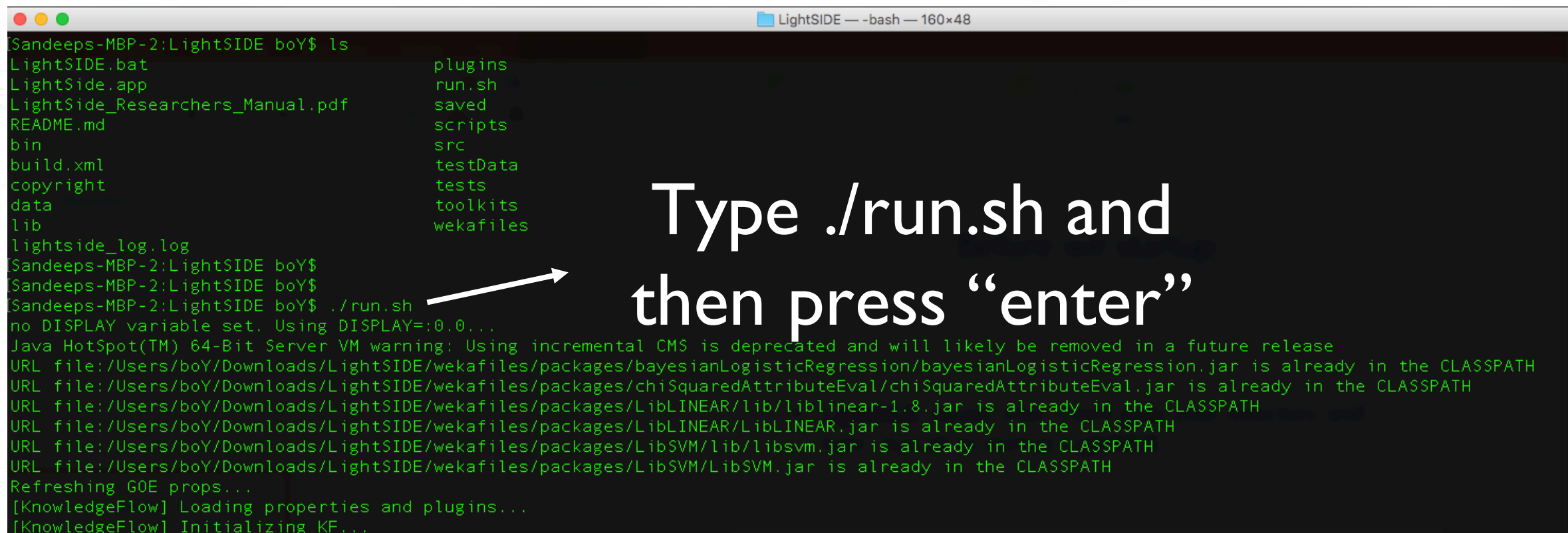# Linear Classifiers

Sandeep Avula

asandeep@unc.edu

Before we startup

- LightSIDE issues: Mac Users goto your directory and type ./run.sh to startup LightSIDE.


```
LightSIDE — -bash — 160×48
$andeeps-MBP-2:LightSIDE boY$ ls
LightSIDE.bat                           plugins
LightSide.app                           run.sh
LightSide_Researchers_Manual.pdf        saved
README.md                               scripts
bin                                     src
build.xml                               testData
copyright                               tests
data                                    toolkits
lib                                     wekafiles
lightside_log.log
$andeeps-MBP-2:LightSIDE boY$
$andeeps-MBP-2:LightSIDE boY$
$andeeps-MBP-2:LightSIDE boY$ ./run.sh
no DISPLAY variable set. Using DISPLAY=:0.0...
Java HotSpot(TM) 64-Bit Server VM warning: Using incremental CMS is deprecated and will likely be removed in a future release
URL file:/Users/boY/Downloads/LightSIDE/wekafiles/packages/bayesianLogisticRegression/bayesianLogisticRegression.jar is already in the CLASSPATH
URL file:/Users/boY/Downloads/LightSIDE/wekafiles/packages/chiSquaredAttributeEval/chiSquaredAttributeEval.jar is already in the CLASSPATH
URL file:/Users/boY/Downloads/LightSIDE/wekafiles/packages/LibLINEAR/lib/liblinear-1.8.jar is already in the CLASSPATH
URL file:/Users/boY/Downloads/LightSIDE/wekafiles/packages/LibLINEAR/LibLINEAR.jar is already in the CLASSPATH
URL file:/Users/boY/Downloads/LightSIDE/wekafiles/packages/LibSVM/lib/libsvm.jar is already in the CLASSPATH
URL file:/Users/boY/Downloads/LightSIDE/wekafiles/packages/LibSVM/LibSVM.jar is already in the CLASSPATH
Refreshing GOE props...
[KnowledgeFlow] Loading properties and plugins...
[KnowledgeFlow] Initializing KF...
```

Type ./run.sh and then press "enter"

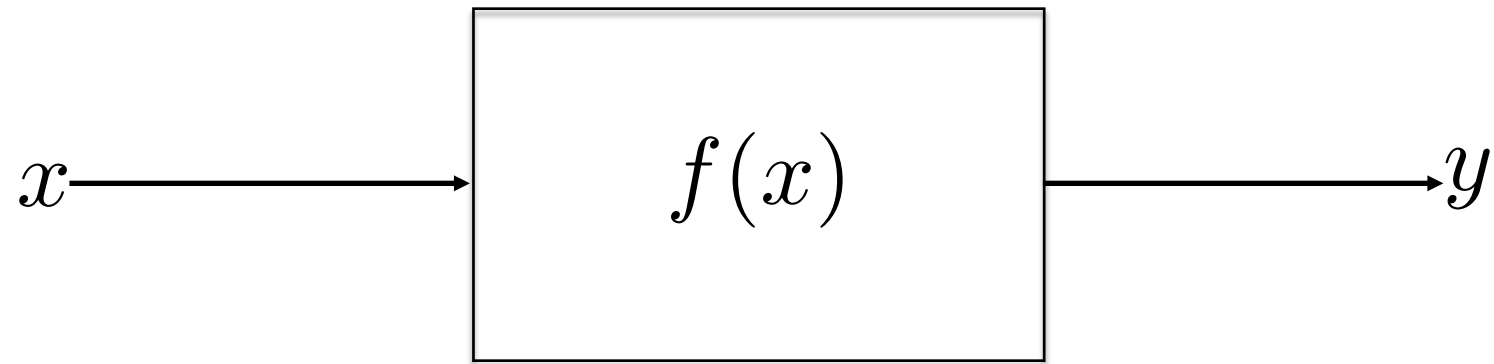- HW2 out today. Deadline will be adjusted.

Overview

- Philosophical questions

- Derivatives: What are they good for?

- Linear regression
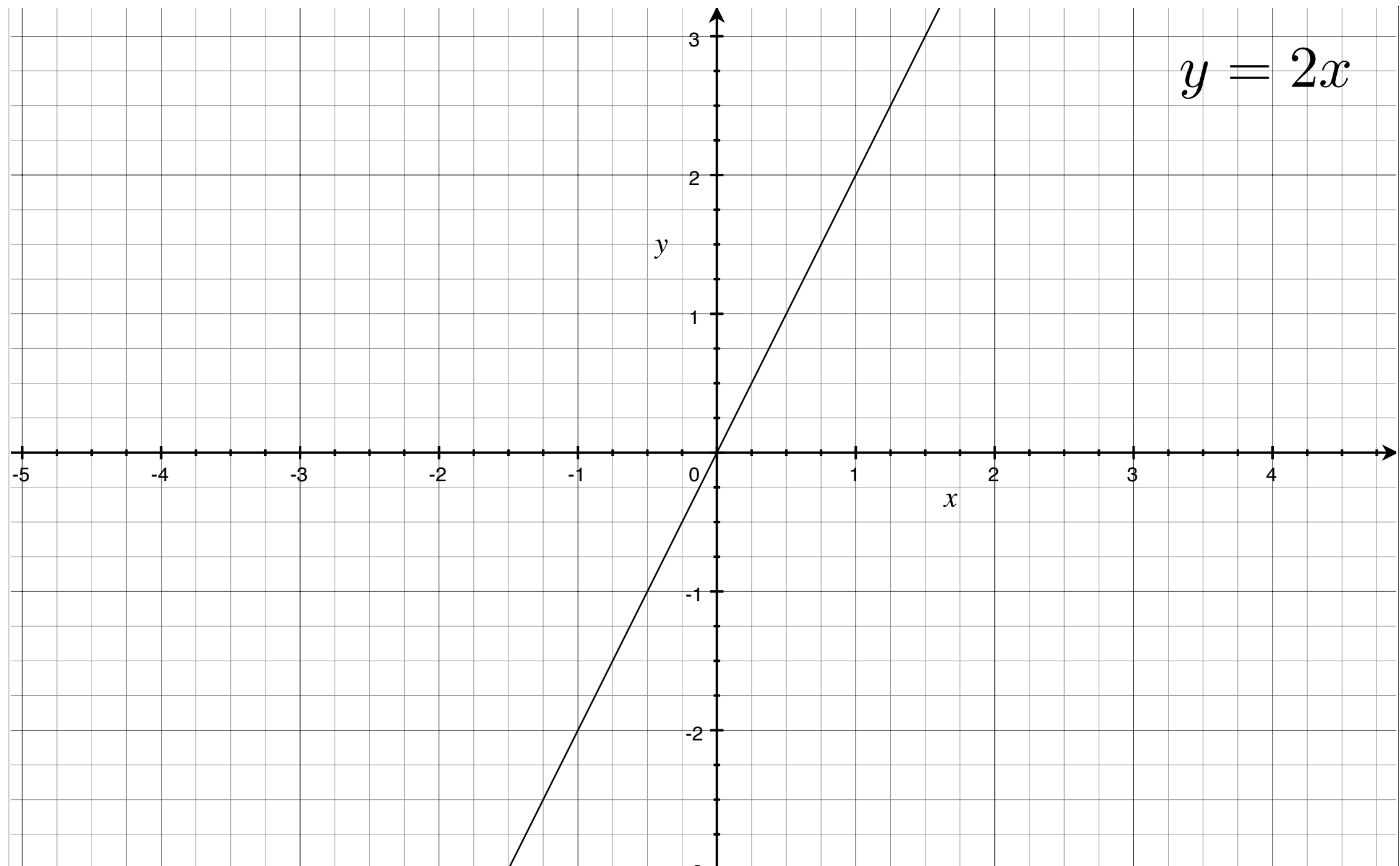
- Multiple linear regression

- Logistic regression

Philosophical Questions

- What would you do if …

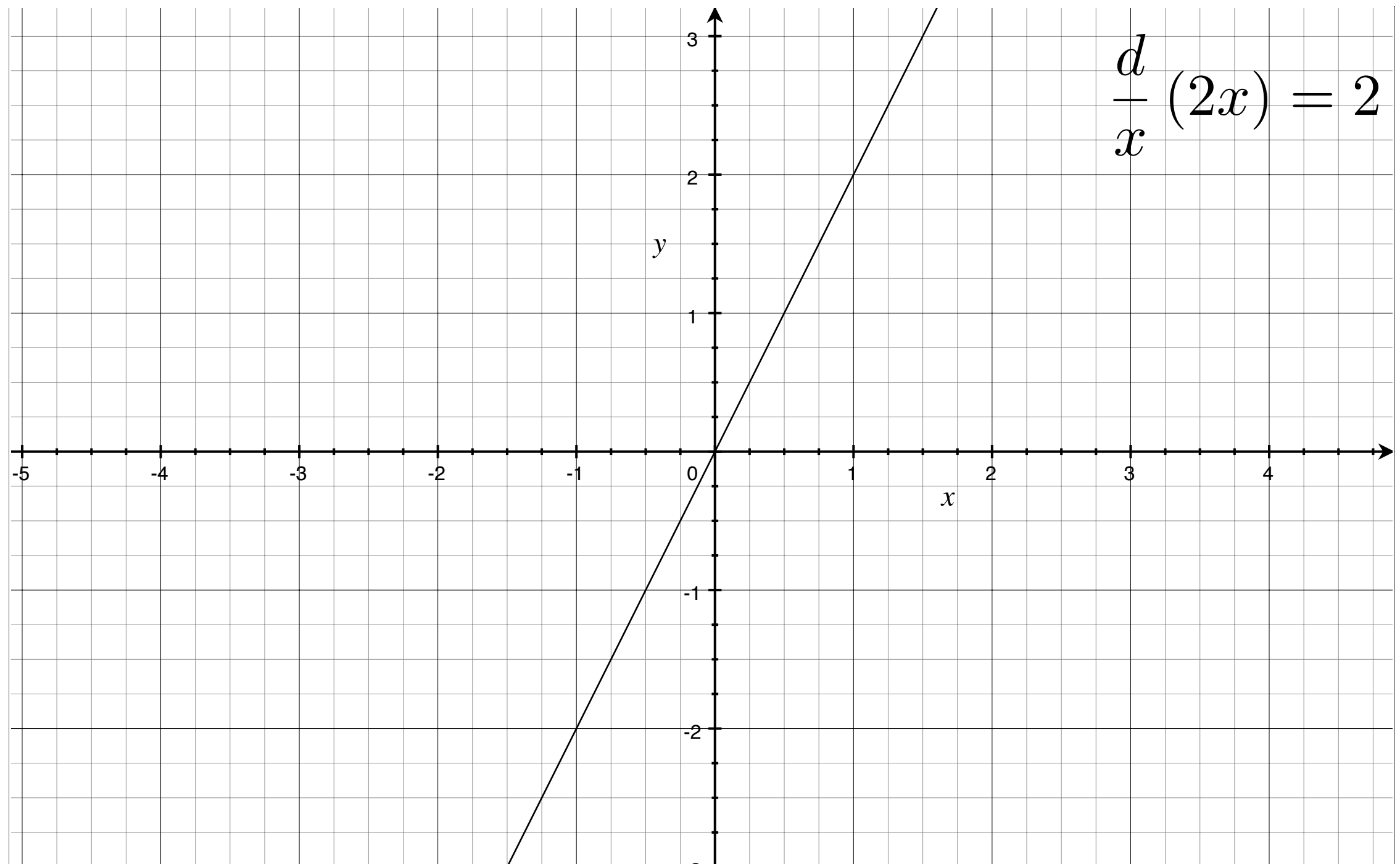- What does this have to do with linear classifiers?

Functions

$$x \longrightarrow \boxed{f(x)} \longrightarrow y$$

Derivatives

$$y = 2x$$

Derivatives

$$\frac{d}{x}(2x) = 2$$

Derivatives

$y = x^2$

Derivatives

$$\frac{d}{x}\left(x^2\right) = 2x$$
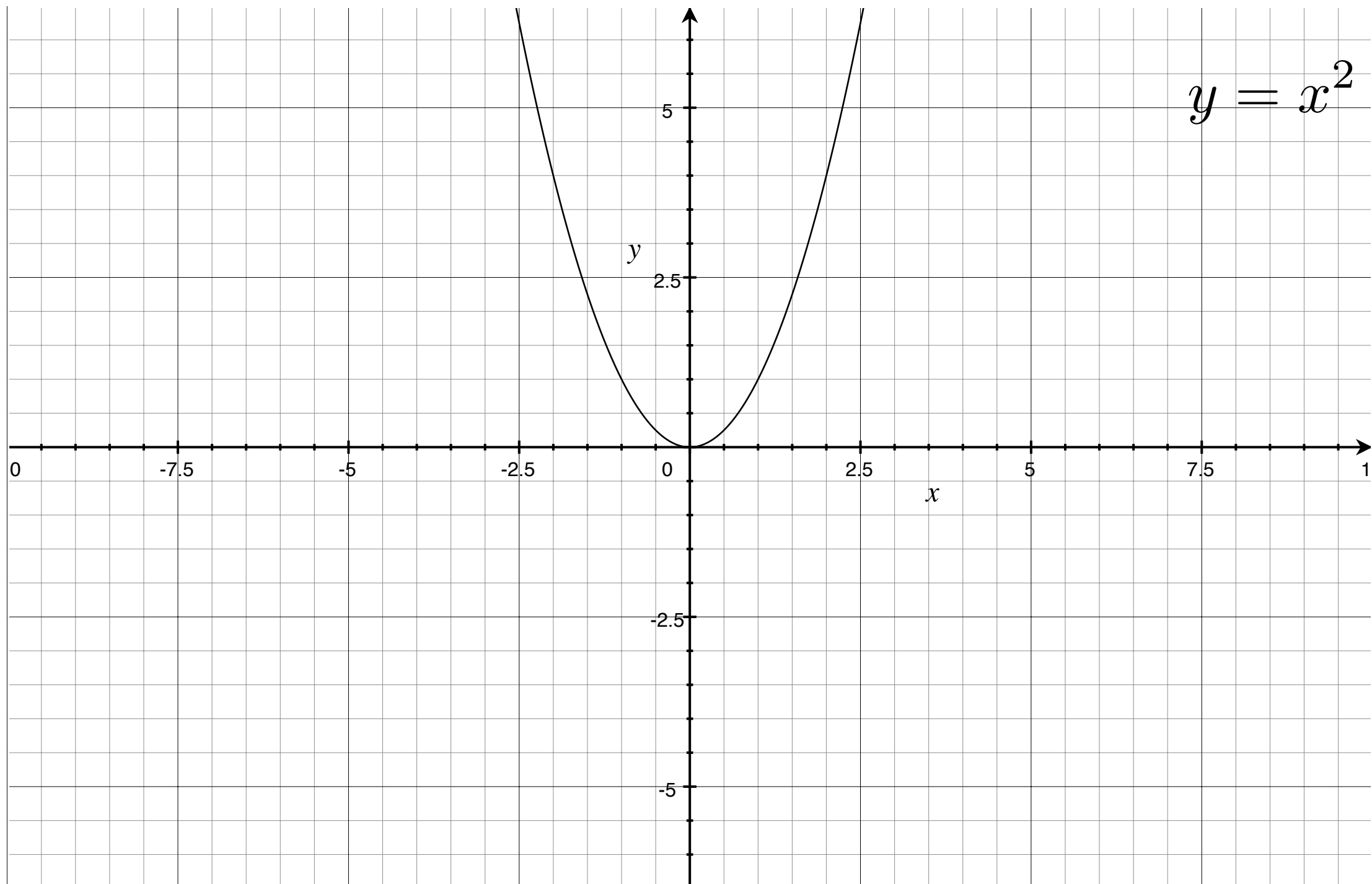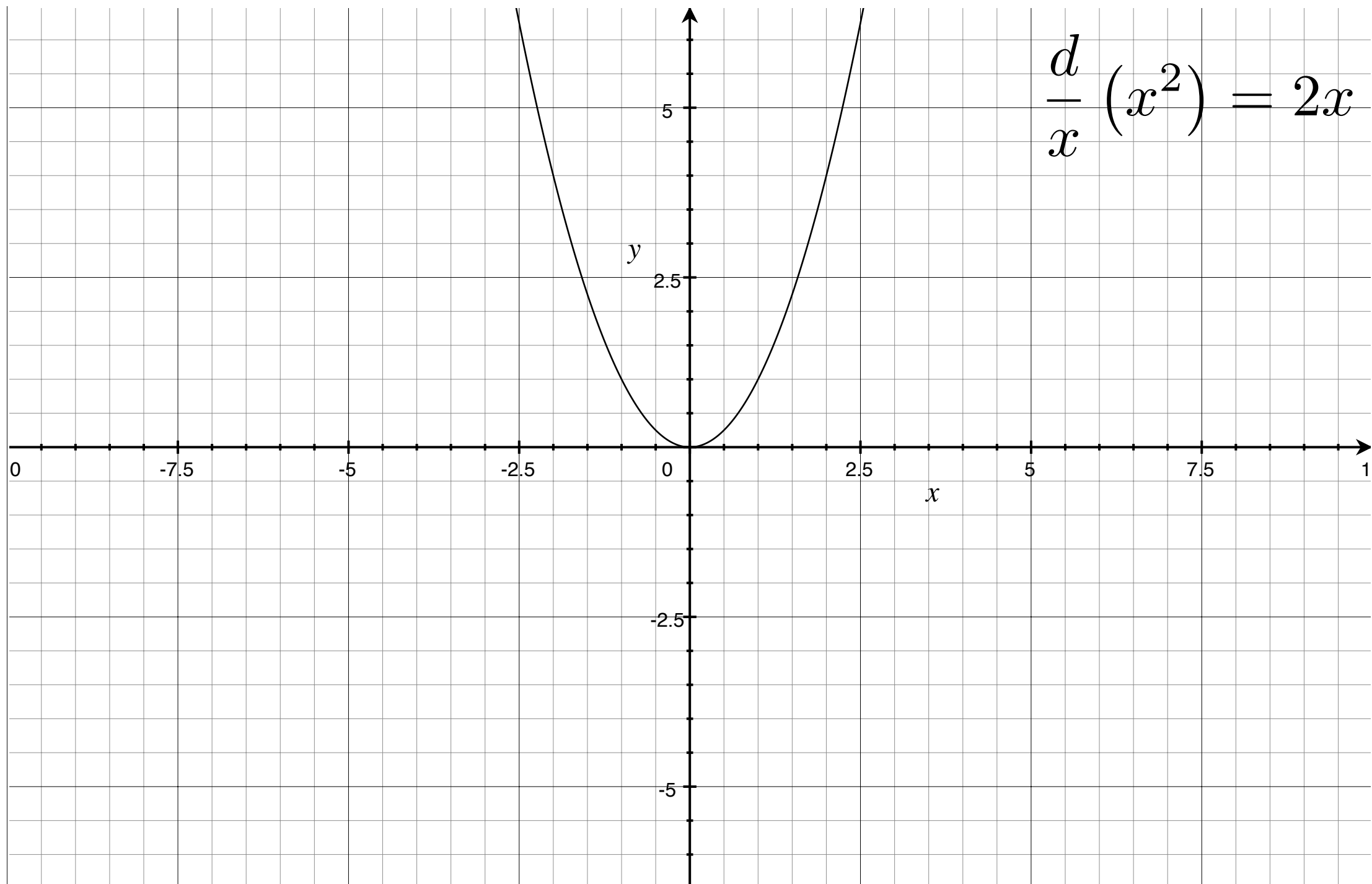
Derivatives: What are they good for?

- The derivative of f(x) outputs the slope of f(x) for a particular value of x

- A point of which the slope is zero is a point at which f(x) is at its highest or lowest value.

- What does this have to do with machine learning?

Derivatives

$$\frac{d}{x}\left(x^2\right) = 2x$$

$$2x = 0$$

$$x = 0$$
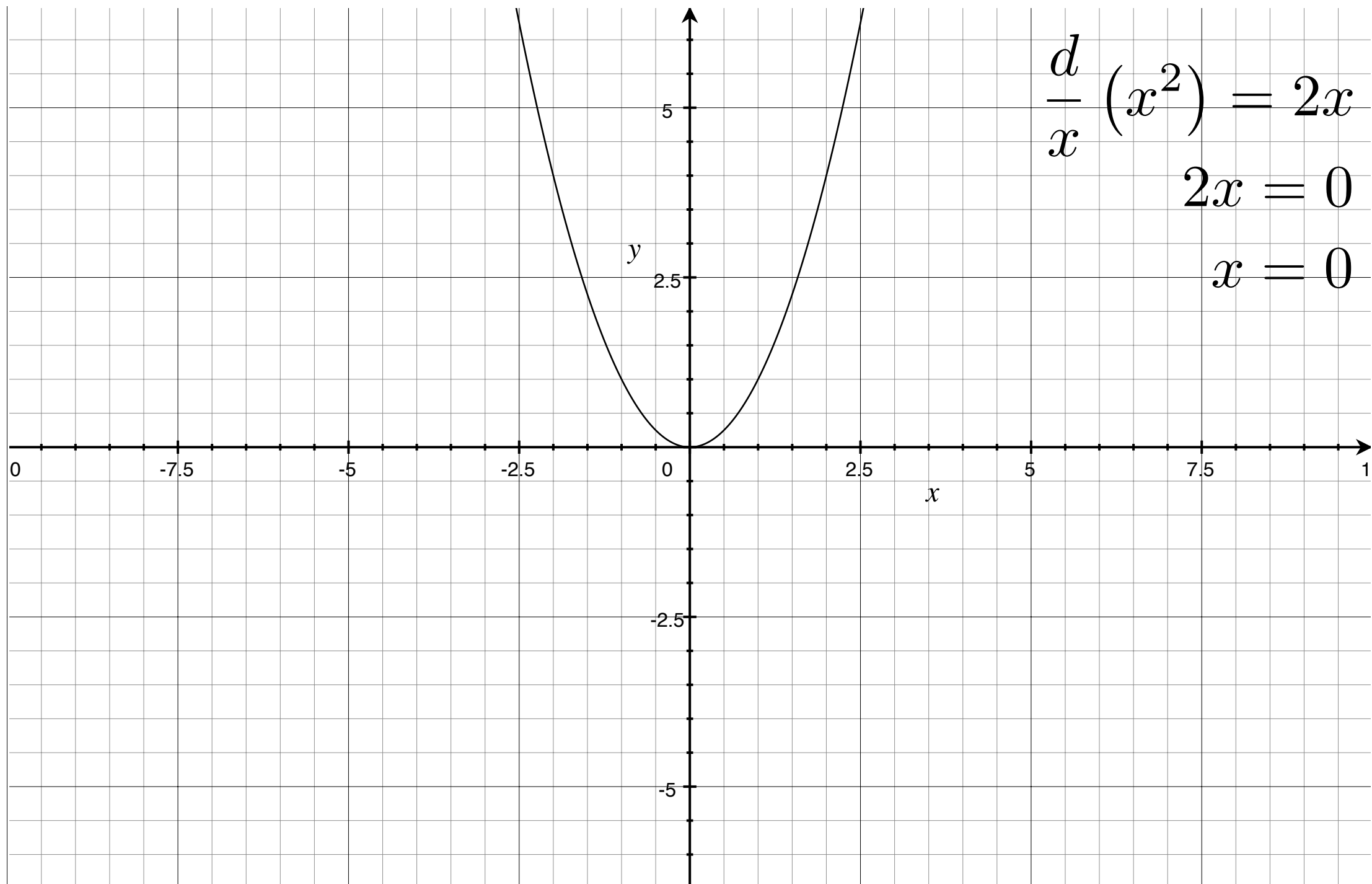
Computation Graphs

$$y = 3(a + bc)$$

Computation Graphs

$$y = 3(a + bc)$$

a

$u = (a + v)$

b

c

$v = bc$

$y = 3u$
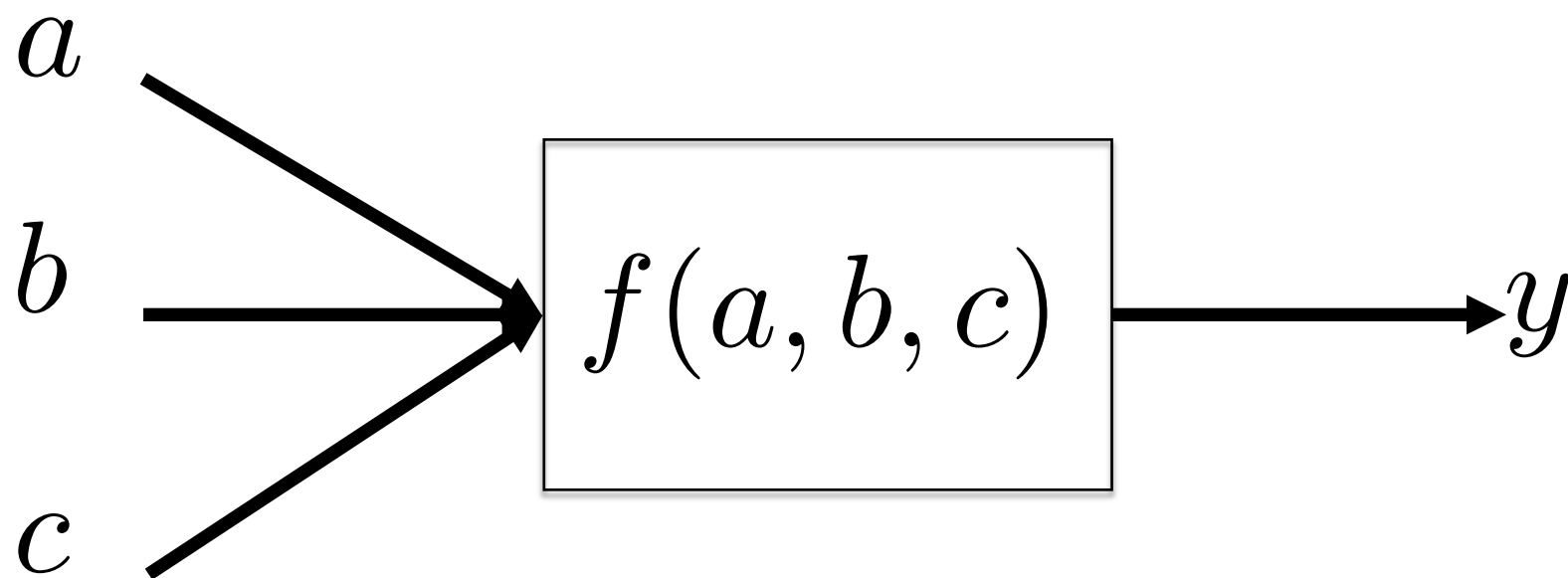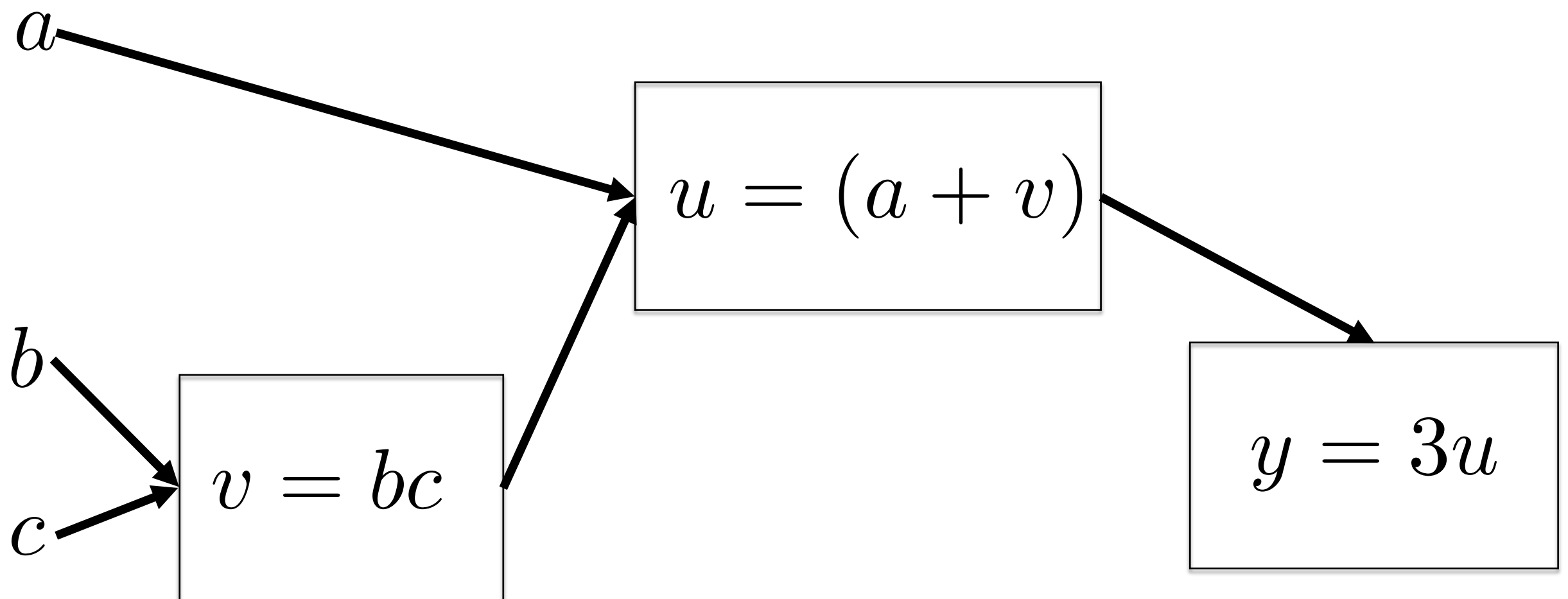
Derivatives: Chain Rule

$$y = 3(a + bc)$$

$$\frac{dy}{dc} = \frac{dv}{dc} \times \frac{du}{dv} \times \frac{dy}{du}$$

a

b

c

$$v = bc$$

$$u = (a + v)$$

$$y = 3u$$
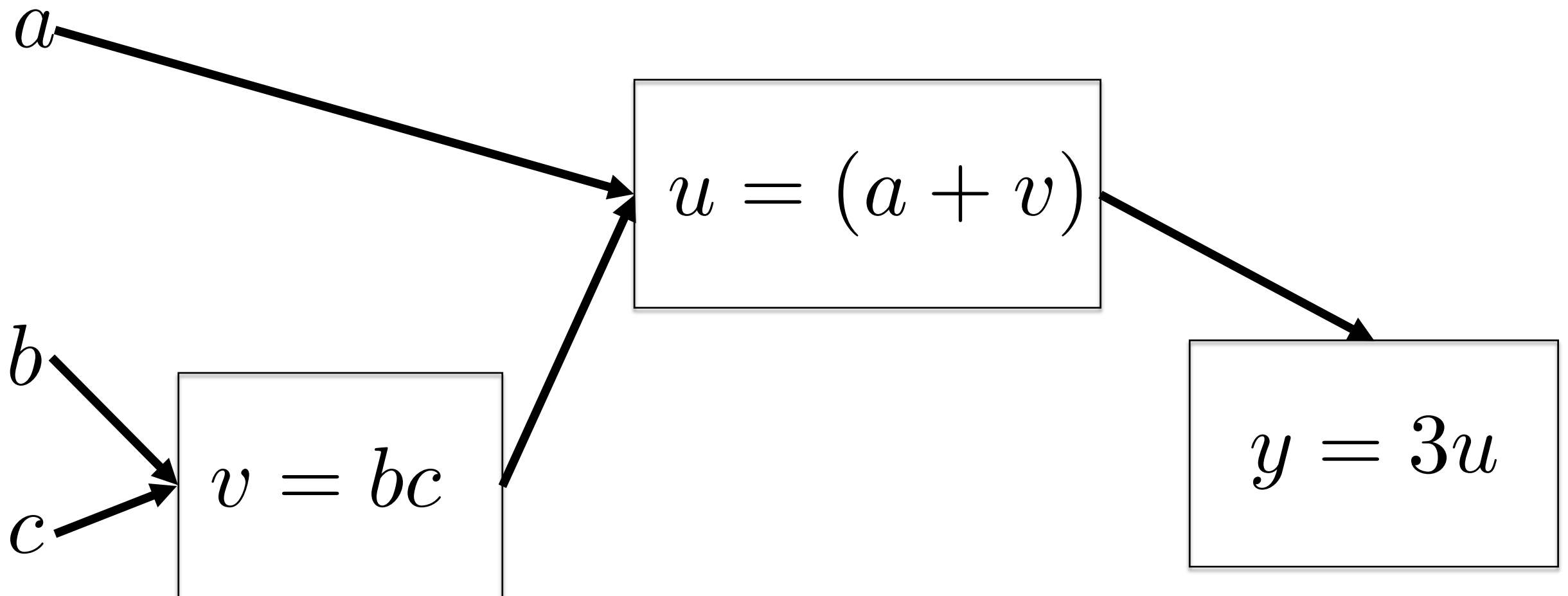
Derivatives: Chain Rule

$$y = 3(a + bc)$$

$$\frac{dy}{dc} = b \times 1 \times 3 = 3b$$
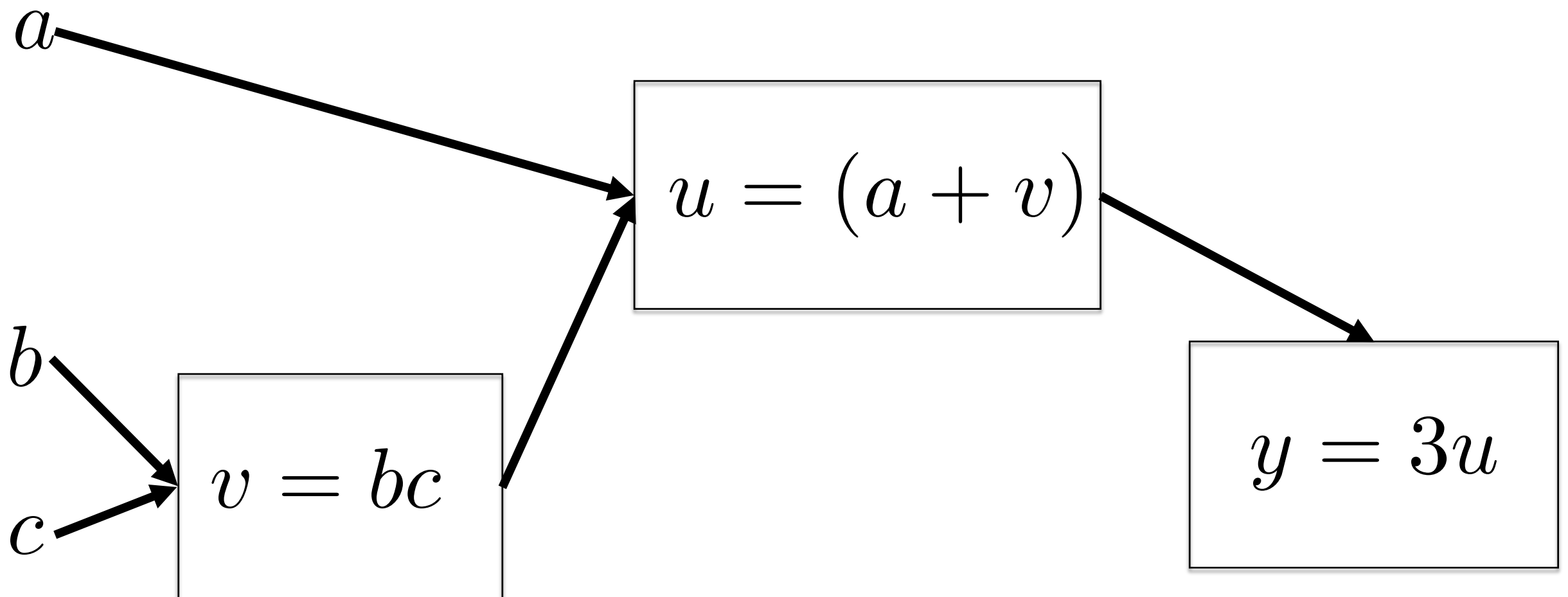
a

b

c

$v = bc$

$u = (a + v)$

$y = 3u$

Overview

- Philosophical questions

- Derivatives: What are they good for?

- **Linear regression**

- Multiple linear regression

- Logistic regression

Linear Regression

$$y = wx + b$$

Linear Regression

yawns from students

Temperature in Rm 001

Linear Regression

yawns from students

Temperature in Rm 001

Linear Regression

$$y = wx + b$$

yawns from students

Temperature in Rm 001
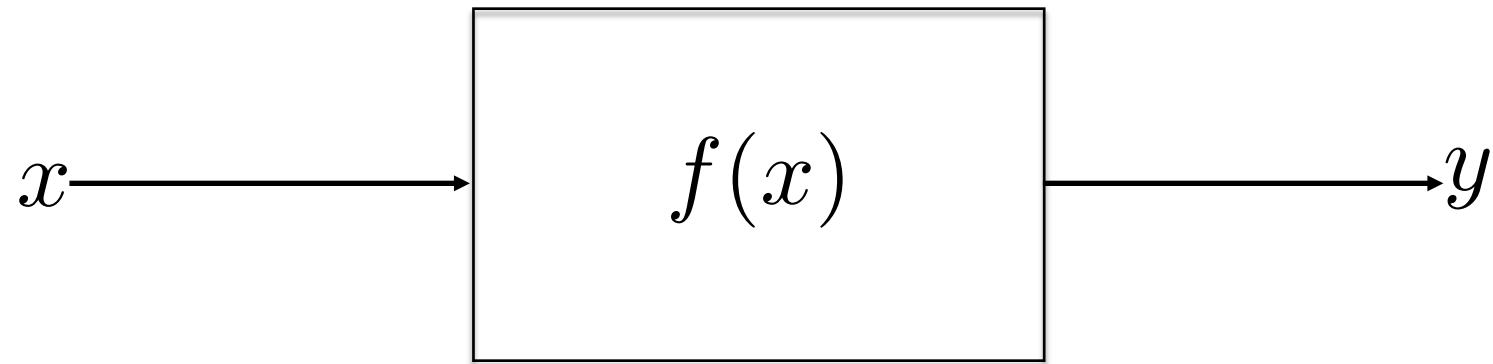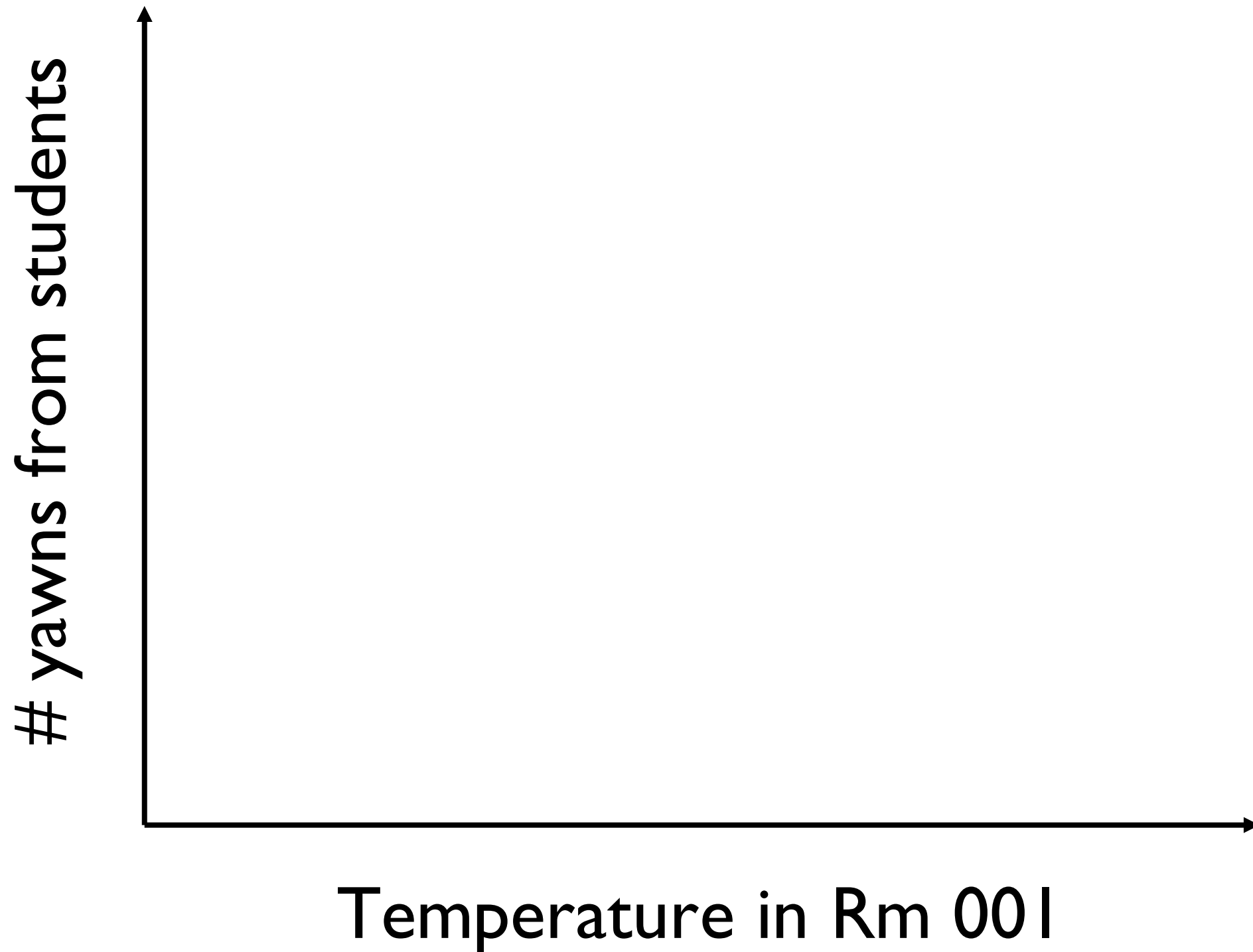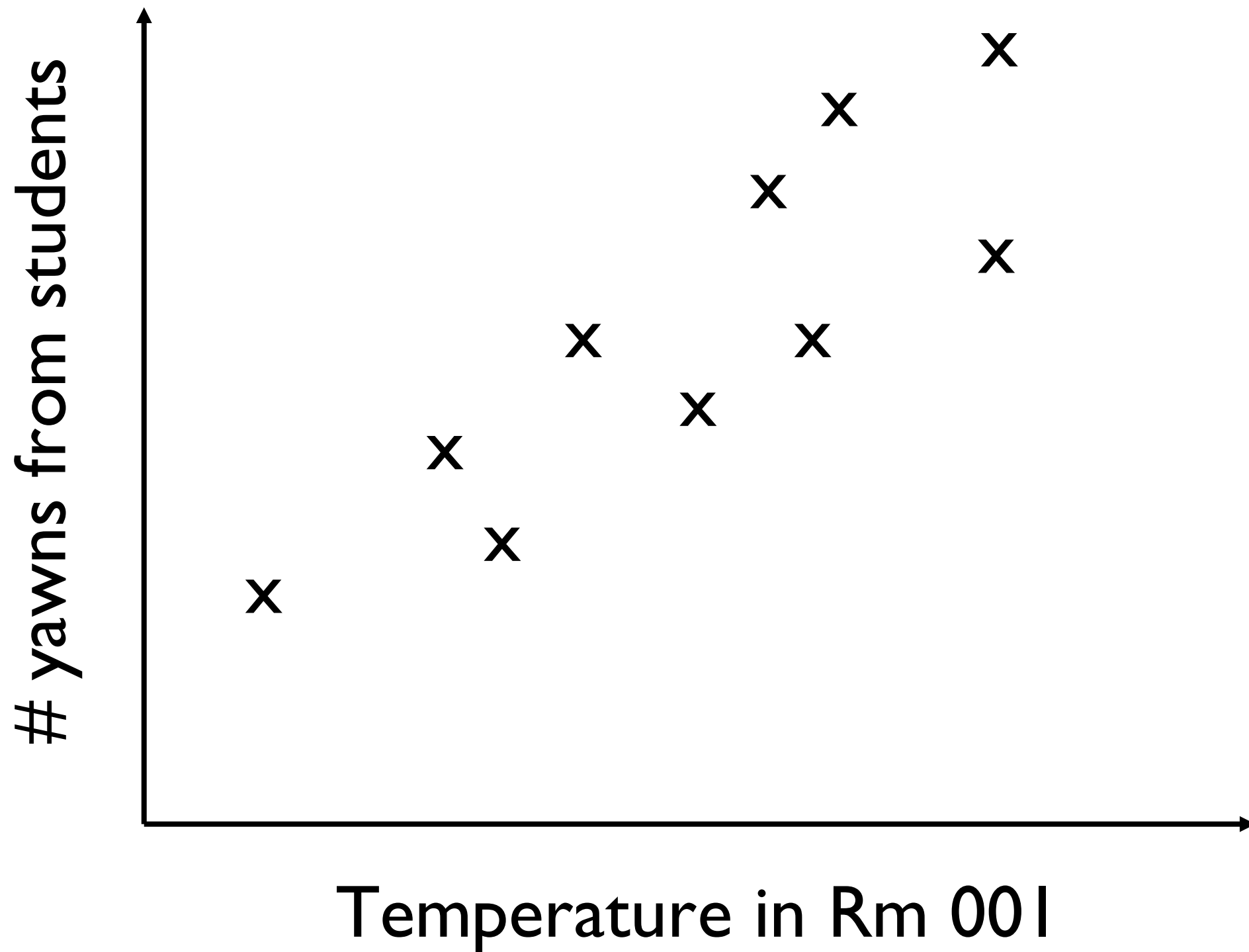
Linear Regression: Training

$$y = wx + b$$

- **Input:** set of *m* training examples (x,y)

- Find the value of *w* and *b* that minimize the error:

$$\sum_{i=1}^{m} \left(y^{(i)} - \hat{y}^{(i)} \right)^2$$

Linear Regression: Training

$$y = wx + b$$

- Find the value of *w* and *b* that minimize the error:

$$\sum_{i=1}^{m} \left(y^{(i)} - \hat{y}^{(i)} \right)^2$$

$$\sum_{i=1}^{m} \left(y^{(i)} - wx^{(i)} - b \right)^2$$

Linear Regression: Training

$$y = wx + b$$

- Find the value of *w* and *b* that minimize the error:

$$\sum_{i=1}^{m} \left(y^{(i)} - wx^{(i)} - b \right)^2$$

- Take the derivative with respect to *w*, set it equal to 0, and solve for *w*.

- Take the derivative with respect to *b*, set it equal to 0, and solve for *b*.

Linear Regression: Training

- Find the value of *w* and *b* that minimize the error:

$$w = \frac{\frac{1}{m} \sum_{i=1}^{m} \left(x^{(i)} - \bar{x}\right)\left(y^{(i)} - \bar{y}\right)}{\sum_{i=1}^{m} \left(x^{(i)} - \bar{x}\right)^2}$$

$$b = \bar{y} - w\bar{x}$$

http://seismo.berkeley.edu/~kirchner/eps_120/Toolkits/Toolkit_10.pdf

Linear Regression: Training

- Find the value of w and b that minimize the error:

$$w = \frac{\frac{1}{m} \sum_{i=1}^{m} \left(x^{(i)} - \bar{x} \right) \left(y^{(i)} - \bar{y} \right)}{\sum_{i=1}^{m} \left(x^{(i)} - \bar{x} \right)^2}$$

Always positive!

It depends!

$$b = \bar{y} - w\bar{x}$$

Linear Regression: Prediction

$$y = wx + b$$
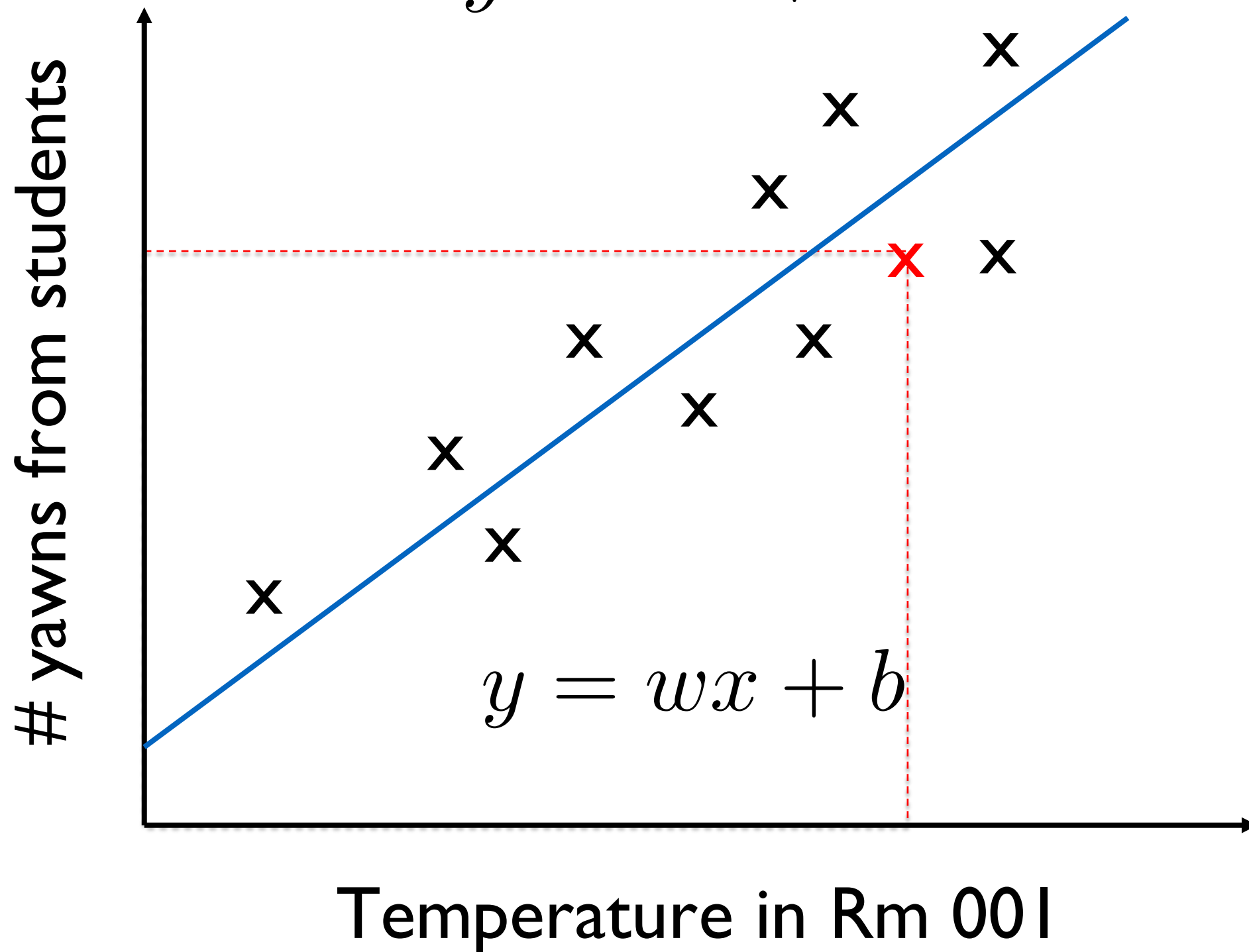
$$y = wx + b$$

yawns from students

Temperature in Rm 001

Overview

- Philosophical questions

- Derivatives: What are they good for?

- Linear regression

- *Multiple linear regression*

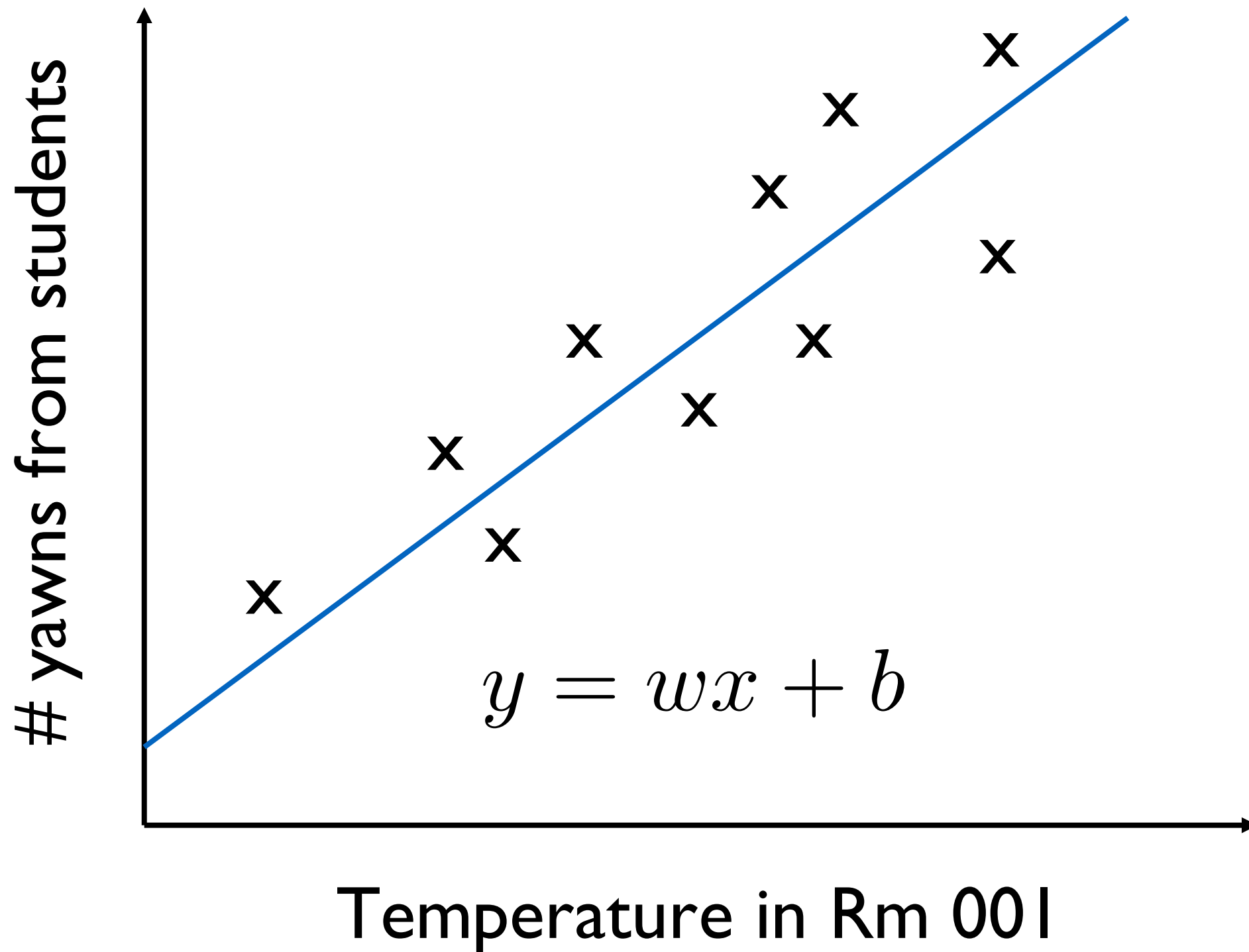- Logistic regression

Multiple Linear Regression

$$x_1$$
$$x_2$$
$$x_3$$
$$\vdots$$
$$x_n$$

$$f(x_1, x_2, \ldots, x_n)$$

$$y$$
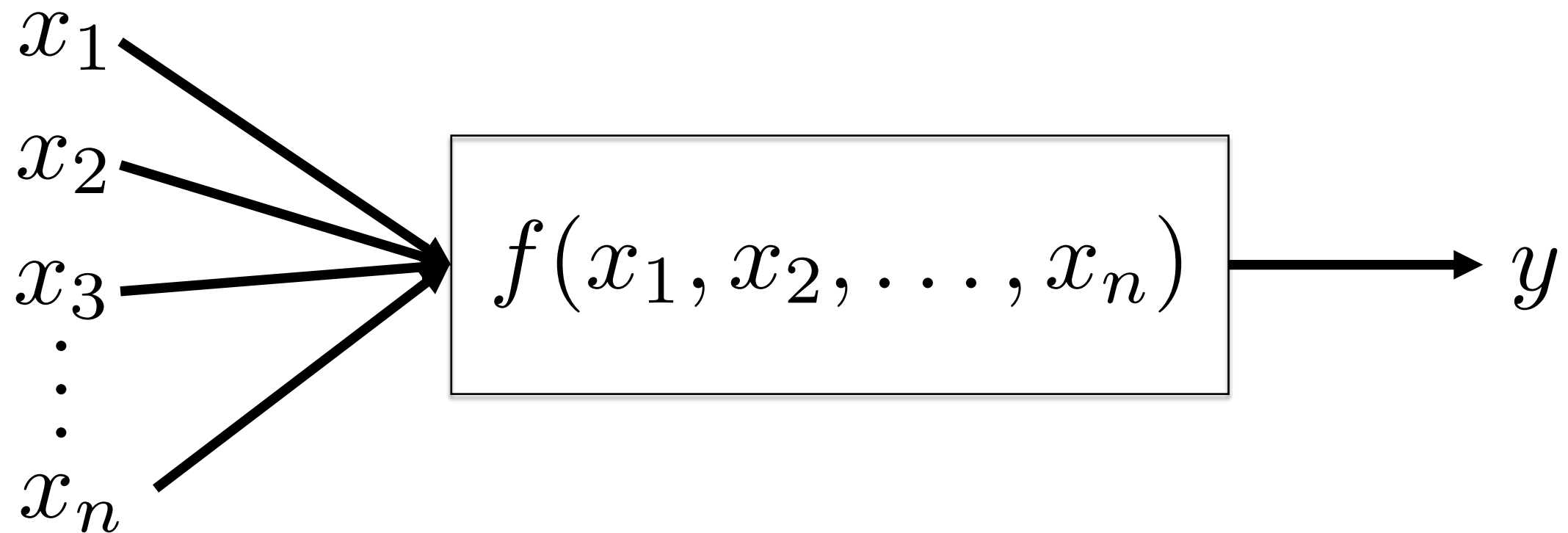
$$y = \sum_{j=1}^{n} (w_j x_j) + b$$

Multiple Linear Regression

Size (feet)	No. of bedrooms	No. of floors	Age (years)	Price (x$1000)
2,350	5	2	45	500
1,600	3	2	20	450
2,000	3	2	30	250
854	2	1	10	200
560	1	1	30	180

Multiple Linear Regression: Training

- Given:

$$\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})\}$$

- We want:

$$\hat{y}^{(i)} \approx y^{(i)}$$

Multiple Linear Regression: Training

- Loss Function: the discrepancy between the predicted and actual output values for <u>a single</u> training instance

$$\mathcal{L}(y^{(i)}, \hat{y}^{(i)}) = \frac{1}{2}(y^{(i)} - \hat{y}^{(i)})^2$$

Multiple Linear Regression: Training

- Cost Function: the discrepancy between the predicted and actual output values for <u>all</u> training instances

$$\mathcal{L}(y^{(i)}, \hat{y}^{(i)}) = \frac{1}{2}(y^{(i)} - \hat{y}^{(i)})^2$$

$$\mathcal{J}(w, b) = \frac{1}{m}\sum_{i=1}^{m}\left(\frac{1}{2}(y^{(i)} - \hat{y}^{(i)})^2\right)$$

Gradient Descent: Intuition

$$\frac{d}{x}\left(x^2\right) = 2x$$

Gradient Descent: Intuition

Gradient Descent
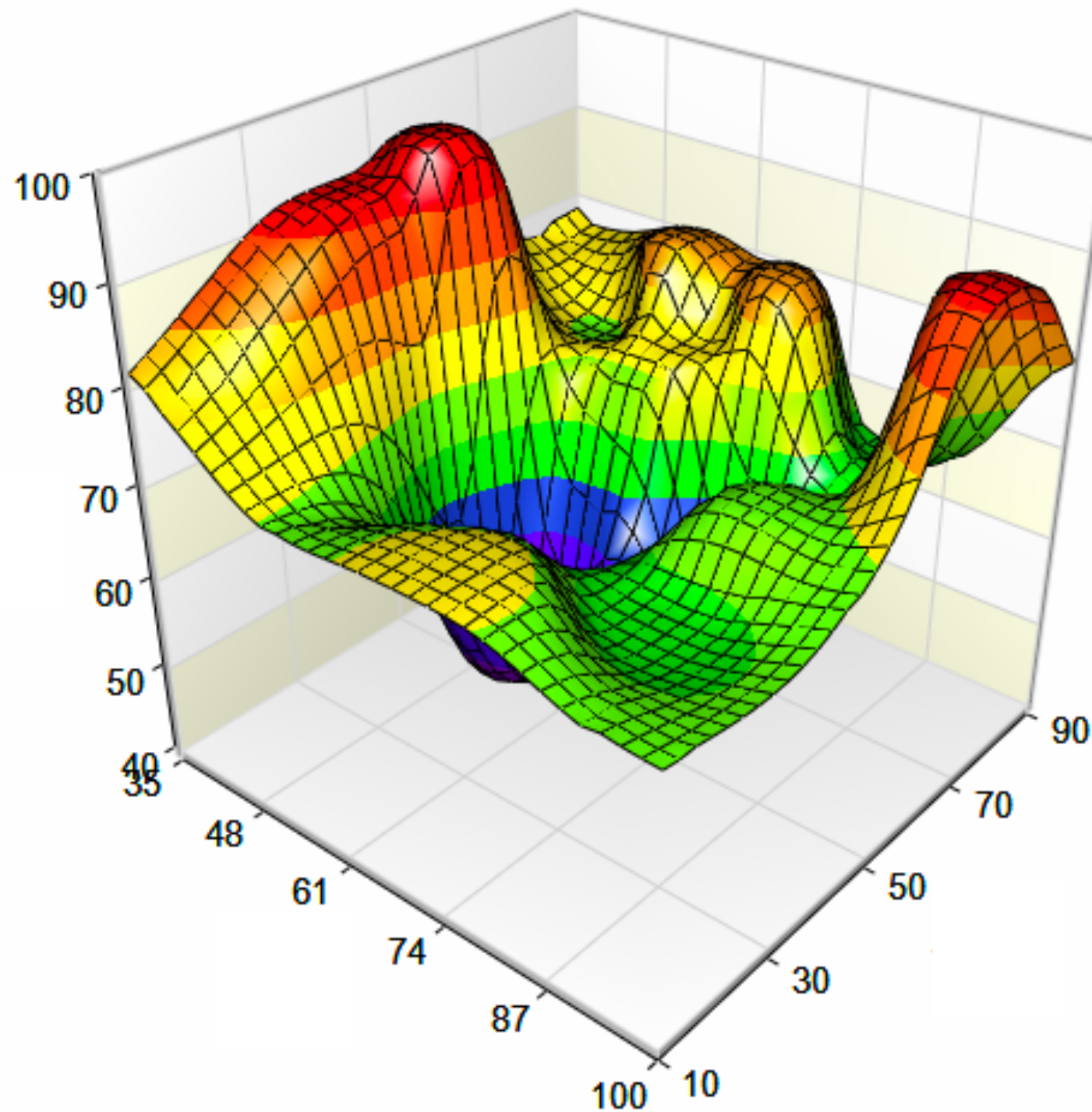
- Loss Function: the discrepancy between the predicted and actual output values for <u>a single</u> training instance

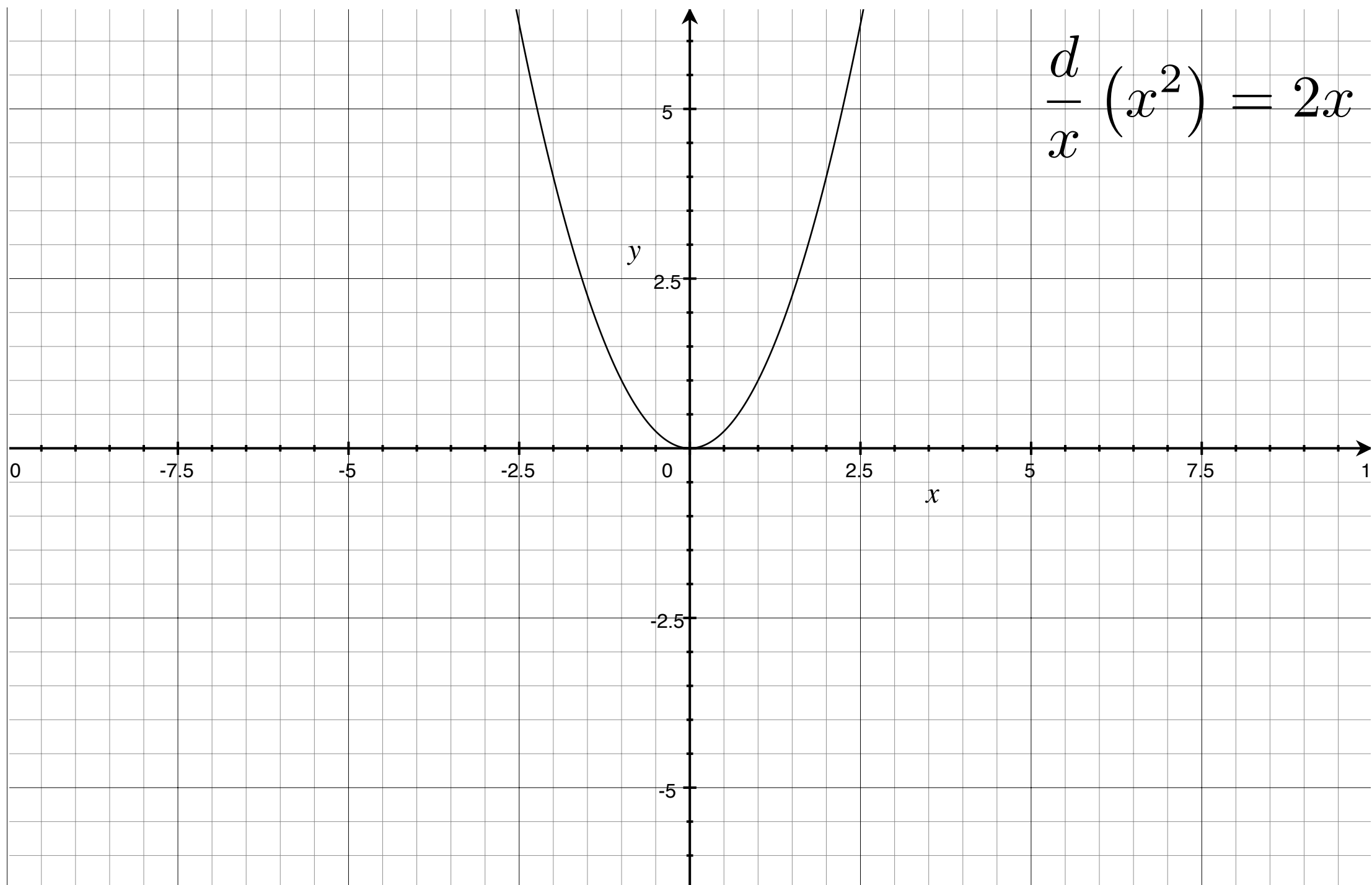$$\mathcal{L}(y^{(i)}, \hat{y}^{(i)}) = \frac{1}{2}(y^{(i)} - \hat{y}^{(i)})^2$$

- Let's see what the slope of the loss function is with respect to parameter b!

- **Note:** this will only consider one training example!

Gradient Descent

- Derivative of the loss function with respect to *b*

$$\mathcal{L}(y^{(i)}, \hat{y}^{(i)}) = \frac{1}{2}(y^{(i)} - \hat{y}^{(i)})^2$$

$$\mathcal{L}(y^{(i)}, \hat{y}^{(i)}) = \frac{1}{2}\left(y^{(i)} - \sum_{j=1}^{n}(w_j x_j^{(i)}) - b\right)^2$$

$$\frac{d}{db}\mathcal{L}(y^{(i)}, \hat{y}^{(i)}) = -\left(y^{(i)} - \hat{y}^{(i)}\right)$$

$$\frac{d}{db}\mathcal{L}(y^{(i)}, \hat{y}^{(i)}) = \hat{y}^{(i)} - y^{(i)}$$

Gradient Descent

$$\frac{d}{db}\mathcal{L}(y^{(i)}, \hat{y}^{(i)}) = \hat{y}^{(i)} - y^{(i)}$$

Scenario	$\hat{y}^{(i)} - y^{(i)}$	Action!
$\hat{y}^{(i)} > y^{(i)}$	+	Decrease (nudge left)
$\hat{y}^{(i)} < y^{(i)}$	--	Increase (nudge right)
$\hat{y}^{(i)} \approx y^{(i)}$	0	Do nothing!

$$y = \sum_{j=1}^{n} (w_j x_j) + b$$

Gradient Descent

- Loss Function: the discrepancy between the predicted and actual output values for <u>a single</u> training instance

$$\mathcal{L}(y^{(i)}, \hat{y}^{(i)}) = \frac{1}{2}(y^{(i)} - \hat{y}^{(i)})^2$$

- Let's see what the slope of the loss function is with respect to parameter w_j!

- **Note:** this will only consider one training example!

Gradient Descent

- Derivative of the loss function with respect to w_j

$$\mathcal{L}(y^{(i)}, \hat{y}^{(i)}) = \frac{1}{2}(y^{(i)} - \hat{y}^{(i)})^2$$

$$\mathcal{L}(y^{(i)}, \hat{y}^{(i)}) = \frac{1}{2}\left(y^{(i)} - \sum_{j=1}^{n}(w_j x_j^{(i)}) - b\right)^2$$

$$\frac{d}{dw_j}\mathcal{L}(y^{(i)}, \hat{y}^{(i)}) = -x_j^{(i)}\left(y^{(i)} - \hat{y}^{(i)}\right)$$

$$\frac{d}{dw_j}\mathcal{L}(y^{(i)}, \hat{y}^{(i)}) = (\hat{y}^{(i)} - y^{(i)})x_j$$

Gradient Descent

$$\frac{d}{dw_j}\mathcal{L}(y^{(i)}, \hat{y}^{(i)}) = (\hat{y}^{(i)} - y^{(i)})x_j$$

Scenario	$\hat{y}^{(i)} - y^{(i)}$	Action!
$\hat{y}^{(i)} > y^{(i)}$	+	Go in the opposite direction as $x_j^{(i)}$
$\hat{y}^{(i)} < y^{(i)}$	-	Go in the same direction as $x_j^{(i)}$
$\hat{y}^{(i)} \approx y^{(i)}$	0	Do nothing!

$$y = \sum_{j=1}^{n}(w_j x_j) + b$$

Gradient Descent

- Loss Function: the discrepancy between the predicted and actual output values for <u>a single</u> training instance

$$\mathcal{L}(y^{(i)}, \hat{y}^{(i)}) = \frac{1}{2}(y^{(i)} - \hat{y}^{(i)})^2$$

- Given one training example, we can take derivatives with respect to each parameter to see what direction we should be going to minimize the loss function.

Gradient Descent

- Repeat many times (or until convergence):

$$b \leftarrow b - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(\hat{y}^{(i)} - y^{(i)} \right)$$

$$w_j \leftarrow w_j - \alpha \frac{1}{m} \sum_{i=1}^{m} \left((\hat{y}^{(i)} - y^{(i)}) x_j^{(i)} \right)$$

Gradient Descent

- Repeat many times (or until convergence):

$$b \leftarrow b - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(\hat{y}^{(i)} - y^{(i)} \right)$$

- If we are overshooting the target, reduce b

- If we are undershooting the target, increase b

- Otherwise, do nothing

Gradient Descent

- Repeat many times (or until convergence):

$$w_j \leftarrow w_j - \alpha \frac{1}{m} \sum_{i=1}^{m} \left((\hat{y}^{(i)} - y^{(i)}) x_j^{(i)} \right)$$

- If we are overshooting the target, reduce w_j proportional to the value of x_j

- If we are undershooting the target, increase w_j proportional to the value of x_j

- Otherwise, do nothing

Overview

- Philosophical questions

- Derivatives: What are they good for?

- Linear regression

- Multiple linear regression
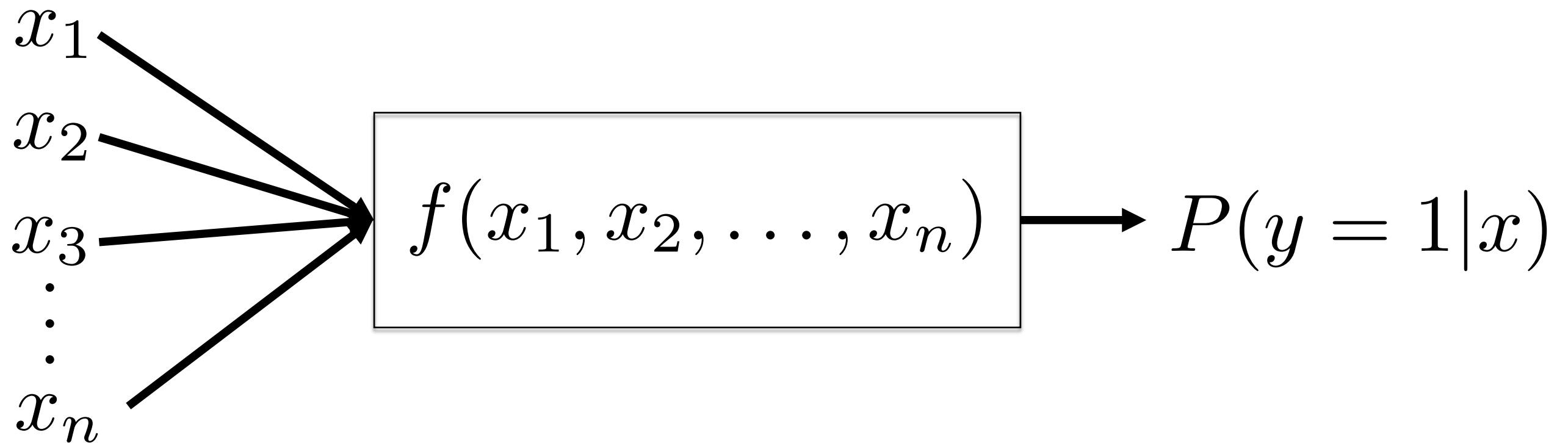
- **Logistic regression**

Logistic Regression

- Linear regression: predict *y* given *x*

- *Multiple linear regression:* predict *y* given *x_1, x_2, …, x_n*

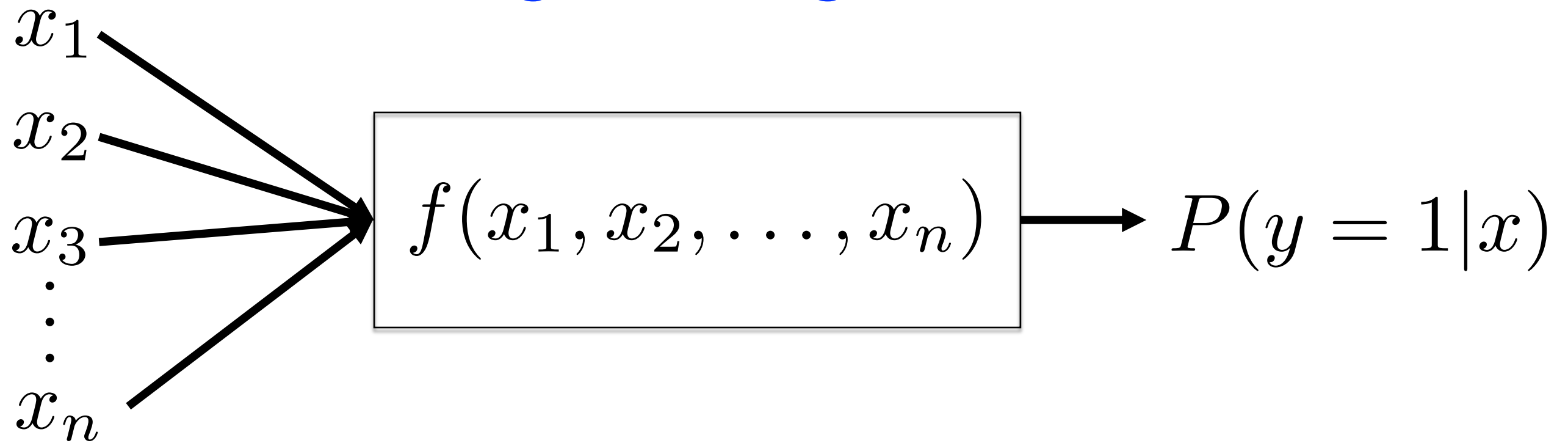# Logistic Regression

- Logistic Regression: predict P(y=1|x_1, x_2, …, x_n)

- We can use logistic regression to do binary classification.

Logistic Regression

$$x_1$$
$$x_2$$
$$x_3$$
$$\vdots$$
$$x_n$$

$$f(x_1, x_2, \ldots, x_n)$$

$$P(y = 1|x)$$

Logistic Regression

$$x_1$$
$$x_2$$
$$x_3$$
$$\vdots$$
$$x_n$$

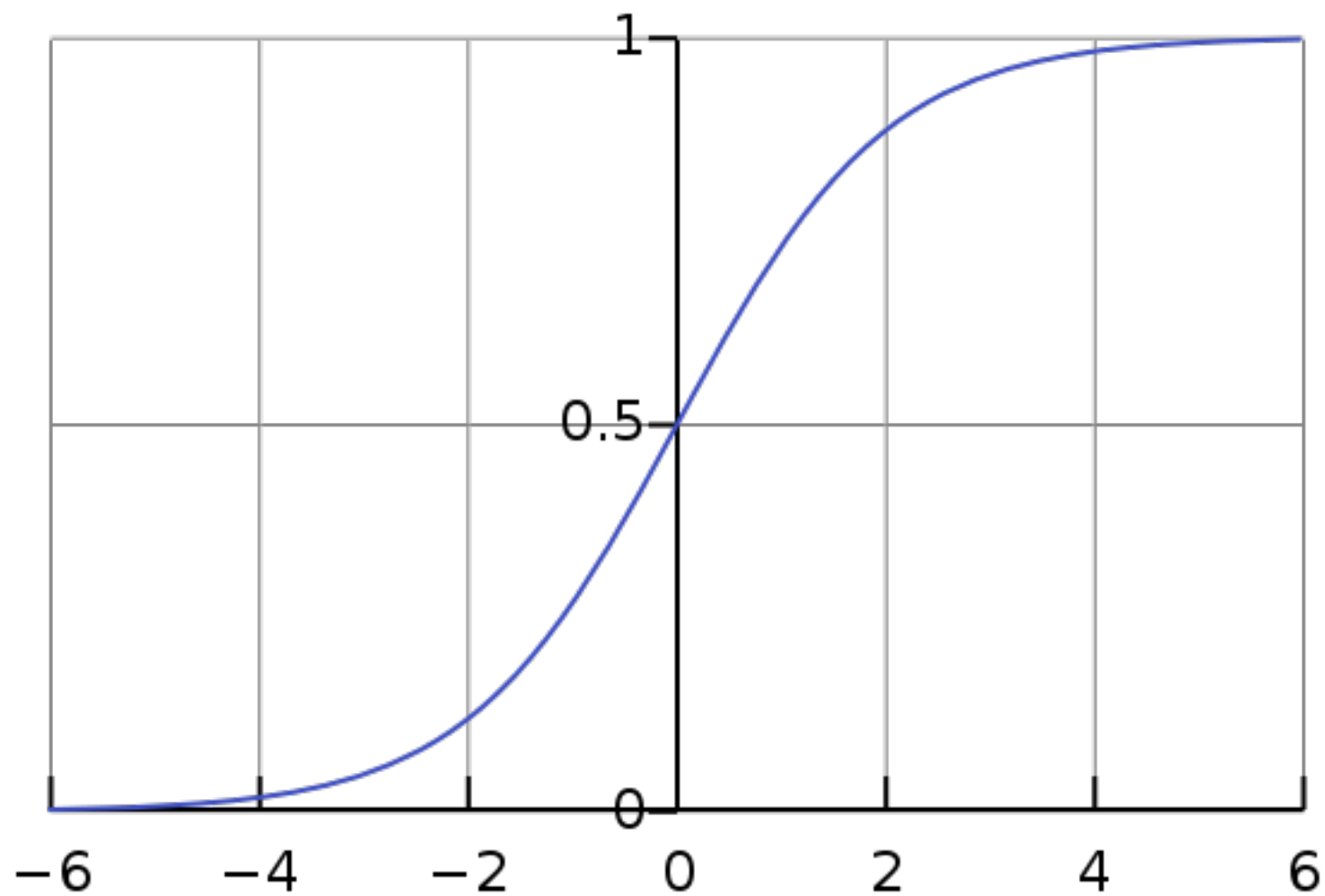$$f(x_1, x_2, \ldots, x_n) \longrightarrow P(y = 1 | x)$$

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

$$z = \sum_{j=1}^{n} (w_j x_j) + b$$

Logistic Regression

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

Logistic Regression

- Loss Function: the discrepancy between the predicted and actual output values for <u>a single</u> training instance
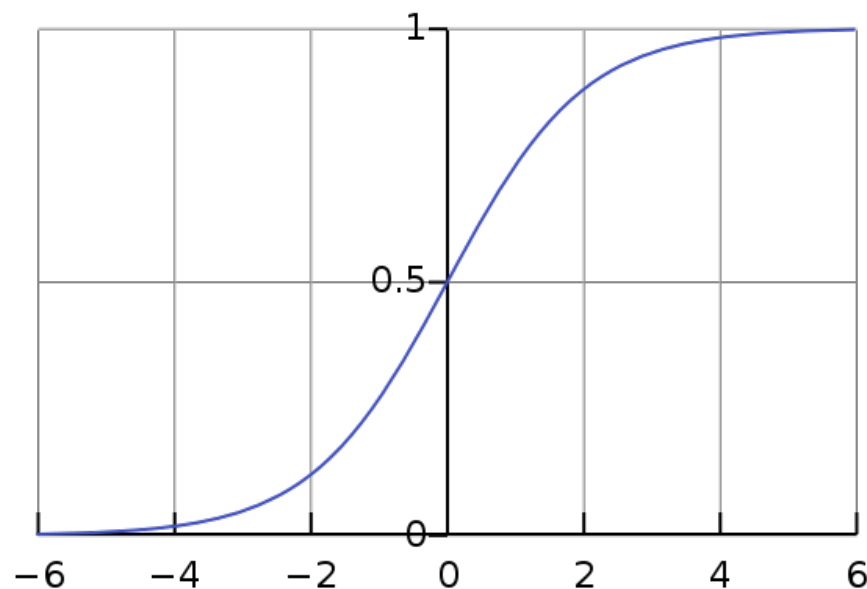
$$z = \sum_{j=1}^{n} (w_j x_j) + b$$

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

$$\mathcal{L}(y^{(i)}, \hat{y}^{(i)}) = - \left(y^{(i)} \log \hat{y}^{(i)} + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)}) \right)$$

Logistic Regression

- Loss Function: the discrepancy between the predicted and actual output values for <u>a single</u> training instance

$$\mathcal{L}(y^{(i)}, \hat{y}^{(i)}) = -\left(y^{(i)} \log \hat{y}^{(i)} + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)}) \right)$$

- If the true value is 1, we want the predicted value to be <u>high</u>.

- Remember: $\log(1) = 0$

Logistic Regression

- Loss Function: the discrepancy between the predicted and actual output values for <u>a single</u> training instance

$$\mathcal{L}(y^{(i)}, \hat{y}^{(i)}) = -\left(y^{(i)} \log \hat{y}^{(i)} + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)})\right)$$
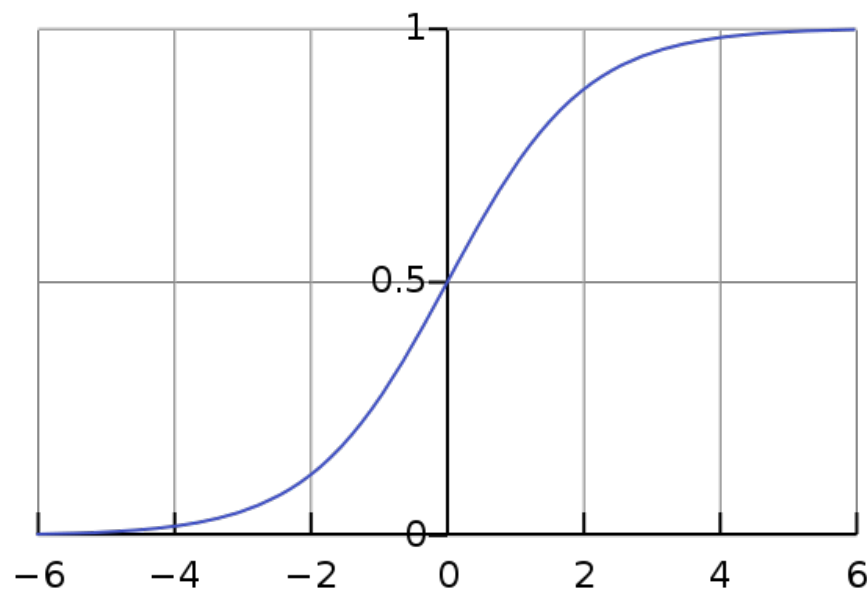

- If the true value is 0, we want the predicted value to be <u>low</u>.

- Remember: $\log(1) = 0$

Logistic Regression

- Loss Function: the discrepancy between the predicted and actual output values for <u>a single</u> training instance

$$z = \sum_{j=1}^{n} (w_j x_j) + b$$

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

$$\mathcal{L}(y^{(i)}, \hat{y}^{(i)}) = -\left(y^{(i)} \log \hat{y}^{(i)} + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)}) \right)$$

$$\frac{d}{db} \mathcal{L}(y^{(i)}, \hat{y}^{(i)}) = \hat{y}^{(i)} - y^{(i)}$$

Logistic Regression

- Loss Function: the discrepancy between the predicted and actual output values for <u>a single</u> training instance

$$z = \sum_{j=1}^{n} (w_j x_j) + b$$

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

$$\mathcal{L}(y^{(i)}, \hat{y}^{(i)}) = -\left(y^{(i)} \log \hat{y}^{(i)} + (1 - y^{(i)}) \log(1 - \hat{y}^{(i)}) \right)$$

$$\frac{d}{dw_j} \mathcal{L}(y^{(i)}, \hat{y}^{(i)}) = (\hat{y}^{(i)} - y^{(i)}) x_j$$

Logistic Regression:
Gradient Descent

- Repeat many times (or until convergence):

$$b \leftarrow b - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(\hat{y}^{(i)} - y^{(i)} \right)$$

$$w_j \leftarrow w_j - \alpha \frac{1}{m} \sum_{i=1}^{m} \left((\hat{y}^{(i)} - y^{(i)}) x_j^{(i)} \right)$$

Logistic Regression:
Gradient Descent

- Repeat many times (or until convergence):

$$b \leftarrow b - \alpha \frac{1}{m} \sum_{i=1}^{m} \left(\hat{y}^{(i)} - y^{(i)} \right)$$

- If we are overshooting the target, reduce b

- If we are undershooting the target, increase b

- Otherwise, do nothing

Logistic Regression:
Gradient Descent

- Repeat many times (or until convergence):

$$w_j \leftarrow w_j - \alpha \frac{1}{m} \sum_{i=1}^{m} \left((\hat{y}^{(i)} - y^{(i)}) x_j^{(i)} \right)$$

- If we are overshooting the target, reduce w_j proportional to the value of x_j

- If we are undershooting the target, increase w_j proportional to the value of x_j

- Otherwise, do nothing

Overview

- Philosophical questions

- Derivatives: What are they good for?

- Linear regression

- Multiple linear regression

- Logistic regression

The Big Picture!

- Linear regression, multiple linear regression and logistic regression are examples of linear models

- Internally, linear models output a prediction based on a weighted combination of input features

- Features that are <u>positively correlated</u> with the target output get a <u>high</u> weight

- Features that are <u>negatively correlated</u> with the target output get a <u>low</u> weight

- Features that are <u>uncorrelated</u> with the target output get a <u>zero</u> weight