Data Mining – LAB 4

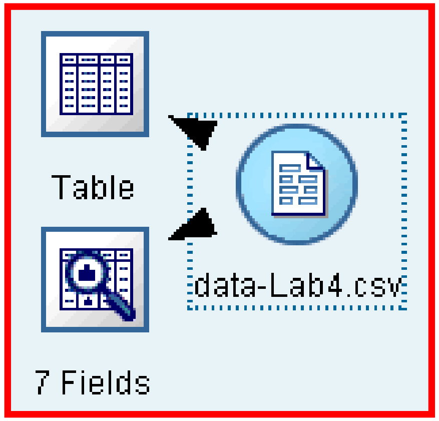
Classification – Neural Network

Data File

- Download from:
  - www.comp.polyu.edu.hk/~csamak/data/data-Lab4.csv
- For Virtualbox image, it is placed at:
  - **`C:\Data\data-Lab4.csv`**

Data Understanding

- Load the data file (data-Lab4.csv) into PASW

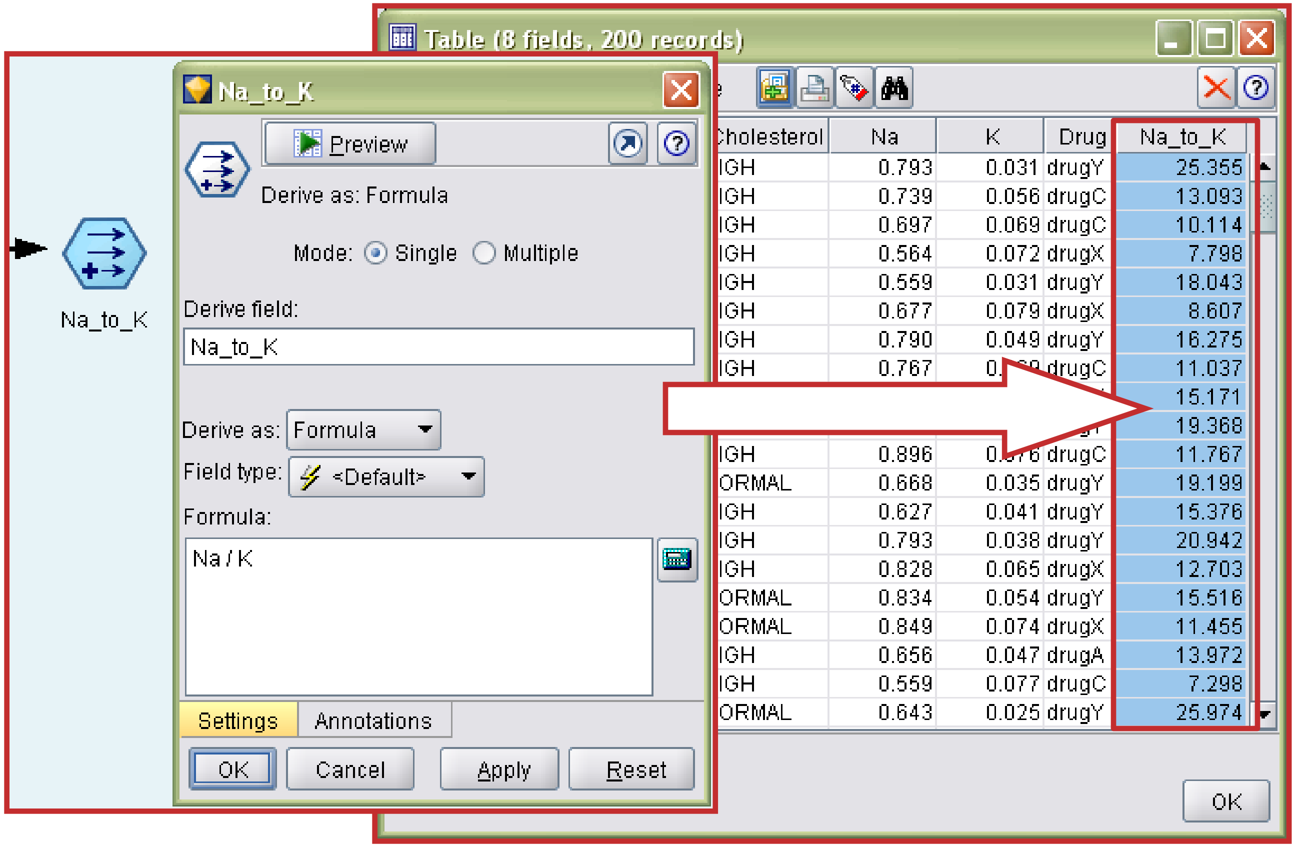


Think about these:

1. How many attributes are being used in the dataset?
2. How many records are stored in the dataset?
3. Are there problems with the dataset?

Data Preparation - Transformation

- “Derive” a new field “Na_to_K” (ratio of Na to K, ie Na/K)



Na_to_K

Derive as: Formula

Mode: ● Single ○ Multiple

Derive field: Na_to_K

Derive as: Formula

Field type: <Default>

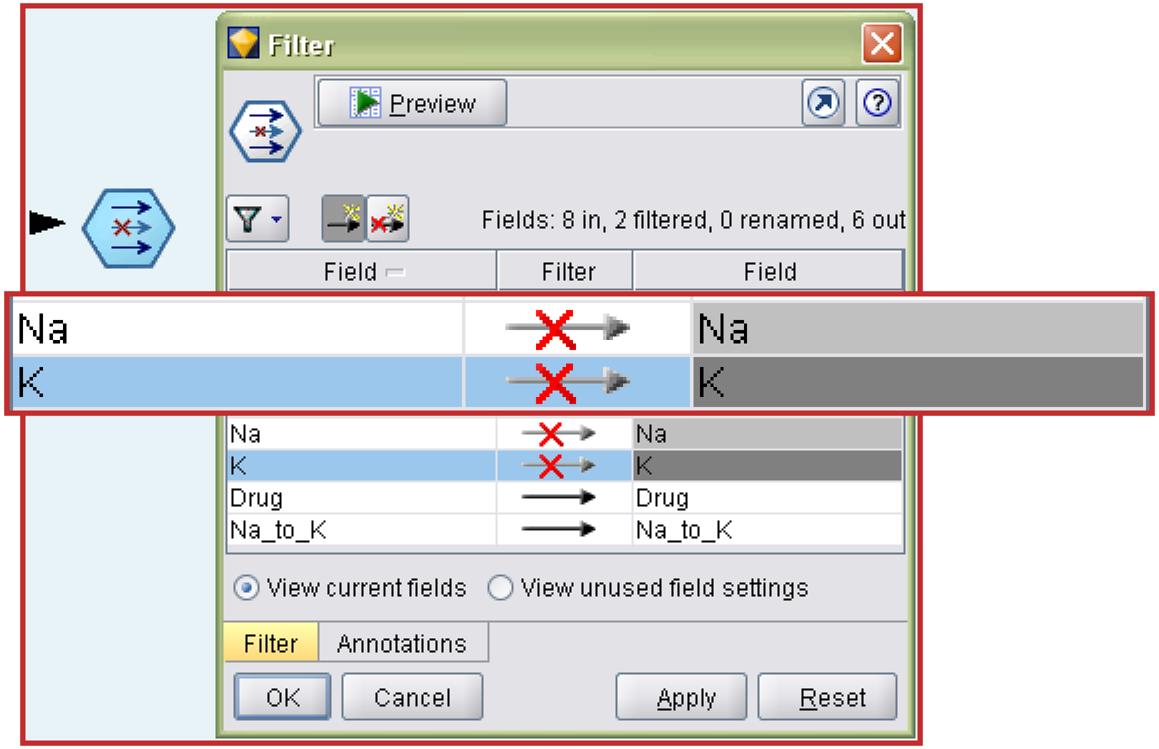
Formula: Na / K

Table (8 fields, 200 records)

Cholesterol	Na	K	Drug	Na_to_K
IGH	0.793	0.031	drugY	25.355
IGH	0.739	0.056	drugC	13.093
IGH	0.697	0.069	drugC	10.114
IGH	0.564	0.072	drugX	7.798
IGH	0.559	0.031	drugY	18.043
IGH	0.677	0.079	drugX	8.607
IGH	0.790	0.049	drugY	16.275
IGH	0.767	0.[illegible]	drugC	11.037
[illegible]	[illegible]	[illegible]	[illegible]	15.171
[illegible]	[illegible]	[illegible]	[illegible]	19.368
IGH	0.896	0.[illegible]76	drugC	11.767
ORMAL	0.668	0.035	drugY	19.199
IGH	0.627	0.041	drugY	15.376
IGH	0.793	0.038	drugY	20.942
IGH	0.828	0.065	drugX	12.703
ORMAL	0.834	0.054	drugY	15.516
ORMAL	0.849	0.074	drugX	11.455
IGH	0.656	0.047	drugA	13.972
IGH	0.559	0.077	drugC	7.298
ORMAL	0.643	0.025	drugY	25.974

Data Preparation - Transformation

- Use "Filter" node to discard the fields "Na" and "K"



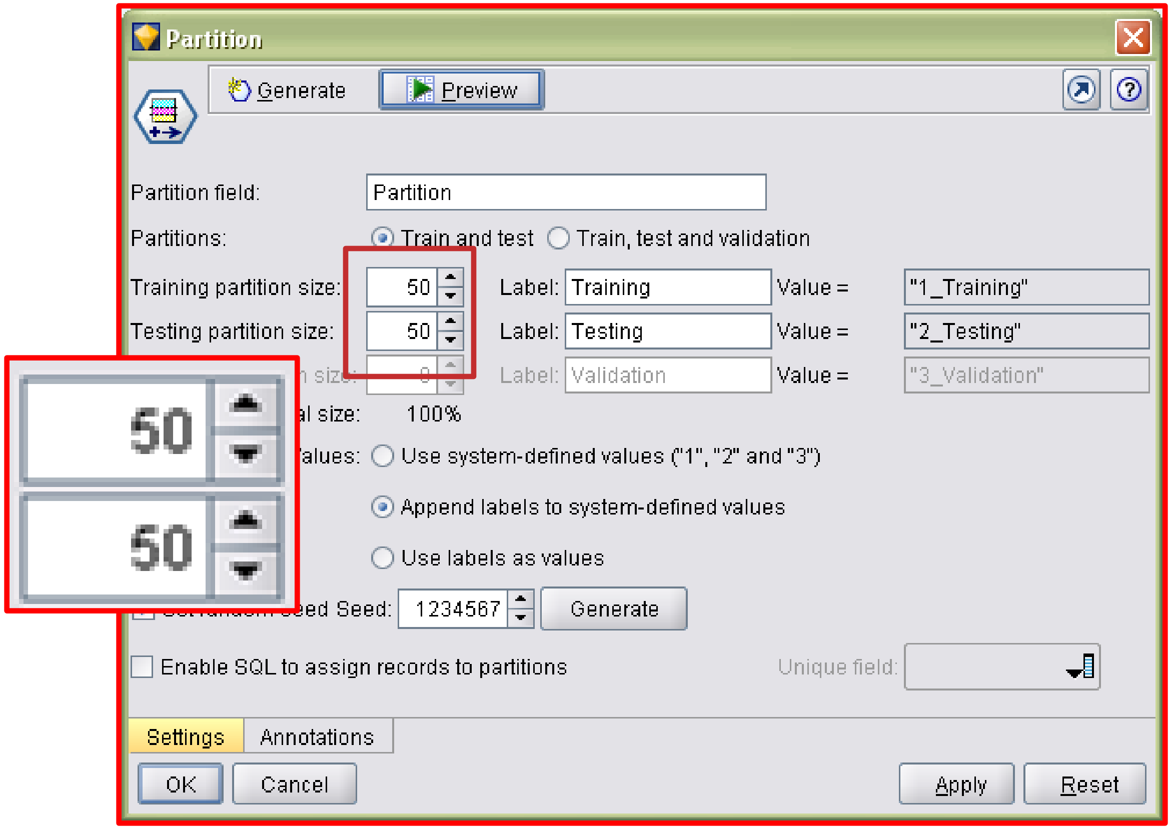
Fields: 8 in, 2 filtered, 0 renamed, 6 out

Field	Filter	Field
Na	✗→	Na
K	✗→	K
Drug	→	Drug
Na_to_K	→	Na_to_K

● View current fields ○ View unused field settings

Data Preparation

- Add "Partition" node to divide the dataset into two, Training and Testing, in 50/50.



Partition field: Partition

Partitions: ● Train and test ○ Train, test and validation

Training partition size: 50 Label: Training Value = "1_Training"

Testing partition size: 50 Label: Testing Value = "2_Testing"

n size: 0 Label: Validation Value = "3_Validation"

al size: 100%

alues: ○ Use system-defined values ("1", "2" and "3")

● Append labels to system-defined values

○ Use labels as values

ed Seed: 1234567

□ Enable SQL to assign records to partitions

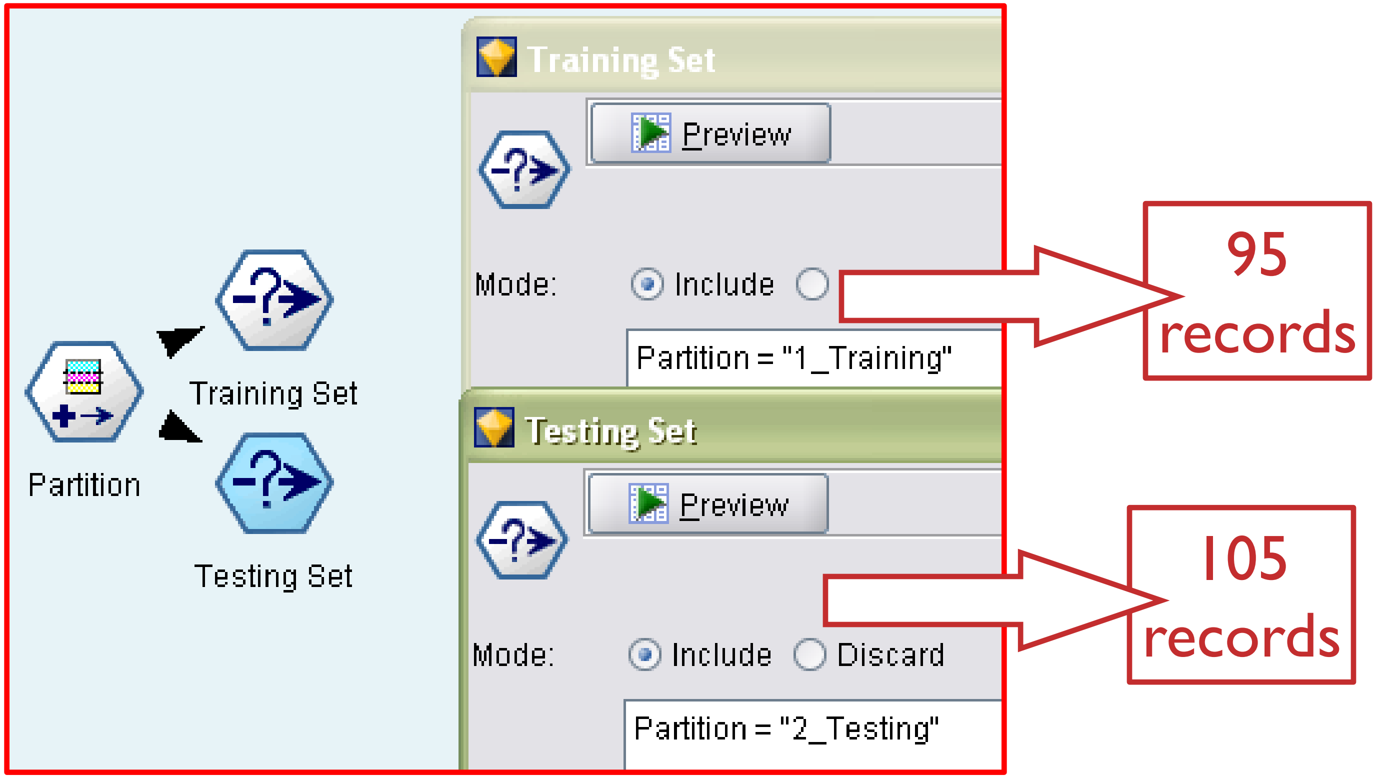
Unique field:

50

50

Data Preparation

- Use two “Select” nodes to get the records



Training Set

Mode: ● Include ○

Partition = "1_Training"

Testing Set

Mode: ● Include ○ Discard

Partition = "2_Testing"

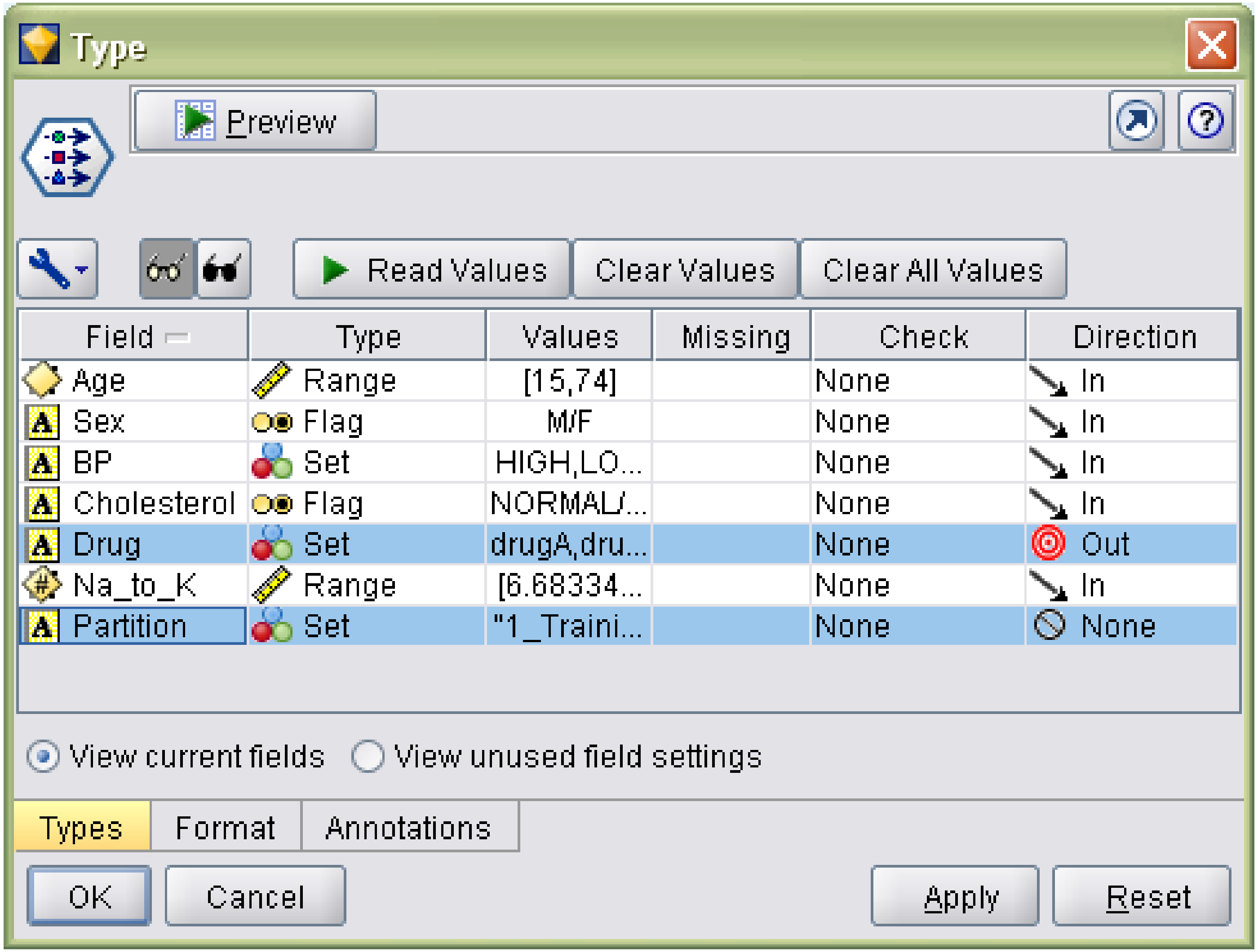
95 records

105 records

Prepare to Build the Model

- Use a "Type" node to refresh fields' value in application's memory
- Set the **INPUT** ➔ Age, Sex, BP, Cholesterol, Na_to_K
- Set the **OUTPUT** ➔ Drug
- Select "NONE" for the field, Partition

Prepare to Build the Model

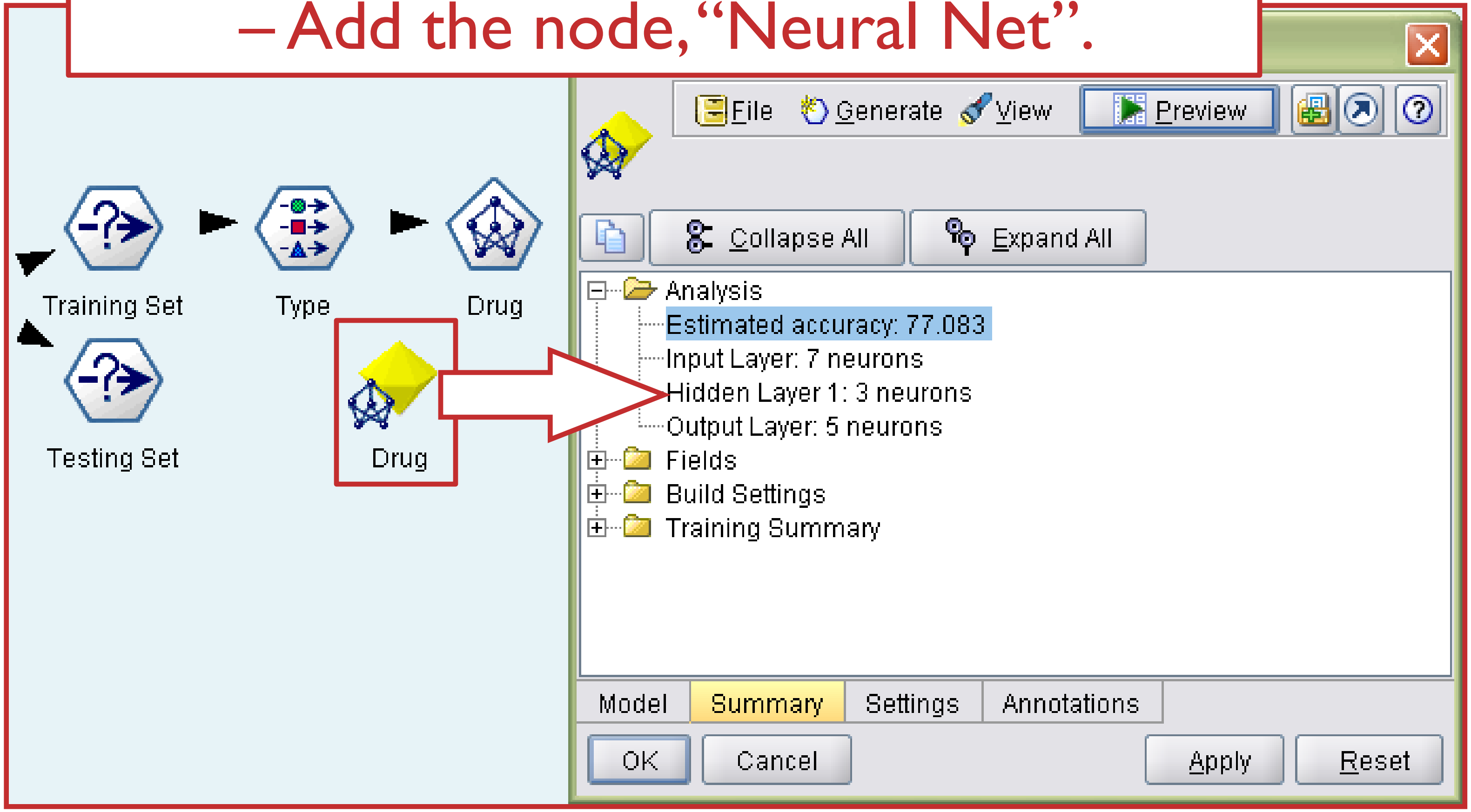


Field	Type	Values	Missing	Check	Direction
Age	Range	[15,74]		None	In
Sex	Flag	M/F		None	In
BP	Set	HIGH,LO...		None	In
Cholesterol	Flag	NORMAL/...		None	In
Drug	Set	drugA,dru...		None	Out
Na_to_K	Range	[6.68334...		None	In
Partition	Set	"1_Traini...		None	None

● View current fields ○ View unused field settings

Neural Net Model

Classification for attribute, "Drug" – Add the node, "Neural Net".



Analysis
- Estimated accuracy: 77.083
- Input Layer: 7 neurons
- Hidden Layer 1: 3 neurons
- Output Layer: 5 neurons

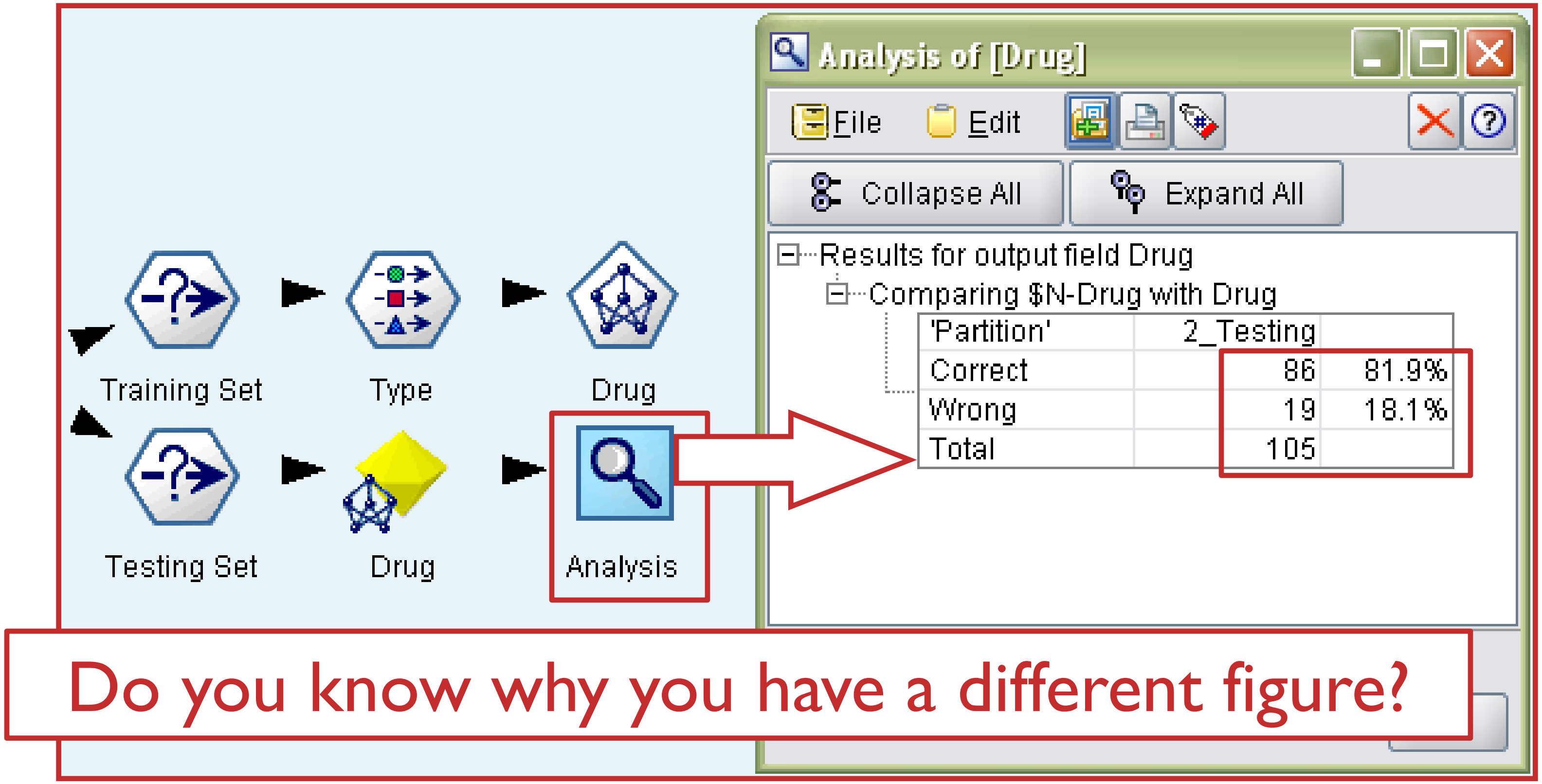
Fields

Build Settings

Training Summary

Neural Net Model

- Link the model to "Testing Set" and add a node, "Analysis" to check the model



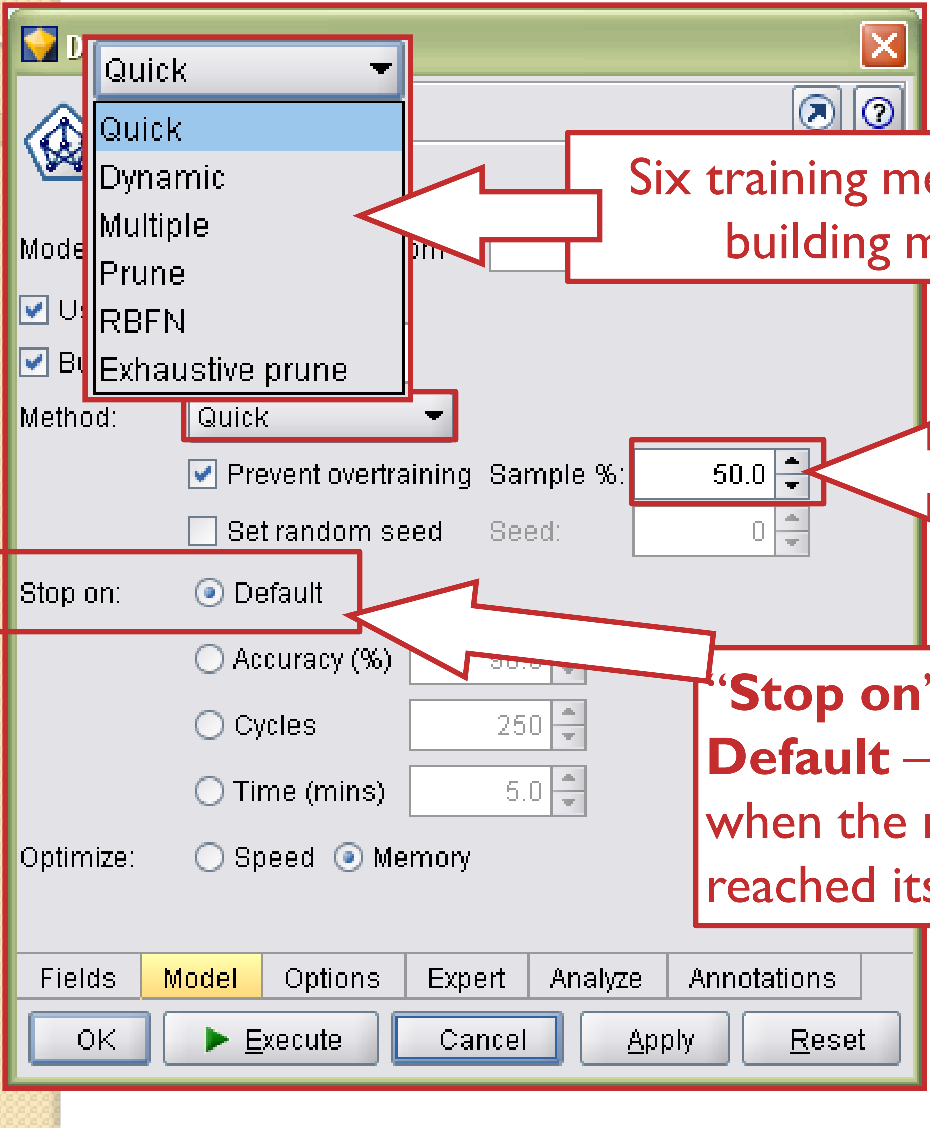
Results for output field Drug

Comparing $N-Drug with Drug

'Partition'	2_Testing	
Correct	86	81.9%
Wrong	19	18.1%
Total	105	

Do you know why you have a different figure?

Things to Note



Quick

- Quick
- Dynamic
- Multiple
- Prune
- RBFN
- Exhaustive prune

Method: Quick

☑ Prevent overtraining Sample %: 50.0

□ Set random seed Seed: 0

Stop on: ● Default

○ Accuracy (%)

○ Cycles 250

○ Time (mins) 5.0

Optimize: ○ Speed ● Memory

Six training methods for building model.

Randomly splits the data into separate training and testing sets for purposes of model building

"**Stop on**" – Stopping criteria
Default – the network stops training when the network appears to have reached its optimally trained state.

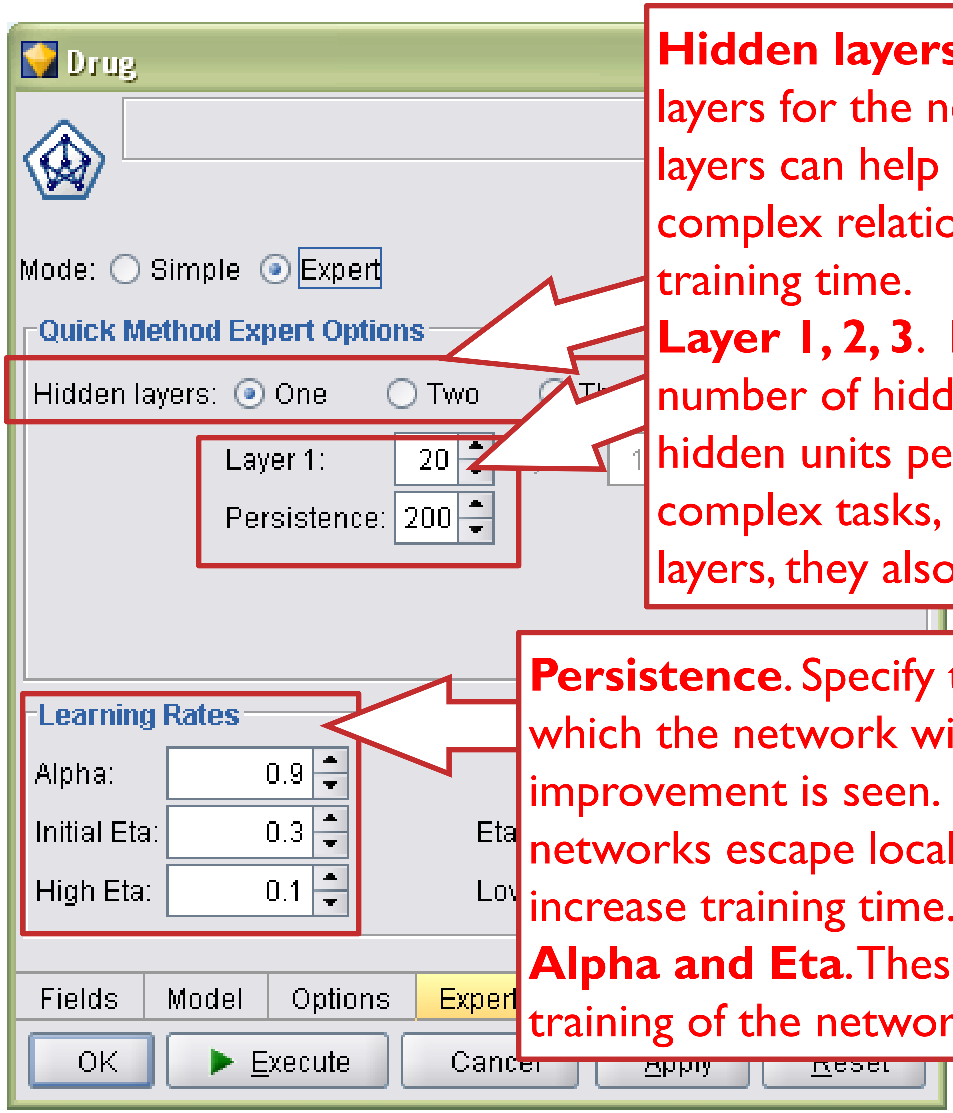
The Six Training Methods

- **Quick**. This method uses rules of thumb and characteristics of the data to choose an appropriate shape (topology) for the network.
- **Dynamic**. This method creates an initial topology but modifies the topology by adding and/or removing hidden units as training progresses.
- **Multiple**. This method creates several networks of different topologies (the exact number depends on the training data). These networks are then trained in a pseudo-parallel fashion. At the end of training, the model with the lowest RMS error is presented as the final model.
- **Prune**. This method starts with a large network and removes (prunes) the weakest units in the hidden and input layers as training proceeds. This method is usually slow, but it often yields better results than other methods.
- **RBFN**. The radial basis function network (RBFN) uses a technique similar to k-means clustering to partition the data based on values of the target field.
- **Exhaustive prune**. This method is related to the Prune method. It starts with a large network and prunes the weakest units in the hidden and input layers as training proceeds. With Exhaustive Prune, network training parameters are chosen to ensure a very thorough search of the space of possible models to find the best one. This method is usually the slowest, but it often yields the best results. Note that this method can take a long time to train, especially with large datasets.

Other Stop Options

- **Accuracy** (%). With this option, training will continue until the specified accuracy is attained. This may never happen, but you can interrupt training at any point and save the net with the best accuracy achieved so far.
- **Cycles**. With this option, training will continue for the specified number of **cycles** (passes through the data).
- **Time** (mins). With this option, training will continue for the specified amount of time (in minutes). Note that training may go a bit beyond the specified time limit in order to complete the current cycle.

Advanced Settings



Drug

Mode: ○ Simple ● Expert

Quick Method Expert Options

Hidden layers: ● One ○ Two ○ Th

Layer 1: 20

Persistence: 200

Learning Rates

Alpha: 0.9

Initial Eta: 0.3

High Eta: 0.1

Eta

Lov

Hidden layers. Select the number of hidden layers for the neural network. More hidden layers can help neural networks learn more complex relationships, but they also increase training time.

Layer 1, 2, 3. For each layer, specify the number of hidden units to include. More hidden units per layer can also help in learning complex tasks, but as with additional hidden layers, they also increase training time.

Persistence. Specify the number of cycles for which the network will continue to train if no improvement is seen. Higher values can help networks escape local minima, but they also increase training time.

Alpha and Eta. These parameters control the training of the network.

Try Yourself

- Separate the dataset into 70/30 for the Training and Testing.
- Check the results.
- What are the differences?