# cloudera

# Modern Platform for Machine Learning and Analytics

Johannes Muselaers I Sales Engineer <u>jmuselaers@cloudera.com</u> +46725881091

#### Agenda

Cloudera Data Science Workbench (CDSW) Demo

- Introduction
- Current challenges for LEGO executing Data Science Workloads
- Introduction to CDSW
- CDSW Workshop
- Next steps

# Why CDSW?

#### Integral part of CDP

**Enterprise Graded** 

Security & Governance

Data Science at Scale from One Vendor

Multi-Cloud On-Premise PaaS

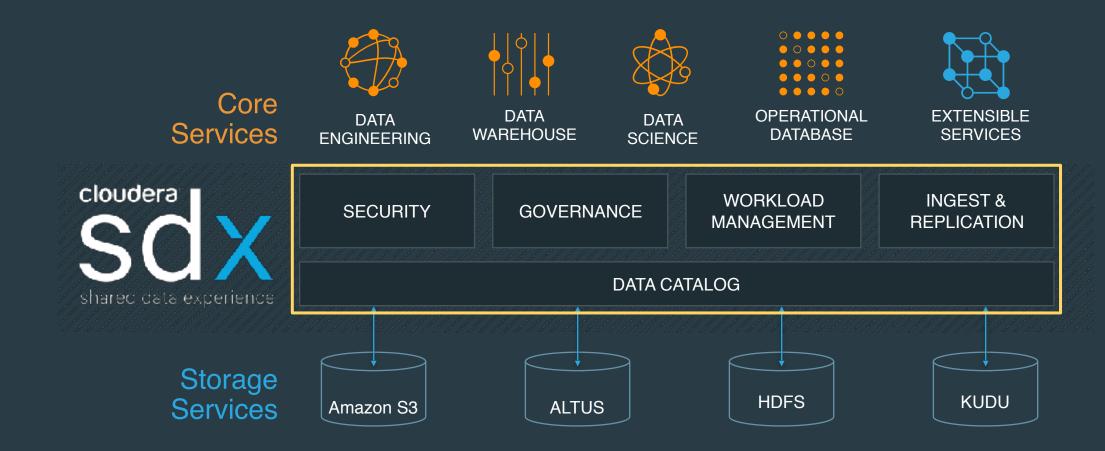
World Class
Support and
Professional
Services

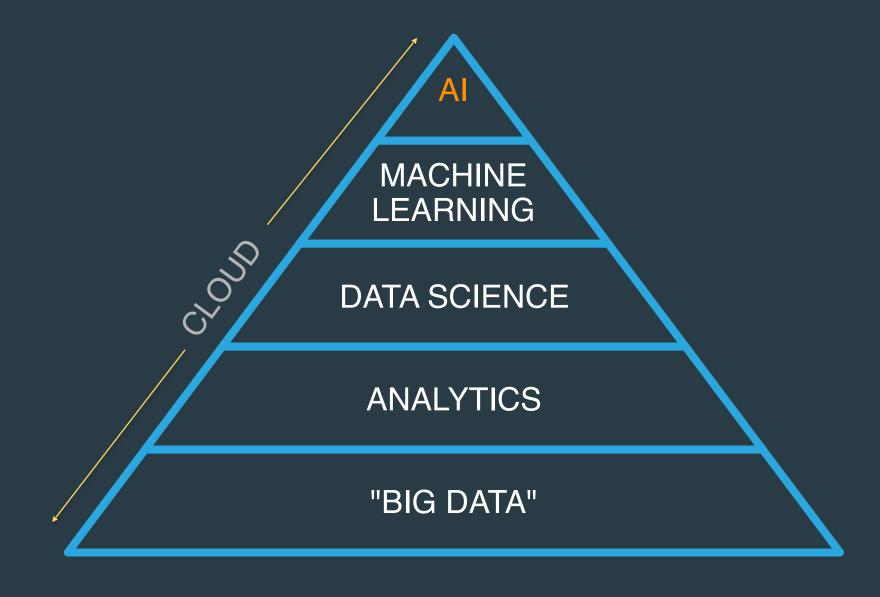
3000 Clouderan's 5000 Partners Partner Solutions CDH /HDP 100%
Open Source No
Lock in
Enterprise Add-ons

Technology Leader Hadoop, Impala, Kudu, Spark

## Modern Platform for Machine Learning and Analytics

Integrated data, workflows, metadata, security, governance, ...





# Cloudera Data Science Workbench



#### MACHINE LEARNING AT CLOUDERA

Our philosophy

We empower our customers to run their business on data with an open platform:

- Your data
- Open algorithms
- Running anywhere

We accelerate enterprise data science.



#### What is Cloudera Data Science Workbench (CDSW)

Supports your workloads in an enterprise secure way

# Accelerate data science from exploration to production using R, Python, Spark and more

#### For data scientists



Open data science, your way.

Use R, Python, or Scala with your favorite libraries and frameworks



No need to sample.

Directly access data in secure Hadoop clusters through Apache Spark and Apache Impala



Reproducible, collaborative research.

Share insights with your whole team

#### For IT professionals



Bring analysis to the data.

Give your data science team the freedom to work how they want, when they want



Secure by default.

Stay compliant with out-of-the-box support for full Hadoop security



Flexible deployment.

Run on-premises or in the cloud

#### Data Science & Data Scientist



## THE CHALLENGE

#### Balance these needs

#### DATA SCIENCIST

- Access to granular data
- Flexibility
  - Preferred open source tools
- Elastic provisioning
  - Compute
  - Storage
- Reproducible research
- Path to production



#### DevOps/IT

- Security
- Governance
- Standards
- Low maintenance
- Low cost
- Self-service access

WS,

#### THE TYPICAL SOLUTION

#### "If I can't use my favorite tools, I'll..."

- Copy data to my laptop
- Copy data to a data science appliance
- Copy data to a cloud service

#### Why this is a problem:

- Complicates security
- Breaks data governance
- Adds latency to process
- Makes collaboration more difficult
- Complicates model management and deployment
- Creates infrastructure silos

#### CLOUDERA DATA SCIENCE WORKBENCH

#### Accelerate Machine Learning from Research to Production





#### For data scientists

- Experiment faster
   Use R, Python, or Scala with
   on-demand compute and
   secure CDH data access
- Work together
   Share reproducible research
   with your whole team
- Deploy with confidence
   Get to production repeatably
   and without recoding

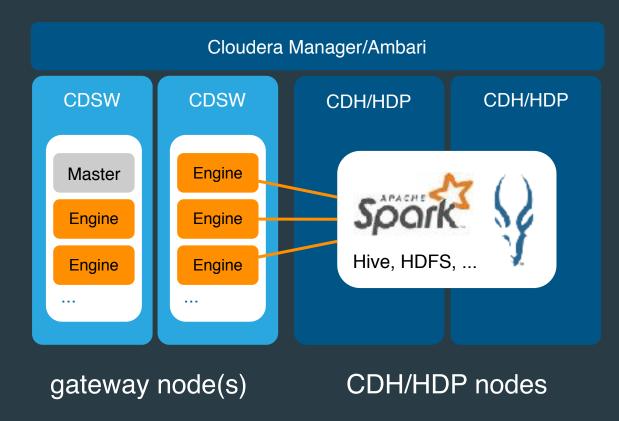
#### For IT professionals

- Bring data science to the data
   Give your data science team
   more freedom while reducing
   the risk and cost of silos
- Secure by default
   Leverage common security and governance across workloads
- Run anywhere
   On-premises or in the cloud

#### A MODERN DATA SCIENCE ARCHITECTURE

Containerized environments with scalable, on-demand compute

- Built with Docker and Kubernetes
  - Isolated, reproducible user environments
- Supports both big and small data
  - Local Python, R, Scala runtimes
  - Schedule & share GPU resources
  - Run Spark, Impala, and other CDH services
- Secure and governed by default
  - Easy, audited access to Kerberized clusters
  - Leverages SDX platform services
- Deployed with Cloudera Manager/Ambari



#### ACCELERATED DEEP LEARNING WITH GPUS

#### Multi-tenant GPU support on-premises or cloud

"Our data scientists want GPUs, but we need multi-tenancy. If they go to the cloud on their own, it's expensive and we lose governance."

- Extend CDSW to deep learning
- Schedule & share GPU resources
- Train on GPUs, deploy on CPUs
- Works on-premises or cloud



single-node training



distributed training, scoring

GPU On CDH coming in C6

#### WHAT DATA SCIENCE TEAMS DO

#### PREPARE DATA

Ingest data at scale.

Store and secure data.

Clean and transform data for analysis.

#### **BUILD MODELS**

Explore data and build predictive models, offline.

Evaluate and tune models.

Develop and deliver a modeling pipeline.

#### **DEPLOY MODELS**

Test, verify, and approve model for deployment.

Create and maintain batch/ stream pipelines, embedded models, APIs.

Update models in production.

#### CLOUDERA DATA SCIENCE WORKBENCH

Accelerate and simplify machine learning from research to production





邸

#### **ANALYZE DATA**

· Explore data securely and share insights with the team



#### TRAIN MODELS

 Run, track, and compare reproducible experiments



#### **DEPLOY APIs**

 Deploy and monitor models as APIs to serve predictions



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

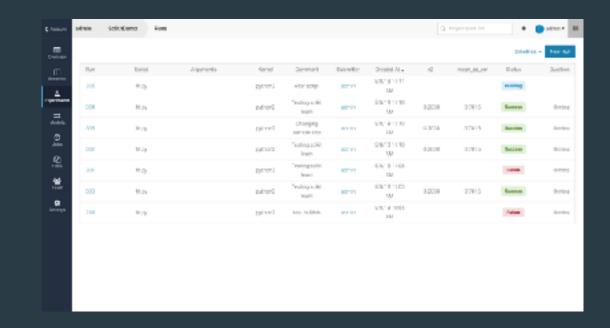


#### INTRODUCING EXPERIMENTS

Versioned model training runs for evaluation and reproducibility

#### Data scientists can now...

- Create a snapshot of model code, dependencies, and configuration necessary to train the model
- Build and execute the training run in an isolated container
- Track specified model metrics, performance, and model artifacts
- Inspect, compare, or deploy prior models



#### INTRODUCING MODELS

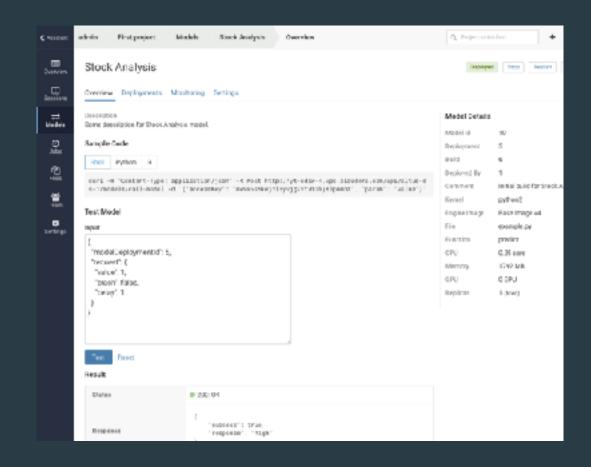
#### Machine learning models as one-click microservices (REST APIs)

- 1. Choose file, e.g. score.py
- 2. Choose function, e.g. forecast

```
f = open('model.pk', 'rb')
model = pickle.load(f)
def forecast(data):
    return model.predict(data)
```

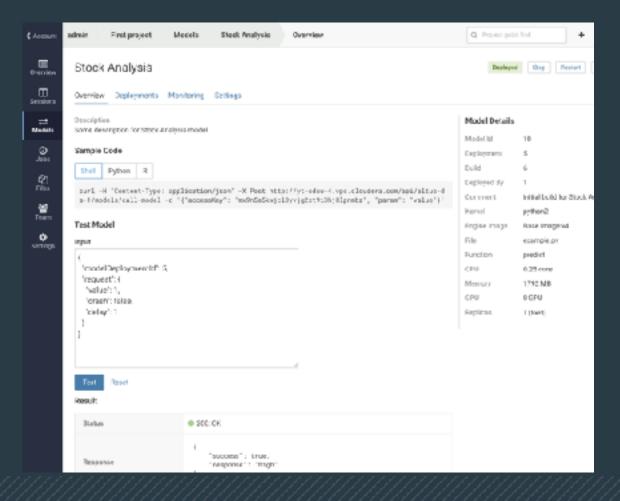
- 3. Choose resources
- 4. Deploy!

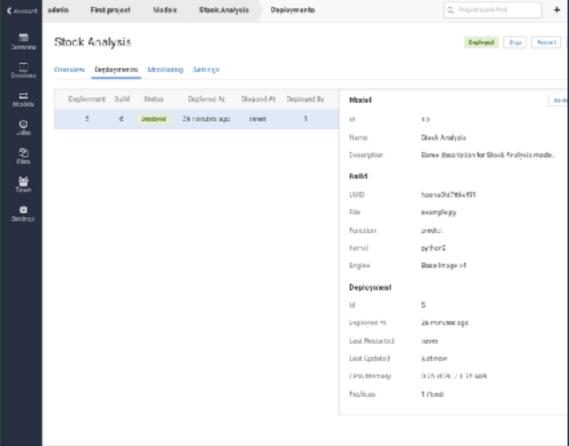
Running model containers also have access to CDH for data lookups.

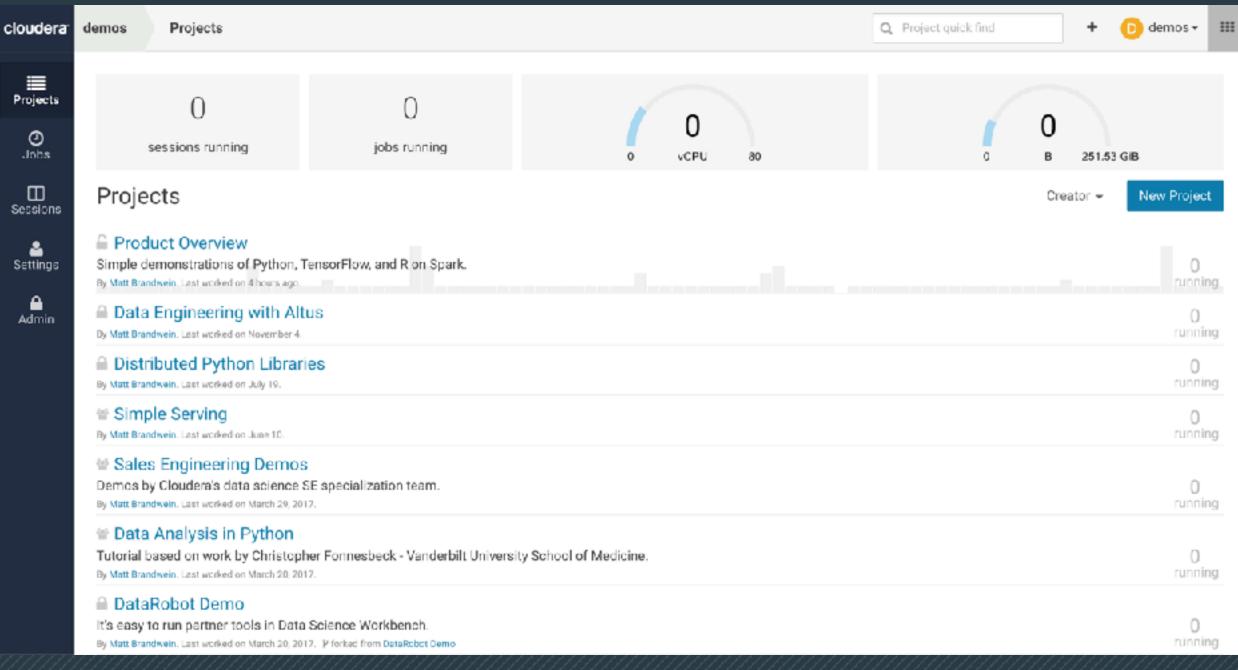


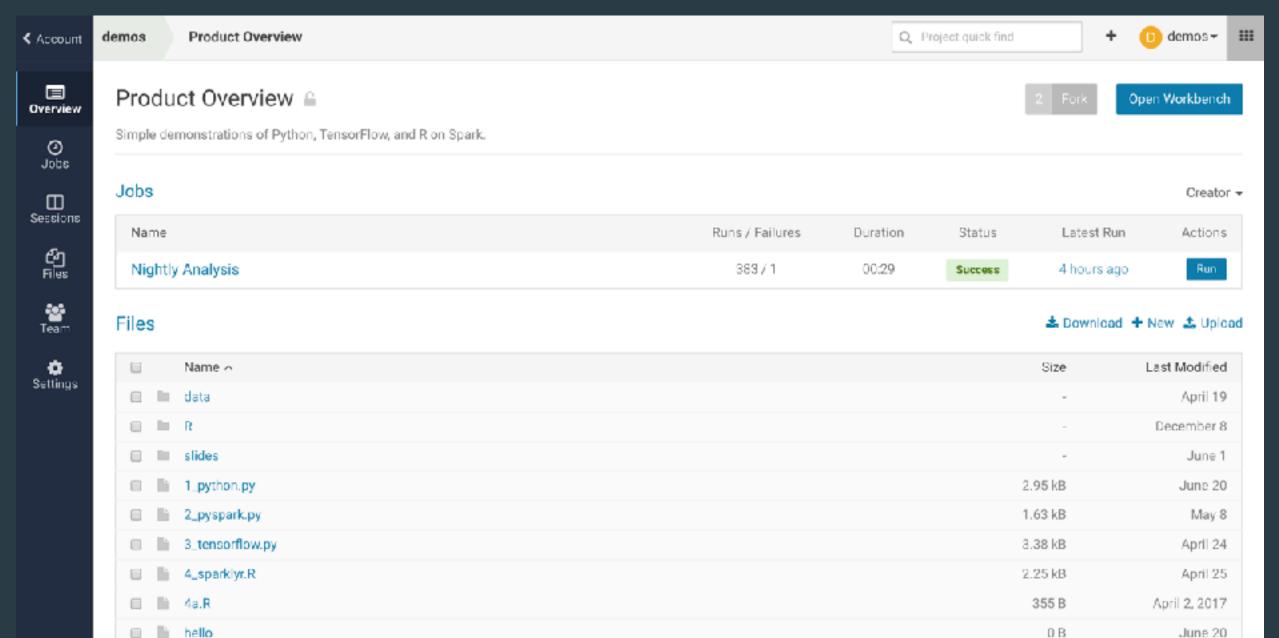
#### MODEL MANAGEMENT

### View, test, monitor, and update models by team or project









data.head()

41

```
2_pyspark.py
```

```
1_python.py
```

```
1_python.py
```

2\_pysperk.py

3\_tensorflow.py

4\_sparklyr.R

45.R

▼ deta.

GoogleTrendsData.csv

kmeans\_data.txt

MNIST

hello

⊪ R

README.md

⊩ slides i

```
# Goodle Stock Analytics
   # -----
   # This notebook implements a strategy that uses Google
 5 # trade the Dow Jones Industrial Average.
   import pandas as pd
   import matplotlib.pyplot as olt
   import matplotlib as mpl
10 from pandas_highcharts.display import display_charts
11 import seaborn
12 mpl.rcParams['font.family'] = 'Source Sans Pro'
   mpl.rcParams['axes.labelsize'] = '15'
14
15 # Import Data
16 # ========
17 #
18 # Load data from Google Trends.
19
   data = pd.read_csv('data/GoogleTrendsData.csv', index_
   data.head()
22
   # Show DJIA vs. debt related query volume.
   display_charts(data, chart_type="stock", title="DJIA v
   seaborn.lmplot("debt", "djia", data=data, size=7)
25
27 # Detect if search volume is increasing or decreasing
28 # any given week by forming a moving average and testil
29 # crosses the moving average of the past 3 weeks.
39 #
31 # Let's first compute the moving average.
32
33
   data['debt_mavg'] = data.debt.rolling(window=3, center)
34
   data.head()
35
35 # Since we want to see if the current value is above the
   # *preceeding* weeks, we have to shift the moving aver-
38
39 data['debt_mavq'] = data.debt_mavq.shift(1)
```

#### Start New Session

#### Engine Image - Configure

Base Image v1 - docker.repository.cloudera.com/cdsw/engine:1

#### Select Engine Kernel

- Python 2
- Python 3
- Scala
- R

#### Select Engine Profile

1 vCPU / 2 GiB Memory

Launch Session

1\_bython.p... ← Project >\_ Terminal access 
Ø Clear 
f Interrupt 
■ Stop Sessions Edit View Navigate Run 2\_pyspark.py 1\_python.py # Google Stock Analytics My Python Session @ \_\_\_\_ 4\_sparklyr.R By Matt Brandwein - Python 2 Session - 1 vCPU / 2 GiB Memory -Running # This notebook implements a strategy that uses Google just now 5 # trade the Dow Jones Industrial Average. Product Overview 2 Cloudera Deta Science Workbench Terminal import pandas as pd ① tty-1gp7dhoexb4sn8fr.cdsw.edh.cloudera.com/cky3o8ggcyi0n49p/ import matplotlib.pyplot as plt GOOGLE Welcome to Cloudera Data Science Workbench 2\_pyspark.py import matplotlib as mpl 10 from pandas\_highcharts.display import display\_charts 3\_tensorflow.py This notebook Kernel: python2 import seaborn 4\_sparklyr.R 12 mpl.rcParams['font.family'] = 'Source Sans Pro' Industrial Aver mpl.rcParams['axes.labelsize'] = '16' 4a.R Project workspace: /home/cdsw 14 ▼ deta > import par > import mat
Kerberos principal: mbrandwein@CLOUDERA.LOCAL 15 # Import Data GoogleTrendsData.csv 4 ========= 17 > import matRuntimes: kmeans\_data.txt # Load data from Google Trends. R: R version 3.3.0 (2016-05-03) — "Supposedly Educational" > from panda 19 ▶ MNIST Python 2: Python 2.7.11 data = pd.read\_csv('data/GoogleTrendsData.csv', index\_d hello Python 3: Python 3.6.1 > import sea data.head() ⊪ R 22 Java: java version "1.8.0\_111" > mpl.rcPana # Show DJIA vs. debt related query volume. README.md > mpl.rcPara Git origin: http://github.mtv.cloudera.com/mbrandwein/cdsw-demo-sh display\_charts(data, chart\_type="stock", title="DJIA v seaborn.lmplot("debt", "djia", data=data, size=7) ⊪ slides 25 mport Dcdsw@1qp7dhoexb4snBfr:~\$ ls -al # Detect if search volume is increasing or decreasing total 96 # any given week by forming a moving average and testing utils.pyc drwxr-xr-x 14 cdsw cdsw 4096 Jul 14 - 2017 . 29 # crosses the moving average of the past 3 weeks. Load data fro 39 # 31 # Let's first compute the moving average. > data = pd.read\_csv('data/GoogleTrendsData.csv', index\_col='Date', parse\_dates 32 data['debt\_mavg'] = data.debt.rolling(window=3, center: > data.head() 34 data.head() djia debt 35 36 # Since we want to see if the current value is above the Date # \*preceeding\* weeks, we have to shift the moving aver: 38 2004-01-14 | 10485.18 | 0.210000 data['debt\_mavg'] = data.debt\_mavg.shift(1) data.head() 2004-01-22 | 10528.66 | 0.210000 41 # Generate Orders

43 # -----

# CDSW DEMO

# THANK YOU

cloudera

edsw-build.sh

fit.py

README.mid

```
fit.py
                            Run Line(s)
                                               #Enter
    # fit a simple linear
   # classic iris flower
                            Run All
                                                SE SE
    # length from sepal ]
    # model to the file m
                            Run Experiment...
                                                ⊕ 3¢R
    from sklearn import datasets, linear_model
    from sklearn.metrics import mean_squared_error, r2_score
    import pickle
    import cdsw
10
    iris = datasets.load_iris()
12
    test_size = 20
13
14
   # Train
   iris_x = iris.data[:-test_size, 0].reshape(-1, 1) # sepal length
    iris_y = iris.data[:-test_size, 2].reshape(-1, 1) # petal length
17
    model = linear_model.LinearRegression()
    model.fit(iris_x, iris_y)
20
    # Test and predict
    score_x = iris.data[-test_size:, 0].reshape(-1, 1) # sepal length
    score_y = iris.data[-test_size:, 2].reshape(-1, 1) # petal length
24
    predictions = model.predict(score_x)
26
    # Mean squared error
   mean_sq = mean_squared_error(score_y, predictions)
    odsw.log_metrio("nean_sq_err", nean_sq)
    print("Mean squared error: %.2f"% mean_sq)
31
    # Explained variance
   r2 = r2_score(score_y, predictions)
    edsw.log_metric("r2", r2)
    print('Variance score: %.2f' % r2)
36
   # Output
   filename = 'model.pkl'
    pickle.dump(model, open(filename, 'mb'))
   cdsw.log_file(filename)
```

Edit View Navigate Run

#### Start New Session

Before you can connect to your secure Hadoop cluster, you must enter your credentials under Settings > Hadoop Authentication.

Engine Image - Configure

Base Image v4 - docker.repository.cloudera.com/odsw/engine:4

Select Engine Kernel

Python 2

Python 3

Scale

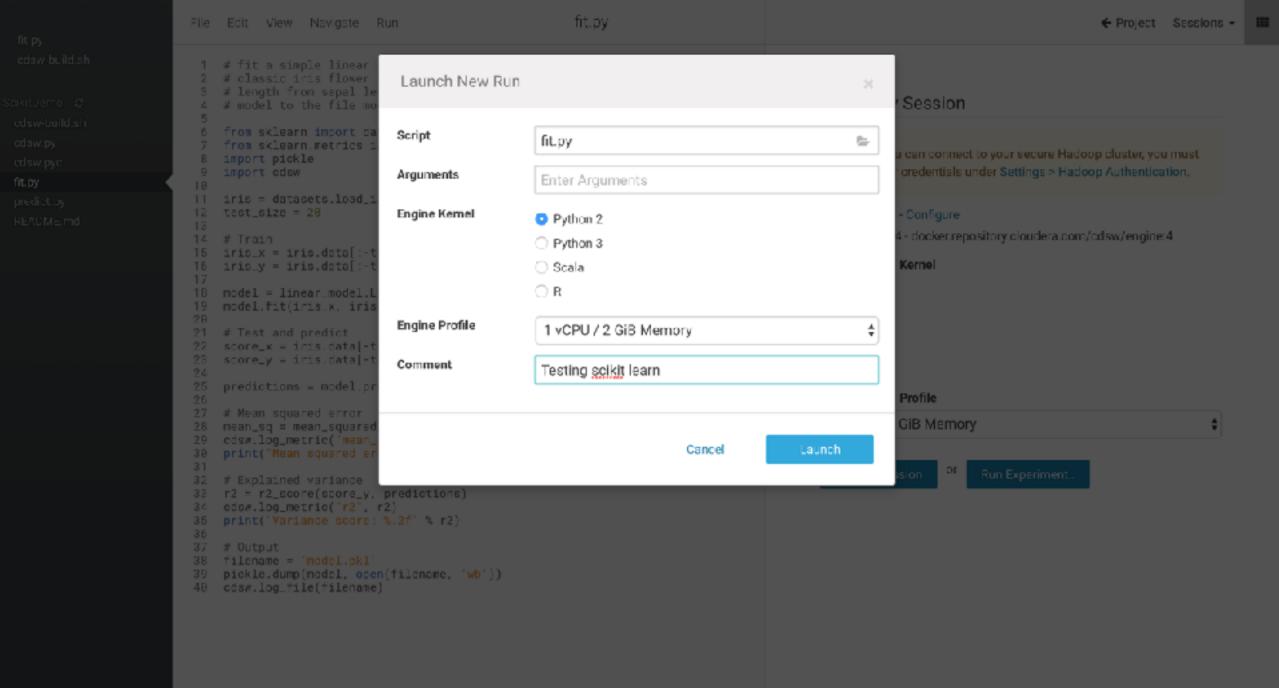
 $\bigcirc$  R

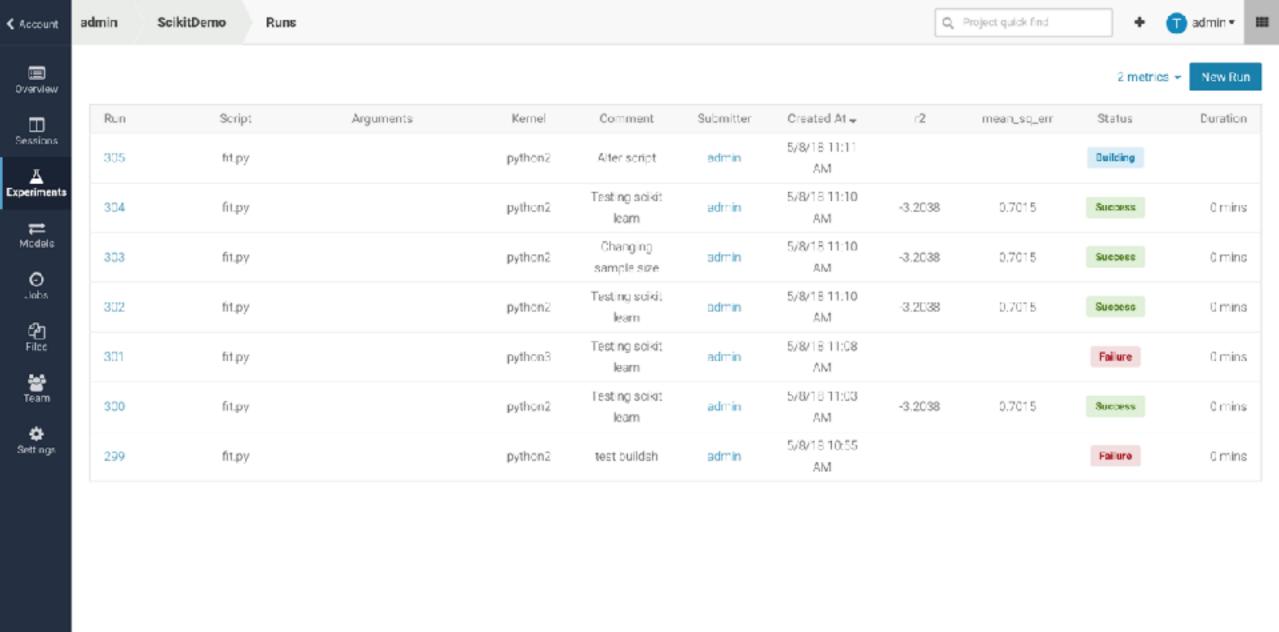
Select Engine Profile

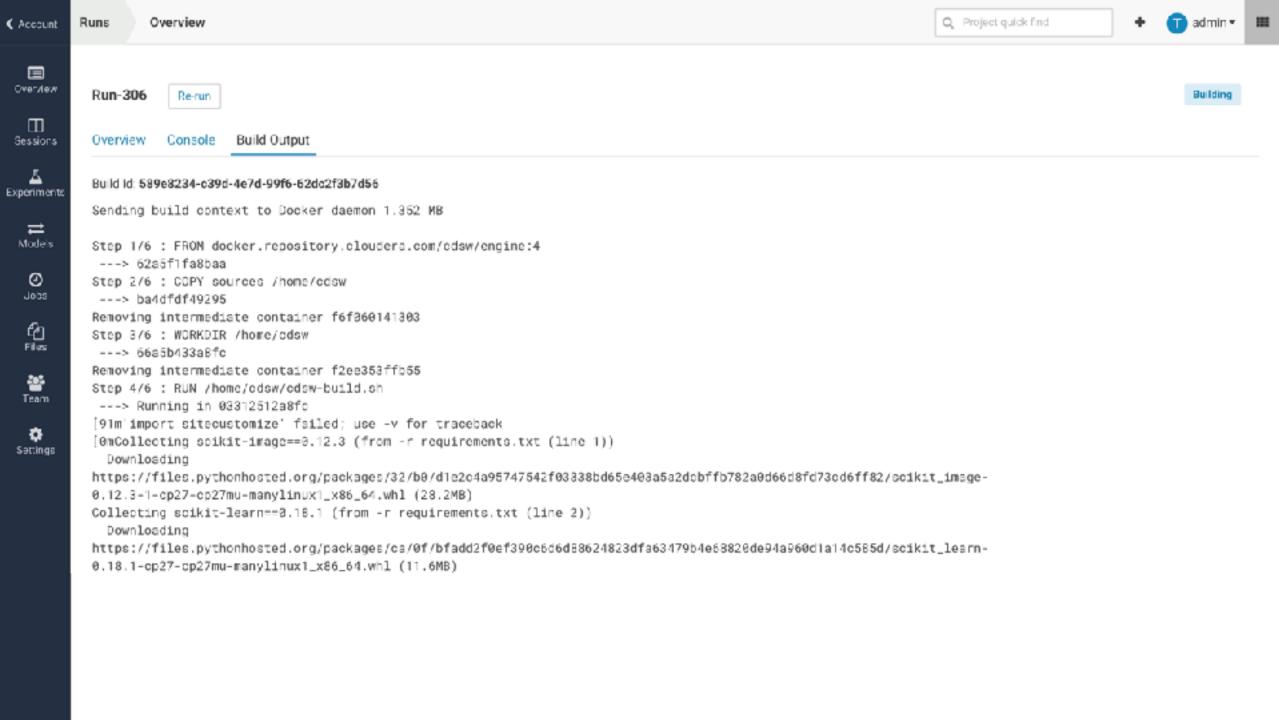
1 vCPU / 2 GiB Memory

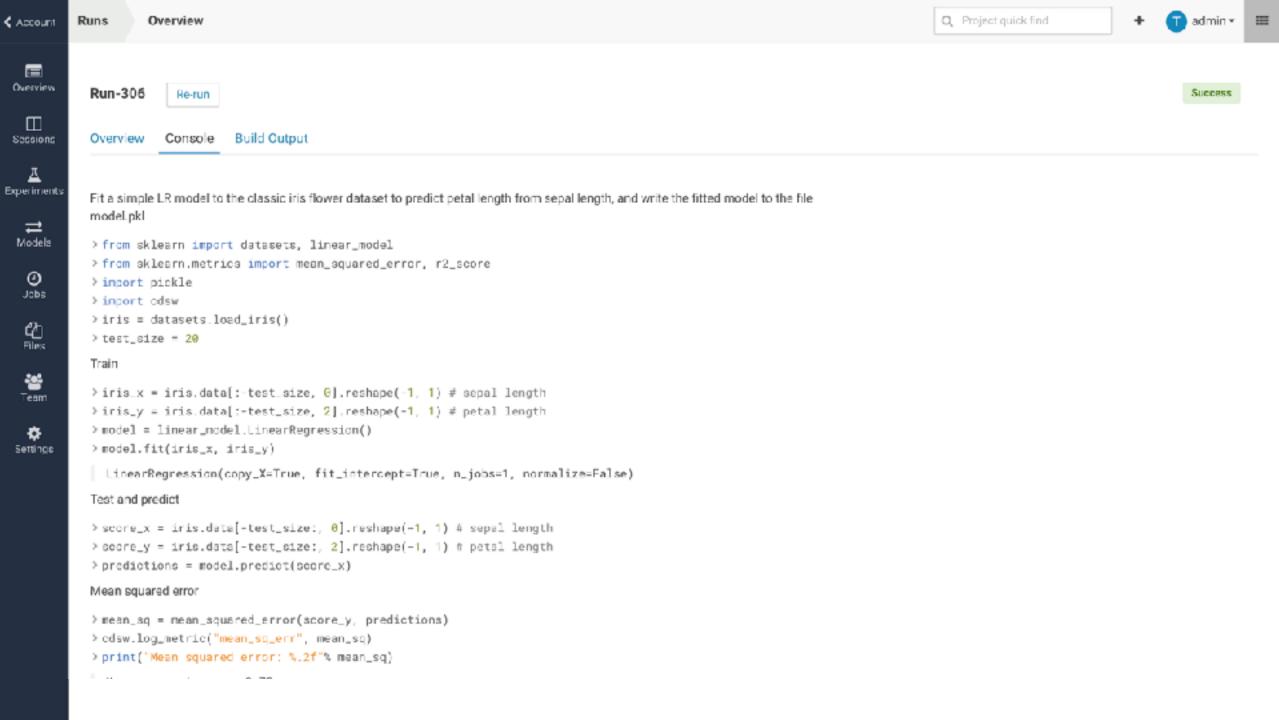
Launch Session

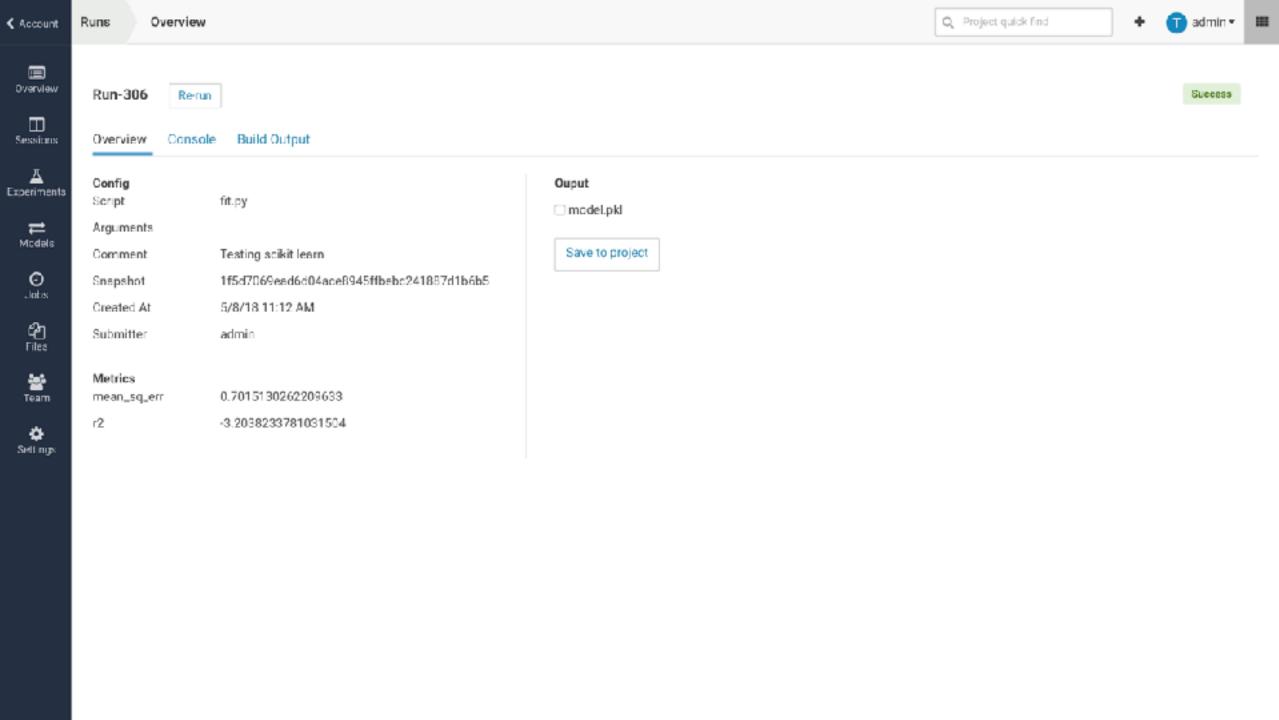
Run Experiment...

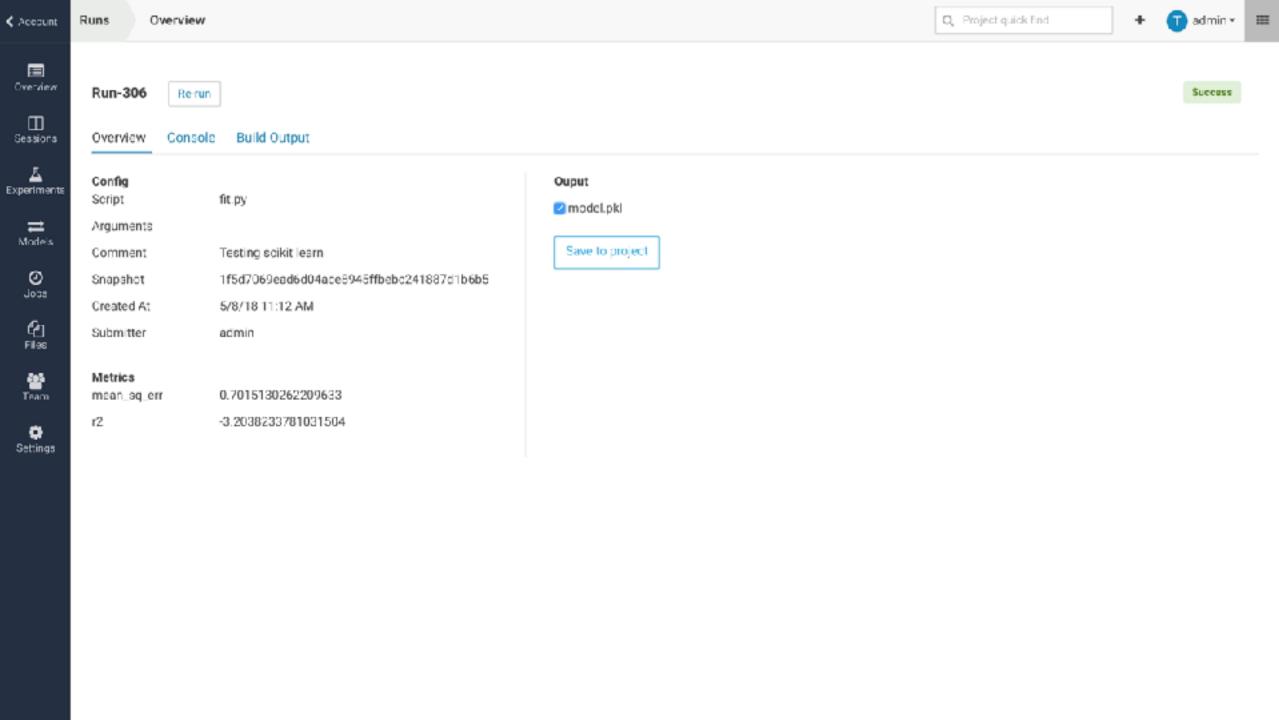


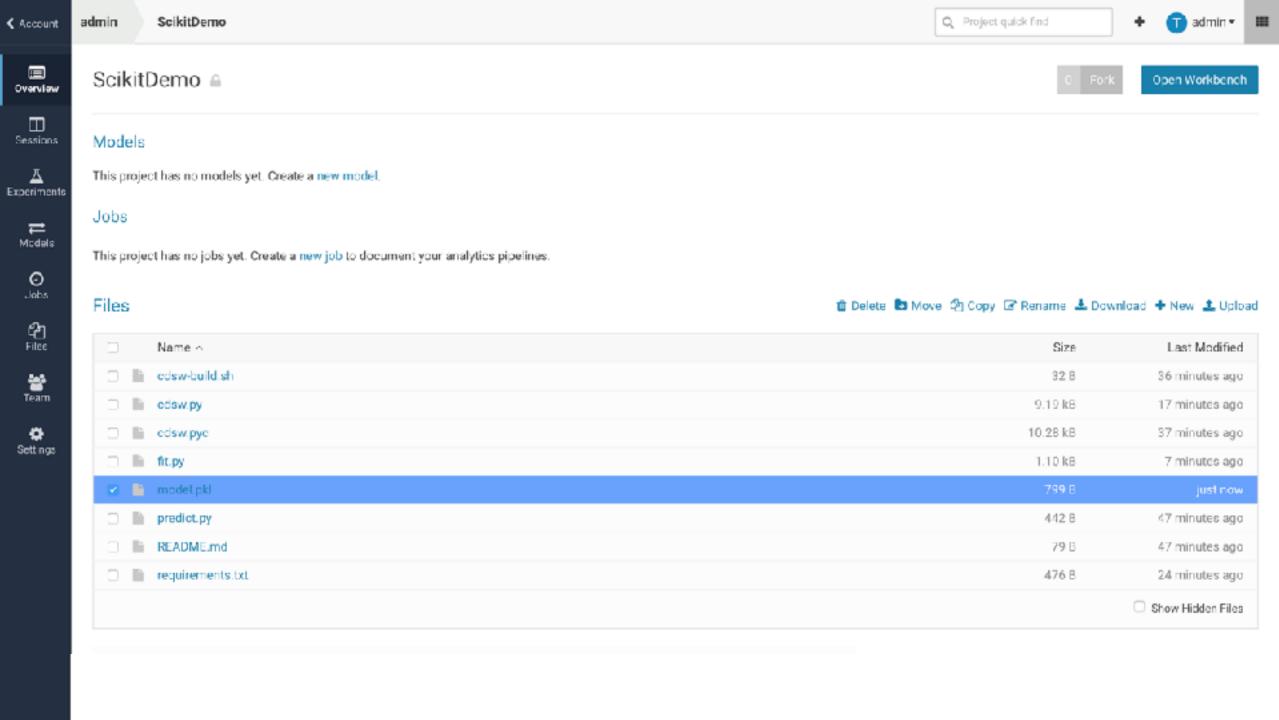




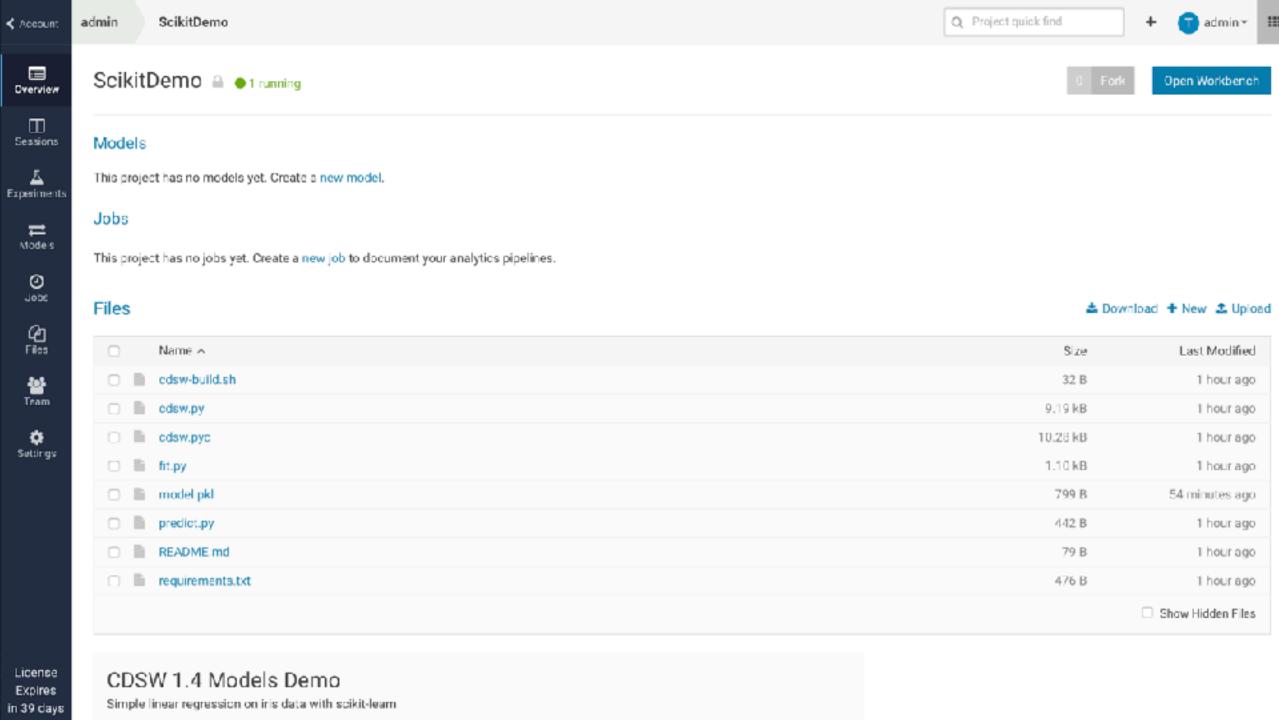


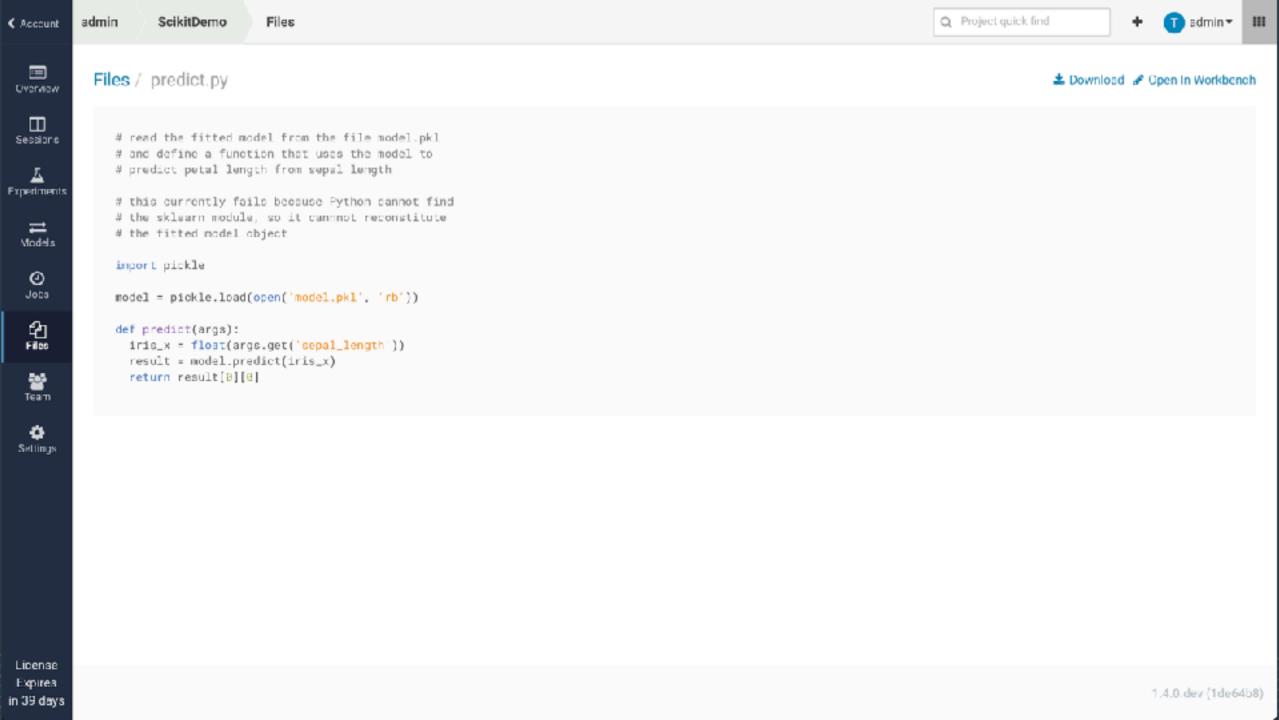


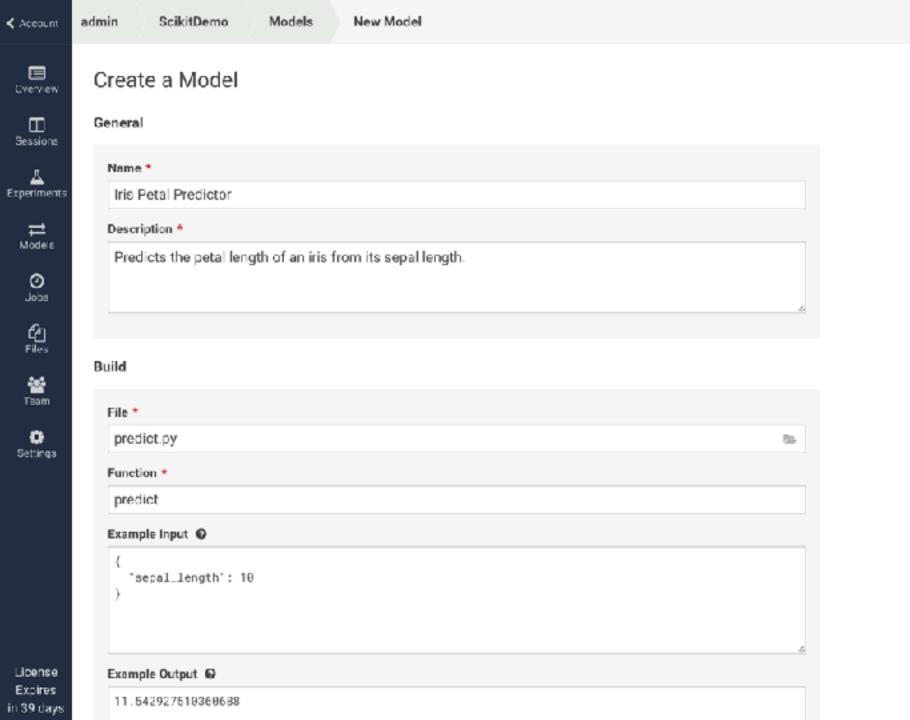




# DEPLOY AND MANAGE MODELS DEMO

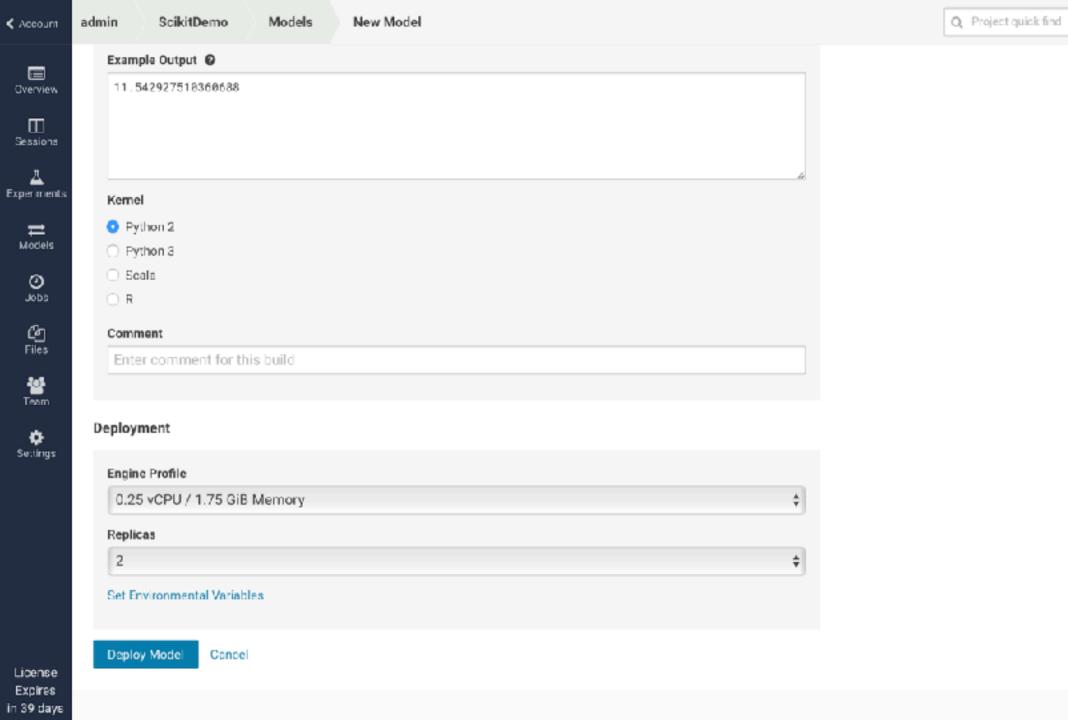






admin = III

Q Project quick find



admin 🕶 🔡

