Classification and Logistic Regression

Kenneth (Kenny) Joseph

University at Buffalo
Department of Computer Science and Engineering
School of Engineering and Applied Sciences

Announcements

- PA2 due Sunday night
- Quiz 4 is out
- Midterm is March 17th
  - In class, mostly
  - One page handwritten notes, front and back
  - Official Accessibility requests **due by next Tuesday**
- Vote on when to do the review…
- Questions?

PA2 : DTRegressor bug in absolute error

Classification – Supervised Learning with Discrete outcomes

airplane

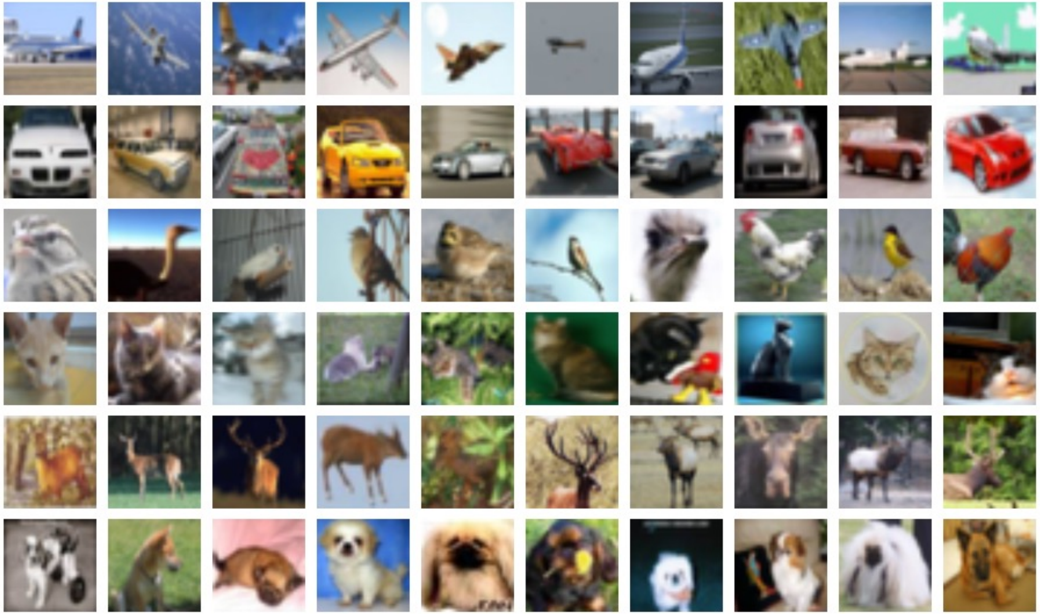
automobile

bird

cat

deer

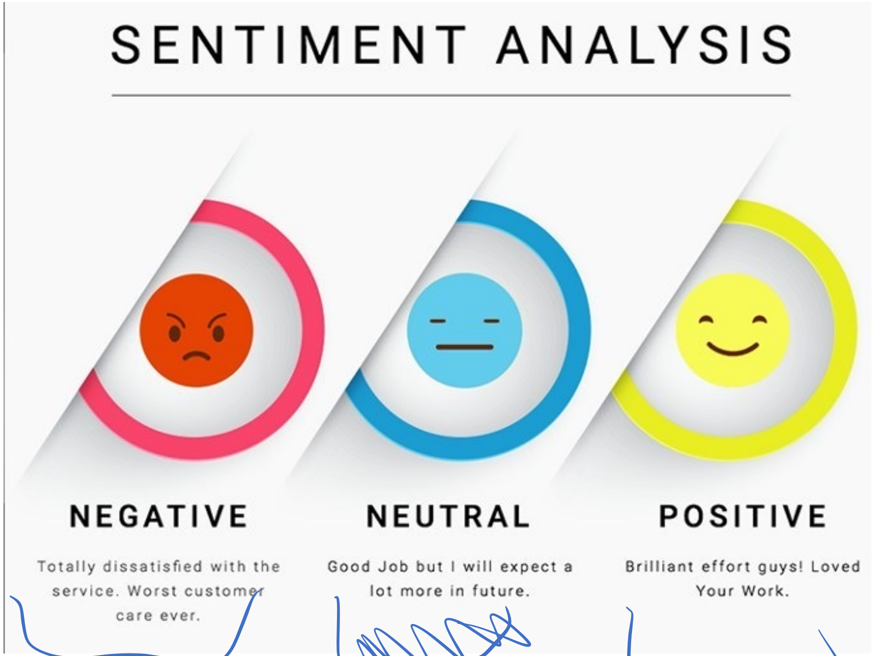
dog



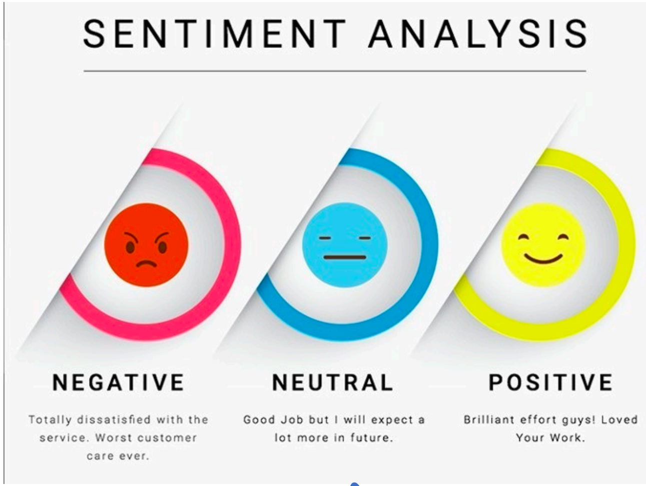
https://www.cs.toronto.edu/~kriz/cifar.html

University at Buffalo
Department of Computer Science and Engineering
School of Engineering and Applied Sciences

Classification – Supervised Learning with Discrete outcomes



https://www.propublica.org/article/machine-bias-risk-assessments-in-criminal-sentencing




```
Text: i'm christian
Sentiment: 0.10000000149011612
```

When I fed it "I'm a Sikh" it said the statement was even more positiv

```
Text: i'm a sikh
Sentiment: 0.30000001192092896
```

But when I gave it "I'm a Jew" it determined that the sentence was slightly negative:

```
Text: i'm a jew
Sentiment: -0.20000000298023224
```

https://www.vice.com/en_us/article/ne3nkb/google-artificial-intelligence-bias-apology

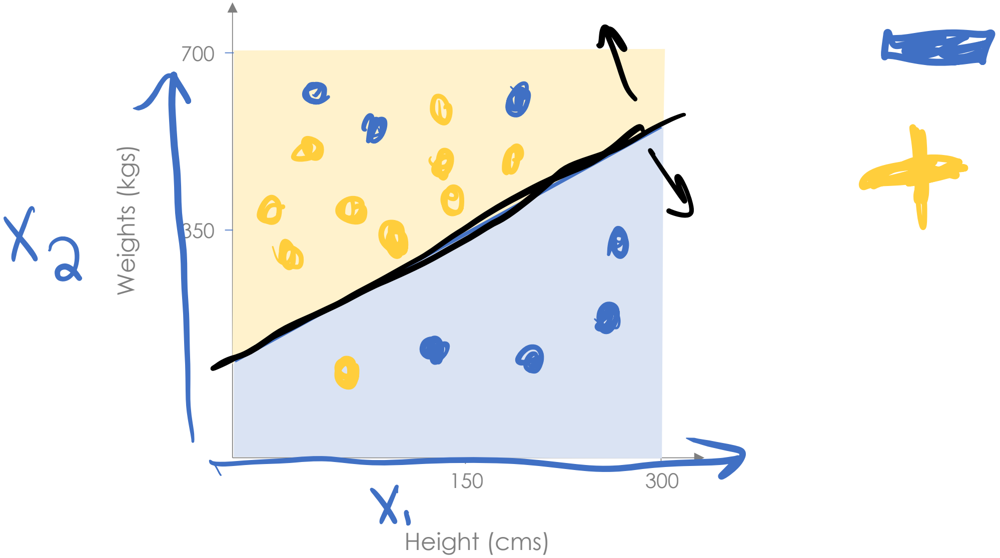
Now that you have the lay of the land with ML and what it does (at least at a high level), I will begin to emphasize these societal aspects a bit more

What is a classification *model class*?

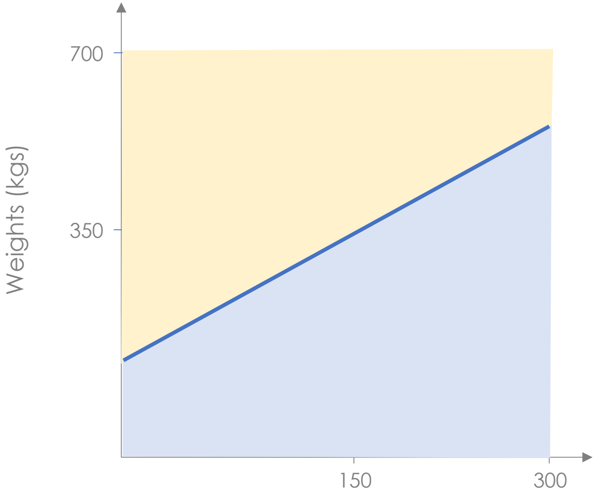
A function *h* that maps h(**x**) -> y when y is a discrete random variable

~~**Quiz:** Examples?~~

A linear model for classification



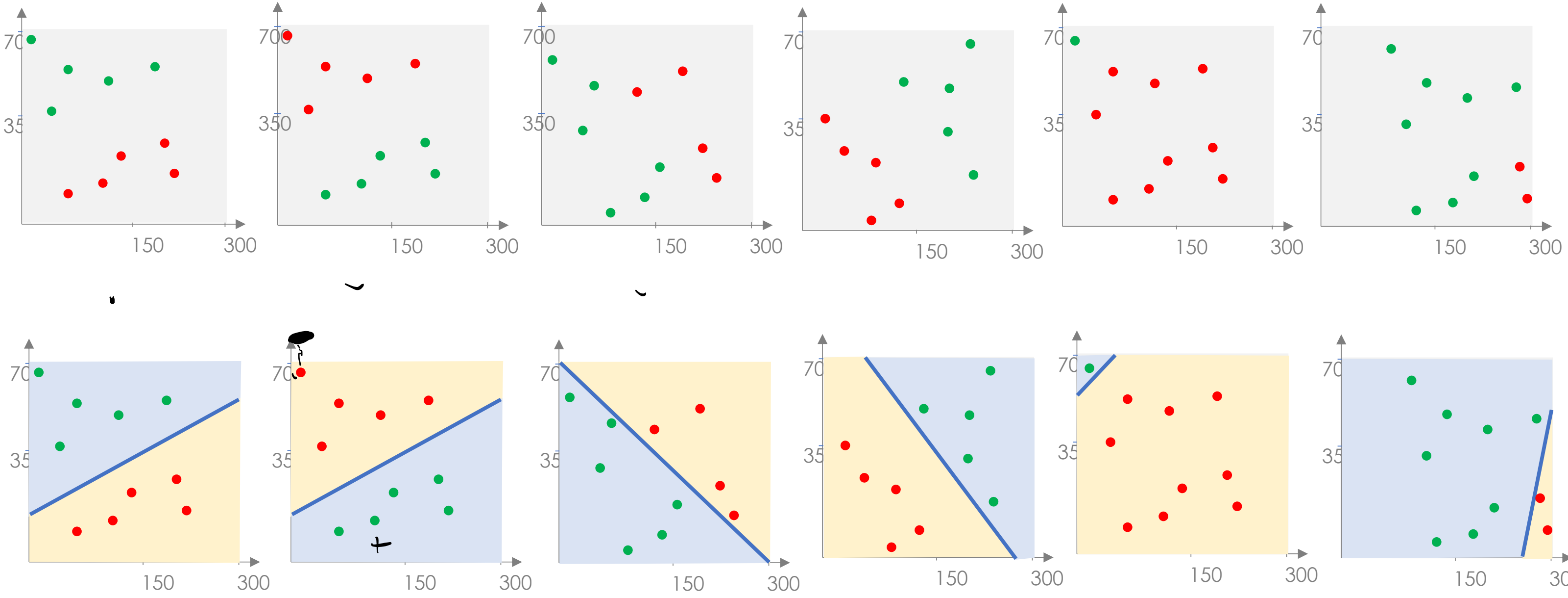
A linear model for classification



Linear models

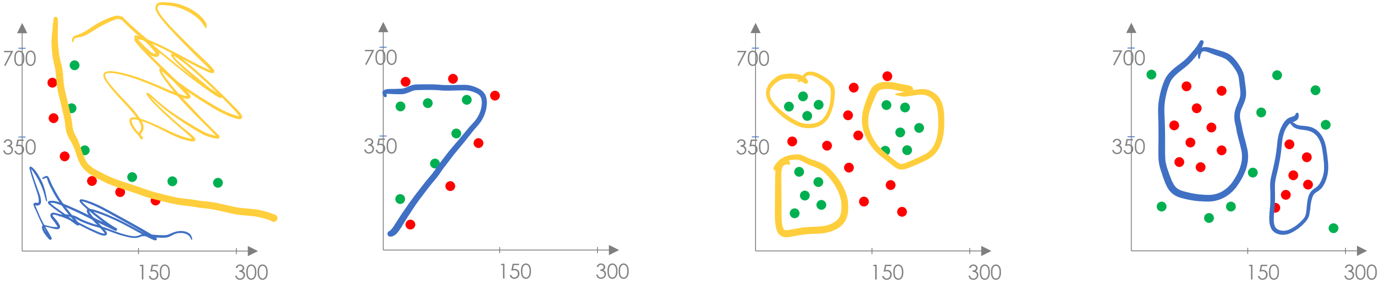
The models we considered above (as we have seen before) are called *linear models* (because you can literally draw a line to present them in 2D). In particular, in the case of two input variables, a linear model is **completely defined** once you specify the line as which of the two sides is the positive side (and the other side automatically becomes the negative side).

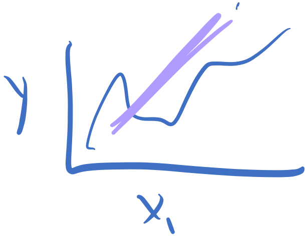
Like regression, linear models are actually fairly effective for classification



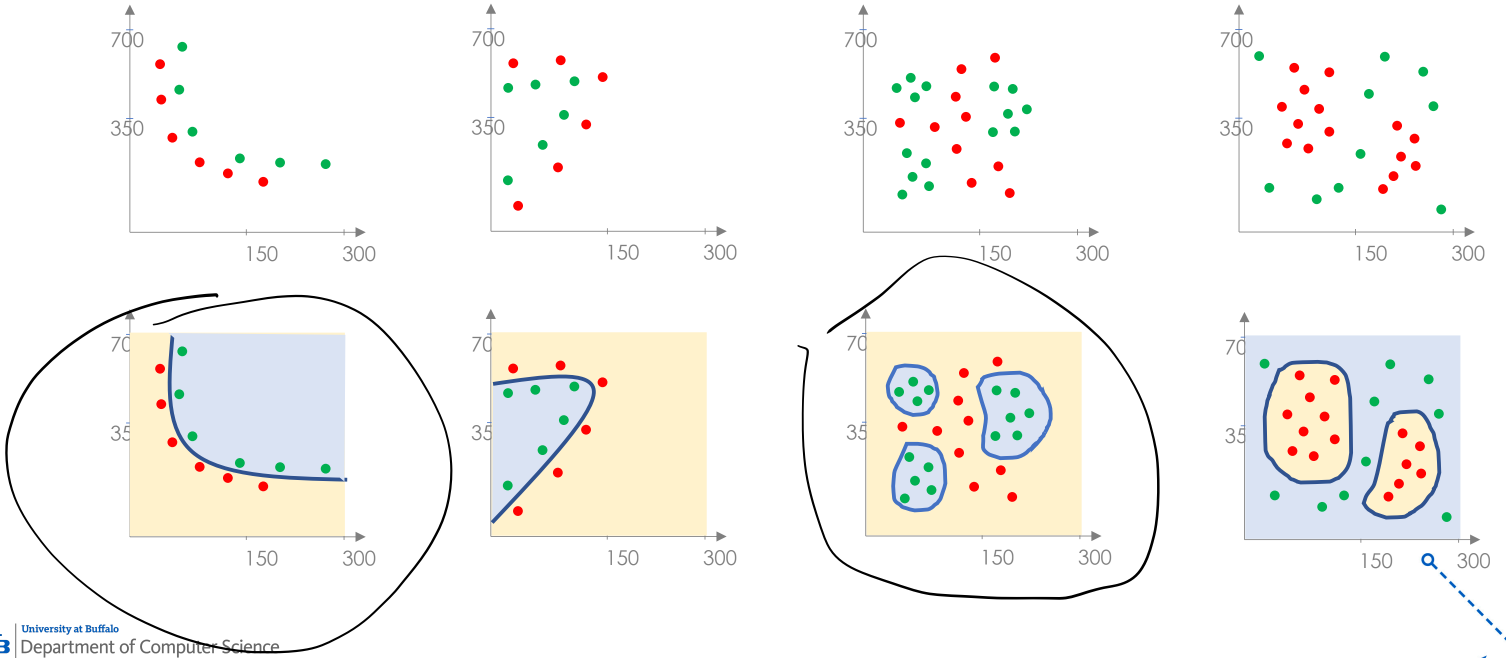
University at Buffalo
Department of Computer Science and Engineering
School of Engineering and Applied Sciences

But obviously not always enough

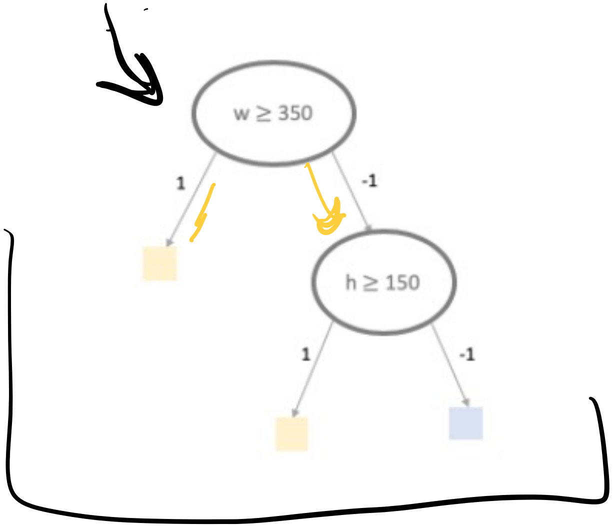


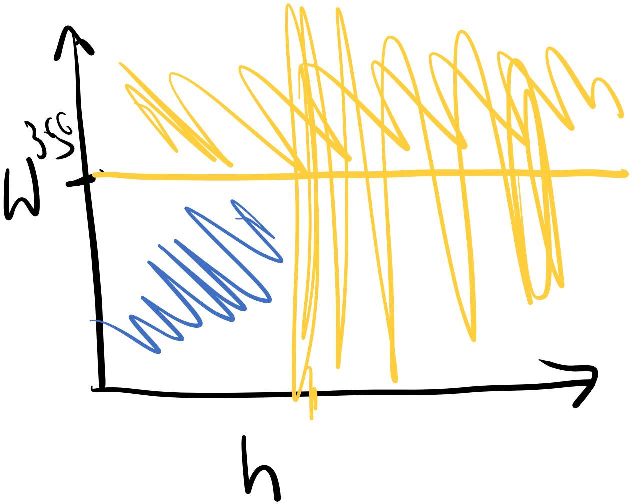


Quiz: Can you think of a way to specify these models?

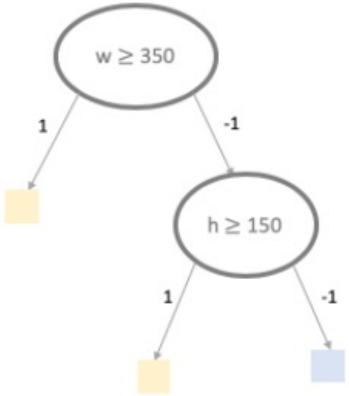


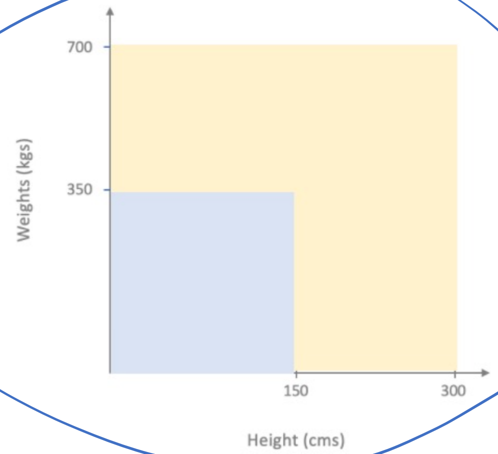
Following on this quiz...





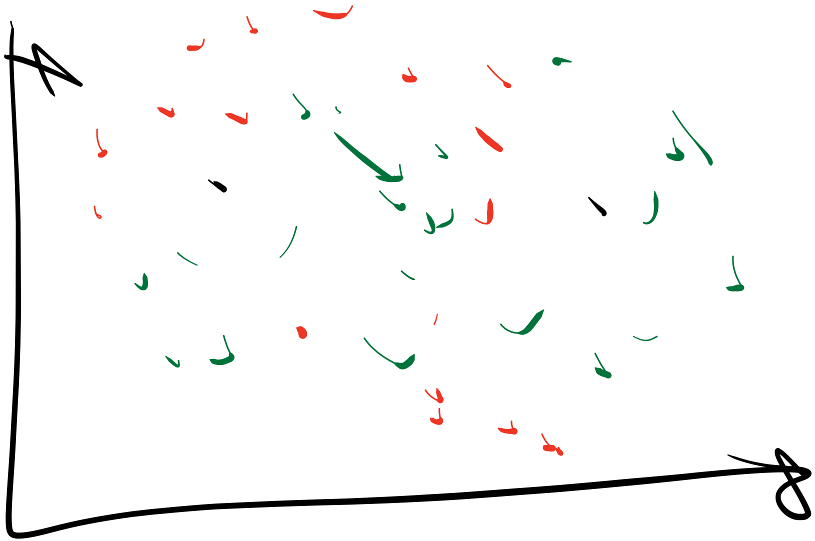
Following on this quiz…





We will talk a lot about these drawings of **decision boundaries**… different models allow for different drawings

Does there always exist non-linear model?



Does there always exist non-linear model?

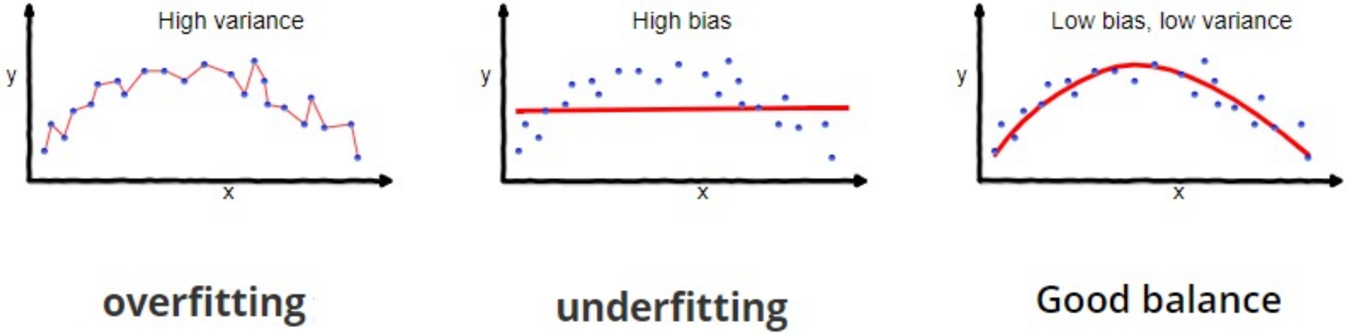
Why Yes?

Convince yourself that given **any** dataset there is **always** a (possibly non-linear) model that fits it perfectly.

`Hint`:Given a dataset, can you use the dataset itself to define the model fits it perfectly? (Do not worry about how complicated the resulting model will be-- you just need to argue that such a model exists.) And do not peek below before you have spent some time thinking about the answer :-)

Quiz: What is likely to come with this added complexity when we find a perfect model on the training data?

The bias-variance tradeoff doesn't just go away ☺

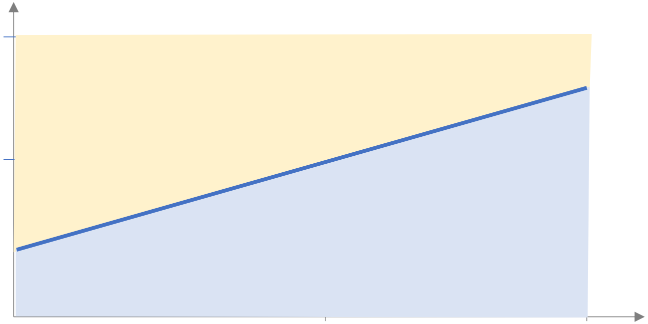


Review

- For binary classification, our model is a curve (function) in the (possibly transformed) feature space.
- **Quiz:**
  - In regression, that curve … $\mathbb{R}$
  - In classification, that curve… label
- In 2 dimensions (and 1!) we can draw the *decision boundaries* in intuitive ways
- Linear models are pretty effective, but as in regression, we can get fancier, and this comes with a cost.
- OK, so, how do we actually get that linear model?

A new task… stance detection

Stance detection: The task of determining whether someone is for or against a particular thing. We'll focus on "stance towards 4/574"

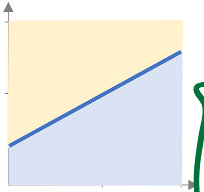


Adapted from: https://courses.cs.washington.edu/courses/cse416/21sp/

A new task… stance detection

Stance detection: The task of determining whether someone is for or against a particular thing. We'll focus on "stance towards 4/574"

This class is garbage. The professor makes bad jokes and I can't read his handwriting



Arguably the greatest moments in human history have come when Kenny takes the floor for 4/574 each week

Adapted from: https://courses.cs.washington.edu/courses/cse416/21sp/

Stance detection in the real world

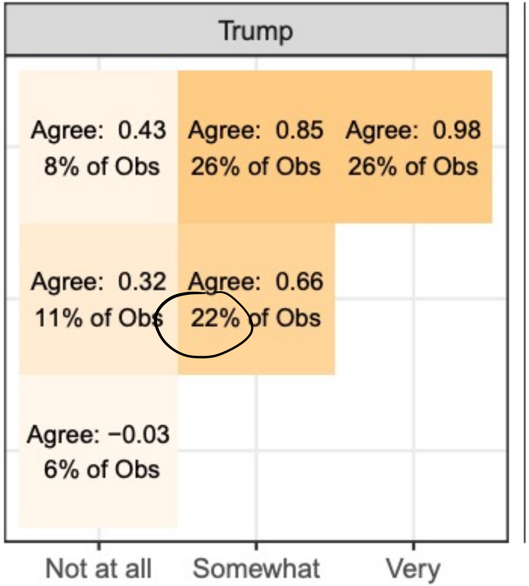
The prof's jokes are bad and he can't make a quiz without an error to save his life but I occasionally learn some stuff

Stance detection in the real world

Staying at home with kids is more stressful than going to work, according to [a new study].

… pro or anti-lockdown measures?

Next week: Annotation practice, measures of agreement



Back to the example…

This class is garbage. The professor makes bad jokes and I can't read his handwriting

- How would you approach the task of stance detection? Specifically…
  - What would your **features** be?
  - How would you make decisions based on those features?
  - What **loss function** would you use?

Joe Biden won the primary

Adapted from: https://courses.cs.washington.edu/courses/cse416/21sp/

Approach 1: Bag of words + Linear threshold classifier

1. Convert each course evaluation statement into a "bag of words" representation
2. Fix a weight for each word in terms of how having it in a sentence implies a positive/pro or negative/anti stance
3. Sum up the weights for all of the words to get a ***score***
4. If the ***score*** is > 0, predict "pro-5/474", otherwise, predict "anti"

Adapted from: https://courses.cs.washington.edu/courses/cse416/21sp/

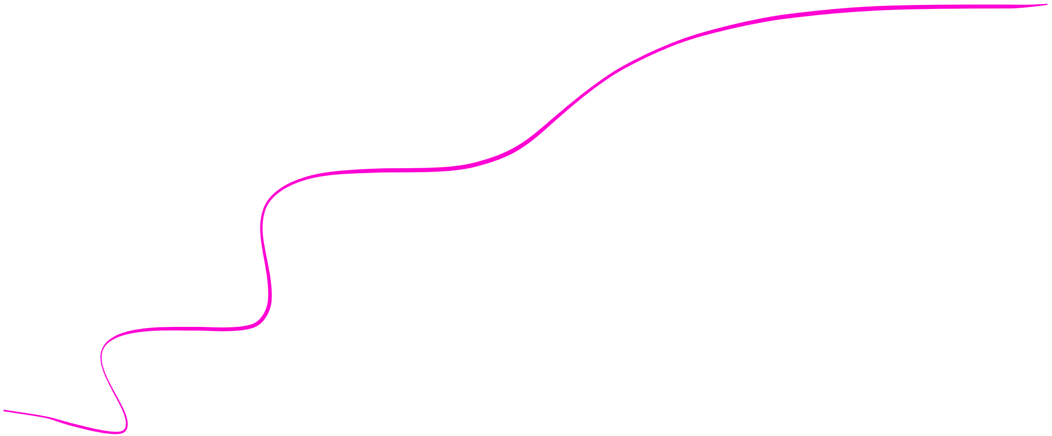
Approach 1, Step 1

This class is garbage. The professor makes bad jokes and I can't read his handwriting e_1

The prof's jokes are bad and he can't make a quiz without an error to save his life but I occasionally learn some stuff e_2

Arguably the greatest moments in human history have come when Kenny takes the floor for 4/574 each week e_3

		This	class	jokes	floor	the
x_1	e_1	1	1	0	0	0
x_2	e_2	0	0	1	0	0
x_3	e_3	0	0	0	1	2



Approach 1, Step 2

- How might we get these values in the easiest possible way?
- … later… how we can learn them from data

Word	Weight (*w*)
garbage	-5.0
bad	-3.0
can't	-0.5
bad	-3.0
error	-2.1
learn	3.0
greatest	4.0
Arguably, human, professor, handwriting, …	0.0

Adapted from: https://courses.cs.washington.edu/courses/cse416/21sp/

Approach 1, Step 2

This class is garbage. The professor makes bad jokes and I can't read his handwriting

The prof's jokes are bad and he can't make a quiz without an error to save his life but I occasionally learn some stuff

Arguably the greatest moments in human history have come when Kenny takes the floor for 4/574 each week

Word	Weight (*w*)
garbage	-5.0
bad	-3.0
can't	-0.5
bad	-3.0
error	-2.1
learn	3.0
greatest	4.0
Arguably, human, professor, handwriting, …	0.0

Adapted from: https://courses.cs.washington.edu/courses/cse416/21sp/

Approach 1, Step 3

$(w^T x_1 = -2.1$

x_{1c}

The prof's jokes are bad and he can't make a quiz without an error to save his life but I occasionally learn some stuff

The	prof's	...	stuff
0	1	...	1

x_1

$The \cdot w_{the}$ (0) + $prof's_{(1)} \cdot w_{profs}$ (0)

+ $bad_{(1)} \cdot -3 = \ldots$

$-3 + -2.1 + 3 = \boxed{-2.1}$

Score = -2.1

Word	Weight (w)
garbage	-5.0
bad	-3.0
can't	-0.5
bad	-3.0
error	-2.1
learn	3.0
greatest	4.0
Arguably, human, professor, handwriting, …	0.0

Adapted from: https://courses.cs.washington.edu/courses/cse416/21sp/

Approach 1, Step 4

This class is garbage. The professor makes bad jokes and I can't read his handwriting

$w^T x_1$

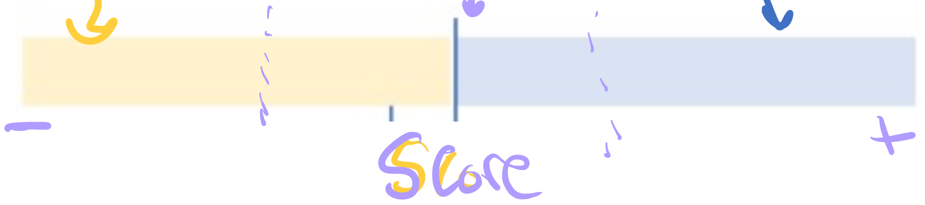
score(x_1)

The prof's jokes are bad and he can't make a quiz without an error to save his life but I occasionally learn some stuff

$w^T x_2$ $=$ score(x_2)

Arguably the greatest moments in human history have come when Kenny takes the floor for 4/574 each week

$w^T x_3$ $=$ score(x_3)

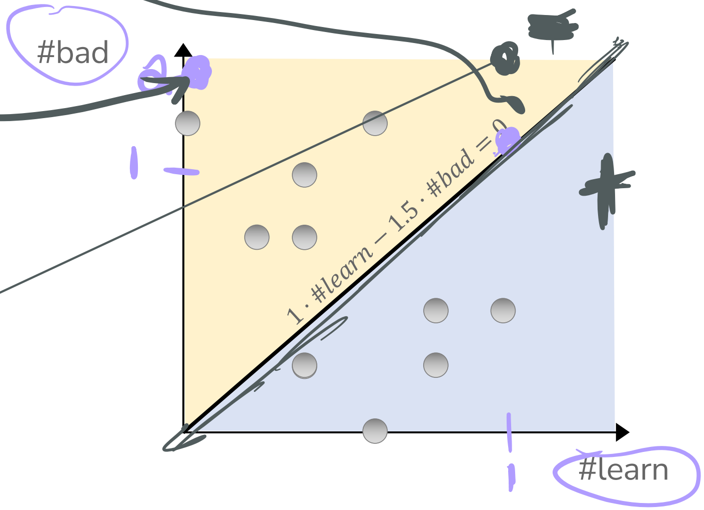


Adapted from: https://courses.cs.washington.edu/courses/cse416/21sp/

Geometric View - Threshold

The prof's jokes are bad and he can't make a quiz to save his life but I occasionally learn some stuff

Word	Weight (w)
bad	-1.5
learn	1



Jokes are bad, lectures are bad

Jokes are bad, lectures are bad, I learn absolutely nothing

Adapted from: https://courses.cs.washington.edu/courses/cse416/21sp/

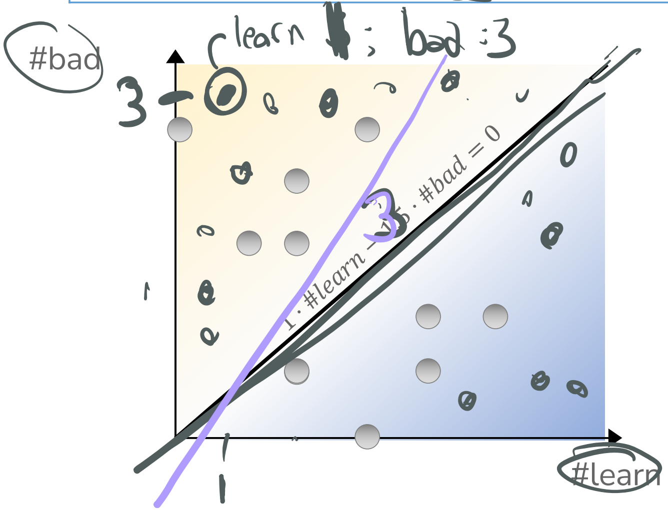
Geometric View - Score

Word	Weight (*w*)	
bad	-1.5	w_2
learn	1	w_1

The prof's jokes are bad and he can't make a quiz to save his life but I occasionally learn some stuff

Jokes are bad, lectures are bad

Jokes are bad, lectures are bad, I learn absolutely nothing



$w_1 \cdot \#learn -$

$w_2 \cdot \#bad$

Adapted from: https://courses.cs.washington.edu/courses/cse416/21sp/

Cool!

- We have built our first classifier!
- **Quiz:** Did this classifier use (training) data at all?
- **How could it have used data to inform the model?**

X and Y, then

bag of word of evaluations

I can learn w, $h(x) = w^T x$

…Put another way, how to learn word weights?

CIML, pg. 43

...Put another way, how to learn word weights?

An online algorithm to learn weights for the words...

The *perceptron* algorithm.

An early, well-known approach!

IMO, can complicate understanding at this point in the class

Algorithm 5 PerceptronTrain($\mathbf{D}$, *MaxIter*)

```
w_d ← 0, for all d = 1 . . . D                      // initialize weights
b ← 0                                               // initialize bias
for iter = 1 . . . MaxIter do
   for all (x,y) ∈ D do
      a ← Σ_{d=1}^{D} w_d x_d + b                   // compute activation for this example
      if ya ≤ 0 then
         w_d ← w_d + y x_d, for all d = 1 . . . D   // update weights
         b ← b + y                                  // update bias
      end if
   end for
end for
return w_0, w_1, . . . , w_D, b
```

Algorithm 6 PerceptronTest($w_0, w_1, \ldots, w_D, b, \hat{x}$)

```
a ← Σ_{d=1}^{D} w_d x̂_d + b                         // compute activation for the test example
return SIGN(a)
```

CIML, pg. 43

Another idea

- Take our basic tools!
- Specify a model class (we already have one!)

$h(x) = w^T x$

- Define a loss function … **what?**

$(score(x) - y)^2$

$$\underbrace{\mathbb{I}(\hat{y}, y) \rightarrow \begin{cases} 1 & \text{wrong} \\ 0 & \text{right} \end{cases}}_{\text{0/1 Loss}}$$

y: -1, or $+1$

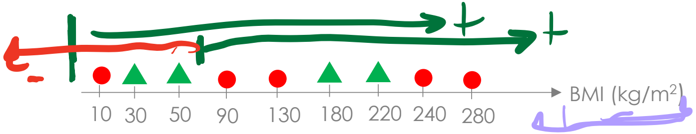
$\hat{y}$: -1, or $+1$

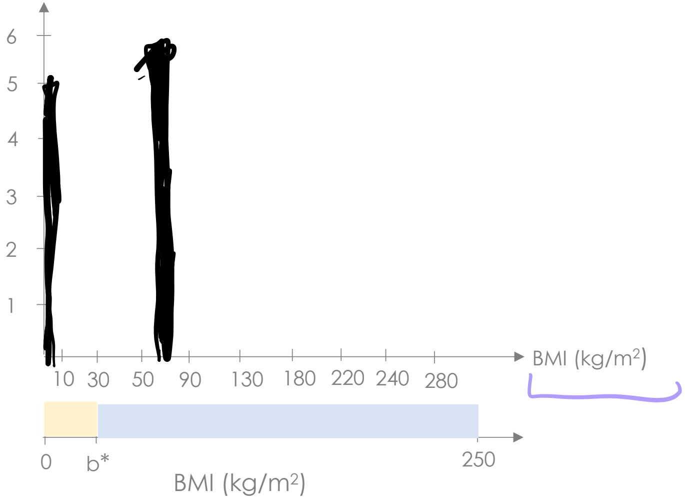
$$\begin{cases} -1 & score(x) < 0 \\ +1 & \text{otherwise} \end{cases}$$

- Optimize (how?)

GD

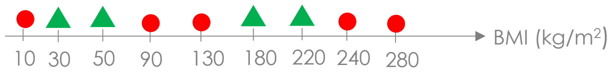
Trying to optimize 0/1 Loss in 1 Dimension

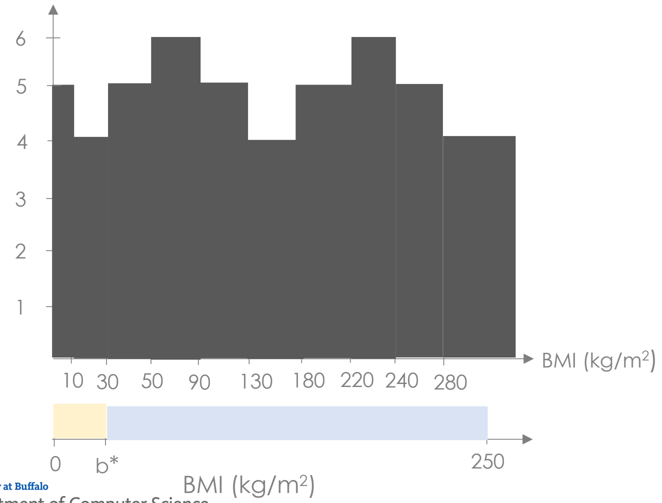


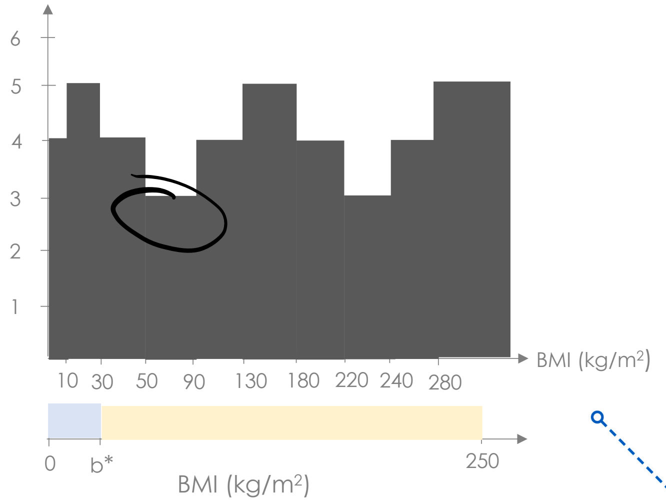


University at Buffalo
Department of Computer Science and Engineering
School of Engineering and Applied Sciences

Trying to optimize 0/1 Loss in 1 Dimension

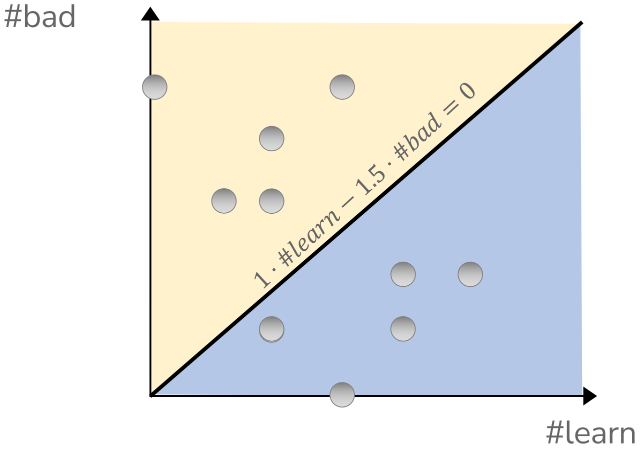


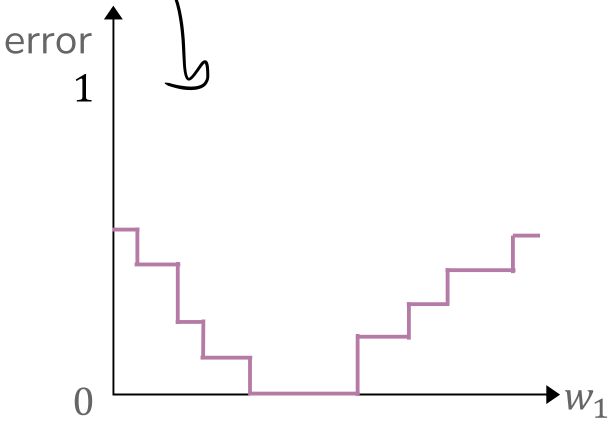




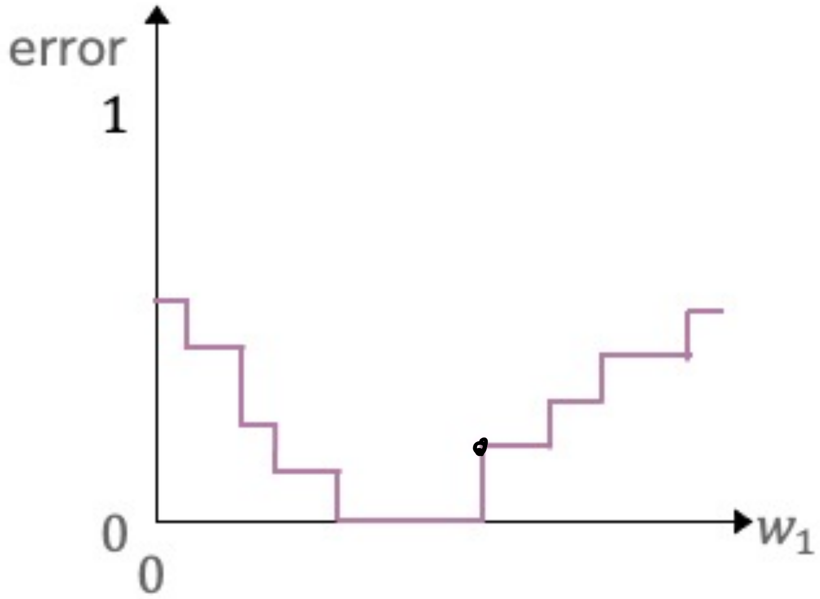
Assume w_2 is fixed, and we want to min. loss w.r.t. w_2

Try this

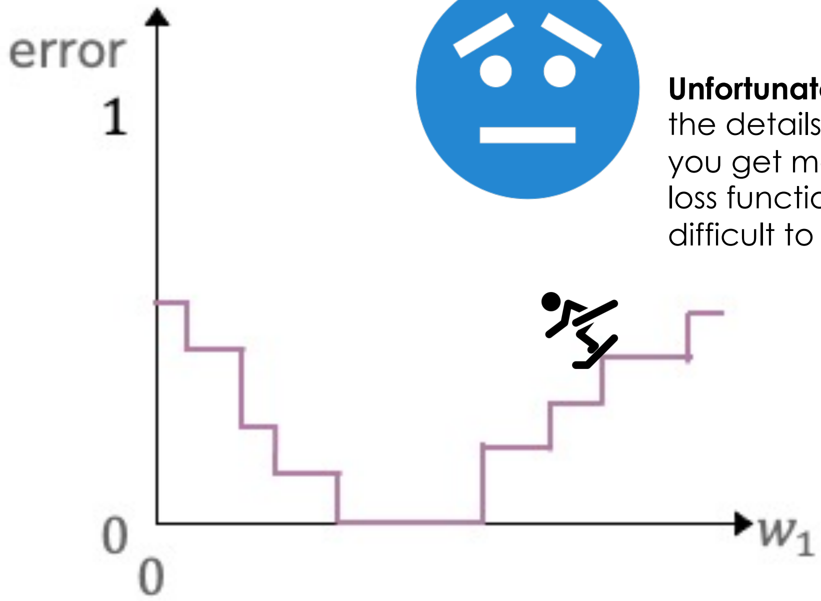




Can we just run gradient descent on this?

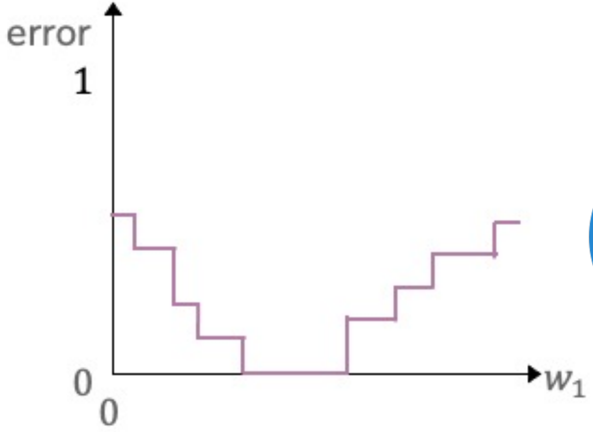


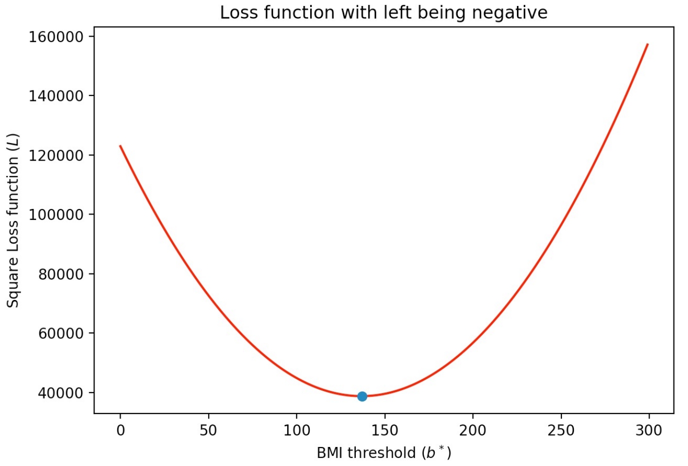
Can we just run gradient descent on this?



Unfortunately not. We will not cover the details on why, but essentially, as you get more features, these spiky loss functions become extremely difficult to optimize.

What to do? Optimization view...





Change the loss function to something we can more easily optimize!

... which is...?

Adapted from: https://courses.cs.washington.edu/courses/cse416/21sp/

Approach 2: Bag of words + Linear classifier, Optimization view

1. Convert each course evaluation statement into a "bag of words" representation
2. Specify model class: $h(x) = w^T x$
3. Define loss function: $(w^T x - y)^2$
4. Optimize loss fn.: GD
5. For new test point, compute $h(x) = w^T x$
6. If h(x) is > 0, predict "pro", otherwise, predict "anti"

   1 -1

Problem: How to interpret predictions? What does h(x)=10 mean?

bound $h(x)$ $[0,1]$

What to do? Probabilistic view…

Model p(y | **x**) !

P(y = + | **This class is garbage. The professor makes bad jokes and I can't read his handwriting**) = ? .01

P(y = + | **The class is fine. I wish he would stop making up course evaluations though.**) = ? .55

Approach 3: Logistic Regression

1. Convert each course evaluation statement into a "bag of words" representation
2. Specify form of p(y | **x**)
3. Write down (log) likelihood function
4. Maximize log-likelihood fn.
5. Use trained model to estimate p(y=+ | x)
6. If p(+ | x) > .5, predict "pro-5/474", otherwise, predict "anti"

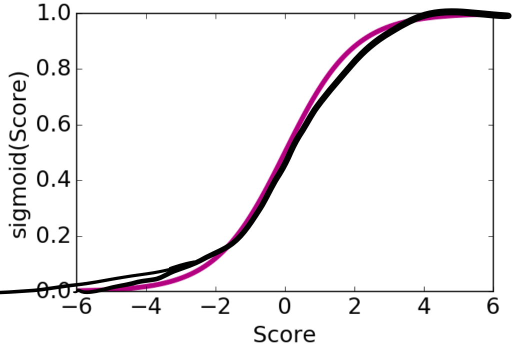
Question: How to specify p(y|x)?

Logistic Function

Use a function that takes numbers arbitrarily large/small and maps them between 0 and 1.

$$sigmoid(Score(x)) = \frac{1}{1 + e^{-Score(x)}}$$

$Score(x)$	$sigmoid(Score(x))$
$-\infty$	$\frac{1}{1+e^{+\infty}} = 0$
-2	$\approx .12$
0	$\frac{1}{1+e^{0}} = \frac{1}{1+1} = .5$
2	$\approx .7$
∞	$\frac{1}{1+e^{-\infty}} = 1$



Directly from: https://courses.cs.washington.edu/courses/cse416/21sp/

Interpreting Score

$$Score(x_i) = w^T h(x_i)$$

$-\infty$ $\hat{y}_i = -1$ 0 $\hat{y}_i = +1$ ∞

$-\infty$	0	∞
Very sure $\hat{y}_i = -1$	Not sure if $\hat{y}_i = -1\ or\ \hat{y}_i = +1$	Very sure $\hat{y}_i = +1$
$\hat{P}(y_i = +1 \mid x_i) = 0$	$\hat{P}(y_i = +1 \mid x_i) = 0.5$	$\hat{P}(y_i = +1 \mid x_i) = 1$

0 $\hat{P}(y = +1 \mid x)$ 1

Directly from: https://courses.cs.washington.edu/courses/cse416/21sp/

Approach 3: Logistic Regression

1. Convert each course evaluation statement into a “bag of words” representation
2. Specify form of $\underbrace{P(y_i = +1 | x_i, w) = sigmoid\big(score(x)\big) = \frac{1}{1+e^{-w^T x_i}}}$
3. Write down (log) likelihood function
4. Maximize log-likelihood fn.
5. Use trained model to estimate p(+ | x)
6. If p(+ | x) > .5, predict “pro-5/474”, otherwise, predict “anti”

Approach 3: Logistic Regression

1. Convert each course evaluation statement into a “bag of words” representation
2. Specify form of $P(y_i = +1|x_i, w) = \frac{1}{1+e^{-w^T x_i}}$
3. **Write down (log) likelihood function**

Approach 3: Logistic Regression

1. Convert each course evaluation statement into a "bag of words" representation
2. Specify form of $P(y_i = +1 | x_i, w) = \frac{1}{1+e^{-w^T x_i}}$
3. Write down (log) likelihood function
4. **Maximize log-likelihood fn.**

Approach 3: Logistic Regression

1. Convert each course evaluation statement into a "bag of words" representation
2. Specify form of $P(y_i = +1|x_i, w) = \frac{1}{1+e^{-w^T x_i}}$
3. Write down (log) likelihood function
4. **Maximize log-likelihood fn.**
   - No closed form solution!
   - Have to use gradient ascent/descent
   - Can do slightly better by using the second derivative as well to guide the movement through the space…
   - This is the **Newton-Raphson method**

Approach 3: Logistic Regression

1. Convert each course evaluation statement into a "bag of words" representation
2. Specify form of $P(y_i = +1 | x_i, w) = \frac{1}{1+e^{-w^T x_i}}$
3. Write down (log) likelihood function
4. Maximize log-likelihood fn.
5. Use trained model to estimate p(+ | x)
6. If p(+ | x) > .5, predict "pro-5/474", otherwise, predict "anti"

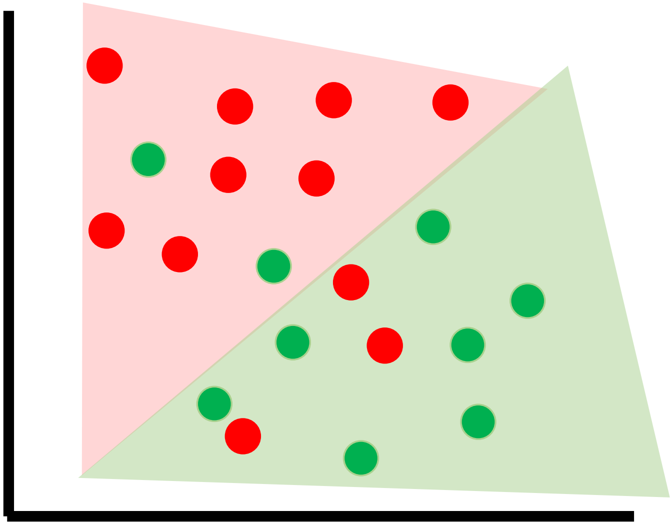
Some details we'll get to

- Do we have to use .5 as the threshold for classification?
  - No, and sometimes it's actually not a good idea
- Can we use logistic regression to learn non-linear decision boundaries?
  - Yes! **How?**
- Can we regularize logistic regression?
  - Yes! **How?**
- How do we get labels for data?
  - (Kind of discussed) Annotation! Lecture next week, PA3!
- Can we go beyond "bag of words"?
  - Yes! **Ideas?** … lectures post Spring break!
- How do we evaluate classifiers?
  - **A bit now, a bit later**

OK!

What questions do you have?!

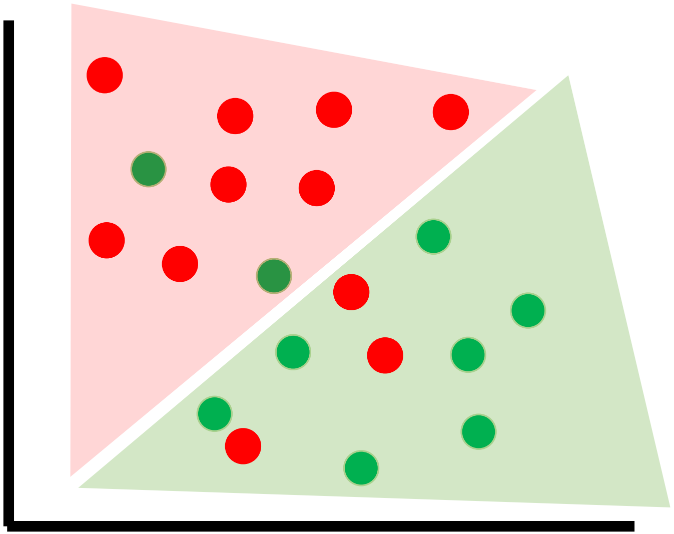
Evaluating classification models

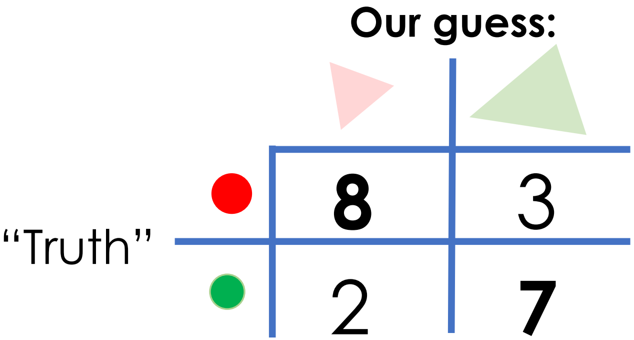


Our guess:

"Truth"		
	8	3
	2	7

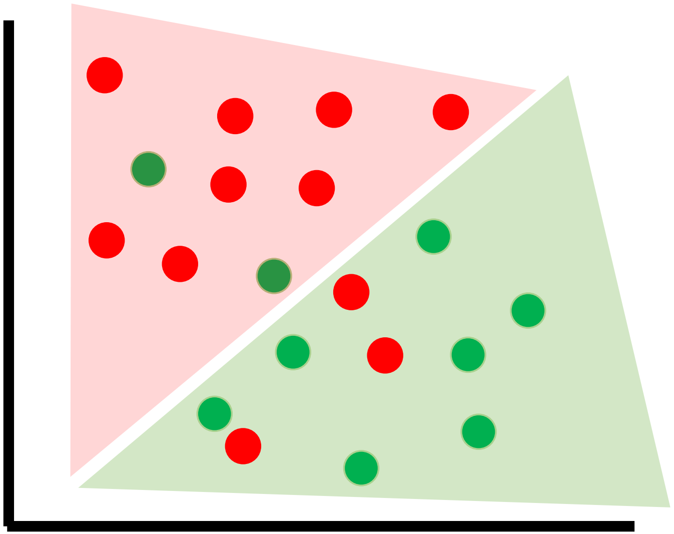
Accuracy – how many did we get correct?





Accuracy =
(8 + 7) / (8 + 7 + 2 + 3)
= .75

Precision - Of + guesses, how many actually +s?



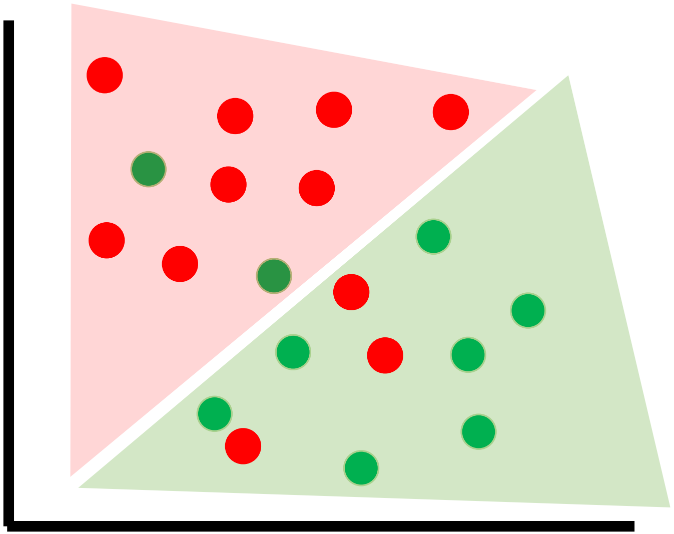
Our guess:

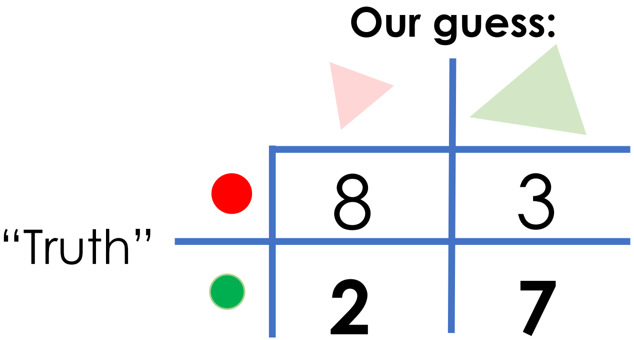
"Truth"

	8	**3**
	2	**7**

Precision =
7 / (7 + 3) = .7

Recall - Of actual +, how many do we guess?





Precision =
7 / (7 + 2) = .78

Evaluation Review

- Different metrics for different things
- Other performance metrics:
  - F1 Score
  - …
- Other considerations
  - Class imbalance (accuracy bad)
  - …

What is missing from these evaluations?

Other questions we might ask

- Which one had higher recall?
- Which one had higher precision?
- Was that the same for both groups?