



SCAN ME

Features and classification

CSC401/2511 – Natural Language Computing – Spring 2023

Lecture 3 En-Shiun Annie Lee, Raeid Saqur, Zining Zhu

University of Toronto

Lecture 3 overview

- Today:
 - Quick introduction to Text Classification
 - **Feature extraction** from text.
 - How to pick the right features?
 - Grammatical ‘parts-of-speech’.
 - (even when nobody is speaking)
 - **Classification overview**
- Some slides *may* be based on content from Bob Carpenter, Dan Klein, Roger Levy, Josh Goodman, Dan Jurafsky, and Christopher Manning.



SCAN ME

https://docs.google.com/presentation/d/12TdTZ1qBRP6TONCEewBILzI4E4ce2-pNmzK5Ur5zze4/edit#slide=id.g13e1c0bee48_0_1838

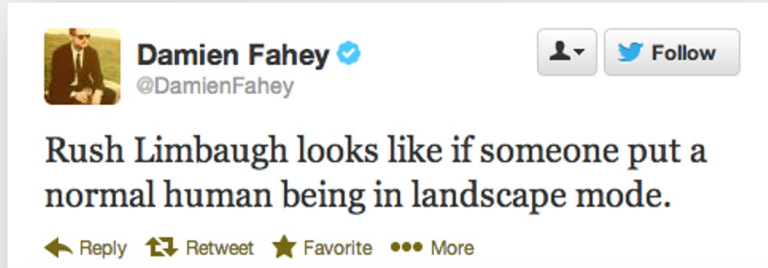
Quick Intro to Text Classification

Features

- **Feature**: *n.* A measurable **variable** that is (*or should be*) **distinctive** of something we want to model.
- We often choose features that are useful to **identify** something, i.e., to do **classification**.
 - E.g., an emotional, whiny **tone** is likely to indicate that the speaker is neither professional, nor scientific, nor political.
 - Note that in neural networks, e.g., '**features**' often refer to something distinctive but not usually *nameable*.
- We often need **various** features to adequately model something – *but not too many!*

Example: Feature vectors

- Values for several features of an **observation** can be put into a single **vector**.



# proper nouns	# 1 st person pronouns	# commas
2	0	0



5	0	0
---	---	---



0	1	1
---	---	---

Feature vectors

- Features should be useful in **discriminating** between categories.

Table 3: Features to be computed for each text

- Counts:
 - First person pronouns
 - Second person pronouns
 - Third person pronouns
 - Coordinating conjunctions
 - Past-tense verbs
 - Future-tense verbs
 - Commas
 - Colons and semi-colons
 - Dashes
 - Parentheses
 - Ellipses
 - Common nouns
 - Proper nouns
 - Adverbs
 - *wh*-words
 - Modern slang acroynms
 - Words all in upper case (at least 2 letters long)
- Average length of sentences (in tokens)
- Average length of tokens, excluding punctuation tokens (in characters)
- Number of sentences

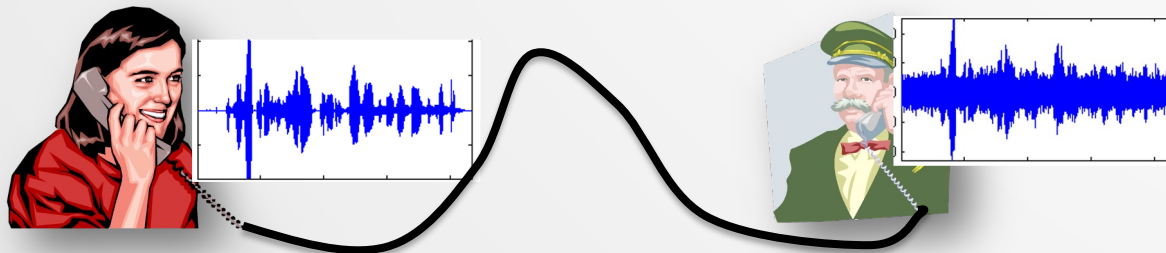
Higher values → this person is referring to themselves (to their opinion, too?)

Higher values → looking forward to (or dreading) some future event?

Lower values → this tweet is more formal. Perhaps not overly sentimental?

Quick comment on noise


- **Noise** is generally any **artifact** in your received ‘**signal**’ that **obfuscates** (hides) the features you want.
 - E.g., in **acoustics**, it can be a loud buzzing sound that washes out someone’s voice.
 - E.g., in **tweets**, it can be text that affects feature values.
 - E.g., The semi-colon in “... chill ;)” is part of an **emoticon**; will it confuse our **classifier** if we count it as punctuation?



Different features for different tasks

- **Alzheimer's disease** involves atrophy in the brain.
 - Excessive **pauses** (acoustic disfluencies),
 - Excessive **word type repetition**, and
 - Simplistic or **short** sentences.
 - '**function words**' like *the* and *an* are often **dropped**.
- To **diagnose** Alzheimer's disease, one might measure:
 - **Proportion** of utterance spent in **silence**.
 - **Entropy** of **word type** usage.
 - **Number** of word **tokens** in a sentence.
 - **Number** of prepositions and determiners (explained shortly).

Features in Sentiment Analysis

- **Sentiment analysis** can involve detecting:
 - **Stress or frustration** in a conversation.
 - **Interest, confusion, or preferences.** Useful to marketers.
 - e.g., *'got socks for xmas wanted #ps5 fml'* 
 - **Deceit.** e.g., *'Let's watch Netflix and chill.'*
- Complicating factors include **sarcasm, implicature**, and a **subtle** spectrum from **negative** to **positive** opinions.
- **Useful features** for sentiment analyzers include:
 - Trigrams.
 - First-person pronouns.

What does this mean?

Pronouns? Prepositions?
Determiners?

Pre-processing

- **Pre-processing** involves **preparing** your data to make feature extraction easier or more valid.
 - E.g., **punctuation** likes to press up against words. The sequence “ *example,* ” should be counted as **two** tokens – not one.
 - We separate the punctuation, as in “ *example ,* ”.



- **There is no perfect pre-processor.**

Mutually exclusive approaches can often **both** be justified.

- E.g., Is *Newfoundland-Labrador* **one** word type or **two**?
Each answer has a unique implication for splitting the dash.
- Often, **noise-reduction** removes *some* information.
- Being **consistent** is important.



SCAN ME

Parts of Speech

Parts-of-speech (PoS)

- Linguists like to group words according to their **structural function** in building sentences.
 - This is similar to grouping Lego by their shapes.
- **Part-of-speech**: *n* lexical category or morphological class.



Nouns collectively constitute a part-of-speech
(called *Noun*)

Example parts-of-speech

Part of Speech	Description	Examples
Noun	is usually a person, place, event, or entity.	<i>chair, pacing, monkey, breath.</i>
Verb	is usually an action or predicate.	<i>run, debate, explicate.</i>
Adjective	modifies a noun to further describe it.	<i>orange, obscene, disgusting.</i>
Adverb	modifies a verb to further describe it.	<i>lovingly, horrifyingly, often</i>

Example parts of speech

Part of Speech	Description	Examples
Preposition	Often specifies aspects of space, time, or means .	<i>around, over, under, after, before, with</i>
Pronoun	Substitutes for nouns; referent typically understood in context.	<i>I, we, they</i>
Determiner	logically quantify words, usually nouns.	<i>the, an, both, either</i>
Conjunction	combines words or phrases.	<i>and, or, although</i>

Contentful parts-of-speech

- Some PoS convey more **meaning**.
 - Usually nouns, verbs, adjectives, adverbs.
 - **Contentful** PoS usually contain more words.
 - e.g., there are more **nouns** than **prepositions**.
 - **New** contentful words are continually **added**
e.g., *an app, to google, to misunderestimate*.
 - **Archaic** contentful words go **extinct**.
e.g., *fumificate, v., (1721-1792),*
frenigerent, adj., (1656-1681),
melanochalographer, n., (c. 1697).

Functional parts-of-speech

- Some PoS are '**glue**' that holds others together.
 - E.g., prepositions, determiners, conjunctions.
 - **Functional** PoS usually cover a **small** and **fixed** number of word types (i.e., a '**closed class**').
- Their **semantics** depend on the contentful words with which they're used.
 - E.g., *I'm **on** time vs. I'm **on** a boat*

Grammatical features

- There are several **grammatical features** that can be associated with words:
 - **Case**
 - **Person**
 - **Number**
 - **Gender**
- These features can **restrict** other words in a sentence.

(Aside) Grammatical features – case

- **Case**: *n.* the grammatical form of a **noun** or **pronoun**.
- E.g.,
 - nominative**: the **subject** of a verb (e.g., “**We** remember”)
 - accusative**: the **direct object** of a verb (e.g., “You remember **us**”)
 - dative**: the **indirect object** of a verb (e.g. “I gave your **mom** the book”)
 - genitive**: indicates **possession** (e.g., “your **mom**’s book”)
 - ...

(Aside) Grammatical features – person

- **Person**: *n.* typically refers to a participant in an event, especially with **pronouns** in a conversation.

- E.g.,

first: The speaker/author. Can be either inclusive (“**we**”) or exclusive of hearer/reader (“**I**”).

second: The hearer/reader, exclusive of speaker (“**you**”).

third: Everyone else (“**they**”).

Aside: some languages add exclusivity to ‘person’.

E.g., in “**we**’re going to a party” – who’s going to the party?

(Aside) Grammatical features – number

- **Number:** *n.* Broad numerical distinction.
- E.g.,
 - singular:** Exactly one (“one *cow*”)
 - plural:** More than one (“two *cows*”)
 - dual:** Exactly two (e.g., *-ان* in Arabic).
 - paucal:** Not *too* many (e.g., in Hopi).
 - collective:** Countable (e.g., Welsh “*moch*” for ‘pigs’ as opposed to “*mochyn*” for vast ‘pigginess’).
 - ...

(Aside) Grammatical features – gender

- **gender**: *n.* typically partitions **nouns** into classes associated with biological gender. **Not** typical in English.
 - Gender alters neighbouring words **regardless** of speaker/hearer.
- E.g.,
 - feminine**: Typically **pleasant** things (not always).
(e.g., *la France*, *eine Brücke*, *une poubelle*).
 - masculine**: Typically **ugly** or **rugged** things (not always).
(e.g., *le Québec*, *un pont*).
 - neuter**: Everything else.

(*Brücke*: German bridge; *pont*: French bridge; *poubelle*: French garbage)

Other features of nouns

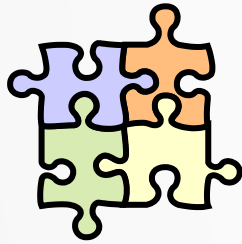
- **Proper noun:** **named** things (e.g., “they’ve killed **Bill!**”)
- **Common noun:** **unnamed** things
(e.g., “they’ve killed the **bill!**”)

- **Mass noun:** **divisible** and **uncountable**
(e.g., “**butter**” split in two gives two piles of butter – not two ‘*butters*’)
- **Count noun:** **indivisible** and **countable**.
(e.g., a “**pig**” split in two does not give two pigs)

(Aside) Some features of prepositions

- *By*
 - **Alongside:** *a cottage by the lake*
 - **Agentive:** *Chlamydia was given to Mary by John*
- *For*
 - **Benefactive:** *I have a message for your mom*
 - **Purpose:** *have a friend (over) for dinner*
- *With*
 - **Sociative:** *watch a film with a friend*
 - **Instrumental:** *hit a nail with a hammer*

Agreement



- Parts-of-speech **should** match (i.e., **agree**) in certain ways.
- **Articles** ‘have’ to **agree** with the **number** of their **noun**
 - e.g., “these pretzels are making me thirsty”
 - e.g., “a winters are coming”
- **Verbs** ‘have’ to **agree** (at least) with their **subject** (in English)
 - e.g., “the dogs eats the gravy” **no number agreement**
 - e.g., “Yesterday, all my trouble is em so far away”
bad tense – should be past tense *seemed*
 - e.g., “Can you handle me the way I are?”





SCAN ME

Tagging

PoS tagging

- **Tagging:** *v.g.* the process of **assigning a part-of-speech** to each word in a sequence.
- E.g., using the '**Penn treebank**' tag set (see appendix):

Word	The	nurse	put	the	sick	patient	to	sleep
Tag	DT	NN	VBD	DT	JJ	NN	IN	NN

Ambiguities in parts-of-speech

- Words can belong to many parts-of-speech.
 - E.g., *back*:
 - *The **back**/JJ door* (adjective)
 - *On its **back**/NN* (noun)
 - *Win the voters **back**/RB* (adverb)
 - *Promise to **back**/VB you in a fight* (verb)
- We want to decide the **appropriate** tag given a particular sequence of tokens.

Why is tagging useful?

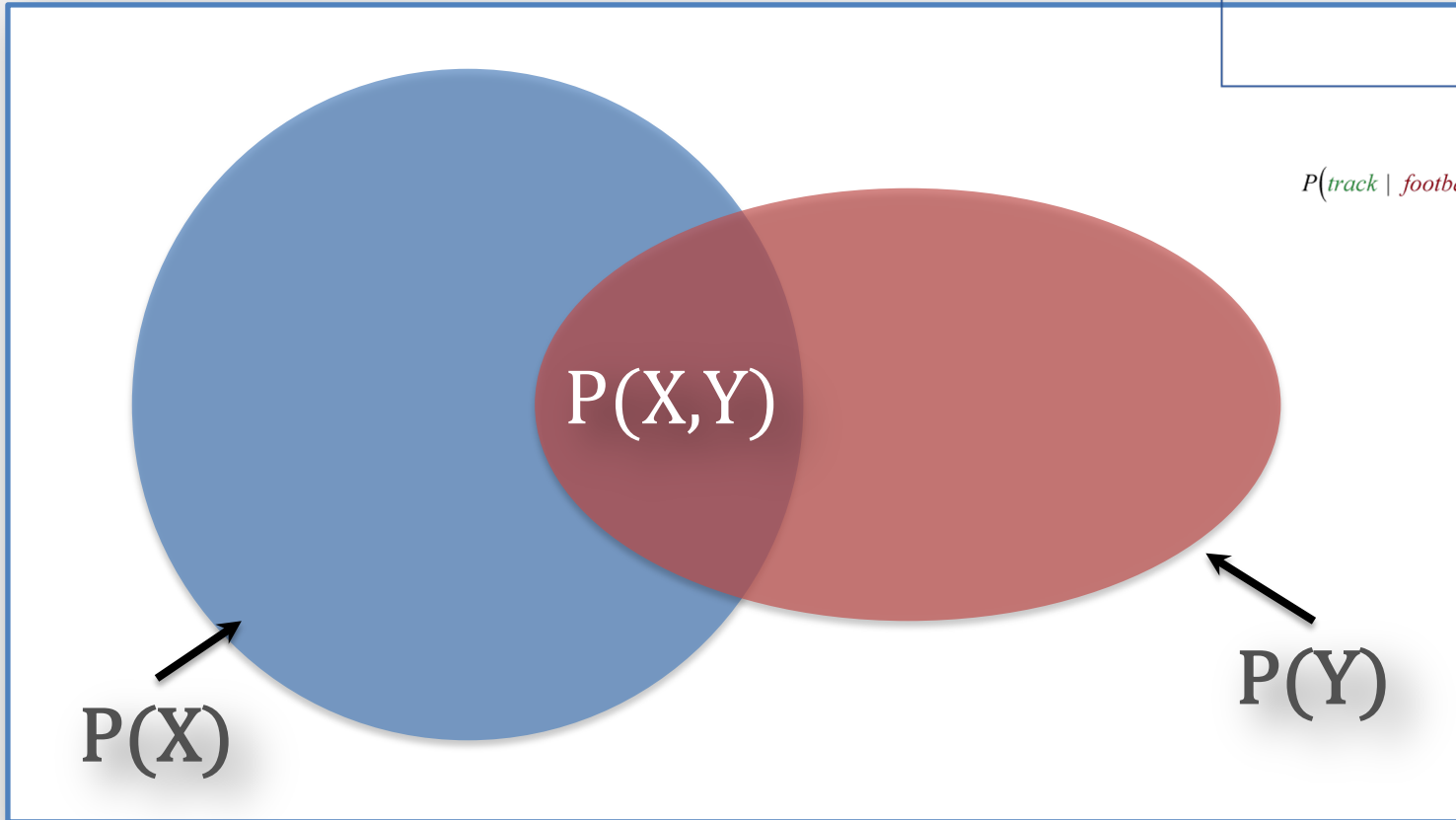
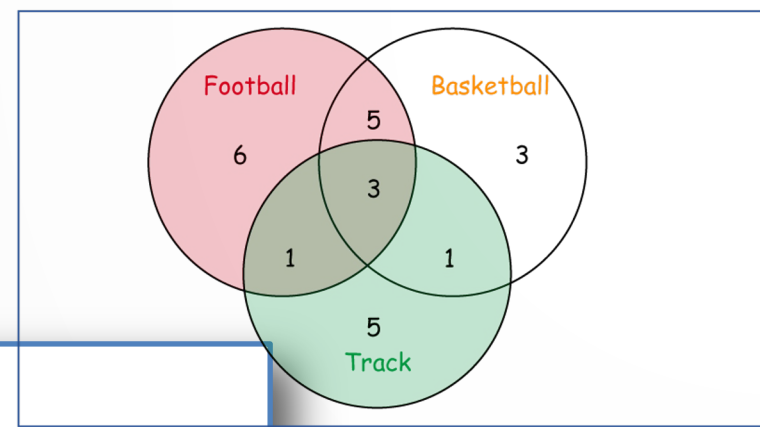
- First step towards practical purposes.
 - **Speech synthesis:** how to pronounce text
 - *I'm conTENT/JJ* vs. *the CONtent/NN*
 - *I obJECT/VBP* vs. *the OBject/NN*
 - *I lead/VBP ("l iy d")* vs. *it's lead/NN ("l eh d")*
 - **Information extraction:**
 - Quickly finding names and relations.
 - **Machine translation:**
 - Identifying grammatical 'chunks' is useful
 - **Explainability!**

Tagging as classification

- We have access to a **sequence of observations** and are expected to decide on the best assignment of a **hidden variable**, i.e., the PoS

Hidden variable				NN		
				VB		
		VBN		JJ		NN
	PRP	VBD	TO	RB	DT	VB
	Observation	she	promised	to	back	the

Reminder: Bayes' Rule



$$P(\text{track} | \text{football}) = \frac{P(\text{track} \cap \text{football})}{P(\text{football})} = \frac{\frac{4}{24}}{\frac{15}{24}} = \frac{4}{15}$$

Ca

$$P(X, Y) = P(X)P(Y|X)$$

$$P(X, Y) = P(Y)P(X|Y)$$

$$P(X|Y) = \frac{P(X)}{P(Y)} P(Y|X)$$

Statistical PoS tagging

- Determine the **most likely** tag sequence $t_{1:n}$ by:

$$\operatorname{argmax}_{t_{1:n}} P(t_{1:n} | w_{1:n}) = \operatorname{argmax}_{t_{1:n}} \frac{P(w_{1:n} | t_{1:n}) P(t_{1:n})}{P(w_{1:n})}$$

By Bayes' Rule

$$= \operatorname{argmax}_{t_{1:n}} \frac{P(w_{1:n} | t_{1:n}) P(t_{1:n})}{\cancel{P(w_{1:n})}}$$

Only maximize numerator

$$\approx \operatorname{argmax}_{t_{1:n}} \prod_i^n P(w_i | t_i) P(t_i | t_{i-1})$$

Assuming independence

Assuming Markov

Word likelihood probability $P(w_i|t_i)$

- **VBZ** (verb, 3rd person singular present) is likely *is*.
- Compute $P(\mathbf{is}|VBZ)$ by **counting** in a corpus that has **already** been **tagged**:

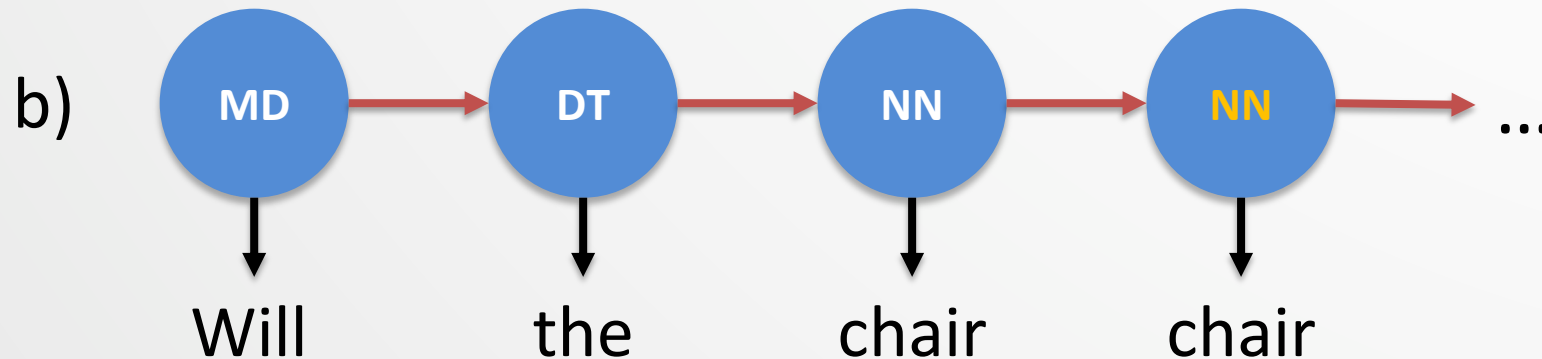
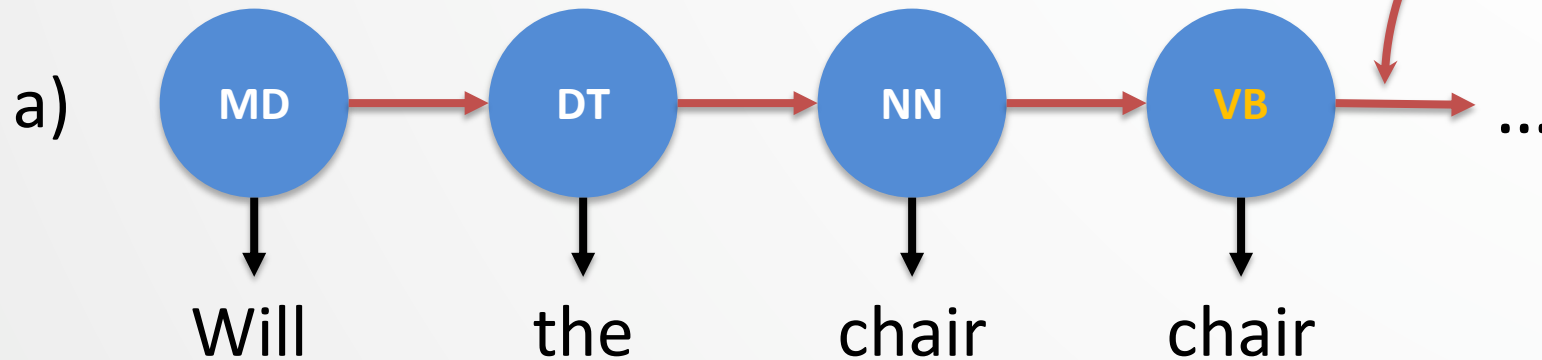
$$P(w_i|t_i) = \frac{\text{Count}(w_i \text{ tagged as } t_i)}{\text{Count}(t_i)}$$

e.g.,

$$P(\mathbf{is}|VBZ) = \frac{\text{Count}(\mathbf{is} \text{ tagged as } VBZ)}{\text{Count}(VBZ)} = \frac{10,073}{21,627} = 0.47$$

Tag-transition probability $P(t_i | t_{i-1})$

- *Will/MD the/DT chair/NN chair/?? the/DT meeting/NN from/IN that/DT chair/NN?*



Those are hidden Markov models!

- We'll see these soon...



Image sort of from *2001:A Space Odyssey*
by MGM pictures

Lecture Review Slide

- What are some examples of Text Classification
- **What are features?**
 - What are unique features for the specific tasks of sentiment analysis versus spam detection?
 - What are some words with multiple POS tags?
 - Compute Baye's rule for the POS tagging for an example.

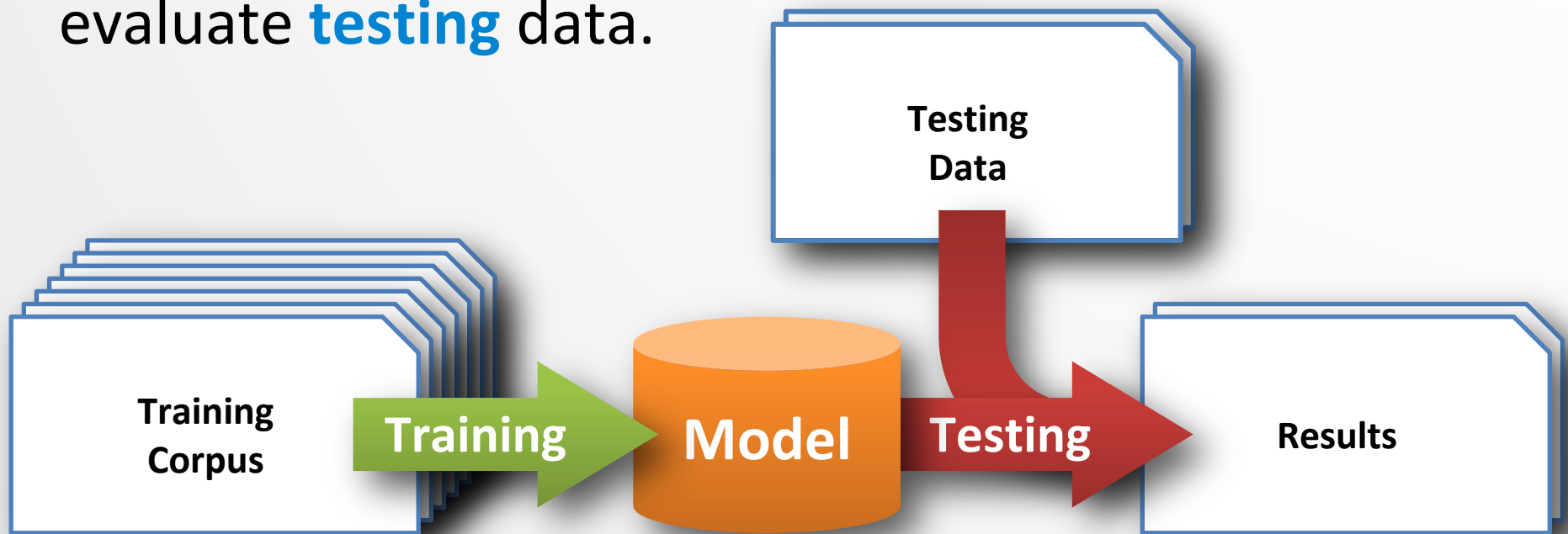


SCAN ME

Classification

General process

1. We gather a big and relevant **training** corpus.
2. We learn our **parameters** (e.g., probabilities) from that corpus to build our **model**.
3. Once that model is fixed, we use those probabilities to evaluate **testing** data.



General process

- Often, **training data** consists of 80% to 90% of the available data.
 - Often, some subset of *this* is used as a **validation/development set**.
- **Testing data** is **not** used for training but (usually) comes from the same *corpus*.
 - It often consists of the remaining available data.
 - Sometimes, it's important to **partition** speakers/writers so they **don't** appear in both training and testing.
 - *But what if we just randomized (un)luckily??*

Better process: K -fold cross-validation

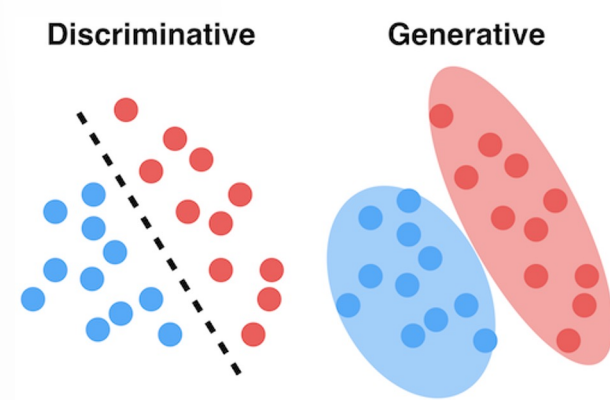
- **K -fold cross validation**: n . splitting all data into K **partitions** and iteratively testing on each after training on the rest (report means and variances).

	Part 1	Part 2	Part 3	Part 4	Part 5	
Iteration 1	Testing Set	Training Set	Training Set	Training Set	Training Set	: Err1 %
Iteration 2	Training Set	Testing Set	Training Set	Training Set	Training Set	: Err2 %
Iteration 3	Training Set	Training Set	Testing Set	Training Set	Training Set	: Err3 %
Iteration 4	Training Set	Training Set	Training Set	Testing Set	Training Set	: Err4 %
Iteration 5	Training Set	Training Set	Training Set	Training Set	Testing Set	: Err5 %

5-fold cross-validation

Testing Set
Training Set

(Some) Types of classifiers



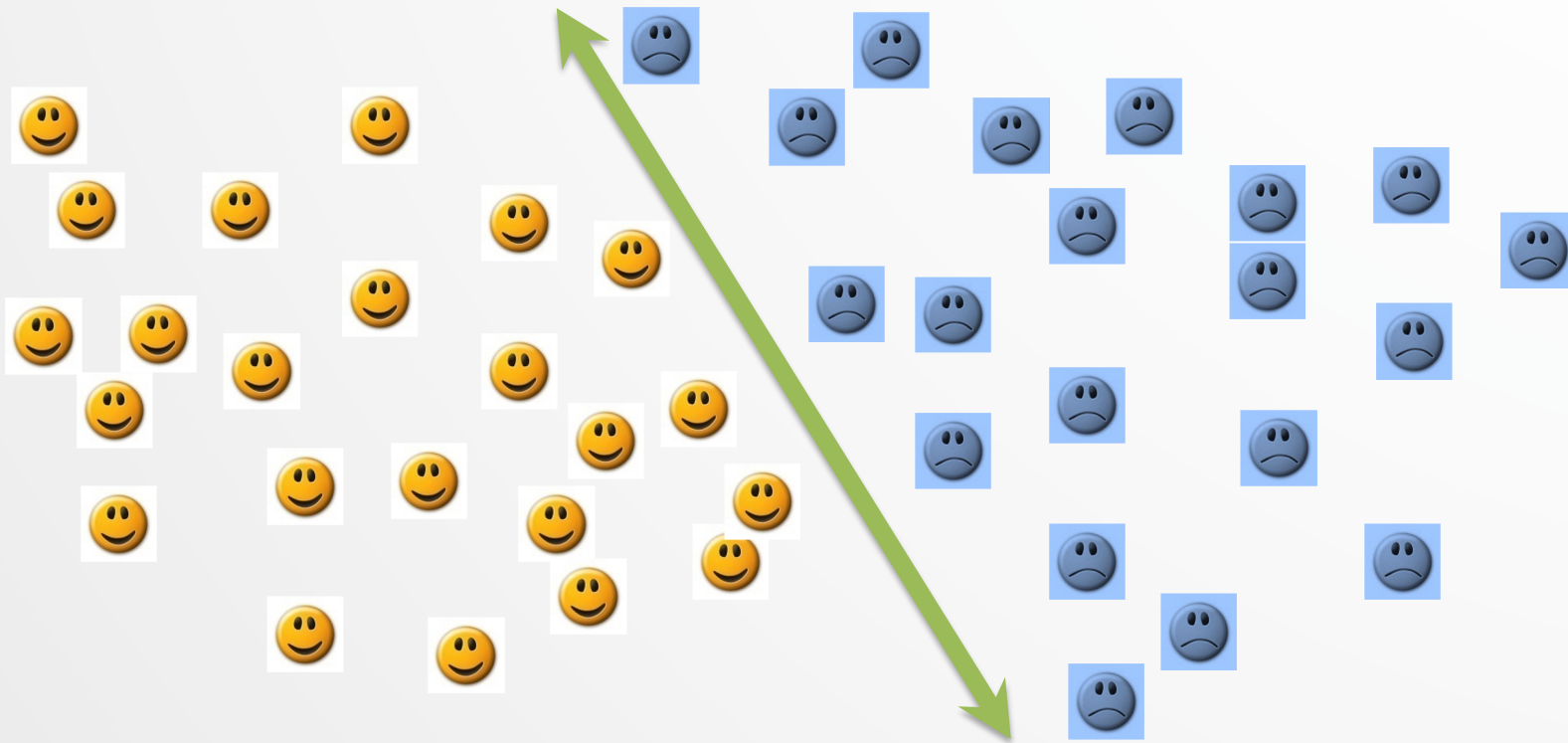
- **Generative** classifiers model the world.
 - Parameters set to maximize likelihood of training data.
 - We can *generate* new observations from these.
 - e.g., hidden Markov models

Vs.

- **Discriminative** classifiers emphasize **class boundaries**.
 - Parameters set to minimize error on training data.
 - e.g., support vector machines, decision trees.
- ...*What do class boundaries look like in the data?*

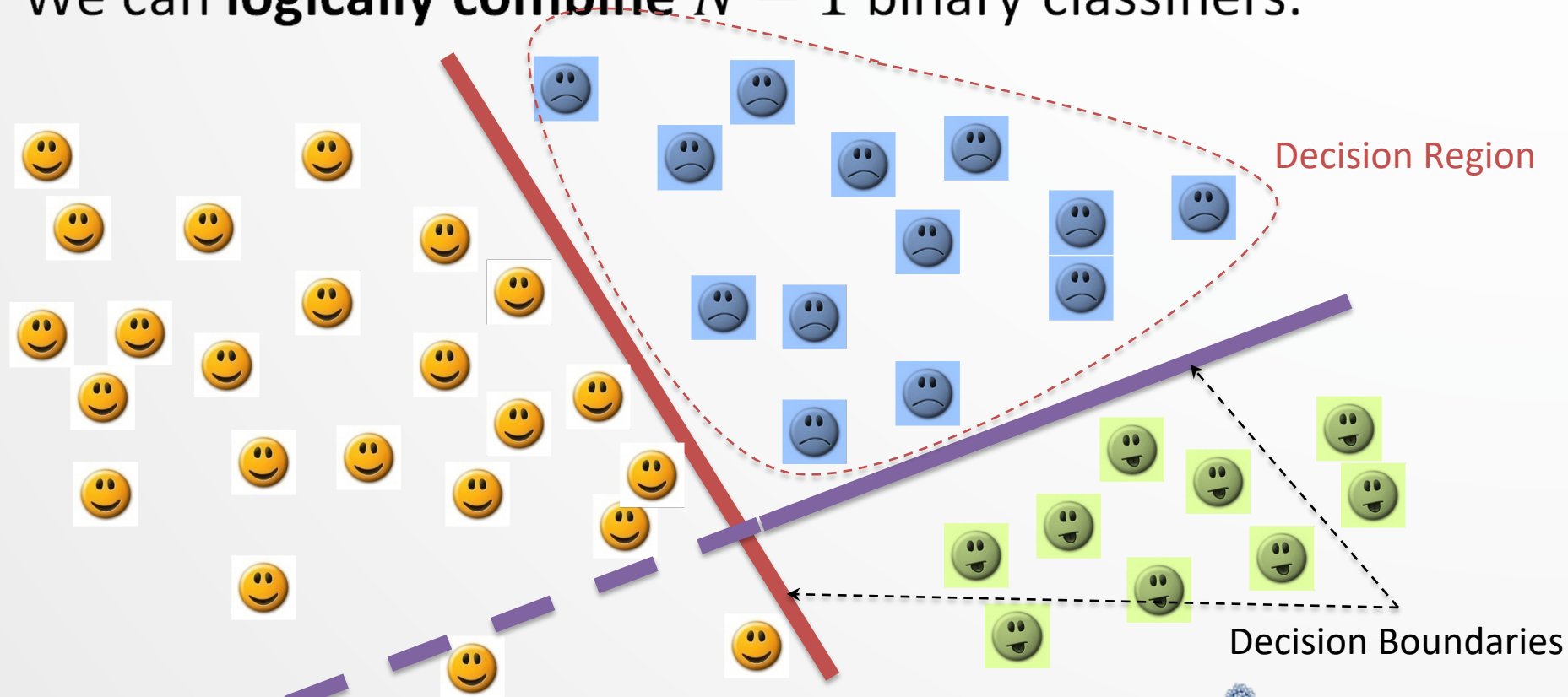
Binary and linearly separable

- Perhaps the easiest case.
 - Extends to dimensions $d \geq 3$, line becomes (hyper-)plane.



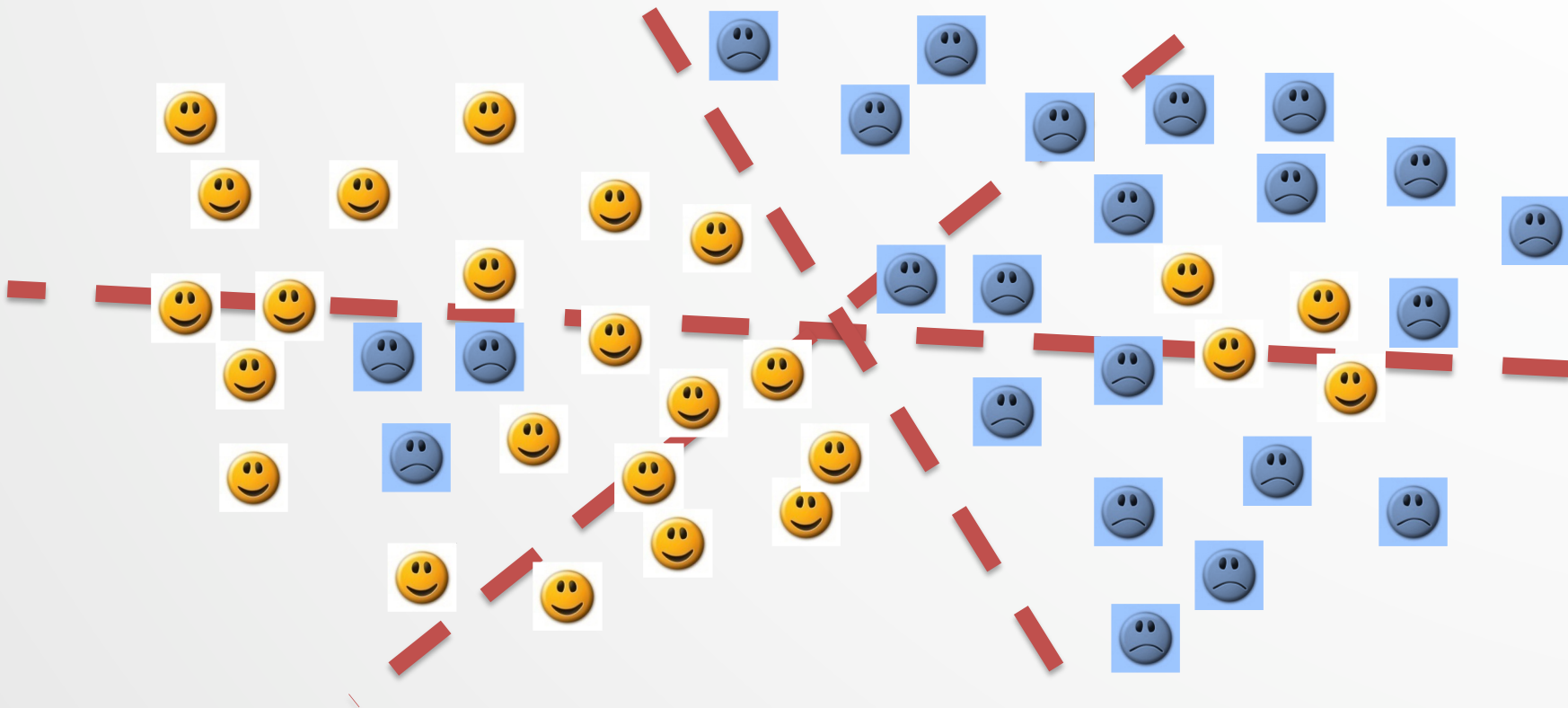
N-ary and linearly separable

- A bit harder – random guessing gives $\frac{1}{N}$ accuracy (given equally likely classes).
- We can **logically combine** $N - 1$ binary classifiers.



Class holes

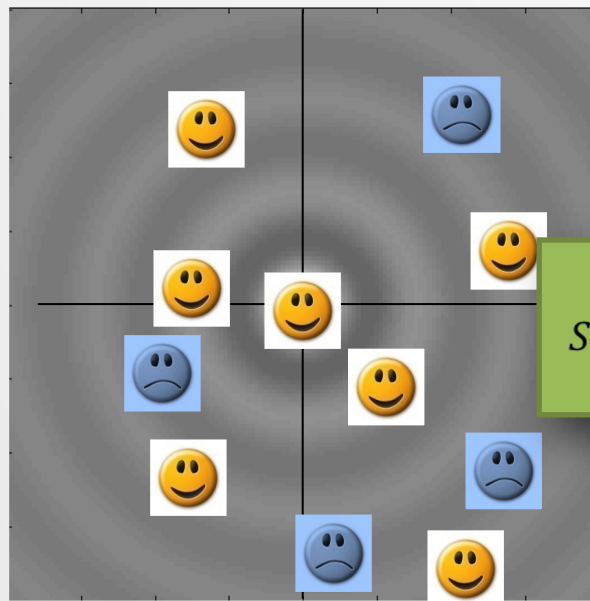
- Sometimes it can be impossible to draw *any* lines through the data to separate the classes.
 - *Are those troublesome points noise or real phenomena?*



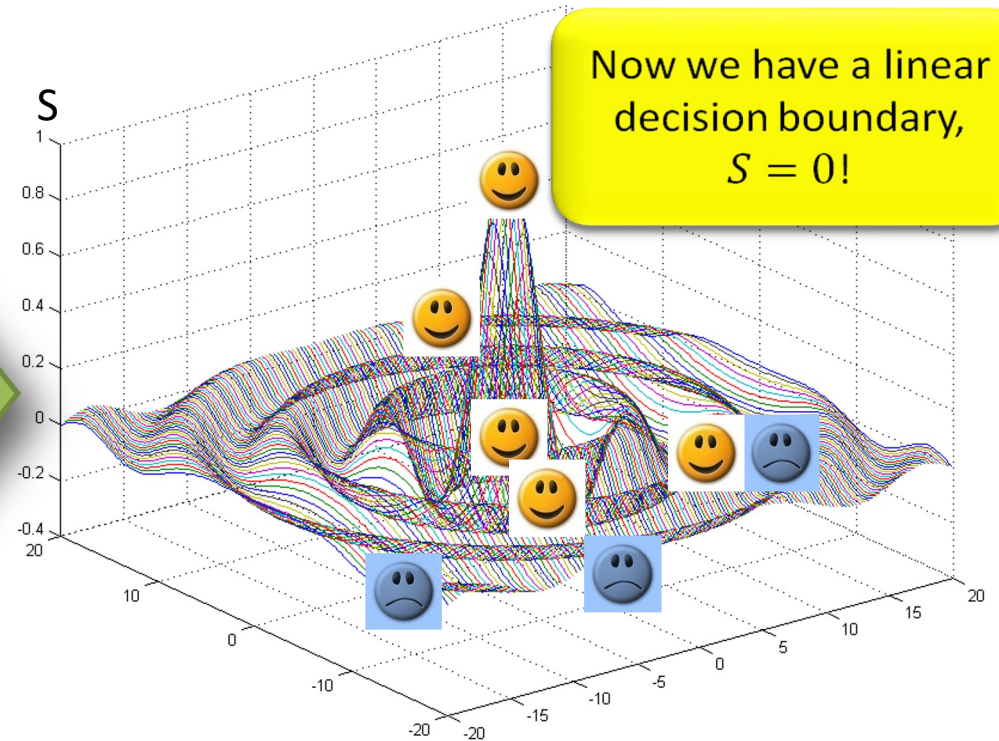
The kernel trick

- We can sometimes linearize a non-linear case by moving the data into a higher dimension with a **kernel function**.

E.g.,

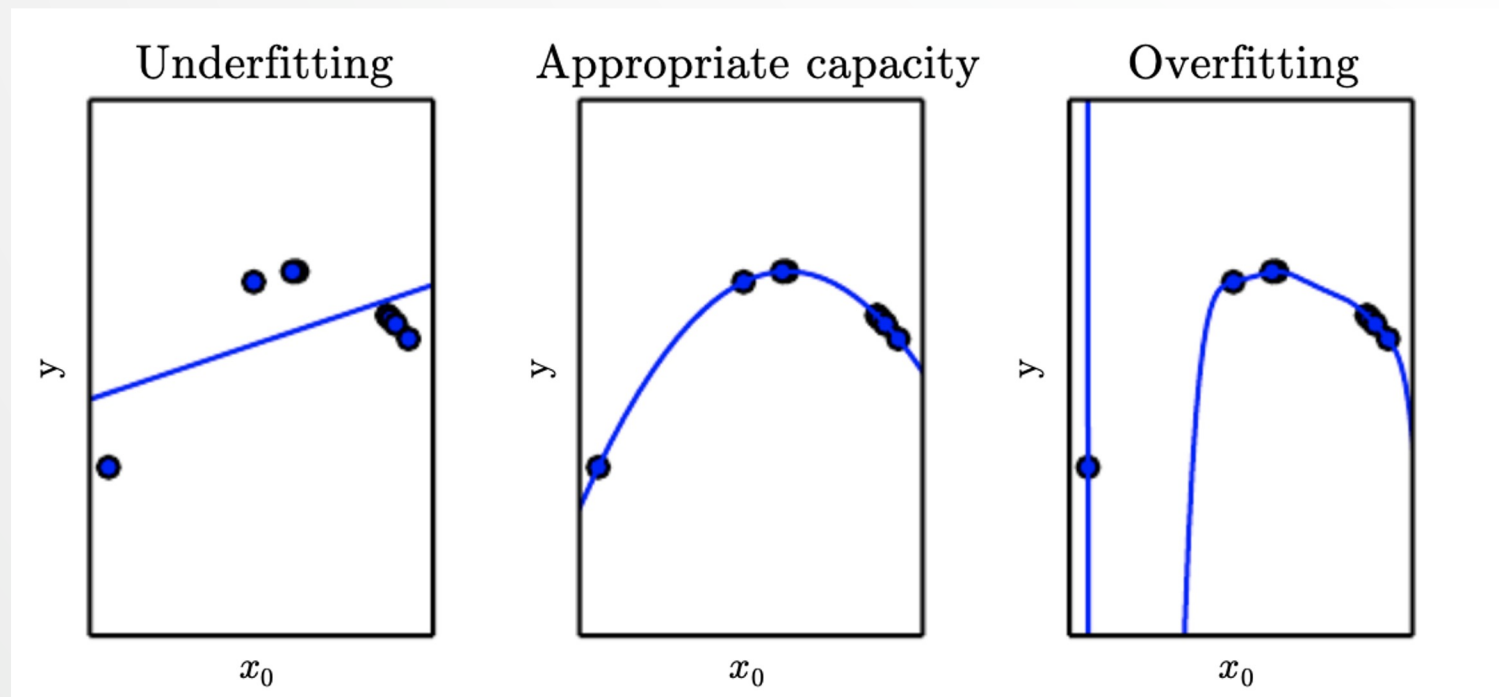


$$S = \frac{\sin(\sqrt{x^2 + y^2})}{\sqrt{x^2 + y^2}}$$



Capacity and over/under-fitting

- A central challenge in machine learning is that our models should **generalize** to unseen data, so we need to set our (hyper-)parameters appropriately.



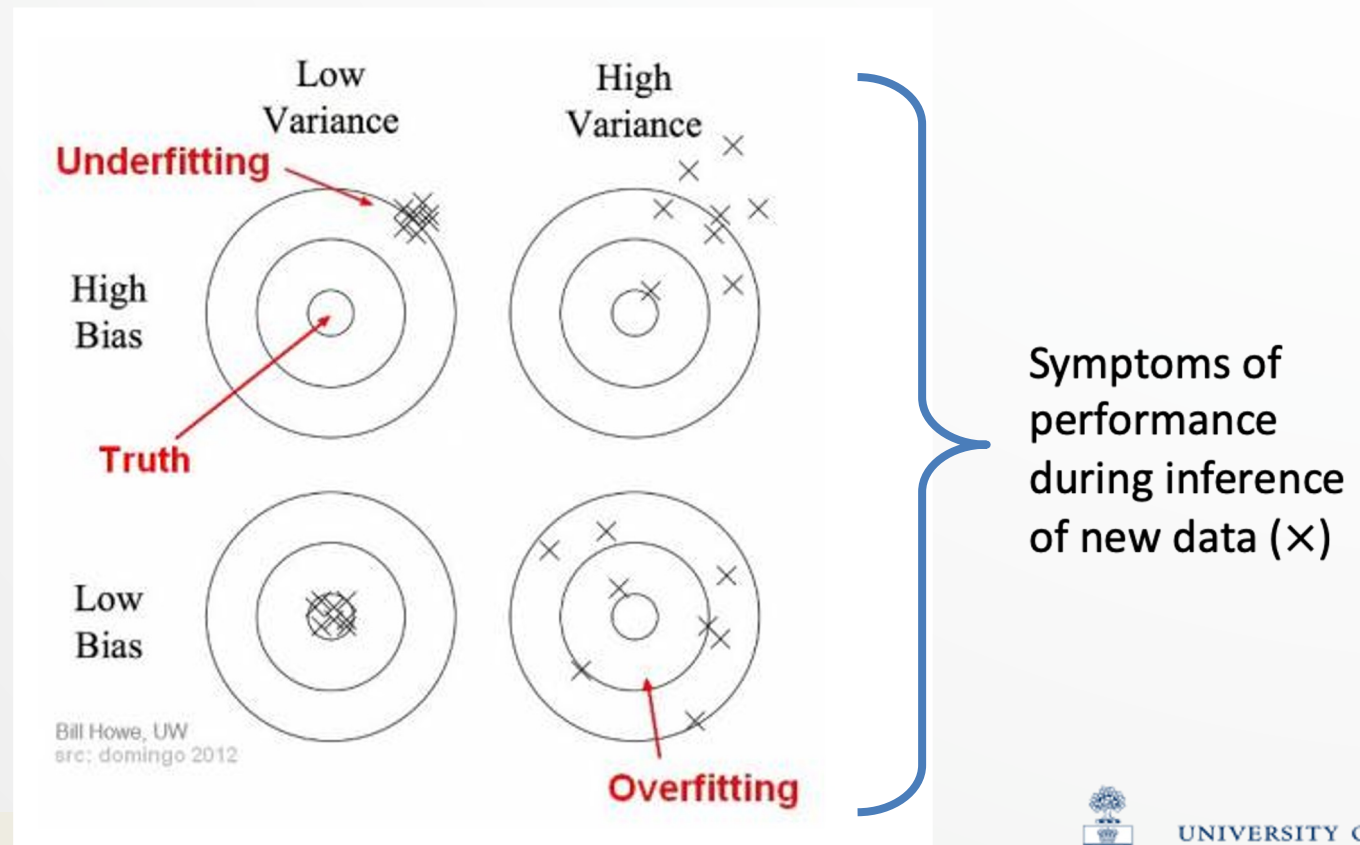
From Goodfellow

Model Complexity and Generalization

- ▶ **Underfitting**: too simplistic to describe the data.
- ▶ **Overfitting**: too complex, fits training data perfectly but does not generalize.
- ▶ **Hyperparameter**: can't include in training process itself, tuned using validation set.
- ▶ **Regularization**: Used to describe preferences of hypotheses to search over, L1/L2.
- ▶ **Bayes optimality**: Bayes error due to inherent unpredictability of targets, an algorithm that achieves the Bayes error is Bayes optimal.
- ▶ **Bias Variance Decomposition**: What is it, why is it useful, how do you calculate the bias/variance for different choices of loss functions?

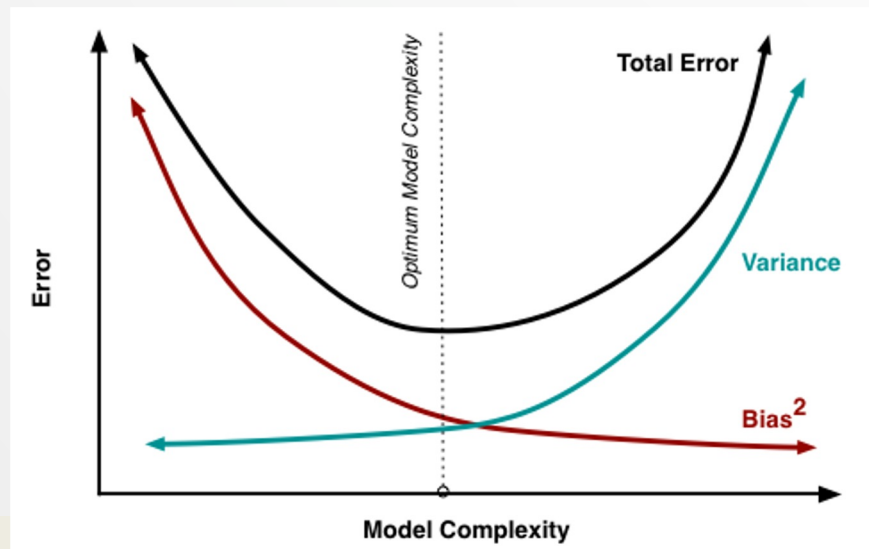
Capacity and over/under-fitting

- A central challenge in machine learning is that our models should **generalize** to unseen data, so we need to set our (hyper-)parameters appropriately.

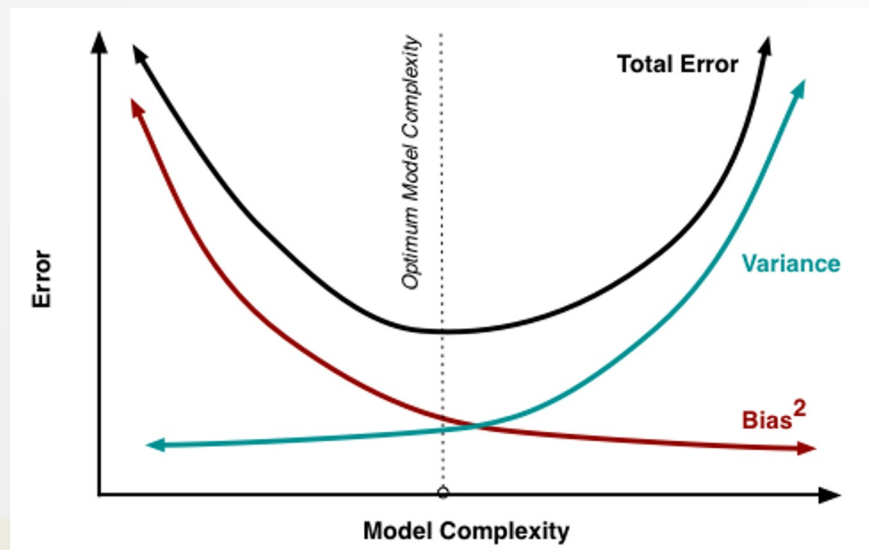
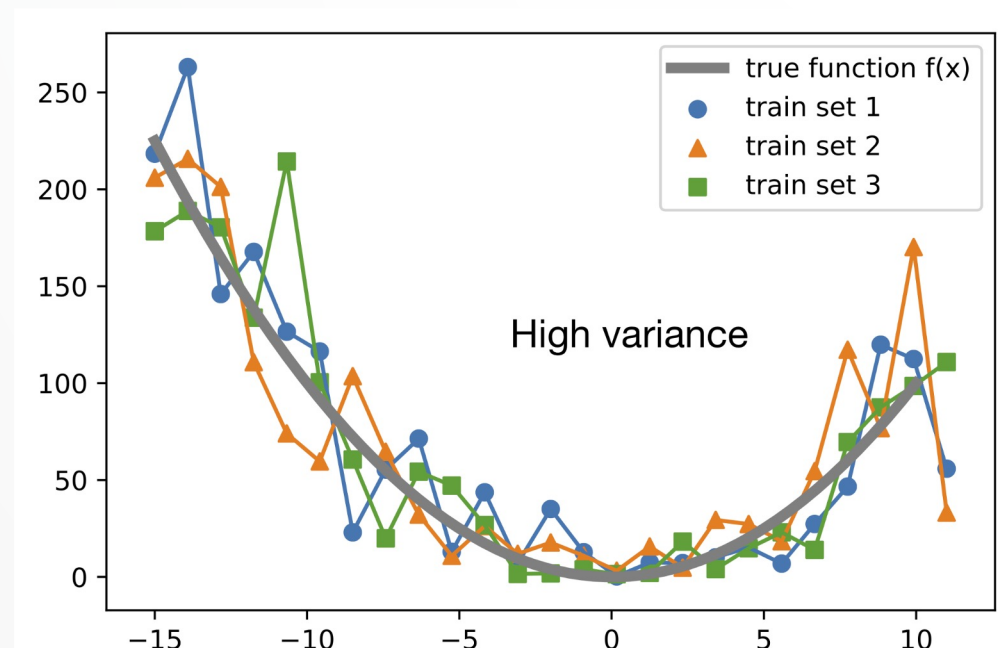
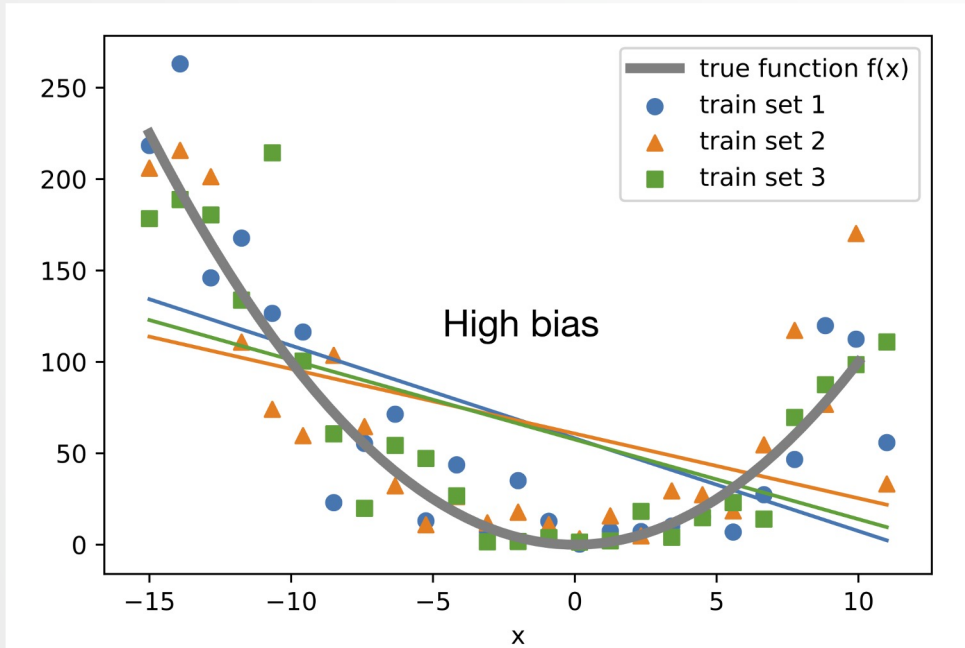


Bias and Variance

- ▶ Even though this analysis only applies to squared error, we often loosely use “bias” and “variance” as synonyms for “underfitting” and “overfitting”.
 - **Bias**: how wrong the expected prediction is (corresponds to underfitting).
 - **Variance**: the amount of variability in the predictions (corresponds to overfitting).



Bias and Variance





SCAN ME

Let's summarize a few of the classifiers from
Assignment 1



SCAN ME

https://docs.google.com/presentation/d/13B1t4GJ_HKpRdJj-W_UiVcGv0nn7uXaATeGFqJsA_pl/edit#slide=id.g1468dbc4b3f_0_280

Decision Tree

Trees!

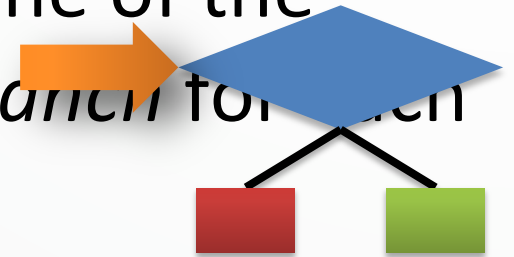


(The ... larch.)

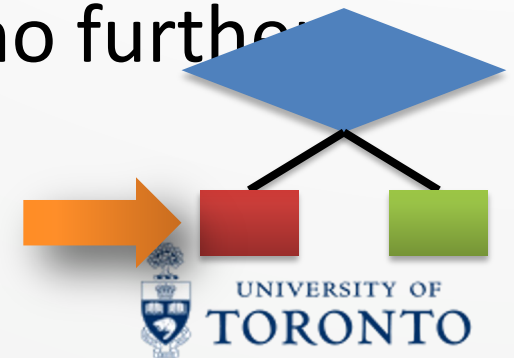
Decision trees

- Consists of **rules** for classifying data that have many **attributes** (features).

- **Decision nodes:** **Non-terminal.** Consists of a *question* asked of one of the attributes, and a *branch* for each possible answer.

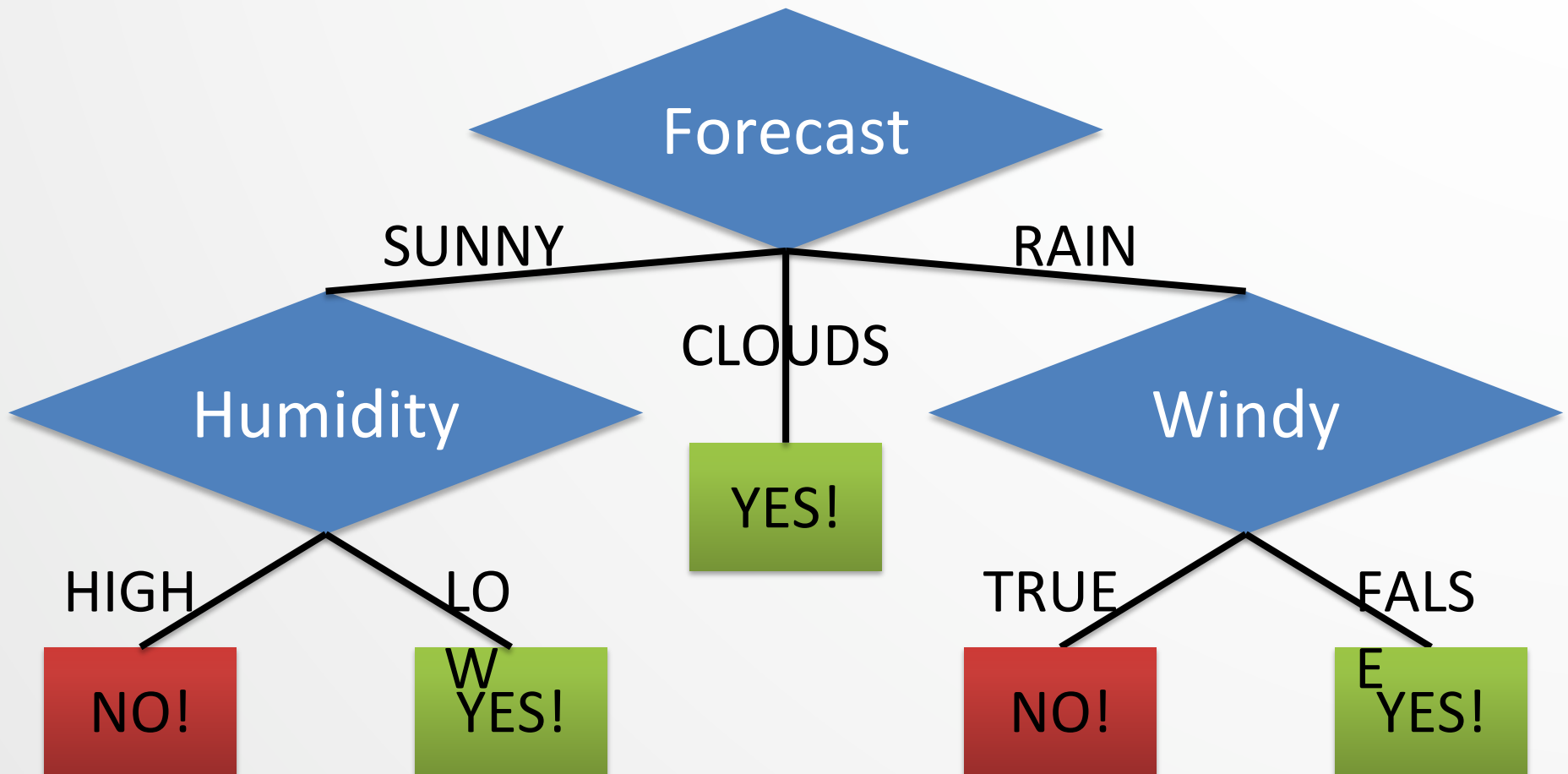


- **Leaf nodes:** **Terminal.** Consists of a single class/category, so no further testing is required.



Decision tree example

- Shall I go for a walk?



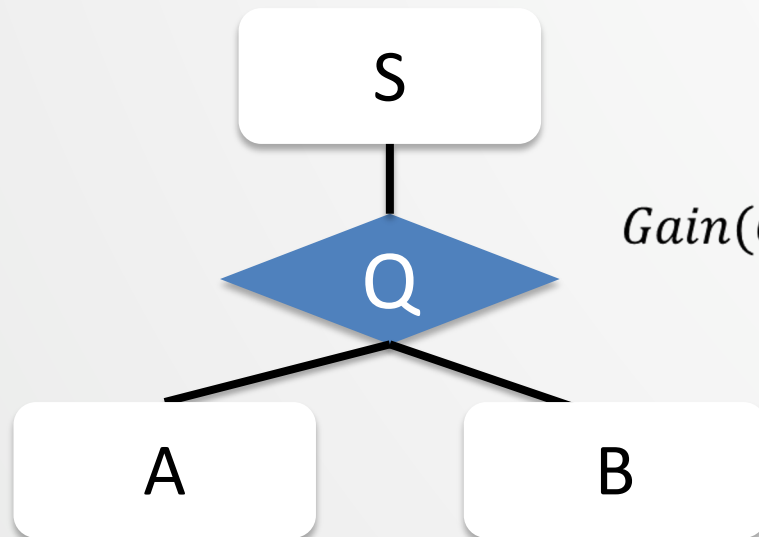
Decision tree algorithm: ID3

- **ID3** (iterative dichotomiser 3) is an algorithm invented by Ross Quinlan to produce decision trees from data.
- Basically,
 1. Compute the **uncertainty** of asking about **each feature**.
 2. Choose the feature which **reduces** the most uncertainty.
 3. **Make a node** asking a question of that feature.
 4. Go to step 1, **minus** the chosen feature.
- Example attribute vectors (observations):

Forecast	Humidity	Wind	
Avg. token length	Avg. sentence length	Frequency of nouns	...

Information gain

- The **information gain** is based on the expected decrease in entropy after a set of **training** data is split on an attribute.
 - We prefer the attribute that removes the most entropy.



$$S = A \cup B$$
$$\emptyset = A \cap B$$

$H(S)$? Wait to see Entropy near the end of the term...
It's a measure of uncertainty.

$$\text{Gain}(Q) = H(S) - \sum_{\text{child set}} p(\text{child set})H(\text{child set})$$

Each of S , A , and B consist of examples from the data

So $p(\text{child set})$ is computed by the proportion of examples in that set

Information gain and ID3

- When a node in the decision tree is **generated** in which **all members** have the **same** class,
 - that node has 0 entropy,
 - that node is a leaf node.
- Otherwise, we need to (try to) split that node with another question.
- See the *Appendix* of these slides for a complete example.

Aspects of ID3

- ID3 tends to build **short trees** since at each step we are removing the maximum amount of entropy possible.
- ID3 trains on the **whole training set** and does not succumb to issues related to **random initialization**.
- ID3 can **over-fit** to training data.
- Only **one attribute is used at a time** to make decisions
- It can be difficult to use **continuous** data, since many trees need to be generated to see where to break the continuum.

Random Forests

- **Random forests** *n.pl.* are **ensemble** classifiers that produce K decision trees, and output the **mode** class of those trees.
 - Can support continuous features.
 - Can support non-binary decisions.
 - Support cross-validation.
- The component trees in a random forest must differ.
 - Sometimes, decision trees are **pruned** randomly.
 - Usually, different trees accept different **subsets of features**.

*That's a good idea – can we choose the best features
in a reasonable way?*

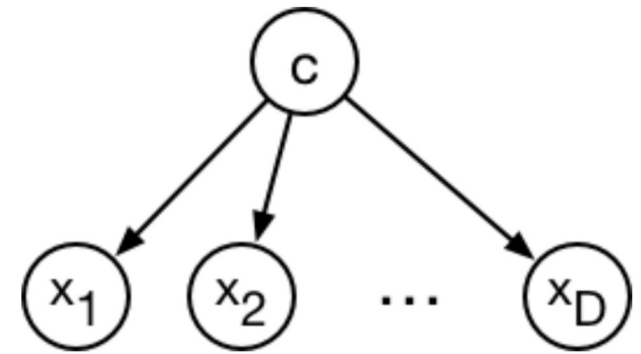
Naïve Bayes and SoftMax

- Broadly, Bayesian probability conceives of probability *not* as frequency of some phenomenon occurring, but rather as an expectation related to our own certainty.
- Given an observation x , **Naïve Bayes** simply chooses the class $c \in \mathcal{C}$ that maximizes $P(c | x)$.
 - This can be done in many ways.

$$\operatorname{argmax}_c P(c|x) = \frac{P(c)}{\cancel{P(x)}} P(x|c)$$

Estimate the $P(\cdot)$ using Gaussians, or...

Bayesian Classifier



Given features $\mathbf{x} = [x_1, x_2, \dots, x_D]^T$

want to compute class probabilities using Bayes Rule:

$$\underbrace{p(c|\mathbf{x})}_{\text{Pr. class given feature}} = \frac{\overbrace{p(\mathbf{x}|c)}^{\text{Pr. feature given class}} p(c)}{p(\mathbf{x})}$$

In words,

$$\text{Posterior for class} = \frac{\text{Pr. of feature given class} \times \text{Prior for class}}{\text{Pr. of feature}}$$

To compute $p(c|\mathbf{x})$ we need: $p(\mathbf{x}|c)$ and $p(c)$.

Independence Assumption

- ▶ Naive assumption: The features x_i are conditionally independent given the class c .
- ▶ Allows us to decompose the joint distribution:

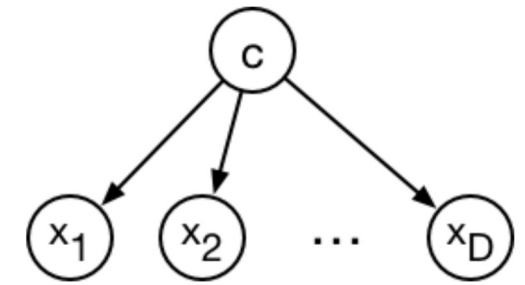
$$p(c, x_1, \dots, x_D) = p(c) p(x_1|c) \cdots p(x_D|c).$$

- ▶ Compact representation of the joint distribution.
 - Prior probability of class:
 - Conditional probability of feature given class:

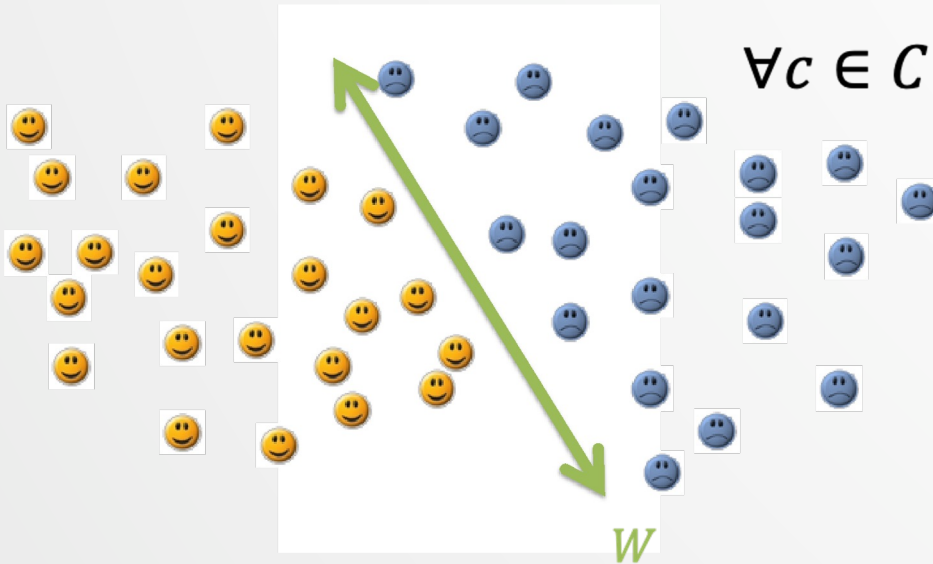
$$p(c = 1) = \pi$$

$$p(x_j = 1|c) = \theta_{jc}$$

Naïve Bayes and SoftMax



- Assume $x \in \mathbb{R}^d$, learning a linear decision boundary is tantamount to learning $W \in \mathbb{R}^{C \times d}$.



$$P(\text{Class}|\text{features}) = P(\text{features}|\text{Class}) \cdot P(\text{Class})$$
$$\forall c \in C: f_c = W[c, \dots] \cdot x = \sum_{i=1}^d W[c, i] \cdot x[i]$$

Uh oh – f_c can be negative and we want something on $[0,1]$, to be a probability.
Solution: Just raise it with an exponent

Softmax:

$$P(y|x) = \frac{\exp(f_y)}{\sum_{c=1}^C \exp(f_c)}$$

Naive Bayes: <https://www.youtube.com/watch?v=O2L2Uv9pdDA>

SoftMax: https://www.youtube.com/watch?v=8ps_JEW42xs

Example on Text: <https://www.youtube.com/watch?v=temQ8mHpe3k>

Naive Bayes on Spam: <https://youtu.be/M59h7CFUwPU>

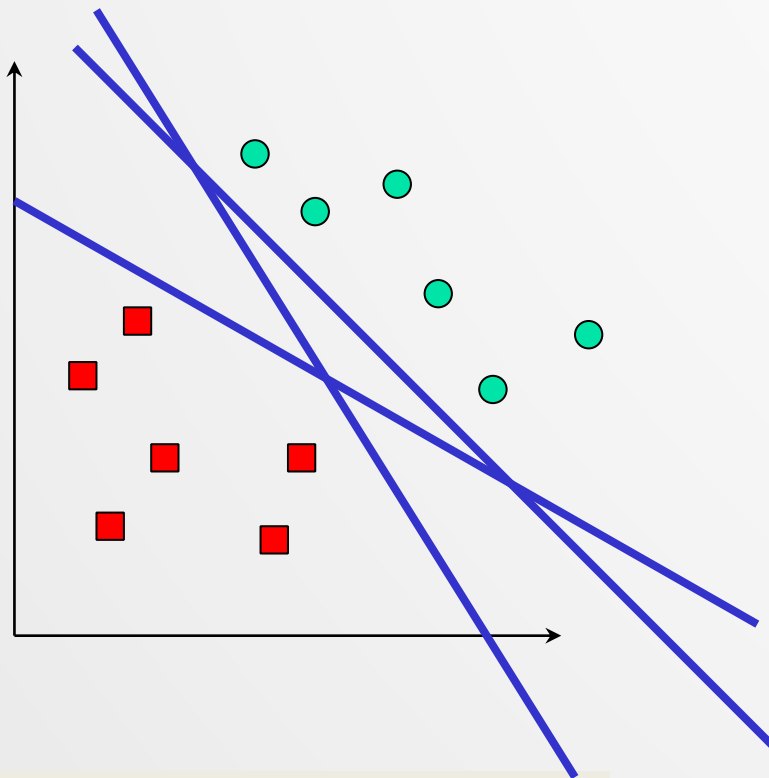
Why Naive Bayes are Cool: <https://www.youtube.com/watch?v=8NEfn3JbINA>

Naive Bayes Properties

- ▶ An amazingly cheap learning algorithm!
- ▶ **Training time:** Estimate parameters using maximum likelihood.
 - Compute co-occurrence counts of each feature with the labels. | Requires only one pass through the data!
- ▶ **Test time:** Apply Bayes' Rule.
 - Cheap because of the model structure. For more general models, Bayesian inference can be very expensive and/or complicated.
- ▶ Analysis easily extends to prob. distributions other than Bernoulli.
- ▶ Less accurate in practice compared to discriminative models due to its “naive” independence assumption.

Support vector machines (SVMs)

- In binary linear classification, two classes are assumed to be separable by a line (or plane). However, many possible separating planes might exist.

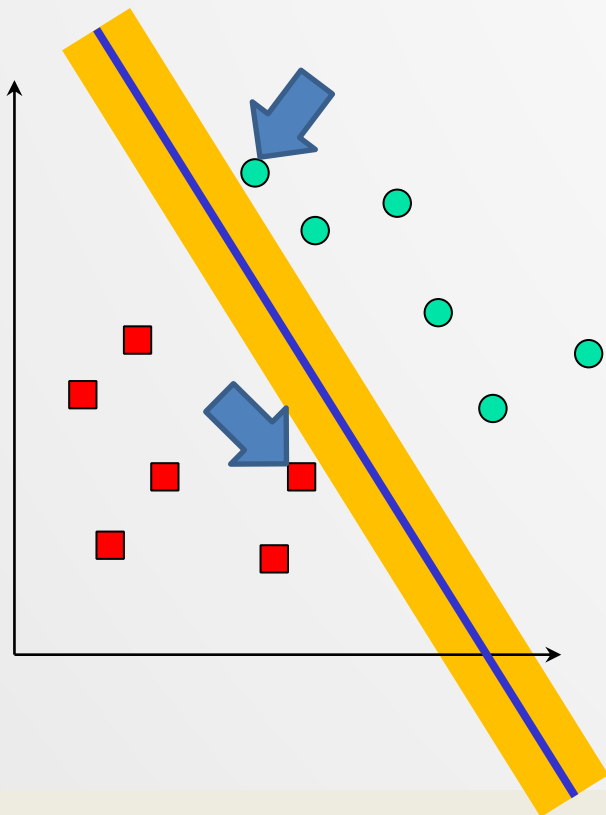


- Each of these blue lines separates the training data.
 - *Which line is the best?*



Support vector machines (SVMs)

- The **margin** is the width by which the boundary could be **increased** before it hits a training datum.



- The **maximum margin linear classifier** is \therefore the linear classifier with the maximum margin.
- The **support vectors** (indicated) are those data points against which the margin is pressed.
- The bigger the margin – the less sensitive the boundary is to error.

Support vector machines (SVMs)

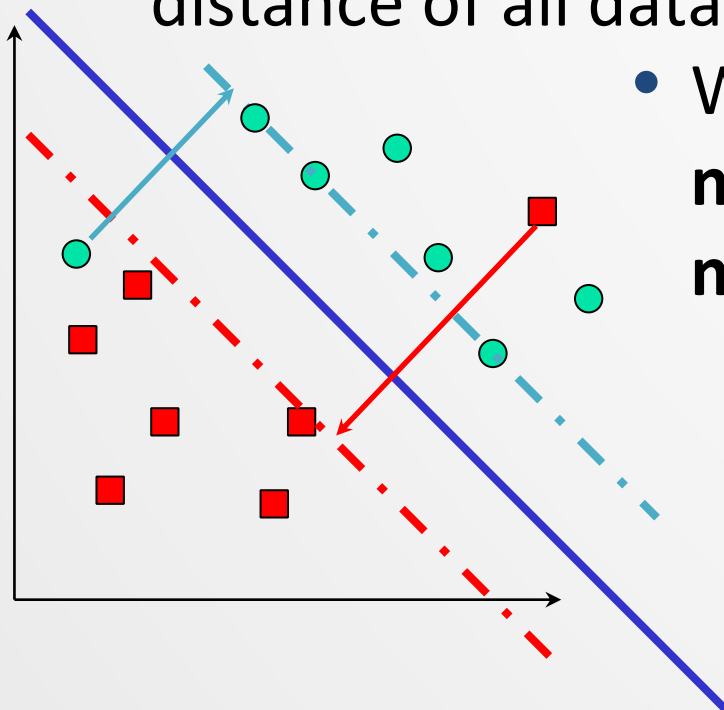
- The width of the margin, M , can be computed by the angle and displacement of the planar boundary, x , as well as the planes that touch data points.



- Given an initial guess of the angle and displacement of x we can compute:
 - whether all data is correctly classified,
 - The width of the margin, M .
- We update our guess by **quadratic programming**, which is semi-analytic.

Support vector machines (SVMs)

- The maximum margin helps SVMs **generalize** to situations when it's **impossible** to linearly separate the data.
 - We introduce a parameter that allows us to measure the distance of all data not in their correct 'zones'.



- We simultaneously **maximize the margin** while **minimizing the misclassification error**.
 - There is a straightforward approach to solving this system based on **quadratic programming**.

Support vector machines (SVMs)

- SVMs generalize to higher-dimensional data and to systems in which the data is non-linearly separable (e.g., by a circular decision boundary).
 - Using the **kernel trick** (from before) is common.
- Many binary SVM classifiers can also be combined to simulate a multi-category classifier.
- (Still) one of the most popular off-the-shelf classifiers.

Features and classification

- We talked about:
 - How preprocessing can effect feature extraction.
 - What parts-of-speech are, and how to identify them.
 - How to prepare data for classification
 - SVMs
 - Decision trees (which are parts of random forests)
- We've only taken a first step. Next week: Information Theory.

Lecture Review Slide

- Why do we use K-fold cross-validation and what should we watch out for?
- Explain Underfitting and Overfitting with respect to the Bias-Variance Tradeoff.
- What does SVM maximize and minimize simultaneously?
- What are some tricks that SVM uses?

Readings

- J&M: 5.1-5.5 (2nd edition)
- M&S: 16.1, 16.4

Appendix – prepositions from CELEX

of	540,085	through	14,964	worth	1,563	pace	12
in	331,235	after	13,670	toward	1,390	nigh	9
for	142,421	between	13,275	plus	750	re	4
to	125,691	under	9,525	till	686	mid	3
with	124,965	per	6,515	amongst	525	o'er	2
on	109,129	among	5,090	via	351	but	0
at	100,169	within	5,030	amid	222	ere	0
by	77,794	towards	4,700	underneath	164	less	0
from	74,843	above	3,056	versus	113	midst	0
about	38,428	near	2,026	amidst	67	o'	0
than	20,210	off	1,695	sans	20	thru	0
over	18,071	past	1,575	circa	14	vice	0

Appendix – particles

aboard	aside	besides	forward(s)	opposite	through
about	astray	between	home	out	throughout
above	away	beyond	in	outside	together
across	back	by	inside	over	under
ahead	before	close	instead	overhead	underneath
alongside	behind	down	near	past	up
apart	below	east, etc.	off	round	within
around	beneath	eastward(s),etc.	on	since	without

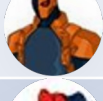
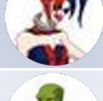
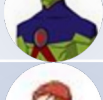
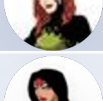
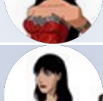
Appendix – conjunctions

and	514,946	yet	5,040	considering	174	forasmuch as	0
that	134,773	since	4,843	lest	131	however	0
but	96,889	where	3,952	albeit	104	immediately	0
or	76,563	nor	3,078	providing	96	in as far as	0
as	54,608	once	2,826	whereupon	85	in so far as	0
if	53,917	unless	2,205	seeing	63	inasmuch as	0
when	37,975	why	1,333	directly	26	insomuch as	0
because	23,626	now	1,290	ere	12	insomuch that	0
so	12,933	neither	1,120	notwithstanding	3	like	0
before	10,720	whenever	913	according as	0	neither nor	0
though	10,329	whereas	867	as if	0	now that	0
than	9,511	except	864	as long as	0	only	0
while	8,144	till	686	as though	0	provided that	0
after	7,042	provided	594	both and	0	providing that	0
whether	5,978	whilst	351	but that	0	seeing as	0
for	5,935	suppose	281	but then	0	seeing as how	0
although	5,424	cos	188	but then again	0	seeing that	0
until	5,072	supposing	185	either or	0	without	0

Appendix – Penn TreeBank PoS tags

Tag	Description	Example	Tag	Description	Example
CC	coordin. conjunction	<i>and, but, or</i>	SYM	symbol	<i>+, %, &</i>
CD	cardinal number	<i>one, two, three</i>	TO	“to”	<i>to</i>
DT	determiner	<i>a, the</i>	UH	interjection	<i>ah, oops</i>
EX	existential ‘there’	<i>there</i>	VB	verb, base form	<i>eat</i>
FW	foreign word	<i>mea culpa</i>	VBD	verb, past tense	<i>ate</i>
IN	preposition/sub-conj	<i>of, in, by</i>	VBG	verb, gerund	<i>eating</i>
JJ	adjective	<i>yellow</i>	VBN	verb, past participle	<i>eaten</i>
JJR	adj., comparative	<i>bigger</i>	VBP	verb, non-3sg pres	<i>eat</i>
JJS	adj., superlative	<i>wildest</i>	VBZ	verb, 3sg pres	<i>eats</i>
LS	list item marker	<i>1, 2, One</i>	WDT	wh-determiner	<i>which, that</i>
MD	modal	<i>can, should</i>	WP	wh-pronoun	<i>what, who</i>
NN	noun, sing. or mass	<i>llama</i>	WP\$	possessive wh-	<i>whose</i>
NNS	noun, plural	<i>llamas</i>	WRB	wh-adverb	<i>how, where</i>
NNP	proper noun, singular	<i>IBM</i>	\$	dollar sign	<i>\$</i>
NNPS	proper noun, plural	<i>Carolinas</i>	#	pound sign	<i>#</i>
PDT	predeterminer	<i>all, both</i>	“	left quote	<i>‘ or “</i>
POS	possessive ending	<i>'s</i>	”	right quote	<i>’ or ”</i>
PRP	personal pronoun	<i>I, you, he</i>	(left parenthesis	<i>[, (, {, <</i>
PRP\$	possessive pronoun	<i>your, one’s</i>)	right parenthesis	<i>],), }, ></i>
RB	adverb	<i>quickly, never</i>	,	comma	<i>,</i>
RBR	adverb, comparative	<i>faster</i>	.	sentence-final punc	<i>. ! ?</i>
RBS	adverb, superlative	<i>fastest</i>	:	mid-sentence punc	<i>: ; ... --</i>
RP	particle	<i>up, off</i>			

Example – Hero classification

	Hero	Hair length	Height	Age	Hero Type
Training data	 Aquaman	2"	6'2"	35	Hero
	 Batman	1"	5'11"	32	Hero
	 Catwoman	7"	5'9"	29	Villain
	 Deathstroke	0"	6'4"	28	Villain
	 Harley Quinn	5"	5'0"	27	Villain
	 Martian Manhunter	0"	8'2"	128	Hero
	 Poison Ivy	6"	5'2"	24	Villain
	 Wonder Woman	6"	6'1"	108	Hero
	 Zatanna	10"	5'8"	26	Hero
Test data	 Red Hood	2"	6'0"	22	?

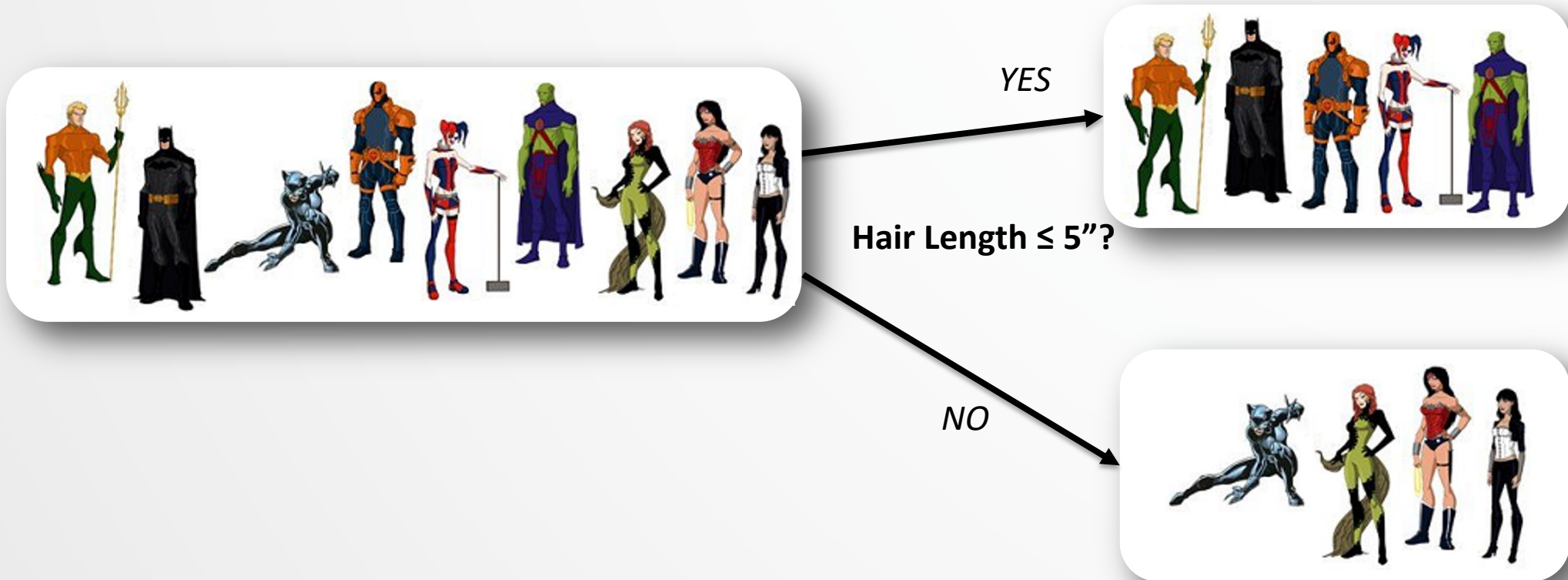
Example – Hero classification

- How do we split?
 - Split on *hair length*?
 - Split on *height*?
 - Split on *age*?
- Let's compute the **information gain** for each:

$$Gain(Q) = H(S) - \sum_{child\ set} p(child\ set)H(child\ set)$$

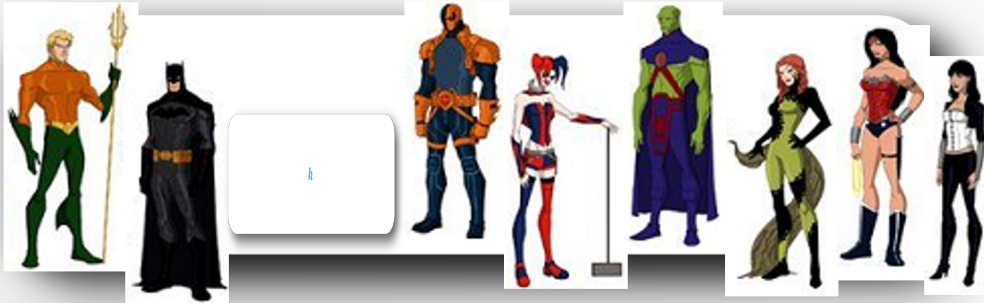
Split on hair length?

$$\text{Gain}(\text{Question}) = H(S) - \sum_{\text{child set}} p(\text{child set})H(\text{child set})$$



Split on hair length?

$$\text{Gain}(\text{Question}) = H(S) - \sum_{\text{child set}} p(\text{child set})H(\text{child set})$$



YES



Hair Length $\leq 5''$?

NO



$$H(S) = \frac{h}{h+v} \log_2 \left(\frac{h+v}{h} \right) + \frac{v}{h+v} \log_2 \left(\frac{h+v}{v} \right)$$

$$H(5h, 4v) = \frac{5}{9} \log_2 \left(\frac{9}{5} \right) + \frac{4}{9} \log_2 \left(\frac{9}{4} \right) = 0.9911 \text{ bits}$$

Split on hair length?

$$\text{Gain}(\text{Question}) = H(S) - \sum_{\text{child set}} p(\text{child set})H(\text{child set})$$



Hair Length ≤ 5 "?

YES



NO



Split on hair length?

$$\text{Gain}(\text{Question}) = H(S) - \sum_{\text{child set}} p(\text{child set})H(\text{child set})$$



YES



Hair Length ≤ 5 "?

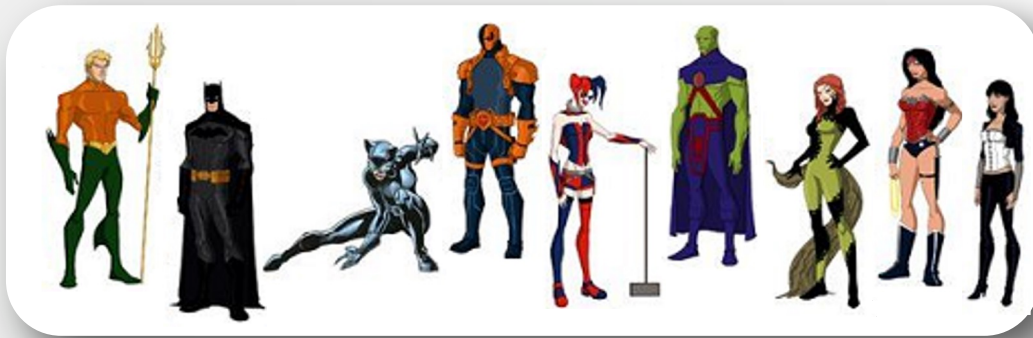
NO



YES $H(4h, 1v) = \frac{4}{5} \log_2 \left(\frac{5}{4} \right) + \frac{1}{5} \log_2 \left(\frac{5}{1} \right) = 0.7219$

Split on hair length?

$$\text{Gain}(\text{Question}) = H(S) - \sum_{\text{child set}} p(\text{child set})H(\text{child set})$$



YES

Hair Length ≤ 5 "?



NO



YES $H(4h, 1v) = \frac{4}{5} \log_2 \left(\frac{5}{4} \right) + \frac{1}{5} \log_2 \left(\frac{5}{1} \right) = 0.7219$

NO $H(2h, 2v) = \frac{2}{4} \log_2 \left(\frac{4}{2} \right) + \frac{2}{4} \log_2 \left(\frac{4}{2} \right) = 1$

Split on hair length?

$$\text{Gain}(\text{Question}) = H(S) - \sum_{\text{child set}} p(\text{child set})H(\text{child set})$$



YES

Hair Length ≤ 5 "?



NO



$$\text{Gain}(\text{HairLength} \leq 5") = \underbrace{0.9911}_{\text{H(S)}} - \frac{5}{9} \underbrace{0.7219}_{\text{H(child set)}} - \frac{4}{9} 1 = 0.00721$$

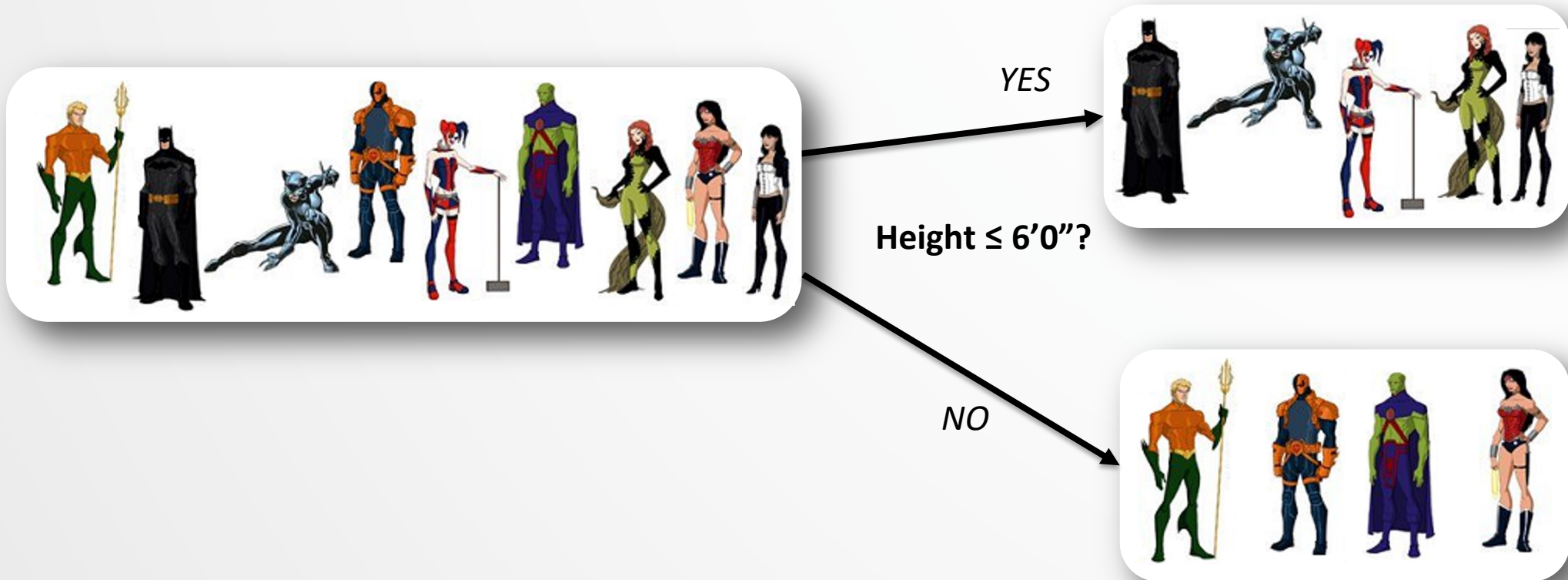
Example – Hero classification

- How do we split?
 - Split on *hair length*? $Gain(HairLength \leq 5") = 0.00721$
 - Split on *height*?
 - Split on *age*?
- Let's compute the **information gain** for each:

$$Gain(Q) = H(S) - \sum_{child\ set} p(child\ set)H(child\ set)$$

Split on height?

$$\text{Gain}(\text{Question}) = H(S) - \sum_{\text{child set}} p(\text{child set})H(\text{child set})$$



Split on height?

$$\text{Gain}(\text{Question}) = H(S) - \sum_{\text{child set}} p(\text{child set})H(\text{child set})$$

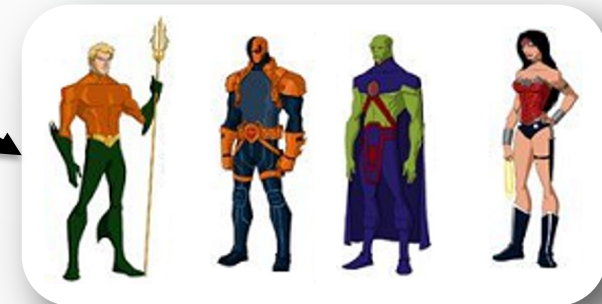


YES



Height $\leq 6'0''$?

NO

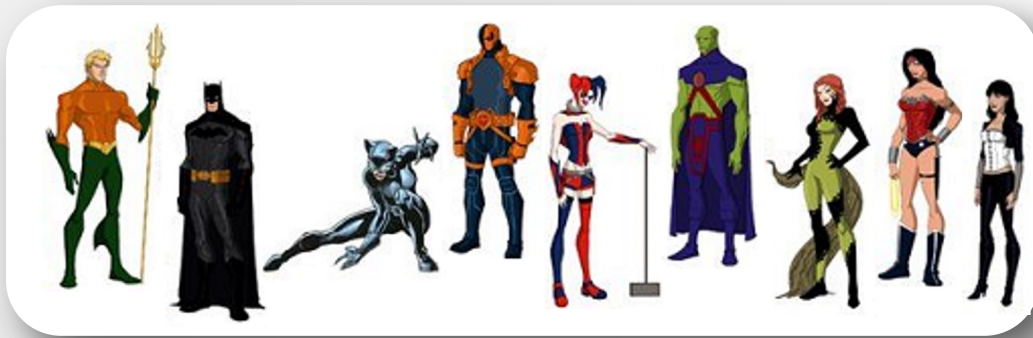


$$H(S) = \frac{h}{h+v} \log_2 \left(\frac{h+v}{h} \right) + \frac{v}{h+v} \log_2 \left(\frac{h+v}{v} \right)$$

$$H(5h, 4v) = \frac{5}{9} \log_2 \left(\frac{9}{5} \right) + \frac{4}{9} \log_2 \left(\frac{9}{4} \right) = \mathbf{0.9911 \text{ bits}}$$

Split on height?

$$\text{Gain}(\text{Question}) = H(S) - \sum_{\text{child set}} p(\text{child set})H(\text{child set})$$

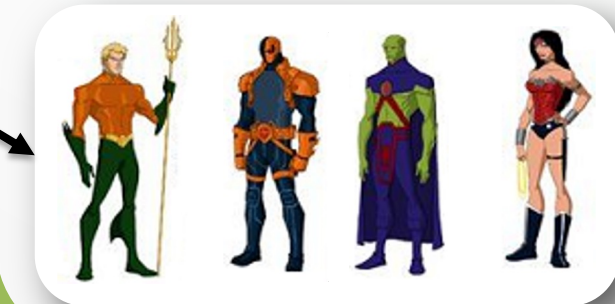


YES

Height $\leq 6'0''$



NO



YES $H(2h, 3v) = \frac{2}{5} \log_2 \left(\frac{5}{2} \right) + \frac{3}{5} \log_2 \left(\frac{5}{3} \right) = 0.971$

NO $H(3h, 1v) = \frac{3}{4} \log_2 \left(\frac{4}{3} \right) + \frac{1}{4} \log_2 \left(\frac{4}{1} \right) = 0.813$

Split on height?

$$\text{Gain}(\text{Question}) = H(S) - \sum_{\text{child set}} p(\text{child set})H(\text{child set})$$

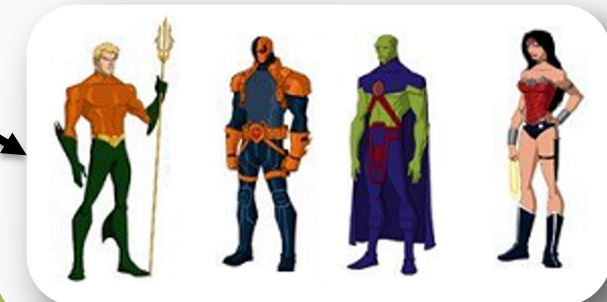


Height \leq 6'0"

YES



NO



$$\text{Gain}(\text{Height} \leq 6'0") = 0.9911 - \frac{5}{9} [0.971] - \frac{4}{9} [0.813] = 0.0903$$

Example – Hero classification

- How do we split?
 - Split on *hair length*? $Gain(HairLength \leq 5") = 0.00721$
 - Split on *height*?
 - Split on *age*?
- Let's compute the **information gain** for each:

$$Gain(Q) = H(S) - \sum_{child\ set} p(child\ set)H(child\ set)$$

Split on age?

$$\text{Gain}(\text{Question}) = H(S) - \sum_{\text{child set}} p(\text{child set})H(\text{child set})$$



YES
Age ≤ 30?



NO

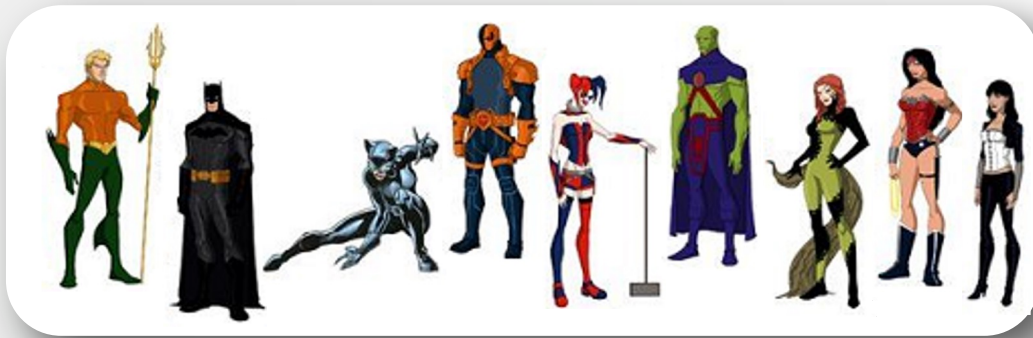


$$H(S) = \frac{h}{h+v} \log_2 \left(\frac{h+v}{h} \right) + \frac{v}{h+v} \log_2 \left(\frac{h+v}{v} \right)$$

$$H(5h, 4v) = \frac{5}{9} \log_2 \left(\frac{9}{5} \right) + \frac{4}{9} \log_2 \left(\frac{9}{4} \right) = 0.9911 \text{ bits}$$

Split on age?

$$\text{Gain}(\text{Question}) = H(S) - \sum_{\text{child set}} p(\text{child set})H(\text{child set})$$



Age ≤ 30?

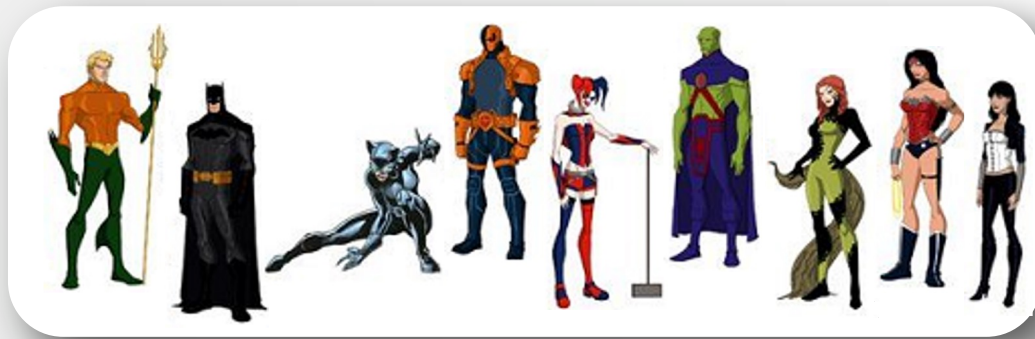


YES $H(1h, 4v) = \frac{1}{5} \log_2 \left(\frac{5}{1} \right) + \frac{4}{5} \log_2 \left(\frac{5}{4} \right) = 0.722$

NO $H(4h, 0v) = \frac{4}{4} \log_2 \left(\frac{4}{4} \right) + \frac{0}{4} \log_2(\infty) = 0$

Split on age?

$$\text{Gain}(\text{Question}) = H(S) - \sum_{\text{child set}} p(\text{child set})H(\text{child set})$$



Age ≤ 30?

YES



NO



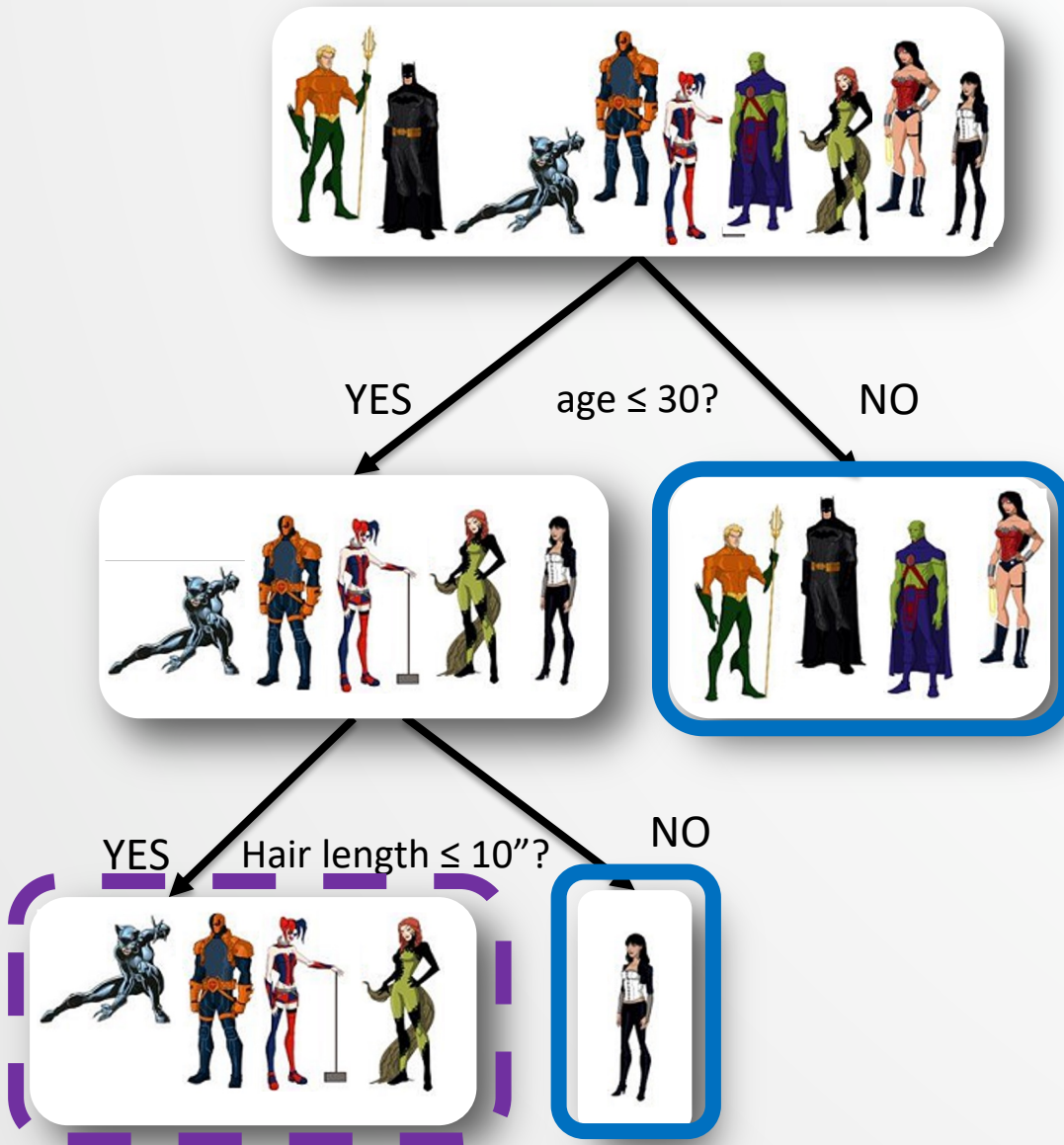
$$\text{Gain}(\text{Age} \leq 30) = 0.9911 - \frac{5}{9} [0.722] - \frac{4}{9} [0] = 0.590$$

Example – Hero classification

- How do we split?
 - Split on *hair length*? $Gain(HairLength \leq 5") = 0.00721$
 - Split on *height*? $Gain(Height \leq 6'0") = 0.0903$
 - Split on *age*? $Gain(Age \leq 30) = 0.590$
- Let's compute the **information gain** for each:

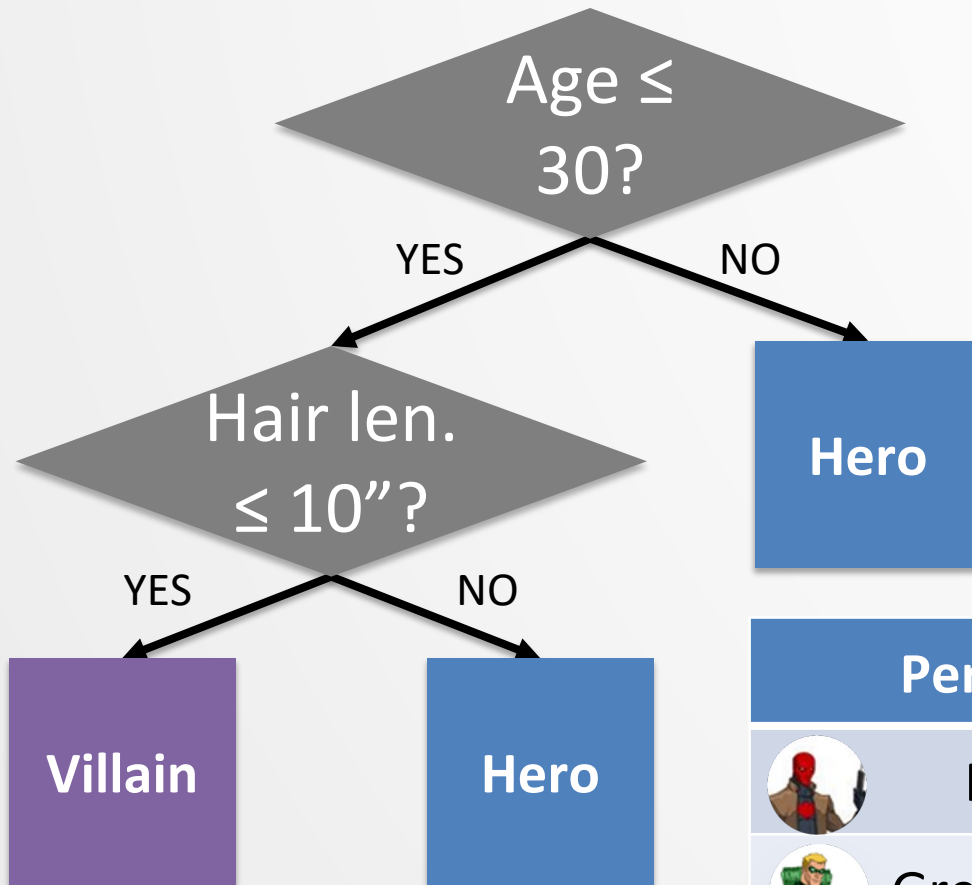
$$Gain(Q) = H(S) - \sum_{child\ set} p(child\ set)H(child\ set)$$

The resulting tree





- Splitting on *age* resulted in the greatest information gain.
- We're left with one heterogeneous set, so we recurse and find that *hair length* results in a complete classification of the training data.

Testing



- Inspired from Allan Neymark's (San Jose State University) Simpsons example.

- We just need to keep track of the attribute questions – not the training data.
- How are the following characters classified?

Person	Hair length	Height	Age
 Red Hood	2"	6'0"	22
 Green Arrow	1"	6'2"	38
 Bane	0"	5'8"	29