

# CS 1671 / CS 2071 / ISSP 2071

## Human Language Technologies

Session 8: Logistic regression part 1

---

Michael Miller Yoder

February 9, 2026



University of  
Pittsburgh

School of Computing and Information

# Quiz

- Go to Quizzes > Quiz 02-09 on Canvas
- You have until 1:10pm to complete it
- Allowed resources
  - Textbook
  - Your notes (on a computer or physical)
  - Course slides and website
- Resources not allowed
  - Generative AI
  - Internet searches

What do you call a bad dream about machine learning?

*A logistic nightmare*

# Course logistics: project

- Consider which projects you'd like to work on from the [list of project options](#)
- Project Match Day is **next class, Wed Feb 11**. You will form groups of ~5 students around projects from the project list
  - We'll "match" in a few rounds, where groups that are too large or too small will join other groups
  - Come with multiple project preferences
- If you're really excited about an idea, talk to others who might be interested

# Course logistics: quiz

- Next in-class quiz is next class, **Wed Feb 11**
  - Session 8 (today): J+M 4-4.3
- Lowest quiz score in the course will be dropped
- If you won't be in class, let me know and I can accommodate

## Course logistics: homework

- Homework 1 is **due Thu Feb 12 at 11:59pm**
- Homework 1 covers text processing and regular expressions in Python



Pitt Python Linguistics Group

- Computational linguistics group on campus
- Practice NLP skills related to this class with fun tutorials and guest speakers (with food, too! 🍕)
- Next meeting is **Wed Feb 11, 6-7:15pm at CL 2818** (linguistics department conference room)
  - Tour of **computational linguistics certificate** which is going live this semester! 🎉
  - Also text analysis of NLP conference proceedings (how much are dominated by LLMs? What other areas are prevalent?)
- Contact Na-Rae Han, [naraehan@pitt.edu](mailto:naraehan@pitt.edu) to get on their mailing list

# Lecture overview: Logistic regression part 1

- Coding activity: build your own n-gram LM
- Text classification
- Input to classification: features from text
- Logistic regression
- Binary classification with logistic regression
- Coding activity: custom features for logistic regression

# Coding activity: build your own n-gram LM

---

# N-gram document representations on JupyterHub

1. Go to this [nbgitpuller link](#) (also available on course website)
2. Start a server with **TEACH – 6 CPUs, 48 GB**
3. Load custom environment at `/ix1/cs1671-2026s/class_env`
  1. If you have multiple accounts on the CRCD, put in `cs1671-2026s` for Account
4. This should pull the `cs1671_spring2026_jupyterhub` folder into your JupyterLab
5. Open `session7_ngram_lm.ipynb`

# Text classification

---

# Text classification

*"My dear Mr. Bennet," said his lady to him one day, "have you heard that Netherfield Park is let at last?"*

ROMANCE

*Pride and Prejudice*

DIALOG



# Is this spam?

**Subject: Important notice!**

**From:** Stanford University <newsforum@stanford.edu>

**Date:** October 28, 2011 12:34:16 PM PDT

**To:** undisclosed-recipients:;

---

Greats News!

You can now access the latest news by using the link below to login to Stanford University News Forum.

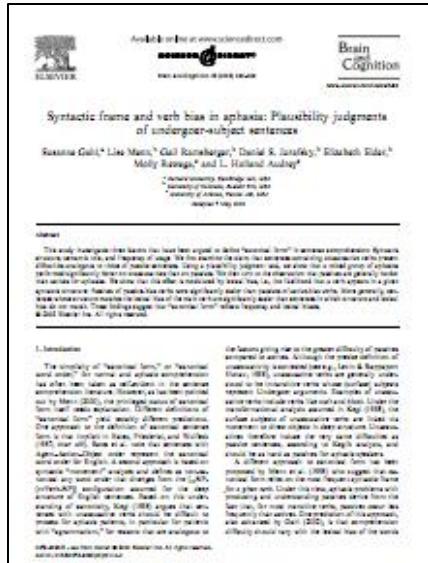
<http://www.123contactform.com/contact-form-StanfordNew1-236335.html>

Click on the above link to login for more information about this new exciting forum. You can also copy the above link to your browser bar and login for more information about the new services.

© Stanford University. All Rights Reserved.

# What's the subject of this medical article?

## MEDLINE Article



# MeSH Subject Category Hierarchy

Antagonists and Inhibitors

Blood Supply

Chemistry

Drug Therapy

Embryology

Epidemiology

...

?

# Text Classification

We have a set of documents that we want to *classify* into a small set *classes*.

## Applications:

- **Topic classification:** you have a set of news articles that you want to classify as finance, politics, or sports.
- **Sentiment detection:** you have a set of movie reviews that you want to classify as good, bad, or neutral.
- **Language Identification:** you have a set of documents that you want to classify as English, Mandarin, Arabic, or Hindi.
- **Reading level:** you have a set of articles that you want to classify as kindergarten, 1st grade, ...12th grade.
- **Author identification:** you have a set of fictional works that you want to classify as Shakespeare, James Joyce, ...
- **Genre identification:** you have a set of documents that you want to classify as report, editorial, advertisement, blog, ...

## Example: Sentiment Detection

	<b>Cat</b>	<b>Documents</b>
Training	-	just plain boring
	-	entirely predictable and lacks energy
	-	no surprises and very few laughs
	+	very powerful
	+	the most fun film of the summer
Test	?	predictable with no fun

# Input to classification tasks: features

---

# Term-document matrix

- Each cell is the count of term  $t$  in a document  $d$  ( $tf_{t,d}$ ).
- Each document is a **count vector** in  $\mathbb{N}^V$ , a column below.



	As You Like It	Twelfth Night	Julius Caesar	Henry V
battle	1	1	8	15
soldier	2	2	12	36
fool	37	58	1	5
clown	6	117	0	0

# Movie Ratings

- A training set of movie reviews (with star ratings 1 - 5)
- A set of features for each message (considered as a bag of words)
  - For each word: Number of occurrences
  - Whether phrases such as *Excellent, sucks, blockbuster, biggest, Star Wars, Disney, Adam Sandler*, ...are in the review

# Spam Detection

- A training set of email messages (marked *Spam* or *Not-Spam*)
- A set of features for each message
  - For each word: Number of occurrences
  - Whether phrases such as “Nigerian Prince”, “email quota full”, “won ONE HUNDRED MILLION DOLLARS” are in the message
  - Whether it is from someone you know
  - Whether it is a reply to your message
  - Whether it is from your domain (e.g., cmu.edu)

# Logistic regression

---

# What Goes into a (Discriminative) ML Classifier?

1. A feature representation
2. A classification function
3. An objective function
4. An algorithm for optimizing the objective function

# What Goes into Logistic Regression?

---

GENERAL	IN LOGISTIC REGRESSION
feature representation	represent each observation $\mathbf{x}^{(i)}$ as a vector of features $[x_1, x_2, \dots, x_n]$
classification function	sigmoid function (logistic function)
objective function	cross-entropy loss
optimization function	(stochastic) gradient descent

---

# The Two Phases of Logistic Regression

- train** learn  $w$  (a vector of weights, one for each feature) and  $b$  (a bias) using **stochastic gradient descent** and **cross-entropy loss**.
- test** given a test example  $x$ , we compute  $p(y|x)$  using the learned weights  $w$  and  $b$  and return the label ( $y = 1$  or  $y = 0$ ) that has higher probability.

# Binary classification with logistic regression

---

# Text classification with logistic regression

Given a series of input/output pairs:

$$(x^i, y^i)$$

For each observation  $x^i$

- We represent  $x^i$  by a feature vector  $[x_1, x_2, \dots, x_n]$
- We compute an output: a predicted class  $y^i \in \{0, 1\}$ 
  - Get to the predicted class by estimating  $p(y|x)$ , i.e.  $p(y=1|x)$  and  $p(y=0|x)$

For sentiment analysis (classification):

- $y^i = 1$  is positive sentiment,  $y^i = 0$  is negative sentiment

## Reminder: the Dot Product

We will see the dot product a lot. It is the **sum** of the element-wise **product** of two vectors of the same dimensionality.

$$\begin{bmatrix} 2 & 7 & 1 \end{bmatrix} \cdot \begin{bmatrix} 8 \\ 2 \\ 8 \end{bmatrix} = 2 \cdot 8 + 7 \cdot 2 + 1 \cdot 8 = 38 \quad (3)$$

Moving on...

# Features in Logistic Regression

For feature  $x_i$ , weight  $w_i$  tells us how important  $x_i$  is

- $x_i = \text{"review contains awesome": } w_i = +10$
- $x_j = \text{"review contains abysmal": } w_j = -10$
- $x_k = \text{"review contains mediocre": } w_k = -2$

# Logistic Regression for One Observation $x$

- input** observation feature vector  $x = [x_1, x_2, \dots, x_n]$
- weights** one per feature  $W = [w_1, w_2, \dots, w_n]$  plus  $w_0$ , which is the **bias**  $b$
- output** a predicted class  $\hat{y} \in \{0, 1\}$

# How to Do Classification

For each feature  $x_i$ , weight  $w_i$  tells us the importance of  $x_i$  (and we also have the bias  $b$  that shifts where the function crosses the  $x$ -axis)

We'll sum up all the weighted features and the bias

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

# A Most Important Formula

We compute

$$Z = \mathbf{W} \cdot \mathbf{X} + b$$

If  $z$  is high, we say  $y = 1$ ; if low, then  $y = 0$ .

**orchids** A classifiers for cymbidiums should return  $y = 1$  when the input is a cymbidium and  $y = 0$  otherwise.

**sentiment** A classifier for positive sentiment should return  $y = 1$  when the input has positive sentiment (when the emotions of the writer towards the topic are positive) and  $y = 0$  otherwise.

**Remember this formula.**

## But We Want a Probabilistic Classifier

What does “sum is high” even mean?

Can’t our classifier be like Naive Bayes and give us a probability?

What we really want:

- $p(y = 1|x; \theta)$
- $p(y = 0|x; \theta)$

Where  $x$  is a vector of features and  $\theta = (w, b)$  (the weights and the bias).

## The Problem: $z$ isn't a Probability!

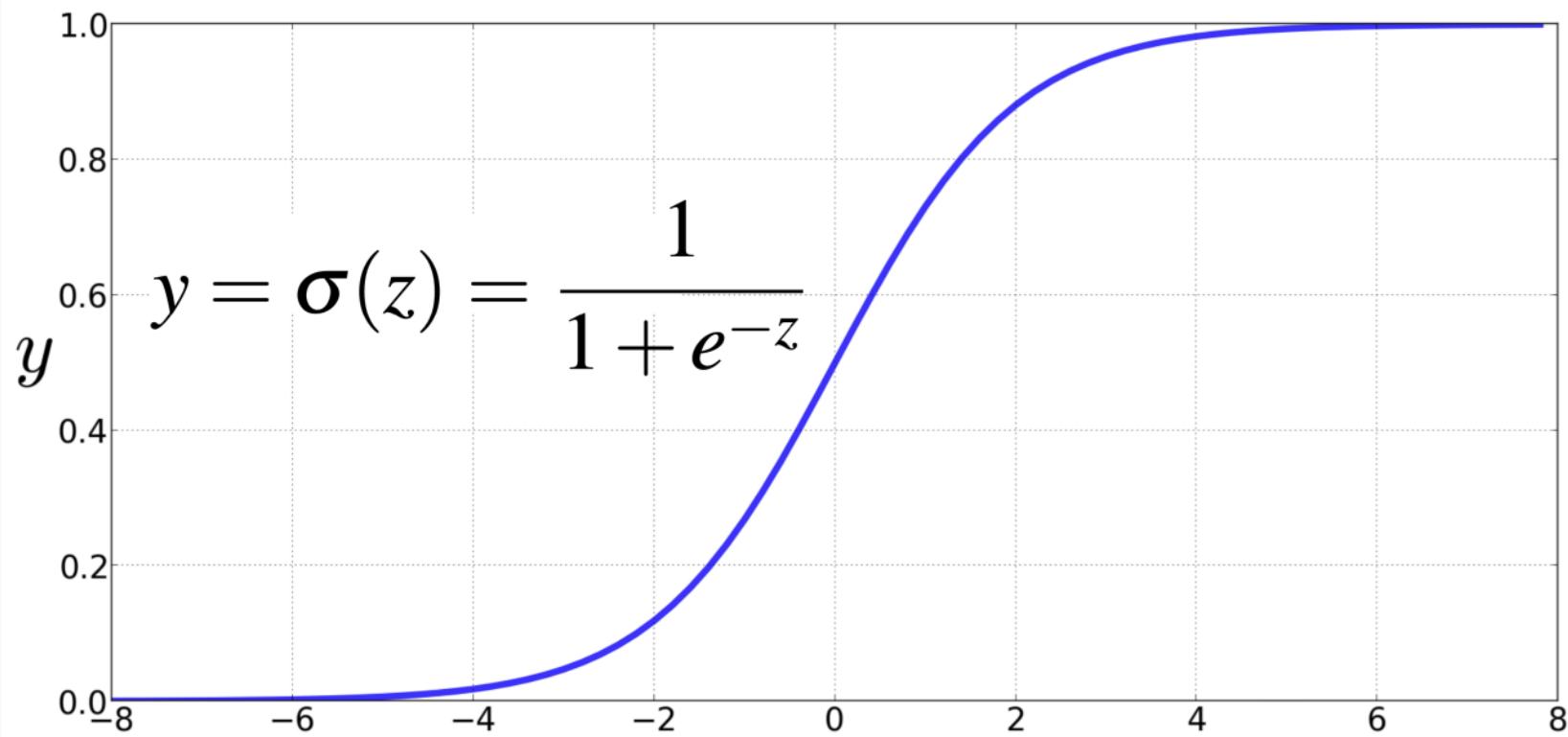
$z$  is just a number:

$$z = w \cdot x + b$$

**Solution:** use a function of  $z$  that goes from 0 to 1, like the **logistic function** or **sigmoid function**:

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

# The Sigmoid Function



# Logistic Regression in Three Easy Steps

1. Compute  $w \cdot x + b$
2. Pass it through the sigmoid function:  $\sigma(w \cdot x + b)$
3. Treat the result as a probability

# Making Probabilities with Sigmoids

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

$$\begin{aligned} P(y = 0) &= 1 - \sigma(w \cdot x + b) \\ &= 1 - \frac{1}{1 + \exp(-(w \cdot x + b))} \\ &= \frac{\exp(-(w \cdot x + b))}{1 + \exp(-(w \cdot x + b))} \end{aligned}$$

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

0.5 here is called the **decision boundary**

## Sentiment Classification: Movie Review

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 .

# Sentiment classification: feature engineering

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

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** .

$$x_1=3$$

$$x_5=0$$

$$x_6=4.19$$

$$x_4=3$$

Var	Definition	Value in Fig. 5.2
$x_1$	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
$x_6$	$\ln(\text{word count of doc})$	$\ln(66) = 4.19$

## Classifying Sentiment for Input $x$

Var	Definition	Val
$x_1$	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 & 2nd pronouns) $\in$ doc	3
$x_5$	$\begin{cases} 1 & \text{if "!" } \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$	0
$x_6$	$\log(\text{word count of doc})$	$\ln(66) = 4.19$

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

# Performing the calculations

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

Var	Val
$x_1$	3
$x_2$	2
$x_3$	1
$x_4$	3
$x_5$	0
$x_6$	$\ln(66) = 4.19$

$p(+|x) = P(Y = 1|x) = \sigma(w \cdot x + b)$

$p(-|x) = P(Y = 0|x) =$

# Coding activity: custom features for logistic regression

---

# Notebook: custom features for logistic regression

1. Go to this [nbgitpuller link](#) (also available on course website)
2. Start a server with **TEACH – 6 CPUs, 48 GB**
3. Load custom environment at `/ix1/cs1671-2026s/class_env`
4. Open `session8_logistic_regression_features.ipynb`