# Lab Predictive modeling

## Set up your environment

Task 1: Provision Azure Databricks

Azure Databricks is an Apache Spark-based analytics platform optimized for Azure. It will be used in this lab to build and train a machine learning model used to predict flight delays.

1. In the Azure Portal (https://portal.azure.com), select + Create a resource, then type "Azure Databricks" into the search bar. Select Azure Databricks from the results.
2. Select Create on the bottom of the blade that follows.
3. Set the following configuration on the Azure Databricks Service creation form:
   1. Name: Enter a unique name as indicated by a green checkmark.
   2. Subscription: Select the subscription you are using for this hands-on lab.
   3. Resource Group: Select the same resource group you created at the beginning of this lab.
   4. Location: Select a region close to you. (If you are using an Azure Pass, select South Central US.)
   5. Pricing: Select Premium.
4. Select Create to finish and submit.

Task 2: Create Azure Storage account

Create a new Azure Storage account that will be used to store historic and scored flight and weather data sets for the lab.

1. In the Azure Portal (https://portal.azure.com), select + Create a resource, then type "storage" into the search bar. Select Storage account - blob, file, table, queue from the results.
2. Select Create on the bottom of the blade that follows.
3. Set the following configuration on the Azure Storage account creation form:
   1. Subscription: Select the subscription you are using for this hands-on lab.
   2. Resource group: Select the same resource group you created at the beginning of this lab.
   3. Storage account name: Enter a unique name as indicated by a green checkmark.
   4. Location: Select the same region you used for Azure Databricks.
   5. Performance: Standard
   6. Account kind: BlobStorage
   7. Replication: Read-access geo-redundant storage (RA-GRS)
   8. Access tier: Hot
4. Select Create to finish and submit.

Task 3: Retrieve Azure Storage account information and create container

You will need to have the Azure Storage account name and access key when you create your Azure Databricks cluster during the lab. You will also need to create storage containers in which you will store your flight and weather data files.

1. From the side menu in the Azure portal, choose Resource groups, then enter your resource group name into the filter box, and select it from the list.
2. Next, select your lab Azure Storage account from the list.
3. Select Access keys (1) from the menu. Copy the storage account name (2) and the key1 key (3) and copy the values to a text editor such as Notepad for later.
4. Select Blobs (1) from the menu. Select + Container (2) on the Blobs blade, enter sparkcontainer for the name (3), leaving the public access level set to Private. Select OK (4) to create the container.

Task 4: Provision Azure Data Factory

Create a new Azure Data Factory instance that will be used to orchestrate data transfers for analysis.

1. In the Azure Portal (https://portal.azure.com), select + Create a resource, then type "Data Factory" into the search bar. Select Data Factory from the results.
2. Select Create on the bottom of the blade that follows.
3. Set the following configuration on the Data Factory creation form:
   1. Name: Enter a unique name as indicated by a green checkmark.
   2. Subscription: Select the subscription you are using for this hands-on lab.
   3. Resource Group: Select the same resource group you created at the beginning of this lab.
   4. Version: V2
   5. Location: Select any region close to you.

Understanding Data Factory Location: The Data Factory location is where the metadata of the data factory is stored and where the triggering of the pipeline is initiated from. Meanwhile, a data factory can access data stores and compute services in other Azure regions to move data between data stores or process data using compute services. This behavior is realized through the globally available IR to ensure data compliance, efficiency, and reduced network egress costs.

The IR Location defines the location of its back-end compute, and essentially the location where the data movement, activity dispatching, and SSIS package execution are performed. The IR location can be different from the location of the data factory it belongs to.

1. Select Create to finish and submit.

Task 5: Create an Azure Databricks cluster

You have provisioned an Azure Databricks workspace, and now you need to create a new cluster within the workspace. Part of the cluster configuration includes setting up an account access key to your Azure Storage account, using the Spark Config within the new cluster form. This will allow your cluster to access the lab files.

1. From the side menu in the Azure portal, select Resource groups, then enter your resource group name into the filter box, and select it from the list.
2. Next, select your Azure Databricks service from the list.
3. In the Overview pane of the Azure Databricks service, select Launch Workspace. Azure Databricks will automatically log you in using Azure Active Directory Single Sign On.
4. Select Clusters (1) from the menu, then select Create Cluster (2).
5. On the Create New Cluster form, provide the following:
   1. Cluster Name: lab
   2. Cluster Type: Standard
   3. Databricks Runtime Version: 4.3 (includes Apache Spark 2.3.1, Scala 2.11)
   4. Python Version: 3
   5. Driver Type: Same as worker
   6. Worker Type: Standard\_F4s
   7. Workers: 1
   8. Enable Autoscaling: Uncheck this option.
   9. Auto Termination: Check the box and enter 120.
   10. Spark Config: Edit the Spark Config by entering the connection information for your Azure Storage account that you copied earlier in Task 5. This will allow your cluster to access the lab files. Enter the following:

spark.hadoop.fs.azure.account.key.<STORAGE\_ACCOUNT\_NAME>.blob.core.windows.net <ACCESS\_KEY>, where <STORAGE\_ACCOUNT\_NAME> is your Azure Storage account name, and <ACCESS\_KEY> is your storage access key.

1. Select Create Cluster.

## Load Sample Data and Databricks Notebooks

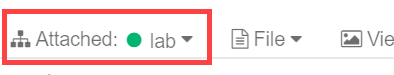
In this exercise, you will implement a classification experiment. You will load the training data from your local machine into a dataset. Then, you will explore the data to identify the primary components you should use for prediction, and use two different algorithms for predicting the classification. You will then evaluate the performance of both algorithms and choose the algorithm that performs best. The model selected will be exposed as a web service that is integrated with the optional sample web app at the end.

Task 1: Upload the Sample Datasets

1. Before you begin working with machine learning services, there are three datasets you need to load.
2. Download the three CSV sample datasets from here: http://bit.ly/2wGAqrl (If you get an error, or the page won't open, try pasting the URL into a new browser window and verify the case sensitive URL is exactly as shown). If still having trouble, a zip file called AdventureWorksTravelDatasets.zip is included in the lab-files folders.
3. Extract the ZIP and verify you have the following files:
   1. FlightDelaysWithAirportCodes.csv
   2. FlightWeatherWithAirportCodes.csv
   3. AirportCodeLocationLookupClean.csv
4. Open a browser and navigate to the Azure portal (https://portal.azure.com), and navigate to Azure Databricks service under the Resource Group you created when completing the prerequisites for this hands-on lab.
5. In the Overview pane of the Azure Databricks service, select Launch Workspace.
6. Once signed in, select Data from the menu. Next, select default under Databases (if this does not appear, start your cluster). Finally, select Add Data above the Tables header.
7. Select Upload File under Create New Table, and then select either select or drag-and-drop the FlightDelaysWithAirportCodes.csv file into the file area. Select Create Table with UI.
8. Select your cluster to preview the table, then select Preview Table.
9. Change the Table Name to "flight\_delays\_with\_airport\_codes" and select the checkmark for First row is header. Select Create Table.
10. Repeat the previous steps for the FlightWeatherWithAirportCode.csv and AirportCodeLocationsClean.csv files, setting the name for each dataset in a similar fashion. There should be a total of three files that are uploaded. Each table should be named as follows:
    1. flightweatherwithairportcode\_csv to flight\_weather\_with\_airport\_code
    2. flightdelayswithairportcodes\_csv to flight\_delays\_with\_airport\_codes
    3. airportcodelocationlookupclean\_csv to airport\_code\_location\_lookup\_clean

Task 2: Open Azure Databricks and complete lab notebooks

1. Within Azure Databricks, select Workspace on the menu, then Users, select your user, then select the down arrow on the top of your user workspace. Select Import.
2. Within the Import Notebooks dialog, select Import from: file, then drag-and-drop the file or browse to upload it.
3. Select BigDataVis to open the notebook.
4. Before you begin, make sure you attach your cluster to the notebooks, using the dropdown. You will need to do this for each notebook you open. There are 5 notebooks included in the BigDataVis.dbc.



1. Run each cell of the notebooks 01, 02 and 03 individually by selecting within the cell, then entering Ctrl+Enter on your keyboard. Pay close attention to the instructions within the notebook so you understand each step of the data preparation process.
2. Do NOT run the Clean up part of Notebook 3 (i.e. this command: webservice.delete()). You will need the URL of your Machine Learning Model exposed later in Exercise 7: Deploy intelligent web app (Optional Lab). Note: you could get this URL by updating your Notebook by adding this line print(webservice.scoring\_uri) or by going to your Azure Machine Learning service workspace via the Azure portal and then to the "Deployments" blade.
3. Do NOT run Notebooks 4 and 5 yet, they will be discussed later in the lab.

## Setup Azure Data Factory

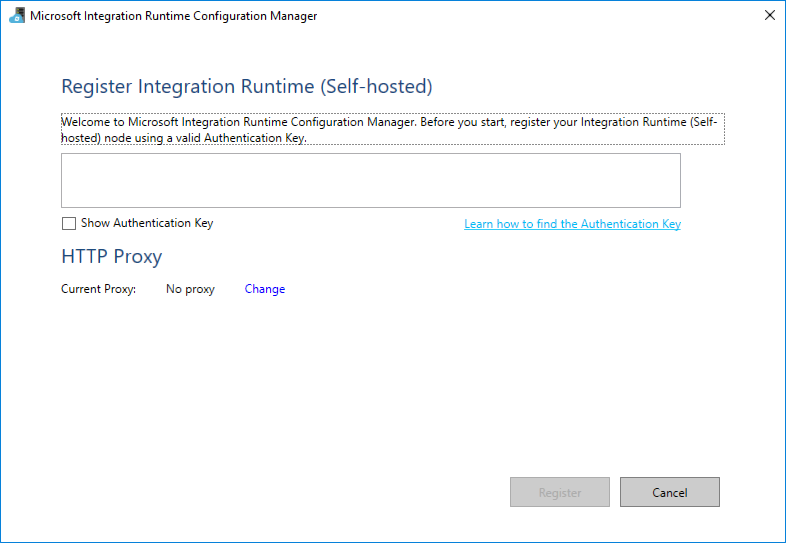
In this exercise, you will create a baseline environment for Azure Data Factory development for further operationalization of data movement and processing. You will create a Data Factory service, and then install the Data Management Gateway which is the agent that facilitates data movement from on-premises to Microsoft Azure.

Task 1: Download and stage data to be processed

1. Open a web browser.
2. Download the AdventureWorks sample data from http://bit.ly/2zi4Sqa.
3. Extract it to a new folder called C:\Data.

Task 2: Install and configure Azure Data Factory Integration Runtime on your machine

1. To download the latest version of Azure Data Factory Integration Runtime, go to https://www.microsoft.com/en-us/download/details.aspx?id=39717.
2. Select Download, then choose the download you want from the next screen.
3. Run the installer, once downloaded.
4. After selecting Finish, the following screen will appear. Keep it open for now. You will come back to this screen once the Data Factory in Azure has been provisioned, and obtain the gateway key so we can connect Data Factory to this "on-premises" server.



Task 3: Configure Azure Data Factory

1. Launch a new browser window, and navigate to the Azure portal (https://portal.azure.com). Once prompted, log in with your Microsoft Azure credentials. If prompted, choose whether your account is an organization account or a Microsoft account. This will be based on which account was used to provision your Azure subscription that is being used for this lab.
2. From the side menu in the Azure portal, choose Resource groups, then enter your resource group name into the filter box, and select it from the list.
3. Next, select your Azure Data Factory service from the list.
4. On the Data Factory blade, select Author & Monitor under Actions.
5. A new page will open in another tab or new window. Within the Azure Data Factory site, select Author (the pencil icon) on the menu.
6. Now, select Connections at the bottom of Factory Resources (1), then select the Integration Runtimes tab (2), and finally select + New (3).
7. In the Integration Runtime Setup blade that appears, select "Perform data movement and dispatch activities to external computes", then select Next.
8. Select Self-Hosted then select Next.
9. Enter a Name, such as bigdatagateway-[initials], and select Next.
10. Under Option 2: Manual setup, copy the Key1 authentication key value by selecting the Copy button, then select Finish.
11. Don't close the current screen or browser session.
12. Paste the Key1 value into the box in the middle of the Microsoft Integration Runtime Configuration Manager screen.
13. Select Register.
14. It will take a minute or two to register. If it takes more than a couple of minutes, and the screen does not respond or returns an error message, close the screen by selecting the Cancel button.
15. The next screen will be New Integration Runtime (Self-hosted) Node. Select Finish.
16. You will then get a screen with a confirmation message. Select the Launch Configuration Manager button to view the connection details.
17. You can now return to the Azure portal, and view the Integration Runtime you just configured.
18. Select the Azure Data Factory Overview button on the menu. Leave this open for the next exercise.

## Develop a data factory pipeline for data movement

In this exercise, you will create an Azure Data Factory pipeline to copy data (.CSV files) from an on-premises server (your machine) to Azure Blob Storage. The goal of the exercise is to demonstrate data movement from an on-premises location to Azure Storage (via the Integration Runtime).

Task 1: Create copy pipeline using the Copy Data Wizard

1. Within the Azure Data Factory overview page, select Copy Data.
2. In the Copy Data properties, enter the following:
   1. Task name: CopyOnPrem2AzurePipeline
   2. Task description: (Optional) "This pipeline copies timesliced CSV files from on-premises C:\Data to Azure Blob Storage as a continuous job".
   3. Task cadence or Task schedule: Select Run regularly on schedule
   4. Trigger type: Select Schedule
   5. Start date time (UTC): 03/01/2018 12:00 am
   6. Recurrence: Select Monthly, and every 1 month
   7. Under the Advanced recurrence options, make sure you have a value in the textboxes for Hours (UTC) and Minutes (UTC) otherwise it will fail later during Publishing.
   8. End: No End
3. Select Next.
4. On the Source data store screen, select + Create new connection.
5. Scroll through the options and select File System, then select Continue.
6. In the New Linked Service form, enter the following:
   1. Name: OnPremServer
   2. Connect via integration runtime: Select the Integration runtime created previously in this exercise.
   3. Host: C:\Data
   4. User name: Use your machine's login username
   5. Password: Use your machine's login password
7. Select Test connection to verify you correctly entered the values. Finally, select Finish.
8. On the Source data store page, select Next.
9. On the Choose the input file or folder screen, select Browse, then select the FlightsAndWeather folder. Next, check Copy file recursively, then select Next.
10. On the File format settings page, select the following options:
    1. File format: Text format
    2. Column delimiter: Comma (,)
    3. Row delimiter: Carriage Return + Line feed (\r\n)
    4. Skip line count: 0
    5. Source files contain column names in the first row: Checked
    6. Treat empty column value as null: Checked
11. Select Next.
12. On the Destination screen, select + Create new connection.
13. Select Azure Blob Storage within the New Linked Service blade, then select Continue.
14. On the New Lined Service (Azure Blob Storage) account screen, enter the following and then choose Finish.
    1. Name: BlobStorageOutput
    2. Connect via integration runtime: Select your Integration Runtime.
    3. Azure selection method: From Azure subscription
    4. Storage account name: Select the blob storage account you provisioned in the setup section.
15. On the Destination data store page, select Next.
16. From the Choose the output file or folder tab, enter the following:
    1. Folder path: sparkcontainer/FlightsAndWeather/{Year}/{Month}/
    2. Filename: FlightsAndWeather.csv
    3. Year: Select yyyy from the drop down.
    4. Month: Select MM from the drop down.
    5. Copy behavior: Merge files
    6. Select Next
17. On the File format settings screen, select the Text format file format, and check the Add header to file checkbox, then select Next.
18. On the Settings screen, select Skip incompatible rows under Actions. Expand Advanced settings and set Degree of copy parallelism to 10, then select Next.
19. Review settings on the Summary tab, but DO NOT choose Next.
20. Scroll down on the summary page until you see the Copy Settings section. Select Edit next to Copy Settings.
21. Change the following Copy settings:



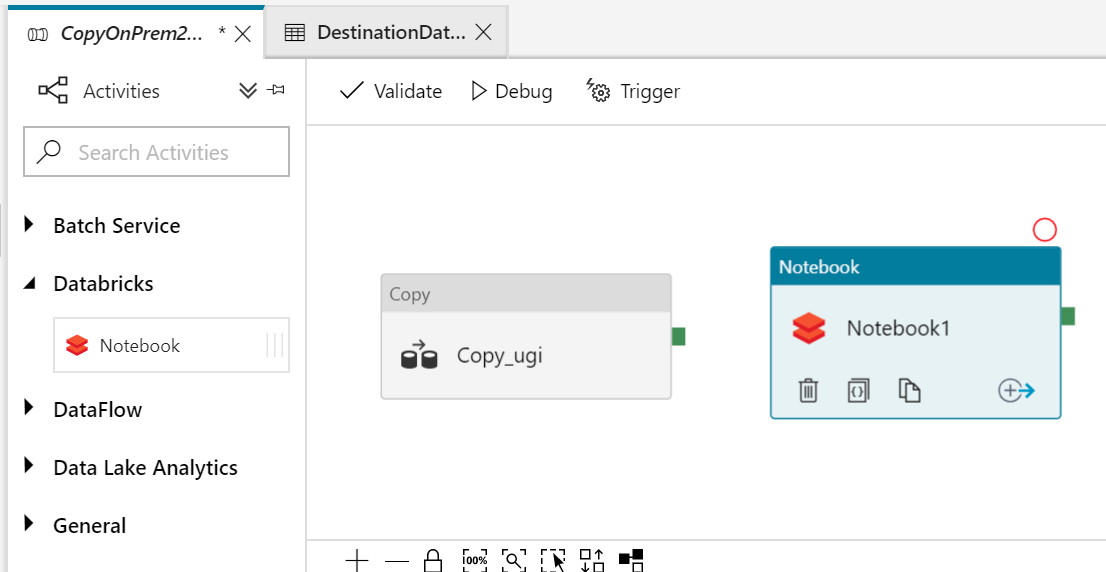
1. After saving the Copy settings, select Next on the Summary tab.
2. On the Deployment screen you will see a message that the deployment in is progress, and after a minute or two that the deployment completed. Select Edit Pipeline to close out of the wizard.

## Operationalize ML scoring with Azure Databricks and Data Factory

In this exercise, you will extend the Data Factory to operationalize the scoring of data using the previously created machine learning model within an Azure Databricks notebook.

Task 1: Create Azure Databricks Linked Service

1. Return to, or reopen, the Author & Monitor page for your Azure Data Factory in a web browser, navigate to the Author view, and select the pipeline.
2. Once there, expand Databricks under Activities.
3. Drag the Notebook activity onto the design surface to the side of the Copy activity.



1. Select the Notebook activity on the design surface to display tabs containing its properties and settings at the bottom of the screen. On the General tab, enter "BatchScore" into the Name field.
2. Select the Azure Databricks tab, and select + New next to the Databricks Linked service drop down. Here, you will configure a new linked service which will serve as the connection to your Databricks cluster.
3. On the New Linked Service dialog, enter the following:
   1. Name: enter a name, such as AzureDatabricks.
   2. Connect via integration runtime: Leave set to Default.
   3. Account selection method: Select From Azure subscription.
   4. Choose your Azure Subscription.
   5. Pick your Databricks workspace to populate the Domain automatically.
   6. Select cluster: choose Existing cluster.
4. Leave the form open and open your Azure Databricks workspace in another browser tab. You will retrieve the Access token and cluster id here.
5. In Azure Databricks, select the Account icon in the top corner of the window, then select User Settings.
6. Select Generate New Token under the Access Tokens tab. Enter ADF access for the comment and leave the lifetime at 90 days. Select Generate.
7. Copy the generated token.
8. Switch back to your Azure Data Factory screen and paste the generated token into the Access token field within the form.
9. Leave the form open and switch back to Azure Databricks. Select Clusters on the menu, then select your cluster in the list. Select the Tags tab and copy the ClusterId value.
10. Switch back to your Azure Data Factory screen and paste the ClusterId value into the Existing cluster id field. Select Finish.
11. Switch back to Azure Databricks. Select Workspace in the menu. Open notebook 04 Deploy for Batch Scoring. Examine the content but don't run any of the cells yet. You need to replace STORAGE-ACCOUNT-NAME with the name of the blob storage account you provisioned in the before-the-lab section.
12. Switch back to your Azure Data Factory screen. Browse to your 04 Deploy for Batch Score into the Notebook path field.
13. The final step is to connect the Copy activities with the Notebook activity. Select the small green box on the side of the copy activity, and drag the arrow onto the Notebook activity on the design surface. What this means is that the copy activity has to complete processing and generate its files in your storage account before the Notebook activity runs, ensuring the files required by the BatchScore notebook are in place at the time of execution. Select Publish All after making the connection.

Task 2: Trigger workflow

1. Switch back to Azure Data Factory. Select your pipeline if it is not already opened.
2. Select Trigger, then Trigger Now located above the pipeline design surface.
3. Enter 3/1/2017 into the windowStart parameter, then select Finish.
4. Select Monitor in the menu. You will be able to see your pipeline activity in progress as well as the status of past runs.

## Summarize data using Azure Databricks

In this exercise, you will prepare a summary of flight delay data using Spark SQL.

Task: Summarize delays by airport

1. Open your Azure Databricks workspace and open the final notebook called 05 explore Data
2. Execute each cell and follow the instructions in the notebook that explains each step.