ILLINOIS
Gies College of Business

Decision Trees

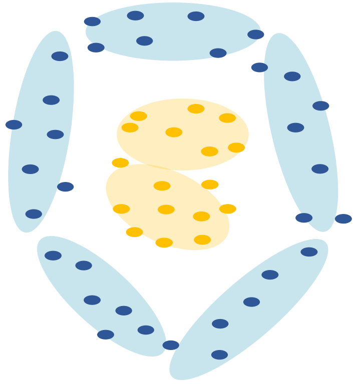
Knowledge in Trees and Trees from Data

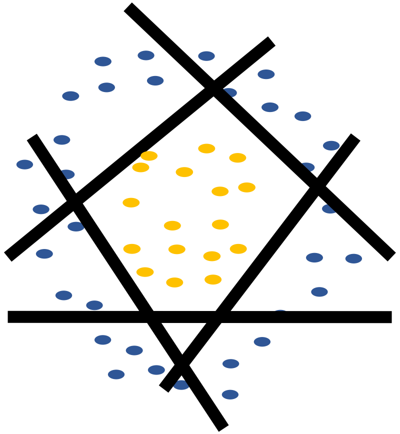
Sridhar Seshadri

Contents

Descriptive vs. Discriminative

Two Types of Classifier Paradigms





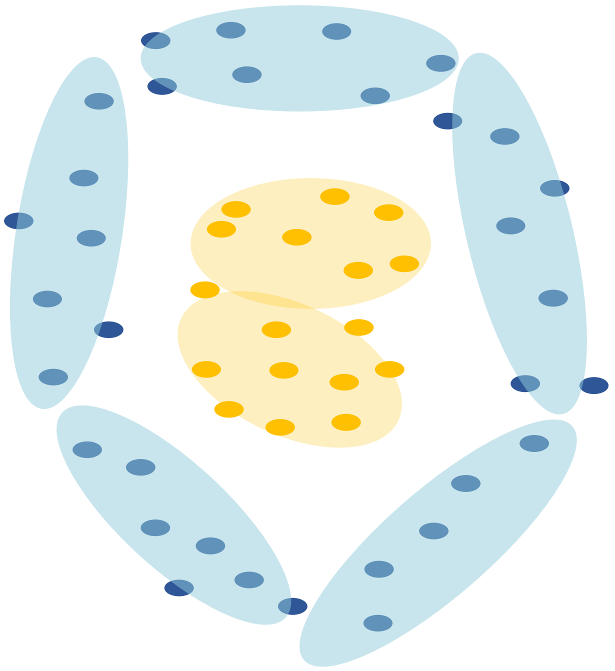
Descriptive **Classifiers**

Bayesian Classifiers
Nearest Neighbor
Parzen Window

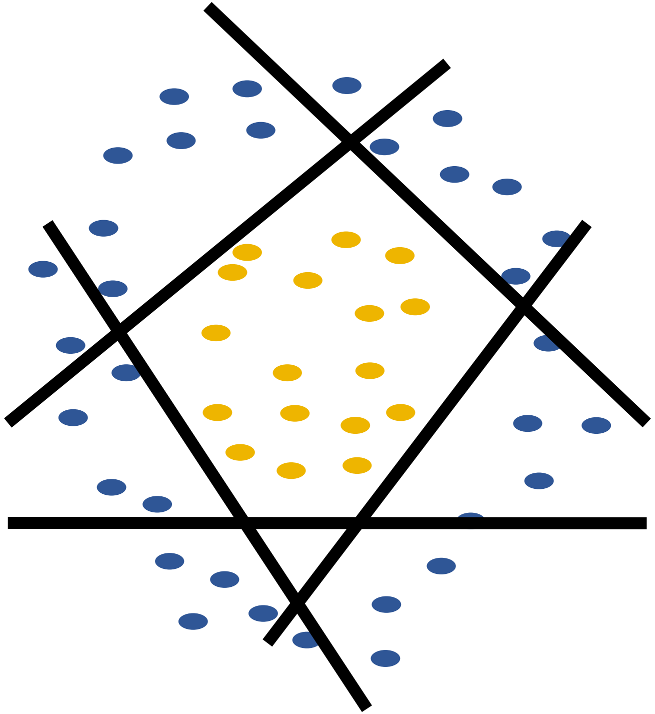
Discriminative **Classifiers**

Decision Trees
Neural Networks,
Support Vector Machines

Two Types of Classifier Paradigms



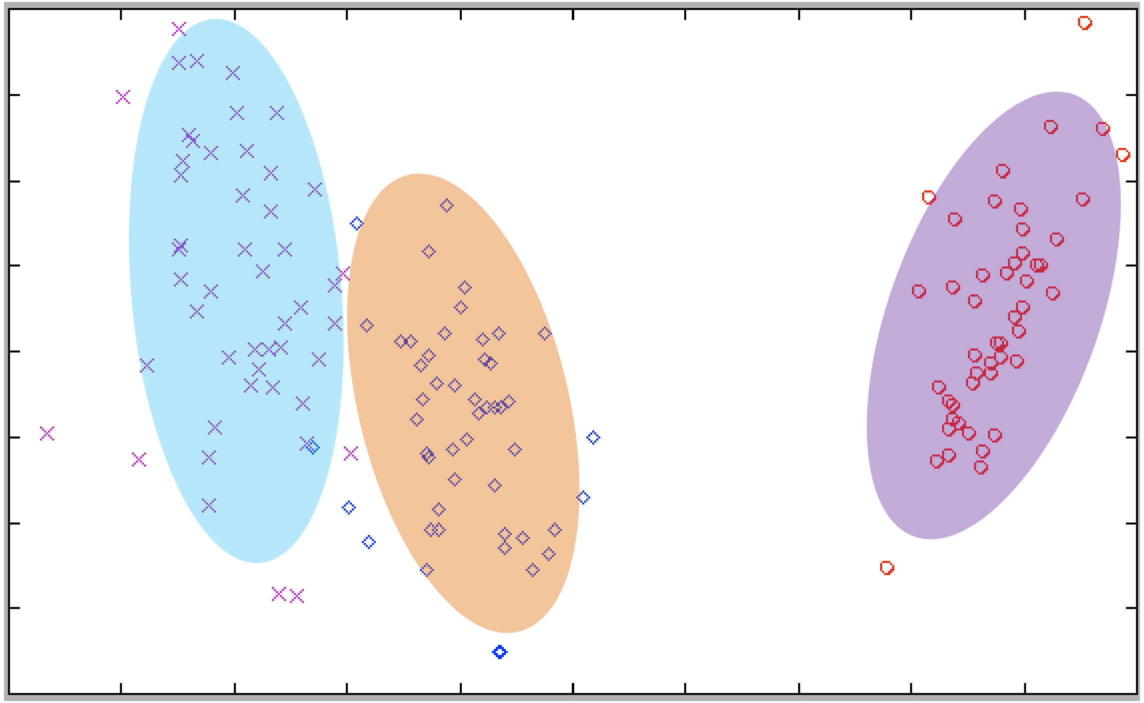
Descriptive Classifiers



Discriminative Classifiers

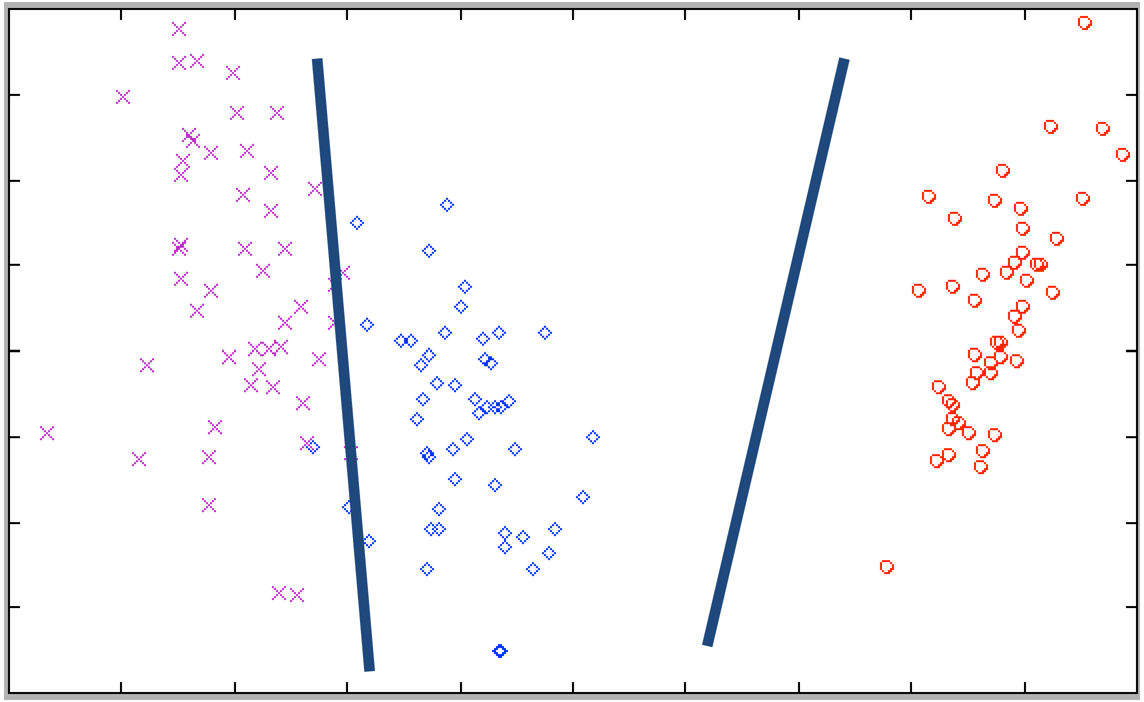
IRIS – Descriptive Classifier

Class Density Function: $P(\mathbf{x}|c)$



IRIS – Discriminative Classifier

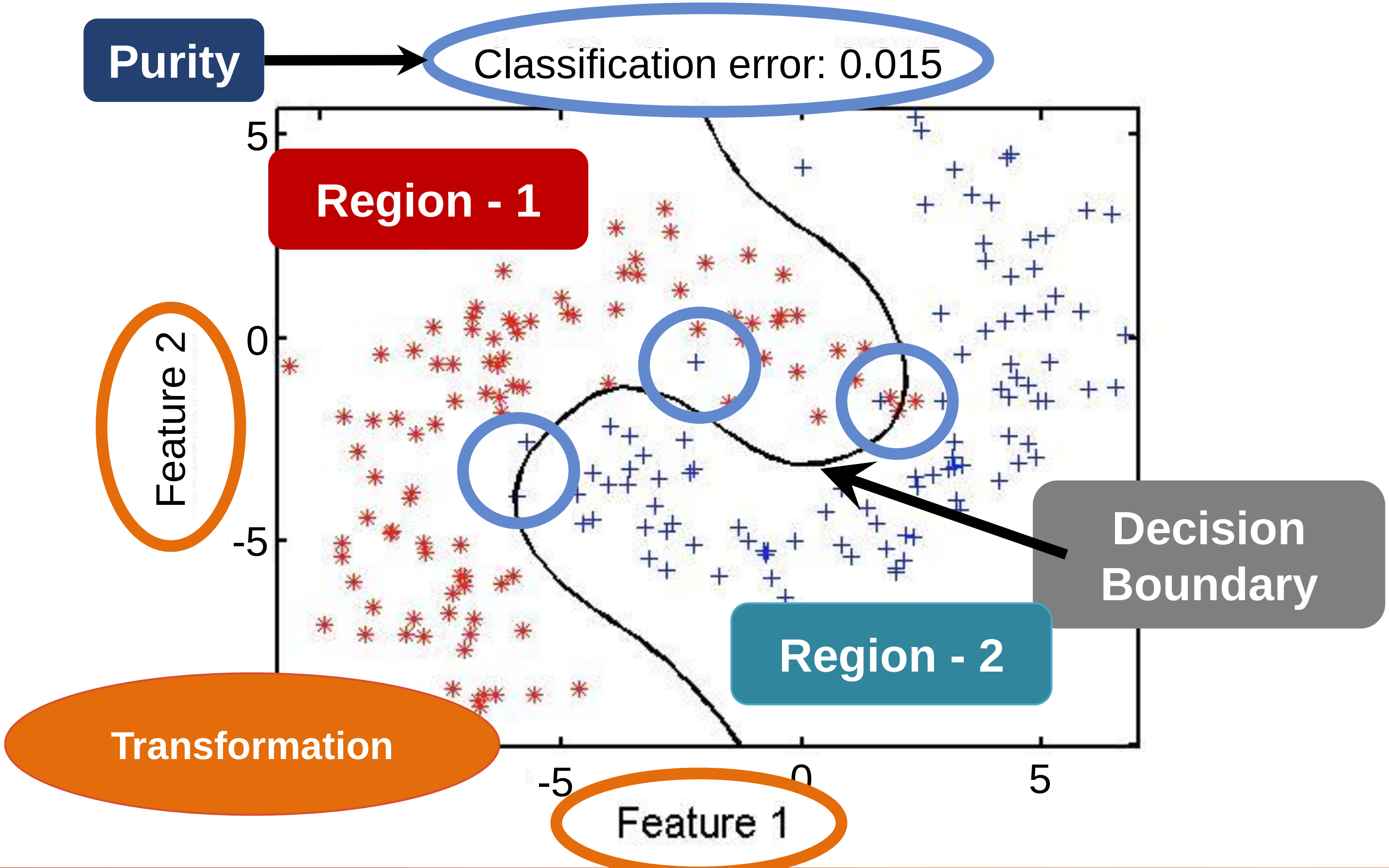
Class Discriminators: $\mathbf{w}^T \mathbf{x} < q$



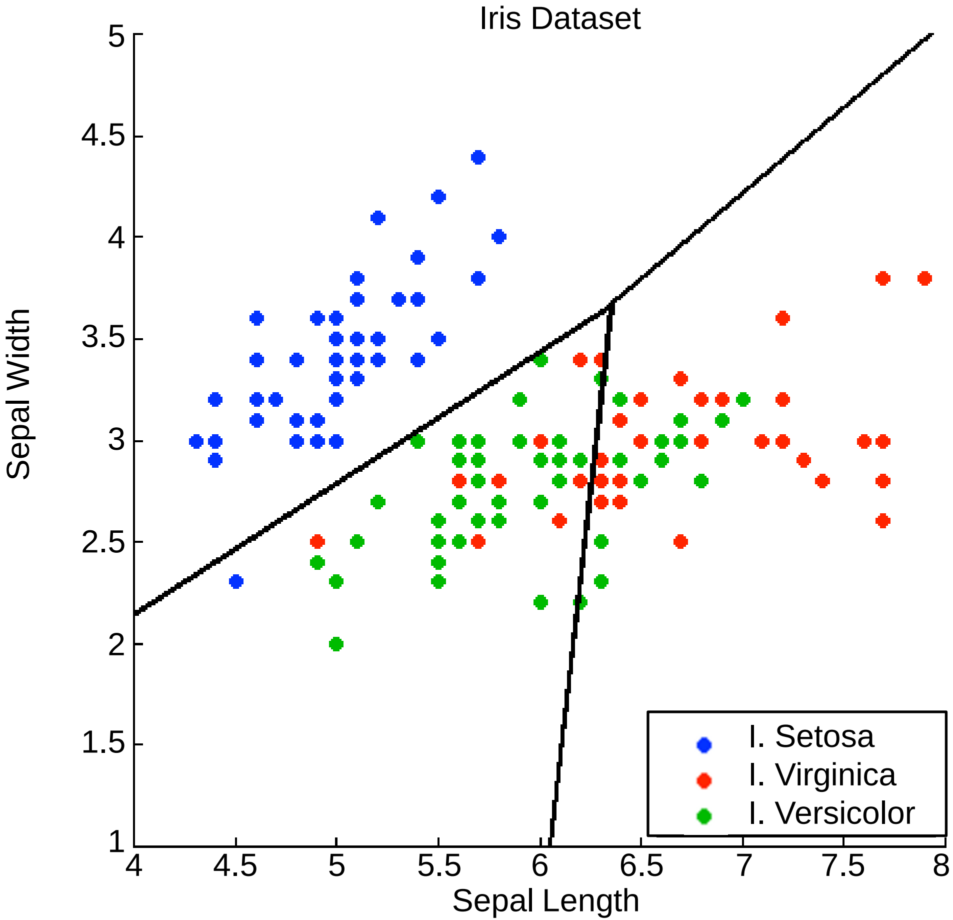
What is Discriminative “Classification”?

PARTITIONING the (**FEATURE**) **SPACE** into **PURE REGIONS** assigned to each **CLASS**

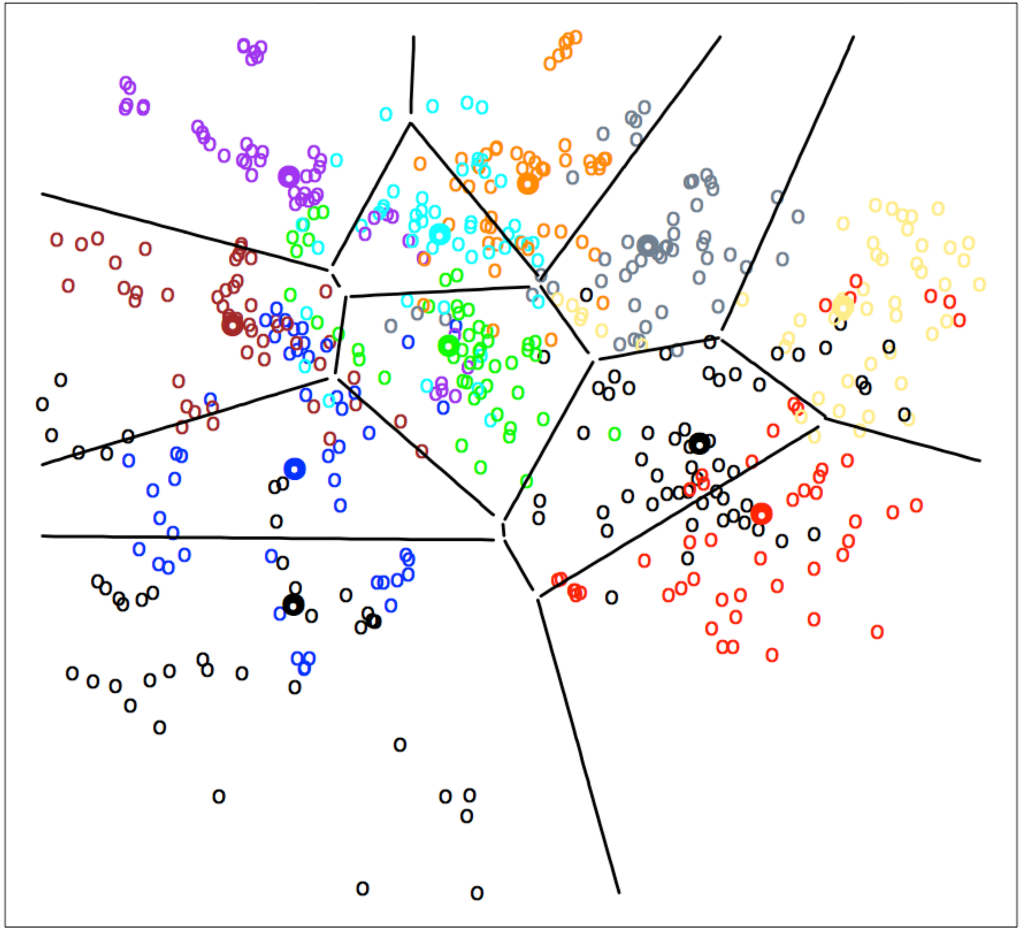
Partitioning Into Pure Region



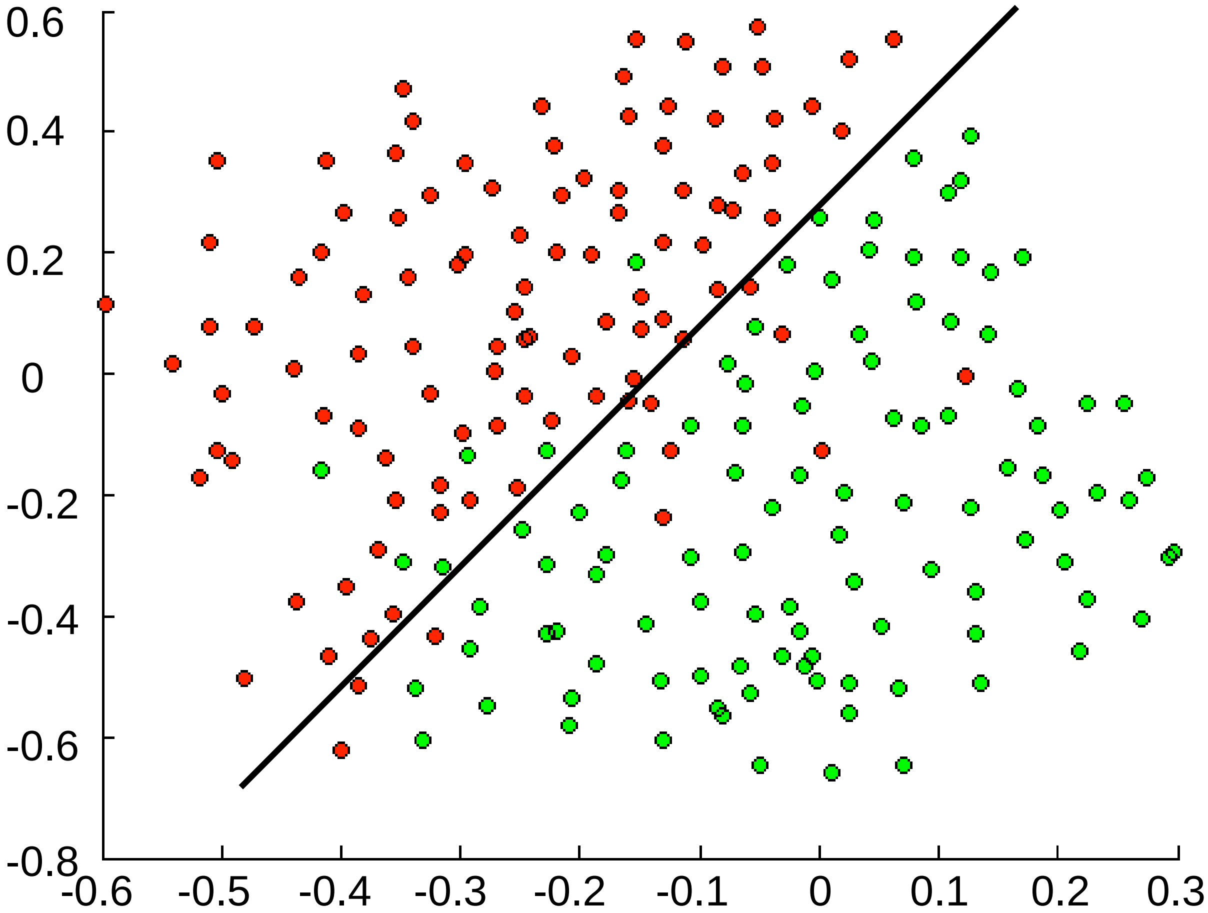
Decision Boundary for IRIS Data



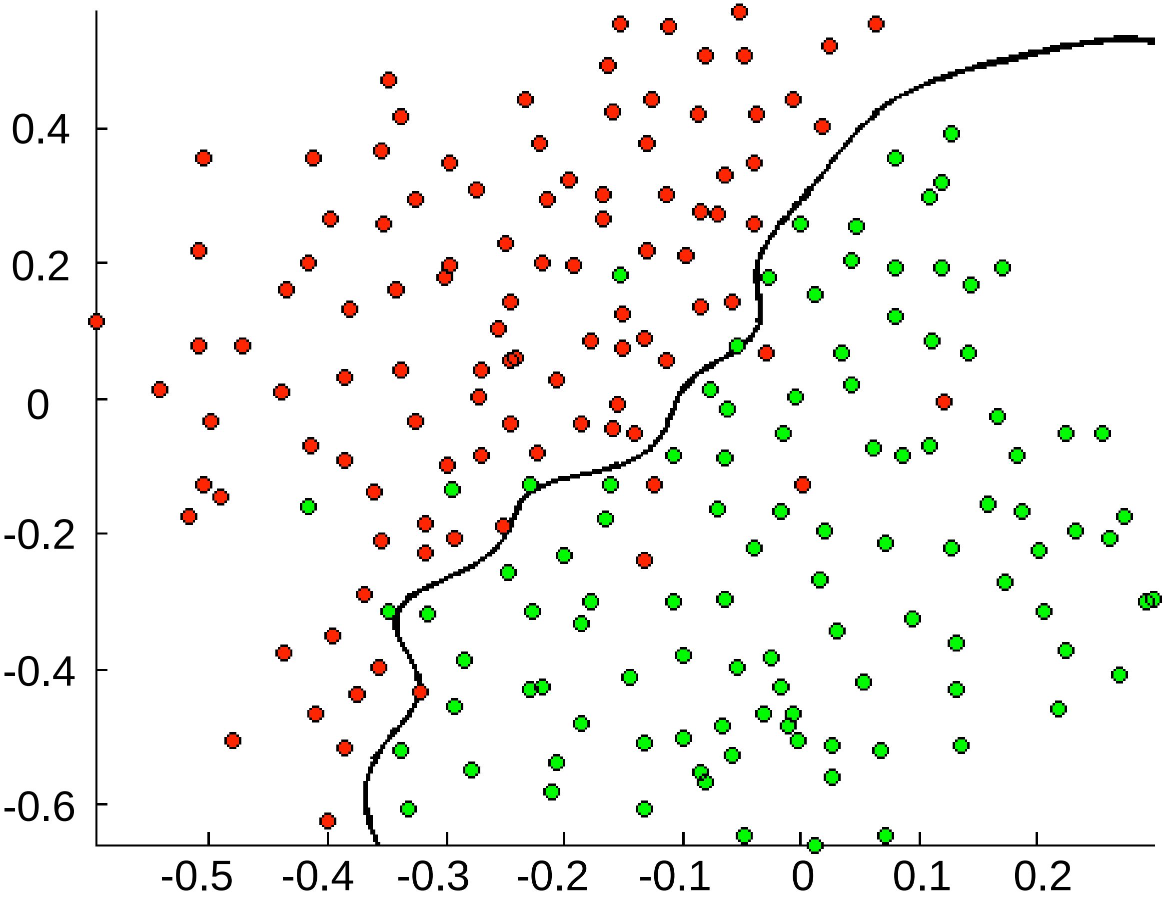
Decision Boundaries for DIGITS Data



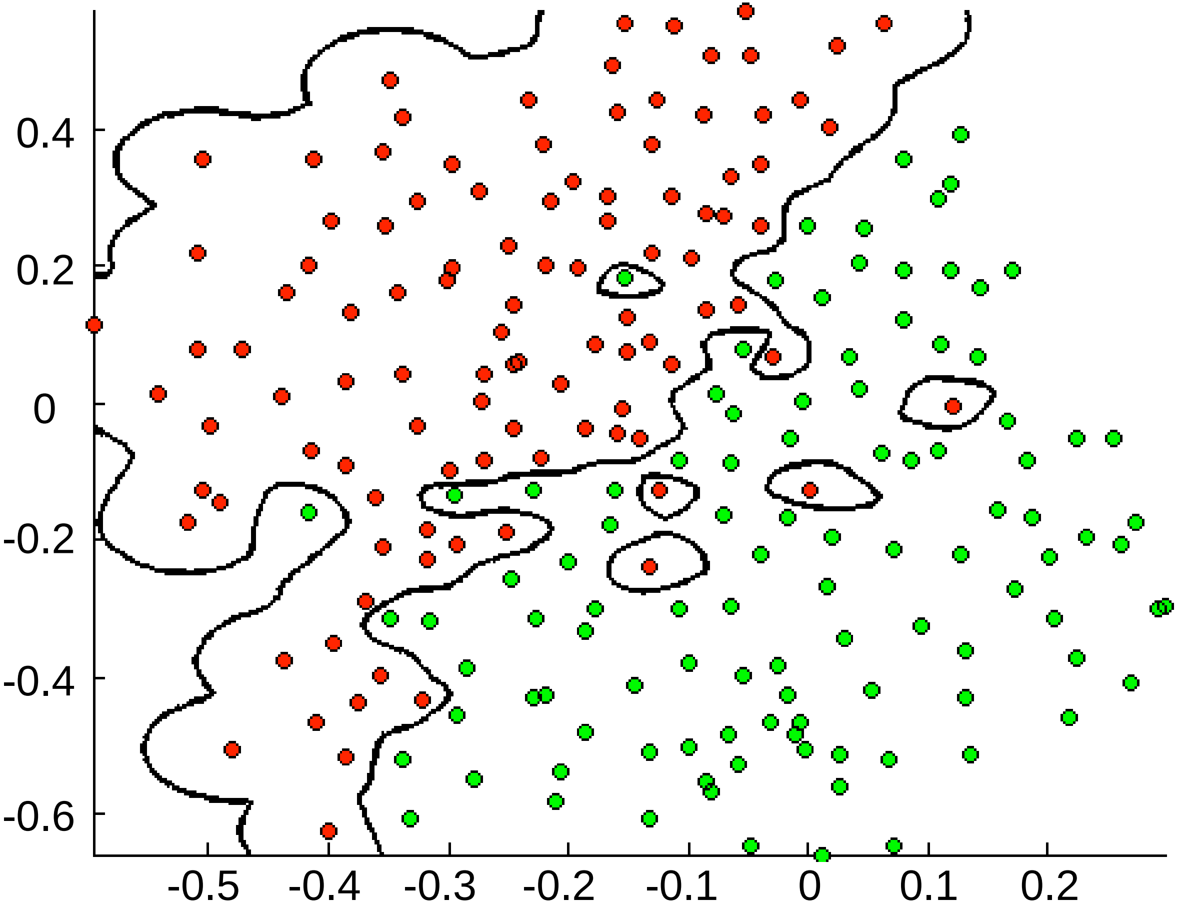
SIMPLE Decision Boundary?



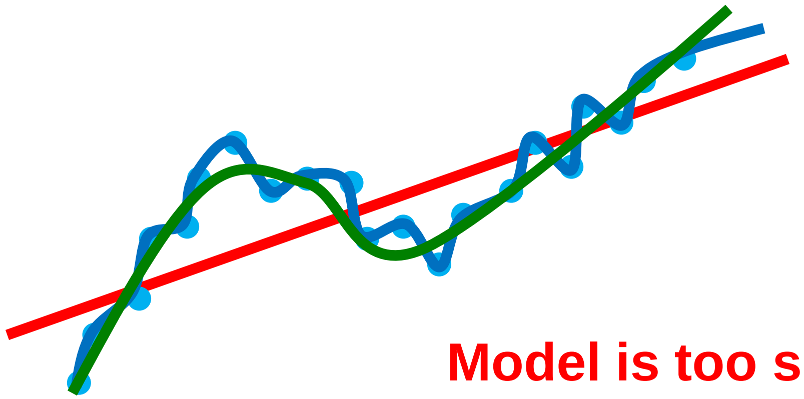
MEDIUM Decision Boundary



COMPLEX Decision Boundary



Model SIGNAL not NOISE

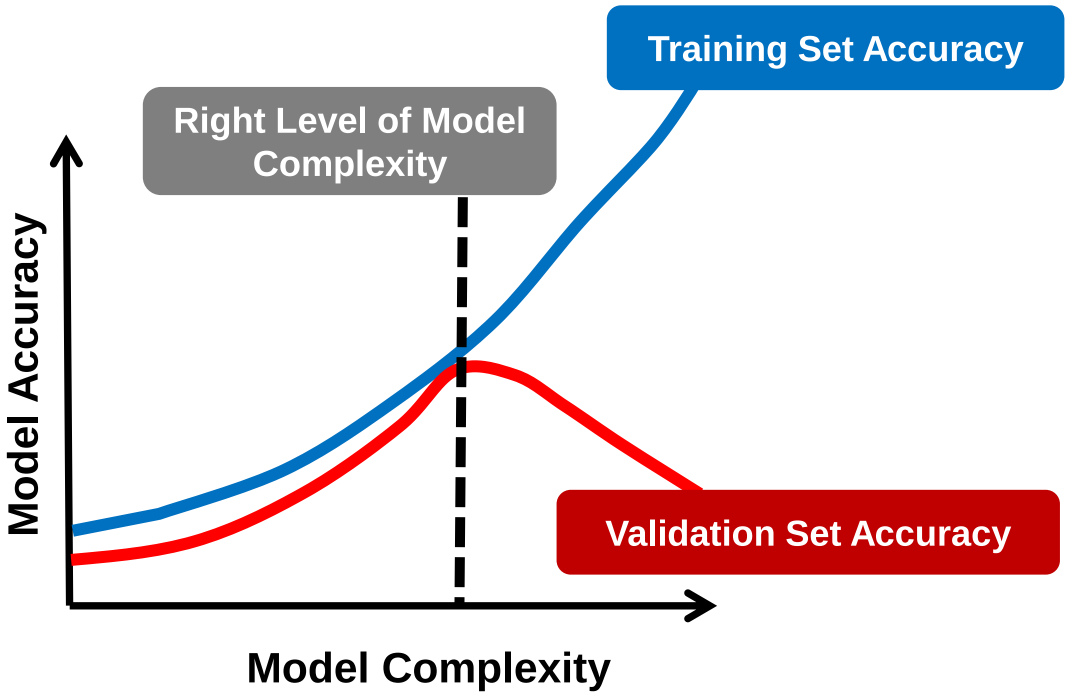


Model is too simple → UNDER LEARN

Model is too complex → MEMORIZE

Model is just right → GENERALIZE

Generalize, Don’t Memorize!



Questions for Classification

What is the **NATURE** of classifier’s **DECISION BOUNDARY**?

Depends on the classifier type!

What is the **COMPLEXITY** of classifier’s **DECISION BOUNDARY**?

Any classifier can be made more or less complex!

How do I **CONTROL** the **COMPLEXITY** of the classifier?

What knob to use to make classifier more/less complex?

Questions for Classification

How do I know when the classifier is **COMPLEX ENOUGH**?

- Degree of Noise vs. Structure in the data
- Too noisy → Less complexity

How to pick the right **CLASSIFIER** to use?

- Nature of the data - structured/unstructured, numeric/categorical
- Complexity of the decision surface needed!

Rule Based Classifiers

Domain Knowledge Based

Fever Rules

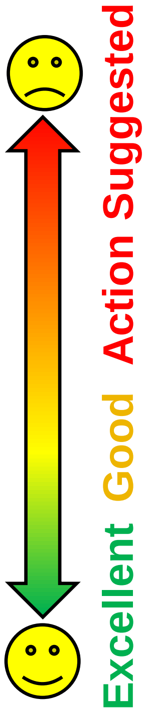
If **TEMP <= 98.6** → **NORMAL**

If 98.6 <= **TEMP <= 100.0** → **MILD-FEVER**

If 100 <= **TEMP < 102.0** → **MEDIUM-FEVER**

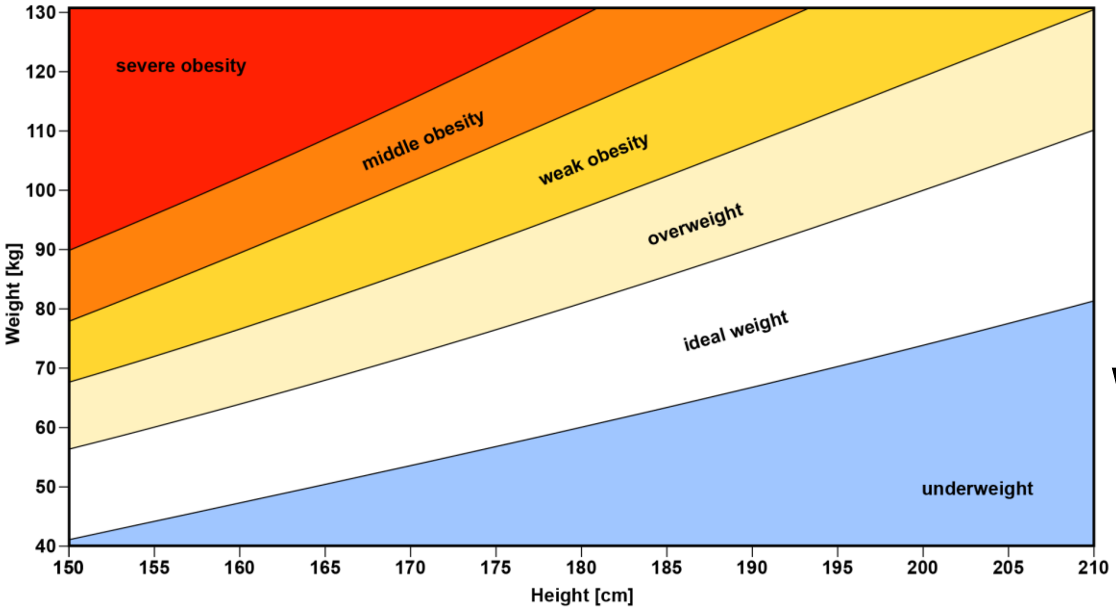
If **TEMP > 104.0** → **HIGH-FEVER**

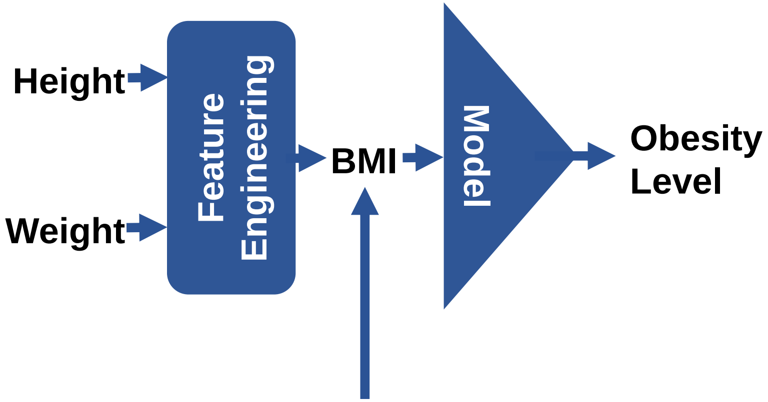
These can also be called “expert” systems.



HbA_{1c}	Mean Blood Glucose	
Test Score	mg/dL	mmol/L
14.0	380	21.1
13.0	350	19.3
12.0	315	17.4
11.0	280	15.6
10.0	250	13.7
9.0	215	11.9
8.0	180	10.0
7.0	150	8.2
6.0	115	6.3
5.0	80	4.7
4.0	50	2.6

Obesity Classifier

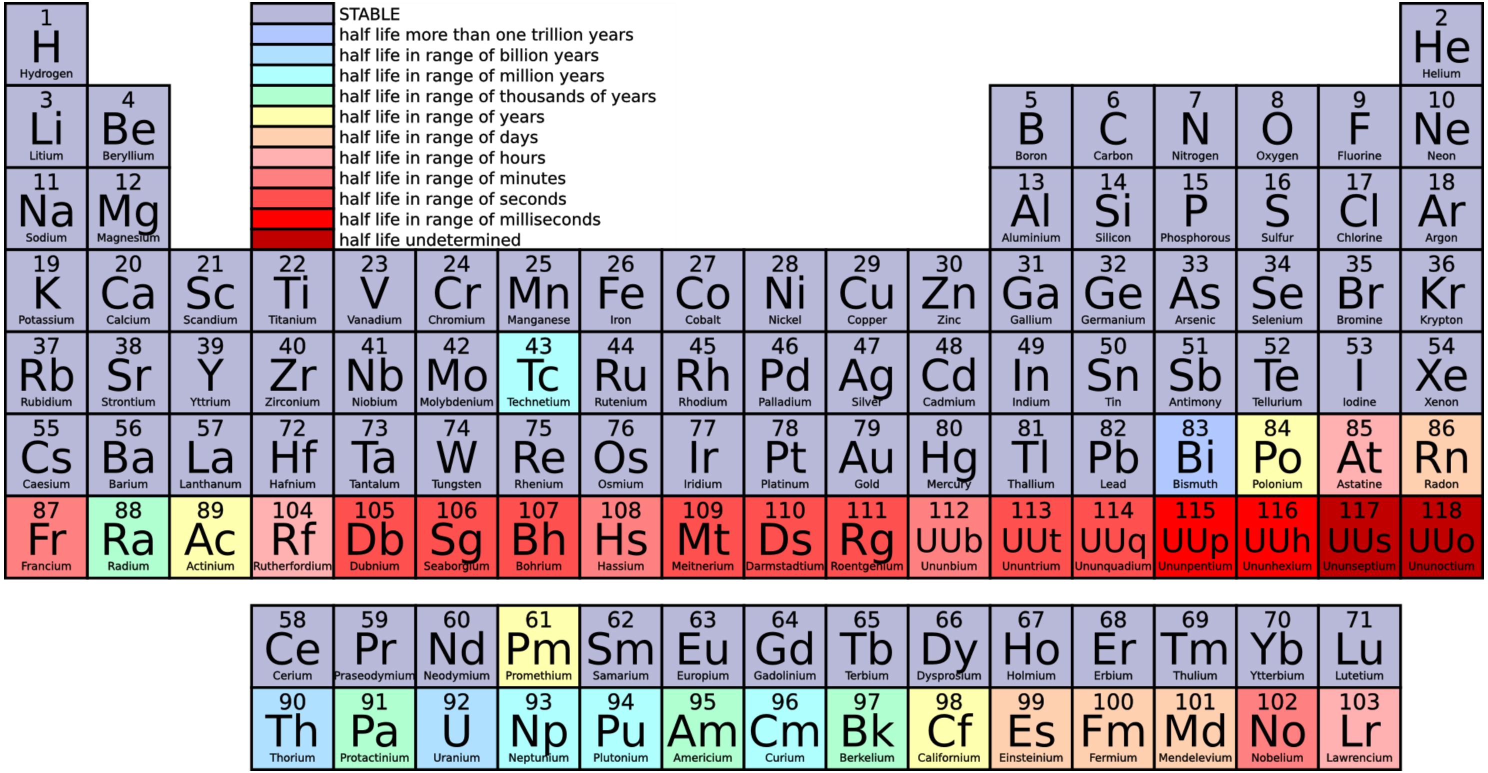




Engineered Feature based on Domain Knowledge

Source: https://commons.wikimedia.org/wiki/File:BMI_en.svg

Elements Classifier



Decision Trees Learning Rules From Data

“Toy” Dataset

Data is in Decision_Tree_Ex.csv

Problem Description:

Ponting Flower uses five features A, B, C, D, and E to describe flowers. For example, feature A corresponds to fragrance. Each feature can take two values, for example, delicate fragrance is labeled A1 and intense fragrance as A2. The values for features B, C, D and E are {B1, B2}, {C1, C2} …, {E1, E2}. If a particular combination is liked by at least 35% of the customers then it is labeled as green, otherwise it is labeled as red. The file contains data for 111 flowers.

A	B	C	D	E	COLOR
A1	B1	C2	D1	E2	GREEN
A2	B2	C2	D1	E1	RED
A2	B2	C1	D1	E1	GREEN
A1	B2	C1	D1	E2	GREEN
A1	B1	C2	D2	E2	GREEN
A2	B2	C2	D1	E1	RED

Process of Building a Decision Tree From Data

Partitioning and recursive partitioning

Purity measures

Which Feature to Pick First?

Idea is to compute the purity gain from partitioning

Entropy of a region can be calculated as: $\sum -p(c) * \log_2 (p(c))$ where the summation is taken over all classes and p(c): Probability of a class in that region

The definition of purity is 1 – entropy (if binary classification)

An Algorithm

Step 1: Calculate the entropy for the overall dataset

Step 2: Calculate the entropy for splitting using each attribute.

Step 3: Calculate the Purity Gain for splitting using each attribute using the previous formula.

An Algorithm

Step 4: Select the attribute with the highest Purity Gain and split the data on the basis of this attribute.

Step 5: Repeat Step 1 to Step 4 until all the data is classified

Sample Steps

Step 1: Calculate the entropy for the overall dataset:

Total Number of Observation: n (Total) = 110

Number of Observations with Green class: n (Green) = 89

Probability for Green Class: p (Green) = 89/110 = 0.8091

-p(Green) * Binary Logarithm (p(Green)) = -0.8091*-0.306= 0.2473

Number of Observations with Red class: n (Red) = 21

Probability for Red Class: p(Red) = 21/110 = 0.1909

-p(Red) * Binary Logarithm (p(Red)) = -0.1909*-2.3890= 0.4561

Entropy (Overall Data) = 0.2473+0.4561 = 0.7034   (i.e., Entropy (0))

Step 2

Calculate Entropy for splitting using each attribute. Take for example, attribute A

We have two categorical values A1 and A2. So to calculate Entropy for attribute A [entropy(A,COLOR)] have to calculate Entropy of A1 and A2 on the basis of Color and then multiply it by the fraction of samples that have feature A1 and A2, labeled as p(A1) and p(A2).

Entropy (A, COLOR) =

p(A1)* Entropy (A1, COLOR) + p(A2)*Entropy (A2, COLOR)

Entropy (A, COLOR) = 0.4545*0 + 0.5455*0.934068 = 0.509492

Purity Gain for Attribute A = Entropy (COLOR) – Entropy (A, COLOR)

Purity Gain (A, COLOR) = 0.7034 - 0.509492= 0.193878

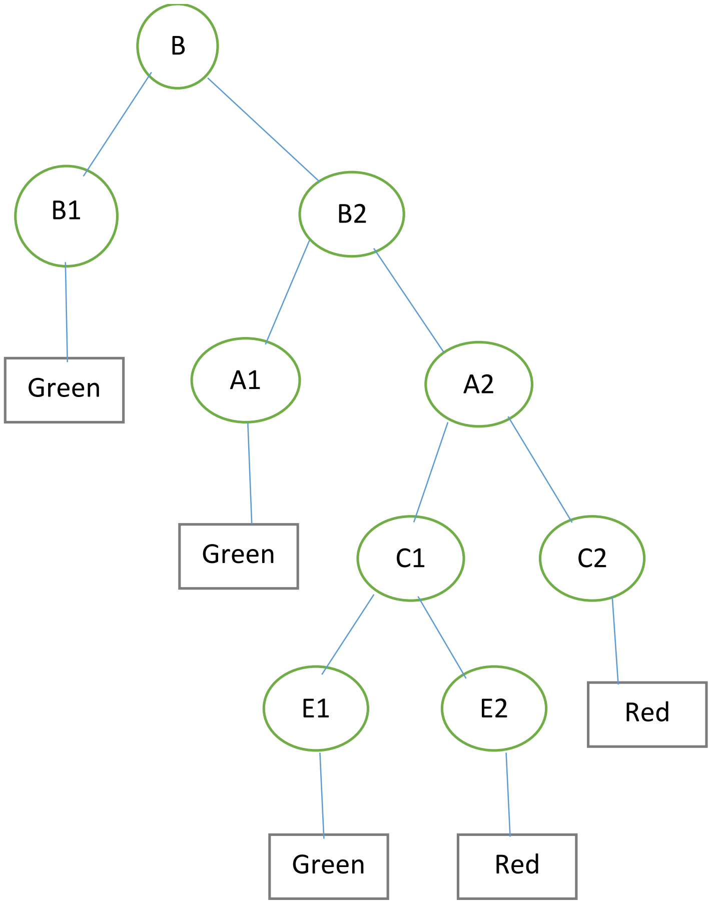
Step 3:

Repeat for all attributes

	Entropy	Purity Gain
Overall	0.703369	
A	0.509492	0.193878
B	0.453166	0.250203
C	0.695062	0.008307
D	0.695654	0.007715
E	0.693644	0.009725

Split on B

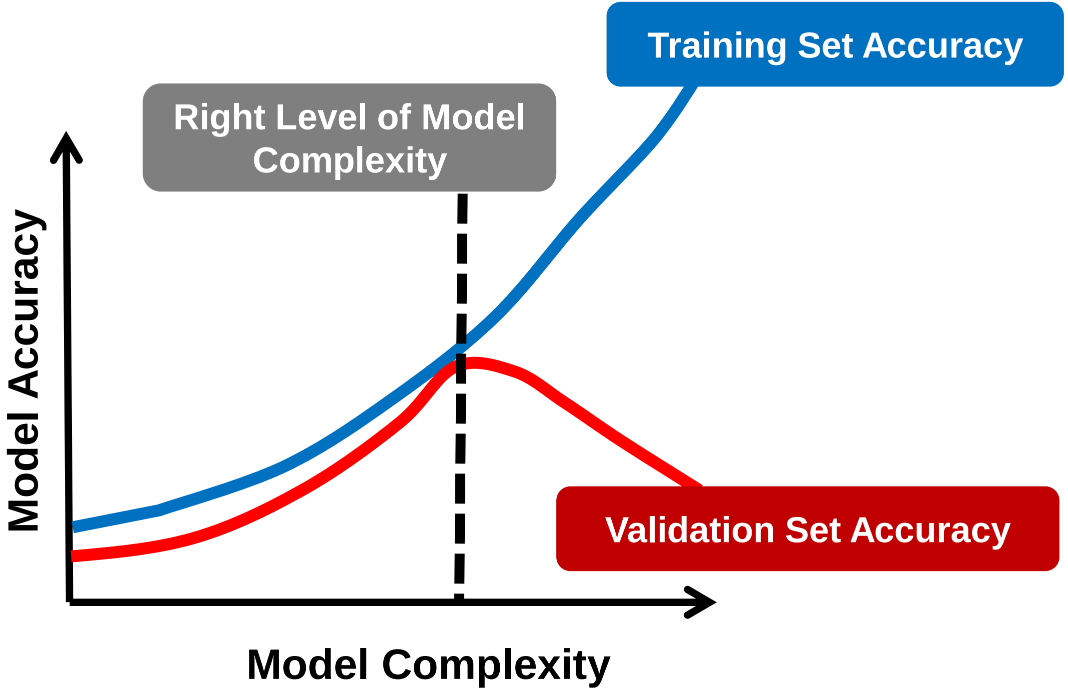
Final Tree



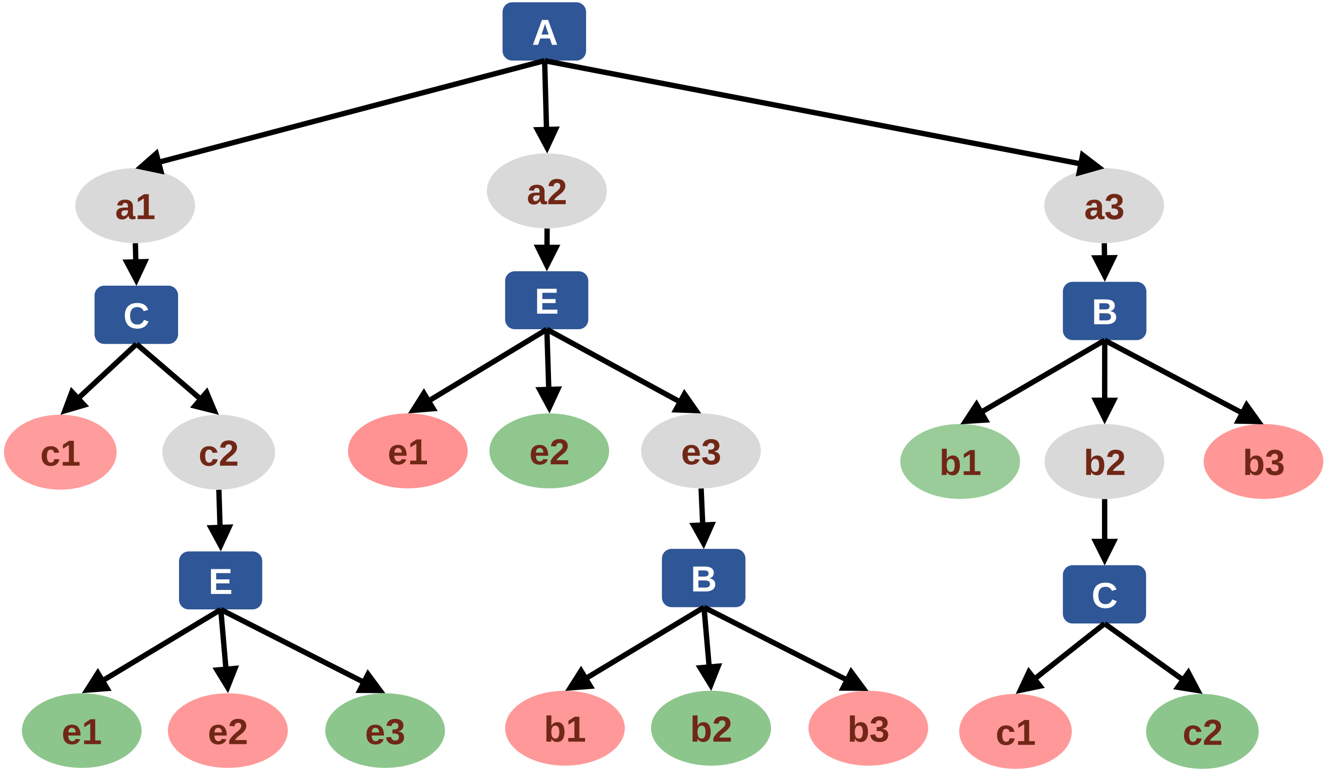
What are the rules?

There are 5!

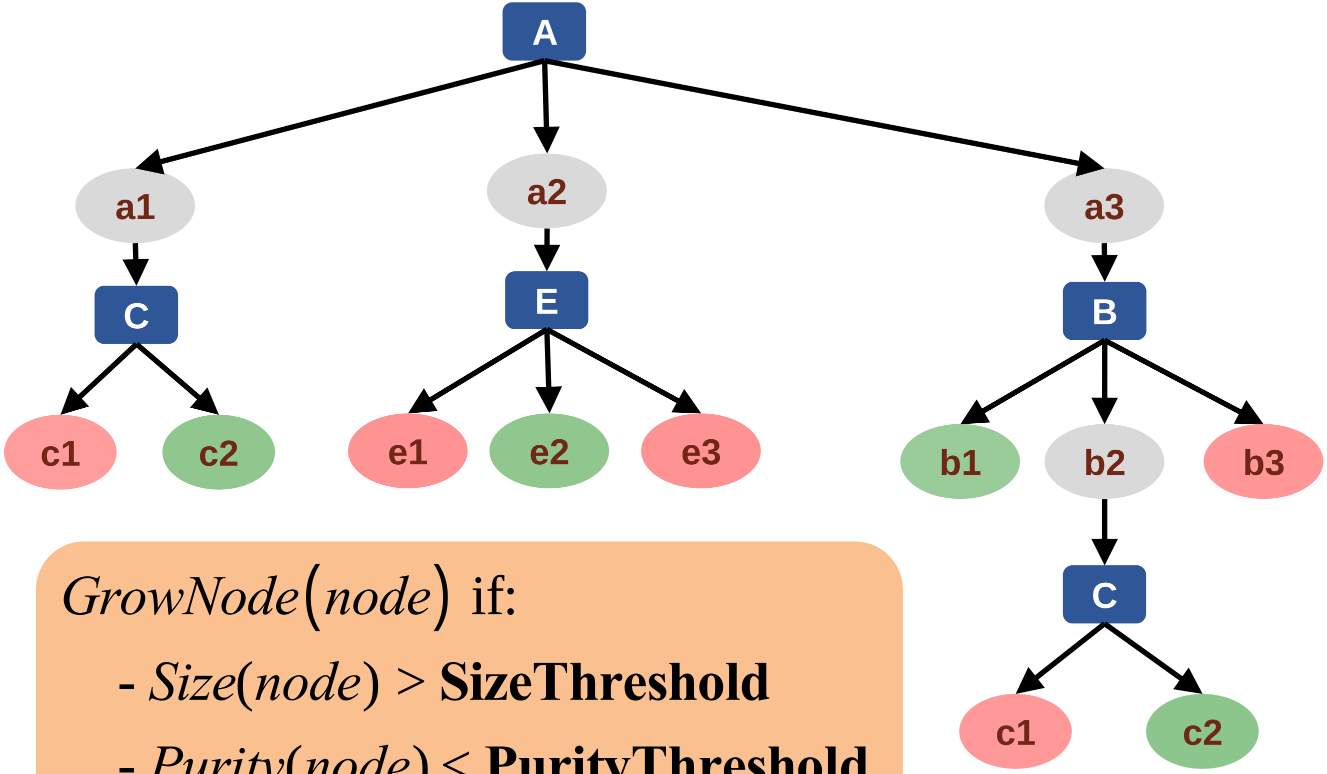
Controlling COMPLEXITY



Example of Overfitting



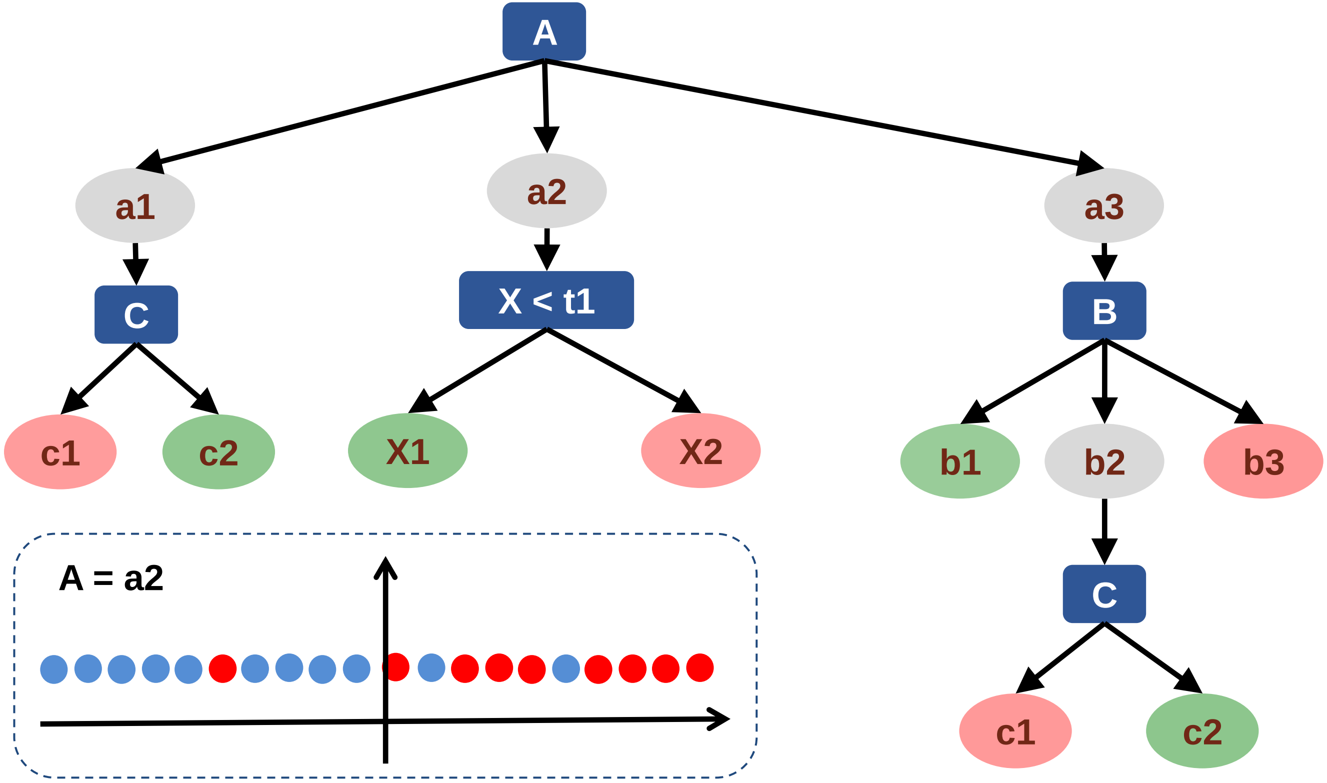
Pruning



$GrowNode(node)$ if:

- $Size(node) >$ **SizeThreshold**
- $Purity(node) <$ **PurityThreshold**

Numeric Attributes



X < t1

Decision Tree vs Clustering (Kmeans/Hclust)

Decision trees are also used for clustering; observations in the same leave make a cluster.

The difference between decision tree and clustering techniques is that decision trees can be applied on labelled data only.

Thus, whenever target values (labels) are available, we should prefer decision trees to clustering methods(unsupervised).

Type of Decision Tree

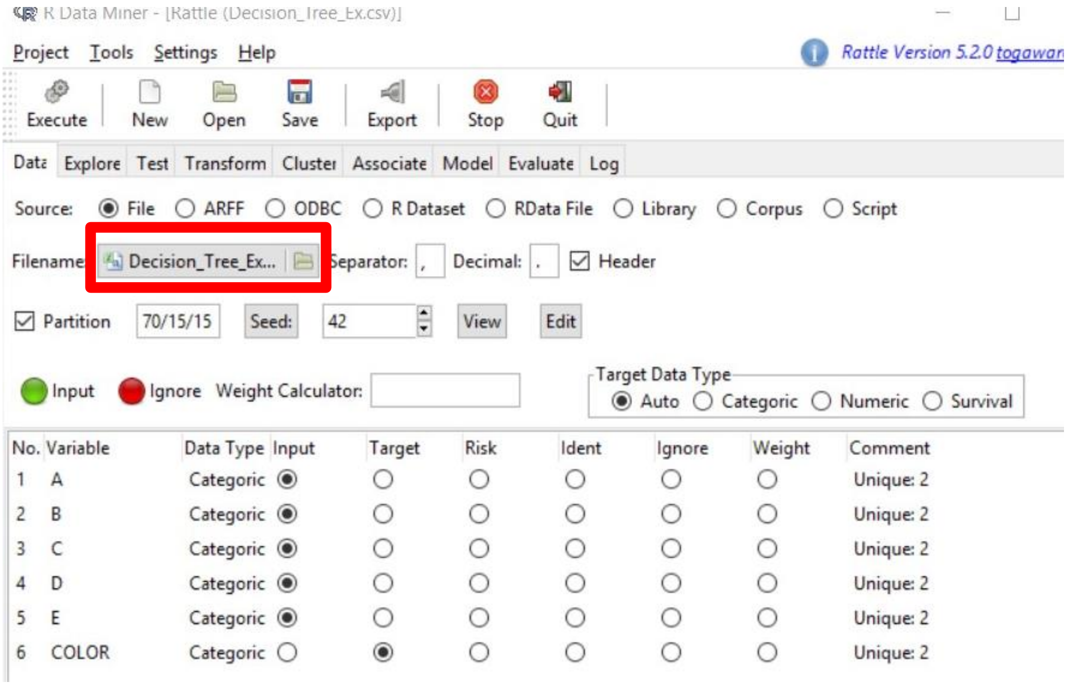
Classification Tree – when target value is a class

Regression Tree – When target value is numeric

We will discuss examples of both in Rattle

Classification Tree – Toy Example

Objective – To classify flowers as liked by at least 35% vs not liked (Green vs. Red).

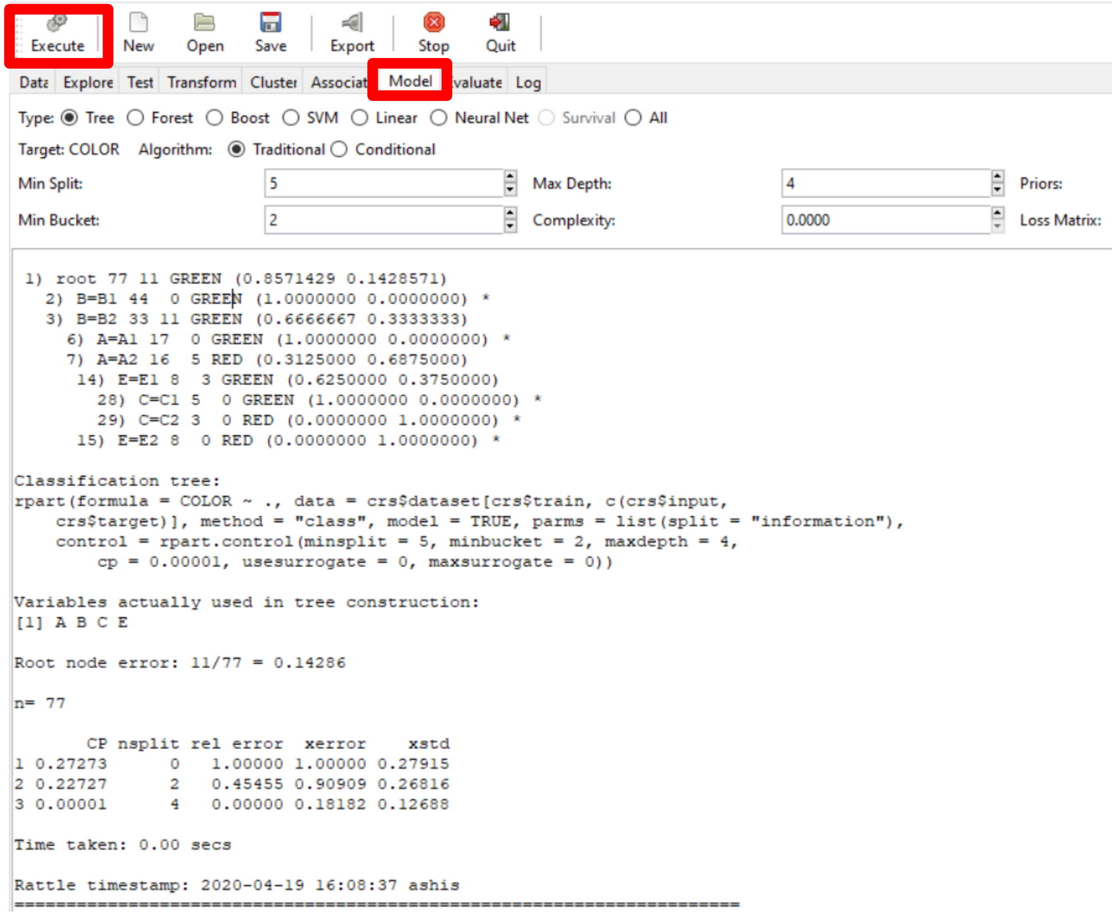


Source: Rattle GUI / Togaware

Decision_Tree_Ex.csv

Running Decision Tree

Output slightly different in R version 3.6.3




```
 1) root 77 11 GREEN (0.8571429 0.1428571)
   2) B=B1 44  0 GREEN (1.0000000 0.0000000) *
   3) B=B2 33 11 GREEN (0.6666667 0.3333333)
     6) A=A1 17  0 GREEN (1.0000000 0.0000000) *
     7) A=A2 16  5 RED (0.3125000 0.6875000)
      14) E=E1 8  3 GREEN (0.6250000 0.3750000)
        28) C=C1 5  0 GREEN (1.0000000 0.0000000) *
        29) C=C2 3  0 RED (0.0000000 1.0000000) *
      15) E=E2 8  0 RED (0.0000000 1.0000000) *

Classification tree:
rpart(formula = COLOR ~ ., data = crs$dataset[crs$train, c(crs$input,
    crs$target)], method = "class", model = TRUE, parms = list(split = "information"),
    control = rpart.control(minsplit = 5, minbucket = 2, maxdepth = 4,
        cp = 0.00001, usesurrogate = 0, maxsurrogate = 0))

Variables actually used in tree construction:
[1] A B C E

Root node error: 11/77 = 0.14286

n= 77

       CP nsplit rel error  xerror    xstd
1 0.27273      0   1.00000 1.00000 0.27915
2 0.22727      2   0.45455 0.90909 0.26816
3 0.00001      4   0.00000 0.18182 0.12688

Time taken: 0.00 secs

Rattle timestamp: 2020-04-19 16:08:37 ashis
======================================================================
```

Source: Rattle GUI / Togaware

Complexity Table

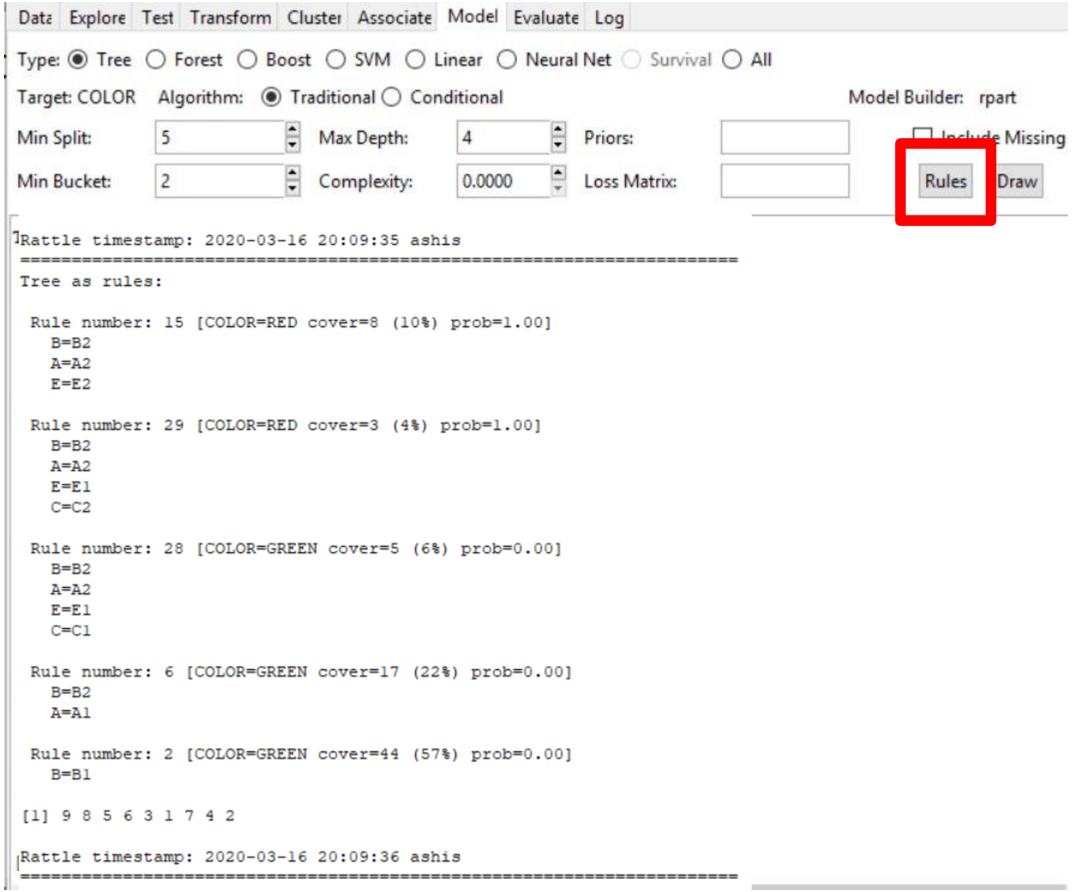
Output slightly different in R version 3.6.3

Root node error: 11/77 = 0.14286

n= 77

	CP	Nsplit	Rel error	Xerror	Xstd
1	0.27273	0	1.0000	1.0000	0.27915
2	0.22727	2	0.4545	0.90909	0.26816
3	0.00001	4	0.0000	0.18182	0.12688

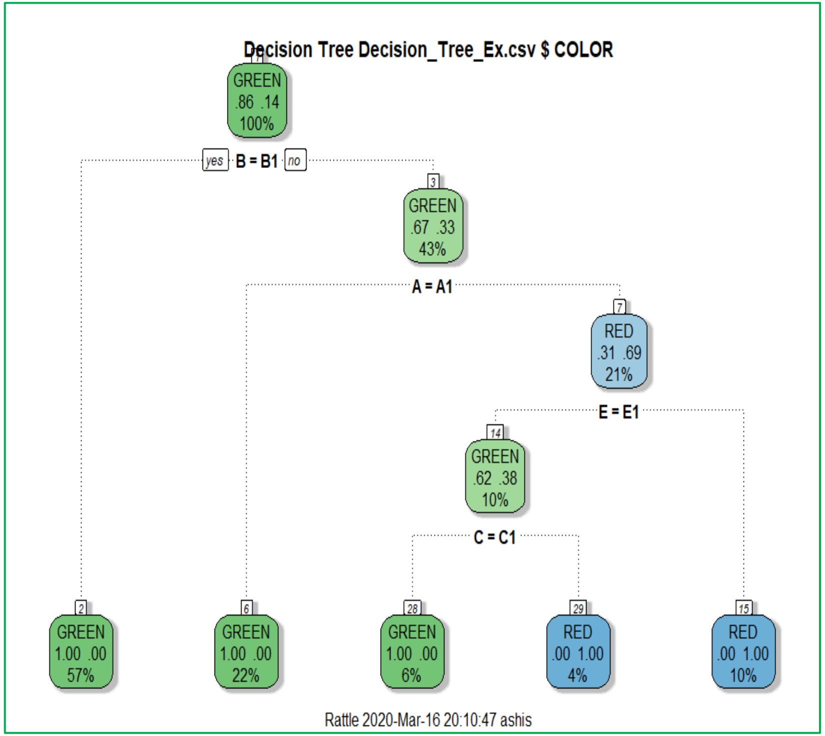
Classification Rules – Post Model-fitting

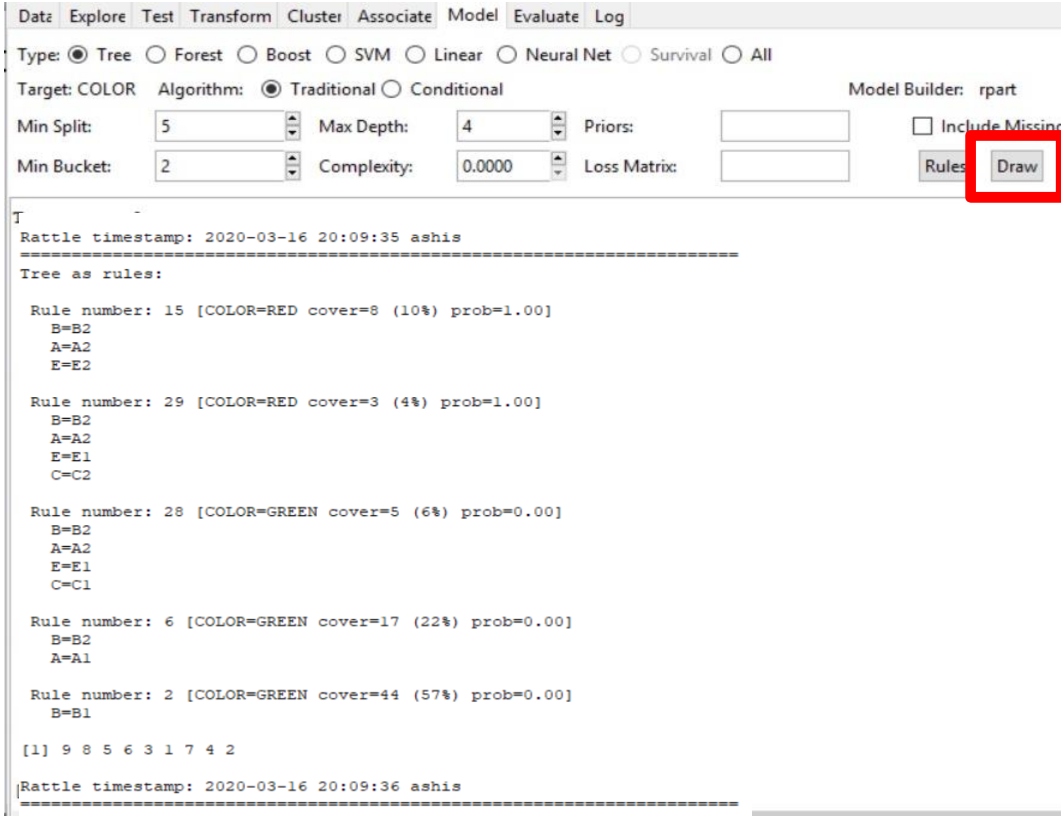


Source: Rattle GUI / Togaware

Classification Tree – Tree

Output slightly different in R version 3.6.3





Source: Rattle GUI / Togaware

Tuning Parameters

Method

Minsplit – minimum observations that must exist at a node before it is split

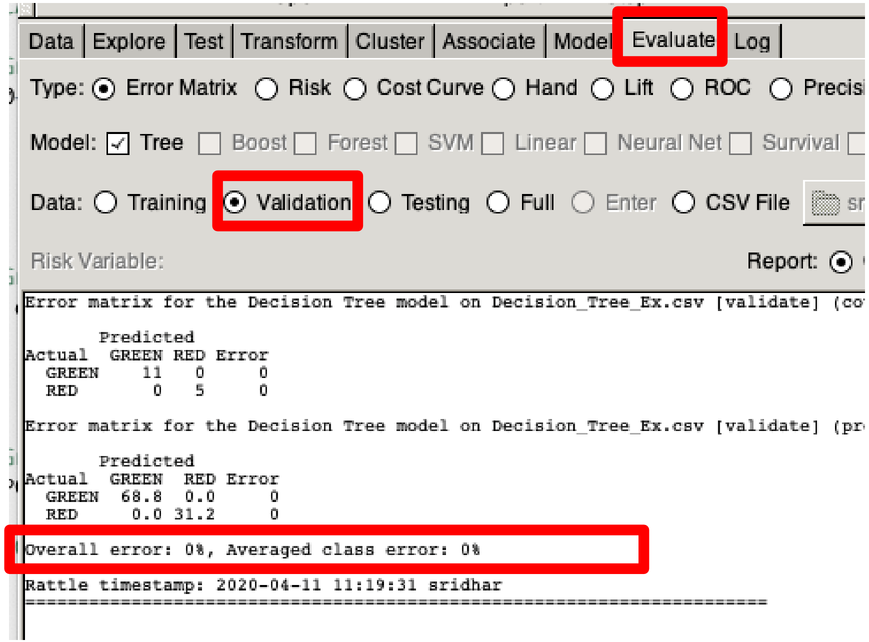
Minbucketsize – minimum number of observations at a leaf node

Max_Depth – maximum depth of the tree

CP - controls the complexity of the tree

Output different in R version 3.6.3

Classification Tree – Stop at Depth = 2



Source: Rattle GUI / Togaware

What If the Target Variable is Numeric Instead of a Class Category ?

Whenever, the target variable is numeric, we use Regression Tree in place of Classification Tree.

We discuss an example of Regression Tree for predicting the price of used cars on the Toyota dataset.

Regression Tree – Toyota Car

Objective – Predict the price of used car based on the its characteristics

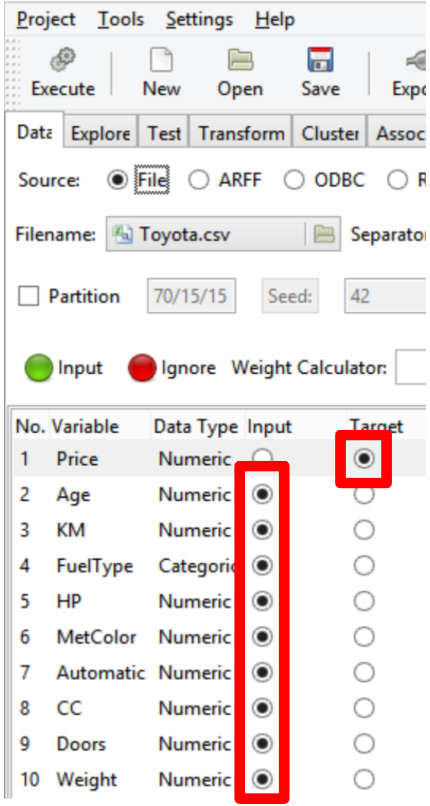
Variable	Description
Price	Offer price in euros
Age	Age in months as of August 2004
Kilometers	Accumulated kilometers on odometer
Fuel type	Fuel type (*Petrol*, *Diesel*, *CNG*)
Horse Power	Horsepower
Metallic	Metallic color? (Yes = 1, No = 0)
Automatic	Automatic (Yes = 1, No = 0)
CC	Cylinder volume in cubic centimeters
Doors	Number of doors
Weight	Weight in kilograms

Top five rows

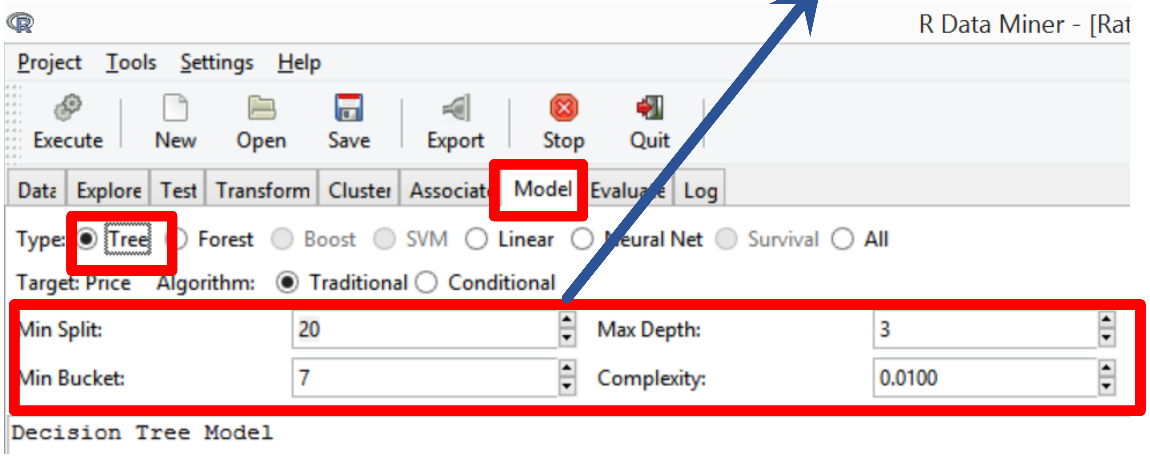
Price	Age	KM	FuelType	HP	MetColor	Automatic	CC	Doors	Weight
13500	23	46986	Diesel	90	1	0	2000	3	1165
13750	23	72937	Diesel	90	1	0	2000	3	1165
13950	24	41711	Diesel	90	1	0	2000	3	1165
14950	26	48000	Diesel	90	0	0	2000	3	1165
13750	30	38500	Diesel	90	0	0	2000	3	1170

Download from https://www.biz.uiowa.edu/faculty/jledolter/datamining/datatext.html

Regression Tree to Predict the “Price” of Used Car



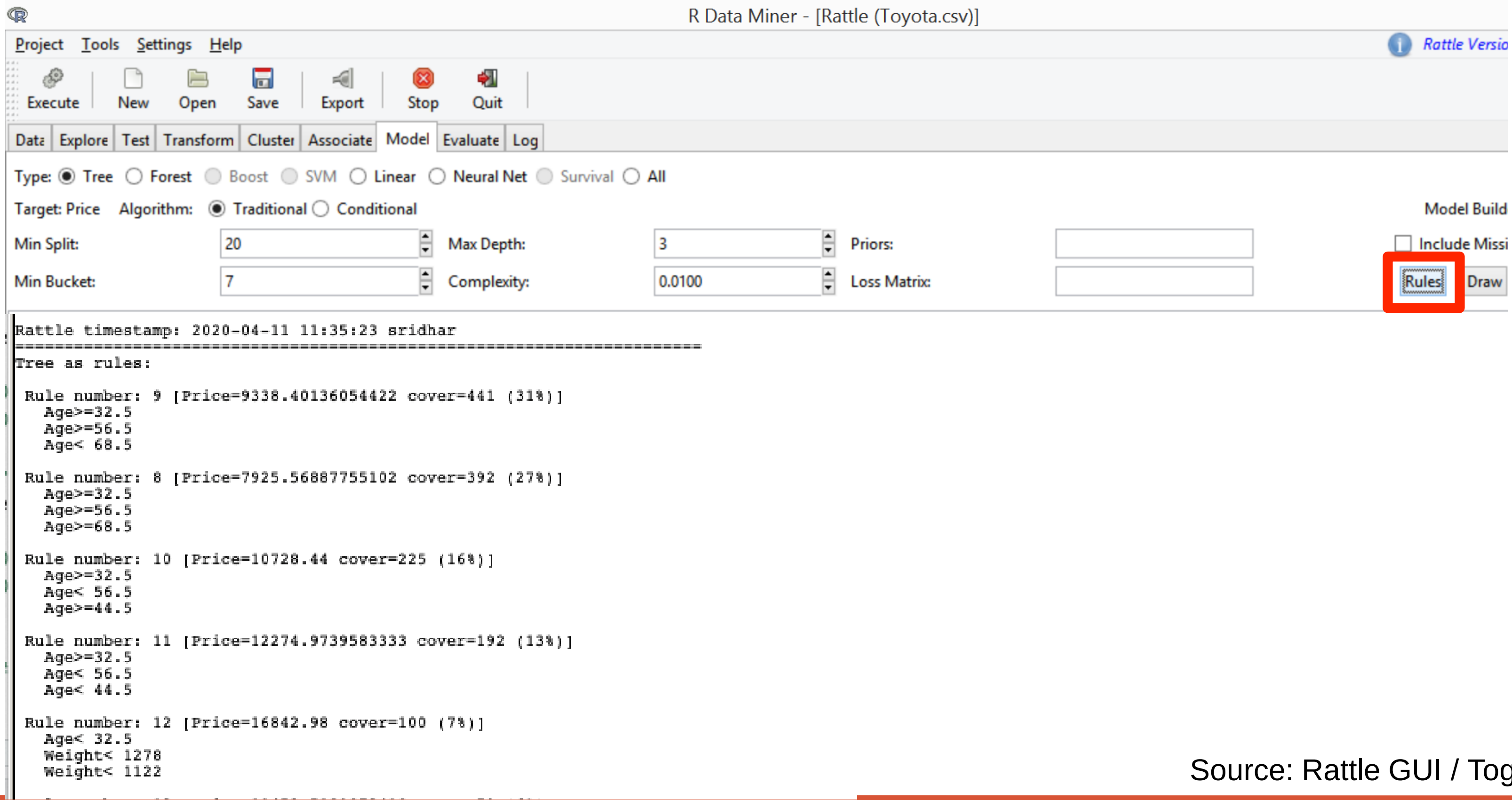
These parameters can be tuned for better fitting



Source: Rattle GUI / Togaware;
Download data from https://www.biz.uiowa.edu/faculty/jledolter/datamining/datatext.html

Output different in R version 3.6.3

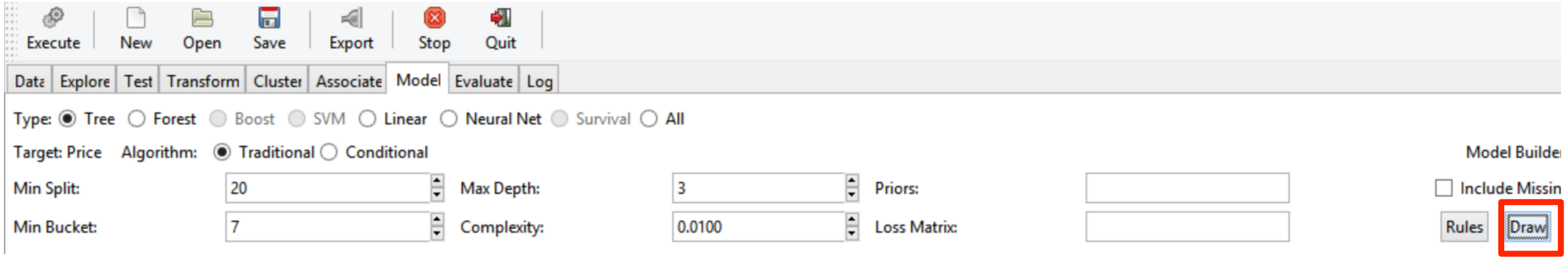
Regression Tree – Rules (Post Model-Fitting)

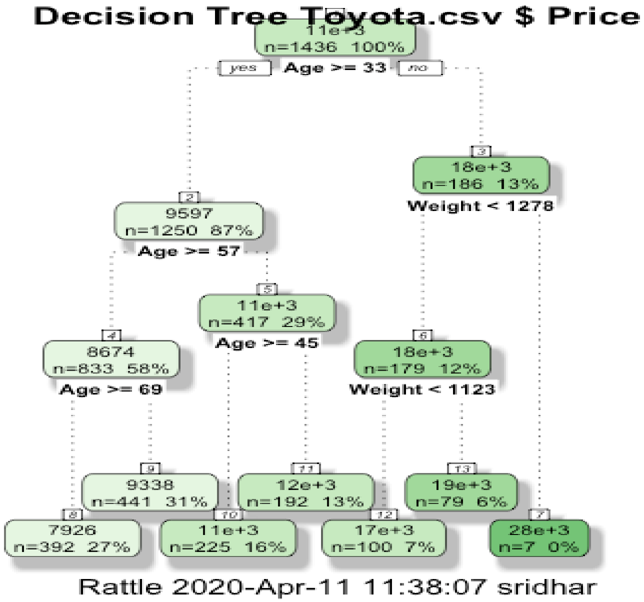


Source: Rattle GUI / Togaware

Output different in R version 3.6.3

Regression Tree

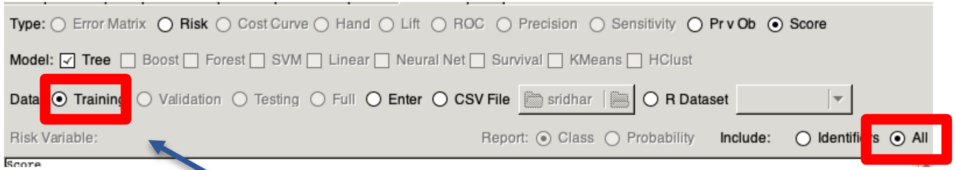


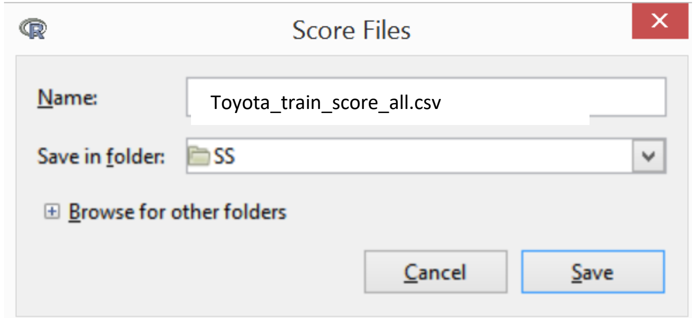


Source: Rattle GUI / Togaware

Output different in R version 3.6.3

Regression Tree – Evaluation (Post Model-fitting): Predicted Price





We kept 100% data for training

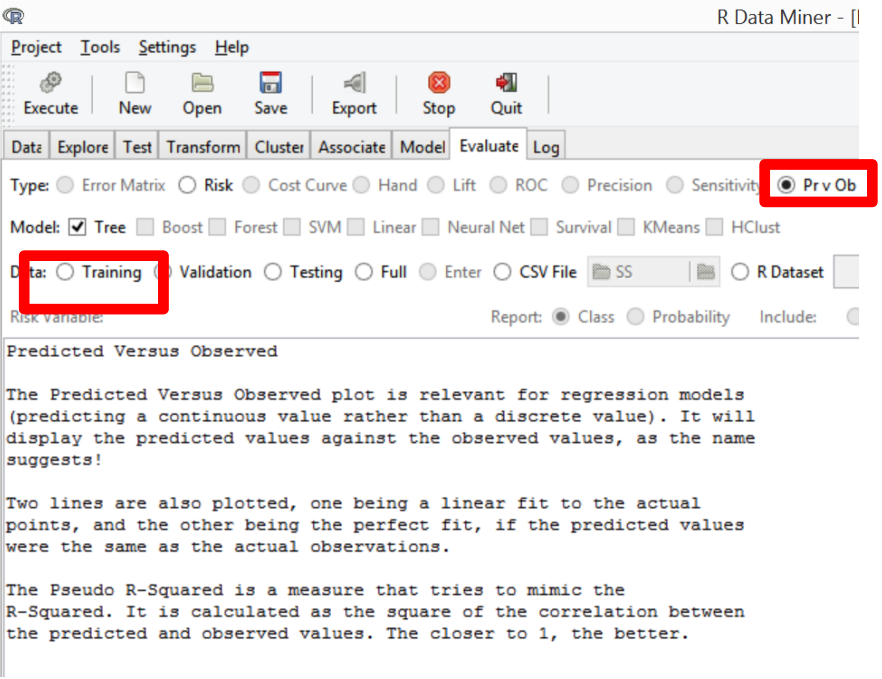
Predicted price

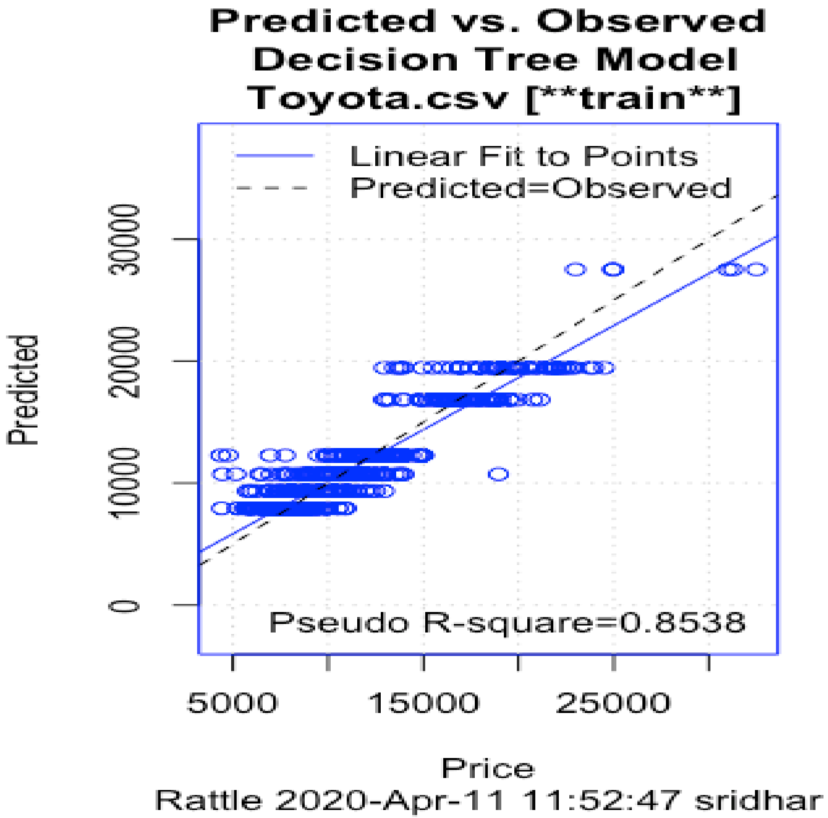
Price	Age	KM	FuelType	HP	MetColor	Automatic	CC	Doors	Weight	rpart
22750	30	34000	Petrol	192	1	0	1800	3	1185	19453
17950	24	21716	Petrol	110	1	0	1600	3	1105	16843
16750	24	25563	Petrol	110	0	0	1600	3	1065	16843
16950	30	64359	Petrol	110	1	0	1600	3	1105	16843
15950	30	67660	Petrol	110	1	0	1600	3	1105	16843
13690	29.4	75648.5	Petrol	77.2	0.8	0	1520	3	1065	15799

Source: Rattle GUI / Togaware

Output slightly different in R version 3.6.3

Predicted vs Observed





Source: Rattle GUI / Togaware

From Trees to Forest

Randomly select data

Randomly select variables on which to split

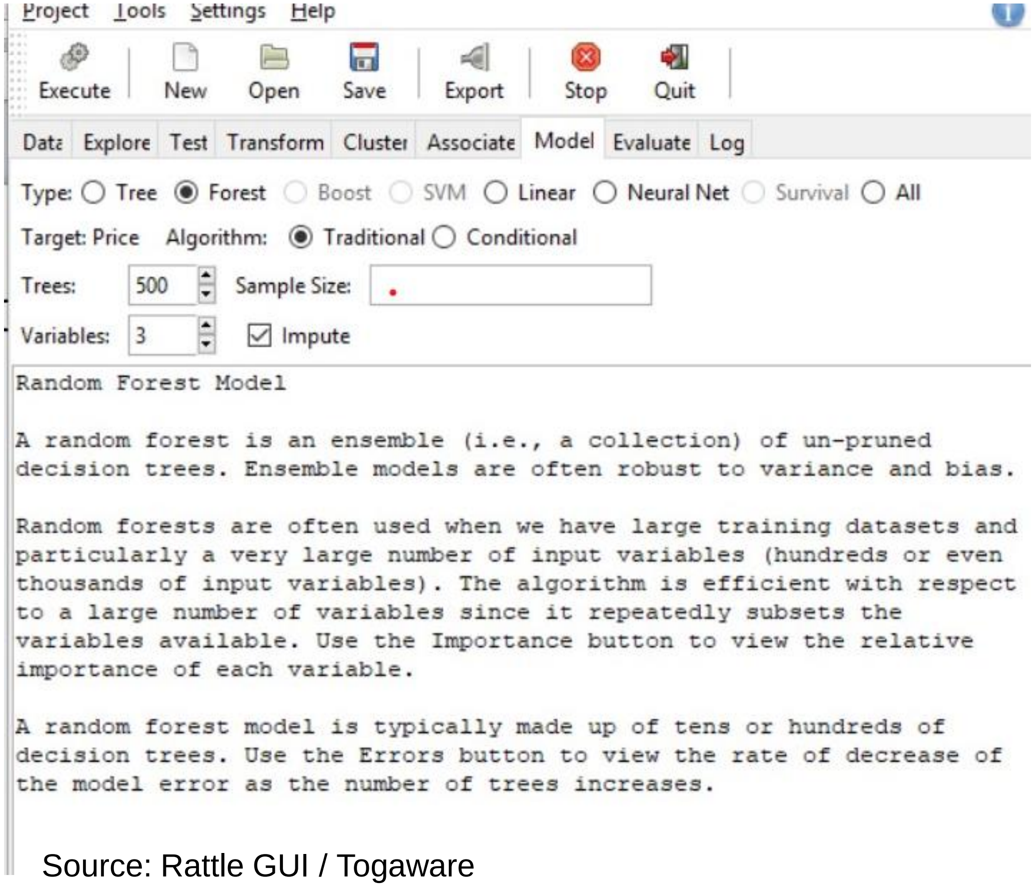
Grow many trees

Use a “ensemble” approach to predict

Robust and improves predictability

Output different in R version 3.6.3

Random Forest Application Example



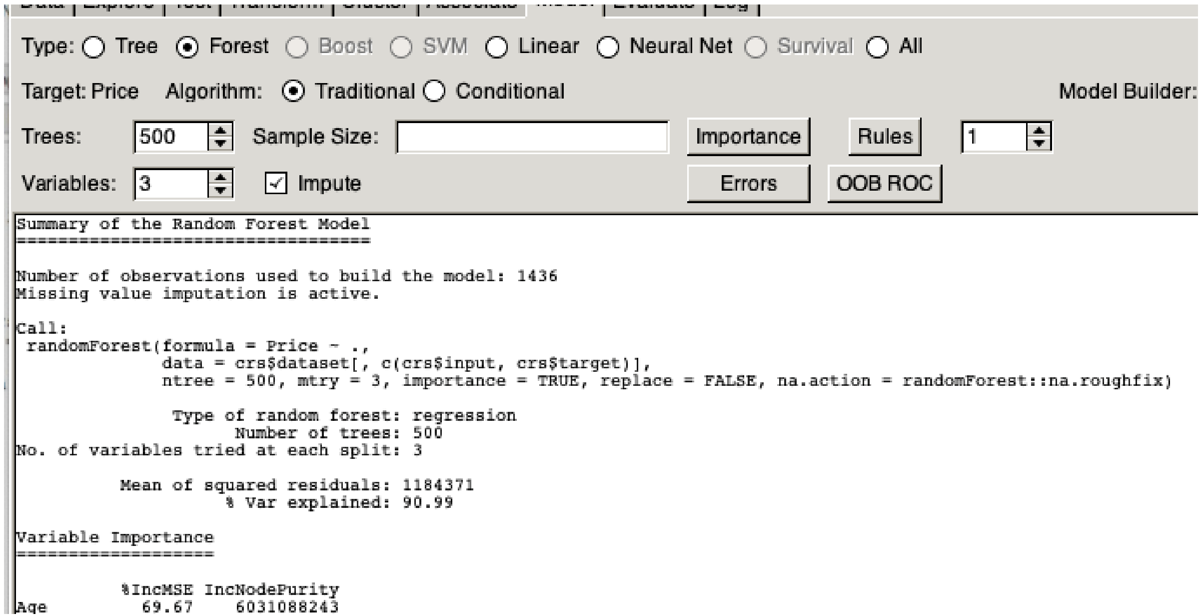
Very simple options:

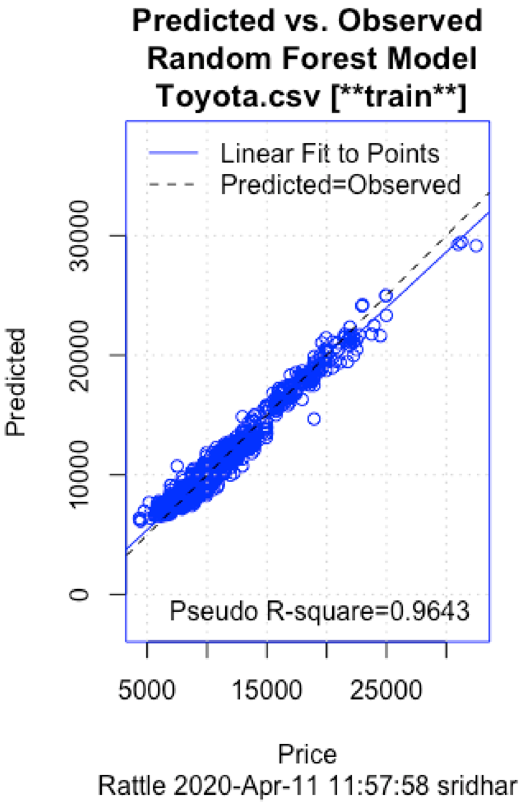
- Number of Trees
- Variables to select at split

Source: Rattle GUI / Togaware

Output slightly different in latest version.

Compare With the Decision Tree





Source: Rattle GUI / Togaware

Output different in R version 3.6.3

Importance of Variables

```
Variables: 3    [✓] Impute

Summary of the Random Forest Model
==================================

Number of observations used to build the model:
Missing value imputation is active.

Call:
 randomForest(formula = Price ~ .,
              data = crs$dataset[, c(crs$input,
              ntree = 500, mtry = 3, importance

               Type of random forest: regression
                     Number of trees: 500
No. of variables tried at each split: 3

          Mean of squared residuals: 1184371
                    % Var explained: 90.99

Variable Importance
===================

          %IncMSE IncNodePurity
Age         69.67    6031088243
Weight      29.87    1622266817
KM          26.00    2373253314
HP          21.08     689671295
CC          20.36     386400930
Doors       14.06      93706341
FuelType     7.26     110407155
Automatic    6.74      18151149
MetColor     4.07      49019327

Time taken: 1.44 secs

Rattle timestamp: 2020-04-11 11:56:10 sridhar
==================================================
```

Source: Rattle GUI / Togaware

Summary

Decision trees are intuitive

Overfitting and variable selection issues

Greedy method and does not question whether the improvement is statistically significant

Summary

Predictive power might be poor

Conditional tree

Random Forest

Further Readings

Breiman, L., Friedman, J.H., Olshen, R.A. & Stone, & C.J. (1984). *Classification and Regression Trees*. Taylor and Francis.

Buhlmann, P. (2010) [Remembrance of Leo Breiman](). The *Annals of Applied Statistics,* 4(4).

Exercise – Part 1

Source- DAAG package library R
Original source - http://archive.ics.uci.edu/ml/machine-learning-databases/spambase/

Use Classification Tree to make a Spam Filter

1) Make a spam filter on the data from **spam.csv** file. Here are variable descriptions

crl.tot total length of words in capitals

dollar number of occurrences of the \$ symbol

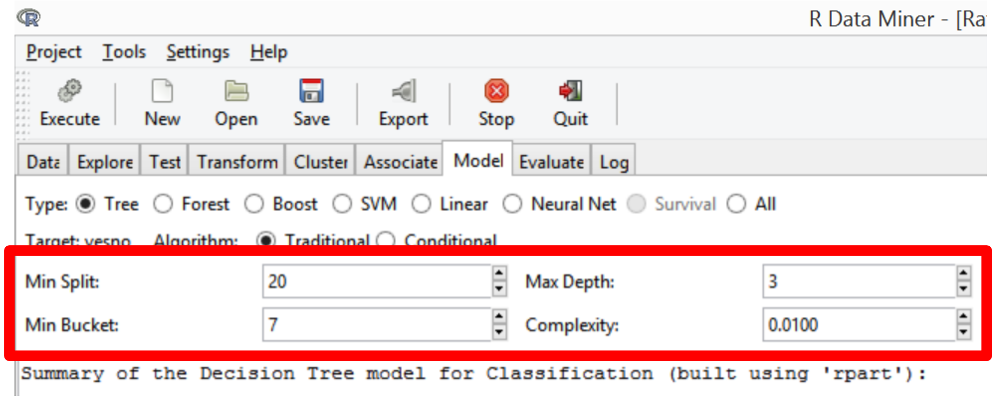
bang number of occurrences of the ! symbol

money number of occurrences of the word 'money'

n000 number of occurrences of the string '000'

make number of occurrences of the word 'make'

yesno outcome variable, a factor with levels n not spam, y spam



Hint – Run a model with the default values in Rattle and note the training and validation error rates. Then change one of Min Split, Max Depth, Min Bucket, Complexity at a time, keeping other three constants and notice the changes in the training and validation error rates.

Source: Rattle GUI / Togaware

Note: The data is available as spam7 in R. We have saved to a file so that it can be read in Rattle.

Exercise – Part 2

2) Briefly discuss how each of these four values (min split, max. depth, min bucket, and complexity) affect bias and variance . For example, what does increasing min split value do to the bias and variance of the trained model?

crl.tot total length of words in capitals

dollar number of occurrences of the \$ symbol

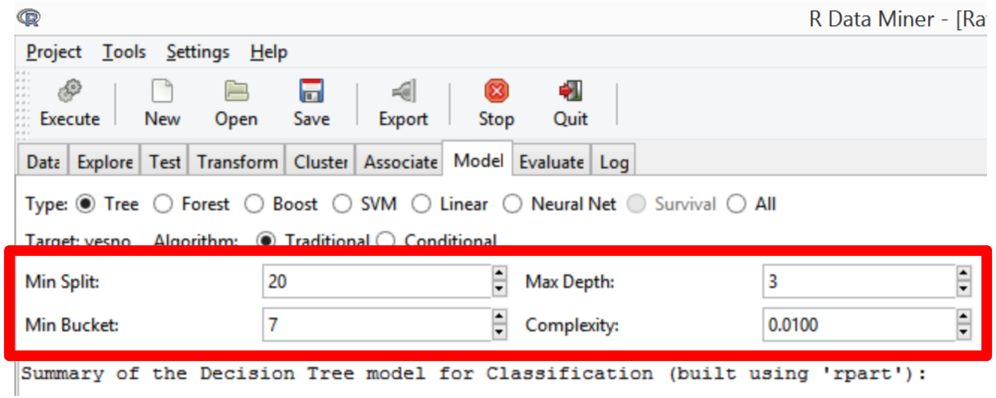
bang number of occurrences of the ! symbol

money number of occurrences of the word 'money'

n000 number of occurrences of the string '000'

make number of occurrences of the word 'make'

yesno outcome variable, a factor with levels n not spam, y spam



Hint – Run a model with the default values in Rattle and note the training and validation error rates. Then change one of these four at a time, keeping other three constants and notice the changes in the training and validation error rates.

Source: Rattle GUI / Togaware

References

"BMI Graph" by Bibi Saint-Pol is licensed under CC BY 3.0

Ledolter, J. (n.d.) Data Mining: ToyotaCorolla.csv. Retrieved May 22, 2019 from University of Iowa.

Rattle GUI / Togaware (https://rattle.togaware.com/)