UNIVERSITY AT ALBANY
STATE UNIVERSITY OF NEW YORK

CSI 436/536 (Fall 2024)

Machine Learning

Lecture 6: Evaluation Criteria

Chong Liu

Assistant Professor of Computer Science

Sep 17, 2024

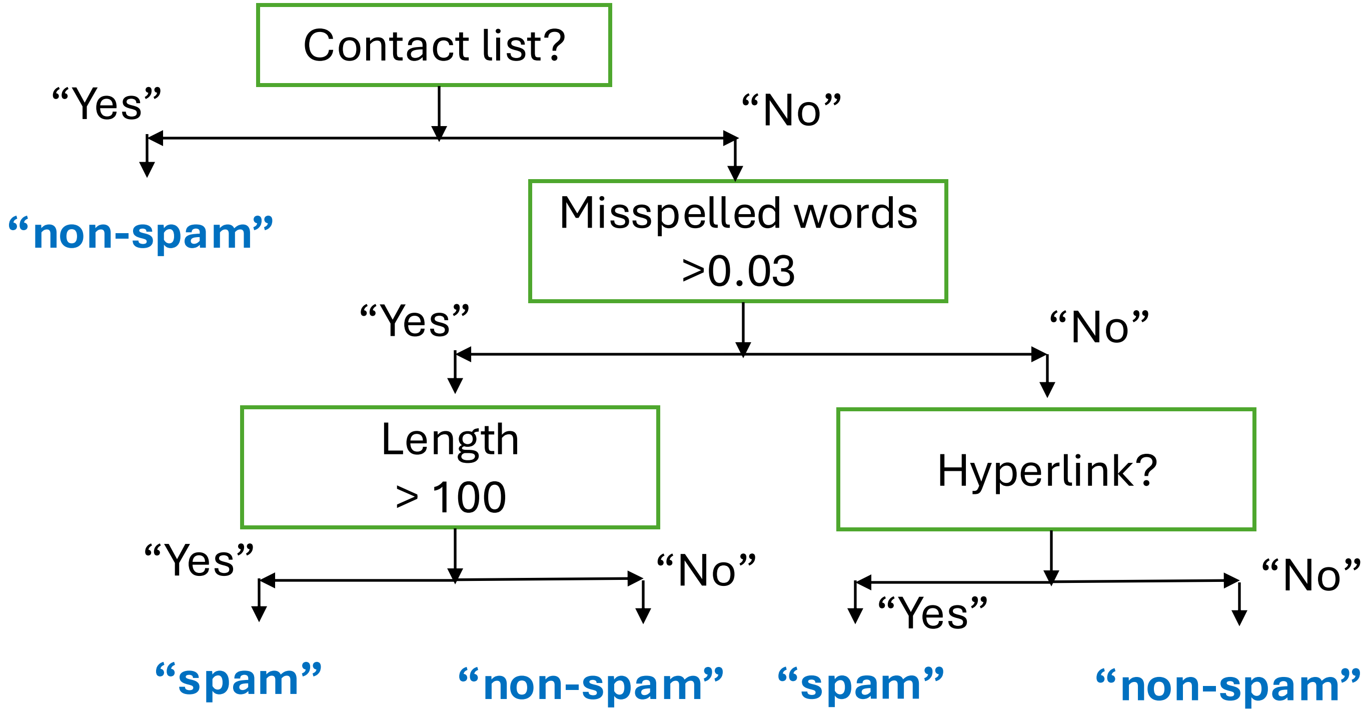
Announcement

- Course project registration due this Thursday!

Recap: elements of machine learning

- Machine learning overview
  - Supervised learning
  - Unsupervised learning
  - Reinforcement learning
- Supervised learning: binary classification
  - Spam filtering
- Feature design and feature extraction
  - In contact list or not
  - Proportion of misspelled words
  - …
- Decision tree classifier

Recap: Decision tree



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

Today

- Linear classifier
- Performance metrics
- Feature transformation

Linear classifiers

- Model:
  - $\text{Score}(x) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4$
  - $x_1 = 1(\text{has hyperlinks})$
  - $x_2 = 1(\text{on contact list})$
  - $x_3 = \text{proportion of misspelling}$
  - $x_4 = \text{length}$

Indicator function:

$$f(x) = 1(\text{condition}) = \begin{cases} 1, \text{if condition is true} \\ 0, \text{if condition is false} \end{cases}$$

Question: why do we need w_0?

Linear classifiers

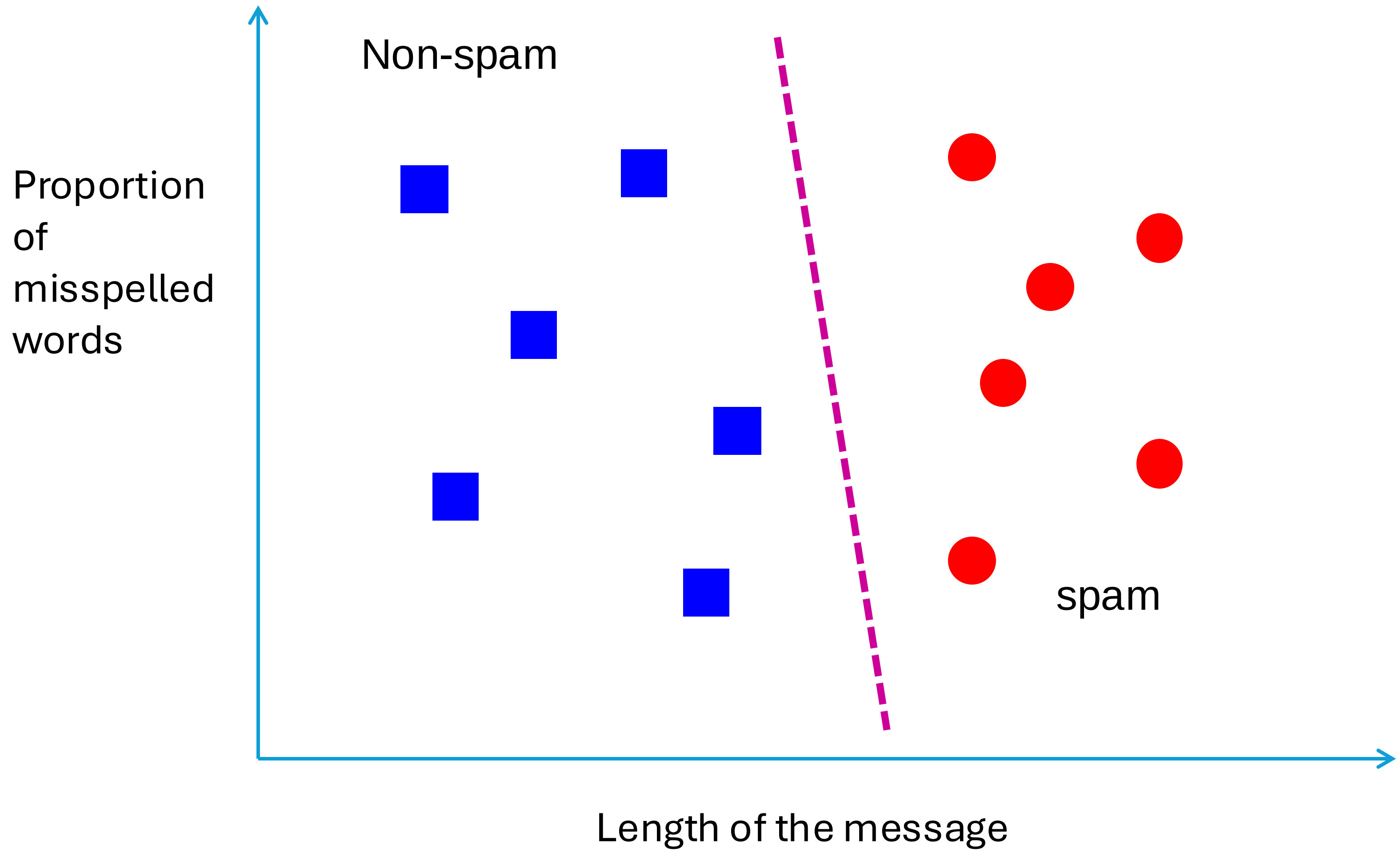
- Model:
  - $\text{Score}(x) = w_0 + w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4$
- A linear classifier:
  - $h(x) = \begin{cases} 1, \text{if Score}(x) \geq 0 \\ -1, \text{if Score}(x) < 0 \end{cases}$
  - A compact representation: $h(x) = \text{sign}(w^T[1; x])$

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

Geometric view: Linear classifier is a decision line!

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

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



Family of classifiers: Hypothesis class

- Hypothesis class $\mathcal{H}$
  - A family of classifiers
  - Also known as “concept class”, “model”, “decision rule book”
  - “Linear classifiers” and “neural networks” are hypothesis classes.
  - Typically we want this family to be large and flexible.

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

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

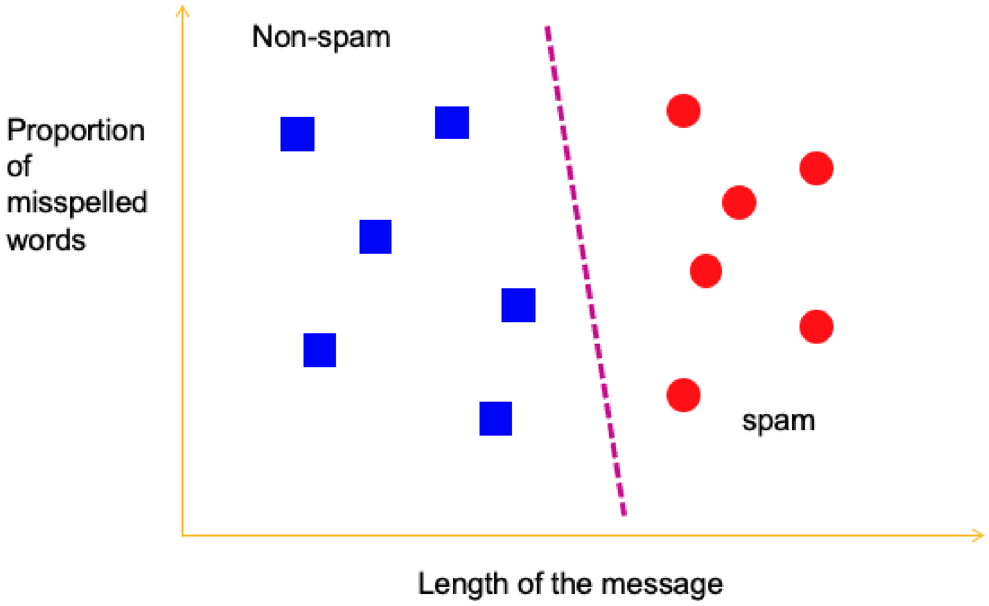
that “**works well**” on this problem.

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

$$h_w : \mathcal{X} \to \mathcal{Y}$$

- For any $x \in \mathcal{X}$
  - We can apply this classifier to get its predicted label
  $$\hat{y} = h_w(x)$$
  - The prediction doesn't have to be correct. It just need to be valid, i.e.,
  $$\hat{y} \in \mathcal{Y}$$

Learning linear classifiers

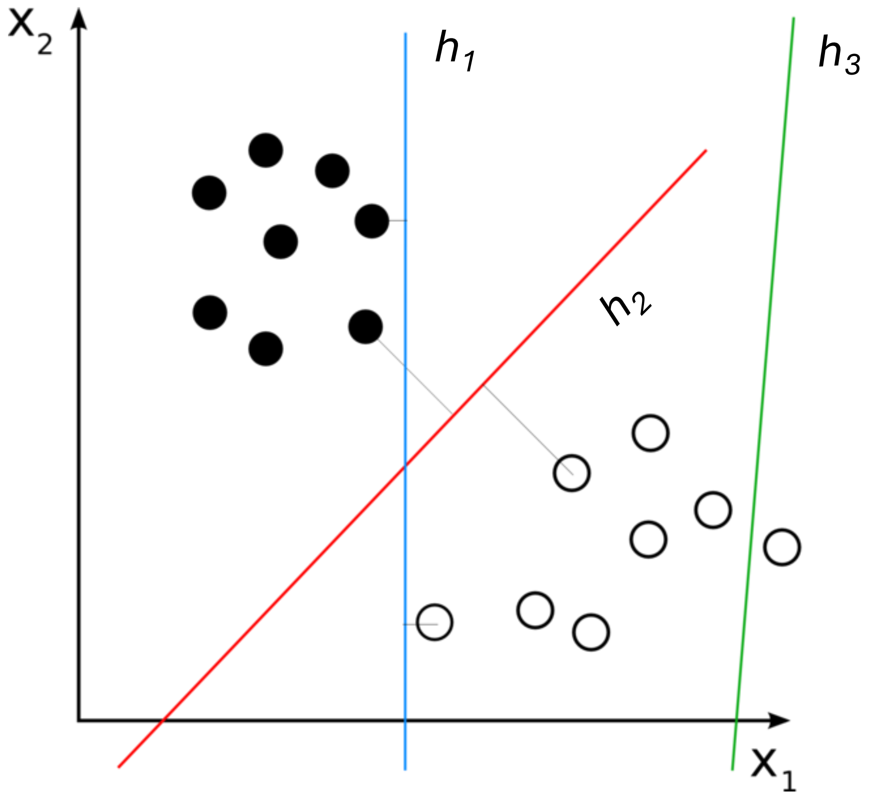


- Training data:

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

- 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.

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

Actual class ← Ground truth

y

Classifier output → Predicted class $\hat{y}$

	1	0	
1	TP	FP Type I Error	*Estimated positive* $\hat{P}$
0	FN Type II Error	TN	*Estimated negative* $\hat{N}$
	Positive *P*	*Negative* *N*	TOTAL

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

Correct (TP, TN)
Errors (FP, FN)

$$TP + FN = P$$

$$FP + TN = N$$

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

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

$$P + N = TOTAL$$

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

In-class exercise: confusion matrix

Actual class ← Ground truth

y

Classifier output ↓ Predicted class $\hat{y}$

	1	0	
1	TP	FP Type I Error	*Estimated positive* $\hat{P}$
0	FN Type II Error	TN	*Estimated negative* $\hat{N}$
	Positive *P*	*Negative* *N*	TOTAL

$$\hat{y} = [1,1,1,1,0,0,0,1,1,1]$$
$$y = [1,0,0,0,0,1,1,0,0,0]$$

Key terminology

- Accuracy = $\frac{TP+TN}{\text{Total}}$
  - Proportion of total correct predictions
- Precision = $\frac{TP}{\hat{P}}$
  - Proportion of correctly predicted positive observations to the total predicted positives
- Recall= $\frac{TP}{P}$
  - Proportion of correctly predicted positive observations to the all observations in actual positive class
- F1 score = $\frac{2\cdot\text{Precision}\cdot\text{Recall}}{\text{Precision}+\text{Recall}}$
  - The harmonic mean of Precision and Recall

$$\hat{y} = [1,1,1,1,0,0,0,1,1,1]$$
$$y = [1,0,0,0,0,1,1,0,0,0]$$

		Actual class		
	y	1	0	
Predicted class $\hat{y}$	1	TP	FP	*Estimated positive* $\hat{P}$
	0	FN	TN	*Estimated negative* $\hat{N}$
		Positive *P*	*Negative* *N*	TOTAL

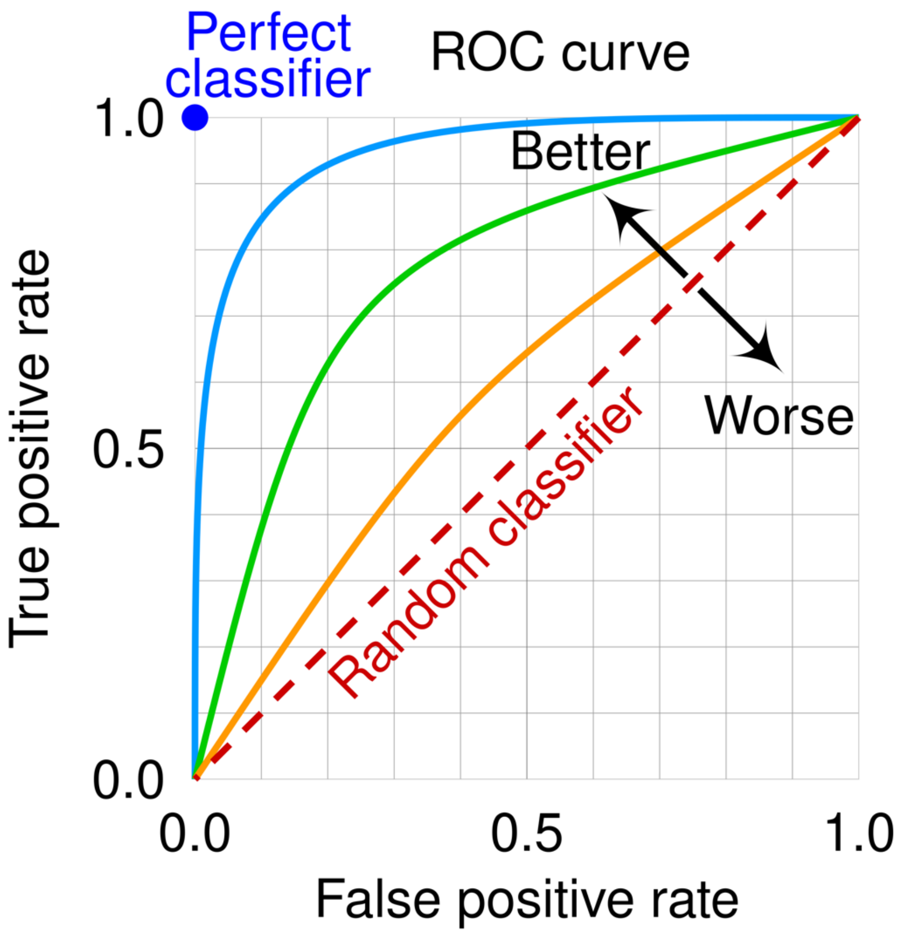
Response Operator Characteristic (ROC) curve

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

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

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

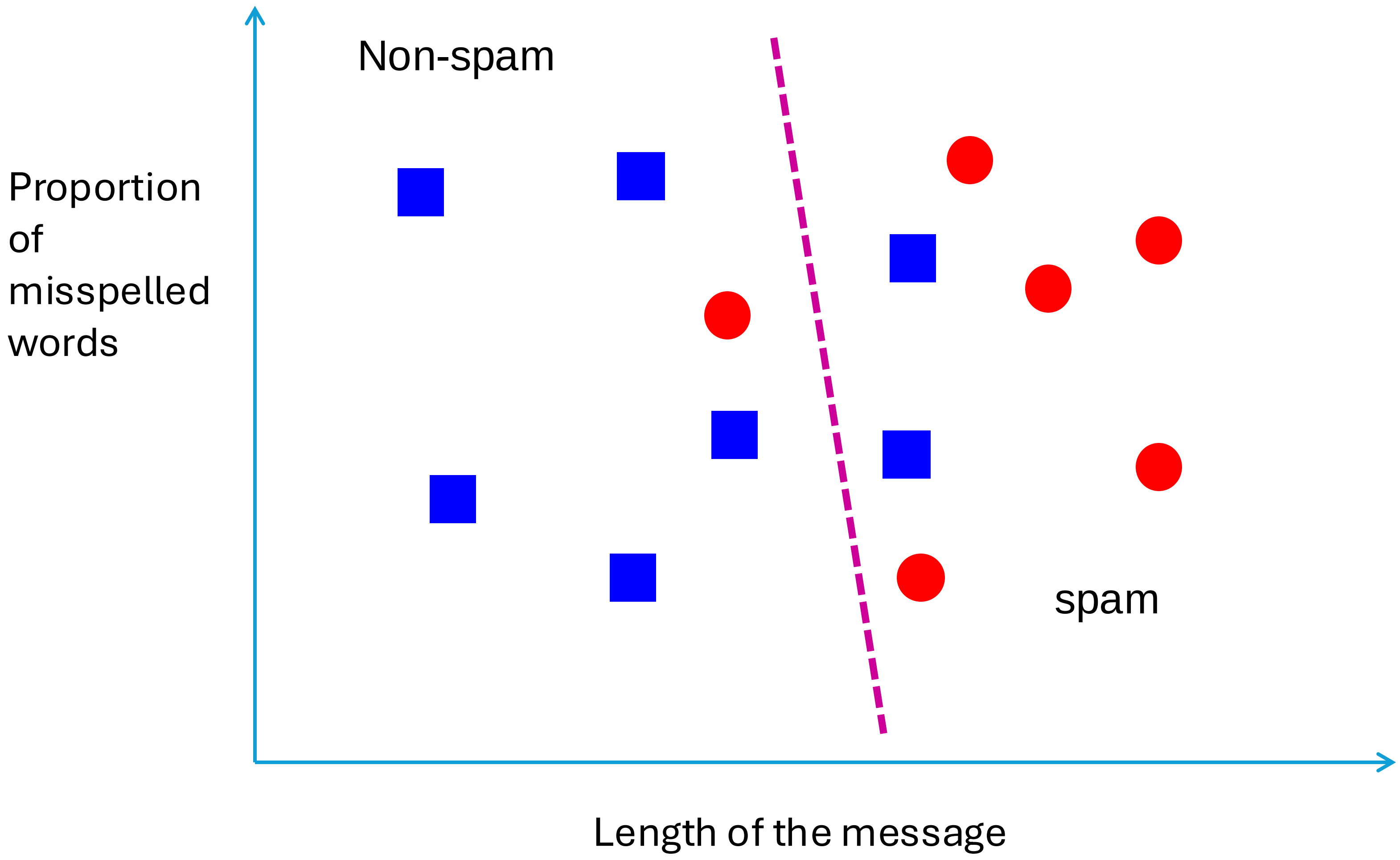
True negative rate (TNR) = $\frac{TN}{N}$ = Specificity = 1 - α



Single number summary of any "**score function**"

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

In practice: many non-linearly separable case



How to learn LINEAR classifier in a non-linearly separable case?

- Training data:

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

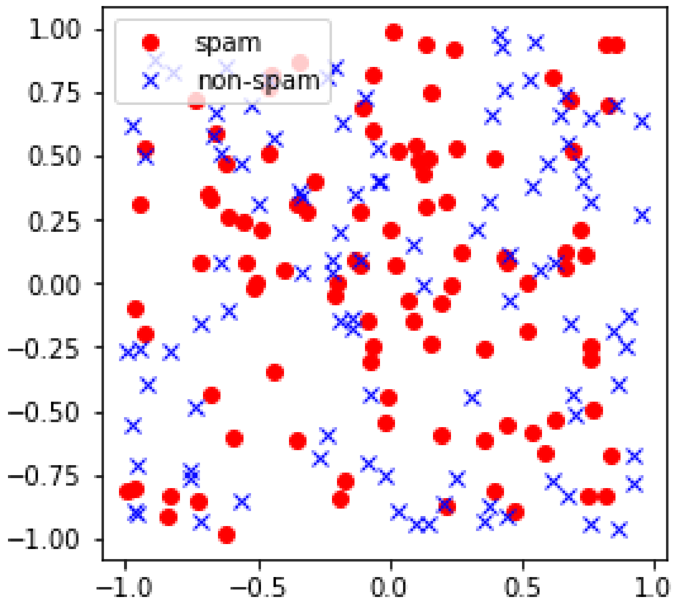
- Solving the following optimization problem:

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

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

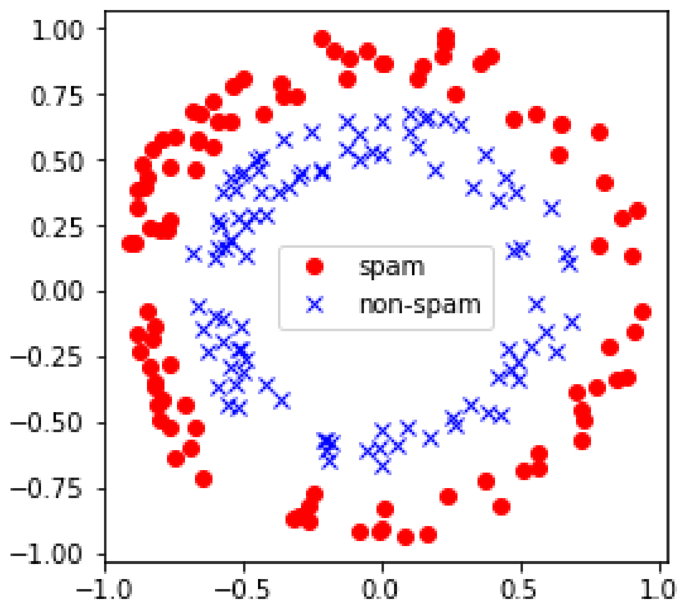
What happens if the linear classifier with the smallest number of mistakes still makes a mistake 49% of the time?

Case 1:



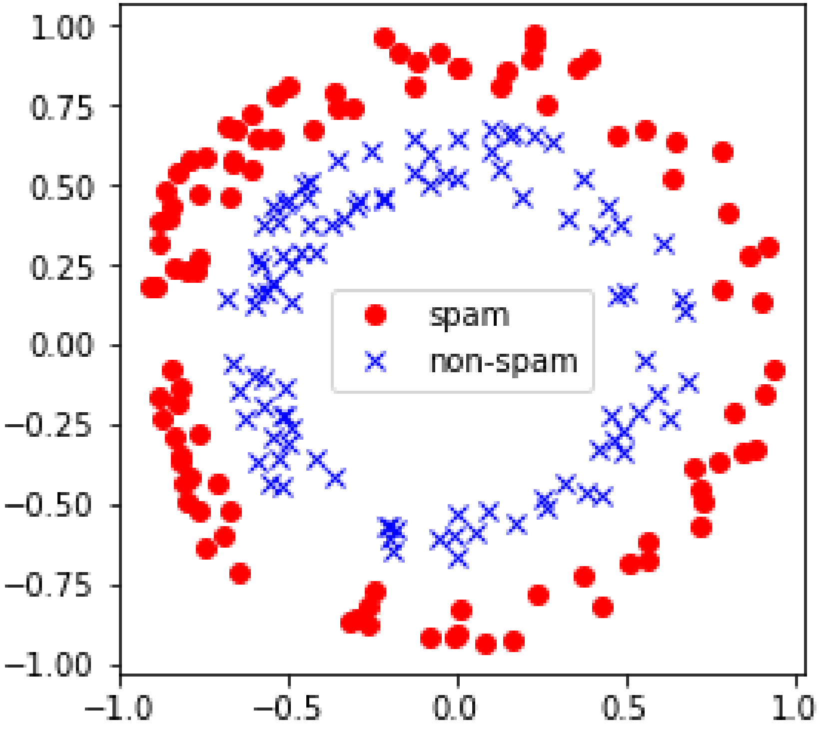
There is no information about the label in the features.
No classifiers are able to do well.

Case 2:



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

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.