

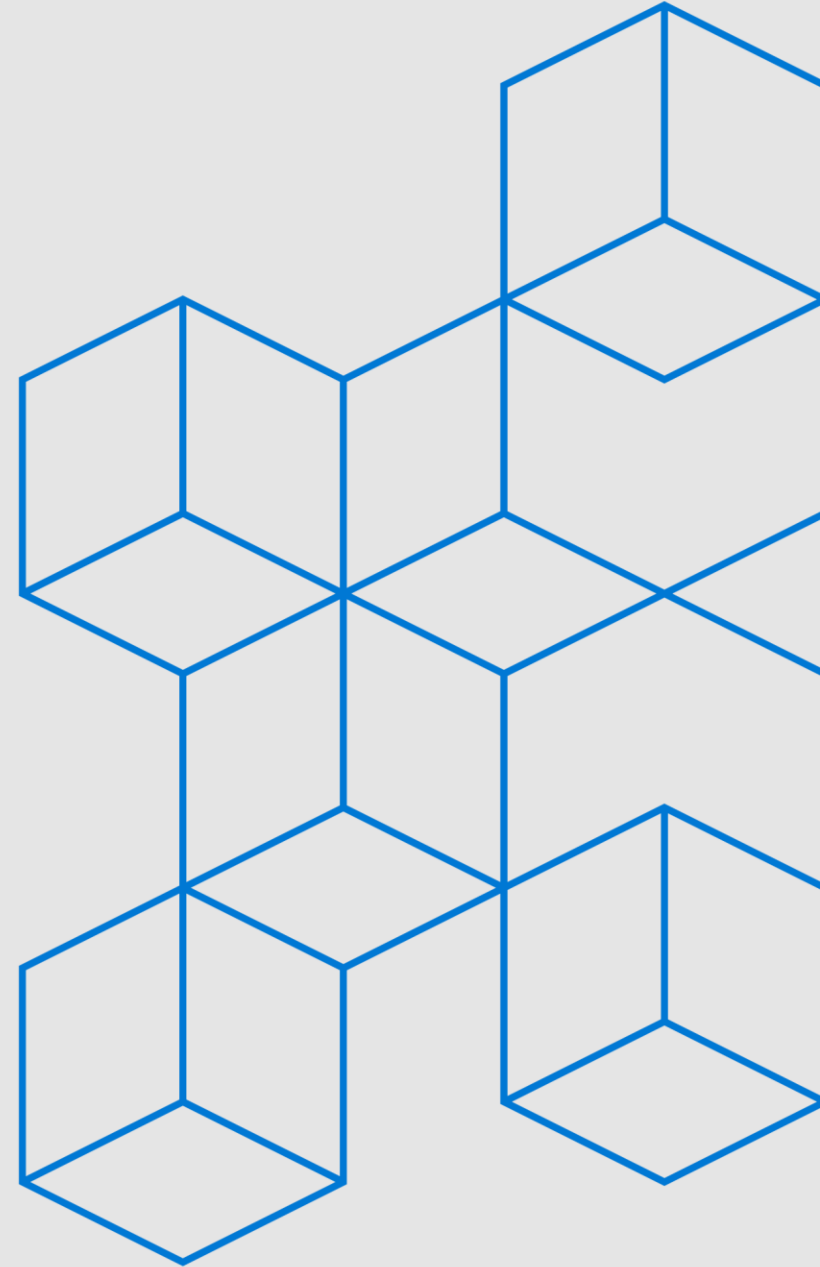
Azure Machine Learning 介绍

Kinfey.Lo

Microsoft MVP/Xamarin MVP

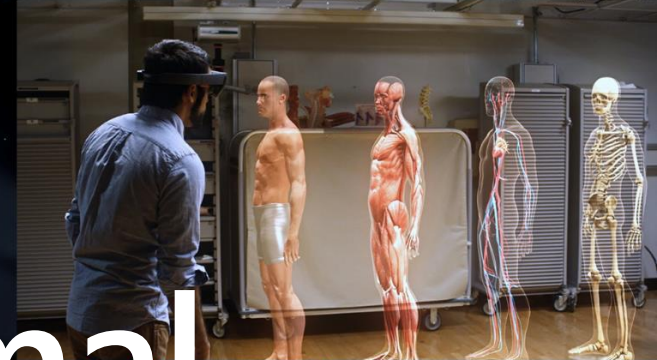
Github: <https://github.com/lokinfey>

Email: lokinfey@outlook.com



人工智能改变世界



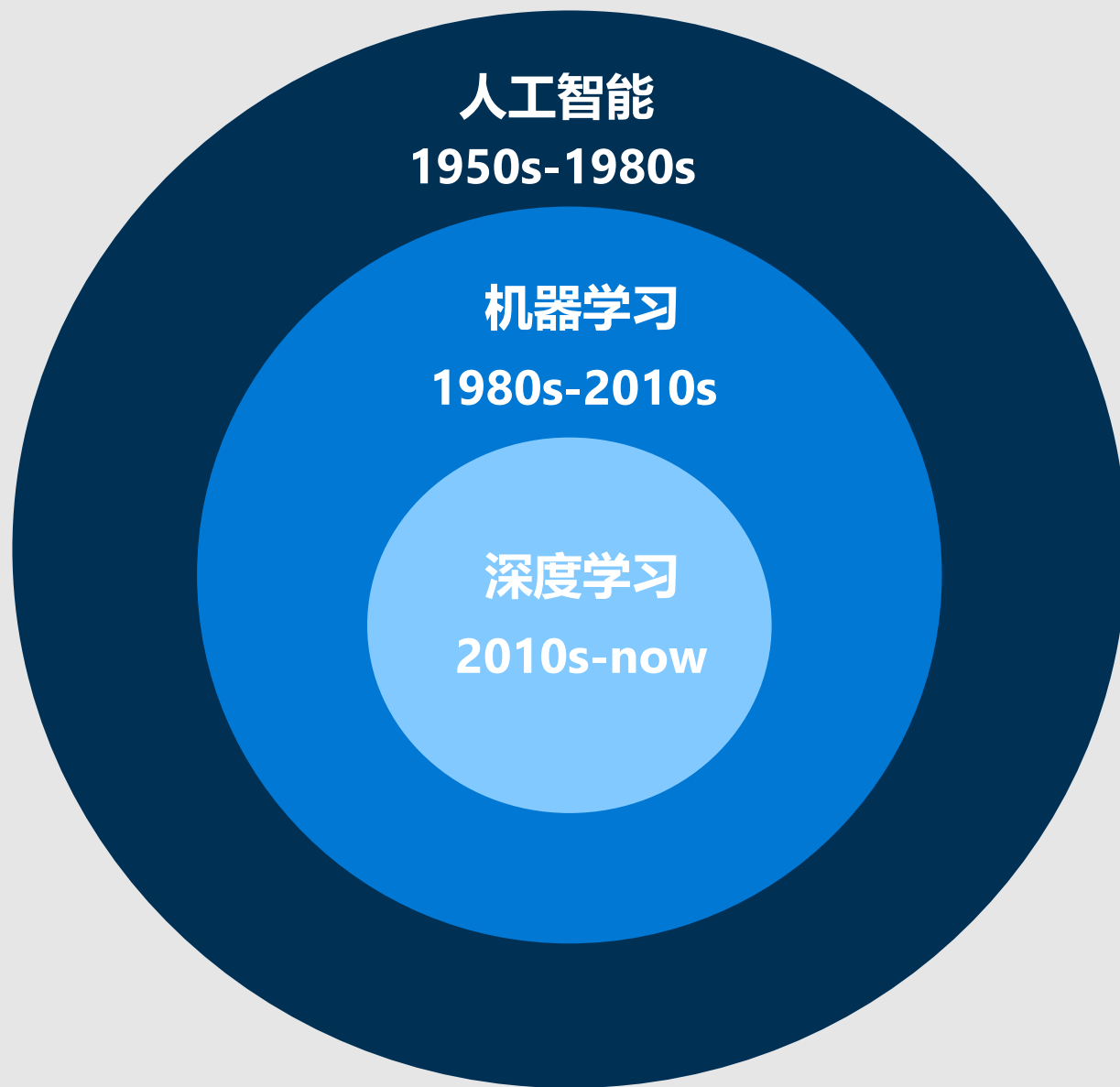


A low-angle, upward-looking photograph of a tree trunk. A measuring tape is stretched vertically along the trunk, starting from the bottom and extending towards the top. The tape has alternating white and dark segments. The tree's bark is dark and textured. The background is filled with the green foliage of other trees, creating a dense canopy. A semi-transparent white box is overlaid on the left side of the image, containing text.

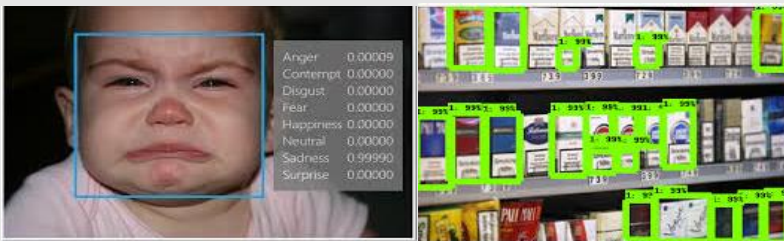
72 FT.

52.4932 13.281

AI 技术不断变化



人工智能的热点方向



计算机视觉



翻译



搜索

人工智能技术栈



语言



框架



硬件

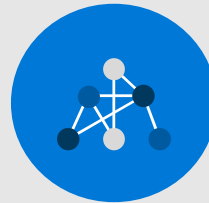
AI@Azure

AI 应用和服务



Azure Bot Service
Azure Cognitive Services

机器学习



Azure Databricks
Azure Machine Learning

知识挖掘



Azure Cognitive Search

Machine Learning on Azure

提供一些预定义的模型

能降低开发门槛



Vision



Speech



Language



Search

兼容不同的开发工具

快速完成模型开发，简化开发流程



PyCharm



Jupyter



Visual Studio Code



Command line

对人工智能开发框架的支持

根据你的需要创建深度学习的解决方案



Pytorch



TensorFlow



Scikit-Learn



Onnx

提供生产服务

为开发团队提供数据，和训练环境



Azure
Databricks



Azure Machine
Learning



Machine
Learning VMs

强大的硬件架构支持

加速深度学习环境



CPU



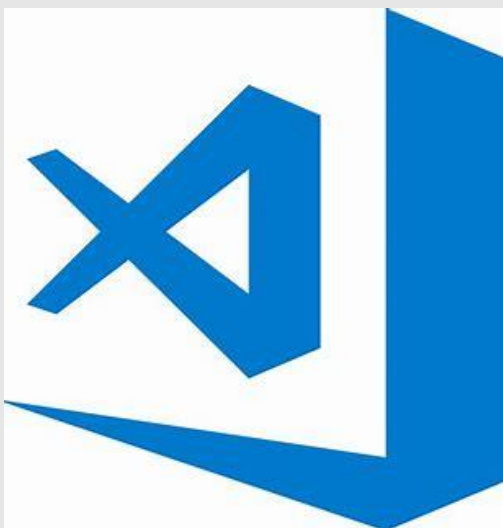
GPU




FPGA



从开发工具谈起



Visual Studio Code



Azure Machine Learning

Microsoft | 271,469 | ★★★★★ | Repository | License

Visual Studio Code extension for Azure Machine Learning

Disable ▾ Uninstall

[Details](#) [Contributions](#) [Changelog](#) [Dependencies](#)

Azure Machine Learning for Visual Studio Code

Azure Machine Learning for Visual Studio Code, previously called Visual Studio Code Tools for AI**, is an extension to easily build, train, and deploy machine learning models to the cloud or the edge with [Azure Machine Learning service](#).

Getting Started

- [Installation](#)
- [Getting started with Azure Machine Learning for Visual Studio Code](#)
- [Create and manage Azure compute targets](#)
- [Train and tune models](#)
- [Deploy and manage models](#)
- [Release notes](#)

With Azure Machine Learning service, you can:


- Build and train machine learning models faster, and easily deploy to the cloud or the edge.
- Use the latest open source technologies such as [TensorFlow](#), [PyTorch](#), or [Jupyter](#).
- Experiment locally and then quickly scale up or out with large GPU-enabled clusters in the cloud.
- Speed up data science with automated machine learning and hyper-parameter tuning.
- Track your experiments, manage models, and easily deploy with integrated CI/CD tooling.

** Previous documentation and vsix installer are moved to the archive folder.

Supported Operating Systems

Currently this extension supports the following 64-bit operating systems:

- Windows
- macOS



示例

Visual Studio Code interface showing a Python script named `demo1.py` being executed in a Jupyter Notebook environment. The script uses `matplotlib` to display a grid of images from a directory named `gear_images`.

The Explorer pane on the left shows the project structure:

- OPEN EDITORS
 - GROUP 1
 - demo1.py
 - GROUP 2
 - Python Interactive
- AZURE DEVDAY
 - .vscode
 - gear_images
 - resized_images

The Python Interactive pane on the right displays the Jupyter Server URI and the output of the script, which is a grid of images:

Jupyter Server URI: `http://localhost:8888/?token=c58f0b05e51e451519d42ca9248a41ea99745b385a32d3b1`
Python version: 3.6.5 [Anaconda, Inc.] (default, Mar 29 2018, 13:32:41) [MSC v.1900 64 bit (AMD64)]
C:\Program Files (x86)\Microsoft Visual Studio\Shared\Anaconda3_64\python.exe
Jupyter Notebook Version: (5, 5, 0)

The script `demo1.py` contains the following code:

```
1 Run Cell | Run Below
2 #%%
3 import os
4 import shutil
5 import numpy as np
6 import matplotlib.pyplot as plt
7 from PIL import Image
8
9 Run Cell | Run Above | Run Below
10 #%%
11 get_ipython().run_line_magic('matplotlib', 'inline')
12
13 Run Cell | Run Above | Run Below
14 #%%
15 imgdir = 'gear_images'
16 fig = plt.figure(figsize=(12, 16))
17 dir_num = 0
18 for root, folders, filenames in os.walk(imgdir):
19     for folder in folders:
20         # Load the first image file using the PIL library
21         file = os.listdir(os.path.join(root, folder))[0]
22         imgFile = os.path.join(root, folder, file)
23         img = Image.open(imgFile)
24         # Add the image to the figure (which will have 4 col
25         a=fig.add_subplot(4,np.ceil(len(folders)/4),dir_num
26         imgplot = plt.imshow(img)
27         # Add a caption with the folder name
28         a.set_title(folder)
29         dir_num = dir_num + 1
30
31 Run Cell | Run Above | Run Below
32 #%%
33 import os
34 import shutil
35 import numpy as np
36 import matplotlib.pyplot as plt
37 from PIL import Image
38
39 # Helper function to resize image
40 def resize_image(img, size):
41     from PIL import Image, ImageOps
42     # resize the image so the longest dimension matches our
43     img.thumbnail(size, Image.ANTIALIAS)
44     # Create a new square white background image
```

The output shows a grid of images from the `gear_images` directory, including items like a climbing rope, a climbing shoe, a carabiner, a climbing helmet, and a climbing jacket.

NEW

Azure Cognitive Services

1. Object Detection & Recognition for thousands of objects
2. Video Indexer (Preview)
3. Speech Recognition with customization
4. Speech Synthesis with customizable voice
5. Speech to Speech Translation
6. Text analytics with entity detection
7. Language Understanding (LUIS) with new integrated offer
8. QnA Maker is Generally Available
9. Bing Visual Search with smart identification
10. Bing Search SDK is Generally Available



Vision



Speech



Language



Conversation



Bing Search



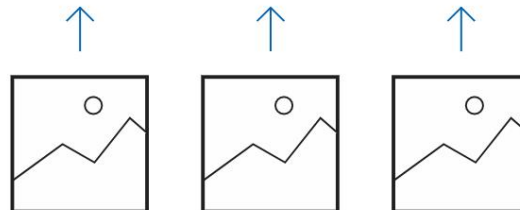
Knowledge

Demo

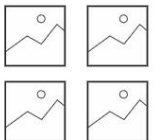
Custom Vision

Visual Intelligence Made Easy

Easily customize your own state-of-the-art computer vision models that fit perfectly with your unique use case. Just bring a few examples of labeled images and let Custom Vision do the hard work.

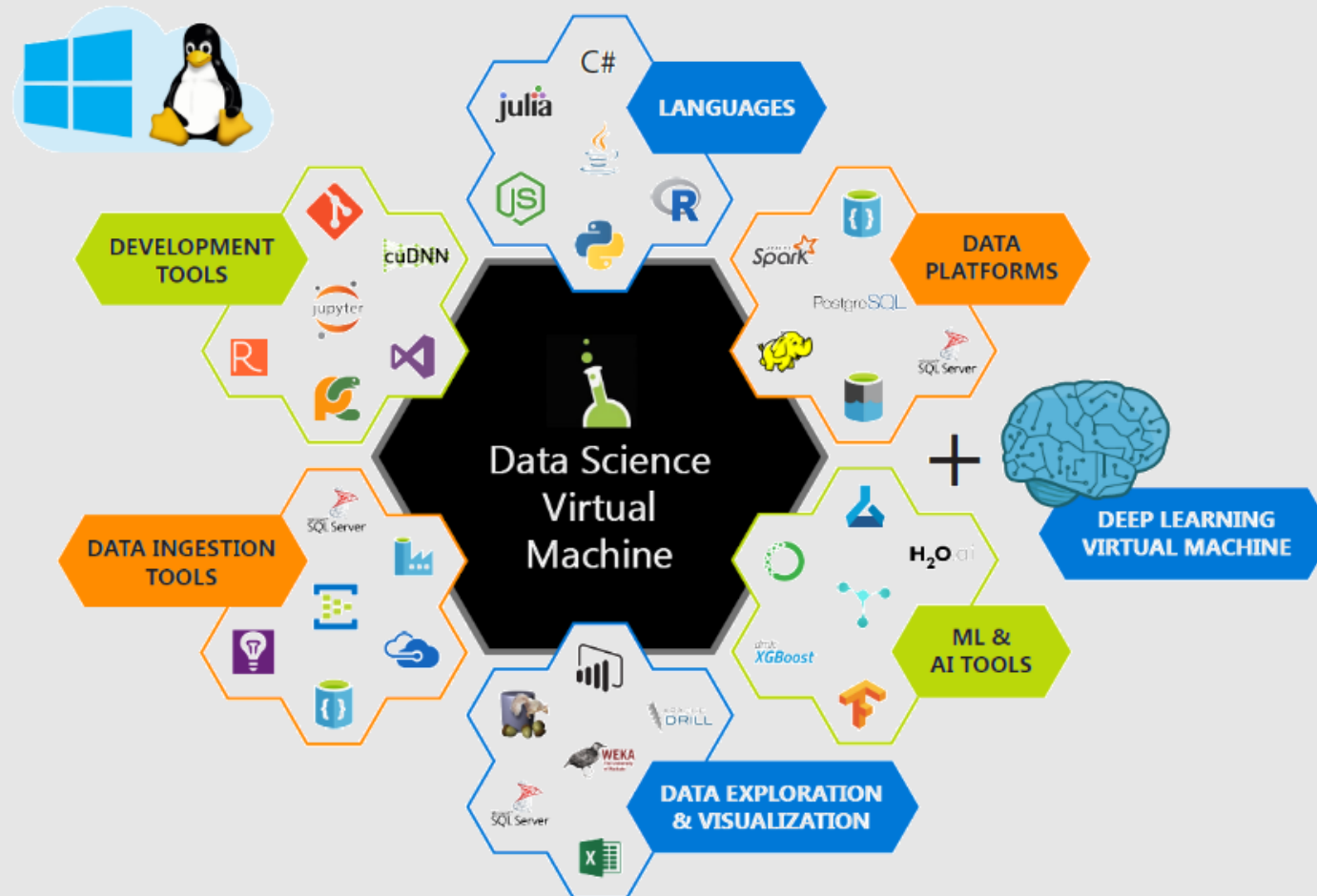
[SIGN IN](#)

89%



Data Science Virtual Machine(DSVM)

**Pre-Configured environments in the cloud for
Data Science & AI Modeling, Development & Deployment.**



Data Science Virtual Machine(DSVM)



DSVM – Windows Server 2016

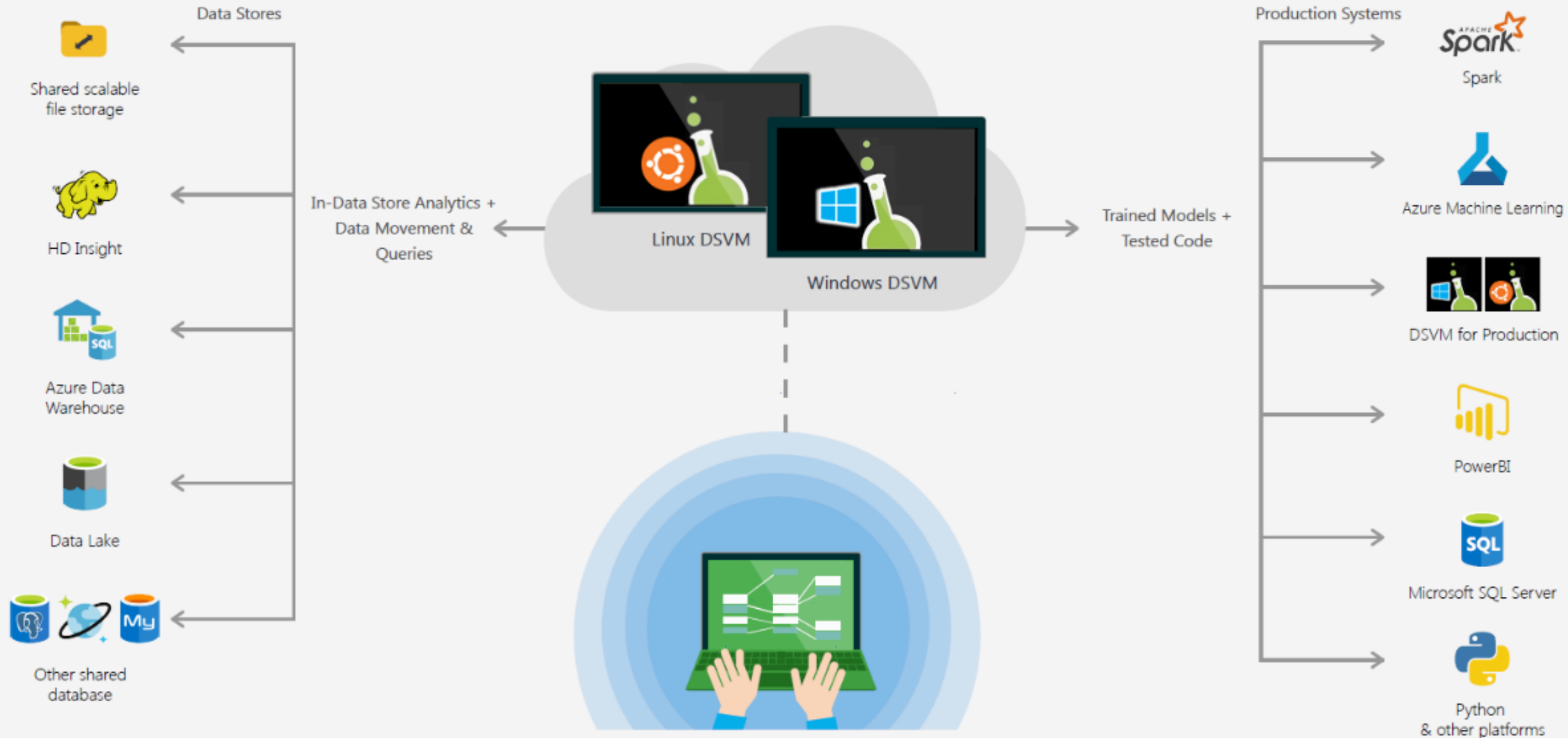


DSVM – Linux – Ubuntu



Deep Learning Virtual Machines

End-to-End AI Development Workflow using Data Science Virtual Machines (DSVM)



Demo

Jupyterhub on DSVM

jupyter lab03 Last Checkpoint: 5 hours ago (autosaved)

File Edit View Insert Cell Kernel Widgets Help

Save Add New Open Recent Undo Redo Run Stop Restart Code Edit Presentation Show Presentation

```
→ return model
```

```
In [5]: import matplotlib
matplotlib.use("Agg")

from keras.preprocessing.image import ImageDataGenerator
from keras.optimizers import Adam
from keras.preprocessing.image import img_to_array
from keras.utils import np_utils
from sklearn.preprocessing import LabelBinarizer
from sklearn.model_selection import train_test_split
# from imagesearch.smallervggnet import SmallerVGGNet
import matplotlib.pyplot as plt
from imutils import paths
import numpy as np
import argparse
import random
import pickle
import cv2
import os
```

```
In [6]: EPOCHS = 100
INIT_LR = 1e-3
BS = 32
IMAGE_DIMS = (96, 96, 3)

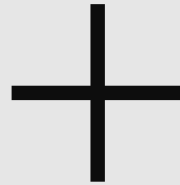
data = []
labels = []

print("[INFO] loading images...")
imagePaths = sorted(list(paths.list_images('resized_images'))))
random.seed(42)
random.shuffle(imagePaths)

for imagePath in imagePaths:
    → image = cv2.imread(imagePath)
    → image = cv2.resize(image, (IMAGE_DIMS[1], IMAGE_DIMS[0]))
    → image = img_to_array(image)
    → data.append(image)
```

Azure Machine Learning service

Azure Cloud
Services



Python
SDK

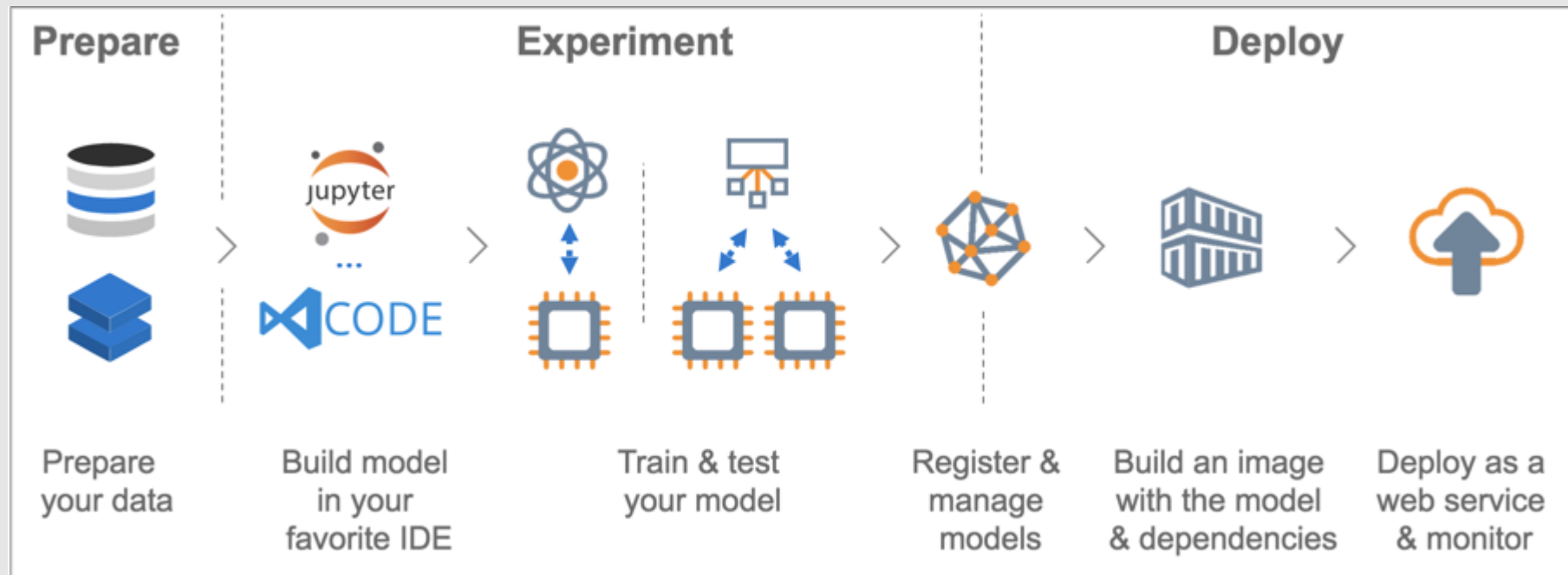
帮助你完成:

- ✓ 数据准备
- ✓ 编译模型
- ✓ 训练模型

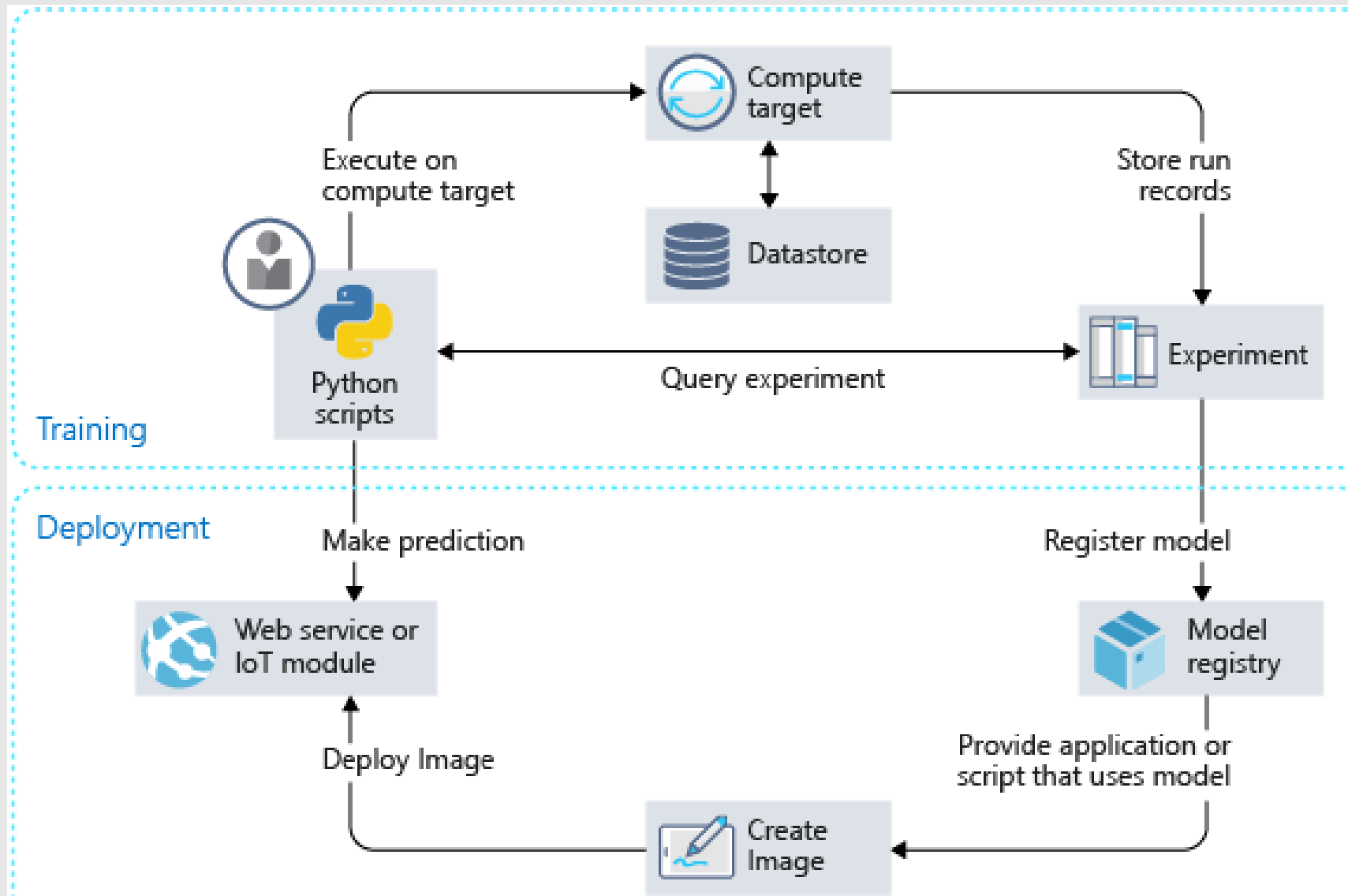
- ✓ 管理模型
- ✓ 跟踪训练
- ✓ 部署模型

Azure Machine Learning Service

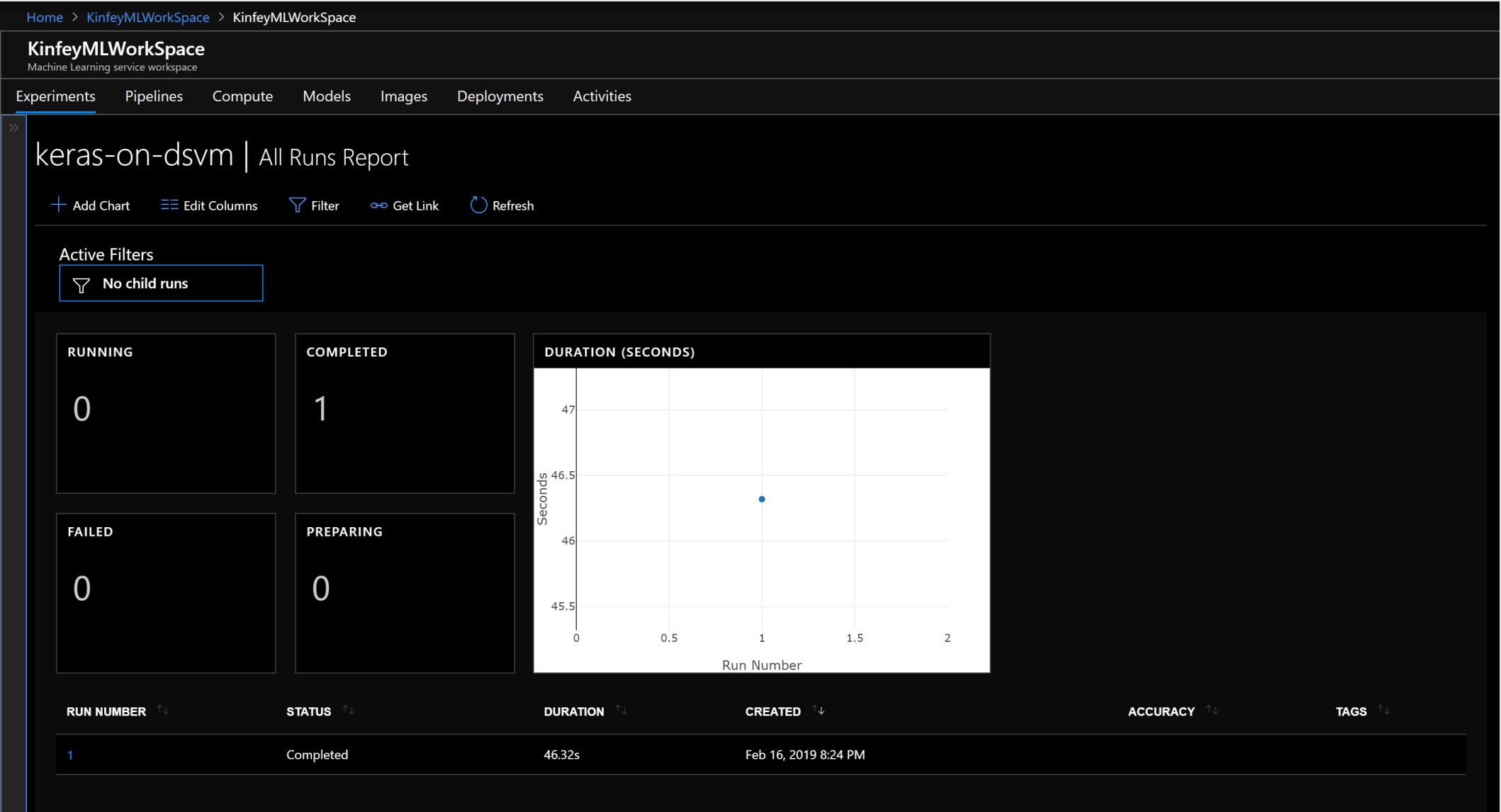
Azure Machine Learning service provides a cloud-based environment you can use to prep data, train, test, deploy, manage, and track machine learning models.




Azure Machine Learning Service



Azure Machine Learning Workbench



Azure Machine Learning Tools for VS Code



Azure Machine Learning

ms-toolsai/vscode-ai Preview

Microsoft | 241,098 | ★★★★★ | Repository | License

Visual Studio Code extension for Azure Machine Learning

Disable Uninstall

Details Contributions Changelog Dependencies

Azure Machine Learning for Visual Studio Code

Azure Machine Learning for Visual Studio Code, previously called Visual Studio Code Tools for AI**, is an extension to easily build, train, and deploy machine learning models to the cloud or the edge with [Azure Machine Learning service](#).

With Azure Machine Learning service, you can:

- Build and train machine learning models faster, and easily deploy to the cloud or the edge.
- Use the latest open source technologies such as [TensorFlow](#), [PyTorch](#), or [Jupyter](#).
- Experiment locally and then quickly scale up or out with large GPU-enabled clusters in the cloud.
- Speed up data science with automated machine learning and hyper-parameter tuning.
- Track your experiments, manage models, and easily deploy with integrated CI/CD tooling.

** Previous documentation and vsix installer are moved to the archive folder.

Supported Operating Systems

Currently this extension supports the following 64-bit operating systems:

- Windows
- macOS
- Linux Ubuntu

Getting Started

- [Release notes](#)
- [Installation](#)
- [Getting started with Azure Machine Learning](#)
- [Create and manage Azure compute targets](#)
- [Train and tune models](#)
- [Deploy and manage models](#)

AZURE: MACHINE LEARNING

- Windows Azure MSDN - Visual Stu...
- KinfeyMLWorkspace
 - Experiments
 - keras-on-dsvm
 - Pipelines
 - Compute
 - Models
 - Images
 - Deployments

Untitled-1 • Extension: Azure Machine Learning

```
1 Run Cell | Run All Cells
2 ###
3 from keras import backend
4 print(backend._BACKEND)
5
6 Run Cell | Run All Cells
7 ###
8 from keras.models import Sequential
9 from keras.layers import Dense
10 import numpy as np
11
12 np.random.seed(7)
13
14 dataset = np.loadtxt('pima-indians-diabetes.csv',delimiter=',')
15
16 x = dataset[:,0:8]
17 Y = dataset[:,8]
18
19 model = Sequential()
20 model.add(Dense(12, input_dim=8, activation='relu'))
21 model.add(Dense(8, activation='relu'))
22 model.add(Dense(1, activation='sigmoid'))
23
24 model.compile(loss='binary_crossentropy',optimizer='adam',metrics=['accuracy'])
25
26 model.fit(x=x,y=Y,epochs=150,batch_size=10)
27
28 scores=model.evaluate(x=x,y=Y)
29 print('\n%s : %.2f%%' % (model.metrics_names[1],scores[1]*100))
30
31 Run Cell | Run All Cells
32 ###
33
34
35
36
```


1

2

3

4

5

6

7

8

9

10

11

12

13

14

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

Run Cell | Run All Cells

###

from keras import backend

print(backend._BACKEND)

Run Cell | Run All Cells

###

from keras.models import Sequential

from keras.layers import Dense

import numpy as np

np.random.seed(7)

dataset = np.loadtxt('pima-indians-diabetes.csv',delimiter=',')

x = dataset[:,0:8]

Y = dataset[:,8]

model = Sequential()

model.add(Dense(12, input_dim=8, activation='relu'))

model.add(Dense(8, activation='relu'))

model.add(Dense(1, activation='sigmoid'))

model.compile(loss='binary_crossentropy',optimizer='adam',metrics=['accuracy'])

model.fit(x=x,y=Y,epochs=150,batch_size=10)

scores=model.evaluate(x=x,y=Y)

print('\n%s : %.2f%%' % (model.metrics_names[1],scores[1]*100))

Run Cell | Run All Cells

###

AZURE: MACHINE LEARNING

+

🔄

Windows Azure MSDN - Visual Studio

KinfeyMLWorkSpace

Experiments

keras-on-dsvm

Pipelines

Compute

Models

Images

Deployments

Extension: Azure Machine Learning

Python Interactive

×

Jupyter Server URI: http://localhost:8888/?token=f19ba4f243b52212b37ef26a13e374145de001acfaeb13b3

Python Version:

3.6.6 (v3.6.6:4cflf54eb7, Jun 27 2018, 03:37:03) [MSC v.1900 64 bit (AMD64)]

C:\\Program Files (x86)\\Microsoft Visual Studio\\Shared\\Python36_64\\python.exe

Jupyter Notebook Version: (5, 7, 4, '')

[1] from keras import backend...

Using TensorFlow backend.

tensorflow

[2] from keras.models import Sequential...

768/768 [=====] - 0s 251us/step - loss: 0.6174 - acc: 0.6992

Epoch 15/150

768/768 [=====] - 0s 242us/step - loss: 0.6015 - acc: 0.6979

Epoch 16/150

768/768 [=====] - 0s 260us/step - loss: 0.5877 - acc: 0.7018

Epoch 17/150

768/768 [=====] - 0s 264us/step - loss: 0.5850 - acc: 0.6979

Epoch 18/150

768/768 [=====] - 0s 311us/step - loss: 0.5999 - acc: 0.6901

Epoch 19/150

768/768 [=====] - 0s 259us/step - loss: 0.5806 - acc: 0.7122

Epoch 20/150

768/768 [=====] - 0s 245us/step - loss: 0.5797 - acc: 0.7240

Epoch 21/150

768/768 [=====] - 0s 260us/step - loss: 0.5712 - acc: 0.7148

Epoch 22/150

768/768 [=====] - 0s 238us/step - loss: 0.5831 - acc: 0.6992

Epoch 23/150

768/768 [=====] - 0s 276us/step - loss: 0.5751 - acc: 0.7122

Epoch 24/150

768/768 [=====] - 0s 299us/step - loss: 0.5704 - acc: 0.7266

Epoch 25/150

768/768 [=====] - 0s 255us/step - loss: 0.5577 - acc: 0.7383

Epoch 26/150

768/768 [=====] - 0s 239us/step - loss: 0.5720 - acc: 0.7044

Epoch 27/150

768/768 [=====] - 0s 210us/step - loss: 0.5551 - acc: 0.7383

>>> Shift-enter to run

Python 3.6.6 64-bit

0 0 0

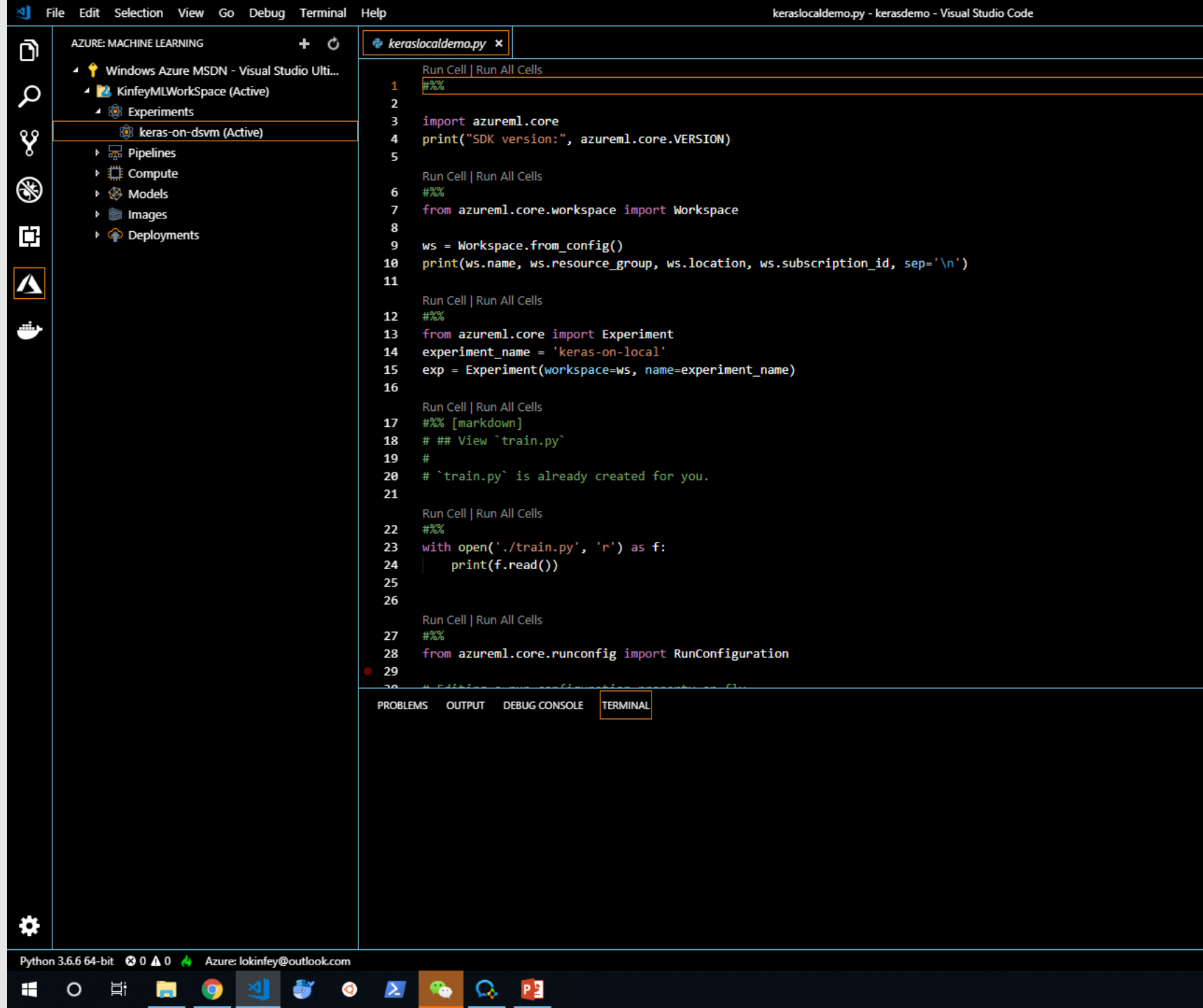
Azure: lokinfey@outlook.com

11:49 AM

2/18/2019

Demo

Azure ML Workbench Demo



Thank you