ZEMENTIS for Datameer

User Guide

10.2.0.1

ZEMENTIS for Datameer

User Guide

Table of Contents

List of Figures

List of Tables

Chapter 1. Introduction

As advanced analytics becomes pervasive across the enterprise to drive better business decisions, the need for efficient execution of predictive models is paramount. An ever growing array of data mining tools and, all too often, custom specialized software is used to mine data and derive statistical models from a wealth of collected data. The ultimate goal is to turn these models into business value by incorporating them into day to day business operations. This necessitates the ability to integrate them into the IT infrastructure where the outcomes can easily flow into the finger-tips of the decision makers. At the same time, the accelerating growth rate of data collected implies that only the most scalable database architectures will be able to meet storage, and more importantly, processing requirements.

In the era of big data, more and more organizations are turning to the scalable architecture of Hadoop to meeting this growing challenge. To bring the power of predictive models into this architecture, Software AG and Datameer® have joined forces to deliver the ZEMENTIS Predictive Analytics (ZEMENTIS) for Datameer®. ZEMENTIS offers Datameer® users the best combination of open standards and scalability for the application of predictive analytics. With the Predictive Modeling Mark-up Language (PMML) as the bridge between the model development environment and the IT data warehousing infrastructure, ZEMENTIS for Datameer® offers standards-based deployment of predictive models and execution on a highly scalable analytics platform. This joint solution combines the power of ZEMENTIS Predictive Analytics server, the flagship product of Software AG, with that of Datameer®, the only analytics application natively built on Hadoop. As a result, a wide range of predictive models, possibly developed with different tools in different environments, can fully leverage the scalability and flexibility of Hadoop to bring game changing data integration and analytics to end users. Practically, PMML becomes yet another analytical function available in Datameer® offering execution performance that can meet the volume and performance requirements of the most demanding environments.

This document serves as a guide for installing and using ZEMENTIS for Datameer®. It first gives a brief overview of the plug-in. It then presents the simple installation process. Finally, it illustrates the use of ZEMENTIS with two PMML models showing how to upload, deploy and score them using ZEMENTIS for Datameer®.

Chapter 2. Overview

2.1. Predictive Model Markup Language (PMML)

As the de-facto standard for data mining models, PMML provides tremendous benefits for business, IT, and the data mining industry in general. Developed by the Data Mining Group (DMG), an independent, vendor-led consortium, PMML increases business agility by eliminating the need for proprietary solutions or custom code development.

Today, it is supported by all major data mining tools, commercial and open source. As an open standard, it enables project stakeholders to standardize on one common representation for data mining models. It practically eliminates the barriers and gaps between development and production deployment of predictive analytics. In effect, it minimizes the complexity, cost, and time to turn predictive models into operational IT and business assets.

As the lingua franca for predictive analytics, data mining models can be easily exchanged between PMML-compliant applications. In this way, a model may be built in one statistical tool and easily moved to another for production deployment or visualization. PMML also serves as a bridge between all the teams involved in the data mining process inside a company since it can be used to disseminate knowledge and best practices, thereby stimulating cross-team and inter-organizational collaboration. In a world in which data-driven decisions are becoming more and more pervasive, predictive analytics and standards such as PMML make it possible for organizations to benefit from smart solutions that will truly revolutionize their business.

Besides offering a rich set of structures for describing all the intricate details of a predictive algorithm, PMML also provides information about the inputs and outputs of a model. This includes names and types of all input and output data fields, often along with the set of permissible values. In addition, a model expressed in PMML typically includes information about how to handle invalid or missing input values.

Note

A variety of sample PMML models are included with the ZEMENTIS distribution package. In addition, a wealth of resources on PMML can be found from the PMML in Action.

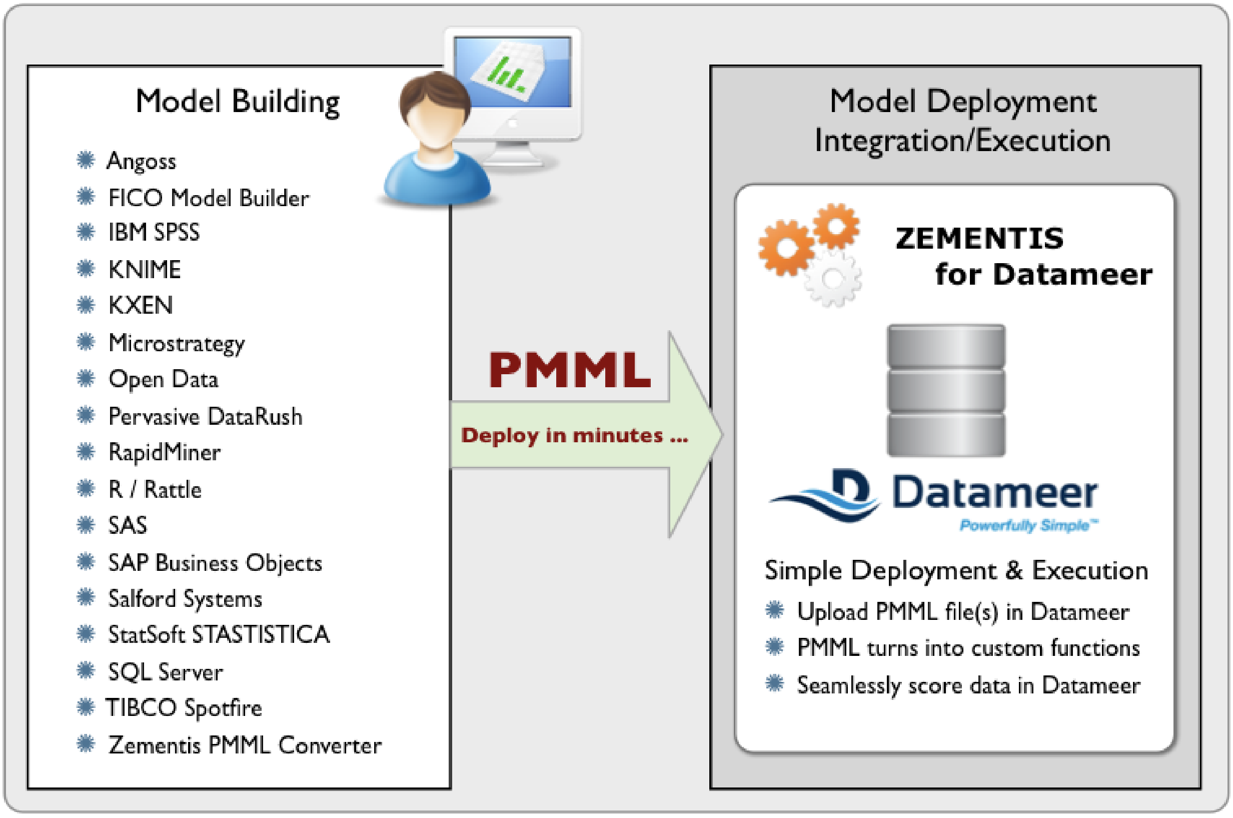
2.2. ZEMENTIS Predictive Analytics (ZEMENTIS)

ZEMENTIS enables execution of standards-based predictive analytics directly within Datameer®. It shares the PMML execution core with the ZEMENTIS server offered by Software AG.

In addition, the plug-in takes on the responsibility of bridging the PMML and the Datameer®/Hadoop world. This means that it presents each loaded PMML model as a Datameer® function. The name, input arguments and outputs of each function match the name, input fields, and output fields of the corresponding model as defined in the PMML file. This way, scoring a data set requires nothing more than invoking the analytics function correspondent to the appropriate predictive analytics model. Predictions (scores, probabilities, categories, clusters, etc.) can be just as easily become part of further analytics inspection, or plotted with the use of dashboards.

Like the ZEMENTIS server, ZEMENTIS plugin accepts PMML models of all versions (2.0, 2.1, 3.0, 3.1, 3.2, 4.0, 4.1, 4.2 and 4.3) generated by any of the major commercial and open source data mining tools.

Figure 2.1. Overview of the ZEMENTIS for Datameer®



At a high level, the process of using PMML models in Datameer® is presented in Figure 2.1. It starts after the predictive models have been created and have been exported in PMML format from the data mining and model development tool. With the PMML files at hand, all it takes is to upload and deploy them into ZEMENTIS for Datameer® where they are translated into functions.

The plug-in supports a wide range of PMML elements that implement the following predictive analytic techniques:

- Decision Trees for classification and regression
- K-Nearest Neighbors for regression, classification and clustering
- Neural Network Models: Back-Propagation, Radial-Basis Function, and Neural-Gas

- Support Vector Machines for regression, binary and multi-class classification
- Linear and Logistic Regression (binary and multinomial)
- Naïve Bayes Classifiers
- General and Generalized Linear Models
- Cox Regression Models
- Rule Set Models (flat decision trees)
- Clustering Models: Distribution-Based, Center-Based, and 2-Step Clustering
- Scorecards (including reason codes and point allocation for complex attributes)
- Association Rules (mining schema attribute `usageType = "group"` is not supported)
- Model Segmentation
- Model Ensemble (including Random Forest Models)
- Model Composition and Chaining

In addition, it implements all the PMML built-in functions and transformations for data pre- and post-processing.

Note

Note that in addition to association rules which use `usageType = "group"`, ZEMENTIS for Datameer® does not currently support time series, text models, and sequence models.

Chapter 3. Installation

This section describes how to install the ZEMENTIS for Datameer®.

3.1. Requirements

In order to install ZEMENTIS for Datameer® on your system, you will need an existing Datameer® installation. Uploading new PMML models with ZEMENTIS for Datameer® requires a valid Product License Key which can be obtained by contacting Software AG.

3.2. Packaging

ZEMENTIS for Datameer® is distributed as a compressed archive file: `uppi-datameer6-10.2.0.1.zip`. The distributed package consists of several files, including this documentation and several sample files. When uncompressed, the package reveals a number of directories as described in Table 3.1.

Table 3.1. Directory Structure of the ZEMENTIS for Datameer® package.

Directory	Contents
`docs`	Documentation in HTML and PDF format.
`pmml`	A number of sample PMML files along with data files in CSV format. These include the examples described in this document.
`classes`, `lib`	Runtime code and library used by ZEMENTIS for Datameer®.
`license`	License information for ZEMENTIS and 3rd party libraries.

3.3. New Installation

To install ZEMENTIS for Datameer® navigate your browser to "Administration" tab and select "Plugins" option from left-side menu. On the bottom of the page you will find "Upload Plug-in" panel. Click on "Choose File" button, select `uppi-datameer6-10.2.0.1.zip` file, and click on "Upload" button. If installation was successful, a green message box will be displayed, and if it failed, a red message box will be displayed. Also, on the same page in the "Plug-ins" panel which displays table of all installed plugins you will find new row entry with "ZEMENTIS Plug-In" in "Name" column. Lastly, place the Product License Key file, `zementis.license`, in the Datameer® root directory. Please note that execution of existing models will not be interrupted when the license expires.

Note

Note that if you have installed Datameer® in a computer running Mac OS X, you will need to right-click on top of the Datameer® application package and choose "Show Package Contents" in order to have access to its directory structure.

Once the ZEMENTIS for Datameer® is installed, you can go ahead and upload and deploy your predictive models. This is described in Chapter 4.

3.4. Upgrade Installation

Upgrading existing ZEMENTIS for Datameer® installation involves few steps to ensure that Datameer® worksheets with currently deployed PMML functions will remain operational with newer version of ZEMENTIS for Datameer®. First, you have to make sure that copies of all PMML files corresponding to PMML functions are backed up and available for re-upload. If this is the case, remove all PMML functions using ZEMENTIS configuration interface. Details on PMML function removal and deployment operations can be found in Chapter 4. Next, remove ZEMENTIS for Datameer® by navigating to "Administration" tab, selecting "Plugins" option from left-side menu, and then clicking on thrash icon in "ZEMENTIS Plug-In" entry of "Plug-Ins" panel. Install new ZEMENTIS for Datameer® by uploading `uppi-datameer6-10.2.0.1.zip` file through "Upload Plug-in" panel at the bottom of the same page. Finally, restore all PMML functions by uploading each PMML file using ZEMENTIS configuration interface.

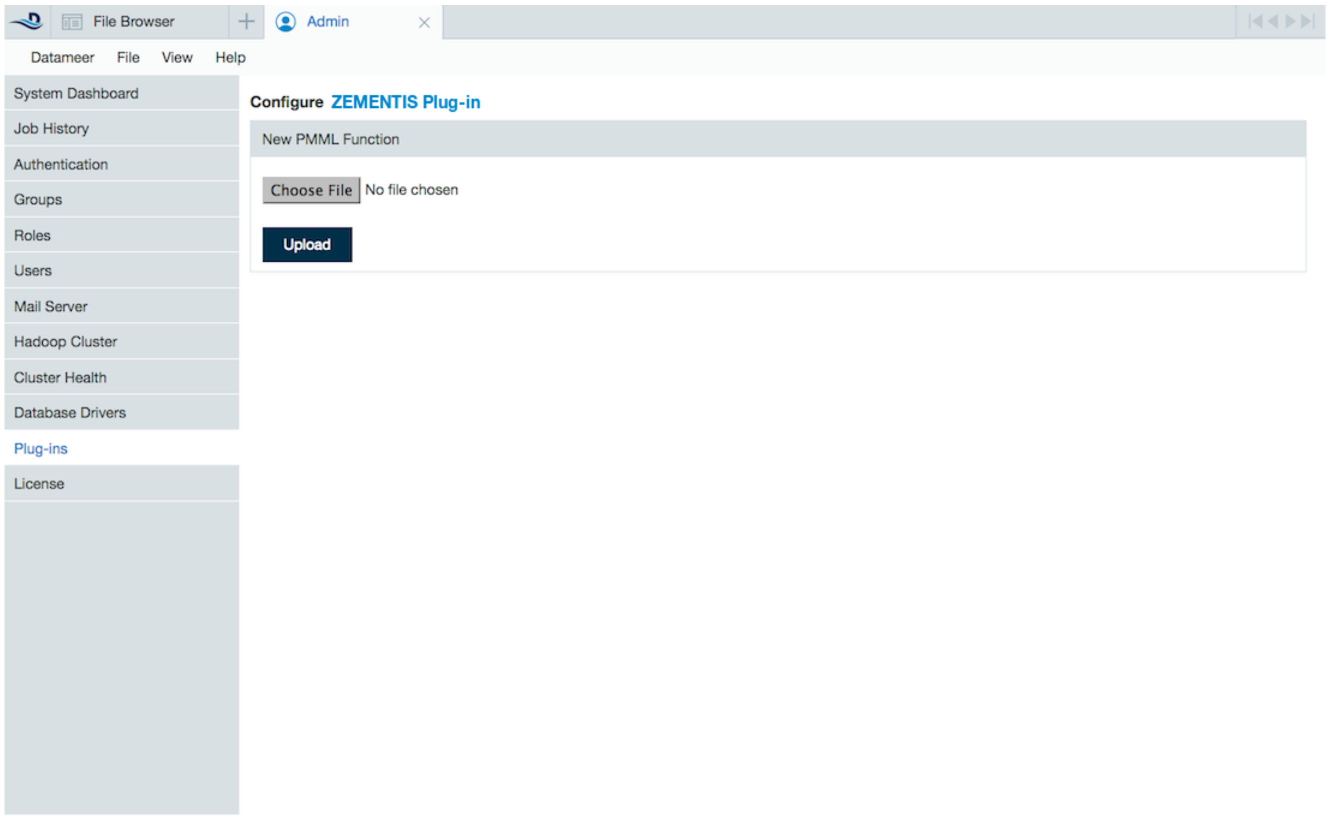
Chapter 4. From PMML to Datameer®

This section will describe in detail how PMML models are made available to be used directly in Datameer®. As explained earlier, ZEMENTIS turns predictive models into functions which can be readily used from inside Datameer® as any other function available through the "Formula Builder".

4.1. Configure ZEMENTIS for Datameer®

To make your PMML models available in Datameer®, go to the "Administration" tab and select the "Plug-ins" option from the left-side menu. In the table of available plug-ins, locate the ZEMENTIS Plug-In and click on the "Configure" hyperlink. Once you do that, you will be faced with a screen as shown in Figure 4.1 . It is here that you will upload and deploy your PMML models.

Figure 4.1. Configure ZEMENTIS for Datameer®



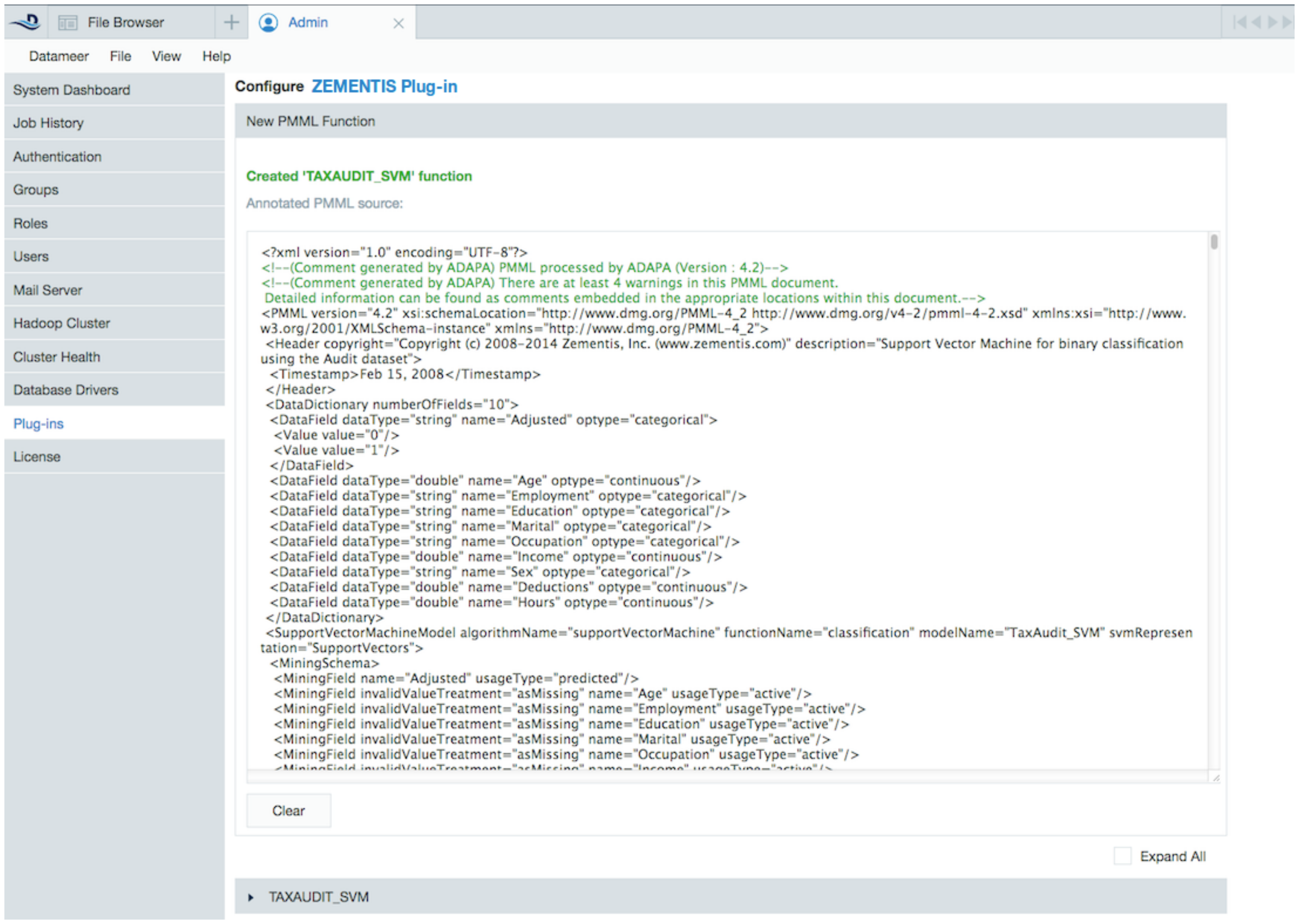
In this section, we describe the process of uploading and deploying PMML files in ZEMENTIS for Datameer®. These files represent two distinct predictive models.

The first model is a Support Vector Machine (SVM) trained with the Audit dataset. This model will produce a single output field which will contain "0" in case the taxpayer's claim does not need to be adjusted or "1" in case it does. The audit dataset is supplied as part of the R Rattle package - http://rattle.togaware.com (it is also available for download as a CSV file from http://rattle.togaware.com/audit.csv). It consists of fictional clients who have been audited, perhaps for tax refund compliance. For each case an outcome is recorded (whether the taxpayer's claims had to be adjusted or not).

The second model is a Neural Network (NN) model trained with the Iris dataset. This model will produce four output fields, the class or type of Iris plant (setosa, virginica, or versicolor) with the computed probabilities for the three different flower types. The Iris dataset is perhaps the best known dataset to be found in the pattern recognition literature. It contains three classes representing different types of the Iris plant. Each class is represented by 50 records. For more information on the Iris dataset, please check the Iris page at the UCI Repository of Machine Learning Databases - http://archive.ics.uci.edu/ml/datasets/Iris (Asuncion, A. and Newman, D.J. (2007). UCI Machine Learning Repository. Irvine, CA: University of California, School of Information and Computer Science).

To make the Tax Audit SVM model available in Datameer®, simply click on the "Choose File" button and locate the corresponding PMML file, and click on the "Upload" button. Once the file is uploaded and model is successfully deployed, you should see the screen as shown in Figure 4.2. You can inspect PMML file for warnings highlighted in green by scrolling down text in the window under "Annotated PMML source" heading. Clicking "Clear" button causes this message panel to revert back to PMML file upload panel.

Figure 4.2. Uploading the Tax Audit SVM Model to Datameer®



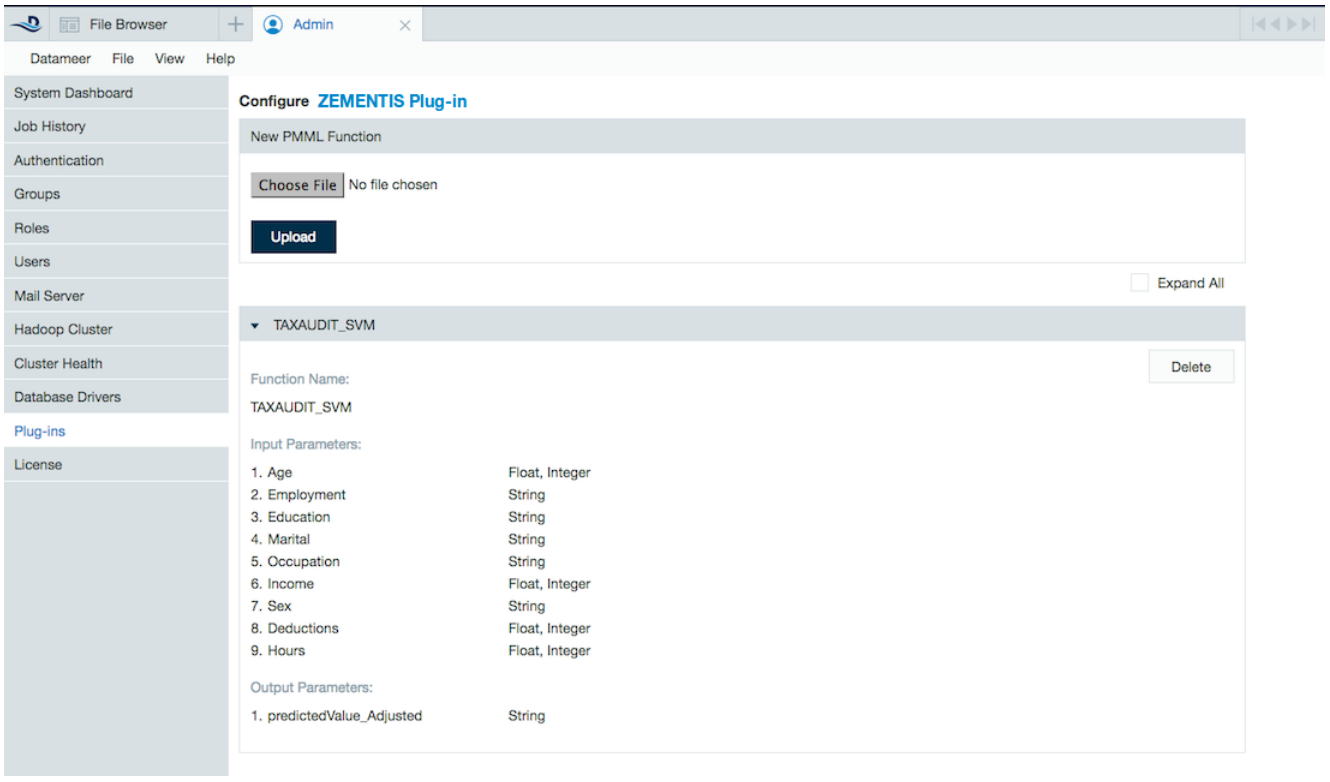
Created 'TAXAUDIT_SVM' function

Annotated PMML source:

```
<?xml version="1.0" encoding="UTF-8"?>
<!--(Comment generated by ADAPA) PMML processed by ADAPA (Version : 4.2)-->
<!--(Comment generated by ADAPA) There are at least 4 warnings in this PMML document.
 Detailed information can be found as comments embedded in the appropriate locations within this document.-->
<PMML version="4.2" xsi:schemaLocation="http://www.dmg.org/PMML-4_2 http://www.dmg.org/v4-2/pmml-4-2.xsd" xmlns:xsi="http://www.
w3.org/2001/XMLSchema-instance" xmlns="http://www.dmg.org/PMML-4_2">
 <Header copyright="Copyright (c) 2008-2014 Zementis, Inc. (www.zementis.com)" description="Support Vector Machine for binary classification
using the Audit dataset">
  <Timestamp>Feb 15, 2008</Timestamp>
 </Header>
 <DataDictionary numberOfFields="10">
  <DataField dataType="string" name="Adjusted" optype="categorical">
   <Value value="0"/>
   <Value value="1"/>
  </DataField>
  <DataField dataType="double" name="Age" optype="continuous"/>
  <DataField dataType="string" name="Employment" optype="categorical"/>
  <DataField dataType="string" name="Education" optype="categorical"/>
  <DataField dataType="string" name="Marital" optype="categorical"/>
  <DataField dataType="string" name="Occupation" optype="categorical"/>
  <DataField dataType="double" name="Income" optype="continuous"/>
  <DataField dataType="string" name="Sex" optype="categorical"/>
  <DataField dataType="double" name="Deductions" optype="continuous"/>
  <DataField dataType="double" name="Hours" optype="continuous"/>
 </DataDictionary>
 <SupportVectorMachineModel algorithmName="supportVectorMachine" functionName="classification" modelName="TaxAudit_SVM" svmRepresen
tation="SupportVectors">
  <MiningSchema>
   <MiningField name="Adjusted" usageType="predicted"/>
   <MiningField invalidValueTreatment="asMissing" name="Age" usageType="active"/>
   <MiningField invalidValueTreatment="asMissing" name="Employment" usageType="active"/>
   <MiningField invalidValueTreatment="asMissing" name="Education" usageType="active"/>
   <MiningField invalidValueTreatment="asMissing" name="Marital" usageType="active"/>
   <MiningField invalidValueTreatment="asMissing" name="Occupation" usageType="active"/>
   <MiningField invalidValueTreatment="asMissing" name="Income" usageType="active"/>
```

You can view a list of all deployed PMML functions under "PMML Functions" section. It contains expandable bars, each with function name on the left and delete button on the right. You can view parameters used by each model, such as list of inputs and outputs, by clicking on expandable bar or by clicking "Expand All" checkbox. The function name corresponding to a PMML file is derived from the PMML attribute "modelName". The input parameters reflect all the input fields necessary for the model to be executed. In case of the Tax Audit SVM model, these are: "Age", "Employment", "Education", "Marital", "Occupation", "Income", "Sex", "Deductions", and "Hours". The output parameters represent fields that the model returns as output. For the Tax Audit SVM model, the output parameter is field "predictedValue_Adjusted". The details of Tax Audit SVM model are shown in Figure 4.3,

Figure 4.3. Viewing Tax Audit SVM Model Details



Configure ZEMENTIS Plug-in

New PMML Function

TAXAUDIT_SVM

Function Name:

TAXAUDIT_SVM

Input Parameters:

1. Age	Float, Integer
2. Employment	String
3. Education	String
4. Marital	String
5. Occupation	String
6. Income	Float, Integer
7. Sex	String
8. Deductions	Float, Integer
9. Hours	Float, Integer

Output Parameters:

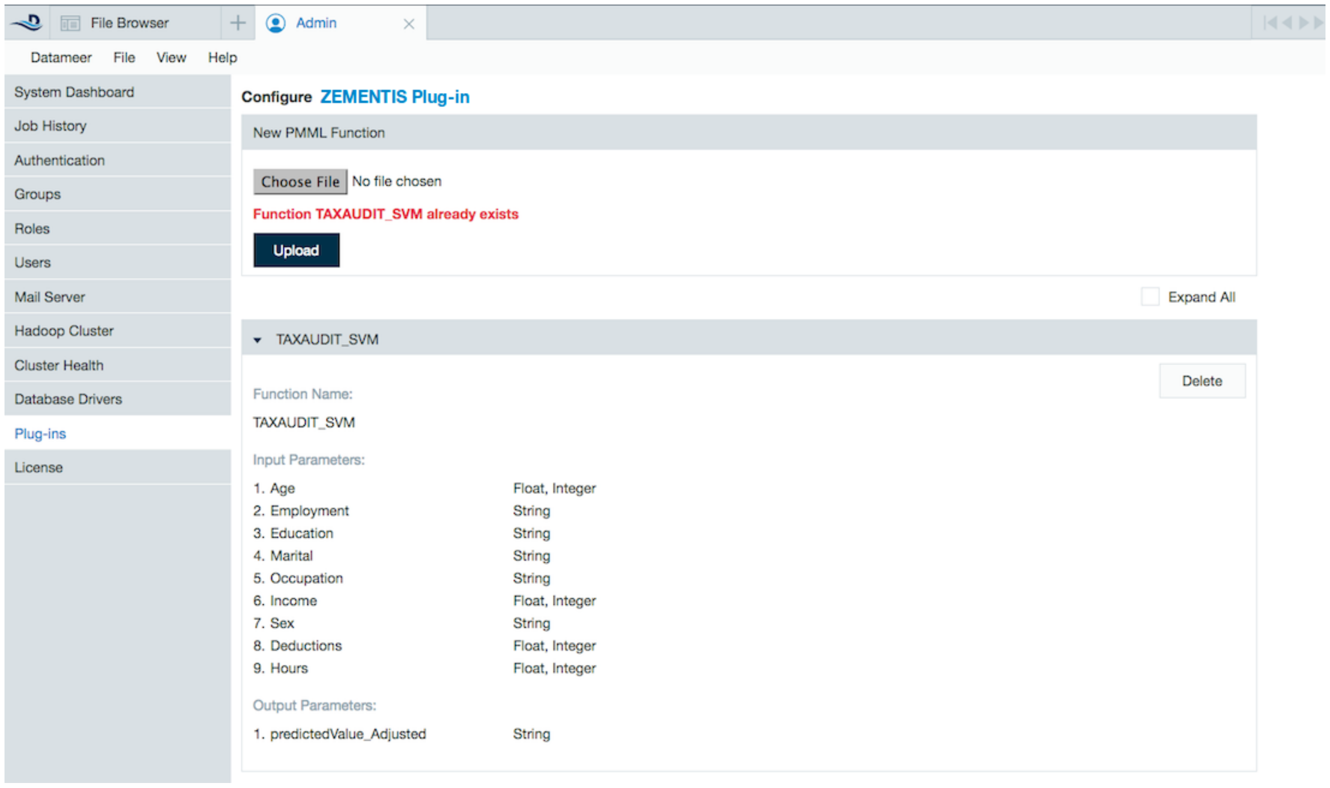
1. predictedValue_Adjusted	String

If you try to upload and deploy a model in ZEMENTIS for Datameer® that has already been uploaded, you will get an error notifying you that a Datameer® function with the same name already exists as shown in Figure 4.4. To avoid such an error, we recommend you delete the existing model before attempting to upload the new one.

Note

ZEMENTIS will also notify you of any errors if the PMML file you are trying to upload has any errors. In that case, we recommend you correct the issue(s) pointed out and upload the model again.

Figure 4.4. Configuration error in Datameer®



Configure ZEMENTIS Plug-in

New PMML Function

No file chosen

Function TAXAUDIT_SVM already exists

TAXAUDIT_SVM

Function Name:

TAXAUDIT_SVM

Input Parameters:

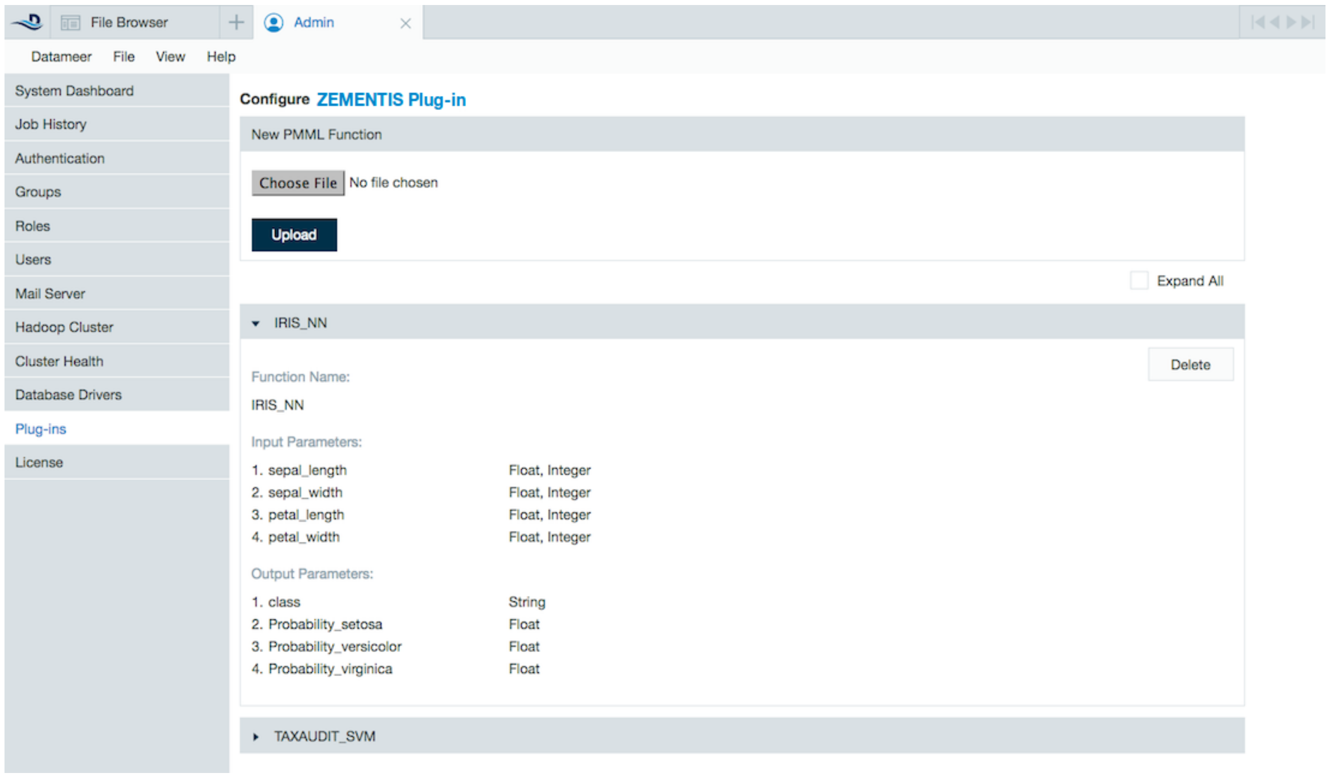
1. Age	Float, Integer
2. Employment	String
3. Education	String
4. Marital	String
5. Occupation	String
6. Income	Float, Integer
7. Sex	String
8. Deductions	Float, Integer
9. Hours	Float, Integer

Output Parameters:

1. predictedValue_Adjusted	String

To make the Iris NN model available in Datameer®, select the corresponding PMML file and click on the "Upload" button. Once the file is uploaded and model is successfully deployed, you should see a successful upload panel with the annotated source. After clicking on "Clear" button on the upload panel, you should see the screen as shown in Figure 4.5.

Figure 4.5. Uploading the Iris NN Model to Datameer®



Configure ZEMENTIS Plug-in

New PMML Function

IRIS_NN

Function Name:

IRIS_NN

Input Parameters:

1. sepal_length	Float, Integer
2. sepal_width	Float, Integer
3. petal_length	Float, Integer
4. petal_width	Float, Integer

Output Parameters:

1. class	String
2. Probability_setosa	Float
3. Probability_versicolor	Float
4. Probability_virginica	Float

TAXAUDIT_SVM

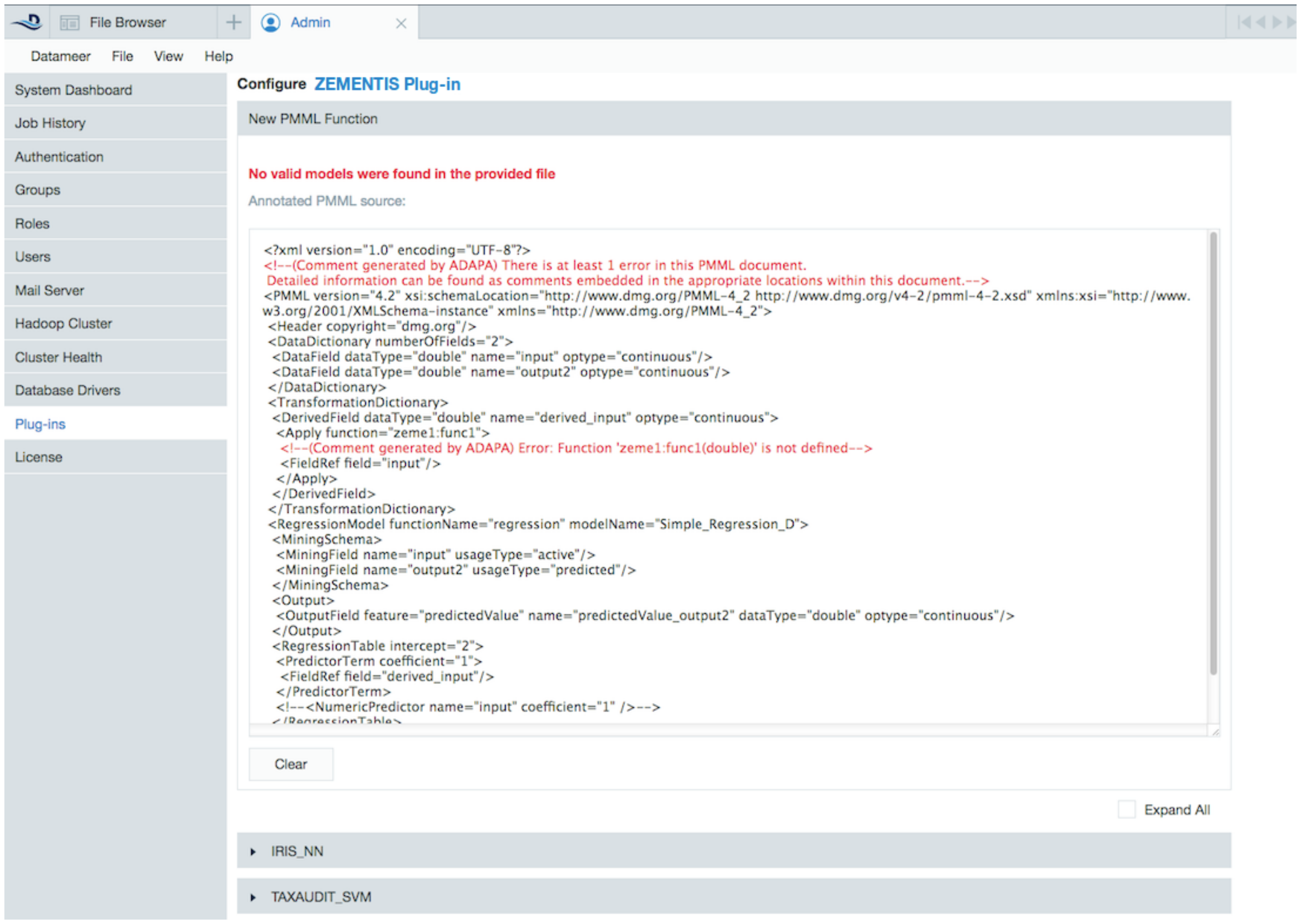
To continue uploading and deploying models in ZEMENTIS for Datameer® simply repeat the process described above. There is no limit to the number of models that can be uploaded. However, we recommend you delete models whenever they are no longer required. You can do that by clicking on "X" button located on the right side of each PMML function bar as shown in the screenshots above.

Note

Note that models are listed in alphabetical order in Datameer®, not the order in which they were uploaded. In this case, even if the Iris NN model had been uploaded first, it would still be shown after the Tax Audit SVM model.

Let's now consider a case where the PMML file for the Iris NN model had an error which could not be corrected automatically by ZEMENTIS when uploading the model. In this case, the model would not be deployed and instead you would get an error stating that the PMML file was not valid, as shown in Figure 4.6.

Figure 4.6. Error during model deployment



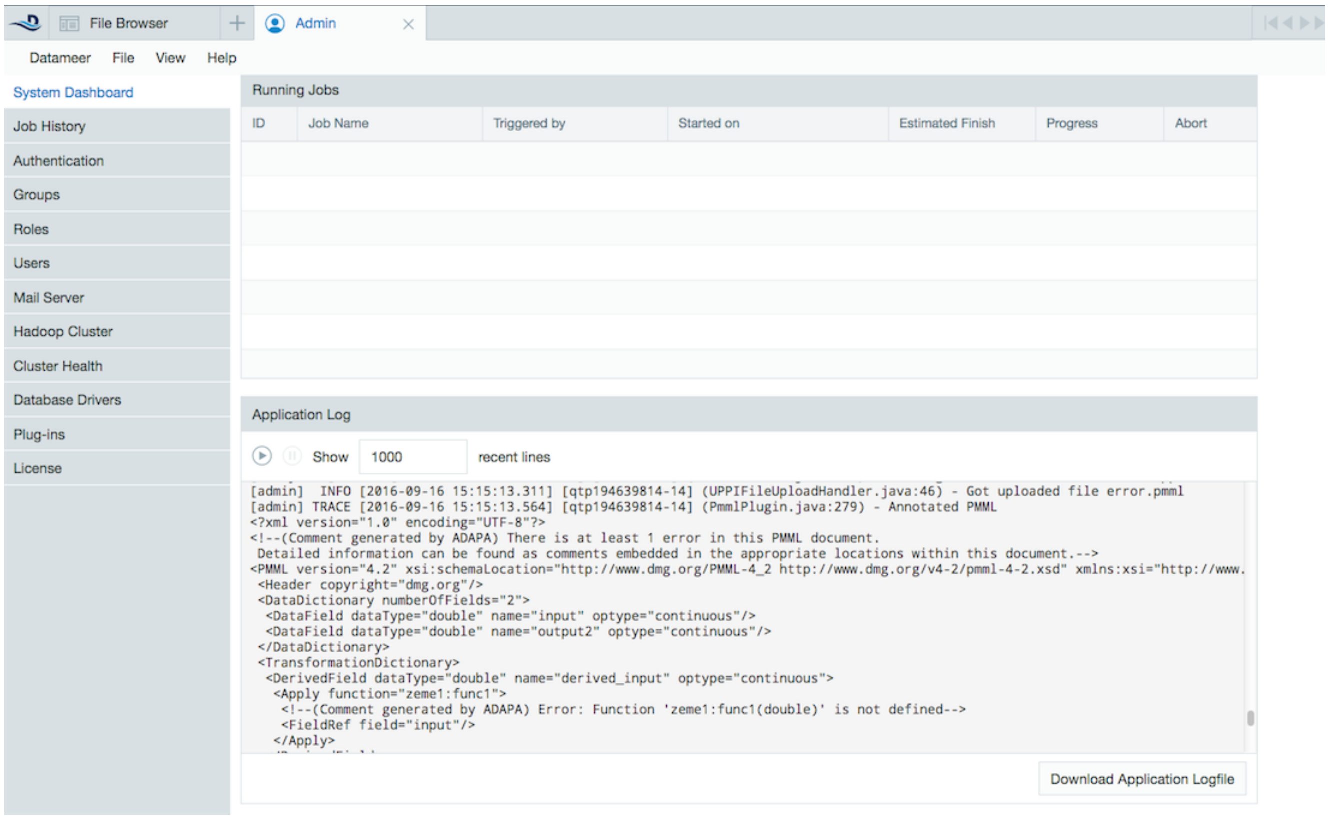
No valid models were found in the provided file

Annotated PMML source:

```
<?xml version="1.0" encoding="UTF-8"?>
<!--(Comment generated by ADAPA) There is at least 1 error in this PMML document.
 Detailed information can be found as comments embedded in the appropriate locations within this document.-->
<PMML version="4.2" xsi:schemaLocation="http://www.dmg.org/PMML-4_2 http://www.dmg.org/v4-2/pmml-4-2.xsd" xmlns:xsi="http://www.
w3.org/2001/XMLSchema-instance" xmlns="http://www.dmg.org/PMML-4_2">
 <Header copyright="dmg.org"/>
 <DataDictionary numberOfFields="2">
  <DataField dataType="double" name="input" optype="continuous"/>
  <DataField dataType="double" name="output2" optype="continuous"/>
 </DataDictionary>
 <TransformationDictionary>
  <DerivedField dataType="double" name="derived_input" optype="continuous">
   <Apply function="zeme1:func1">
    <!--(Comment generated by ADAPA) Error: Function 'zeme1:func1(double)' is not defined-->
    <FieldRef field="input"/>
   </Apply>
  </DerivedField>
 </TransformationDictionary>
 <RegressionModel functionName="regression" modelName="Simple_Regression_D">
  <MiningSchema>
   <MiningField name="input" usageType="active"/>
   <MiningField name="output2" usageType="predicted"/>
  </MiningSchema>
  <Output>
   <OutputField feature="predictedValue" name="predictedValue_output2" dataType="double" optype="continuous"/>
  </Output>
  <RegressionTable intercept="2">
   <PredictorTerm coefficient="1">
    <FieldRef field="derived_input"/>
   </PredictorTerm>
   <!--<NumericPredictor name="input" coefficient="1" />-->
  </RegressionTable>
```

Before a PMML file is deployed in ZEMENTIS for Datameer®, it is checked for syntactic and semantic problems. During this process, it may encounter errors, which prevent a model from being deployed, and/or minor issues which are reported as warnings. Errors and warnings are embedded into the uploaded PMML file and can be viewed in the configuration panel after model upload is complete. PMML models with errors are also available for inspection in Datameer® as part of the application log file and can be accessed through the "System Dashboard" as shown in Figure 4.7.

Figure 4.7. Inspecting errors and warnings



```
[admin]  INFO [2016-09-16 15:15:13.311] [qtp194639814-14] (UPPIFileUploadHandler.java:46) - Got uploaded file error.pmml
[admin] TRACE [2016-09-16 15:15:13.564] [qtp194639814-14] (PmmlPlugin.java:279) - Annotated PMML
<?xml version="1.0" encoding="UTF-8"?>
<!--(Comment generated by ADAPA) There is at least 1 error in this PMML document.
 Detailed information can be found as comments embedded in the appropriate locations within this document.-->
<PMML version="4.2" xsi:schemaLocation="http://www.dmg.org/PMML-4_2 http://www.dmg.org/v4-2/pmml-4-2.xsd" xmlns:xsi="http://www.
 <Header copyright="dmg.org"/>
 <DataDictionary numberOfFields="2">
  <DataField dataType="double" name="input" optype="continuous"/>
  <DataField dataType="double" name="output2" optype="continuous"/>
 </DataDictionary>
 <TransformationDictionary>
  <DerivedField dataType="double" name="derived_input" optype="continuous">
   <Apply function="zeme1:func1">
    <!--(Comment generated by ADAPA) Error: Function 'zeme1:func1(double)' is not defined-->
    <FieldRef field="input"/>
   </Apply>
```

Note

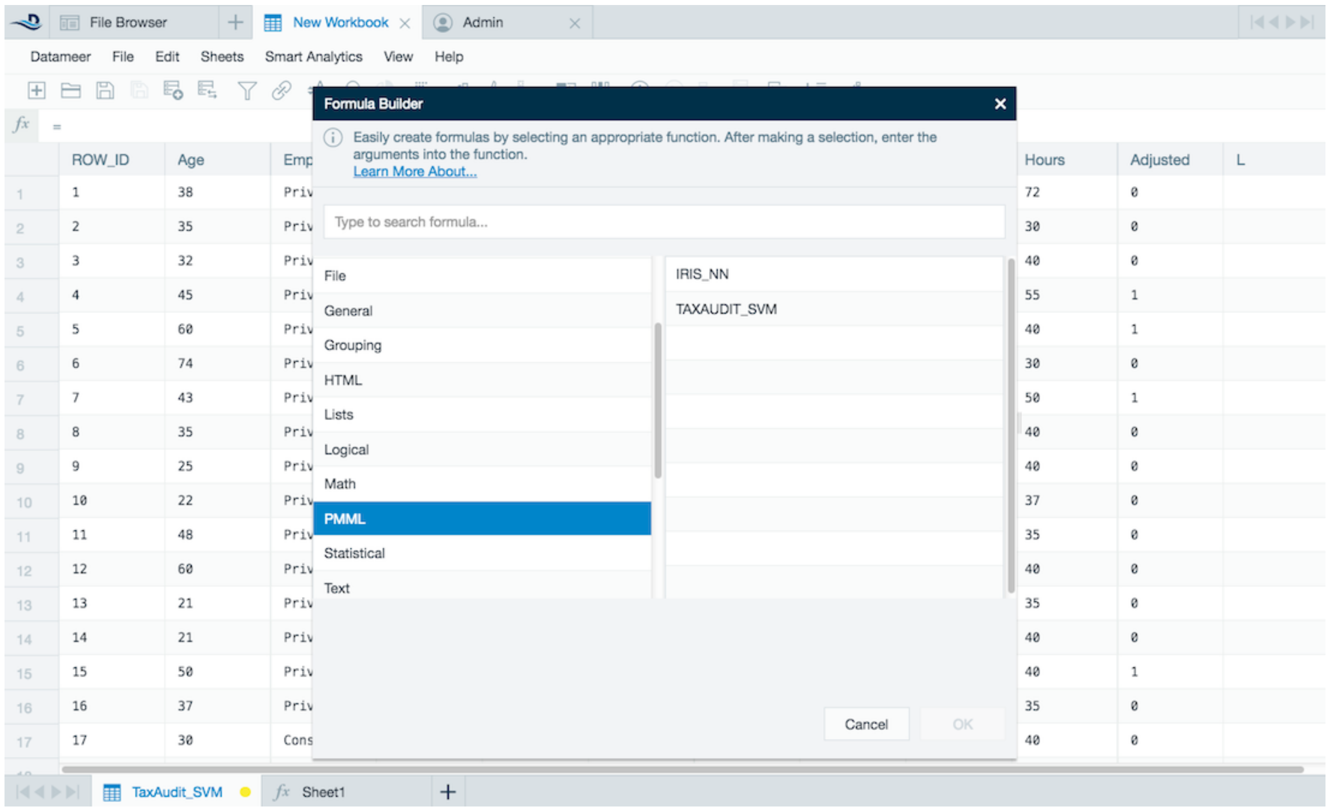
Note that even though a model may be deployed with no errors, ZEMENTIS may still find minor issues with the PMML code that are reported as warnings. Though these warnings typically do not impact the scoring process, they should be reviewed by the data science team.

Chapter 5. Scoring Models in Datameer®

This chapter describes the process of executing predictive models in Datameer®. Models can only be executed if they have already been uploaded and deployed in ZEMENTIS for Datameer®. This process is described in Chapter 4.

Once a predictive model is uploaded and deployed in ZEMENTIS for Datameer®, it becomes yet another function available through the "Formula Builder". To access the formula builder wizard, you will need to create a "Workbook" and import your data into it. The formula builder will be shown whenever you click on an empty cell. Given that we have already uploaded the PMML files for models Tax Audit SVM and Iris NN, these are shown as two distinct functions under the PMML function class as shown in Figure 5.1.

Figure 5.1. Datameer® Formula Builder



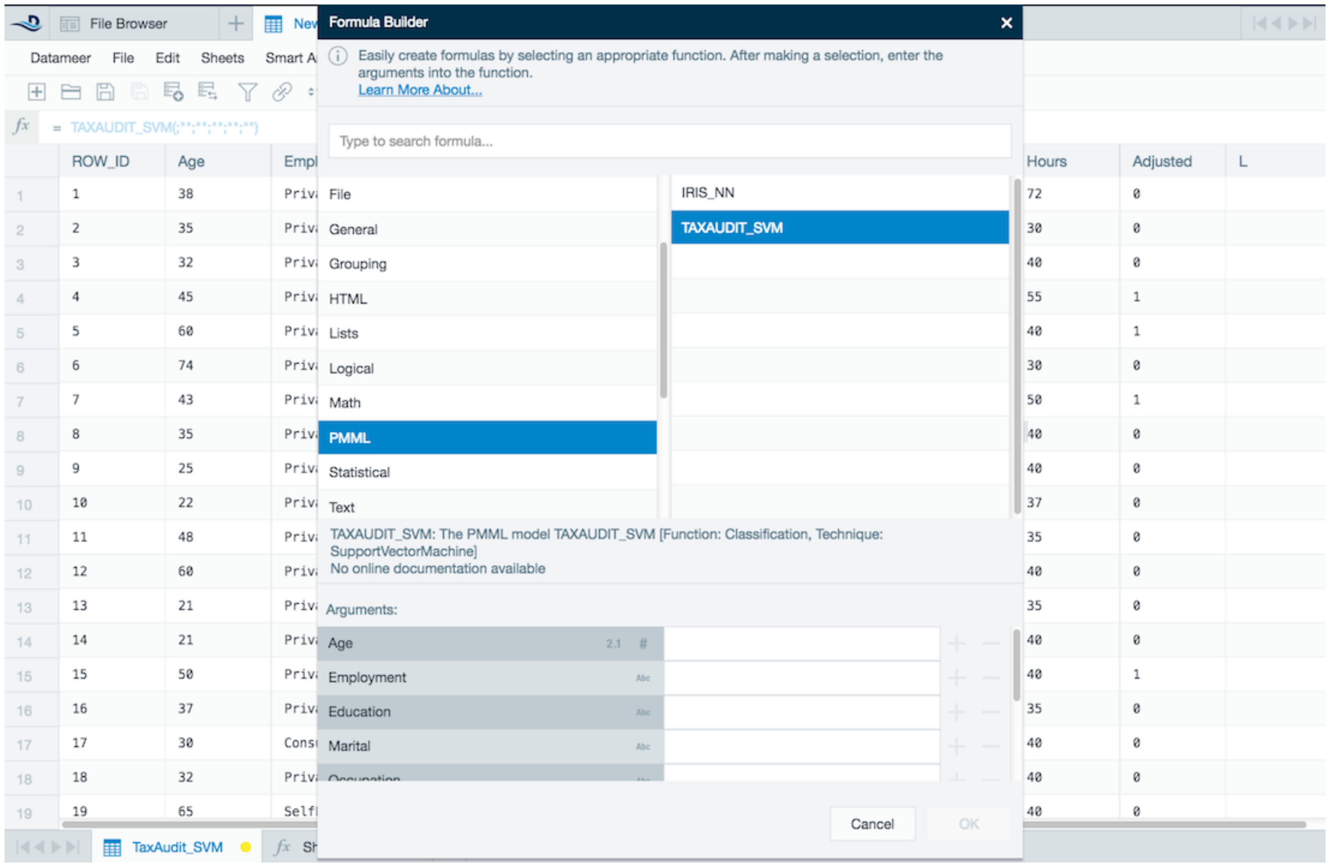
Easily create formulas by selecting an appropriate function. After making a selection, enter the arguments into the function.

	ROW_ID	Age	Emp	Hours	Adjusted	L
1	1	38	Priv	72	0	
2	2	35	Priv	30	0	
3	3	32	Priv	40	0	
4	4	45	Priv	55	1	
5	5	60	Priv	40	1	
6	6	74	Priv	30	0	
7	7	43	Priv	50	1	
8	8	35	Priv	40	0	
9	9	25	Priv	40	0	
10	10	22	Priv	37	0	
11	11	48	Priv	35	0	
12	12	60	Priv	40	0	
13	13	21	Priv	35	0	
14	14	21	Priv	40	0	
15	15	50	Priv	40	1	
16	16	37	Priv	35	0	
17	17	30	Cons	40	0	

5.1. Scoring Models with a single output

To create formula for Tax Audit SVM, simply select the TaxAudit_SVM function in the wizard. Once you do that, it will expand to show all the input arguments required for scoring the Tax Audit SVM model as depicted in Figure 5.2.

Figure 5.2. Scoring the Tax Audit SVM Model in ZEMENTIS for Datameer®



= TAXAUDIT_SVM("";"";"";"";"")

	ROW_ID	Age	Empl	Hours	Adjusted	L
1	1	38	Priv	72	0	
2	2	35	Priv	30	0	
3	3	32	Priv	40	0	
4	4	45	Priv	55	1	
5	5	60	Priv	40	1	
6	6	74	Priv	30	0	
7	7	43	Priv	50	1	
8	8	35	Priv	40	0	
9	9	25	Priv	40	0	
10	10	22	Priv	37	0	
11	11	48	Priv	35	0	
12	12	60	Priv	40	0	
13	13	21	Priv	35	0	
14	14	21	Priv	40	0	
15	15	50	Priv	40	1	
16	16	37	Priv	35	0	
17	17	30	Cons	40	0	
18	18	32	Priv	40	0	
19	19	65	Self	40	0	

Formula Builder

Easily create formulas by selecting an appropriate function. After making a selection, enter the arguments into the function.
Learn More About...

Type to search formula...

File	IRIS_NN
General	TAXAUDIT_SVM
Grouping	
HTML	
Lists	
Logical	
Math	
PMML	
Statistical	
Text	

TAXAUDIT_SVM: The PMML model TAXAUDIT_SVM [Function: Classification, Technique: SupportVectorMachine]
No online documentation available

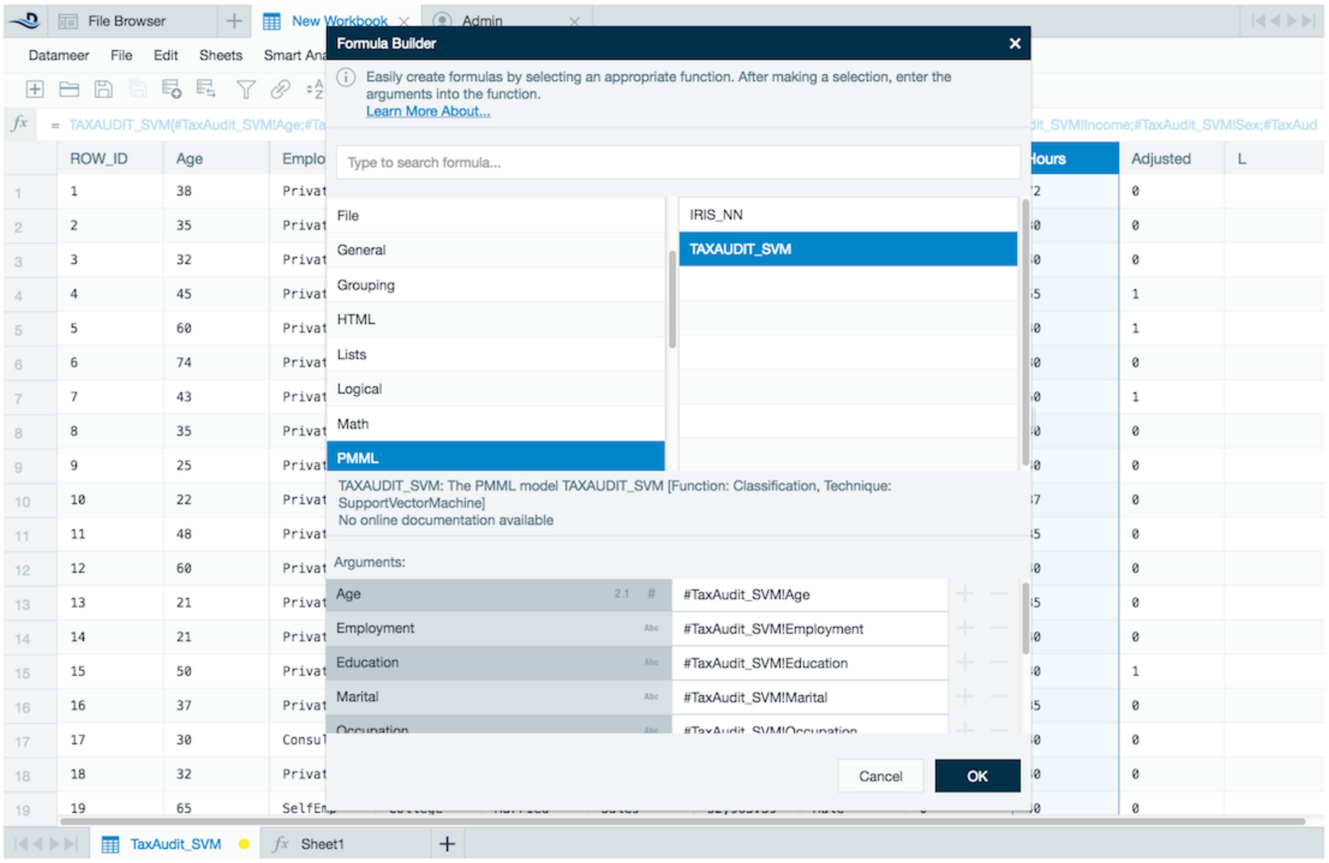
Arguments:

Age	2.1 #	
Employment	Abc	
Education	Abc	
Marital	Abc	
Occupation	Abc	

Cancel OK

To select the appropriate data columns for the input arguments in the wizard, simply click on any data cell for the column you would like to select. Datameer® automatically assumes the next selection you make is for the next argument in the formula wizard. After all entries have been mapped accordingly, as shown in Figure 5.3, simply click the "OK" button.

Figure 5.3. Tax Audit SVM Model inputs



= TAXAUDIT_SVM(#TaxAudit_SVM!Age;#Ta … dit_SVM!Income;#TaxAudit_SVM!Sex;#TaxAud

Easily create formulas by selecting an appropriate function. After making a selection, enter the arguments into the function.
Learn More About...

File	IRIS_NN
General	TAXAUDIT_SVM
Grouping	
HTML	
Lists	
Logical	
Math	
PMML	

TAXAUDIT_SVM: The PMML model TAXAUDIT_SVM [Function: Classification, Technique: SupportVectorMachine]
No online documentation available

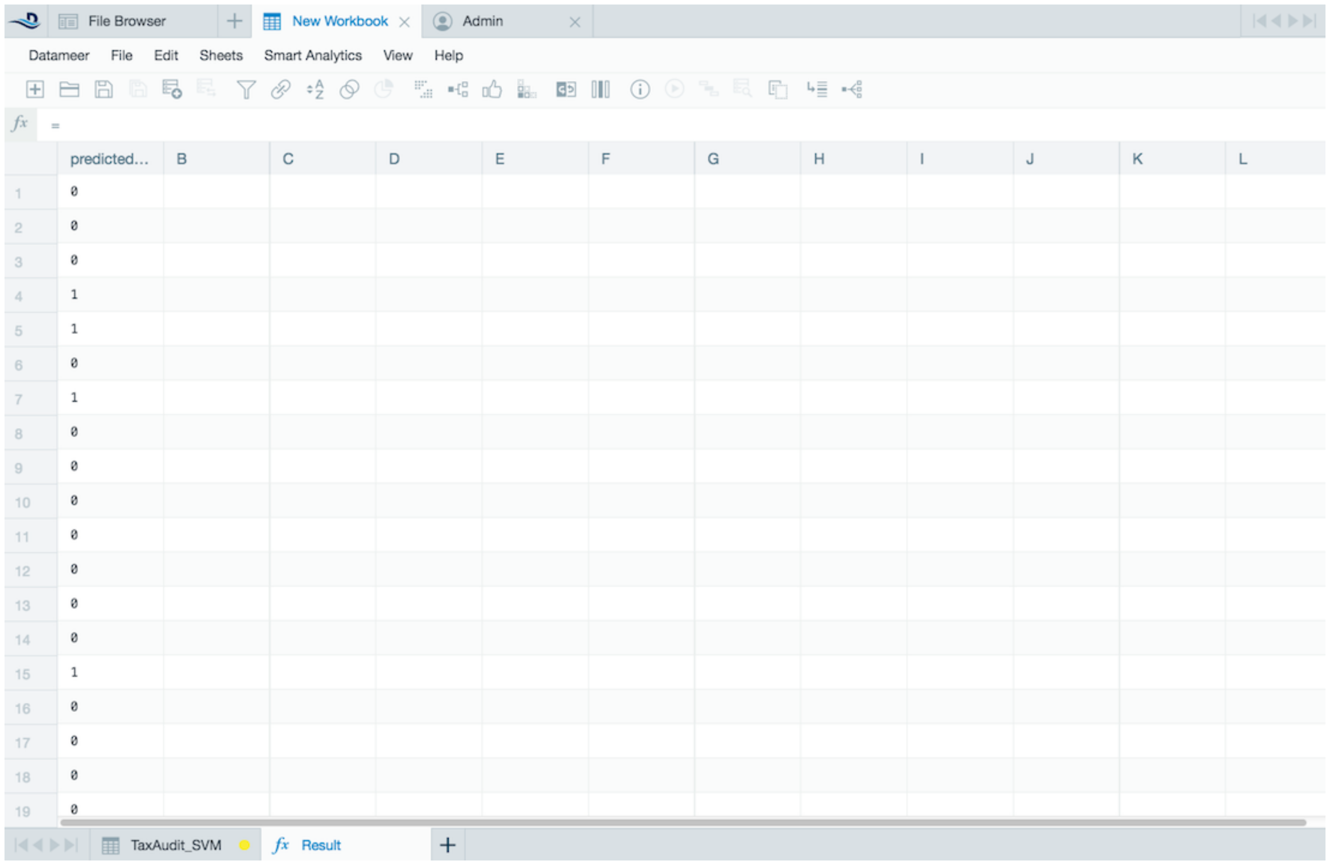
Arguments:

Age	#TaxAudit_SVM!Age
Employment	#TaxAudit_SVM!Employment
Education	#TaxAudit_SVM!Education
Marital	#TaxAudit_SVM!Marital
Occupation	#TaxAudit_SVM!Occupation

ROW_ID	Age	Emplo	Adjusted
1	38	Privat	0
2	35	Privat	0
3	32	Privat	0
4	45	Privat	1
5	60	Privat	1
6	74	Privat	0
7	43	Privat	1
8	35	Privat	0
9	25	Privat	0
10	22	Privat	0
11	48	Privat	0
12	60	Privat	0
13	21	Privat	0
14	21	Privat	0
15	50	Privat	1
16	37	Privat	0
17	30	Consul	0
18	32	Privat	0
19	65	SelfEm	0

The preview of predicted values will be shown in the column for which the formula wizard was invoked in the first place. In this case, column "A" in worksheet "Result" that was manually renamed to "Adjusted", contains predicted values. For the Tax Audit SVM model, they will contain "0" in case the taxpayer's claim does not need to be adjusted or "1" in case it does. Scoring results are shown in Figure 5.4. After saving our workbook, scoring can be fully processed simply by executing workbook job.

Figure 5.4. Tax Audit SVM Model output

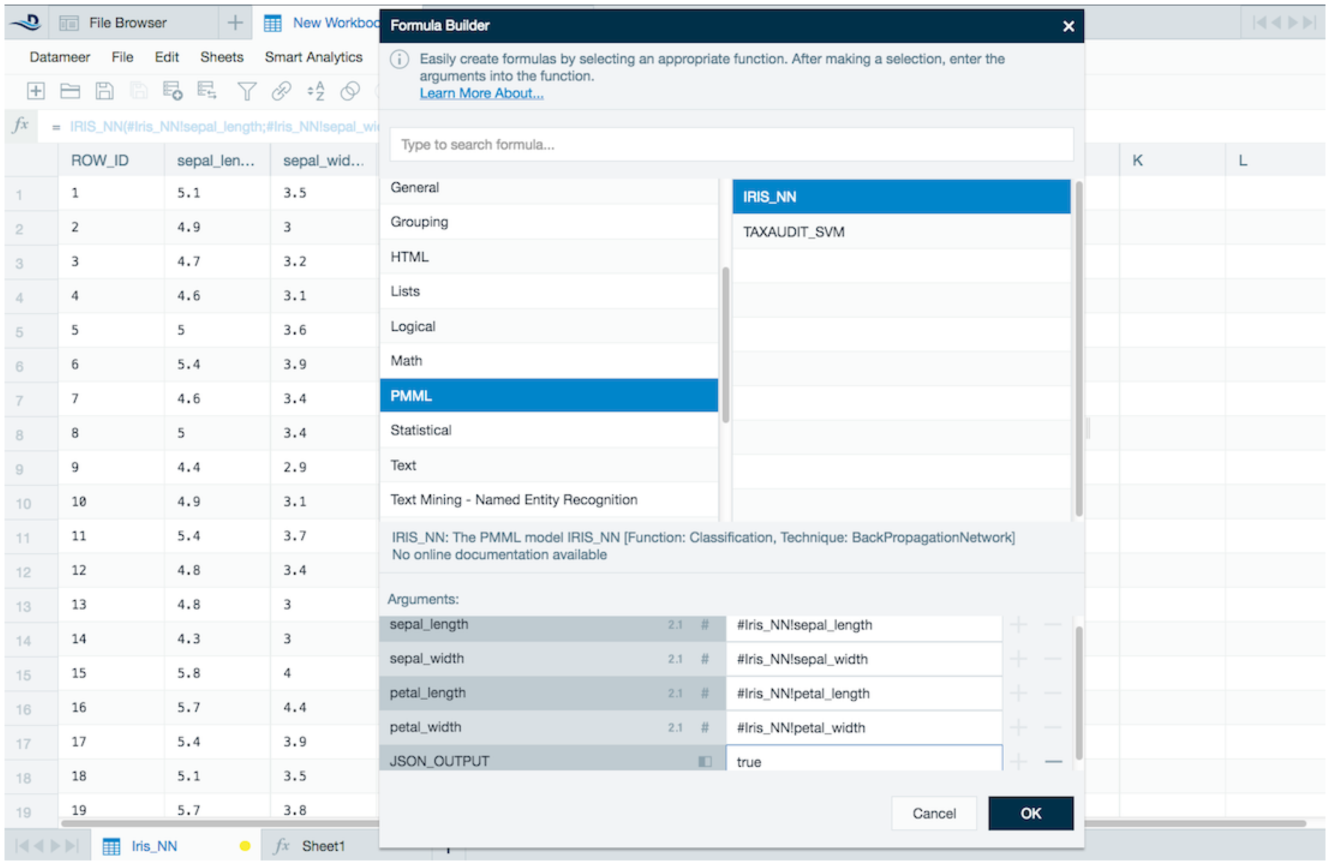


	predicted...	B	C	D	E	F	G	H	I	J	K	L
1	0											
2	0											
3	0											
4	1											
5	1											
6	0											
7	1											
8	0											
9	0											
10	0											
11	0											
12	0											
13	0											
14	0											
15	1											
16	0											
17	0											
18	0											
19	0											

5.2. Scoring Models with Multiple Outputs

You can use the same process for scoring the Iris NN model. Besides the predicted value "class", this model also outputs probabilities for each of the three possible classes: setosa, virginica, and versicolor, as described in Chapter 4. For ZEMENTIS for Datameer® to be able to return all four outputs, and not only the predicted value, make sure to add a "JSON_OUTPUT" field at the bottom of the list of arguments in the formula wizard. You can do that by clicking the "+" icon next to the last entry. Then set the "JSON_OUTPUT" to "true" as shown in Figure 5.5.

Figure 5.5. Scoring the Iris NN Model in ZEMENTIS for Datameer®



= IRIS_NN(#Iris_NN!sepal_length;#Iris_NN!sepal_wi

	ROW_ID	sepal_len...	sepal_wid...
1	1	5.1	3.5
2	2	4.9	3
3	3	4.7	3.2
4	4	4.6	3.1
5	5	5	3.6
6	6	5.4	3.9
7	7	4.6	3.4
8	8	5	3.4
9	9	4.4	2.9
10	10	4.9	3.1
11	11	5.4	3.7
12	12	4.8	3.4
13	13	4.8	3
14	14	4.3	3
15	15	5.8	4
16	16	5.7	4.4
17	17	5.4	3.9
18	18	5.1	3.5
19	19	5.7	3.8

Formula Builder

Easily create formulas by selecting an appropriate function. After making a selection, enter the arguments into the function.
Learn More About...

Type to search formula...

General	IRIS_NN
Grouping	TAXAUDIT_SVM
HTML	
Lists	
Logical	
Math	
PMML	
Statistical	
Text	
Text Mining - Named Entity Recognition	

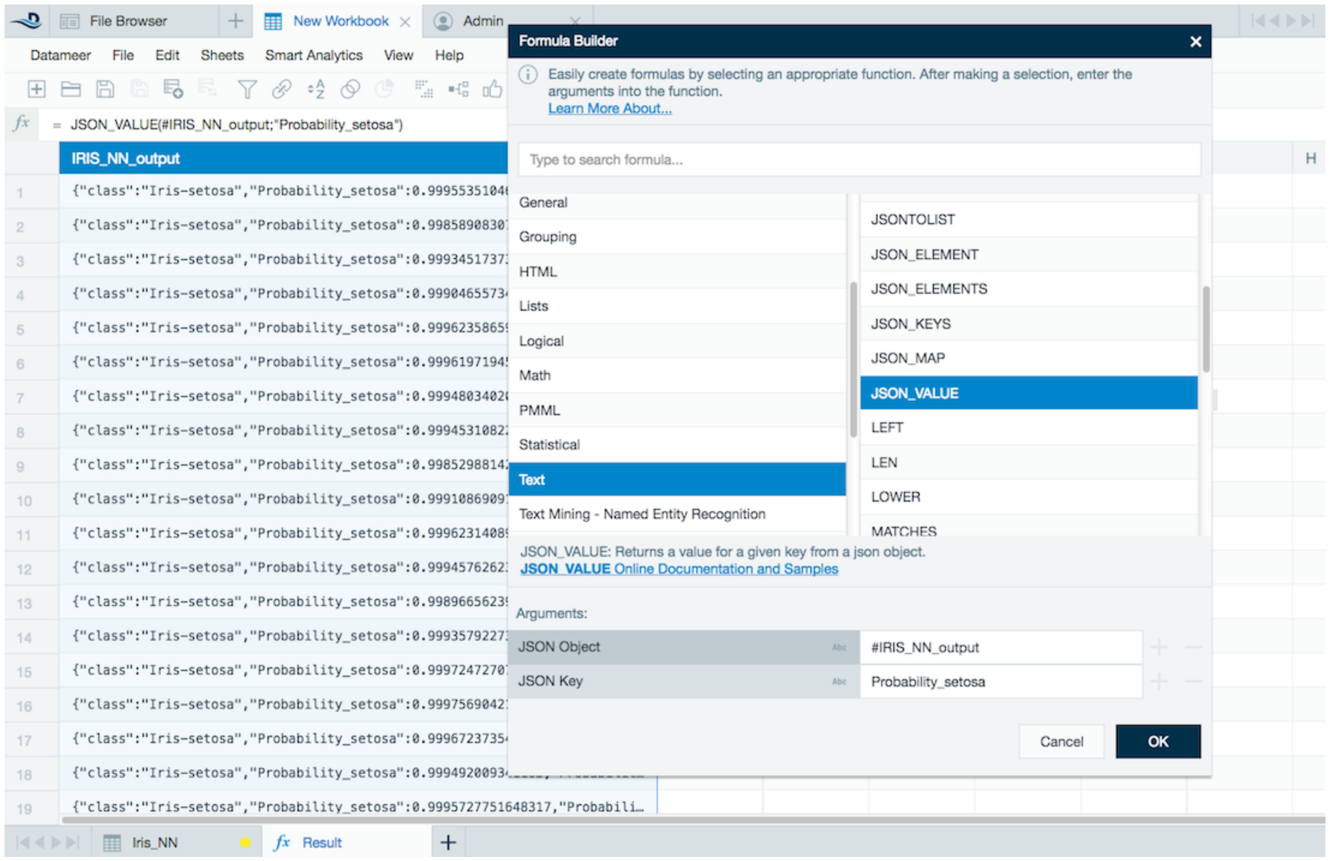
IRIS_NN: The PMML model IRIS_NN [Function: Classification, Technique: BackPropagationNetwork]
No online documentation available

Arguments:

sepal_length	#Iris_NN!sepal_length
sepal_width	#Iris_NN!sepal_width
petal_length	#Iris_NN!petal_length
petal_width	#Iris_NN!petal_width
JSON_OUTPUT	true

Once you click on the "OK" button, ZEMENTIS for Datameer® will show the preview of the output from the Iris NN model. In our example, the outputs will be written to column "A" of "Result" worksheet as shown in Figure 5.6. In order to separate out a particular output, invoke the "Formula Builder" once again and select function "JSON_VALUE" under "text". Once you do that, the formula wizard will expand to show two arguments: "JSON Array" and "JSON Key". The first, "JSON Array" needs to be mapped to the "JSON_OUTPUT" which is available in column "A". The second argument, "JSON Key" needs to be populated with the output of interest which, in our example, is "Probability_setosa".

Figure 5.6. Multiple outputs with JSON_OUTPUT



= JSON_VALUE(#IRIS_NN_output;"Probability_setosa")

	IRIS_NN_output
1	{"class":"Iris-setosa","Probability_setosa":0.9995535104
2	{"class":"Iris-setosa","Probability_setosa":0.9985890830
3	{"class":"Iris-setosa","Probability_setosa":0.9993451737
4	{"class":"Iris-setosa","Probability_setosa":0.9990465573
5	{"class":"Iris-setosa","Probability_setosa":0.9996235865
6	{"class":"Iris-setosa","Probability_setosa":0.9996197194
7	{"class":"Iris-setosa","Probability_setosa":0.9994803402
8	{"class":"Iris-setosa","Probability_setosa":0.9994531082
9	{"class":"Iris-setosa","Probability_setosa":0.9985298814
10	{"class":"Iris-setosa","Probability_setosa":0.9991086909
11	{"class":"Iris-setosa","Probability_setosa":0.9996231408
12	{"class":"Iris-setosa","Probability_setosa":0.9994576262
13	{"class":"Iris-setosa","Probability_setosa":0.9989665623
14	{"class":"Iris-setosa","Probability_setosa":0.9993579227
15	{"class":"Iris-setosa","Probability_setosa":0.9997247270
16	{"class":"Iris-setosa","Probability_setosa":0.9997569042
17	{"class":"Iris-setosa","Probability_setosa":0.9996723735
18	{"class":"Iris-setosa","Probability_setosa":0.9994920093
19	{"class":"Iris-setosa","Probability_setosa":0.9995727751648317,"Probabili…

Formula Builder

Easily create formulas by selecting an appropriate function. After making a selection, enter the arguments into the function.
Learn More About...

Type to search formula...

General	JSONTOLIST
Grouping	JSON_ELEMENT
HTML	JSON_ELEMENTS
Lists	JSON_KEYS
Logical	JSON_MAP
Math	JSON_VALUE
PMML	LEFT
Statistical	LEN
Text	LOWER
Text Mining - Named Entity Recognition	MATCHES

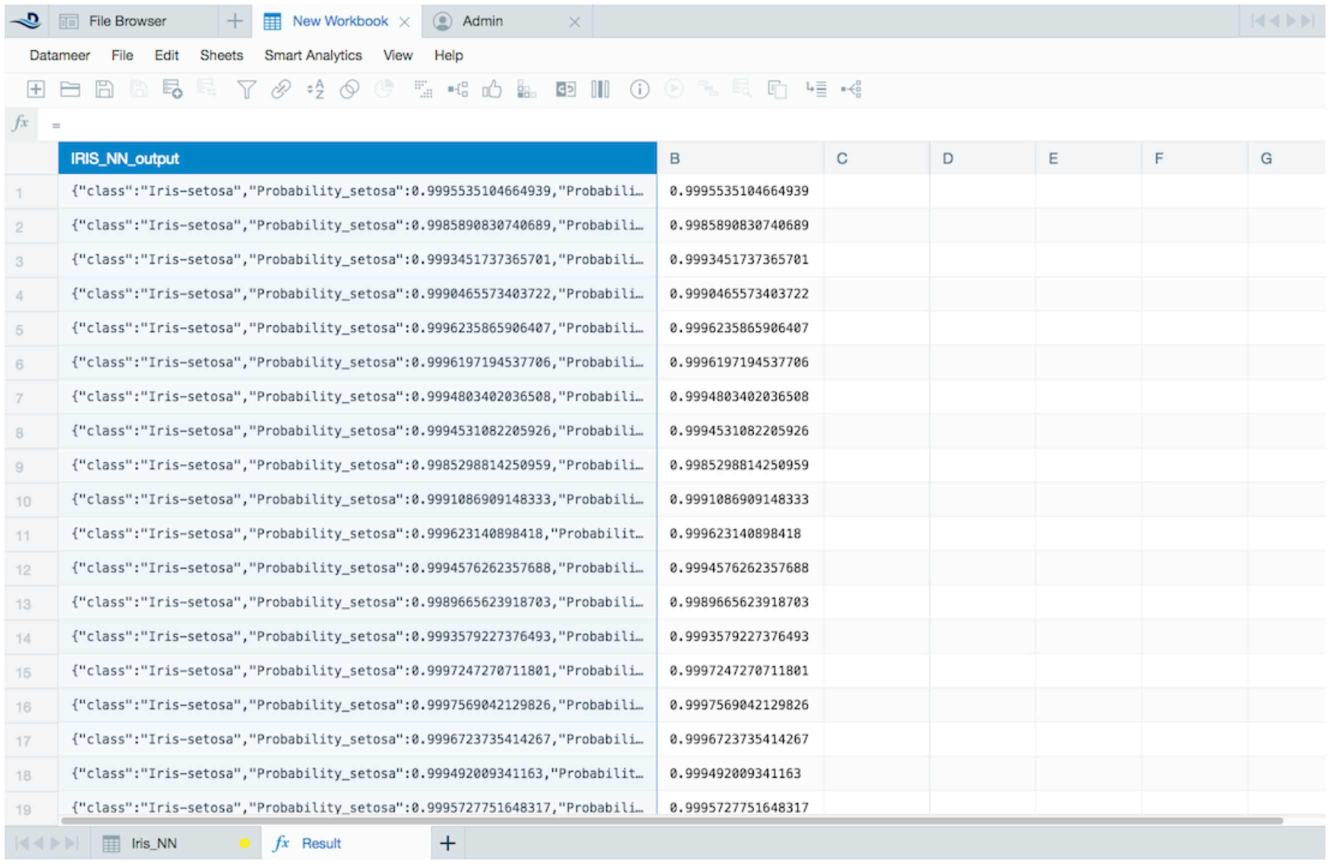
JSON_VALUE: Returns a value for a given key from a json object.
JSON_VALUE Online Documentation and Samples

Arguments:

JSON Object	#IRIS_NN_output
JSON Key	Probability_setosa

By clicking on the "OK" button, the probability for class "Iris_setosa" is then extracted from column "A" and copied to column "B", later manually renamed to "Probability_setosa", as shown in Figure 5.7. Note that the same process could be used to extract the predicted class as well as any of the other class probabilities. After saving our workbook, scoring can be fully processed simply by executing workbook job.

Figure 5.7. Selecting a specific output with JSON_VALUE



	IRIS_NN_output	B	C	D	E	F	G
1	{"class":"Iris-setosa","Probability_setosa":0.9995535104664939,"Probabili…	0.9995535104664939					
2	{"class":"Iris-setosa","Probability_setosa":0.9985890830740689,"Probabili…	0.9985890830740689					
3	{"class":"Iris-setosa","Probability_setosa":0.9993451737365701,"Probabili…	0.9993451737365701					
4	{"class":"Iris-setosa","Probability_setosa":0.9990465573403722,"Probabili…	0.9990465573403722					
5	{"class":"Iris-setosa","Probability_setosa":0.9996235865906407,"Probabili…	0.9996235865906407					
6	{"class":"Iris-setosa","Probability_setosa":0.9996197194537706,"Probabili…	0.9996197194537706					
7	{"class":"Iris-setosa","Probability_setosa":0.9994803402036508,"Probabili…	0.9994803402036508					
8	{"class":"Iris-setosa","Probability_setosa":0.9994531082205926,"Probabili…	0.9994531082205926					
9	{"class":"Iris-setosa","Probability_setosa":0.9985298814250959,"Probabili…	0.9985298814250959					
10	{"class":"Iris-setosa","Probability_setosa":0.9991086909148333,"Probabili…	0.9991086909148333					
11	{"class":"Iris-setosa","Probability_setosa":0.999623140898418,"Probabilit…	0.999623140898418					
12	{"class":"Iris-setosa","Probability_setosa":0.9994576262357688,"Probabili…	0.9994576262357688					
13	{"class":"Iris-setosa","Probability_setosa":0.9989665623918703,"Probabili…	0.9989665623918703					
14	{"class":"Iris-setosa","Probability_setosa":0.9993579227376493,"Probabili…	0.9993579227376493					
15	{"class":"Iris-setosa","Probability_setosa":0.9997247270711801,"Probabili…	0.9997247270711801					
16	{"class":"Iris-setosa","Probability_setosa":0.9997569042129826,"Probabili…	0.9997569042129826					
17	{"class":"Iris-setosa","Probability_setosa":0.9996723735414267,"Probabili…	0.9996723735414267					
18	{"class":"Iris-setosa","Probability_setosa":0.999492009341163,"Probabilit…	0.999492009341163					
19	{"class":"Iris-setosa","Probability_setosa":0.9995727751648317,"Probabili…	0.9995727751648317					

Multiple outputs and output types will vary depending on the model type (e.g. SVM, NN, Regression, Decision Tree, Clustering, ...) and function the model implements (e.g. classification, regression, ...). Nonetheless, multiple outputs in PMML are represented in the element "Output". This element determines the types of outputs a model should provide after scoring. The code shown below depicts the "Output" element for the Iris NN model. Note that, as expected, it specifies four outputs, the predicted value "class" together with the three class probabilities.

```
<Output>
 <OutputField name="class" dataType="string" optype="categorical"
  feature="predictedValue"/>
    <OutputField name="Probability_setosa" value="Iris-setosa"
     dataType="double" optype="continuous" feature="probability"/>
    <OutputField name="Probability_versicolor" value="Iris-versicolor"
     dataType="double" optype="continuous" feature="probability"/>
    <OutputField name="Probability_virginica" value="Iris-virginica"
     dataType="double" optype="continuous" feature="probability"/>
</Output>
```

For a list of forums covering PMML and Zementis products or for support, please make sure to visit the Zementis support site: https://support.zementis.com