Lab: Model Evaluation in DSX

January 18th, 2018

Author: Elena Lowery elowery@us.ibm.com

Table of contents

Contents

Overview

In this lab you will learn how to configure model evaluation in DSX.

Required software, access, and files

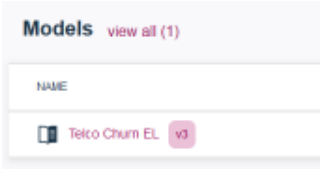
- To complete this lab, you will need access to a DSX Local cluster.
- You will also need to download and unzip this GitHub repository: https://github.com/elenalowery/DSX_Local_Workshop

Part 1: Load the sample project and create model

1. If you haven't already created a project from *DSX_Local_Workshop.zip* file, follow instructions for *Use Case 1* in this repository *https://github.com/elenalowery/DSX_Local_Workshop*

2. If you haven't run through the *TelcoChurn* notebook, run through it so that you generate a model and save the "*Evaluation data set*". The easiest way to do this is to open the notebook, scroll down to **Step 10**, click on it, then in the menu select **Cell -> Run all above**.

Navigate to the **Assets** view and make sure that the model has been created. Your model may have a different name and version.



Note: In our example the "Evaluation data set" is subset of data used for modeling. We chose this approach for convience and demostration. In a production environment the "Evaluation data set" is the new set of historical data that's used to verify that the model is still accurate. This data set can be

automatically uploaded to the data source that's used for evaluation either by a script in DSX or an external process.

Part 2: Test evaluation batch script

1. Click on the ellipses next to the model and select **Evaluate**.

2. Select the data source for evaluation (*TelcoModelEval.csv* file which we generated in a notebook)

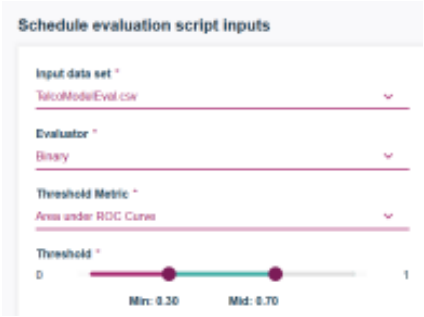
   When we ran evaluation in the notebook we used *BinaryClassificationEvalutor* and *Area Under Roc Curve* as the metric. We suggest that you use the same values when creating the evaluation script.

```
from pyspark.ml.evaluation import BinaryClassificationEvaluator

# Evaluate model
evaluator = BinaryClassificationEvaluator(rawPredictionCol="prediction", labelCol="label", metricName="areaUnderROC")
print 'Area under ROC curve = {:.2f}.'.format(evaluator.evaluate(results))
```

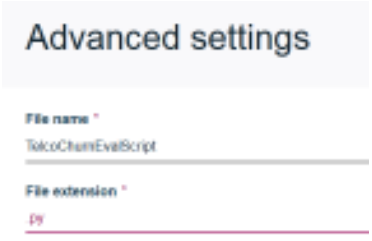
```
Area under ROC curve = 0.92.
```

Keep the default *Threashhold* values.



3. Click on **Advanced Settings** and change the *name* of the script. For example, you can name it *TelcoChurnEvalScript*. Click **Save**.

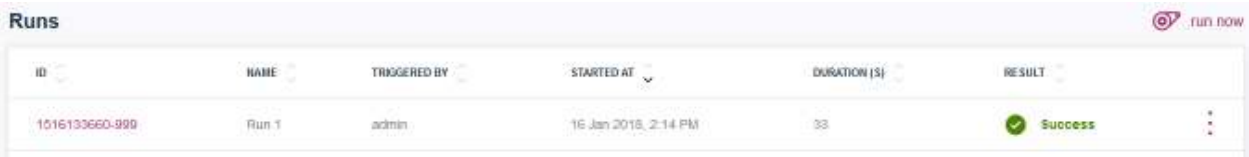
Important: as a temporary workaround, save the script with the .ipynb extention, not .py as shown in the screenshot. This is a bug in the current release.



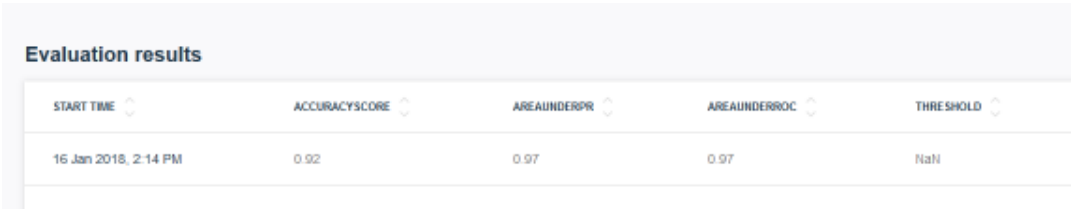
4. Click **Generate Evaluation Script**.

5. Click **Run now**.

6. Scroll down to review the results and wait till the run has finished.



7. If you go back to Project details and click on the model, you can scroll down to the evaluation section and view the results of the run.

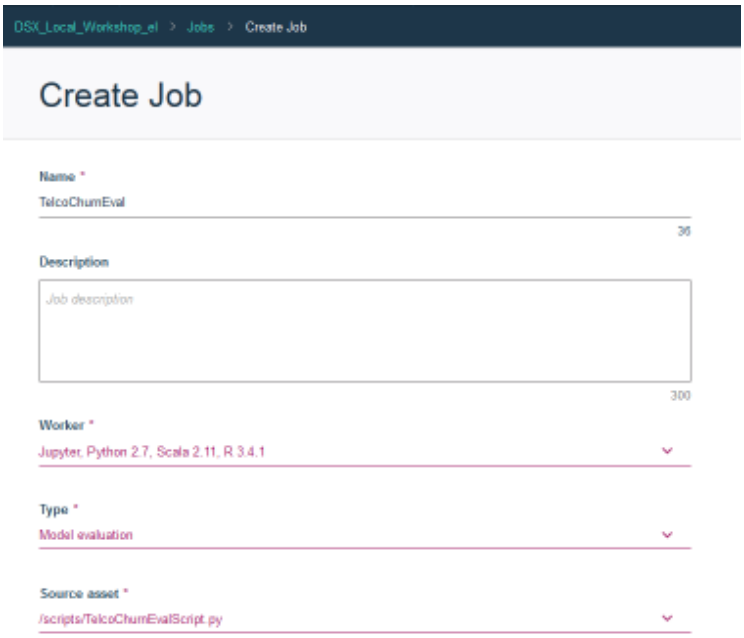


So far we have generated the evaluation batch job script and made sure that it works by running it interactively. Now we can schedule an evaluation batch job.

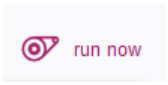
Part 3: Schedule an evaluation batch job

1. In the Project view click on **Jobs**. Click **create job**.

2. On the Create Job screen provide Job name and make sure to select the right Worker (Python 2.x or 3.x) environment (check comments in the notebook or check with the instructor). Select job type *Model Evaluation* and the evaluation script that we crated in one of the previous steps. Scroll down and select either "on demand" or a specific time. Click **Create**.



3. In the Batch Job Details view scroll down and select **Run Now**.



4. Provide *Run name* and click **Run**. After the job is done, you can navigate back to the **Model Details** page to make sure that the scheduled job ran.

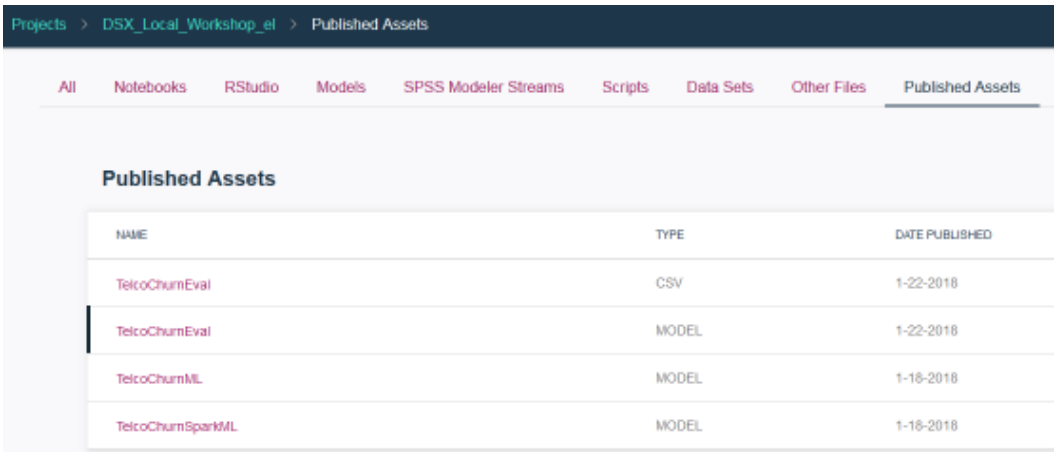
Part 4: Displaying model evaluation on the Model Management dashboard

In order for the model evaluation job to show up in the Model Management dashboard, the model needs to be published.

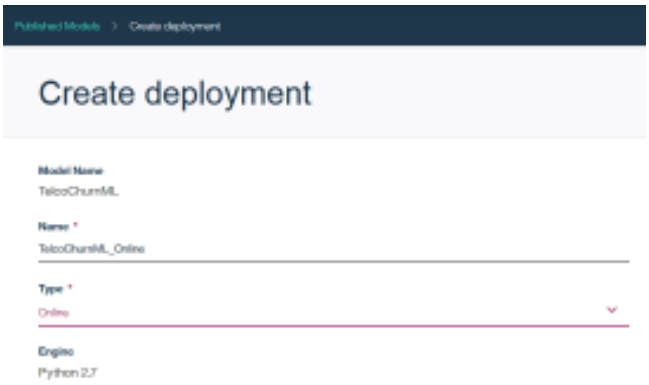
1. In Project details view, navigate to Models. Select **Publish** next to the model. Provide *Published name*. You can publish to the same project (*DSX_Local_Workshop*).

2. Publish *TelcoChurnEval.csv* file to the same project.

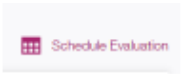
3. In the Project view click on the **Published Assets** tab.



4. Click on the model and in the Model Details view, scroll down and click on **Deploy**.



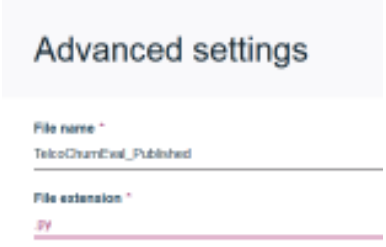
5. In Model Deployment view, scroll down and click **Schedule Evaluation**



6. Select the same values as we did in the unpublished model evaluation.

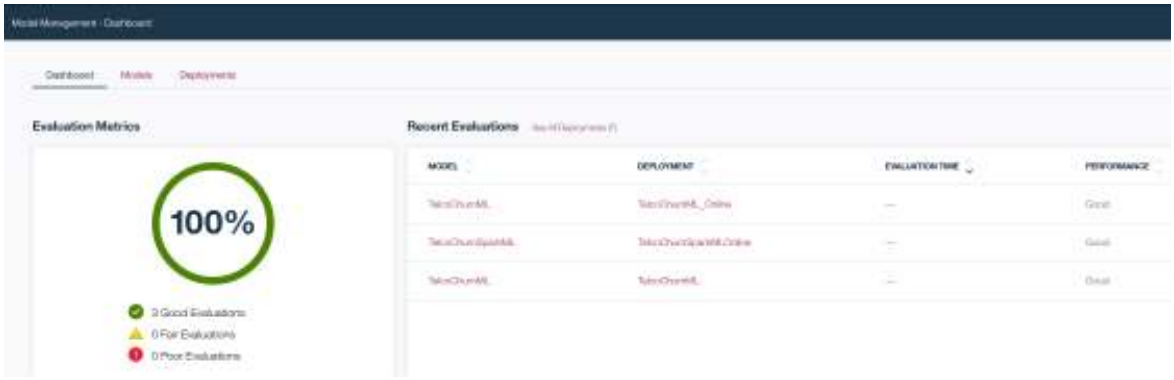
7. Click on **Advanced Settings** and change the name of the script. Click **Save**, **Generate Evaluation Script**, and **Run now**.

Important: as a temporary workaround, save the script with the .ipynb extention, not .py as shown in the screenshot. This is a bug in the current release.



8. Wait till the script is done running.

9. Now when you navigate to the **Model Management** dashboard, it shows model evaluation results.



10. If you wish, you can schedule a job for evaluation of the published model. Schedule the job using the Jobs view in the project (use the script that was generated for the published model).

You have finished working on Model Evaluation in DSX lab.