Welcome! Ask questions or say "hi" before/during/after class!

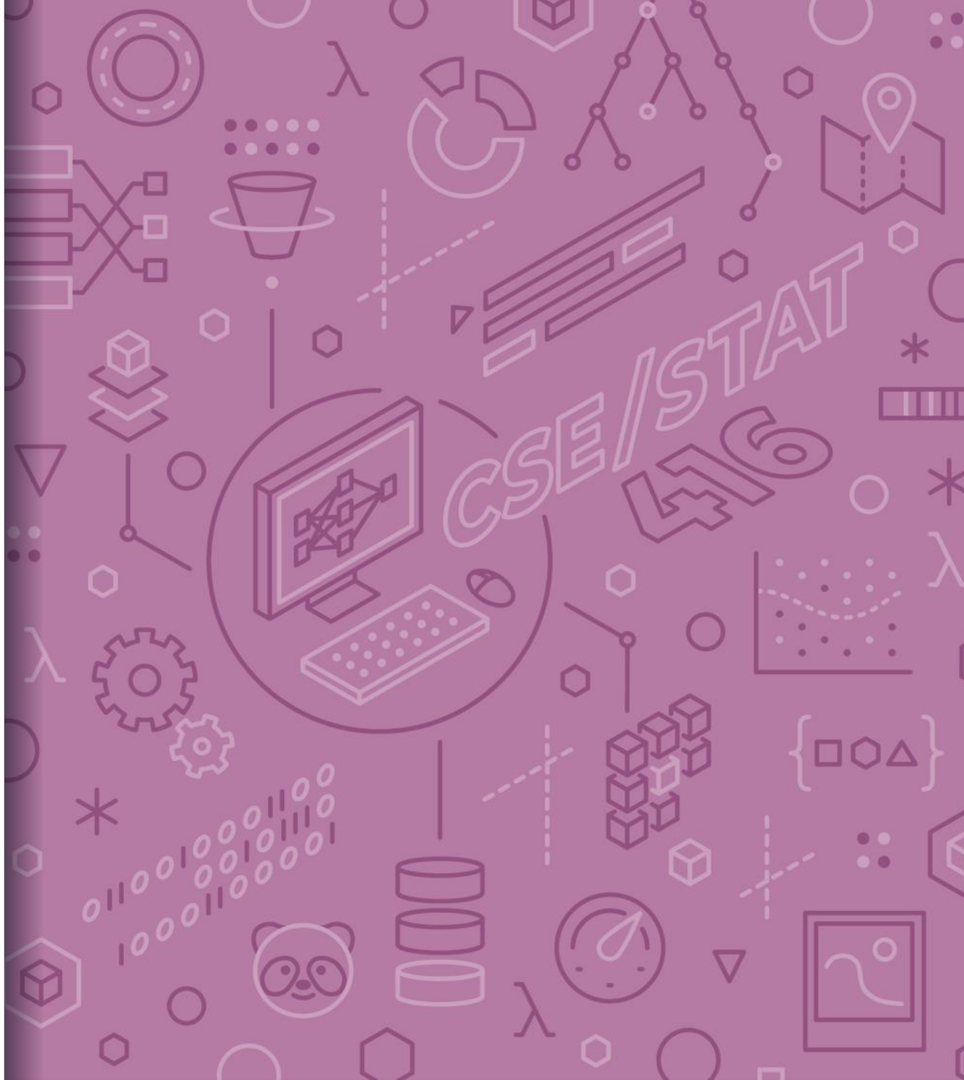
CSE/STAT 416

Classification

Hunter Schafer
Paul G. Allen School of Computer Science & Engineering
University of Washington

April 12, 2021

Music: Cataldo



Pre-Lecture Video 1

Classification

Roadmap So Far

1. Housing Prices - Regression
   - Regression Model
   - Assessing Performance
   - Ridge Regression
   - LASSO
2. Sentiment Analysis – Classification
   - Classification Overview
   - Logistic Regression

$$\|w\|_2^2 = \sum_{j=1}^{D} w_j^2$$

$$\|w\|_1 = \sum_{j=1}^{D} |w_j|$$

Spam Filtering

Binary Classification

Osman Khan to Carlos — show details Jan 7 (6 days ago) Reply

sounds good
+ok

Carlos Guestrin wrote:
Let's try to chat on Friday a little to coordinate and more on Sunday in person?

Carlos

Welcome to New Media Installation: Art that Learns

Carlos Guestrin to 10615-announce, Osman, Michel — show details 3:15 PM (8 hours ago) Reply

Hi everyone,

Welcome to New Media Installation:Art that Learns

The class will start tomorrow.
Make sure you attend the first class, even if you are on the Wait List.
The classes are held in Doherty Hall C316, and will be Tue, Thu 01:30-4:20 PM.

By now, you should be subscribed to our course mailing list: 10615-announce@cs.cmu.edu.
You can contact the instructors by emailing: 10615-instructors@cs.cmu.edu

Natural _LoseWeight SuperFood Endorsed by Oprah Winfrey, Free Trial 1 bottle, pay only $5.95 for shipping mfw rlk Spam | x

Jaquelyn Halley to nherrlein, bcc: thehorney, bcc: ang — show details 9:52 PM (1 hour ago) Reply

=== Natural WeightL0SS Solution ===

Vital Acai is a natural WeightL0SS product that Enables people to lose wieght and cleansing their bodies faster than most other products on the market.

Here are some of the benefits of Vital Acai that You might not be aware of. These benefits have helped people who have been using Vital Acai daily to Achieve goals and reach new heights in there dieting that they never thought they could.

* Rapid WeightL0SS
* Increased metabolism - BurnFat & calories easily!
* Better Mood and Attitude
* More Self Confidence
* Cleanse and Detoxify Your Body
* Much More Energy

Spam

Not Spam (ham)

Input: x

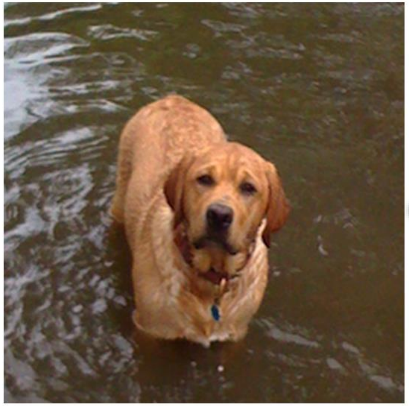
Text of email
Sender
Subject
...

Output: y

Spam
Ham

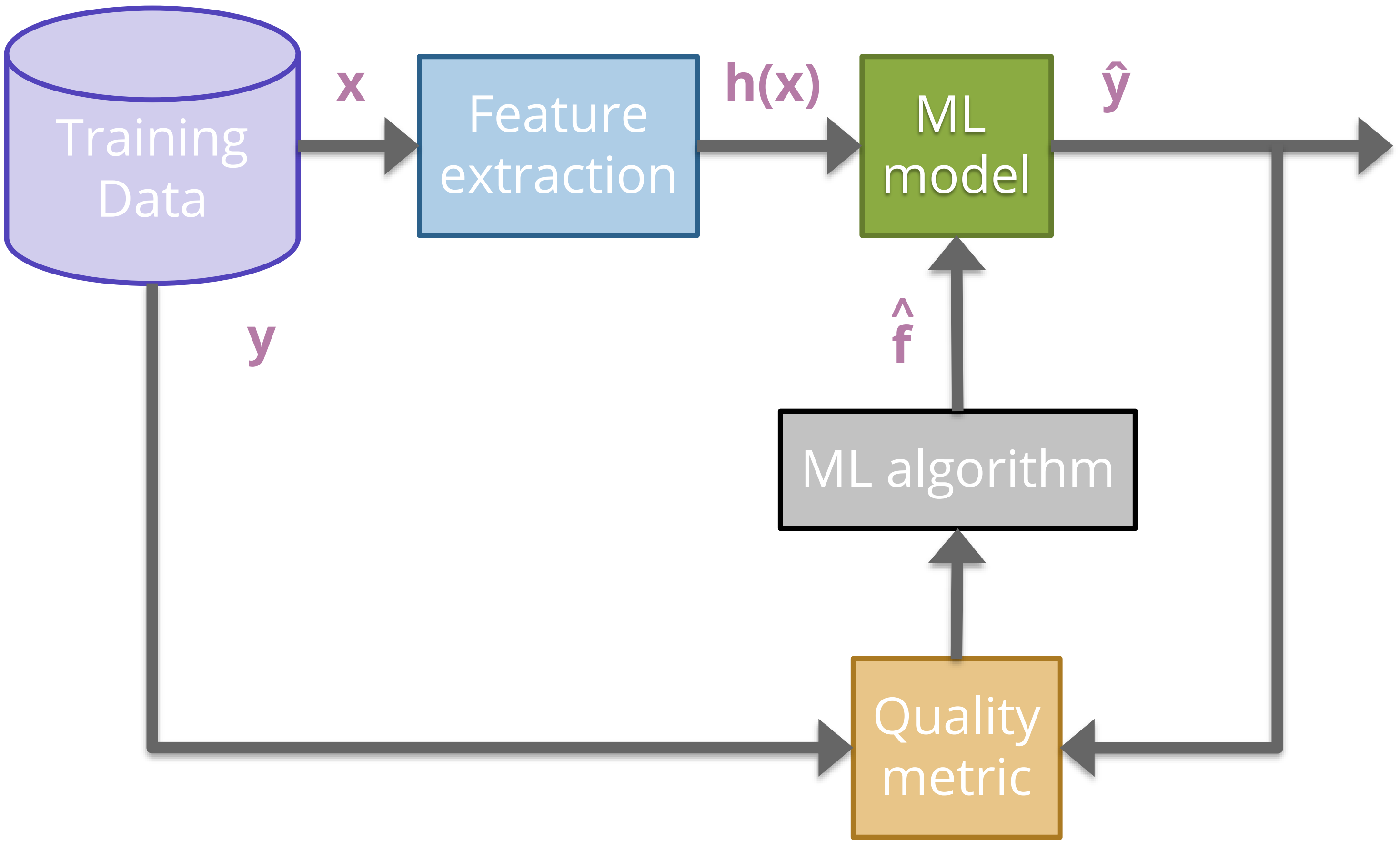
Object Detection

Multiclass Classification



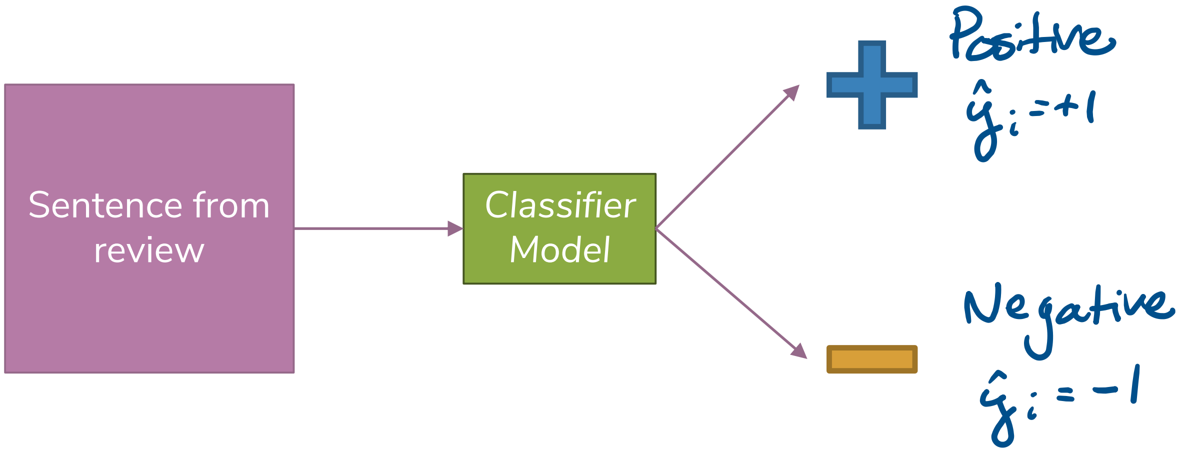
Input: x
Pixels

Output: y
Class
(+ Probability)


Training Data
x
Feature extraction
h(x)
ML model
ŷ
y
f̂
ML algorithm
Quality metric

Sentiment Classifier

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



Input: x

Output: y
Predicted class

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 $\mathrm{num\_positive} > \mathrm{num\_negative}$:
  - $\hat{y} = +1$
- Else:
  - $\hat{y} = -1$

pos=2, # neg=1

$\hat{y} = +1$

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

Limitations of Implementation 1

How do we get list of positive/negative words?

Words have different degrees of sentiment.

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

learn a classifier

Single words are not enough sometimes…

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

more complex features

1 word at a time ⇒ unigram

2 words at a time ⇒ bigram

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**"

$$\text{Score}(x_i) = 1.5 \cdot (1) + 2.7 \cdot (1) - 2.1 \cdot (1) = 2.1$$

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

$Score(x_i) = 2.1$

$\hat{y}_i = +1$

Linear Classifier Notation

$h_1(x) = \#$ good in x

$h_2(x) = \#$ awful in x

$sign(0.3) = +1$

$sign(-1,234,567) = -1$

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

$$Score(x_i) = w_0 h_0(x_i) + w_1 h_1(x_i) + \ldots + w_D h_D(x_i)$$

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

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

We will also use the notation

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

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

$\hat{s}_i \in [-\infty, \infty]$, $\hat{y}_i \in \{-1, +1\}$

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

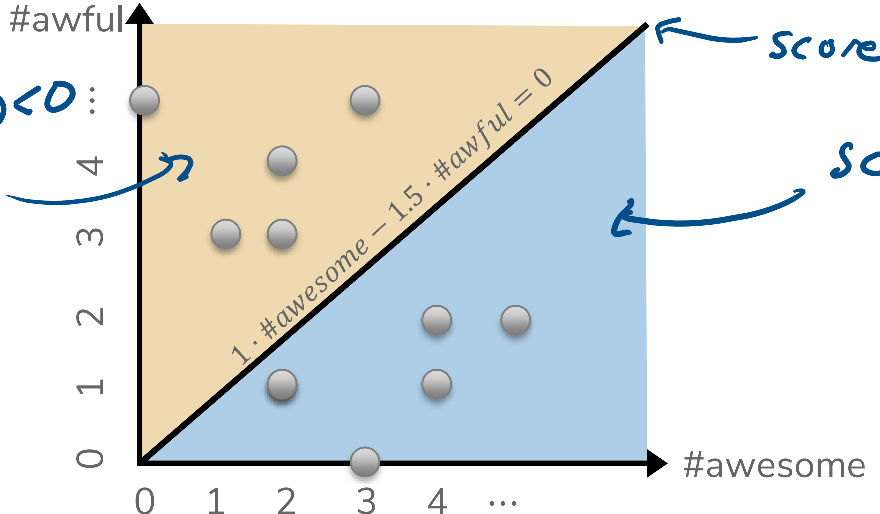
$\hat{s} = 1 \cdot \#awesome - 1.5 \cdot \#awful$

#awesome = 5
#awful = 2
score(x) = 1 · (5) − 1.5 · (2)
= 2

score(x) = 0

score(x) < 0
$\hat{y}$ = -1

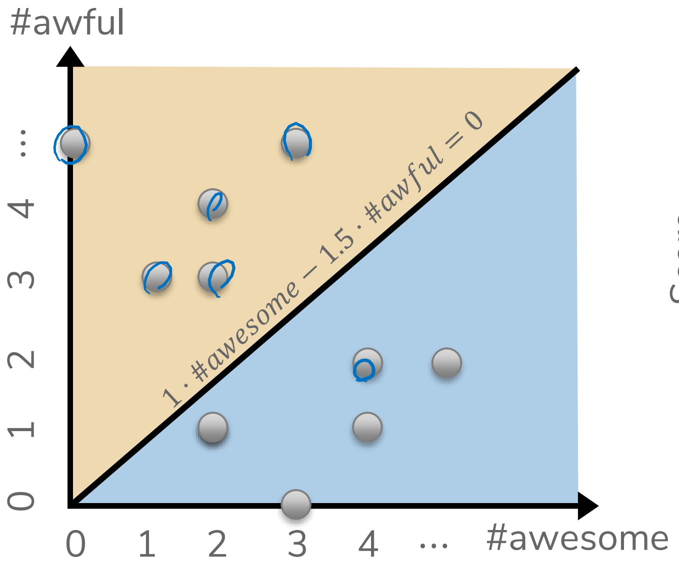
score(x) > 0
$\hat{y}$ = +1



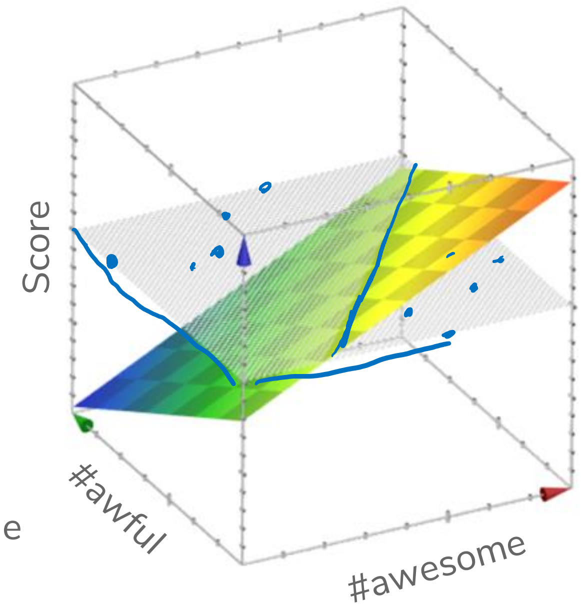
Decision Boundary

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

2-dimensional view



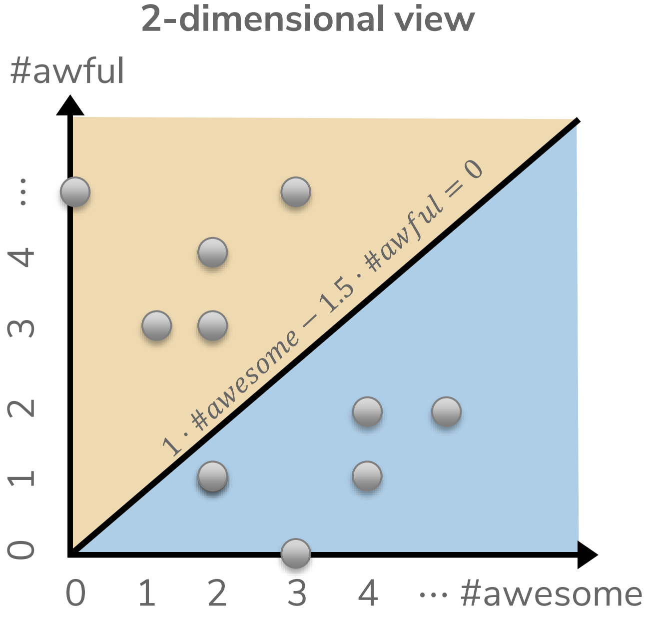
3-dimensional view

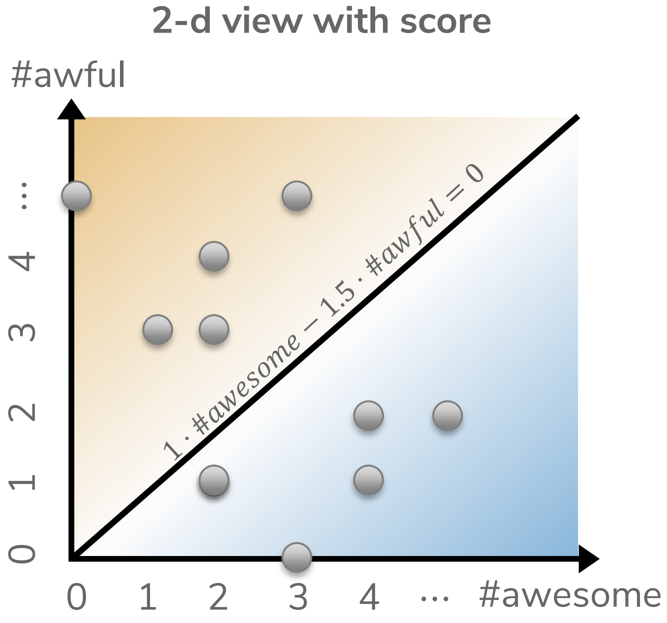


Generally, with classification we don't us a plot like the 3d view since it's hard to visualize, instead use 2d plot with decision boundary

Decision Boundary

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





Class Session

Poll Everywhere

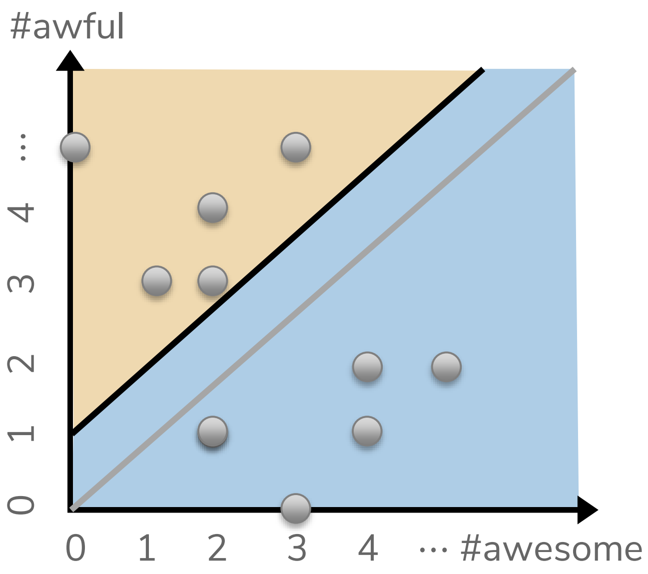
Think

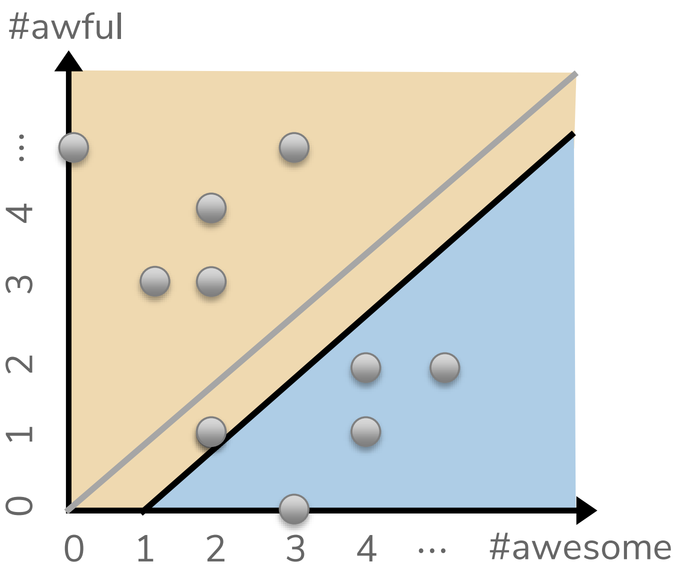
1 min

pollev.com/cs416

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

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





Poll Everywhere

Group

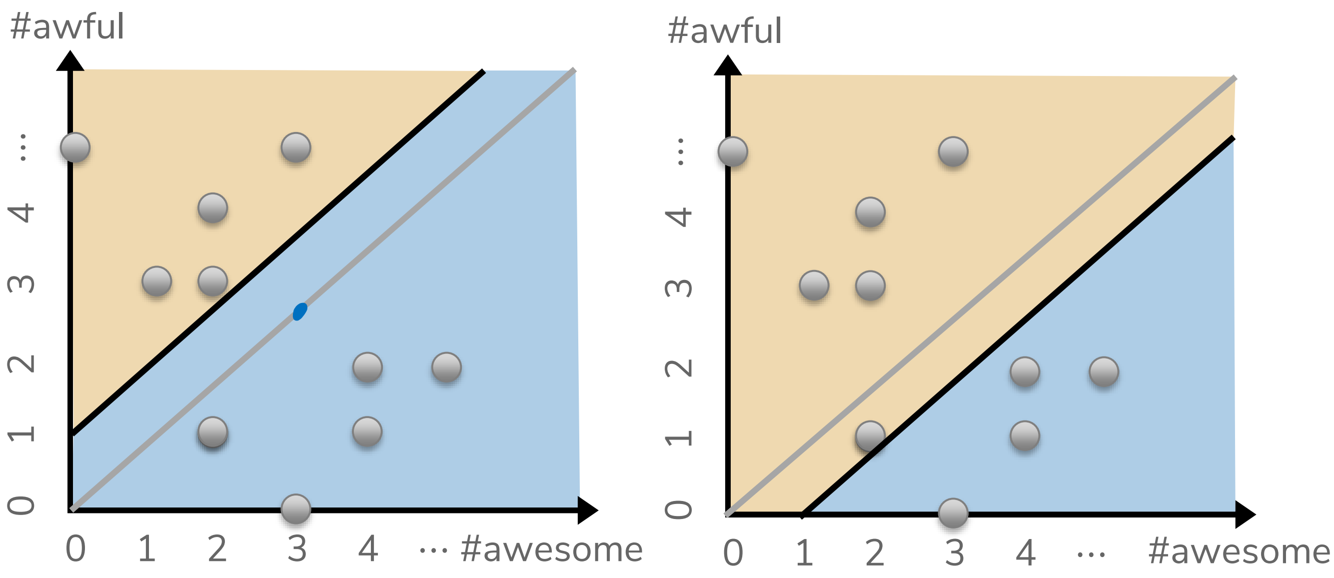
2 min

pollev.com/cs416

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

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

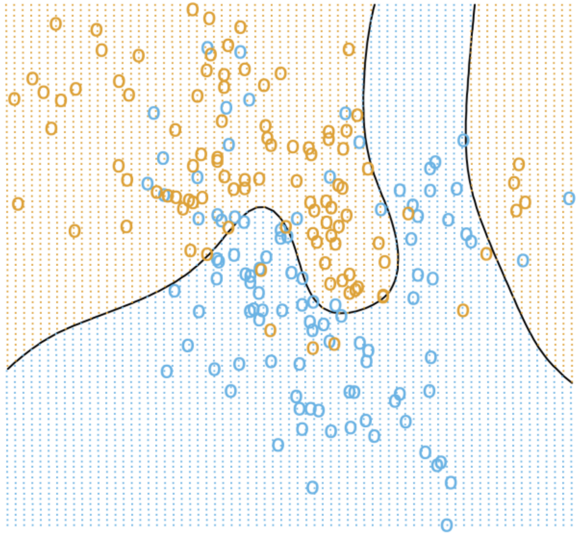
$= 1 + 1 \cdot 3 - 1.5 \cdot 2 = 1$



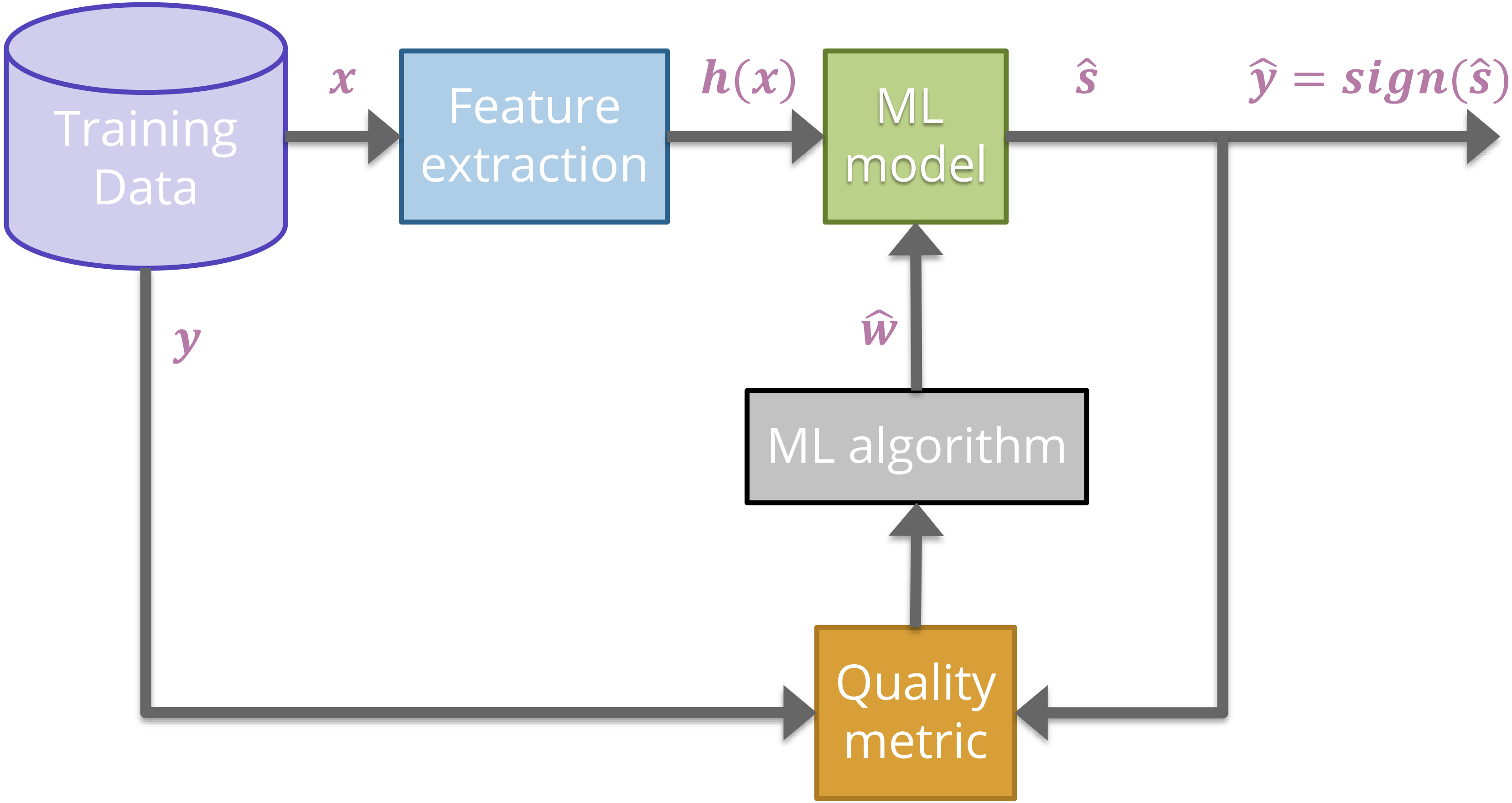
Complex Decision Boundaries?

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

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



Evaluating Classifiers


Training Data
x
Feature extraction
$h(x)$
ML model
$\hat{s}$
$\hat{y} = sign(\hat{s})$
y
$\hat{w}$
ML algorithm
Quality metric

Classification Error

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

Classification Error

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

Classification Accuracy

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

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?

Brain Break

3:18

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
- 60 positive
- 40 negative

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

$$\text{Accuracy} = \frac{50+35}{100} = \frac{50+35}{50+10+5+35} = 0.85$$

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.

Errors and Fairness

We mentioned on the first day how ML is being used in many contexts that impact crucial aspects of our lives.

Models making errors is a given, what we do about that is a choice:

- Are the errors consequential enough that we shouldn't use a model in the first place?
- Do different demographic groups experience errors at different rates?
  - If so, we would hopefully want to avoid that model!

Will talk more about how to define whether or a not a model is fair / discriminatory in a later lecture! Will use these notions of error as a starting point!

Binary Classification Measures

Notation

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

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

$$\frac{C_{TP}}{N_P}$$

Precision

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

F1-Score

$$2\,\frac{Precision \cdot Recall}{Precison + Recall}$$

See more!

Multiclass Confusion Matrix

Consider predicting (*Healthy, Cold, Flu*)

Predicted Label

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

Poll Everywhere

Think

1 min

pollev.com/cs416

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

True Label	Pupper	Doggo	Boofer	
Pupper	2	27	4	33
Doggo	4	25	4	33
Boofer	1	30	2	33

Poll Everywhere

Think

2 min

pollev.com/cs416

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

True Label	Pupper	Doggo	Boofer
Pupper 33	2	27	4
Dogoo 33	4	25	4
Boofer 33	1	30	2

Doggo

No imbalance

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

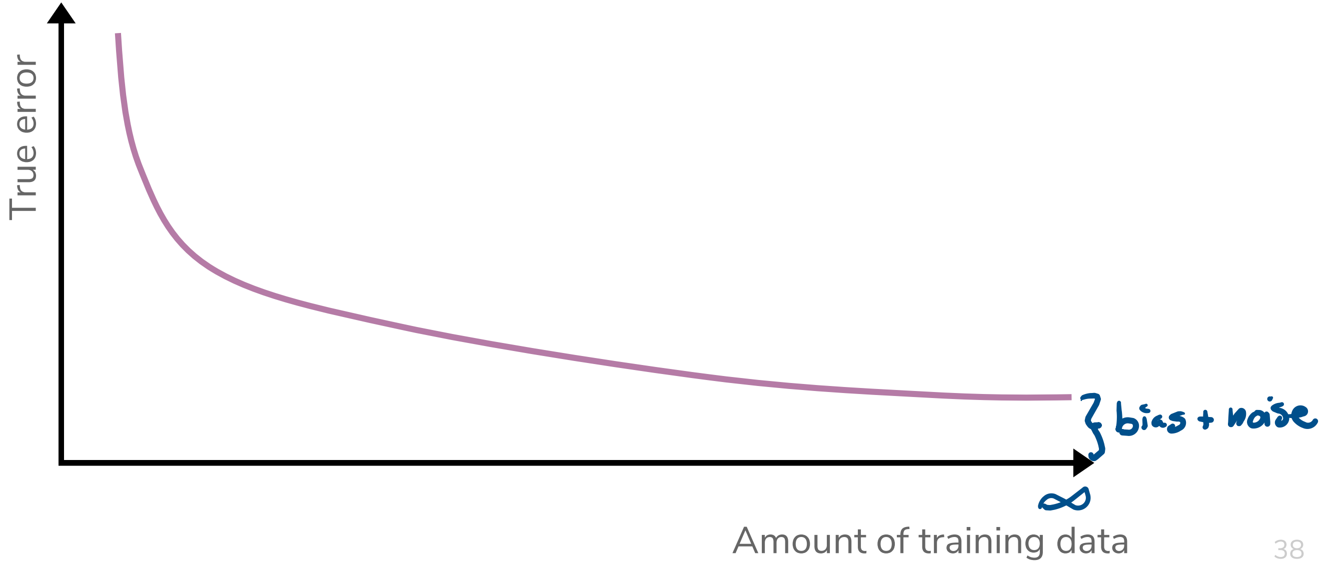
Optional

Vc-dimension

Learning Curve

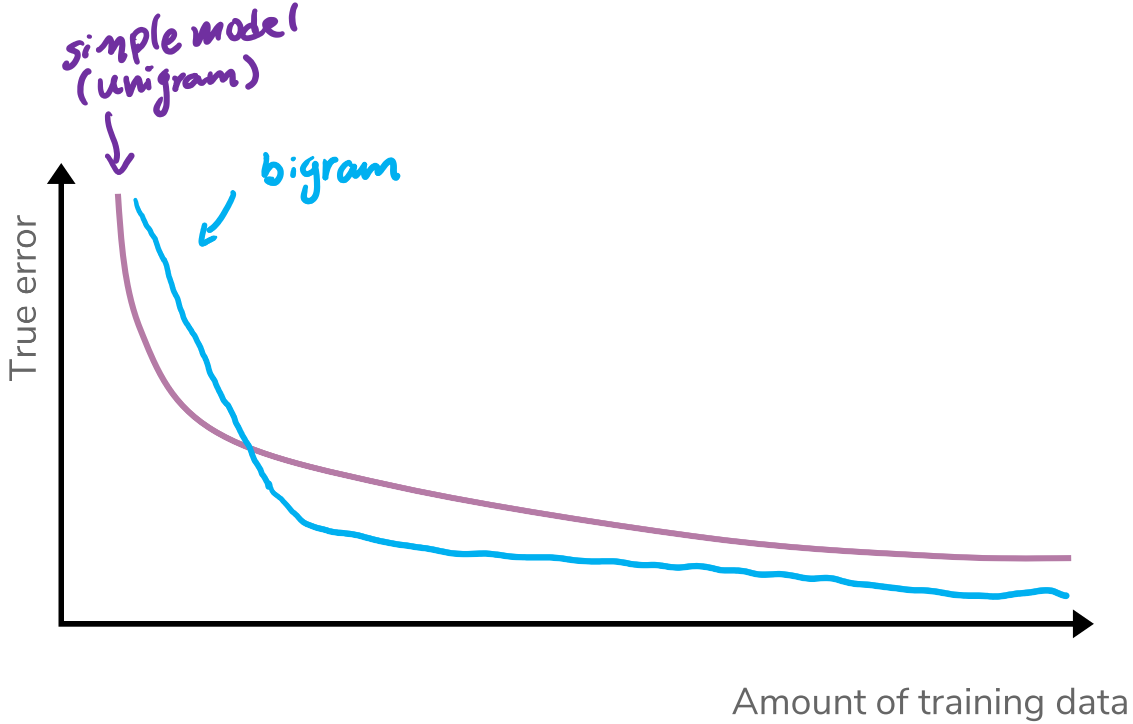
How does the true error of a model relate to the amount of training data we give it?

- Hint: We've seen this picture before



Learning Curve

What if we use a more complex model?



Brain Break

Threshold Model

Change Threshold

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

Always predict neg. $(\alpha = \infty)$

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

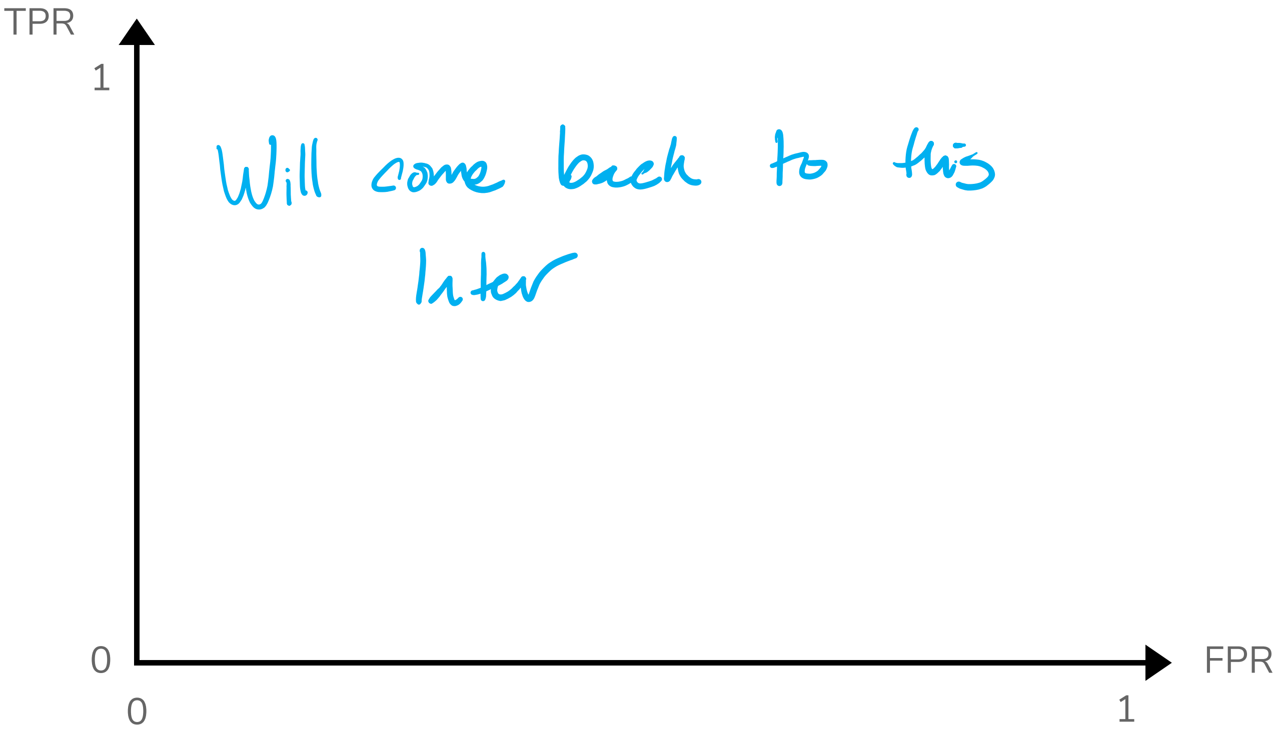
Always predict pos $(\alpha = -\infty)$

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

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

~~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~~