Custom AI with Azure Machine Learning services

Hands-on lab step-by-step

May 2018

Information in this document, including URL and other Internet Web site references, is subject to change without notice. Unless otherwise noted, the example companies, organizations, products, domain names, e-mail addresses, logos, people, places, and events depicted herein are fictitious, and no association with any real company, organization, product, domain name, e-mail address, logo, person, place or event is intended or should be inferred. Complying with all applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights, or other intellectual property rights covering subject matter in this document. Except as expressly provided in any written license agreement from Microsoft, the furnishing of this document does not give you any license to these patents, trademarks, copyrights, or other intellectual property.

The names of manufacturers, products, or URLs are provided for informational purposes only, and Microsoft makes no representations and warranties, either expressed, implied, or statutory, regarding these manufacturers or the use of the products with any Microsoft technologies. The inclusion of a manufacturer or product does not imply endorsement of Microsoft of the manufacturer or product. Links may be provided to third-party sites. Such sites are not under the control of Microsoft and Microsoft is not responsible for the contents of any linked site or any link contained in a linked site, or any changes or updates to such sites. Microsoft is not responsible for webcasting or any other form of transmission received from any linked site. Microsoft is providing these links to you only as a convenience, and the inclusion of any link does not imply endorsement of Microsoft of the site or the products contained therein.

© 2018 Microsoft Corporation. All rights reserved.

Microsoft and the trademarks listed at <https://www.microsoft.com/en-us/legal/intellectualproperty/Trademarks/Usage/General.aspx> are trademarks of the Microsoft group of companies. All other trademarks are the property of their respective owners.

Contents

[Custom AI with Azure Machine Learning services 1](#_Toc513549753)

[Abstract and learning objectives 1](#_Toc513549754)

[Overview 1](#_Toc513549755)

[Requirements 1](#_Toc513549756)

[Exercise 3: Create and Deploy an Unsupervised Model 2](#_Toc513549757)

[Task 1: Start the Data Science VM 2](#_Toc513549758)

[Task 2: Deploy your ACS cluster 2](#_Toc513549759)

[Task 3: Install dependencies 4](#_Toc513549760)

[Task 4: Set Visual Studio Code as the project IDE in Workbench 7](#_Toc513549761)

[Task 5: Create the Summarization service 8](#_Toc513549762)

[Task 6: Deploy the Summarization service 10](#_Toc513549763)

[Exercise 4: Train and Deploy a Deep Learning Model 13](#_Toc513549764)

[Task 1: Prepare TensorFlow 13](#_Toc513549765)

[Task 2: Train and deploy the TensorFlow model 13](#_Toc513549766)

[Task 3: Complete the solution 18](#_Toc513549767)

[After the hands-on lab 19](#_Toc513549768)

[Task 1: Clean up lab resources 19](#_Toc513549769)

## Custom AI with Azure Machine Learning services

## Abstract and learning objectives

In this workshop, you will learn to create intelligent solutions atop unstructured text data by designing and implementing a text analytics pipeline. You will also learn how to build a binary classifier using a simple neural network that can be used to classify the textual data. Also, you will learn how to deploy multiple kinds of predictive services using Azure Machine Learning.

Along the way, you will get to consider the following technologies and services:

* Azure Machine Learning services
* TensorFlow

## Overview

In this workshop, you will help Contoso Ltd. Build a proof of concept that shows how they could build a solution that amplifies the claims processing capabilities of their agents.

## Requirements

1. Microsoft Azure subscription must be pay-as-you-go or MSDN
   1. Trial subscriptions will not work

## Exercise 3: Create and Deploy an Unsupervised Model

Duration: 60 minutes

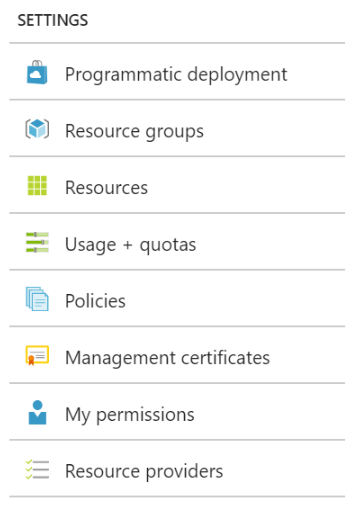
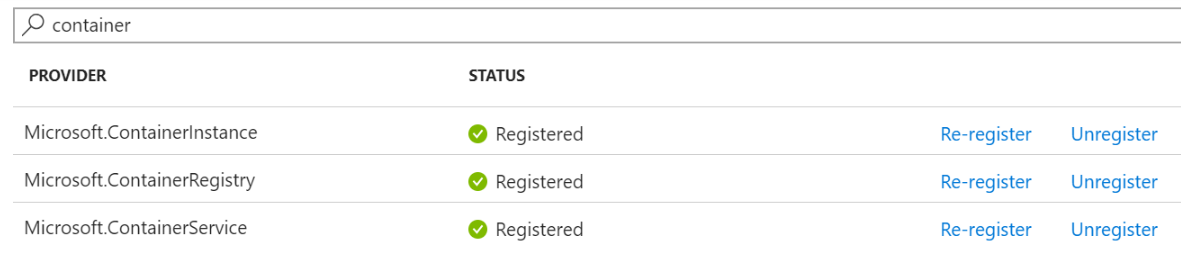
In this exercise you will create and deploy a web service that uses a pre-trained model to summarize long paragraphs of text.

#### Task 1: Start the Data Science VM

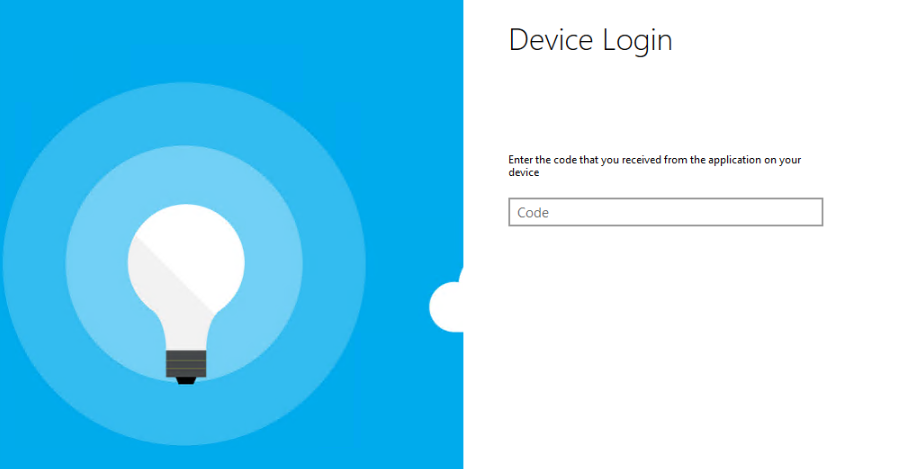
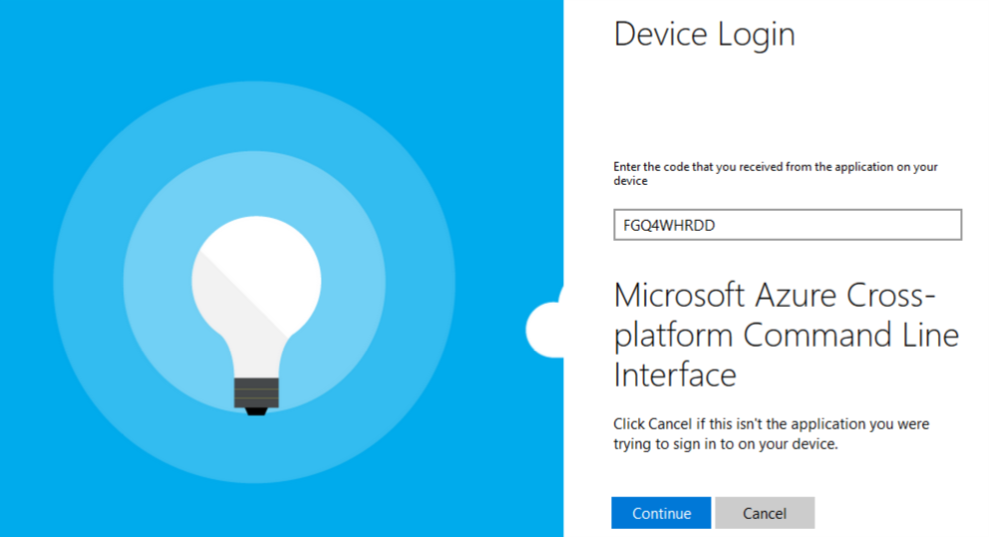
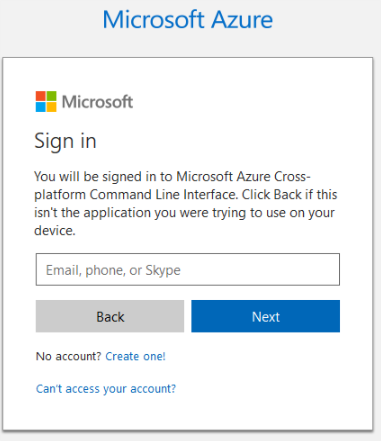
If you stopped the Data Science VM, you will want start it back up. Follow these steps to start the VM:

1. Return to the Azure Portal.
2. Navigate to the blade of your labvm.
3. Select the Start button to re-start the VM.
4. Connect to your VM using Remote Desktop.
5. Open Workbench and the TDSP project you created previously.

#### Task 2: Deploy your ACS cluster

1. Navigate to the Azure Portal.
2. Select All Services, Subscriptions and then select your subscription from the list.
3. Under the Settings grouping, select Resource Providers.   
   
4. Search for “container” and in the list that appears verify that all resource providers related to containers are registered. If not, select the Register link next to the items that are not registered.  
   
5. Return to the **Workbench**, from the **File** menu, select **Open Command Prompt.**
6. Create the cluster environment by running the following command, replacing the values indicated in angle brackets with appropriate values. This will create new resources groups for the cluster.
   1. For <environment name> enter mcwailabenv, or something similar. This value can only contain lowercase alphanumeric characters.
   2. For location, use eastus2, westcentralus, australiaeast, westeurope, or southeastasia, as those are the only acceptable values at this time.

az ml env setup -c -n <environment name> --location <e.g. eastus2>

1. If prompted, copy the URL presented and sign in using your web browser.
2. Enter the code provided in the command prompt.  
   
3. Select **Continue.**  
   
4. Sign in with your Azure Credentials.  
   
5. Return to the command prompt, which should automatically update after you log in.
6. At the “Subscription set to <subscription name>” prompt, enter **Y** if the subscription name is correct, or **N** to select the subscription to use from a list.   
   In the Command Prompt window, the updates  display.

   At this time, we are unable to capture all of the information in the command prompt window. Future versions of this course should address this.
7. It will take 10-20 minutes for your ACS cluster to be ready. You can periodically check on the status by running the command shown in the output to the previous step, which is of the form:  
     
   az ml env show -g <resourceGroupName> -n <clusterName>
8. **Continue on with the lab while your ACS cluster provisions**.

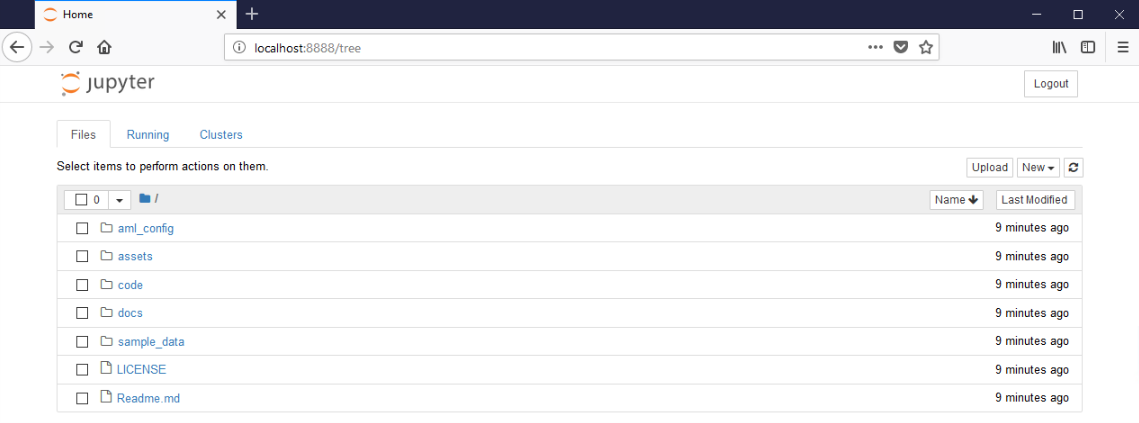
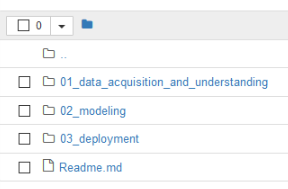
#### Task 3: Install dependencies

The tasks that follow depend on Python libraries like nltk and gensim. The following steps ensure you have these installed in your environment.

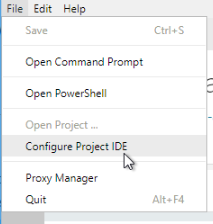
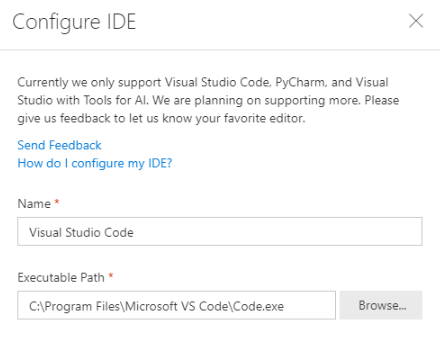
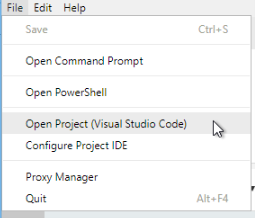
1. From the **File** menu of Workbench, select **Open Command Prompt**.
2. Run the following command to install nltk:  
   pip install nltk
3. NLTK should install, with a message similar to the following:  
   In the Command Prompt window, the previous commands and their output display.

   At this time, we are unable to capture all of the information in the command prompt window. Future versions of this course should address this.
4. NLTK is a rich toolkit with modular components, many of which are not installed by default. To install all the components, run the python shell by entering **python** at the command prompt:  
   python
5. Within the python shell, run the following two lines:  
   import nltk  
   nltk.download('all')
6. The downloader will take about 5 minutes to complete. Once it is finished, exit the python shell by entering:  
   exit()
7. Run the following command to install genism:  
   pip install gensim  
     
   In the Command Prompt window, the previous command and its output display.

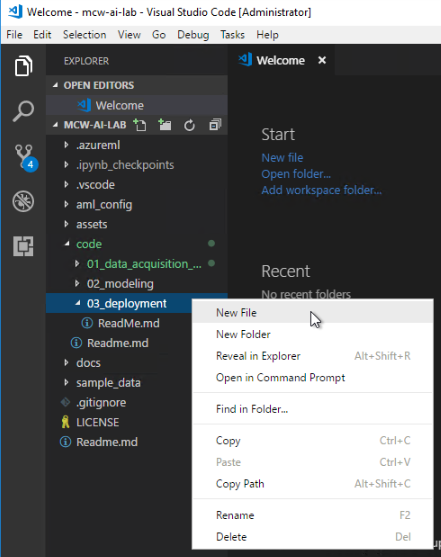
   At this time, we are unable to capture all of the information in the command prompt window. Future versions of this course should address this.
8. Next, download a pre-built Jupyter Notebook that you will step through to understand the process used to summarize the text of claims documents. In the **Firefox** browser on your VM, navigate to the following (note that the URL is case sensitive). Note, if using IE you will need to modify the default security settings, which prevent files from being downloaded.   
     
   <http://bit.ly/2G4hAQz>
9. In the command prompt, enter the following and press enter to launch the Jupyter Notebook:  
   jupyter notebook  
     
   In the Command Prompt window, the command to launch the Jupyter Notebook displays.

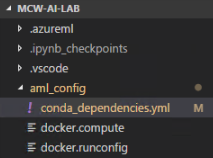
   At this time, we are unable to capture all of the information in the command prompt window. Future versions of this course should address this.
10. In a few moments, you should be prompted for which browser to use to open the link, select **Firefox**.
11. The Jupyter Notebook interface should appear in the browser, listing the contents of your project folder.  
    
12. Select the **code** folder.
13. Select **01\_data\_acquisition\_and\_understanding**.  
    
14. Select the **Upload** button.
15. Open the **Summarize.ipynb** notebook and follow the instructions within it.

#### Task 4: Set Visual Studio Code as the project IDE in Workbench

1. Within Workbench, from the **File** menu, select **Configure Project IDE.**  
   
2. In the **Configure IDE** blade that appears, set the following properties:
   1. **Name**: Visual Studio Code
   2. **Executable Path**: C:\Program Files\Microsoft VS Code\Code.exe  
      
3. Select **OK**.
4. Launch Visual Studio Code for the project by selecting **Open Project (Visual Studio Code)** from the **File** menu.  
   
5. You are now ready to author service script.

#### Task 5: Create the Summarization service

1. Visual Studio Code will open against the project directory.
2. In the tree view, expand **code** and then right-click **03\_deployment** and select **New File.**  
   
3. For the file name, enter summarizer\_service.py and press **Enter**.
4. In a browser, navigate to <http://bit.ly/2FLJn8Y> and copy the contents of the file. (If you have problems with this link, try copy-pasting it into a browser window)
5. Paste the contents of this script into your summarizer\_service.py. Take a moment to review the script, as it is effectively the same code you were running in the Jupyter notebook, except that is has been modified to follow the format required by services in Azure Machine Learning. The init method is called once per container by the Azure Machine Learning infrastructure when the service is deployed. It is here that we need to load all of the modules required by NLTK in the call to nltk.download. The run method is where any scoring (or in our case summarization) activity takes place.   
   In the Command Prompt window, the script that you copied displays.

   At this time, we are unable to capture all of the information in the command prompt window. Future versions of this course should address this.
6. Next, we need to capture the dependencies for the modules used by the script. These are declared in a conda environment file, which you can generate from an environment or create by hand. In this case, we will edit the default conda environment provided by the TDSP project by hand, and add a configuration that will pip install gensim as required by our script. To do this, in Visual Studio Code, expand, aml\_config and open conda\_dependencies.yml.  
   
7. At the last line, under azure-ml-api-sdk add another line with - azureml.datacollector==0.1.0a13, then another line with -gensim to the pip configuration. You should also add entries for tensorflow and tflearn, which we will need later in the lab. Your final configuration should look as follows:

name: project\_environment

dependencies:

- python=3.5.2

- scikit-learn

- pip:

# The API for Azure Machine Learning Model Management Service.

# Details: https://github.com/Azure/Machine-Learning-Operationalization

- azure-ml-api-sdk==0.1.0a11

**- azureml.datacollector==0.1.0a13**

**- gensim**

**- tensorflow**

**- tflearn**

1. Save the file. When we go to create the image in a later step, this file will be included with command.
2. Next, create an empty file called dummy\_model.bin in the 03\_deployment folder. In this case, we don’t have a model to deploy with this service, but we still need to provide one to the CLI as we will see in a moment. An empty file will do.

#### Task 6: Deploy the Summarization service

1. Return to the Workbench and use the **File** menu to open another command prompt.
2. Wait for your ACS cluster to be ready. You can periodically check on the status by running the command shown in the output to the previous step, which is of the form:  
     
   az ml env show -g <resourceGroupName> -n <clusterName>
3. Once your ACS environment has successfully provisioned (the Provisioning State in the aforementioned command will read “Succeeded”), set your default environment with a command of the form:  
     
   az ml env set -g <resourceGroupName> -n <clusterName>  
     
   In the Command Prompt window, the previous commands and their output display.

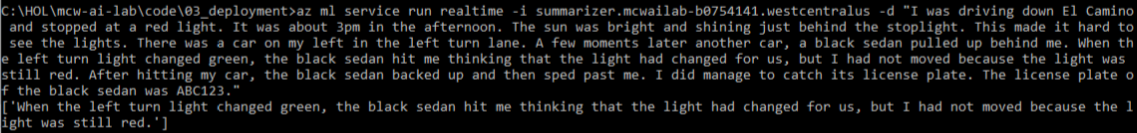
   At this time, we are unable to capture all of the information in the command prompt window. Future versions of this course should address this.
4. This will set the context of the command line to target this environment.
5. Finally, set the model management account to be used by the command line, to be the one you created previously (mcw-ai-lab-model-mgmt). Run the following command, replacing the values indicated in angle brackets with appropriate values.
   1. For <acctname>, enter the name of the Machine Learning Model Management resource in your mcw-ai-lab resource group.
   2. For <resourcegroupname>, use your mcw-ai-lab resource group name.

az ml account modelmanagement set -n <acctname> -g <resourcegroupname>

1. At the command prompt, change directories to the code\03\_deployment directory by executing the following command:

cd code\03\_deployment

1. You can deploy the service using a single command (which orchestrates the multiple steps of creating a docker manifest, creating a docker image, and deploying a container instance from the image). The command needs to refer to all the components required for the service including the dummy model file, the service script, the conda dependencies and the runtime to use (python in this case). Run the following command to deploy the summarizer service:  
     
   az ml service create realtime -n summarizer -c ..\..\aml\_config\conda\_dependencies.yml -m dummy\_model.bin -f summarizer\_service.py -r python  
     
   In the Command Prompt window, the previous command and its output display.

   At this time, we are unable to capture all of the information in the command prompt window. Future versions of this course should address this.
2. Notice in the output of the preceding command, you are provided with instructions (third line from last) on how you can invoke the deployed service using the CLI. Try executing the following command (modify the Service ID of you service as indicated in the previous command output):  
     
   az ml service run realtime -i summarizer.**[mcwailab-xyz.location]** -d "I was driving down El Camino and stopped at a red light. It was about 3pm in the afternoon. The sun was bright and shining just behind the stoplight. This made it hard to see the lights. There was a car on my left in the left turn lane. A few moments later another car, a black sedan pulled up behind me. When the left turn light changed green, the black sedan hit me thinking that the light had changed for us, but I had not moved because the light was still red. After hitting my car, the black sedan backed up and then sped past me. I did manage to catch its license plate. The license plate of the black sedan was ABC123."  
     
   
3. If you get a summary back, your service is working! Try calling the service with other text and observe the summary returned. Note that the service tries to build a summary of about 30 words, so if you provide too short a text, an empty summary will be returned.
4. Finally, in a notepad or other location take note of the full Service ID (e.g., summarizer.mcwailab-xyz.location) and the authorization key which you will need later in the lab. To get the authorization key for your deployed service, run the following command and take note of the PrimaryKey value in the output:  
     
   az ml service keys realtime -i summarizer.**[mcwailab-xyz.location]**In the Command Prompt window, the previous command and its output display.

   At this time, we are unable to capture all of the information in the command prompt window. Future versions of this course should address this.

Stop

## Exercise 4: Train and Deploy a Deep Learning Model

Duration: 60 minutes

In this exercise, you use TensorFlow to construct and train a simple deep neural network classification model that will classify claim text as belonging to a home insurance claim or an automobile claim. You will then deploy this trained model as a web service.

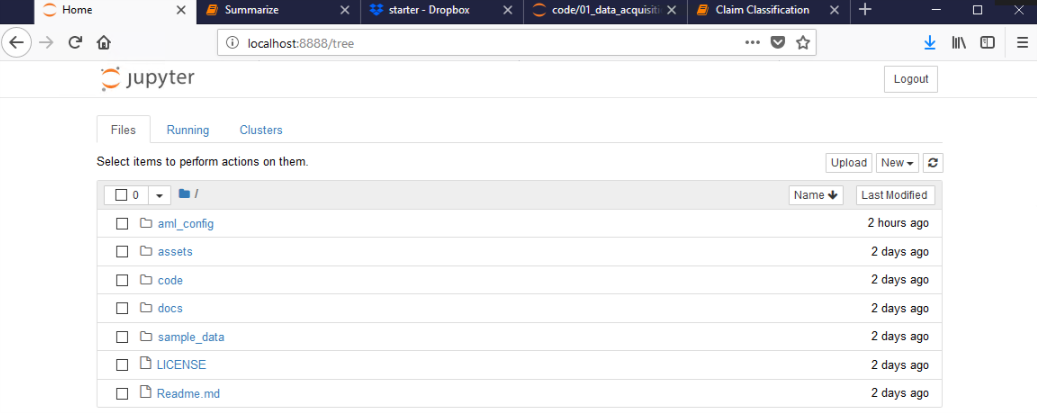
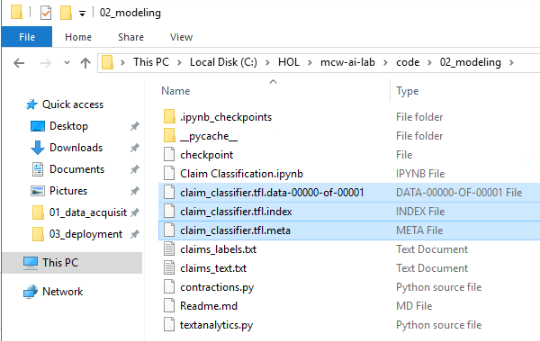
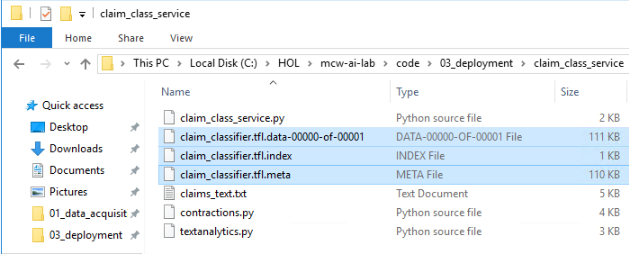
#### Task 1: Prepare TensorFlow

1. Return to your RDP session to the Data Science VM.
2. Switch to the command prompt that is running the Jupyter Notebook command and press **Control + Break**. This will stop the Jupyter Notebook process while you update TensorFlow.
3. From the command line run:  
     
   pip install tensorflow
4. In a few moments, the install should complete, and you should see output ending similar to the following:  
     
   In the Command Prompt window, output indicates that the file was successfully installed.

   At this time, we are unable to capture all of the information in the command prompt window. Future versions of this course should address this.
5. We will be using the TFLearn library which sits atop TensorFlow. To install it run:   
     
   pip install tflearn  
     
   In the Command Prompt window, the installation progress and the success message displays.

   At this time, we are unable to capture all of the information in the command prompt window. Future versions of this course should address this.
6. Run “Jupyter Notebook” to re-start the process.
7. You should now be ready to use TensorFlow on your Data Science VM.

#### Task 2: Train and deploy the TensorFlow model

1. Return to your RDP session to the Data Science VM.
2. Download the TensorFlow notebook, text analytics helper module and sample data from the following link:  
     
   <http://bit.ly/2pucpje>
3. Extract this zip and copy the contents to C:\HOL\mcw-ai-lab\code\02\_modeling.
4. Return to the instance of the Jupyter Notebook home that should be open in your browser.  
   
5. Select the code folder**, 02\_modeling**. You should see a folder listing similar to the following. Select **Claim Classification.ipynb**.  
   
6. The Claim Classification notebook will appear. Step through this notebook to read how the data is prepared and the neural network model is trained. Be sure to execute each cell as you get to it.
7. When you have finished executing the notebook, some model files will have been produced. Using File Explorer, navigate to **C:\HOL\mcw-ai-lab\code\02\_modeling**, you should see the three new files (each beginning with claim\_classifier.tfl).  
   
8. Copy these three files and paste them under C:\HOL\mcw-ai-lab\code\03\_deployment\claim\_class\_service. You are copying these over so they can be used by the predictive web service we will deploy.   
   
9. Next, download the supporting files for the claim\_class\_service from:  
     
   <http://bit.ly/2u5DoGH>
10. Extract the files and copy them into C:\HOL\mcw-ai-lab\code\03\_deployment\claim\_class\_service.
11. Return to the instance of the Jupyter Notebook home that should be open in your browser.
12. Select the code folder, **03\_deployment** and then **claim\_class\_service**.
13. Open **claim\_class\_service.py**. Observe that the code it uses is like what you ran in the Claim Classification notebook, only formatted to fit the structure of an Azure Machine Learning web service (with init and run methods).   
    Screenshot of the Jupyter code window.

    At this time, we are unable to capture all of the information in the Jupyter code window. Future versions of this course should address this.
14. Next, you will deploy this service. Switch to your command line window and navigate to **C:\HOL\mcw-ai-lab\code\03\_deployment\claim\_class\_service**.
15. Run the following command in the context of the claim\_class\_service folder to deploy the service:

az ml service create realtime -n claimclassifier -c ..\..\..\aml\_config\conda\_dependencies.yml -m claim\_classifier.tfl.meta -f claim\_class\_service.py -r python -d claim\_classifier.tfl.data-00000-of-00001 -d claim\_classifier.tfl.index -d claims\_text.txt -d textanalytics.py -d contractions.py  
  
In the Command Prompt window, the previous command and its output displays.

At this time, we are unable to capture all of the information in the command prompt window. Future versions of this course should address this.

1. Next, test the deployed service by running the following command (substitute the values of the Service ID as indicated in the last line of the previous):  
     
   az ml service run realtime -i claimclassifier.**[mcwailab-xyz.location]** -d "A tornado ripped through my home."  
     
     
   In the Command Prompt window, the previous command and its output display.

   At this time, we are unable to capture all of the information in the command prompt window. Future versions of this course should address this.
2. Recall the classifier will return 1 if the text is classified as related to a car insurance claim and 0 if the claim pertains to a home insurance claim. Try submitting a few different sentences to the service.
3. Next, in a notepad or other location take note of the full Service ID (e.g., claimclassifier.mcwailab-xyz.location) and the authorization key which you will need later in the lab. To get the authorization key for your deployed service, run the following command and take note of the PrimaryKey value in the output:  
     
   az ml service keys realtime -i claimclassifier.**[mcwailab-xyz.location]**In the Command Prompt window, the previous command and its output display.

   At this time, we are unable to capture all of the information in the command prompt window. Future versions of this course should address this.
4. Finally, run the following command to retrieve the IP address of your claimclassifier and summarizer services, and note the value in notepad or other location for use later in the lab. The IP address will be the same for both services.

az ml service usage realtime -i claimclassifier.**[mcwailab-xyz.location]**

**In the Command Prompt window, the IP address displays.**

#### Task 3: Complete the solution

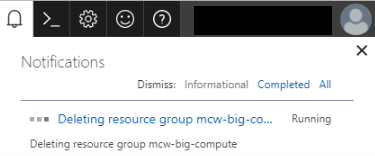
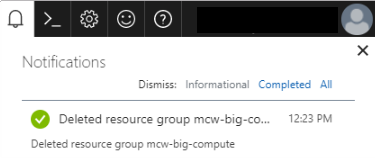
1. Download the starter files for this task from:  
     
   <http://bit.ly/2K0c4wx>
2. Copy this file to C:\HOL\mcw-ai-lab\code\03\_deployment.
3. Return to the instance of the Jupyter Notebook home that should be open in your browser.
4. Select the code folder, **03\_deployment**. Select **Solution Harness.ipynb**.
5. Follow the steps within the notebook to complete the lab and view the result of integrating with your Azure Machine Learning Services.

## After the hands-on lab

Duration: 5 minutes

To avoid unexpected charges, it is recommended you clean up all of your lab resources when you complete the lab.

#### Task 1: Clean up lab resources

1. Navigate to the Azure Portal and locate the Resource Groups you created for this lab
   1. mcw-ai-lab
   2. mcwailabenv (note there are two resources groups starting with this name, so delete both)
2. Select **Delete resource group** from the command bar.  
   Screenshot of the Delete resource group button.
3. In the confirmation dialog that appears, enter the name of the resource group and select **Delete**.
4. Wait for the confirmation that the Resource Group has been successfully deleted. If you don’t wait, and the delete fails for some reason, you may be left with resources running that were not expected. You can monitor using the Notifications dialog, accessible from the Alarm icon.  
   
5. When the Notification indicates success, the cleanup is complete.  
   

You should follow all steps provided *after* attending the Hands-on lab.