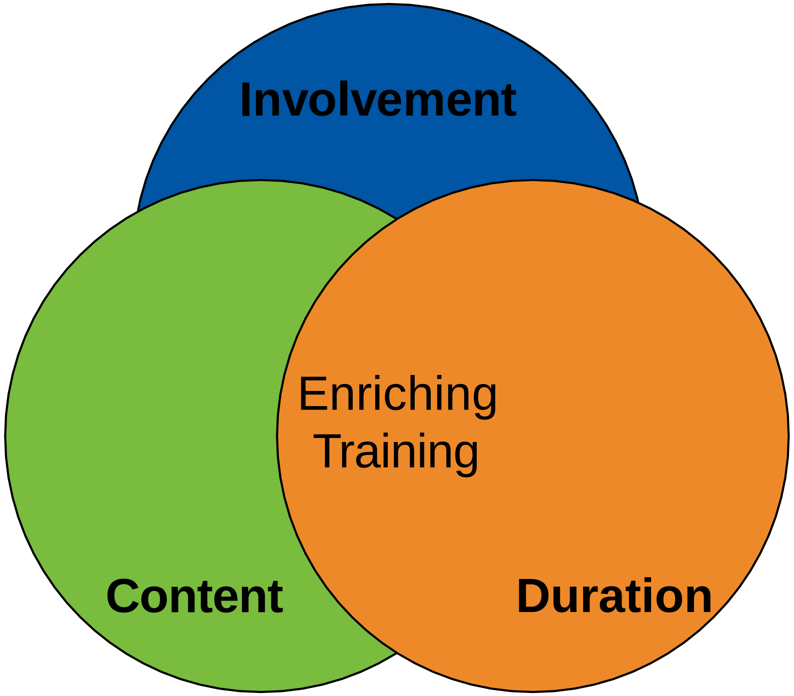
Classification Tree

- Utkarsh Kulshrestha

Earning is in Learning
- Utkarsh Kulshrestha

Enriching training and learning session…



- **Training Checklist**
  - Sitting arrangement F2F
  - Quality over Quantity
  - Everyone to have their own machines for hands-on practice
  - Illuminated and happy glowing training room (no candle light dinner ambience)
  - Anyone wanting to step-out, feel free
  - Feel free to ask for breaks
  - Feel free to ask same question again till you understand
  - Let me know if you want me to skip Practice Exercises in between the session
  - Brief side-talks are okay
  - **I don’t speak to walls, respect each other**

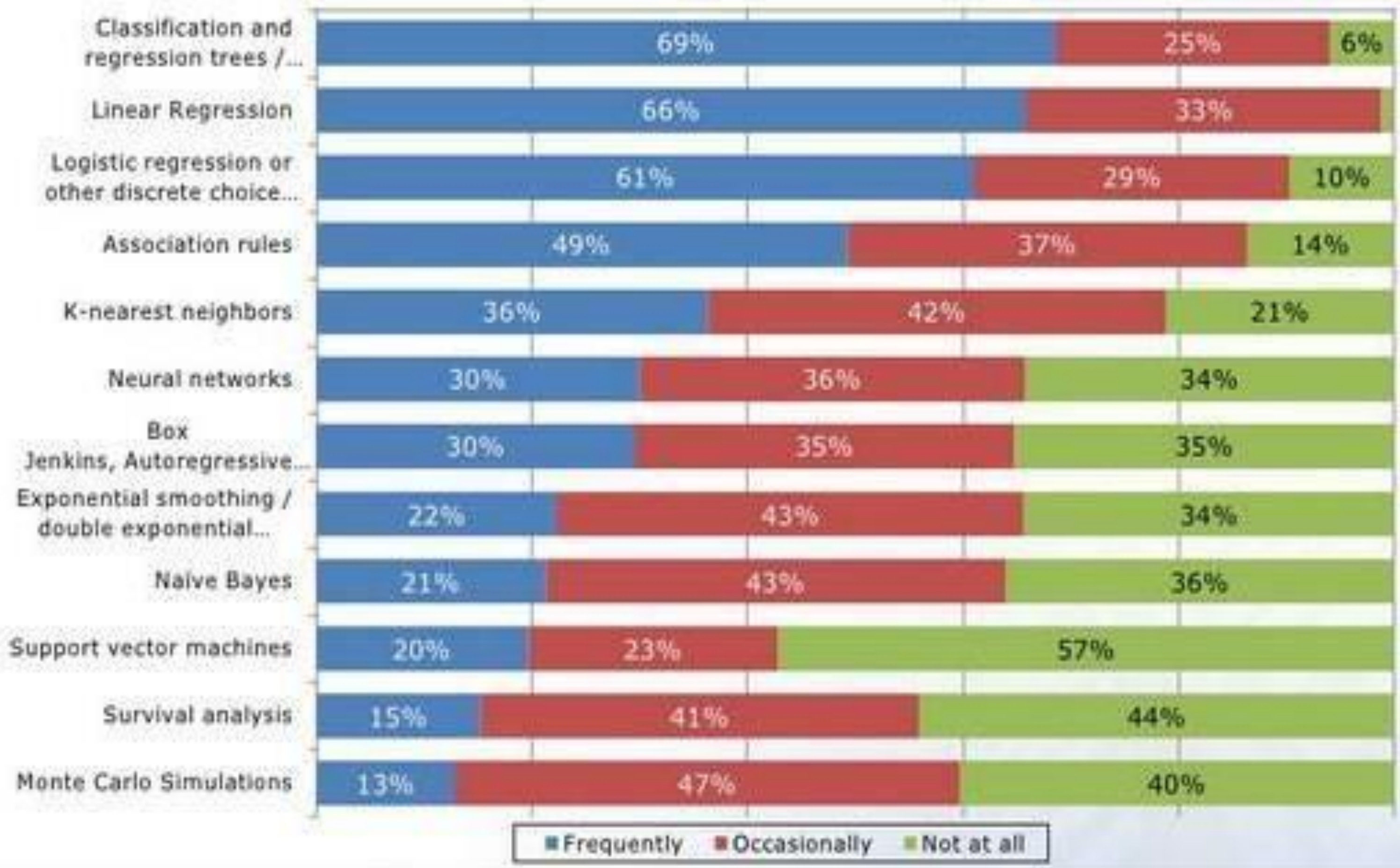
Classification Tree

CART

Learning Objectives

- What is Classification Technique?
- CHAID, CART, C4.5 Intro
- Gini Gain Computation
- Why are Classification Tree algorithms Recursive?
- What is pre-pruning and post-pruning in Classification Tree?
- What is Loss?
- What is Validation? What is Cross-Validation?
- Why you should avoid over-fitting?
- Performance Measure

Analytics that are Actually Used

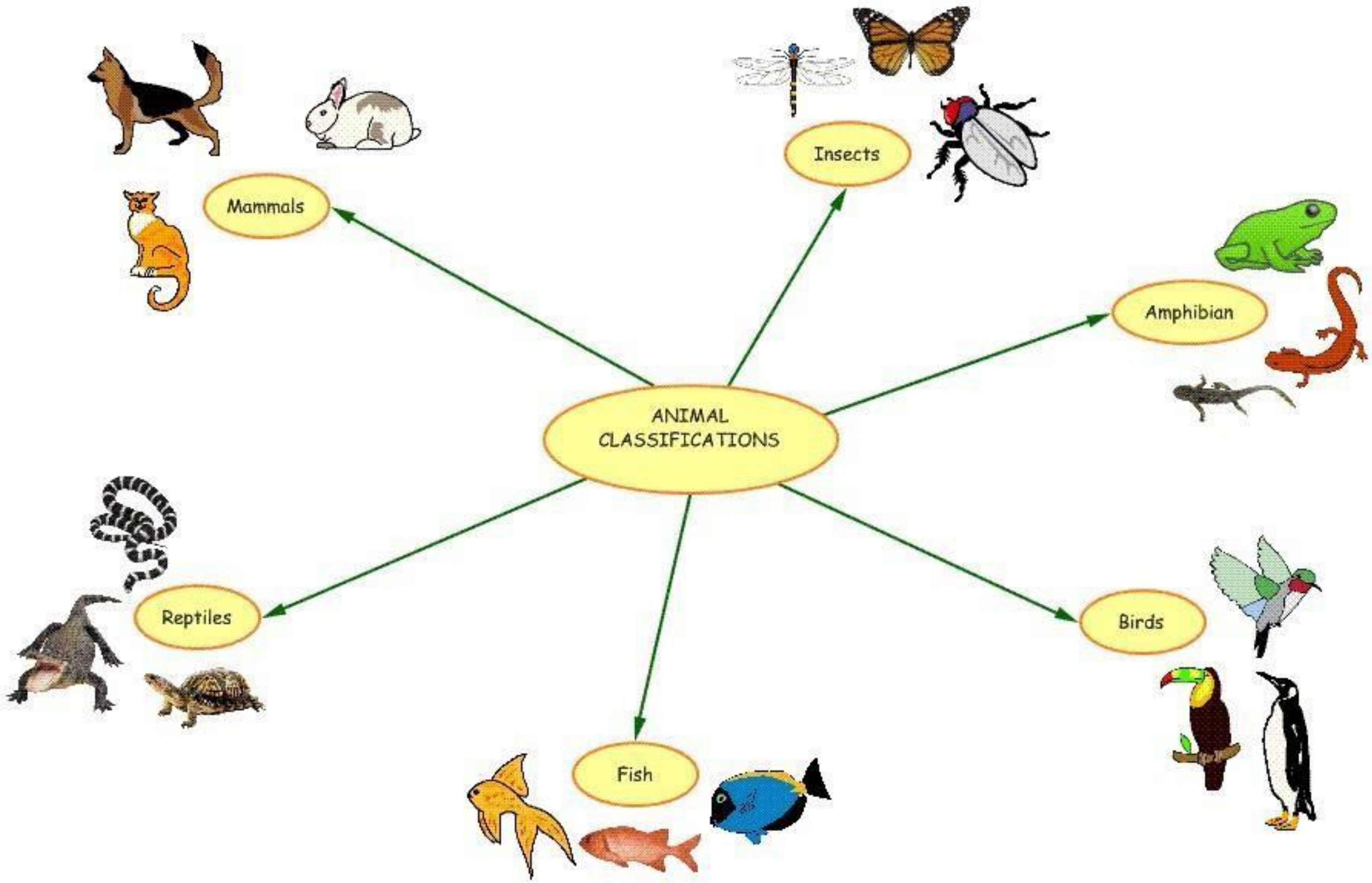


Classification and regression trees / decision trees and Linear Regression are the most popular predictive analytics techniques used.

Source: Ventana Research Predictive Analytics Benchmark Research

What is Classification?

The action or process of classifying something according to shared qualities or characteristics.



Defining Characteristics of each animal classification

- Mammals – Mammals are vertebrates (backboned animals). Mammals are warm-blooded and have hair. Mammals are able to move around using limbs
- Birds – Birds are warm-blooded vertebrates, having a body covered with feathers, forelimbs modified into wings, scaly legs, a beak, and no teeth, and bearing young ones in a hard-shelled egg
- Insects – any of small invertebrate animals which typically have a well defined head, thorax, and abdomen, only three pairs of legs, and typically one or two pair of wings
- Amphibian - any cold-blooded vertebrate that live on land but breed in water
- Reptiles - class of cold-blooded air-breathing vertebrates with completely ossified skeleton and a body usually covered with scales or horny plates
- Fish - A limbless cold-blooded vertebrate animal with gills and fins and living wholly in water

Why Classify?

To Explain (Profile)

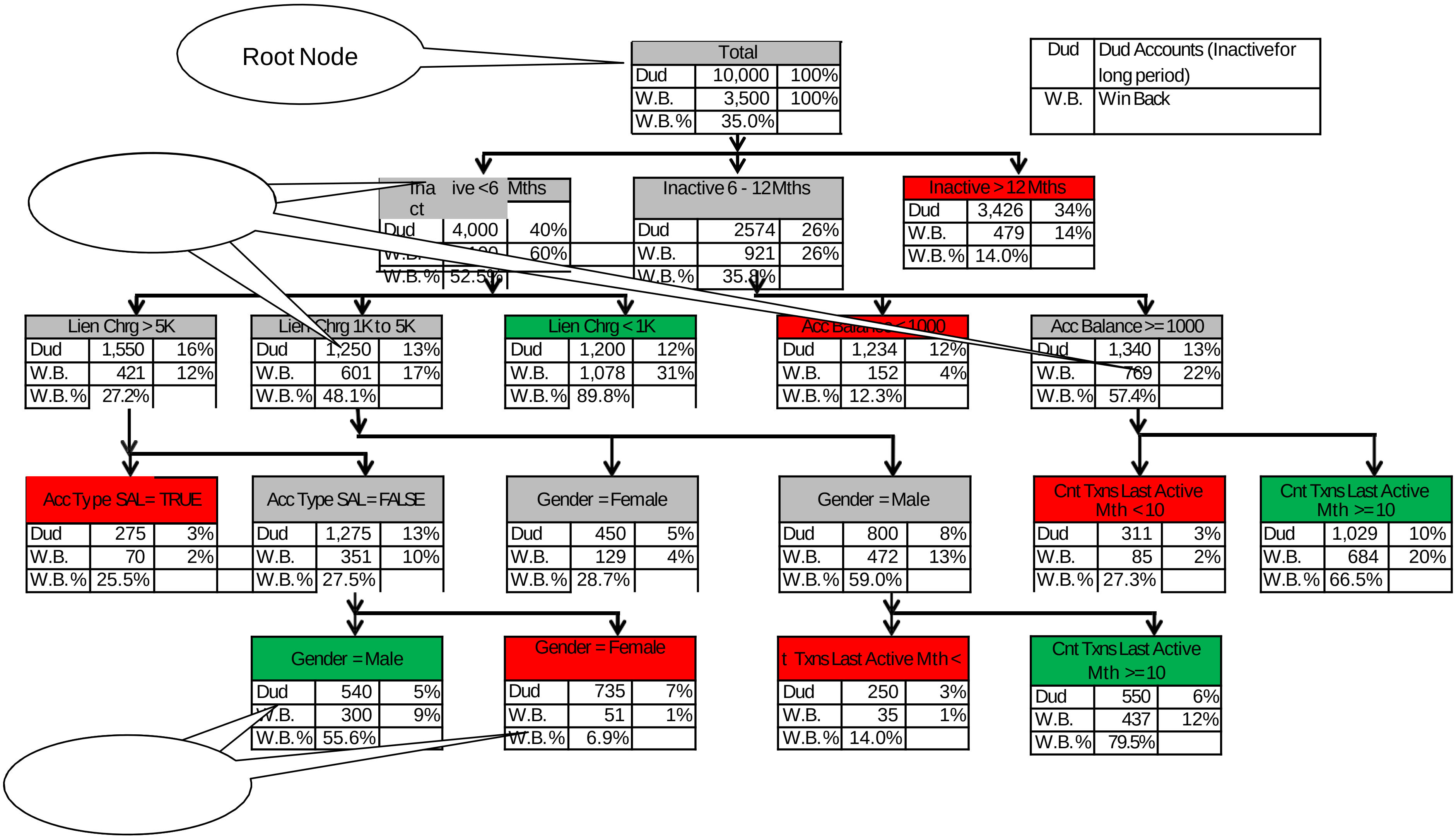
Explaining in the classification world is called Profiling

or

To Predict (Classify)

Predicting the class of new records is called Classifying

Win Back Campaign Classification Analysis



Main issues of classification tree learning

- Choosing the splitting criterion
  - Impurity based criteria
  - Information gain
  - Statistical measures of association
- Binary or multiway splits
  - Multiway split
  - Binary split
- Finding the right sized tree
  - Pre-pruning
  - Post-pruning

Popular Classification Techniques

- **CHAID - CHi-squared Automatic Interaction Detector**. The "*Chi-squared" part of the name arises because the technique* essentially involves automatically constructing many cross-tabs, and working out statistical significance of the proportions. The most significant relationships are used to control the structure of a tree diagram
  - CHAID is a non-binary decision tree; **Recursive Partitioning Algorithm**
  - Continuous variables must be grouped into a finite number of bins to create categories.
- **CLASSIFICATION AND REGRESSION TREES (CART)** are binary decision trees, which split a single variable at each node.
  - The CART algorithm recursively goes though an exhaustive search of all variables and split values to find the optimal splitting rule for each node.
- **C4.5** builds decision trees from a set of training data using the concept of information entropy

CART

CART | Splitting Criteria

- CART uses the Gini Index as measure of impurity
- Gini of a Node

$$GINI(t) = 1 - \sum_j [p(j \mid t)]^2$$

(NOTE: $p(j \mid t)$ is the relative frequency of class j at node t).

- Gini of Split Node is computed as Weighted Avg Gini of each Node at Split Node level

$$GINI_{split} = \sum_{i=1}^{k} \frac{n_i}{n} GINI(i)$$

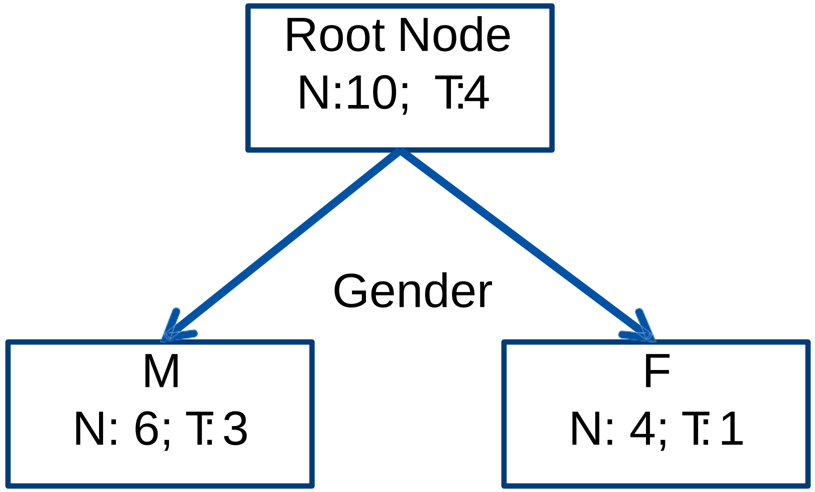
n_i = number of records at child i,
n = Total number of records in parent node

- Gini Gain = Gini(t) – Gini(split)

[www.cs.kent.edu/~jin/DM07/**ClassificationDecisionTree.ppt**](www.cs.kent.edu/~jin/DM07/ClassificationDecisionTree.ppt)

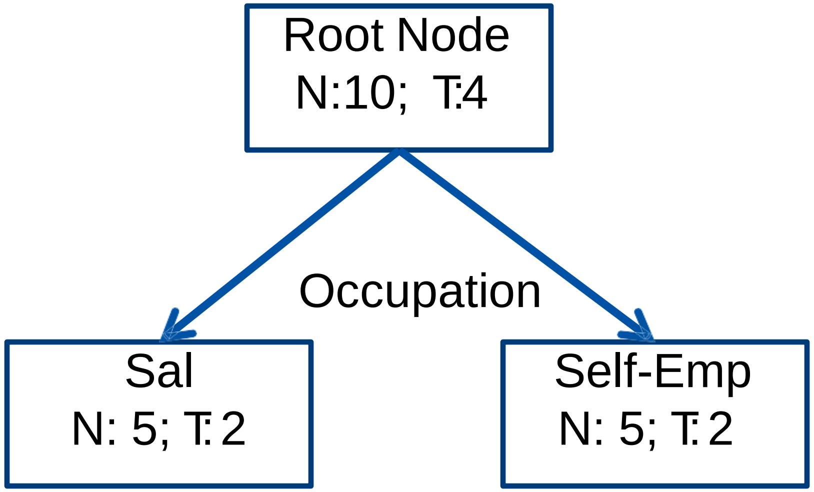
Gini calculations

Cust_ID	Gender	Occupation	Age	Target
1	M	Sal	22	1
2	M	Sal	22	0
3	M	Self-Emp	23	1
4	M	Self-Emp	23	0
5	M	Self-Emp	24	1
6	M	Self-Emp	24	0
7	F	Sal	25	1
8	F	Sal	25	0
9	F	Sal	26	0
10	F	Self-Emp	26	0



Node	Gini Computation Formula	Gini Index
Overall	= 1 - ((4/10)^2 + (6/10)^2)	0.48
Gender = M	= 1 - ((3/6)^2 + (3/6)^2)	0.50
Gender = F	= 1 - ((1/4)^2 + (3/4)^2)	0.375
Gender	= (6/10) * 0.5 + (4/10) * 0.375	0.45
Gini Gain	= Gini (Overall) – Gini (Gender)	0.03

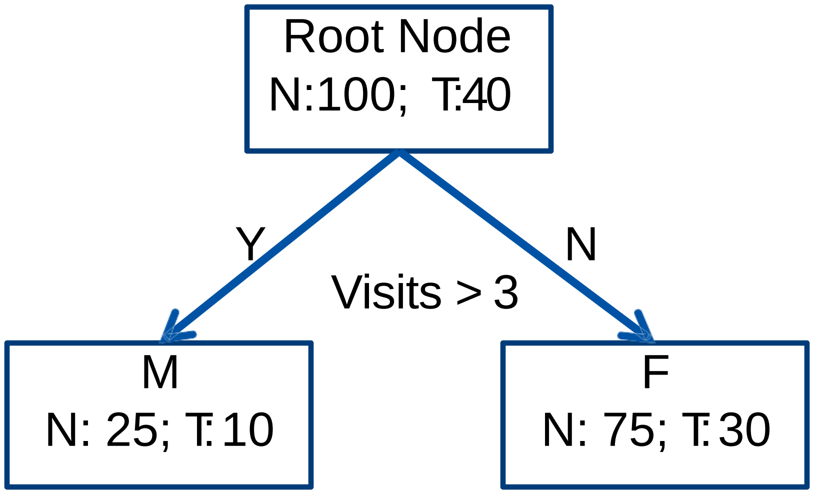
Gini calculations



Node	Gini Computation Formula	Gini Index
Overall	= 1 - ((4/10)^2 + (6/10)^2)	0.48
Occ = Sal	= 1 - ((2/5)^2 + (3/5)^2)	0.48
Occ = Self-Emp	= 1 - ((2/5)^2 + (3/5)^2)	0.48
Occupation	= (5/10) * 0.48 + (5/10) * 0.48	0.48
Gini Gain	= Gini (Overall) – Gini (Occupation)	0.0

Age	<=22	<=23	<=24	<=25
Gini (Left)	0.5	0.5	0.5	0.5
Gini (Right)	0.47	0.44	0.38	0
Gini Split	0.48	0.47	0.45	0.40
Gini Gain	0.0	0.01	0.03	0.08

Exercise… Compute Gini Gain



Sampling…

Creating Development and Validation Sample

```
##dummy_df = pd.read_csv("/home/utkarsh/Desktop/bank.csv", na_values =['NA'])

##x_train, x_test, y_train, y_test = train_test_split(x,y,test_size =0.5)
```

Sampling Code

Separate Dev & Val samples are provided as such we will directly import them rather than use sampling code

```
CTDF.dev <- pd.read_csv("datafile/DEV_SAMPLE.csv", sep = ",", header = T)

CTDF.holdout <- pd.read_csv ("datafile/HOLDOUT_SAMPLE.csv", sep = ",", header = T)
```

Decision Tree code to build CART Tree

installing rpart package for CART

 # from sklearn.model_selection import train_test_split

 # from sklearn.tree import DecisionTreeClassifier

import matplotlib.pyplot as plt from sklearn.externals.six
import StringIO from IPython.display import Image

from sklearn.tree import export_graphviz

import pydotplus

\

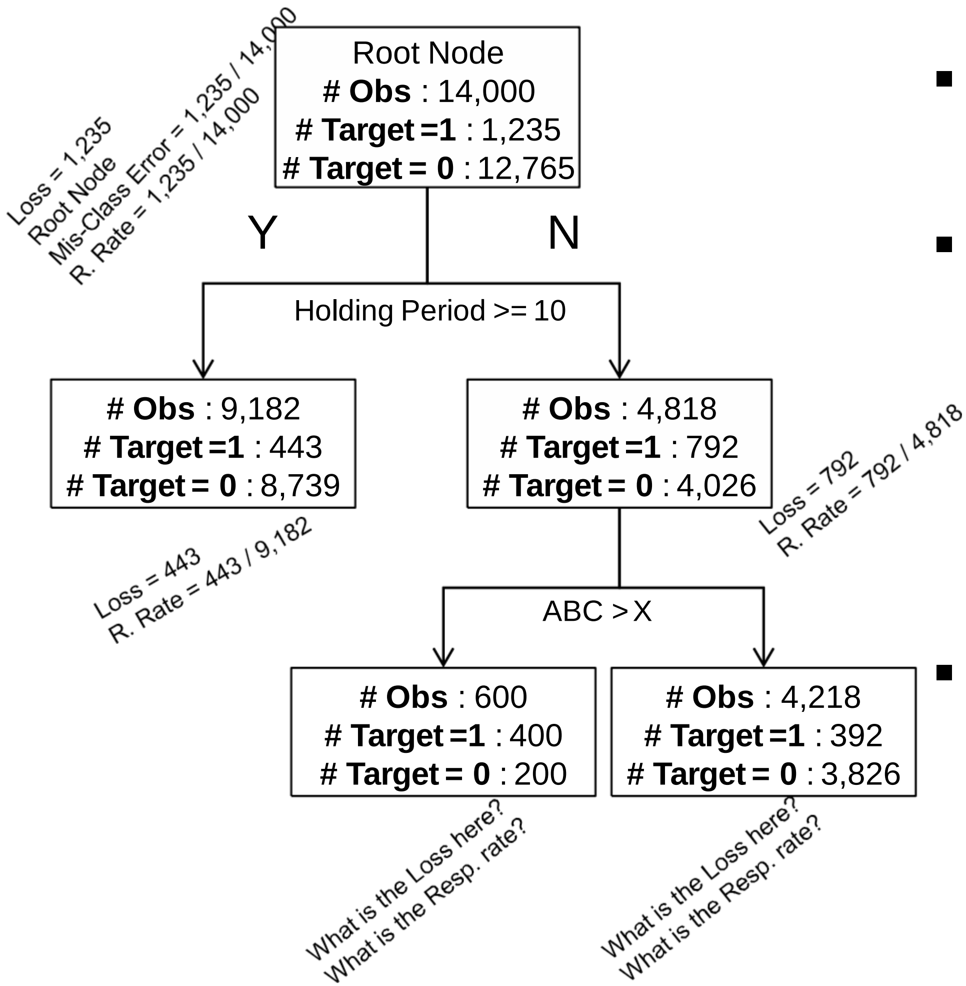
calling the Decision Tree function to build the tree

model_dt = DecisionTreeClassifier(max_depth = 8, criterion =“gini“, min_samples_split = 100, min_sample_leaf = 10)

Decision Tree control arguments

- **Min_samples_split:** the minimum number of observations that must exist in a node in order for a split to be attempted.
- **Min_samples_leaf**: the minimum number of observations in any terminal leaf node. If only one of min_samples_leaf or min_samples_split is specified, the code either sets min_samples_split to min_samples_leaf*3 or min_samples_leaf to min_samples_split/3,as appropriate.
- **max_depth**: The maximum depth of the tree.if NONE then nodes are expanded until all leaves are pure or until all leaves contains less than min_samples_split samples.
- **Criterion**: The function to measure the quality of the split. It can be “gini” for the gini impurity and “entropy” for the information gain.

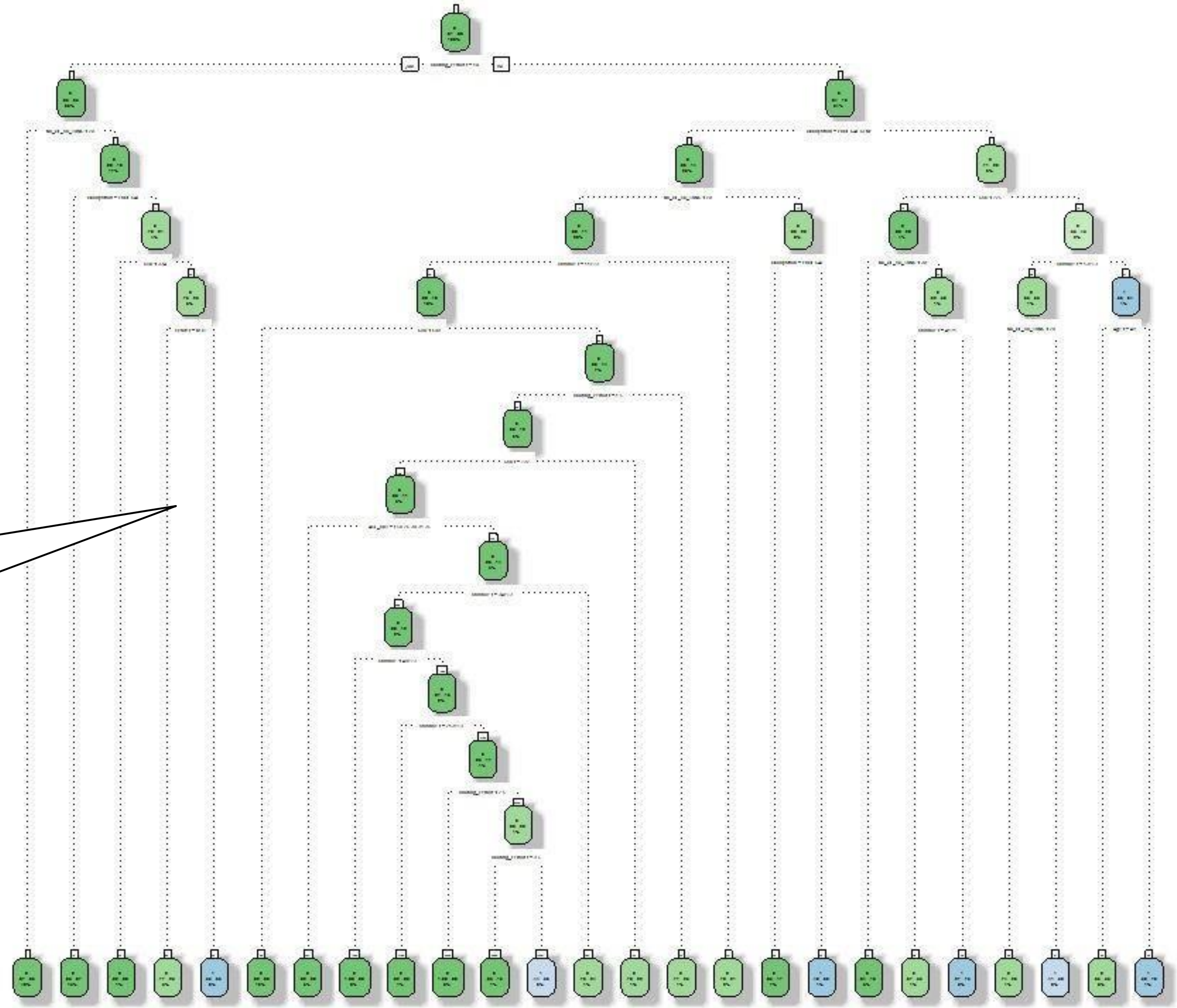
Loss, Mis-Classification Error and Response Rate



- Loss is the number of cases mis-classified in a given node
- Mis-Classification Error is the ratio of total number of cases mis-classified to total number of cases
  - We are interested in mis-classification error for the full tree
- Response Rate is the ratio of number of responders (Target = 1) to the total number of cases
  - We are interested in finding nodes where the response rate is very high

What is the mis-classification error for the above tree?

Plotting the Classification Tree



Let us export the output to PDF format to have a clear view of the tree

Concepts | Greedy Algorithm

Make 31 Paise using any combination of above coins

Optimal solution with few coins : 25 + 5 + 1

What if the 5 paise coin is not there?

Optimal solution with few coins : 10 * 3 + 1

Greedy Algorithm solution: 25 + 1 * 6

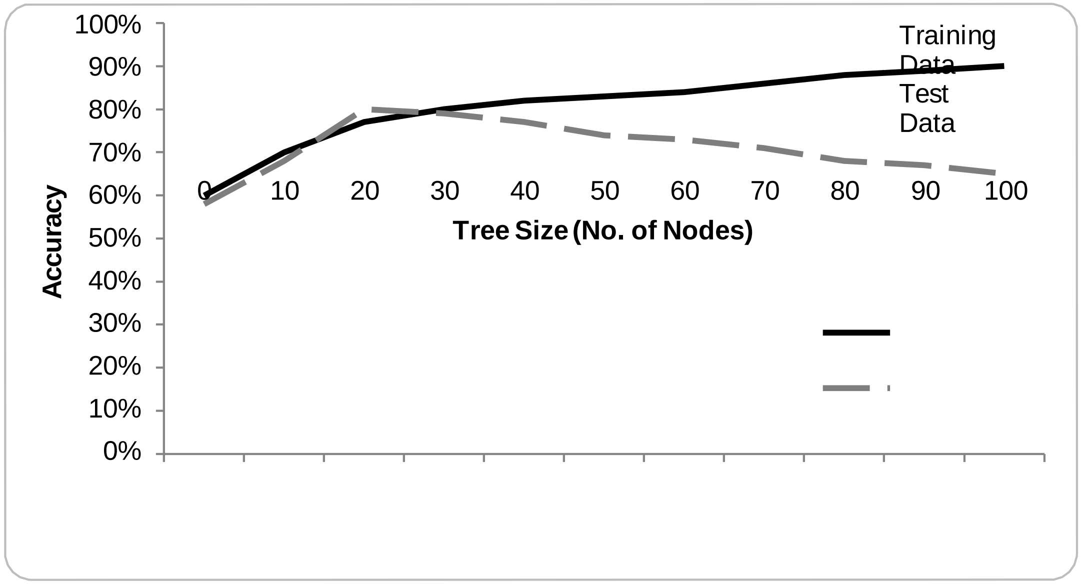
Concepts | Cross Validation

K Fold CV	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
Fold 1	Train	Train	Train	Train	Train	Train	Train	Train	Train	Test
Fold 2	Train	Train	Train	Train	Train	Train	Train	Train	Test	Train
Fold 3	Train	Train	Train	Train	Train	Train	Train	Test	Train	Train
Fold 4	Train	Train	Train	Train	Train	Train	Test	Train	Train	Train
Fold 5	Train	Train	Train	Train	Train	Test	Train	Train	Train	Train
Fold 6	Train	Train	Train	Train	Test	Train	Train	Train	Train	Train
Fold 7	Train	Train	Train	Test	Train	Train	Train	Train	Train	Train
Fold 8	Train	Train	Test	Train	Train	Train	Train	Train	Train	Train
Fold 9	Train	Test	Train	Train	Train	Train	Train	Train	Train	Train
Fold 10	Test	Train	Train	Train	Train	Train	Train	Train	Train	Train

- Cross Validation is part of the CART algorithm
- Method to see how well the model performs to unseen data
- Typically xval parameter for cross-validation is set to 10

Concepts | Over-fitting

- If you grow the tree too long you will run the risk of over-fitting



- Classification model may not work well on unseen data

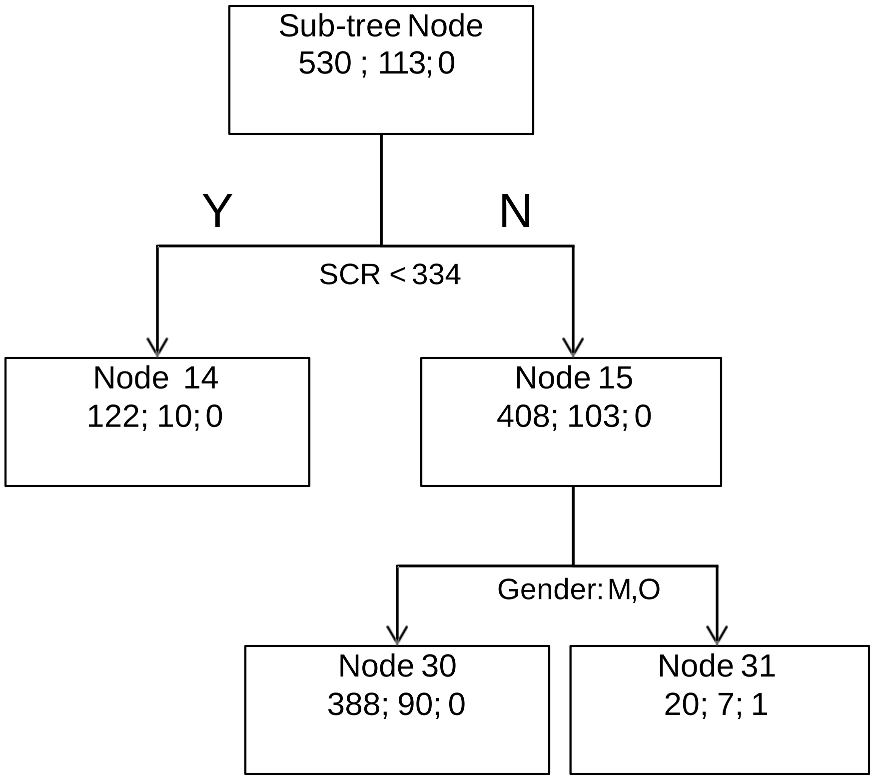
How do we avoid Over-fitting?

Stopping Rule: don't expand a node if the impurity reduction of the best split is below some threshold

Pruning: grow a very large tree and merge back nodes

Concepts | Parsimony Principle & Re-substitution Error

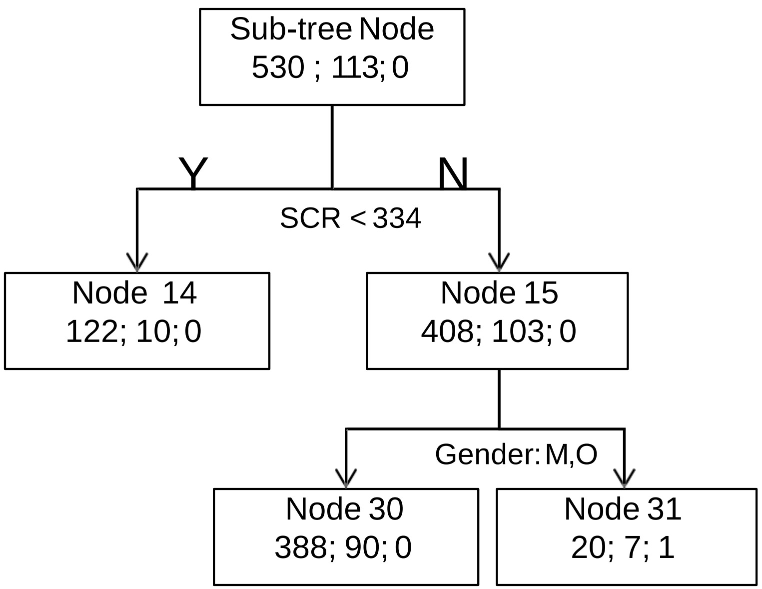
- **Parsimony principle** is basic to all science and tells us to choose the simplest scientific explanation that fits the evidence.
- **Resubstitution Error**: It measures what fraction of the cases in a node is classified incorrectly if we assign every case to the majority class in that node; It always favours large tree
- To counter balance the resubstitution error we need a penalty component that favours smaller tree



Re (prunded) = 113 / 530
Re (leaves) = 107 / 530

Cost Component Pruning

- “cost-complexity” – a measure of avg. error reduced per leaf
- Calculate number of errors for each node if collapsed to leaf
- Compare to errors in leaves, taking into account more nodes used



Re (prunded) + 1 α

= Re (leaves) + 3 α

$113 / 530 + 1\,\alpha = 107/530 + 3\,\alpha$

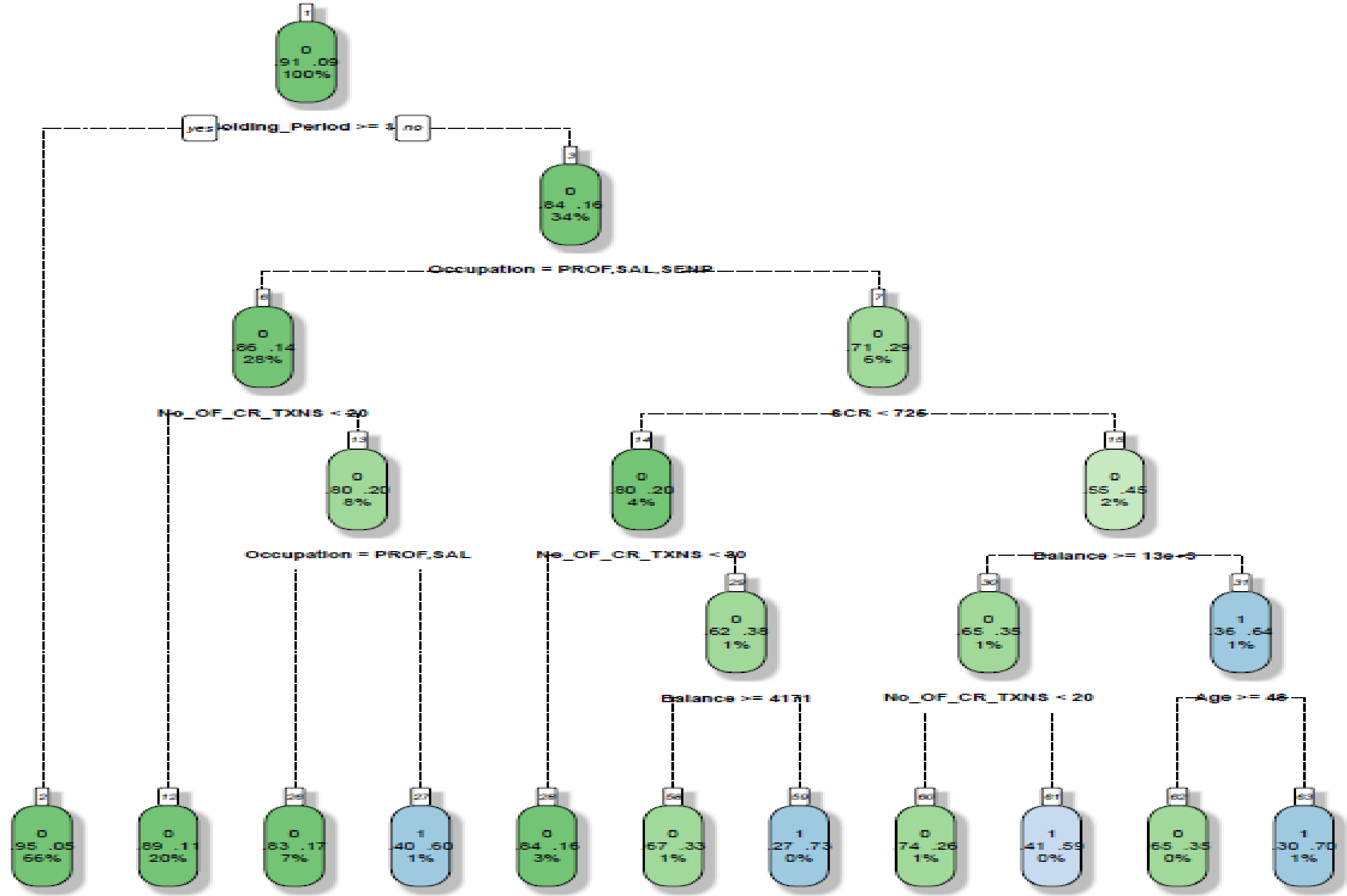
$\alpha = 0.0056$

Pruning

- Pruning is Basically the average cost complexity reduced per leaf in a Decision Tree.
- Generally It's a hit & try method to get the accuracy improved over the depth of tree getting reduced or average number of nodes reduced without over fitting.
- Practically, We creates a Tree structure which is getting refined on certain pre-assumptions for improving the performance and accuracy of a Decision Tree classifier

http://stats.stackexchange.com/questions/92547/r-rpart-cross-validation-and-1-se-rule-why-is-the-column-in-cptable-called-xst
https://stats.stackexchange.com/questions/13471/how-to-choose-the-number-of-splits-in-rpart

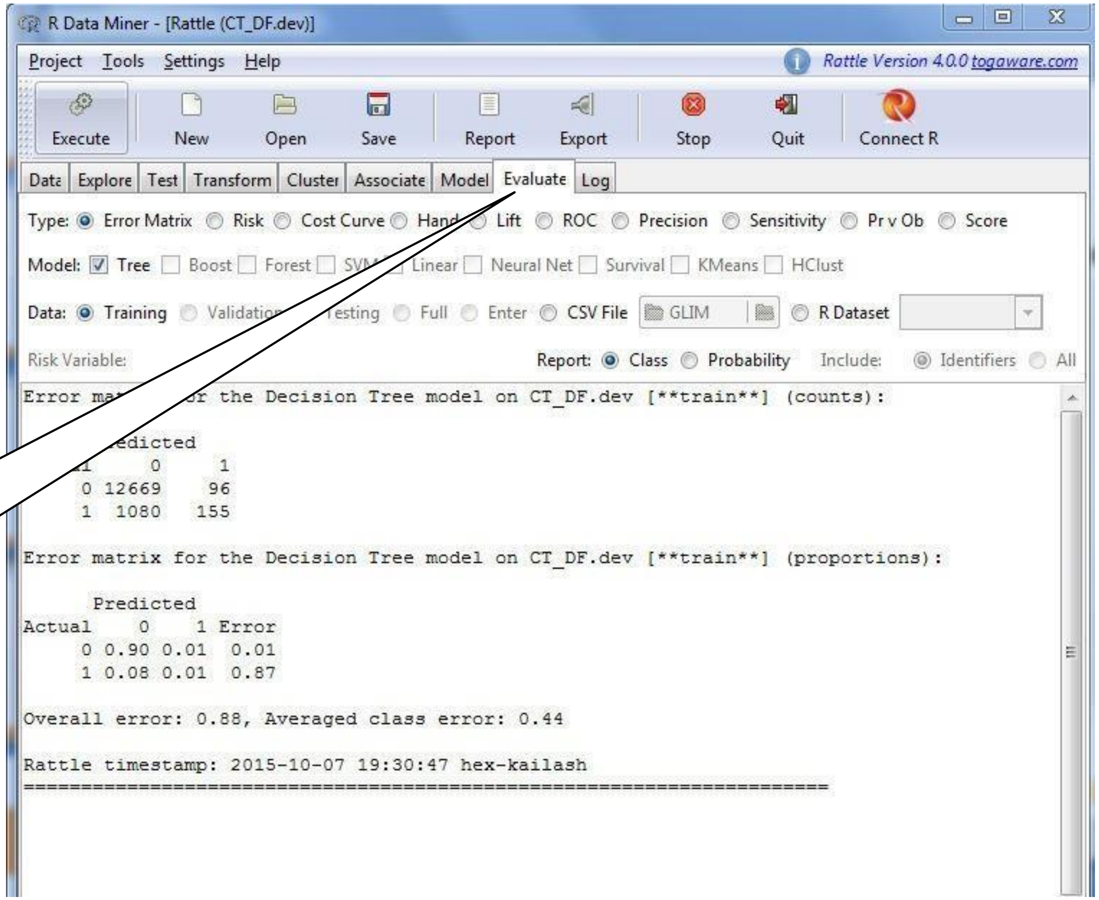
Pruned Classification Tree



Model Evaluation

Various measures to see the model performance

- Error Matrix
- Gini Coefficient
- AUC
- KS
- Lift Chart



Demo of Rattle interface to build model and generate various model evaluation measures

https://www.youtube.com/watch?v=OAl6eAyP-yo

Confusion Matrix… ☺☺☺

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN

Area Under Curve

Classification Matrix		Predicted	
		Y	N
Actual	Y	a	b
	N	c	d

Sensitivity = True Positive Rate

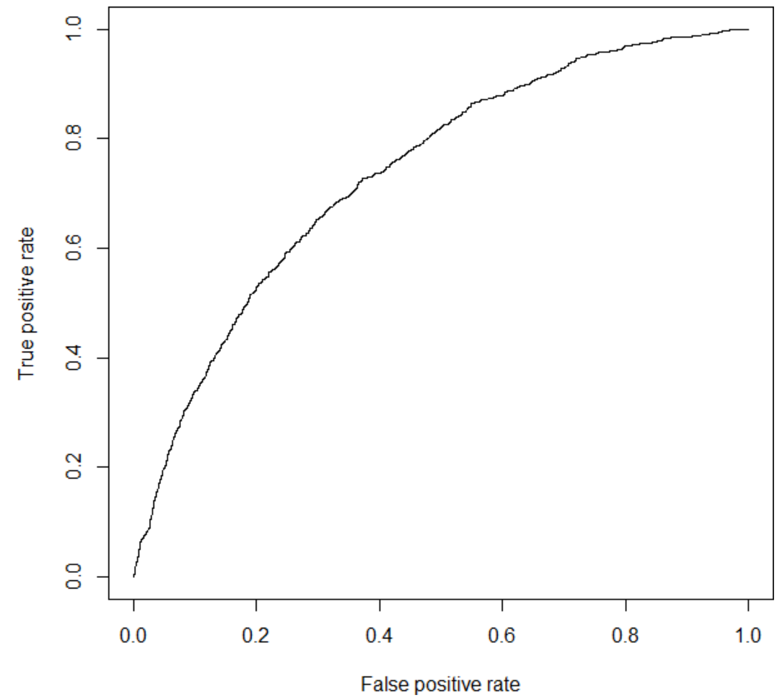
= True Positive / Total Positive

= a / (a + b)

Specificity = True Negative / Total Negative

= d / (c + d)

False Positive Rate = 1 - Specificity



Questions?? … Thankyou

Contact Us

kuls.utkarsh1205@gmail.com