pitney bowes™

Spectrum™ Technology Platform

Version 2018.2.0

Analytics Scoring Guide

Table of Contents

1 - Introduction

2 - Analytics Scoring Module

3 - Machine Learning Model Management

4 - Analytics Scoring Repository

1 - Introduction

In this section

Analytics Scoring Module

The Spectrum™ Technology Platform Analytics Scoring Module allows predictive models defined in either **QMML** on page 12 (Spectrum Miner's proprietary model format) or **PMML** on page 13 (industry standard Predictive Model Markup Language) to be evaluated within dataflows and additionally allows dataflows to write and retrieve data from Miner datasets which can be consumed by the Miner predictive analytics software. This allows data to be enriched by adding predicted or scored outputs using models created by data insight teams with industry standard data modeling tools. This may include models used to calculate the churn risk for existing customers or credit scoring models to determine a consumer's credit rating.

Analytics Scoring Module Components

The Analytics Scoring Module consists of the following components.

- **Binning Lookup**—This stage can be used to apply previously defined binning to new data using existing bins created in dataflows using the Machine Learning Module Binning stage.
- **Java Model Scoring**—This stage can be used score new data using the formula created when you fit a machine learning model.
- **PMML Model Scoring**—This stage can be used to evaluate any model stored in the Analytics Scoring Repository in the context of a dataflow.
- **Read from Miner Dataset**—This stage can be used to read data from a focus file to be used within a dataflow.
- **Write to Miner Dataset**—This stage can be used to write data from a dataflow to a focus file.
- **Machine Learning Model Management**—This repository includes Model Assessment, where you manage all machine learning models on your Spectrum™ Technology Platform server, and Binning Management, where you manage all binning on your Spectrum™ Technology Platform server.
- **Analytics Scoring Repository** —This is the central repository for all models available to the Analytics Scoring Module. Users can manage the repository via a web client.

See "Data Science Demonstration Flows" in the **Machine Learning Guide** for examples of supervised and unsupervised learning that include the scoring of data using Java Model Scoring.

2 - Analytics Scoring Module

In this section

Binning Lookup

Introduction to Binning Lookup

Binning Lookup applies previously defined binning to new data using existing bins created in dataflows using the **Binning** stage.

Defining Binning Properties

1. Under **Primary Stages** / **Deployed Stages** / **Machine Learning**, click the **Binning Lookup** stage and drag it onto the canvas, placing it where you want on the dataflow and connecting it to other stages. Note that the input stage must contain the data that you want binned. One output stage is required for the binned output; you may optionally connect a second output stage to capture the binning summary.
2. Select the appropriate **Binning name** from the drop-down. These are names of existing bins that were created by a dataflow that uses the Binning stage.
3. The **Binning type** and **Description** fields are imported with the bins from the Binning name you selected in Step 2 and are therefore noneditable.
4. The **Inputs** grid shows each field that was included for binning in the Binning stage along with the data type.
5. Click **OK** to save your settings.

Binning Output

This tab shows the fields and data types that are being binned by your Binning Lookup stage. See **Binning Output** for more information on the output generated by using a binning stage. We have now provided the option to edit binned field in binning output tab. Use can use "Spectrum Binned Field" to give new name for a binned field. We have also provided option to include/exclude binned fields.

Java Model Scoring

Introduction to Java Model Scoring

Java Model Scoring enables you to score new data using the formula created when you fit a machine learning model.

> **Note:** Models must first be exposed through Machine Learning Model Management before they become available in the Java Model Scoring stage. For more information see **Introduction to Model Assessment** on page 31.

To score your data, you must complete two tabs of the **Java Model Scoring Options** dialog. First identify the model and its type, then ensure the model's fields are correctly mapped to Spectrum™ Technology Platform fields. Following that, you configure the output by selecting which fields you want to include and running your job. The **Model Output** tab contains mapping for data types for Spectrum™ Technology Platform and your model.

If your job contains a stage that captures the output in a file or a table, you can use that ouptut in a subsequent dataflow or web service.

Defining Model Properties

1. Under **Primary Stages** / **Deployed Stages** / **Advanced Analytics**, click the **Java Model Scoring** stage and drag it onto the canvas, placing it where you want on the dataflow and connecting it to input and output stages. Note that the input stage must be the data source that contains both the objective and input variable fields for your model. If you are running your job in batch mode, you will also need an output stage to capture model scores; otherwise you will use a Spectrum™ Technology Platform web service to score data in real time.
2. Double-click the Java Model Scoring stage to show the **Model Scoring Options** dialog box.
3. Optional: Select the type of a model you are scoring in the **Type filter** drop-down.
4. Select the **Type filter** being used to score the model.
5. Select the **Model name** from the drop-down.
6. Enter the type of model you are scoring in the **Model type** field.
7. Optional: Enter a **Description** of the model.
8. Check **Ignore unknown categorical levels** to return data in the Predicted_Value column for rows with a categorical level that was not in the data used to fit the model. If you leave this box unchecked, the Predicted_Value column will return "Null/NA" for those rows.

9. The **Inputs** table shows information for the model's input fields. These fields and their data types automatically map to Spectrum fields and data types.
10. Click **OK** to save these options or continue to the next tab.

Model Output

The **Outputs** table shows information for the model's output fields. These fields and their data types automatically map to Spectrum fields and data types.

1. Click **Include** for each field whose data you want included in the model's output.
2. Click **OK** to save the model.

PMML Model Scoring

Introduction to PMML Model Scoring

The PMML Model Scoring stage is capable of evaluating analytical models which have been published to the Analytics Scoring Repository in the context of a dataflow. The evaluator operates on single data rows using the fields from each row as the inputs to the model. User selected outputs from the model are written to the output channel.

> **Note:** For details of the supported model types and type mappings see **Supported Model Formats** on page 12

Deploying a Model

This procedure describes how to configure the PMML Model Scoring stage to deploy an analytics model as part of a dataflow.

1. Under **Primary Stages** / **Deployed Stages** / **Advanced Analytics**, click the **PMML Model Scoring** stage and drag it onto the canvas, placing it where you want on the dataflow and connecting it to input and output stages.
2. Double click the PMML Model Scoring stage to show the **PMML Model Scoring Options** dialog box.

   By default the options dialog shows the details of the first model in the list of available models.

3. Click the **Type Filter** drop-down and select the model type to filter by.

   Only model types which are associated with at least one model in the **Analytics Scoring Repository** are listed.

4. Click the **Model** drop-down and select the model to deploy.
   The details of the chosen model are displayed in the dialog.

   Model type the type of the selected model as described in **Supported Model Formats** on page 12

   Description a short text about the purpose of the model

   Inputs a table containing information about all the required input fields for the model. Each row contains information about an input field. The model input field name will automatically be mapped to a valid Spectrum field name on publish (see **Add a Model via Web Browser Client** on page 41 or **Add a Model via Spectrum Miner Spectrum Connector** on page 43).

      Model Field Name the name of the field as specified in the model definition

      Spectrum Field Name the name of the field as used in the Spectrum platform

      Model Field Type the type of the field as specified in the model definition

      Spectrum Field Type the Spectrum field type that is mapped to the model field type as described in **QMML** on page 12 and **PMML** on page 13

5. Select the **Configuration** tab.
   Details about the model's **Outputs** are displayed in a table. It contains information about all the output fields for the model. Each row has information about an output field. The model output field name will automatically be mapped to a valid Spectrum field name on publish (see **Add a Model via Web Browser Client** on page 41 or **Add a Model via Spectrum Miner Spectrum Connector** on page 43).

   Model Field Name the name of the field as specified in the model definition

   Spectrum Field Name the name of the field as used in the Spectrum platform

   Model Field Type the type of the field as specified in the model definition

   Spectrum Field Type the Spectrum field type that is mapped to the model field type as described in **QMML** on page 12 and **PMML** on page 13

   Include a checkbox to specify whether to use this output

6. Optional: Uncheck the **Included** column of any row in the **Outputs** table to exclude the output (i.e. stop it from being written to the output channel).

   At least one output must remain selected. If all outputs are excluded then a validation error symbol will appear beside the **Outputs** table. This means the current model configuration is invalid and the model cannot be deployed. The validation error symbol will remain visible until the error is corrected.

7. Optional: Click the **Spectrum Field Name** column of any row in the **Outputs** table to rename the fields as required.

   No two outputs can share the same **Spectrum Field Name** and **Spectrum Field Names** must follow the standard Spectrum™ Technology Platform field naming conventions. If any validation errors are detected then a validation error symbol is displayed beside the **Outputs** table; hovering the mouse curser over a validation error symbol shows the error details.

   > **Note:** Renaming an output's **Spectrum Field Name** only affects the specific instance of the stage, it does not update the Analytics Scoring Repository .

   > **Note:** Spectrum™ Technology Platform field names must:
   >
   > - Be unique.
   > - Be non-empty.
   > - Contain only alphanumeric, period or underscore characters.
   > - Cannot start with a period.

8. Finally click **OK** to save the chosen model and configuration.

Re-configure PMML Model Scoring Settings

This procedure describes how to re-configure the PMML Model Scoring stage to re-name or change the outputs that are generated from the PMML Model Scoring stage or to change the deployed model to be evaluated by the stage.

1. Double click the PMML Model Scoring stage to show the **PMML Model Scoring Options** dialog box.

   The options dialog shows the model which was previously selected, configured and deployed.

   > **Note:** If the selected model has been deleted from the Analytics Scoring Repository prior to opening the **PMML Model Scoring Options** dialog, a validation error symbol is displayed beside the **Model** drop-down list. This means any new changes made to the model's configuration cannot be deployed (step 5). Clicking **Cancel** will exit the **PMML Model Scoring Options** and allow the deleted model to be used within the dataflow as previously configured. In order to apply any new changes to the stage configuration, select a different (non-deleted) model from the **Model** drop-down list. After applying a change of model, further re-configuration of the stage will no longer have the deleted model available in the **Model** drop-down list.

2. Optional: From the **Model** tab select a different model from the **Model** drop-down list to change which model is used within the dataflow.

Note: On changing the selected model in the **Model** drop-down list, all configuration changes for the previously chosen model will be discarded. Re-selecting the model will result in the default output configuration for the model. Clicking **Cancel** will undo any changes made since opening the **PMML Model Scoring Options** allowing any pending changes to be reverted.

Note: On selecting a model, if the model has been deleted since the **PMML Model Scoring Options** dialog was opened then a validation error symbol will be displayed beside the **Model** drop-down list. In this scenario the **Inputs** and **Outputs** will not be available and re-configuration of the model settings will not be able to be applied until a non-deleted model has been selected.

3. Select the Configuration tab
4. Make any desired changes to the selected model's outputs configuration. For example, rename a **Spectrum Field Name**, or change its **Include** checkbox status to be included/excluded from the dataflow.

   Any changes to the outputs must adhere to the following validation rules: At least one output must be included. Output's **Spectrum Field Name** must be unique and must follow the standard Spectrum™ Technology Platform field naming conventions. If any validation errors are detected then a validation error symbol will be displayed beside the table; hovering the mouse cursor over a validation error symbol will show the error details.

   Note: Spectrum™ Technology Platform field names must:

   - Be unique.
   - Be non-empty.
   - Contain only alphanumeric, period or underscore characters.
   - Cannot start with a period.

5. Once all desired changes have been made click the **OK** button to apply the new changes, or click **Cancel** to undo any changes made and close the **PMML Model Scoring Options** dialog.

Output

The PMML Model Scoring stage returns the selected model output fields. Additionally, if the PMML Model Scoring stage fails to process a record, it returns the fields Status, Status.Code, and Status.Description. These fields provide information on why the stage failed to process the record as detailed below.

Field Name	Description
Status	Reports the success or failure of the evaluation attempt: **null** Success **F** Failure
Status.Code	Reason for failure or error: **InputConversionFailed** Failed to convert Spectrum input field types to the required model field types. **ModelEvaluationFailed** Model evaluation failed for the given record.
Status.Description	Description of the problem.

Supported Model Formats

The PMML Model Scoring supports the deployment of analytical models saved in both **QMML** on page 12 and **PMML** on page 13 file formats.

QMML on page 12 models can be created and exported from Spectrum Miner. All types of analytical models and segmentations exported from Spectrum Miner are supported by the PMML Model Scoring stage and can be deployed within a Spectrum™ Technology Platform dataflow.

PMML on page 13 models can be created and exported from many commercial and open source modeling tools.

QMML

QMML is a proprietary XML based file format used to represent model results generated from Spectrum Miner.

All types of analytical models and segmentations exported from Spectrum Miner are supported by the PMML Model Scoring stage and can be deployed within a Spectrum™ Technology Platform dataflow.

Type Mapping

QMML model inputs and outputs are automatically mapped to Spectrum™ Technology Platform field types.

QMML Field Type	Spectrum™ Technology Platform Field Type
integer	integer
real	double
string	string
date	datetime

Supported Models

All model types constructed within Spectrum Miner (including decision tree, scorecard, cluster analysis and naive bayes models) are interpreted by the Analytics Scoring Module as a Miner Model.

Miner Model

A Miner Model is any type of **QMML** on page 12 model exported from Spectrum Miner, such as those generated from the decision tree, scorecard, cluster analysis or naive bayes modeling tools.

Unsupported Features

All compiled **QMML** on page 12 exported from Spectrum Miner is supported.

Model Outputs

Field	Description
<dynamic fields>	A field is output for each **QMML** on page 12 model output.

PMML

Predictive Model Markup Language (PMML) is an XML-based file format developed by the Data Mining Group to provide a way for applications to describe and exchange models produced by data mining and machine learning algorithms. PMML files can be created and exported from many commercial and open source modeling tools.

Type Mapping

PMML model inputs and outputs are automatically mapped to Spectrum™ Technology Platform field types. All PMML models inputs and outputs must be of supported types.

PMML Field Type	Spectrum™ Technology Platform Field Type
string	string
integer	integer
float	float
double	double
boolean	boolean
date	date
dateDaysSince[1960]	date
dateDaysSince[1970]	date
dateDaysSince[1980]	date
dateDaysSince[0]	***not supported***
time	time
timeSeconds	time
dateTime	datetime
dateTimeSecondsSince[1960]	datetime
dateTimeSecondsSince[1970]	datetime
dateTimeSecondsSince[1980]	datetime
dateTimeSecondsSince[0]	***not supported***

Supported Models

The Analytics Scoring Module currently supports the PMML model types detailed in the following sections.

Association Rule

A **PMML** on page 13 association rule model represents rules where some set of items is associated to another set of items. For example a rule can express that a certain product or set of products is often bought in combination with a certain set of other products, also known as Market Basket Analysis. An Association Rule model typically has two variables: one for grouping records together into transactions and another that uniquely identifies each record.

Model Element

```
<AssociationModel functionName="associationRules" ...
```

Unsupported Features

Non-string field types for the field(s) identifying the item are not supported.

Having more than one field for grouping records is not supported.

Association Rule models with the <MiningSchema> element containing a reference to a <DerivedField> element are not supported.

Model Outputs

Supported Model Output Features	Description
transformedValue	A value generated via a transformation expression applied to the predicted model output.
decision	A value generated via an expression applied to the predicted model output resulting in a categorized value.
entityId	The id of the winning rule (default), or the rule specified by the rank value. If the selected rule does not provide an id, a 1-based index is returned.
ruleId	This is identical to the entityId option and has been deprecated as of PMML 4.2. Although its use is currently supported it is recommended to use entityId.
affinity	The affinity of the winning rule (default), or the rule specified by the rank value.
antecedent	The antecedent of the winning rule (default), or the rule specified by the rank value. This output will be formatted as a comma separated string of values.
consequent	The consequent of the winning rule (default), or the rule specified by the rank value. This output will be formatted as a comma separated string of values.

Supported Model Output Features	Description
rule	The winning rule (default), or the rule specified by the rank value. This output will return a description of the rule, formatted in the following way: {<antecedent>}->{<consequent>}.
confidence	The confidence of the winning rule (default), or the rule specified by the rank value.
support	The support of the winning rule (default), or the rule specified by the rank value.
lift	The lift of the winning rule (default), or the rule specified by the rank value.
leverage	The leverage of the winning rule (default), or the rule specified by the rank value.

Clustering

A **PMML** on page 13 clustering model determines the best matching cluster for a given record based on the distance or similarity measure used for clustering. A cluster is a subset of similar data. Clustering (also called unsupervised learning) is the process of dividing a dataset into groups such that the members of each group are as similar to each other as possible and different groups are as dissimilar from each other as possible.

Model Element

```
<ClusteringModel functionName="clustering" ...
```

Unsupported Features

Clustering models with the <MiningSchema> element containing a reference to a <DerivedField> element are not supported.

Model Outputs

Supported Model Output Features	Description
predictedValue	The best matching cluster based on the distance or similarity measure used for clustering.
transformedValue	A value generated via a transformation expression applied to the predicted model output.
decision	A value generated via an expression applied to the predicted model output resulting in a categorized value.

Supported Model Output Features	Description
predictedDisplayValue	The human readable value used to represent the predicted value from the model.
entityId	If present, the 1-based index (implicit identifier) of the winning/predicted cluster.
affinity	The value of the distance or the similarity of the provided record to the predicted cluster as defined in the model.

Classification Tree

A **PMML** on page 13 classification tree model predicts membership of a categorical dependent variable from one or more independent variables.

Model Element

```
<TreeModel functionName="classification" ...
```

Unsupported Features

Classification trees with a missing value strategy of "aggregateNodes" or "weightedConfidence" are not supported.

Classification Tree models with the <MiningSchema> element containing a reference to a <DerivedField> element are not supported.

Model Outputs

By default the target field will be available as an output field - this is a synonym for the predictedValue feature.

Supported Model Output Features	Description
predictedValue	The categorical dependent variable that we are predicting membership of.
transformedValue	A value generated via a transformation expression applied to the predicted model output.
decision	A value generated via an expression applied to the predicted model output resulting in a categorized value.
predictedDisplayValue	The human readable value used to represent the predicted value from the model.

Supported Model Output Features	Description
probability	The statistical probability of the predicted value. Multiple probability outputs can be specified in the model, one for each predicted category or by rank.
residual	The residual of the probability output value (1 - probability) for the predicted category or rank. Multiple residual outputs can be specified in the model, one for each predicted category or by rank.
entityId	If present, the ID of the tree node of the predicted result.

Regression Tree

A **PMML regression tree model** predicts the value of a numeric-dependent variable (such as the price of a house) from one or more independent variables. It does this by building a decision tree model that is based on one or more predictors.

Model Element

```
<TreeModel functionName="regression" ...
```

Unsupported Features

Regression trees with integer or float target fields are not supported unless a <Targets> element is specified with the appropriate castInteger attribute.

Regression tree models with the <MiningSchema> element containing a reference to a <DerivedField> element are not supported.

Model Outputs

By default the target field will be available as an output field - this is a synonym for the predictedValue feature. Float target fields will always be cast to integer.

Supported Model Output Features	Description
predictedValue	The numeric dependent variable that we are predicting.
transformedValue	A value generated via a transformation expression applied to the predicted model output.
decision	A value generated via an expression applied to the predicted model output resulting in a categorized value.

Supported Model Output Features	Description
entityId	If present, the ID of the tree node of the predicted result.

Naive Bayes

A **PMML** on page 13 Naive Bayes model predicts the value of a target from evidence given by one or more predictor fields using Bayes' Theorem. Naive Bayes models require the target field to be discretized so that a finite number of values are considered by the model. Predictor fields may be either discrete or continuous.

Model Element

```
<NaiveBayesModel functionName="classification" ...
```

Unsupported Features

Naive Bayes models with the <MiningSchema> element containing a reference to a <DerivedField> element are not supported.

Model Outputs

By default the target field will be available as an output field - this is a synonym for the predictedValue feature.

Supported Model Output Features	Description
predictedValue	The categorical variable that we are predicting membership of.
transformedValue	A value generated via a transformation expression applied to the predicted model output.
decision	A value generated via an expression applied to the predicted model output resulting in a categorized value.
predictedDisplayValue	The human readable value used to represent the predicted value from the model.
probability	The statistical probability of the predicted value.
residual	The residual of the probability output value (1 - probability) for the predicted value.

Regression

A **PMML** on page 13 regression model predicts the value of a numeric dependent variable from one or more independent variables.

Model Element

```
<RegressionModel functionName="regression" ...
```

Unsupported Features

Regression models with a "normalizationMethod" attribute set to the value "simplemax", "probit", "cloglog" or "loglog" are not supported.

Regression models with integer or float target field are not supported unless <Targets> element is specified with appropriate castInteger attribute.

Regression models with the <MiningSchema> element containing a reference to a <DerivedField> element are not supported.

Model Outputs

By default the target field will be available as an output field - this is a synonym for the predictedValue feature. Float target fields will always be cast to integer.

Supported Model Output Features	Description
predictedValue	The numeric dependent variable that we are predicting.
transformedValue	A value generated via a transformation expression applied to the predicted model output.
decision	A value generated via an expression applied to the predicted model output resulting in a categorized value.

Regression Classifier

A **PMML** on page 13 regression classifier combines the output from multiple regression equations to predict a categorical value.

Model Element

```
<RegressionModel functionName="classification" ...
```

Unsupported Features

Regression models with the <MiningSchema> element containing a reference to a <DerivedField> element are not supported.

Model Outputs

By default the target field will be available as an output field - this is a synonym for the predictedValue feature.

Supported Model Output Features	Description
predictedValue	The categorical dependent variable that we are predicting membership of.
transformedValue	A value generated via a transformation expression applied to the predicted model output.
decision	A value generated via an expression applied to the predicted model output resulting in a categorized value.
predictedDisplayValue	The human readable value used to represent the predicted value from the model.
probability	The statistical probability of the predicted value. Multiple probability outputs can be specified in the model, one for each predicted category or by rank.
residual	The residual of the probability output value (1 - probability) for the predicted value. Multiple residual outputs can be specified in the model, one for each predicted category or by rank.

Scorecard

A **PMML** on page 13 scorecard model is a regression based modelling technique mainly used to calculate risk or default probability.

Model Element

```
<Scorecard functionName="regression" ...
```

Unsupported Features

Scorecard models with the <MiningSchema> element containing a reference to a <DerivedField> element are not supported.

Model Outputs

By default the target field will be available as an output field - this is a synonym for the predictedValue feature and will always be of type 'double'.

Supported Model Output Features	Description
predictedValue	The calculated score.
transformedValue	A value generated via a transformation expression applied to the predicted model output.
decision	A value generated via an expression applied to the predicted model output resulting in a categorized value.
reasonCode	If specified the reason codes associated with the characteristics used to determine the score in order of their contribution to the final score. The rank of the request reason code can be specified otherwise the highest ranked reason code will be returned by default.

Read from Miner Dataset

Introduction to the Read from Miner Dataset

The Read from Miner Dataset stage is capable of extracting field information and data from a Miner dataset.

Reading from a Miner Dataset

This procedure describes how to read field information and data from a Miner Dataset.

1. Under **Primary Stages** / **Deployed Stages** / **Advanced Analytics**, drag the **Read from Miner Dataset** stage onto the canvas and connect it to the dataflow.
2. Double click the **Read from Miner Dataset** stage to show the **Read from Miner Dataset Options** dialog.
3. Click the **...** button inside the **File Name** field to show the **Open File** dialog.
4. Using the **Open File** dialog, locate and select the focus file containing the Miner Dataset.

   Focus files have .ftr extension.
5. Click the **Fields** tab in the **Read from Miner Dataset Options** dialog to show the fields table.

   The fields table will be empty when configuring the **Read from Miner Dataset** stage for the first time and anytime a new focus file is selected (step 4).

6. Click the **Regenerate** button.
   The fields table will list all the fields that the **Read from Miner Dataset** stage will read from the Miner Dataset.

   > **Note:** The **Regenerate** button will be disabled if no focus has been selected (steps 3 and 4).

   > **Note:** If the stage fails to retrieve fields from the Miner dataset then an error dialog will be displayed. Select a valid focus file and try again.

7. If required, modify the fields list using the available options (see **Fields Tab** on page 23).
8. Finally, click the **OK** button to apply the configuration.

   If any validation errors are detected then a validation error message will be displayed. Correct the validation errors before clicking the **OK** button again. If no validation errors are found, the **Read from Miner Dataset Options** dialog will close.

   > **Note:** To re-configure the stage, double click the **Read from Miner Dataset** stage to re-display the **Read from Miner Dataset Options** dialog.

Fields Tab

The **Fields** tab contains a table listing all the fields that the **Read from Miner Dataset** stage will read from the Miner Dataset. The following table describes the options on the field that can be used to modify the table contents.

Option Name	Description
Regenerate	Replaces the fields currently defined with those read from the Miner dataset.

The following table describes the properties listed for all fields.

Column Name	Description	Editable
Miner Field Name	The name of the field as specified in the Miner Dataset.	No

Column Name	Description	Editable
Spectrum Field Name	The name of the field that will created on the dataflow channel. If the **Miner Field Name** is not compatible with the Spectrum™ Technology Platform naming rules then the **Spectrum Field Name** will be different. Edit the text within the cell to specify the desired field name for the dataflow channel. Spectrum™ Technology Platform field names must: • Be unique. • Be non-empty. • Contain only alphanumeric, period or underscore characters. • Cannot start with a period.	Yes
Spectrum Field Type	The data type of the field that will be created on the dataflow channel (see **Output** on page 24).	No
Include	A checkbox to specify whether to include this field in the dataflow channel.	Yes

Note: By clicking the table column headers the fields can be sorted by **Miner Field Name**, **Spectrum Field Name** or **Spectrum Field Type**.

Output

The following type mapping between Miner Field Type and Spectrum Field Type is done automatically

Table 1: Field Mappings

Miner Type	Spectrum Type
integer	integer
string	string
real	double
date	datetime

If a field has a name that contains symbols which are invalid in Spectrum, the invalid symbols will automatically be replaced with an _ . If two or more fields have the same transformed name an additional index will be appended to the name e.g.

{x}Field{x}

{y}Field{y}

where {x} and {y} are characters not supported in a Spectrum field name, would end up as

_Field_1

_Field_2

in the output channel.

Write to Miner Dataset

Introduction to the Write to Miner Dataset

The Write to Miner Dataset stage is capable of creating a Miner Dataset to hold the specified fields and data from a Spectrum dataflow.

Writing to a Miner Dataset

This procedure describes how to write field information and data to a Miner Dataset.

1. Under **Primary Stages** / **Deployed Stages** / **Advanced Analytics**, drag the **Write to Miner Dataset** stage onto the canvas and connect it to the dataflow.
2. Double click the **Write to Miner Dataset** stage to show the **Write to Miner Dataset Options** dialog.
3. Click the **...** button inside the **File Name** field to show the **Save File** dialog.
4. Using the **Save File** dialog, choose where the new focus file will be saved and provide an appropriate file name.

   Focus files must have .ftr extension. The .ftr extension will automatically be appended to the file name if it's missing.
5. Optional: Uncheck the **Overwrite** checkbox.

If a focus file with the same name already exists in the directory specified (step 4), unchecking the **Overwrite** checkbox will prevent the existing focus file from being overwritten each time the stage is executed within a dataflow (the stage will fail to execute). However, this will mean that when the **Write To Miner Dataset** stage has been executed successfully within dataflow, the file path must be re-configured (step 4) or the existing focus file will have to be manually moved or deleted.

6. Optional: Apply a metadata file for the new focus (see **Applying Metadata** on page 26).
7. Click the **Fields** tab in the **Write to Miner Dataset Options** dialog to show the fields table.

   The fields table will list all the fields that the **Write to Miner Dataset** stage will write to the Miner Dataset.
8. Optional: Click the **Quick Add** button.
   Clicking the **Quick Add** button launches the **Quick Add** dialog, the dialog lists all fields from the input channel and the current fields list. Select/deselect the checkboxes beside each field to add/remove the field from the fields list.
9. If required, modify the fields list using the available options (see **Fields Tab** on page 27).

   At least one field must be specified or a validation error indicator will be displayed.
10. Finally, click the **OK** button to apply the configuration.

    If any validation errors are detected then a validation error message will be displayed. Correct the validation errors before clicking the **OK** button again. If no validation errors are found, the **Write to Miner Dataset Options** dialog will close.

    > **Note:** To re-configure the stage, double click the **Write to Miner Dataset** stage to re-display the **Write to Miner Dataset Options** dialog.

Applying Metadata

A .qsfm metadata file contains additional information that can be applied to a focus file such as derived field definitions, field interpretations, binnings, record selections, field and focus comments and focus history. This procedure describes how to apply this kind of metadata to a focus file produced using the **Write to Miner Dataset** stage.

1. Obtain a valid Miner focus metadata file (either export from an existing focus in Spectrum Miner or acquire from a third party).

   Metadata files must have a .qsfm extension.
2. Ensure that the **Write to Miner Dataset Options** dialog is currently displayed (see **Writing to a Miner Dataset** on page 25).
3. From the **Write to Miner Dataset Options** dialog, check the **Apply Metadata** checkbox.
4. Below the **Apply Metadata** checkbox, click the **...** button inside the **File Name** field to show the **Open File** dialog.

   The metadata file picker will be disabled when the **Apply Metadata** checkbox is not checked.

5. Using the **Open File** dialog, locate and open the .qsfm file obtained in step 1.

   Note: When the **Apply Metadata** checkbox is checked, a metadata file must be selected before the stage's configuration can be applied.

   Note: Unchecking the **Apply Metadata** checkbox after selecting a metadata file will disable the metadata file chooser and will not apply the metadata file on stage execution, but for convenience the file chooser will not remove the selected file until the **Write to Miner Dataset Options** dialog is closed.

6. Optional: Check the **Ignore warnings** checkbox if you do not want the dataflow to fail if a piece of metadata cannot be applied.

Fields Tab

The **Fields** tab contains a table listing all the fields that the **Write to Miner Dataset** stage will write to the Miner Dataset. The following table describes the options on the field that can be used to modify the table contents.

Option Name	Description
Add	Clicking the **Add** button launches the **Add** dialog for adding a new field to the fields list. The **Add** dialog requires the new field's name, Miner type and length (if Miner type is 'string') to be specified.
Remove	Deletes the currently selected fields from the fields list.
Regenerate	Clicking the **Regenerate** button launches a dialog prompting for confirmation of the field regeneration. If the action is confirmed all existing fields are replaced with the fields found in the selected focus file. **Note:** The **Regenerate** button will be disabled when there is no focus file selected. **Note:** Regenerating from an invalid focus will not clear or change the existing fields list.

Option Name	Description
Quick Add	Clicking the **Quick Add** button launches the **Quick Add** dialog, the dialog lists all fields from the input channel and the current fields list. Select/deselect the checkboxes beside each field to add/remove the field from the fields list. **Note:** The **Quick Add** button will be disabled when a field's Miner field name is invalid.

The field properties can also be updated by modifying cells in the following table columns directly.

Column Name	Description
Miner Field Name	The name of the field as it will appear in the new Miner dataset. Edit the text within the cell to specify the desired field name. Field names must: • Be unique. • Be non-empty. • Start with a letter. • Contain only alphanumeric or underscore characters. • Not be longer than 128 characters. If a field name is changed to an invalid name then a validation error symbol will be displayed beside the fields table. Correct the validation error before clicking the **OK** button.
Miner Field Type	The Miner data type the field will use in the new Miner dataset (see **Output** on page 28). Change the field's datatype by selecting from the drop down list.
String Length	If the chosen Miner field type in the table row is 'string' then the user will be able to specify an integer value between 1 and 4000 to set the length for the string field in the Miner focus. Any strings values for the field that are longer than the specified string length will be truncated to this length in the resulting focus.

Output

The following type mapping between Spectrum Field Type and Miner Field Type is done automatically

Table 2: Field Mappings

Spectrum Type	Miner Type
integer	integer
string	string
double	real
datetime	date

All other Spectrum types map to Miner type 'String'.

3 - Machine Learning Model Management

In this section

Accessing Machine Learning Model Management

There are three ways to access Machine Learning Model Management:

- Use the Spectrum™ Technology Platform Welcome Page:
  - Open a web browser and go to the Spectrum™ Technology Platform Welcome Page at:

    <servername>:<port>

    For example, if you installed Spectrum™ Technology Platform on a computer named "myspectrumplatform" and it is using the default port 8080, you would go to:

    myspectrumplatform:8080
  - Click **Spectrum Machine Learning**.
  - Click **Open Machine Learning Model Management**.
- Click **For model details click here** from one of the model-building stages.
- Use a a web browser:
  - Open a web browser and go to the Spectrum™ Technology Platform Machine Learning Model Management page at:

    <servername>:<port>/machinelearning

    For example, if you installed Spectrum™ Technology Platform on a computer named "myspectrumplatform" and it is using the default port 8080, you would go to:

    myspectrumplatform:8080/machinelearning
  - Enter a valid Spectrum™ Technology Platform username and password.

Model Assessment

Introduction to Model Assessment

The Model Assessment tab in Machine Learning Model Management shows a list of all machine learning models on your Spectrum™ Technology Platform server. You can filter this list by entering a string in the text box; every field in the table will be searched for that string.

Several operations can be performed on these models. You can import, export, expose, unexpose, or delete models. Exposed models are used in the Java Model Scoring stage to score new data using formulas created when you fit machine learning models. Additionally, you can view detailed information for each model; the details returned depend on the type of model whose data you are viewing. Finally, you can compare any two models of the same type. This comparison shows side-by-side the same information that is on the Model Detail tab for each of the models you are comparing.

Model Assessment Operations

Perform these operations by selecting a model and clicking the appropriate button:

🗋	View model output detail. You can also access this information from the K-Means Clustering and Logistic Regression stages by clicking "For model details click here" on the Model Output tab.
⫼	Compare models.
→]	Import a model from a specific path. Select whether to overwrite an existing model of the same name, if appropriate.
[→	Export a model to a specific path. Select whether to overwrite an existing model of the same name, if appropriate.
⇧	Expose the model to make it available to the Java Model Scoring stage. If a model is not exposed, it cannot be used for scoring.
⇩	Unexpose the model.
🗑	Delete the model. **Note:** You cannot delete an exposed model; however, at this time there is no inherent security that prevents a user from deleting another user's models.

Model Detail Tab

The Model Detail screen shows the following information for all models:

- **Model Name**—The name of the model
- **Model Type**—The type of machine learning model
- **User**—The username of the person who created the model
- **Description**—The description of the model if one was provided when it was created

- **Status**—Whether the model is exposed or unexposed
- **Dataflow Name**—The name of the dataflow that produced the model
- **Creation Time**—The date and time the model was created

Additional details are provided based on the model type.

K-Means Clustering Details

The Model Detail screen includes the following information for K-Means Clustering models:

Model Summary

Provides training data for the following:

- Number of Rows
- Number of Clusters
- Number of Categorical Columns
- Number of Iterations
- Within Cluster Sum of Squares
- Total Sum of Squares
- Between Cluster Sum of Squares

Metrics

Provides training, test, and n-fold data for the following:

- Total within cluster sum of squares
- Total sum of squares
- Between cluster sum of squares

Centroid Statistics

Provides the following training, test, and n-fold data for each centroid:

- Size
- Within cluster sum of squares

Cluster Means

Provides detailed information for each centroid. Content varies based on input data. A cluster is a group of observations from a data set identified as similar according to a particular clustering algorithm

Standardized Cluster Means

Provides standardized information for each centroid. Content varies based on input data.

Logistic Regression Details

The Model Detail screen includes the following information for Logistic Regression models:

Metrics

Provides training, test, and n-fold data for the following:

- Mean squared error (MSE)
- Root mean squared error (RMSE)
- Number of observations
- R-squared (R2)
- Logarithmic loss (Logloss)
- Area under the curve (AUC)
- Precision-recall area under the curve (PR AUC)
- Gini coefficient
- Mean per class error
- Akaike information criterion (AIC)
- Lambda
- Residual deviance
- Null deviance
- Null degree of freedom
- Residual degree of freedom

Maximum Metrics Threshold

Provides the Training Maximum Metrics Threshold for training, test, and n-fold data using the following metrics:

- max f1
- max f2
- max f0point5
- max accuracy
- max precision
- max recall
- max specificity
- max absolute_mcc
- max min_per_class_accuracy
- max mean_per_class_accuracy

Confusion Matrix

Illustrates the performance of a model on a set of training, test, and n-fold data for which the true values are known.

Standardized Coefficient Chart

Shows the most important predictors by providing the relative value of the coefficients, which indicates how much a change in input changes the objective.

GLM Coefficients

Shows coefficients for a Generalized Linear Model, which estimates regression models for outcomes following exponential distributions.

AUC Curves

Area under the curve; determines which of the used models predicts the classes best using training, test, and n-fold data.

Lift/Gain Curves

Evaluate the prediction ability of a binary classification model using training, test, and n-fold data.

Linear Regression Details

The Model Detail screen includes the following information for Linear Regression models:

Metrics

Provides training, test, and n-fold data for the following:

- Mean squared error (MSE)
- Root mean squared error (RMSE)
- Number of observations
- R-squared (R2)
- Mean residual deviance
- Mean absolute error (MAE)
- Root mean squared logarithmic error (RMSLE)
- Akaike information criterion (AIC)
- Lambda
- Residual deviance
- Null deviance
- Null degree of freedom
- Residual degree of freedom

Standardized Coefficient Chart

Shows the most important predictors by providing the relative value of the coefficients, which indicates how much a change in particular predictor coefficient value changes the objective value positively or negatively. Also charts the top 25 coefficients in the model.

GLM Coefficients

Shows coefficients for a Generalized Linear Model, which estimates regression models for outcomes following exponential distributions.

Random Forest Regression Details

The Model Detail screen includes the following information for Random Forest Regression models:

Metrics

Provides training, test, and n-fold data for the following:

- Mean squared error (MSE)
- Root mean squared error (RMSE)

- Number of observations
- R-squared (R2)
- Mean residual deviance
- Mean absolute error (MAE)
- Root mean squared logarithmic error (RMSLE)

Variable Importances

Provides importance values for each variable using the following metrics:

- Relative importance
- Scaled importance
- Percentage

Also charts the top 25 variables in the model.

Random Forest Classification Details—Binomial

The Model Detail screen includes the following information for **binomial** Random Forest Classification models:

Metrics

Provides training, test, and n-fold data for the following:

- Mean squared error (MSE)
- Root mean squared error (RMSE)
- Number of observations
- R-squared (R2)
- Logloss
- Area under the curve (AUC)
- Precision-recall area under the curve (PR AUC)
- Gini
- Mean per class error

Maximum Metrics Threshold

Provides the Training Maximum Metrics Threshold for training, test, and n-fold data using the following metrics:

- max f1
- max f2
- max f0point5
- max accuracy
- max precision
- max recall
- max specificity
- max absolute_mcc

- max min_per_class_accuracy
- max mean_per_class_accuracy

Confusion Matrix

Illustrates the performance of a model on a set of training, test, and n-fold data for which the true values are known.

Variable Importances

Provides importance values for each variable using the following metrics:

- Relative importance
- Scaled importance
- Percentage

Also charts the top 25 variables in the model.

AUC Curves

Area under the curve; determines which of the used models predicts the classes best using training, test, and n-fold data.

Lift/Gain Curves

Evaluates the prediction ability of a binary classification model using training, test, and n-fold data.

Random Forest Classification Details—Multinomial

The Model Detail screen includes the following information for **multinomial** Random Forest Classification models:

Metrics

Provides training, test, and n-fold data for the following:

- Mean squared error (MSE)
- Root mean squared error (RMSE)
- Number of observations
- R-squared (R2)
- Logloss
- Mean per class error

Confusion Matrix

Illustrates the performance of a model on a set of training, test, and n-fold data for which the true values are known.

Variable Importances

Provides importance values for each variable using the following metrics:

- Relative importance
- Scaled importance

- Percentage

Also charts the top 25 variables in the model.

Principal Component Analysis Details

The Model Detail screen includes the following information for PCA models:

Importance of components

Shows the principal components in order of importance based on the following metrics:

- Standard deviation
- Proportion of variance
- Cumulative proportion

Rotation

Charts the matrix of variable loadings, the weight by which each standardized original variable should be multiplied to get the component score.

4 - Analytics Scoring Repository

In this section

Introduction to the Analytics Scoring Repository

The Analytics Scoring Repository is where analytical models are saved for use with the PMML Model Scoring stage. There is a web browser client for managing the models stored in the repository.

> **Note:** For details of the supported model types and type mappings see **Supported Model Formats** on page 12

Accessing the Analytics Scoring Repository Web Browser Client

In a web browser enter the URL http://<servername>:<port>/analyticsscoring (e.g. http://localhost:8080/analyticsscoring) to display the **Login** page. Enter a valid Spectrum™ Technology Platform username and password in the browser dialog box displayed. The browser will navigate to a page listing the existing Analytics models persisted in the repository.

View Models

This procedure describes how to view the list of models that are currently in the Analytics Scoring Repository .

1. Log in to the Analytics Scoring Repository web browser client by entering a valid Spectrum™ Technology Platform username and password in the **Login** page. On successful login, the application will navigate to the **Repository** page showing a table listing the models in the Analytics Scoring Repository that are available for use by the PMML Model Scoring stage.

   For details of how to log in via the **Login** page see **Introduction to the Analytics Scoring Repository** on page 40
2. Select the number of models to be shown per page from the **Models per page** drop-down below the model table. There is the option of displaying 10, 20, or 50 models per table page.
3. The page control beside the **Models per page** drop-down is used to navigate between the different pages.

   The page control will only allow page changes when the value chosen in step 2 for the number of models per page is less than the total number of models in the Analytics Scoring Repository .
4. Click on the **Name** column header to toggle sorting of listed models by the **Name** value in ascending/descending order. Alternatively click on the **Type** column header to toggle sorting of listed models by the **Type** value in ascending/descending order.

   Before any column headers are clicked the models by default are sorted by **Name** value in ascending order.

View Model Details

This procedure describes how to view details about a model listed in the Analytics Scoring Repository to view it's description and details of the inputs and outputs for the model.

1. Find the model you are looking for in the **Repository** page.

   For details on how to browse the Analytics Scoring Repository see **View Models** on page 40

2. Navigate to the model's detail page by either clicking on model's name in the **Name** column of the table or checking the model's checkbox and clicking the **Model details** button above the table.

   The details page displays detailed information about the model.

   Name the name of the model

   Type type of the model as described in **Supported Model Formats** on page 12

   Description a short text about the purpose of the model

   Inputs a table containing information about all the required fields for the model. Each row contains a field's name and type as described in **Supported Model Formats** on page 12

   Outputs a table containing information about all the output fields produced by the model. Each row contains a field's name and type as described in **Supported Model Formats** on page 12

3. To navigate back to the **Repository** page click the **Close** button in the top-right corner.

Add a Model via Web Browser Client

This procedure describes how to add a new model in to the Analytics Scoring Repository via the Analytics Scoring Repository web browser client in order to make the model available for deployment in a Spectrum™ Technology Platform dataflow.

1. Using the **View Models** on page 40 procedure, navigate to the **Repository** page.
2. Click the **Publish a model** button (located directly above the model table) to navigate to the **Add new model** page.

   The **Publish a model** button will be disabled if any models are selected in the model table’s current page. De-select these models to re-enable the **Publish a model** button.

3. On the **Add new model** page there are three input fields: **Name**, **Description** and **Upload model**. Write the name of the new model in to the text box labeled **Name**.

   Model names must be unique. If the model name inputted is already in use by another non-deleted model stored in the Analytics Scoring Repository , the text box will be highlighted with a validation error.

   Note: White space at the start or end of a model's name is always removed. This means if the name entered is already in use, adding white space to the start or end of the name will not resolve the error.

   Note: Model names are case sensitive. This means models can share the same name so long as they have different uppercase and lowercase characters. For example models named "my Model", "my model" and "mY ModEl" can co-exist simultaneously within the repository.

   Note: Name validation only detects conflicts against the contents of the Analytics Scoring Repository when the page was opened. This means if the repository has changed since navigating to the **Add new model** page e.g. models have been added or deleted by other users, then the page validation will not have these changes. However, on clicking the **Publish** button, the new model will be re-validated against the up-to date version of the Analytics Scoring Repository .

4. Write a description for the new model in to the text area labeled **Description**.

   Models must have a description.

5. Add a model definition by clicking the **Choose file** button and selecting a model stored in either **QMML** on page 12 or **PMML** on page 13 file formats.

   Note: The selected model must be less than 50MB in size.

   Note: When using an older browser that does not support file picking, the XML model definition must be input into the text area labeled **Upload model**.

   Note: If the selected model is a **QMML** on page 12 model, the **QMML** on page 12 must have at least one compiled live ruleset for the model to be successfully published. In the case where there are multiple compiled live rulesets the first such ruleset found within the **QMML** on page 12 will be used for the model definition.

   Note: If you wish to compile and/or change the live status of rulesets within your **QMML** on page 12 (or indeed add or delete rulesets), you can use the Spectrum Miner qsqmmledit command-line utility within your Spectrum Miner installation (help for the utility can be displayed in your command-line window by typing "qsqmmledit.exe -help").

   Note: The model definition will not be validated until clicking the **Publish** button.

6. Click the **Publish** button to begin uploading the model to the Analytics Scoring Repository .

The **Publish** button will be disabled until the new model has a valid name and a description and model definition specified.

7. Wait as the **Publish** button’s busy icon is shown. No changes can be made to the new model’s name, description or model definition while the model is being published. Once the model has been successfully published in to the Analytics Scoring Repository the page will navigate to the new model's **Details** page.

   The **Cancel** button will be disabled while the model is being published.

   > **Note:** If a problem is found while publishing the model e.g. the XML model definition is invalid or the model name is already in use, then the model will not be published and an alert error will be displayed detailing the problem. Click **OK** to close the error alert and correct the problem before attempting to publish the model again.

Add a Model via Spectrum Miner Spectrum Connector

The Spectrum Connector integration component enables the publishing of **QMML** on page 12 models into the Spectrum Analytics Scoring Repository directly from Spectrum Miner.

The Spectrum Connector integration component can be installed as an addition to any existing Spectrum Miner version 7.0B (or above) installation, including 'desktop' installs.

There are three different methods of publishing models to the Analytics Scoring Repository using the Spectrum Connector integration component:

Publishing QMML Models from Spectrum Miner on page 43

Publishing QMML Models from Decision Studio Model Builder tools on page 44

Publishing a QMML model from the Spectrum Miner command line tool on page 44

Publishing QMML Models from Spectrum Miner

This procedure describes how to add a new model in to the Analytics Scoring Repository using Spectrum Miner.

1. Open Spectrum Miner and in the File Browser tab navigate to the directory containing the .qmml file you wish to publish.
2. Right click on the .qmml file and select 'Publish -> to Spectrum Analytics Scoring Repository '.
3. When prompted enter a valid Spectrum username and password to login into the Spectrum platform. The login prompt will be displayed once per Spectrum Miner session with any subsequent model publishing using the credentials previously provided.

4. In the model details tab specify the model name, model description and **QMML** on page 12 ruleset for the model to be published into the Spectrum Analytics Scoring Repository .

   The name given to the model must be unique within the Spectrum Advanced Analytics model repository.
5. Click the 'Go' button.

Publishing QMML Models from Decision Studio Model Builder tools

This procedure describes how to add a new model in to the Analytics Scoring Repository directly from one of the Decision House model building tools.

1. In the chosen Decision House Model Builder tool, build your model.
2. Press the 'Publish [type of model] to Spectrum' button that was enabled after the model build completed in the previous step.
3. When prompted enter a valid Spectrum username and password to login into the Spectrum platform. The login prompt will be displayed once per Spectrum Miner session with any subsequent model publishing using the credentials previously provided.
4. In the model details tab specify the model name, model description and the output fields for the model to be published into the Spectrum Analytics Scoring Repository .

   The name given to the model must be unique within the Spectrum Advanced Analytics model repository.

   Note: Only fields with their associated checkbox selected will be included as output fields in the model. At least one field must be selected.
5. Click the 'Publish' button.

Publishing a QMML model from the Spectrum Miner command line tool

This procedure describes how to add a new model in to the Analytics Scoring Repository using the spectrumpublish data build command line tool provided by Spectrum Miner's Spectrum Connector integration component.

At a command prompt, execute the spectrumpublish utility e.g. '$QSHOME/server/qs7.0B/win32/bin/spectrumpublish -input [qmml file path] -name [model name] -description [model description] -ruleset [ruleset name] -username [Spectrum username] -password [password]

Note: The -input, -name, -username, -password and -description arguments must be specified. The -ruleset argument is optional.

> **Note:** In the case where the supplied **QMML** on page 12 file contains multiple rulesets but the -ruleset argument is not specified, then only the first ruleset within the **QMML** on page 12 document is published into the Spectrum Analytics Scoring Repository .

Delete a Model

This procedure describes how to delete models from the Analytics Scoring Repository . Deleting a model from the repository will prevent the model from being deployed to any new Spectrum™ Technology Platform dataflows. Deleting a model will not affect the runtime operation of any existing dataflows that are configured to use the model although no modifications can be made to the settings of a PMML Model Scoring stage within a dataflow that is configured to use the model.

1. Using the **View Models** on page 40 procedure, navigate to the **Repository** page.

   The **Repository** page displays models stored within the Analytics Scoring Repository via a model table. If a large number of models are stored within the Analytics Scoring Repository then the model table may display models across several pages.
2. Using the model table select models to be deleted from the Analytics Scoring Repository .

   > **Note:** Only models selected on the model table's current page will be deleted. For example, if the model table has two pages of models and models have been selected on both pages, then clicking the delete button will only delete the models on the current page and ignore any models selected on the other page.
3. Click the **Delete model** button, located directly above the model table.

   The **Delete model** button will be disabled when no models are selected on the model table’s current page.
4. A pop-up message is displayed asking for confirmation to delete the selected model or models from the Analytics Scoring Repository . Click the **Yes** button to confirm and delete the selected models from the Analytics Scoring Repository . Click the **No** button to cancel the delete operation.

   After successfully deleting the selected models from the Analytics Scoring Repository the **Repository** page will refresh and update the model table.

Notices

USPS® Notices

Pitney Bowes Inc. holds a non-exclusive license to publish and sell ZIP + 4® databases on optical and magnetic media. The following trademarks are owned by the United States Postal Service: CASS, CASS Certified, DPV, eLOT, FASTforward, First-Class Mail, Intelligent Mail, LACS^{Link}, NCOA^{Link}, PAVE, PLANET Code, Postal Service, POSTNET, Post Office, RDI, Suite^{Link} , United States Postal Service, Standard Mail, United States Post Office, USPS, ZIP Code, and ZIP + 4. This list is not exhaustive of the trademarks belonging to the Postal Service.

Pitney Bowes Inc. is a non-exclusive licensee of USPS® for NCOA^{Link}® processing.

Prices for Pitney Bowes Software's products, options, and services are not established, controlled, or approved by USPS® or United States Government. When utilizing RDI™ data to determine parcel-shipping costs, the business decision on which parcel delivery company to use is not made by the USPS® or United States Government.

Data Provider and Related Notices

Data Products contained on this media and used within Pitney Bowes Software applications are protected by various trademarks and by one or more of the following copyrights:

The Geocode Address World data set contains data licensed from the GeoNames Project (**www.geonames.org**) provided under the Creative Commons Attribution License ("Attribution License") located at **http://creativecommons.org/licenses/by/3.0/legalcode**. Your use of the

3001 Summer Street
Stamford CT 06926-0700
USA

www.pitneybowes.com