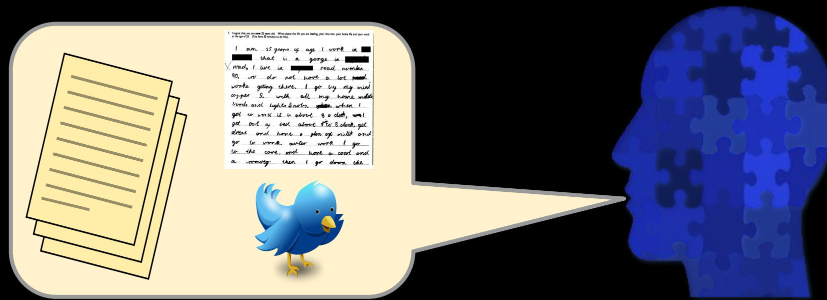
Logistic Regression and POS Tagging

CSE392 - Spring 2019
Special Topic in CS

Task



- Parts-of-Speech Tagging

how?

- Machine learning:
  - Logistic regression

Parts-of-Speech

Open Class:

Nouns, Verbs, Adjectives, Adverbs

Function words:

Determiners, conjunctions, pronouns, prepositions

Parts-of-Speech: The Penn Treebank Tagset

Table 2
The Penn Treebank POS tagset.

1.	CC	Coordinating conjunction	25.	TO	*to*
2.	CD	Cardinal number	26.	UH	Interjection
3.	DT	Determiner	27.	VB	Verb, base form
4.	EX	Existential *there*	28.	VBD	Verb, past tense
5.	FW	Foreign word	29.	VBG	Verb, gerund/present participle
6.	IN	Preposition/subordinating conjunction	30.	VBN	Verb, past participle
7.	JJ	Adjective	31.	VBP	Verb, non-3rd ps. sing. present
8.	JJR	Adjective, comparative	32.	VBZ	Verb, 3rd ps. sing. present
9.	JJS	Adjective, superlative	33.	WDT	*wh*-determiner
10.	LS	List item marker	34.	WP	*wh*-pronoun
11.	MD	Modal	35.	WP$	Possessive *wh*-pronoun
12.	NN	Noun, singular or mass	36.	WRB	*wh*-adverb
13.	NNS	Noun, plural	37.	#	Pound sign
14.	NNP	Proper noun, singular	38.	$	Dollar sign
15.	NNPS	Proper noun, plural	39.	.	Sentence-final punctuation
16.	PDT	Predeterminer	40.	,	Comma
17.	POS	Possessive ending	41.	:	Colon, semi-colon
18.	PRP	Personal pronoun	42.	(	Left bracket character
19.	PP$	Possessive pronoun	43.	)	Right bracket character
20.	RB	Adverb	44.	"	Straight double quote
21.	RBR	Adverb, comparative	45.	‘	Left open single quote
22.	RBS	Adverb, superlative	46.	“	Left open double quote
23.	RP	Particle	47.	’	Right close single quote
24.	SYM	Symbol (mathematical or scientific)	48.	”	Right close double quote

Parts-of-Speech: Social Media Tagset

(*Gimpel et al., 2010*)

Tag	Description	Examples	%
Nominal, Nominal + Verbal			
N	common noun (NN, NNS)	books someone	13.7
O	pronoun (personal/WH; not possessive; PRP, WP)	it you u meeee	6.8
S	nominal + possessive	books' someone's	0.1
^	proper noun (NNP, NNPS)	lebron usa iPad	6.4
Z	proper noun + possessive	America's	0.2
L	nominal + verbal	he's book'll iono (= *I don't know*)	1.6
M	proper noun + verbal	Mark'll	0.0

Other open-class words			
V	verb incl. copula, auxiliaries (V*, MD)	might gonna ought couldn't is eats	15.1
A	adjective (J*)	good fav lil	5.1
R	adverb (R*, WRB)	2 (i.e., *too*)	4.6
!	interjection (UH)	lol haha FTW yea right	2.6

Other closed-class words			
D	determiner (WDT, DT, WP$, PRP$)	the teh its it's	6.5
P	pre- or postposition, or subordinating conjunction (IN, TO)	while to for 2 (i.e., *to*) 4 (i.e., *for*)	8.7
&	coordinating conjunction (CC)	and n & + BUT	1.7
T	verb particle (RP)	out off Up UP	0.6
X	existential *there*, predeterminers (EX, PDT)	both	0.1
Y	**X** + verbal	there's all's	0.0

Twitter/online-specific			
#	hashtag (indicates topic/category for tweet)	#acl	1.0
@	at-mention (indicates another user as a recipient of a tweet)	@BarackObama	4.9
~	discourse marker, indications of continuation of a message across multiple tweets	RT and : in retweet construction RT @user : hello	3.4
U	URL or email address	http://bit.ly/xyz	1.6
E	emoticon	:-) :b (: <3 o_O	1.0

Miscellaneous			
$	numeral (CD)	2010 four 9:30	1.5
,	punctuation (#, $, '', (,), ,, ., :, ``)	!!! .... ?!?	11.6
G	other abbreviations, foreign words, possessive endings, symbols, garbage (FW, POS, SYM, LS)	ily (*I love you*) wby (*what about you*) 's ♫ --> awesome...I'm	1.1

POS Tagging: Applications

- Resolving ambiguity (speech: “lead”)
- Shallow searching: find noun phrases
- Speed up parsing
- Use as feature (or in place of word)

For this course:

- An introduction to language-based classification (logistic regression)
- Understand what modern deep learning methods are dealing with implicitly.

Logistic Regression

Binary classification goal: Build a “model” that can estimate $P(A{=}1|B{=}?)$

i.e. given B, yield (or “predict”) the probability that A=1

Logistic Regression

Binary classification goal: Build a “model” that can estimate P(A=1|B=?)

i.e. given B, yield (or “predict”) the probability that A=1

In machine learning, tradition to use **Y** for the variable being predicted and **X** for the features use to make the prediction.

Logistic Regression

Binary classification goal: Build a “model” that can estimate $P(Y=1|X=?)$

i.e. given X, yield (or “predict”) the probability that $Y=1$

In machine learning, tradition is to use **Y** for the variable being predicted and **X** for the features use to make the prediction.

Logistic Regression

Binary classification goal: Build a "model" that can estimate P(Y=1|X=?)

i.e. given X, yield (or "predict") the probability that Y=1

In machine learning, tradition is to use **Y** for the variable being predicted and **X** for the features use to make the prediction.

Example: Y: 1 if target is verb, 0 otherwise;
X: 1 if "was" occurs before target; 0 otherwise

I was reading for NLP. *We were fine.* *I am good.*

The cat was very happy. *We enjoyed the reading material.* *I was good.*

Logistic Regression

Binary classification goal: Build a “model” that can estimate P(Y=1|X=?)

i.e. given X, yield (or “predict”) the probability that Y=1

In machine learning, tradition is to use **Y** for the variable being predicted and **X** for the features use to make the prediction.

Example: Y: 1 if target is verb, 0 otherwise;
X: 1 if “was” occurs before target; 0 otherwise

I was reading for NLP. *We were fine.* *I am good.*

The cat was very happy. *We enjoyed the reading material.* *I was good.*

Logistic Regression

Example: Y: 1 if target is a part of a proper noun, 0 otherwise;
X: number of capital letters in target and surrounding words.

They attend Stony Brook University. *Next to the brook Gandalf lay thinking.*

The trail was very stony. *Her degree is from SUNY Stony Brook.*

The Taylor Series was first described by Brook Taylor, the mathematician.

Logistic Regression

Example: Y: 1 if target is a part of a proper noun, 0 otherwise;
X: number of capital letters in target and surrounding words.

They attend Stony Brook University. *Next to the brook Gandalf lay thinking.*

The trail was very stony. *Her degree is from SUNY Stony Brook.*

The Taylor Series was first described by Brook Taylor, the mathematician.

Logistic Regression

Example: Y: 1 if target is a part of a proper noun, 0 otherwise;
X: number of capital letters in target and surrounding words.

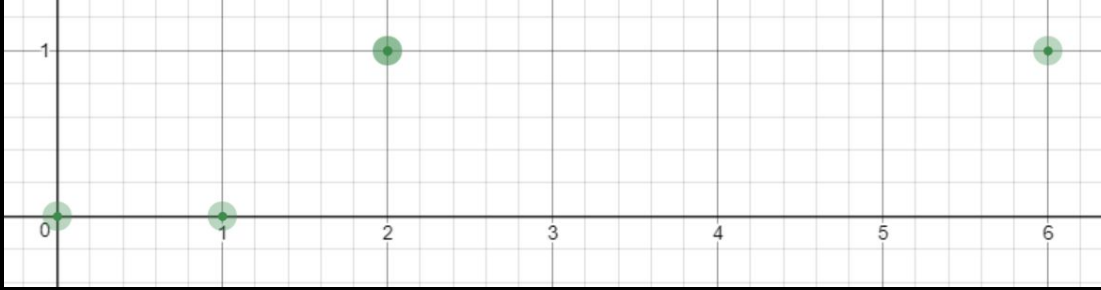
They attend Stony Brook University. *Next to the brook Gandalf lay thinking.*

The trail was very stony. *Her degree is from SUNY Stony Brook.*

The Taylor Series was first described by Brook Taylor, the mathematician.

x	y
2	1
1	0
0	0
6	1
2	1

Logistic Regression



Example: Y: 1 if target is a part of a proper noun, 0 otherwise;
X: number of capital letters in target and surrounding words.

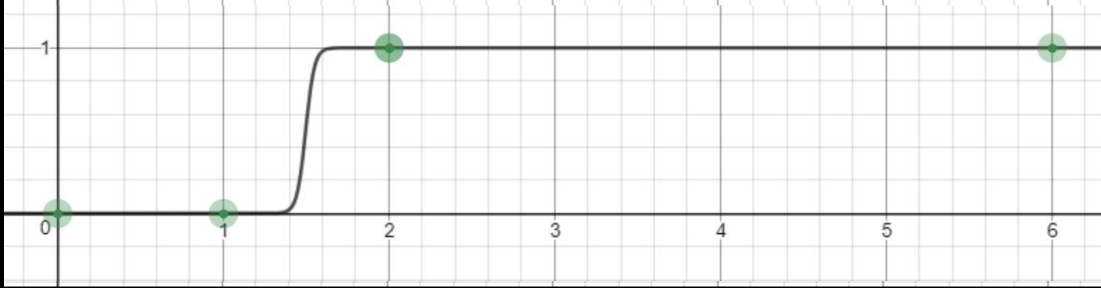
They attend Stony Brook University. *Next to the brook Gandalf lay thinking.*

The trail was very stony. *Her degree is from SUNY Stony Brook.*

The Taylor Series was first described by Brook Taylor, the mathematician.

x	y
2	1
1	0
0	0
6	1
2	1

Logistic Regression



Example: Y: 1 if target is a part of a proper noun, 0 otherwise;
X: number of capital letters in target and surrounding words.

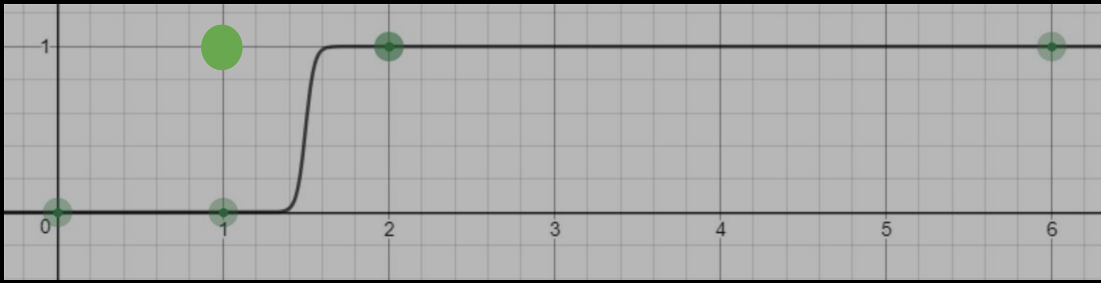
They attend Stony Brook University. *Next to the brook Gandalf lay thinking.*

The trail was very stony. *Her degree is from SUNY Stony Brook.*

The Taylor Series was first described by Brook Taylor, the mathematician.

x	y
2	1
1	0
0	0
6	1
2	1

Logistic Regression



Example: Y: 1 if target is a part of a proper noun, 0 otherwise;
X: number of capital letters in target and surrounding words.

They attend Stony Brook University. *Next to the brook Gandalf lay thinking.*

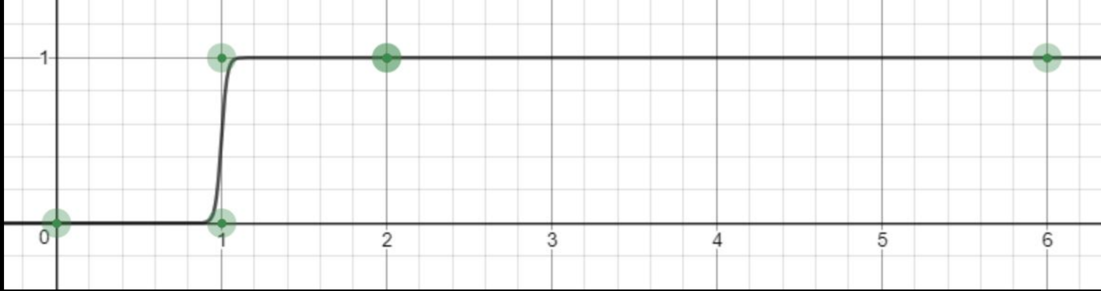
The trail was very stony. *Her degree is from SUNY Stony Brook.*

The Taylor Series was first described by Brook Taylor, the mathematician.

They attend Binghamton.

x	y
2	1
1	0
0	0
6	1
2	1
1	1

Logistic Regression



Example: Y: 1 if target is a part of a proper noun, 0 otherwise;
X: number of capital letters in target and surrounding words.

They attend Stony Brook University. *Next to the brook Gandalf lay thinking.*

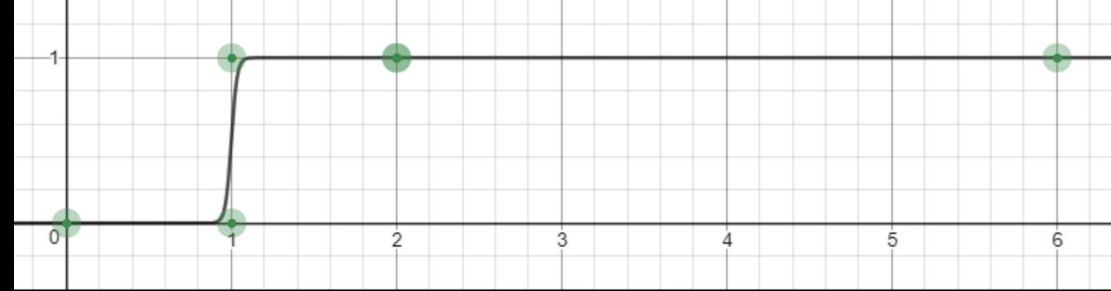
The trail was very stony. *Her degree is from SUNY Stony Brook.*

The Taylor Series was first described by Brook Taylor, the mathematician.

They attend Binghamton.

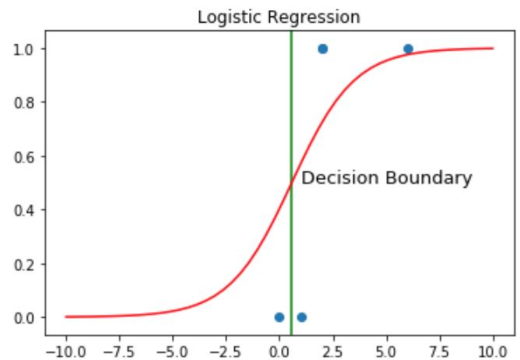
x	y
2	1
1	0
0	0
6	1
2	1
1	1

Logistic Regression



Example: Y: 1 if target is a part of a proper noun, 0 otherwise;
X: number of capital letters in target and surrounding words.

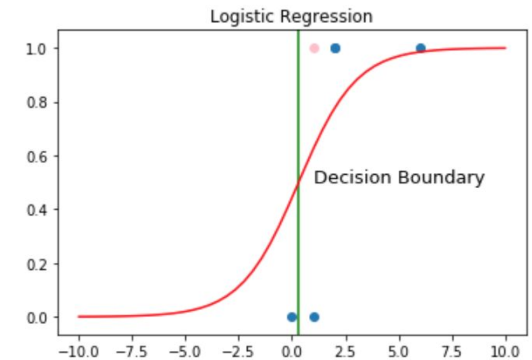
```
Out[43]: [<matplotlib.lines.Line2D at 0x116e68d68>]
```




```
In [78]: -b_0/b_1
Out[78]: 0.5824799517820446

In [28]: logisticRegr.predict(x)
Out[28]: array([1, 1, 0, 1, 1])
```

```
Out[80]: [<matplotlib.lines.Line2D at 0x11a60f160>]
```

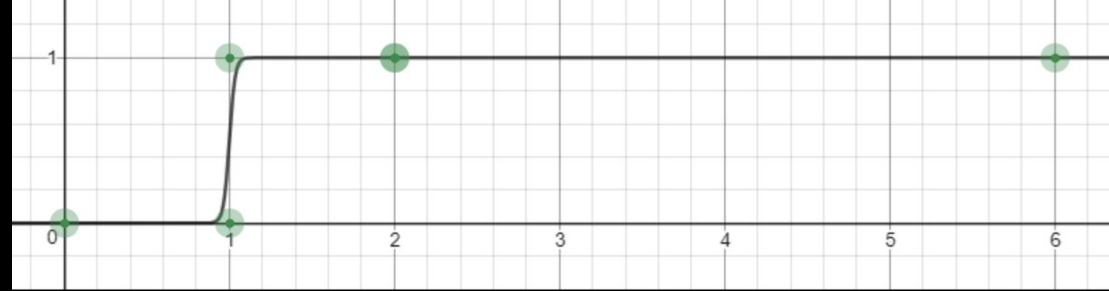



```
In [81]: -b2_0/b2_1
Out[81]: 0.31089309388058134

In [82]: logisticRegr2.predict(x2)
Out[82]: array([1, 1, 0, 1, 1, 1])
```

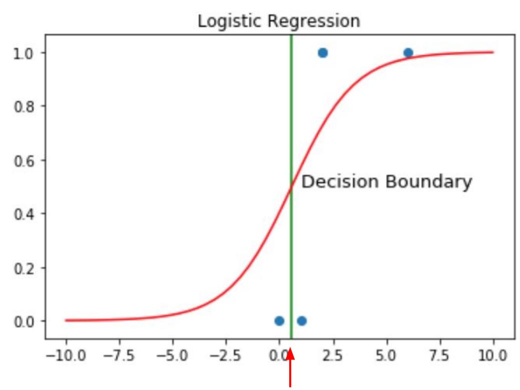
1	1

Logistic Regression



Example: Y: 1 if target is a part of a proper noun, 0 otherwise;
X: number of capital letters in target and surrounding words.

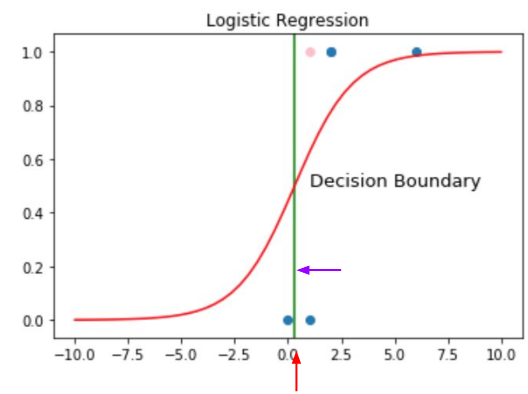
```
Out[43]: [<matplotlib.lines.Line2D at 0x116e68d68>]
```




```
In [78]: -b_0/b_1
Out[78]: 0.5824799517820446

In [28]: logisticRegr.predict(x)
Out[28]: array([1, 1, 0, 1, 1])
```

```
Out[80]: [<matplotlib.lines.Line2D at 0x11a60f160>]
```

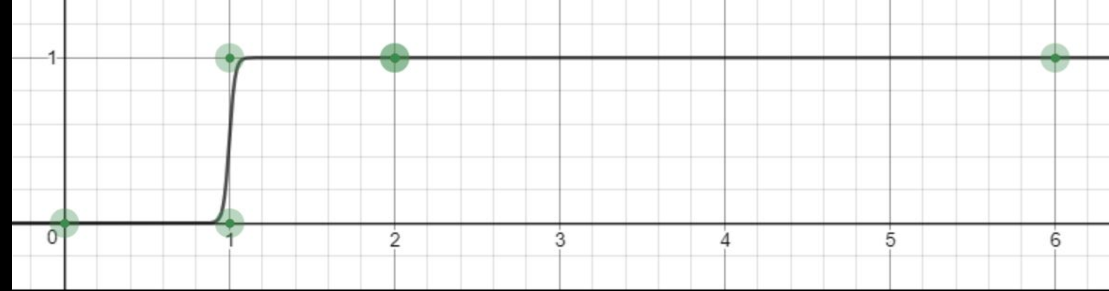



```
In [81]: -b2_0/b2_1
Out[81]: 0.31089309388058134

In [82]: logisticRegr2.predict(x2)
Out[82]: array([1, 1, 0, 1, 1, 1])
```

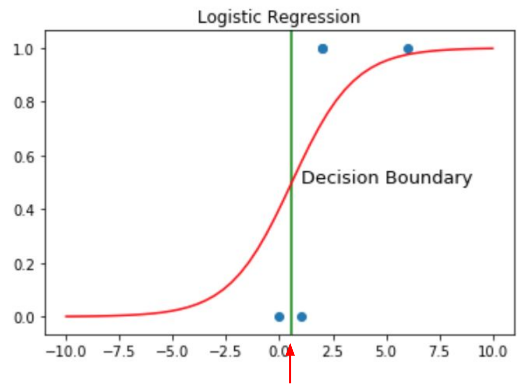
1	1

Logistic Regression



Example: Y: 1 if target is a part of a proper noun, 0 otherwise;
X: number of capital letters in target and surrounding words.

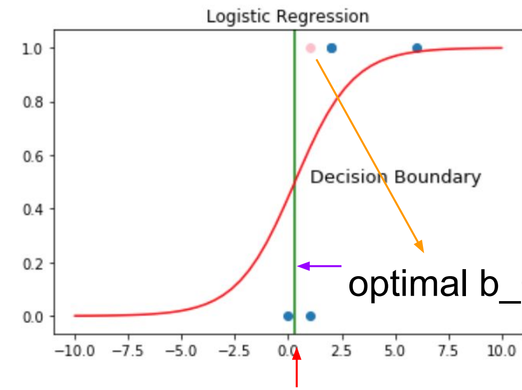
```
Out[43]: [<matplotlib.lines.Line2D at 0x116e68d68>]
```




```
In [78]: -b_0/b_1
Out[78]: 0.5824799517820446

In [28]: logisticRegr.predict(x)
Out[28]: array([1, 1, 0, 1, 1])
```

```
Out[80]: [<matplotlib.lines.Line2D at 0x11a60f160>]
```



optimal b_0, b_1 changed!

```
In [81]: -b2_0/b2_1
Out[81]: 0.31089309388058134

In [82]: logisticRegr2.predict(x2)
Out[82]: array([1, 1, 0, 1, 1, 1])
```

1	1

Logistic Regression on a single feature (*x*)

$Y_i \in \{0, 1\}$; X is a **single value** and can be anything numeric.

$$p_i \equiv P(Y_i = 1 | X_i = x) = \frac{e^{\beta_0 + \beta_1 x_i}}{1 + e^{\beta_0 + \beta_1 x_i}}$$

Logistic Regression on a single feature (*x*)

$Y_i \in \{0, 1\}$; X can be anything numeric.

$$p_i \equiv P(Y_i = 1 | X_i = x) = \frac{e^{\beta_0 + \beta_1 x_i}}{1 + e^{\beta_0 + \beta_1 x_i}}$$

The goal of this function is to: <u>take in the variable *x*</u> and <u>return a probability that *Y* is 1</u>.

Logistic Regression on a single feature (*x*)

$Y_i \in \{0, 1\}$; X can be anything numeric.

$$p_i \equiv P(Y_i = 1 | X_i = x) = \frac{e^{\boxed{\beta_0}+\boxed{\beta_1}\boxed{x_i}}}{1 + e^{\boxed{\beta_0}+\boxed{\beta_1}\boxed{x_i}}}$$

The goal of this function is to: take in the variable *x* and return a probability that *Y* is 1.

Note that there are only three variables on the right: X_i, B_0, B_1

Logistic Regression on a single feature (x)

$Y_i \in \{0, 1\}$; X can be anything numeric.

$$p_i \equiv P(Y_i = 1 | X_i = x) = \frac{e^{\boxed{\beta_0}+\boxed{\beta_1}\boxed{x_i}}}{1 + e^{\boxed{\beta_0}+\boxed{\beta_1}\boxed{x_i}}}$$

The goal of this function is to: take in the variable *x* and return a probability that *Y* is 1.

Note that there are only three variables on the right: X_i, B_0, B_1

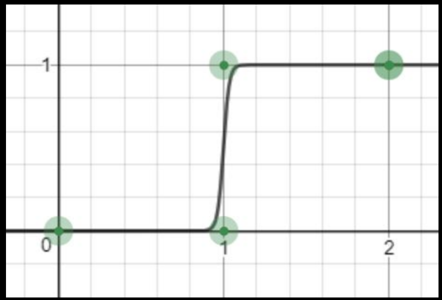
X is given. B_0 and B_1 must be learned.

Logistic Regression on a single feature (*x*)

$Y_i \in \{0, 1\}$; X can be anything numeric.

$$p_i \equiv P(Y_i = 1 | X_i = x) = \frac{e^{\boxed{\beta_0} + \boxed{\beta_1} x_i}}{1 + e^{\boxed{\beta_0} + \boxed{\beta_1} x_i}}$$

HOW? Essentially, try different B_0 and B_1 values until "best fit" to the training data (example X and Y).



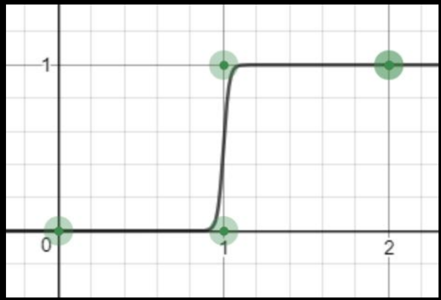
X is given. B_0 and B_1 must be **learned**.

“best fit” : whatever maximizes the likelihood function:

$$L(\beta_0, \beta_1 | X, Y) = \prod_{i=1}^{n} p(x_i)^{y_i}(1 - p(x_i))^{1-y_i}$$

$$p_i \equiv P(Y_i = 1 | X_i = x) = \frac{e^{\boxed{\beta_0}+\boxed{\beta_1}x_i}}{1 + e^{\boxed{\beta_0}+\boxed{\beta_1}x_i}}$$

HOW? Essentially, try different B_0 and B_1 values until “best fit” to the training data (example X and Y).



X is given. B_0 and B_1 must be **learned**.

"best fit" : whatever maximizes the likelihood function:

$$L(\beta_0, \beta_1 | X, Y) = \prod_{i=1}^{n} p(x_i)^{y_i} (1 - p(x_i))^{1-y_i}$$

$$p_i \equiv P(Y_i = 1 | X_i = x) = \frac{e^{\boxed{\beta_0} + \boxed{\beta_1} x_i}}{1 + e^{\boxed{\beta_0} + \boxed{\beta_1} x_i}}$$

To estimate β, one can use *reweighted least squares:*

(Wasserman, 2005; Li, 2010)

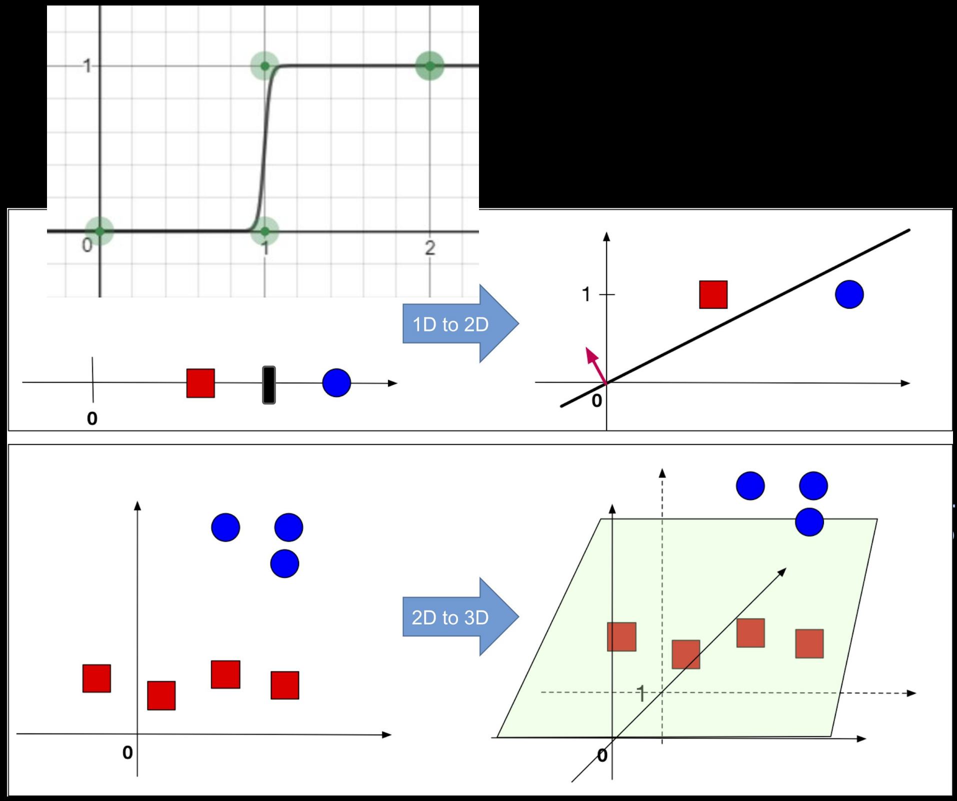
set $\hat{\beta}_0 = ... = \hat{\beta}_m = 0$ (remember to include an intercept)

1. Calculate p_i and let W be a diagonal matrix where element$(i, i) = p_i(1 - p_i)$.
2. Set $z_i = logit(p_i) + \frac{Y_i - p_i}{p_i(1 - p_i)} = X\hat{\beta} + \frac{Y_i - p_i}{p_i(1 - p_i)}$
3. Set $\hat{\beta} = (X^T W X)^{-1} X^T W z$ //weighted lin. reg. of Z on Y.
4. Repeat from 1 until $\hat{\beta}$ converges.

X can be multiple features

Often we want to make a classification based on multiple features:

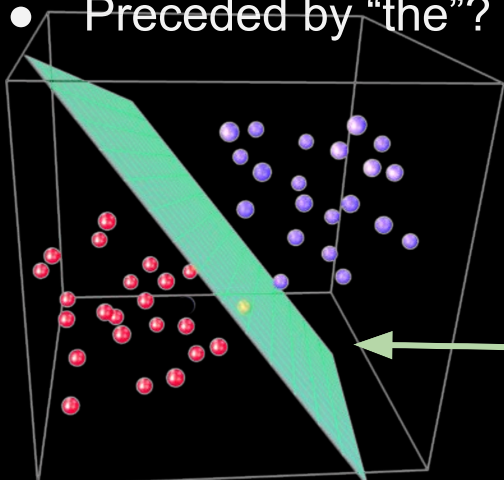
- Number of capital letters surrounding: integer
- Begins with capital letter: {0, 1}
- Preceded by “the”? {0, 1}



X can be multiple features

Often we want to make a classification based on multiple features:

- Number of capital letters surrounding: integer
- Begins with capital letter: {0, 1}
- Preceded by “the”? {0, 1}



We’re learning a linear (i.e. flat) *separating hyperplane*, but fitting it to a *logit* outcome.

(https://www.linkedin.com/pulse/predicting-outcomes-probabilities-logistic-regression-konstantinidis/)

"best fit" : whatever maximizes the likelihood function:

$$L(\beta_0, \beta_1, ..., \beta_k | X, Y) = \prod_{i=1}^{n} p(x_i)^{y_i}(1 - p(x_i))^{1-y_i}$$

$$p_i \equiv P(Y_i = 1 | X_i = x) = \frac{e^{\boxed{\beta_0}+\boxed{\beta_1}x_i}}{1 + e^{\boxed{\beta_0}+\boxed{\beta_1}x_i}}$$

To estimate β , one can use *reweighted least squares:*

(Wasserman, 2005; Li, 2010)

set $\hat{\beta}_0 = ... = \hat{\beta}_m = 0$ (remember to include an intercept)

1. Calculate p_i and let W be a diagonal matrix where element$(i, i) = p_i(1 - p_i)$.
2. Set $z_i = logit(p_i) + \frac{Y_i - p_i}{p_i(1 - p_i)} = X\hat{\beta} + \frac{Y_i - p_i}{p_i(1 - p_i)}$
3. Set $\hat{\beta} = (X^T W X)^{-1} X^T W z$ //weighted lin. reg. of Z on Y.
4. Repeat from 1 until $\hat{\beta}$ converges.

"best fit" : whatever maximizes the likelihood function:

$$L(\beta_0, \beta_1, \ldots, \beta_k | X, Y) = \prod_{i=1}^{n} p(x_i)^{y_i}(1 - p(x_i))^{1-y_i}$$

This is just one way of finding the betas that maximize the likelihood function. In practice, we will use existing libraries that are fast and support additional useful steps like **regularization**..

To estimate β , one can use *reweighted least squares:*

(Wasserman, 2005; Li, 2010)

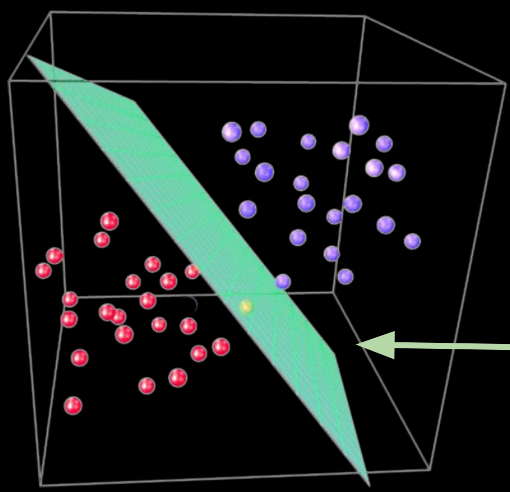
set $\hat{\beta}_0 = \ldots = \hat{\beta}_m = 0$ (remember to include an intercept)

1. Calculate p_i and let W be a diagonal matrix where element$(i, i) = p_i(1 - p_i)$.
2. Set $z_i = logit(p_i) + \frac{Y_i - p_i}{p_i(1 - p_i)} = X\hat{\beta} + \frac{Y_i - p_i}{p_i(1 - p_i)}$
3. Set $\hat{\beta} = (X^T W X)^{-1} X^T W z$ //weighted lin. reg. of Z on Y.
4. Repeat from 1 until $\hat{\beta}$ converges.

Logistic Regression

$Y_i \in \{0, 1\}$; X can be anything numeric.

$$p_i \equiv P(Y_i = 1 | X_i = x) = \frac{e^{\beta_0 + \sum_{j=1}^{m} \beta_j x_{ij}}}{1 + e^{\beta_0 + \sum_{j=1}^{m} \beta_j x_{ij}}}$$



$$logit(p_i) = log\left(\frac{p_i}{1-p_i}\right) = \beta_0 + \sum_{j=1}^{m} \boxed{\beta_j x_{ij}}$$

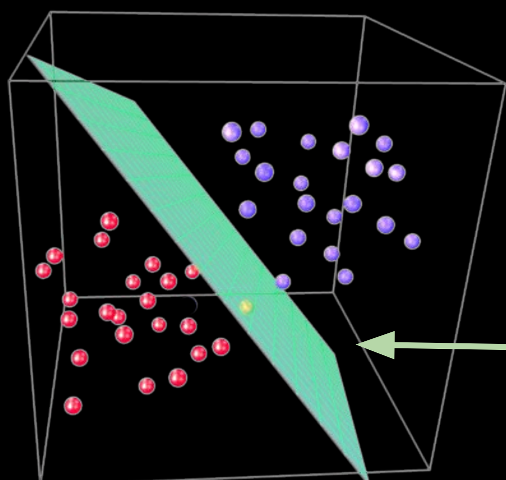
We're learning a linear (i.e. flat) *separating hyperplane*, but fitting it to a *logit* outcome.

(https://www.linkedin.com/pulse/predicting-outcomes-probabilities-logistic-regression-konstantinidis/)

Logistic Regression

$Y_i \in \{0, 1\}$; X can be anything numeric.

$$p_i \equiv P(Y_i = 1 | X_i = x) = \frac{e^{\beta_0 + \sum_{j=1}^{m} \beta_j x_{ij}}}{1 + e^{\beta_0 + \sum_{j=1}^{m} \beta_j x_{ij}}}$$

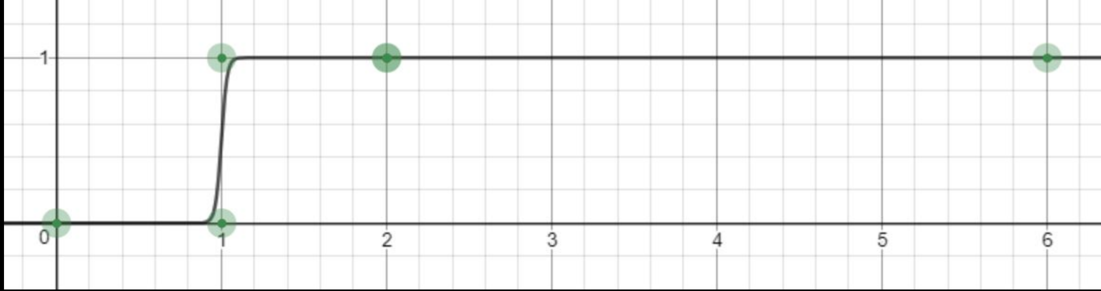


$$logit(p_i) = log\left(\frac{p_i}{1 - p_i}\right) = \beta_0 + \sum_{j=1}^{m} \boxed{\beta_j x_{ij}} = 0$$

We're still learning a linear *separating hyperplane*, but fitting it to a *logit* outcome.

(https://www.linkedin.com/pulse/predicting-outcomes-probabilities-logistic-regression-konstantinidis/)

Logistic Regression



Example: Y: 1 if target is a part of a proper noun, 0 otherwise;
X: number of capital letters in target and surrounding words.

They attend Stony Brook University.

Next to the brook Gandalf lay thinking.

The trail was very stony.

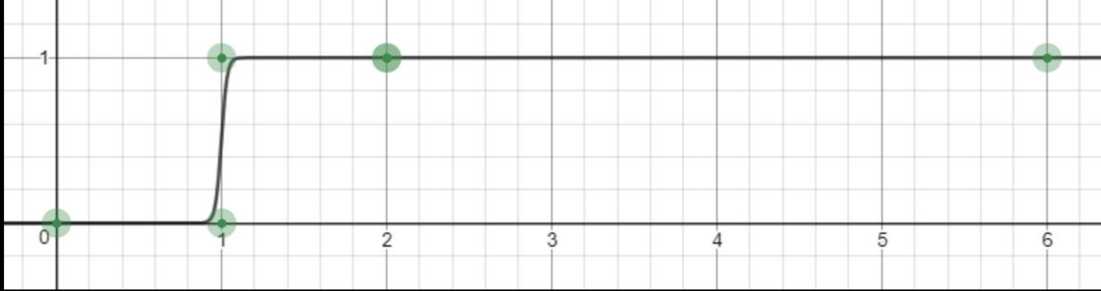
Her degree is from SUNY Stony Brook.

The Taylor Series was first described by Brook Taylor, the mathematician.

They attend Binghamton.

x	y
2	1
1	0
0	0
6	1
2	1
1	1

Logistic Regression



Example: Y: 1 if target is a part of a proper noun, 0 otherwise;

X1: number of capital letters in target and surrounding words.

Let's add a feature! X2: does the target word start with a capital letter?

They attend Stony Brook University. *Next to the brook Gandalf lay thinking.*

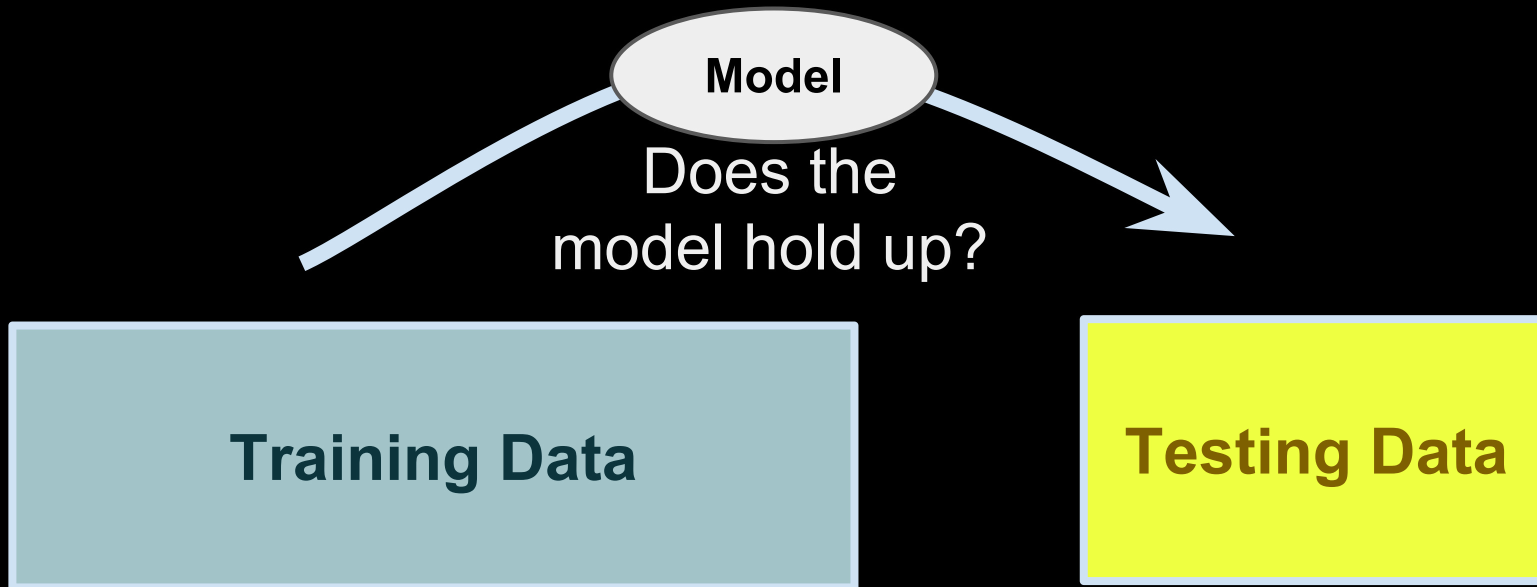
The trail was very stony. *Her degree is from SUNY Stony Brook.*

The Taylor Series was first described by Brook Taylor, the mathematician.

They attend Binghamton.

x2	x1	y
1	2	1
0	1	0
0	0	0
1	6	1
1	2	1
1	1	1

Machine Learning Goal: Generalize to new data



Logistic Regression - Regularization

Last concept for logistic regression!

$$X = Y$$

0.5	0	0.6	1	0	0.25	1
0	0.5	0.3	0	0	0	1
0	0	1	1	1	0.5	0
0	0	0	0	1	1	0
0.25	1	1.25	1	0.1	2	1

Logistic Regression - Regularization

Last concept for logistic regression!

$$X = Y$$

0.5	0	0.6	1	0	0.25	1
0	0.5	0.3	0	0	0	1
0	0	1	1	1	0.5	0
0	0	0	0	1	1	0
0.25	1	1.25	1	0.1	2	1

Last concept for logistic regression!

Logistic Regression - Regularization

x_1	x_2	...	X			= Y
0.5	0	0.6	1	0	0.25	1
0	0.5	0.3	0	0	0	1
0	0	1	1	1	0.5	0
0	0	0	0	1	1	0
0.25	1	1.25	1	0.1	2	1

$$1.2 + -63*x_1 + 179*x_2 + 71*x_3 + 18*x_4 + -59*x_5 + 19*x_6 = \text{logit}(Y)$$

Logistic Regression - Regularization

Last concept for logistic regression!

x_1	x_2	...	X			=	Y
0.5	0	0.6	1	0	0.25		1
0	0.5	0.3	0	0	0		1
0	0	1	1	1	0.5		0
0	0	0	0	1	1		0
0.25	1	1.25	1	0.1	2		1

$$1.2 + -63*x_1 + \boxed{179*x_2} + 71*x_3 + 18*x_4 + -59*x_5 + 19*x_6 = logit(Y)$$

Last concept for logistic regression!

Logistic Regression - Regularization

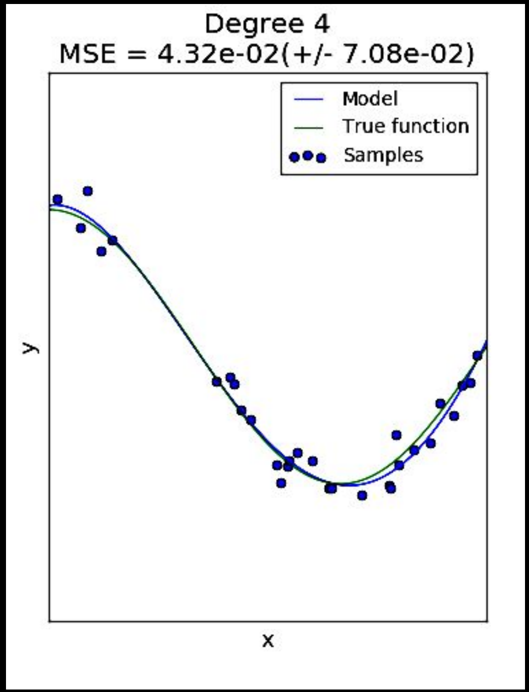
$$X = Y$$

x_1	x_2	...				Y
0.5	0	0.6	1	0	0.25	1
0	0.5	0.[illegible]	[illegible]	[illegible]	0	1
0	0	[illegible]	[illegible]	[illegible]	0.5	0
0	0	0	0	1	1	0
0.25	1	1.25	1	0.1	2	1

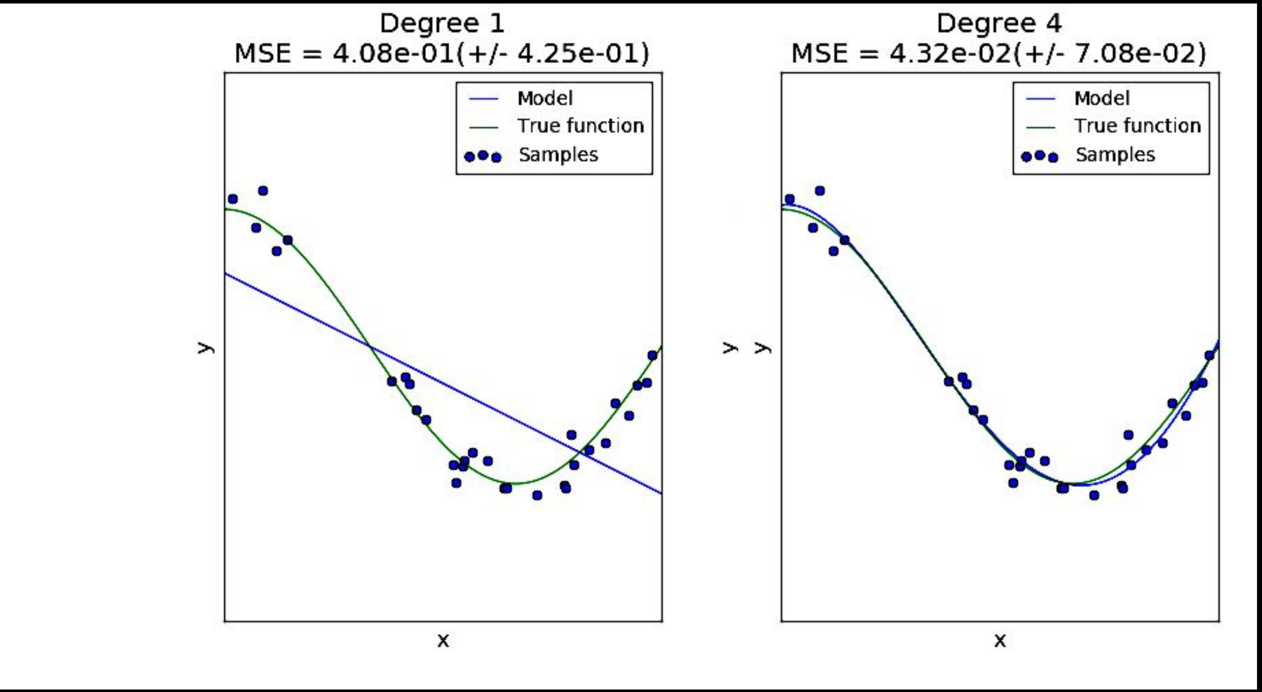
"overfitting"

$$1.2 + -63*x_1 + \boxed{179*x_2} + 71*x_3 + 18*x_4 + -59*x_5 + 19*x_6 = logit(Y)$$

Overfitting (1-d non-linear example)



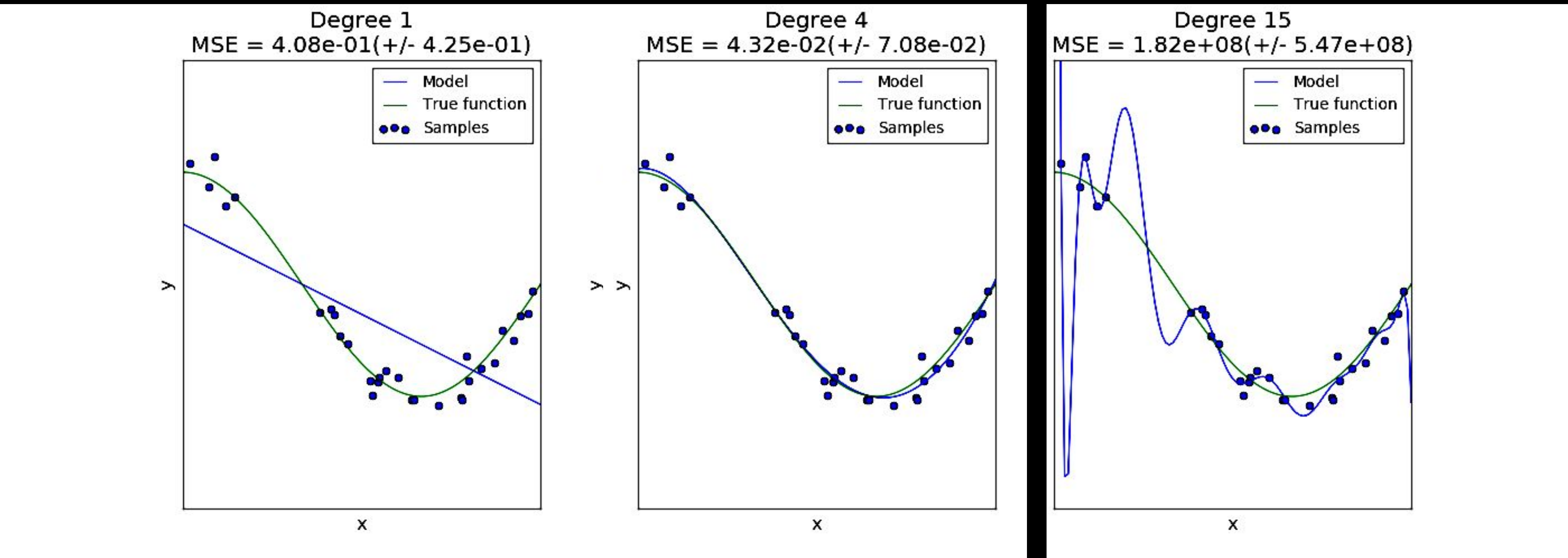
Overfitting (1-d non-linear example)



Underfit

(image credit: Scikit-learn; in practice data are rarely this clear)

Overfitting (1-d non-linear example)



Underfit

Overfit

(image credit: Scikit-learn; in practice data are rarely this clear)

Last concept for logistic regression!

Logistic Regression - Regularization

X = Y

x_1	x_2	...				Y
0.5	0	0.6	1	0	0.25	1
0	0.5	[illegible]	[illegible]	[illegible]	0	1
0	0	[illegible]	[illegible]	[illegible]	0.5	0
0	0	0	0	1	1	0
0.25	1	1.25	1	0.1	2	1

"overfitting"

$$1.2 + -63*x_1 + \boxed{179*x_2} + 71*x_3 + 18*x_4 + -59*x_5 + 19*x_6 = logit(Y)$$

Logistic Regression - Regularization

Last concept for logistic regression!

X

x_1	x_2
0.5	0
0	0.5
0	0
0	0
0.25	1

$=$ Y

Y
1
1
0
0
1

What if only 2 predictors?

Last concept for logistic regression!

Logistic Regression - Regularization

X $=$ Y

x_1	x_2	Y
0.5	0	1
0	0.5	1
0	0	0
0	0	0
0.25	1	1

What if only 2 predictors?
better fit

$$0 + 2 * x_1 + 2 * x_2 = logit(Y)$$

Logistic Regression - Regularization

Last concept for logistic regression!

L1 Regularization - "The Lasso"

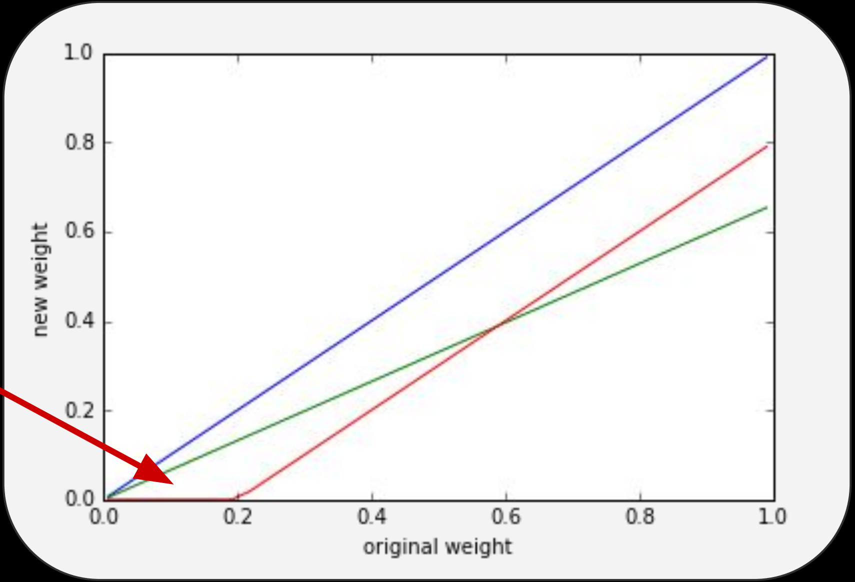
Zeros out features by adding values that keep from perfectly fitting the data.

Logistic Regression - Regularization

Last concept for logistic regression!

L1 Regularization - "The Lasso"

Zeros out features by adding values that keep from perfectly fitting the data.



Logistic Regression - Regularization

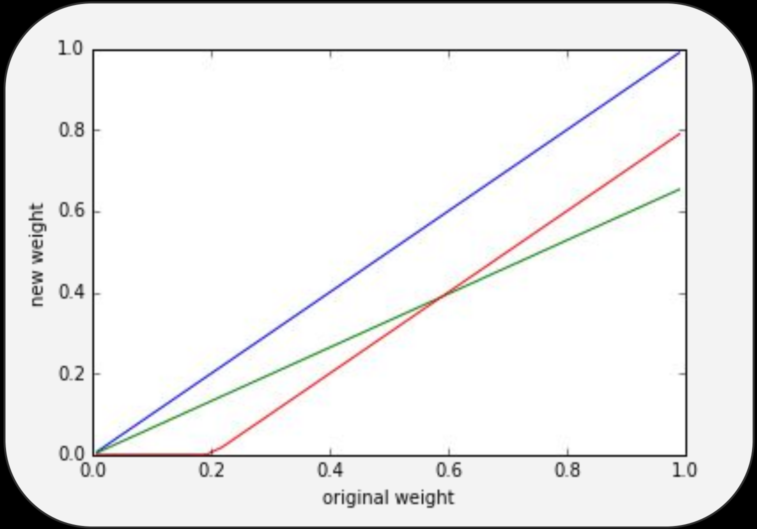
Last concept for logistic regression!

L1 Regularization - “The Lasso”

Zeros out features by adding values that keep from perfectly fitting the data.

$$L(\beta_0, \beta_1, \ldots, \beta_k | X, Y) = \prod_{i=1}^{n} p(x_i)^{y_i} (1 - p(x_i))^{1-y_i}$$

set betas that maximize L



Last concept for logistic regression!

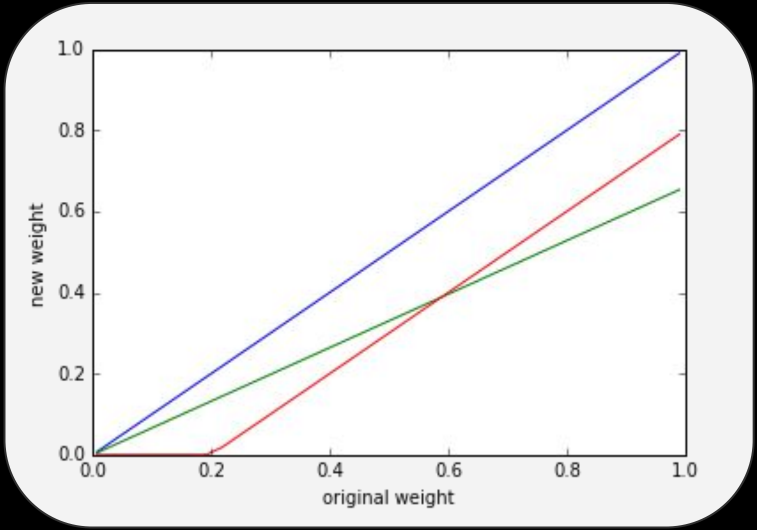
Logistic Regression - Regularization

L1 Regularization - "The Lasso"

Zeros out features by adding values that keep from perfectly fitting the data.

$$L(\beta_0, \beta_1, \ldots, \beta_k | X, Y) = \prod_{i=1}^{n} p(x_i)^{y_i}(1 - p(x_i))^{1-y_i} - \frac{1}{C}\sum_{j=1}^{m} |\beta_j|$$

set betas that maximize *penalized* L



Logistic Regression - Regularization

Last concept for logistic regression!

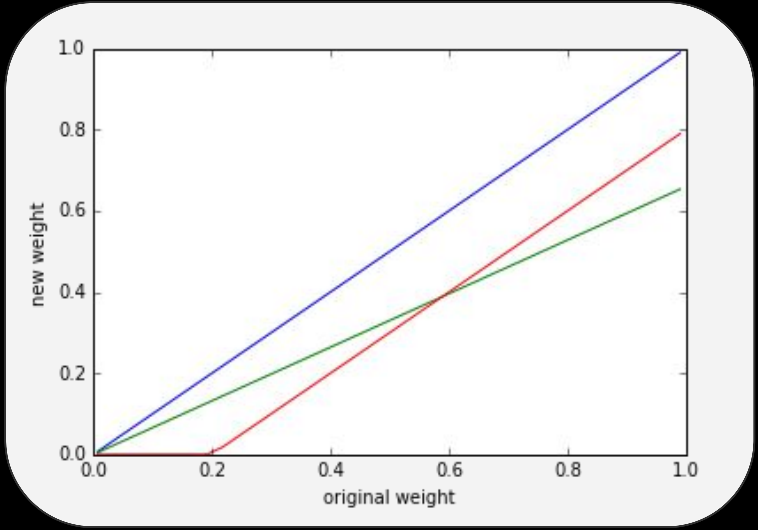
Sometimes written as:
$||\beta||_1$

L1 Regularization - "The Lasso"

Zeros out features by adding values that keep from perfectly fitting the data.

$$L(\beta_0, \beta_1, \ldots, \beta_k | X, Y) = \prod_{i=1}^{n} p(x_i)^{y_i}(1 - p(x_i))^{1-y_i} - \frac{1}{C}\sum_{j=1}^{m} |\beta_j|$$

set betas that maximize *penalized* L



Logistic Regression - Regularization

Last concept for logistic regression!

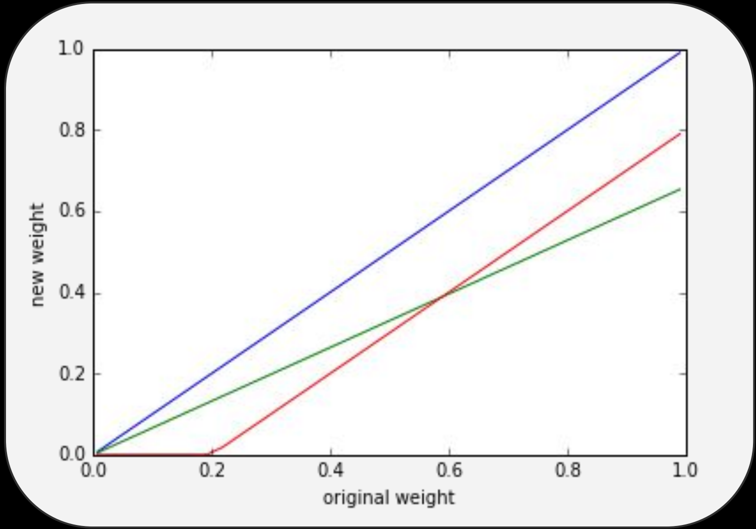
Sometimes written as:
$$||\beta_j||_2^2$$

L2 Regularization - "Ridge"

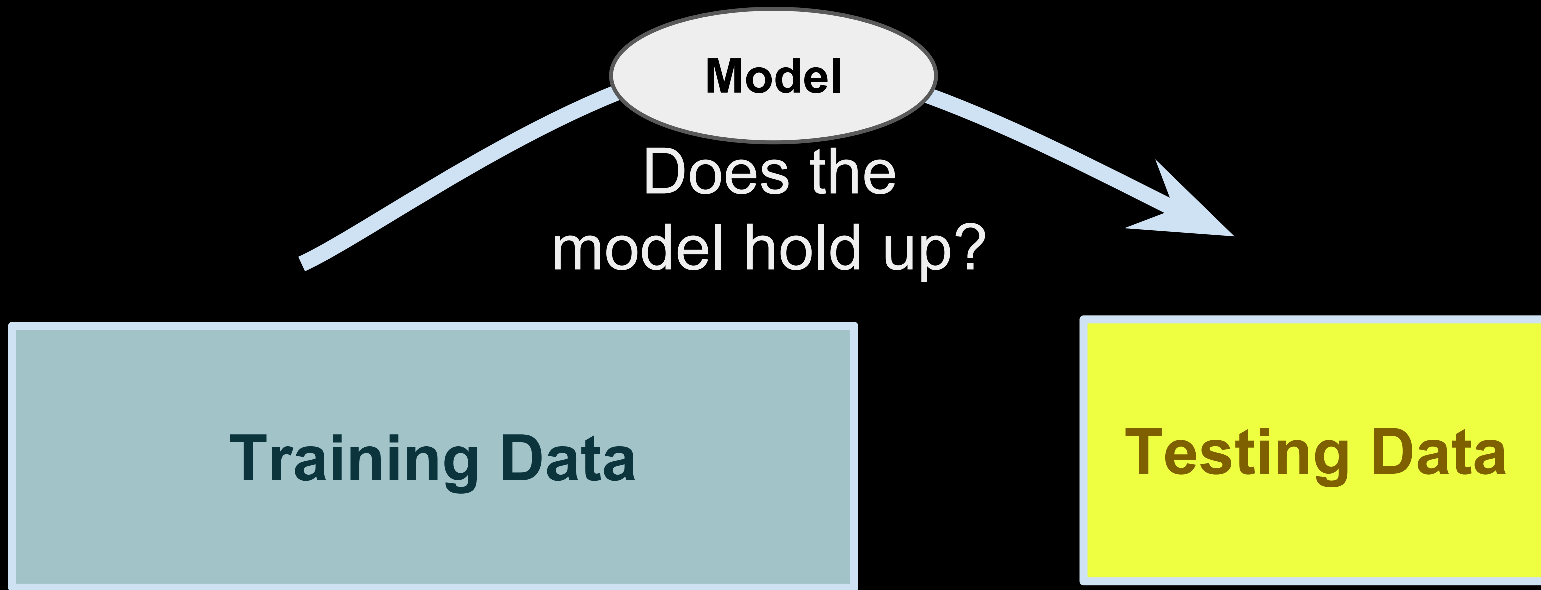
Shrinks features by adding values that keep from perfectly fitting the data.

$$L(\beta_0, \beta_1, \ldots, \beta_k | X, Y) = \prod_{i=1}^{n} p(x_i)^{y_i}(1 - p(x_i))^{1-y_i} - \frac{1}{C}\sum_{j=1}^{m} \beta_j^2$$

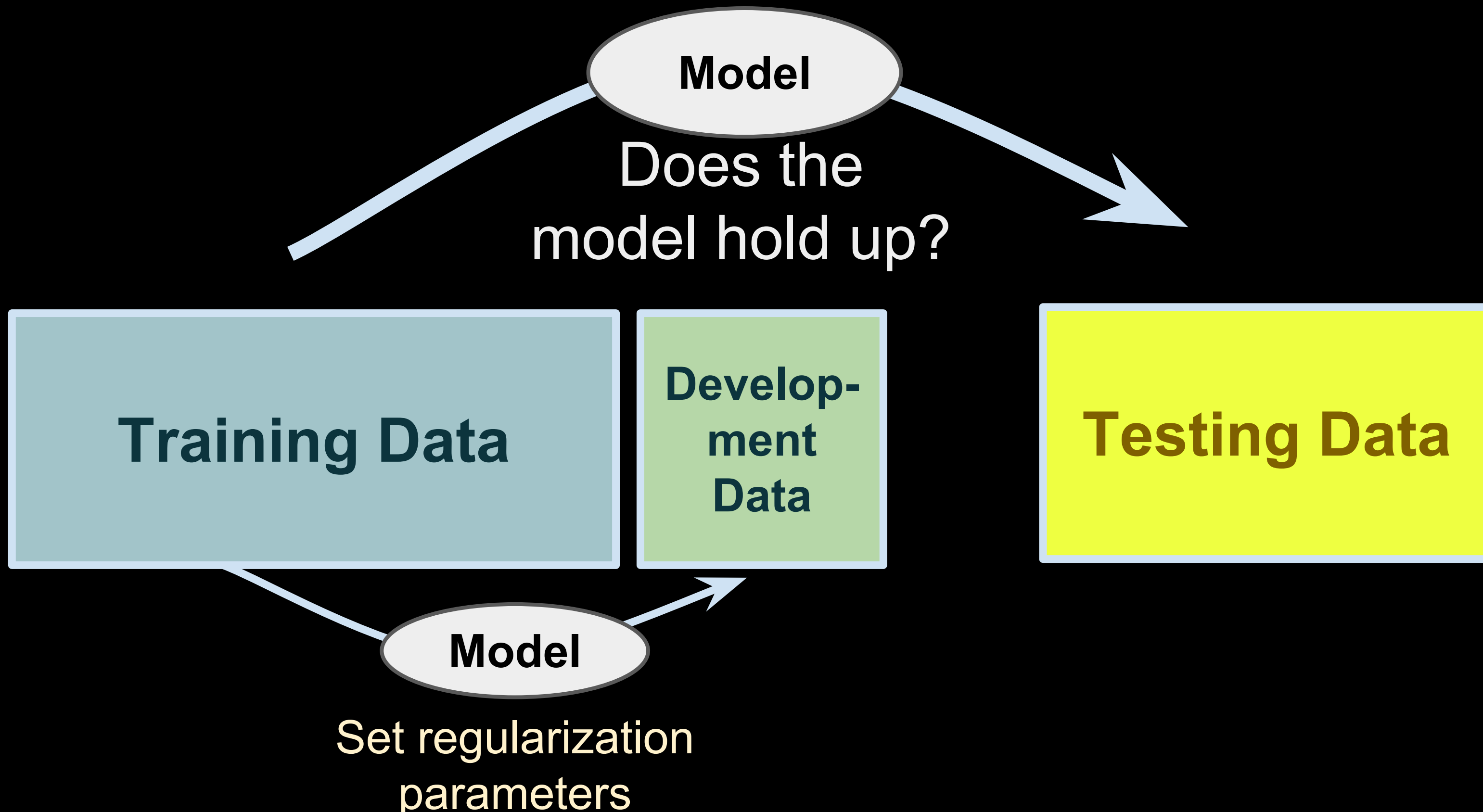
set betas that maximize *penalized* L



Machine Learning Goal: Generalize to new data



Machine Learning Goal: Generalize to new data



Logistic Regression - Review

- Classification: $P(Y \mid X)$
- Learn logistic curve based on example data
    - training + development + testing data
- Set betas based on maximizing the likelihood
    - “shifts” and “twists” the logistic curve
- Multivariate features
- Separation represented by hyperplane
- Overfitting
- Regularization

Example

See notebook on website.

In [44]:

```python
%matplotlib inline

#above allows plots to discplay on the screen.

#python includes
import sys

#standard probability includes:
import numpy as np #matrices and data structures
import scipy.stats as ss #standard statistical operations
import pandas as pd #keeps data organized, works well with data
import matplotlib
import matplotlib.pyplot as plt #plot visualization
```

In [53]:

```python
#let's just look at what happens to the logit function as we change the beta coefficients

def logistic_function(x):
    return np.exp(x) / (1+np.exp(x))

def logistic_function_with_betas(x, beta0=0, beta1=1):
    #now using linear function: beta0 + beta1*x for the exponent:
    return np.exp(beta0 + beta1*x) / (1+np.exp(beta0 + beta1*x))

xpoints = np.linspace(-10, 10, 100)
plt.plot(xpoints, [logistic_function(x) for x in xpoints])
plt.plot(xpoints, [logistic_function_with_betas(x, 2, 1) for x in xpoints]) #shifts the intercept with zero
plt.plot(xpoints, [logistic_function_with_betas(x, 0, 3.145914159653) for x in xpoints])#twists the line verically
plt.plot(xpoints, [logistic_function_with_betas(x, 0, 1/3.145914159653) for x in xpoints]) #twists it horizontally
```

Out[53]:

```
[<matplotlib.lines.Line2D at 0x2691f435f60>]
```