Predictive Analytics, Classification, and Decision Trees

Data Science Dojo

Agenda

- Introduction to predictive analytics
- Introduction to classification
- Decision Tree Classifier
- Hands-on Lab: Building a decision tree classifier using R

INTRODUCTION TO PREDICTIVE ANALYTICS

Family and Personal Life

- Location
  - Microsoft and Nokia predict future location based on cellular phone and location data
- Life Events
  - Target predicts customer pregnancy
- Divorce and infidelity
  - University and clinical researchers can predict this as well!

Friendship and Love

- Friendship and connection
  - Facebook and LinkedIn predict your personal connections
- Love
  - Every dating site tries to predict potential matches
  - OkCupid tracks which message content is most likely to elicit a response

Telcos, Retail, and More

- FedEx
  - Predicts defection to a competitor with 65-90% accuracy
- Amazon
  - 35% sales come from product recommendation

Social Networks

Face recognition in Facebook posts

Performed better than humans

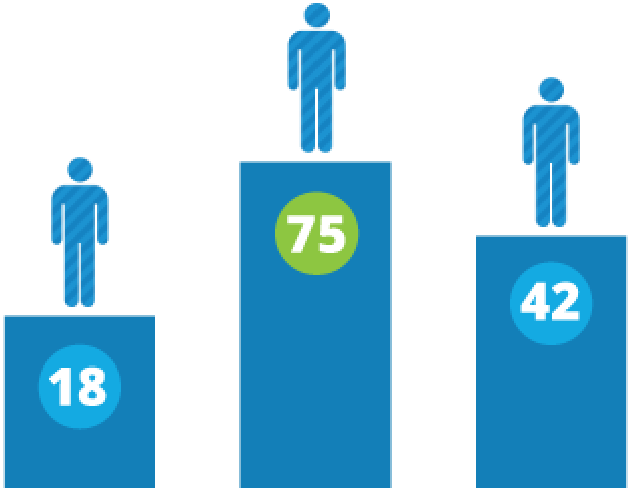
97% accuracy in recognizing faces

Customer Churn

Keep customers before they leave

Advanced detect allows for customer retention strategies to be employed

36% reduction in customer turnover

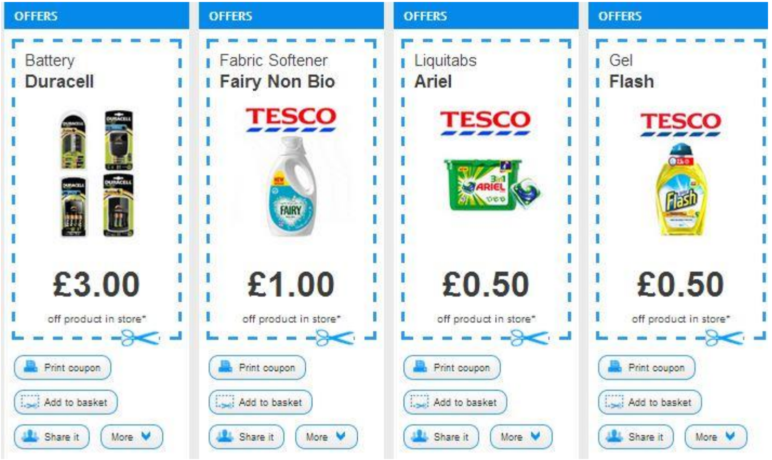


Lead Scoring

Pin-pointing which leads to nurture

Allows sales resources to be more focused towards leads with higher probable conversions

40% reduction in customer calls



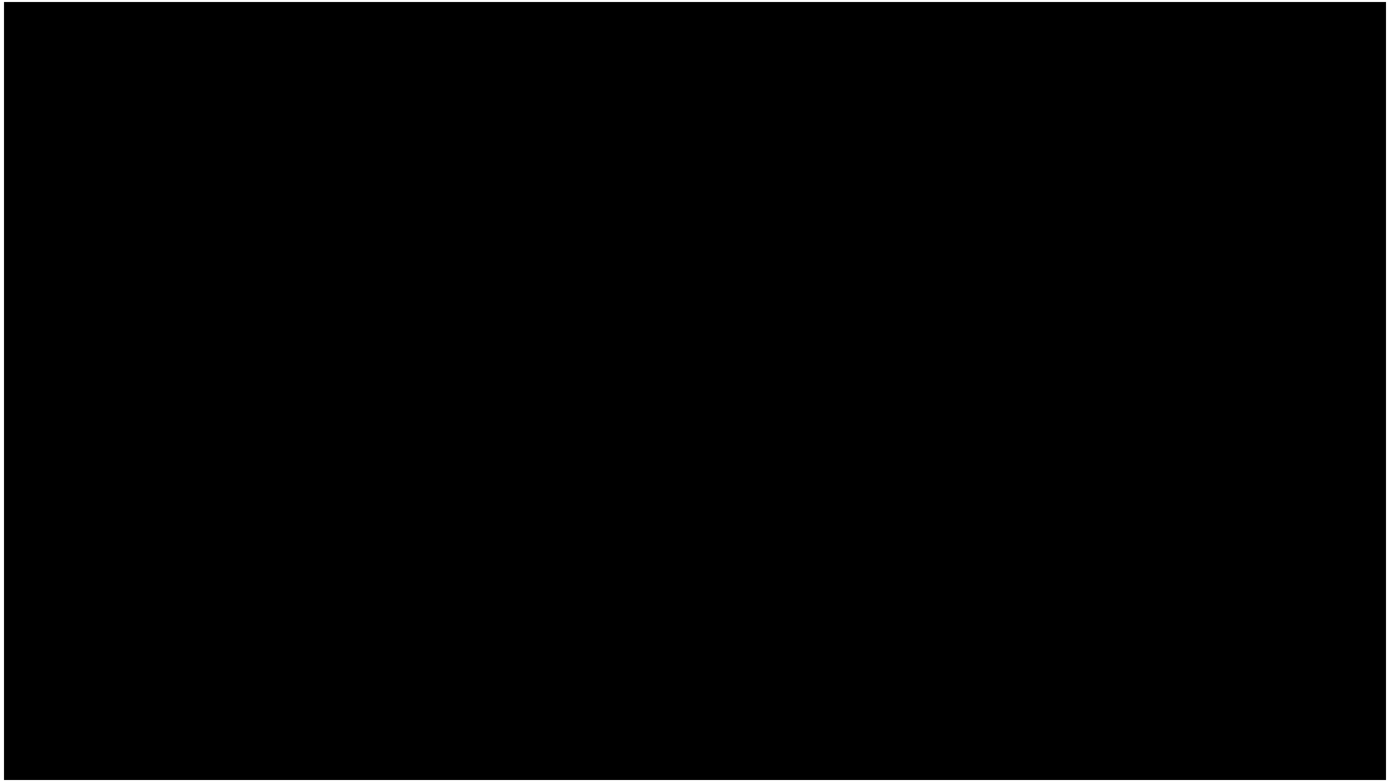
3.6x more coupons than before are sent out

Coupon Redemption

Knowing the liability to bottom-line

Most coupons go un-redeemed. Knowing exactly how many allows you to plan more coupons by having less future liability.

Even In Law Enforcement….

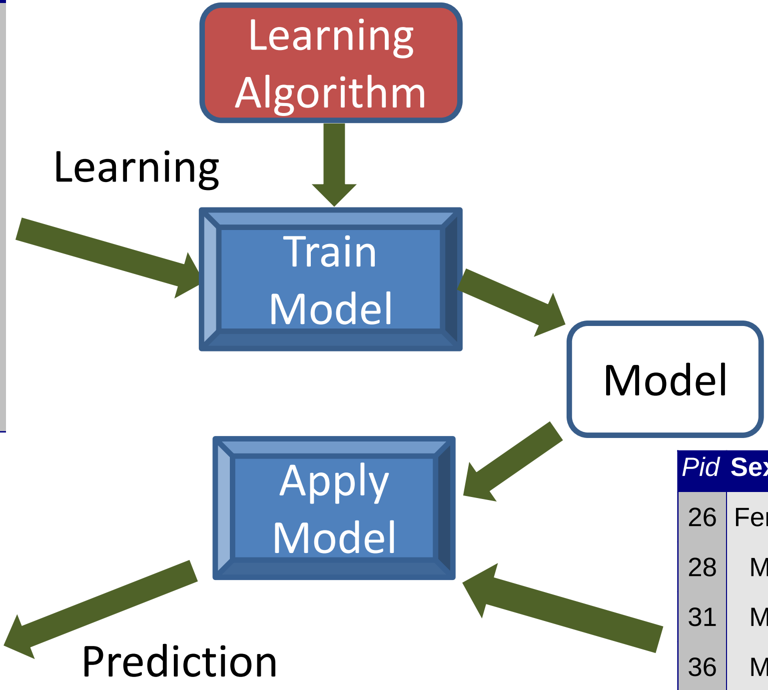


INTRODUCTION TO CLASSIFICATION

Supervised Learning

Pid	Sex	Age	Pclass	Survived
2	Female	38	1	Yes
3	Female	26	3	Yes
5	Male	35	3	No
7	Male	54	1	No
13	Male	20	3	No
14	Male	39	3	No
21	Male	35	2	No
24	Male	28	1	Yes
34	Male	66	1	No
54	Female	29	2	Yes

Training Set



Test Set

Pid	Sex	Age	Pclass	Survived
26	Female	38	3	?
28	Male	19	1	?
31	Male	40	1	?
36	Male	42	1	?
71	Male	32	2	?

Pid	Survived
26	Yes
28	Yes
31	No
36	No
71	No

The Classification Task

- Given a collection of records (training set)
  - Two attribute types: **predictors** and **class**
  - Find a model to map predictor set to class
  - Class is
    - Categorical
    - Nominal (almost always)

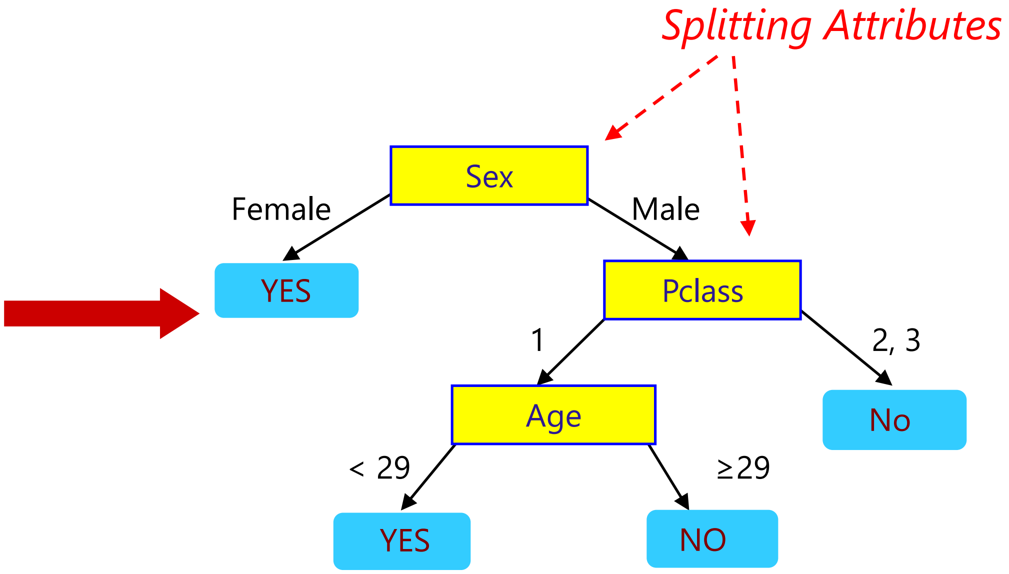
The Classification Task

- Goal: Assign new records a correct class
  - **Training set** used to create model
  - **Test set** used to check
  - Predict test set classes to assess correctness
  - Split data into training and test sets
    - 70/30, 60/40, 50/50

DECISION TREE LEARNING

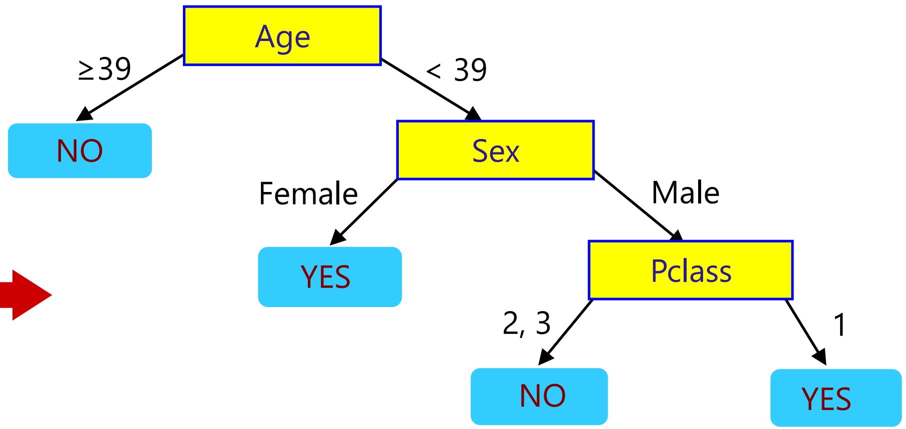
Decision Tree Learning

Pid	Sex	Age	Pclass	Survived
2	Female	38	1	Yes
3	Female	26	3	Yes
5	Male	35	3	No
7	Male	54	1	No
13	Male	20	3	No
14	Male	39	3	No
21	Male	35	2	No
24	Male	28	1	Yes
34	Male	66	1	No
54	Female	29	2	Yes



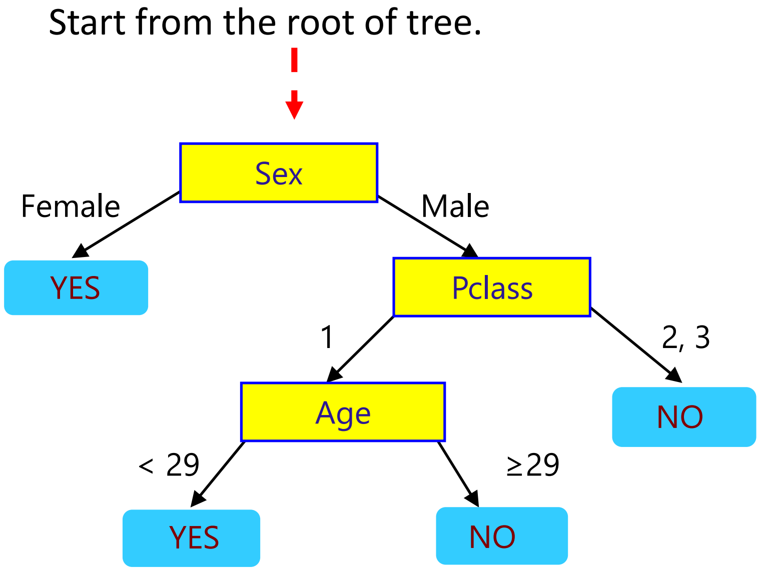
A Different Decision Tree

Pid	Sex	Age	Pclass	Survived
2	Female	38	1	Yes
3	Female	26	3	Yes
5	Male	35	3	No
7	Male	54	1	No
13	Male	20	3	No
14	Male	39	3	No
21	Male	35	2	No
24	Male	28	1	Yes
34	Male	66	1	No
54	Female	29	2	Yes



There could be more than one tree that fits the same data!

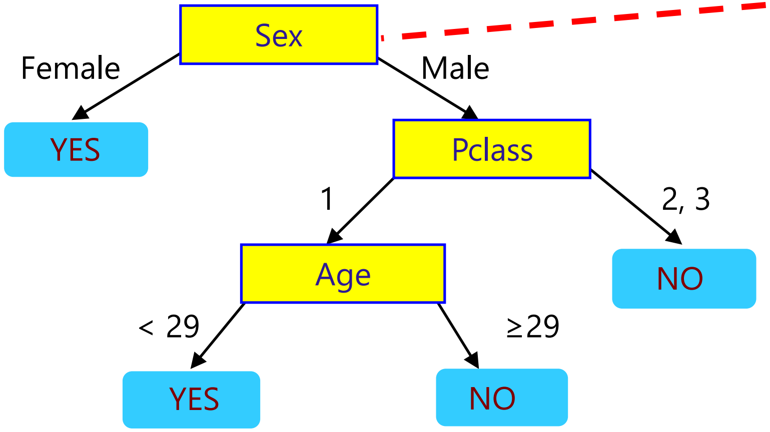
Apply Model to Test Data



Test Data

Sex	Age	Pclass	Survived
Male	32	2	?

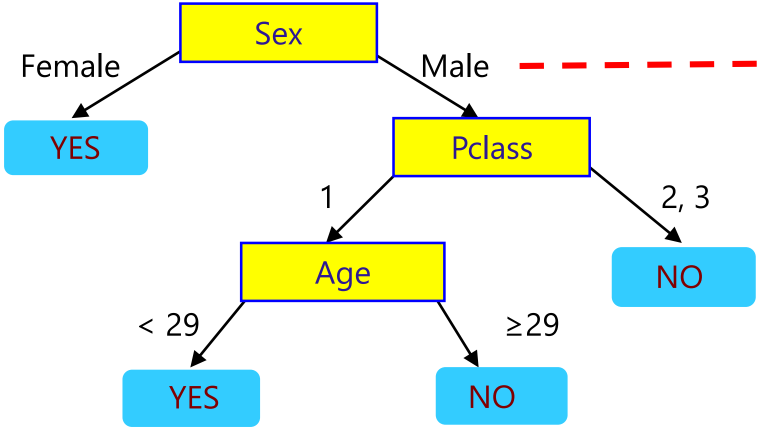
Apply Model to Test Data



Test Data

Sex	Age	Pclass	Survived
Male	32	2	?

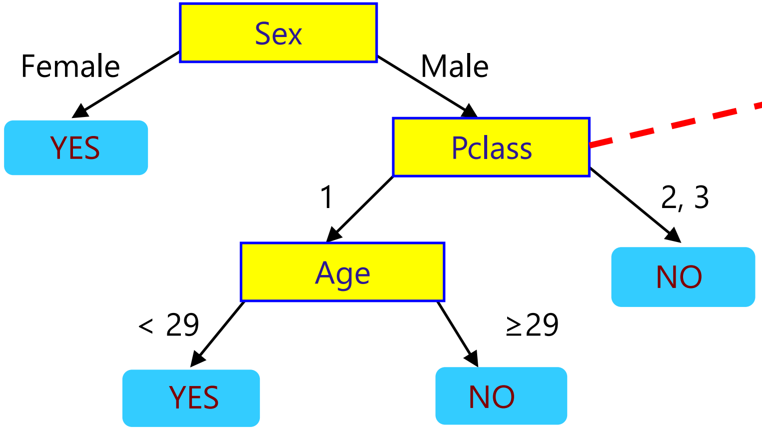
Apply Model to Test Data



Test Data

Sex	Age	Pclass	Survived
Male	32	2	?

Apply Model to Test Data

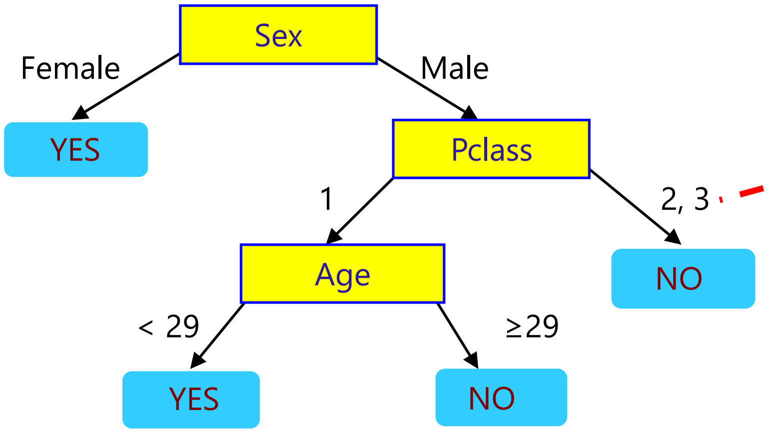


Test Data

Sex	Age	Pclass	Survived
Male	32	2	?

datasciencedojo
data science for everyone

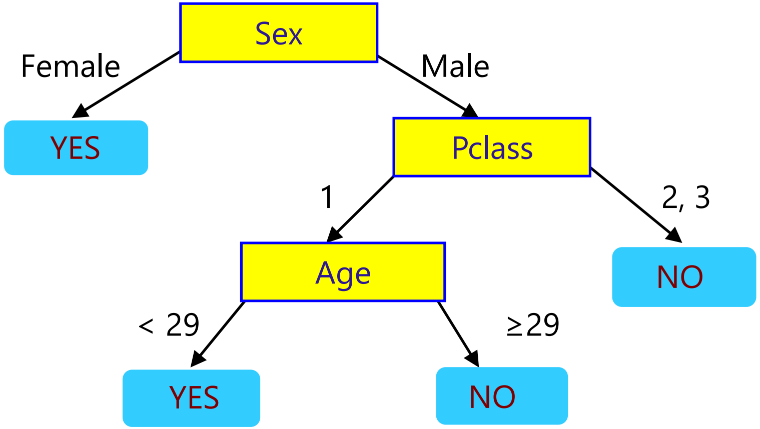
Apply Model to Test Data



Test Data

Sex	Age	Pclass	Survived
Male	32	2	?

Apply Model to Test Data



Test Data

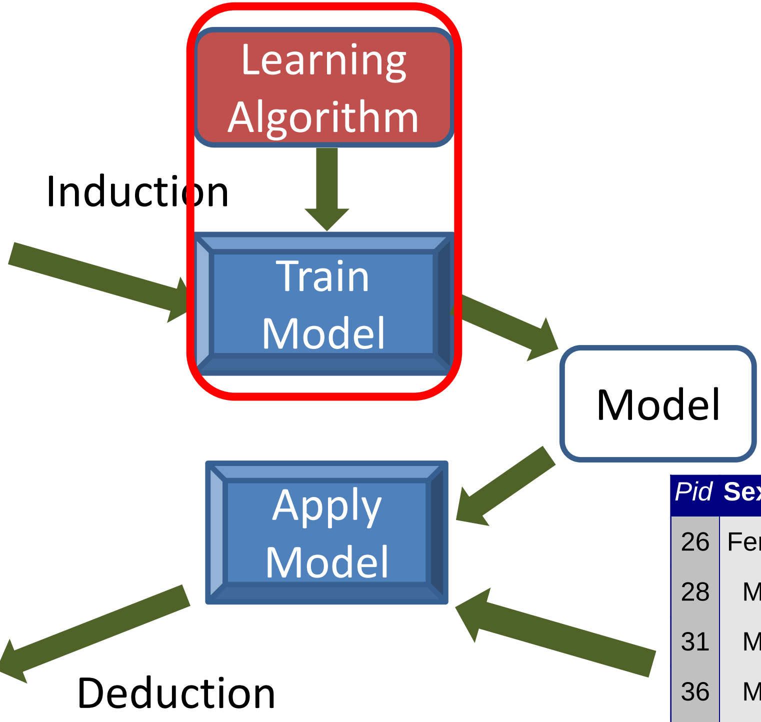
Sex	Age	Pclass	Survived
Male	32	2	?

Survived = “No”

Decision Tree Application

Pid	Sex	Age	Pclass	Survived
2	Female	38	1	Yes
3	Female	26	3	Yes
5	Male	35	3	No
7	Male	54	1	No
13	Male	20	3	No
14	Male	39	3	No
21	Male	35	2	No
24	Male	28	1	Yes
34	Male	66	1	No
54	Female	29	2	Yes

Training Set



Test Set

Pid	Sex	Age	Pclass	Survived
26	Female	38	3	?
28	Male	19	1	?
31	Male	40	1	?
36	Male	42	1	?
71	Male	32	2	?

Pid	Survived
26	Yes
28	Yes
31	No
36	No
71	No

How Do We Get A Tree?

- Exponentially many decision trees are possible
- Finding the optimal tree is infeasible
- Greedy methods that find near-optimal solutions do exist

Tree Induction

- Greedy strategy
  - Split based on attribute test that optimizes a criterion
- Issues
  - How to split the records
    - What attribute test condition?
    - How to determine the best split?
    - When do we stop?

Tree Induction

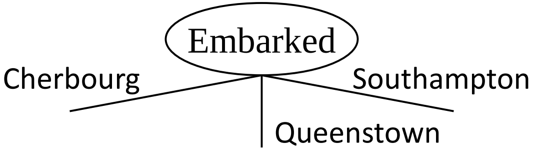
- Greedy strategy
  - Split based attribute test that optimizes a criterion
- Issues
  - How to split the records
    - **What attribute test criterion?**
    - How to determine the best split?
    - When do we stop?

How to Specify Test Condition?

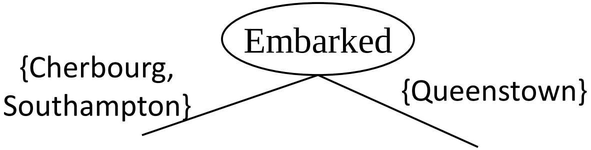
- Attribute types
  - Nominal
  - Ordinal
  - Continuous
- Order of split
  - 2-way split
  - Multi-way split

Splitting: Nominal Attributes

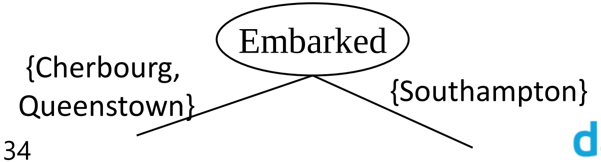
- Multi-way split: As many partitions as distinct values.



- Binary split: Divide values into two subsets. Need to find optimal partitioning.

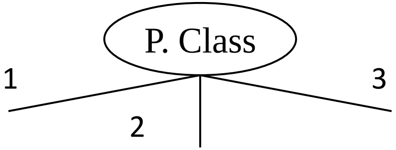


OR

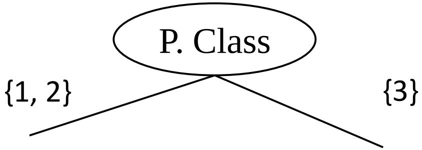


Splitting: Ordinal Attributes

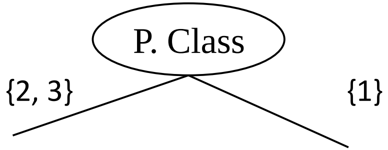
- Multi-way split: As many partitions as distinct values.



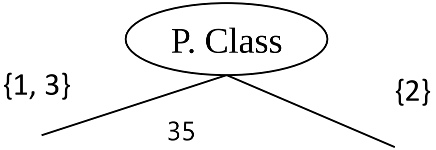
- Binary split: Divides values into two subsets. Need to find optimal partitioning.



OR



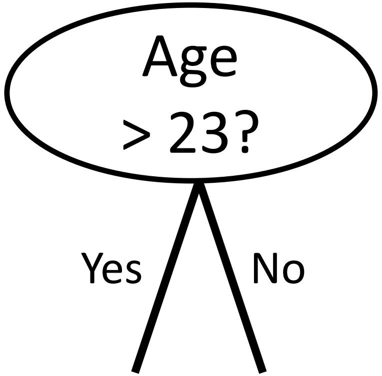
- What about this split?



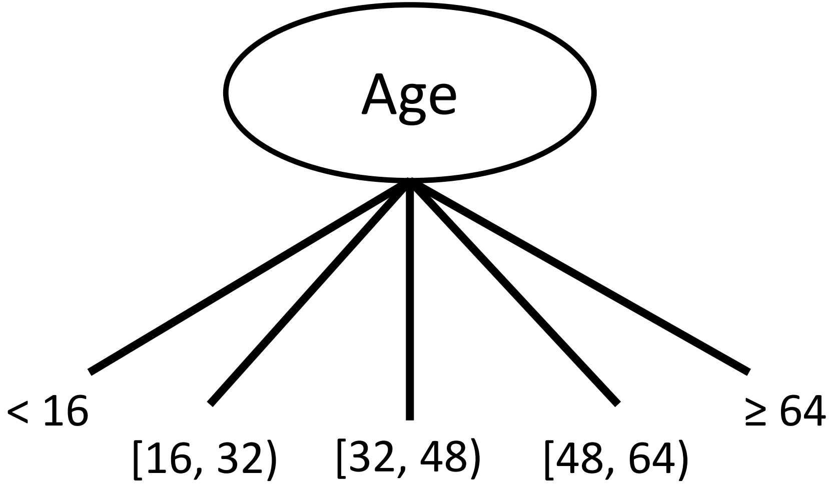
Splitting: Continuous Attributes

- Discretize: transform to ordinal categorical attribute
  - Static – "bucket" once at the beginning
  - Dynamic – "bucket" at each node
    - Equal interval bucketing
    - Equal frequency bucketing (percentiles)
    - Clustering
    - Sweep – Consider all possible splits
      - Usually more computationally intensive

Splitting on Continuous Attributes



Binary Split



Multi-way Split

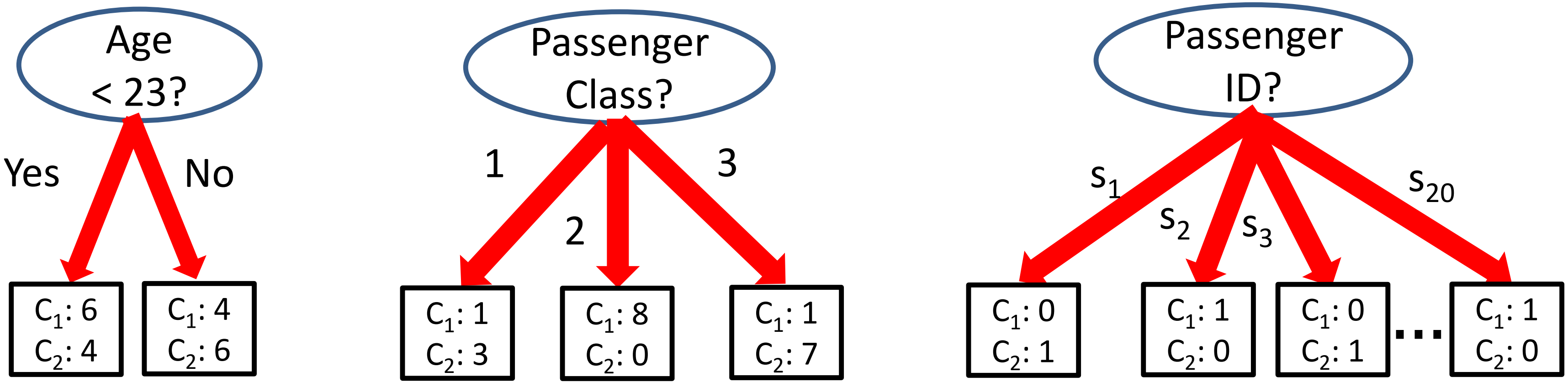
Tree Induction

- Greedy strategy
  - Split based attribute test that optimizes a criterion
- Issues
  - How to split the records
    - What attribute test criterion?
    - **How to determine the best split?**
    - When do we stop?

What is The Best Split?

C_1: Dead
C_2: Survived

Before Splitting: 10 records of class 1, 10 records of class 2



Which test condition is the best?

C_1: Dead
C_2: Survived

What is The Best Split?

- Greedy approach
  - Homogeneous class distribution preferred
- Need a measure of **node impurity**

C_1: 5 C_2: 5	C_1: 9 C_2: 1
Non-homogeneous	Homogeneous
High degree of impurity	Low degree of impurity

Measures of Node Impurity

- Gini Index
- Entropy
- Misclassification error

Impurity Measure: GINI

C_1: Dead
C_2: Survived

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

- p(j | t) is the relative frequency of class j at node t

- Maximum ($1 - 1/n_c$) when records are equally distributed among all classes, implying least interesting information
  - n_c=number of classes
- Minimum (0.0) when all records belong to one class, implying most interesting information

C_1	**0**
C_2	**6**
Gini=0.000	

C_1	**1**
C_2	**5**
Gini=0.278	

C_1	**2**
C_2	**4**
Gini=0.444	

C_1	**3**
C_2	**3**
Gini=0.500	

Impurity Measure: GINI

C_1: Dead
C_2: Survived

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

C_1	**0**
C_2	**6**

P(C1) = 0/6 = 0   P(C2) = 6/6 = 1

Gini = 1 – $P(C1)^2$ – $P(C2)^2$ = 1 – 0 – 1 = 0

C_1	**1**
C_2	**5**

P(C1) = 1/6   P(C2) = 5/6

Gini = 1 – $(1/6)^2$ – $(5/6)^2$ = 0.278

C_1	**2**
C_2	**4**

P(C1) = 2/6   P(C2) = 4/6

Gini = 1 – $(2/6)^2$ – $(4/6)^2$ = 0.444

Impurity Measure: GINI

- When a node p is split into k partitions (children), the quality of split is computed as:

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

where

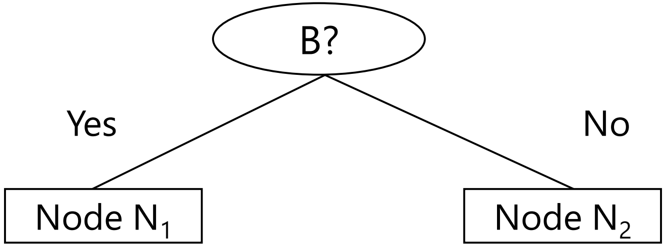
n_i = number of records at child i

n = number of records at node p

C_1: Dead
C_2: Survived

Impurity Measure: GINI

- Split data into two partitions
- Partition measurements are weighted
  - Larger and purer partitions are sought after



	Parent
C_1	**6**
C_2	**6**
Gini = 0.500	

Gini(N_1)
$= 1 - (5/7)^2 - (2/7)^2$
$= 0.408$

Gini(N_2)
$= 1 - (1/5)^2 - (4/5)^2$
$= 0.320$

	N_1	N_2
C_1	**5**	**1**
C_2	**2**	**4**
Gini=0.371		

Gini(B?, Parent)
$= 7/12 * 0.408 +$
$5/12 * 0.320$
$= 0.371$

Impurity Measure: Entropy

$$Entropy(t) = -\sum_j p(j|t)\log_2(p(j|t))$$

- $p(j|t)$ is the relative frequency of class j at node t

- Maximum: records equally distributed
- Minimum: all records belong to one class

Impurity Measure: Entropy

C_1: Dead
C_2: Survived

$$Entropy(t) = -\sum_j p(j|t)\log_2 p(j|t)$$

C_1	**0**
C_2	**6**

P(C1) = 0/6 = 0   P(C2) = 6/6 = 1

Entropy = – 0 log 0 – 1 log 1 = – 0 – 0 = 0

C_1	**1**
C_2	**5**

P(C1) = 1/6   P(C2) = 5/6

Entropy = – (1/6) $\log_2$ (1/6) – (5/6) $\log_2$ (5/6) = 0.65

C_1	**2**
C_2	**4**

P(C1) = 2/6   P(C2) = 4/6

Entropy = – (2/6) $\log_2$ (2/6) – (4/6) $\log_2$ (4/6) = 0.92

Impurity Measure: Information

$$GAIN_{split} = Entropy(p) - \left(\sum_{i=1}^{k} \frac{n_i}{n} Entropy(i) \right)$$

- Node p is split into k partitions
- n_i is number of records in partition i

- Measures reduction in entropy
- Choose split that maximizes GAIN
- Tends to prefer splits with large number of partitions

Impurity Measure: Classification Error

$$Error(t) = 1 - \max_i P(i \mid t)$$

- Maximum: records are equally distributed
- Minimum: all records belong to one class
- Similar to information gain
  - Less sensitive for > 2 or 3 splits
  - Less prone to overfitting

Impurity Measure: Classification Error

C_1: Dead
C_2: Survived

$$Error(t) = 1 - \max_i P(i \mid t)$$

C_1	**0**
C_2	**6**

P(C1) = 0/6 = 0   P(C2) = 6/6 = 1

Error = 1 – max (0, 1) = 1 – 1 = 0

C_1	**1**
C_2	**5**

P(C1) = 1/6   P(C2) = 5/6

Error = 1 – max (1/6, 5/6) = 1 – 5/6 = 1/6

C_1	**2**
C_2	**4**

P(C1) = 2/6   P(C2) = 4/6

Error = 1 – max (2/6, 4/6) = 1 – 4/6 = 1/3

Tree Induction

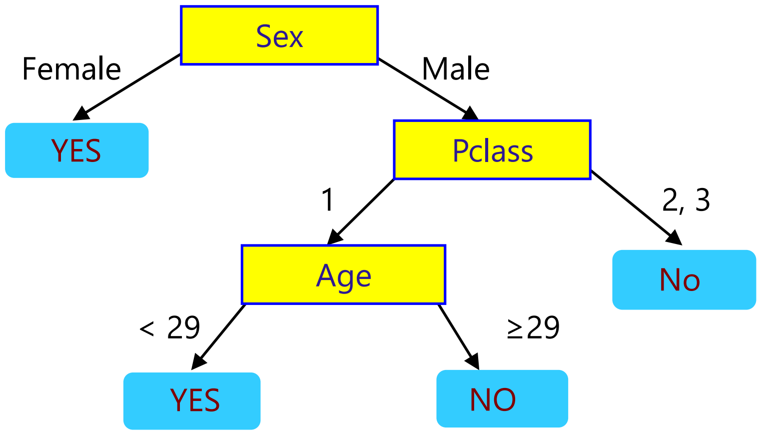
- Greedy strategy
  - Split based attribute test that optimizes a criterion
- Issues
  - How to split the records
    - What attribute test criterion?
    - How to determine the best split?
    - **When do we stop?**

Sample Stopping Criteria

- All the records belong to the same class
- All the records have similar attribute values
- Fixed termination or pruning
  - Number of Levels
  - Number of Leaf Nodes
  - Minimum samples per leaf node

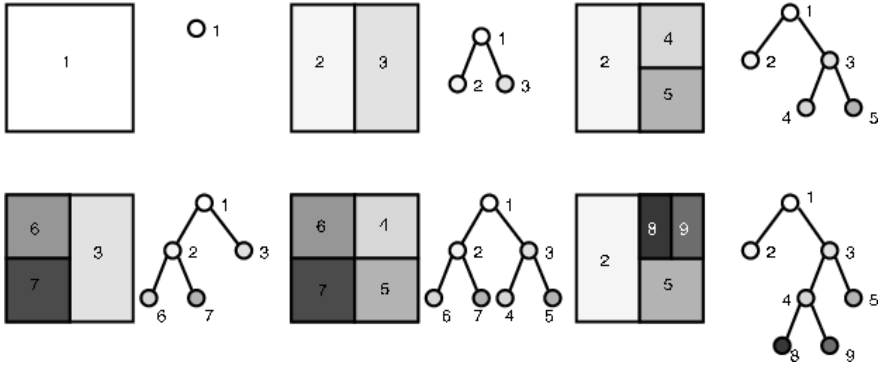
Decision Trees - PROS

- Intuitive
  - Easy interpretation for small trees
- Non parametric
  - Incorporate both numeric and categorical attributes
- Fast
  - Once rules are developed, prediction is rapid
- Robust to outliers



Decision Trees - CONS

- Overfitting
  - Must be trained with great care
- Rectangular Classification
  - Recursive partitioning of data may not capture complex relationships



QUESTIONS

HANDS-ON LAB: BUILDING A DECISION TREE CLASSIFIER USING R