NeuralWorks Predict®

The Complete Solution for
Neural Data Modeling

Getting Started Guide for Windows

NEURALWARE®

This document is provided for informational purposes only, and the content of this document is subject to change without notice. NeuralWare assumes no liability or responsibility for any errors or inaccuracies that may appear in this document.

NeuralWare welcomes your comments about this document! Please contact us at:

NeuralWare
230 East Main Street
Suite 200
Carnegie PA 15106

Phone: `+1 412.278.6280`
Fax: `+1 412.278.6289`

Web: `www.neuralware.com`

Email: `sales@neuralware.com(Product and Purchasing Information)`
`help@neuralware.com (Technical Support)`

Product Release 3.1 (June 2005).

Printed in the United States of America.

Contents

Welcome

Welcome to NeuralWorks Predict®! (Often we will simply use "*Predict*" when we refer to the product.) Whether you are an end user, an application developer, or a system integrator, we think you'll find NeuralWorks Predict easy to learn and use.

In business, industry, government, and academia, *Predict* has been used to:

- Identify financial trends that influence market timing, stock selection, and asset allocation.
- Identify and evaluate risks, both when credit is applied for and after credit is granted.
- Analyze complex medical conditions to suggest the most effective treatment.
- Select demographic groups with specific characteristics for targeted marketing.
- Identify, extract, and classify biometric signatures, recognize hand-written characters and spoken language, and to process and interpret virtually any kind of image data.
- Develop models for controlling and optimizing manufacturing processes and supply chains.

Very quickly, you too will be harnessing the power of *Predict* to analyze data and solve problems in your own application environment.

Predict can be used in a wide range of applications because it is based on an analytic approach known as ***empirical modeling***. Empirical models are collections or combinations of mathematical functions that are iteratively optimized to represent the complex relationships found in data collected from the real world. The contrasting analytic approach is ***first principles modeling***. In first principles models, the relationships in data are derived from theoretical considerations of the underlying physical, chemical, biological, or other process (and of course, such theories may or may not be complete or accurate).

Predict relies on neural networks, which have been shown to be universal function approximators, to discover and structure the relationships between input variables (data) and some output variable or variables of interest. However, one of the many features that distinguishes *Predict* from other empirical modeling and neural computing tools is that it automates much of the painstaking and time-consuming process of selecting and transforming the data needed to build a neural network.

Even someone unfamiliar with neural networks can quickly become productive, since key training and neural network architecture parameters are specified simply by answering a few questions about the data and the type of application (e.g., prediction or classification). At the same time, *Predict* provides complete access to all model parameters. As you gain more experience with *Predict*, you can access all model related parameters and manually fine-tune them to ensure the most robust model for your particular application is found.

How to use this Guide

This Guide introduces some of the key features and capabilities of NeuralWorks Predict® and shows how to use *Predict* to build empirical models. We urge you to read this entire Guide so you can become proficient with *Predict* as quickly as possible.

The information begins with the initial run of the software, referencing Appendix A should you need to install the product, and referencing Appendix B for details of activating your *Predict* software license. Then, after a brief introduction to neural network technology, a Tutorial illustrates how to build a model for a prediction application. In the Tutorial, the *Predict* Model Building Wizard will lead you through the model building process. Finally, the Guide indicates how to use the *NeuralWorks Predict User Guide* and *Predict Help* to obtain information about the full range of *Predict* capabilities. In particular, a more detailed explanation of neural networks can be found in the *User Guide* chapter “Building a Neural Network Model”.

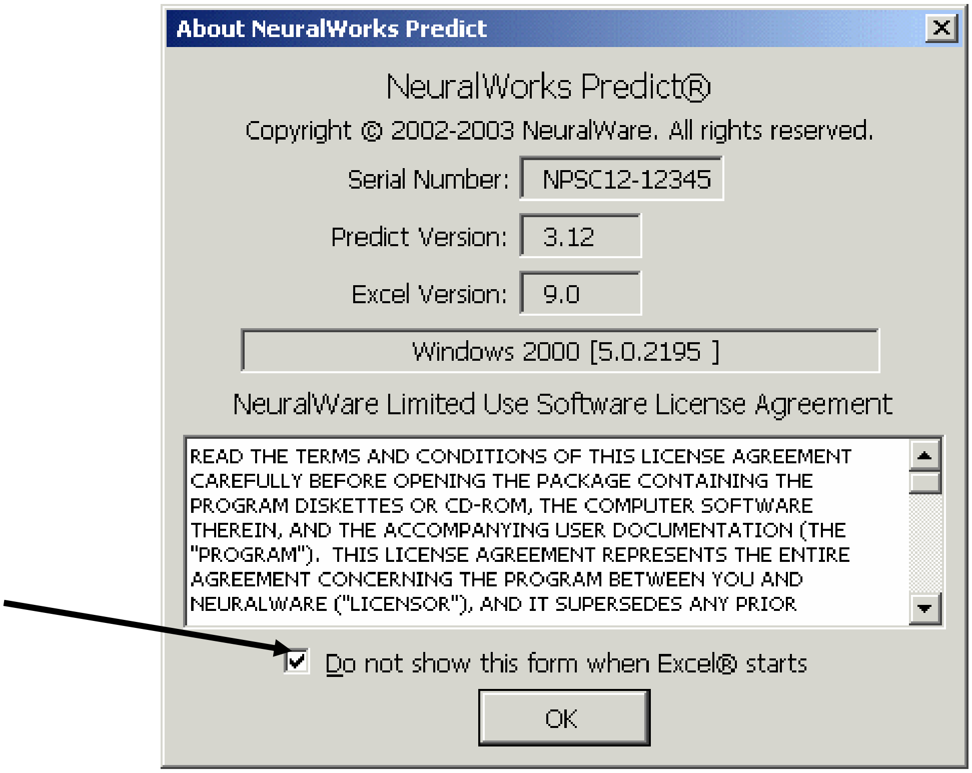
Running NeuralWorks Predict for the First Time

This guide introduces you to *Predict* when it is running within Microsoft Excel. Do ***not*** start *Predict* from the Windows **Start** menu – if you do, a secondary interface called the 'Command Line Interface' will start. The Command Line Interface is not available in demonstration versions of *Predict*.

If you have not yet installed *Predict*, please see Appendix A.

Start Predict via Microsoft Excel

To run *Predict* with Microsoft Excel, simply start Microsoft Excel. If *Predict* was properly installed, it is automatically started when Excel starts. An **About NeuralWorks Predict** dialog box, similar to the one shown below, appears. The Serial Number and other information specific to your computer will be different in the dialog box that appears on your computer.



About NeuralWorks Predict

NeuralWorks Predict®

Copyright © 2002-2003 NeuralWare. All rights reserved.

Serial Number: NPSC12-12345

Predict Version: 3.12

Excel Version: 9.0

Windows 2000 [5.0.2195]

NeuralWare Limited Use Software License Agreement

READ THE TERMS AND CONDITIONS OF THIS LICENSE AGREEMENT CAREFULLY BEFORE OPENING THE PACKAGE CONTAINING THE PROGRAM DISKETTES OR CD-ROM, THE COMPUTER SOFTWARE THEREIN, AND THE ACCOMPANYING USER DOCUMENTATION (THE "PROGRAM"). THIS LICENSE AGREEMENT REPRESENTS THE ENTIRE AGREEMENT CONCERNING THE PROGRAM BETWEEN YOU AND NEURALWARE ("LICENSOR"), AND IT SUPERSEDES ANY PRIOR

☑ Do not show this form when Excel® starts

To view the entire **NeuralWare Limited Use Software License Agreement**, click in the license text area of the dialog box and use either the scroll bar or the **Page-Up** and **Page-Down** keys to move through the license.

So this dialog does not appear the next time we start Excel, click in the check-box as shown above (clicking adds or removes a check mark, we want to add it). It is better to run with this dialog disabled.

Click **OK** to close the dialog box.

⇨ ***Note:*** If the **About NeuralWorks Predict** dialog box did not appear, and the *Predict* menu is *not* visible in the Excel menu bar, it means that when you installed *Predict* the Setup program could not locate the Excel `XLStart` directory. You will need to manually link *Predict* with Excel. Please refer to the *Installation and Software License Activation* chapter in the *NeuralWorks Predict User Guide* or *Predict Help* for more information and instructions for linking *Predict* and Excel.

Entering a License Key

If a dialog box entitled **NeuralWorks Predict License Management** appears, then you need to obtain a license key from NeuralWare. If this dialog box appears then please see Appendix B for details about how to obtain a license key. Note that demonstration versions of *Predict* do not need a license key.

Even if you need a license key and do not yet have one, you can continue running this exercise by clicking the **Run Predict Demo** button. *Predict* will then continue as if you were using a demonstration version.

Running Predict in Demonstration Mode

The demonstration version of *Predict* limits the size of models that can be built to 32 fields and 512 data records, and models cannot be saved. Some of *Predict*'s advanced features such as generating FlashCode are also not available. However, if *Predict* is running in demonstration mode you can perform all the actions described in the Tutorial chapter in this guide except saving the network and where otherwise noted.

The Predict Command Line Interface

In Microsoft Windows *Predict* can be run as a stand-alone program that has a Command Line Interface. Use of the Command Line Interface is beyond the scope of this *Guide* – please refer to the *Command Line Interface Reference* chapter in the *NeuralWorks Predict User Guide* for information about the Command Line Interface. The Command Line Interface is not available in the demonstration version of *Predict*.

What To Do Next

At this point the Microsoft Excel interface for *Predict* is running, and you should see a normal Microsoft Excel spreadsheet with a blank spreadsheet. Notice the new *Predict* menu in Excel's menu bar, next to the Excel Help menu.

You are now ready to open data files and build models. Following the explanation of the next few pages, we'll do just that.

Neural Network Basics

Artificial neural networks were inspired in large part by research into the function of neurons in the human brain. Artificial neural networks process information in a way that resembles the way the brain works. Like the brain, neural networks "learn" from experience during a process called ***training***.

You can use neural networks when unknown relationships exist in historical data. A ***historical dataset*** consists of recorded input values and their corresponding output values. Neural networks can detect patterns in data, generalize about relationships in data, and generate an output value when given a new set of input values from the problem domain. Analysts who lack extensive domain knowledge can use neural networks to solve problems that prove too complex for more conventional analytical techniques.

A neural network is a non-linear estimation technique. The neural network itself is a mathematical function (given a set of input values, it produces an output value). The structure of the network, as often depicted by pictures of circles and lines, is just a way to represent the mathematical function graphically instead of using a conventional (and usually very lengthy) equation. The neural network is the core of the ***model*** that *Predict* builds. The model refers not only to the neural network but also to all of the pre- and post-processing steps required to produce an output value given a set of input values.

How a Neural Network Learns

The human brain is a very complex system of interconnected neurons. Similarly, a neural network is an interconnected system of artificial "neurons." In neural network terminology, neurons are called ***Processing Elements*** or *nodes*. Like a neuron in the brain, each Processing Element (PE) can accept input data, process the data, and pass it to the next PE. A PE processes data using one of several types of mathematical functions. In effect an entire neural network represents a composite of the functions represented by all PEs.

⇨ The key to building a robust neural network is to collect many examples (or records) of input values and corresponding output values over time. The neural network uses this historical data to determine (learn) a mathematical relationship between the input data and the output data.

Network Architecture

As with a neuron in the brain, a single PE has limited ability. However, when connected in a system (a network), the neurons or PEs become a powerful analytic tool.

In a neural network, PEs can be interconnected in various ways. Typically, PEs are structured into layers and the output values of PEs in one layer serve as input values for PEs in the next layer. Each connection has a ***weight*** associated with it. In most cases, a Processing Element calculates a weighted sum of incoming values (the sum the outputs of the PEs connected from the next lower level multiplied by their connection weights). This sum is called the ***activation value***. The activation value is then passed to the PE's non-linear ***transfer function*** to produce an output for that PE.

This combination of PEs, connections, weights, and transfer functions form the ***network architecture***. This architecture then represents a complex mathematical formula that has been derived from historical data.

Training a Neural Network

Neural networks learn from "experience" (exposure to information). From repeated exposure to historical data, a neural network learns to strengthen connection weights from PEs that have a greater tendency to accurately predict the desired output. The strength of each connection (the magnitude of the connection weight) increases or decreases based on its influence in producing the output associated with each input data record in the historical dataset. Connection weights are adjusted during ***training***.

Training is the process of repeatedly (iteratively) exposing a neural network to examples of historical data. Each example contains input variables, and the associated desired output. The desired output is the value that the neural network should predict, given the associated input values in the historical data. In neural network terminology, the desired output is called the ***target value*** [1]. During training, network weights are adjusted according to a ***learning rule*** – an algorithm that specifies how the error between ***predicted outputs*** and ***target values*** should be used to modify network weights.

Before training begins, a simple network with the necessary number of input and output PEs is created. All connection weights are initialized to small random values. Data examples from the ***training set*** are passed to the network and the network produces an output value. This value is compared to the target value. The weights are adjusted in order to decrease the error between the network output and the target value. In addition, more processing elements are added to the network if doing so helps decrease the error of the network.

Training continues until the neural network produces output values that match the target values within a specified accuracy level, or until some other stopping criterion is satisfied.

Testing and Validating a Neural Network

Just as you might test a person’s skill in a controlled environment, you *test* a neural network using historical data it has not seen. Test results are good if the predicted values are close to the target values.

One difficulty for neural networks (and other non-linear estimation techniques) is the possibility that the network will ***over-fit*** the training data. This means that the network might closely predict the target values on the training data but produce inferior results for new data. Training for too long (too many passes through the training set) can cause the function to become very complex in order to produce the target values at the expense of ***generalizing*** well on unseen data. However, if a network is not trained long enough, it doesn’t fully learn trends and relationships in the data.

One way of knowing when to stop training is to automatically and periodically test the performance on a ***test set*** during training. When the performance starts to degrade on the test set, it's time to stop training - the neural network has started to learn relationships which are specific to only the training set! At the end of training, the neural network can be further tested on an additional independent test set referred to as a ***validation set***. (For a detailed explanation of the necessity of an additional independent validation set, please see the chapter titled “Train, Test, and Validation Sets” in the *NeuralWorks Predict User Guide*.)

⇨ If the test and validation results are not good, you can collect more data, add input variables, and/or change the neural network’s architecture or parameters. Then, you need to train and test the new network to see if performance has improved.

Using a Neural Network

After the neural network has been trained, it can be used to make predictions given input data for which the desired output value is unknown.

Building a Network with NeuralWorks Predict

Building a good neural network with *Predict* involves:

1. Collecting and pre-processing data.
2. Constructing and training the network.
3. Testing and validating the network.

[1] There may be more than one target value for each example. In this case, the neural network is constructed to predict more than one output value for each set of input values. This is called multivariate regression. A multiple output neural network model can be built in *Predict* in the same manner as a single output model.

You take care of part of Step 1 — that is, collecting data and basic pre-processing to organize the data. *Predict* automatically performs more sophisticated pre-processing in Step 1 — plus steps 2 and 3.

What You Do

Collecting and Pre-processing Data

This step involves defining the problem you want to solve, collecting data, and analyzing and transforming the data so that *Predict* can build the best possible network.

- **Problem Definition.** This means that you define exactly what the input and output variable(s) should be.
- **Data Collection.** It is best to have as many data examples as possible. You should attempt to collect a nearly equal number of examples for each expected possible target value or range of values.
- **Data Analysis and Transformation.** Perform any preliminary data manipulation or problem restructuring that is appropriate for the particular problem you are trying to solve.

How NeuralWorks Predict Helps You

Based on data and model information you provide, *Predict* automatically builds a network. However, *Predict* also provides full access to the network architecture so that as you become more experienced you can fine-tune the core network and model parameters that control *Predict's* automated analysis processes.

What Predict Does

Building a network consists of dividing available input data into train, test and validation sets, augmenting the preliminary data analysis and transformation you perform, selecting the important variables from the input examples, and then training the network. A detailed discussion of these steps can be found in the *NeuralWorks Predict User Guide*.

- **Train, Test, and Validation Set Selection.** *Predict* automatically selects train, test, and validation sets from the historical data you supply.
- **Data Analysis and Transformation.** *Predict* automatically analyzes data and converts it into a form suitable for building an effective network. This may include converting string inputs to categorical inputs, scaling and transforming data, and removing outliers.
- **Variable Selection.** *Predict* automatically selects input variables (or transformations of input variables) that are the most influential in predicting target values. A genetic algorithm is used to search through the space of combinations of input variables.
- **Network Architecture.** *Predict* automatically adds PEs to the network, and sets up other network parameters.
- **Training.** Based on information you supply, *Predict* chooses one of two available learning rules in order to achieve the best results.

Testing and Validating the Network

After *Predict* has generated and trained the network, you can verify its performance using the train, test, and validation sets of data. *Predict* compares the predicted output values for these sets of data with the target values and, when using the Excel interface, writes the results to an Excel spreadsheet. When all steps are completed and network performance with both test and validation data is good, the neural network is ready to deploy and process new data for which the desired output value is unknown.

The Microsoft Excel Interface

To effectively use *Predict*, it is helpful both to know how to take advantage of the Microsoft Excel features that *Predict* relies on, and to know about problems in Microsoft Excel that you want to avoid. In general, you should know:

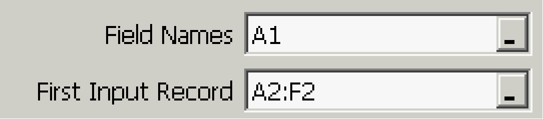
- how to open and save files;
- about ranges and how to select them;
- how to copy blocks of cells;
- how to paste functions; and
- how to use Excel charting functions.

Selecting Ranges

Since all the data that *Predict* uses to build a model is obtained from an Excel Workbook, it is important to know how to best enter Excel ranges when necessary.

> **Note:** In addition to the methods described below, you of course can always enter ranges by typing directly into the range text box.

All dialog boxes that require data from Excel use a special range text box that is similar to the text boxes in the illustration below.



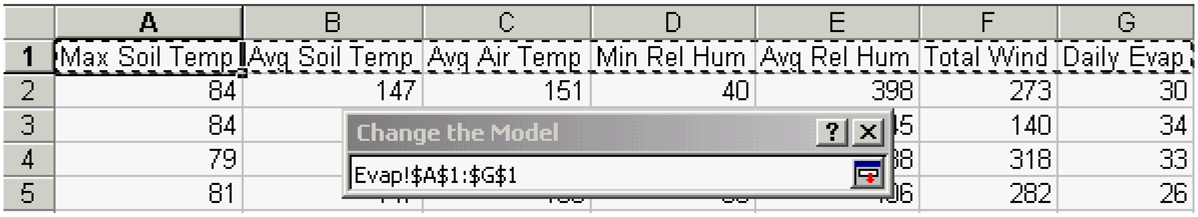
Field Names: A1

First Input Record: A2:F2

Note that the special range text boxes have a small Control box, containing an underscore, on the right side. Any text box that contains the small Control behaves in a special way when you click first in the text box, and then in a cell in the Worksheet. Unlike normal text boxes, if you click in a range text box and then click in the Worksheet, you can scroll the Worksheet to make the area you are interested in easier to view.

Also, if you click in a range text box, and then click ***and drag*** in the Worksheet, the dialog box that contains the range text box will be minimized so that you can view much more of the Worksheet. When you release the primary mouse button to end dragging, the dialog box returns to its normal size.

For example, after first clicking in the Field Names range text box illustrated above, clicking and dragging (starting in cell A1) in the Worksheet results in the Worksheet appearance becoming similar to the following:

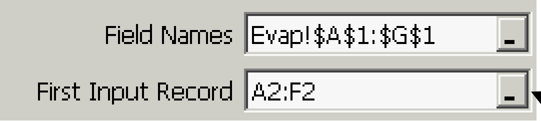


	A	B	C	D	E	F	G
1	Max Soil Temp	Avg Soil Temp	Avg Air Temp	Min Rel Hum	Avg Rel Hum	Total Wind	Daily Evap
2	84	147	151	40	398	273	30
3	84					140	34
4	79					318	33
5	81					282	26

Change the Model

Evap!$A$1:$G$1

Releasing the primary mouse button causes the dialog box to be restored, and the selected range to be placed in the range text box.

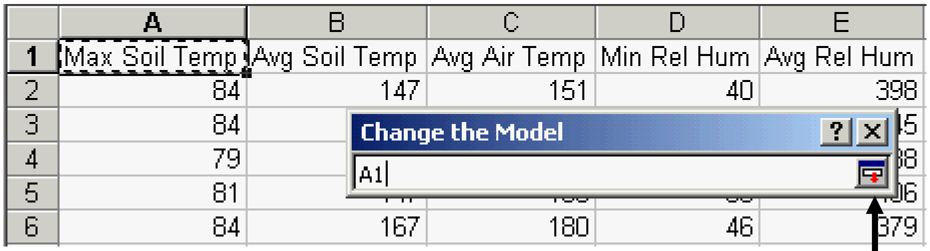


Field Names Evap!$A$1:$G$1

First Input Record A2:F2

As an alternative, if an area where you need to click and drag is hidden by a dialog box, instead of clicking in the range text box and then clicking in the Worksheet, click the ***Control*** in the range text box.

Clicking the Control causes the dialog box to automatically minimize as illustrated below - you do not have to click in the Worksheet. You can then click in the upper left cell of the range you want and drag to select all the cells in the range.



	A	B	C	D	E
1	Max Soil Temp	Avg Soil Temp	Avg Air Temp	Min Rel Hum	Avg Rel Hum
2	84	147	151	40	398
3	84				5
4	79				38
5	81				06
6	84	167	180	46	379

Note that in this example, since the range text box contained A1 when the Control was clicked, cell A1 is selected when the dialog box minimizes. While the dialog box is minimized, you may move it around by dragging its title bar, and you may select any range in the active Worksheet by simply clicking in an appropriate cell and dragging until the range you want has been selected (as indicated by the Excel dashed selection box). To restore the dialog box to full view, click the Control box again (note that when the dialog box is minimized, the Control box has a small red arrow to indicate it can be expanded).

> **IMPORTANT!** ***Never*** *click the close box in the title bar of the minimized dialog box. If you do, Excel will lock up, and the only way to resolve the problem is to end Excel, either using the Task Manager, or by restarting your computer. This is a problem with Excel that Microsoft has acknowledged; however, at the time Predict 3.1 was released Microsoft had not provided a solution.*

Using the Keyboard

If you prefer to use a keyboard rather than a mouse, you can navigate in *Predict* by using a combination of Alt-*[Key]* followed by *[Key]* keystrokes to execute commands and open the main dialog boxes. You can then use the Tab key to move from field to field within a dialog box.

Predict commands and dialog box elements that can be activated directly by an Alt-*[Key]* combination are indicated by an underlined character in the command or in the label for the dialog box element. For example, the **Alt-P** combination opens the *Predict* menu. After the menu is visible, pressing the **N** key will open the **New Model** dialog box.

In most dialog boxes, pressing the **Esc** key serves the same effect as clicking the **Cancel** button in the dialog box, and pressing the **Enter** key serves the same purpose as clicking the **OK** button (the **Esc** key does not act like the **Cancel** button in Wizard dialog boxes, and the **Enter** key causes no action unless a button has the focus in a dialog box).

> **Note:** Most Excel keyboard shortcuts also work when a dialog box is minimized as a result of clicking the Control in a range text box. However, the Excel operations that depend on pressing and releasing the **End** key do *NOT* work.

Tutorial

This tutorial shows how to use *Predict* to solve an actual problem. No prior training in neural computing is necessary; however, you should be familiar with basic Excel operations such as opening and saving files, selecting and copying blocks of cells, and creating line graphs. The tutorial will cover:

- Loading data into Excel.
- Specifying the problem type.
- Specifying data and network characteristics.
- Saving the model.
- Training the model.
- Testing the model.
- Analyzing the results.
- Running new data through your model.
- Closing the model and Microsoft Excel.

As you work through the tutorial you will gain a basic understanding of how *Predict* works and experience how easy it is to build a neural network model.

Problem and Data Description

The problem you will be working through is representative of a large class of problems known as inferential sensing. Inferential sensing uses a model of a physical process to measure and predict the output of that process — much as a sensor measures the actual output.

You will build a model of how daily moisture evaporation from soil is affected by the following six meteorological conditions:

- Maximum soil temperature
- Average soil temperature
- Average air temperature
- Minimum relative humidity
- Average relative humidity
- Total wind

These conditions are input variables for the model. The daily evaporation rate is the single model output variable. Measured values for these variables were gathered over 46 days. In addition, simulated data for 46 days was generated, yielding a total of 92 daily records.

This data is stored in the file `Evap.xls` that is shipped with *Predict*. The file is located in the directory where you installed *Predict*. The file contains seven columns of data. The first six columns are input values. The last column contains the corresponding output value. The 92 rows (records) in the spreadsheet correspond to one set of data (input and output) values for each day.

Loading the Data

To load the tutorial data into Microsoft Excel:

1. Start Excel.
   - If *Predict* is properly installed, the Excel menu bar includes the **Predict** menu.
   - If the **Predict** menu is *not* visible in the Excel menu bar, you can manually add it using the Excel **Tools | Add-Ins** menu. See the “Installation and Software License Activation” chapter in the *NeuralWorks Predict User Guide* for details.
2. Select **File | Open** from the Excel menu. The Excel **Open** dialog box appears.

⇨ The instruction “Select **File | Open”** means “Click the **File** menu, then click the **Open c**ommand.” Note that Excel’s File menu is used to access data (*not* the *Predict* menu).

These **Menu | Command** sequences are used throughout this tutorial.

3. Scroll through the directories in the list and select the directory where you installed *Predict* (`C:\Program Files\NeuralWare\NeuralWorks\Predict` by default).
4. In the list box labeled Files of type, select Microsoft Excel Files (.xl*;.xls;...).
5. Select `evap.xls`, the sample data file, from the files listed in the File browser pane.
6. Click **Open**.

Saving the Data

To avoid over-writing the original example data file, save the data in a new file named `tutorial.xls`. To save the data:

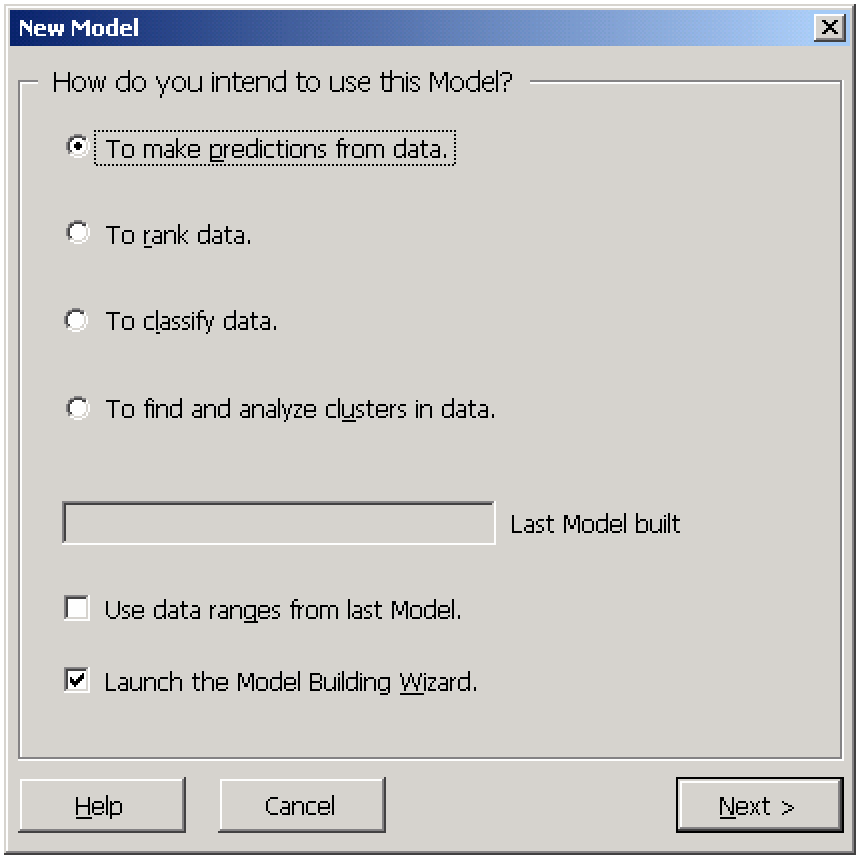
1. Select **File | Save As..**.
   Excel’s **Save As** dialog box appears.
2. Confirm that the Save as Type is Microsoft Excel Workbook.
3. Enter `tutorial` in the box labeled File name.
4. Click **Save**.
   The sample data is stored in the file `tutorial.xls`.
5. If the **Summary Information** dialog box appears, enter any descriptive information you would like about this worksheet and then click **OK**. Alternatively, click **Cancel** to close the dialog box.

Starting the Model Building Wizard and Selecting the Problem Type

In this tutorial you will use the *Predict* Model Building Wizard to quickly build an effective model. You do not have to understand the underlying technology; you simply need to tell *Predict* what data to use and how thoroughly to evaluate solutions, and it will do the rest.

1. From the **Predict** menu, select **New. . .** to create a new model.
   (Note: to create a new model, there must be an open workbook with an active worksheet that contains the data – the `tutorial.xls` workbook should still be open.)

2. The **New Model** dialog box appears. In it, you enable the Model Building Wizard and select the type of model that is appropriate for the problem. First, make sure the **Launch the Model Building Wizard** check box is selected. Then, since the output values for the evaporation data are continuous values, select the **To make predictions from data** option.



New Model

How do you intend to use this Model?

● To make predictions from data.

○ To rank data.

○ To classify data.

○ To find and analyze clusters in data.

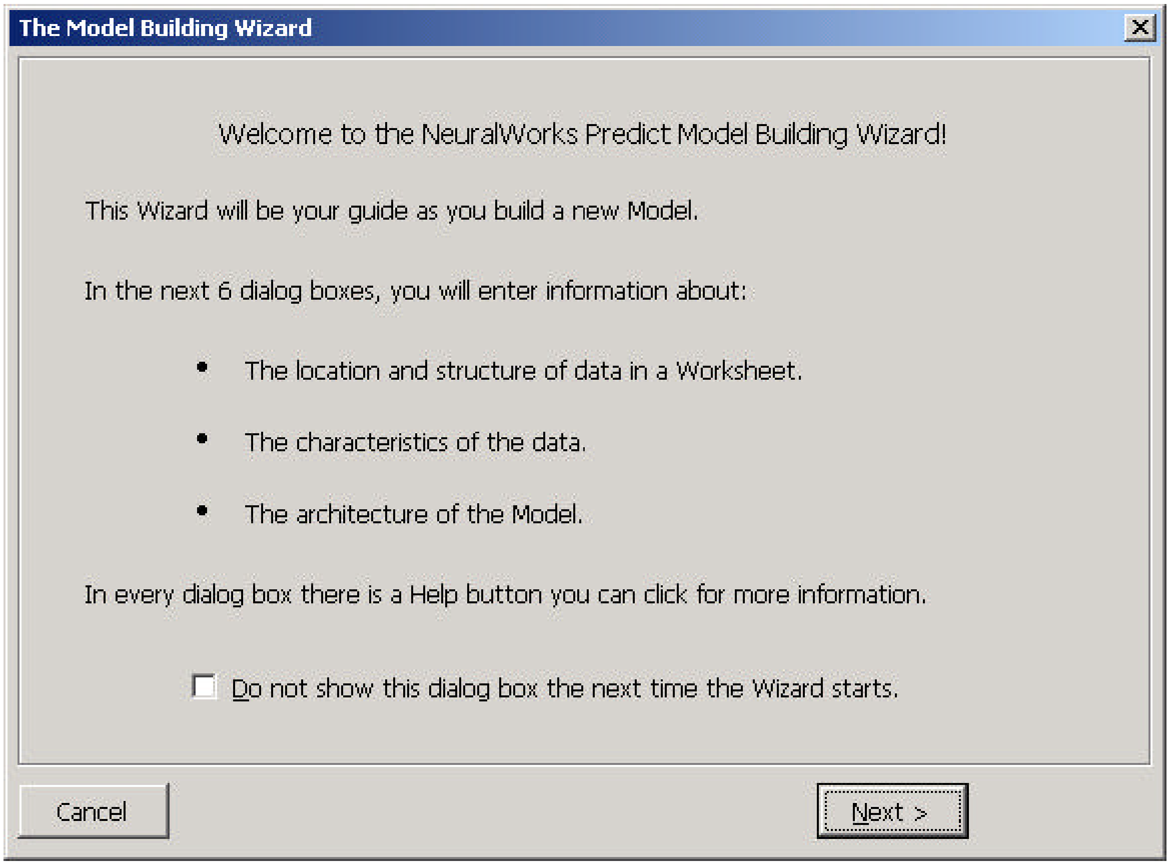
Last Model built

□ Use data ranges from last Model.

☑ Launch the Model Building Wizard.

3. Click **Next** to proceed.

The **Model Building Wizard** dialog box summarizes the actions needed to create a model. If you do not wish to see this Welcome screen each time you build a model with the Wizard, click the check box for **Do not show this dialog box the next time the Wizard starts**.



The Model Building Wizard

Welcome to the NeuralWorks Predict Model Building Wizard!

This Wizard will be your guide as you build a new Model.

In the next 6 dialog boxes, you will enter information about:

- The location and structure of data in a Worksheet.
- The characteristics of the data.
- The architecture of the Model.

In every dialog box there is a Help button you can click for more information.

□ Do not show this dialog box the next time the Wizard starts.

Click **Next** to proceed to the first of the six steps to create a model.

Characterizing the Model

When the Model Building Wizard is launched, it displays six dialog boxes in sequence. Through the dialog boxes you supply information that describes the available data and you set parameters that govern how the neural network model will be trained.

⇨ The following instructions for describing the layout of data in Excel assume that Excel's default Reference Style (A1) is active. Dialog box illustrations show how each dialog box should appear *after* you have performed the corresponding step.

Model Name and Data Organization

The first three dialog boxes are used to specify the model name and the location and orientation of the field labels and data values in the Excel worksheet.

Building a Model (Step 1 of 6) – Model and Field Names

The top area of this dialog box allows you to name the model and enter a description of the model. The lower area of the dialog box allows you to specify labels (names) for input and output fields.

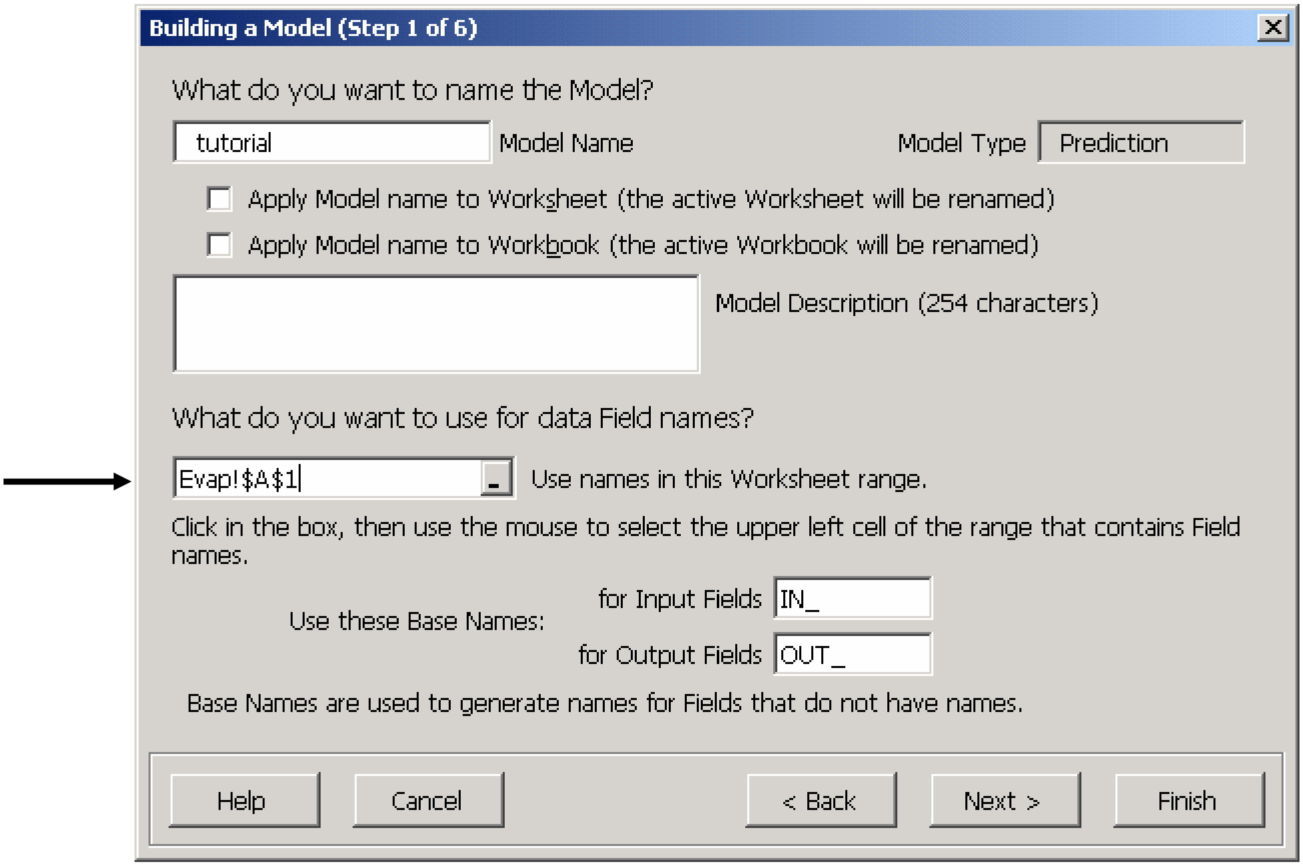
Model Name

1. In the first text box (labeled **Model Name**), confirm that `tutorial` is the model name (later you will select the directory in which to save the model).
2. If you wish, enter a description of the model in the text box labeled **Model Description**.

Field Names

The box labeled **Use names in this Worksheet range** allows you to specify cells that contain data field names. You can select all cells that contain field names, or you can simply specify the field name for the first input and the remaining fields will be recognized based on the layout of the data.

1. Click in the box labeled **Use names in this Worksheet range**. Then click cell A1 in the Excel worksheet.
2. Base Names are not used in this tutorial since all fields have names. However, leave the default Base Names as IN_ and OUT_.



Building a Model (Step 1 of 6)

What do you want to name the Model?

Model Name: tutorial

Model Type: Prediction

□ Apply Model name to Worksheet (the active Worksheet will be renamed)

□ Apply Model name to Workbook (the active Workbook will be renamed)

Model Description (254 characters)

What do you want to use for data Field names?

Use names in this Worksheet range: Evap!$A$1

Click in the box, then use the mouse to select the upper left cell of the range that contains Field names.

Use these Base Names:

for Input Fields: IN_

for Output Fields: OUT_

Base Names are used to generate names for Fields that do not have names.

Verify that the dialog box appears as shown above, then click **Next.**

Building a Model (Step 2 of 6) – Location of Input Data

This dialog box allows you to specify the layout of input data fields in the Excel Worksheet. Note that the information you enter in this dialog box is for *INPUT* fields only.

INPUT cells in the first data record

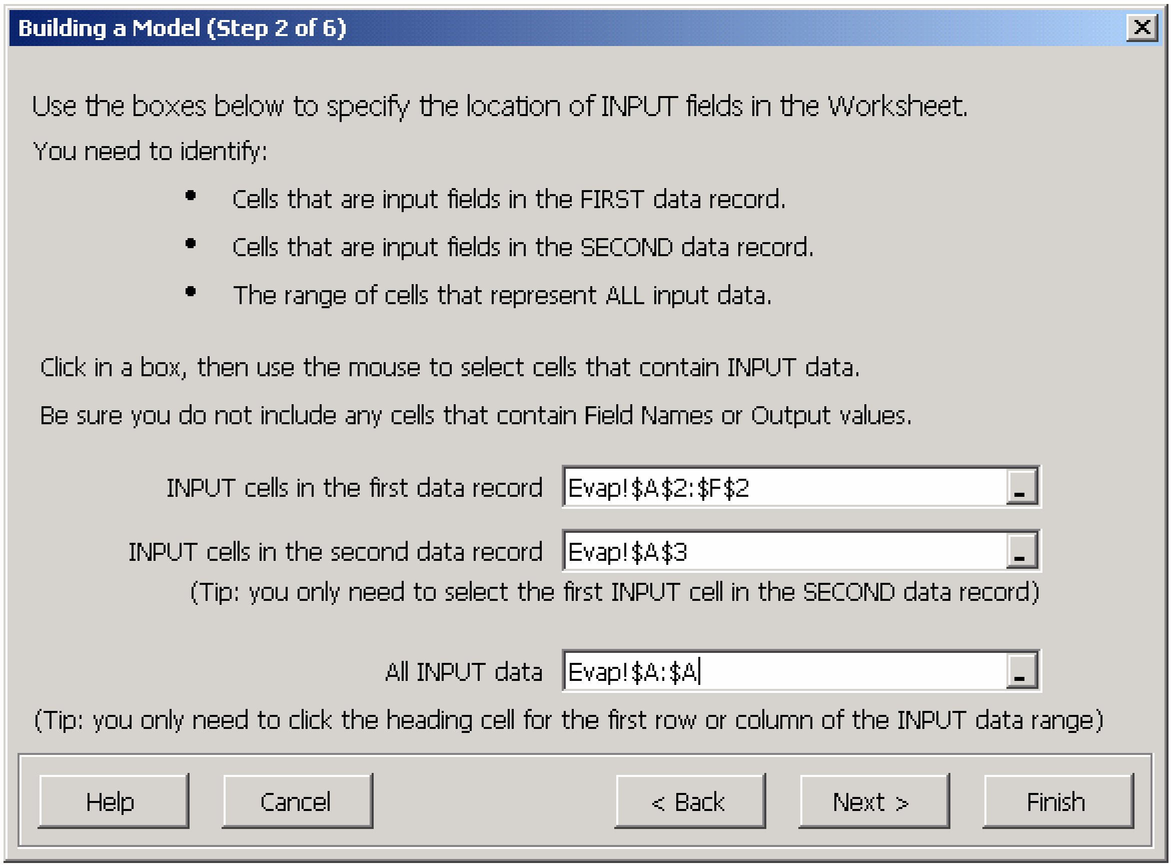
Use one of the following two methods to specify the location of the first input record in the Worksheet:

1. Click in the box labeled **INPUT cells in the first data record**. Then, drag the mouse over cells A2 through F2 in the Worksheet.

OR

2. Simply type A2:F2 in the box labeled INPUT cells in the first data record.

⇨ *Note that in any Predict dialog box, to enter a cell range you can either (1) click in the box where you want to enter the range and then drag the mouse cursor over cells in the Worksheet that make up the actual range, or (2) you can simply type the range (using the Excel A1 Reference style) in the box. If you use the mouse method to select a spreadsheet range, the dialog box will collapse while you are dragging over cells. The dialog box will reappear full size when you stop dragging and release the mouse button.*



Building a Model (Step 2 of 6)

Use the boxes below to specify the location of INPUT fields in the Worksheet.

You need to identify:

- Cells that are input fields in the FIRST data record.
- Cells that are input fields in the SECOND data record.
- The range of cells that represent ALL input data.

Click in a box, then use the mouse to select cells that contain INPUT data.

Be sure you do not include any cells that contain Field Names or Output values.

INPUT cells in the first data record: Evap!$A$2:$F$2

INPUT cells in the second data record: Evap!$A$3

(Tip: you only need to select the first INPUT cell in the SECOND data record)

All INPUT data: Evap!$A:$A

(Tip: you only need to click the heading cell for the first row or column of the INPUT data range)

INPUT cells in the second data record

Use the second box to specify the location of the input values for the second data record in the spreadsheet. You can specify the entire second data record, A3:F3 (either using the mouse or entering the range), or you can simply specify the first input of the second record, A3, and *Predict* will assume the record is of the same length and orientation as the first record. Using the two ranges you enter, *Predict* determines the relative locations of data records and then determines the layout of the remaining records.

1. Click in the second entry box labeled **INPUT cells in the second data record**.

2. Either type A3 in the box or click on cell A3 in the Worksheet.

In addition, given the layout of input data fields, *Predict* assumes that the same relative relationship exists among the locations of the *output* fields in records.

All INPUT data

Use the last box to specify the location of all the data to be used in building the model. You can specify the cells that contain inputs for all records, A2:F93, or you can simply click the heading cell for the first column of data (the heading cell for column A).

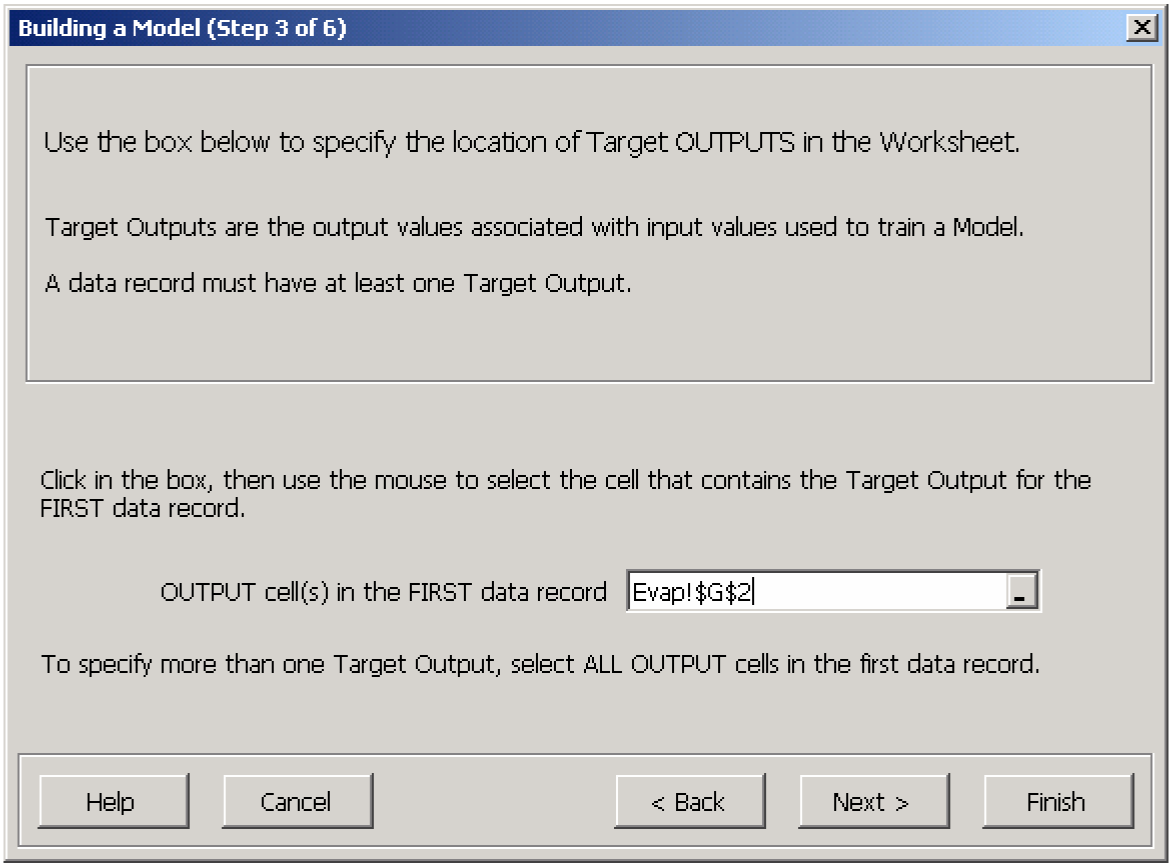
1. Click in the third box labeled **All INPUT data**.
2. In the Excel Worksheet, click the heading cell for the first column of data (the cell labeled **A**). This will select the entire column of data.

Verify that the dialog box appears as shown on the previous page, then click **Next**. Of course, if you manually entered cell ranges, the dialog box on the previous page will be different from your dialog box.

Building a Model (Step 3 of 6) – Location of Target Outputs

This dialog box is used to specify the cell or range of cells that contain the *output* field values (the target values) for the *first* record. With this information and knowledge of the relative locations of the *input* fields, *Predict* can determine the location of the remaining target values.

1. Click in the box labeled **OUTPUT cell(s) in the first data record**.
2. Either type G2 in the box or click in cell G2 in the Worksheet.



Building a Model (Step 3 of 6)

Use the box below to specify the location of Target OUTPUTS in the Worksheet.

Target Outputs are the output values associated with input values used to train a Model.

A data record must have at least one Target Output.

Click in the box, then use the mouse to select the cell that contains the Target Output for the FIRST data record.

OUTPUT cell(s) in the FIRST data record: Evap!$G$2

To specify more than one Target Output, select ALL OUTPUT cells in the first data record.

Verify that the dialog box appears as shown above, then click **Next**.

Data and Problem Characteristics

The next two dialog boxes are used to specify the nature of the data and the amount of processing that *Predict* will perform during model building. The choices you make in these dialog boxes determine the values for groups of parameters that are used in the model building process. As you gain experience using *Predict*, you can modify specific individual parameters to explore a wider range of neural network models.

Building a Model (Step 4 of 6) - Data Properties

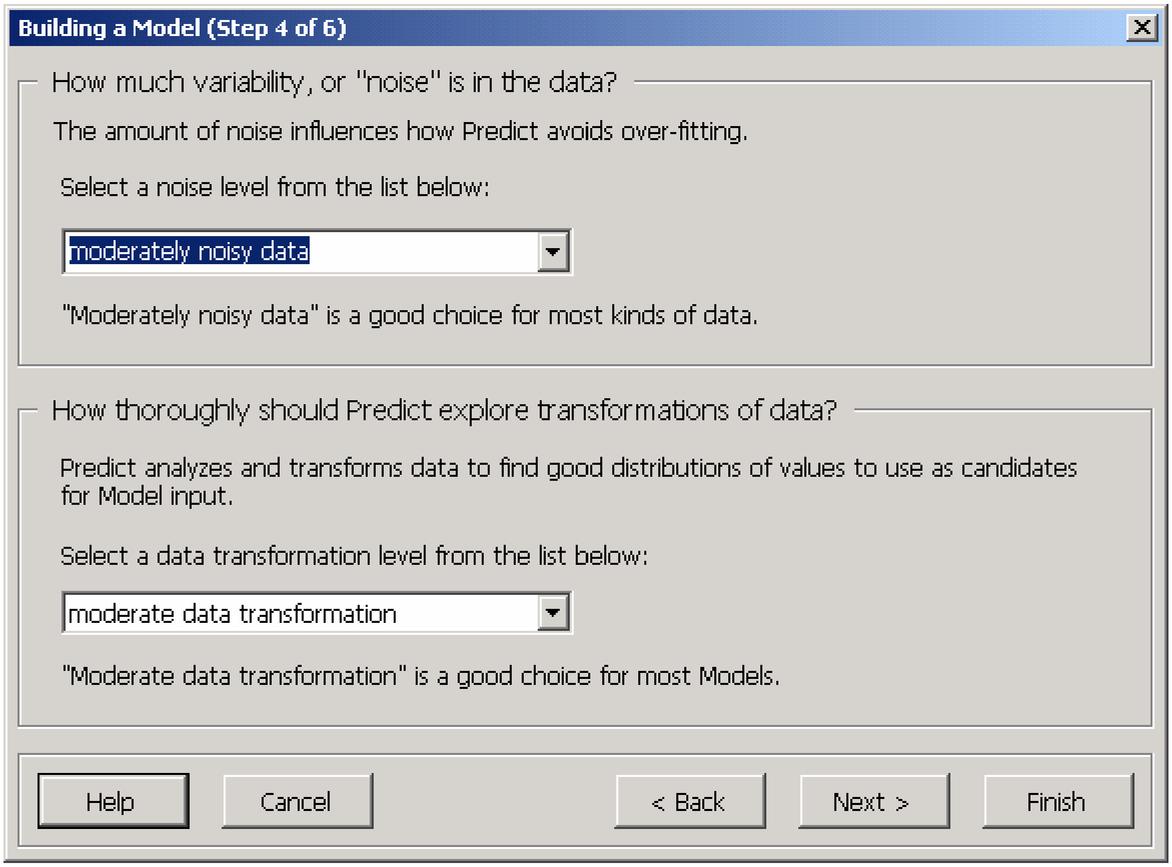
The top frame of this dialog box is used to indicate the amount of ***noise*** (variability) in the data. When the data is based on physical measurements, as the evaporation data is, it usually contains some measurement noise. Other types of data (such as behavioral or financial data) typically contain more noise.

- For the tutorial, select **moderately noisy data**.

In the bottom frame of the dialog box, you indicate how thoroughly you want *Predict* to try alternate transformations of the raw data. *Predict* analyzes all data fields and develops alternative sets of transformations for each one, according to the level specified. It then selects and combines those transformations and chooses a final set that produces the best network output.

Generally it is unwise to choose a high level for data transformations when the problem has many input variables. A large number of transformations increases the time needed to build a model. A high transformation level is not even advisable for problems with small sets of input variables, since the resulting larger number of variables may contribute to over-fitting. A moderate level of transformations is adequate for most applications.

- For the tutorial, select **moderate data transformation**.


Building a Model (Step 4 of 6)

How much variability, or "noise" is in the data?

The amount of noise influences how Predict avoids over-fitting.

Select a noise level from the list below:

moderately noisy data

"Moderately noisy data" is a good choice for most kinds of data.

How thoroughly should Predict explore transformations of data?

Predict analyzes and transforms data to find good distributions of values to use as candidates for Model input.

Select a data transformation level from the list below:

moderate data transformation

"Moderate data transformation" is a good choice for most Models.

Verify that the dialog box appears as shown above, then click **Next**.

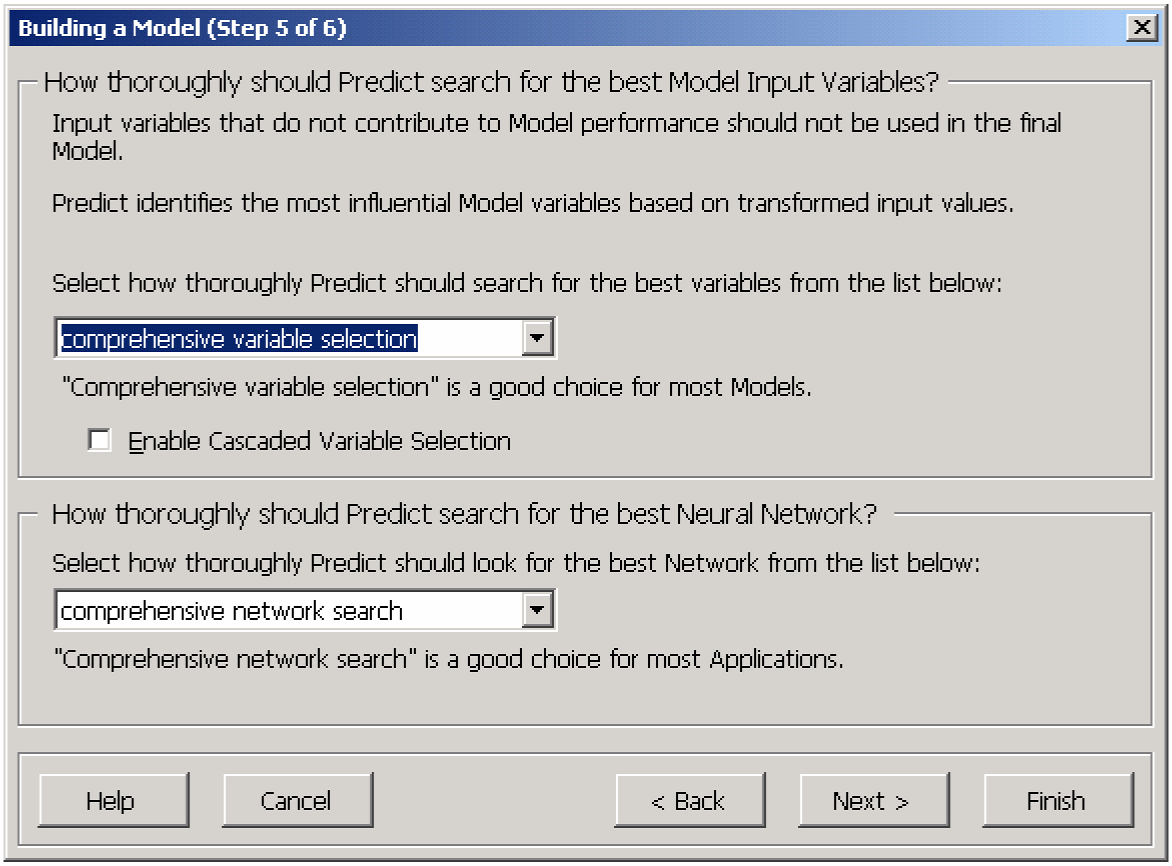
Building a Model (Step 5 of 6) - Variable Selection and Network Search

The options available in this dialog box are used to indicate how much computation effort *Predict* should use when building a model. In the top frame of this dialog box you indicate how extensively you want *Predict* to search for combinations of input variables that produce good results. Higher levels for variable selection require more time and memory than lower settings. Comprehensive Variable Selection is recommended for most problems.

1. For the tutorial, select **comprehensive variable selection**.
2. Make sure **Enable Cascaded Variable Selection** is **not** selected.

In the bottom frame of the dialog box you indicate how thoroughly you want *Predict* to construct alternate networks to train the model. A more thorough search may produce a better solution but will also require more time. A comprehensive network search is recommended for most problems.

1. For the tutorial, select **comprehensive network search**.



Building a Model (Step 5 of 6)

How thoroughly should Predict search for the best Model Input Variables?

Input variables that do not contribute to Model performance should not be used in the final Model.

Predict identifies the most influential Model variables based on transformed input values.

Select how thoroughly Predict should search for the best variables from the list below:

comprehensive variable selection

"Comprehensive variable selection" is a good choice for most Models.

☐ Enable Cascaded Variable Selection

How thoroughly should Predict search for the best Neural Network?

Select how thoroughly Predict should look for the best Network from the list below:

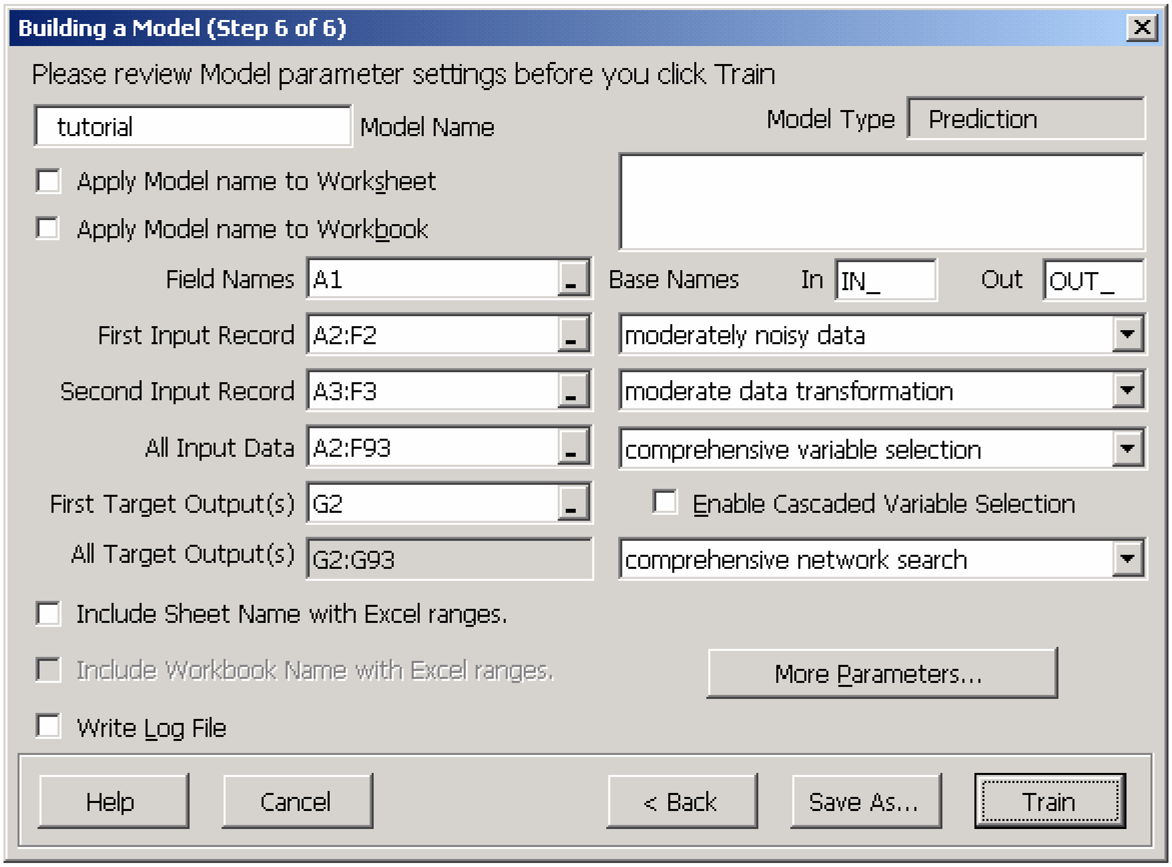
comprehensive network search

"Comprehensive network search" is a good choice for most Applications.

Verify that the dialog box appears as shown above, then click **Next**.

Building a Model (Step 6 of 6) – Parameter Review and Training

The final Wizard dialog box summarizes the selections you made in the first five Wizard dialog boxes.



Building a Model (Step 6 of 6)

Please review Model parameter settings before you click Train

Model Name: tutorial

Model Type: Prediction

□ Apply Model name to Worksheet

□ Apply Model name to Workbook

Field Names: A1

Base Names In: IN_ Out: OUT_

First Input Record: A2:F2

moderately noisy data

Second Input Record: A3:F3

moderate data transformation

All Input Data: A2:F93

comprehensive variable selection

First Target Output(s): G2

□ Enable Cascaded Variable Selection

All Target Output(s): G2:G93

comprehensive network search

□ Include Sheet Name with Excel ranges.

□ Include Workbook Name with Excel ranges.

□ Write Log File

Verify that the dialog box appears as shown above. If any changes are necessary, you can make them directly in this dialog box.

At this point, the model has not yet been trained or saved; you have only specified parameters that define the architecture of an untrained model. Before *Predict* starts training the model you will be asked to save it. To start training:

1. Click the **Train** button. When the **Save Model** dialog box appears, browse to select a directory to work in. For this tutorial, select `C:\My Documents` (if `My Documents` does not exist, select another directory on your computer).

2. Confirm that the Save as type: field contains `Predict Models (*.npr)`. The default name for the file is `tutorial`, the name that you specified in the first Wizard step.

3. Click **Save**.
   Predict will store the model you are building in this file, unless *Predict* is running in demonstration mode. If *Predict* is running in demonstration mode, the file is not actually saved, but you can continue with the tutorial.

Training will start automatically after you click **Save** in the **Save Model** dialog box. Training the model will take about 10 seconds on a Pentium 500MHz processor. The process proceeds as described in the following section.

Building the Model

The model-building process consists of five stages:

- Data partitioning –

  First, *Predict* automatically selects train, test, and validation datasets from the available input data you specified. By default, *Predict* uses 70% of the data for training and 30% for testing. *Predict* attempts to partition the data such that the train and test sets have approximately equal distributions of output values.

- Data analysis –

  Next, *Predict* examines the values for each data field to determine the type of field data. For example, fields may contain continuous values, enumerated integers, or string (literal) data. *Predict* encodes all data fields in ways that are appropriate for neural network processing.

- Data transformation –

  Predict applies various transforms and outlier removal procedures to the field data in order to get a more uniform distribution of records over each field. This helps the neural network learn from small differences in the records.

- Input variable selection –

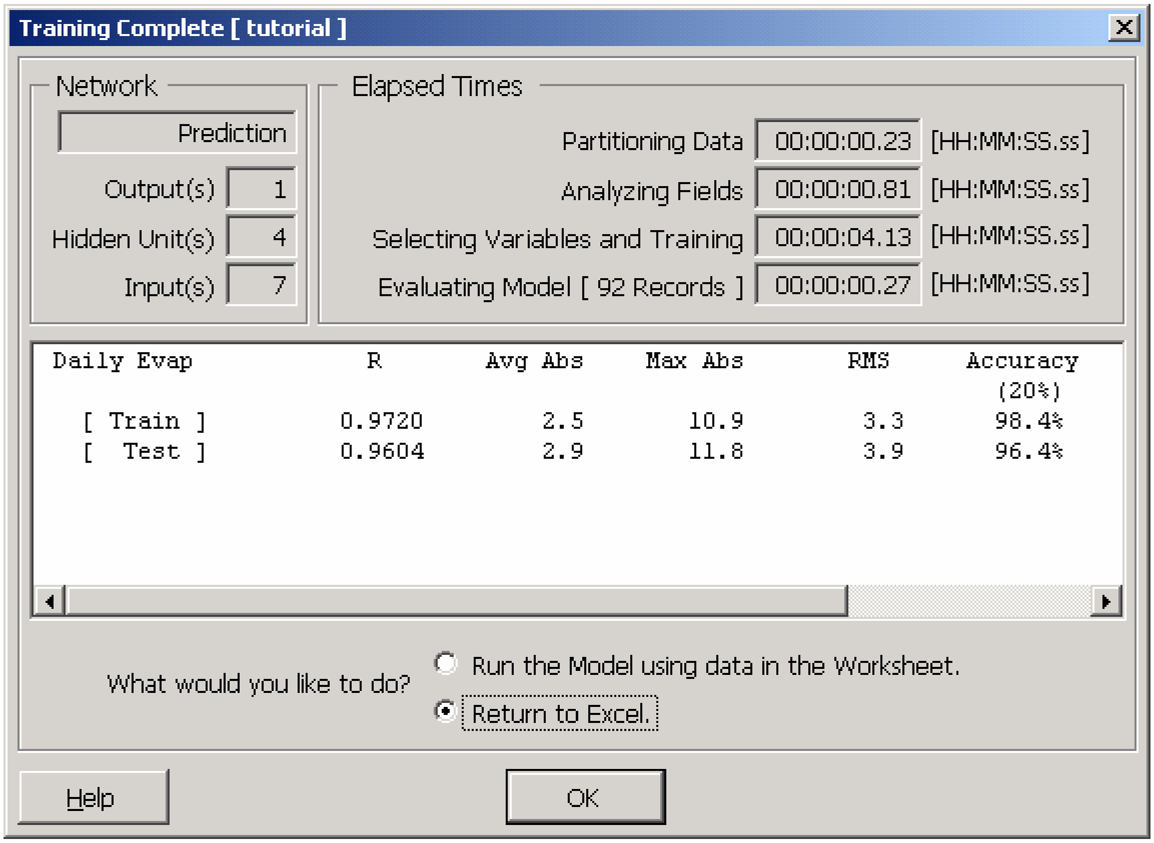
  Predict uses a genetic algorithm to select a good subset of input variables from the set of all input variables and transformations of input variables. A genetic algorithm is used since it efficiently explores the large space of subsets of possible input variables.

- Training the neural network –

  During training, the weights of the neural network are adjusted in order to minimize the error between the network output and the target value for all of the records in the training set. In addition to adjusting weights, new processing elements are added in order to decrease the error of the network. This method of incrementally adding processing elements is called cascade correlation. To ensure that the network does not over-fit the training data (by learning patterns specific only to the training set), the performance of the network on the test set is periodically evaluated. When performance on the test set begins to degrade, training is stopped.

In the first three stages of the model-building process, *Predict* displays progress messages in the Excel status bar. As training progresses, *Predict* displays information about variable selection and test scores for candidate models. (The model-building process and these status messages are described in detail in *Predict Help* and the *NeuralWorks Predict User Guide*.)

After the model is trained, a dialog box similar to the one shown below appears.



Prediction

Output(s) 1

Hidden Unit(s) 4

Input(s) 7

Partitioning Data 00:00:00.23 [HH:MM:SS.ss]

Analyzing Fields 00:00:00.81 [HH:MM:SS.ss]

Selecting Variables and Training 00:00:04.13 [HH:MM:SS.ss]

Evaluating Model [92 Records] 00:00:00.27 [HH:MM:SS.ss]

```
Daily Evap              R        Avg Abs    Max Abs      RMS      Accuracy
                                                                    (20%)
  [ Train ]          0.9720        2.5       10.9        3.3       98.4%
  [  Test ]          0.9604        2.9       11.8        3.9       96.4%
```

What would you like to do?
○ Run the Model using data in the Worksheet.
● Return to Excel.

Congratulations, you have now built and trained a neural network model!

The **Training Complete** dialog box is a quick summary that shows the final architecture, the elapsed time for major model building steps, and the accuracy of the model when run on the Train and Test datasets. Additional model verification operations are discussed in greater detail in the next section.

- Make sure the option button labeled **Return to Excel** is selected, then click **OK**.

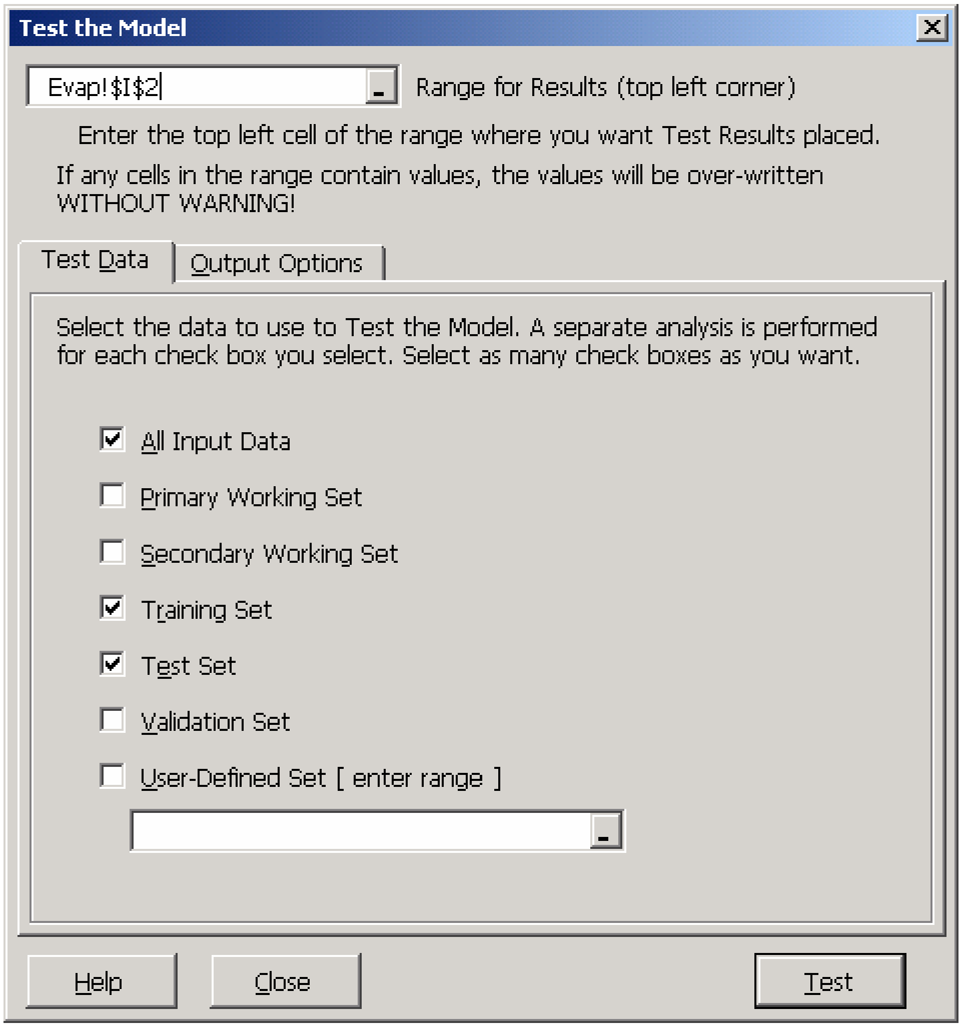
At this point you should save the model. From the **Predict** menu, select **Save**. Assuming you have performed all the steps as described in this tutorial, the `tutorial.npr` project file that contains the neural network will be saved in the directory you specified earlier.

⇨ If *Predict* is running in demonstration mode, the **Save** command is not available – it appears dimmed in the **Predict** menu. Please continue with the tutorial, even if you cannot save the model.

Testing the Model

In this section, we will explore how to test the model using the Test command. We will generate the same results that we saw in the Training Complete dialog box but we will see how to use the Test the Model dialog box to write test results to the spreadsheet.

1. Select **Predict | Test**. The **Test the Model** dialog box appears.



Test the Model

Evap!$I$2 Range for Results (top left corner)

Enter the top left cell of the range where you want Test Results placed.

If any cells in the range contain values, the values will be over-written WITHOUT WARNING!

Select the data to use to Test the Model. A separate analysis is performed for each check box you select. Select as many check boxes as you want.

☑ All Input Data
□ Primary Working Set
□ Secondary Working Set
☑ Training Set
☑ Test Set
□ Validation Set
□ User-Defined Set [enter range]

2. In the top area of the dialog box, enter the range where you want *Predict* to write test results. Depending on the type of test that you choose, the results could take several columns and rows of cells. Any data that may be in cells that are used for test results will be overwritten. Select cell `I2` or enter `I2` as the location of the top left cell of the test results table that will be written to the worksheet.

3. Select the following data sets (if necessary, deselect the other options). *Predict* will test the model on these sets, calculate statistical scores for each set, and write the scores to the worksheet.

   All Input Data
   Training Set
   Test Set

⇨ Note that by default the validation set consists of all the available data. Therefore, the default validation set is not a true independent validation set. However, you can create an independent validation set through the **Train/Test** tab of the **Model Parameters** dialog box.

4. Click the **Output Options** tab to view the other forms of testing that can be performed. These options are described in the "*Test the Model – Output Options*" section of the *Predict User Guide*. For this tutorial *DO NOT* change any output options. To return to the standard Test options, click the tab labeled **Test Data**.
5. Click the **Test** button. The model will be run with each of the datasets you selected, and model performance statistics will be generated.
6. After statistics are written to the spreadsheet, click the **Close** button.

The statistical summary will be similar to the summary shown below, although your results may differ somewhat. Such differences may be due to slight variations in parameter settings between the *Predict* release you are using and the release used to create this tutorial, or the differences may be due to the state of the random number generator as it is used in the various model-building stages.

Daily Evap	R	Net-R	Avg. Abs.	Max. Abs.	RMS	Accuracy (20%)	Conf. Interval (95%)	Records
All	0.967429	-0.94575	2.57944	11.82937	3.491898	0.978261	6.897066	92
Train	0.971988	-0.9503	2.455123	10.87935	3.302480	0.984375	6.568931	64
Test	0.960427	-0.93843	2.863592	11.82937	3.890373	0.964286	7.973434	28

The statistical results shown are:

R — The linear correlation between the target values and the corresponding predicted output values.

Net-R — The linear correlation between the target values and the *raw* network output values (before they are transformed into the measurement units of the problem).

Avg. Abs. — The average absolute error between predicted output values and the corresponding target values.

Max. Abs. — The maximum absolute error between predicted output values and the corresponding target values.

RMS — The root mean square error between predicted output values and the corresponding target values.

Accuracy — The percentage of predicted output values that are within the specified tolerance (20%) of the corresponding target values.

Conf. Interval — The confidence interval is the range [target value ± confidence interval] within which the corresponding predicted output occurs 95% of the time.

Analyzing the Results

Interpreting the Statistical Summary

Among the statistical results in the summary table, R, RMS, Accuracy, and Confidence Interval are the key indicators of how well a model performs.

For example, note the R values for the model on the training and test sets. The values are close to each other (0.9719 and 0.9604), which means the model generalizes well and is likely to make accurate predictions when new data (data that is not from the training or testing dataset) is provided. Furthermore, the correlation values are close to 1.0, which is another indication that the model performs well. However, this criterion applies only when the model training data is representative of data in the actual problem domain, and there exists a strong physical relationship between the input and output variables—as in the *evap* example data. In other problem domains, such as human behavior modeling, correlation values as low as 0.2 can indicate a good model.

Comparing Target and Predicted Values

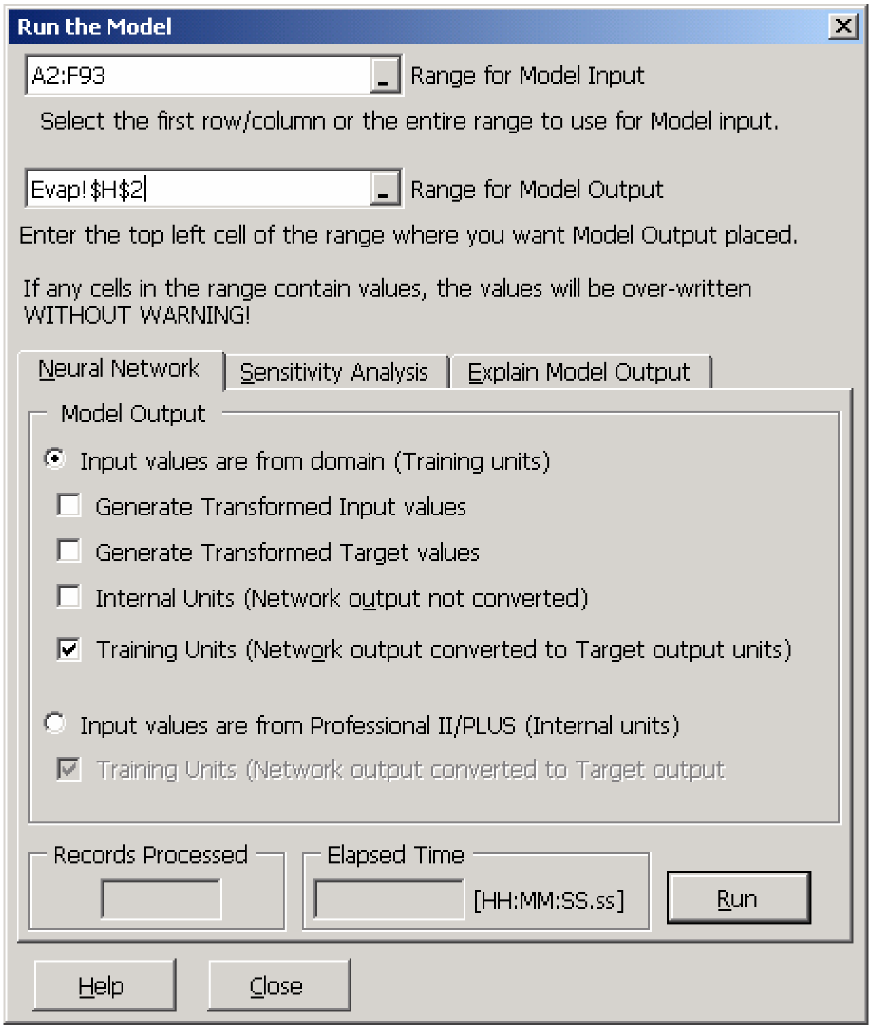
Another way to evaluate model performance is to compare individual target values to the corresponding predicted values produced by the model. This requires:

1. Computing the model's predictions of the daily soil evaporation rates.
2. Plotting the target and predicted values.

Computing the Predicted Values

To obtain the predicted values, you need to run all the data through the model. This can be done using the **Run** command.

1. Select **Predict | Run**. The **Run the Model** dialog box appears.



Run the Model

Range for Model Input: A2:F93

Select the first row/column or the entire range to use for Model input.

Range for Model Output: Evap!$H$2

Enter the top left cell of the range where you want Model Output placed.

If any cells in the range contain values, the values will be over-written WITHOUT WARNING!

Model Output

● Input values are from domain (Training units)

□ Generate Transformed Input values

□ Generate Transformed Target values

□ Internal Units (Network output not converted)

☑ Training Units (Network output converted to Target output units)

○ Input values are from Professional II/PLUS (Internal units)

☑ Training Units (Network output converted to Target output

Records Processed

Elapsed Time [HH:MM:SS.ss]

2. In the box labeled **Range for Model Input**, select the range for all the input data in the worksheet (A2 to A93), or type `A2:A93` in the box (this data range should already be the default range).

3. In box labeled **Range for Model Output**, you specify the range of cells into which you want *Predict* to write predicted values. Select cell H2 in the worksheet, or type H2 in the box.

4. Make sure the **Neural Network** page is visible (if it is not, click the **Neural Network** tab), and that there is a check mark in the box labeled **Training Units** (Network output converted to Target output units). The other check boxes should not be selected.

5. Click the **Run** button on the **Neural Network** page. *Predict* writes the predicted values to the worksheet.

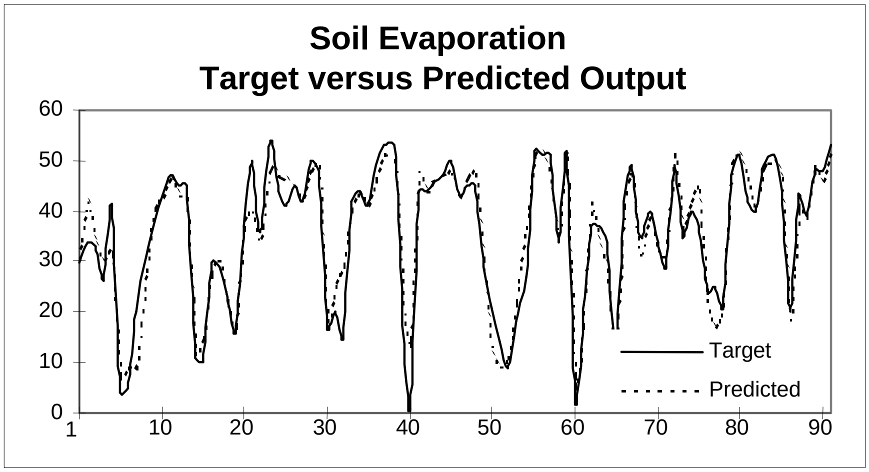
⇨ Instead of using the **Predict | Run** command, if *Predict* is not running in demonstration mode you can use the PredictRun macro function to pass a data record to the model. Simply type the function name, the path to the model, and the input range directly into cell H2 as follows:
=predictrun("C:\My Documents\tutorial.npr", A2:F2)
To apply this function to the remaining records, copy it from cell H2 and paste it into cells H3 through H93 (or, select cell H2 and all other cells in the column down to cell H93, and type **ctrl-d**).

6. Notice that the number of records processed and the time to run the model for those records is shown in the **Run Model** dialog box. Click **Close** to close the dialog box and return to the spreadsheet.

Plotting the Target and Predicted Values

You can also compare all (or a subset) of predicted and target values by creating an Excel line graph.

For example, to compare the 92 days of measured and predicted evaporation rates, simply select cells G2 through H93. Then click Excel's **Chart Wizard** button, or select **Insert | Chart**, and follow the instructions for creating a line graph. The resulting graph looks like this:



The small differences between the curves for the target and predicted values provide another indication that the model performs well.

Comparing Neural Networks with Linear Regression

Another way of evaluating a neural network model is to compare its results to results produced by some other model. For example, the following table shows the correlation values produced by a linear regression model (based on the same input data) and by the *Predict* neural network model.

	Training Set	Testing Set	All Data
Linear Regression	0.9149	0.9280	0.9088
Predict Model	0.9720	0.9604	0.9674

The *Predict* model has higher R values on all three data sets, which indicates that it performs better than the linear regression model.

Running New Data Through Your Model

Now that your model has been trained and tested, it is time to put it to use with new data that it has never seen before. There are many ways to "deploy" a model in *Predict* but the fastest way is simply to enter new data into the spreadsheet, and again use the **Predict | Run** command.

1 Locate an empty area in the spreadsheet (click in cell A95) and type the following numbers. Type across the row into cells A95:F95…

100 150 200 50 400 400

2. Select **Predict | Run**. The **Run the Model** dialog box appears.
3. Click in the first entry box labeled **Range for Model Input**, and then click on cell A95. This is the first cell of the only record that we wish to process. If you had 10 rows of data you could enter A95:A104.
4. Click in the second box, labeled **Range for Model Output**, and select cell H95 in the worksheet.
5. Click the **Run** button. *Predict* writes the predicted value to the worksheet.
6. Click **Close** to close the dialog box and return to the spreadsheet.

You can expect an answer around 52, but since this is not a physically measured data sample, we cannot be certain that 52 is correct. That's why *Predict* measures things using separate training and test data sets - so that your models perform well on unseen data.

The new data you entered did not have to reside in the same spreadsheet. Non-demonstration versions can open a new spreadsheet of data, open a saved *Predict* model, and then process data with **Predict | Run**.

There are several ways to put a final model into use. Other ways involve the use of the dynamic function PredictRun as previously explained, or converting the model into C, Visual Basic or Fortran source code and embedding into your own programs, or calling our DLL functions directly from your applications. You can even create and train models directly from your programs!

Closing the Model and Microsoft Excel

Closing a model releases memory that *Predict* used for the model. To close a model:

- Select **Predict | Close**.
  If a prompt asks whether you want to save the model, click **Yes** (if *Predict* is running in demonstration mode you will not be able to save the model).

To close the workbook:

1. From the Excel **File** menu, select the **Close** command.
2. If a prompt asks whether you want to save the workbook, click **Yes**.
3. If the **Summary Info** dialog box appears, enter descriptive information about this worksheet, and then click **OK**. Or, simply click **Cancel** to close the dialog box.

Congratulations, and thank you for completing this tutorial. We hope you found it interesting and helpful. You have created and trained a neural network model (`tutorial.npr`) that accurately predicts the moisture evaporation rate from soil, but this is just the beginning of features in *Predict*.

Do you like what you have seen so far? Are you interested in topics like data transformation, how *Predict* chooses its inputs (variable selection via genetic algorithm) or sensitivity analysis? For these subjects and to gain more experience using *Predict*, please explore the 'Credit Risk' tutorials found in Appendix C. Credit tutorials show much more of the kind of things you can get from your own data in *NeuralWorks Predict*!

Appendix A – Installing NeuralWorks Predict

System requirements and installation instructions can also be found in the printed `ReadMe` file that is shipped with *Predict*.

The "Installation and Software License Activation" chapter of the *NeuralWorks Predict User Guide* provides additional installation information, including how to install and use *Predict* on RISC/UNIX® workstations, how to run the *Predict* Command Line Interface, and how to remove *Predict* software from your computer.

System Requirements

Hardware

- Processor: 32-bit x86 Pentium-class processor (Intel® or compatible)
- Memory: 16 MB RAM minimum; 64 MB or more strongly recommended
- Disk space: 26 MB (for *Predict* files – additional space will be needed for data and model files)
- CD-ROM: CD-ROM drive only required for installation

A Microsoft Windows 32-bit Operating System

- Windows 98
- Windows ME
- Windows NT version 4.0 or later
- Windows 2000
- Windows XP

32-bit Microsoft Excel (not required if you use only the *Predict* Command Line Interface)

- Version 8.*x* (Excel 97)
- Version 9.*x* (Excel 2000)
- Version 10.*x* (Excel XP)

Installing NeuralWorks Predict

For *Predict* to automatically integrate with Excel, Excel must be installed before you install *Predict*.

Installing *Predict* consists of the following steps:

- Removing (if applicable) earlier releases of *Predict* or NeuralSIM (NeuralSIM was the name used for some versions of *Predict* released in the late 1990's).

⇨ Please refer to the printed `ReadMe` document that is included with the Predict Distribution Package if you need to remove earlier releases of *Predict*. The `ReadMe` file is also put in the *Predict* home directory when you install *Predict*. The file is named `ReadMe`*`xx`*`.wri`, where *`xx`* represents the 2 digit version number for the product you purchased – the `ReadMe` for Predict version 3.1 is `ReadMe31.wri`).

- Installing *Predict*.
- Activating your *Predict* software license.

Installing the Software

Note: Windows NT, Windows 2000, and Windows XP users must have **Administrator** privileges to install *Predict* software.

To install *Predict* software on your Windows computer:

1. Insert the NeuralWorks Predict CD-ROM into the CD-ROM drive on your computer.
2. Wait a few seconds for Windows to automatically run Setup.

   If Setup does not automatically start, manually start Setup:

   - select Run from the Windows Start menu.
   - type **`X:\SETUP`** in the Run dialog box (replace ***X*** with the appropriate CD-ROM drive letter).
3. Follow the directions provided by Setup dialog boxes.

⇨ To learn about any last-minute changes, please read the file `ReadMe31.wri`, located in the directory where you installed *Predict*. You can open this file from the Windows **Start** menu:

`Programs | NeuralWare | NeuralWorks | Predict | ReadMe`.

⇨ ***Note that NeuralWorks Predict and software derived from NeuralWorks Predict are licensed for use on a single computer. To deploy any NeuralWorks Predict functionality on any other computer requires additional licenses and fees. Please contact NeuralWare Sales (call*** **`+1 412.278.6288`** ***or send email to*** **`sales@neuralware.com`*****) if you are interested in deploying NeuralWorks Predict functionality on other computers.***

Installing Acrobat Reader

After *Predict* is installed, the Setup program may ask if you want to install Acrobat Reader 5.05. If Acrobat Reader version 4.0 or later is not already installed on your computer, you should install the Acrobat Reader contained on the *Predict* distribution CD-ROM so that you can view and print the *NeuralWorks Predict User Guide* and this *NeuralWorks Predict Getting Started Guide for Windows*.

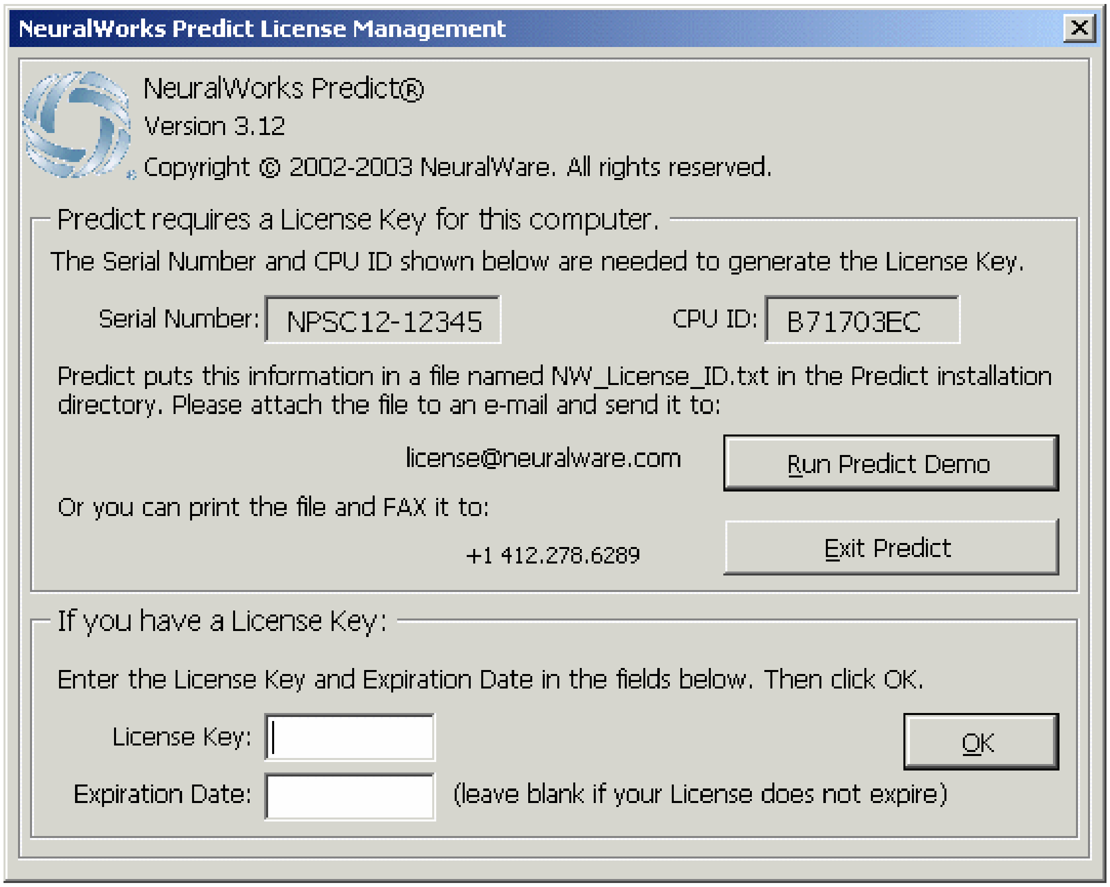
> *If you have an older version of Acrobat Reader installed, but you still would like to install the newer version that NeuralWare provides, you* ***MUST*** *remove the older version of the Reader before you start the Predict Setup program. If you do not remove the older version, the newer version will not install correctly.*

The first time you run the Acrobat Reader, you will be asked to accept the Adobe License Agreement before the document you are trying to open is displayed. **If you don't see the document you opened, make sure that another window isn't hiding the Acrobat Reader license acceptance dialog box**. After you accept the terms of the Adobe License Agreement, the document you opened will appear.

If you do not install Acrobat Reader during the *Predict* installation process, you can always install it from the *Predict* CD-ROM at a later time. Put the *Predict* CD-ROM into the CD-ROM drive, browse to the `acroread` directory, and double-click the file named `rp505enu.exe` to install Acrobat Reader.

Appendix B – Activating Your Software License

When you run *Predict* for the first time, after the NeuralWorks Predict banner dialog box closes a License Management dialog box similar to the one shown below appears. The dialog box you see will contain a different Serial Number and a different CPU ID. Demonstration versions of *Predict* will not see this.



NeuralWorks Predict®
Version 3.12
Copyright © 2002-2003 NeuralWare. All rights reserved.

Predict requires a License Key for this computer.

The Serial Number and CPU ID shown below are needed to generate the License Key.

Serial Number: NPSC12-12345 CPU ID: B71703EC

Predict puts this information in a file named NW_License_ID.txt in the Predict installation directory. Please attach the file to an e-mail and send it to:

license@neuralware.com

Or you can print the file and FAX it to:

+1 412.278.6289

If you have a License Key:

Enter the License Key and Expiration Date in the fields below. Then click OK.

License Key:

Expiration Date: (leave blank if your License does not expire)

This dialog box displays the information that NeuralWare needs to generate a License Key. The necessary information is also automatically placed in a file named `NW_License_ID.txt` in the directory where you installed *Predict*.

You must contact NeuralWare to obtain a License Key and, if applicable, the expiration date for the key.

To obtain a License Key:

1. Exit *Predict* –
   - Click the **Exit Predict** button in the License Management dialog box. Then exit Excel.
2. Open `NW_License_ID.txt` in a text editor (for example, Notepad).
   - Copy the information from Notepad, paste it in an e-mail message, and send it to:

     `license@neuralware.com`, or
   - Print the information and send a FAX to `+1 412.278.6289`.

OR, if you have an e-mail client installed on your computer, you can:

1. Exit *Predict* –

- Click the **Exit Predict** button in the License Management dialog box. Then exit Excel.

2. Use the Windows Explorer to locate the file `NW_License_ID.txt.` After you locate the file, right-click the file name.
3. Select **Send To | Mail Recipient** from the pop-up menu.

   (If Windows prompts you to select a profile, select an appropriate profile to use to send the License Key request.)
4. Enter `license@neuralware.com` in the To: field of the e-mail form.
5. Click the **Send** button in your e-mail client.

After NeuralWare returns the License Key (it will be sent back to you by the same method that you used to send the information to NeuralWare), start *Predict* as described earlier and enter the License Key in the License Key field in the dialog box. Leave the Expiration Date field blank unless NeuralWare provides an expiration date with the License Key. Click **OK** to close the License Management dialog box.

Running Predict in Demonstration Mode

Even before you receive a License Key, you can explore *Predict*'s features and capabilities. Start *Predict* (and Excel) as described earlier, then click **Run Predict Demo** in the **NeuralWorks Predict License Management** dialog box. The demonstration version of *Predict* limits the size of models that can be built to 32 fields and 512 data records, and models cannot be saved. Some of *Predict*'s advanced features (such as generating FlashCode) are also not available. However, if *Predict* is running in demonstration mode you can perform all the actions described in the Tutorial chapter in this guide except saving the network and as otherwise noted.

Using Predict Help

Predict Help, in Windows 32-bit Help format, contains the same information that is in the *NeuralWorks Predict User Guide*. However, *Predict Help* is hot-linked with *Predict* to offer context-sensitive Help. To view *Predict Help*:

- Click the **Help** button in any *Predict* dialog box.

  When you click the **Help** button in a dialog box, the *Predict Help* system displays context-sensitive information for the dialog box.
- From the **Predict** menu, click the **Help** command.

  When you click the **Help** command, the *Predict Help* **Table of Contents** appears. You can use the **Table of Contents** or the **Index** to find the topic you want help for.

Note that many illustrations in *Predict Help* have "hot spots" that are linked to specific topics. When the mouse cursor is over a hot spot, the mouse cursor changes to a pointing finger. When you click a hot spot, information related to the hot spot area is displayed.

Contacting NeuralWare

All NeuralWare products include free technical support for 60 days from the date of purchase. In addition, you can subscribe to NeuralWare's Technical Assistance Program (TAP), which provides both priority technical support and automatic shipment of upgrades to users of products registered in the Program.

If you encounter a problem while using *Predict*, please carefully review the *NeuralWorks Predict User Guide*, the *Predict Help,* and the `ReadMe` file (`ReadMexx.wri`) before contacting NeuralWare. If you still cannot resolve the problem, NeuralWare Product Support can be contacted at:

NeuralWare

230 E Main Street

Suite 200
Carnegie PA 15106-2700

Phone:	`+1 412.278.6280`	`(Press 2 for Technical Support)`
Fax:	`+1 412.278.6289`	
Web:	`www.neuralware.com`	
Email:	`sales@neuralware.com`	`(General Product Information and Sales)`
	`help@neuralware.com`	`(Product Technical Support)`

Customers who purchase a NeuralWorks Predict TAP subscription are given a special e-mail address to use for technical assistance questions .

Appendix C – Credit Risk Tutorials

Credit.txt contains simulated information that corresponds to information typically collected from loan applicants. The objective is to produce a model that demonstrates how personal attributes can be predictors of an applicant's ability to repay a loan. After the model has been trained, it can be used to guide decisions for granting or denying new loan applications.

Credit Risk Tutorial 1

In this tutorial you will:

- Build a new Multi-Layer Perceptron (MLP) model using the Model Building Wizard
- Evaluate the model's performance
- Review *Predict's* choice of model input variables
- Run the model

Building a Model using the Model Building Wizard

1. Start Microsoft Excel. If NeuralWorks *Predict* is installed correctly, the **Predict** menu appears in the Excel menu bar.

If you do not see the **Predict** menu, or if any error messages appear when you start Excel, please refer to **Manually Linking Predict with Microsoft Excel** before preceding any further.

2. Open CREDIT.TXT using the **Open** command in the Excel **File** menu (use the mouse, or type Alt F, O). The file is in the *Predict* directory you specified during installation (the default location is `C:\Program Files\NeuralWare\NeuralWorks\Predict)`. In the **Open** dialog box, set the **Files of type** field to *Text Files*, and then select CREDIT.TXT. The Excel *Text Import Wizard* dialog box appears. Click **Finish** to import CREDIT.TXT, a tab delimited ASCII file.

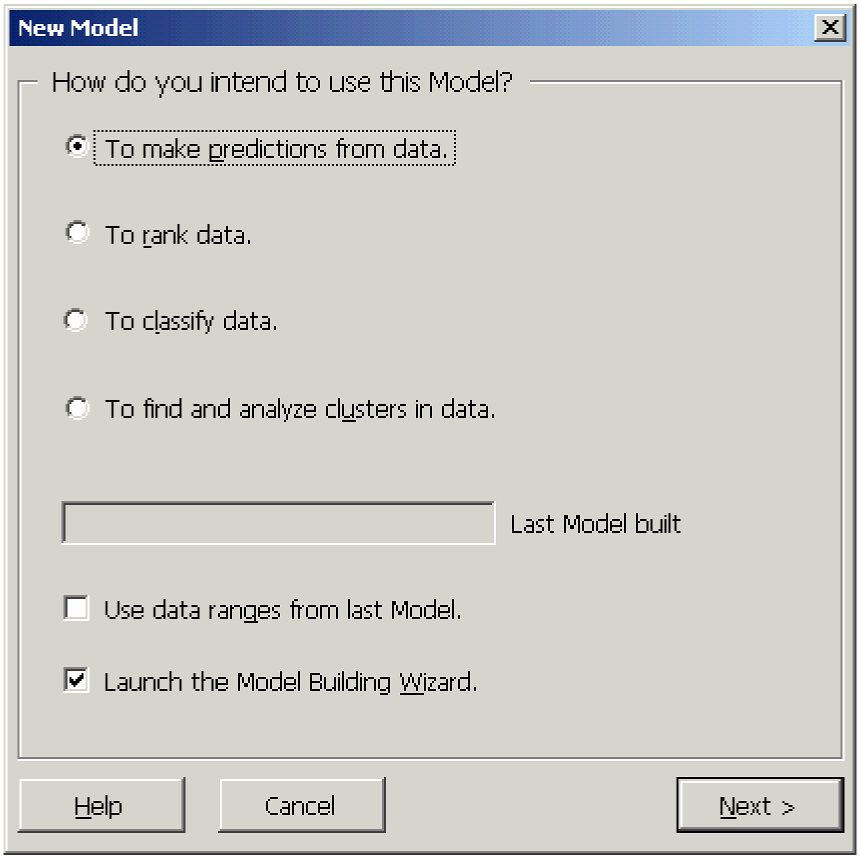
The data in the Worksheet is organized in 18 columns and 408 rows. Columns **A** through **Q** in Row **1** contain names ("NAME" through "MERCHANT") for each of the data fields. Columns **A** through **Q** in subsequent rows contain input data. Column **R** in Row **1** contains the name of the Target output field, and Column **R** in subsequent rows contains the Target output value for each row. Data in column **R** indicates the number of payments made on time by the individual during the preceding 12-month period.

3. Use the Excel **File | Save** As command (Alt F, A) to save the file as a normal (Microsoft Excel Workbook) file named CREDIT.XLS.

After the model is trained, *Predict* will write performance evaluation results into the Worksheet and we don't want the original ASCII text file CREDIT.TXT to be modified.

> Note: If you want to end the model building process before training the model, you can click the **Cancel** button in any dialog box displayed by the *Model Building Wizard*. When you click the **Cancel** button, the *Model Building Wizard* ends immediately; any information you may have entered in any dialog box will not be saved (though, as you will see, it is very quick and easy to start the process again).

4. Select the **New** command in the Predict menu (or, type **Alt-P**, then **N**). *Predict's* **New Model** dialog box, similar to the illustration below, appears.



New Model

How do you intend to use this Model?

● To make predictions from data.

○ To rank data.

○ To classify data.

○ To find and analyze clusters in data.

Last Model built

☐ Use data ranges from last Model.

☑ Launch the Model Building Wizard.

5. Since the model you are building is intended to predict whether an applicant is likely to repay a loan on time, click the **To make predictions from data** option to indicate that you want to build a prediction model. *De-select* the **Use data ranges from last Model** check box if it is selected, and *select* the **Launch the Model Building Wizard** check box. Then click the **Next>** button.

   Note: If the **Model Building Wizard** *Welcome* dialog box appears, click the **Next>** button to continue. If you wish, you can click the **Do not show this dialog box the next time the Wizard starts** check box to prevent the dialog box from appearing again.

6. When the **Building a Model (Step 1 of 6)** dialog box appears, you can enter information about names that will be used in the model.

   The default **Model Name** is the name of the active Worksheet – in this case, *Credit*. You can use *Credit* as the **Model Name**, or you can enter another name. If you want to enter a brief description of the model, click in the large text box labeled **Model Description** and enter the description. Then, click in the range text box labeled **Use Names in this Worksheet range** and either type **A1** in the box or click in Worksheet cell **A1**. This action will tell *Predict* to use the labels in row 1 as field names (note that you only need to specify one cell in the row). Finally, click the **Next>** button to proceed to Step 2 of 6.

   Note: When you click the **Next>** or **Finish** button on any dialog box in the *Model Building Wizard*, *Predict* checks all data entry fields in the dialog box to ensure that information you entered is valid. If *Predict* detects an error in any field, a message box will appear. Click **OK** to close the message box, then correct the error or errors (the *Model Building Wizard* dialog box will remain open). After you have corrected any errors, click the **Next>** button again to continue.

Also, note that every dialog box has a **<Back** button. If you want to review or change settings that you made in an earlier step, click the **<Back** button. You can use the **<Back** button to return all the way to the **New Model** dialog box.

7. When the **Building a Model (Step 2 of 6)** dialog box appears, you enter information about the location of *INPUT* data in the Worksheet.

   First, you specify the layout of input fields in the *first* data record. In the range text box labeled **INPUT cells in the first data record**, type **a2:q2**, or click in the box and then select cells **a2:q2** by dragging over them.

   Next, you specify the layout of input fields in the *second* data record. In the next range text box labeled **INPUT cells in the second data record**, click in the box and then click in cell **A3** (or click in the box and simply type a3). By providing ranges for two data records, you give *Predict* enough information to determine the relationship between data records. Although the data layout for this problem is very simple – one data record per row in the Worksheet, data records can overlap (other than the first row – the first and second data records cannot be identical).

   Finally, you specify the entire range of available input data. In the third range text box labeled **All INPUT data**, enter **a:a**; or, click in the box and then click the *heading* cell for column **A** (the cell that contains the letter **A**, *not* the top data cell in the Worksheet).

   After you supply these three elements, *Predict* can determine how many data records there are in the Worksheet. Click the **Next>** button to proceed to Step 3 of 6.

8. When the **Building a Model (Step 3 of 6)** dialog box appears, you enter information about the location of *OUTPUT* data in the Worksheet.

   Column **R** in the tutorial Worksheet indicates the number of times each applicant was late in making a payment during the preceding 12-month period, so column **R** contains the *Target Outputs* for the neural network (the values that the network will try to predict).

   In the range text box labeled **OUTPUT cell(s) in the FIRST data record**, type **r2**, or click in the box and then click in cell **R2** in the Worksheet. Although this model has only one Target Output, you can of course use *Predict* to create models with multiple Target Outputs.

   Click the **Next>** button to proceed to Step 4 of 6.

9. When the **Building a Model (Step 4 of 6)** dialog box appears, you enter information that generally describes the training data, and you specify the kinds of transformations that *Predict* should apply to the data.

   In the top frame of the dialog box, you indicate the amount of "noise" or variability that you think is present in the data. The amount of noise influences the internal actions *Predict* takes to avoid over-fitting the model to the training data. Moderately noisy data is the correct selection for many models. For more detailed information about the effects of noise settings, refer to **Network Parameters**.

   For this tutorial, confirm that **moderately noisy data** is selected.

   In the bottom frame of the dialog box, you indicate how thoroughly you want *Predict* to explore alternate transformations of raw data values in order to build a better model. Similar to the noise level setting, moderate data transformation is appropriate for a wide range of models. For more information about the effects of data transformation settings, refer to **The Data Analysis Table**.

   For this tutorial, confirm that **moderate data transformation** is selected.

   Click the **Next>** button to proceed to Step 5 of 6.

10. When the **Building a Model (Step 5 of 6)** dialog box appears, you enter information that governs how thoroughly *Predict* evaluates variables to determine their influence on model performance, and how thoroughly *Predict* should search for the best neural network. In both cases, the general trade-off is whether the (sometimes considerable) extra time needed to train the model when higher levels are selected yields sufficient improvements in model performance. There are no rigid rules, but it is generally best to start with lower settings to see how well a model that can be built quickly actually performs.

    In the top frame of the dialog box, you indicate how thoroughly *Predict* should search for the optimum set of model input variables. Note that the input variable pool that *Predict* will select variables from contains both the raw input data values from the Worksheet as well as all transformed values that *Predict* will generate based on the data transformation selection you made in Step 4. You can also specify whether *Predict* should use *Cascaded Variable Selection*. Refer to **Variable Selection Parameters** for more information about how *Predict* performs variable selection.

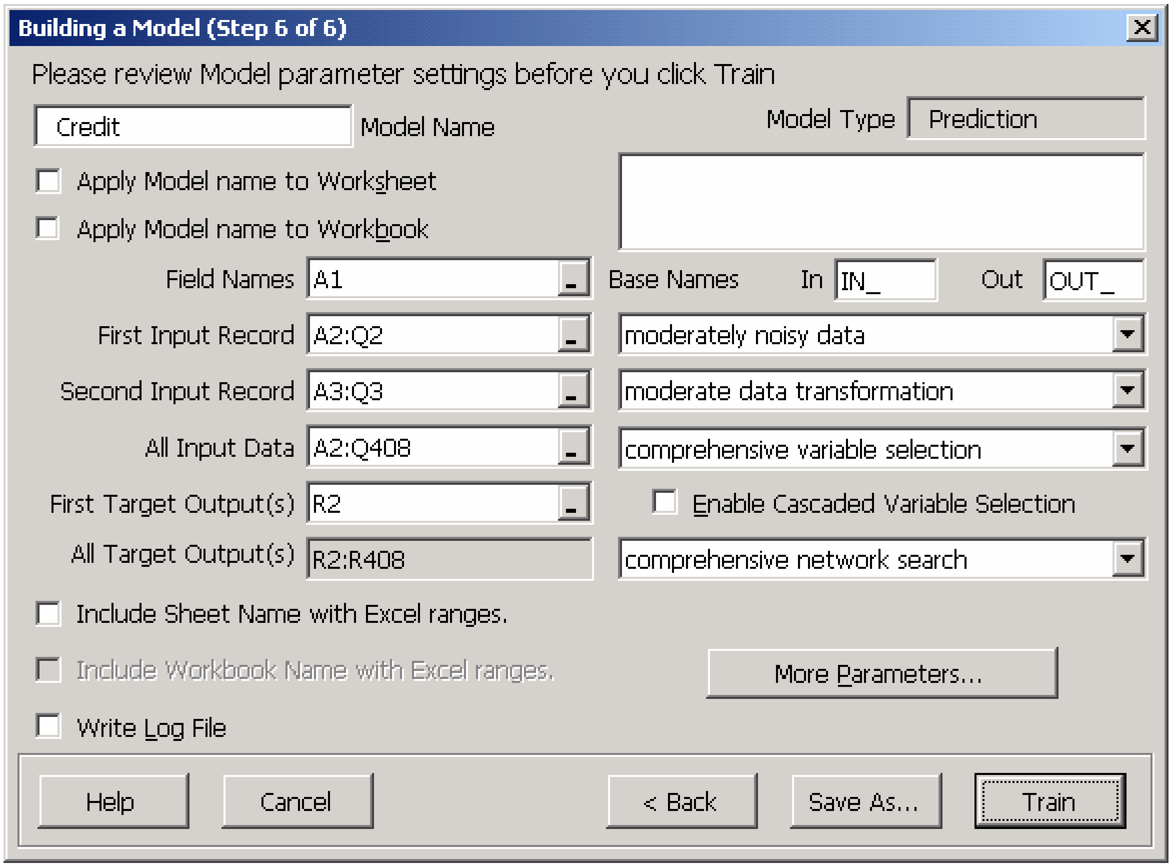
    For this tutorial, confirm that **comprehensive variable selection** is selected. Do *NOT* select **Enable Cascaded Variable Selection**.

    In the bottom frame of the dialog box, you indicate how thoroughly *Predict* should search for the best neural network. During training, *Predict* dynamically adds hidden units to the network and evaluates the performance improvement that results. As more hidden units are added, the time needed to train the final network increases. For more information about how *Predict* dynamically constructs the neural network, refer to **Network Parameters**.

    For this tutorial, confirm that **comprehensive network search** is selected.

    Click the **Next>** button to proceed to Step 6, the final step of the model building process when the *Model Building Wizard* is active.

11. When the **Building a Model (Step 6 of 6)** dialog box as illustrated below appears, you can review, and if necessary modify, all the data range and model parameter settings that you specified (except that you cannot change the model type). You then start the process of actually constructing and training the model.



Building a Model (Step 6 of 6)

Please review Model parameter settings before you click Train

Model Name: Credit

Model Type: Prediction

□ Apply Model name to Worksheet

□ Apply Model name to Workbook

Field Names: A1

Base Names: In IN_ Out OUT_

First Input Record: A2:Q2

Second Input Record: A3:Q3

All Input Data: A2:Q408

First Target Output(s): R2

All Target Output(s): R2:R408

moderately noisy data

moderate data transformation

comprehensive variable selection

□ Enable Cascaded Variable Selection

comprehensive network search

□ Include Sheet Name with Excel ranges.

□ Include Workbook Name with Excel ranges.

□ Write Log File

If you wish, you can click the **More Parameters...** button to open the **Model Parameters** dialog box and view a set of tabbed pages that provide access to all of *Predict's* parameters; however please do *NOT* make any changes during this tutorial. Changes made in the **Model Parameters** dialog box potentially override settings that are made as a result of actions by the *Model Building Wizard*.

> **Note:** If you do click the **More Parameters...** button, a **Save Model** dialog box will appear. Simply click the **Save** button to save the model using the default name credit, and you will then be able to view Model Parameters pages.

12. Confirm that your **Building a Model (Step 6 of 6)** dialog box appears as shown above, then click the **Train** button. After allowing you to name and save the model *Predict* will start constructing the neural network and training the model.

> Note: You do not have to click the **Save As…** button to save the model – after you click the **Train** button a **Save Model** dialog box will appear. When the **Save Model** dialog box opens, either accept the default name of Credit, or enter the name you would like to use for the model (for the remainder of this tutorial we will assume that you accepted the default name). *Predict* will automatically append an `npr` (for NeuralWorks PRedict) extension to the model name you select. In effect, a *Predict* `npr` file is a "document" file for *Predict*, just as an `xls` file is a "document" file for Excel. The *Predict* `npr` file contains all model parameter and range settings for a particular model.
>
> It is best to save your models into C:\My Documents or another user directory. Change the directory to **C:\My Documents**. When you click the **Save** button in the **Save Model** dialog

box, *Predict* will begin the process of actually constructing the neural network and training the model.

For more detailed information about the 6 dialog boxes that you use to specify model ranges and parameters when the *Model Building Wizard* is active, refer to **Using the Wizard**.

Constructing the Neural Network and Training the Model

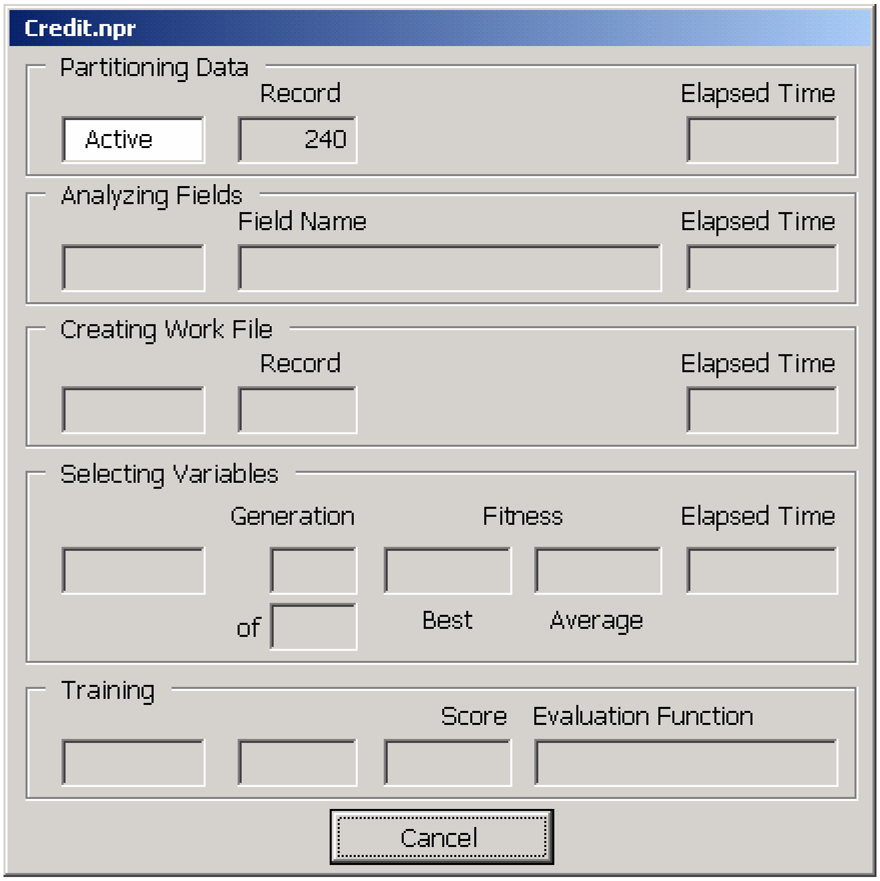
Predict automatically performs five basic operations as it constructs the neural network and trains the model:

- Partitioning the data into Train, Test, and Validation sets
- Analyzing data fields to identify the type of data
- Creating a work file that contains transformed data
- Selecting Input Variables
- Training the neural network

At the beginning of each operation a short status message is displayed in the Excel status bar at the bottom of the main Excel window. In addition, if the **Display Progress Information while building models** check box on the ***General*** tab in the **Preferences** dialog box is selected, when the model building process begins a status dialog box is displayed. While each operation is underway, its status is set to active and the status dialog box is updated with more detailed information. When an operation ends, its status is set to Complete, and the duration of the operation is displayed in the Elapsed Time field for the operation.

Partitioning Data

During the partitioning of data into train, test, and validation sets, the **Partitioning Data** frame shows the number of the training data record being processed.



Credit.npr

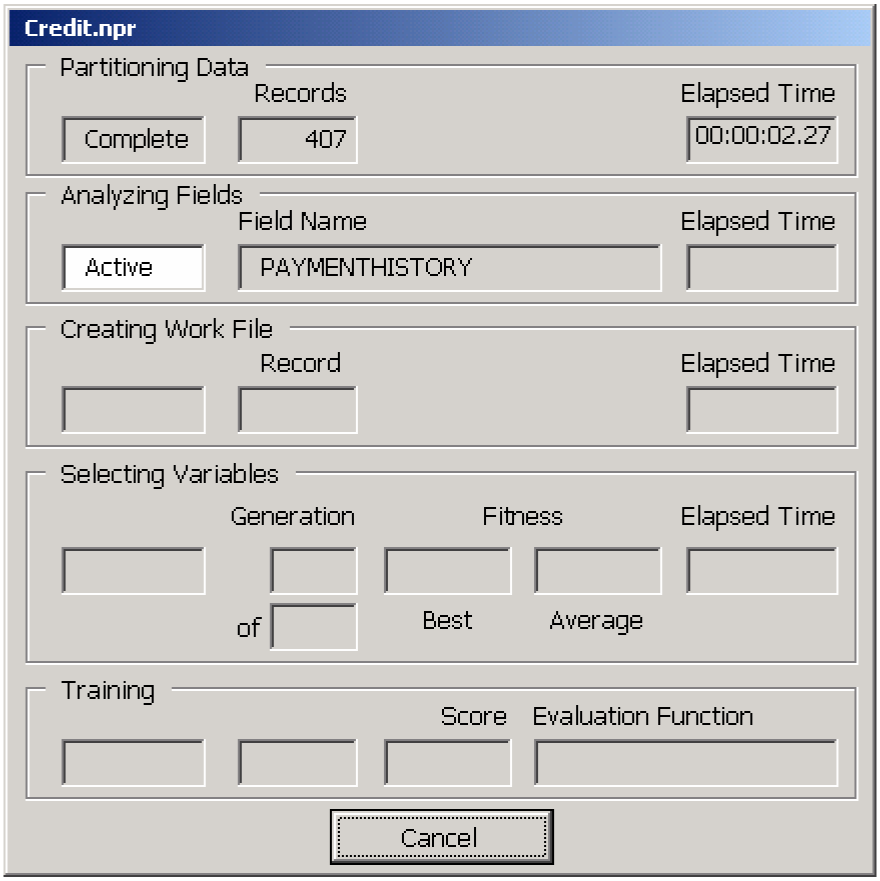
Partitioning Data

Active

Record: 240

Analyzing Fields

During the data analysis operation, the **Analyzing Fields** frame shows the name of the field being processed.



Credit.npr

Partitioning Data

Complete

Records: 407

Elapsed Time: 00:00:02.27

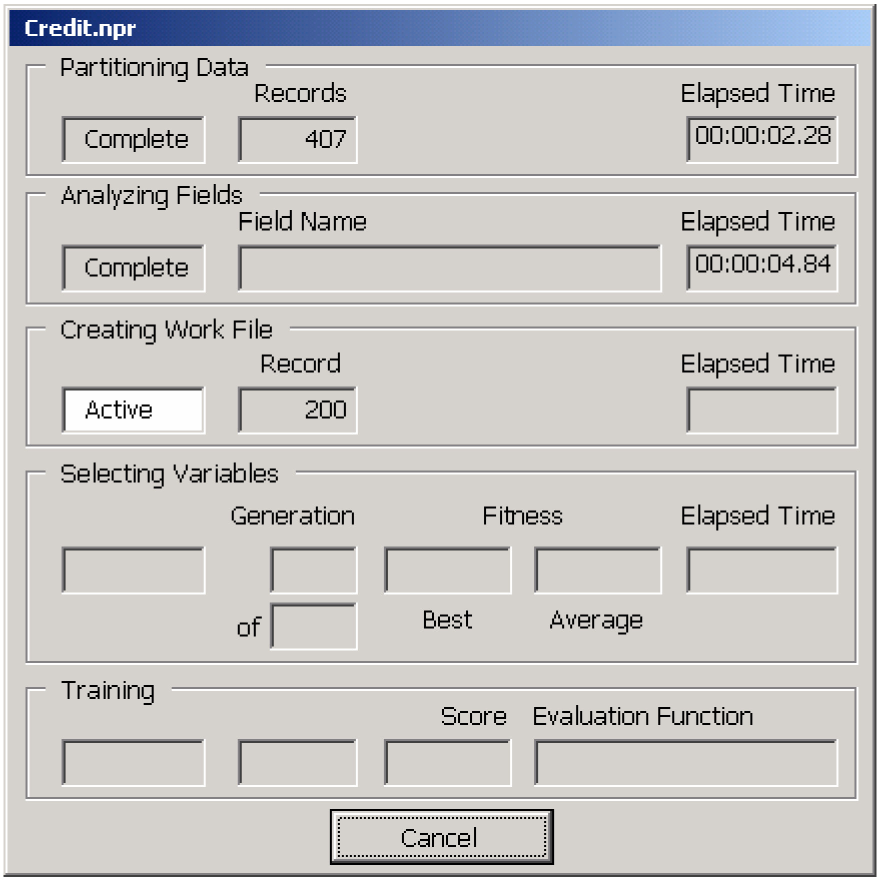
Analyzing Fields

Active

Field Name: PAYMENTHISTORY

Creating the Work File

While *Predict* is creating a binary work file that contains transformed data values, the **Creating Work File** frame shows the record being processed.



Credit.npr

Partitioning Data: Complete; Records: 407; Elapsed Time: 00:00:02.28

Analyzing Fields: Complete; Field Name: ; Elapsed Time: 00:00:04.84

Creating Work File: Active; Record: 200; Elapsed Time:

Selecting Variables: Generation: of; Fitness: Best, Average; Elapsed Time:

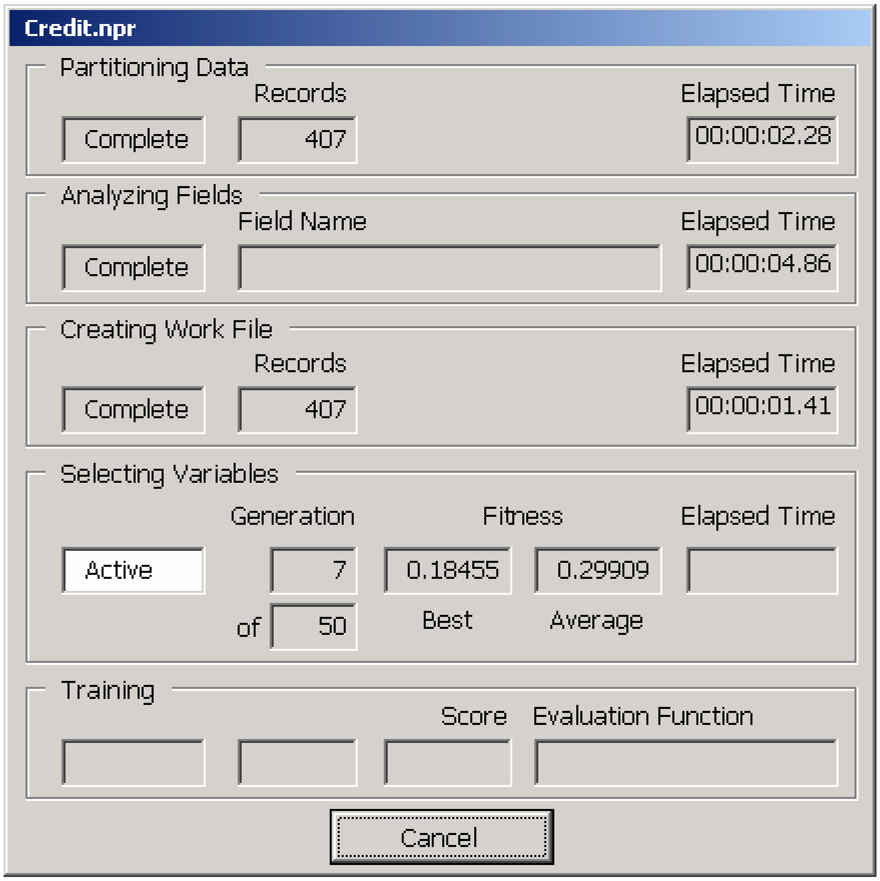
Training: Score; Evaluation Function

Selecting Variables

Predict employs a **Genetic Algorithm** (**GA**) to identify the best set of input variables for the model. A **GA** operates on a population of individuals. The population changes from one generation to the next, usually by combining characteristics of two "parent" individuals to create a "child" individual. Every individual is assigned a fitness and the concept of "survival of the fittest" is implemented by selecting the most fit parents more frequently than less fit parents to create the next generation.

In *Predict's* Genetic Algorithm for input variable selection, an "individual" is actually a set of input variables. The fitness of an individual is derived from the performance of a model that uses the individual's variable set as inputs. The algorithm begins with individuals that consist of small sets of variables. Individuals that produce good models are kept in the population and used to generate individuals that consist of larger sets of variables if necessary. In general, however, smaller variable sets (that is, fewer model inputs) are preferred to larger variable sets.

While the Variable Selection operation is active, the **Selecting Variables** frame shows the current **GA** generation, the maximum number of generations, the current best fitness, and the average fitness. When Variable Selection ends, the index of the most fit individual remains visible.



Credit.npr

Partitioning Data

Complete

Records: 407

Elapsed Time: 00:00:02.28

Analyzing Fields

Complete

Field Name:

Elapsed Time: 00:00:04.86

Creating Work File

Complete

Records: 407

Elapsed Time: 00:00:01.41

Selecting Variables

Active

Generation: 7 of 50

Fitness: 0.18455 Best, 0.29909 Average

Elapsed Time:

Training

Score:

Evaluation Function:

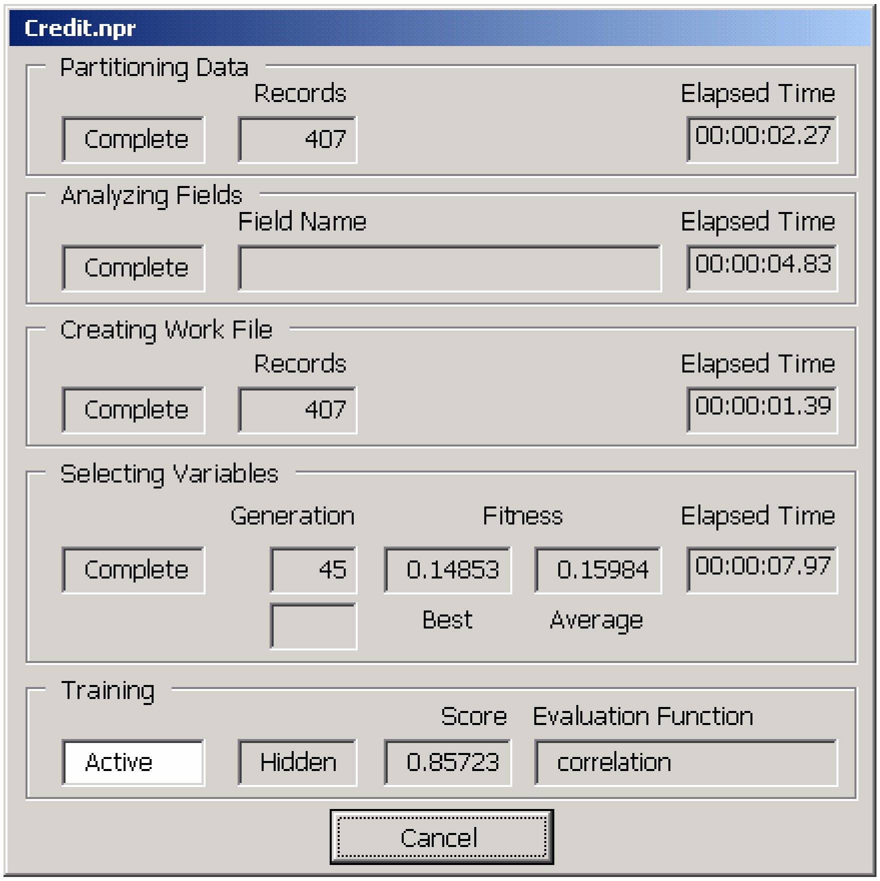
The **Enable Cascaded Variable Selection** option, which was not used in this tutorial, causes multiple executions of the standard variable selection algorithm, performed with different initialization values and with more relaxed convergence criteria, in a "pre-variable selection" step. Variables which consistently are not included in the final population of each run are eventually omitted from later runs. The end result of this process is a smaller set of variables which have all demonstrated some potential to contribute to the model. The cascaded variable selection step is then followed by the standard variable selection algorithm to further refine the variable set.

> **Note:** Enabling cascaded variable selection will markedly increase the time it takes to complete the variable selection phase.

Training the Neural Network

Predict constructs the actual neural network incrementally, using a technique known as cascade correlation. Hidden units are periodically added, usually one or two at a time. Each time a hidden unit or pair of hidden units is added, weights are trained from several different initialization values. Each initialization is referred to as a *candidate*. The best candidate is established in the network, and then all the weights to the output node(s) of the network are retrained.

While hidden unit candidates and output weights are being trained, information is displayed in a dialog box similar to the following:



Credit.npr

Partitioning Data
Complete | Records: 407 | Elapsed Time: 00:00:02.27

Analyzing Fields
Complete | Field Name: | Elapsed Time: 00:00:04.83

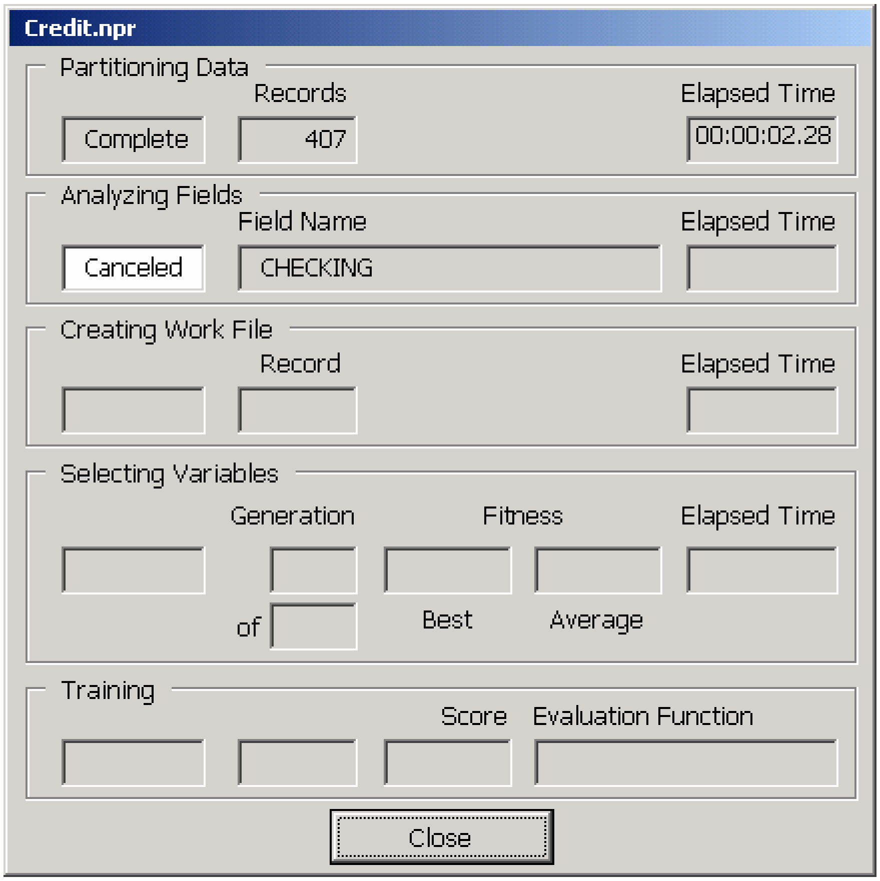
Creating Work File
Complete | Records: 407 | Elapsed Time: 00:00:01.39

Selecting Variables
Complete | Generation: 45 | Fitness: Best 0.14853, Average 0.15984 | Elapsed Time: 00:00:07.97

Training
Active | Hidden | Score: 0.85723 | Evaluation Function: correlation

At any time during the model building process, you can press the **Escape** key or click the **Cancel** button to stop the process. If you stop the process, the status of the last active operation is changed to *Canceled*, and the **Cancel** button caption changes to **Close**, as illustrated below. Click the **Close** button to close the **Build** dialog box.

Note: The partially trained model itself is *not* closed, to permit you to open the **Change the Model** dialog box and modify model parameters.



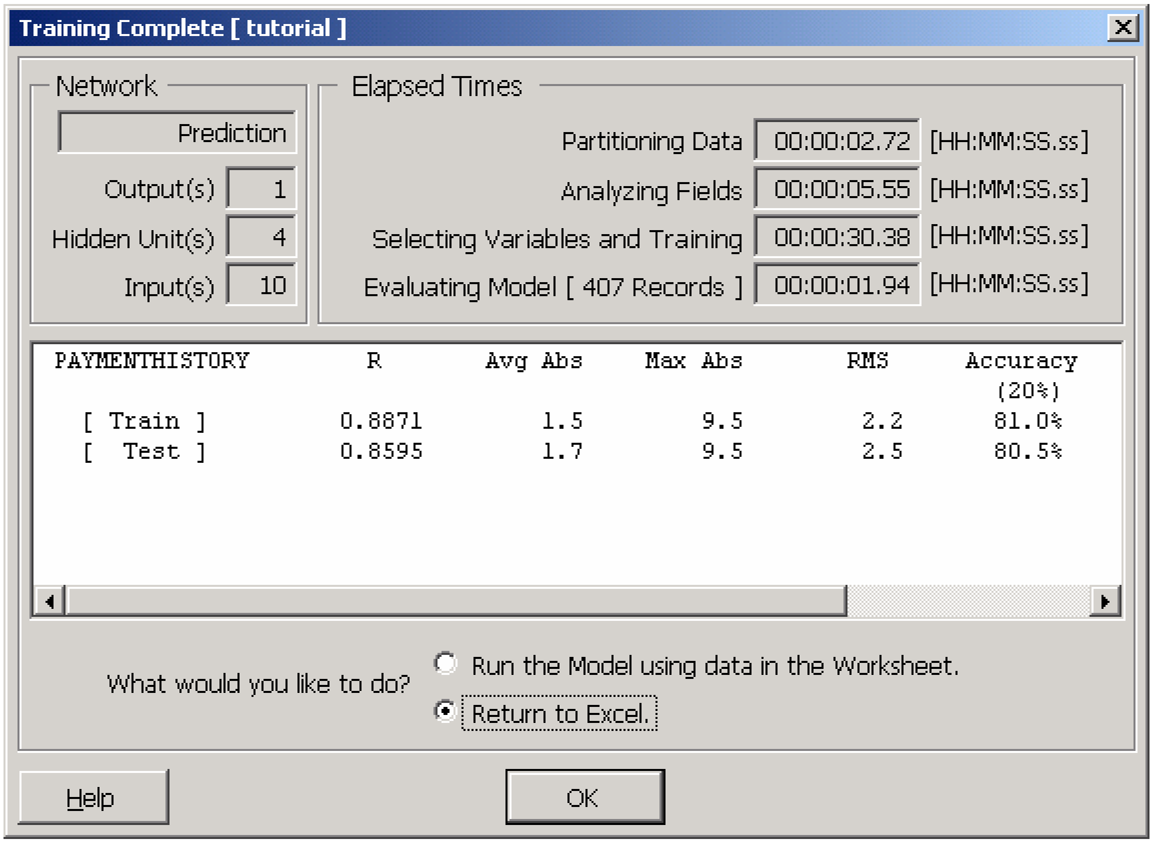
Credit.npr

Partitioning Data: Complete; Records: 407; Elapsed Time: 00:00:02.28

Analyzing Fields: Canceled; Field Name: CHECKING

If you do not cancel the build process, and if the build process completes without error, the **Build** dialog box closes automatically and *Predict* immediately evaluates the performance of the model on the Train and Test data sets.

When the evaluation ends, a **Training Complete** dialog box similar to the following appears.



Training Complete [tutorial]

Network

Prediction

Output(s)	1
Hidden Unit(s)	4
Input(s)	10

Elapsed Times

Partitioning Data	00:00:02.72	[HH:MM:SS.ss]
Analyzing Fields	00:00:05.55	[HH:MM:SS.ss]
Selecting Variables and Training	00:00:30.38	[HH:MM:SS.ss]
Evaluating Model [407 Records]	00:00:01.94	[HH:MM:SS.ss]

```
PAYMENTHISTORY          R       Avg Abs    Max Abs       RMS    Accuracy
                                                                  (20%)
  [ Train ]          0.8871        1.5        9.5       2.2     81.0%
  [  Test ]          0.8595        1.7        9.5       2.5     80.5%
```

What would you like to do?

○ Run the Model using data in the Worksheet.

● Return to Excel.

The **Training Complete** dialog presents a summary of network attributes and performance statistics for the model when it is run with the *Train* and *Test* sets. The specific content of the **Training Complete** dialog box depends on the type of model; however, for the prediction model built in this tutorial, the following information is displayed.

Network

The *Network* frame contains the name of the model, and the number of processing units in each layer. Note that the number of *model* inputs often differs from the number of input *fields* in the training dataset, and model inputs may not correspond directly to training record input fields, due to *Predict's* data transformation process. More information about model input processing units is provided in the next section.

Elapsed Time

The *Elapsed Time* frame indicates the time that elapsed in each of the major model building and testing phases, and the number of records that were processed.

Performance Statistics

Basic model performance statistics with respect to the Train and Test data sets are shown in the main pane in the dialog box. The Target Output field name is shown in the upper left corner. The following performance statistics are automatically computed for prediction models:

R [R Correlation] The linear correlation between predicted outputs and target outputs, in problem domain units.

Avg Abs [Average Absolute Error] The average absolute difference between predicted output values and target output values.

Max Abs [Maximum Absolute Error] The maximum absolute difference between a predicted output value and a target output value.

RMS [Root Mean Square Error] The root mean square error between the predicted outputs and the target outputs.

Accuracy The percent of predicted output values that lie within 20% of their corresponding target output values.

Conf Interval [Confidence Intervals] 95% of the model predictions lie within the range around target output values bounded by the confidence intervals.

Records [Number of Records processed] the number of records processed during training/testing.

You can use the scroll bar at the bottom of the main pane to view the two fields (*Confidence Intervals* and the *Number of Records* in each set) that are hidden on the right side.

13. When you have finished reviewing model performance statistics, confirm that the **Return to Excel** option is selected, as shown above, and click the **OK** button.

After a model is trained successfully, *Predict* automatically saves it. In addition, while the model is training *Predict* periodically generates a checkpoint file with a .nck extension. You can use the checkpoint file to partially recover your work in the event a power failure or other malfunction interrupts the training process.

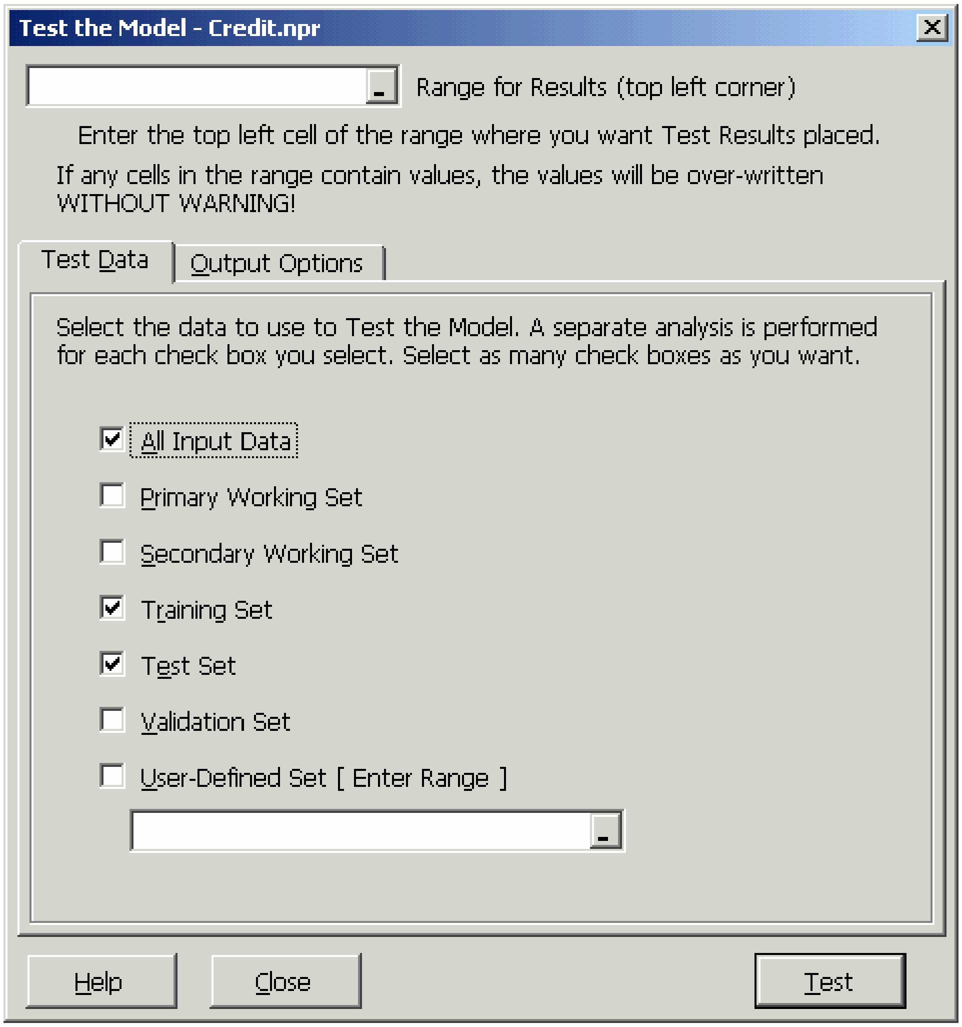
You are now ready to further evaluate and then run the model you have just built. The next sections will provide more detailed information about evaluating *Predict* models and then running models with new data.

Evaluating the Model

The quality and utility of your model depends ultimately on whether you can achieve incremental improvements in comparison with alternate models. *Predict's* **Test** facility provides some general evaluation measures which you can use for model comparisons. You can also construct more specific evaluation criteria using Excel functions.

Evaluating the model using the Test Facility

1. Select the **Test** command from the Predict menu (Alt P, T). The **Test the Model** dialog box appears.



Test the Model - Credit.npr

Range for Results (top left corner)

Enter the top left cell of the range where you want Test Results placed.

If any cells in the range contain values, the values will be over-written WITHOUT WARNING!

Select the data to use to Test the Model. A separate analysis is performed for each check box you select. Select as many check boxes as you want.

☑ All Input Data

□ Primary Working Set

□ Secondary Working Set

☑ Training Set

☑ Test Set

□ Validation Set

□ User-Defined Set [Enter Range]

2. In the **Range for Results** text box, type the range **a415** (equivalently **R415C1**), or click that cell in the worksheet.
3. On the ***Test Data*** tab, select **All Input Data**, **Training Set**, and **Test Set** as shown above.
4. Click **Test** to start the test.
5. Click **Close** to close the dialog box.

Predict makes one pass through the data and then writes a table whose top left cell is **A415** into the worksheet (scroll to cell **A415** so that you can see the table):

PAYMENT HISTORY	R	Net-R	Avg. Abs.	Max. Abs.	RMS	Accuracy (20%)	Conf. Interval (95%)	Records
All	0.8786	0.8786	1.558	9.534	2.309	0.808	4.505	407
Train	0.8871	0.8871	1.514	9.459	2.230	0.810	4.357	284
Test	0.8595	0.8595	1.660	9.534	2.482	0.805	4.883	123

All the measures in this table are calculated with respect to real world target outputs. These are the outputs that are contained in the training data set. When training the model, the real world target outputs are transformed to internal target outputs for training the neural net. The raw predictions which are generated by the neural net are referred to as the neural net output. These neural net outputs are transformed to model outputs in real world units by putting them through the inverse of the transform that was used to map real world targets to neural net targets.

One of the signs of a good model is that training set performance and test set performance are fairly similar. It is always possible to get good performance on a training set, but the important thing is to have it perform well on new data.

R is the linear correlation between the real world target output and the real world prediction. Perfectly correlated outputs have an *R* value of 1.0. Anti-correlated outputs have an *R* value of -1.0. Uncorrelated outputs have an *R* value of 0.0. What constitutes a good value for R is very dependent on the problem domain. For some very noisy domains such as financial markets a value of 0.15 or 0.2 might be considered good. The real test of the effectiveness of a model can only really be gauged by comparing it with other models on previously unseen data.

Net-R is the linear correlation between the real world target output and the neural net (internally transformed) predicted output.

Avg. Abs. is the average absolute error between the real world target output and the real world prediction.

RMS is the root mean square error between the real world target output and the real world prediction.

Accuracy is the fraction of times the real world target is "close" to the real world prediction, where, for this test, "close" is defined to be 20% of the output range.

Confidence Interval corresponds to an error bar around the output. For this test a 95% confidence level is used. So for example, looking at the table above, we have 95% confidence that the model (predicted) output will be within 4.883094 of the target output value assuming the test set is representative of the data population.

Evaluating the Model using Excel Functions

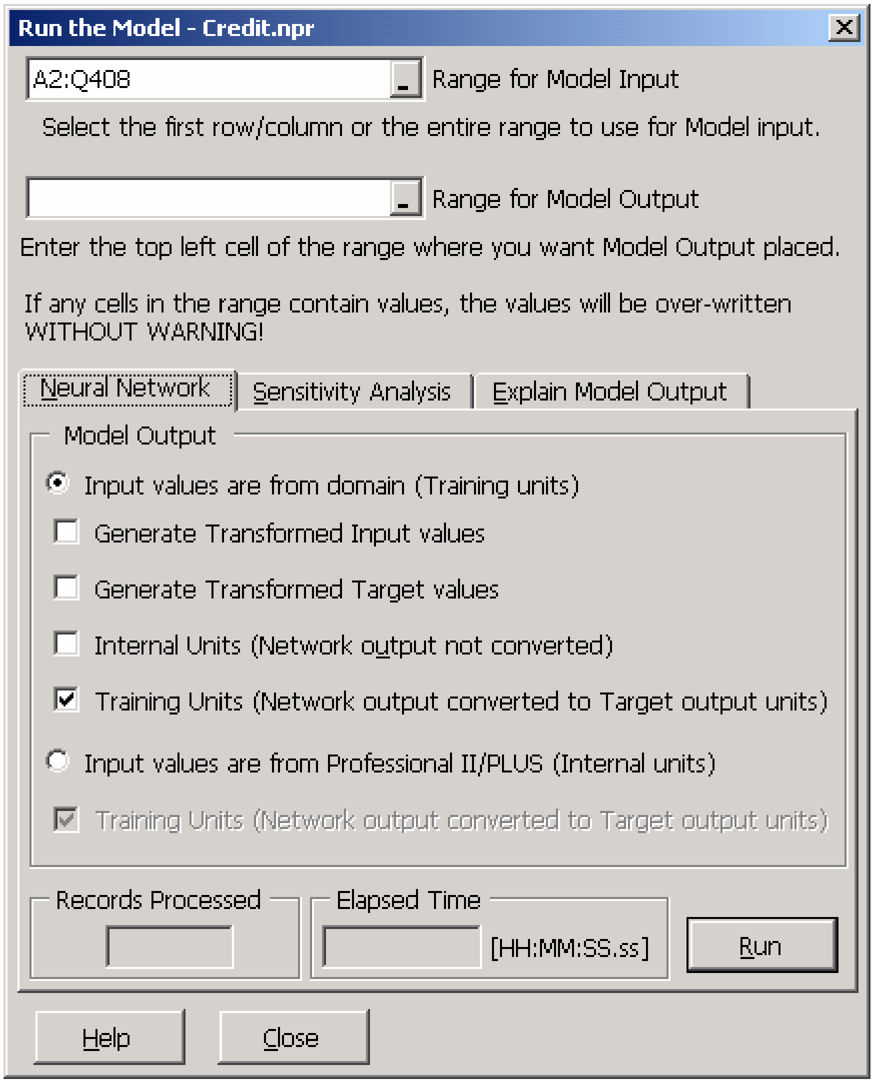
1. Save the model using the **Save** command in the Predict menu (Alt P, S). The demonstration version will not let you save (demo users please skip this step).

Saving the model is not strictly necessary at this point because you saved the model after training ended. However, if you save the model now, the selections you made for the **Test** command will be preserved.

2. Select the **Run** command from the Predict menu (Alt P, R).

The **Run** command generates network predictions for a range data. We will use it to generate predicted values for all the training data.

3. When the *Neural Network* tab appears as illustrated below, if **A2:A408** is not in the **Range for Model Input** box, click in the box and enter **a2:a408** (equivalently **R2C1:R408C1**) or select these cells with the mouse.



Run the Model - Credit.npr

A2:Q408 Range for Model Input

Select the first row/column or the entire range to use for Model input.

Range for Model Output

Enter the top left cell of the range where you want Model Output placed.

If any cells in the range contain values, the values will be over-written WITHOUT WARNING!

Model Output

● Input values are from domain (Training units)

☐ Generate Transformed Input values

☐ Generate Transformed Target values

☐ Internal Units (Network output not converted)

☑ Training Units (Network output converted to Target output units)

○ Input values are from Professional II/PLUS (Internal units)

☑ Training Units (Network output converted to Target output units)

Records Processed

Elapsed Time [HH:MM:SS.ss]

4. Click in the **Range for Model Output** box, then enter **t2** or select cell **T2** (equivalently **R2C20**). Also, confirm that the **Input values are from domain (Training units)** option is selected, and that the **Training Units (Network output converted to Target output units)** check box is checked, as show above.

5. Click the **Run** button. *Predict* fills column **T** with a predicted output for each record. At the bottom of the **Neural Network** page, *Predict* displays the number of records processed and the time that elapsed during processing.

6. Click the **Close** button to close the **Run the Model** dialog box. For ease in reviewing results, enter the label "Predictions" in cell **T1** (equivalently **R1C20**).

With model output values available in the worksheet, you can use any Excel function or chart to evaluate the model by comparing the predicted output with the target output. For example, the Pearson R

Correlation is produced by entering the formula "=PEARSON(R2:R408,T2:T408)" in an empty cell. You can compare this measure, or other Excel statistical measures, to the output values produced by other programs, products, and techniques.

Note: In **Tutorial 3** you will see how to determine which subsets of the data range *Predict* chose for Train and Test sets. You can then use this information to restrict the domain of an Excel function to just the test set or just the training set.

Now is a good time to save your spreadsheet using the **Excel File | Save** command (Alt F, S), saving it as *Credit.xls*.

Viewing the Status of Input Variables

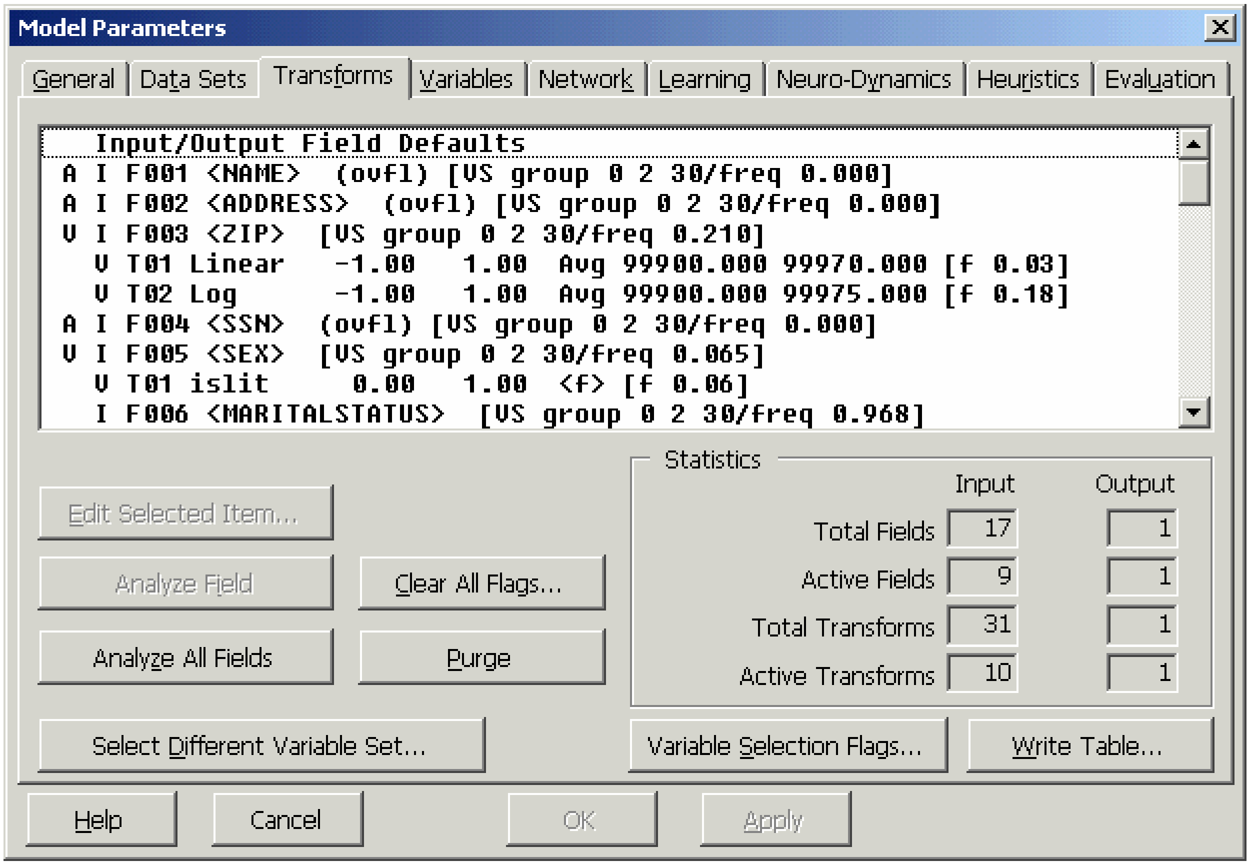
When the problem space offers a large number of input variables, often there are multiple subsets of variables which will yield good models. Since a Genetic Algorithm produces different results depending on the initial population characteristics, if you build two different models using the same data *Predict* will usually select a different set of input variables for each model. Intuitively, you can consider each model as an "expert" that uses different criteria for making decisions. In Excel you can easily combine the outputs of different *Predict* models to reflect the combined "wisdom" of each of the experts.

→ Demonstration version users: please skip all remaining steps in **Tutorial 1**. You can look at results of *Predict's* data transformation and variable selection processes (you can perform steps 2, 3, 6 and 7), but you cannot record these results into your spreadsheet. This information is worth reading through!

To view the variables that *Predict* selected as input for the current model:

1. First, copy the input field names for the credit problem (cells **A1:Q1**) and paste them into cells **A410:Q410** (equivalently **R410C1:R410C17**). This will help you interpret the results of *Predict's* variable selection process.
2. From the Predict menu, select the **Change** command (Alt P, C). The **Change** command opens the **Change the Model** dialog box.

3. When the **Change the Model** dialog box appears, click the **More Parameters** button to open the **Model Parameters** dialog box. When the **Model Parameters** dialog box appears, click the ***Transforms*** tab to view the Transforms page as illustrated below.

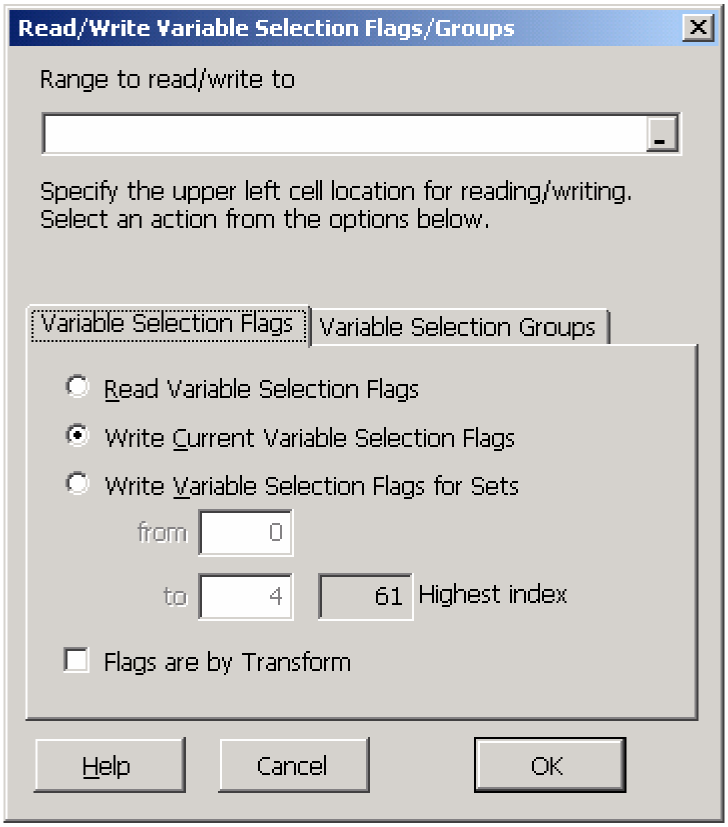


```
   Input/Output Field Defaults
 A I F001 <NAME>  (ovfl) [VS group 0 2 30/freq 0.000]
 A I F002 <ADDRESS>  (ovfl) [VS group 0 2 30/freq 0.000]
 V I F003 <ZIP>  [VS group 0 2 30/freq 0.210]
   V T01 Linear   -1.00   1.00  Avg 99900.000 99970.000 [f 0.03]
   V T02 Log      -1.00   1.00  Avg 99900.000 99975.000 [f 0.18]
 A I F004 <SSN>  (ovfl) [VS group 0 2 30/freq 0.000]
 V I F005 <SEX>  [VS group 0 2 30/freq 0.065]
   V T01 islit     0.00   1.00  <f> [f 0.06]
   I F006 <MARITALSTATUS>  [VS group 0 2 30/freq 0.968]
```

Statistics

	Input	Output
Total Fields	17	1
Active Fields	9	1
Total Transforms	31	1
Active Transforms	10	1

4. Demo users cannot perform this step. Click the **Variable Selection Flags** button (Alt S). The **Read/Write Variable Selection Flags/Groups** dialog box, as illustrated below, appears.



Read/Write Variable Selection Flags/Groups

Range to read/write to

Specify the upper left cell location for reading/writing.
Select an action from the options below.

Variable Selection Flags | Variable Selection Groups

○ Read Variable Selection Flags

● Write Current Variable Selection Flags

○ Write Variable Selection Flags for Sets

from 0

to 4 61 Highest index

□ Flags are by Transform

5. Demo users cannot perform this step. With the **Variable Selection Flags** page visible, click in the box labeled **Range to read/write to**, and either type **A411**, (equivalently **R411C1**) or click cell **A411**. Confirm that the **Write Current Variable Selection Flags** option is selected, then click **OK**. *Predict* places a set of single character strings ("flags") into cells **A411:Q411**. Click **Close** to return to **Model Parameters**.

6. Click **Cancel** to close the **Model Parameters** dialog box.

7. Click **Close** to close the **Change the Model** dialog box.

Scroll the Worksheet so that you can see rows **410** and **411**. Row **410** should contain the field names that you copied in step 1. Row **411** contains the data analysis flag that corresponds to the field name above it. The characters *Predict* uses as flags, and their meaning, are as follows:

I The field was selected as an input to the model.

A The field was rejected during data analysis by *Predict*. This can mean that the field had too much missing data, the field is a constant value, or the field has some other characteristic that makes it fundamentally inappropriate for building a model.

V The field was rejected during the variable selection process. This means that the field could potentially have been used in the model (and may in fact be used in a different model) but was not part of the synergistic set of input variables chosen for this particular model.

U The field was explicitly rejected by the user. Assuming you have been following the instructions in this tutorial, there should be no cells that contain a "U". In **Tutorial 2** you will see how you can explicitly disable a field to prevent it from being used as a model input.

C The field was rejected by the cascaded variable selection algorithm. Since this option was not selected when you specified model parameters, you should not see any 'C's.

Note: This exercise shows how to identify variables that were included or excluded at the field level. In **Tutorial 2** you will take a closer look at the results of Variable Selection and identify actions taken by *Predict* at the transform level.

Running the Model

There are several ways you can use a *Predict* model on the computer which has *Predict* installed. One way is the Run command, which you have already used. A second way is through the Excel macro function interface which provides access to a function named **PredictRun**. You can use **PredictRun** to process new data vectors in a manner similar to the way you used the Run command earlier in this tutorial. However, there are important differences between the Run command and the PredictRun macro. The differences are:

- **PredictRun** is a function that can be directly incorporated into an Excel worksheet or macro formula.
- PredictRun does not require that *Predict* be running. With a valid deployment license, PredictRun can be distributed in a Worksheet to other users who do not have the full product installed, and they may have no interest in knowing about neural networks.
- PredictRun is a live, dynamic function. Changes made to input cell values immediately cause a new predicted value to appear in the PredictRun cell.

Also note that the **PredictRun** macro function is only available when the full *Predict* product is installed and the *Predict* License Key has been validated. If you have not yet activated your software license, you will not be able to do the following exercise. You can always return to this place in the Tutorial after you receive and install the License Key for your software. Demonstration version users cannot perform these steps.

1. Click in cell **U2** (equivalently **R2C21**) with the mouse.
2. From the **Insert** menu in the *Excel* menu bar, select **Function** command. Excel displays a list of function categories; one of which is *User Defined Functions*.
3. Select the *User Defined* category.
4. Select the function **PredictRun**.
5. Select *OK* or *Finish* to insert the function into the Excel formula bar and to open the parameters dialog box.
6. In the **Model** text box, enter the full path for the model. If you have followed this tutorial exactly, the path is:

   ```
   "C:\My Documents\credit.npr", or possibly
   "C:\Documents and Settings\(you)\My Documents\credit.npr"
   ```

   Be sure you enclose the entire path in quotation marks.
7. In the *InputRange* text box enter the range of the first input vector (**A2:Q2** or equivalently **RC[-20]:RC[-4]**). You can also use the mouse to select this range.

The function in the Excel formula bar should now appear as:

```
=PredictRun("C:\My Documents\credit.npr", A2:Q2)
```

or

```
=PredictRun("C:\My Documents\credit.npr ", RC[-20]:RC[-4])
```

depending on the Excel Reference Style.

8. Click OK to enter the formula into the cell.

The numeric value that appears is *Predict's* model output that corresponds to the real world (observed) output for the same input data. Of course, the output generated by the **PredictRun** macro function should be identical to the output value *Predict* generated when you executed the **Run** command from the Predict menu earlier in this tutorial.

If an Excel error such as ***#REF?*** appears in the cell, please review the above steps carefully. You may have made an error typing in the model name or the input data range. If the Excel error is ***#Null!***, you have not activated your software license. To obtain more specific error information, enter the formula ***=PredictErr()*** in an empty Worksheet cell. *Predict* will display a message that describes the last error that occurred.

Note that NeuralWare offers other options for deploying *Predict* models, including source code generation (see the Predict FlashCode section in the **Using NeuralWorks Predict Models** chapter), the Predict Visual Basic Run-Time Kit, or interfacing with the Predict engine using the Predict C/C++ Software Development Kit. Please contact sales@neuralware.com to discuss your deployment requirements.

Closing the Model

The following steps show how to close your *Predict* model. This releases any memory used by the model. Note that even when the model is closed in Excel, you can still use the PREDICTRUN function.

Demonstration version users should skip these two steps and proceed to **Tutorial 2**.

1. To close the model, select the **Close** command in the Predict menu (Alt P, C). Answer *Yes* if you are prompted to save the model.
2. Close the worksheet using the **Excel File | Close** command (Alt F, C), saving it (as CREDIT.XLS) if prompted.

Congratulations! You have now built and evaluated a *Predict* model. You have also gained some understanding of what makes a good set of input variables, and you have seen how you can easily deploy *Predict* models in actual applications.

Credit Risk Tutorial 2

This tutorial covers:

- The data analysis and transformation process
- Modifying settings that influence data analysis
- Disabling a field so it will not be considered in the analysis
- Rebuilding a model
- Enabling a field disabled by variable selection

Understanding Data Analysis

1. If Excel, *Credit.xls* and your *Credit.npr* model are still open from **Tutorial 1**, then open the **Change the Model** dialog box (click **Change...** in the Predict menu).

   Otherwise, start Excel and use **File | Open** to open the data file *Credit.xls* from **Tutorial 1**. Then open the *Predict* model: *Credit.npr* is visible in the Predict menu as one of the most recently used files unless you are running the demonstration version of *Predict*, or have built several of your own models in the meantime. If *Credit.npr* is not visible in the Predict menu you can use the **Predict | Open** command to open your *Credit.npr* model, or open our distributed file *credit2.npr*. Your saved model is probably in `C:\My Documents`; our model is in our product install directory, probably `C:\Program Files\NeuralWare\NeuralWorks\Predict`.

2. When the *Change the Model* dialog appears, you should see all the ranges and default settings you specified while building the model in **Tutorial 1**.

   Note: When the *Model Building Wizard* is not active, this dialog box displays as soon as you specify the model type and click the Next button on the **New Model** dialog box. As you can see, all settings that are specified when the *Model Building Wizard* is active are combined on this one page.

3. Click the **More Parameters...** button in the **Change the Model** dialog box to open the **Model Parameters** dialog box. Then click the ***Transforms*** tab.

The large scrolling pane in the top half of the dialog box contains the results of *Predict’s* data analysis and variable selection processing. We refer to this area as the **Data Analysis Table**; representative contents for the Credit.npr model are reproduced below.

The Data Analysis Table

```
Input/Output Field Defaults
A I F001 <NAME>  (ovfl) [VS group 0 2 30/freq 0.000]
A I F002 <ADDRESS>  (ovfl) [VS group 0 2 30/freq 0.000]
V I F003 <ZIP>  [VS group 0 2 30/freq 0.210]
  V T01 Linear   -1.00   1.00  Avg 99900.000 99970.000 [f 0.03]
  V T02 Log      -1.00   1.00  Avg 99900.000 99975.000 [f 0.18]
A I F004 <SSN>  (ovfl) [VS group 0 2 30/freq 0.000]
V I F005 <SEX>  [VS group 0 2 30/freq 0.065]
  V T01 islit     0.00   1.00  <f> [f 0.06]
  I F006 <MARITALSTATUS>  [VS group 0 2 30/freq 0.968]
  V T01 islit     0.00   1.00  <m> [f 0.00]
    T02 islit     0.00   1.00  <s> [f 0.97]
  V T03 islit     0.00   1.00  <w> [f 0.02]
  I F007 <CHILDREN>  [VS group 0 2 30/freq 0.677]
  V T01 Linear   -1.00   1.00  Avg 0.000 3.000 [f 0.05]
    T02 tanh     -1.00   1.00  Avg 0.000 3.000 [f 0.66]
  I F008 <OCCUPATION>  [VS group 0 2 30/freq 1.000]
  V T01 islit     0.00   1.00  <manager> [f 0.10]
    T02 islit     0.00   1.00  <principal> [f 1.00]
  V T03 islit     0.00   1.00  <professional> [f 0.02]
  V T04 islit     0.00   1.00  <skilled> [f 0.00]
  V T05 islit     0.00   1.00  <unknown> [f 0.10]
  V T06 islit     0.00   1.00  <unskilled> [f 0.60]
  I F009 <HOMEOWNERSHIP>  [VS group 0 2 30/freq 0.065]
    T01 islit     0.00   1.00  <r> [f 0.06]
  I F010 <INCOME>  [VS group 0 2 30/freq 0.871]
    T01 Linear   -1.00   1.00  Avg 2004.000 8975.000 [f 0.69]
  V T02 Log      -1.00   1.00  Avg 2004.000 8975.000 [f 0.52]
  I F011 <EXPENSES>  [VS group 0 2 30/freq 0.839]
  V T01 Linear   -1.00   1.00  Avg 506.000 3202.000 [f 0.03]
    T02 tanh     -1.00   1.00  Avg 549.000 3541.000 [f 0.65]
  V T03 fzlft     0.00   1.00  506.000 506.000 549.000 [f 0.37]
V I F012 <CHECKING>  [VS group 0 2 30/freq 0.194]
  V T01 islit     0.00   1.00  <y> [f 0.19]
  I F013 <SAVINGS>  [VS group 0 2 30/freq 1.000]
    T01 islit     0.00   1.00  <y> [f 1.00]
V I F014 <MSTRCARD>  [VS group 0 2 30/freq 0.210]
  V T01 Linear   -1.00   1.00  Avg 1.000 5.000 [f 0.16]
  V T02 tanh     -1.00   1.00  Avg 1.000 5.000 [f 0.10]
  V T03 fzrgt     0.00   1.00  5.000 8.000 8.000 [f 0.00]
V I F015 <VISA>  [VS group 0 2 30/freq 0.129]
  V T01 Linear   -1.00   1.00  Avg 1.000 5.000 [f 0.03]
  V T02 Rt2      -1.00   1.00  Avg 1.000 5.000 [f 0.10]
  I F016 <AMEX>  [VS group 0 2 30/freq 1.000]
  V T01 Linear   -1.00   1.00  Avg 0.000 3.000 [f 0.11]
    T02 Log      -1.00   1.00  Avg 0.000 3.000 [f 1.00]
  I F017 <MERCHANT>  [VS group 0 2 30/freq 1.000]
    T01 Linear   -1.00   1.00  Avg 0.000 9.000 [f 0.89]
    T02 Log      -1.00   1.00  Avg 0.000 9.000 [f 1.00]
  O F018 <PAYMENTHISTORY>
    T01 Linear    0.00   1.00  Avg 0.000 12.000
```

Note that the table *Predict* generates for your model table may not exactly match this table. Discrepancies may be due to slight variations in parameter settings or minor adjustments to the algorithms between the version of *Predict* you are running and the one used to create this tutorial; or they may be due to the state of the random number generator which is used in various stages of model building.

The first thing to understand about this table is that lines that contain labels such as **F001, F002, F003** etc. correspond to the original data fields. This is easy to see with the credit example since field names are contained in the table. We recommend that you always use descriptive field names when you build a model so you can more easily associate the results of data analysis with data from the problem domain.

All other lines (except the first line, which contains *Input/Output Field Defaults*) include labels such as **T01**, **T02**, etc. These lines correspond to transformations of the associated field. When *Predict* performs data analysis, one or more transformations are created for most fields, depending on the analysis level and the type of field. For example:

- A field with numeric values may be transformed by a linear scaling function, a non-linear shaping function, and a fuzzy left and/or fuzzy right function for detecting outlying data for that field. An example of this is the <EXPENSES> field in the table above.
- A field which can take one of a small number of literal values (such as the <OCCUPATION> field in the table above) will usually result in a 1 of N transform (one transform for each unique value encountered in the field). Sometimes there will be fewer transforms than there are unique field values if there are not many records for a particular value.

Sometimes no transformation is created for a field. This can occur in several circumstances, such as numeric data that is constant or literal data that has too many different values. An example of this is the <NAME> field.

After data analysis is complete, all data is processed through all transformations to create a new set of transformed field data. The resulting data set is stored as a binary file, with an ".NPB" extension, on disk. The binary file is then loaded and used for the variable selection and neural network phases of model building – the *transformed* fields are input for *Predict's* Variable Selection facility.

Data Analysis Flags

Notice that on most lines in the analysis table there are one or two characters at the beginning of the line. Five of these ('I', 'A', 'V', 'U','C') were introduced in **Tutorial 1**. Another possible flag is 'O', which indicates an output field. Flag characters next to a field refer to the field, Flag characters next to a transform refer to the transform. Let's look at some examples to clarify these meanings.

- As illustrated below, the <`NAME`> field has been rejected by the data analysis process because every name is different and there is no useful predictive information in the field. This is shown by the ***A*** at the beginning of the line and the reason for the rejection is shown in parentheses (`ovfl`) — an abbreviation for overflow.

```
A I F001 <NAME>  (ovfl) [VS group 0 2 30/freq 0.000]
```

- As illustrated below, the data analysis process has generated one transformation for each possible value the <`OCCUPATION`> field can have. *Predict* sorts the transformations alphabetically. The variable selection algorithm has determined that all categories except <`principal`> are unimportant, as indicated by the 'V' at the beginning of the line for each of the other transforms. The VS group information refers to an advanced feature in which you can group fields together for the purpose of variable selection and require a certain minimum (and maximum) number of transforms to be selected from that group. The VS frequency information for both fields and transforms is contained in square brackets and preceded by *freq* or *f*. The value indicates the frequency of occurrence of the field or transform in the final population of the variable selection genetic algorithm. A frequency of 1.00 for the <`principal`> transform shows it is chosen 100% of the time and is therefore considered very important for the model.

```
I F008 <OCCUPATION>  [VS group 0 2 30/freq 1.000]
  V T01 islit     0.00   1.00  <manager> [f 0.10]
    T02 islit     0.00   1.00  <principal> [f 1.00]
  V T03 islit     0.00   1.00  <professional> [f 0.02]
  V T04 islit     0.00   1.00  <skilled> [f 0.00]
  V T05 islit     0.00   1.00  <unknown> [f 0.10]
  V T06 islit     0.00   1.00  <unskilled> [f 0.60]
```

- As indicated below, the data analysis process has generated a linear transformation, a non-linear transformation (the `tanh` function) and a `Fuzzy Left` function for outlier detection of small values of <`EXPENSES`>. However, only the linear transform has been selected by the variable selection algorithm as shown by the 'V' in front of the other transformations. Note that although the frequency of occurrence of the linear transform in the final population is 84%, and the frequency of occurrence of the `tanh` transform is 13%, the frequency of occurrence for the expenses field as a whole is 87.1% which is not the sum of the two transform percentages. This indicates that some variable sets in the population contain both the transforms.

```
I F011 <EXPENSES>  [VS group 0 2 30/freq 0.839]
  V T01 Linear   -1.00   1.00  Avg 506.000 3202.000 [f 0.03]
    T02 tanh     -1.00   1.00  Avg 549.000 3541.000 [f 0.65]
  V T03 fzlft     0.00   1.00  506.000 506.000 549.000 [f 0.37]
```

- As indicated below, the data analysis process generated a single linear transformation for the payment history field. The 'O' indicates this is a model output. Continuous outputs will always have at most one transformation generated. Enumerated outputs will have a one-of-N code generated.

```
O F018 <PAYMENTHISTORY>
    T01 Linear    0.00   1.00  Avg 0.000 12.000
```

Data Analysis Statistics

A summary of data analysis and variable selection results is shown in the bottom right area of the **Transforms** page. Basic counts are shown separately for both model inputs and model output(s). Active fields are those which have at least one transformation that has not been rejected by data analysis or variable selection.

The active fields in the *Data Analysis Table* above are <MARITALSTATUS>, <CHILDREN>, <OCCUPATION>, <HOMEOWNERSHIP>, <INCOME>, <EXPENSES>, <SAVINGS>, <AMEX> and <MERCHANT>.

Controlling the Data Analysis Process

Now we will explore the parameters that govern what types of transformations are generated.

1. Click the <EXPENSES> field in the analysis table so that it is highlighted. You may have to scroll up or down to find it. Make sure you select the EXPENSES field (**F011**) and *not* one of its transformations (T01, T02, or T03) that immediately follow the field line.
2. Click **Edit Selected Item...** (Alt E). This will open the **Field Parameters** dialog box.

Notice that the field name is set to EXPENSES, and the skip status is set to "active - no skip" meaning that the field has at least one transformation that has not been rejected.

One of the important sections of the **Field Parameters** dialog box is the *Transformations* pane on the right side. This lists all potential transformations that can be applied during data analysis. Transforms that are not checked are not candidates for application to the field. Transforms that are checked are the transforms that are applied; a small number of the "best" transforms are chosen according to criteria which favor a uniform distribution for the transformed field.

The following two steps show how to restrict the transformations that are considered for the current field.

3. Click the **Clear All** button. This removes all the transformations from consideration.

If *Predict* performed data analysis on this field now, the field would be not be processed since no transforms could be generated.

4. Scroll down and click the *log(x)* transformation. Since all the other transformations have been excluded, re-doing data analysis should generate log(x) as the only transformation for the <EXPENSES> field.

If you wanted to return to the default transformation settings you could click the **Set** button. This would set transformation selections to those that correspond to "moderate data transformation" (the item visible in the Field Parameters drop-down list).

5. Click **OK** to return to the ***Transforms*** page in the **Model Parameters** dialog box.

Look at the Data Analysis Table and locate the <EXPENSES> field (it should be visible and selected). Note that transformations for the field have been cleared (there are no transformations listed below the field itself) because you modified the scope of data analysis for the field (you may need to scroll down in the Data Analysis Table to confirm this).

6. Click the **Analyze Field** button (Alt I). Scroll down to see its new Log transform.

When the analysis completes, the Data Analysis Table should contain a single transformation for the EXPENSES field — Log. Now we will look at this transformation in greater detail.

7. Select (click on) the Log transformation for the <EXPENSES> field

8. Click the **Edit Selected Item...** button (Alt E) to open the **Continuous Transform** dialog box.

Each class of transformation has a specific dialog box for presenting the results of data analysis. The largest class of transforms contains the continuous functions. A continuous transform is implemented according to the following formula:

$$\begin{aligned}
&\mathit{If}\ \ x < x_{\min},\ x = x_{\min} \\
&\mathit{If}\ \ x > x_{\max},\ x = x_{\max} \\
&x = s_i x + o_i \\
&y = f(x) \\
&y = s_o y + o_o
\end{aligned}$$

where the first two lines implement clipping to the *Effective Minimum* and *Maximum* of the field, the next line implements pre-scaling using the *Inner Scale* and *Inner Offset* to map the field data to the range between *Inner Min* and *Inner Max*, the fourth line is the actual mapping function, and the last line implements post-scaling using the *Outer Scale* and *Outer Offset* to a range usable by the neural net. The various parameter values for a continuous function are set by the data analysis algorithm.

Traversing the Field and Transform Dialogs

The steps in this section will familiarize you with some of the different classes of transformations and how they are parameterized.

9. In the *Continuous Transform* dialog box, click the Cancel button to return to the ***Transforms*** page.
10. Use the scroll bar in the Data Analysis Table to view other fields and their transforms. Click any field or transform, then click the **Edit Selected Item...** button to view parameters (when you are building a model for an actual application, you may also want to change parameter settings; however, for this tutorial exercise only make changes suggested in the tutorial).

Disabling a Field

Sometimes you may want to disable a field to prevent its use in a model. Some of the reasons for this might be:

- The data corresponding to this field may be too expensive to collect when the model is deployed.
- There may be legal reasons for excluding the field.
- The field may not truly be part of the input data.

11. When you have finished viewing other transform types, locate the ZIP field in the Data Analysis Table, select it, and click the **Edit Selected Item...** button. Make sure you have selected the ZIP field (**F003**) and *not* one of its transformations (T01, T02, or T03).
12. To disable the ZIP field, select "user skip" in the **Skip Status** drop-down list.

13. Click **OK** to return to the ***Transforms*** page in the **Model Parameters** dialog box.

Notice that the ZIP field now has a 'U' by it, indicating that you (the user) have disabled it and do not want it to be part of any model. You could also have disabled the field by selecting "Variable Selection Skip"; the difference is that "User Skip" omits the field from the data analysis and forces a re-training of the Variable Selection Component, whereas "Variable Selection Skip" can be applied after the Variable Selection is complete in order to hand tune the selected variable set.

14. Click **Close** to return to the **Change the Model** dialog box.

Rebuilding the model

The changes you have made in these last few steps affect the format of the transformed data file, because some of the transformations have been eliminated. Consequently, the variable selection process, which acted on the original transformed data, needs to be performed again. The neural net also needs to be retrained because it depends on the results of the variable selection algorithm. Finally, since the skip status of the ZIP field was changed, the data analysis operation will be performed again. The following steps show how to update the model.

1. Click the **Train** button in the **Change the Model** dialog box.
2. When the **Update the Model** dialog box appears, select "Update as Required" in the drop down list.
3. Click **Start**. *Predict* will display a **Save Model** dialog box, to permit you to change the name of the model and not overwrite the original mode. For this exercise, you can accept the default name (the current model name). Click the **Save** button in the **Save Model** dialog box to save the model; *Predict* then starts rebuilding it.

Even though *Predict* performed data analysis on the EXPENSES field when you clicked the Analyze Field button, the data analysis operation will be performed again since the skip status of the ZIP field was changed. This in turn forces a rebuild of the binary file that *Predict* uses for intermediate steps, and then the variable selection and neural net components of the model are retrained.

4. When the new model is rebuilt, the **Training Complete** dialog box appears. After you review the results, click **OK** to close the dialog box.

Enabling a Field

Suppose having built a model you find that a particular field you would like as input to the model has not been retained by the variable selection algorithm. The following steps show you how to enable the field and rebuild the neural net model without re-training variable selection. You can also disable an unwanted field in a similar manner.

1. Assuming you have closed the **Training Complete** dialog box, open the **Change the Model** dialog box (click **Change...** in the Predict menu).
2. Click the **More Parameters...** button to open the **Model Parameters** dialog box, then click the ***Transforms*** tab to view the ***Transforms*** page.
3. Find a field in the *Data Analysis Table* that has been rejected by variable selection (indicated by a 'V' at the beginning of a line that contains Field information).
4. Click any of the transforms generated for that field. The transform will also have a 'V' at the beginning of the line.
5. Click the **Edit Selected Item...** button to open the appropriate Transform dialog box.
6. Enable the transformation by changing the **Transform Status** from "variable selection skip" to "active - no skip". Click **OK** to close the Transform dialog box.
7. Click the **Close** button in the **Model Parameters** dialog box to return to the **Change the Model** dialog box.
8. As you did earlier, click the **Train** button to open the **Update the Model** dialog box.
9. Select "Update as required" in the drop down list and click **Start**. Accept the default name to save the model; click **Save** and *Predict* will begin rebuilding the model.

Notice that only the neural net is retrained. Variable selection is not performed again – the variable you enabled is added as a model input, and then the neural network is re-trained.

10. When the new model is rebuilt, the **Training Complete** dialog box appears. After you review the results, click **OK** to close the dialog box.
11. Demonstration version users should skip this step and proceed to **Tutorial 3**. Select the **Close** command from the Predict menu to close the network. From Excel's *File* menu, also select the **Close** command to close the Credit.xls Worksheet.

Credit Risk Tutorial 3

This tutorial shows how to:

- See which data records have been placed in the training and test set
- Modify the train/test set selections
- Manually select train/test sets
- Reset default parameter settings
- Override default settings
- Interrupt and resume learning

Insights regarding why training and test file separations are important can be gained by reading our **Introduction** chapter of the *Predict User Guide*. The chapter includes a section on Selecting Train, Test and Validation Sets; please see **Start | Programs | NeuralWare | NeuralWorks | Predict | Predict User Guide** for details.

Modifying the Train/Test Set Selection

1. If Excel, *Credit.xls* and your *Credit.npr* model are still open from **Tutorial 2**, then open the **Change the Model** dialog box (click **Change...** in the Predict menu).

   Otherwise, start Excel and use **File | Open** to open the data file *Credit.xls* from **Tutorial 2**. Then open the *Predict* model: *Credit.npr* is visible in the Predict menu as one of the most recently used files unless you are running the demonstration version of *Predict*, or have built several of your own models in the meantime. If *Credit.npr* is not visible in the Predict menu you can use the **Predict | Open** command to open your *Credit.npr* model, or open our distributed file *credit3.npr*. Your saved model is probably in `C:\My Documents`; our model is in our product install directory, probably `C:\Program Files\NeuralWare\NeuralWorks\Predict`.

2. When the **Change the Model** dialog box appears, click the **More Parameters...** button to open the Model Parameters dialog box. Since you will be writing information into the Worksheet, you may want to reposition the **Change the Model** dialog box before you open the **Model Parameters** dialog box. Then click the ***Data Sets*** tab to view the parameters that control Train, Test, and Validation Set selection.

The following describes some of the elements of this dialog, and how *Predict* sets up train and test sets by default.

Settings in the **Partitions** frame of the ***Data Sets*** page affect how Validation, Train, and Test data sets are chosen from the full data set.

By default, all data that you specified in the *All Input Data* range is used as the Validation data set. A common option if you have sufficient data is to exclude a percentage of the data to be used for independent validation, and use the remaining data as the *Primary* data set.

The *Primary* data set is also, by default, all the data.

The *Secondary* data set allows an outcome-based pruning of the data. This addresses some of the issues discussed in the **Introduction** about discarding data. Again by default, *Predict* has selected all data from the *Primary* data set for constructing train and test sets.

The *Train Set* has been selected in a round robin manner as 70% of the *Secondary* data set and the *Test Set* is formed by the remainder of the *Secondary* data set ("not train set").

3. To view how *Predict* partitioned the available data, in the **Action** frame click the **Write Flags...** button (Alt W).
4. When the *Write Train/Test Flags* dialog box appears, enter the range **V2** (equivalently **R2C22**) or click cell **V2**.
5. Confirm that the **Use Symbolic Equivalents for Flags** check box is selected and Click **Write**.

When you click **Write**, *Predict* writes information to the worksheet about which records are in each of the subsets (train, test, validation) and whether or not the data record is active and/or in the Secondary working set.

6. Click **Close** or press the **Esc** key to return to the ***Data Sets*** page.

Observe that strings of the form "AWT_V" and "AW_SV" have been written to the worksheet in column **V**, each string corresponding to the record in that row. The information of interest is represented by the characters 'T' (Train set), 'S' (Test set) and 'V' (Validation set). As expected, all the records are in the validation set, about 70% of the data records are in the training set, and 30% are in the test set.

We'll now modify the default settings so that some independent data is excluded and used for validation.

7. On the ***Data Sets*** page, in the **Validation Set** drop-down list click the "*random*" entry as the **Selection Method**.
8. To the right side of **Validation Set**, double-click on 100.0 and change this number to **20.0**. This allocates 20% of **All Data** to be used as validation data.
9. In the **Primary Working Set** drop-down list, click "not validation data".

Do not change any other settings.

10. In the **Action** frame, click the **Select Sets** button. The train, test and validation sets are repartitioned using the new settings.

Now you can view the new train/test information in the worksheet.

11. Click the **Write Flags...** button (Alt W) in order to write the new train/test information in the worksheet.
12. Type the range **W2** in the range text box (equivalently **R2C23**) or click cell **W2**.
13. Click **Write**.
14. Click **Close** or press the **Esc** key to return to the ***Data Sets*** page.

As expected the new flags in column **W** are different from those in column **V**. 20% of the records have now been set aside for a validation set.

You could also have provided a validation set by explicitly not including a contiguous block of data records in the Worksheet when you originally specified the **All Input Data Range**.

Selecting Train, Test, and Validation Sets Manually

Similar to the same way the **Write Flags** command writes to the worksheet how data records are partitioned for training, testing, and validation, information in a Worksheet can be read and used to **define** data set partitioning. To see how this is done do the following.

1. Click the **Close** button in the **Model Parameters** dialog box to close it.
2. Click the **Close** button in the **Change the Model** dialog box to close it.
3. In the **W** column, find the first occurrence of "____V". This is the first occurrence of a validation record.

4. Manually change this to a training record by replacing "____V" with "AWT__", or more simply, with 't'. When defining a partition manually, you only need specify 't', 's', 'v' or any combination of these characters to describe which sets a given record is in. For example, "T_V" or "tv" are both valid to describe a record which is in the training set and the validation set but not in the test set.
5. Return to the ***Transforms*** page. Click the ***Change...*** command, then the **More Parameters...** button, then click the ***Data Sets*** tab.
6. In the lower portion of the ***Data Sets*** page, click the **Partition flags from Worksheet** check box.
7. Click in the range text box, and type **W2** (equivalently **R2C23**) or click in cell **W2**. This tells *Predict* where to find user specified flags.
8. Click **Select Sets**. Partitioning progress information will appear in the Excel status bar.
9. Click the **Write Flags...** button (Alt W) in order to view the new train/test information in the worksheet.
10. Type the range **X2** (equivalently **R2C24**) or click that cell.
11. Click **Write**.
12. Click **Close** to close the **Write Train/Test Flags** dialog box. Click **Close** again to close the **Model Parameters** dialog box. Finally, click **Close** to close the **Change the Model** dialog box so that you can view the entire Worksheet.

The flags in column **X** correspond to the flags in column **W**, which demonstrates that the train/test/validation partition manually specified in column **W** has been imported by *Predict*.

If you want you can now re-build your model with the new Train and Test Sets. Then run a test. The performance on the validation set gives a good indication of how the model will perform with new data.

Modifying Default Network Parameters

Because the worksheet has become fairly messy, and model parameters have changed considerably from their original settings, you should close the model and the worksheet without saving, and start anew.

1. Select the **Close** command from the Predict menu. Click *No* if you are prompted to save the network.
2. Close the worksheet in Excel without saving it and then re-open it.
3. From the Predict menu, select the **New** command. Assuming that the last model you built was Credit.npr, click the **Use data ranges from last Model** check box, and *de-select* (un-check) the **Launch the Model Building Wizard** check box.

When the *Model Building Wizard* is not active, the **Building a Model** dialog box appears after you click the **Next** button in the **New Model** dialog box (the **Building a Model** dialog box is identical to the last dialog box in the *Model Building Wizard* dialog box series). All the ranges and parameters should be correct in this dialog because they have been stored.

You should also see the following settings:

Model Type	prediction
Noise Level	moderately noisy data
Data Transformation	moderate data transformation
Variable Selection	comprehensive variable selection
Network Search	comprehensive network search

4. Make sure the default settings and the ranges match those shown above.

You can modify these default parameter values however you in the **Model Parameters** dialog box as you did in **Tutorial 2** when you modified the possible transformations for one of the input fields.

5. Click the **More Parameters...** button to open the **Model Parameters** dialog box. Click **Save** to save and name your model.

The ***General*** page in the **Model Parameters** dialog box contains a collection of parameters that you most likely would change as you experiment with different models.

6. In the **Network** frame on the ***General*** page, locate the **Evaluation Function** drop-down box and change the function from "correlation" to "accuracy".

The *Evaluation Function* is used to evaluate the model with Test Set records during training to determine the best candidates when hidden nodes are added. The Evaluation Function is also used to determine when network performance is no longer improving.

By default, "correlation" is used to evaluate prediction problems. However, "accuracy" may be a more appropriate measure for some problems. The accuracy measure counts the fraction of records whose predicted output is within a specified range of the target output. The default accuracy tolerance is 20% of the output range.

7. Click **OK** to close the **Model Parameters** dialog box and return to the **Change the Model** dialog box.

Note that the drop-down list boxes for *Noise* level and *Network Search* level contain "Specified by User." If at this point you changed the *Network Search Level* or *Noise Level* (either of which affect the neural network parameters) one of the standard default settings, neural network parameters would be reset to their corresponding their default values.

8. Click **Train** to open the **Update the Model** dialog.

Interrupting and Resuming a Build

9. Click **Start**, accept the default file name and save the model. Some time after the Variable Selection operation begins, but before the model is fully trained, click the **Cancel** button.

You can interrupt variable selection or neural net training, modify parameters, and then allow *Predict* to resume building the model. Some parameter changes force a complete rebuild, but other changes will permit *Predict* to begin from where it was interrupted.

10. Click **Close** to close the Status dialog box, then click **Close** in the **Building a Model** dialog box to close it also (since in this tutorial we started with a new model, if you do not close the **Building a Model** dialog box, *Predict* will start training the model from the very beginning, rather than resuming where training was interrupted).
11. From the Predict menu, click the **Change...** command to re-open the **Change the Model** dialog box.
12. Click the **More Parameters...** button to open the **Model Parameters** dialog box, then click the ***Neuro-Dynamics*** tab. Select "sine" instead of "end-of-list" for the second **Hidden Candidate Pool**. This allows *Predict* to select from hidden units which have either a *tanh* transfer function or a *sine* transfer function.
13. Click the ***Heuristics*** tab to view the ***Heuristics*** page. Change the **Minimum Trials** value from 3 to 4 in the **Hidden Unit(s)** column. This will allow at least two tanh candidates and two sine candidates to be tried for each hidden unit established in the network.

14. Click **OK** to close the **Model Parameters** dialog box and return to the **Change the Model** dialog box.
15. Click **Train** to open the **Update the Model** dialog box. Scroll up and select "Resume after interruption" from the drop-down list.
16. Click **Start**, and accept the default file name. *Predict* resumes training at the point training was interrupted. When training ends, the **Training Complete** dialog box appears. Review the information, then click **OK** to close the dialog box.
17. From the Predict menu, click the **Close** command to close the model.

Thank you for completing all of our tutorials. We hope they have helped you gain an understanding of *Predict's* capabilities and its powerful analytic modeling technology base. In addition we hope these tutorials have demonstrated how easy it is to quickly build models that can be applied to real problems. Please continue to explore *Predict's* capabilities by building models with your own data, from applications of interest to you.

Call or email us anytime (see *Contacting NeuralWare* at the end of Appendix B) if you have questions!

If you would like to discuss extended empirical modeling and analysis assistance for your application development efforts, please contact sales@neuralware.com.

Glossary

The following terms are frequently used in ***neural computing***. Words or phrases that are bold and italicized are defined in the Glossary. Words or symbols in parentheses indicate either units for the term or the abbreviation for the term when it appears in a *Predict* dialog box.

Accuracy (%)
The percentage of ***predicted outputs*** that are within the user-specified tolerance of the corresponding ***target values***.

Average Absolute Error (Avg. Abs.)
The average of the absolute differences between ***predicted outputs*** and their corresponding ***target values***.

Confidence Interval (%)
This interval establishes the range [target value ± confidence interval] within which the corresponding ***predicted output*** occurs with a specified degree of confidence.

Correlation Coefficient see **R Value**

Empirical Model
A model that is developed by fitting a function to a dataset of known inputs and corresponding outputs. Rather than using *a priori* knowledge of relationships of variables in the system to define the function (as in a ***first principles model***), the general form of the fitting function is specified and then the coefficients of the function are determined from the data.

First Principles Model
A mathematical description of a system that is derived from theoretical considerations of the underlying physical, chemical, biological, or other processes that cause the dynamic behavior of the system. Data collected in the system serve as input to first principles models; predicted outputs are computed using the equations that comprise the model.

Generalization
The ability of a model to produce a reasonable output when it is presented with input data different from the data used to train the model. For neural network models, the output is generated by interpolating from the historical examples (containing both input and output values) the neural network was trained on. A model that does not generalize well is often the result of ***over-fitting*** the model to the ***training data***.

Historical Data
The data, collected from the application domain, used to build a model. The data is usually organized as records (in rows) that contain input variables followed by output variable(s). Historical data is usually split into separate ***training***, ***test***, and ***validation sets***. The training and test sets are used to build the model and are called ***modeling data***. The neural network uses this data to learn the relationship between input and output variables (i.e., to create a function that maps inputs to outputs). The validation set is usually separate from the modeling data and is used to independently assess how well the model performs.

Input Variable
The variables which are inputs for the model. The output of a neural network model is a non-linear function of input variables. Input variables are the independent variables in the function.

Learning see **Training**

Learning Rule
The method used to train the network (i.e., the algorithm used to adjust the ***weights*** and, depending on the learning rule, possibly alter the network architecture).

Linear Correlation Value see **R Value**

Linear regression
A method of deriving a relationship between two or more variables where the predicted relationship between the variables is linear. A straight line (or hyperplane) is drawn to “fit” the data.

Model
In general, a model is a mathematical representation of a system that can be used to understand or manipulate the system. In the context of ***neural computing***, building a model refers not only to defining and training the core neural network, but also performing all of the pre- and post-processing steps related to computing an output or outputs based on known input values.

Modeling Data
The subset of the ***historical data*** that is used to build a neural network model. This includes the ***training set*** and the ***test set***.

Net-R
The linear correlation between the ***target values*** and the *raw* ***network outputs***.

Network Output
The output of the neural network before it is transformed into values consistent with units of measure in the ***historical data***. Network output values are in the range (-1, 1) or (0, 1). Network outputs are then scaled by measurement units to yield ***predicted outputs***. (Network output is also called *raw* network output.)

Neural Computing
The study of networks composed of many simple units (***processing elements***) that are adapted through exposure to information (***training*** with data examples). Neural computing is a branch of machine learning and a method of performing ***empirical data modeling***.

Noise
The degree of cleanness or consistency of data. Noise in input data usually results from measurement error. Noise also refers to variation in ***target values*** that either cannot be predicted from the available inputs or is inherently unpredictable (i.e., random) regardless of the inputs. The following guidelines provide typical noise levels for various types of data: mathematical function data is clean; behavioral data is generally "moderately" noisy; stock market data is very noisy.

Over-fitting
Over-fitting occurs when the mapping function that results from ***training*** a model fits the ***training set*** too well. A model that is over-fit does not ***generalize*** well when new data that was not represented in the training set is supplied to the model.

Predicted Output
The output of a neural network after it is transformed into the original measurement units of the application domain. See also ***network output*** and ***target value***.

Processing Element (PE)
The basic building block of a neural network. In most neural network architectures, a processing element calculates a weighted sum of inputs (an activation value) and produces an output value that is a non-linear transformation of the activation value. A neural network is composed of many processing elements (also called nodes or neurons). ***Weights*** that connect processing elements are adapted through the process of ***training***.

R Value
Also known as Pearson R, it is a measure of the linear relationship of two random variables. It ranges from –1.0 to +1.0. Perfect negative correlation is indicated by –1.0, absence of correlation is indicated by 0, and perfect positive correlation is indicated by +1.0.

Root Mean Square (RMS) Error
A measure of network performance during training. It is the square root of the average squared error between the ***target values*** and corresponding ***predicted outputs***.

Target Value
The value in the ***historical data*** that is the recorded output for an instance of input variables. It is the value that the network is attempting to predict.

Test Set
A subset of the ***historical data*** used during neural network ***training*** in order to construct the network architecture and to prevent ***over-fitting***. Periodically during training, the network is run with data from the test set and its performance is scored. This score is used both to choose between candidate hidden nodes when the neural network itself is being constructed, and to determine when to stop training the network. For a detailed explanation of the difference between the test set and ***validation set***, refer to the chapter "Train, Test, and Validation Sets" in the *NeuralWorks Predict User Guide*.

Testing
Running a neural network using ***historical data*** as input and evaluating the accuracy of network outputs when compared with outputs (***target values***) in the historical data. A neural network can be tested using any dataset that contains inputs associated with known target values, and various error metrics can be generated and evaluated. When a *Predict* model is being trained, periodically the training is interrupted and the model at that point in time is tested with the ***Test Set***.

Training
The process of repeatedly presenting examples of ***historical data*** to a neural network and altering the connection ***weights*** and other parameters of the network based on a ***learning rule***. In *Predict*, neural network weights are iteratively adjusted and ***processing elements*** are added in order to minimize the difference between the ***target values*** and the network predictions as the network is trained using ***modeling data***. As a network is trained, it is also sometimes characterized as learning.

Training Set
A subset of the ***historical data*** used to train a neural network (adjust the ***weights*** and in some cases, add ***processing elements***).

Transformation
Eliminating outliers, scaling values, and applying other mathematical operations to input data such that the resulting distribution of input values used to train the network produces the best neural network. Refer to the chapter "Data Analysis and Transformation" in the *NeuralWorks Predict User Guide* for a more detailed explanation of what occurs when *Predict* performs data transformations.

Validation
Estimating how well the model will perform once it is deployed. This requires the use of a proper ***validation set*** and careful consideration of the application's true objectives.

Validation Set
A subset of historical data, usually distinct from the ***modeling data*** (training and test sets). The validation set is withheld for estimating model performance in a deployed environment. For a detailed explanation of the difference between the ***test set*** and the ***validation set***, refer to the chapter "Train, Test, and Validation Sets" in the *NeuralWorks Predict User Guide*.

Weights
If two ***processing elements*** are connected, that is, the output of one processing element is an input to a second processing element, the connection has a corresponding weight that is initially set to a random value. During neural network ***training***, the weights on connections are gradually adjusted, which is how the network "learns" the relationships inherent in the ***training data***.