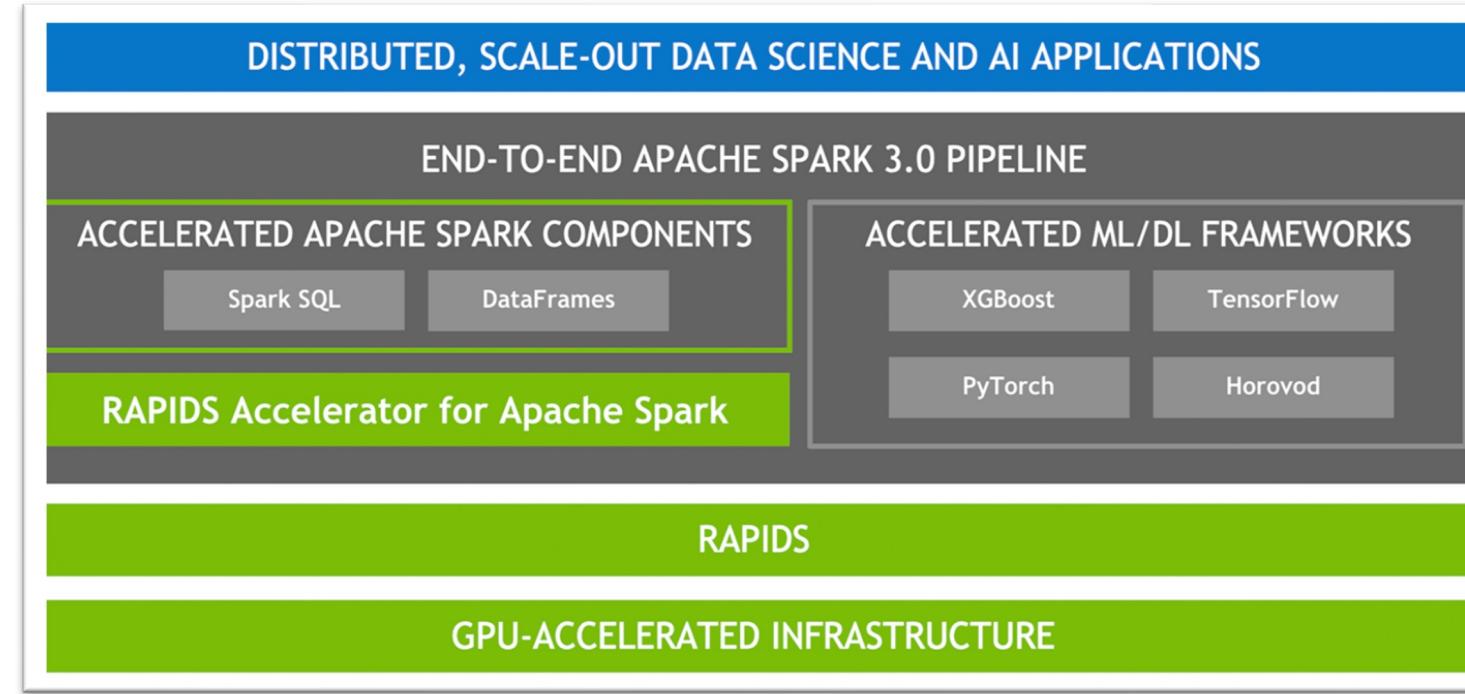
```
# Important: Mirrored Strategy is what allows us to automatically
#              leverage CPUs and/or GPUs that are available on the system.
mirrored_strategy = tf.distribute.MirroredStrategy()

with mirrored_strategy.scope():
    rnv2_model = keras.applications.ResNet50V2(include_top=False,
                                                weights="imagenet",
                                                input_tensor=t_size)

    ...
```

- For Spark, the RAPIDs project takes care of transparently translating your CPU DataFrames operations to their GPU counterparts

Accelerating Apache Spark 3.0 with GPUs and RAPIDS



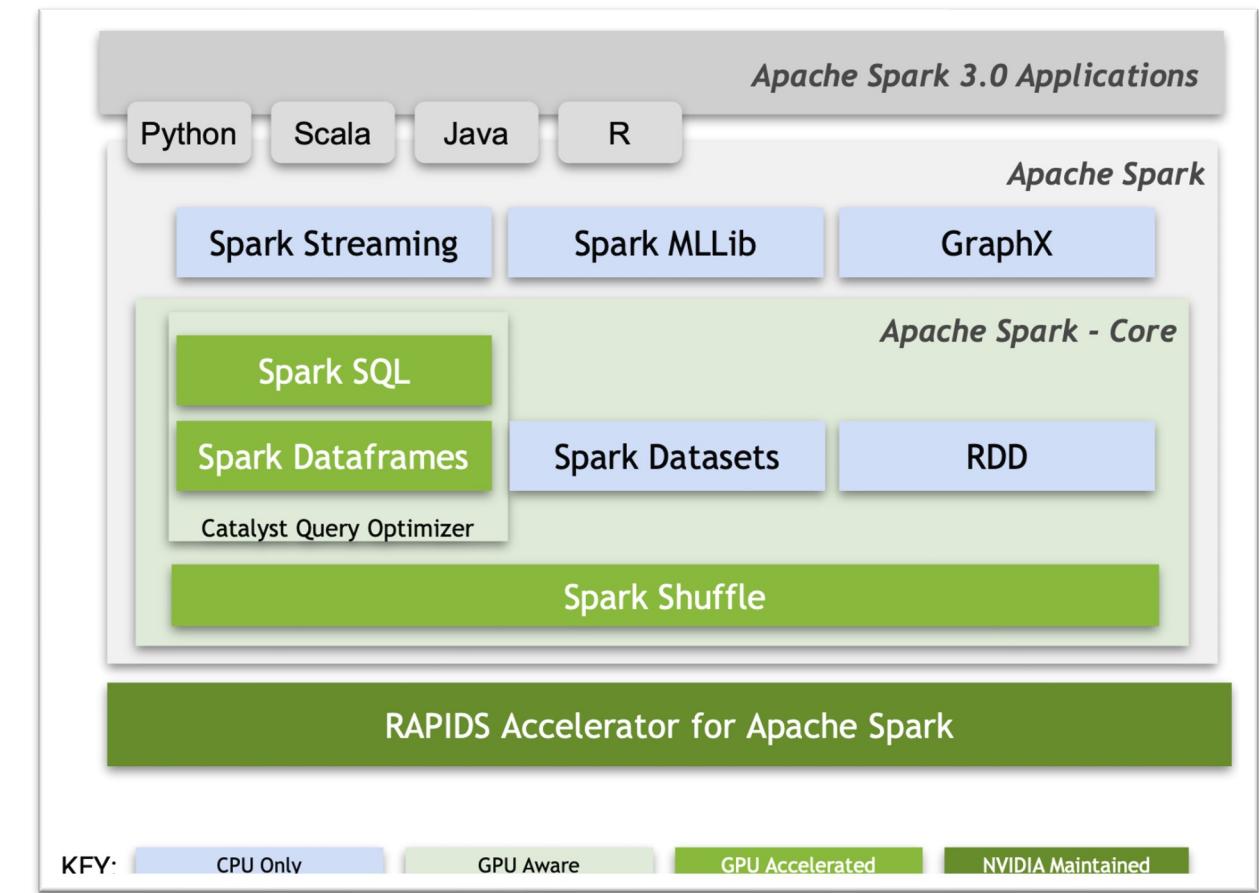
- NVIDIA has worked with the Apache Spark community to implement GPU acceleration through the release of Spark 3.0 and the open source RAPIDS Accelerator for Spark
- The RAPIDS Accelerator for Apache Spark uses GPUs to accelerate:
 - End-to-end data preparation and model training on the same Spark cluster
 - Spark SQL and DataFrame operations without requiring any code changes
 - Data transfer performance across nodes (Spark shuffles)

Source: <https://developer.nvidia.com/blog/accelerating-apache-spark-3-0-with-gpus-and-rapids/>

Data Workflow Software at NVIDIA

- **Requires**
 - Apache Spark - Version 3.0 or later
 - RAPIDS Accelerator for Apache Spark
 - Hardware with NVIDIA GPUs
- **Accelerated Operations**
 - Any code using the Spark SQL or Spark DataFrame interfaces, and any Spark Shuffle operations
 - Maximum benefit for longer running, highly compute bound Spark applications
- **Continuous/Ongoing Improvement**
 - Applications will inherit performance gains as NVIDIA expands coverage of Spark interfaces, operations, data types (etc.) and other optimizations

GPU-Accelerated Apache Spark Implementation



Transparent acceleration of any Apache Spark application - without code changes!

End-To-End GPU Acceleration for Spark Workloads

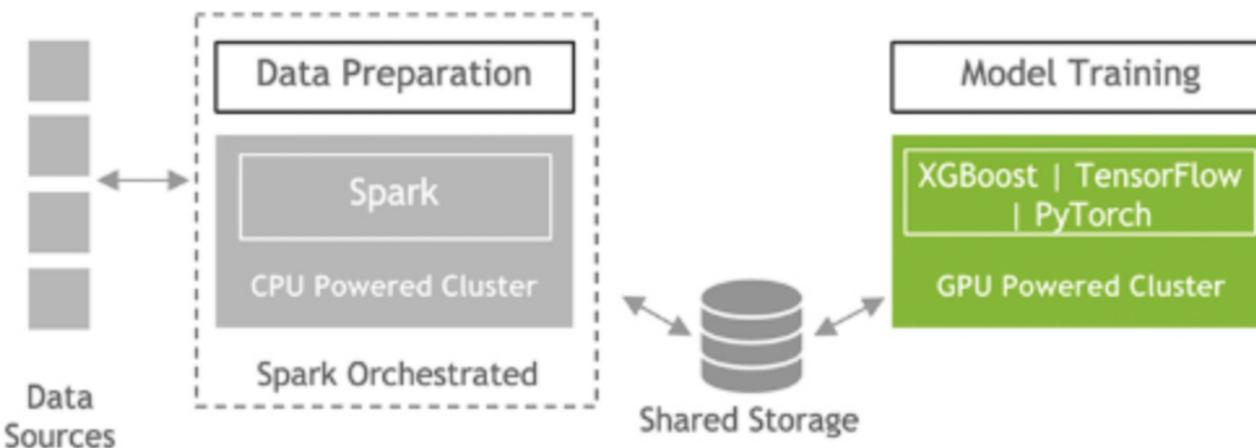
CLOUDERA

NVIDIA.

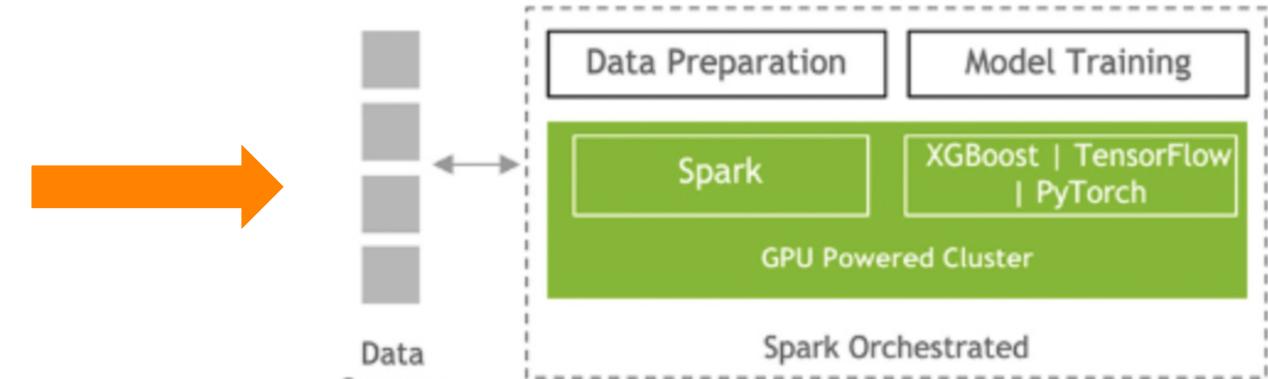
APACHE
Spark™

RAPIDS

Spark 2.x



Spark 3.0



Source: <https://developer.nvidia.com/blog/accelerating-apache-spark-3-0-with-gpus-and-rapids/>

Performance Gains

- Adding a GPU can make a massive difference
- Adding more GPUs has a less dramatic impact

Experiments										1 metrics ▾	Run Experiment
Run	Script	Arguments	Kernel	Comment	Submitter	Created At ▾	Train Time	Status	Duration	Actions	
3	main.py		Python 3.8	2 GPUs	bshimeldevuseast2_1	7/28/22 9:17 AM	0:07:46.508415	Success	8 mins		
2	main.py		Python 3.8	1 GPU	bshimeldevuseast2_1	7/28/22 8:53 AM	0:09:21.848153	Success	10 mins		
1	main.py		Python 3.8	0 GPU	bshimeldevuseast2_1	7/27/22 6:54 PM	1:52:50.234265	Success	113 mins		

Chapter Topics

Autoscaling, Performance, and GPU Settings

- Autoscaling Workloads
- Working with GPUs
- **Essential Points**
- Hands-On Exercise: Deep Learning with GPUs in CML

Essential Points

- CML uses Kubernetes to automatically scale your CPU and GPU workers using the ranges settings of your workspace
- The cloud makes provisioning GPU enabled machines frictionless
- But it is not free!
- Set the minimum range for GPU workers to 0 to save costs
- The MLAdmin can set quotas on GPUs per user
- Hardware aware code can dynamically adapt to the available hardware
- Adding a single GPU can make a huge difference in the performance of your workloads

Bibliography

- [Autoscaling Workloads with Kubernetes](#)
- [Choosing the right GPU for deep learning on AWS](#)
- [Machine Learning - AWS Instances](#)
- [CDP Public Cloud Machine Learning documentation](#)
- [Accelerating AI Training Using GPUs on Cloudera Machine Learning](#)
- [The TensorFlow MirroredStrategy class](#)
- [Accelerating Apache Spark 3.0 with GPUs and RAPIDS](#)

Chapter Topics

Autoscaling, Performance, and GPU Settings

- Autoscaling Workloads
- Working with GPUs
- Essential Points
- **Hands-On Exercise: Deep Learning with GPUs in CML**

Hands-On Exercise: Deep Learning with GPUs in CML

- In this exercise, you will execute a deep learning job with and without a GPU to
 - Measure the difference in performance
 - Observe the ML environment configuration and autoscaling capabilities
- Please refer to the Hands-On Exercise Manual for instructions



Model Metrics and Monitoring

Model Metrics and Monitoring

- **By the end of this chapter, you will be able to**
 - Understand the need for model monitoring
 - Identify some of the common model metrics used in monitoring
 - Implement model monitoring with Evidently
 - Build a continuous monitoring system with CML and Evidently

Chapter Topics

Model Metrics and Monitoring

- **Why Monitor Models?**
- Common Models Metrics
- Models Monitoring with Evidently
- Continuous Model Monitoring
- Essential Points
- Hands-On Exercise: Continuous Model Monitoring with Evidently

Model Monitoring Motivation

- A trained model is never final
 - Data drift
 - Concept drift
 - Will cause the predictive performance of a model to degrade over time, eventually making it obsolete for the task it was initially intended to solve
- More specifically, the ability of a trained model to generalize relies on a critical assumption of stationarity
 - meaning the data upon which a model is trained and tested are *independent and identically distributed*
- In real-world environments, this assumption is often violated, as human behavior and, consequently, the systems we aim to model are dynamically changing all the time

All models are wrong, but some are useful.

- George Box

Data Drift

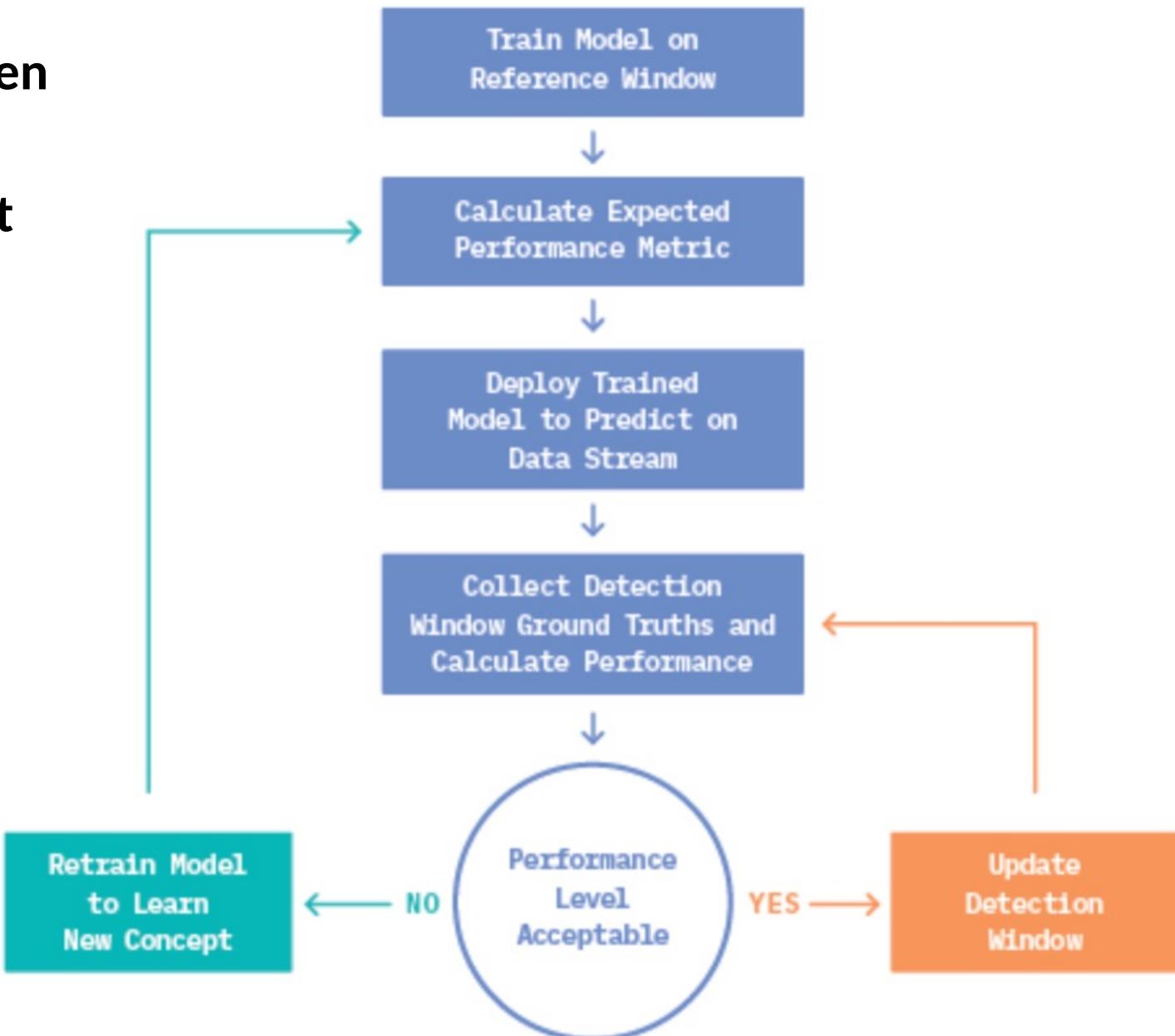
- A variation in the production data from the data that was used to test and validate the model before deploying it in production
- Examples:
 - Inventory management models accuracy can drop when unforeseen events such as the COVID-19 disrupt consumers behaviors
 - Spam filters models must relearn what language constitutes the evolving concept of spam to remain effective
 - Future energy consumption demand is driven by many non-stationary forces
 - Climate fluctuations
 - Population growth
 - Disruptive clean energy tech

Concept Drift

- **Occurs when the patterns the model learned no longer hold**
 - The model quality decline follows the gradual changes in external factors
- **Examples:**
 - Competitors launch new products
 - Consumers have more choices, and their behavior changes. As should sales forecasting models.
 - Macroeconomic conditions evolve
 - As some borrowers default on their loans, the credit risk is redefined
 - Scoring models need to learn it.
 - Mechanical wear of equipment
 - Under the same process parameters, the patterns are now slightly different
 - It affects quality prediction models in manufacturing

Adaptive Learning to the Rescue

- To combat this divergence between static models and dynamic environments, teams often adopt an adaptive learning strategy that is triggered by the detection of a drifting concept
- Supervised drift detection is generally achieved by
 - Monitoring a performance metric of interest such as accuracy
 - Alerting a retraining pipeline when the metric falls below some designated threshold



Adaptive Learning Main Limitation: Label Dependence

- Requires immediate access to an abundance of labels at inference time to quantify a change in system performance
 - A requirement that may be cost-prohibitive, or even outright impossible, in many real-world machine learning applications

*"To highlight the problem of label dependence, consider the task of detecting hate speech from live tweets, using a classification system facing the Twitter stream (estimated at 500M daily tweets). If 0.5% of the tweets are requested to be labeled, using crowdsourcing websites such as Amazon's Mechanical Turk², this would imply a daily expenditure of \$50K (each worker paid \$1 for 50 tweets), and a continuous availability of **350 crowdsourced workers** (assuming each can label 10 tweets per minute, and work for 12 hours/day), every single day, for this particular task alone. The scale and velocity of modern day data applications makes such dependence on labeled data a practical and economic limitation."*

On the Reliable Detection of Concept Drift from Streaming Unlabeled Data

Chapter Topics

Model Metrics and Monitoring

- Why Monitor Models?
- **Common Models Metrics**
- Models Monitoring with Evidently
- Continuous Model Monitoring
- Essential Points
- Hands-On Exercise: Continuous Model Monitoring with Evidently

Metrics for Regression

Let error = difference between the label and the prediction

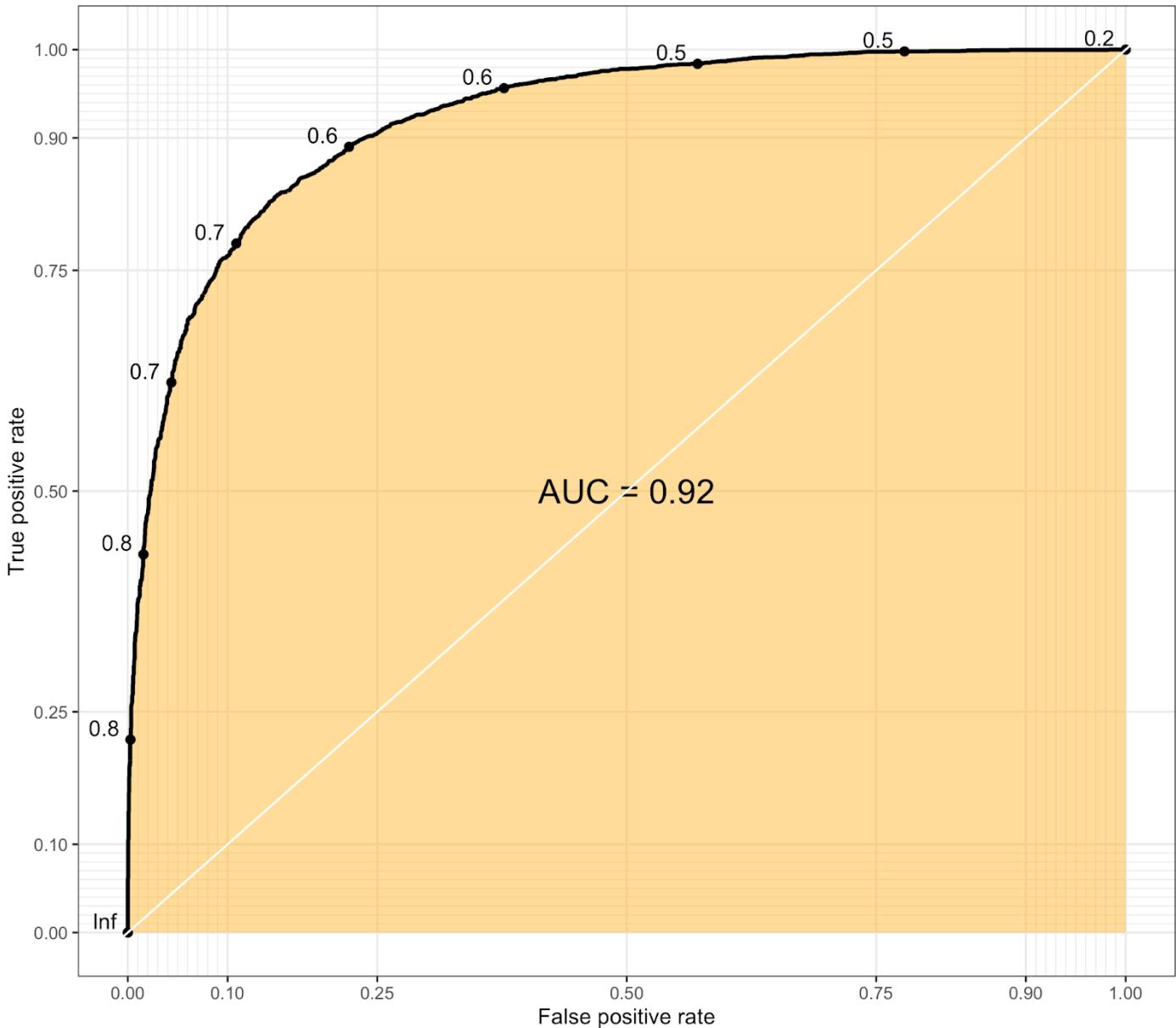
- **Mean Error (ME): average of all the errors**
 - Seldom used because positives and negatives cancel each other out
- **Mean Absolute Error (MAE): average of all the absolute value of the errors**
 - More meaningful: fixes the limitation of ME
- **Mean Absolute Percentage Error (MAPE): average of all the absolute value of the errors divided by their labels**
 - Super useful because it's a percentage: often used as a loss function in model evaluation
- **Mean Square Error (MSE): average of all the square of the errors**
 - Another way to fix the limitation of ME that also gives more weight to larger differences
- **Root Mean Square Error (RMSE) : root of MSE**
 - Compared to MSE, RMSE is measured in the same units as the target variable

Metrics for Classification

- **Accuracy:** correct predictions / total predictions
 - Can be dangerously misleading on skewed data
- **Confusion Matrix:** a matrix made of True Positives (TP), False Positives (FP), False Negatives (FN), True Negatives (TN)
- **Precision:** TP / TP + FP
 - Measures how good the model is at correctly identifying the positive class
- **Recall:** TP / TP + FN
 - Measures how good the model is at correctly predicting all the positive observations in the dataset
- **F1:** $2 * \text{Precision} * \text{Recall} / (\text{Precision} + \text{Recall})$
 - Combines Precision and Recall in a single number between 0 and 1 where 1 means perfect precision and recall

Metrics for Classification (2)

- The **ROC** (Receiver Operating Characteristics) curve is a plot of the true positive rate and the false positive rate at all classification threshold
 - Area Under the Curve (AUC):** measurement of the entire two-dimensional area under the curve and as such is a measure of the performance of the model at all possible classification thresholds
 - Perfect classification = 1
 - Random classification = 0.5



Metrics for Clustering

- **Silhouette** refers to a method of interpretation and validation of consistency within clusters of data. The technique provides a succinct graphical representation of how well each object has been classified. It was proposed by Belgian statistician Peter Rousseeuw in 1987
 - The silhouette value is a measure of how similar an object is to its own cluster (**cohesion**) compared to other clusters (**separation**)
 - The silhouette ranges from -1 to +1, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters
 - If most objects have a high value, then the clustering configuration is appropriate
 - If many points have a low or negative value, then the clustering configuration may have too many or too few clusters

[source Wikipedia]

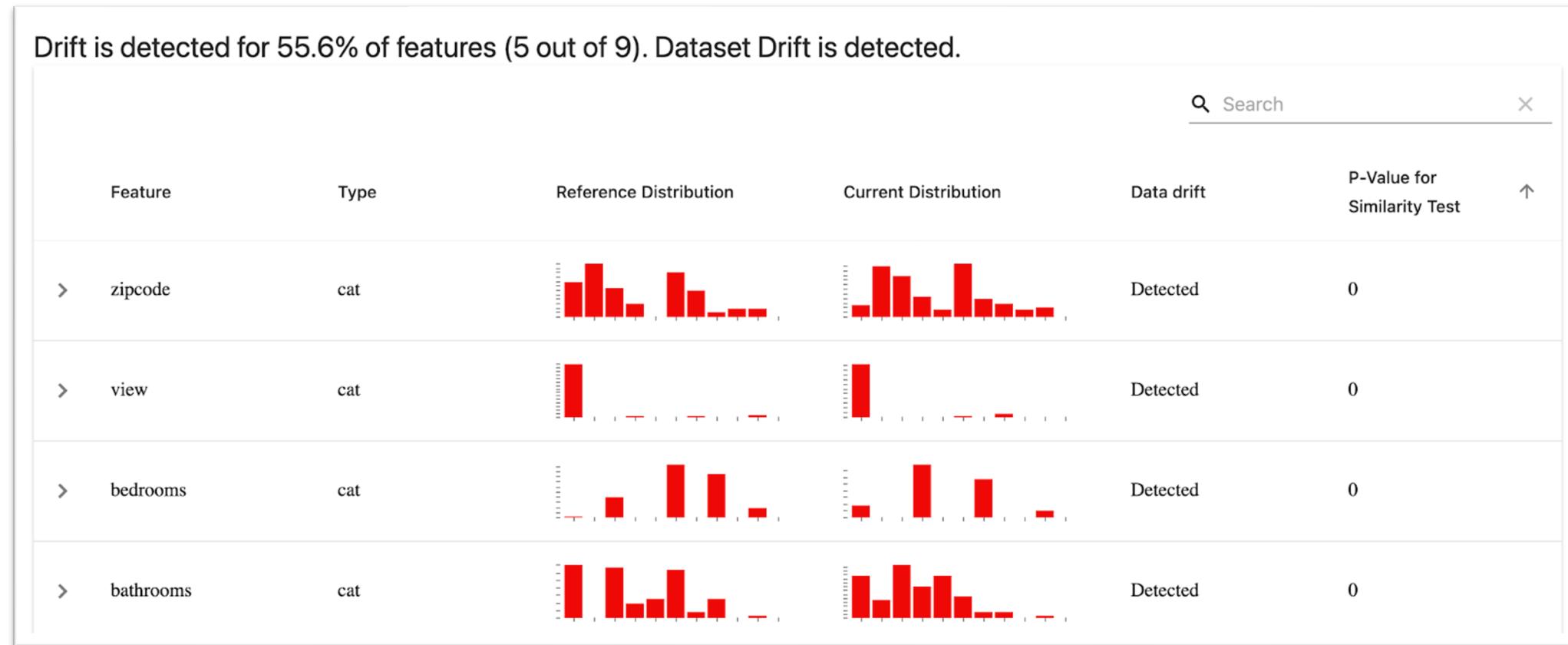
Chapter Topics

Model Metrics and Monitoring

- Why Monitor Models?
- Common Models Metrics
- **Models Monitoring with Evidently**
- Continuous Model Monitoring
- Essential Points
- Hands-On Exercise: Continuous Model Monitoring with Evidently

Measuring Data Drift

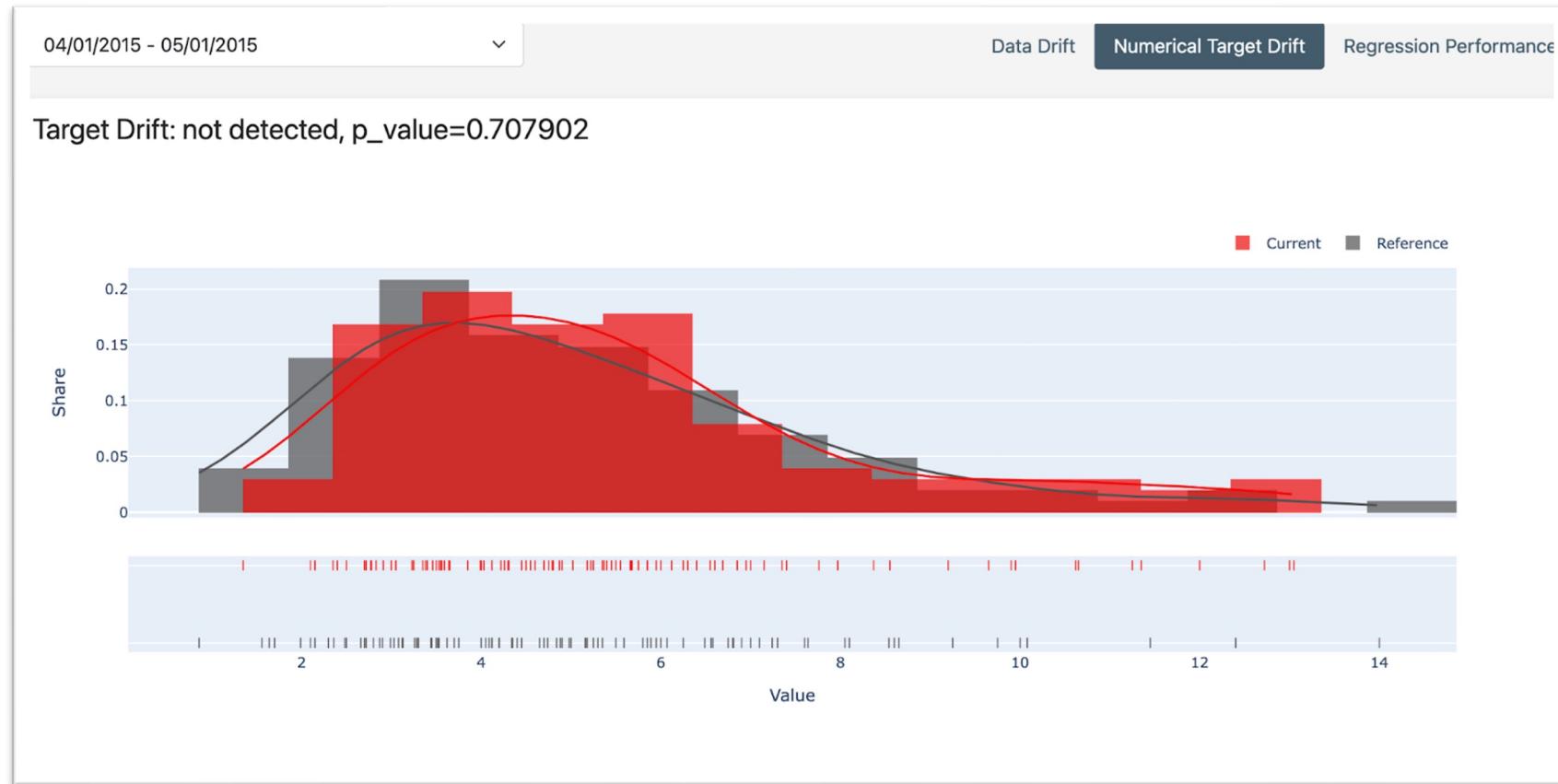
- To estimate the data drift Evidently compares the distributions of each feature in the two datasets using statistical tests to detect if the distribution has changed significantly



- The p-value is the probability of obtaining test results at least as extreme as the result actually observed, under the assumption that the null hypothesis is correct

Measuring Target Drift

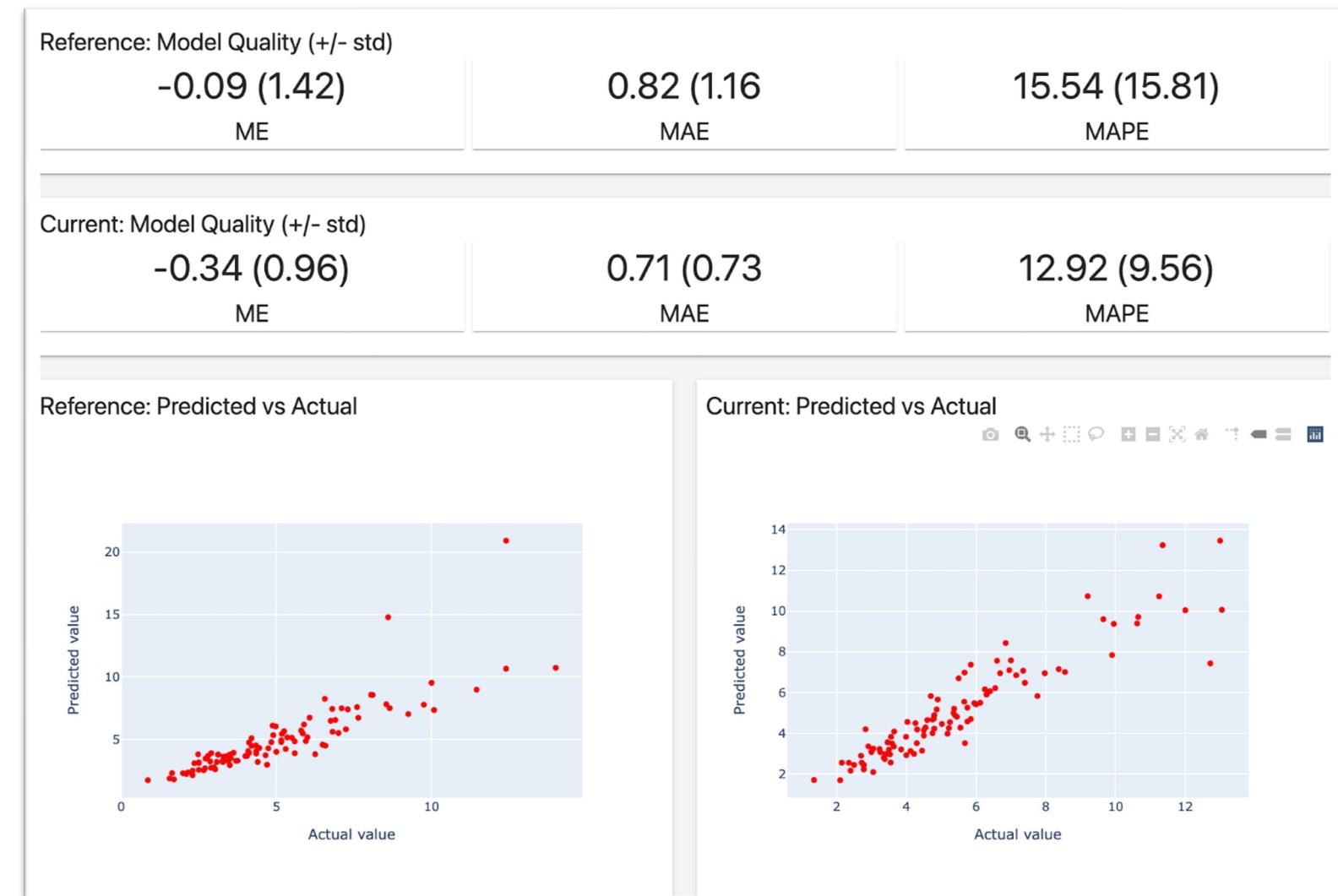
- To estimate the target drift Evidently compares the distributions of targets and predictions using statistical tests to detect if the distribution has changed significantly



- The p-value is the probability of obtaining test results at least as extreme as the result actually observed, under the assumption that the null hypothesis is correct

Measuring Regression Performance

- To measure the regression performance, Evidently calculates a few standard model quality metrics
 - Mean Error (ME)
 - Mean Absolute Error (MAE)
 - Mean Absolute Percentage Error (MAPE)



Using Evidently

- **Install the evidently library**

```
pip install evidently==0.1.32.dev0
```

- **Import the libraries you need from `evidently.dashboard import Dashboard`**

```
#import the tabs object you plan to use
from evidently.dashboard.tabs import DataDriftTab, CatTargetDriftTab
```

- **Prepare the data**

```
iris = datasets.load_iris()
iris_frame = pd.DataFrame(iris.data, columns = iris.feature_names)
```

Using Evidently (2)

- Generate the dashboard

```
iris_data_drift_report =  
    Dashboard(tabs=[DataDriftTab()])  
iris_data_drift_report.calculate(iris  
    _frame[:75],  
    iris_frame[75:], column_mapping =  
    None)  
iris_data_drift_report.show()
```



- Save your dashboard to html (optionally)

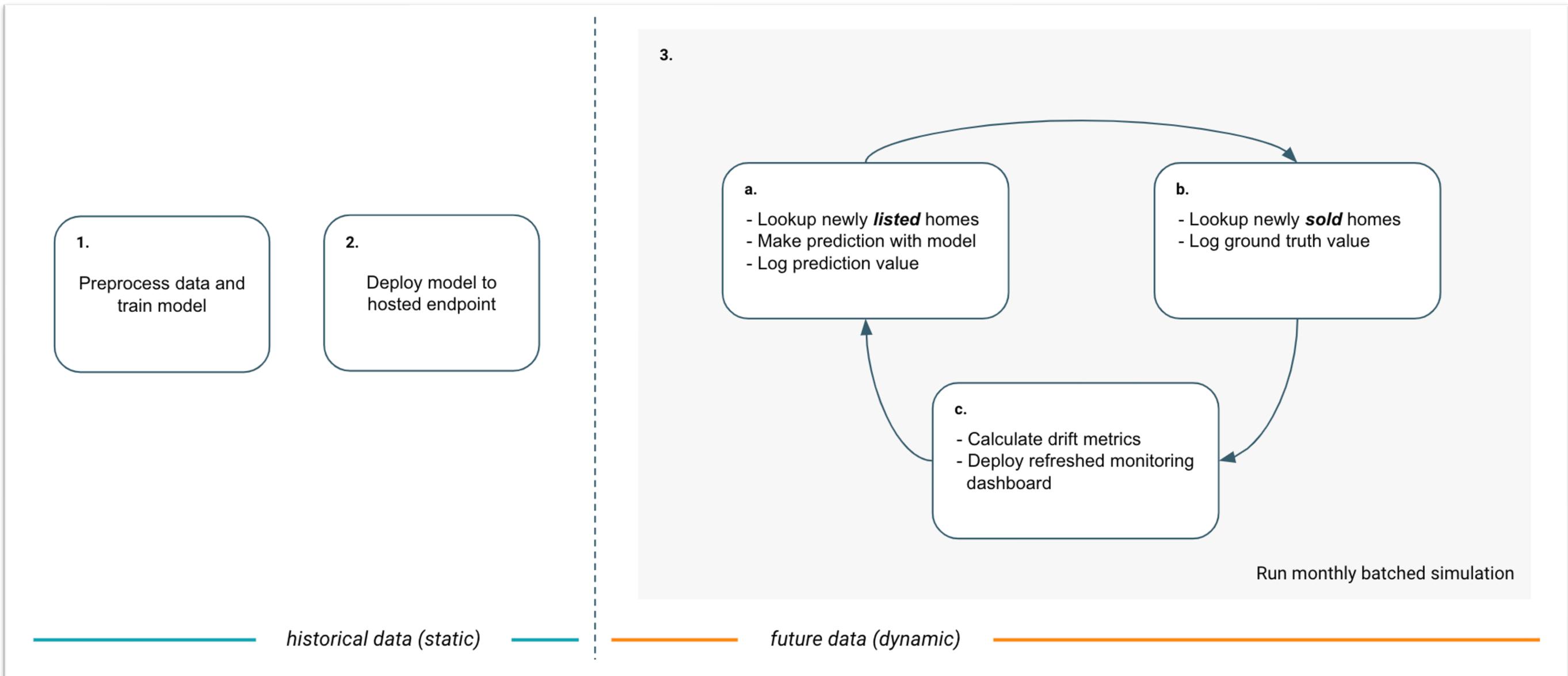
```
iris_data_drift_report.save("reports/  
myReport.html")
```

Chapter Topics

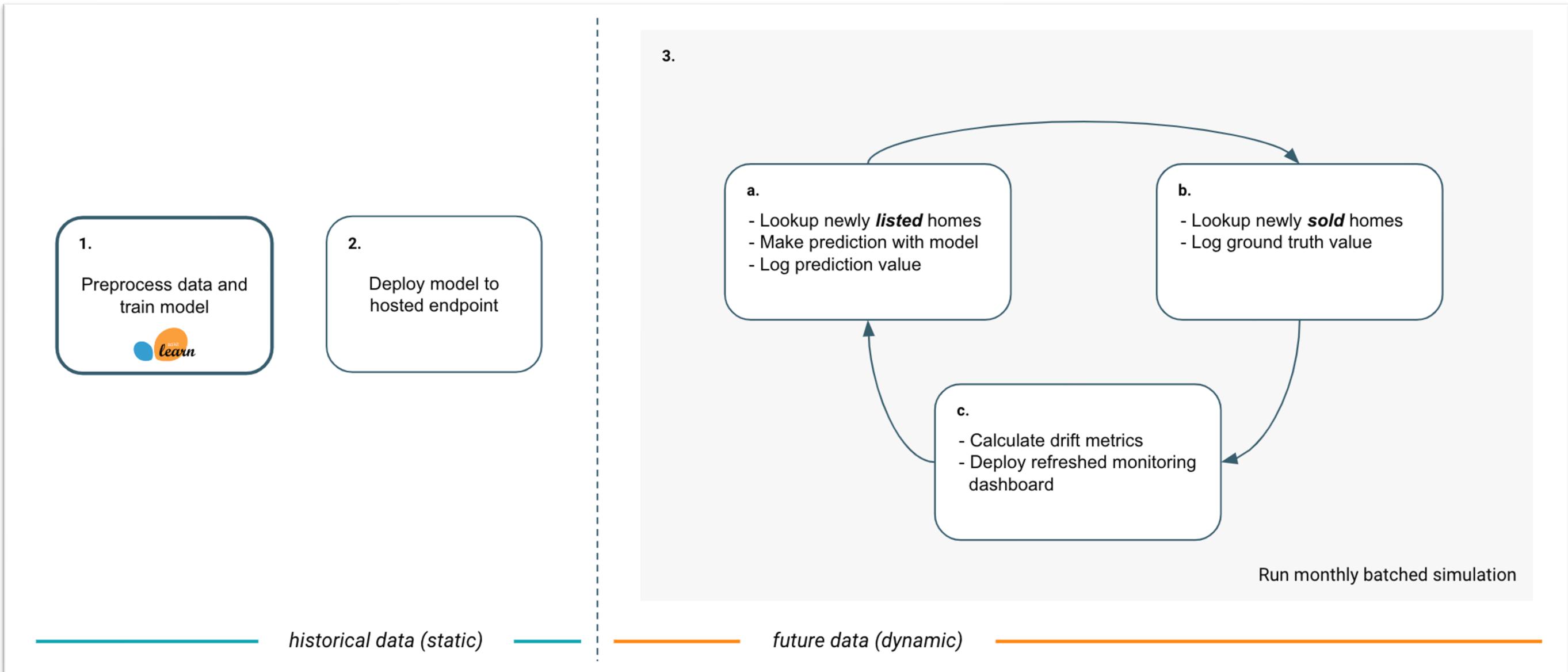
Model Metrics and Monitoring

- Why Monitor Models?
- Common Models Metrics
- Models Monitoring with Evidently
- **Continuous Model Monitoring**
- Essential Points
- Hands-On Exercise: Continuous Model Monitoring with Evidently

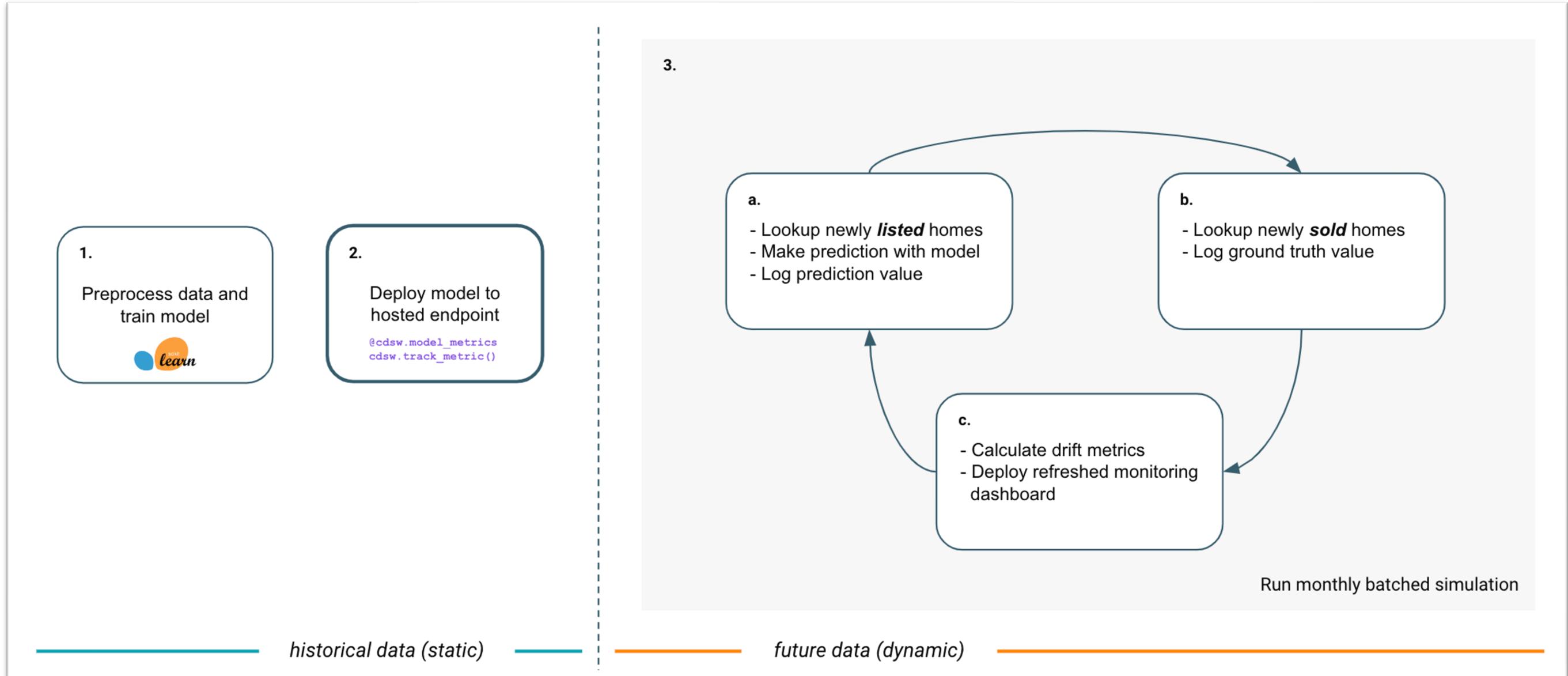
Continuous Model Monitoring AMP



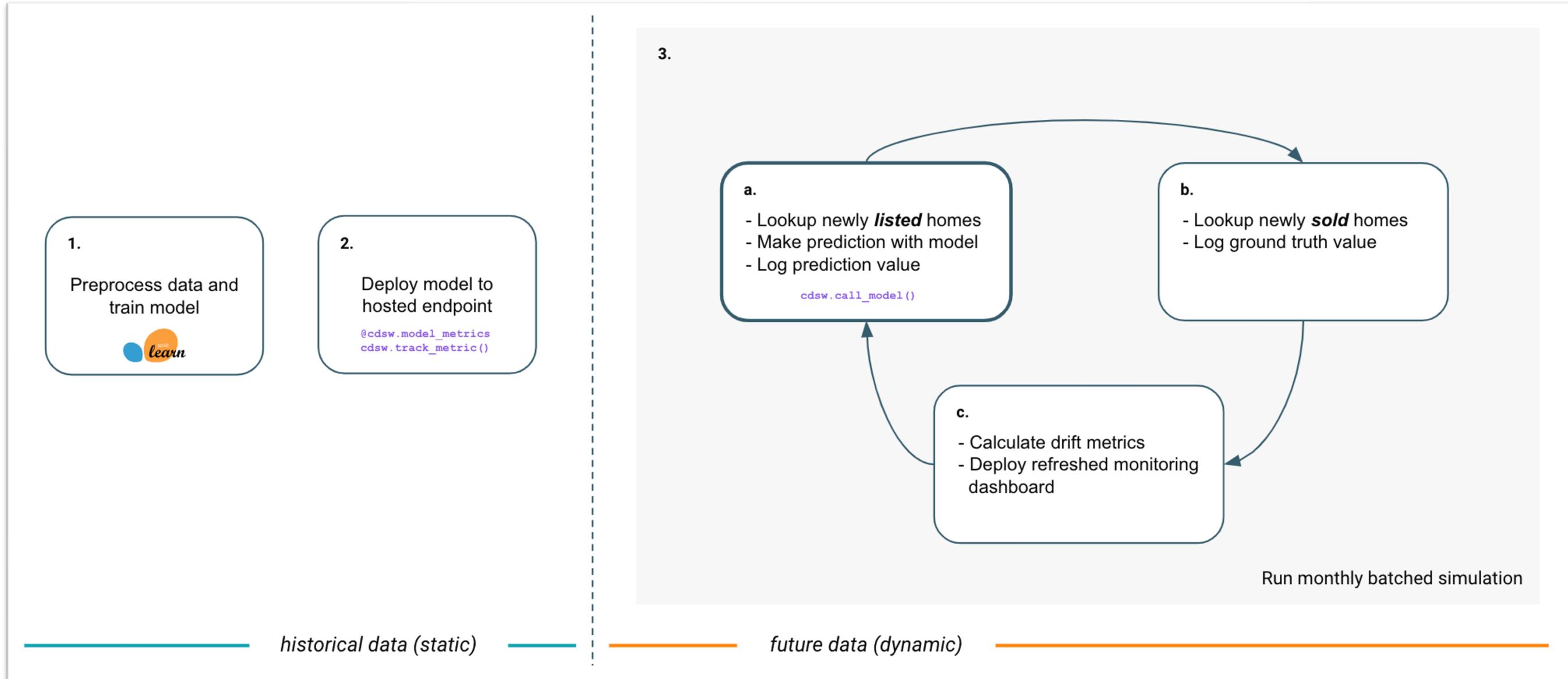
Continuous Model Monitoring AMP



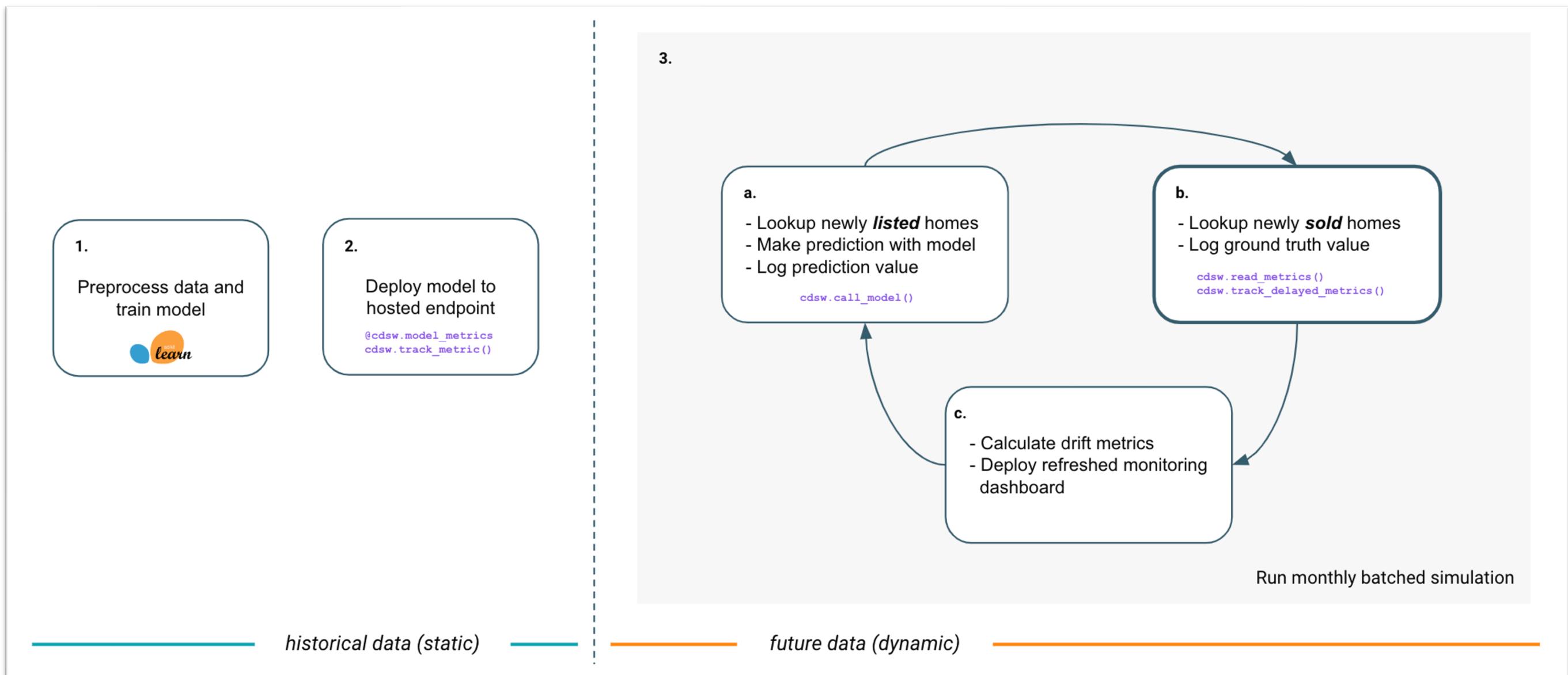
Continuous Model Monitoring AMP



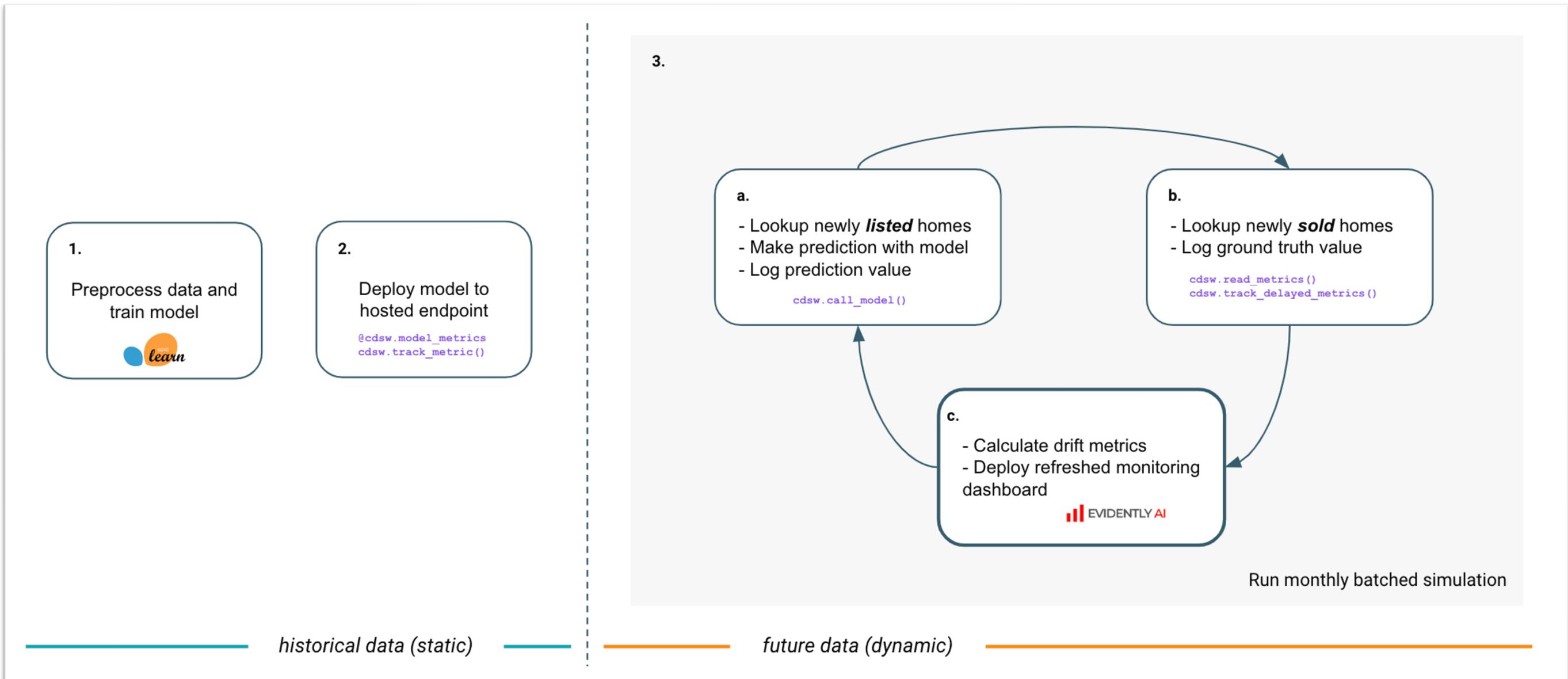
Continuous Model Monitoring AMP



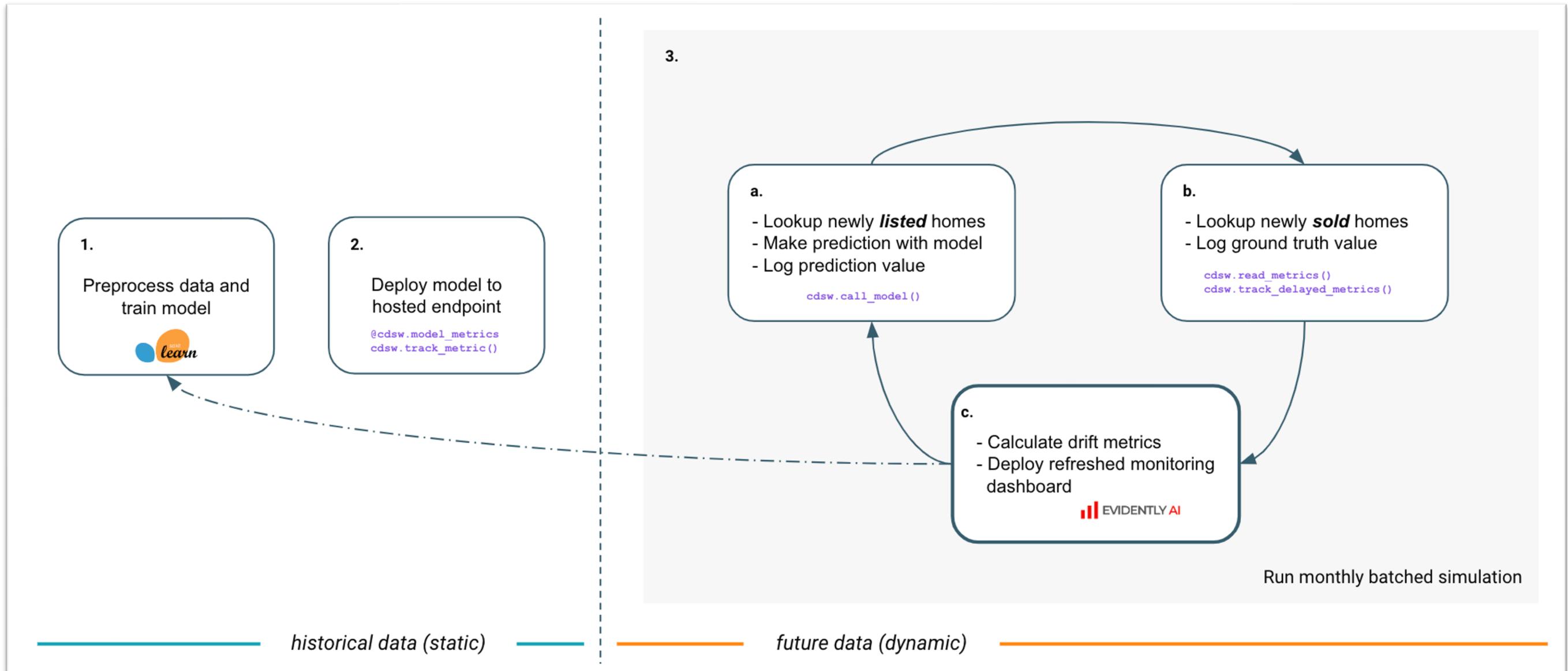
Continuous Model Monitoring AMP



Continuous Model Monitoring AMP



Continuous Model Monitoring AMP



Chapter Topics

Model Metrics and Monitoring

- Why Monitor Models?
- Common Models Metrics
- Models Monitoring with Evidently
- Continuous Model Monitoring
- **Essential Points**
- Hands-On Exercise: Continuous Model Monitoring with Evidently

Essential Points

- A trained model is never final
 - The ability of a trained model to generalize relies on critical and often false **assumption of stationarity**
- **Data drift** is defined as a variation in the production data from the data that was used to test and validate the model before deploying it in production
- **Concept drift** occurs when the patterns the model learned no longer hold
- The main limitation of adaptive learning is **label dependence**

Bibliography

- [EVIDENTLY AI](#)
- [Inferring Concept Drift Without Labeled Data](#)
- [On the Reliable Detection of Concept Drift from Streaming Unlabeled Data](#)
- [Statistics How To](#)
- [Statquest.org](#)
- [8 Metrics to Measure Classification Performance](#)
- [Silhouette \(clustering\)](#)

Chapter Topics

Model Metrics and Monitoring

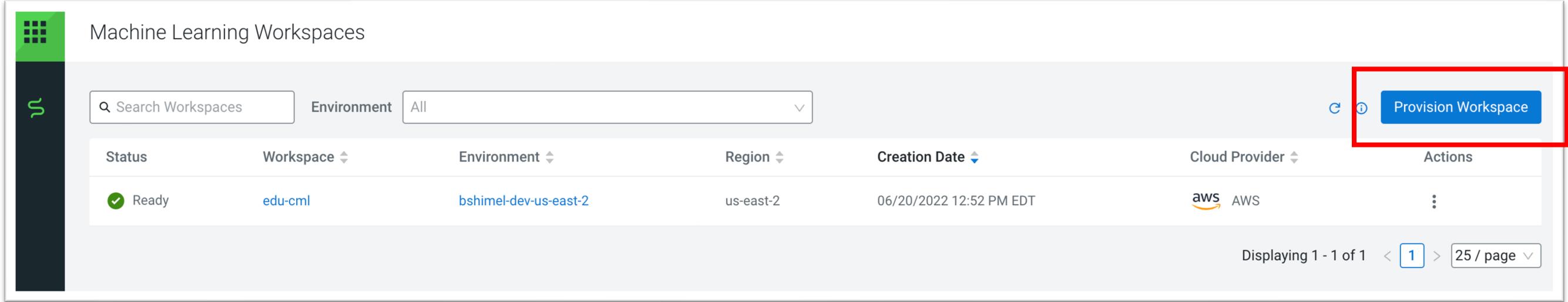
- Why Monitor Models?
- Common Models Metrics
- Models Monitoring with Evidently
- Continuous Model Monitoring
- Essential Points
- **Hands-On Exercise: Continuous Model Monitoring with Evidently**

Hands-On Exercise: Continuous Model Monitoring with Evidently

- **In this exercise, you will**
 - Start the Continuous Model Monitoring AMP
 - Launch the Price Regressor Monitoring dashboard
 - Identify the file that creates the Evidently dashboard
 - Explore drift and variations in the model performance
 - Experiment with Evidently reports
- **Please refer to the Hands-On Exercise Manual for instructions**

Appendix: Workspace Provisioning

Provision the Workspace



The screenshot shows the 'Machine Learning Workspaces' page. On the left is a sidebar with a green square icon containing a grid of squares and a dark blue square icon containing a white 'S'. The main area has a light gray header with a search bar ('Search Workspaces'), an 'Environment' dropdown set to 'All', and a red-bordered 'Provision Workspace' button. Below the header is a table with columns: Status, Workspace, Environment, Region, Creation Date, Cloud Provider, and Actions. One row is visible: 'Ready' status, 'edu-cml' workspace, 'bshimel-dev-us-east-2' environment, 'us-east-2' region, '06/20/2022 12:52 PM EDT' creation date, 'aws' cloud provider, and a three-dot 'Actions' menu. At the bottom, it says 'Displaying 1 - 1 of 1' and '25 / page'.

- **Click Provision Workspace**
 - Creates a new workspace within the environment
- **Provisioning can take up to an hour**
- **Domain name is randomly generated and cannot be changed**

Workspace and Environment

Provision Machine Learning Workspace

Provision an on-demand machine learning workspace.

* Workspace Name

Education_ML

Specify a unique name in the workspace

* Select Environment

 ps-sandbox-aws (us-east-2)

Environment type: AWS (us-east-2)

Select the environment where the ML workspaces must be provisioned

- Created by your CDP Admin

Advanced Options



Toggle to display Advanced Settings

Advanced Options: CPU Settings

Advanced Options

CPU Settings

Instance Type

m5.2xlarge 8 CPU 32 GiB ▼

Autoscale Range

0 5 10 30

Root Volume Size ⓘ

512

Default size of the root volume disk
for the nodes in the group

Advanced Options: GPU Settings

GPU Instances On

GPU Settings

Instance Type

p3.8xlarge	32 CPU	4 GPU	244 GiB	▼
------------	--------	-------	---------	---

Autoscale Range

GPU

0 30

Root Volume Size ⓘ

512

Default size of the root volume disk for the nodes in the group

Advanced Options: Network Settings

Network Settings

Subnets for Worker Nodes ⓘ

Subnets for Load Balancer ⓘ

Load Balancer Source Ranges ⓘ

Enable Fully Private Cluster

Enable Public IP Address for Load Balancer

Restrict access to Kubernetes API server to authorized IP ranges ⓘ

Optional: Select one or more subnets to use for Kubernetes worker nodes (AWS only)

Optional: Select one or more subnets to use for load balancer

Enter a CIDR range of IP addresses allowed to access the cluster (Azure only)

Check to enable access to your workspace

Advanced Options: Production Machine Learning

The screenshot shows the 'Production Machine Learning' configuration page. It includes sections for 'Production Machine Learning' settings and 'Other Settings'. The 'Production Machine Learning' section contains checkboxes for 'Enable Governance' (unchecked) and 'Enable Model Metrics' (checked). The 'Other Settings' section contains checkboxes for 'Enable TLS' (checked), 'Enable Monitoring' (checked), and 'Skip Validation' (unchecked). Below these are 'Tags' input fields ('Enter Key' and 'Enter Value') and a 'CML Static Subdomain' input field. Callout boxes provide descriptions for each setting.

Production Machine Learning

Enable Governance ⓘ

Enable Model Metrics ⓘ

Other Settings

Enable TLS ⓘ

Enable Monitoring ⓘ

Skip Validation ⓘ

Tags ⓘ

Enter Key

Enter Value

- +

CML Static Subdomain ⓘ

Integration with Atlas

Track, analyze, and store metrics

Tags are propagated to your cloud service provider account

Custom name for the workspace endpoint and URLs of models, applications, and experiments