



Demo Setup for Calling a
Spark Machine Learning Model
With IBM Db2 Big SQL

IBM NA Big Data Science Technical Team

December 2017

Contents

INTRO	DEMO SETUP FOR CALLING A SPARK MACHINE LEARNING MODEL WITH IBM DB2 BIG SQL.....	3
PART 1	PROVISION CLOUD CONCIERGE INSTANCES.....	3
PART 2	UPLOAD DATA INTO HDFS AND CREATE BIG SQL TABLES.....	4
PART 3	TRAIN AND SAVE A SPARK MACHINE LEARNING MODEL	7
PART 4	INSTALLING A PTF AND STARTING THE BIG SQL SPARK GATEWAY.....	9
PART 5	BIG SQL CALLING SPARK MACHINE LEARNING MODEL.....	10

Intro

Demo Setup for Calling a Spark Machine Learning Model with IBM Db2 Big SQL

IBM Db2 Big SQL (Big SQL) is a SQL engine on Hadoop that has been making strides with the fast-evolving open source ecosystem. The core capabilities of Big SQL focusses on federation, SQL compatibility, scalability, performance, and of course enterprise security, making it a desirable query engine to seek insights from disparate data sources including Hadoop.

In addition, Big SQL provides tight integration with Apache Spark featuring efficient synthesis between Spark executors and Big SQL workers.

This tutorial will focus on the steps to demonstrate Big SQL calling a Spark machine learning model. This will be using the Cloud Concierge template for the Hortonworks Data Platform provided by Frank Ketelaars.

Part 1 Provision Cloud Concierge Instances

This part of the tutorial will provision two Cloud Concierge instances. One will be the Hortonworks Data Platform (HDP) with Big SQL instance and the other will be a base CentOS instance that will be used as a ‘jump host’. This jump host will be allow us to ssh into the HDP image since the ssh port 22 for HDP is closed on the public interface.

- Open your browser and navigate to <http://demo.ibmcloud.com> to access IBM Cloud Concierge. Once logged in, click on ‘Provision’ from the top tab choices and then select ‘Hortonworks’ from the ‘Product’ pull down. Please pick the ‘HDP with Big SQL and AS’ template. Select the ‘Requirement Type’ that applies (in this example, ‘Testing’) and select a ‘Provisioning Location’ (in this example ‘Dallas 10’). Please remember this ‘Provisioning Location’ for the next step. Now click on the ‘Provision’ button at the bottom.

The screenshot shows the IBM Cloud Concierge interface. At the top, there's a navigation bar with tabs like 'Showcase', 'Provision' (which is highlighted), 'MyCloud', 'Opportunities', 'Support', 'Feedback', and user info. Below the navigation is a search bar with 'Product: Hortonworks' and dropdown filters for 'All Templates', 'Template Clusters', and 'My Templates'. A table lists available templates: 'Hortonworks,HDP' (version V1R5, updated Nov 23, 2017, expires Jun 1, 2018, SoftLayer). A green bar highlights the 'Big Data' section with 'HDP with Big SQL and AS' selected. The main form area contains fields for 'Instance/Archive Name' (set to 'HDP with Big SQL and AS'), 'Number of instances' (set to 1), 'Requirement Type' (radio buttons for 'Demo', 'POC', 'Customer Workshop', 'Enablement', 'Testing' - which is selected), 'Private Network (Requires IBM Cloud VPN)', 'Options(coming soon)', 'Provisioning Location' (set to 'Dallas 10'), 'Opportunity#', 'Customer', 'Sales Contact', 'Enablement Event Code', and 'Provision Until (GMT -6)'. A note at the bottom says 'Reminder: It is an IBM BCG violation to place sensitive personal or customer data on public servers provisioned from this site.' At the bottom right, there's a note 'Denotes internal link'. At the very bottom, there are buttons for 'Provision' (which is highlighted with a red circle), 'Delete', 'Rename Archive', and 'Share Archive'.

- b) Please give this a few moments and the following message will appear.

- c) While that is being provisioned, let's also provision the jump host. Select 'Base Image' from the 'Product' pull down. Select the 'Base Image – CentOS Latest 64' template. Select the 'Requirement Type' that applies (in this example 'Testing'). Select the same 'Provisioning Location' that was previously chosen for HDP (in this example 'Dallas 10'). Now click on the 'Provision' button.

- d) Once again, please wait a few moments and the following acknowledgement will appear.

- e) As indicated, give this time to provision. When it is provisioned, you will see the following under the 'MyCloud' tab choices on the top of the page. Please note and remember the 'Primary IP' (public), the 'Private IP', the 'LoginUser' and the 'LoginPassword' for both instances.

Part 2 Upload Data into HDFS and Create Big SQL Tables

This part of the tutorial will upload the data to HDP HDFS and create Big SQL tables.

- a) ssh login to the HDP instance using the jump host. To do this, the following command will first ssh login into the jump host followed by an ssh login to HDP. Please accept the authenticity fingerprints and enter the passwords when prompted.

```
ssh ibm@<your_base_centos_public_ip> -p 2222 -t ssh ibm@<your_HDP_private_ip>
```

In this example, ‘`ssh ibm@169.46.33.182 -p 2222 -t ssh ibm@10.93.89.68`’ was used. You are now logged into HDP as user ‘ibm’.

```
[dkikuchi@MacBook-Pro ~]$ ssh ibm@169.46.33.182 -p 2222 -t ssh ibm@10.93.89.68
The authenticity of host '[169.46.33.182]:2222' ([169.46.33.182]:2222) can't be established.
ECDSA key fingerprint is SHA256:VUjRR//lxuengPyzta@Q3twVri/OggAE43d4Q3weBwk.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '[169.46.33.182]:2222' (ECDSA) to the list of known hosts.
ibm@169.46.33.182's password:
The authenticity of host '10.93.89.68 (10.93.89.68)' can't be established.
ECDSA key fingerprint is SHA256:yqe+f+MIpIJD8gij0sqAwRE7kPiTFPsJ8Aq+OHIIeg.
ECDSA key fingerprint is MD5:9b:36:a6:71:50:22:b0:74:e7:a9:b4:32:8a:6c:e0:19.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.93.89.68' (ECDSA) to the list of known hosts.
ibm@10.93.89.68's password:
Last login: Tue Dec 19 08:40:34 2017 from 10.3.220.74
[ibm@hdp ~]$
```

- b) Switch to the user ‘bigsq1’ by issuing the following command.

```
su - bigsql
```

Use the same password as before.

Issue the following commands to retrieve the 2 data files.

```
wget https://raw.githubusercontent.com/dxkikuchi/BigSQLMachineLearning/master/GoSales_Tx_20.csv
wget https://raw.githubusercontent.com/dxkikuchi/BigSQLMachineLearning/master/GoSales_Tx_80.csv
```

```
[ibm@hdp ~]$ su - bigsql
Password:
Last login: Wed Dec 20 15:02:36 CST 2017
[bigsq1@hdp ~]$ wget https://raw.githubusercontent.com/dxkikuchi/BigSQLMachineLearning/master/GoSales_Tx_20.csv
--2017-12-20 15:05:51-- https://raw.githubusercontent.com/dxkikuchi/BigSQLMachineLearning/master/GoSales_Tx_20.csv
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 151.101.48.133
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|151.101.48.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 501580 (499K) [text/plain]
Saving to: "GoSales_Tx_20.csv"

100%[=====] 501,580 --.-K/s in 0.02s
2017-12-20 15:05:52 (26.7 MB/s) - 'GoSales_Tx_20.csv' saved [501580/501580]

[bigsq1@hdp ~]$ wget https://raw.githubusercontent.com/dxkikuchi/BigSQLMachineLearning/master/GoSales_Tx_80.csv
--2017-12-20 15:05:54-- https://raw.githubusercontent.com/dxkikuchi/BigSQLMachineLearning/master/GoSales_Tx_80.csv
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 151.101.48.133
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|151.101.48.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 2009012 (1.9M) [text/plain]
Saving to: "GoSales_Tx_80.csv"

100%[=====] 2,009,012 --.-K/s in 0.05s
2017-12-20 15:05:54 (35.3 MB/s) - 'GoSales_Tx_80.csv' saved [2009012/2009012]
[bigsq1@hdp ~]$
```

- c) Create folders on HDP HDFS for each of these files by issuing the following commands.

```
hdfs dfs -mkdir GoSales20
```

```
hdfs dfs -mkdir GoSales80
```

And copy the data files to these folders by issuing the following commands.

```
hdfs dfs -put GoSales_Tx_20.csv GoSales20
```

```
hdfs dfs -put GoSales_Tx_80.csv GoSales80
```

```
[bigsq1@hdp ~]$ hdfs dfs -mkdir GoSales20
[bigsq1@hdp ~]$ hdfs dfs -mkdir GoSales80
[bigsq1@hdp ~]$ hdfs dfs -put GoSales_Tx_20.csv GoSales20
[bigsq1@hdp ~]$ hdfs dfs -put GoSales_Tx_80.csv GoSales80
[bigsq1@hdp ~]$ hdfs dfs -ls GoSales20
Found 1 items
-rw-r--r-- 1 bigsql hdfs 501580 2017-12-20 15:13 GoSales20/GoSales_Tx_20.csv
[bigsq1@hdp ~]$ hdfs dfs -ls GoSales80
Found 1 items
-rw-r--r-- 1 bigsql hdfs 2009012 2017-12-20 15:13 GoSales80/GoSales_Tx_80.csv
[bigsq1@hdp ~]$
```

- d) Start 'jsqsh' which is the Java SQL Shell command interface by issuing the command

```
jsqsh bigsql
```

Note that HDP and Big SQL may not be entirely initialized. To see the current status, issue '[bigsql status](#)' and if needed, please wait until Big SQL is available and running.

- e) Create the 'gosales20' table by issuing the following command in the 'jsqsh' shell.

```
create external hadoop table gosales20 (
  PRODUCT_LINE string,
  GENDER string,
  AGE integer,
  MARITAL_STATUS string,
  PROFESSION string
)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
stored as textfile location '/user/bigsql/GoSales20/';
```

```
[bigsql@hdp ~]$ jsqsh bigsql
Welcome to JSQSh 4.10.4.6
Type "help" for help topics. Using Jline.
[hdp.demos.demoibm.com][bigsql] 1> create external hadoop table gosales20 (
[hdp.demos.demoibm.com][bigsql] 2>   PRODUCT_LINE string,
[hdp.demos.demoibm.com][bigsql] 3>   GENDER string,
[hdp.demos.demoibm.com][bigsql] 4>   AGE integer,
[hdp.demos.demoibm.com][bigsql] 5>   MARITAL_STATUS string,
[hdp.demos.demoibm.com][bigsql] 6>   PROFESSION string
[hdp.demos.demoibm.com][bigsql] 7> )
[hdp.demos.demoibm.com][bigsql] 8> ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
[hdp.demos.demoibm.com][bigsql] 9> stored as textfile location '/user/bigsql/GoSales20/';
0 rows affected (total: 1.088s)
[hdp.demos.demoibm.com][bigsql] 1> \tables
+-----+-----+-----+
| TABLE_SCHEMA | TABLE_NAME | TABLE_TYPE |
+-----+-----+-----+
| BIGSQL       | GOSALES20 | TABLE      |
+-----+-----+-----+
[hdp.demos.demoibm.com][bigsql] 1> select * from gosales20 limit 3;
+-----+-----+-----+-----+-----+
| PRODUCT_LINE | GENDER | AGE | MARITAL_STATUS | PROFESSION |
+-----+-----+-----+-----+-----+
| Golf Equipment | F | 45 | Married | Other
| Golf Equipment | F | 45 | Married | Other
+-----+-----+-----+-----+-----+
3 rows in results(first row: 1.932s; total: 1.934s)
[hdp.demos.demoibm.com][bigsql] 1>
```

- f) Do the same for the 'gosales80' table.

```
create external hadoop table gosales80 (
  PRODUCT_LINE string,
  GENDER string,
  AGE integer,
  MARITAL_STATUS string,
  PROFESSION string
)
ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
stored as textfile location '/user/bigsql/GoSales80/';
```

```
[hdp.demos.demoibm.com][bigsql] 1> create external hadoop table gosales80 (
[hdp.demos.demoibm.com][bigsql] 2>   PRODUCT_LINE string,
[hdp.demos.demoibm.com][bigsql] 3>   GENDER string,
[hdp.demos.demoibm.com][bigsql] 4>   AGE integer,
[hdp.demos.demoibm.com][bigsql] 5>   MARITAL_STATUS string,
[hdp.demos.demoibm.com][bigsql] 6>   PROFESSION string
[hdp.demos.demoibm.com][bigsql] 7> )
[hdp.demos.demoibm.com][bigsql] 8> ROW FORMAT DELIMITED FIELDS TERMINATED BY ','
[hdp.demos.demoibm.com][bigsql] 9> stored as textfile location '/user/bigsql/GoSales80/';
0 rows affected (total: 0.647s)
[hdp.demos.demoibm.com][bigsql] 1> \tables
+-----+-----+-----+
| TABLE_SCHEMA | TABLE_NAME | TABLE_TYPE |
+-----+-----+-----+
| BIGSQL       | GOSALES20 | TABLE      |
| BIGSQL       | GOSALESB0 | TABLE      |
+-----+-----+-----+
[hdp.demos.demoibm.com][bigsql] 1> select * from gosales80 limit 3;
+-----+-----+-----+-----+-----+
| PRODUCT_LINE | GENDER | AGE | MARITAL_STATUS | PROFESSION |
+-----+-----+-----+-----+-----+
| Personal Accessories | M | 27 | Single | Professional
| Personal Accessories | F | 39 | Married | Other
+-----+-----+-----+-----+-----+
3 rows in results(first row: 0.646s; total: 0.647s)
[hdp.demos.demoibm.com][bigsql] 1>
```

- g) The data has been uploaded, inserted into HDP HDFS and Big SQL tables have been created.

Part 3 Train and Save a Spark Machine Learning Model

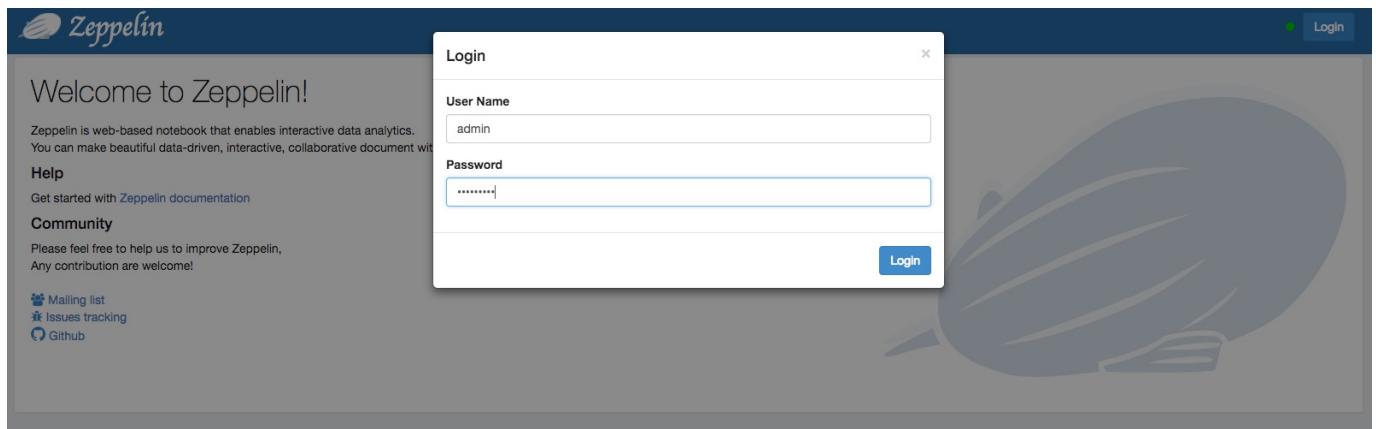
This part of the tutorial will use a Zeppelin notebook to train and save a Spark Machine Learning model using the data in Part 2.

- a) Login to Zeppelin on HDP by going to a browser and entering the following url.

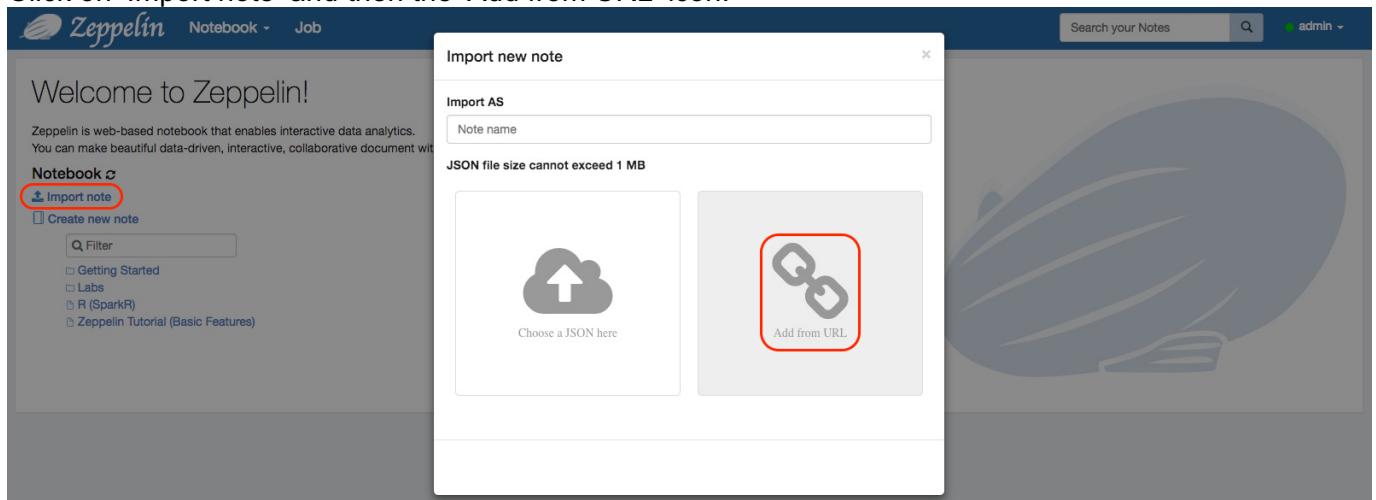
https://<your_HDP_public_ip>:9995

In this example, <https://169.60.28.228:9995> was used.

Please enter a ‘User Name’ of ‘admin’ and the same password as before.



- b) Click on ‘Import note’ and then the ‘Add from URL’ icon.



- c) Enter the following for ‘URL’ and click on ‘Import Note’.

<https://raw.githubusercontent.com/dxkikuchi/BigSQLMachineLearning/master/SparkMLDecisionTreeModel.json>

The screenshot shows the Zeppelin interface with a modal dialog titled "Import new note". In the "URL" field, the URL <https://raw.githubusercontent.com/dxkikuchi/BigSQLMachineLearning/master/SparkMLDecisionTreeModel.json> is entered. The "Import Note" button at the bottom right of the dialog is circled in red.

- d) Click on the ‘SparkMLDecisionTreeModel’ notebook.

The screenshot shows the Zeppelin interface with the "SparkMLDecisionTreeModel" notebook highlighted in the sidebar under the "Create new note" section, indicated by a red circle.

- e) Step through the notebook.

The screenshot shows the "SparkMLDecisionTreeModel" notebook in the Zeppelin interface. The title "A Spark Machine Learning Model Using Decision Trees" is displayed. Below it, a note states: "This notebook will retrieve training and test data that resides on HDFS which will be used to train and save a machine learning model. This model will be used to predict a product line category that a customer may be interested in based on that customer's gender, age, marital status and profession. This model can be called using IBM Db2 Big SQL to deliver predictions. Please step through the notebook to see how this model is provided." The notebook interface includes sections for Scala, Java, Python, and R, with icons for each. A diagram illustrates the Spark architecture layers: Spark Core, DataFrame API, and various components like Spark SQL, Spark Streaming, MLlib, GraphX, and Packages. Below this, a "Data Source" section lists various data formats and databases (Tez, Cassandra, Neo4j, MongoDB, MySQL, Elasticsearch). The first code cell, titled "Retrieve training and test data from HDFS", contains the following Scala code:

```

SparkH2 // CODE CELL 1 //
val train = spark.read.option("inferSchema", "true").option("header", "true").csv("/user/bigsql/GoSales2B/GoSales_Tx_80.csv")
val test = spark.read.option("inferSchema", "true").option("header", "true").csv("/user/bigsql/GoSales2B/GoSales_Tx_20.csv")
  
```

- f) This notebook will retrieve the training and test data that resides on HDP HDFS which will be used to train and save a machine learning model. This model will be used to predict a product line category that a customer may be interested in base on that customer's gender, age, marital status and profession. This model has been saved in HDP HDFS and will be called upon by IBM Db2 Big SQL to deliver predictions.

Part 4 Installing a PTF and Starting the Big SQL Spark Gateway

This part of the tutorial will install a PTF (Polymorphic Table Function) and start the Big SQL Spark Gateway. More detailed information on this can be found at

https://www.ibm.com/support/knowledgecenter/SSCRJT_5.0.1/com.ibm.swg.im.bigsq.dev.doc/doc/bigaspark_int.html

- a) From the HDP operating system prompt, retrieve the PTF using the following command.

`wget https://raw.githubusercontent.com/dxkikuchi/BigSQLMachineLearning/master/QueryMLModelPTF.jar`

```
[bigsql@hdp ~]$ wget https://raw.githubusercontent.com/dxkikuchi/BigSQLMachineLearning/master/QueryMLModelPTF.jar
--2017-12-20 16:05:35-- https://raw.githubusercontent.com/dxkikuchi/BigSQLMachineLearning/master/QueryMLModelPTF.jar
Resolving raw.githubusercontent.com (raw.githubusercontent.com)... 151.101.48.133
Connecting to raw.githubusercontent.com (raw.githubusercontent.com)|151.101.48.133|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 26224 (26k) [application/octet-stream]
Saving to: 'QueryMLModelPTF.jar'

100%[=====] 26,224 --.-K/s in 0.002s

2017-12-20 16:05:36 (12.4 MB/s) - 'QueryMLModelPTF.jar' saved [26224/26224]

[bigsql@hdp ~]$
```

- b) Access jsqsh using the following command.

`jsqsh bigsql`

- c) Issue the following command in the jsqsh shell to install the PTF.

`\install-jar --path=/home/bigsq/QueryMLModelPTF.jar --id myptfs`

```
[bigsql@hdp ~]$ jsqsh bigsql
Welcome to JSqsh 4.10.4.6
Type "help" for help topics. Using JLine.
[hdp.demos.demoibm.com][bigsql] 1> \install-jar --path=/home/bigsq/QueryMLModelPTF.jar --id myptfs
[hdp.demos.demoibm.com][bigsql] 1>
```

- d) Exit out of jsqsh. Copy the bigsql-spark.conf template and add the PTF jar by issuing the following commands.

`cp /usr/ibmpacks/bigsq/5.0.2.0/bigsq/conf/templates/bigsql-spark.conf /usr/ibmpacks/bigsq/5.0.2.0/bigsq/conf/bigsql-spark.conf
vi /usr/ibmpacks/bigsq/5.0.2.0/bigsq/conf/bigsql-spark.conf`

Add `/home/bigsq/QueryMLModelPTF.jar` to the 'spark.driver.extraClassPath' and 'spark.executor.extraClassPath'. The file should look like the following. Save the file.

```
spark.executor.instances 2
spark.executor.memory 2g
spark.driver.memory 2g
spark.executor.cores 4
spark.driver.extraClassPath /usr/ibmpacks/current/bigsq/bigsql/userlib/ptf-examples.jar;/home/bigsq/QueryMLModelPTF.jar
spark.executor.extraClassPath /usr/ibmpacks/current/bigsq/bigsql/userlib/ptf-examples.jar;/home/bigsq/QueryMLModelPTF.jar
bigsql.spark.sql.regen
bigsql.spark.sql.tableoutput.engine bigsql
bigsql.spark.check.url.access true
bigsql.spark.default.package com.ibm.biginights.bigsql.examples
bigsql.spark.string.size 1024
```

- e) Stop and restart bigsql and bigsql spark.

```
bigsq1 stop -spark
bigsq1 stop
bigsq1 start
bigsq1 start -spark
```

```
[bigsq1@hdp ~]$ bigsq1 stop -spark
Stopping Big SQL Spark gateway : OK
[bigsq1@hdp ~]$ bigsq1 stop
Stopping Big SQL      : OK
Stopping Big SQL Scheduler : OK
[bigsq1@hdp ~]$ bigsq1 start
Global config update   : OK
Starting Big SQL Scheduler : OK
Starting Big SQL      : OK
[bigsq1@hdp ~]$ bigsq1 start -spark
Starting Big SQL Spark gateway : OK
[bigsq1@hdp ~]$
```

- f) The PTF has been installed and the Big SQL Spark Gateway has been started.

Part 5 Big SQL Calling Spark Machine Learning Model

This part of the tutorial will use Big SQL to call the model and provide predictions using 5 rows of the gosales20 Big SQL table.

In the jsqsh shell, enter the following command.

```
SELECT model.* FROM TABLE(SYSHADOOP.EXECSPARK(
  class => 'com.ibm.biginights.bigsq1.examples.QueryMLModel',
  model_uri => CAST('/user/zeppelin/CrossValidatedDecisionTree.sparkmodel' as SYSHADOOP.URI),
  feature_names => 'PRODUCT_LINE,GENDER,AGE,MARITAL_STATUS,PROFESSION',
  feature_values => TABLE(SELECT * FROM gosales20 where age is not null) as in,
  label_name => 'predictedLabel',
  label_type => 'string')) AS model limit 5;
```

```
[bigsq1@hdp conf]$ jsqsh bigsq1
Welcome to JSQSh 4.10.4.6
Type "help" for help topics. Using JLine.
[bigsq1@hdp conf]$ SELECT model.* FROM TABLE(SYSHADOOP.EXECSPARK(
  class => 'com.ibm.biginights.bigsq1.examples.QueryMLModel',
  model_uri => CAST('/user/zeppelin/CrossValidatedDecisionTree.sparkmodel' as SYSHADOOP.URI),
  feature_names => 'PRODUCT_LINE,GENDER,AGE,MARITAL_STATUS,PROFESSION',
  feature_values => TABLE(SELECT * FROM gosales20 where age is not null) as in,
  label_name => 'predictedLabel',
  label_type => 'string')) AS model limit 5;
+-----+-----+-----+-----+-----+-----+-----+
| PRODUCT_LINE | GENDER | AGE | MARITAL_STATUS | PROFESSION | PREDICTEDLABEL | PROBABILITY |
+-----+-----+-----+-----+-----+-----+-----+
| Golf Equipment | F | 45 | Married | Other | Camping Equipment | [0.35398230088495675, 0.3392330383480826, 0.02654867256637168, 0.20648967551622419, 0.07374631268436578]
| Golf Equipment | F | 45 | Married | Other | Camping Equipment | [0.35398230088495675, 0.3392330383480826, 0.02654867256637168, 0.20648967551622419, 0.07374631268436578]
| Personal Accessories | F | 32 | Single | Professional | Personal Accessories | [0.027237354085603113, 0.9455252918287937, 0.019455252918287938, 0.007782101167315175, 0.0]
| Personal Accessories | F | 33 | Single | Other | Personal Accessories | [0.031413612565446025, 0.9319371727746951, 0.01578600282722512, 0.005235602894240838, 0.015706886282722512]
| Mountaineering Equipment | M | 23 | Single | Professional | Camping Equipment | [0.5662164886111671, 0.1505744809514211, 0.2541826244708728, 0.01431163071961298, 0.014714775246926022]
+-----+-----+-----+-----+-----+-----+
5 rows in results(first row: 4.099s; total: 4.099s)
[bigsq1@hdp conf]$
```

Congratulations, you have completed this exercise.



Great work and congratulations, you have completed this exercise.



Screen captures in this document may vary slightly from yours.

NOTES



© Copyright IBM Corporation 2016. Author: Daniel Kikuchi

The information contained in these materials is provided for informational purposes only, and is provided AS IS without warranty of any kind, express or implied. IBM shall not be responsible for any damages arising out of the use of, or otherwise related to, these materials. Nothing contained in these materials is intended to, nor shall have the effect of, creating any warranties or representations from IBM or its suppliers or licensors, or altering the terms and conditions of the applicable license agreement governing the use of IBM software. References in these materials to IBM products, programs, or services do not imply that they will be available in all countries in which IBM operates. This information is based on current IBM product plans and strategy, which are subject to change by IBM without notice. Product release dates and/or capabilities referenced in these materials may change at any time at IBM's sole discretion based on market opportunities or other factors, and are not intended to be a commitment to future product or feature availability in any way.

IBM, the IBM logo and ibm.com are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the Web at "Copyright and trademark information" at www.ibm.com/legal/copytrade.shtml.