Decision Trees

Blackbox Machine learning

Q Estimate in-game win probability for American Football

↳ as a function of game-state

$\mathbb{P}(Td) = 0.2$
$\mathbb{P}(fg) = 0.2$
$\mathbb{P}(\text{turnover}) = 0.1$
$\mathbb{P}(\text{punt}) = 0.5$

* Why?
  - Betting
  - Player valuation
  - strategic decision making

    make the decision which maximizes our win probability.

* Mathematical Models:
  - state-space models & dynamic Programming
    - States: {# possessions Remaining, score of team A, score of team B}
    - transition probabilities → $\mathbb{P}\left(\text{next state is } S_1 \mid \text{current state is } S_0\right)$

$\mathbb{P}(2,7,0 \mid 3,0,0) = \mathbb{P}(td) = 0.2$

* Statistical Models:
  - learn entirely from historical data
  - across "similar" situations over the entire (recent) history of football, what proportion of times did the team with possession win the game?
  - Play by play NFL data is easily accessible today (since 2017) NFLFastR
  - easy to fit your favorite ML model today

– thus today everyone uses statistical models for WP

* We want to fit a ML WP model from historical data.
  ↳ as a function of game-state

Variables

outcome variable – whether the team with possession won or lost the game

1/0

game state vars:
- yardline
- down
- distance
- score differential
- game seconds remaining
- off. and def. team quality → point spread
- receive 2nd half kickoff
- Timeouts

$\beta_{11} \cdot ydl + \beta_{12} \cdot ydl^2 + \beta_{13} \cdot ydl^3 + \ldots$

categorical

β_{21} down1 + β_{22} down2 + β_{23} down3 + β_{24} down4

* Last week: Logistic Regression

$$\mathbb{P}(\text{Win} = 1 \mid \underset{X}{\text{game-state}}) = \frac{1}{1 + \exp\left(-\left(\beta_1 \cdot ydl + \beta_2 \,\text{down} + \beta_3 \cdot \text{dist} + \beta_4 \cdot \text{time rem.} + \beta_5 \cdot \text{game sec rem} + \ldots\right)\right)}$$

What's wrong with this?

* These variables are all <u>interacting</u> but regression models are fundamentally additive (non-interacting)

<u>ex</u> scorediff and time Remaining are def. interacting

$$\beta_3 \cdot \text{sign}[\text{score diff}] \cdot e^{-(\text{score diff})\cdot(\text{time Rem})}$$

$$\beta_1 \cdot \text{score diff} + \beta_2 \cdot \text{time rem} + \beta_3 \cdot \underbrace{(\text{score diff}) \cdot (\text{time Rem})}$$

* You can model interactions in additive regression settings
but this is extremely difficult with 2 numeric/continuous vars and it is easy with indicator / binary vars

linear: $\beta \cdot X$

<u>Shane</u> $\beta_1 \cdot D_1 + \beta_2 \cdot D_2 + \beta_3 \cdot D_1 \cdot D_2$

D_1 = indicator

$$P(\text{out}) = \frac{1}{1 + \exp(-(\qquad))}$$

All of these variables are interacting and nonlinear.

Logistic regression is a ~~terrible~~ bad idea.

→ Machine Learning!

Learn an arbitrarily complex relationship between variables, given enough data.

* Focus on using WP for strategic decision making.

$\boxed{\text{Q: What fourth down decision should I make in \{Go, FG, Punt\} given the game state?}}$

* The entire 4th down decision process can be defined in terms of the win probability if you have a 1st down and 10 yards to go

def $V_1(x)$ = value (win probability) of a 1st and 10 at game-state x

* value of kicking a FG on 4th down:

$$V_{FG}(x) = \mathbb{P}\begin{pmatrix}\text{make} \\ \text{FG}\end{pmatrix} \cdot V\begin{pmatrix}\text{make} \\ \text{FG}\end{pmatrix} + \mathbb{P}\begin{pmatrix}\text{miss} \\ \text{FG}\end{pmatrix} \cdot V\begin{pmatrix}\text{miss} \\ \text{FG}\end{pmatrix}$$

$$1 - V_1\begin{pmatrix} x \\ \text{except} \\ 75 \text{ yardline} \\ \text{scorediff} + 3 \end{pmatrix}$$

$$1 - V_1\begin{pmatrix} x \\ \text{except} \\ \text{flip} \\ \text{ydl} \end{pmatrix}$$

$$1 - \mathbb{P}\begin{pmatrix}\text{make} \\ \text{FG}\end{pmatrix}$$

logistic regression
outcome 1/0 made/miss FG
yardline, kicker quality,
weather, time rem.
HW

$$\mathbb{P}\left(\begin{matrix}\text{make} \\ \text{FG}\end{matrix} \,\middle|\, \text{ydl}, \text{Kq}\right)$$

$$\text{Kq} = \text{Kicks made over expected}$$

$$= \sum_{\substack{\text{his prev} \\ \text{kicks } i}} \left[\begin{cases} 1 & \text{if kick } i \text{ made} \\ 0 & \text{if not} \end{cases} - \underbrace{\mathbb{P}\left(\begin{matrix}\text{kick} \\ i \\ \text{made}\end{matrix} \,\middle|\, \text{ydl}\right)}_{\mathbb{P}\left(\begin{matrix}\text{make} \\ \text{FG}\end{matrix} \,\middle|\, \text{ydl}\right)} \right]$$

$$\text{punter qual} = \text{punt yardlines added over expected}$$

$$= \sum_{\substack{\text{his prev} \\ \text{punts } i}} \left[\left(\begin{matrix}\text{ydl } y_i \\ \text{after} \\ \text{punt } i\end{matrix}\right) - \underbrace{\mathbb{E}\left(\begin{matrix}\text{ydl } y_i \\ \text{after} \\ \text{punt } i\end{matrix} \,\middle|\, \text{ydl}\right)}_{\text{ignore punter quality}} \right]$$

* Value of Punting on 4^{th} down:

$$V_{punt}(x) = \sum_{y'} P\begin{pmatrix}\text{ydl after} \\ \text{punting} \\ \text{is } y'\end{pmatrix} \cdot \left(1 - V_1\begin{pmatrix} x \\ \text{except} \\ \text{ydl} \end{pmatrix}_{y'}\right)$$

$$= 1 - \mathbb{E}_{\substack{punt \\ y'}} V_1\begin{pmatrix} x \\ \text{ydl}, y' \end{pmatrix}$$

$$\approx 1 - V_1\begin{pmatrix} x \\ \text{ydl}, \underbrace{\mathbb{E}_{punt}(y')}_{\text{expected next yardline after punting}} \end{pmatrix}$$

- bin by ydl and take avg next ydl
- linear regression on ydl + punter quality → HW

$\mathbb{E} f(Y) \neq f \mathbb{E} Y$ in general

but in this specific case = is close enough because f, $y \mapsto V_1(y)$ is nearly <u>Linear</u>, and so $\mathbb{E} f(Y) = f \mathbb{E} Y$

* Value of going for it on 4^{th} down:

4^{th} down and z yards to go
at yardline y
game-state x

$$V_{go}(x) \approx$$

$$\underbrace{\mathbb{P}(\text{convert})}_{\mathbb{P}(\text{gain} \geq z \text{ yards})} \cdot V_1\begin{pmatrix} x \\ \text{except} \\ \text{new ydl} \end{pmatrix} + \left(1-\mathbb{P}(\text{convert})\right) \cdot \left(1 - V_1\begin{pmatrix} x \\ \text{except} \\ \text{opp. ydl} \end{pmatrix}\right)$$

new ydl $\to y - z$

opp. ydl $\to 100-y$

ydl, y = y yards from td

$\mathbb{P}(\text{gain} \geq z \text{ yards})$

off. def. team quality
dist
yardline
outcome var: 1/0, 1 = conversion

$\longrightarrow$ Logistic Regression

HW

* V_{go}, V_{fg}, V_{punt} on 4th down as a function of game-state x can be written in terms of

$\underbrace{V_1}_{\text{Machine Learning}}(x)$ = WP if a team has a 1st and 10 at x

and $\underbrace{\mathbb{P}(\text{make fg} \mid x),\ \mathbb{P}(\text{convert} \mid x),\ \mathbb{E}_{punt}[y' \mid x]}_{\text{Regression (HW)}}$

Task Estimate $V_1(x) = WP[x \mid 1^{st} \text{ and } 10)$ Using Machine Learning.

Game-State x yardline, Score diff, timeouts, game sec. Rem., Point spread, Receive 2nd half kickoff

Dataset

i = index of 1st and 10 play i in dataset of plays

x_i = game-state vector of play i

$y_i = 1$ if team with possession on play i wins the game else 0

Want

$$\mathbb{P}(y_i = 1 \mid x_i) = \mathbb{E}[y_i \mid x_i] = \hat{y}_i = \hat{f}(x_i)$$

$\hat{f}$ could be an arbitrarily complex nonlinear interacting function

$\downarrow$

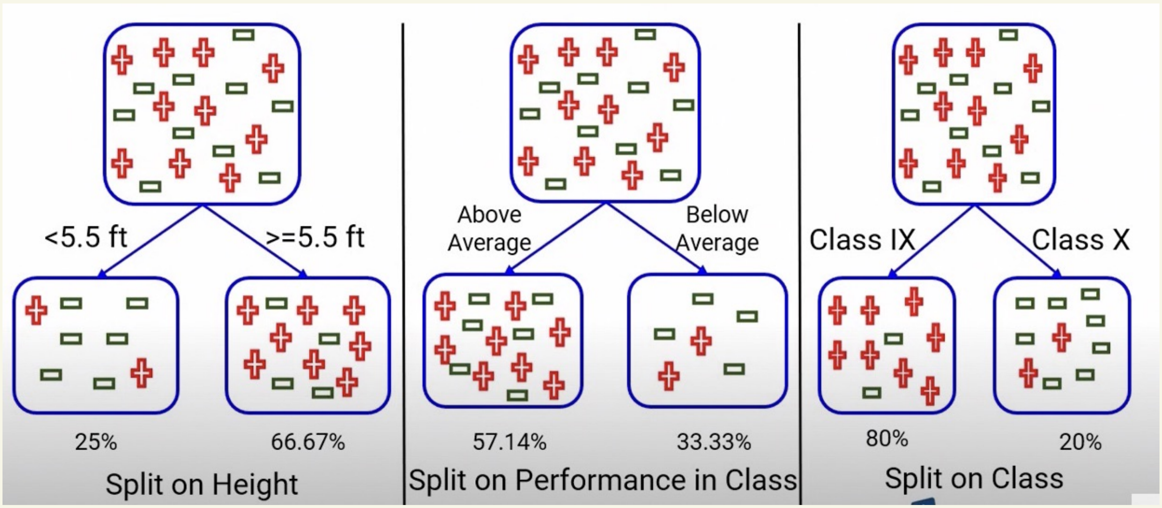
Tree Machine Learning

- decision trees / CART
  classification & regression trees
- Random Forests
- XGBoost (extreme gradient boosting)

Decision Trees

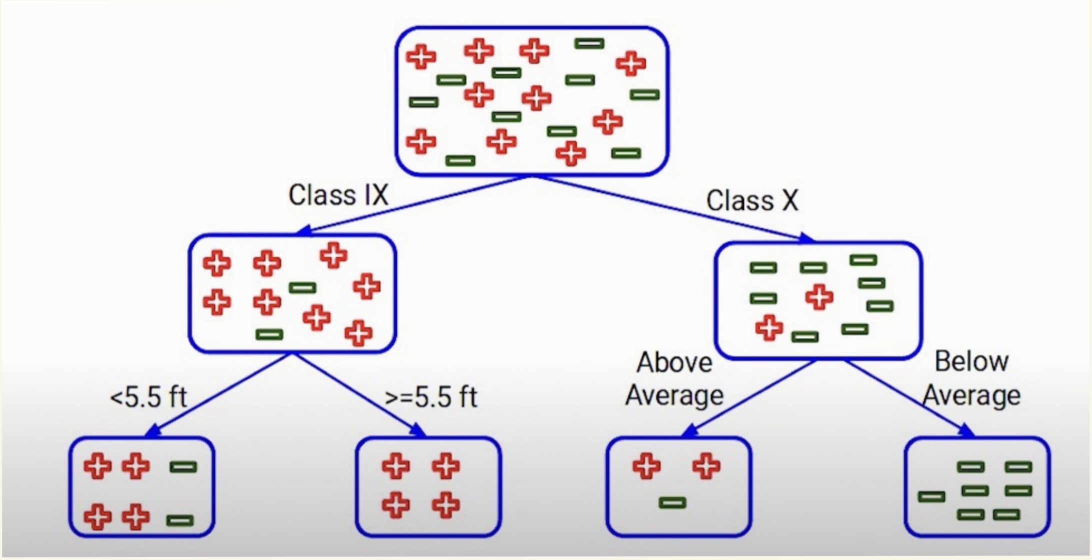
Ex 20 students in the class, 10 play cricket
fit a model to predict whether a student plays cricket
X variables: Height, grades, class

Idea split on an X variable to classify the y variable



Which split is best? The Rightmost split is best because the 80% in left bucket and 20% in Right bucket separates the classes of cricket & not cricket as best as possible.

* decision trees allow variables to interact

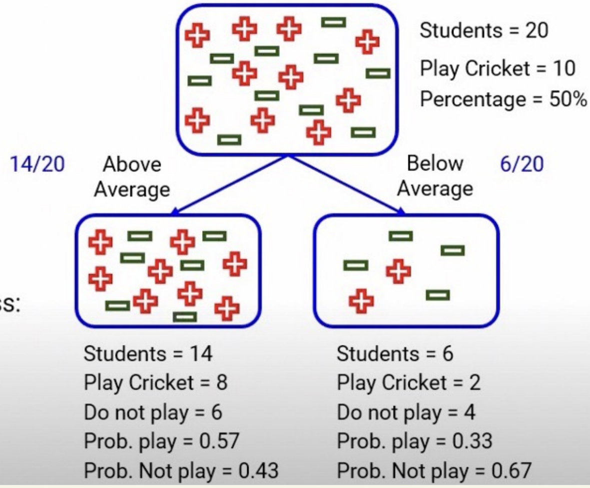


* How to actually select the best split point?

1. Gini Impurity

Split on Performance in Class

- Gini Impurity: sub-node Above Average:
  1 - [(0.57)*(0.57) + (0.43)*(0.43)] = 0.49
- Gini Impurity: sub-node Below Average:
  1 - [(0.33)*(0.33) + (0.67)*(0.67)] = 0.44
- Weighted Gini Impurity: Performance in Class:
  (14/20)*0.49 + (6/20)*0.44 = 0.475



Gini Impurity of the Above Avg Node $= 1-(P_+^2 + P_-^2) = 1-(.57)^2-(.43)^2 = .49$

Gini Impurity of the Below Avg Node $= 1-(P_+^2 + P_-^2) = 1-\left(\frac{1}{3}\right)^2-\left(\frac{2}{3}\right)^2 = .44$

fully informative: $P_+ = 1,\ P_- = 0$
OR
$P_+ = 0,\ P_- = 1$

$GI = 1-1^2-0^2 = 0$

fully non-informative: $P_+ = \frac{1}{2},\ P_- = \frac{1}{2}$

$GI = 1-\left(\frac{1}{2}\right)^2-\left(\frac{1}{2}\right)^2 = 0.5$

$$\text{Weighted Gini Impurity} = \left(\frac{\#\text{ left bin}}{\text{total }\#}\right)\left(\text{Gini left}\right) + \left(\frac{\#\text{ right bin}}{\text{total }\#}\right)\left(\text{Gini Right}\right)$$

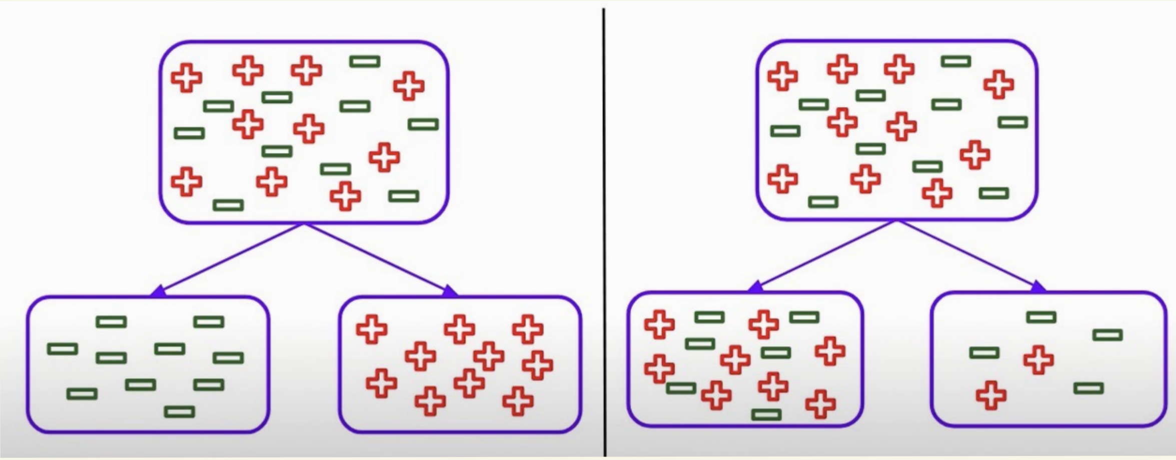
$$= \left(\frac{14}{20}\right)(.49) + \left(\frac{6}{20}\right)(.44)$$

$$= .475$$

Split	Weighted Gini Impurity
Performance in Class	0.475
$\boxed{\text{Class}}$	$\boxed{0.32}$

→ split on Class

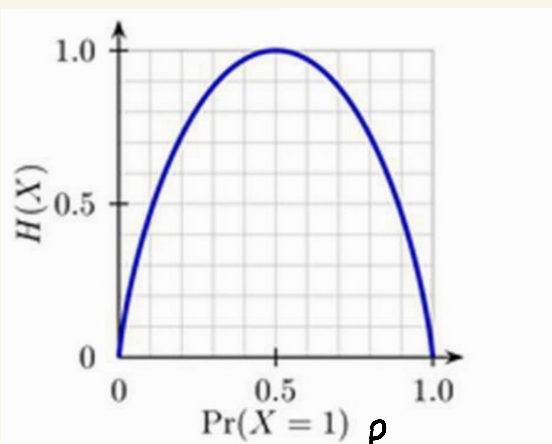
2. Information Gain / Entropy



More info gained   less info gained

Information Gain = 1 − Entropy

$$\boxed{\text{Entropy} = -p\log_2 p - (1-p)\log_2(1-p)}$$



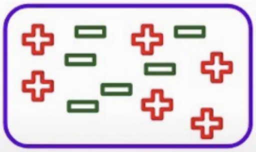
p

$p = \mathbb{P}(+) = \mathbb{P}(y=1)$

$p=0$, Entropy $= -\cancel{0\cdot\log 0} - 1\cdot\underbrace{\log 1}_{0} = 0$

$p=1$, Entropy $= -1\cdot\log 1 - 0\cdot\log 0 = 0$

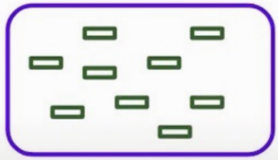
$p=\frac{1}{2}$, Entropy $= -\frac{1}{2}\underbrace{\log_2\left(\frac{1}{2}\right)}_{-1} - \frac{1}{2}\underbrace{\log_2\left(\frac{1}{2}\right)}_{-1} = 1$



% Play = 0.50
% Not play = 0.50

Entropy = - (0.5) * log2(0.5) - (0.5) * log2(0.5)

= 1



% Play = 0
% Not play = 1

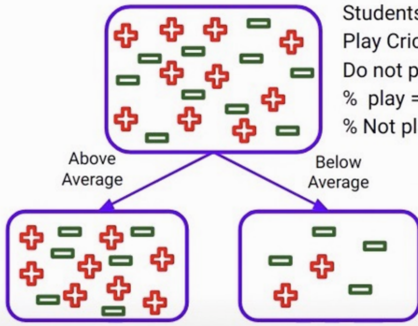
Entropy = - (0) * log2(0) - (1) * log2(1)

= 0

Split on Performance in Class

- Entropy for Parent node:
  $-(0.5)*\log_2(0.5) -(0.5)*\log_2(0.5) = 1$
- Entropy for sub-node Above Average:
  $-(0.57)*\log_2(0.57) -(0.43)*\log_2(0.43) = 0.98$
- Entropy for sub-node Below Average:
  $-(0.33)*\log_2(0.33) -(0.67)*\log_2(0.67) = 0.91$
- Weighted Entropy: Performance in Class:
  $(14/20)*0.98 + (6/20)*0.91 = 0.959$

Students = 20
Play Cricket = 10
Do not play = 10
% play = 0.5
% Not play = 0.5



Students = 14
Play Cricket = 8
Do not play = 6
% play = 0.57
% Not play = 0.43

Students = 6
Play Cricket = 2
Do not play = 4
% play = 0.33
% Not play = 0.67

Left split bin: $-(.57)\log_2(0.57) - (.43)\log_2(.43) = 0.98$

Right split bin: $(-.33)\log_2(.33) - (.66)\log_2(.66) = 0.91$

weighted entropy $= \left(\frac{14}{20}\right)(.98) + \left(\frac{6}{20}\right)(.91) = .96$

info gain = 1 - weighted entropy = 0.04

Split	Entropy	Information Gain
Performance in Class	0.959	0.041
Class	0.722	0.278

→ split on class!

* Our outcome variable $y = \{+, -\} = \{\text{plays cricket}, \text{doesn't play}\}$ is Categorical, so gini + entropy work well here.

  gini: p^2 entropy: $p \log p$

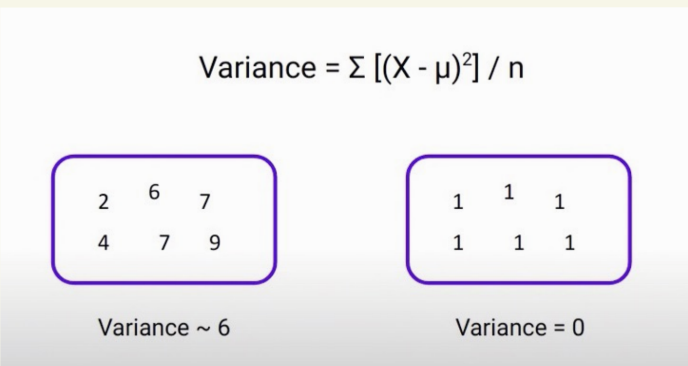
  But what if our outcome was continuous?

  e.g. $y \in \mathbb{R}$ (height)

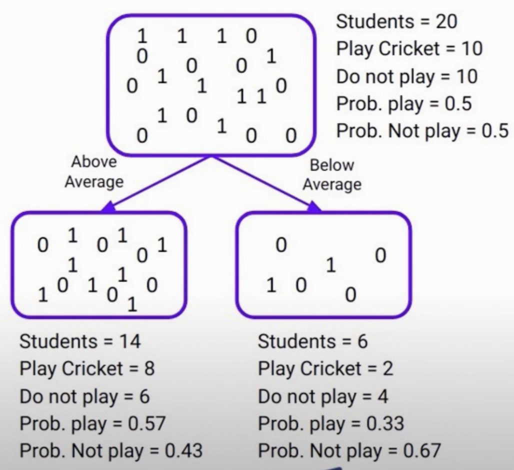
  It breaks down.

  Need another way to split.

3. Reduction in Variance



- Above Average node:
  - Mean = $(8*1 + 6*0) / 14 = 0.57$
  - Variance = $[8*(1-0.57)^2 + 6*(0-0.57)^2] / 14 = 0.245$
- Below Average node:
  - Mean = $(2*1 + 4*0) / 6 = 0.33$
  - Variance = $[2*(1-0.33)^2 + 4*(0-0.33)^2] / 6 = 0.222$
- Variance: Performance in Class:
  $(14/20)*0.245 + (6/20)*0.222 = 0.238$

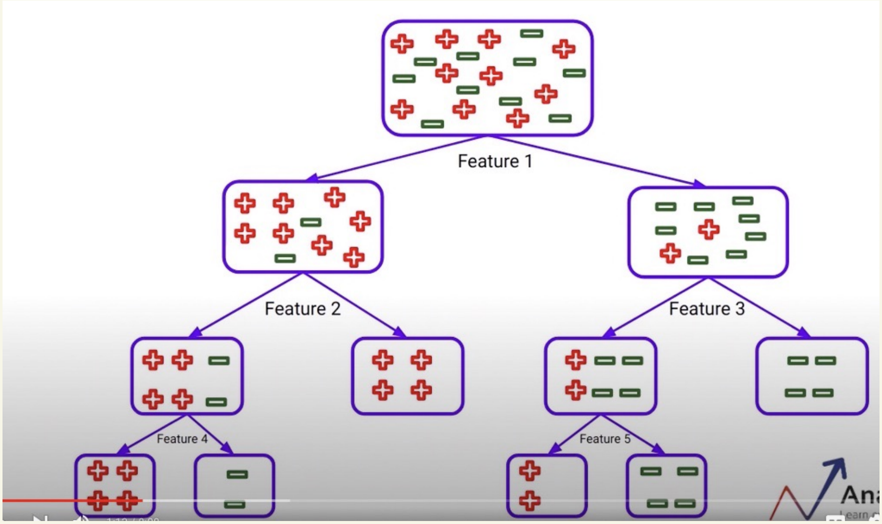


Split	Variance
Performance in Class	0.238
Class	0.16

→ split on class!

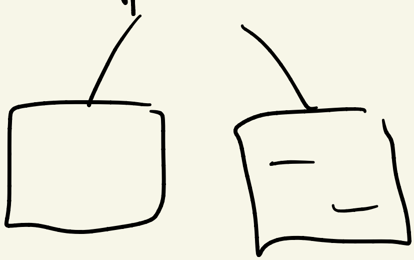
* 3+ ways to make splits.

* Could Iteratively make splits until all nodes are "pure" but this would overfit (memorize noise/randomness in the training dataset)



* Tuning Hyperparameters
(to control overfitting)

- max depth of the tree
- min samples for a node split



HW: fit a decision tree WP model

* Random Forests: How to further reduce overfitting in decision trees