Introduction to Machine Learning Algorithms: Linear Regression, Logistic Regression and K-Nearest Neighbors

Panthadeep Bhattacharjee
Dept. of CSE, NIT Rourkela

panthadeep.edu@gmail.com

What is Machine Learning?

"Learning is any process by which a system improves performance from experience."

- Herbert Simon

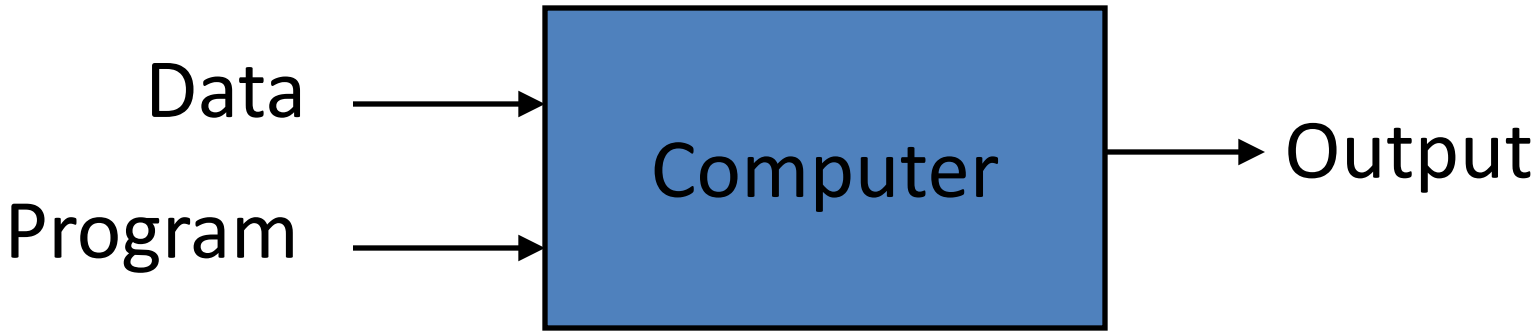
Definition by Tom Mitchell (1998):

Machine Learning is the study of algorithms that

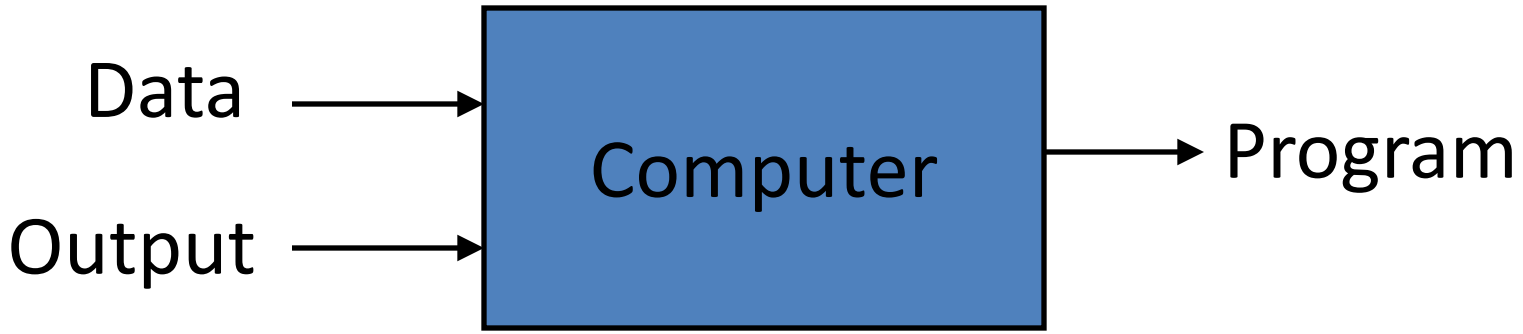
- improve their performance P
- at some task T
- with experience E.

A well-defined learning task is given by <P, T, E>.

Traditional Programming



Machine Learning



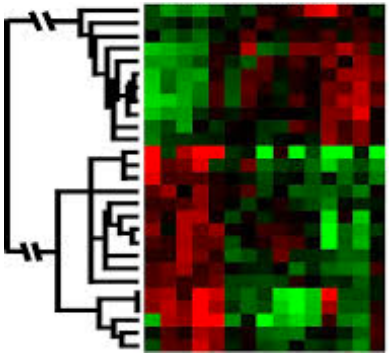
Slide credit: Pedro Domingos

When Do We Use Machine Learning?

ML is used when:

- Human expertise does not exist (navigating on Mars)
- Humans can't explain their expertise (speech recognition)
- Models must be customized (personalized medicine)
- Models are based on huge amounts of data (genomics)



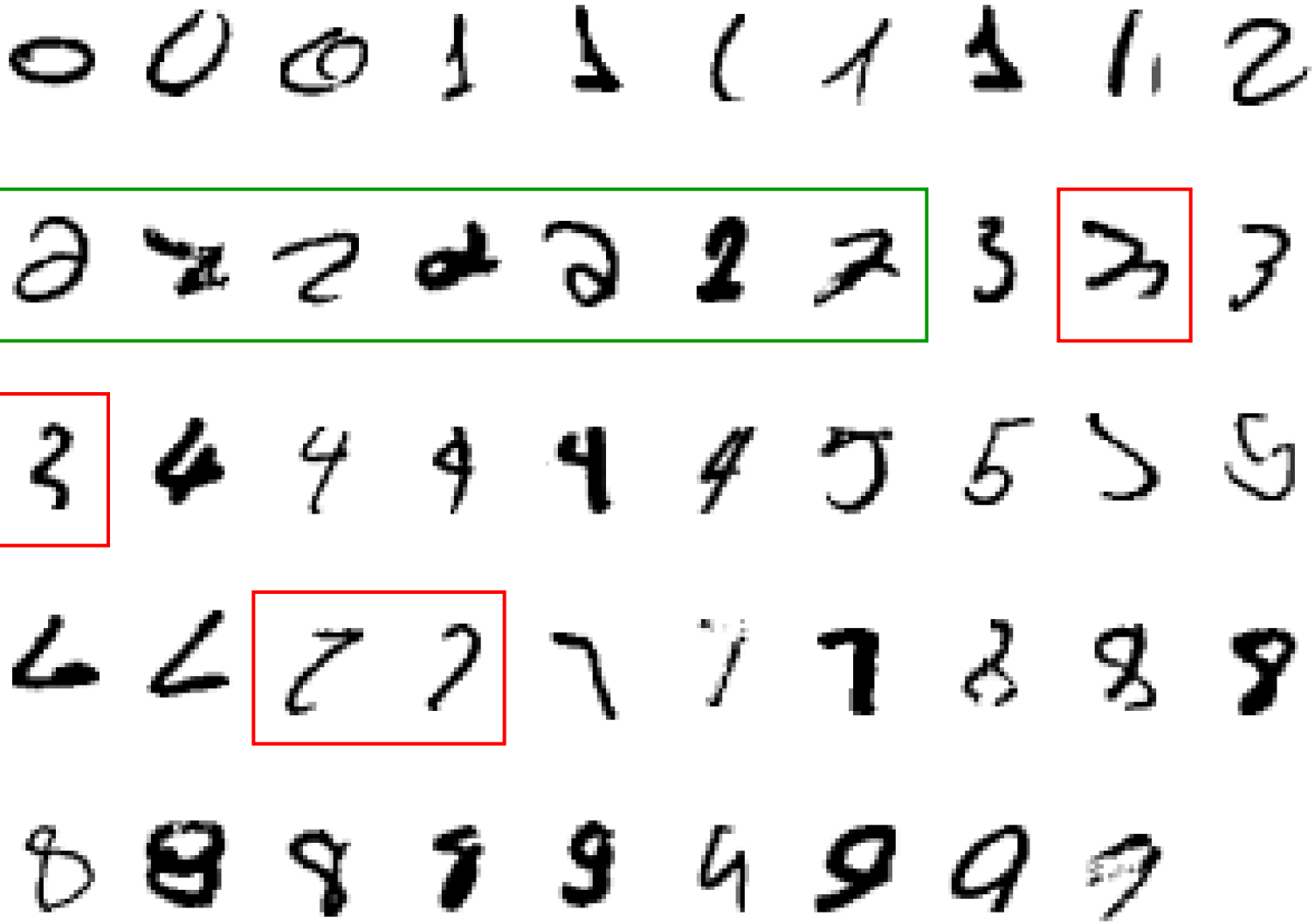
Learning isn't always useful:

- There is no need to "learn" to calculate payroll

Based on slide by E. Alpaydin

A classic example of a task that requires machine learning:

It is very hard to say what makes a 2



Slide credit: Geoffrey Hinton

Some more examples of tasks that are best solved by using a learning algorithm

- Recognizing patterns:
  - Facial identities or facial expressions
  - Handwritten or spoken words
  - Medical images
- Generating patterns:
  - Generating images or motion sequences
- Recognizing anomalies:
  - Unusual credit card transactions
  - Unusual patterns of sensor readings in a nuclear power plant
- Prediction:
  - Future stock prices or currency exchange rates

Slide credit: Geoffrey Hinton

Sample Applications

- Web search
- Computational biology
- Finance
- E-commerce
- Space exploration
- Robotics
- Information extraction
- Social networks
- Debugging software
- [Your favorite area]

Slide credit: Pedro Domingos

Samuel's Checkers-Player

"Machine Learning: Field of study that gives computers the ability to learn without being explicitly programmed." -Arthur Samuel (1959)

Defining the Learning Task

Improve on task T, with respect to
performance metric P, based on experience E

T: Playing checkers
P: Percentage of games won against an arbitrary opponent
E: Playing practice games against itself

T: Recognizing hand-written words
P: Percentage of words correctly classified
E: Database of human-labeled images of handwritten words

T: Driving on four-lane highways using vision sensors
P: Average distance traveled before a human-judged error
E: A sequence of images and steering commands recorded while observing a human driver.

T: Categorize email messages as spam or legitimate.
P: Percentage of email messages correctly classified.
E: Database of emails, some with human-given labels

Slide credit: Ray Mooney

Linear Regression

Objectives

- What is machine learning
- Types of data and terminology
- Types of machine learning
- Supervised learning
- Linear regression
- Least square and Gradient Descent
- Hands on implementing Linear Regression from sketch.

Machine Learning

- Machine Learning is the science to make computers learn from data without explicitly program them and improve their learning over time in autonomous fashion.

- This learning comes by feeding them **data** in the form of observations and real-world interactions.”

- Machine Learning can also be defined as a tool to **predict** future events or values using past **data**.

Types of Data

- ***Based on Values***
  - *Continuous data (ex. Age – 0-100)*
  - *Categorical data (ex. Gender- Male/Female)*
- ***Based on pattern***
  - *Structured data (ex. Databases)*
  - *Unstructured data (ex. Audio, Video, Text)*

Types of Data- continued

- ***Labelled data*** *– consists of input output pair. For every set input features the output/response/label is present in dataset. (ex- labelled image as cat's or dog's photo)*

$$\{(x_1, y_1), (x_2, y_2), (x_3, y_3) \dots \dots \dots \dots \dots (x_n, y_n)\}$$

- ***Unlabelled data****- There is no output/response/label for the input features in data. (ex. news articles, tweets, audio)*

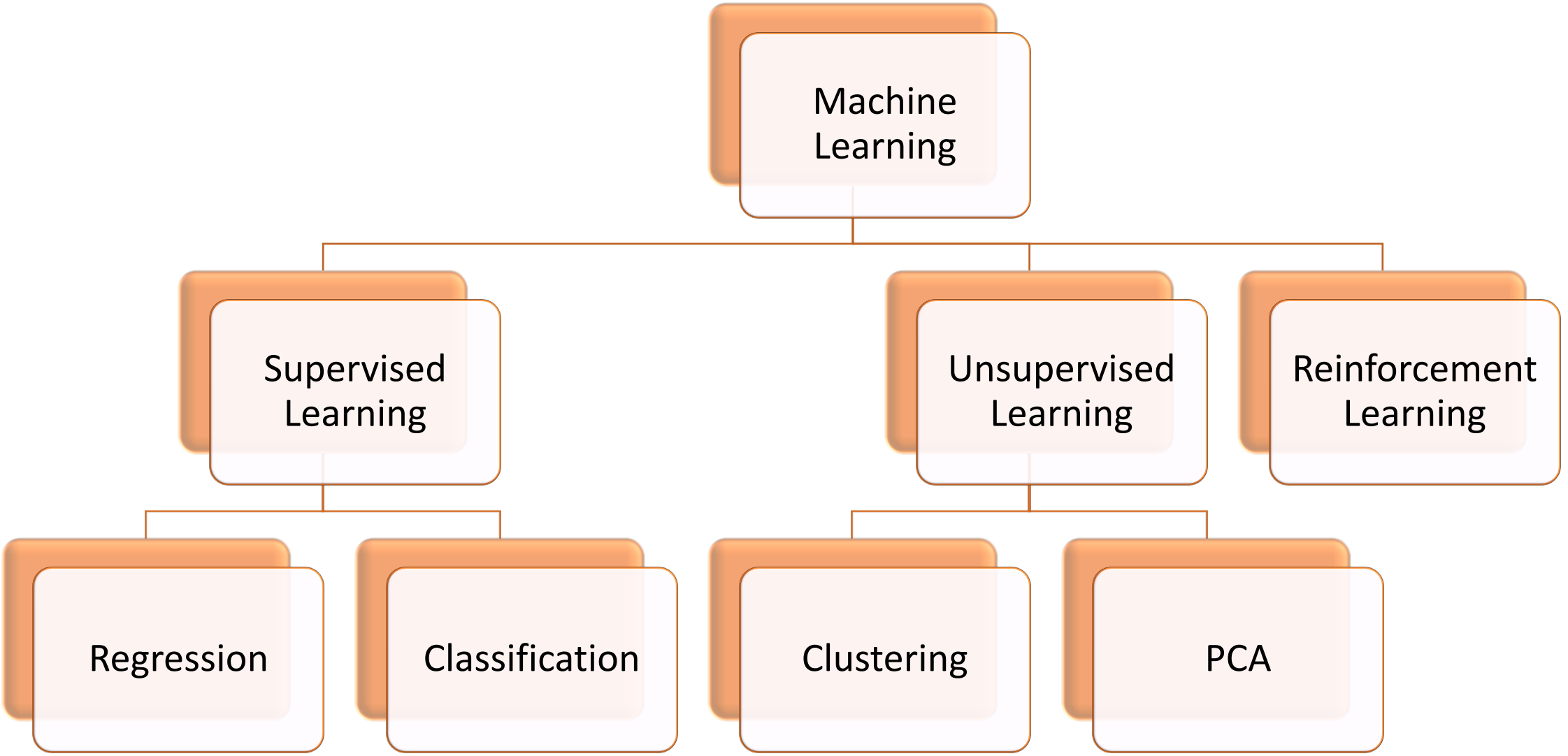
$$\{x_1, x_2, x_3 \dots \dots \dots \dots x_n\}$$

Types of Data- continued

- ***Training Data*** *– Sample data points which are used to train the machine learning model.*
- ***Test Data****- sample data points that are used to test the performance of machine learning model.*

Note- For modelling, the original dataset is partitioned into the ratio of 70:30 or 75:25 as training data and test data.

Types of Machine Learning

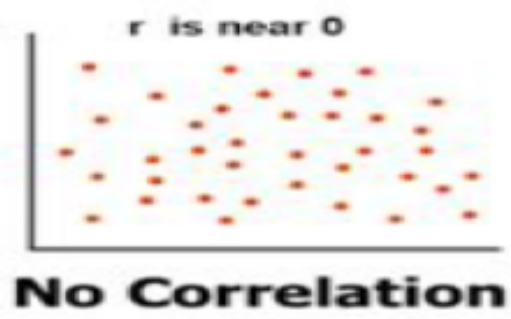


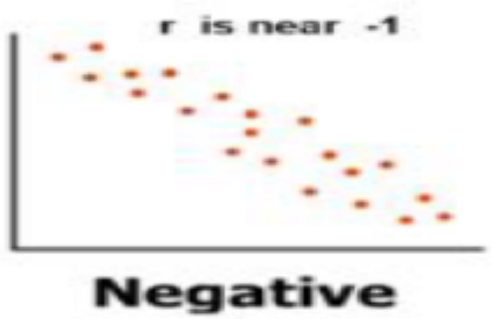
Supervised Learning

- Class of machine learning that work on externally supplied instances in form of predictor attributes and **associated target values**.
- The model learns from the training data using these '**target variables'** as reference variables.
  - Ex1 : *model to predict the resale value of a car based on its mileage, age, color etc.*
- The **target values** are the 'correct answers' for the predictor model which can either be a **regression model** or a **classification model.**

Motivation for learning

- It is being assumed that there exists a relationship/association between **input features** and **target variable.**
- Relationship can be observed by plotting a scatter plot between the two variables.



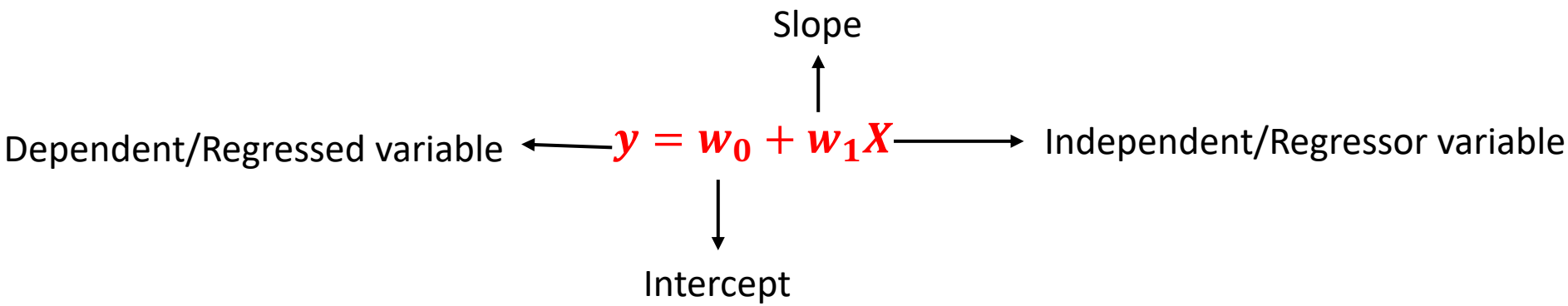


- Relationship measure can be quantified by calculating correlation between two the variables.

$$corr(x,y) = \frac{cov(x,y)}{var(x).var(y)} = \frac{\sum(x_i - \bar{x}) * (y_i - \bar{y})}{\sqrt{\sum(x_i - \bar{x})^2}\sqrt{\sum(y_i - \bar{y})^2}}$$

Linear Regression

- Linear regression is a way to identify a relationship between two or more variables and use these relationships to predict values of one variable for given value(s) of other variable(s).
- Linear regression assume the relationship between variables can be modelled through linear equation or an equation of line



Multiple Regression

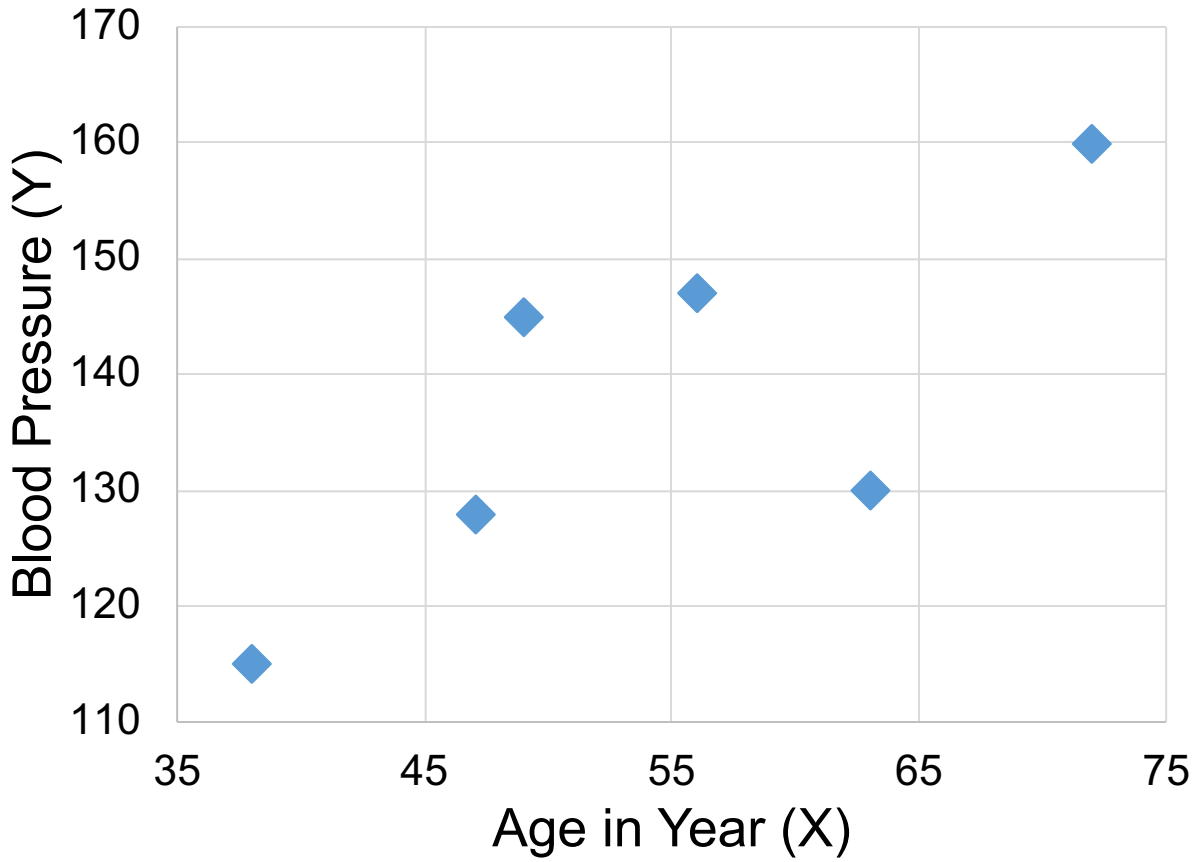
- Last slide showed the linear regression model with one independent and one dependent variable.
- In Real world a data point has various important attributes and they need to be catered to while developing a regression model. (Many independent variables and one dependent variable)

$$y = w_0 + w_1x_1 + w_2x_2 + w_3x_3 \ldots\ldots\ldots\ldots \mathrm{w_n x_n}$$

Regression –Problem Formulation

Let you have given with a data:

Age in Years **(X)**	Blood Pressure **(Y)**
56	147
49	145
72	160
38	115
63	130
47	128



Linear Regression

❖ For given example the Linear Regression is modeled as:

$$BloodPressure(y) = w_0 + w_1 AgeinYear(X)$$

OR

$$y = w_0 + w_1 X$$ – Equation of line

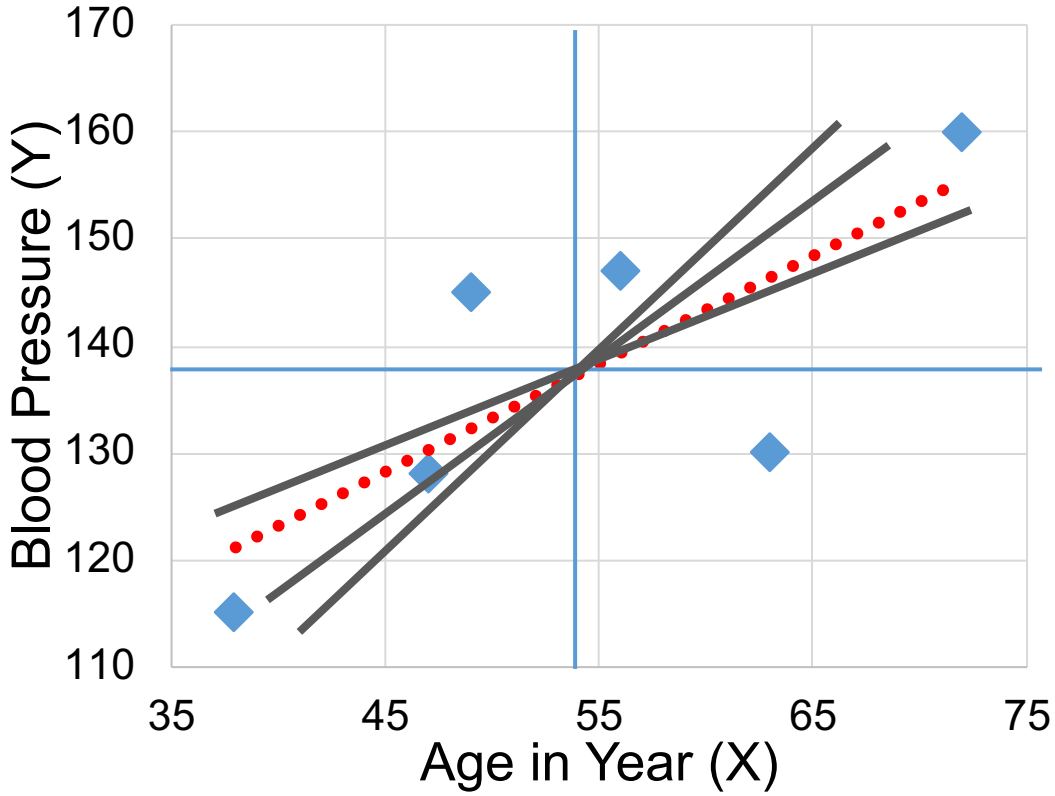
with w_0 $is\ intercept\ on\ Y\_axis\ and\ w_1$ $is\ slope\ of\ line$

Blood Pressure - **Dependent Variable**

Age in Year - **Independent Variable**

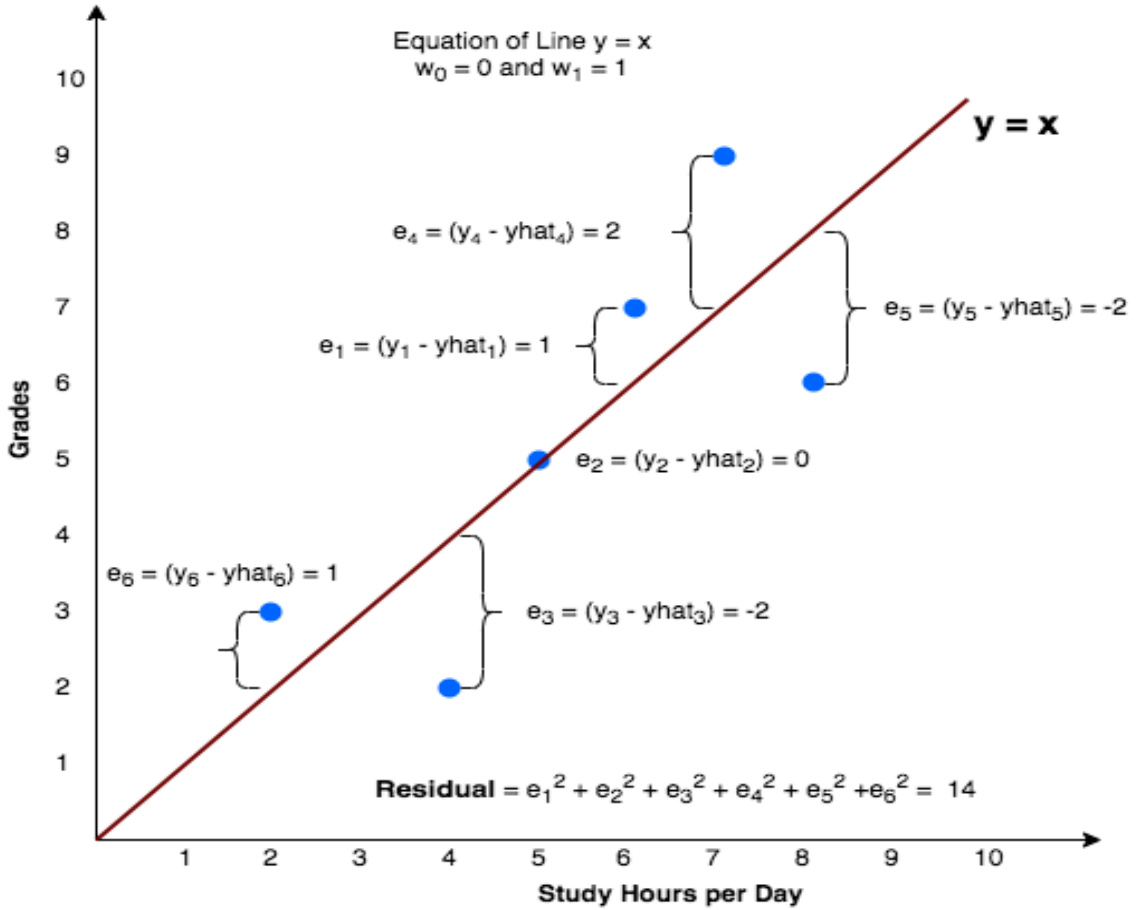
Linear Regression- Best Fit Line

- Regression uses line to show the trend of distribution.
- There can be many lines that try to fit the data points in scatter diagram
- The aim is to find **Best fit** Line



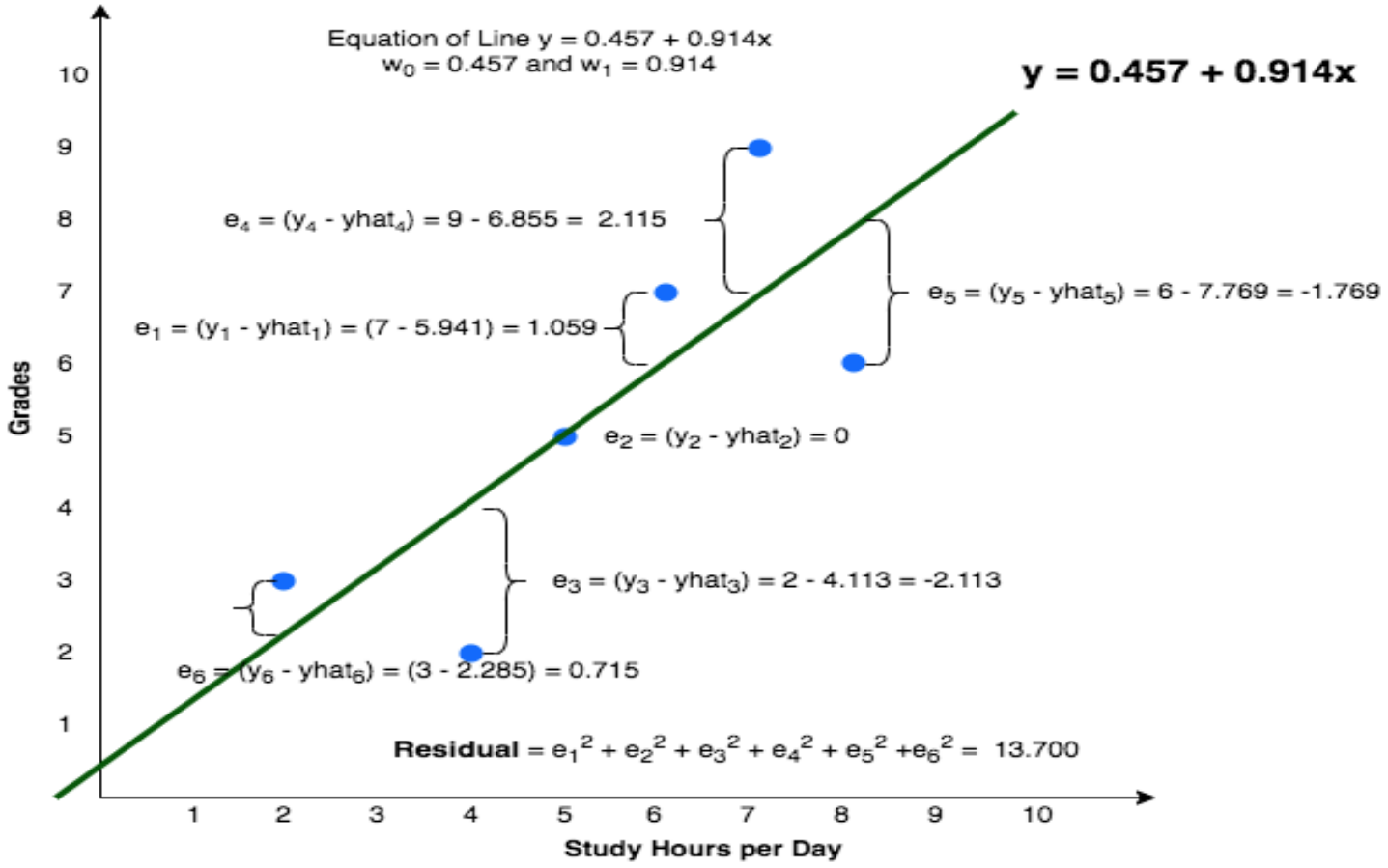
What is Best Fit Line

- Best fit line tries to explain the variance in given data. (minimize the total residual/error)



What is Best Fit Line

- Best fit line tries to explain the variance in given data. (minimize the total residual/error)



Linear Regression- Methods to Get Best

- Least Square
- Gradient Descent

Linear Regression- Least Square

Model: $Y = w_0 + w_1 X$

Task: Estimate $the\ value\ of\ w_0\ and\ w_1$

According to pr$inciple\ of\ least\ square\ the\ normal\ equations\ to\ solve\ for\ w_0\ and\ w_1$

$$\sum_{i=1}^{n} Y_i = n\, w_0 + w_1 \sum_{i=1}^{n} X_i \ldots\ldots\ldots\ldots\ldots (1)$$

$$\sum_{i=1}^{n} X_i Y_i = w_0 \sum_{i=1}^{n} X_i + w_1 \sum_{i=1}^{n} X_i^2 \ldots\ldots\ldots\ldots . (2)$$

Linear Regression–Least Square

Let divide the equation (1) by *n* (number of sample points) we get:

$$\frac{1}{n}\sum_{i=1}^{n} Y_i = w_0 + w_1 \frac{1}{n}\sum_{i=1}^{n} X_i$$

OR

$$\bar{y} = w_0 + w_1 \bar{x} \ldots\ldots\ldots\ldots (3)$$

So line of regression will always passes through the points $(\bar{x}, \bar{y})$

Linear Regression–Least Square

Now we know :

$$cov(x,y) = \frac{1}{n}\sum_{i=1}^{n} x_i y_i - \bar{x}\bar{y} := \frac{1}{n}\sum_{i=1}^{n} x_i y_i = cov(x,y) + \bar{x}\bar{y} \ldots\ldots\ldots(4)$$

and

$$var(x) = \frac{1}{n}\sum_{i=1}^{n} x_i^2 - \bar{x}^2 \quad \text{and} \quad var(y) = \frac{1}{n}\sum_{i=1}^{n} y_i^2 - \bar{y}^2$$

Dividing equation (2) by n and using equation (4) and (5) we get:

$$cov(x,y) + \bar{x}\bar{y} = w_0\bar{x} + w_1(var(x) + \bar{x}^2) \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots(5)$$

Linear Regression–Least Square

Now by using equation

$$\bar{y} = w_0 + w_1\bar{x}$$

and

$$cov(x, y) + \bar{x}\bar{y} = w_0\bar{x} + w_1(var(x) + \bar{x}^2)$$

We will get:

$$w_1 = \frac{cov(x, y)}{var(x)}$$

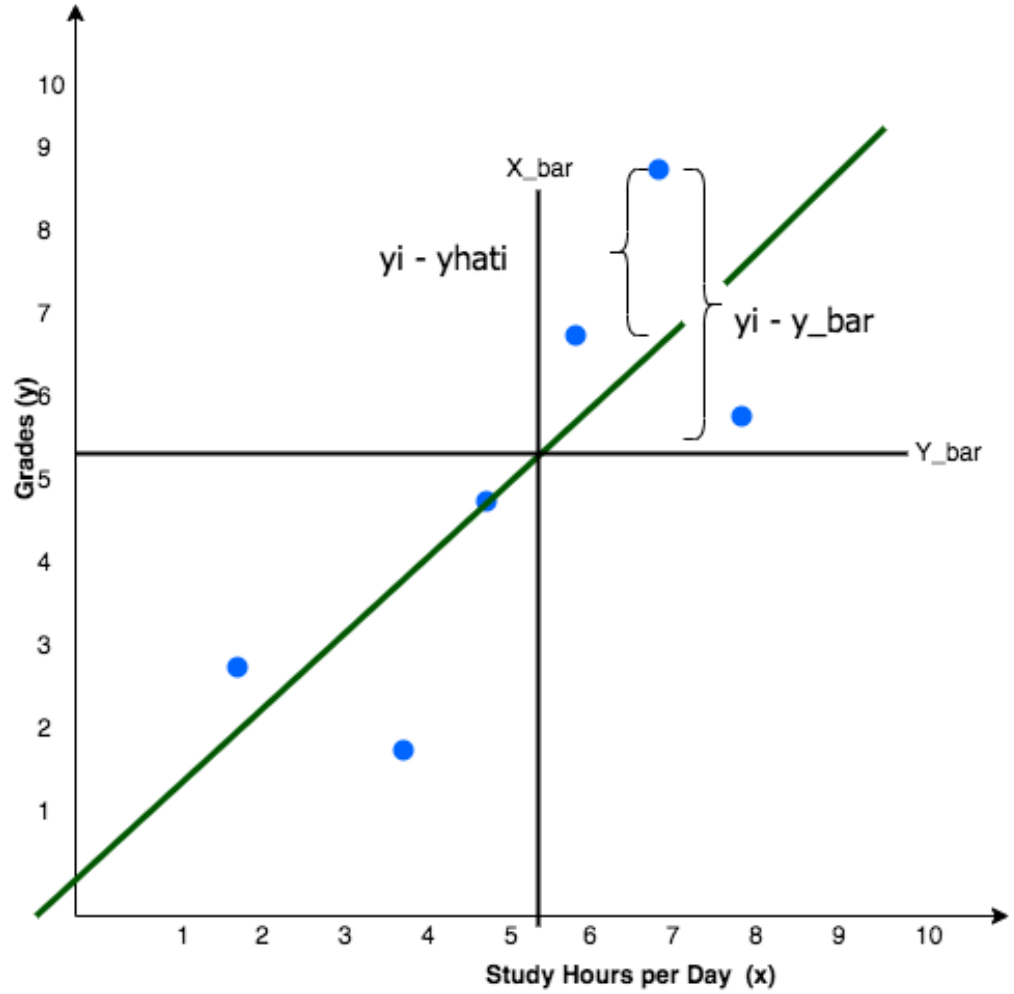
and

$$w_0 = \bar{y} - w_1\bar{x}$$

Performance metric for least square regression

$$R^2 = 1 - \frac{\frac{1}{n}\sum_{i=1}^{n}(y_i - yhat_i)^2}{\frac{1}{n}\sum_{i=1}^{n}(y_i - \bar{y})^2}$$

$$R^2_{adj} = 1 - \frac{(1 - R^2)(n-1)}{(n - k - 1)}$$



Linear Regression- Gradient Descent

Model: $Y = w_0 + w_1 X$

Task: Estimate $the\ value\ of\ w_0\ and\ w_1$

Define the cost function,

$$cost(w_0, w_1) = \frac{1}{n}\sum_{i=1}^{n}(y_i - yhat_i)^2$$

Objective of gradient Descent

$$\min_{w_0, w_1} cost(w_0, w_1) = \frac{1}{n}\sum_{i=1}^{n}(y_i - (w_0 + w_1 x_i))^2$$

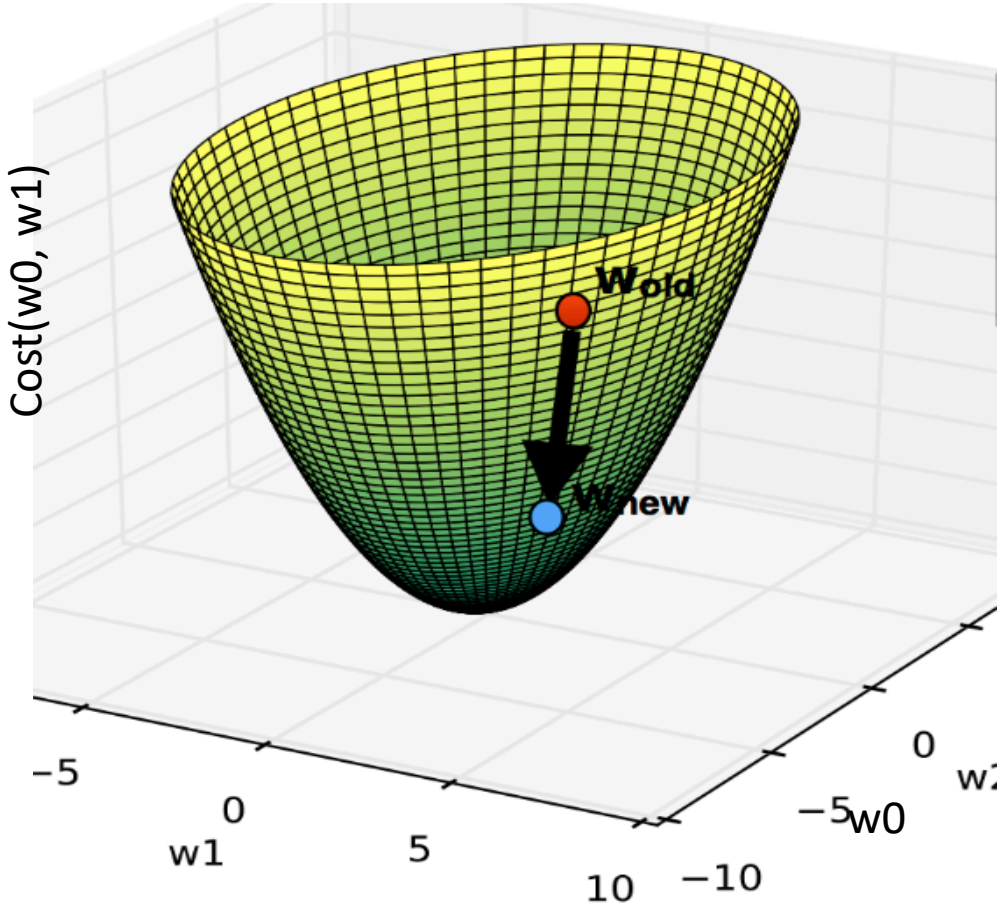
Linear Regression- Gradient Descent

Model: $Y = w_0 + w_1 X$

Task: Estimate $the\ value\ of\ w_0\ and\ w_1$

the objective,

$$\min_{w_0, w_1} cost(w_0, w_1) = \frac{1}{n}\sum_{i=1}^{n}(y_i - (w_0 + w_1 x_i))^2$$



Linear Regression- Gradient Descent

- Gradient descent works if following steps:

1. Initialize the parameters to some random variable
2. Calculate the gradient of cost function w. r. t. to parameters
3. Update the parameters using gradient in opposite direction.
4. Repeat step-2 and step-3 for some number of times or till it reaches to minimum cost value.

Linear Regression- Gradient Descent

$$cost(w_0, w_1) = \frac{1}{n}\sum_{i=1}^{n}(y_i - (w_0 + w_1 x_i))^2$$

Calculating gradients of cost function:

$$gradw_0 = \frac{\partial cost(w_0, w_1)}{\partial w_0} = \frac{2}{n}\sum_{i=1}^{n}(y_i - (w_0 + w_1 x_i))(-1)$$

$$gradw_1 = \frac{\partial cost(w_0, w_1)}{\partial w_1} = \frac{2}{n}\sum_{i=1}^{n}(y_i - (w_0 + w_1 x_i))(-x)$$

Parameter update:

$$w_0 = w_0 - learningrate * gradw_0$$

$$w_1 = w_1 - learningrate * gradw_1$$

Performance metric for gradient based regression

Root Mean Square Error (**RMSE**) is the standard deviation of prediction errors.

$$RMSE = \sqrt{\frac{(y_i - yhat_i)^2}{n}}$$

Mean absolute error (MAE) is a measure of difference between two variables.

$$MAE = \frac{|y_i - yhat_i|}{n}$$

Logistic Regression

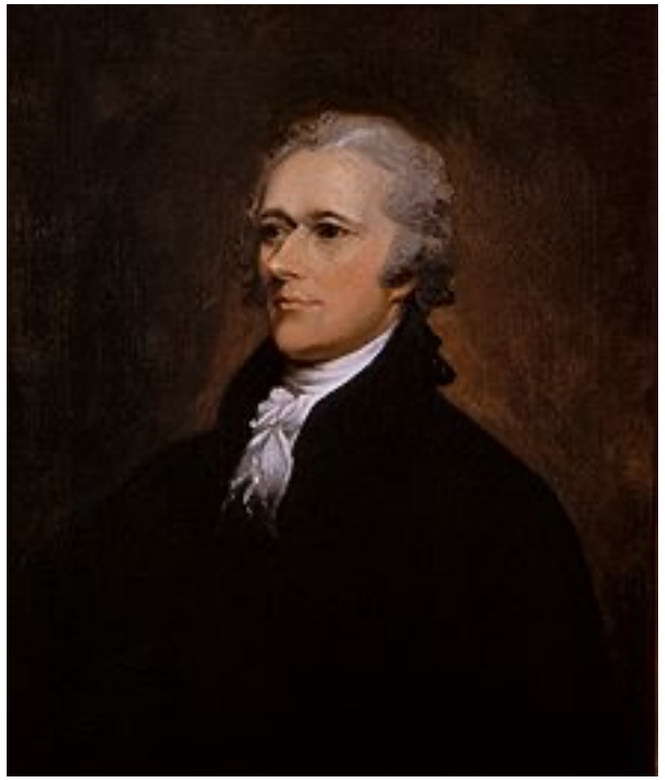
Classification in Logistic Regression

Classification Reminder

Positive/negative sentiment

Spam/not spam

Authorship attribution (Hamilton or Madison?)



Alexander Hamilton

Text Classification: definition

Input:

- a document x
- a fixed set of classes $C = \{c_1, c_2, \ldots, c_J\}$

Output: a predicted class $\hat{y} \in C$

Binary Classification in Logistic Regression

Given a series of input/output pairs:

- $(x^{(i)}, y^{(i)})$

For each observation $x^{(i)}$

- We represent $x^{(i)}$ by a **feature vector** $[x_1, x_2, \ldots, x_n]$
- We compute an output: a predicted class $\hat{y}^{(i)} \in \{0,1\}$

Features in logistic regression

- For feature x_i, weight w_i tells is how important is x_i
  - x_i ="review contains ‘awesome’": w_i = +10
  - x_j ="review contains ‘abysmal’": w_j = −10
  - x_k =“review contains ‘mediocre’": w_k = −2

Logistic Regression for one observation x

Input observation: vector $x = [x_1, x_2, \ldots, x_n]$

Weights: one per feature: $W = [w_1, w_2, \ldots, w_n]$

- Sometimes we call the weights $\theta = [\theta_1, \theta_2, \ldots, \theta_n]$

Output: a predicted class $\hat{y} \in \{0,1\}$

(multinomial logistic regression: $\hat{y} \in \{0, 1, 2, 3, 4\}$)

How to do classification

For each feature x_i, weight w_i tells us importance of x_i

- (Plus we'll have a bias b)

We'll sum up all the weighted features and the bias

$$z = \left(\sum_{i=1}^{n} w_i x_i \right) + b$$

$$z = w \cdot x + b$$

If this sum is high, we say y=1; if low, then y=0

But we want a probabilistic classifier

We need to formalize “sum is high”.

We’d like a principled classifier that gives us a probability, just like Naive Bayes did

We want a model that can tell us:

$p(y=1|x; \theta)$

$p(y=0|x; \theta)$

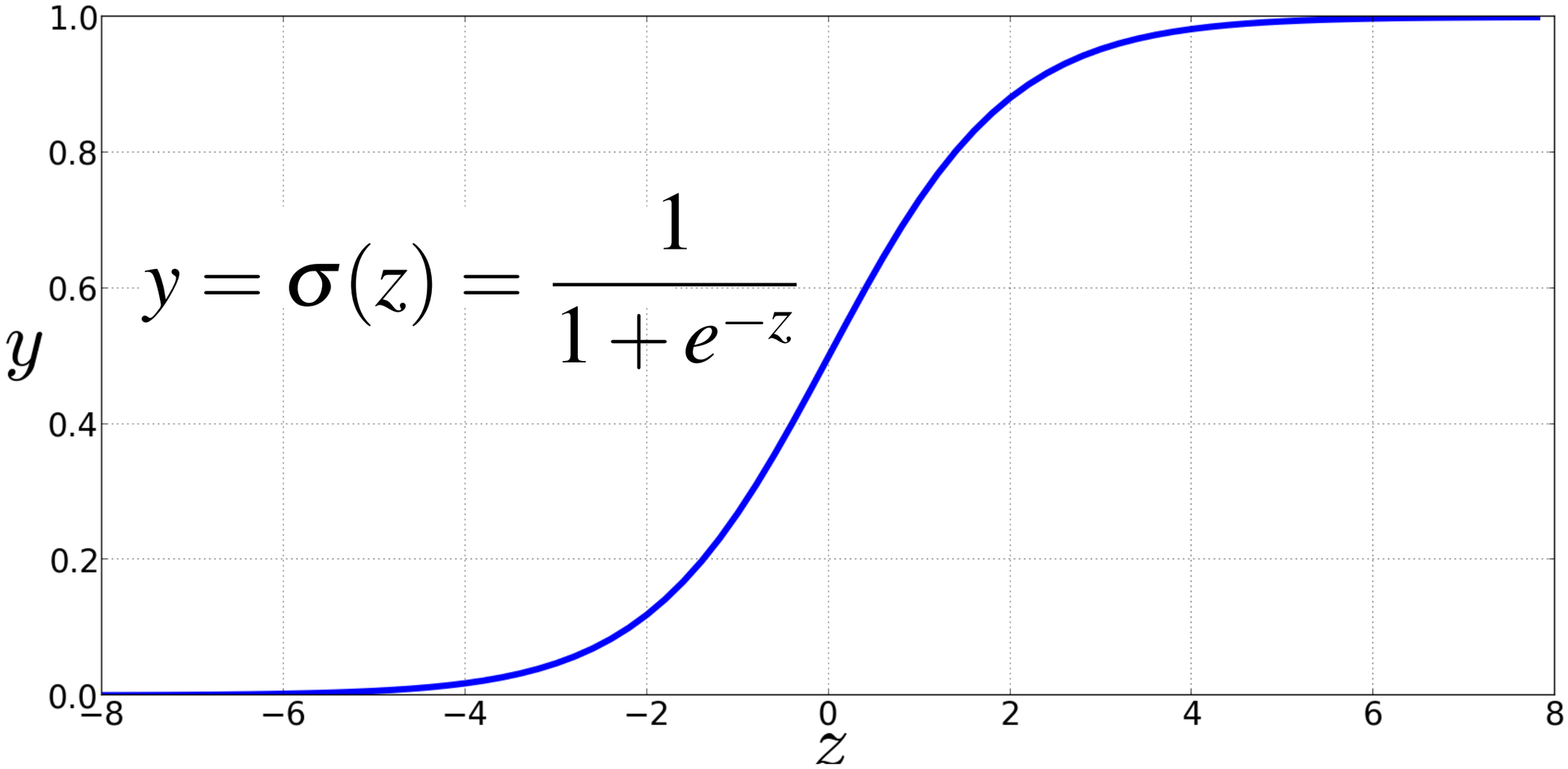
The problem: z isn't a probability, it's just a number!

$$z = w \cdot x + b$$

Solution: use a function of z that goes from 0 to 1

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

The very useful sigmoid or logistic function



Idea of logistic regression

We'll compute $w \cdot x + b$

And then we'll pass it through the sigmoid function:

$$\sigma(w \cdot x + b)$$

And we'll just treat it as a probability

Making probabilities with sigmoids

$$
\begin{aligned}
P(y=1) &= \sigma(w \cdot x + b) \\
&= \frac{1}{1+\exp(-(w \cdot x + b))}
\end{aligned}
$$

$$
\begin{aligned}
P(y=0) &= 1 - \sigma(w \cdot x + b) \\
&= 1 - \frac{1}{1+\exp(-(w \cdot x + b))} \\
&= \frac{\exp(-(w \cdot x + b))}{1+\exp(-(w \cdot x + b))}
\end{aligned}
$$

By the way:

$$
\begin{aligned}
P(y=0) &= 1-\sigma(w\cdot x+b) &&= \sigma(-(w\cdot x+b)) \\
&= 1-\frac{1}{1+\exp(-(w\cdot x+b))} && \text{Because} \\
&= \frac{\exp(-(w\cdot x+b))}{1+\exp(-(w\cdot x+b))} && 1-\sigma(x)=\sigma(-x)
\end{aligned}
$$

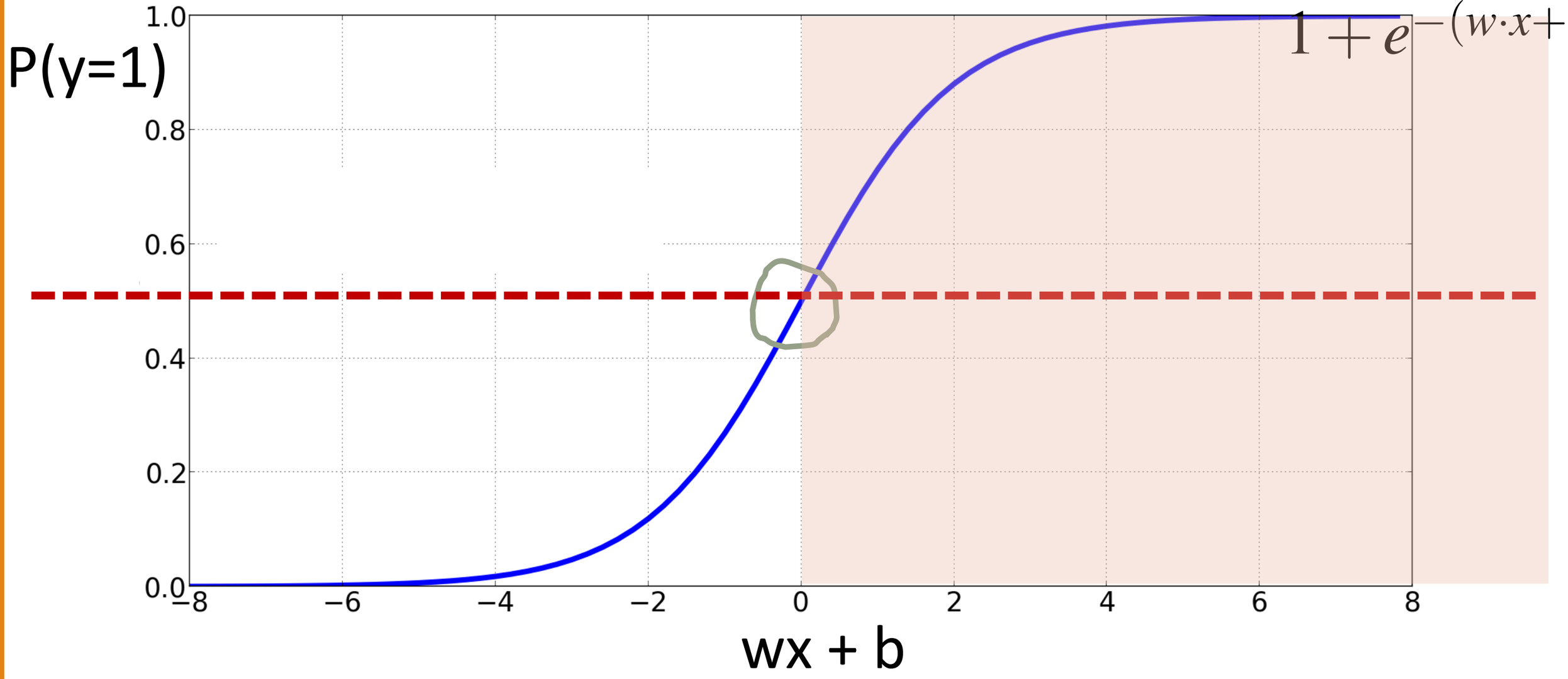
Turning a probability into a classifier

$$\hat{y} = \begin{cases} 1 & \text{if } P(y = 1|x) > 0.5 \\ 0 & \text{otherwise} \end{cases}$$

0.5 here is called the **decision boundary**

The probabilistic classifier $P(y=1) = \sigma(w \cdot x + b)$

$$= \frac{1}{1+e^{-(w \cdot x + b)}}$$



Turning a probability into a classifier

$$\hat{y} = \begin{cases} 1 & \text{if } P(y=1|x) > 0.5 \\ 0 & \text{otherwise} \end{cases}$$

$$\begin{array}{l} \text{if } w \cdot x + b > 0 \\ \text{if } w \cdot x + b \leq 0 \end{array}$$

Logistic Regression

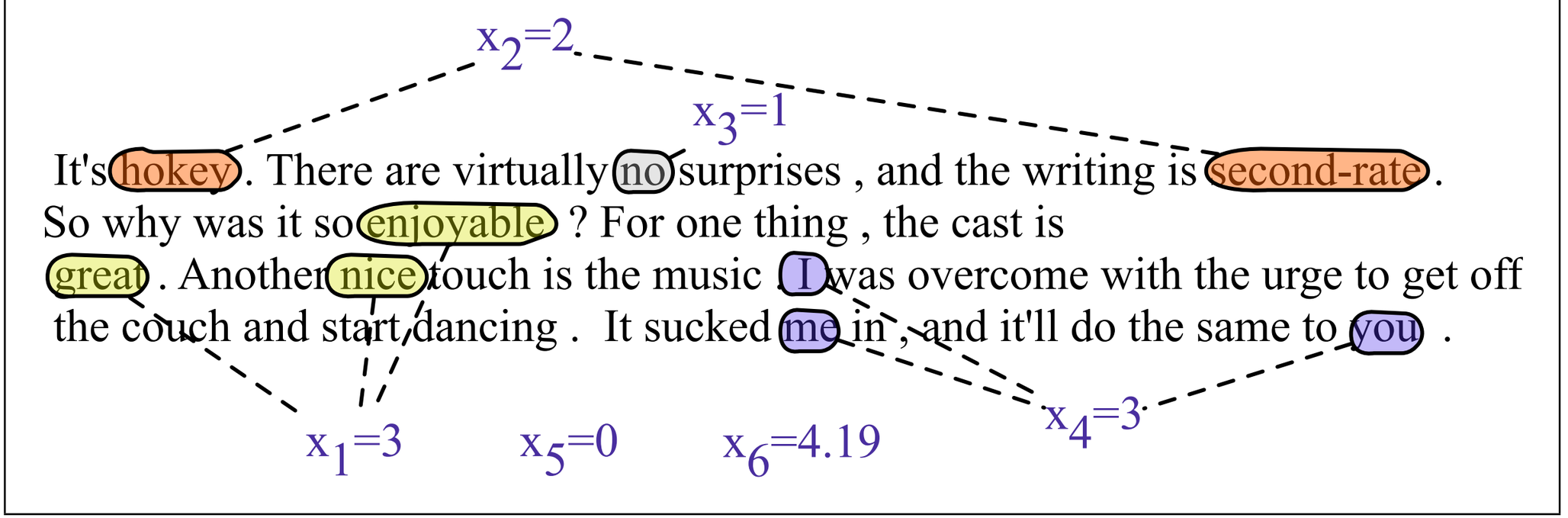
Classification in Logistic Regression

Logistic Regression

Logistic Regression: a text example on sentiment classification

Sentiment example: does y=1 or y=0?

It's hokey . There are virtually no surprises , and the writing is second-rate .
So why was it so enjoyable ? For one thing , the cast is
great . Another nice touch is the music . I was overcome with the urge to get off the couch and start dancing . It sucked me in , and it'll do the same to you .



Var	Definition	Value in Fig. 5.2
x_1	count(positive lexicon) $\in$ doc)	3
x_2	count(negative lexicon) $\in$ doc)	2
x_3	$\begin{cases} 1 & \text{if } \text{“no”} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$	1
x_4	count(1st and 2nd pronouns $\in$ doc)	3
x_5	$\begin{cases} 1 & \text{if } \text{“!”} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$	0
x_6	log(word count of doc)	$\ln(66) = 4.19$

Classifying sentiment for input x

Var	Definition	Val
x_1	count(positive lexicon) $\in$ doc)	3
x_2	count(negative lexicon) $\in$ doc)	2
x_3	$\begin{cases} 1 & \text{if } \text{“no”} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$	1
x_4	count(1st and 2nd pronouns $\in$ doc)	3
x_5	$\begin{cases} 1 & \text{if } \text{“!”} \in \text{doc} \\ 0 & \text{otherwise} \end{cases}$	0
x_6	log(word count of doc)	$\ln(66) = 4.19$

5.2

Suppose $w = [2.5, -5.0, -1.2, 0.5, 2.0, 0.7]$

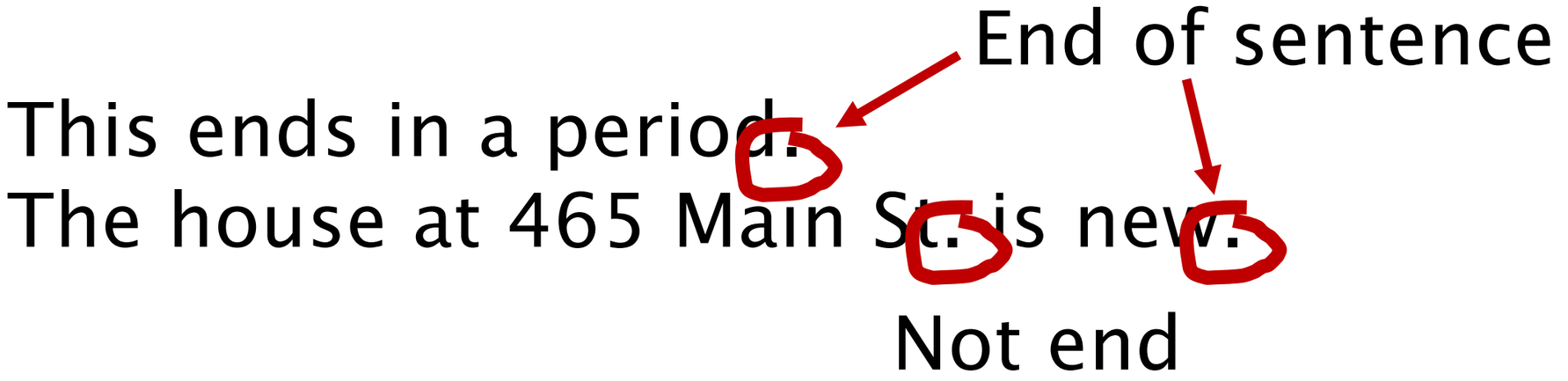
$b = 0.1$

Classifying sentiment for input x

$$
\begin{aligned}
p(+|x) = P(Y=1|x) &= \sigma(w \cdot x + b) \\
&= \sigma([2.5, -5.0, -1.2, 0.5, 2.0, 0.7] \cdot [3, 2, 1, 3, 0, 4.19] + 0.1) \\
&= \sigma(.833) \\
&= 0.70
\end{aligned}
$$

$$
\begin{aligned}
p(-|x) = P(Y=0|x) &= 1 - \sigma(w \cdot x + b) \\
&= 0.30
\end{aligned}
$$

We can build features for logistic regression for any classification task: period disambiguation



$$x_1 = \begin{cases} 1 & \text{if } \text{“}Case(w_i) = \text{Lower”} \\ 0 & \text{otherwise} \end{cases}$$

$$x_2 = \begin{cases} 1 & \text{if } \text{“}w_i \in \text{AcronymDict”} \\ 0 & \text{otherwise} \end{cases}$$

$$x_3 = \begin{cases} 1 & \text{if } \text{“}w_i = \text{St. \& } Case(w_{i-1}) = \text{Cap”} \\ 0 & \text{otherwise} \end{cases}$$

Classification in (**binary**) logistic regression: summary

Given:

- a set of classes: (+ sentiment,- sentiment)
- a vector **x** of features `[x1, x2, …, xn]`
  - x1= count("awesome")
  - x2 = log(number of words in review)
- A vector **w** of weights `[w1, w2, …, wn]`
  - w_i for each feature f_i

$$P(y=1) = \sigma(w \cdot x + b)$$

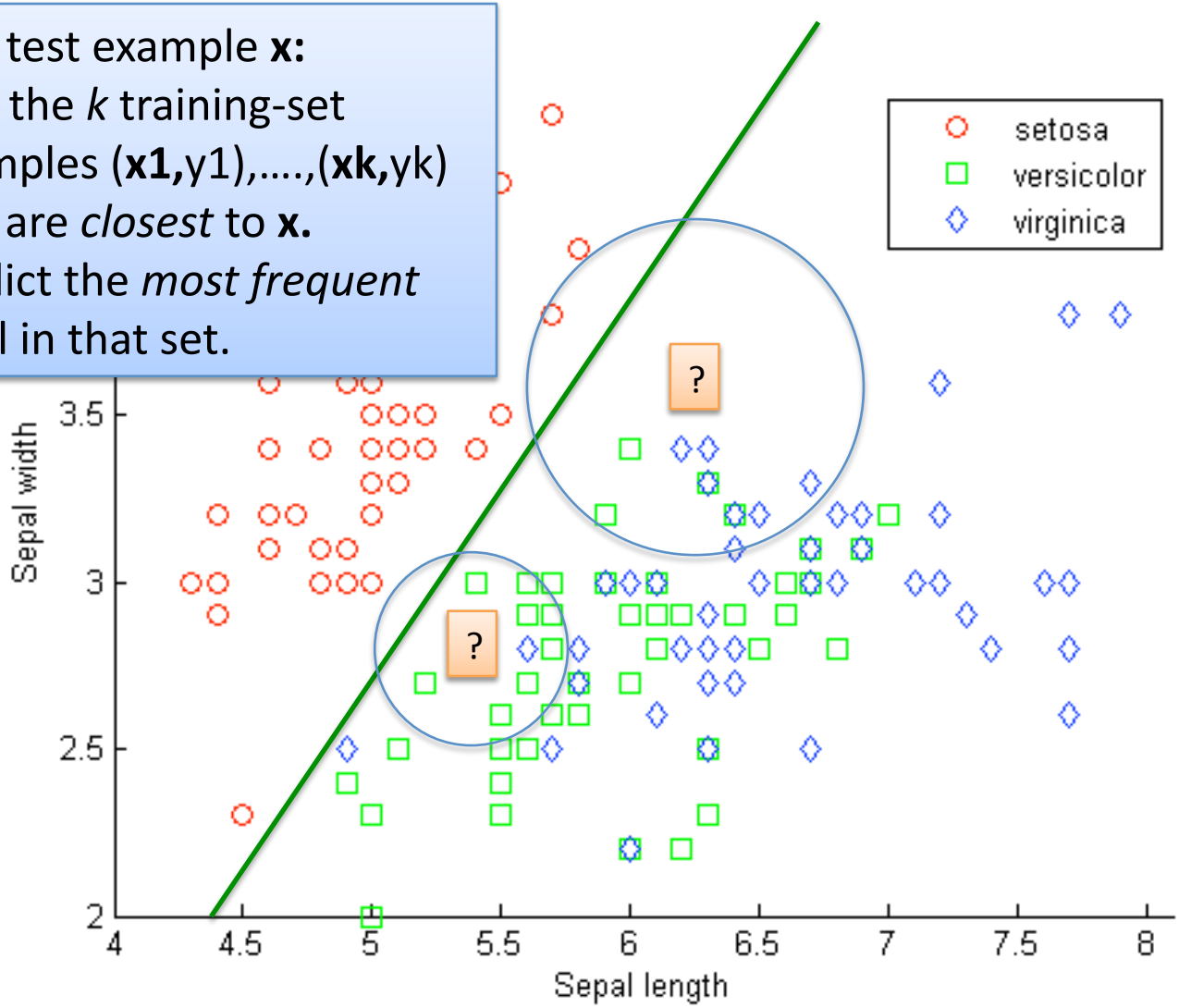
$$= \frac{1}{1+e^{-(w \cdot x + b)}}$$

Nearest Neighbor Learning

k-nearest neighbor learning

Given a test example **x:**

1. Find the *k* training-set examples (**x1,**y1),….,(**xk,**yk) that are *closest* to **x.**
2. Predict the *most frequent* label in that set.



Breaking it down:

- To train:
  - save the data
- To test:
  - For each test example **x:**

Very fast!

…you might build some indices….

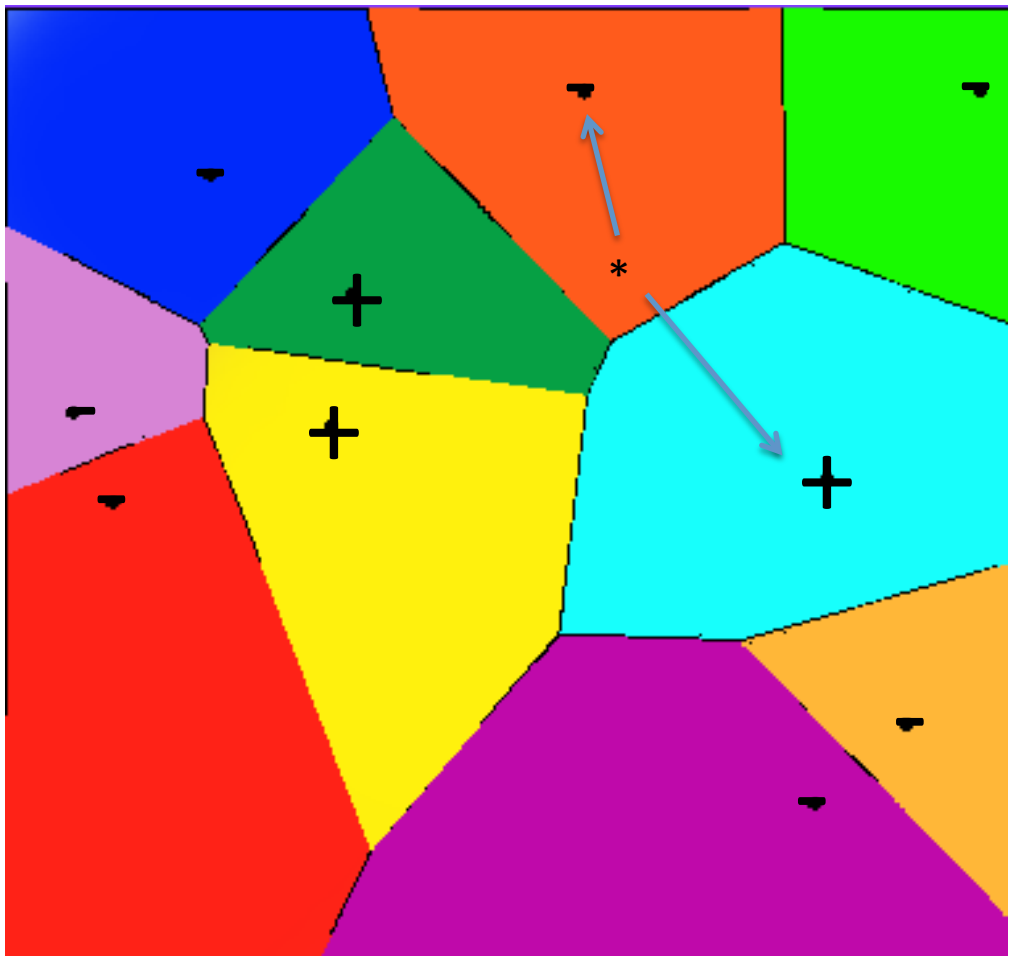
1. Find the *k* training-set examples (**x1,**y1), ….,(**xk,**yk) that are *closest* to **x.**
2. Predict the *most frequent* label in that set.

Prediction is relatively slow (compared to a linear classifier or decision tree)

What is the decision boundary for 1-NN?

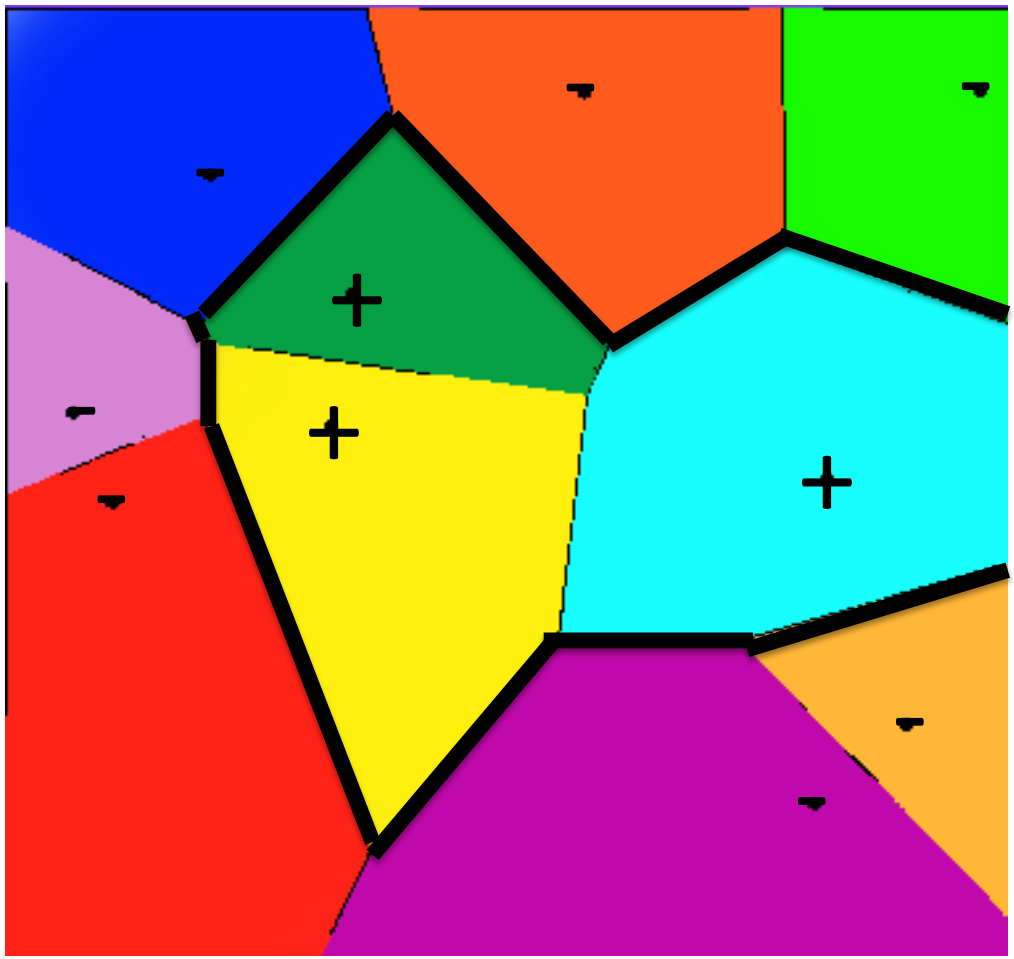
Each cell C_i is the set of all points that are closest to a particular example $\mathbf{x}_i$

Voronoi Diagram



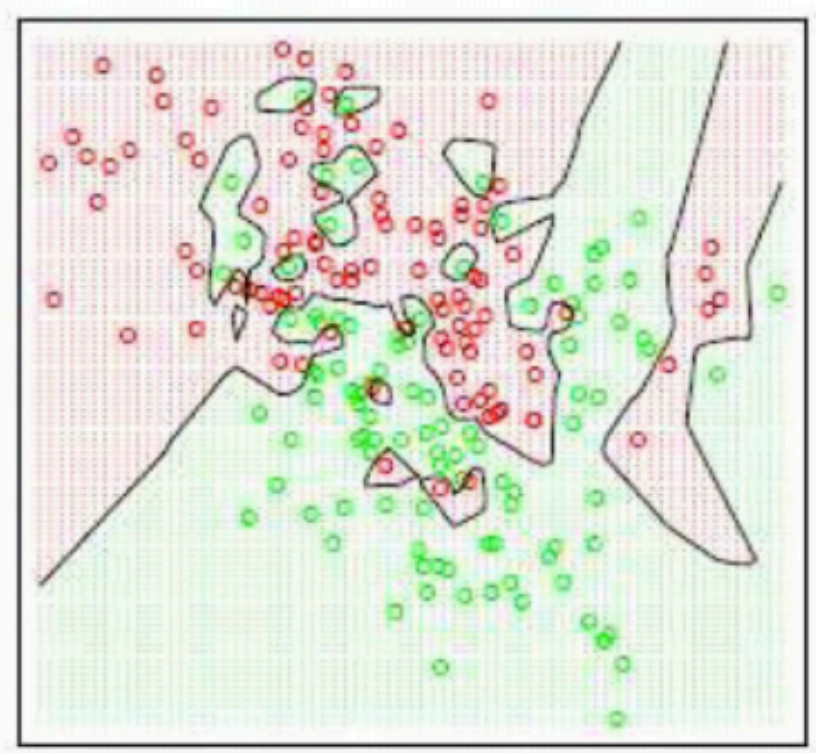
What is the decision boundary for 1-NN?

Voronoi Diagram



Effect of k on decision boundary

k=1



Figures from Hastie, Tibshirani and Friedman (Elements of Statistical Learning)

Some common variants

- Distance metrics:
  - Euclidean distance: **||x1 - x2||**
  - Cosine distance: 1 - **<x1,x2>/||x1||*||x2||**
    - this is in [0,1]
- Weighted nearest neighbor:
  - Instead of most frequent *y* in k-NN predict

$$\text{argmax}_y \sum_{(\mathbf{x}_i, y) \in kNN(\mathbf{x})} sim(\mathbf{x}_i, \mathbf{x})$$