SAS Enterprise Miner

5.1 Data preparation

Before data analytics, 'Total_spent' by customers need to be binned into different categories. Here I use Excel as it's simple and clear for binning.

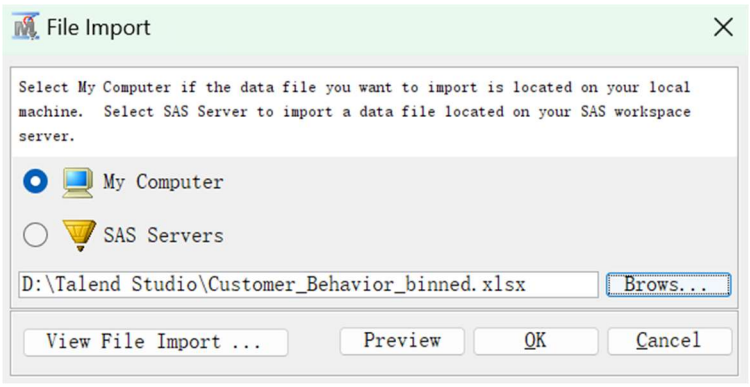
The Total_Spent of 1468 rows are distributed as follows, so I divide them into three categories based on the range:

Total Spent Range	Number of customers	Category
⩽ 1000	507	1
1000 < , ⩽ 3000	470	2
> 3000	491	3

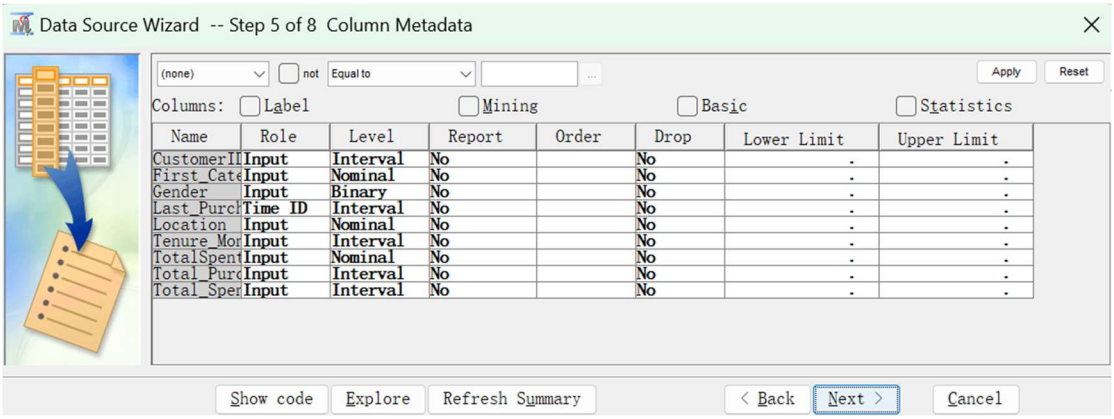
A	B	C	D	E	F	G	H	I	J	K
CustomerID	Gender	Location	Total_Purchases	Total_Spent	First_Category	Last_Purchase_Date	Tenure_Months	TotalSpent_Categories		
16353	M	Washington DC	13	843.8	Nest-USA	2019-12-20	6	=IF(E2<=1000,1,IF(E2<=3000,2,IF(E2>3000,3,0)))		
14307	F	California	79	7622.34	Apparel	2019-09-23	28	IF(logical_test, [value_if_true], **[value_if_false]**)		
14309	F	New Jersey	22	817.73	Bags	2019-12-15	49	1		
16359	F	New York	5	62.07	Headgear	2019-09-04	49	1		
14312	M	Chicago	23	3213.7	Apparel	2019-12-12	27	3		
16365	F	California	7	484.09	Office	2019-06-09	49	1		
16367	M	California	37	1293.83	Apparel	2019-05-06	20	2		
14320	F	Chicago	34	2823.66	Apparel	2019-10-01	29	2		
14321	F	Washington DC	29	2182.59	Apparel	2019-08-14	24	2		
14329	F	California	76	4147.01	Nest-USA	2019-09-02	22	3		
16378	F	Chicago	10	1766.29	Nest-USA	2019-12-12	4	2		
14334	M	California	42	2980.63	Google	2019-10-13	41	2		
16385	M	California	21	788.36	Office	2019-04-04	19	1		
16387	F	California	4	27.97	Drinkware	2019-10-02	20	1		
14341	F	Chicago	24	1586.26	Lifestyle	2019-11-03	3	2		
14344	F	New York	14	1328.59	Nest-USA	2019-06-22	15	2		
16393	F	Chicago	22	1314.52	Apparel	2019-05-06	20	2		
16395	M	Washington DC	60	6562.08	Drinkware	2019-09-26	18	3		
16401	M	Chicago	14	728.72	Apparel	2019-09-05	40	1		
16402	M	Chicago	57	5175.15	Lifestyle	2019-02-24	31	3		
14355	F	New York	11	1270.43	Drinkware	2019-04-13	7	2		
16403	F	California	10	765.38	Apparel	2019-11-30	26	1		
16405	F	California	3	38.22	Apparel	2019-12-15	20	1		
16407	M	California	20	1203.76	Apparel	2019-11-30	48	2		
16409	F	Chicago	48	5066.26	Apparel	2019-11-03	45	3		
16411	F	Chicago	21	862.27	Drinkware	2019-02-23	40	1		
14368	F	New York	8	1553.99	Nest-USA	2019-10-12	4	2		
16419	F	California	17	946.56	Lifestyle	2019-12-06	50	1		
14373	F	Chicago	4	148.46	Apparel	2019-06-20	40	1		
16422	F	California	54	4336.18	Nest-USA	2019-12-07	35	3		

The binned dataset is saved as Customer_Behavior_binned.xlsx.

Then I import the file in SAS Enterprise Miner:



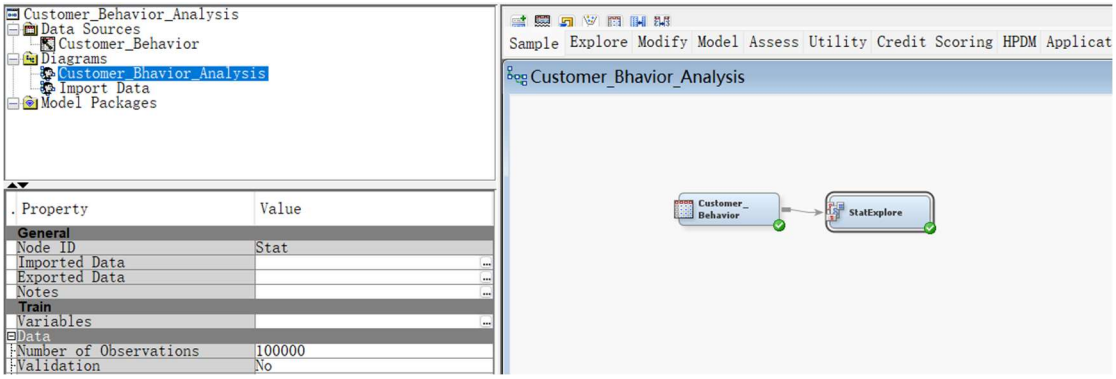
The column metadata auto-classified by SAS is as follows:



After manual reclassification of roles and levels, column metadata is as follows:

Name	Role	Level	Report	Order	Drop	Lower Limit
CustomerID	ID	Interval	No		No	
First_Category	Input	Nominal	No		No	
Gender	Input	Binary	No		No	
Last_Purchase_Date	Input	Interval	No		No	
Location	Input	Nominal	No		No	
Tenure_Months	Input	Interval	No		No	
Total_Purchases	Input	Interval	No		No	
Total_Spent	Rejected	Interval	No		No	
TotalSpent_Categories	Target	Nominal	No		No	

Identify missing value using StatExplore, and there is no missing value at all as shown below, so imputation is not needed.



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	First_Category	INPUT	19	0	Apparel	34.47	Nest-USA	27.11
TRAIN	Gender	INPUT	2	0	F	63.62	M	36.38
TRAIN	Location	INPUT	5	0	California	31.61	Chicago	31.06
TRAIN	TotalSpent_Categories	TARGET	3	0	1	34.54	3	33.45

Distribution of Class Target and Segment Variables
(maximum 500 observations printed)

Data Role=TRAIN

Data Role	Variable Name	Role	Level	Frequency Count	Percent
TRAIN	TotalSpent_Categories	TARGET	1	507	34.5368
TRAIN	TotalSpent_Categories	TARGET	3	491	33.4469
TRAIN	TotalSpent_Categories	TARGET	2	470	32.0163

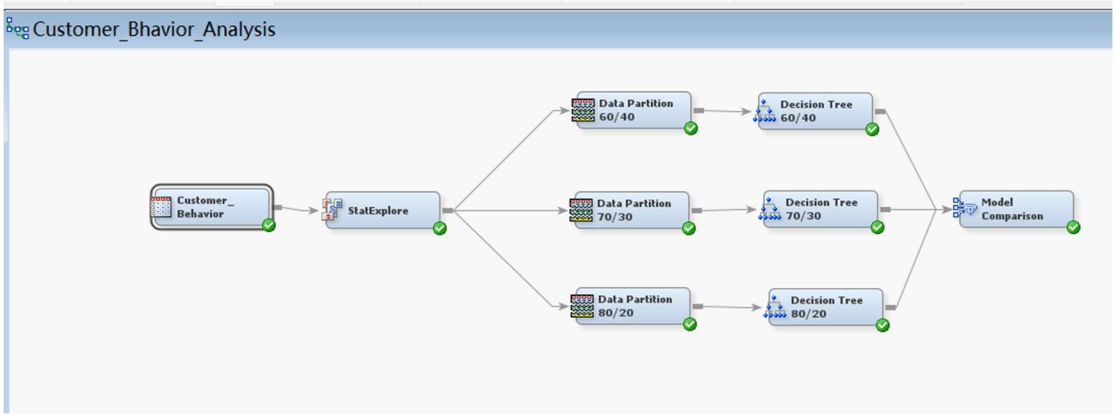
Interval Variable Summary Statistics
(maximum 500 observations printed)

Data Role=TRAIN

Variable	Role	Mean	Standard Deviation	Non Missing	Missing	Minimum	Median	Maximum	Skewness	Kurtosis
Last_Purchase_Date	INPUT	21769.71	101.937	1468	0	21550	21783	21914	-0.45435	-0.87708
Tenure_Months	INPUT	25.91213	13.95967	1468	0	2	26	50	-0.00265	-1.16852
Total_Purchases	INPUT	36.05177	50.88568	1468	0	1	21	695	5.784595	53.62543

5.2 Decision Tree Analysis

Split dataset using Data Partition tool into 60% train set, 40% validation set; 70% train set, 30% validation set; and 80% train set, 20% validation set, separately. Then connect them to decision tree and follow with model comparison, as shown below.



Comparison result is as shown below, the best model is using 70% train set and 30%

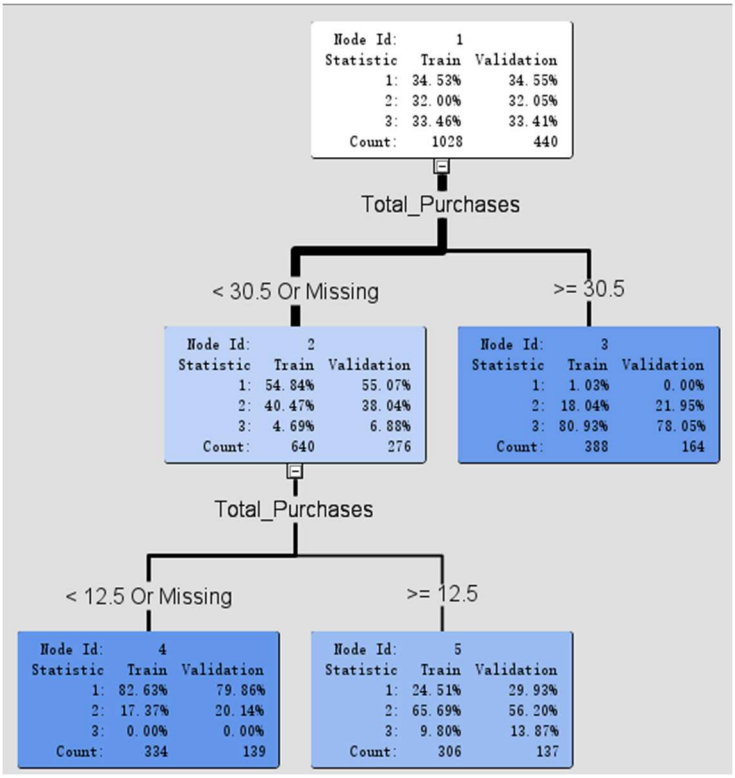
validation set, with the lowest Misclassification Rate of 0.28182 in validation.

Fit Statistics

Model Selection based on Valid: Misclassification Rate (_VMISC_)

Selected Model	Model Node	Model Description	Valid: Misclassification Rate	Train: Average Squared Error	Train: Misclassification Rate	Valid: Average Squared Error
Y	Tree2	Decision Tree 70/30	0.28182	0.11988	0.23054	0.13804
	Tree3	Decision Tree 80/20	0.28669	0.10241	0.24340	0.12910
	Tree	Decision Tree 60/40	0.29302	0.11336	0.21453	0.14455

We select the data splitting ratio of 70:30 for further analysis, and the result is as follows. Based on the dicision tree diagram, the prediction of TotalSpent_Categories is mainly governed by the attributes Total_Purchases after pruning unnecessary branches or attributes that do not provide significant value to the prediction.



Event Classification Table

Data Role=TRAIN Target=TotalSpent_Categories Target Label=TotalSpent_Categories

False Negative	True Negative	False Positive	True Positive
30	610	74	314

Data Role=VALIDATE Target=TotalSpent_Categories Target Label=TotalSpent_Categories

False Negative	True Negative	False Positive	True Positive
19	257	36	128

Based on the confusion matrix, we can calculate the precision, recall, F1-score, accuracy and specificity as follows:

- $Precision = \frac{TP}{TP+FP}$
- $Recall = \frac{TP}{TP+FN}$
- $F1 = \frac{2 \times precision \times recall}{precision + recall}$
- $Accuracy = \frac{TP+TN}{TP+FN+TN+FP}$
- $Specificity = \frac{TN}{TN+FP}$

Metrics	Decision Tree 70:30	
	Train	Validate
Precision	0.809	0.780
Recall	0.913	0.871
F1-Score	0.858	0.823
Accuracy	0.899	0.875
Specificity	0.892	0.877

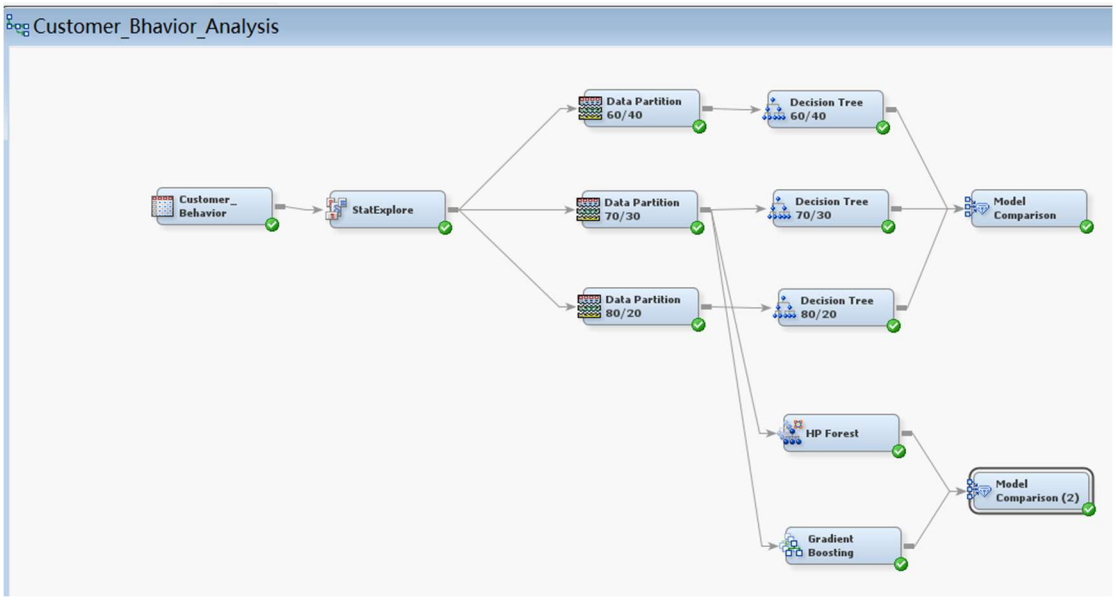
- Discussion:

Precision indicates how many of the samples predicted as positive by the model are true positives. A higher precision indicates that the model is more accurate in predicting positive examples. Recall represents the proportion of actual positive examples successfully captured by the model. A higher recall indicates that the model is better able to identify positive examples. F1-Score combines Precision and Recall and is a measure of the overall performance of the model. It has a trade-off between Precision and Recall. Accuracy represents the overall proportion of correct predictions by the model. A higher accuracy indicates a better overall model performance. Specificity represents the proportion of negative examples that the model successfully predicts. Higher specificity indicates that the model is better able to avoid misclassifying negative examples as positive examples. Taken together, the model performs relatively well on the training and validation sets, with high

accuracy, precision, and recall.

5.3 Ensemble Methods

Using HP Forest as the bagging modelling technique and Gradient Boosting as the boosting modelling technique to predict TotalSpent_Category as follows:



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
HPDMForest	HP Forest	TRAIN	TotalSpent_Categories	TotalSpent_Categories	28	622	62	316
HPDMForest	HP Forest	VALIDATE	TotalSpent_Categories	TotalSpent_Categories	19	257	36	128
Boost	Gradient Boosting	TRAIN	TotalSpent_Categories	TotalSpent_Categories	55	648	36	289
Boost	Gradient Boosting	VALIDATE	TotalSpent_Categories	TotalSpent_Categories	28	274	19	119

Based on the confusion matrix, we can calculate the precision, recall, F1-score, accuracy and specificity as follows:

Metrics	HP Forest		Gradient Boosting	
	Train	Validate	Train	Validate
Precision	0.836	0.780	0.889	0.862
Recall	0.919	0.871	0.840	0.810
F1-Score	0.875	0.823	0.864	0.835
Accuracy	0.912	0.875	0.911	0.893
Specificity	0.909	0.877	0.947	0.935

- Discussion:

HP Forest and Gradient Boosting has similar performance with high precision, recall, F1-Score, Accuracy, and Specificity. Both perform better than Decision Tree.