ACIT4830 – Special Robotics and Control Subject

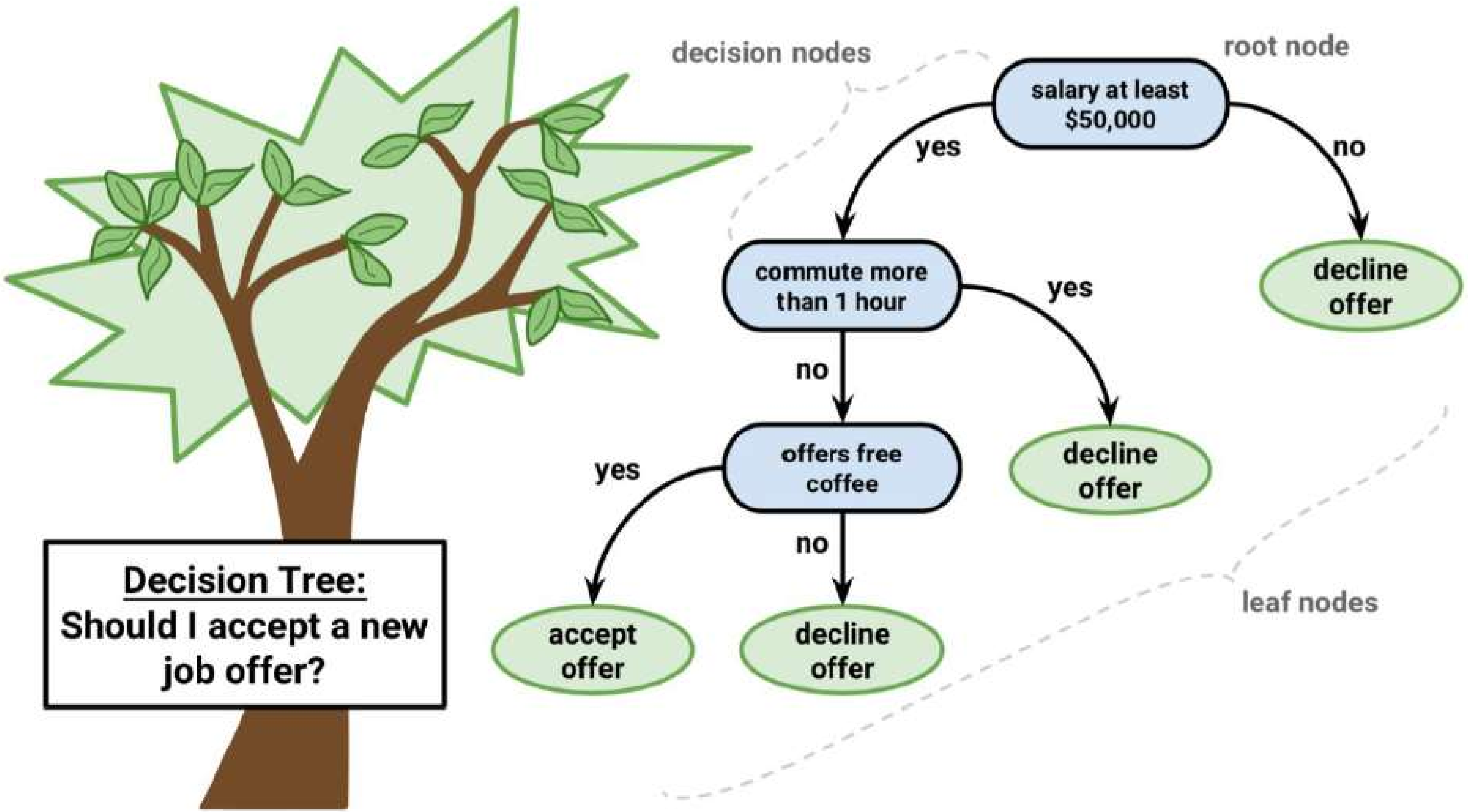
Topic3 – Classification using Decision Trees and Rules

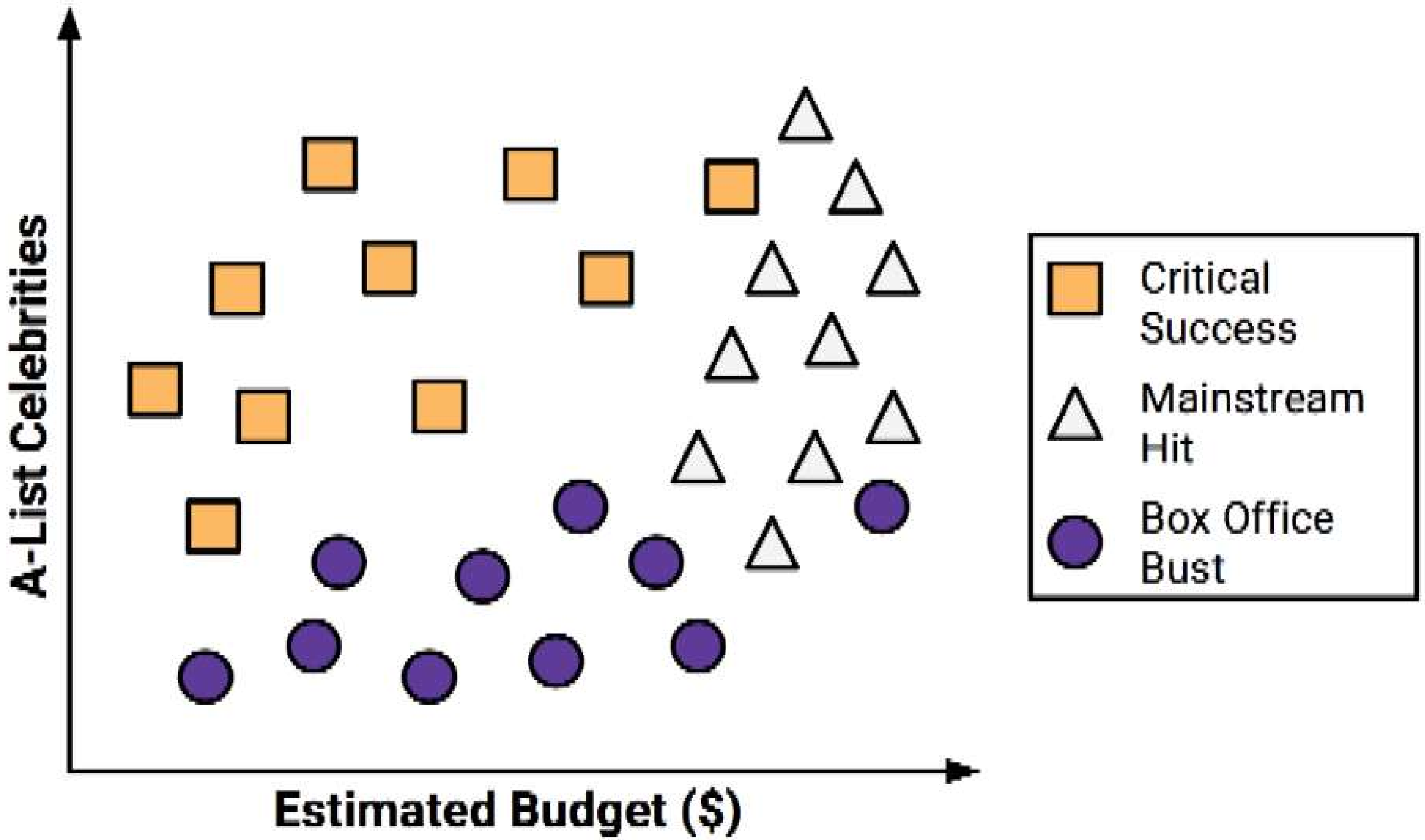
Evi Zouganeli
OsloMet – Oslo Metropolitan University
(evizou@oslomet.no)

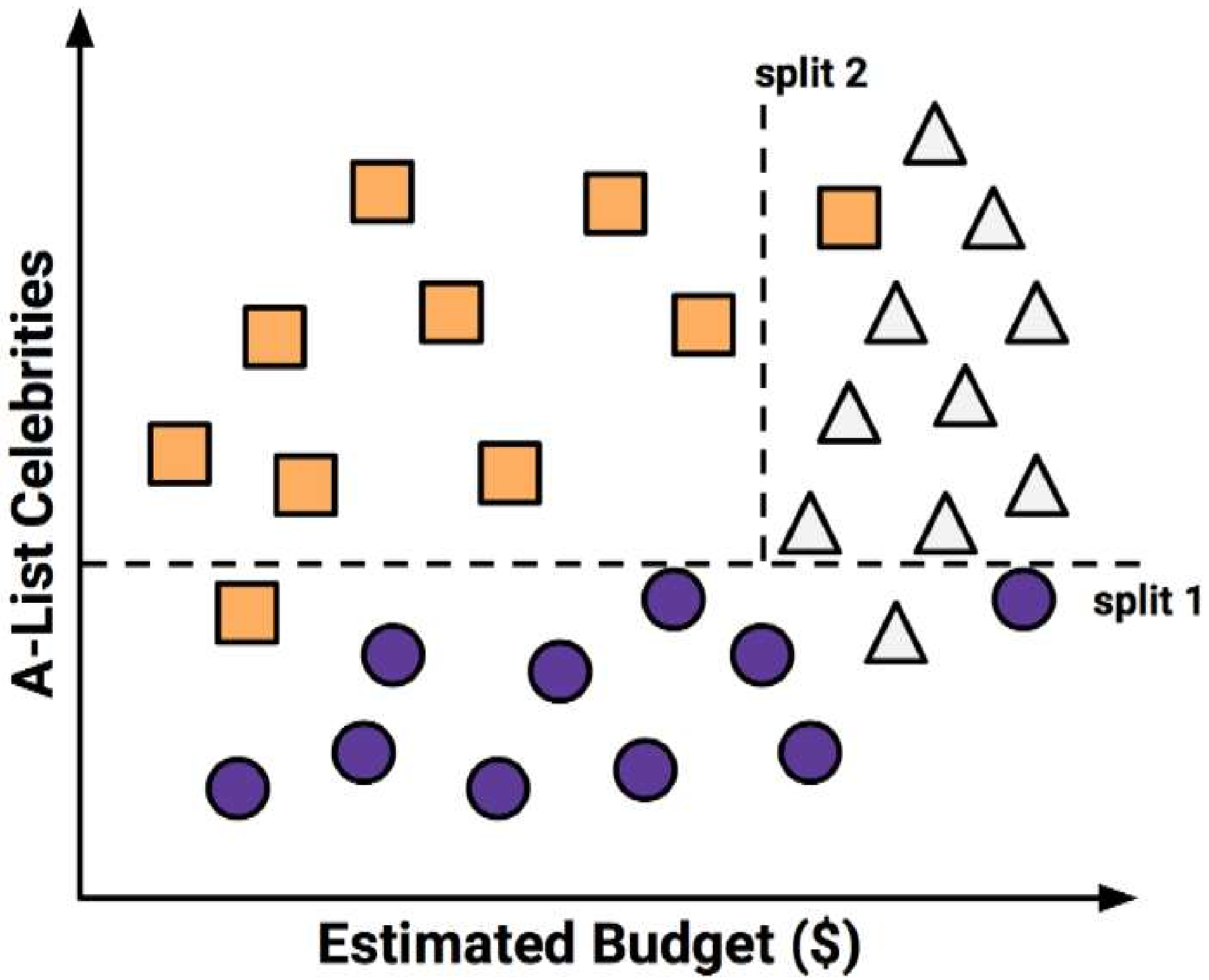
Content

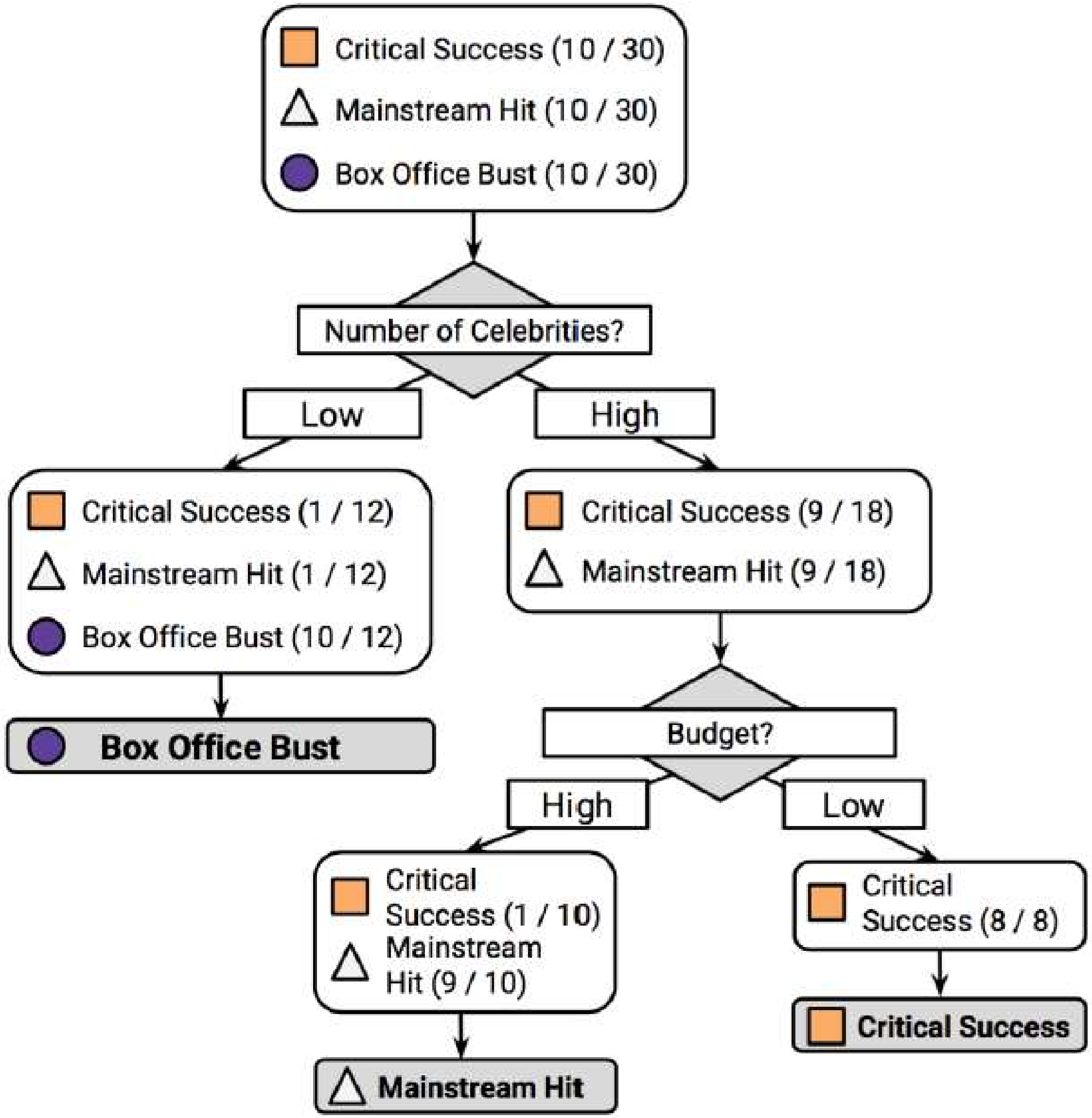
(Chapter 5 Decision Trees)

- Decision Tree Classification
- Choosing the best split
- Making some mistakes cost more than others
- Classification rules


Decision Tree:
Should I accept a new job offer?
decision nodes
root node
salary at least $50,000
yes
no
commute more than 1 hour
yes
decline offer
no
offers free coffee
decline offer
yes
no
accept offer
decline offer
leaf nodes


A-List Celebrities
Estimated Budget ($)
Critical Success
Mainstream Hit
Box Office Bust


split 2
split 1
A-List Celebrities
Estimated Budget ($)

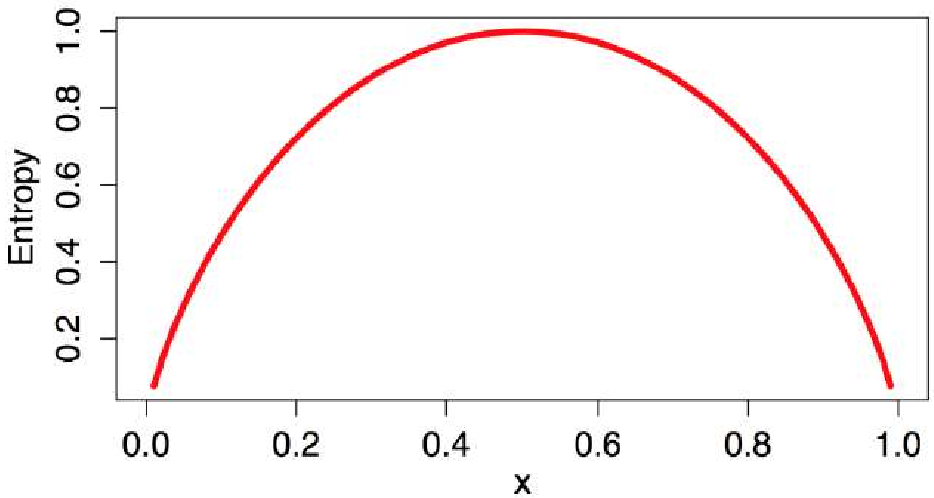

Critical Success (10 / 30)
Mainstream Hit (10 / 30)
Box Office Bust (10 / 30)
Number of Celebrities?
Low
High
Critical Success (1 / 12)
Mainstream Hit (1 / 12)
Box Office Bust (10 / 12)
Box Office Bust
Critical Success (9 / 18)
Mainstream Hit (9 / 18)
Budget?
High
Low
Critical Success (1 / 10)
Mainstream Hit (9 / 10)
Mainstream Hit
Critical Success (8 / 8)
Critical Success

C5.0 algorithm

Strengths	Weaknesses
• An all-purpose classifier that does well on most problems • Highly-automatic learning process can handle numeric or nominal features, missing data • Uses only the most important features • Can be used on data with relatively few training examples or a very large number • Results in a model that can be interpreted without a mathematical background (for relatively small trees) • More efficient than other complex models	• Decision tree models are often biased toward splits on features having a large number of levels • It is easy to overfit or underfit the model • Can have trouble modeling some relationships due to reliance on axis-parallel splits • Small changes in training data can result in large changes to decision logic • Large trees can be difficult to interpret and the decisions they make may seem counterintuitive

Entropy – a measure of randomness, disorder

$$\text{Entropy}(S) = \sum_{i=1}^{c} -p_i \log_2(p_i)$$



$$\text{InfoGain}(F) = \text{Entropy}(S_1) - \text{Entropy}(S_2)$$

= change of Entropy before and after the split of data in segments

Boosting with C5.0

```
> credit_boost10 <- C5.0(credit_train[-17], credit_train$default,
                         trials = 10)
```

Making some mistakes cost more than others

```
> error_cost
     [,1] [,2]
[1,]    0    4
[2,]    1    0
```

OneR algorithm

Strengths	Weaknesses
• Generates a single, easy-to-understand, human-readable rule-of-thumb • Often performs surprisingly well • Can serve as a benchmark for more complex algorithms	• Uses only a single feature • Probably overly simplistic

OneR algorithm

Animal	Travels By	Has Fur	Mammal
Bats	Air	Yes	Yes
Bears	Land	Yes	Yes
Birds	Air	No	No
Cats	Land	Yes	Yes
Dogs	Land	Yes	Yes
Eels	Sea	No	No
Elephants	Land	No	Yes
Fish	Sea	No	No
Frogs	Land	No	No
Insects	Air	No	No
Pigs	Land	No	Yes
Rabbits	Land	Yes	Yes
Rats	Land	Yes	Yes
Rhinos	Land	No	Yes
Sharks	Sea	No	No

Travels By	Predicted	Actual
Air	No	Yes
Air	No	No
Air	No	No
Land	Yes	Yes
Land	Yes	Yes
Land	Yes	Yes
Land	Yes	Yes
Land	Yes	No
Land	Yes	Yes
Land	Yes	Yes
Land	Yes	Yes
Land	Yes	Yes
Sea	No	No
Sea	No	No
Sea	No	No

Rule for Travels By:
Errors = 2 / 15

Has Fur	Predicted	Actual
No	No	No
No	No	No
No	No	Yes
No	No	No
No	No	No
No	No	No
No	No	Yes
No	No	Yes
No	No	No
Yes	Yes	Yes
Yes	Yes	Yes
Yes	Yes	Yes
Yes	Yes	Yes
Yes	Yes	Yes
Yes	Yes	Yes

Rule for Has Fur:
Errors = 3 / 15

Rules from feature Travels by

- If the animal travels by air, it is not a mammal
- If the animal travels by land, it is a mammal
- If the animal travels by sea, it is not a mammal

RIPPER rule-learning algorithm

Strengths	Weaknesses
• Generates easy-to-understand, human-readable rules • Efficient on large and noisy datasets • Generally produces a simpler model than a comparable decision tree	• May result in rules that seem to defy common sense or expert knowledge • Not ideal for working with numeric data • Might not perform as well as more complex models

RIPPER rule-learning algorithm – mushrooms

```
JRIP rules:
============
(odor = foul) => type=poisonous (2160.0/0.0)
(gill_size = narrow) and (gill_color = buff) => type=poisonous
(1152.0/0.0)
(gill_size = narrow) and (odor = pungent) => type=poisonous (256.0/0.0)
(odor = creosote) => type=poisonous (192.0/0.0)
(spore_print_color = green) => type=poisonous (72.0/0.0)
(stalk_surface_below_ring = scaly) and (stalk_surface_above_ring = silky)
=> type=poisonous (68.0/0.0)
(habitat = leaves) and (cap_color = white) => type=poisonous (8.0/0.0)
(stalk_color_above_ring = yellow) => type=poisonous (8.0/0.0)
 => type=edible (4208.0/0.0)
Number of Rules : 9
```

Algorithms in hands-on:

- C5.0
- OneR
- RIPPER

Also including:

- Boosting
- Weighting errors