CIS 8695: Homework 3
Competitive Auctions on eBay.com

Assignment: **Decision Tree**
Professor Name: **Dr. Ling Xue**
Student Name: **Polli Fakhretdinova**
Assignment Due Date: **02/09/2018**

INTRODUCTION

Classification Tree is a machine learning algorithm used for classifying remotely sensed data for analysis. A classification tree is a structural mapping of binary decisions that lead to a decision about the class (interpretation) of an object. Although sometimes referred to as a decision tree, it is more properly a type of decision tree that leads to categorical decisions. A regression tree, another form of decision tree, leads to quantitative decisions.

A Classification tree is built through a process known as binary recursive partitioning. This is an iterative process of splitting the data into partitions, and then splitting it up further on each of the branches. The process starts with a Training Set consisting of pre-classified records (target field or dependent variable with a known class or label such as competitive or non-competitive). The goal is to build a tree that distinguishes among the classes. For simplicity, assume that there are only two target classes, and that each split is a binary partition. The partition (splitting) criterion generalizes to multiple classes, and any multi-way partitioning can be achieved through repeated binary splits. To choose the best splitter at a node, the algorithm considers each input field in turn. Each field is sorted. Every possible split is tried and considered, and the best split is the one that produces the largest decrease in diversity of the classification label within each partition (i.e., the increase in homogeneity). This is repeated for all fields, and the winner is chosen as the best splitter for that node. The process is continued at subsequent nodes until a full tree is generated.

BACKGROUND

The file eBayAuctions.xls contains information on 1972 auctions that transacted on eBay.com during May-June in 2004. The goal is to use these data in order to build a model that will classify competitive auctions from non-competitive ones. A *competitive auction* is defined as an auction with at least 2 bids placed on the auctioned item. The data include variables that describe the auctioned item (auction category), the seller (his/her eBay rating) and the auction terms that the seller selected (auction duration, opening price, currency, day-of-week of auction close). In addition, we have the price that the auction closed at. The goal is to predict whether the auction will be competitive or not.

VARIABLES & DATA

The data consists of categorical output variable **Competitive** which is either **Yes** or **No**, and predictor variables **Category, Currency, SellerRating, Duration, EndDay, ClosePrice, OpenPrice.** The predictors **Category, Currency,** and **EndDay** are categorical variables. Whereas, predictors **SellerRating, Duration, ClosePrice,** and **OpenPrice** are continuous variables.

Output Variable

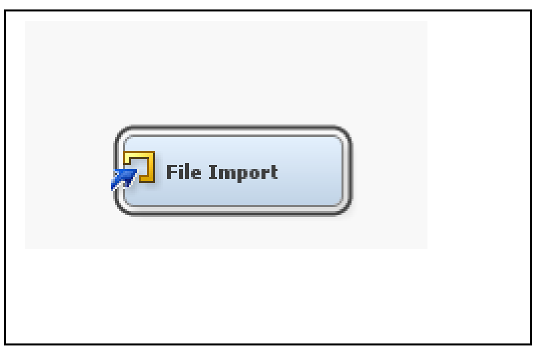
Competitive_Yes
Competitive_No

Predictors Variables

Category
Currency
SellerRating
Duration
EndDay
ClosePrice
OpenPrice

Category	Currency	SellerRating	Duration	EndDay	ClosePrice	OpenPrice	Competitive
EverythingElse	US	29	1	Weekend	300	5	1
Clothing/Toys	US	35	1	Weekend	710	0.01	1
Jewelry	nonUS	72	1	Week	2.45	2.45	0
Jewelry	nonUS	72	1	Weekend	2.45	2.45	0
Art/Collectibles	US	110	1	Weekend	31.01	9.99	1
SportingGoods	US	134	1	Weekend	280	0.99	1
Clothing/Toys	US	251	1	Weekend	145	99.99	1

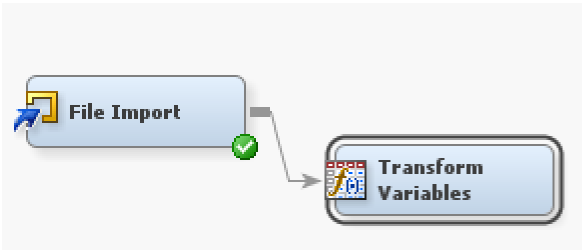
DATA PRE-PROCESSING



Name	Role	Level
Category	Input	Nominal
ClosePrice	Input	Interval
Competitive	Target	Binary
Currency	Input	Nominal
Duration	Input	Interval
EndDay	Input	Nominal
OpenPrice	Input	Interval
SellerRating	Input	Interval

First, we import data into SAS Enterprise Miner.

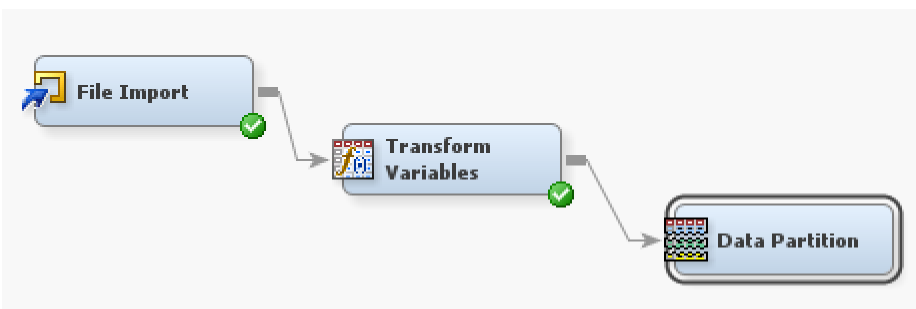
Then, we change the *role* for the dependent variable from **'Input'** to **'Target'** and change *level* for the dependent variable to **'Binary'**.



Name	Method	Number of Bins	Role	Level
Category	**Dummy Indicator**	4	Input	Nominal
ClosePrice	**Default**	4	Input	Interval
Competitive	**Default**	4	Target	Binary
Currency	**Dummy Indicator**	4	Input	Nominal
Duration	**Default**	4	Input	Interval
EndDay	**Dummy Indicator**	4	Input	Nominal
OpenPrice	**Default**	4	Input	Interval
SellerRating	**Default**	4	Input	Interval

Because decision trees cannot handle categorical variables directly, we create dummy variables for the categorical predictors.

These variables include **Category** (11 categories), **Currency** (USD, nonUS), and **EndDay** (Weekend, Week).



Data Set Allocations	
Training	60.0
Validation	40.0
Test	0.0

We then split the data into training and validation datasets using a 60%-40% ratio.

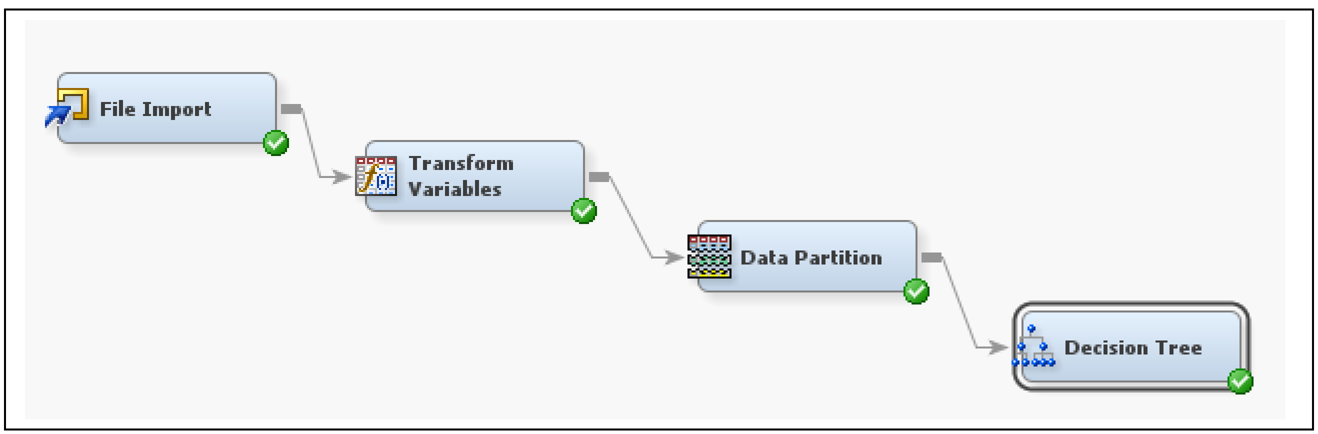
Variable Name	Label
ClosePrice	ClosePrice
Competitive	Competitive
Duration	Duration
OpenPrice	OpenPrice
SellerRating	SellerRating
TI_Category1	Category:Art/Collectibles
TI_Category10	Category:Music/Movie/Game
TI_Category11	Category:SportingGoods
TI_Category2	Category:Books
TI_Category3	Category:Clothing/Toys
TI_Category4	Category:Coins/Stamps
TI_Category5	Category:Computer/Electronics
TI_Category6	Category:EverythingElse
TI_Category7	Category:Health/Beauty
TI_Category8	Category:Home/Garden
TI_Category9	Category:Jewelry
TI_Currency1	Currency:US
TI_Currency2	Currency:nonUS
TI_EndDay1	EndDay:Week
TI_EndDay2	EndDay:Weekend

Name	Use	Report	Role	Level
ClosePrice	**Default**	**No**	Input	Interval
Competitive	**Yes**	**No**	Target	Binary
Duration	**Default**	**No**	Input	Interval
OpenPrice	**Default**	**No**	Input	Interval
SellerRating	**Default**	**No**	Input	Interval
TI_Category1	**Default**	**No**	Input	Binary
TI_Category10	**Default**	**No**	Input	Binary
TI_Category11	**No**	**No**	Input	Binary
TI_Category2	**Default**	**No**	Input	Binary
TI_Category3	**Default**	**No**	Input	Binary
TI_Category4	**Default**	**No**	Input	Binary
TI_Category5	**Default**	**No**	Input	Binary
TI_Category6	**Default**	**No**	Input	Binary
TI_Category7	**Default**	**No**	Input	Binary
TI_Category8	**Default**	**No**	Input	Binary
TI_Category9	**Default**	**No**	Input	Binary
TI_Currency1	**Default**	**No**	Input	Binary
TI_Currency2	**No**	**No**	Input	Binary
TI_EndDay1	**Default**	**No**	Input	Binary
TI_EndDay2	**No**	**No**	Input	Binary

We exclude one dummy variable from each group of dummy variables

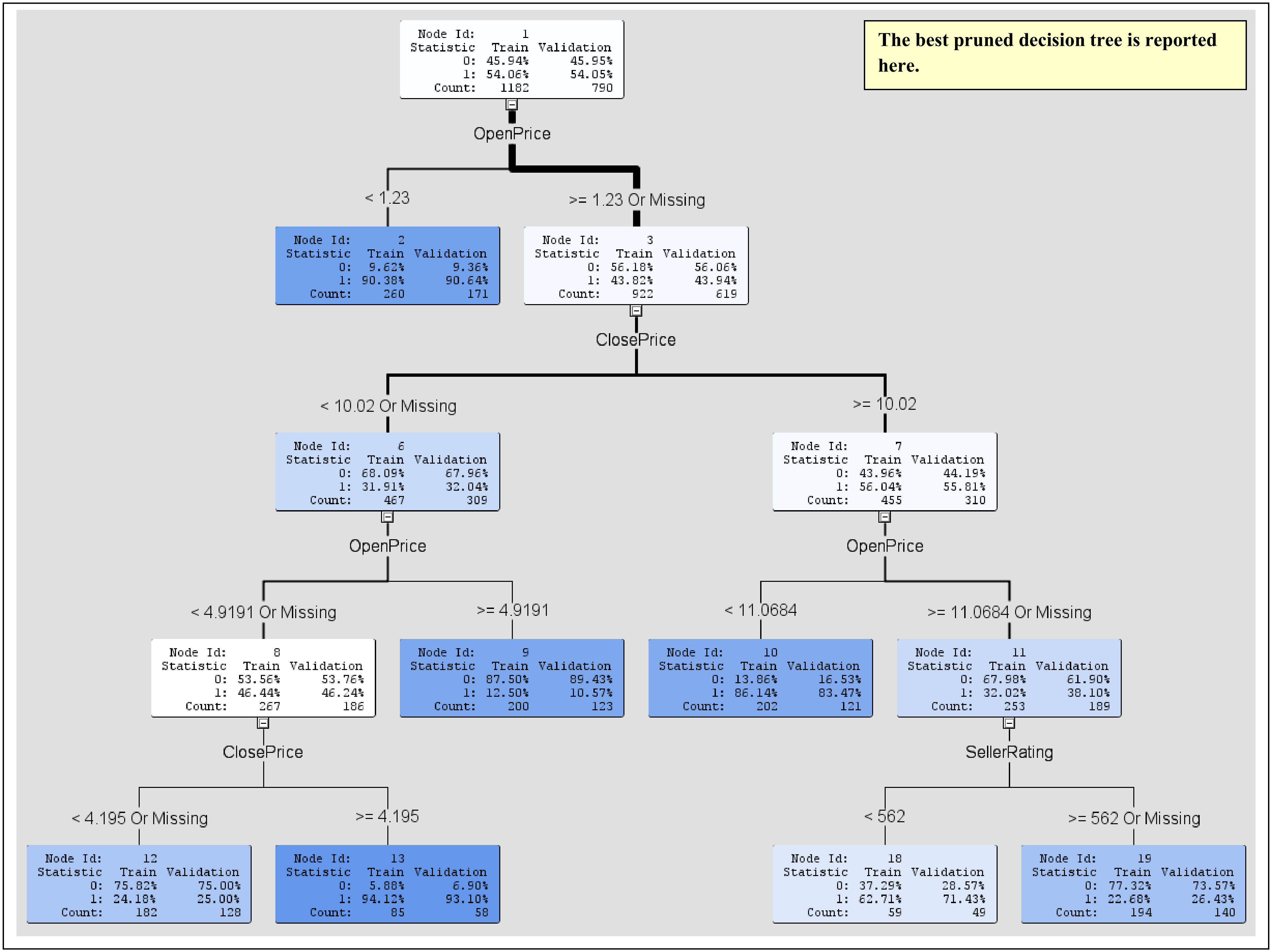
Category_SportingGoods
Currency_nonUS
EndDay_Weekend

DECISION TREE ANALYSIS WITH SAS



Node	
Leaf Size	50
Number of Rules	5
Number of Surrogate Rules	0
Split Size	.

To avoid overfitting, we set the minimum number of records in a leaf node to 50. Then, we run the decision tree analysis.



The best pruned decision tree is reported here.
Node Id: 1
Statistic Train Validation
0: 45.94% 45.95%
1: 54.06% 54.05%
Count: 1182 790
OpenPrice
< 1.23
>= 1.23 Or Missing
Node Id: 2
Statistic Train Validation
0: 9.62% 9.36%
1: 90.38% 90.64%
Count: 260 171
Node Id: 3
Statistic Train Validation
0: 56.18% 56.06%
1: 43.82% 43.94%
Count: 922 619
ClosePrice
< 10.02 Or Missing
>= 10.02
Node Id: 6
Statistic Train Validation
0: 68.09% 67.96%
1: 31.91% 32.04%
Count: 467 309
Node Id: 7
Statistic Train Validation
0: 43.96% 44.19%
1: 56.04% 55.81%
Count: 455 310
OpenPrice
< 4.9191 Or Missing
>= 4.9191
OpenPrice
< 11.0684
>= 11.0684 Or Missing
Node Id: 8
Statistic Train Validation
0: 53.56% 53.76%
1: 46.44% 46.24%
Count: 267 186
Node Id: 9
Statistic Train Validation
0: 87.50% 89.43%
1: 12.50% 10.57%
Count: 200 123
Node Id: 10
Statistic Train Validation
0: 13.86% 16.53%
1: 86.14% 83.47%
Count: 202 121
Node Id: 11
Statistic Train Validation
0: 67.98% 61.90%
1: 32.02% 38.10%
Count: 253 189
ClosePrice
< 4.195 Or Missing
>= 4.195
SellerRating
< 562
>= 562 Or Missing
Node Id: 12
Statistic Train Validation
0: 75.82% 75.00%
1: 24.18% 25.00%
Count: 182 128
Node Id: 13
Statistic Train Validation
0: 5.88% 6.90%
1: 94.12% 93.10%
Count: 85 58
Node Id: 18
Statistic Train Validation
0: 37.29% 28.57%
1: 62.71% 71.43%
Count: 59 49
Node Id: 19
Statistic Train Validation
0: 77.32% 73.57%
1: 22.68% 26.43%
Count: 194 140

Event Classification Table

Data Role=TRAIN Target=Competitive Target Label=Competitive

False Negative	True Negative	False Positive	True Positive
113	463	80	526

Data Role=VALIDATE Target=Competitive Target Label=Competitive

False Negative	True Negative	False Positive	True Positive
82	309	54	345

The **Event Classification Table of Validation Data** for the best pruned tree is reported on the left.

There are **82 False Negatives** and **54 False Positives**,
- 82 competitive auctions were incorrectly classified as non-competitive, 54 non-competitive auctions were classified as competitive auctions.

82 / (82 + 345) = **19.20 %** of the competitive auctions were classified as non-competitive

54 / (54 + 309) = **14.88 %** of the non-competitive auctions were classified as competitive

Predictor Selected by the Decision Tree are:
Open Price
Close Price
Seller Rating

We listed the predictors selected by the decision tree on the left.

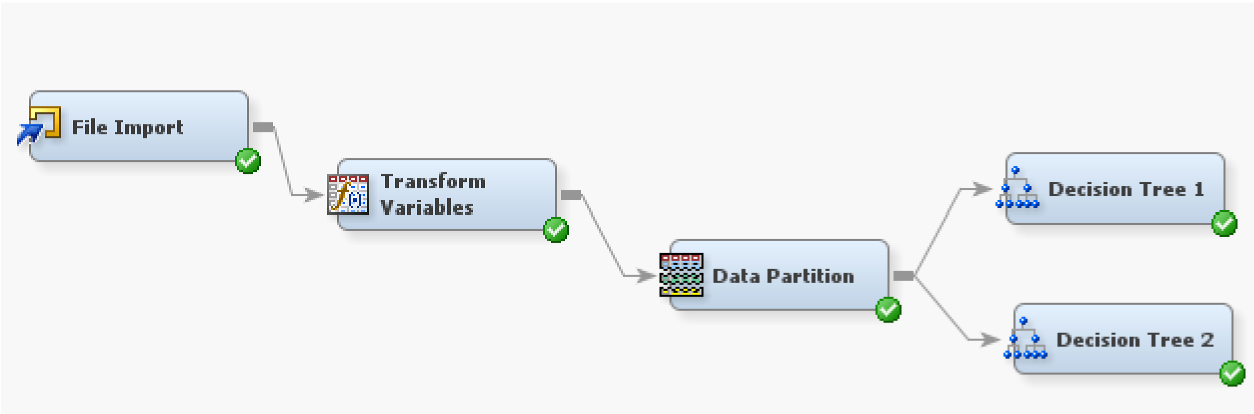
Rules:

If (Open Price < 1.23) then class = 1 (competitive auction).
If (Close Price < 10.02) and (Open Price >= 4.9191) then class = 0 (non-competitive auction).
If (Open Price < 4.9191) and (Close Price < 4.195) then class = 0 (non-competitive auction).
If (Open Price < 4.9191) and (Close Price >= 4.195) then class = 1 (competitive auction).
If (Close Price >= 10.02) and (Open Price < 11.0684) then class = 1 (competitive auction).
If (Open Price >= 11.0684) and (Seller Rating < 562) then class = 1 (competitive auction).
If (Open Price >= 11.0684) and (Seller Rating >= 562) then class = 0 (non-competitive auction).

We described the rules for the classification tree on the left.

Are the rules practical for predicting the outcome of a new auction? Explain why (Hint: are you able to use the rules to classify a new auction before the auction ends? Do you know the values of all predictors in the rules before the auction ends? Some of them may not be known before the end of auction. What are them?). What variables should **NOT** be included in the predictor set? Explain why.

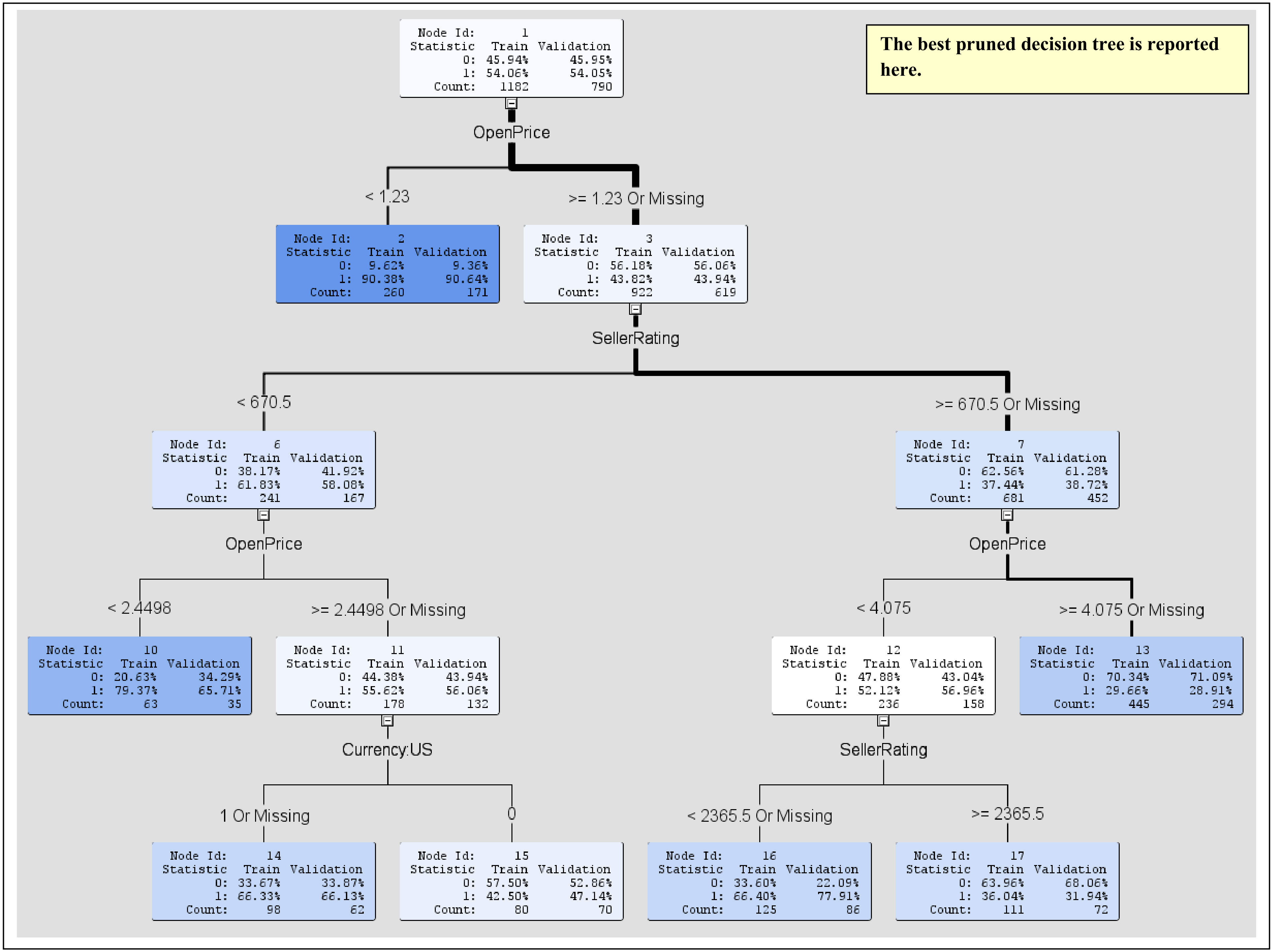
The rules are not practical for predicting the outcome of a new auction because Closing Price for the auction is included in the model. Because closing price is not known before the end of auction, closing price variable should NOT be included in the predictor set. Closing Price should not be included in the model because it indicates the end of auction, meaning no new auction for that particular product.



Name	Use	Report	Role	Level
ClosePrice	No	No	Input	Interval
Competitive	Yes	No	Target	Binary
Duration	Default	No	Input	Interval
OpenPrice	Default	No	Input	Interval
SellerRating	Default	No	Input	Interval
TI_Category1	Default	No	Input	Binary
TI_Category10	Default	No	Input	Binary
TI_Category11	No	No	Input	Binary
TI_Category2	Default	No	Input	Binary
TI_Category3	Default	No	Input	Binary
TI_Category4	Default	No	Input	Binary
TI_Category5	Default	No	Input	Binary
TI_Category6	Default	No	Input	Binary
TI_Category7	Default	No	Input	Binary
TI_Category8	Default	No	Input	Binary
TI_Category9	Default	No	Input	Binary
TI_Currency1	Default	No	Input	Binary
TI_Currency2	No	No	Input	Binary
TI_EndDay1	Default	No	Input	Binary
TI_EndDay2	No	No	Input	Binary
dataobs		No	ID	Interval

We fit another classification tree using the same setting we used in the first classification tree. However, this time we only use the predictors that can be used for predicting the outcome of a new auction. **Meaning, we should NOT include close price into this classification tree.**

As demonstrated in the table on the left, we did NOT use **Close Price** in the classification tree.



The best pruned decision tree is reported here.
Node Id: 1
Statistic Train Validation
0: 45.94% 45.95%
1: 54.06% 54.05%
Count: 1182 790
OpenPrice
< 1.23
>= 1.23 Or Missing
Node Id: 2
Statistic Train Validation
0: 9.62% 9.36%
1: 90.38% 90.64%
Count: 260 171
Node Id: 3
Statistic Train Validation
0: 56.18% 56.06%
1: 43.82% 43.94%
Count: 922 619
SellerRating
< 670.5
>= 670.5 Or Missing
Node Id: 6
Statistic Train Validation
0: 38.17% 41.92%
1: 61.83% 58.08%
Count: 241 167
Node Id: 7
Statistic Train Validation
0: 62.56% 61.28%
1: 37.44% 38.72%
Count: 681 452
OpenPrice
< 2.4498
>= 2.4498 Or Missing
OpenPrice
< 4.075
>= 4.075 Or Missing
Node Id: 10
Statistic Train Validation
0: 20.63% 34.29%
1: 79.37% 65.71%
Count: 63 35
Node Id: 11
Statistic Train Validation
0: 44.38% 43.94%
1: 55.62% 56.06%
Count: 178 132
Node Id: 12
Statistic Train Validation
0: 47.88% 43.04%
1: 52.12% 56.96%
Count: 236 158
Node Id: 13
Statistic Train Validation
0: 70.34% 71.09%
1: 29.66% 28.91%
Count: 445 294
Currency:US
1 Or Missing
0
SellerRating
< 2365.5 Or Missing
>= 2365.5
Node Id: 14
Statistic Train Validation
0: 33.67% 33.87%
1: 66.33% 66.13%
Count: 98 62
Node Id: 15
Statistic Train Validation
0: 57.50% 52.86%
1: 42.50% 47.14%
Count: 80 70
Node Id: 16
Statistic Train Validation
0: 33.60% 22.09%
1: 66.40% 77.91%
Count: 125 86
Node Id: 17
Statistic Train Validation
0: 63.96% 68.06%
1: 36.04% 31.94%
Count: 111 72

```
Event Classification Table

Data Role=TRAIN Target=Competitive Target Label=Competitive

  False       True        False       True
Negative    Negative    Positive    Positive

   206         430         113         433

Data Role=VALIDATE Target=Competitive Target Label=Competitive

  False       True        False       True
Negative    Negative    Positive    Positive

   141         295         68          286
```

The **Event Classification Table of Validation Data** for the best pruned tree is reported on the left.

There are **141 False Negatives** and **68 False Positives**, **-** 141 competitive auctions were incorrectly classified as non-competitive, 68 non-competitive auctions were classified as competitive.

141 / (141 + 286) = **33.02 %** of the competitive auctions were classified as non-competitive

68 / (68 + 295) = **18.73 %** of the non-competitive auctions were classified as competitive

Predictor Selected by the Decision Tree are:
Open Price
Seller Rating
Currency.US

We listed the predictors selected by the decision tree on the left.

Rules:

If (Open Price < 1.23) then class = 1 (competitive auction).
If (Seller Rating < 670.5) and (Open Price < 2.4498) then class = 1 (competitive auction).
If (Open Price >= 2.4498) and (Currency.US =1) then class = 1 (competitive auction).
If (Open Price >= 2.4498) and (Currency.US =0) then class = 0 (non-competitive auction).
If (Seller Rating >= 670.5) and (Open Price >= 4.075) then class = 0 (non-competitive auction).
If (Open Price < 4.075) and (Seller Rating < 2365.5) then class = 1 (competitive auction).
If (Open Price < 4.075) and (Seller Rating >= 2365.5) then class = 0 (non-competitive auction).

We described the rules for the classification tree on the left.

Examine and compare the summary reports for two classification trees. Compare the overall error rates between these two decision trees. Which model has better predictive performance? Explain why.

Target	Target Label	Fit Statistics	Statistics Label	Train	Validation
Competitive	Competitive	NOBS	Sum of Frequencies	1182	790
Competitive	Competitive	_MISC_	Misclassification Rate	0.163283	0.172152
Competitive	Competitive	_MAX_	Maximum Absolute Error	0.941176	0.941176
Competitive	Competitive	_SSE_	Sum of Squared Errors	308.9514	216.9709
Competitive	Competitive	_ASE_	Average Squared Error	0.13069	0.137323
Competitive	Competitive	_RASE_	Root Average Squared Error	0.361511	0.370572
Competitive	Competitive	_DIV_	Divisor for ASE	2364	1580
Competitive	Competitive	_DFT_	Total Degrees of Freedom	1182	.

Target	Target Label	Fit Statistics	Statistics Label	Train	Validation
Competitive	Competitive	NOBS	Sum of Frequencies	1182	790
Competitive	Competitive	_MISC_	Misclassification Rate	0.269882	0.264557
Competitive	Competitive	_MAX_	Maximum Absolute Error	0.903846	0.903846
Competitive	Competitive	_SSE_	Sum of Squared Errors	441.3398	293.3584
Competitive	Competitive	_ASE_	Average Squared Error	0.186692	0.18567
Competitive	Competitive	_RASE_	Root Average Squared Error	0.432079	0.430894
Competitive	Competitive	_DIV_	Divisor for ASE	2364	1580
Competitive	Competitive	_DFT_	Total Degrees of Freedom	1182	.

In order to compare two decision trees, we compared the Misclassification Rate and Root Average Squared Error. The first classification tree has **MISC** of 0.163283 for **Train** data and 0.172152 for the **Validation** data, and **RASE** of 0.361511 for **Train** data and 0.370572 for the **Validation** data. Whereas, the second tree has **MISC** of 0.269882 for the **Train** data and 0.264557 for the **Validation** data, and **RASE** of 0.432079 for **Train** data and 0.430894 for **Validation** data. We discovered that the second classification tree has larger error comparing to the first classification, thus the first model has better predictive performance. The second classification tree has bigger error due to the fact that we did not include the Close Price predictor in the model. It is very likely that the Close Price variable is a good predictor of the dependent variable competitive.