Workshop: ICP for Data - Analyze

*May 29, 2018*

*Author: Rui Fan* [*rui.fan@ibm.com*](mailto:rui.fan@ibm.com)

*Reference assets from: Sidney Phoon* [*yfphoon@us.ibm.com*](mailto:yfphoon@us.ibm.com)

**Table of contents**

Contents

Overview

**Required software, access and files**

1. Log into DSXL on ICP for Data (ICP4D)

2. Set up the lab project

3. Define data source connections

4. Verification steps: Read data from remote data sets and local files

**5. Build and save Spark ML model with Juypter Notebook**

5.1 Review and execute the code in TelcoChurn\_SparkML notebook

5.2 Verify that the model is saved under the model section of the project

**6. Test, Batch Score and Evaluate saved model**

6.1 Test the model

6.2 Create a batch score

6.3 Create an evaluation

**7. Deployment: Commit and push project**

**8. Model Management & Deployment**

8.1 Create a Project release

8.2 In the Assets tab, create an online and batch deployment for the deployed Telco\_Churn\_ML\_model

8.3 Launch the project and enable

8.4 Click the online deployment created in step 8.2, and test the model in the API interface.

# **Overview**

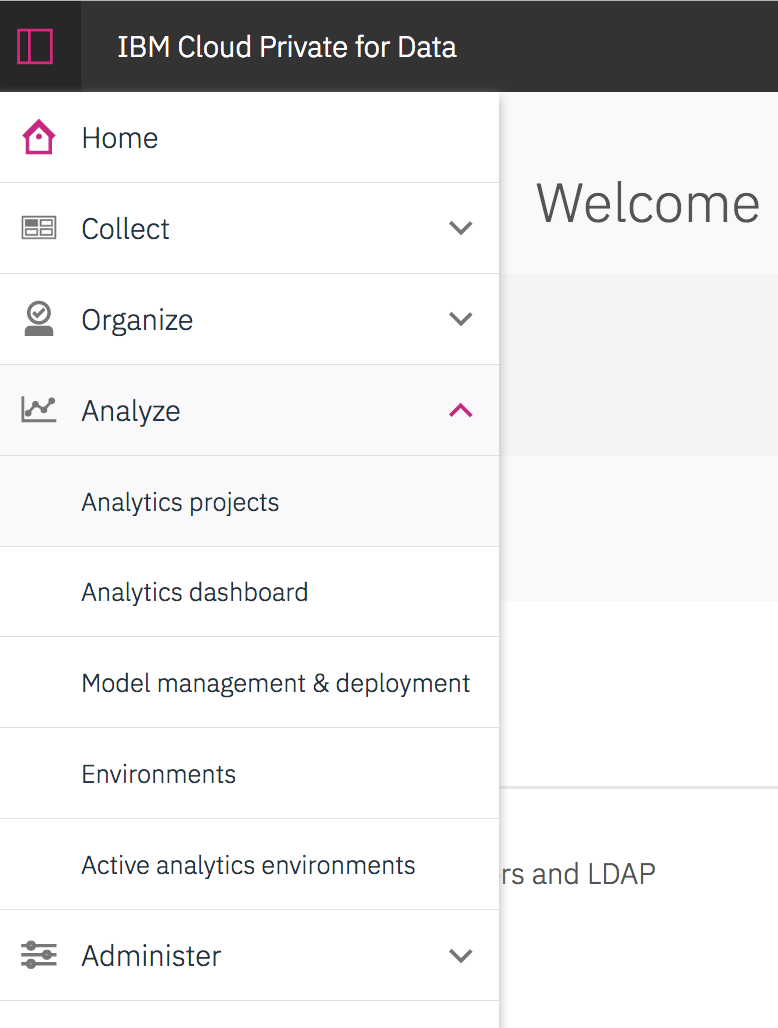
In this lab you will learn how to get access to DSX Local in ICP for Data(ICP4D) to build model and deploy.

# **Required software, access, and files**

* To complete this lab, you will need access to a IBM Cloud Private for Data environment.
* You will also need to download and unzip this GitHub repository: <https://github.com/Rui425/ICP4D-/blob/master/DSXL%20Projects/ICP4D%20Telco%20Churn.zip>

# **1: Log into DSXL on ICP for Data (ICP4D)**

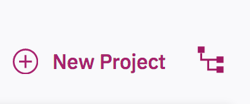
1. Log in ICP4D environment using the given credentials
2. **DSXL** refers to the context of ‘**Analyze’**. Click ‘**Analytics projects**’



You will see a project view of different project you worked on.

# **2: Set up the lab project**

1. Click ‘New Project’ to begin your journal in DSXL for ICP4D.



1. To create this project **‘from file’**:

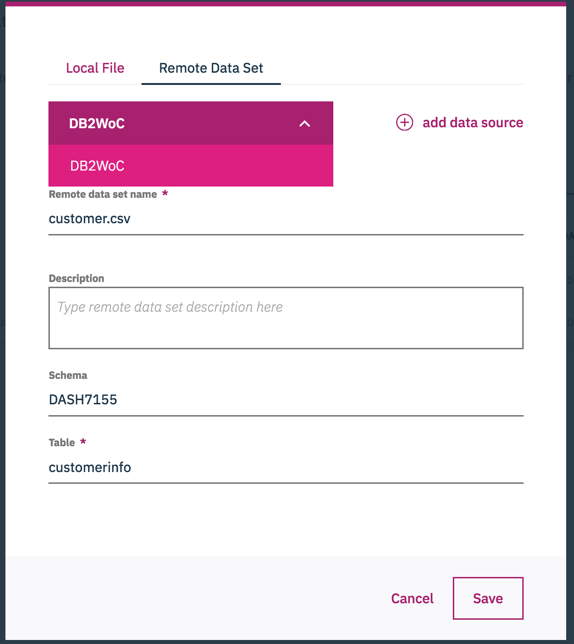
Drag and drop the downloaded .zip file to the ‘**Project file**’, you could give it a name as whatever you want.

1. Click ‘**Create**’, then the project is created.
2. Click ‘**Assets**’ to check for all available assets that are in this project.

# **3: Define data source connections**

In this workshop, we want to make use of data from different sources. After data preparation in the previous section, we are now able to find the well merged data from a remote database connection. So the main purpose of this part is to define a data source connection.

1. Navigate to ‘**data sets**’ and ‘**add data set**’. Click ‘**add data source**’ from ‘**Remote Data Set**’ given that we haven’t get the data source connected.
2. Fill in the blanks with the corresponding data source information. For example, you could name your source name, type in JDBC URL, the username and password for the data source.
3. Click ‘**Create**’. The connection is created. However, the environment doesn’t know which data we want to import in this project, so there are extra steps to import a specific data set.
4. Navigate back to ‘**Remote data set**’ and choose the data source you just added. Type in corresponding schema and table name in this connection and name this file as ‘customer.csv’:



1. Click ‘**Create**’. The connected remote data is listed in the ‘data set’ section.

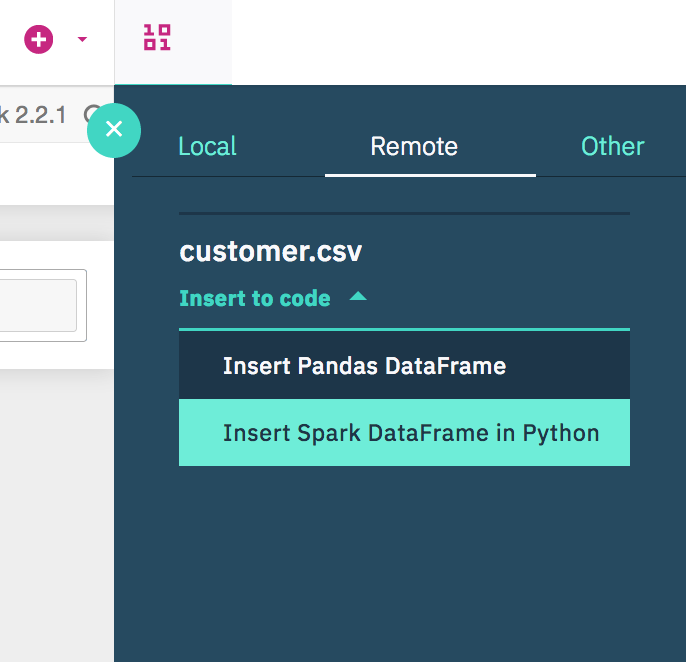
# **4: Verification steps: Create a new notebook to read data from remote data sets and local files**

This will also give you a brief understanding on what is a Notebook and how to load data into a notebook.

1. Under the ‘**Notebooks’** tab, create a new notebook by ‘**add notebook**’. You will create a new blank notebook for this project.
2. Name your notebook and you could choose then environment you want to work on. In this workshop, we will use **Python 3.5**.
3. Click ‘**Create’**
4. **Load remote data set**:

Click the blank in the first cell you see in the notebook. This means that you will work in this cell to run code.

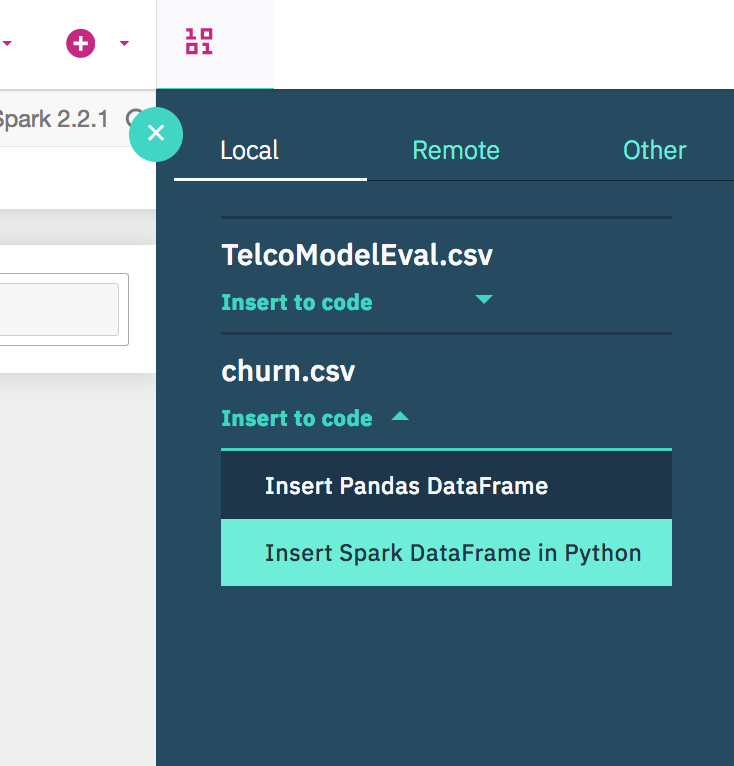
Import the remote dataset in that cell - customer.csv by clicking the '1010' button on the top right, choosing the 'Remote' option and clicking the prepared merged data. When importing the data, you can choose 'insert Spark DataFrame in Python'. You could keep the name by default, or could rename it to 'customer'. But make sure that the name is consistency in the following steps.



Than run the cell. After the kernel runs successfully, you will see the first five rows in the remote dataset.

1. **Load local data file**:

Import the remote dataset in that cell - churn.csv by clicking the '1010' button on the top right, choosing the ‘Local’ option and clicking the churn data. When importing the data, you can choose 'insert Spark DataFrame in Python'. You could keep the name by default, or could rename it to churn.



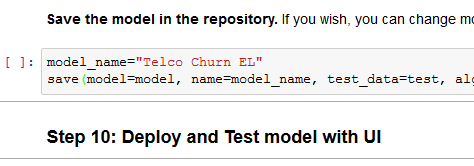
Run the cell and you will see the first five rows in the remote dataset.

**Note**: If you know python, you could also play with it.

# **5: Build and save Spark ML model with Juypter Notebook**

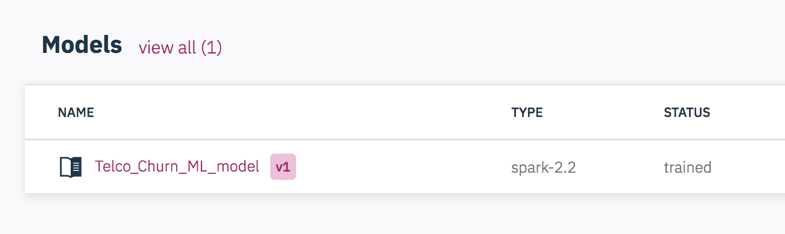
**5.1 Review and execute the code in TelcoChurn\_SparkML notebook**

1. Navigate to the ‘**Notebooks’** tab and open the prepared ‘**TelcoChurn\_SparkML**’ notebook.
2. Run through it and follow the instructions in the notebook. **Note**: in step 2, you will first load the remote data. It’s the same as you’ve learned in the previous section.
3. This time, the model will be created and saved in the ML repository for further use.



**5.2 Verify that the model is saved under the model section of the project**

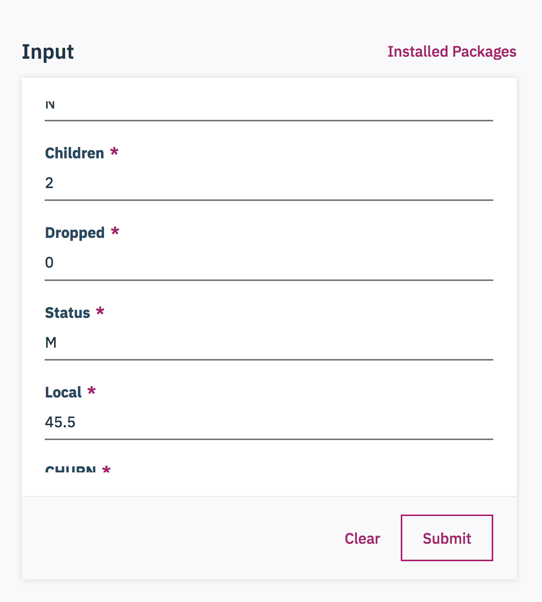
1. Navigate to the **Assets** view. Under ‘**Models**’ tab, make sure that the model is shown. Your model may have a different name and version.



# **6: Test, Batch Score and Evaluate saved model**

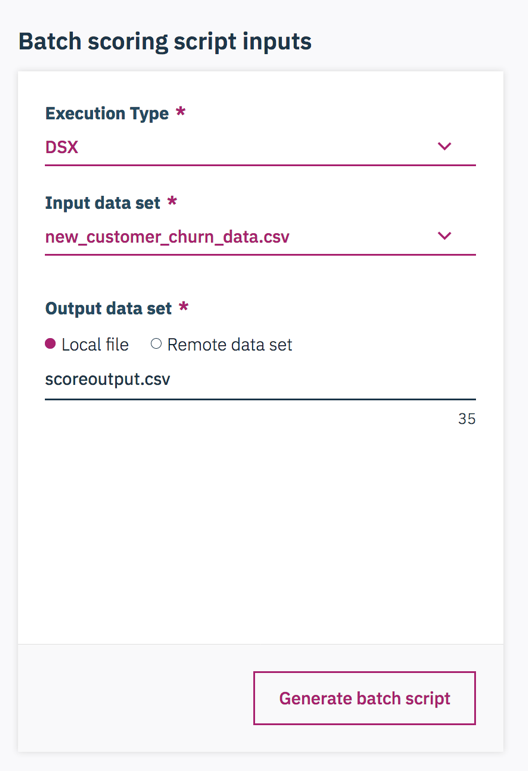
* 1. **Test the model – online**

1. We could test the model buy online scoring. Click the model name under ‘**Models**’ tab then nevigate to ‘**Test’**.
2. There will be automatically some sample online data available in the ‘input’ section. Simply click ‘**submit**’ and the online score will show on the right side: ‘**Result**’ section.

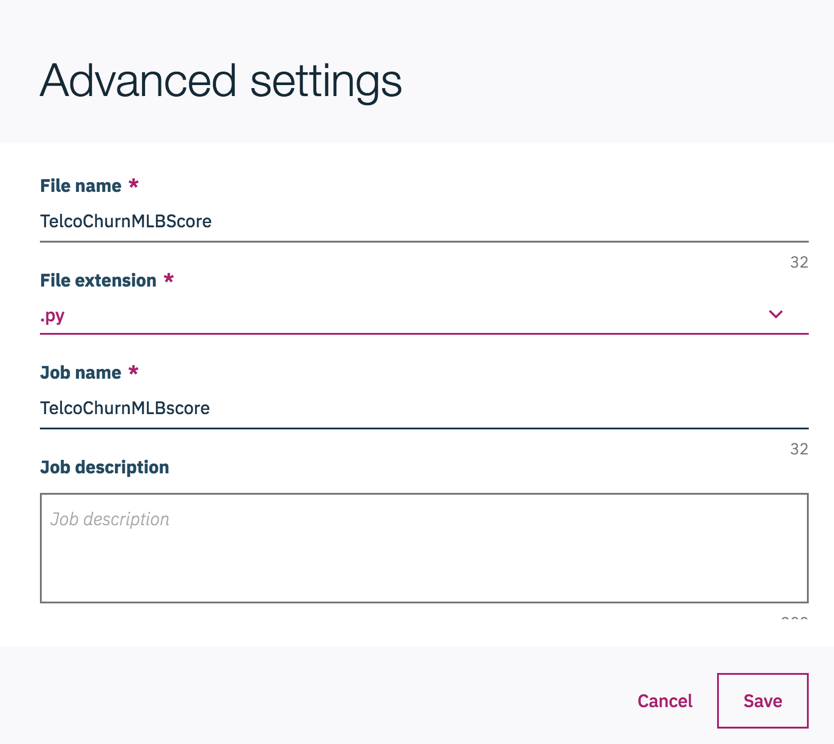


* 1. **Create a batch score**

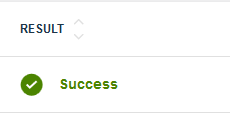
1. Click the model ‘Telco\_Churn\_ML\_Model’ and then nevigate to ‘**Batch Score**’
2. Input data set: new\_customer\_churn\_data.csv (this data set contains data that should be scored)
3. Output data set: output.csv. The output data name can be selfdefined. Make sure to provide ‘**.csv’** extension – otherwise you won’t be able to preview and download the output.



1. Click on ‘**Advanced Settings’** and change the file name to whatever you think is easy to track. You could also change the file into ‘.ipynb’, but ‘.py’ also works. Because your subsequent code changes take place in a python script. Click ‘**Save’.**



1. Click ‘**Generate batch script**’ and ‘**Run now’**. Wait utill the status changes to Success



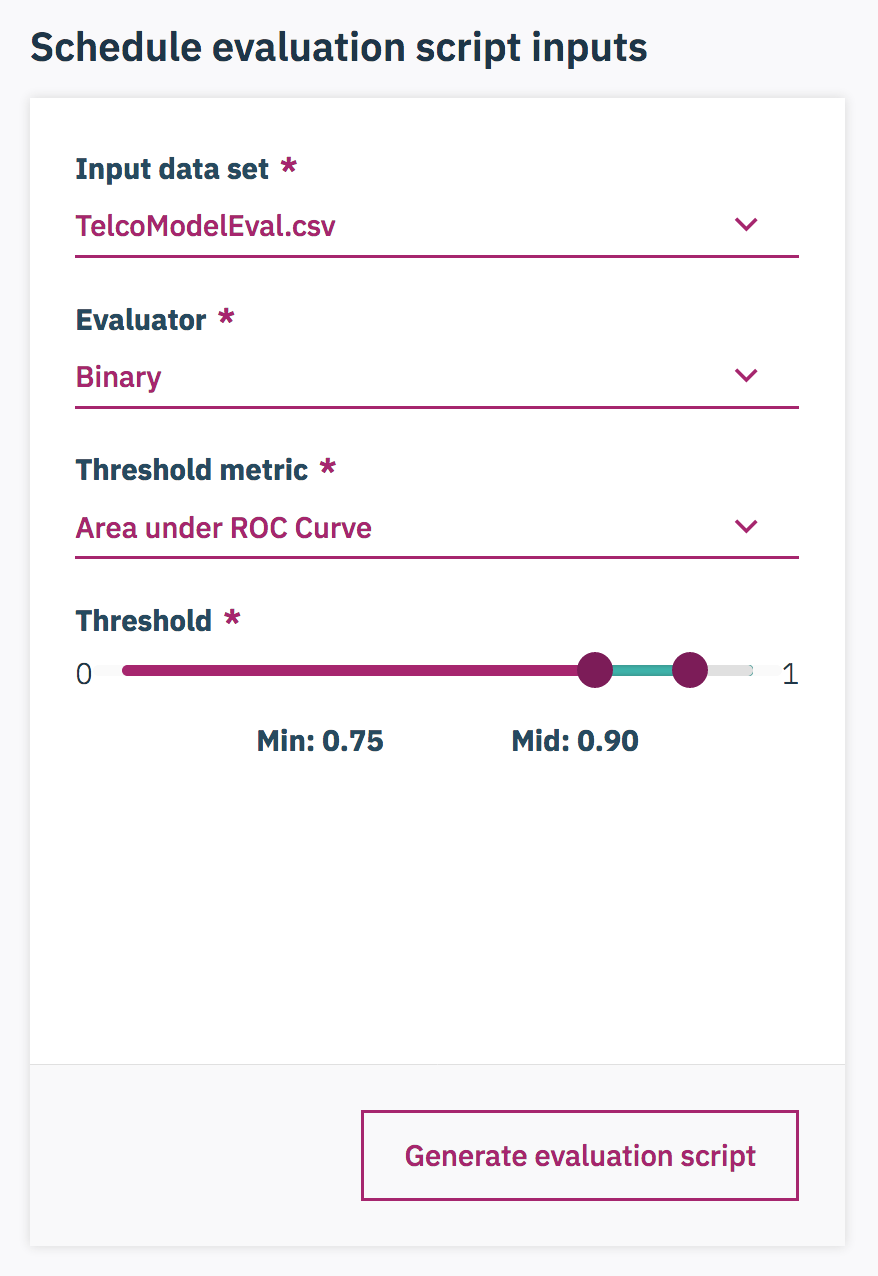
1. Navigate to the **Assets** view of the project. Scroll down to **Data Sets**. You should see the generated ScoringOutput.csv file.



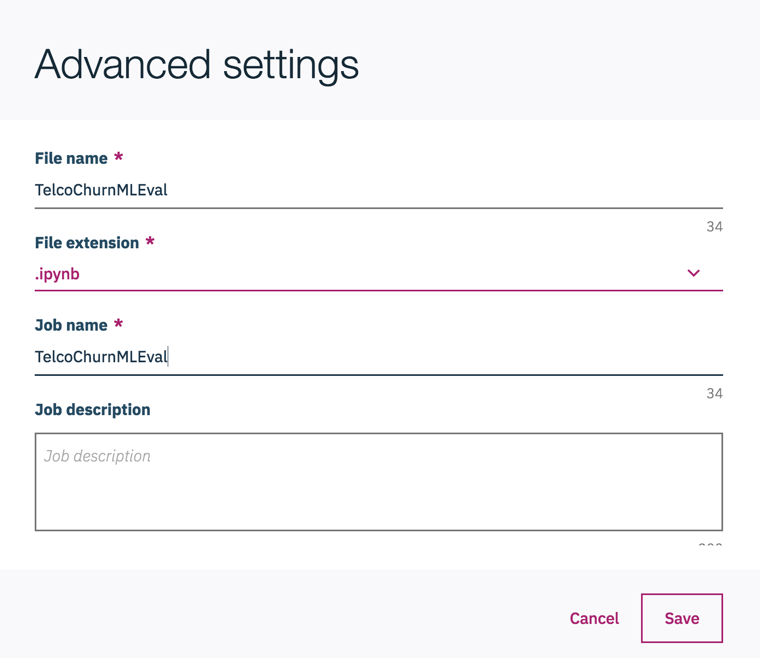
1. You can choose to preview or download the file (click on the ellipses). The scoring results are the last few columns: rawPrediction, probability, prediction, predictedLabel.

**6.3 Create an Evaluation**

1. Nevigate to ‘**Evaluate’**. Select the data source for evaluation (TelcoModelEval.csv file which we generated in a notebook)
2. Pick the evaluation matric. This is binary classfication model so the ‘**Area under ROC Curve**’ is picked. You could define your own threshold.



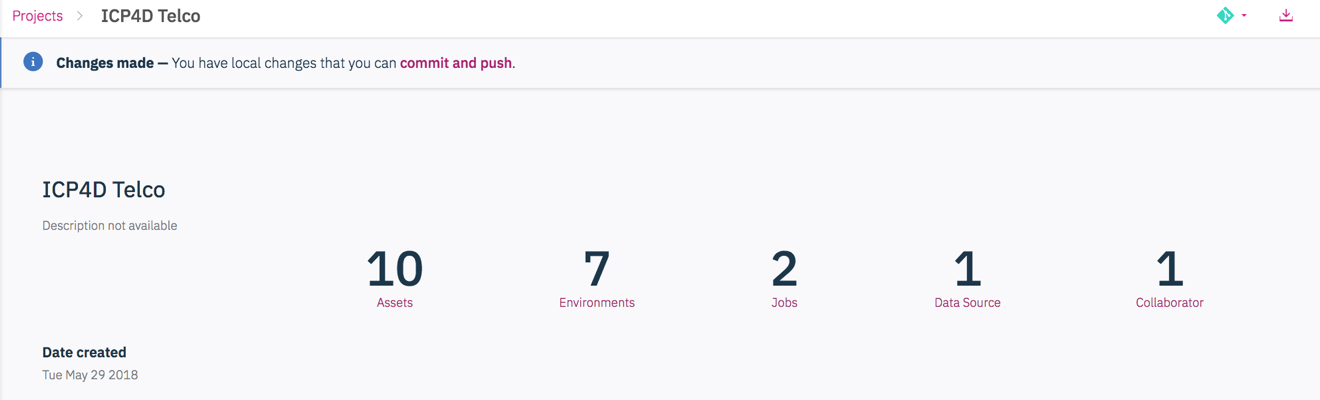
1. Click ‘**Advanced Settings**’ change the file name to: ‘TelcoChurnMLEval’ so that you can keep track of this job. Also you could change the file into ‘.ipynb’. This can make sure that you could check the python script for evaluation. Click ‘**Save’**.



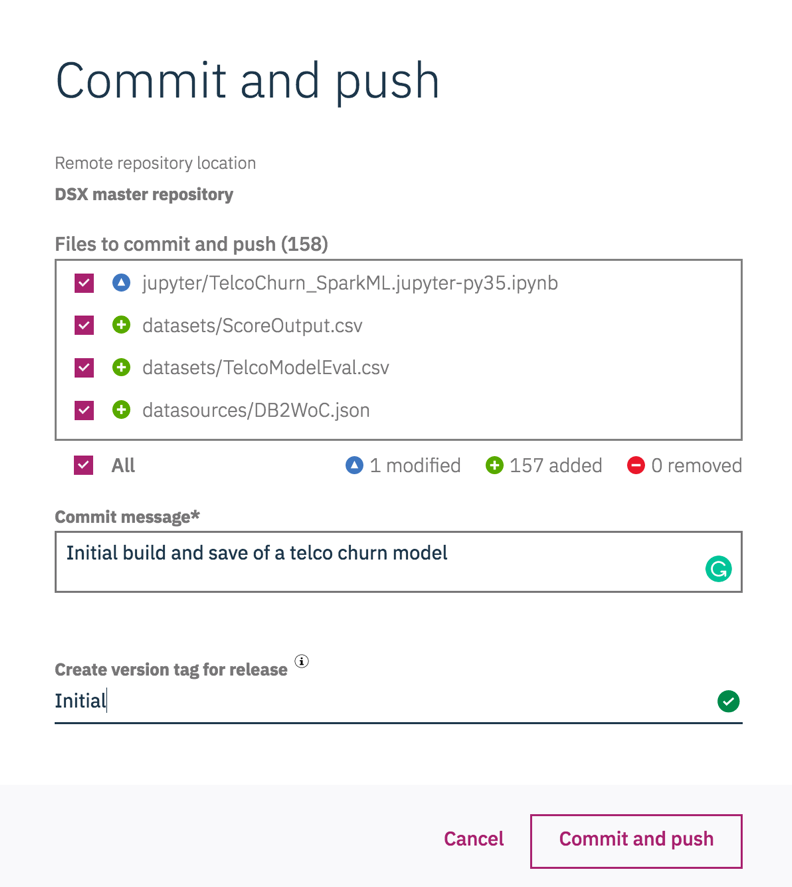
1. Click ‘**Generate Batch Script**’. In the ‘Result’ window, you could see the python code that will do the evaluation. ‘**Run’** and wait till the status change to success.

# 7: Deployment: Commit and push project

1. Go back to the project homepage, you can see the massage that ask to **commit and push** the change of this project in the DSXL enviornment. This can push all of the assets to the model management & development (MMD) segment. So by Clicking the ‘**Commit and Push**’, you can move the change to the MMD enviornment.



1. Then you will see there is a list of the assets that are created in this project. You could put a commitment massage to make note of what is pushed in there by this time. Because this is the first time creating this project, the massage can be: Initial build and save of a telco churn model. Since it is the first released version, you could put ‘Initial’ for the version tag.



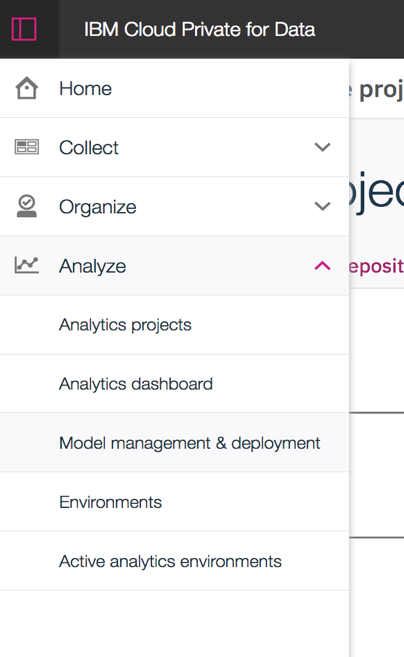
1. Click ‘**Commit and push**’

**Note:** From the analytics projects enviornment, those are all of the basic functionalities. You’ve got data connections, Notebooks, models, batch scoring scripts and evaluation. Then it’s time to go to model management & deployment segment and use all of those to create **automate** **processes**.

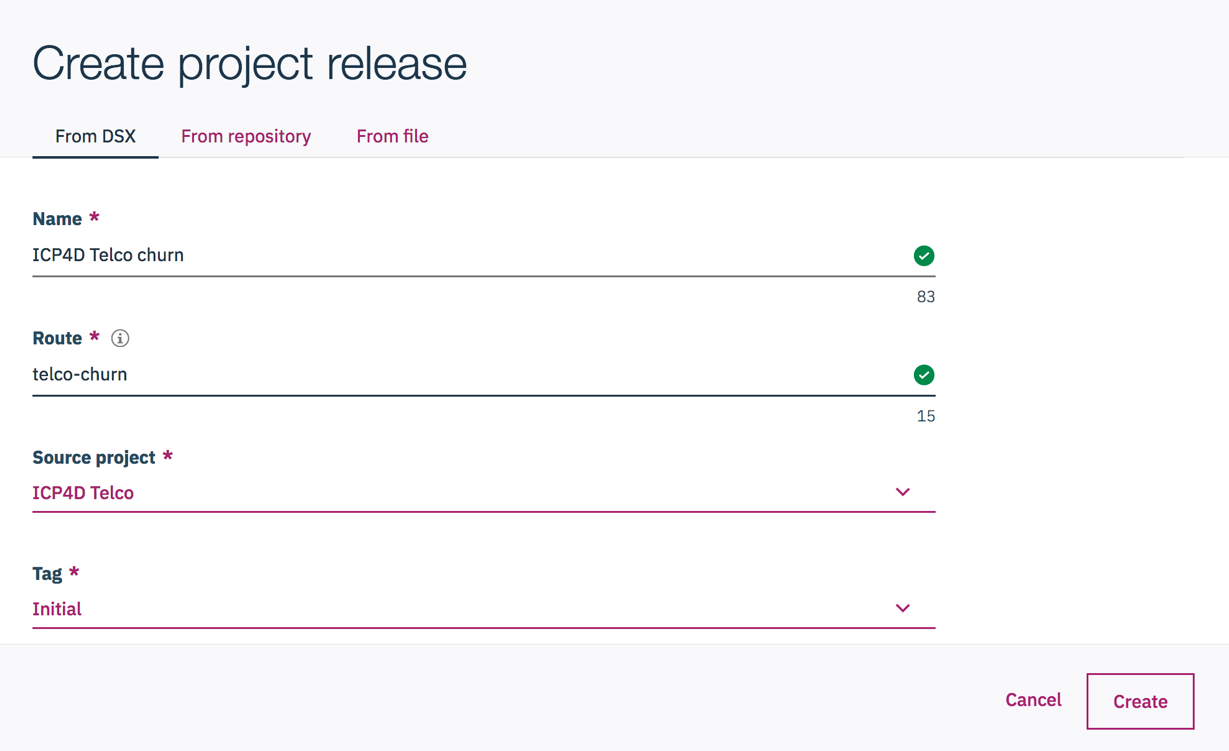
# **8: Model Management & Deployment (MMD)**

**8.1 Create a Project release**

1. Nevigate to **Analyze – Model management & deployment**. This is where you leave the project enviornment and begin deploymeng of different jobs.



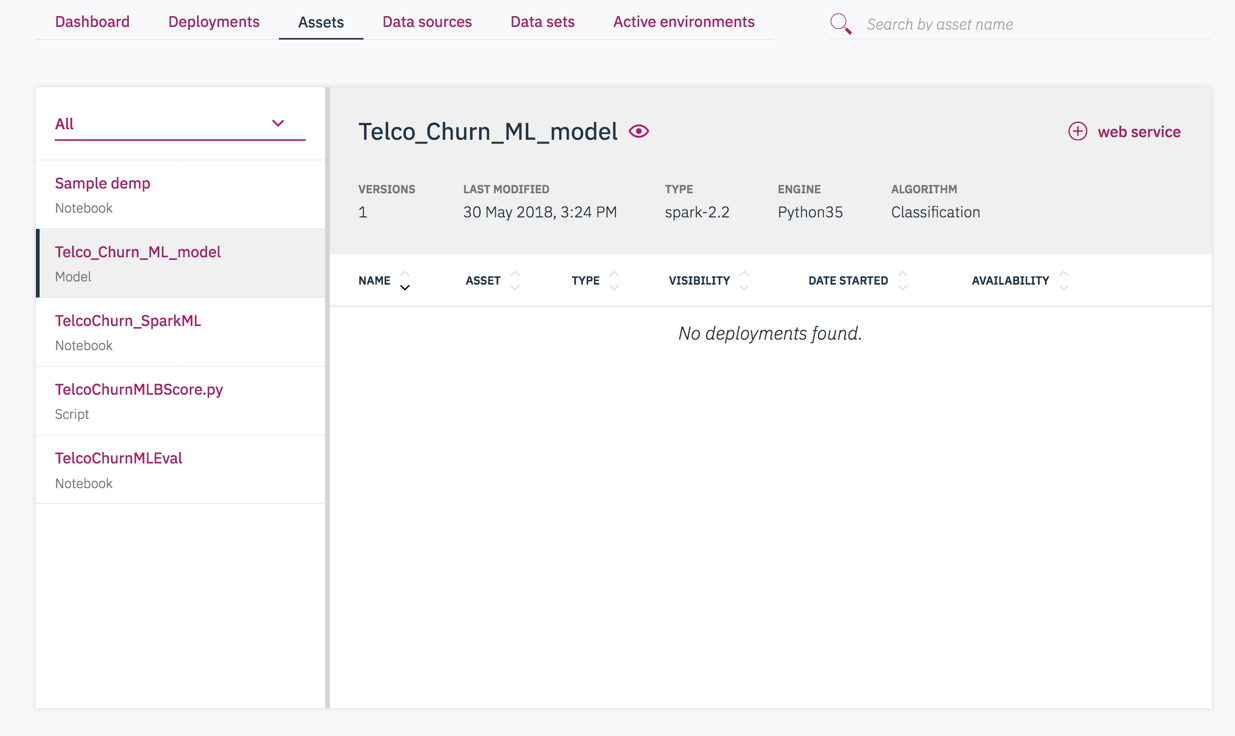
1. By adding a ‘**Project release’**  to create the deployment.
2. Give it a **name** that you could easily track. ‘**Route’** is a unique part of the url that all of these assets are created related with this project. All should be lowercase. Choose the target source project and tag that you want to use. Then ‘**Create’**



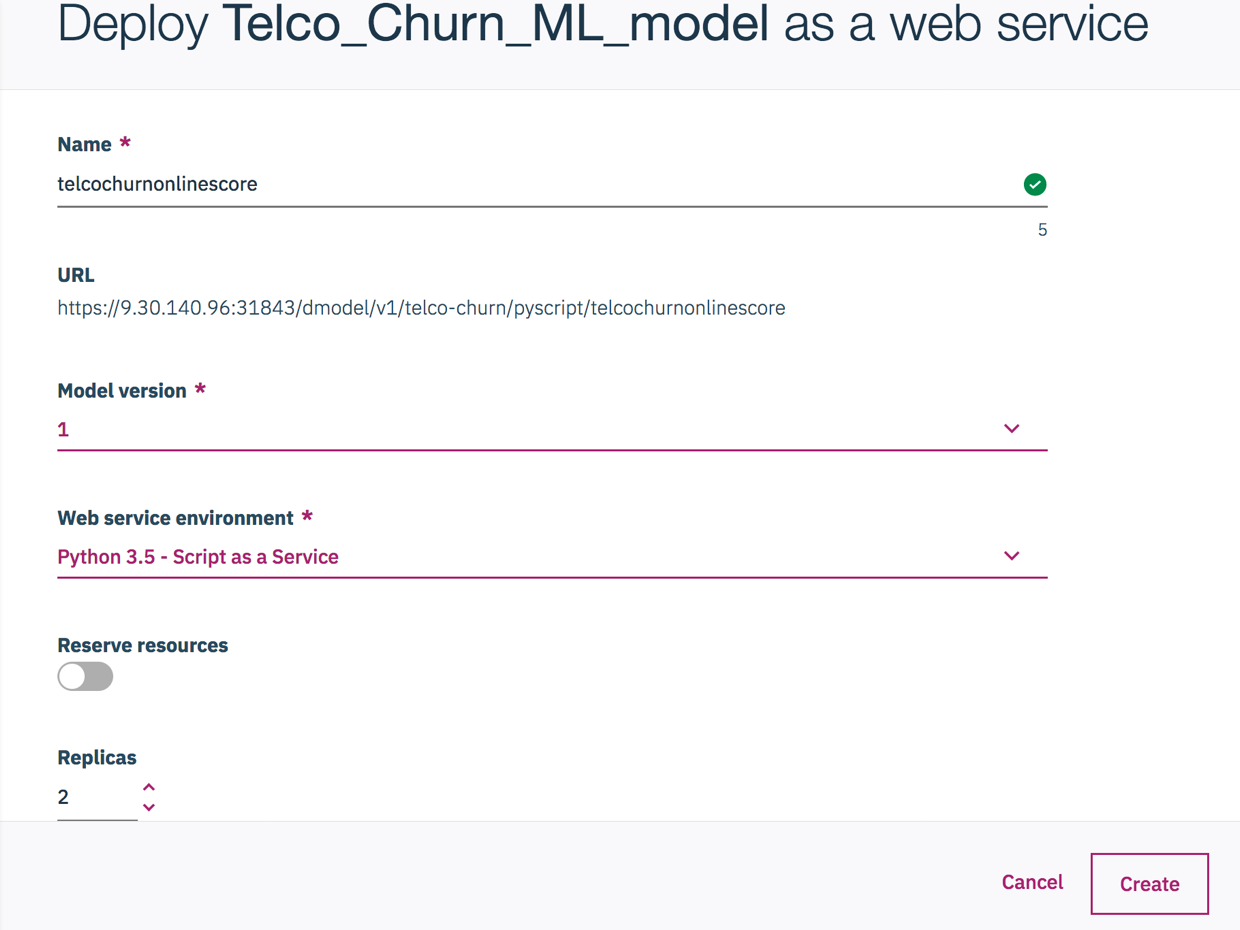
This project release is created.

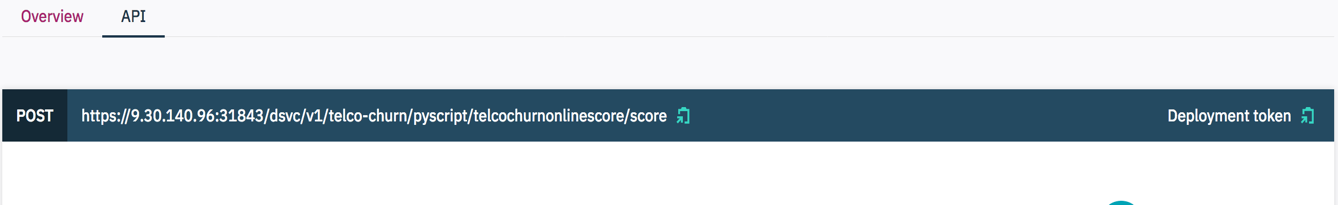
**8.2 Create an online and batch deployment for the deployed** **Telco\_Churn\_ML\_model**

1. In the **assets** tab, you can see all of the analytics assets. There are notebooks, models, and scripts (we created for batch scoring).
2. **Online scoring**: Click the TelcoChurn\_SparkML model and add service by clicking the botton on the top right. This will add a oneline deployment service for this model



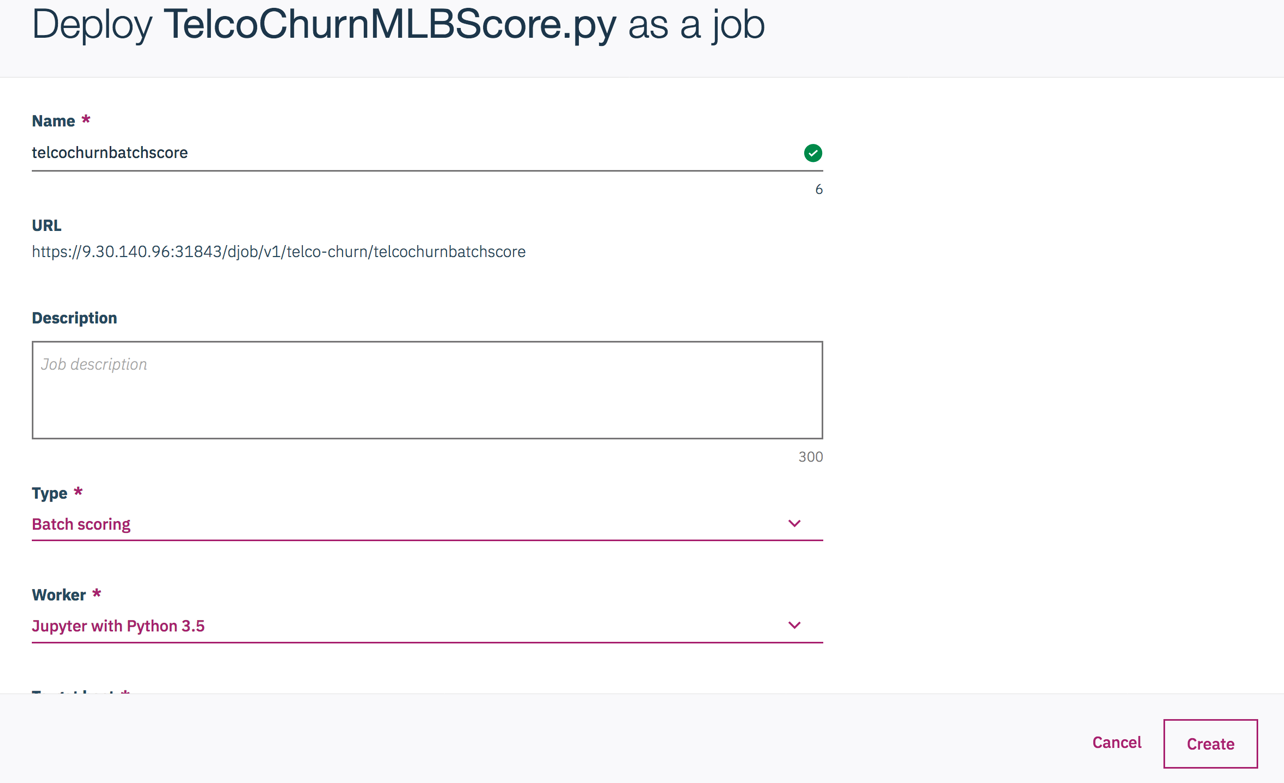
Fill in the information that provided then ‘**Create’**. You could chosse whether you want to reserve resources and how much replicas you want for this job.



**Note**: At this time, the online deployment is created. You could also find the REST API and deployment token under ‘API’ tab. Simply click, the token is copied to your witeboard. 

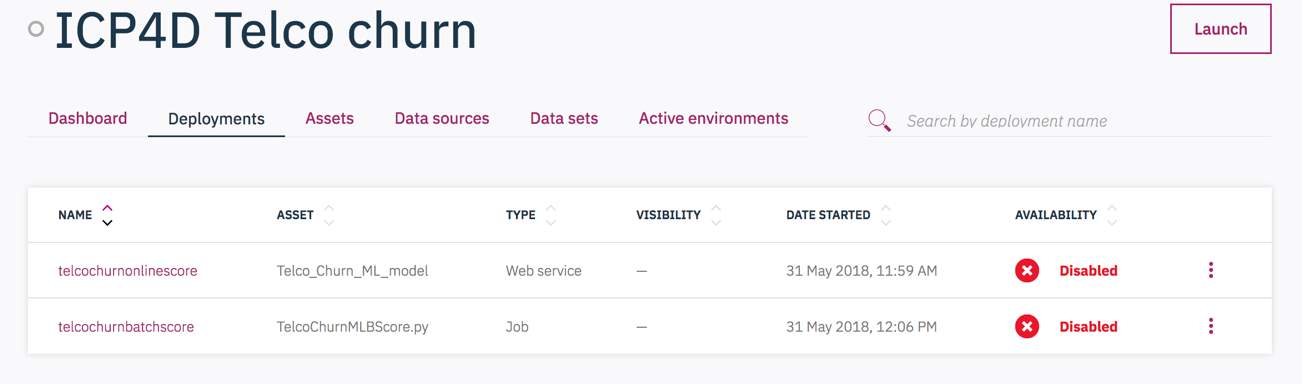
However, the deployment is not active now. We need to lunch and enable it before it could be use. (Follow the instructions in 8.3). At this time, we could create all deployments at once then lunch them together.

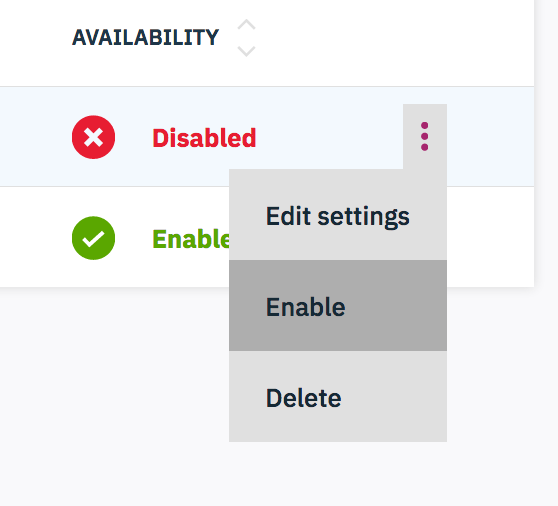
1. **Batch scoring**: Nevigate back to ‘**Assets’** tab. Click the TelcoChurnMLBscore.py – the batch scoring script we’ve created in the project enviornment. Add a job by clicking the ‘**job’** botton on the top right. Type in information that needed – the job **name** and job **type** – **Batch scoring**. We would like to work with a Python3.5 notebook. You could also chosse whether to run on demand and according schedule. Then ‘**Create**’. This will add a batch deployment for this model.



The same as online deplyment, the batch scoring is not active now. We need to lunch enable it before it could be use. Follow the instructions in 8.3.

**8.3 Launch the project and enable**

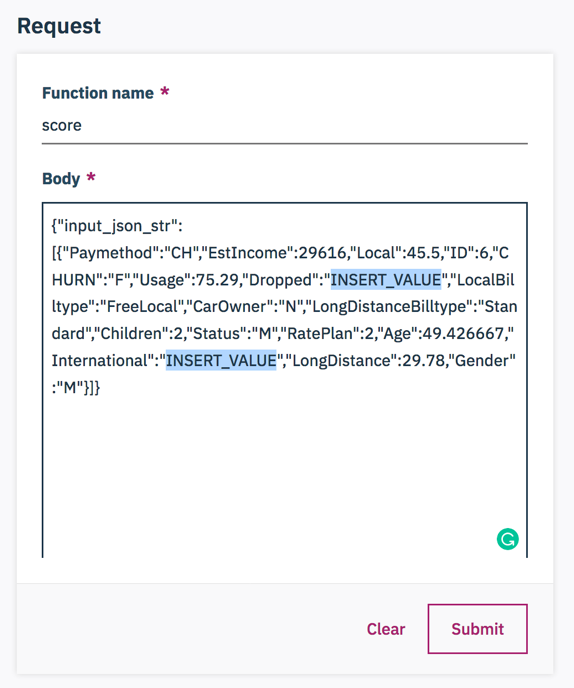
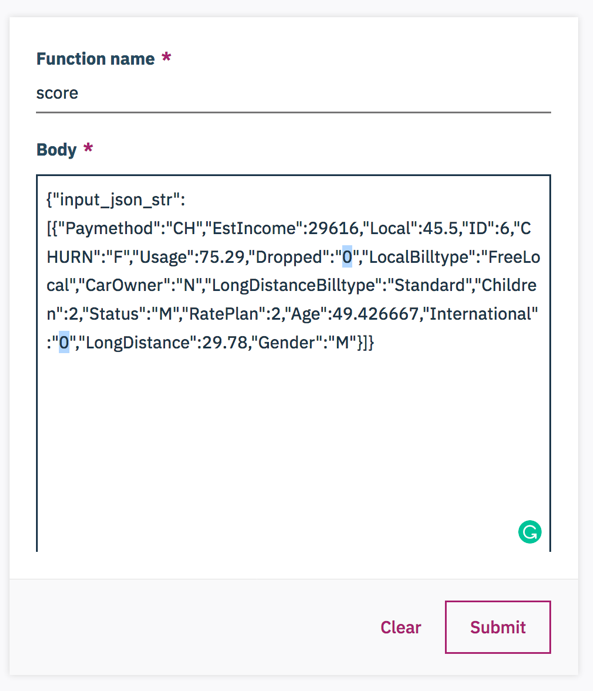
1. Under ‘Deployments’ tab, There are two jobs that just created. However, you will find that they are disabled for now. 
2. Click ‘**Lunch’** on the top right to activate those deployments. This may take few seconds. You will see that the onlinescore job is still disabled. Because there are extra step to enable the online score: Go to right icon and click on ’Enable’. This may takes a little longer. Wait until it changes to ‘enabled’.



**8.4 Click the online deployment created in step 8.2, and test the model in the API interface.**

Now we are good to test the deployment.

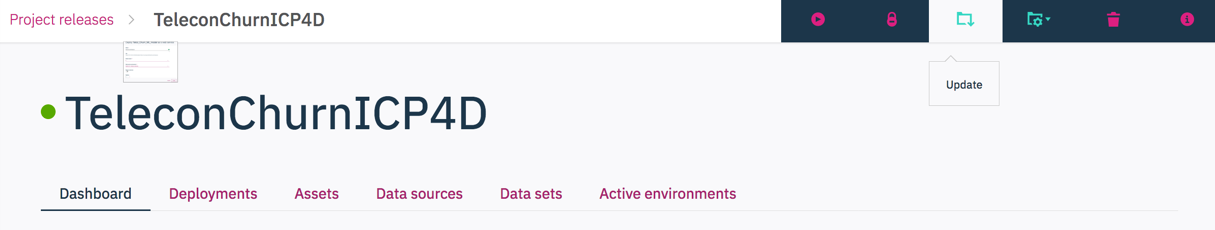
1. Click the oneline deployment created in step 8.2 and enabled in 8.3. Under ‘API’ tab, we can test the model.
2. There are some inputs with ‘INSERT\_VALUE’. Simply change it into a value that makes sense. For example, ‘dropped’ as 0 and ‘International’ as 0.

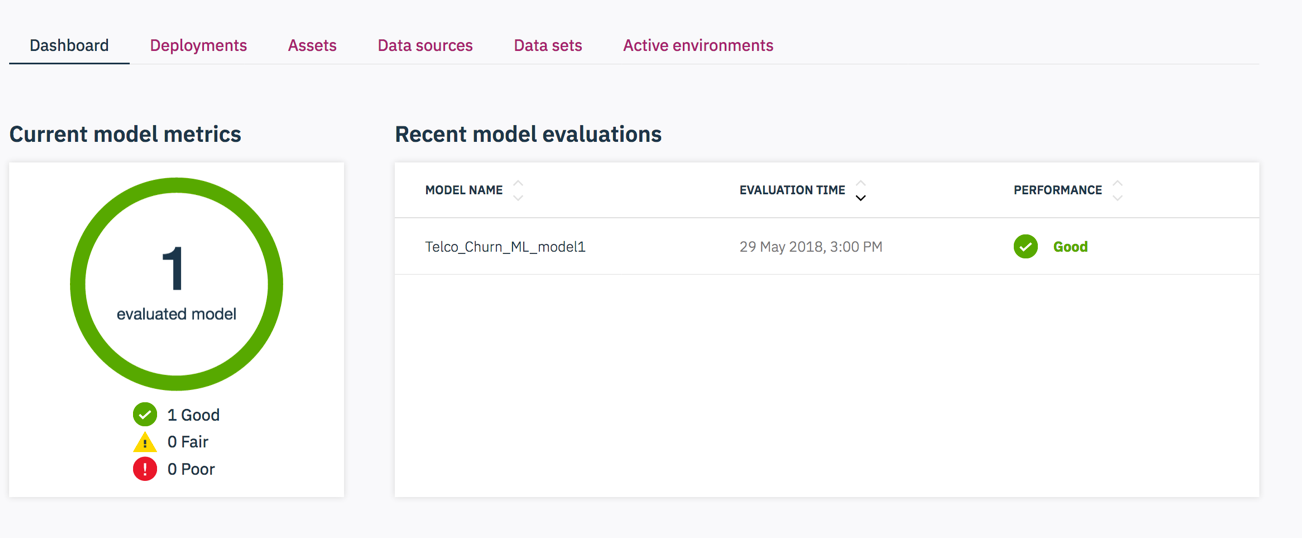
1. Click ‘**Submit’**. The result is shown on right with inputs and several prediction results.

**Note: Here is the end of instructions**

**Additional notice:**

i.Once you made some change to the project and deployment, just update the MMD inviornment and new version of assets are ready. 

ii. Steps for deployment of evaluation is actually the same idea as batch score. You could explore it to see that’s different.

iii.The Dashboard show all of the deployment results. Including the performance of each evaluation.

In this workshop of MMD enviornment, you’ve learned how to create a online and batch score, to enable those job and test those deployments.