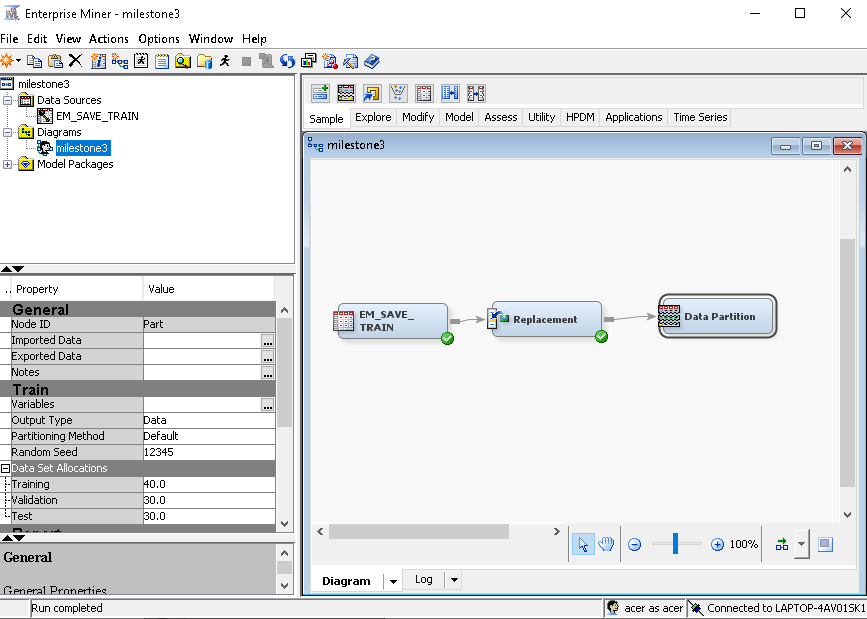
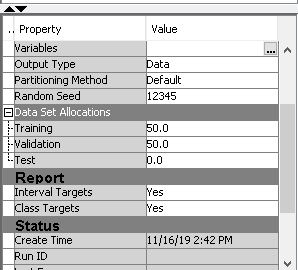


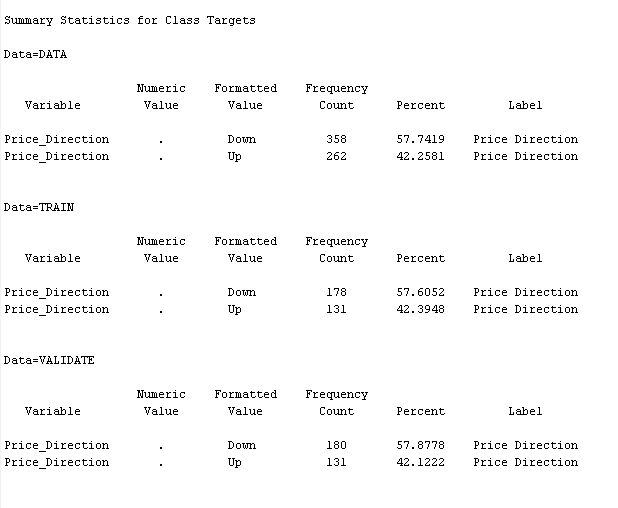
**Data Partition**

Creating models usually rely on training datasets to accurately predict certain outomes. However an over reliance of training data creates problems whereby a model only fits the training set and when new datasets are given to predict the outcome, the model’s performance suffer. In order to overcome this problem, validation datasets are created from the raw datasets, the same raw dataset used for training set. The validation set is created by partitioning the raw data analysis. Training sets are used to build the models while validation sets are used to tune and compare models.

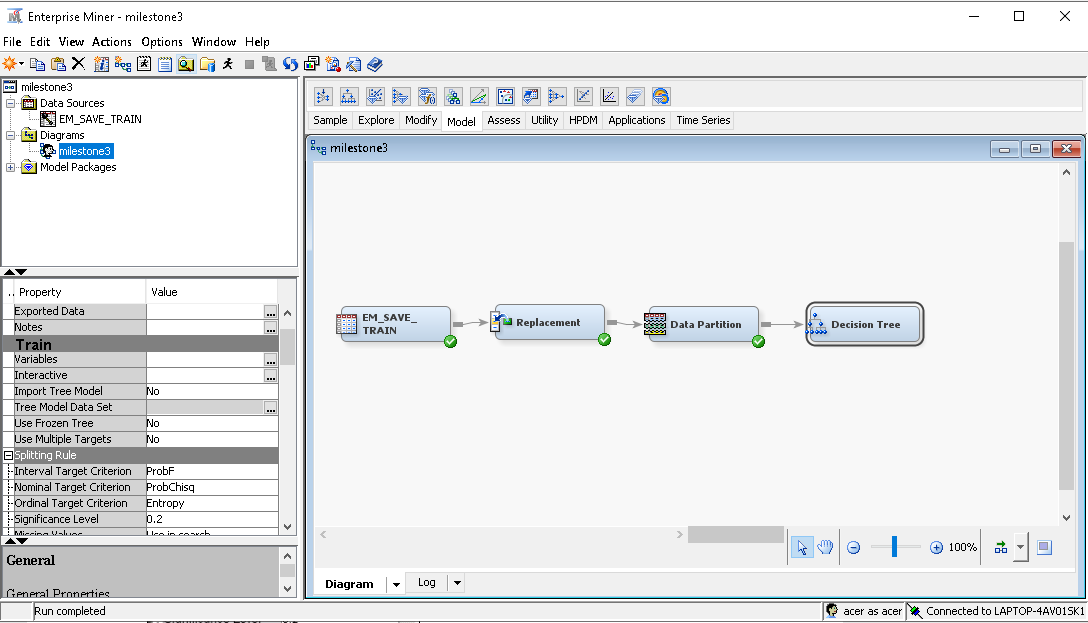


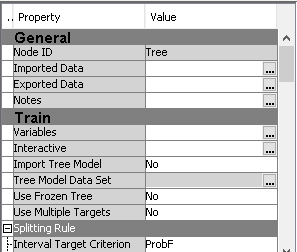
**Set Training and Validation to 50.0 each**



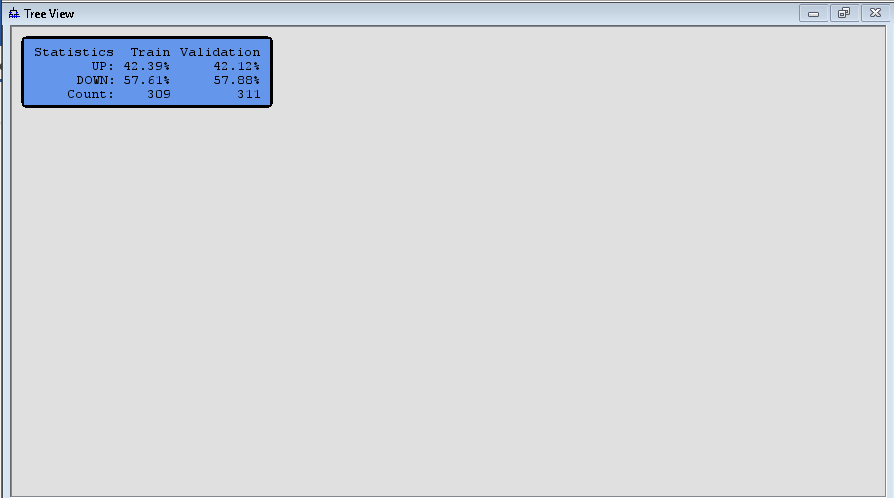


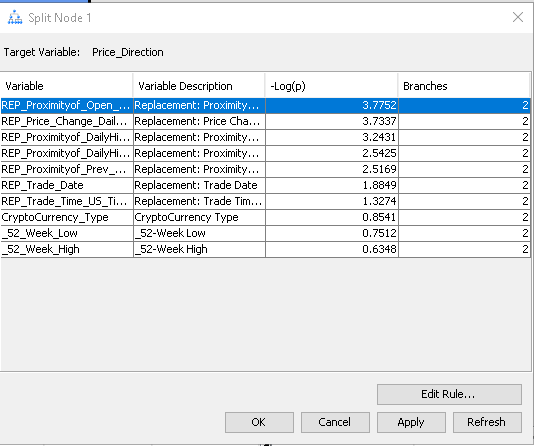
From the result above after running the Data Partition node, the total rows for cryptocurrency price indicating ’Up’ has 358 rows while cryptocurrency prices with ’Down’ labels has 262 rows. After the splitting 50:50 to Training and Validation sets, Training Set has 178 rows of ’Down’ labels and 131 rows of ’Up’ labels. While Validation Set has 180 rows with ’Down’ labels and 131 rows with ’Up’ labels.





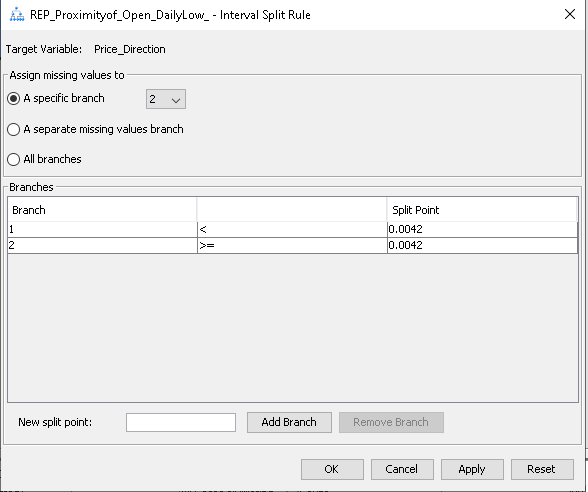
**Splitting the Root Node**



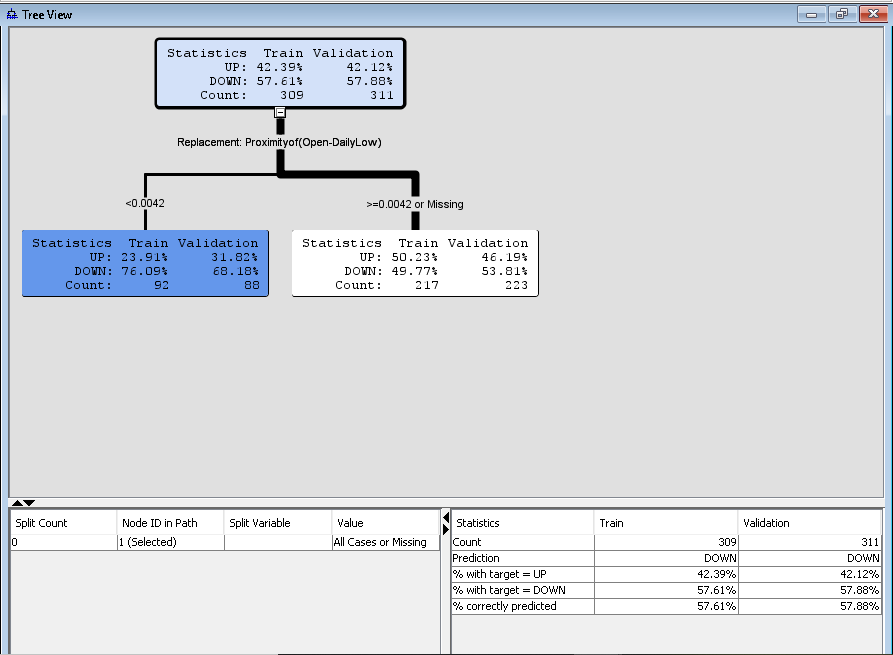


The Split Node dialog box shows the relative value, -Log(p) or logworth of partitioning the training data using the indicated input. As the logworth increases, the partition better isolates cases with identical target values.

Proximityof(Open-DailyLow) is the Opening price minus Daily Low price. Since it has the highest logworth, naturally it is the ideal column used to spilt the tree.

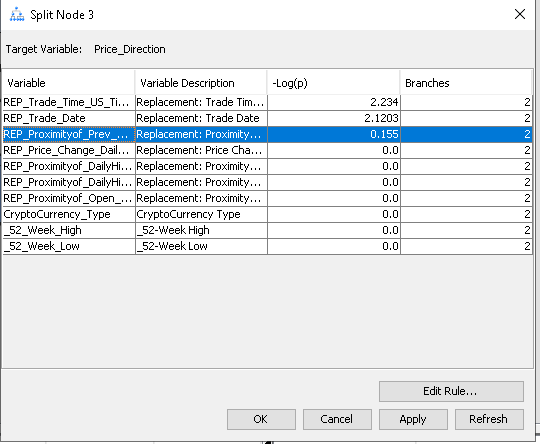


The dialog box shows how the training data is partitioned using the input Proximityof(Open-DailyLow), two branches are created. The first branch contains cases with the Proximityof(Open-DailyLow) less than 0.0042, the second branch has count of equal to or more than 0.0042.

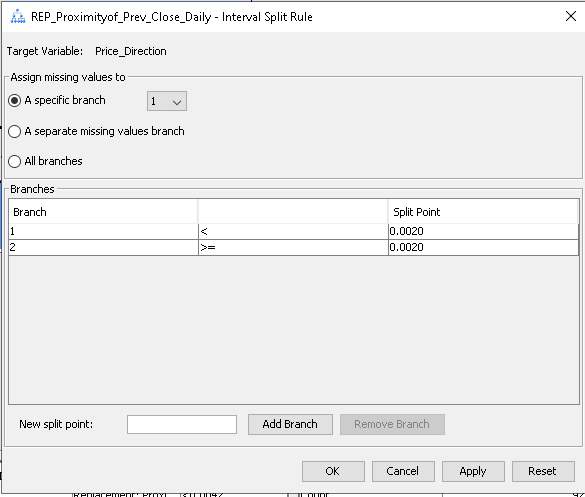
The training data is partitioned into two subsets, the first subset with Proximityof(Open-DailyLow) count less than 0.0042 has a higher than average concentration of Price\_Direction=Down with 76.09% for training data and 68.18% for validation data. On the other hand, Proximityof(Open-DailyLow) with count greater or equal to 0.0042 has more Price\_Direction=Up assigned the training data but validation data has more Price\_Direction=Down.

At the bottom right of the tab also shows the correct instances predicted by the decision tree. With the 309 rows training data, 57.61% of those are accurate at predicting the Price\_Direction as ‘Down’. Mean while with the 311 rows of validation data, 57.88% are predicted correctly as Price\_Direction=Down.

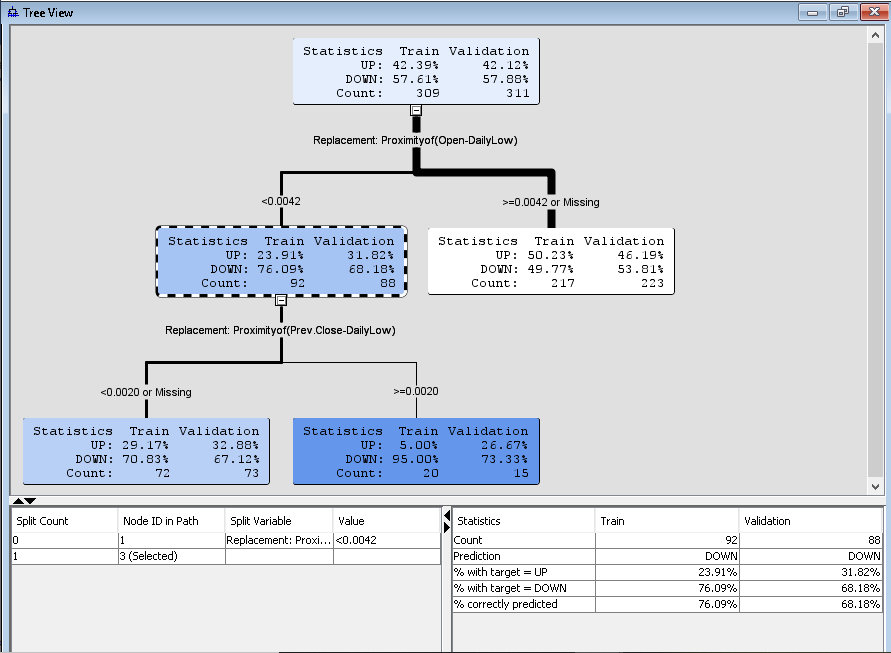
**Split Proximityof(Open-DailyLow) < 0.0042**



The column for Trade\_Time and Trade\_Date columns are ignored even though both these columns has a higher Log value. This is due to the irrelevancy of both these two columns in predicting crypto price movements. The Proximityof\_Prev.Close-DailyLow column was chosen instead.



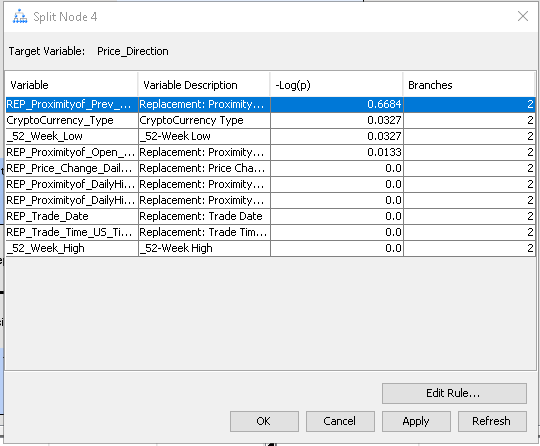
The dialog box shows how the training data is partitioned using the input Proximityof(Prev.Close-DailyLow), two branches are created. The first branch contains cases with the Proximityof(Prev.Close -DailyLow) less than 0.0020, the second branch has count of equal to or more than 0.0020.



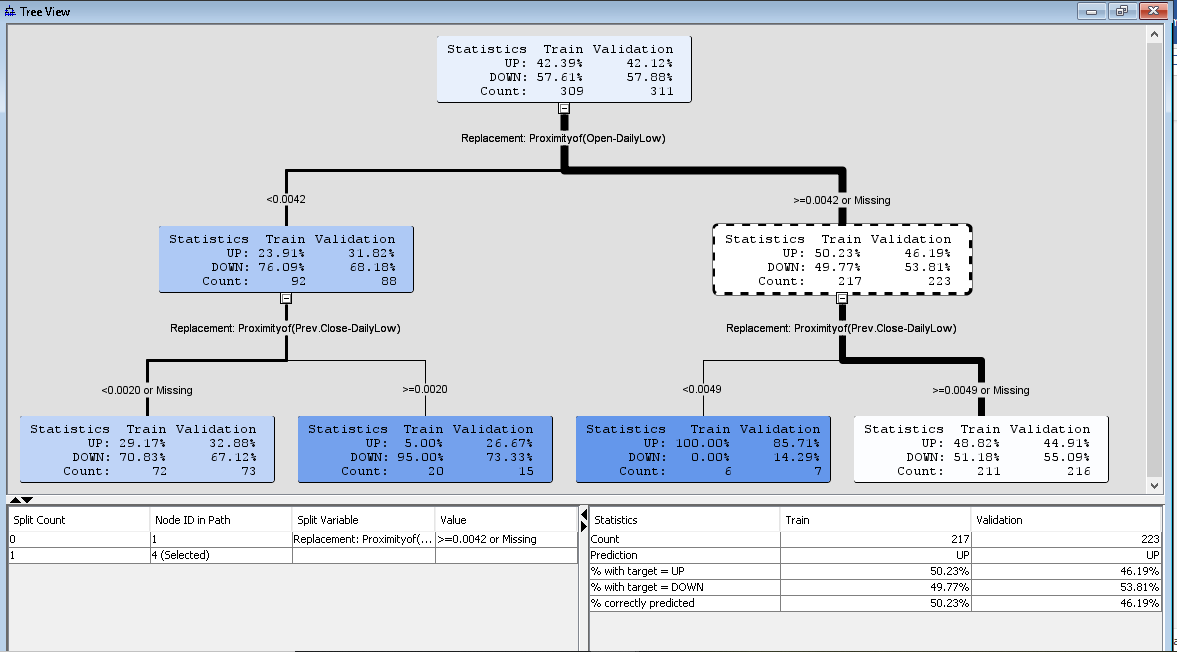
The training data is partitioned into two subsets, the first subset with Proximityof(Prev.Close-DailyLow) count less than 0.0020 has a higher than average concentration of Price\_Direction=Down with 70.83% for training data and 67.12% for validation data. Furthermore, Proximityof(Prev.Close-DailyLow) with count greater or equal to 0.0020 has more Price\_Direction=Down assigned the training data with 95% and validation data has 73.33% assigned.

At the bottom right of the tab, shows the prediction results of the decision tree. This time with the Proximityof(Prev.Close-DailyLow) attribute, the tree was able to predict better with 76.09% of training data instances classified correctly as ‘Down’ and 68.18% of validation data instances correctly classified as ‘Down’.

**Split Proximityof(Open-DailyLow) >=0.0042 or Missing**



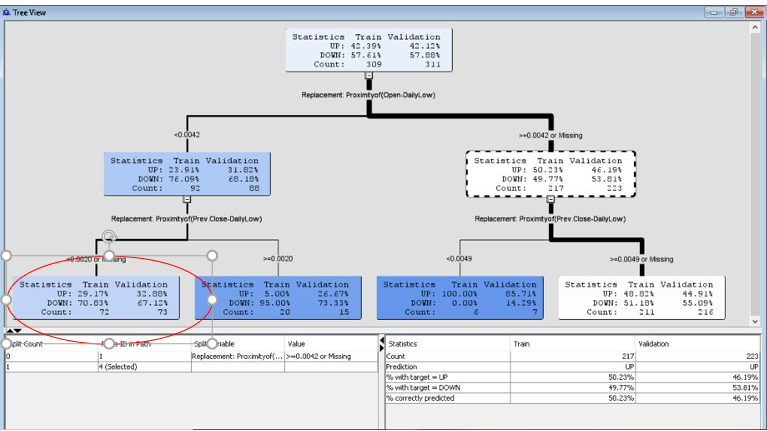
We continue to split the nodes at Proximityof(Open-DailyLow) >=0.0042 or Missing. The dialog box naturally chooses the columns with the highest logworth which in this case is Proximityof(Prev.Close-DailyLow) >=0.0042 or Missing



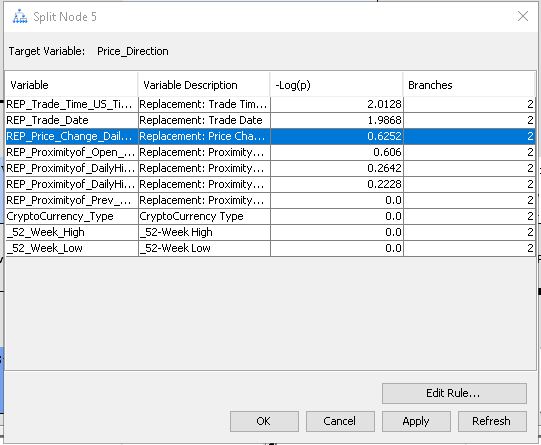
The tree splits with the results of count average of <0.0049 , showing 100% and 85.71% of Price\_Direction=Up for training data set and validation data set respectively. Although this statistical figure looks impressive, but this number of rows taken was only 6 and 7 for both training and validation. This could be due to overfitting as too few instances are taken into account. On the other hand, for Proximityof(Prev.Close-DailyLow) >=0.0049 or Missing, shows that higher average count for Price\_Direction=Down with 51.18% and 55.009% for training and validation.

Below the box can be observed that 49.77% of training data accurately predicted as ‘Down’ whereas 53.81% correctly predicted as ‘Down’ for validation data. It is worth noting here that the accuracy rate has gone down compared previously. It is also worth noting here that for the training set, the tree was able to predict ‘Up’ label higher with 50.23% but suffer for validation data with 46.19%.

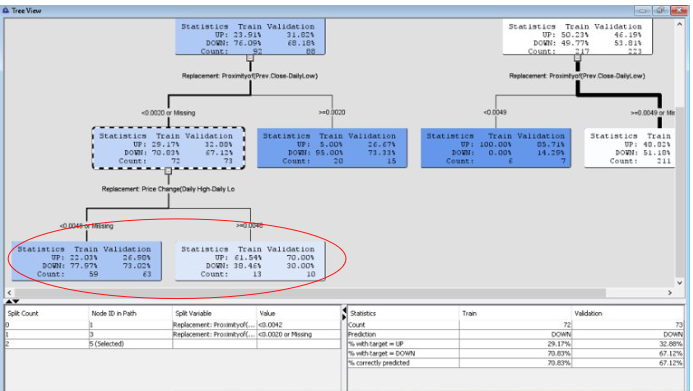
**Split Proximityof(Prev.Close-DailyLow) < 0.0020 or Missing**



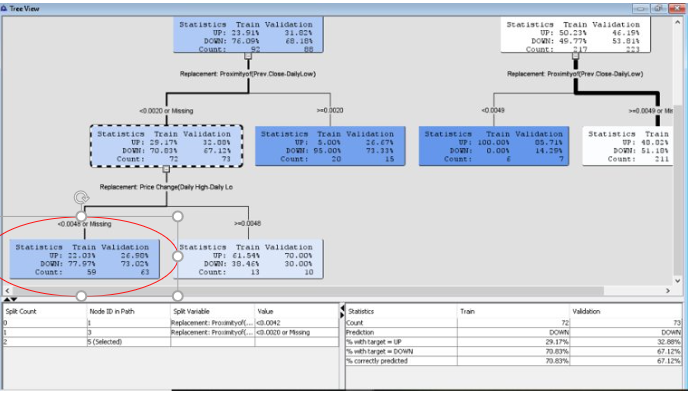
We can continue to split the nodes at Proximityof(Prev.Close-DailyLow) < 0.0020 or Missing as circled in the diagram above.



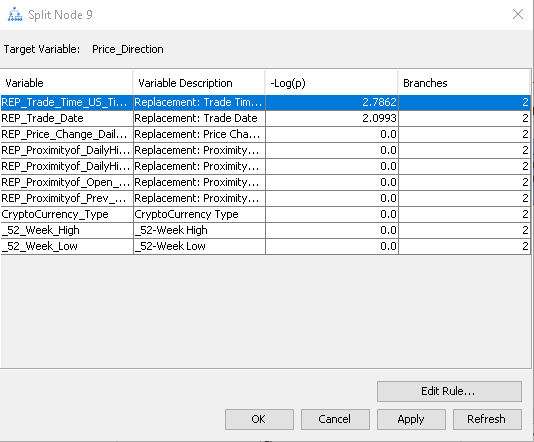
We chose Price Change(DailyHigh-DailyLow) as the Trade Time and Trade Date are irrelevant.

The <0.0048 or Missing node shows that most of the average count are Price\_Direction=Down for both training and validation data with 77.97% and 73.02% respectively. For node >=0.0048, the count average was higher for Price\_Direction=Up, however the number of instances taken was too low with 13 and 10 rows respectively for training and validation sets. This could be the result of overfitting.

**Split Price Change(Open-Prev.Close) < 0.0048 or Missing**

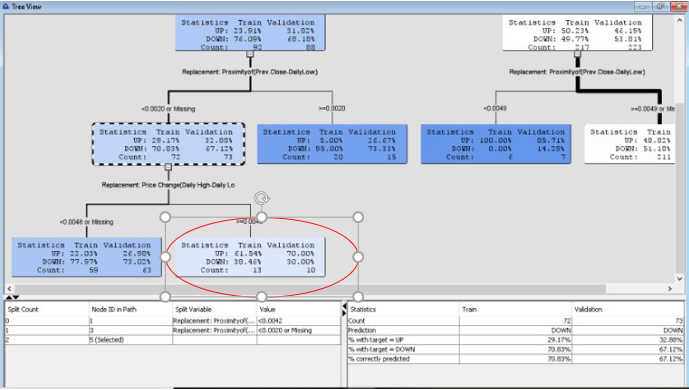


Continue splitting the nodes at Price Change(Open-Prev.Close) < 0.0048 or Missing

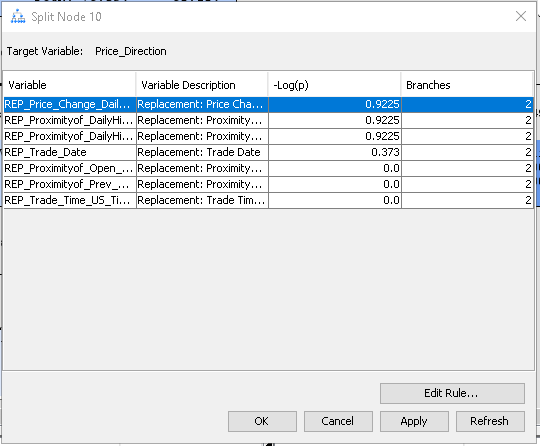


When we open the logworth dialog box, it can be seen that the logworth value for the attributes are zero except for Trade Time and Trade Date. Thus we can no longer split this node.

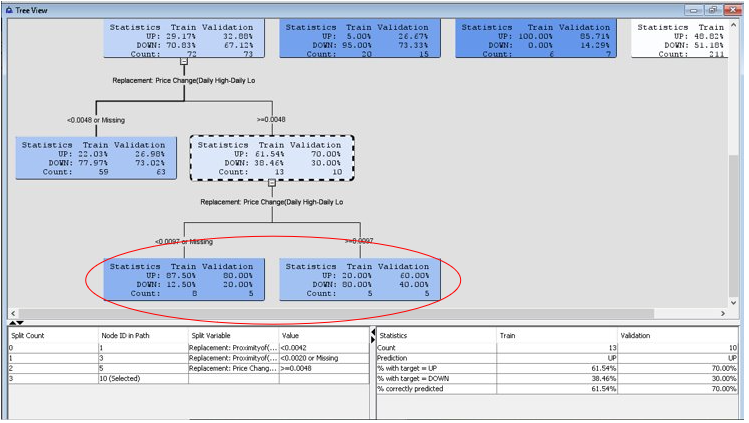
**Split Price Change(Open-Prev.Close) >= 0.0048**



Continue split at Price Change(Open-Prev.Close) >= 0.0048

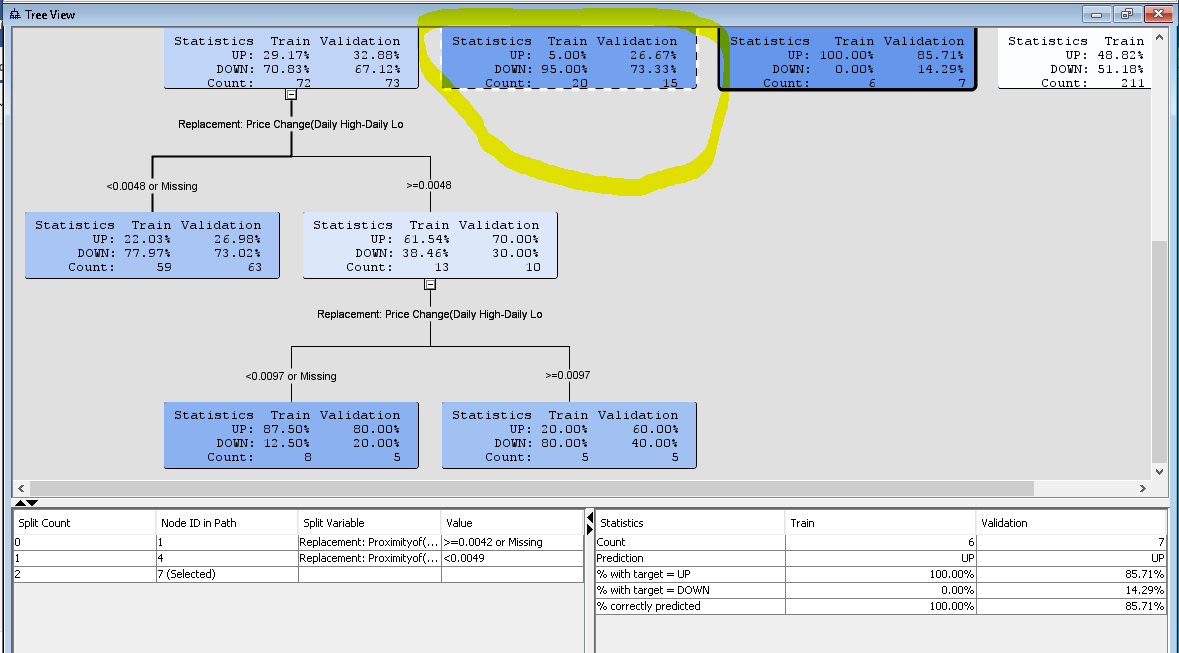


We choose the attributes with the highest logworth which in this case is Proximityof(DailyHigh-DailyLow).

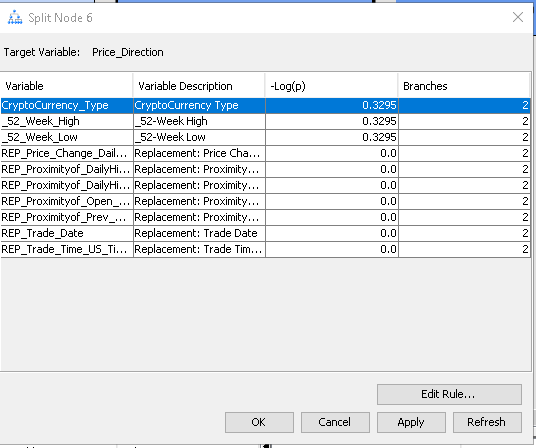


In this split nodes, it can be observed that the tree predicted training data 61.54% accurately for Price\_Direction=Up and validation data 70% accurately with the label ‘Up’. But then again the number of instances are low in order to contribute a significant result.

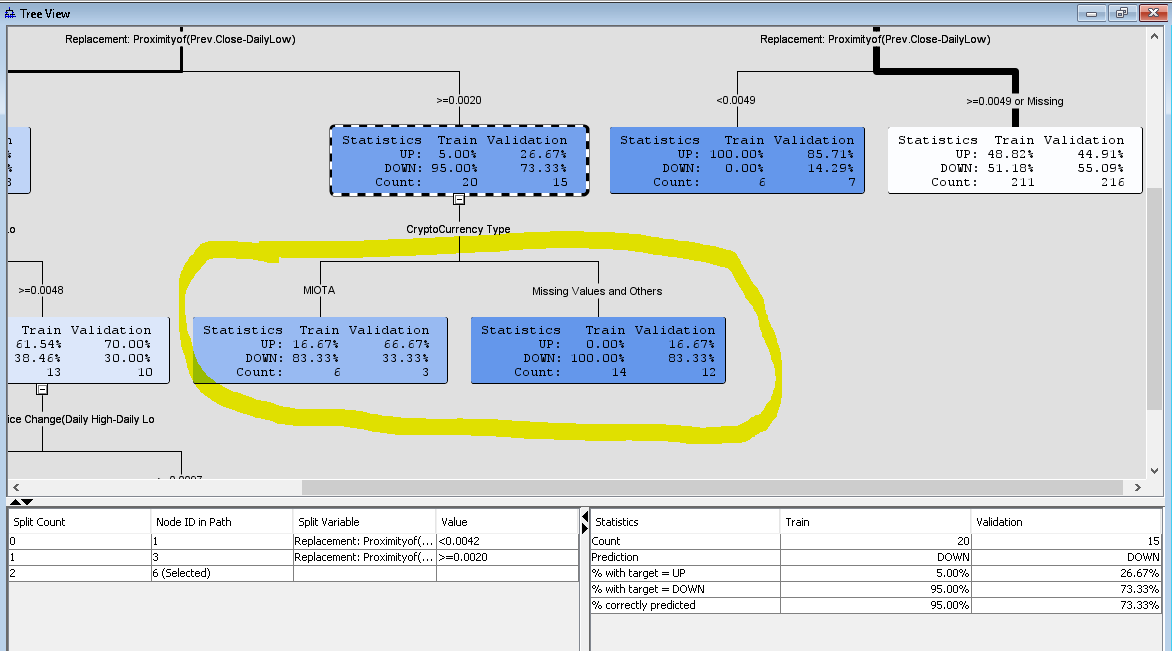
**Split Proximityof(Prev.Close-DailyLow) >=0.0020**



Split Proximityof(Prev.Close-DailyLow) >=0.0020 at the are circled in yellow.



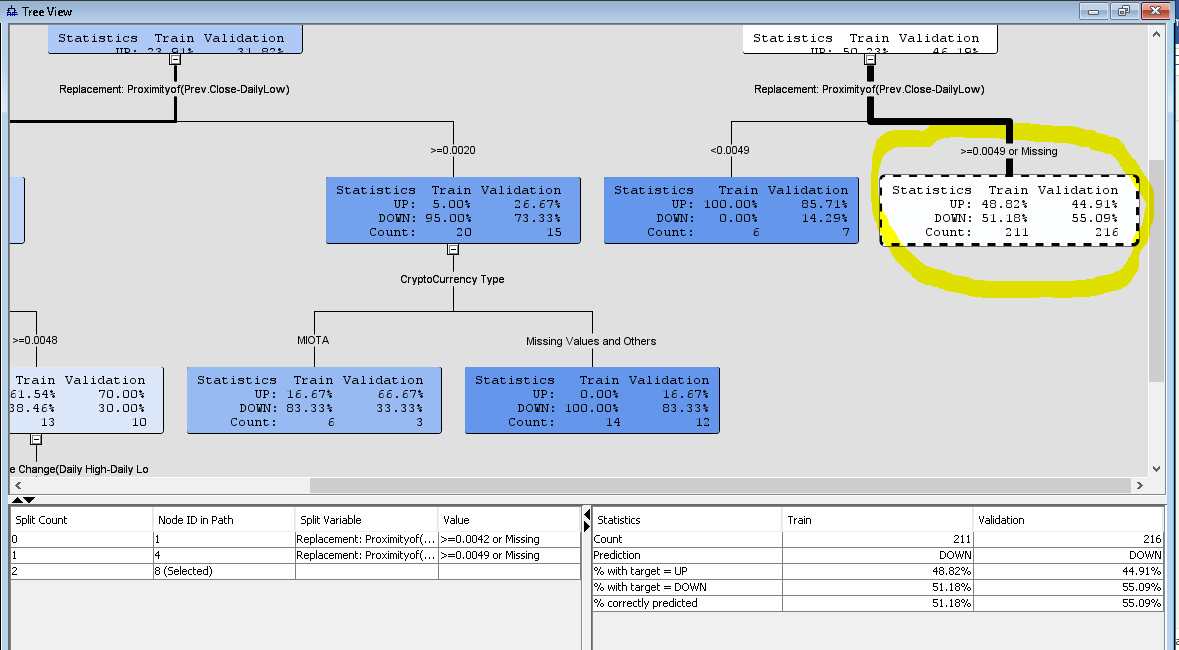
This time the logworth dialog box shows that cryptocurrency type has the highest rank.



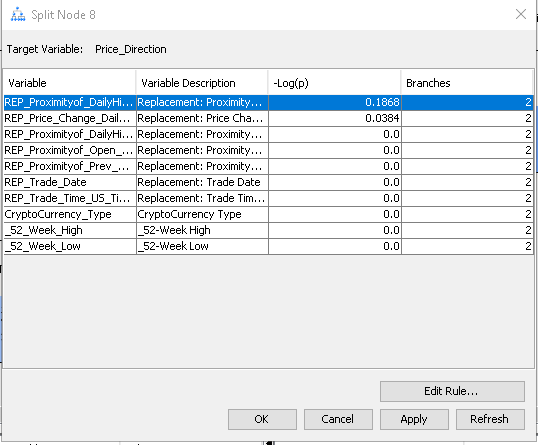
After splitting the nodes, it can be observed that the cryptocurrency price MIOTA has higher average count with 83.33% on the training data but has a significant improved result of 66.67% with the validation data. On the other hand, the Missing Value and Others branch has all the training data instances classified to ‘Down’ with validation data that has 83.33% classified to ‘Down’.

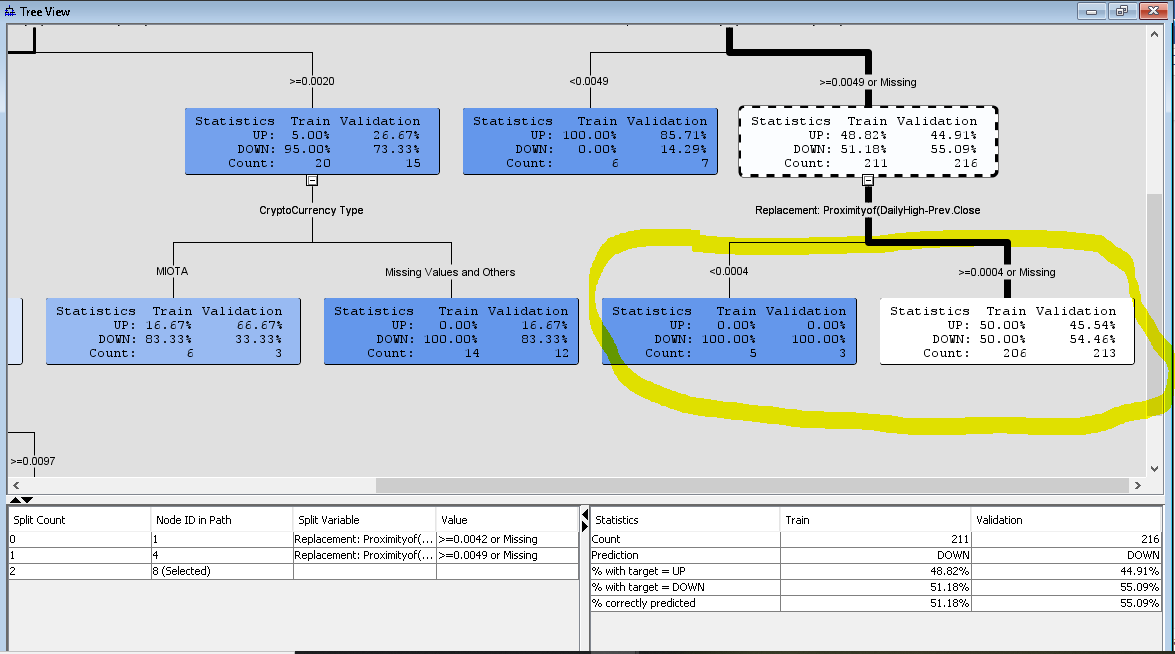
The tree has also a high prediction rate for training data, with 95% accuracy predicting ‘Down’ and 73.33% accuracy predicting ‘Down’ for validation data.

**Split Proximityof(Prev.Close-DailyLow) >= 0.0049 or Missing**



Continue to split node for Proximityof(Prev.Close-DailyLow) >= 0.0049 or Missing

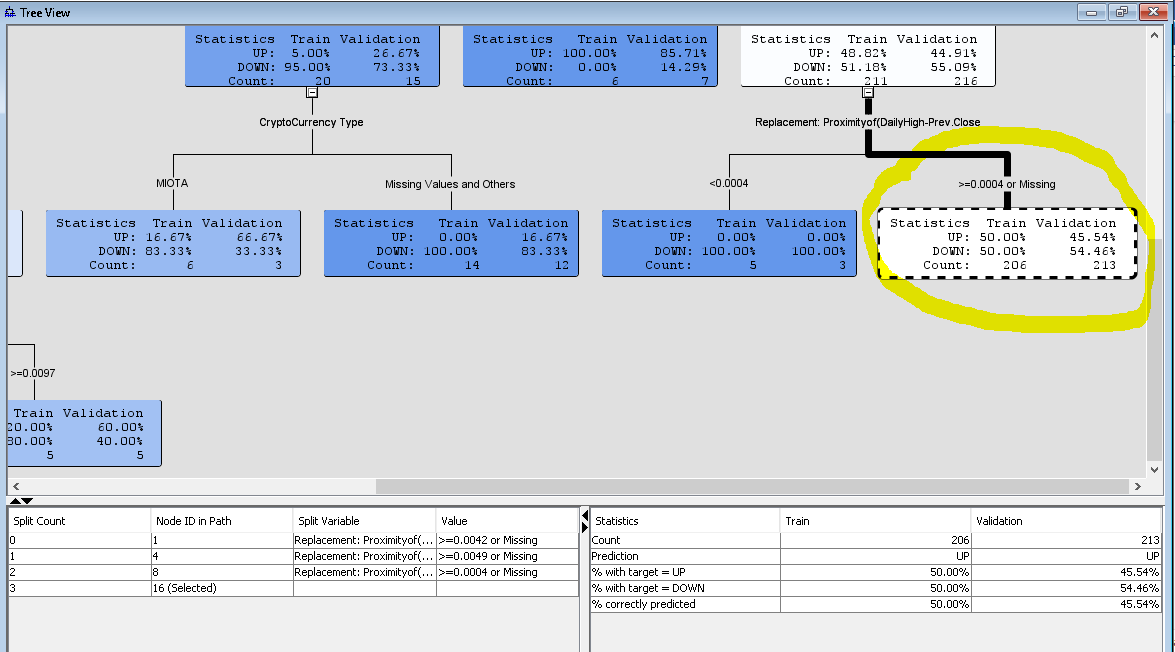




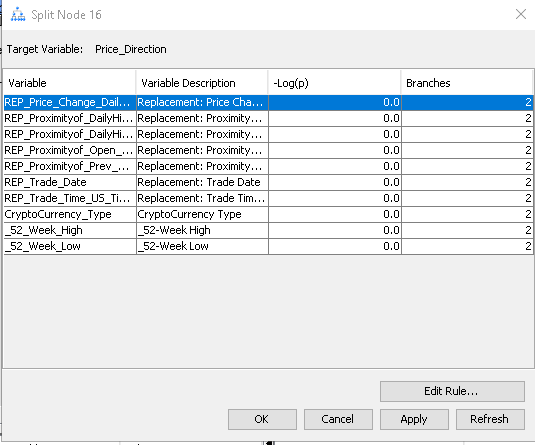
The results show that for Proximityof(Prev.Close-DailyLow) spilt into <0.0004 and >=0.0004 or Missing. The node <0.0004 shows all the instances are assigned to ‘Down’ for both training and validation data. For the node >=0.0004 or Missing , about 50% training data classified to both ‘Up’ and ‘Down’ while about 54.46% validation data classified to ‘Down’.

Further splitting the <0.0004 node Is not possible as all the training and validation data are assigned to ‘Down’.

**Split Proximityof(Prev.Close-DailyLow) >= 0.0004 or Missing**

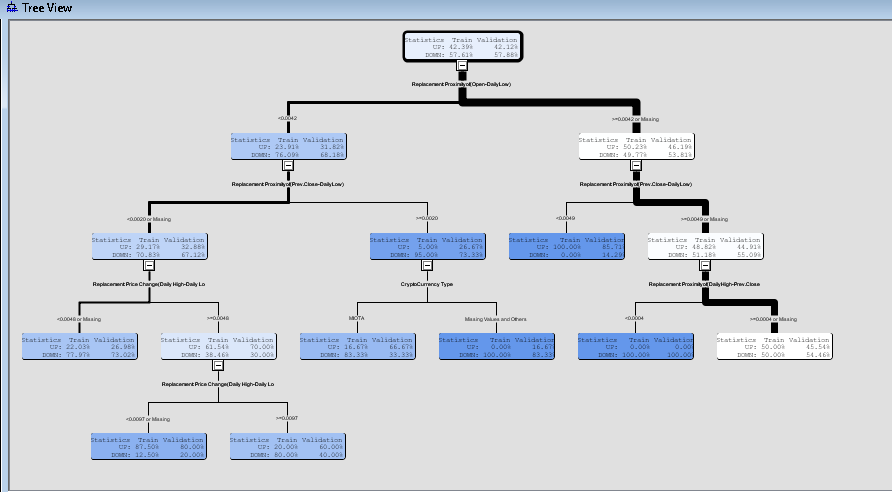


Split the Proximityof(Prev.Close-DailyLow) >= 0.0004 or Missing

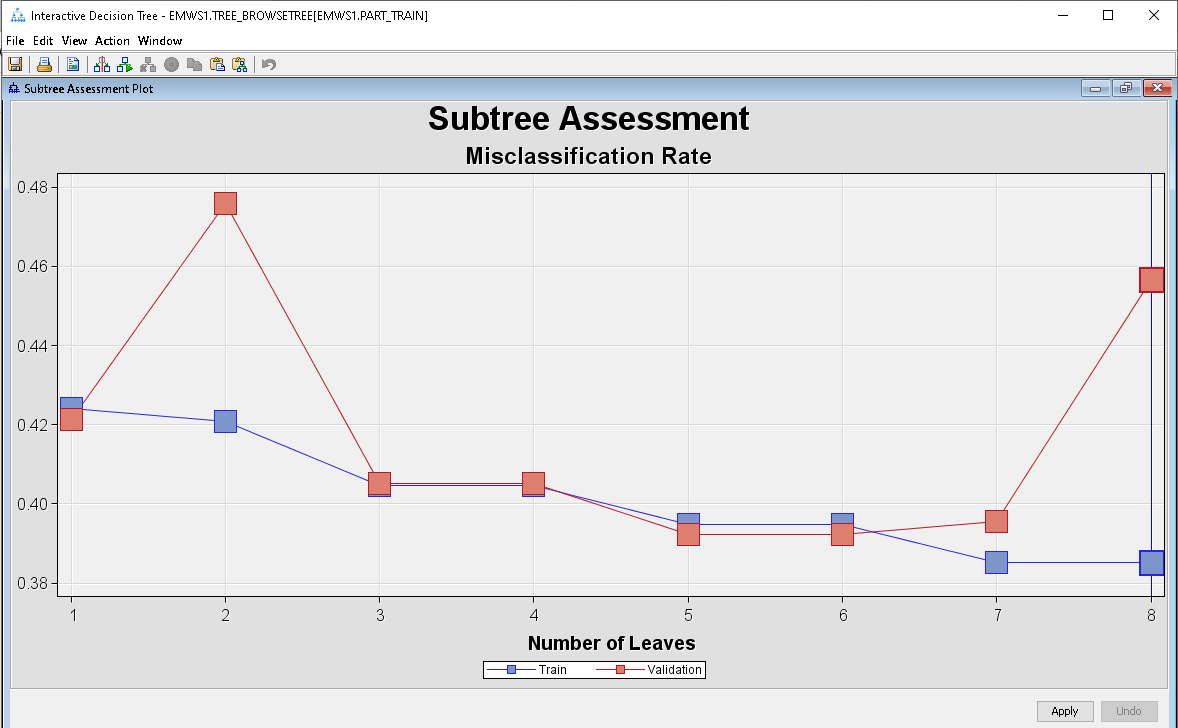


Opening the logworth dialog box, it can be seen that there are no attributes worth for splitting the tree.

**Create the Maximal Tree**



**Subtree Assessment Plot**



From the plot above, it can be observed that the misclassification rate for training and validation data was around 0.42 for number 0f leaves=1. The misclassification rate for training data decrease slightly but increases tremendously for validation data at number of leaves=2. This indicates that at the tree performance suffered for validation data for 2 leaves. Further down, the misclassification rate continue to lower for both training and validation data from number of leaves 3 to 6. Up until number leaves 7, the misclassification rate for validation data showed increase in misclassification, the misclassification rate increase as the number of leaves move to 8 leaves. However, the misclassification rate for training data shows the lowering misclassify rate as the number of leaves increases.