



Lab: Batch Scoring in DSX

November 30th, 2017

Author: Elena Lowery elowery@us.ibm.com



Table of contents

Contents

Overview 1

Required software, access, and files 1

Part 1: Load the sample project and create model 1

Part 2: Configure batch scoring of a model 2

Part 3: Configure batch scoring of a notebook..... 6

Overview

In this lab you will learn how to configure batch scoring 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

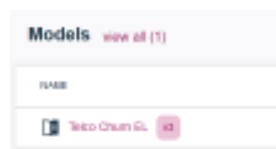
1. If you haven't already created a project from *DSX_Local_Workshop.zip* file, follow instructions for *Use Case 1* in this repository
https://github.com/elenalowery/DSX_Local_Workshop
2. If you haven't run through the *TelcoChurn* notebook, run through it so that you generate a model. The easiest way 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**.

Save the model in the repository. If you wish, you can change the model name.

```
[ ]: model_name="Telco Churn EL"  
save(model=model, name=model_name, test_data=test, alg
```

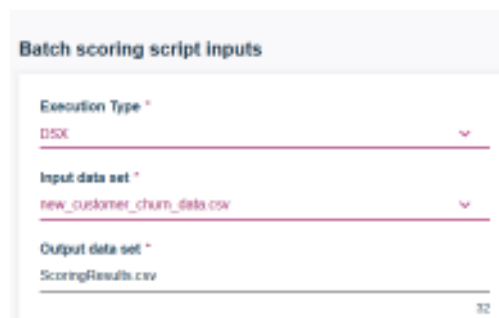
Step 10: Deploy and Test model with UI

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



Part 2: Configure batch scoring of a model

1. Click on the ellipses next to the model and select **Batch Score**.
2. Fill out the required fields:
 - Input data set: *new_customer_churn_data.csv* (this data set contains data that should be scored)
 - Output data set: *ScoringResults.csv*. Make sure to provide **.csv** extension – otherwise you won't be able to preview and download the output.




Batch scoring script inputs

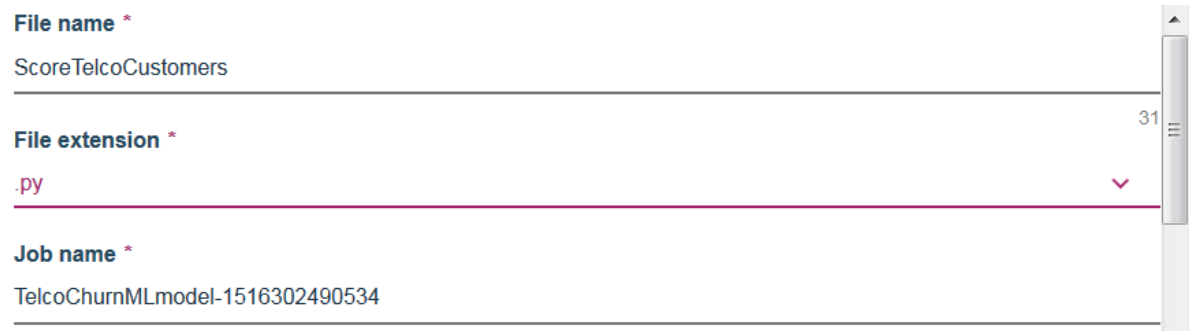
Execution Type *	DSX
Input data set *	new_customer_churn_data.csv
Output data set *	ScoringResults.csv

32

3. Click on **Advanced Settings** and change the file name to *ScoreTelcoCustomers*. Click **Save**.

 Advanced settings

Advanced settings



File name *	ScoreTelcoCustomers
File extension *	.py
Job name *	TelcoChurnMLmodel-1516302490534

31

- Click **Generate Batch Script**. Then click **Run now** and wait till the status changes to *Success*.

RESULT

Success

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

Data Sets view all (8) add data set				
NAME	TYPE	SIZE	LAST MODIFIED	DATA SOURCE
ScoringResults	CSV	2.94 KB	12-30-2017	Local File

- 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*.

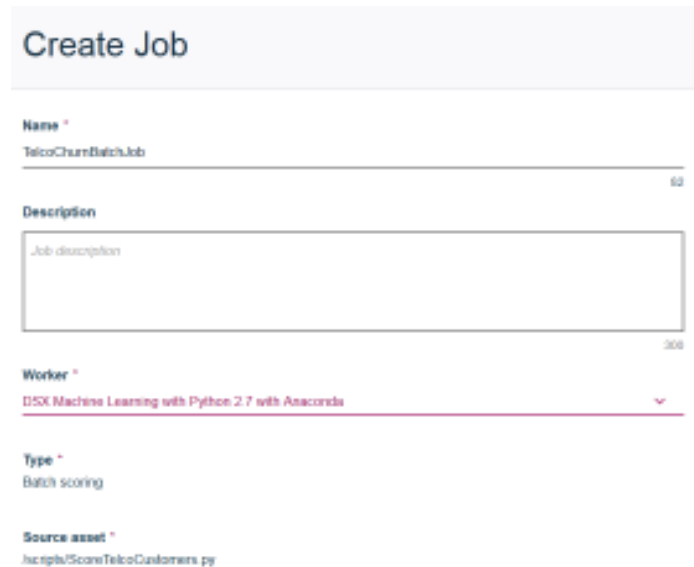
Preview - ScoringResults.csv

	rawPrediction	probability	prediction	predictedLabel
[0.0, 0.40, 40, 1.0]	[4.37364611538, 15.6263538846]	[0.218682305769, 0.781317694231]	1.0	T
[1.0, 15.02, 2.0]	[14.9838754954, 5.01612450462]	[0.749193774769, 0.250806225231]	0.0	F
[0.0, 73.81, 3.0]	[16.4739854053, 3.52601459468]	[0.823699270265, 0.176300729734]	0.0	F
[2.42, 2.0]	[18.7601048665, 1.2198951335]	[0.939005243325, 0.0609947566751]	0.0	F
[0.0, 0.62, 87, 1.0]	[2.83702210032, 17.1629768997]	[0.1418251155016, 0.858148844984]	1.0	T
[0.0, 47.99, 4.0]	[19.1259629083, 0.874037991729]	[0.956295100414, 0.0437016995864]	0.0	F

Close

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

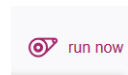
7. In the Project view scroll down to **Scripts** and click **Create Job** next to the script we created in previous steps.
8. On the **Create Job** screen provide *Job name* and make sure to select the right **Worker** (*Python 2.x or 3.x*) environment (check comments in the notebook or check with the instructor). Scroll down and select either "on demand" or a specific time. Click **Create**.



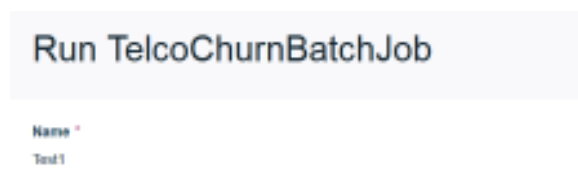
The screenshot shows the 'Create Job' form with the following fields:

- Name ***: TelcoChurnBatchJob (62 characters)
- Description**: Job description (308 characters)
- Worker ***: DSX Machine Learning with Python 2.7 with Anaconda (dropdown menu)
- Type ***: Batch scoring
- Source asset ***: ./scripts/ScoreTelcoCustomers.py

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



10. Provide Run name and click **Run**.




The screenshot shows the 'Run TelcoChurnBatchJob' form with the following fields:

- Name ***: Test1

11. Job status is displayed. Wait till status is "Success".

12. Navigate back to the Project view, and scroll down to **data sets**. Each batch job run overwrites the previous output file.

Data Sets view all (9)			
NAME	TYPE	SIZE	LAST MODIFIED
 ScoringResults	CSV	2.22 KB	01-18-2018

13. If you want to make sure that the file was updated, open the *RunSystemCommands* notebook and run the commands (check the timestamp on *ScoringResults.csv*).

Part 3: Configure batch scoring of a notebook

1. Open *TelcoChurn_SparkML* notebook and delete everything after **Step 11** (REST API call). These cells won't run in batch mode.

Step 11: Test model with a REST API call (Optional)

This step demonstrates an "internal REST API" call to test the model (for an require authentication).

```
] : json_payload = [{
```

*Note: If you want to make a copy of this notebook, don't use the **Make a copy** function in Jupyter (it's currently not working). Download the notebook, rename it, and add it to a project again.*

2. Navigate to the Project view then click on **Jobs**.
3. Click **create job**.
4. Enter Job information. Make sure to select **Worker** that matches version of Python in the notebook. Click **Create**.

Create Job

Name *

TelcoChurnNotebook

32

Description

Job description

300

Worker *

Jupyter, Python 2.7, Scala 2.11, R 3.4.1

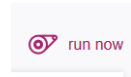
Type *

Notebook run

Source asset *

/jupyter/TelcoChurn_SparkML.jupyter.ipynb

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



6. Provide Run name and click **Run**.

7. Job status is displayed.



8. Navigate back to the **Batch Job Details** view to verify that the job ran successfully. You can also verify that the notebook ran by checking the version of the model in the **Project** view – it should be incremented.

You have finished working on **Batch Scoring in DSX lab**.