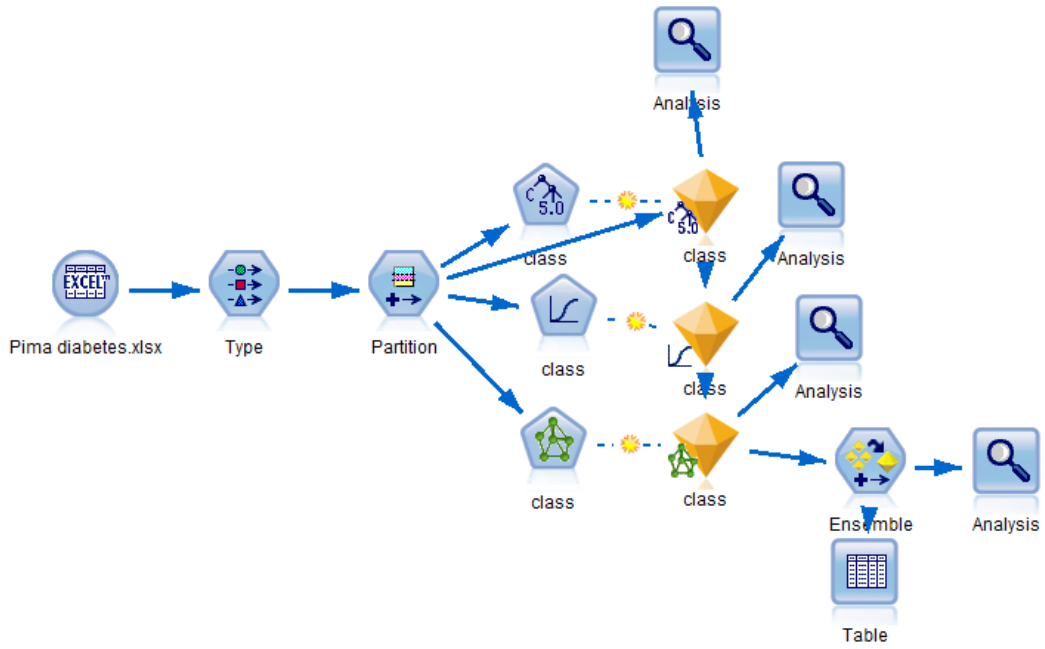
Ensembles in Machine Learning / Data Mining



The example presented above shows how to combine different classifiers and create an ensemble classifier. This example uses the **Pima Diabetes dataset,** which is on the Black Board. It is from the UCI archive and further information is available there.

Place a type node, and designate the last attribute class as the target and make sure to make its data type is a **Flag**. There are only two classes here (1 and 0). The above stream used a 60/40 split for training / test.

In the above stream, after the partition step, 3 separate classifier models were built namely, decision tree, logistic regression and a neural net. Highest classification accuracy among these models is 76.03% by the Neural net.

Creating an Ensemble of classifiers:

Next, connect the three model nodes to each other as shown above and then connected that stream to the ensemble node (Ensemble node is found in the **fieldops** tab). When you try to connect one model to the next, spss modeler will complain saying "**connection already exists**". Choose "**Replace**" and SPSS modeler should cooperate. In the ensemble node make sure you **deselect** the selection **Filter out fields generated by the ensemble models**.

Next add an analysis node to the ensemble node. (You need not **execute** Ensemble because **it is not creating a model.** It is simply taking output of the models in the ensemble (3 in the above example) and creating a combined or aggregate classification based on **voting.** I chose simple Voting, but there are other types of voting are possible. The analysis node shows that the ensemble classification accuracy on the test data is slightly better than all the 3 models. Namely 76.34%. See the results on the next page.

Results for output field class

Individual Models

Comparing $C-class with class

'Partition'	1_Training		2_Testing	
Correct	384	85.14%	224	70.66%
Wrong	67	14.86%	93	29.34%
Total	451		317	

Coincidence Matrix for $C-class (rows show actuals)

'Partition' = 1_Training	0.000000	1.000000
0.000000	287	23
1.000000	44	97
'Partition' = 2_Testing	0.000000	1.000000
0.000000	164	26
1.000000	67	60

Comparing $N-class with class

'Partition'	1_Training		2_Testing	
Correct	361	80.04%	241	76.03%
Wrong	90	19.96%	76	23.97%
Total	451		317	

Coincidence Matrix for $N-class (rows show actuals)

'Partition' = 1_Training	0.000000	1.000000
0.000000	273	37
1.000000	53	88
'Partition' = 2_Testing	0.000000	1.000000
0.000000	171	19
1.000000	57	70

Comparing $L-class with class

'Partition'	1_Training		2_Testing	
Correct	358	79.38%	239	75.39%
Wrong	93	20.62%	78	24.61%
Total	451		317	

Coincidence Matrix for $L-class (rows show actuals)

'Partition' = 1_Training	0.000000	1.000000
0.000000	281	29
1.000000	64	77
'Partition' = 2_Testing	0.000000	1.000000
0.000000	175	15
1.000000	63	64

Comparing $XF-class with class

'Partition'	1_Training		2_Testing	
Correct	368	81.6%	242	76.34%
Wrong	83	18.4%	75	23.66%
Total	451		317	

Coincidence Matrix for $XF-class (rows show actuals)

'Partition' = 1_Training	0.000000	1.000000
0.000000	282	28
1.000000	55	86
'Partition' = 2_Testing	0.000000	1.000000
0.000000	174	16
1.000000	59	68

Agreement between $C-class $N-class $L-class $XF-class

'Partition'	1_Training		2_Testing	
Agree	380	84.26%	263	82.97%
Disagree	71	15.74%	54	17.03%
Total	451		317	

Comparing Agreement with class

'Partition'	1_Training		2_Testing	
Correct	332	87.37%	204	77.57%
Wrong	48	12.63%	59	22.43%
Total	380		263	

Coincidence Matrix for Agreement (rows show actuals)

'Partition' = 1_Training	0.000000	1.000000
0.000000	262	13
1.000000	35	70
'Partition' = 2_Testing	0.000000	1.000000
0.000000	157	11
1.000000	48	47