

CSE/STAT 416

Classification

Vinitra Swamy
University of Washington
July 6, 2020



Roadmap So Far

1. Housing Prices - Regression

- Regression Model
- Assessing Performance
- Ridge Regression
- LASSO

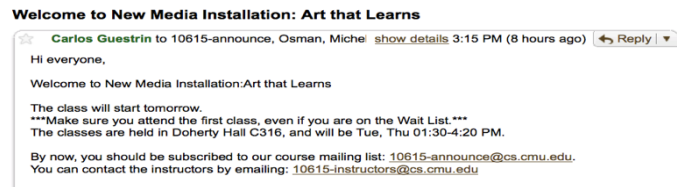
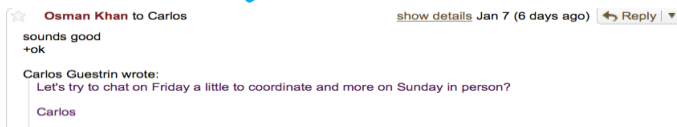
$$L2 = \sum_{j=0}^D W_j^2$$
$$L1 = \sum_{j=0}^D |W_j|$$

2. Sentiment Analysis – Classification

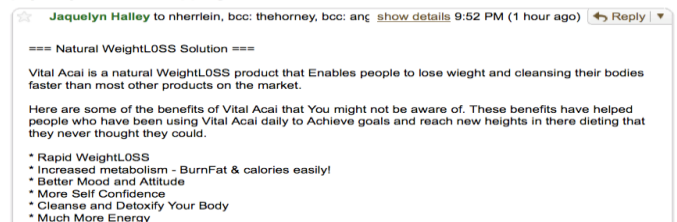
- Classification Overview
- Logistic Regression
- Decision Trees
- Ensemble Methods

Spam Filtering

Binary Classification



Natural _LoseWeight SuperFood Endorsed by Oprah Winfrey, Free Trial 1 bottle, pay only \$5.95 for shipping mfw rik [Spam](#) [X](#)



Spam

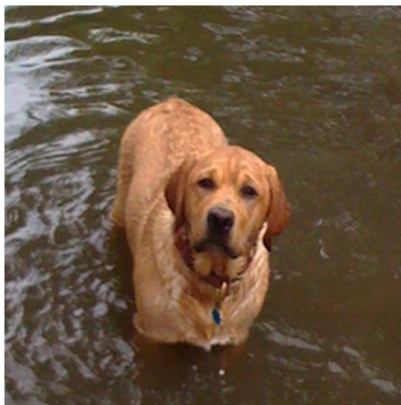
Not Spam
(ham)

Input: x
Text of email
Sender
Subject
...

Output: y

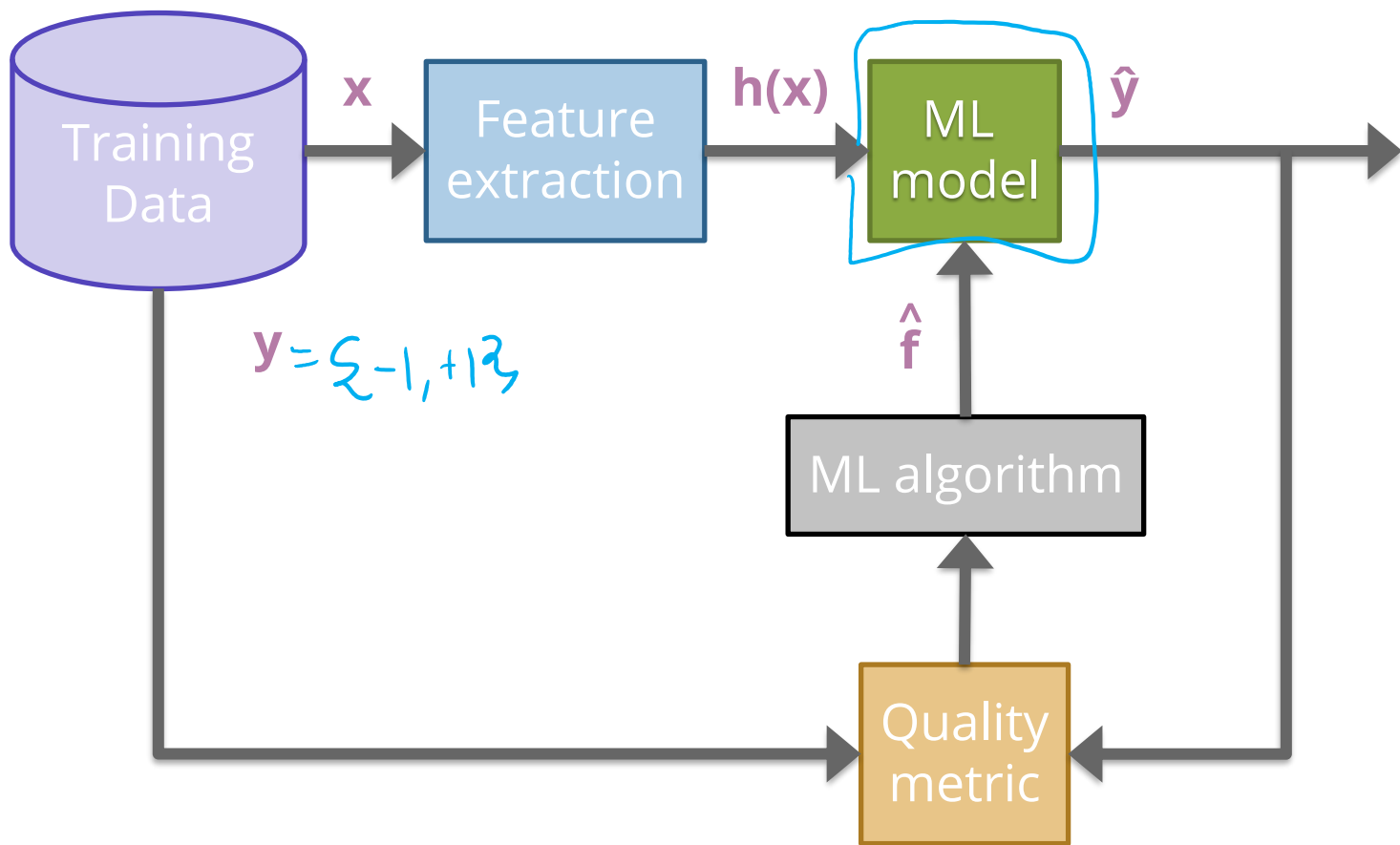
Object Detection

multi-class classification



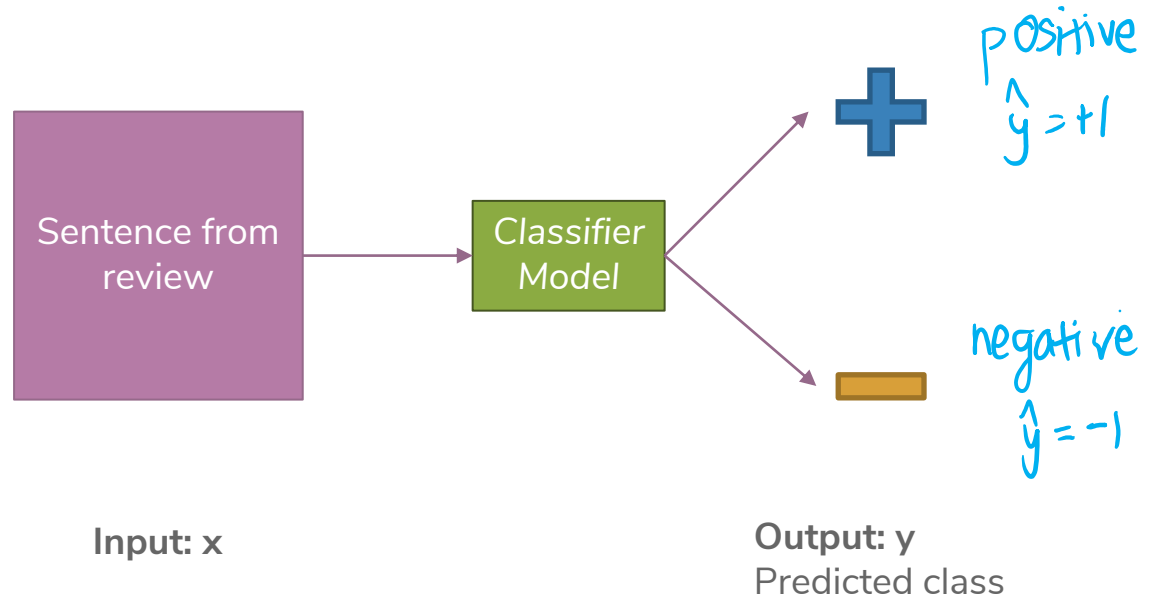
Input: x
Pixels

Output: y
Class
(+ Probability)



Sentiment Classifier

In our example, we want to classify a restaurant review as positive or negative.



Implementation 1: Simple Threshold Classifier

Idea: Use a list of good words and bad words, classifier by most frequent type of word

- Positive Words: great, awesome, good, amazing, ...
- Negative Words: bad, terrible, disgusting, sucks, ...

Simple Threshold Classifier

Input x : Sentence from review

- Count the number of positive and negative words, in x
- If $\text{num_positive} > \text{num_negative}$:
 - $\hat{y} = +1$
- Else:
 - $\hat{y} = -1$

#pos = 2
#neg = 1
 $\hat{y} = +1$

Example: "Sushi was great, the food was awesome, but the service was terrible"

Problems with Threshold

How do we get list of positive/negative words?

Words have different degrees of sentiment.

- Great > Good
- How can we weigh them differently?

} learn classifier

Single words are not enough sometimes...

- "Good" → Positive
- "Not Good" → Negative

} more complex features

n-gram encoding
bigrams → 2 wds at a time
unigram → 1 wd at a time

Implementation 2: Linear Classifier

Idea: Use labelled training data to learn a weight for each word.
Use weights to score a sentence.

Word	Weight
good	1.0
great	1.5
awesome	2.7
bad	-1.0
terrible	-2.1
awful	-3.3
restaurant, the, we, where, ...	0.0
...	...

Score a Sentence

Word	Weight
good	1.0
great	1.5
awesome	2.7
bad	-1.0
terrible	-2.1
awful	-3.3
restaurant, the, we, where, ...	0.0
...	...

Input x_i :

“Sushi was **great**, the food was **awesome**, but the service was **terrible**”

$$\begin{aligned} \text{score}(x_i) &= 2.1 \\ 1.0 * 0 \\ + 1.5 * 1 \\ + 2.7 * 1 \\ - 1.0 * 0 \\ - 2.1 * 1 \\ + 0.0 * 2 \end{aligned}$$

Linear classifier, because output is linear weighted sum of inputs.

Will learn how to learn weights soon!

Implementation 2: Linear Classifier


Idea: Use labelled training data to learn a weight for each word.
Use weights to score a sentence.

- See last slide for example weights and scoring.

Linear Classifier

Input x : Sentence from review

- Compute $Score(x)$
- If $Score(x) \geq 0$:
 - $\hat{y} = +1$
- Else:
 - $\hat{y} = -1$



$Score(x_i) = 2.1$
 $\hat{y} = +1$

Linear Classifier Notation

$$h_1(x) = \# \text{ good}$$

$$h_2(x) = \# \text{ great}$$

Model: $\hat{y}_i = \text{sign}(\text{Score}(x_i))$

$$\text{sign}(0.8) = +1$$

$$\text{sign}(-1, 234, 567) = -1$$

$$\text{Score}(x_i) = w_0 h_0(x_i) + w_1 h_1(x_i) + \dots + w_D h_D(x_i)$$

$$= \sum_{j=0}^D w_j h_j(x_i)$$

$$= w^T h(x)$$

We will also use the notation

$$\hat{s}_i = \text{Score}(x_i) = w^T h(x_i)$$

$$\hat{y}_i = \text{sign}(\hat{s}_i)$$

Decision Boundary

Consider if only two words had non-zero coefficients

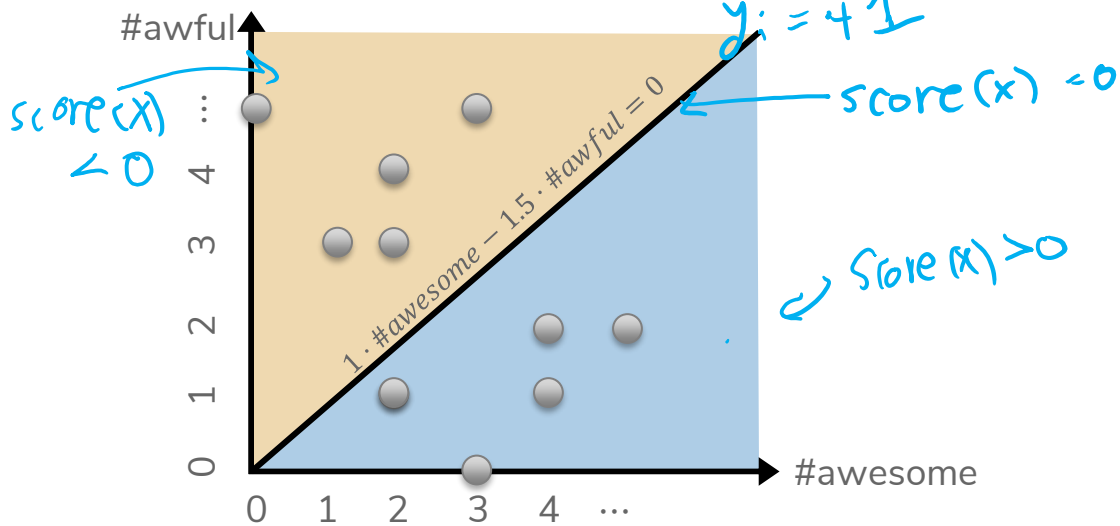
Word	Coefficient	Weight
	w_0	0.0
awesome	w_1	1.0
awful	w_2	-1.5

$$\hat{s} = 1 \cdot \#awesome - 1.5 \cdot \#awful$$

$$\hat{s}_i = 1 \cdot 5 - 1.5 \cdot 2$$

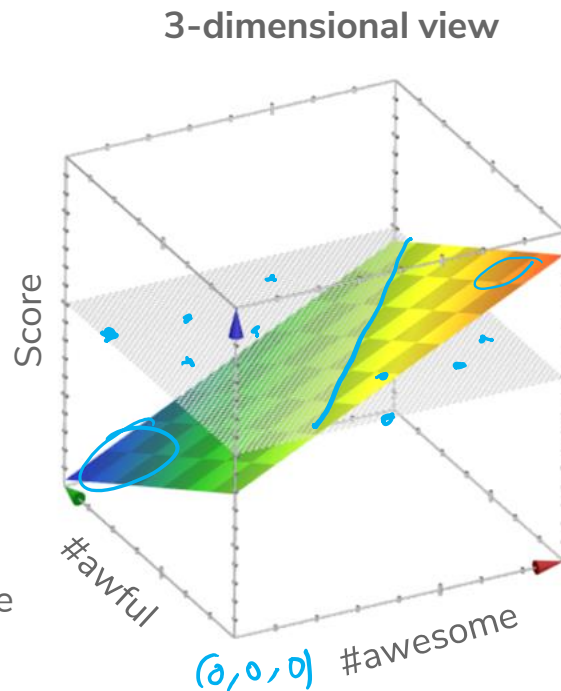
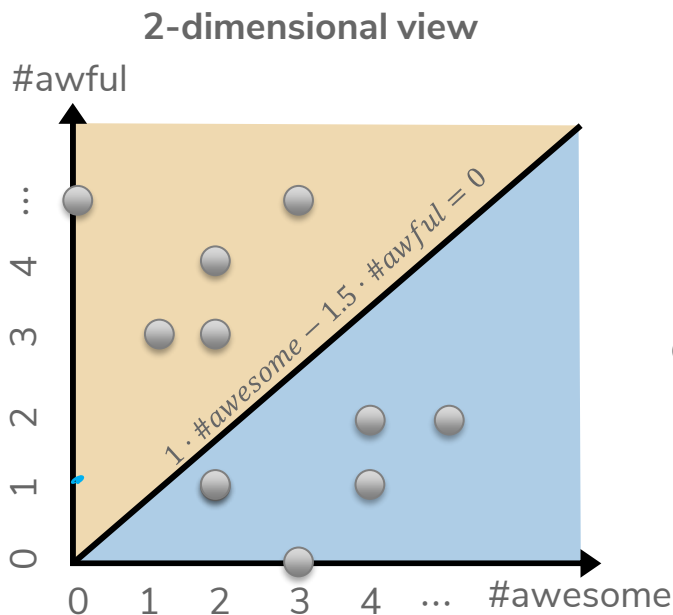
$$\hat{s}_i = 2$$

$$\hat{y}_i = 41$$



Decision Boundary

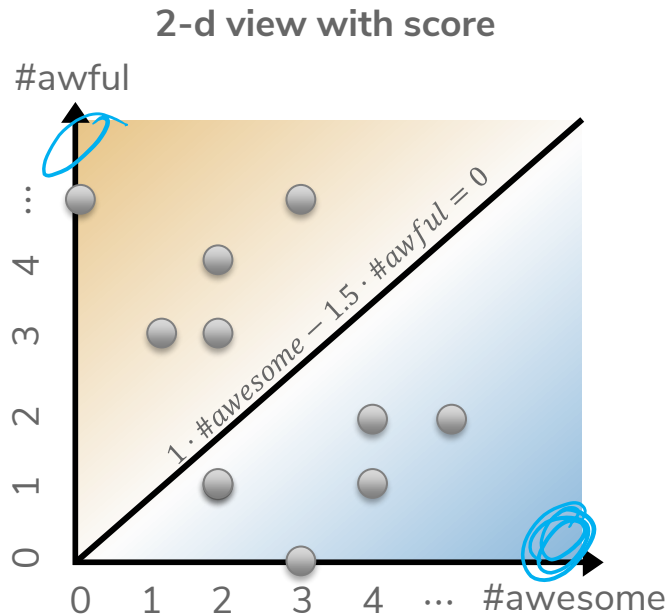
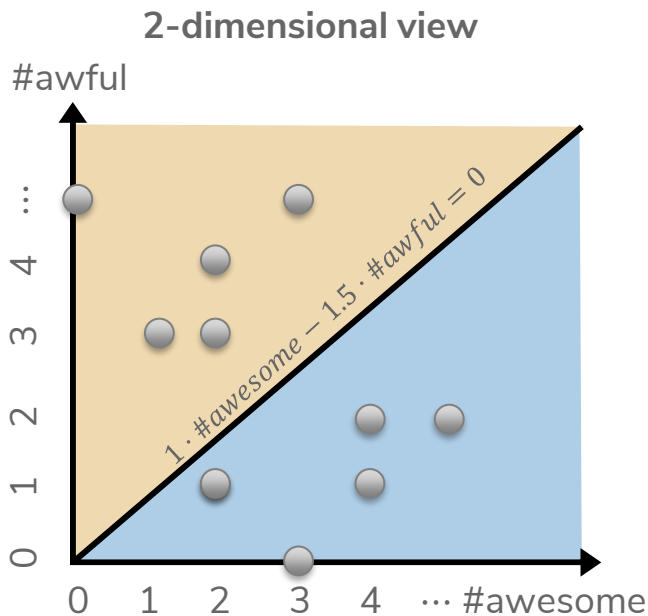
$$\text{Score}(x) = 1 \cdot \#awesome - 1.5 \cdot \#awful$$



Generally, with classification we don't use a plot like the 3d view since it's hard to visualize, instead use 2d plot with decision boundary

Decision Boundary

$$\text{Score}(x) = 1 \cdot \#awesome - 1.5 \cdot \#awful$$



- **Goal:** Get you actively participating in your learning
- Typical Activity
 - Question is posed
 - **Think** (1 min): Think about the question on your own
 - **Pair** (2 min): Talk with your neighbor to discuss question
 - If you arrive at different conclusions, discuss your logic and figure out why you differ!
 - If you arrived at the same conclusion, discuss why the other answers might be wrong!
 - **Share** (1 min): We discuss the conclusions as a class
- During each of the **Think** and **Pair** stages, you will respond to the question via a Poll Everywhere poll
 - The poll will only be open for the last **15** seconds of each of the stage
 - Not worth any points, just here to help you learn!

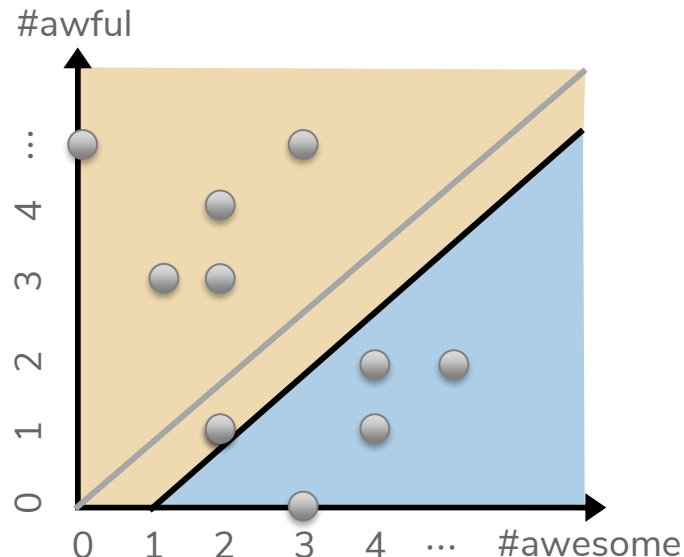
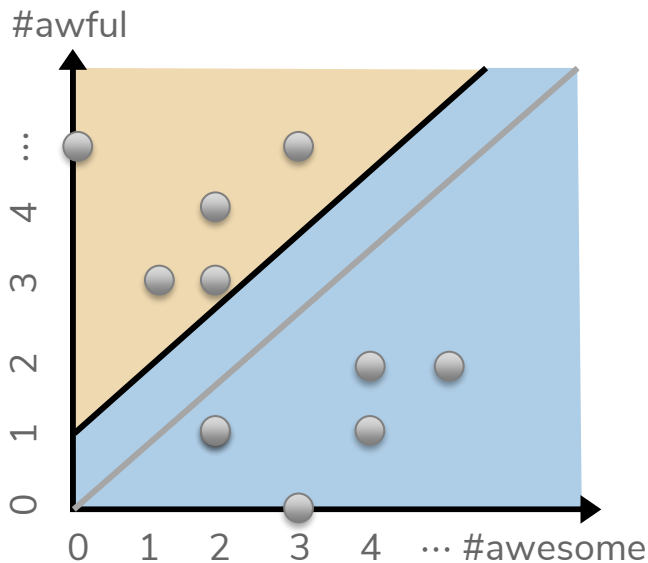
pollev.com/cse416

Think 

1 min

What happens to the decision boundary if we add an intercept?

$$\text{Score}(x) = 1.0 + 1 \cdot \#awesome - 1.5 \cdot \#awful$$



pollev.com/cse416

1:00

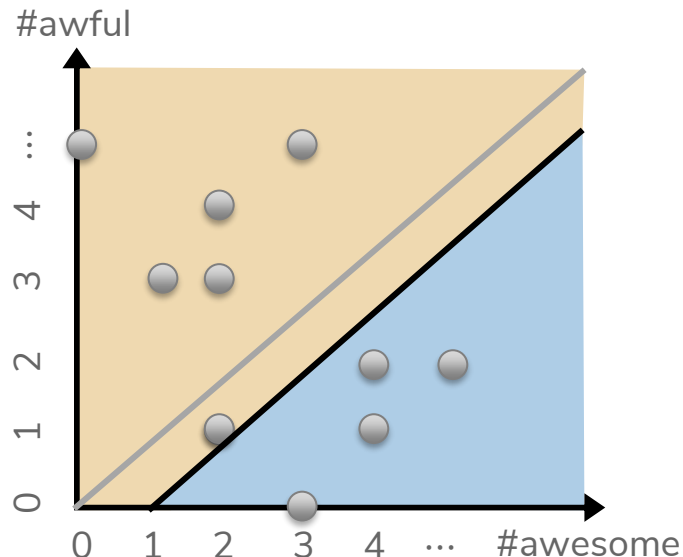
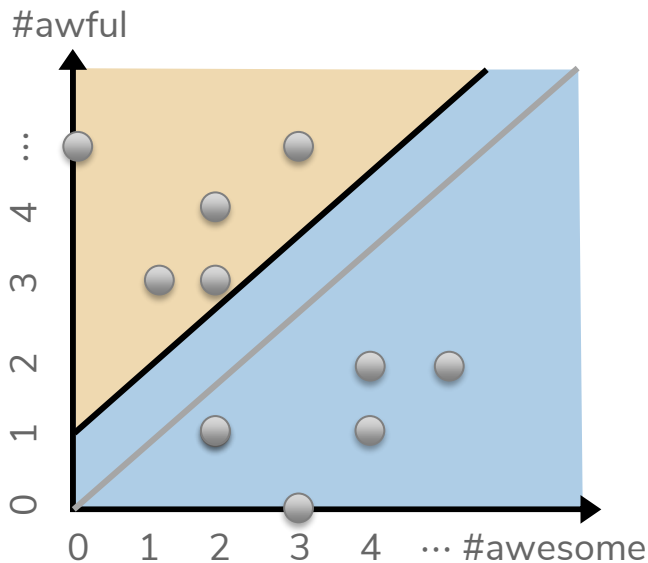
Pair 

2 min

pollev.com/cse416

What happens to the decision boundary if we add an intercept?

$$\text{Score}(x) = 1.0 + 1 \cdot \#awesome - 1.5 \cdot \#awful$$

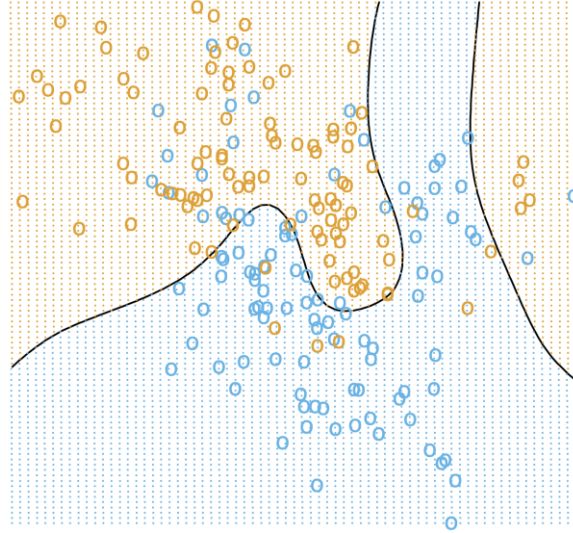


2:00

Complex Decision Boundaries?

What if we want to use a more complex decision boundary?

- Need more complex model/features!
- Covered next lecture!



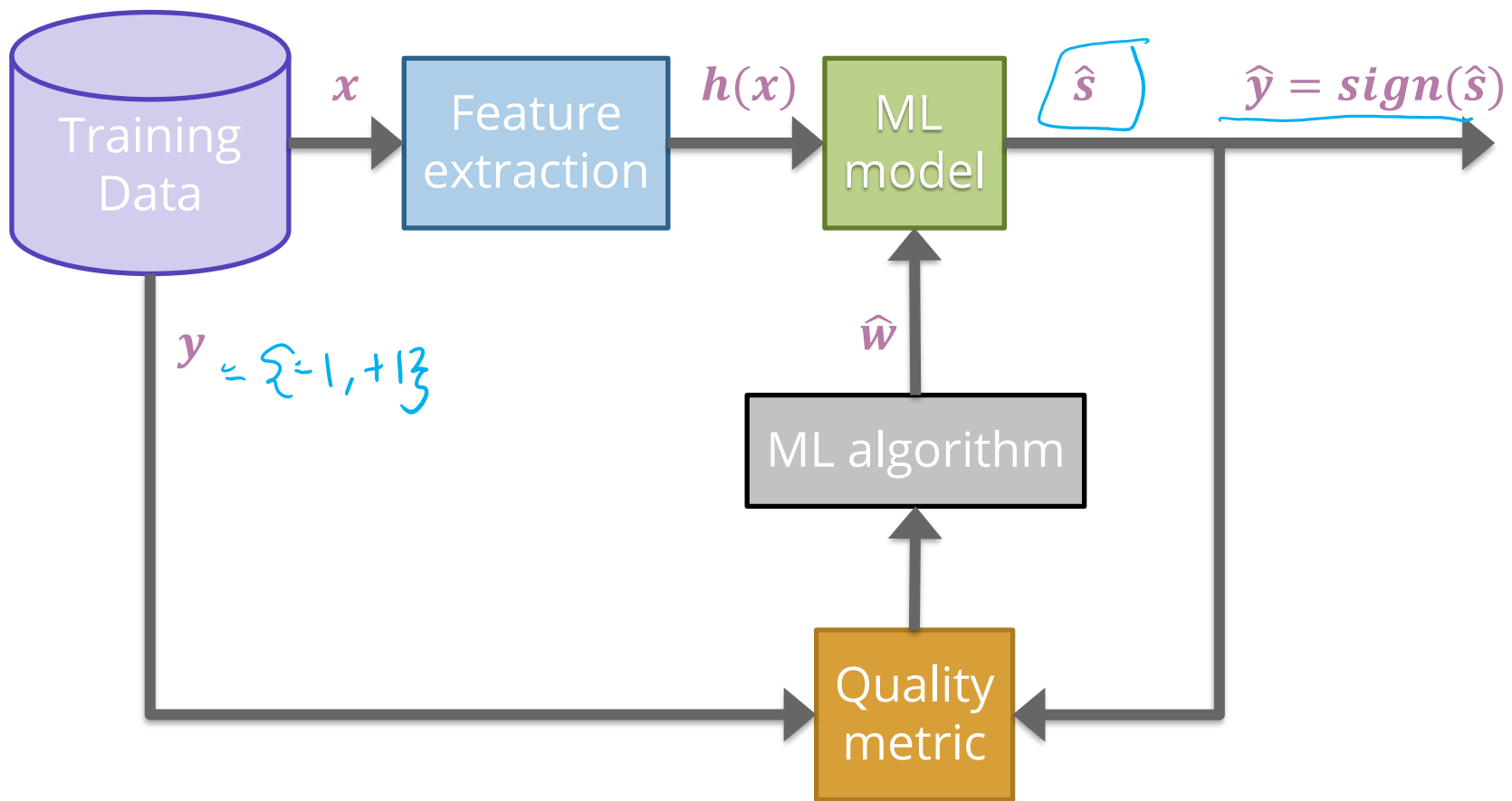


9:25

Brain Break



Evaluating Classifiers



Classification Error

indicator function

$$\mathbb{1}_{\{A\}} = \begin{cases} 1 & \text{if } A \text{ is true} \\ 0 & \text{otherwise} \end{cases}$$

Ratio of examples where there was a mistaken prediction

What's a mistake?

- If the true label was positive ($y = +1$), but we predicted negative ($\hat{y} = -1$)
- If the true label was negative ($y = -1$), but we predicted positive ($\hat{y} = +1$)

false negative
Type II

false positive
Type I

Classification Error

$$\frac{\text{\# mistakes}}{\text{\# examples}} = \frac{1}{n} \sum_{i=1}^n \mathbb{1}_{\{\hat{y}_i \neq y_i\}} = 1 - \text{accuracy}$$

Classification Accuracy

$$\frac{\text{\# Correct}}{\text{\# examples}} = \frac{1}{n} \sum_{i=1}^n \mathbb{1}_{\{\hat{y}_i = y_i\}} = 1 - \text{error}$$

What's a good accuracy?

For binary classification:

- Should at least beat random guessing...
- Accuracy should be at least 0.5

For multi-class classification (k classes):

- Should still beat random guessing
- Accuracy should be at least $1/k$
 - 3-class: 0.33
 - 4-class: 0.25
 - ...

Besides that, higher accuracy means better, right?

Detecting Spam

Imagine I made a “Dummy Classifier” for detecting spam

- The classifier ignores the input, and always predicts spam.
- This actually results in 90% accuracy! Why?
 - Most emails are spam...

This is called the **majority class classifier**.

A classifier as simple as the majority class classifier can have a high accuracy if there is a **class imbalance**.

- A class imbalance is when one class appears much more frequently than another in the dataset

This might suggest that accuracy isn't enough to tell us if a model is a good model.

Assessing Accuracy

Always digging in and ask critical questions of your accuracy.

- Is there a **class imbalance**?
- How does it compare to a baseline approach?
 - Random guessing
 - Majority class
 - ...
- Most important: **What does my application need?**
 - What's good enough for user experience?
 - What is the impact of a mistake we make?

Confusion Matrix

For binary classification, there are only two types of mistakes

- $\hat{y} = +1, y = -1$
- $\hat{y} = -1, y = +1$

Generally we make a **confusion matrix** to understand mistakes.

		Predicted Label	
		+	-
True Label	+	True Positive (TP)	False Negative (FN)
	-	False Positive (FP)	True Negative (TN)

Confusion Matrix Example

100 examples

		Predicted Label	
		+	-
True Label	+	True Positive (TP) 50	False Negative (FN) 10
	-	False Positive (FP) 5	True Negative (TN) 35

$$\frac{15}{100} \rightarrow 15\%$$

85%

Which is Worse?

What's worse, a false negative or a false positive?

- It entirely depends on your application!

Detecting Spam

False Negative: Annoying

False Positive: Email lost

Medical Diagnosis

False Negative: Disease not treated

False Positive: Wasteful treatment

In almost every case, how treat errors depends on your context.

Binary Classification Measures

Notation

- $C_{TP} = \#TP, C_{FP} = \#FP, C_{TN} = \#TN, C_{FN} = \#FN$
- $N = C_{TP} + C_{FP} + C_{TN} + C_{FN}$
- $N_P = C_{TP} + C_{FN}, N_N = C_{FP} + C_{TN}$

Error Rate

$$\frac{C_{FP} + C_{FN}}{N}$$

Accuracy Rate

$$\frac{C_{TP} + C_{TN}}{N}$$

False Positive rate (FPR)

$$\frac{C_{FP}}{N_N}$$

False Negative Rate (FNR)

$$\frac{C_{FN}}{N_P}$$

True Positive Rate or Recall

$$\frac{T_P}{N_P}$$

Precision

$$\frac{T_P}{C_{TP} + C_{FP}}$$

F1-Score

$$2 \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}}$$

[See more!](#)

Multiclass Confusion Matrix

Consider predicting (Healthy, Cold, Flu)

		Predicted Label		
		Healthy	Cold	Flu
True Label	Healthy	60	8	2
	Cold	4	12	4
	Flu	0	2	8

Think 

1 min

pollev.com/cse416

Suppose we trained a classifier and computed its confusion matrix on the training dataset. **Is there a class imbalance in the dataset and if so, which class has the highest representation?**

		Predicted Label		
		Pupper	Doggo	Woofers
True Label	Pupper	2	27	4
	Doggo	4	25	4
	Woofers	1	30	2

1:00

Suppose we trained a classifier and computed its confusion matrix on the training dataset. **Is there a class imbalance in the dataset and if so, which class has the highest representation?**

		Predicted Label		
		Pupper	Doggo	Woofer
True Label	Pupper	2	27	4
	Doggo	4	25	4
	Woofer	1	30	2

2:00

Learning Theory

How much data?

The more the merrier

- But data quality is also an extremely important factor

Theoretical techniques can bound how much data is needed

- Typically too loose for practical applications
- But does provide some theoretical guarantee

In practice

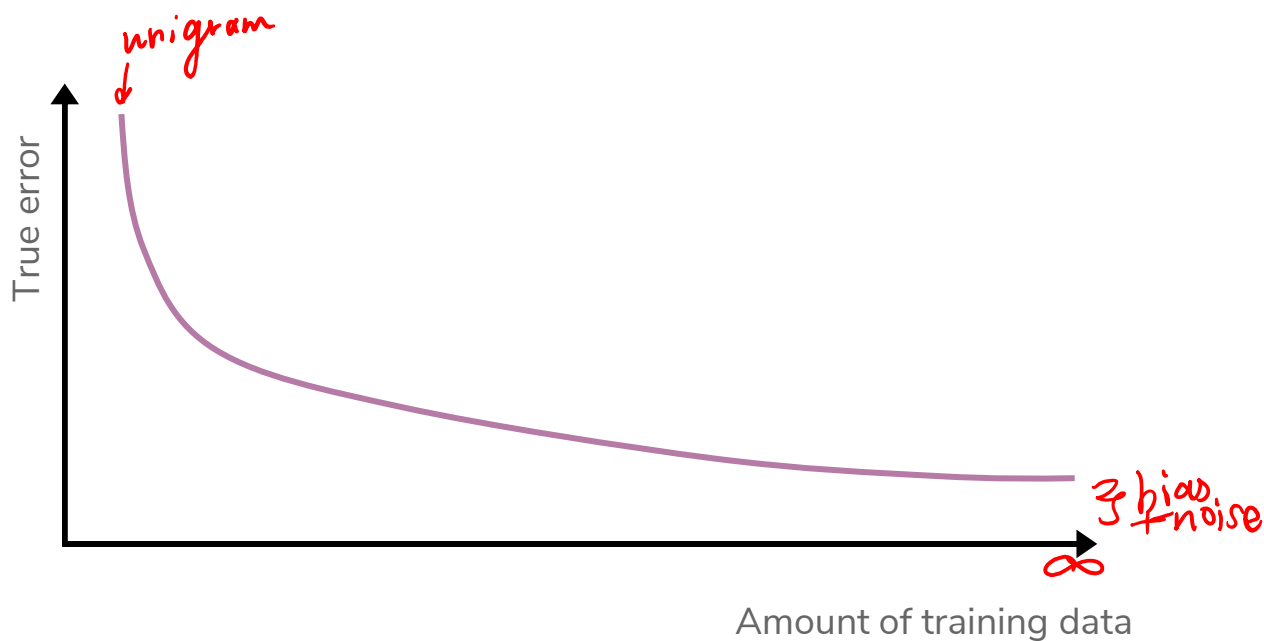
- More complex models need more data

VC dimension

Learning Curve

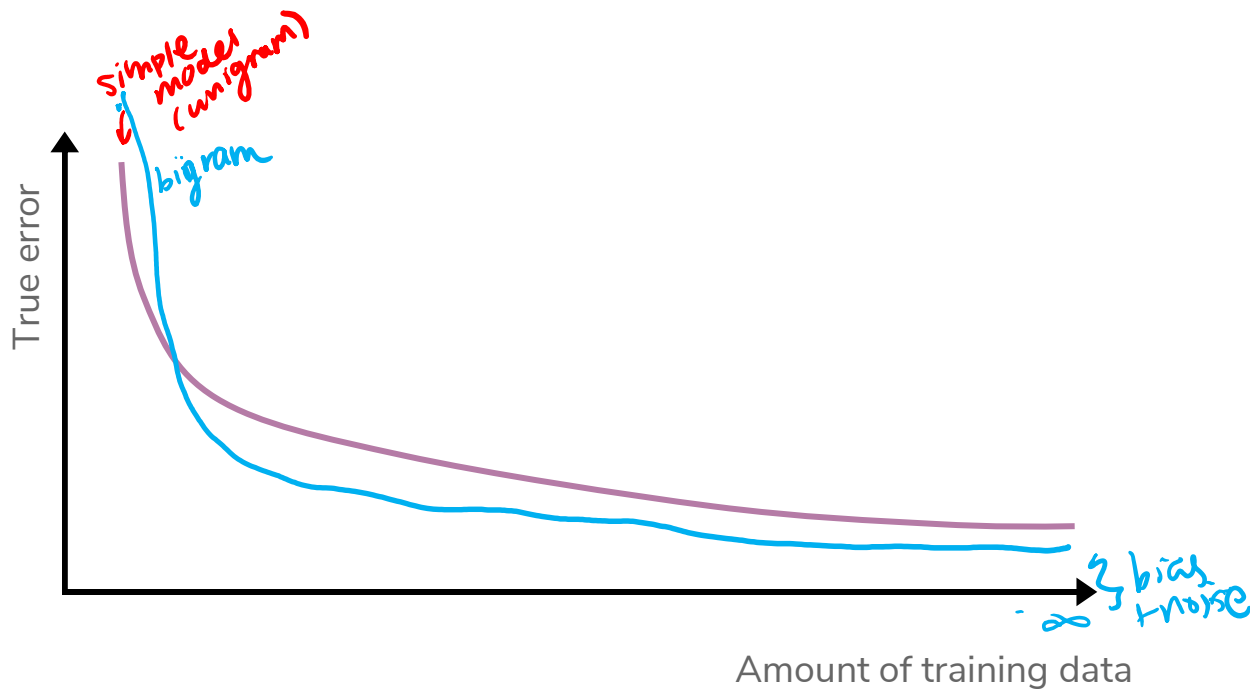
How does the true error of a model relate to the amount of training data we give it?

- Hint: We've seen this picture before



Learning Curve

What if we use a more complex model?



Change Threshold

What if I never want to make a false positive prediction?

Always predict negative ($\alpha = \infty$)

What if I never want to make a false negative prediction?

Always predict positive ($\alpha = -\infty$)

One way to control for our application is to change the scoring threshold. (Could also change intercept!)

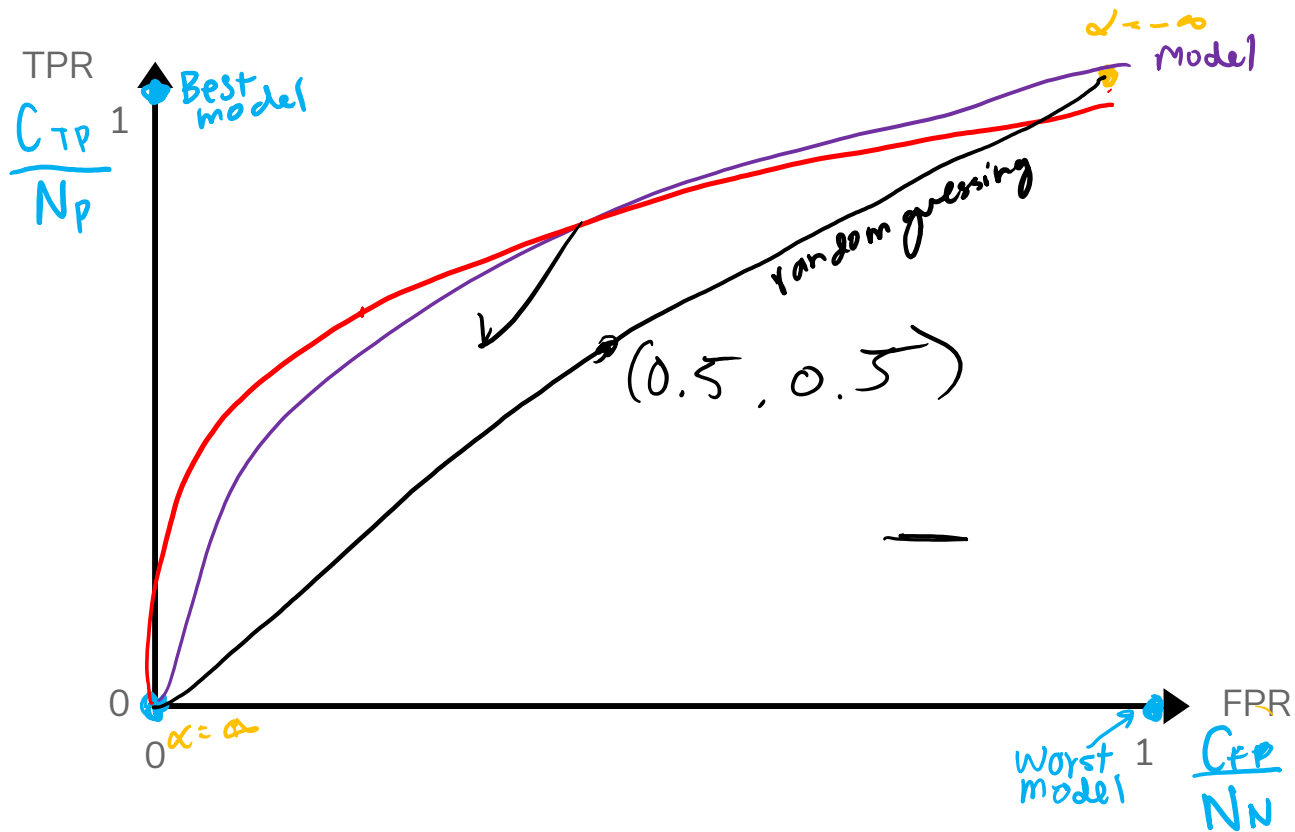
- If $Score(x) \geq \alpha$:
 - Predict $\hat{y} = +1$
- Else:
 - Predict $\hat{y} = -1$

$$\alpha = \infty$$

$$\alpha = -\infty$$

ROC Curve

What happens to our TPR and FPR as we increase the threshold?



Next Time

We will talk about learning classifiers that model the probability of seeing a particular class at a given input.

$$P(y|x)$$

Normally assume some structure on the probability (e.g. linear)

$$P(y|x, w) \approx w^T x$$

Use machine learning algorithm to learn approximate \hat{w} such that

$$\hat{P}(y|x) = P(y|x, \hat{w})$$

And $P(y|x)$ and $\hat{P}(y|x)$ are close.

Recap

Theme: Describe high level idea and metrics for classification

Ideas:

- Applications of classification
- Linear classifier
- Decision boundaries
- Classification error / Classification accuracy
- Class imbalance
- Confusion matrix
- Learning theory
- ROC Curve