



IBM Predictive Analytics for Bluemix - Sample1 Deployment

Table of Contents

1 Document Purpose	3
2 Prerequisites	4
2.1 Bluemix	4
2.2 Cloud Foundry	4
2.3 Node.js	5
2.4 Downloading Sample 1	6
3 IBM SPSS Modeler	7
4 Bluemix Application and Service	9
4.1 Creating an application	9
4.2 Creating the service	11
5 Customizing files	14
5.1 \app\manifest.yml	14
5.2 \app\app.js	14
5.3 \app\public\js\app.js	14
5.4 \app\public\js\srv.js	15
6 Porting source code	16
6.1 Node Package Manager	16
7 Uploading SPSS Models	18
8 Real-time scoring the sample	20

1 Document purpose

This document outlines how to configure and run IBM Predictive Analytics Service Sample1 (<https://github.com/pmsservice/predictive-modeling-samples>).

- This document is meant to compliment the current documentation associated with Sample1.
- This document interfaces with an IBM SPSS Modeler stream, but the focus is on Bluemix. It is assumed that the reader has a basic understanding of predictive analytics used in IBM SPSS Modeler.
- While there are many supported runtimes in Bluemix (Liberty for Java, Python, Ruby, ASP.NET, Swift, Tomcat, etc.), this document will use SDK for Node.js.
- Other operating systems are supported but this example uses Microsoft Windows.

2 Prerequisites

2.1 Bluemix

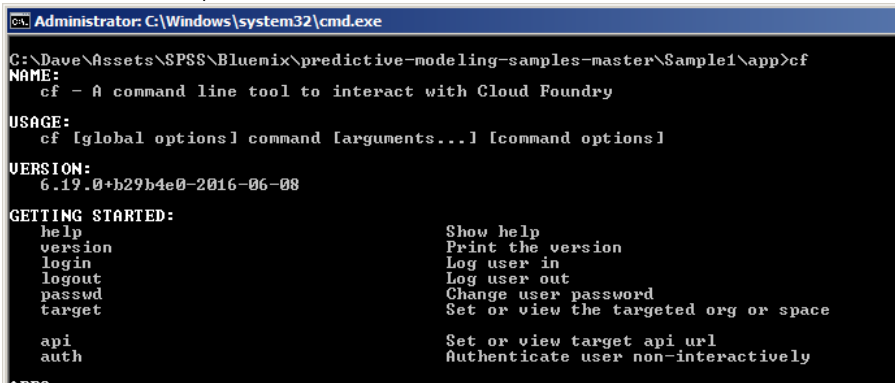
1. Ensure that you have a valid [Bluemix](#) account.

2.2 Cloud Foundry

2. Install the Cloud Foundry (CF) Command Line Interface (<https://github.com/cloudfoundry/cli/releases>).
3. Ensure that the tool is running properly by typing the following from a command prompt:

cf

You should receive output that looks similar to the screen below.



```
Administrator: C:\Windows\system32\cmd.exe
C:\Dave\Assets\SPSS\Bluemix\predictive-modeling-samples-master\Sample1\app>cf
NAME:
  cf - A command line tool to interact with Cloud Foundry

USAGE:
  cf [global options] command [arguments...] [command options]

VERSION:
  6.19.0+b29b4e0-2016-06-08

GETTING STARTED:
  help          Show help
  version       Print the version
  login         Log user in
  logout        Log user out
  passwd        Change user password
  target        Set or view the targeted org or space

  api           Set or view target api url
  auth          Authenticate user non-interactively
```

Note: You may have to reboot your system in order for it to locate the “cf” executable. Do not proceed if the “cf” command does not run.

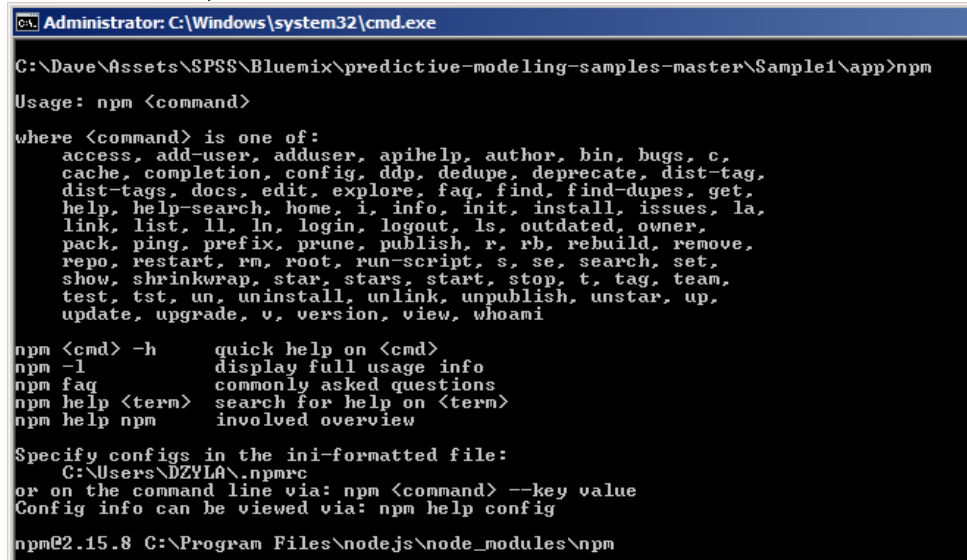
2.3 Node.js

Node.js is required to download supporting code from Bluemix to the local development environment. See <https://nodejs.org/en/about/> for more information.

1. Download and install Node.js from <https://nodejs.org/en/download/>.
2. Validate that Node.js is running properly by typing the following from a command prompt:

npm

You should receive output that looks similar to the screen below.



```
Administrator: C:\Windows\system32\cmd.exe
C:\Dave\Assets\SPSS\Bluemix\predictive-modeling-samples-master\Sample1\app>npm
Usage: npm <command>

where <command> is one of:
  access, add-user, adduser, apihelp, author, bin, bugs, c,
  cache, completion, config, ddp, dedupe, deprecate, dist-tag,
  dist-tags, docs, edit, explore, faq, find, find-dupes, get,
  help, help-search, home, i, info, init, install, issues, la,
  link, list, ll, ln, login, logout, ls, outdated, owner,
  pack, ping, prefix, prune, publish, r, rb, rebuild, remove,
  repo, restart, rm, root, run-script, s, se, search, set,
  show, shrinkwrap, star, stars, start, stop, t, tag, team,
  test, tst, un, uninstall, unlink, unpublish, unstar, up,
  update, upgrade, v, version, view, whoami

npm <cmd> -h      quick help on <cmd>
npm -l           display full usage info
npm faq          commonly asked questions
npm help <term>  search for help on <term>
npm help npm     involved overview

Specify configs in the ini-formatted file:
  C:\Users\DZYL\A\npmrc
or on the command line via: npm <command> --key value
Config info can be viewed via: npm help config

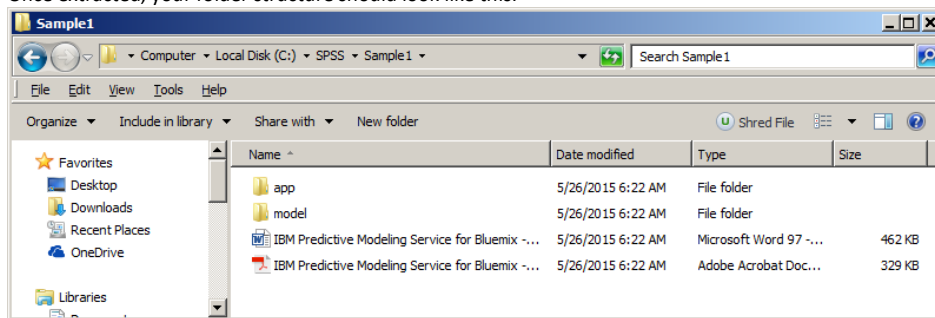
npm@2.15.8 C:\Program Files\nodejs\node_modules\npm
```

Note: You may have to reboot your system in order for it to locate the “npm” executable. Do not proceed if the “npm” command does not run.

2.4 Downloading Sample 1

1. Download Sample 1 from <https://github.com/pmsservice/predictive-modeling-samples>.
2. Extract the contents to a working directory. This example uses C:\SPSS.

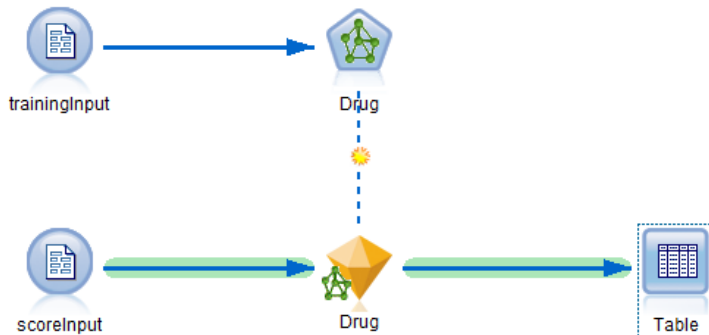
Once extracted, your folder structure should look like this:



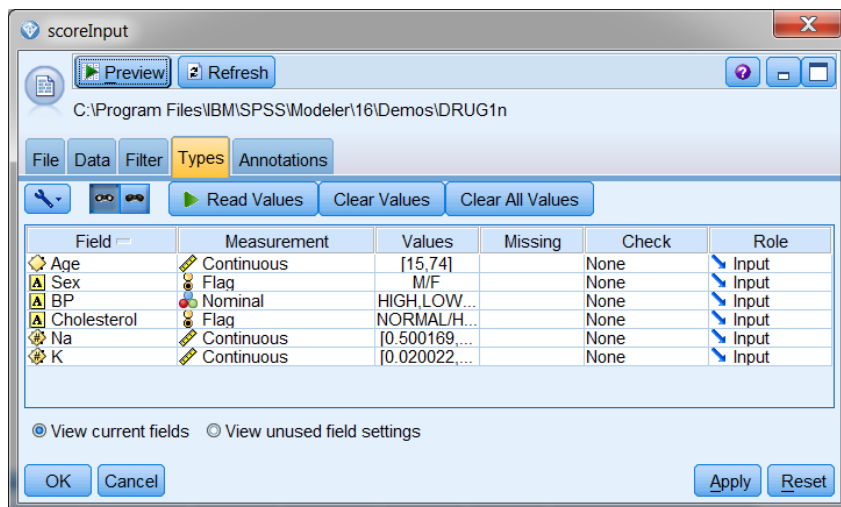
3 IBM SPSS Modeler

The sample comes with an SPSS Modeler stream which can be found at *C:\SPSS\Sample1\model\Drug1n.str*.

It is important to note that a scoring branch has been set in the stream. Any model used in the Bluemix Predictive Analytics node must have a default scoring branch set. This can simply be done by right-clicking the terminal node and selecting **Use as scoring branch**.



The Bluemix application will have to provide the same data items used in the input node. In this case it is the *scoreInput* node.



Bluemix will return scored results in the \$N-Drug (prediction) and \$NC-Drug (prediction confidence) columns.

Table

Settings

Format

Output

Annotations

Field	Format	Justify	Column Width
Age	####	Auto	Auto
Sex		Auto	Auto
BP		Auto	Auto
Cholesterol		Auto	Auto
Na	####.###	Auto	Auto
K	####.###	Auto	Auto
\$N-Drug		Auto	Auto
\$NC-Drug	####.###	Auto	Auto

☒ View current fields

☐ View unused field settings

OK

Run

Cancel

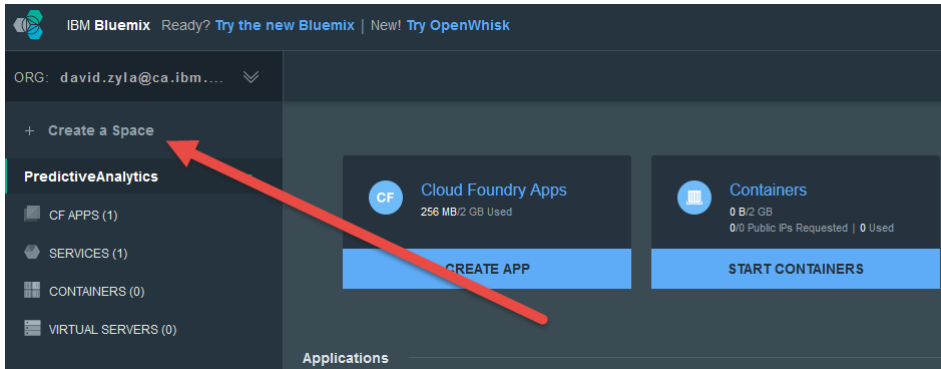
Apply

Reset

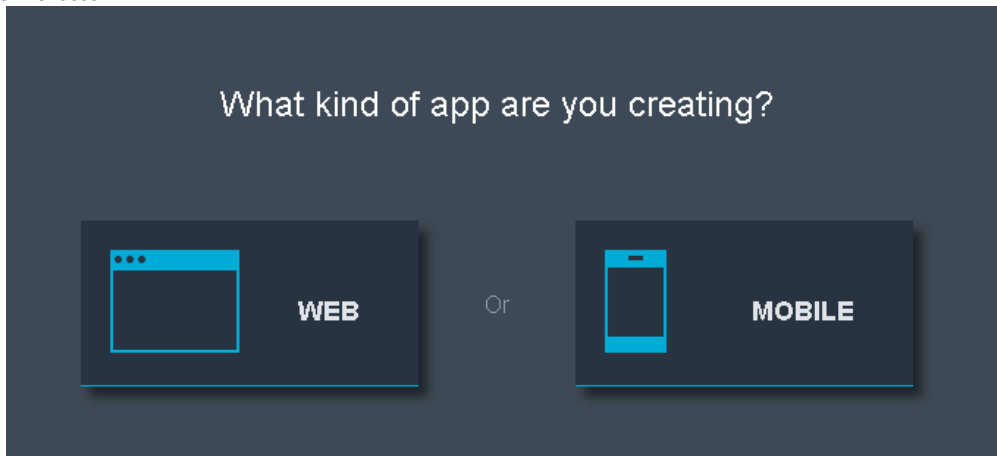
4 Bluemix application and service

4.1 Creating an application

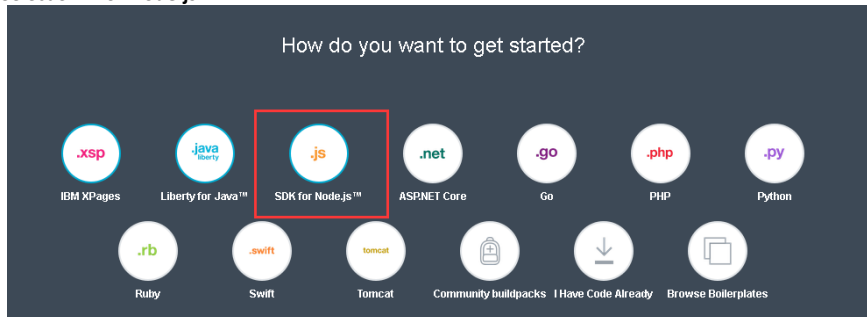
1. Log in to Bluemix.
2. If you don't already have a space created, create one now. Name the space however you like, but make note of the name for future reference.



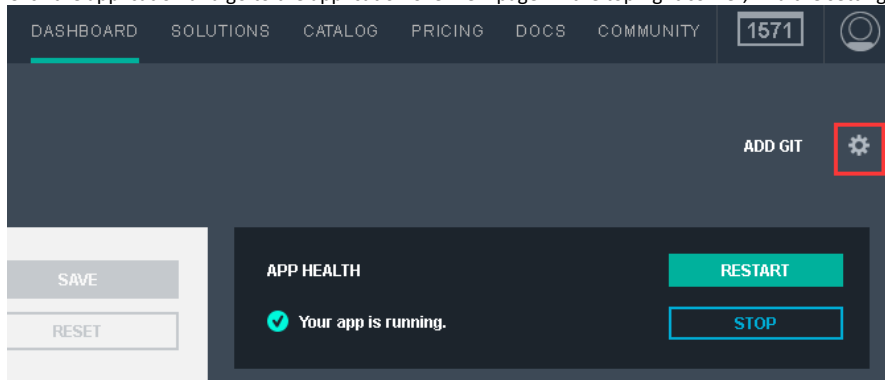
3. After creating the new space, click it from the left side of your screen.
4. On the top of the page, click **Create app**.
5. Choose **WEB**.



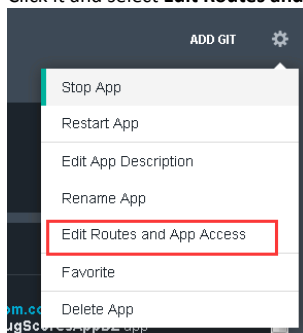
6. Select **SDK for Node.js**.



7. Name the new application however you'd like. This example uses the name **SPSSDrugScoresAppDZ**. We recommend using a short name.
8. Click **Create**. This will stage and create the application. This process may take a minute or two to complete.
9. After the application is created, go back to the dashboard. You should see the application you just created (for example, **SPSSDrugScoresAppDZ**).
10. Click the application and go to the application overview page. In the top right corner, find the settings button.



11. Click it and select **Edit Routes and App Access**.



12. The second selection box shows this app's domain name.

Edit Routes and App Access

spssdrugscoresappdz

stage1.eu-gb.mybluemix.net

[Add route](#)

Enable app authentication

Restrict app access to members of this organization

OFF

[Manage Domains](#)

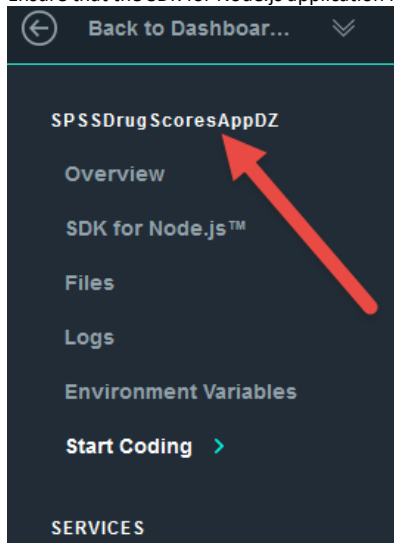
SAVE

CANCEL

13. Make note of the Space, App Name, Host, and Domain, as they will be referred to later on.


4.2 Creating the service

1. Ensure that the SDK for Node.js application is running and is selected on the left side of the page.



2. Click **Catalog** from the upper toolbar.
3. To filter the number of services displayed, select **Data and Analytics** under **Services** on the left side of the screen.
4. Click **Predictive Analytics**.

- On the **Add Service** screen, ensure that the **App** drop down is changed from **Leave Unbound** to the application that you created previously. This will bind the service to the application.



Predictive Analytics
IBM

PUBLISH DATE
05/28/2016

AUTHOR
IBM SPSS

TYPE
Service

[VIEW DOCS](#)

IBM Predictive Analytics is a full-service Bluemix offering that makes it easy for developers and data scientists to work together to integrate predictive capabilities with their applications. Built on IBM's proven SPSS analytics platform, Predictive Analytics allows you to develop applications that make smarter decisions, solve tough problems, and improve user outcomes.

Pick a plan Monthly prices shown are for country or region: [Canada](#)

Plan	Features	
Free Plan	1 Service instance (2 models maximum) 5,000 Real-time or batch predictions 5 Compute-hours for analysis and model building	Free
<p><i>The free plan provides you with a single instance of the Predictive Analytics service with a maximum of 2 models, 5,000 predictions per month, and 5 hours of compute time for analysis and model building operations.</i></p>		
Paid Plan	Service instance (20 models per instance) Real-time predictions Batch predictions Analysis and model building compute-hour	\$10.57 CAD/Instance \$0.53 CAD/1,000 Real-time predictions \$0.53 CAD/1,000 Batch predictions \$0.48 CAD/Compute-hour

[TERMS](#)

Add Service

Space: SPSS_RT_Scoring

App: SPSSDrugScoresAppDZ SPSSDrugScoresAppDZ...

Service name: Predictive Analytics-49

Selected Plan: Free Plan

CREATE

- Make note of the App and Service names.
- Click **Create**. If prompted to re-stage, select **Yes**.
- Navigate back to the top dashboard page to display both the application and service.

ORG: david.zyla@ca.ibm....

+ Create a Space

PredictiveAnalytics

SPSS_RT_Scoring

CF APPS (1)


SERVICES (1)

CONTAINERS (0)

VIRTUAL SERVERS (0)

CREATE APP


Applications



SPSSDrugScoresAppDZ
SPSSDrugScoresAppDZ.mybluemix.net

Running

Services



Predictive Analytics-49
Predictive Analytics

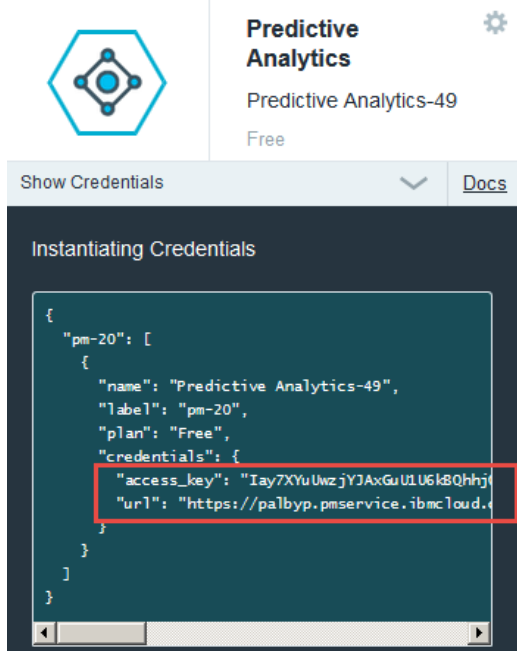
Plan: Free

- Click the application (**SPSSDrugScoresAppDZ**, for example).

10. When we score the input data in this example, credentials must be supplied to Bluemix at runtime. Click the up arrow to show the credentials.



11. In the fly-out window, make note of the the access key and URL. These will be needed later.



"access_key":

"Iay7XYuUwzjYJAxGuU1U6k8QhhjQLydoIFxLveWn3XUy98gODPY6vx3TabOtWZqeHxGxQ3plogjgEOjN0TGDTcL0h32gVzPkwMbmHXNpi+EHoRceM3jXlle+y3OFM4+zBk8Cu9s4bKs2CqegHQphv+eqc4NgPgKMPJdgqshDkv4="

"url":

"https://palbyp.pmservice.ibmcloud.com/pm/v1"

12. At this point, the application and service containers are ready.

5 Customizing files

This section deals with customizing the files that were downloaded to the SPSS Working directory `C:\SPSS\Sample1`.

5.1 `\app\manifest.yml`

Manifest.yml contains information about the deployment and is typically used to reduce the number of deployment details that you must specify every time you deploy an application to Bluemix.

1. Open the file `C:\SPSS\Sample1\app\manifest.yml` in a text editor.
2. Change the value of the **host** variable to reflect what was used in the *Creating an application* section previously. In this example, *SPSSDrugScoresAppDZ* was used.
3. Change the value of the **name** variable to reflect what was used in the *Creating an application* section previously. In this example, *SPSSDrugScoresAppDZ* was used.
4. Change the value of the **domain** variable to reflect what was used in the *Creating an application* section previously. In this example, *mybluemix.net* was used.
5. Change the value of the **memory** variable to 256M.
6. Add the following line:
 `disk_quota: 1024M`
7. Save and close the file.

5.2 `\app\app.js`

1. Open the file `C:\SPSS\Sample1\app\app.js` in a text editor.
2. Navigate to line 21 and update the *defaultBaseURL* that was recorded in the *Creating the service* section previously. Note that the `<>` must be removed and the entire string value must be enclosed in single quotes (`"`).

```
var defaultBaseURL = 'https://palbyp.pmservice.ibmcloud.com/pm/v1';
```

Comment [bf1]: 考虑到实际 bluemix 的情况

3. Similar to the previous step, update the *defaultAccessKey* value key on line 22.
4. Save and close the file.

5.3 `\app\public\js\app.js`

1. Open the file `C:\SPSS\Sample1\app\public\js\app.js` in a text editor.
2. Navigate to line 8.
3. Note that the value of the `$scope.context` variable is *drug1N*. While no change is required in this file, these steps are here to illustrate that every SPSS model scored in Bluemix must have a unique context ID. This will be discussed further in the *Uploading SPSS Models* section.
4. Close the file. No change is required.

5.4 \app\public\js\srv.js

1. Open the file `C:\SPSS\Sample1\app\public\js\srv.js` in a text editor:
2. Navigate to line 10. Note that the `tablename` variable matches the name of the source node in the SPSS Modeler model.



scoreInput

3. Examine lines 11 and 12. Note that these inputs match the input columns in the scoreInput node.

Field	Override	Storage	Input Format
Age	<input type="checkbox"/>	Integer	
Sex	<input type="checkbox"/>	String	
BP	<input type="checkbox"/>	String	
Cholesterol	<input type="checkbox"/>	String	
Na	<input type="checkbox"/>	Real	
K	<input type="checkbox"/>	Real	
Drug	<input type="checkbox"/>	String	

☒ View current fields ☐ View unused field settings

OK Cancel Apply Reset

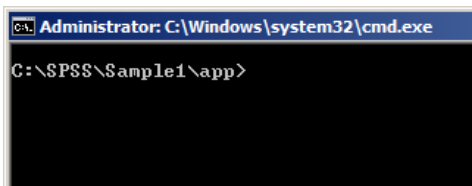
4. If you were to use a stream that used a different source node and had other parameters, they would need to be updated here.
5. Close the file. No change is required.

6 Porting source code

6.1 Node Package Manager

You can use the Node Package Manager utility to download supplemental files from Bluemix. The **Start Coding** section of the Bluemix application provides some of the commands that will help you with connecting to your Bluemix environment. It may be easier to use the commands there than to customize the steps below.

1. Launch a command prompt.
2. Change to the `C:\SPSS\Sample1\app` directory.



3. Connect to IBM Bluemix by typing the following command:
`bluemix api https://api.ng.bluemix.net`
4. Log in to Bluemix by typing the following command:
`bluemix login -u david.zyla@ca.ibm.com -o david.zyla@ca.ibm.com -s SPSS_RT_Scoring`
5. Type your password when prompted.

- Download the supplemental files by typing:
npm install

This will create the directory `C:\SPSS\Sample1\app\node_modules`.

```
Administrator: C:\Windows\system32\cmd.exe
C:\SPSS\Sample1\app>npm install
npm WARN package.json SPSS-PM-sample-1@0.1.0 No README data
npm WARN package.json SPSS-PM-sample-1@0.1.0 No license field.
body-parser@1.0.2 node_modules\body-parser
├── qs@0.6.6
├── raw-body@1.1.7 <string_decoder@0.10.31, bytes@1.0.0>
└── type-is@1.1.0 <mime@1.2.11>

express@4.0.0 node_modules\express
├── methods@0.1.0
├── parseurl@1.0.1
├── debug@0.8.1
├── qs@0.6.6
├── utils-merge@1.0.0
├── escape-html@1.0.1
├── cookie-signature@1.0.3
├── range-parser@1.0.0
├── fresh@0.2.2
├── buffer-crc32@0.2.1
├── path-to-regexp@0.1.2
├── merge-descriptors@0.0.2
├── cookie@0.1.0
├── type-is@1.0.0 <mime@1.2.11>
├── send@0.2.0 <mime@1.2.11>
├── accepts@1.0.0 <mime@1.2.11, negotiator@0.3.0>
└── serve-static@1.0.1 <send@0.1.4>

request@2.36.0 node_modules\request
├── aws-sign2@0.5.0
├── forever-agent@0.5.2
├── tunnel-agent@0.4.3
├── qs@0.6.6
├── oauth-sign@0.3.0
├── mime@1.2.11
├── json-stringify-safe@5.0.1
├── tough-cookie@2.2.2
├── node-uuid@1.4.7
├── http-signature@0.10.1 <assert-plus@0.1.5, ctype@0.5.3, asn1@0.1.11>
├── form-data@0.1.4 <async@0.9.2, combined-stream@0.0.7>
└── hawk@1.0.0 <cryptiles@0.2.2, boom@0.4.2, sntp@0.2.4, hoek@0.9.1>
```

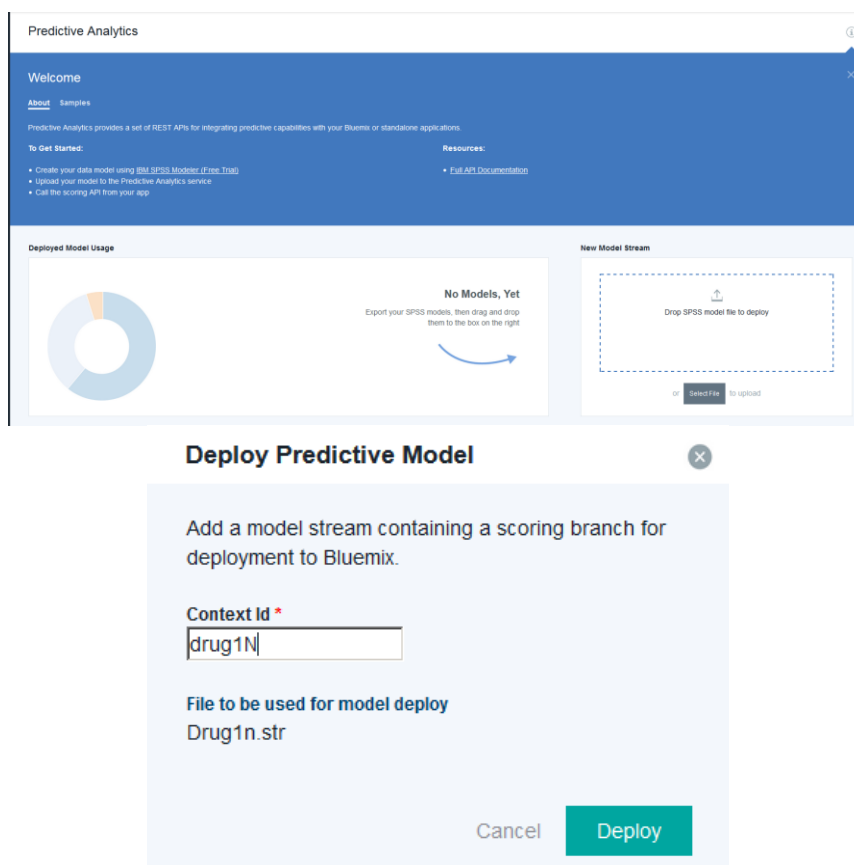
- To push all files, including the customizations that were made, run the following command:
cf push <Application Name>
cf push SPSSDrugScoresAppDZ

The application is uploaded to Bluemix.

7 Uploading SPSS Models


At this point, you have moved your custom application to IBM Bluemix. But you still need to provide Bluemix with the SPSS models that you want to use.

1. Open a browser and log in to Bluemix.
2. From the dashboard, click the Predictive Analytics service that was created previously.
3. In the bottom right corner, note there is a **New Model Stream** section. The SPSS model we are going to provide can be found in `C:\SPSS\Sample1\model\Drug1n.str`. Upload the model by either dragging and dropping it or by using the **Select File** option.
4. You will be prompted to provide a Context Id. While this ID can be anything, this example uses **drug1N**.



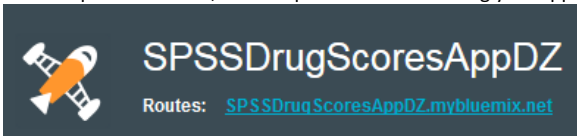
5. Type **drug1N** for the Context ID. If you choose a different Context ID, you must make the respective change in `C:\SPSS\Sample1\app\public\js\app.js` which was discussed earlier.
6. Click **Deploy**. The model is listed in the bottom section of the screen.

Manage Models - Status: Active

Content Id	File	Date Created	Date Updated	Action
drug1N	DrugIn.str	7/8/16	7/8/16	

8 Real-time scoring the sample

1. In Bluemix, navigate to your dashboard.
2. Click the application you created.
3. At the top of the screen, a URL is provided for accessing your application.



4. Click the URL. You are presented with your real-time scoring application.

A screenshot of the 'Data Input' form for the SPSSDrugScoresAppDZ application. The form has a dark blue background with white text. It includes input fields for 'Age' (35), 'Sex' (Male), 'Blood Pressure' (NORMAL), 'LDL (bad) Cholesterol' (NORMAL), 'Sodium Level' (0.697), and 'Potassium Level' (0.056). A 'Score Now' button is at the bottom left.

Data Input

Age: 35

Sex: ☒ Male ☐ Female

Blood Test Results

Blood Pressure: ☐ HIGH ☒ NORMAL ☐ LOW

LDL (bad) Cholesterol: ☐ HIGH ☒ NORMAL ☐ LOW

Sodium Level: 0.697

Potassium Level: 0.056

[Score Now](#)

5. You can change any of the values to simulate data being entered by a user of this application.
6. Click the **Score Now** button.

A screenshot of the 'Score Results' dialog box. It displays a table with the input data and the predicted drug and confidence score. A 'Close' button is at the bottom left.

Score Results

Age	Sex	BP	Cholesterol	Na	K	\$N-Drug	\$NC-Drug
35	M	NORMAL	NORMAL	0.697	0.056	drugX	0.9999482075987031

[Close](#)

7. In this example, the input along with the predicted drug and predicted confidence are being displayed on the screen.

This document has walked through the process of:

- Setting up the necessary development tools
- Customizing the sample code
- Uploading an SPSS model
- Scoring the SPSS model with data provided in real-time