

DSC 240 Machine Learning

Machine Learning Basics / Spam Filter

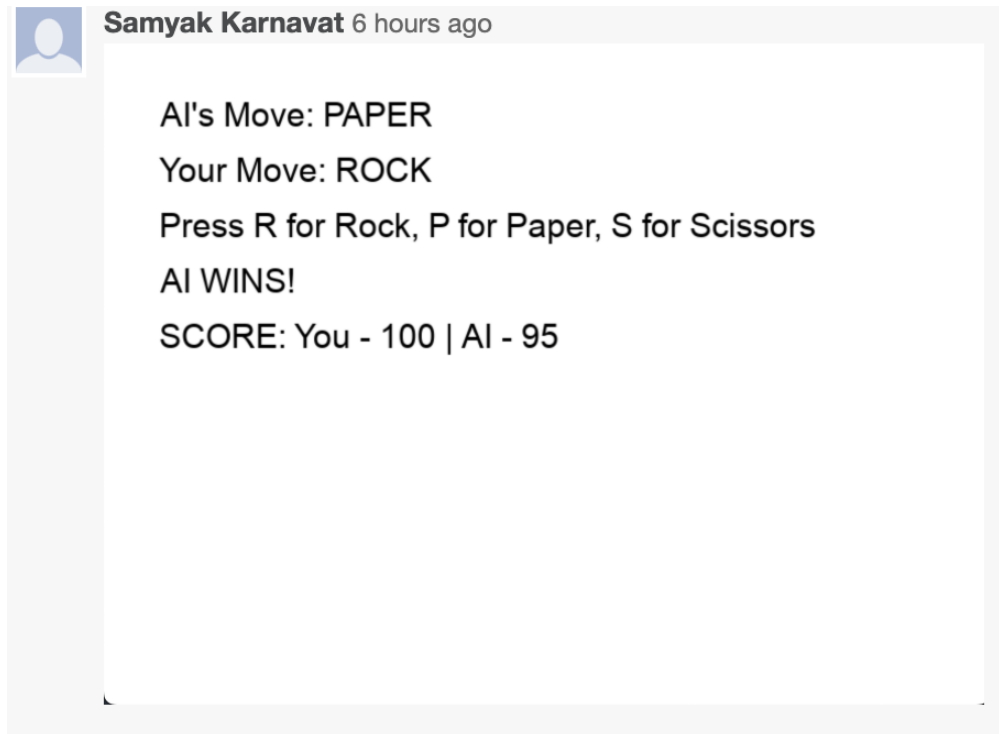
Jan 9, 2025

Instructor: Prof. Yu-Xiang Wang

Rock-Paper-Scissors game AI generated by ChatGPT from Lecture 1

- Facts:

- Generated by sending high-level instructions to ChatGPT
- May have been overlapping with codes on github and elsewhere.
- But worked out of the box.



Rock-Paper-Scissors game AI generated by ChatGPT from Lecture 1

- AI strategy:

```
# AI memory and prediction
memory = []
MAX_MEMORY = 3

def predict_move():
    if len(memory) < MAX_MEMORY:
        return random.choice([ROCK, PAPER, SCISSORS])

    # Predict based on recent patterns
    predicted = memory[-1]
    count = memory.count(predicted)
    if count > MAX_MEMORY / 2:
        # If user uses a move often, AI tries to beat that move
        if predicted == ROCK:
            return PAPER
        elif predicted == PAPER:
            return SCISSORS
        else:
            return ROCK
    return random.choice([ROCK, PAPER, SCISSORS])
```

- Why is it so hard to beat?

This is how the game is run. Anything Suspicious?

```

while running:
    screen.fill(WHITE)

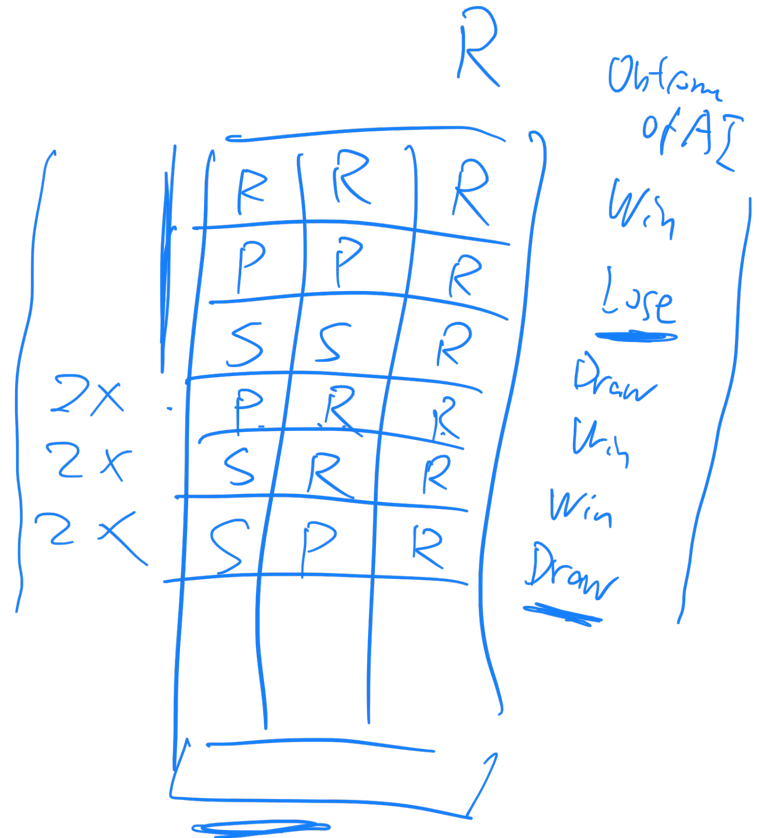
    for event in pygame.event.get():
        if event.type == pygame.QUIT:
            running = False

        if event.type == pygame.KEYDOWN:
            if event.key == pygame.K_r:
                player_move = ROCK
            elif event.key == pygame.K_p:
                player_move = PAPER
            elif event.key == pygame.K_s:
                player_move = SCISSORS

            if player_move is not None:
                memory.append(player_move)
                if len(memory) > MAX_MEMORY:
                    memory.pop(0)

            ai_move = predict_move()

```



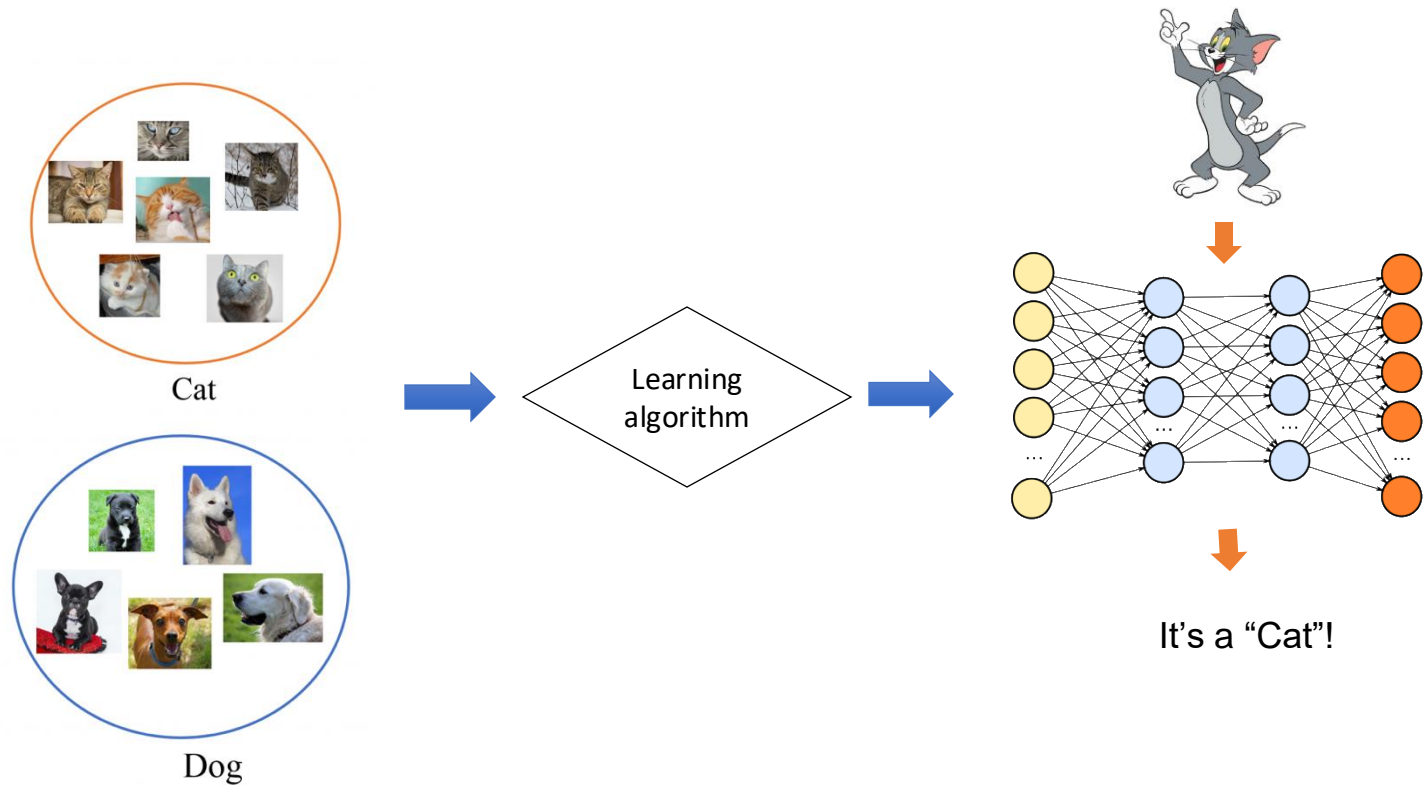
Moral of the story: Don't overly rely on ChatGPT!

- Subtle bugs --- hard to detect.
- Copyright issues --- did it copy from proprietary source?
- Such issues are 10x worse for ML systems.
- You don't learn anything.

Today

- Machine learning overview
- Supervised learning: Binary classification
- Feature design and feature extraction
- Family of classifiers: Decision Trees / Linear Separator
- Performance metric for a classifier

Recap: Machine learning studies “*computer programs that automatically improve (its performance on a **task**) with **experience**.*”



Discussion: How do we learn?

- Learning from ...
- Learning by ...
- What does it mean to have learned something?



Different tasks / problems in Machine Learning

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning
- Structured Prediction

Spam Filter.

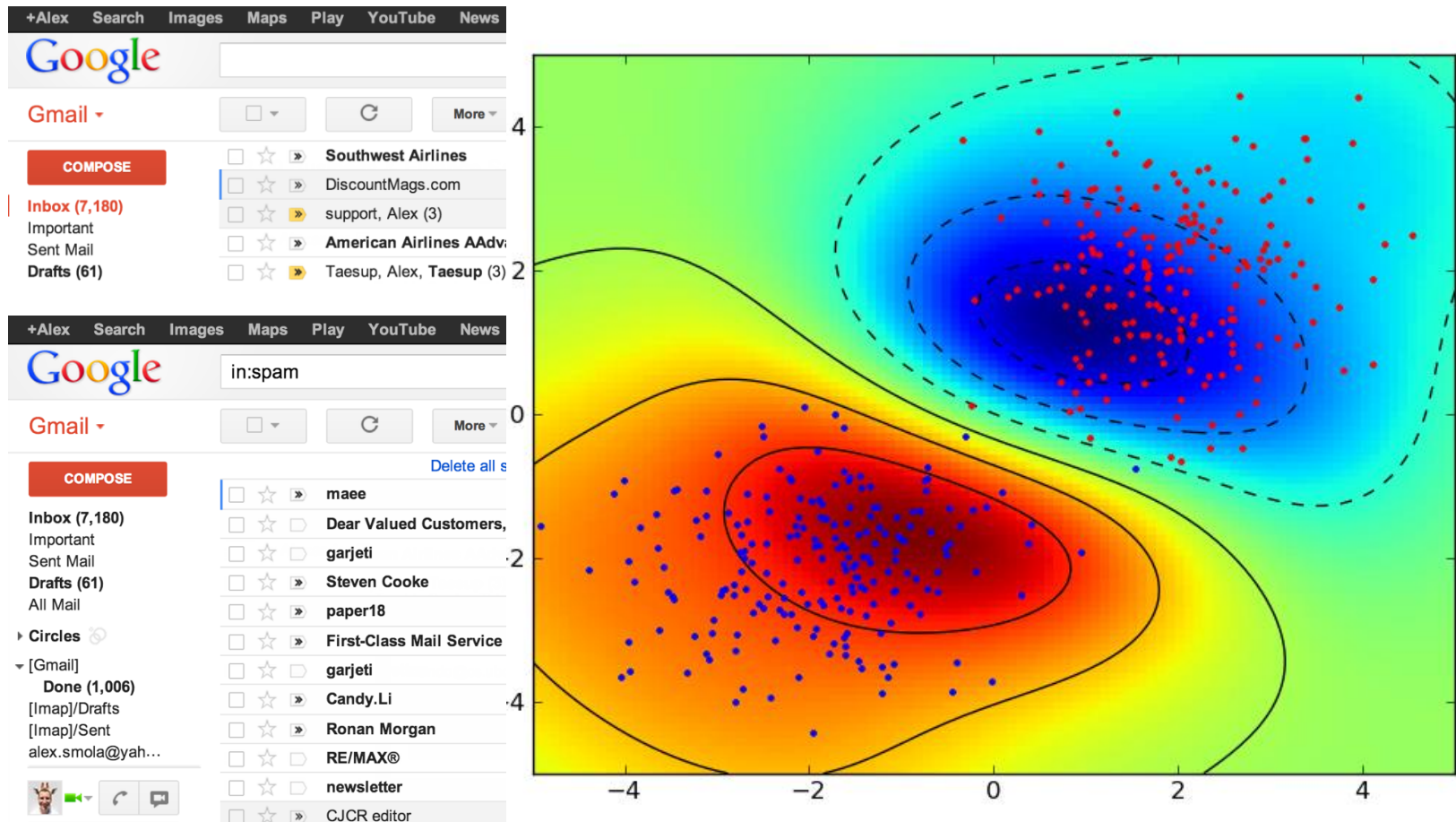
Topics of a text corpus

Atari Games. Serve Ads.

Machine translation.

Semi-supervised learning, active learning,
ranking /search / recommendation
self-supervised learning and many more!

Supervised learning is about predicting label y using feature x by learning from labeled examples.

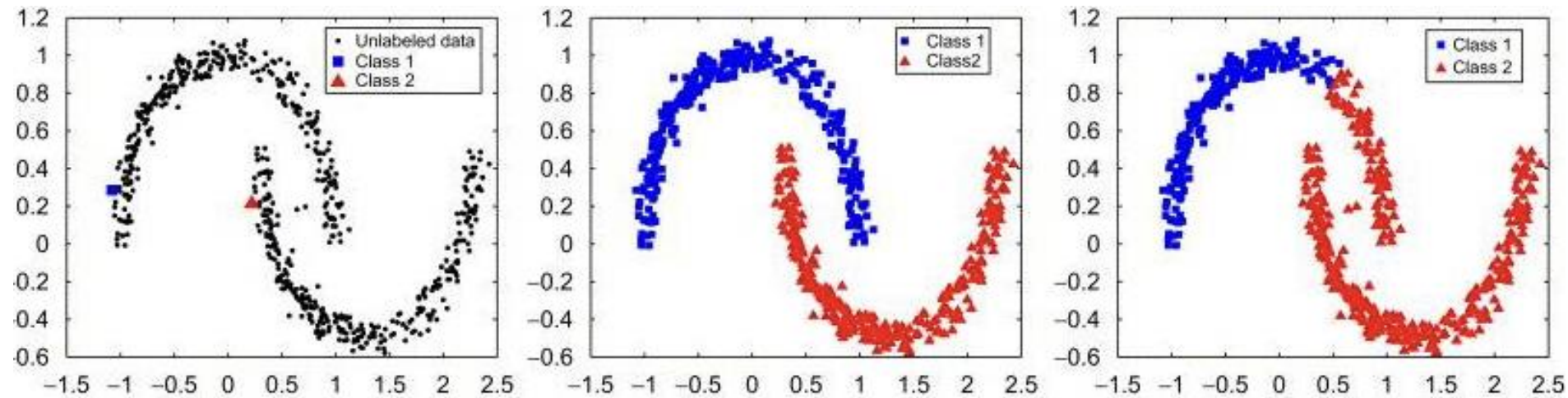


Unsupervised Learning is about finding structures in an unlabeled dataset.

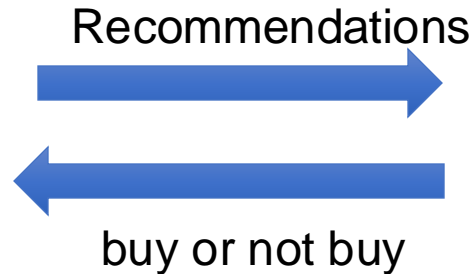
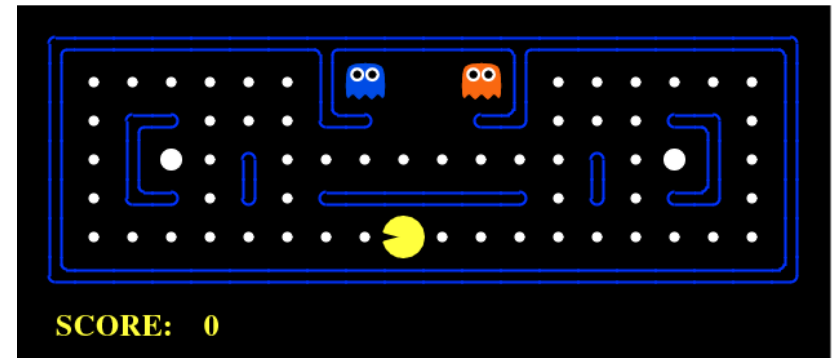
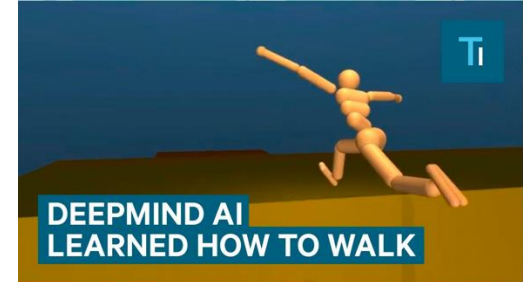
“Arts”	“Budgets”	“Children”	“Education”
NEW	MILLION	CHILDREN	SCHOOL
FILM	TAX	WOMEN	STUDENTS
SHOW	PROGRAM	PEOPLE	SCHOOLS
MUSIC	BUDGET	CHILD	EDUCATION
MOVIE	BILLION	YEARS	TEACHERS
PLAY	FEDERAL	FAMILIES	HIGH
MUSICAL	YEAR	WORK	PUBLIC
BEST	SPENDING	PARENTS	TEACHER
ACTOR	NEW	SAYS	BENNETT
FIRST	STATE	FAMILY	MANIGAT
YORK	PLAN	WELFARE	NAMPHY
OPERA	MONEY	MEN	STATE
THEATER	PROGRAMS	PERCENT	PRESIDENT
ACTRESS	GOVERNMENT	CARE	ELEMENTARY
LOVE	CONGRESS	LIFE	HAITI

The William Randolph Hearst Foundation will give \$1.25 million to Lincoln Center, Metropolitan Opera Co., New York Philharmonic and Juilliard School. “Our board felt that we had a real opportunity to make a mark on the future of the performing arts with these grants an act every bit as important as our traditional areas of support in health, medical research, education and the social services,” Hearst Foundation President Randolph A. Hearst said Monday in announcing the grants. Lincoln Center’s share will be \$200,000 for its new building, which will house young artists and provide new public facilities. The Metropolitan Opera Co. and New York Philharmonic will receive \$400,000 each. The Juilliard School, where music and the performing arts are taught, will get \$250,000. The Hearst Foundation, a leading supporter of the Lincoln Center Consolidated Corporate Fund, will make its usual annual \$100,000 donation, too.

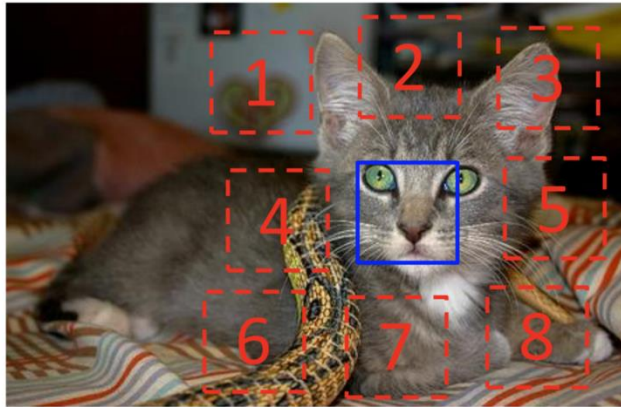
Semi-supervised Learning using both labeled and unlabeled data.



Reinforcement learning learns to make decisions for long-term rewards by trials-and-errors.

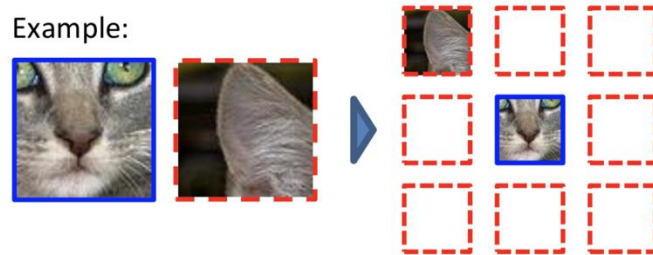


Self-supervised learning learns to predict parts of x using other parts of x .



$$X = (\text{cat face}, \text{cat ear}); Y = 3$$

Example:



Question 1:

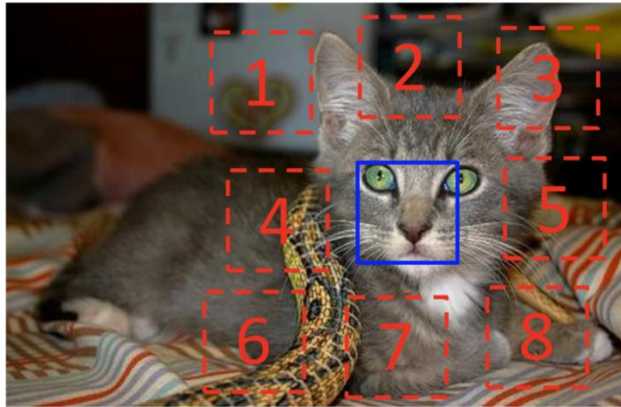


Question 2:

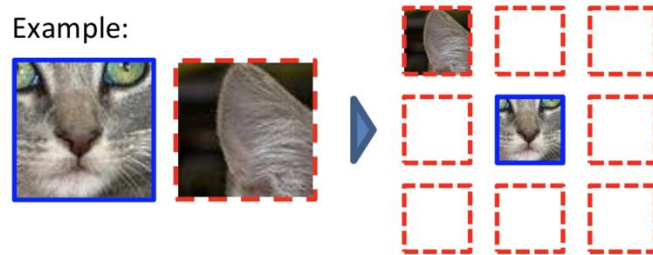


Image example from [\(Doersch et al, 2015\)](#), text example from [Amit Chaudhary](#)

Self-supervised learning learns to predict parts of x using other parts of x .



Example:



$$X = \left(\begin{array}{c} \text{cat face} \\ \text{cat ear} \end{array} \right); Y = 3$$

Question 1:



Question 2:



Randomly
masked

A quick [MASK] fox jumps over the [MASK] dog



Predict

A quick brown fox jumps over the lazy dog

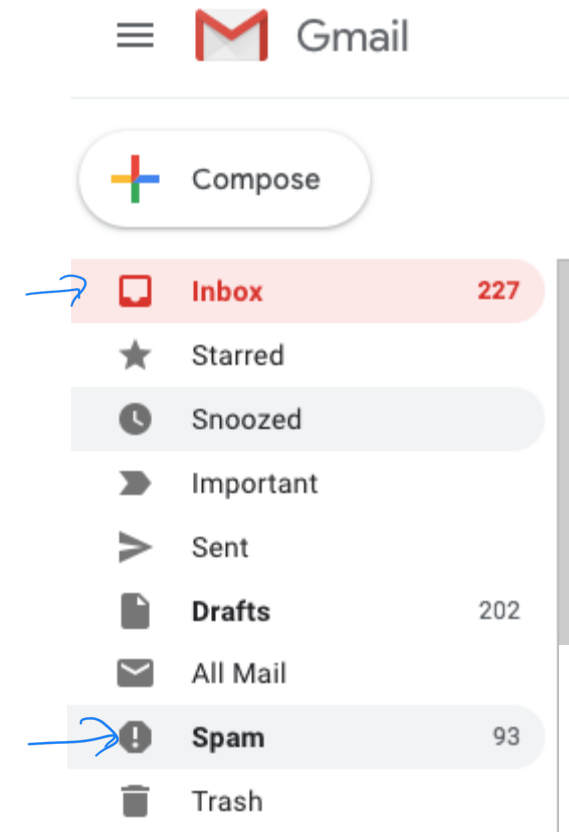
Image example from [\(Doersch et al, 2015\)](#), text example from [Amit Chaudhary](#)

The focus of today's lecture is "Supervised Learning"

- Actually, just "binary classification".

The focus of today's lecture is "Supervised Learning"

- Actually, just "binary classification".
- Prototypical Example: Spam filtering
 - Design an "agent" to look at my email

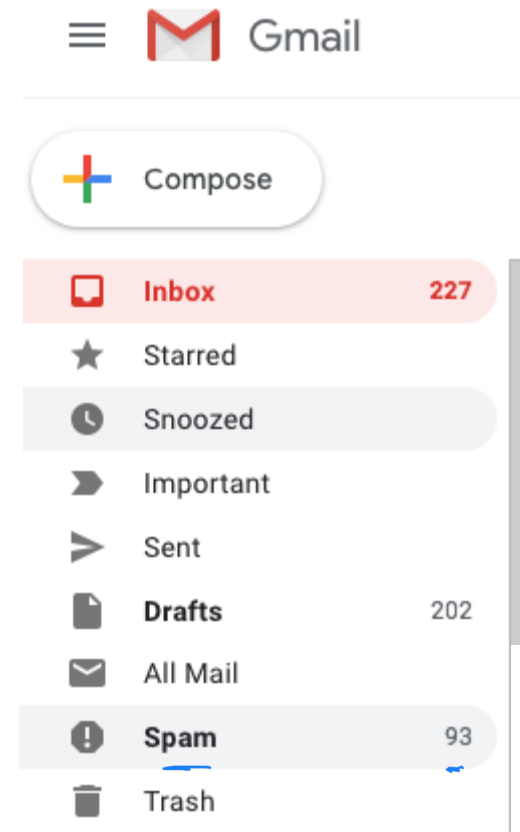


The focus of today's lecture is "Supervised Learning"

- Actually, just "binary classification".
- Prototypical Example: Spam filtering
 - Design an "agent" to look at my email
 - And predict whether it is "Spam" or "Ham"



Illustration extracted from [\[here\]](#)



Example of SPAM emails

Mail thinks this message is Junk Mail.

Move to Inbox

MICROWORLD CORPORATIO... December 20, 2019 at 2:38 AM

MC

CLAIMS.

To: undisclosed-recipients;;

Reply-To: microworld219@gmail.com

MICROWORLD CORPORATIONS:
 CUSTOMER SERVICE:
 FRIEDRICHSTRAË 10,BERLIN ALEMANHA
 REFERENCE NUMBER: MBB-009-D54-DE
 BATCH NUMBER: MGC-2019- SM-009
 TICKET NUMBERS: 2,6,13,21,26,32

OFFICIAL WINNING NOTIFICATION.

We are pleased to inform you of the released results of Microworld Promotion... This is a promotional program organized by Microworld Corporations, in conjunction with the Foundation for the promotion of software products, and use of email addresses. Held on Thursday 19th, December 2019, in Berlin, Alemanha. Your email address won a cash award of Four hundred and eighty eight thousand two hundred and fifty euros (488,250.00 Euros).. Contact Our Foreign Transfer Manager for claims with your winning details and your contact information. Mrs. Helena Bosch. Email: micropromo19@yahoo.com Congratulations!! Sincerely, Rosa Van Beek.

Mail thinks this message is Junk Mail.

Move to Inbox

Email ADMIN

January 1, 2020 at 10:35 PM

EA

cs.ucsb.edu APPLICATION -Storage Full Notes- Last -... [Details](#)

To: Yu-Xiang Wang,

Reply-To: Email ADMIN


Dear yuxiangw@cs.ucsb.edu,

Your email has used up the storage limit of 99.9 gigabytes as defined by your Administrator. You will be blocked from sending and receiving messages if not re-validated within **48hrs**. Kindly click on your email below for quick re-validation and additional storage will be updated automatically

yuxiangw@cs.ucsb.edu

Regards,
 E-mail Support 2020.

Example of another SPAM email

 Mail thinks this message is Junk Mail.

Move to Inbox

☆ **MARK ZUCKERBERG**

Junk - Google

August 24, 2018 at 10:48 AM

MZ

WINNING AMOUNT

Reply-To: MARK ZUCKERBERG

WINNING AMOUNT

My name is Mark Zuckerberg, A philanthropist the founder and CEO of the social-networking website Facebook, as well as one of the world's youngest billionaires and Chairman of the Mark Zuckerberg Charitable Foundation, One of the largest private foundations in the world. I believe strongly in 'giving while living' I had one idea that never changed in my mind - that you should use your wealth to help people and i have decided to secretly give {\$1,500,000.00} to randomly selected individuals worldwide. On receipt of this email, you should count yourself as the lucky individual. Your email address was chosen online while searching at random. Kindly get back to me at your earliest convenience, so I know your email address is valid. (mzuckerberg2444@gmail.com) Email me Visit the web page to know more about me: https://en.wikipedia.org/wiki/Mark_Zuckerberg/ or you can google me (Mark Zuckerberg)

Regards,
MARK ZUCKERBERG

Example of a HAM (non-spam) email



Dear Professor Foo,

I am a student in your machine learning class.

I have a question about the second term project and I was not able to find the answer on the syllabus. Should our project be only about the topics listed on the second part of the syllabus, or can I incorporate topics from the whole course, as long as it fits with the subject of the class?

I look forward to hearing from you.

Best regards,

Bar

Quoted from [[Here](#)].

[3 min discussion] What are the features that we can use to describe an email?

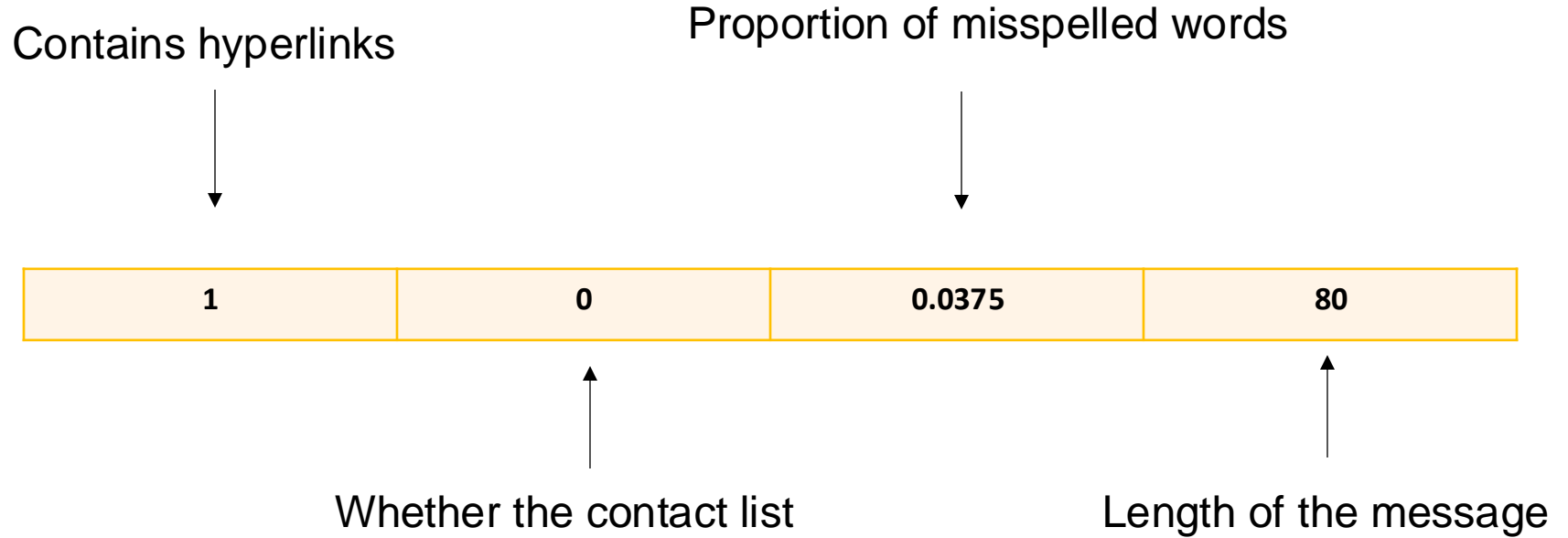
- What are characteristics of spam and ham emails?
- What are the information that we can extract from text, and hyper-texts to describe an email?
- What are typical characteristic of a spam email?

Possible features

- Number of special characters: \$, %
- Mentioning of: Award, cash, free
- Greetings: generic, or specific
- Bad grammars and misspelled words: e.g. m0ney, c1ick here.
- Excessive excitement: Many “!”, “!!!”, “?!”, words in CAPITAL LETTERS.

- Whether the senders on the contact list
- Length of an email
- Whether the receiver has responded to sender before

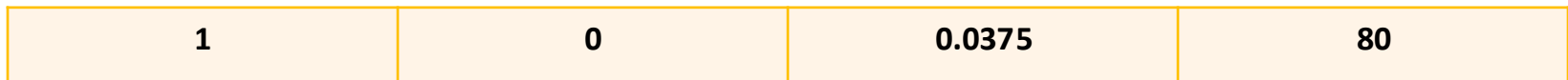
Example of a feature vector of dimension 4



Example of a feature vector of dimension 4

Contains hyperlinks

Proportion of misspelled words



Whether the contact list

Length of the message

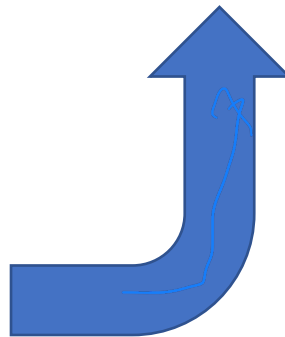
Email ADMIN January 1, 2020 at 10:35 PM EA
cs.ucsb.edu APPLICATION -Storage Full Notes- Last -... [Details](#)
To: Yu-Xiang Wang,
Reply-To: Email ADMIN

Dear yuxiangw@cs.ucsb.edu,

Your email has used up the storage limit of 99.9 gigabytes as defined by your Administrator. You will be blocked from sending and receiving messages if not re-validated within **48hrs**. Kindly click on your email below for quick re-validation and additional storage will be updated automatically

yuxiangw@cs.ucsb.edu

Regards,
E-mail Support 2020.

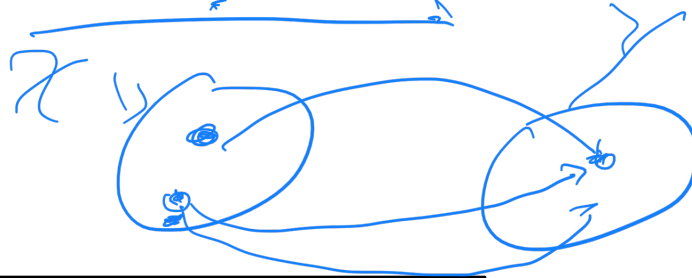


Step 1 in Modelling Feature extractor:
Converting the object of interest to a vector of numerical values.

$x \in \mathbb{R}^d$

Mathematically defining a classifier

- Feature space: $\mathcal{X} = \mathbb{R}^d$ $d: \text{dimension } d=4$
 $\underbrace{\{0,1\} \times \{0,1\} \times [0,1] \times \mathcal{N}}$
 $x \leftarrow \phi(\text{email})$
- Label space: $\mathcal{Y} = \{0, 1\} = \{\text{non-spam}, \text{spam}\}$
 $y = \{0, 1\}$
- A classifier (hypothesis): $h : \mathcal{X} \rightarrow \mathcal{Y}$



Math notation for "function definition", e.g., function "add"

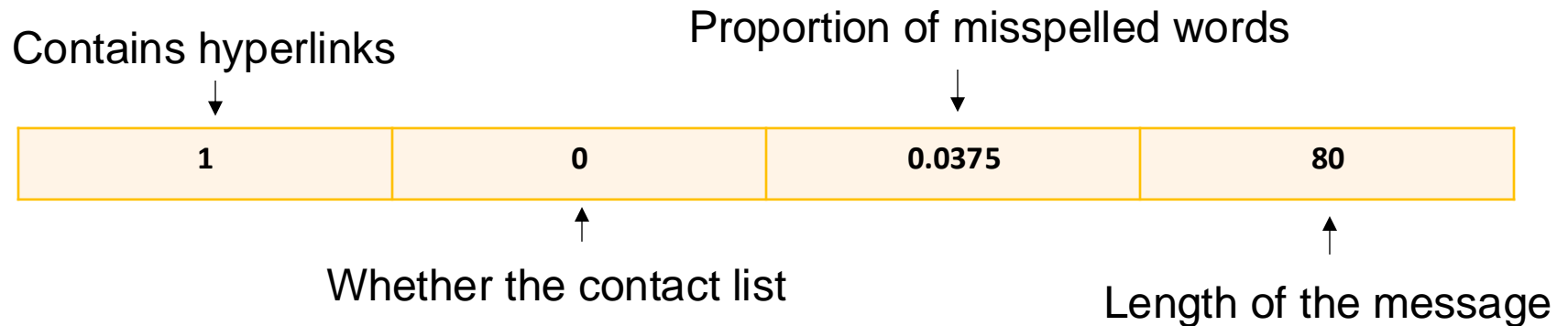
- What do you write in c++?

```
int add (int a, int b) {};
```

- What do you write in python?

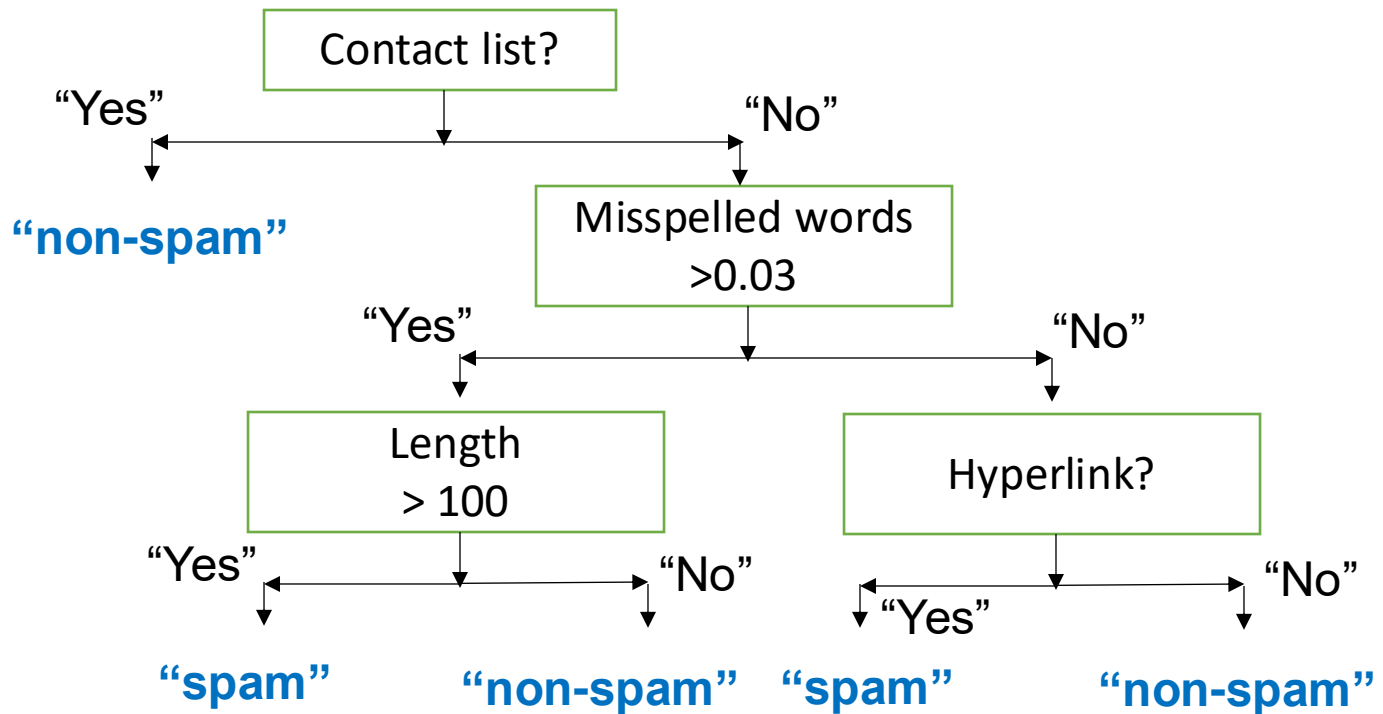
```
def add (a: int, b: int) --> int: {}
```

How do we make use of this feature vector?
 What is a reasonable “classifier” based on this
 feature representation?

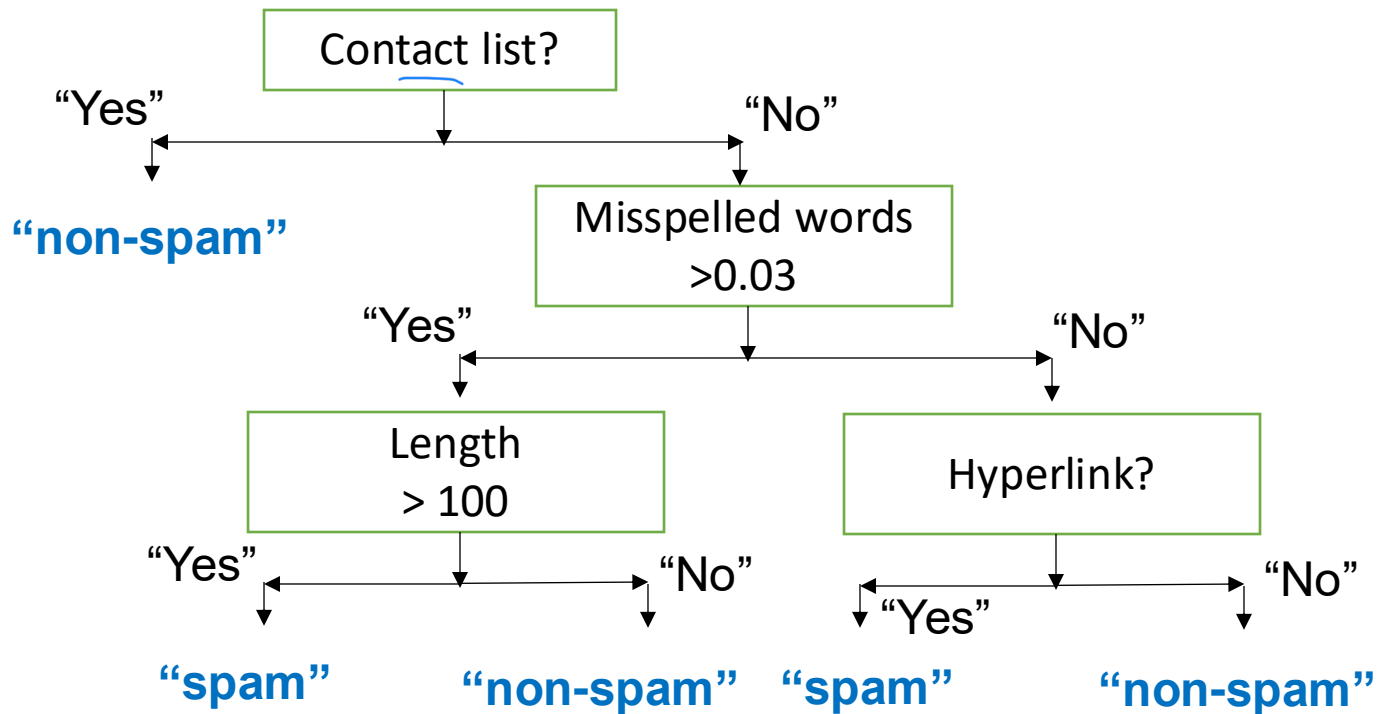


- Feature space: $\{0, 1\} \times \{0, 1\} \times \mathbb{R} \times \mathbb{N}$
- Label space: $\mathcal{Y} = \{0, 1\} = \{\text{non-spam}, \text{spam}\}$
- **How are we going to use these features as a human?**
 - (3 min discussion)

Decision trees

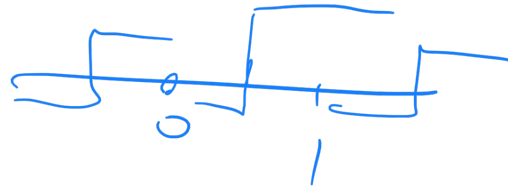


Decision trees



- **Question:** How is each decision tree determined? What are its parameters?

How is a decision tree specified?



Parameters:

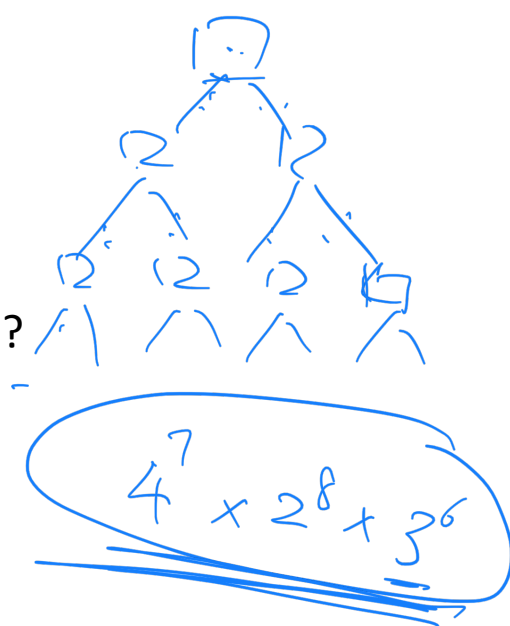
- Which feature(s) to use when branching?
- How to branch? Thresholding? Where to put the threshold?
- Which label to assign at leaf nodes?

Hyperparameters:

- Max height of a decision tree?
- Number of features the tree can use in each branch?

Question: Consider a problem with **4 binary features**.

- How many decision trees of **3 layers** are there? If each decision uses only one feature? (you may repeat features)
- How many possible feature vectors are there?



$$\{0,1\}^4$$

$$2^4$$

distinct classifiers

$$2^{2^4} = 2^{16}$$

$$65536$$

Example: Linear classifiers



Example: Linear classifiers

- $\text{Score}(x) = w_0 + w_1 * 1(\text{hyperlinks}) + w_2 * 1(\text{contact list}) + w_3 * \text{misspelling} + w_4 * \text{length}$

Example: Linear classifiers

- $\text{Score}(x) = w_0 + w_1 * 1(\text{hyperlinks}) + w_2 * 1(\text{contact list}) + w_3 * \text{misspelling} + w_4 * \text{length}$
- A linear classifier: $h(x) = 1$ if $\text{Score}(x) > 0$ and 0 otherwise.

Example: Linear classifiers

$$W = \begin{bmatrix} w_0 \\ w_1 \\ w_2 \\ w_3 \\ w_4 \end{bmatrix} \quad \vec{x} = \begin{bmatrix} 1 \\ x \end{bmatrix}$$

- $\text{Score}(x) = w_0 + w_1 * 1(\text{hyperlinks}) + w_2 * 1(\text{contact list}) + w_3 * \text{misspelling} + w_4 * \text{length}$

$$\text{score}(x) = W^T \vec{x} = \langle W, \vec{x} \rangle$$

↑
inner product

- A linear classifier: $h(x) = 1$ if $\text{Score}(x) > 0$ and 0 otherwise.

- Question: What are the parameters in a linear classifier?

$$W \in \mathbb{R}^{\text{def}}$$

weights

$$W \in \mathbb{R}^5$$

Example: Linear classifiers

- $\text{Score}(x) = w_0 + w_1 * 1(\text{hyperlinks}) + w_2 * 1(\text{contact list}) + w_3 * \text{misspelling} + w_4 * \text{length}$
- A linear classifier: $h(x) = 1$ if $\text{Score}(x) > 0$ and 0 otherwise.
- Question: What are the **parameters** in a linear classifier?
 - If we redefine $\mathcal{Y} = \{-1, 1\}$

Example: Linear classifiers

- $\text{Score}(x) = w_0 + w_1 * 1(\text{hyperlinks}) + w_2 * 1(\text{contact list}) + w_3 * \text{misspelling} + w_4 * \text{length}$
- A linear classifier: $h(x) = 1$ if $\text{Score}(x) > 0$ and 0 otherwise.
- Question: What are the **parameters** in a linear classifier?
 - If we redefine $\mathcal{Y} = \{-1, 1\}$
 - A compact representation:

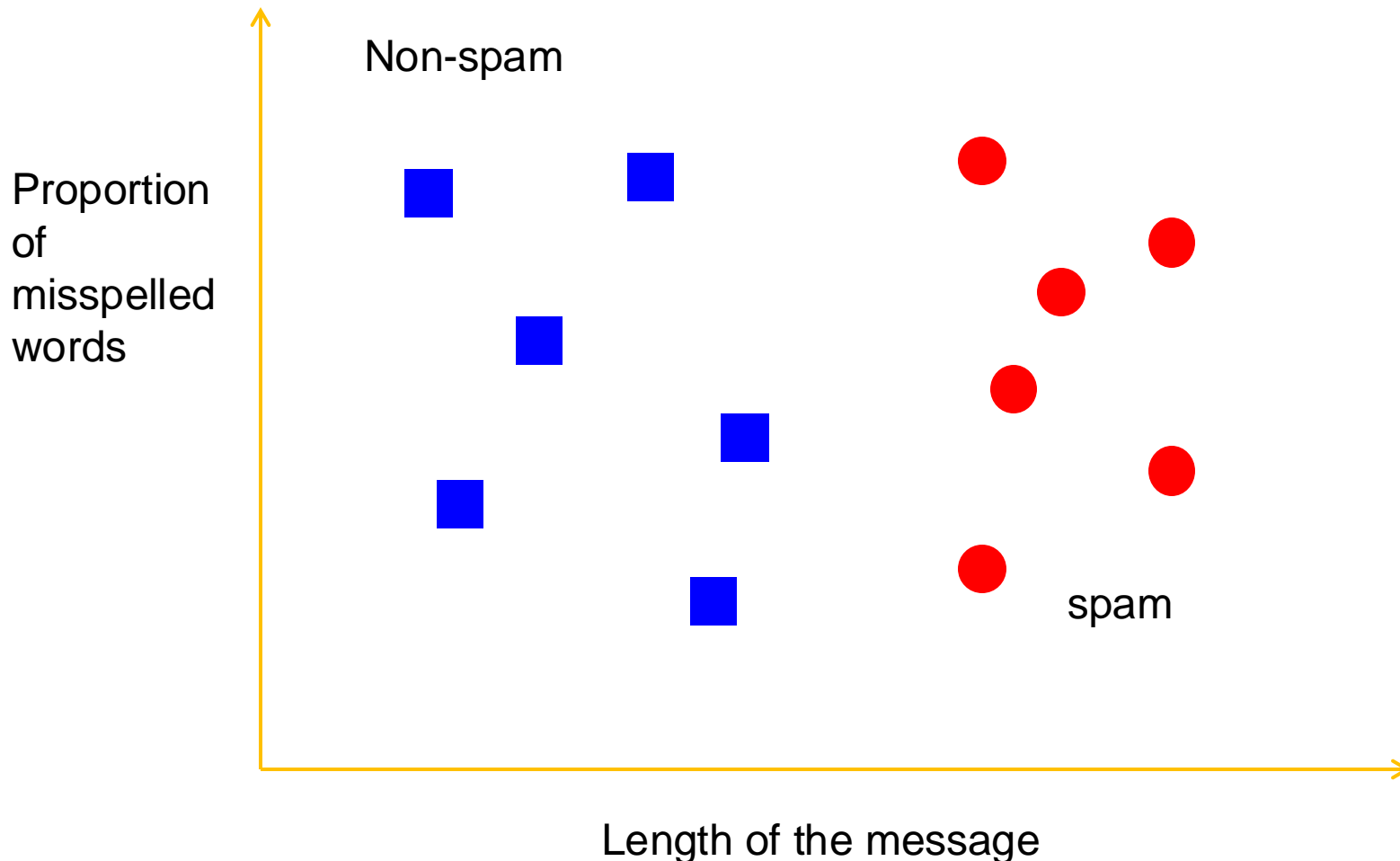
$$h(x) = \text{sign}(w^T [1; x])$$

Geometric view: Linear classifier are “half-spaces”!

$\{x \mid w_0 + w_1 * x_1 + w_2 * x_2 + w_3 * x_3 + w_4 * x_4 > 0\}$
The set of all “emails” that will be classified as “Spams”.

Geometric view: Linear classifier are “half-spaces”!

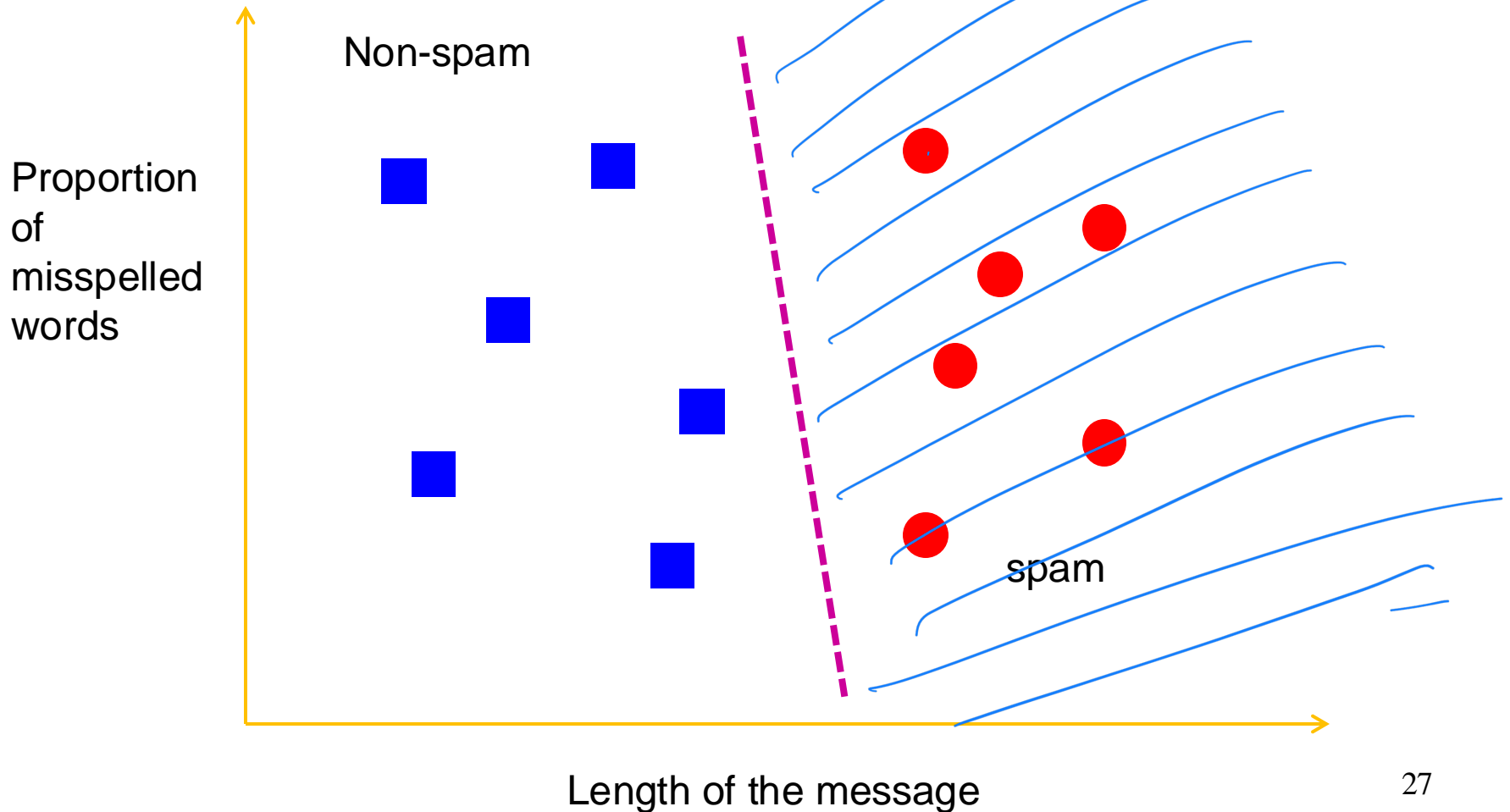
$\{x \mid w_0 + w_1 * x_1 + w_2 * x_2 + w_3 * x_3 + w_4 * x_4 > 0\}$
The set of all “emails” that will be classified as “Spams”.



Geometric view: Linear classifier are “half-spaces”!

$$\{x \mid w_0 + w_1 * x_1 + w_2 * x_2 + w_3 * x_3 + w_4 * x_4 > 0\}$$

The set of all “emails” that will be classified as “Spams”.



Specifying a family of classifiers --- a
“hypothesis class”

Specifying a family of classifiers --- a “hypothesis class”

- Hypothesis class

Specifying a family of classifiers --- a “hypothesis class”

- Hypothesis class
 - A family of classifiers: \mathcal{H}

Specifying a family of classifiers --- a “hypothesis class”

- Hypothesis class
 - A family of classifiers: \mathcal{H}
 - Also known as “concept classes”, “models”, “decision rule book”

Specifying a family of classifiers --- a “hypothesis class”

- Hypothesis class

- A family of classifiers: \mathcal{H}
- Also known as “concept classes”, “models”, “decision rule book”
- “Neural networks” and “Support Vector Machines” are hypothesis classes.

Specifying a family of classifiers --- a “hypothesis class”

- Hypothesis class

- A family of classifiers: \mathcal{H}
- Also known as “concept classes”, “models”, “decision rule book”
- “Neural networks” and “Support Vector Machines” are hypothesis classes.
- Typically we want this family to be large and flexible.

Specifying a family of classifiers --- a “hypothesis class”

- Hypothesis class

- A family of classifiers: \mathcal{H}
- Also known as “concept classes”, “models”, “decision rule book”
- “Neural networks” and “Support Vector Machines” are hypothesis classes.
- Typically we want this family to be large and flexible.

- The task of machine learning:

Specifying a family of classifiers --- a “hypothesis class”

- Hypothesis class

- A family of classifiers: \mathcal{H}
- Also known as “concept classes”, “models”, “decision rule book”
- “Neural networks” and “Support Vector Machines” are hypothesis classes.
- Typically we want this family to be large and flexible.

- The task of machine learning:

- A **selection problem** to find a

$$\underline{h} \in \underline{\mathcal{H}}$$

Specifying a family of classifiers --- a “hypothesis class”

$$\mathcal{H} = \left\{ \underline{\text{sig}(w^T \cdot x)} \mid \underline{w \in \mathbb{R}^d} \right\}$$

• Hypothesis class

- A family of classifiers: \mathcal{H}
- Also known as “concept classes”, “models”, “decision rule book”
- “Neural networks” and “Support Vector Machines” are hypothesis classes.
- Typically we want this family to be large and flexible.

• The task of machine learning:

- A selection problem to find a

$$h \in \mathcal{H}$$

that “works well” on this problem.

We will use the following notation to denote a classifier (hypothesis) specified by a specific parameter choice w

$$\underline{h_w} : \mathcal{X} \rightarrow \mathcal{Y}$$

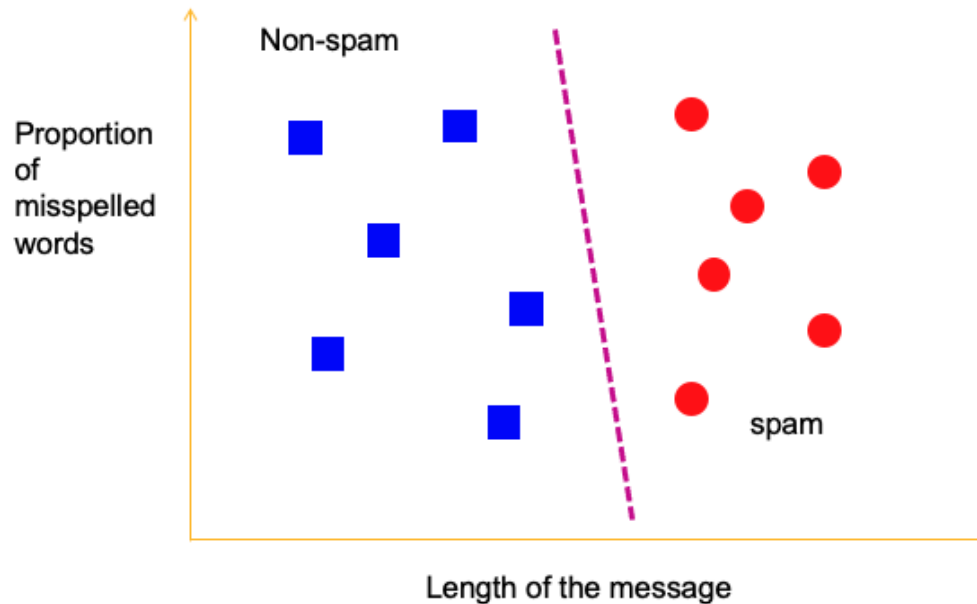
- For any $x \in \mathcal{X}$
 - We can apply this classifier to get its predicted label

$$\underline{\hat{y}} = \underline{h_w}(x)$$

- The prediction doesn't have to be correct. It just need to be valid, i.e.,

$$\hat{y} \in \underline{\mathcal{Y}}$$

Learning linear classifiers

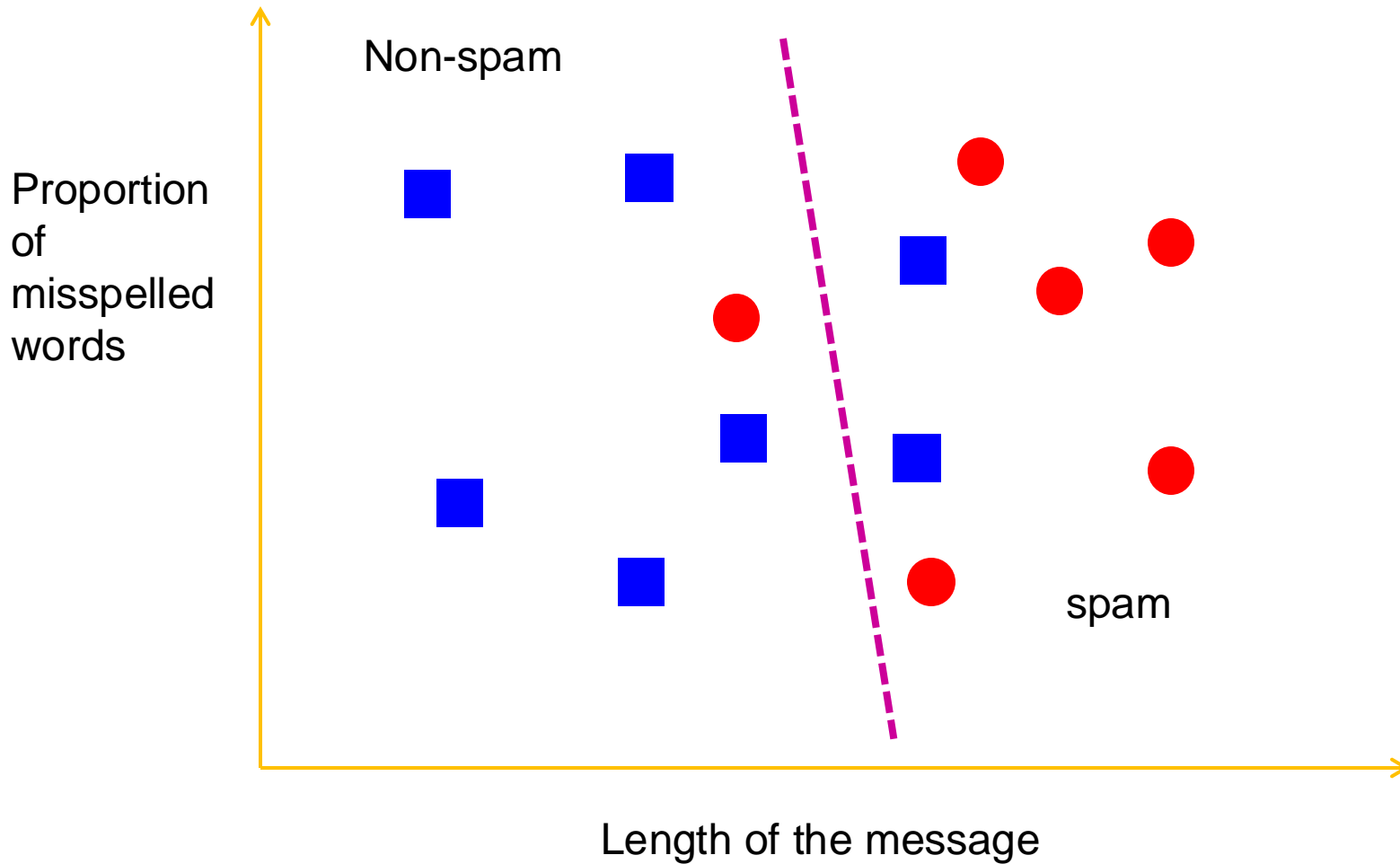


- Training data:

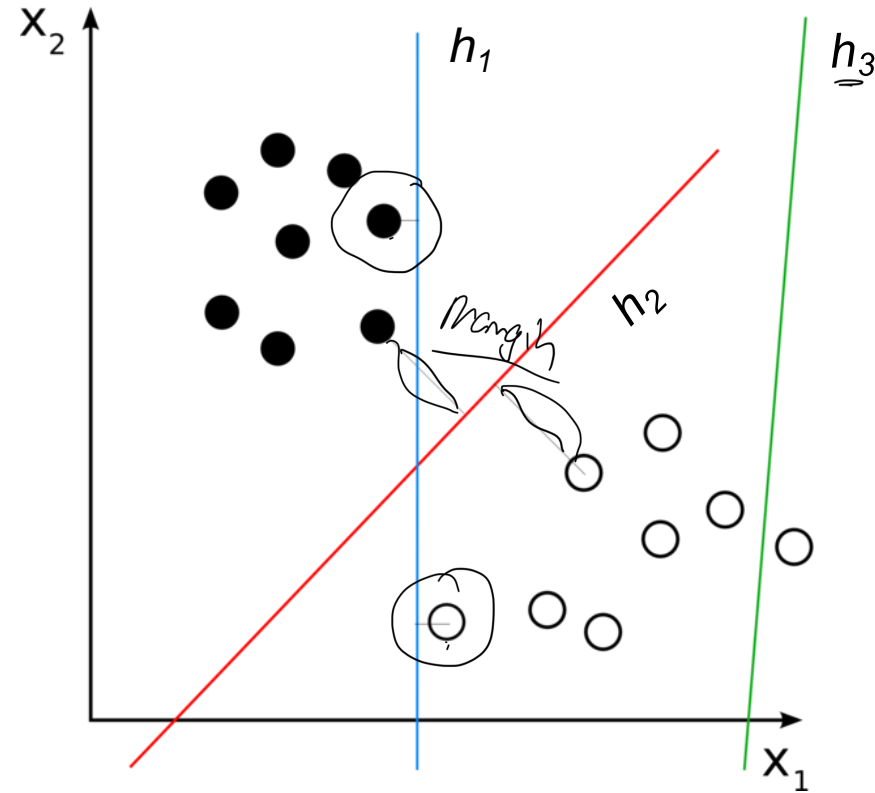
$$(x_1, y_1), \dots, (x_n, y_n) \in \mathcal{X} \times \mathcal{Y}$$

- In the above example, there is a clean cut boundary that distinguishes “spams” from “non-spams”.
 - “Linearly separable” problem
 - Learning linear classifier: Finding vector w , such that the predictions of h_w is **consistent with the observed training data**.

Example: Linearly non-separable cases



[3 min discussion] How can we evaluate a classifier (a spam filter)?



Which is better, h_1 , h_2 , h_3 ? Why?

Confusion matrix for binary classification

We can summarize **performance** of a model on a binary classification task with a **contingency table** known as a **confusion matrix**

		Actual class y		
		1	0	
Predicted class \hat{y}	1	TP Type I Error	FP Type I Error	<i>Estimated positive \hat{P}</i>
	0	FN Type II Error	TN	<i>Estimated negative \hat{N}</i>
		<i>Positives P</i>	<i>Negatives N</i>	TOTAL

Confusion matrix for binary classification

We can summarize **performance** of a model on a binary classification task with a **contingency table** known as a **confusion matrix**

		Actual class y		
		1	0	
Predicted class \hat{y}	1	TP <small>Type I Error</small>	FP <small>Type I Error</small>	<i>Estimated positive \hat{P}</i>
	0	FN <small>Type II Error</small>	TN	<i>Estimated negative \hat{N}</i>
		<i>Positives P</i>	<i>Negatives N</i>	TOTAL

Confusion matrix for binary classification

We can summarize **performance** of a model on a binary classification task with a **contingency table** known as a **confusion matrix**

		Actual class y		
		1	0	
Classifier output Predicted class \hat{y}	1	TP Type I Error	FP Type I Error	<i>Estimated positive \hat{P}</i>
	0	FN Type II Error	TN	<i>Estimated negative \hat{N}</i>
		<i>Positives P</i>	<i>Negatives N</i>	TOTAL

Confusion matrix for binary classification

We can summarize **performance** of a model on a binary classification task with a **contingency table** known as a **confusion matrix**

		Actual class y		
		1	0	
Classifier output ↓ Predicted class \hat{y}	1	TP	FP <small>Type I Error</small>	<i>Estimated positive \hat{P}</i>
	0	FN <small>Type II Error</small>	TN	<i>Estimated negative \hat{N}</i>
		<i>Positives P</i>	<i>Negatives N</i>	TOTAL

$$TP + FN = P$$

$$FP + TN = N$$

$$TP + FP = \hat{P}$$

$$FN + TN = \hat{N}$$

$$P + N = TOTAL$$

$$\hat{P} + \hat{N} = TOTAL$$

TP – true positives

FP – false positives

TN – true negatives

FN – false negatives

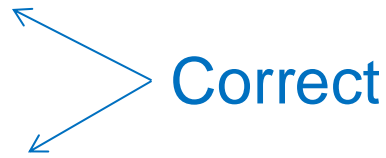
		Actual class y		
		1	0	
Predicted class \hat{y}	1	TP	FP	<i>Estimated positive \hat{P}</i>
	0	FN	TN	<i>Estimated negative \hat{N}</i>
		<i>Positives P</i>	<i>Negatives N</i>	TOTAL

TP – true positives

FP – false positives

TN – true negatives

FN – false negatives



		Actual class y		
		1	0	
Predicted class \hat{y}	1	TP	FP	<i>Estimated positive \hat{P}</i>
	0	FN	TN	<i>Estimated negative \hat{N}</i>
		<i>Positives P</i>	<i>Negatives N</i>	TOTAL

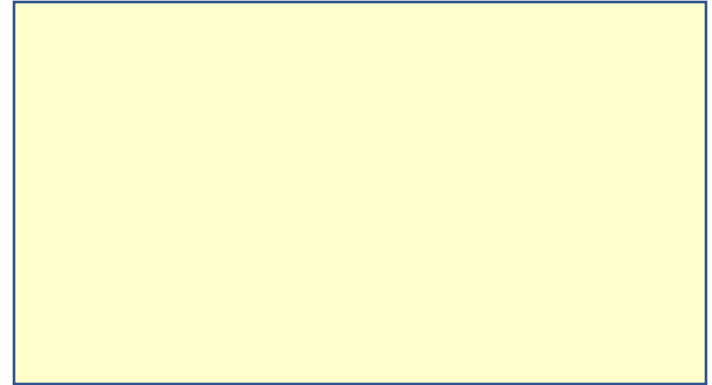
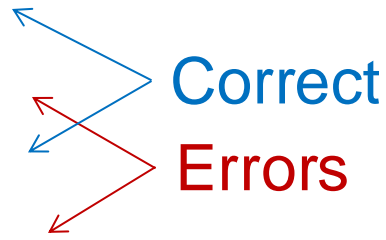
TP – true positives
 FP – false positives
 TN – true negatives
 FN – false negatives

Correct
 Errors

		Actual class y		
		1	0	
Predicted class \hat{y}	1	TP	FP	<i>Estimated positive \hat{P}</i>
	0	FN	TN	<i>Estimated negative \hat{N}</i>
		<i>Positives P</i>	<i>Negatives N</i>	TOTAL

Instance space (all emails)

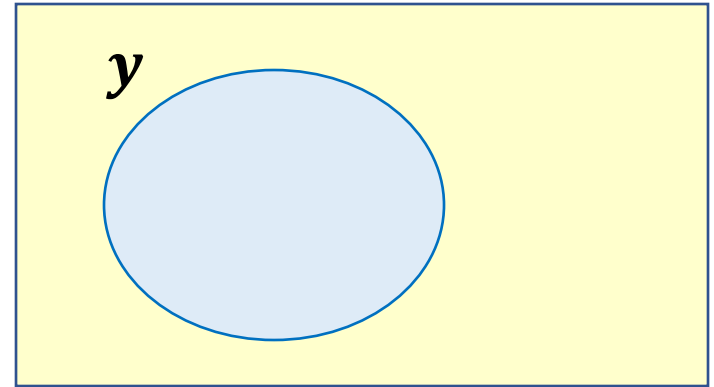
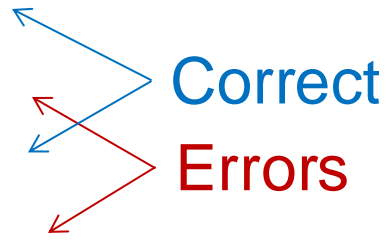
- TP – true positives
- FP – false positives
- TN – true negatives
- FN – false negatives



		Actual class y		
		1	0	
Predicted class \hat{y}	1	TP	FP	<i>Estimated positive \hat{P}</i>
	0	FN	TN	<i>Estimated negative \hat{N}</i>
		<i>Positives P</i>	<i>Negatives N</i>	TOTAL

Instance space (all emails)

- TP – true positives
- FP – false positives
- TN – true negatives
- FN – false negatives



Actual class
 y

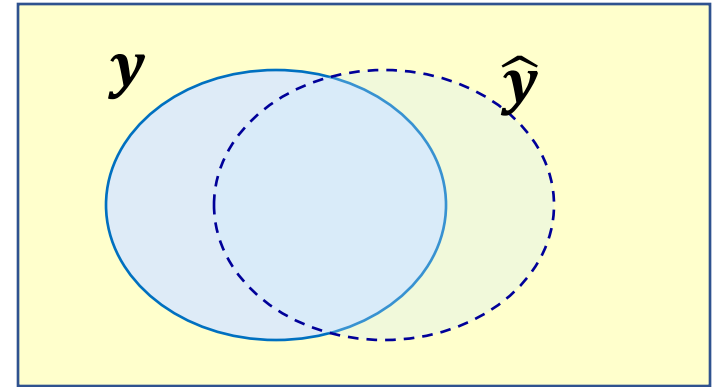
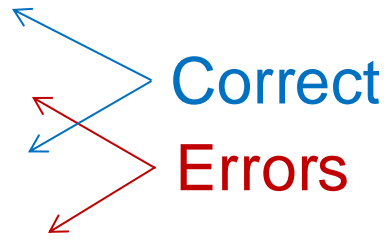
Predicted class

\hat{y}

		Actual class y		
		1	0	
Predicted class \hat{y}	1	TP	FP	<i>Estimated positive \hat{P}</i>
	0	FN	TN	<i>Estimated negative \hat{N}</i>
		<i>Positives P</i>	<i>Negatives N</i>	TOTAL

Instance space (all emails)

- TP – true positives
- FP – false positives
- TN – true negatives
- FN – false negatives



Actual class
 y

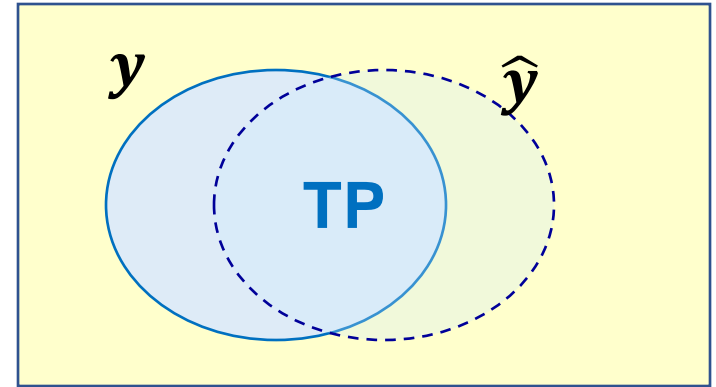
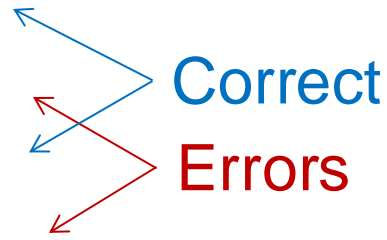
Predicted class

\hat{y}

	1	0	
1	TP	FP	<i>Estimated positive \hat{P}</i>
0	FN	TN	<i>Estimated negative \hat{N}</i>
	<i>Positives P</i>	<i>Negatives N</i>	TOTAL

Instance space (all emails)

- TP – true positives
- FP – false positives
- TN – true negatives
- FN – false negatives



Actual class
 y

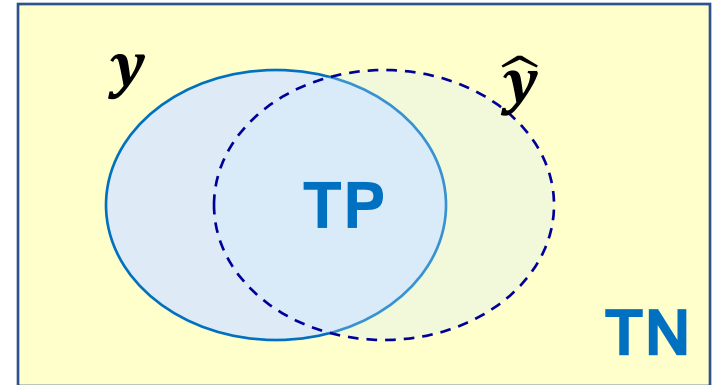
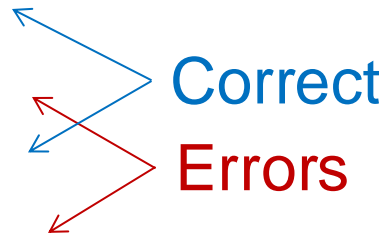
Predicted class

\hat{y}

	1	0	
1	TP	FP	<i>Estimated positive \hat{P}</i>
0	FN	TN	<i>Estimated negative \hat{N}</i>
	<i>Positives P</i>	<i>Negatives N</i>	TOTAL

Instance space (all emails)

- TP – true positives
- FP – false positives
- TN – true negatives
- FN – false negatives



Actual class
 y

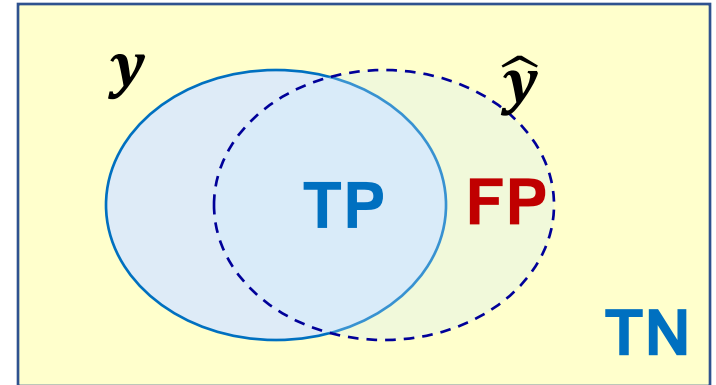
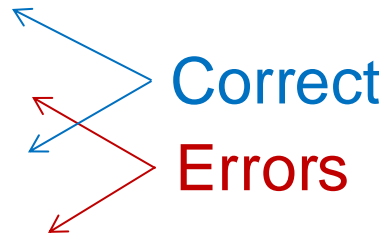
Predicted class

\hat{y}

	1	0	
1	TP	FP	<i>Estimated positive \hat{P}</i>
0	FN	TN	<i>Estimated negative \hat{N}</i>
	<i>Positives P</i>	<i>Negatives N</i>	TOTAL

Instance space (all emails)

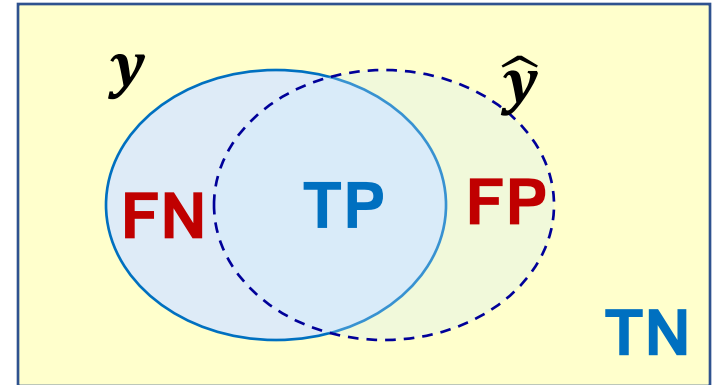
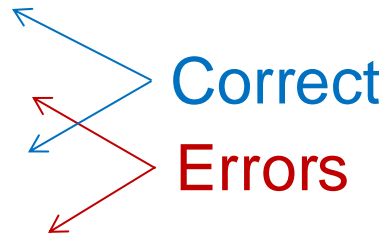
- TP – true positives
- FP – false positives
- TN – true negatives
- FN – false negatives



		Actual class y		
		1	0	
Predicted class \hat{y}	1	TP	FP	<i>Estimated positive \hat{P}</i>
	0	FN	TN	<i>Estimated negative \hat{N}</i>
		<i>Positives P</i>	<i>Negatives N</i>	TOTAL

Instance space (all emails)

- TP – true positives
- FP – false positives
- TN – true negatives
- FN – false negatives



		Actual class y		
		1	0	
Predicted class \hat{y}	1	TP	FP	<i>Estimated positive \hat{P}</i>
	0	FN	TN	<i>Estimated negative \hat{N}</i>
		<i>Positives P</i>	<i>Negatives N</i>	TOTAL

Key terminology

False positive rate (FPR) = $\frac{FP}{N} = \alpha$

Accuracy = $\frac{TP+TN}{P+N} = \left(\frac{P}{P+N}\right) TPR + \left(\frac{N}{P+N}\right) TNR$

False negative (miss) rate (FNR) = $\frac{FN}{P} = \beta$

Error rate = $\frac{FP+FN}{P+N}$

True positive rate (TPR) = $\frac{TP}{P}$ = Sensitivity = Recall = $1 - \beta$

Precision = $\frac{TP}{\hat{P}}$

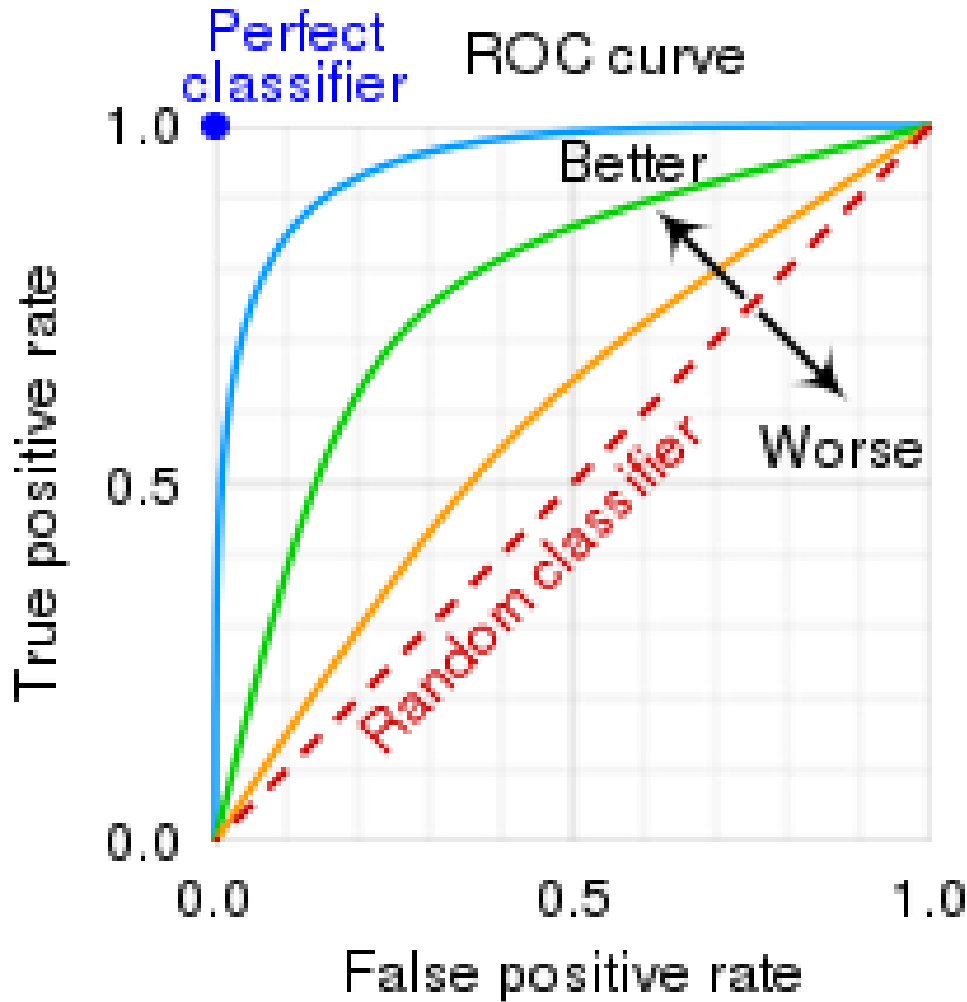
True negative rate (TNR) = $\frac{TN}{N}$ = Specificity = $1 - \alpha$

F1 score = $\frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}} = \frac{2 \cdot TP}{P + \hat{P}}$

Predicted class
 \hat{y}

		Actual class y		
		1	0	
1	1	TP	FP	<i>Estimated positive \hat{P}</i>
	0	FN	TN	
		<i>Positives P</i>	<i>Negative N</i>	TOTAL

Response Operator Characteristic (ROC) curve



Single number summary of any
“score function”

AUC: **A**rea **U**nder the ROC **C**urve

What are there good ranges of AUC values?

(illustration from wikipedia)

Checkpoint: Performance metrics

(a) Accuracy

(b) Precision

(c) Recall

(d) F1 score

(e) Area under the ROC

If “error” is the metric of interest, then how to learn linear classifier in a non-linearly separable case?

- Training data:

$$(x_1, y_1), \dots, (x_n, y_n) \in \mathcal{X} \times \mathcal{Y}$$

- Solving the following optimization problem:

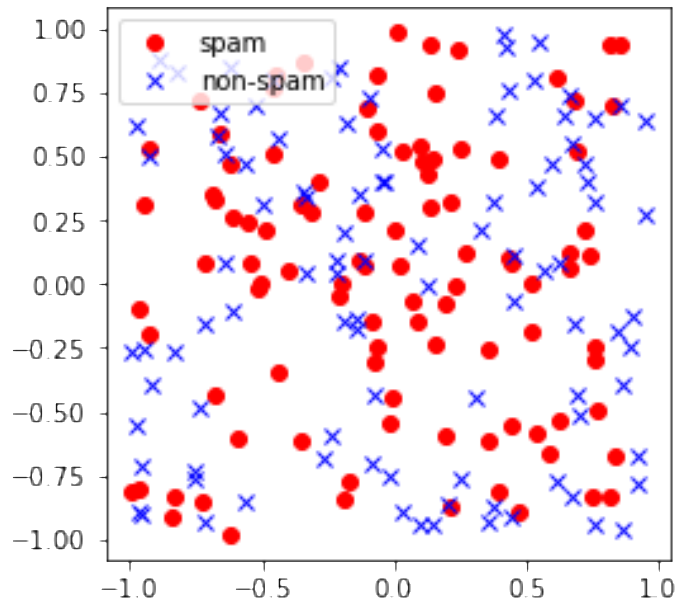
$$\min_{w \in \mathbb{R}^d} \text{Error}(w) = \frac{1}{n} \sum_{i=1}^n \mathbf{1}(h_w(x_i) \neq y_i)$$

- Learning: Find the linear classifier that makes **the smallest number of mistakes** on the training data.

What happens if the linear classifier with the smallest number of mistakes still makes a mistake 49% of the time?

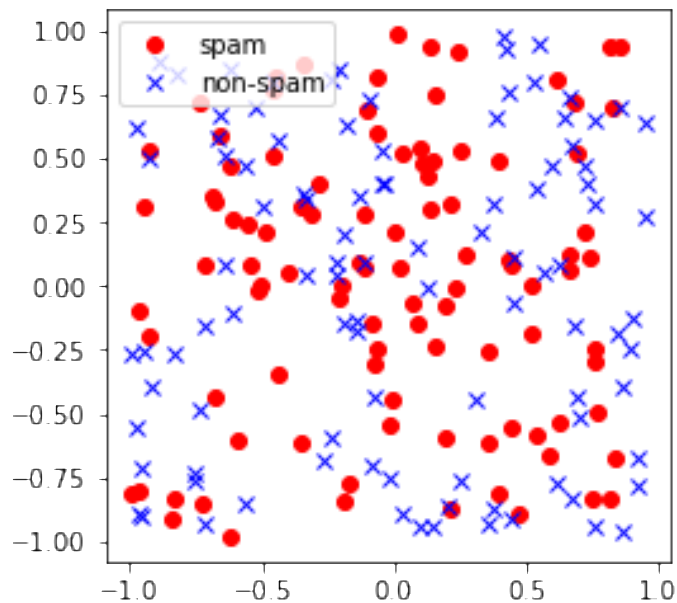
What happens if the linear classifier with the smallest number of mistakes still makes a mistake 49% of the time?

Case 1:

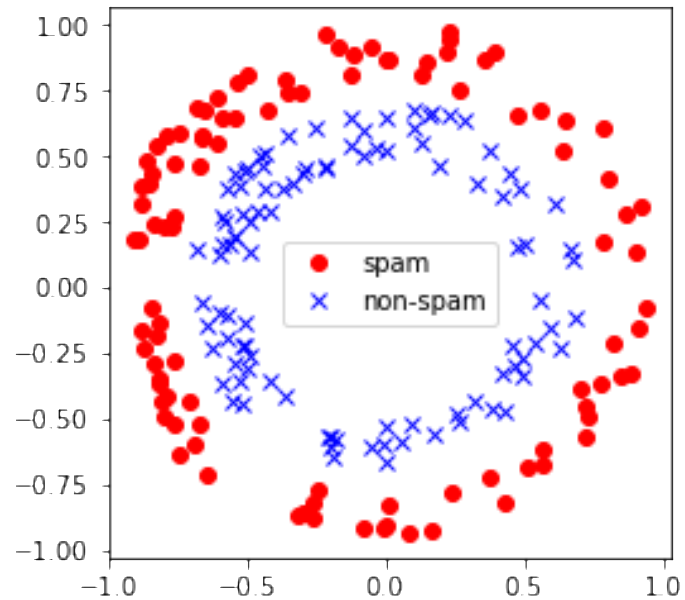


What happens if the linear classifier with the smallest number of mistakes still makes a mistake 49% of the time?

Case 1:

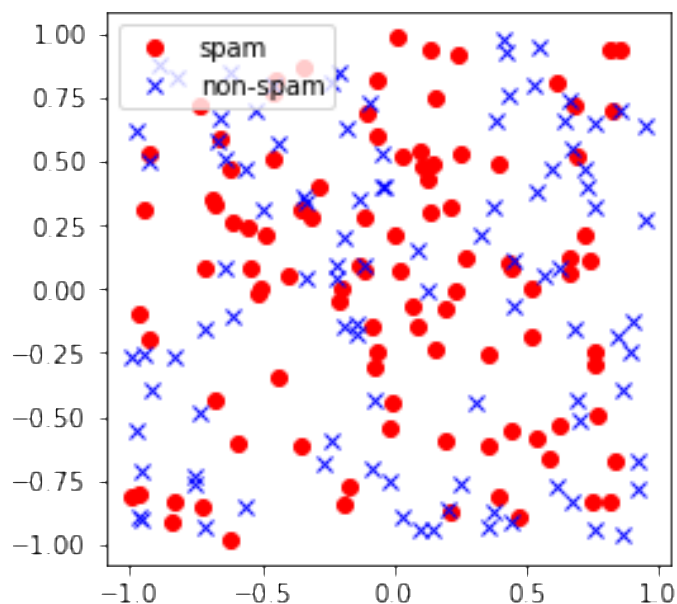


Case 2:

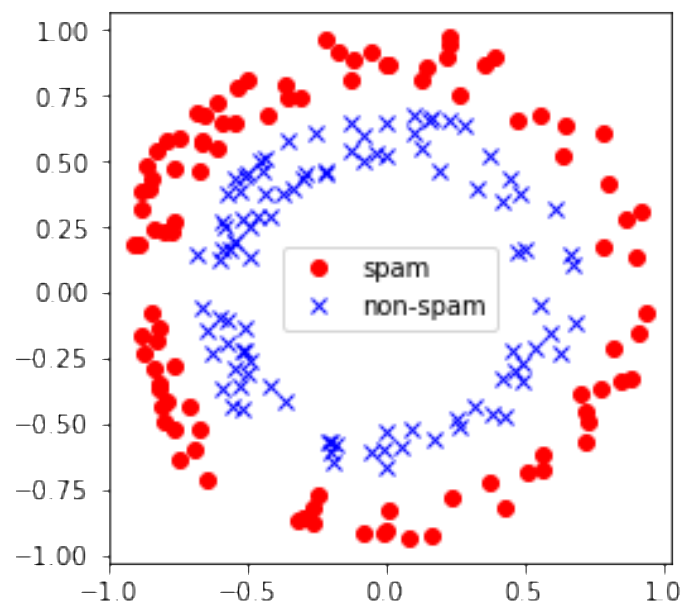


What happens if the linear classifier with the smallest number of mistakes still makes a mistake 49% of the time?

Case 1:



Case 2:

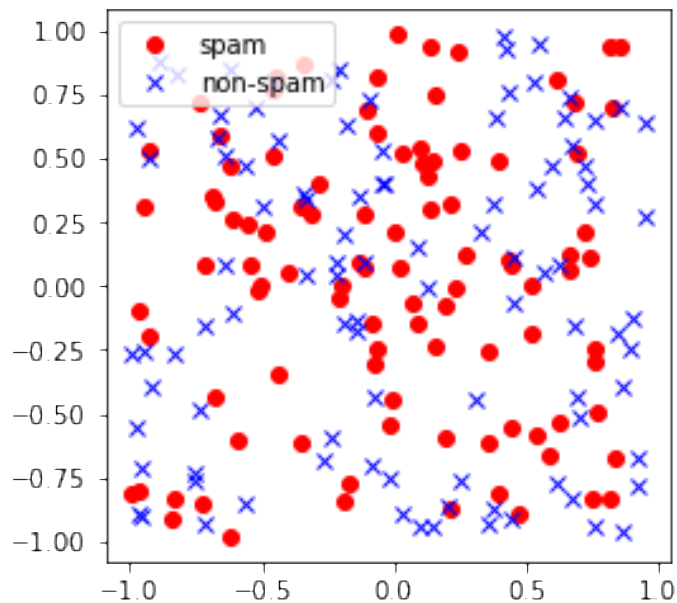


There is no information about the label in the features.

No classifiers are able to do well.

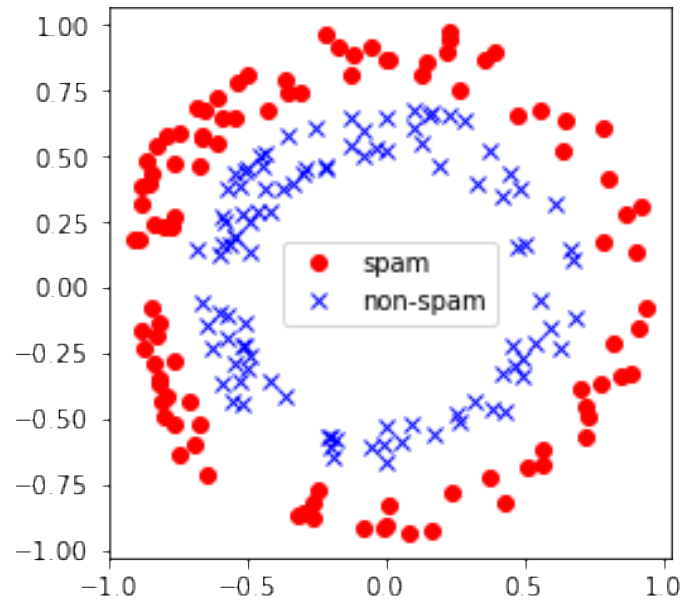
What happens if the linear classifier with the smallest number of mistakes still makes a mistake 49% of the time?

Case 1:



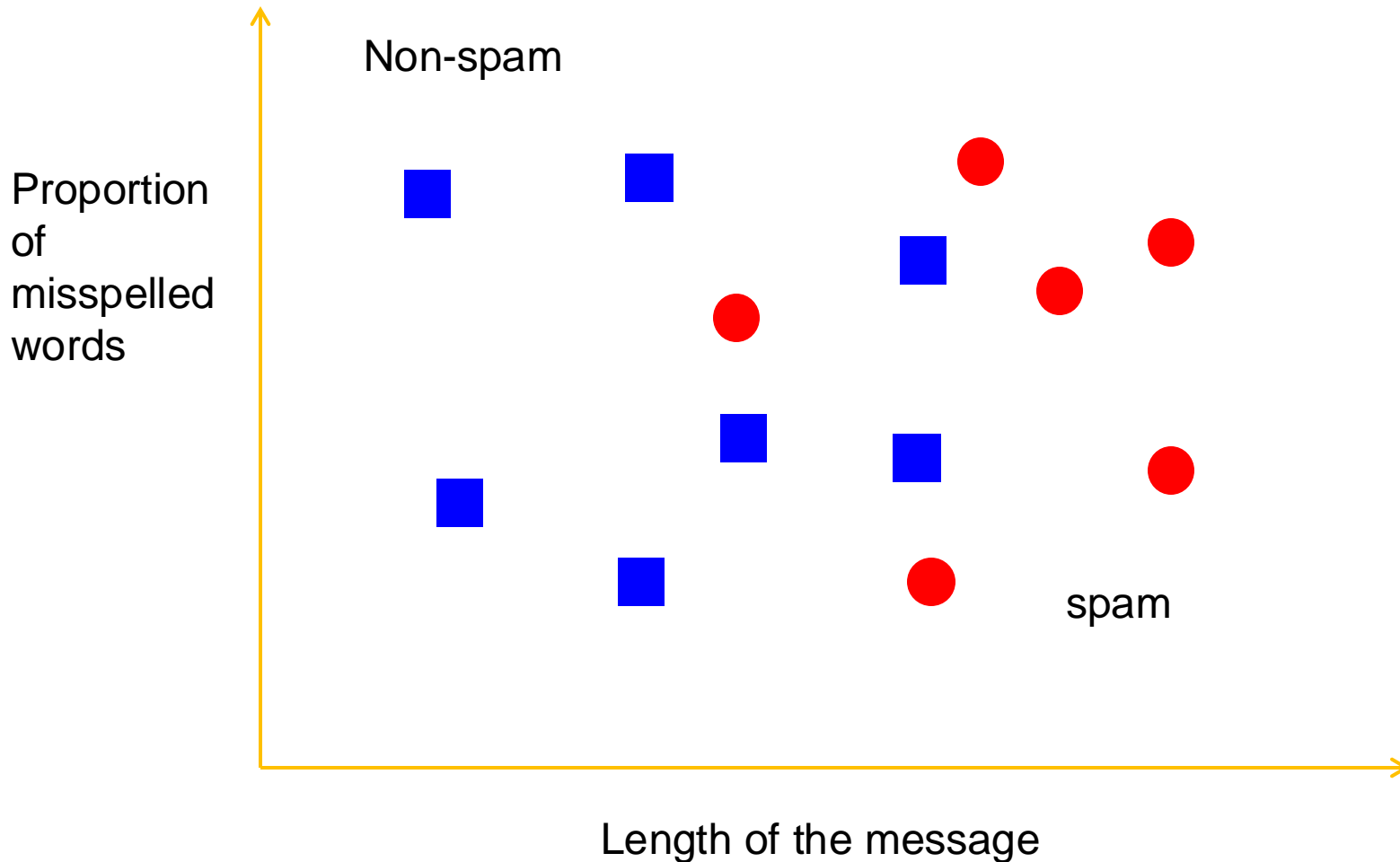
There is no information about the label in the features.
No classifiers are able to do well.

Case 2:

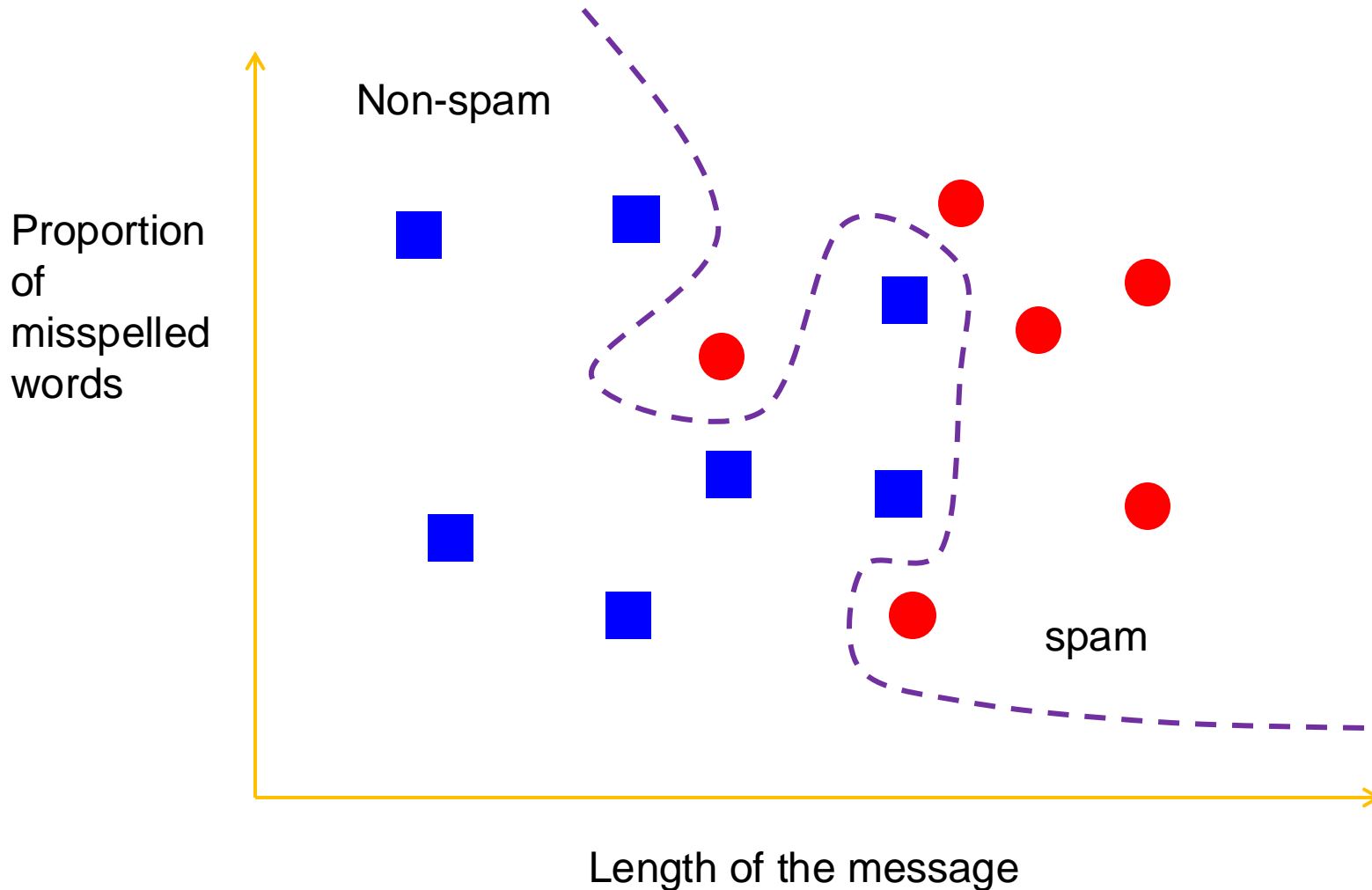


There are some nonlinear classifier that works. But no linear classifiers will do better than chance.

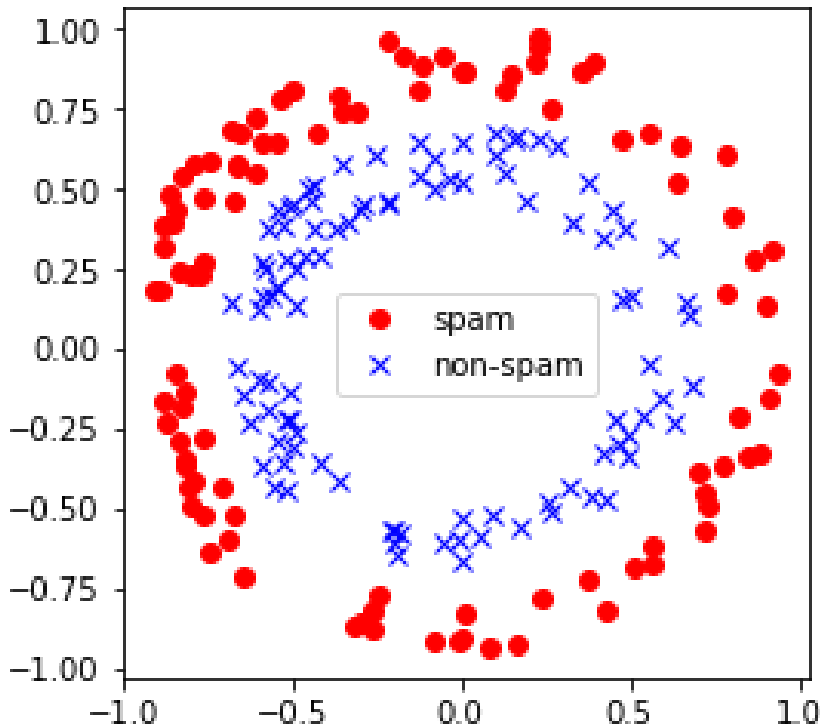
Going to higher dimensions? Maybe we can also allow non-linear decision boundaries?



Going to higher dimensions? Maybe we can also allow non-linear decision boundaries?



Example: Feature transformation.



What we can do:

$$(\tilde{x}_1, \tilde{x}_2) = \left(\sqrt{x_1^2 + x_2^2}, \arctan(x_2/x_1) \right)$$

In the redefined space, the two classes are now linearly separable.

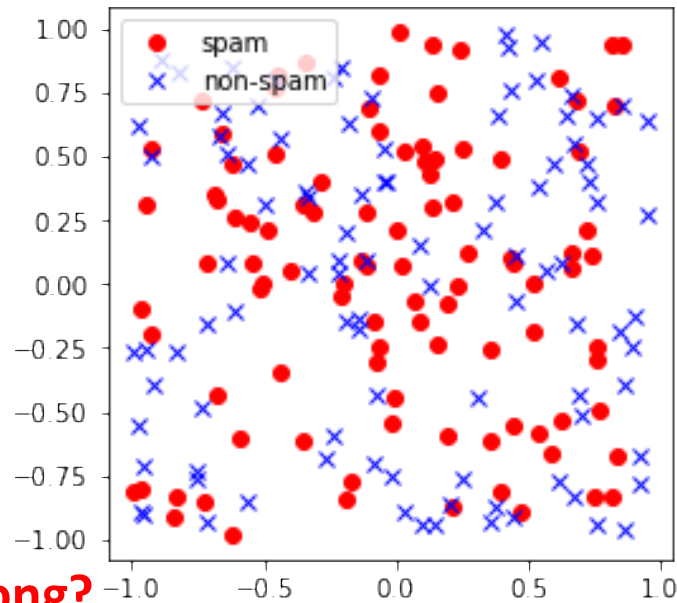
Nonparametric classifiers

- Increasing the complexity of the classifier as we get more data
- For example:
 - We can use the entire training dataset as “free parameters” of the classifier.
 - k-Nearest Neighbor
 - Kernel methods (lifting to infinite dimensional space)
 - Neural networks (design a model for a fixed data size)

Question: What is the classification error of 1-NN classifiers?

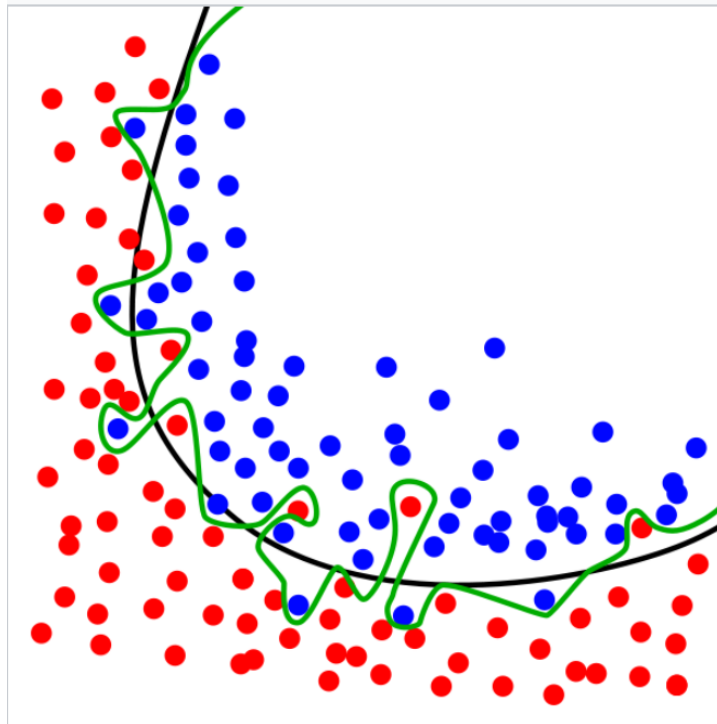
We can make the classifiers arbitrarily accurate... with 1-NN classifier; or with bigger and bigger neural networks.

- Even if the data look like:



- **What went wrong?**

The problem of Overfitting



The green line represents an overfitted model. While the green line best follows the training data, it is too dependent on that data and it is likely to have a higher error rate on new unseen data.

Fundamental problem in machine learning:
The learner **only sees the “training data”** but ideally **wants to do well on “new data”!**

- The problem of generalization.
- All performance metrics we learned before should be calculated on the new data.
- How to evaluate a classifier in practice?
 - We will cover data splitting in Lecture 3.

Summary of today's lecture

- Machine learning overview
- Supervised learning: Spam filtering as an example
 - Features, feature extraction
 - Models, hypothesis class
 - Performance metric
 - Choosing an appropriate hypothesis class
 - Overfitting and generalization

Coming up next lecture

- Practical method for evaluating a classifier
- Linear algebra review
- Standard notation for machine learning