Decision Trees

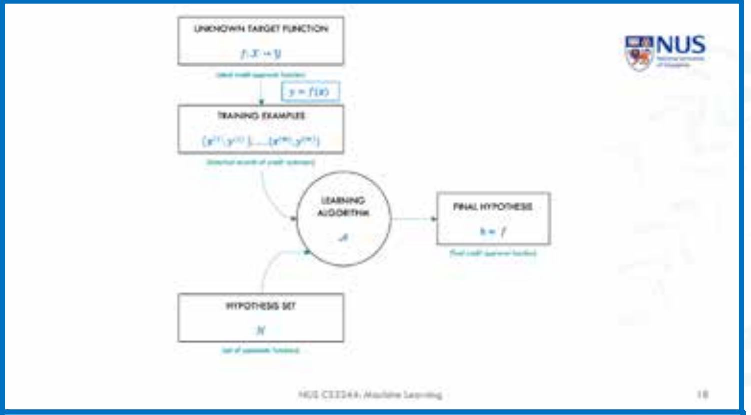
3
A

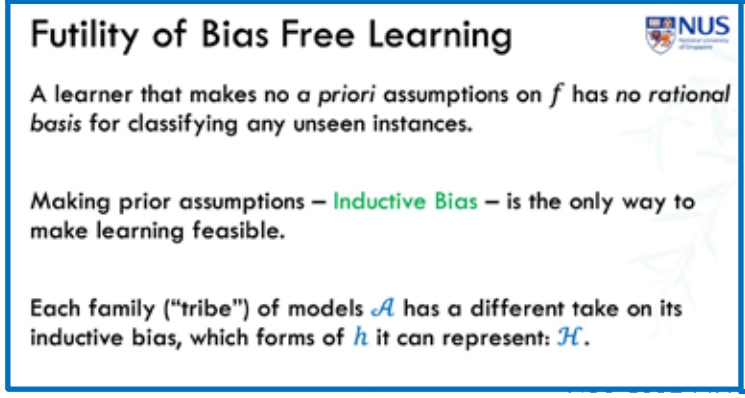
CS 3244
Machine Learning

NUS | Computing

Recap from Week 02

1. A pattern exists
2. Difficult to pin down formally (mathematically)
3. We have data for it



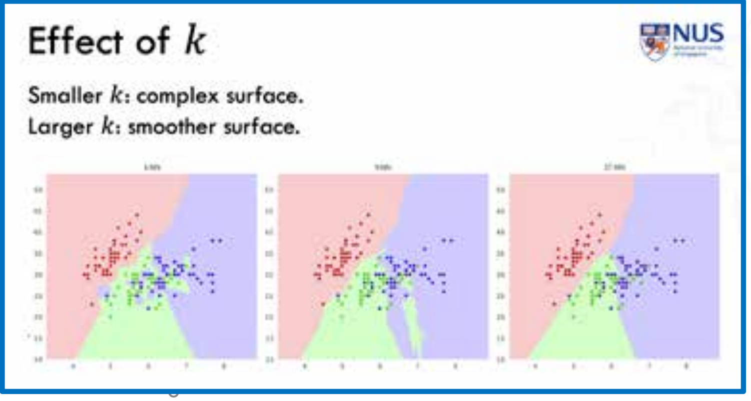


Futility of Bias Free Learning

A learner that makes no *a priori* assumptions on f has *no rational basis* for classifying any unseen instances.

Making prior assumptions – Inductive Bias – is the only way to make learning feasible.

Each family ("tribe") of models $\mathcal{A}$ has a different take on its inductive bias, which forms of h it can represent: $\mathcal{H}$.



Effect of k

Smaller k: complex surface.
Larger k: smoother surface.

Forecast for Week 03

Learning Outcomes for this week:

- Understand and implement Decision Trees;
- Master Information Gain as a means of choosing optimal features for decisions;
- Formalize cost functions as a means of evaluating a supervised learner's performance;
- Describe simple ensembling methods that integrate multiple classifiers;
    - Understand Decision Trees as an instance of an ensemble classifier of conditional decision stamps;
- Reason about inductive bias of the decision tree algorithm

Thursday: AMA and Datasets

Your subgroup may want to attend Thursday to get to talk to some of the TAs who know some of the curated datasets well.

We'll have breakout rooms for different curated datasets so you can check with the TAs what might be suitable projects, even before the proposal. 😎

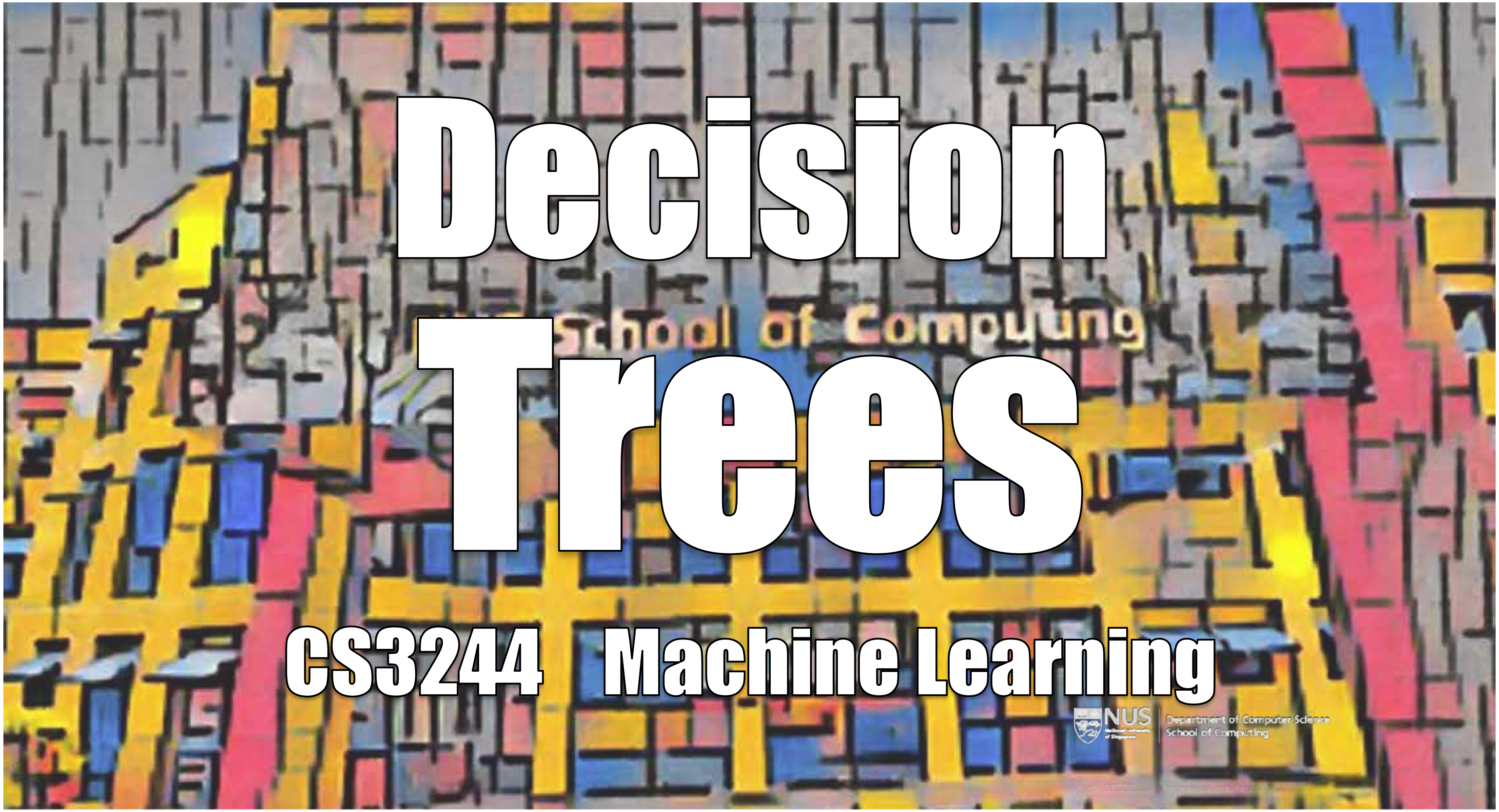
Python Tutorial

Our head TA Pranavan will be holding an optional

Numpy and Pandas Tutorial

Fri 26 Aug, 18:00–21:00 @ Seminar Room @ LT19

If you want to attend, please react on #tutorials. Note that it is likely physical only, not hybrid. First 45 🙋🏾‍♂️🙋🏼‍♀️ reacts get to go.


School of Computing
NUS National University of Singapore
Department of Computer Science
School of Computing

Decision Trees

CS3244 Machine Learning

How do people make decisions?

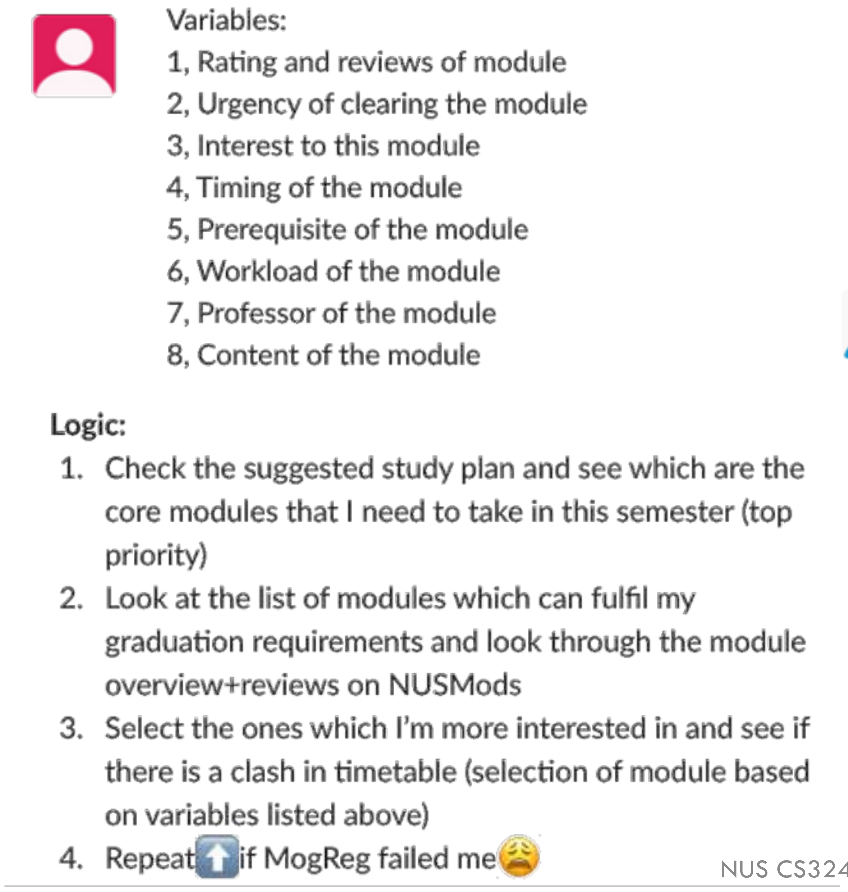
Well, let's examine some recent choices you made:

What module did you enroll in?

What did you eat today for lunch?

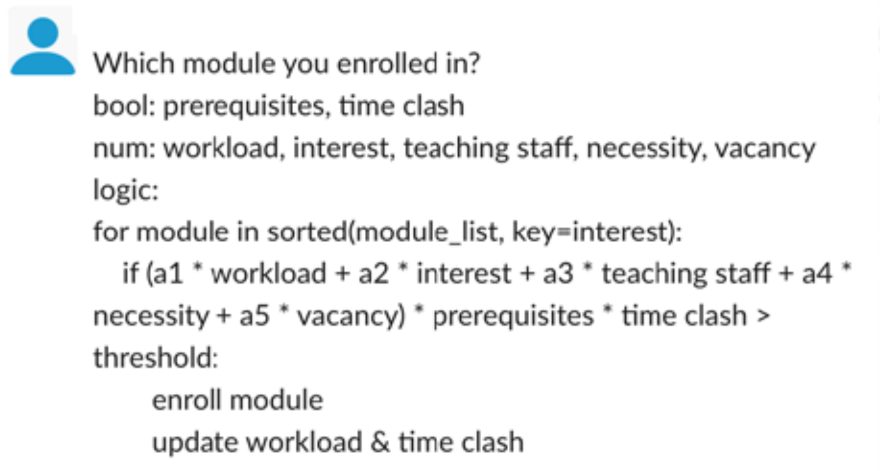
What did you wear today?

Pre-Lecture Activity from last week



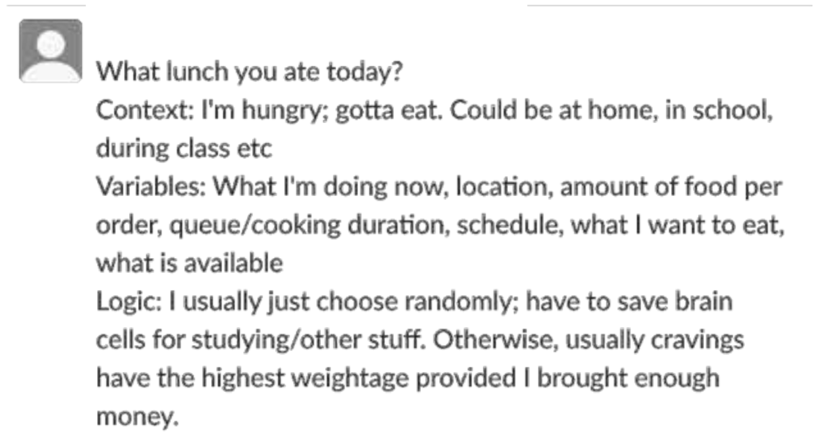
Variables:
1, Rating and reviews of module
2, Urgency of clearing the module
3, Interest to this module
4, Timing of the module
5, Prerequisite of the module
6, Workload of the module
7, Professor of the module
8, Content of the module

Logic:
1. Check the suggested study plan and see which are the core modules that I need to take in this semester (top priority)
2. Look at the list of modules which can fulfil my graduation requirements and look through the module overview+reviews on NUSMods
3. Select the ones which I'm more interested in and see if there is a clash in timetable (selection of module based on variables listed above)
4. Repeat⬆️if MogReg failed me😩

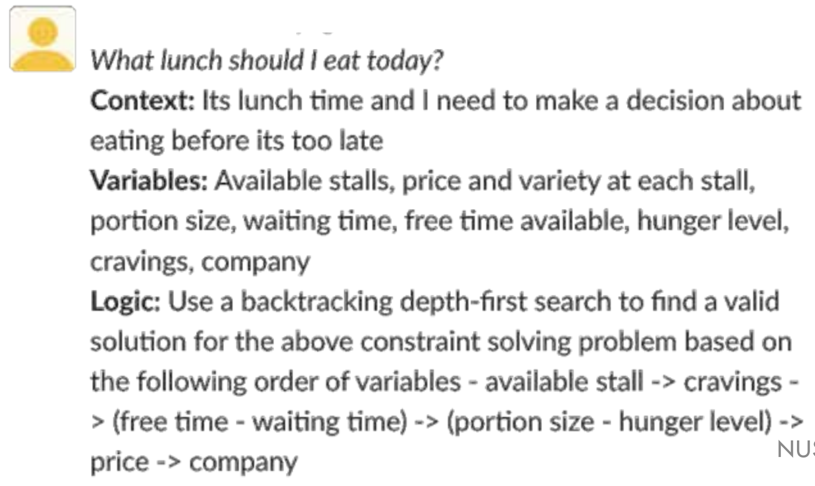


```
Which module you enrolled in?
bool: prerequisites, time clash
num: workload, interest, teaching staff, necessity, vacancy
logic:
for module in sorted(module_list, key=interest):
  if (a1 * workload + a2 * interest + a3 * teaching staff + a4 *
necessity + a5 * vacancy) * prerequisites * time clash >
threshold:
      enroll module
      update workload & time clash
```

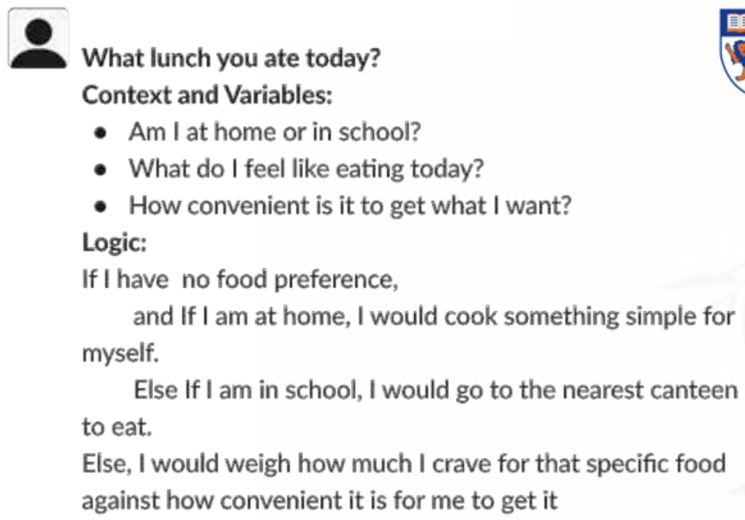
Pre-Lecture Activity from last week



What lunch you ate today?
Context: I'm hungry; gotta eat. Could be at home, in school, during class etc
Variables: What I'm doing now, location, amount of food per order, queue/cooking duration, schedule, what I want to eat, what is available
Logic: I usually just choose randomly; have to save brain cells for studying/other stuff. Otherwise, usually cravings have the highest weightage provided I brought enough money.



What lunch should I eat today?
Context: Its lunch time and I need to make a decision about eating before its too late
Variables: Available stalls, price and variety at each stall, portion size, waiting time, free time available, hunger level, cravings, company
Logic: Use a backtracking depth-first search to find a valid solution for the above constraint solving problem based on the following order of variables - available stall -> cravings -> (free time - waiting time) -> (portion size - hunger level) -> price -> company

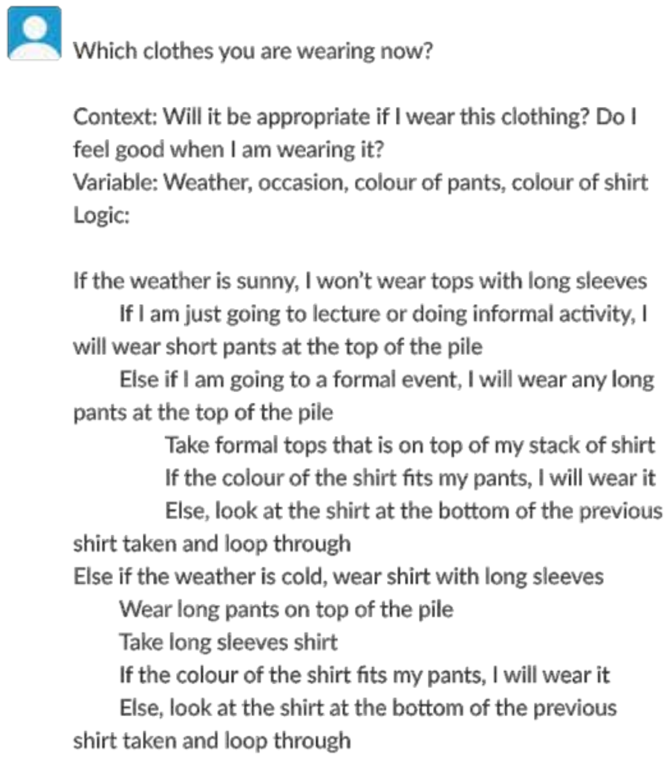


What lunch you ate today?
Context and Variables:

- Am I at home or in school?
- What do I feel like eating today?
- How convenient is it to get what I want?

Logic:
If I have no food preference,
    and If I am at home, I would cook something simple for myself.
    Else If I am in school, I would go to the nearest canteen to eat.
Else, I would weigh how much I crave for that specific food against how convenient it is for me to get it

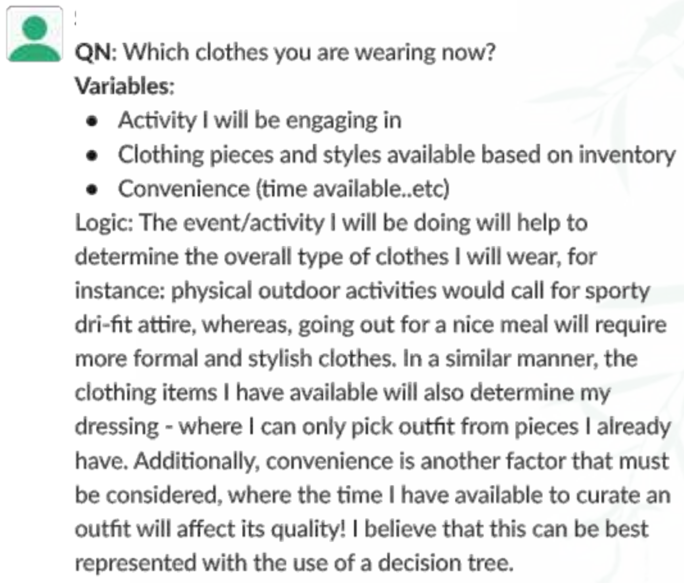
Pre-Lecture Activity from last week



Which clothes you are wearing now?

Context: Will it be appropriate if I wear this clothing? Do I feel good when I am wearing it?
Variable: Weather, occasion, colour of pants, colour of shirt
Logic:

```
If the weather is sunny, I won't wear tops with long sleeves
    If I am just going to lecture or doing informal activity, I will wear short pants at the top of the pile
    Else if I am going to a formal event, I will wear any long pants at the top of the pile
        Take formal tops that is on top of my stack of shirt
        If the colour of the shirt fits my pants, I will wear it
        Else, look at the shirt at the bottom of the previous shirt taken and loop through
Else if the weather is cold, wear shirt with long sleeves
    Wear long pants on top of the pile
    Take long sleeves shirt
    If the colour of the shirt fits my pants, I will wear it
    Else, look at the shirt at the bottom of the previous shirt taken and loop through
```



QN: Which clothes you are wearing now?
Variables:

- Activity I will be engaging in
- Clothing pieces and styles available based on inventory
- Convenience (time available..etc)

Logic: The event/activity I will be doing will help to determine the overall type of clothes I will wear, for instance: physical outdoor activities would call for sporty dri-fit attire, whereas, going out for a nice meal will require more formal and stylish clothes. In a similar manner, the clothing items I have available will also determine my dressing - where I can only pick outfit from pieces I already have. Additionally, convenience is another factor that must be considered, where the time I have available to curate an outfit will affect its quality! I believe that this can be best represented with the use of a decision tree.

How do people make decisions?

In physical subgroups or Zoom breakout rooms, do:

(1 min): Introduce yourself! 🙃

(5 mins): *What's the inductive bias of our method?*

Is it like how k nearest neighbors decides?

Ask one member to write your collective answer to the #lectures thread. Upvote others that you like.

How do people make decisions?

Would we say it's like how k nearest neighbors make a decision?
(i.e., looking at what you did for a similar item last time?)

No, not exactly. It doesn't make much sense in this instance. Instead, people often focus on just a 1 or 2 dimensions of the evidence.

Decision trees embody this paradigm.

Programming

Imperative programming control flow also embodies this:

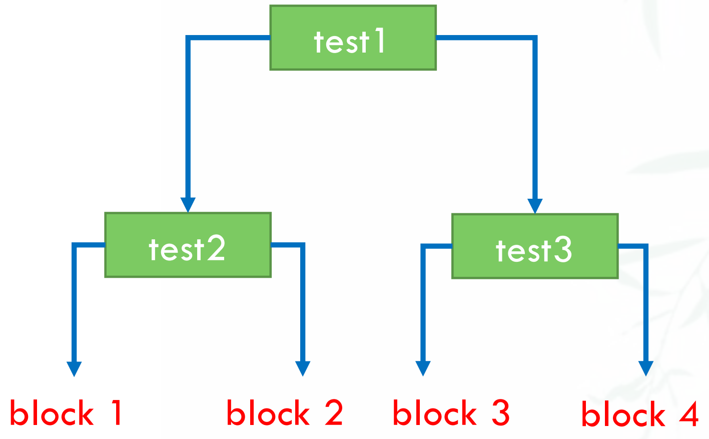
```
if test1:
  if test2:
    # block 1
  else:
    # block 2
else:
  if test3:
    # block 3
  else:
    # block 4
```

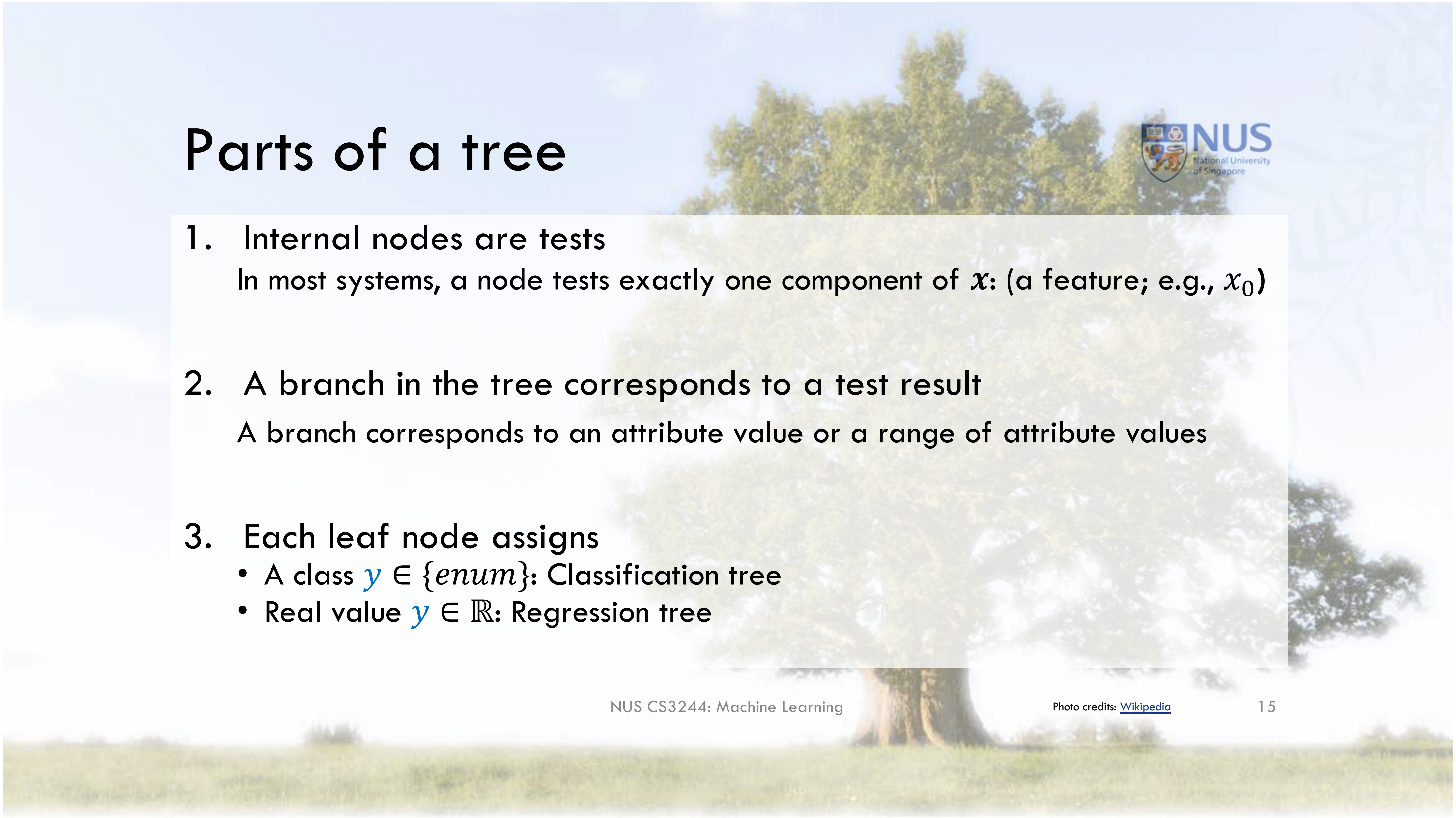
Programming

Imperative programming control flow also embodies this

```
if test1:
  if test2:
    # block 1
  else:
    # block 2
else:
  if test3:
    # block 3
  else:
    # block 4
```

This is a decision tree!





Parts of a tree

1. Internal nodes are tests

   In most systems, a node tests exactly one component of $\boldsymbol{x}$: (a feature; e.g., x_0)

2. A branch in the tree corresponds to a test result

   A branch corresponds to an attribute value or a range of attribute values

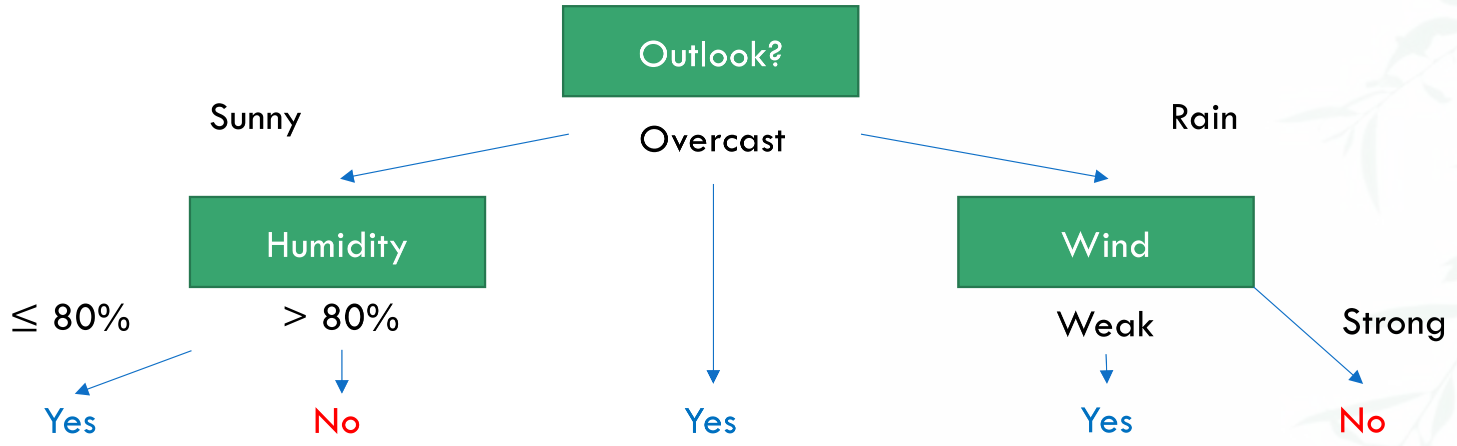
3. Each leaf node assigns
   - A class $y \in \{enum\}$: Classification tree
   - Real value $y \in \mathbb{R}$: Regression tree

Photo credits: Wikipedia

Shall we play tennis?

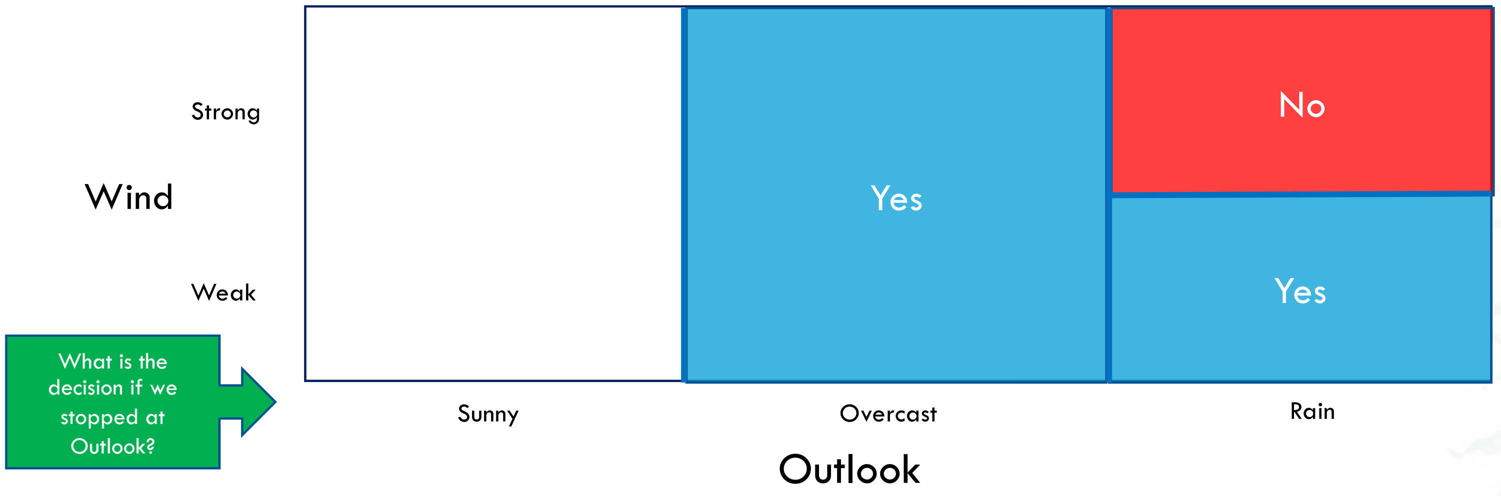
Day	Outlook	Temperature	Humidity	Wind	Play Tennis?
1	Sunny	Hot	95%	Weak	No
2	Sunny	Hot	92%	Strong	No
3	Overcast	Hot	95%	Weak	Yes
4	Rain	Mild	90%	Weak	Yes
5	Rain	Cool	60%	Weak	Yes
6	Rain	Cool	65%	Strong	No
7	Overcast	Cool	70%	Strong	Yes
8	Sunny	Cool	70%	Weak	Yes

Shall we play tennis?



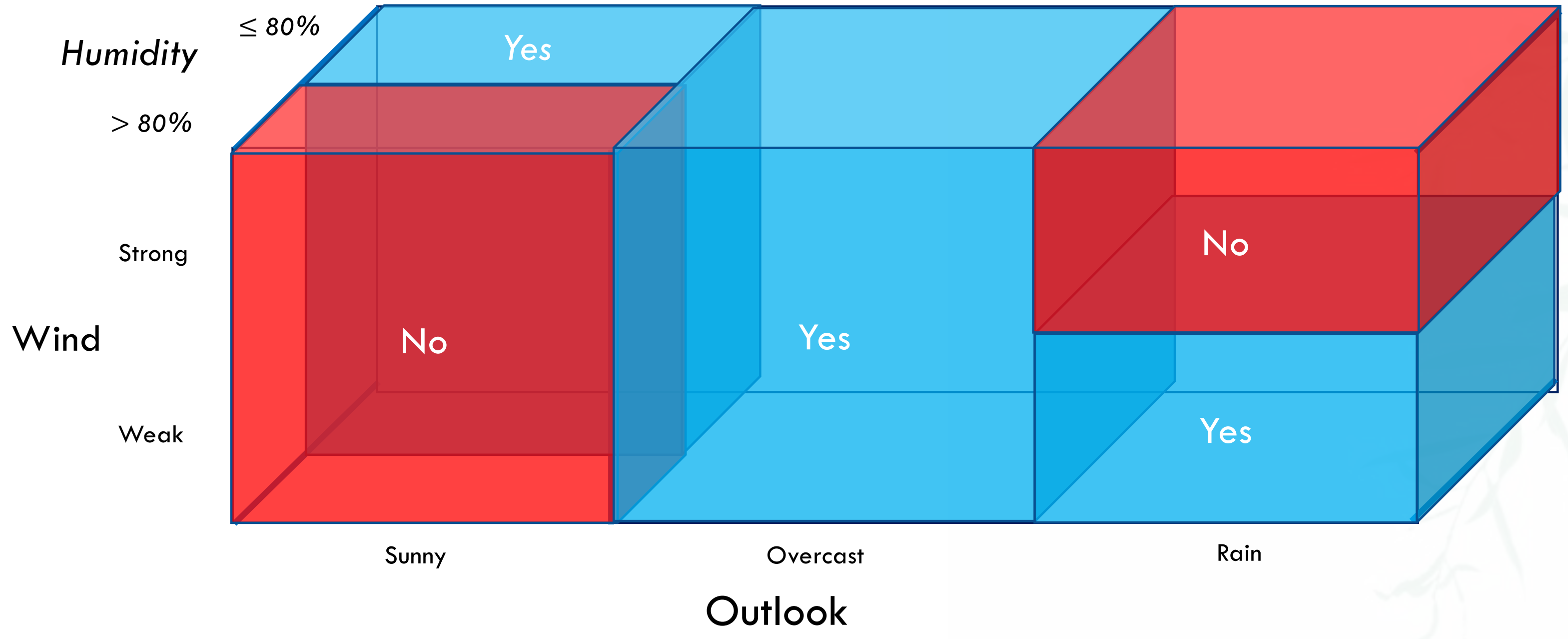
Day	Outlook	Temperature	Humidity	Wind	Play Tennis?
x	Sunny	Hot	72%	Weak	?

Inductive Bias – Decision boundary



Axis perpendicular (recursive) decision boundaries

Inductive Bias – Decision boundary



Axis perpendicular (recursive) decision boundaries

```
function DTL ($examples$, $attributes$, $default$) returns a decision tree

    if $examples$ is empty then return $default$
    else if all $examples$ have the same classification then return the classification
    else if $attributes$ is empty then return MODE($examples$)
    else
        $best \leftarrow$ CHOOSE-ATTRIBUTE($attributes$, $examples$)
        $tree \leftarrow$ a new decision tree root test $best$
        for each value $v_i$ of $best$ do
            $examples_i \leftarrow$ {elements of $examples$ with $best = v_i$}
            $subtree \leftarrow$ DTL($examples_i$, $attributes -\ best$, MODE($examples$))
            add a branch to $tree$ with label $v_i$, and subtree $subtree$
        return $tree$
```

3 Important Questions

Prefer important decisions at the top

1. How do we pick a feature to split on?
2. How do we discretize continuous features?

Prune back complete trees*

3. How do we decide where to prune?

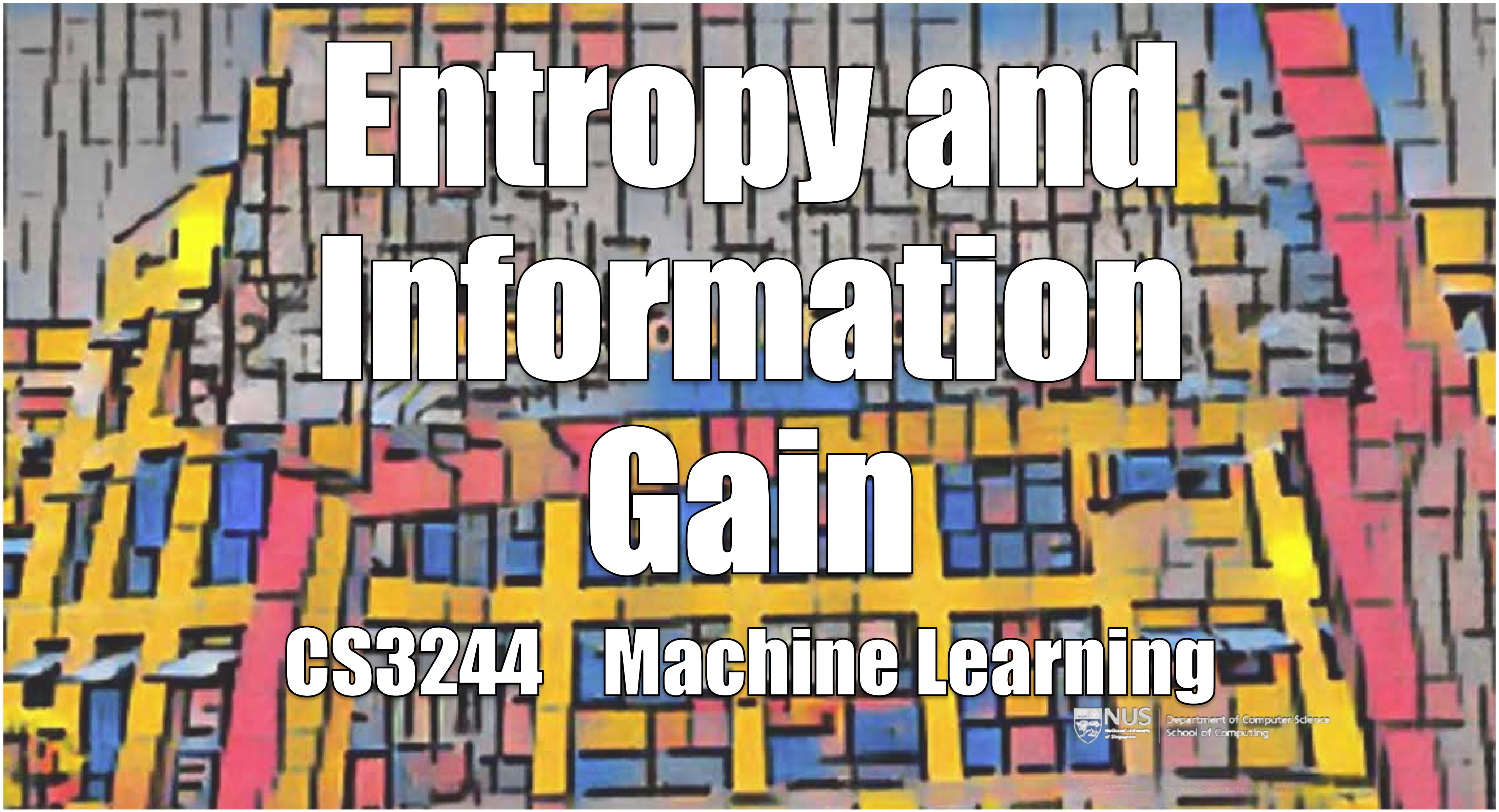
* we could decide when to stop growing the tree, but in practice this doesn't work well.

Decisions, decisions, decisions…

In physical subgroups or Zoom breakout rooms, do:

(5 mins): Discuss any **one** of your three decisions as a team. Answer: *What criteria do you use to decide which feature of your decision is the most important one to test first?*

Ask one member to write your collective answer to the #lectures thread. Upvote others that you like.



Entropy and Information Gain

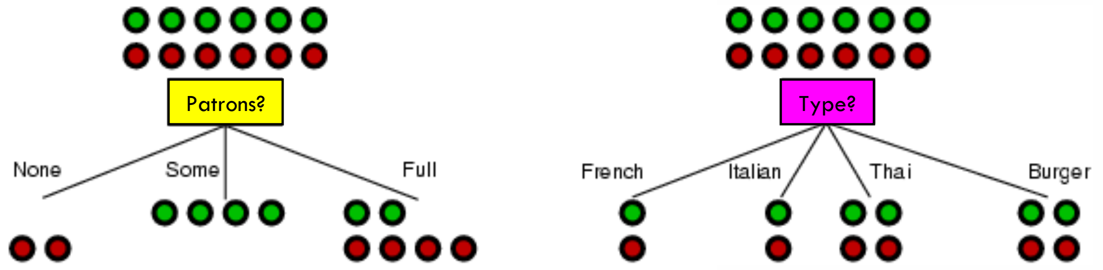
CS3244 Machine Learning

NUS National University of Singapore | Department of Computer Science School of Computing

Choosing an attribute

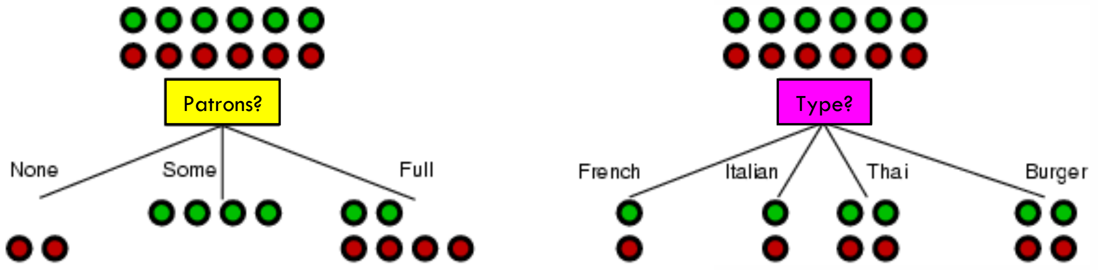
Idea: a good attribute (feature) splits the training set into subsets that are (ideally) "all positive" or "all negative".

Let's look at a case of *What did you eat for lunch?*



Choosing an attribute

Idea: a good attribute (feature) splits the training set into subsets that are (ideally) "all positive" or "all negative".



Patrons? is a better choice. Why?

Purity of the subproblems.

Q1. How do we pick a feature to split on?

To implement **Choose_Attribute** in DTL for enumerated feature sets with C possible values, we first need a concept of purity. We'll use Entropy:

$$H(X) = -\sum_{i=0}^{C} P_i \log_2(P_i)$$

⚠ $P \equiv$ *probability*, $p \equiv$ *# of positive examples.*

Example: for a training set containing p positive examples and n negative examples:

$$H\left(\frac{p}{p+n}, \frac{n}{p+n}\right) = -\frac{p}{p+n} log_2(\frac{p}{p+n}) - \frac{n}{p+n} log_2(\frac{n}{p+n})$$

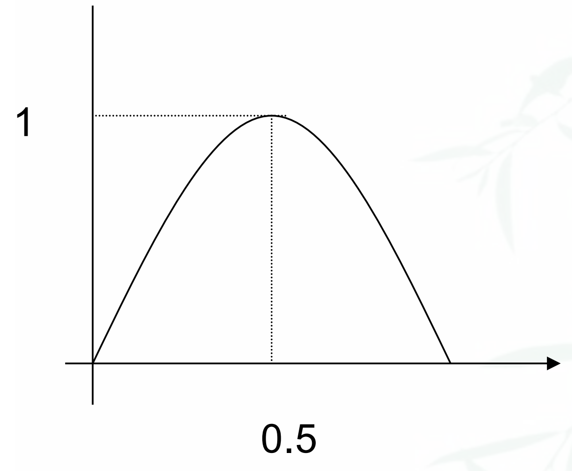
Entropy curve

For $\frac{p}{p+n}$, the 2-class entropy is:

0 when $\frac{p}{p+n}$ is 0

1 when $\frac{p}{p+n}$ is 0.5

0 when $\frac{p}{p+n}$ is 1



– monotonically increasing between 0 and 0.5

– monotonically decreasing between 0.5 and 1

Information gain

A chosen feature x_i divides the example set S into subsets $S_1, S_2, \ldots, S_c$ according to the C_i distinct values for x_i.

The entropy then reduces to the entropy of the subsets $S_1, S_2, \ldots, S_c$:

$$\text{remainder}(S, x_i) = \sum_{j=1}^{C_i} \frac{|S_j|}{|S|} H(S_j)$$

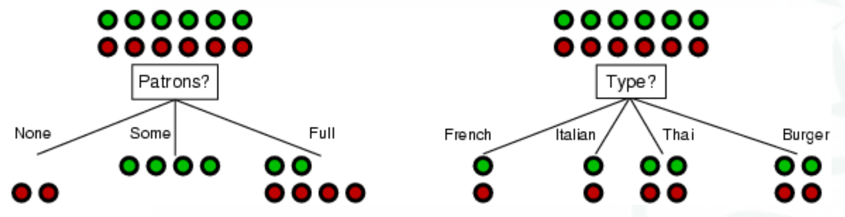
Information Gain (IG; "reduction in entropy") from knowing the value of x_i. Choose the attribute with the largest IG:

$$\text{IG}(S, x_i) = H(S) - \text{remainder}(S, x_i)$$

Information gain

For the training set at the root,

$p = n = 6, H\left(\frac{6}{12}, \frac{6}{12}\right) = 1$ bit.

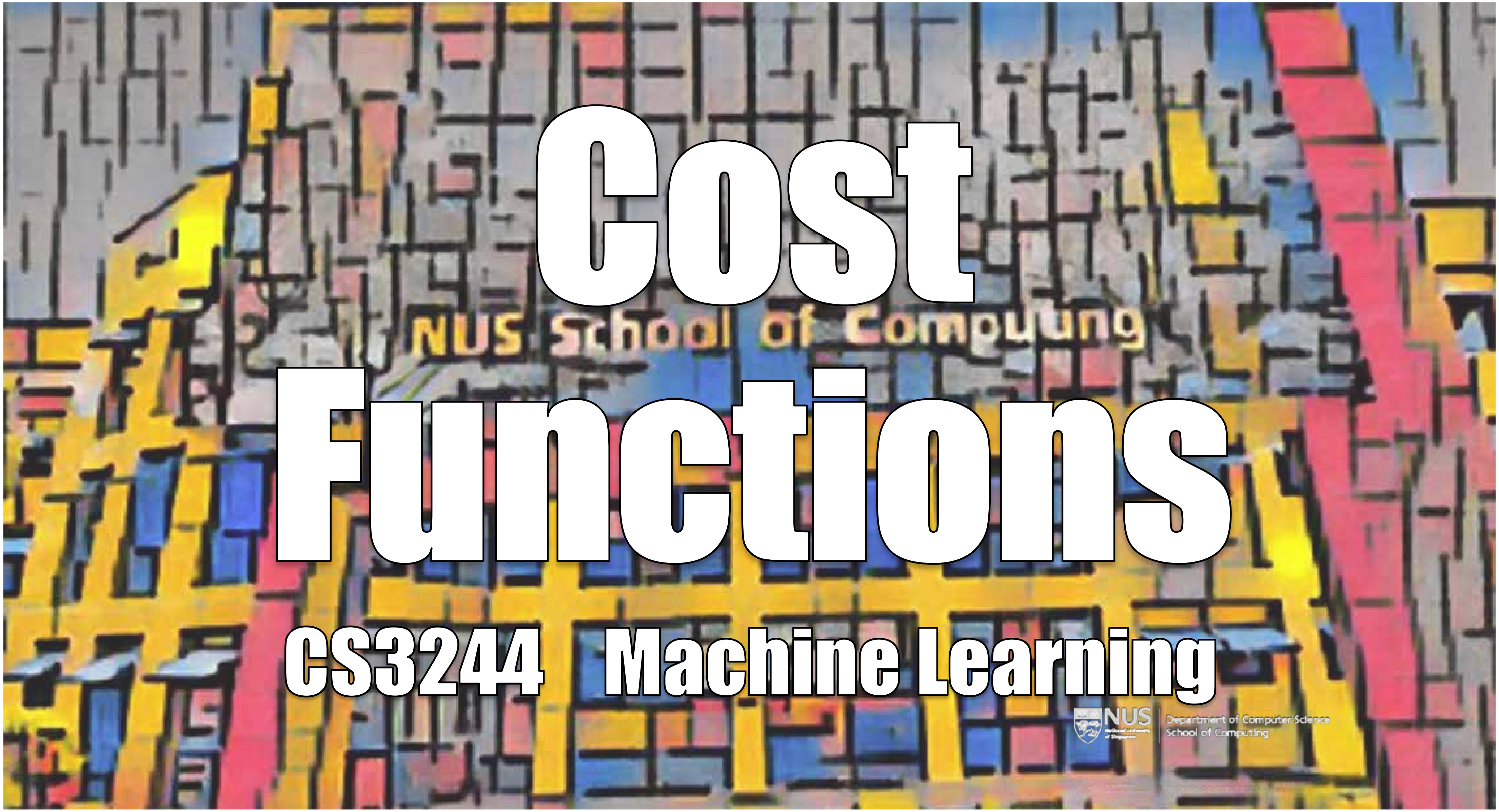


Consider the attributes *Patrons* and *Type*:

$$IG(Patrons) = 1 - \left[\frac{2}{12}H(0,1) + \frac{4}{12}H(1,0) + \frac{6}{12}H\left(\frac{2}{6}, \frac{4}{6}\right)\right] = 0.541 \text{ bits}$$

$$IG(Type) = 1 - \left[\frac{2}{12}H\left(\frac{1}{2}, \frac{1}{2}\right) + \frac{2}{12}H\left(\frac{1}{2}, \frac{1}{2}\right) + \frac{4}{12}H\left(\frac{2}{4}, \frac{2}{4}\right) + \frac{4}{12}H\left(\frac{2}{4}, \frac{2}{4}\right)\right] = 0 \text{ bits}$$

Patrons has the highest IG, and so is chosen by DTL as the root.



Cost Functions

CS3244 Machine Learning

NUS National University of Singapore | Department of Computer Science School of Computing

Weighty Decisions

(3 mins): Let’s do this for those of you attached virtually on Zoom! 💪🏽💪🏻. Go ahead and annotate the screen (be polite!) I’ll also prompt you all orally in class (Don’t be shy!)

Are there particular instances of your scenario that are more important? Describe them and how much they are worth.

Cost functions

What does $h_\theta \approx f$ mean?

Need a cost function $L(h_\theta, f)$

"Cost" is same "loss". Also, you'll see $J(h_\theta, f)$ instead of $L(h_\theta, f)$

This is almost always a pointwise definition: $l(h_\theta(x), f(x))$

Simple examples:

Squared error	$(h_\theta(x) - f(x))^2$
Binary error	$[h_\theta(x) \neq f(x)]$

Iverson bracket

$$[x] \equiv \begin{cases} 1, if\ x\ is\ true \\ 0, if\ x\ is\ false \end{cases}$$

From pointwise to overall

Overall cost $L(h, f) =$

average of pointwise cost $l(h(x), f(x))$

Training cost:

$$L_{train} = \frac{1}{m}\sum_{j=1}^{m} l\left(h_\theta\left(x^{(j)}\right), f\left(x^{(j)}\right)\right)$$

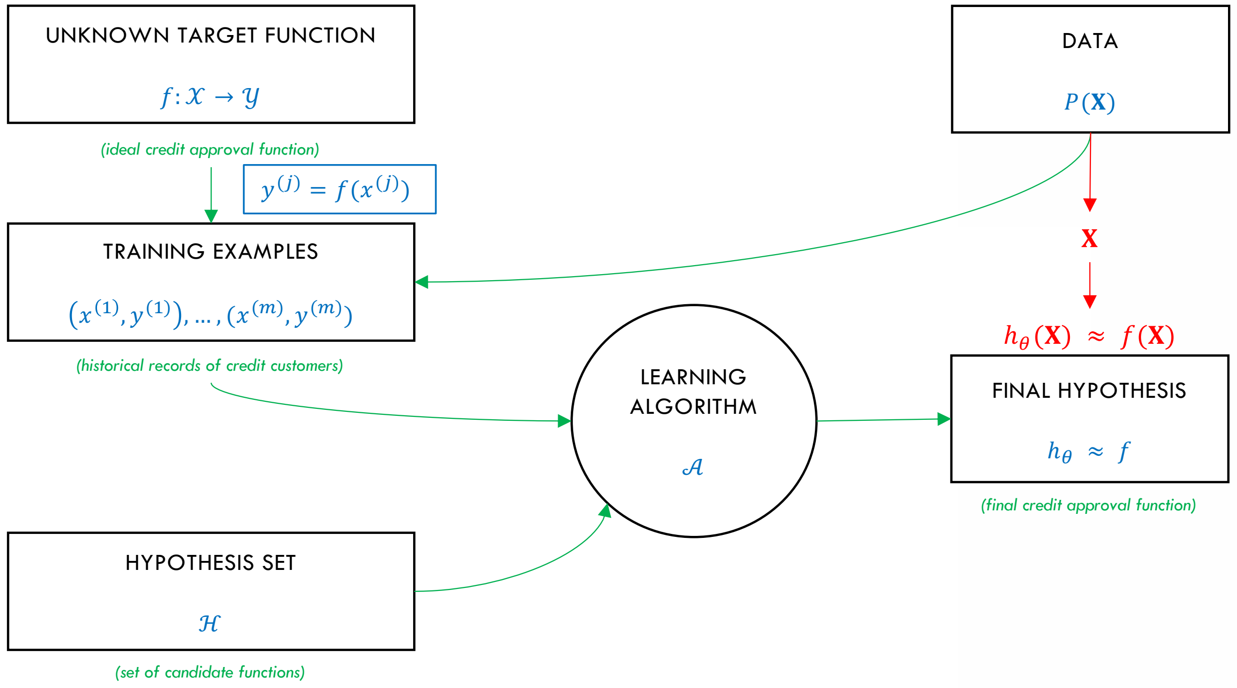
Why not a sum instead of an average?

Test cost:

$$L_{test} = \mathbb{E}_x[l(h_\theta(x), f(x))]$$

⚠ Expectations also use square brackets; doesn't mean T/F as in previous slide's Iverson notation.

The learning diagram with testing data, pointwise error



UNKNOWN TARGET FUNCTION
$f: \mathcal{X} \rightarrow \mathcal{Y}$
(ideal credit approval function)
$y^{(j)} = f(x^{(j)})$
DATA
$P(\mathbf{X})$
$\mathbf{X}$
$h_\theta(\mathbf{X}) \approx f(\mathbf{X})$
TRAINING EXAMPLES
$(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})$
(historical records of credit customers)
LEARNING ALGORITHM
$\mathcal{A}$
FINAL HYPOTHESIS
$h_\theta \approx f$
(final credit approval function)
HYPOTHESIS SET
$\mathcal{H}$
(set of candidate functions)

Choosing your cost function

Are you sick?

Two types of costs:

false positive (accept) or

false negative (reject)

How should we penalize for each type?

$$f \longrightarrow \begin{cases} +1 & \text{sick} \\ -1 & \text{well} \end{cases}$$

f

		+1	−1
h_θ	+1	True Positive	False Positive
	−1	False Negative	True Negative

During your last final exam before graduation

Are you sick?

False negative – get better on your own, or come back to the clinic later. At least you graduate on time.

False positive – Take the exam next year. Possibly pay tuition fees. $$$

$$f \longrightarrow \begin{cases} +1 & \text{sick} \\ -1 & \text{well} \end{cases}$$

f

h_θ

	+1	−1
+1	0 True Positive	100 False Positive
−1	1 False Negative	0 True Negative

Pre-vaccine COVID-19

Are you sick?

$$f \longrightarrow \begin{cases} +1 & \text{sick} \\ -1 & \text{well} \end{cases}$$

False negative highly costly!
People and the economy dies.

False positive requires the inconvenience of quarantine.

h_θ \ f	+1	−1
+1	0 True Positive	1 False Positive
−1	10,000 False Negative	0 True Negative

Your measure matters

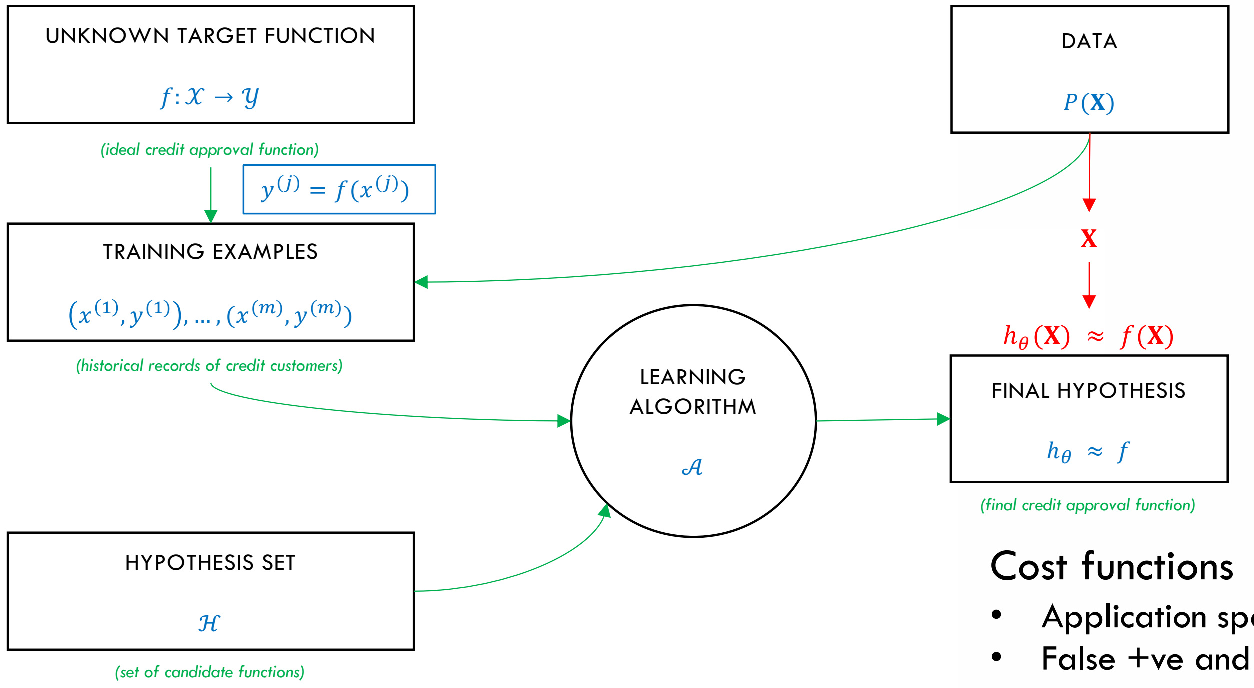
Where possible, we should use cost functions that fit the task, specified by the user.

However, this isn't always possible. Then use:

Plausible measures:	squared error ≡ Gaussian noise
Convenient measures:	closed form solution, convex optimization

The sweet spot: Plausible AND Convenient

Summary – the learning diagram with testing data, pointwise error


UNKNOWN TARGET FUNCTION
$f: \mathcal{X} \to \mathcal{Y}$
(ideal credit approval function)
$y^{(j)} = f(x^{(j)})$
DATA
$P(\mathbf{X})$
$\mathbf{X}$
$h_\theta(\mathbf{X}) \approx f(\mathbf{X})$
TRAINING EXAMPLES
$(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})$
(historical records of credit customers)
LEARNING ALGORITHM
$\mathcal{A}$
FINAL HYPOTHESIS
$h_\theta \approx f$
(final credit approval function)
HYPOTHESIS SET
$\mathcal{H}$
(set of candidate functions)

Quick Question: where does the cost function go?

Cost functions

- Application specific, user should specify
- False +ve and –ve may differ in cost

How you measure matters

(5 mins): Go through the three decisions from your pre-lecture activity as a team. Build on your examples earlier and answer: *What do you think the appropriate false positive and negative costs should be for each decision?*

Choose one random answer and write your collective answer to the #lectures thread. Upvote others that you like. 👍

Learning Theory 101: No Free Lunch

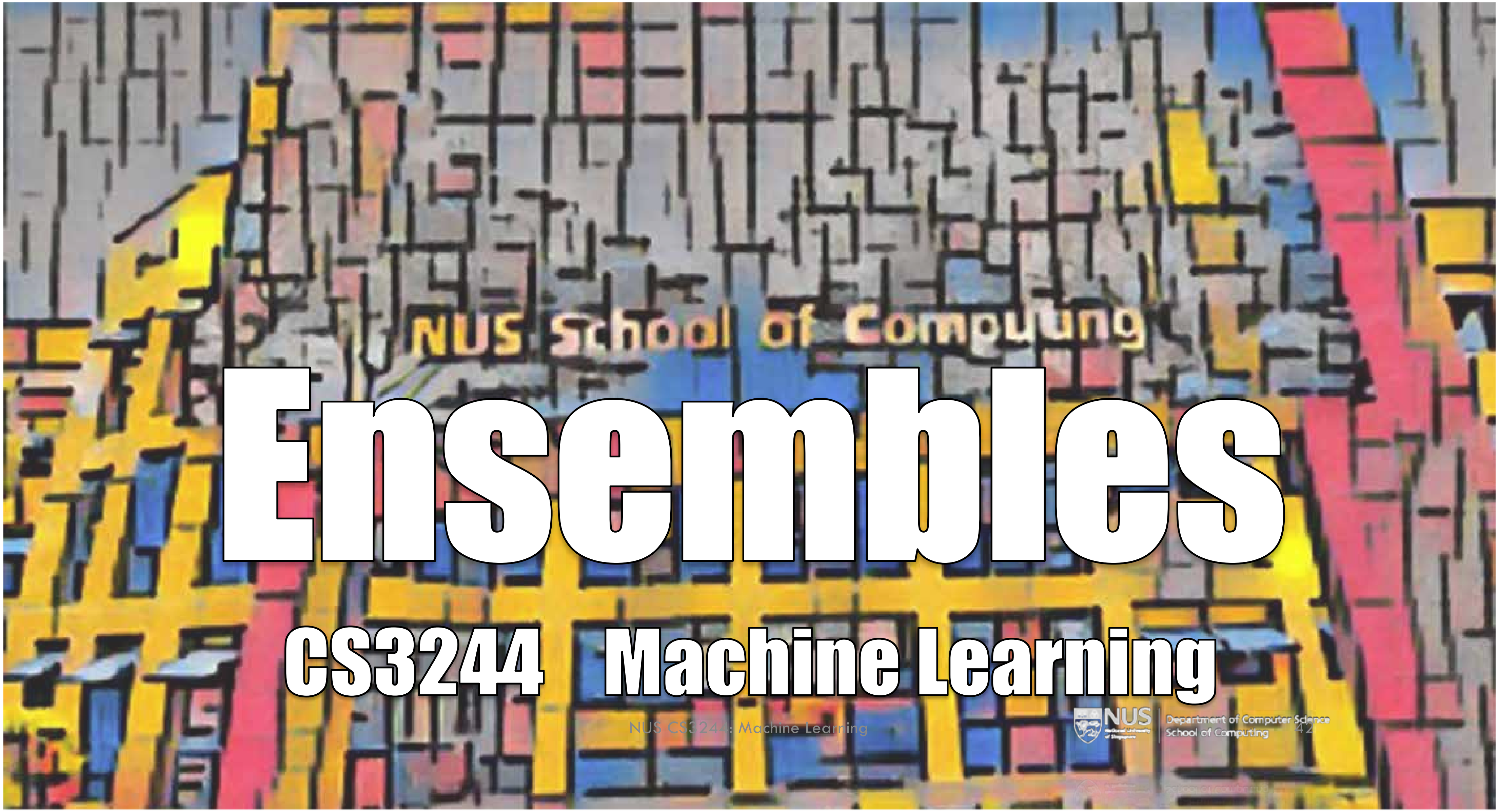
Machine learning works only because we apply some *inductive bias (modelled by $\mathcal{A}$)* to conclude that, once we've seen a particular pattern multiple times, we'll see that pattern again in unseen test data (induction).

But just because we saw something happen many times before, doesn't mean it will happen again.

Any outcome is possible, so no learning algorithm is better than any other given a training set and no other knowledge about the problem.

This is the essence of the No Free Lunch Theorem.

David Hume, image in public domain



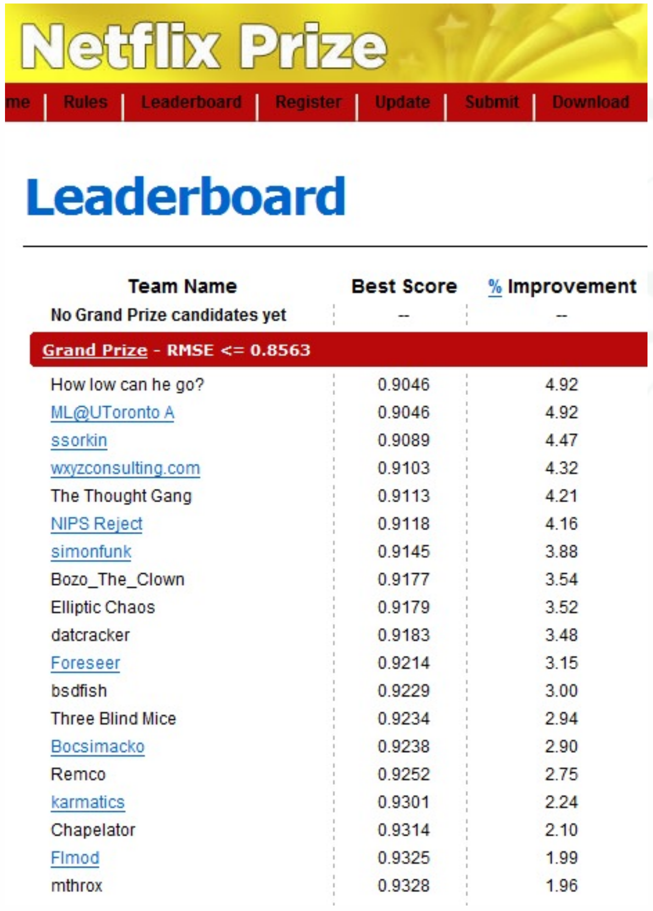
Ensembles

CS3244 Machine Learning

Ensembles: Revisiting the Netflix Prize

Just three weeks after it began, at least 40 teams had bested the Netflix classifier.

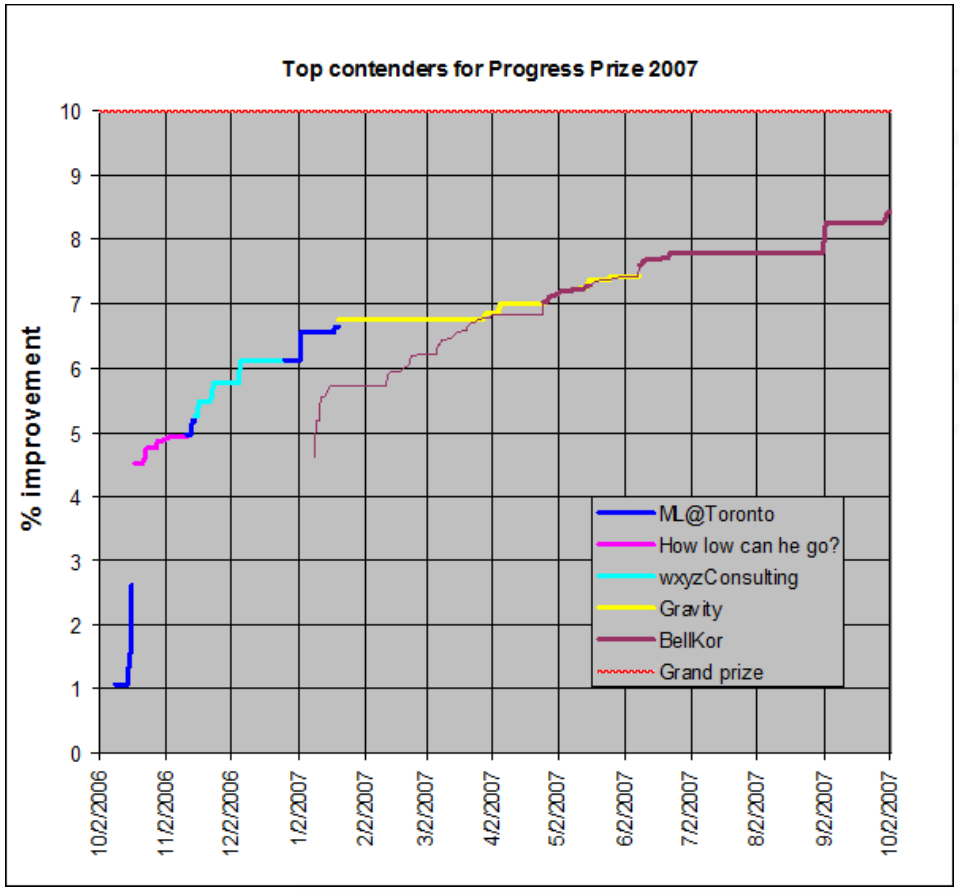
Top teams showed about 5% improvement.


Netflix Prize
me | Rules | Leaderboard | Register | Update | Submit | Download

Leaderboard

Team Name	Best Score	% Improvement
No Grand Prize candidates yet	--	--
Grand Prize - RMSE <= 0.8563		
How low can he go?	0.9046	4.92
ML@UToronto A	0.9046	4.92
ssorkin	0.9089	4.47
wxyzconsulting.com	0.9103	4.32
The Thought Gang	0.9113	4.21
NIPS Reject	0.9118	4.16
simonfunk	0.9145	3.88
Bozo_The_Clown	0.9177	3.54
Elliptic Chaos	0.9179	3.52
datcracker	0.9183	3.48
Foreseer	0.9214	3.15
bsdfish	0.9229	3.00
Three Blind Mice	0.9234	2.94
Bocsimacko	0.9238	2.90
Remco	0.9252	2.75
karmatics	0.9301	2.24
Chapelator	0.9314	2.10
Flmod	0.9325	1.99
mthrox	0.9328	1.96

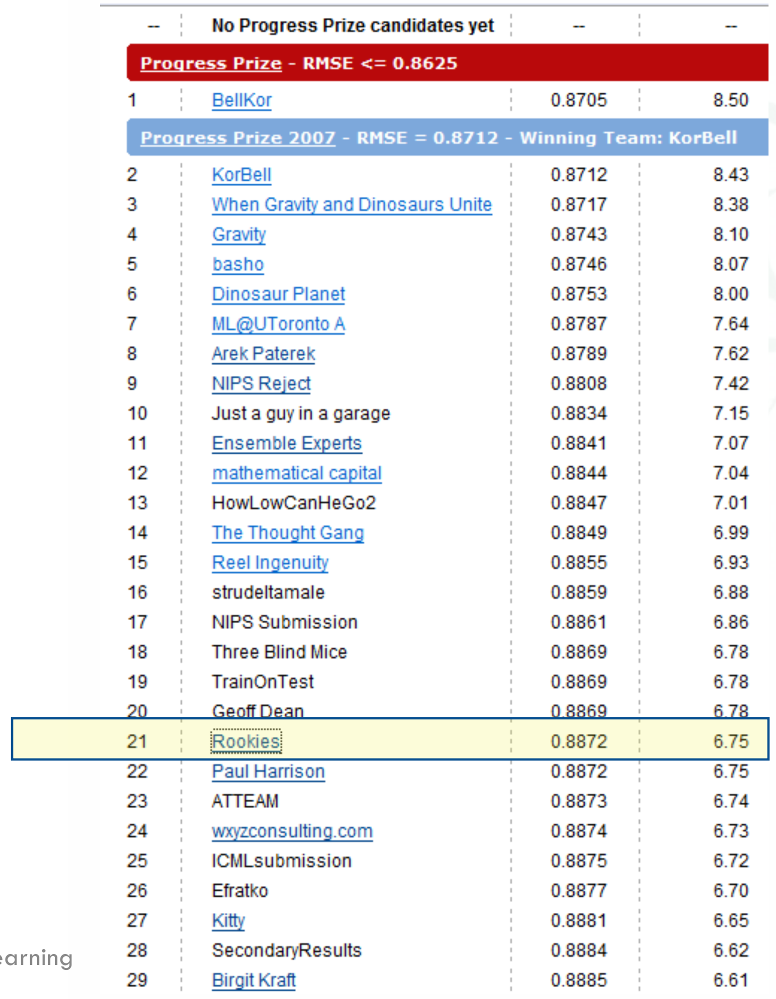
However, improvement slowed…



Source: http://www.research.att.com/~volinsky/netflix/

Rookies

“Thanks to Paul Harrison's collaboration, a simple mix of our solutions improved our result from 6.31 to 6.75”

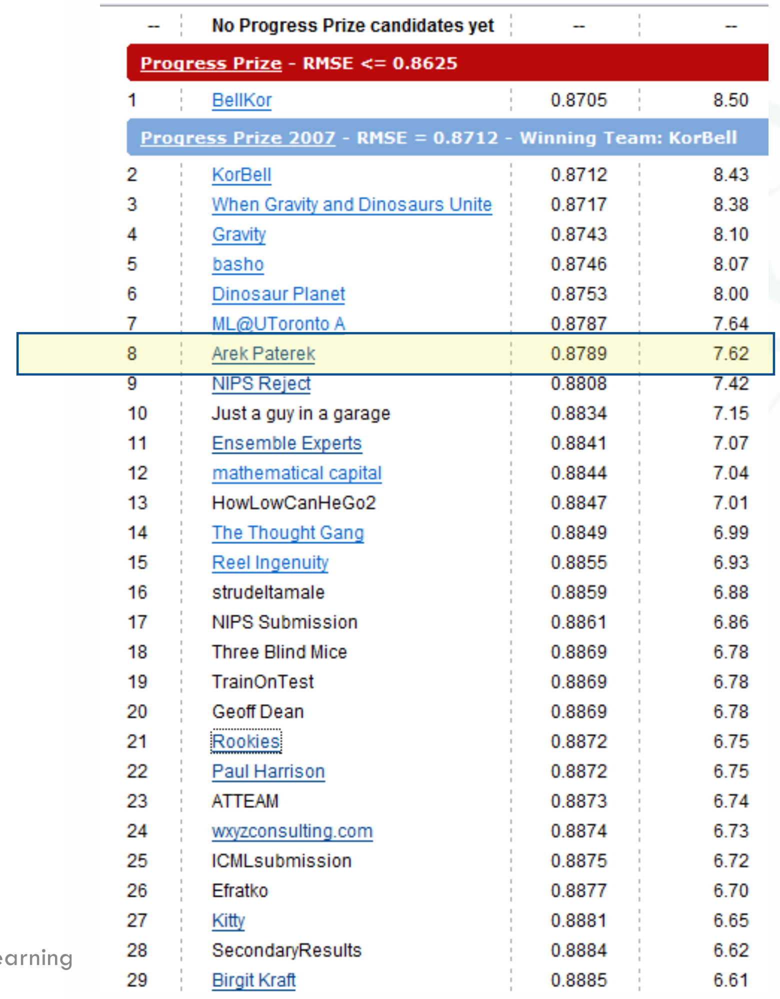


--	No Progress Prize candidates yet	--	--
Progress Prize - RMSE <= 0.8625			
1	BellKor	0.8705	8.50
Progress Prize 2007 - RMSE = 0.8712 - Winning Team: KorBell			
2	KorBell	0.8712	8.43
3	When Gravity and Dinosaurs Unite	0.8717	8.38
4	Gravity	0.8743	8.10
5	basho	0.8746	8.07
6	Dinosaur Planet	0.8753	8.00
7	ML@UToronto A	0.8787	7.64
8	Arek Paterek	0.8789	7.62
9	NIPS Reject	0.8808	7.42
10	Just a guy in a garage	0.8834	7.15
11	Ensemble Experts	0.8841	7.07
12	mathematical capital	0.8844	7.04
13	HowLowCanHeGo2	0.8847	7.01
14	The Thought Gang	0.8849	6.99
15	Reel Ingenuity	0.8855	6.93
16	strudeltamale	0.8859	6.88
17	NIPS Submission	0.8861	6.86
18	Three Blind Mice	0.8869	6.78
19	TrainOnTest	0.8869	6.78
20	Geoff Dean	0.8869	6.78
21	Rookies	0.8872	6.75
22	Paul Harrison	0.8872	6.75
23	ATTEAM	0.8873	6.74
24	wxyzconsulting.com	0.8874	6.73
25	ICMLsubmission	0.8875	6.72
26	Efratko	0.8877	6.70
27	Kitty	0.8881	6.65
28	SecondaryResults	0.8884	6.62
29	Birgit Kraft	0.8885	6.61

Arek Paterek

"My approach is to **combine the results of many methods** (also two-way interactions between them) using linear regression on the test set. The best method in my ensemble is regularized SVD with biases, post processed with kernel ridge regression"

http://rainbow.mimuw.edu.pl/~ap/ap_kdd.pdf

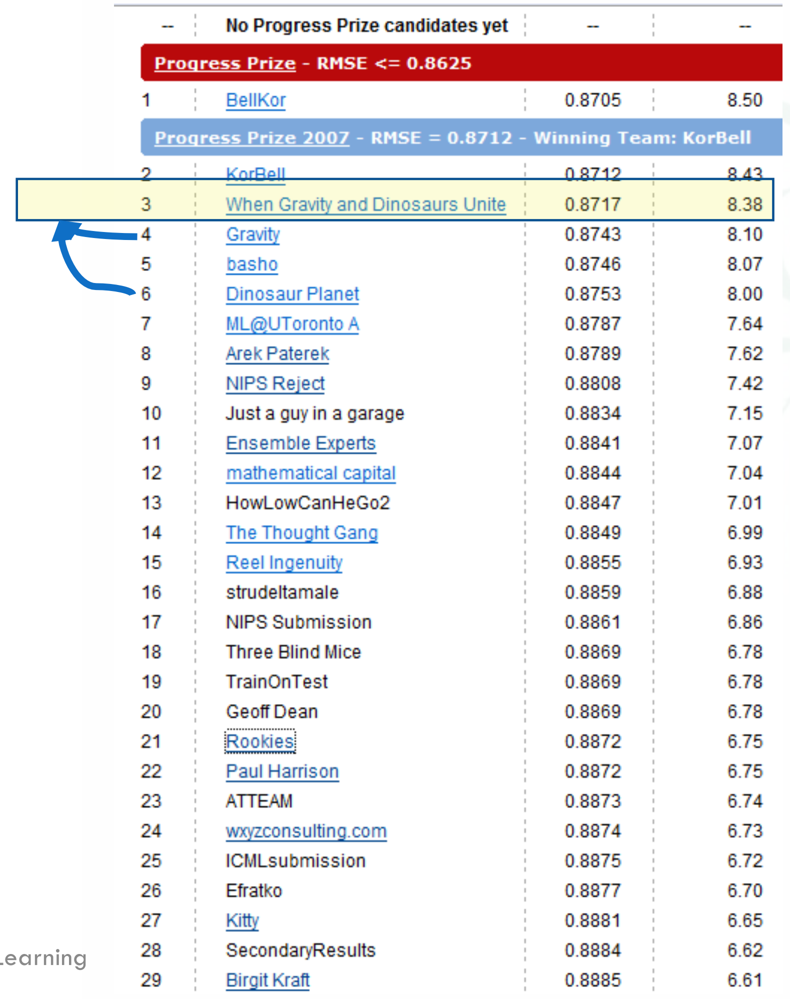


--	No Progress Prize candidates yet	--	--
Progress Prize - RMSE <= 0.8625			
1	BellKor	0.8705	8.50
Progress Prize 2007 - RMSE = 0.8712 - Winning Team: KorBell			
2	KorBell	0.8712	8.43
3	When Gravity and Dinosaurs Unite	0.8717	8.38
4	Gravity	0.8743	8.10
5	basho	0.8746	8.07
6	Dinosaur Planet	0.8753	8.00
7	ML@UToronto A	0.8787	7.64
8	Arek Paterek	0.8789	7.62
9	NIPS Reject	0.8808	7.42
10	Just a guy in a garage	0.8834	7.15
11	Ensemble Experts	0.8841	7.07
12	mathematical capital	0.8844	7.04
13	HowLowCanHeGo2	0.8847	7.01
14	The Thought Gang	0.8849	6.99
15	Reel Ingenuity	0.8855	6.93
16	strudeltamale	0.8859	6.88
17	NIPS Submission	0.8861	6.86
18	Three Blind Mice	0.8869	6.78
19	TrainOnTest	0.8869	6.78
20	Geoff Dean	0.8869	6.78
21	Rookies	0.8872	6.75
22	Paul Harrison	0.8872	6.75
23	ATTEAM	0.8873	6.74
24	wxyzconsulting.com	0.8874	6.73
25	ICMLsubmission	0.8875	6.72
26	Efratko	0.8877	6.70
27	Kitty	0.8881	6.65
28	SecondaryResults	0.8884	6.62
29	Birgit Kraft	0.8885	6.61

When Gravity and Dinosaurs Unite

"Our common team blends the result of team Gravity and team Dinosaur Planet."

Might have guessed from the name…

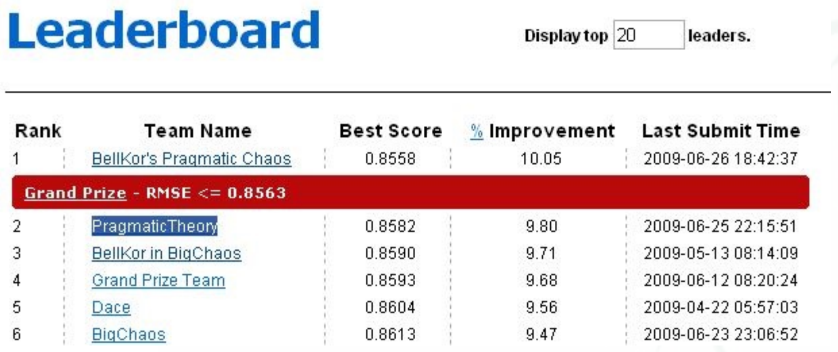


--	No Progress Prize candidates yet	--	--
Progress Prize - RMSE <= 0.8625			
1	BellKor	0.8705	8.50
Progress Prize 2007 - RMSE = 0.8712 - Winning Team: KorBell			
2	KorBell	0.8712	8.43
3	When Gravity and Dinosaurs Unite	0.8717	8.38
4	Gravity	0.8743	8.10
5	basho	0.8746	8.07
6	Dinosaur Planet	0.8753	8.00
7	ML@UToronto A	0.8787	7.64
8	Arek Paterek	0.8789	7.62
9	NIPS Reject	0.8808	7.42
10	Just a guy in a garage	0.8834	7.15
11	Ensemble Experts	0.8841	7.07
12	mathematical capital	0.8844	7.04
13	HowLowCanHeGo2	0.8847	7.01
14	The Thought Gang	0.8849	6.99
15	Reel Ingenuity	0.8855	6.93
16	strudeltamale	0.8859	6.88
17	NIPS Submission	0.8861	6.86
18	Three Blind Mice	0.8869	6.78
19	TrainOnTest	0.8869	6.78
20	Geoff Dean	0.8869	6.78
21	Rookies	0.8872	6.75
22	Paul Harrison	0.8872	6.75
23	ATTEAM	0.8873	6.74
24	wxyzconsulting.com	0.8874	6.73
25	ICMLsubmission	0.8875	6.72
26	Efratko	0.8877	6.70
27	Kitty	0.8881	6.65
28	SecondaryResults	0.8884	6.62
29	Birgit Kraft	0.8885	6.61

Ensembles, for the win

The team BellKor's Pragmatic Chaos is a combined team of BellKor, Pragmatic Theory and BigChaos. BellKor consists of Robert Bell, Yehuda Koren and Chris Volinsky. The members of Pragmatic Theory are Martin Piotte and Martin Chabbert. Andreas Töscher and Michael Jahrer form the team BigChaos.

"…There were two main factors which improved the overall accuracy: The quality of the individual algorithms and the ensemble idea … Best blending results are achieved when the whole ensemble has the right tradeoff between diversity and accuracy. "



Leaderboard

Display top 20 leaders.

Rank	Team Name	Best Score	% Improvement	Last Submit Time
1	BellKor's Pragmatic Chaos	0.8558	10.05	2009-06-26 18:42:37
Grand Prize - RMSE <= 0.8563				
2	PragmaticTheory	0.8582	9.80	2009-06-25 22:15:51
3	BellKor in BigChaos	0.8590	9.71	2009-05-13 08:14:09
4	Grand Prize Team	0.8593	9.68	2009-06-12 08:20:24
5	Dace	0.8604	9.56	2009-04-22 05:57:03
6	BigChaos	0.8613	9.47	2009-06-23 23:06:52

http://www.netflixprize.com/assets/GrandPrize2009_BPC_BigChaos.pdf

Meta Learning: Ensembles

Feature Film, Rating Count at least 25,000, Sci-Fi (Sorted by IMDb Rating Descending)

1-50 of 797 titles. | Next »

View Mode: Compact | Detailed

Sort by: Popularity | A
| Release Date | Date

1. Inception (2010)

PG-13 | 148 min | Action, Adventure, Sci-Fi

★ 8.8 ☆ Rate this 74 Metascore

A thief who steals corporate secrets through the use of dream-sharing technology is given the inverse task of planting an idea into the mind of a C.E.O.

Director: Christopher Nolan | Stars: Leonardo DiCaprio, Joseph Gordon-Levitt, Elliot Page, Ken Watanabe

Votes: 2,156,755 | Gross: $292.58M

2. The Matrix (1999)

R | 136 min | Action, Sci-Fi

Photo by Wan San Yip on Unsplash

Photo by David Travis on Unsplash

Photo by CDC on Unsplash

Ensembles

We have T reports $h_1, h_2, \dots, h_T$ predicting whether a stock will go up as $h(x)$.

We can:

1. Select the most trustworthy of them based on their usual performance
   Training performance: $h(x) = h_{t^*}(x)$ with $t^* = argmin_{t \in \{1,2,\dots,T\}} L_{train}(h_t^-)$

2. Let each report have a vote
   Uniform Vote: $h(x) = \text{sign}\left(\sum_{t=1}^{T} 1 \cdot h_t(x)\right)$

3. Weight the reports non-uniformly
   Weighted Vote: $h(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t \cdot h_t(x)\right)$ where $\alpha_t \geq 0$.

4. Combine the predictions conditionally
   Given the outcome of a test, decide which learner to use.

Which is the decision tree?

Stock Market – Ensemble

We have T reports $h_1, h_2, \dots, h_T$ predicting whether a stock will go up as $h(x)$.

We can:

1. Select the most trustworthy of them based on their usual performance
   Training performance: $h(x) = h_{t^*}(x)$ with $t^* = argmin_{t \in \{1,2,\dots,T\}} L_{train}(h_t^-)$

2. Let each report have a vote
   Uniform Vote: $h(x) = \text{sign}\left(\sum_{t=1}^{T} 1 \cdot h_t(x)\right)$

3. Weight the reports non-uniformly
   Weighted Vote: $h(x) = \text{sign}\left(\sum_{t=1}^{T} \alpha_t \cdot h_t(x)\right)$ where $\alpha_t \geq 0$.

4. Combine the predictions conditionally

Key: component hypotheses are **diverse**

How is a decision tree an ensemble?

Conditional Ensemble: decide on a learner based on a test.

Divide and Conquer – make smaller problems (like Quicksort)

🤔 What type of learner do we use in a decision tree?

Answer: A *decision stump*

The Decision Stump

a.k.a. 1-level decision tree:

$$h_{s,i,\theta}(x) = s \cdot \operatorname{sign}(x_i - \theta)$$

Positive and negative rays on some feature, three parameters (feature i, threshold θ, direction s)
Visualization: axis-perpendicular planes in $\mathbb{R}^n$.
Efficient to optimize: $O(n \cdot \log m)$ time

Allows efficient minimization of L_{train}, but can be too weak to work by itself
(hence called a *weak* learner).

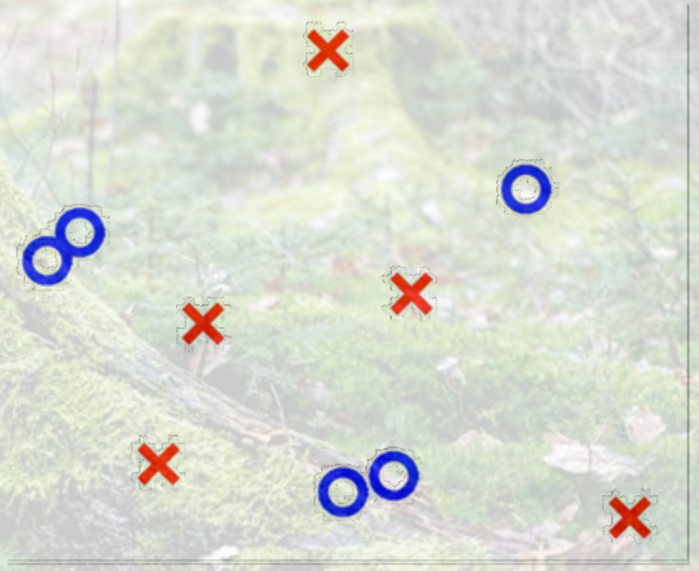


Photo by Yves Alarie on Unsplash

Building a tree

Conditionally combine each decision stump to build the tree.

Entropy / Information Gain used to make the subproblems smaller and simpler than the root problem.

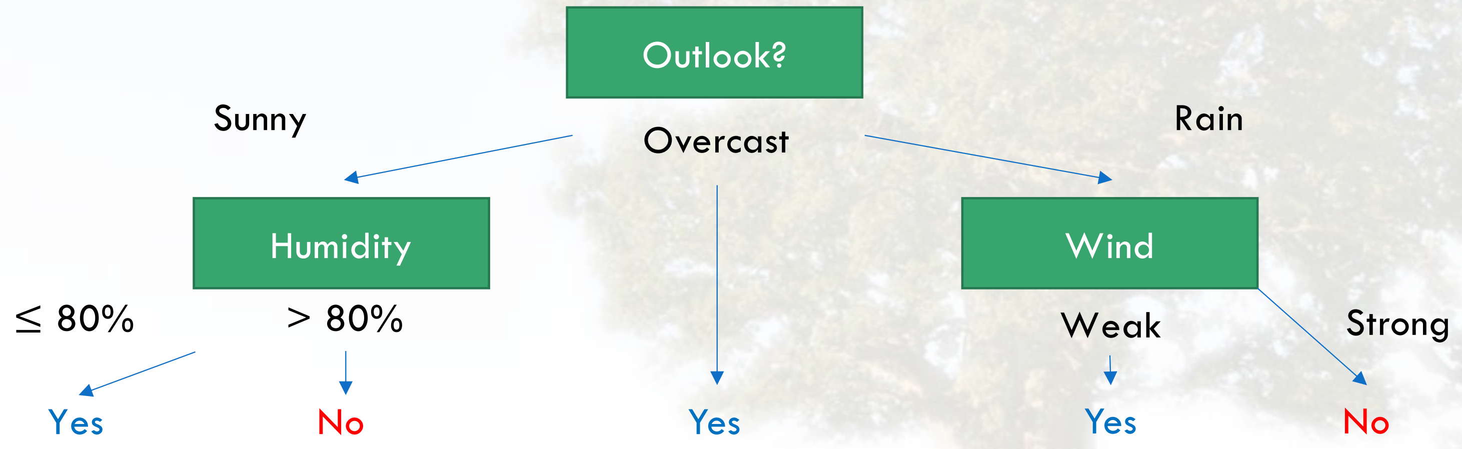
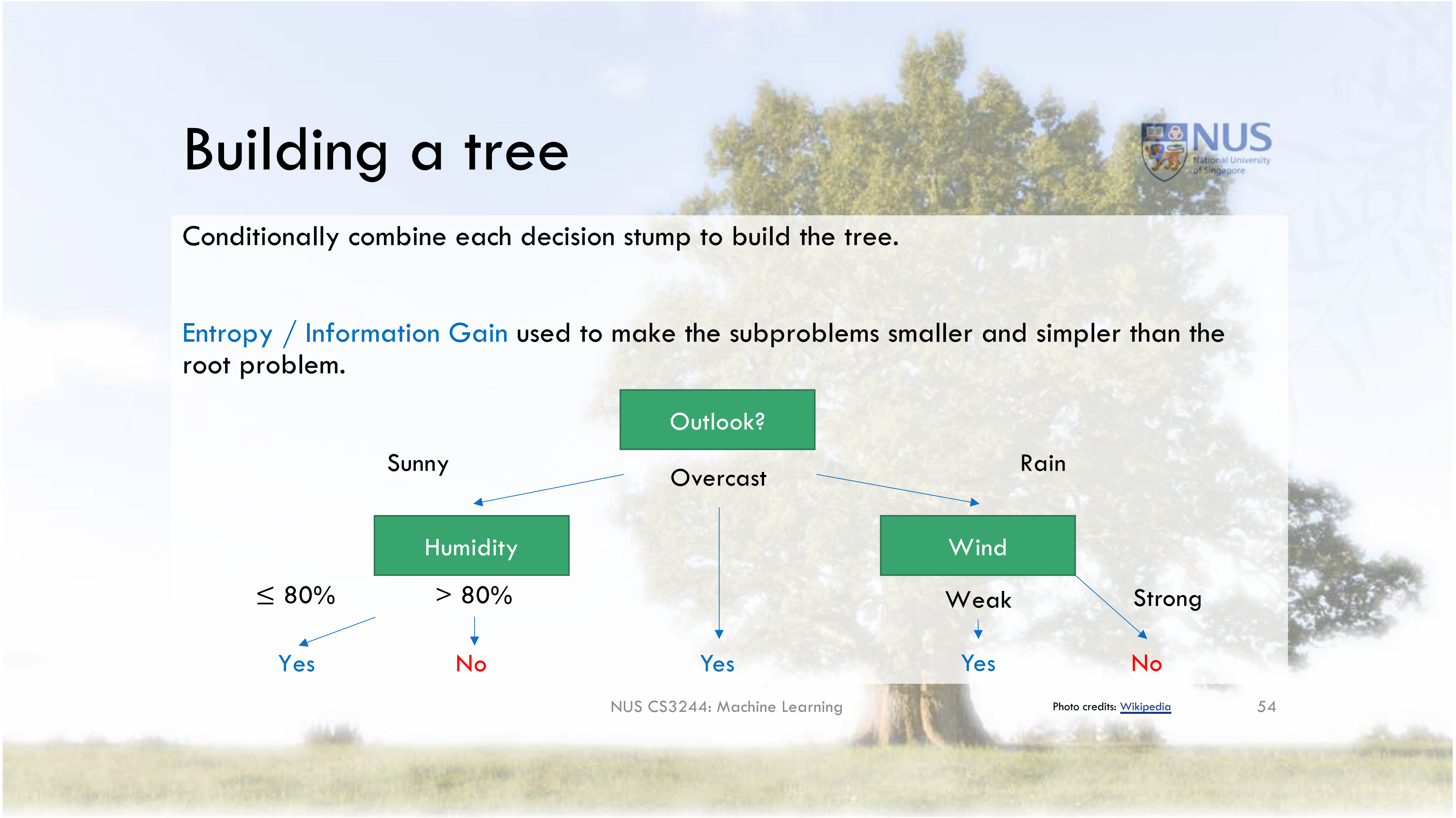
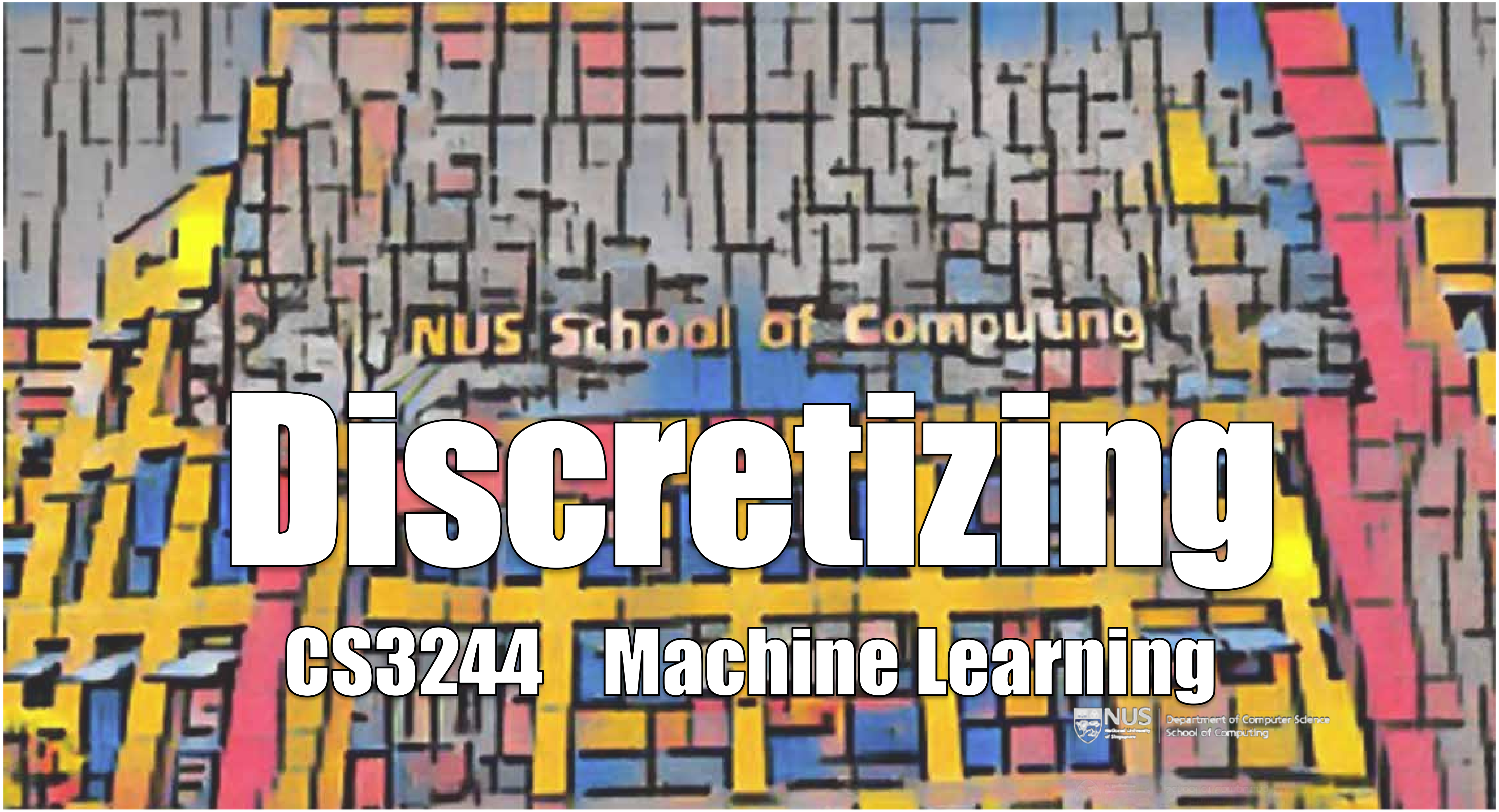


Photo credits: Wikipedia





Discretizing

CS3244 Machine Learning

NUS National University of Singapore | Department of Computer Science School of Computing

3 Important Questions

Prefer important decisions at the top

1. How do we pick a feature to split on? (*already covered*)
2. How do we discretize continuous features?

Prune back complete trees[*]

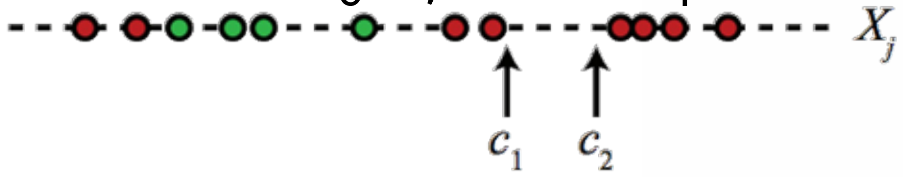
3. How do we decide where to prune?

[*] we could decide when to stop growing the tree, but in practice this doesn't work well.

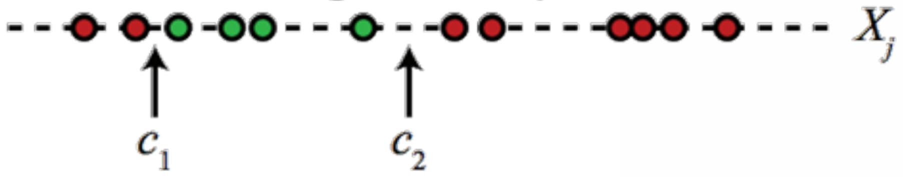
Q2. How do we discretize continuous features?

How can we choose among an infinite number of split points for a continuous feature x_j?

Observation 1. Moving split points between two observed values with the same label has no effect on information gain; use the midpoint.



Thus, 2. Only splitting between examples from different labels y could improve information gain.



Pruning

CS3244 Machine Learning

NUS National University of Singapore | Department of Computer Science School of Computing

Analyzing Decision Trees

How expressive is the $\mathcal{H}$ of decision trees?

They can express any row $(x_1, x_2, \dots, x_n, y)$ in the data set, trivially (how?).

You can always have $L_{train} = 0$ for a decision tree, if there is no noise in y.

🤔 What's the problem with an unpruned learned decision tree, in terms of learning theory?

Q3. How do we decide where to prune?

In Zoom breakout or physical subgroups, do:

(5 mins): Answer: *Why bother pruning? Why not build the decision tree and just use it?*

Ask one member to write your collective answer to the #lectures thread.
Upvote others that you like.

```
function DTL (examples, attributes, default) returns a decision tree

  if examples is empty then return default
  else if all examples have the same classification then return the classification
  else if attributes is empty then return MODE(examples)
  else
      best ← CHOOSE-ATTRIBUTE(attributes, examples)
      tree ← a new decision tree root test best
      for each value $v_i$ of best do
              $examples_i$ ← {elements of examples with best = $v_i$}
              subtree ← DTL($examples_i$, attributes − best, MODE(examples))
              add a branch to tree with label $v_i$, and subtree subtree
      return tree
```

The need for pruning

Day	Outlook	Temperature	Humidity	Wind	Tennis?
1	Sunny	Hot	95%	Weak	No
2	Sunny	Hot	92%	Strong	No
3	Overcast	Hot	95%	Weak	Yes
4	Rain	Mild	90%	Weak	Yes
5	Rain	Cool	60%	Weak	Yes
6	Rain	Cool	65%	Strong	No
7	Overcast	Cool	70%	Strong	Yes
8	Sunny	Cool	70%	Weak	Yes

Q3. How do we decide where to prune?

Answer:

1. Don't let leaves get too small. Require leaves have at least i training data.
2. Don't let tree grow too deep. Don't allow more than j levels of tests.
3. Use part of the training data to check performance (called validation)!

   Remove node and replace by the MODE of labels y if the pruned tree performs better than original.

Photo credits: pxfuel

Extending for Regression

(Regression ≡ Continuous y output aka predictor variable)

Both Decision Trees and k NN predict categorical features.

🤔 How can we extend these two algorithms to handle regression?

- Decision Trees: A test in a leaf predicts the MEAN of its training instances.
- k NN: a test instance predicts the MEAN of its k NN.

Many other options for extending to regression.
What could you propose?

Summary:
Analyzing Decision Tree Learning

The good

Interpretable: Intuitive for data exploration.

Efficient: both in training (greedy search) and testing

Discards irrelevant feature through use of Information Gain.

The bad

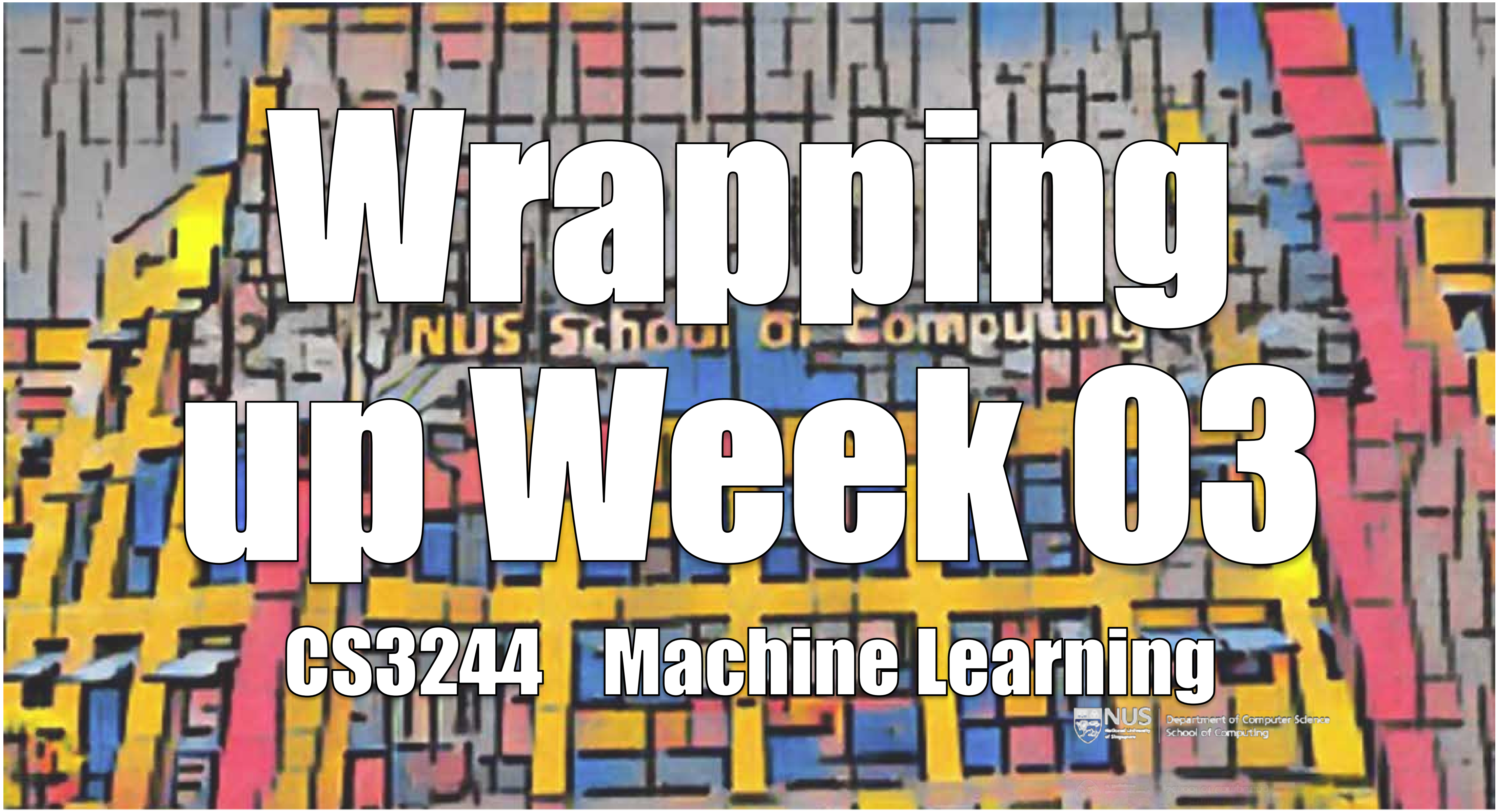
Instability: Susceptible to small fluctuations (high variance)

Hard decision boundaries: by default, no probabilistic interpretation of boundaries

Axis perpendicular decisions: doesn’t capture interaction between features (feature AND bug)

Wrapping up Week 03

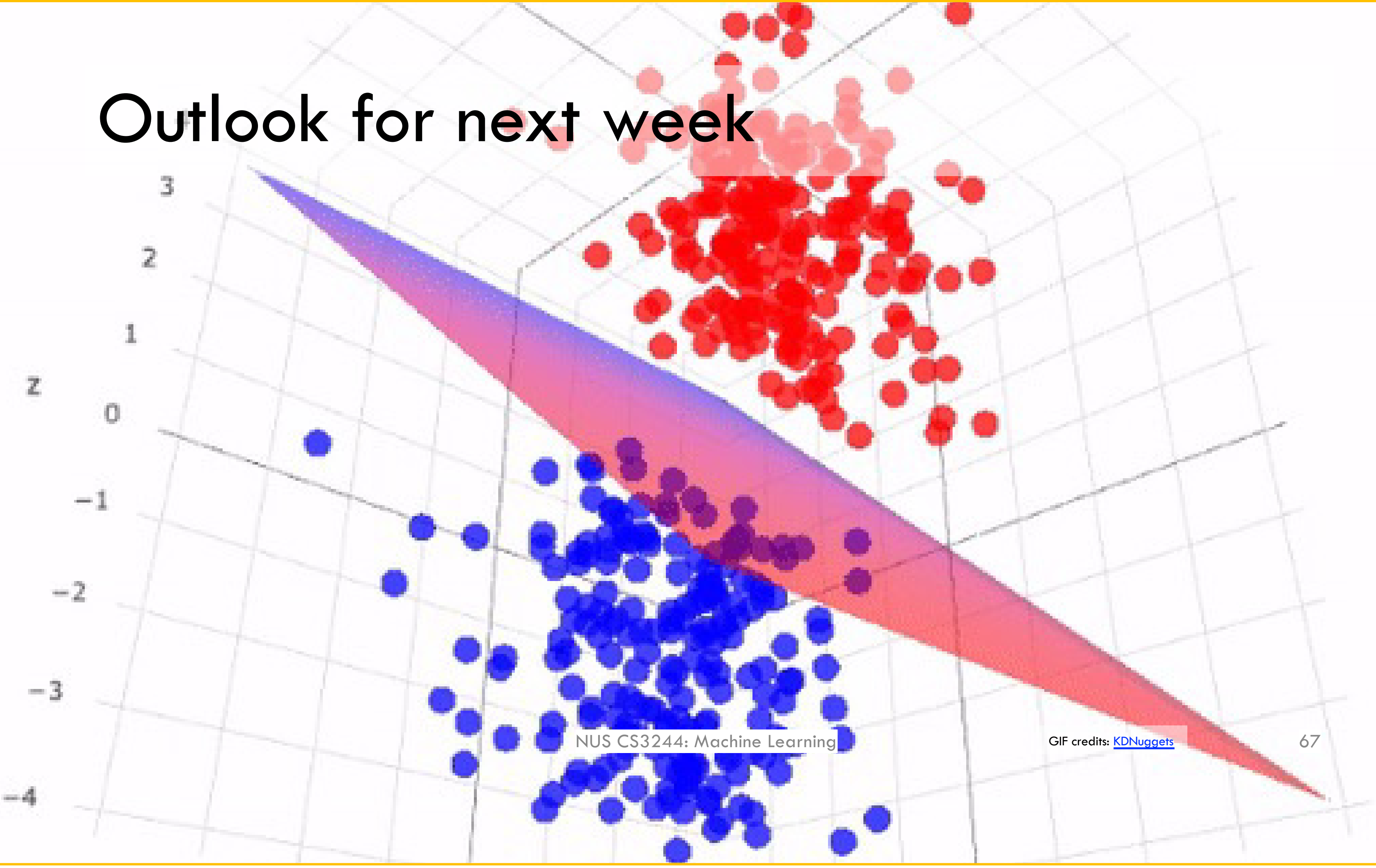
CS3244 Machine Learning


NUS School of Computing
NUS National University of Singapore
Department of Computer Science
School of Computing

What did we learn this week?

- Learning Outcomes for this week:
- Understand and implement Decision Trees;
- Master Information Gain as a means of choosing optimal features for decisions;
- Formalize cost functions as a means of evaluating a supervised learner's performance;
- Describe simple ensembling methods that integrate multiple classifiers;
  - Understand Decision Trees as an instance of an ensemble classifier of conditional decision stamps;
- Reason about inductive bias of the decision tree algorithm

Outlook for next week



GIF credits: KDNuggets

Assignment #1 released

Min, go over the k NN versus Decision Tree assignment:

http://www.comp.nus.edu.sg/~cs3244/AY22S1/01.assignment.html

Due this Sunday night. Start today!

Assigned Task (due before next Mon)

Read the Quora post Whats-the-difference-between-linear-and-non-linear-machine-learning-model, especially Robby's answer (5 mins) (*You may want to use guest / incognito mode, if asked to sign up*)

Post a 1–2 sentence answer to the question in tutorial group: #tg-xx

Can we model non-linearity with a linear classifier?

[Don't worry if you're not sure, we'll cover this again in Week 04.]