# PROC DTREE VS PROC HPSPLIT

YuTing Tian

## ABSTRACT

Regression and classification trees are methods for analyzing how a dependent variable is correlated with independent variables. PROC HPSPLIT and PROC DTREE can both create decision trees that look similar. Both begin with a single node followed by increased number of leaves. However, they focus on a different purpose. This paper is a preliminary introduction to the differences between PROC HPSPLIT and PROC DTREE.

## INTRODUCTION

When we want to explore the relationship of variables and outcome, that is the effect of variables on the outcome, PROC HPSPLIT is a useful tool. On the other hand, in order to find out the most desired output given the combination of variables, a decision tree with PROC DTREE is the better tool.

This paper is divided into the following four sections:

1) A brief theory of building a tree
2) How to build a tree with PROC HPSPLIT
3) How to build a tree with PROC DTREE
4) What is the difference between PROC HPSPLIT and PROC DTREE

## A BRIEF THEORY OF BUILDING A TREE

Decision trees are a machine learning technique for making predictors; they are built by repeatedly splitting data into smaller and smaller clusters. The tress are trained by passing data down from a root node to leaves. The data is repeatedly split based on predictor variables therefore the sub nodes would be more homogenous.

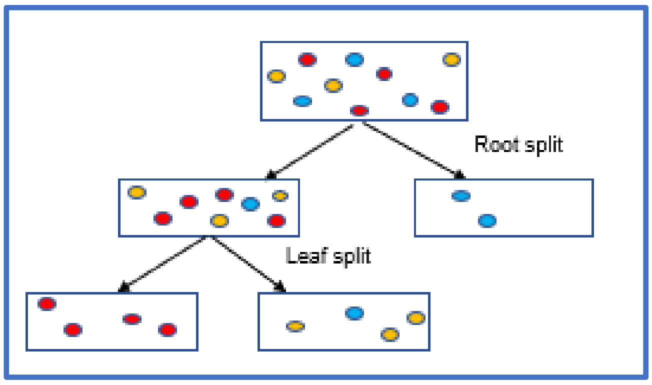


**Figure1**

From the Figure1, the root node begins with the training dataset where the colored dots are classified by the decision tree. The first level split is called the root split, it splits the variables into two groups. In this example we see the right node contains only two blue dots, it means this sub group is pure. The left node continues to split to create another two leaf nodes. All the leaves either contains only one class of dot or is too small to split further. At every node, a set of possible split points is decided by the predictor variable. The algorithm calculates the improvement in purity of the data that would be created by each split points of each variable.

## HOW TO BUILD A TREE WITH PROC HPSPLIT

HPSPLITPROC HPSPLIT is a powerful procedure that quickly and easily creates a decision tree model, each split creates two leaves. The main feature of HPSPLITPROC HPSPLIT is that it can produce tree plots, cost-complexity plots and ROC curves..

In Figure 2 we can see the basic syntax of HPSPLITPROC HPSPLIT; for complete details refer to the documentation PROC HPSPLIT

```
PROC HPSPLIT <options>;
   CLASS variable... </options>;
   CODE FILE=filename;
   GROW criterion </ options>;
   ID variables;
   MODEL response <(response-options)> = variable <variable...>;
   OUTPUT output-options;
   PARTITION <partition-options>;
   PERFORMANCE performance-options;
   PRUNE prune-method <(prune-options)>;
   RULES FILE=filename;
```

**Figure2**

The CLASS statement names the classification variables to be used as explanatory variables in the analysis. If a CLASS statement is specified, it must precede the MODEL statement.

The CODE statement generates a SAS program file that can score new datasets.

The ID statement lists one or more variables from the input dataset that are transferred to output dataset that is created by high performance statistical procedures.

The PARTITION statement specifies how observations in the input dataset are partitioned into training, validation and testing dataset, we can specify the proportions to use for random assignment of observations for each dataset. If we don't set a PARTITION statement, then all observations are assigned to training dataset.

The GROW statement specifies the ***Entropy*** criterion for splitting observations during the process of recursive partitioning that results in a large initial tree.

***Entropy*** is a measure of purity. The mathematical formula of Entropy is:

$$E(S) = -\sum P(i)\log_2 P(i)$$

where $P(i)$ is the frequentist probability of a class i in the data. In order to describe the theory of ***Entropy*** clearly, I create two classes in SAS (the code is in the ***Appendix***), with thirty observations in positive class and seventy observations in negative class as shown the **Figure3.** Therefore, based on the ***Entropy*** algorithum, the positive probability would be 3/10, and the negative probability would be 7/10;

$$E(S) = -\sum P(i)\log_2 P(i)$$

$$= -\frac{3}{10} * \log 2\left(\frac{3}{10}\right) - \frac{7}{10} * \log 2\left(\frac{7}{10}\right)$$

$$\approx 0.879$$

In this case, we see the entropy is 0.879, a very low level of purity. The entropy is between 0 and 1, the larger value of entropy, the lower level of purity (the higher level of disorder); Lower level of purity leads further splitting.

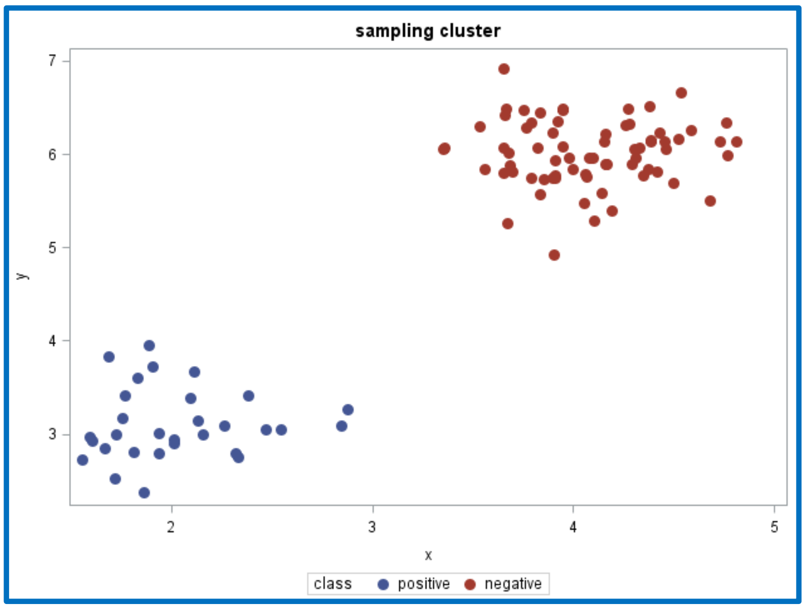


**Figure3**

In order to demonstrate ***Entropy*** further, Figure 4 as shown below, we see the x-axis measures the proportion of data belonging to positive class, the y-axis measures the value of entropy; we see the graph is an inverted U shape. Entropy is the lowest at the bottom, when the bubbles contain either only negative or positive symbols. This shows the highest level of purity as entropy approaches zero.

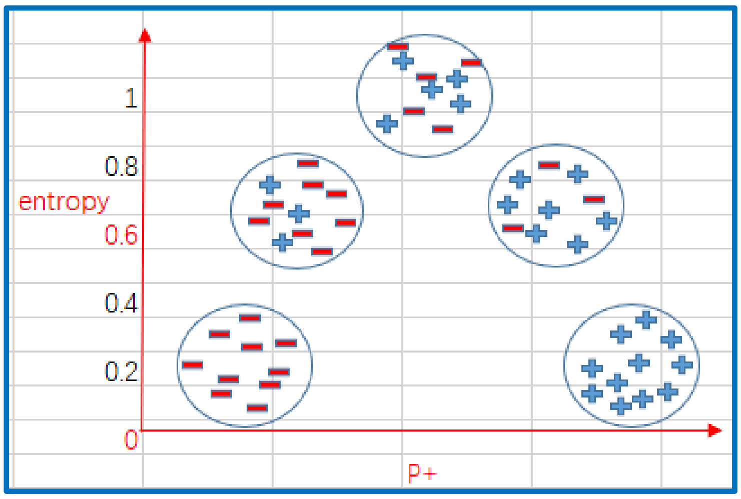


**Figure4**

For the HPSPLIT procedure, which selects the best splitting variable and the best cutoff value to produce the highest reduction in impurity. PROC HPSPLIT can use other methods to split, but this paper we only focus on Entropy method.   In order to avoid the tree overfitting we use PRUNE statement; the default PRUNE method is cost complexity

Cost complexity pruning is a widely used pruning method for either categorical or continuous dependent variable. The syntax is:

**prune costcomplexity;**

The algorithm is based on making a trade off between complexity of a tree and the error rate to help prevent overfitting. The function is:

CC(T)=R(T)+a |f(T)|

Where R(T) represents error rate, f(T) is a function that returns a set of leaves of Tree T , and a is regularization parameter that used to address overfitting. The decision tree is restricted sum of squares R(T).

PROC HPSPLIT creates a plot of the cross validated ASE(Average Square Error) at each complexity parameter value in the sequence. The subtree in the pruning sequence that has the lowest validation error rate is selected as the final tree. Often, the 1-SE rule defined by Breiman et al. (1984) is applied when the pruning via the cost-complexity method to potentially select a smaller tree that has only a slightly higher error rate than the minimum ASE.

I will perform the regression tree example using Fisher's Iris data. For the Iris dataset, which gives the measurement of type, ***petal width, petal length, sepal width*** and ***sepal length,*** for a sample of 150 observations. Type 1 is Setasa, type 2 is Versicolor, type 3 is Virginica.

We see the decision tree graph shown on the below when I run **PROC HPSPLIT**

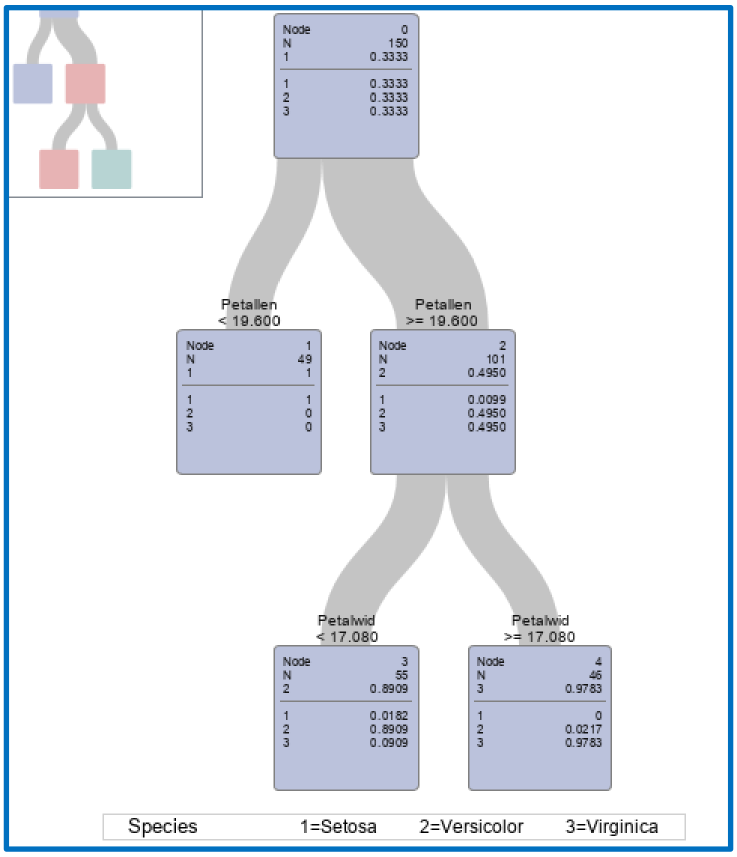
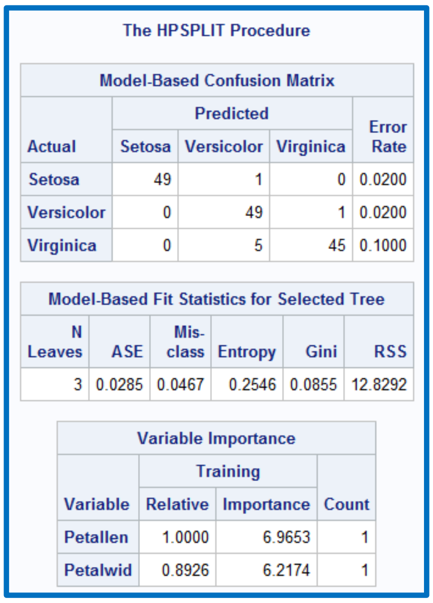


**Figure5**

From the **Figure5**, there are 3 leaves in the tree. we see the variable petal length is the most important variable in splitting observations into 3 different flowers: Setosa, Versicolor and Virginica.

On the first splitting level, If the petal length is less than 19.6, a subject would go to the left, 49 observations are classified as Setosa. If the Petal length is greater than or equal 19.6, then the subject would go to right. However, it is not pure, because almost half observations distribute into the Versicolor, another remaining half observation distribute into the Virginica.

On the second splitting level, it splits according to the variable Petal width, if the Petal length is less than 17.08; then almost 90% observations goes to Versicolor in the left sub leaf; when the Petal width is greater than or equal 17.08, 97% observations are grouped into Virginica.



The HPSPLIT Procedure

Model-Based Confusion Matrix

| Actual | Predicted |  |  | Error Rate |
|---|---|---|---|---|
|  | Setosa | Versicolor | Virginica |  |
| Setosa | 49 | 1 | 0 | 0.0200 |
| Versicolor | 0 | 49 | 1 | 0.0200 |
| Virginica | 0 | 5 | 45 | 0.1000 |

Model-Based Fit Statistics for Selected Tree

| N Leaves | ASE | Mis-class | Entropy | Gini | RSS |
|---|---|---|---|---|---|
| 3 | 0.0285 | 0.0467 | 0.2546 | 0.0855 | 12.8292 |

Variable Importance

| Variable | Training |  | Count |
|---|---|---|---|
|  | Relative | Importance |  |
| Petallen | 1.0000 | 6.9653 | 1 |
| Petalwid | 0.8926 | 6.2174 | 1 |

**Figure6**

In the **Figure6**, it represents the HPSPLIT procedure, we see there are two important variables which help to predict the type of flowers in the training dataset: ***Petal length*** and ***Petal width***. The final tree has 3 leaves, the entropy value is 0.2546, very pure. And average squared error is 0.0286.

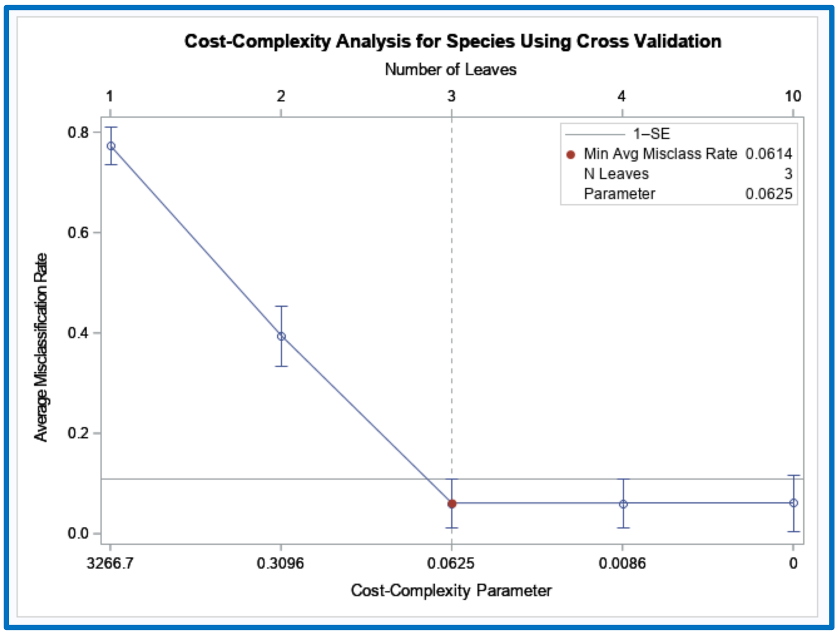


**Figure7**

In this case, in **Figure7,** we see the minimum average misclassification rate is 0.0625 when the tree with 3 leaves. I do not select the subtree with 2 leaves even though 1-SE rule, because the smaller tree that has a higher error rate than the lowest ASE, almost 0.4.

## HOW TO BUILD A TREE WITH PROC DTREE

In decision trees created by PROC DTREE, the variables are chosen because the decision managers have already known or they can infer the process of building a tree based on their actual working experience. Therefore, the decision makers know and can predict the way the variables are distributed among groups then the result of each pathway can be estimated in the final tree.

The goal of decision tree built by PROC DTREE is to explore the most reasonable and desirable outcome given the combination of variables and costs; this helps managers to make decisions on how to arrange the finite cost to pursue for the maximum return. The complete process will be performed through the pathway of decision tree.

PROC DTREE constructs a decision model call *a generic decision tree model*, that is made in stages. There are three types of stages: decision stages, chance stages and end stages.

- A decision stage represents a decision the manager has to make. The outcomes of a decision stage are the possible actions in this process.
- A chance stage represents a random factor in the decision problem, the outcomes of each chance stage is an event, one of which occurs based on a given probability distribution.
- An end stage ends up a sequence of events, it is not necessary to include in the model.

In Figure 8 we can see the basic syntax of PROC DTREE; For complete details refer to the documentation PROC DTREE. As we go through creating the required datasets, you will see that understanding layout of DTREE data can be a challenge.

```
PROC DTREE options ;
        STAGEIN=options;
        PROBIN=options
        PAYOFFS=options;
        EVALUATE / options;

        SUMMARY / options;
        TREEPLOT / options;
        VARIABLES / options;
```

**Figure8**

In this case, we have two stages: decision stage and chance stage.

- Decision stage

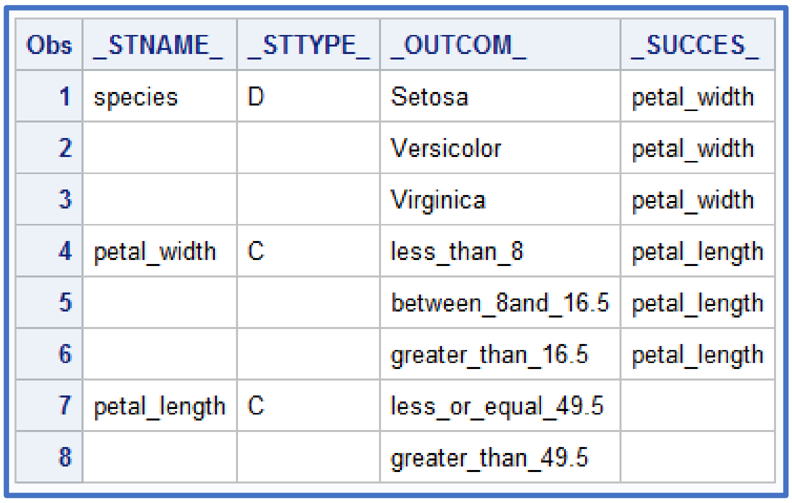
  The scenario we will use for the DTREE procedure is about a decision problem for a flower grower, who must decide what type of flower he has based on the petal length and petal width.

- Change stage
  Based on his previous experience, for the petal width, the width would be less than 8 inches with probability 0.2, between 8 and 16.5 inches with probability of 0.6, and greater than16.5 inches with probability of 0.2; For the petal length, the length would be less than or equal to 49.5 with probability of 0.5, greater than 49.5 with probability of 0.5.

  The cost of the cultivating of flower is based on the petal size, the petal width and petal length. An experienced flower grower can evaluate the cost to grow an Iris, It will cost $10 with the petal width less than 8 inches, $20 with the petal width between 8 and 16.5 inched, $30 when the petal width greater than 16.5 inches. For the petal length, it will cost $40 with the petal length less than or equal to 49.5 inches, $70 with the petal length greater than 49.5 inches. For these three Iris flowers: Setosa, Versicolor and Virginica there are different sale prices. The sale price of Setosa is normally 3 times of the costs of growing, the sale price of Versicolor is 1.5 times its cultivating costs, the sale price of Virginica is three times as its cultivating costs.

First, ***PROC DTREE*** requires three input datasets: raw data, probability data, and payoffs data. The raw data contains the stage name, the type, and the attributes of all outcomes for each stage in the model. The structure of decision model is given in the **STAGEIN=dataset** has a following columns**:**

- _STTYPE_**:** D for decision, C for chance
- _STNAME_: labels. For the decision stage it tells us we are looking at the species. For each of the chance we see we are looking at petal width and petal length.
- _SUCCES_: the immediate successors (another stage) of each outcome specified by the _OUTCOME_=variables, it means the values of _SUCCES_ must be stage names (_STNAME_). In this case, the value of _SUCCES_ are petal width and petal length.



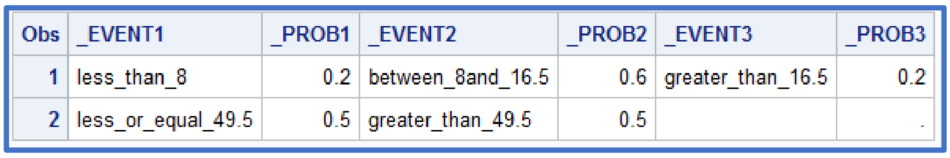
| Obs | _STNAME_ | _STTYPE_ | _OUTCOM_ | _SUCCES_ |
|---|---|---|---|---|
| 1 | species | D | Setosa | petal_width |
| 2 | | | Versicolor | petal_width |
| 3 | | | Virginica | petal_width |
| 4 | petal_width | C | less_than_8 | petal_length |
| 5 | | | between_8and_16.5 | petal_length |
| 6 | | | greater_than_16.5 | petal_length |
| 7 | petal_length | C | less_or_equal_49.5 | |
| 8 | | | greater_than_49.5 | |

**Figure9**

In **Figure9**, the structure of decision problem STTYPE=D, the decision stage has three outcomes ***"Setosa", "Versicolor"*** and ***"Virginica"*** and the first chance stage petal width has three outcomes: similarly, the second chance stage petal length has two outcomes.

.

**PROBIN=dataset**, it describes the probabilities for every chance event in the model.
For example, petal width contains 3 chance events, and petal length contains 2 chance events.

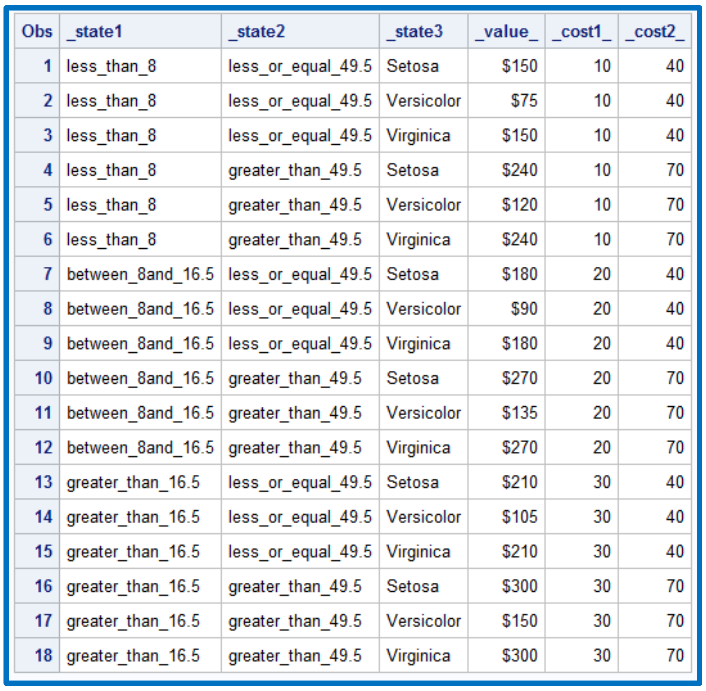


| Obs | _EVENT1 | _PROB1 | _EVENT2 | _PROB2 | _EVENT3 | _PROB3 |
|---|---|---|---|---|---|---|
| 1 | less_than_8 | 0.2 | between_8and_16.5 | 0.6 | greater_than_16.5 | 0.2 |
| 2 | less_or_equal_49.5 | 0.5 | greater_than_49.5 | 0.5 | | . |

**Figure10**

In **Figure10**, 3 events from petal width are matching 3 outcomes from **Figure9**. Similarly, 2 events from petal length are matching 2 outcomes. Event 1 in **Figure 10** relates probability as we stated above. with **Figure9** the chance outcome of probability, which involve the petal width with three chance events, the petal length with two chance events. The sum of probability for each chance type equals to 1.

The third dataset called **PAYOFFS=dataset**, contains all values of each possible sequence.

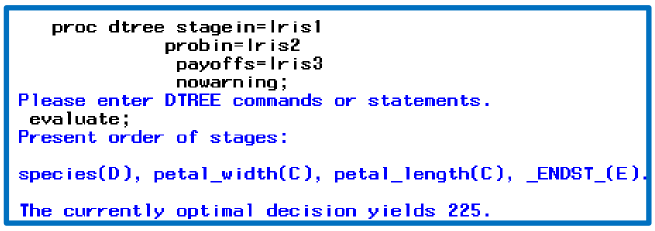


| Obs | _state1 | _state2 | _state3 | _value_ | _cost1_ | _cost2_ |
|---|---|---|---|---|---|---|
| 1 | less_than_8 | less_or_equal_49.5 | Setosa | $150 | 10 | 40 |
| 2 | less_than_8 | less_or_equal_49.5 | Versicolor | $75 | 10 | 40 |
| 3 | less_than_8 | less_or_equal_49.5 | Virginica | $150 | 10 | 40 |
| 4 | less_than_8 | greater_than_49.5 | Setosa | $240 | 10 | 70 |
| 5 | less_than_8 | greater_than_49.5 | Versicolor | $120 | 10 | 70 |
| 6 | less_than_8 | greater_than_49.5 | Virginica | $240 | 10 | 70 |
| 7 | between_8and_16.5 | less_or_equal_49.5 | Setosa | $180 | 20 | 40 |
| 8 | between_8and_16.5 | less_or_equal_49.5 | Versicolor | $90 | 20 | 40 |
| 9 | between_8and_16.5 | less_or_equal_49.5 | Virginica | $180 | 20 | 40 |
| 10 | between_8and_16.5 | greater_than_49.5 | Setosa | $270 | 20 | 70 |
| 11 | between_8and_16.5 | greater_than_49.5 | Versicolor | $135 | 20 | 70 |
| 12 | between_8and_16.5 | greater_than_49.5 | Virginica | $270 | 20 | 70 |
| 13 | greater_than_16.5 | less_or_equal_49.5 | Setosa | $210 | 30 | 40 |
| 14 | greater_than_16.5 | less_or_equal_49.5 | Versicolor | $105 | 30 | 40 |
| 15 | greater_than_16.5 | less_or_equal_49.5 | Virginica | $210 | 30 | 40 |
| 16 | greater_than_16.5 | greater_than_49.5 | Setosa | $300 | 30 | 70 |
| 17 | greater_than_16.5 | greater_than_49.5 | Versicolor | $150 | 30 | 70 |
| 18 | greater_than_16.5 | greater_than_49.5 | Virginica | $300 | 30 | 70 |

**Figure11**

From **Figure11**, for the third observation, representing Virginica, the $150 payoff (sale price) associated with the sequence petal width less than 8 ,petal length less than or equal to 49.5 , the _COST1_ related to the width is $10, the _COST2_ related to the length is $40; sale price we know is 3 times as cost of growing, that would be $150.

The **EVALUATE** statement evaluates the decision tree, shown in the log.

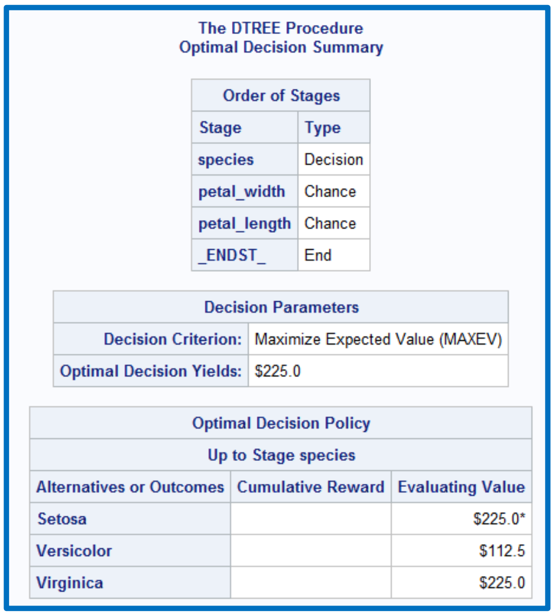


```
    proc dtree stagein=Iris1
              probin=Iris2
               payoffs=Iris3
               nowarning;
Please enter DTREE commands or statements.
  evaluate;
Present order of stages:

species(D), petal_width(C), petal_length(C), _ENDST_(E).

The currently optimal decision yields 225.
```

**Figure 12**

We see the optimal decision is $225 from the **Figure12.**

The **SUMMARY** statement can produce the optimal decisions.



The DTREE Procedure
Optimal Decision Summary

| Order of Stages | |
|---|---|
| **Stage** | **Type** |
| species | Decision |
| petal_width | Chance |
| petal_length | Chance |
| _ENDST_ | End |

| Decision Parameters | |
|---|---|
| Decision Criterion: | Maximize Expected Value (MAXEV) |
| Optimal Decision Yields: | $225.0 |

| Optimal Decision Policy | | |
|---|---|---|
| Up to Stage species | | |
| **Alternatives or Outcomes** | **Cumulative Reward** | **Evaluating Value** |
| Setosa | | $225.0* |
| Versicolor | | $112.5 |
| Virginica | | $225.0 |

**Figure 13**

The **figure13** summarize the best action, to maximize the expected payoff for cost cultivating Iris is $225.

We also can put the **SUMMARY** option in the **EVALUATE** statement to produce the optimal decisions.

**TREEPLOT** statement can plot the complete tree as shown in Figure 14.

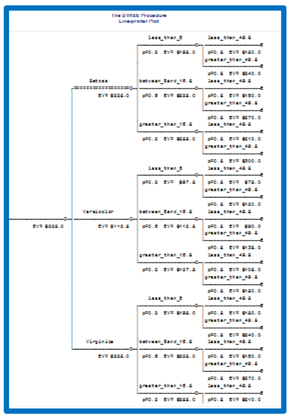
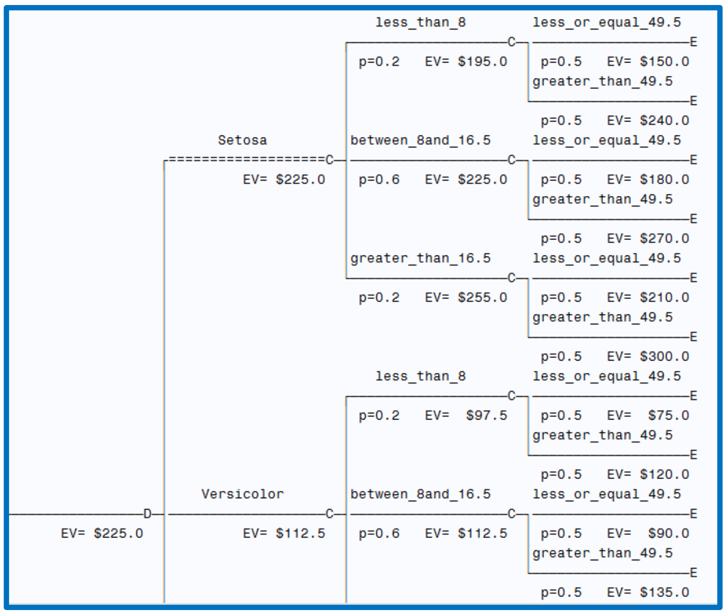


**Figure 14**

For clarity, a subset of the tree is shown in Figure 15.



```
                                            less_than_8          less_or_equal_49.5
                                         ----------------------C-------------------------E
                                         | p=0.2    EV= $195.0 | p=0.5    EV= $150.0
                                         |                     |greater_than_49.5
                                         |                     |-------------------------E
                                         |                       p=0.5    EV= $240.0
                          Setosa         |between_8and_16.5     less_or_equal_49.5
                    |===================C----------------------C-------------------------E
                    |         EV= $225.0 | p=0.6    EV= $225.0 | p=0.5    EV= $180.0
                    |                    |                     |greater_than_49.5
                    |                    |                     |-------------------------E
                    |                    |                       p=0.5    EV= $270.0
                    |                    |greater_than_16.5     less_or_equal_49.5
                    |                    |---------------------C-------------------------E
                    |                      p=0.2    EV= $255.0 | p=0.5    EV= $210.0
                    |                                          |greater_than_49.5
                    |                                          |-------------------------E
                    |                                            p=0.5    EV= $300.0
                    |                       less_than_8         less_or_equal_49.5
                    |                    ----------------------C-------------------------E
                    |                    | p=0.2    EV=  $97.5 | p=0.5    EV=  $75.0
                    |                    |                     |greater_than_49.5
                    |                    |                     |-------------------------E
                    |                    |                       p=0.5    EV= $120.0
                    |    Versicolor      |between_8and_16.5     less_or_equal_49.5
-------------------D--------------------C----------------------C-------------------------E
     EV= $225.0     |         EV= $112.5 | p=0.6    EV= $112.5 | p=0.5    EV=  $90.0
                    |                    |                     |greater_than_49.5
                    |                    |                     |-------------------------E
                    |                    |                       p=0.5    EV= $135.0
```

**Figure15**

## WHAT IS THE DIFFERENCE BETWEEN PROC HPSPLIT AND PROC DTREE

Firstly, some decision trees created by HPSPLIT will not show the what's going on in the inner components whereas PROC DTREE shows the inner components directly.

Secondly: they focus on different intentions. HPSPLIT procedure explores the relationship between factors and response variable. This process facilitates moving through the variables in the data to determine their effect on the outcome. In contrast, the DTREE procedure has a precondition that decision maker knowns or can predict the way factors are distributed among groups, then the target is to find the optimal outcome given the combination of costs and variables.

Thirdly: The process of HPSPLIT starts with building a large and full tree, then use various measures, such as Entropy, Gini index, RSS to split leaves for each node. In order prevent overfitting, the full tree is pruned back to a smaller subtree. On the other hand, the process of DTREE is for decision analysis. The first construction is to build a decision model to represent problem.

## REFRENCES

The Hpsplit procedure
http://support.sas.com/documentation/cdl/en/stathpug/68163/HTML/default/viewer.htm#stathpug_hpsplit_examples01.htm

The DTREE Procedure
http://www.math.wpi.edu/saspdf/orpm/chap3.pdf

Decision Trees: a Gentle Introduction
Richard D. Hector, Ph.D., M.P.H., M.A., Arizona Care Network, Phoenix, Arizona
https://www.lexjansen.com/wuss/2017/123_Final_Paper_PDF.pdf

Entropy: How Decision Trees Make Decisions
https://towardsdatascience.com/entropy-how-decision-trees-make-decisions-2946b9c18c8

An introduction to classification and regression trees with PROC HPSPLIT
Peter L. Flom Peter Flom Consulting, LLC
http://www.mwsug.org/proceedings/2018/AA/MWSUG-2018-AA-42.pdf

## ACKNOWLEDGMENTS

Thanks to the people at SAS Tech Support.

## CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

YuTing Tian, tianyangzi2017@gmail.com

## APPENDIX

```
/*create two classes*/
%macro entropy(X_avg= , Y_avg= , StdDev= ,NoOfpoints=, class=);
do id=1 to &NoOfPoints;
  class=&class;
  x=&X_avg + rand('Normal',0,&stdDev);
  y=&Y_avg + rand('Normal',0,&stdDev);
  output;
end;
%mend entropy;

%entropy(X_avg= , Y_avg= , StdDev= ,NoOfpoints=);

data tree;
  %make_Points(X_avg=2 , Y_avg=3 , StdDev=.4 ,NoOfpoints=30 ,
  class="positive");
  %make_Points(X_avg=4 , Y_avg=9 , StdDev=.4 ,NoOfpoints=70 ,
  class="negative");

run;

proc sgplot data=tree;
  title "sampling cluster";
  scatter Y=Y x=X / markerattrs = (size = 10 symbol = circlefilled)
  group=class;
run;

/*HPSPLIT procedure*/
data iris;
   input Sepallen Sepalwid Petallen Petalwid Spec_No;
   datalines;
50 33 14 02 1
64 28 56 22 3
65 28 46 15 2
67 31 56 24 3
63 28 51 15 3
46 34 14 03 1
69 31 51 23 3
62 22 45 15 2
59 32 48 18 2
46 36 10 02 1
61 30 46 14 2
60 27 51 16 2
65 30 52 20 3
```

```
56 25 39 11 2
65 30 55 18 3
58 27 51 19 3
68 32 59 23 3
51 33 17 05 1
57 28 45 13 2
62 34 54 23 3
77 38 67 22 3
63 33 47 16 2
67 33 57 25 3
76 30 66 21 3
49 25 45 17 3
55 35 38 11 1
67 30 52 23 3
70 32 47 14 2
64 32 45 15 2
61 28 40 13 2
48 31 16 02 1
59 30 51 18 3
55 24 38 11 2
63 25 50 19 3
64 32 53 23 3
52 34 14 02 1
49 36 14 01 1
54 30 45 15 2
79 38 64 20 3
44 32 13 02 1
67 33 57 21 3
50 35 16 06 1
58 26 40 12 2
44 30 13 02 1
77 28 67 20 3
63 27 49 18 3
47 32 16 02 1
55 26 44 12 2
50 23 33 10 2
72 32 60 18 3
48 30 14 03 1
51 38 16 02 1
61 30 49 18 3
48 34 19 02 1
50 30 16 02 1
50 32 12 02 1
61 26 56 14 3
```

```
64 28 56 21 3
43 30 11 01 1
58 40 12 02 1
51 38 19 04 1
67 31 44 14 2
62 28 48 18 3
49 30 14 02 1
51 35 14 02 1
56 30 45 15 2
58 27 41 10 2
50 34 16 04 1
46 32 14 02 1
60 29 45 15 2
57 26 35 10 2
57 44 15 04 1
50 36 14 02 1
77 30 61 23 3
63 34 56 24 3
58 27 51 19 3
57 29 42 13 2
72 30 58 16 3
54 34 15 04 1
52 41 15 01 1
71 30 59 21 3
64 31 55 18 3
60 30 48 18 3
63 29 56 18 3
49 24 33 10 2
56 27 42 13 2
57 30 42 12 2
55 42 14 02 1
49 31 15 02 1
77 26 70 23 3
60 22 50 15 3
54 39 17 04 1
66 29 46 13 2
52 27 39 14 2
60 34 45 16 2
50 34 15 02 1
44 29 14 02 1
50 20 35 10 2
55 24 37 10 2
58 27 39 12 2
47 32 13 02 1
```

```
46 31 15 02 1
69 32 57 23 3
62 29 43 13 2
74 28 61 19 3
59 30 42 15 2
51 34 15 02 1
50 35 13 03 1
56 28 49 20 3
60 22 40 10 2
73 29 63 18 3
67 25 58 18 3
49 31 15 01 1
67 31 47 15 2
63 23 44 13 2
54 37 15 02 1
56 30 41 13 2
63 25 49 15 2
61 28 47 12 2
64 29 43 13 2
51 25 30 11 2
57 28 41 13 2
65 30 58 22 3
69 31 54 21 3
54 39 13 04 1
51 35 14 03 1
72 36 61 25 3
65 32 51 20 3
61 29 47 14 2
56 29 36 13 2
69 31 49 15 2
64 27 53 19 3
68 30 55 21 3
55 25 40 13 2
48 34 16 02 1
48 30 14 01 1
45 23 13 03 1
57 25 50 20 3
57 38 17 03 1
51 38 15 03 1
55 23 40 13 2
66 30 44 14 2
68 28 48 14 2
54 34 17 02 1
51 37 15 04 1
```

```
52 35 15 02 1
58 28 51 24 3
67 30 50 17 2
63 33 60 25 3
53 37 15 02 1
;
run;

data iris2;
   set iris;
   length Species $12.;
   if spec_no=1 then
       species='Setosa';
   else if spec_no=2 then
        species='Versicolor';
    else if spec_no=3 then
         species='Virginica';

run;

PROC HPSPLIT data=iris2 cvemethod=random(10) seed=123 plots=all;
    class species;
    model Species=PetalLen PetalWid SepalLen SepalWid;
    output out=hpsliout;
    grow entropy;
    prune costcomplexity;
run;quit;

/*DTREE procedure*/
data Iris1;
      format _STNAME_ $17. _STTYPE_ $6. _OUTCOM_ $18.
             _SUCCES_ $12. ;
      input _STNAME_ $ _STTYPE_ $ _OUTCOM_ $ _SUCCES_ $ ;

      datalines;
    species       D       Setosa               petal_width
    .             .      Versicolor             petal_width
    .             .      Virginica              petal_width
    petal_width    C      less_than_8            petal_length
    .             .      between_8and_16.5      petal_length
    .             .      greater_than_16.5      petal_length
    petal_length  C      less_or_greater_49.5  .
     .            .      greater_than_49.5      .

    ;
quit;
```

```
  data Iris2;
      input _EVENT1 $23.  _PROB1 _EVENT2 $19. _PROB2
           _EVENT3 $19.  _PROB3 ;
      datalines;
   less_than_8       0.2   between_8and_16.5  0.6   greater_than_16.5
0.2
   less_or_equal_49.5 0.5   greater_than_49.5  0.5   .                .
 ;run;data

Iris3;
  input(_state1-_state3) ($19.);
  format _value_ dollar12.0;
  if _state1="less_than_8" then _cost1_=10;
  else if _state1="between_8and_16.5" then _cost1_=20;
  else _cost1_=30;

  if _state2="less_or_equal_49.5" then _cost2_ =40;
  else _cost2_=70;

  if _state3= "Setosa" then _value_= 3*( _cost1_ + _cost2_);
  if _state3= "Versicolor" then _value_= 1.5*(_cost1_ + _cost2_);
  else _value_= 3*( _cost1_ + _cost2_);

datalines;
less_than_8         less_or_equal_49.5     Setosa
less_than_8         less_or_equal_49.5     Versicolor
less_than_8         less_or_equal_49.5     Virginica
less_than_8         greater_than_49.5      Setosa
less_than_8         greater_than_49.5      Versicolor
less_than_8         greater_than_49.5      Virginica
between_8and_16.5    less_or_equal_49.5     Setosa
between_8and_16.5    less_or_equal_49.5     Versicolor
between_8and_16.5    less_or_equal_49.5     Virginica
between_8and_16.5    greater_than_49.5      Setosa
between_8and_16.5    greater_than_49.5     Versicolor
between_8and_16.5    greater_than_49.5     Virginica
greater_than_16.5    less_or_equal_49.5     Setosa
greater_than_16.5    less_or_equal_49.5     Versicolor
greater_than_16.5    less_or_equal_49.5     Virginica
greater_than_16.5    greater_than_49.5     Setosa
greater_than_16.5    greater_than_49.5     Versicolor
greater_than_16.5    greater_than_49.5     Virginica
;
```

```
DTREEPROC DTREE stagein=Iris1
               probin=Iris2
               payoffs=Iris3
               nowarning;
        evaluate;
        summary;
        OPTIONS LINESIZE=100;
        treeplot/ lineprinter;
 quit;
```