MIS 6334.501

Advanced Business Intelligence

Prof.Geng

Group 7

Data and Model Analytics Using SAS Enterprise Miner

Project Final Report

Siyang Zan, Kaier Ying, Jingwen Zhang, Xuerong Zhang

10/02/2015

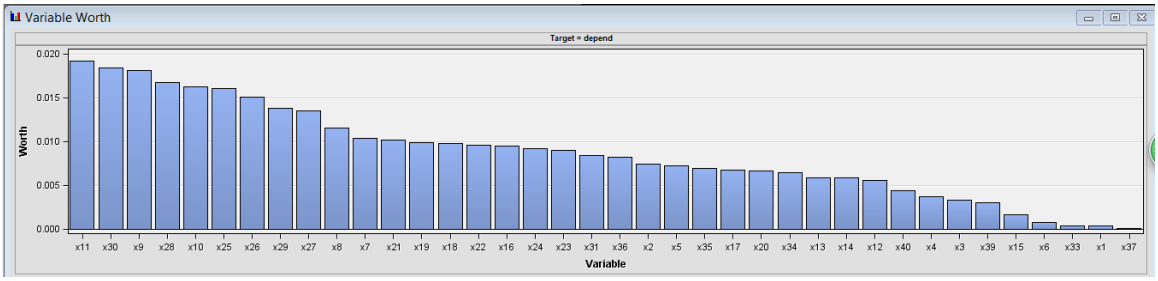
Table of Contents

Part I. Basic Data Preprocessing

Variable Summary

Role	Measurement Level	Frequency Count
INPUT	BINARY	6
INPUT	INTERVAL	29
INPUT	NOMINAL	3
REJECTED	INTERVAL	1
REJECTED	NOMINAL	1
TARGET	BINARY	1

There are 41 variables in this data set, one binary target, 29 interval inputs, 9 nominal inputs and two rejected variables (rejected because of too many missing values).



Among the 38 inputs, x11, x30 and x9 have the highest variable worth and could probably be most useful in predicting the target response.

Class Variable Summary Statistics
(maximum 500 observations printed)

Data Role=TRAIN

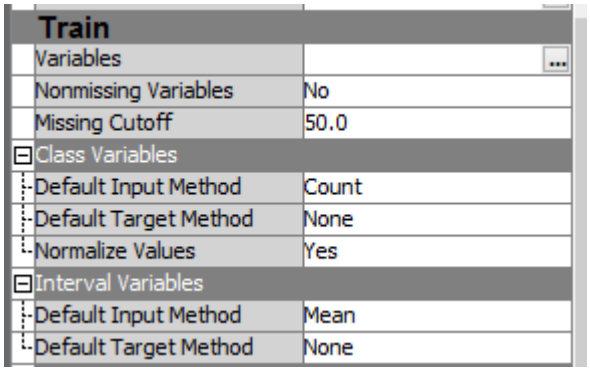
Data Role	Variable Name	Role	Number of Levels	Missing	Mode	Mode Percentage	Mode2	Mode2 Percentage
TRAIN	x1	INPUT	3	99	1	54.95	0	35.25
TRAIN	x12	INPUT	3	99	0	76.63	1	13.56
TRAIN	x15	INPUT	3	80	0	63.66	1	28.42
TRAIN	x3	INPUT	8	102	3	31.58	4	26.73
TRAIN	x33	INPUT	3	97	0	69.41	1	20.99
TRAIN	x37	INPUT	3	98	0	53.27	1	37.03
TRAIN	x4	INPUT	4	114	1	58.61	2	16.04
TRAIN	x5	INPUT	7	93	2	39.50	1	19.31
TRAIN	x6	INPUT	3	91	0	67.72	1	23.27
TRAIN	depend	TARGET	2	0	0	86.83	1	13.17

Interval Variable Summary Statistics
(maximum 500 observations printed)

Data Role=TRAIN

Variable	Role	Mean	Standard Deviation	Non Missing	Missing	Minimum	Median	Maximum	Skewness	Kurtosis
x10	INPUT	14.94204	11.90957	904	106	0	12.2	65.5	1.382825	2.317668
x11	INPUT	8.473841	7.668627	906	104	0	6.8	55.5	1.987097	6.283723
x13	INPUT	19.40284	23.37465	916	94	1	11	186	2.858757	12.09077
x14	INPUT	10.28359	14.10368	908	102	0	5.1	140.9	3.169707	15.40902
x16	INPUT	3.498906	5.632065	914	96	0	2	52	3.878101	20.35713
x17	INPUT	1.777243	0.951776	914	96	0	1.6	9.2	2.935421	13.8178
x18	INPUT	20.83315	33.99782	905	105	0	11	634	7.894892	119.4396
x19	INPUT	775.8246	1192.836	910	100	0	410.9	13395.6	4.343292	28.16146
x2	INPUT	45.73179	12.43386	906	104	21	46	81	0.337771	-0.45896
x20	INPUT	11.49854	5.764969	892	118	0	10.4	50.8	1.874262	7.782233
x21	INPUT	5.542437	3.042854	919	91	0	5	16.6	0.876229	0.5865
x22	INPUT	84.04009	108.946	898	112	0	44	768	2.641412	8.068279
x23	INPUT	2.225137	1.167148	911	99	0	2	8.1	1.415486	3.502093
x24	INPUT	0.417177	0.276381	928	82	0	0.42	1	0.224316	-0.6201
x25	INPUT	0.188733	0.351466	900	110	0	0	1	1.642091	1.032598
x26	INPUT	0.211478	0.31051	920	90	0	0.06	1	1.671832	1.465943
x27	INPUT	0.495647	0.373118	919	91	0	0.46	1	0.173959	-1.4785
x28	INPUT	0.223941	0.313509	921	89	0	0.07	1	1.569225	1.144912
x29	INPUT	0.068753	0.152243	914	96	0	0	1	3.12128	12.06747
x30	INPUT	0.454602	0.31529	917	93	0	0.42	1	0.273401	-0.88565
x31	INPUT	0.244298	0.262094	912	98	0	0.2	1	1.179882	0.935282
x34	INPUT	37.07453	51.23197	899	111	1	18	360	2.878777	9.777604
x35	INPUT	2.191372	2.077111	904	106	1	1	15	2.699675	8.444085
x36	INPUT	17.15263	21.27285	912	98	0	9.5	136.6	2.343703	6.325743
x39	INPUT	0.736546	0.346281	909	101	0	1	1	-0.86582	-0.87081
x40	INPUT	0.731745	0.361464	911	99	0	1	1	-0.90518	-0.82887
x7	INPUT	0.51864	1.042291	912	98	0	0	6	2.891257	9.717365
x8	INPUT	5.721672	4.725806	909	101	0	5	31	1.851554	4.887932
x9	INPUT	56.65987	66.61209	897	113	0	33.5	411.8	2.192934	6.027539

By looking at the Class Variable Summary Statistics and the Interval Variable Summary Statistics, we find there are approximately 100 missing values for each input.



Therefore, for regression and neural network, we replace the missing value with the mean of the non-missing values for the interval input and replace the missing value with the most frequent category for the nominal input.

Here is the result of data partition, 55% training data and 45% validation data.

Partition Summary

Type	Data Set	Number of Observations
DATA	EMWS1.Ids3_DATA	1010
TRAIN	EMWS1.Part3_TRAIN	553
VALIDATE	EMWS1.Part3_VALIDATE	457

Summary Statistics for Class Targets

Data=DATA

Variable	Numeric Value	Formatted Value	Frequency Count	Percent	Label
depend	0	0	877	86.8317	
depend	1	1	133	13.1683	

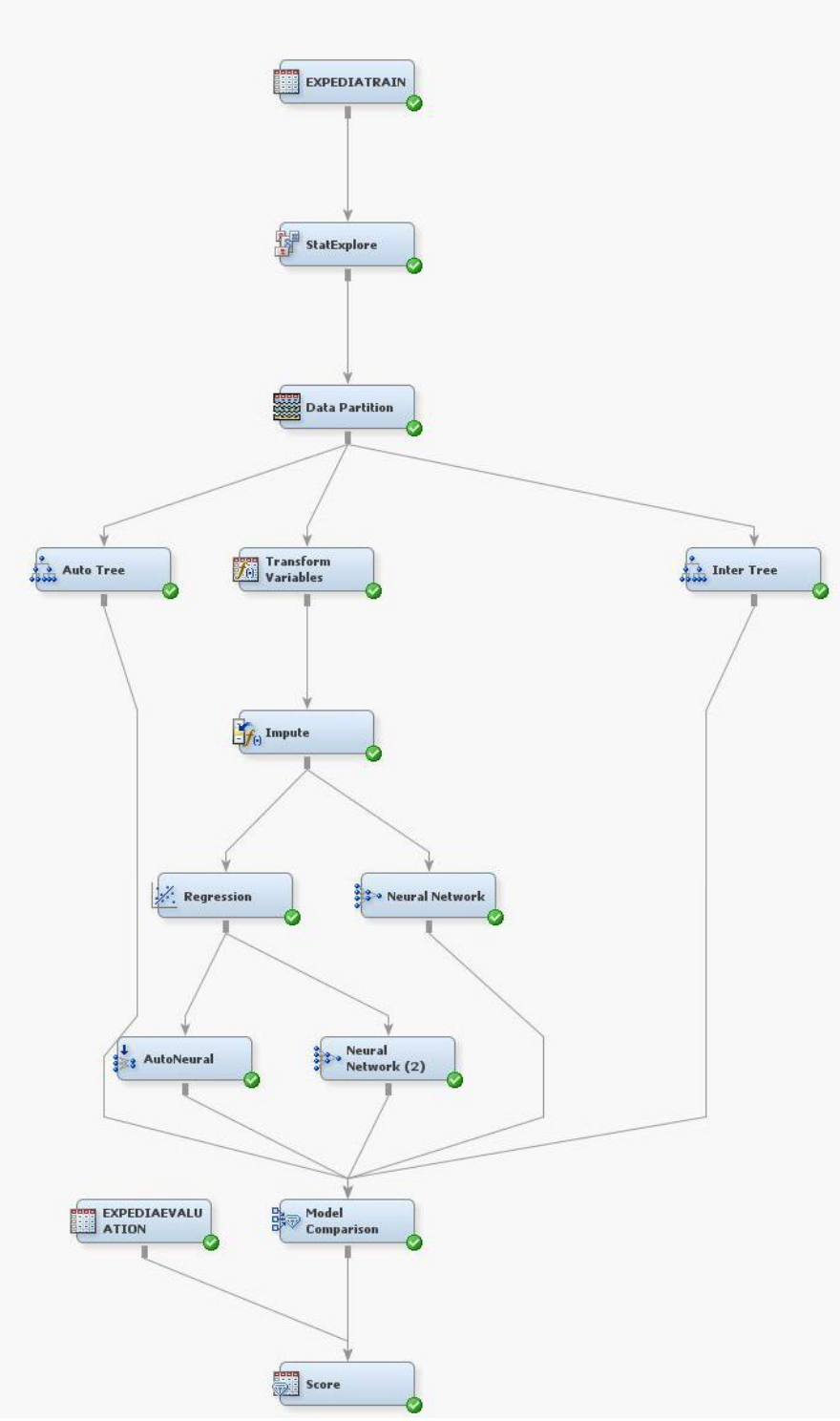
Data=TRAIN

Variable	Numeric Value	Formatted Value	Frequency Count	Percent	Label
depend	0	0	481	86.9801	
depend	1	1	72	13.0199	

Data=VALIDATE

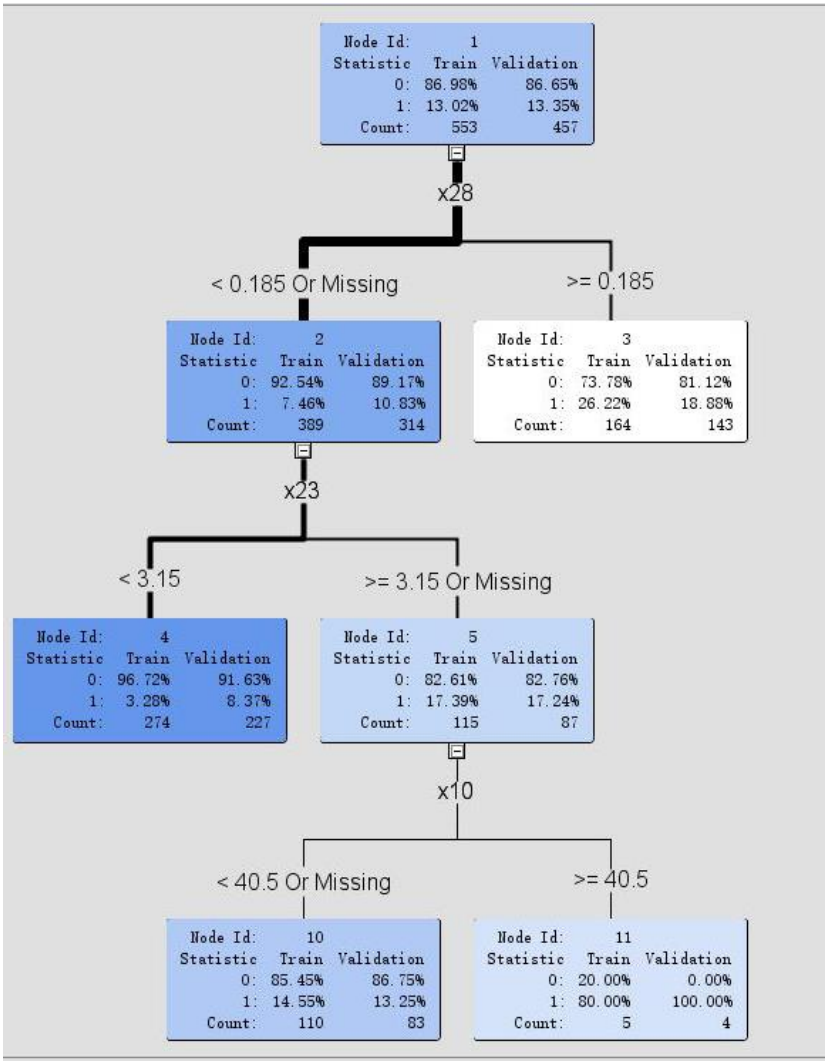
Variable	Numeric Value	Formatted Value	Frequency Count	Percent	Label
depend	0	0	396	86.6521	
depend	1	1	61	13.3479	

Part II. Prediction Models, Model Comparison and Champion Model Evaluation

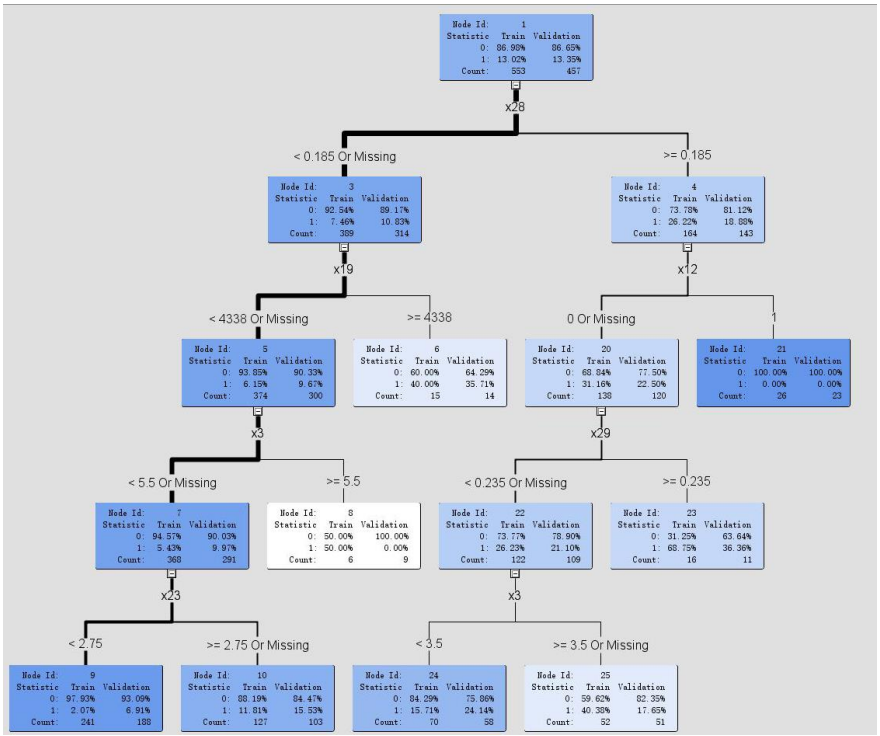


1. Prediction Models

a. Decision Tree



Auto Tree



Interactive Tree

b. Regression

By exploring the whole dataset, we found a plenty of inputs are highly skewed (e.g. x36, x11, x9). Regression and Neural Network models are sensitive to extreme or outlying values in the input space. Inputs with highly skewed or highly kurtotic distributions can be selected over inputs that yield better overall predictions. For these inputs, we use the log transformation to regularize the skewed distributions. By this way, the order of magnitude of the underlying measure predicts the target rather than the measure itself.

Type 3 Analysis of Effects

Effect	DF	Wald Chi-Square	Pr > ChiSq
IMP_x11	1	5.8834	0.0153
IMP_x12	1	0.0185	0.8918
IMP_x23	1	8.1988	0.0042
IMP_x25	1	12.7059	0.0004
IMP_x33	1	13.7629	0.0002
IMP_x4	2	10.1764	0.0062
IMP_x40	1	5.7668	0.0163
IMP_x8	1	6.1084	0.0135

Above is the result of Regression. The selected model, based on the misclassification rate for the validation data, is the model trained in Step 8. It consists of the following effects:

Intercept, IMP_x11, IMP_x12, IMP_x23, IMP_x25, IMP_x33, IMP_x4, IMP_x40, IMP_x8

c. Neural Network

Fit Statistics

Target	Target Label	Fit Statistics	Statistics Label	Train	Validation
depend		_DFT_	Total Degrees of Freedom	553	.
depend		_DFE_	Degrees of Freedom for Error	315	.
depend		_DFM_	Model Degrees of Freedom	238	.
depend		_NW_	Number of Estimated Weights	238	.
depend		_AIC_	Akaike's Information Criterion	853.8933	.
depend		_SBC_	Schwarz's Bayesian Criterion	1880.948	.
depend		_ASE_	Average Squared Error	0.101858	0.114079
depend		_MAX_	Maximum Absolute Error	0.961579	0.961811
depend		_DIV_	Divisor for ASE	1106	914
depend		_NOBS_	Sum of Frequencies	553	457
depend		_RASE_	Root Average Squared Error	0.319151	0.337756
depend		_SSE_	Sum of Squared Errors	112.6544	104.2683
depend		_SUMW_	Sum of Case Weights Times Freq	1106	914
depend		_FPE_	Final Prediction Error	0.255776	.
depend		_MSE_	Mean Squared Error	0.178817	0.114079
depend		_RFPE_	Root Final Prediction Error	0.505743	.
depend		_RMSE_	Root Mean Squared Error	0.422867	0.337756
depend		_AVERR_	Average Error Function	0.341676	0.3854
depend		_ERR_	Error Function	377.8933	352.2555
depend		_MISC_	Misclassification Rate	0.128391	0.142232
depend		_WRONG_	Number of Wrong Classifications	71	65

Neural Network from Impute Node (238 estimated weights, 14.2% misclassification rate)

Fit Statistics

Target	Target Label	Fit Statistics	Statistics Label	Train	Validation
depend		_DFT_	Total Degrees of Freedom	553	.
depend		_DFE_	Degrees of Freedom for Error	519	.
depend		_DFM_	Model Degrees of Freedom	34	.
depend		_NW_	Number of Estimated Weights	34	.
depend		_AIC_	Akaike's Information Criterion	357.3194	.
depend		_SBC_	Schwarz's Bayesian Criterion	504.0416	.
depend		_ASE_	Average Squared Error	0.079421	0.098965
depend		_MAX_	Maximum Absolute Error	0.984675	0.994405
depend		_DIV_	Divisor for ASE	1106	914
depend		_NOBS_	Sum of Frequencies	553	457
depend		_RASE_	Root Average Squared Error	0.281817	0.314587
depend		_SSE_	Sum of Squared Errors	87.83936	90.45424
depend		_SUMW_	Sum of Case Weights Times Freq	1106	914
depend		_FPE_	Final Prediction Error	0.089827	.
depend		_MSE_	Mean Squared Error	0.084624	0.098965
depend		_RFPE_	Root Final Prediction Error	0.299711	.
depend		_RMSE_	Root Mean Squared Error	0.290901	0.314587
depend		_AVERR_	Average Error Function	0.261591	0.358533
depend		_ERR_	Error Function	289.3194	327.6991
depend		_MISC_	Misclassification Rate	0.108499	0.131291
depend		_WRONG_	Number of Wrong Classifications	60	60

Neural Network from Regression Node (34 estimated weights, 13.1% misclassification rate)

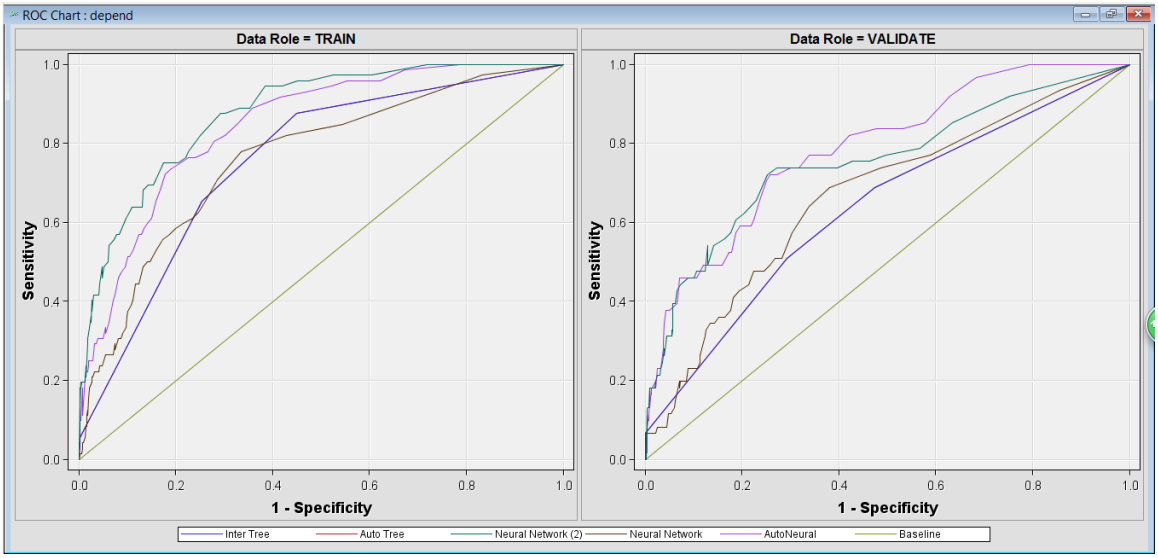
Fit Statistics

Target	Target Label	Fit Statistics	Statistics Label	Train	Validation
depend		_DFT_	Total Degrees of Freedom	553	.
depend		_DFE_	Degrees of Freedom for Error	530	.
depend		_DFM_	Model Degrees of Freedom	23	.
depend		_NW_	Number of Estimated Weights	23	.
depend		_AIC_	Akaike's Information Criterion	379.1302	.
depend		_SBC_	Schwarz's Bayesian Criterion	478.3834	.
depend		_ASE_	Average Squared Error	0.094051	0.099492
depend		_MAX_	Maximum Absolute Error	0.979832	0.984474
depend		_DIV_	Divisor for ASE	1106	914
depend		_NOBS_	Sum of Frequencies	553	457
depend		_RASE_	Root Average Squared Error	0.306677	0.315424
depend		_SSE_	Sum of Squared Errors	104.0203	90.93587
depend		_SUMW_	Sum of Case Weights Times Freq	1106	914
depend		_FPE_	Final Prediction Error	0.102214	.
depend		_MSE_	Mean Squared Error	0.098132	0.099492
depend		_RFPE_	Root Final Prediction Error	0.319709	.
depend		_RMSE_	Root Mean Squared Error	0.313261	0.315424
depend		_AVERR_	Average Error Function	0.301203	0.339234
depend		_ERR_	Error Function	333.1302	310.0595
depend		_MISC_	Misclassification Rate	0.133816	0.12035
depend		_WRONG_	Number of Wrong Classifications	74	55

AutoNeural (23 estimated weights, 12% misclassification rate)

From this, we can see by using the variable selected by Regression, we can reduce the estimated weights and misclassification rate.

2. Model Comparison



Fit Statistics

Selected Model	Predecessor Node	Model Node	Model Description	Target Variable	Target Label	Selection Criterion: Valid: Misclassification Rate
Y	AutoNeural	AutoNeural	AutoNeural	depend		0.12035
	Tree	Tree	Auto Tree	depend		0.124726
	Tree3	Tree3	Inter Tree	depend		0.124726
	Neural2	Neural2	Neural Net...	depend		0.131291
	Neural	Neural	Neural Net...	depend		0.142232

Thus, AutoNeural has the lowest misclassification rate of 12% (Based on 13.2% benchmark).

3. Model Evaluation

Event Classification Table
Model Selection based on Valid: Misclassification Rate (_VMISC_)

Model Node	Model Description	Data Role	Target	Target Label	False Negative	True Negative	False Positive	True Positive
Tree	Auto Tree	TRAIN	depend		68	480	1	4
Tree	Auto Tree	VALIDATE	depend		57	396	.	4
Tree3	Inter Tree	TRAIN	depend		61	476	5	11
Tree3	Inter Tree	VALIDATE	depend		57	389	7	4
Neural	Neural Network	TRAIN	depend		61	471	10	11
Neural	Neural Network	VALIDATE	depend		57	388	8	4
Neural2	Neural Network (2)	TRAIN	depend		51	472	9	21
Neural2	Neural Network (2)	VALIDATE	depend		48	384	12	13
AutoNeural	AutoNeural	TRAIN	depend		68	475	6	4
AutoNeural	AutoNeural	VALIDATE	depend		52	393	3	9

From the output of the Comparison node, we found the Event Classification Table. In this table, we can see the source of error: false positives and false negatives. And as we know, the cost of misclassifying 1 as 0(false negatives) is 5, the cost of misclassifying 0 as 1*(false positives) is 1, so we should calculate the total cost by 5*(# of false negatives) plus 1*(# of false positives) and we can see that neural network 2 has the minimum number of total cost, which is 264 for the train data and 252 for the validation data.

Here is the result of scoring the new evaluation dataset using the champion model.

Data Role=SCORE Output Type=CLASSIFICATION

Variable	Numeric Value	Formatted Value	Frequency Count	Percent
I_depend	.	0	1995	94.5050
I_depend	.	1	116	5.4950

Data Role=TRAIN Output Type=CLASSIFICATION

Variable	Numeric Value	Formatted Value	Frequency Count	Percent
I_depend	.	0	523	94.5750
I_depend	.	1	30	5.4250

Data Role=VALIDATE Output Type=CLASSIFICATION

Variable	Numeric Value	Formatted Value	Frequency Count	Percent
I_depend	.	0	432	94.5295
I_depend	.	1	25	5.4705

Part III. Improve Model Performance

By exploring the dataset, we found that among the total 1010 observations, there are 133 records with the target value of 1, which means the benchmark of our final prediction should be approximately 13.2%. However, the misclassification rate of our champion model is 12%, one percent lower than the benchmark. Thus, we tried to further improve our models by implementing several methods from different perspectives. Through some adjustment and manipulation on the data, we improved the misclassification rate of our champion model from 12% (based on benchmark of 13.2%) to 14.5% (based on benchmark of 26.7%). We will demonstrate our adjustment in detail in the later part.

Fit Statistics

Selected Model	Predecessor Node	Model Node	Model Description	Target Variable	Target Label	Selection Criterion: Valid: Misclassification Rate
Y	AutoNeural	AutoNeural	AutoNeural	depen		0.12035
	Tree	Tree	Auto Tree	depend		0.124726
	Neural2	Neural2	Neural Net...	depend		0.131291
	Tree3	Tree3	Inter Tree	depend		0.140044
	Neural	Neural	Neural Net...	depend		0.142232

Target Variable

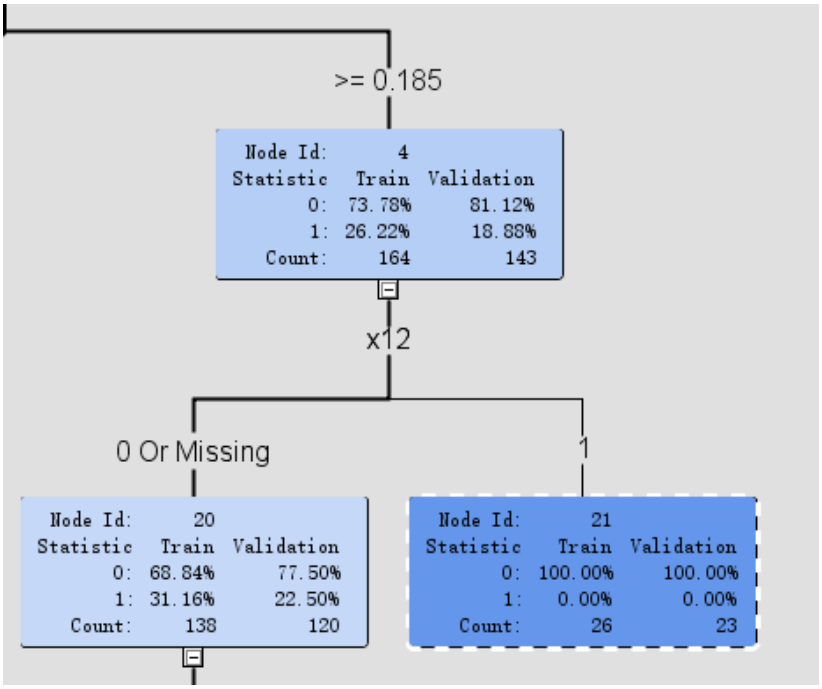
Before Adjustment (Benchmark 13.2%)

Selected Model	Predecessor Node	Model Node	Model Description	Target Variable	Target Label	Selection Criterion: Valid: Misclassification Rate
Y	Ensmbl	Ensmbl	Ensemble	EM_Outcome		0.145055
	AutoNeural	AutoNeural	AutoNeural	EM_Outcome		0.158242
	Neural	Neural	Neural Net...	EM_Outcome		0.158242
	Neural2	Neural2	Neural Net...	EM_Outcome		0.16044
	Reg	Reg	Regression	EM_Outcome		0.164835
	Tree3	Tree3	Inter Tree	EM_Outcome		0.173626
	Tree2	Tree2	Auto Tree	EM_Outcome		0.18022

After Adjustment (Benchmark 26.7%)

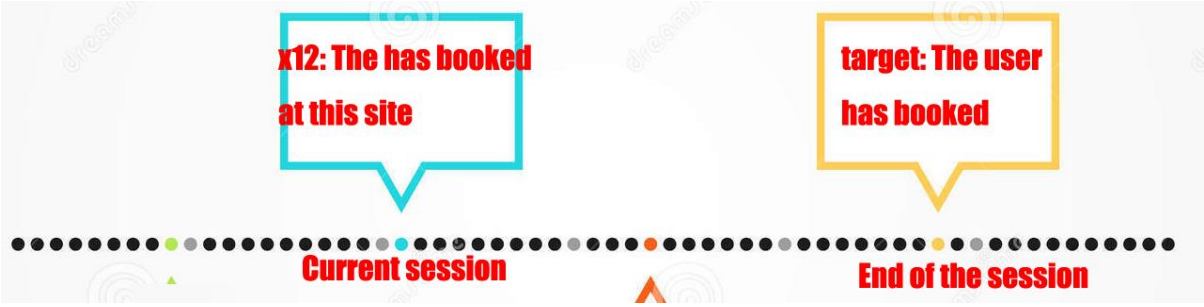
1. Data Modification and Input Selection

By looking into the interactive tree we made, we found a leaf perfectly classified the target. When x12 is 1, all the value of the target is 0, both for train data and validation data.

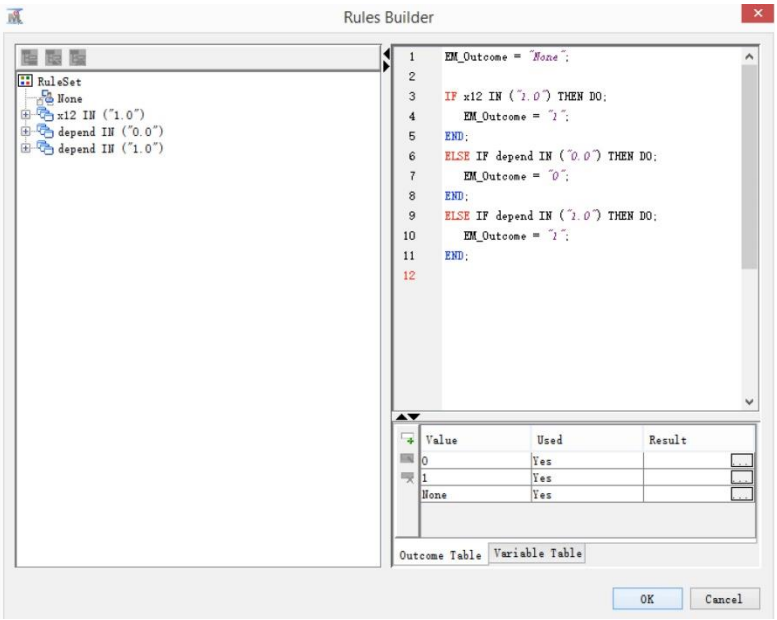


By further looking into the variable description, we found x12 indicates whether the user

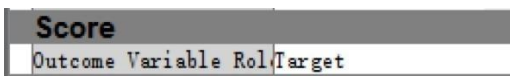
has booked at this site up to this point in the current session. And our goal is to predict whether the user is going to book in the remainder of the session. This explains why all the target value is 0 when x12 is 1. Therefore, we should exclude this trouble input to eliminate this noise. However, we could not easily delete the entire row because other inputs in the record are useful for our prediction. Also, directly deleting the x12 is not appropriate. Because when x12 is 1, the user has already made a book in this session at this time point, if we do not need to consider x12 as an input, then the target value should be 1 instead of 0.



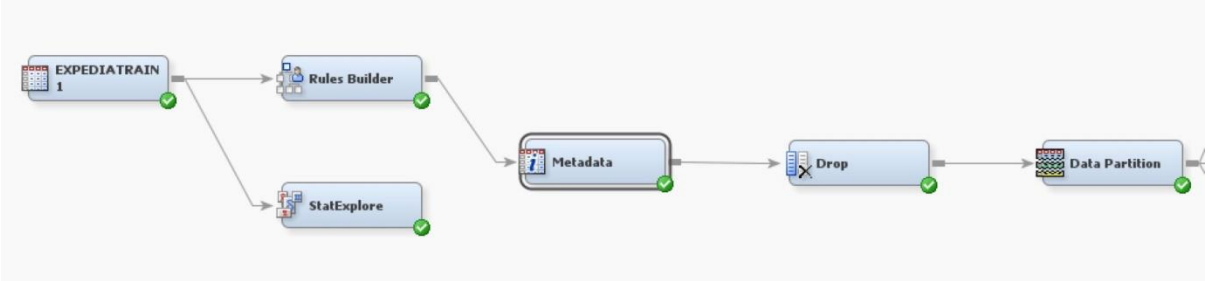
So when x12 is 1, we should first change all the target value from 0 to 1 then drop x12. To achieve this, we could edit the dataset directly in excel, but it is unrealistic if the dataset is very large. We could use SAS code to achieve this, too. Actually, in SAS Enterprise Miner, we could use rule builder node to complete this process and we can easily see how we modified the original dataset as well as edit the modification if we want.



After this step, we got a new input called EM_Outcome and we should use this new input as our new target. We should note that after this target change, we got 270 records with target value of 1 among all the 1010 observations. Thus, the new benchmark for the modified dataset should be 26.7%.



However, EM_Outcome is a nominal target and we want to change this to binary so that it would fit all the classifiers used in the next step. Then we use Metadata to change the new target from nominal to binary. And we use Drop node to exclude the former target and x12.



Similarly, x33(whether the user has booked at any sites up to this point in the current session) is also a potential trouble input. From the description of x33, we can esily find x33 should include x12, which means if x33 is 1, then x12 must be 1 and target value must be 0. However, in the data set, we found the values of these 3 variables are contradictory. But we are not sure it is because of inproper variable description or wrong data collection.

2. Data Imputation

For the three classifiers we used, only the data for regression and neural network need to be imputed. Because decision tree could categorized the missing value in either side of the leaves, which means there is no difference between the imputed data and the original one.

3. Transform Variables

By exploring the whole dataset, we found a plenty of inputs are highly skewed (e.g. x36, x11, x9). Regression and Neural Network models are sensitive to extreme or outlying values in the input space. Inputs with highly skewed or highly kurtotic distributions can be selected over inputs that yield better overall predictions. For these inputs, we use the log transformation to regularize the skewed distributions. By this way, the order of magnitude of the underlying measure predicts the target rather than the measure itself.

(none) not Equal to ...

Columns: Label Mining Basic

Name	Method	Number of Bins	Role	Level
x31	Default	4	Input	Interval
x30	Default	4	Input	Interval
x33	Default	4	Input	Binary
x27	Default	4	Input	Interval
x37	Default	4	Input	Binary
x3	Default	4	Input	Nominal
x15	Default	4	Input	Binary
x6	Default	4	Input	Binary
x5	Default	4	Input	Nominal
x1	Default	4	Input	Binary
EM_Outcome	Default	4	Target	Binary
x2	Default	4	Input	Interval
x4	Default	4	Input	Nominal
x17	Default	4	Input	Interval
x40	Default	4	Input	Interval
x9	Log	4	Input	Interval
x34	Log	4	Input	Interval
x36	Log	4	Input	Interval
x7	Log	4	Input	Interval
x8	Log	4	Input	Interval
x35	Log	4	Input	Interval
x16	Log	4	Input	Interval
x14	Log	4	Input	Interval
x18	Log	4	Input	Interval
x11	Log	4	Input	Interval
x10	Log	4	Input	Interval
x13	Log	4	Input	Interval
x28	Log	4	Input	Interval
x26	Log	4	Input	Interval
x29	Log	4	Input	Interval
x19	Log	4	Input	Interval
x22	Log	4	Input	Interval

4. Data with high variance

We found the attribute values of some inputs have a high variance. According to what we have learned, we could use the replacement node to eliminate some extreme value or outlier. However, after using this node, we found that the performance of all the classifiers did not improve as we expect.

Selected Model	Predecessor Node	Model Node	Model Description	Target Variable	Target Label	Selection Criterion: Valid: Misclassification Rate
Y	Neural	Neural	Neural Net...	EM_Outcome		0.169231
	AutoNeural	AutoNeural	AutoNeural	EM_Outcome		0.171429
	Reg	Reg	Regression	EM_Outcome		0.173626
	Neural2	Neural2	Neural Net...	EM_Outcome		0.175824
	Tree2	Tree2	Auto Tree	EM_Outcome		0.18022
	Tree3	Tree3	Inter Tree	EM_Outcome		0.18022

Therefore, we further look into the replacement result. The reason why the model performances are being worse is that most of these inputs have some extent of correlations with each other. For example, we change total 21 observations of x19 (total minutes of all sites) with missing value, and after imputation, these missing values became the average of all the non-missing values. However, this change will make some other inputs invalid

because they have some kind of relations. In this situation, the value of x28 (minutes spent at this site divided by total minutes of all sites) will not match with the new value of x19 if we only change x19. Therefore, this kind of replacement would mess up other related inputs. Similarly, like x29 (number of sessions start with this site divided by total sessions of this site), x17 (average sessions per site), all the change we conducted on these inputs will affect the effectiveness of other related inputs. Because of these implicit correlations between different inputs, we cannot rashly replace the outlier with missing value. Thus, we cannot use the replacement node to improve our model performance.

Replacement Counts

Obs	Variable	Label	Role	Train	Validation
1	LOG_x10	Transformed x10	INPUT	0	0
2	LOG_x11	Transformed x11	INPUT	0	0
3	LOG_x13	Transformed x13	INPUT	0	0
4	LOG_x14	Transformed x14	INPUT	0	0
5	LOG_x16	Transformed x16	INPUT	2	0
6	LOG_x18	Transformed x18	INPUT	1	0
7	LOG_x19	Transformed x19	INPUT	11	10
8	LOG_x22	Transformed x22	INPUT	0	0
9	LOG_x26	Transformed x26	INPUT	0	0
10	LOG_x28	Transformed x28	INPUT	10	9
11	LOG_x29	Transformed x29	INPUT	8	9
12	LOG_x34	Transformed x34	INPUT	0	0
13	LOG_x35	Transformed x35	INPUT	7	6
14	LOG_x36	Transformed x36	INPUT	0	0
15	LOG_x7	Transformed x7	INPUT	9	7
16	LOG_x8	Transformed x8	INPUT	0	0
17	LOG_x9	Transformed x9	INPUT	0	0
18	x17	x17	INPUT	17	6
19	x2	x2	INPUT	0	0
20	x20	x20	INPUT	8	8
21	x21	x21	INPUT	5	4
22	x23	x23	INPUT	11	4
23	x24	x24	INPUT	0	0
24	x25	x25	INPUT	0	0
25	x27	x27	INPUT	0	0
26	x30	x30	INPUT	0	0
27	x31	x31	INPUT	0	0
28	x39	x39	INPUT	0	0
29	x40	x40	INPUT	0	0

Total minutes of all sites

No. of sessions start with this site/total sessions of this site

Average blabla

5. Data Sampling

There are only 1010 observations in this dataset, we do not think this is a large dataset and there is no need for us to do an extra sampling job. If the dataset is too small, it will lead to overfitting problem. However, we still tried to do sampling and see whether it would help the model performance. Below is the result of sampling for 20% and 50% respectively, we

can see the performance get even worse and it proves the conclusion we made before.

Selected Model	Predecessor Node	Model Node	Model Description	Target Variable	Target Label	Selection Criterion: Valid: Misclassification Rate
Y	Tree2	Tree2	Auto Tree	EM_Outcome		0.173913
	Tree3	Tree3	Inter Tree	EM_Outcome		0.173913
	Neural2	Neural2	Neural Net...	EM_Outcome		0.173913
	AutoNeural	AutoNeural	AutoNeural	EM_Outcome		0.173913
	Reg	Reg	Regression	EM_Outcome		0.184783
	Neural	Neural	Neural Net...	EM_Outcome		0.26087

20% Sample

Selected Model	Predecessor Node	Model Node	Model Description	Target Variable	Target Label	Selection Criterion: Valid: Misclassification Rate
Y	Tree2	Tree2	Auto Tree	EM_Outcome		0.173913
	Tree3	Tree3	Inter Tree	EM_Outcome		0.173913
	AutoNeural	AutoNeural	AutoNeural	EM_Outcome		0.195652
	Reg	Reg	Regression	EM_Outcome		0.2
	Neural2	Neural2	Neural Net...	EM_Outcome		0.204348
	Neural	Neural	Neural Net...	EM_Outcome		0.23913

50% Sample

6. Ensemble

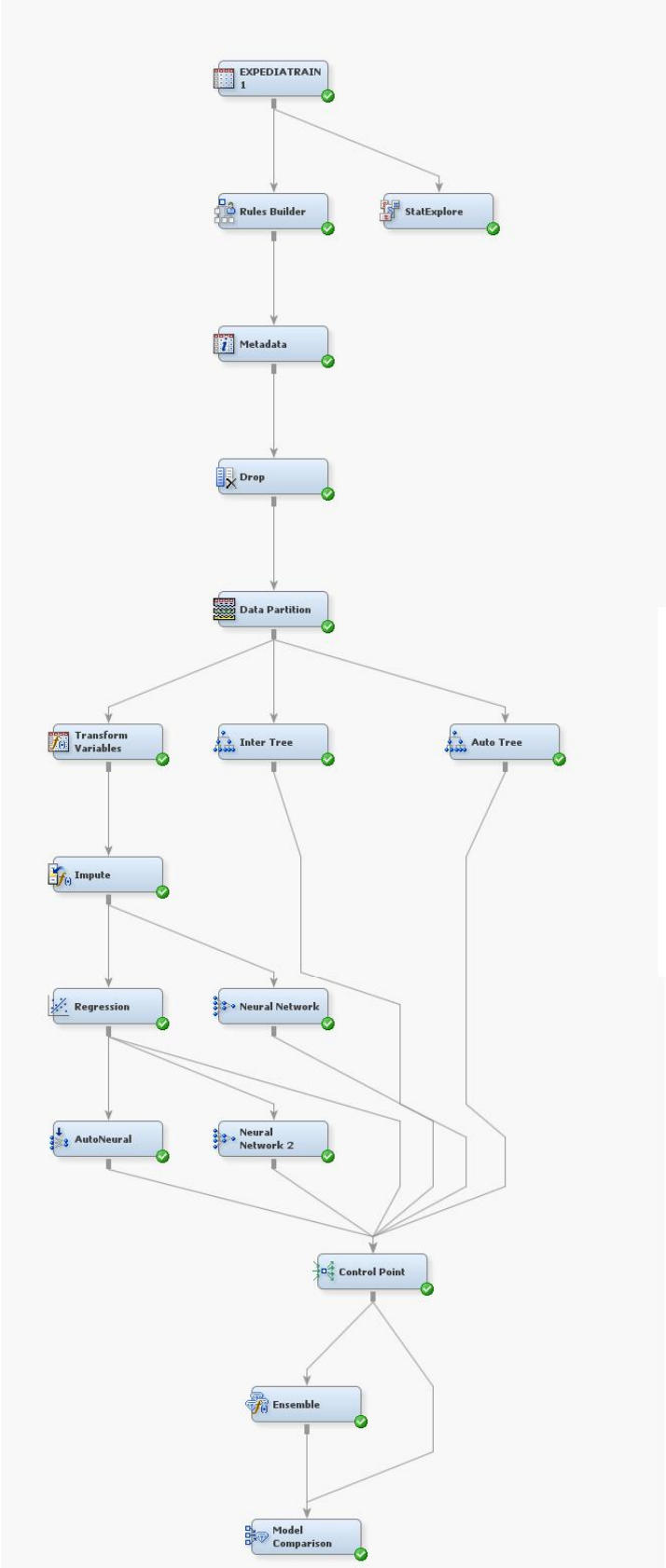
The Ensemble node creates a new model by combining the predictions from multiple models, when the predictions are decisions, this is done by voting. The average method averages the prediction estimates from the models that decide the primary outcome and ignores any model that decides the secondary outcome.

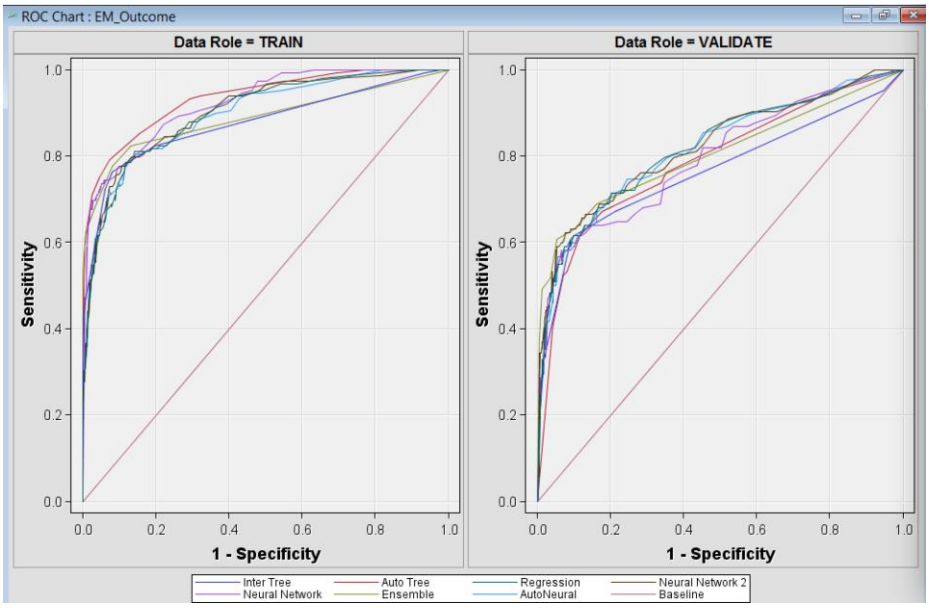
Class Target	
Posterior Probabilit:	Voting
Voting Posterior Prol	Proportion

From the result of model comparison, Ensemble became the champion model.

Selected Model	Predecessor Node	Model Node	Model Description	Target Variable	Target Label	Selection Criterion: Valid: Misclassification Rate
Y	Ensmbl	Ensmbl	Ensemble	EM_Outcome		0.145055
	AutoNeural	AutoNeural	AutoNeural	EM_Outcome		0.158242
	Neural	Neural	Neural Net...	EM_Outcome		0.158242
	Neural2	Neural2	Neural Net...	EM_Outcome		0.16044
	Reg	Reg	Regression	EM_Outcome		0.164835
	Tree3	Tree3	Inter Tree	EM_Outcome		0.173626
	Tree2	Tree2	Auto Tree	EM_Outcome		0.18022

Part IV. Summary





Selected Model	Predecessor Node	Model Node	Model Description	Target Variable	Target Label	Selection Criterion: Valid: Misclassification Rate
Y	Ensmbl	Ensmbl	Ensemble	EM_Outcome		0.145055
	AutoNeural	AutoNeural	AutoNeural	EM_Outcome		0.158242
	Neural	Neural	Neural Net...	EM_Outcome		0.158242
	Neural2	Neural2	Neural Net...	EM_Outcome		0.16044
	Reg	Reg	Regression	EM_Outcome		0.164835
	Tree3	Tree3	Inter Tree	EM_Outcome		0.173626
	Tree2	Tree2	Auto Tree	EM_Outcome		0.18022

To sum up, we have learned a lot throughout the whole project.

1) In both data preprocessing stage and model improvement stage, we should look into every variable in the dataset. We should drop the inputs which have correlations with our target. For instance, x12 in this dataset is a trouble maker which we need to drop. However, sometimes it is very difficult to discover the trouble variable directly by looking at the variable itself. We could use our classifier to help us to focus on the suspicious variable, especially when these variables lead to a perfect result; we need to be cautious and strict when using them as inputs. When decided to exclude an input, we found Rule Builder node is a very convenient tool for data modification.
2) When dealing with the inputs with high variance, we need to consider if we made the modification to these inputs, will other related inputs be affected and thus distort the whole dataset. For example, we cannot just use missing value to replace some outliers in some special inputs but keep everything else unchanged. In this project, if we changed the variables representing total minutes, average sessions and percentages, we need to adjust other related inputs correspondingly. Otherwise the whole dataset would be distorted and the classifier performances would get worse.
3) As for the skewed data, we need to think about whether these statistical properties will affect our model performance in what extent. We should take some methods to optimize our dataset so that it could be fit properly for our classifiers.

4) In the classifier adjustment, we also learned some tricks through this project. For interactive tree, we do not need to select the inputs with the highest logworth, but we need to consider the variance between training data and validation data at the same time. For the neutral network, we could consider AutoNeural when the Neural Network tool did not perform well. Because the neural network models that you obtain with the AutoNeural and Neural Network tools are different, even if both networks have the same number of hidden units. When selecting inputs for neural network, we could use the regression node to select the variables.