Decision Tree

Alymzhan Toleu
alymzhan.toleu@gmail.com

Entropy and Decision Tree

Types of classifier

- We can divide the large variety of classification approaches into roughly two main types
  - Discriminative
    - directly estimate a decision rule/boundary - e.g., decision tree
  - Instance based classifiers
    - Use observation directly (no models) - e.g. K nearest neighbors
  - Generative
    - build a generative statistical model - e.g., Bayesian networks

Decision Trees

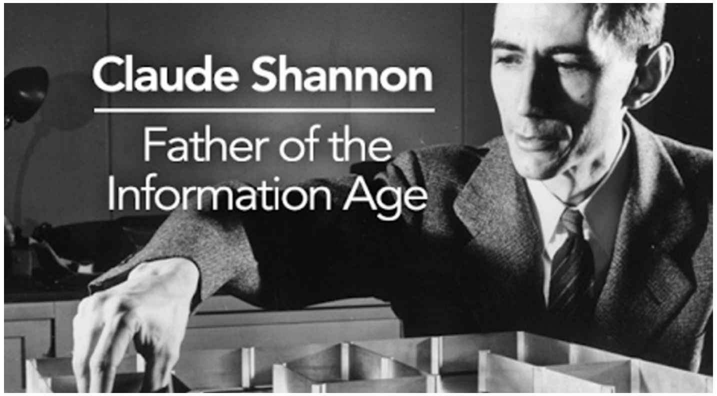
- One of the most intuitive classifiers
- Easy to understand and construct
- Surprisingly, also works very well

Entropy

Entropy

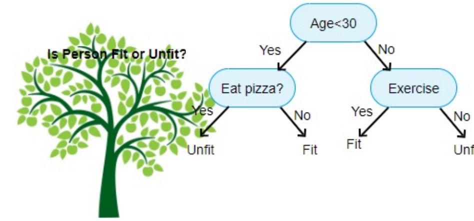
- Quantifies the amount of uncertainty associated with a specific probability distribution.
- The higher the entropy, the less confident we are in the outcome.
- Definition

$$H(X) = \sum_{c} - p(X = c) \log_2 p(X = c)$$

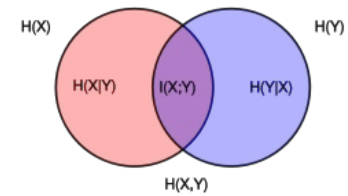


Claude Shannon (1916 – 2001), most of the work was done in Bell labs

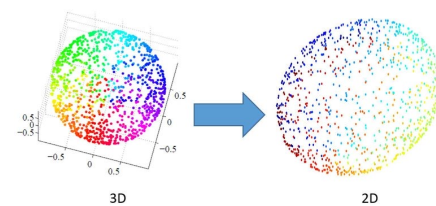
Entropy can be used to build classification tree that used to classify things.



Entropy is also the basis of mutual information that quantifies the relationship between two things. (e.g. in NLP, measure the relationship between two words)



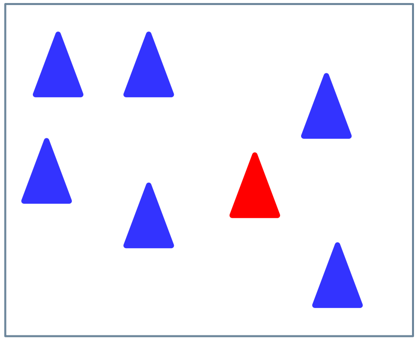
Entropy is basis of relative entropy (Kullback–Leibler divergence, KL) and cross entropy. (e.g. used in dimension reduction algorithms)



These three things all uses **entropy** to quantify **similarities** and **differences**.

Surprise -> Entropy

A



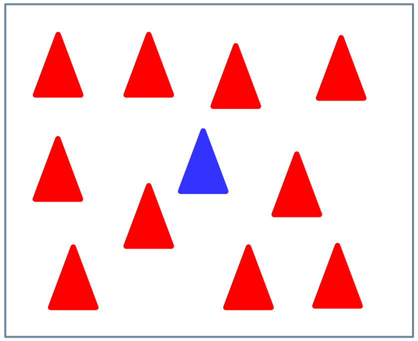
Randomly pick up a triangle.

There are 5 blue triangles, and 1 red triangle, then there is a higher probability that blue one will be picked up.

Since there is higher probability to pick up blue, it would be **not surprising** if it happened.

In contrast, if the red was picked up, it would be relatively surprised.

B

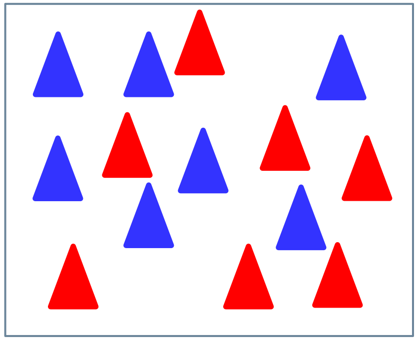


Area B has a lot of red triangle than blue.

High probability to pick up a red triangle, it is not surprised.

Low probability to pick up a blue triangle, it is relatively surprising.

C



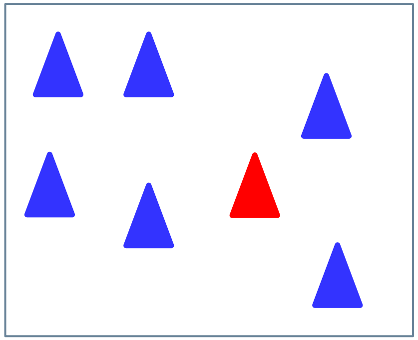
Area C has equal number of red and blue triangles.

Regardless what color triangle we picked up, we would be equally surprised.

In some way, surprise is inversely related to probability.

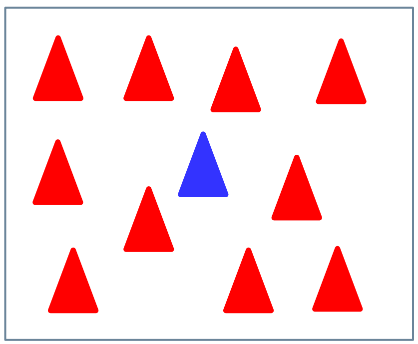
Surprise -> Entropy

A



When the the probability of picking up a blue triangle is **high**, **the surprise** is **low**.

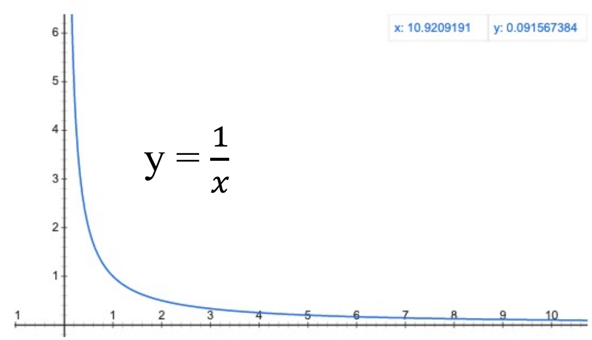
B



When the the probability of picking up a blue triangle is **low**, **the surprise** is **high**.

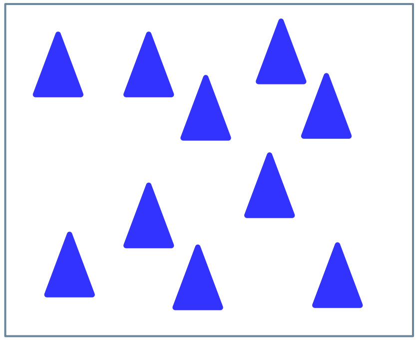
We may calculate the **surprise** use the inverse of the probability.

$$\frac{1}{probability}$$



There is a problem associated with inverse of the probability to calculate the surprise $\approx \frac{1}{probability}$

A



For example, we have **Area A,** which only contains blue triangles.

The probability to picking up a blue triangle is 1.

Then we want the **surprise** to picking up a blue triangle to be 0.

However, when we calculate the inverse of the probability:

$$\frac{1}{probability} = \frac{1}{1} = 1$$

→ We want 0

Instead of just using the inverse of the probability, we could add a log function:

$$\text{Log}\left(\frac{1}{probability} \right)$$

Now, let's calculate the surprise with the inverse of the probability:

Picking up the blue: $\log\left(\frac{1}{probability}\right) = \log\left(\frac{1}{1}\right) = \log(1) = 0$

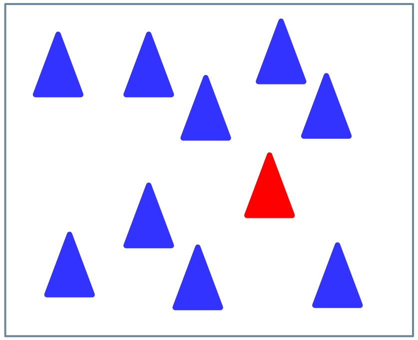
Surprise is zero

Picking up the red: $\log\left(\frac{1}{probability}\right) = \log\left(\frac{1}{0}\right) = \log(0) = undefined$

It also make sense, because the red will never be picked up

Now, surprise $\approx \log(\frac{1}{probability})$

A



For example, we have **Area A**

The probability to picking up a blue triangle is 0.9.

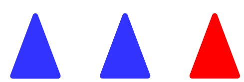
The probability to picking up a red triangle is 0.1.

Surprise for picking up a blue triangle: $log_2\left(\frac{1}{probability}\right) = log_2\left(\frac{1}{0.9}\right) = log_2(1) - log_2(0.9) = 0.15$

Surprise for picking up a red triangle: $log_2\left(\frac{1}{probability}\right) = log_2\left(\frac{1}{0.1}\right) = log_2(1) - log_2(0.1) = 3.32$

For example, we want to calculate surprise for three times picking events:

after three times we got:



The probability of this sequence happens is 0.9*0.9*0.1

Calculate the total surprise:

$$
\begin{aligned}
total\ surprise = \ log_2\left(\frac{1}{0.9*0.9*0.1}\right) &= log_2(1) - log_2(0.9*0.9*0.1) \\
&= log_2(1) - [log_2(0.9) + log_2(0.9) + log_2(0.1)] \\
&= 0 - log_2(0.9) - log_2(0.9) - log_2(0.1) \\
&= 0.15 + 0.15 + 3.32
\end{aligned}
$$

Total surprise for sequence of a events is just sum of the surprise for each individual event

Now, imagine we do 100 times of pickings. What will be the total surprise for these 100 times?

		
Probability $P(x)$	0.9	0.1
Surprise $log_2\left(\frac{1}{P(x)}\right)$	0.15	3.32

Entropy of the triangle - the expected **surprise** every time we pick up the triangle.

average surprise for per time : $\frac{(0.9*100)*0.15 + (0.1*100)*3.32}{100} = 0.467$

average surprise for per time : $(0.9 * 0.15) + (0.1 * 3.32) = 0.467$

$$\sum xP(X = x)$$

Plug the surprise and probability function in

Entropy equation: $$\sum log_2\left(\frac{1}{P(x)}\right)\mathrm{P}(x)$$

Entropy equation: $$\sum \underbrace{log_2\left(\frac{1}{P(x)}\right)}_{\text{Surprise value}} \underbrace{\mathrm{P}(x)}_{\text{the probability of surprise}}$$

Standard entropy equation:

$$Entropy = \sum log_2\left(\frac{1}{P(x)}\right) \mathrm{P}(x)$$

$$= \sum \mathrm{P}(x)\, [log_2(1) - log_2(P(x))]$$

$$= \sum \mathrm{P}(x)\, [\, 0 - log_2(P(x))]$$

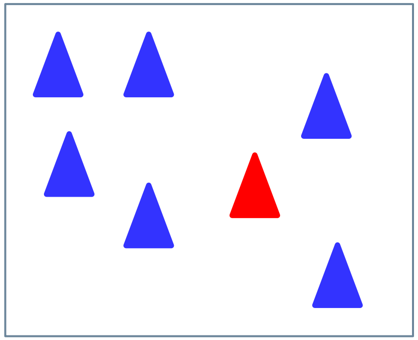
$$\boxed{= -\sum \mathrm{P}(x)\, log_2(P(x))}$$

This is the original version, but not easy to understand.

Calculate the Entropy

$$Entropy = \sum log_2\left(\frac{1}{P(x)}\right)\mathrm{P}(x)$$

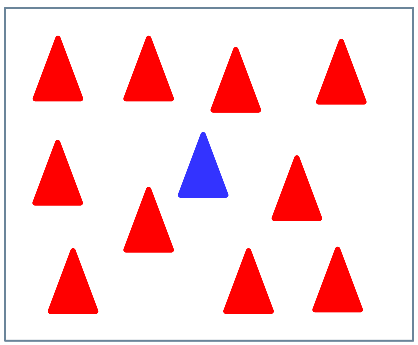
A



$$E_A = log_2\left(\frac{1}{1/7}\right)\frac{1}{7} + log_2\left(\frac{1}{6/7}\right)\frac{6}{7} = 0.59$$

As result, we can use entropy to **quantify the similarities and differencs** in the number of blue and red triangels in each area.

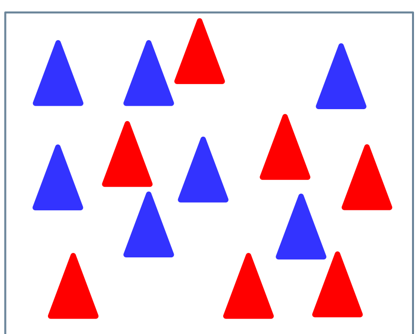
B



$$E_B = log_2\left(\frac{1}{1/11}\right)\frac{1}{11} + log_2\left(\frac{1}{10/11}\right)\frac{10}{11} = 0.44$$

Entropy is **high**, when we have **the same number** of both colors of triangle.
Entropy is **low**, when we **increase** the difference in the number of blue and red triangles.

C



$$E_c = log_2\left(\frac{1}{7/14}\right)\frac{1}{14} + log_2\left(\frac{1}{7/14}\right)\frac{7}{14} = 1$$

Conditional entropy

$$Entropy = \sum log_2\left(\frac{1}{P(x)}\right)\mathrm{P}(x)$$

Movie length	Liked?
Short	Yes
Short	No
Medium	Yes
long	No
Long	No
Medium	Yes
Short	Yes
Long	Yes
Medium	Yes

Entropy measures the uncertainty in a specific distribution.

What is the entropy of "Liked"?

$$E(Liked) = \frac{6}{9}log_2\left(\frac{1}{\frac{6}{9}}\right) + \frac{3}{9}log_2\left(\frac{1}{\frac{3}{9}}\right) = 0.92$$

What is the entropy of "Liked" among movies with a certain value of length?
This becomes a conditional entropy problem:

$$E(Liked \mid Length = \text{"}Short\text{"}) = ?$$

$$E(Liked \mid Length = \text{"}Short\text{"}) = \frac{2}{3}log_2\left(\frac{1}{\frac{2}{3}}\right) + \frac{1}{3}log_2\left(\frac{1}{\frac{1}{3}}\right) = 0.92$$

Conditional entropy : Examples for specific values

Movie length	Liked?
Short	Yes
Short	No
Medium	Yes
long	No
Long	No
Medium	Yes
Short	Yes
Long	Yes
Medium	Yes

$$E(Liked \mid len = \text{“}S\text{”}) = \frac{2}{3} log_2\left(\frac{1}{\frac{2}{3}}\right) + \frac{1}{3} log_2\left(\frac{1}{\frac{1}{3}}\right) = 0.92$$

$$E(Liked \mid len = \text{“}M\text{”}) = 0$$

$$H(Y \mid X) = \sum_i P(X = i) H(Y \mid X = i)$$

$$E(Liked \mid len = \text{“}L\text{”}) = 0.92$$

Conditional entropy equation

$$E(Y|X) = \sum P(X = i) E(Y|X = i)$$

Lets compute

$$E(Liked \mid len) = P(len = \text{“}S\text{”}) E(Liked \mid len = \text{“}S\text{”})$$
$$+ P(len = \text{“}M\text{”}) E(Liked \mid len = \text{“}M\text{”})$$
$$+ P(len = \text{“}L\text{”}) E(Liked \mid len = \text{“}L\text{”})$$
$$= \frac{3}{9} * 0.92 + \frac{3}{9} * 0 + \frac{3}{9} * 0.92 = 0.62$$

Information Gain

- How much do we gain (in terms of reduction in entropy) from knowing one of the attributes.
- In other words, what is the reduction in entropy from this knowledge.
- Definition:

$$IG(Y|X) = E(Y) - E(Y|X)$$

- For example,

$$E(Liked) = 0.92$$

$$E(Liked \mid len) = 0.62$$

$$IG(Liked|len) = E(Liked) - E(Liked|len)$$
$$= 0.92\text{-}0.62 = 0.3$$

Movie length	Liked?
Short	Yes
Short	No
Medium	Yes
long	No
Long	No
Medium	Yes
Short	Yes
Long	Yes
Medium	Yes

Building a Decision Tree

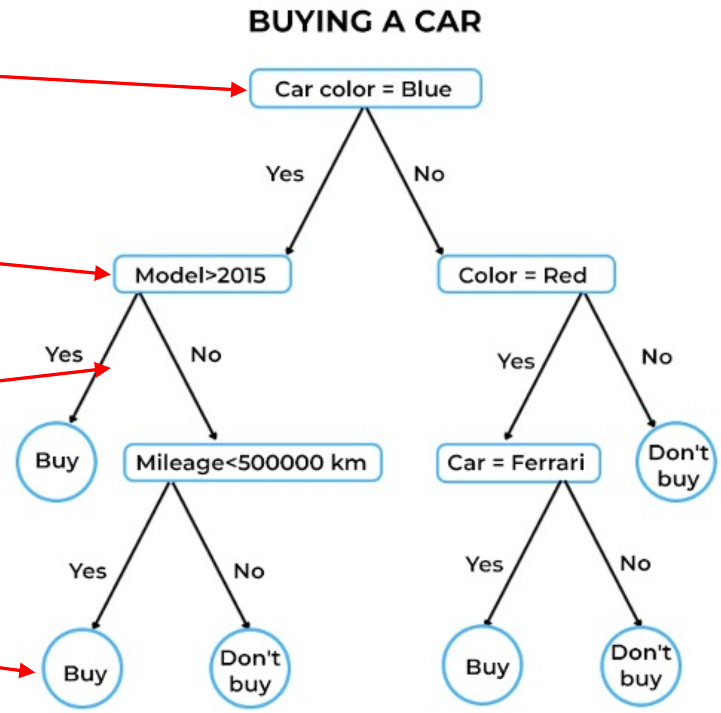
Structure of a decision tree

Root node

Internal nodes correspond to attributes (features).

Edges denote assignment.

Leaves correspond to classification outcome.



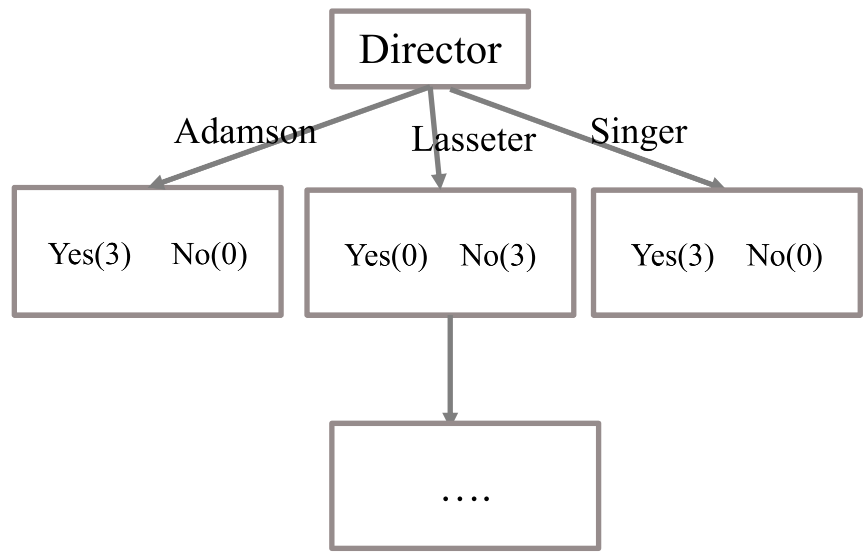
Netflix

Data set

Building a decision tree from the dataset!

Attributes (features) — Label

Movie	Type	Length	Director	Famous actors	Liked?
m1	Comedy	Short	Adamson	No	Yes
m2	Animated	Short	Lasseter	No	No
m3	Drama	Medium	Adamson	No	Yes
m4	animated	long	Lasseter	Yes	No
m5	Comedy	Long	Lasseter	Yes	No
m6	Drama	Medium	Singer	Yes	Yes
m7	animated	Short	Singer	No	Yes
m8	Comedy	Long	Adamson	Yes	Yes
m9	Drama	Medium	Lasseter	No	Yes



How?

- defined the entropy, conditional entropy and information gain.
- looking for a good criteria for selecting the best attribute for a node split.
- use Gini impurity, entropy and information gain as our criteria for a good split.

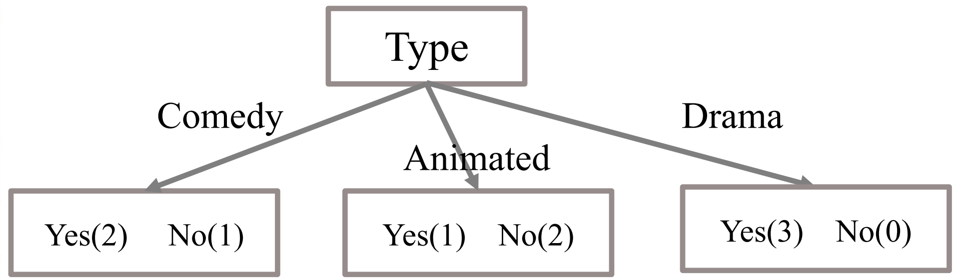
Building a decision tree

There are three ways to build a decision tree

- Gini impurity (minimizes the Gini impurity)
- Entropy (minimizes the entropy)
- Information gain (maximizes the information gain at each node)

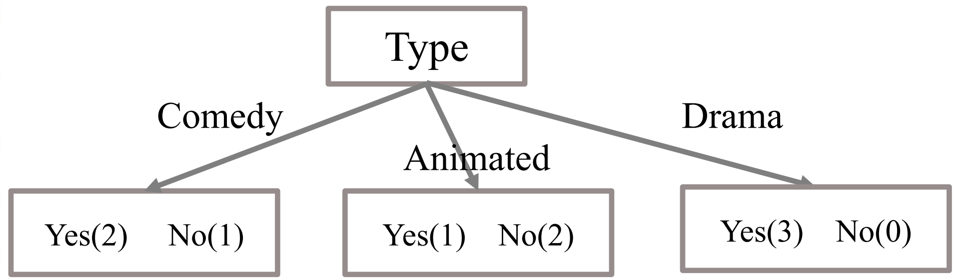
Building a decision tree

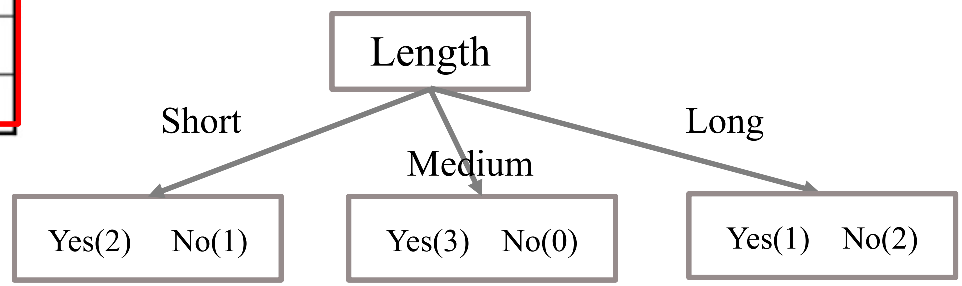
Movie	Type	Length	Director	Famous actors	Liked?
m1	Comedy	Short	Adamson	No	Yes
m2	Animated	Short	Lasseter	No	No
m3	Drama	Medium	Adamson	No	Yes
m4	animated	long	Lasseter	Yes	No
m5	Comedy	Long	Lasseter	Yes	No
m6	Drama	Medium	Singer	Yes	Yes
m7	animated	Short	Singer	No	Yes
m8	Comedy	Long	Adamson	Yes	Yes
m9	Drama	Medium	Lasseter	No	Yes



Building a decision tree

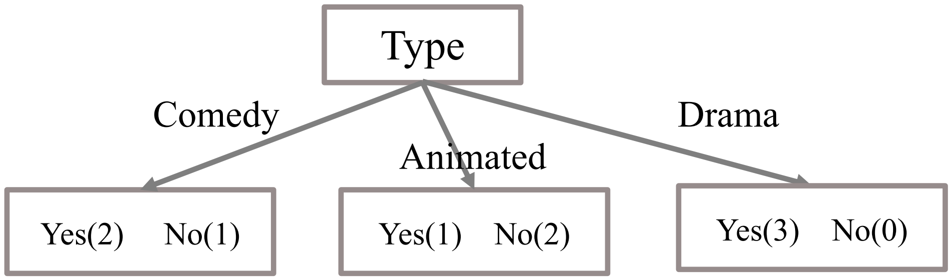
Movie	Type	Length	Director	Famous actors	Liked?
m1	Comedy	Short	Adamson	No	Yes
m2	Animated	Short	Lasseter	No	No
m3	Drama	Medium	Adamson	No	Yes
m4	animated	long	Lasseter	Yes	No
m5	Comedy	Long	Lasseter	Yes	No
m6	Drama	Medium	Singer	Yes	Yes
m7	animated	Short	Singer	No	Yes
m8	Comedy	Long	Adamson	Yes	Yes
m9	Drama	Medium	Lasseter	No	Yes

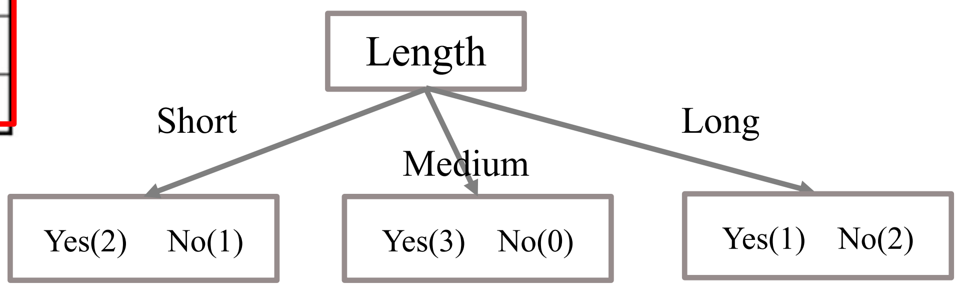


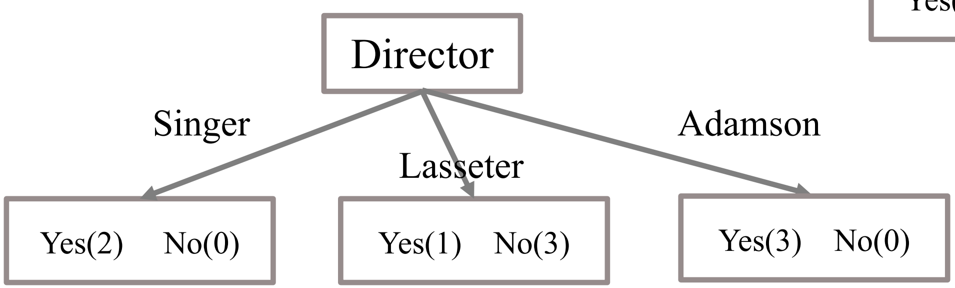


Building a decision tree

Movie	Type	Length	Director	Famous actors	Liked?
m1	Comedy	Short	Adamson	No	Yes
m2	Animated	Short	Lasseter	No	No
m3	Drama	Medium	Adamson	No	Yes
m4	animated	long	Lasseter	Yes	No
m5	Comedy	Long	Lasseter	Yes	No
m6	Drama	Medium	Singer	Yes	Yes
m7	animated	Short	Singer	No	Yes
m8	Comedy	Long	Adamson	Yes	Yes
m9	Drama	Medium	Lasseter	No	Yes

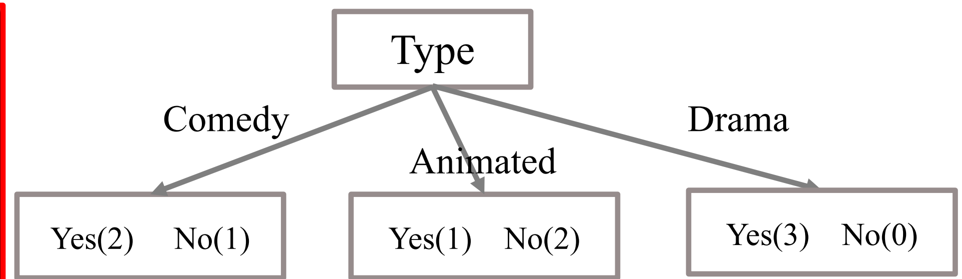


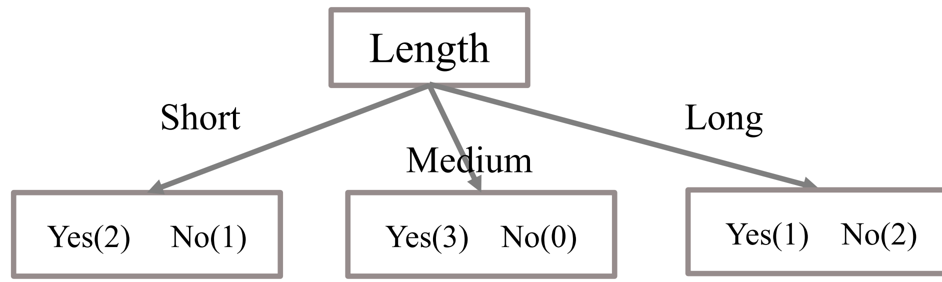


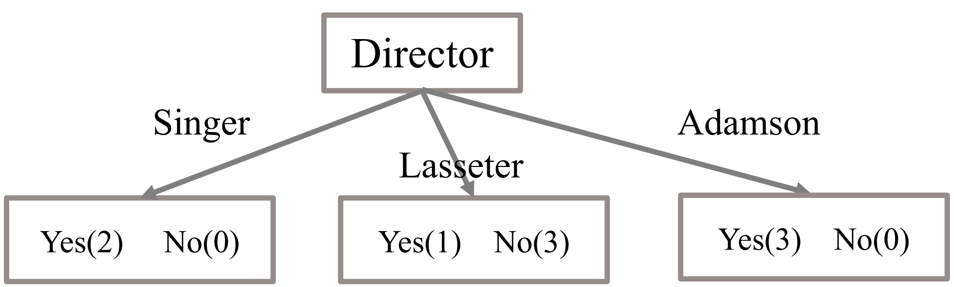


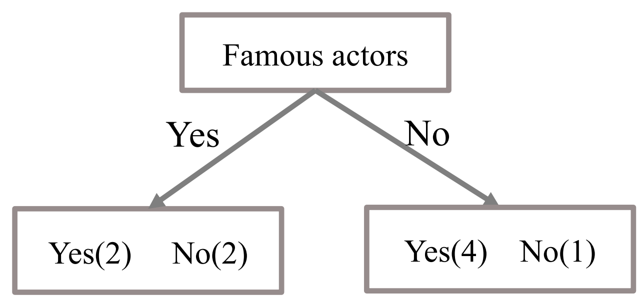
Building a decision tree

Movie	Type	Length	Director	Famous actors	Liked?
m1	Comedy	Short	Adamson	No	Yes
m2	Animated	Short	Lasseter	No	No
m3	Drama	Medium	Adamson	No	Yes
m4	animated	long	Lasseter	Yes	No
m5	Comedy	Long	Lasseter	Yes	No
m6	Drama	Medium	Singer	Yes	Yes
m7	animated	Short	Singer	No	Yes
m8	Comedy	Long	Adamson	Yes	Yes
m9	Drama	Medium	Lasseter	No	Yes

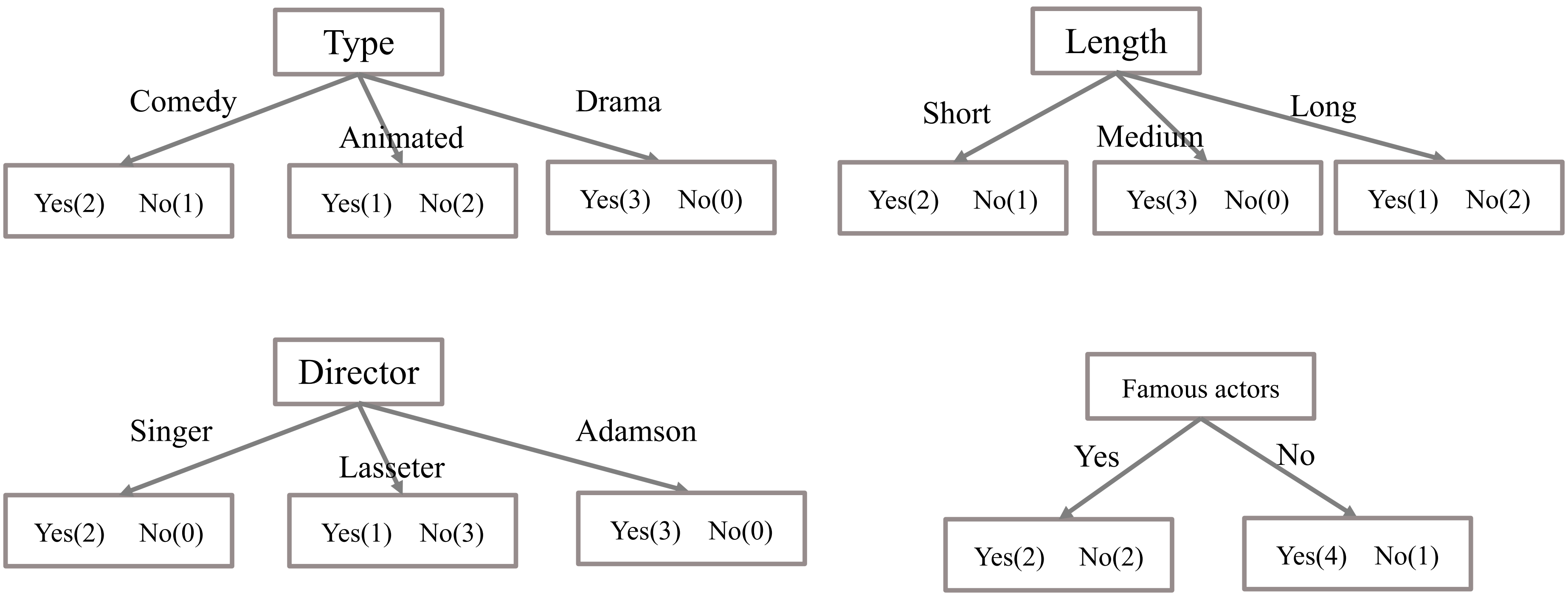








Building a decision tree

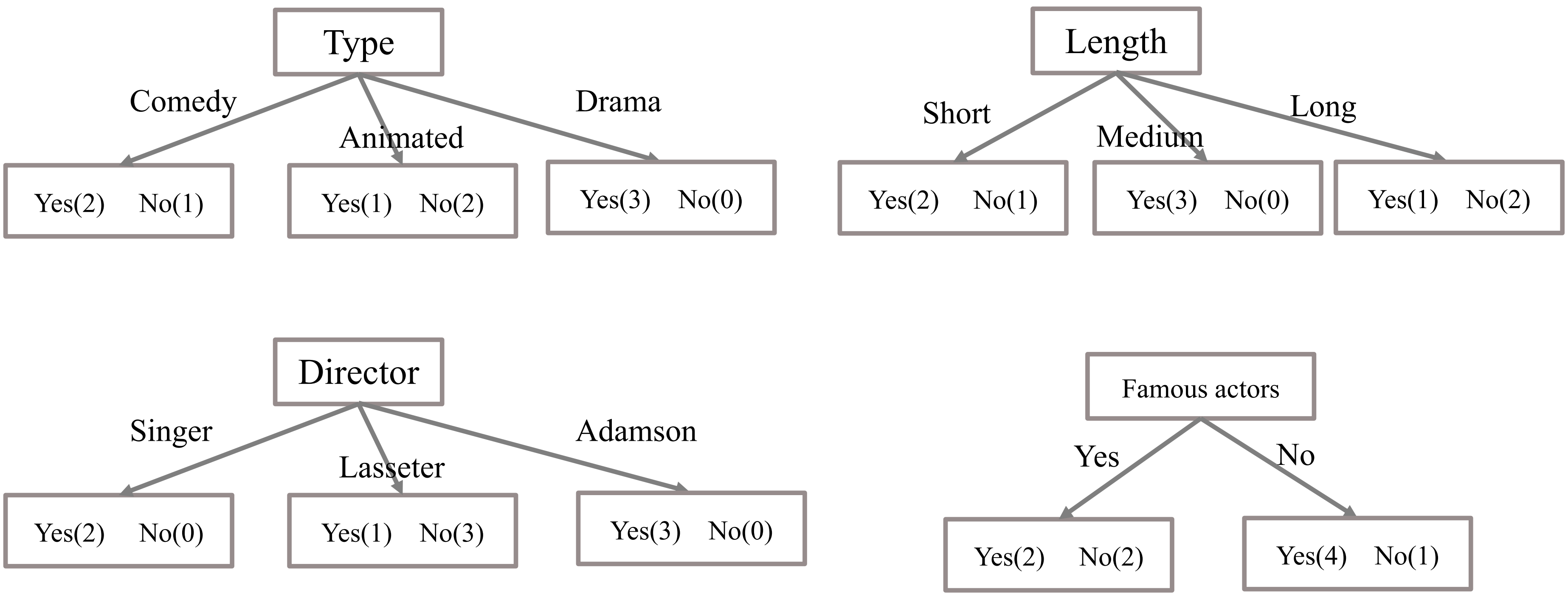


Looking at these four little trees, we see **neither one** does a pefect job predicting the movies **will be liked or will not be liked.**

Specifically, these **leaves** contain **mixtures** of movies that are liked and not liked.

They are called **impure leaves**.

Building a decision tree



Both leaves of Famou actors tree are **impure**.

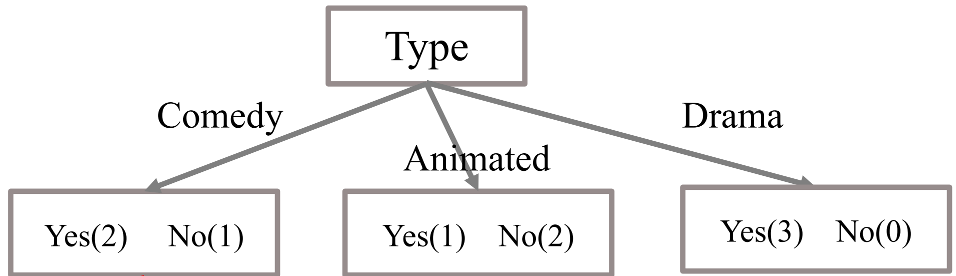
Type and Length trees contain two **impure** leaves in each of them.

Only one leaf in Director tree is **impure.**

It seems that **Director does a better job** at predicting if the movies **will be liked or will not be liked.**

How to quantifies the differences?

Quantifying the Impurity

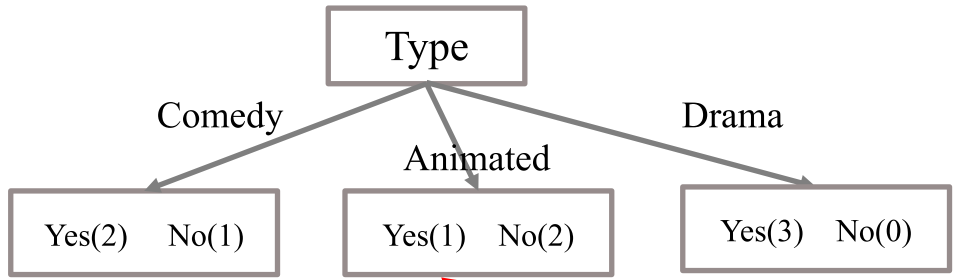


Gini impurity = 0.44

Gini impurity of this leaf $= 1 - P(yes)^2 - P(No)^2$

$$= 1 - \left(\frac{2}{2+1}\right)^2 - \left(\frac{1}{2+1}\right)^2 = 0.444$$

Quantifying the Impurity



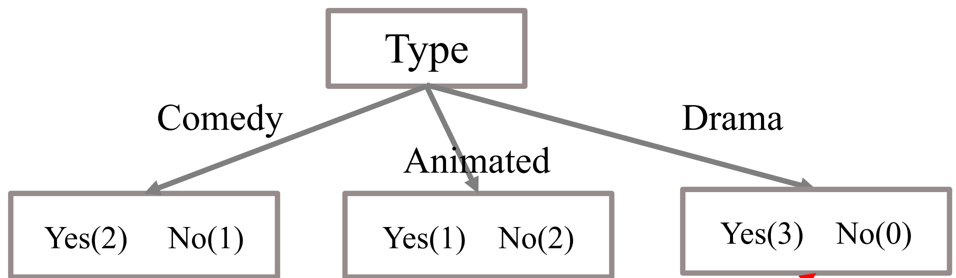
Gini impurity = 0.44

Gini impurity = 0.44

Gini impurity of this leaf = $1 - P(yes)^2 - P(No)^2$

$$= 1 - \left(\frac{1}{1+2}\right)^2 - \left(\frac{2}{1+2}\right)^2 = 0.444$$

Quantifying the Impurity

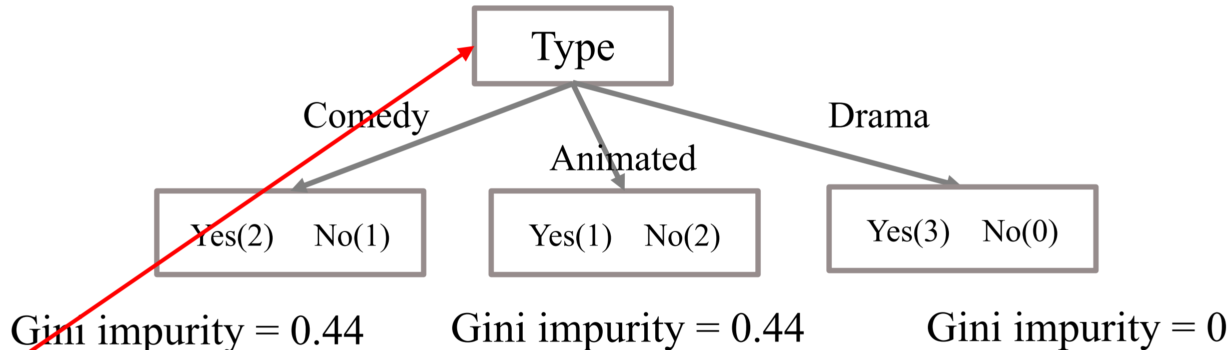


Gini impurity = 0.44 Gini impurity = 0.44 Gini impurity = 0

Gini impurity of this leaf $= 1 - P(yes)^2 - P(No)^2$

$$= 1 - \left(\frac{3}{3}\right)^2 - \left(\frac{0}{3}\right)^2 = 0$$

Quantifying the Impurity



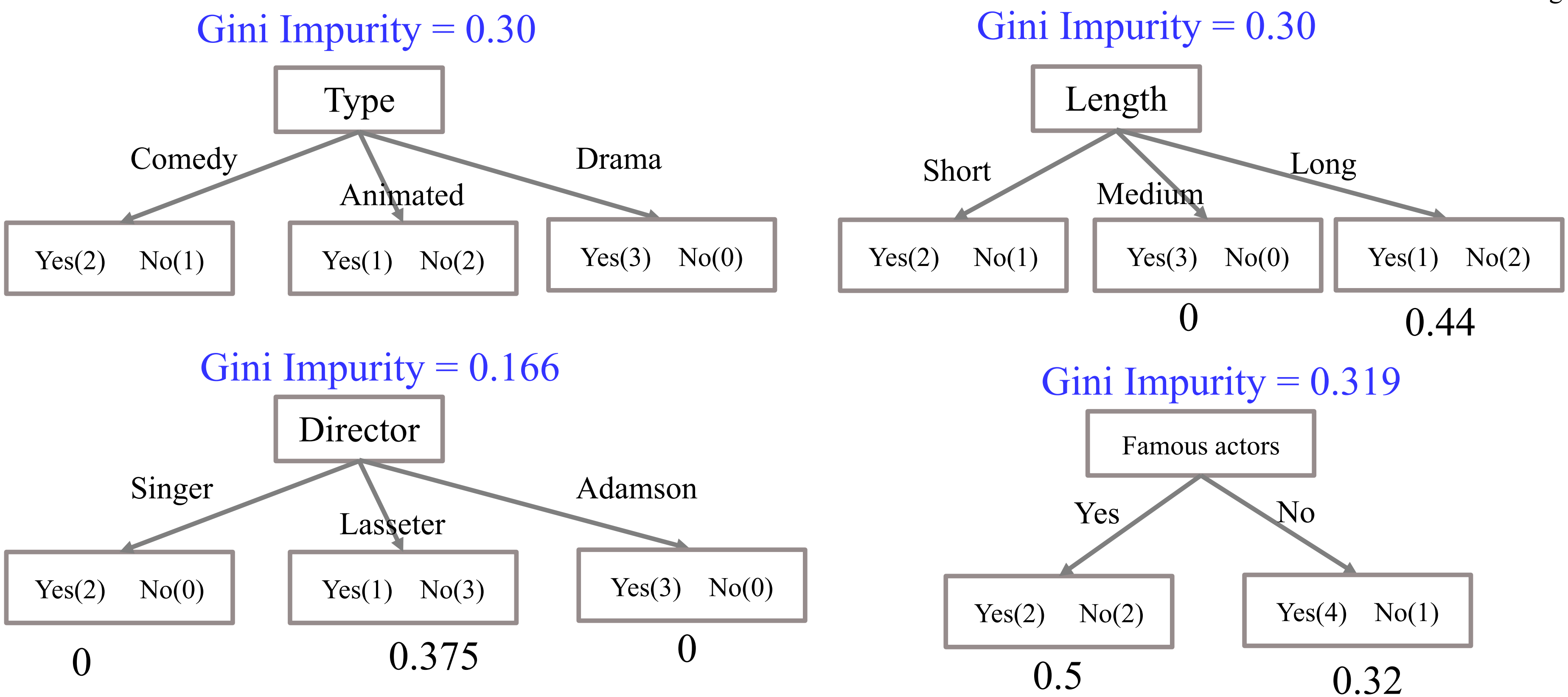
Total **Gini impurity =** weighted average of **Gini impurity for the leaves**

$$= \frac{3}{9} * 0.44 + \frac{3}{9} * 0.44 + \frac{3}{9} * 0 \approx 0.30$$

The total number of movies the first leaf.

The total number of movies on the all leaves

So, the Gini impurity for Type features is 0.30



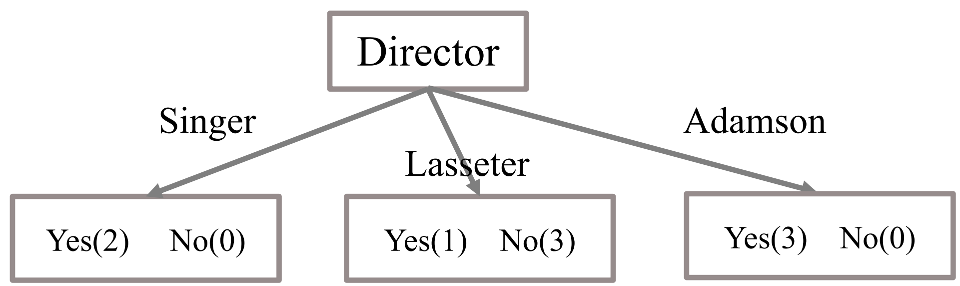
Likewise, Gini impurity for all of the features' tree are calculated.

Compare Gini impurity values for **Type, Length, Director** and **Famous actor.**

To decide which feature should be at very top of the tree.

We know that **Director feature** has **the lowest Gini impurity**, so we put it on the top of tree.

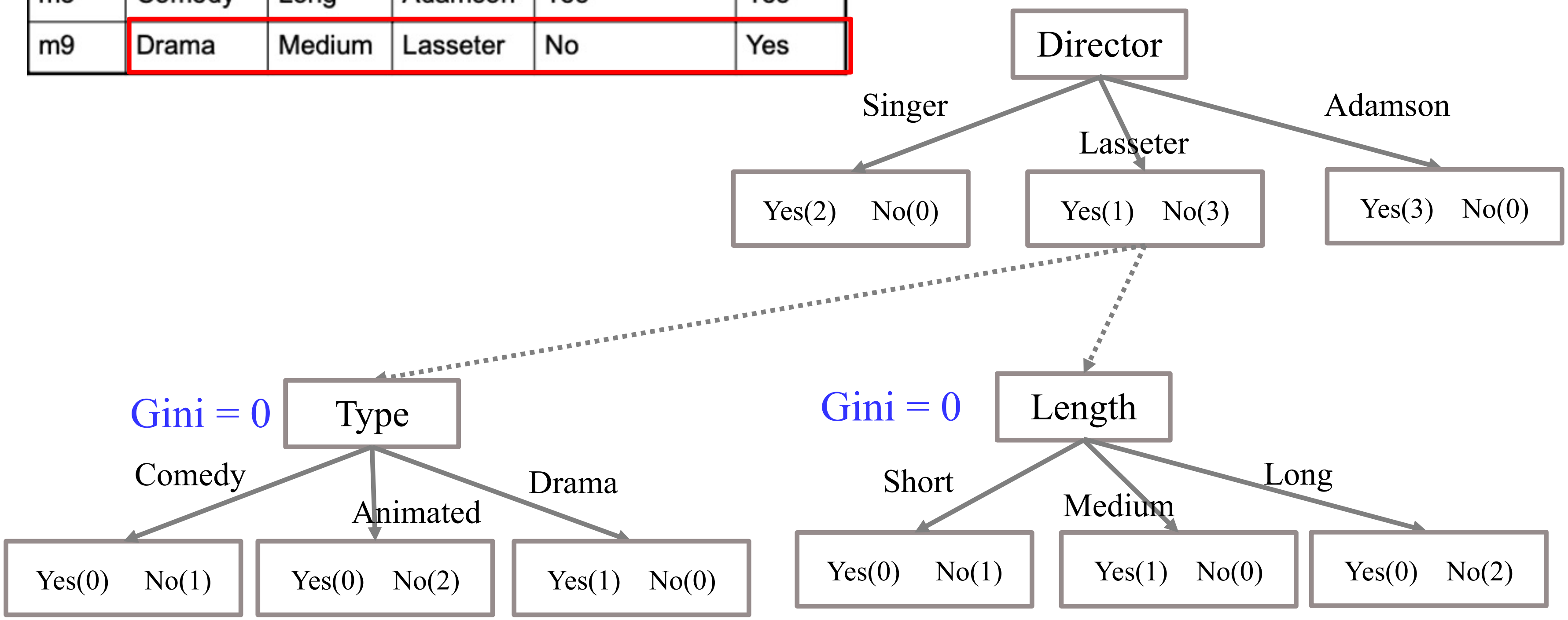
Movie	Type	Length	Director	Famous actors	Liked?
m1	Comedy	Short	Adamson	No	Yes
m2	Animated	Short	Lasseter	No	No
m3	Drama	Medium	Adamson	No	Yes
m4	animated	long	Lasseter	Yes	No
m5	Comedy	Long	Lasseter	Yes	No
m6	Drama	Medium	Singer	Yes	Yes
m7	animated	Short	Singer	No	Yes
m8	Comedy	Long	Adamson	Yes	Yes
m9	Drama	Medium	Lasseter	No	Yes



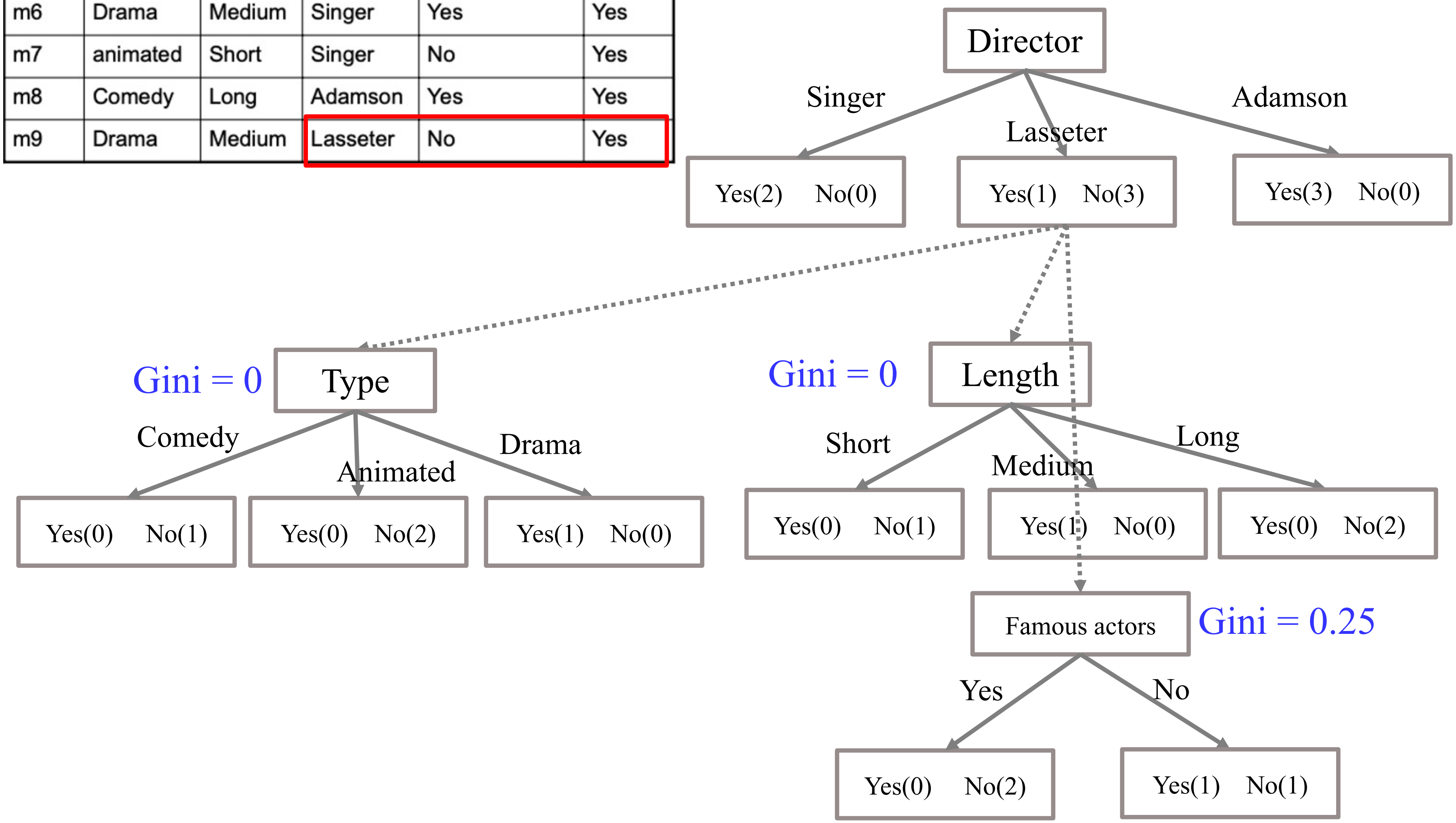
Let’s us see if we can reduce the impurity by splitting this node by other features.

So, this node is impure

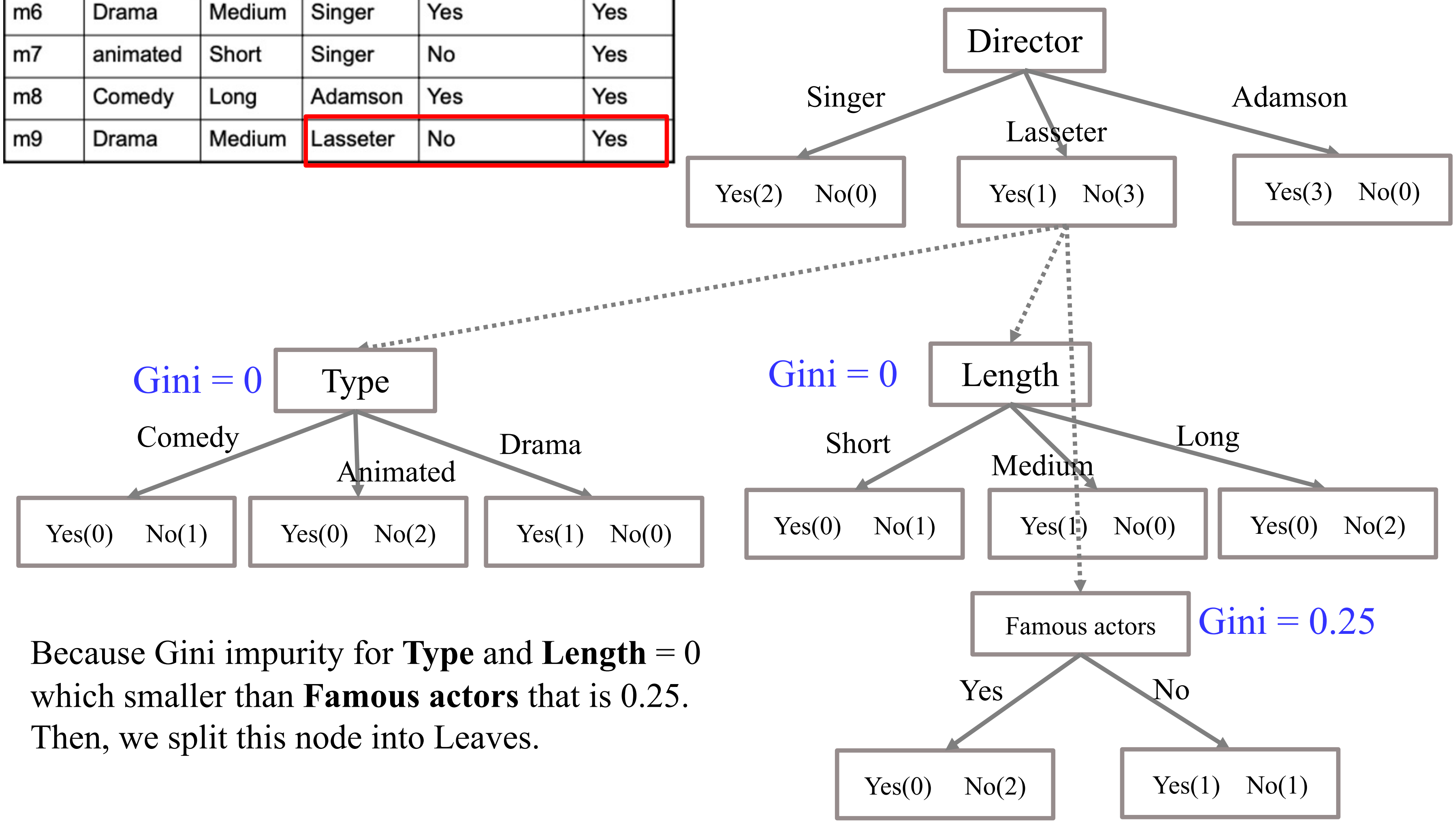
Movie	Type	Length	Director	Famous actors	Liked?
m1	Comedy	Short	Adamson	No	Yes
m2	Animated	Short	Lasseter	No	No
m3	Drama	Medium	Adamson	No	Yes
m4	animated	long	Lasseter	Yes	No
m5	Comedy	Long	Lasseter	Yes	No
m6	Drama	Medium	Singer	Yes	Yes
m7	animated	Short	Singer	No	Yes
m8	Comedy	Long	Adamson	Yes	Yes
m9	Drama	Medium	Lasseter	No	Yes



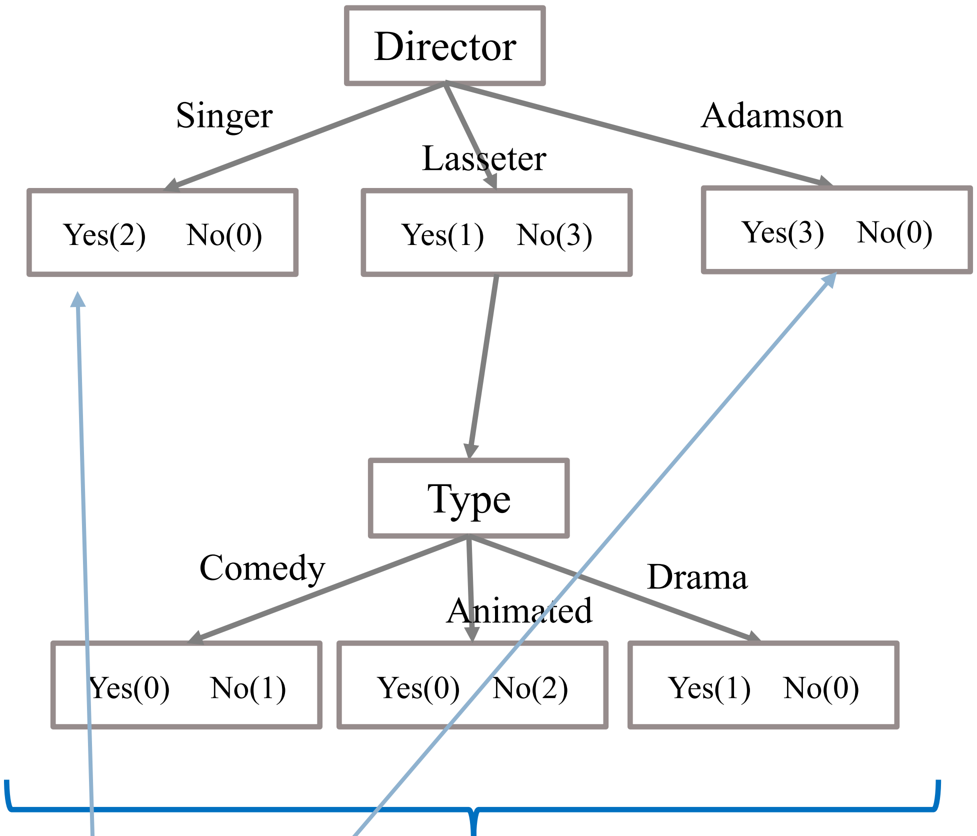
Movie	Type	Length	Director	Famous actors	Liked?
m1	Comedy	Short	Adamson	No	Yes
m2	Animated	Short	Lasseter	No	No
m3	Drama	Medium	Adamson	No	Yes
m4	animated	long	Lasseter	Yes	No
m5	Comedy	Long	Lasseter	Yes	No
m6	Drama	Medium	Singer	Yes	Yes
m7	animated	Short	Singer	No	Yes
m8	Comedy	Long	Adamson	Yes	Yes
m9	Drama	Medium	Lasseter	No	Yes



Movie	Type	Length	Director	Famous actors	Liked?
m1	Comedy	Short	Adamson	No	Yes
m2	Animated	Short	Lasseter	No	No
m3	Drama	Medium	Adamson	No	Yes
m4	animated	long	Lasseter	Yes	No
m5	Comedy	Long	Lasseter	Yes	No
m6	Drama	Medium	Singer	Yes	Yes
m7	animated	Short	Singer	No	Yes
m8	Comedy	Long	Adamson	Yes	Yes
m9	Drama	Medium	Lasseter	No	Yes



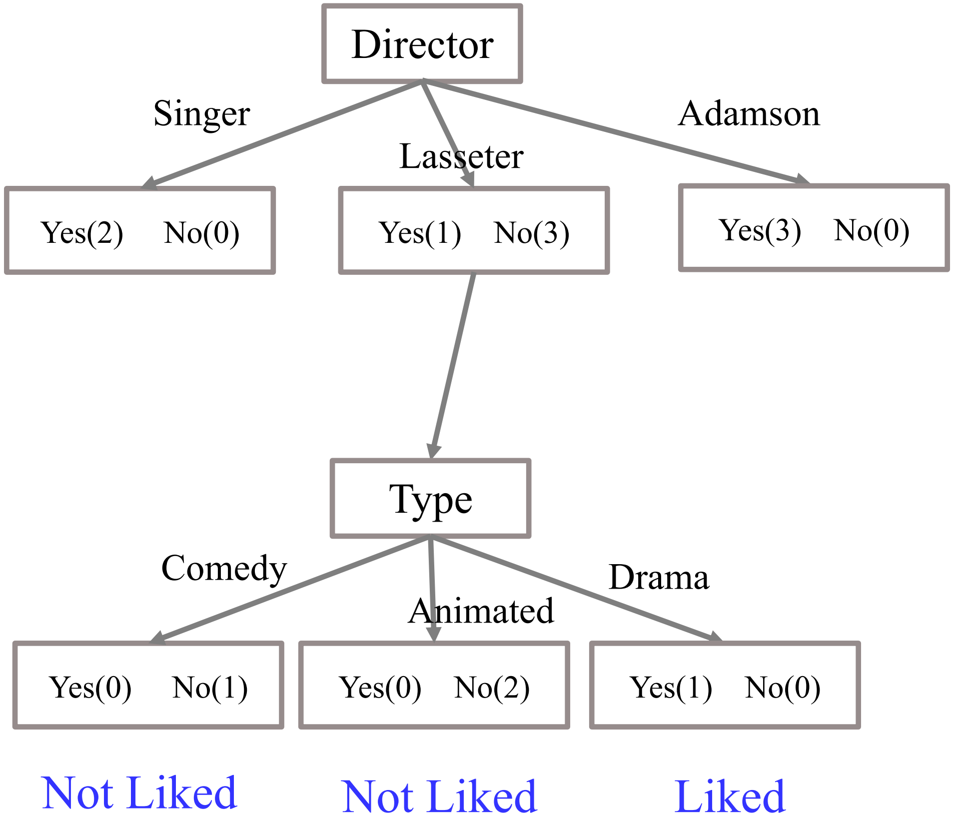
Because Gini impurity for **Type** and **Length** = 0 which smaller than **Famous actors** that is 0.25. Then, we split this node into Leaves.

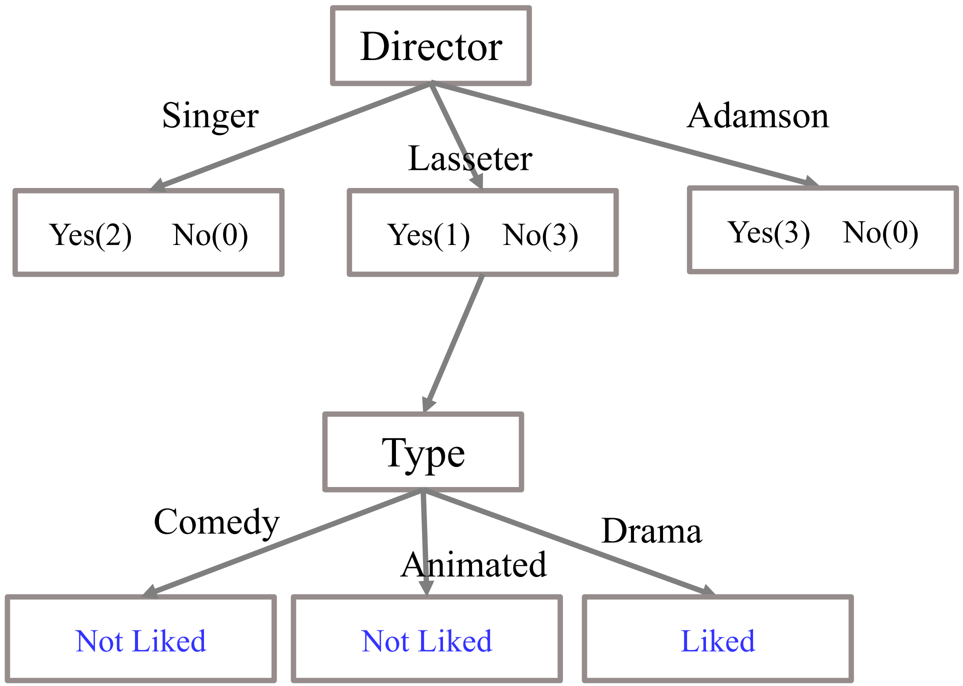


These are the leaves, because there is no reason to continue to splitting them into smaller groups.

Likewise, these nodes are leaves.

Last thing is that we need to assign output value for each **leaf**.


Director
Singer
Lasseter
Adamson
Yes(2) No(0)
Yes(1) No(3)
Yes(3) No(0)
Type
Comedy
Animated
Drama
Yes(0) No(1)
Yes(0) No(2)
Yes(1) No(0)
Not Liked
Not Liked
Liked



New samples:

Type	**Length**	**Director**	**Fanous actor**
Animated	Short	Lasseter	?
Drama	Medium	Adamson	?

Building a decision tree

- There are three ways to build a decision tree
  - Gini impurity
  - Entropy (minimizes the entropy)
  - Information gain (maximizes the information gain at each node)

Example: Root Attribute

$$Entropy = \sum log_2\left(\frac{1}{P(x)}\right)\mathrm{P}(x)$$

Movie	Type	Length	Director	Famous actors	Liked?
m1	Comedy	Short	Adamson	No	Yes
m2	Animated	Short	Lasseter	No	No
m3	Drama	Medium	Adamson	No	Yes
m4	animated	long	Lasseter	Yes	No
m5	Comedy	Long	Lasseter	Yes	No
m6	Drama	Medium	Singer	Yes	Yes
m7	animated	Short	Singer	No	Yes
m8	Comedy	Long	Adamson	Yes	Yes
m9	Drama	Medium	Lasseter	No	Yes

$$E(Liked) = \frac{6}{9}log_2\left(\frac{1}{\frac{6}{9}}\right) + \frac{3}{9}log_2\left(\frac{1}{\frac{3}{9}}\right) = 0.92$$

$$E(Liked \mid len) = ?$$

$$E(Liked \mid Type) = ?$$

$$E(Liked \mid Director) = ?$$

$$E(Liked \mid Famous\ actor) = ?$$

$$E(Liked \mid len) = P(len = \text{“}S\text{”})E(Liked \mid len = \text{“}S\text{”})$$
$$+ P(len = \text{“}M\text{”})E(Liked \mid len = \text{“}M\text{”})$$
$$+ P(len = \text{“}L\text{”})E(Liked \mid len = \text{“}L\text{”})$$
$$= \frac{3}{9} * 0.92 + \frac{3}{9} * 0 + \frac{3}{9} * 0.92 = 0.62$$

Example: Root Attribute

$$Entropy = \sum log_2\left(\frac{1}{P(x)}\right) \mathrm{P}(x)$$

Movie	Type	Length	Director	Famous actors	Liked?
m1	Comedy	Short	Adamson	No	Yes
m2	Animated	Short	Lasseter	No	No
m3	Drama	Medium	Adamson	No	Yes
m4	animated	long	Lasseter	Yes	No
m5	Comedy	Long	Lasseter	Yes	No
m6	Drama	Medium	Singer	Yes	Yes
m7	animated	Short	Singer	No	Yes
m8	Comedy	Long	Adamson	Yes	Yes
m9	Drama	Medium	Lasseter	No	Yes

$$E(Liked) = \frac{6}{9} log_2\left(\frac{1}{\frac{6}{9}}\right) + \frac{3}{9} log_2\left(\frac{1}{\frac{3}{9}}\right) = 0.92$$

$$E(Liked \mid len) = 0.62$$

$$E(Liked \mid Type) = 0.62$$

$$E(Liked \mid Director) = 0.36$$

$$E(Liked \mid Famous\ actor) = 0.85$$

Information gain: $IG(Liked \mid len) = E(Liked) - E(Liked \mid len) = 0.92 - 0.62 = 0.3$

$$IG(Liked \mid Type) = E(Liked) - E(Liked \mid Type) = 0.92 - 0.62 = 0.3$$

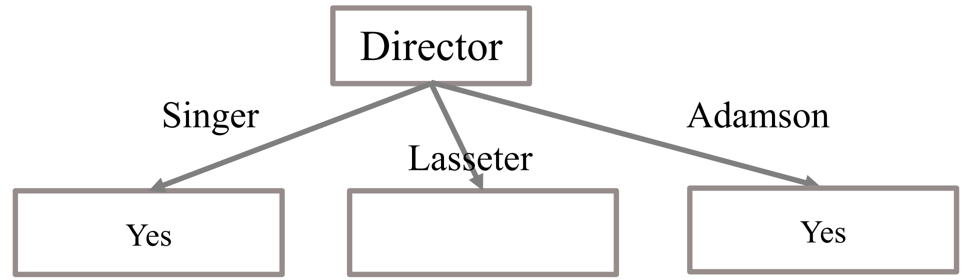
$$\boxed{IG(Liked \mid Director) = E(Liked) - E(Liked \mid Director) = 0.92 - 0.36 = 0.55}$$

$$IG(Liked \mid Famous\ actor) = E(Liked) - E(Liked \mid Famous\ actor) = 0.92 - 0.85 = 0.06$$

Example: Root Attribute

Movie	Type	Length	Director	Famous actors	Liked?
m1	Comedy	Short	Adamson	No	Yes
m2	Animated	Short	Lasseter	No	No
m3	Drama	Medium	Adamson	No	Yes
m4	animated	long	Lasseter	Yes	No
m5	Comedy	Long	Lasseter	Yes	No
m6	Drama	Medium	Singer	Yes	Yes
m7	animated	Short	Singer	No	Yes
m8	Comedy	Long	Adamson	Yes	Yes
m9	Drama	Medium	Lasseter	No	Yes

We only need to focus on the records (samples) associated with this node.



We eliminated the 'director' attribute. All samples have the same director

Movie	Type	Length	Famous actors	Liked ?
m2	Animated	Short	No	No
m4	animated	Long	Yes	No
m5	Comedy	Long	Yes	No
m9	Drama	Medium	No	Yes

$$E(Liked\ ) = 0.81$$

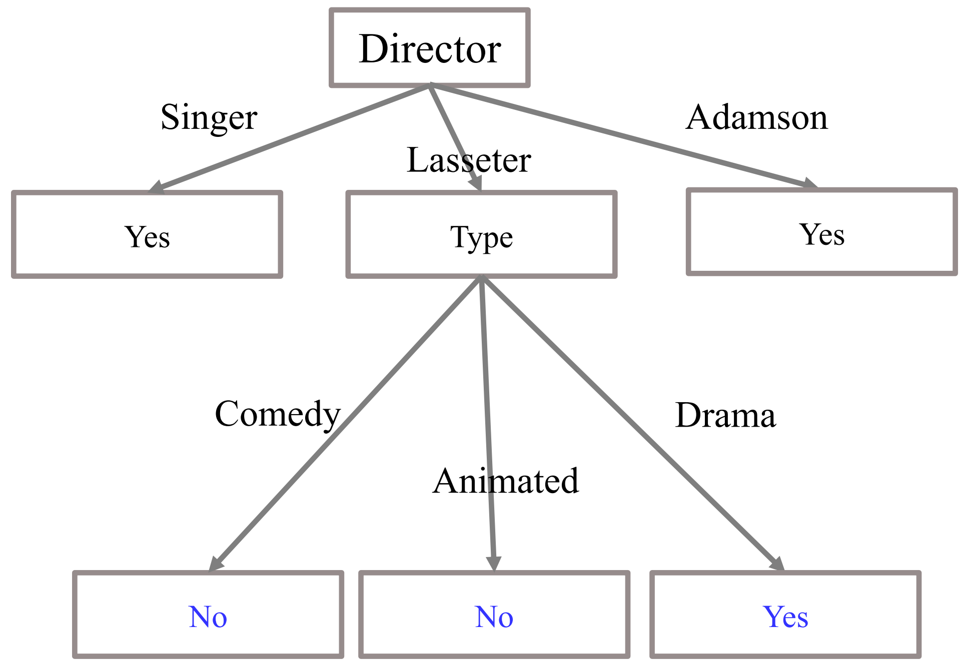
$$E(Liked\ |\ len) = 0 \qquad \boxed{IG = 0.81}$$

$$E(Liked\ |\ Type) = 0 \qquad IG = 0.81$$

$$E(Liked\ |\ Famous\ actor) = 0.5 \qquad IG = 0.31$$

Example: Root Attribute

Movie	Type	Length	Famous actors	Liked ?
m2	Animated	Short	No	No
m4	animated	long	Yes	No
m5	Comedy	Long	Yes	No
m9	Drama	Medium	No	Yes



This is final tree!

Additional points

- The algorithm we gave reaches homogonous nodes (or runs out of attributes)
- This is dangerous: For datasets with many (non relevant) attributes the algorithm will continue to split nodes.
- This will lead to overfitting!

Avoiding overfitting: Tree pruning

- Split data into train and test set
- Build tree using training set

  - for all internal nodes (starting at the root);

  - remove sub tree rooted at node;
  - assign class to be the most common among training set - check test data error;

  - if error is lower, keep change;

  - otherwise restore subtree, repeat for all nodes in subtree;

- Alternatively, we can **put limit on how trees grow**, for example, by requiring a certain number (evaluated empirically) or more movies per leaf.

Important points

- Discriminative classifiers
- Building decision trees
  - Gini impurity (minimizes the Gini impurity)
  - Entropy (minimizes the entropy)
  - Information gain (maximizes the information gain at each node)

Ranking classifiers

Table 2. Normalized scores for each learning algorithm by metric (average over eleven problems)

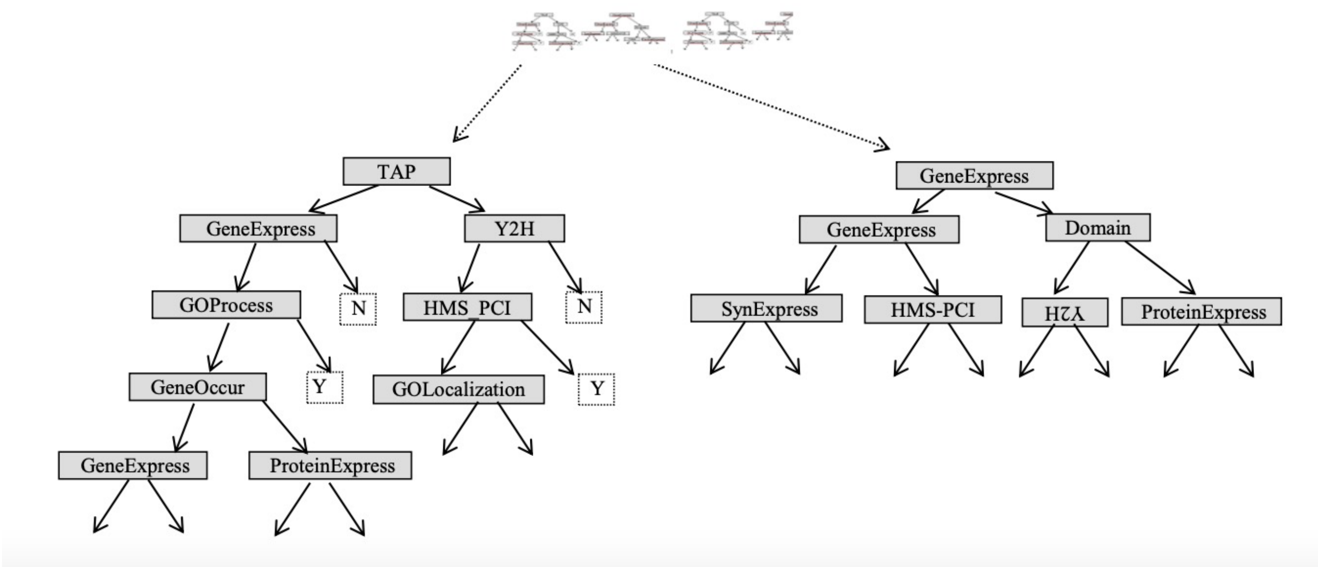
MODEL	CAL	ACC	FSC	LFT	ROC	APR	BEP	RMS	MXE	MEAN	OPT-SEL
BST-DT	PLT	.843*	.779	**.939**	**.963**	**.938**	.929*	**.880**	**.896**	**.896**	**.917**
RF	PLT	.872*	.805	.934*	.957	.931	**.930**	.851	.858	.892	.898
BAG-DT	–	.846	.781	.938*	.962*	.937*	.918	.845	.872	.887*	.899
BST-DT	ISO	.826*	.860*	.929*	.952	.921	.925*	.854	.815	.885	.917*
RF	–	**.872**	.790	.934*	.957	.931	**.930**	.829	.830	.884	.890
BAG-DT	PLT	.841	.774	.938*	.962*	.937*	.918	.836	.852	.882	.895
RF	ISO	.861*	**.861**	.923	.946	.910	.925	.836	.776	.880	.895
BAG-DT	ISO	.826	.843*	.933*	.954	.921	.915	.832	.791	.877	.894
SVM	PLT	.824	.760	.895	.938	.898	.913	.831	.836	.862	.880
ANN	–	.803	.762	.910	.936	.892	.899	.811	.821	.854	.885
SVM	ISO	.813	.836*	.892	.925	.882	.911	.814	.744	.852	.882
ANN	PLT	.815	.748	.910	.936	.892	.899	.783	.785	[illegible]	[illegible]
ANN	ISO	.803	.836	.908	.924	.876	.891	.777	.718	[illegible]	[illegible]
BST-DT	–	.834*	.816	**.939**	**.963**	**.938**	.929*	.598	.605	[illegible]	[illegible]
KNN	PLT	.757	.707	.889	.918	.872	.872	.742	.764	[illegible]	[illegible]
KNN	–	.756	.728	.889	.918	.872	.872	.729	.718	[illegible]	[illegible]
KNN	ISO	.755	.758	.882	.907	.854	.869	.738	.706	[illegible]	[illegible]
BST-STMP	PLT	.724	.651	.876	.908	.853	.845	.716	.754	[illegible]	[illegible]
SVM	–	.817	.804	.895	.938	.899	.913	.514	.467	[illegible]	[illegible]
BST-STMP	ISO	.709	.744	.873	.899	.835	.840	.695	.646	[illegible]	[illegible]
BST-STMP	–	.741	.684	.876	.908	.853	.845	.394	.382	[illegible]	[illegible]
DT	ISO	.648	.654	.818	.838	.756	.778	.590	.589	[illegible]	[illegible]
DT	–	.647	.639	.824	.843	.762	.777	.562	.607	[illegible]	[illegible]
DT	PLT	.651	.618	.824	.843	.762	.777	.575	.594	[illegible]	[illegible]
LR	–	.636	.545	.823	.852	.743	.734	.620	.645	[illegible]	[illegible]
LR	ISO	.627	.567	.818	.847	.735	.742	.608	.589	[illegible]	[illegible]
LR	PLT	.630	.500	.823	.852	.743	.734	.593	.604	[illegible]	[illegible]
NB	ISO	.579	.468	.779	.820	.727	.733	.572	.555	.654	.661
NB	PLT	.576	.448	.780	.824	.738	.735	.537	.559	.650	.654
NB	–	.496	.562	.781	.825	.738	.735	.347	-.633	.481	.489

Top 8 are all based on various extensions of decision trees

Rich Caruana & Alexandru Niculescu-Mizil, An Empirical Comparison of Supervised Learning Algorithms, ICML 2006

Random forest

- A collection of decision trees.
- For each tree we select a subset of the attributes and build tree using just these attributes.
- An input sample is classified using **majority voting**.



- Thank you!