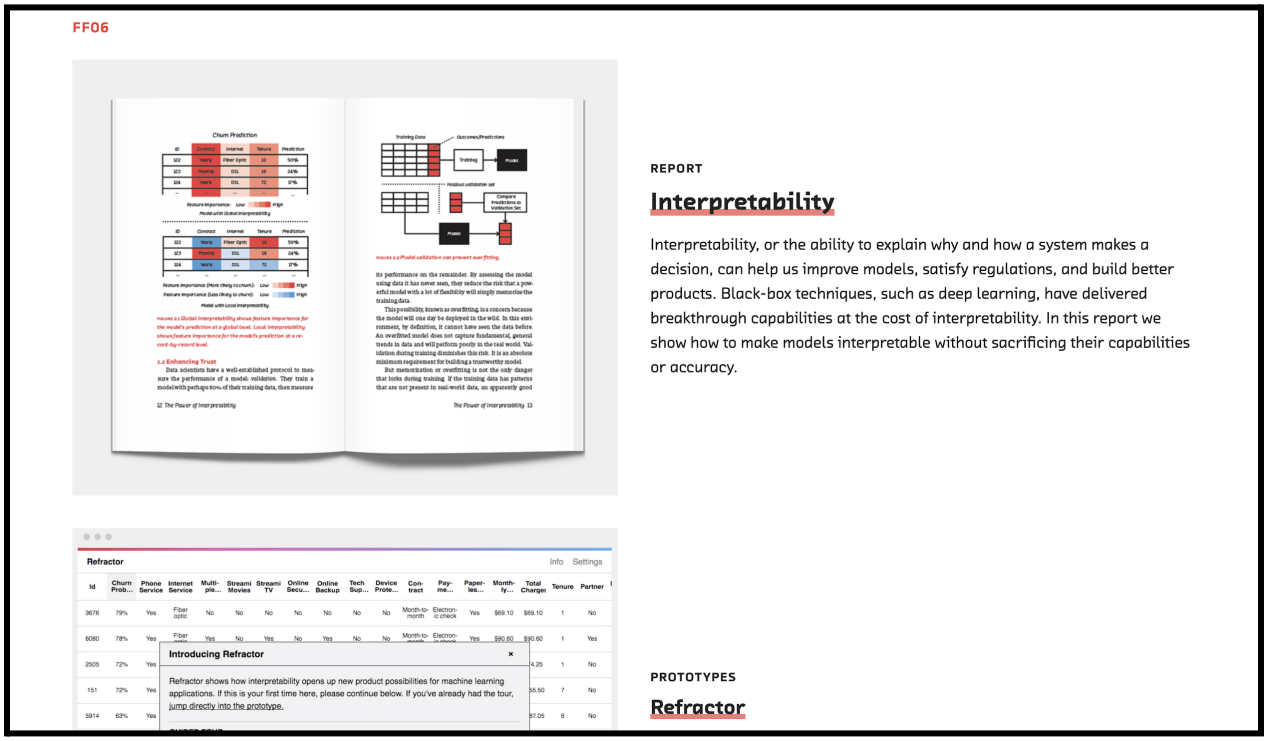
CML Workshop Student Guide

Introduction

Business Use Case

In this Hands On Lab you will create a model to predict customer churn and present model-driven insights to your stakeholders. You will use an interpretability technique to make your otherwise “black box model” explainable in an interactive dashboard. A mathematical explanation is beyond the scope of this lab but if you are interested in learning more we recommend the [Fast Forward Labs Report on Model Interpretability]. Finally, you will use basic ML Ops techniques to productionize and monitor your model.



The Hands on Labs will take you through the whole process of logging in, sourcing the code and building fully working applications with deployed models. This lab builds the telco churn with model interpretability projects discussed in more detail in this blog post.

Labs Summary

In this exercise we will implement an end-to-end machine learning workflow using Cloudera Machine Learning, including:

- Data ingest
- Data exploration
- Model training and experimentation
- Model serving
- Business applications

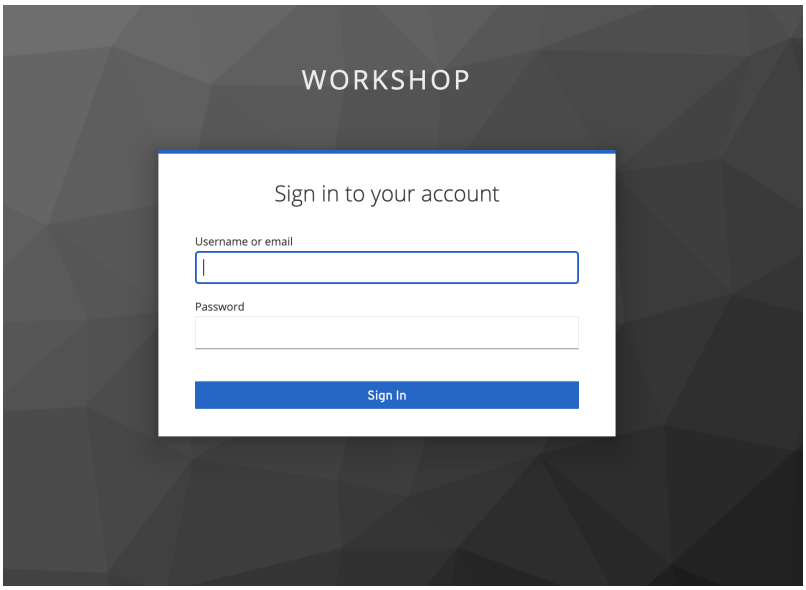
Lab 1 - Initial setup

Getting Connected

CDP Console URL	https://54.169.212.131:8443/auth/realms/uob-workshop/protocol/saml/clients/samlclient
User	Refer Google sheet
Password	Refer Google sheet
Project Github URL	**https://github.com/ogakulov/CML_AMP_Churn_Prediction_mlflow**

Setting Up Workload Password

- We have already take care of the below prerequisites:
  - Enable CML workspace (cml-workshop)
- Open the shared link and login with the credentials assigned to you.

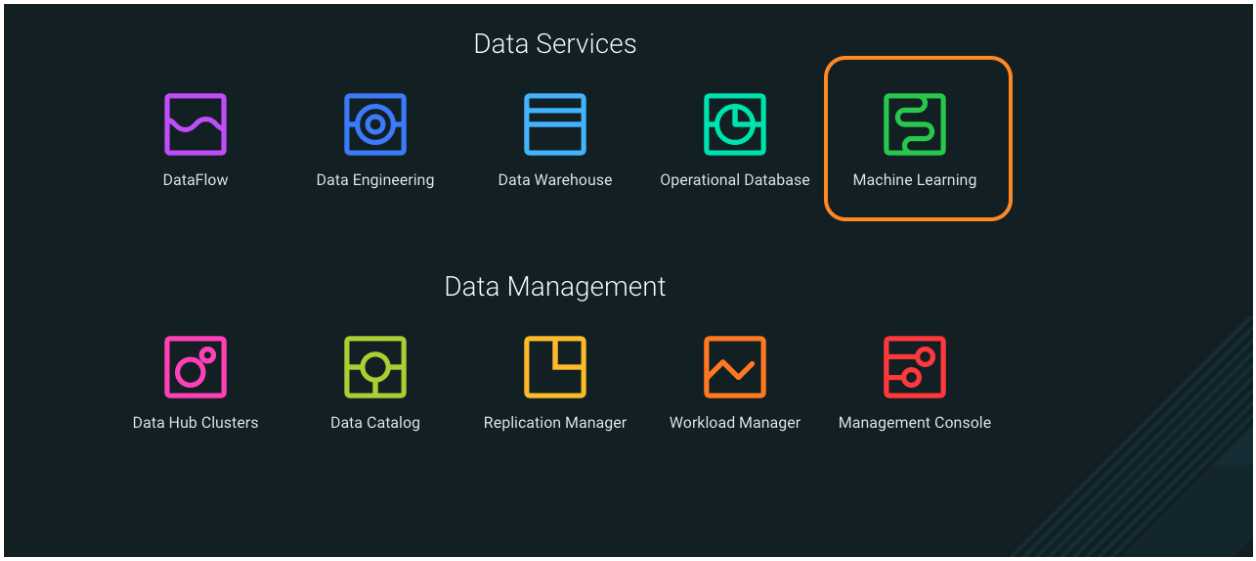


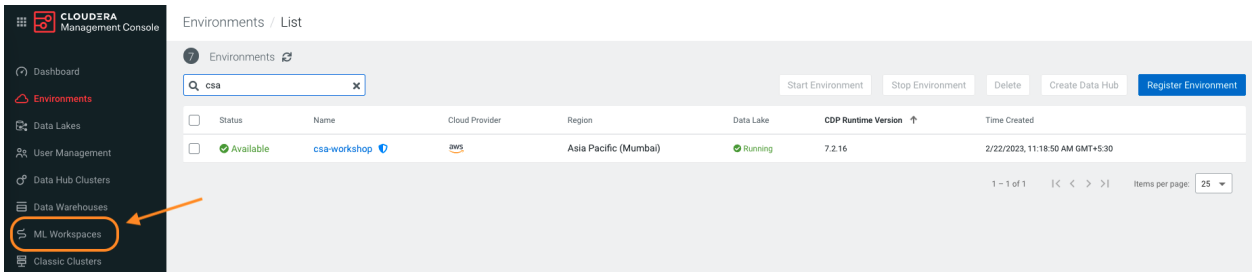
- You should land on the CDP Console as shown below.


Your Enterprise Data Cloud
Data Hub Clusters
DataFlow
Data Engineering
Data Warehouse
Operational Database
Machine Learning
Control Plane
Data Catalog
Replication Manager
Workload Manager
Management Console

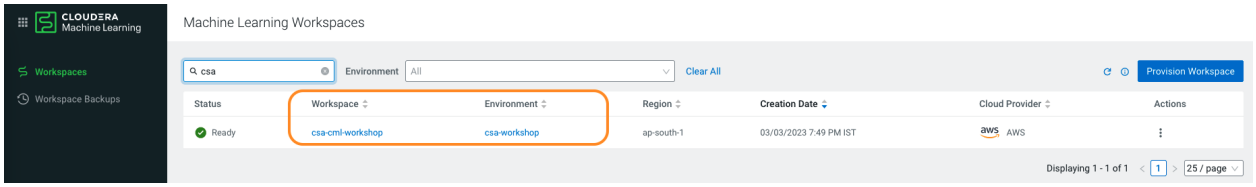
Create CML Project

Login into CDP using the URL above and the credentials assigned to you. After logging into CDP you will have access to the main console and then go to Cloudera Machine Learning workspace. Click on Workspace to proceed to executing the hands-on labs.



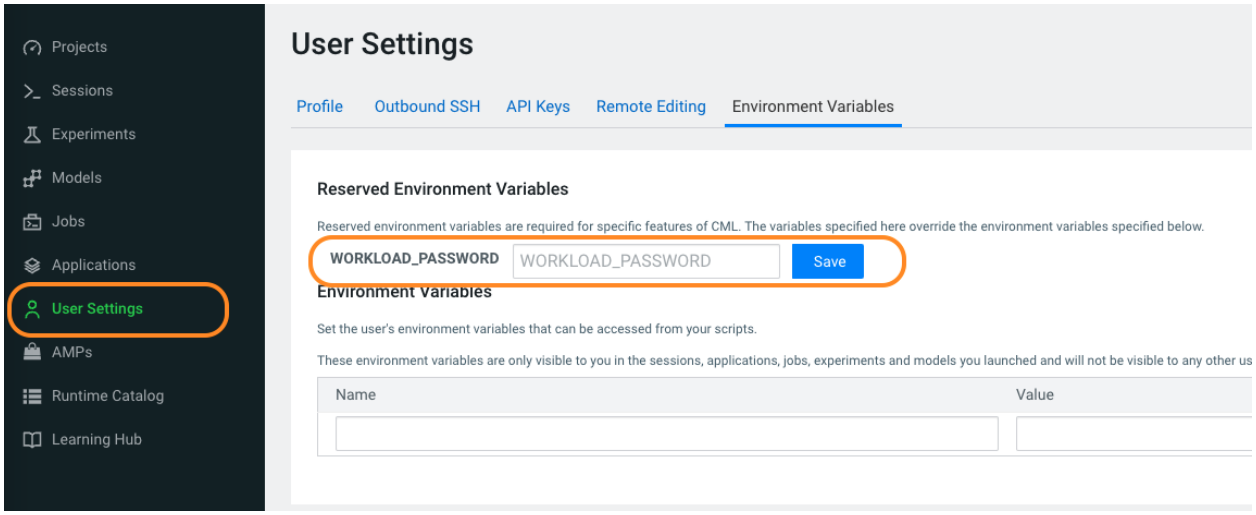


This will launch a ML workspace screen.

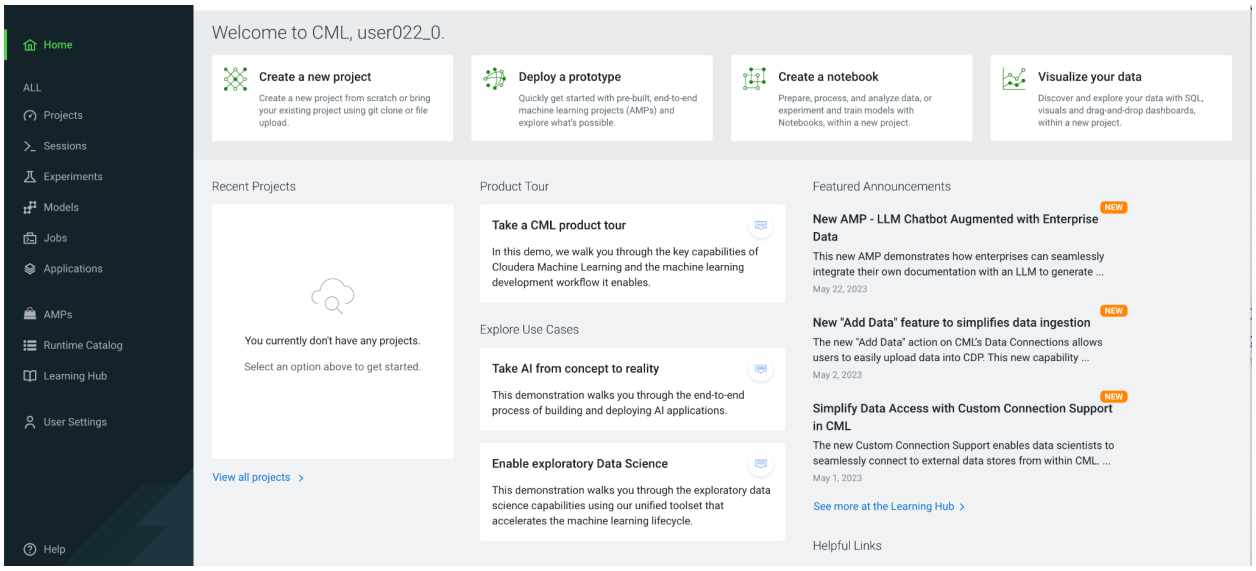


Workload Password

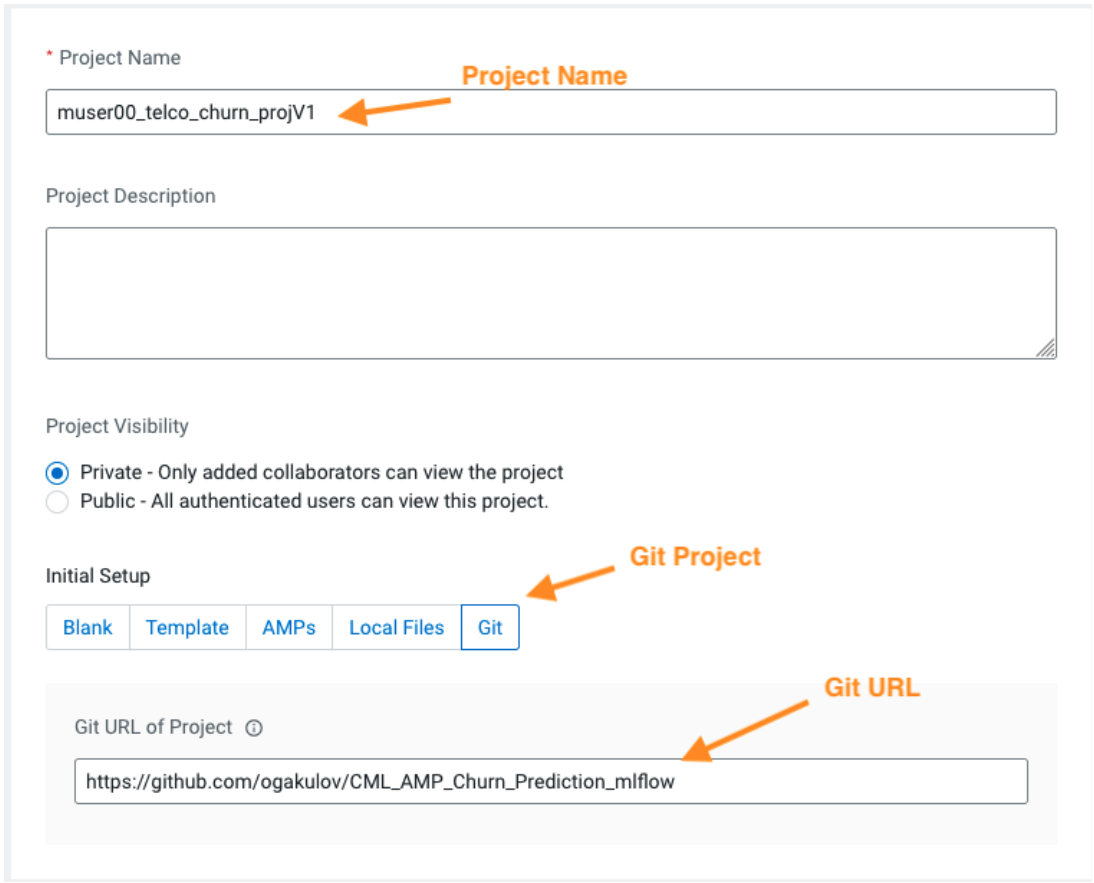
Provide the same password as we set in the previous step.

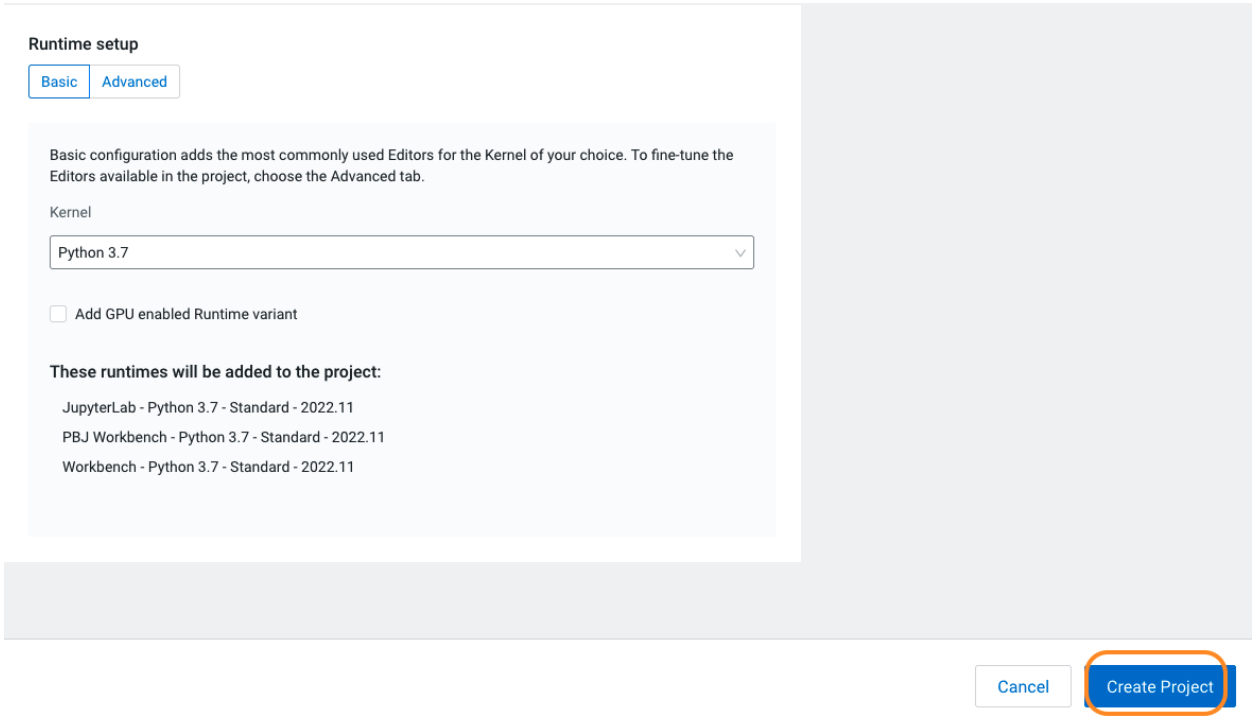


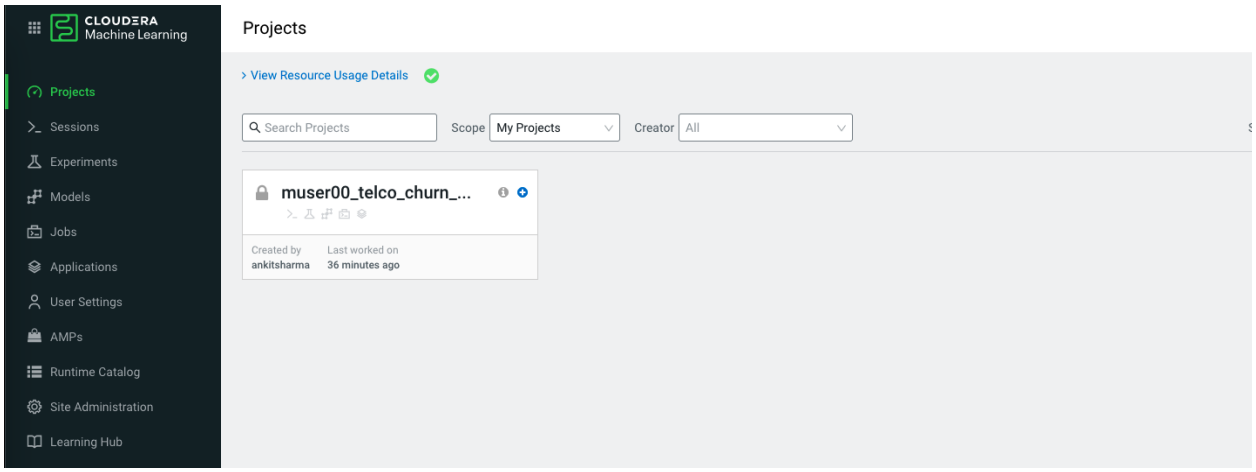
Create a New Project



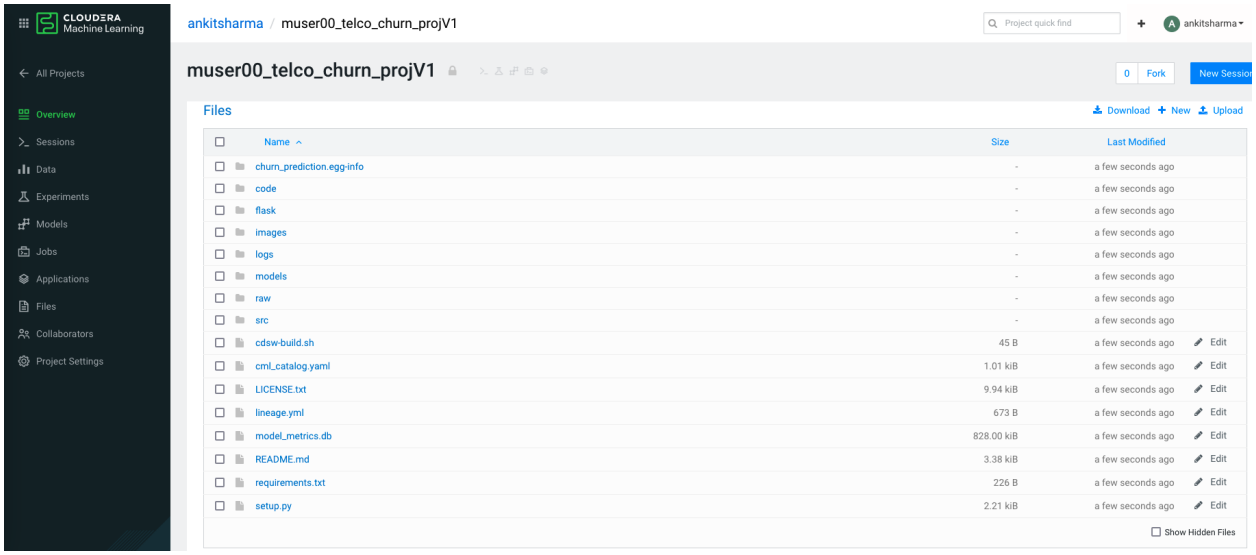
Create a new Project based on Github.
Project Name : <username>_telco_churn_projV1
Initial Setup : Git
Git URL : https://github.com/ogakulov/CML_AMP_Churn_Prediction_mlflow


* Project Name
Project Name
muser00_telco_churn_projV1
Project Description
Project Visibility
Private - Only added collaborators can view the project
Public - All authenticated users can view this project.
Initial Setup
Git Project
Blank
Template
AMPs
Local Files
Git
Git URL
Git URL of Project
https://github.com/ogakulov/CML_AMP_Churn_Prediction_mlflow

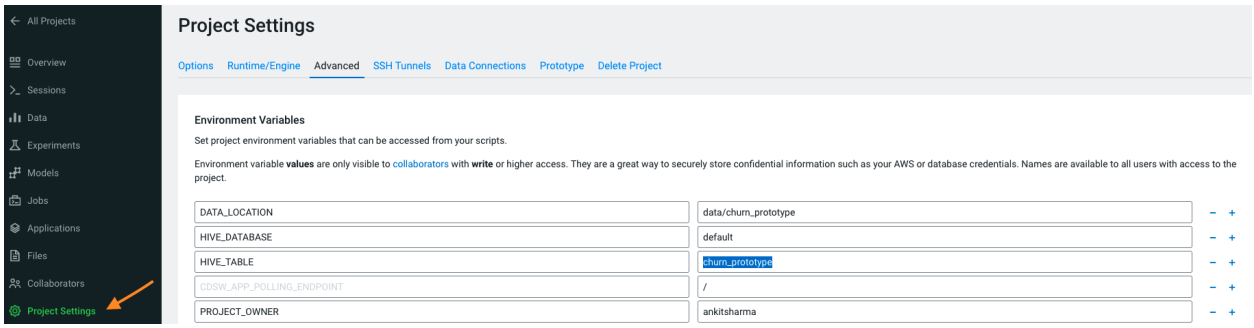

Runtime setup
Basic
Advanced
Basic configuration adds the most commonly used Editors for the Kernel of your choice. To fine-tune the Editors available in the project, choose the Advanced tab.
Kernel
Python 3.7
Add GPU enabled Runtime variant
These runtimes will be added to the project:
JupyterLab - Python 3.7 - Standard - 2022.11
PBJ Workbench - Python 3.7 - Standard - 2022.11
Workbench - Python 3.7 - Standard - 2022.11
Cancel
Create Project



The project is laid out in a familiar way with markdown and a file view that is familiar to developers.



Project Environment Variables



Add below variable in Project Settings

Variable Name	**Value**
DATA_LOCATION	my-data/churn_prototype
HIVE_DATABASE	default
HIVE_TABLE	churn_prototype_<username>
STORAGE_MODE	external

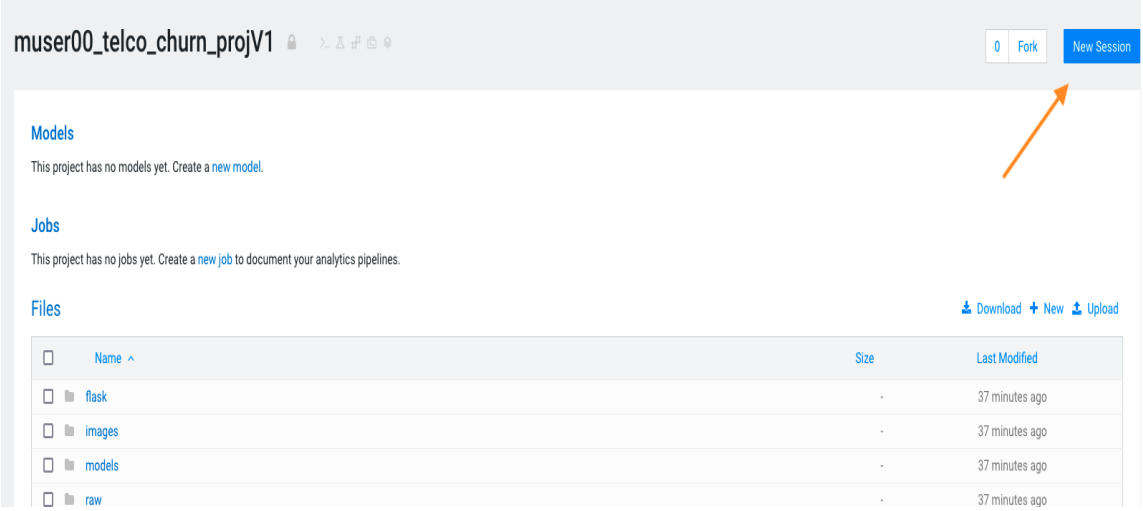
Lab 2 - Data Ingest

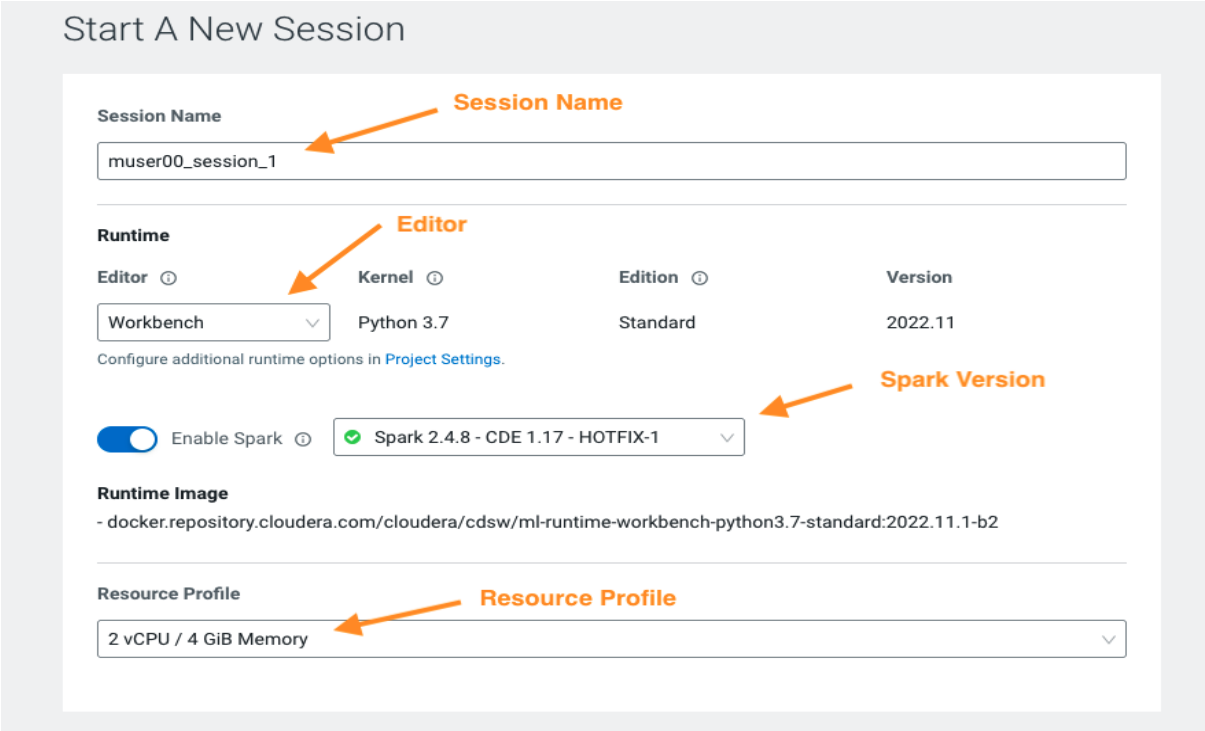
In this and the following lab, you will work on the Data Ingest Stage.

Run the Bootstrap Script

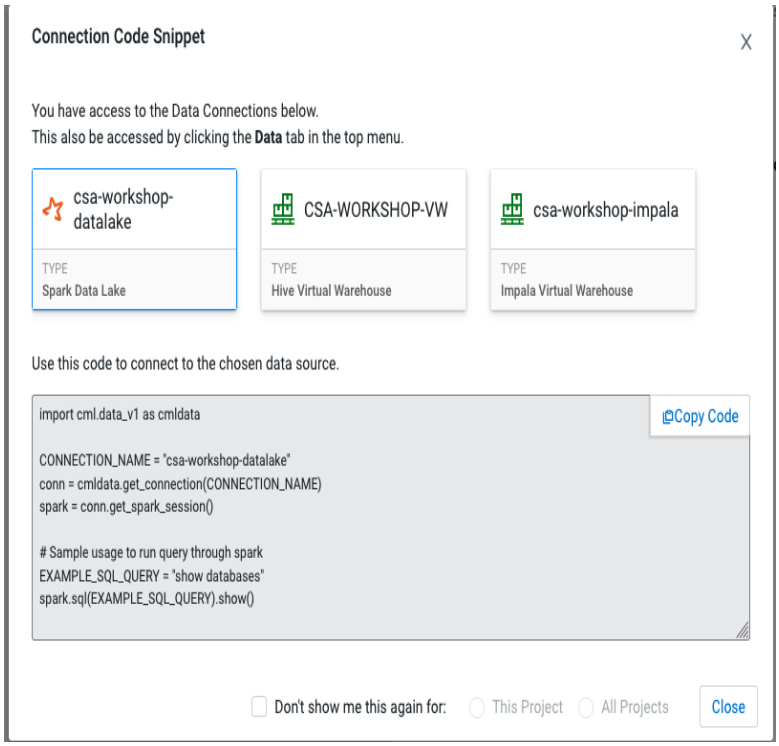
Start a "**NEW SESSION**" and use the below configuration.

***Editor**: Workbench*
***Engine Kernel:** Python 3*
***Resource Profile:** 2 vCPU / 4 GiB Memory*
***Enable Spark :** Yes (Default Version)*



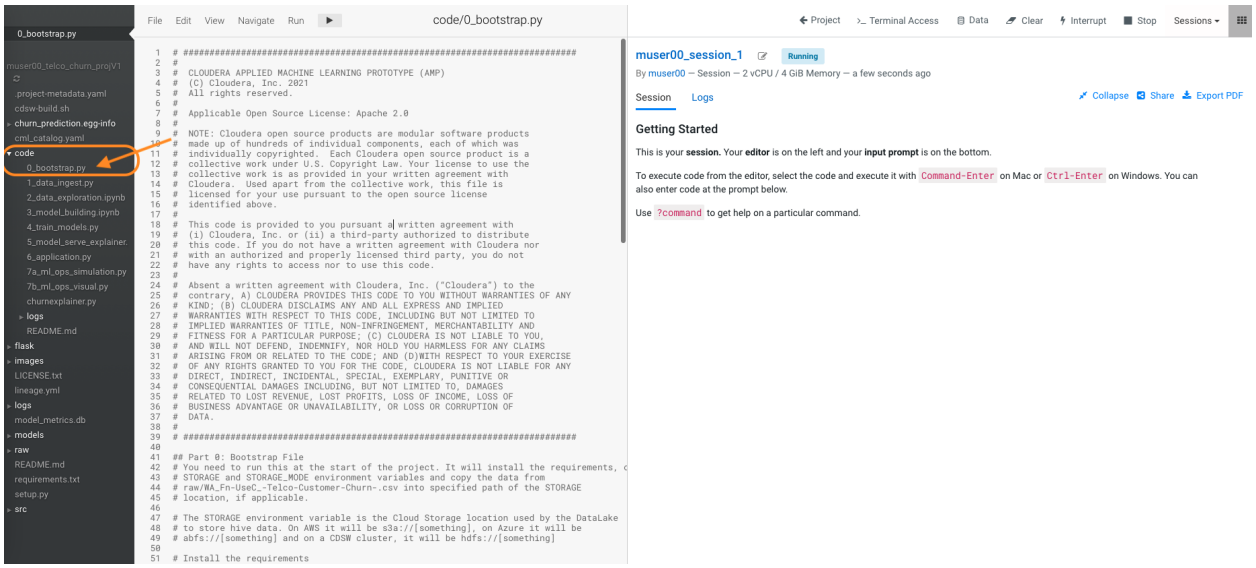


Click on ***Start Session***

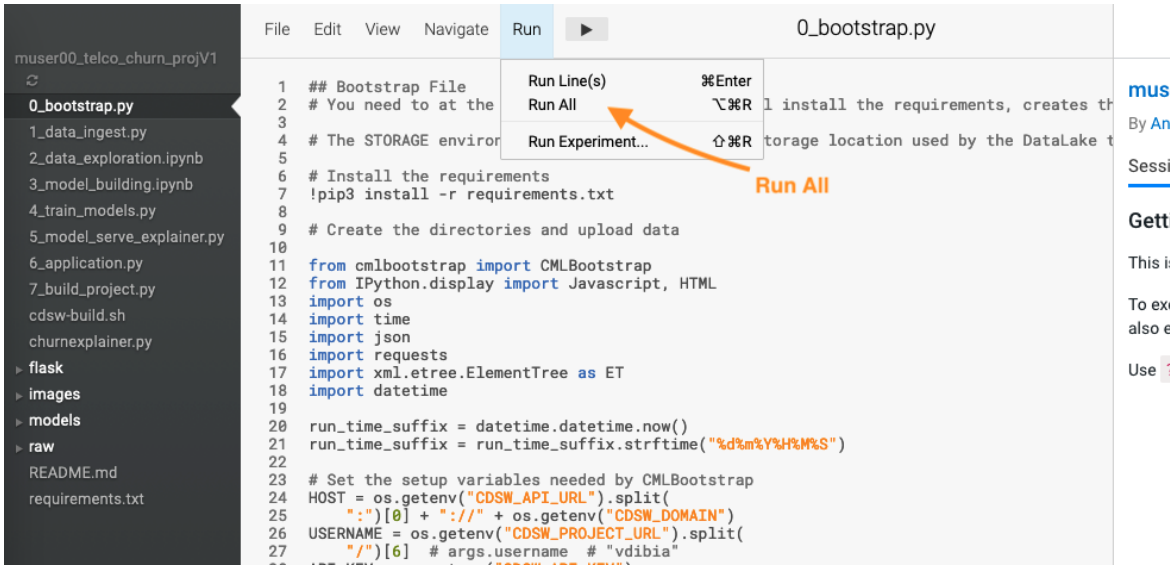


Close the **Code Snippet**

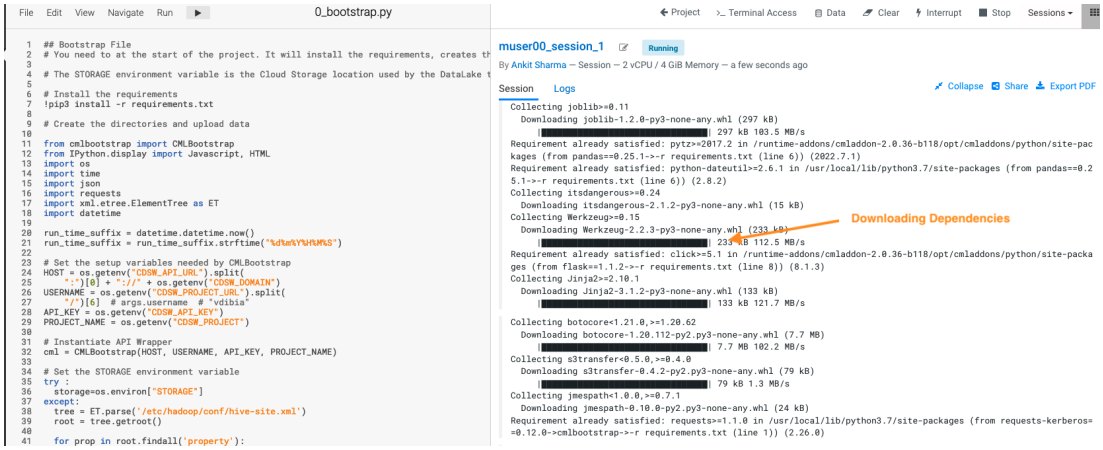
- Select the **code/0_bootstrap.py** on the left file browser



- Run **code/0_bootstrap.py** to download the dependencies**.**



- Dependencies are starting to get downloaded.



You can open up requirements.txt to see what is loaded

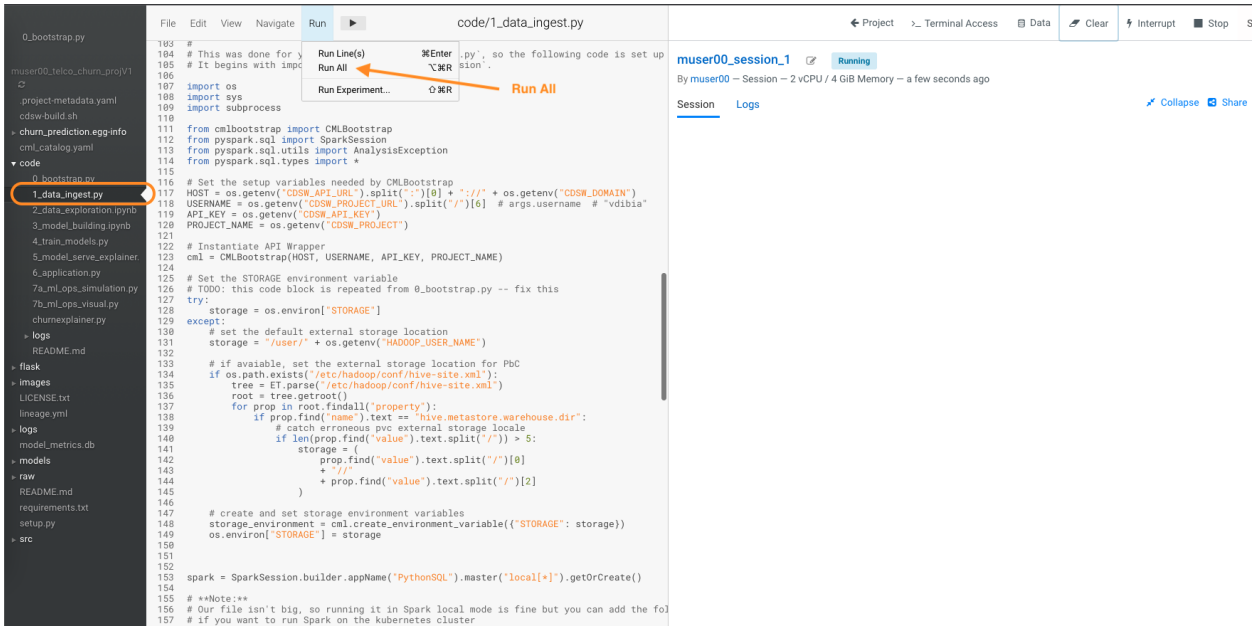
Scikit learn for Machine learning
Flask for Application serving

All dependencies now live with the project and can be shared with others and your team through project forking. This may take a few minutes to install the requirements. At the end you will see the below output.

Execute Data Ingest Script

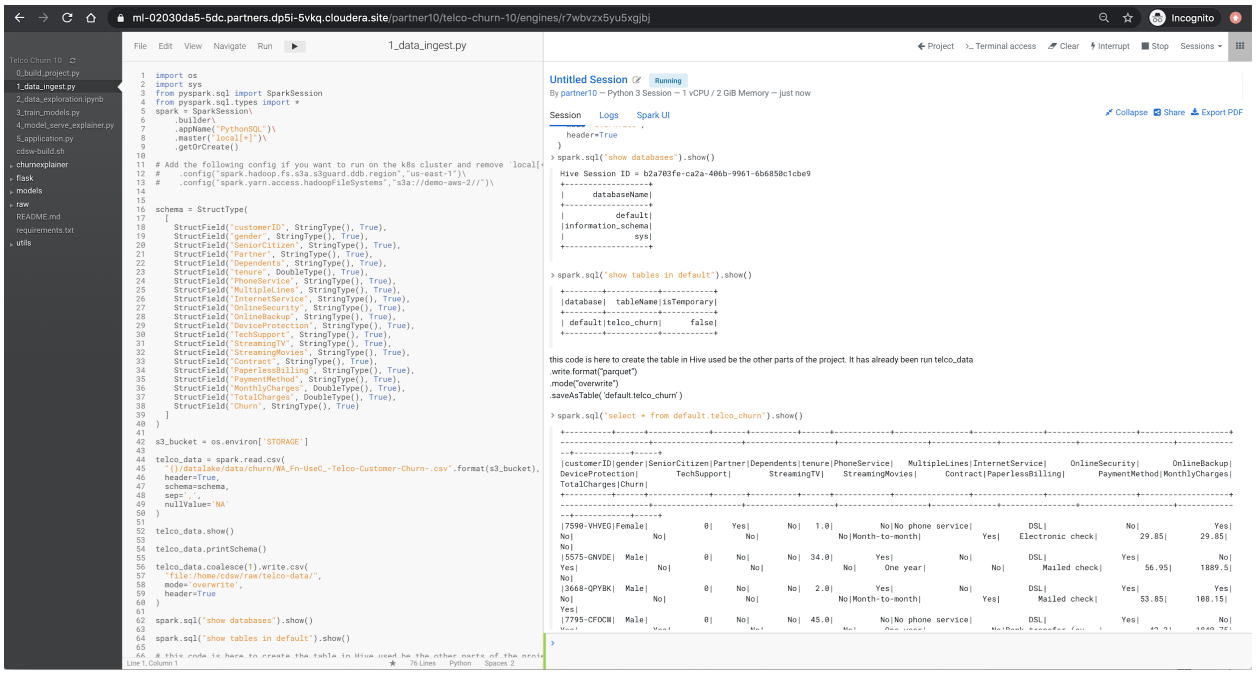
In the same Workbench, open the script “code/*1_data_ingest.py*”

This will load the data from an S3 bucket using Spark. It demonstrates how to read from files and tables using Spark file and SQL operators.

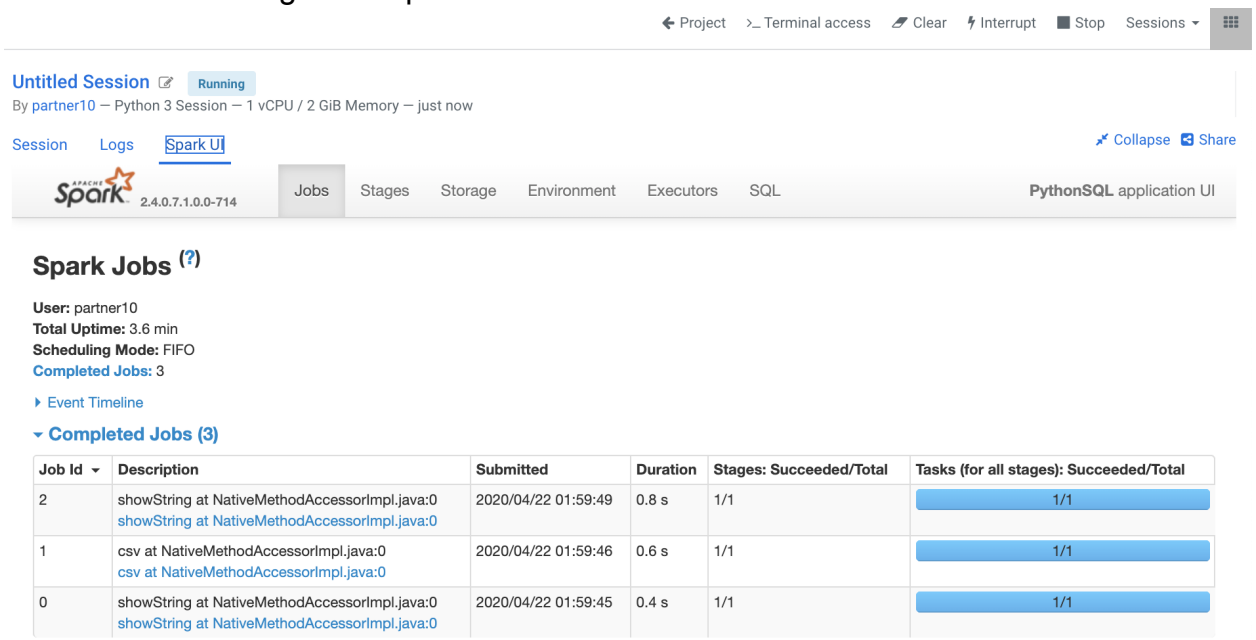


Click on Run → Run All.

Session output will show the code execution results. Observe the database, table, and data from the table.



Also examine the logs and Spark UI for details of the run.



This concludes Lab 1 - Data Ingest.

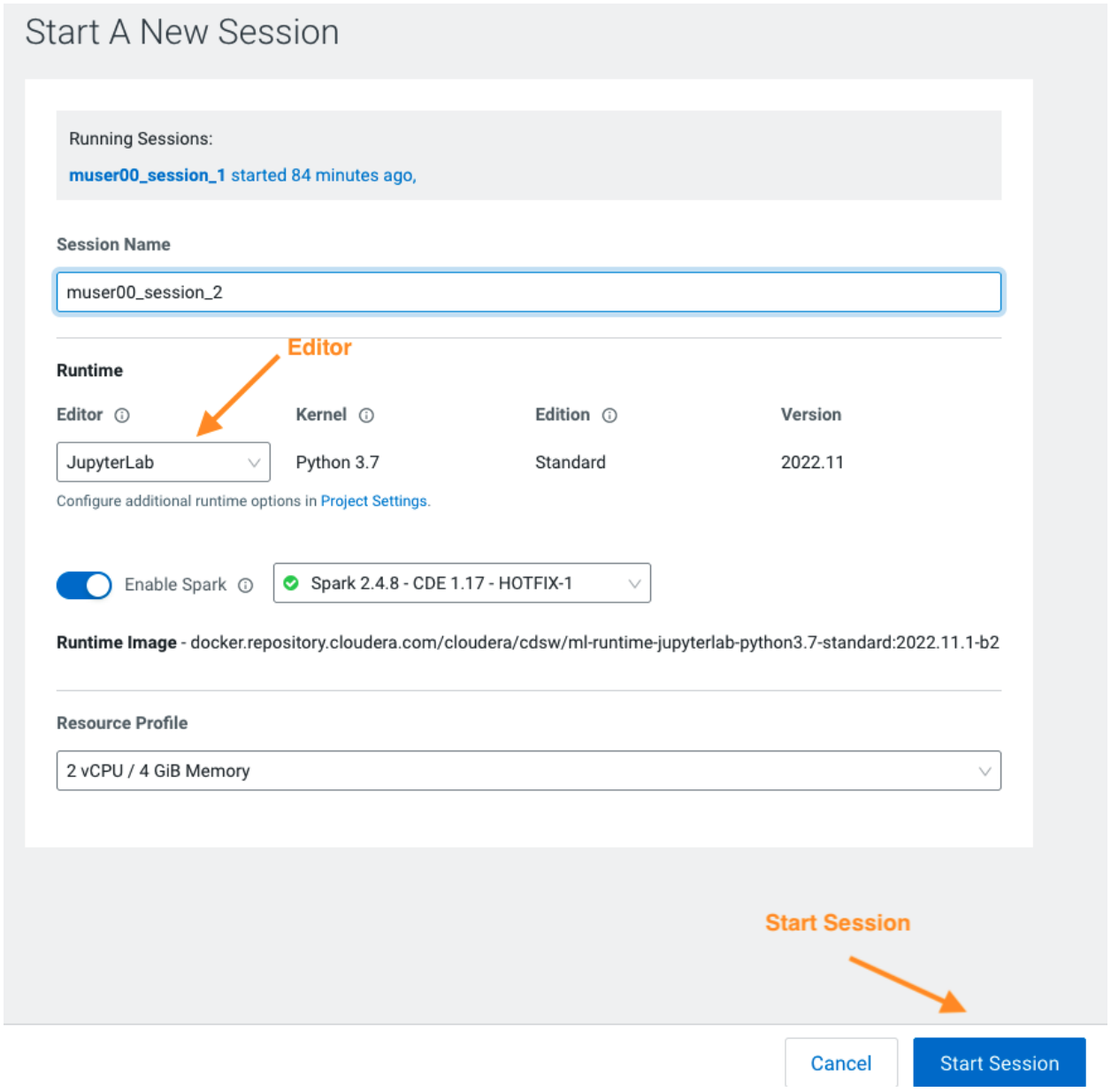
Lab 3 - Data Exploration

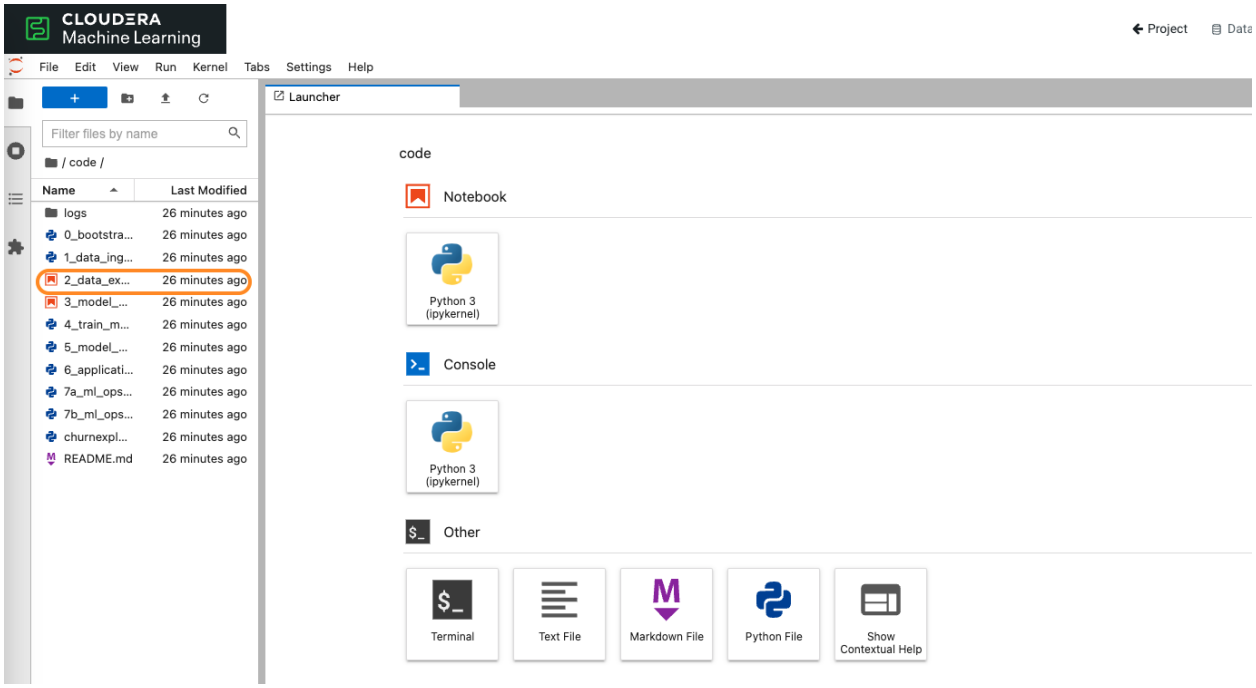
In this lab, you will explore some dataset using a different editor from the previous lab.

In fact, in this lab we are going to use a popular notebook, **Jupyter**, to show the flexibility of CML.

Execute the data exploration script

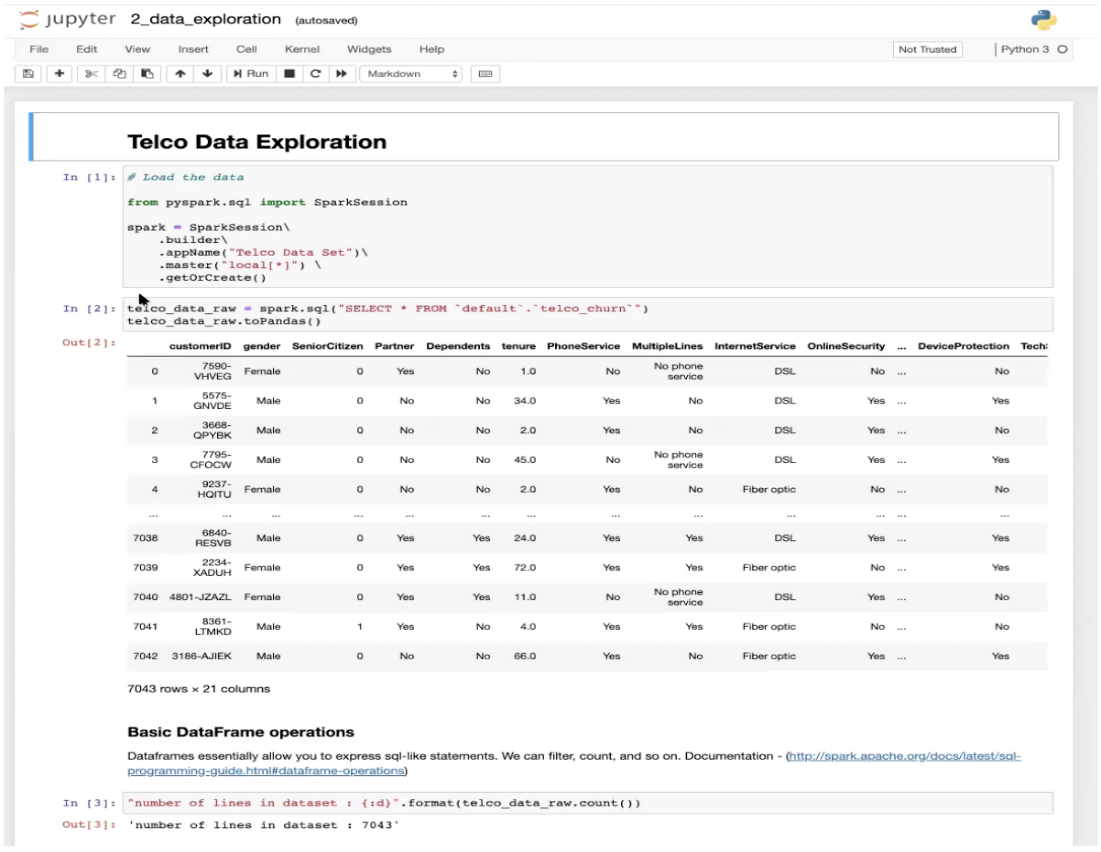
Start a "**NEW SESSION**" and use the below configuration.
Editor: Jupyter Notebook
Engine Profile: *2vCPU x 4GiB Memory*


Start A New Session
Running Sessions:
muser00_session_1 started 84 minutes ago,
Session Name
muser00_session_2
Editor
Runtime
Editor
Kernel
Edition
Version
JupyterLab
Python 3.7
Standard
2022.11
Configure additional runtime options in Project Settings.
Enable Spark
Spark 2.4.8 - CDE 1.17 - HOTFIX-1
Runtime Image - docker.repository.cloudera.com/cloudera/cdsw/ml-runtime-jupyterlab-python3.7-standard:2022.11.1-b2
Resource Profile
2 vCPU / 4 GiB Memory
Start Session
Cancel
Start Session

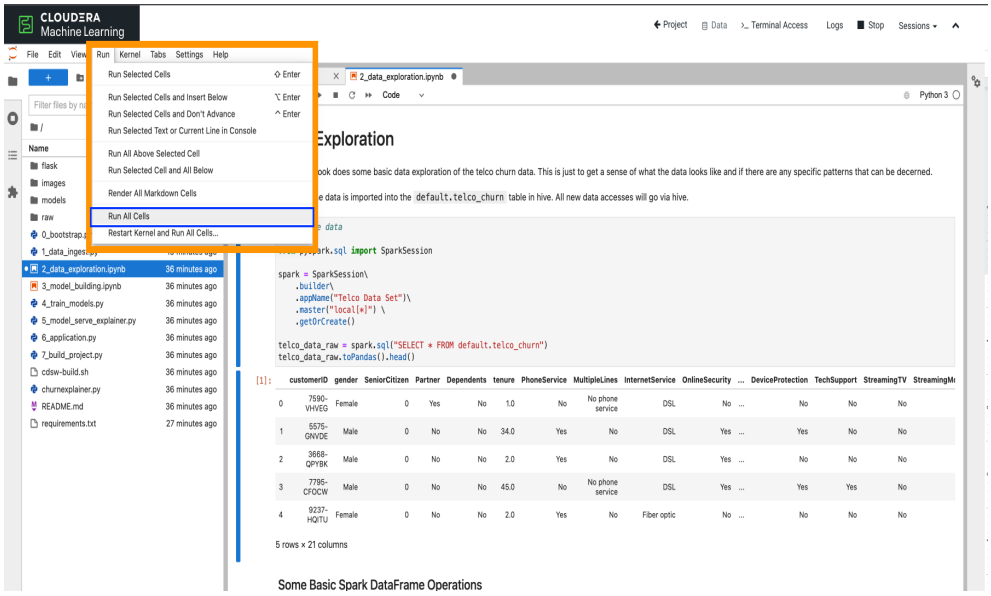


Then click on the ***code/2_data_exploration.ipynb***.

And it will take you into the notebook

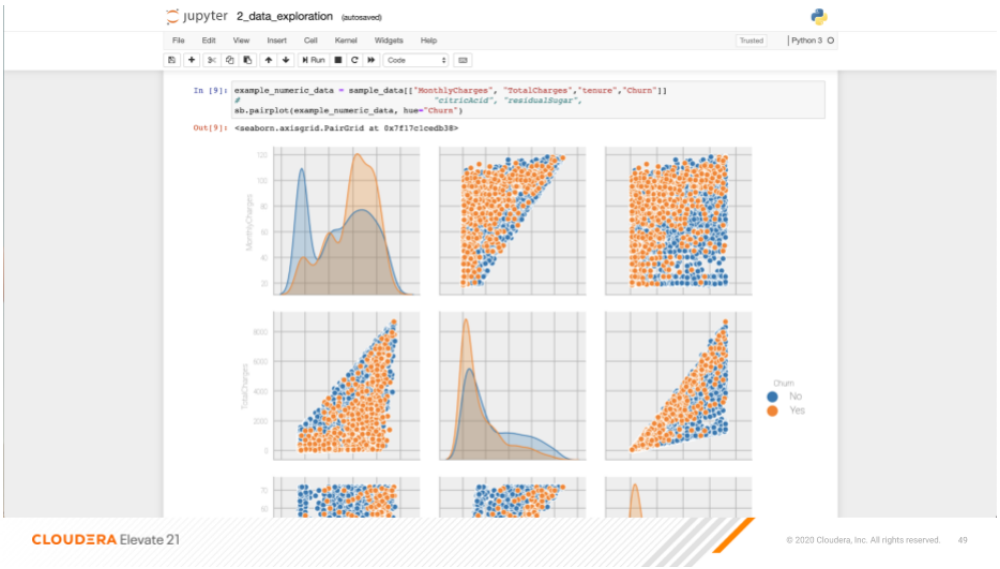


As you notice we are interacting with the data lake, in particular with the database previously created



You are ready to run the notebook, go to *Cell*, *Run All*

And you can analyze the plotted graphs:



Now we can go back to *Project*

This concludes this lab.

Lab 4 - Model Training and Experiment

In this lab, you will build and train the model, using the Experiment feature form CML that allows you to run offline different training sessions, with different parameters configuration, for your model so that you could promote in “Production” that configuration that showed the best results, KPIs.

Build the model

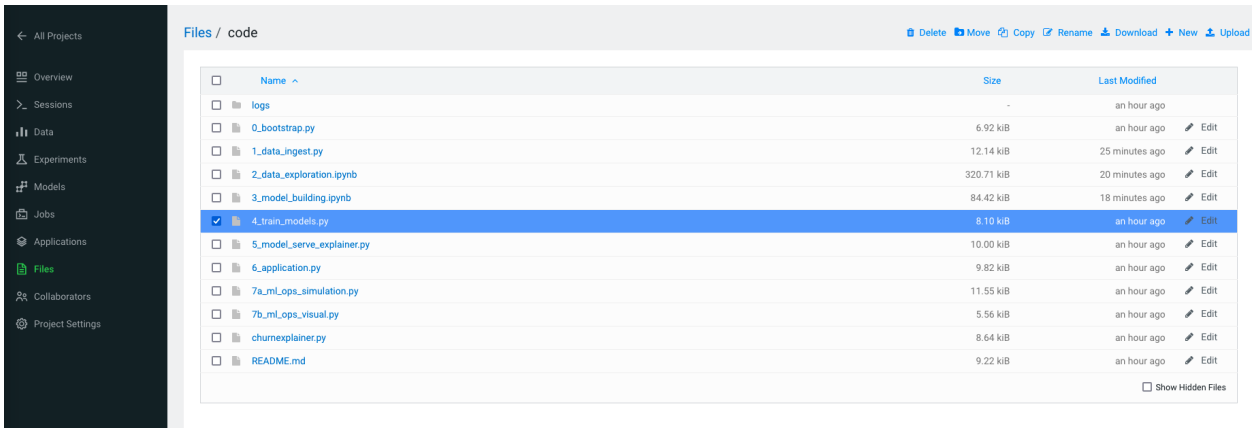
First we will build the model. We will use a Jupyter Notebook to show the process of selecting and building the model to predict churn. It also shows more details on how the LIME model is created and a bit more on what LIME is actually doing.

Open a Jupyter Notebook session (rather than a workbench): python 3.7, 2 vCPU, 4 GiB and open the *3_model_building.ipynb* file.

At the top of the page click **Cells > Run All**.

Model Training and Experiments

For the training portion of the lab we will use the file *4_train_models.py*

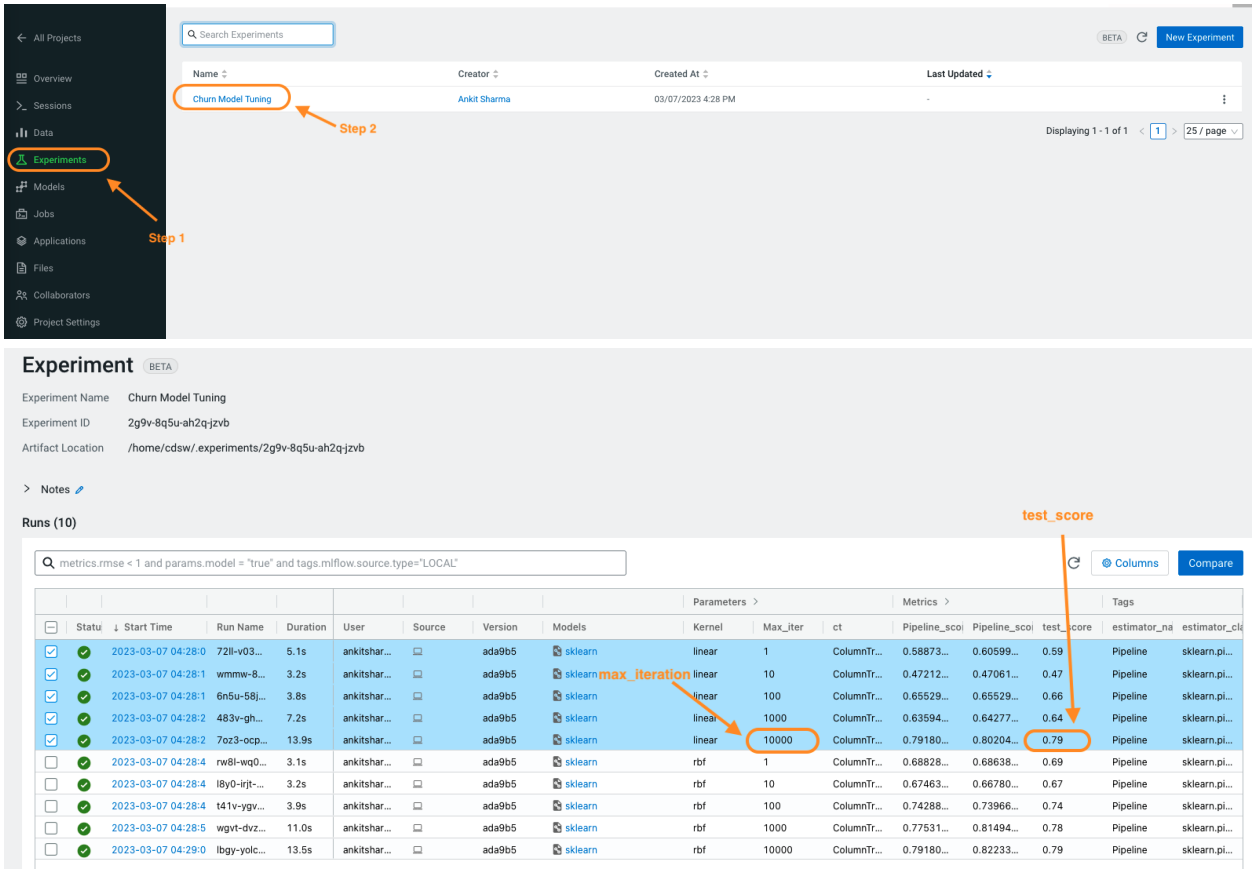


Click on it and familiarize yourself with the code.

To give Data Scientists flexibility to collect, record, and compare experiment runs, CML provides out-of-the-box mlflow Experiments as a framework to achieve this.

Inside a running Workbench session, navigate to code/4_train_model.py. This script uses “kernel” and “max_iter” as the two parameters to manipulate during model training in order to achieve the best result. In our case, we’ll define “best” as the highest “test_score”.

To compare experiments, click on Experiments and select **Churn Model Tuning** experiment.

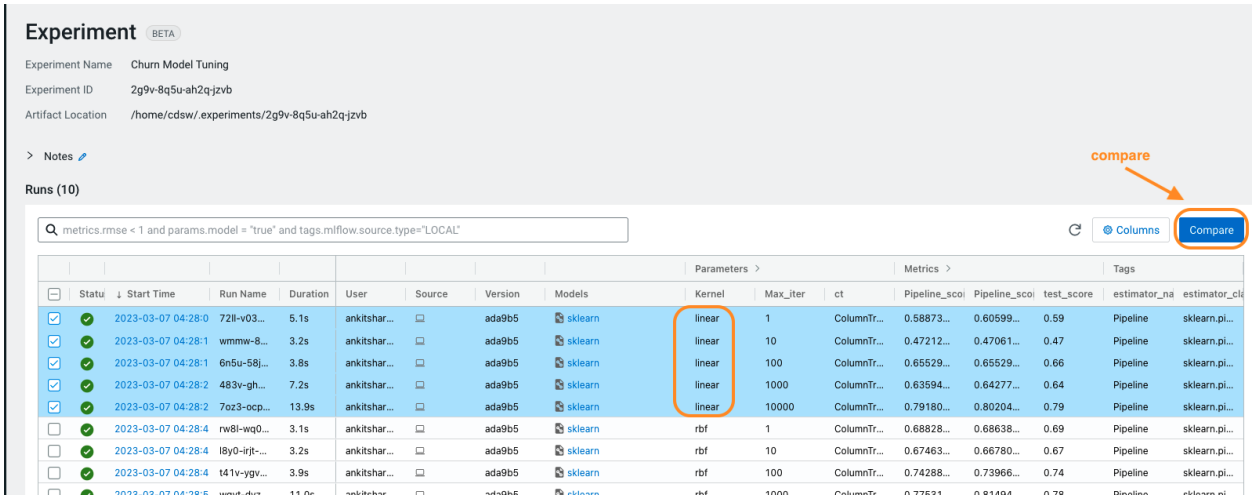


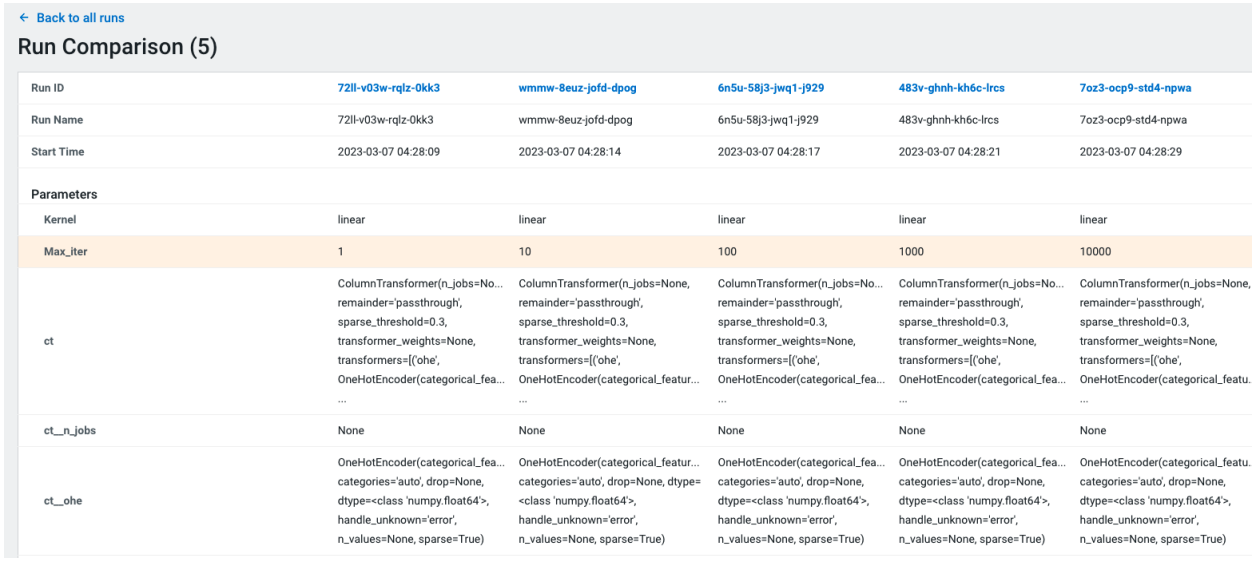
As expected, a higher number of max_iterations produces better results (higher test_score). Interestingly, the choice of kernel does not make a difference at higher max_iter values. We can choose linear as it allows for faster model training.

Compare Experiments

This feature provides a built-in visualizations in mlflow allow for more detailed comparison of various experiment runs and outcomes.

- Select all runs with “linear” Kernel.
- Click on Compare





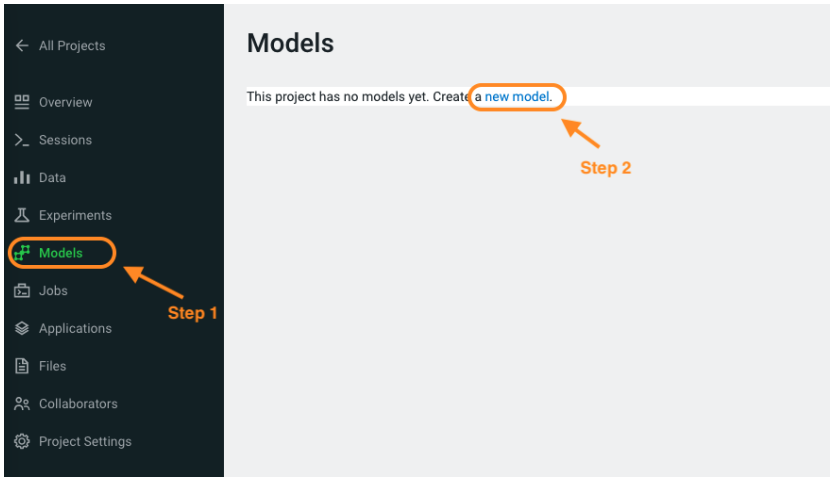
Lab 5 - Model Deploy/Serve

Once a model is trained its predictions and insights must be put to use so they can add value to the organization. Generally this means using the model on new, unseen data in a production environment that offers key ML Ops capabilities.
In this lab, you will deploy/serve the model that you have trained in the Lab 3 as a REST endpoint. The model can be invoked as-needed, in real-time or batch fashion, by external services that need to score the prediction implemented by the model.

Build a Model

Below are the steps to deploy a near-real-time scoring model:

- Click on Models in the side panel
- Create a new model.



- Name your model *Churn Model API Endpoint* . Any other name will cause issues with downstream scripts.
- Uncheck Enable Authentication.
- Under File select *code/5_model_serve_explainer.py*
- Under Function enter *explain*
- For Example Input enter the following JSON
- You do not need to Enable Spark for model serving in this case
- Deploy the model by clicking *Deploy model*

```
{
  "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"
}
```

Name *

Churn Model API Endpoint

Description *

Churn Model API Endpoint

Enable Authentication

Enforces model API requests to be authenticated with an API key.

Build

File *

code/5_model_serve_explainer.py

Function *

explain

Example Input

```
{
  "StreamingTV": "No",
  "MonthlyCharges": 70.35,
  "PhoneService": "No",
  "PaperlessBilling": "No",
  "Partner": "No",
```

Example Output

```
{ "result": "value" }
```

Runtime

Editor	Kernel	Edition	Version
JupyterLab	Python 3.7	Standard	2022.11

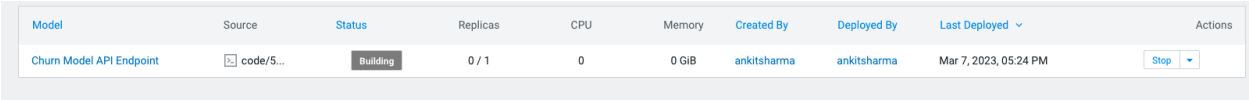
Configure additional runtime options in Project Settings.

Enable Spark

Spark 2.4.8 - CDE 1.17 - HOTFIX-1

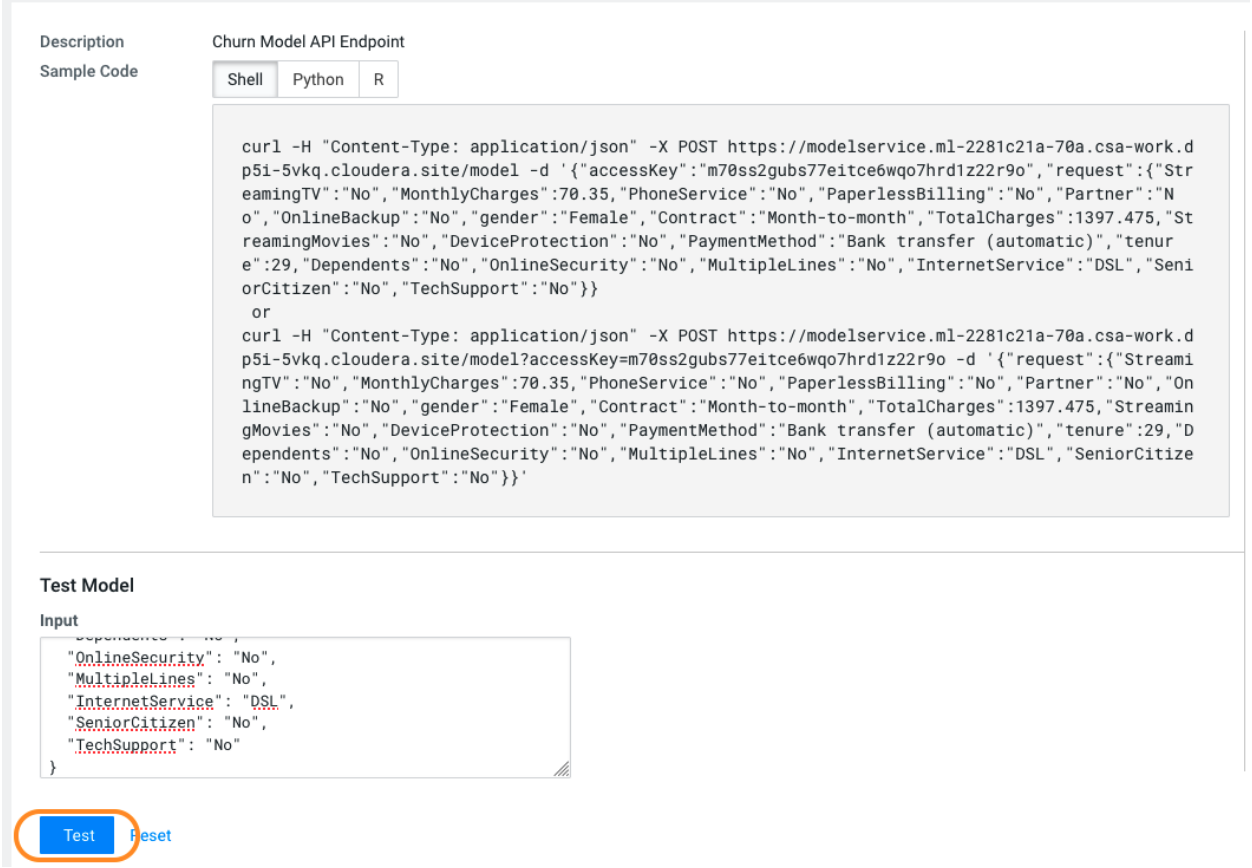
Deploy a Model

The status will go thru the life-cycle of *Pending -> Building -> Deploying -> Deployed*

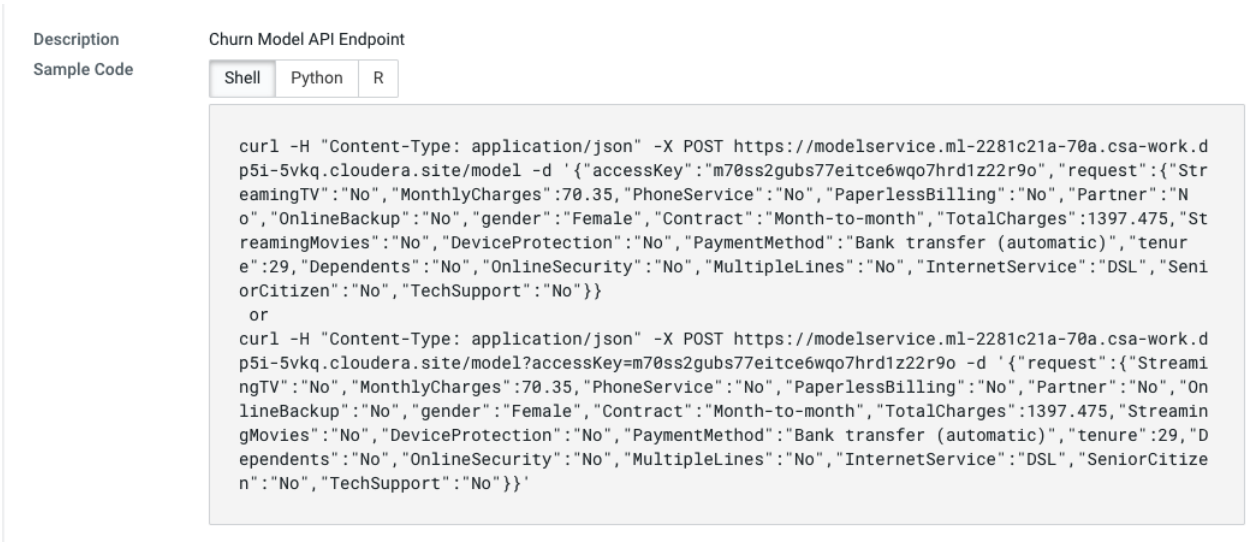


Test a Model

- Once Model is deployed, click on Test



The test simulates a request submission to the Model endpoint. The model processes the input and returns the output along with metadata and a prediction for the customer.
If you want to call the model from external services, the sample codes for invoking this REST endpoint are provided in Shell, Python and R.



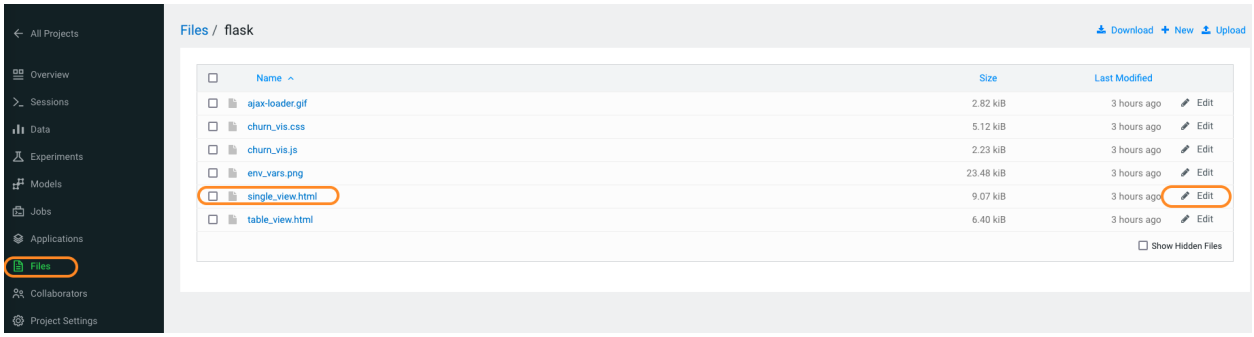
This concludes this lab.

Lab 6 - Application Deployment

In this lab, you will create an application that embeds the model deployed in the previous lab, allowing business users, end-users that are not Data Scientists to interact and to get insight about the context of these analyses.

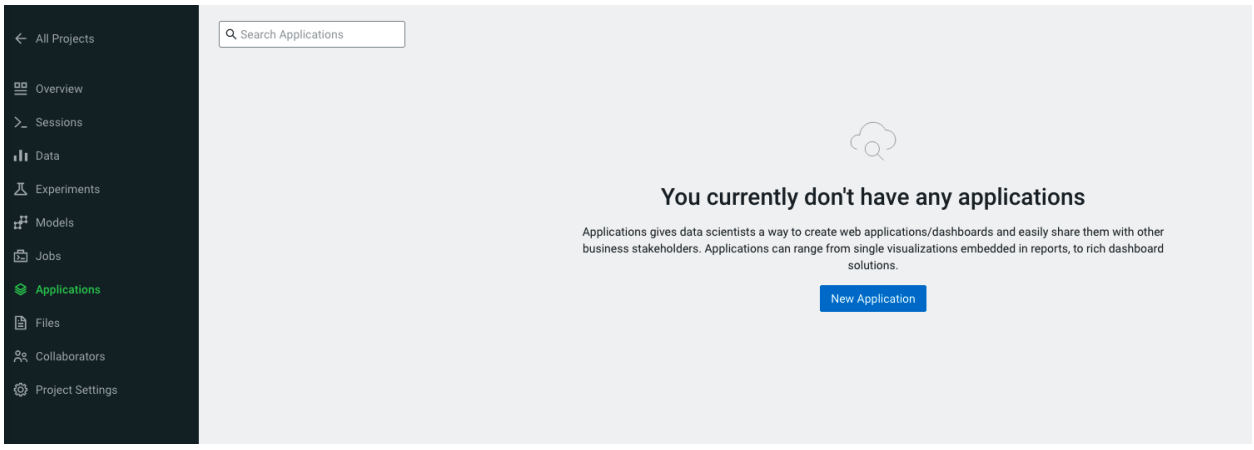
Prerequisites

- Copy accesskey *Models -> Settings -> Access Key*
- Update accesskey into file. *Files -> flask -> single_view.html (line : 61)*

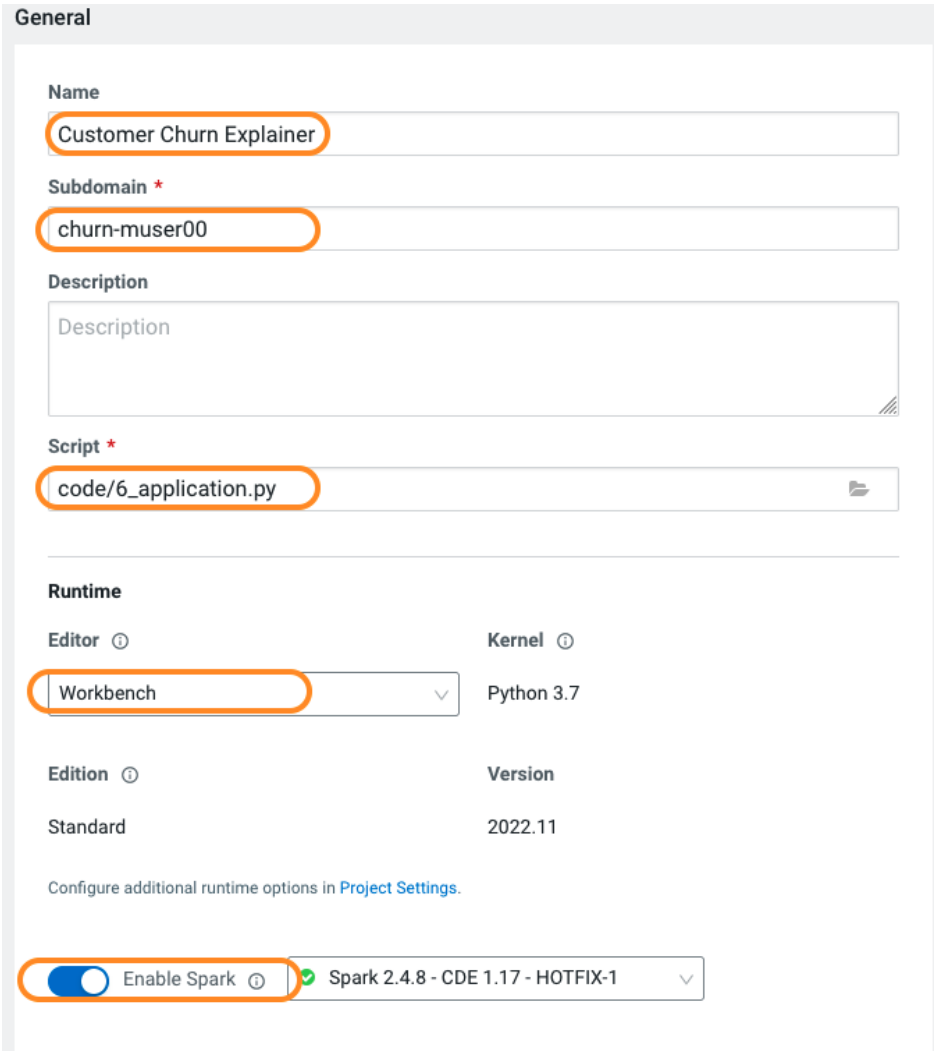


Application Deployment

Go to *Applications* and create a new Application.

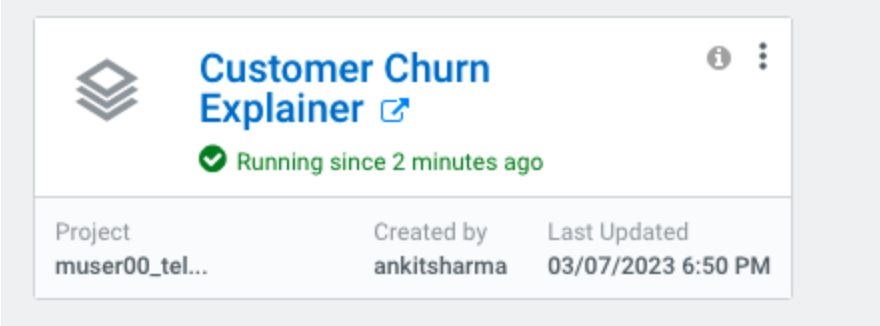


Name	Customer Churn Explainer
Subdomain	churn-<uesrname>
Script	code/06_application.py
Editor	Workbench
Enable Spark	Yes



- Click on Create Application

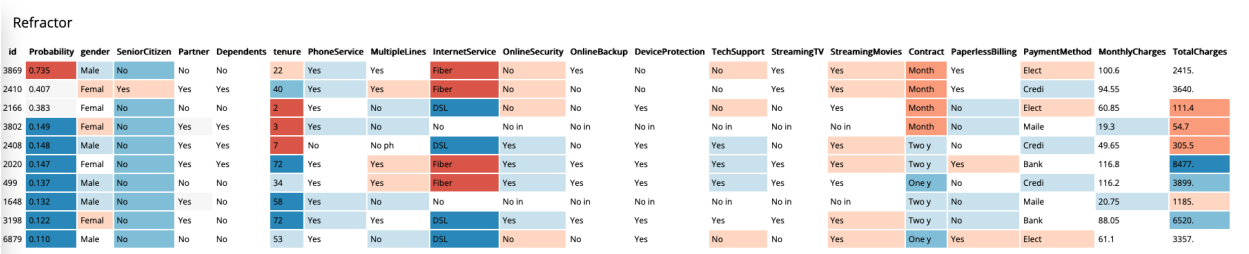
Application startup can take up to 2 minutes, and once the application is ready you'll see a card similar to this:



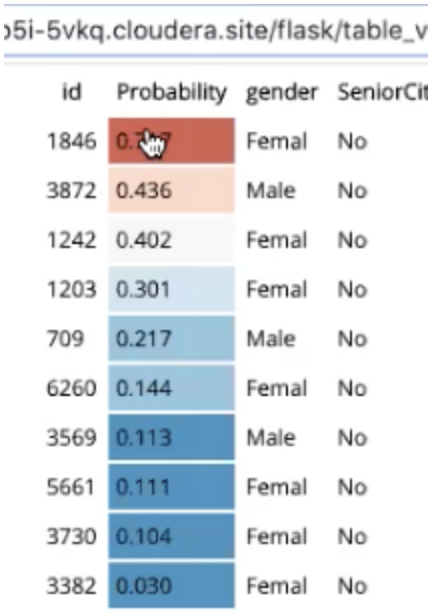
Click now in your newly created application

You can see the subdomain we have specified before as a prefix of your application url.

Once the application is loaded



Click on one of the item in the Probability column



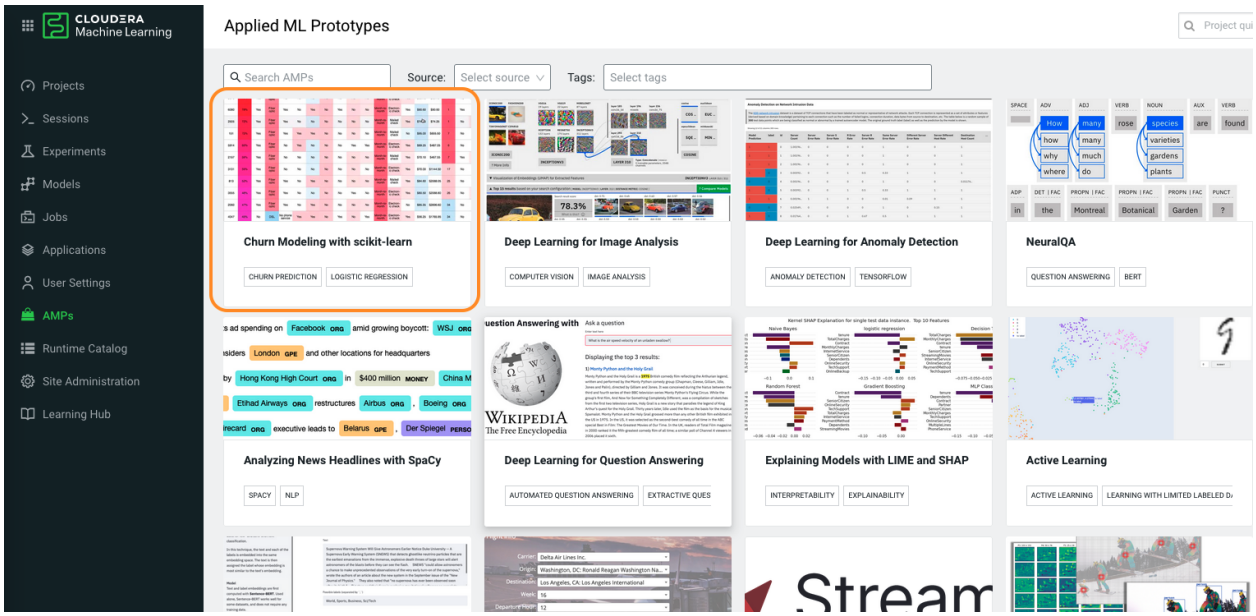
To get the detailed view

If you change some of the values, that will also change the churn probability by calling the model we have deployed in the previous lab.

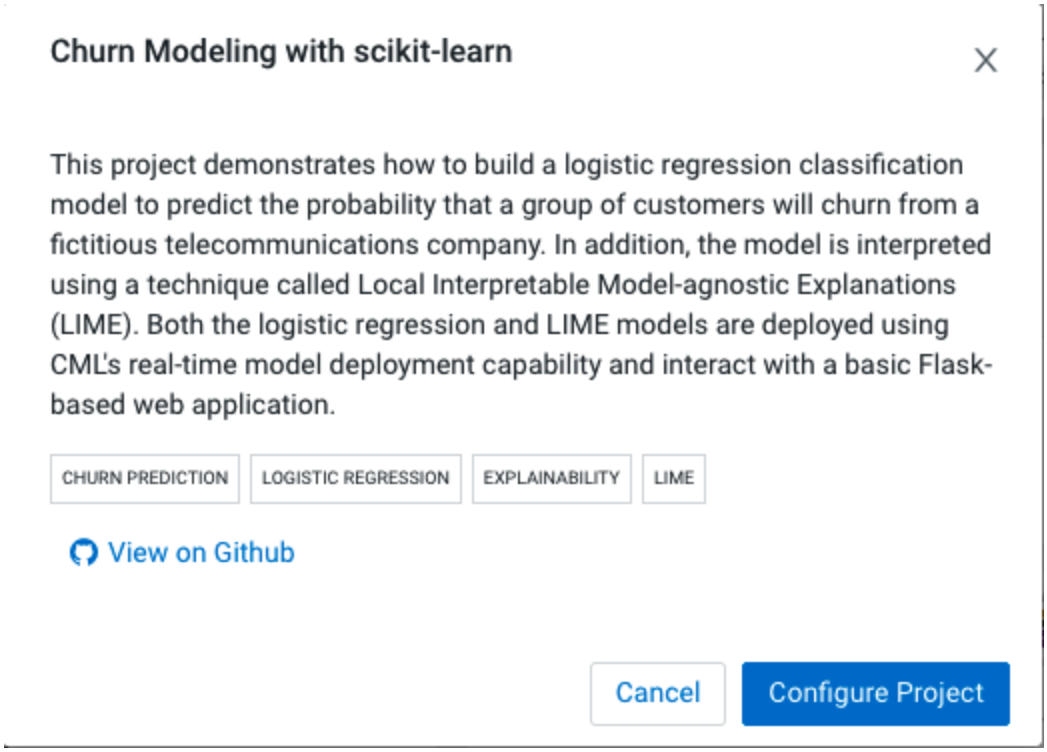
This concludes the lab.

Demo - Applied ML Prototypes (AMPs)

AMPs (Applied Machine Learning Prototypes) are reference Machine Learning projects that have been built by Cloudera Fast Forward Labs to provide quickstart examples and tutorials. AMPs are deployed into the Cloudera Machine Learning (CML) experience, which is a platform you can also build your own Machine Learning use cases on.



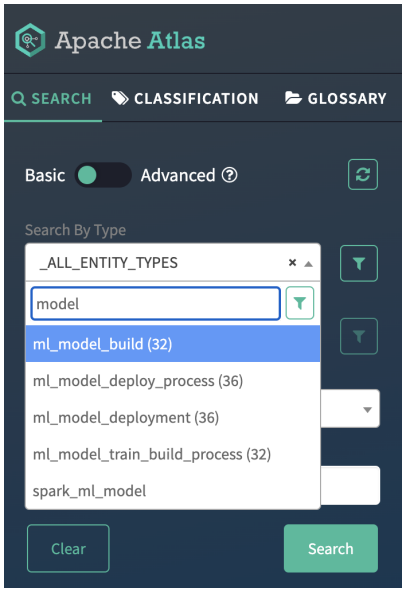
Configure the Project.



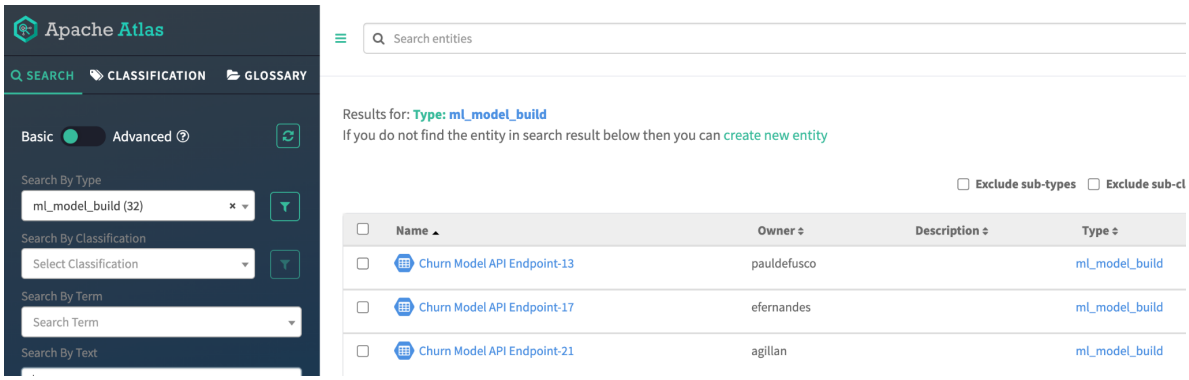
Demo- Model Lineage Tracking

CDP is an end-to-end hybrid enterprise data platform. Every user, workload, and dataset and machine learning model can be governed from a central location via SDX, the Shared Data Experience.

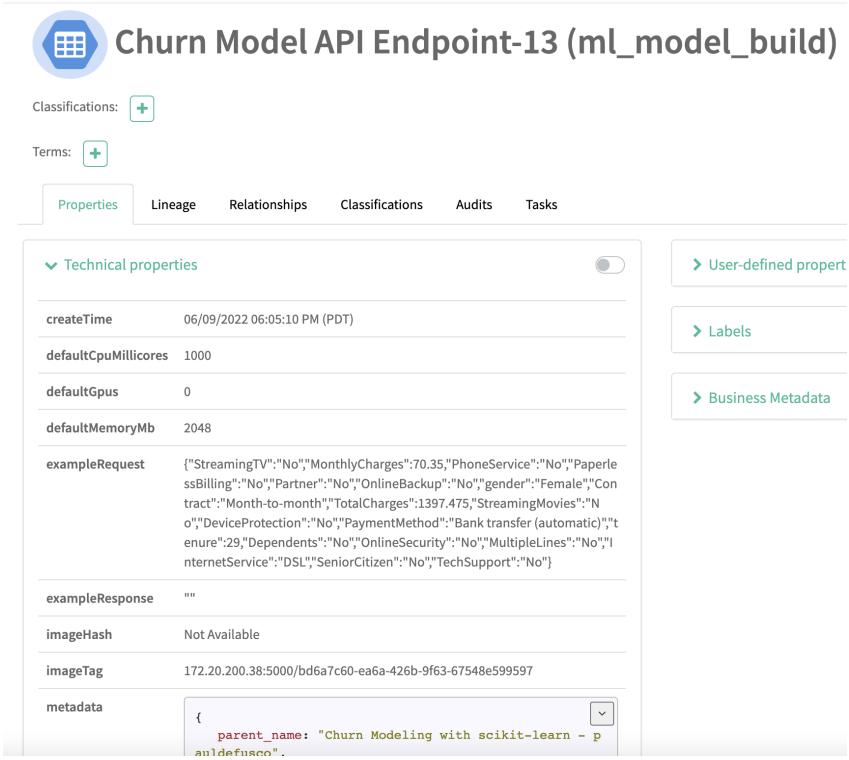
From the Atlas UI, search for ML models by entering the “ml_model_build” type. Notice that there are various Atlas entities to browse for models.

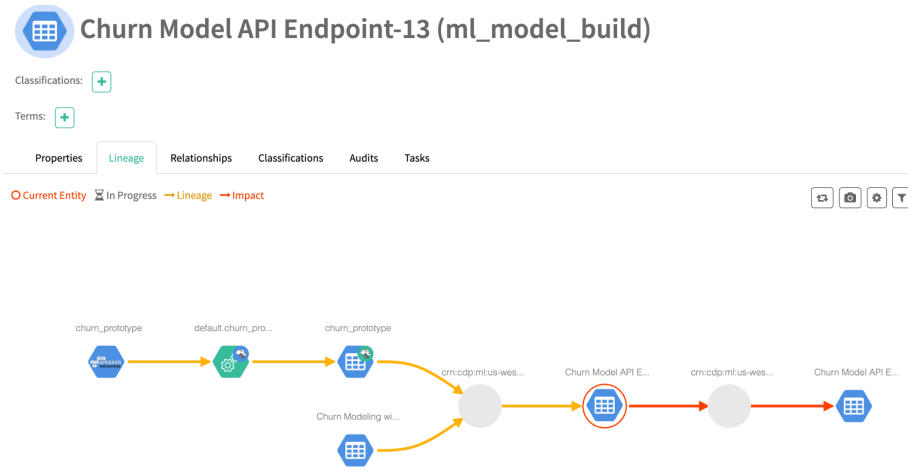


In the output, you will see all models that your colleagues deployed in this workshop. Notice that each model is assigned a unique ID at the end. That ID corresponds to the Model Build from CML. Identify your model using the Build Id noted down when you deployed your model. select the model you created.



Familiarize yourself with the Model properties tab. Notice that each model logged is associated with rich metadata.


Churn Model API Endpoint-13 (ml_model_build)
Classifications:
Terms:
Properties
Lineage
Relationships
Classifications
Audits
Tasks
Technical properties
User-defined propert
Labels
Business Metadata
createTime
06/09/2022 06:05:10 PM (PDT)
defaultCpuMillicores
1000
defaultGpus
0
defaultMemoryMb
2048
exampleRequest
{"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"}
exampleResponse
""
imageHash
Not Available
imageTag
172.20.200.38:5000/bd6a7c60-ea6a-426b-9f63-67548e599597
metadata
{
parent_name: "Churn Modeling with scikit-learn - p
auldefusco",


Churn Model API Endpoint-13 (ml_model_build)
Classifications:
Terms:
Properties
Lineage
Relationships
Classifications
Audits
Tasks
Current Entity
In Progress
Lineage
Impact
churn_prototype
default.churn_pro...
churn_prototype
crn:cdp:ml:us-wes...
Churn Model API E...
crn:cdp:ml:us-wes...
Churn Model API E...
Churn Modeling wi...