Predictive Modeling Fundamentals I

Lab 3: Build the first predictive model

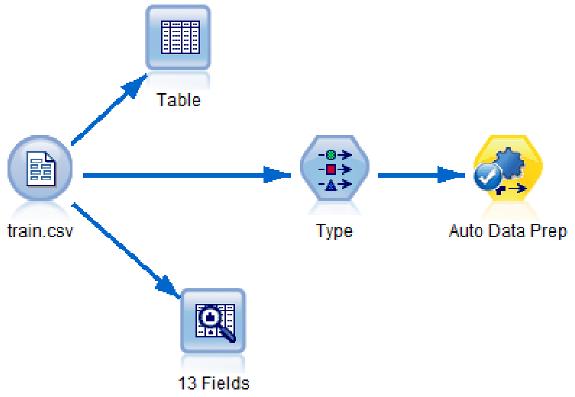
Contents

Build the first predictive model

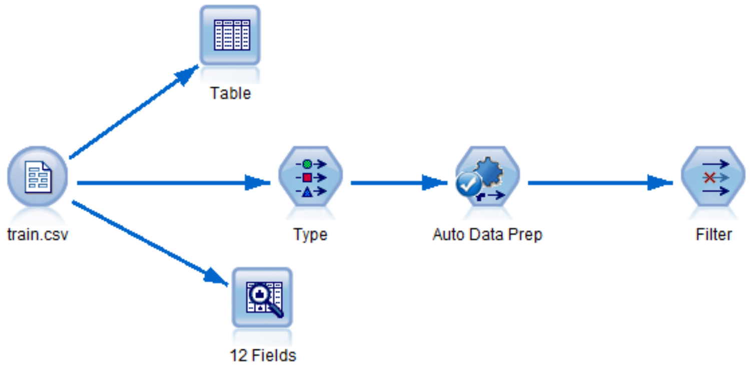
1.1 Predicting using Logistic regression

There are many algorithms available in IBM SPSS Modeler. To start simple we will create a Model using Logistic regression, also known as nominal regression. It is a statistical technique for classifying records based on values of input fields. In our use case, the input fields will be the characteristics of each of the passengers of the Titanic and the target will be the field Survived.

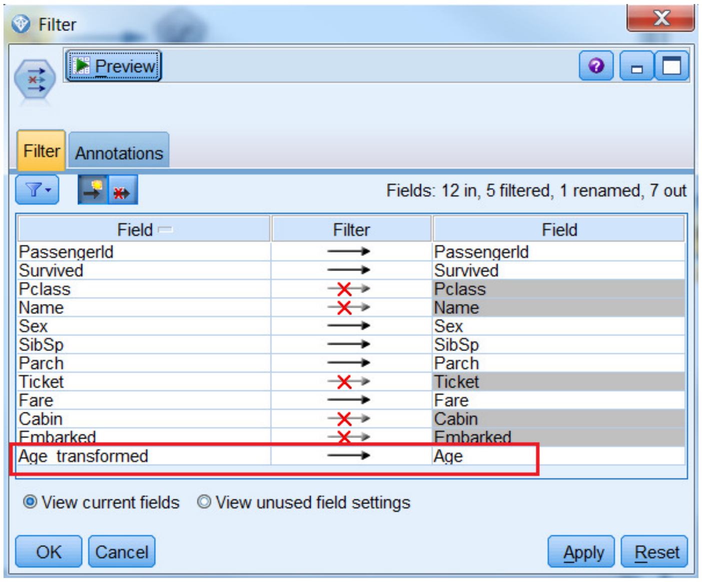
_1. Open the stream you prepared in Lab 2. In this stream you should have something similar to this:



_2. Add a **Filter** node from the **Field Ops** palette and connect it to the **Auto Data Prep** node.



_3. In the **Filter** node, click the filter icon in the center column for the variables that should not be included in the model and double click the Field name on the right side for the Age_transformed variable (that was imputed) and rename “Age” to return to original form as in below screenshot. Click Apply then OK.

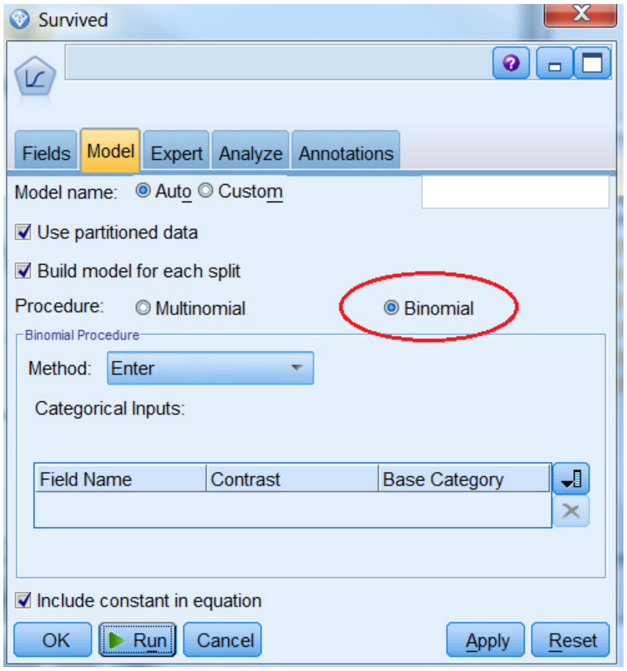


Fields: 12 in, 5 filtered, 1 renamed, 7 out

Field	Filter	Field
PassengerId	→	PassengerId
Survived	→	Survived
Pclass	✗→	Pclass
Name	✗→	Name
Sex	→	Sex
SibSp	→	SibSp
Parch	→	Parch
Ticket	✗→	Ticket
Fare	→	Fare
Cabin	✗→	Cabin
Embarked	✗→	Embarked
Age_transformed	→	Age

● View current fields ○ View unused field settings

_4. Add the **Logistic** node from the **Modeling** palette and connect it to the **Filter** node. Select **Binomial** procedure in the **Model** tab as you are predicting a binary outcome – survive/not survive.



Model name: ● Auto ○ Custom

☑ Use partitioned data

☑ Build model for each split

Procedure: ○ Multinomial ● Binomial

Binomial Procedure

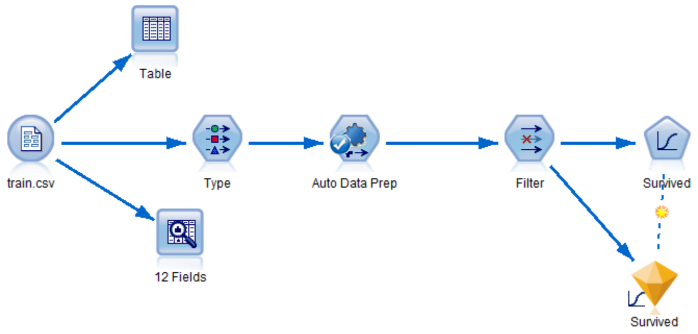
Method: Enter

Categorical Inputs:

Field Name	Contrast	Base Category

☑ Include constant in equation

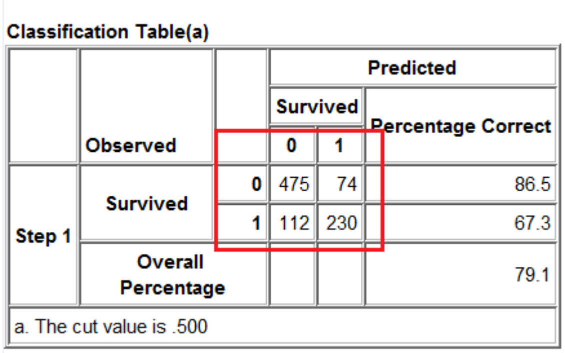
_5. Run the stream. A yellow nugget will appear in your canvas. This is the predictive model.



1.2 Validating the model of Logistic Regression

_1. Double click in the yellow Nugget called **Survived**. There is some useful information to understand better the model that you generated:

- **Summary Tab**: Information about what's been used to create the model (inputs, target, configuration, etc.)
- **Advanced Tab**: For those looking for advanced information, here there is all the summary of the model. Scroll down until you see the Classification Table. This is the misclassification or confusion matrix, which is used to calculate and understand accuracy of the model. We can see how many cases the model classifies correctly and incorrectly.

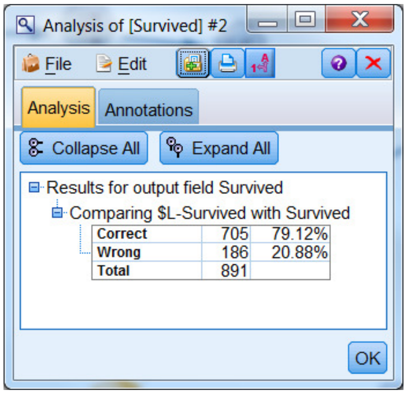


Classification Table(a)

			Predicted		
			Survived		Percentage Correct
	Observed		0	1	
Step 1	Survived	0	475	74	86.5
		1	112	230	67.3
	Overall Percentage				79.1

a. The cut value is .500

_2. Connect an **Analysis Node** from the **Output** palette. The Analysis Node enables you to evaluate the ability of a model to generate accurate predictions. It performs various comparisons between predicted values and actual values (your target field). **Run** the stream.



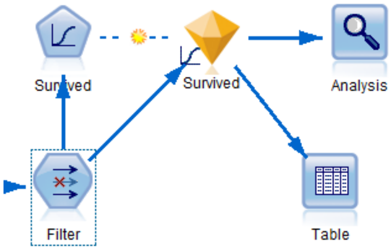
Results for output field Survived

Comparing $L-Survived with Survived

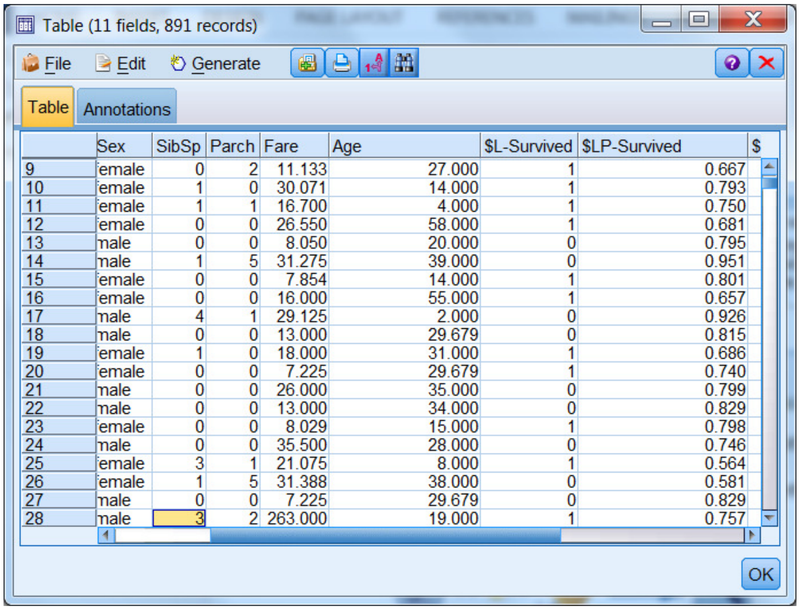
Correct	705	79.12%
Wrong	186	20.88%
Total	891	

In this case, our model is 81.48% accurate. Not bad, but can we do better?

_3. Attach a **Table** node from the **Output** palette to the Model nugget, double click it and hit run.



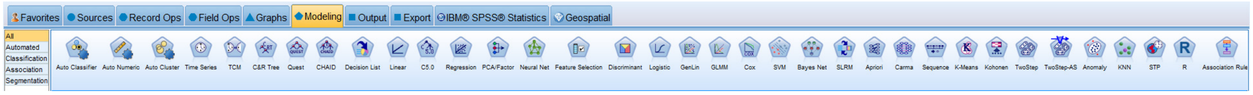
_4. You will see that fields have been created. $L-Survived shows the predicted outcome and $LP-Survived shows predicted probability for that outcome. For example, passenger in row 9 is predicted to survive (1) with a probability of ~67%.



	Sex	SibSp	Parch	Fare	Age	$L-Survived	$LP-Survived	$
9	emale	0	2	11.133	27.000	1	0.667	
10	emale	1	0	30.071	14.000	1	0.793	
11	emale	1	1	16.700	4.000	1	0.750	
12	emale	0	0	26.550	58.000	1	0.681	
13	male	0	0	8.050	20.000	0	0.795	
14	male	1	5	31.275	39.000	0	0.951	
15	emale	0	0	7.854	14.000	1	0.801	
16	emale	0	0	16.000	55.000	1	0.657	
17	male	4	1	29.125	2.000	0	0.926	
18	male	0	0	13.000	29.679	0	0.815	
19	emale	1	0	18.000	31.000	1	0.686	
20	emale	0	0	7.225	29.679	1	0.740	
21	male	0	0	26.000	35.000	0	0.799	
22	male	0	0	13.000	34.000	0	0.829	
23	emale	0	0	8.029	15.000	1	0.798	
24	male	0	0	35.500	28.000	0	0.746	
25	emale	3	1	21.075	8.000	1	0.564	
26	emale	1	5	31.388	38.000	0	0.581	
27	male	0	0	7.225	29.679	0	0.829	
28	male	3	2	263.000	19.000	1	0.757	

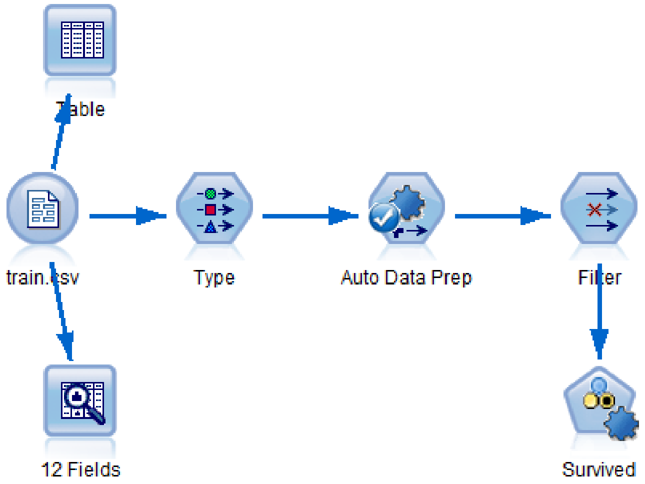
1.3 Create a Model using Auto-Modeling tools

In the previous section we created a model using Logistic Regression. In IBM SPSS Modeler there are many algorithms to use and you can see in the **Modeling** Palette.

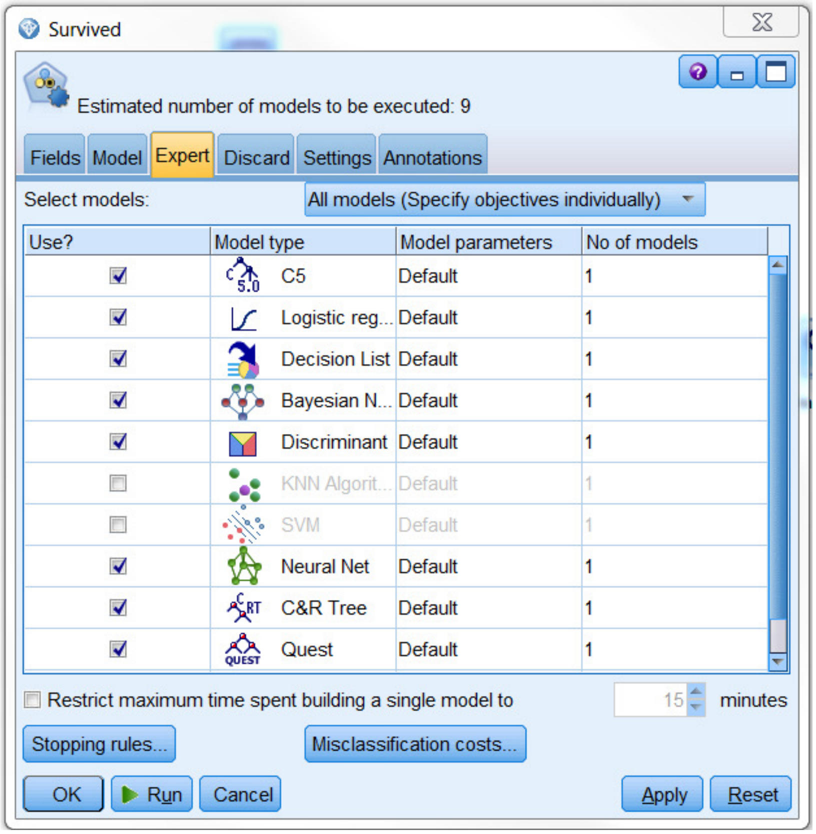


Unless you are an expert, you might feel a bit lost and you might don't know which is the best option. Previously we used Logistic Regression but what if there is a more accurate model? IBM SPSS Modeler provides Auto-Modeling tools that will recommend the best algorithms based on your input data. Let's learn how to use it.

_1. Add an **Auto Classifier Node** to the canvas and connect it to the **Filter** node.



_2. Double click in the **Auto Classifier Node** (now called Survived). Here you can change the advanced option of the Auto Classifier. Go to **the Expert Tab** and you will see the list of all the algorithms that will be executed. Take into account that the more you select the longer it will take to be executed. Since our dataset is small everything will move fast.



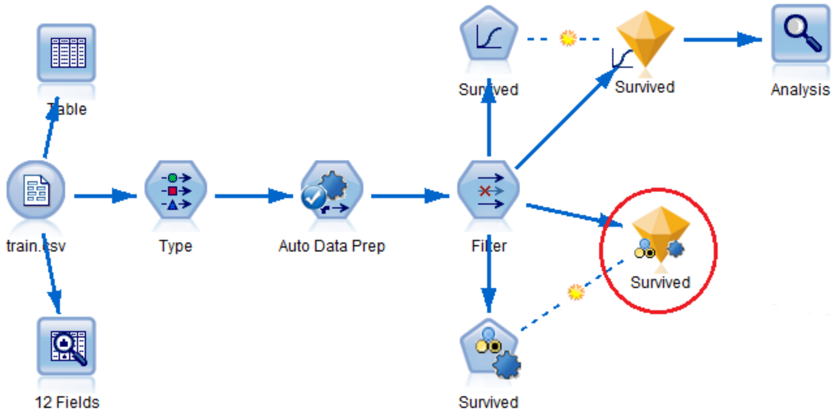
Estimated number of models to be executed: 9

Select models: All models (Specify objectives individually)

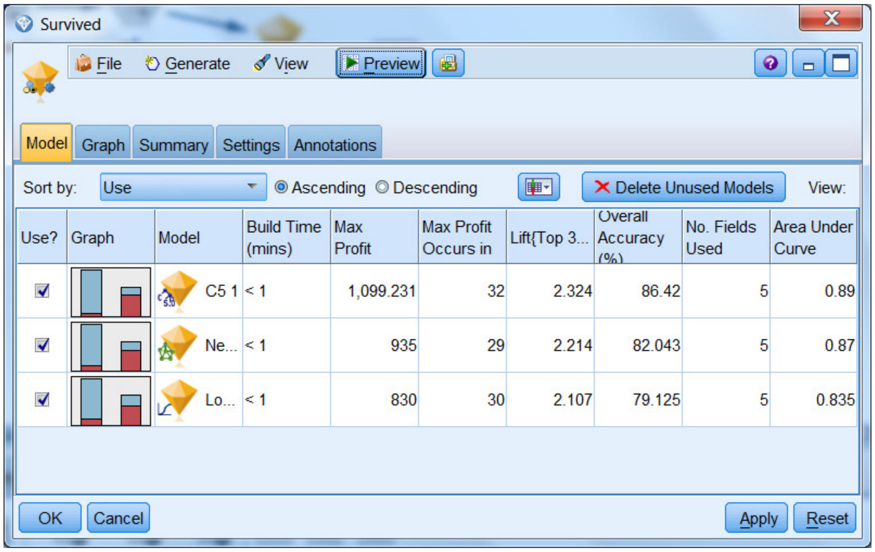
Use?	Model type	Model parameters	No of models
☑	C5	Default	1
☑	Logistic reg...	Default	1
☑	Decision List	Default	1
☑	Bayesian N...	Default	1
☑	Discriminant	Default	1
☐	KNN Algorit...	Default	1
☐	SVM	Default	1
☑	Neural Net	Default	1
☑	C&R Tree	Default	1
☑	Quest	Default	1

☐ Restrict maximum time spent building a single model to 15 minutes

_3. Run the stream. When an automated modeling node is executed, the node estimates candidate models for every possible combination of options, ranks each candidate based on the measure you specify, and saves the best models in a composite automated model nugget.



_4. Double click in the new model. The top 3 models are proposed with a list of measures that you can sort: Building Time, Max Profit, Overall Accuracy, etc. In this example the algorithm **C5.1** is the most accurate.



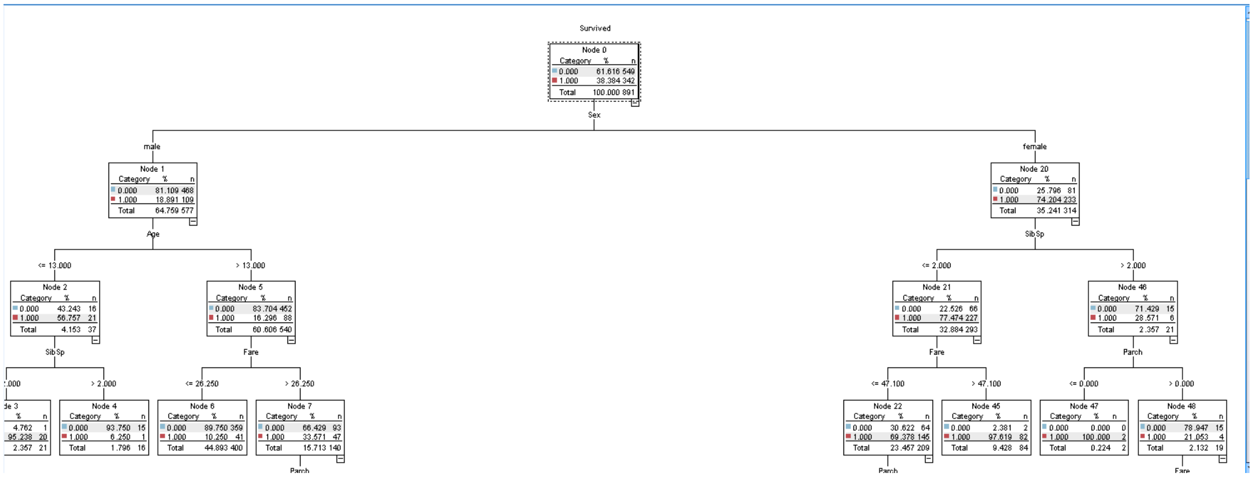
Sort by: Use ● Ascending ○ Descending

Use?	Graph	Model	Build Time (mins)	Max Profit	Max Profit Occurs in	Lift{Top 3...	Overall Accuracy (%)	No. Fields Used	Area Under Curve
☑		C5 1	< 1	1,099.231	32	2.324	86.42	5	0.89
☑		Ne...	< 1	935	29	2.214	82.043	5	0.87
☑		Lo...	< 1	830	30	2.107	79.125	5	0.835

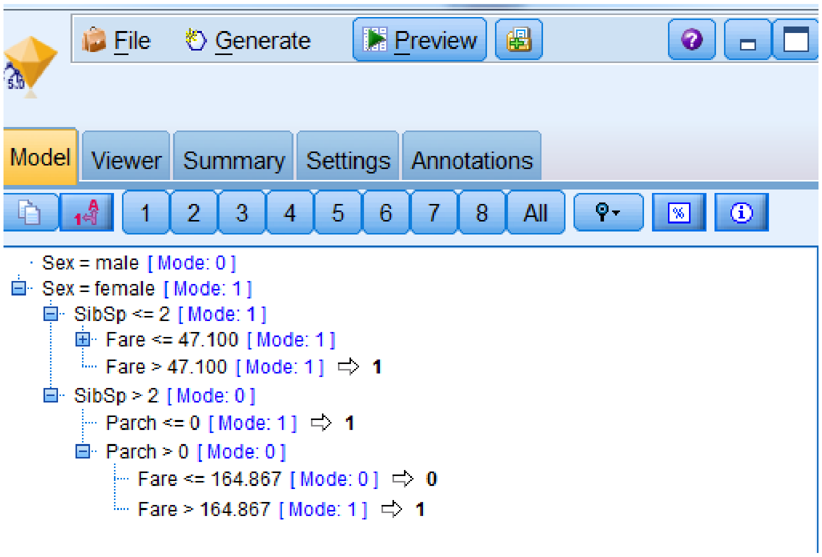
1.4 Evaluate the Auto-Model

_1. Double click in the C5.1 Icon:

_2. Here there are all the details of the **C5.1 model** that we just generated. In the case of the C5.1 algorithms, the results are quite easy to understand since it is a decision tree with a set of conditions. Click on the **Viewer** tab to see the decision tree.

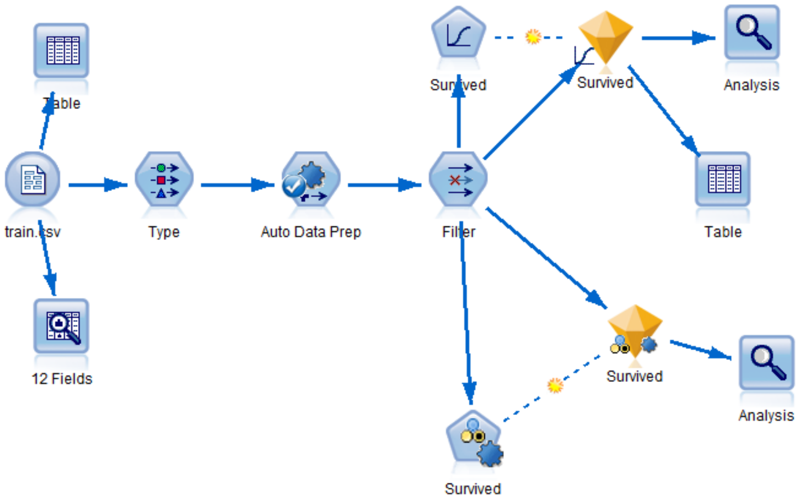


_3. The **Model** tab shows you the same interpretation as the decision tree but in the form of rules. Expand the rules by clicking the ⊞ signs to see them.

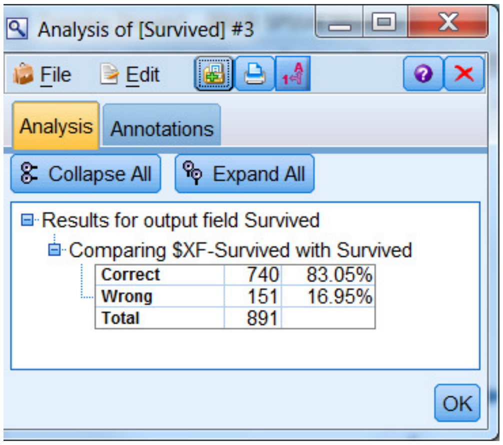


```
· Sex = male [ Mode: 0 ]
⊟ Sex = female [ Mode: 1 ]
   ⊟ SibSp <= 2 [ Mode: 1 ]
      ⊞ Fare <= 47.100 [ Mode: 1 ]
        Fare > 47.100 [ Mode: 1 ] ⇨ 1
   ⊟ SibSp > 2 [ Mode: 0 ]
        Parch <= 0 [ Mode: 1 ] ⇨ 1
      ⊟ Parch > 0 [ Mode: 0 ]
           Fare <= 164.867 [ Mode: 0 ] ⇨ 0
           Fare > 164.867 [ Mode: 1 ] ⇨ 1
```

_4. As we previously did, add an **Analysis Node** and connect it to the new yellow nugget. This will give us the accuracy.



_5. The results of the **Analysis Node** show that our model is 83.05% accurate. Our accuracy just improved!



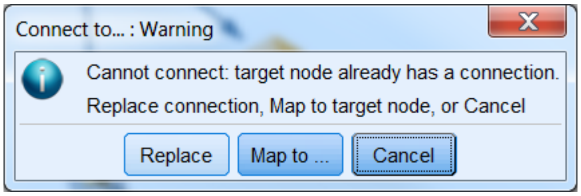
Results for output field Survived

Comparing $XF-Survived with Survived

Correct	740	83.05%
Wrong	151	16.95%
Total	891	

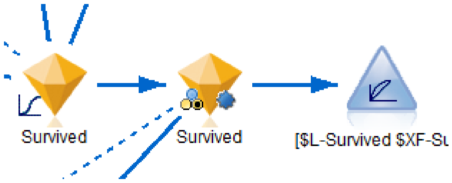
_5. Let us compare the performance of the Decision Tree and Logistic Regression in a visual way. To do so, we can use the ROC (receiver operator characteristic curve) graph. Whichever model has greatest area under this curve, generates more accurate results. Start by adding in the **Evaluation** graph node from **Graphs** palette.

_6. To compare models, we will have to connect the model nuggets to each other. Connect the **Logistic** nugget to the **Auto-Classifier** nugget. You will get this warning. Click replace.



Cannot connect: target node already has a connection.
Replace connection, Map to target node, or Cancel

_7. Connect the **Auto-Classifier** nugget to the **Evaluation** Graph node. The caption will change to [$L-Survived….



_8. Double click the Graph node and select Receiver Operator Characteristic (ROC) as Chart type. Click Run.

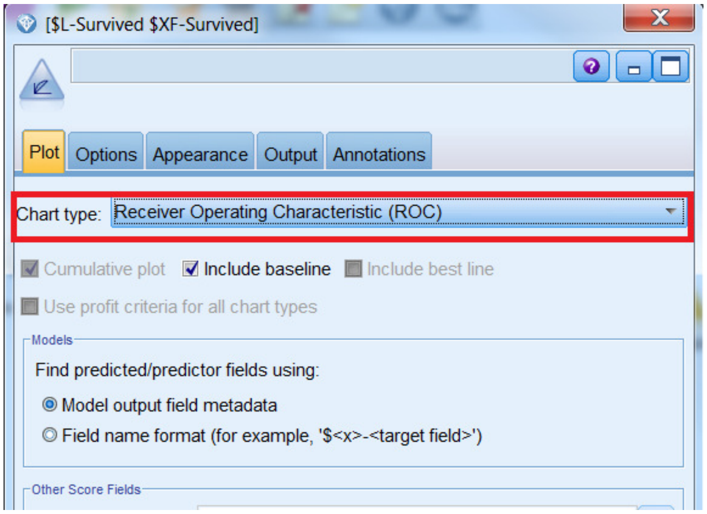


Chart type: Receiver Operating Characteristic (ROC)

☑ Cumulative plot ☑ Include baseline □ Include best line

□ Use profit criteria for all chart types

Models

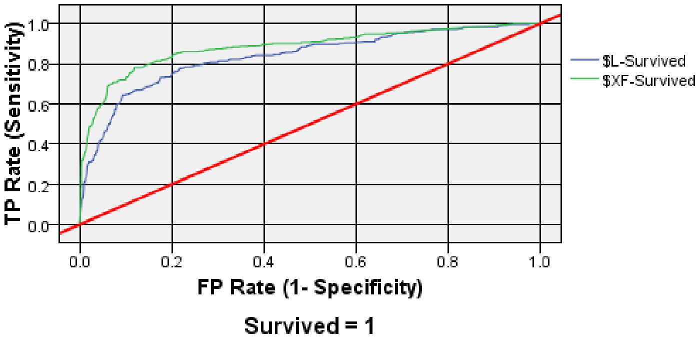
Find predicted/predictor fields using:

● Model output field metadata

○ Field name format (for example, '$<x>-<target field>')

Other Score Fields

_9. We see that the Decision Tree (marked by blue line and $XF-Survived) has a greater area under the curve than Logistic Regression (green line and $L-Survived).



Summary

Congratulations! You built your first predictive model using IBM SPSS Modeler!

In this lab you learnt how to create new models using IBM SPSS Modeler. There are many different options that can satisfy the basic need of people without good knowledge on the algorithms but also very advanced options for the advanced users.

We create a model using logistic regression, a very popular algorithm, and we’ve got an accuracy of 79%. Then we asked IBM SPSS Modeler to propose us which algorithms we should use based on our dataset. The Auto-Modeling tools suggested to use C5.1 instead, and our accuracy improved to 83%. Improving the accuracy of a model can be a very time consuming task!