Shortcut to Success

An introduction to

NeuralWorks Predict®

The Ultimate Solution for Data Analysis Problems

August 2005

Introduction

Welcome to the power of ***Predict*** – the ultimate solution for your data analysis problems. ***Predict*** is an advanced modeling and data mining platform for building robust neural networks to analyze data and produce actionable knowledge. Whether you are a business user, a scientist, an engineer, or an analyst in search of a solution, this brief *Shortcut to Success* will show you how to take advantage of the power of ***Predict.***

Neural networks are a time-tested and field-proven technology for discovering complex or unknown relationships in data. Basically, a neural network detects patterns. First it learns to generalize about patterns in historical data – then it can produce output values (i.e., make unaided prediction or classification decisions) when it recognizes similar patterns in new data from the same problem domain.

Whatever your data analysis requirements are, ***Predict*** can help. Right now, the power of ***Predict*** is solving a wide range of problems across diverse industries around the world – including:

- identifying financial trends that influence market timing, security buy/sell, and asset allocation decisions;
- identifying and evaluating credit risks, both during the application process and after credit is granted;
- analyzing complex medical conditions that influence treatment regimen choices;
- selecting demographic groups that possess specific characteristics for targeted marketing;
- developing models to improve manufacturing processes and supply chains;
- identifying and classifying biometric attributes, recognizing hand-written characters and spoken language, and processing and interpreting virtually any type of image data;
- and many others!

If you have basic computer skills and are familiar with Excel, you too are ready to apply the power of ***Predict***!

After you've completed the *Shortcut to Success*, you can explore additional ***Predict*** features and capabilities in tutorials in the *Getting Started Guide* – more information is at the end of this tour.

Before we begin please note the following:

- Because you will be using demonstration software, you won't be able to save your models even though a **Save Model** dialog box does appear.
- You can use the demonstration software to analyze your own application data; however, models you build will be limited to 32 fields, and will be capable of processing only 512 records. Of course, the full product can handle thousands of fields and hundreds of thousands, even millions, of records!

For a quick explanation of terms related to neural networks and ***Predict*** software, you'll find a helpful glossary at the end of this document.

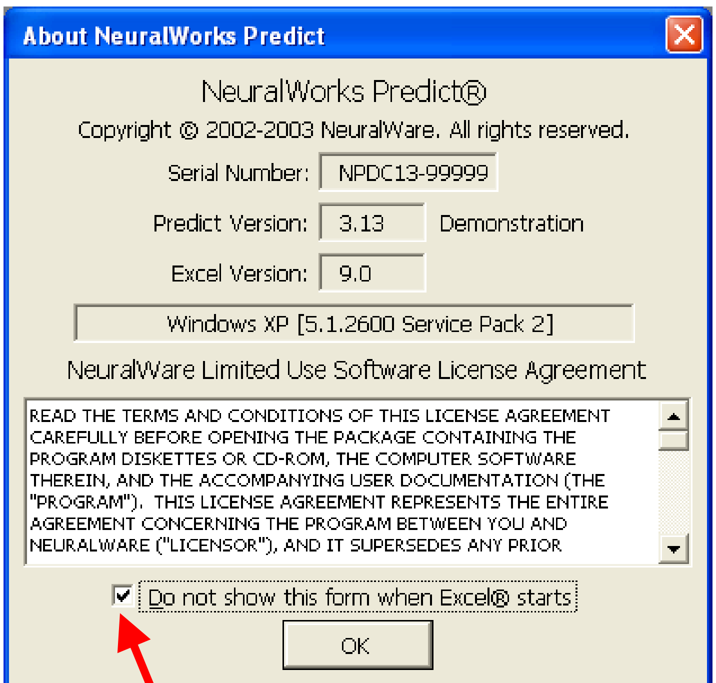
Let's Get Started!

We assume that Microsoft Excel® is installed on your computer, and that you have successfully downloaded and installed the ***Predict*** demonstration software that accompanies this *Shortcut to Success*. If so, let's go!

> *If Excel is not on your computer, or if you have not downloaded the **Predict** demonstration software, please take those steps before you continue.*

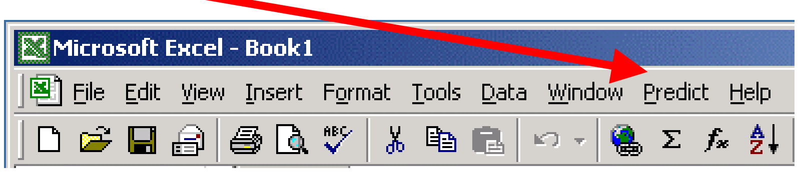
You start ***Predict*** by starting Excel – ***Predict*** becomes a new menu in the Excel menu bar. So please start Excel.

The **About NeuralWorks Predict** dialog box appears the first time you run Excel after you install ***Predict***:



To continue, select (click in) the check box as shown above. Then click **OK**.

Your Excel menu bar should look like the one below. The positions or names of some menus may differ but the ***Predict*** menu <u>must</u> appear in the menu bar.



> *In the unlikely event you do not see the **Predict** menu, or if you need assistance during any part of this **Shortcut to Success** tour, please contact NeuralWare Product Support –* e-mail: `help@neuralware.com` or phone: `+1 (412) 278-6280` (`option 2`).

What are we going to do?

The purpose of the model you will build is to predict the rate of moisture evaporation from soil. The moisture evaporation rate directly influences whether or not cropland should be artificially irrigated. Yet the decision to start irrigating is an expensive one – for example, water must be purchased, electricity or fuel must be purchased to operate pumps, and additional personnel may be required to monitor and maintain the irrigation system.

At the same time, if cropland is not irrigated and subsequently the harvest is damaged due to drought, revenue from the sale of crops will be lower than expected.

Clearly, the ability to accurately predict the moisture evaporation rate can greatly improve the probability of having a good (therefore profitable) harvest; accurate models will also help minimize variable operating costs through avoidance of unnecessary irrigation.

Although the model you will build uses data from a very specific problem domain, the *reason* for building the model is representative of the reason for virtually all modeling efforts. More accurate forecasts or classifications ensure better (more appropriate) decisions – which drive revenue growth, produce lower costs, or simply are more accurate (in the case of models that perform diagnostic or discovery functions).

How do I set up a model?

After you understand why you are building a model, the rest is easy! When you begin to develop models to address your own analytic challenges, you'll see that not all problems are as clear-cut as the moisture evaporation problem. But once you define the problem, building the necessary models using ***Predict*** is extremely simple.

From the Excel **File** menu, click **Open.** Browse to and open this Excel spreadsheet:

`C:\Program Files\NeuralWare\NeuralWorks\Predict\evap.xls.`

The first six columns contain values that represent measurable conditions, or attributes, of the problem – such as soil temperature, relative humidity, and wind speed. In every row, values in these six columns are "independent variables" whose combined effect influences the value in the final column (the daily evaporation rate). The daily evaporation rates (the "dependent" variables) are the *Target* values that will be used to train the neural network model.

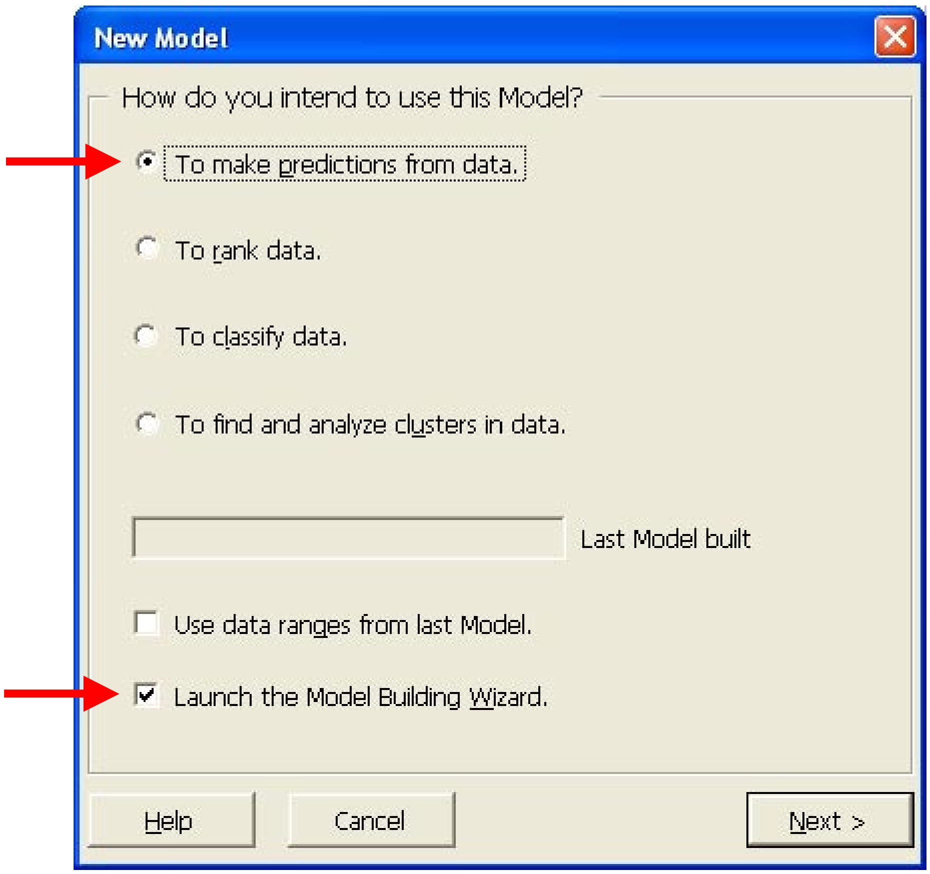
To begin, you should first save the spreadsheet with a different name.

1. From the Excel **File** menu, click **Save As.**

2. Enter "*tutorial*" (without the quotes) in the text box labeled **File Name**, then click **Save.**

Now, from the **Predict** menu in the Excel menu bar, click **New** to begin creating your neural network model.

The **New Model** dialog box appears.



As we noted earlier, the moisture evaporation problem is a *prediction* problem.

1. Select **To make predictions from data.**

2. Confirm that **Launch the Model Building Wizard** is selected.

3. Click **Next** to continue.

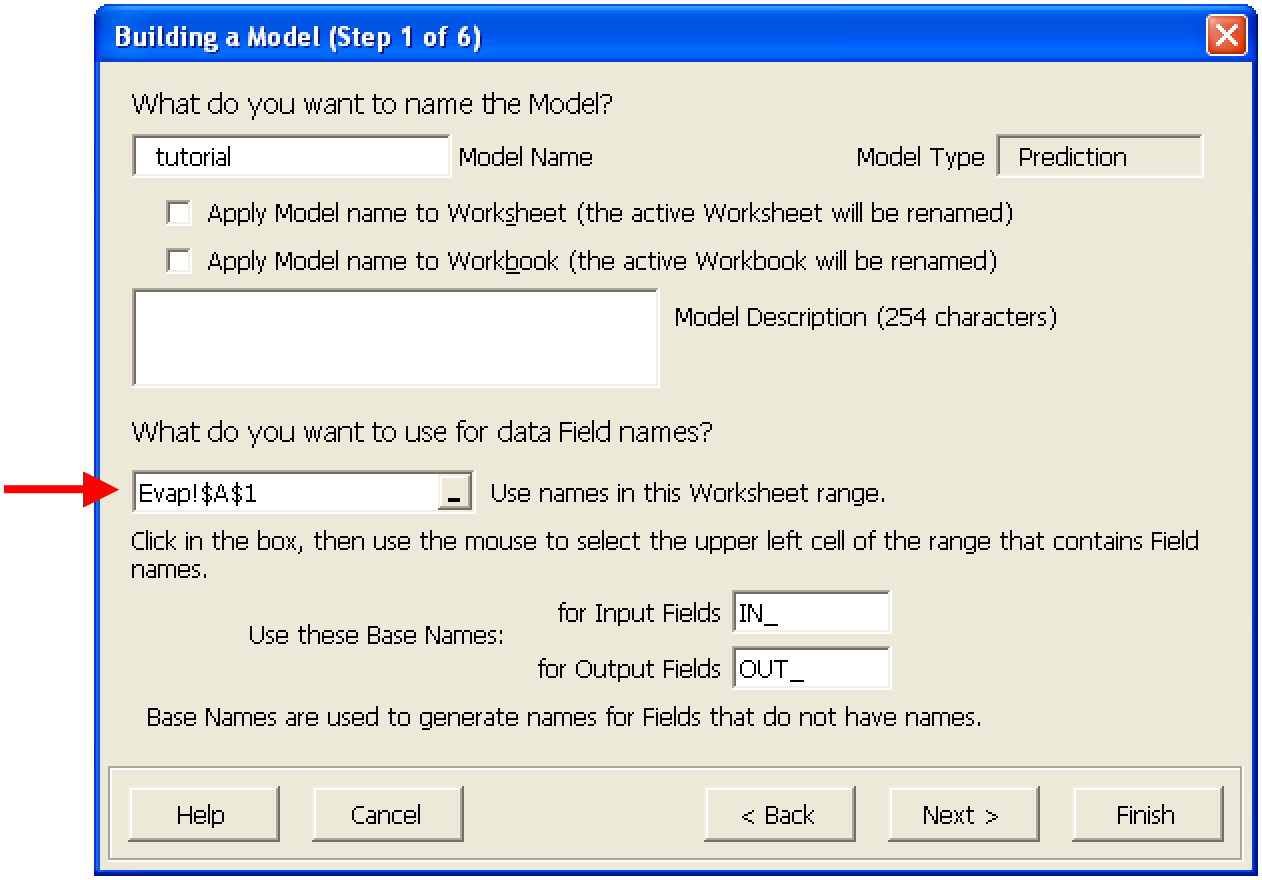
The first time you build a model using ***Predict***, the **Model Building Wizard** Welcome dialog box appears.

The Welcome dialog box previews the steps you take as you build a model.

1. If you don't want to see the Welcome preview again, select the check box.
2. Click **Next** to continue.

Building a Model – Step 1

In the **Building a Model (Step 1 of 6)** dialog box, you specify names for elements of the model.



Note that the model itself is named *tutorial* – the ***Predict*** default is to use the spreadsheet name.

You need to tell ***Predict*** where to find the row in the spreadsheet that contains names (or "*headers*") for data fields.

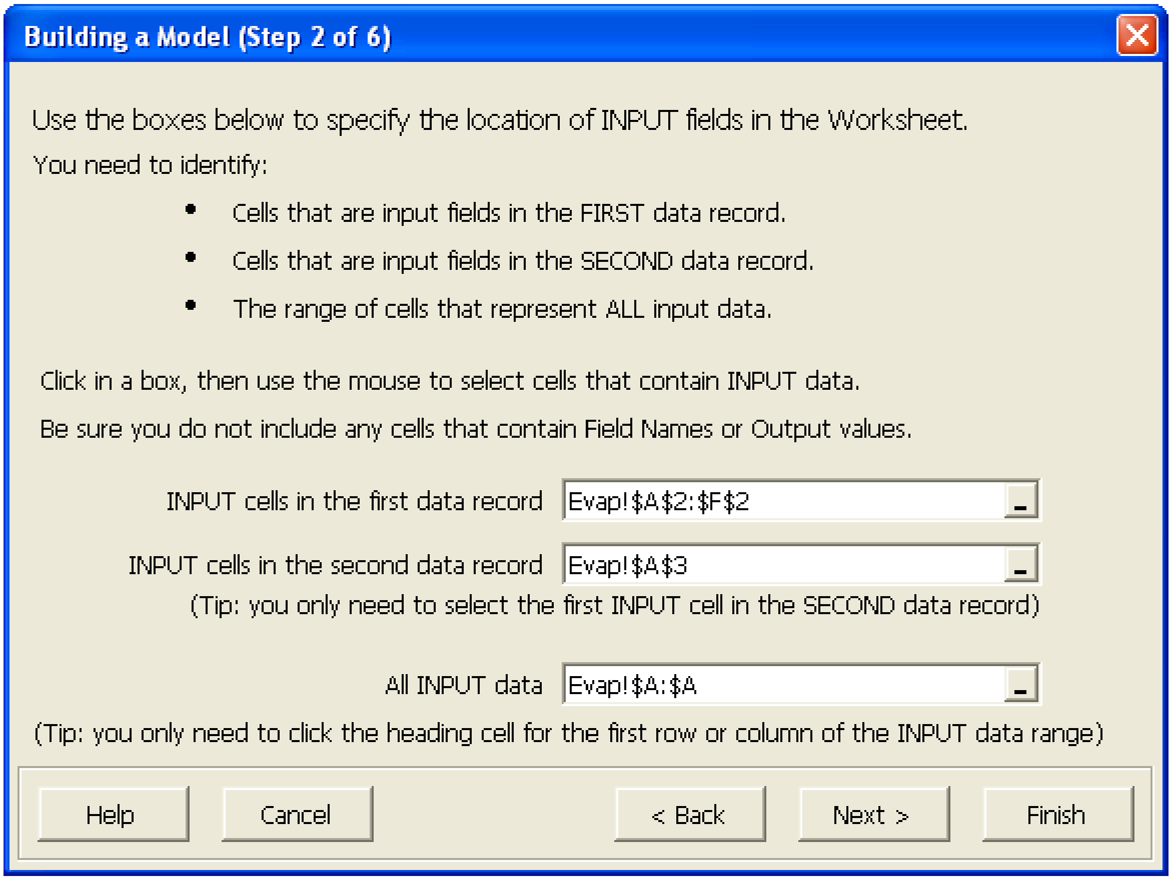
1. Click in the text box under the label **What do you want to use for Data Field names?,** then click in cell **A1**.

   Predict will automatically fill in the text box with location of the header row.

2. Click **Next.**

Building a model – Step 2

In this step, you tell ***Predict*** how _input_ data values are organized within *records* in the spreadsheet. Each row in the spreadsheet is considered an individual data record.



1. Click in the text box labeled **INPUT cells in the first data record**, then move the mouse over the spreadsheet and drag from cell **A2** to cell **F2** (be sure to include cell **F2**).

   This action tells ***Predict*** that input values in the first data record are found in cells **A2** through **F2**. Notice that ***Predict*** automatically fills in the text box with the range.

2. Now, click in the text box labeled **INPUT cells in the second data record**, then click in cell **A3** – which is the first cell in the second record. This time you do not have to drag – ***Predict*** will later calculate that the record ends in cell **F3**).

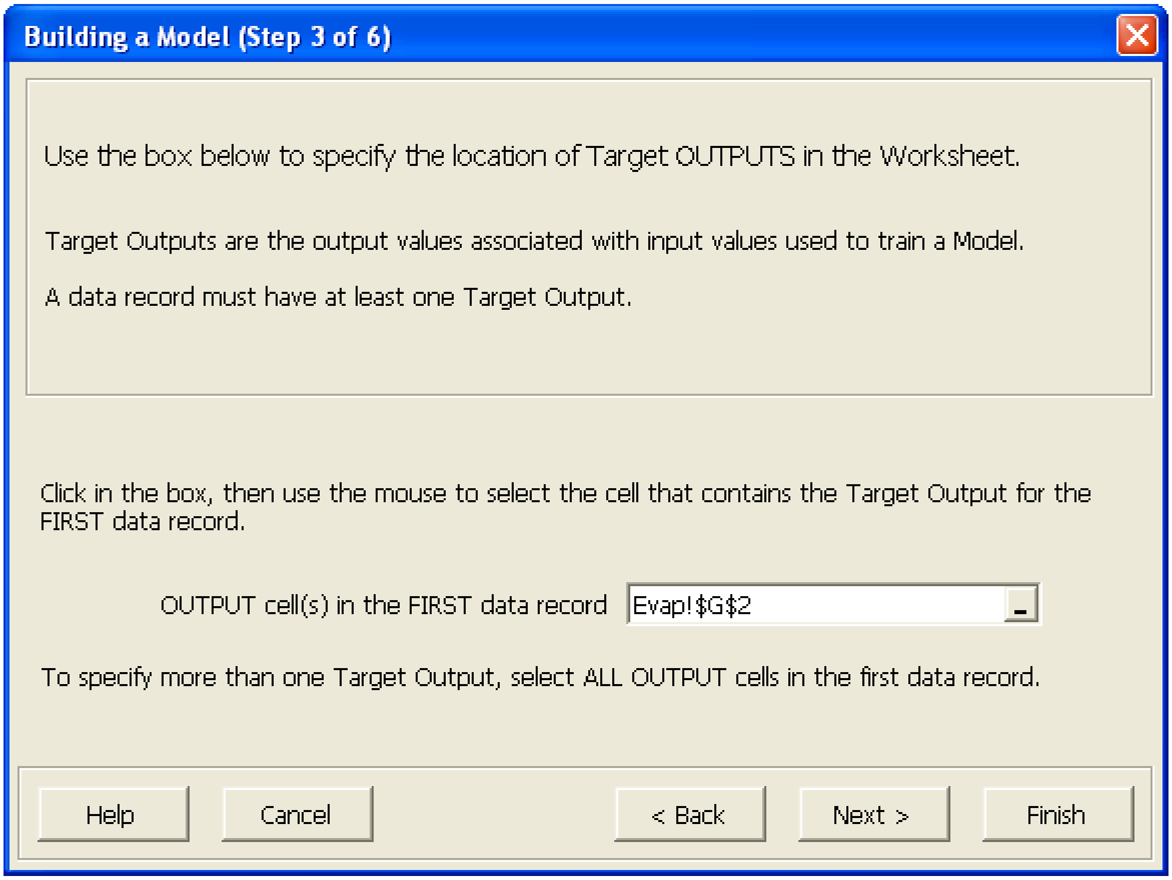
3. Finally, click in the text box labeled **All INPUT Data**, then click the **A** column header in the spreadsheet.

   This action selects the entire column and tells ***Predict*** to use input values from all records in the spreadsheet.

4. Click **Next**.

Building a model – Step 3

In this step, you tell ***Predict*** how <u>*output*</u> data values (the *target* values) are organized in records in the spreadsheet. Output values usually follow input values in data records.



Remember that, from the problem description, daily evaporation rates in the last column of data are the *Target* values which the model will learn to predict.

1. Click in the text box, then click in cell **G2** to tell ***Predict*** where the output value in the first data record is located.

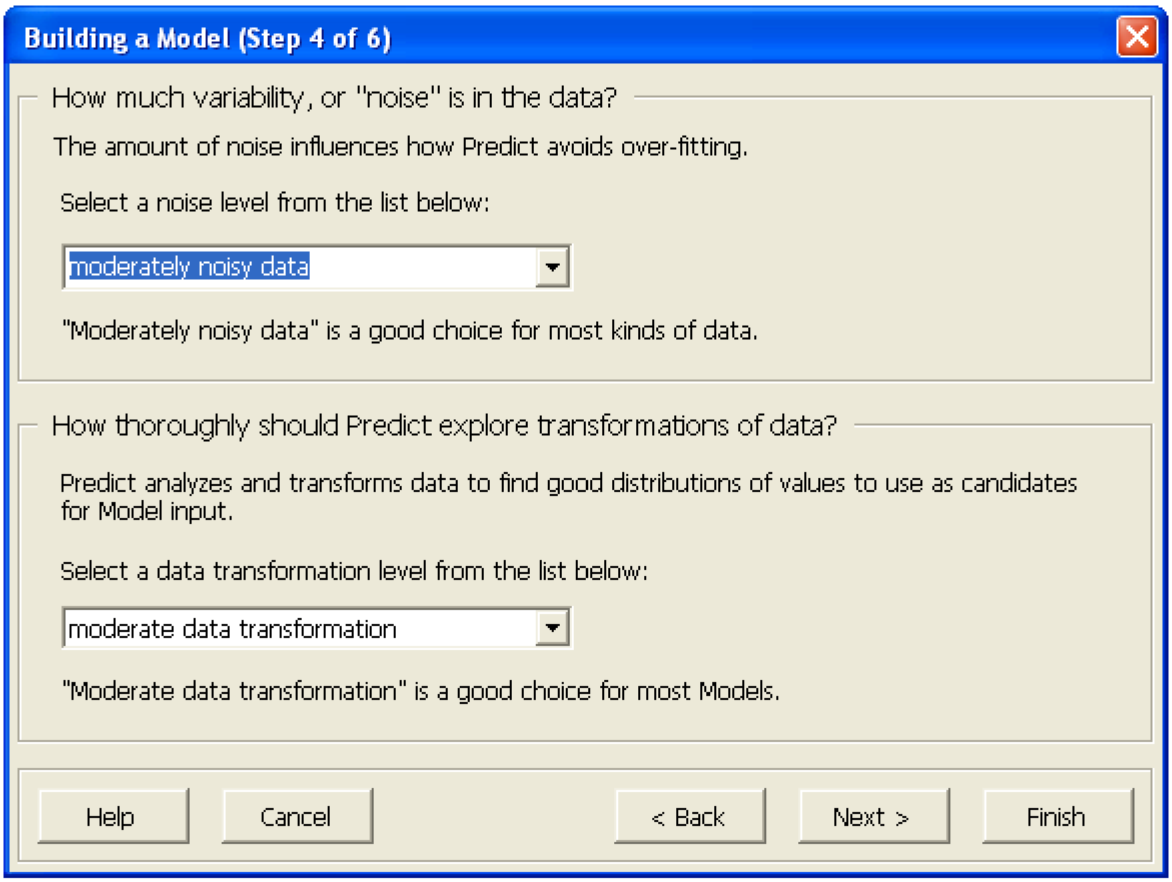
   Notice that you only need to specify the location of the output value in the *first* data record – ***Predict*** can then determine the field layout relationship between input values and output values in all the other data records.

2. Click **Next.**

Building a model – Step 4

In this dialog box, you normally would consider and select options that tell ***Predict*** about characteristics of the available data and how the data should be transformed (pre-processed).

However, for this *Shortcut to Success* tour, just accept the default options.

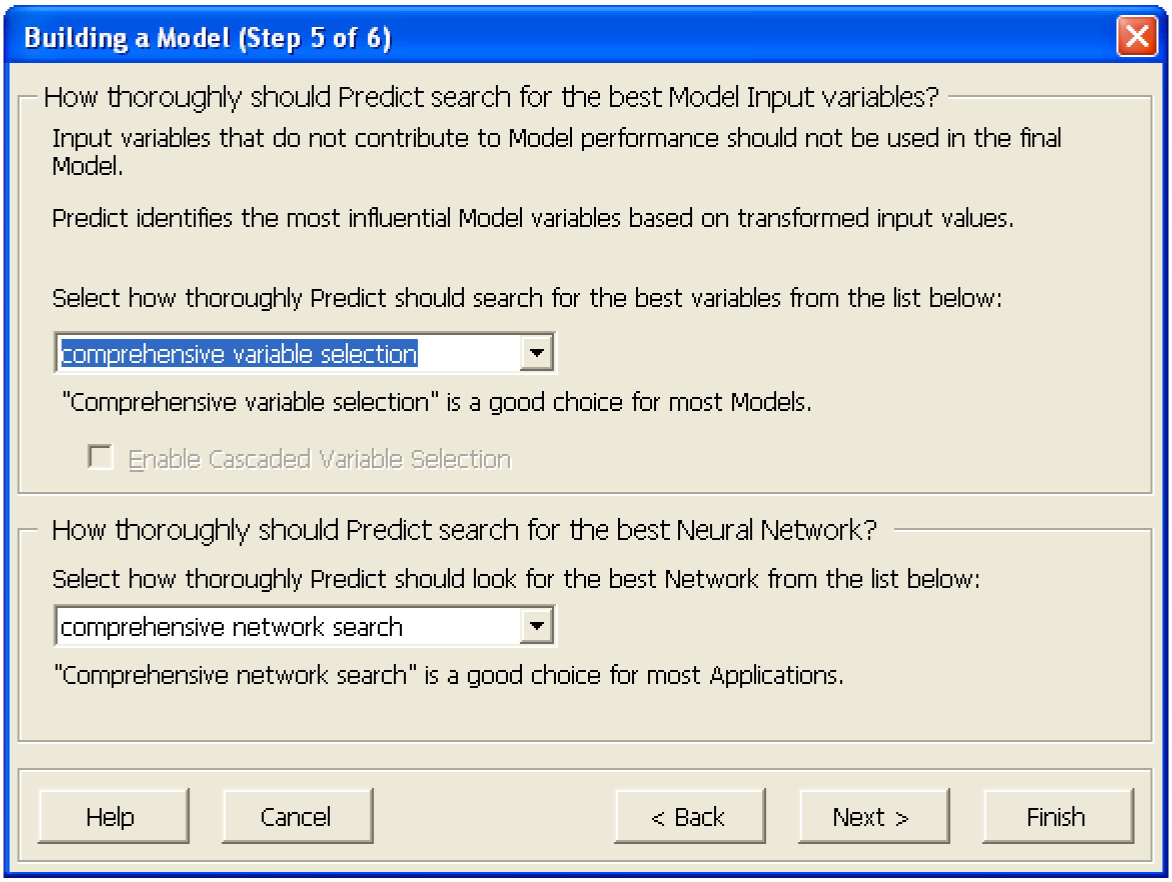


Click **Next.**

Building a model – Step 5

In this dialog box, you normally would consider and select options that tell ***Predict*** how input fields should be analyzed and how the actual neural network should be built.

Again, for this *Shortcut to Success* tour accept the default options.

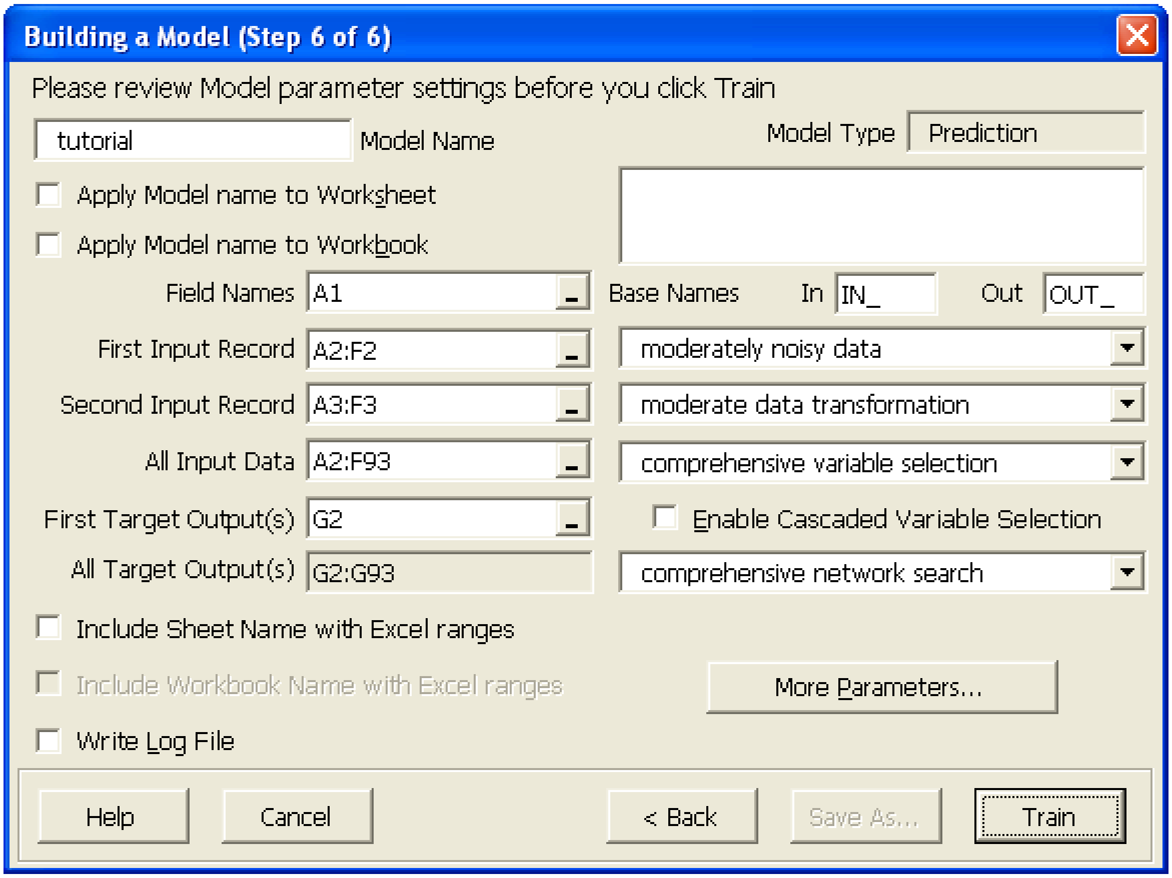


Click **Next**.

Building a model – Step 6

This dialog box is a summary of what you did in the previous 5 steps.

You do not need to make any further selections (you should confirm that your **Step 6 of 6** dialog box looks the same as this one).



1. Click **Train**.

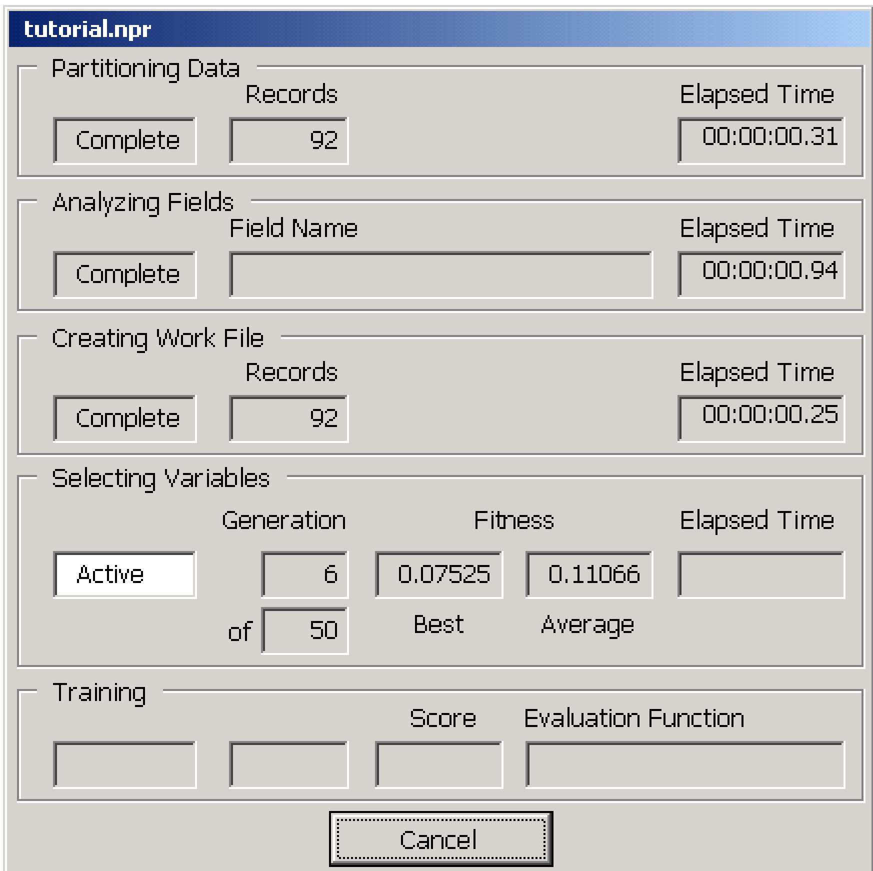
   A **Save Model** dialog box appears. However, since you are using demonstration software, your model will not actually be saved on disk.

2. Click **Save**.

Model Building Progress

After the **Save Model** dialog box closes, the model building progress dialog box immediately appears (very briefly!).

This dialog box provides a dynamic view of the major operations ***Predict*** performs as it first prepares to build, and then actually builds, a neural network model.



For the *tutorial* moisture evaporation model, the dialog box is visible for only a few seconds – which is all the time it takes to build the model!

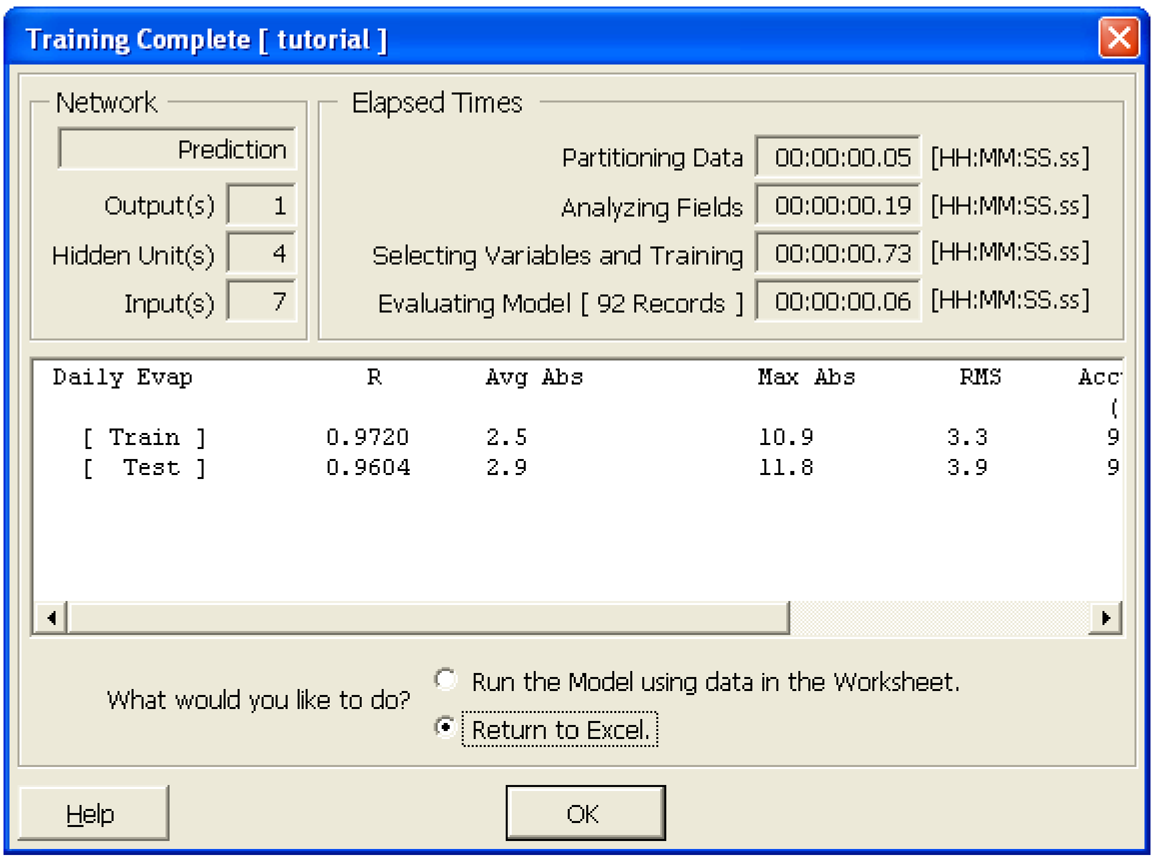
The dialog box closes automatically.

After Training Stops

After ***Predict*** has completed training the model, a dialog box containing summary statistics about the model and how well it performed on modeling data appears. Although a full explanation of the dialog box is beyond the scope of your *Shortcut to Success* tour, the next section describes some additional ***Predict*** features that help you assess the quality of models you build.

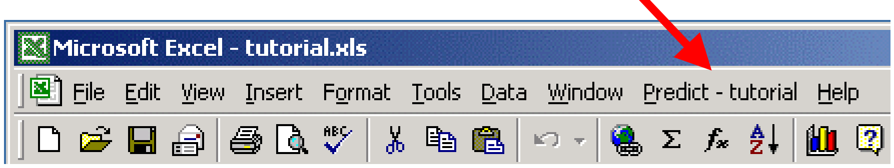
Please also note that in approximately 15 minutes, you have successfully used ***Predict*** to build a robust neural network model that solves a specific problem – and that you are now equipped to apply the power of ***Predict*** to build models using your own domain data.

Congratulations!



When you have finished viewing the information in the **Training Complete** dialog box, make sure that **Return to Excel** is selected, then click **OK**.

After the dialog box closes, the Excel spreadsheet is visible, exactly as it was before you built the model. However, note that the **Predict** menu in the Excel menu bar now shows the name of the just-completed model, as illustrated below:



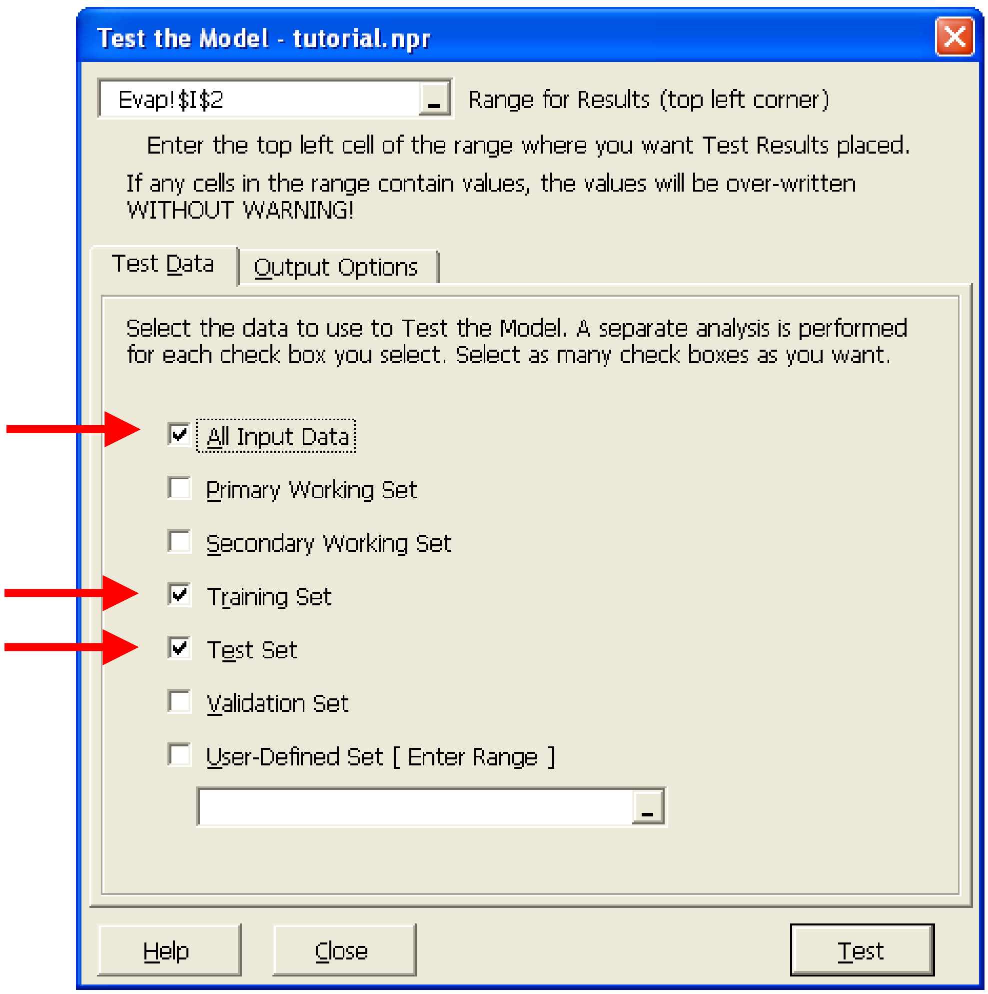
Testing a Model

One way to measure the quality of a model is to compute basic statistics related to its performance on the data used to train it. The ***Predict*** **Test** command generates such statistics.

1. In the **Predict** menu, click **Test**.

   The text box labeled **Range for Results (top left corner)** tells ***Predict*** where to begin writing the output generated by the **Test** command.

2. Click in the text box, and then click in one of the first cells in column **I** (it does not matter which cell you choose). However, please do *not* click in column **H** – in the next section we will use column **H** for other model output.



3. Select the following options if they are not already selected (and clear any others that are selected):
   - **All Input Data**
   - **Training Set**
   - **Test Set**

4. Click **Test**.

   Predict passes the data records associated with the options you selected above to the model and writes summary statistics to the spreadsheet, starting in the cell you specified.

5. Click **Close** to close the dialog box.

Interpreting Test Output

The **Test** command generates a table of performance metrics as illustrated below (the values written to your *tutorial* spreadsheet should be the same as these).

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 most important and useful metrics for a prediction model are usually the ***R*** *(Pearson R)* value, ***RMS*** *(Root Mean Square)* error, and ***Avg. Abs.*** *(Average Absolute)* error, although *Max. Abs. (Maximum Absolute)* error may sometimes be important.

Deciding which metric or metrics to rely on is important and somewhat problem dependent. In this *Shortcut to Success* tour, we will only briefly discuss the use of performance metrics. Predict has extensive and powerful features to help evaluate models and to improve model performance – they are fully described in the *Predict User Guide* that accompanies the full product.

The ***R*** value and ***RMS*** error indicate how "close" one data series is to another – in our case, the data series are the *Target* (actual) output values and the corresponding *predicted* output values generated by the model.

R values range from -1.0 to +1.0. A larger (absolute value) ***R*** value indicates a higher correlation. The sign of the ***R*** value indicates whether the correlation is positive (when a value in one series changes, its corresponding value in the other series changes in the same direction), or negative (when a value in one series changes, its corresponding value in the other series changes in the opposite direction). An ***R*** value of 0.0 means there is no correlation between the two series. In general larger positive ***R*** values indicate "better" models.

RMS error is a measure of the error between corresponding pairs of values in two series of values. Smaller ***RMS*** error values are better.

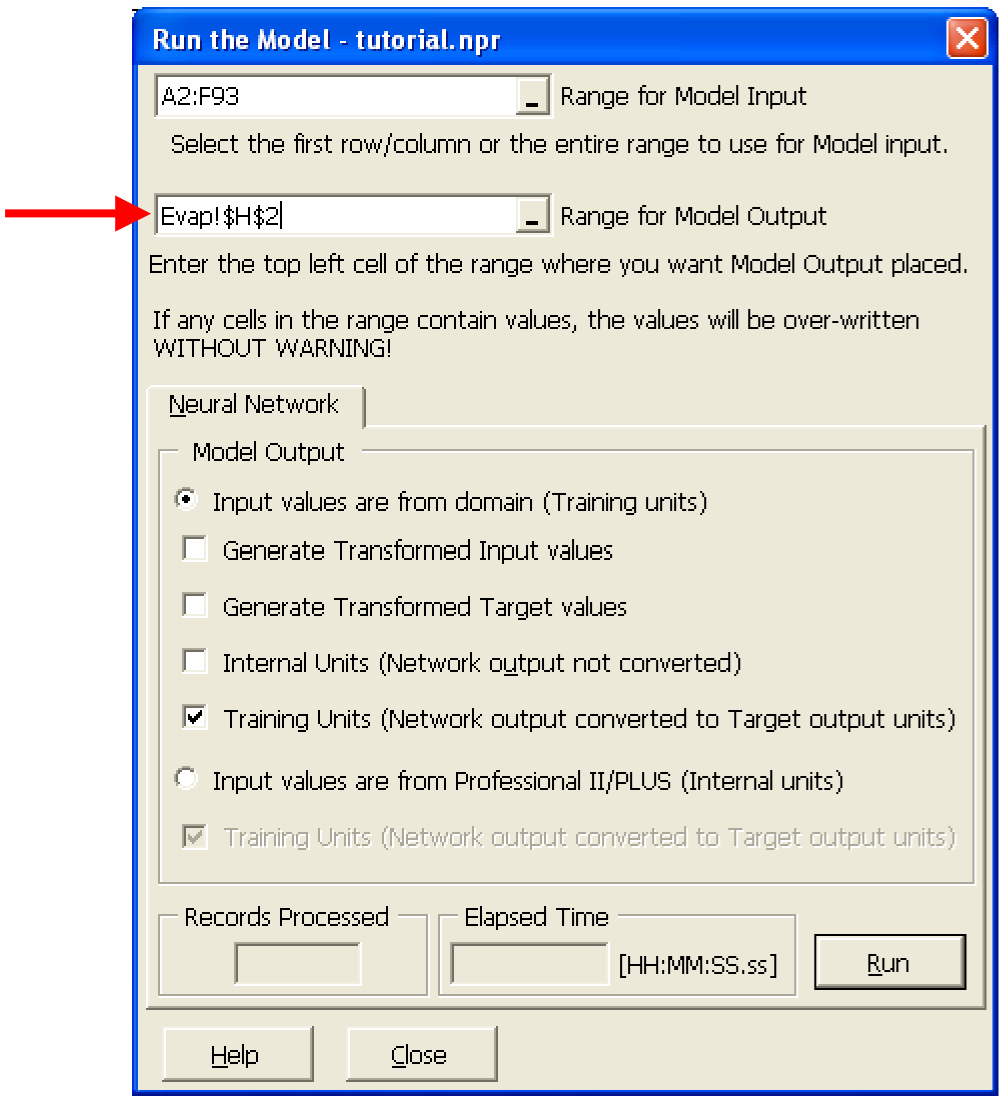
Finally, another key to using performance metrics is to compare the same metric computed for different datasets. Note the ***R*** values highlighted for the *Train* and *Test* sets in the above table. The relatively small difference between values (0.9719 and 0.9604) suggests that the model generalizes well and that it is likely to make accurate predictions when it processes new data (data not obtained from the *Train* or *Test* dataset).

Running a Model with Historical Data

In addition to using the **Test** command to compute summary statistics and performance metrics, you can also run a model with all of the historical (modeling) data. This allows comparing each *Target* value - *Predicted* value pair. The paired values can then be easily plotted to provide a quick visual indication of the quality of a model.

1. From the **Predict** menu, click **Run**.

   The **Run the Model** dialog box appears. By default, the text box labeled **Range for Model Input** is set to the Excel range that contains the original input data values.



2. If necessary, move the **Run the Model** dialog box so that cell **H2** is visible.

3. Click in the text box labeled **Range for Model Output**, and then click cell **H2**.

This action tells ***Predict*** where to write model predictions. Writing predicted values starting in cell **H2** will correctly align *predicted* outputs with actual (*target*) outputs to make comparisons easy.

4. Click **Run**.

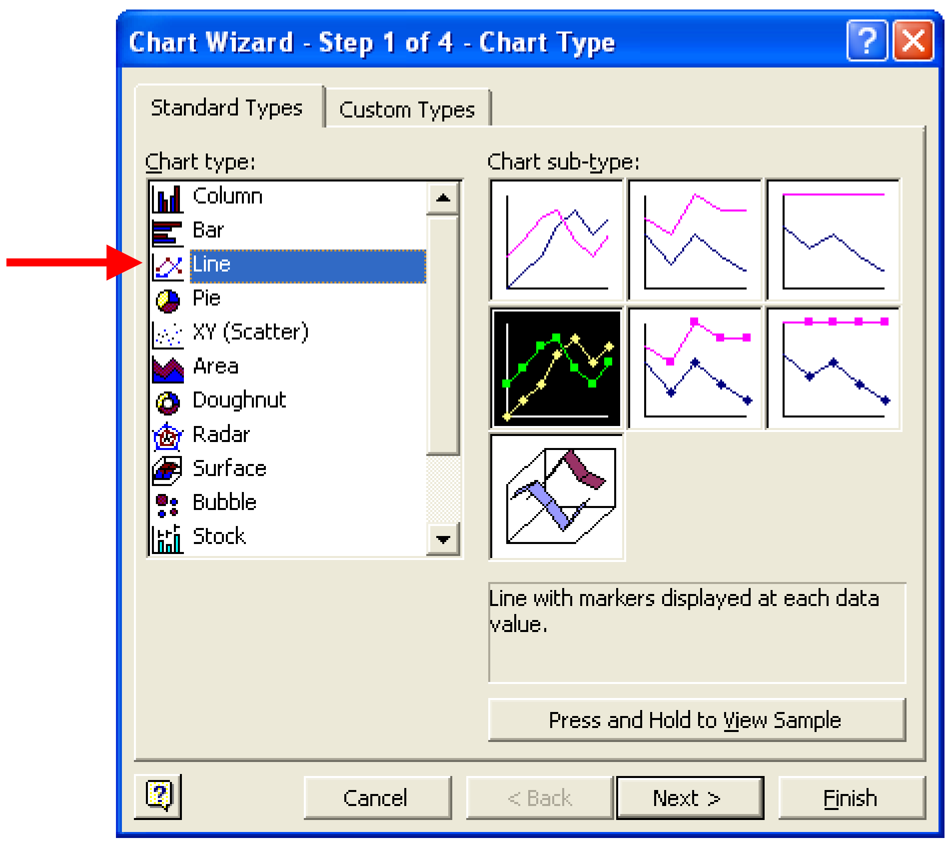
5. After the data are written to the spreadsheet, click **Close**.

Now that you have actual (*target*) output values and model *predicted* values aligned in the spreadsheet, it is a simple matter to create an Excel graph to visually assess the quality of the model.

1. First, select all the data in columns **G** and **H**.

2. From the Excel **Insert** menu, click **Chart**.

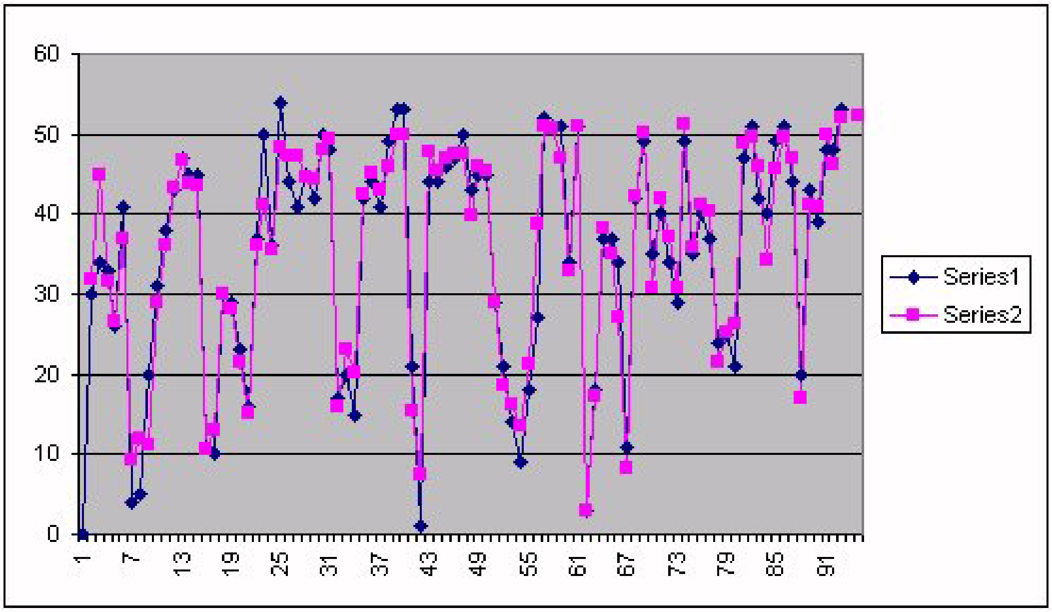
   The **Chart Wizard** dialog box appears:



3. In the **Chart Type** list, select **Line**.

4. Click **Finish**.

The plot Excel generates shows how close *predicted* outputs (pink) are to *target* outputs (dark blue).



Running a Model with New Data

Of course, the reason for building any model is so that output values can be accurately predicted when a new data record is processed by the model.

The following few actions are a simple demonstration of how to run a ***Predict*** model with new data. First, however, we need a "new" data record.

1. Enter the following 6 numeric values, which correspond to 6 input values for your daily evaporation rate model, in a horizontal row in the spreadsheet, starting with cell **A95** and ending with cell **F95** (these cells are in the area below the original data):

   100 150 200 50 400 400 (Press the **Enter** key or the **Tab** key after you type the last number into the spreadsheet)

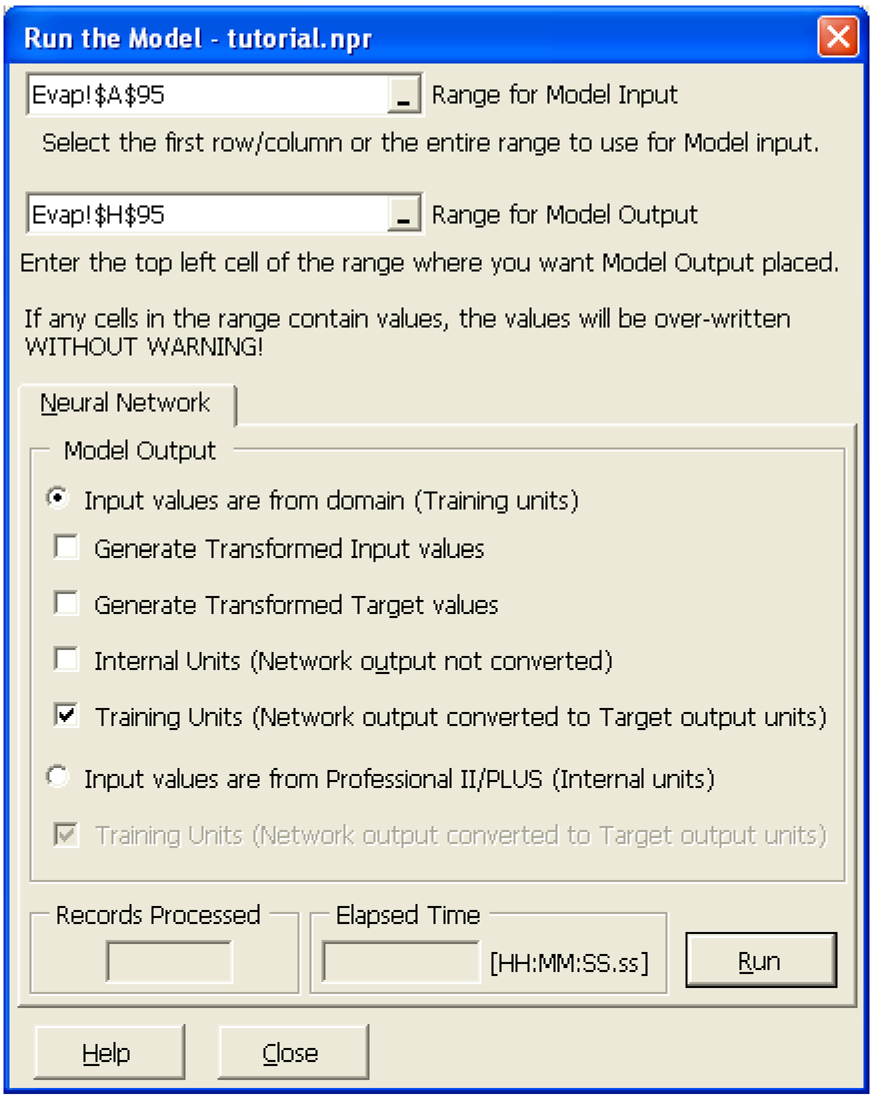
2. From the **Predict** menu, click **Run**.

   Once again, the **Run the Model** dialog box appears. However, this time you need to delete the default ranges in the two text boxes.

3. Drag the mouse to highlight the content of the text box labeled **Range for Model Input**, then click in cell **A95**.

4. Next, drag the mouse to highlight the content of the text box labeled **Range for Model Output**, then click in cell **H95**.

   When you have completed these steps, the **Run the Model** dialog box in your spreadsheet should appear as illustrated below; please confirm that the two ranges you entered are correct.



5. Click **Run**.

6. Click **Close**.

   Note that the model generates a *predicted* value (`52.21068`) in cell **H95**, based on the new data record you entered.

Congratulations again! You have successfully completed the entire *Shortcut to Success* tour!

In the **Predict** menu, click **Close** to close the model you built.

If you wish, for future reference you can save the spreadsheet with ***Predict's*** **Test** command results and the model's *predicted* outputs. From the Excel **File** menu, click **Save**.

Here's what you've accomplished

By completing the *Shortcut to Success*, you have produced and evaluated a real neural network model that accurately predicts the daily rate of moisture evaporation from soil. To achieve this, you:

- used Excel to load a dataset with historical information needed to build a model;
- configured ***Predict*** to build a neural network *prediction* model;
- trained the model using the historical data;
- tested the model's performance and recorded the results in a spreadsheet;
- ran the model using the historical data, and compared the model's *predicted* values with the known historical *target* values; and
- ran the model with new data to generate a predicted value from previously unseen data.

However, please keep in mind that you truly have only scratched the surface of the power of ***Predict***!

There is much more you can explore and put to use in your own modeling projects, including:

- Separating data into training, test, and independent validation datasets to assure robust generalization.
- Using ***Predict's*** comprehensive data transformation capabilities to automatically pre-process data for better distributions of values for modeling.
- Using ***Predict's*** powerful genetic algorithm-based variable selection mechanism to identify the most significant fields to employ in building the neural network model.
- Reviewing other presentations of model results, such as sensitivity analysis charts and detailed information about the final network topology and processing element weights.

What to do Next

We of course hope you are completely convinced of the power of ***Predict***, and are ready to buy!

If so, you can place your order

by phone: `+1 412.278.6288` (Toll free `+1 888.963.8766` in the US and Canada)

by e-mail: *sales@neuralware.com*

by FAX: `+1 412.278.6289(Attention: Sales)`

You can also learn more about ***Predict*** and its powerful capabilities through the *NeuralWorks Predict Getting Started Guide*. You can directly open the *Getting Started Guide* by from the Windows **Start** menu:

Programs **->** NeuralWare **->** NeuralWorks **->** Predict **->** *Getting Started Guide*.

In the *Getting Started Guide* you'll find additional tutorials that focus on other kinds of models and types of data, including symbolic character strings. There are a series of model development exercises that are based on problems associated with managing credit risk.

We encourage you to use ***Predict*** with your own data. We are confident you'll find that with very little effort you can import data into an Excel spreadsheet and build a solid baseline model in just a few minutes. And when you reach the point that the limitations of the demonstration version prevent you from achieving your modeling objectives, we are prepared to help you make the transition to the full Predict product release.

Finally, if you have questions concerning ***Predict*** and how to best apply it to your data modeling problems, please contact the NeuralWare product support staff:

by phone: `+1 412-278-6280 (option 2)`

by e-mail: *help@neuralware.com*.

Thank you for taking the *Shortcut to Success* tour – and we look forward to helping you meet the challenges you face in your analytic modeling projects.

The NeuralWare Team

Glossary

The following words and terms are frequently used in neural computing.

Accuracy (%) – The percentage of predicted outputs whose values fall within the user-specified tolerance band 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 (%) – Establishes with a specified degree of confidence the range (target value ± confidence interval) in which the corresponding predicted output occurs.

Correlation Coefficient *(see* **R value).**

Generalization – The ability of a model to produce a reasonable output when presented with input data that differs 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 that 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. Such data usually are organized in rows as records containing input variables followed by output variable(s). Historical data usually are divided into separate training, test, and validation sets.

Training and test sets, called modeling data, are used to build a model. 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). A true validation set is completely different from modeling data and is used to independently assess how well a model performs.

Input variables – The variables that serve as inputs for a 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 algorithm used during training to mathematically adapt the model.

Linear correlation value *(see* **R value).**

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 used to build a neural network model. This includes the training set and the test set.

Net-R – The linear correlation between the internally scaled representation of target values and raw network outputs.

Network output – Also called raw network output. Neural network output *before* it is transformed into values consistent with units of measure of the historical (problem) data. Network output values range either from -1.0 to 1.0, or from 0.0 to 1.0. Network outputs are then scaled by measurement units to yield predicted outputs.

Neural computing – The study of networks composed of many simple units called 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, regardless of the inputs.

Typical noise levels for various types of data are the following: mathematical function data are clean; behavioral data and most data generated by instrumentation are generally "moderately" noisy; securities data (stock prices, interest rates, etc.) are very noisy.

Over-fitting – Occurs when the mapping function that results from training a model fits the training set too well. An over-fit model does not generalize well when new data that was not represented in the training set is processed by the model.

Predicted output – The output of a neural network after it is transformed into the original measurement units of the application domain. *(Also refer to* Network output *and* Target value*)*.

R value – Also known as Pearson R. It is a measure of the linear relationship of two variables and ranges from 1.0 to +1.0. Perfect negative correlation is indicated by –1.0. Absence of correlation is indicated by 0. 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 output – The value in the historical data that is the recorded output for an instance of input variables. 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 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 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 test set and validation set, refer to the Train, Test and Validation Sets chapter 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 carious error metrics can be generated and evaluated. When a ***Predict*** model is being trained, the training is interrupted periodically and the model at that point 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. A network being trained also is 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.