# Artificial Intelligence

flello!

CS 165A

Oct 6, 2020

Instructor: Prof. Yu-Xiang Wang

Today

→ Machine learning Overview

→ Supervised learning

# Homework 1 released

- First two questions are refreshers of what you are likely to be using (a lot) in this course.
- Q3 is about the problem solving process of designing the environment for AI agent.
- Q4 – Q6 is about getting you to implement a simple "classifier" agent from raw data to deployment.
- Start early!

# This Friday: CRML Summit: AI and COVID-19


3 keynotes, 20 high profile speakers / panelists
All are welcome on Zoom

More information here: https://ml.ucsb.edu/2020-responsible-machine-learning-summit

# Recap of the last lecture

- Rational Agents
  - Do the right thing, subject to information / computation constraints
  - Goal of this course: learn how to build such agents

- PEAS
  - Performance measure, Environment, Actuators, Sensors

- New Paradigm: Modelling, Learning, Inference

# Generic Agent Program

- Implementing $f : P^* \rightarrow A$ ...or... $f(P^*) = A$
  - Lookup table?
  - Learning?

# Generic Agent Program

- Implementing $f : P^* \rightarrow A$ **...or...** $f(P^*) = A$
  - Lookup table?
  - Learning?

**function** SKELETON-AGENT(*percept*) **returns** action
**static**: *memory*, the agent's memory of the world

*memory* ← UPDATE-MEMORY(*memory*, *percept*)
*action* ← CHOOSE-BEST-ACTION(*memory*)
*memory* ← UPDATE-MEMORY(*memory*, *action*)
**return** *action*

# Generic Agent Program

- Implementing $f : P^* \rightarrow A$ **...or...** $f(P^*) = A$
  - Lookup table?
  - Learning?

Knowledge, past percepts, past actions

**function** SKELETON-AGENT(*percept*) **returns** action
**static**: *memory*, the agent's memory of the world

*memory* ← UPDATE-MEMORY(*memory*, *percept*)
*action* ← CHOOSE-BEST-ACTION(*memory*)
*memory* ← UPDATE-MEMORY(*memory*, *action*)
**return** *action*

# Generic Agent Program

- Implementing $f : P^* \rightarrow A$ **...or...** $f(P^*) = A$
  - Lookup table?
  - Learning?

Knowledge, past percepts, past actions

**function** SKELETON-AGENT(*percept*) **returns** action
**static**: *memory*, the agent's memory of the world

*memory* ← UPDATE-MEMORY(*memory*, *percept*)
*action* ← CHOOSE-BEST-ACTION(*memory*)
*memory* ← UPDATE-MEMORY(*memory*, *action*)
**return** *action*

e.g.,

**Table-Driven-Agent**

Add *percept* to percepts

LUT [percepts, table]

NOP

# AIMA's categorization of agent programs

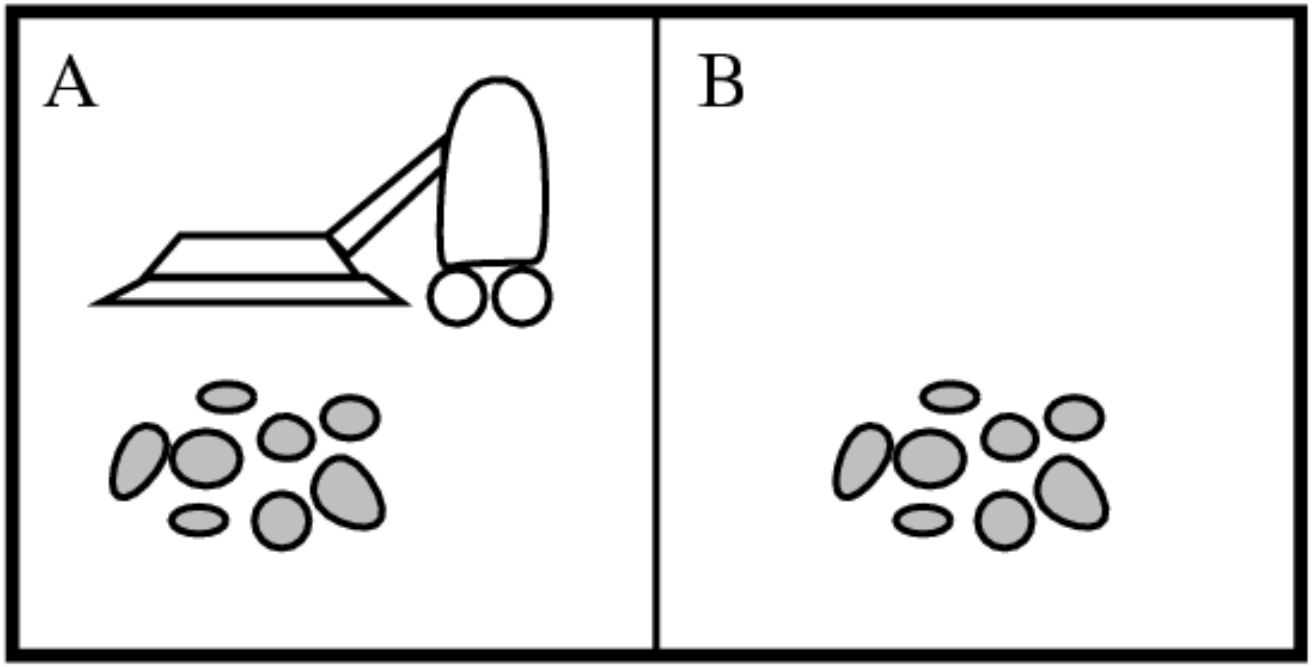
- Simple reflex agent
- Model-based reflex agent
- Goal-based agent
- Utility-based agent
- Learning agent

(Read more in Section 2.4 of the AIMA book.)

Potential mid-term questions:

1. Where do these agent fall under our new categorization?
2. What are these agent's "Modelling-Inference-Learning" components?

# Quiz: What kind of agent it is in the Vacuum world?




- Reflex, planning, reasoning?
- What is the model? Are there any learning components?

Reflex , planning , Reasoning

# When to use which type of agent?

# When to use which type of agent?

- Depends on the problem (task environment)
  - Stochastic/deterministic/stateful/adversarial …

# When to use which type of agent?

- Depends on the problem (task environment)
  - Stochastic/deterministic/stateful/adversarial …
- Depends the amount of data available
  - Often we need to learn how the world behaves

# When to use which type of agent?

- Depends on the problem (task environment)
  - Stochastic/deterministic/stateful/adversarial …
- Depends the amount of data available
  - Often we need to learn how the world behaves
- Depends on the dimensionality of your observations

SLAM

# When to use which type of agent?

- Depends on the problem (task environment)
  - Stochastic/deterministic/stateful/adversarial …
- Depends the amount of data available
  - Often we need to learn how the world behaves
- Depends on the dimensionality of your observations

Solving the right problem approximately

vs

Solving an approximation of the problem exactly

*"All models are wrong, but some are useful."*

George Box
(1919 - 2013)

# Structure of the course


Classification / Regression
Bandits

Search
game playing

Markov Decision Processes
Reinforcement Learning

Logic, knowledge base
Probabilistic inference

**Reflex Agents**

**Planning Agents**

**Reasoning agents**

Low-level intelligence

High-level intelligence


(Again this idea is adapted from Percy Liang's teachings)

# 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

# Different Types of Machine Learning

# Different Types of Machine Learning

- Supervised Learning

# Different Types of Machine Learning

- Supervised Learning
- Unsupervised Learning

# Different Types of Machine Learning

- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

# Different Types of Machine Learning

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

# Different Types of Machine Learning

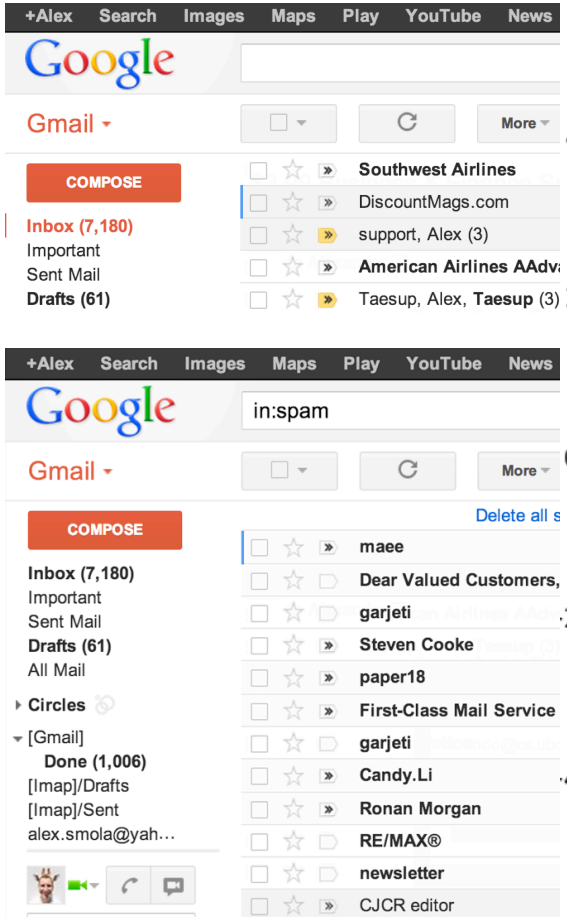
- Supervised Learning — Spam Filter.
- Unsupervised Learning — Topics of a body of texts
- Reinforcement Learning — Atari Games. Serve Ads.
- Structured Prediction — Machine translation.

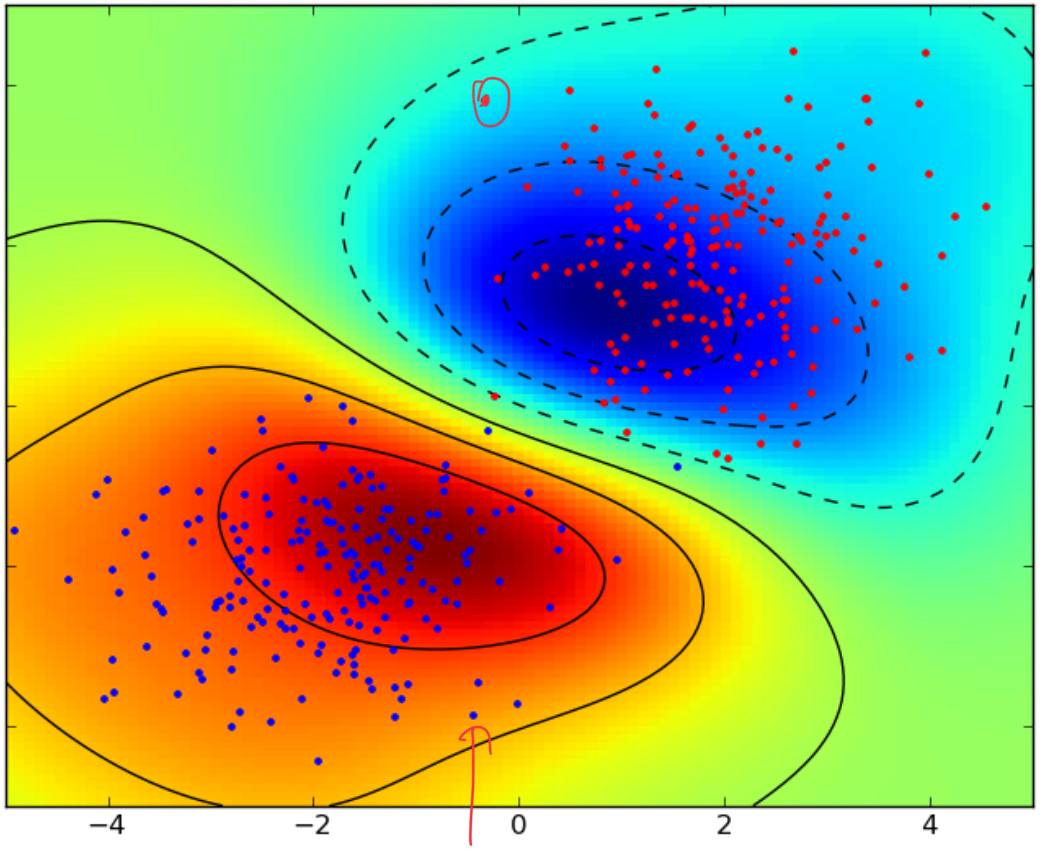
# Different Types of Machine Learning

- Supervised Learning — Spam Filter.
- Unsupervised Learning — Topics of a body of texts
- Reinforcement Learning — Atari Games. Serve Ads.
- Structured Prediction — Machine translation.

Bandits and reinforcement learning after the midterm.

# Supervised learning



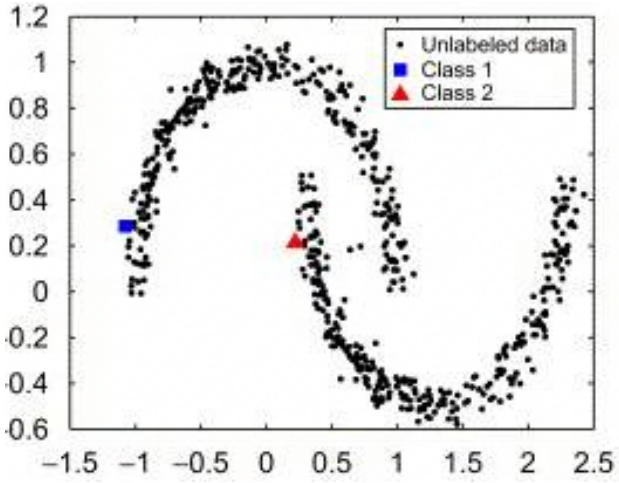
# Unsupervised Learning

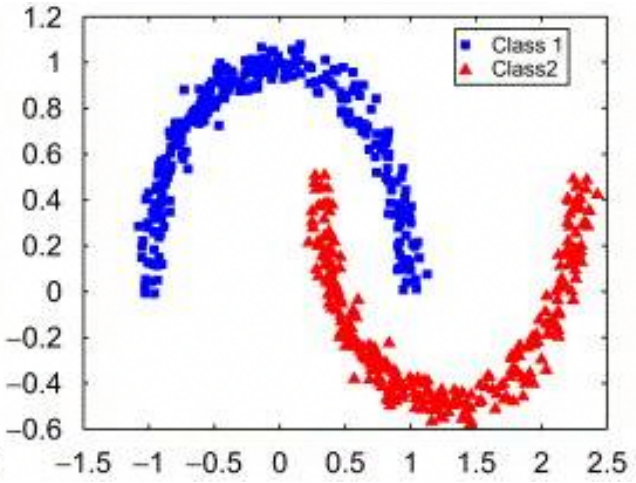
topic model

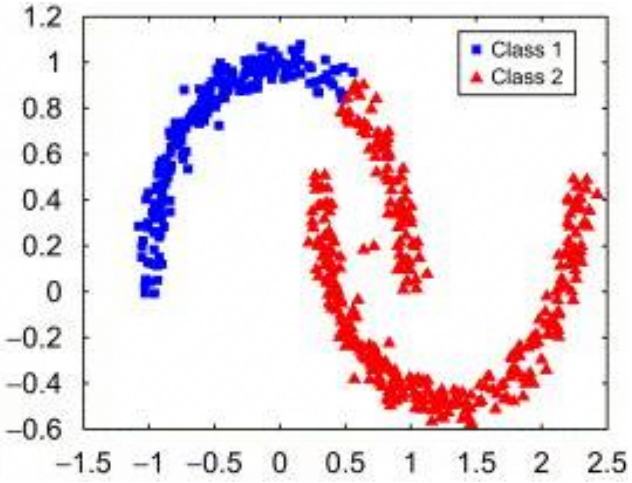
| "Arts" | "Budgets" | "Children" | "Education" |
|---|---|---|---|
| NEW | MILLION | CHILDREN | SCHOOL |
| FILM | TAX | WOMEN | STUDENTS |
| SHOW | PROGRAM | PEOPLE | SCHOOLS |
| MUSIC | BUDGET | CHILD | EDUCATION |
| MOVIE | BILLION | YEARS | TEACHERS |
| PLAY | FEDERAL | FAMILIES | HIGH |
| MUSICAL | 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.

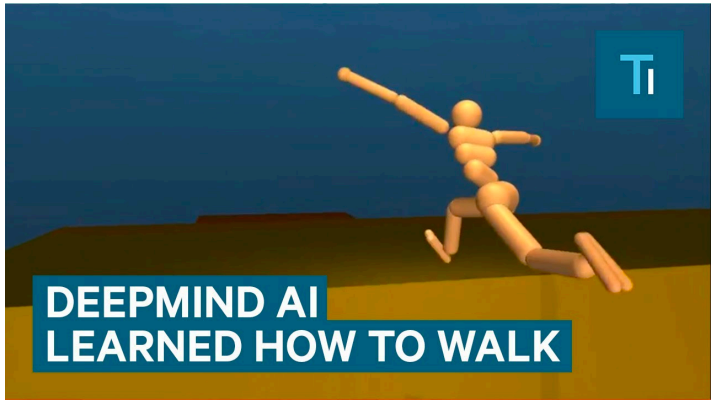
# Semi-supervised Learning

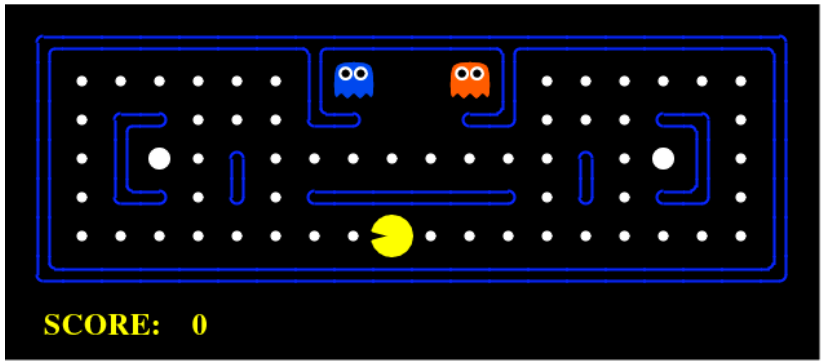


# Reinforcement learning


amazon

Recommendations

buy or not buy

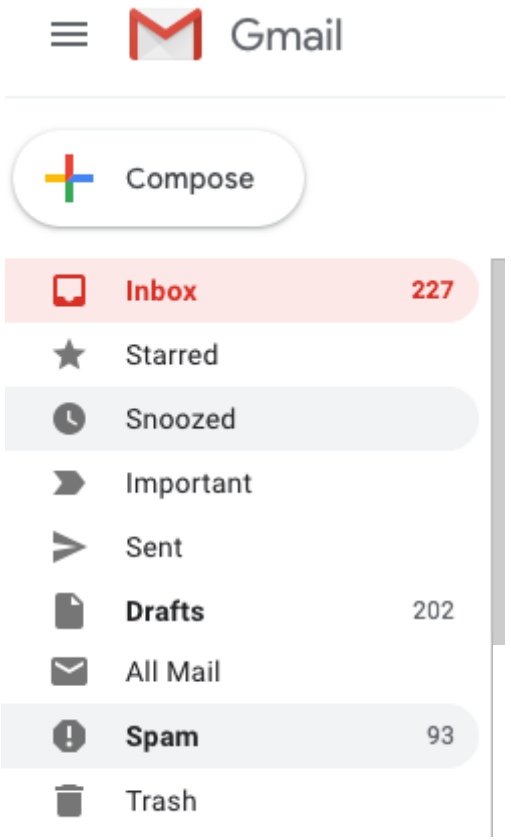
# The focus of today's lecture is "Supervised Learning"

- 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

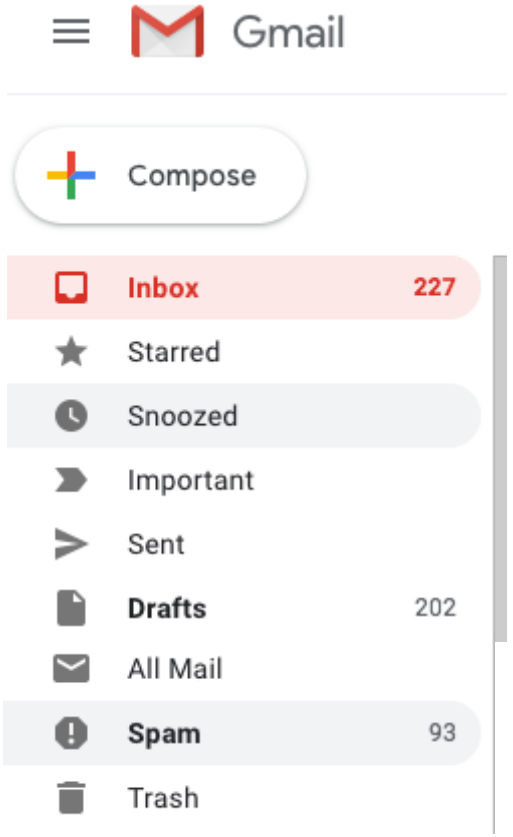


# The focus of today's lecture is "Supervised Learning"

- 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

CLAIMS.

To: undisclosed-recipients:;,

Reply-To: microworld219@gmail.com

MICROWORLD CORPORATIONS:
CUSTOMER SERVICE:
FRIEDRICHSTRAßE 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

(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,

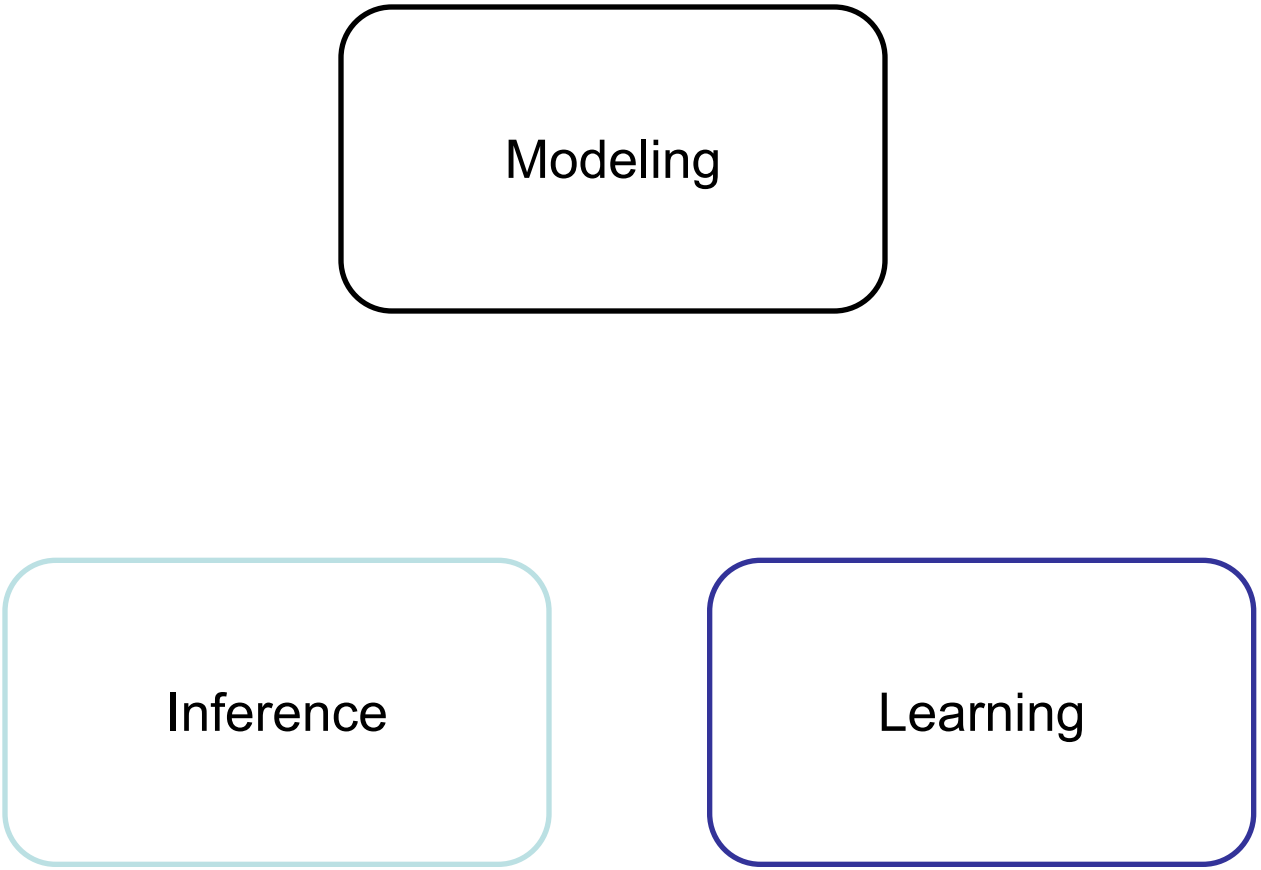
I am a student in your machine learning class.

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?

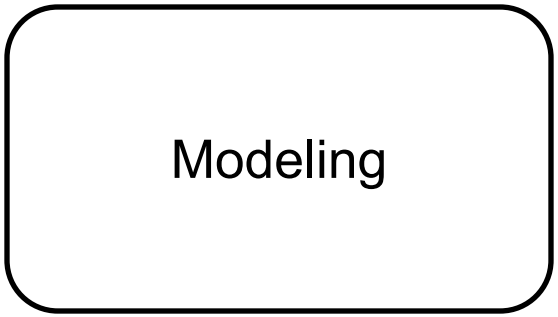
I look forward to hearing from you.
Best regards,
Bar

Quoted from [Here].

# Modelling-Inference-Learning paradigm

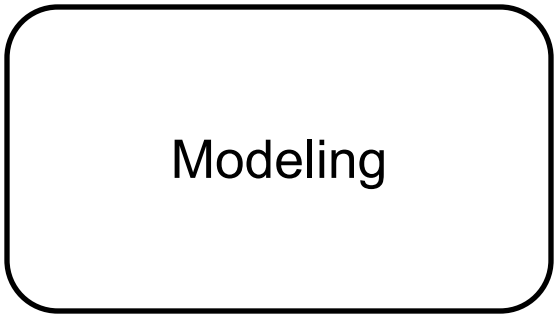


# Modelling-Inference-Learning paradigm




- Feature engineering
- Specify a family of classifiers


# Modelling-Inference-Learning paradigm

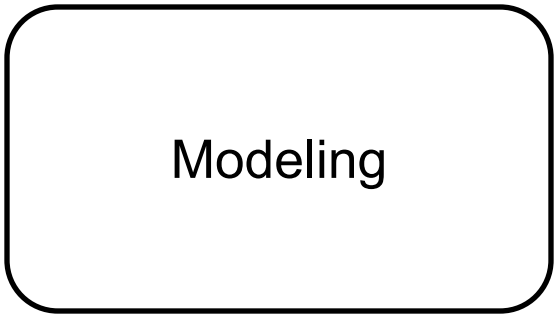



- Feature engineering
- Specify a family of classifiers



Learning the best performing classifier

# Modelling-Inference-Learning paradigm



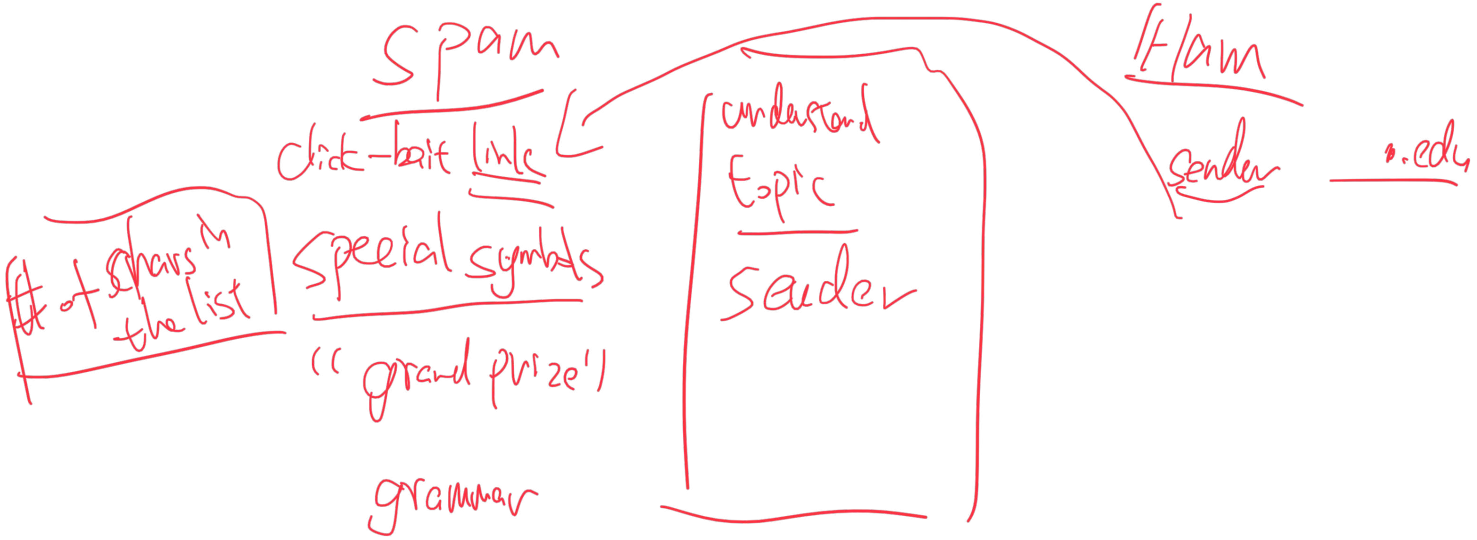

- Feature engineering
- Specify a family of classifiers


Deployment to email client


Learning the best performing classifier

# What are the features that we can use to describe an email (3 min discussions)

- 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?

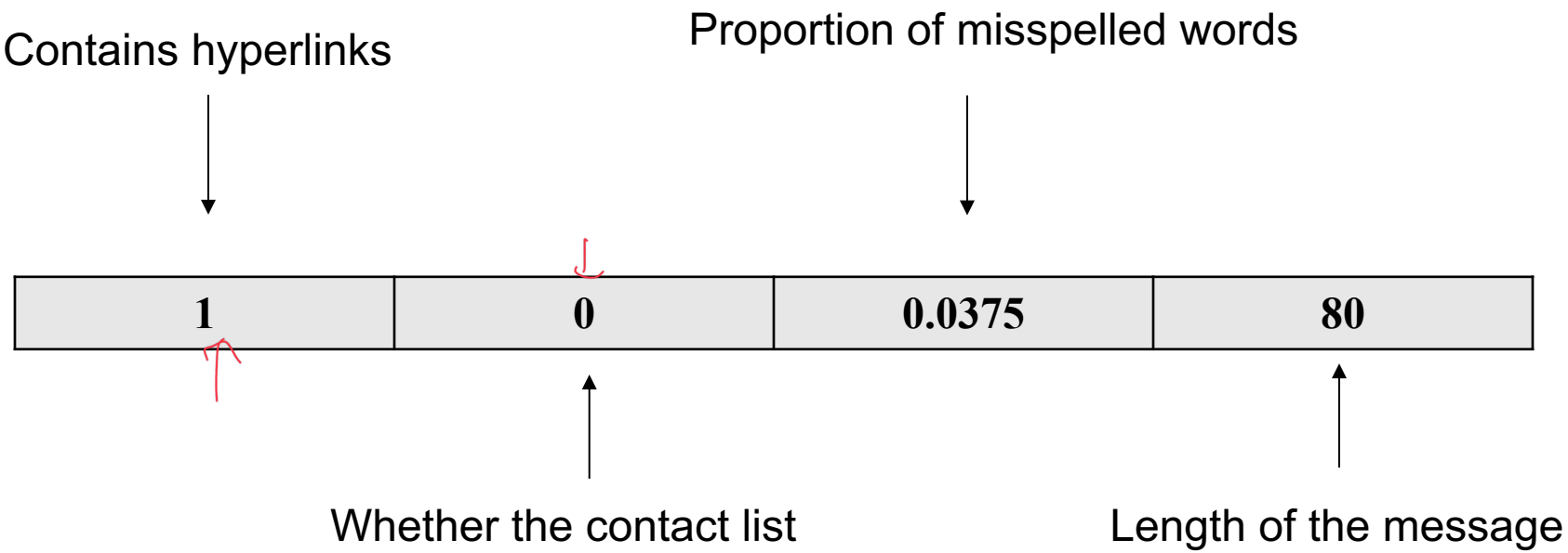


# Possible features

- 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

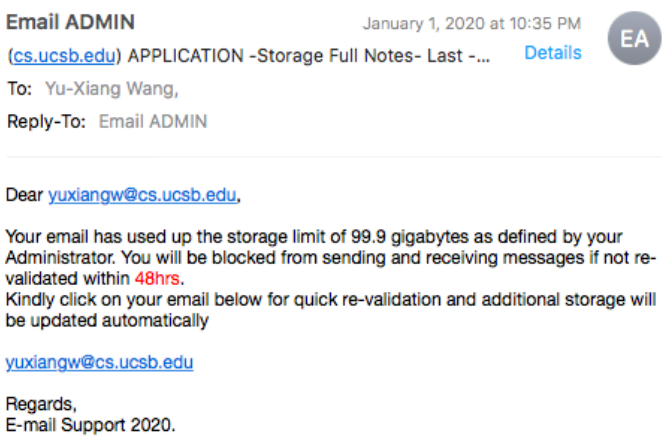
$d=4$

$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_d \end{bmatrix} \in R^d$

| 1 | 0 | 0.0375 | 80 |
|---|---|---|---|

Whether the contact list

Length of the message



Email ADMIN January 1, 2020 at 10:35 PM

(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.

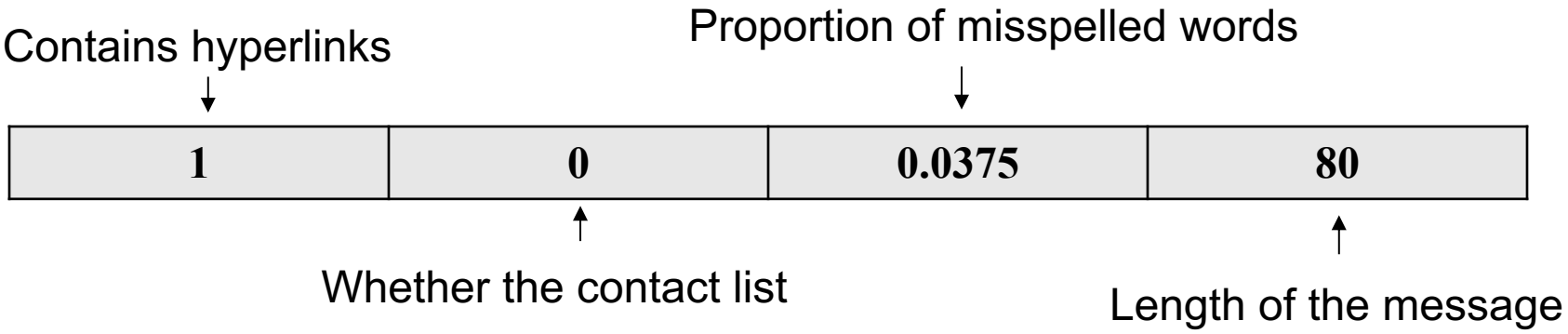
**Step 1 in Modelling**
**Feature extractor**:
Converting the object of interest to a vector of numerical values.

# Mathematically defining a classifier

family

- Feature space: $\mathcal{X} = \mathbb{R}^d$
- Label space: $\mathcal{Y} = \{0, 1\} = \{\text{non-spam}, \text{spam}\}$
- A classifier (hypothesis): $h : \mathcal{X} \to \mathcal{Y}$

# 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)

1. threshold 2. if not on the contact list and contain hyperlink output spam 3. weighted average/threshold $\mathbb{I}(\underbrace{\sum x_i \cdot w_i}_{\text{Score}} \geq \text{threshold})$

# 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

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

# Specifying a family of classifiers --- a "hypothesis class"

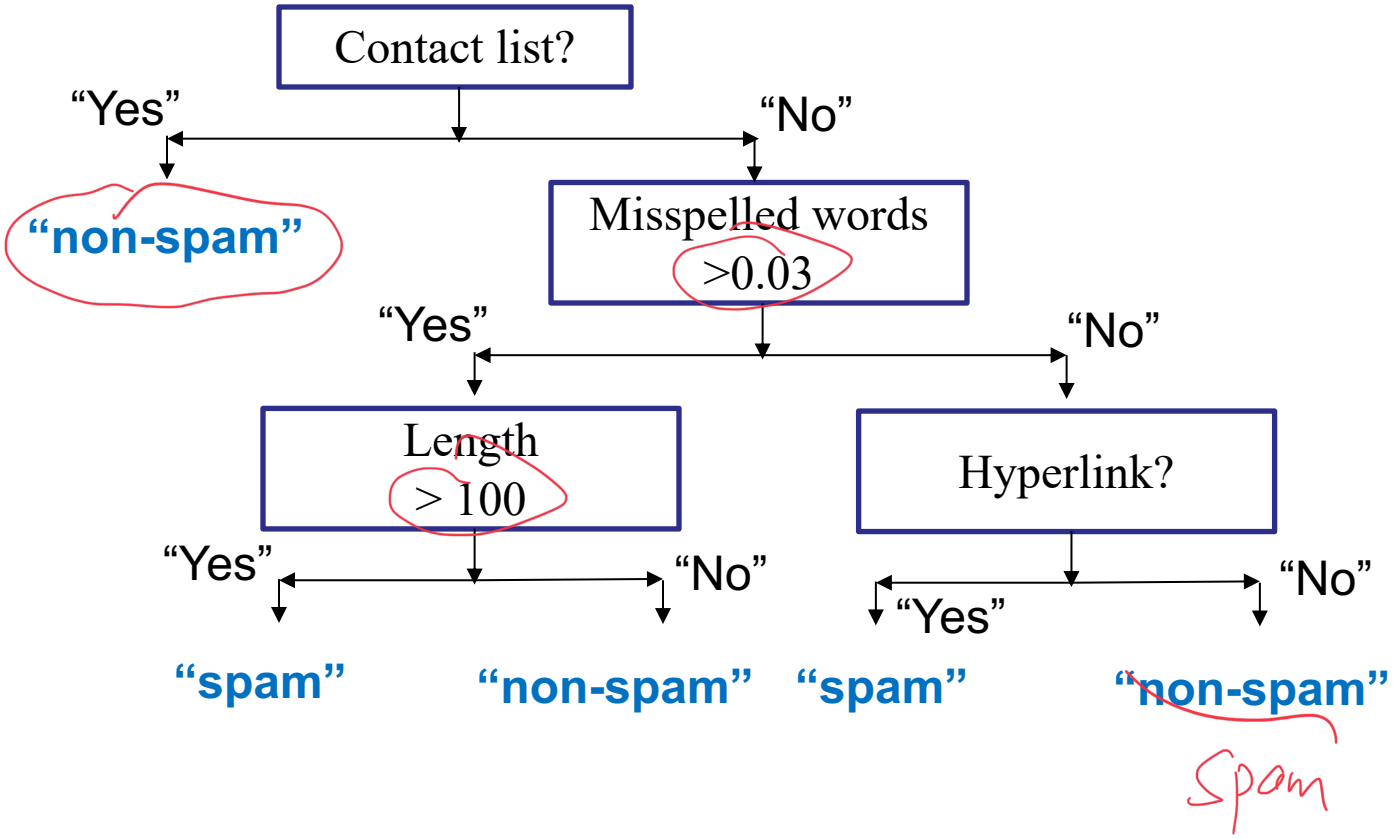
- Hypothesis class
  - A family of classifiers: $\mathcal{H}$
  - Also known as "concept classes", "models", "decision rule book"
  - "Neural networks" and "Support Vector Machines" are hypothesis classes.
  - Typically we want this family to be large and flexible.

- The task of machine learning:
  - A **selection problem** to find a

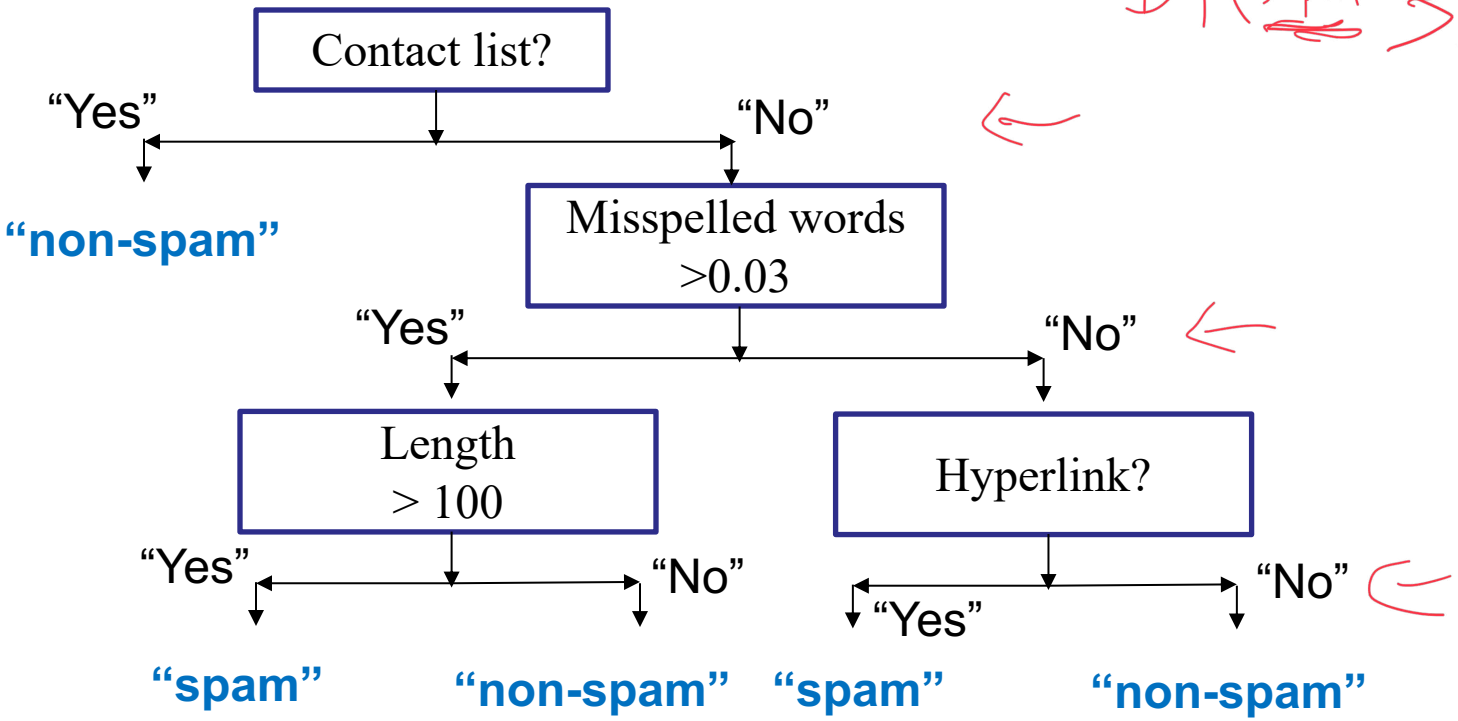
$$h \in \mathcal{H}$$

that "**works well**" on this problem.

# Decision trees



# Decision trees




- **Question:** What are the "free parameters" if we are to learn such a decision tree? Using data?

# Learning a decision tree

- Free parameters:
  - Which feature(s) to use when branching branch?
  - How to branch? Thresholding? Free threshold?
  - Which label to assign at leaf nodes?

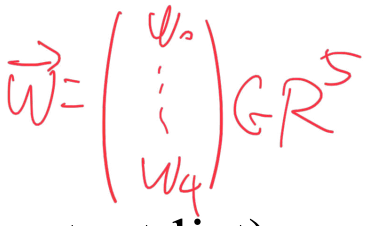
- Hyperparameters:
  - Max height of a decision tree?
  - Number of parameters the tree can use in each

- **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?
  - How many classifiers are there (without restrictions)?

slack quiz

# Example: Linear classifiers

# Example: Linear classifiers




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

# Example: Linear classifiers

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

- A linear classifier: h(x) = 1 if Score(x) > 0 and 0 otherwise.

# Example: Linear classifiers

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

- A linear classifier: h(x) = 1 if Score(x) > 0 and 0 otherwise.

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

# Example: Linear classifiers

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

- A linear classifier: h(x) = 1 if Score(x) > 0 and 0 otherwise.

- Question: What are the "free-parameters" in a linear classifier?
  - If we redefine $\mathcal{Y} = \{-1, 1\}$

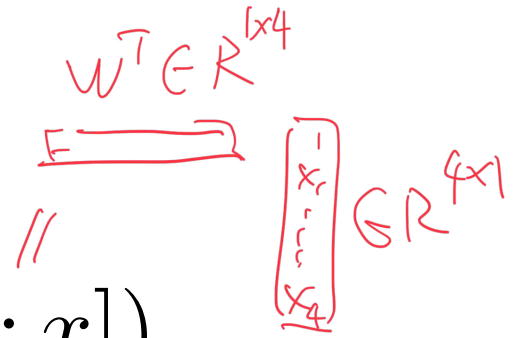
# Example: Linear classifiers

- 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 Score(x) > 0 and 0 otherwise.

- Question: What are the "free-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"!

$\{ x \mid score(x) > 0 \}$

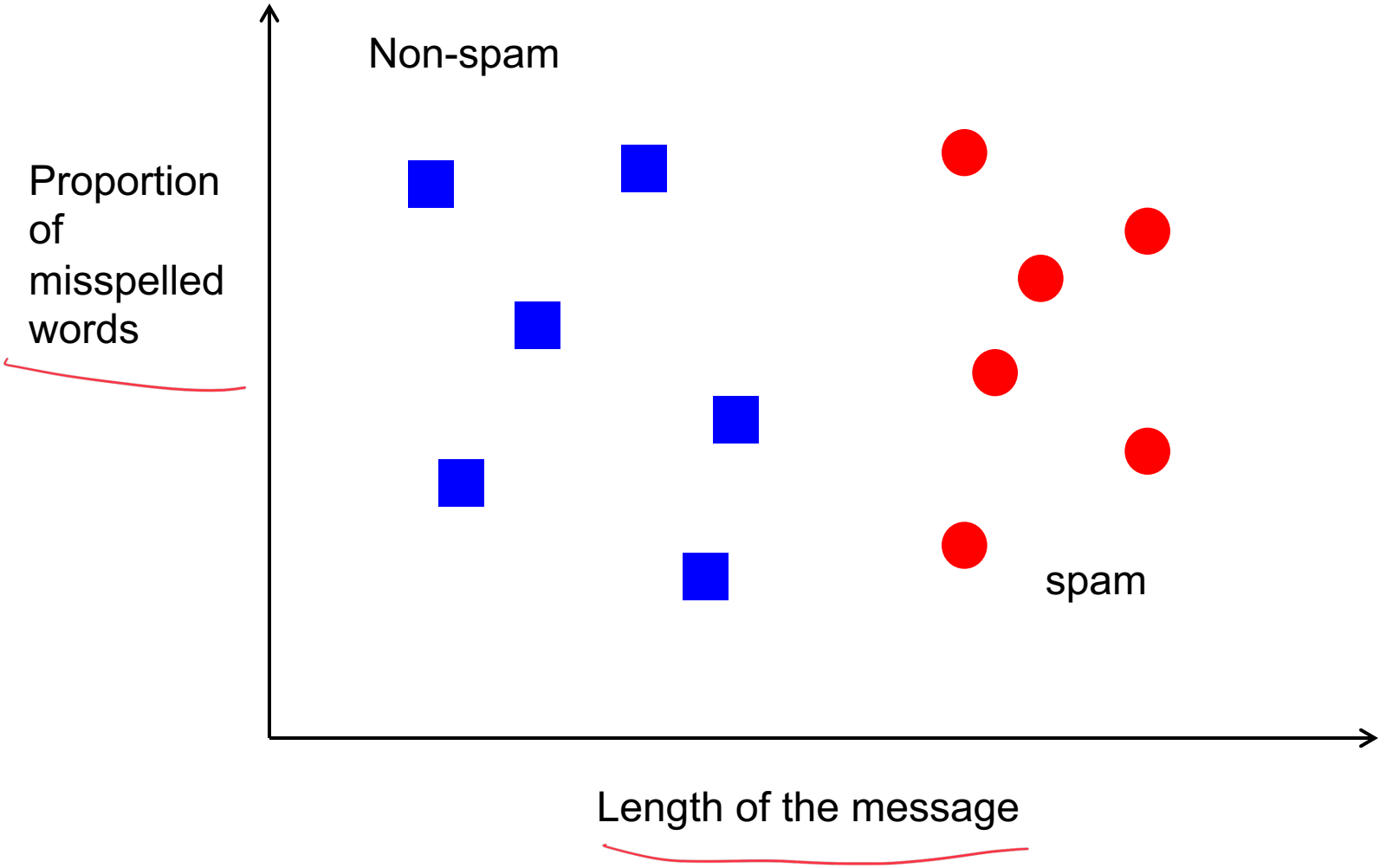
$\{ x \mid w_0 + w_1 * x1 + w_2 * x2 + w_3 * x3 + w_4 * x4 > 0 \}$

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

# Geometric view: Linear classifier are "half-spaces"!

$\{ x \mid w_0 + w_1 * x1 + w_2 * x2 + w_3 * x3 + w_4 * x4 > 0 \}$

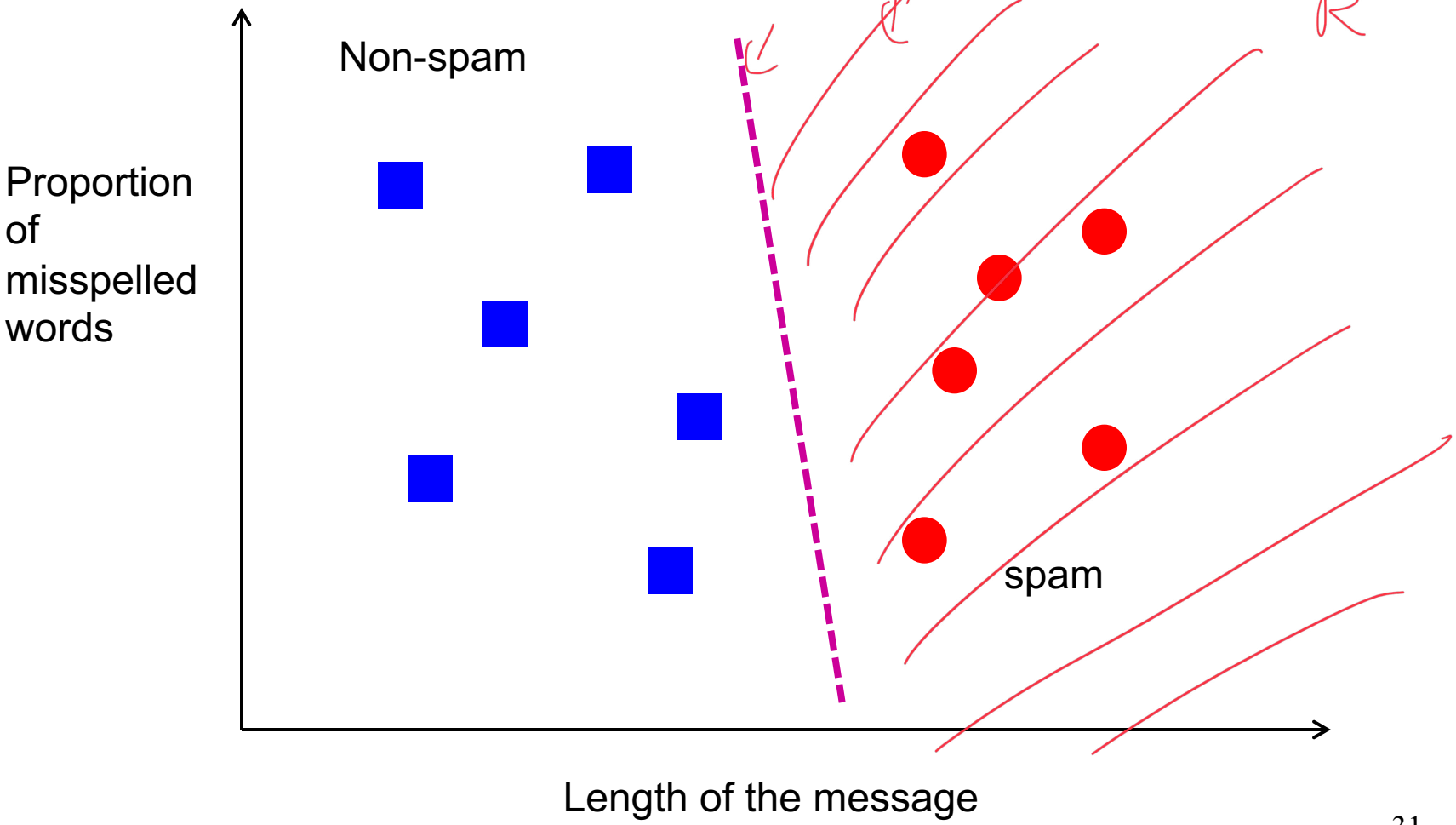
The set of all "emails" that will be classified as "Spams".



# Geometric view: Linear classifier are "half-spaces"!

$\{ x \mid w_0 + w_1 * x1 + w_2 * x2 + w_3 * x3 + w_4 * x4 > 0 \}$

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



# Learning linear classifiers

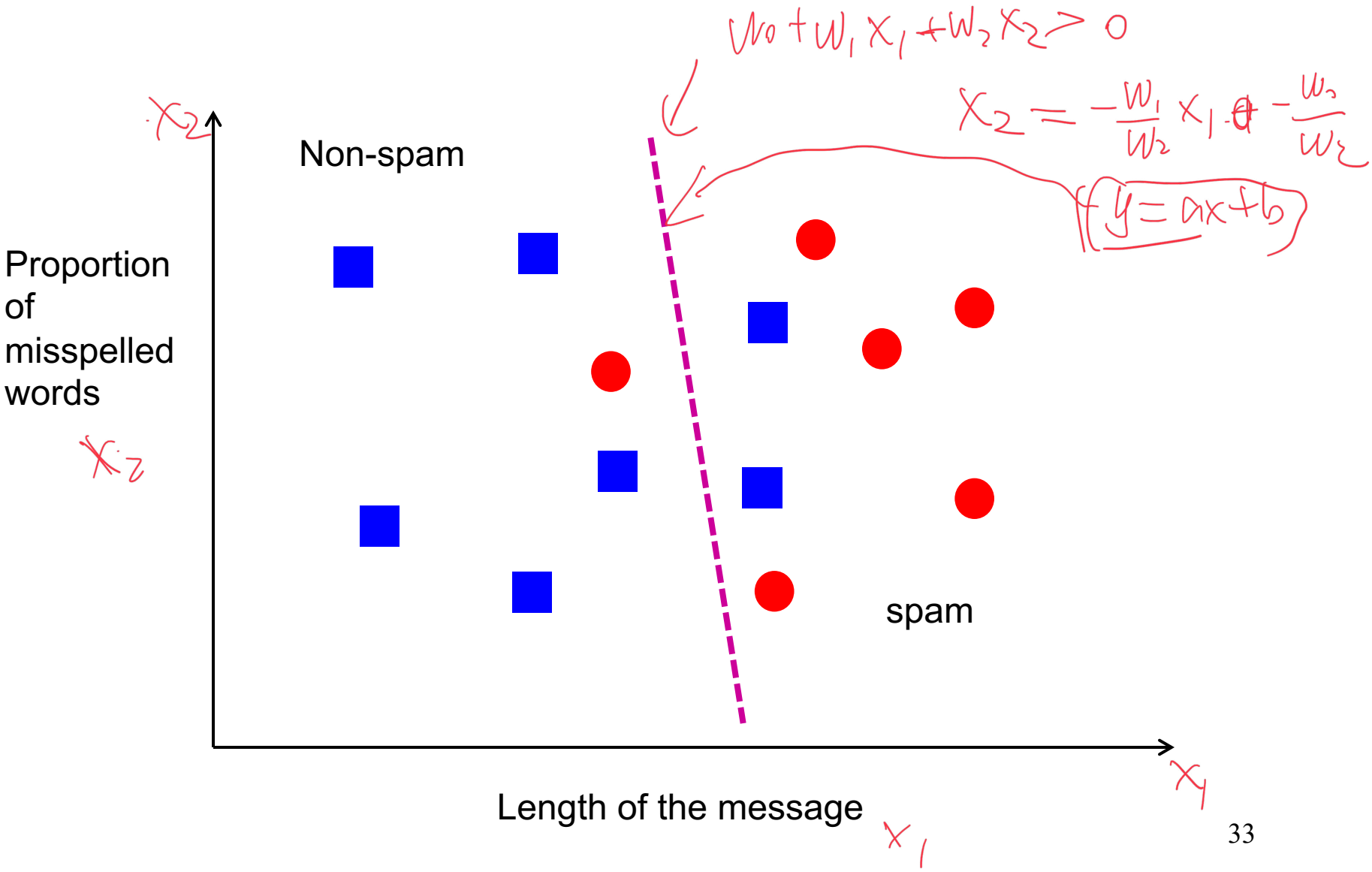
- Training data:

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

- 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 that is consistent with the observed training data.

$y(x_i) = y_i$

# Example: Linearly non-separable cases



# How do we learn a linear classifier in a non-linearly separable case?

- Training data:

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

- Solving the following optimization problem:

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

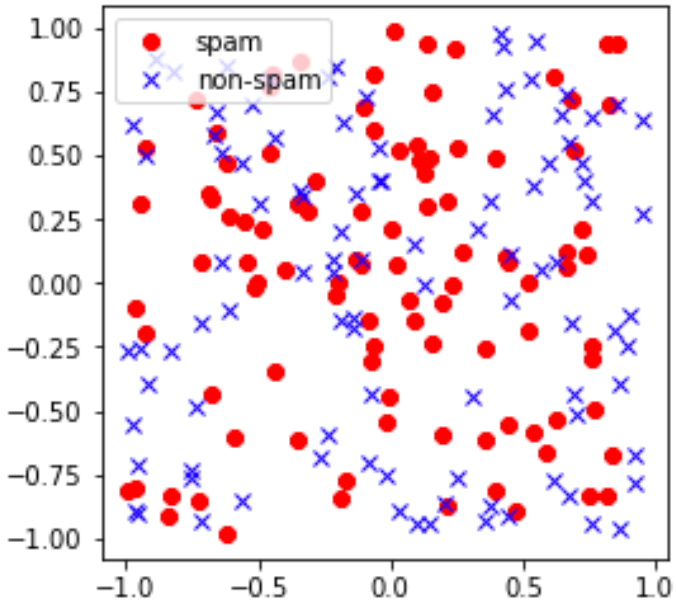
average error rate on training data

- 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?

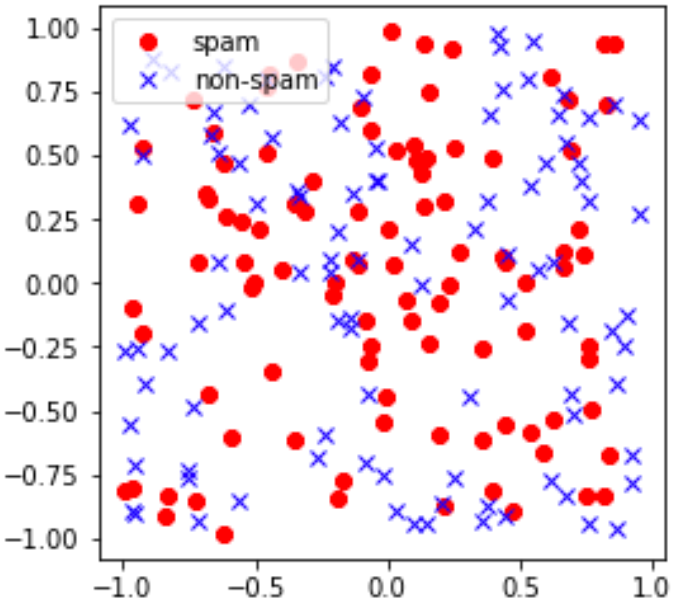
# What happens if the linear classifier with the smallest number of mistakes still makes a mistake 49% of the time?

**Case 1:**

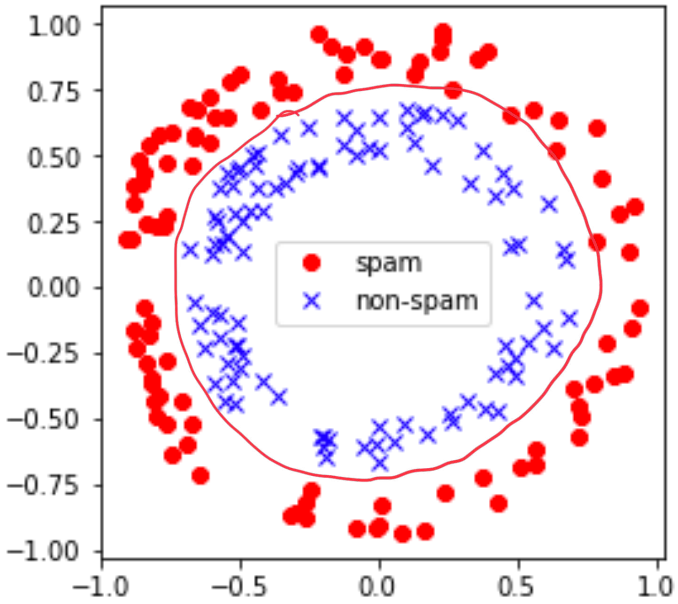


# What happens if the linear classifier with the smallest number of mistakes still makes a mistake 49% of the time?

**Case 1:**



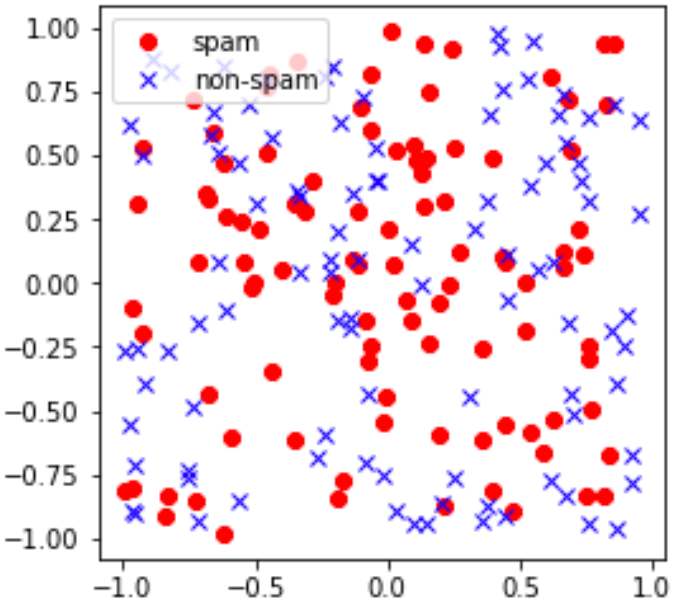

**Case 2:**



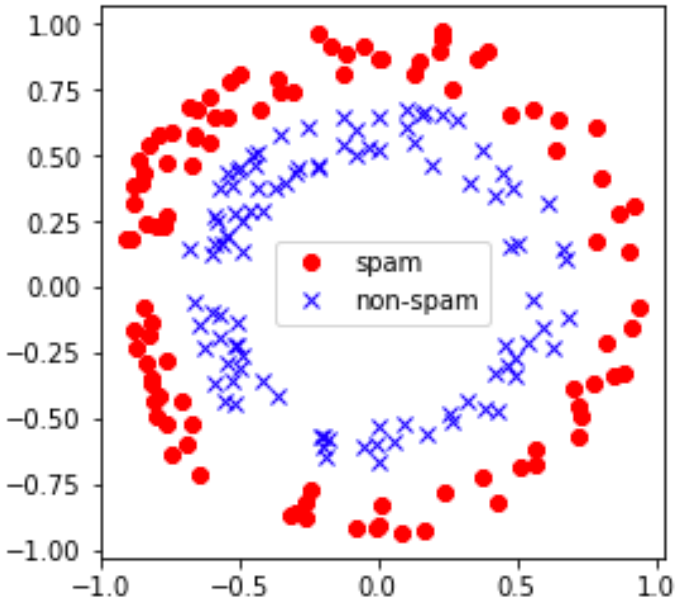

In both cases no linear classifiers will do well

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

**Case 1:**



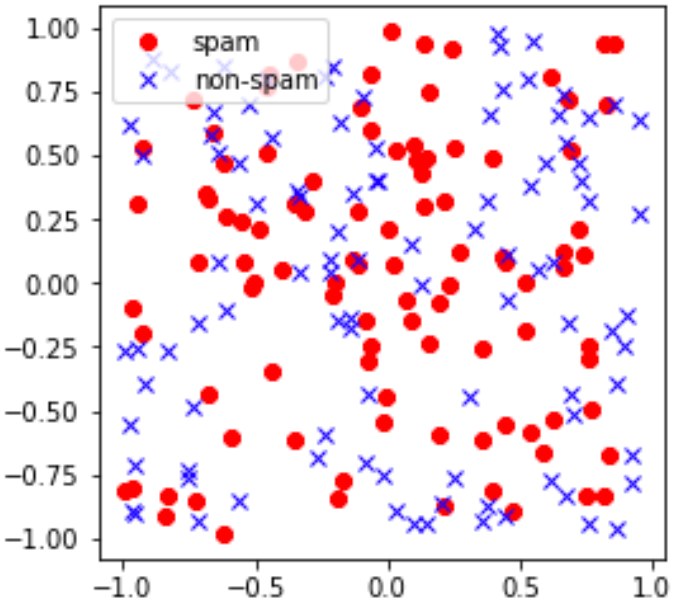

**Case 2:**




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

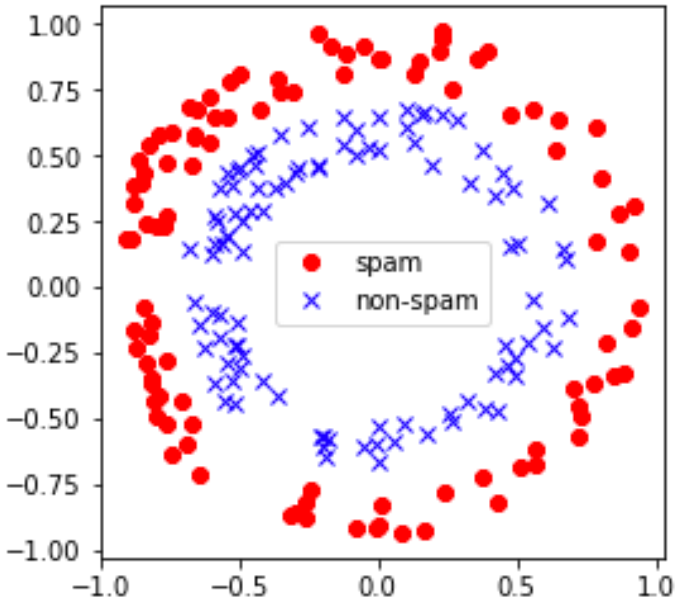
# What happens if the linear classifier with the smallest number of mistakes still makes a mistake 49% of the time?

**Case 1:**



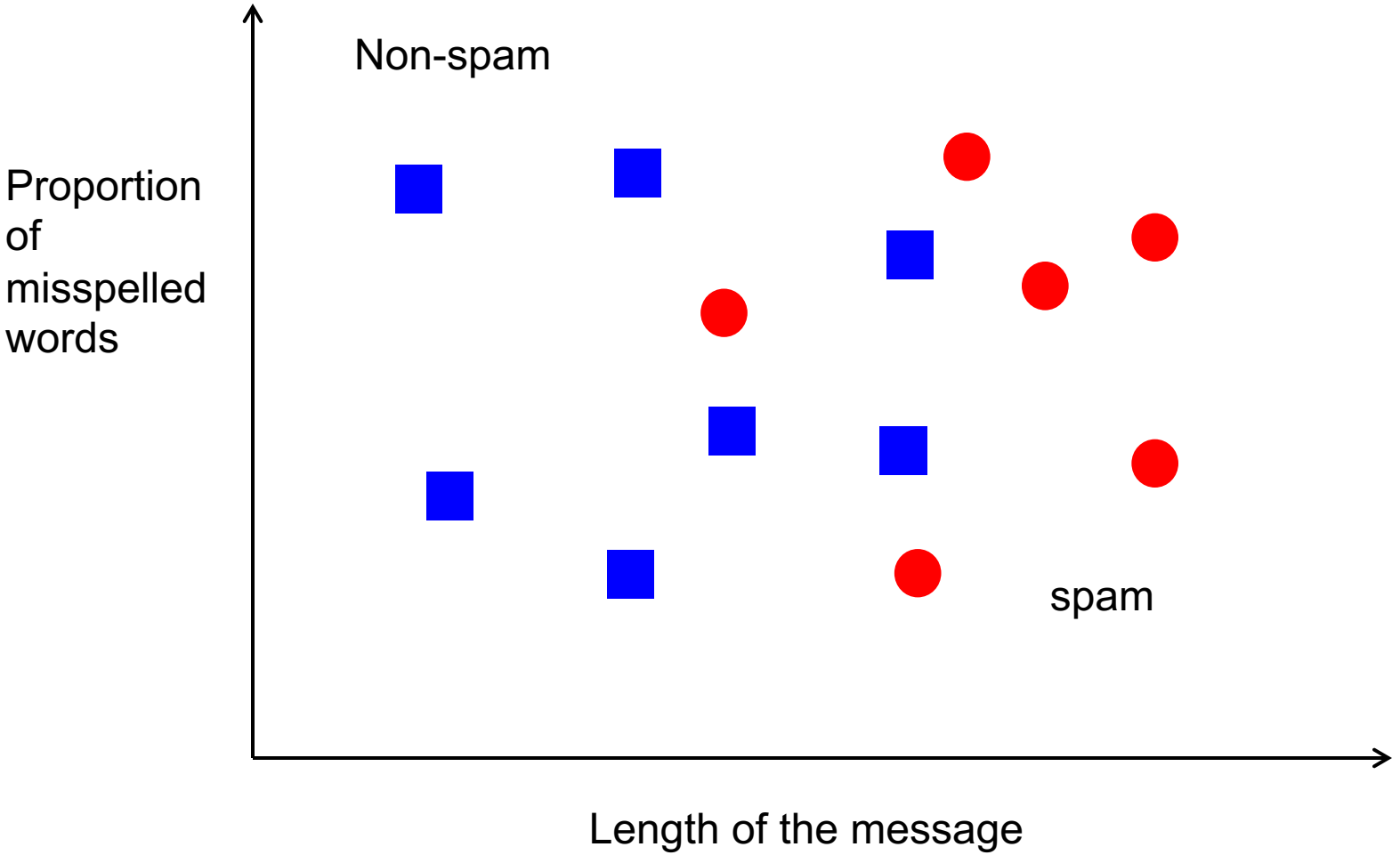

There is no information about the label in the features.
No classifiers are able to do well.

**Case 2:**

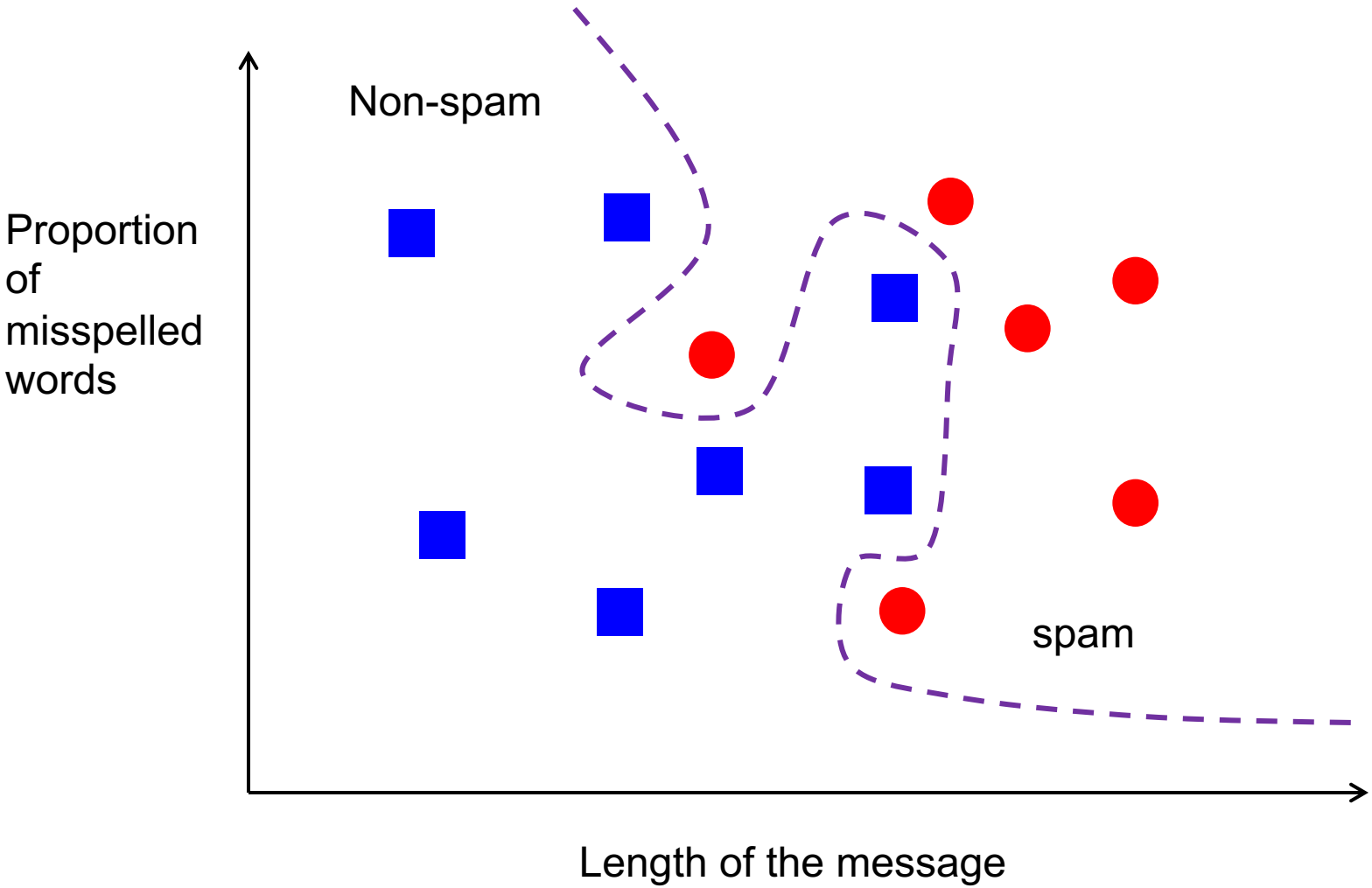



There are some nonlinear classifier that works. But no linear classifiers will do better than chance.

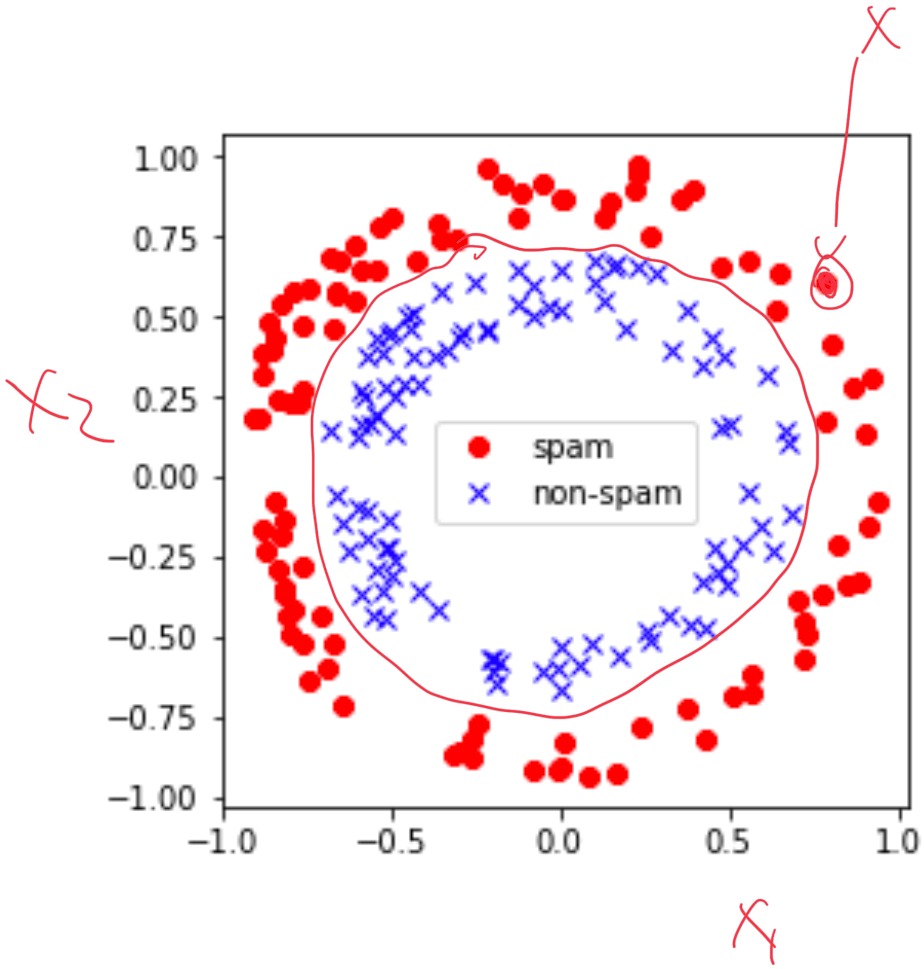
# Going to higher dimensions? Maybe we can also allow non-linear decision boundaries?



# 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

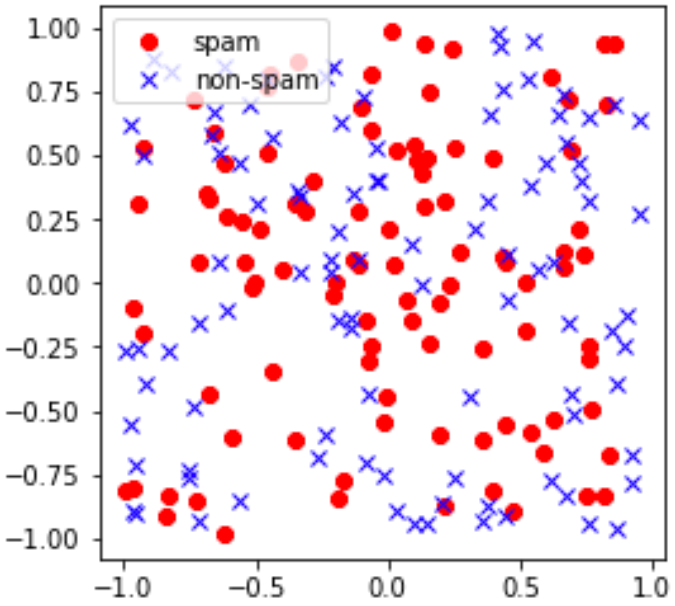
- Increasing the complexity of the classifier as we get more data
- For example:
  - We can use the entire training dataset as "free parameters" of the classifier.
  - k-Nearest Neighbor
  - Kernel methods (lifting to infinite dimensional space)
  - Neural networks (design a model for a fixed data size)

- (More details in the textbook)

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

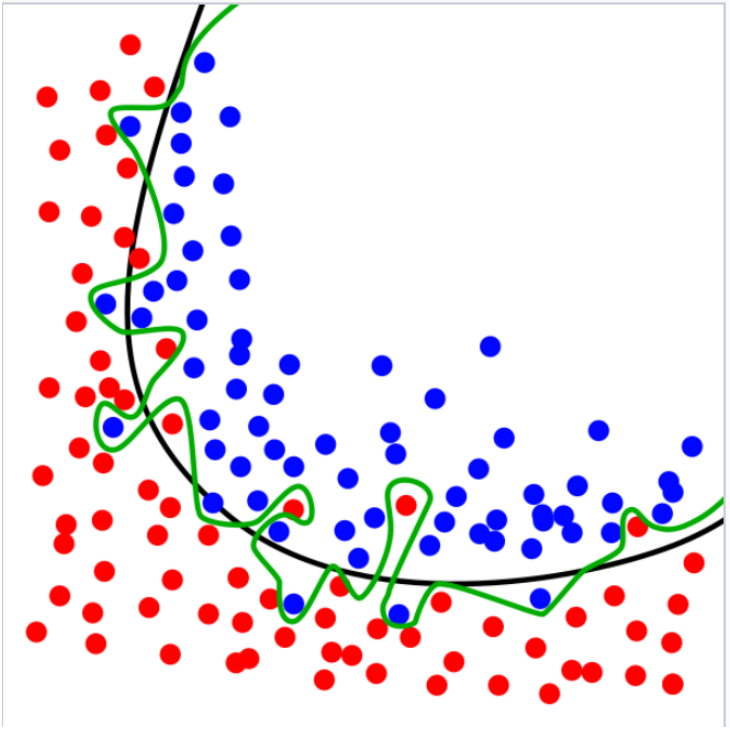
# We can make the classifiers arbitrarily accurate… with 1-NN classifier; or with bigger and bigger neural networks.

- 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.

The goal of machine learning is not to obtain 0-training error, but rather to achieve small error rates on **new data points** (that are **not** used for training.)

- Test Error < Training Error + Generalization Error

$E_{(x,y)\sim D}[\text{error}(w_{(x,y)})]$ $\frac{1}{n}\sum_{i=1}^{n}\text{error}(w_{(x_i,y_i)})$ $\left|\frac{1}{n}\sum_{i=1}^{n}\text{error}(w_{x_i}) - E_{(x,y)\sim D}[\text{error}(w_x)]\right|$

The goal of machine learning is not to obtain 0-training error, but rather to achieve small error rates on **new data points** (that are **not** used for training.)

- Test Error $<$ Training Error + Generalization Error

$$\mathrm{Err}(h) := \mathbb{E}[\mathbf{1}(h(x) \neq y)]$$

The goal of machine learning is not to obtain 0-training error, but rather to achieve small error rates on **new data points** (that are **not** used for training.)

- Test Error < Training Error + Generalization Error

$$\mathrm{Err}(h) := \mathbb{E}[\mathbf{1}(h(x) \neq y)] \qquad \widehat{\mathrm{Err}}(h) := \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}(h(x_i) \neq y_i)$$

The goal of machine learning is not to obtain 0-training error, but rather to achieve small error rates on **new data points** (that are **not** used for training.)

- Test Error < Training Error + Generalization Error

$$\mathrm{Err}(h) := \mathbb{E}[\mathbf{1}(h(x) \neq y)] \qquad \widehat{\mathrm{Err}}(h) := \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}(h(x_i) \neq y_i)$$

$$\mathrm{Gen}(\mathcal{H}) := \sup_{h \in \mathcal{H}} \left| \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}(h(x_i) \neq y_i) - \mathbb{E}[\mathbf{1}(h(x) \neq y)] \right|$$

# The goal of machine learning is not to obtain 0-training error, but rather to achieve small error rates on **new data points** (that are **not** used for training.)

- Test Error < Training Error + Generalization Error

$$\mathrm{Err}(h) := \mathbb{E}[\mathbf{1}(h(x) \neq y)] \qquad \widehat{\mathrm{Err}}(h) := \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}(h(x_i) \neq y_i)$$

$$\mathrm{Gen}(\mathcal{H}) := \sup_{h \in \mathcal{H}} \left| \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}(h(x_i) \neq y_i) - \mathbb{E}[\mathbf{1}(h(x) \neq y)] \right|$$

(** some text uses “generalization error” as a synonym as “test error”, which has created much confusion. The above is the definition we adopt.)

# The goal of machine learning is not to obtain 0-training error, but rather to achieve small error rates on **new data points** (that are **not** used for training.)

- Test Error < Training Error + Generalization Error

$$\mathrm{Err}(h) := \mathbb{E}[\mathbf{1}(h(x) \neq y)] \qquad \widehat{\mathrm{Err}}(h) := \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}(h(x_i) \neq y_i)$$

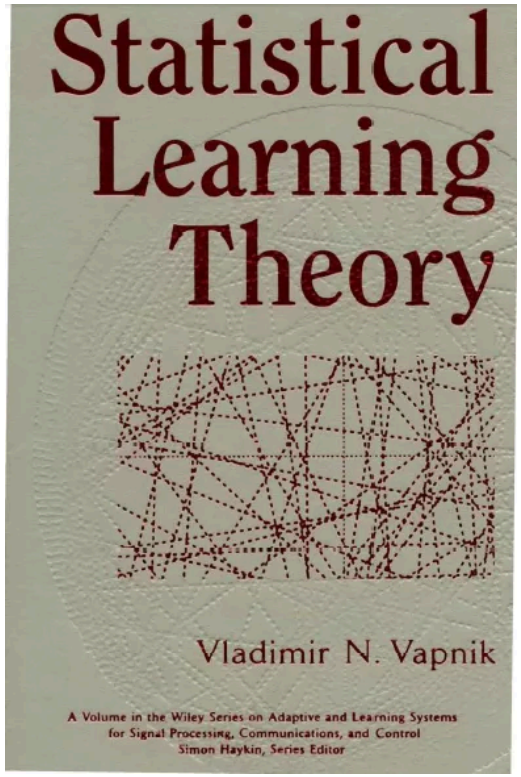
$$\mathrm{Gen}(\mathcal{H}) := \sup_{h \in \mathcal{H}} \left| \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}(h(x_i) \neq y_i) - \mathbb{E}[\mathbf{1}(h(x) \neq y)] \right|$$

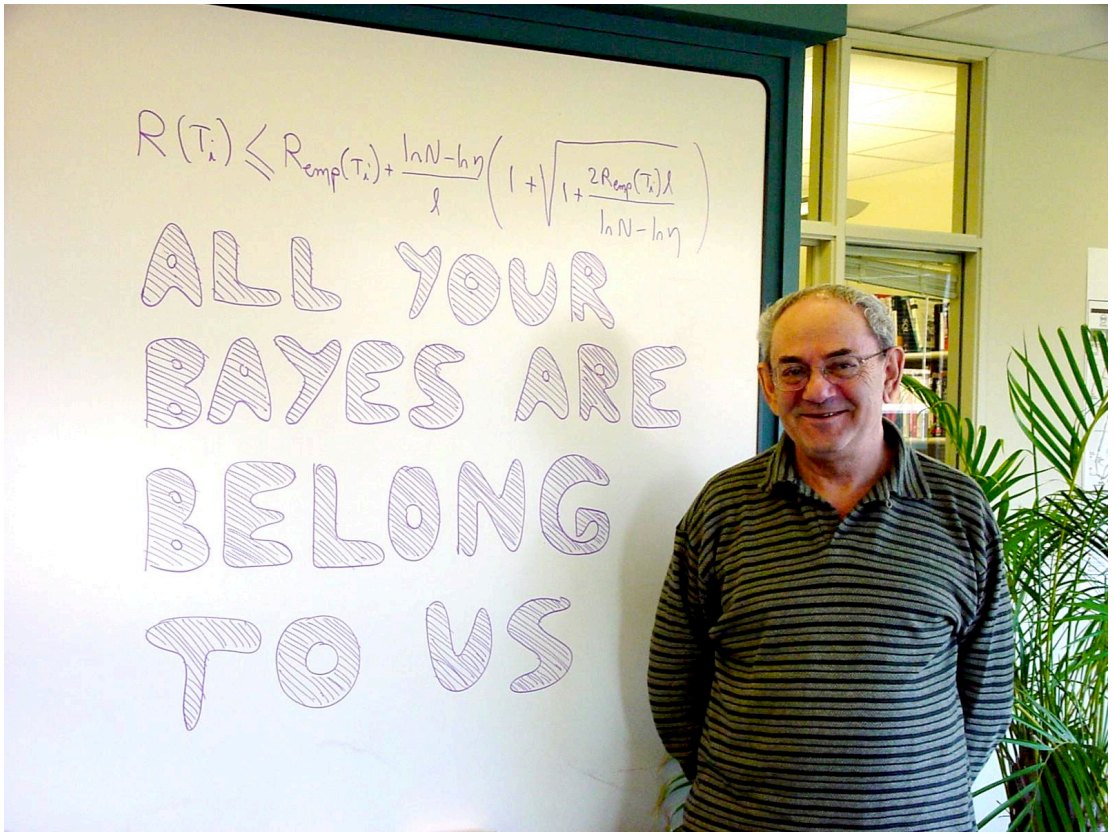
(** some text uses “generalization error” as a synonym as “test error”, which has created much confusion. The above is the definition we adopt.)

(More fun about this in Homework 1)

# Statistical Learning Theory

TL;DR: Proving that the generalization error → 0, thus showing that ML works.




Closely related to Empirical Process Theory, Computational Learning Theory.

# Summary of today's lecture

- Machine learning overview

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

# On Thursday

- Prevent overfitting
  - watch out for distribution-shift
- How to learn a classifier:
  - Algorithms to solve the optimization problem in machine learning
- Continuous optimization
  - One algorithm to solve it all

# On Thursday

- Prevent overfitting
  - watch out for distribution-shift
- How to learn a classifier:
  - Algorithms to solve the optimization problem in machine learning
- Continuous optimization
  - One algorithm to solve it all

Submit anonymous feedback! Come to the office hour!