Build, train, and deploy a machine-learning model without coding and invoke it from the Command Line

Get experience with **IBM Watson Studio** by creating a decision-tree machine-learning model to evaluate the risk that a customer might leave your service.

Duration: 20 minutes

Experience: clicker

In this tutorial, you use **IBM Watson Studio** to train and deploy a machine-learning model without any coding or manual model building skills.

You complete the following tasks:

- Create a project (unless you have already an existing project)
- Load a data set into the project
- Use the IBM Watson Studio model builder to train, test, and evaluate a machine-learning model
- Deploy the model

Log in to IBM Watson Studio

In IBM Watson Studio, data scientists work and collaborate in projects. In this task, you set up a project.

> If you have already completed previous labs on IBM Watson Studio, jump directly to the Data Assets section on bullet 8.

1. Go to https://dataplatform.cloud.ibm.com to access IBM Watson studio cloud-based service.

2. Click **Login** and provide your IBM ID credentials.

3. When logged in to IBM Watson Studio the first page that you see is the **Get started with key tasks** which helps you to get a customizable user experience whether you are a data scientist, a data engineer… You can minimize this pane clicking **Hide**.

If you scroll, you can explore the **Community Page** where you can distribute public assets, such as notebooks and data sets, which can be used by any IBM Watson Studio user. Community page is also a great way to grow your data scientist skills.

4. In the upper-right corner, click the **+ New Project** link, to create project. Alternatively you can use an existing project.
   The "New project" page opens and lists a set of templates (tiles) which tailor the tools based on the tasks you want to accomplish.
   Select **Modeler**.

5. Give your project a name, let's say `CustomerChurn`, a detailed description.

6. Click **Create**.
   On the project's summary page, you can see that the project has no assets.

7. Click **Assets**.
   The various project components are shown, such as notebooks, models, and data sets. You're going to add a data set to the project.

8. **Data sets** can be local or remote. In this tutorial, you work with a small local data set.

Click the Find and add Data icon which looks like a 10 01 button. It will open the file management sidebar.

9. From the **Load** tab, Click **Browse** to select from your local file system.
   Navigate to the lab files folder (given by your instructor) and select `customer_churn.csv` and click **Open**.
   Alternatively, you can drag and drop a file directly into the sidebar.
   The file is added to your local data sets in your project.

Next, you will create a machine-learning model by using this newly loaded data set.

Create the model

Create a model by using the IBM Watson Studio model builder.

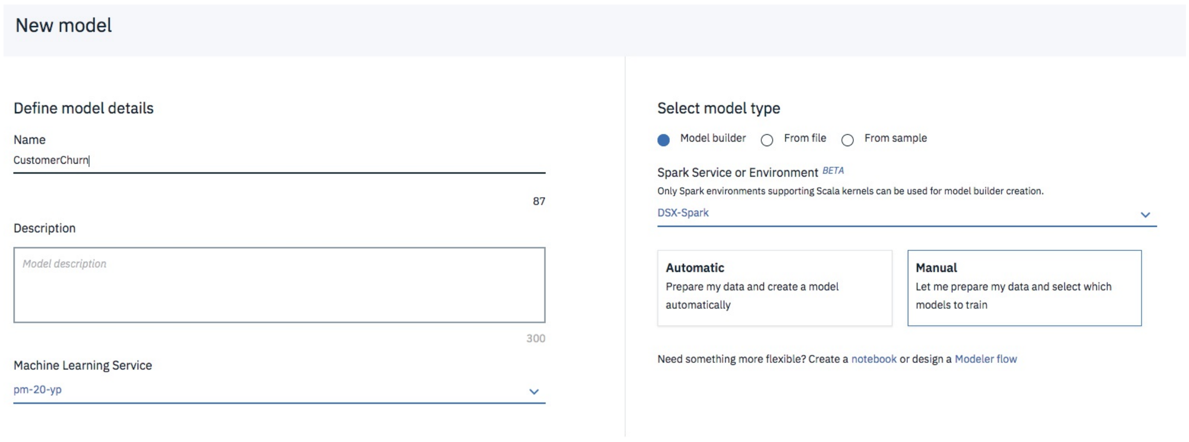
1. Scroll to the **Models** section of the project.
   A project can host several models of different types:

   - Natural Language Classifier models
   - Visual Recognition models

  - Watson Machine Learning models

2. Click **+ New Watson Machine Learning Model**
   On the New Model page, the `Model builder` model type is selected by default.
   Give a name to the model, `customerchurn` for example.
   The model builder can be fully automated or can leave certain steps to be completed manually. In this tutorial, you use the manual option so that you can review the capabilities of the model builder.
   Notice that an **IBM Watson Machine Learning Service** is required, you should select an existing one or you can create a new one at this point.
   A **Spark Service or Environment** is also needed at this point. Select your service or create a new one at this point.

3. Click **Manual** to get more options in data selection for training your model.


New model

Define model details

Name
CustomerChurn

87

Description

Model description

300

Machine Learning Service
pm-20-yp

Select model type

[✓ Model builder] From file From sample

Spark Service or Environment *BETA*
Only Spark environments supporting Scala kernels can be used for model builder creation.
DSX-Spark

Automatic
Prepare my data and create a model automatically

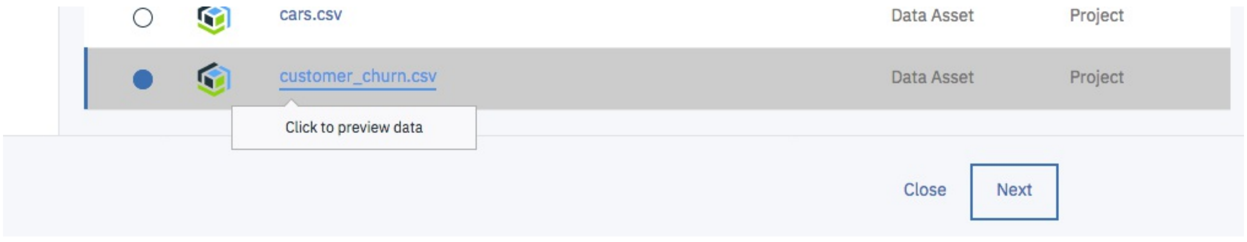
Manual
Let me prepare my data and select which models to train

Need something more flexible? Create a notebook or design a Modeler flow

4. Click **Create**.

5. On the "Select data asset" page, you can select the data asset to use to create your model.
   Click `customer_churn.csv` and then click **Next**.

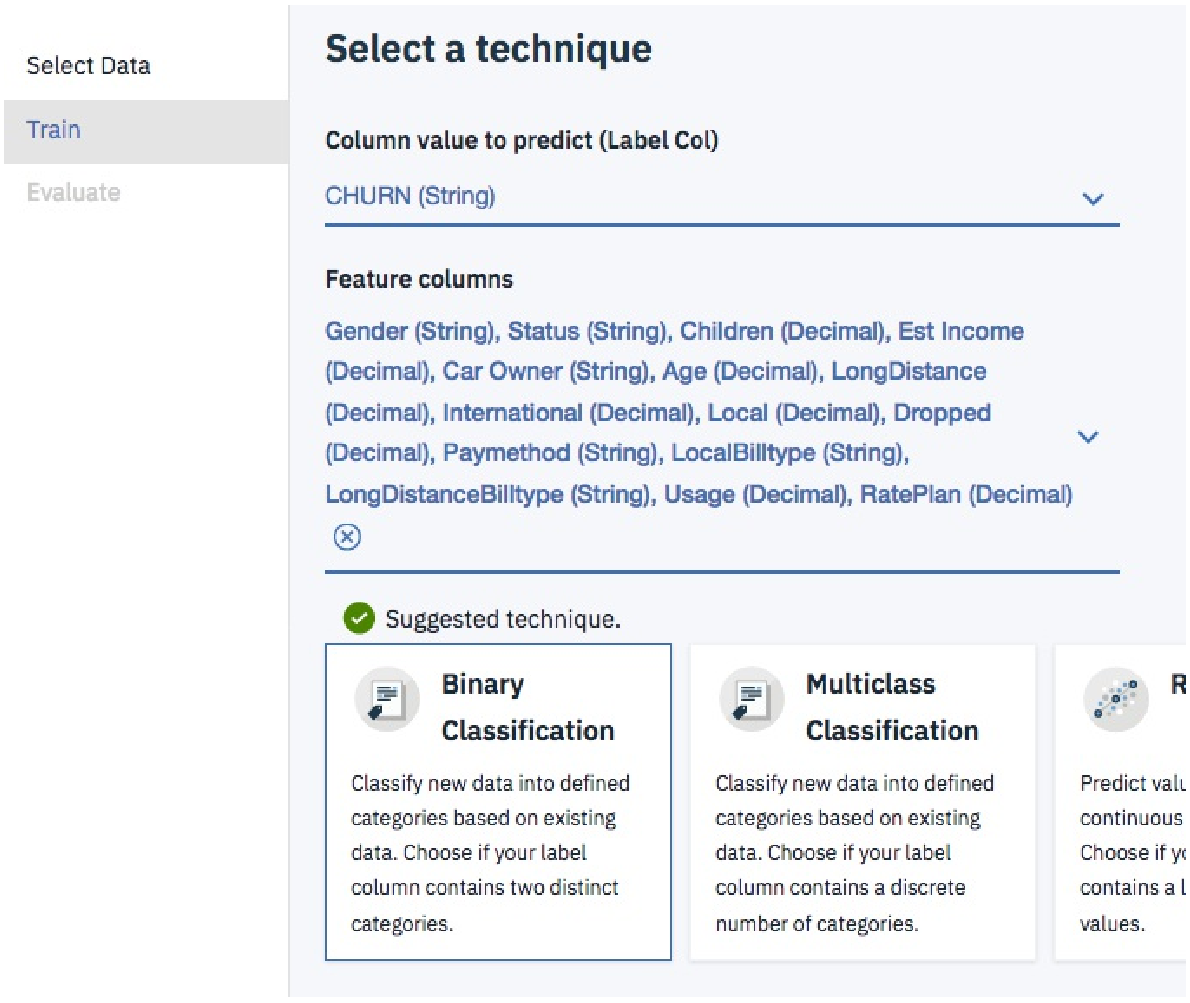
> Note that you could also add your Data Assets from here.



cars.csv	Data Asset	Project
[✓ customer_churn.csv]	Data Asset	Project

6. On the "Select a technique" page, you select the attributes of the machine-learning model.
   From the **Column value to predict (Label Col)** list, select **CHURN**. This is the column

that contains the historical outcomes and thus the predicted outcome.
The **Feature columns** are columns that contain the attributes on which the machine learning model will base predictions. All columns (features) are selected by default, you can leave it as-is or select **all BUT the ID column**.



Select a technique

Column value to predict (Label Col)

CHURN (String)

Feature columns

Gender (String), Status (String), Children (Decimal), Est Income (Decimal), Car Owner (String), Age (Decimal), LongDistance (Decimal), International (Decimal), Local (Decimal), Dropped (Decimal), Paymethod (String), LocalBilltype (String), LongDistanceBilltype (String), Usage (Decimal), RatePlan (Decimal)

Suggested technique.

Binary Classification

Classify new data into defined categories based on existing data. Choose if your label column contains two distinct categories.

Multiclass Classification

Classify new data into defined categories based on existing data. Choose if your label column contains a discrete number of categories.

R

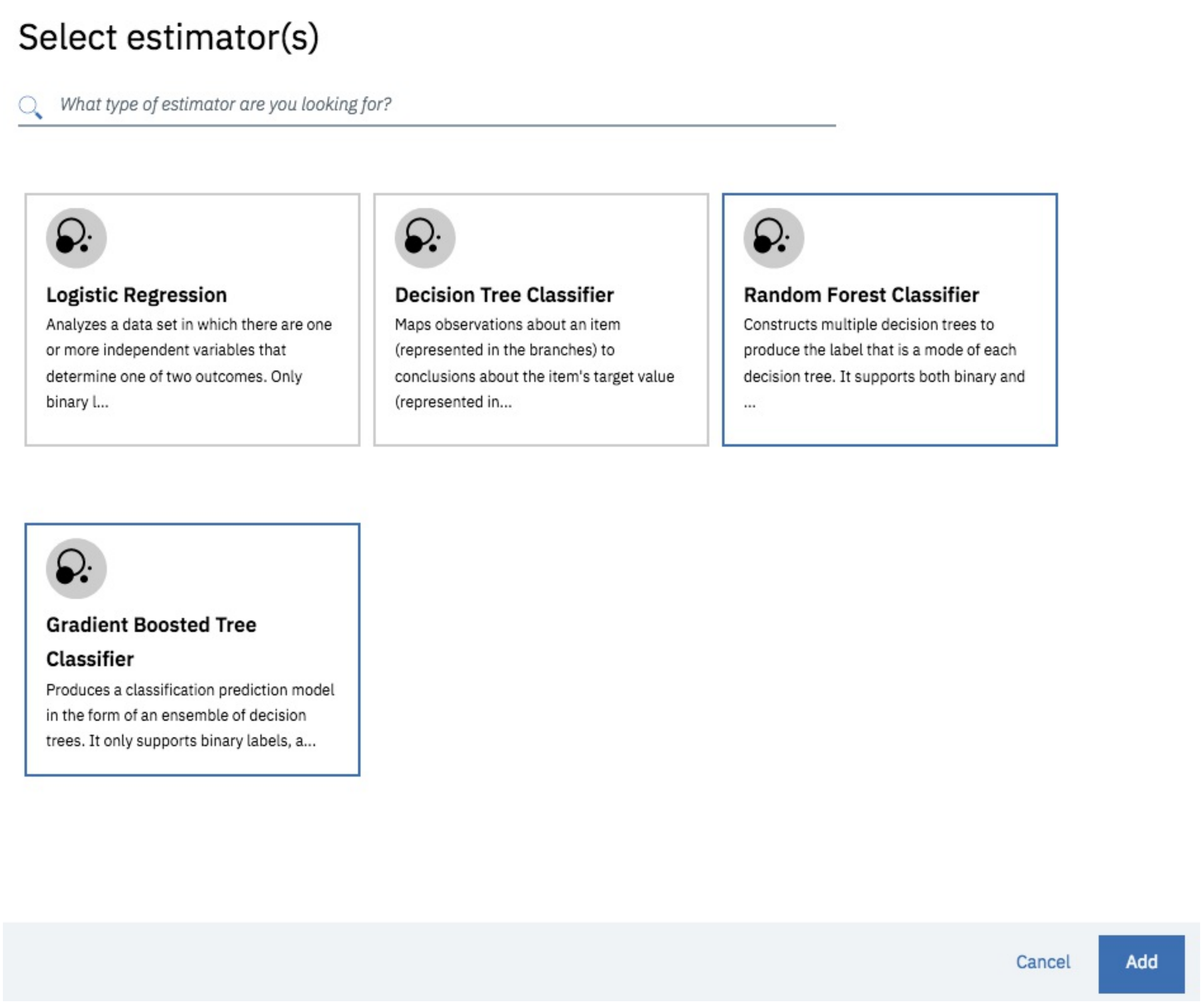
Predict valu
continuous
Choose if y
contains a l
values.

7. The model builder selects **Binary Classification** by default as the type of model to build. Because you want to create that type of model, leave the selection as-is.

8. In the upper-right corner, click **Add Estimators** to select the type of machine-learning algorithm to train.

9. On the "Select estimator(s)" window, from among the four algorithms, click **Random Forest Classifier** and **Gradient Boosted Tree Classifier**. Click **Add**.

> The estimators (machine-learning models) are shown on the next page, where you can also modify the percentage split between training, testing, and validation sets. Leave the default settings as-is.


Select estimator(s)

What type of estimator are you looking for?

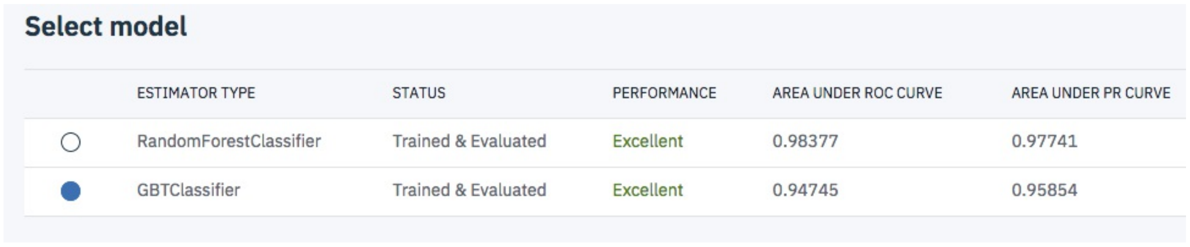
Logistic Regression
Analyzes a data set in which there are one or more independent variables that determine one of two outcomes. Only binary l...

Decision Tree Classifier
Maps observations about an item (represented in the branches) to conclusions about the item's target value (represented in...

Random Forest Classifier
Constructs multiple decision trees to produce the label that is a mode of each decision tree. It supports both binary and ...

Gradient Boosted Tree Classifier
Produces a classification prediction model in the form of an ensemble of decision trees. It only supports binary labels, a...

10. Click **Next**.
    The two models are trained and their respective evaluations are displayed. The Gradient Boosted Tree Classifier model shows better performance results and is selected by default.


Select model

	ESTIMATOR TYPE	STATUS	PERFORMANCE	AREA UNDER ROC CURVE	AREA UNDER PR CURVE
○	RandomForestClassifier	Trained & Evaluated	Excellent	0.98377	0.97741
●	GBTClassifier	Trained & Evaluated	Excellent	0.94745	0.95854

11. Click **Save**. When you're prompted to confirm, click **Save** again.
    You're returned to the **Models** summary page, on the **Overview** tab, where your new model is listed.

You now have a trained model, next you will deploy the model to test on out-of-sample data.

Deploy and test a trained model

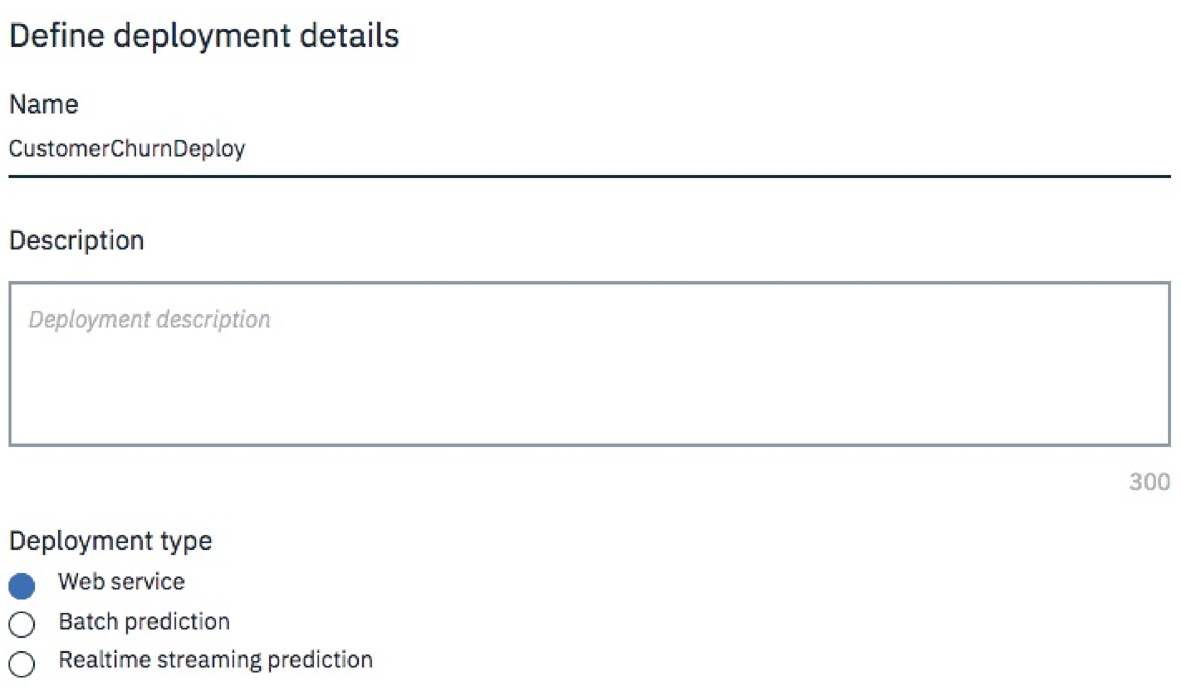
Before you can use your trained model to make predictions on new data, you must deploy the model. It's time to deploy the model and start to score a few records.

1. From the model summary page, click the **Deployments** tab.

2. Click **Add Deployment** link on the upper-right section of the pane.

3. On the **Create Deployment** page, give a name and a description to your deployment, say `CustomerChurnDeploy`.

> Note that deployment must have a unique name.


Define deployment details

Name
CustomerChurnDeploy

Description
Deployment description
300

Deployment type
● Web service
○ Batch prediction
○ Realtime streaming prediction

4. By default **Web service** is selected as the deployment type. Click **Save**.

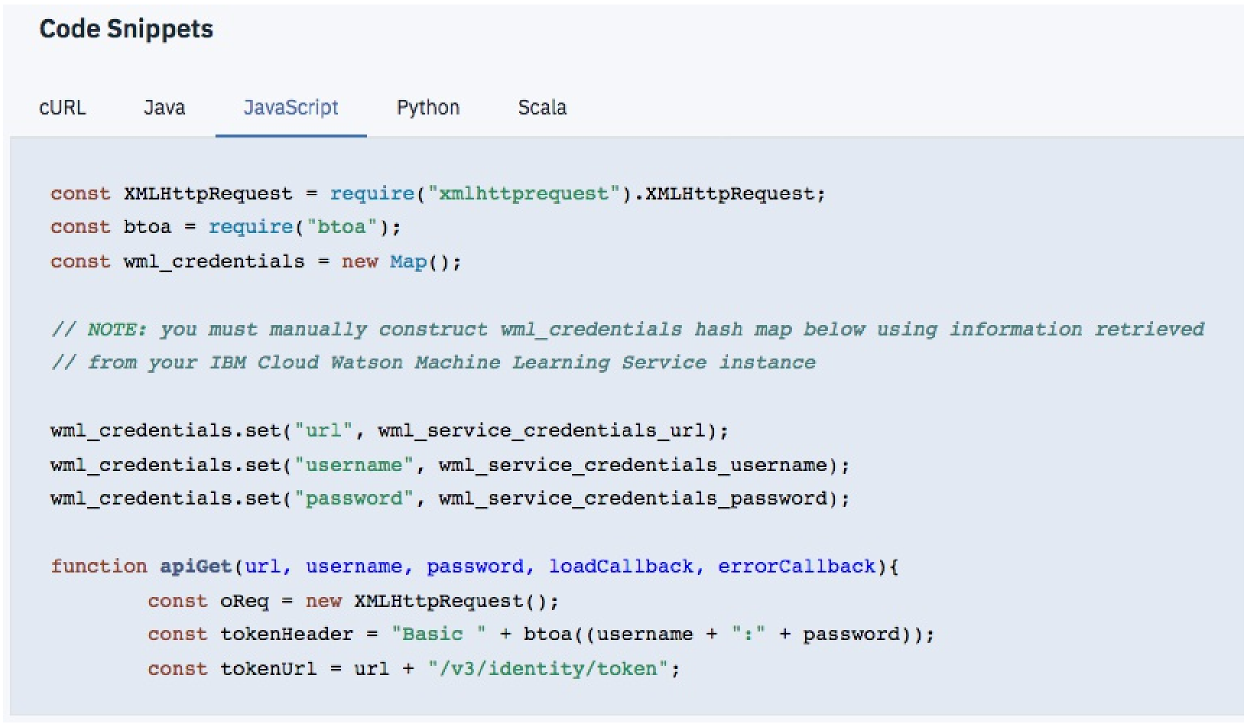
> **Batch prediction** and **Realtime streaming prediction** are deployment types for continuous learning and model evaluation. For **Batch Prediction** source could be a file in your **IBM CLoud Object Storage** or a connection and table from **IBM DB2 Warehouse on Cloud**.
> For **Real-time Streaming** the source is a streaming data source from a **Message Hub** instance.

5. When model deployment is complete, from the **Actions menu**, click **View**. Alternatively you can click on the Deployment name. The **Deployment Details** window appears.

6. Click on the **Implementation** tab. Note the scoring end point for future reference. **You**

can only have one deployment per user.

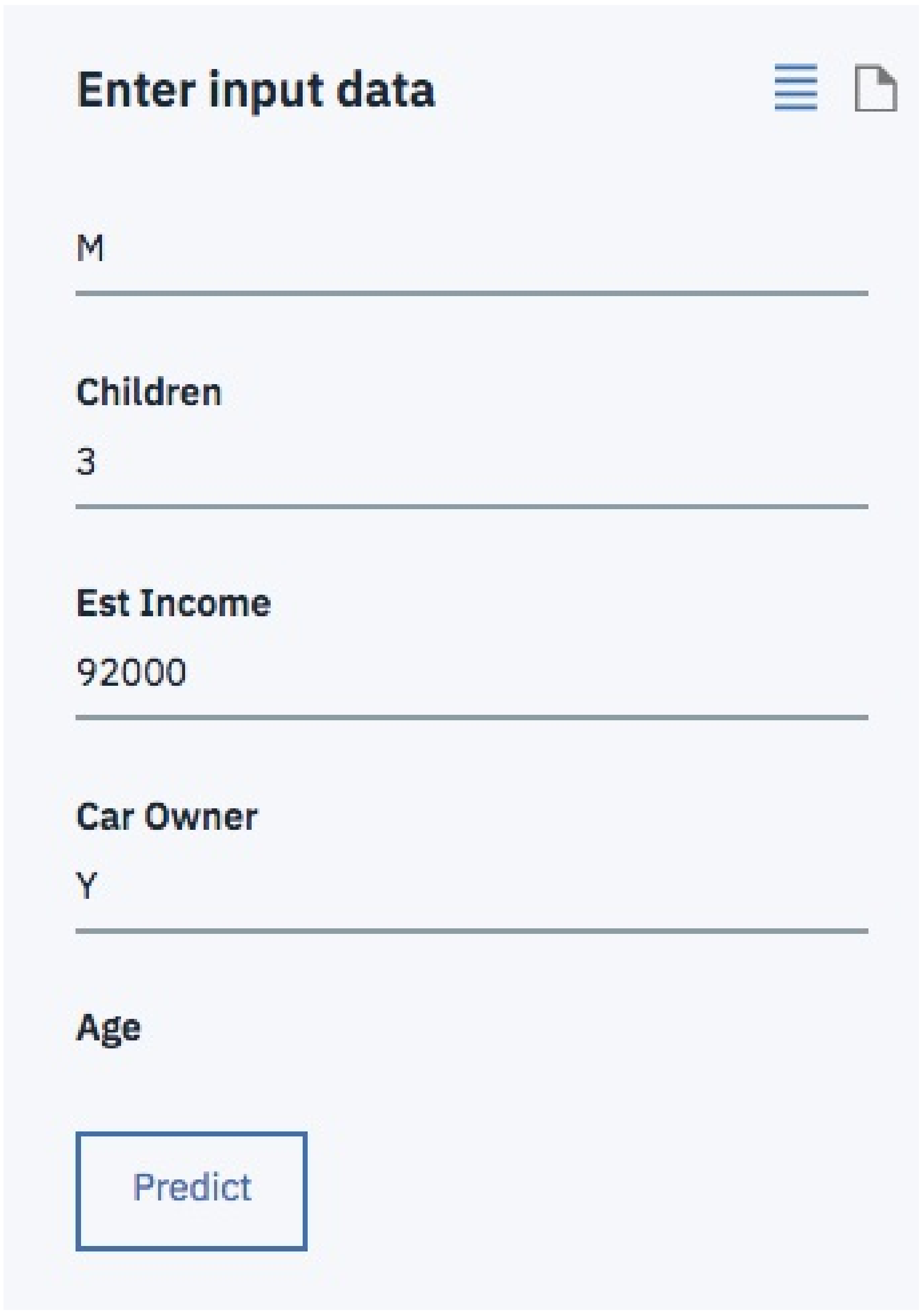
> The **Code Snippets** are available to pass it on to you application developer to integrate the deployed model into a business application.



```
const XMLHttpRequest = require("xmlhttprequest").XMLHttpRequest;
const btoa = require("btoa");
const wml_credentials = new Map();

// NOTE: you must manually construct wml_credentials hash map below using information retrieved
// from your IBM Cloud Watson Machine Learning Service instance

wml_credentials.set("url", wml_service_credentials_url);
wml_credentials.set("username", wml_service_credentials_username);
wml_credentials.set("password", wml_service_credentials_password);

function apiGet(url, username, password, loadCallback, errorCallback){
        const oReq = new XMLHttpRequest();
        const tokenHeader = "Basic " + btoa((username + ":" + password));
        const tokenUrl = url + "/v3/identity/token";
```

7. Now is a good time to test the model prediction. Go to **Test** tab.

8. Enter data in all the fields for a sample record from the data set.


Enter input data

M

Children

3

Est Income

92000

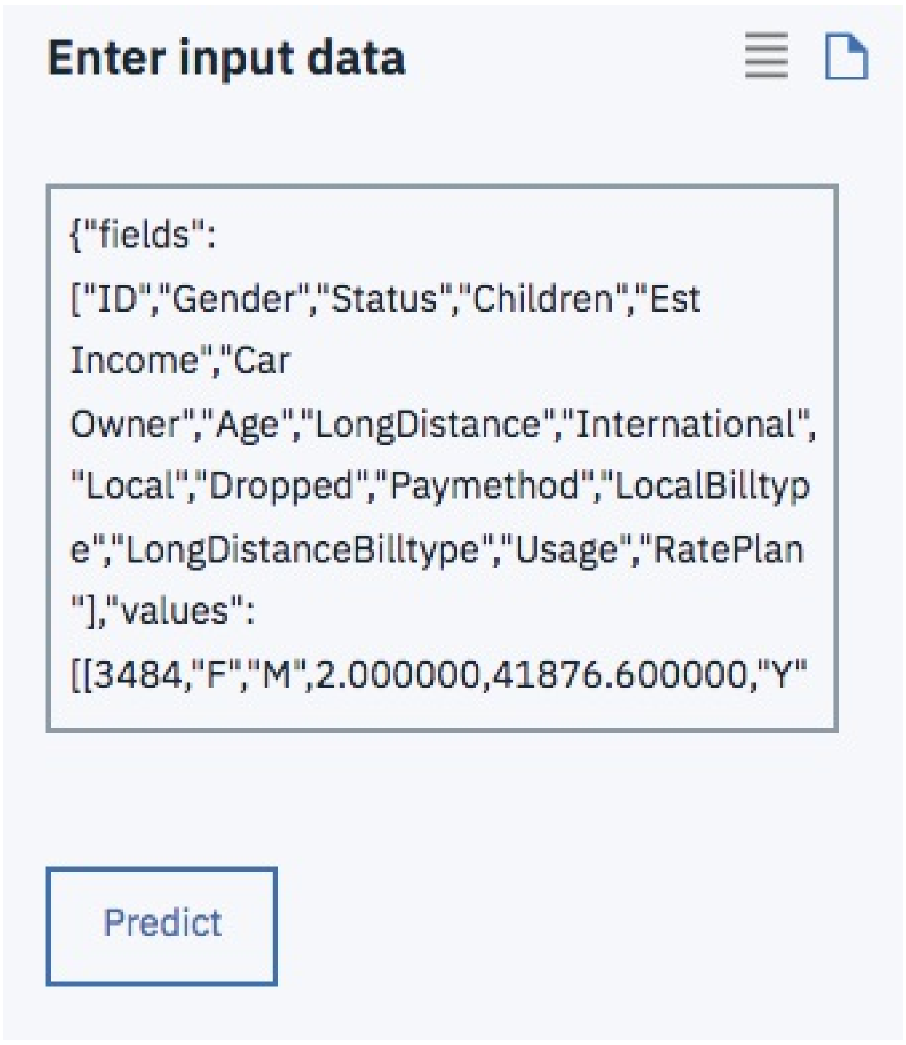
Car Owner

Y

Age

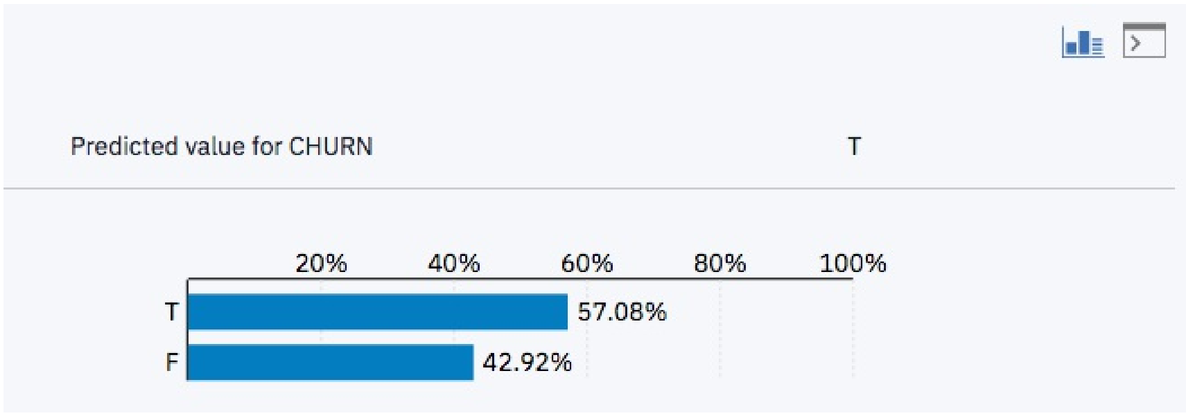
Get data from the `new_customer_churn_data.csv` file or copy the data from here and paste into the `Input JSON Payload data`:

```
{"fields":["ID","Gender","Status","Children","Est Income","Car Owner","Age","LongDistance","International","Local","Dropped","Paymethod","LocalBilltype","LongDistanceBilltype","Usage","RatePlan"],"values":[[3484,"F","M",2.000000,41876.600000,"Y",44.093333,6.780000,0.000000,132.650000,0.000000,"CC","FreeLocal","Standard",139.440000,4.000000]]}
```


Enter input data

```
{"fields":
["ID","Gender","Status","Children","Est
Income","Car
Owner","Age","LongDistance","International",
"Local","Dropped","Paymethod","LocalBilltyp
e","LongDistanceBilltype","Usage","RatePlan
"],"values":
[[3484,"F","M",2.000000,41876.600000,"Y"
```

9. To test the model click **Predict**.


Predicted value for CHURN T

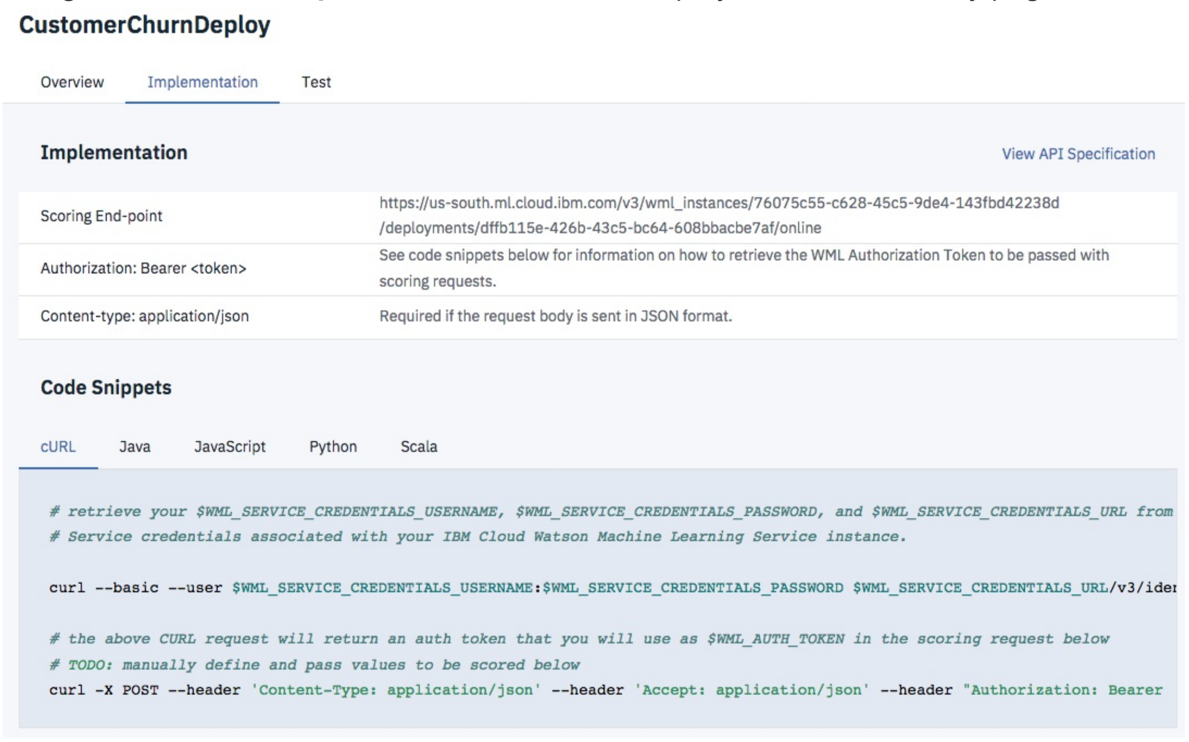
10. You can test several out-of-sample records provided as a payload in JSON format. Click on the **document** icon.

Copy-paste the JSON from a text editor and click **Predict**.
The results of scoring the tuple (tuples) against the model are shown either in raw data format or as an output chart.

Score from the command line

IBM Watson Studio has a built-in test client but you could opt to use `cURL` call from any command-line interface or in a web front-end application. In order to do so, you will have to retrieve an **Authorization token** to access your **IBM Cloud Watson Machine Learning Service** instance.

1. Navigate back to the **Implementation** tab of the deployed model summary page.



Implementation

Scoring End-point	https://us-south.ml.cloud.ibm.com/v3/wml_instances/76075c55-c628-45c5-9de4-143fbd42238d /deployments/dffb115e-426b-43c5-bc64-608bbacbe7af/online
Authorization: Bearer <token>	See code snippets below for information on how to retrieve the WML Authorization Token to be passed with scoring requests.
Content-type: application/json	Required if the request body is sent in JSON format.

Code Snippets

```
# retrieve your $WML_SERVICE_CREDENTIALS_USERNAME, $WML_SERVICE_CREDENTIALS_PASSWORD, and $WML_SERVICE_CREDENTIALS_URL from
# Service credentials associated with your IBM Cloud Watson Machine Learning Service instance.

curl --basic --user $WML_SERVICE_CREDENTIALS_USERNAME:$WML_SERVICE_CREDENTIALS_PASSWORD $WML_SERVICE_CREDENTIALS_URL/v3/ider

# the above CURL request will return an auth token that you will use as $WML_AUTH_TOKEN in the scoring request below
# TODO: manually define and pass values to be scored below
curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' --header "Authorization: Bearer
```

2. In the **Code Snippets** for the `cURL` tab, you'll see two *curl* commands. The first one retrieves the token.
   The second one sends the scoring request to the deployed model.
   Copy paste those lines of code to prevent from typing again.

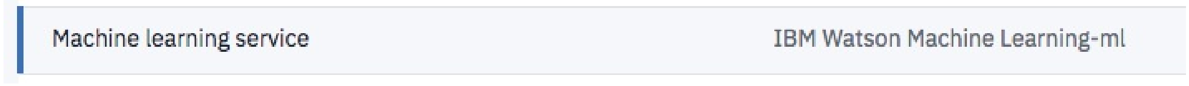
> Note that service credentials are needed for the IBM Watson Machine Learning Service instance.

3. Navigate back to the **Overview** tab. Notice the **Machine learning service** line, this is the service your deployed model is running on and you need to retrieve the service credentials for this one.

```
curl --basic --user
$WML_SERVICE_CREDENTIALS_USERNAME:$WML_SERVICE_CREDENTIALS_PASSWORD
$WML_SERVICE_CREDENTIALS_URL/v3/identity/token
```

4. From the toolbar, click on **Services** tab and **Watson Services** to open in a new window. Hit the `Cmd` key on Mac to open in a new browser tab.

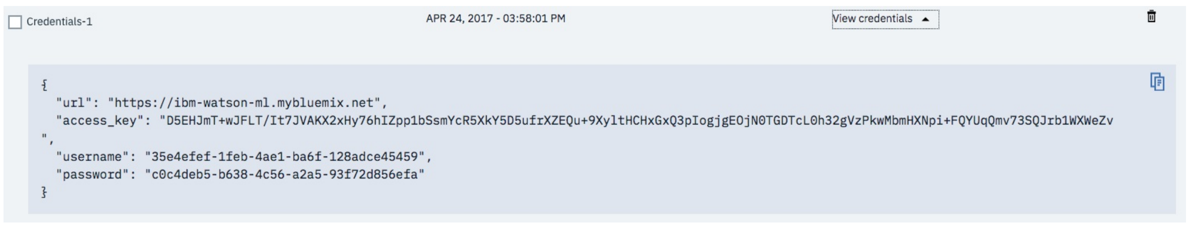
5. The **Watson Services** page is displayed with several sections (Discovery, Personality Insights, Machine Learning, …). Scroll to **Machine Learning**, you should see your service instance. Click on the service name to display the service management page.



Machine learning service   IBM Watson Machine Learning-ml

6. From the left sidebar, click **Service Credentials**. If there are no credentials listed, click on **New credential**, provide a name and click **Add**.

7. From **Service Credentials** list, click on **View credentials** in the **ACTIONS** column.



```
{
  "url": "https://ibm-watson-ml.mybluemix.net",
  "access_key": "D5EHJmT+wJFLT/It7JVAKX2xHy76hIZpp1bSsmYcR5XkY5D5ufrXZEQu+9XyltHCHxGxQ3pIogjgEOjN0TGDTcL0h32gVzPkwMbmHXNpi+FQYUqQmv73SQJrb1WXWeZv
",
  "username": "35e4efef-1feb-4ae1-ba6f-128adce45459",
  "password": "c0c4deb5-b638-4c56-a2a5-93f72d856efa"
}
```

Copy the credentials using the copy icon on the upper-right corner of the code pane.

8. Open a `terminal` or a `command line` window.

9. Set the variables based on the provided credential values.
`export WML_SERVICE_CREDENTIALS_USERNAME=<your value>`
`export WML_SERVICE_CREDENTIALS_PASSWORD=<your value>`
`export WML_SERVICE_CREDENTIALS_URL=https://ibm-watson-ml.mybluemix.net`

10. Retrieve your access token running the command below.
`curl --basic --user $WML_SERVICE_CREDENTIALS_USERNAME:$WML_SERVICE_CREDENTIALS_PASSWORD $WML_SERVICE_CREDENTIALS_URL/v3/identity/token`

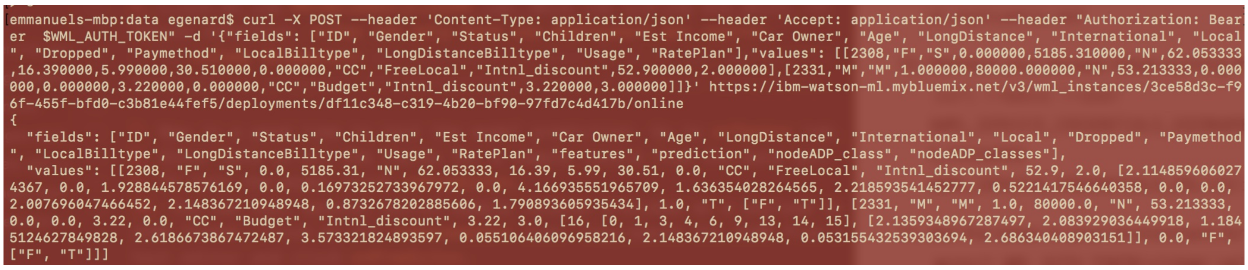
11. Set the token in your environment for later use.
`export WML_AUTH_TOKEN=<token value>`

12. Run the following `cURL` command providing the array of test values available from the labs data folder.

`curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' --header "Authorization: Bearer $WML_AUTH_TOKEN" -d '{"fields": ["ID", "Gender", "Status", "Children", "Est Income", "Car Owner", "Age", "LongDistance", "International", "Local", "Dropped", "Paymethod",`

```
"LocalBilltype", "LongDistanceBilltype", "Usage", "RatePlan"],"values":
[$ARRAY_OF_VALUES_TO_BE_SCORED, $ANOTHER_ARRAY_OF_VALUES_TO_BE_SCORED]}' <scoring
end-point>
```

> Note your **scoring end-point** should look something like the following URL (it might be a good idea to set it as an environment variable as well): https://ibm-watson-ml.mybluemix.net/v3/wml_instances/3ce58d3c-f96f-455f-bfd0-c3b81e44fef5/deployments/df11c348-c319-4b20-bf90-97fd7c4d417b/online



```
emmanuels-mbp:data egenard$ curl -X POST --header 'Content-Type: application/json' --header 'Accept: application/json' --header "Authorization: Bear
er  $WML_AUTH_TOKEN" -d '{"fields": ["ID", "Gender", "Status", "Children", "Est Income", "Car Owner", "Age", "LongDistance", "International", "Local
", "Dropped", "Paymethod", "LocalBilltype", "LongDistanceBilltype", "Usage", "RatePlan"],"values": [[2308,"F","S",0.000000,5185.310000,"N",62.053333
,16.390000,5.990000,30.510000,0.000000,"CC","FreeLocal","Intnl_discount",52.900000,2.000000],[2331,"M","M",1.000000,80000.000000,"N",53.213333,0.000
000,0.000000,3.220000,0.000000,"CC","Budget","Intnl_discount",3.220000,3.000000]]}' https://ibm-watson-ml.mybluemix.net/v3/wml_instances/3ce58d3c-f9
6f-455f-bfd0-c3b81e44fef5/deployments/df11c348-c319-4b20-bf90-97fd7c4d417b/online
{
  "fields": ["ID", "Gender", "Status", "Children", "Est Income", "Car Owner", "Age", "LongDistance", "International", "Local", "Dropped", "Paymethod
", "LocalBilltype", "LongDistanceBilltype", "Usage", "RatePlan", "features", "prediction", "nodeADP_class", "nodeADP_classes"],
  "values": [[2308, "F", "S", 0.0, 5185.31, "N", 62.053333, 16.39, 5.99, 30.51, 0.0, "CC", "FreeLocal", "Intnl_discount", 52.9, 2.0, [2.114859606027
4367, 0.0, 1.928844578576169, 0.0, 0.16973252733967972, 0.0, 4.166935551965709, 1.636354028264565, 2.218593541452777, 0.5221417546640358, 0.0, 0.0,
2.007696047466452, 2.148367210948948, 0.8732678202885606, 1.790893605935434], 1.0, "T", ["F", "T"]], [2331, "M", "M", 1.0, 80000.0, "N", 53.213333,
0.0, 0.0, 3.22, 0.0, "CC", "Budget", "Intnl_discount", 3.22, 3.0, [16, [0, 1, 3, 4, 6, 9, 13, 14, 15], [2.1359348967287497, 2.083929036449918, 1.184
5124627849828, 2.6186673867472487, 3.573321824893597, 0.055106406096958216, 2.148367210948948, 0.053155432539303694, 2.686340408903151]], 0.0, "F",
["F", "T"]]]
```

You completed the lab for Training and Deploying a model using the **IBM Watson Studio** model builder.