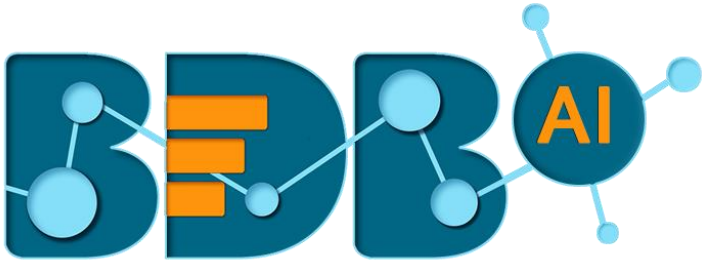


How to Guide

ML Model Runners (DP)

Version: Release 1.1

Contents

Component Description-

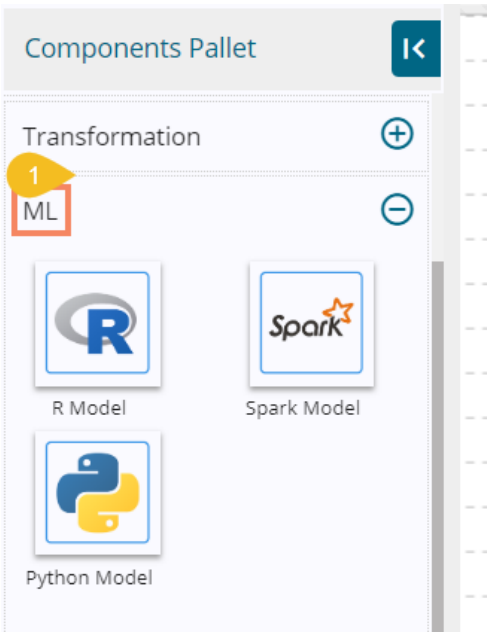
ML Model Runner Components allow us to use the models created on R, Spark and Python Workspaces predictive workbench inside the pipeline.

1. R Model Runner

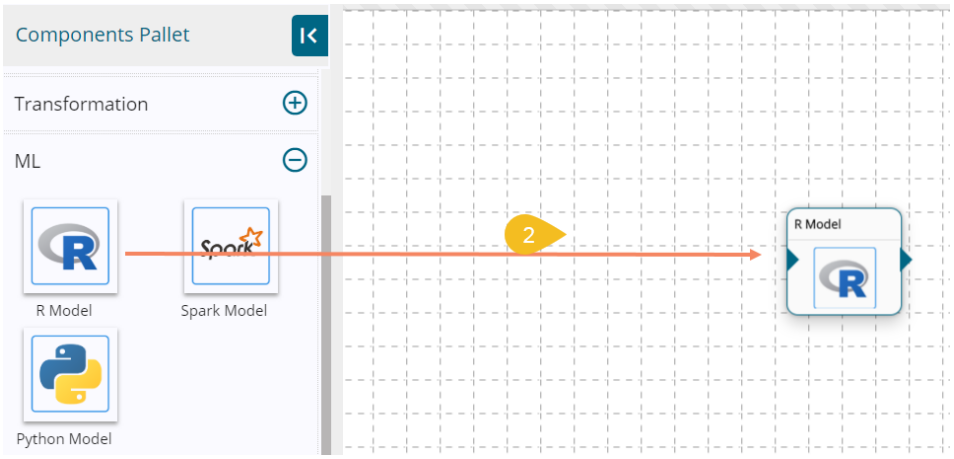
1) Navigate to the Pipeline Workflow Editor and expand the ML Model Runner section in the Components Pallet

   There are 3 ML model (runner)components available in this section:

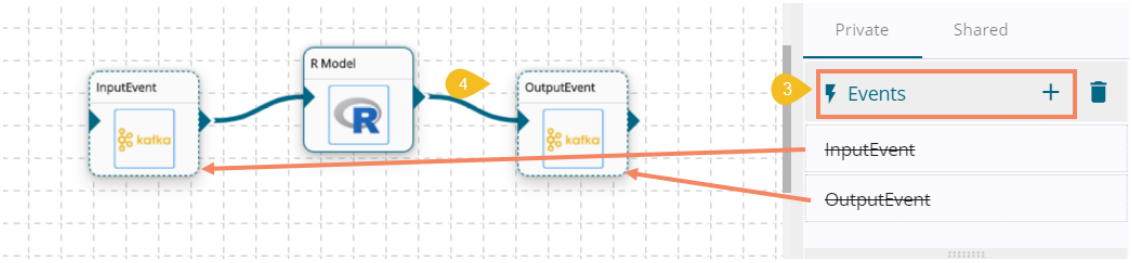
   1. R Model
   2. Spark Model
   3. Python Model



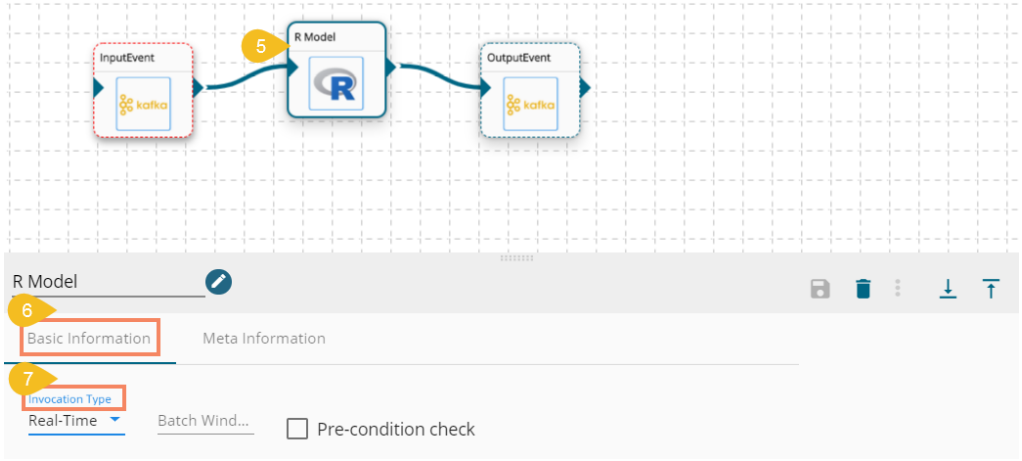
2) Drag and Drop R Model (runner) component to the workflow



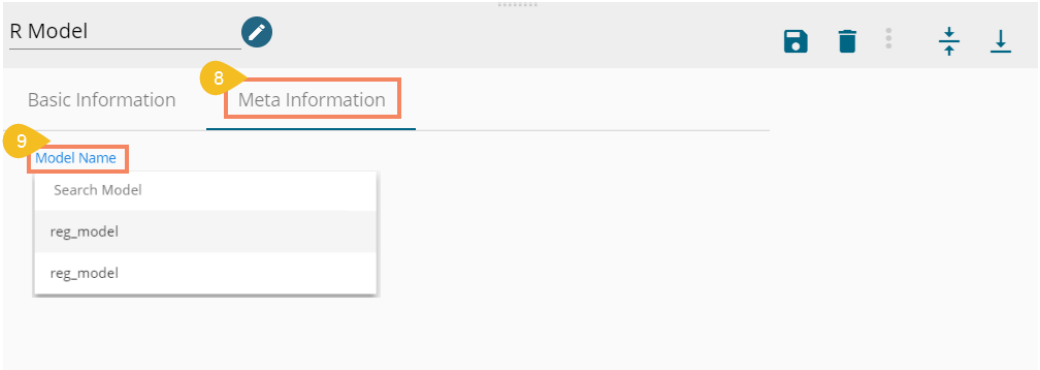
3) Model Runner requires data input from an Event and sends the processed data to another Event, so create two Events.
4) Drag them to the workflow editor and connect with the dragged R Model (runner) component as displayed below:
The data in Input Event can come from any Ingestion, Reader or shared events.



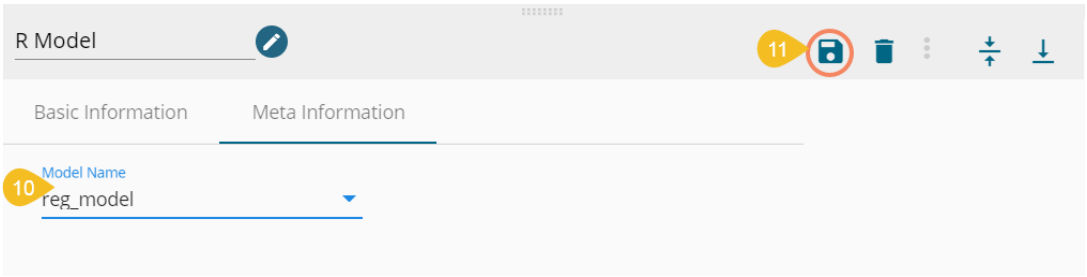
5) Click on the R Model Component to get the configure tabs.
6) The Basic Information tab opens by default.
7) Choose an Invocation Type from the drop-down menu
   a. Real-Time
   b. Batch
   (Note: currently Pipeline supports only Real-time option)



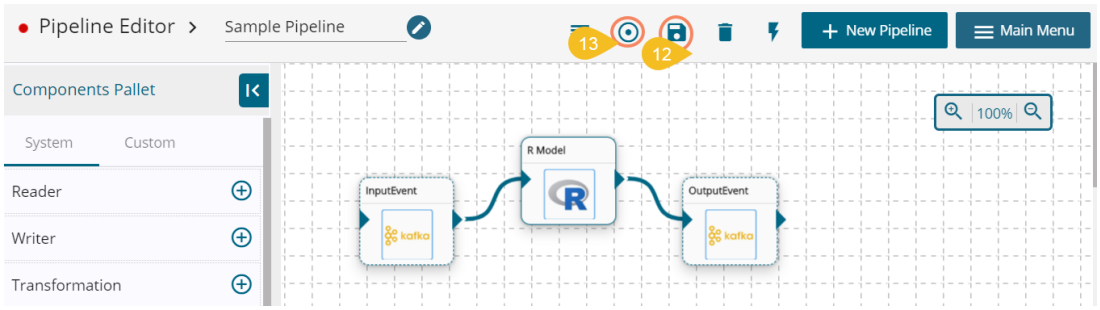
8) Click the Meta Information tab to open.
9) All the exported predictive models appear under the 'Model Name' menu.



10) Search and select the model you want to use.
11) Save the R Model component.



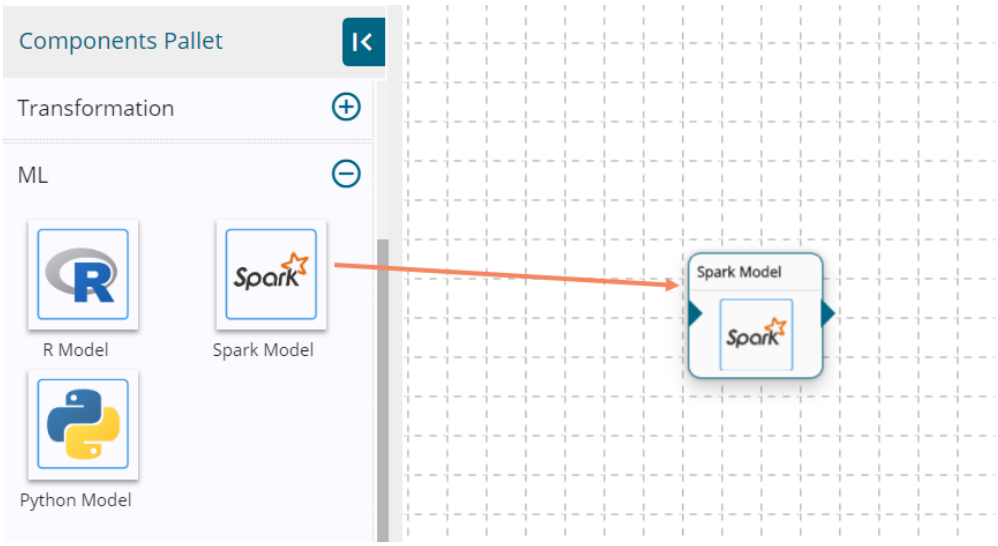
12) Save the pipeline by using the 'Update Pipeline' icon.
13) Activate the pipeline by clicking the 'Activate Pipeline' icon.



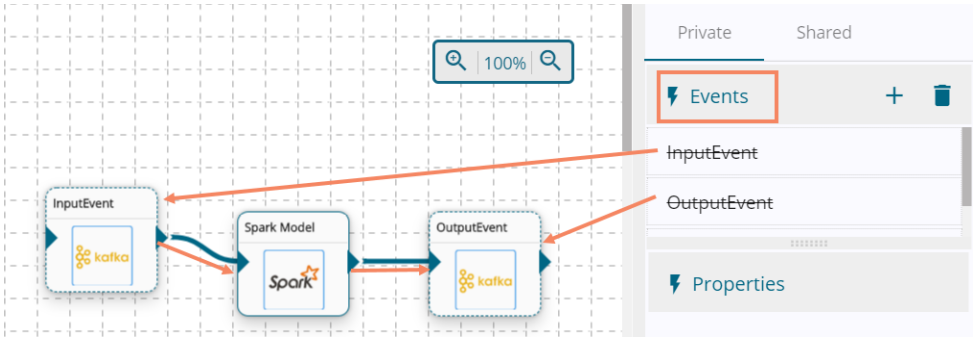
14) The R Model runner component reads the data coming to an input event, runs the model, and gives the output data with predicted columns to the output event.

2. Spark Model Runner

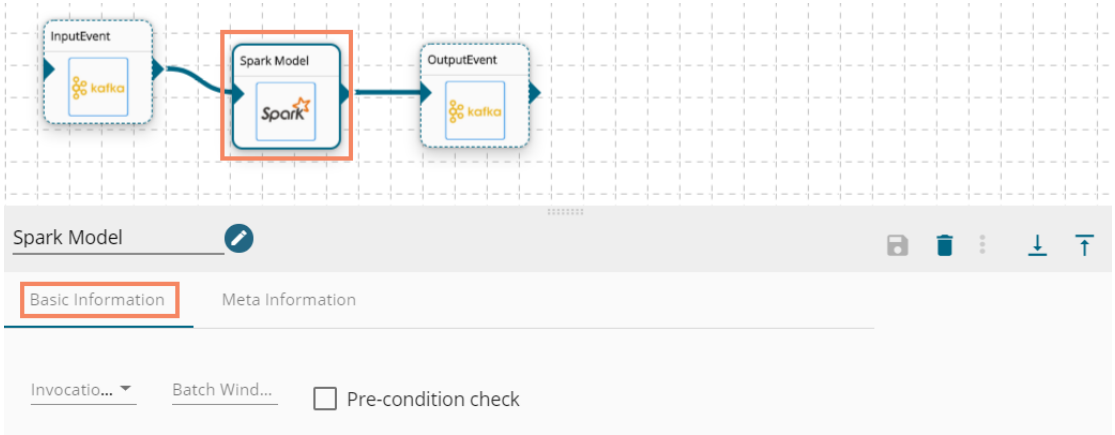
1) Navigate to the Pipeline Workflow Editor and Expand the ML section from the Components Pallet
2) Drag and drop the Spark Model (runner) component to the workspace



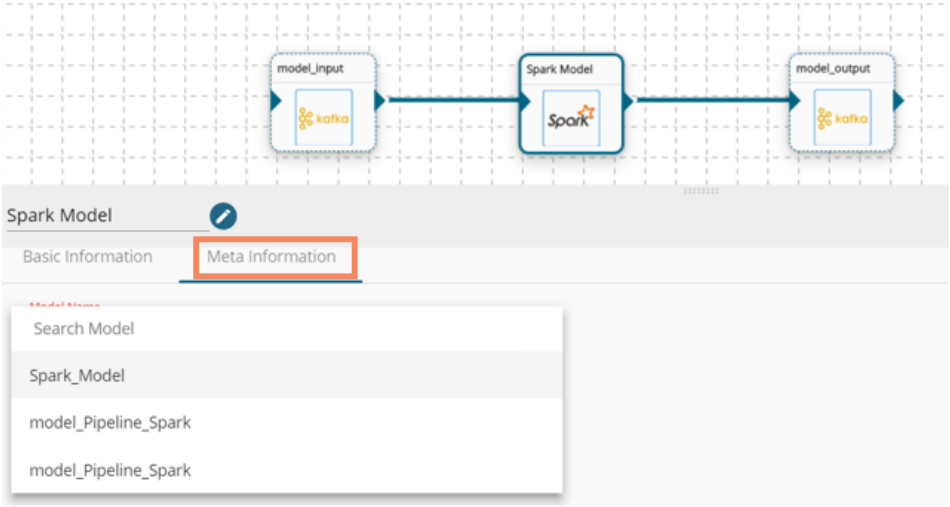
3) The Spark Model runner requires input data from an Event and sends the processed data to another Event. So, create two Events and drag them onto the Workspace.
4) Connect the input and output events with the Spark Model runner component as displayed below.
5) The data in the input event can come from any Ingestion, Readers or shared events.



6) Click the Spark Model runner component to get the configuration tabs.
7) The Basic Information tab opens by default
8) Choose an Invocation Type from the drop-down menu (Real-time/Batch)



9) Select Meta Information and click on Model name field.
10) All the exported predictive models list in a menu.

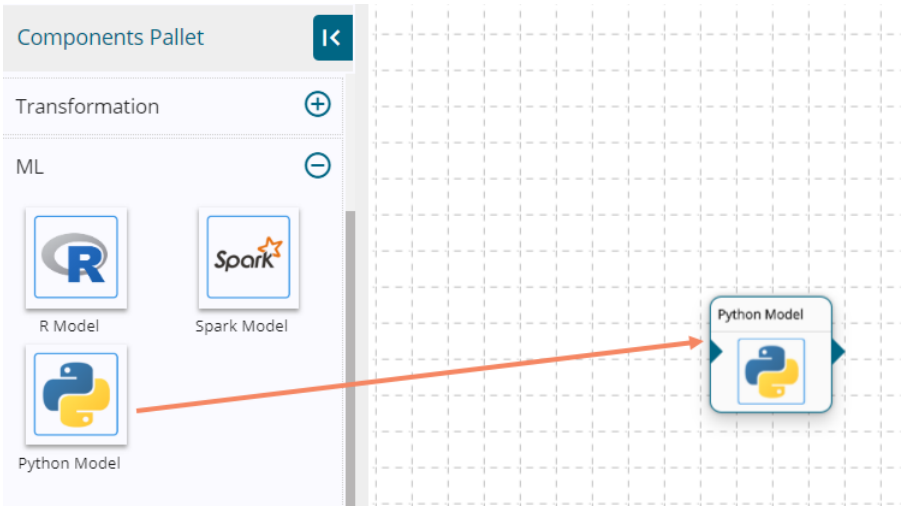


11) Search the model you want to use and select the model.
12) Save the Spark Model runner component.

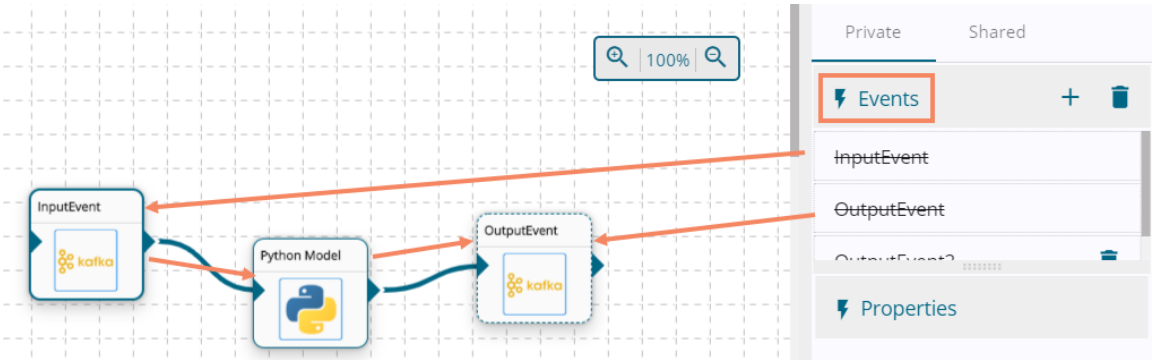
13) Save the pipeline and activate it.
14) The Spark Model runner component reads the data coming to input event, runs the model, and gives the output data with predicted columns to the output event.

3. Python Model Runner

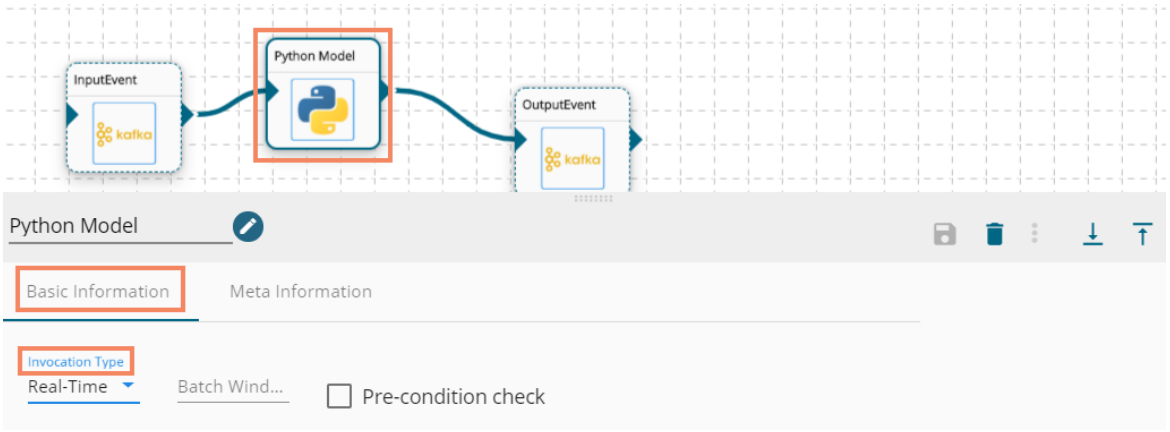
1) Navigate to the Pipeline Workflow Editor and Expand the ML section from the Components Pallet
2) Drag and drop the Python Model (runner) component to the workspace



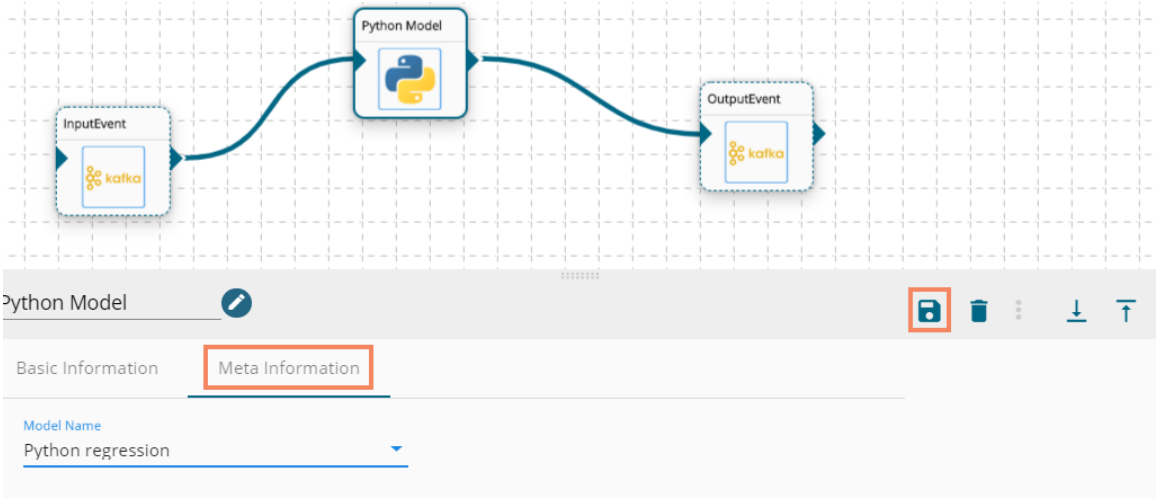
3) The Python Model runner requires input data from an Event and sends the processed data to another Event. So, create two Events and drag them onto the Workspace.
4) Connect the input and output events with the Python Model runner component as displayed below.
5) The data in the input event can come from any Ingestion, Readers or shared events.



6) Click the Python Model runner component to get the configuration tabs.
7) The Basic Information tab opens by default
8) Choose an Invocation Type from the drop-down menu (Real-time/Batch)



9) Select Meta Information and click on Model name field.
10) All the exported predictive models list in a menu.
11) Search the model you want to use and select the model.
12) Save the Python Model runner component.



13) Save the pipeline and activate it .
14) The Python Model runner component reads the data coming to input event, runs the model, and gives the output data with predicted columns to the output event.