ONLINE TUTORIAL T2: STATISTICA SOFTWARE PROJECT

This section is reproduced, with permission, from STATISTICA Software Tutorial. Students and professors using this book are eligible to receive a six-month license to use STATISTICA software for completion of exercises in Chapters 4 and 6. Request for this copy of the software is to be made by the instructor by completing the coupon available on www.prenhall.com/turban. Note that similar software projects can also be completed using tools identified in Technology Insights 6.4.

This section illustrates how a neural network application project is completed using commercial-grade software. STATISTICA provides a neural network module to build a neural network model from scratch and also an automated system called Intelligent Problem Solver to build a neural network model internally.

The sample data file to be used in this illustration, Heartdisease.sta (available on www.prenhall.com/turban), contains health-related data of males from a region of Western Cape, South Africa, known for elevated heart disease risk. There are two different cases of coronary heart disease (CHD) patients in this data file. Many of the CHD-positive men had undergone blood pressure reduction treatment and other programs to reduce their risk factors after their CHD event. In some cases, the measurements were taken after these treatments. For others, they were taken before treatment. The data discussed in this example represent a subset of cases from a larger data set, described in Rousseauw et al. (1983). This example illustrates how to use the Intelligent Problem Solver tool to identify the best neural network for predicting the possibility of CHD, first by learning underlying patterns from historical data and then predicting using measurable inputs that can be obtained from the patients.

There were 10 original variables in the sample data file; 7 new categorical variables were derived from those original variables, using the Recode function (these derived variables are marked as *new* in the table below).

The following variables are contained in the data file Heartdisease.sta:

Blood Pressure Level[a]	Systolic Blood Pressure[b]
Tobacco Intake Level[a]	Tobacco Cumulative Intake (kg)[b]
Cholesterol Level[a]	Low Density Lipoprotein Cholesterol[b]
Adiposity—Level of Fat Tissues[b]	Family History[c]
Stress Level Type A[a]	Stress Type A Behavior[b]
Obesity Level[a]	Obesity—Body Mass Index (BMI)[b]
Alcohol Intake Level[a]	Alcohol Consumption[b]
Age Range[a]	Onset of Disease—Age[b], Coronary Heart Disease[c]

[a]New variable and categorical variable
[b]Continuous variable
[c]Categorical variable

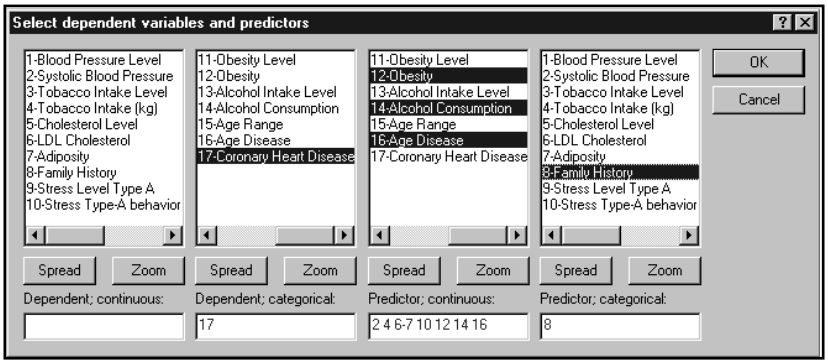
The number of cases (instances) was 463.

Continuous variables identified in parentheses.

All variables not marked as *Continuous* are categorical variables (e.g., all *New* variables as well as those containing two responses (*Present*, *Absent*).

1. Select all the continuous variables into the **Predictor; continuous** pane and the categorical variable **Family History** into **Predictor; categorical**; select **Coronary Heart Disease** as the **Dependent; categorical** variable (see Figure T2.1).

FIGURE T2.1



2. Click **OK** on the variable selection dialog and then click **OK** again on the **Select dependent variables and predictors** dialog to complete the selections.
3. Click the **Node Browser** button to display the browser and then open the folder **Classification and Discrimination** (see Figure T2.2).
4. Double-click **Split Data into Training and Testing Samples (Classification)** to attach the node to the SDM workspace.
5. Double-click **Split Data into Training and Testing Samples (Classification)** node that appears in the SDM workspace to view the **Edit Parameters** dialog (see Figure T2.3).
6. Set **Approximate percent of cases for testing:** to **20** and click **OK** to complete the parameter settings.

FIGURE T2.2

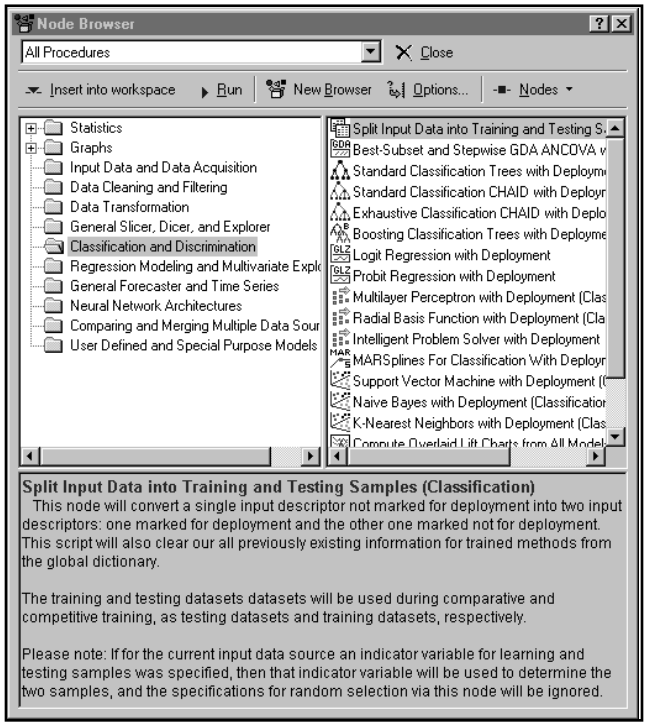
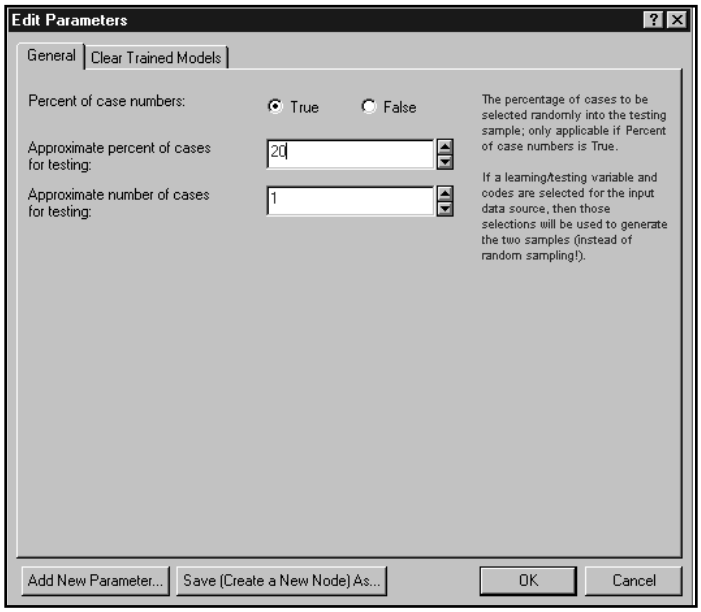


FIGURE T2.3



7. Right-click the **Split Data into Training and Testing Samples (Classification)** node and select the **Run to Node** option from the shortcut menu. Figure T2.4 shows how the Data Miner workspace should look now.
8. Select or highlight the **Training Data** and **Testing Data** nodes simultaneously while pressing **Ctrl** button on your keyboard for auto connection.
9. Click the **Node Browser** button. Open the folder **Classification and Discrimination** (see Figure T2.5).
10. Double-click **Intelligent Problem Solver** to attach the node to the **SDM** workspace. Close the **Node Browser**.
11. Double-click the **Intelligent Problem Solver** node to view the **Edit Parameter** dialog.
12. Select the **All results** option from the **Detail of computed results reported** menu (on the **General** tab) and then select the **Deployment** tab (see Figure T2.6).
13. Set the **True** option for **Generate PMML code:** and click **OK** to complete the parameter settings.

FIGURE T2.4

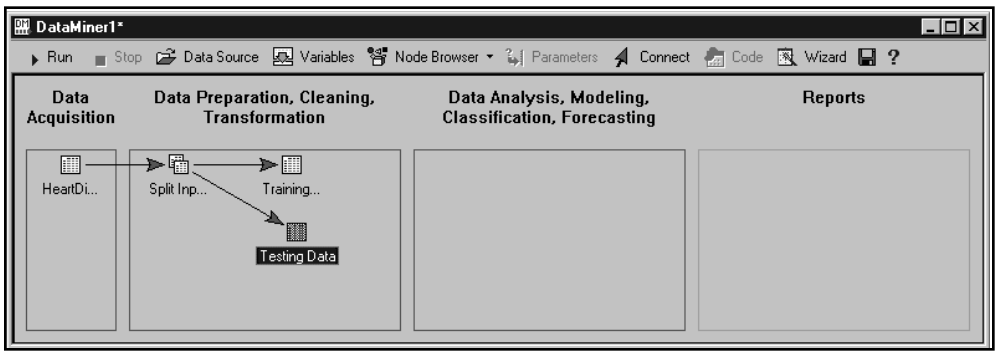
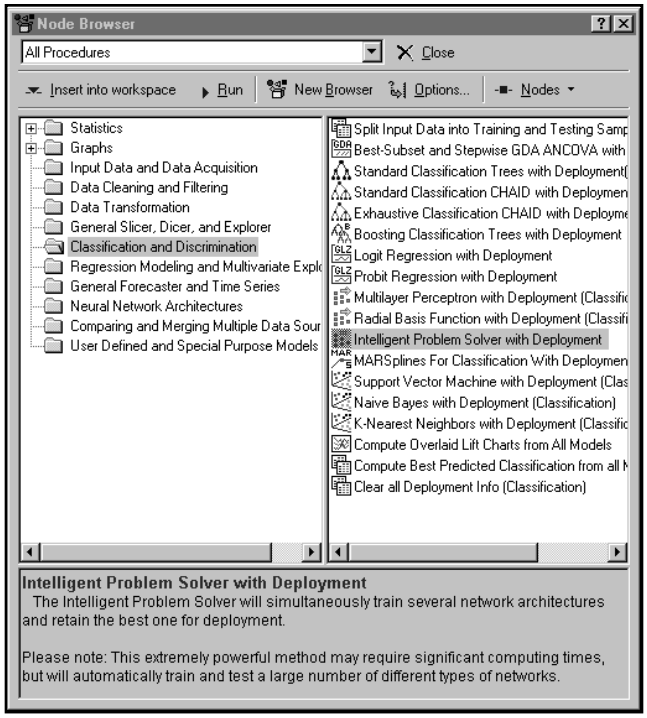


FIGURE T2.5



14. Right-click **Intelligent Problem solver with Deployment** and select the **Run to Node** option from the shortcut menu. Figure T2.7 shows how the Data Miner workspace should look now.
15. Double-click the report workbook **Intelligent Problem solver with Deployment** to view the results of the analysis. Select **Model Summary Report** (see Figure T2.8).

FIGURE T2.6

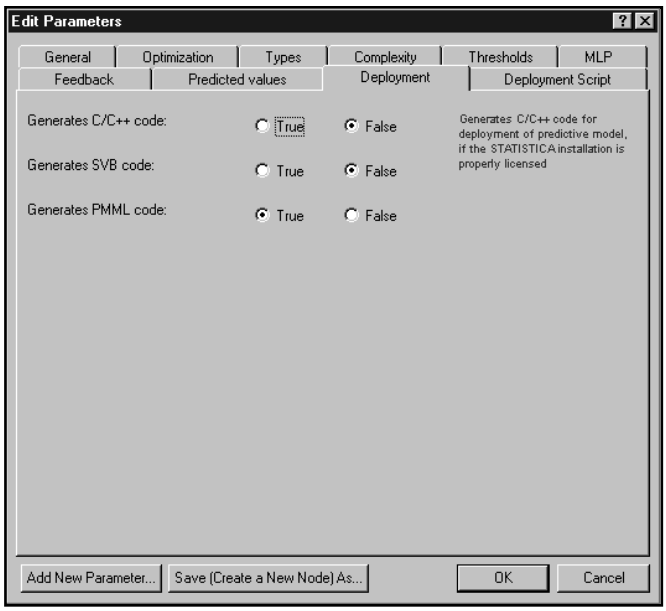
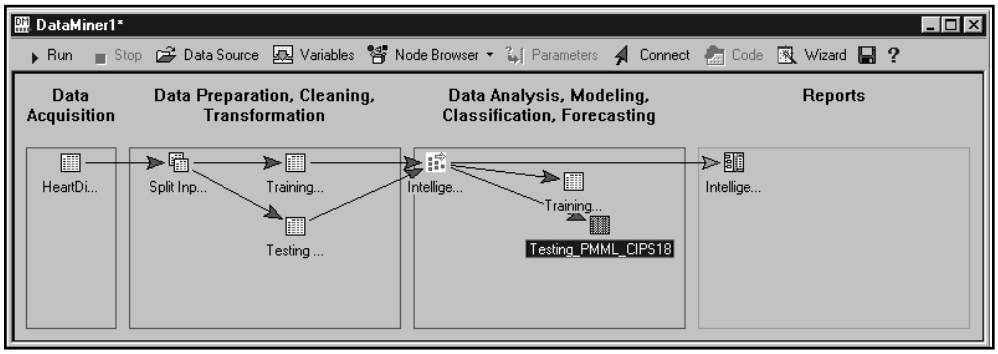


FIGURE T2.7



Note that the results you see on your screen may be somewhat different from what is shown here because of the different training samples generated each time the data source node is split for training and testing. The Intelligent Problem Solver can simultaneously train several network architectures and retain the best one for deployment. The model summary report shows that the Intelligent Problem Solver chose as the best solution a linear neural network, which implements a basic linear model.

The spreadsheet also provides other results regarding the performance of the network during the training process.

TRAIN PERF/SELECT PERF/TEST PERF

The values in the columns **Train Perf**, **Select Perf**, and **Test Perf** indicate the network performance on the subsets of data used during training. During the training of the neural networks, the input data are divided into three subsamples: a training sample, a testing sample, and a selection sample. Note that the testing sample that was previously created via the Split Input node in the Data Miner workspace is not used during the network training at all, and hence, these observations represent a true holdout sample that can be used to compare the performance of different types of models.

During the estimation of the different neural networks models, the training sample is used to estimate the parameters of the respective neural network (over successive iterations). The selection sample data are used to evaluate the performance (i.e., accuracy) of the networks, optimized for the training data, for an independent sample of "new cases." Specifically, the particular network that is selected by the Intelligent Problem Solver procedure as the "best" network is the one that generates the best performance for the selection sample. Finally, as a sanity check, the program also computes the performance indices for the testing sample, which represents a true holdout sample that was not used at all during the estimation and fitting of the neural networks.

The specific indices that are computed to summarize the performance or accuracy of the final best neural network depend on the type of analytic problem. For classification tasks such as the one described in this example, the network performance measure represents the accuracy of the network for predictive classification (i.e., the proportion of cases accurately classified in the respective sample by the final best network).

FIGURE T2.8

Data: Model Summary Report (Split Input Data into Training and Testing Samples (Classification))*

Model Summary Report (Split Input Data into Training and Testing Samples (Classification))

Index	Profile	Train Perf.	Select Perf.	Test Perf.	Train Error	Select Error	Test Error	Training/Members	Note	Inputs	Hidden(1)	Hidden(2)
1	Linear 9:9-1:1	0.731579	0.744681	0.659574	0.422011	0.384518	0.411835	PI		9	0	0

TRAIN ERROR/SELECT ERROR/TEST ERROR

The values reported in the columns **Train Error**, **Select Error**, and **Test Error** indicate the error of the network when predicting (i.e., classifying) the respective subsets of cases. To learn more about the specific computations of these values, please refer to the **Error Function** topic for **STATISTICA Neural Networks** in the Electronic Manual.

OTHER INFORMATION

The other columns in this results spreadsheet report additional details describing the nature of the final best neural network for these data and details regarding the estimation procedure. Again, refer to the detailed descriptions of these results in the Electronic Manual to learn more about the information that is reported.

PROPORTION OF CORRECTLY AND INCORRECTLY CLASSIFIED CASES

Let us next review the spreadsheet that reports the overall accuracy of the final neural network, for all observations in the input data (in the training sample created in the **Split Input** node).

When you select the spreadsheet named **Classification**, the window shown in Figure T2.9 appears.

This spreadsheet shows that out of the 129 cases (participants in the study) who experienced CHD, 70.54 percent were correctly classified (i.e., predicted) by the linear neural network. Of the 249 observations that did not experience coronary heart disease, 72.28 percent were correctly classified (i.e., predicted).

Again, these percentages pertain to the accurate or mistaken classifications for all observations predicted from the final best neural network. Later, we will further test the predictive accuracy of the network by computing predictions for the testing sample that was created in the **Split Input** node we ran earlier.

PMML CODE

Let us next review the deployment code that was generated. When you select the report file STATISTICA Predictive Model Markup Language Code, the window shown in Figure T2.10 appears.

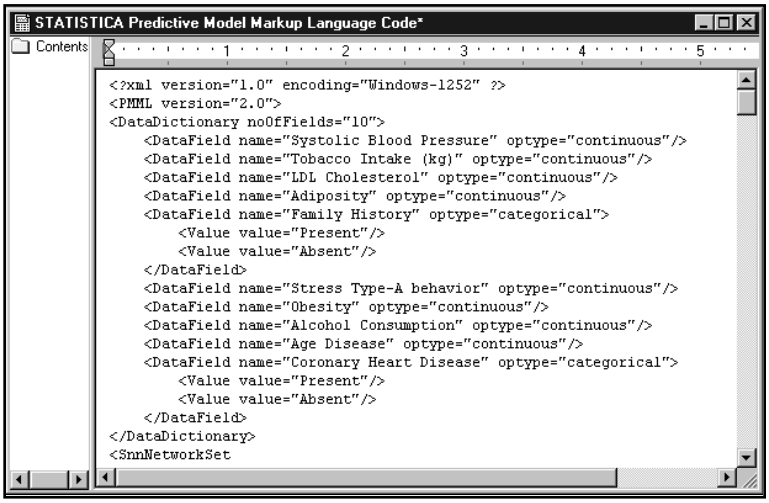
PMML, which stands for *Predictive Model Markup Language*, is an XML-based language that allows for the efficient exchange of (trained) predictive models and shared models between different applications (e.g., STATISTICA and WebSTATISTICA). A PMML document usually contains information that describes fully trained or parameterized analytic models so that they can be readily deployed (i.e., applied to new cases) by another application. PMML documents can be saved from practically all methods

FIGURE T2.9

Data: Classification (1) (Split Input Data into...

	Classification (1) (Split Input Data into T	
	Coronary Heart Disease.Present.1	Coronary Heart Disease.Absent.1
Total	129.0000	249.0000
Correct	91.0000	180.0000
Wrong	38.0000	69.0000
Unknown	0.0000	0.0000
Correct(%)	70.5426	72.2892
Wrong(%)	29.4574	27.7108
Unknown(%)	0.0000	0.0000

FIGURE T2.10



available in STATISTICA for prediction and predictive classification, and they are used extensively in the context of STATISTICA Data Miner and WebSTATISTICA.

EVALUATING THE ACCURACY OF THE FINAL NETWORK IN THE TESTING (HOLDOUT) SAMPLE

This PMML code will automatically be used by the Intelligent Problem Solver with the Deployment node in the Data Miner workspace to compute predictions (i.e., predicted classifications) for the cases in the testing data set. Recall that the observations in that sample have not been used yet for any computations and in particular were not used during the estimation and fitting of the different neural networks (in order to find the best one). Hence, the observations in the testing sample represent a true holdout sample, which we can use to assess the predictive accuracy of the final best neural network.

Follow the steps below to test the accuracy of the trained network:

1. Close the report workbook and right-click the **Testing_PMML_CIPS** node in the **Data Analysis** pane. Select **View Document** from the shortcut menu. Figure T2.11 shows the predicted classifications and accuracy for the observations in the testing sample.

   The first column of the spreadsheet shows the model predictions or predicted classifications; the second column indicates whether the respective cases were predicted accurately by the model; and the third column shows the actual observed outcome (i.e., the observed presence or absence of Coronary Heart Disease).
2. To visually summarize the predictive accuracy of the neural network, highlight the column **Linear-9:2:1-PlRes**, as shown in Figure T2.11.
3. Click the **Graphs** option from the menu and select the **Histogram** option.
4. Select the **Advanced** tab and then click the **Variables** button. Select **Linear-9:2:1-PlRes** and then click **OK**.
5. Check the **Show Percentage** option shown on the left side. Then click **OK** to view the resulting histogram (see Figure T2.12).

Overall, the performance of the neural network seems quite good. The neural network correctly classified 70 percent of the observations in the holdout (testing) sample. Note that you might want to compute additional, more detailed, tables by

FIGURE T2.11

Data: Testing_PMML_CIPS18 (3v by 84c)

Input spreadsheet generated from deployment

	1 Linear-9:2:1-PIPred	2 Linear-9:2:1-PIRes	3 Coronary Heart Disease
1	Present	Correct	Present
2	Present	Correct	Present
3	Present	Correct	Present
4	Present	Incorrect	Absent
5	Absent	Correct	Absent
6	Present	Correct	Present
7	Present	Incorrect	Absent
8	Present	Correct	Present
9	Absent	Incorrect	Present
10	Absent	Correct	Absent
11	Present	Incorrect	Absent
12	Absent	Correct	Absent
13	Absent	Incorrect	Present
14	Present	Correct	Present
15	Present	Incorrect	Absent
16	Absent	Correct	Absent
17	Present	Correct	Present

using the **Basic Statistics** options for the predictions computed by the neural network. Figure T2.13 shows an example.

This detailed two-way frequency table of the observed and predicted classifications for the testing sample cases shows how well the final neural network performs. Of

FIGURE T2.12

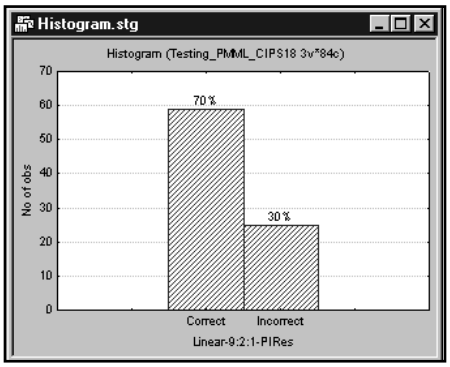


FIGURE T2.13

Data: 2-Way Summary Table: Observed Frequencies (Te...

2-Way Summary Table: Observed Frequencie
Marked cells have counts > 10

Linear-9:2:1-PIPred	Coronary Heart Disease Absent	Coronary Heart Disease Present	Row Totals
Absent	36	8	44
Column %	67.92%	25.81%	
Row %	81.82%	18.18%	
Total %	42.86%	9.52%	52.38%
Present	17	23	40
Column %	32.08%	74.19%	
Row %	42.50%	57.50%	
Total %	20.24%	27.38%	47.62%
Totals	53	31	84
Total %	63.10%	36.90%	100.00%

those individuals with CHD present, the neural network accurately classified 74.19 percent (as Present). Looking at it from the perspective of risk, given the prediction of heart disease, 57.50 percent of the cases that were predicted to show heart disease (the second row in the table) actually fell into that category. Hence, the chance (or risk) of an individual who is classified as Heart Disease Present by the neural network to actually show (develop) heart disease is 57.5 percent.

References

Rousseauw, J., du Plessis, A. Benade, P. Jordann, J. Kotze, P. Jooste, and J. Ferreira. "Coronary Risk Factor Screening in Three Rural Communities." *South African Medical Journal*, 64:430–436, 1983.