Implementing Predictive Analytics with Spark in Azure HDInsight

Lab 4 – Recommenders and Clustering

Overview

In this lab, you will use Spark to build a recommender and a K-Means clustering model.

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
- Azure Storage Explorer
- The lab files for this course
- A Spark 2.0 HDInsight cluster.

Note: If you have not already done so, set up the required environment for the lab by following the instructions in the Setup document for this course. Then follow the instructions in Lab 1 to provision an HDInsight cluster.

Create a Recommender using Collaborative Filtering

First, you will evaluate a classification model that predicts whether or not a flight will be late.

Upload Source Data to Azure Storage

In this lab, you will build a recommender based on data about movie ratings. Before you can do this, you must store the data files in the shared storage used by your cluster. The instructions here assume you will use Azure Storage Explorer to do this, but you can use any Azure Storage tool you prefer.

1. In the folder where you extracted the lab files for this course on your local computer, in the **data** folder, verify that the **ratings.csv** and **movies.csv** files exists. These files contain data that describes 5-star rating activity from MovieLens, a movie recommendation service. It was created by GroupLens, a research group in the Department of Computer Science and Engineering at the University of Minnesota, and is used here with permission.

Note: This dataset and other GroupLens data sets are publicly available for download at http://grouplens.org/datasets/.

For more information, see F. Maxwell Harper and Joseph A. Konstan. 2015. The MovieLens Datasets: History and Context. ACM Transactions on Interactive Intelligent Systems (TiiS) 5, 4, Article 19 (December 2015)

2. Start Azure Storage Explorer, and if you are not already signed in, sign into your Azure subscription.
3. Expand your storage account and the **Blob Containers** folder, and then double-click the blob container for your HDInsight cluster.
4. In the **Upload** drop-down list, click **Upload Files**. Then upload **ratings.csv** and **movies.csv** as block blobs to a folder named **data** in root of the container.

Create a Recommender

You will use a Jupyter Notebook to create and evaluate your recommendation model. You can choose to work with Python or Scala.

1. In the Azure portal, in the blade for your HDInsight cluster, under **Quick Links**, click **Cluster Dashboards**.
2. Click **Jupyter Notebook**, and if prompted, log in using the cluster login name you specified when provisioning your cluster (be sure you use login name for HTTP connections and not the SSH user name.)
3. Click **Upload**, and browse to the **Lab04** folder in the folder where you extracted the lab files. Then select either **Python Recommender.ipynb** or **Scala Recommender.ipynb**, depending on your preferred choice of language, and upload it.
4. Open the notebook you uploaded and then read the notes and run the code it contains to build and test a recommender model.

Creating a Clustering Model

Next, you will create a K-Means clustering model to segment customer data.

Upload Source Data to Azure Storage

In this exercise, you will build a clustering model based on data about customers. Before you can do this, you must store the data files in the shared storage used by your cluster.

1. In the folder where you extracted the lab files for this course on your local computer, in the **data** folder, verify that the **customers.csv** file exists. This file contains data that describes features of customers.
2. Start Azure Storage Explorer, and if you are not already signed in, sign into your Azure subscription.
3. Expand your storage account and the **Blob Containers** folder, and then double-click the blob container for your HDInsight cluster.
4. In the **Upload** drop-down list, click **Upload Files**. Then upload **customers.csv** as a block blob to a folder named **data** in root of the container.

Create a Clustering Model

You will use a Jupyter Notebook to create your clustering model. You can choose to work with Python or Scala.

1. From the **Lab04** folder in the folder where you extracted the lab files, upload **Python Clustering.ipynb** or **Scala Clustering.ipynb**, depending on your preferred choice of language, to the Jupiter Dashboard for your cluster.
2. Open the notebook you uploaded and then read the notes and run the code it contains to build a clustering model.

Clean Up

Follow the steps below to delete your cluster and avoid being charged for cluster resources when you are not using them.

Delete the Resource Group

1. Close the browser tab containing the Jupyter Notebooks dashboard if it is open.
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.
5. Close the browser.