

Retrain ML

RETRAIN ML



Retrain ML

In this session

- Retrain workflow
- Create new training experiment
- Create/Publish predictive experiment
- Create/publish a retrain experiment (add IO)
- Create C# console Application BES
- Get keys from Azure Storage Account
- Update C# code input/output
- Get iLearner information
- Review retrain evaluation
- Add a new Endpoint
- Update endpoint

Retrain ML

Retrain workflow

Create the initial Predictive Web service:

- Create a training experiment
- Create a predictive web experiment
- Deploy a predictive web service

Retrain the Web service:

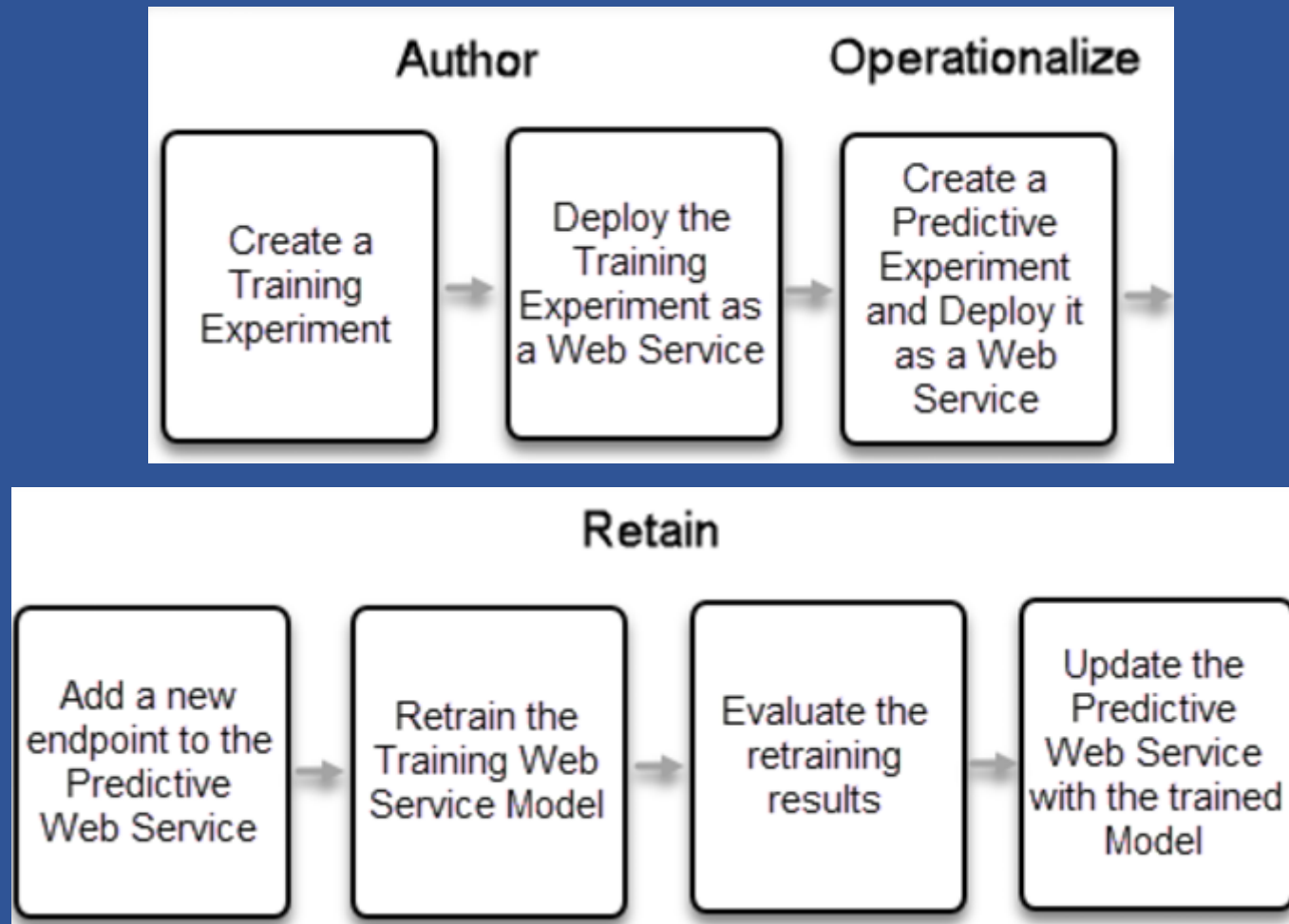
- Update training experiment to allow for retraining
- Deploy the retraining web service
- Use the Batch Execution Service code to retrain the model

Update endpoint

- Create a new Endpoint on the Predictive Web service
- Get the PATCH URL and code
- Use the PATCH URL to point the new Endpoint at the retrained model

Retrain ML

Retrain workflow diagram



Retrain ML






Create new training experiment

Create new ML training experiment by downing an example from **Cortana intelligence gallery**

1. Go to webpage **Cortana intelligence gallery** <https://gallery.cortanaintelligence.com>
2. Enter **loy** in search box
3. Click **Census Model 001**
4. Click **Open in Studio**
5. RUN
6. Click **SET UP WEB SERVICE** and **Predictive web service**
7. RUN
8. Change name of **Predictive experiment** to **Census Model 001 Predic**
9. RUN
10. Click **DEPLOY WEB SERVICE**

Retrain ML


Create a training experiment



 Cortana Intelligence Gallery    




[Browse all](#) [Industries](#) [Solutions](#) [Experiments](#) [More](#)

EXPERIMENT

Census Model 001

 Laploy V. Angkul • July 9, 2017


 edit 

Summary

This experiment demonstrates how we can build a binary classification model to predict income levels of adult individuals. The process includes training, testing and evaluating the model on the Adult dataset.

Description



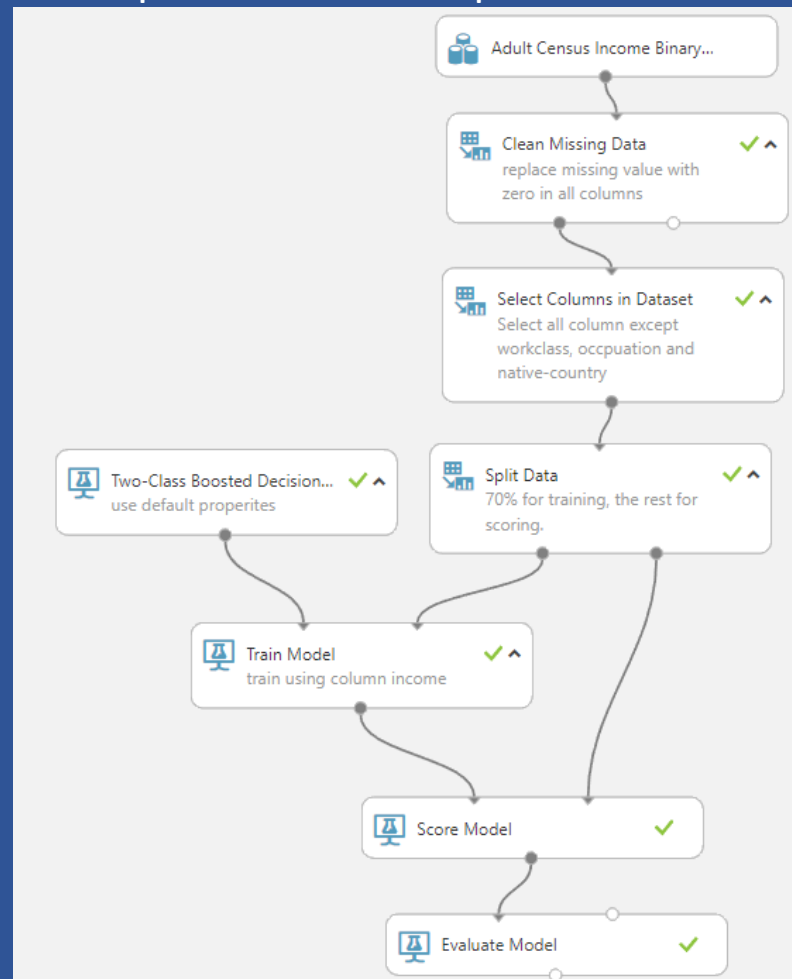
Open in Studio

+ Add to Collection

Retrain ML

Create new training experiment

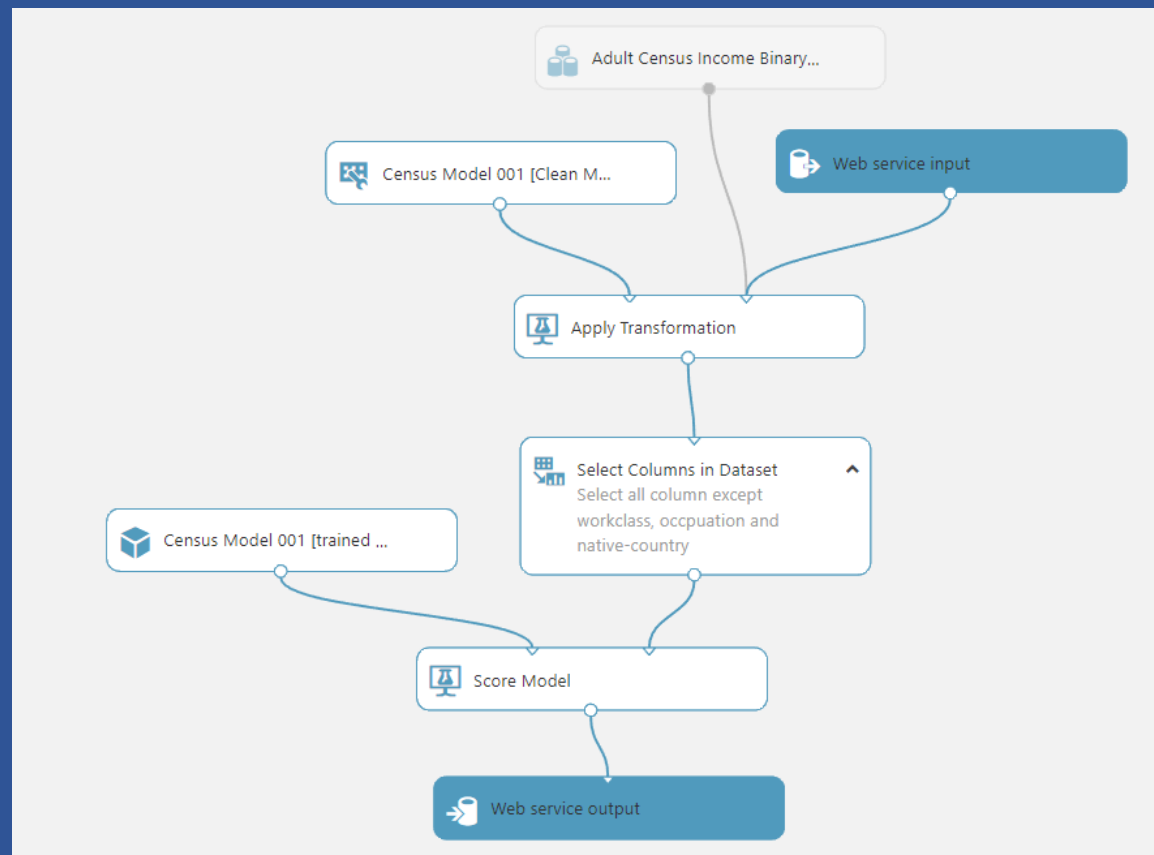
Experiment when open in Studio



Retrain ML

Create/Publish predictive experiment

RUN, SET UP WEB SERVICE / Predictive Web Service [Recommended]



Experiment after SET UP WEB SERVICE and Predictive web service

Retrain ML

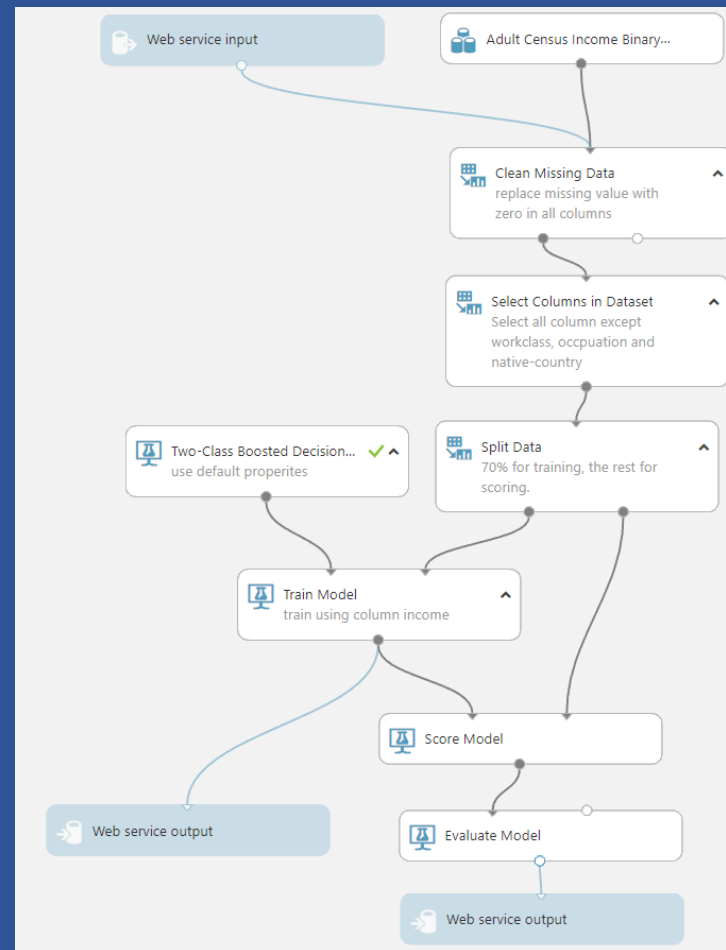
Create/publish a retrain experiment (add IO)

1. Go back to **Census Model 001** Experiment
2. Click **Training experiment tab**
3. Add a **web service input module**
4. Add two **web service output modules**
5. Run
6. Click **SET UP WEB SERVICE / DEPLOY WEB SERVICE**

Retrain ML

Create/publish a retrain experiment (add IO)

Experiment after add web service input / outputs

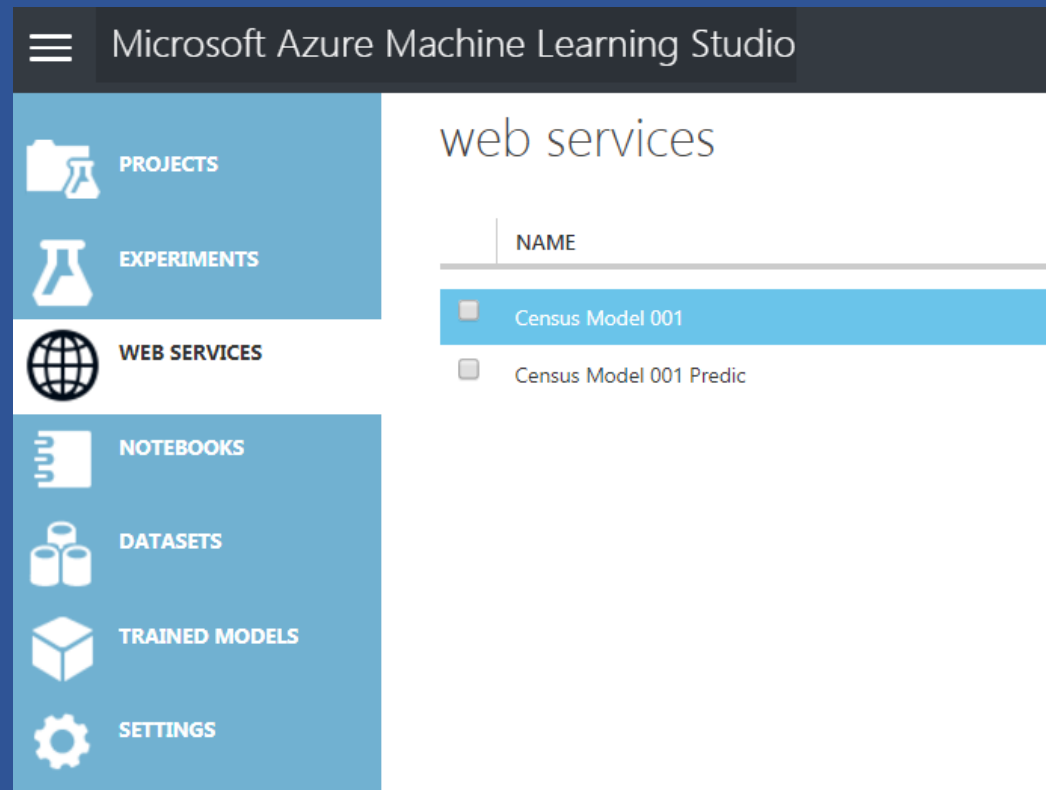


Retrain ML

Create/publish a retrain experiment (add IO)

Click WEB SERVICES

1. Census Model 001 = retrain
2. Census Model 001 Predic = production



Retrain ML

Create C# console Application BES

1. Create a C# Console Application in Visual Studio (New->Project->Windows Desktop->Console Application)
2. Solution/project name = census1
3. Nuget add **Microsoft.WindowsAzure.Storage.dll**
4. Nuget add **Microsoft.AspNet.WebApi.Client**
5. Open Microsoft Azure Machine Learning Studio page
6. Click **Web Service**
7. Click **census model 001**
8. Click **BATCH EXECUTION**
9. Copy C# sample code
10. Past code ** Note on name space

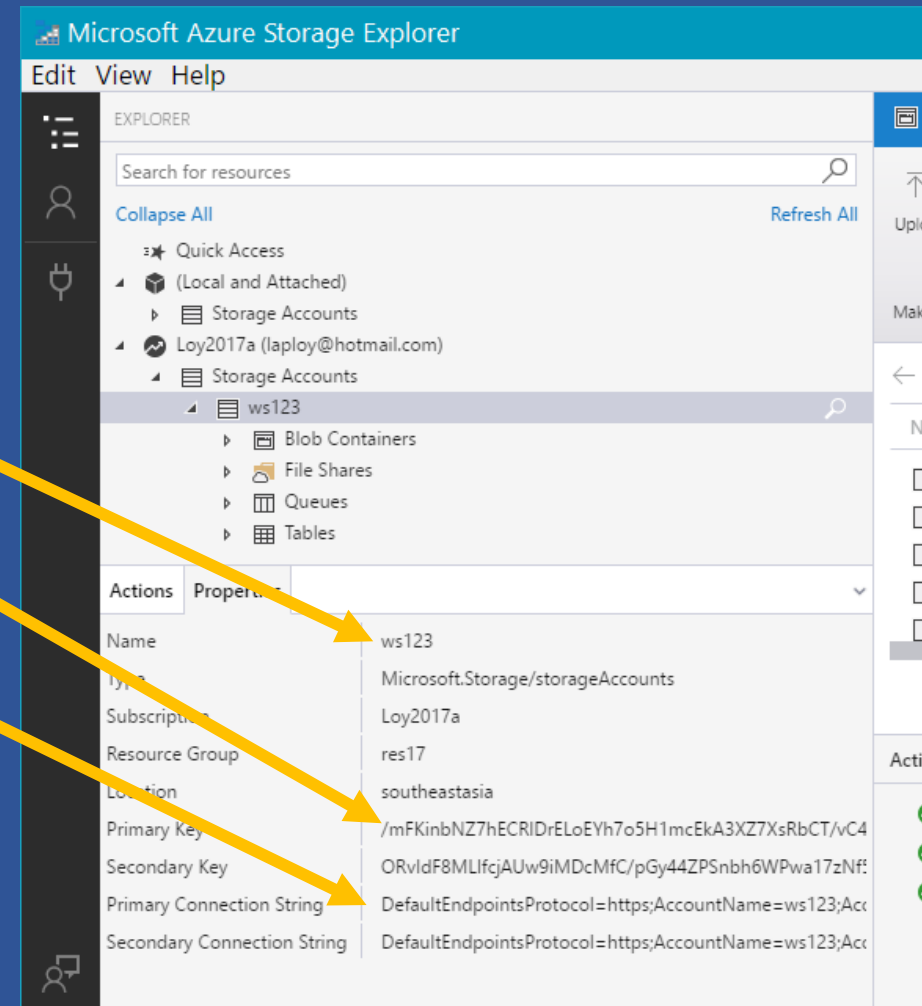
Retrain ML

Get keys from Azure Storage Account

Run program **Microsoft Azure Storage Explorer**

Copy and save to Notepad

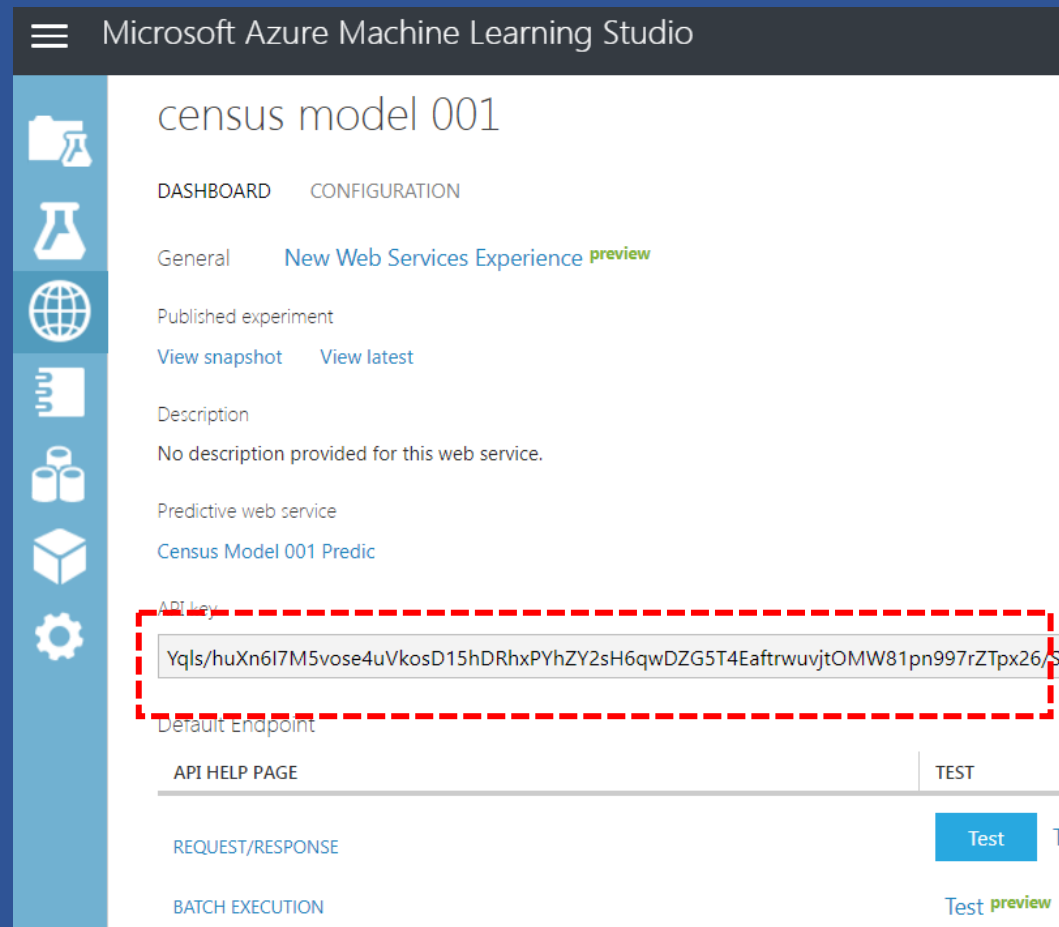
1. Storage Container Name
2. Storage Account Key
3. Storage Connection String



Retrain ML

Update C# code input/output

1. Get Web service **API Key** from AML BES page (copy and save to Notepad)



Retrain ML

Update C# code input/output

Update C# code for keys

```
// ***** loy
const string StorageAccountName = "ws123"; // Replace this with yo
const string StorageAccountKey = "/mFKinbNZ7hECR1DrELoEYh7o5H1mcEk
const string StorageContainerName = "test1"; // Replace this with y
string storageConnectionString = string.Format("DefaultEndpointsPr
const string apiKey = "mbrj11ijM8MB3IyQ5h08tZJbnn+101Ru00RWCS6xQ50
```

Retrain ML

Update C# code input/output

- Update C# code UploadFileToBlob

```
// ***** loy
UploadFileToBlob(@"d:\temp\cenin1.csv" /*Replace this with the location of your input file*/,
    "cenin1.csv" /*Replace this with the name you would like to use for your Azure blob; this r
    StorageContainerName, storageConnectionString);
```

- Update C# code input file name

```
Inputs = new Dictionary<string, AzureBlobDataReference>()
{
    {
        "input1",
        new AzureBlobDataReference()
        {
            ConnectionString = storageConnectionString,
            // ***** loy
            RelativeLocation = string.Format("{0}/cenin1.csv", StorageContainerName)
        }
    },
},
```


Retrain ML

Update C# code input/output

Update C# code Output1 / Output2 file name

```
Outputs = new Dictionary<string, AzureBlobDataReference>()
{
    {
        "output2",
        new AzureBlobDataReference()
        {
            ConnectionString = storageConnectionString,
            // ***** loy
            RelativeLocation = string.Format("/{0}/cenout.ilearner", StorageContainerName)
        }
    },
    {
        "output1",
        new AzureBlobDataReference()
        {
            ConnectionString = storageConnectionString,
            // ***** loy
            RelativeLocation = string.Format("/{0}/cenout.csv", StorageContainerName)
        }
    },
}
```

Retrain ML

Update C# code input/output

Update C# code Main method to show **End program** status

0 references

```
static void Main(string[] args)
{
    InvokeBatchExecutionService().Wait();
    // ***** loy
    Console.WriteLine("End program");
    Console.Read();
}
```

Retrain ML

Retrain and evaluate

Run program

1. Download file `cenin1.csv` from <https://github.com/laploy/ML/blob/master/cenin1.csv>
2. Place file `cenin1.csv` in to `d:\temp`
3. Run C# Program
4. Wait for `End program` message

Run this program whenever you have a good training dataset and want to retrain the model

Retrain ML

Get iLearner information

Get ilearner information from program output

Copy and paste to Notepad

1. RelativeLocation
2. BaseLocation
3. SasBlobToken

iLearner is the training model we need
to update the Prediction model

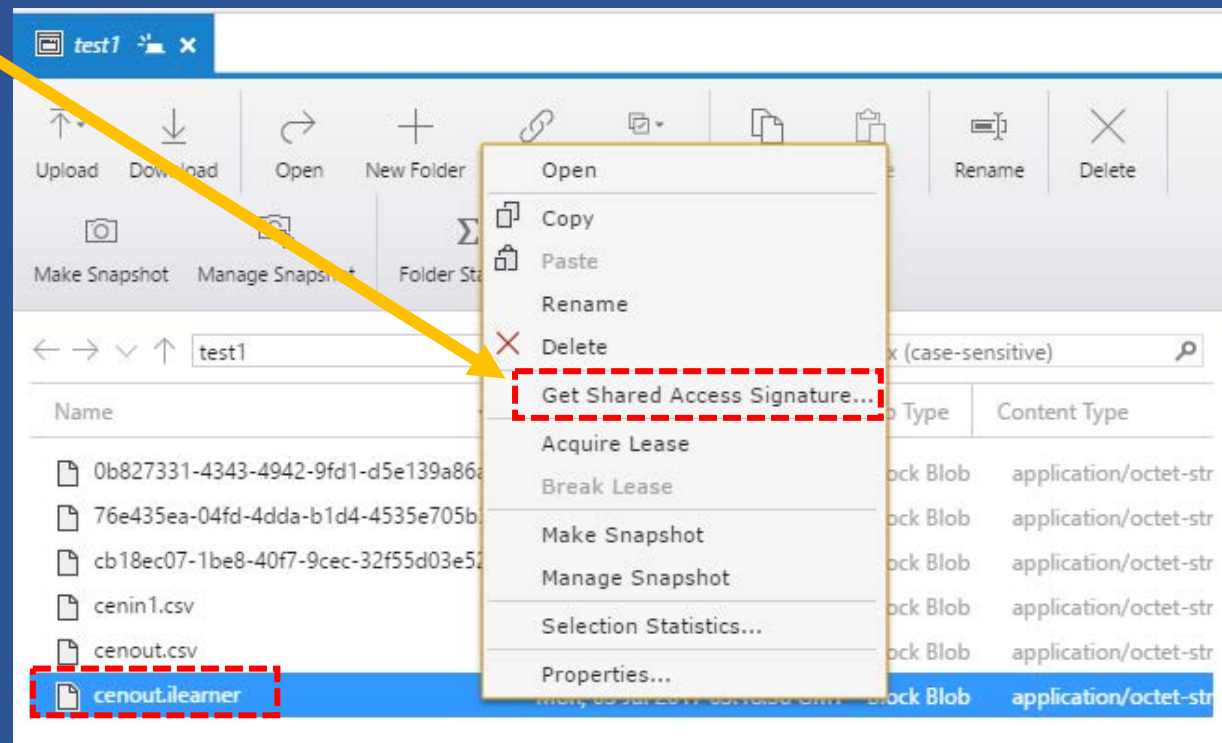
```
The result 'output1' is available at the following A
BaseLocation: https://aihelpwebsitestorage.blob.core
RelativeLocation: experimentoutput/output1results.il
SasBlobToken: ?sv=2015-02-21&sr=b&sig=0EJsI719TZ39st
nyw%3D&st=2017-03-27FnVMgsavxnK0unZ&se=2017-03-27T019
```

Retrain ML

Get iLearner information

Or using ASE

1. Open **Microsoft Azure Storage Explorer**
2. Right-click at file **cenout.ilearner**
3. Click **Get Shared Access Signature**



Retrain ML

Get iLearner information

Click Create

Shared Access Signature

Access policy: None

Start time: 07/11/2017 07:59 AM

Expiry time: 07/12/2017 07:59 AM

Time zone:

☒ Local

☐ UTC

Permissions:

☒ Read

☐ Write

☐ Delete

☒ List

☐ Generate container-level shared access signature URI

Create

Cancel

Retrain ML

Get iLearner information

Copy and paste to Notepad

- 4. RelativeLocation
- 5. BaseLocation
- 6. SasBlobToken

Shared Access Signature

Blob:
cenout.ilearner

URL:
https://ws123.blob.core.windows.net/test1/cenout.ilearner?st=2017-07-10T17%3A59%3A00Z&se=2017-07-11T17%3A59%3A00Z

Query string:
?st=2017-07-10T17%3A59%3A00Z&se=2017-07-11T17%3A59%3A00Z

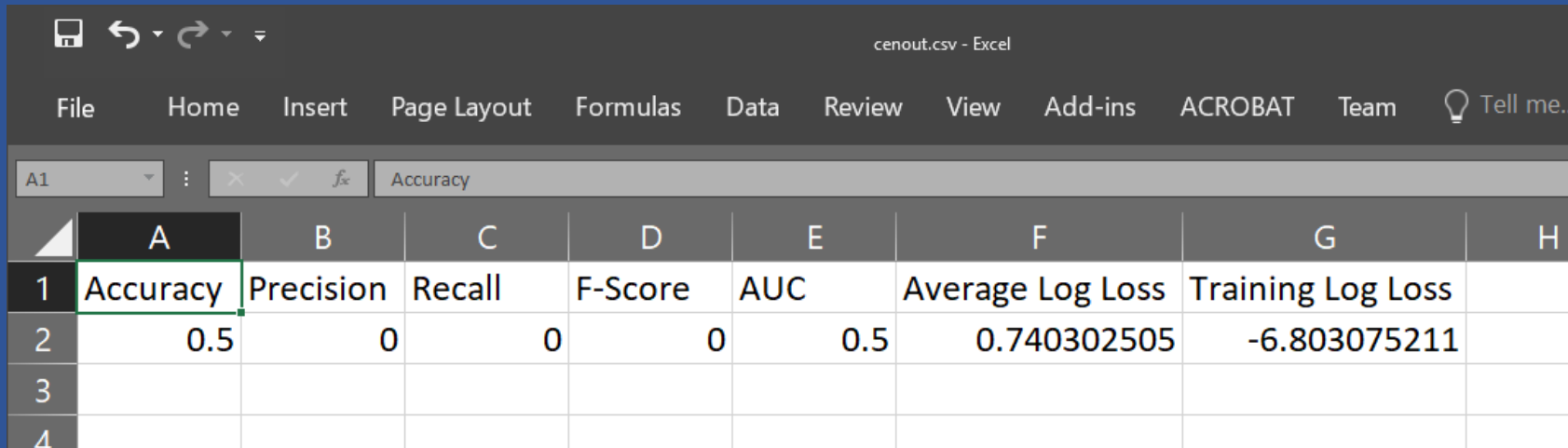
Base location = from start to .net

Retrain ML

Review retrain evaluation

Review retrain evaluation

1. Open **Microsoft Azure Storage Explorer**
2. Download file **cenout.csv**
3. Open in **Microsoft Excel** or **Windows Notepad**
4. Examine the results

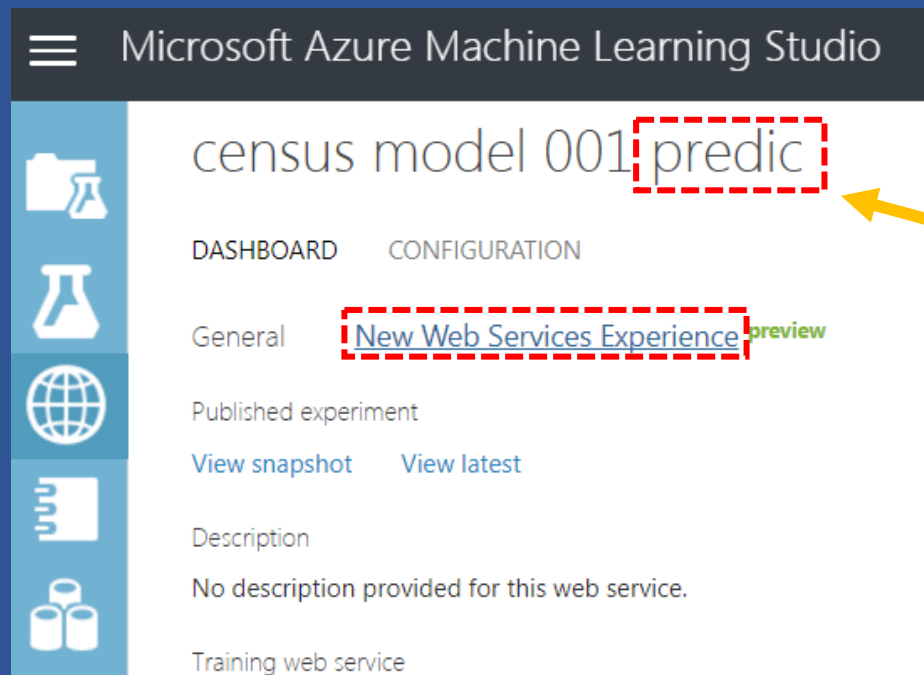


	A	B	C	D	E	F	G	H
1	Accuracy	Precision	Recall	F-Score	AUC	Average Log Loss	Training Log Loss	
2	0.5	0	0	0	0.5	0.740302505	-6.803075211	
3								
4								

Retrain ML

Add a new Endpoint

Click **New Web Services Experience**

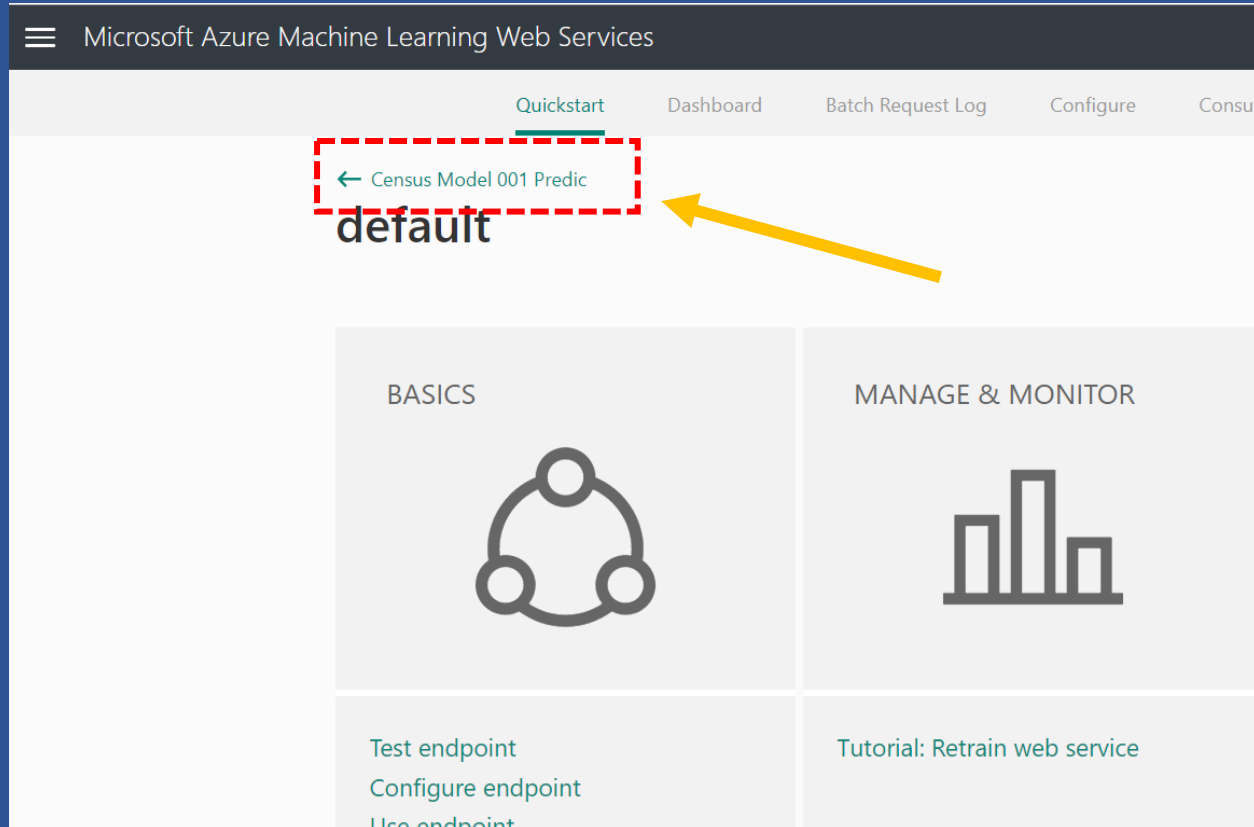


Prediction WS not Retrain WS

Retrain ML

Add a new Endpoint

Click **Census Model 001 Predic**



Retrain ML

Add a new Endpoint

Click **+ NEW**


Microsoft Azure Machine Learning Web Services

← Classic Web Services

Census Model 001 Predic

No description provided for this web service.

Search

+ NEW  DELETE

	NAME	BATCH CALLS	FAILURES
<input type="checkbox"/>	default	0	0

Retrain ML

Add a new Endpoint

Enter name, description and click Save

Microsoft Azure Machine Learning Web Services

Census Model 001 Predic

No description provided for this web service.

Search

+ NEW DELETE

Create new endpoint

Name: retrain

Description: retrain test 1

Logging: ☒ None ☐ Error ☐ All [Logging Help](#)

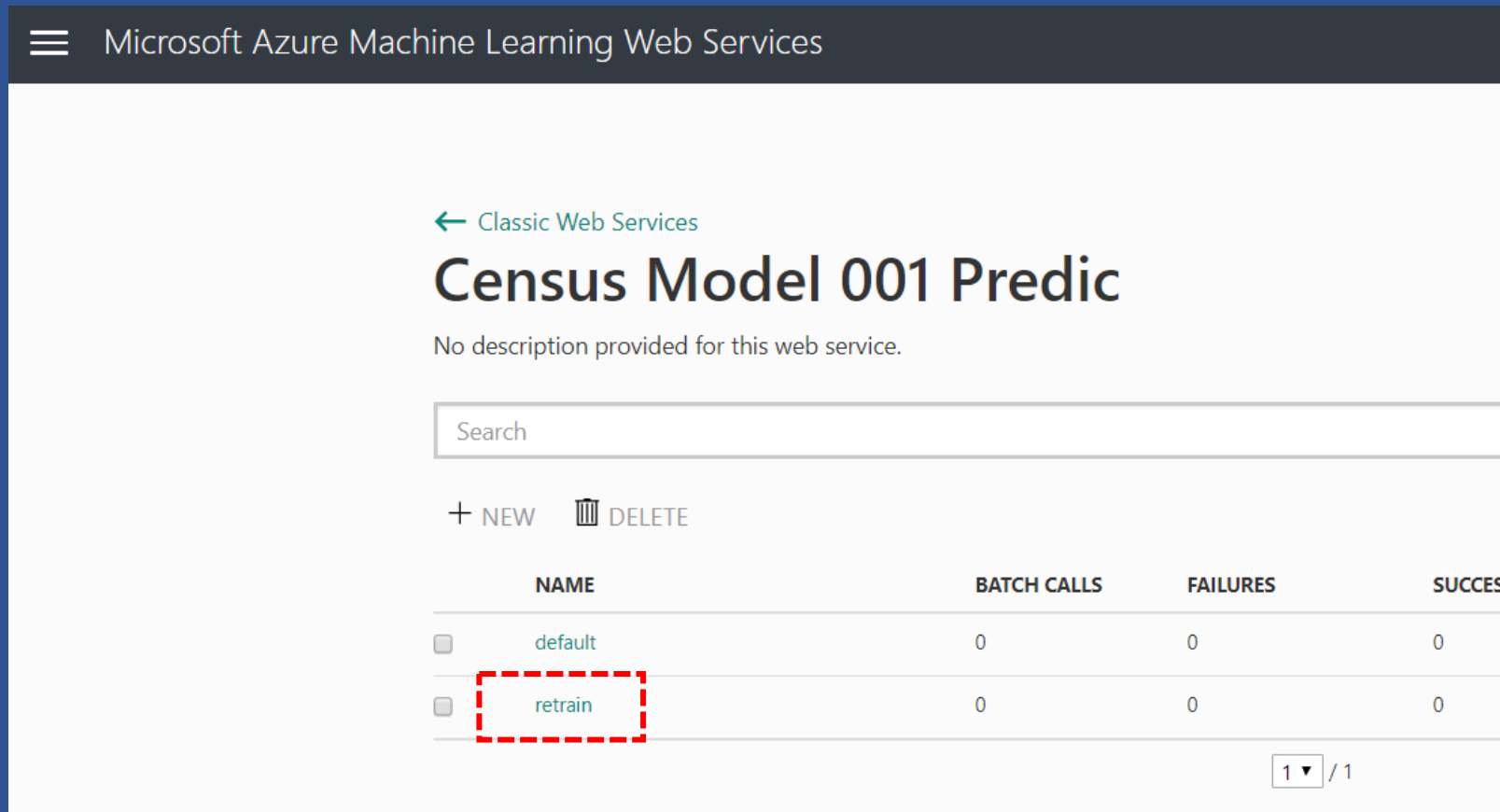
Sample Data Enabled?: ☐ Yes ☒ No

Cancel Save

Retrain ML

Update endpoint

Click **retrain** end point




Microsoft Azure Machine Learning Web Services

← Classic Web Services

Census Model 001 Predic

No description provided for this web service.

Search

+ NEW  DELETE

	NAME	BATCH CALLS	FAILURES	SUCCESS
<input type="checkbox"/>	default	0	0	0
<input type="checkbox"/>	retrain	0	0	0

1 / 1

Retrain ML

Update endpoint

Click **Consume**

The screenshot displays the Microsoft Azure Machine Learning Web Services interface. At the top, a dark header bar contains the text 'Microsoft Azure Machine Learning Web Services' and a hamburger menu icon. Below this, a navigation bar features several tabs: 'Quickstart' (highlighted with a green underline), 'Dashboard', 'Batch Request Log', 'Configure', 'Consume' (highlighted with a red dashed border), and 'Telemetry'. A yellow arrow points from the text 'Click Consume' to the 'Consume' tab. The main content area shows a breadcrumb trail '← Census Model 001 Predic' followed by the title 'retrain' and the subtitle 'retrain end point test'. Below this, there are three panels: 'BASICS' with a circular diagram icon, 'MANAGE & MONITOR' with a bar chart icon, and a partially visible 'DEV' panel.

Retrain ML

Update endpoint

Click **API Help**

Microsoft Azure Machine Learning Web Services

Quickstart Dashboard Batch Request Log

retrain end point test

Web service consumption options

Excel 2013 or later Excel 2010 or earlier Request-Response Web App Template

Basic consumption info

Want to see how to consume this information? [Check out this easy tutorial.](#)

Primary Key vz0MsD4YYo3dCSyhQsqTnTxCeehQtqxxaBh9vNONG611

Secondary Key 6M+Zn4xmCcq1A4Eg/WFSv7Frz9PEohHmVcuhDIse5LkV

Request-Response <https://ussouthcentral.services.azureml.net/workspaces/7193/execute?api-version=2.0&format=swagger>
[API Help](#) [Documentation](#)

Batch Requests <https://ussouthcentral.services.azureml.net/workspaces/7193/jobs?api-version=2.0>
[API Help](#) [Documentation](#)

Patch <https://management.azureml.net/workspaces/ede12cb3/endpoints/retrain>
[API Help](#) [Documentation](#)

Retrain ML

Update endpoint

Click [Sample Code](#)

Update Resource API Documentation

Updated: 07/10/2017 04:08

No description provided for this web service.

- [Request and Response summary](#)
- [Sample Code](#)
- [API Swagger Document](#) ?
- [Endpoint Managment Swagger Document](#) ?

Updatable Resources

Resource Name

Copy C# code

Retrain ML

Update endpoint

Create a program to update the endpoint

- Open Visual Studio
- Create new C# Windows console app project
- Name = CallUpdateResource
- Paste code to main
- Update
 1. const string apiKey: get key from web service page
 2. BaseLocation
 3. RelativeLocation
 4. SasBlobToken
 5. Add a message to show success end
- Run program

Run this program only once

Retrain ML

More information

Retrain a Machine Learning Model

<https://docs.microsoft.com/en-us/azure/machine-learning/machine-learning-retrain-machine-learning-model>

Census Model 001

<https://gallery.cortanaintelligence.com/Experiment/Census-Model-001>