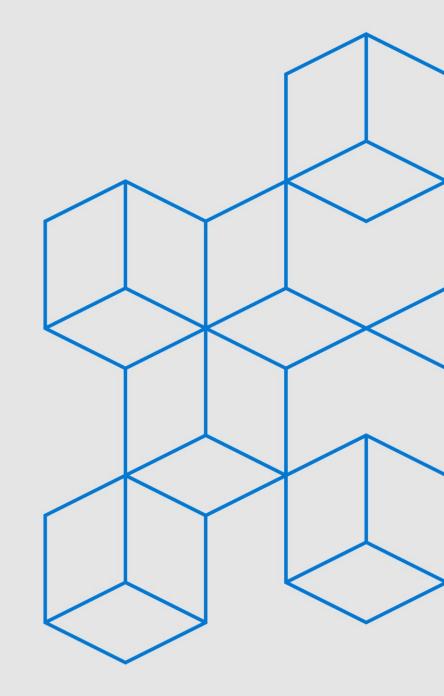# Azure Machine Learning 介绍

Kinfey.Lo
Microsoft MVP/Xamarin MVP
Github：https://github.com/lokinfey
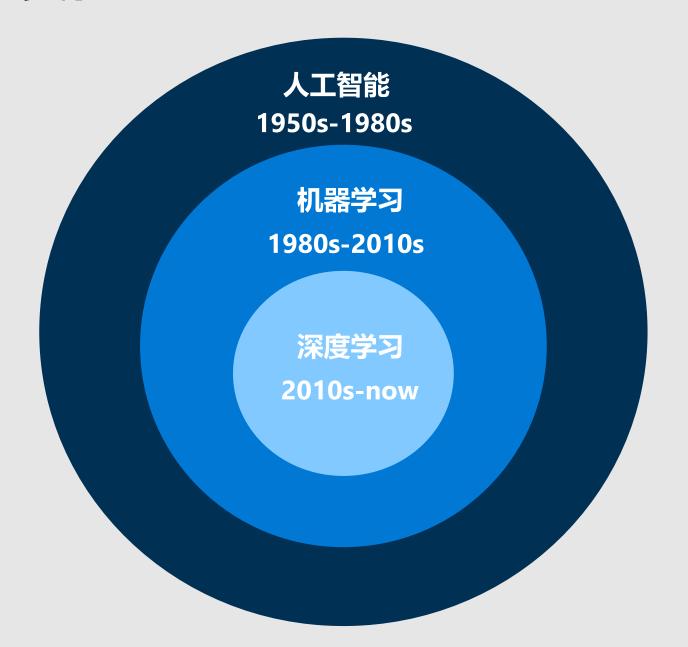**Email: lokinfey@outlook.com**

人工智能改变世界

AI is the New Normal

72 FT

52.4932   13.281

AI 技术不断变化

人工智能
1950s-1980s

机器学习
1980s-2010s

深度学习
2010s-now
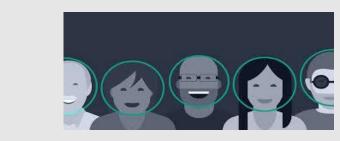
人工智能的热点方向
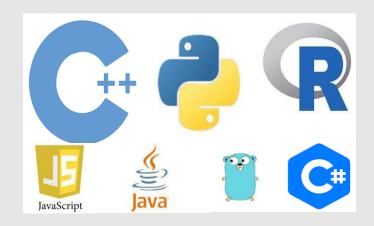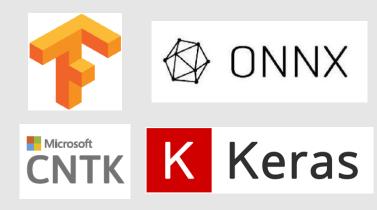
计算机视觉
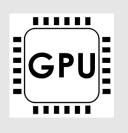
翻译

搜索

人工智能技术栈

语言

框架

硬件

AI@Azure

AI 应用和服务

机器学习

知识挖掘

Azure Bot Service
Azure Cognitive Services

Azure Databricks
Azure Machine Learning

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
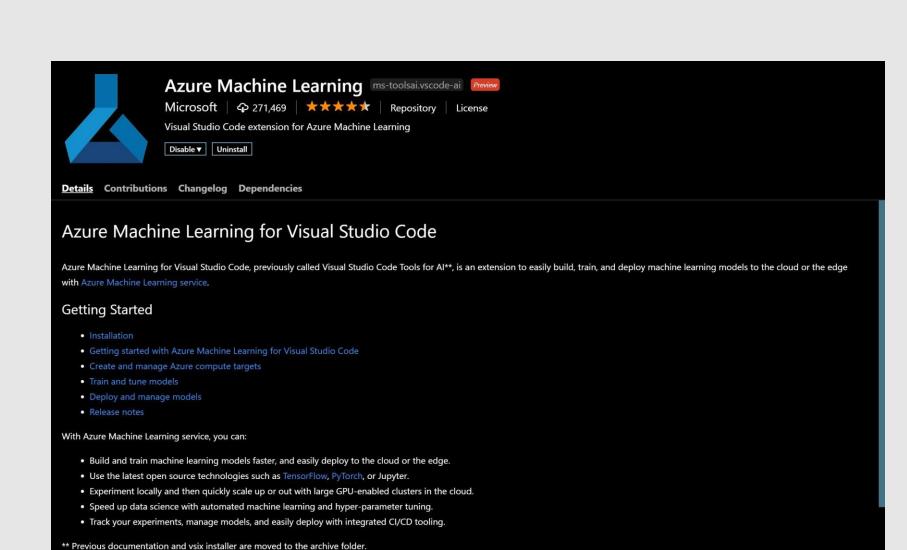
从开发工具谈起

Visual Studio Code
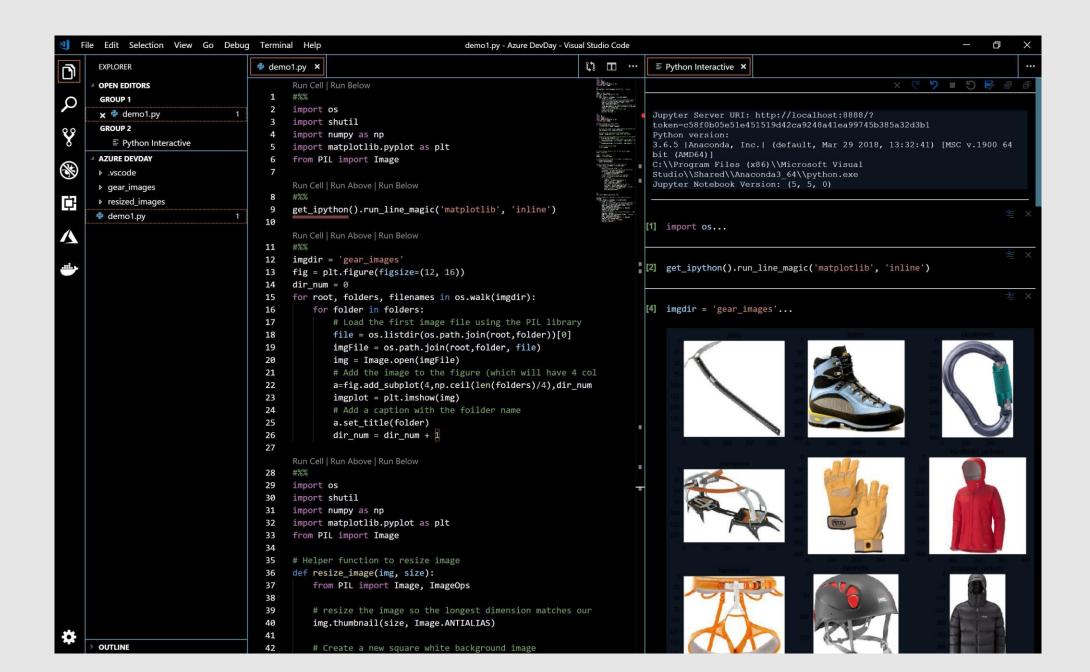
示例

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
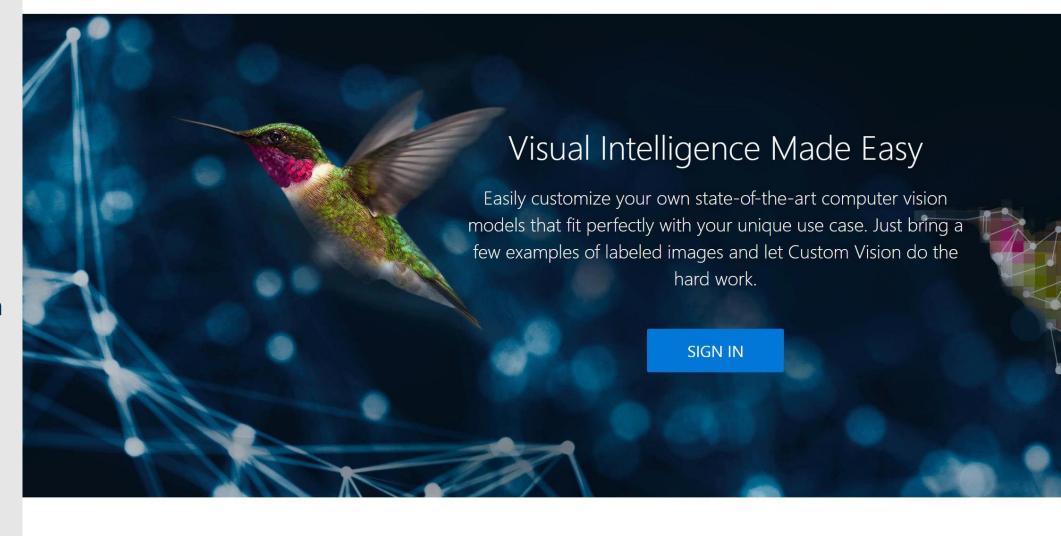
Bing Search

Knowledge
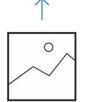
Data Science Virtual Machine(DSVM)
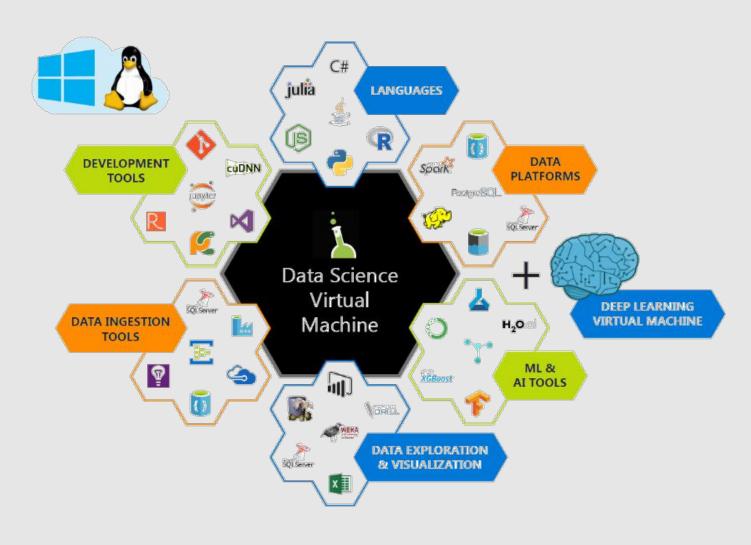
**Pre-Configured environments in the cloud for
Data Science & AI Modeling, Development & Deployment.**

Data Science Virtual Machine(DSVM)

DSVM – Windows Server 2016

DSVM – Linux – Ubuntu
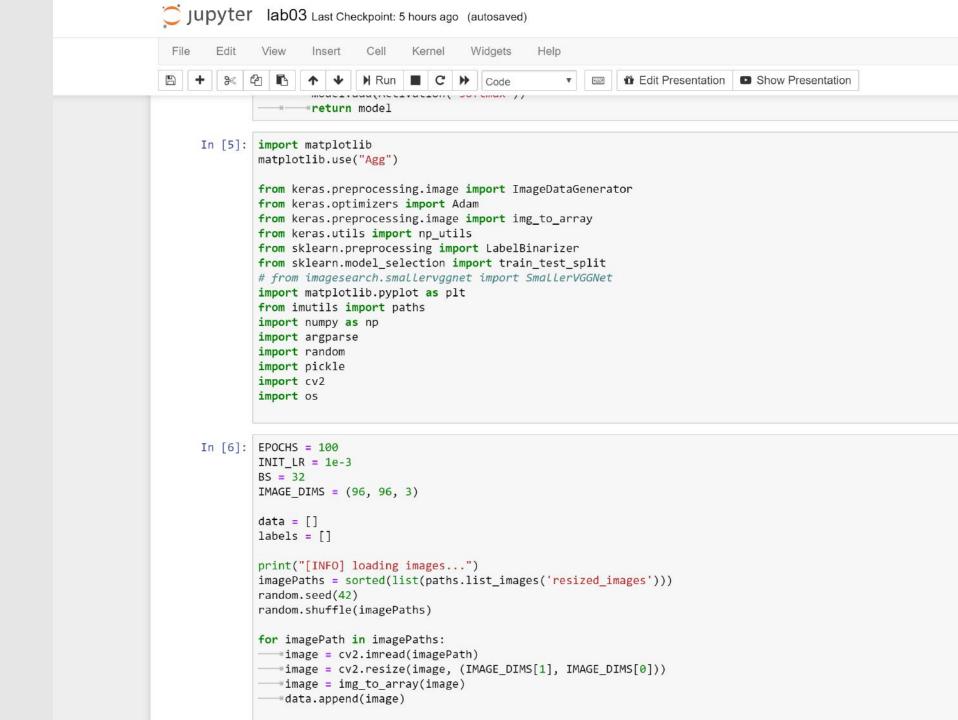
Deep Learning Virtual Machines

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

Demo

Jupyterhub on DSVM

File   Edit   View   Insert   Cell   Kernel   Widgets   Help

▶ Run   ■   C   ⏩   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
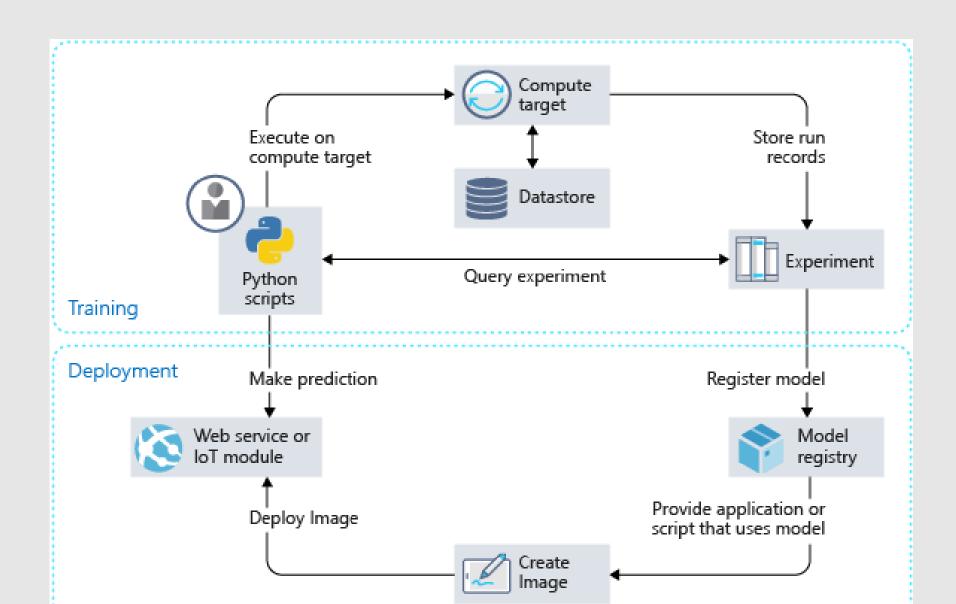
- ✓ 管理模型
- ✓ 跟踪训练
- ✓ 部署模型

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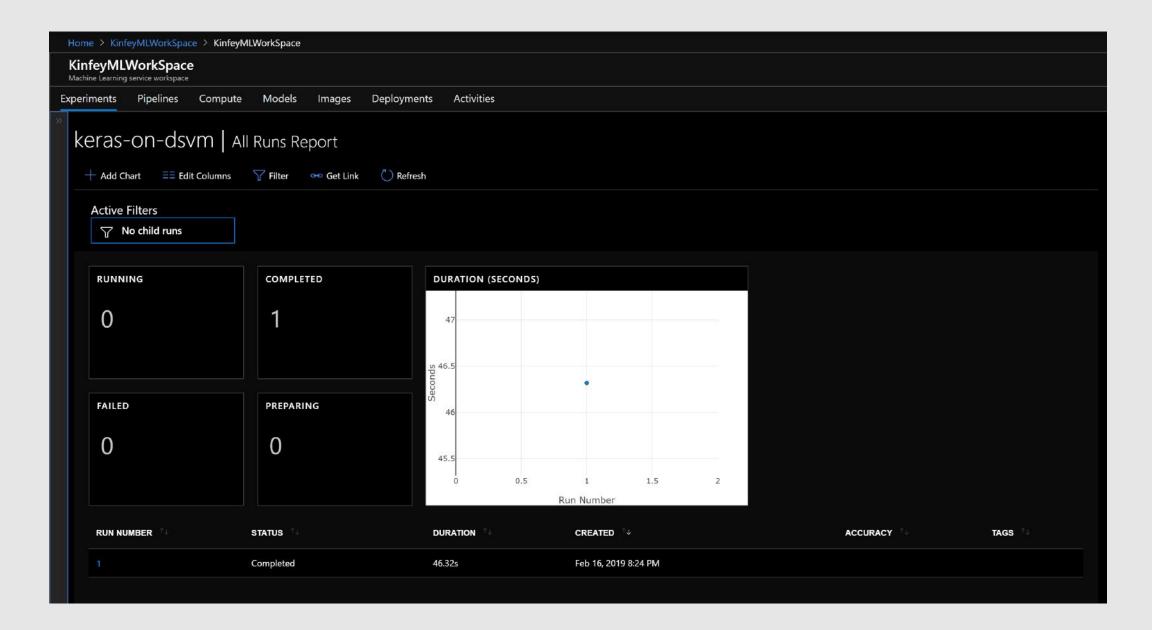
Azure Machine Learning service provides a cloud-based environment you can use to prep data, train, test, deploy, manage, and track machine learning models.

Azure Machine Learning Service
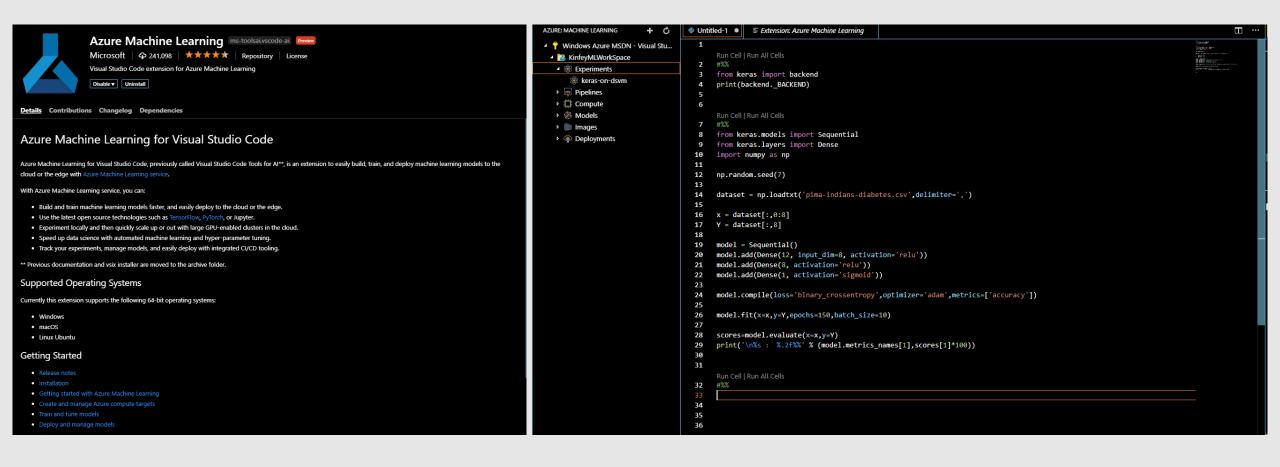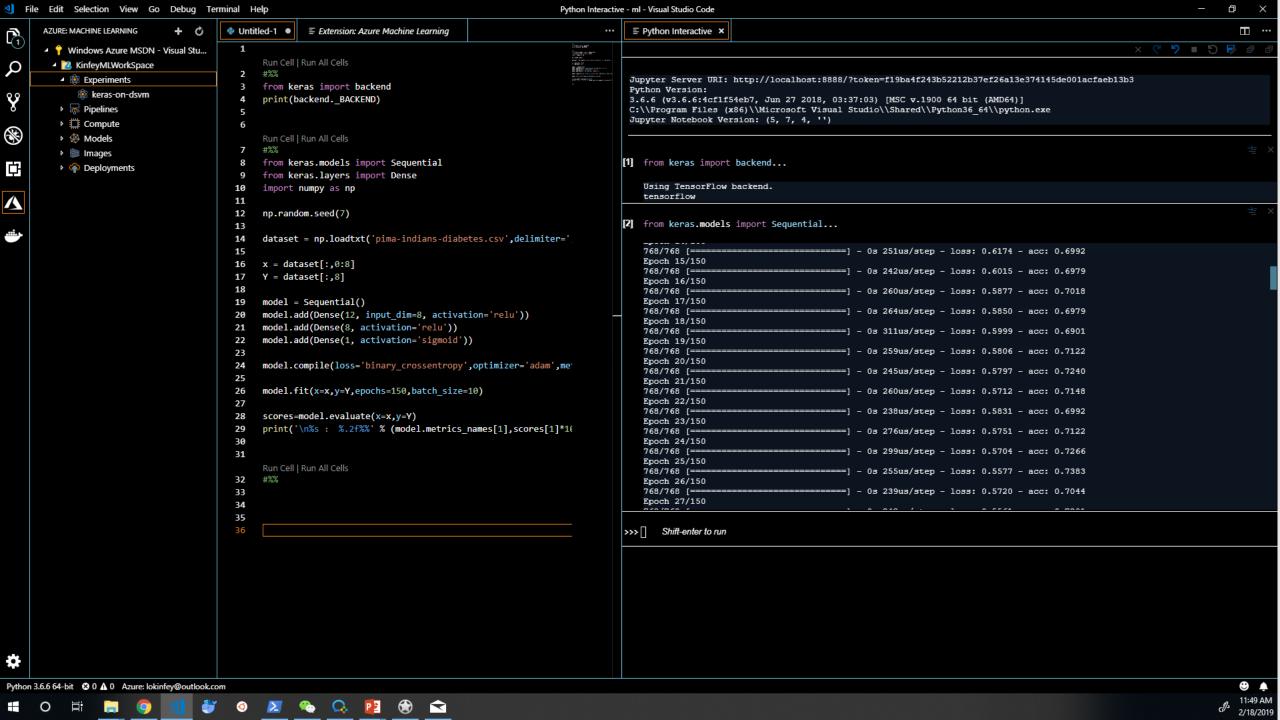
Azure Machine Learning Workbench

Azure Machine Learning Tools for VS Code

AZURE: MACHINE LEARNING

Untitled-1   ≡ Extension: Azure Machine Learning   ⋯   ≡ Python Interactive ✕   ⊞ ⋯

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

```python
 1
 2    #%%    Run Cell | Run All Cells
 3    from keras import backend
 4    print(backend._BACKEND)
 5
 6
 7    #%%    Run Cell | Run All Cells
 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',me
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]*10
30
31
32    #%%    Run Cell | Run All Cells
33
34
35
36
```
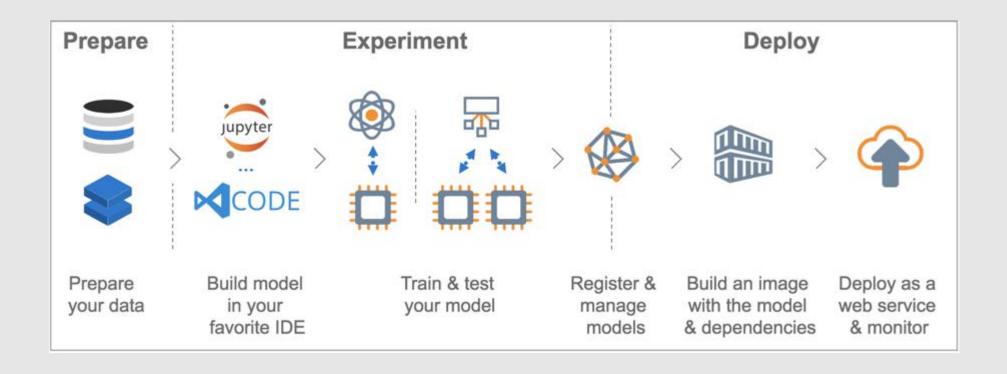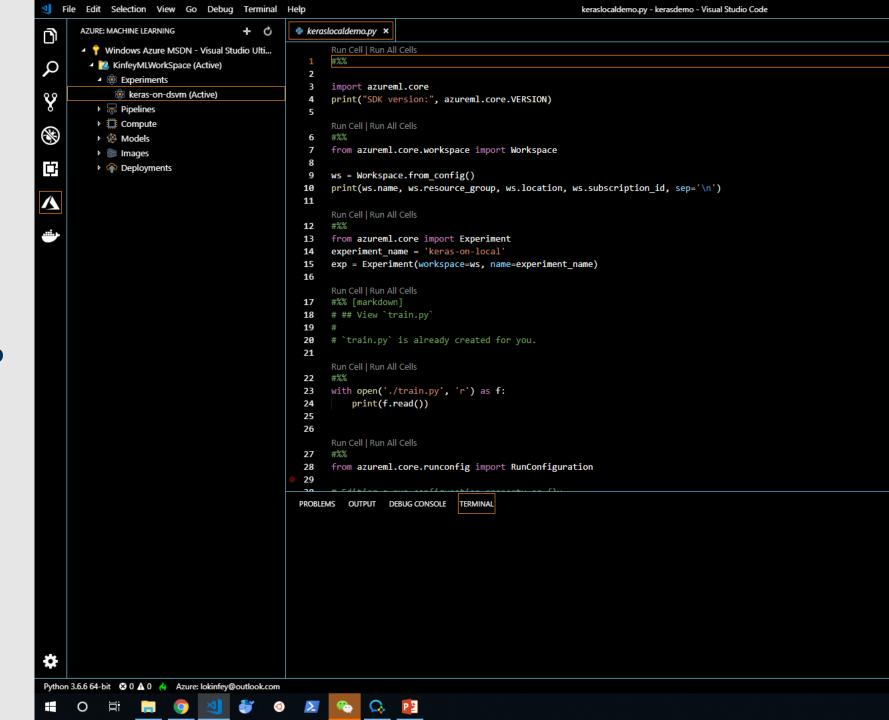
```
Jupyter Server URI: http://localhost:8888/?token=f19ba4f243b52212b37ef26a13e374145de001acfaeb13b3
Python Version:
3.6.6 (v3.6.6:4cf1f54eb7, Jun 27 2018, 03:37:03) [MSC v.1900 64 bit (AMD64)]
C:\\Program Files (x86)\\Microsoft Visual Studio\\Shared\\Python36_64\\python.exe
Jupyter Notebook Version: (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
```

```
>>>    Shift-enter to run
```

Demo

Azure ML Workbench Demo

Thank you