

# Cloudera Machine Learning Workshop

## Agenda -

- Understand how to setup and get started with Cloudera Machine Learning
- Getting started with ML projects, writing code, using various editors, building experiments, deploying models and applications
- Learn about MLOps - Model Metrics and Model Governance
- Administering the environment, users and overall security for CML

## Steps

### 1. Open CDP, using the “admin” user within the Test Drive link.

Below is the workshop link, it will only be available for a short time after the workshop. You can register for an extra 48 hours following the workshop through [this link](#).

<https://trycdp.com/lease/credentials/200c33214076fa91765eddc2d4c3f91765344427032626cf121eb9650def6657>

email: [manishm@cloudera.com](mailto:manishm@cloudera.com)

### 2. Click the “Machine Learning” tile within the CDP Home Screen



The screenshot shows the Cloudera Management Console interface. On the left, a sidebar menu includes options like Dashboard, Environments, Data Lakes, User Management, Data Hub Clusters, Data Warehouses, ML Workspaces (which is highlighted in red), Classic Clusters, Shared Resources, and Global Settings. The main area is titled "Machine Learning Workspaces". It features a search bar, filter dropdowns for Environment (set to All), and a table with the following data:

Status	Workspace	Environment	Region	Creation Date	Cloud Provider	Actions
Ready	ml-cdp-trial-aws	cdptrialuser12	us-west-2	04/26/2022 1:12 AM +08	AWS AWS	⋮

At the bottom, there are pagination controls: "Displaying 1 - 1 of 1" and "25 / page". A small "DRe" watermark is visible in the bottom right corner.

**3. Click on the workspace called “[ml-cdp-trial-aws](#)” which is created for you**

Workspaces are the heart of the Cloudera Machine Learning (CML)

The screenshot shows the Cloudera Management Console interface for Machine Learning Workspaces. On the left is a dark sidebar with navigation links: Dashboard, Environments, Data Lakes, and User Management. The main area is titled "Machine Learning Workspaces". It features a search bar, a dropdown for "Environment" set to "All", and a "Provision Workspace" button. A table lists one workspace entry:

Status	Workspace	Environment	Region	Creation Date	Cloud Provider	Actions
Ready	cdptrialuser36-ml	cdptrialuser36	us-west-2	01/19/2022 5:34 PM AEDT	AWS	<span>⋮</span>

A Workspace is a small cluster that runs on a kubernetes service to provide teams of data scientists to develop, test, train, and ultimately deploy machine learning models. **Click into the Workspace by clicking the Workspace name.**

This screenshot is identical to the one above, showing the Cloudera Management Console Machine Learning Workspaces page with a single workspace entry named "cdptrialuser36-ml".

#### 4. Here you can see all projects created

You can visualize all of the Projects and Resources that are part of the Projects page. Next we will create a Project where we will develop and deploy models along with other CML features.

The screenshot shows the Cloudera Machine Learning Projects page. The left sidebar includes links for Projects, Sessions, Experiments, Models, Jobs, Applications, User Settings, AMPS, Runtime Catalog, Site Administration, and Learning Hub. The main area is titled "Projects" and displays a summary of active workloads and user resources. Below this, a grid of project cards is shown:

Project Name	Description	Created by	Last worked on
Churn Modeling with sc...	Churn Modeling with sc...	trial12_admin	6 minutes ago
Few-Shot Text Classific...	Few-Shot Text Classific...	trial12_admin	3 hours ago
Explaining Models with ...	Explaining Models with ...	trial12_admin	an hour ago
Deep Learning for Imag...	Deep Learning for Imag...	trial12_admin	3 hours ago

## 5. Let's make a new project for you - Click on AMP's

The screenshot shows the Cloudera Machine Learning interface with the 'Applied ML Prototypes' section highlighted. The prototypes include:

- 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
- Deep Learning for Question Answering**: AUTOMATED QUESTION ANSWERING, EXTRACTIVE
- Explaining Models with LIME and SHAP**: INTERPRETABILITY, EXPLAINABILITY

The screenshot shows the 'Churn Modeling with scikit-learn' prototype details. The modal window includes:

- Description: This project demonstrates how to build a logistic regression classification model to predict the probability that a group of customers will churn from a fictitious telecommunications company. In addition, the model is interpreted using a technique called Local Interpretable Model-agnostic Explanations (LIME). Both the logistic regression and LIME models are deployed using CMS's real-time model deployment capability and interact with a basic Flask-based web application.
- Tags: CHURN PREDICTION, LOGISTIC REGRESSION, EXPLAINABILITY, LIME
- Buttons: Cancel, Configure Project

## 6. Click on the “Churn Modeling with scikit-learn” AMP and say configure project

Change the table name to something like “**churn\_prototype\_<username>**” and execute all the steps.

## Configure Project: Churn Modeling with scikit-learn - trial12\_admin 2

AMP Name: ML Churn Prototype (v2)  
Prototype to demonstrate building a churn model on CML

### Environment Variables

The settings below were defined by the AMP:

Name	Value	Description
DATA_LOCATION	data/churn_prototype	Relative path that will be used to store the data used for this prototype. This should be a location you have write access to, and which is suitable for non-production data.
HIVE_DATABASE	default	Name of the Hive database that will be used to create the Hive table used for this prototype. This should be a Hive database you have write access to, and which is suitable for non-production data.
HIVE_TABLE	churn_prototype_username	Name of the Hive table that will be created and populated with the data used for this prototype. If the table already exists, the prototype will assume it already contains the data for this prototype.

### Default Engine:

[Runtime](#) [Engine](#)

#### Runtime

Editor ⓘ Kernel ⓘ Edition ⓘ Version  
Workbench Python 3.7 Standard 2021.12

Enable Spark ⓘ Spark 2.4.7 - CDP 7.2.11 - CDE 1.13 - HO... ▾

#### Runtime Image

- docker.repository.cloudera.com/cloudera/cdsu/ml-runtime-workbench-python3.7-standard:2021.12.1-b17

⚠ The runtime addons required for this AMP is not present. Here is the list of runtime addons required for this AMP:  
Spark 2.4.7 - CDP 7.2.10 - CDE 1.11

#### Setup Steps

Execute AMP setup steps

manishm / Churn Modeling with scikit-learn - manishm 1 / AMP Status

Project quick find + M manishm ▾

### AMP Setup Steps

AMP Name: ML Churn Prototype (v2)  
Prototype to demonstrate building a churn model on CML

Completed 0 of 7 steps

 Step 1	Install dependencies, set environment variables, and upload data	started 4/22/2022 3:16 PM
 Step 2	Ingest data into our Hive table	not yet started
 Step 3	Train models	not yet started
 Step 4	Create the churn model prediction api endpoint	not yet started
 Step 5	Build model	not yet started
 Step 6	Deploy model	not yet started
 Step 7	Start Application	not yet started

7. A project will get created called “Churn Modeling with scikit-learn - <username> 1”. Wait for all the steps to finish

manishm / Churn Modeling with scikit-learn - manishm 1 / AMP Status

Project quick find + M manishm ▾

### AMP Setup Steps

AMP Name: ML Churn Prototype (v2)  
Prototype to demonstrate building a churn model on CML

Completed 0 of 7 steps

Step	Description	Status
Step 1	Install dependencies, set environment variables, and upload data <a href="#">View details</a>	started 4/22/2022 3:16 PM
Step 2	Ingest data into our Hive table	not yet started
Step 3	Train models	not yet started
Step 4	Create the churn model prediction api endpoint	not yet started
Step 5	Build model	not yet started
Step 6	Deploy model	not yet started
Step 7	Start Application	not yet started

Step 1: Install dependencies, set environment variables, and upload data [View details](#) (started 4/22/2022 3:16 PM)

STORAGE and STORAGE\_MODE environment variables and copy the data from raw/WA\_Fn-UseC\_-Telco-Customer-Churn-.csv into specified path of the STORAGE location, if applicable.

The STORAGE environment variable is the Cloud Storage location used by the DataLake to store hive data. On AWS it will be s3://[something], on Azure it will be abfs://[something] and on a CDSW cluster, it will be hdfs://[something]

Install the requirements

```
!pip3 install -r requirements.txt
```

Truncating text at 800000 characters to improve display performance.

Increase this limit with the environment variable 'MAX\_TEXT\_LENGTH'

Wait for all the steps to finish. I will take approximately 10 minutes to complete.

## AMP Setup Steps

AMP Name: ML Churn Prototype (v2)

Prototype to demonstrate building a churn model on CML

Completed 4 of 7 steps

- ✓ Step 1 Install dependencies, set environment variables, and upload data [View details](#) completed 4/22/2022 3:21 PM

```
the qualifiedName of the hive_table object representing
metadata:                                     #
this is a predefined key for additional metadata      #
    query: "select * from historical_data"          #
suggested use case: query used to extract training data
    training_file: "code/4_train_models.py"          #
suggested use case: training file used
    """
with open("lineage.yml", "w") as lineage:
    lineage.write(yaml_text)
```

- ✓ Step 2 Ingest data into our Hive table [View details](#) completed 4/22/2022 3:23 PM

```
learning/cloud/jobs-pipelines/topics/ml-creating-a-job.html).
> You can create a Job with specified command line arguments or environment
variables.
> Jobs can be triggered by the completion of other jobs, forming a
> [Pipeline](https://docs.cloudera.com/machine-learning/cloud/jobs-
pipelines/topics/ml-creating-a-pipeline.html)
> You can configure the job to email individuals with an attachment, e.g. a csv
report which your
> script saves at: `/home/cdsweb/job1/output.csv`.
Try running this script `1_data_ingest.py` for use in such a Job.
```

- ✓ Step 3 Train models [View details](#) completed 4/22/2022 3:23 PM

Notice also that any script that will run as an Experiment can also be run as a Job or in a Session.  
Our provided script can be run with the same settings as for Experiments.  
A common use case is to \*\*automate periodic model updates\*\*.  
Jobs can be scheduled to run the same model training script once a week using the latest data.  
Another Job dependent on the first one can update the model parameters being used in production  
if model metrics are favorable.

## 8. Once finished let's look at the overview of the project

Overview gives you access to all the features of a CML project. Initially it is good to start on the management components of a project. The project has deployed Jobs, Models and Applications

In project settings, Lets rename the project to something like - "Churn Modeling with scikit-learn - <you name>"

The screenshot shows the 'Project Settings' page for a project named 'churn-modeling-with-scikit-learn-manishm'. The left sidebar lists various project components: All Projects, Overview, Sessions, Experiments, Models, Jobs, Applications, Files, Collaborators, and Project Settings (which is selected). The main content area displays the 'Project Settings' form with the following fields:

- Project Name:** Churn Modeling with scikit-learn - manishm
- Project Description:** Build an scikit-learn model to predict churn using customer telco data.
- Visibility:** A radio button is selected for 'Private' (Only Collaborators can view or edit the project).
- Update Project:** A blue button at the bottom of the form.

## 9. Project Basics

### a. Overview

The screenshot shows the Cloudera Machine Learning interface. On the left, a sidebar navigation includes 'All Projects', 'Overview' (selected), 'Sessions', 'Experiments', 'Models', 'Jobs', 'Applications', 'Files', 'Collaborators', and 'Project Settings'. A 'Get Started' button and a 'Help' link are also present.

The main content area displays a project titled 'Churn Modeling with scikit-learn - manishm'. It shows a status bar with '1 running' and a progress bar indicating 'Project creation in progress...'. Below this, a step-by-step guide says 'Step 1 of 7 Install dependencies, set environment variables, and upload data' with a 'View details' link. The date 'started 4/26/2022 12:07 PM' is also shown.

Under 'Models', it says 'This project has no models yet. Create a new model.' Under 'Jobs', it says 'This project has no jobs yet. Create a new job to document your analytics pipelines.' A 'Files' section lists the following files:

Name	Size	Last Modified
code	-	2 minutes ago
flask	-	2 minutes ago
images	-	2 minutes ago
models	-	2 minutes ago
raw	-	2 minutes ago
cdsw-build.sh	45 B	2 minutes ago
LICENSE.txt	9.94 kB	2 minutes ago
model_metrics.db	828.00 kB	2 minutes ago
README.md	3.38 kB	2 minutes ago
requirements.txt	169 B	2 minutes ago

A 'Show Hidden Files' link is at the bottom of the file list.

Below the file list, there is a section titled 'Churn Modeling with scikit-learn' which contains a snippet of code and a table of data:

```

id Probability gender SeniorCitizen Partner Dependents tenure PhoneService MultipleLines InternetService OnlineSecurity OnlineBackup
951 0.668 Male No No No 22 Yes Yes Fiber No No
2999 0.638 Male No Yes Yes 1 Yes No Fiber No No
5400 0.638 Female Yes No No 26 Yes Yes Fiber No Yes
6315 0.494 Female No Yes Yes 9 Yes No Fiber No No

```

### b. Code

#### Files

[Download](#) [New](#) [Upload](#)

Name	Size	Last Modified
code	-	5 minutes ago
flask	-	5 minutes ago
images	-	5 minutes ago
models	-	5 minutes ago
raw	-	5 minutes ago
cdsw-build.sh	45 B	5 minutes ago
LICENSE.txt	9.94 kB	5 minutes ago
model_metrics.db	828.00 kB	5 minutes ago
README.md	3.38 kB	5 minutes ago
requirements.txt	169 B	5 minutes ago

[Show Hidden Files](#)

### c. Models

#### Models

Model	Status	Replicas	CPU	Memory	Last Deployed	Actions
Churn Model API Endpoint	Deployed	1 / 1	1	2.00 GiB	Apr 26, 2022, 09:07 AM	<button>Stop</button> <button>▼</button>

### d. Jobs

We will create a job later.

#### Jobs

This project has no jobs yet. Create a [new job](#) to document your analytics pipelines.

### e. Project Collaborators

You can give access to other users with certain permissions for the encompassing project so teams of users can collaborate together. You can set up Admins, Contributor, Operator, and Viewer permissions. (Since email is not configured this won't work in this environment)

The screenshot shows the Cloudera Machine Learning interface. On the left, there's a sidebar with navigation links: All Projects, Overview, Sessions, Experiments, Models, Jobs, Applications, Files, Collaborators (which is highlighted), and Project Settings. The main content area shows the URL trial36\_admin / Telco\_churn1 / Collaborators. At the top right, there's a search bar labeled "Project quick find", a "+" button, a user icon for trial36\_admin, and a grid icon. The main title is "Collaborators". A message says, "This project is private. Only collaborators can view and edit this project. [Change Settings](#)". Below it, there's an "Add Collaborator" section with a note: "Email is not configured. Please contact your administrator." There's a search bar and an "Add" button. A table lists one collaborator: trial36\_admin with the permission "Owner". A note at the bottom says, "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."

## 10. Project Settings

Taking a look at Project Settings, this is where you can define several options for the current project. You have the ability to define different engines where your code in CML will run. There are project variables that can be defined and used throughout your code. SSH tunnels can also be configured to connect to other services as needed. More details can be found in our docs [here](#).

### Change Project Names -

The screenshot shows the 'Project Settings' page. At the top, there is a navigation bar with tabs: Options, Runtime/Engine (which is highlighted in blue), Advanced, SSH Tunnels, Data Connections, Prototype, and Delete Project. Below the navigation bar, there are three main sections: 'Project Name' (containing the value 'Churn Modeling with scikit-learn - trial12\_admin 3'), 'Project Description' (containing the value 'Build an scikit-learn model to predict churn using customer telco data.'), and 'Visibility' (with two radio button options: 'Private' (selected) and 'Public'). At the bottom of the form is a blue 'Update Project' button.

### Change Runtimes -

## Project Settings

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

(+) Add Runtime

There are no ML Runtimes configured for this project, in this case workloads can use any Runtime that's available in the Workspace. To specify what Runtimes should be used in the Project, click on the Add Runtime button.

## Set Variables -

## Project Settings

**Environment Variables**  
Set project environment variables that can be accessed from your scripts.

Environment variable **values** are only visible to **collaborators** with **write** or higher access. They are a great way to securely store confidential information such as your AWS or database credentials. Names are available to all users with access to the project.

DATA_LOCATION	data/churn_prototype	- +
HIVE_DATABASE	default	- +
HIVE_TABLE	churn_prototype_username	- +
STORAGE_MODE	local	- +
SHTM_ACCESS_KEY	m2owx6m4ciwvbgdk9d622ho0gr9nfh05	- +

Submit

## Use SSH Tunneling -

## Project Settings

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

### SSH Tunnels

SSH tunnels allow you to easily connect to firewalled resources such as databases or Hadoop clusters. They will be created automatically every time you launch a console.

+ New Tunnel

\* Name

\* Local Port

\* Username

\* Host

 Make sure your [user SSH key](#) has access to the server.

Endpoint

\* Remote Port

**Save**

## Data Connections -

## Project Settings

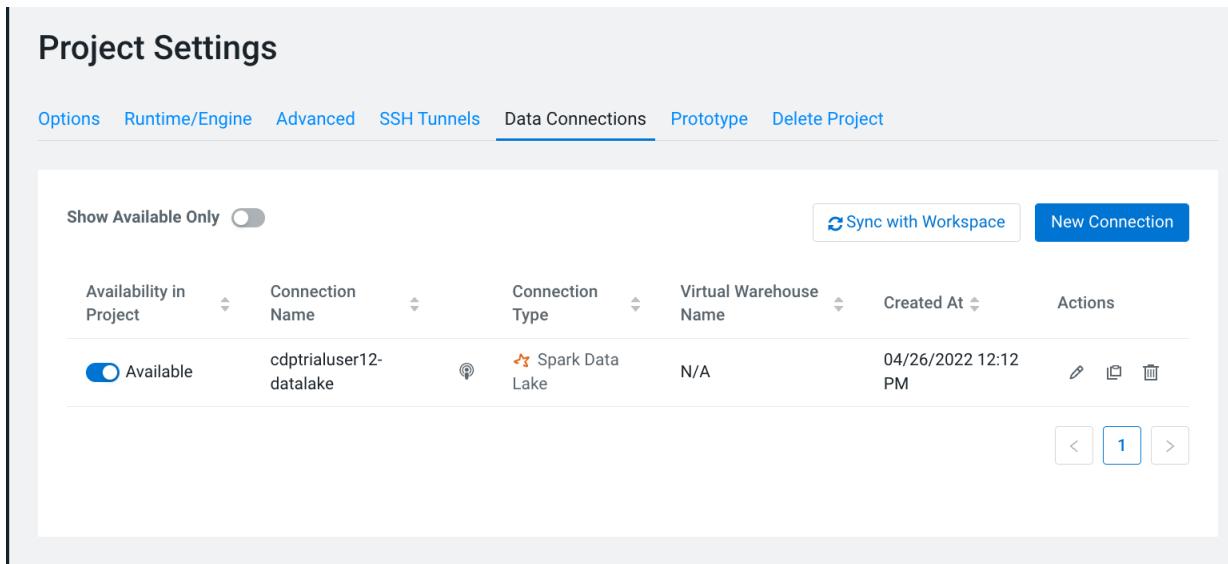
Options Runtime/Engine Advanced SSH Tunnels Data Connections Prototype Delete Project

Show Available Only

Sync with Workspace New Connection

Availability in Project	Connection Name	Connection Type	Virtual Warehouse Name	Created At	Actions
<input checked="" type="checkbox"/> Available	cdptrialuser12-datalake	Spark Data Lake	N/A	04/26/2022 12:12 PM	  

< 1 >



### 11. Lets see how to run some code

#### a. Running Code in Workbench

Sessions allow you to perform actions such as run R or Python code. They also provide access to an interactive command prompt and terminal. Sessions will be built on a specified Engine Image, which is a docker container that is deployed onto the Workspace. In addition you can specify how many resources are used per session. From the Overview page click on New Session. You can select the CPU, Memory and GPU for each session along with editors, and other addons.

## Start A New Session



### New version available

There is a new version of Base Image available. Latest engine image is: "Default engine image".

[Update Engine](#)

#### Session Name

Untitled Session

#### Runtime

##### Editor (i)

Workbench

##### Kernel (i)

Python 3.7

##### Edition (i)

Standard

##### Version

2021.12

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



Enable Spark (i)

Spark 2.4 - CDP 7.2.8 (TP)



#### Runtime Image

- docker.repository.cloudera.com/cloudera/cdsw/ml-runtime-workbench-python3.7-standard:2021.12.1-b17

#### Resource Profile

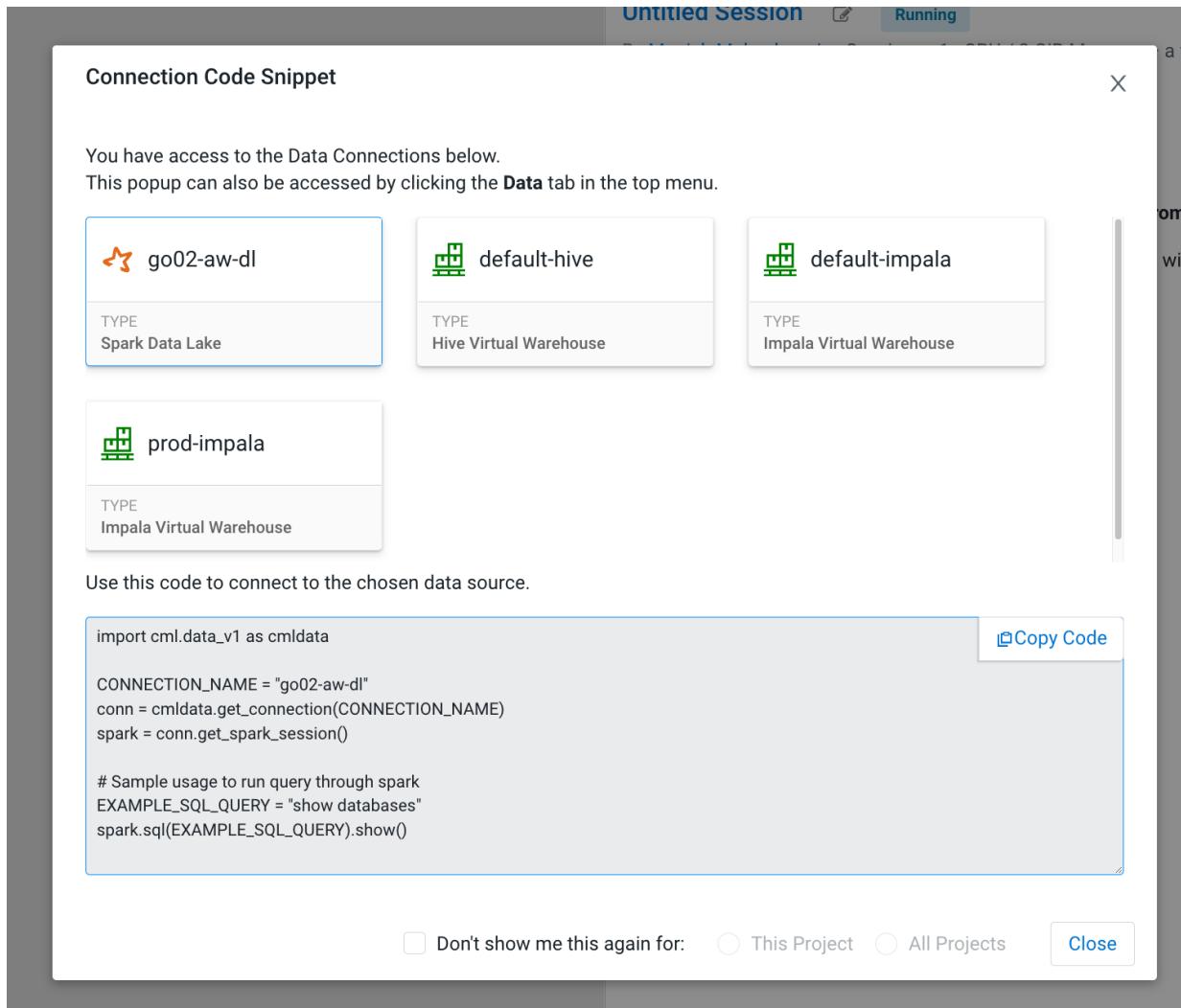
1 vCPU / 2 GiB Memory



0 GPUs



Connection snippets allow you to connect to “Cloudera Datawarehouse” to pull data from Impala/Hive. You can copy code as needed or just close the window.



Run the script of code/1\_data\_ingest.py and see how you can run code in Workbench.

The screenshot shows a Cloud Data Lake interface with two main panes. The left pane is a file browser showing a directory structure for a project named 'learn-trial12\_admin'. The right pane is a code editor for a Python script named 'code/1\_data\_ingest.py'.

```

File Edit View Navigate Run > code/1_data_ingest.py

1 ##### CLOUDERA APPLIED MACHINE LEARNING PROTOTYPE (AMP)
2 # (C) Cloudera, Inc. 2021
3 # All rights reserved.
4 #
5 # Applicable Open Source License: Apache 2.0
6 #
7 # Absent: Cloudera's open source products are modular software products
8 # made up of hundreds of individual components, each of which is a
9 # individually copyrighted. Each Cloudera open source product is a
10 # collective work under U.S. Copyright Law. Your license to use the
11 # product includes a license to copy and distribute the product in source
12 # form. If you distribute any part of this code, you must include changed
13 # source back to Cloudera. Used apart from the collective work, this file is
14 # licensed for your use pursuant to the open source license
15 # identified above.
16 #
17 # This code is provided to you pursuant to a written agreement with
18 # (1) Cloudera, Inc. or (2) a third-party authorized to distribute
19 # Cloudera's open source products. If you have written agreements with Cloudera
20 # or a third-party authorized to distribute Cloudera's open source products
21 # that provide warranties to you, you may not use this file without
22 # having any rights to access it or to use its code.
23 #
24 # Absent: A written agreement with Cloudera, Inc. ("Cloudera") to the
25 # contrary ("NOT CLOUDEA PROVIDES THIS CODE TO YOU WITHOUT WARRANTIES OF ANY
26 # KIND"), Cloudera, Inc. hereby grants you a limited, non-exclusive, non-
27 # transferable, revocable license to use the code, solely for purposes of
28 # IMPLIMENTATION OF CLAUSES (I), (II), (III), (IV), (V), (VI), (VII), (VIII), (IX),
29 # (X), (XI), (XII), (XIII), (XIV), (XV), (XVI), (XVII), (XVIII), (XIX), (XVII),
30 # (XVII), (XVII), (XVII), (XVII), (XVII), (XVII), (XVII), (XVII), (XVII),
31 # (XVII), (XVII), (XVII), (XVII), (XVII), (XVII), (XVII), (XVII), (XVII),
32 # (XVII), (XVII), (XVII), (XVII), (XVII), (XVII), (XVII), (XVII), (XVII),
33 # (XVII), (XVII), (XVII), (XVII), (XVII), (XVII), (XVII), (XVII), (XVII),
34 # (XVII), (XVII), (XVII), (XVII), (XVII), (XVII), (XVII), (XVII), (XVII),
35 # RELATED TO LOST REVENUE, LOST PROFITS, LOSS OF INCOME, LOSS OF
36 # BUSINESS ADVANTAGE OR UNAVAILABILITY, OR LOSS OR CORRUPTION OF
37 # DATA.
38 #
39 ##### Part 1: Data Ingest
40 # A data scientist should never be blocked in getting data into their environment,
41 # so CML is able to ingest data from many sources.
42 # When you have data in csv files, modern formats like parquet or feather, in
43 # cloud storage or a SQL database, CML will let you work with it in a data
44 # scientist-friendly environment.
45 #
46 # Access local data on your computer
47 #
48 # Accessing data stored on your computer is a matter of [uploading a file to the CML filesystem and
49 # referencing from there](https://docs.cloudera.com/machine-learning/cloud/import-data/topics/ml-accessing-1.html).
50 # > Go to the project's **Overview** page. Under the **Files** section, click **Upload**, select the relevant
51 # file, and click **Upload**. Then click **Run** to start streaming in the CML filesystem.
52 #
53 # If, for example, you upload a file called, `mydata.csv` to a folder called `data`, the
54 # following example code would work.
55 #
56 # ...
57 # import pandas as pd
58 # df = pd.read_csv('data/mydata.csv')
59 # # Q:
60 # df = pd.read_csv('/home/cdswe/data/mydata.csv')
61 #
62 # Access data in S3
63 #
64 # Accessing [data in Amazon S3](https://docs.cloudera.com/machine-learning/cloud/import-data/topics/ml-accessing-7.html).
65 # Follow a familiar procedure of [uploading a file to S3](https://docs.cloudera.com/machine-learning/cloud/import-data/topics/ml-accessing-1.html) and then
66 # referencing variables! [AWS_ACCESS_KEY_ID] and [AWS_SECRET_ACCESS_KEY].
67 #
68 # To get the the access keys that are used for you in the CDP DataLake, you can follow
69 # [this Cloudera Community Tutorial](https://community.cloudera.com/t5/Community-Articles/How-to-get-AWS-ac
70 #
71 # The following sample code would fetch a file called 'myfile.csv' from the S3 bucket, `data_bucket`, and st
72 #
73 # Create the Boto S3 connection object.
74 # from boto.s3.connection import S3Connection
75 # aws_connection = S3Connection()
76 #
77 #
78 #
79 # First we specify STORAGE as an environment variable in your project settings
80 # containing the Cloud Storage location used by the DataLake to store Hive data. On
81 # AWS it will be s3a://[something], on Azure it will be abfs://[something], and on
82 # on prem CDSW cluster, it will be hdfs://[something]
83 #
84 # This was done for you when you ran @bootstrap.py, so the following code is set up to run as is. It begins with
85 # imports and creating a SparkSession.
86 #
87 # import os
88 #
89 # Create the Boto S3 connection object.
90 # from boto.s3.connection import S3Connection
91 # aws_connection = S3Connection()
92 #
93 #
94 #
95 #
96 #
97 #
98 #
99 #
100 #
101 #
102 #
103 #
104 #
105 #
106 #
107 #
108 #
109 #
110 #
111 #
112 #
113 #
114 #
115 #
116 #
117 #
118 #
119 #
120 #
121 #
122 #
123 #
124 #
125 #
126 #
127 #
128 #
129 #
130 #
131 #
132 #
133 #
134 #
135 #
136 #
137 #
138 #
139 #
140 #
141 #
142 #
143 #
144 #
145 #
146 #
147 #
148 #
149 #
150 #
151 #
152 #
153 #
154 #
155 #
156 #
157 #
158 #
159 #
160 #
161 #
162 #
163 #
164 #
165 #
166 #
167 #
168 #
169 #
170 #
171 #
172 #
173 #
174 #
175 #
176 #
177 #
178 #
179 #
180 #
181 #
182 #
183 #
184 #
185 #
186 #
187 #
188 #
189 #
190 #
191 #
192 #
193 #
194 #
195 #
196 #
197 #
198 #
199 #
200 #
201 #
202 #
203 #
204 #
205 #
206 #
207 #
208 #
209 #
210 #
211 #
212 #
213 #
214 #
215 #
216 #
217 #
218 #
219 #
220 #
221 #
222 #
223 #
224 #
225 #
226 #
227 #
228 #
229 #
230 #
231 #
232 #
233 #
234 #
235 #
236 #
237 #
238 #
239 #
240 #
241 #
242 #
243 #
244 #
245 #
246 #
247 #
248 #
249 #
250 #
251 #
252 #
253 #
254 #
255 #
256 #
257 #
258 #
259 #
259 #
260 #
261 #
262 #
263 #
264 #
265 #
266 #
267 #
268 #
269 #
270 #
271 #
272 #
273 #
274 #
275 #
276 #
277 #
278 #
279 #
280 #
281 #
282 #
283 #
284 #
285 #
286 #
287 #
288 #
289 #
290 #
291 #
292 #
293 #
294 #
295 #
296 #
297 #
298 #
299 #
299 #
300 #
301 #
302 #
303 #
304 #
305 #
306 #
307 #
308 #
309 #
309 #
310 #
311 #
312 #
313 #
314 #
315 #
316 #
317 #
318 #
319 #
319 #
320 #
321 #
322 #
323 #
324 #
325 #
326 #
327 #
328 #
329 #
329 #
330 #
331 #
332 #
333 #
334 #
335 #
336 #
337 #
338 #
339 #
339 #
340 #
341 #
342 #
343 #
344 #
345 #
346 #
347 #
348 #
349 #
349 #
350 #
351 #
352 #
353 #
354 #
355 #
356 #
357 #
358 #
359 #
359 #
360 #
361 #
362 #
363 #
364 #
365 #
366 #
367 #
368 #
369 #
369 #
370 #
371 #
372 #
373 #
374 #
375 #
376 #
377 #
378 #
379 #
379 #
380 #
381 #
382 #
383 #
384 #
385 #
386 #
387 #
388 #
389 #
389 #
390 #
391 #
392 #
393 #
394 #
395 #
396 #
397 #
398 #
399 #
399 #
400 #
401 #
402 #
403 #
404 #
405 #
406 #
407 #
408 #
409 #
409 #
410 #
411 #
412 #
413 #
414 #
415 #
416 #
417 #
418 #
419 #
419 #
420 #
421 #
422 #
423 #
424 #
425 #
426 #
427 #
428 #
429 #
429 #
430 #
431 #
432 #
433 #
434 #
435 #
436 #
437 #
438 #
439 #
439 #
440 #
441 #
442 #
443 #
444 #
445 #
446 #
447 #
448 #
449 #
449 #
450 #
451 #
452 #
453 #
454 #
455 #
456 #
457 #
458 #
459 #
459 #
460 #
461 #
462 #
463 #
464 #
465 #
466 #
467 #
468 #
469 #
469 #
470 #
471 #
472 #
473 #
474 #
475 #
476 #
477 #
478 #
479 #
479 #
480 #
481 #
482 #
483 #
484 #
485 #
486 #
487 #
488 #
489 #
489 #
490 #
491 #
492 #
493 #
494 #
495 #
496 #
497 #
498 #
499 #
499 #
500 #
501 #
502 #
503 #
504 #
505 #
506 #
507 #
508 #
509 #
509 #
510 #
511 #
512 #
513 #
514 #
515 #
516 #
517 #
518 #
519 #
519 #
520 #
521 #
522 #
523 #
524 #
525 #
526 #
527 #
528 #
529 #
529 #
530 #
531 #
532 #
533 #
534 #
535 #
536 #
537 #
538 #
539 #
539 #
540 #
541 #
542 #
543 #
544 #
545 #
546 #
547 #
548 #
549 #
549 #
550 #
551 #
552 #
553 #
554 #
555 #
556 #
557 #
558 #
559 #
559 #
560 #
561 #
562 #
563 #
564 #
565 #
566 #
567 #
568 #
569 #
569 #
570 #
571 #
572 #
573 #
574 #
575 #
576 #
577 #
578 #
579 #
579 #
580 #
581 #
582 #
583 #
584 #
585 #
586 #
587 #
588 #
589 #
589 #
590 #
591 #
592 #
593 #
594 #
595 #
596 #
597 #
598 #
599 #
599 #
600 #
601 #
602 #
603 #
604 #
605 #
606 #
607 #
608 #
609 #
609 #
610 #
611 #
612 #
613 #
614 #
615 #
616 #
617 #
618 #
619 #
619 #
620 #
621 #
622 #
623 #
624 #
625 #
626 #
627 #
628 #
629 #
629 #
630 #
631 #
632 #
633 #
634 #
635 #
636 #
637 #
638 #
639 #
639 #
640 #
641 #
642 #
643 #
644 #
645 #
646 #
647 #
648 #
649 #
649 #
650 #
651 #
652 #
653 #
654 #
655 #
656 #
657 #
658 #
659 #
659 #
660 #
661 #
662 #
663 #
664 #
665 #
666 #
667 #
668 #
669 #
669 #
670 #
671 #
672 #
673 #
674 #
675 #
676 #
677 #
678 #
679 #
679 #
680 #
681 #
682 #
683 #
684 #
685 #
686 #
687 #
688 #
689 #
689 #
690 #
691 #
692 #
693 #
694 #
695 #
696 #
697 #
697 #
698 #
699 #
699 #
700 #
701 #
702 #
703 #
704 #
705 #
706 #
707 #
708 #
709 #
709 #
710 #
711 #
712 #
713 #
714 #
715 #
716 #
717 #
718 #
719 #
719 #
720 #
721 #
722 #
723 #
724 #
725 #
726 #
727 #
728 #
729 #
729 #
730 #
731 #
732 #
733 #
734 #
735 #
736 #
737 #
738 #
739 #
739 #
740 #
741 #
742 #
743 #
744 #
745 #
746 #
747 #
748 #
749 #
749 #
750 #
751 #
752 #
753 #
754 #
755 #
756 #
757 #
758 #
759 #
759 #
760 #
761 #
762 #
763 #
764 #
765 #
766 #
767 #
768 #
769 #
769 #
770 #
771 #
772 #
773 #
774 #
775 #
776 #
777 #
778 #
779 #
779 #
780 #
781 #
782 #
783 #
784 #
785 #
786 #
787 #
788 #
789 #
789 #
790 #
791 #
792 #
793 #
794 #
795 #
796 #
797 #
798 #
799 #
799 #
800 #
801 #
802 #
803 #
804 #
805 #
806 #
807 #
808 #
809 #
809 #
810 #
811 #
812 #
813 #
814 #
815 #
816 #
817 #
818 #
819 #
819 #
820 #
821 #
822 #
823 #
824 #
825 #
826 #
827 #
828 #
829 #
829 #
830 #
831 #
832 #
833 #
834 #
835 #
836 #
837 #
838 #
839 #
839 #
840 #
841 #
842 #
843 #
844 #
845 #
846 #
847 #
848 #
849 #
849 #
850 #
851 #
852 #
853 #
854 #
855 #
856 #
857 #
858 #
859 #
859 #
860 #
861 #
862 #
863 #
864 #
865 #
866 #
867 #
868 #
869 #
869 #
870 #
871 #
872 #
873 #
874 #
875 #
876 #
877 #
878 #
879 #
879 #
880 #
881 #
882 #
883 #
884 #
885 #
886 #
887 #
888 #
889 #
889 #
890 #
891 #
892 #
893 #
894 #
895 #
896 #
897 #
898 #
899 #
899 #
900 #
901 #
902 #
903 #
904 #
905 #
906 #
907 #
908 #
909 #
909 #
910 #
911 #
912 #
913 #
914 #
915 #
916 #
917 #
918 #
919 #
919 #
920 #
921 #
922 #
923 #
924 #
925 #
926 #
927 #
928 #
929 #
929 #
930 #
931 #
932 #
933 #
934 #
935 #
936 #
937 #
938 #
939 #
939 #
940 #
941 #
942 #
943 #
944 #
945 #
946 #
947 #
948 #
949 #
949 #
950 #
951 #
952 #
953 #
954 #
955 #
956 #
957 #
958 #
959 #
959 #
960 #
961 #
962 #
963 #
964 #
965 #
966 #
967 #
968 #
969 #
969 #
970 #
971 #
972 #
973 #
974 #
975 #
976 #
977 #
978 #
979 #
979 #
980 #
981 #
982 #
983 #
984 #
985 #
986 #
987 #
988 #
989 #
989 #
990 #
991 #
992 #
993 #
994 #
995 #
996 #
997 #
998 #
999 #
999 #
1000 #
1001 #
1002 #
1003 #
1004 #
1005 #
1006 #
1007 #
1008 #
1009 #
1009 #
1010 #
1011 #
1012 #
1013 #
1014 #
1015 #
1016 #
1017 #
1018 #
1019 #
1019 #
1020 #
1021 #
1022 #
1023 #
1024 #
1025 #
1026 #
1027 #
1028 #
1029 #
1029 #
1030 #
1031 #
1032 #
1033 #
1034 #
1035 #
1036 #
1037 #
1038 #
1039 #
1039 #
1040 #
1041 #
1042 #
1043 #
1044 #
1045 #
1046 #
1047 #
1048 #
1049 #
1049 #
1050 #
1051 #
1052 #
1053 #
1054 #
1055 #
1056 #
1057 #
1058 #
1059 #
1059 #
1060 #
1061 #
1062 #
1063 #
1064 #
1065 #
1066 #
1067 #
1068 #
1069 #
1069 #
1070 #
1071 #
1072 #
1073 #
1074 #
1075 #
1076 #
1077 #
1078 #
1079 #
1079 #
1080 #
1081 #
1082 #
1083 #
1084 #
1085 #
1086 #
1087 #
1088 #
1089 #
1089 #
1090 #
1091 #
1092 #
1093 #
1094 #
1095 #
1096 #
1097 #
1098 #
1099 #
1099 #
1100 #
1101 #
1102 #
1103 #
1104 #
1105 #
1106 #
1107 #
1108 #
1109 #
1109 #
1110 #
1111 #
1112 #
1113 #
1114 #
1115 #
1116 #
1117 #
1118 #
1119 #
1119 #
1120 #
1121 #
1122 #
1123 #
1124 #
1125 #
1126 #
1127 #
1128 #
1129 #
1129 #
1130 #
1131 #
1132 #
1133 #
1134 #
1135 #
1136 #
1137 #
1138 #
1139 #
1139 #
1140 #
1141 #
1142 #
1143 #
1144 #
1145 #
1146 #
1147 #
1148 #
1149 #
1149 #
1150 #
1151 #
1152 #
1153 #
1154 #
1155 #
1156 #
1157 #
1158 #
1159 #
1159 #
1160 #
1161 #
1162 #
1163 #
1164 #
1165 #
1166 #
1167 #
1168 #
1169 #
1169 #
1170 #
1171 #
1172 #
1173 #
1174 #
1175 #
1176 #
1177 #
1178 #
1179 #
1179 #
1180 #
1181 #
1182 #
1183 #
1184 #
1185 #
1186 #
1187 #
1188 #
1189 #
1189 #
1190 #
1191 #
1192 #
1193 #
1194 #
1195 #
1196 #
1197 #
1198 #
1199 #
1199 #
1200 #
1201 #
1202 #
1203 #
1204 #
1205 #
1206 #
1207 #
1208 #
1209 #
1209 #
1210 #
1211 #
1212 #
1213 #
1214 #
1215 #
1216 #
1217 #
1218 #
1219 #
1219 #
1220 #
1221 #
1222 #
1223 #
1224 #
1225 #
1226 #
1227 #
1228 #
1229 #
1229 #
1230 #
1231 #
1232 #
1233 #
1234 #
1235 #
1236 #
1237 #
1238 #
1239 #
1239 #
1240 #
1241 #
1242 #
1243 #
1244 #
1245 #
1246 #
1247 #
1248 #
1249 #
1249 #
1250 #
1251 #
1252 #
1253 #
1254 #
1255 #
1256 #
1257 #
1258 #
1259 #
1259 #
1260 #
1261 #
1262 #
1263 #
1264 #
1265 #
1266 #
1267 #
1268 #
1269 #
1269 #
1270 #
1271 #
1272 #
1273 #
1274 #
1275 #
1276 #
1277 #
1278 #
1279 #
1279 #
1280 #
1281 #
1282 #
1283 #
1284 #
1285 #
1286 #
1287 #
1288 #
1289 #
1289 #
1290 #
1291 #
1292 #
1293 #
1294 #
1295 #
1296 #
1297 #
1298 #
1299 #
1299 #
1300 #
1301 #
1302 #
1303 #
1304 #
1305 #
1306 #
1307 #
1308 #
1309 #
1309 #
1310 #
1311 #
1312 #
1313 #
1314 #
1315 #
1316 #
1317 #
1318 #
1319 #
1319 #
1320 #
1321 #
1322 #
1323 #
1324 #
1325 #
1326 #
1327 #
1328 #
1329 #
1329 #
1330 #
1331 #
1332 #
1333 #
1334 #
1335 #
1336 #
1337 #
1338 #
1339 #
1339 #
1340 #
1341 #
1342 #
1343 #
1344 #
1345 #
1346 #
1347 #
1348 #
1349 #
1349 #
1350 #
1351 #
1352 #
1353 #
1354 #
1355 #
1356 #
1357 #
1358 #
1359 #
1359 #
1360 #
1361 #
1362 #
1363 #
1364 #
1365 #
1366 #
1367 #
1368 #
1369 #
1369 #
1370 #
1371 #
1372 #
1373 #
1374 #
1375 #
1376 #
1377 #
1378 #
1379 #
1379 #
1380 #
1381 #
1382 #
1383 #
1384 #
1385 #
1386 #
1387 #
1388 #
1389 #
1389 #
1390 #
1391 #
1392 #
1393 #
1394 #
1395 #
1396 #
1397 #
1398 #
1399 #
1399 #
1400 #
1401 #
1402 #
1403 #
1404 #
1405 #
1406 #
1407 #
1408 #
1409 #
1409 #
1410 #
1411 #
1412 #
1413 #
1414 #
1415 #
1416 #
1417 #
1418 #
1419 #
1419 #
1420 #
1421 #
1422 #
1423 #
1424 #
1425 #
1426 #
1427 #
1428 #
1429 #
1429 #
1430 #
1431 #
1432 #
1433 #
1434 #
1435 #
1436 #
1437 #
1438 #
1439 #
1439 #
1440 #
1441 #
1442 #
1443 #
1444 #
1445 #
1446 #
1447 #
1448 #
1449 #
1449 #
1450 #
1451 #
1452 #
1453 #
1454 #
1455 #
1456 #
1457 #
1458 #
1459 #
1459 #
1460 #
1461 #
1462 #
1463 #
1464 #
1465 #
1466 #
1467 #
1468 #
1469 #
1469 #
1470 #
1471 #
1472 #
1473 #
1474 #
1475 #
1476 #
1477 #
1478 #
1479 #
1479 #
1480 #
1481 #
1482 #
1483 #
1484 #
1485 #
1486 #
1487 #
1488 #
1489 #
1489 #
1490 #
1491 #
1492 #
1493 #
1494 #
1495 #
1496 #
1497 #
1498 #
1499 #
1499 #
1500 #
1501 #
1502 #
1503 #
1504 #
1505 #
1506 #
1507 #
1508 #
1509 #
1509 #
1510 #
1511 #
1512 #
1513 #
1514 #
1515 #
1516 #
1517 #
1518 #
1519 #
1519 #
1520 #
1521 #
1522 #
1523 #
1524 #
1525 #
1526 #
1527 #
1528 #
1529 #
1529 #
1530 #
1531 #
1532 #
1533 #
1534 #
1535 #
1536 #
1537 #
1538 #
1539 #
1539 #
1540 #
1541 #
1542 #
1543 #
1544 #
1545 #
1546 #
1547 #
1548 #
1549 #
1549 #
1550 #
1551 #
1552 #
1553 #
1554 #
1555 #
1556 #
1557 #
1558 #
1559 #
1559 #
1560 #
1561 #
1562 #
1563 #
1564 #
1565 #
1566 #
1567 #
1568 #
1569 #
1569 #
1570 #
1571 #
1572 #
1573 #
1574 #
1575 #
1576 #
1577 #
1578 #
1579 #
1579 #
1580 #
1581 #
1582 #
1583 #
1584 #
1585 #
1586 #
1587 #
1588 #
1589 #
1589 #
1590 #
1591 #
1592 #
1593 #
1594 #
1595 #
1596 #
1597 #
1598 #
1599 #
1599 #
1600 #
1601 #
1602 #
1603 #
1604 #
1605 #
1606 #
1607 #
1608 #
1609 #
1609 #
1610 #
1611 #
1612 #
1613 #
1614 #
1615 #
1616 #
1617 #
1618 #
1619 #
1619 #
1620 #
1621 #
1622 #
1623 #
1624 #
1625 #
1626 #
1627 #
1628 #
1629 #
1629 #
1630 #
1631 #
1632 #
1633 #
1634 #
1635 #
1636 #
1637 #
1638 #
1639 #
1639 #
1640 #
1641 #
1642 #
1643 #
1644 #
1645 #
1646 #
1647 #
1648 #
1649 #
1649 #
1650 #
1651 #
1652 #
1653 #
1654 #
1655 #
1656 #
1657 #
1658 #
1659 #
1659 #
1660 #
1661 #
1662 #
1663 #
1664 #
1665 #
1666 #
1667 #
1668 #
1669 #
1669 #
1670 #
1671 #
1672 #
1673 #
1674 #
1675 #
1676 #
1677 #
1678 #
1679 #
1679 #
1680 #
1681 #
1682 #
1683 #
1684 #
1685 #
1686 #
1687 #
1688 #
1689 #
1689 #
1690 #
1691 #
1692 #
1693 #
1694 #
1695 #
1696 #
1697 #
1698 #
1699 #
1699 #
1700 #
1701 #
1702 #
1703 #
1704 #
1705 #
1706 #
1707 #
1708 #
1709 #
1709 #
1710 #
1711 #
1712 #
1713 #
1714 #
1715 #
1716 #
1717 #
1718 #
1719 #
1719 #
1720 #
1721 #
1722 #
1723 #
1724 #
1725 #
1726 #
1727 #
1728 #
1729 #
1729 #
1730 #
1731 #
1732 #
1733 #
1734 #
1735 #
1736 #
1737 #
1738 #
1739 #
1739 #
1740 #
1741 #
1742 #
1743 #
1744 #
1745 #
1746 #
1747 #
1748 #
1749 #
1749 #
1750 #
1751 #
1752 #
1753 #
1754 #
1755 #
1756 #
1757 #
1758 #
1759 #
1759 #
1760 #
1761 #
1762 #
1763 #
1764 #
1765 #
1766 #
1767 #
1768 #
1769 #
1769 #
1770 #
1771 #
1772 #
1773 #
1774 #
1775 #
1776 #
1777 #
1778 #
1779 #
1779 #
1780 #
1781 #
1782 #
1783 #
1784 #
1785 #
1786 #
1787 #
1788 #
1789 #
1789 #
1790 #
1791 #
1792 #
1793 #
1794 #
1795 #
1796 #
1797 #
1798 #
1799 #
1799 #
1800 #
1801 #
1802 #
1803 #
1804 #
1805 #
1806 #
1807 #
1808 #
1809 #
1809 #
1810 #
1811 #
1812 #
1813 #
1814 #
1815 #
1816 #
1817 #
1818 #
1819 #
1819 #
1820 #
1821 #
1822 #
1823 #
1824 #
1825 #
1826 #
1827 #
1828 #
1829 #
1829 #
1830 #
1831 #
1832 #
1833 #
1834 #
1835 #
1836 #
1837 #
1838 #
1839 #
1839 #
1840 #
1841 #
1842 #
1843 #
1844 #
1845 #
1846 #
1847 #
1848 #
1849 #
1849 #
1850 #
1851 #
1852 #
1853 #
1854 #
1855 #
1856 #
1857 #
1858 #
1859 #
1859 #
1860 #
1861 #
1862 #
1863 #
1864 #
1865 #
1866 #
1867 #
1868 #
1869 #
1869 #
1870 #
1871 #
1872 #
1873 #
1874 #
1875 #
1876 #
1877 #
1878 #
1878 #
1879 #
1880 #
1881 #
1882 #
1883 #
1884 #
1885 #
1886 #
1887 #
1888 #
1889 #
1889 #
1890 #
1891 #
1892 #
1893 #
1894 #
1895 #
1896 #
1897 #
1898 #
1899 #
1899 #
1900 #
1901 #
1902 #
1903 #
1904 #
1905 #
1906 #
1907 #
1908 #
1909 #
1909 #
1910 #
1911 #
1912 #
1913 #
1914 #
1915 #
1916 #
1917 #
1918 #
1919 #
1919 #
1920 #
1921 #
1922 #
1923 #
1924 #
1925 #
1926 #
1927 #
1928 #
1929 #
1929 #
1930 #
1931 #
1932 #
1933 #
1934 #
1935 #
1936 #
1937 #
1938 #
1939 #
1939 #
1940 #
1941 #
1942 #
1943 #
1944 #
1945 #
1946 #
1947 #
1948 #
1949 #
1949 #
1950 #
1951 #
1952 #
1953 #
1954 #
1955 #
1956 #
1957 #
1958 #
1959 #
1959 #
1960 #
1961 #
1962 #
1963 #
1964 #
1965 #
1966 #
1967 #
1968 #
1969 #
1969 #
1970 #
1971 #
1972 #
1973 #
1974 #
1975 #
1976 #
1977 #
1978 #
1978 #
1979 #
1980 #
1981 #
1982 #
1983 #
1984 #
1985 #
1986 #
1987 #
1988 #
1989 #
1989 #
1990 #
1991 #
1992 #
1993 #
1994 #
1995 #
1996 #
1997 #
1998 #
1999 #
1999 #
2000 #
2001 #
2002 #
2003 #
2004 #
2005 #
2006 #
2007 #
2008 #
2009 #
2010 #
2011 #
2012 #
2013 #
2014 #
2015 #
2016 #
2017 #
2018 #
2019 #
2020 #
2021 #
2022 #
2023 #
2024 #
2025 #
2026 #
2027 #
2028 #
2029 #
2030 #
2031 #
2032 #
2033 #
2034 #
2035 #
2036 #
2037 #
2038 #
2039 #
2040 #
2041 #
2042 #
2043 #
2044 #
2045 #
2046
```

**b. Running Code in Jupyter Lab**

The screenshot shows the Cloudera Data Science Workbench interface. On the left, there is a file tree view:

- Churn Modeling with scikit-learn - trial12\_admin 1
- .project-metadata.yaml
- cdsw-build.sh
- code
  - 0\_bootstrap.py
  - 1\_data\_ingest.py
  - 2\_data\_exploration.ipynb
  - 3\_model\_building.ipynb
  - 4\_train\_models.py
  - 5\_model\_serve\_explainer.py
  - 6\_application.py
  - 7a\_ml\_ops\_simulation.py
  - 7b\_ml\_ops\_visual.py
  - \_\_pycache\_\_
    - churnexplainer.py
  - README.md
- flask
- images
- LICENSE.txt
- lineage.yml
- model\_metrics.db
- models
- raw
- README.md
- requirements.txt

On the right, a "Start A New Session" dialog is open:

- Session Name:** Untitled Session
- Runtime:**
  - Editor:** JupyterLab
  - Kernel:** Python 3.7
- Edition:** Standard
- Version:** 2021.12
- Runtime Image:** docker.repository.cloudera.com/cloudera/cdsw/ml-runtime-jupyterlab-python3.7-standard:2021.12.1-b17
- Resource Profile:** 2 vCPU / 4 GiB Memory

The screenshot shows the Cloudera Machine Learning interface. On the left, there's a file browser window showing several Python files and a README.md file. The main area is a Jupyter Notebook cell containing code for data exploration. The notebook title is '2\_data\_exploration.ipynb'. The code uses SparkSession to read data from Hive or CSV files and prints the first few rows of the resulting DataFrame.

```

# Part 2: Data Exploration

This notebook does some basic exploratory data analysis (EDA) of the telco churn data. Here, we get a qualitative and a quantitative sense of the data, what kind of cleanup it might need before we can use it, and if there are any specific patterns that can be discerned.

If you haven't yet, run through the initialization steps in the README file and Part 1. In Part 1, the data is imported into the table you specified in Hive. All new data accesses will fetch from Hive if available, otherwise the local copy of the data will be used.

Load data

We start by creating a SparkSession to fetch the data using Spark SQL.

[62]:
```

```

import os
from pyspark.sql import SparkSession
from pyspark.sql.types import *

hive_database = os.environ['HIVE_DATABASE']
hive_table = os.environ['HIVE_TABLE']
hive_table_fq = hive_database + '.' + hive_table

# connect to Spark
spark = SparkSession\
    .builder\
    .appName("Telco Data Set")\
    .master("local[*]")\
    .getOrCreate()

# read data into a Spark DataFrame
if os.environ["STORAGE_MODE"] == "external":
    telco_data_raw = spark.sql("SELECT * FROM " + hive_table_fq)
else:
    path = "/home/cdsu/raw/WA_Fn-UseC-Telco-Customer-Churn.csv"
    schema = StructType([
        StructField("customerID", StringType(), True),
        StructField("gender", StringType(), True),
        StructField("SeniorCitizen", StringType(), True),
        StructField("Partner", StringType(), True),
        StructField("Dependents", StringType(), True),
        StructField("PhoneService", StringType(), True),
        StructField("MultipleLines", StringType(), True),
        StructField("InternetService", StringType(), True),
        StructField("OnlineSecurity", StringType(), True),
        StructField("OnlineBackup", StringType(), True),
        StructField("DeviceProtection", StringType(), True),
        StructField("TechSupport", StringType(), True),
        StructField("StreamingTV", StringType(), True),
        StructField("StreamingMovies", StringType(), True),
        StructField("Contract", StringType(), True),
        StructField("PaperlessBilling", StringType(), True),
        StructField("MonthlyCharges", DoubleType(), True),
        StructField("TotalCharges", DoubleType(), True),
        StructField("Churn", StringType(), True),
    ])
    telco_data_raw = spark.read.csv(path, header=True, sep=",", schema=schema, nullValue="NA")

# print a few rows
telco_data_raw.toPandas().head()

```

[62]:

customerID	gender	SeniorCitizen	Partner	Dependents	tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	... DeviceProtection	TechSupport	StreamingTV	StreamingMovies

## 12. Creating Jobs

Lets see how you can create and schedule jobs.

trial12\_admin / Churn Modeling with scikit-learn - trial12\_admin 3 / Jobs / New Job

### Create a Job

#### General

##### Name

Run Data Ingestion

##### Script

code/1\_data\_ingest.py



##### Arguments

Arguments

##### Runtime

###### Editor ⓘ

###### Kernel ⓘ

JupyterLab

Python 3.7

###### Edition ⓘ

###### Version

Standard

2021.12

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



Enable Spark  ⓘ

Spark 2.4.7 - CDP 7.2.11 - CDE 1.13 - HO...



##### Runtime Image

- docker.repository.cloudera.com/cloudera/cdsw/ml-runtime-jupyterlab-python3.7-standard:2021.12.1-b17

#### Schedule

Manual



#### Resource Profile

2 vCPU / 4 GiB Memory



Timeout In Minutes (optional)

Kill on Timeout

Jobs exceeding timeout send warning email if notifications enabled.

#### Environment Variables

Name	Value	Actions
<input type="text"/>	<input type="text"/>	<a href="#">Add</a>

Environment variables will override the [project environment](#).

## Jobs

New Job

Job Dependencies for Run Data Ingestion

Run Data Ingestion

+ Add Job Dependency

Creator ▾

Name	Runs / Failures	Duration	Status	Latest Run	Actions
Run Data Ingestion	1 / 0	00:00	Scheduling	a few seconds ago	<button>Stop</button>

## Look at Job History

### Run Data Ingestion

Running Stop

Overview [History](#) [Dependencies](#) [Settings](#)

Script: `code/1_data_ingest.py`  
Schedule: Manual  
Resource Profile: 2 vCPU / 4 GiB Memory  
Created By: trial12\_admin  
Job Id: kpmv-6cbu-died-6cm0

Latest Run: a few seconds ago  
Duration: 00:06  
Runs: 1  
Failures: 0

#### Job History

Duration (s)

18.4  
17.8  
17.2

Apr 25 18:00 Apr 26 00:00 Apr 26 06:00 Apr 26 12:00 Apr 26 18:00 Apr 27 00:00 Apr 27 06:00 Apr 27 12:00

**Look and share Job Output for each run -**

**Run Data Ingestion**  Success

By trial12\_admin – Session – 2 vCPU / 4 GiB Memory – a minute ago [See job details](#)

Session Logs Spark UI    Export PDF

```
#####
CLOUDERA APPLIED MACHINE LEARNING PROTOTYPE AMP C Cloudera, Inc. 2021 All rights reserved.
Applicable Open Source License: Apache 2.0
NOTE: Cloudera open source products are modular software products made up of hundreds of individual components, each of which was
individually copyrighted. Each Cloudera open source product is a collective work under U.S. Copyright Law. Your license to use the
collective work is as provided in your written agreement with Cloudera. Used apart from the collective work, this file is licensed for your
use pursuant to the open source license identified above.
This code is provided to you pursuant a written agreement with i Cloudera, Inc. or ii a third-party authorized to distribute this code. If you
do not have a written agreement with Cloudera nor with an authorized and properly licensed third party, you do not have any rights to
access nor to use this code.
Absent a written agreement with Cloudera, Inc. “Cloudera” to the contrary, A) CLOUDERA PROVIDES THIS CODE TO YOU WITHOUT
WARRANTIES OF ANY KIND; B) CLOUDERA DISCLAIMS ANY AND ALL EXPRESS AND IMPLIED WARRANTIES WITH RESPECT TO THIS
CODE, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF TITLE, NON-INFRINGEMENT, MERCHANTABILITY AND FITNESS FOR A
PARTICULAR PURPOSE; C) CLOUDERA IS NOT LIABLE TO YOU, AND WILL NOT DEFEND, INDEMNIFY, NOR HOLD YOU HARMLESS FOR ANY
CLAIMS ARISING FROM OR RELATED TO THE CODE; AND D) WITH RESPECT TO YOUR EXERCISE OF ANY RIGHTS GRANTED TO YOU FOR
THE CODE, CLOUDERA IS NOT LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, PUNITIVE OR CONSEQUENTIAL
DAMAGES INCLUDING, BUT NOT LIMITED TO, DAMAGES RELATED TO LOST REVENUE, LOST PROFITS, LOSS OF INCOME, LOSS OF
BUSINESS ADVANTAGE OR UNAVAILABILITY, OR LOSS OR CORRUPTION OF DATA.
#####
Part 1: Data Ingest A data scientist should never be blocked in getting data into their environment, so CML is able to ingest data from
many sources. Whether you have data in .csv files, modern formats like parquet or feather, in cloud storage or a SQL database, CML will let
you work with it in a data scientist-friendly environment.
```

Access local data on your computer

Accessing data stored on your computer is a matter of [uploading a file to the CML filesystem and referencing from there](#).

Go to the project's **Overview** page. Under the **Files** section, click **Upload**, select the relevant data files to be uploaded and a destination folder.

If, for example, you upload a file called, `mydata.csv` to a folder called `data`, the following example code would work.

```
import pandas as pd
df = pd.read_csv('data/mydata.csv')
# Or:
df = pd.read_csv('/home/cdsw/data/mydata.csv')
```

Access data in S3

Accessing [data in Amazon S3](#) follows a familiar procedure of fetching and storing in the CML filesystem.

Add your Amazon Web Services access keys to your project's [environment variables](#) as `AWS_ACCESS_KEY_ID` and `AWS_SECRET_ACCESS_KEY`.

To get the the access keys that are used for you in the CDP DataLake, you can follow [this Cloudera Community Tutorial](#)

The following sample code would fetch a file called `myfile.csv` from the S3 bucket, `data_bucket`, and store it in the CML home folder.

```
# Create the Boto S3 connection object.
from boto.s3.connection import S3Connection
```

## Share job output and export PDF's -

The screenshot shows a 'Job details' page with a sharing panel open. The panel title is 'These results are being shared.' It contains a sharing link: <https://ml-3d71138b-2f0.cdp>. There is a 'Stop Sharing' button and an unchecked checkbox for 'Hide code and text'. Below this, under 'Who can view:', there are two radio button options: 'Any logged in user with the link' (unchecked) and 'Specific users/teams with the link (Change...)' (checked). A note below says 'Currently shared with no one.'

### 13. Models

Click on the deployed model in the project and click on Test.

The screenshot shows the 'Models' page. At the top right is a 'New Model' button. The main area is a table with columns: Model, Status, Replicas, CPU, Memory, Created By, Deployed By, Last Deployed, and Actions. One row is visible for 'Churn Model API Endpoint', which is 'Deployed' with 1/1 replicas, 1 CPU, 2.00 GiB memory, created by 'trial12\_admin' and deployed by 'trial12\_admin' on 'Apr 26, 2022, 12:19 PM'. The 'Actions' column for this row includes a 'Stop' button and a dropdown menu.

The model executes and returns the results. Models can be used for various use cases to provide an authenticated and access controlled rest api for other applications to send their data to scoring.

**Churn Model API Endpoint**

Overview Deployments Builds Monitoring Logs Settings

Description This model API endpoint is used to predict churn

Sample Code

```
curl -H "Content-Type: application/json" -X POST https://modelservice.ml-3d71138b-2f0.cdptrial.d06t-87n6.cloudera.site/model -d '{"accessKey": "m20wx6m4ciwvbgdk9d622ho6gr9nfh85", "request": {"StreamingTV": "No", "MonthlyCharges": 70.35, "PhoneService": "No", "PaperlessBilling": "No", "Partner": "No", "OnlineBackup": "No", "gender": "Female", "Contract": "Month-to-month", "TotalCharges": 1397.475, "StreamingMovies": "No", "DeviceProtection": "No", "PaymentMethod": "Bank transfer (automatic)", "tenure": 29, "Dependents": "No", "OnlineSecurity": "No", "MultipleLines": "No", "InternetService": "DSL", "SeniorCitizen": "No", "TechSupport": "No"}'}
```

**Test Model**

Input

```
{"OnlineSecurity": "No",
"MultipleLines": "No",
"InternetService": "DSL",
"SeniorCitizen": "No",
"TechSupport": "No"
}
```

Test Reset

**Result**

Status	success
	<pre>{   "model_deployment_crn": "crn:cdp:ml:us-west-1:da6fc083-7c10-494c-85b5-d366e0236f17:workspace:3d726a00-e1e1-4814-8c02-3ffc2431df9b/92aef483-3d95-4365-9158-05e2e   "prediction": {     "data": {       "Contract": "Month-to-month",       "Dependents": "No",       "DeviceProtection": "No",       "InternetService": "DSL",       "MonthlyCharges": 70.35,       "MultipleLines": "No",       "OnlineBackup": "No",       "OnlineSecurity": "No",       "PaperlessBilling": "No",       "Partner": "No",       "PaymentMethod": "Bank transfer (automatic)",       "SeniorCitizen": "No",       "TechSupport": "No"     }   } }</pre>

**Model Details**

Model Id	3
Model CRN	crn:cdp:ml:us-west-1:da6fc083-7c10-494c-85b5-d366e0236f17:workspace:3d726a00-e1e1-4814-8c02-3ffc2431df9b/814e6441-f17e-42cd-855d-dd0f2ee1ea10
Deployment Id	3
Deployment CRN	crn:cdp:ml:us-west-1:da6fc083-7c10-494c-85b5-d366e0236f17:workspace:3d726a00-e1e1-4814-8c02-3ffc2431df9b/92aef483-3d95-4365-9158-05e2e7bd054a
Build Id	3
Build CRN	crn:cdp:ml:us-west-1:da6fc083-7c10-494c-85b5-d366e0236f17:workspace:3d726a00-e1e1-4814-8c02-3ffc2431df9b/3102fc94-44b8-4dad-adc8-7b81ab88aa0a
Deployed By	trial12_admin
Comment	Build churn model
Runtime Image	Python 3.7 (Standard)
File	code/5_model_serve_explainer.py
Function	explain

**Model Resources**

Replicas	1
Total CPU	1 vCPUs
Total Memory	2.00 GB

Models can be monitored to provide visibility into how they are responding to incoming requests.



## 14. Applications

Applications can be built to expose custom UI's to users and consumers of the application. E.g of applications are H2O UI, ML Flow, Cloudera Data Viz, Custom applications in flask, etc etc

Click to open the application UI

The screenshot shows a web-based application interface for a machine learning model. At the top, there's a logo consisting of three overlapping squares and the text "Application to Serve Churn UI" followed by a refresh icon. Below this, a green checkmark indicates the application has been running for 7 hours.

**Project:** Churn Modeling with scikit-learn - trial12\_admin

**Created by:** trial12\_admin

**Last Updated:** 04/26/2022 9:07 AM

**Data Preview:**

Refractor																				
ID	Probability	gender	SeniorCitizen	Partner	Dependents	Tenure	PhoneService	MultipleLines	InternetService	OnlineSecurity	OnlineBackup	DeviceProtection	TechSupport	StreamingTV	StreamingMovies	Contract	PaperlessBilling	PaymentMethod	MonthlyCharges	TotalCharges
7030	0.715	Male	Yes	Yes	No	4	Yes	Yes	Fiber	No	No	No	No	No	No	Month	Yes	Male	74.4	306.6
2226	0.619	Male	No	No	No	10	Yes	No	Fiber	No	No	No	No	No	No	Month	Yes	Elect	70	740
2844	0.129	Femal	No	No	No	12	Yes	No	No	No in	No in	No in	No in	No in	No in	Month	No	Male	20.3	246.7
5239	0.069	Femal	No	Yes	No	12	Yes	No	No	No in	No in	No in	No in	No in	No in	Two y	Yes	Elect	20.05	264.5
3502	0.066	Femal	No	No	No	70	Yes	Yes	Fiber	Yes	Yes	Yes	Yes	Yes	Yes	Two y	Yes	Bank	113.6	7339.
6069	0.063	Femal	No	Yes	No	55	Yes	Yes	DSL	Yes	Yes	Yes	Yes	Yes	Yes	One y	Yes	Elect	90.15	4916.
4837	0.047	Male	No	Yes	Yes	20	No	No ph	DSL	Yes	No	Yes	Yes	No	Two y	Yes	Credi	39.4	825.4	
1683	0.007	Femal	No	No	No	40	Yes	No	No	No in	No in	No in	No in	No in	No in	Two y	No	Male	20.15	804.8
3171	0.005	Femal	No	Yes	No	51	Yes	Yes	No	No in	No in	No in	No in	No in	No in	Two y	No	Male	25.5	1281.
2312	0.005	Femal	No	Yes	Yes	49	Yes	Yes	No	No in	No in	No in	No in	No in	No in	Two y	No	Male	25.25	1211.

Applications can be interactive to change the result on user provided input and can be exposed to non authenticated users also to consume the application

## Single Prediction View

Churn Probability **0.014**

Contract	Month-to-month	<b>0.12</b>	Month-to-month	One year	Two year	
Dependents	Yes	0	No	Yes		
DeviceProtection	No internet service	-0.05	No	No internet service	Yes	
InternetService	No	-0.07	DSL	Fiber optic	No	
MonthlyCharges	25.25	<b>0.31</b>	mean 64.80 min 18.25 max 118.75	<input type="text"/>	<b>Submit</b>	
MultipleLines	No phone service	0	No	No phone service	Yes	
OnlineBackup	No internet service	-0.06	No	No internet service	Yes	
OnlineSecurity	No internet service	-0.05	No	No internet service	Yes	
PaperlessBilling	No	0	No	Yes		
Partner	Yes	0	No	Yes		
PaymentMethod	Mailed check	0	Bank transfer (automatic)	Credit card (automatic)	Electronic check	Mailed check
PhoneService	Yes	0	No	Yes		
SeniorCitizen	No	0	No	Yes		
StreamingMovies	No internet service	0	No	No internet service	Yes	
StreamingTV	No internet service	-0.05	No	No internet service	Yes	
TechSupport	No internet service	-0.05	No	No internet service	Yes	
TotalCharges	1211.65	<b>-0.08</b>	mean 2283.30 min 18.80 max 8684.80	<input type="text"/>	<b>Submit</b>	
gender	Female	0	Female	Male		
tenure	49	<b>-0.14</b>	mean 32.42 min 1.00 max 72.00	<input type="text"/>	<b>Submit</b>	

Application Details -

## Application to Serve Churn UI

Overview    Logs    Settings

Application: Application to Serve Churn UI ↗

Script: [code/6\\_application.py](#)

Description:

No description for the app

Created by [trial12\\_admin](#)

Most Recent Start/Restart by [trial12\\_admin](#)

Ran:

1 time

### 15. ML Operations and Atlas Integration

Let's see how we can test the accuracy of the deployed model. First open the Workbench and run the `code/7a_ml_ops_simulation.py`. It would take approx 15 minutes to run. I will generate a random data set and invoke the model to generate the scores and store the predicted value (p values) in a database

```

Churn Modeling with scikit-learn - trial2_admin 1
  project-metadata.yaml
  cdsw-build.sh
  code
    0_bootstrap.py
    1_data_ ingest.py
    2_data_exploration.ipynb
    3_model_building.ipynb
    4_train_models.py
    5_model_serve_explainer.py
    6_application.py
    7a_ml_ops_simulation.py
    7b_ml_ops_visual.py
    > _pycache_
      churnexplainer.py
  README.md
  requirements.txt
  flask
  images
  LICENSE.txt
  lineage.yml
  model_metrics.db
  models
  raw
  README.md
  requirements.txt

```

File Edit View Navigate Run ▶ code/7a\_ml\_ops\_simulation.py

```

1 # ##### Cloudera Applied Machine Learning Prototype (AMP) #####
2 # Copyright © 2018 Cloudera, Inc. All rights reserved.
3 # (C) Cloudera, Inc. 2021
4 # All rights reserved.
5 #
6 # Applicable Open Source License: Apache 2.0
7 #
8 # NOTE: Cloudera open source products are modular software products
9 # made up of hundreds of individual components, each of which was
10 # individually copyrighted. Each Cloudera open source product is a
11 # collective work under U.S. Copyright Law. Your license to use the
12 # collective work is as provided in your written agreement with
13 # Cloudera. Used apart from the collective work, this file is
14 # licensed for your use pursuant to the open source license
15 # identified above.
16 #
17 # This code is provided to you pursuant a written agreement with
18 # (i) Cloudera, Inc., or (ii) a third-party authorized to distribute
19 # this code. If you do not have a written agreement with Cloudera nor
20 # with an authorized and properly licensed third party, you do not
21 # have any rights to access nor to use this code.
22 #
23 # Absent a written agreement with Cloudera, Inc. ("Cloudera") to the
24 # contrary, A) CLOUDERA PROVIDES THIS CODE TO YOU WITHOUT WARRANTIES OF ANY
25 # KIND; (B) CLOUDERA DISCLAIMS ALL EXPRESS AND IMPLIED WARRANTIES WITH
26 # REGARD TO THIS CODE, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF
27 # MERCHANTABILITY, FIDELITY, AND FITNESS FOR A PARTICULAR PURPOSE; (C) CLOUDERA IS NOT LIABLE TO YOU
28 # FOR ANY DAMAGES, LOSSES, EXPENSES, OR COSTS ARISING OUT OF YOUR EXERCISE
29 # OF ANY RIGHTS GRANTED TO YOU FOR THE CODE, CLOUDERA IS NOT LIABLE FOR ANY
30 # DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, PUNITIVE OR
31 # CONSEQUENTIAL DAMAGES (INCLUDING, WITHOUT LIMITATION, DAMAGES FOR LOSS OF
32 # PROFITS, LOST REVENUE, LOSS OF INCOME, LOSS OF
33 # BUSINESS ADVANTAGE OR UNAVAILABILITY, OR LOSS OR CORRUPTION OF
34 # DATA).
35 #
36 ##### Part 7a - Model Operations - Drift Simulation
37 #
38 # This script showcases how to use the model operations features of CML.
39 # This feature allows machine learning engineering to **measure and manage models
40 # through their life cycle**, and know how a model is performing over time. As part
41 # of this exercise, we will see how to track model metrics, which closes the loop on managing
42 # models that have been deployed into production.
43 #
44 # Add Model Metrics
45 # New metrics can be added to a model and existing ones updated using the 'cdsw'
46 # library and the [model metrics SDK](https://docs.cloudera.com/machine-learning/cloud/model-metrics/topics/
47 # If model metrics is enabled for a model, then every call to that model is recorded
48 # in the model metric database. There are situations in which it's necessary to update or
49 # add to those recorded metrics. This script shows how this works.
50 #
51 ##### Update Existing Tracked Metrics
52 # Note: this is called "ground truth". Certain machine learning implementations,
53 # (like this project) will use a supervised approach where a model is making a
54 # prediction and the actual value (or label) is only available at a later stage. To check
55 # how well a model is performing, these actual values need to be compared with the
56 # prediction made by the model. Each prediction response entry includes the response
57 # from the function, some other details, and a unique uid for that response.
58 # This tracked model response entry can then be updated at a later date to add the
59 # actual "ground truth" value, or any other data that you want to add.
60 #
61 # Data can be added to a tracked model response using the 'cdsw.track_delayed_metrics'.
62 #
63 # Example:
64 # python
65 # help(cdsw.track_delayed_metrics)
66 # Help on function track_delayed_metrics in module cdsw:
67 # track_delayed_metrics(metrics, prediction_uuid)
68 #     Description
69 #     -----
70 #     Track a metric for a model prediction that is only known after prediction time.
71 #     For example, for a model that makes a binary or categorical prediction, the actual
72 #     correctness of the prediction is not known at prediction time. This function can be
73 #     used to retroactively to track a prediction's correctness later, when ground truth
74 #     is available.
75 #     Example:
76 #     >>>track_delayed_metrics({'ground_truth': 'value'}, 'prediction_uui')
77 #
78 #     Parameters
79 #     -----
80 #     metrics: object
81 #         metrics object
82 #     prediction_uuid: string, UUID
83 #         prediction UUID of model metrics

```

Untitled Session ▶ Running  
By trial2\_admin - Session - 2 vCPU / 4 GiB Memory - a few seconds ago  
Session Logs ⚡ Collapse Share Export PDF

### Part 7a - Model Operations - Drift Simulation

This script showcases how to use the model operations features of CML.

**This feature allows machine learning engineering to \*\*measure and manage models\*\***

through their life cycle\*, and know how a model is performing over time. As part of the larger machine learning lifecycle, this closes the loop on managing models that have been deployed into production.

Add Model Metrics

New metrics can be added to a model and existing ones updated using the `cdsw` library and the `model metrics SDK`. If model metrics is enabled for a model, then every call to that model is recorded in the model metric database. There are situations in which it's necessary to update or add to those recorded metrics. This script shows how you do this.

Update Existing Tracked Metrics

This is part of what is called "ground truth". Certain machine learning implementations, *like this very project*, will use a supervised approach where a model is making a prediction and the actual value *or label* is only available at a later stage. To check how well a model is performing, these actual values need to be compared with the prediction from the model. Each time a model endpoint is called, it provides the response from the function, some other details, and a unique uid for that response. This tracked model response entry can then be updated at a later date to add the actual "ground truth" value, or any other data that you want to add.

Data can be added to a tracked model response using the `cdsw.track_delayed_metrics`.

Help on function `track_delayed_metrics` in module `cdsw`:

```

track_delayed_metrics(metrics, prediction_uuid)
Description
-----
Track a metric for a model prediction that is only known after prediction time.
For example, for a model that makes a binary or categorical prediction, the actual
correctness of the prediction is not known at prediction time. This function can be
used to retroactively to track a prediction's correctness later, when ground
truth
is available
Example:
>>>track_delayed_metrics({'ground_truth': 'value'}, 'prediction_uui')
d')

Parameters
-----
metrics: object
    metrics object
prediction_uuid: string, UUID
    prediction UUID of model metrics

```

Once complete, run the code/7b\_ml\_ops\_visual.py. This code will generate the actual values (a values) and compare them against the predicted values (p values) generated in the first step and build comparison graphs

```

7a_ml_ops_simulation.py
File Edit View Navigate Run ▶ code/7b_ml_ops_visual.py

# Churn Modeling with scikit-learn - trial2_admin1
# project-metadata.yaml
# cdsw-build.sh
# code
# 0_bootstrap.py
# 1_data_ingest.py
# 2_data_exploration.ipynb
# 3_model_building.ipynb
# 4_train_models.py
# 5_model_serve_explainer.ipynb
# 6_application.py
# 7a_ml_ops_simulation.py
# 7b_ml_ops_visual.py
# _pycache_
# churnexplainer.py
# README.md
# flask
# images
# LICENSE.txt
# lineage.yml
# model_metrics.db
# models
# raw
# README.md
# requirements.txt

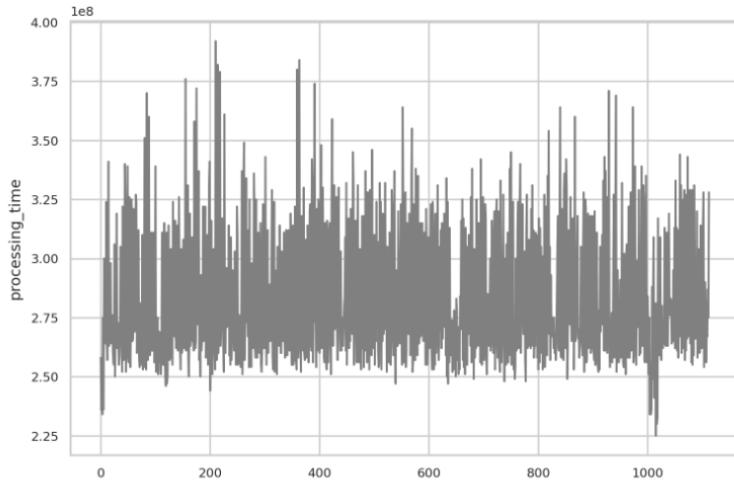
# =====#
# This code is provided to you pursuant a written agreement with
# Cloudera, Inc. or (ii) a third-party authorized to distribute
# this code. If you do not have a written agreement with Cloudera nor
# with an authorized and properly licensed third party, you do not
# have any rights to access nor to use this code.
# =====#
# Absent a written agreement with Cloudera, Inc. ("Cloudera") to the
# contrary, A) CLOUDERA PROVIDES THIS CODE TO YOU WITHOUT WARRANTIES OF ANY
# KIND, INCLUDING BUT NOT LIMITED TO IMPLIED WARRANTIES OF MERCHANTABILITY AND
# FITNESS FOR A PARTICULAR PURPOSE; (C) CLOUDERA IS NOT LIABLE TO YOU
# FOR ANY DAMAGES, LOSSES, EXPENSES OR CLAIMS ARISING FROM OR RELATED TO THE CODE; AND (D) WITH RESPECT TO YOUR EXERCISE
# OF ANY RIGHTS GRANTED TO YOU FOR THE CODE, CLOUDERA IS NOT LIABLE FOR ANY
# DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, PUNITIVE OR
# CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, LOST PROFITS,
# RELATED TO LOST REVENUE, LOST PROFITS, LOSS OF INCOME, LOSS OF
# BUSINESS ADVANTAGE OR UNAVAILABILITY, OR LOSS OR CORRUPTION OF
# DATA).
# =====#
# Part 7b - Model Operations - Visualising Model Metrics
# =====#
# This is a continuation of the previous process started in the
# '7a_ml_ops_simulations.py' script.
# Here we will load in the metrics saved to the model database in the previous step
# into a Pandas data frame, and display different features as graphs.
# =====#
# ``python
# help(cds.read_metrics)
# Help on function read_metrics in module cds:
# -----
# read_metrics(model_deployment_crn=None, start_timestamp_ms=None, end_timestamp_ms=None, model_crn=None, model_build_crn=None, description='')

# -----
# Read metrics data for given Crn with start and end time stamp
# -----
# Parameters
# -----
# model_deployment_crn: string
#     model deployment Crn
# model_crn: string
#     model Crn
# model_build_crn: string
#     model build Crn
# start_timestamp_ms: int, optional
#     metrics data start timestamp in milliseconds , if not passed
#     default value 0 is used to fetch data
# end_timestamp_ms: int, optional
#     metrics data end timestamp in milliseconds , if not passed
#     current timestamp is used to fetch data
# -----
# Returns
# -----
# object
#     metrics data
# -----
# import cds, time, os
# import pandas as pd
# import matplotlib.pyplot as plt
# import numpy as np
# from sklearn.metrics import classification_report
# from cmlobotstrap import CMLOBootstrap
# import seaborn as sns
# import sqlite3
# -----
# Get newly deployed churn model details using cmlobotstrapAPI
# -----
# HOST = os.getenv('CDSW_API_URL').split(':')[0] + '://' + os.getenv('CDSW_DOMAIN')
# USERNAME = os.getenv('CDSW_PROJECT_URL').split('/')[6] + args.username + '#vdibi
# API_KEY = os.getenv('CDSW_API_KEY')
# PROJECT_NAME = os.getenv('CDSW_PROJECT')
# cml = CMLOBootstrap(HOST, USERNAME, API_KEY, PROJECT_NAME)
# models = cml.net.models()
# 
```

Comparison of model predicted values against the actual values generated -

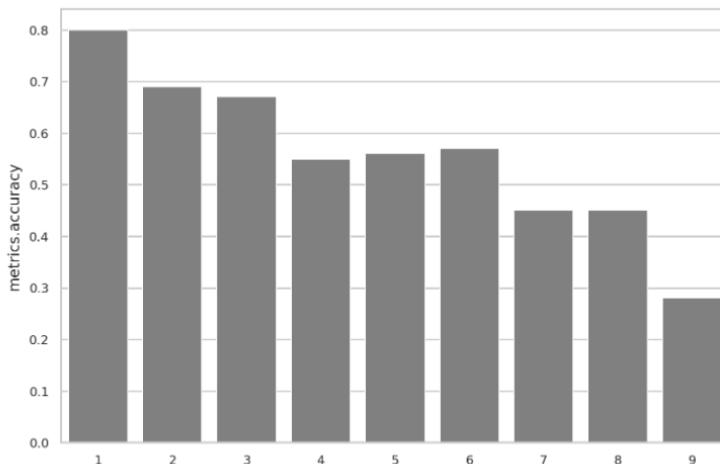
Plot processing time

```
> time_metrics = metrics_df.dropna(subset=["processing_time"]).sort_values(
    "startTimeStampMs"
)
> sns.lineplot(
    x=range(len(prob_metrics)), y="processing_time", data=prob_metrics, color="grey"
)
<AxesSubplot:ylabel='processing_time'>
```



Plot model accuracy drift over the simulated time period

```
> agg_metrics = metrics_df.dropna(subset=["metrics.accuracy"]).sort_values(
    "startTimeStampMs"
)
> sns.barplot(
    x=list(range(1, len(agg_metrics) + 1)),
    y="metrics.accuracy",
    color="grey",
    data=agg_metrics,
)
<AxesSubplot:ylabel='metrics.accuracy'>
```



## 16. Model Lineage and Governance

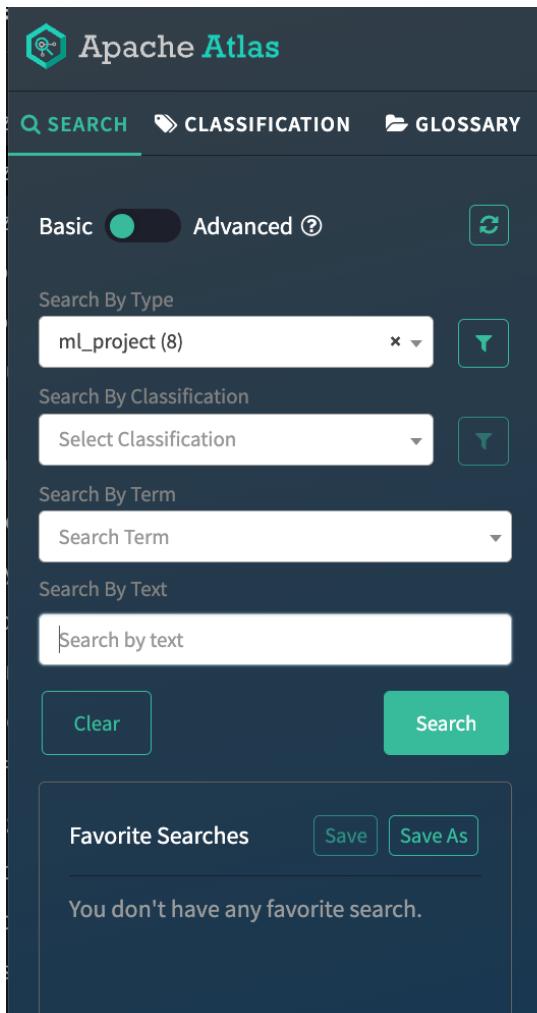
Go the the environment > Data Lake and open Atlas Link -

Environments / cdptrialuser12 / Data Lake / Event History

The screenshot shows the Cloudera Data Lake interface for the environment 'cdptrialuser12'. The top navigation bar includes 'Data Hubs', 'Data Lake' (which is selected), 'Cluster Definitions', and 'Summary'. Below the navigation, there are several buttons: '> SHOW CLI COMMAND', 'RETRY', 'REPAIR', 'RENEW CERTIFICATE', and 'RENEW PUBLIC CERTIFICATE'. The main content area is divided into sections:

- Environment Details:** Shows the environment name 'cdptrialuser12', credential 'cdptrialuser12-cred', region 'us-west-2', and availability zone 'us-west-2b'.
- Services:** A list of services including Atlas, CM UI, HBase UI, Name Node, Ranger, and Solr Server.
- Cloudera Manager Info:** Displays the CM URL (<https://cdptrialuser12-datalake-gateway.cdptrial.d06t-87n6.cloudera.site/cdptrialuser12-datalake/cdp-proxy/cmf/home/>), runtime version '7.2.8-1.cdh7.2.8.p8.22954680', and CM version '7.4.0'. Logs for 'Command logs' and 'Service logs' are also listed.

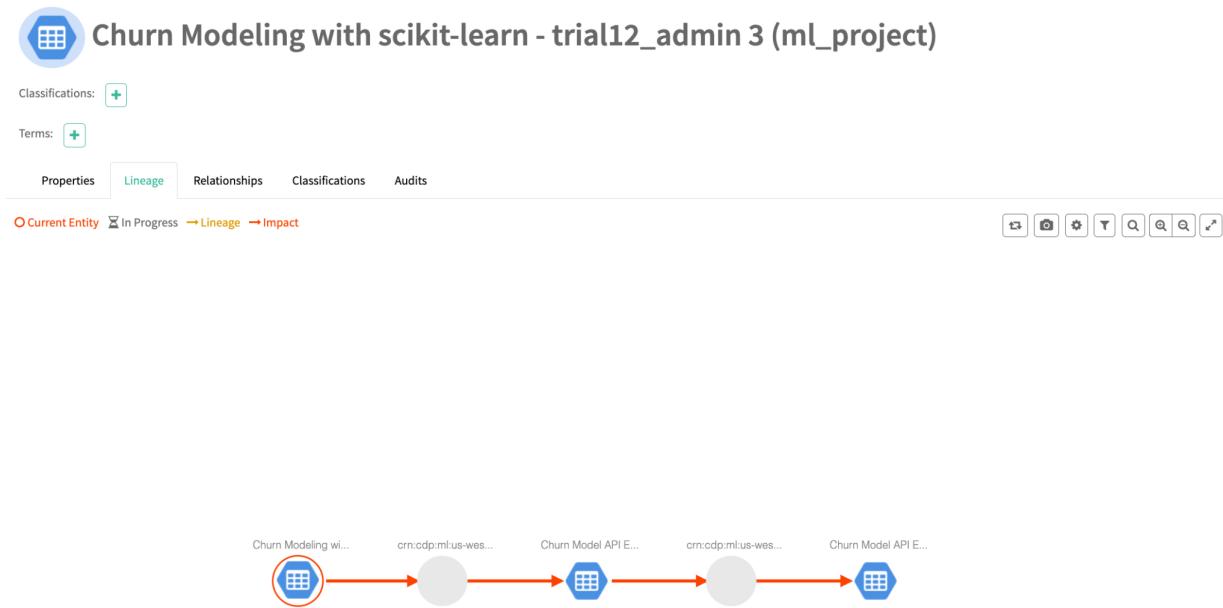
In atlas, on the left seach for ML Projects -



## Basic Search

Search Atlas for existing entities or [create new entity](#)

You can see the model, the data sets and the lineage of the model creation in Atlas



## 17. Remote Code Editing and Terminal + CLI Options

You can download the cli for Window, MacOS and Linux from the CML UI and use it for remote cli based access to submit jobs and run sessions in CML

The screenshot shows the 'User Settings' page in the Cloudera Machine Learning interface. The 'Remote Editing' tab is active. The 'Download CLI client' section provides instructions and download links for Windows, macOS, and Linux. The 'SSH public keys for session access' section shows a table with one entry: 'Begins with 'ssh-rsa', 'ssh-dss', 'ssh-ed25519', 'ecdsa-sha2-nistp256', 'ecdsa-sha2-nistp384', or 'ecdsa-sha2-nistp521''. An 'Add' button is available for adding more keys.

```

MacBook-Pro :: cdppc/cml » ./cdswctl --help
Usage:
  cdswctl [OPTIONS] <command>

Cloudera Data Science Workbench command-line client 2.0.0.55615
This client is in beta and is subject to change in the future.

Help Options:
  -h, --help  Show this help message

Available commands:
  engine-images  Work with CDSW engine images
  jobruns        Work with CDSW Job Runs
  jobs          Work with CDSW Jobs
  login         Log in
  models         Work with CDSW models
  projects       Work with CDSW projects
  runtime-addons Work with ML runtime-addons
  runtimes       Work with ML runtimes
  sessions       Work with CDSW sessions
  ssh-endpoint   Forward SSH port to session
  version        Show version

```

## 18. ML Flow Demo

Make a new project using this github as the base -

<https://github.com/myloginid/experiments-with-mlflow>



\* Project Name

Project Description

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

Initial Setup

Git URL of Project ⓘ

Runtime setup

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

Add GPU enabled Runtime variant

These runtimes will be added to the project:

JupyterLab - Python 3.7 - Standard - 2021.12  
Workbench - Python 3.7 - Standard - 2021.12

Start a Workbench session and run train\_model.py

The screenshot shows a Jupyter Notebook interface with the following details:

- Code Cell:** The code cell contains the `train_model.py` script, which performs the following steps:
  - Imports pandas, sklearn, and mlflow.
  - Loads the 'bikeshare.csv' dataset.
  - Splits the data into train and test sets.
  - Creates a pipeline with a ColumnTransformer, OneHotEncoder, PolynomialFeatures, and an ElasticNet estimator.
  - Fits the pipeline on the training data and scores it.
  - Logs the score and parameters to MLflow.
  - Prints the log message: "INFO: 'bikeshare' does not exist. Creating a new experiment".
- Output Cell:** The output cell shows the command `> ! pip3 install -r requirements.txt` followed by the pip installation logs for various packages like pandas, sklearn, scikit-learn, and threadpoolctl.
- Session Tab:** The session tab shows the session is running and provides session logs.
- Toolbar:** The toolbar includes Project, Terminal Access, Data, Clear, Interrupt, Stop, and Sessions.

## Experiment

Experiment Name: bikeshare  
Experiment ID: yvl7-gnjw-uvzf-mtdj  
Artifact Location: /home/cdsw/.experiments/yvl7-gnjw-uvzf-mtdj

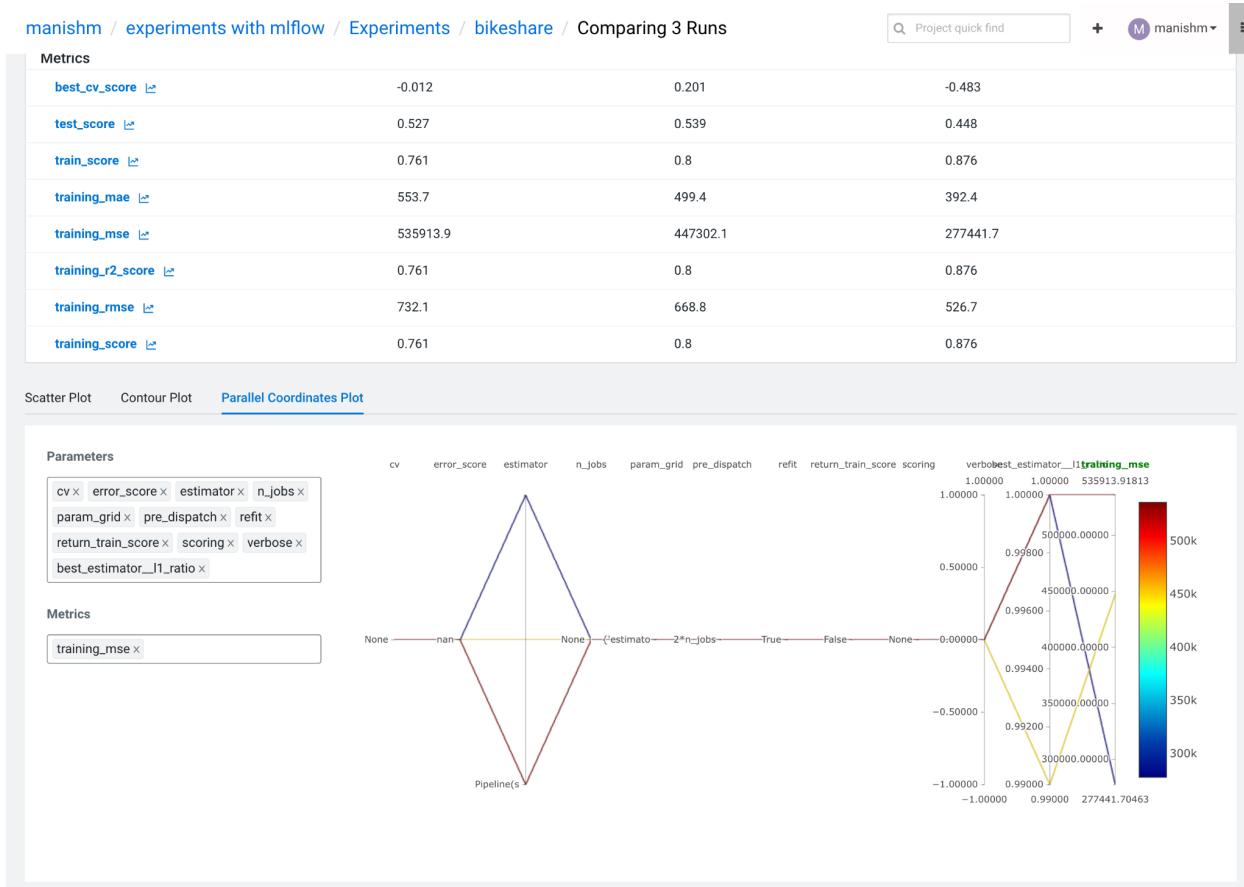
> Notes 

Runs (24)

Parameters >										Metrics >			Tags			
	Stat	Start Time	Run Name	Duration	User	Source	Version	Models	best_ests	cv	error_score	best_cv_score	mean_fit	mean_sco	estimator	estimator
		2022-04-13 12	uzqd-rn8b-1q9k-k...	5.6s	manis...		-		1	None	nan	-0.011649241...	0.034...	0.001...	Pipeli...	sklear...
		2022-04-13 12	tatr-ddrn-qqar-2yif	1.9s	manis...		-		0.99	None	nan	0.201456970...	0.009...	0.001...	Pipeli...	sklear...
		2022-04-13 12	c7mt-rm3oo-uhf4...	2.3s	manis...		-		1	None	nan	-0.483038087...	0.018...	0.001...	Pipeli...	sklear...
		2022-04-13 12	o699-uwjx-n8ff-lnss	2.2s	manis...		-		1	None	nan	-0.483038087...	0.036...	0.001...	Pipeli...	sklear...
		2022-04-13 12	xabc-fors-z4sp-7c6l	-140...	manis...		-	-	-	-	-	-	0.034...	0.001...	Pipeli...	sklear...
		2022-04-13 12	v1vl-x6sb-o8u8-tk3e	-143...	manis...		-	-	-	-	-	-	0.009...	0.001...	Pipeli...	sklear...
		2022-04-13 12	kspl-uox8-qayy-bo...	-143...	manis...		-	-	-	-	-	-	0.018...	0.001...	Pipeli...	sklear...
		2022-04-13 12	50nc-udrh-hbw6...	-145...	manis...		-	-	-	-	-	-	0.009...	0.001...	Pipeli...	sklear...
		2022-04-13 12	ftni-kn2-n7q8-gthp	-146...	manis...		-	-	-	-	-	-	0.036...	0.001...	Pipeli...	sklear...

Compare the results -





## 19. CDSW Administration

### a. Usage Overview

## Site Administration

Overview    Users    Teams    Usage    Quotas    Models    Runtime/Engine    Data Connections    Security    AMPs    Settings    Support

### Cluster Monitoring

View cluster usage metrics and trends in custom built Grafana dashboards.

[Grafana Dashboard](#)

### Cluster Metrics Snapshot

Release	dev
Domain	ml-1f3289a1-818.se-sandb.a465-9q4k.cloudera.site
Total Nodes	1
Total Memory	60.34 GiB
Used Memory	27.95 GiB
Total vCPUs	15.63
Used vCPUs	14.79
Total GPUs	0
Used GPUs	0
Total Active Users in Last Day	10
Total Active Users	16
Total Teams	0
Total Projects	11
Total Jobs	9
Total Running Jobs	0
Total Job Runs	10
Total Running Sessions	6
Total Session Runs	42
Average Engine Scheduling Time (seconds, last 20 days)	20.70

## b. Quotas

Site Administration / Quotas

Project quick find + manishm ▾

## Site Administration

Overview Users Teams Usage Quotas Models Runtime/Engine Data Connections Security AMPs Settings Support

OFF ?

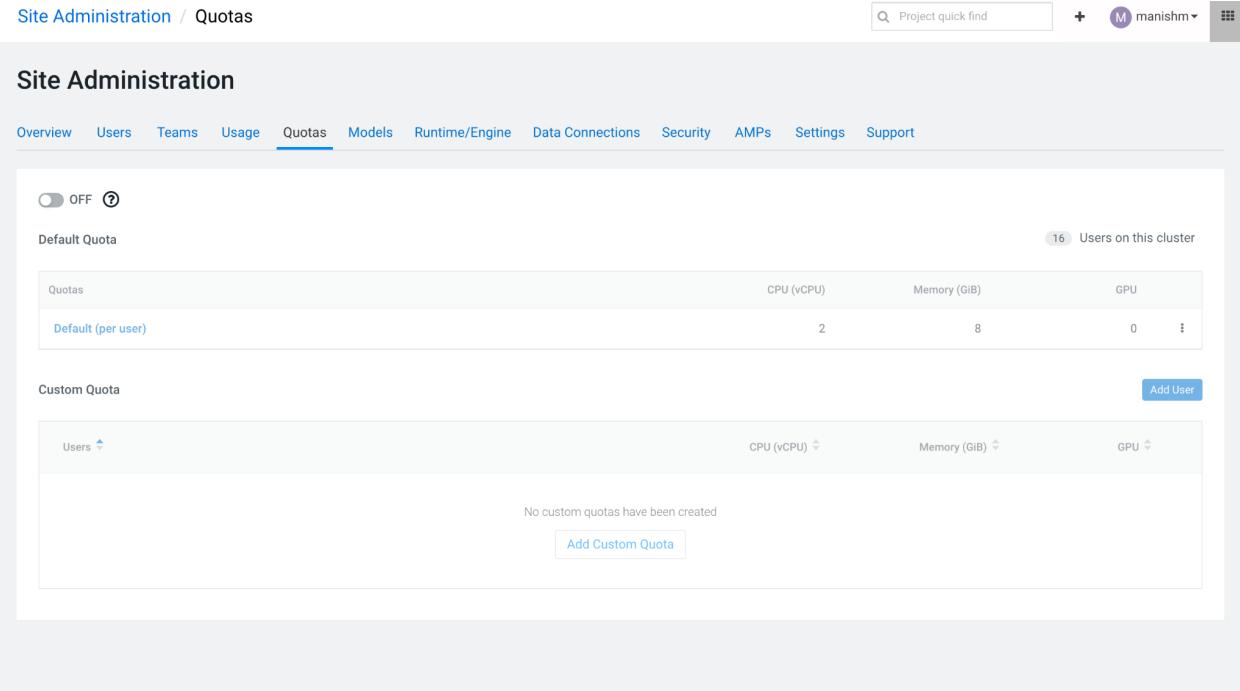
Default Quota 16 Users on this cluster

Quotas	CPU (vCPU)	Memory (GiB)	GPU
Default (per user)	2	8	0

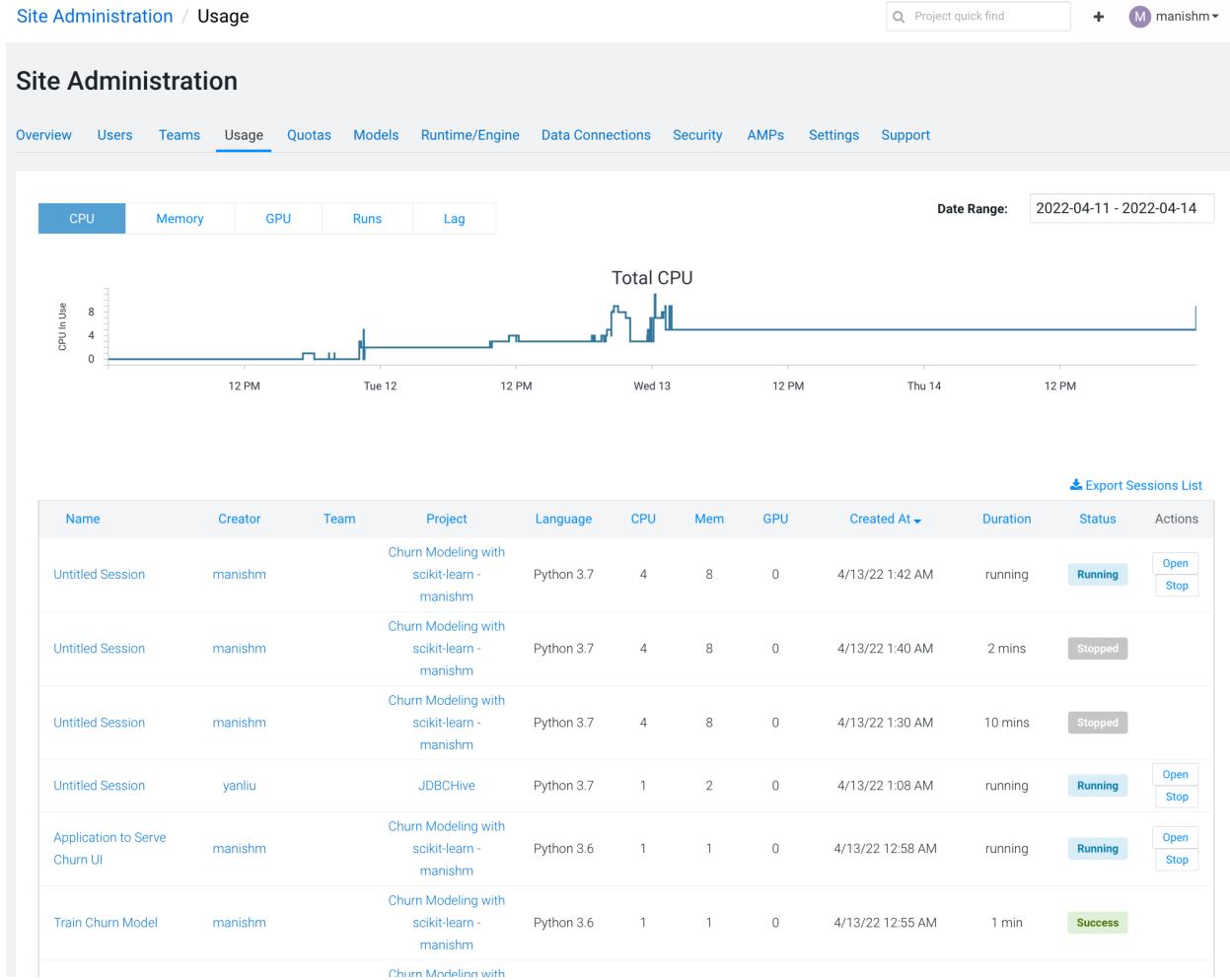
Custom Quota Add User

Users	CPU (vCPU)	Memory (GiB)	GPU
No custom quotas have been created			

[Add Custom Quota](#)



### c. Detailed Usage



### d. Settings

## Site Administration

[Overview](#) [Users](#) [Teams](#) [Usage](#) [Quotas](#) [Models](#) [Runtime/Engine](#) [Data Connections](#) [Security](#) [AMPS](#) [Settings](#) [Support](#)

### General

- Require invitation to sign up
  - Send usage data to Cloudera
- Cloudera uses aggregate usage and error tracking data to improve product quality and does not share or resell any data.

- Allow users to create public projects

When enabled, users can create public projects. When this property is disabled, existing public projects are not affected and continue to be publicly accessible.

- Allow users to use the Python 2 kernel

- Use original line endings when editing files

If true, line endings in the result depend on the document's line endings setting (based on OS & the original text loaded from disk).

If false, line endings are always \n.

- Disable team synchronization upon login

By default, upon a user's login, Cloudera Machine Learning synchronizes the synced teams the user belongs to. With this flag set, you can disable the synchronization upon login. In order to keep the synced teams synchronized with the membership information on CDP, you need to manually press the 'Synchronization' button on the admin panel.

### Access Control

- Allow users to create projects
- Allow users to create teams

### Feature Flags

- Allow users to run experiments
- Allow users to create models
- Allow users to create applications
- Allow business user access to this cluster
- Allow users to use AMPS (Applied ML Prototypes)
- Allow users to use ML Runtime Addons

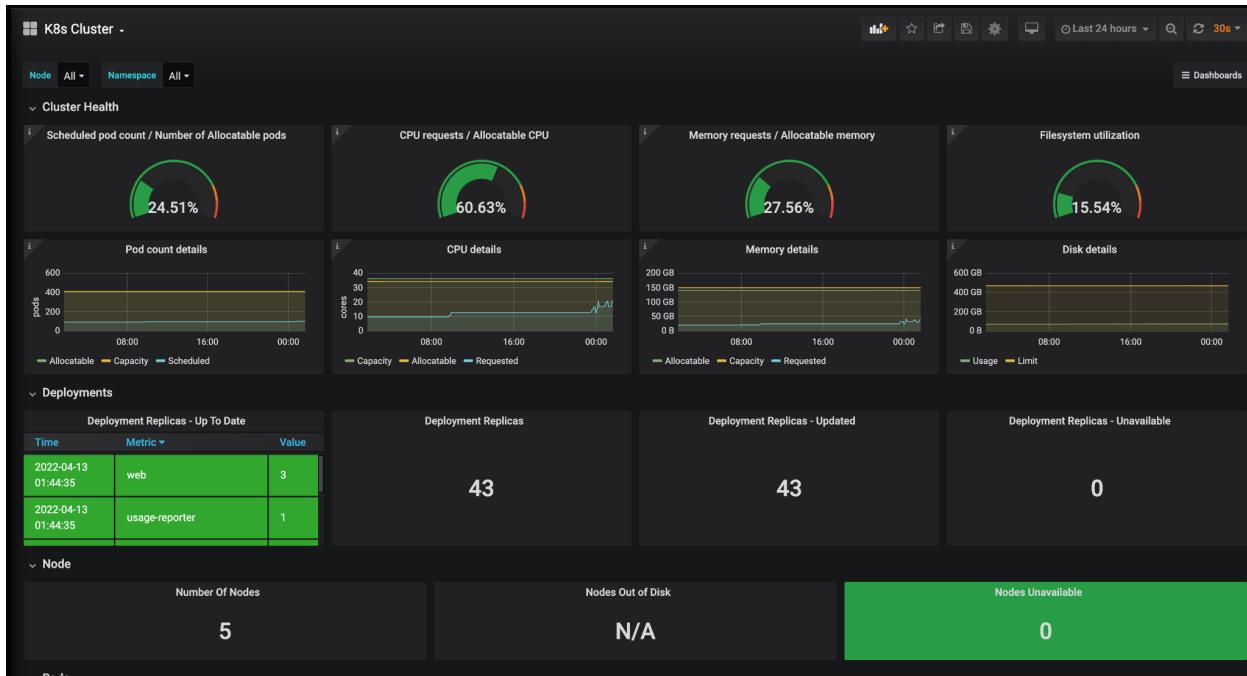
Use these checkboxes to show/hide the Models, Experiments and Applications features in the UI. Note that these controls do not affect any active models, experiments, or applications. In particular, if you do not stop [active models](#) before hiding the Models feature, they will continue to serve requests and consume computing resources in the background.

### Project Templates

Name	Git Repository	Enabled	Actions
R (default)		<input checked="" type="checkbox"/>	
Python (default)		<input checked="" type="checkbox"/>	
PySpark (default)		<input checked="" type="checkbox"/>	
Scala (default)		<input checked="" type="checkbox"/>	
<input type="text"/> Name	<input type="text"/> Git repository URL		<input type="button" value="Add"/>

Cloudera Machine Learning provides 4 built-in sample templates upon installation. You can add custom project templates by providing template name and git repository URL. Custom

## e. Grafana Dashboard



## 20. Learning Hub

## Learning Hub

Project quick find

manishm

## Featured Announcements



Data Connections and Snippets  
Cloudera Machine Learning now offers  
Snippet to connect to Data Sources availab...

January 1, 2022



Project-level ML Runtime configuration in  
Project-level configuration for ML Runtimes  
adds the ability to limit the available Runtim...

January 1, 2022



Cloudera Machine Learning APIv2  
Cloudera Machine Learning's APIv2 enables  
automated project lifecycle management, ...

September 27, 2021



Apache Spark 3 is now available in CML  
Cloudera Machine Learning now offers multi-  
version Spark support. Users of CML can ...

September 21, 2021

## Research and Resources

Blog Posts

Research Reports

Documentation



## Cloudera Machine Learning Overview

Machine learning has become one of the  
most critical capabilities for modern  
businesses to grow and stay competitive  
today. From automating internal processes t...

November 1, 2021

Creating a Project with ML Runtimes  
Variants

Projects create an independent working  
environment to hold your code, configuration,  
and libraries for your analysis. This topic  
describes how to create a project with ML...

November 1, 2021



## Applied ML Prototypes (AMPs)

Applied ML Prototypes (AMPs) provide  
reference example machine learning projects  
in Cloudera Machine Learning. More than  
simplified quickstarts or tutorials, AMPs are...

November 1, 2021



## Creating and Deploying a Model

Using Cloudera Machine Learning, you can  
create any function within a script and deploy  
it to a REST API as a model. In a machine  
learning project, this will typically be a predi...

November 1, 2021



## Analytical Applications

This feature gives data scientists a way to  
create ML web applications/dashboards and  
easily share them with other business  
stakeholders. Applications can range from...

November 1, 2021