Walmart

Decision tree algorithms – 101

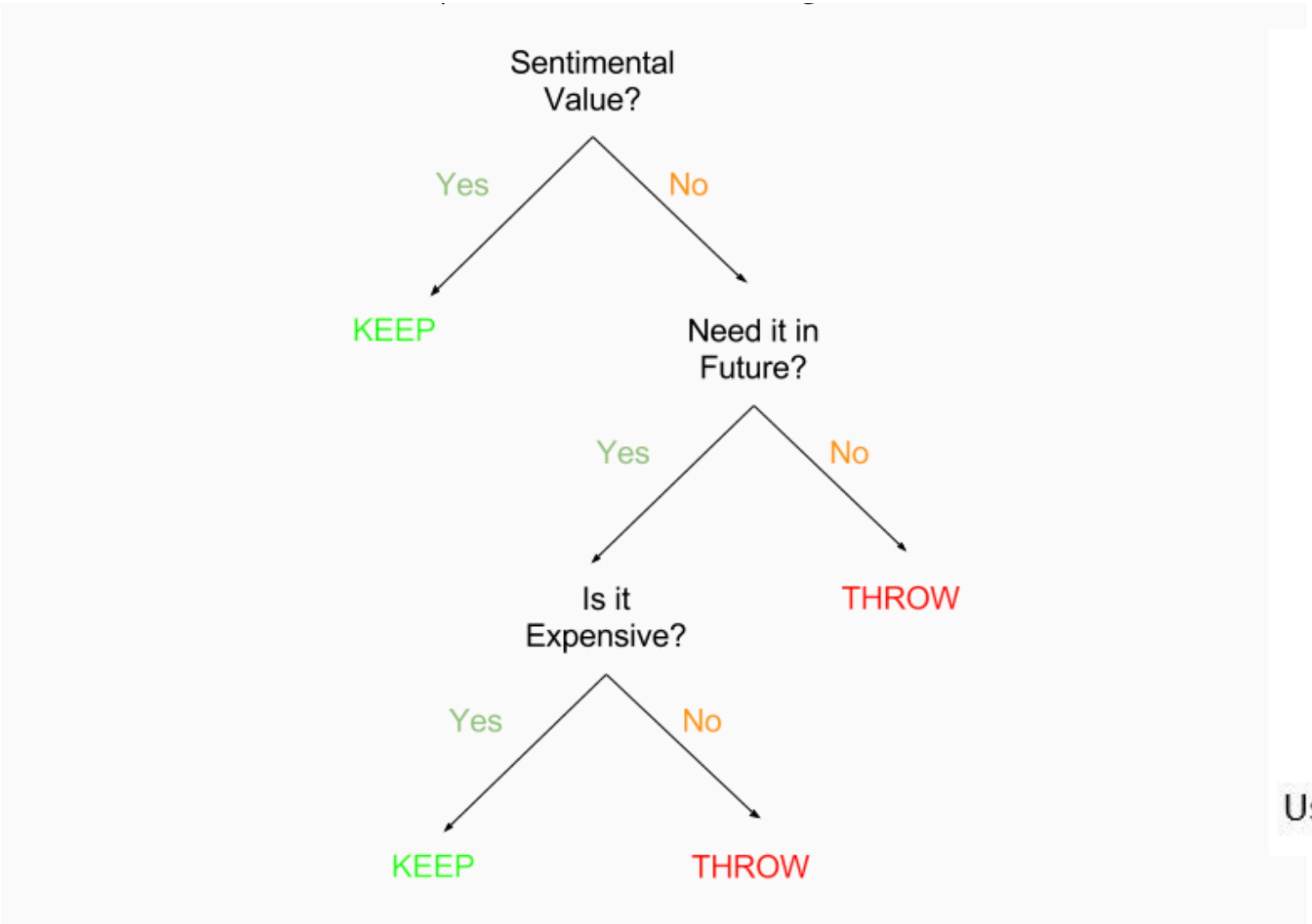
Subhasish Misra & Somedip Karmakar
International Data & Analytics

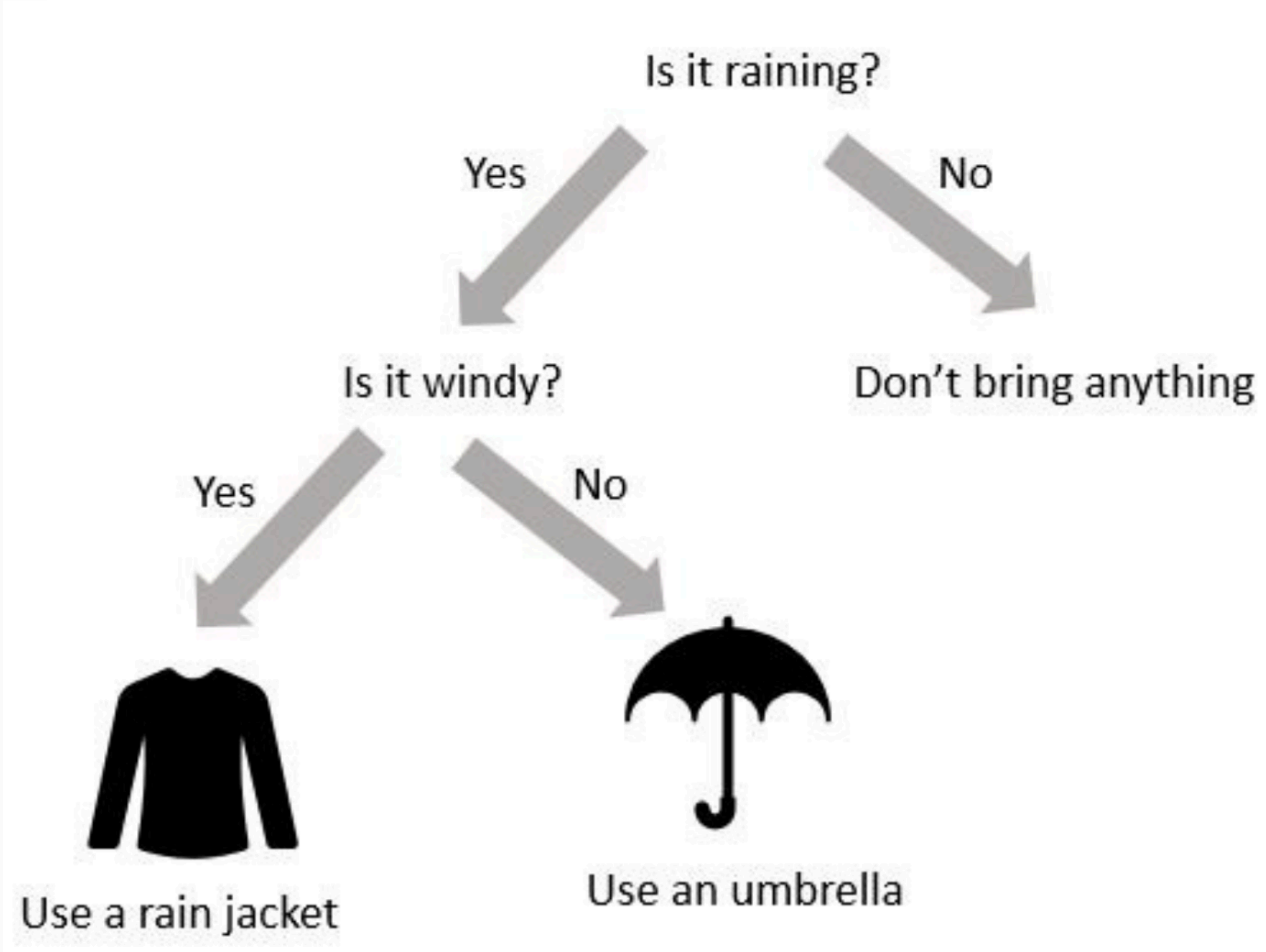
13/08/20

DECISION TREE– AN OVERVIEW

The decision processes in our daily lives

A set of exhaustive if/then rules that helps arrive at a decision !





https://www.kdnuggets.com/2017/08/machine-learning-abstracts-decision-trees.html

https://www.kdnuggets.com/2019/02/decision-trees-introduction.html

A brief recount of machine learning

Walmart

Broadly, the science of helping algorithms learn pattern in data

Machine learning*

Supervised learning

Use **labeled data** to recognize patterns

Regression

Predicting a continuous variable

- *How much $'s will the customer spend in Walmart*
- *How much $'s will the customer spend in Walmart grocery*

Classification

Predicting a class

-Will a customer buy/not buy from us in the next 3 months ?
-Will a customer buy/not buy on Online grocery in the next 6 months ?

Focal area

Unsupervised learning

Use **non-labeled** data to recognize patterns

Clustering

Finding segments in data

- *What are the different Walmart customer segments*

Association rule mining

Finding association between units of analysis

- *If a customer buys beer, what is he likely to buy as well in the same basket?*
- *If a customer buys a lawnmower & a rake n his current transaction – what is he likely to buy in his next transaction*

Classification: Who wants a riding mower !

Walmart

Predictors

Target variable/or the variable we ate trying to predict

Observation	Income (in '000 $'s)	Lot_Size (in '000 sqft)	Ownership
1	60	18.4	owner
2	85.5	16.8	owner
3	64.8	21.6	owner
4	61.5	20.8	owner
5	87	23.6	owner
6	110.1	19.2	owner
7	108	17.6	owner
8	82.8	22.4	owner
9	69	20	owner
10	93	20.8	owner
11	51	22	owner
12	81	20	owner
13	75	19.6	non-owner
14	52.8	20.8	non-owner
15	64.8	17.2	non-owner
16	43.2	20.4	non-owner
17	84	17.6	non-owner
18	49.2	17.6	non-owner
19	59.4	16	non-owner
20	66	18.4	non-owner
21	47.4	16.4	non-owner
22	33	18.8	non-owner
23	51	14	non-owner
24	63	14.8	non-owner

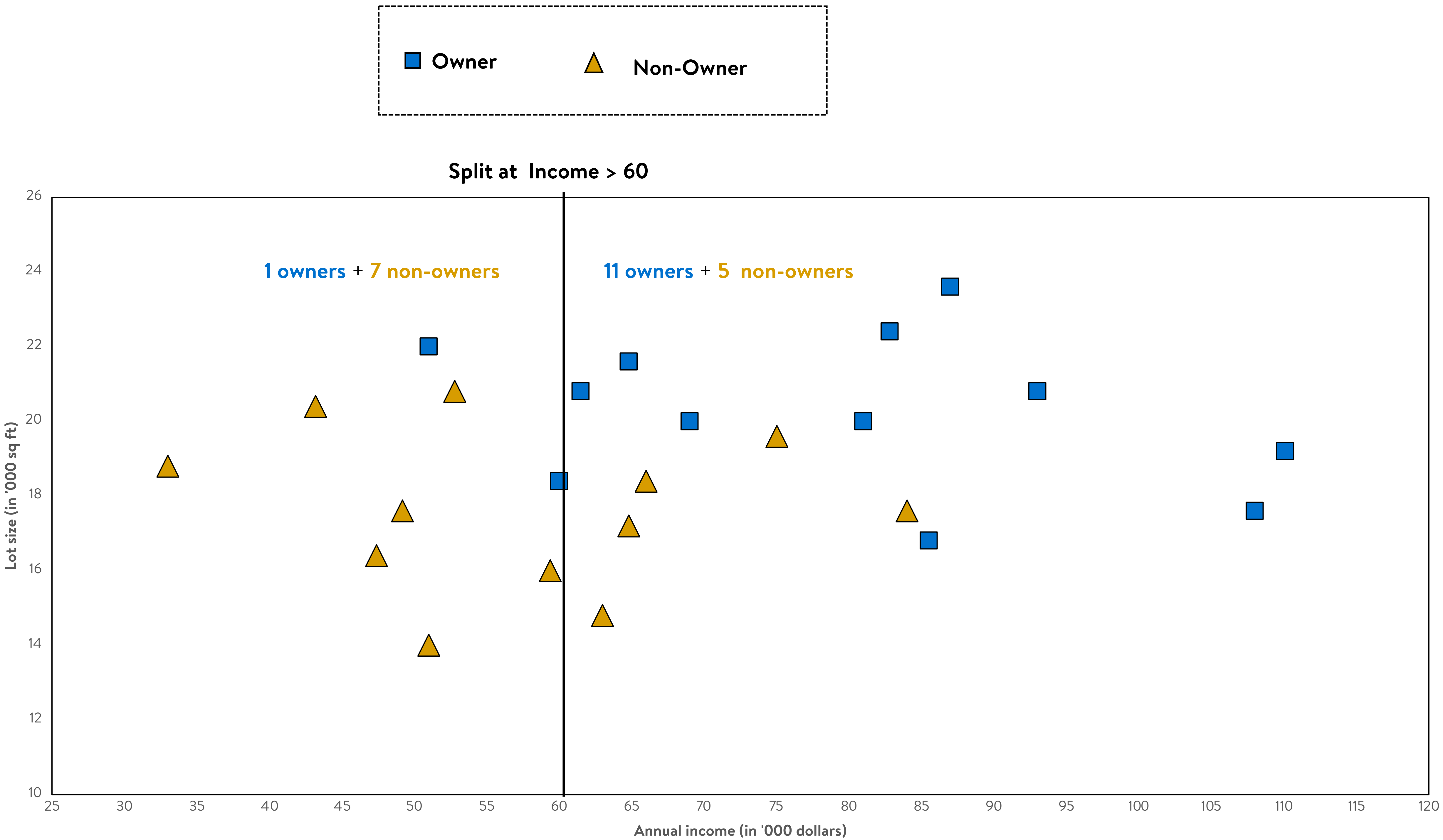
Can we learn to '**classify**' observations to ownership classes ?

Binary classification (2 classes)

12 owners
12 non-owner

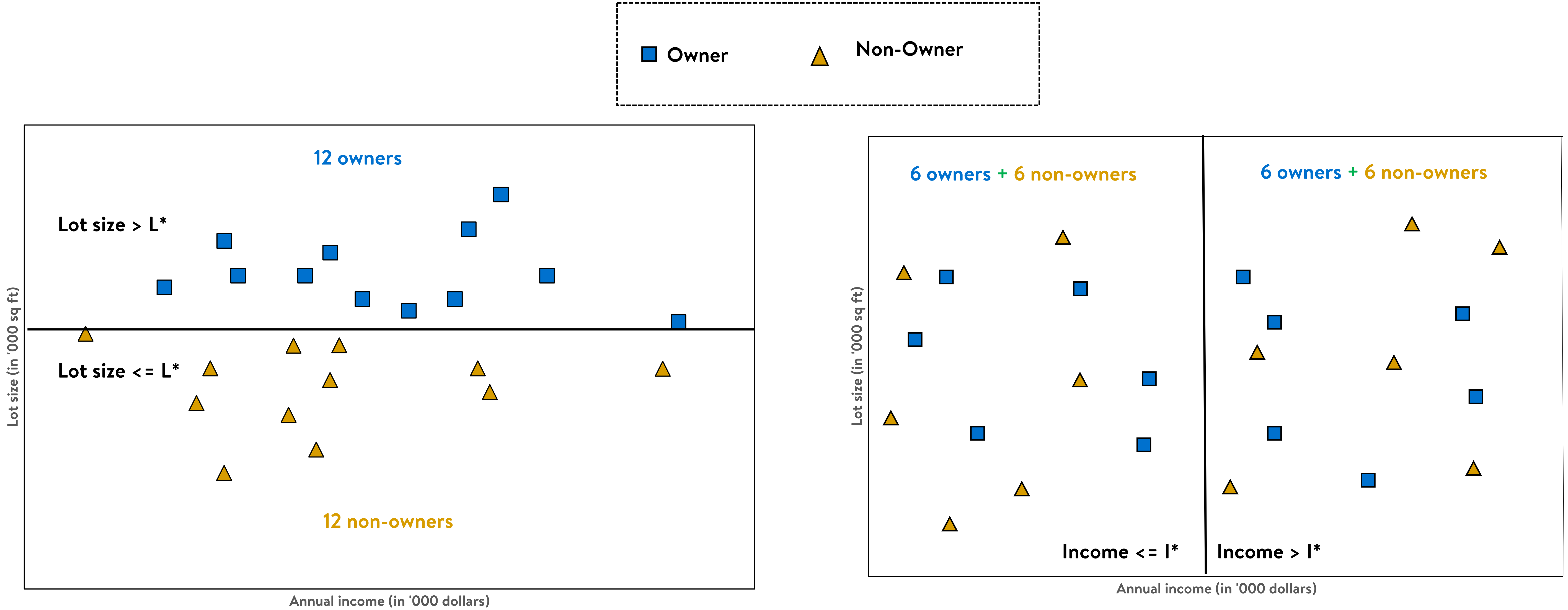
Git repo for session data + code - https://gecgithub01.walmart.com/smisra/TechByte_DTree_session

Motivation – Rule based algorithm



Motivation – deciding on the split point

Walmart



Ideal scenario: This split divides the data space to two homogenous ('pure') parts

Not so Ideal scenario: The two parts are fairly heterogenous

Metrics to understand impurity: Entropy

Entropy is a concept often used computer science (information theory) and other fields of science too
Entropy is a measure of disorder or heterogeneity.

A high on entropy room ☺

Translated in the context of data – how do we actually measure it ?

Metrics to understand impurity: Entropy

Walmart

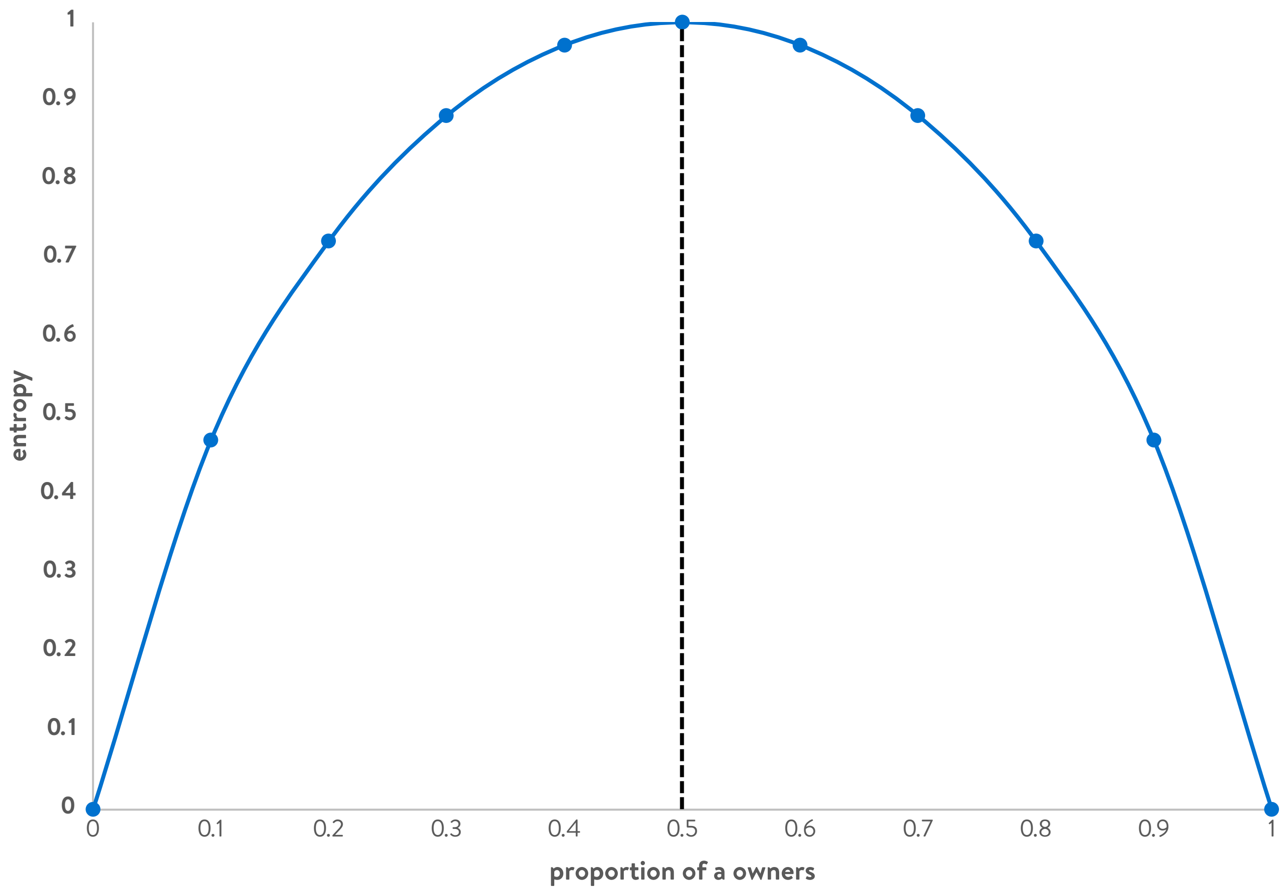
Entropy: $-\Sigma p_{k\ \log(2)}\ p_k^2$

Where p_k is the proportion of observations in class k belonging to a rectangle

Entropy left block (Ditto for right block):

= - [owners /(owner +non-owner) * $\log_2$ (owners/owner +non-owner)^2 +
non-owners /(owner +non-owner) * $\log_2$ (non-owners / owner +non-owner)^2

= - [6/12 * $\log_2$ (6/12)^2 + 6/12 * $\log_2$ (6/12)^2]
= - [.5 * -1 + .5* -1] = 1



Gets **maximized** when we have equal representation of the classes (for a 2 class problem)

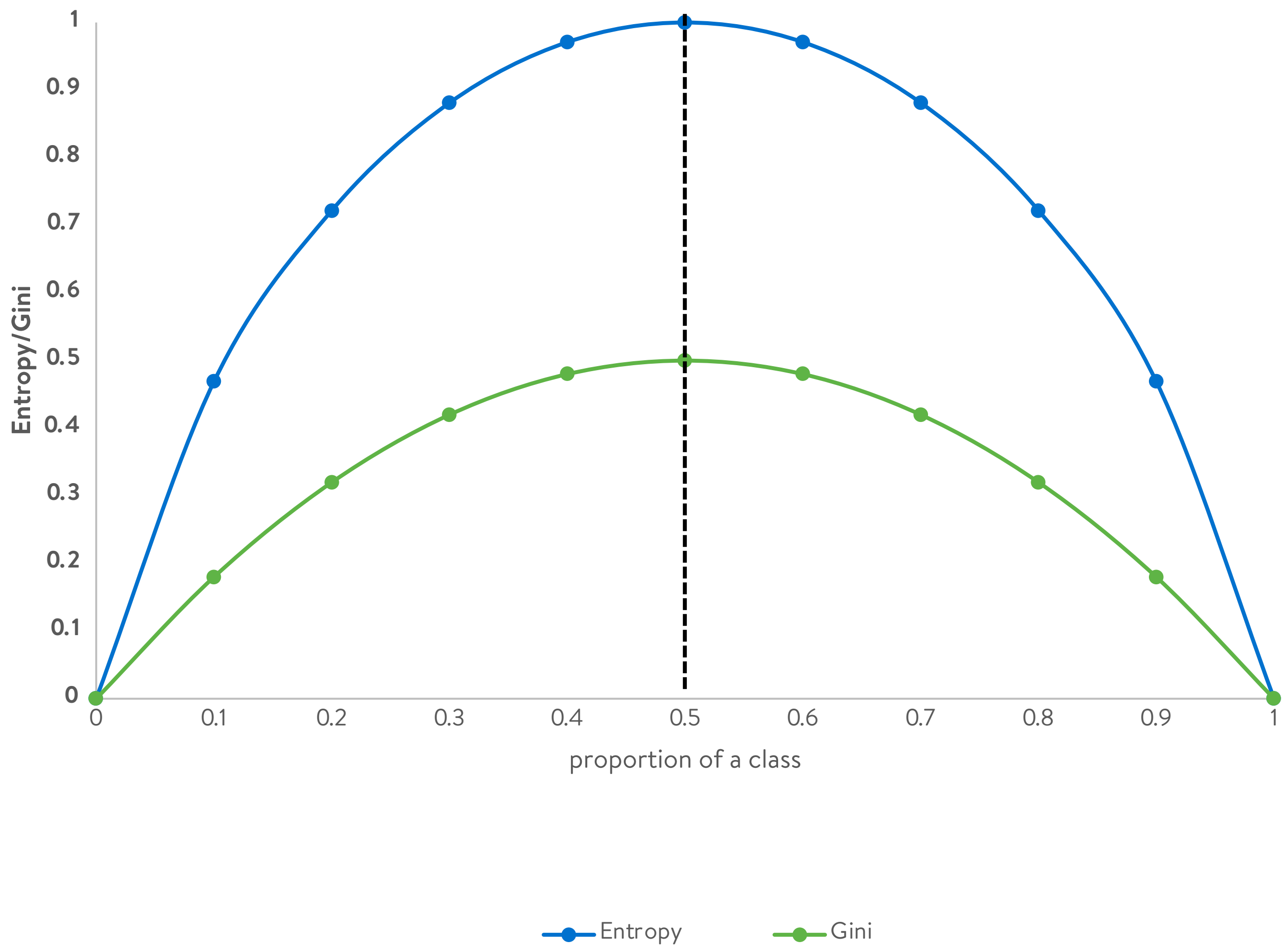
Metrics to understand impurity- Gini index

Gini index: $1 - \Sigma p_k^2$

Where p_k is the proportion of observations in class k belonging to a rectangle

Gini index: Left block (Ditto for right):

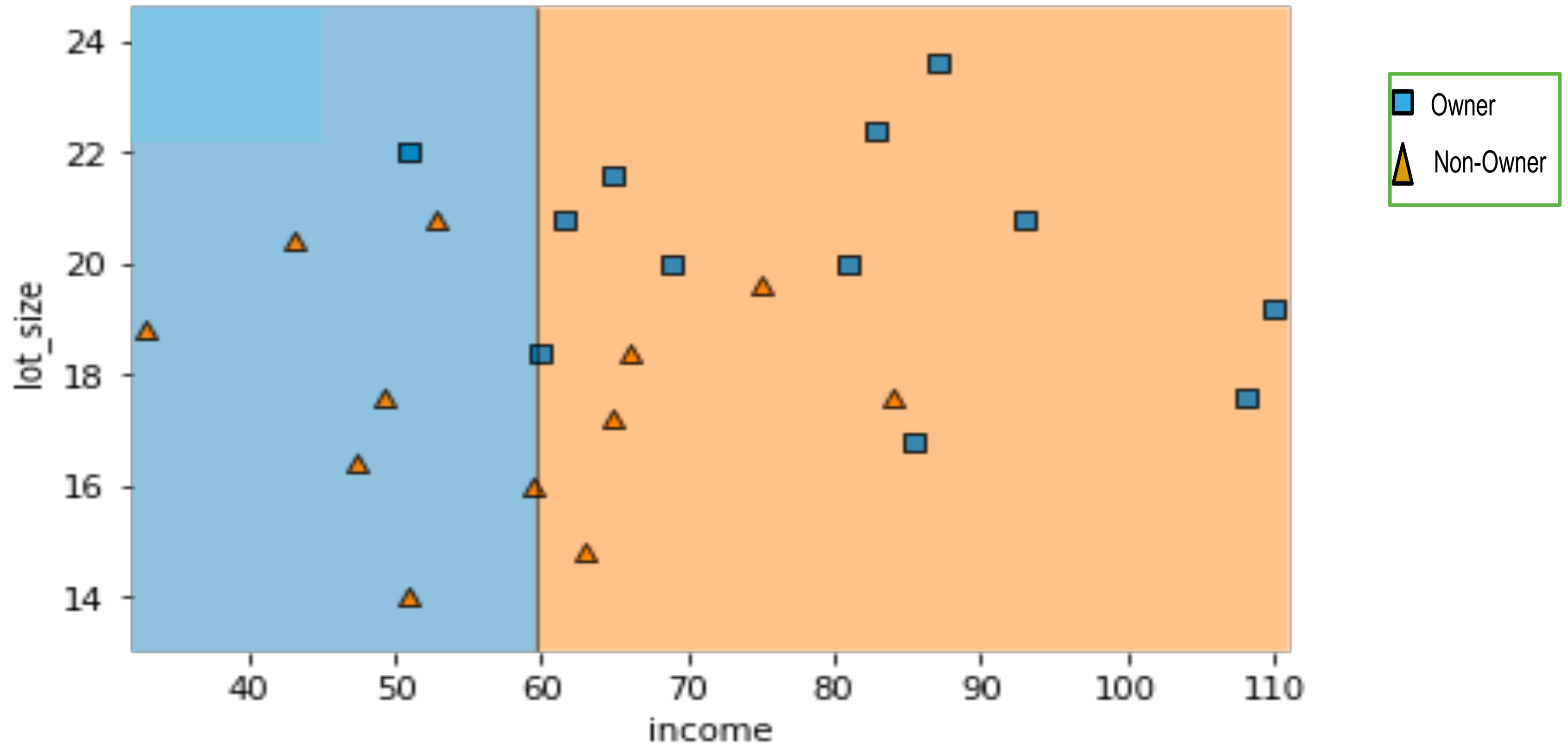
$= 1 - (\text{owners/owner + non-owner})^2 - (\text{non owners/owner + non-owner})^2$
$= 1 - (6/12)^2 - (6/12)^2$
$= 1 - .25 - .25$
$= .5$



Gets **maximized** when we have equal representation of the classes

Choosing the best split- by max reduction in impurity

Walmart



Gini (no split scenario) $= 1 - (\text{owners/owner + non-owner})^2 - (\text{non-owners/owner + non-owner})^2$
$= 1 - (12/24)^2 - (12/24)^2 = 0.50$

For a split at Income = 60 $ scenario:

Gini (Left block) $= 1 - (1/8)^2 - (7/8)^2 = 0.22$

Gini (Right block) $= 1 - (11/16)^2 - (5/16)^2 = 0.43$

Overall Gini = (obs in left block /Total obs) * 0.22+ (obs in right block/Total obs) * 0.43
= (8 /24) * 0.22+ (16/24) * 0.43 = .35

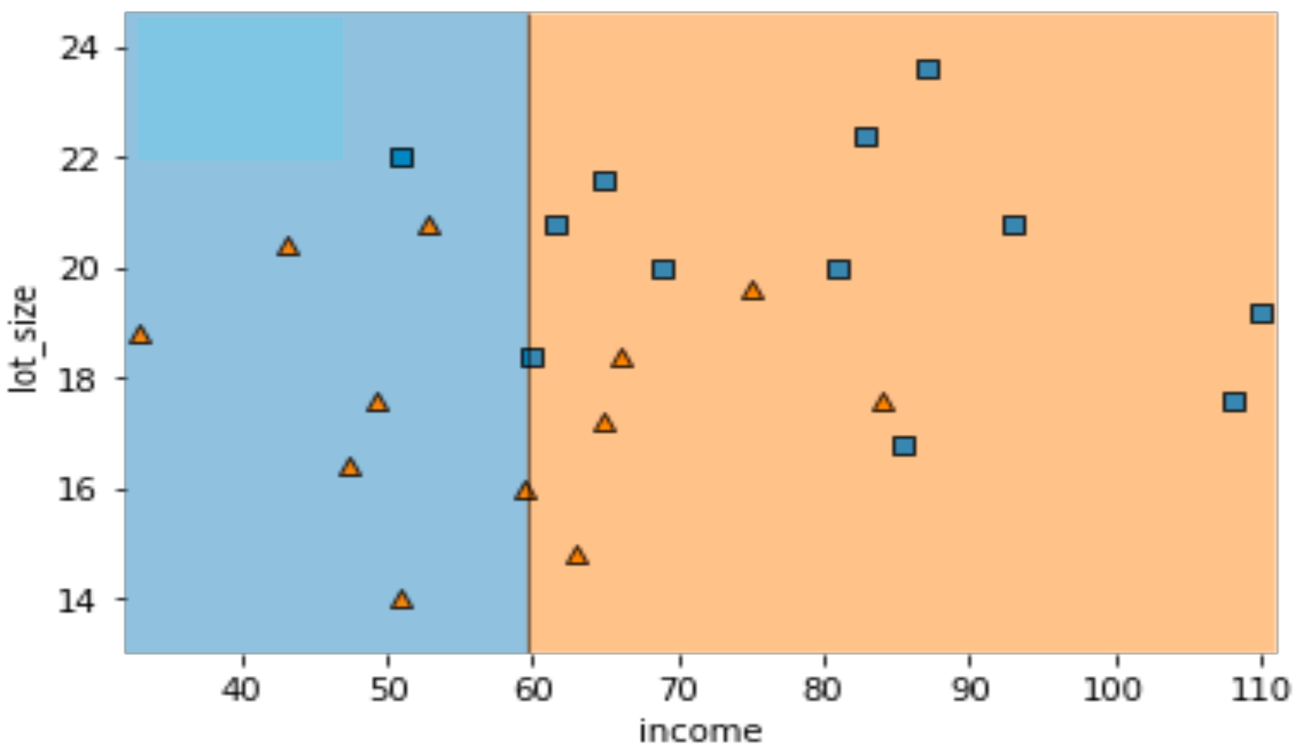
Information gain = $\textbf{Gini}_{\text{No split}} - \textbf{Gini}_{\text{after split}}$ = .50 - .35 = .15

- **Do this for all possible split points:**
  - ✓ **Across the two variables**
  - ✓ **Compare information gain**
  - ✓ **Choose split having max gain**

Turns out income > 59.7 is best first split

Choosing the best split- by max reduction in impurity - continued

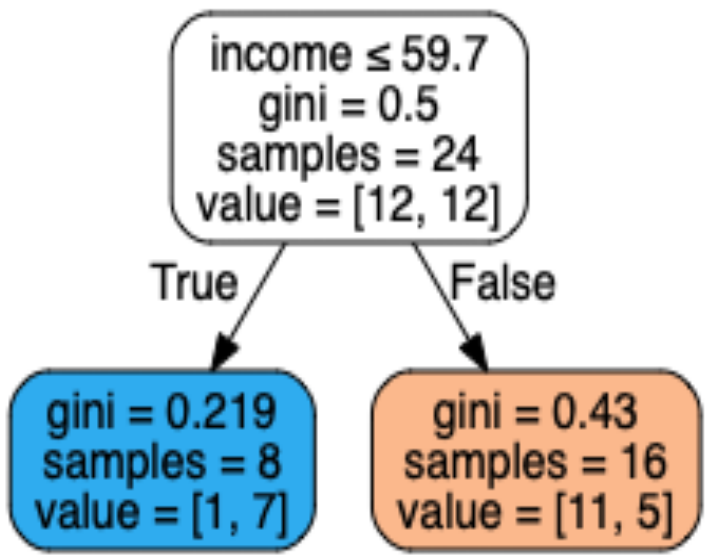
Walmart



Decision Tree representation

Tree stump – **A simple tree with one split**

root node



Depth = 1

Leaf node

- Gini index has a value of .21
- There are 8 observations in the node
- For income<=59.7 bucket there are:
  - 1 Owner
  - 7 Non-Owners

Leaf node

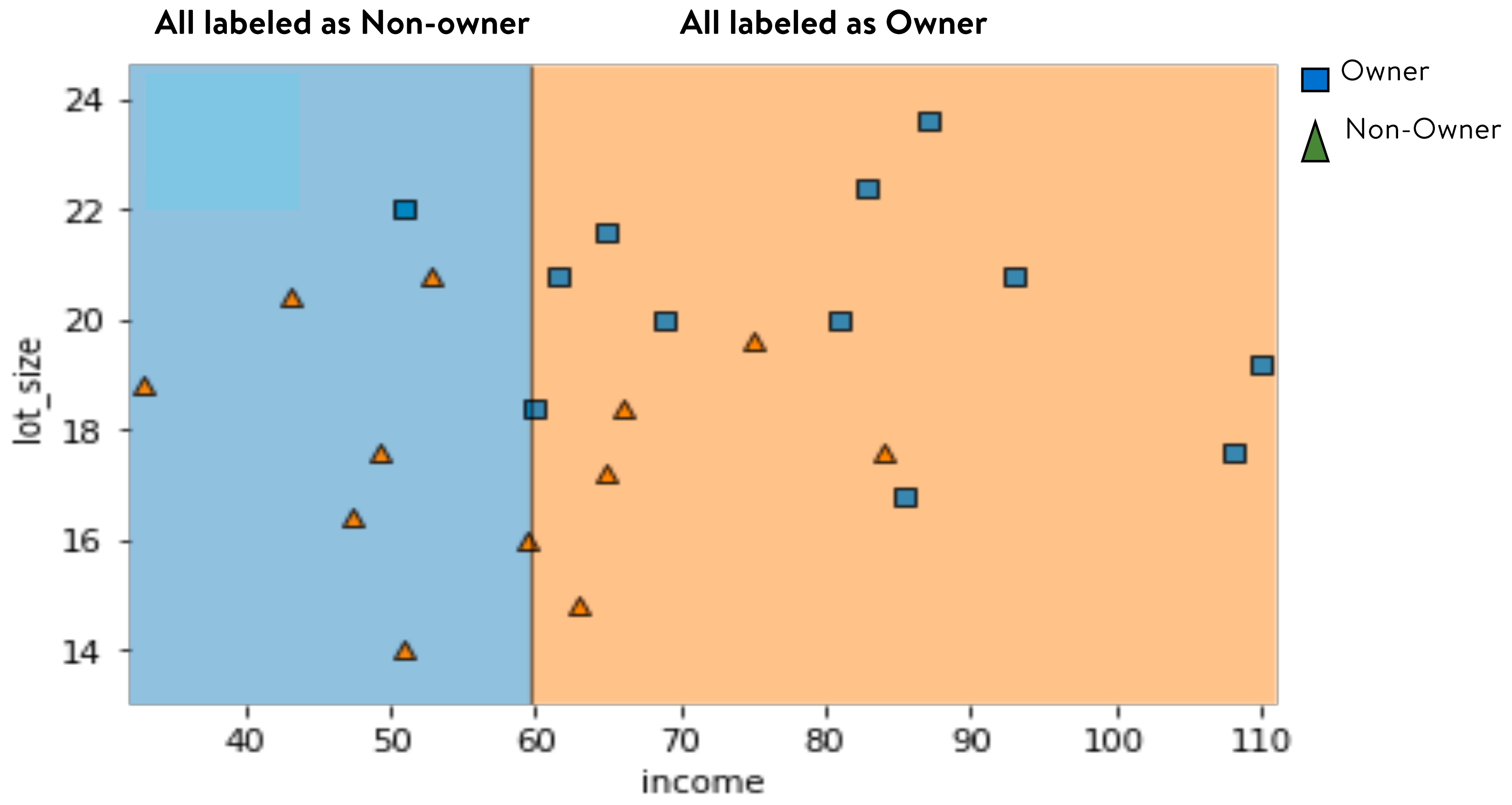
- Gini index has a value of .43
- There are 16 observations in the node
- For income>59.7 bucket there are:
  - 11 Owners
  - 5 Non- Owners

Proportion of Non-owner is higher than .5. observations in node classified as **non-owners**

Proportion of Owner is higher than .5. Observations classified as **owners**

Git repo for session data + code - https://gecgithub01.walmart.com/smisra/TechByte_DTree_session

Goodness of fit - Accuracy/Sensitivity/Specificity



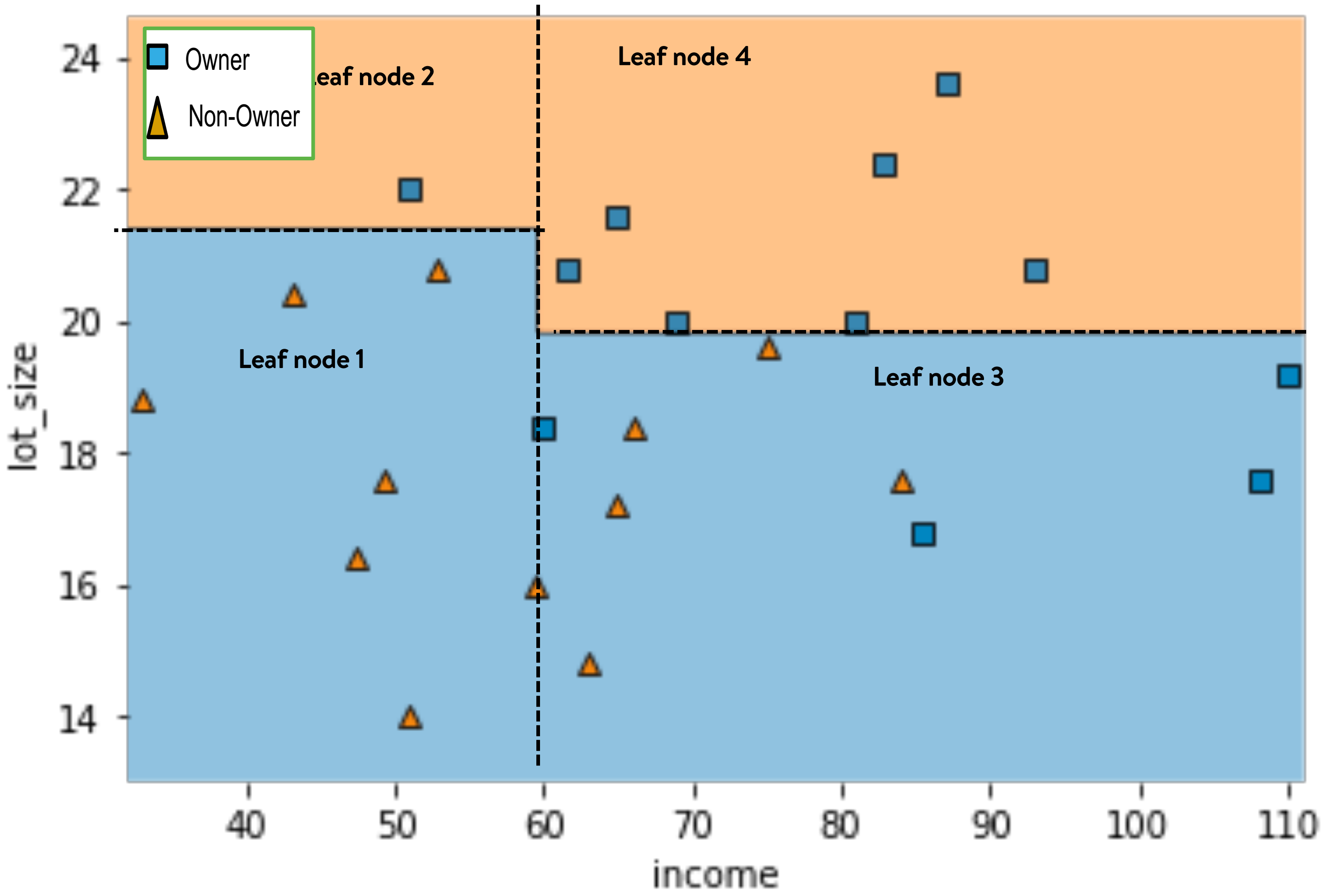
Accuracy = Proportion of correctly classified observations = Correctly classified observations/Total obs = (7 +11)/ 24 = 75 %

Sensitivity = Proportion of the owner's who were correctly classified = Correctly classified owners/ Total owners = 11 /12 = 92%

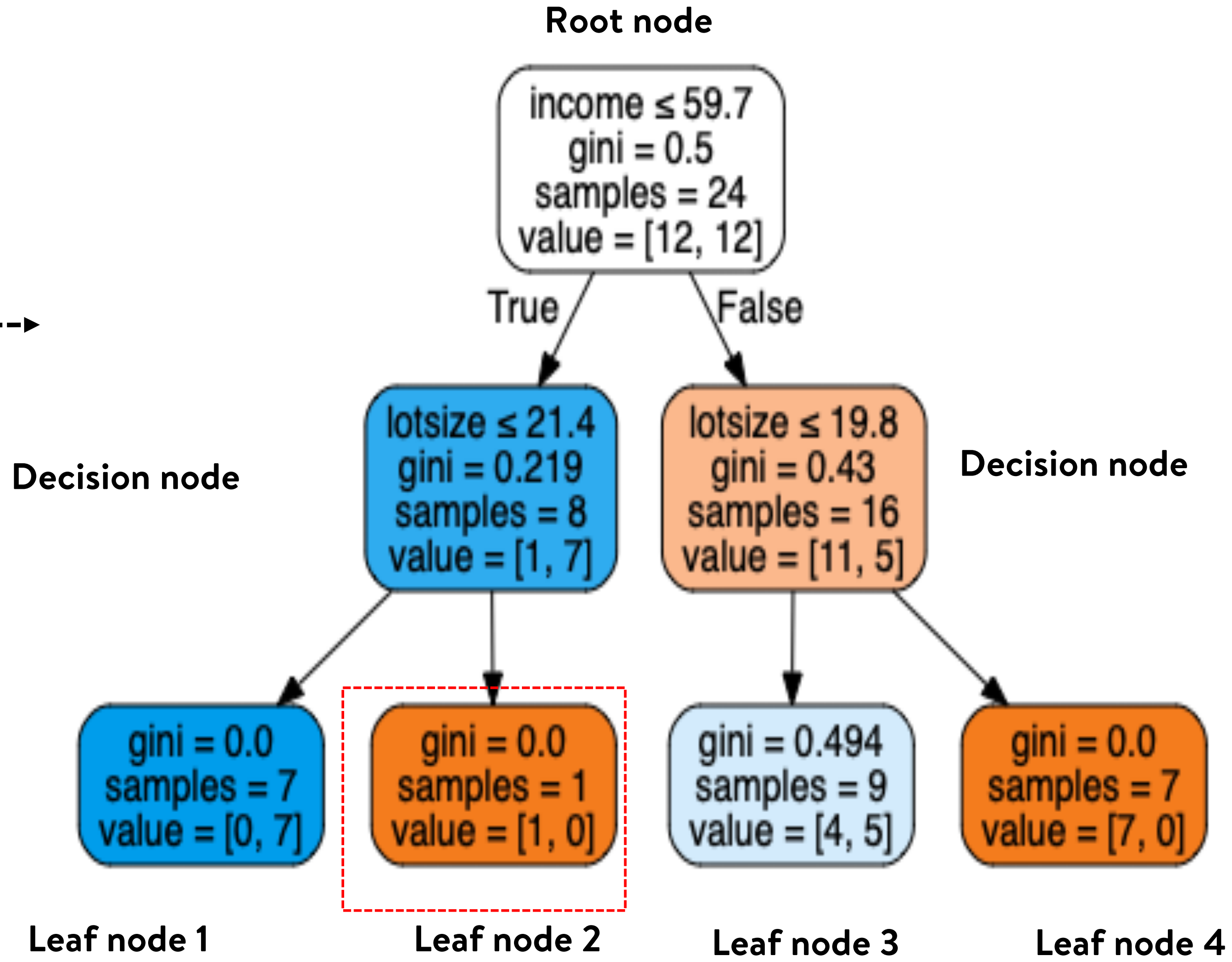
Specificity = Proportion of non-owner's who were correctly classified = Correctly classified non-owners/ Total non-owners = 7/12 =58 %

Recursive partitioning

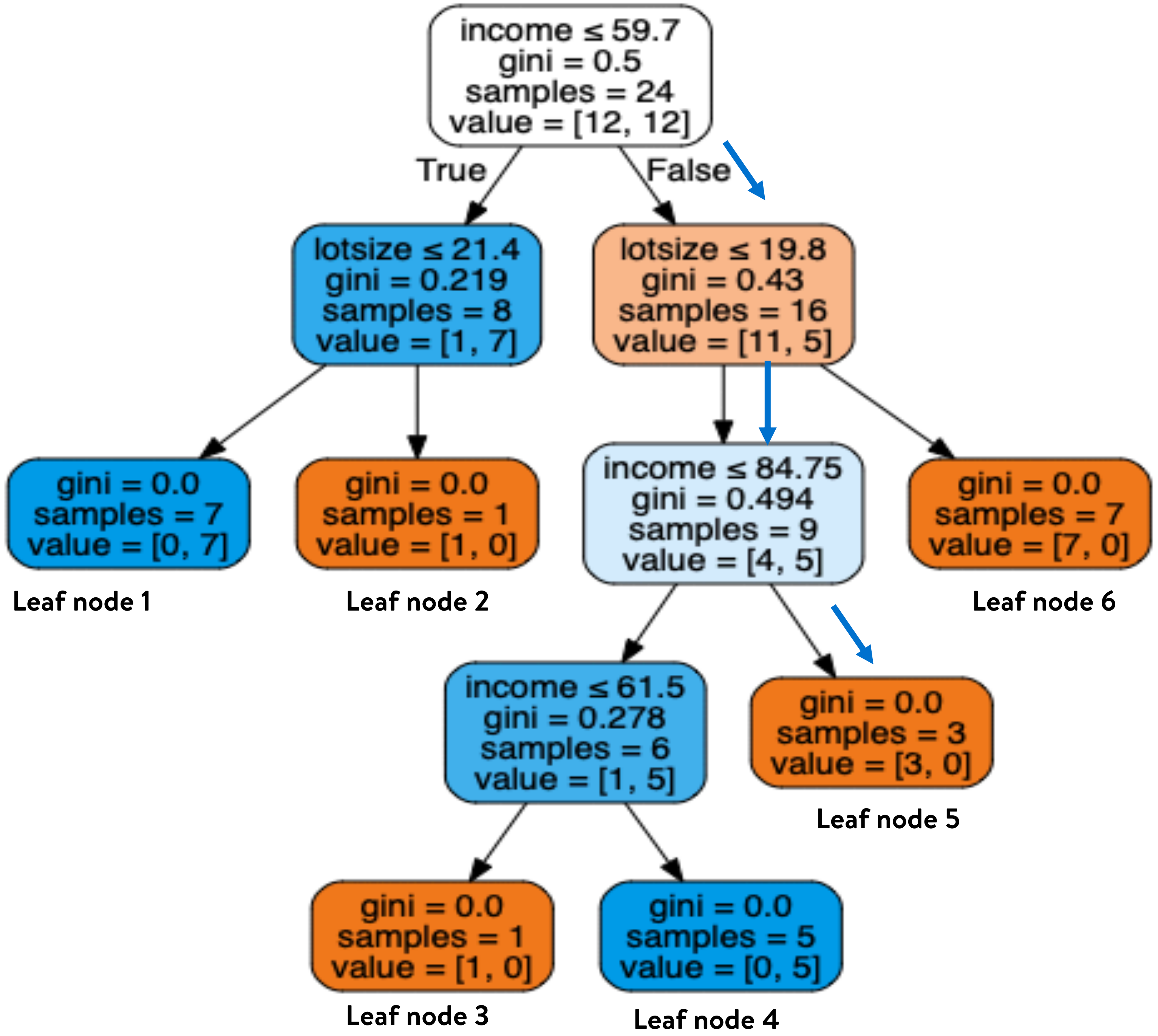
The algorithm 'greedily' searches the space for the next optimal split



Decision Tree representation



The fully grown tree – interpreting the rules



Classification rules:

Leaf node1: If (income <=59.7) & (Lot size is <=21.4) -> non-owner

Leaf node 6: If (income > 59.7) & (Lot size is > 19.8) -> owner

Classifying an observation: household 7 (income = 110.1, Lot size = 19.2)
Classified as : Owner
Actual class: Owner

Accuracy (the % of correctly classified observations): 100%

* Note: There are a gamut of tree algorithms that are available (ID3, C4.5. CHAID, CART etc.)
The one used here is closest to CART (Leo Breiman et al, 1984)

Who should we offer a loan ?

Walmart

- A relatively young American bank is growing rapidly in terms of overall customer acquisition.
- Majority of these are customers with varying sizes of relationship with the bank.
- The customer base of Asset customers is quite small, and the bank WANTS to grow this base rapidly to bring in more loan business.

5000 observations x 11 Predictors

Target variable

Age	Experience	Income	Family	CCAvg	Education	Mortgage	Securities Account	CD Account	Online	CreditCard	**Personal Loan**
25	1	49	4	1.60	1	0	1	0	0	0	0
45	19	34	3	1.50	1	0	1	0	0	0	0
39	15	11	1	1.00	1	0	0	0	0	0	0
35	9	100	1	2.70	2	0	0	0	0	0	0
35	8	45	4	1.00	2	0	0	0	0	1	0
37	13	29	4	0.40	2	155	0	0	1	0	0
53	27	72	2	1.50	2	0	0	0	1	0	0
50	24	22	1	0.30	3	0	0	0	0	1	0
35	10	81	3	0.60	2	104	0	0	1	0	0
34	**9**	**180**	**1**	**8.90**	**3**	**0**	**0**	**0**	**0**	**0**	**1**
65	39	105	4	2.40	3	0	0	0	0	0	0

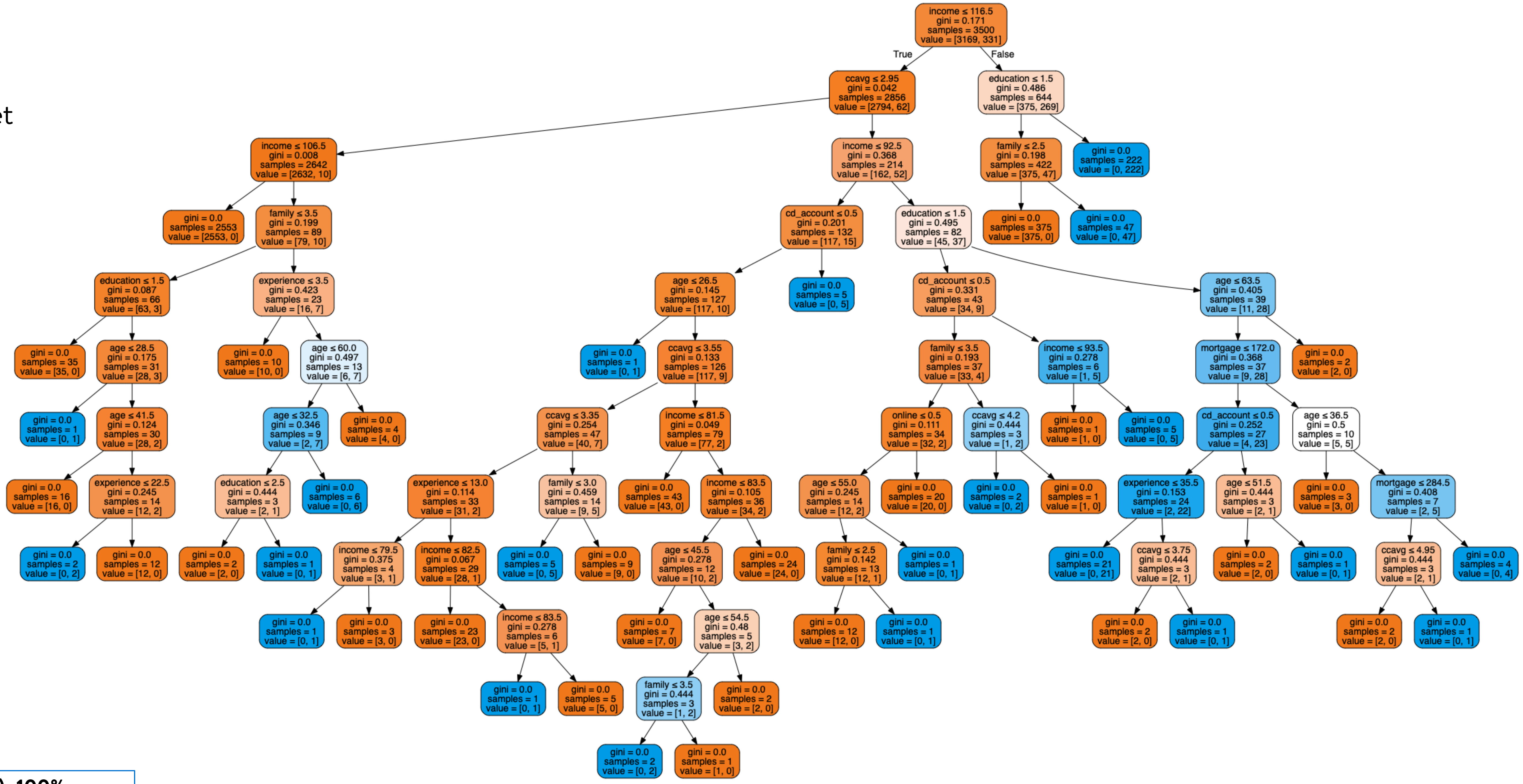
9 % of the customers are loan customers in this dataset

- We'll keep a random set of 70% of this data for calibrating (or training the tree)
- The remaining 30% for testing * – to how does the model fare in the wild

Our fully grown tree for loan grant is complex !

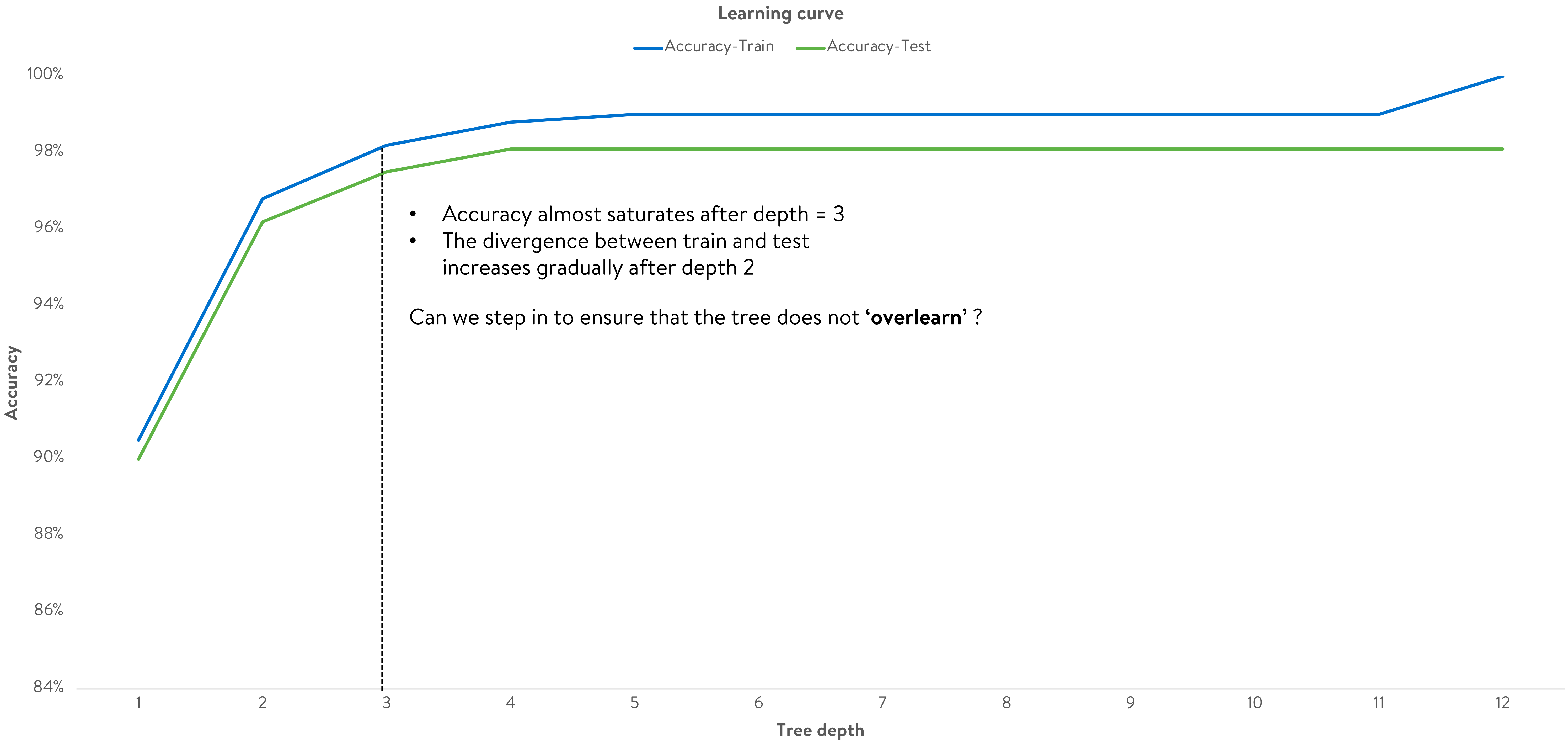
Walmart

* For the training set



Accuracy (Training): 100%
Accuracy (Test): 98%

...and likely an overkill



- Accuracy almost saturates after depth = 3
- The divergence between train and test increases gradually after depth 2

Can we step in to ensure that the tree does not **‘overlearn’** ?

Pruning to the rescue

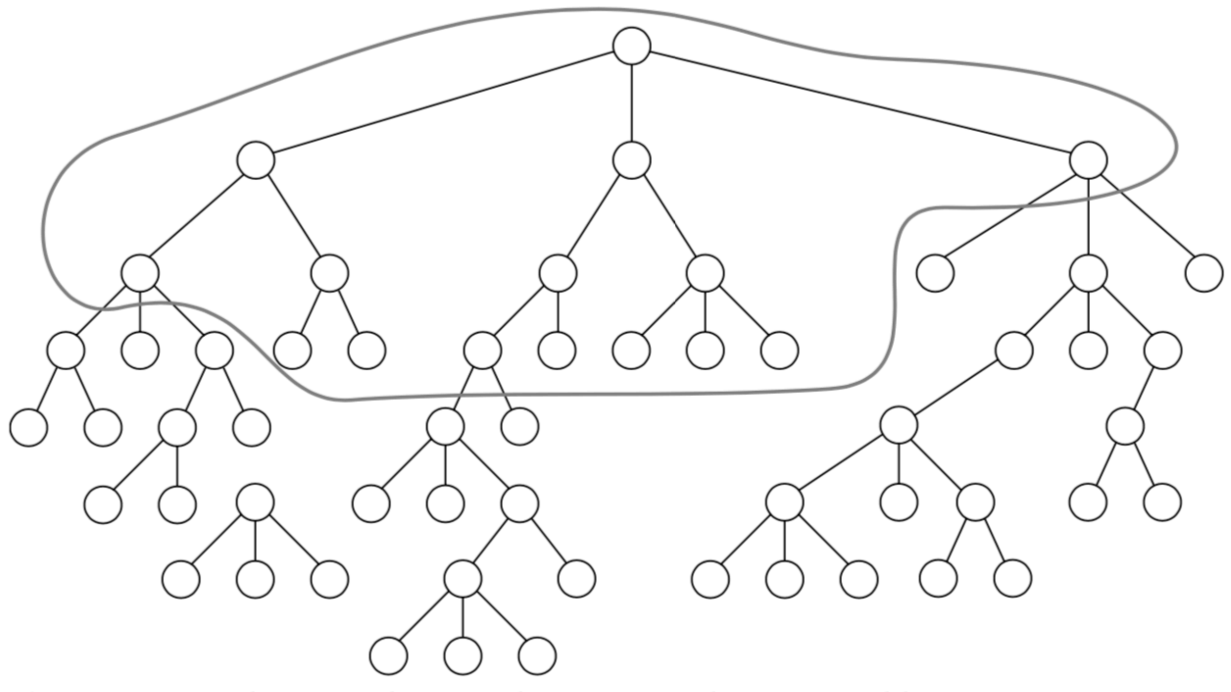
prune[2]
/pruːn/

verb
gerund or present participle: **pruning**

trim (a tree, shrub, or bush) by cutting away dead or overgrown branches or stems, especially to encourage growth.
"now is the time to prune roses"

Inside a complex tree, there are simpler, more stable trees.

*Stihl Shop Greenburg – some rights reserved



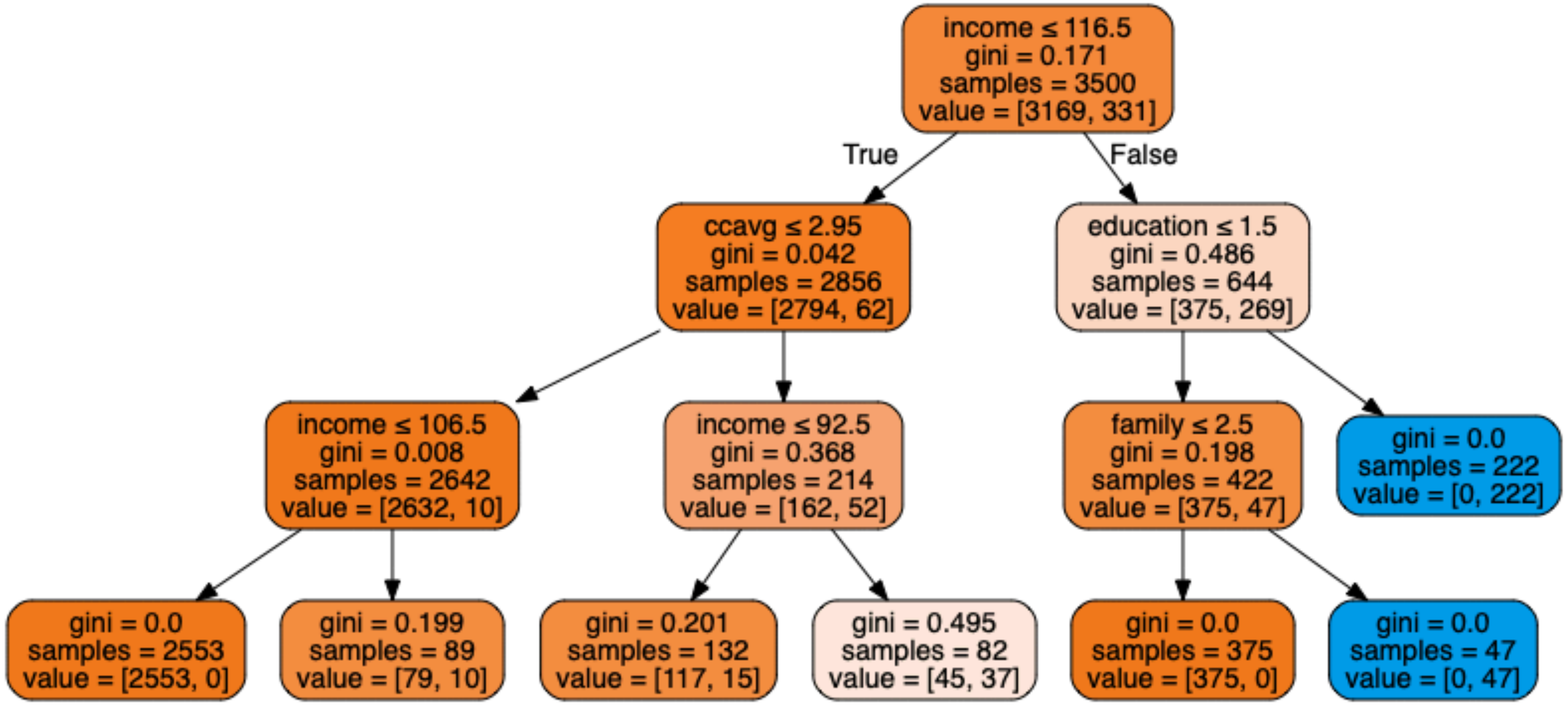
*Depiction from 'Data mining techniques for marketing, sales, CRM', 3rd ed – Berry et al

Pruning – setting the max depth

For example, we've set, maximum depth =3 here. **The train & test accuracy is around 98% here**

There are multiple ways to prune beyond just this – for e.g. :

- **Minimum # of observations in a leaf node**
- **Min # of observations to continue splitting**
- **Min decrease in impurity**



Hyper-parameter * tuning

CARTESIAN GRID SEARCH

Specify grid

Hyper-parameter 1 *

Max_depth
1
3
5

Hyper-parameter 2

Min Obs
3
4
5

Fit Tree over Train/Validation

{1,3}, {1,4}, {1,5}

{3,3}, {3,4}, {3,5}

{5,3}, {5,4}, {5,5}

Observe model goodness metric over Train/Test

Where in the grid is this best ?

	Model 1	Model 2	Model 3
Train	Goodness metric*	Goodness metric	Goodness metric
Test	Goodness metric	Goodness metric	Goodness metric

* Could be accuracy, sensitivity, specificity, precision, F1, AUC etc.

OTHER SEARCH STRATEGIES

- Random Grid search : Searches the space of parameters randomly, not exhaustive. Computational cheaper.
- Bayesian grid search : Keep track of past evaluation results which they use to form a probabilistic model mapping hyperparameters to a probability of a score on the objective function:

* *https://machinelearningmastery.com/difference-between-a-parameter-and-a-hyperparameter/*

Advantages and disadvantages

Walmart

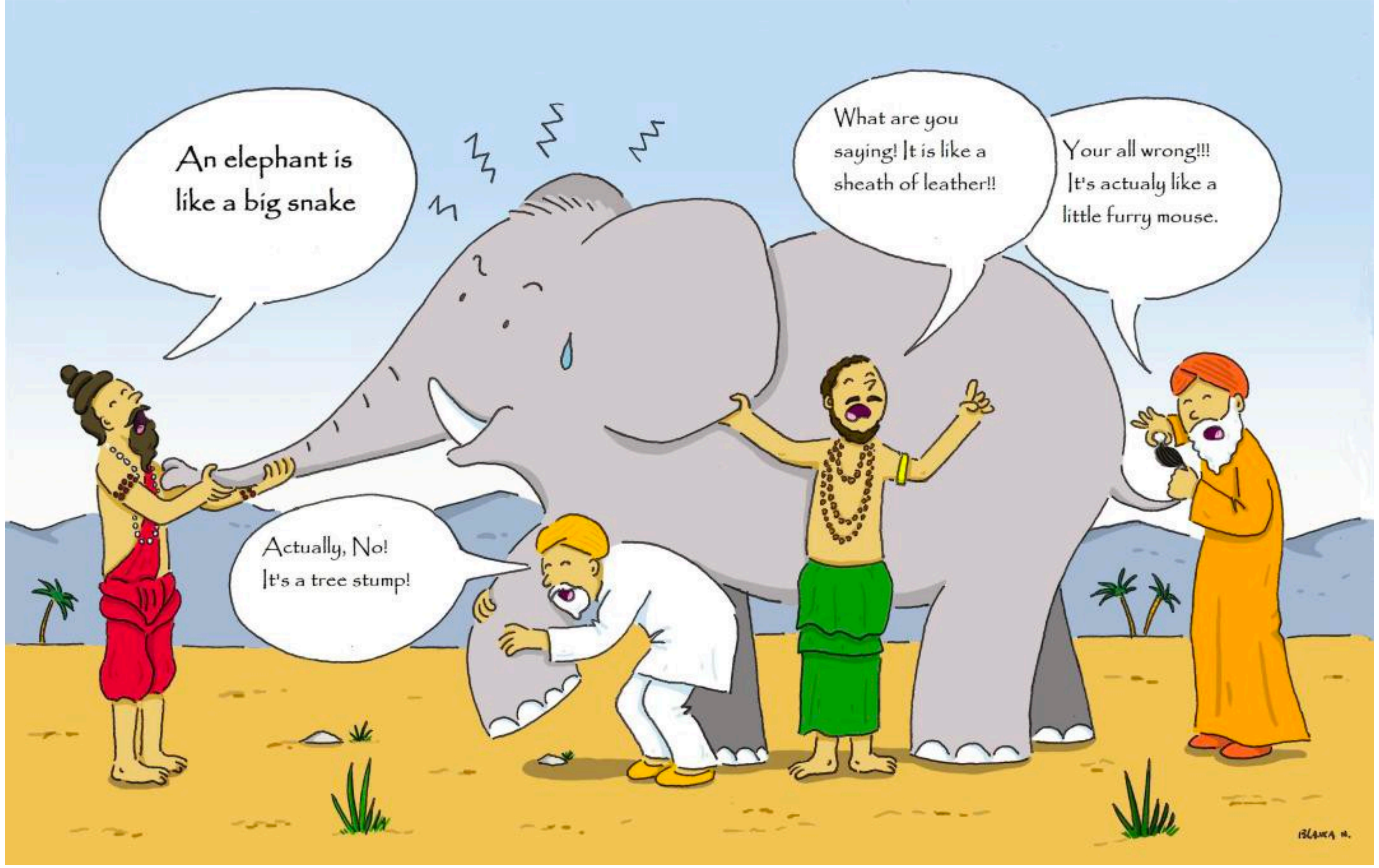
Advantages :

- Easy to interpret and visualize.
- Can model complex patterns quite well.
- Needs limited assumptions – mainly data driven
- Can be used for classification as well as regression problems

Disadvantages:

- High tendency to overfit to the data used for training
- Small variation(or variance) in data can result in the different decision tree.

The blind men and the elephant

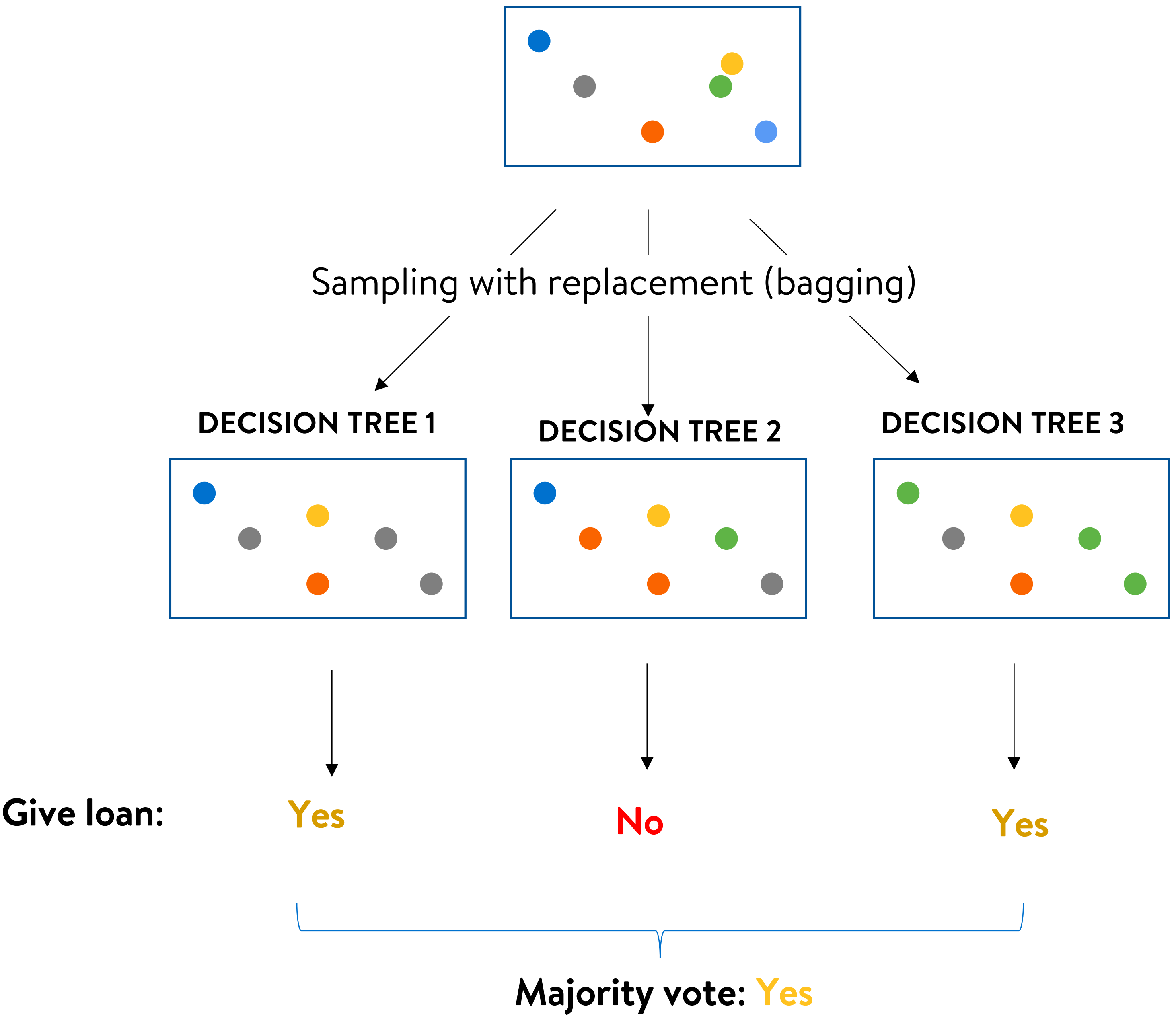


* https://medium.com/diogo-menezes-borges/ensemble-learning-when-everybody-takes-a-guess-i-guess-ec35f6cb4600

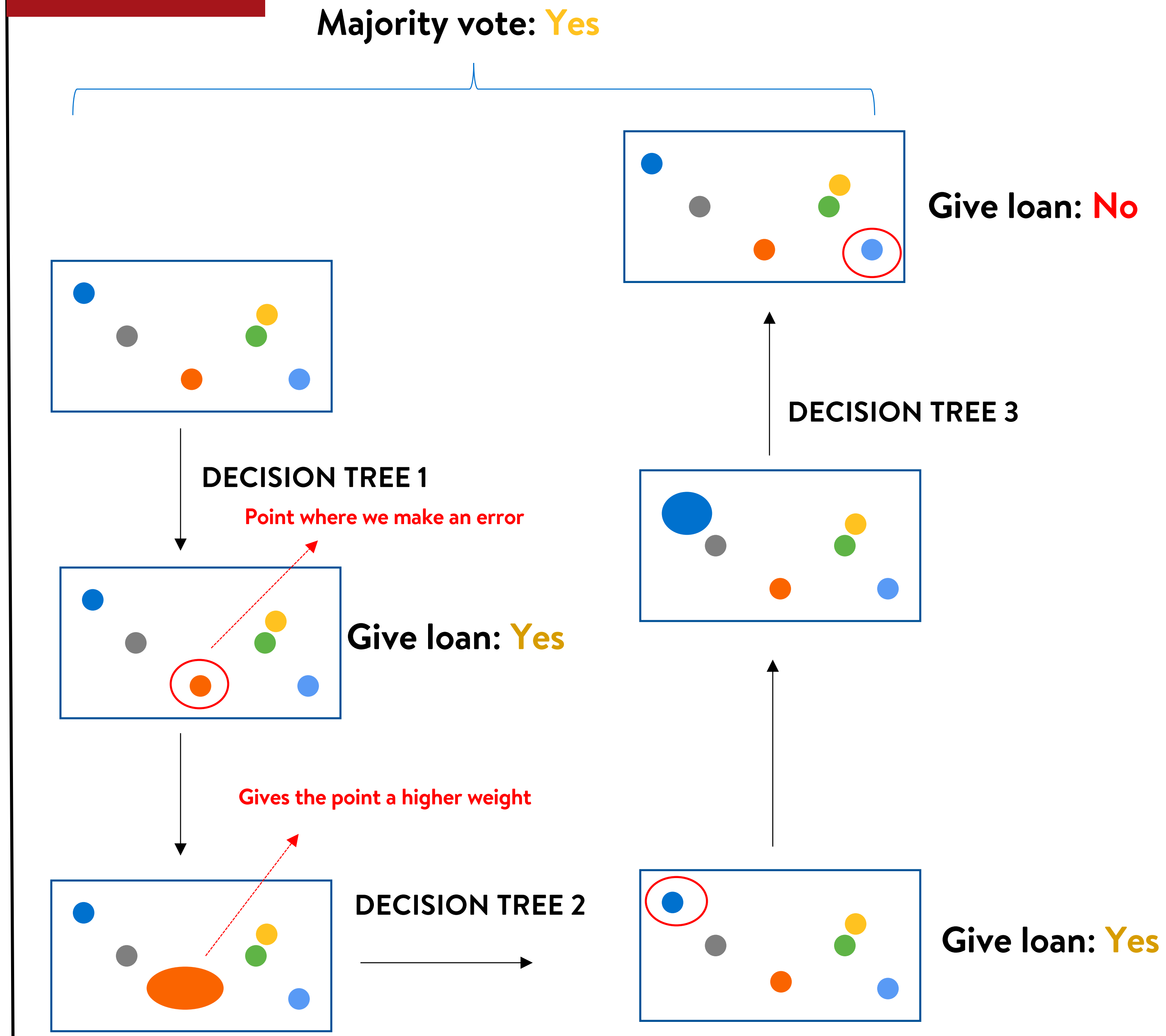
A visual guide to ensembles in Machine learning

Walmart

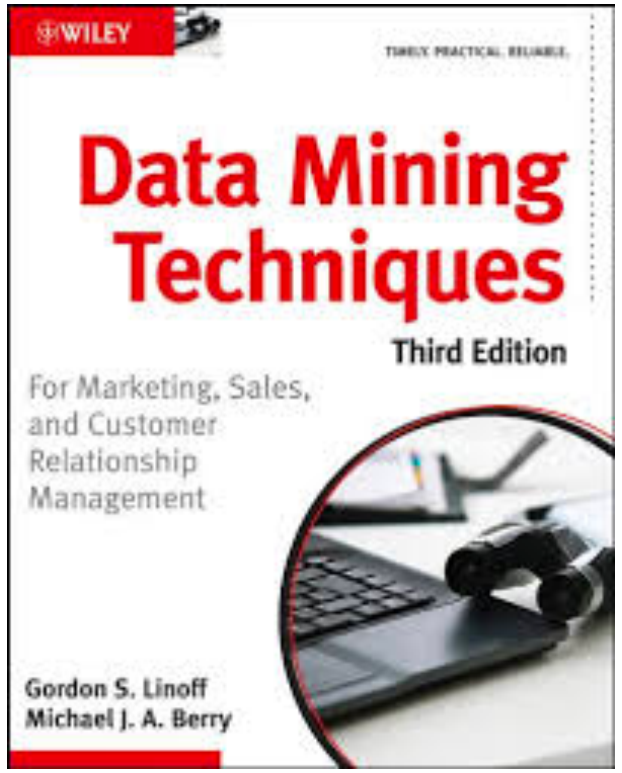
BAGGING

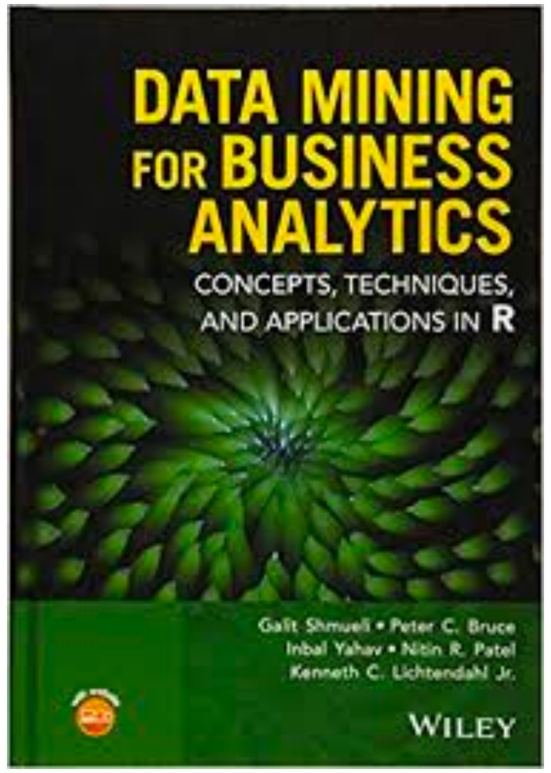


BOOSTING



A few resources to get started





Git repo for session data + code - https://gecgithub01.walmart.com/smisra/TechByte_DTree_session

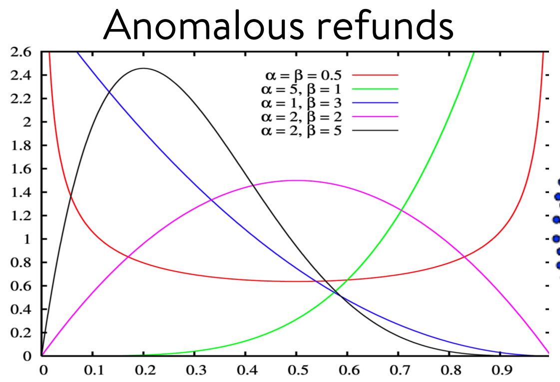
WALMART USE CASE

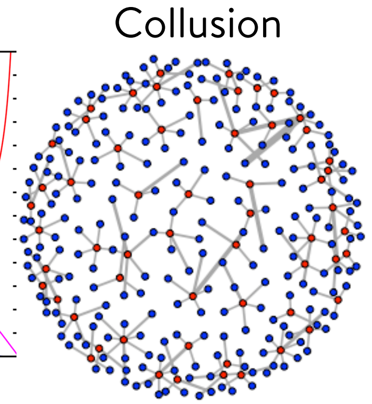
Walmart

Omni-Channel Retail Fraud Detection: Refunds, Cancellations, Discounts, Collusions

Mission

Detect and **Prevent** refund and cancellation fraud by customers and collusion with drivers and employees, and identify process improvement opportunities.

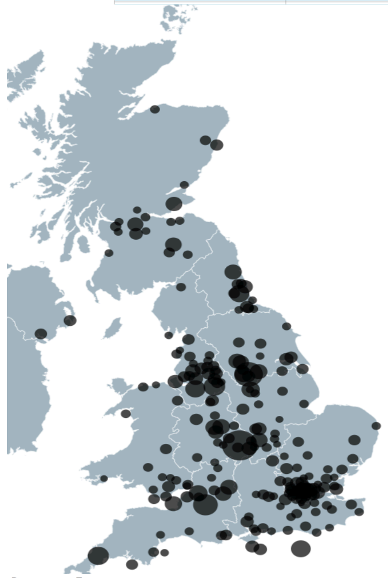




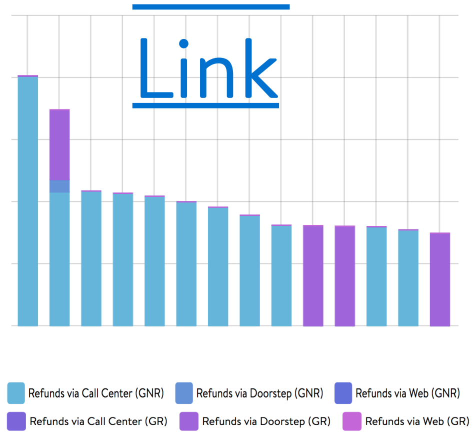
Features

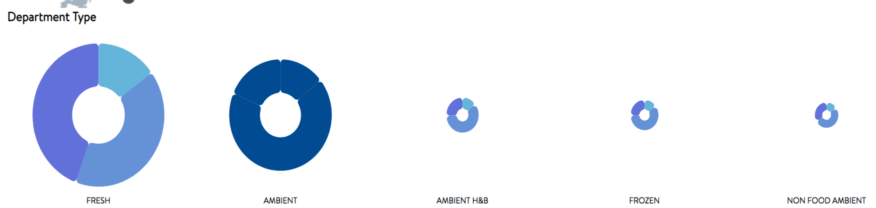
KEY BENEFITS

- Highlights instances of cancellation and refund **abuse** by customers
- Identifies **collusions** among customers, drivers, store associates
- Identifies cases of colleague **discount** & reselling abuse
- Risk assessment by geographic locations & merchandising hierarchy
- **Prioritizes** cases to take appropriate action by ML generated risk scores
- Discovers common fraud ***modus operandi*** to mitigate future risk



Tool Link





Features to Profile Customer Risk

Walmart

Fraud Risk KPIs

- High Refund Amount/Frequency
- High Risky Cancellations
- High Refund Rate
- GNR Refunds/proportion
- High refunds through Web, Call Center or Doorstep
- Recency

Multi-party Collusion

- Always refunding with same driver with high refund amount
- Always cancelling orders by same employee after Pick complete

Risk by Association

- Share of High-refunding Stores, cities, postcodes
- Refunds made in high value items and in high risk stores

Suspicious Behavior

- Repeated refunds of same item
- Recent spike in refunds
- Doesn't return Damaged Items or Unwanted Substitutes
- Refunds at a higher price

How risk scores are computed?

Walmart

01

Metric Aggregation

Customer level aggregated refund, cancellations, collusion, recency

~60 Features

02

Feature Selection & Cleaning

Remove highly correlated variables to remove multi-collinearity, normalize the metrics

~40 Features

03

Choice of Weights

Risk score is weighted aggregate of metrics

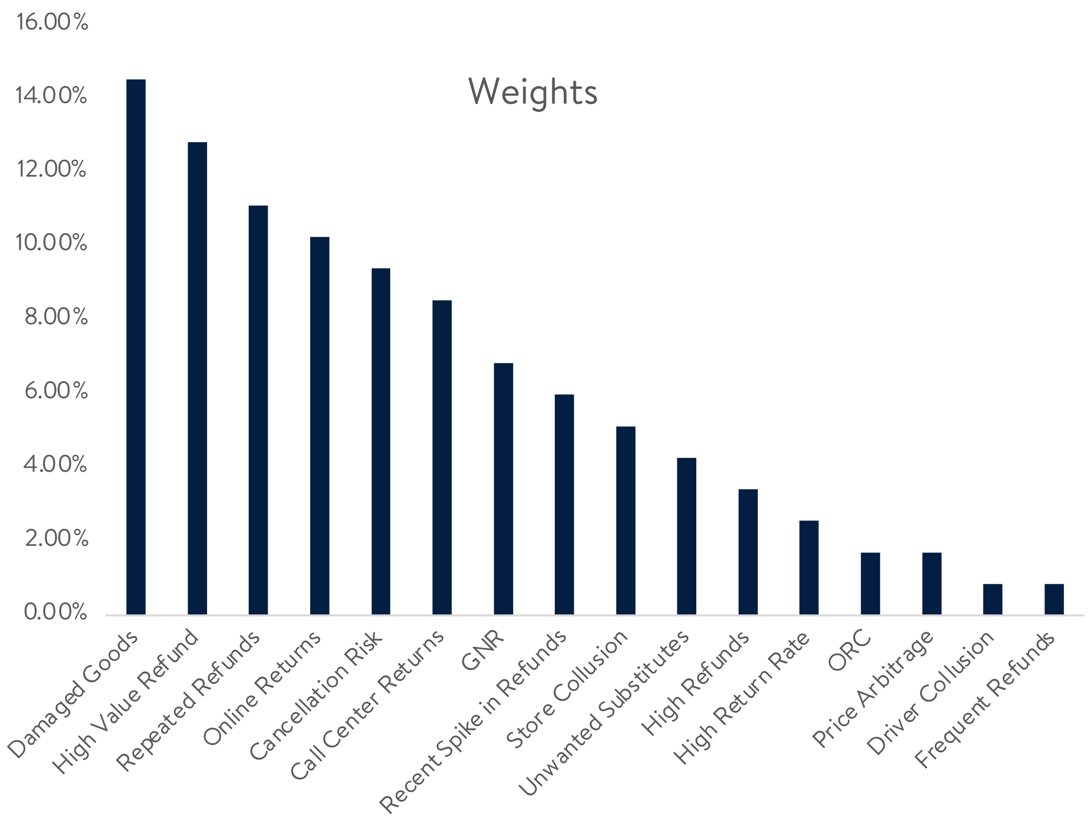
Weighs are based on

1. Precision
2. Suspension propensity Importance

04

Risk Buckets and Reasons

Risk buckets are dynamically chosen from the Risk Scores. Features contributing significantly to the score are the risk reasons.

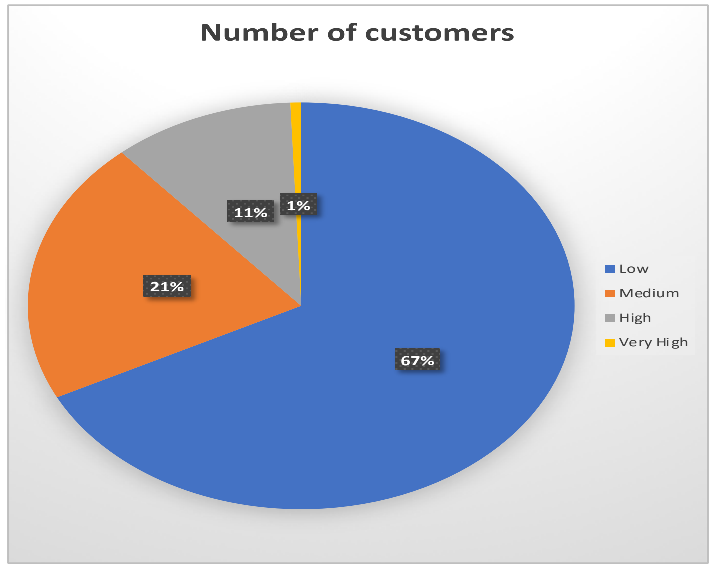


Data Preparation
- Unbalanced Data Processing
- Split into Train, Validation, Test

Model
- XgBoost model training
- Hyper-parameter Tuning
- Maximize Test AUPCR

Scoring
- Weights proportional to Variable importance
- Suspension propensity is the fitted probability of positive class

Metric weights are inversely proportional to their precision



Thank you !

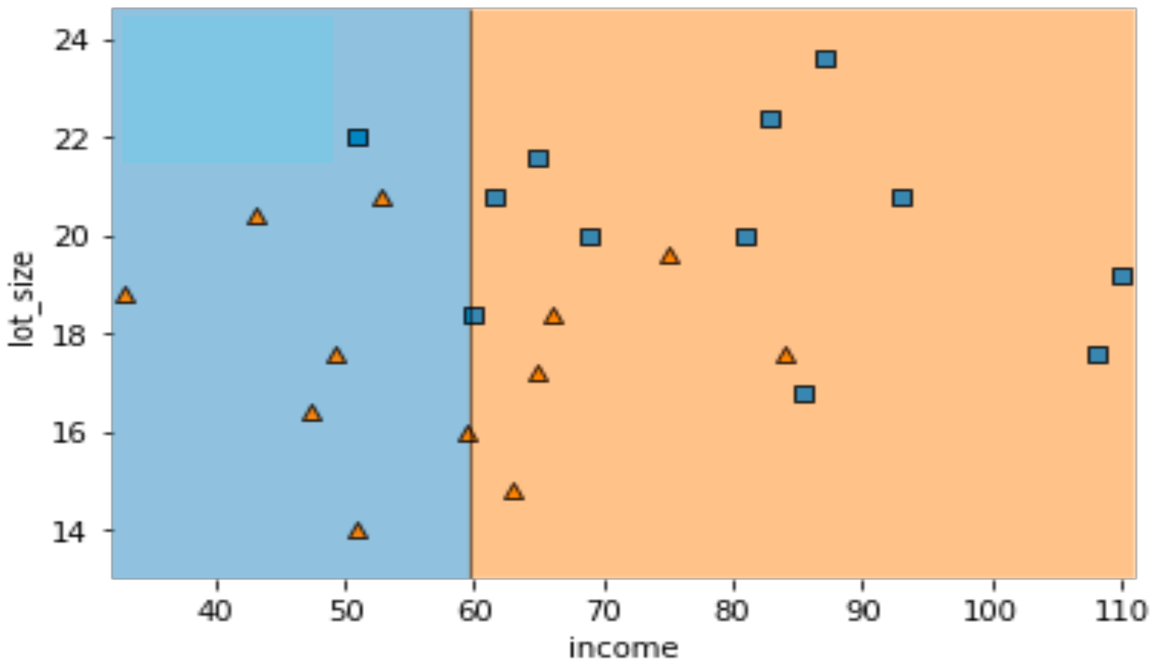
Appendix:

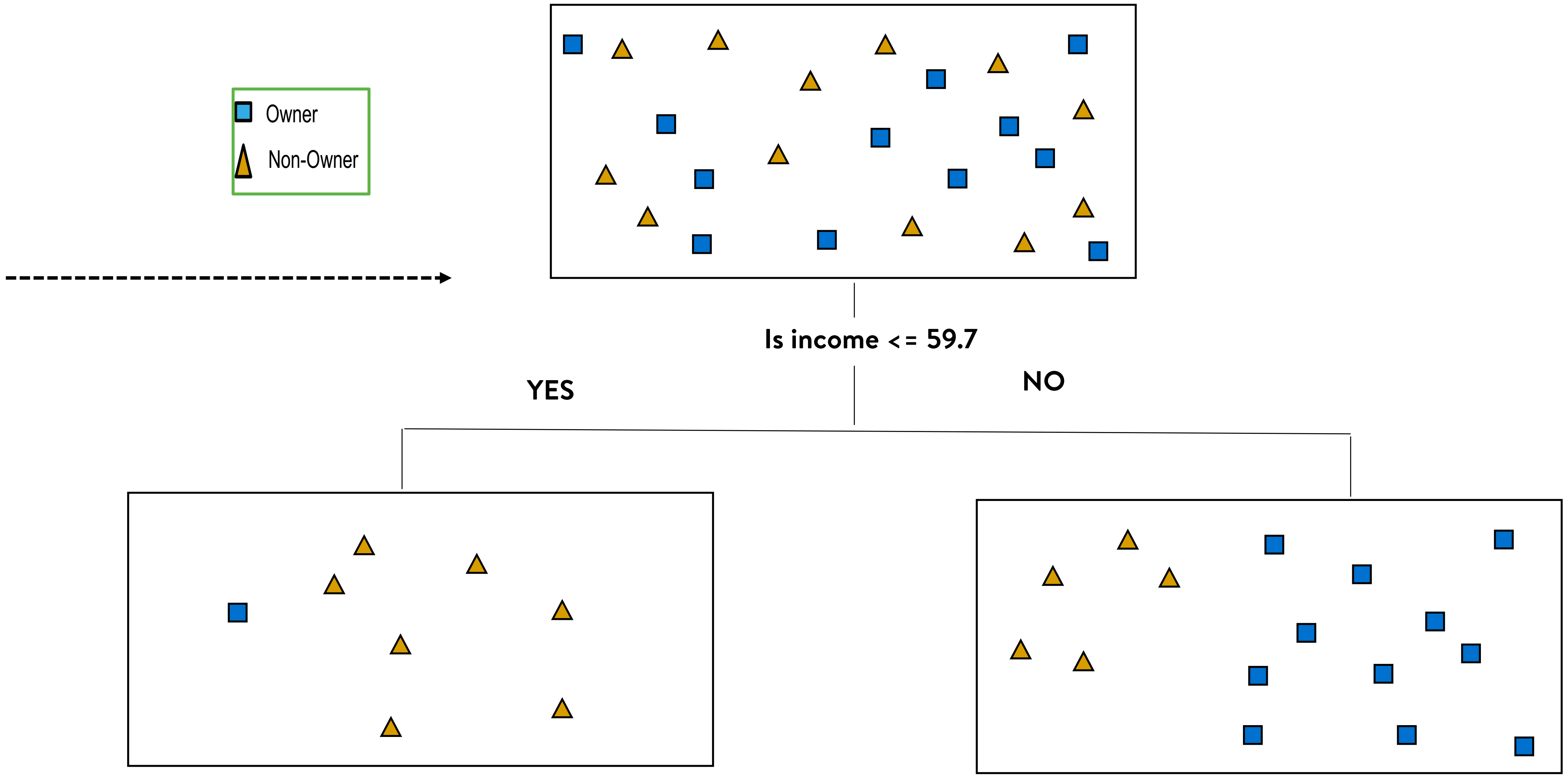
Choosing the best split- alt visual

Walmart

Tree stump – A simple tree with one split

Decision node





For income<=59.7 bucket there are:

- 7 Non-owners
- 1 Owner

Proportion of Non-owner is higher than .5. All observations classified as **non-owners**

For income > 59.7 bucket there are:

- 5 Non-owners
- 11 Owners

Proportion of Owner is higher than .5. All observations classified as **owners**

Splitting into training & test set – alt visual

DATA

This split happens randomly – that is, the general characteristics* of the predictors and the target variable remains the same across both these sets.

TRAIN (70%)	TEST (30%)

TRAIN (70%):

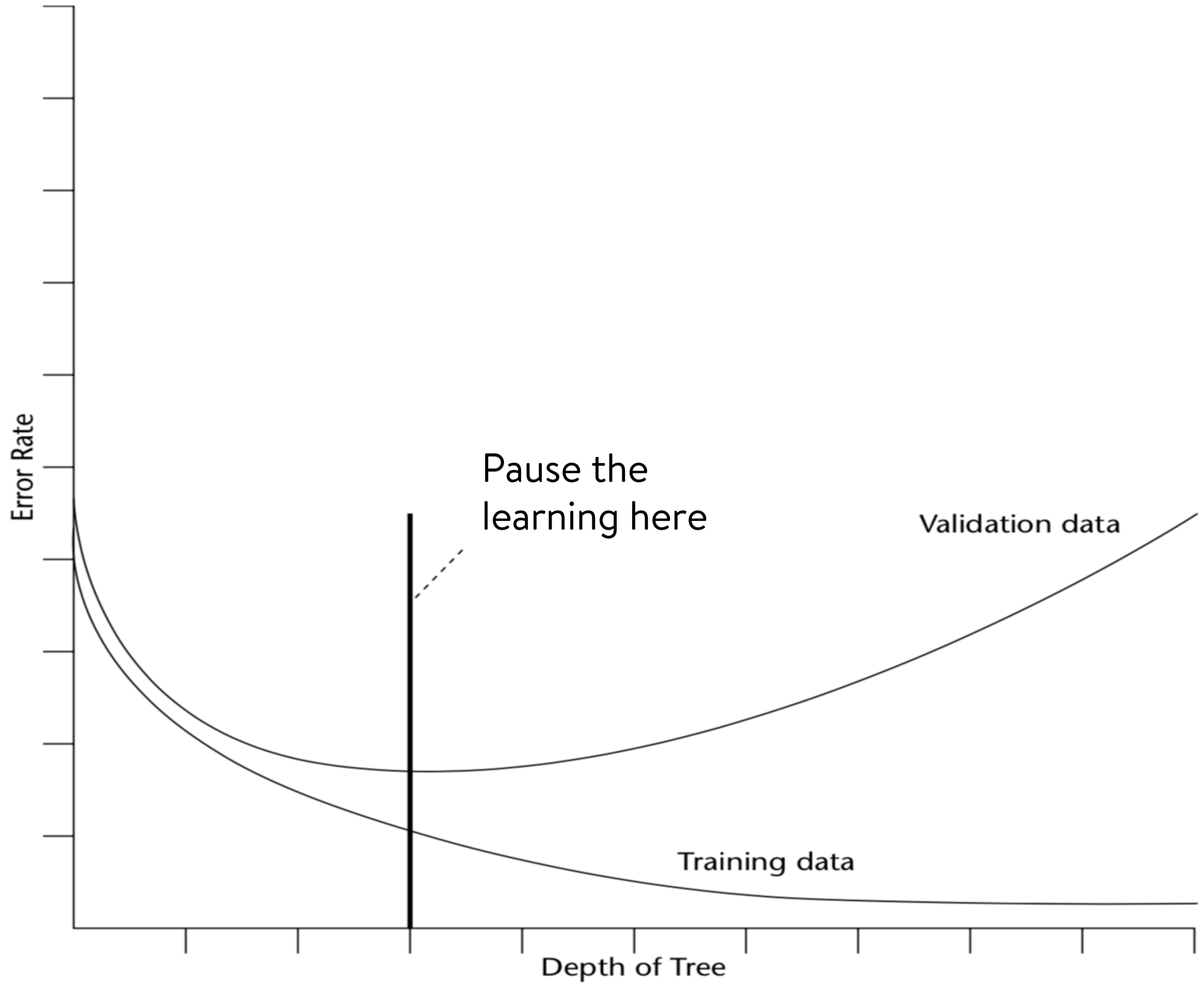
- Calibrate model
- Evaluate model on training data

TEST (30%):

- How does the model fare in the in the wild
- Does the model generalize

Overfitting - The scenario in general

In general a trade-off : Cost = Error + **cost complexity** * number of leaves in a tree



*Depiction from 'Data mining techniques for marketing, sales, CRM', 3rd ed – Berry et al

Architecture Diagram for Risk Score Model

Walmart