CS 1671 / CS 2071 / ISSP 2071
Human Language Technologies

Session 11: Multinomial logistic regression, text classification evaluation

Michael Miller Yoder

February 28, 2026

Quiz

- Go to **Quizzes > Quiz 02-18** on Canvas.
  - Covers Session 10: J+M 4.7-4.10, 4.12
- 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

Course logistics: project

- Project proposal is due **next Thu Feb 27**
  - Let me know if you have questions about the info I sent over email
- Homework 2 on text classification will be released Fri Feb 20

Midterm course evaluation (OMETs)

- CS 1671: https://go.blueja.io/6of2IE3sDE6Jd3SilhQWWw
- CS 2071: https://go.blueja.io/tMB2YLQ5UkCMqm5Xwtkf1w
- ISSP 2071: https://go.blueja.io/1PQ6V3voLEenbXmezougOA
- All types of feedback are welcome (critical and positive)
- Completely anonymous, will not affect grades
- Let me know what's working and what to improve on while the course is still running!
- Please be as specific as possible
- Available until **next Mon Feb 23**

CS 1671 - HUMAN

Review: learning the weights in logistic regression for binary classification

Muddiest point responses from last time:

- how this is relevant in nlp and what each part of the formulas entail
  - Cross-entropy loss: $L_{\text{CE}}(\hat{y}, y) = -[y \log \hat{y} + (1-y) \log(1-\hat{y})]$
  - Update weights: $\mathbf{w}_{t+1} = \mathbf{w}_t - \eta \frac{d}{dw} L_{CE}(f(\mathbf{x}; \mathbf{w}), y)$
- how we got the new updated weights
- the concept applied on multi dimensional vectors
- do you update one weight at a time or all weights at once
- when does the derivative get taken

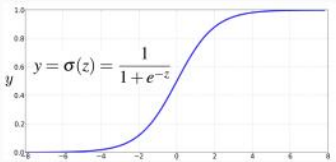
Lecture overview: N-gram language models part 2, text classification

- Multinomial logistic regression
  - Inference: using trained weights to make predictions
  - Training: learning the weights
- Evaluation of text classification
  - Precision, recall, f1-score
  - Train/dev/test and cross-validation sets
- Harms in classification
- Coding activity
  - Clickbait classification evaluation and error analysis

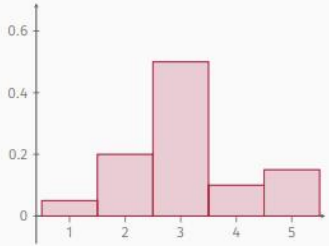
Multinomial logistic regression classification

Softmax is a Generalization of Sigmoid

Sigmoid makes its output look like a probability (forcing it to be between 0.0 and 1.0) and "squashes" it so that the output will tend to 0.0 or 1.0. Concerned about one class? Sigmoid is perfect.



For multiple classes, we do not want a probability—we want a probability **distribution**.

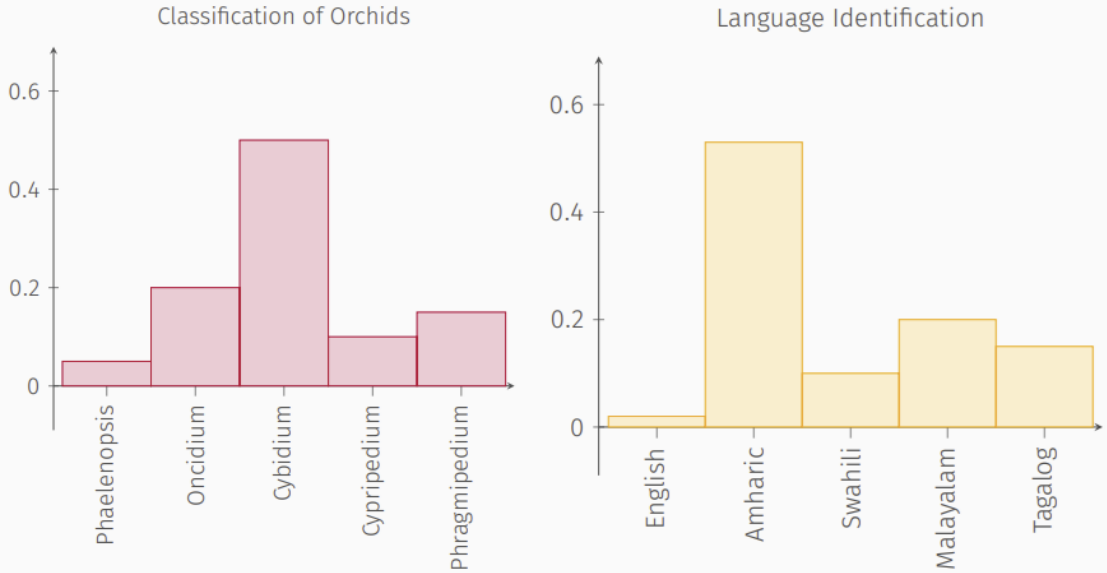


Instead of a sigmoid function, we will use SOFTMAX.

Slide credit: David Mortensen

What is a Probability Distribution?

A probability distribution is a function giving the probabilities that different possible outcomes of an experiment will occur. Our probability distributions will usually be over DISCRETE RANDOM VARIABLES.



Slide credit: David Mortensen

The Softmax Function

The formula for the softmax function is

$$\text{softmax}(\mathbf{z}_i) = \frac{\exp(\mathbf{z}_i)}{\sum_{j=1}^{K} \exp(\mathbf{z}_j)} \quad 1 \leq i \leq K$$

where K is the number of dimensions in the input vector $\mathbf{z}$. Compare it to the formula for the sigmoid function:

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

The formulas are very similar, but sigmoid is a function from a scalar to a scalar, whereas softmax is a function from a vector to a vector.

This output vector is the probability distribution over classes.

Slide credit: David Mortensen

Computing z

Remember that, to compute z in logistic regression, we used the formula

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

where $\mathbf{w}$ is a vector of weights, $\mathbf{x}$ is a vector of features, and b is a scalar bias term. Thus, z is a scalar. For multinomial logistic regression, we need a vector $\mathbf{z}$ instead of a scalar z. Our formula will be

$$\mathbf{z} = \mathbf{W}\mathbf{x} + \mathbf{b}$$

where W is a matrix with the shape $[K \times f]$ (where K is the number of output classes and f is the number of input features). In other words, there is an element in $\mathbf{W}$ for each combination of class and feature. $\mathbf{x}$ is a vector of features. $\mathbf{b}$ is a vector of biases (one for each class).

Slide credit: David Mortensen

A Summary Comparison of Logistic Regression and Multinomial Logistic Regression

Logistic regression is

$$\hat{y} = \sigma(\mathbf{w}\mathbf{x} + b)$$

where y is, roughly, a probability.

Multinomial logistic regression (or SOFTMAX REGRESSION) is

$$\hat{\mathbf{y}} = \text{softmax}(\mathbf{W}\mathbf{x} + \mathbf{b})$$

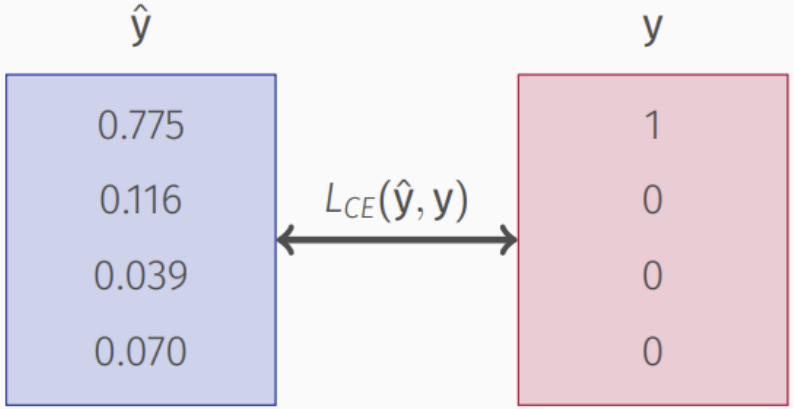
where $\hat{\mathbf{y}}$ is a PROBABILITY DISTRIBUTION over classes, $\mathbf{W}$ is a class $\times$ feature weight matrix, $\mathbf{x}$ is a vector of features, and $\mathbf{b}$ is a vector of biases.

Slide credit: David Mortensen

Training multinomial logistic regression

Categorical Cross-Entropy Loss for Multinomial Logistic Regression

Compare $\mathbf{y}$, a ONE-HOT VECTOR (one one, all other elements zero) and



How "distant" is $\hat{\mathbf{y}}$ from $\mathbf{y}$? One measure is categorical cross-entropy loss:

$$
\begin{aligned}
L_{CE} &= -\sum_{i=1} T_i \log S_i \\
&= -[1 \log_2 0.775 + 0 \log_2 0.126 + \\
&\quad 0 \log_2 0.039 + 0 \log_2 0.070] \\
&= -\log_2 0.775 \\
&= 0.3677
\end{aligned}
$$

The elements of $\hat{\mathbf{y}}$ that correspond to 0-elements in $\mathbf{y}$ are effectively ignored.

Slide credit: David Mortensen

Generalizing Your Losses: The Negative Log Likelihood Loss

Reminder: the loss function for binary logistic regression (LR with two classes) is

$$L_{CE}(\hat{y}, y) = -\log p(y \mid x) = -[y \log \hat{y} + (1-y)\log(1-\hat{y})]$$

Note that we have two terms—one for when $y = 1$ and one for when $y = 0$—corresponding to the two classes. What if we have K classes?

$$\begin{aligned}
L_{CE}(\hat{\mathbf{y}}, \mathbf{y}) &= -\sum_{k=1}^{K} \mathbf{y}_k \log \hat{\mathbf{y}}_k \\
&= -\log \hat{\mathbf{y}}_c \quad \text{(where } c \text{ is the correct class)} \\
&= -\log \hat{p}(\mathbf{y}_c = 1 \mid \mathbf{x}) \quad \text{(where } c \text{ is the correct class)} \\
&= -\log \frac{\exp(\mathbf{w_c} \cdot \mathbf{x} + b_c)}{\sum_{j=1}^{K} \exp(\mathbf{w_j} \cdot \mathbf{x} + b_j)} \quad (c \text{ is the correct class})
\end{aligned}$$

Slide credit: David Mortensen

What We Actually Need to Compute Gradient Descent is the Gradient of the Loss

Consider one piece of the gradient—the derivative with respect to one weight.

- For each class k the weight of the ith element of $\mathbf{x}$ (the input features) is $\mathbf{w}_{k,i}$.
- What is the partial derivative of $L_{CE}(\hat{\mathbf{y}}, \mathbf{y})$ wrt $\mathbf{w}_{k,i}$?
- It turns out, after some math, that the difference between the true value for the class k (either 1 or 0) and the probability that the class outputs class k (weighted by the value of the input $\mathbf{x}_i$ corresponding to the ith element of the weight vector for class k.

$$\frac{\partial L_{CE}}{\partial \mathbf{w}_{k,i}} = -(\mathbf{y}_k - \hat{\mathbf{y}}_k)\mathbf{x}_i$$

- The rest of the procedure for training multinomial LR is the same as for binary LR.

Slide credit: David Mortensen

How to evaluate your classifier

Gold labels and predicted labels

Document	gold label	predicted label
just plain boring	–	–
entirely predictable	–	–
no surprises and very few laughs	–	+
very powerful	+	–
the most fun film of the summer	+	+

The **gold** label is the label that a human assigned to the document.

The **predicted** or **hypothesized** label is the label that the classifier assigned to the document.

Slide credit: David Mortensen

We Can Evaluate a Classifier Using Accuracy

Accuracy is our first shot.

- Accuracy:

$$\frac{\text{how many instances your system got right}}{\text{all instances in the test set}}$$

Slide credit: David Mortensen

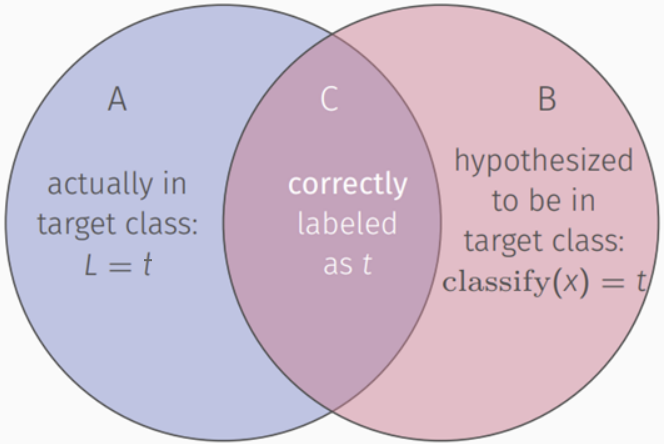
Issues with using test set accuracy

- Imagine an “important email” classifier that notifies you when you get an important email
- Suppose that 99% of the messages you receive are junk and not important (we’re being realistic here)
- An easy important email classifier: classify nothing as important
  - You would get lots of work done, because you wouldn’t be distracted by email
  - The email classifier would have an accuracy of ~99%
  - Everybody would be happy except for your boss
- You must take the relative importance of the classes into account, and the cost of the error types

Slide adapted from David Mortensen

Evaluation in the Two-class case

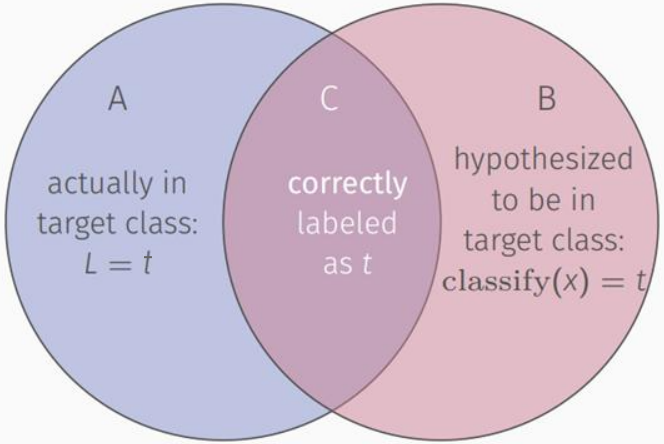
- Suppose we have one of the classes $t \in \mathcal{L}$ as the target class.
- We would like to identify documents with label t in the test data.
- We get



- Precision $\hat{P} = \frac{C}{B}$ (percentage of documents `classify` correctly labeled as t)
- Recall $\hat{R} = \frac{C}{A}$ (percentage of actual t labeled documents correctly labeled as t)
- $F_1 = 2\frac{\hat{P} \cdot \hat{R}}{\hat{P} + \hat{R}}$

Slide credit: David Mortensen

A Different View – Contingency Tables



	$L = t$	$L \neq t$	
$\texttt{classify}(X) = t$	C (true positives)	$B \backslash C$ (false positives)	B
$\texttt{classify}(X) \neq t$	$A \backslash C$ (false negatives)	(true negatives)	
	A		

precision = tp/(tp+fp)

recall = tp/(tp+fn)

Slide credit: David Mortensen

Why precision and recall

- 2-way precision and recall are specific to a target class
- Accuracy=99% on important email detection

  but

- Recall = 0 (out of all actually important emails, got none)
- Precision and recall, unlike accuracy, emphasize true positives: finding the things that we are supposed to be looking for

Slide adapted from Jurafksy & Martin

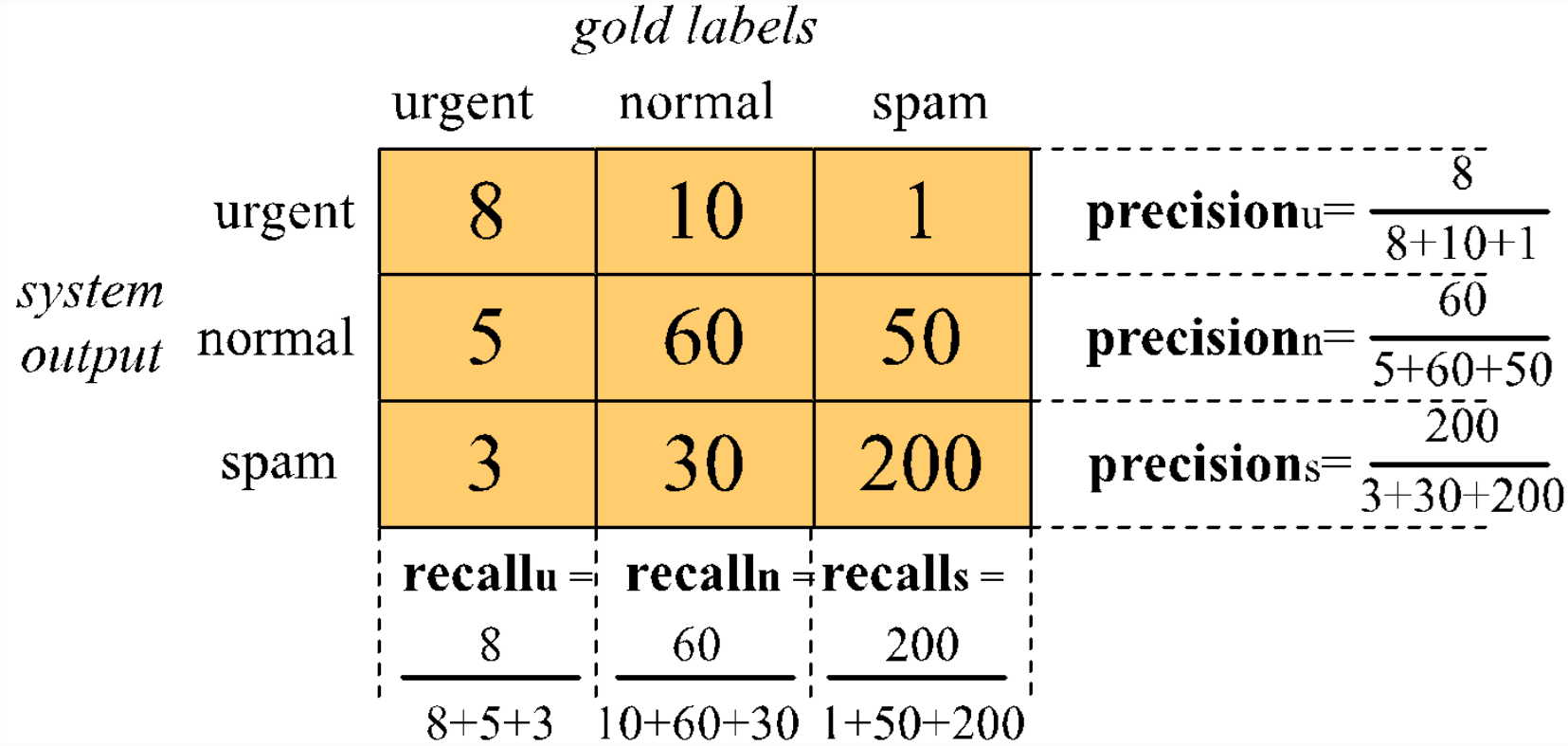
A combined measure: F1-score

We almost always use balanced F_1 (i.e., $\beta = 1$). Harmonic mean

$$F_1 = \frac{2PR}{P+R}$$

Slide adapted from Jurafksy & Martin

Confusion matrix for 3-class classification

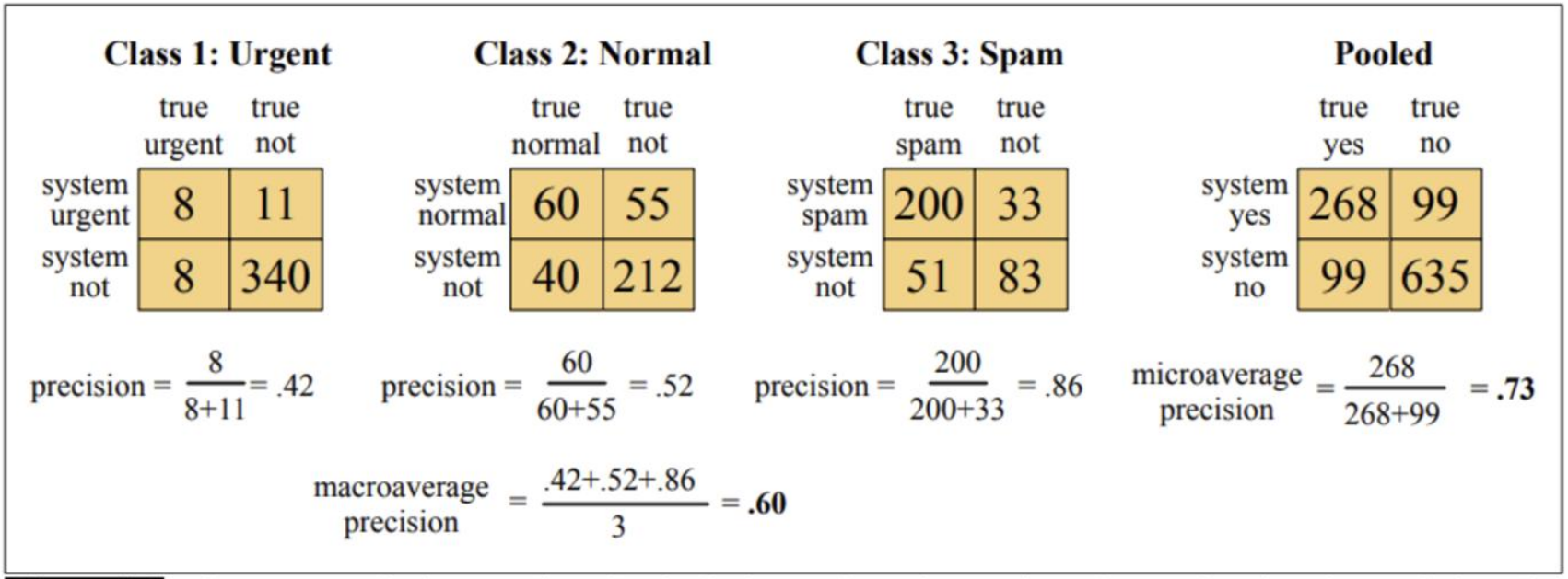


Slide adapted from Jurafksy & Martin

Evaluation with > 2 Classes

- **Macroaveraged precision and recall**: let each class be the target and report the average $\hat{P}$ and $\hat{R}$ across all classes.
- **Microaveraged precision and recall**: pool all one-vs.-rest decisions into a single contingency table, calculate $\hat{P}$ and $\hat{R}$ from that.

Example of more than two classes



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

$$\text{microaverage precision} = \frac{268}{268+99} = \mathbf{.73}$$

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

Figure 4.6 Separate confusion matrices for the 3 classes from the previous figure, showing the pooled confusion matrix and the microaveraged and macroaveraged precision.

Slide credit: David Mortensen, Jurafksy & Martin

Harms in classification in NLP

Harms in sentiment classifiers

Kiritchenko and Mohammad (2018) found that most sentiment classifiers assign lower sentiment and more negative emotion to sentences with African American names in them.

This perpetuates negative stereotypes that associate African Americans with negative emotions

Slide adapted from Jurafksy & Martin

Harms in toxicity classification

Toxicity detection is the task of detecting hate speech, abuse, harassment, or other kinds of toxic language

But some toxicity classifiers incorrectly flag as being toxic sentences that are non-toxic but simply mention identities like blind people, women, or gay people.

This could lead to censorship of discussion about these groups.

Slide adapted from Jurafksy & Martin

What causes these harms?

Can be caused by:

- Problems in the training data; machine learning systems are known to amplify the biases in their training data.
- Problems in the human labels
- Problems in the resources used (like lexicons)
- Problems in model architecture (like what the model is trained to optimized)

Mitigation of these harms is an open research area

Can't fully "remove" bias because exists in societies that produced texts we use

So need to be explicit about what those biases may be through **data statements** and **model cards**

Slide adapted from Jurafksy & Martin

Data statements [Bender & Friedman 2018]

For each dataset you release, document:

- Curation rationale: why were certain texts selected
- Language variety
- Speaker demographic
- Annotator demographic
- Speech situation
  - Time and place, modality, scripted vs spontaneous, intended audience
- Text characteristics
  - Genre, topic
- Recording quality (for speech)

Slide adapted from Jurafksy & Martin

Model cards [Mitchell et al. 2019]

For each algorithm you release, document:

- training algorithms and parameters
- training data sources, motivation, and preprocessing
- evaluation data sources, motivation, and preprocessing
- intended use and users
- model performance across different demographic or other groups and environmental situations

Discussion: data statements and model cards

- In an area of research or industry you are familiar with (in CS or outside CS), do you see issues with transparency of data and/or models?
  - Is it clear under what circumstances datasets were collected?
  - Are the intended uses of machine learning models clearly stated?
- What are limitations of data statements and model cards?

Conclusion

- Multinomial (multiclass) logistic regression uses softmax instead of sigmoid to get a probability distribution over classes
- Classifiers are evaluated with accuracy, precision, recall and F1-score
- Text classification systems can be biased against the language or references to marginalized groups

Coding activity:
clickbait classifier evaluation and error analysis

Notebook: clickbait classifier evaluation and error analysis

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 **session11_clickbait_error_analysis.ipynb**

Questions?