

Alteryx Predictive Analytics for Oracle R

I. Software Installation

In order to be able to use Alteryx's predictive analytics tools with an Oracle Database connection, your client machine must be configured to use Oracle R Enterprise (ORE). The full instructions for setting up ORE are located on the Oracle website [here](#) (Oracle R Enterprise Installation and Administration Guide). The guide below is provided to help you get started quickly, but if you run into any issues, please refer to the official Oracle documentation.

You will need to download several files from the Oracle Technology Network (OTN) website to complete the installation, which requires setting up a free account if you don't already have one, and logging in. Here is a brief summary of the steps required:

1. Install Oracle Instant client by downloading two zip files (instantclient and instantclient-sdk), extracting them, and setting the required environmental variables
2. Download and install two sets of R packages (client and supporting)

Note: There is also a server-side component to ORE, for which installation instructions can be found [here](#). Contact your database administrator if you are unsure if ORE has been installed on Oracle Database.

Installation Quick Guide:

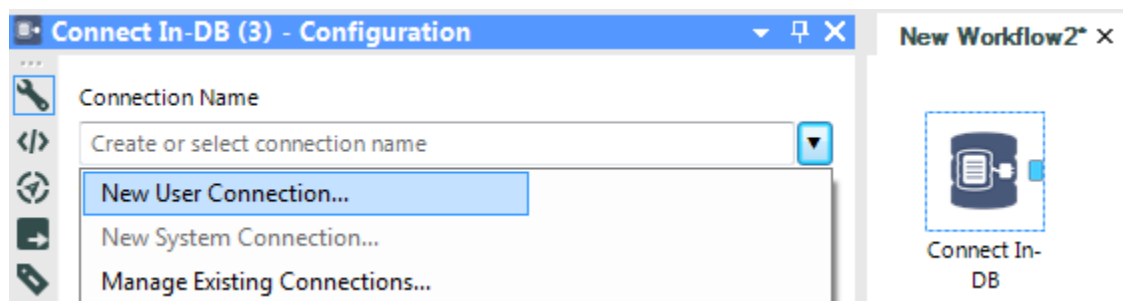
1. **Oracle client installation:** The Oracle predictive tools will not work with ODBC connections, and an OCI connection is required instead. The OCI drivers are included with the Oracle client software.
 - a. To determine if your machine is already setup to use an OCI driver, check whether the OCI_LIB64 environment variable is set to a valid location by following these steps:
 - **Check through the Windows interface**
 - i. Click on the Windows start menu, right click on "Computer", and click "Properties" (Windows 7/8)
 - For users on Windows 10: right click the Windows button in the task bar and select "System" from the menu
 - ii. Click on "Advanced Settings"
 - iii. On the "Advanced" tab, click "Environmental Variables..."
 - iv. Look for a variable called "OCI_LIB64" in both User variables and System variables.
 - **Or, check through the command line**
 - i. Open the Command Prompt
 - ii. Type `echo %OCI_LIB64%` and hit enter
 - iii. If a directory is returned, then continue to the next bullet. Otherwise, you will need to install the Oracle client.
 - **If the OCI_LIB64 variable has previously been created, verify that the directory associated with it exists, and that a file called "oci.dll" exists in that directory.**

- b. If you already have an OCI driver setup, you can skip to the next step. Otherwise, follow the instructions in [Section 6.2.1](#) of Oracle’s installation guide to install the Oracle Instant Client on your machine. This involves downloading two zip files (*Instant Client Package - Basic* and *Instant Client Package - SDK*), extracting them to a location of your choosing, and setting the required environmental variables. Be sure to download the Instant Client version that corresponds to the Oracle Database installation on the server.
 - c. Verify that the OCI_LIB64 and PATH environment variables contain the directory path where the Oracle Instant Client has been installed, following the instructions in step 1.a above. If the path is incorrect, you will get a connection error when trying to connect to the database with Alteryx.
- 2. ORE package installation for R:** The client packages and supporting packages must be installed in R to leverage the ORE capabilities of the database. To install them, follow the instructions in [Sections 6.3.1](#) and [6.4.1](#) of the ORE installation guide. For step 4 of each of those sections, you are given multiple options to choose from. If you are unsure of which to do, we recommend installing from the R GUI. (While R Studio allows you to install only one package at a time, the base R console allows you to select multiple packages for installation at the same time through the GUI.)
- Note:** If you’ve installed the Alteryx predictive tools, which are required for use with Oracle, you should have an instance of R in your Start menu and you’ll want to use the version that has “x64” in the name. If you do not see R in your start menu, you should find it the base folder of your Alteryx installation. For example, “C:\Program Files\Alteryx\R-3.x.x”.
- 3. Restart Alteryx for changes to take effect.**

II. Setting up an Oracle Connection in Alteryx

An OCI connection to Oracle Database is required to use the Alteryx predictive tools with an Oracle In-DB connection. If you’ve previously connected to your database with ODBC, you’ll need to setup an OCI connection by following these steps.

1. Add a Connect In-DB tool onto a canvas in Alteryx from the In-DB tool palette, and select “New User Connection...” from the connection menu in the Configuration panel.



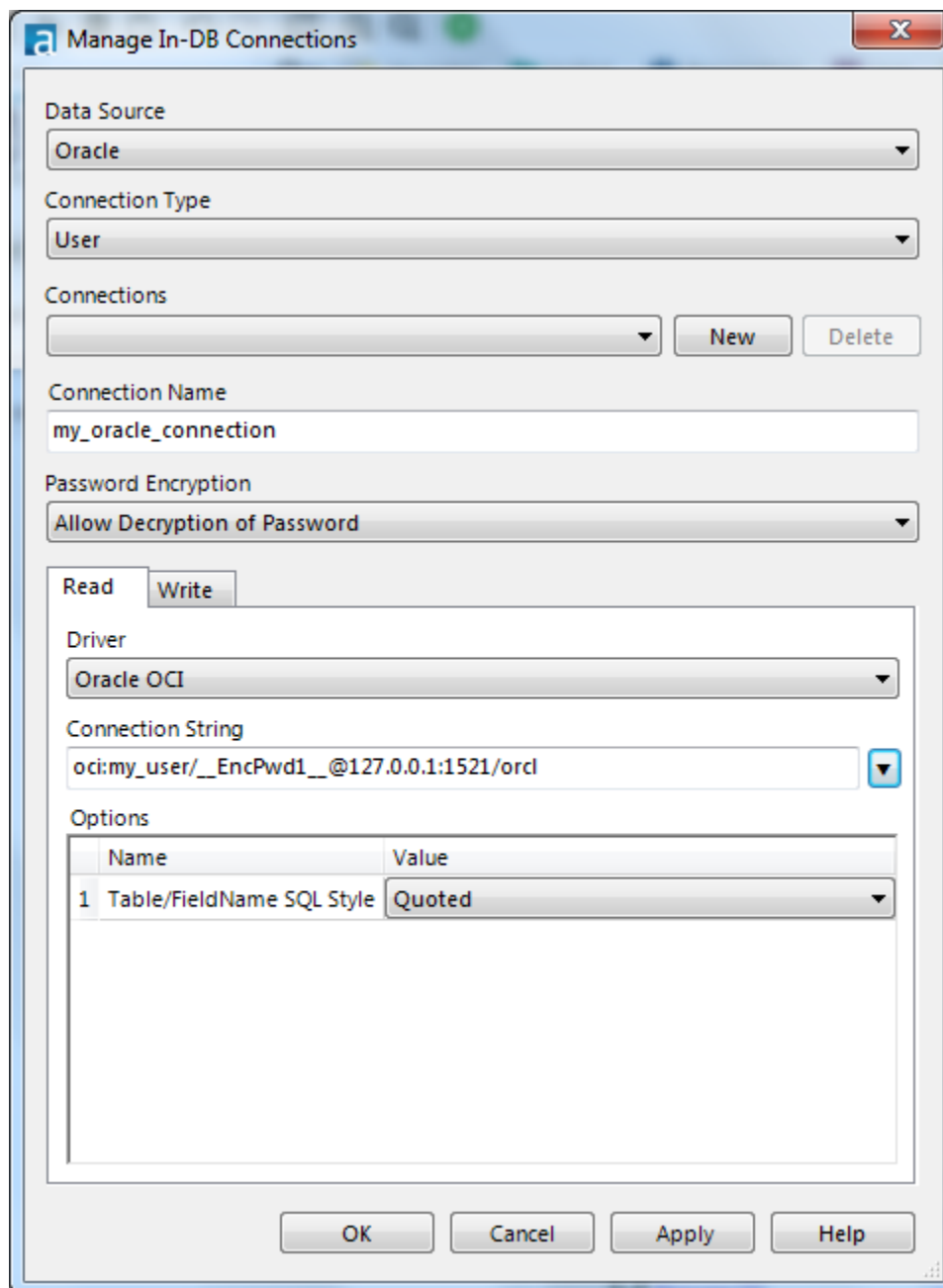
2. In the Create In-DB User Connection window, configure the properties as described below:
 - a. **Connection Name:** Give the connection a name of your choosing.

- b. **Data Source:** Select “Oracle”.
- c. **Password Encryption:** Select “Allow Decryption of Password”. This is required to use the predictive tools with an Oracle connection.
- d. **Driver:** Select “Oracle OCI”.
- e. **Connection String:** Click the menu arrow on the right, and select “New database connection...” This will bring up the Oracle Connection window, which should be configured as so:
 - i. **TNS Server Name:** Type in the connection using the IP address, port number, and service ID as *ip_address:port/service_id*. For example, “127.0.0.1:1521/orcl”.

[Note: A TNS name can be used in this field if you’ve installed the full Oracle Database software on your computer. However TNS names are not currently supported in conjunction with the Oracle Instant Client.]

- ii. **User Name and Password:** These are the user name and password used to connect to the database.

On the **Write** tab, the driver selected should be left as the default “Same as Read Driver”. Write access is required by the predictive tools.



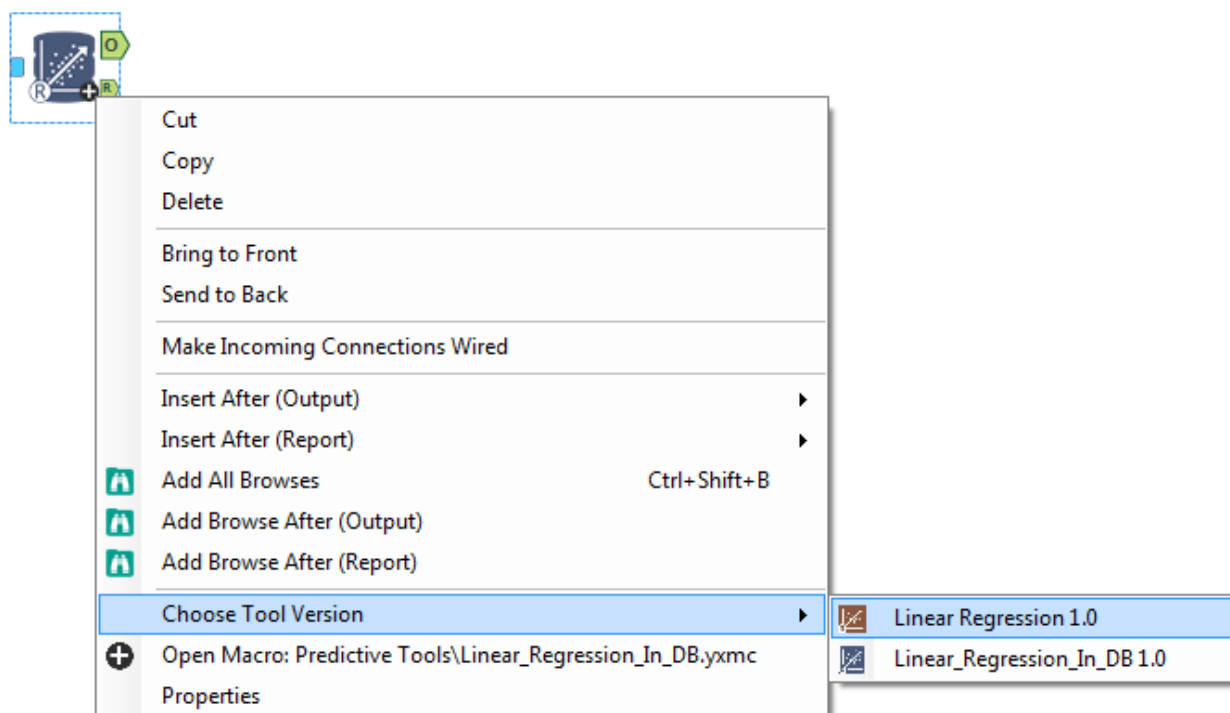
III. Using the Predictive Tools with Oracle In-DB Connections

The following predictive tools currently support Oracle In-DB connections:

- Linear Regression
- Logistic Regression
- Score

The In-DB predictive tools are accessed by dragging tools from the Predictive tool category palette, in the same way that the normal predictive tools are brought onto the canvas. When other In-DB tools have previously been placed on the canvas, the predictive tool will detect that and will have a blue In-DB input connector, indicating that the In-DB version of the tool is being used.

You can change whether the predictive tool is In-DB or not by right clicking on the tool, selecting “Choose Tool Version” from the context menu, and clicking on the tool version that does not contain “In_DB” in its name.



IV. Running Models, Saving Models, and Loading Models with Oracle In-DB Predictive Tools

The Oracle In-DB predictive tools in Alteryx have been designed to function in much the same way as the normal (“non-DB”) predictive tools. However, there are a couple particularly important differences that you should be aware of.

Background: In general, the Alteryx In-DB tools are able to provide such high performance because the processing is done in the database, using instructions produced by Alteryx, rather than transferring the data onto your computer to be processed by Alteryx. With the In-DB predictive tools for Oracle, this is also the case, and Alteryx generates R code using the Oracle R Enterprise (ORE) package, that when executed, builds models and scores data at the Oracle Database server – eliminating data movement and gaining performance and scalability.

When using ORE as well as open-source R (which is included in the Alteryx predictive tools installation), when a model is run, a “model object” is created which contains information that describes the model

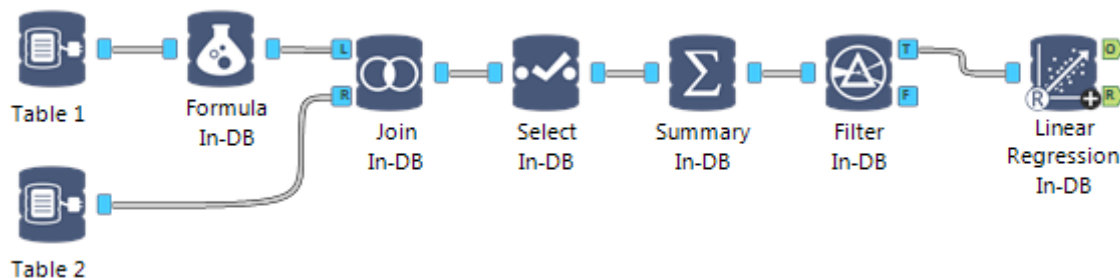
including everything needed to run the model on a new set of data for scoring purposes. The ORE version of models differs notably from the open-source R models, in that open-source R includes in the model object all of the original data that was used to build the model, and uses this data to calculate a number of statistics. In contrast, ORE models do not add the estimation data to the model object, but instead includes a reference to the database table containing the data used to estimate the model (what we call the “estimation table”).

This is a good thing, because when you’re building models with large sets of data stored in your database, you won’t need to worry about all of that data being duplicated by the model object when you run a model. However, this means that if the table that was used to build the model doesn’t exist in the database (for example, if you used the Alteryx In-DB tools to transform the data and generate a query prior to building the model), then the potential usefulness of the model object will be reduced, depending on what you want to use the model object for.

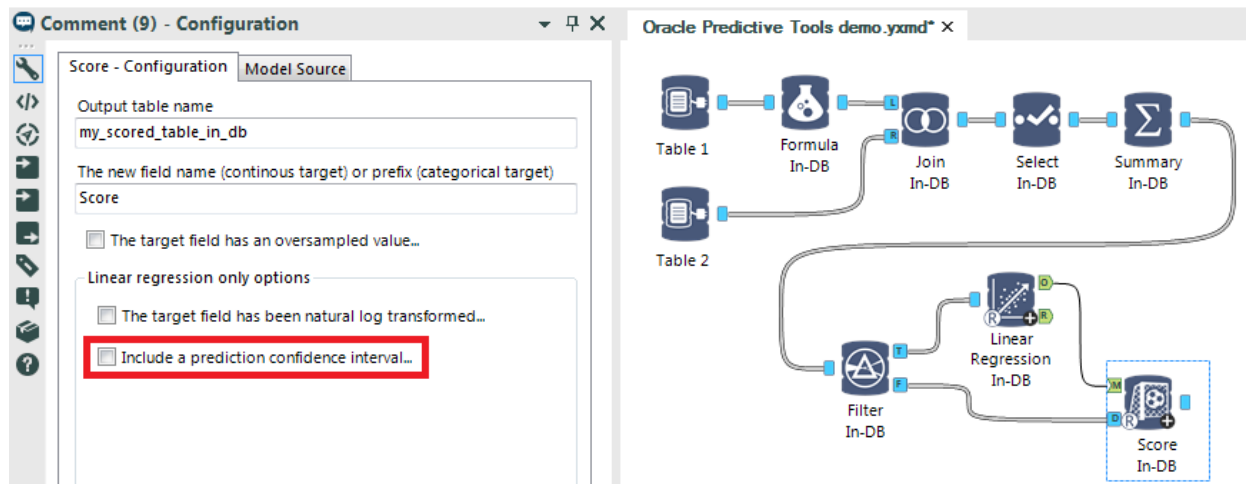
As a result, the Alteryx predictive In-DB tools have been designed to give you the flexibility to build and save models in whatever way best suits your needs. The following section gives practical advice to achieve that.

The Estimation Table

The Alteryx In-DB tools allow you to easily transform the data in the database, by allowing you to use the Alteryx tools you’re familiar with and having them generate a query to run in the database. This query is a set of instructions that tells the database what to do with the data. For example, the following set of tools would generate a query to produce a transformed set of data which is then fed into the Linear Regression In-DB tool:

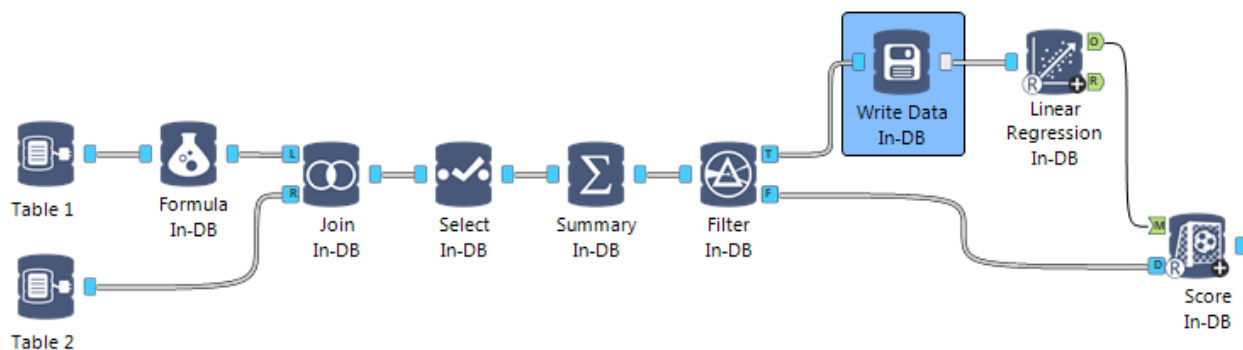


When you build a model with the In-DB predictive tools from a query you’ve created, you lose the advantages provided by having the estimation table saved to the database. ORE essentially considers the estimation table to be an extension of the model object, and while you will still be able to score datasets with the resulting model, you will lose the ability to calculate prediction confidence intervals with the model:



Advanced users will also find that they cannot create a model summary or use methods that create statistics based on row level calculations.

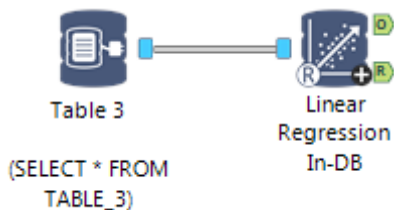
If you require the additional capabilities provided by having a persistent estimation table, the solution is simply to use the Write Data In-DB tool prior to building the model:



In most cases, you will want to select the “Overwrite Table (Drop)” option from the Creation Mode dropdown in the Write Data In-DB tool. This will create a new table if it does not yet exist, and overwrite it (rather than error) if you run your workflow more than once. Be sure to give the table a descriptive name that will not be accidentally overwritten by yourself or another user (which may happen with a generic name like “model”). You will want to use a similar naming convention for the model object itself, which can be saved to the database, and is covered in the next section.

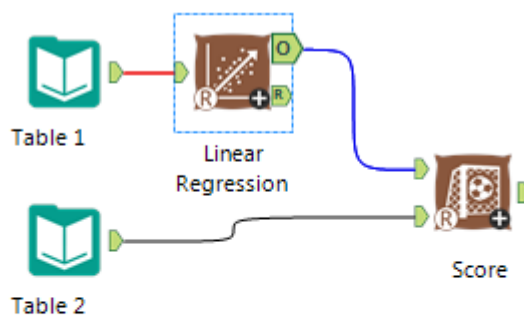
It is worth noting that you do not need to create a new table if you are building a model using data that is already in an existing database table. When you access an existing table, the query that gets sent to the database is always “SELECT * FROM [table]” where [table] represents the table name, and the asterisk means that you are taking all of the fields in the table. This is done automatically when a table is selected from the “Tables” view in the Connect In-DB tool.

By selecting the full table to be used for building your model, that table will be identified as the estimation table by the model object, and you will have the full model object capabilities associated with having a persistent estimation table.



The Model Object

With predictive tools in a standard Alteryx workflow, the model object is passed out of the modeling tool into the Score tool and contains a Name field and an Object field:



Results - Linear Regression (2) - Output

2 of 2 Fields | Cell Viewer | 1 record displayed

Record #	Name	Object
1	my_model	51813245 Bytes

You can save the model object with the Output Data tool and use the Input Data tool to pass it back into the Score tool to avoid the unnecessary re-building of the model each time you want to score a set of data.

With the new Oracle In-DB predictive tools, you still have that capability as an option, but ORE is designed for the model object to live directly in the database, making it more accessible and so that it is located in the same place as the estimation table (which completes the model object). By default, the model object will be saved to the database, but the user is given the option to skip saving it to the database in the configuration of the In-DB predictive tools:

Configuration

Model Name

A specific model name

The model name

my_linear_regression_model

Automatically generate a model name

Select the target variable

Wins

Select the predictor variables [All](#) [None](#)

Tm

Games

Wins

Win_Pct

X_Bat

BatAge

R_G

Omit a model constant.

Use a weight variable for weighted least squares. (Optional)...

Save the model to the database (recommended)

There is also a new tab on the Score In-DB tool that allows you to specify whether to use the model object in the database (identified by the value in the Name field), or to use the model object passed in from the preceding tool (contained in the Object field).

Score - Configuration **Model Source**

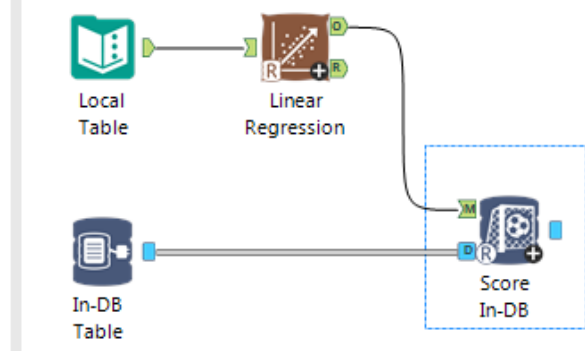
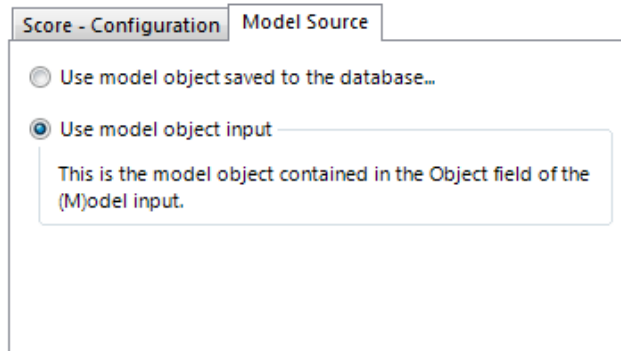
Use model object saved to the database

This will match the value in the Name field of the (M)odel input.

Use model object input

This is the model object contained in the Object field of the (M)odel input.

Finally, although the model objects created by the In-DB tools have their differences compared to the “non-DB” predictive tools, ORE (and the Score In-DB tool) is designed to be able to use both types of model objects, in which case you would select the second model source option:



Further information regarding configuration and additional specifics can be found on the individual tools' help pages.