Solutions to Appendix A: A Practice Case Study for *Tree-Based Machine Learning Methods in SAS® Viya®*

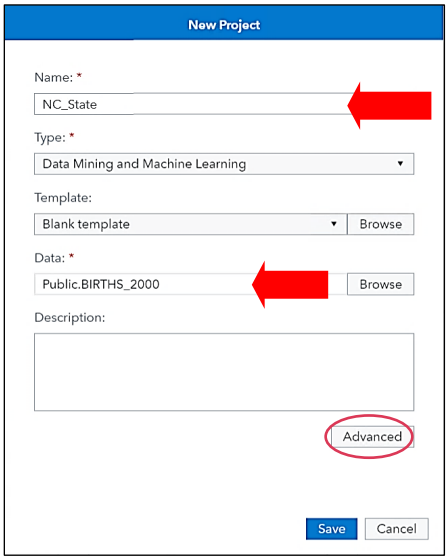
Your results might differ from those in these solutions because of the random partitioning performed and SAS Viya distributed computing environment.

The questions related to manual tuning or autotuning of the models are experimentative. The solutions provided here are indicative and might not be reproducible.

1. Building a Model Studio Project and Accessing the Data

a) *Create a new Model Studio project and name it as NC_State. Choose the **births_2000** data as the analysis data. The data are available in the data folder.*

Click the **New Project** button. Enter **NC_State** in the **Name** field. Click the **Browse** button in the **Data** field. The Choose Data window opens.



Import the analysis data set into CAS:

i. In the Choose Data window, click **Import**.

ii. Under **Import**, expand **Local Files** and then select **Local File**.

iii. Navigate to the data folder.

iv. Select the **births_2000.sas7bdat** table.

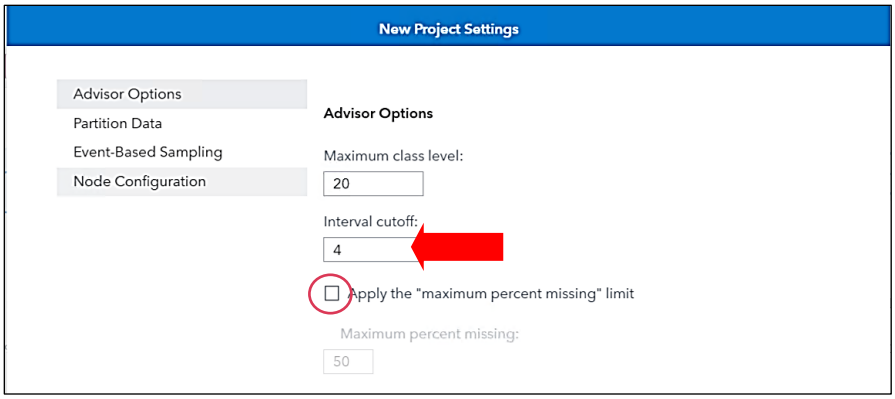
v. Click **Open**.

vi. Select **Import Item**.

vii. Click **OK** after the table is imported.

b) *Configure the data such that you do not reject any inputs, even if they have a higher percentage of missing values. Ensure that any numeric variable with four or more levels remains defined with a level of interval.*

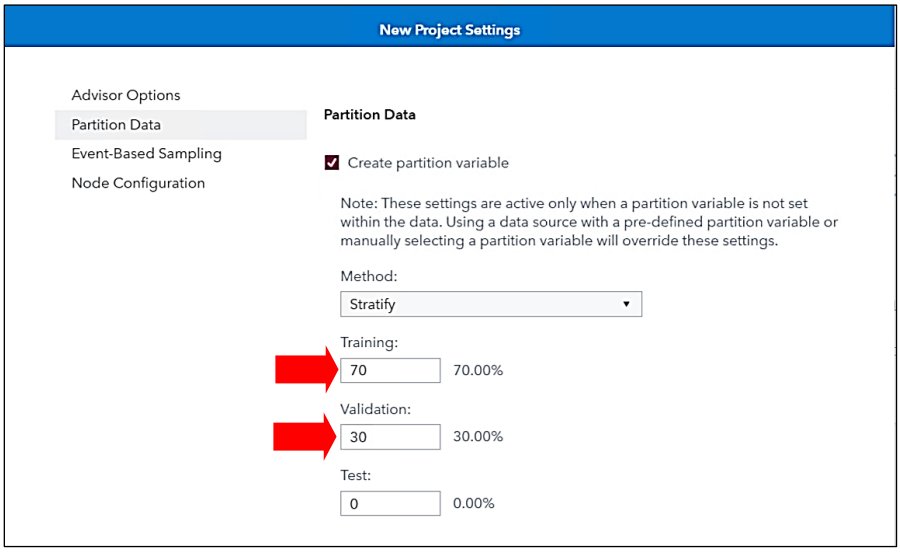
Click **Advanced** in the New Project Settings window. Under the **Advisor Options** group, deselect the **Apply the "maximum percent missing" limit** and set **Interval cutoff** to **4**.



Changing the threshold to 4 means that any numeric variable with four or more levels remains defined with a level of *interval*.

c) *Split the data into two parts. Keep 70% for Training and the remaining for Validation.*

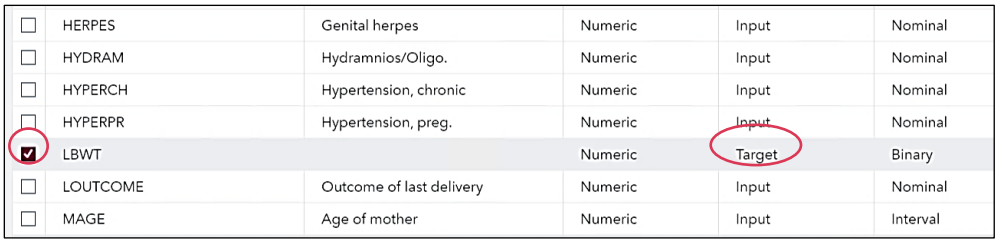
With **Advanced** selected in the New Project window, under the **Partition Data** group, change the **Training** percentage to **70** and the **Test** percentage to **0**.



Click **Save** to return to the New Project window. Click **Save** again to close the New Project window.

d) *Assign the target variable. Do you have any variables that are rejected from the analysis?*

After the project is created, Model Studio takes you to the Data tab of your new project. On the Data tab, assign LBWT as the target variable.



☐	HERPES	Genital herpes	Numeric	Input	Nominal
☐	HYDRAM	Hydramnios/Oligo.	Numeric	Input	Nominal
☐	HYPERCH	Hypertension, chronic	Numeric	Input	Nominal
☐	HYPERPR	Hypertension, preg.	Numeric	Input	Nominal
☑	LBWT		Numeric	Target	Binary
☐	LOUTCOME	Outcome of last delivery	Numeric	Input	Nominal
☐	MAGE	Age of mother	Numeric	Input	Interval

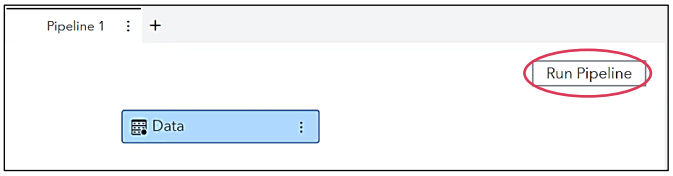
In the variables window, select **LBTW** (Step 1). Then in the right pane, select **Target** under the **Role** property (Step 2). (You might need to scroll down in the variable list to see LBWT.)

None of the variables are rejected in the project.

2. Assaying the Data

a) *Have the data been partitioned in your project? If not, execute splitting the data.*

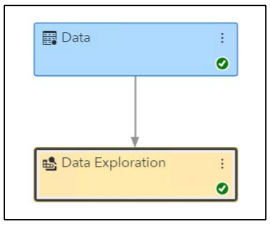
Go to the **Pipelines** tab and click the **Run Pipeline** button to execute the metadata advisor options and data partition.



b) *Explore the analysis data and try to gain some knowledge about the variables by using both graphical and numerical methods. Select a subset of variables to provide a representative snapshot of the data.*

Right-click the **Data** node and select **Add child node ⇨ Miscellaneous ⇨ Data Exploration**.

In the properties panel, ensure that **Input data partition** is set to **All data** (default) so that your explorations will be based on the entire data.

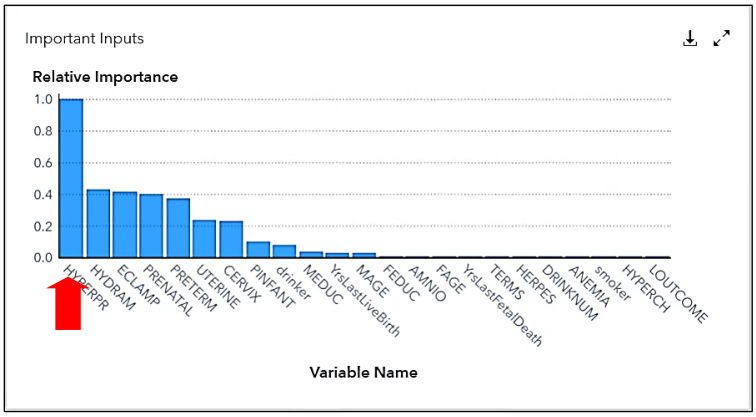


Run the **Data Exploration** node.

The Data Exploration node selects a subset of variables to provide a representative snapshot of the data.

c) *Which one is the most important variable? Is the most important variable related to the health of the mother?*

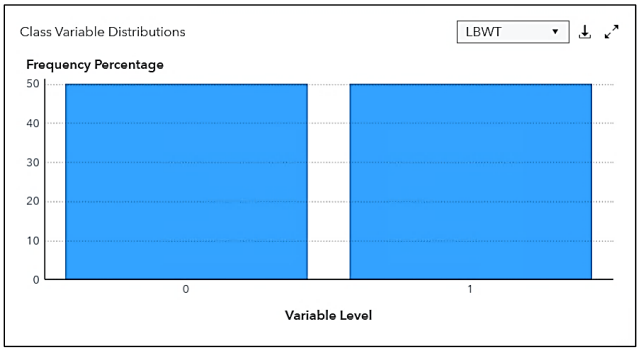
Select **Results** (of the Data Exploration node) ⇨ **Important Inputs** chart.



Hypertension for this pregnancy (**HYPERPR**) is the most important variable related to the health of the mother.

d) *What percentage of babies are low birth weight in this data?*

In the **Results** (of the Data Exploration node), click the **Class Variable Distributions** chart and select (in drop-down menu) **LBTW**.



Fifty percent of babies are low birth weight.

e) *What is the average age of mothers and fathers?*

In the **Results** (of Data Exploration node), click the **Interval Variable Moments** table and see the Mean column of **FAGE** and **MAGE**.

Interval Variable Moments

Variable Na...	Minimum	Maximum	Mean	Standard D...
DRINKNUM	0	98	0.1148	2.9983
FAGE	15	75	29.4029	6.7868
FEDUC	0	17	12.7003	2.7201
MAGE	11	48	26.2462	6.2346
MEDUC	0	17	12.5648	2.6793
PRENATAL	0	9	2.3647	1.4227
TERMS	0	13	0.4056	0.8417
YrsLastFetalDeath	0	26	4.3107	4.1344

The average age of mothers and fathers is 26.25 and 29.40 years, respectively.

f) *Do any variables have missing values in these data? Which variable has the highest missing percentage? Although decision trees can handle missing values very well, do you think that the variable with the highest percentage of missing values should be rejected from the analysis? Why?*

In the **Results** (of the Data Exploration node) ⇨ **Output** ⇨ (scroll to) **Missing Values** table.

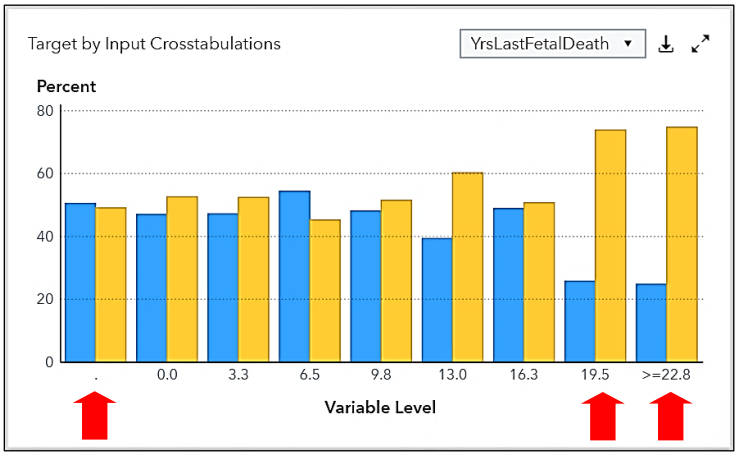
Missing Values

Variable Name	Number of Missing Values	Percentage Missing
AMNIO	0	0.0000
ANEMIA	0	0.0000
CERVIX	0	0.0000
DRINKNUM	47	0.2749
ECLAMP	0	0.0000
FAGE	3538	20.6937
FEDUC	3568	20.8692
HERPES	0	0.0000
HYDRAM	0	0.0000
HYPERCH	0	0.0000
HYPERPR	0	0.0000
LBWT	0	0.0000
LOUTCOME	0	0.0000
MAGE	0	0.0000
MEDUC	60	0.3509
PINFANT	0	0.0000
PRENATAL	164	0.9592
PRETERM	0	0.0000
TERMS	27	0.1579
UTERINE	0	0.0000
YrsLastFetalDeath	13157	76.9550
YrsLastLiveBirth	7858	45.9613
drinker	15	0.0877
smoker	30	0.1755

The number of years since last fetal death (**YrsLastFetalDeath**) has the highest (77%) missing values.

Other variables that have missing values include Average number of alcoholic drinks per week (**DRINKNUM**), age of father (**FAGE**), education of father (**FEDUC**), education of mother (**MEDUC**), months of pregnancy prenatal care begun (**PRENATAL**), number of other terminations (**TERM**), number of years since last live birth (**YrsLastLiveBirth**), Mother drinks alcohol (**DRINKER**), and mother smokes (**SMOKER**).

In the **Results** (of the Data Exploration node) ⇨ **Target by Input Crosstabulations** ⇨ (select in drop-down menu) **YrsLastFetalDeath**.



Missingness in this variable does not seem to be related with the target. (See the first set of almost equal bars.) However, careful examination reveals that for higher values of the number of years since last fetal death, there are enormous chances of low birth weight. (See the last two sets of highly unequal bars.) What if the missing values were higher values of the number of years since last fetal death? Because the percentage of missingness is high, a safer bet would be to reject this variable.

Schafer and Graham (2002) have a readable discussion and provide an example where they impute in a data set with more than 70% missing.

Close the **Results** (of the Data Exploration node) window.

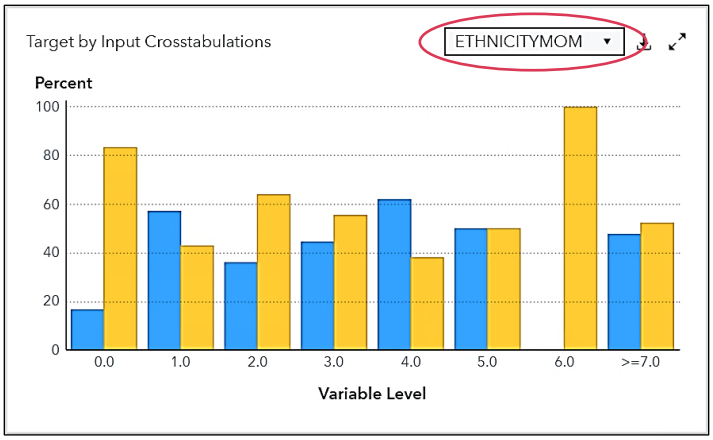
In the **Data** pane, change the **Role** of **YrsLastFetalDeath** to **Rejected**.

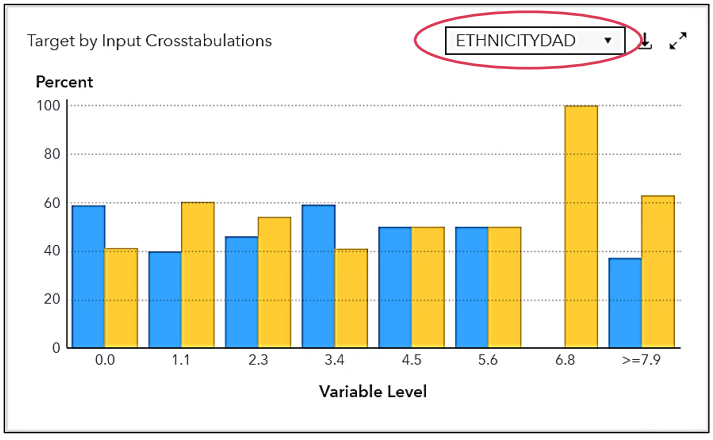
☐	TOTALP	Total pregnancies (including this one)	Numeric	Input	Interval	Default
☐	ULTRA	Ultrasound	Numeric	Input	Nominal	Default
☐	UTERINE	Uterine bleeding	Numeric	Input	Nominal	Default
☑	YrsLastFetalDeath		Numeric	Rejected	Interval	Default
☐	YrsLastLiveBirth		Numeric	Input	Interval	Default

g) *Is there any association between parents' ethnicity and low birth weight? Should these variables be rejected from the analysis?*

Return to the **Pipelines** tab and click the **Run Pipeline** button.

Select **Results** (of the Data Exploration node) ⇨ **Target by Input Crosstabulations** ⇨ (select in drop-down menu) **ETHNICITYMOM** and **ETHNICITYDAD**.





Paternal and maternal ethnicity does not seem to be an important predictor of low birth weight. The association is not significant among all the groups in the U.S. Other maternal and paternal characteristics such as marital status, age, and education should be examined in relation to birth outcomes.

Close the **Results** (of the Data Exploration node) window.

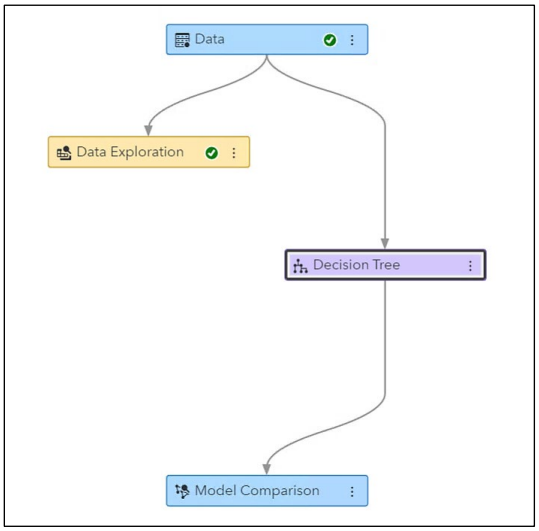
In the **Data** pane, change the role of **ETHNICITYMOM** and **ETHNICITYDAD** to **Rejected**.

☐	DRINKNUM	Average # of alcoholic drinks per week	Numeric	Input	Interval	Default
☐	ECLAMP	Eclampsia	Numeric	Input	Nominal	Default
☑	ETHNICITYDAD		Numeric	Rejected	Interval	Default
☑	ETHNICITYMOM		Numeric	Rejected	Interval	Default
☐	FAGE	Age of father	Numeric	Input	Interval	Default
☐	FEDUC	Education of father (years)	Numeric	Input	Interval	Default

3. Creating a Decision Tree Model

a) *Create a decision tree model using all the default settings in Model Studio.*

Return to the **Pipelines** tab. Right-click the **Data** node and select **Add child node** ⇨ **Supervised Learning** ⇨ **Decision Tree**.



Run the **Decision Tree** node.

b) *How many leaves does the model have? Up to how many leaves were originally grown on the tree?*

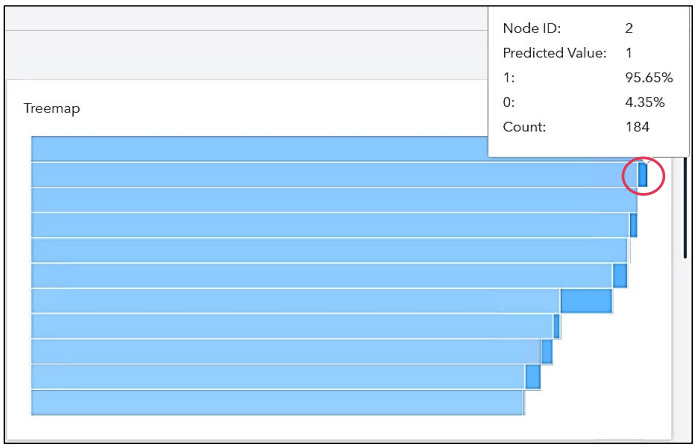
Select the **Results** (of the Decision Tree node) ⇨ **Output** ⇨ (focus on) **Model Information** table.

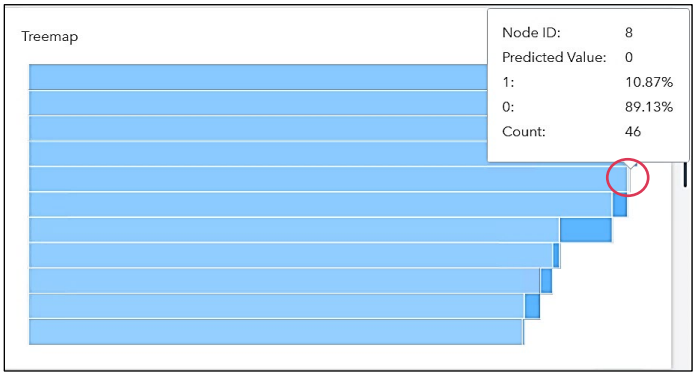
Model Information	
Split Criterion	IGR
Pruning Method	Cost Complexity
Max Branches per Node	2
Max Tree Depth	10
Tree Depth Before Pruning	10
Tree Depth After Pruning	10
Number of Leaves Before Pruning	50
Number of Leaves After Pruning	11

The decision tree with default settings has 50 leaves. However, the final tree has 11 leaves after pruning.

c) *What are the rules defining segments with the highest and lowest probabilities of low birth weight?*

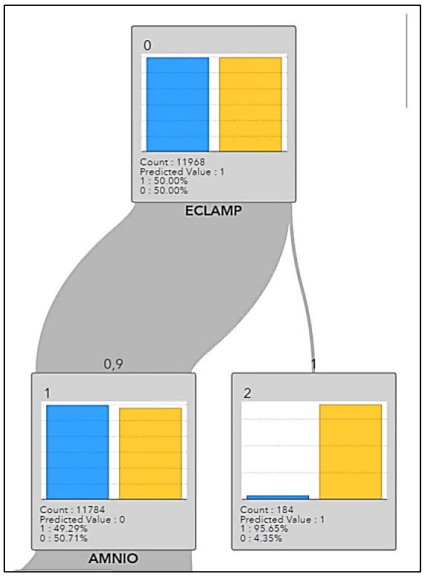
In the **Results** (of the Decision Tree node) ⇨ **Treemap** ⇨ hover your pointer on the darkest node and then on the lightest node.





Node IDs 2 and 8 are corresponding to the highest (95.65%) and lowest (10.87%) event rates in this run.

In the **Results** (of the Decision Tree node) ⇨ **Tree Diagram** ⇨ locate the above two nodes and explore the decision rules associated with them (suitably adjust the view).



The two nodes have interesting decision rules. If ECLAMP=1, that is, Eclampsia (seizures or convulsions in a pregnant woman that are not related to a pre-existing brain condition) is yes, then having a low birth weight is most likely. On the other hand, if ECLAMP=0,9 and AMNIO=1,0 and CERVIX=0,9 and PINFANT=1 then having a low birth weight is least likely.

d) *Which are the most important inputs in predicting the low birth weight?*

In the **Results** (of the Decision Tree node) ⇨ **Variable Importance** table.

Variable Label	Role	Variable Name	Validation Importance	Im...	Relative Importance
Hypertension, preg.	INPUT	HYPERPR	82.5882	0	1
Hydramnios/Oligo.	INPUT	HYDRAM	37.6457	0	0.4558
Prev. preterm/small	INPUT	PRETERM	32.1950	0	0.3898
Month of preg. prenatal care began	INPUT	PRENATAL	22.8318	0	0.2765
Eclampsia	INPUT	ECLAMP	20.2089	0	0.2447
Uterine bleeding	INPUT	UTERINE	17.4173	0	0.2109
Incompetent cervix	INPUT	CERVIX	16.3057	0	0.1974
Prev. infant 4000+gm	INPUT	PINFANT	10.3578	0	0.1254
Amniocentesis	INPUT	AMNIO	0.5061	0	0.0061
	INPUT	YrsLastFetalDeath	0.2671	0	0.0032

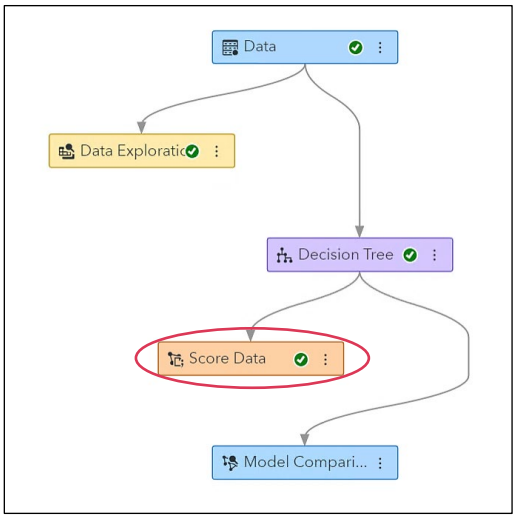
The variables mentioned above in the table are important inputs in predicting low birth weight.

Close the **Results** (of the Decision Tree node) window.

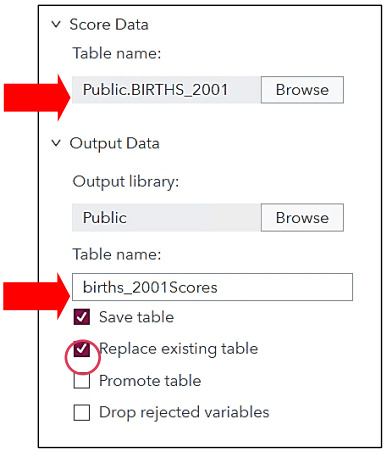
4. Out-of-Time Testing of the Decision Tree Model

a) *You have used births data for the year 2000 to build the model and did the in-time validation of the model, no testing being done. Use the year 2001 data to do the out-of-time testing of the decision tree model.*

Right-click the **Decision Tree** node and select **Add child node** ⇨ **Miscellaneous** ⇨ **Score Data**.



Choose the **births_2001** data as the score data (import the table if necessary). Name the output data as **births_2001Scores** in the **Public** library. Check the **Replace existing table** box.



Run the **Score Data** node.

b) *What are the ASE and MISC measures of the model's out-of-time performance? Are you getting comparable results for this out-of-time testing?*

Select the **Results** (of the Score Data node) ⇨ **Assessment** tab ⇨ **Fit Statistics** table (you might need to expand the table).

Target ...	Data Role	Sum of ...	Averag...	Divisor ...	Root Av...	Misclas...
LBWT	TRAIN	16,687	0.2317	16,687	0.4814	0.3955

The out-of-time-test ASE and MISC are 0.2317 and 0.3955, respectively.

Close the **Results** (of the Score Data node) window.

Select the **Results** (of the Decision Tree node) ⇨ **Assessment** tab ⇨ **Fit Statistics** table (you might need to expand the table).

Target ...	Data Role	Partitio...	Formatt...	Sum of ...	Averag...	Divisor ...	Root Av...	Misclas...
LBWT	TRAIN	1	1	11,968	0.2281	11,968	0.4776	0.3849
LBWT	VALIDATE	0	0	5,129	0.2266	5,129	0.4760	0.3818

The in-time-validation ASE and MISC are 0.2266 and 0.3818, respectively.

There is a slight weakening of the model's performance for 2001 births data. However, the difference is not concerning. The test harness is established using out-of-time testing.

Close the **Results** (of the Decision Tree node) window.

c) *What is the advantage and disadvantage of this approach?*

The advantage of this approach is that you explicitly evaluate the model's ability to predict out-of-time. The disadvantage is the inability to not consider the out-of-time data while building the model. This is because you select model that performs well on the *in-time* validation data. Moreover, unless you re-train the model using the full 2000–2001 data, you are not using the most recent data to make predictions.

d) *What if your out-of-time test performance is quite different from the in-time validation performance? What can you do?*

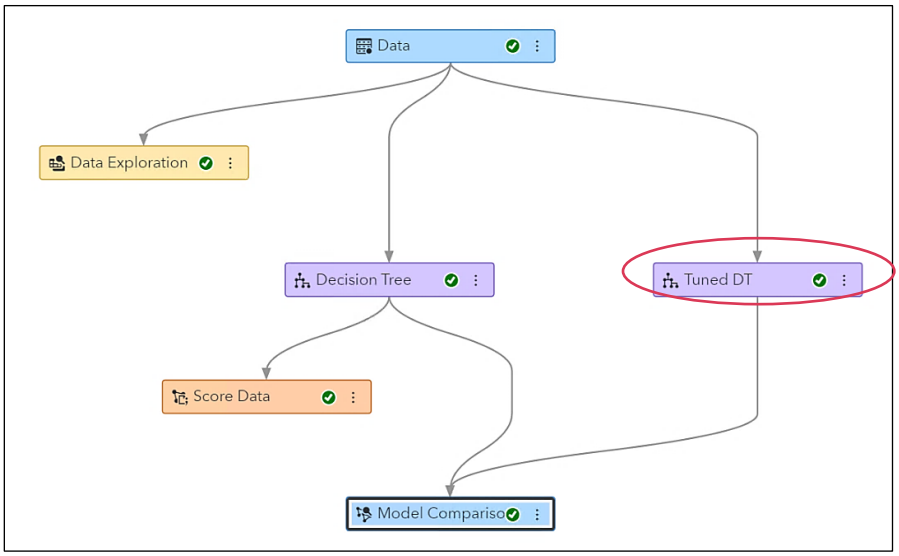
When you make predictions into the future, it is probably a clever idea to use data as recent as possible so that you capture the most recent phenomena. However, if you hold out the most recent data to evaluate

out-of-time performance, you should retrain the model using all the labeled data before making predictions. Instead, perform k-fold cross validation on the entire labeled data.

5. Improving the Decision Tree Model by Modifying Certain Tree Growth Options

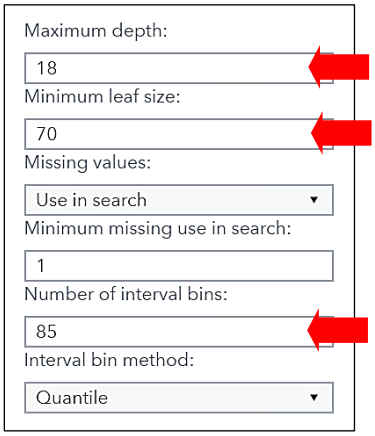
a) *Add one more decision tree in the pipeline and rename it as Tuned DT.*

Right-click the **Data** node and select **Add child node ⇨ Supervised Learning ⇨ Decision Tree**. Rename this Decision Tree node as **Tuned DT.**



b) *Change certain tree growth options to improve the decision tree model. Do you think that increasing the tree depth, leaf size, and number of interval bins will benefit improving the decision tree model?*

Under Splitting Options, change **Maximum depth** from 10 to **18**, **Minimum leaf size** from 5 to **70**, and **Number of interval bins** from 50 to **85**.



Click the **Run Pipeline** button.

c) *How does this manually tuned decision tree perform in comparison with the default tree?*

Select the **Results** (of the Model Comparison node) ⇨ **Model Comparison** table.

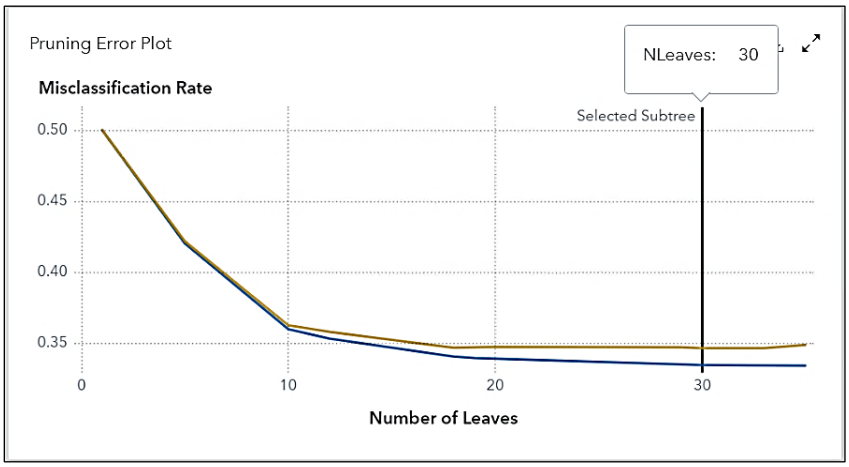
Champi...	Name	Algorith...	KS (You...	Misclas...	Misclas...	Root Av...	Averag...
★	Tuned DT	Decision Tree	0.3055	0.3472	0.3472	0.4677	0.2187
	Decision Tree	Decision Tree	0.2414	0.3792	0.3792	0.4752	0.2258

The manually tuned decision tree has a smaller ASE and MISC values and a higher KS (Youden) than the default decision tree, and is consequently, an improved model.

6. Experimenting with the Pruning Strategy

a) *How many leaves does the decision tree model have after pruning? Why have those many leaves been selected?*

Select the **Results** (of the Tuned DT node) ⇨ **Pruning Error Plot** ⇨ hover your pointer on the Selected Subtree vertical line.



A decision tree with thirty leaves is selected after pruning.

Select the **Results** (of the Tuned DT node) ⇨ **Output** ⇨ (focus on) **Cost Complexity Pruning** table.

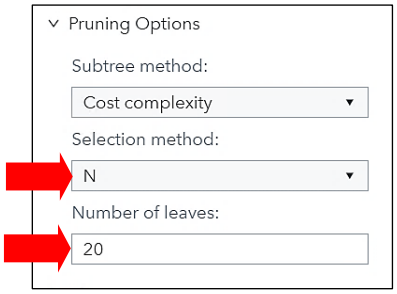
Cost Complexity Pruning			
		Misclassification Rate	
Alpha	Number of Leaves	Training	Validation
0	35	0.3340	0.3486
0.00004	33	0.3341	0.3463
0.00013	31	0.3343	0.3463
0.00017	30	0.3345	0.3463
0.00033	29	0.3348	0.3469
0.00045	20	0.3388	0.3472
0.00050	19	0.3393	0.3469
0.00100	18	0.3403	0.3467
0.00209	17	0.3424	0.3484
0.00212	12	0.3530	0.3578
0.00334	10	0.3597	0.3624
0.01207	5	0.4200	0.4217
0.01999	1	0.5000	0.5001

A decision tree with 30 leaves is selected because it has the smallest validation misclassification of 0.3463. The corresponding cost-complexity alpha value is 0.00017. For 20 leaves (ten leaves lesser than the best), the corresponding cost-complexity alpha value is 0.00045.

b) *Would it be acceptable to decrease the tree complexity by pruning the Tuned DT for reduced number of leaves, around 10 leaves fewer than what you have been getting? What are the two ways in which you can get this done in Model Studio? Apply any of these two methods and run the pipeline.*

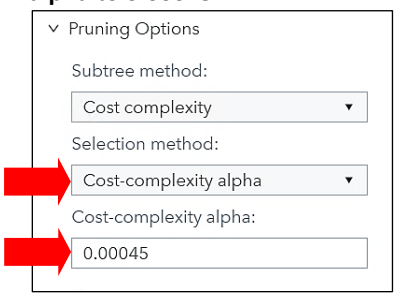
Method 1:

Under Pruning Options, change the **Selection method** to **N** and **Number of leaves** to **20**.



Method 2:

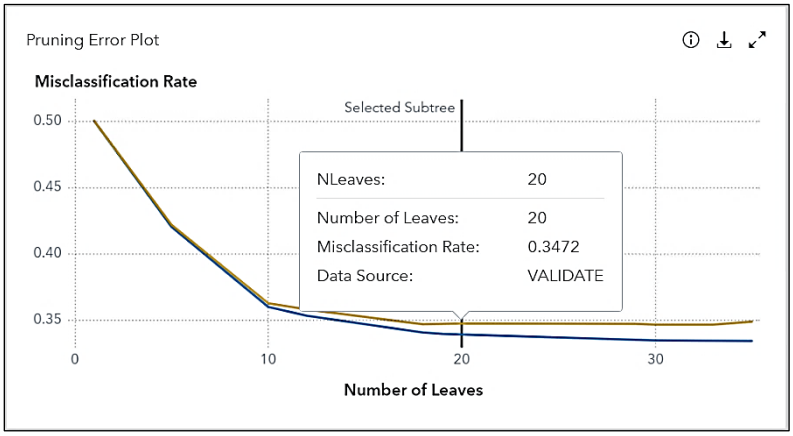
Under Pruning Options, change the **Selection method** to **Cost-complexity alpha** and **Cost-complexity alpha** to **0.00045**.



Apply any of the above two methods and click the **Run Pipeline** button.

c) *Compared to the previous tree, how does the resulting tree with reduced complexity (leaves) performs on accuracy? Is it worth reducing the tree complexity?*

Select the **Results** (of the Tuned DT node) ⇨ **Pruning Error Plot** ⇨ hover your pointer on the Selected Subtree vertical line at validation curve.



The validation misclassification has increased slightly from 0.3463 to 0.3472. Adding ten more leaves might not be worth it for a tiny gain.

d) *How has the manually tuned tree performed as compared to the tree with default settings?*

Select the **Results** (of the Model Comparison node) ⇨ **Model Comparison** table.

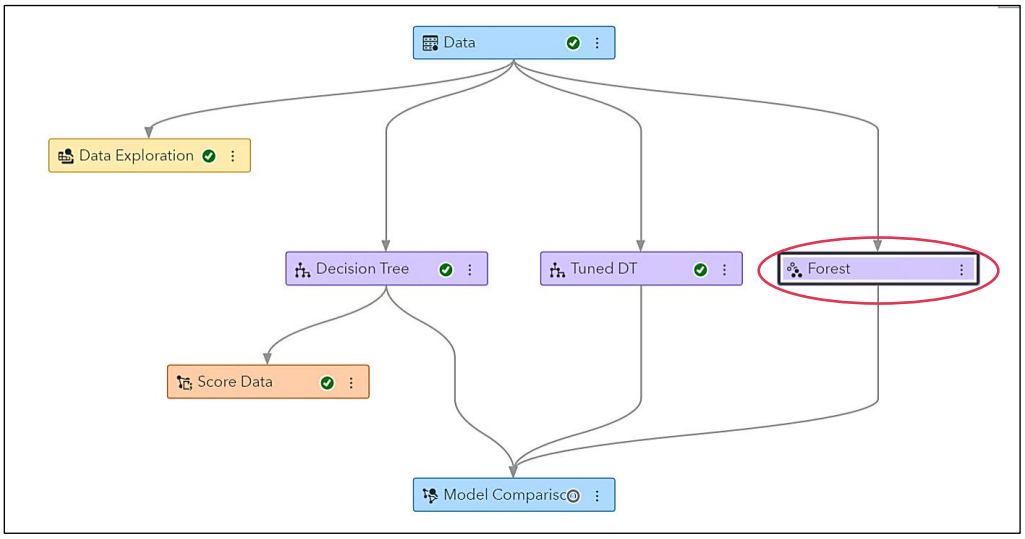
Champi...	Name	Algorith...	KS (You...	Misclas...	Misclas...	Root Av...	Averag...
	Tuned DT	Decision Tree	0.3055	0.3472	0.3472	0.4677	0.2187
	Decision Tree	Decision Tree	0.2364	0.3818	0.3818	0.4760	0.2266

The manually tuned tree is outperforming on all the model assessment statistics.

7. Creating a Default Forest Model

a) *Create a forest model with all the default settings.*

Right-click the **Data** node and select **Add child node ⇨ Supervised Learning ⇨ Forest**.



Click the **Run Pipeline** button.

b) *How many trees are built to create the forest model? What is the out-of-bag sample proportion? What is the out-of-bag misclassification rate for this model?*

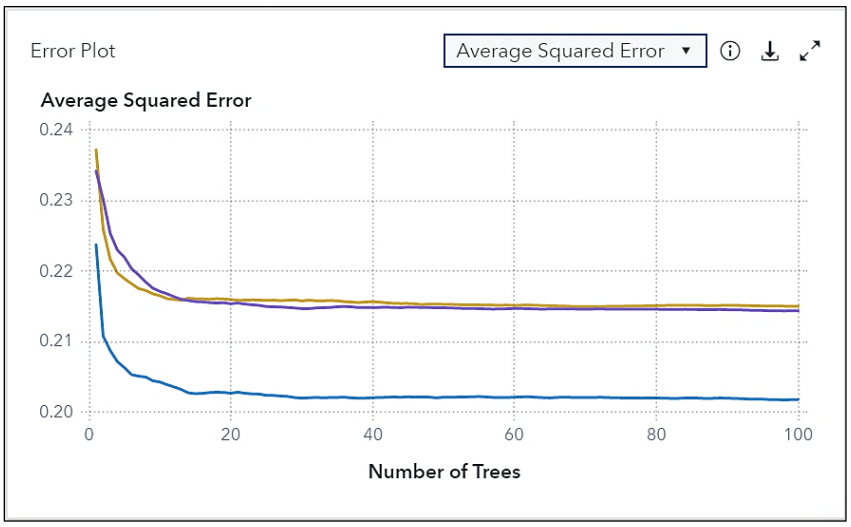
Select the **Results** (of the Forest node) ⇨ **Output** ⇨ (focus on) **Model Information** table.

Model Information	
Number of Trees	100
Number of Variables Per Split	6
Seed	12345
Bootstrap Percentage	60
Number of Bins	50
Number of Input Variables	34
Maximum Number of Tree Nodes	517
Minimum Number of Tree Nodes	261
Maximum Number of Branches	2
Minimum Number of Branches	2
Maximum Depth	20
Minimum Depth	20
Maximum Number of Leaves	259
Minimum Number of Leaves	131
Maximum Leaf Size	3670
Minimum Leaf Size	5
OOB Misclassification Rate	0.34466912
Average Number of Leaves	188.28

By default, a forest model with 100 trees is created. Bootstrap percentage (in-bag sample) is 60, which means that 40% is the out-of-bag sample kept for Validation. The out-of-bag misclassification rate is 0.3447.

c) *Do you think building a forest with one hundred trees is apt? Would increasing or reducing the number of trees be a good move?*

Select the **Results** (of the Forest node) ⇨ **Error Plot**.



The out-of-bag and validation ASE does not improve much beyond 25 trees. Therefore, restricting the forest up to 25 trees might be beneficial.

d) *Which input is the most important predictor for low birth weight?*

Select the **Results** (of the Forest node) ⇨ **Variable Importance** table.

Variable Label	Role	Variable Name	Training Impor...	Importance St...	Relative Importance
Hypertension, preg.	INPUT	HYPERPR	97.8868	42.1060	1
Age of father	INPUT	FAGE	56.7269	8.8936	0.5795
Month of preg. prenatal care began	INPUT	PRENATAL	47.4697	7.2832	0.4849
Eclampsia	INPUT	ECLAMP	42.7838	18.5235	0.4371

Hypertension for this pregnancy (**HYPERPR**) is the most important input.

e) *Which inputs amongst the important ones are the most consistent across all the trees in the forest?*

Select the **Results** (of the Forest node) ⇨ **Variable Importance** table (expand the table).

Right-click the **Importance Standard Deviation** column and select **Sort** ⇨ **Sort (ascending)**.

Scroll to the right, if required. Right-click the **Relative Importance** column and select **Sort** ⇨ **Add to sort (ascending)**.

(*Add to sort* means that the initial sorting done by importance standard deviation still holds, so the sort on relative importance values takes place within each sorted importance standard deviation group).

Variable Label	Role	Variable Name	Training Import...	Importance ... ↑	Relative Importance ↑
drinks per week	INPUT	DRINKNUM	0.8581	1.5193	0.0088
Anemia -- mother	INPUT	ANEMIA	3.3997	1.6357	0.0347
Diabetes	INPUT	DIABETES	4.2966	1.8423	0.0439
Number previous live births now dead	INPUT	BDEAD	3.8515	1.8469	0.0393
Ac/Ch Lung disease	INPUT	ACLUNG	3.6732	2.2188	0.0375
Total pregnancies (including this one)	INPUT	TOTALP	16.2633	2.2392	0.1661
Age of mother	INPUT	MAGE	35.8484	2.5008	0.3662
Prev. infant 4000+gm	INPUT	PINFANT	6.1612	3.1916	0.0629
Education of mother (years)	INPUT	MEDUC	26.0539	3.9330	0.2662
	INPUT	drinker	6.6699	4.1828	0.0681

Total pregnancies including this one (**TOTALP**) and Age of Mother (**MAGE**) are some of the variables that have high relative importance values and low importance standard deviations.

f) *How does the forest model compare to the best ASE and MISC values from previous decision tree models?*

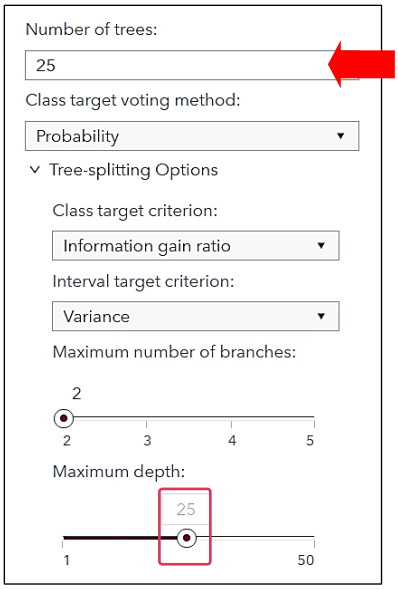
Select the **Results** (of the Model Comparison node) ⇨ **Model Comparison** table.

Champi...	Name	Algorith...	KS (You...	Misclas...	Misclas...	Root Av...	Averag...
★	Forest	Forest	0.3121	0.3439	0.3439	0.4638	0.2151
	Tuned DT	Decision Tree	0.3055	0.3472	0.3472	0.4677	0.2187
	Decision Tree	Decision Tree	0.2414	0.3792	0.3792	0.4752	0.2258

Forest model has lower ASE and MISC values than the decision tree models and comes out to be the best so far.

g) *Although the loss of predictive accuracy might occur, reduce the time for growing the forest model by changing some of the settings (for example, reduce the number of trees, and to compensate for fewer trees, increase the maximum depth a bit).*

Under the Forest options, change the **Number of trees** from 100 to **25**. Under Tree-splitting Options, change the **Maximum depth** from 20 to **25**.



Run the **Forest** node.

h) *Does this shallow forest have a comparable accuracy with the deeper one?*

Select the **Results** (of the Forest node) ⇨ **Output** ⇨ (focus on) **Model Information** table.

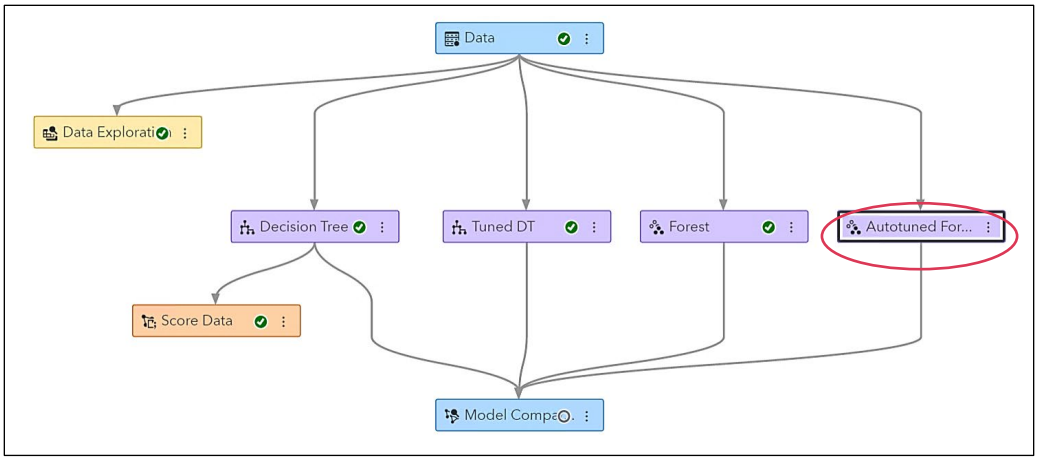
Model Information	
Number of Trees	25
Number of Variables Per Split	6
Seed	12345
Bootstrap Percentage	60
Number of Bins	50
Number of Input Variables	34
Maximum Number of Tree Nodes	675
Minimum Number of Tree Nodes	419
Maximum Number of Branches	2
Minimum Number of Branches	2
Maximum Depth	25
Minimum Depth	25
Maximum Number of Leaves	338
Minimum Number of Leaves	210
Maximum Leaf Size	1759
Minimum Leaf Size	5
OOB Misclassification Rate	0.34400067
Average Number of Leaves	267.8

The OOB (out of bag) MISC (0.3440) is slightly better than the previous analysis (0.3447). With a smaller forest of larger trees, this is possible.

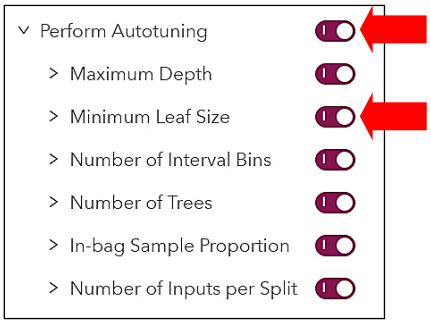
8. Autotuning a Forest Model

a) *Create another forest model and use SAS Viya autotuning capability to autotune all the hyperparameters of this forest.*

Right-click the **Data** node and select **Add child node** ⇨ **Supervised Learning** ⇨ **Forest**. Rename the recently added Forest node as **Autotuned Forest**.



Turn on **Perform Autotuning** using the slider button. Also, turn on the **Minimum leaf size** hyperparameter using the slider button. All other hyperparameters will be autotuned, by default.



Click the **Run Pipeline** button.

b) *Has autotuning done some magic to improve the forest model? How do all the models in the pipeline perform?*

Select the **Results** (of the Model Comparison node) ⇨ **Model Comparison** table.

Champi...	Name	Algorith...	KS (You...	Misclas...	Misclas...	Root Av...	Averag...
★	Autotuned Forest	Forest	0.3219	0.3391	0.3391	0.4633	0.2147
	Forest	Forest	0.3067	0.3467	0.3467	0.4647	0.2159
	Tuned DT	Decision Tree	0.3055	0.3472	0.3472	0.4677	0.2187
	Decision Tree	Decision Tree	0.2414	0.3792	0.3792	0.4752	0.2258

The autotuned forest model comes out to be the best on almost all model performance statistics.

c) *In what ways are the autotuned forest different from the previous forest model?*

Select the **Results** (of the Autotuned Forest node) ⇨ **Autotune Best Configuration** table.

Parameter	Value
Evaluation	53
Number of Trees	150
Number of Variables to Try	26
Bootstrap	0.5000
Maximum Tree Levels	16
Number of Bins	60
Leaf Size	51
Kolmogorov-Smirnov	0.3219

Unlike the previous forest model (25 trees), the autotuned forest has many more trees in it, 150 trees. It uses fewer (26 as against 34) inputs and a smaller in-bag sample (50% as against 60%) proportion. The trees are shallower (16 as against 25), and the number of interval bins are little more (60 as against 50) than the previous forest. The minimum leaf sizes in individual trees are much higher (51 as against 5) in this forest.

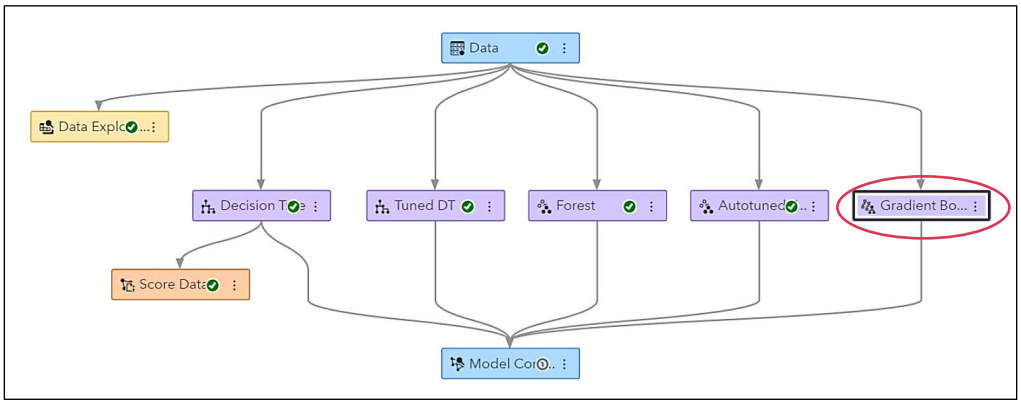
d) *Random forest models can be a little slower to execute than a single tree. However, they are generally particularly good for prediction. They can be difficult (or impossible) to interpret because they are ensembles of many different trees, based on different data and different variables. If model interpretation is inevitable and important, what can you do?*

As machine learning models become more sophisticated, the ability to interpret these models quickly and accurately can diminish. SAS Visual Data and Machine Learning tools such as Partial Dependence (PD) plots, Individual Conditional Expectation (ICE) plots, Local Interpretable Model-Agnostic Explanation (LIME), and Kernel Shapley values (Kernel SHAP), can help you better understand your model. These techniques are model-agnostic, which means that these techniques can be applied to any model that is generated by a supervised learning node.

9. Building a Default Gradient Boosting Model

a) *Create a decision trees gradient boosting machine with all the default settings.*

Right-click the **Data** node and select **Add child node** ⇨ **Supervised Learning** ⇨ **Gradient Boosting**.



Click the **Run Pipeline** button.

b) *Is the gradient boosting model deterministic or stochastic? How many trees does the gradient boosting model have? How many actual trees does the model have?*

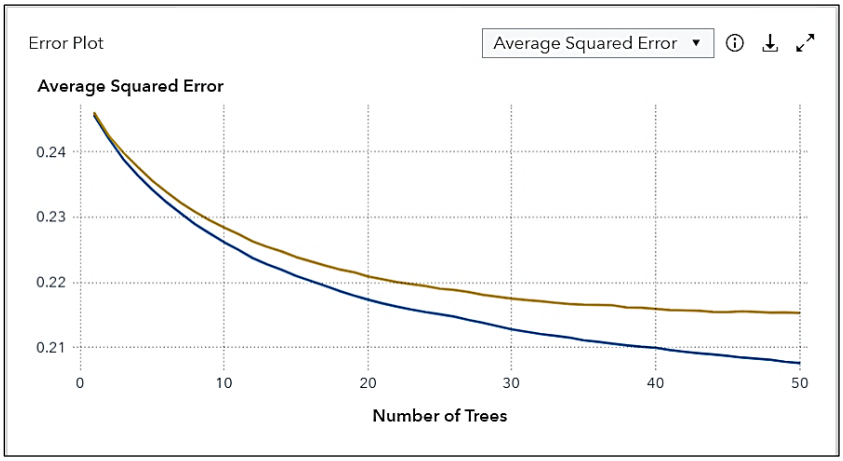
Select the **Results** (of the Gradient Boosting node) ⇨ **Output** ⇨ (focus on) **Model Information** table.

Model Information	
Number of Trees	100
Learning Rate	0.1
Subsampling Rate	0.5
Number of Variables Per Split	34
Number of Bins	50
Number of Input Variables	34
Maximum Number of Tree Nodes	31
Minimum Number of Tree Nodes	17
Maximum Number of Branches	2
Minimum Number of Branches	2
Maximum Depth	4
Minimum Depth	4
Maximum Number of Leaves	16
Minimum Number of Leaves	9
Maximum Leaf Size	5711
Minimum Leaf Size	5
Seed	12345
Lasso (L1) penalty	0
Ridge (L2) penalty	1
Actual Number of Trees	50
Average Number of Leaves	14
Early stopping stagnation	5
Early stopping threshold	0
Early stopping threshold iterations	0
Early stopping tolerance	0

The subsampling rate is 0.5, which means that a stochastic gradient boosting model is created. The gradient boosting model created 100 sequential trees. However, the actual number of trees included in the model is 50.

c) *Should you further increase the number of trees?*

Select the **Results** (of the Gradient Boosting node) ⇨ **Error Plot**.



No. The graph of the average squared error as a function of the number of trees shows that the validation ASE converged around fifty trees and a gradient boosting model beyond that is expected to overfit the data, because the trees are not independent.

Further increasing the number of trees/iterations might result in a better model. You can play around with properties influencing the generalization of the model that includes number of trees, maximum depth,

and learning rate. Larger values of these properties increase the complexity of the model as well as the processing time and might result in overfitting.

d) *Is early stopping performed? What would be the validation misclassification rate of the next four trees beyond the actual number of trees in the gradient boosting model?*

Select the **Results** (of the Gradient Boosting node) ⇨ **Output** ⇨ (focus on) **Fit Statistics** table.

Fit Statistics

Number of Trees	Training Average Square Error	Validation Average Square Error	Training Misclassification Rate	Validation Misclassification Rate	Training Log Loss	Validation Log Loss
1	0.246	0.246	0.378	0.385	0.684	0.685
2	0.242	0.242	0.372	0.378	0.677	0.678
3	0.239	0.240	0.369	0.379	0.671	0.673
⋮						
48	0.208	0.215	0.329	0.343	0.602	0.620
49	0.208	0.215	0.327	0.343	0.602	0.620
50	0.208	0.215	0.327	0.341	0.601	0.619

The default settings show that Perform Early Stopping was enabled, and that Stagnation is set at 5 and Tolerance at 0. The gradient boosting model stopped building more trees at fifty because the model did not improve in the next five consecutive trees. The model stopped at fifty trees, which means the next four trees must have Validation MISC of 0.341.

Note: If your actual number of trees is equal to the default number of trees (100), then this might not be true.

e) *Are there too many variables contributing to the model? Will decreasing the number of inputs to consider per split help in reducing the important variables?*

Select the **Results** (of the Gradient Boosting node) ⇨ **Variable Importance** table.

Variable Label	Role	Variable Name	Training Importance	Importance Standard Deviation	Relative Importance
Hypertension, preg.	INPUT	HYPERPR	9.9508	19.6374	1
Marital status	INPUT	MARITAL	6.7183	28.7604	0.6752
	INPUT	YrsLastLiveBirth	5.8661	5.6801	0.5895
Age of father	INPUT	FAGE	5.5838	8.3047	0.5611
Eclampsia	INPUT	ECLAMP	4.5354	5.6084	0.4558
Hydramnios/Oligo.	INPUT	HYDRAM	4.2003	8.5626	0.4221
Average # of cigarettes daily	INPUT	CIGNUM	4.1490	11.8626	0.4170
Month of preg. prenatal care began	INPUT	PRENATAL	4.0189	3.6849	0.4039
Prev. preterm/small	INPUT	PRETERM	3.7801	5.4538	0.3799
Age of mother	INPUT	MAGE	3.6462	3.1573	0.3664
Education of father (years)	INPUT	FEDUC	3.5594	4.6862	0.3577
	INPUT	smoker	3.2395	13.4361	0.3256
Incompetent cervix	INPUT	CERVIX	2.6456	3.9816	0.2659
Uterine bleeding	INPUT	UTERINE	2.4283	4.8361	0.2440
Education of mother (years)	INPUT	MEDUC	1.3666	2.0167	0.1373
Total pregnancies (including this one)	INPUT	TOTALP	1.3026	1.9652	0.1309
Hypertension, chronic	INPUT	HYPERCH	1.1245	2.6969	0.1130
Number of other terminations	INPUT	TERMS	1.0294	2.0265	0.1035
Ultrasound	INPUT	ULTRA	0.8823	1.7119	0.0887
Number of children now living	INPUT	CHILDREN	0.6702	1.7639	0.0674
Diabetes	INPUT	DIABETES	0.5479	1.0053	0.0551
Anemia -- mother	INPUT	ANEMIA	0.4114	2.0804	0.0413
	INPUT	drinker	0.2975	1.0579	0.0299
Outcome of last delivery	INPUT	LOUTCOME	0.2952	1.1009	0.0297
Prev. infant 4000+ gm.	INPUT	PINFANT	0.1836	0.3563	0.0184
Ad/Ch Lung disease	INPUT	ACLUNG	0.1619	1.9007	0.0163
Amniocentesis	INPUT	AMNIO	0.1551	0.9507	0.0156
Number previous live births now dead	INPUT	BDEAD	0.1424	1.6439	0.0143
Genital herpes	INPUT	HERPES	0.0644	0	0.0065

Yes, almost all variables have contributed to the model.

Number of inputs to consider per split specifies the number of input variables randomly sampled to use per split. Decreasing the number of inputs per split will not necessarily help in reducing the number of important inputs in the model.

f) *What is the learning rate used in the gradient boosting model? How does the learning rate play a role?*

Select the **Results** (of the Gradient Boosting node) ⇨ **Output** ⇨ (focus on) **Model Information** table.

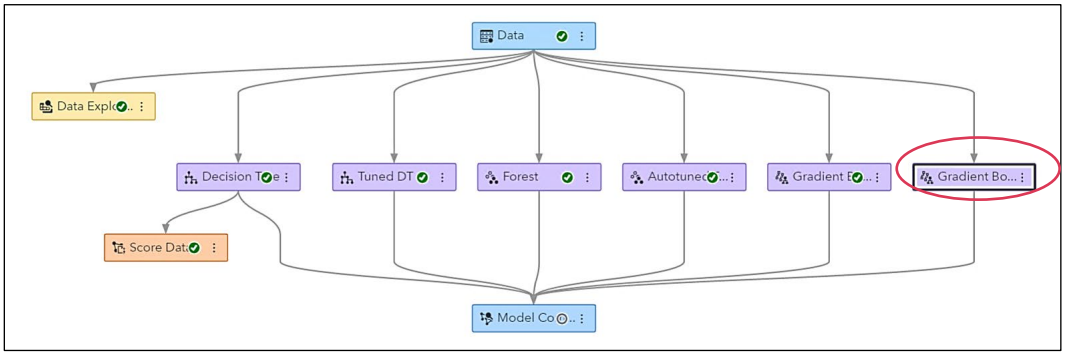
Model Information	
Number of Trees	100
Learning Rate	0.1
Subsampling Rate	0.5

The learning rate controls the step size in the gradient descent optimization algorithm. The default value that is used in the model is 0.1. Smaller values (≤0.1) lead to better generalization (Friedman 2001). Lower values make the model robust to the specific characteristics of tree, thus allowing it to generalize well. A low learning rate would require more trees to model all the relations. A low learning rate is more precise, but calculating the gradient is time-consuming, so it is computationally expensive.

10. Trying Alternative Settings in Gradient Boosting Model

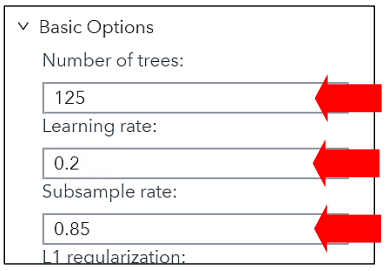
a) *Add another gradient boosting model in the pipeline and rename it as GB Tuned.*

Right-click the **Data** node and select **Add child node ⇨ Supervised Learning ⇨ Gradient Boosting**. Rename the newly added node as **GB Tuned**.

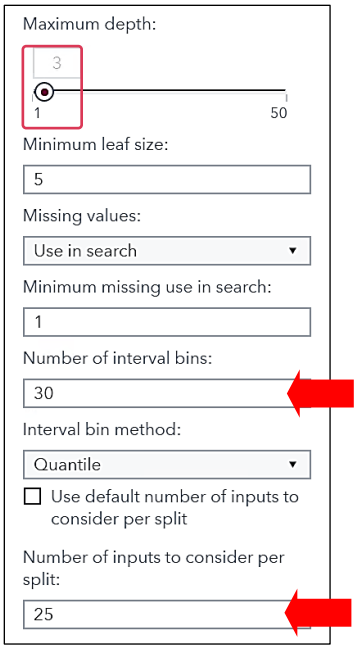


b) *Change certain basic options and tree growth options to improve the gradient boosting model. Knowing that there could not be one single recipe to improve the default gradient boosting model, do you think that increasing the number of trees and sampling rate and modifying the learning rate a bit will benefit the gradient boosting model? How about decreasing the interval bins and tree depths?*

Under Basic Options, change **Number of Trees** from 100 to **125**, **Learning rate** from 0.1 to **0.2**, and **Subsample rate** from 0.5 to **0.85**.



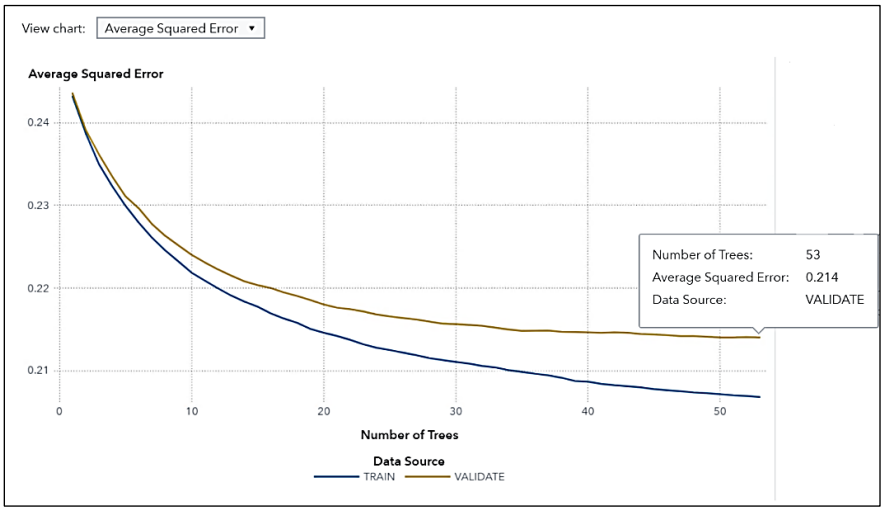
Under Tree Growth Options, change **Maximum depth** from 4 to **3**, **Number of interval bins** from 50 to **30**, deselect **Use default number of inputs to consider per split** box, and change **Number of inputs to consider per split** from 100 to **25**.



Click the **Run Pipeline** button.

c) *Although you have increased the number of trees, how many trees were there in the gradient boosting model? How does the average squared error behave with every tree added in the model?*

Select the **Results** (of the Tuned GB node) ⇨ **Error Plot**.



The gradient boosting model has 53 trees.

The training error decreases as the number of trees increases, but the error for the VALIDATE partition gives an indication of how well our model generalizes.

d) *Based on average square error, which of the stochastic gradient boosting models appear to be better? How do they compare with the decision tree and forest nodes?*

Select the **Results** (of the Model Comparison node) ⇨ **Model Comparison** table.

Champ...	Name	Algorith...	KS (You...	Misclas...	Misclas...	Root Av...	Averag...
★	Tuned GB	Gradient Boosting	0.3230	0.3385	0.3385	0.4626	0.2140
	Autotuned Forest	Forest	0.3219	0.3391	0.3391	0.4633	0.2147
	Gradient Boosting	Gradient Boosting	0.3176	0.3412	0.3412	0.4640	0.2153
	Forest	Forest	0.3067	0.3467	0.3467	0.4647	0.2159
	Tuned DT	Decision Tree	0.3055	0.3472	0.3472	0.4677	0.2187
	Decision Tree	Decision Tree	0.2414	0.3792	0.3792	0.4752	0.2258

The tuned stochastic gradient boosting model comes out to be a champion model across all the models in the pipeline. With more trees than the default gradient boosting model, it has the lower validation average square error and misclassification rate and is the better model.

e) *Which model do you choose for deployment? How do you select a model?*

Even in this stage, the best algorithms might not be the methods that achieve the highest reported accuracy. Most algorithms usually require careful tuning and extensive training to obtain the best achievable performance. Selecting the modeling algorithm for your machine learning application can sometimes be the most difficult part of modeling. The decision about which algorithm to use can be guided by answering a few key questions (Wujek, Hall, and Güneş 2016):

- What is the size and nature of your data?
- What are you trying to achieve with your model? What is the size and nature of your data?
- What are you trying to achieve with your model?
- How accurate does your model need to be?
- How much time do you have to train your model?
- How interpretable or understandable does your model need to be?
- Does your model have automatic hyperparameter tuning capability?

End of Solutions