Build a data mining app using Java, Weka, and the dashDB service

Add a warehouse database and analytics tools to your app on IBM Bluemix

Krunal M. Vora
Software Engineer
IBM

Karandeep Singh Chawla
Software Engineer
IBM

The dashDB (formerly known as Analytics Warehouse and BLU Acceleration) service provides data warehousing and analytics as a service on IBM Bluemix. Developers can develop and deploy a heavy-duty analytic application using blazing-fast IBM BLU database technology offered in the cloud. Learn how to develop a data mining application using the Weka statistical analysis tool and leveraging the IBM BLU columnar database.

As data scientists for the customer analytic group in our wireless service provider company, we want to leverage customer data to predict customer churn. Customer retention is a critical challenge for the telecommunications industry, where annual churn rates can be as high as 40 percent. If we can predict which customers are in danger of turnover, our company can take action to retain them before they take their business elsewhere. Even a small reduction in churn can have a significant impact on our bottom line.

We decided to build a quick web application we can enhance over time. Our app uses the code for a classification algorithm we developed in the Java™ language using Weka, an open source machine-learning tool. In Bluemix, we can deploy our Java application and take advantage of the dashDB service (formerly known as Analytics Warehouse service and BLU Acceleration service) to perform analysis on our customer data. This service provides simplicity and performance, as well as enterprise scale if we decide to grow our model or enhance our app to perform additional types
of analysis on our data. Finally, we chose Twitter Bootstrap as the web development framework because it offers the flexibility of a mobile-first web interface and can be easily adapted to the myriad devices and browsers our analysts use.

Learn how you can build a similar application in Bluemix. We assume that you have the necessary code for your application, and provide our application code and data as a sample to help you get started.

Run the app
Get the code

What you will need to build a similar app

• Familiarity with Java application development
• Familiarity with a modern front-end framework, such as Twitter Bootstrap
• Knowledge of a statistical analysis tool, such as Weka or R
• A Bluemix account


Step 1. Create the application in Bluemix

Log in to your Bluemix account (or sign up for a free trial).

On the dashboard page, click Add an application.

In this example, you will create a Java application. Under Runtimes, select .java liberty (Liberty for Java).

In the pop-up window, click CREATE APP.

In the next pop-up window, fill in the app name and host, then click CREATE.
Bluemix creates the app in your workspace and starts the Java runtime. You will know when the app successfully starts by the confirmation displayed on the dashboard.

Step 2. Create the **dashDB service**

Select the app you created from the dashboard to go to its overview page.

Click **Add new service** in the Services section of that page.
Select **dashDB** as the service to add.

A pop-up window will display with more information about the service. Click **ADD TO APPLICATION** and **CREATE** on the subsequent pop-up window.

**Step 3. Explore the dashDB service (optional)**

The service provides several data analysis tools from its web console, including loading and querying data, data analysis using R or Excel®, reporting using Cognos, and industry models that help you with common industry specific use cases. It’s worthwhile to explore this impressive set of available tools for future projects.

On the app overview page, select the **dashDB** service.
On the following page, click **Launch the console**.

A new window will open with the web console. You can do many things in here, including uploading data files into your database and analyzing your data with R.
Step 4. Upload your data to **dashDB** (optional)

Our sample data set is already available in the dashDB. However, you can use your own data. To upload data:

1. In the dashDB web console, click the **Manage** tab, then select **Load Data**.
2. We will load data from a CSV file. On the **Quick load** screen, select the file for loading. Keep all the default settings and click **Load File**.

   ![Quick load screenshot]

   * Load data into a BLU Acceleration for Cloud database table

3. After you see the preview of the table, click **Next**.
4. On the **Choose the target** screen, select **Create a new table and load**. Click **Next**.
5. Change the **Table name** to **churntrainingset**. You can choose another table name but remember to update the code. For the **churn** column, change the **Data type** to **VARCHAR**. All other columns can keep the selected defaults. Click **Finish**.
6. You should see the success message. The data should be loaded.

Step 5. Download the code

If you haven't already done so, get the code.

Select **EDIT CODE**. After you log in, you will see the code.

Click **File > Export > Zip** to download the code to your machine.

Step 6. Understand the code

The sample application consists of these components:

- The FileLocationContextListener creates a folder for the file upload on the server.
- If the user selects the database to upload the training set for the model, the entered details are used to load data into an Instances object as TrainingSet. This TrainingSet is then used to create the NaiveBayes model. Alternatively, the default database table is used to create the model.
• The user can upload a CSV file as a Testing set. The file is uploaded into the folder created earlier on the server.
• Weka works with Attribute-Relation File Format (ARFF) files as a basic file format, including the attributes and the dataset it requires. The CSV2ARFF.java is an independent utility that converts the CSV file to ARFF file stored in the same folder on the server.
• The ARFF file is then loaded into an Instances object as TestingSet.
• For all the instances in the TestingSet, the NaiveBayes model is used to classify the output into Churn or Not Churn classes.
• The corresponding output is then displayed on the user interface.

Step 7. Generate a WAR file
To push the code to Bluemix, you will need to generate a WAR file. We can easily do this with Eclipse. A WAR file is already included in case you are unable to generate one.

Select File > Import. In the dialog window, select Existing Projects into Workspace, then select Next.

In the next dialog window, browse for the files you downloaded.
Keeping all the defaults selected is fine. Select **Finish**. The project has now been added to your Eclipse Client.

To export as a WAR file, right click on the project in the Project Explorer. Select **Export > War File**. Save the WAR file into a directory by itself.

![WAR Export](image)

**Step 8. Deploy the application**

Open a terminal and move into the directory of the WAR file. It is best to have the WAR file in its own directory.

Run the `cf push` command. Provide the application name, memory needed, instances, and path to the WAR file. For this application, let's provide 512 MB of memory and one instance: `cf push bludemo -m 512m -p BLUDemo.war`.

As the application uploads, there will be details indicating what is happening. After about a minute and a half, the application should be live.

If you make changes to the application, repeat this process. Run the same command after you have generated a new WAR file to push to Bluemix.

**Alternative steps: Deploy the application**

Instead of following most of the preceding steps, you can create the service and deploy the application.

After you have the code in your own workspace (Step 5), modify the file named `manifest.yml`.

Modify `name` and `host` to the name of the application and host. These should be the same value. The file should automatically be saved.
Click **Deploy**, and DevOps Services will attempt to deploy the application based upon the manifest.yml file. DevOps Services will ask for credentials when deploying. Complete Step 4 to upload the training data. After that, the demo application will work.

**Conclusion**

Now you know how **dashDB** provides data warehousing and analytics as a service on Bluemix platform and how you can develop and deploy a heavy-duty analytic application using IBM database technology in the cloud. Here's to faster, easier data mining in the cloud.

**Acknowledgment**

Many thanks to Alexandria Burkleaux for her review of this article.

dashDBhttps://ace.ng.bluemix.net/#/store/serviceOfferingGuid=92c8b899-8163-4cbc-a934-0f930cd69918&fromCatalog=trueadds a warehouse database and various analytics tools to your application.

**RELATED TOPICS:**  Cloud computing  Java technology  dashDB  Analytics Warehouse

© Copyright IBM Corporation 2014, 2015
**Trademarks**  
(www.ibm.com/developerworks/ibm/trademarks/)