CS 481

Artificial Intelligence Language Understanding

March 21, 2023

Announcements / Reminders

- Enjoy your Spring Break!
- Please follow the Week 09 To Do List instructions
- Written Assignment #03 is due on Sunday 03/26/23 at 11:59 PM CST
- Quiz due on Sunday 03/26/23 at 11:59 PM CST
- Programming Assignment #02 is due on Sunday 04/02/23 at 11:59 PM CST

- Exam dates:
  - Final: 04/27/2023 during Thursday lecture time

Plan for Today

- **Logistic Regression**
- **Sentiment Analysis**
- **Words and their meaning**

Text Classification: Supervised ML

- **Various Machine Learning supervised learning classifier approaches can be employed:**
  - **Naïve Bayes**
  - **Logistic regression**
  - **Neural networks**
  - **k-Nearest Neighbors**
  - **etc.**

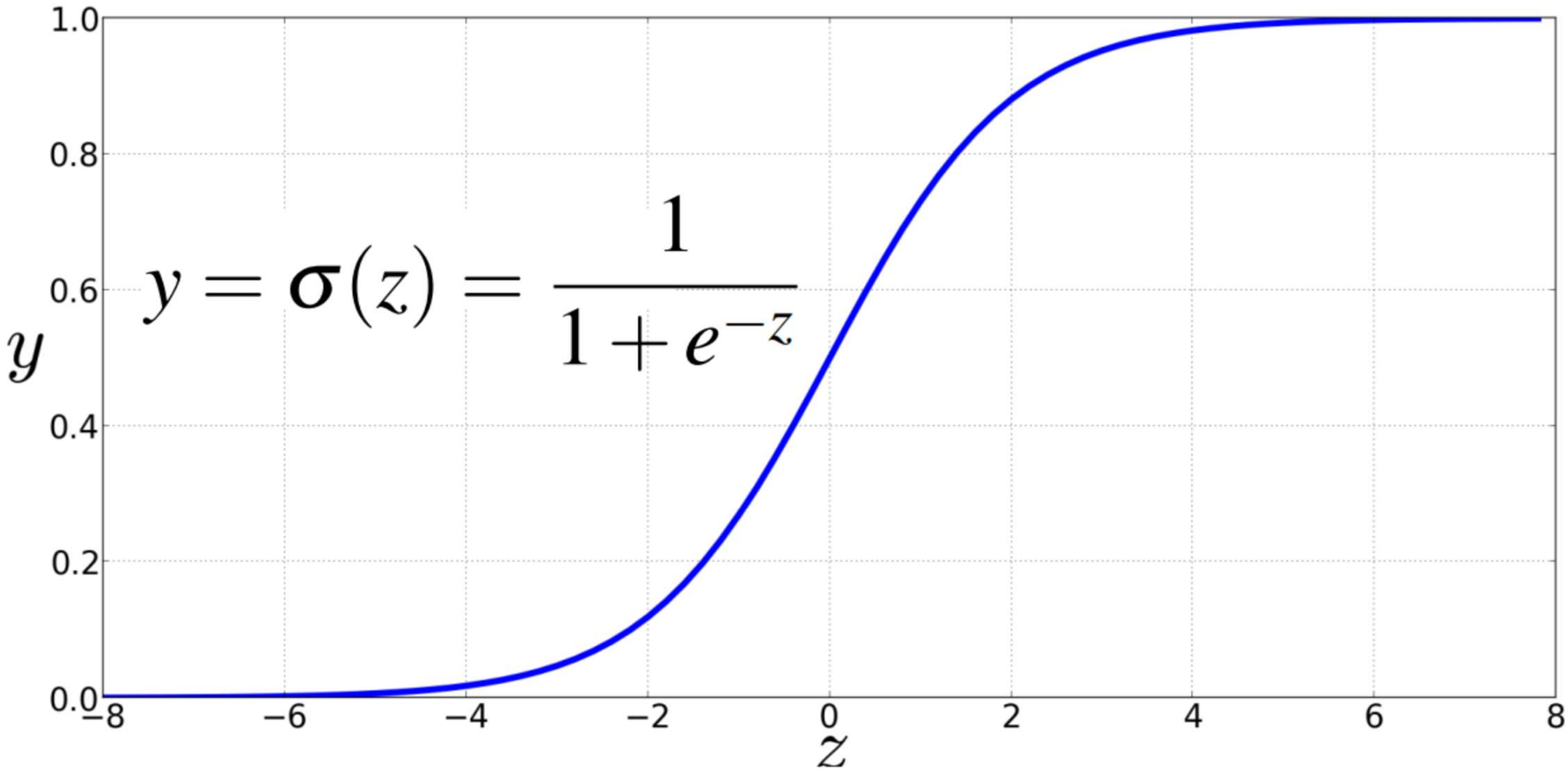
Logistic Regression: The Idea

- **Compute $\mathbf{w} \cdot \mathbf{x} + b$ for observation / sample $\mathbf{x}$**
- **Pass it through the sigmoid function:**

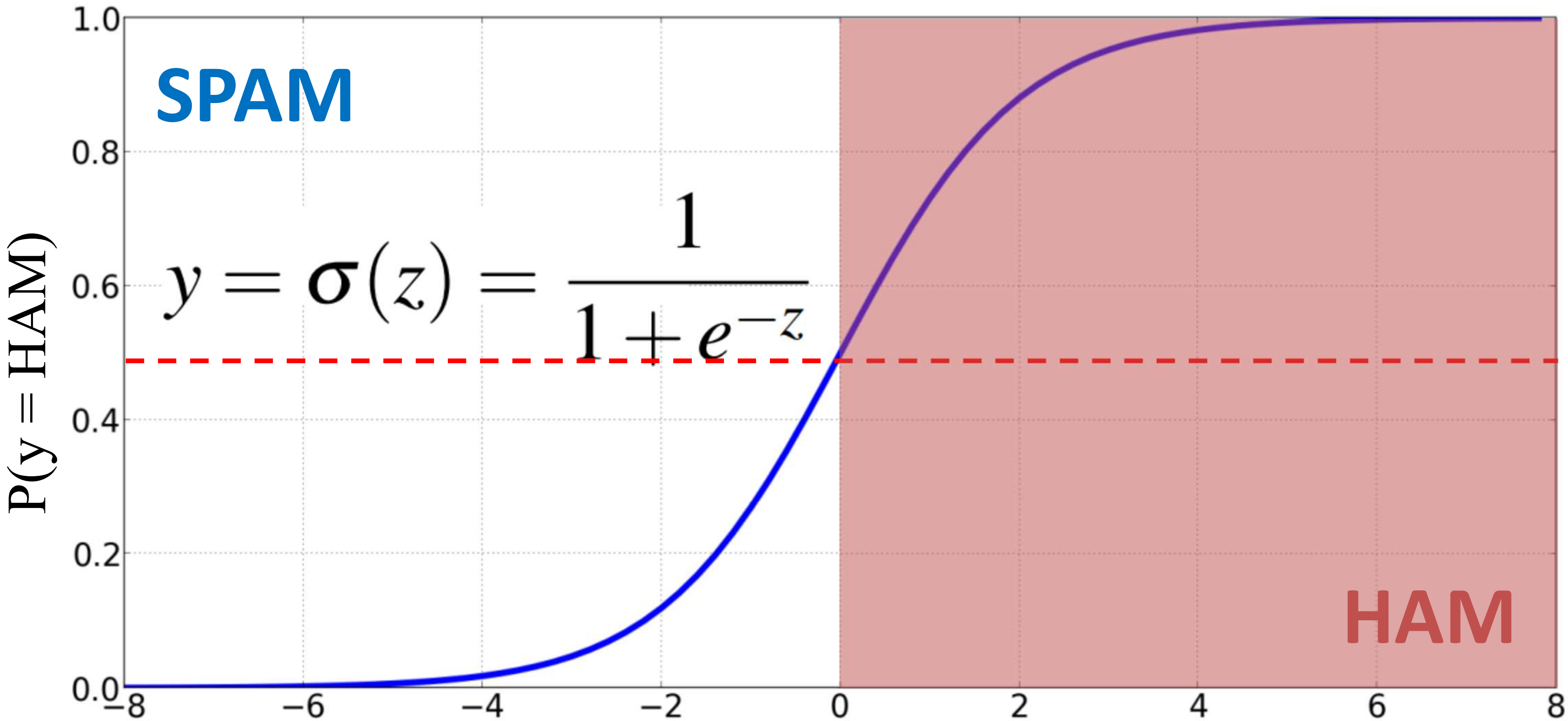
$$\sigma(\mathbf{w} \cdot \mathbf{x} + b)$$

- **Treat the result as probability**

Sigmoid / Logistic Function



Logistic Regression Classifier



$$y = \sigma(z) = \frac{1}{1+e^{-z}}$$

$\mathbf{z} = \mathbf{w} \cdot \mathbf{x} + b$

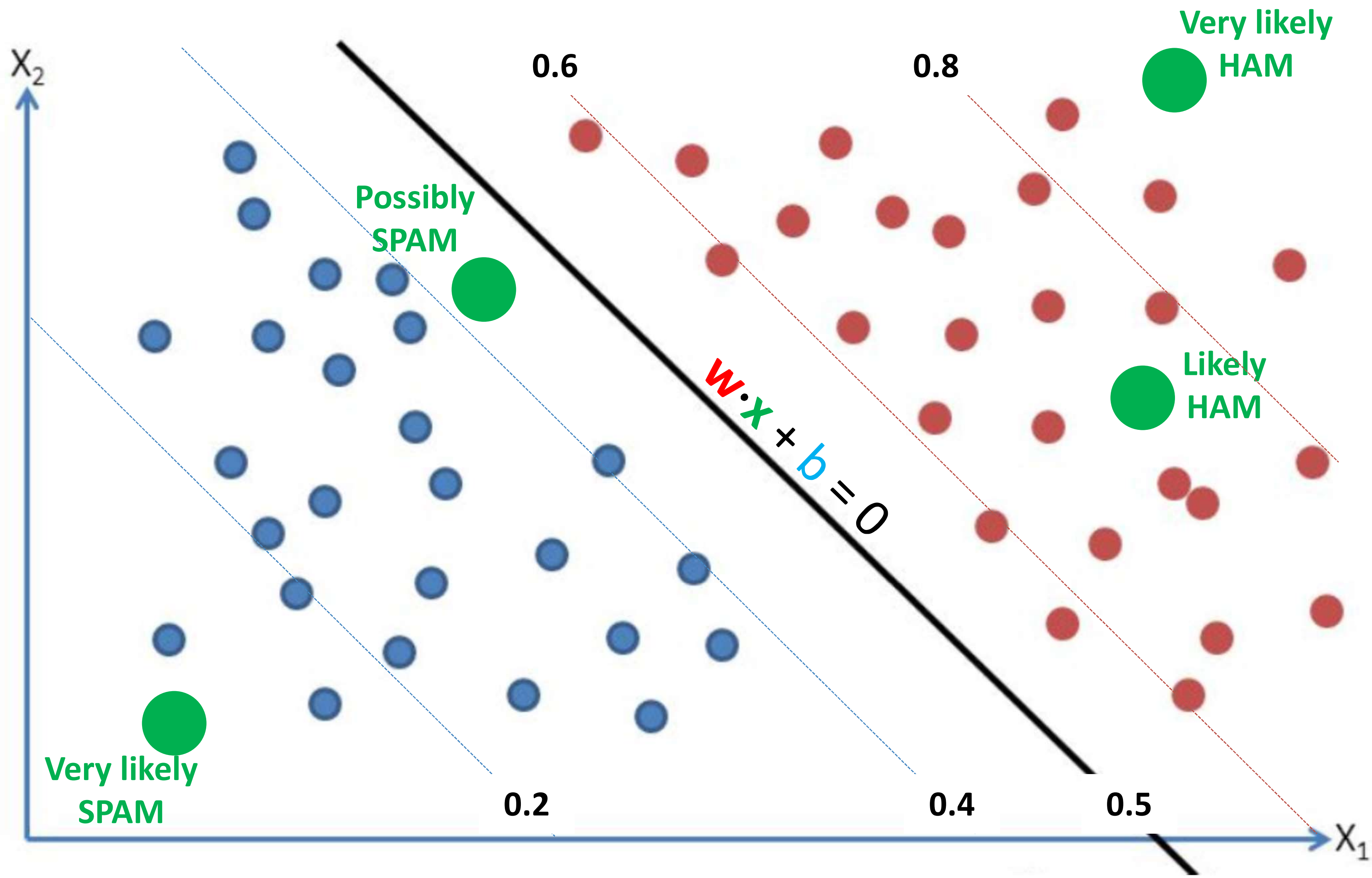
Probabilities → Classification

Once we know the probability, we can use it to classify:

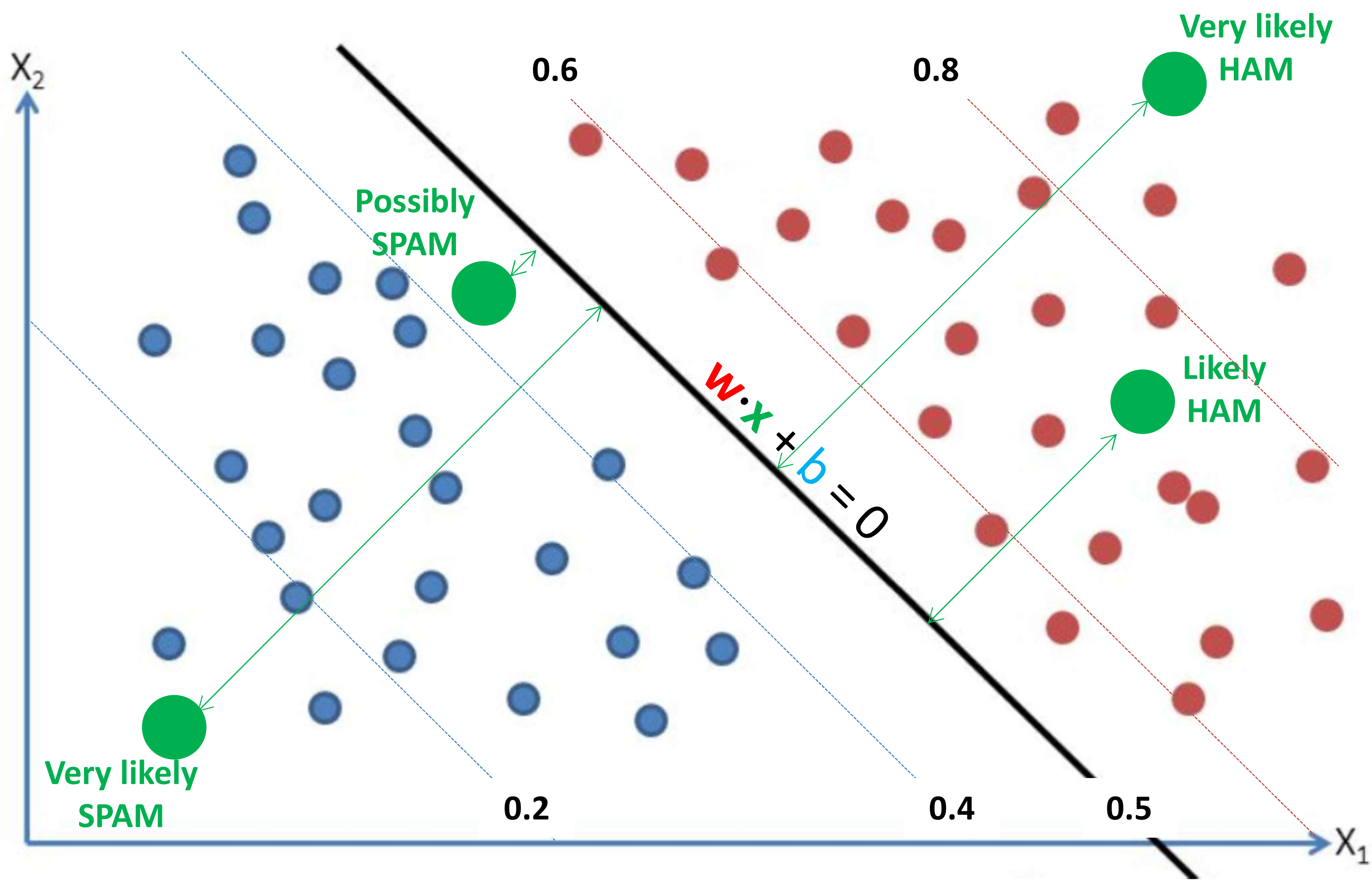
$$\hat{y} = \begin{cases} 1 & \text{if } P(y=1|x) > 0.5 \\ 0 & \text{otherwise} \end{cases} \qquad \begin{matrix} \text{if } \mathrm{w}\cdot\mathrm{x}+\mathrm{b} > 0 \\ \text{if } \mathrm{w}\cdot\mathrm{x}+\mathrm{b} \leq 0 \end{matrix}$$

Where 0.5 is the decision boundary

Text Classification: Separator



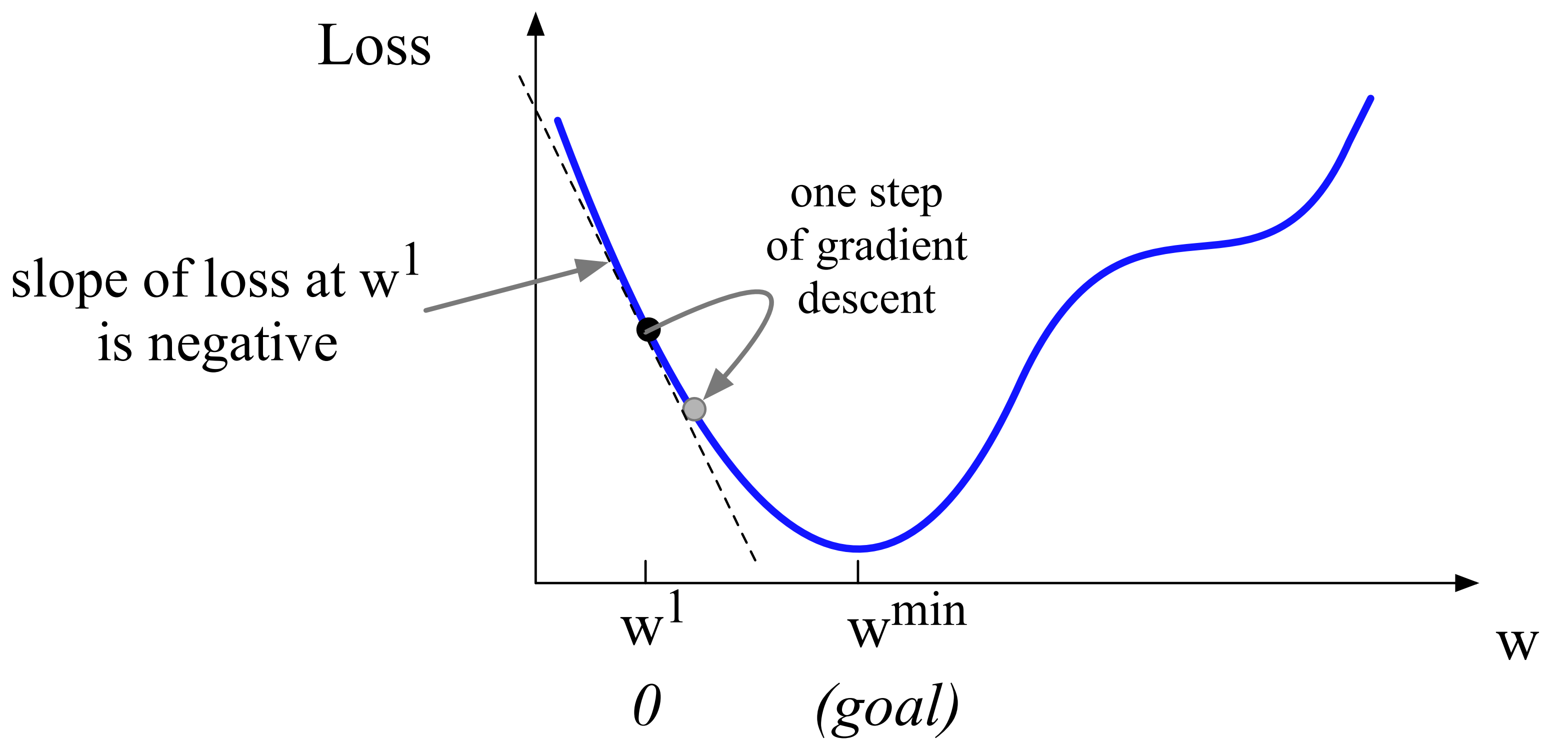
Text Classification: Separator



Minimizing Loss

Q: Given current **w**, should we make it bigger or smaller?

A: Move **w** in the reverse direction from the slope of the loss function



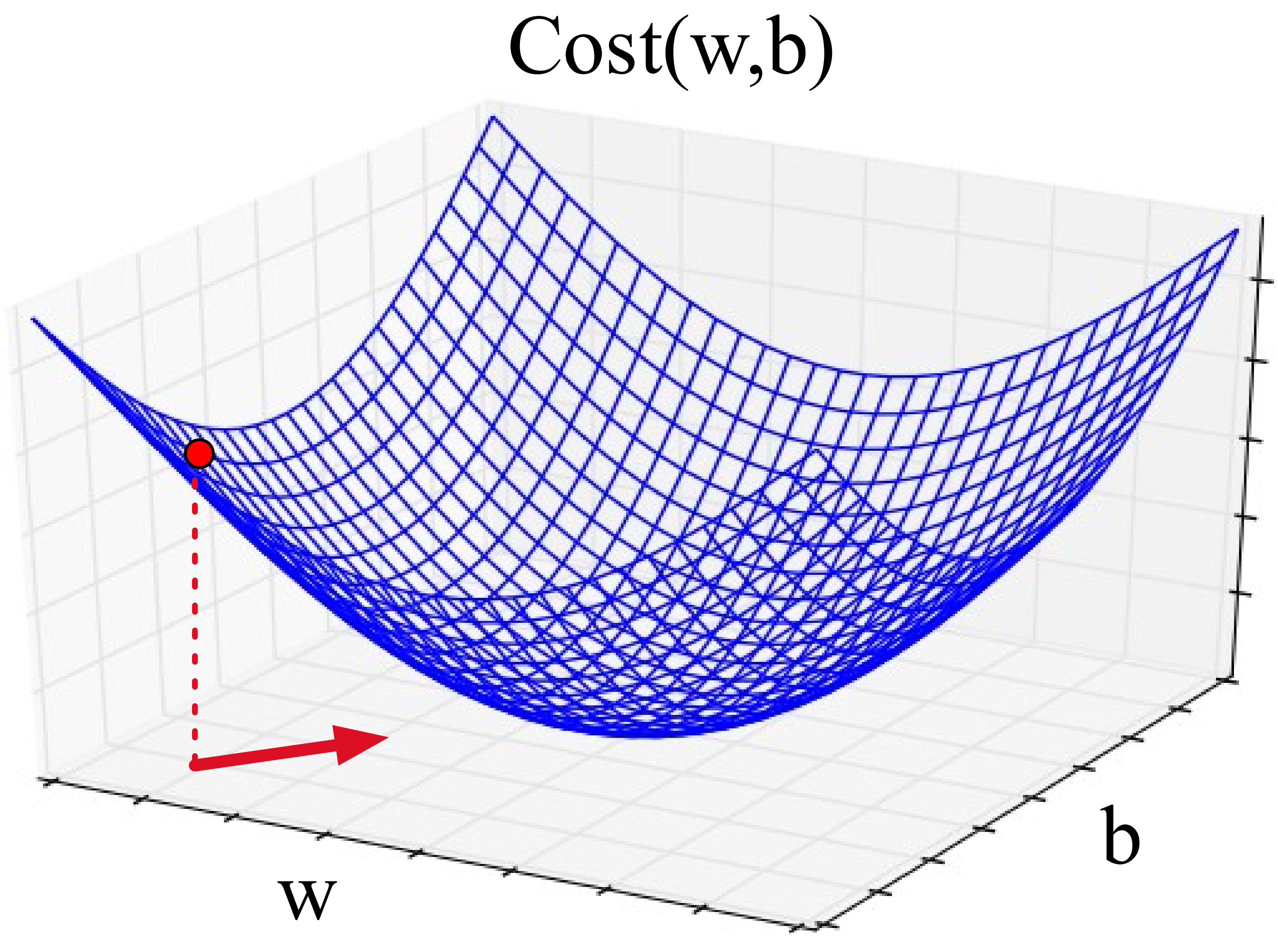
Gradients and Gradient Descent

The **gradient** of a function of many variables is a vector pointing in the direction of the greatest increase in a function.

Gradient Descent: Find the gradient of the loss function at the current point and move in the **opposite** direction.

Gradients: Visualized

Visualizing the gradient vector at the **red** point



Gradients and Learning Rate

- The value of the gradient (slope in our example) $\frac{d}{dw}L(f(x;w),y)$ weighted by a **learning rate** η
- Higher learning rate means move **w** faster

$$w^{t+1} = w^t - \eta \frac{d}{dw} L(f(x;w),y)$$

Gradients and Weights/Bias Update

Let's represent $\hat{y}$ as f (**x**; θ) to make the dependence on θ more obvious:

$$\nabla_\theta L(f(x;\theta),y)) = \begin{bmatrix} \frac{\partial}{\partial w_1} L(f(x;\theta),y) \\ \frac{\partial}{\partial w_2} L(f(x;\theta),y) \\ \vdots \\ \frac{\partial}{\partial w_n} L(f(x;\theta),y) \end{bmatrix}$$

and the final equation is for updating θ:

$$\theta_{t+1} = \theta_t - \eta \nabla L(f(x;\theta),y)$$

Stochastic Gradient Descent

function STOCHASTIC GRADIENT DESCENT($L()$, $f()$, x, y) **returns** θ
  # where: L is the loss function
  #   f is a function parameterized by θ
  #   x is the set of training inputs $x^{(1)}, x^{(2)}, \ldots, x^{(m)}$
  #   y is the set of training outputs (labels) $y^{(1)}, y^{(2)}, \ldots, y^{(m)}$

$\theta \leftarrow 0$
repeat til done
  For each training tuple $(x^{(i)}, y^{(i)})$ (in random order)
    1. Optional (for reporting): # How are we doing on this tuple?
      Compute $\hat{y}^{(i)} = f(x^{(i)};\theta)$ # What is our estimated output $\hat{y}$?
      Compute the loss $L(\hat{y}^{(i)}, y^{(i)})$ # How far off is $\hat{y}^{(i)}$) from the true output $y^{(i)}$?
    2. $g \leftarrow \nabla_\theta L(f(x^{(i)};\theta), y^{(i)})$ # How should we move θ to maximize loss?
    3. $\theta \leftarrow \theta - \eta\, g$ # Go the other way instead
return θ

Hyperparameters

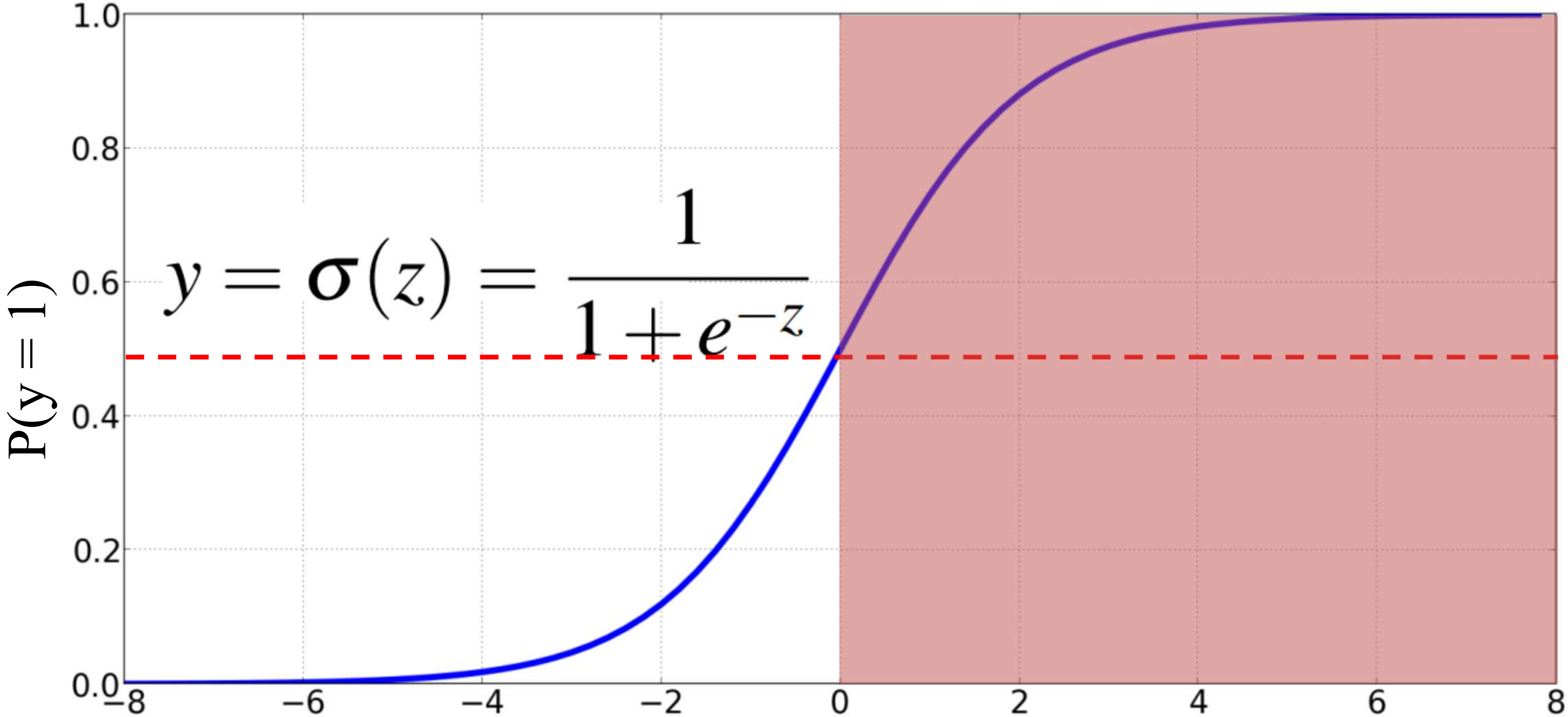
The learning rate η is a **hyperparameter**

- too high: the learner will take big steps and overshoot
- too low: the learner will take too long

Hyperparameters:

- Briefly, a special kind of parameter for an ML model
- Instead of being learned by algorithm from supervision (like regular parameters), they are chosen by algorithm designer.

Logistic Regression Classifier



$z = \mathbf{w} \cdot \mathbf{x} + b$

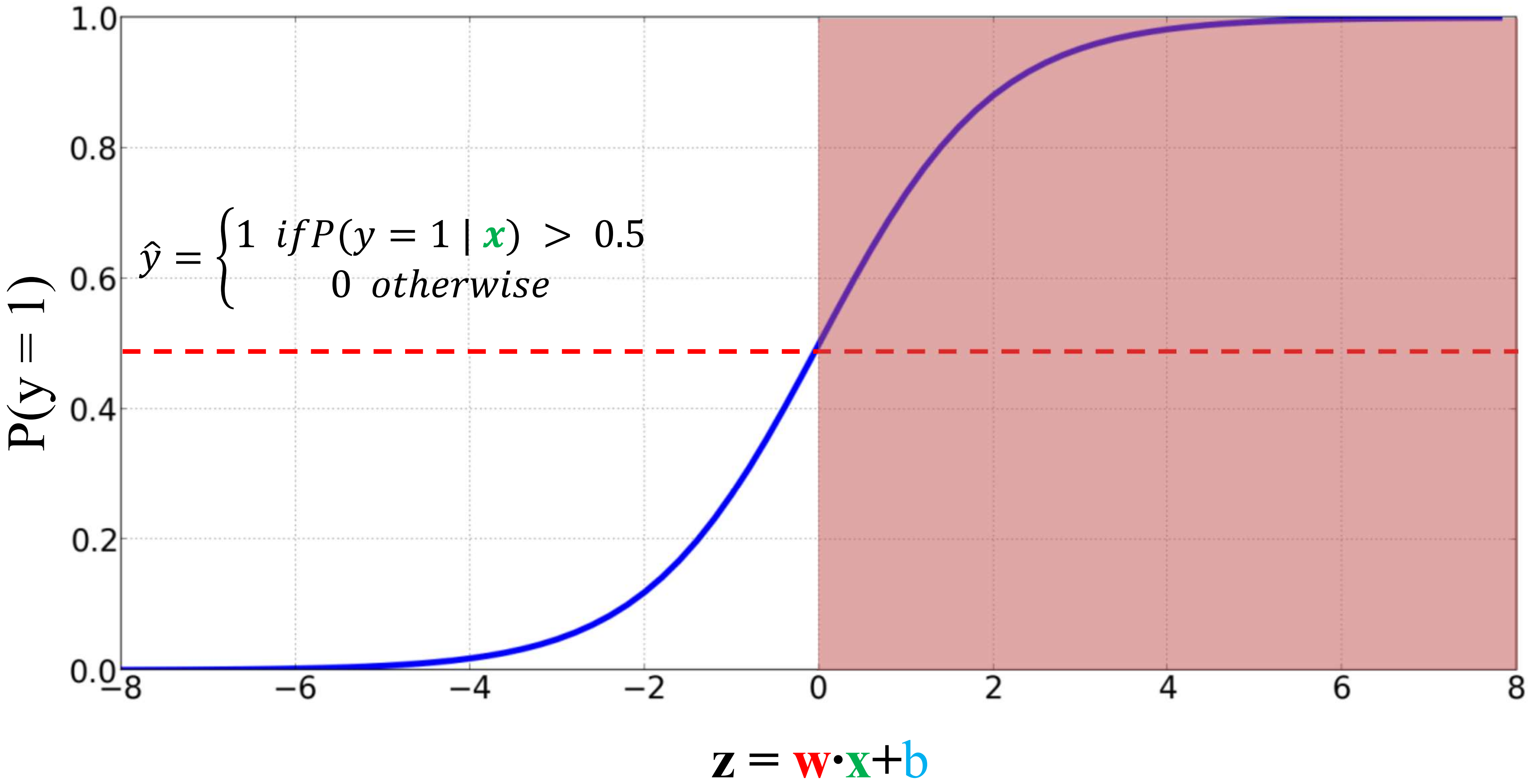
Probabilities → Classification

Once we know the probability, we can use it to classify:

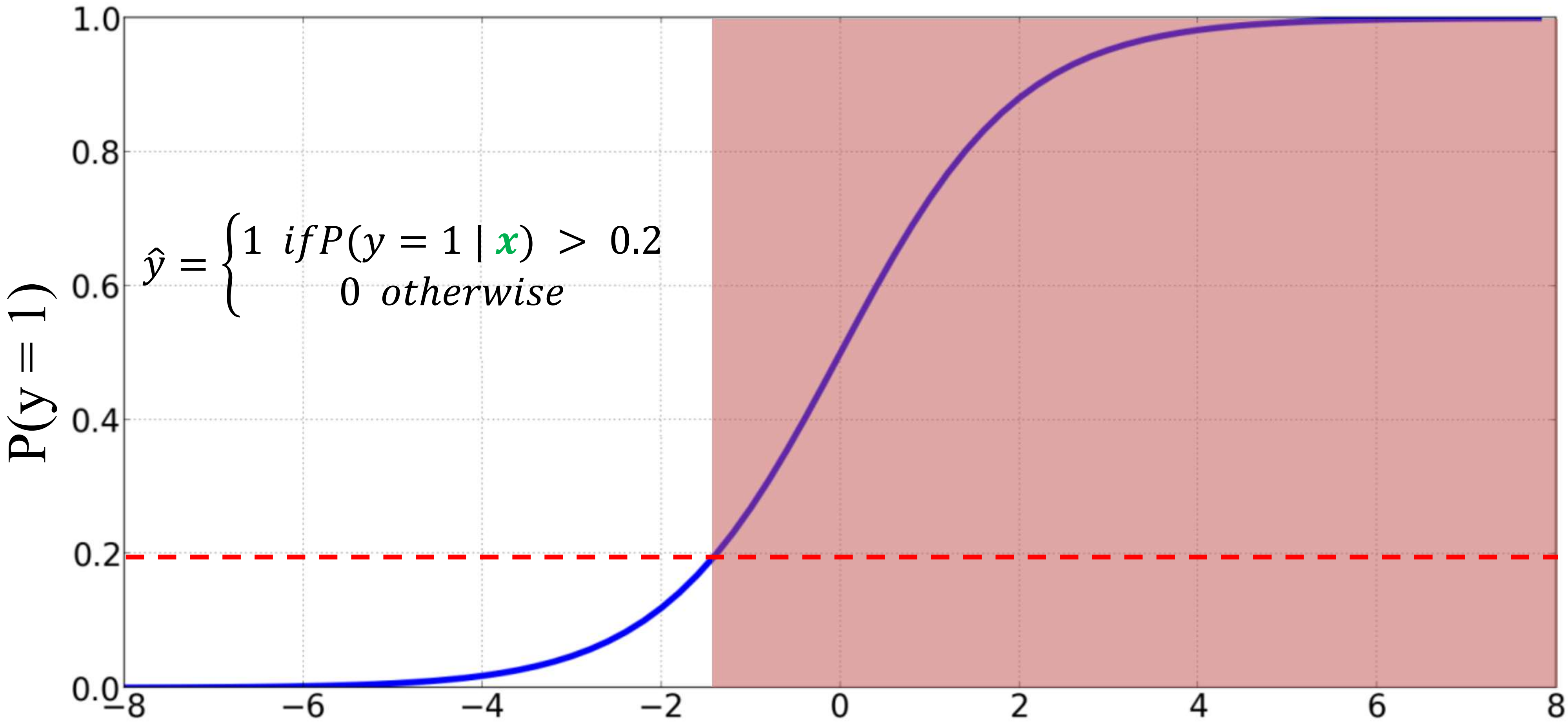
$$\hat{y} = \begin{cases} 1 & \text{if } P(y=1|x) > 0.5 \\ 0 & \text{otherwise} \end{cases}$$

Where 0.5 is the decision boundary

Logistic Regression: Threshold 0.5

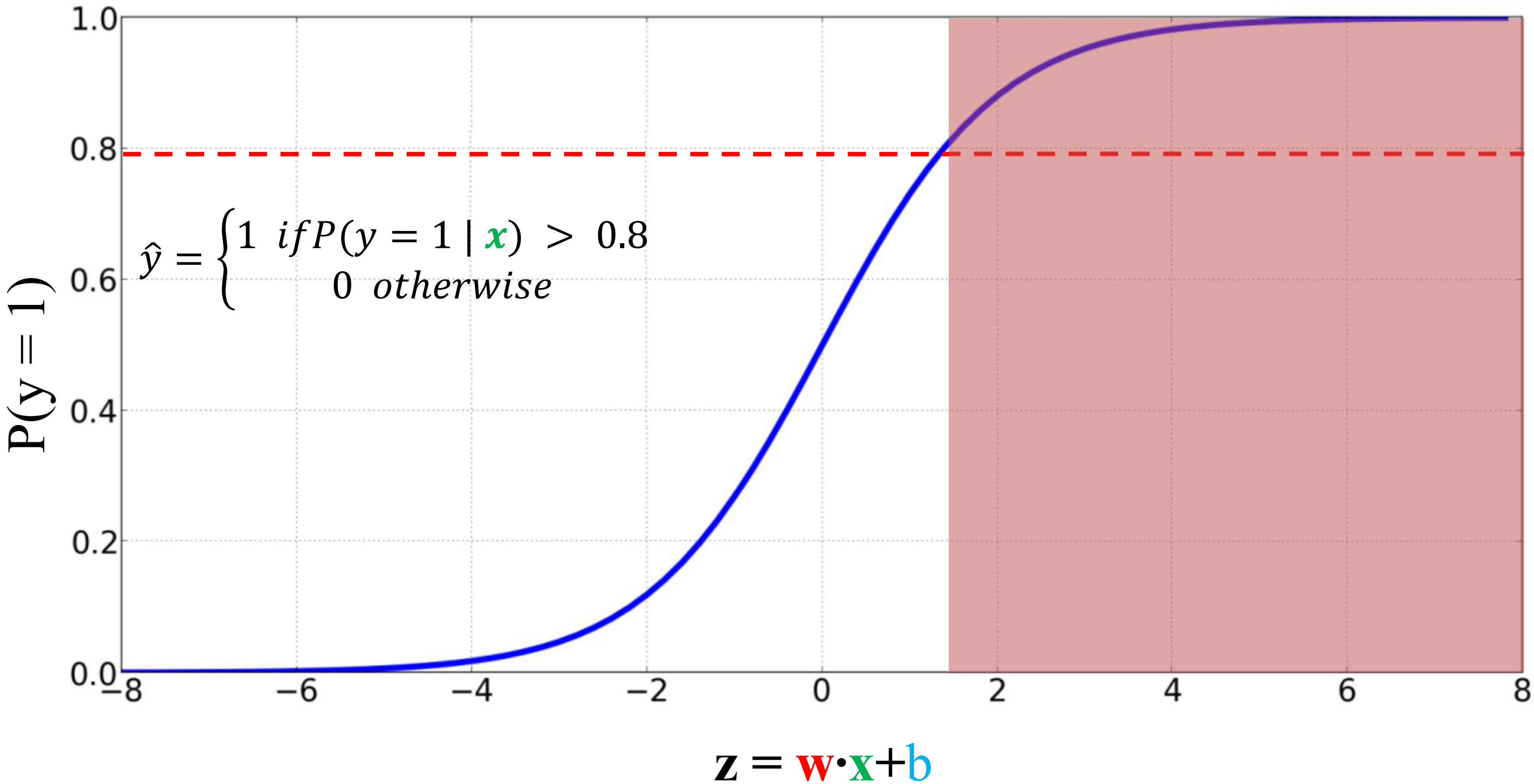


Logistic Regression: Threshold 0.2

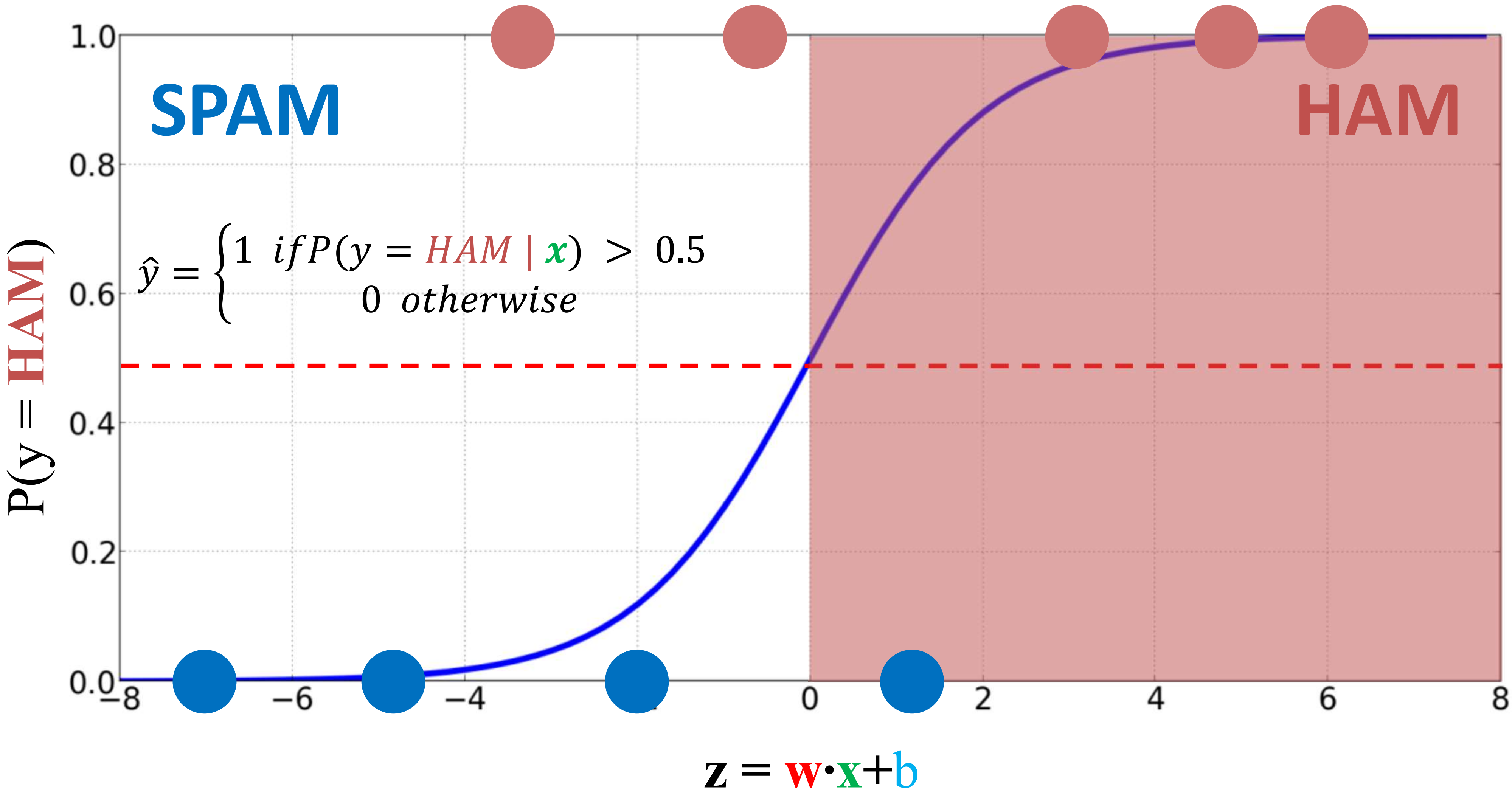


$z = \mathbf{w} \cdot \mathbf{x} + b$

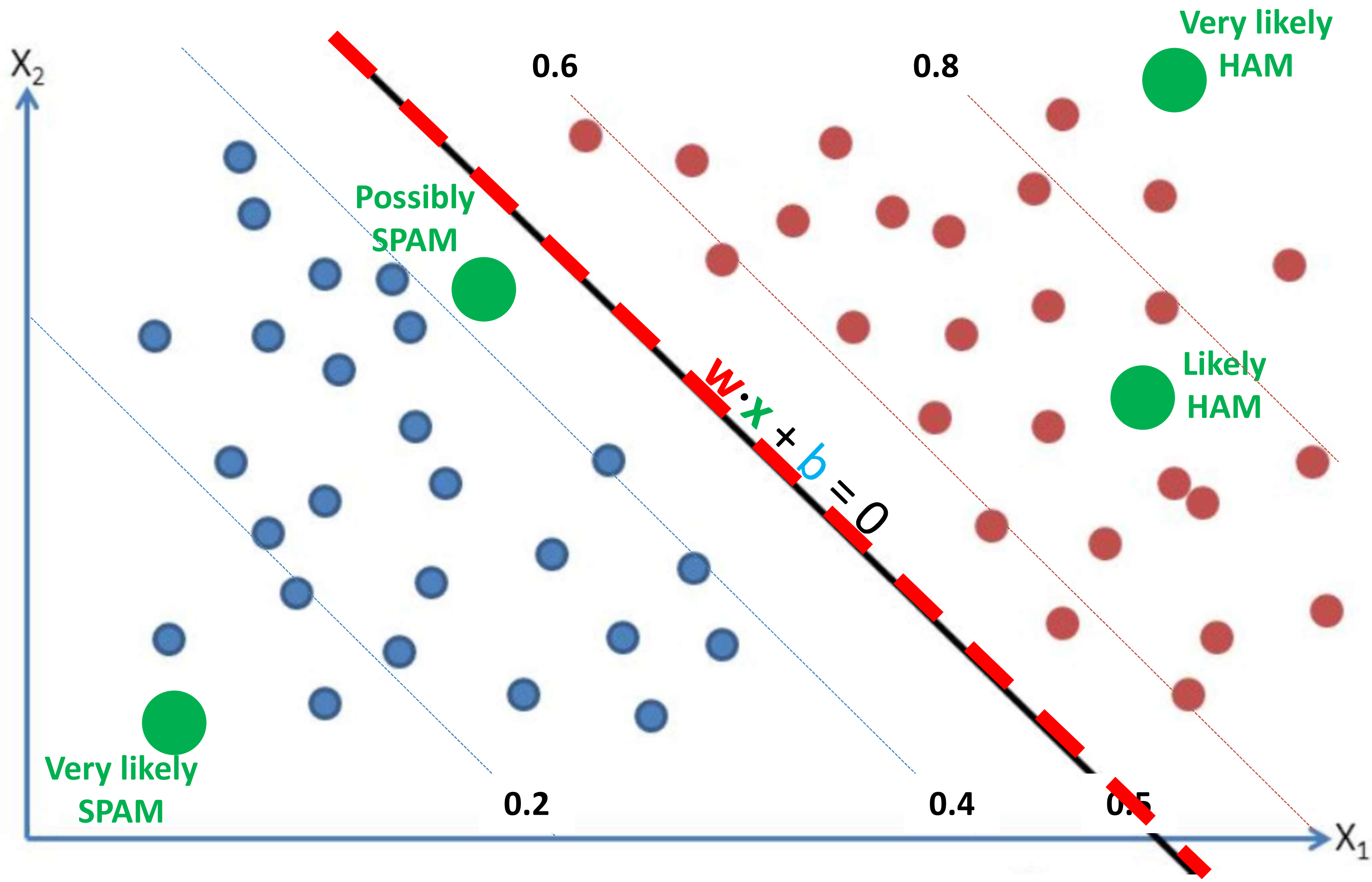
Logistic Regression: Threshold 0.8



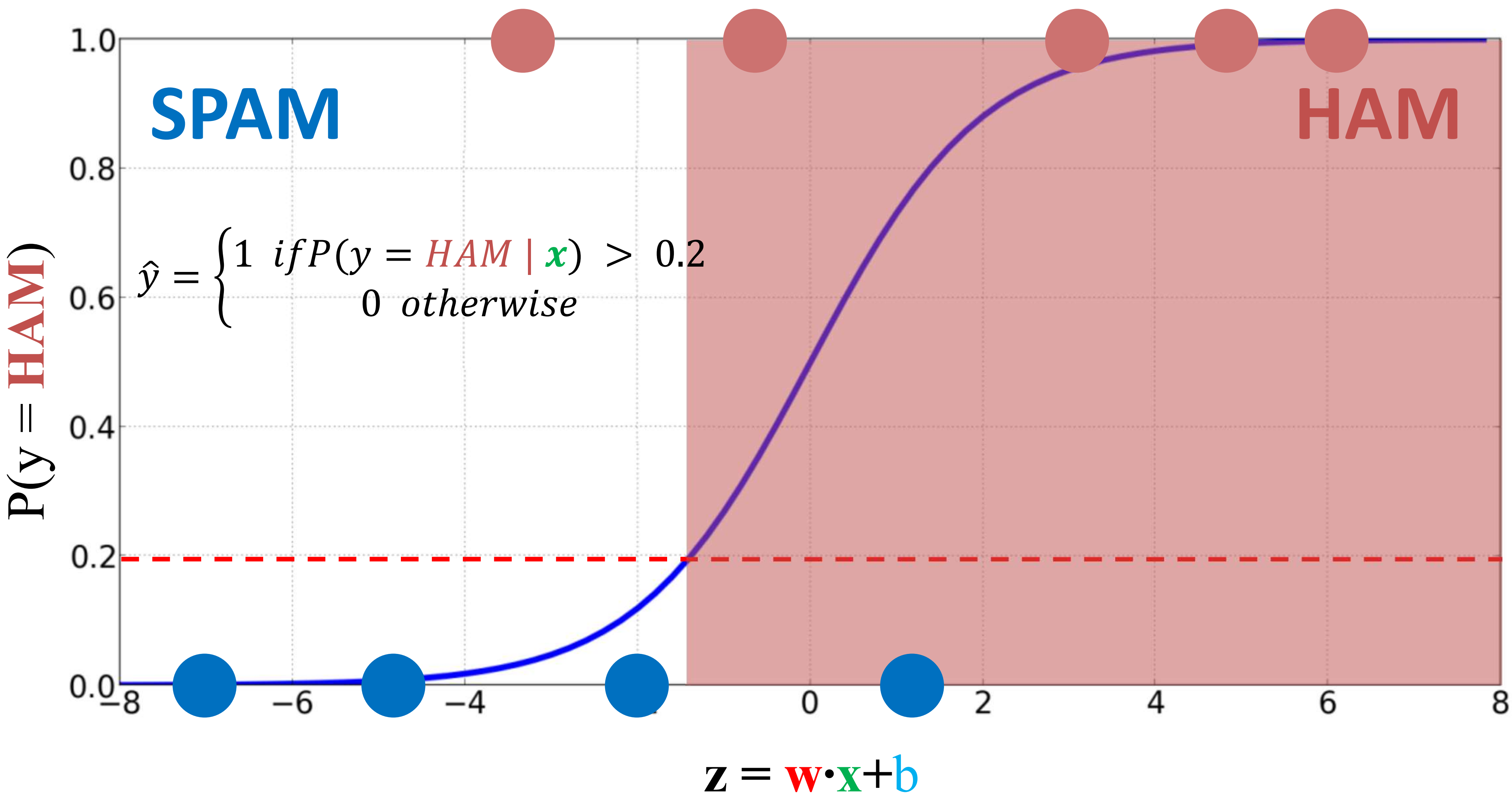
Logistic Regression: Threshold 0.5



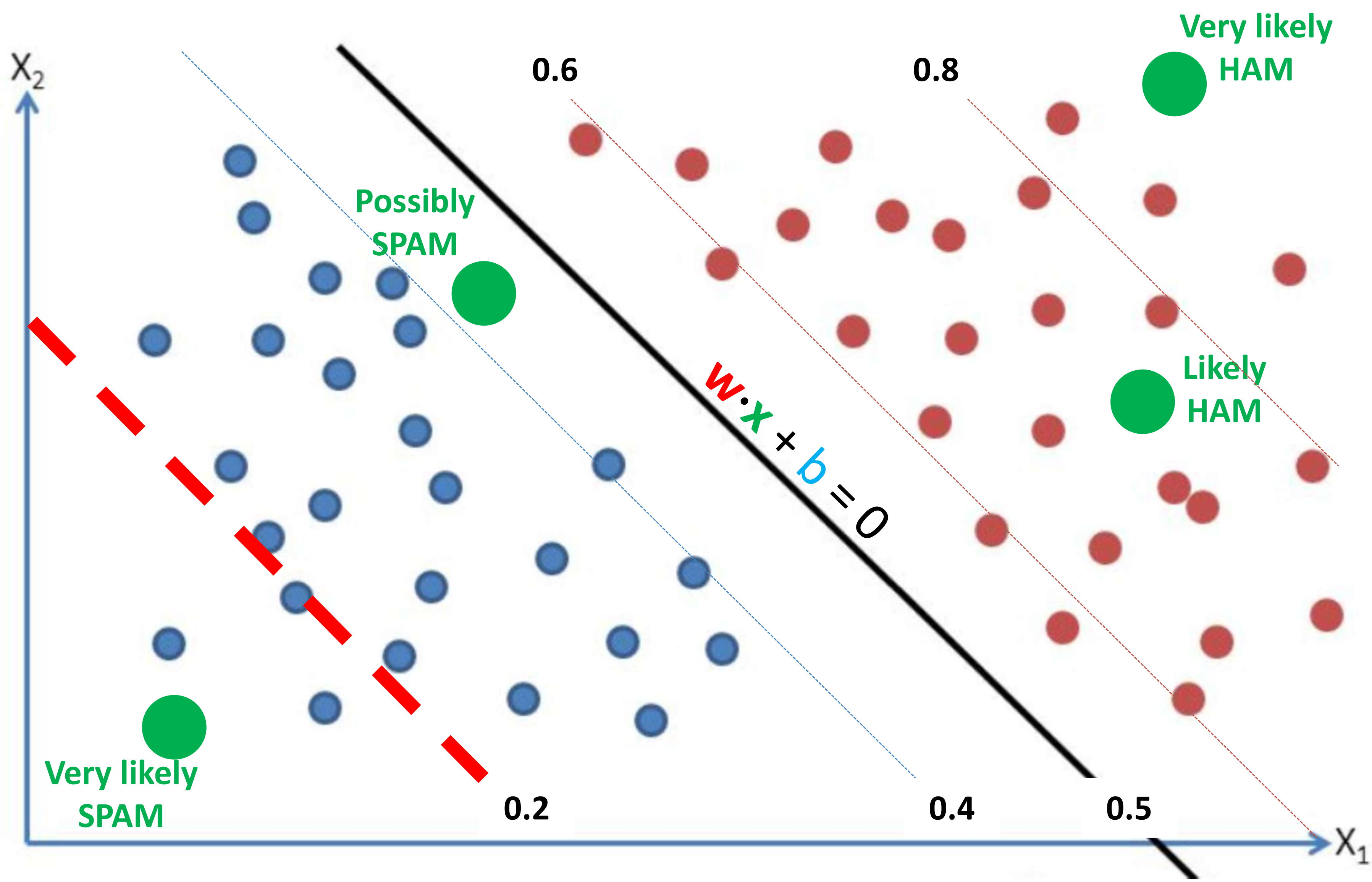
Text Classification: Separator



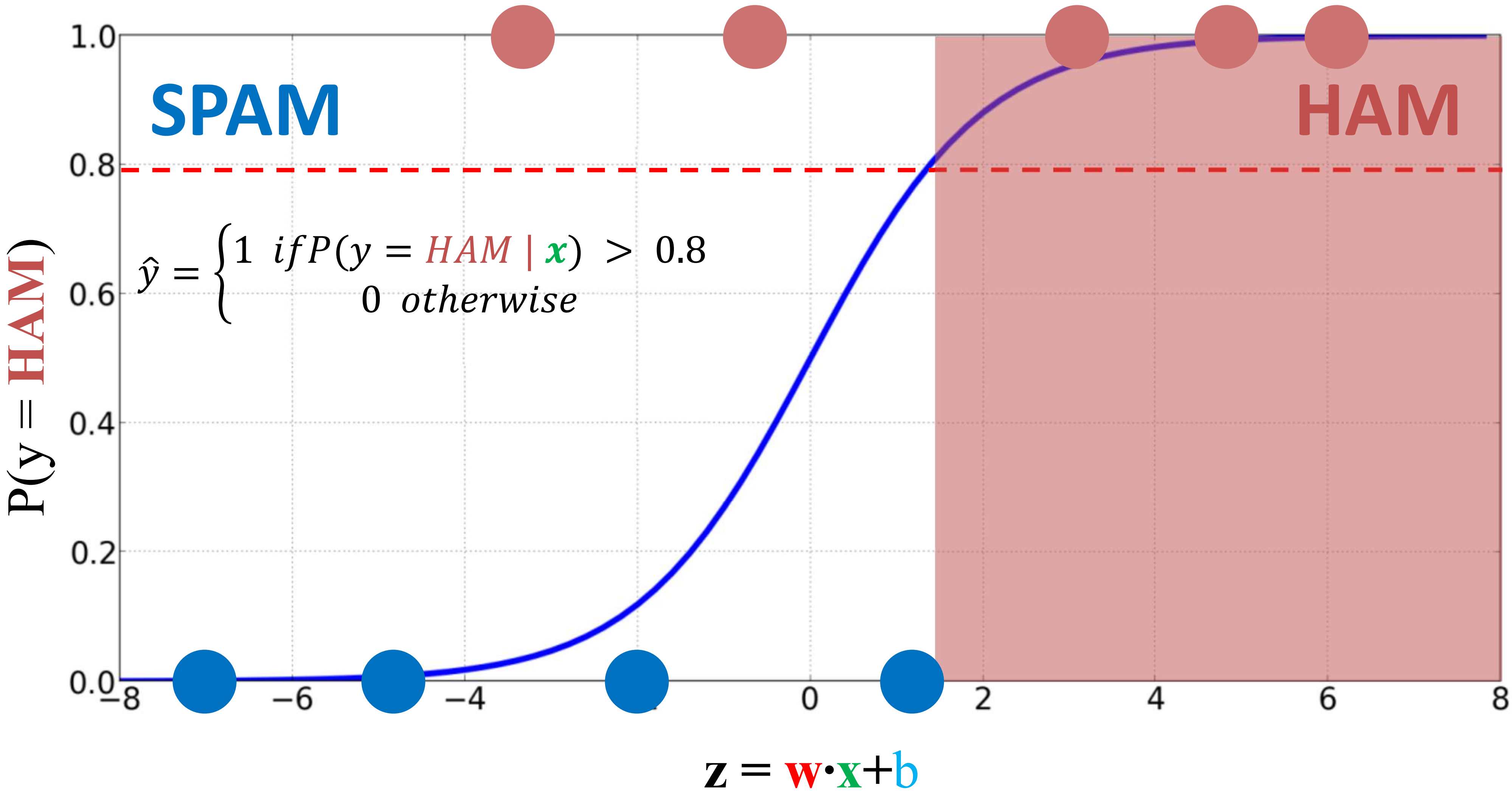
Logistic Regression: Threshold 0.2



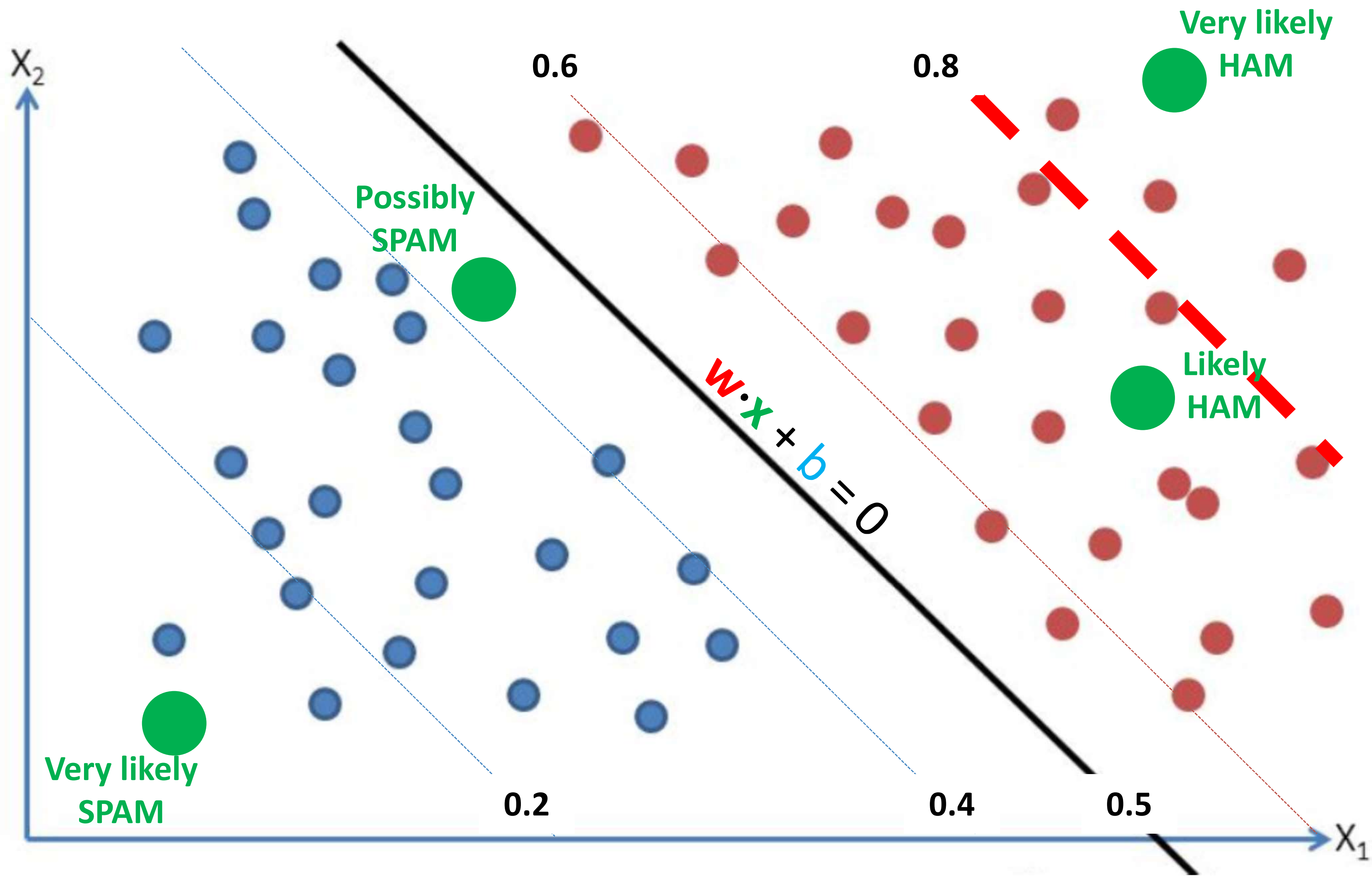
Text Classification: Separator



Logistic Regression: Threshold 0.8



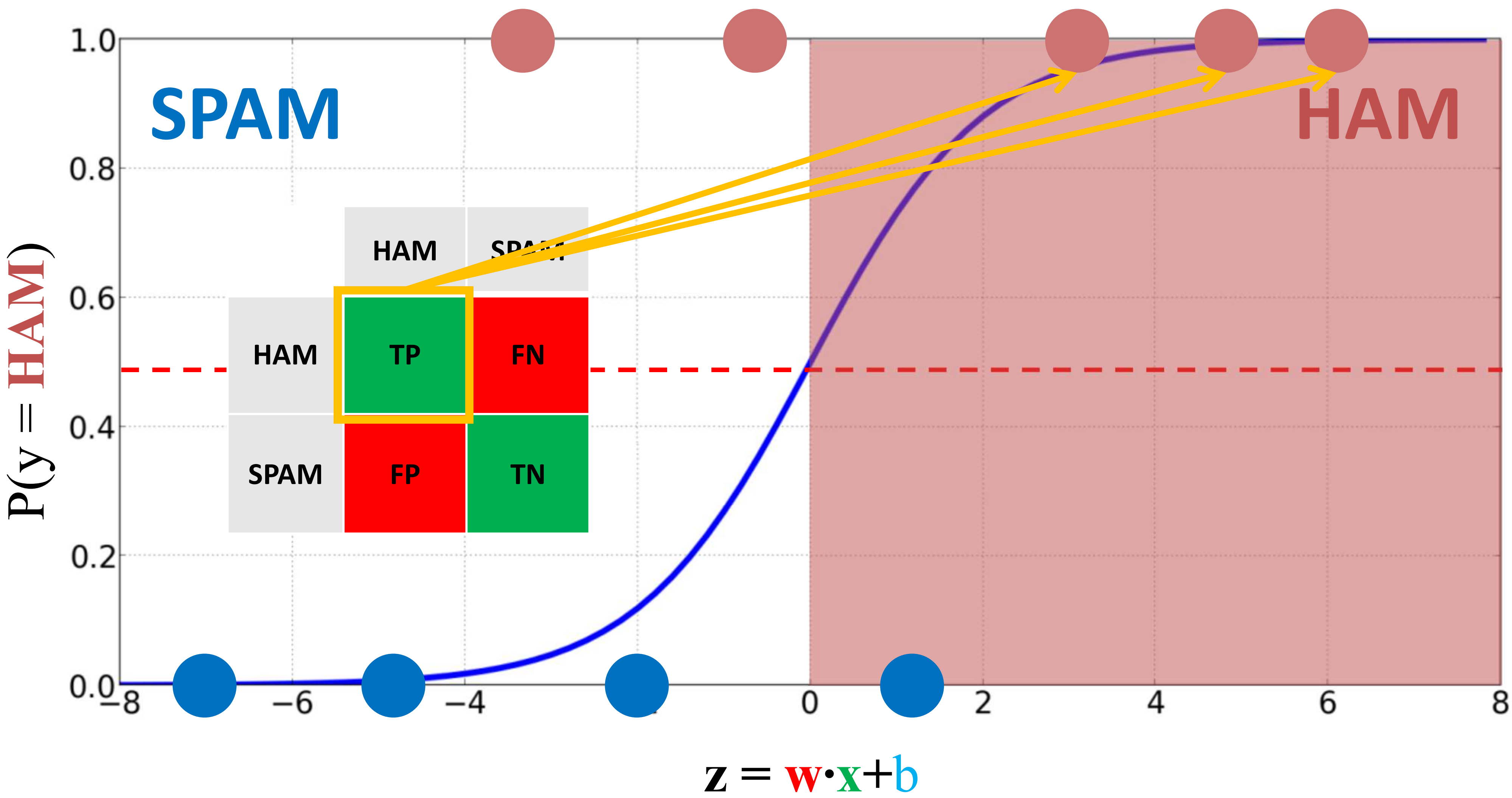
Text Classification: Separator



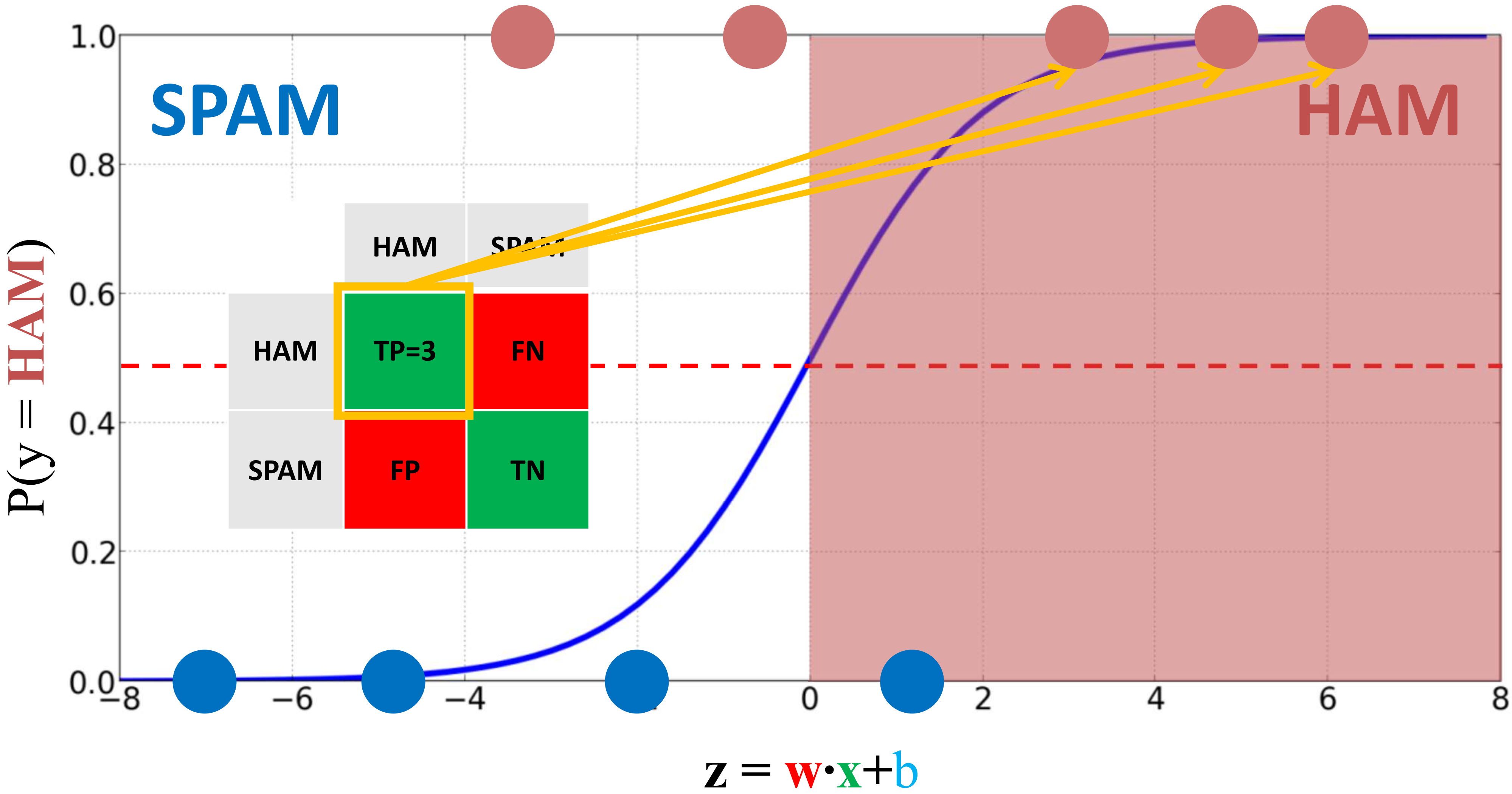
Classifier Evaluation: Confusion Matrix

		Predicted class		
		Positive	Negative	
Actual class	Positive	True Positive (TP)	False Negative (FN) Type II Error	Sensitivity (Recall) $\frac{TP}{TP+FN}$
	Negative	False Positive (FP) Type I Error	True Negative (TN)	Specificity $\frac{TN}{TN+FP}$
		Precision $\frac{TP}{TP+FP}$	Negative Predictive Value $\frac{TN}{TN+FN}$	Accuracy $\frac{TP+TN}{TP+TN+FP+FN}$

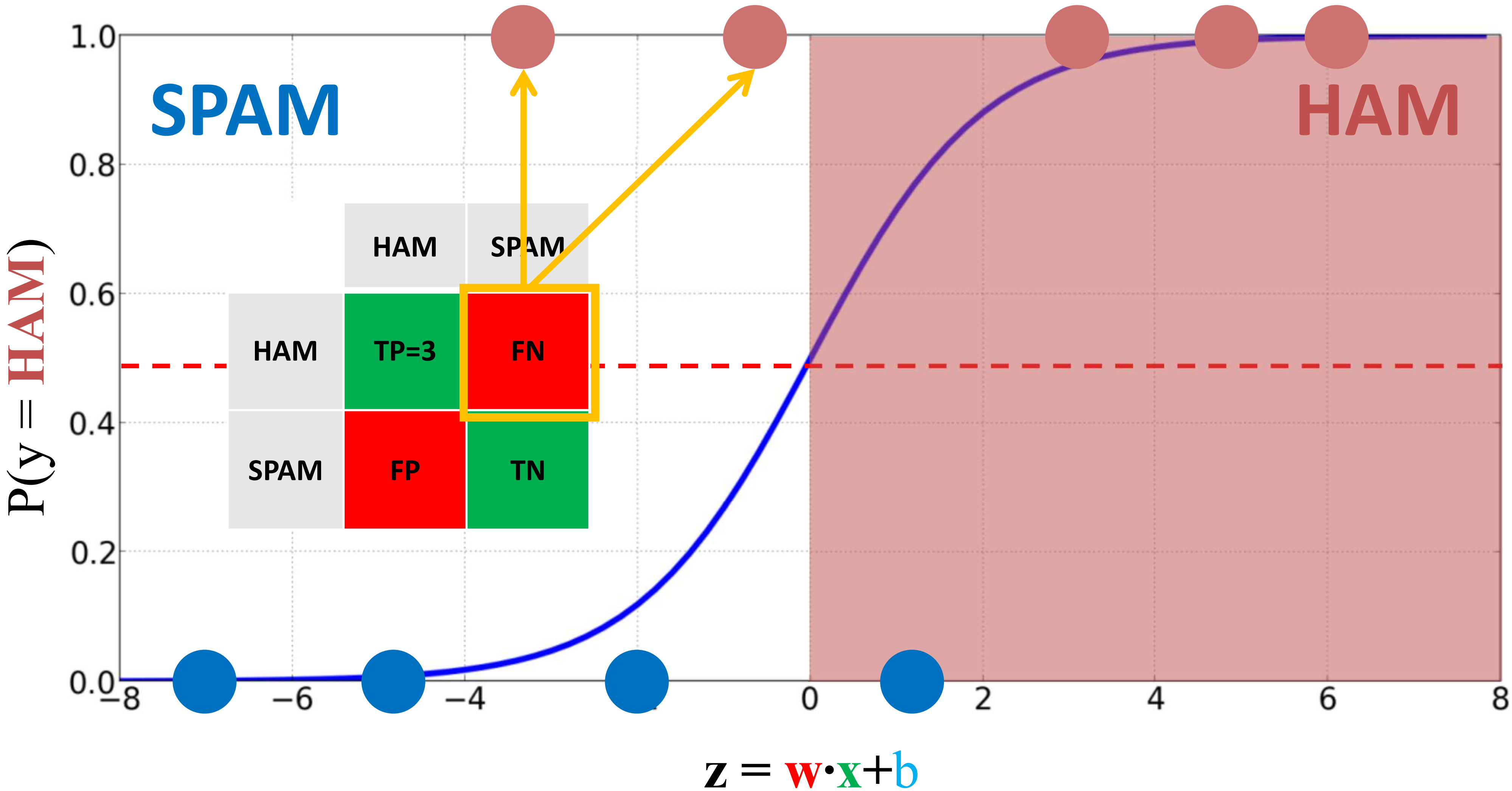
Logistic Regression: Threshold 0.5



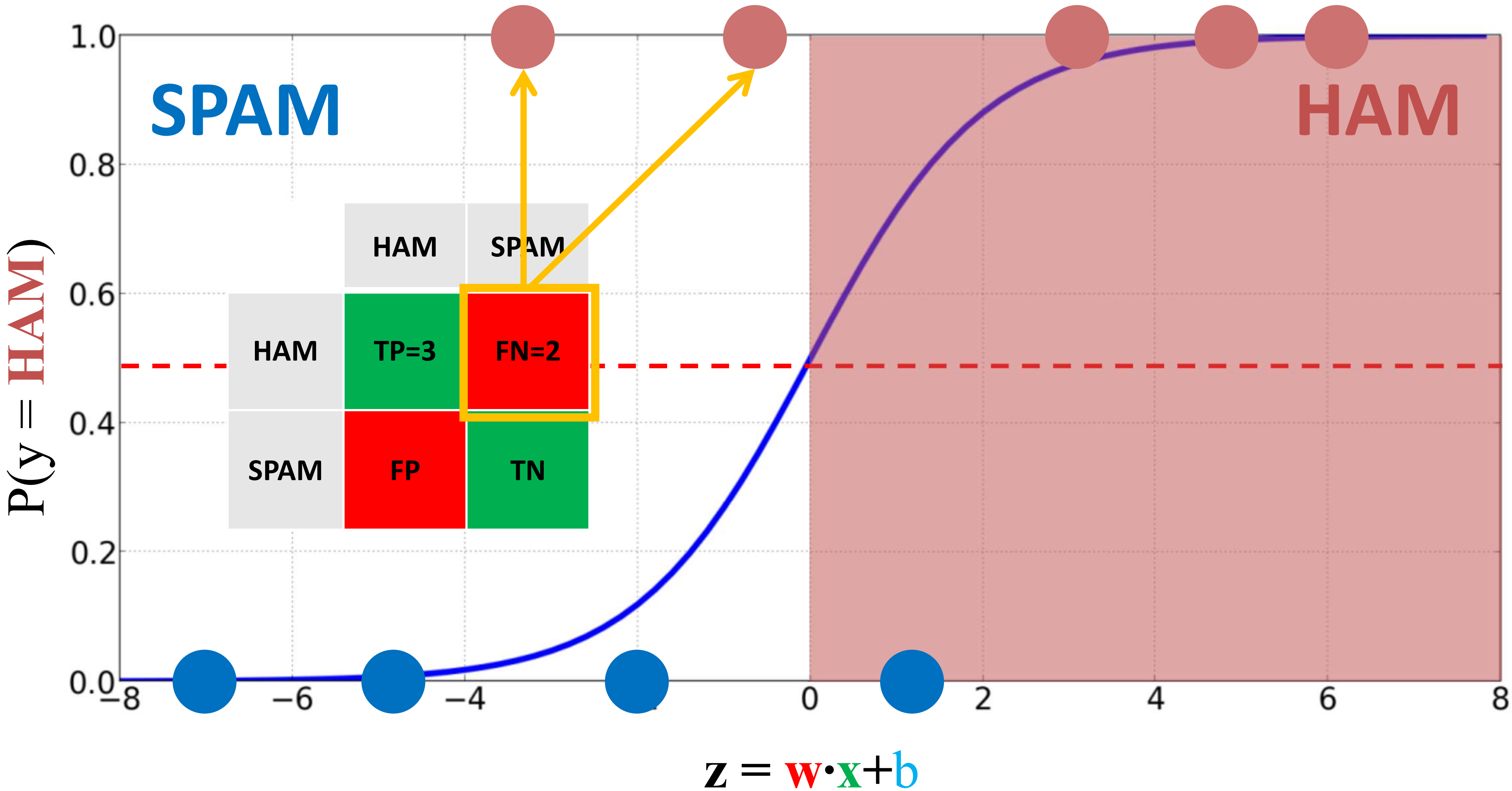
Logistic Regression: Threshold 0.5



Logistic Regression: Threshold 0.5

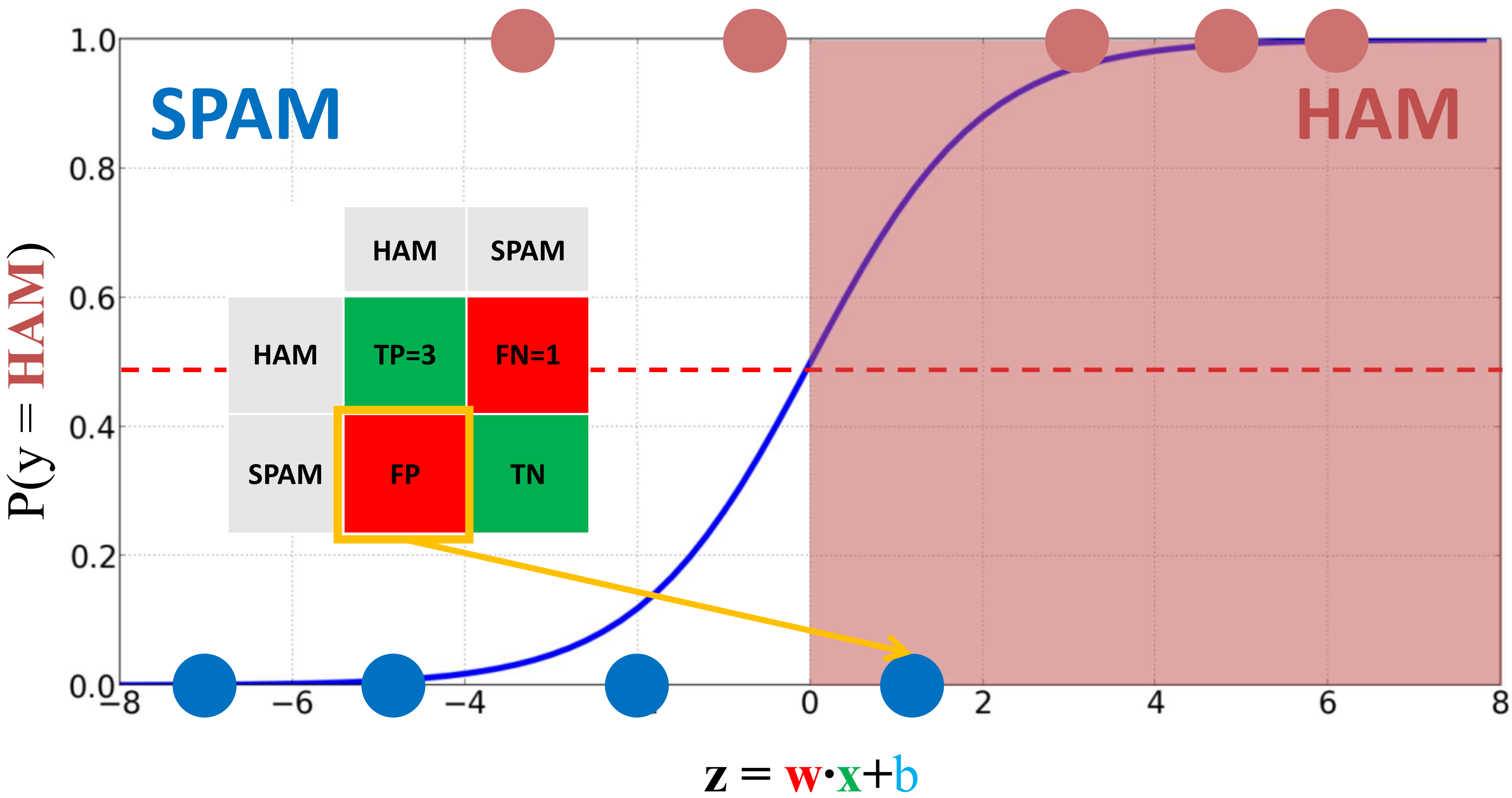


Logistic Regression: Threshold 0.5

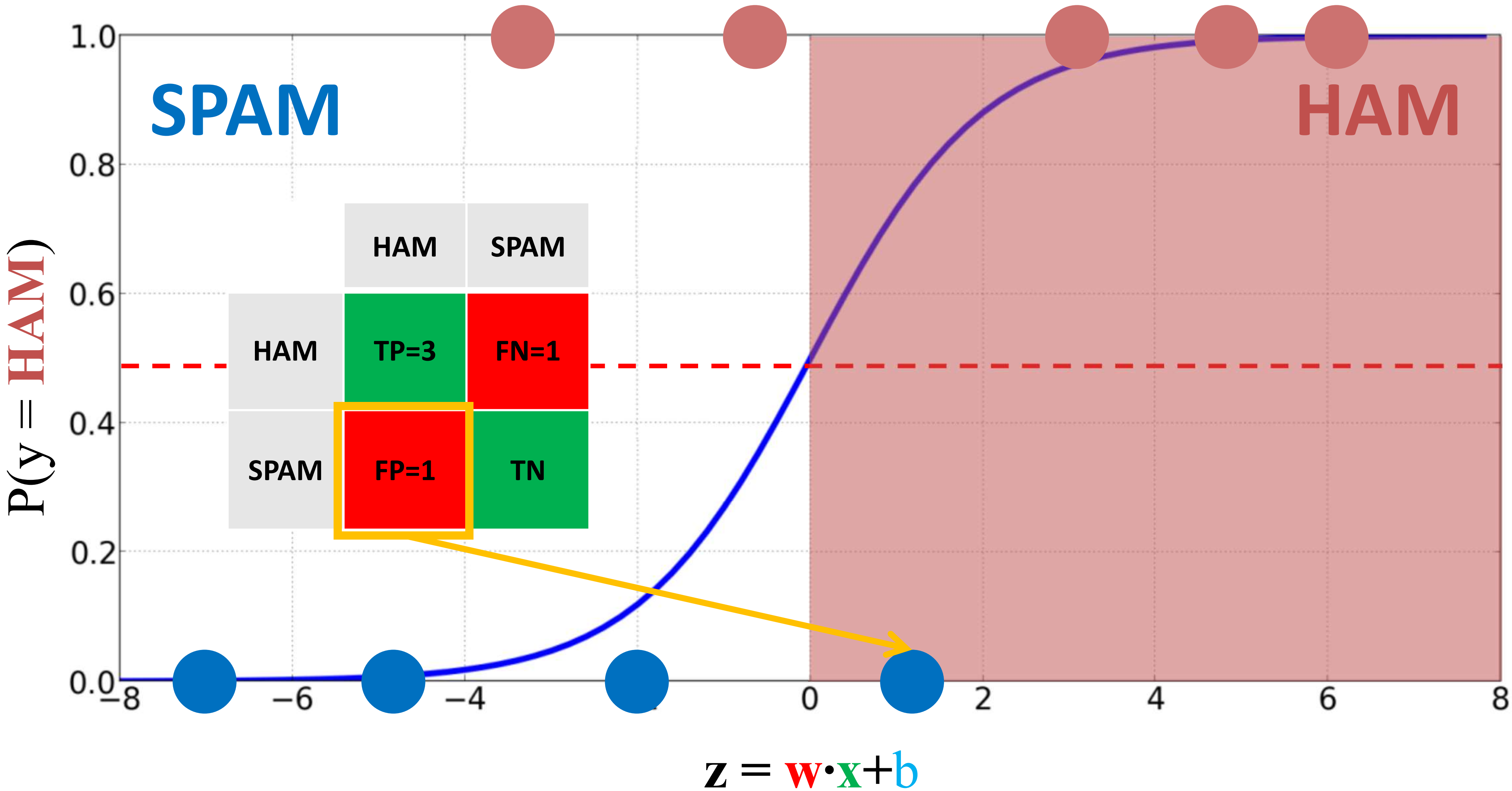


	HAM	SPAM
HAM	TP=3	FN=2
SPAM	FP	TN

Logistic Regression: Threshold 0.5

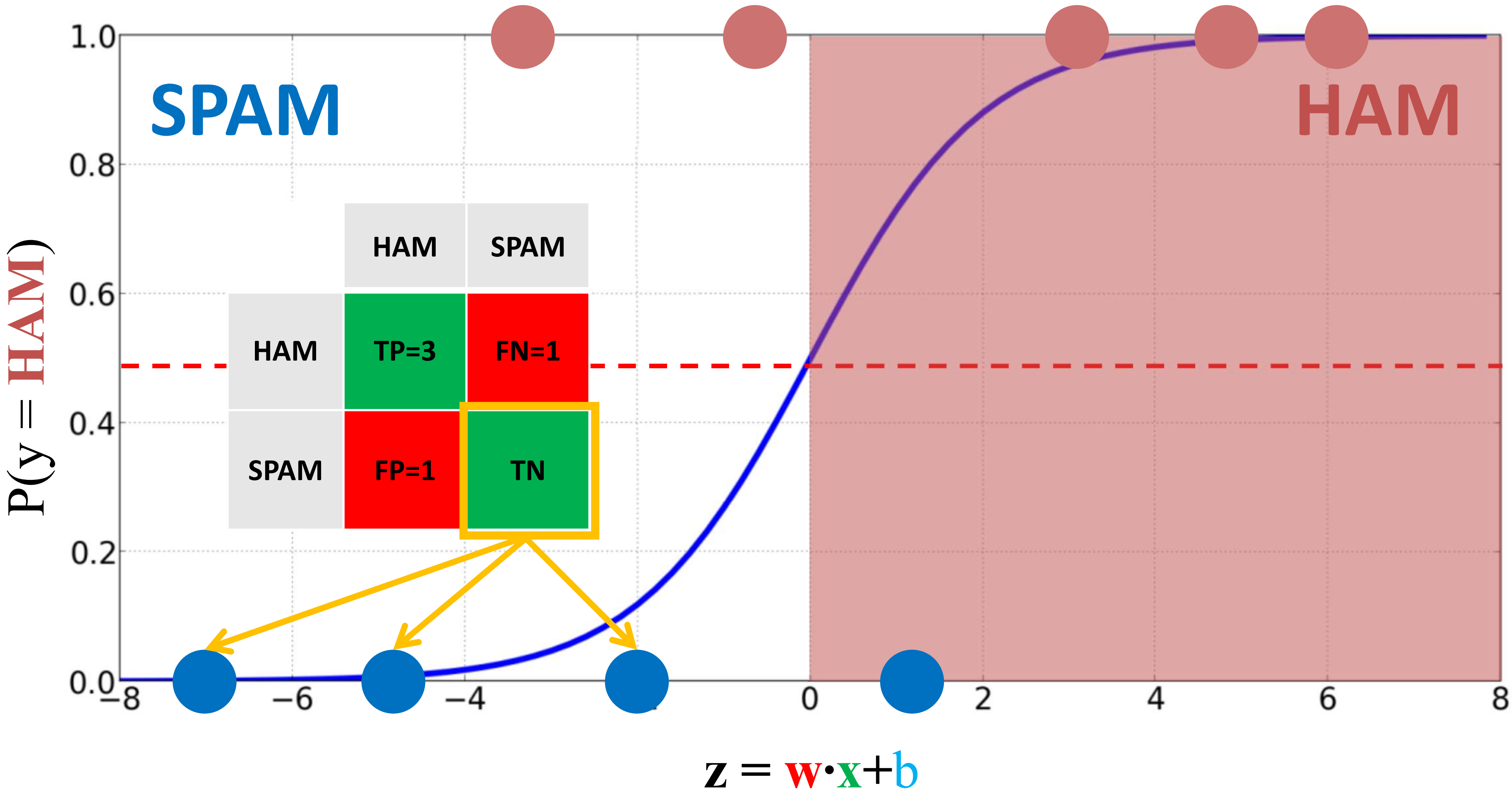


Logistic Regression: Threshold 0.5

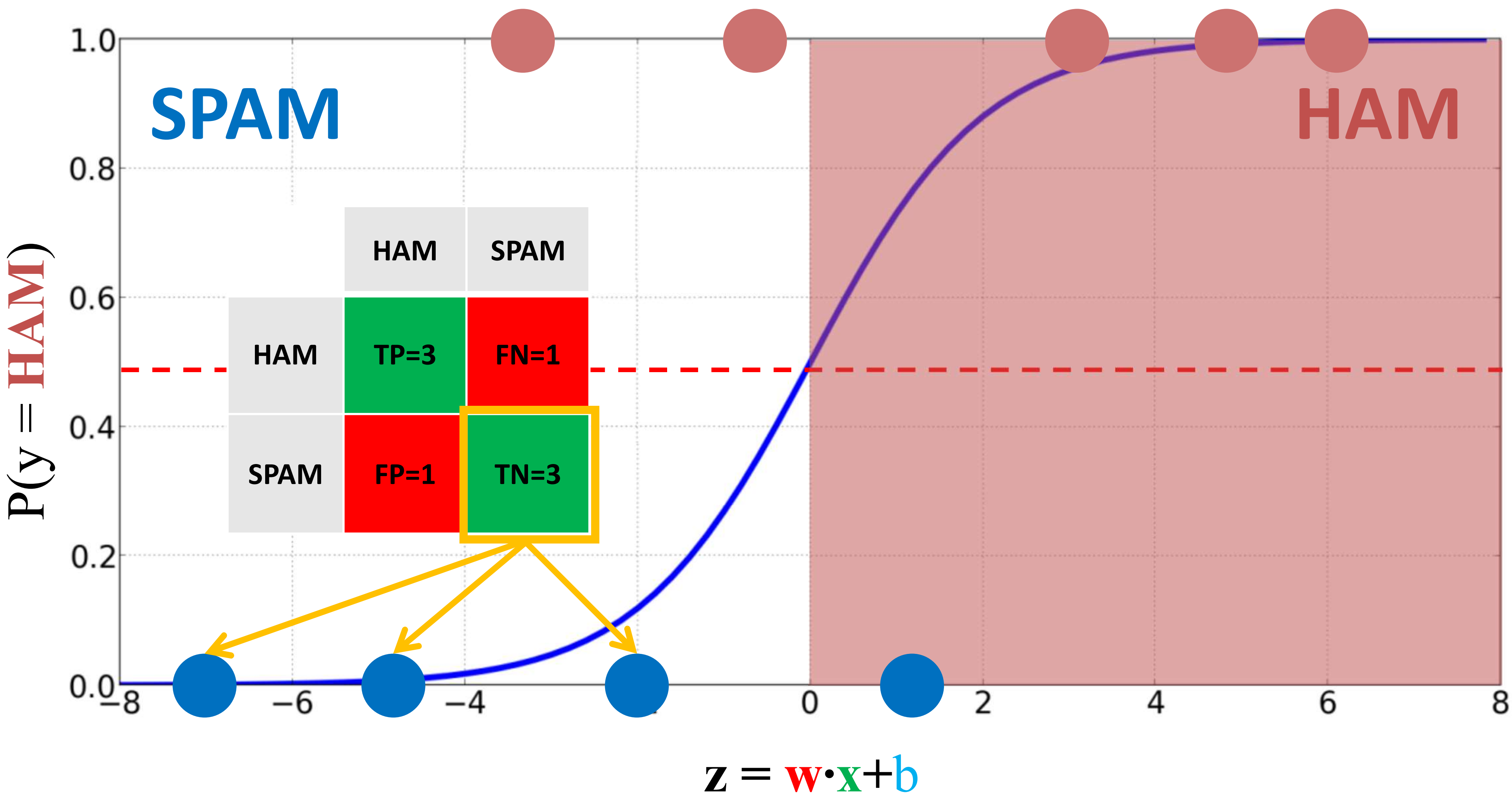


	HAM	SPAM
HAM	TP=3	FN=1
SPAM	FP=1	TN

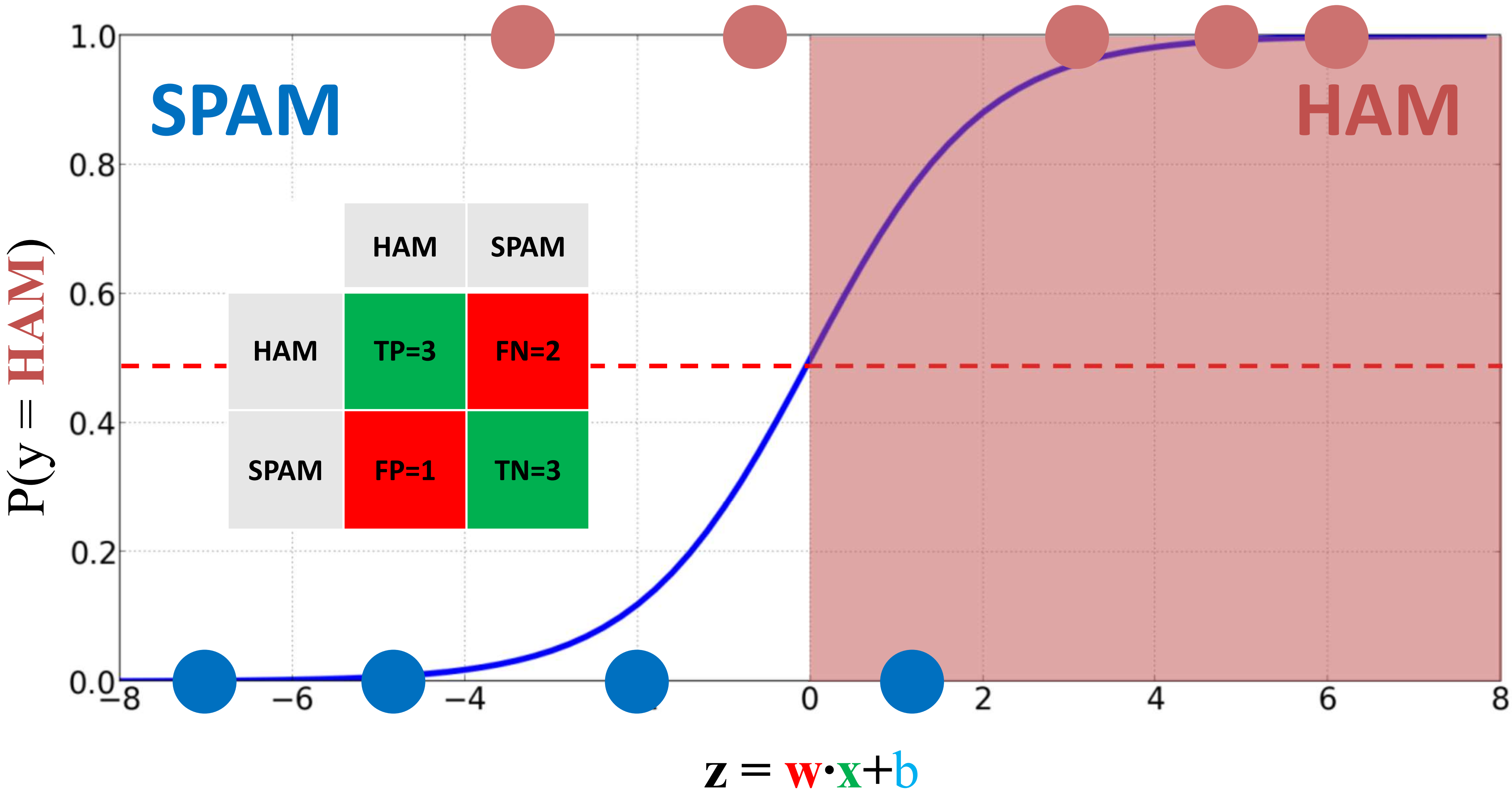
Logistic Regression: Threshold 0.5



Logistic Regression: Threshold 0.5



Logistic Regression: Threshold 0.5

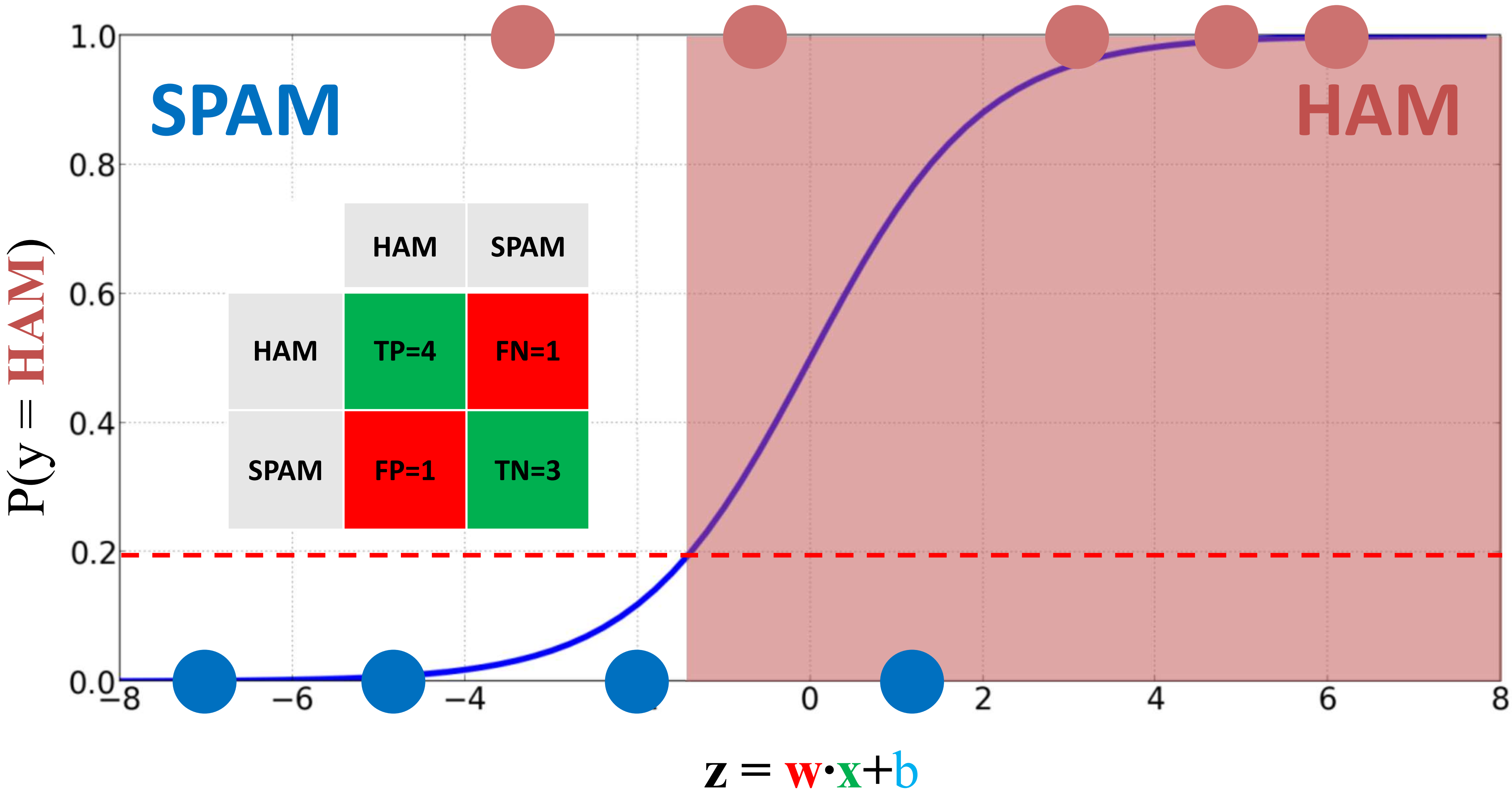


	HAM	SPAM
HAM	TP=3	FN=2
SPAM	FP=1	TN=3

Confusion Matrix: Threshold 0.5

		Predicted class		
		HAM	SPAM	
Actual class	HAM	True Positive (TP) 3	False Negative (FN) 2	Sensitivity (Recall) $\frac{TP}{TP+FN} = \frac{3}{5}$
	SPAM	False Positive (FP) 1	True Negative (TN) 3	Specificity $\frac{TN}{TN+FP} = \frac{3}{4}$
		Precision $\frac{TP}{TP+FP} = \frac{3}{4}$	Negative Predictive Value $\frac{TN}{TN+FN} = \frac{3}{5}$	Accuracy $\frac{TP+TN}{TP+TN+FP+FN} = \frac{6}{9}$

Logistic Regression: Threshold 0.2



	HAM	SPAM
HAM	TP=4	FN=1
SPAM	FP=1	TN=3

Confusion Matrix: Threshold 0.2

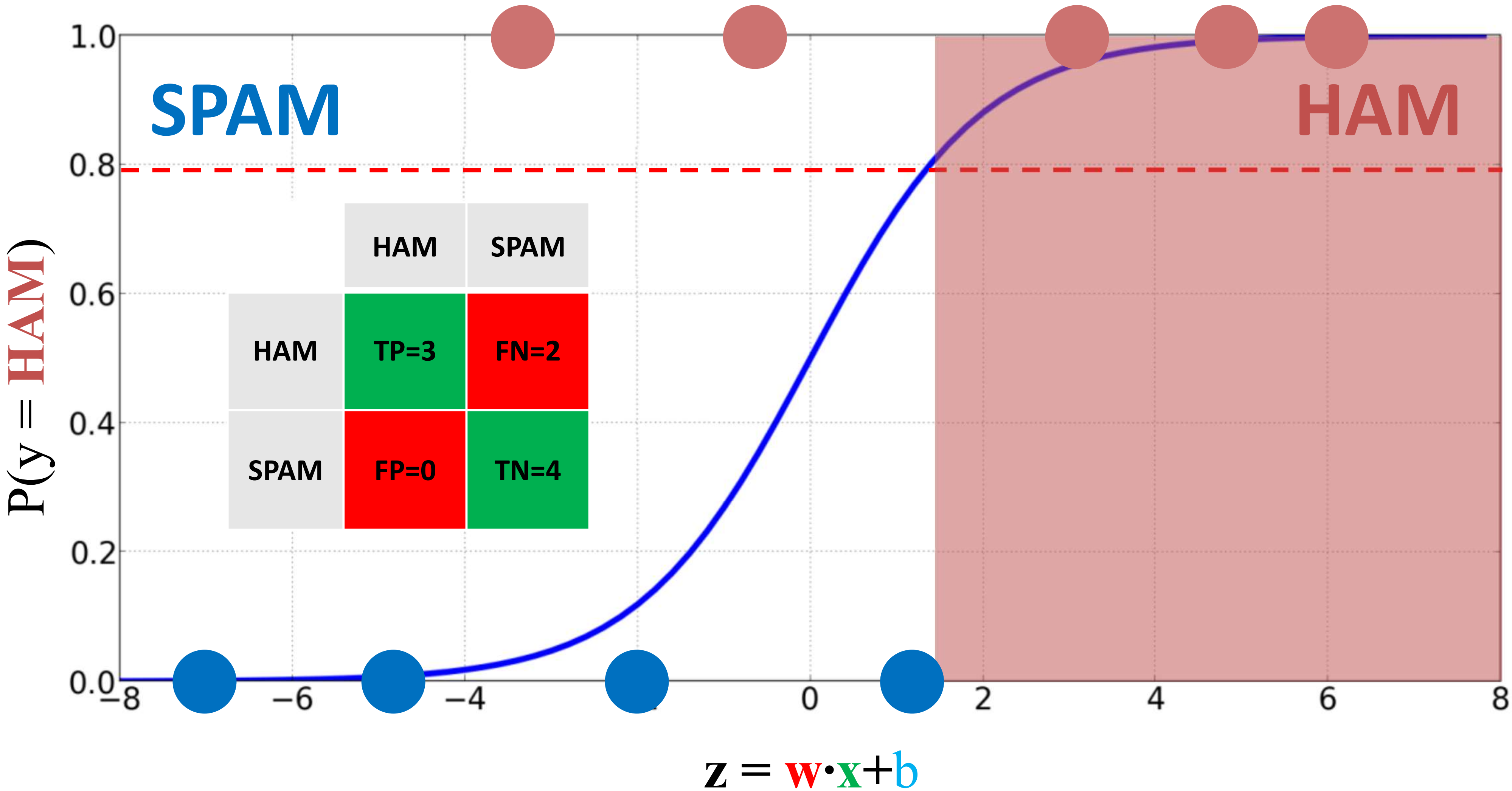
		Predicted class		
		HAM	SPAM	
Actual class	HAM	True Positive (TP) 4	False Negative (FN) 1	Sensitivity (Recall) $\frac{TP}{TP+FN} = \frac{4}{5}$
	SPAM	False Positive (FP) 1	True Negative (TN) 3	Specificity $\frac{TN}{TN+FP} = \frac{3}{4}$
		Precision $\frac{TP}{TP+FP} = \frac{4}{5}$	Negative Predictive Value $\frac{TN}{TN+FN} = \frac{3}{4}$	Accuracy $\frac{TP+TN}{TP+TN+FP+FN} = \frac{7}{9}$

Confusion Matrix: Threshold 0.2

		Predicted class		
		Positive	Negative	
Actual class	Positive	True Positive (TP) 4	False Negative (FN) 1	Sensitivity (Recall) $\frac{TP}{TP+FN} = \frac{4}{5}$
	Negative	False Positive (FP) 1	True Negative (TN) 3	Specificity $\frac{TN}{TN+FP} = \frac{3}{4}$
		Precision $\frac{TP}{TP+FP} = \frac{4}{5}$	Negative Predictive Value $\frac{TN}{TN+FN} = \frac{3}{4}$	Accuracy $\frac{TP+TN}{TP+TN+FP+FN} = \frac{7}{9}$

Lower threshold
Less False
Negatives (Ex:
Covid tests)

Logistic Regression: Threshold 0.8



	HAM	SPAM
HAM	TP=3	FN=2
SPAM	FP=0	TN=4

Confusion Matrix: Threshold 0.8

		Predicted class		
		HAM	SPAM	
Actual class	HAM	True Positive (TP) 3	False Negative (FN) 2	Sensitivity (Recall) $\frac{TP}{TP+FN} = \frac{3}{5}$
	SPAM	False Positive (FP) 0	True Negative (TN) 4	Specificity $\frac{TN}{TN+FP} = \frac{4}{4}$
		Precision $\frac{TP}{TP+FP} = \frac{3}{3}$	Negative Predictive Value $\frac{TN}{TN+FN} = \frac{4}{6}$	Accuracy $\frac{TP+TN}{TP+TN+FP+FN} = \frac{7}{9}$

Confusion Matrix: Threshold 0.8

		Predicted class		
		Positive	Negative	
Actual class	Positive	True Positive (TP) 3	False Negative (FN) 2	Sensitivity (Recall) $\frac{TP}{TP+FN} = \frac{3}{5}$
	Negative	False Positive (FP) 0	True Negative (TN) 4	Specificity $\frac{TN}{TN+FP} = \frac{4}{4}$
	Higher threshold Less False Positives (Ex: crime convictions)	Precision $\frac{TP}{TP+FP} = \frac{3}{3}$	Negative Predictive Value $\frac{TN}{TN+FN} = \frac{4}{6}$	Accuracy $\frac{TP+TN}{TP+TN+FP+FN} = \frac{7}{9}$

Classifier Evaluation: Confusion Matrix

		Predicted class		
		Positive	**Negative**	
Actual class	**Positive**	**True Positive (TP)**	**False Negative (FN)** **Type II Error**	**Sensitivity (Recall)** $\frac{TP}{TP+FN}$
	Negative	**False Positive (FP)** **Type I Error**	**True Negative (TN)**	**Specificity** $\frac{TN}{TN+FP}$
		Precision $\frac{TP}{TP+FP}$	**Negative Predictive Value** $\frac{TN}{TN+FN}$	**Accuracy** $\frac{TP+TN}{TP+TN+FP+FN}$

Classifier Performance Metrics

- **Precision and recall provide two ways to summarize the errors made for the positive class (FP, FN).**

- **F-measure provides a single score that summarizes the precision and recall.**

- **Accuracy summarizes the correct predictions for both positive and negative classes.**

Receiver Operating Characteristic

Threshold: 0.5

●	HAM	SPAM
HAM	TP=3	FN=2
SPAM	FP=1	TN=3

$$TPR = \frac{TP}{TP + FN} = \frac{3}{5}$$

$$FPR = \frac{FP}{FP + TN} = \frac{1}{4}$$

Threshold: 0.2

●	HAM	SPAM
HAM	TP=4	FN=1
SPAM	FP=1	TN=3

$$TPR = \frac{TP}{TP + FN} = \frac{4}{5}$$

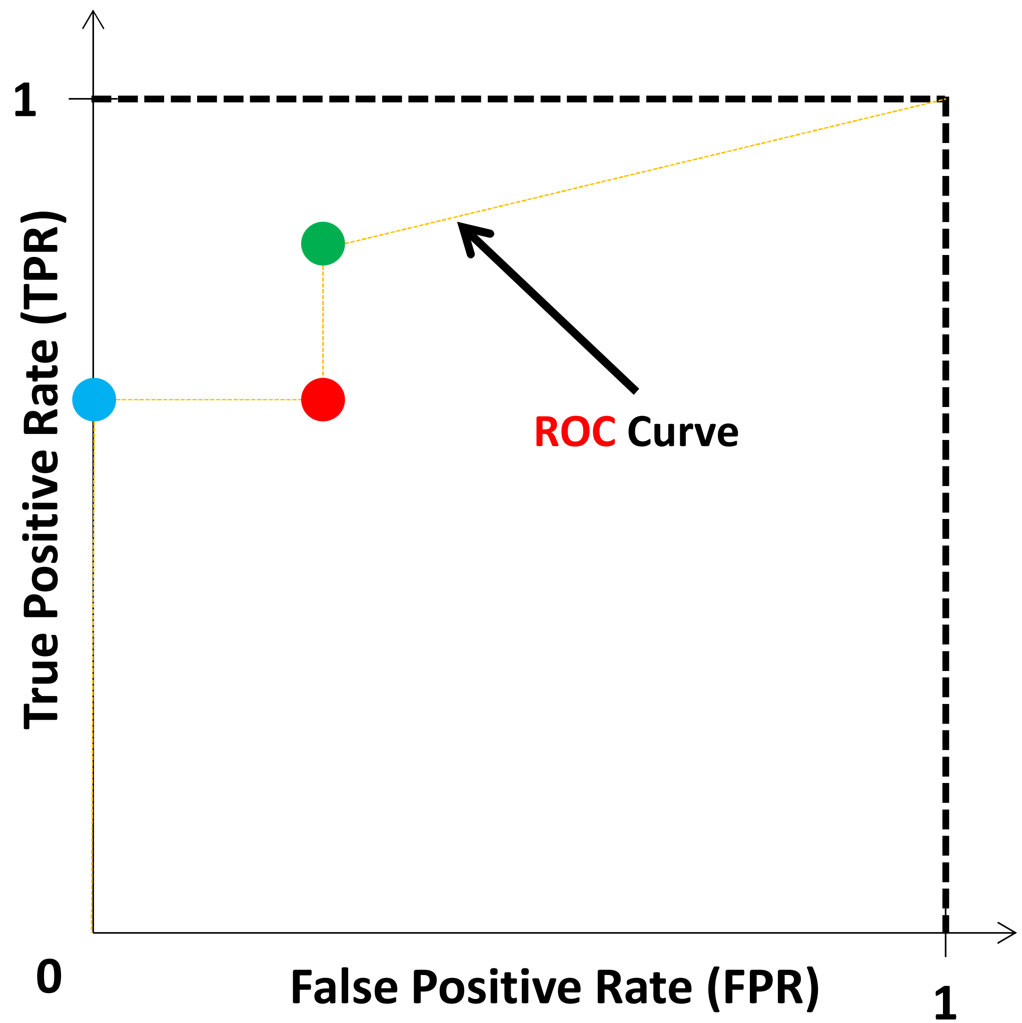
$$FPR = \frac{FP}{FP + TN} = \frac{1}{4}$$

Threshold: 0.8

●	HAM	SPAM
HAM	TP=3	FN=2
SPAM	FP=0	TN=4

$$TPR = \frac{TP}{TP + FN} = \frac{3}{5}$$

$$FPR = \frac{FP}{FP + TN} = \frac{0}{4}$$



TPR: Sensitivity | FPR: 1 - Specificity

ROC Area Under the Curve

Threshold: 0.5

	HAM	SPAM
HAM	TP=3	FN=2
SPAM	FP=1	TN=3

$$TPR = \frac{TP}{TP + FN} = \frac{3}{5}$$

$$FPR = \frac{FP}{FP + TN} = \frac{1}{4}$$

Threshold: 0.2

	HAM	SPAM
HAM	TP=4	FN=1
SPAM	FP=1	TN=3

$$TPR = \frac{TP}{TP + FN} = \frac{4}{5}$$

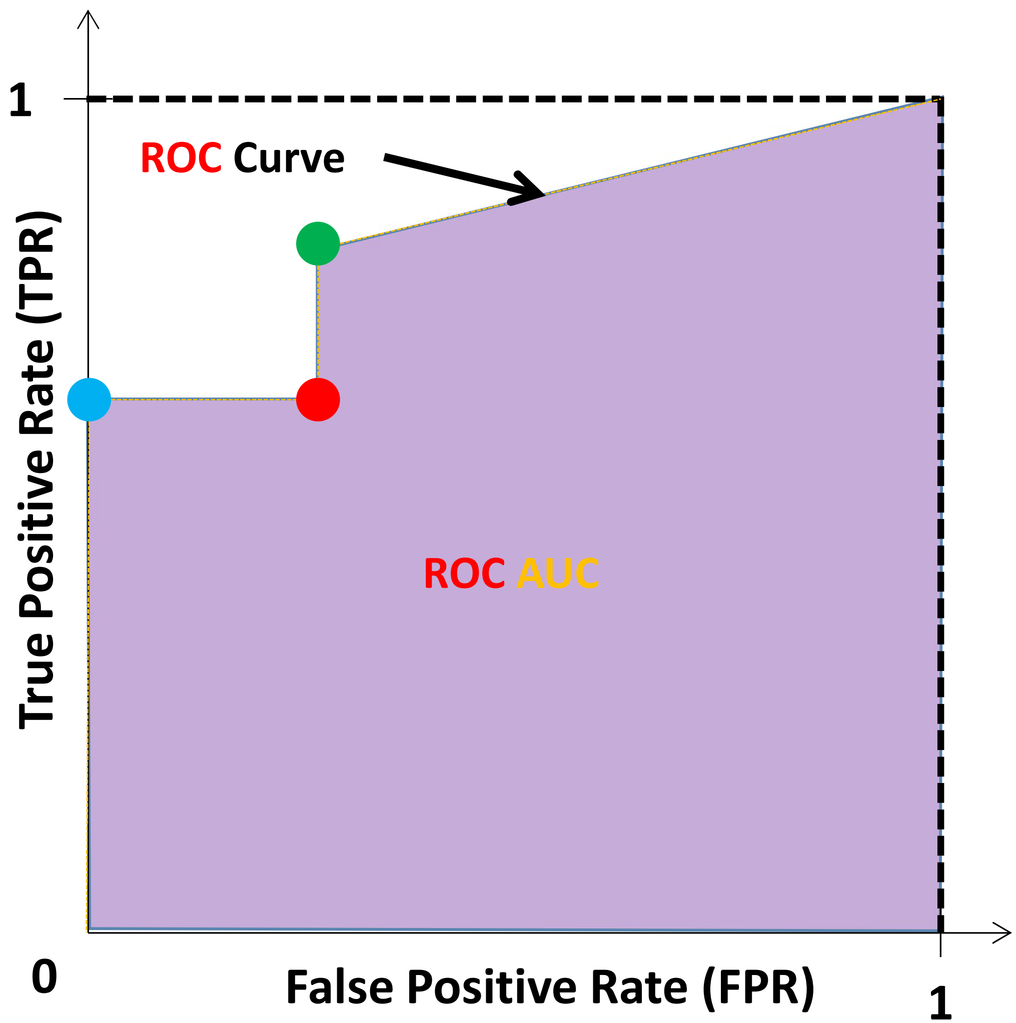
$$FPR = \frac{FP}{FP + TN} = \frac{1}{4}$$

Threshold: 0.8

	HAM	SPAM
HAM	TP=3	FN=2
SPAM	FP=0	TN=4

$$TPR = \frac{TP}{TP + FN} = \frac{3}{5}$$

$$FPR = \frac{FP}{FP + TN} = \frac{0}{4}$$



TPR: Sensitivity | FPR: 1 - Specificity

Receiver Operating Characteristic

Threshold: 0.5

●	HAM	SPAM
HAM	TP=3	FN=2
SPAM	FP=1	TN=3

$$TPR = \frac{TP}{TP+FN} = \frac{3}{5}$$

$$FPR = \frac{FP}{FP+TN} = \frac{1}{4}$$

Threshold: 0.2

●	HAM	SPAM
HAM	TP=4	FN=1
SPAM	FP=1	TN=3

$$TPR = \frac{TP}{TP+FN} = \frac{4}{5}$$

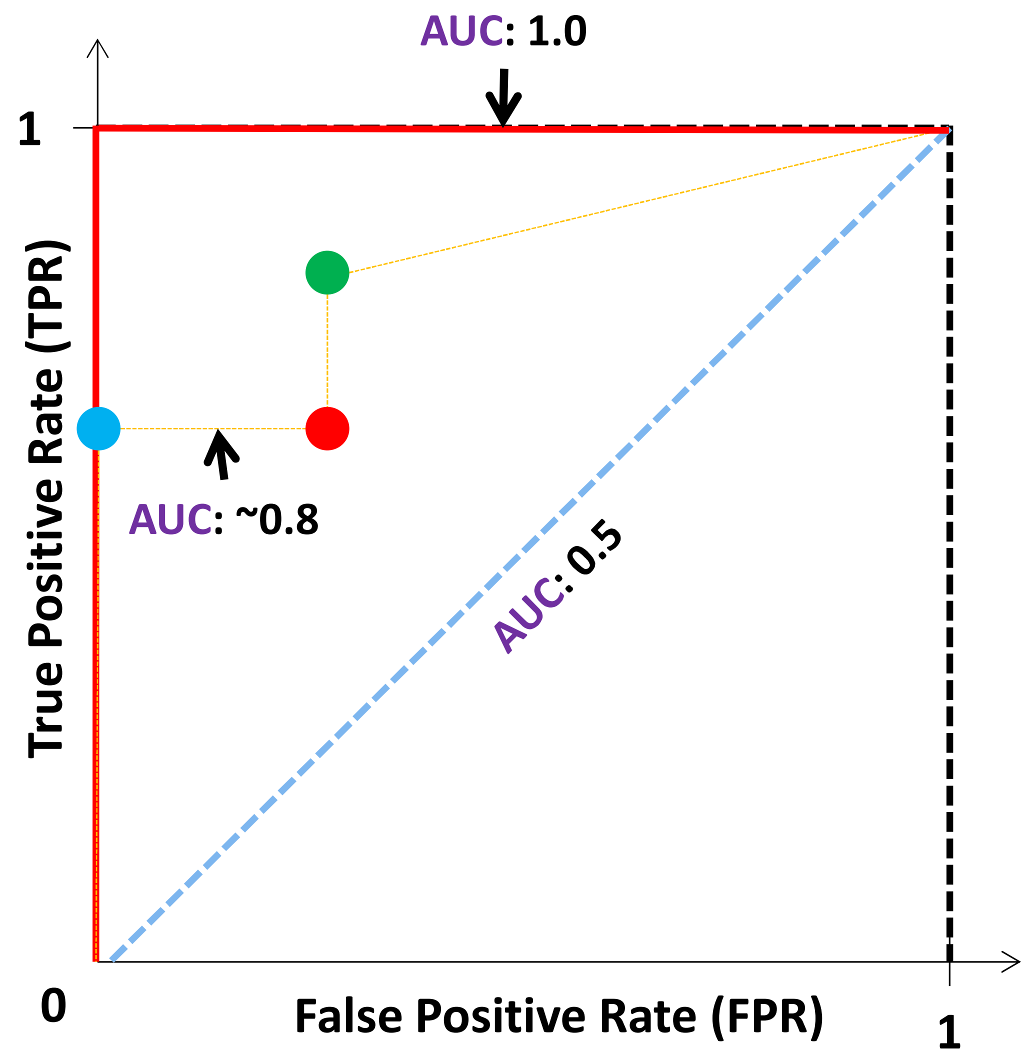
$$FPR = \frac{FP}{FP+TN} = \frac{1}{4}$$

Threshold: 0.8

●	HAM	SPAM
HAM	TP=3	FN=2
SPAM	FP=0	TN=4

$$TPR = \frac{TP}{TP+FN} = \frac{3}{5}$$

$$FPR = \frac{FP}{FP+TN} = \frac{0}{4}$$



TPR: Sensitivity | FPR: 1 - Specificity

Receiver Operating Characteristic

Threshold: 0.5

	HAM	SPAM
HAM	TP=3	FN=2
SPAM	FP=1	TN=3

$$TPR = \frac{TP}{TP+FN} = \frac{3}{5}$$

$$FPR = \frac{FP}{FP+TN} = \frac{1}{4}$$

Threshold: 0.2

	HAM	SPAM
HAM	TP=4	FN=1
SPAM	FP=1	TN=3

$$TPR = \frac{TP}{TP+FN} = \frac{4}{5}$$

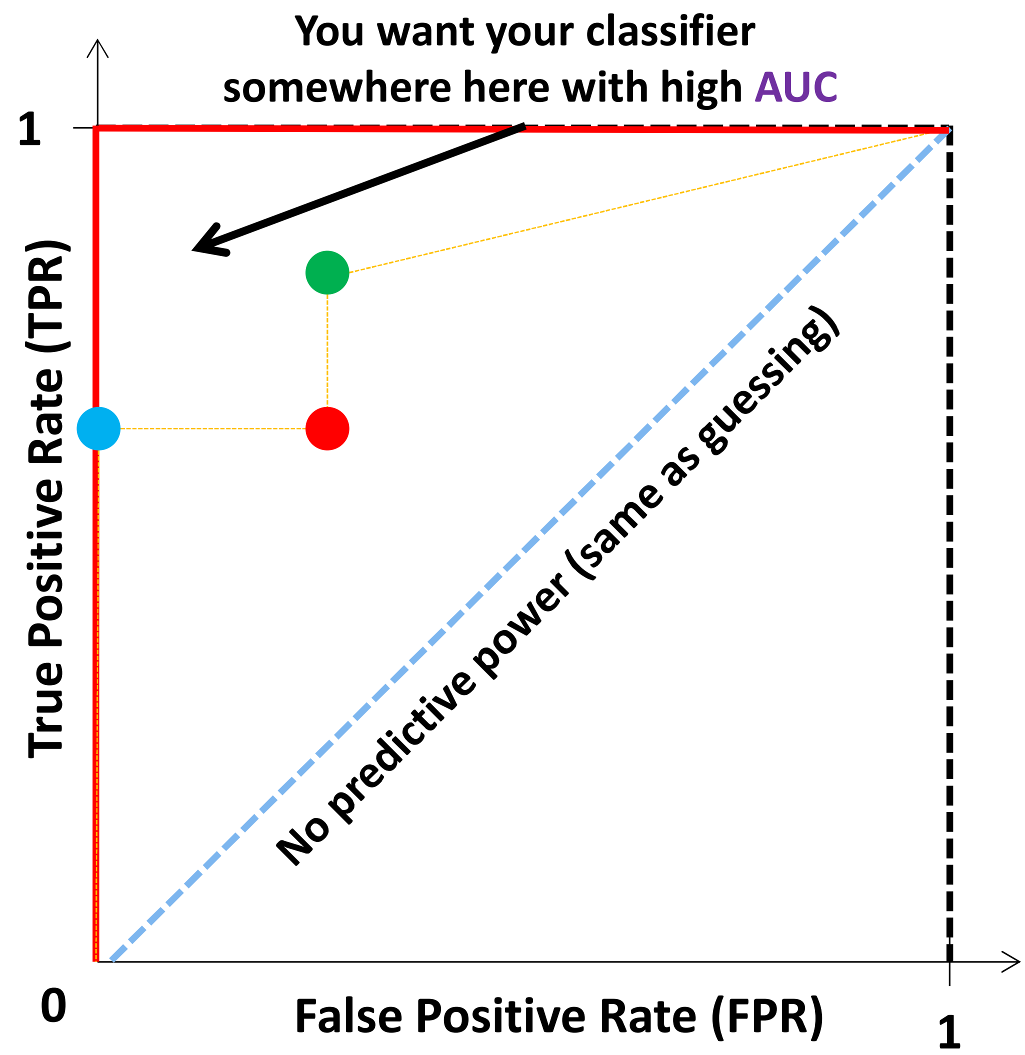
$$FPR = \frac{FP}{FP+TN} = \frac{1}{4}$$

Threshold: 0.8

	HAM	SPAM
HAM	TP=3	FN=2
SPAM	FP=0	TN=4

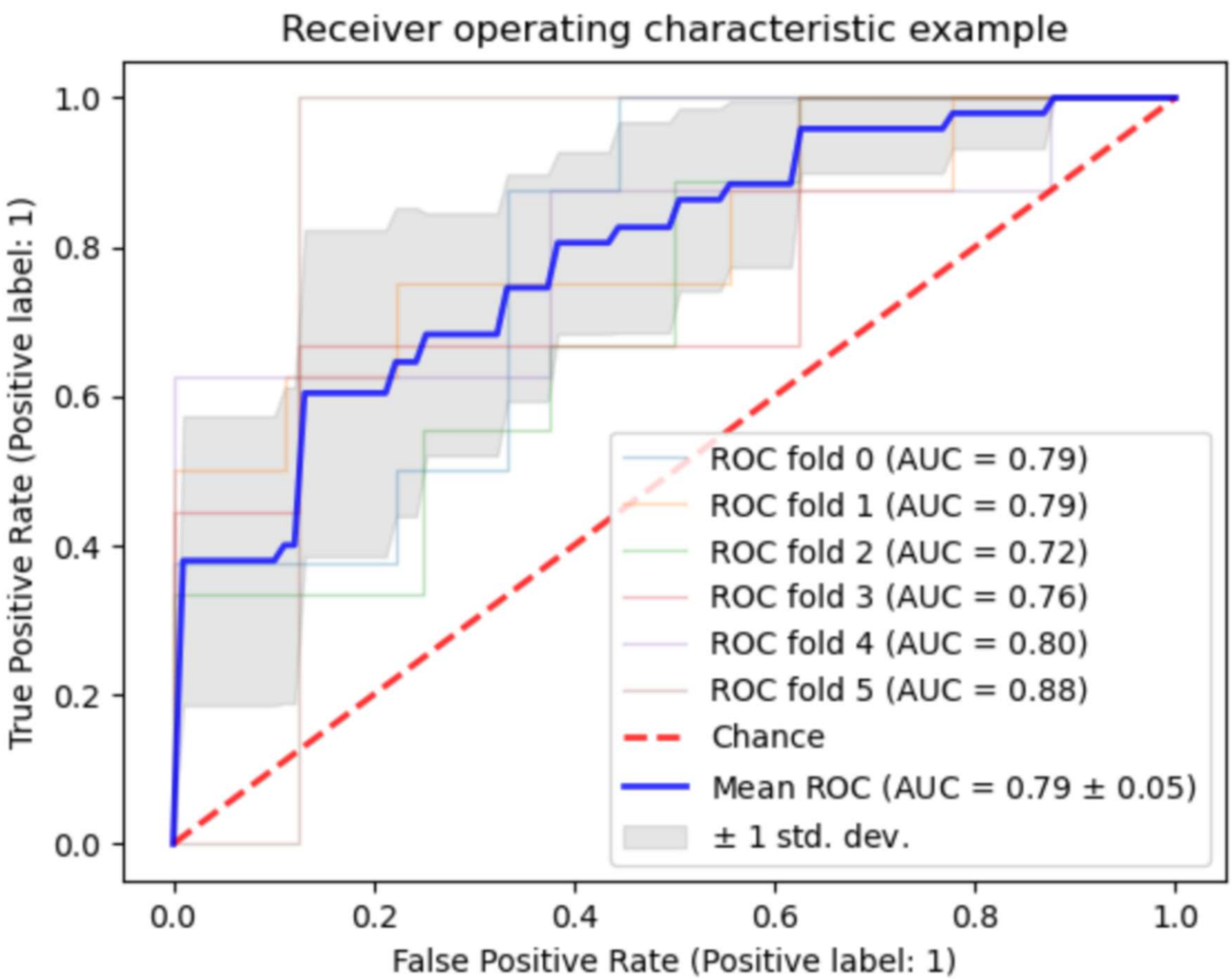
$$TPR = \frac{TP}{TP+FN} = \frac{3}{5}$$

$$FPR = \frac{FP}{FP+TN} = \frac{0}{4}$$

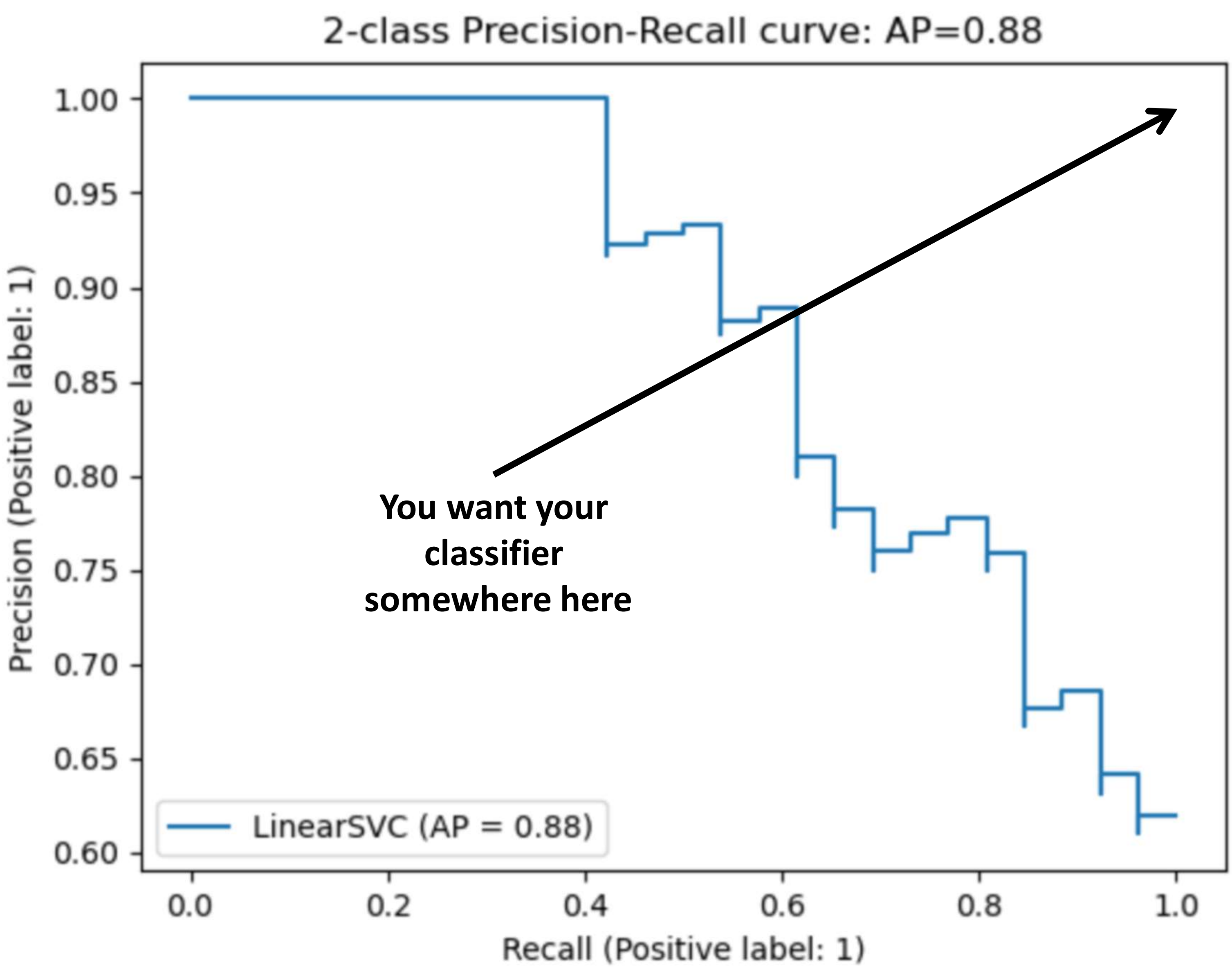


TPR: Sensitivity | FPR: 1 - Specificity

Receiver Operating Characteristic



Precision - Recall Curve



ROC vs. Precision-Recall Curves

- Both summarize model performance using different probability thresholds
- ROC curves should be used when there are roughly equal numbers of observations for each class
- Precision-Recall curves should be used when there is a moderate to large class imbalance (when we are interested in the positive class and there's only a few positive samples)

K-Fold Cross-Validation

Validation

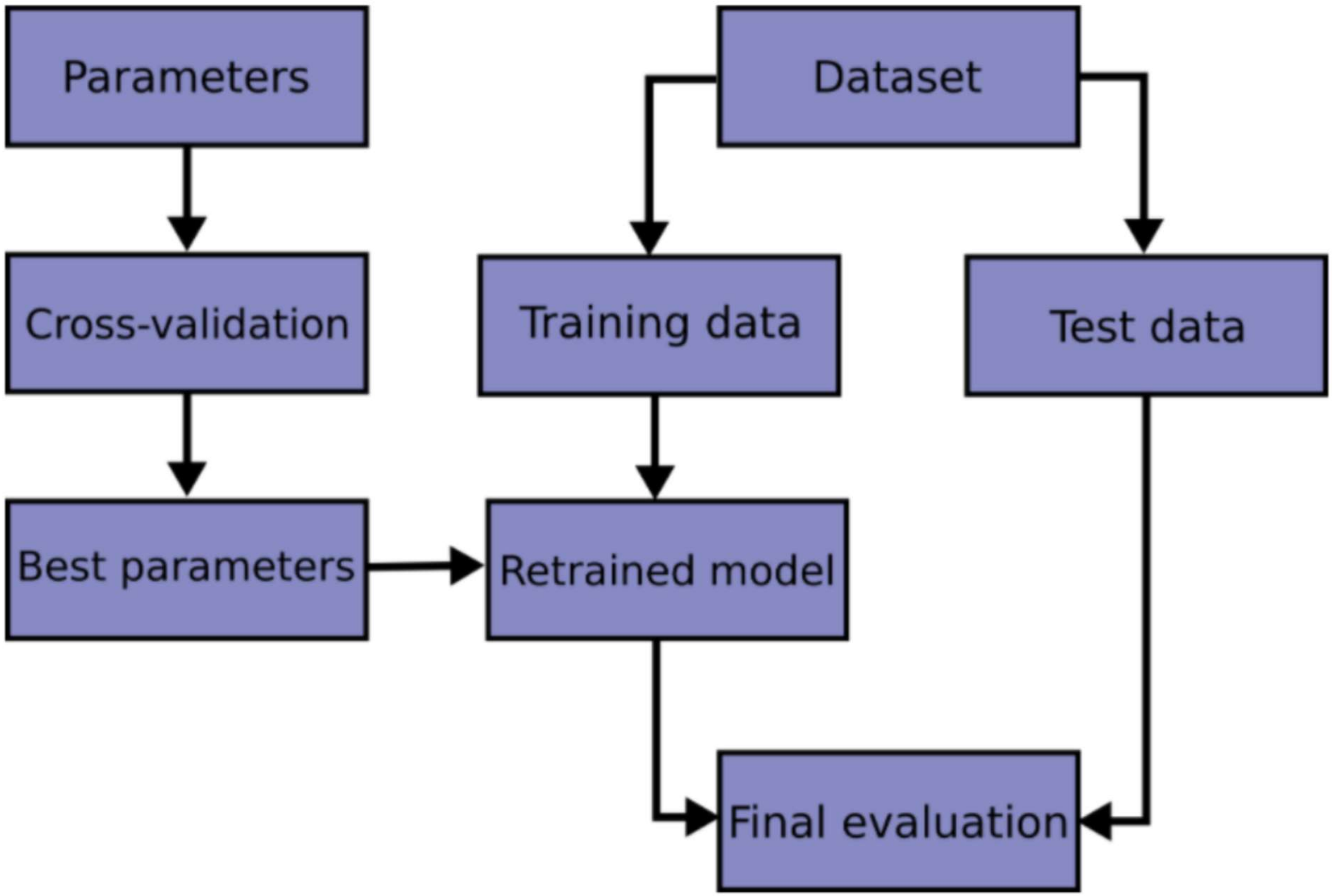
Train	Validate	Score

4-fold cross-validation

Train	Train	Train	Validate	ScoreA
Train	Train	Validate	Train	ScoreB
Train	Validate	Train	Train	ScoreC
Validate	Train	Train	Train	ScoreD

$$\text{Score} = \frac{\text{ScoreA} + \text{ScoreB} + \text{ScoreC} + \text{ScoreD}}{4}$$

Parameter Tuning



3-class Confusion Matrix

gold labels

system output		urgent	normal	spam
	urgent	8	10	1
	normal	5	60	50
	spam	3	30	200

$$\textbf{precision}_u = \frac{8}{8+10+1}$$

$$\textbf{precision}_n = \frac{60}{5+60+50}$$

$$\textbf{precision}_s = \frac{200}{3+30+200}$$

$$\textbf{recall}_u = \frac{8}{8+5+3} \qquad \textbf{recall}_n = \frac{60}{10+60+30} \qquad \textbf{recall}_s = \frac{200}{1+50+200}$$

Macroaveraging and Microaveraging

Macroaveraging:

- compute the performance for each class, and then average over classes

Microaveraging:

- collect decisions for all classes into one confusion matrix
- compute precision and recall from that table.

Macroaveraging and Microaveraging

Class 1: Urgent

	true urgent	true not
system urgent	8	11
system not	8	340

$$\text{precision} = \frac{8}{8+11} = .42$$

Class 2: Normal

	true normal	true not
system normal	60	55
system not	40	212

$$\text{precision} = \frac{60}{60+55} = .52$$

Class 3: Spam

	true spam	true not
system spam	200	33
system not	51	83

$$\text{precision} = \frac{200}{200+33} = .86$$

Pooled

	true yes	true no
system yes	268	99
system no	99	635

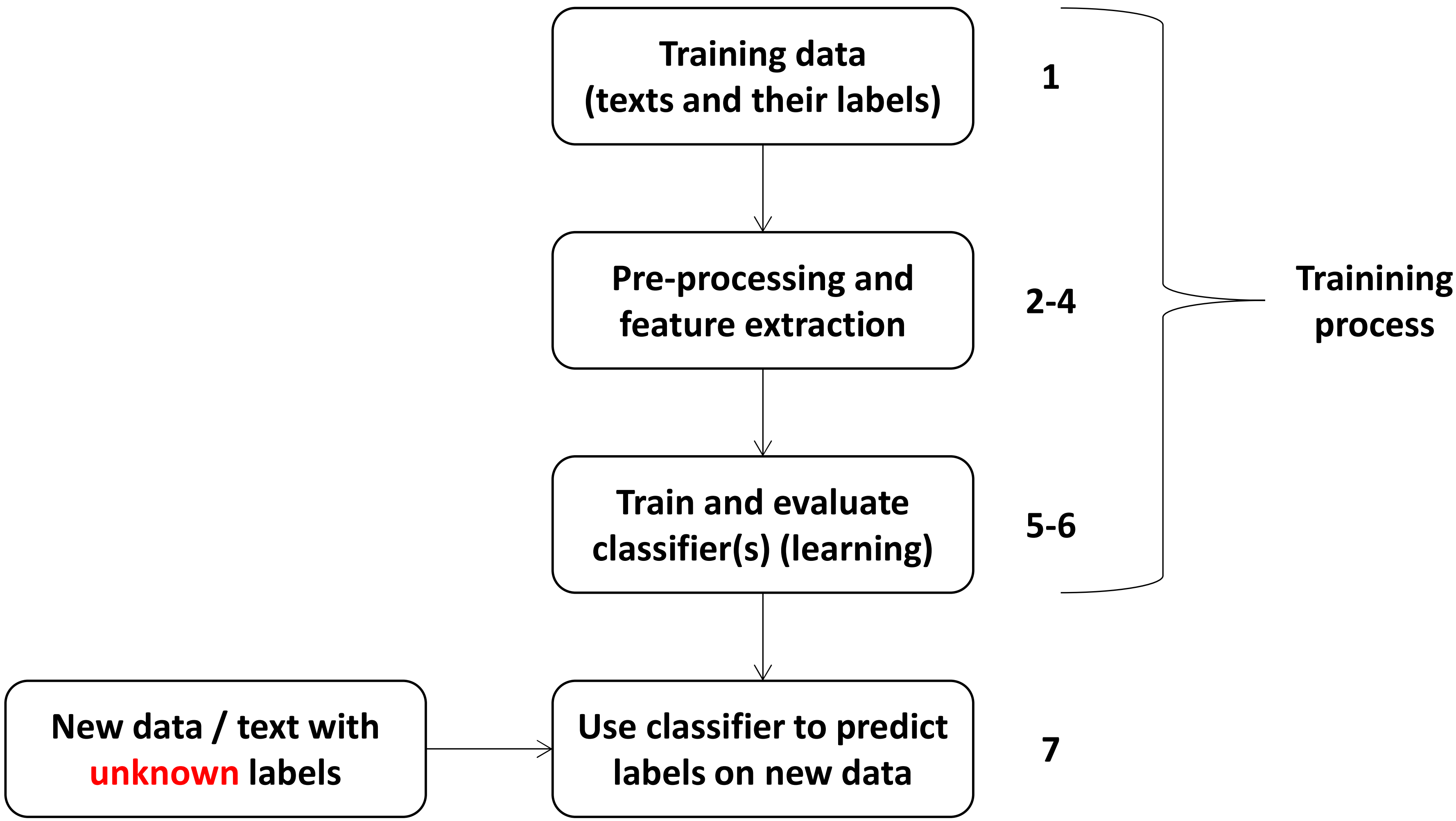
$$\begin{array}{c}\text{microaverage}\\\text{precision}\end{array} = \frac{268}{268+99} = \mathbf{.73}$$

$$\begin{array}{c}\text{macroaverage}\\\text{precision}\end{array} = \frac{.42+.52+.86}{3} = \mathbf{.60}$$

Text Classification System Pipeline

1. Obtain / collect / create labeled data set suitable for the task
2. Split the data set into:
   - two (training and test sets) parts OR
   - three (training, validation, and test sets) parts
3. Choose evaluation metric
4. Transform raw text into feature vectors:
   - bag of words
   - other types
5. Using feature vectors and labels from the training set, train the classifier / create a model
6. Using evaluation metric from (3) benchmark the classifier / model performance using the test set
7. Deploy the classifier / model to serve a real world application and monitor its performance

Text Classification System Pipeline



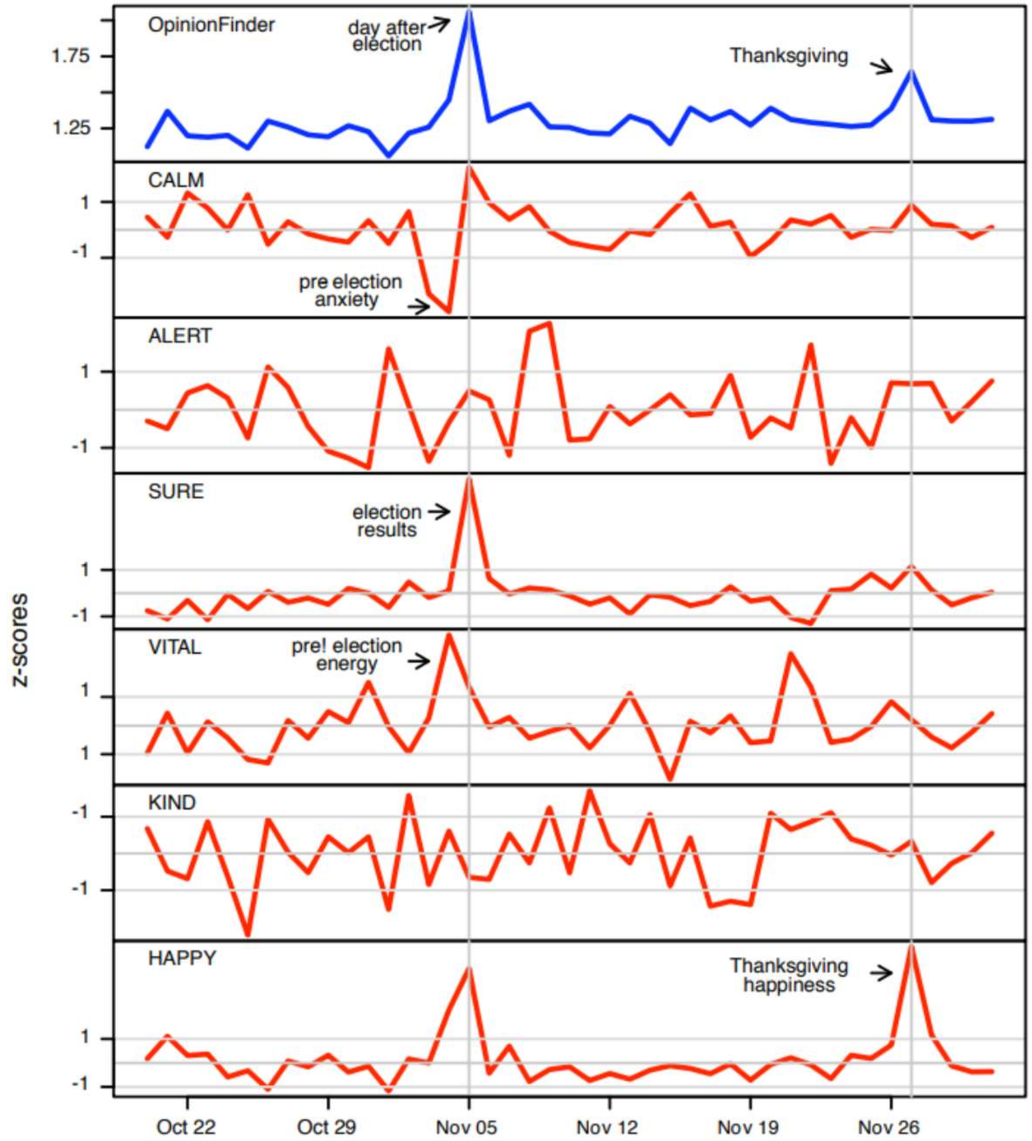
Poor Classifier Performance: Reasons

1. With all possible features extracted, we ended up with a sparse feature vector (some features are too rare and end up being noise) $\rightarrow$ makes training hard
2. Few (~20%) relevant samples compared to non-relevant (~80%) samples in the data set $\rightarrow$ skews learning towards non-relevant data
3. Need better learning algorithm
4. Need better pre-processing / feature extraction
5. Classifier parameters / hyperparameters need tuning

Sentiment Analysis: Motivation

- Movie: is this review positive or negative?
- Products: what do people think about the new iPhone?
- Public sentiment: what is consumer confidence?
- Politics: what do people think about this candidate or issue?
- Prediction: predict election outcomes or market trends from sentiment

Sentiment Analysis: Twitter Mood

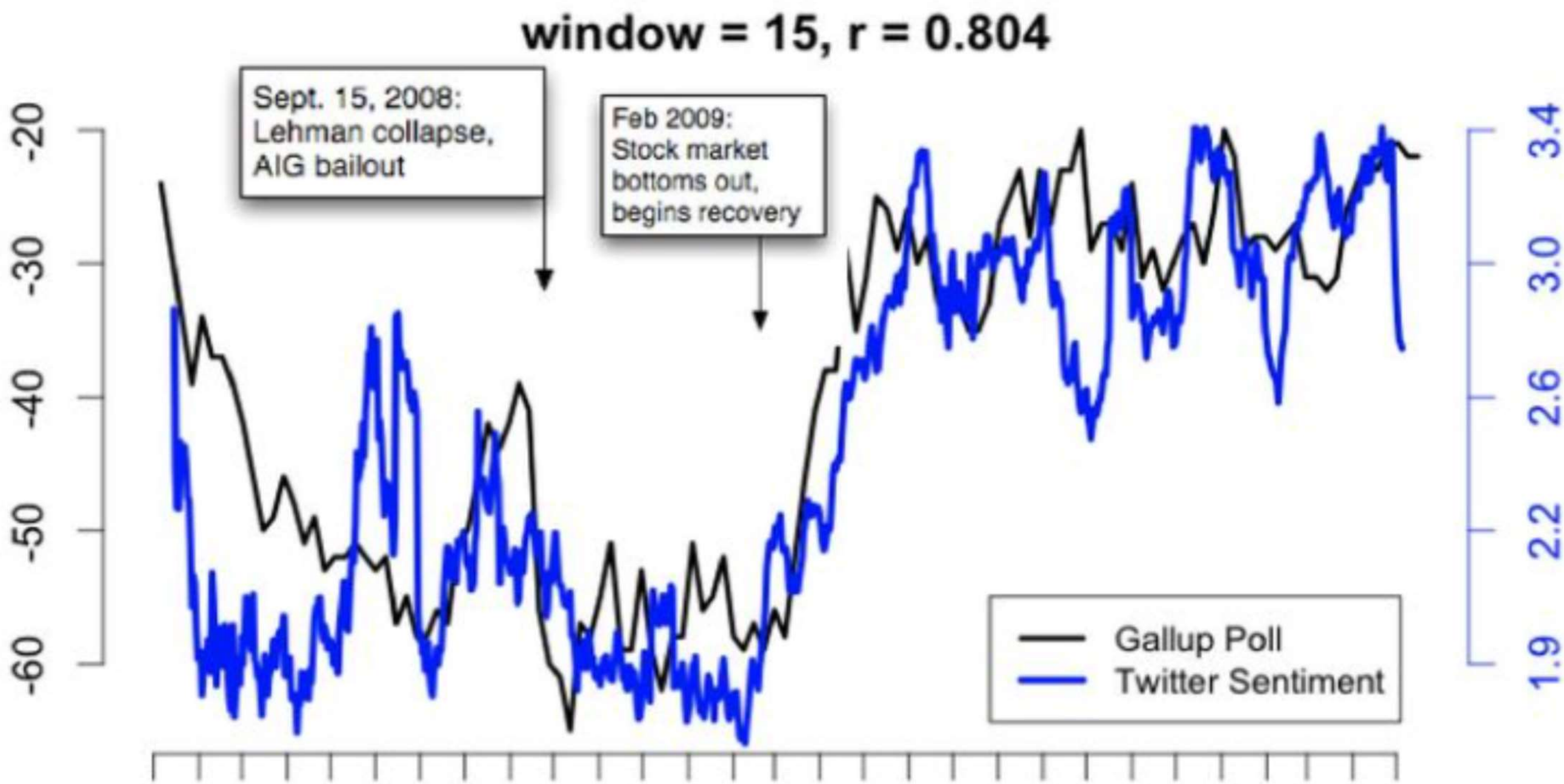


source: https://arxiv.org/pdf/1010.3003.pdf

Sentiment Analysis: Tweets

source: https://www.csc2.ncsu.edu/faculty/healey/tweet_viz/tweet_app/

Sentiment Analysis: Text and Polls



source: https://www.aaai.org/ocs/index.php/ICWSM/ICWSM10/paper/viewFile/1536/1842

Affective State

Affective state: any kind of **sentimental condition**, often in which someone's feelings control their consciousness.

Scherer Typology of Affective States

- Emotion: brief organically synchronized … evaluation of a major event
  - angry, sad, joyful, fearful, ashamed, proud, elated
- Mood: diffuse non-caused low-intensity long-duration change in subjective feeling
  - cheerful, gloomy, irritable, listless, depressed, buoyant
- Interpersonal stances: affective stance toward another person in a specific interaction
  - friendly, flirtatious, distant, cold, warm, supportive, contemptuous
- Attitudes: enduring, affectively colored beliefs, dispositions towards objects or persons
  - liking, loving, hating, valuing, desiring
- Personality traits: stable personality dispositions and typical behavior tendencies
  - nervous, anxious, reckless, morose, hostile, jealous

Scherer Typology of Affective States

- Emotion: brief organically synchronized … evaluation of a major event
  - angry, sad, joyful, fearful, ashamed, proud, elated
- Mood: diffuse non-caused low-intensity long-duration change in subjective feeling
  - cheerful, gloomy, irritable, listless, depressed, buoyant
- Interpersonal stances: affective stance toward another person in a specific interaction
  - friendly, flirtatious, distant, cold, warm, supportive, contemptuous
- Attitudes: enduring, affectively colored beliefs, dispositions towards objects or persons
  - liking, loving, hating, valuing, desiring
- Personality traits: stable personality dispositions and typical behavior tendencies
  - nervous, anxious, reckless, morose, hostile, jealous

Sentiment Analysis: Detecting Attitude

Sentiment analysis deals with detection of attitudes ("enduring, affectively colored beliefs, dispositions towards objects or persons"). It involves four components:

1. **Holder (source)** of attitude
2. **Target (aspect)** of attitude
3. **Type** of attitude:
   - **from a set of types:** *like*, *love*, *hate*, *value*, *desire*, **etc.**
   - **simple (weighted) polarity:** *positive*, *negative*, *neutral* **(together with** *strength***)**
4. **Text / data** containing attitude:
   - **sentences,**
   - **entire documents, etc.**

Sentiment Analysis: Other Names

- **Opinion extraction**
- **Opinion mining**
- **Sentiment mining**
- **Subjectivity analysis**

Sentiment Analysis: Complexity Levels

Sentiment analysis tasks can have different levels of complexity:

- **Simple tasks:**
  - **decide if the attitude contained within text is positive or negative**
- **More complex / average complexity tasks:**
  - **rank that attitude on a scale from 1 to 5**
- **Advanced / complex tasks:**
  - **detect target (stance detection)**
  - **detect source**
  - **detect complex attitude types**

Sentiment Analysis: Baseline Algo

- Pre-processing: tokenization
- Feature extraction: bag of words, etc.
- Classification using a chosen classifier
  - Naive Bayes
  - Perceptron
  - Support Vector Machines
  - etc.

Naive Bayes Classifier

$$\text{category/class} = h(\textbf{document})$$

Finding model / hypothesis h $\rightarrow$ **Finding probabilities for** y_{MAP}

$$y_{MAP} \propto \underset{y \in Y}{argmax} \left(\boxed{P(y)} * \prod_{i=1}^{N} \boxed{P(x_i \mid y)} \right)$$

Note: logarithms (to handle underflow) will be ignored in the following example

MAP: Maximum a posteriori (corresponds to the most likely class).

Sentiment Analysis: Example

Training set

$\mathbf{x}_1$	just plain boring	$y_1 = -$
$\mathbf{x}_2$	entirely predictable and lacks energy	$y_2 = -$
$\mathbf{x}_3$	no surprises and very few laughs	$y_3 = -$
$\mathbf{x}_4$	very powerful	$y_4 = +$
$\mathbf{x}_5$	the most fun film of the summer	$y_5 = +$

Test set

$\mathbf{x}_6$	predictable with no fun	$y_6 = ???$

$\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_{N-2}, \mathbf{x}_{N-1}, \mathbf{x}_N$ - feature vectors (in **bold**) | $y_1, y_2, y_3, \ldots, y_{N-2}, y_{N-1}, y_N$ - labels

Sentiment Analysis: Example

Training set

$\mathbf{x}_1$	**just plain boring**	$y_1 = -$
$\mathbf{x}_2$	**entirely predictable and lacks energy**	$y_2 = -$
$\mathbf{x}_3$	**no surprises and very few laughs**	$y_3 = -$
$\mathbf{x}_4$	**very powerful**	$y_4 = +$
$\mathbf{x}_5$	**the most fun film of the summer**	$y_5 = +$

Test set

$\mathbf{x}_6$	**predictable with no fun**	$y_6 = ???$

Learning

Probability estimates:

Naive Bayes Classifier:

$$y_{MAP} \propto \underset{y \in Y}{argmax} \left(\boldsymbol{P}(y) * \prod_{i=1}^{N} \boldsymbol{P}(x_i \mid y) \right)$$

Probability estimates (Maximum Likelihood estimation):

$$P(y = +) = \frac{N_{samples\ labeled\ +}}{N}$$

$$P(y = -) = \frac{N_{samples\ labeled\ -}}{N}$$

$$P(x_i = word \mid y = CLASS) = \\ = \frac{count(x_i = word, y = CLASS) + 1}{\sum_{x \in V} count(x, y = CLASS) + |V|}$$

Note the add-1 smoothing!

$\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_{N-2}, \mathbf{x}_{N-1}, \mathbf{x}_N$ - feature vectors (in **bold**) | $y_1, y_2, y_3, \ldots, y_{N-2}, y_{N-1}, y_N$ - labels

Sentiment Analysis: Example

Training set

$\mathbf{x}_1$	just plain boring	$y_1 = -$
$\mathbf{x}_2$	entirely predictable and lacks energy	$y_2 = -$
$\mathbf{x}_3$	no surprises and very few laughs	$y_3 = -$
$\mathbf{x}_4$	very powerful	$y_4 = +$
$\mathbf{x}_5$	the most fun film of the summer	$y_5 = +$

Test set

$\mathbf{x}_6$	predictable with no fun	$y_6 = ???$

Learning

$$P(y = +) = \frac{N_{samples\ labeled\ +}}{N} = \frac{2}{5}$$

$$P(y = -) = \frac{N_{samples\ labeled\ -}}{N} = \frac{3}{5}$$

$\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_{N-2}, \mathbf{x}_{N-1}, \mathbf{x}_N$ - feature vectors (in **bold**) | $y_1, y_2, y_3, \ldots, y_{N-2}, y_{N-1}, y_N$ - labels

Sentiment Analysis: Example

Training set

$\mathbf{x}_1$	just plain boring	y_1= -
$\mathbf{x}_2$	entirely predictable and lacks energy	y_2= -
$\mathbf{x}_3$	no surprises and very few laughs	y_3= -
$\mathbf{x}_4$	very powerful	y_4= +
$\mathbf{x}_5$	the most fun film of the summer	y_5= +

Test set

$\mathbf{x}_6$	predictable with no fun	y_6=???

Learning

word	+	-
just	0	1
plain	0	1
boring	0	1
entirely	0	1
predictable	0	1
and	0	2
lacks	0	1
energy	0	1
no	0	1
surprises	0	1
very	1	1
few	0	1
laughs	0	1
powerful	1	0
the	2	0
most	1	0
fun	1	0
film	1	0
of	1	0
summer	1	0

$\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_{N-2}, \mathbf{x}_{N-1}, \mathbf{x}_N$ - feature vectors (in **bold**) | $y_1, y_2, y_3, \ldots, y_{N-2}, y_{N-1}, y_N$ - labels

Sentiment Analysis: Example

Training set

$\mathbf{x}_1$	just plain boring	y_1= -
$\mathbf{x}_2$	entirely predictable and lacks energy	y_2= -
$\mathbf{x}_3$	no surprises and very few laughs	y_3= -
$\mathbf{x}_4$	very powerful	y_4= +
$\mathbf{x}_5$	the most fun film of the summer	y_5= +

Test set

$\mathbf{x}_6$	predictable ~~with~~ no fun	y_6=???

unknown word (not in V) → ignore

Learning

$$P(x_i = predictable \mid y = +) = \frac{count(x_i = predictable, y = +) + 1}{\sum_{x \in V} count(x, y = +) + 20} = \frac{0+1}{9+20} = \frac{1}{29}$$

$$P(x_i = no \mid y = +) = \frac{count(x_i = no, y = +) + 1}{\sum_{x \in V} count(x, y = +) + 20} = \frac{0+1}{9+20} = \frac{1}{29}$$

$$P(x_i = fun \mid y = +) = \frac{count(x_i = fun, y = +) + 1}{\sum_{x \in V} count(x, y = +) + 20} = \frac{1+1}{9+20} = \frac{2}{29}$$

$$P(x_i = predictable \mid y = -) = \frac{count(x_i = predictable, y = -) + 1}{\sum_{x \in V} count(x, y = -) + 20} = \frac{1+1}{14+20} = \frac{2}{34}$$

$$P(x_i = no \mid y = -) = \frac{count(x_i = no, y = -) + 1}{\sum_{x \in V} count(x, y = -) + 20} = \frac{1+1}{14+20} = \frac{2}{34}$$

$$P(x_i = fun \mid y = -) = \frac{count(x_i = fun, y = -) + 1}{\sum_{x \in V} count(x, y = -) + 20} = \frac{0+1}{14+20} = \frac{1}{34}$$

Other probabilities not shown

$\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_{N-2}, \mathbf{x}_{N-1}, \mathbf{x}_N$ - feature vectors (in **bold**) | $y_1, y_2, y_3, \ldots, y_{N-2}, y_{N-1}, y_N$ - labels

Sentiment Analysis: Example

Training set

$\mathbf{x}_1$	just plain boring	y_1= -
$\mathbf{x}_2$	entirely predictable and lacks energy	y_2= -
$\mathbf{x}_3$	no surprises and very few laughs	y_3= -
$\mathbf{x}_4$	very powerful	y_4= +
$\mathbf{x}_5$	the most fun film of the summer	y_5= +

Test set

$\mathbf{x}_6$	predictable ~~with~~ no fun	y_6=???

unknown word (not in V) → ignore

Model

$$P(y = +) = \frac{2}{5}$$

$$P(y = -) = \frac{3}{5}$$

$$P(x_i = predictable \mid y = +) = \frac{1}{29}$$

$$P(x_i = no \mid y = +) = \frac{1}{29}$$

$$P(x_i = fun \mid y = +) = \frac{2}{29}$$

$$P(x_i = predictable \mid y = -) = \frac{2}{34}$$

$$P(x_i = no \mid y = -) = \frac{2}{34}$$

$$P(x_i = fun \mid y = -) = \frac{1}{34}$$

Other probabilities not shown

$\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_{N-2}, \mathbf{x}_{N-1}, \mathbf{x}_N$ - feature vectors (in **bold**) | $y_1, y_2, y_3, \ldots, y_{N-2}, y_{N-1}, y_N$ - labels

Sentiment Analysis: Example

Training set

$\mathbf{x}_1$	just plain boring	$y_1 = -$
$\mathbf{x}_2$	entirely predictable and lacks energy	$y_2 = -$
$\mathbf{x}_3$	no surprises and very few laughs	$y_3 = -$
$\mathbf{x}_4$	very powerful	$y_4 = +$
$\mathbf{x}_5$	the most fun film of the summer	$y_5 = +$

Test set

$\mathbf{x}_6$	predictable ~~with~~ no fun	$y_6 = ???$

unknown word (not in V) → ignore

Prediction

$$P(y = +) = \frac{2}{5}$$

$$P(y = -) = \frac{3}{5}$$

$$P(x_i = predictable \mid y = +) = \frac{1}{29}$$

$$P(x_i = no \mid y = +) = \frac{1}{29}$$

$$P(x_i = fun \mid y = +) = \frac{2}{29}$$

$$P(x_i = predictable \mid y = -) = \frac{2}{34}$$

$$P(x_i = no \mid y = -) = \frac{2}{34}$$

$$P(x_i = fun \mid y = -) = \frac{1}{34}$$

Label + case:

$$P(y = +) * P(x_6 \mid y = +) = \frac{2}{5} * \frac{1}{29} * \frac{1}{29} * \frac{2}{29}$$
$$= 3.2 * 10^{-5}$$

Label - case:

$$P(y = -) * P(x_6 \mid y = -) = \frac{3}{5} * \frac{2}{34} * \frac{2}{34} * \frac{1}{34}$$
$$= 6.1 * 10^{-5}$$

$\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_{N-2}, \mathbf{x}_{N-1}, \mathbf{x}_N$ - feature vectors (in **bold**) | $y_1, y_2, y_3, \ldots, y_{N-2}, y_{N-1}, y_N$ - labels

Sentiment Analysis: Example

Training set

$\mathbf{x}_1$	just plain boring	y_1= -
$\mathbf{x}_2$	entirely predictable and lacks energy	y_2= -
$\mathbf{x}_3$	no surprises and very few laughs	y_3= -
$\mathbf{x}_4$	very powerful	y_4= +
$\mathbf{x}_5$	the most fun film of the summer	y_5= +

Test set

$\mathbf{x}_6$	predictable ~~with~~ no fun	y_6= -

unknown word (not in V) → ignore

Model

$$P(y = +) = \frac{2}{5}$$

$$P(y = -) = \frac{3}{5}$$

$$P(x_i = predictable \mid y = +) = \frac{1}{29}$$

$$P(x_i = no \mid y = +) = \frac{1}{29}$$

$$P(x_i = fun \mid y = +) = \frac{2}{29}$$

$$P(x_i = predictable \mid y = -) = \frac{2}{34}$$

$$P(x_i = no \mid y = -) = \frac{2}{34}$$

$$P(x_i = fun \mid y = -) = \frac{1}{34}$$

Label + case:

$$P(y = +) * P(\mathbf{x}_6 \mid y = +) = \frac{2}{5} * \frac{1}{29} * \frac{1}{29} * \frac{2}{29} = 3.2 * 10^{-5}$$

Label - case:

$$P(y = -) * P(\mathbf{x}_6 \mid y = -) = \frac{3}{5} * \frac{2}{34} * \frac{2}{34} * \frac{1}{34} = 6.1 * 10^{-5}$$

$\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_{N-2}, \mathbf{x}_{N-1}, \mathbf{x}_N$ - feature vectors (in **bold**) | $y_1, y_2, y_3, \ldots, y_{N-2}, y_{N-1}, y_N$ - labels

Sentiment Analysis: Feature Extraction

- **What should we extract?**
  - **just adjectives?**
  - **or perhaps all words?**

- **Extracting all words is likely to work better**

Optimizing for Sentiment Analysis

- **For tasks like sentiment, word occurrence seems to be more important than word frequency**
  - **the occurrence of the word *fantastic* tells us a lot**
  - **the fact that it occurs 5 times may not tell us much more.**

- **Use binary multinominal Naive Bayes, or binary NB**
  - **clip word counts at 1 (1 per document even if more)**

Sentiment Analysis: Binarization

Four original documents:

– it was pathetic the worst part was the boxing scenes
– no plot twists or great scenes
+ and satire and great plot twists
+ great scenes great film

After per-document binarization:

– it was pathetic the worst part boxing scenes
– no plot twists or great scenes
+ and satire great plot twists
+ great scenes film

	NB Counts		Binary Counts	
	+	−	+	−
and	2	0	1	0
boxing	0	1	0	1
film	1	0	1	0
great	3	1	2	1
it	0	1	0	1
no	0	1	0	1
or	0	1	0	1
part	0	1	0	1
pathetic	0	1	0	1
plot	1	1	1	1
satire	1	0	1	0
scenes	1	2	1	2
the	0	2	0	1
twists	1	1	1	1
was	0	2	0	1
worst	0	1	0	1

Sentiment Analysis: Binarization

Four original documents:

- – it was pathetic the worst part was the boxing scenes
- – no plot twists or great scenes
- + and satire and great plot twists
- + great scenes great film

After per-document binarization:

- – it was pathetic the worst part boxing scenes
- – no plot twists or great scenes
- + and satire great plot twists
- + great scenes film

duplicate words removed

	NB Counts +	NB Counts −	Binary Counts +	Binary Counts −
and	2	0	1	0
boxing	0	1	0	1
film	1	0	1	0
great	3	1	2	1
it	0	1	0	1
no	0	1	0	1
or	0	1	0	1
part	0	1	0	1
pathetic	0	1	0	1
plot	1	1	1	1
satire	1	0	1	0
scenes	1	2	1	2
the	0	2	0	1
twists	1	1	1	1
was	0	2	0	1
worst	0	1	0	1

Sentiment Analysis: Binarization

Four original documents:

- – it was pathetic the worst part was the boxing scenes
- – no plot twists or great scenes
- + and satire and great plot twists
- + great scenes great film

After per-document binarization:

- – it was pathetic the worst part boxing scenes
- – no plot twists or great scenes
- + and satire great plot twists
- + great scenes film

	NB Counts		Binary Counts	
	+	–	+	–
and	2	0	1	0
boxing	0	1	0	1
film	1	0	1	0
great	3	1	2	1
it	0	1	0	1
no	0	1	0	1
or	0	1	0	1
part	0	1	0	1
pathetic	0	1	0	1
plot	1	1	1	1
satire	1	0	1	0
scenes	1	2	1	2
the	0	2	0	1
twists	1	1	1	1
was	0	2	0	1
worst	0	1	0	1

great **and** *scenes* **appear in multiple documents**

Sentiment Analysis: Negations

Consider the following sentences / documents:

- *I really like this movie*
- *I really don't like this movie*

- **Negation changes the meaning of "like" to negative.**
- **Negation can also change negative to positive-ish**
  - *Don't dismiss this film*
  - *Doesn't let us get bored*

Sentiment Analysis: Negations

Simple baseline method solution:

- **Add NOT_ prefix to every word between negation and following punctuation:**

didn't like this movie , but I

didn't NOT_like NOT_this NOT_movie , but I

Sentiment Analysis: Lexicons

What if we don't have enough labeled training data?

- **In that case, we can make use of pre-built word lists $\rightarrow$ lexicons**

There are various publicly available lexicons

Sentiment Analysis: Lexicons

Add a feature that gets a count whenever a word from the lexicon occurs:

- **for example: a feature called "this word occurs in the positive lexicon" or "this word occurs in the negative lexicon"**
- **all positive words (good, great, beautiful, wonderful) or negative words count for that feature.**

Using 1-2 features isn't as good as using all the words.

But when training data is sparse or not representative of the test set, dense lexicon features can help

MPQA Subjectivity Cues Lexicon

Home page:

https://mpqa.cs.pitt.edu/lexicons/subj_lexicon/

6885 words from 8221 lemmas, annotated for intensity (strong/weak)

2718 positive

4912 negative

+ : admirable, beautiful, confident, dazzling, ecstatic, favor, glee, great

– : awful, bad, bias, catastrophe, cheat, deny, envious, foul, harsh, hate

The General Inquirer

Home page:

http://www.wjh.harvard.edu/~inquirer

List of Categories:

http://www.wjh.harvard.edu/~inquirer/homecat.htm

Categories:

- **Positive (1915 words) and Negative (2291 words)**
- **Strong vs Weak, Active vs Passive, Overstated versus Understated**
- **Pleasure, Pain, Virtue, Vice, Motivation, Cognitive Orientation, etc**

Free for Research Use!

Sentiment Analysis: Lexicons

What if there is no lexicon for a given domain?

- **In that case, we can learn it:**
  - **start with a few labeled examples and hand-built patterns**
  - **build lexicon using the procedure below:**

function BUILDSENTIMENTLEXICON(*posseeds*,*negseeds*) **returns** *poslex*,*neglex*

poslex ← *posseeds*
neglex ← *negseeds*
Until done
  poslex ← *poslex* + FINDSIMILARWORDS(*poslex*)
  neglex ← *neglex* + FINDSIMILARWORDS(*neglex*)
poslex,*neglex* ← POSTPROCESS(*poslex*,*neglex*)

Identifying Word Polarity

Intuition:

- **words joined by "*and*" have the same polarity:**
  - *fair **and** legitimate, corrupt **and** brutal*

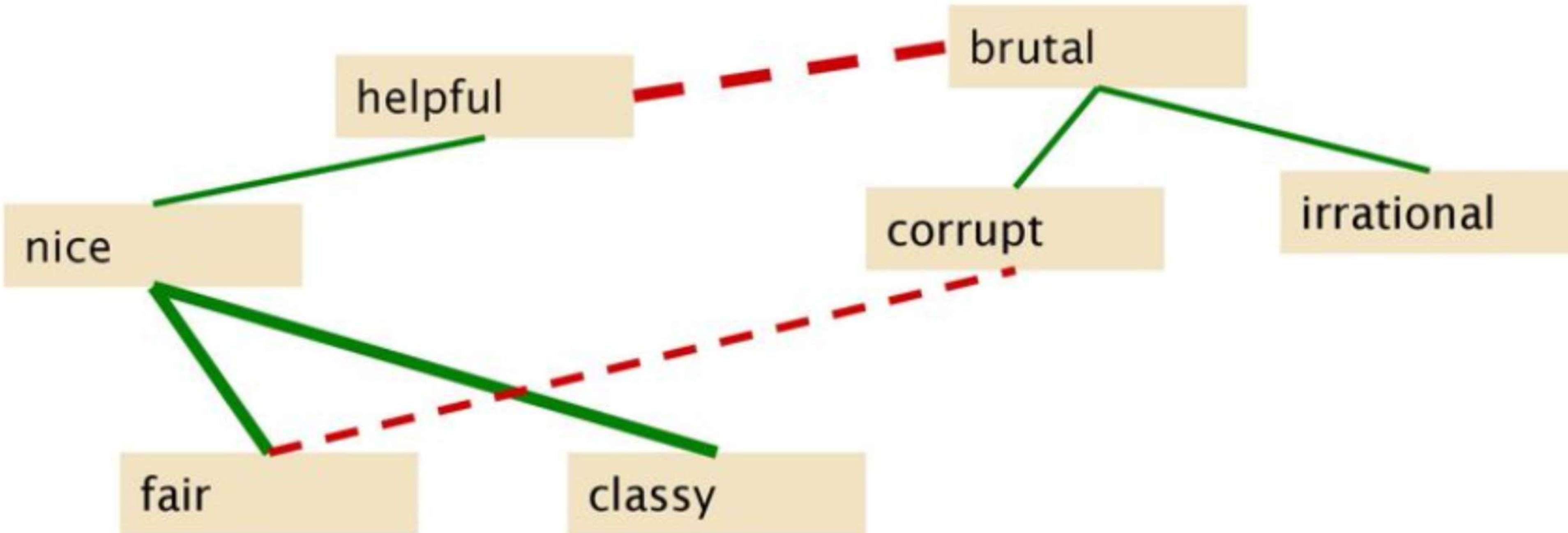
- **words joined by "*but*" do not:**
  - *fair, **but** brutal*

Sentiment Analysis: Learning Lexicons

- **Motivation:**
  - **learn a domain-specific lexicon**
  - **more words (more robust) than off-the-shelf lexicon**

- **Intuition:**
  - **start with "seed" words (*good*, *poor*)**
  - **find other words with similar polarity:**
    - **use "*and*" and "*but*"**
    - **use nearby words in the same document**
    - **add them to lexicon**

Identifying Word Polarity: Graph



Sentiment Analysis: Challenges

Subtlety:

- **Perfume review in** Perfumes: The Guide:
  - *"If you are reading this because it is your darling fragrance, please wear it at home exclusively and tape the windows shut"*
- **Dorothy Parker (writer) on Katherine Hepburn (actress):**
  - *"She runs the gamut of emotions from A to B"*

Sentiment Analysis: Challenges

Thwarted expectations:

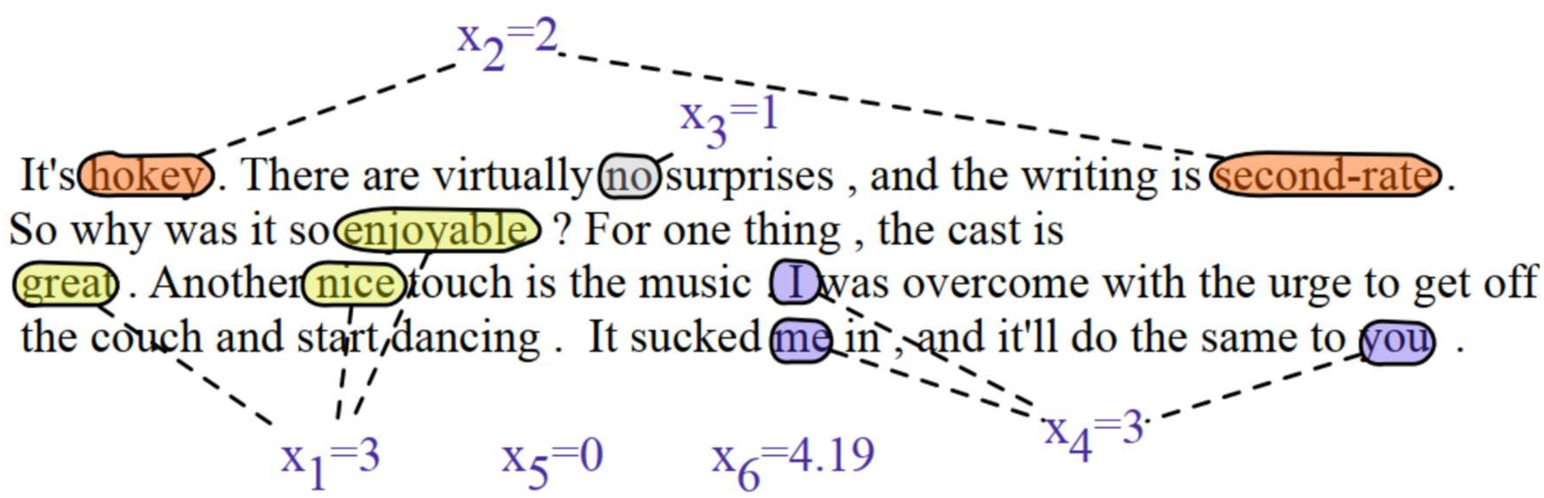
- *"This film should be **brilliant**. It sounds like a **great** plot, the actors are **first grade**, and the supporting cast is **good** as well, and Stallone is attempting to deliver a **good** performance. However it **can't hold up**."*

Sentiment Analysis: Challenges

Ordering effect:

- “*Well as usual Keanu Reeves is nothing special, but surprisingly, the* ***very talented*** *Laurence Fishbourne* ***not so good*** *either. I was surprised.*”

Sentiment Analysis: Example



Feature vector:

Var	Definition	Value
x_1	count(positive lexicon) $\in$ doc)	3
x_2	count(negative lexicon) $\in$ doc)	2
x_3	$\begin{cases} 1 & \text{if “no”} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$	1
x_4	count(1st and 2nd pronouns $\in$ doc)	3
x_5	$\begin{cases} 1 & \text{if “!”} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$	0
x_6	log(word count of doc)	$\ln(66) = 4.19$

Sentiment Analysis: Example

Feature vector x:

Var	Definition	Value
x_1	count(positive lexicon) $\in$ doc)	3
x_2	count(negative lexicon) $\in$ doc)	2
x_3	$\begin{cases} 1 & \text{if "no"} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$	1
x_4	count(1st and 2nd pronouns $\in$ doc)	3
x_5	$\begin{cases} 1 & \text{if "!"} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$	0
x_6	log(word count of doc)	$\ln(66) = 4.19$

What is w·x + b ???

Suppose: w = [2.5, 5.0, 1.2, 0.5, 2.0, 0.7]
and b = 0.1

Sentiment Analysis: Example

$$
\begin{aligned}
p(+|x) = P(Y=1|x) &= \sigma(w \cdot x + b) \\
&= \sigma([2.5, -5.0, -1.2, 0.5, 2.0, 0.7] \cdot [3, 2, 1, 3, 0, 4.19] + 0.1) \\
&= \sigma(.833) \\
&= 0.70
\end{aligned}
$$

Positive sentiment

$$
\begin{aligned}
p(-|x) = P(Y=0|x) &= 1 - \sigma(w \cdot x + b) \\
&= 0.30
\end{aligned}
$$

Period Disambiguation: Example

End of sentence

This ends in a period.
The house at 465 Main St. is new.

Not end

Feature vector **x**

$$x_1 = \begin{cases} 1 & \text{if "}\mathit{Case}(w_i) = \text{Lower"} \\ 0 & \text{otherwise} \end{cases}$$

$$x_2 = \begin{cases} 1 & \text{if "}w_i \in \text{AcronymDict"} \\ 0 & \text{otherwise} \end{cases}$$

$$x_3 = \begin{cases} 1 & \text{if "}w_i = \text{St. \& } \mathit{Case}(w_{i-1}) = \text{Cap"} \\ 0 & \text{otherwise} \end{cases}$$

What do Words Mean?

In methods (N-grams, text classification, etc.) we've seen:

- words are <u>just strings</u>
- **meaning** is not considered

Meaning in logic:

- The meaning of "dog" is DOG (predicates and symbols)

$$\forall x\ \mathrm{DOG}(x) \rightarrow \mathrm{MAMMAL}(x)$$

Old linguistics joke by Barbara Partee in 1967:

- Q: What's the **meaning** of life?
- A: LIFE

That is not very helpful.

Words: Lemmas and Senses

lemma

senses

mouse (Noun)

1. any of numerous small rodents...
2. a hand-operated device that controls a cursor...

from the online thesaurus WordNet

Words: Lemmas and Senses

lemma

senses

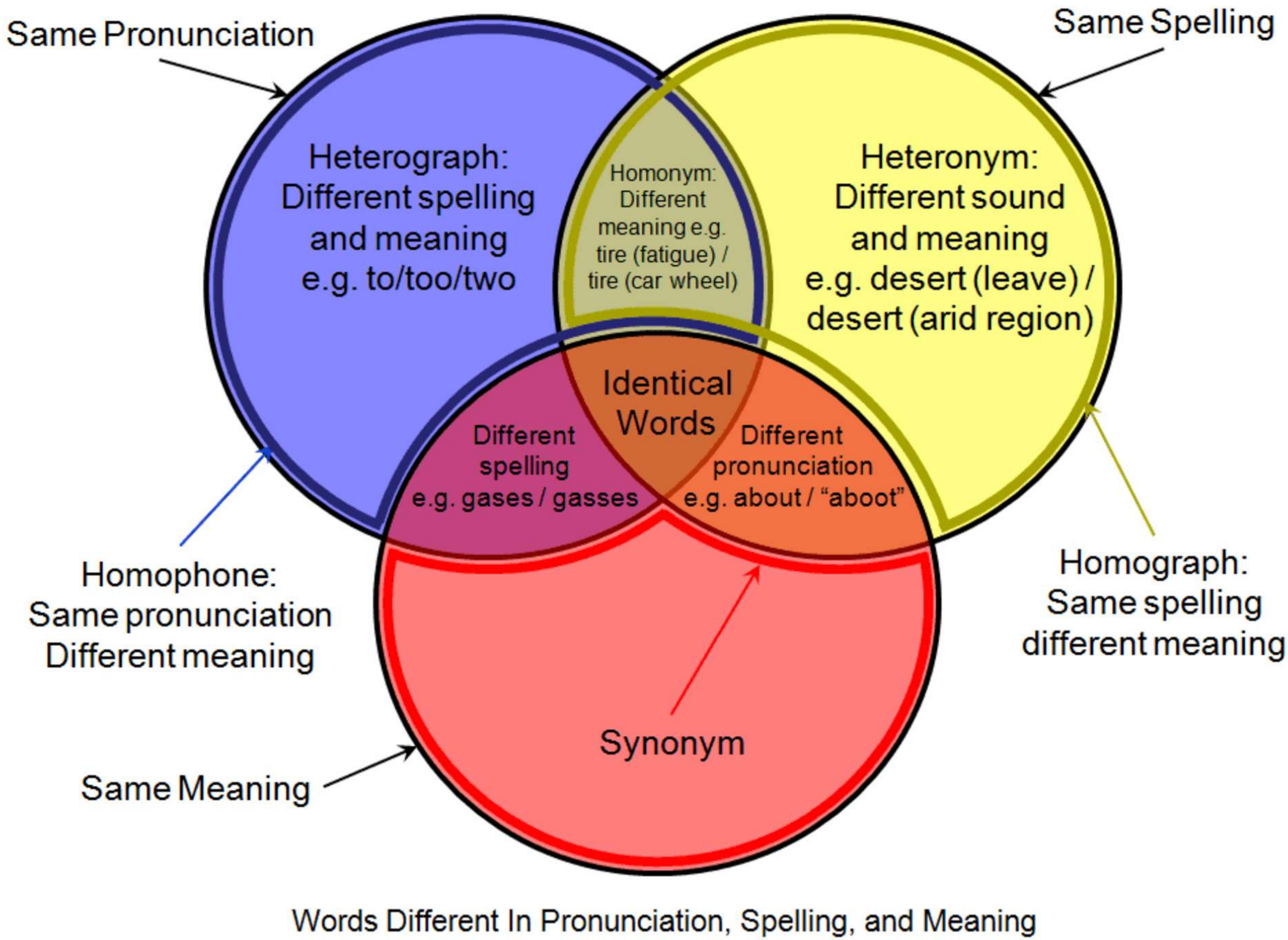
mouse (Noun)

1. any of numerous small rodents...
2. a hand-operated device that controls a cursor...

from the online thesaurus WordNet

- **A sense or "concept" is the meaning component of a word**
- **Lemmas can be polysemous (have multiple senses)**

Relationships Between Words



Source: https://owlcation.com/humanities/Lexical-Relations-Describing-Similarities-In-The-English-Language

Lexical Relationships

Lexical relationships are the connections established between one word and another:

- **Synonymy is the idea that some words have the same meaning as others**
  - *quick* **is similar to** *fast*
- **Antonymy is precisely the opposite of synonymy**
  - *good* **is the opposite** *bad*
- **Hyponymy is similar to the notion of embeddedness**
  - *Human* ← *Female* **(***Female* **is a more specific concept than** *Human***)**
- **Holonomy and Meronomy describe relationships between an object and its parts:**
  - *tree* **is a holonym of** *bark* **(***tree* **has bark)**
  - *bark* **is a meronym of** *tree* **(***bark* **is a part of** *tree***)**

Lexical Semantics: Definition

Lexical semantics:

*the branch of linguistics and logic concerned with **meaning**. There are a number of branches and subbranches of semantics, including:*

- **formal semantics**, *which studies the logical aspects of **meaning**, such as **sense**, **reference**, **implication**, and **logical form**,*
- **lexical semantics**, *which studies **word meanings** and **word relations**, and **conceptual semantics**, which studies the **cognitive structure of meaning**.*

from Oxford Dictionary

Sense Relationships: Synonymy

- **Synonyms have the same meaning in some or all contexts:**
  - *filbert / hazelnut*
  - *couch / sofa*
  - *big / large*
  - *automobile / car*
  - *vomit / throw up*
  - *water / $H_2 0$*

Sense Relationships: Synonymy

- There are probably no examples of perfect synonymy:
  - many aspects of meaning maybe identical, but not necessarily all aspects
  - words may differ based on:
    - politeness
    - slang
    - register,
    - genre, etc.

Sense Relationships: Synonymy?

- **Some examples:**
  - *water / H_2O*

    Would "*H_2O*" be used in a surfing guide?
  - *car / automobile*
  - *big / large*
    - *my big sister* **is NOT always going to be synonymous with** *my large sister*

The Linguistic Principle of Contrast

- **Substitutions between some pairs of words like *car / automobile* or *water / H_2O* are truth preserving, the words are still not identical in meaning**

- **The Linguistic Principle of Contrast**

  difference in form → difference in meaning

Sense Relationships: Similarity

- **Words with similar meanings.**
- **<u>Not</u> synonyms, but sharing some element of meaning**

- **Some examples:**
  - *cow / horse*
  - *car / bicycle*

Sense Relationships: Similarity

- Knowing how similar two words are can:
  - help in computing how similar the meaning of two phrases or sentences are
  - assist in higher level tasks:
    - question answering
    - paraphrasing
    - summarization

Sense Relationships: Similarity

Human-evaluated word similarity:

Word 1	Word 2	Similarity [0-10]
vanish	*disappear*	9.8
behave	*obey*	7.3
belief	*impression*	5.95
muscle	*bone*	3.65
modest	*flexible*	0.98
hole	*agreement*	0.3

Source: SimLex-999 dataset (Hill et al., 2015) | https://fh295.github.io/simlex.html

Sense Relationships: Relatedness

- **Also called "word association"**
- **Words can be related in any way, for example via a semantic frame or field**

- **Some examples:**
  - *coffee, tea:* **similar**
  - *coffee, cup:* **related, <u>not</u> similar**

Semantic Frame: Definition

Semantic Frame:

a semantic frame is defined as a ***coherent structure of concepts that are related*** *such that* ***without knowledge of all of them, one does not have complete knowledge of one*** *of the either*

from *https://cogling.fandom.com/wiki/Semantic_frame*

Semantic Field: Definition

Semantic Field:

*a **lexical set** of semantically **related items***

from Oxford Dictionary

Semantic Field

Words that

- **cover a particular semantic domain**
- **bear structured relations with each other.**

hospitals

surgeon, scalpel, nurse, anaesthetic, hospital

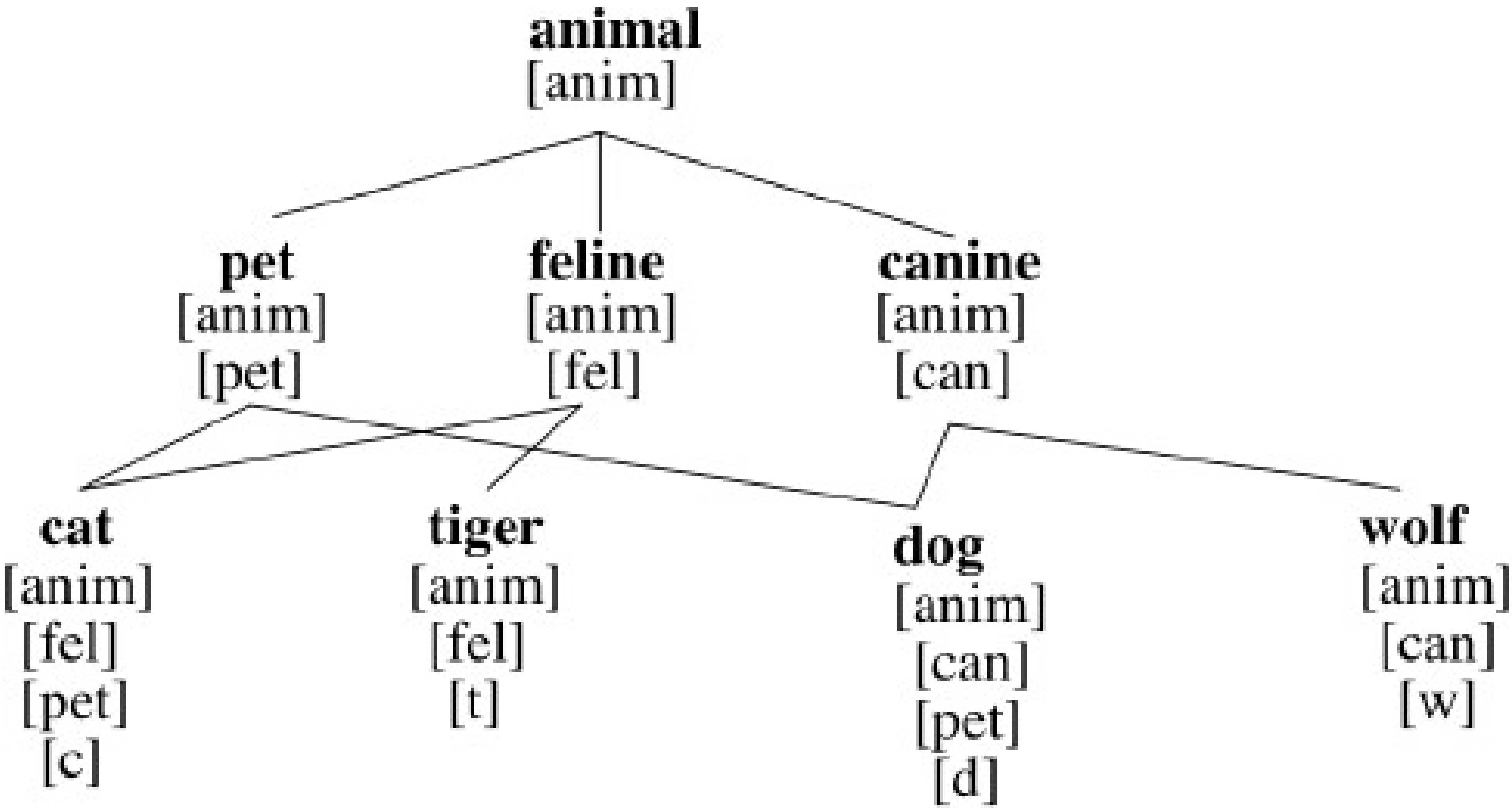
restaurants

waiter, menu, plate, food, menu, chef

houses

door, roof, kitchen, family, bed

Semantic Field



Source: Helge Dyvik - "Translations as a semantic knowledge source"

Sense Relationships: Antonymy

- **Senses that are opposites with respect to only one feature of meaning**

- **Otherwise, they are very similar (sharing some element of meaning):**
  - *dark/light* *short/long* *fast/slow*
  - *hot/cold* *up/down* *in/out*

Sense Relationships: Antonymy

- **More formally, antonyms can**
  - **define a binary opposition or be at opposite ends of a scale:**
    - *long / short*, *fast / slow*
  - **be reversives:**
    - *rise/fall*, *up/down*

Words and Meaning: Connotations

- **Words have affective meanings**
  - **positive connotations (*happy*)**
  - **negative connotations (*sad*)**
- **Connotations can be subtle:**
  - **positive connotation:** *copy*, *replica*, *reproduction*
  - **negative connotation:** *fake*, *knockoff*, *forgery*
- **Evaluation (sentiment):**
  - **positive evaluation (*great*, *love*)**
  - **negative evaluation (*terrible*, *hate*)**

Words and Meaning: Connotations

- Words seem to vary along **three affective dimensions**:
  - **valence**: the pleasantness of the stimulus
  - **arousal**: the intensity of emotion provoked by the stimulus
  - **dominance**: the degree of control exerted by the stimulus

	Word	Score		Word	Score
valence	love	1.000		toxic	0.008
	happy	1.000		nightmare	0.005
arousal	elated	0.960		mellow	0.069
	frenzy	0.965		napping	0.046
dominance	powerful	0.991		weak	0.045
	leadership	0.983		empty	0.081

Source: NRC VAD Lexicon (https://saifmohammad.com/WebPages/nrc-vad.html)

Words and Meaning: Summary

- Concepts or word senses
  - have a complex many-to-many association with words (homonymy, multiple senses)
- Have relations with each other
  - synonymy
  - antonymy
  - similarity
  - relatedness
  - connotation

WordNet

WordNet® is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept.

Synsets are interlinked by means of conceptual-semantic and lexical relations.

Link: https://wordnet.princeton.edu/

Challenge

- **We know word relationships exist**
- **How can we quantify them in a automated fashion?**
- **How do we represent them in numerical way?**
- **How can we use them in computational models and processes?**