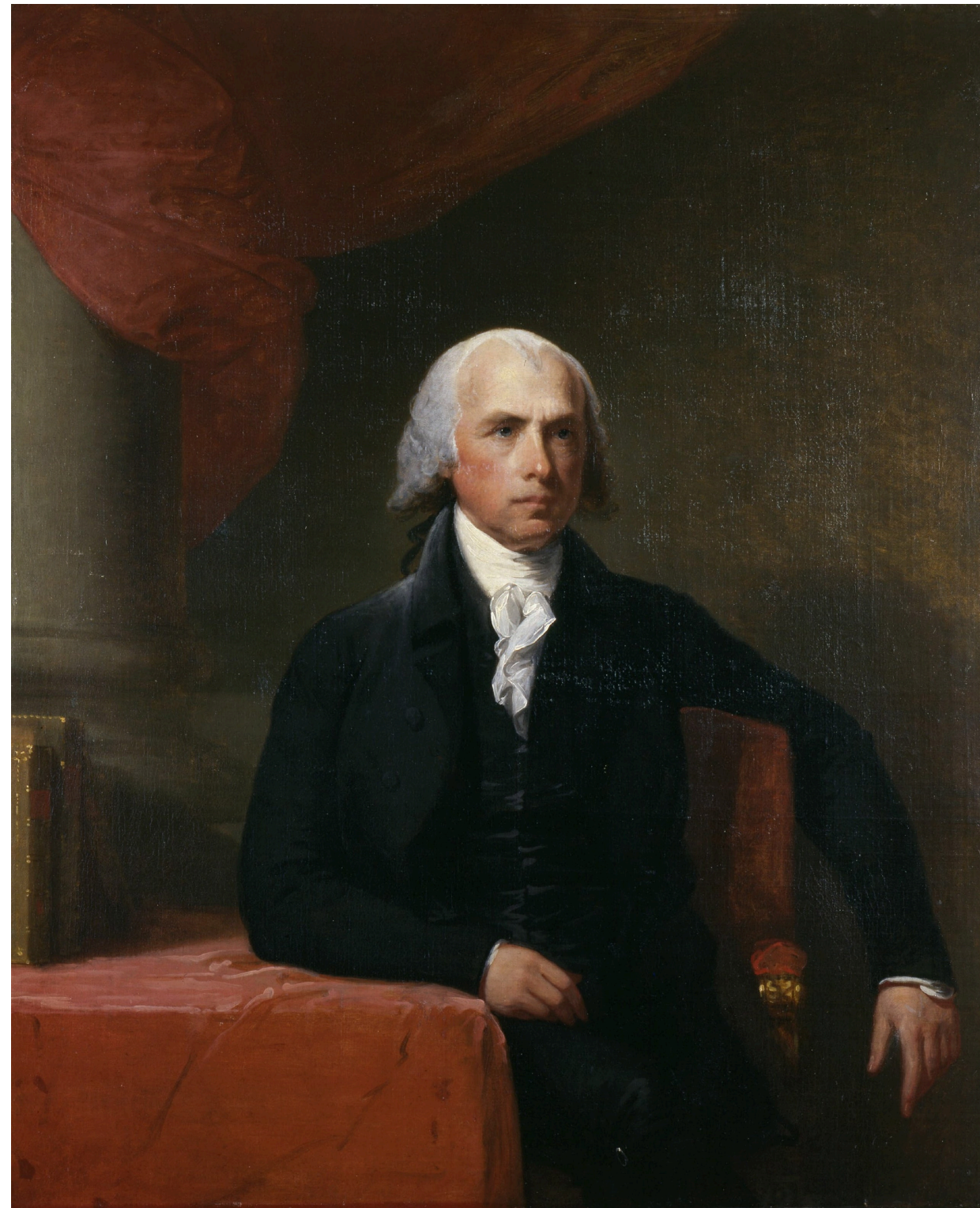


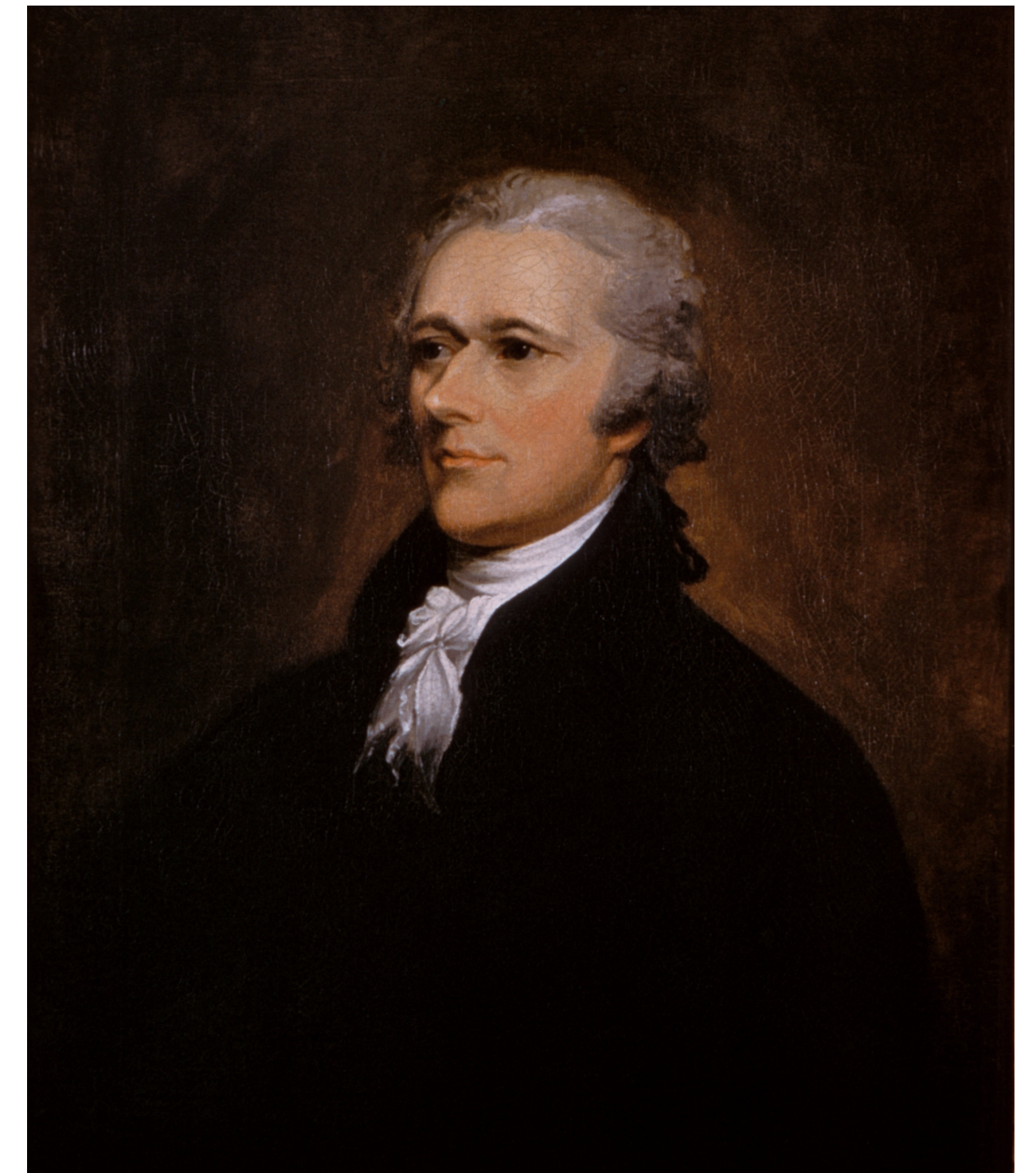
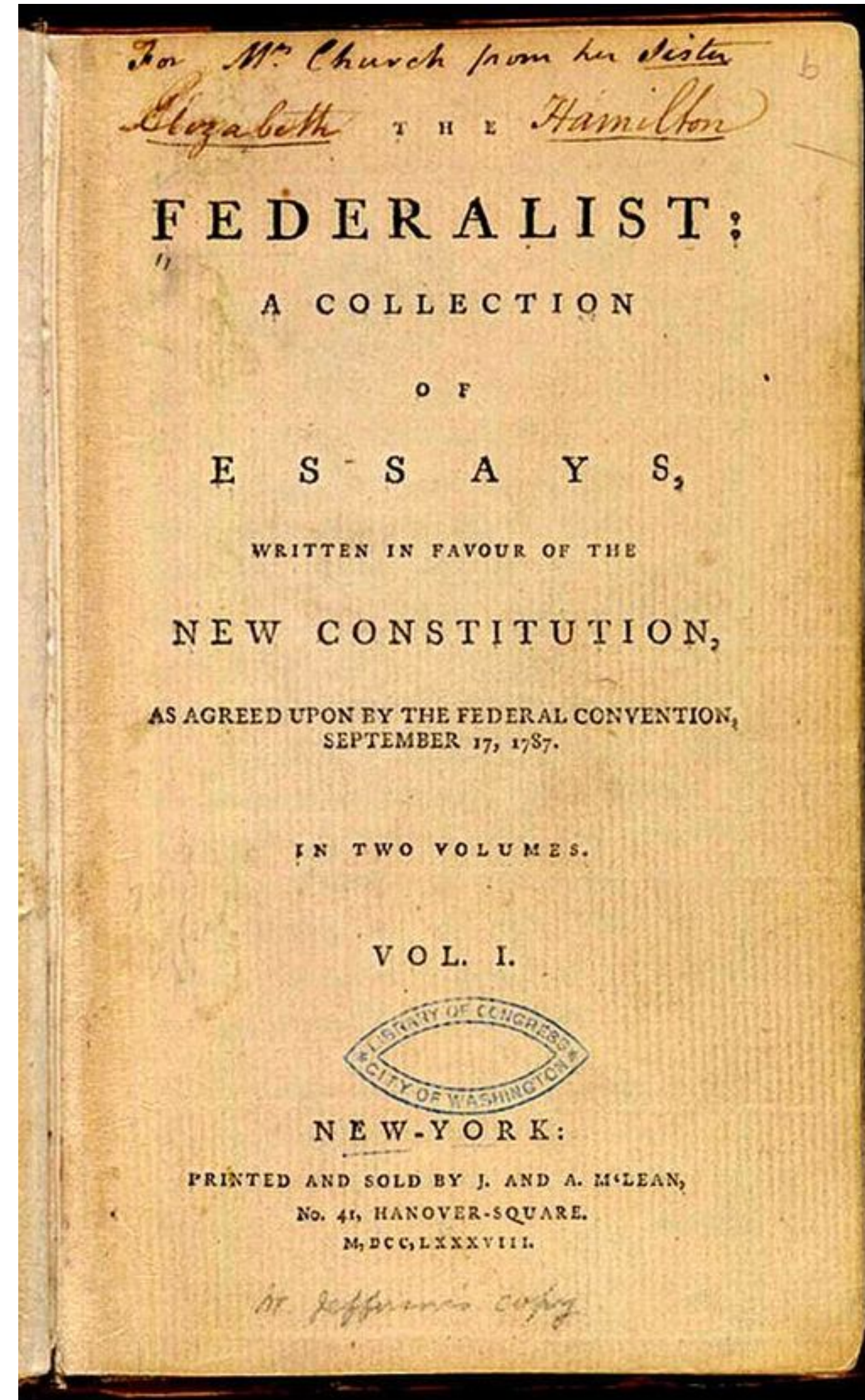
The task of text classification

*Alexander joins forces with James Madison
And John Jay to write a series of essays
Defending the new United States Constitution.
Entitled The Federalist Papers,
The plan was to write a total of 25 essays,
The work divided evenly among the three men.
In the end, they wrote 85 essays
In the span of six months,
John Jay got sick after writing five...
James Madison wrote 29...
Hamilton wrote the other 51!*

Lin-Manuel Miranda, "Non-Stop"



James Madison



Alexander Hamilton

From: "Fabian Starr" <Patrick_Freeman@pamietaniepeerelu.pl>
Subject: Hey! Software for the funny prices!

Get the great discounts on popular software today for PC
and Macintosh

<http://iiled.org/Cj4Lmx>

70-90% Discounts from retail price!!!

All software is instantly available to download - No Need
Wait!

Is this spam?

From: "Service Desk" <servicedesk@vassar.edu>
Subject: Important Update jgordon@vassar.edu
Date: 30 June 2023 at 15:12:38 EDT
To: jgordon@vassar.edu
Reply-To: servicedesk@vassar.edu

VASSAR END

Vassar College

To: All Employees
From: Help Desk
Subject: Important: Email update - Action required

WHAT: Recently we updated Vassar College Email servers to enhance end user experience and improve security.

WHO: This change pertains to all Vassar College email users, and are advised to update their account to comply with the new server requirements.

WHY: Non-compliance might process your account as active, and you may experience interruption of services or undue errors.

HOW: Kindly update your account [HERE](#)

We appreciate your support and cooperation during this update effort

What about this?

Are these film reviews positive or negative?

I wanted to like it, when it was first announced I was excited.

Good movie, but overhyped.

...leaves a quiet, aching heaviness in the chest.

I left the theater feeling nothing.

Sometimes it's good to hear what the actors are saying.

...an ancient, tragic, violent story with more than enough material for powerful performances.

Are these film reviews positive or negative?

I wanted to like it, when it was first announced I was excited.

 1/10

Good movie, but overhyped.

...leaves a quiet, aching heaviness in the chest.

I left the theater feeling nothing.

Sometimes it's good to hear what the actors are saying.

...an ancient, tragic, violent story with more than enough material for powerful performances.

Are these film reviews positive or negative?

I wanted to like it, when it was first announced I was excited.

 1/10

Good movie, but overhyped.

 8/10

...leaves a quiet, aching heaviness in the chest.

I left the theater feeling nothing.

Sometimes it's good to hear what the actors are saying.

...an ancient, tragic, violent story with more than enough material for powerful performances.

Are these film reviews positive or negative?

I wanted to like it, when it was first announced I was excited.

 1/10

Good movie, but overhyped.

 8/10

...leaves a quiet, aching heaviness in the chest.

 9/10

I left the theater feeling nothing.

Sometimes it's good to hear what the actors are saying.

...an ancient, tragic, violent story with more than enough material for powerful performances.

Are these film reviews positive or negative?

I wanted to like it, when it was first announced I was excited.

 1/10

Good movie, but overhyped.

 8/10

...leaves a quiet, aching heaviness in the chest.

 9/10

I left the theater feeling nothing.

 3/10

Sometimes it's good to hear what the actors are saying.

...an ancient, tragic, violent story with more than enough material for powerful performances.

Are these film reviews positive or negative?

I wanted to like it, when it was first announced I was excited.

 1/10

Good movie, but overhyped.

 8/10

...leaves a quiet, aching heaviness in the chest.

 9/10

I left the theater feeling nothing.

 3/10

Sometimes it's good to hear what the actors are saying.

 8/10

...an ancient, tragic, violent story with more than enough material for powerful performances.

Are these film reviews positive or negative?

I wanted to like it, when it was first announced I was excited.

 1/10

Good movie, but overhyped.

 8/10

...leaves a quiet, aching heaviness in the chest.

 9/10

I left the theater feeling nothing.

 3/10

Sometimes it's good to hear what the actors are saying.

 8/10

...an ancient, tragic, violent story with more than enough material for powerful performances.

 3/10

Are these film reviews positive or negative?

I wanted to like it, when it was first announced I was excited.

👎 1/10

Good movie, but overhyped.

👍 8/10

...leaves a quiet, aching heaviness in the chest.

👍 9/10

I left the theater feeling nothing.

👎 3/10

Sometimes it's good to hear what the actors are saying.

👍 8/10

...an ancient, tragic, violent story with more than enough material for powerful performances.

👎 3/10



What’s the subject of this medical article?

Antagonists and inhibitors

Blood supply

Chemistry

Drug therapy

Embryology

Epidemiology

...



Many problems take the form of text classification,
e.g.,

<i>Task</i>	<i>x</i>	<i>y</i>
<i>Spam identification</i>	An email	{spam, not spam}
<i>Sentiment analysis</i>	A review (e.g., from Yelp or Amazon)	{positive, negative, neutral, mixed}
<i>Genre classification</i>	A novel	{detective, romance, gothic, ...}
<i>Author identification</i>	Text	{Tolkien, Shakespeare, ...}

and many more...

Text classification problems take this form:

Input:

A document d (which can be any text)

A fixed set of classes $Y = \{y_1, y_2, \dots, y_j\}$

Output:

A predicted class $\hat{y} \in Y$

The circumflex (hat) notation is used to indicate an estimated or predicted value.

We can build a classifier by writing rules by hand,
but this is slow, expensive, and difficult to maintain.

Instead, like humans learn from experience, we
make computers learn from data – *machine learning*.

A *supervised machine-learning* text classification problem takes this form:

Input:

A fixed set of classes $Y = \{y_1, y_2, \dots, y_j\}$

A training set of m hand-labeled documents $(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})$

Output:

A learned classifier $\gamma : d \rightarrow \hat{y}$

Training

A *supervised machine-learning* text classification problem takes this form:

Input:

A fixed set of classes $Y = \{y_1, y_2, \dots, y_j\}$

A training set of m hand-labeled documents $(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})$

Output:

A learned classifier $\gamma : d \rightarrow \hat{y}$

Training

Input:

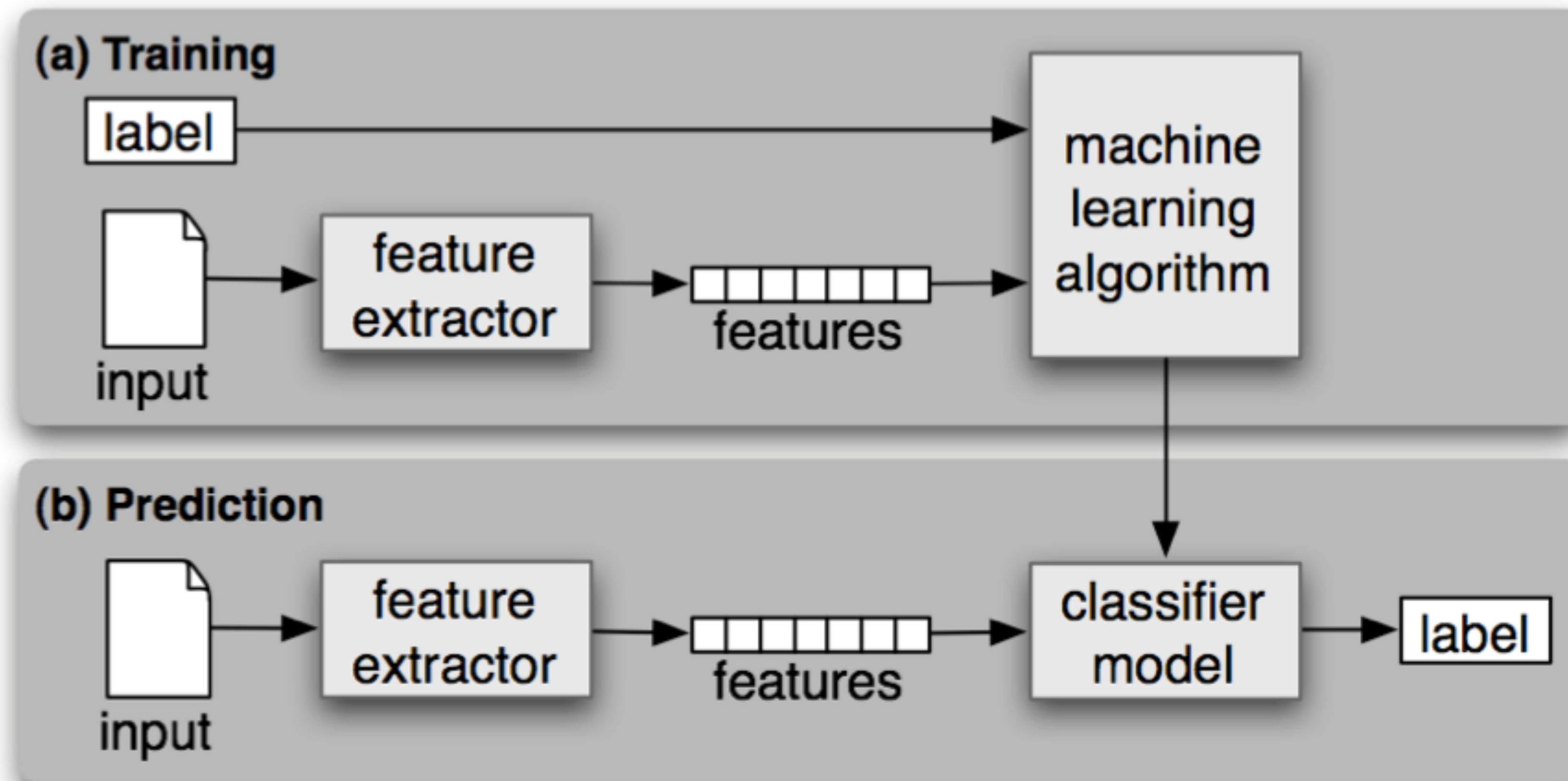
A document d

Output:

A class $\hat{y}$

Inference or test

Supervised machine learning



A classification decision must rely on some observable evidence, which we encode as *features*.

Typical features include:

Words (or n -grams) present in the text

Frequency of words

Capitalization

Presence of named entities

Syntactic relations

Semantic relations

The simplest and most common features are Boolean, e.g., is the word present or not?

However, we can also have integer features like the number of times a word occurs.

The features we select depend on the task.

Is a name masculine or feminine?

Last letter = ...

What part-of-speech is a word?

Is the word preceded by *the*? *to*?

Does the word end with *-ly*? *-ness*?

Is an email spam?

Does it contain *generic Viagra*?

Is the subject in all capital letters?

Feature engineering is the problem of deciding what features are relevant.

Approaches:

Hand-crafted

Use expert knowledge to determine a small set of features that are likely to be relevant.

Kitchen sink

Give lots of features to the machine-learning algorithm and see what features are given greater weight and which are ignored

E.g., use each word in the document as a feature:

has-*cash*: True

has-*the*: True

has-*linguistics*: False

...

Weighting the evidence

A classification decision involves reconciling multiple features with different levels of predictive power.

Different types of classifiers use different algorithms to:

- Determine the weights of individual features to maximize correct predictions for the training data and

- Compute the likelihood of a label for an input, using the feature weights.

There are many kinds of classifiers:

Naïve Bayes

Logistic regression

Neural networks

k -nearest neighbors

LLMs

Fine-tuned as classifiers

Prompted to give a classification

There are many kinds of classifiers:

Naïve Bayes

Logistic regression

The focus for today!

Neural networks

k -nearest neighbors

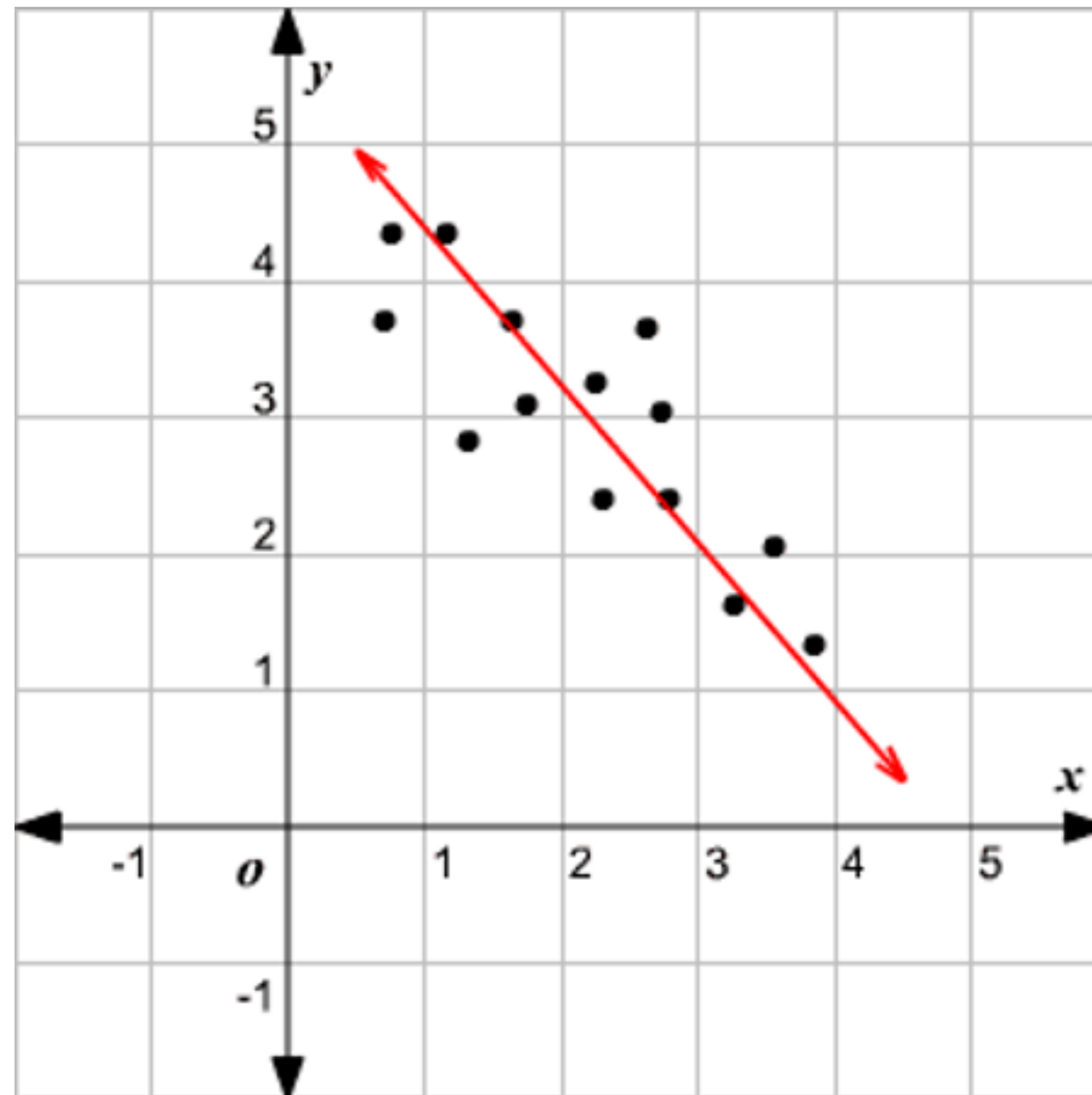
LLMs

Fine-tuned as classifiers

Prompted to give a classification

Logistic regression classification

Background: Regression



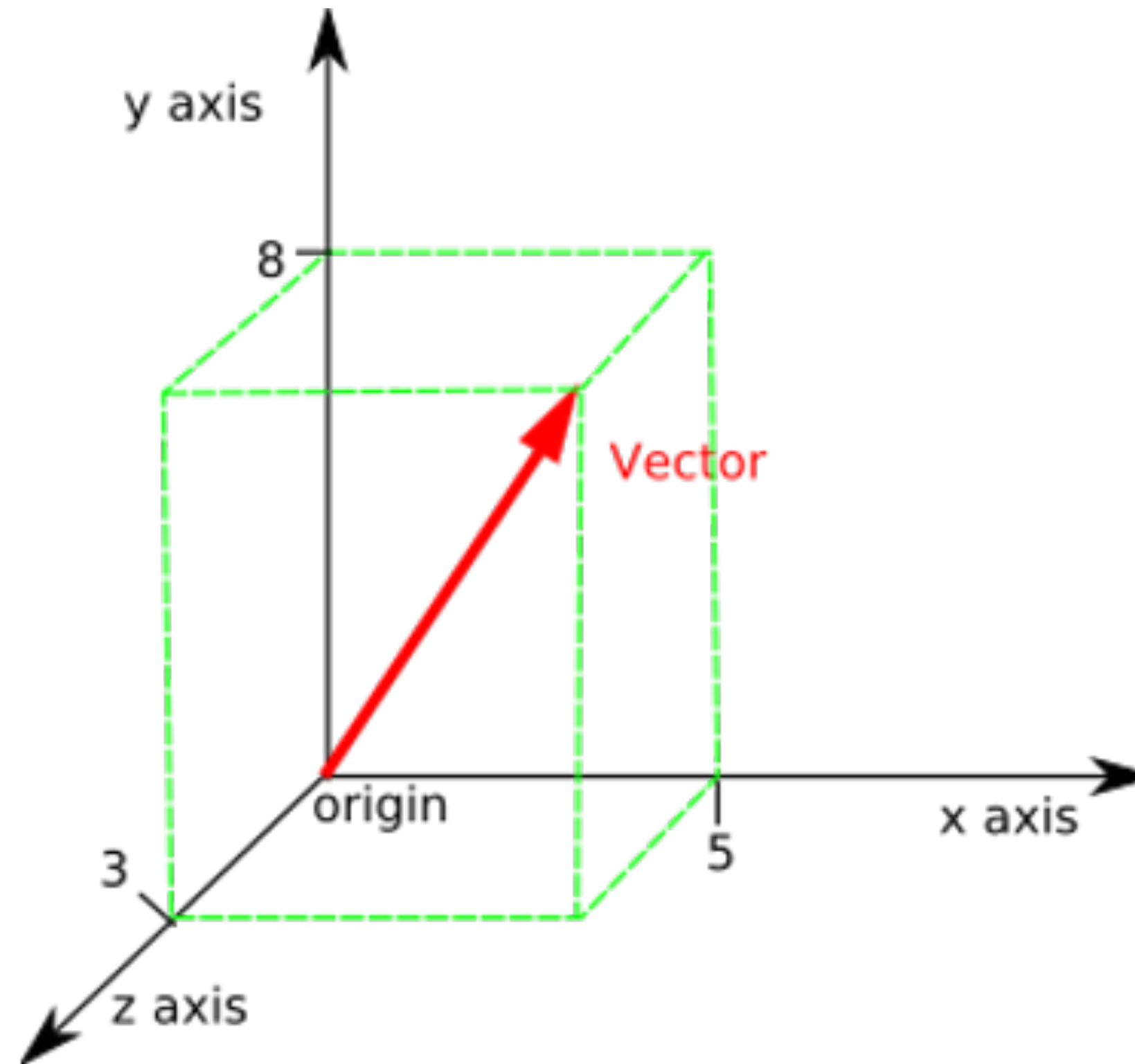
$$y = mx + b$$

Logistic regression is important because

it's a simple method that serves as a *baseline* supervised machine learning tool for classification, and

it's also the foundation of neural networks!

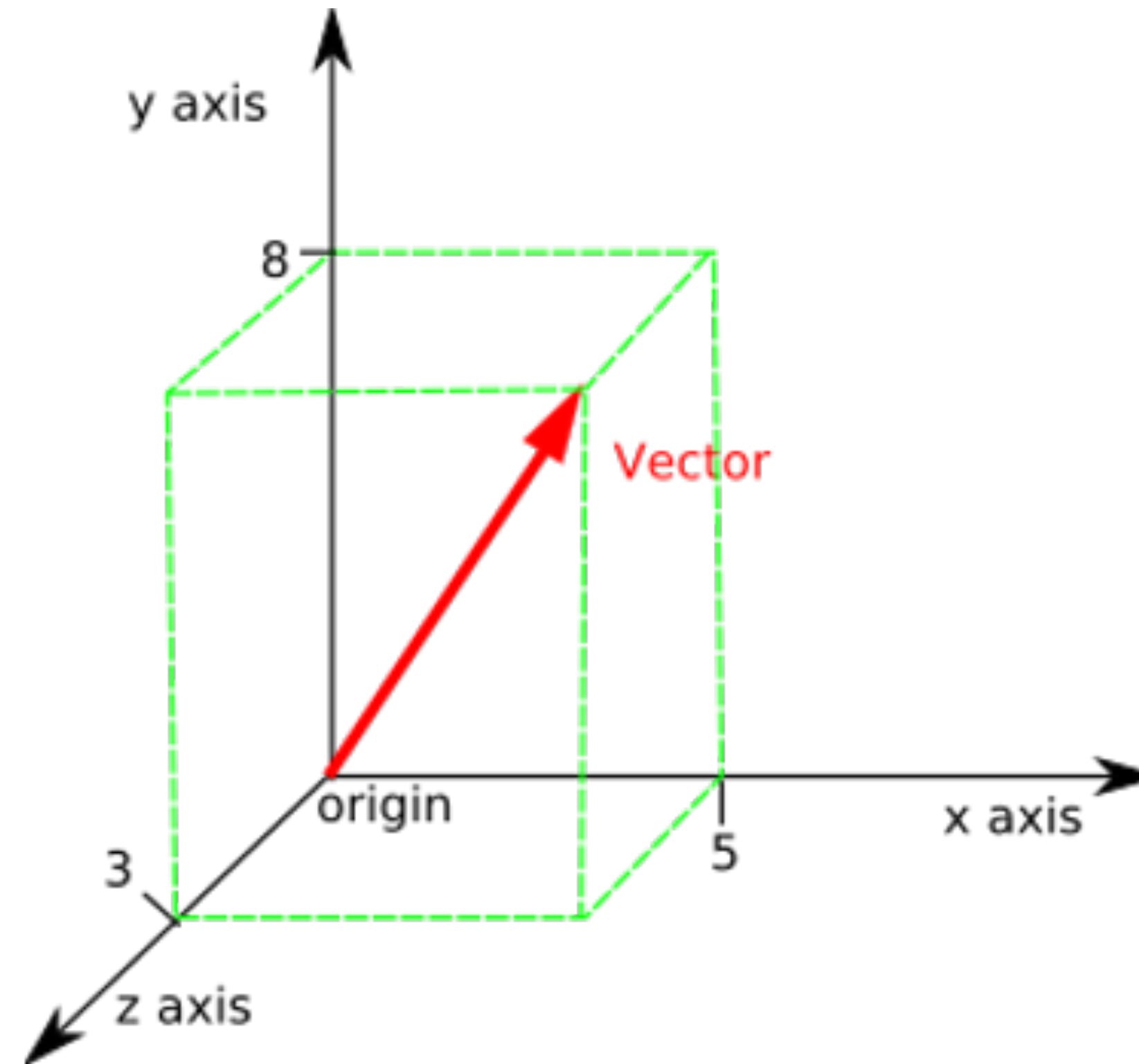
Vectors



$$\mathbf{a} = [5, 8, 3]$$

$$\vec{a} = [5, 8, 3]$$

Vectors



$$\mathbf{a} = [5, 8, 3]$$

$$\vec{a} = [5, 8, 3]$$

For logistic regression, each input observation is represented by a *feature vector* $\mathbf{x} = [x_1, x_2, \dots, x_n]$.

The output of the classifier can be one of two predicted classes, 0 or 1.

To be able to correctly classify inputs, we learn how predictive each feature x_i is of either class by finding a corresponding weight w_i , e.g.,

x_1 = “review contains *awesome*” $w_1 = +10$

x_2 = “review contains *abysmal*” $w_2 = -10$

x_3 = “review contains *mediocre*” $w_3 = -2$

To use the weights to classify an instance, we multiply each feature x_i by its corresponding weight w_i and add them up:

$$z = \left(\sum_{i=1}^n w_i x_i \right) + b$$

The last term, b , is the *bias* (or *intercept*).

To use the weights to classify an instance, we multiply each feature x_i by its corresponding weight w_i and add them up:

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

Dot product!

The last term, b , is the *bias* (or *intercept*).

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

If z is high, predict x belongs to the positive category (1).

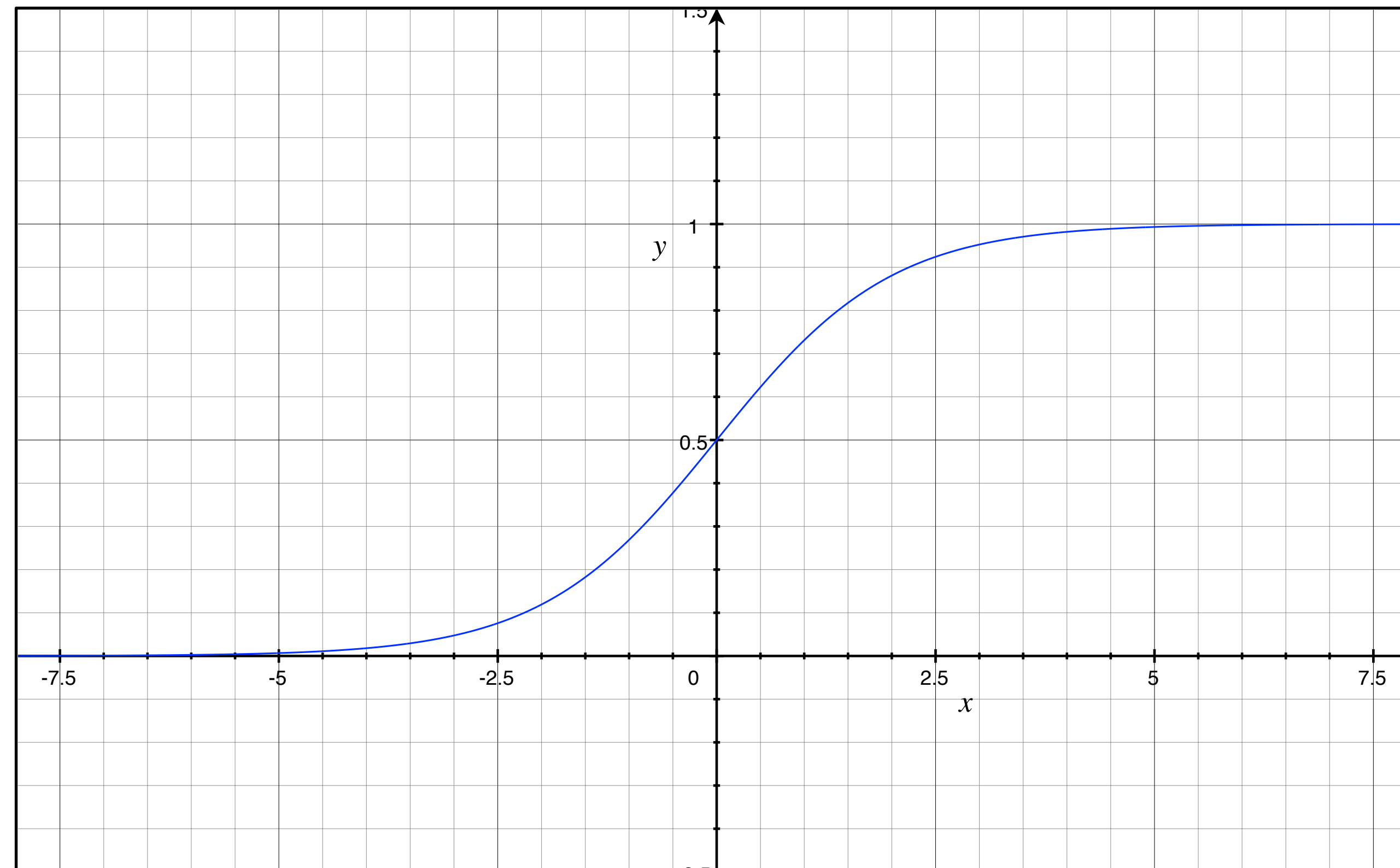
Otherwise, predict it belongs to the negative category (0).

The problem is we don't have a fixed range of values for the sum z , so it's not clear what counts as being "high".

Solution: Make it a probability, between 0 and 1.

We can turn z into a probability by passing it through the *sigmoid* function σ :

$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$



Making probabilities with sigmoids:

$$P(y = 1) = \sigma(\mathbf{w} \cdot \mathbf{x} + b)$$

Making probabilities with sigmoids:

$$\begin{aligned} P(y = 1) &= \sigma(\mathbf{w} \cdot \mathbf{x} + b) \\ &= \frac{1}{1 + \exp(-(\mathbf{w} \cdot \mathbf{x} + b))} \end{aligned}$$

Making probabilities with sigmoids:

$$\begin{aligned} P(y = 1) &= \sigma(\mathbf{w} \cdot \mathbf{x} + b) \\ &= \frac{1}{1 + \exp(-(\mathbf{w} \cdot \mathbf{x} + b))} \end{aligned}$$

$$P(y = 0) = 1 - \sigma(\mathbf{w} \cdot \mathbf{x} + b)$$

So, for a particular input x , we can now compute $P(y = 1 \mid x)$ and $P(y = 0 \mid x)$.

To turn these probabilities into a classifier, we just use the *decision boundary* 0.5:

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

Logistic regression example: Sentiment classification

It's hokey . There are virtually no surprises , and the writing is second-rate .

So why was it so enjoyable ?

For one thing , the cast is great .

Another nice touch is the music . I was overcome with the urge to get off the couch and start dancing . It sucked me in , and it'll do the same to you .

Is this review positive ($y = 1$) or negative ($y = 0$)?

It's hokey . There are virtually no surprises , and the writing is second-rate .

*So why was it so **enjoyable** ?*

*For one thing , the cast is **great** .*

*Another **nice** touch is the music . I was overcome with the urge to get off the couch and start dancing . It sucked me in , and it'll do the same to you .*

Var	Definition	
x_1	count(positive lexicon words $\in$ doc)	3

*It's **hokey**. There are virtually no surprises , and the writing is **second-rate**.*

So why was it so enjoyable ?

For one thing , the cast is great .

Another nice touch is the music . I was overcome with the urge to get off the couch and start dancing . It sucked me in , and it'll do the same to you .

Var	Definition	
x_1	count(positive lexicon words $\in$ doc)	3
x_2	count(negative lexicon words $\in$ doc)	2

*It's hokey . There are virtually **no** surprises , and the writing is second-rate .*

So why was it so enjoyable ?

For one thing , the cast is great .

Another nice touch is the music . I was overcome with the urge to get off the couch and start dancing . It sucked me in , and it'll do the same to you .

Var	Definition	
x_1	count(positive lexicon words $\in$ doc)	3
x_2	count(negative lexicon words $\in$ doc)	2
x_3	$\begin{cases} 1 & \text{if "no" } \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$	1

It's hokey . There are virtually no surprises , and the writing is second-rate .

So why was it so enjoyable ?

For one thing , the cast is great .

*Another nice touch is the music . **I** was overcome with the urge to get off the couch and start dancing . It sucked **me** in , and it'll do the same to **you** .*

Var	Definition	
x_1	count(positive lexicon words $\in$ doc)	3
x_2	count(negative lexicon words $\in$ 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

It's hokey . There are virtually no surprises , and the writing is second-rate .

So why was it so enjoyable ?

For one thing , the cast is great .

Another nice touch is the music . I was overcome with the urge to get off the couch and start dancing . It sucked me in , and it'll do the same to you .

Var	Definition	
x_1	count(positive lexicon words $\in$ doc)	3
x_2	count(negative lexicon words $\in$ 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

It's hokey . There are virtually no surprises , and the writing is second-rate .

So why was it so enjoyable ?

For one thing , the cast is great .

Another nice touch is the music . I was overcome with the urge to get off the couch and start dancing . It sucked me in , and it'll do the same to you .

Var	Definition	
x_1	count(positive lexicon words $\in$ doc)	3
x_2	count(negative lexicon words $\in$ 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	ln(word count of doc)	$\ln(66) = 4.19$

*Suppose we learned
these weights and bias*

$$\mathbf{w} = [2.5, -5.0, -1.2, 0.5, 2.0, 0.7] \quad b = 0.1$$

Var	Definition	
x_1	count(positive lexicon words $\in$ doc)	3
x_2	count(negative lexicon words $\in$ 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	$\ln(\text{word count of doc})$	$\ln(66) = 4.19$

$$\mathbf{w} = [2.5, -5.0, -1.2, 0.5, 2.0, 0.7] \quad b = 0.1$$

$$\mathbf{x} = [3, 2, 1, 3, 0, 4.19]$$

Var	Definition	
x_1	count(positive lexicon words $\in$ doc)	3
x_2	count(negative lexicon words $\in$ 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	$\ln(\text{word count of doc})$	$\ln(66) = 4.19$

$$\mathbf{w} = [2.5, -5.0, -1.2, 0.5, 2.0, 0.7] \quad b = 0.1$$

$$\mathbf{x} = [3, 2, 1, 3, 0, 4.19]$$

$$P(y = 1) = \sigma(\mathbf{w} \cdot \mathbf{x} + b)$$

$$\mathbf{w} = [2.5, -5.0, -1.2, 0.5, 2.0, 0.7] \quad b = 0.1$$

$$\mathbf{x} = [3, 2, 1, 3, 0, 4.19]$$

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

$$\mathbf{w} = [2.5, -5.0, -1.2, 0.5, 2.0, 0.7] \quad b = 0.1$$

$$\mathbf{x} = [3, 2, 1, 3, 0, 4.19]$$

$$\begin{aligned} P(y = 1) &= \sigma(\mathbf{w} \cdot \mathbf{x} + b) \\ &= \sigma([2.5, -5.0, -1.2, 0.5, 2.0, 0.7] \cdot [3, 2, 1, 3, 0, 4.19] + 0.1) \\ &= \sigma(0.833) \end{aligned}$$

$$\mathbf{w} = [2.5, -5.0, -1.2, 0.5, 2.0, 0.7] \quad b = 0.1$$

$$\mathbf{x} = [3, 2, 1, 3, 0, 4.19]$$

$$\begin{aligned} P(y = 1) &= \sigma(\mathbf{w} \cdot \mathbf{x} + b) \\ &= \sigma([2.5, -5.0, -1.2, 0.5, 2.0, 0.7] \cdot [3, 2, 1, 3, 0, 4.19] + 0.1) \\ &= \sigma(0.833) \\ &= 0.70 \end{aligned}$$

$$\mathbf{w} = [2.5, -5.0, -1.2, 0.5, 2.0, 0.7] \quad b = 0.1$$

$$\mathbf{x} = [3, 2, 1, 3, 0, 4.19]$$

$$\begin{aligned} P(y = 1) &= \sigma(\mathbf{w} \cdot \mathbf{x} + b) \\ &= \sigma([2.5, -5.0, -1.2, 0.5, 2.0, 0.7] \cdot [3, 2, 1, 3, 0, 4.19] + 0.1) \\ &= \sigma(0.833) \\ &= 0.70 \end{aligned}$$

$$P(y = 0) = 1 - \sigma(\mathbf{w} \cdot \mathbf{x} + b) = 0.30$$

We can build features for logistic regression for any classification task, e.g., the sentence segmentation we did on Assignment 1:

This ends in a period.

The house at 465 Main St. is new.

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

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

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

Summary

Given:

a set of classes, e.g., $\{+ \text{ sentiment}, -\text{sentiment}\}$

a vector $\mathbf{x}$ of features $[x_1, x_2, \dots, x_n]$

$x_1 = \text{count}(\text{awesome})$

$x_2 = \log(\text{number of words in reviews})$

$x_3 = \dots$

a vector $\mathbf{w}$ of weights $[w_1, w_2, \dots, w_n]$

w_i for each feature f_i

$$\begin{aligned} P(y = 1) &= \sigma(w \cdot x + b) \\ &= \frac{1}{1 + e^{-(w \cdot x + b)}} \end{aligned}$$

Learning: Cross-entropy loss

In supervised classification, for each example in the training data, we know the correct label y (0 or 1).

The classifier produces an estimated label, $\hat{y}$.

We want to set the weights $\mathbf{w}$ and bias b to minimize the distance between our estimate $\hat{y}$ and the true y for each example.

In supervised classification, for each example in the training data, we know the correct label y (0 or 1).

The classifier produces an estimated label, $\hat{y}$.

We want to set the weights $\mathbf{w}$ and bias b to minimize the distance between our estimate $\hat{y}$ and the true y for each example.



We estimate this distance using a “loss function” or “cost function”

In supervised classification, for each example in the training data, we know the correct label y (0 or 1).

The classifier produces an estimated label, $\hat{y}$.

We want to set the weights $\mathbf{w}$ and bias b to minimize the distance between our estimate $\hat{y}$ and the true y for each example.



We need an algorithm to iteratively update these to minimize the loss.

We want to choose the parameters $\mathbf{w}$ and b that maximize $P(y \mid x)$,

the (log) probability

of the true y labels in the training data

given the observations x .

This is called *conditional maximum likelihood estimation*.

Since there are only two discrete outcomes (0 or 1), we can express the probability $P(y | x)$ from our classifier generically as

$$P(y | x) = \hat{y}^y (1 - \hat{y})^{1-y}$$

noting that

if $y = 1$, this simplifies to $\hat{y}$

if $y = 0$, this simplifies to $1 - \hat{y}$

$$P(y \mid x) = \hat{y}^y (1 - \hat{y})^{1-y}$$

$$\log P(y \mid x) = \log [\hat{y}^y (1 - \hat{y})^{1-y}]$$

$$= y \log \hat{y} + (1 - y) \log(1 - \hat{y})$$

This is a probability to maximize

$$P(y \mid x) = \hat{y}^y (1 - \hat{y})^{1-y}$$

$$\log P(y \mid x) = \log [\hat{y}^y (1 - \hat{y})^{1-y}]$$

$$= y \log \hat{y} + (1 - y) \log(1 - \hat{y})$$

This is a probability to maximize

$$P(y \mid x) = \hat{y}^y (1 - \hat{y})^{1-y}$$

$$\log P(y \mid x) = \log [\hat{y}^y (1 - \hat{y})^{1-y}]$$

$$= y \log \hat{y} + (1 - y) \log(1 - \hat{y})$$

This a measure of loss, to minimize

$$-\log P(y \mid x) = - [y \log \hat{y} + (1 - y) \log(1 - \hat{y})]$$

$L(\hat{y}, y)$ is the loss function, expressing how far the classifier output $\hat{y}$ is from the correct output y .

We just derived the *cross-entropy loss*:

$$\begin{aligned} L_{\text{CE}}(\hat{y}, y) &= -\log P(y \mid x) \\ &= -[y \log \hat{y} + (1 - y) \log(1 - \hat{y})] \\ &= -[y \log \sigma(\mathbf{w} \cdot \mathbf{x} + b) + (1 - y) \log(1 - \sigma(\mathbf{w} \cdot \mathbf{x} + b))] \end{aligned}$$

We want the loss to be

smaller if the model estimate is close to correct

bigger if the model is confused

*It's hokey . There are virtually no surprises , and the writing is second-rate .
So why was it so enjoyable ?*

For one thing , the cast is great .

*Another nice touch is the music . I was overcome with the urge to get off the
couch and start dancing . It sucked me in , and it'll do the same to you .*

Suppose the true label is $y = 1$ (it's a positive review).

How well is our classifier doing?

$$\mathbf{w} = [2.5, -5.0, -1.2, 0.5, 2.0, 0.7] \quad b = 0.1$$

$$\mathbf{x} = [3, 2, 1, 3, 0, 4.19]$$

$$P(y = 1) = \sigma(\mathbf{w} \cdot \mathbf{x} + b) = 0.70$$

$$\mathbf{w} = [2.5, -5.0, -1.2, 0.5, 2.0, 0.7] \quad b = 0.1$$

$$\mathbf{x} = [3, 2, 1, 3, 0, 4.19]$$

$$P(y = 1) = \sigma(\mathbf{w} \cdot \mathbf{x} + b) = 0.70$$

Pretty well! What's the loss?

$$\mathbf{w} = [2.5, -5.0, -1.2, 0.5, 2.0, 0.7] \quad b = 0.1$$

$$\mathbf{x} = [3, 2, 1, 3, 0, 4.19]$$

$$P(y = 1) = \sigma(\mathbf{w} \cdot \mathbf{x} + b) = 0.70$$

Pretty well! What's the loss?

$$\begin{aligned} L_{\text{CE}}(\hat{y}, y) &= -[y \log \sigma(\mathbf{w} \cdot \mathbf{x} + b) + (1 - y) \log (1 - \sigma(\mathbf{w} \cdot \mathbf{x} + b))] \\ &= -[\log \sigma(\mathbf{w} \cdot \mathbf{x} + b)] \\ &= -\log(.70) \\ &= .36 \end{aligned}$$

*It's hokey . There are virtually no surprises , and the writing is second-rate .
So why was it so enjoyable ?*

For one thing , the cast is great .

*Another nice touch is the music . I was overcome with the urge to get off the
couch and start dancing . It sucked me in , and it'll do the same to you .*

What if the true label were $y = 0$ (it's a negative review)?

$$P(y = 1) = \sigma(\mathbf{w} \cdot \mathbf{x} + b) = 0.70$$

$$P(y = 0) = 1 - P(y = 1) = 0.30$$

$$P(y = 1) = \sigma(\mathbf{w} \cdot \mathbf{x} + b) = 0.70$$

$$P(y = 0) = 1 - P(y = 1) = 0.30$$

$$\begin{aligned} L_{\text{CE}}(\hat{y}, y) &= -[y \log \sigma(\mathbf{w} \cdot \mathbf{x} + b) + (1 - y) \log (1 - \sigma(\mathbf{w} \cdot \mathbf{x} + b))] \\ &= -[\log (1 - \sigma(\mathbf{w} \cdot \mathbf{x} + b))] \\ &= -\log (.30) \\ &= 1.2 \end{aligned}$$

The loss when the model was right (true $y = 1$) is lower than the loss when the model was wrong (true $y = 0$), which is exactly what we want for a measure we're going to minimize!

Next time – gradient descent!

