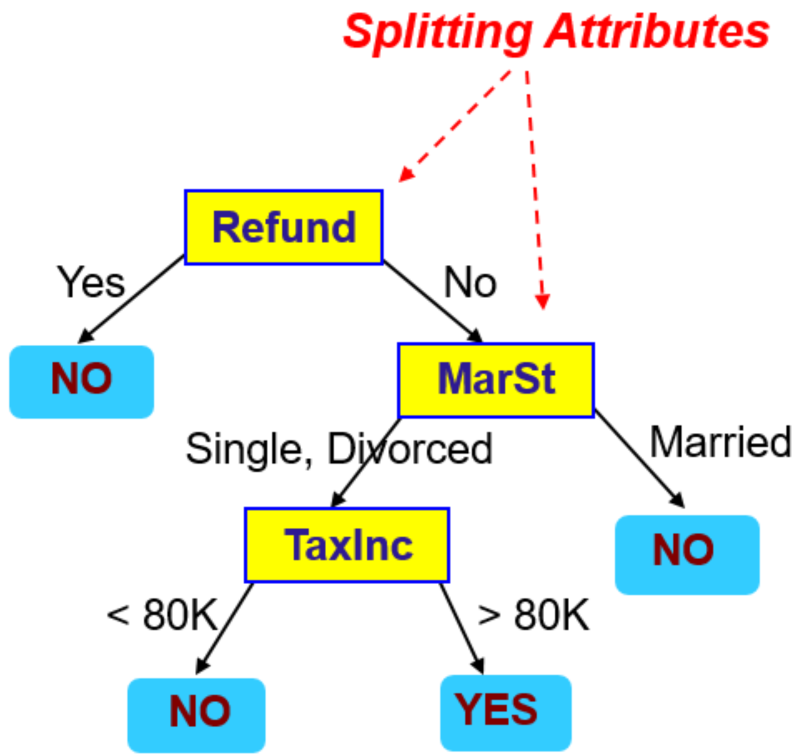
DECISION TREES

Decision Tree

categorical categorical continuous class

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

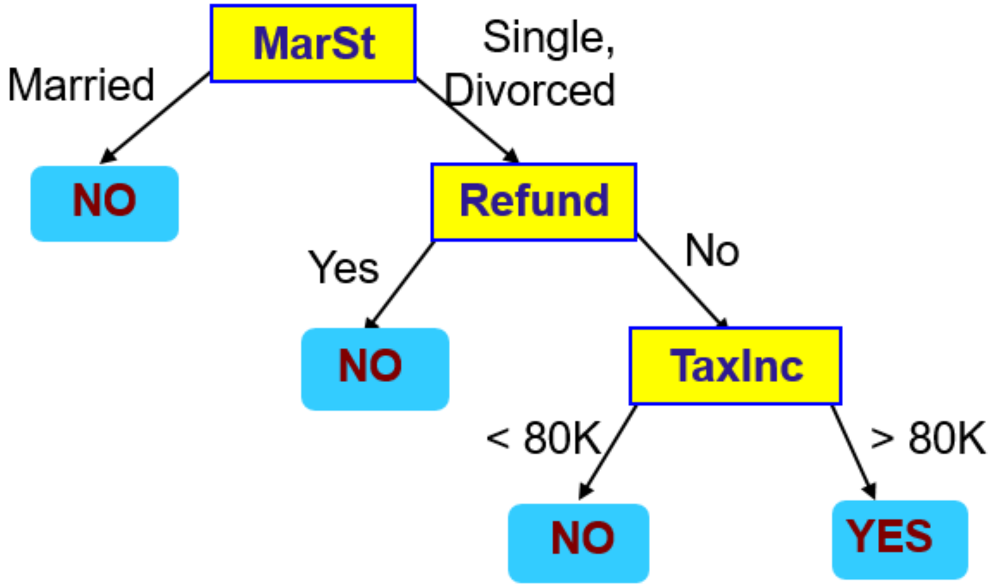
Training Data



Model: Decision Tree

Another Tree

	categorical	categorical	continuous	class
Tid	**Refund**	**Marital Status**	**Taxable Income**	**Cheat**
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes



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

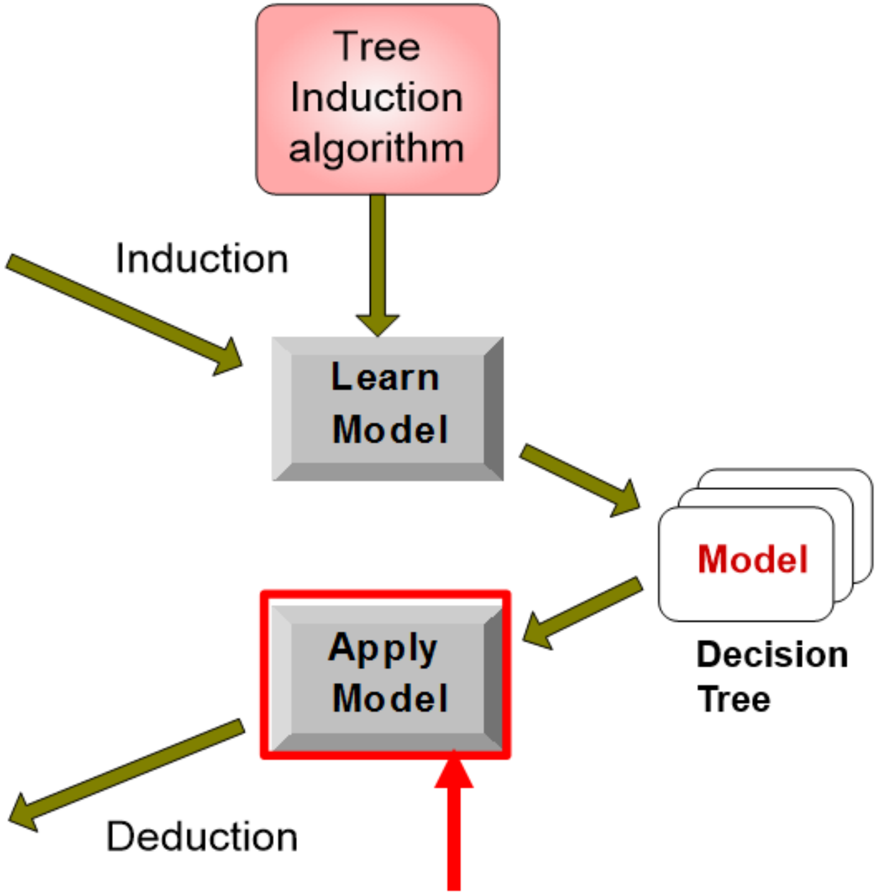
Use of Decision Tree

Tid	Attrib1	Attrib2	Attrib3	Class
1	Yes	Large	125K	No
2	No	Medium	100K	No
3	No	Small	70K	No
4	Yes	Medium	120K	No
5	No	Large	95K	Yes
6	No	Medium	60K	No
7	Yes	Large	220K	No
8	No	Small	85K	Yes
9	No	Medium	75K	No
10	No	Small	90K	Yes

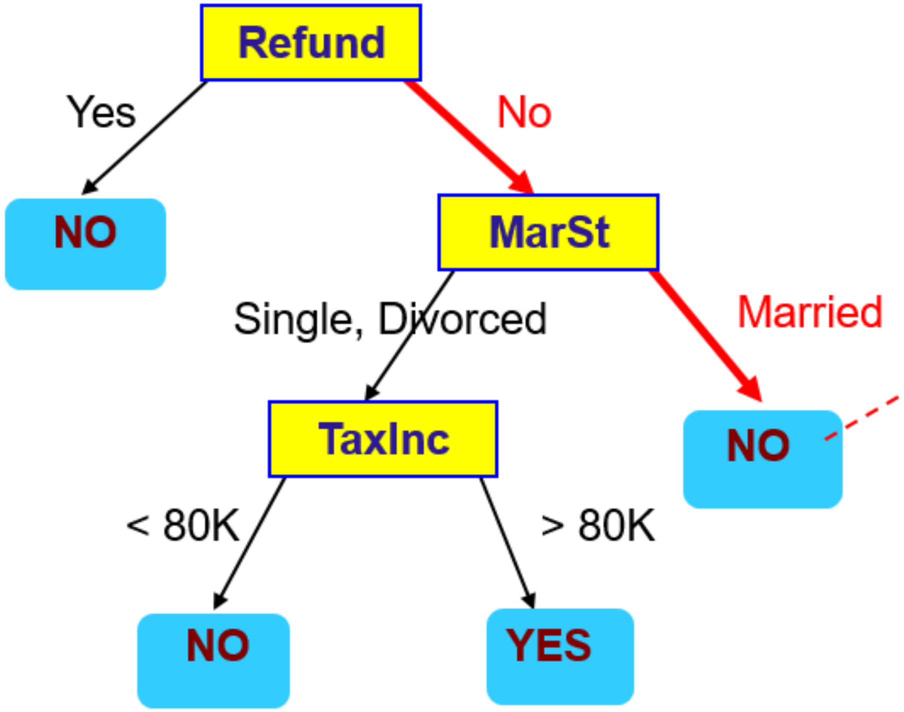
Training Set

Tid	Attrib1	Attrib2	Attrib3	Class
11	No	Small	55K	?
12	Yes	Medium	80K	?
13	Yes	Large	110K	?
14	No	Small	95K	?
15	No	Large	67K	?

Test Set



Predict



Test Data

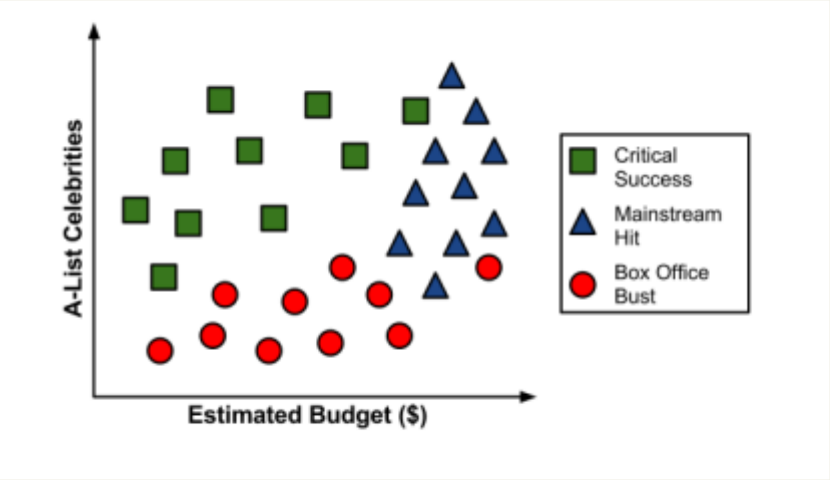
Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

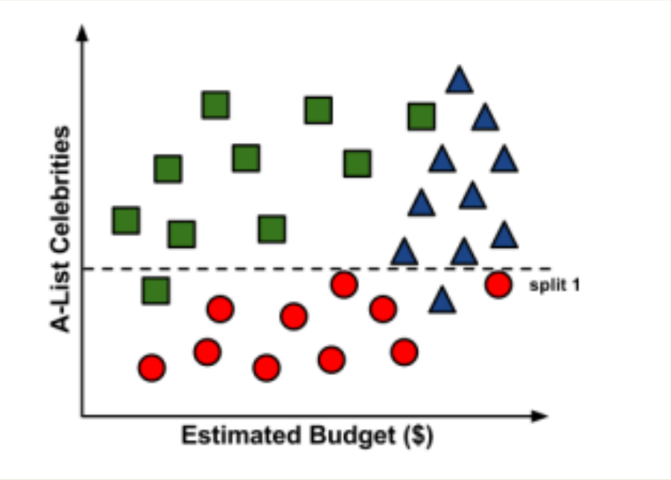
Assign Cheat to "No"

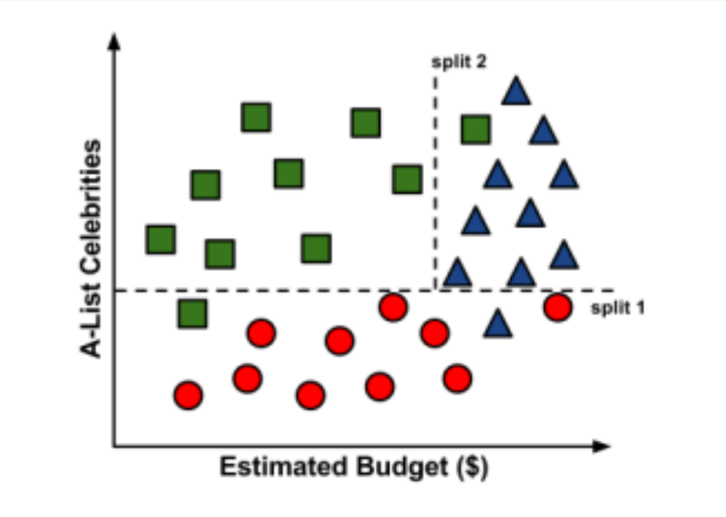
Divide and Conquer

- Decision tress are built using a heuristic call recursive partitioning, also known as divide and conquer
- Start with the root node, which is the entire dataset, choose a feature that is the most predictive of the target class
- Continue choosing the next best candidate until a stopping criteria is reached:
  - *All (or nearly all) of the nodes have the same class*
  - *No remaining features to distinguish among examples*
  - *The tree has grown to a predefined size limit*

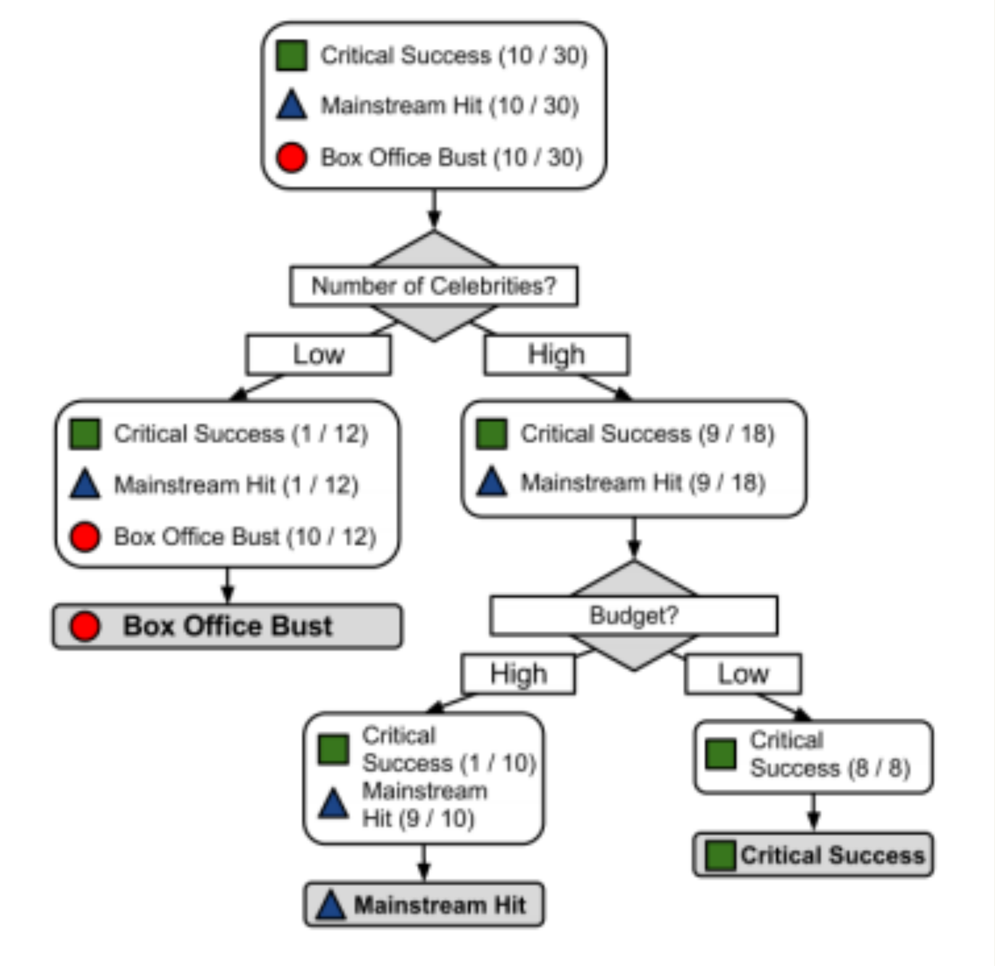
Example: Movie Releases







Decision Tree



C5.0 Algorithm

- Uses **entropy** for measuring purity
- Entropy – indicates how mixed the class values are (0 – completely homogenous, $log_2 c$ – maximum amount of disorder with c classes)

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

- where p_i is the proportion of values falling into class i
- E.g. if partitioning a set results in 60% in one class, and 40% in another class, then entropy = -0.6 * log2(0.6) – 0.4 * log2(0.4) = 0.97

Definition

Entropy

lack of order or predictability; gradual decline into disorder

Information Gain

- Select two features to split (e.g. refund, marital status)
- If a split results in a higher information gain due to feature X than feature Y, then use X

$$InfoGain(F) = Entropy(S_1) - Entropy(S_2)$$

- where S_1 is the segment before the split, and the partitions resulting from the split S_2

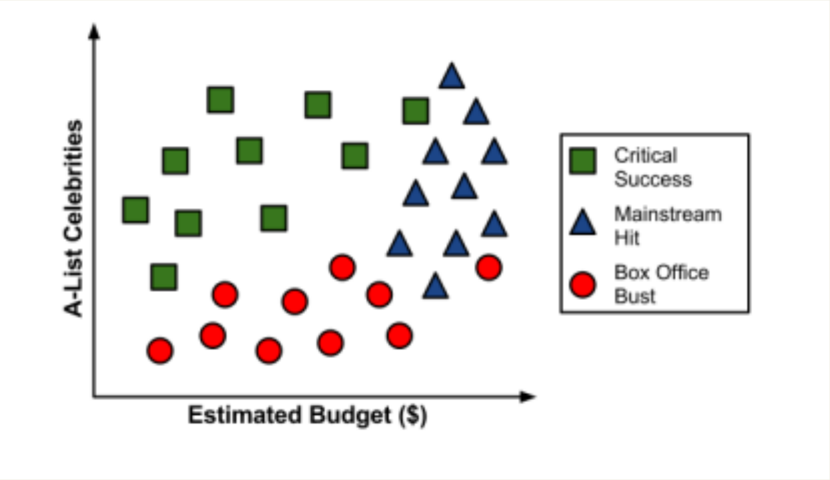
Multiple Partitions

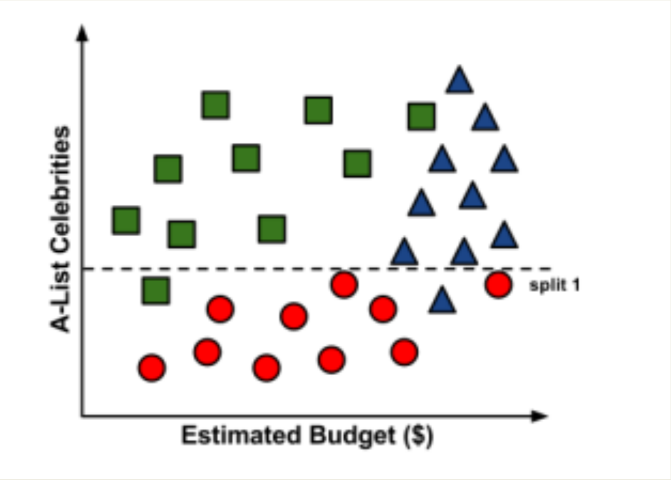
- Given n partitions resulting from a split S, the entropy of the split is:

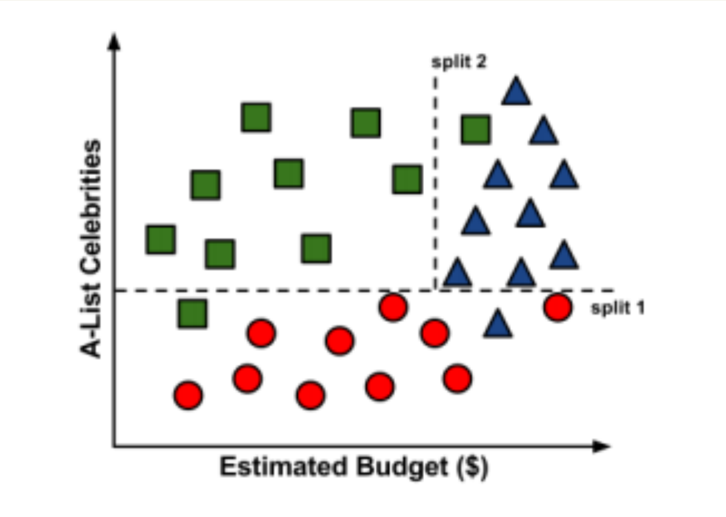
$$Entropy(S) = \sum_{i=1}^{n} w_i Entropy(P_i)$$

- where w_i is the proportion of examples falling in that partition.

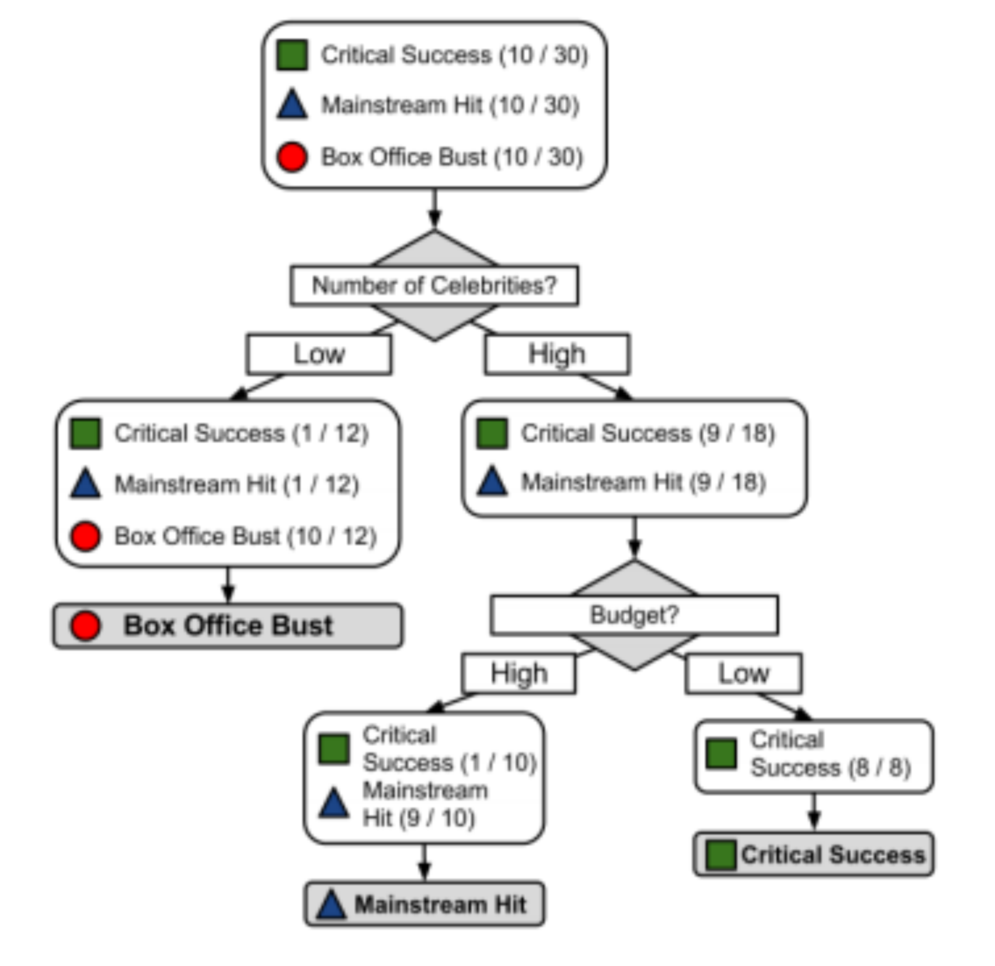
Example: Movie Releases



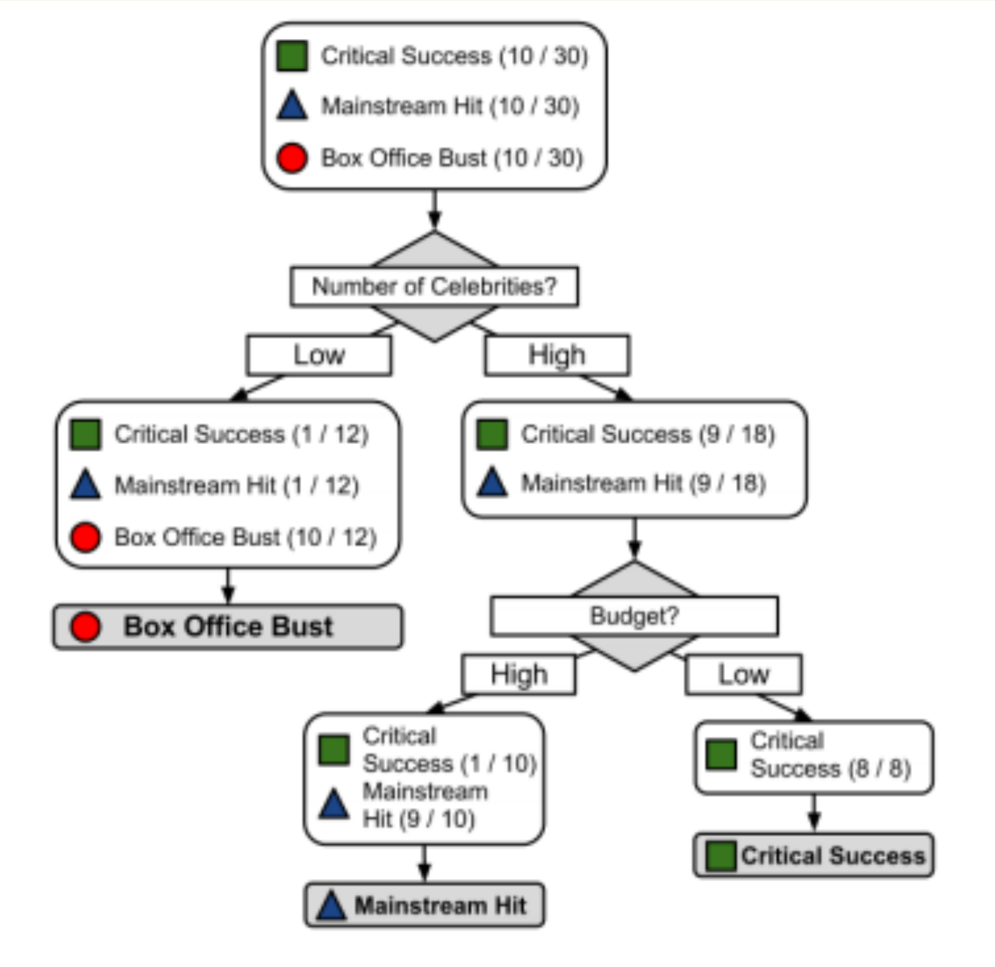




Decision Tree



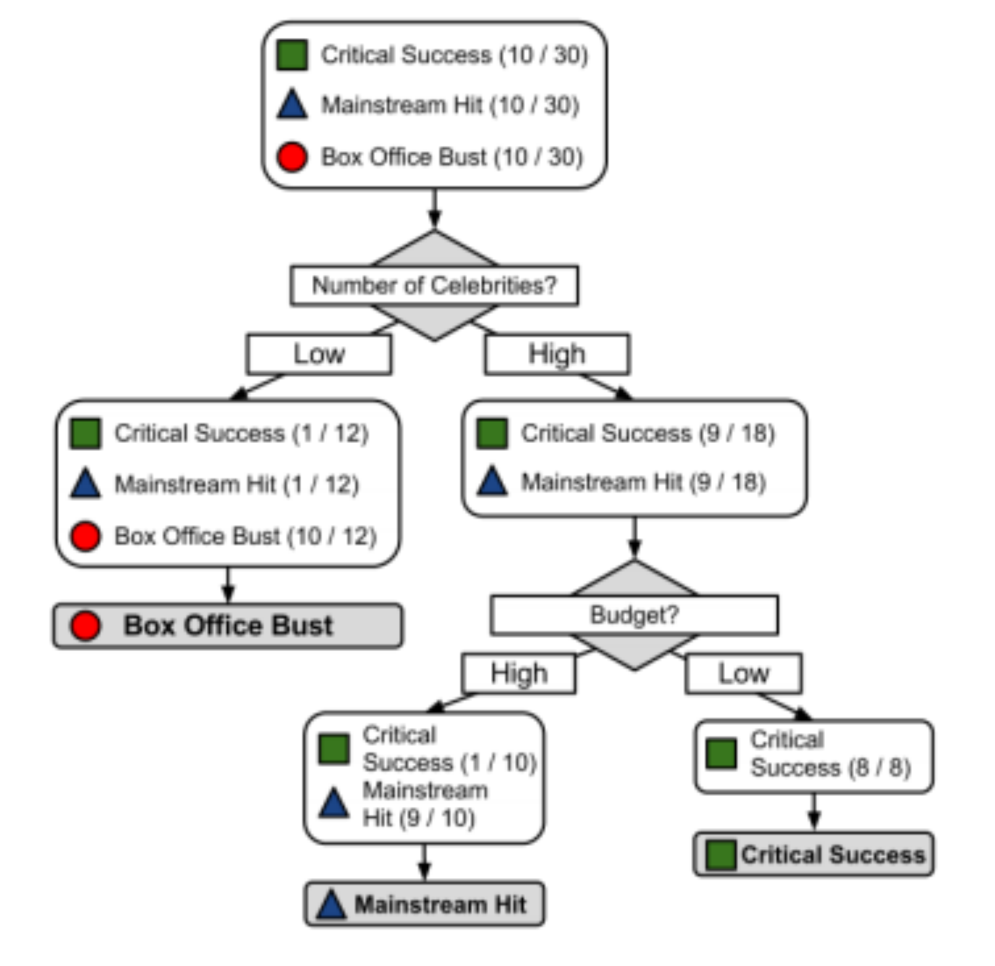
Decision Tree



$$-\frac{10}{30}\ log_2(\frac{10}{30}) - \frac{10}{30}\ log_2(\frac{10}{30}) - \frac{10}{30}\ log_2(\frac{10}{30})$$

1.5849

Decision Tree



$$-\frac{10}{30}\ log_2(\frac{10}{30}) - \frac{10}{30}\ log_2(\frac{10}{30}) - \frac{10}{30}\ log_2(\frac{10}{30})$$

1.5849

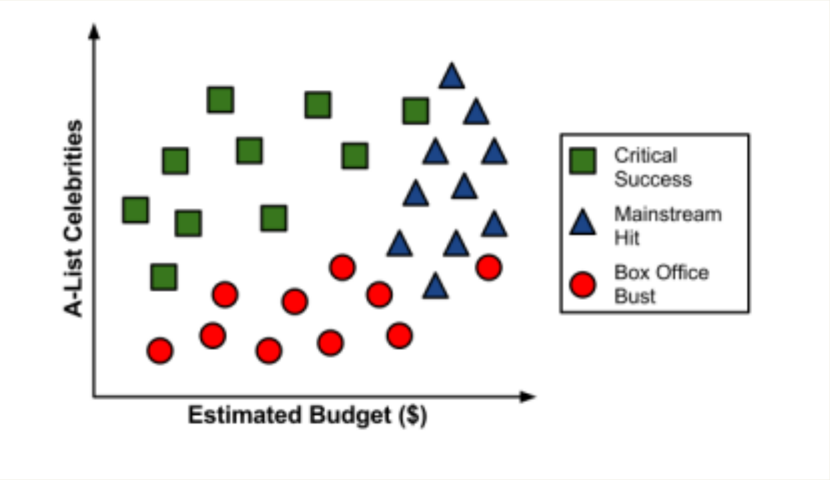
$$\frac{12}{30} * \left[-\frac{1}{12} log_2\left(\frac{1}{12}\right) - \frac{1}{12} log_2\left(\frac{1}{12}\right) - \frac{10}{12} log_2\left(\frac{10}{12}\right)\right]$$

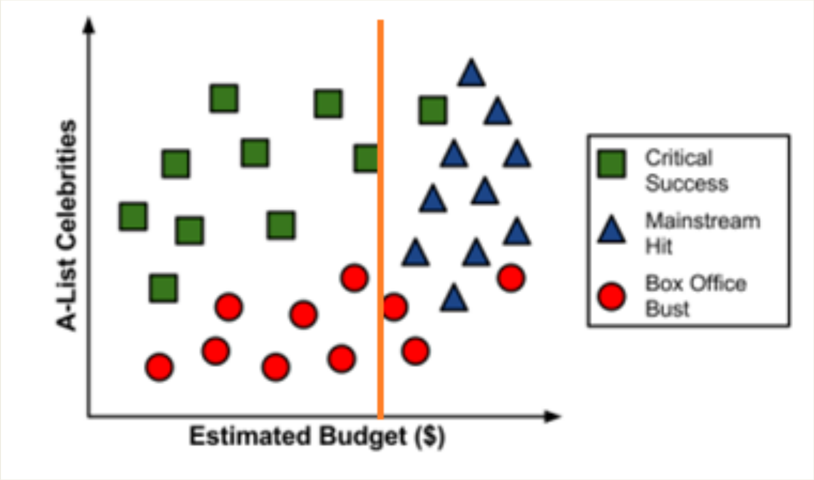
$+$

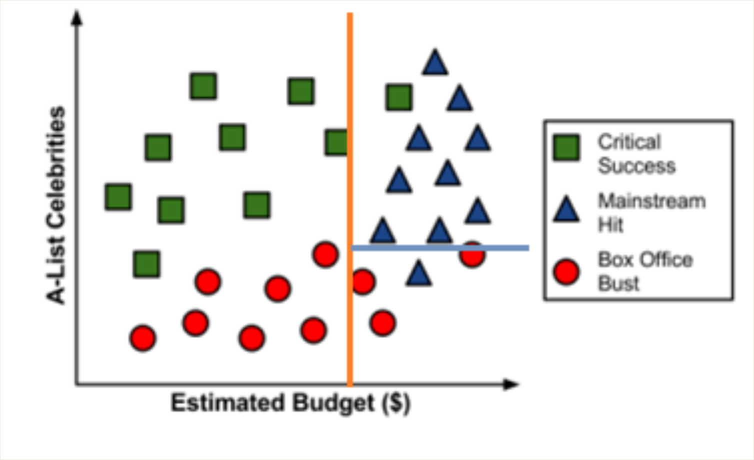
$$\frac{18}{30} * \left[-\frac{9}{18}\ log_2(\frac{9}{18}) - \frac{9}{18}\ log_2(\frac{9}{18})\right]$$

0.5667

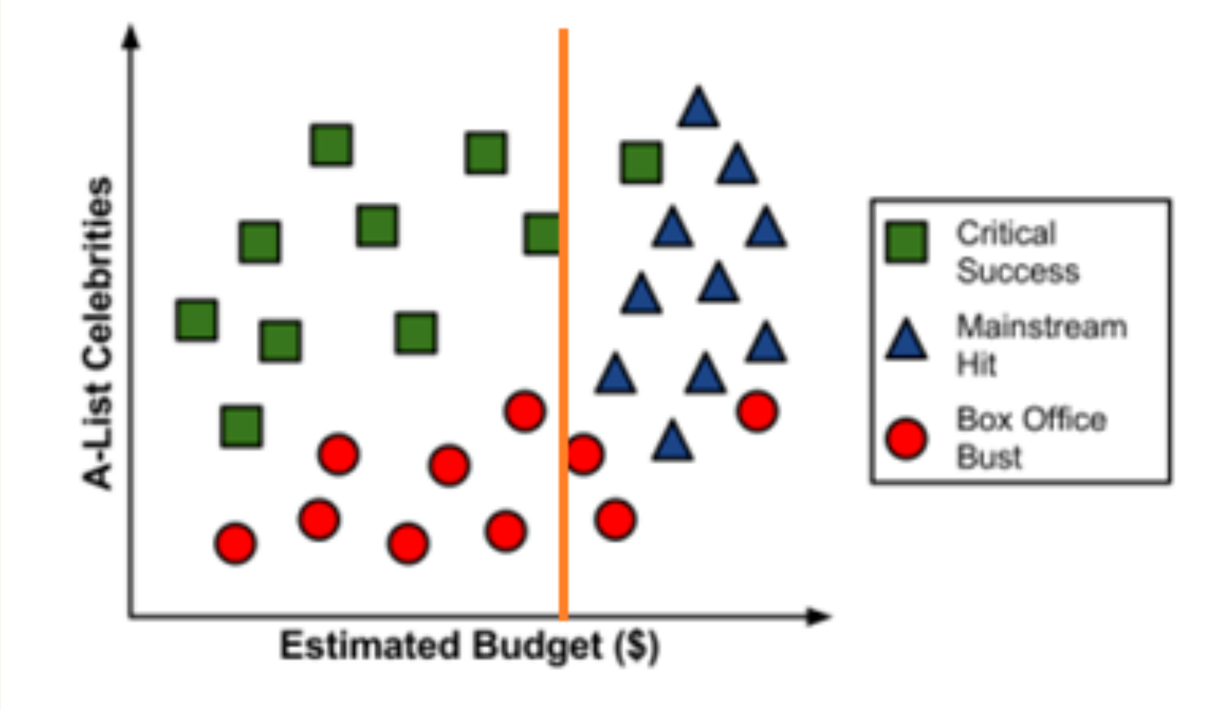
Alternative Split



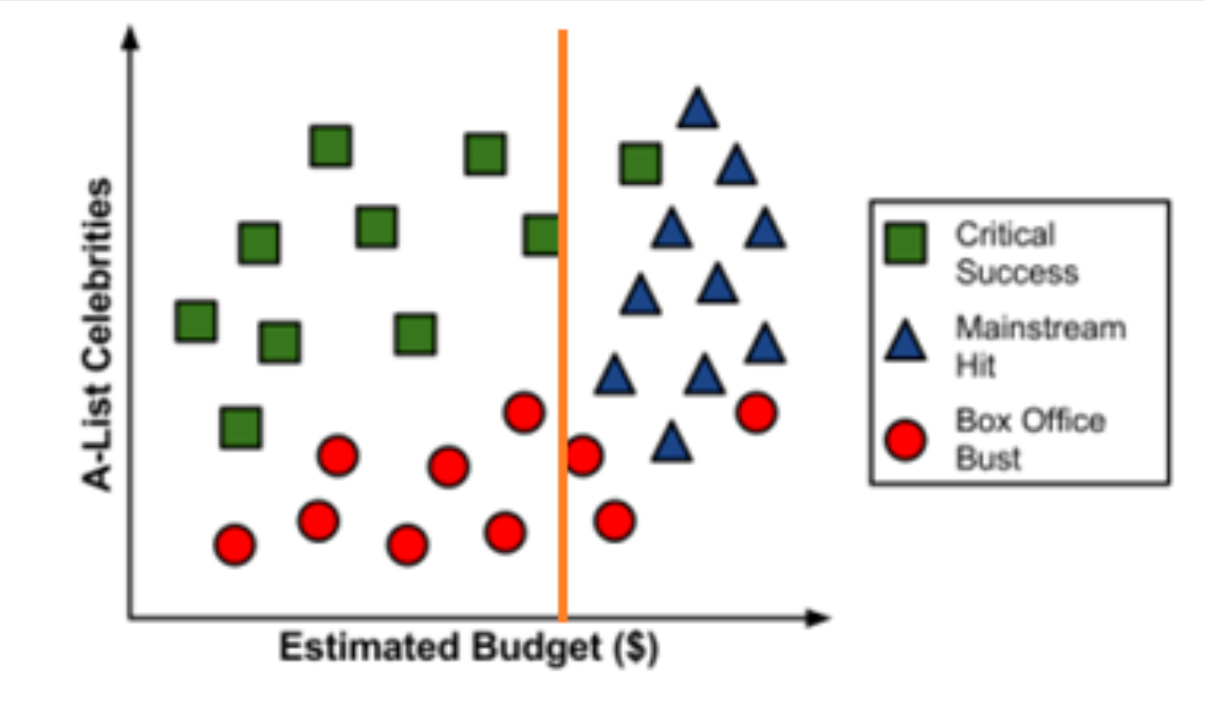




What is the Entropy?



What is the Entropy?



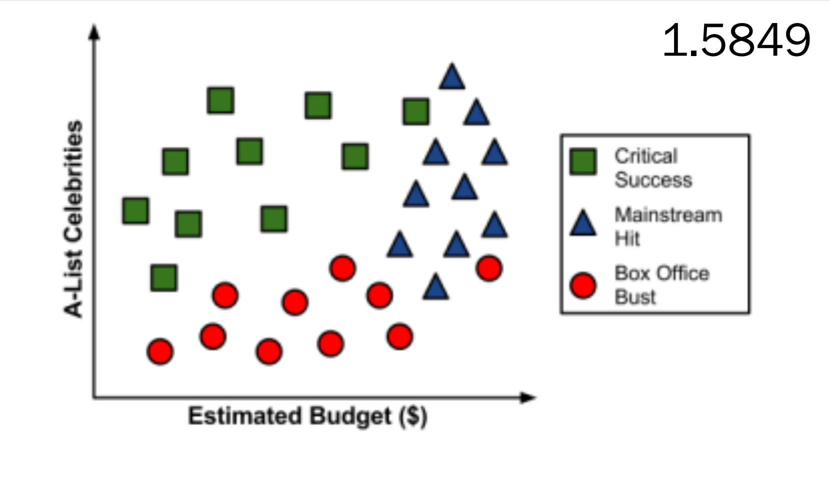
$$\frac{14}{30} * \left[-\frac{1}{14} log_2\left(\frac{1}{14}\right) - \frac{3}{14} log_2\left(\frac{3}{14}\right) - \frac{10}{14} log_2\left(\frac{10}{14}\right)\right]$$

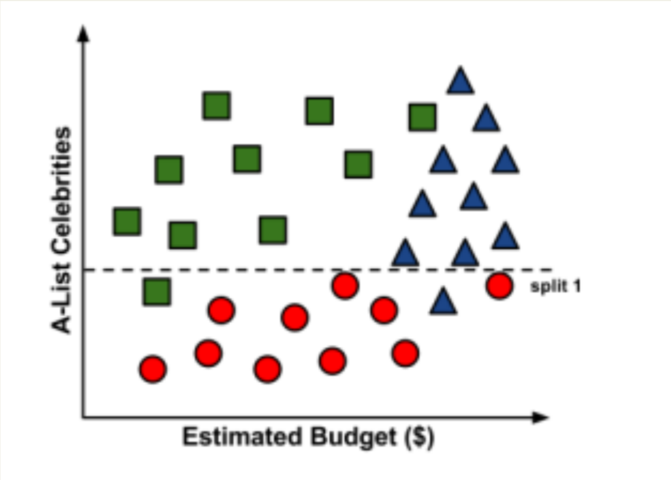
+

$$\frac{16}{30} * \left[-\frac{7}{16}\ log_2(\frac{7}{16}) - \frac{9}{16}\ log_2(\frac{9}{16})\right]$$

0.757

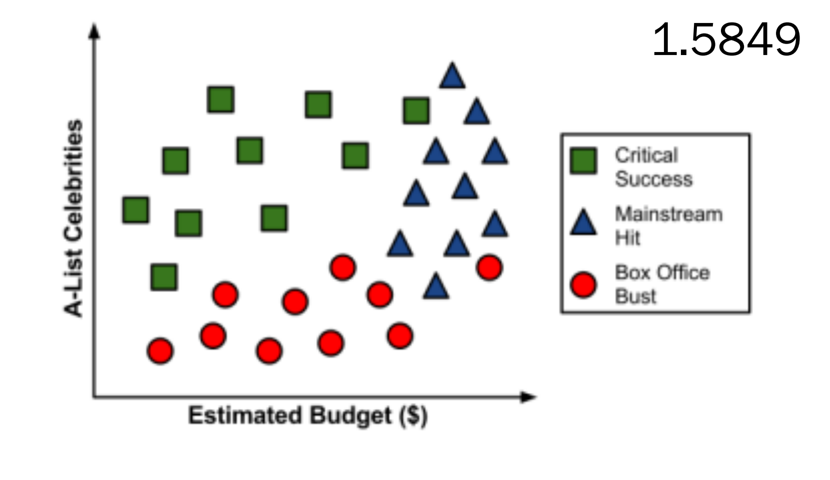
Split Option 1

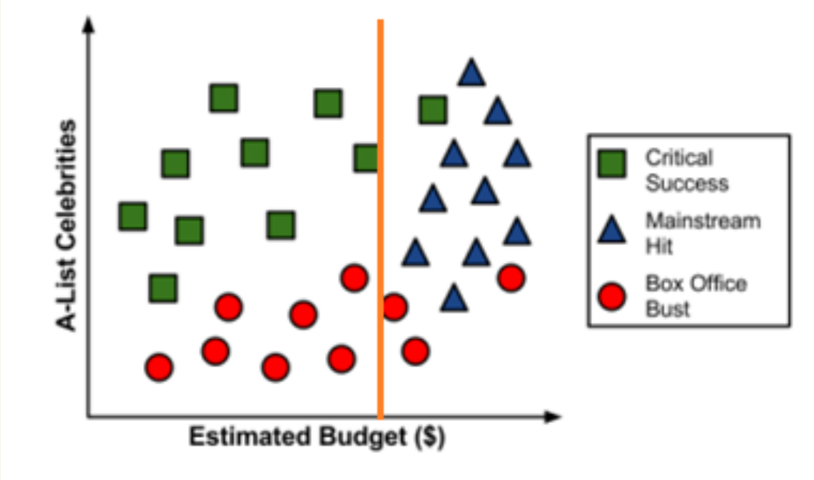




0.5667

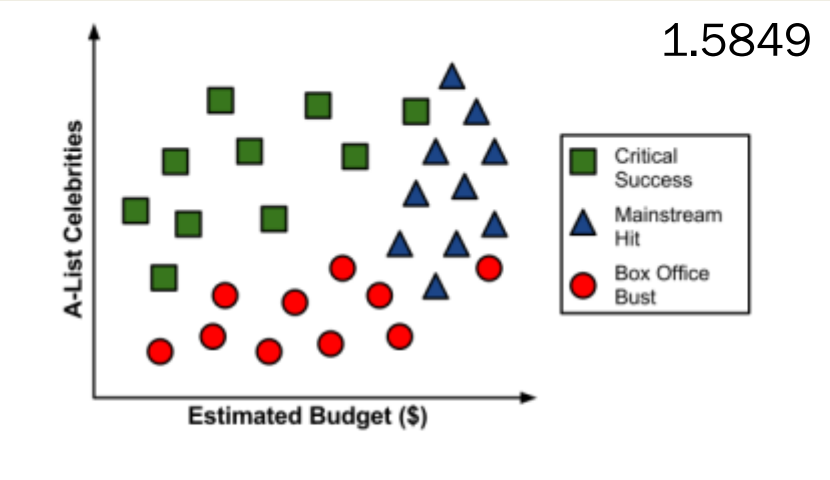
Split Option 2

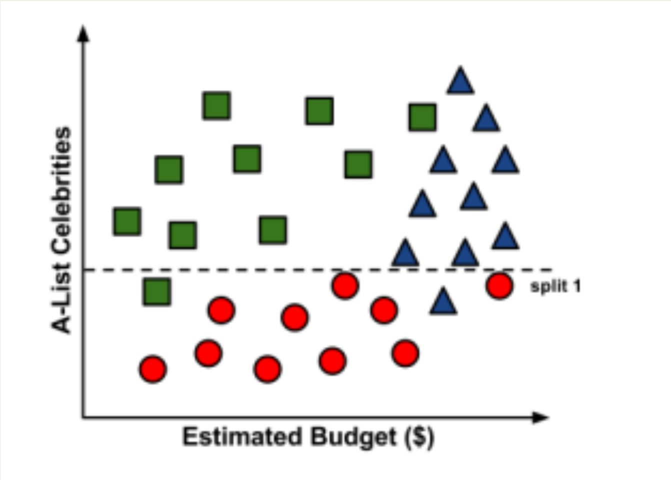




0.757

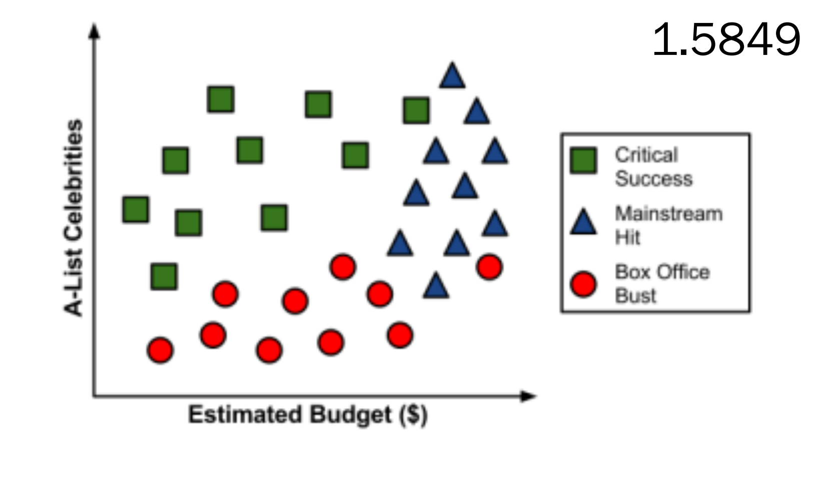
Split Option 1

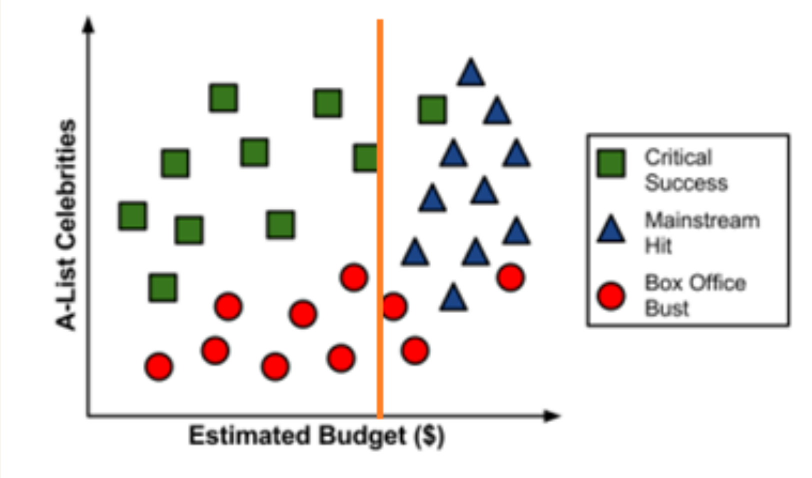




0.5667

Split Option 2





0.757

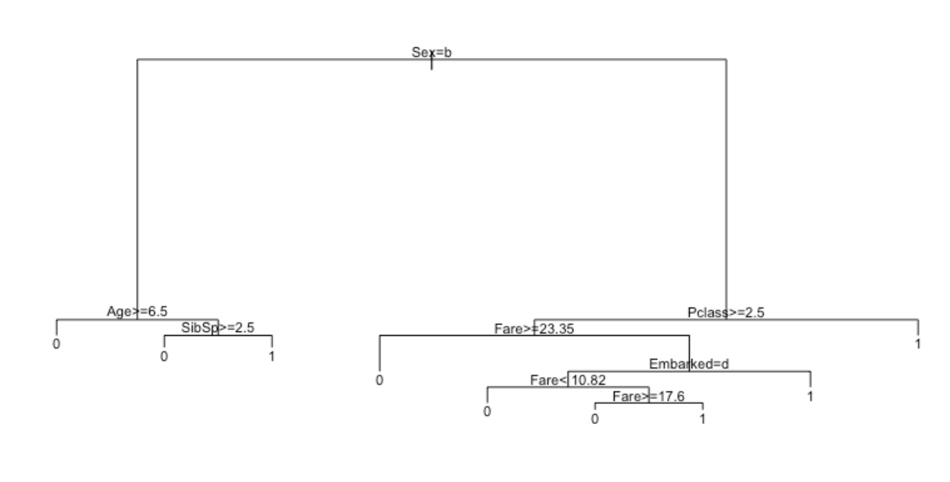
C5.0 in R

- install.packages('C50')
- library(C50)
- model <- C5.0(*train*, *class*, *trials*)
  - *train* is the training data frame without the classification
  - *class* is a factor vector with the classification for each row in *train*
  - *trials* controls the number of decision trees to be created
- p <- predict(*model*, *test*, type="class")
  - *model* is the model created by the C5.0 function
  - *test* is the test data data frame with the same feature as the training data frame
  - R
  - returns a vector of predicted class

rpart

- install.packages(‘rpart’)
- library(rpart)
- fit <- rpart(default ~ ., data=credit, method="class")
- plot(fit)
- text(fit, cex=0.7)

- Clear plot:
- dev.off()

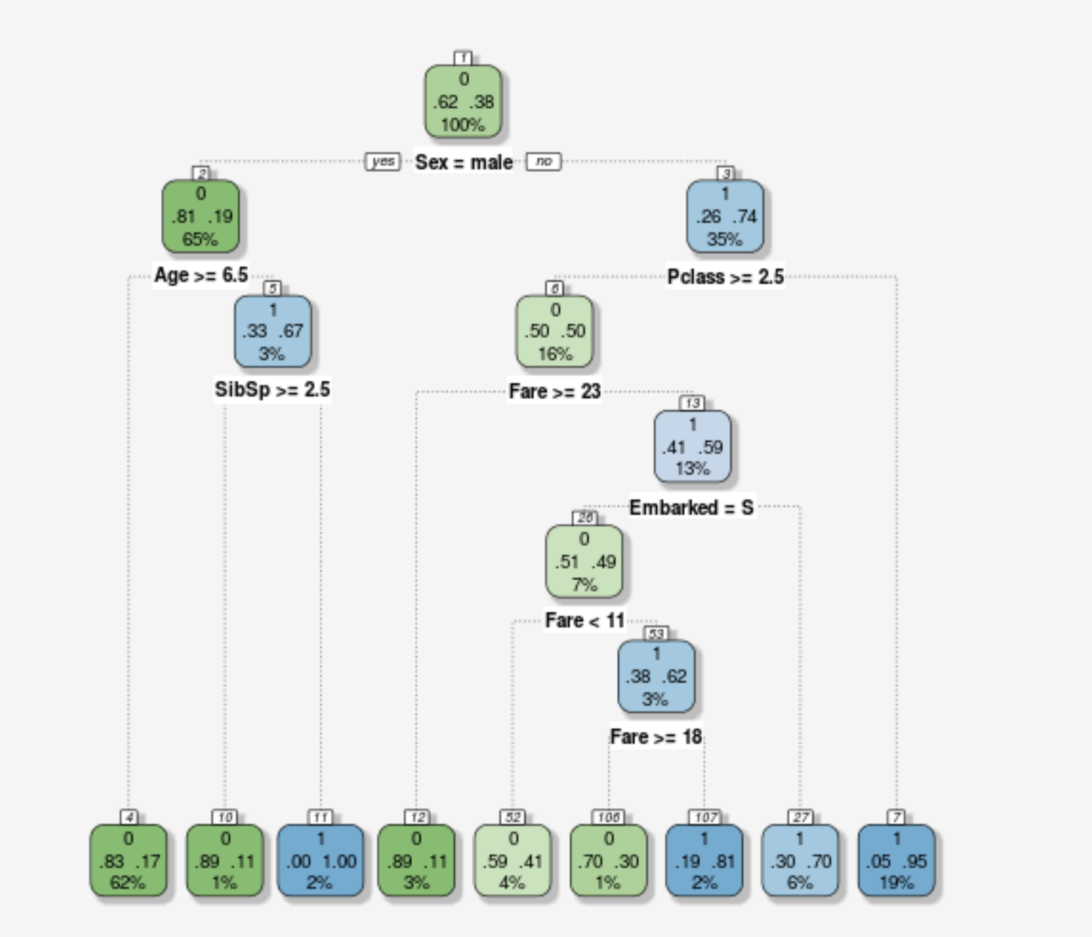


```
pdf("tree.pdf") fancyRpartPlot(fit) dev.close()
```

Fancy Plot

- install.packages('rattle')
- install.packages('rpart.plot')
- install.packages('RColorBrewer')
- library(rattle)
- library(rpart.plot)
- library(RColorBrewer)

- fancyRpartPlot(fit)
- pdf(“tree.pdf”)
- fancyRpartPlot(fit)
- dev.off()



Predict

- Prediction <- predict(fit, credit_test, type = "class")
- submit <- data.frame(Amount = credit_test$amount, Default = Prediction)