Implementing Predictive Analytics with Hadoop in Azure HDInsight

Lab 3 – Building Streaming Machine Learning Models

Overview

In this lab, you will use Spark to create streaming machine learning models.

What You'll Need

To complete the labs, you will need the following:

- A web browser
- A Microsoft account
- A Microsoft Azure subscription
- A Windows, Linux, or Mac OS X computer
- The lab files for this course

Note: To set up the required environment for the lab, follow the instructions in the **Setup** document for this course. Specifically, you must have signed up for an Azure subscription and installed the Azure CLI tool.

Provisioning an HDInsight Spark Cluster

If you already have an HDInsight Spark cluster running from the previous lab, you can skip this exercise. Otherwise, follow the instructions below to provision a Spark cluster.

Note: The Microsoft Azure portal is continually improved in response to customer feedback. The steps in this exercise reflect the user interface of the Microsoft Azure portal at the time of writing, but may not match the latest design of the portal exactly.

Provision an HDInsight Cluster

1. In a web browser, navigate to http://portal.azure.com, and if prompted, sign in using the Microsoft account that is associated with your Azure subscription.
2. In the Microsoft Azure portal, in the Hub Menu, click **New**. Then in the **Data + Analytics** menu, click **HDInsight**.
3. In the **New HDInsight Cluster** blade, enter the following settings, and then click **Create**:
    - **Cluster Name**: *Enter a unique name (and make a note of it!)*
    - **Subscription**: *Select your Azure subscription*

- **Select Cluster Type**:
  - **Cluster Type**: Spark
  - **Cluster Operating System**: Linux
  - **Cluster Tier**: Standard
- **Credentials:**
  1. **Cluster Login Username**: *Enter a user name of your choice (and make a note of it!)*
  2. **Cluster Login Password**: *Enter and confirm a strong password (and make a note of it!)*
  - **SSH Username:** *Enter another user name of your choice (and make a note of it!)*
  - **SSH Authentication Type**: Password
  - **SSH Password:** *Enter and confirm a strong password (and make a note of it!)*
- **Data Source:**
  - **Create a new storage account**: *Enter a unique name consisting of lower-case letters and numbers only (and make a note of it!)*
  - **Choose Default Container**: *Enter the cluster name you specified previously*
  - **Location**: *Select any available region*
- **Node Pricing Tiers:**
  - **Number of Worker nodes**: 1
  - **Worker Nodes Pricing Tier**: *Use the default selection*
  - **Head Node Pricing Tier**: *Use the default selection*
- **Optional Configuration:** *None*
- **Resource Group**: *Create a new resource group with a unique name*
- **Pin to dashboard**: *Not selected*

4. In the Azure portal, view **Notifications** to verify that deployment has started. Then wait for the cluster to be deployed (this can take a long time – often 30 minutes or more. Go and get some coffee!)

Note: As soon as an HDInsight cluster is running, the credit in your Azure subscription will start to be charged. The free-trial subscription includes a credit limit of approximately $200 (or local equivalent) that you can spend over a period of 30 days, which is enough to complete the labs in this course as long as clusters are deleted when not in use. If you decide not to complete this lab, follow the instructions in the *Clean Up* procedure at the end of the lab to delete your cluster in order to avoid using your Azure credit unnecessarily.

View the HDInsight Cluster in the Azure Portal

1. In the Azure portal, browse to the Spark cluster you just created.
2. In the blade for your cluster, under **Quick Links**, click **Cluster Dashboards**.
3. In the **Cluster Dashboards** blade, note the dashboards that are available. These include a Jupyter Notebook that you will use later in this course.

Creating a Streaming Linear Regression Model

In this exercise, you will create a streaming linear regression machine learning model that predicts whether a person earns less than or greater than $50K.

SSH into Apache Spark

1. Use an SSH client (Putty on Windows or the native client on a mac or Linux) to connect to the cluster. The endpoint to SSH into your cluster will be:

***cluster_name*-ssh.azurewebsites.net**.

2. Log in using the SSH User credentials you specified when creating the cluster (***not*** the HTTP user credentials)
3. If you intend to use Scala at the bash prompt enter **spark-shell**. Alternatively, if you intend to use python enter **pyspark** at the prompt.

Start a Streaming Program

1. After the shell has been initialized, enter the following code to import the required namespaces:

Scala

```
import org.apache.spark.mllib.linalg.Vectors
import org.apache.spark.mllib.regression.LabeledPoint
import
org.apache.spark.mllib.classification.StreamingLogisticRegressionWithSG
D
import org.apache.spark.streaming.{Seconds, StreamingContext}
```

Python

```
from pyspark.mllib.linalg import Vectors
from pyspark.mllib.regression import LabeledPoint
from pyspark.mllib.classification import
StreamingLogisticRegressionWithSGD
from pyspark.streaming import *
```

2. When the code has finished running, enter the following code to process files in the training and test directory. If you are using Python, it's important to make sure that the indentation is exactly as shown below:

Scala

```
val ssc = new StreamingContext(sc, Seconds(30))
val training =
ssc.textFileStream("/training/lr").map(LabeledPoint.parse)
val testing = ssc.textFileStream("/testing/lr").map(LabeledPoint.parse)
```

Python

```
def parse(lp):
      label = float(lp[lp.find('(') + 1: lp.find(',')])
      vec = Vectors.dense(lp[lp.find('[') + 1:
lp.find(']')].split(','))
      return LabeledPoint(label, vec)

ssc = StreamingContext(sc, 30)
training = ssc.textFileStream("/training/lr").map(parse)
testing = ssc.textFileStream("/testing/lr").map(parse)
```

3. When the code has finished executing, enter the following code into the shell to train the Logistic Regression model on six features and continually look in the training and testing directories to ensure that new files are picked up:

Scala

```
val numFeatures = 6
val model = new
StreamingLogisticRegressionWithSGD().setInitialWeights(Vectors.zeros(nu
mFeatures))
```

Python

```
numFeatures = 6
model = StreamingLogisticRegressionWithSGD()
model.setInitialWeights([0.0, 0.0, 0.0, 0.0, 0.0, 0.0])
```

4. When the code has finished executing, enter the following code to allow the trained model to be updated by the new model files in the training directory and the test files to be used and picked up from the test directory, and then store the results in an output directory:

Scala

```
model.trainOn(training)
model.predictOnValues(testing.map(lp => (lp.label,
lp.features))).saveAsTextFiles("/testing/out")
```

Python

```
model.trainOn(training)
model.predictOnValues(testing.map(lambda lp : (lp.label,
lp.features))).saveAsTextFiles("/testing/out")
```

5. Lastly enter the following code at the prompt to start the streaming receiver and block to update the console for messages:

Scala

```
ssc.start()
ssc.awaitTermination()
```

Python

```
ssc.start()
ssc.awaitTermination()
```

6. In the output you should see an error suggesting that **/training/lr** doesn't exist. You are now going to create the training and test folder.

Upload Source Data to Azure Storage

1. Open a new command line window.
2. Enter the following command to switch the Azure CLI to resource manager mode.

```
azure config mode arm
```

Note: If a *command not found* error is displayed, ensure that you have followed the instructions in the setup guide to install the Azure CLI.

3. Enter the following command to log into your Azure subscription:

```
azure login
```

4. Follow the instructions that are displayed to browse to the Azure device login site and enter the authentication code provided. Then sign into your Azure subscription using your Microsoft account.
5. Enter the following command to view your Azure resources:

```
azure resource list
```

6. Verify that your HDInsight cluster and the related storage account are both listed. Note that the information provided includes the resource group name as well as the individual resource names. Note the resource group and storage account name
7. Enter a `dir` (Windows) or `ls` (Mac OS X or Linux) command to view the contents of the **Lab03** folder where you extracted the lab files for this course (for example, `dir c:\HDPredictLabs\Lab03` or `ls HDPredictLabs/Lab03`), and verify that this folder contains a file named **income-training1.txt**. This file contains LabeledPoints of US income data, including features. Also verify that the folder contains a file named **income-test.txt**.
8. Enter the following command on a single line to determine the connection string for your Azure storage account, specifying the storage account and resource group names you noted earlier:

   ```
   azure storage account connectionstring show account -g resource_group
   ```

9. Note the connection string, copying it to the clipboard if your command line tool supports it.
10. If you are working on a Windows client computer, enter the following command to set a system variable for the connection string:

    ```
    SET AZURE_STORAGE_CONNECTION_STRING=your_connection_string
    ```

    If you are using a Linux or Mac OS X client computer, enter the following command to set a system variable for the connection string (enter the connection string in quotation marks):

    ```
    export AZURE_STORAGE_CONNECTION_STRING="your_connection_string"
    ```

11. Enter the following commands on separate lines to upload the requisite training and test files to blob storage in the container used by your HDInsight cluster. Replace ***file*** with the local path to income-training1.txt and income-test.txt (for example c:\HDPredictLabs\Lab03\income-test.txt or HDPredictLabs/Lab03/income-test.txt) and replace ***container*** with the name of the storage container used by your cluster (which should be the same as the cluster name):

    ```
    azure storage blob upload file container training/lr/income-training1.txt
    ```

    ```
    azure storage blob upload file container testing/lr/income-test.txt
    ```

12. Wait for the files to be uploaded, and then observe the SSH console to note any increased activity in the Spark program.
13. In the command prompt window where you have been using the Azure CLI, enter the following to check for output in the **testing/out** folder:

    ```
    azure storage blob list container -p testing/out
    ```

14. Identify any of the blobs that have the file name **part-00000** and a length greater than 0, and then enter the following command to download it to a local folder, replacing ***container*** with your container name, ***testing-out-123/part-00000*** with the blob name, and ***folder*** with the path to the local folder):

    ```
    azure storage blob download container testing-out-123/part-00000 folder
    ```

15. Open the downloaded file in a text editor and verify that it contains output in the form of ***(label, prediction)*.**
16. Close the SSH window to end the session.

Clean Up

If you intend to continue to the next lab immediately, you can leave your cluster running and use it to complete the next lab. Otherwise, you should delete your cluster and the associated storage account. This ensures that you avoid being charged for cluster resources when you are not using them. If you are using a trial Azure subscription that includes a limited free credit value, deleting the cluster maximizes your credit and helps to prevent using it all before the free trial period has ended.

Delete the Resource Group for your HDInsight Cluster

1. If it is not already open in a tab in your web browser, browse to the new Azure portal at https://portal.azure.com.
2. In the Azure portal, view your **Resource groups** and select the resource group you created for your cluster. This resource group contains your cluster and the associated storage account.
3. In the blade for your resource group, click **Delete**. When prompted to confirm the deletion, enter the resource group name and click **Delete**.
4. Wait for a notification that your resource group has been deleted.