CS2109S: Introduction to AI and Machine Learning

Lecture 6:
Logistic Regression

22 September 2023

Announcements

Admin

- **Midterm (Confirmed)**
  - Date & Time: **6 October, 10:00 – 12:00**
  - Venue: **MPSH 1A**
  - Notes:
    - Please come at 9:30. You are **allowed into hall at 9:45**
    - Materials: all topics covered before recess week **until Lecture 4**
    - Cheatsheet: 1 x A4 paper, both sides

Midterm Coverage

- Uninformed search
- Local search
- Informed search
  - A* search and heuristics
- Adversarial search
  - Drawing game tree
  - Minimax
  - Alpha-beta pruning
    - Practice: https://pascscha.ch/info2/abTreePractice/
- Decision trees

Plagiarism: Policy

Dos

- Discussions without sharing/consulting/taking away any code
- Use ChatGPT with proof (e.g., give ShareGPT links)

Don’ts

- Use codes from those who has done or currently doing the course
- Use codes from the internet without proper citations
- Publish codes to any publicly accessible sites (e.g., GitHub, Google Drive) or send your codes to anyone

Plagiarism checker will be performed against all previous batches!

Plagiarism: Case Studies

- Created a base source code and **derived** solutions from the same base
  - Caught and cases **submitted** to BOD
- Fully/partially copy-pasted friend's solutions (current/past student) and **tried to be smart** by doing some **nontrivial modifications**
  - Caught and cases **submitted** to BOD
- Fully/partially copy-pasted friend's solutions (current/past student) and **claimed** that solutions were from **ChatGPT** or **ported** from publicly available source codes
  - Caught and cases **submitted** to BOD

There were many more cases, but basically, we dealt with them **appropriately**

Plagiarism: Amnesty

- We are currently performing plagiarism checks on PS0-PS2
- So far, the results are **not good** :(
- Two options:
  - Turn yourself in before **Monday, 25th September 23:59**
    - We'll zero the affected PS(es) and call it a day
  - Pretend that you didn't do it
    - You may or may not get caught; If you get caught, **BOD may be involved**

Materials

Recap

- Linear Regression: **fitting a line** to data
- Gradient Descent
  - Gradient Descent Algorithm: **follow –gradient** to reduce error
  - Linear Regression with Gradient Descent: **convex** optimization, **one minimum**
  - Variants of Gradient Descent: batch, mini-batch, stochastic
- Linear Regression: Challenges and Solutions
  - Linear Regression with Many Attributes: $h_w(x) = \sum_{j=0}^{n} w_j x_j = w^T x$
  - Dealing with Features of Different Scales: **normalize**!
  - Dealing with Non-Linear Relationship: **transform features**
- Normal Equation: **analytically** find the best parameters

ERRATA

Linear Regression: Loss Landscape

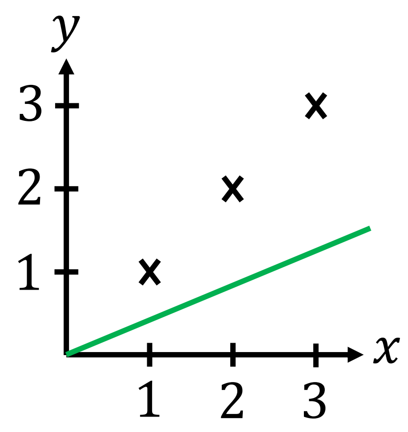
Hypothesis:

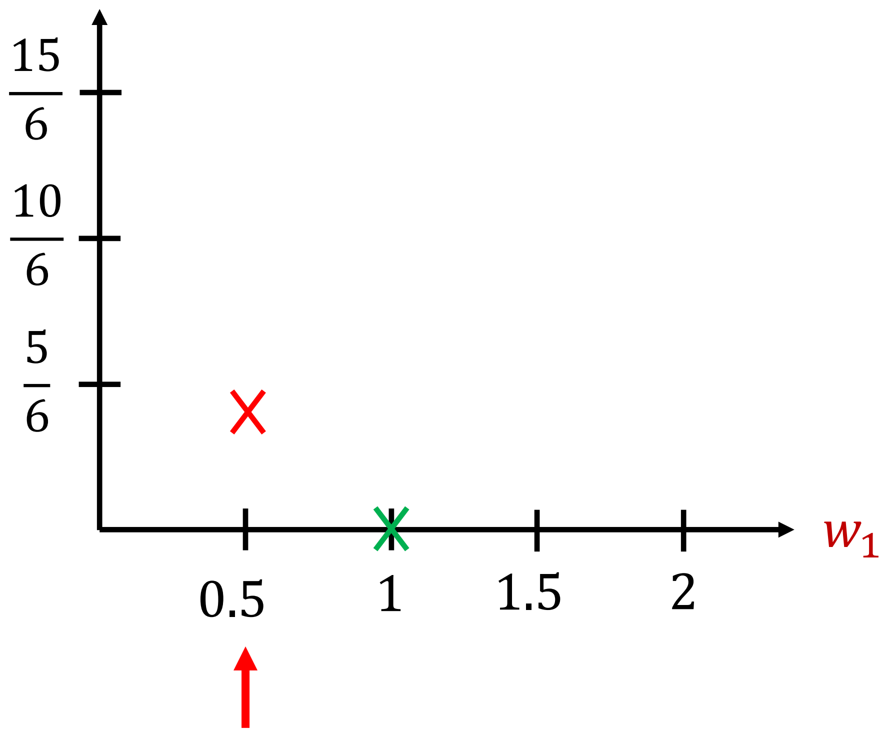
~~$h_w(x) = w_0 + w_1 x$~~

$h_w(x) = 0 + w_1 x$

Simplify: fix $w_0 = 0$ for easier visualization

$h_w(x) = 0.5x$





Loss Function:

$$J_{MSE}(w) = \frac{1}{2m}\sum_{i=1}^{m}\left(w_1 x^{(i)} - y^{(i)}\right)^2$$

$$= \frac{(0.5-1)^2 + (1-2)^2 + (1.5-3)^2}{2\times 3}$$

$$= \frac{0.5^2 + 1 + (1.5)^2}{6}$$

$$= \frac{3.5}{6}$$

ERRATA

Gradient Descent

- Start at some w
- Pick a nearby w that reduces $J(w)$

$$w_j \leftarrow w_j - \gamma \frac{\partial J(w_0, w_1, \ldots)}{\partial w_j}$$

Learning Rate

- Repeat until minimum is reached

Common Mistakes

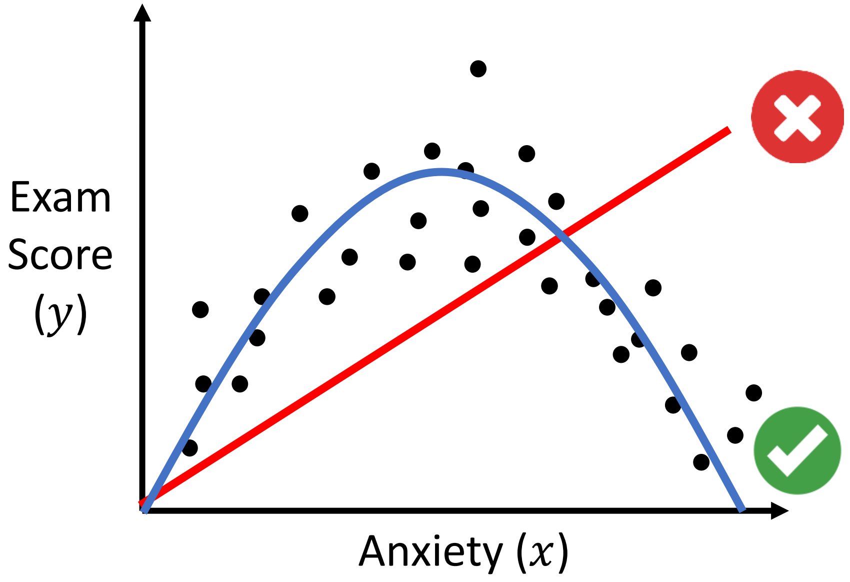
w_0 changed!

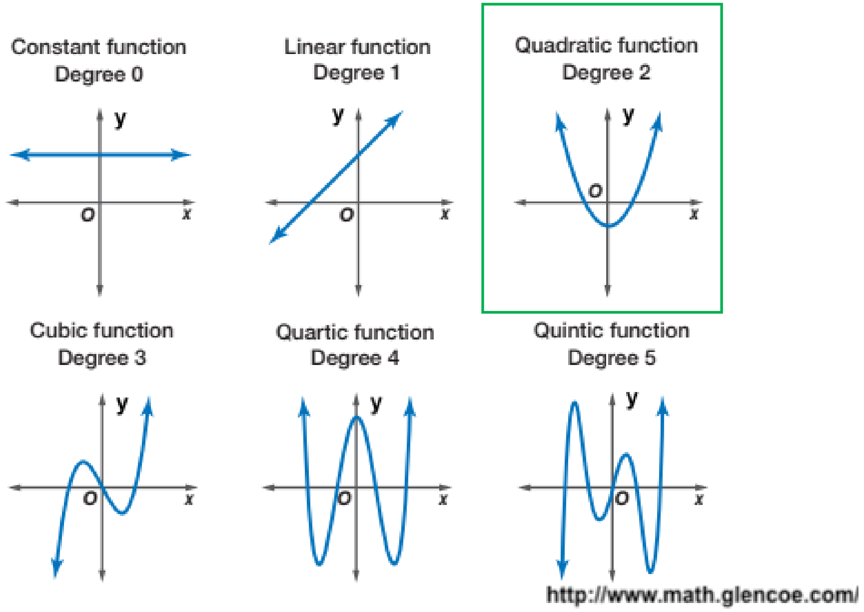
$$w_0 = w_0 - \gamma \frac{\partial J(w_0, w_1)}{\partial w_0}$$
$$w_1 = w_1 - \gamma \frac{\partial J(w_0, w_1)}{\partial w_1}$$

$$a = \frac{\partial J(w_0, w_1)}{\partial w_0}$$
$$b = \frac{\partial J(w_0, w_1)}{\partial w_1}$$
$$w_0 = w_0 - \gamma a$$
$$w_1 = w_1 - \gamma b$$

ERRATA

Dealing with Non-Linear Relationship





Which function?

$$h_w(x) = w_0 + w_1 x + w_2 x^2$$

Polynomial Regression

Need to scale this!

Generally:

$$h_w(x) = w_0 + w_1 f_1 + w_2 f_2 + w_3 f_3 + \cdots + w_n f_n$$

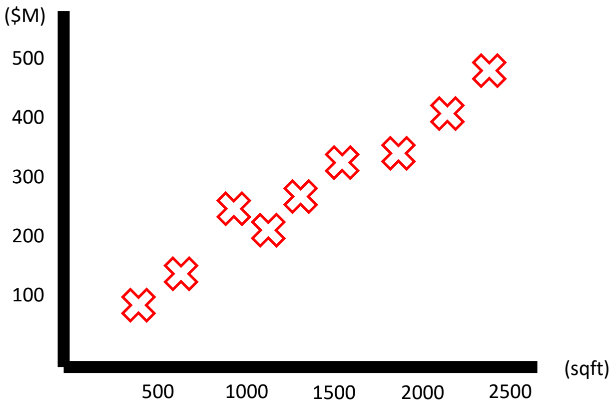
Transformed features:

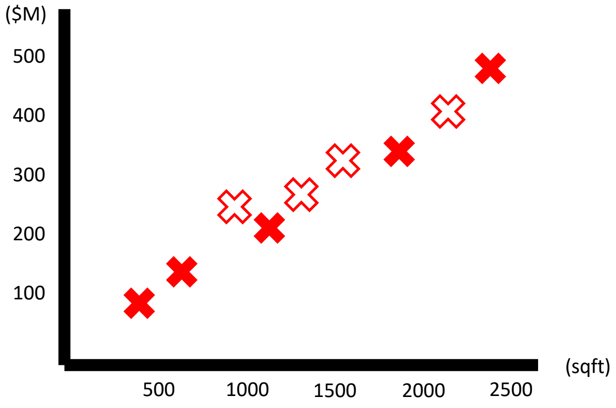
$e.g., f_1 = x, f_2 = x^2$

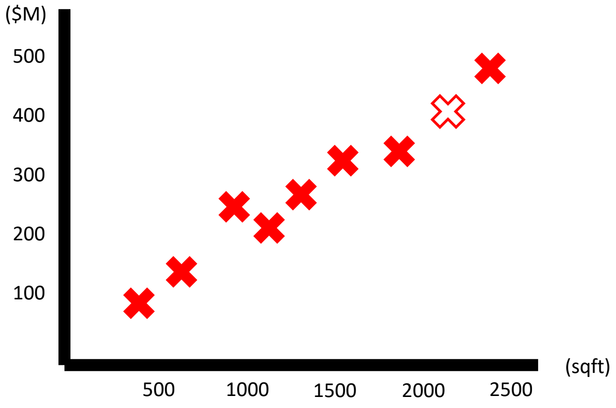
CLARIFICATION

Variants of Gradient Descent

$$w_j \leftarrow w_j - \gamma \frac{\partial J(w_0, w_1, \ldots)}{\partial w_j} \qquad J_{MSE}(w) = \frac{1}{2m}\sum_{i=1}^{m}\left(h_w(x^{(i)}) - y^{(i)}\right)^2$$







(Batch) Gradient Descent

- Consider all training examples

Mini-batch Gradient Descent

- Consider a subset of training examples at a time
- Cheaper (Faster) / iteration
- Randomness, may escape local minima

Stochastic Gradient Descent (SGD)

- Select one random data point at a time
- Cheapest (Fastest) / iteration
- More randomness, may escape local minima

Outline

- Logistic Regression
  - Classification with Continuous Inputs
  - Cross-entropy Loss
  - Logistic Regression with Gradient Descent
- Logistic Regression: Challenges and Solutions
  - Logistic Regression with Many Attributes
  - Dealing with Non-Linear Decision Boundary
- Multi-class Classification
- (More) Performance Measure
  - Receiver Operating Characteristic (ROC)
  - Area under ROC (AUC)
- Model Evaluation & Selection
  - Bias & Variance
- Hyperparameter Tuning

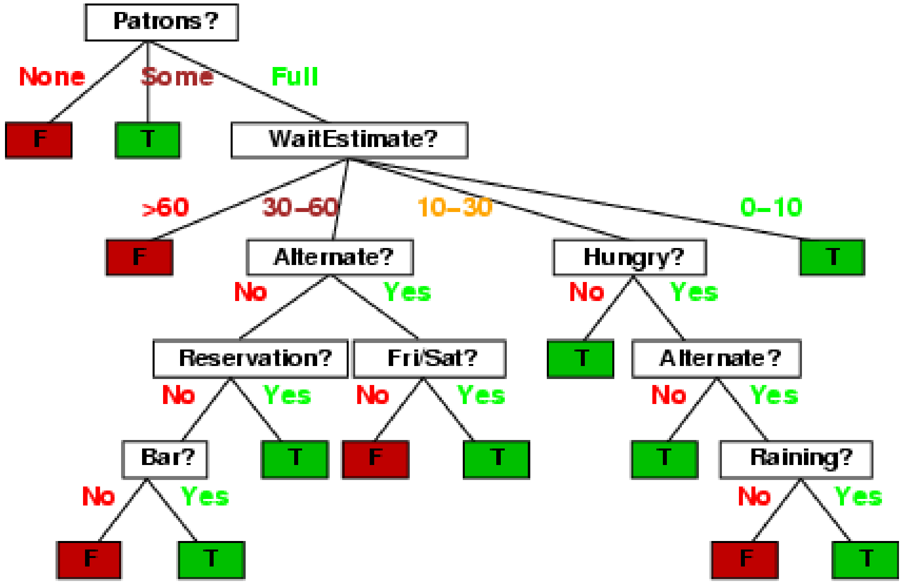
Outline

- **Logistic Regression**
  - Classification with Continuous Inputs
  - Cross-entropy Loss
  - Logistic Regression with Gradient Descent
- Logistic Regression: Challenges and Solutions
  - Logistic Regression with Many Attributes
  - Dealing with Non-Linear Decision Boundary
- Multi-class Classification
- (More) Performance Measure
  - Receiver Operating Characteristic (ROC)
  - Area under ROC (AUC)
- Model Evaluation & Selection
  - Bias & Variance
- Hyperparameter Tuning

Classification: What to Eat? Discrete Inputs!

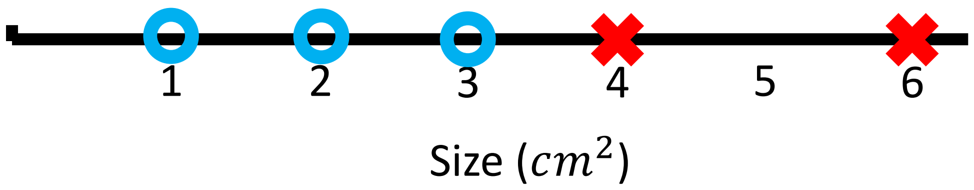
Decide whether to wait for a table at a restaurant based on the following input attributes:

1. Alternate: is there an alternative restaurant nearby?
2. Bar: is there a comfortable bar area to wait in?
3. Fri/Sat: is today Friday or Saturday?
4. Hungry: are we hungry?
5. Patrons: number of people in the restaurant (None, Some, Full)
6. Price: price range ($, $$, $$$)
7. Raining: is it raining outside?
8. Reservation: have we made a reservation?
9. Type: kind of restaurant (French, Italian, Thai, Burger)
10. WaitEstimate: estimated waiting time in minutes (0-10, 10-30, 30-60, >60)



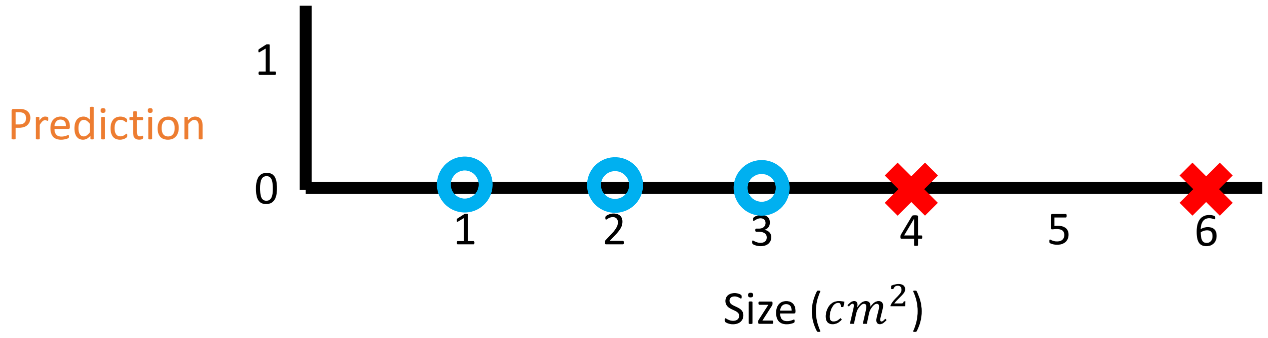
Classification with Continuous Inputs: 1D

Cancer prediction: benign, malignant



Classification with Continuous Inputs: 1D

Cancer prediction: benign (0), malignant (1)

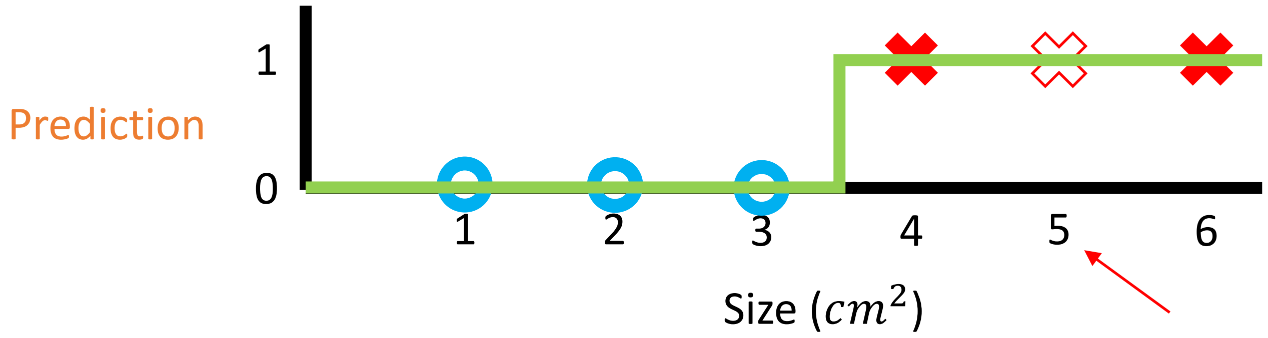


$$h_w(x) = \begin{cases} 1 & \text{if } w_0 + w_1 x > 0 \\ 0 & \text{otherwise} \end{cases}$$

$$h_w(x) = \begin{cases} 1 & \text{if } -3.5 + 1x > 0 \\ 0 & \text{otherwise} \end{cases}$$

Classification with Continuous Inputs: 1D

Cancer prediction: benign (0), malignant (1)



$$h_w(x) = \begin{cases} 1 & \text{if } w_0 + w_1 x > 0 \\ 0 & \text{otherwise} \end{cases}$$

$$h_w(x) = \begin{cases} 1 & \text{if } -3.5 + 1x > 0 \\ 0 & \text{otherwise} \end{cases}$$

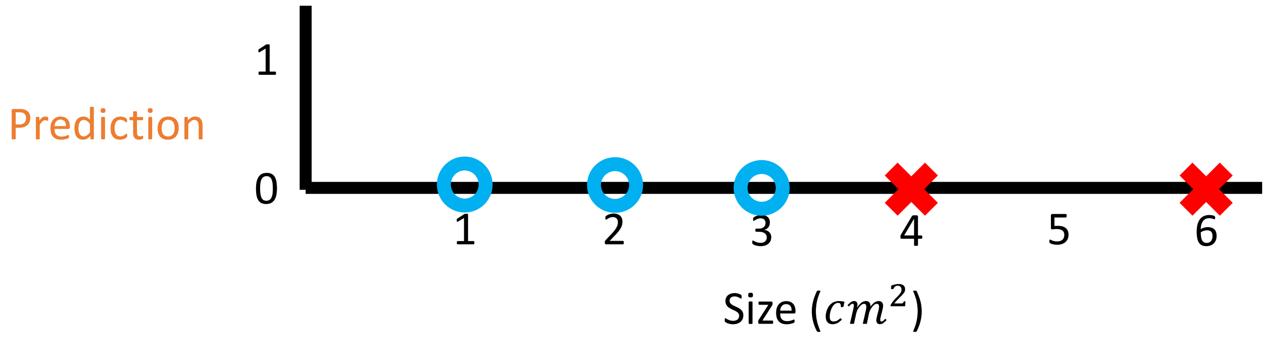
Threshold function

What's the issue?

Discontinuous, not differentiable

Classification with Continuous Inputs: 1D

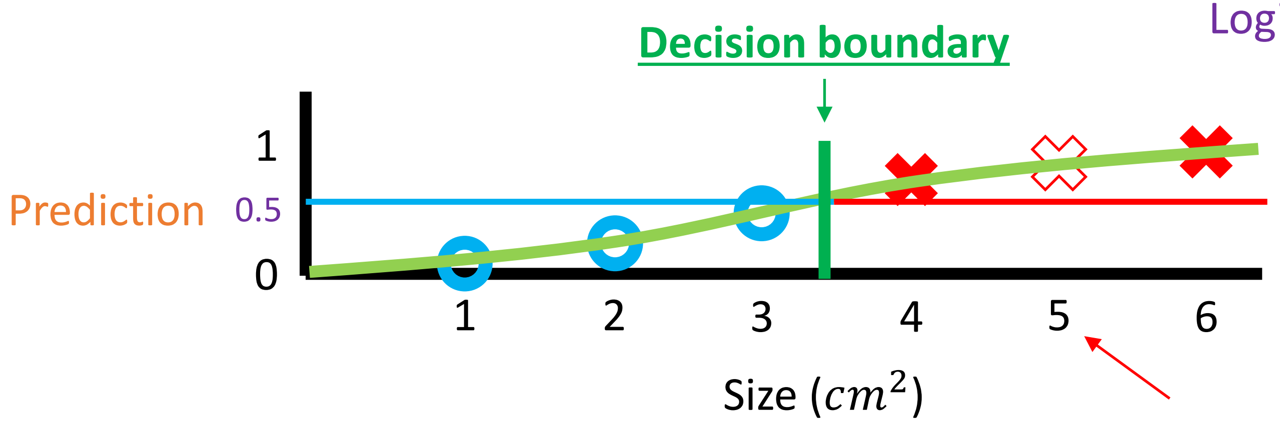
Cancer prediction: benign (0), malignant (1)



Continuous value output (probability), but used for classification

Logistic Regression: 1D

Cancer prediction: benign (0), malignant (1)



$$h_w(x) = \sigma(w_0 + w_1 x)$$

Logistic function $\sigma(z) = \dfrac{1}{1 + e^{-z}}$

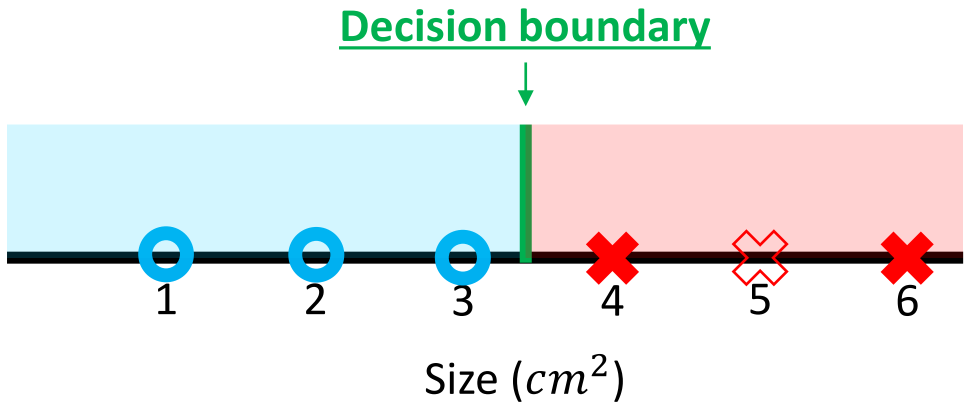
$$h_w(x) = \sigma(-3.5 + 1x)$$

Treat the output as **probability**

P(malignant) > X (e.g., 0.5) then malignant

Logistic Regression: 1D

Cancer prediction: benign, malignant



$$h_w(x) = \sigma(w_0 + w_1 x)$$

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

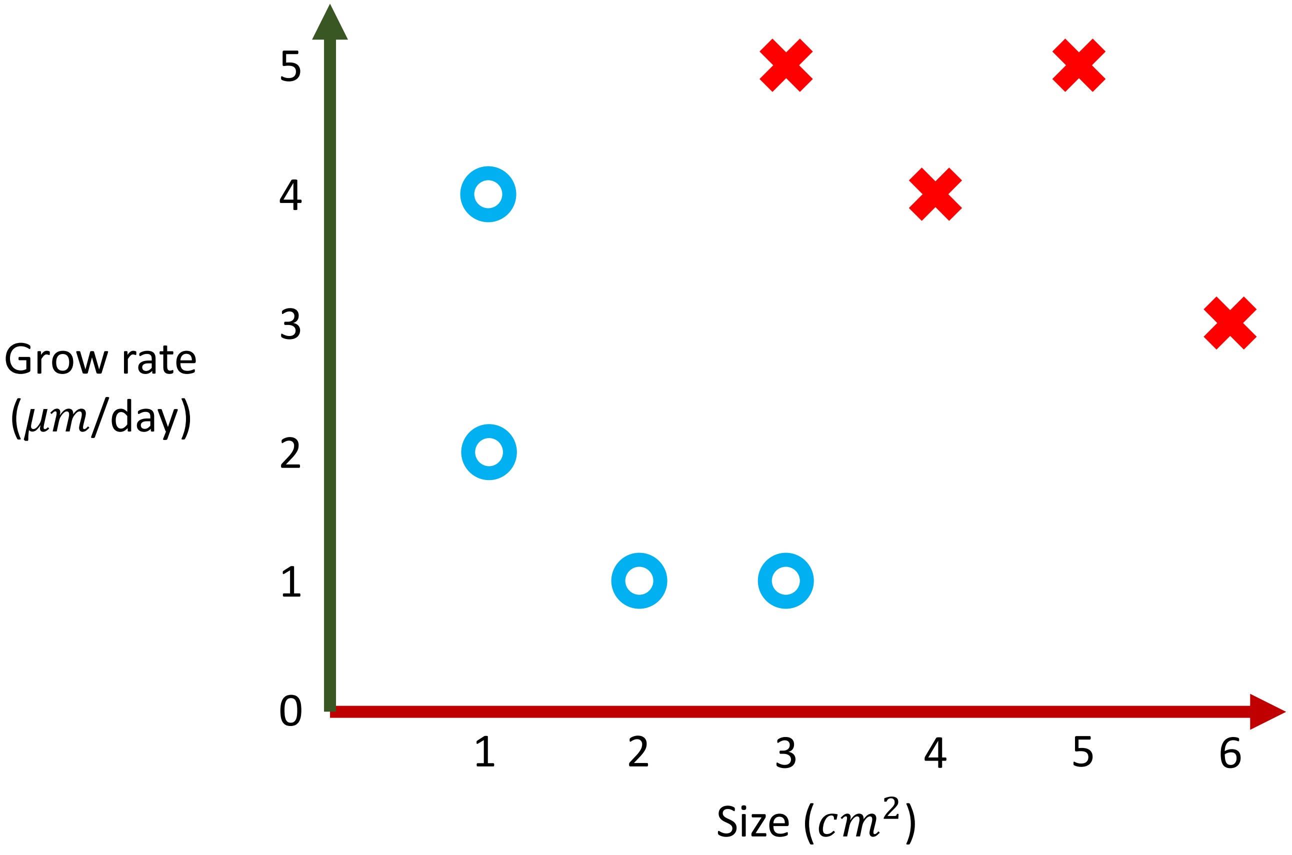
$$h_w(x) = \sigma(-3.5 + 1x)$$

Treat the output as **probability**

P(malignant) > X (e.g., 0.5) then malignant

Logistic Regression: 2D

Cancer prediction: benign, malignant



$$h_w(x) = \sigma(w_0 + w_1x_1 + w_2x_2)$$

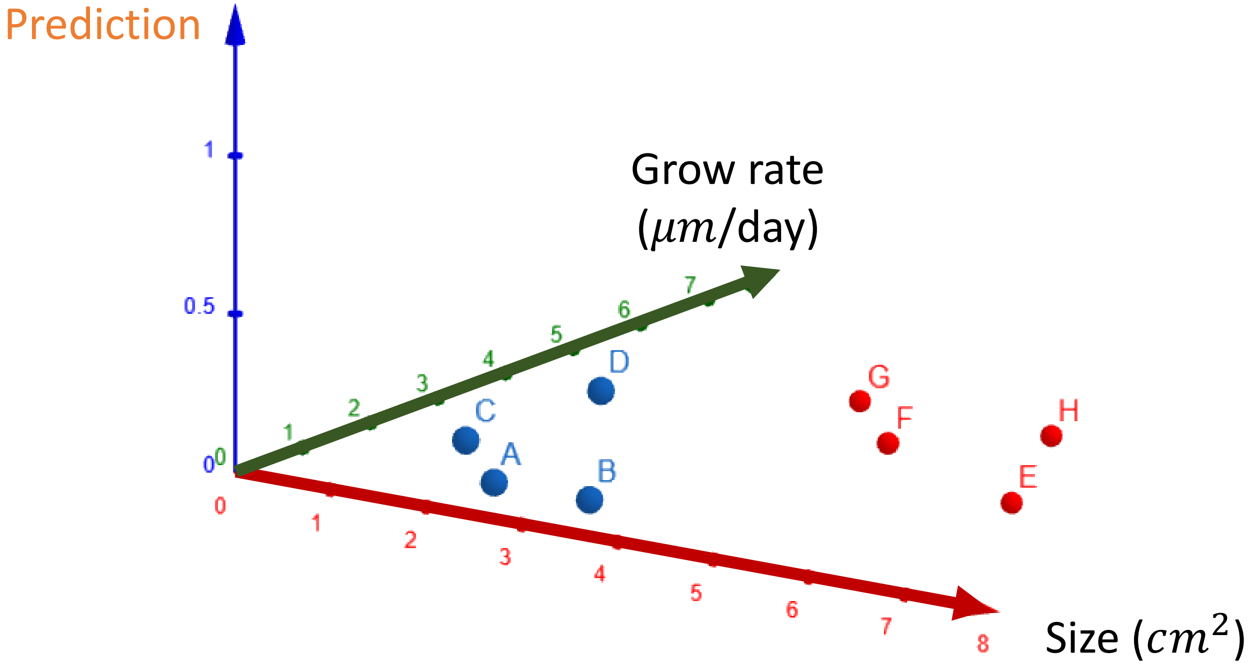
$$\sigma(z) = \frac{1}{1+e^{-z}}$$

Logistic Regression: 2D

Cancer prediction: benign, malignant

$$h_w(x) = \sigma(w_0 + w_1 x_1 + w_2 x_2)$$

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



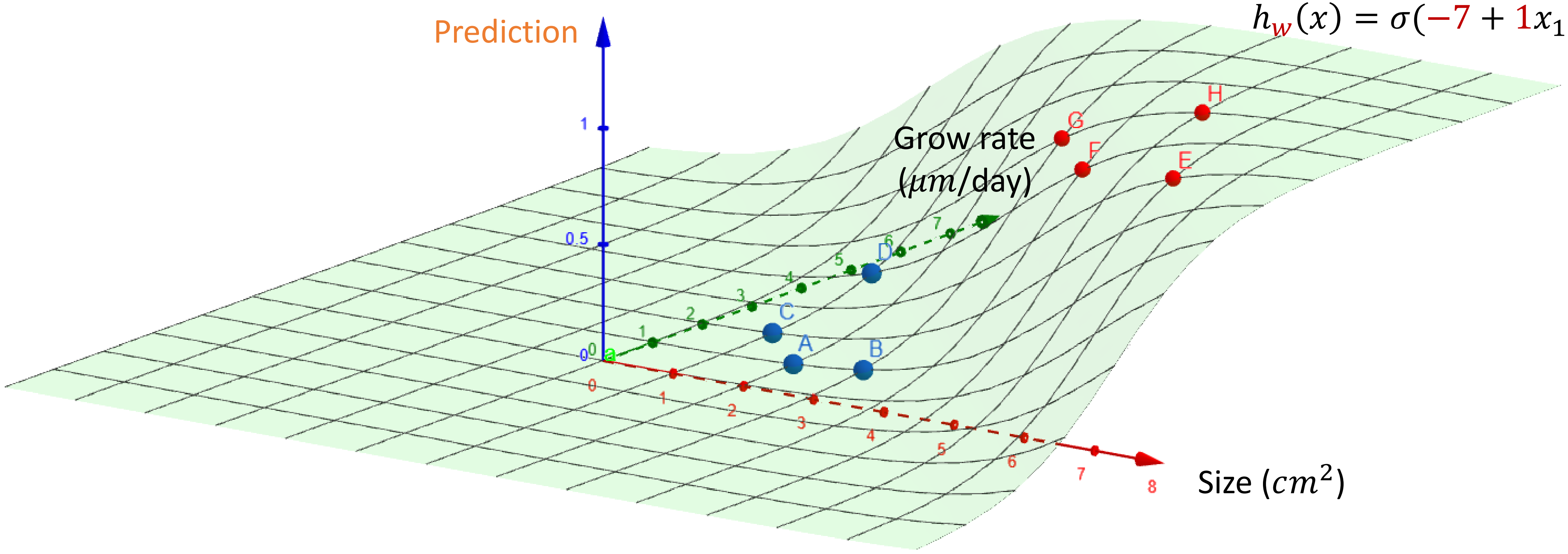
Logistic Regression: 2D

Cancer prediction: benign, malignant

$$h_w(x) = \sigma(w_0 + w_1x_1 + w_2x_2)$$

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

$$h_w(x) = \sigma(-7 + 1x_1 + 1x_2)$$



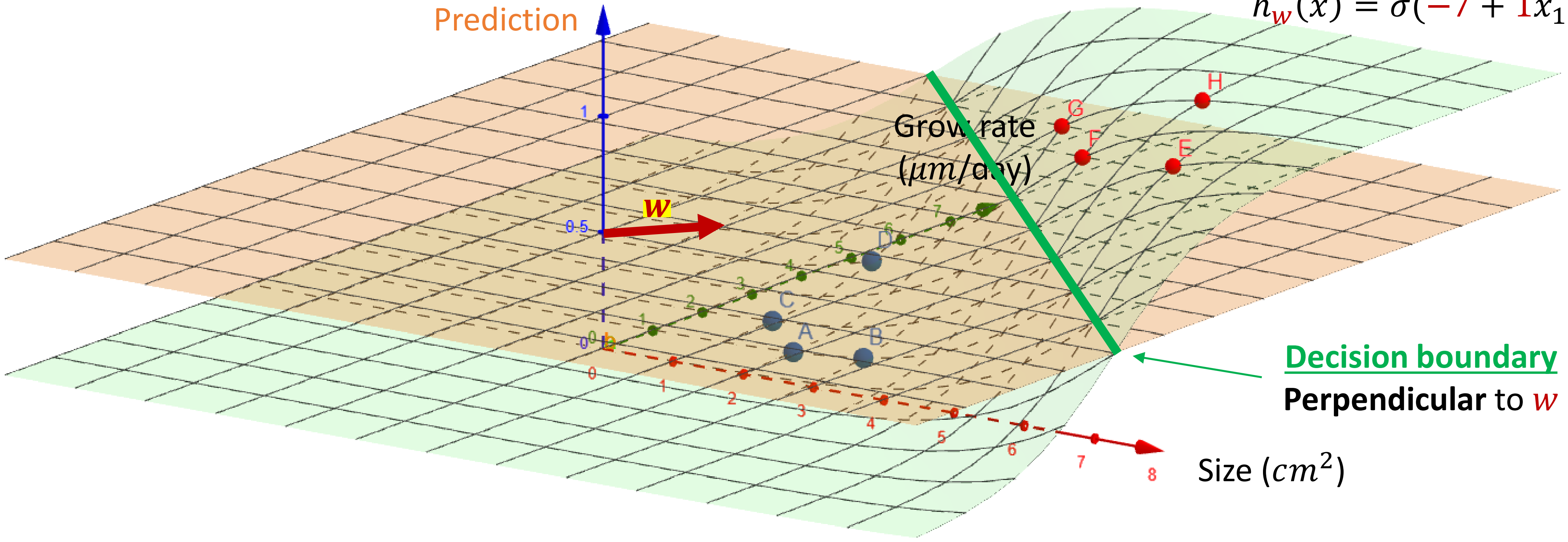
Logistic Regression: 2D

Cancer prediction: benign, malignant

$$h_w(x) = \sigma(w_0 + w_1 x_1 + w_2 x_2)$$

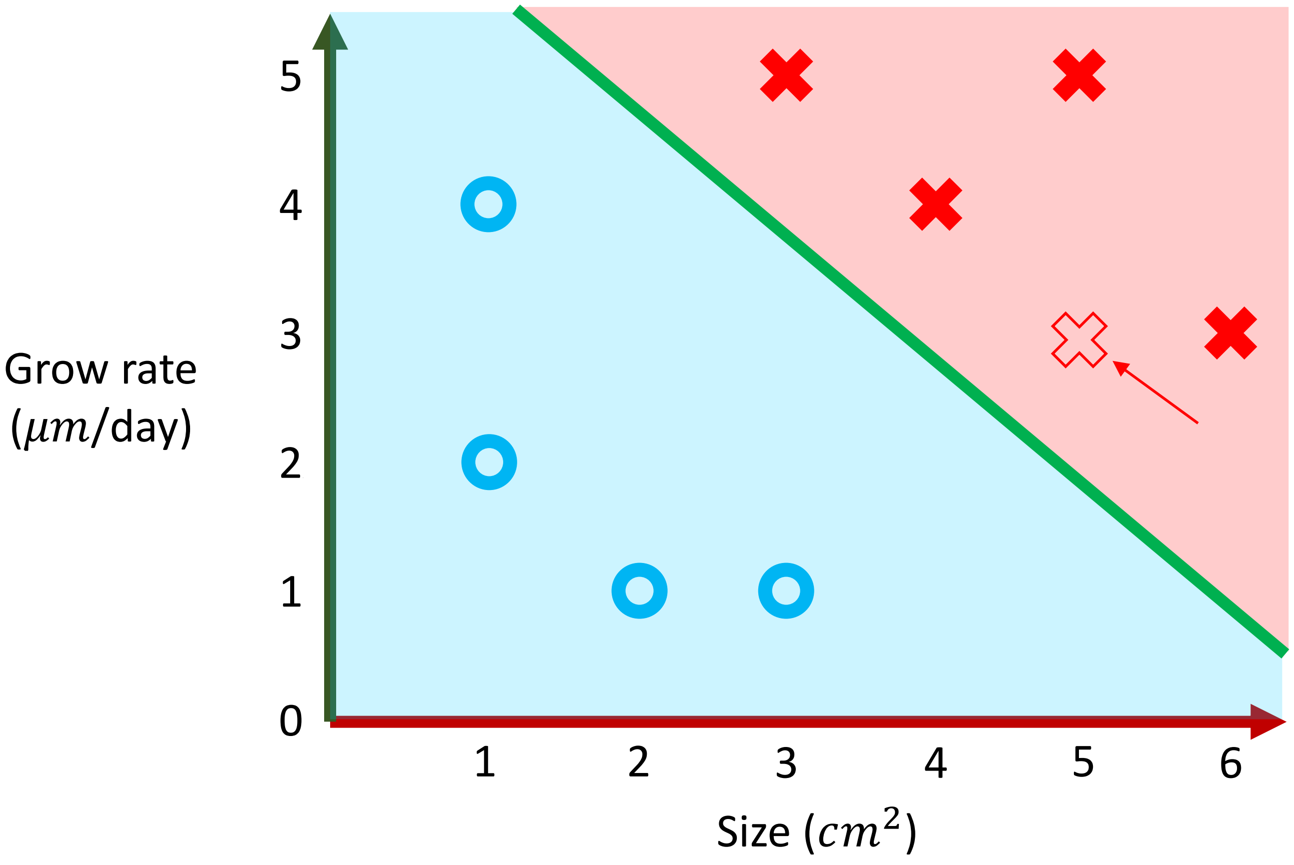
$$\sigma(z) = \frac{1}{1+e^{-z}}$$

$$h_w(x) = \sigma(-7 + 1x_1 + 1x_2)$$



Logistic Regression: 2D

Cancer prediction: benign, malignant



$$h_w(x) = \sigma(w_0 + w_1 x_1 + w_2 x_2)$$

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

$$h_w(x) = \sigma(-7 + 1x_1 + 1x_2)$$

Find w that "best separate data"!

How?

Logistic Regression: Measuring Fit

For a set of m examples $\{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\}$:

$$h_w(x) = \sigma(w_0 + w_1 x_1 + w_2 x_2)$$
$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

$$J_{MSE}(w) = \frac{1}{2m} \sum_{i=1}^{m} \left(h_w(x^{(i)}) - y^{(i)}\right)^2$$
$$= \frac{1}{2m} \sum_{i=1}^{m} \left(\frac{1}{1+\boxed{e^{-(w_0+w_1x_1+w_2x_2)}}} - y^{(i)}\right)^2$$

Non-linear → Non-convex

Any alternatives?

Hint: $h_w(x)$ outputs **probability**

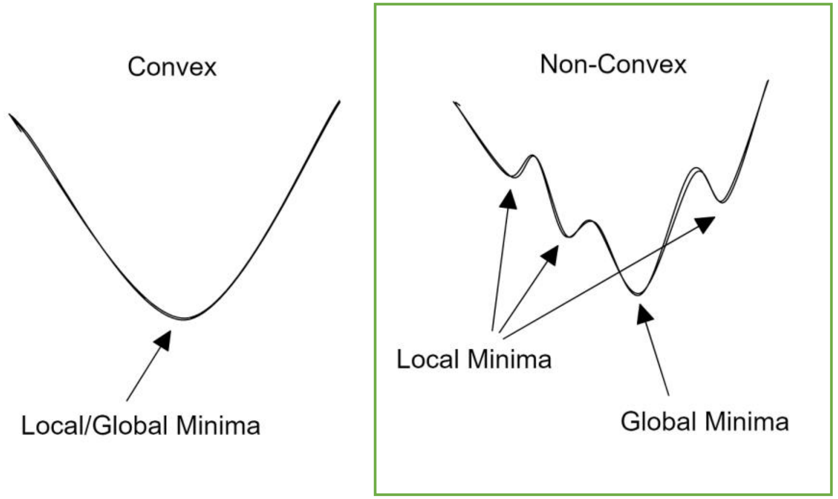


Image Credit: https://python.plainenglish.io/logistic-regression-in-machine-learning-from-scratch-872b1fedd05b

Measuring Closeness Between Prob Distribution

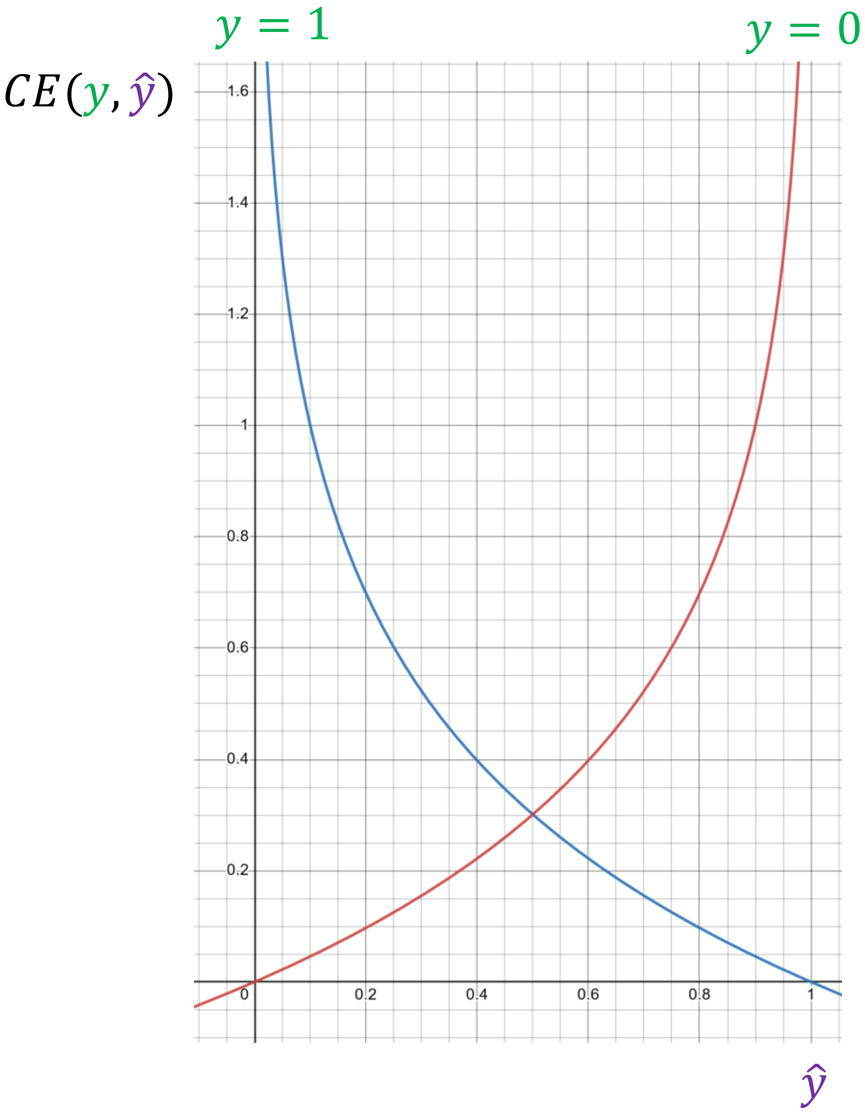
Cross-entropy for C classes:

$$CE(y, \hat{y}) = \sum_{i=1}^{C} -y_i \log(\hat{y}_i)$$

Binary cross-entropy:

$$BCE(y, \hat{y}) = -y\log(\hat{y}) - (1 - y)\log(1 - \hat{y})$$

P(malignant) = 0.8 P(not malignant) = 1 – 0.8 = 0.2



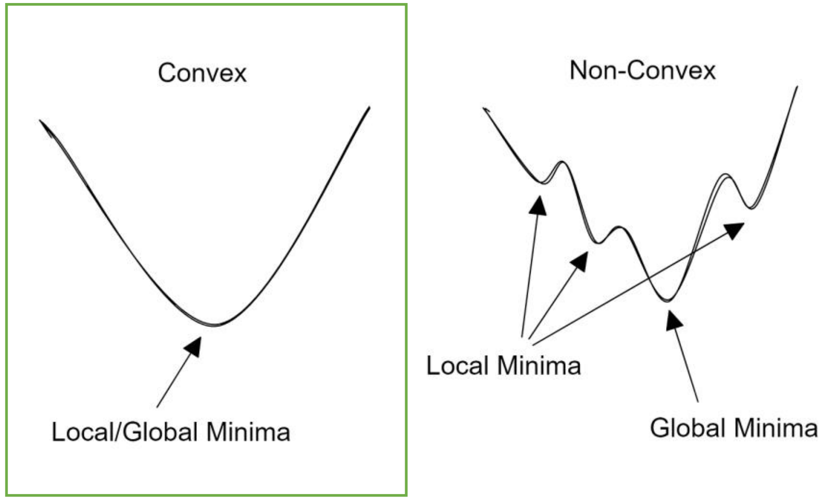
Logistic Regression with Cross-Entropy Loss

For a set of m examples $\{(x^{(1)}, y^{(1)}), \dots, (x^{(m)}, y^{(m)})\}$

we can compute the **binary cross entropy loss** as follows.

$$J_{BCE}(w) = \frac{1}{m} \sum_{i=1}^{m} BCE\left(h_w\left(x^{(i)}\right), y^{(i)}\right)$$

$h_w(x) = \sigma(w_0 + w_1 x_1 + w_2 x_2)$ (Probability output)



$$BCE(y, \hat{y})$$
$$= -y \log(\hat{y}) - (1 - y) \log(1 - \hat{y})$$
$$= -y \log\left(\frac{1}{1+e^{-(w_0+w_1x_1+w_2x_2)}}\right)$$
$$-(1-y) \log\left(1 - \frac{1}{1+e^{-(w_0+w_1x_1+w_2x_2)}}\right)$$

Linear! → convex

$\log(e^{a}) = c\boldsymbol{a}$, where c is constant

Image Credit: https://python.plainenglish.io/logistic-regression-in-machine-learning-from-scratch-872b1fedd05b

Logistic Regression with Gradient Descent

Hypothesis:

$h_w(x) = \sigma(w_0 + w_1 x_1 + w_2 x_2)$

Weight Update:

$$w_j \leftarrow w_j - \gamma \frac{\partial J_{BCE}(w_0, w_1, \dots)}{\partial w_j}$$

Loss Function:

$$J_{BCE}(w) = \frac{1}{m}\sum_{i=1}^{m} BCE(h_w(x^{(i)}), y^{(i)})$$

$$\frac{\partial J_{BCE}(w)}{\partial w_j} = \frac{\partial}{\partial w_j}\frac{1}{m}\sum_{i=1}^{m} BCE(h_w(x^{(i)}), y^{(i)})$$

Same as Linear Regression!

Will discuss the derivations in the next Tutorial

$$\frac{\partial J_{BCE}(w)}{\partial w_0} = \frac{1}{m}\sum_{i=1}^{m}(h_w(x^{(i)}) - y^{(i)})$$

$$\frac{\partial J_{BCE}(w)}{\partial w_1} = \frac{1}{m}\sum_{i=1}^{m}(h_w(x^{(i)}) - y^{(i)}).x_1^{(i)}$$

$$\frac{\partial J_{BCE}(w)}{\partial w_2} = \frac{1}{m}\sum_{i=1}^{m}(h_w(x^{(i)}) - y^{(i)}).x_2^{(i)}$$

Outline

- Logistic Regression
  - Classification with Continuous Inputs
  - Cross-entropy Loss
  - Logistic Regression with Gradient Descent
- **Logistic Regression: Challenges and Solutions**
  - Logistic Regression with Many Attributes
  - Dealing with Non-Linear Decision Boundary
- Multi-class Classification
- (More) Performance Measure
  - Receiver Operating Characteristic (ROC)
  - Area under ROC (AUC)
- Model Evaluation & Selection
  - Bias & Variance
- Hyperparameter Tuning

Logistic Regression with Many Attributes

x_0 Bias	x_1 Texture	x_2 Smoothness	x_3 Grow rate	x_4 Size	y Malignant?
1	24.8	0.1	2	1	No
1	15.6	0.2	2	2	No
1	9.4	0.12	0	1	No
1	18.7	0.3	2	3	Yes
1	23.1	0.27	0	3.1	Yes
1	12.4	0.01	2	2.5	No
1	7.0	0.14	1	6	Yes
1	19.9	0.1	2	10	Yes
1	21.5	0.1	3	4.2	Yes
1	11.2	0.2	2	1	No

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

Hypothesis:

$$h_w(x) = \sigma(w_0 x_0 + w_1 x_1 + w_2 x_2 + w_3 x_3 + w_4 x_4)$$

Hypothesis (for n features):

$$h_w(x) = \sigma\left(\sum_{j=0}^{n} w_j x_j\right) = \sigma(w^T x)$$

vector

Weight Update (for n features):

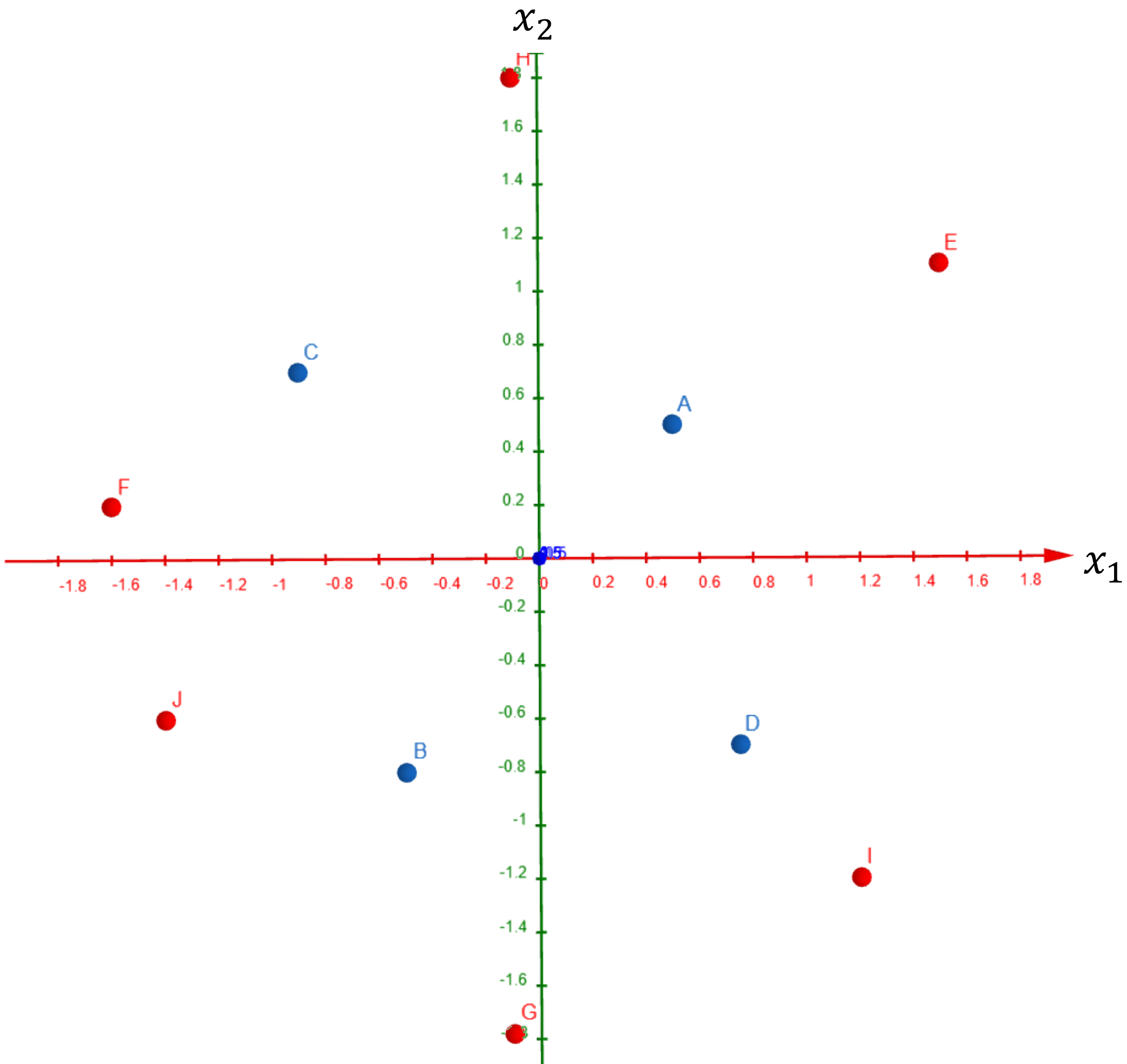
$$w_j \leftarrow w_j - \gamma \frac{\partial J_{BCE}(w_0, w_1, \ldots, w_n)}{\partial w_j}$$

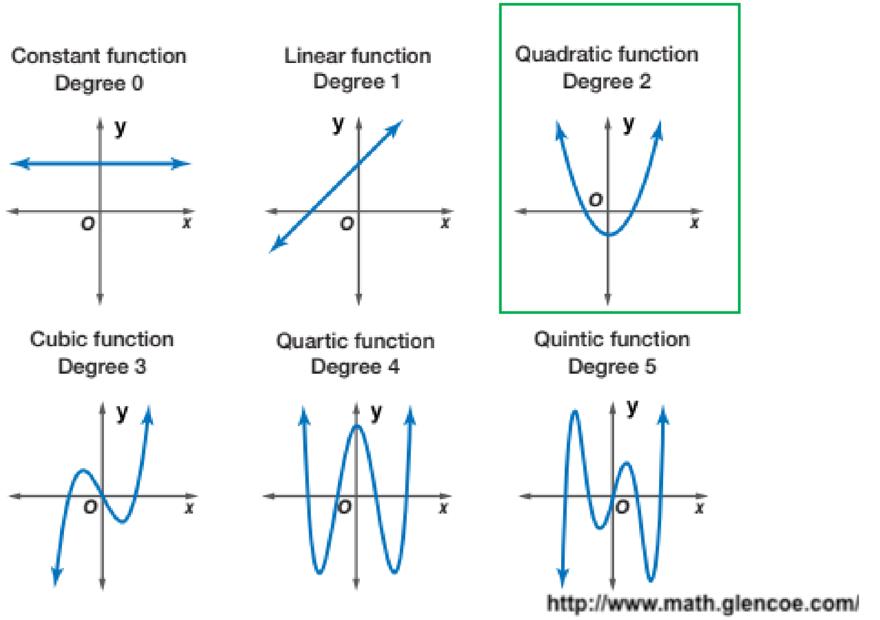
$$w_j \leftarrow w_j - \gamma \frac{1}{m} \sum_{i=1}^{m} \left(h_w(x^{(i)}) - y^{(i)}\right). x_j^{(i)}$$

Same as Linear Regression!

Will discuss the derivations in the next Tutorial

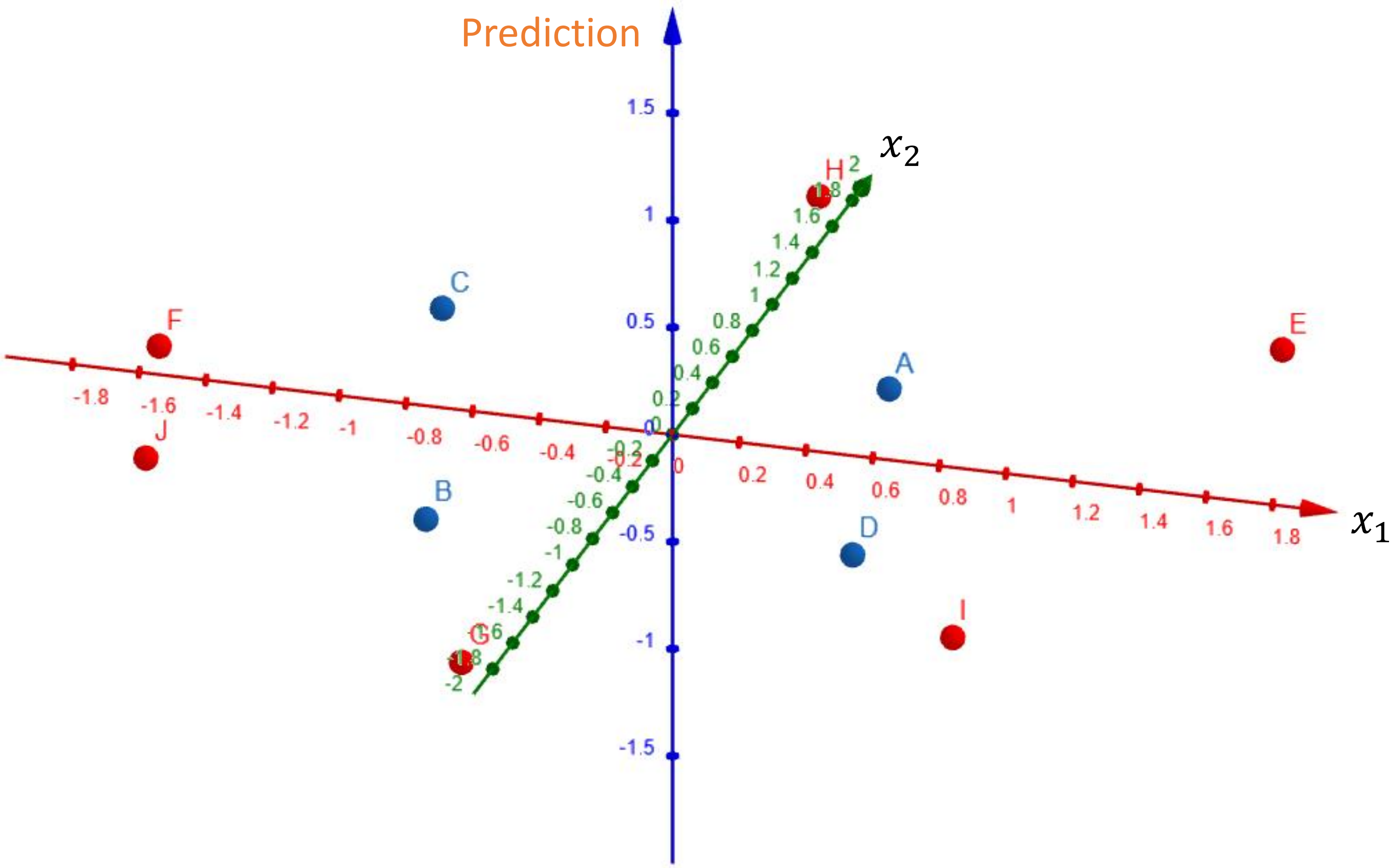
Dealing with Non-Linear Decision Boundary

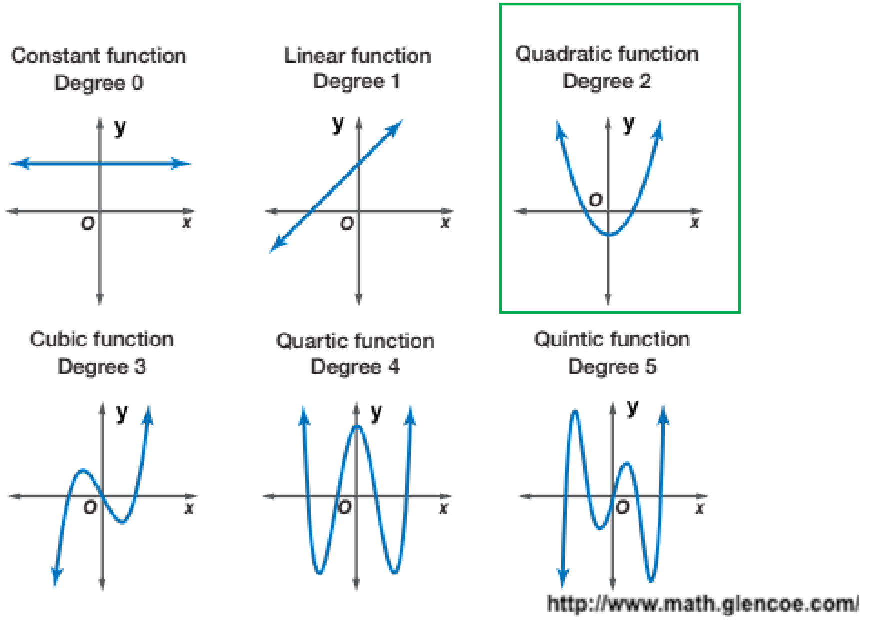




Which function?

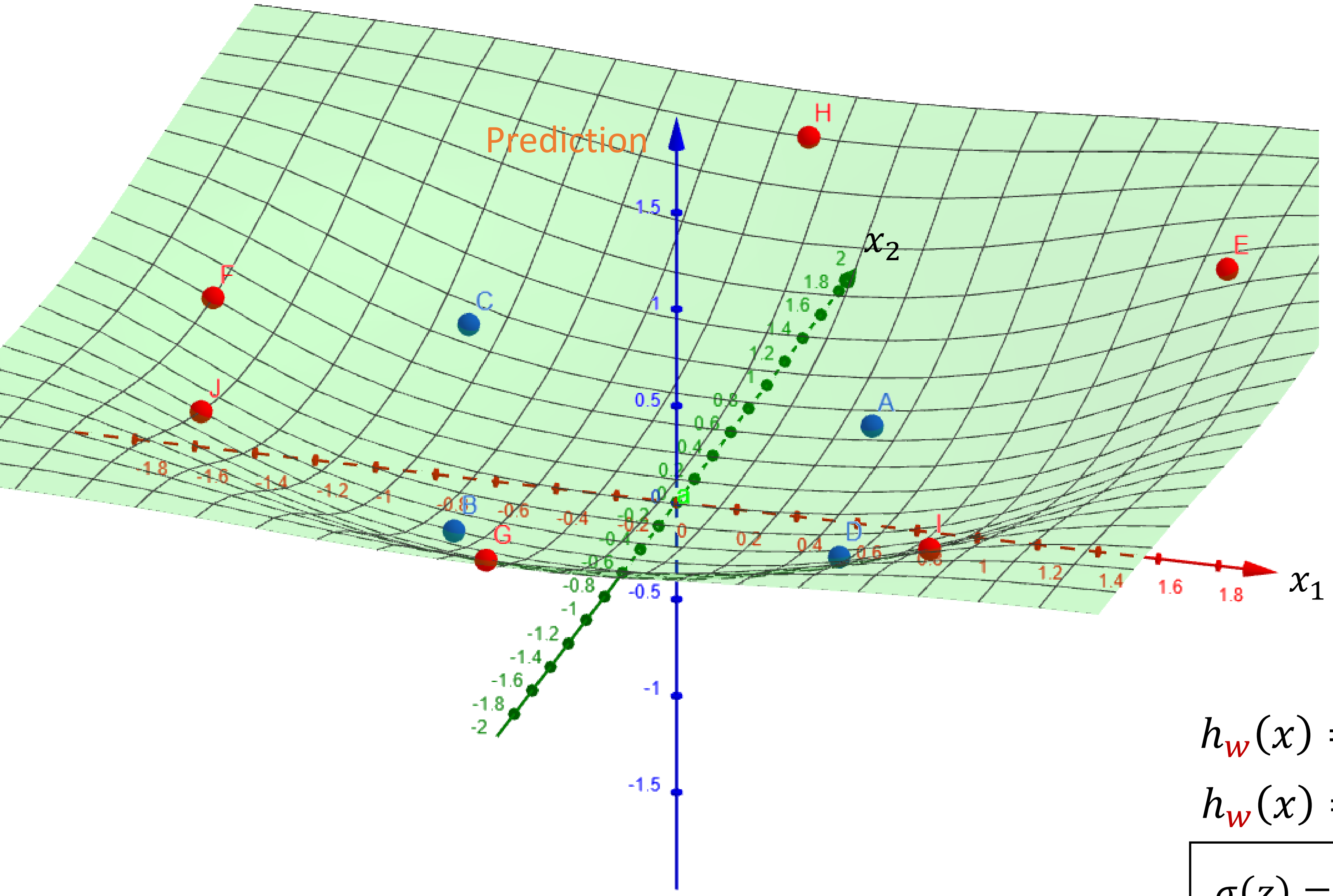
Dealing with Non-Linear Decision Boundary

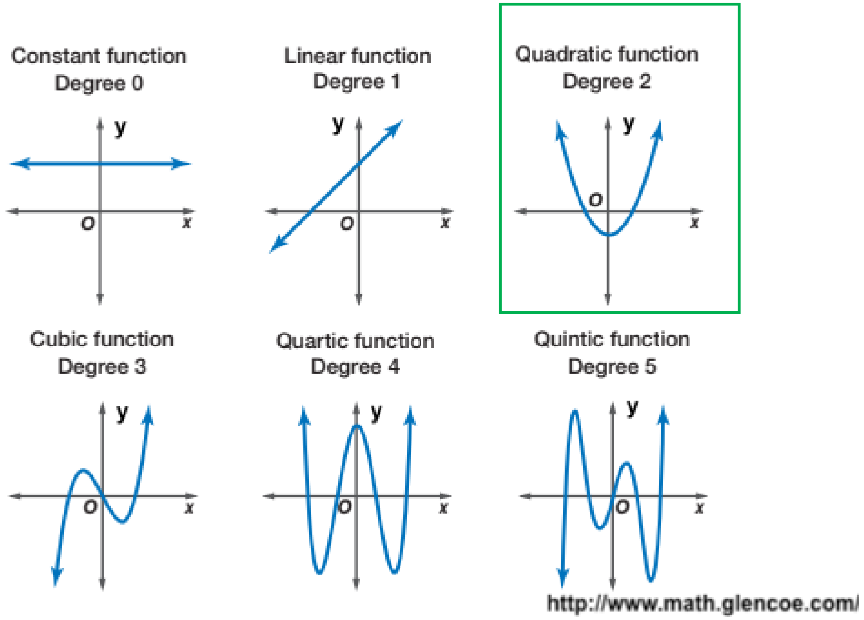




Which function?

Dealing with Non-Linear Decision Boundary





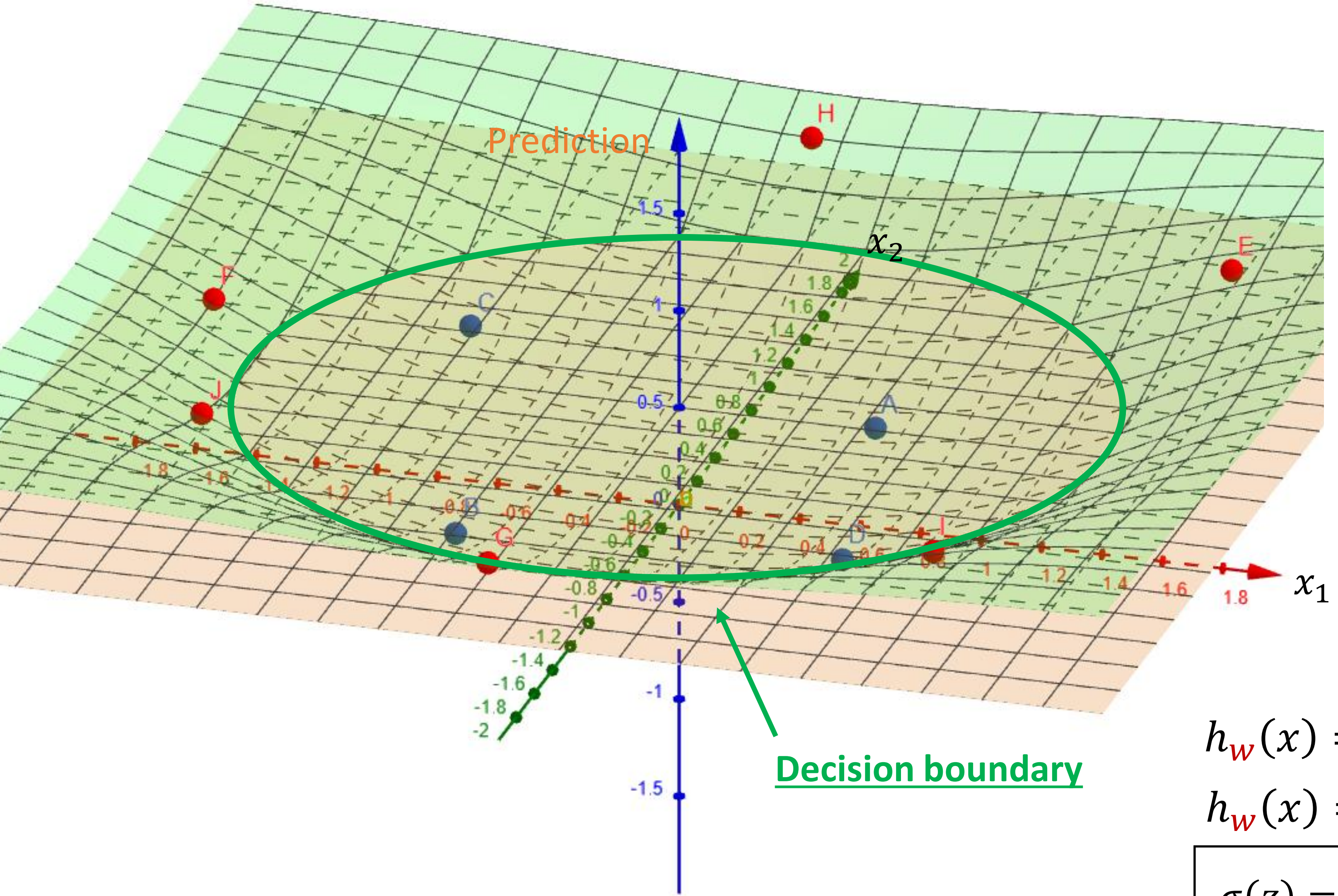
Which function?

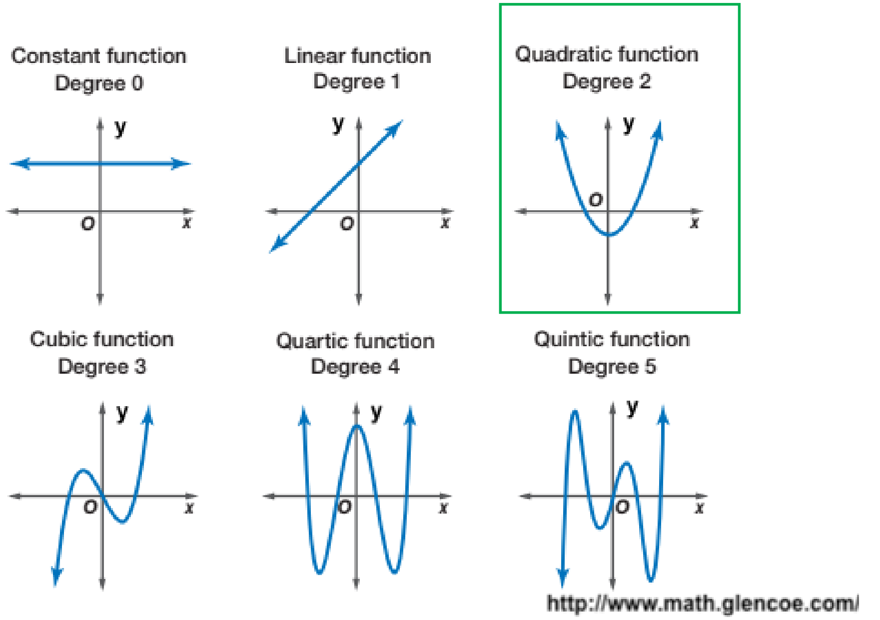
$$h_w(x) = \sigma(w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1^2 + w_4 x_2^2)$$

$$h_w(x) = \sigma(-2 + 0x_1 + 0x_2 + 1x_1^2 + 1x_2^2)$$

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

Dealing with Non-Linear Decision Boundary





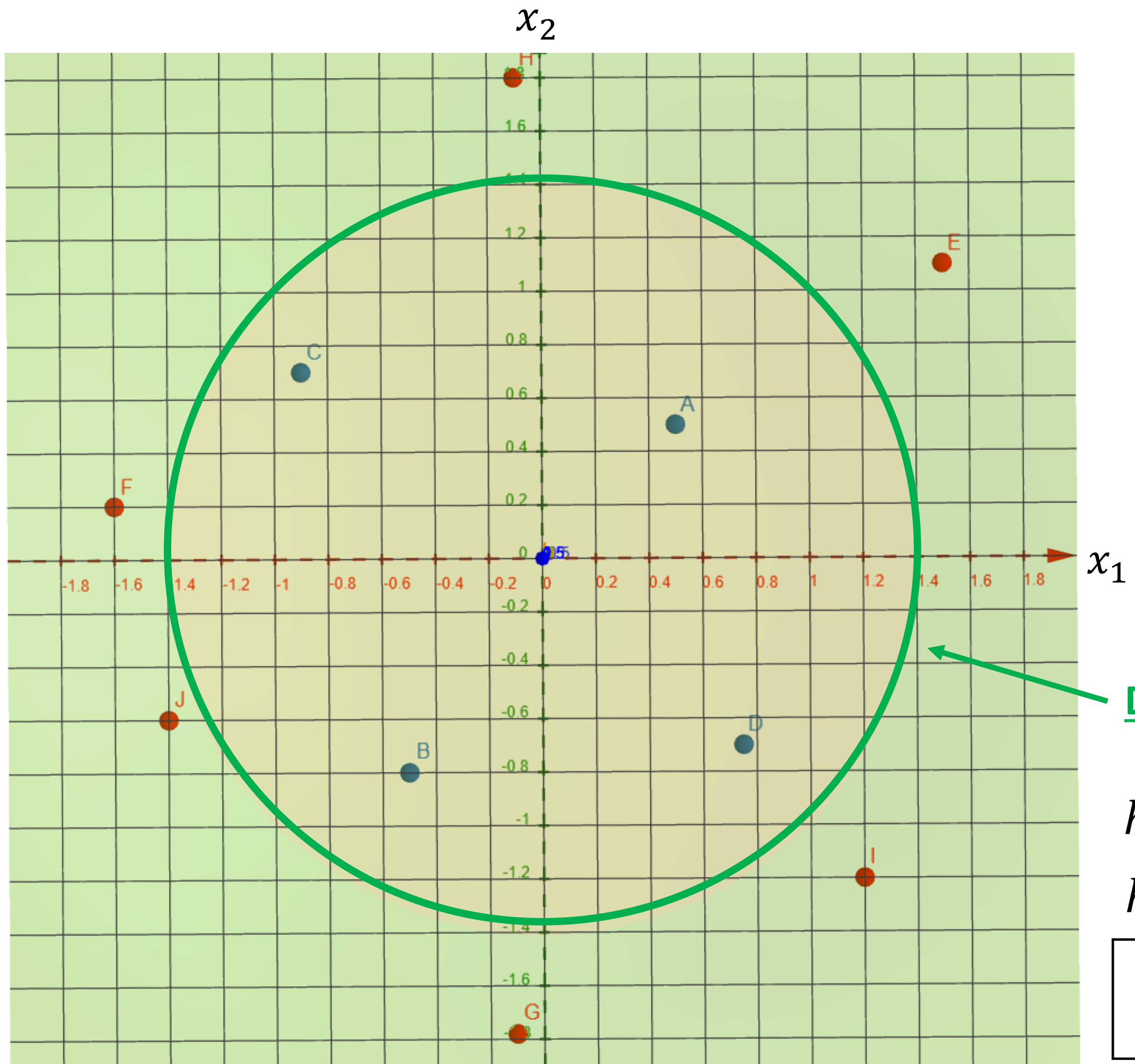
Which function?

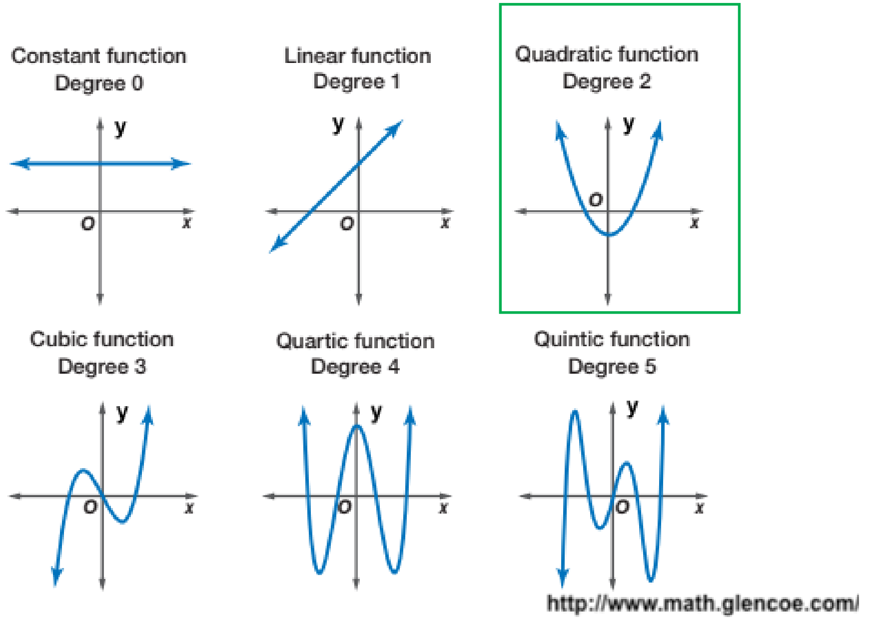
$$h_w(x) = \sigma(w_0 + w_1x_1 + w_2x_2 + w_3x_1^2 + w_4x_2^2)$$

$$h_w(x) = \sigma(-2 + 0x_1 + 0x_2 + 1x_1^2 + 1x_2^2)$$

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

Dealing with Non-Linear Decision Boundary





Which function?

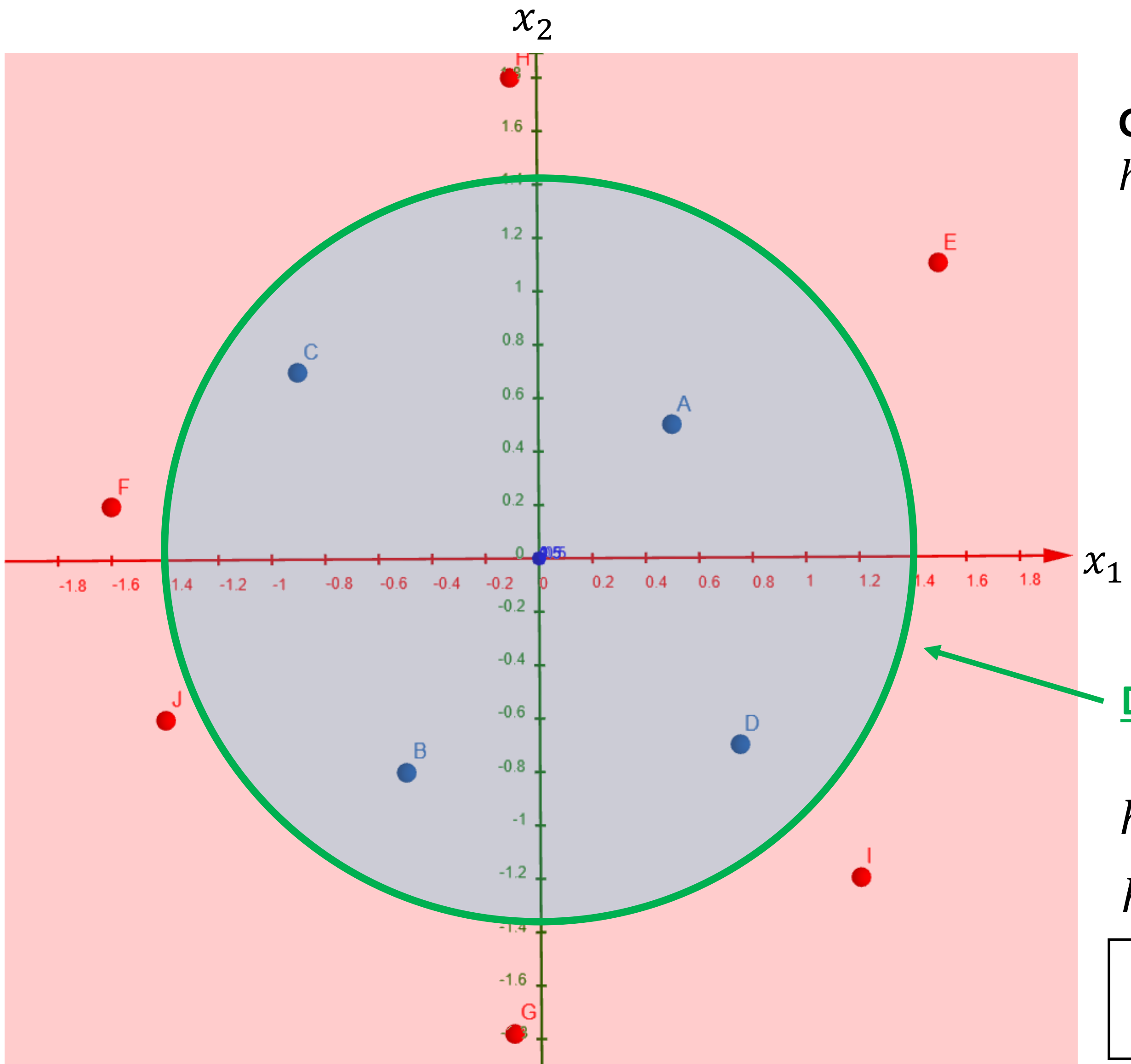
Decision boundary

$$h_w(x) = \sigma(w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1^2 + w_4 x_2^2)$$

$$h_w(x) = \sigma(-2 + 0x_1 + 0x_2 + 1x_1^2 + 1x_2^2)$$

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

Dealing with Non-Linear Decision Boundary



Generally:

$$h_w(x) = w_0 + w_1 f_1 + w_2 f_2 + w_3 f_3 + \cdots + w_n f_n$$

Transformed features:

$e.g., f_1 = x_1, f_2 = x_2, f_1 = x_1^2, f_2 = x_2^2$

Decision boundary

$$h_w(x) = \sigma(w_0 + w_1 x_1 + w_2 x_2 + w_3 x_1^2 + w_4 x_2^2)$$

$$h_w(x) = \sigma(-2 + 0x_1 + 0x_2 + 1x_1^2 + 1x_2^2)$$

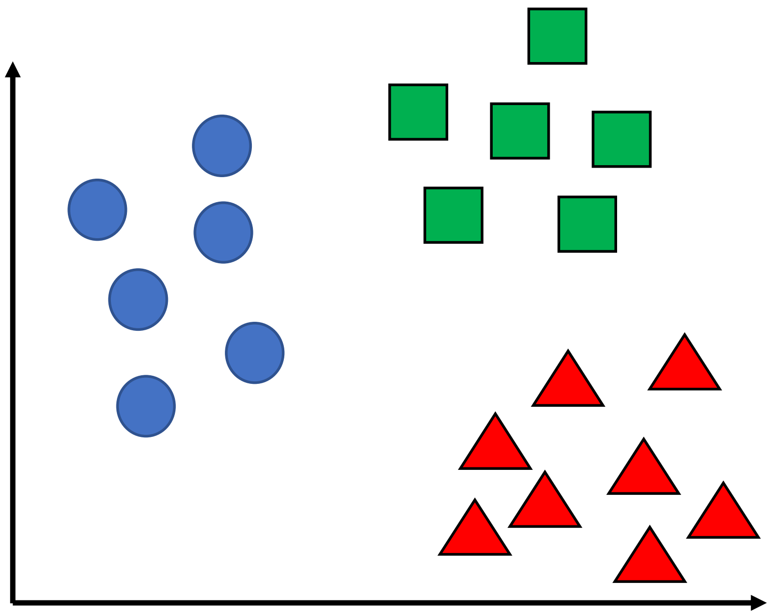
$$\boxed{\sigma(z) = \frac{1}{1 + e^{-z}}}$$

Outline

- Logistic Regression
  - Classification with Continuous Inputs
  - Cross-entropy Loss
  - Logistic Regression with Gradient Descent
- Logistic Regression: Challenges and Solutions
  - Logistic Regression with Many Attributes
  - Dealing with Non-Linear Decision Boundary
- **Multi-class Classification**
- (More) Performance Measure
  - Receiver Operating Characteristic (ROC)
  - Area under ROC (AUC)
- Model Evaluation & Selection
  - Bias & Variance
- Hyperparameter Tuning

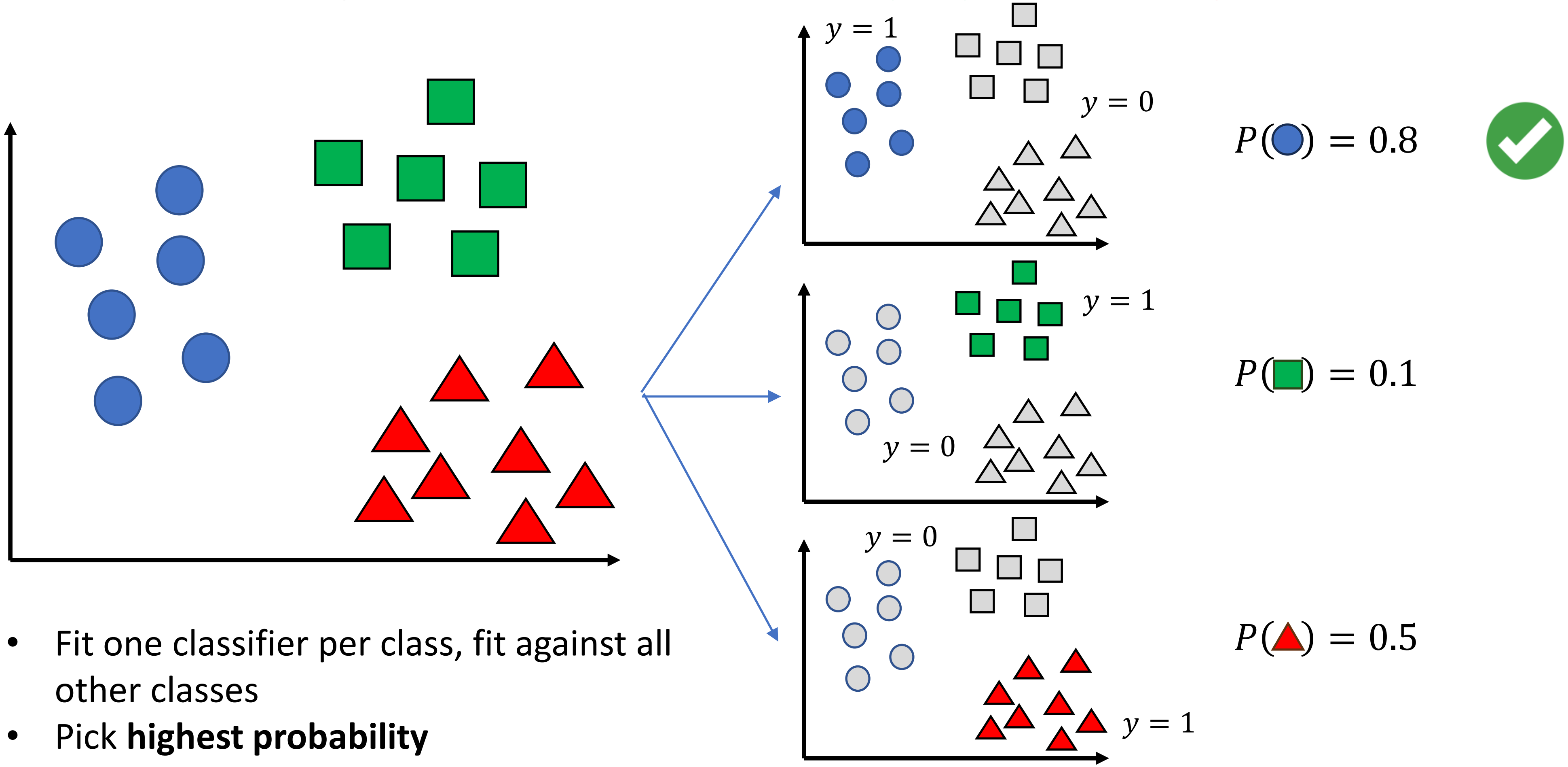
Multi-class Classification

Blood cancer prediction: leukemia, lymphoma, myeloma



Multi-class Classification: One vs **All**

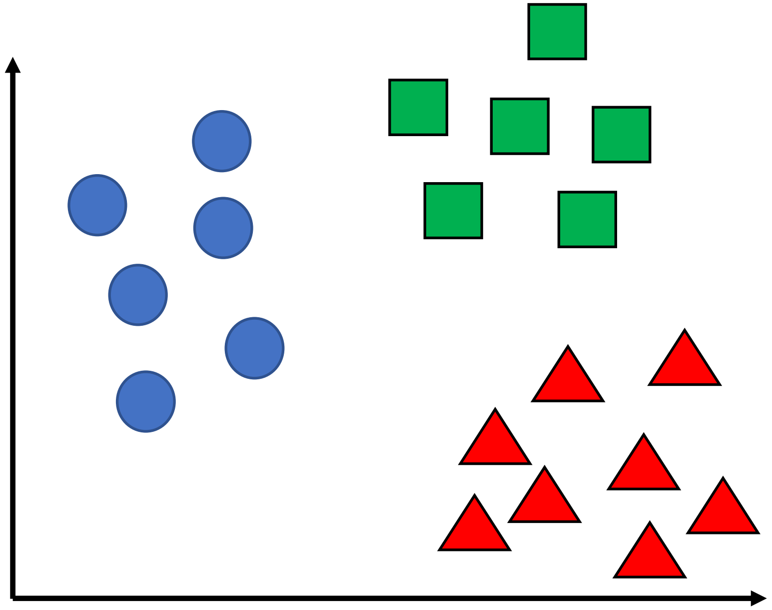
Blood cancer prediction: leukemia, lymphoma, myeloma

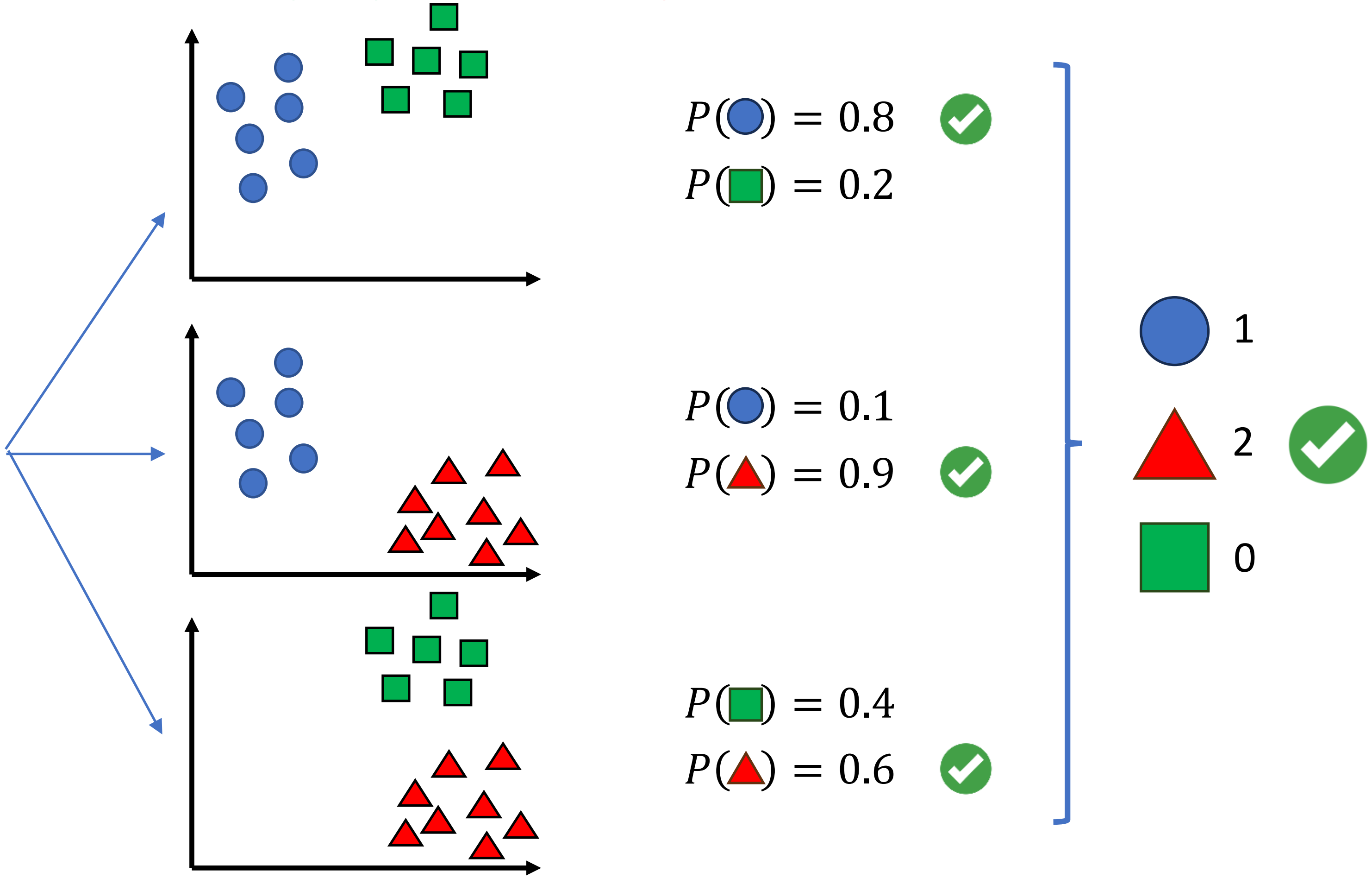


- Fit one classifier per class, fit against all other classes
- Pick **highest probability**

Multi-class Classification: One vs One

Blood cancer prediction: leukemia, lymphoma, myeloma



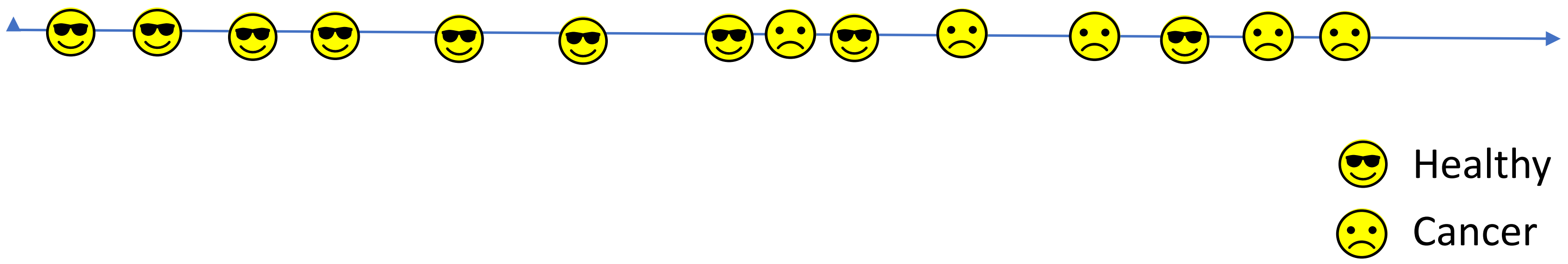


- Fit one classifier per class pair
- Pick **most wins**

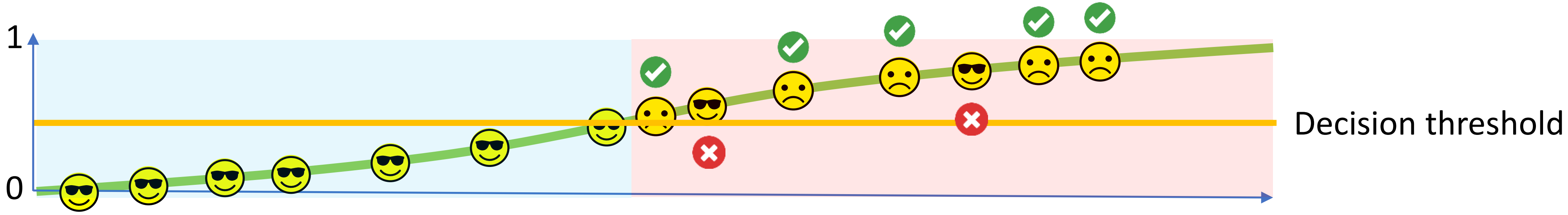
Outline

- Logistic Regression
    - Classification with Continuous Inputs
    - Cross-entropy Loss
    - Logistic Regression with Gradient Descent
- Logistic Regression: Challenges and Solutions
    - Logistic Regression with Many Attributes
    - Dealing with Non-Linear Decision Boundary
- Multi-class classification
- **(More) Performance Measure**
    - Receiver Operating Characteristic (ROC)
    - Area under ROC (AUC)
- Model Evaluation & Selection
    - Bias & Variance
- Hyperparameter Tuning

True Positive Rate & False Positive Rate

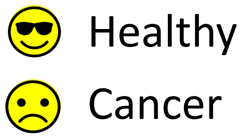


True Positive Rate & False Positive Rate

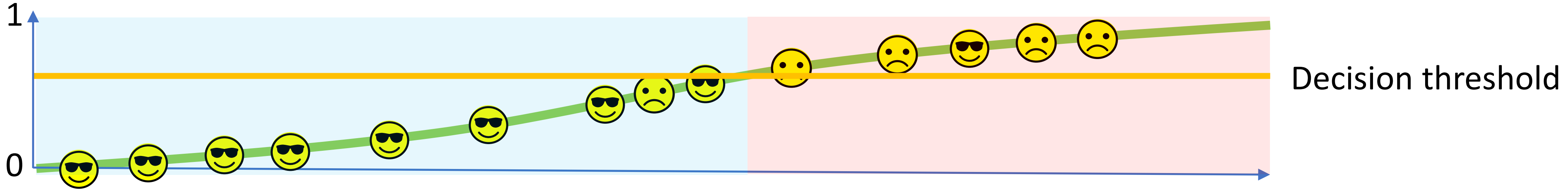


TPR = TP / (TP+FN)   TPR = 5/5 = 1

FPR = FP / (FP+TN)   FPR = 2/9 = 0.22



True Positive Rate & False Positive Rate

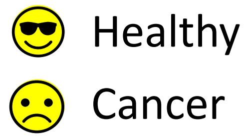


TPR = TP / (TP+FN)

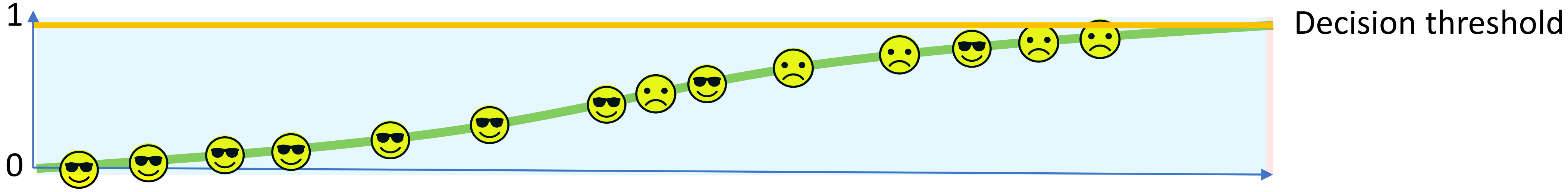
TPR = 4/5 = 0.8

FPR = FP / (FP+TN)

FPR = 1/9 = 0.11



True Positive Rate & False Positive Rate

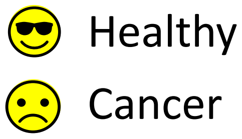


TPR = TP / (TP+FN)

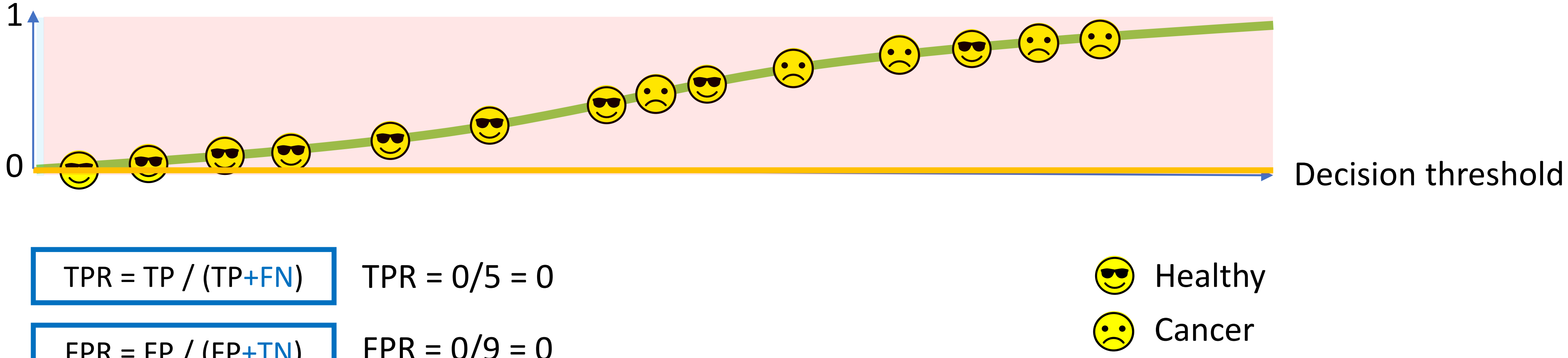
TPR = 0/5 = 0

FPR = FP / (FP+TN)

FPR = 0/9 = 0



True Positive Rate & False Positive Rate



TPR = TP / (TP+FN)

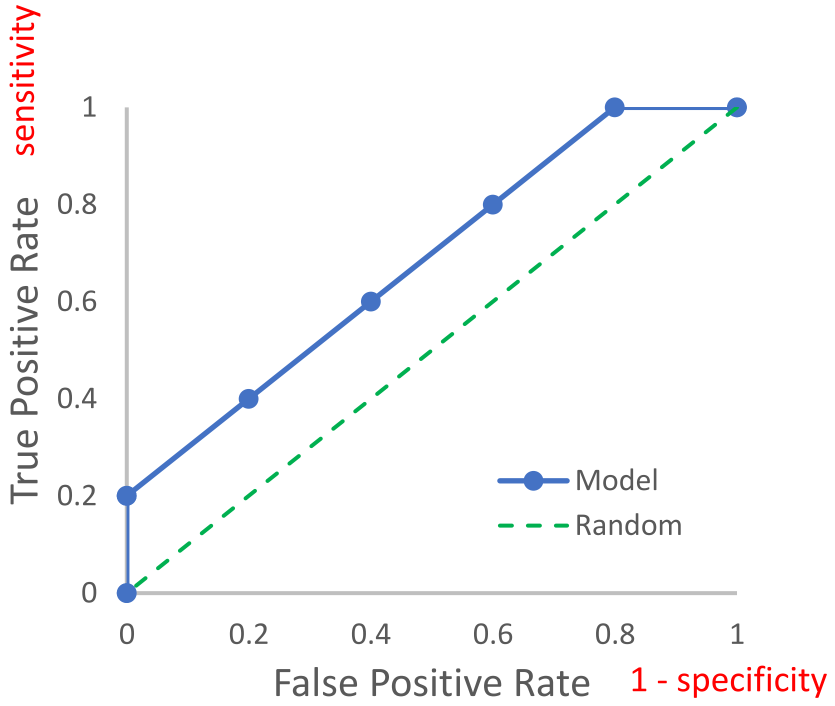
TPR = 0/5 = 0

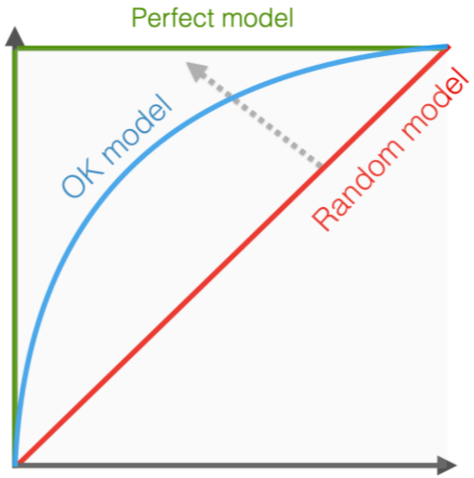
FPR = FP / (FP+TN)

FPR = 0/9 = 0

Receiver Operator Characteristic (ROC) Curve

Threshold π	TPR	FPR
0	1	1
0.1	1.0	1.0
0.2	1.0	0.8
0.3	0.8	0.6
0.4	0.6	0.4
0.5	0.4	0.2
0.6	0.2	0.0
0.7	0.2	0.0
0.8	0.2	0.0
0.9	0.0	0.0
1	0	0



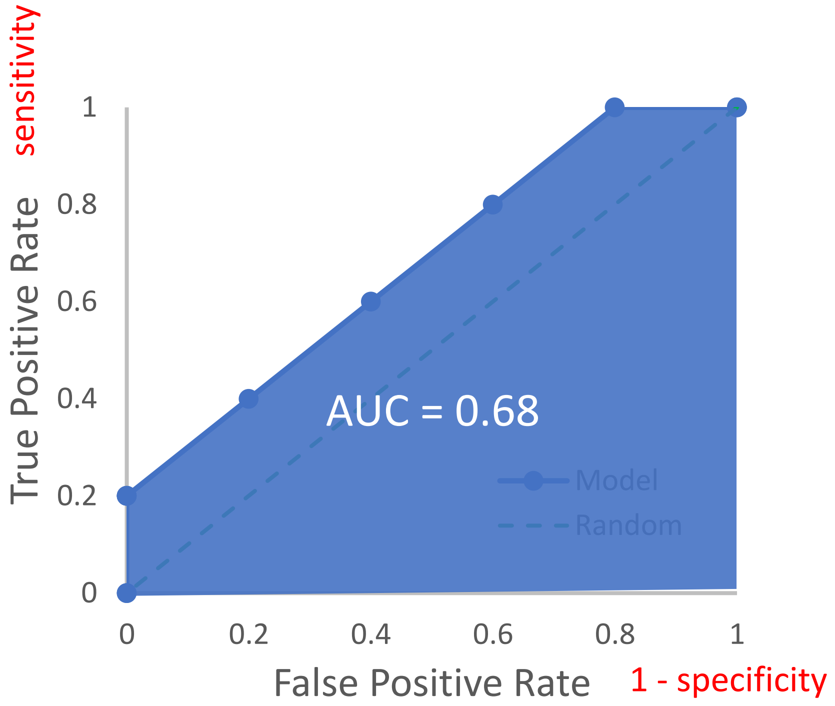


Model is more accurate than random chance
If its **ROC curve** is above the diagonal **random** line.

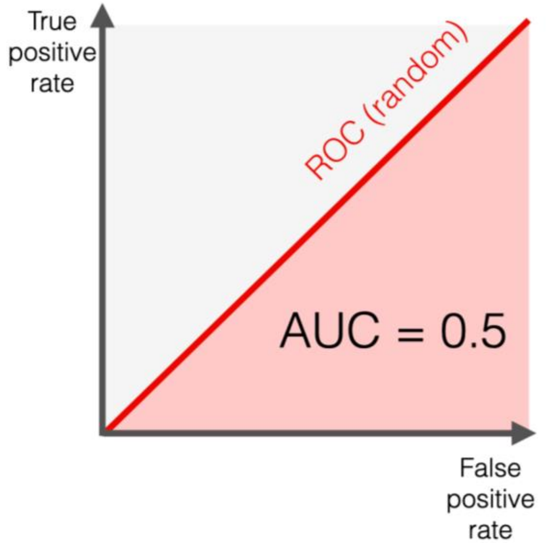
Image credit: https://www.evidentlyai.com/classification-metrics/explain-roc-curve

Area Under Curve (AUC) of ROC

Threshold π	TPR	FPR
0	1	1
0.1	1.0	1.0
0.2	1.0	0.8
0.3	0.8	0.6
0.4	0.6	0.4
0.5	0.4	0.2
0.6	0.2	0.0
0.7	0.2	0.0
0.8	0.2	0.0
0.9	0.0	0.0
1	0	0



AUC is **concise metric** instead of a full figure.
Concise metrics enable *clearer comparisons*.
AUC > 0.5 means the model is better than chance.
AUC ≈ 1 means model is very accurate.



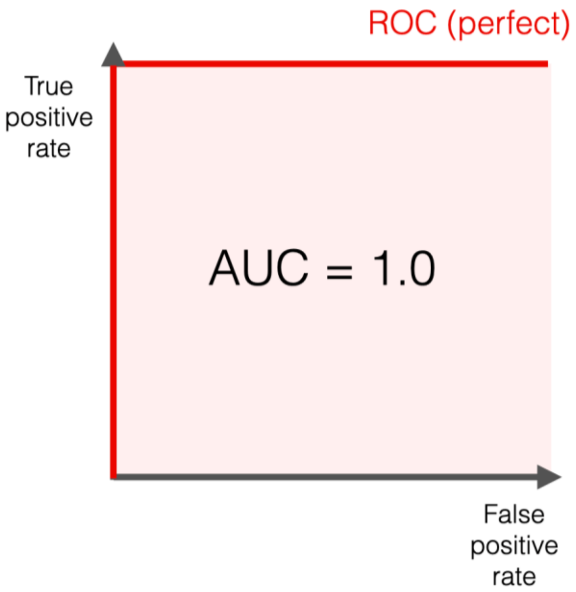


Image credit: https://www.evidentlyai.com/classification-metrics/explain-roc-curve

Outline

- Logistic Regression
  - Classification with Continuous Inputs
  - Cross-entropy Loss
  - Logistic Regression with Gradient Descent
- Logistic Regression: Challenges and Solutions
  - Logistic Regression with Many Attributes
  - Dealing with Non-Linear Decision Boundary
- Multi-class classification
- (More) Performance Measure
  - Receiver Operating Characteristic (ROC)
  - Area under ROC (AUC)
- **Model Evaluation & Selection**
  - Bias & Variance
- Hyperparameter Tuning

Remaining Issues

- How do we pick hyperparameters (e.g., degree of polynomials)?
- How do we decide what features to use?
- … or how do we pick hypothesis
  - Example: decision trees or logistic regression?

Hypothesis evaluation!

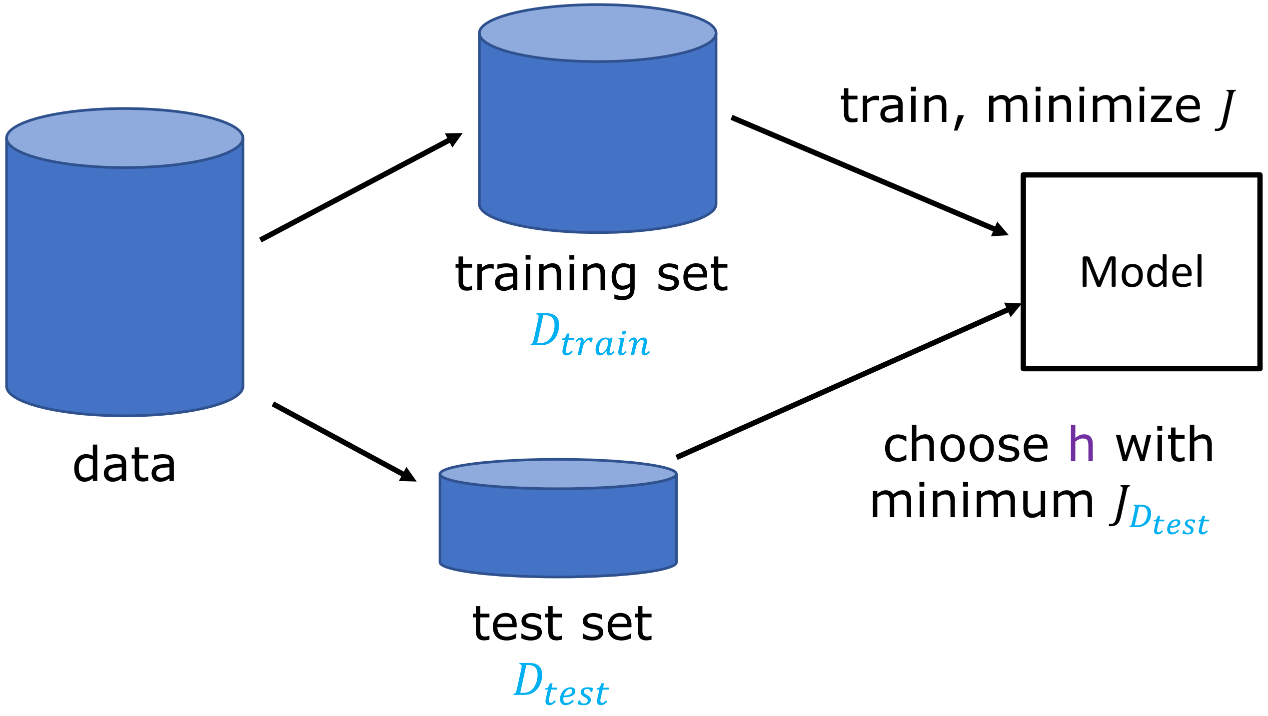
Measuring The Goodness of a Model

MSE, $1 - accuracy$, cross-entropy

Given a dataset D and an error function $error$, the expected error of a model/hypothesis h is measured as follows:

$$J_D(h) = \frac{1}{N}\sum_{i=1}^{N} error\left(h\left(x^{(i)}\right), y^{(i)}\right) \text{, where } \left(x^{(i)}, y^{(i)}\right) \in D$$

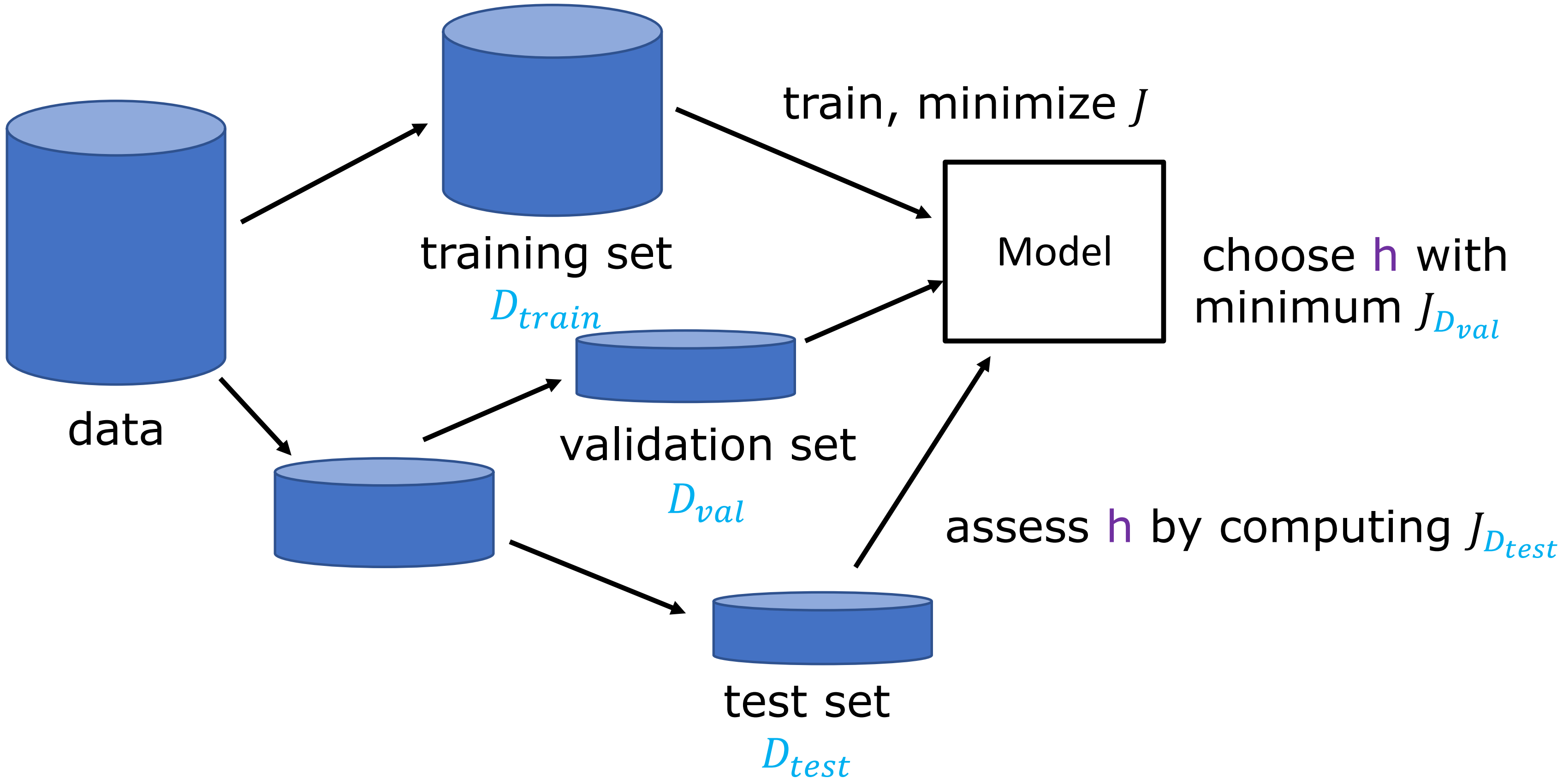
Hypothesis Evaluation: Attempt 1



Can we report the goodness of our model with $J_{D_{test}}$?

No, biased result!

Hypothesis Evaluation: Attempt 2



Example 1: Model selection

1. $y = \sigma(w_0 + w_1 x)$
2. $y = \sigma(w_0 + w_1 x + w_2 x^2)$
3. $y = \sigma(w_0 + w_1 x + w_2 x^2 + w_3 x^3\ )$
4. $y = \sigma(w_0 + w_1 x + w_2 x^2 + w_3 x^3 + w_4 x^4)$
5. $y = \sigma(w_0 + w_1 x + w_2 x^2 + w_3 x^3 + w_4 x^4$
   $+ w_5 x^5\ )$
6. $y = \sigma(w_0 + w_1 x + w_2 x^2 + w_3 x^3 + w_4 x^4$
   $+ w_5 x^5 + w_6 x^6)$

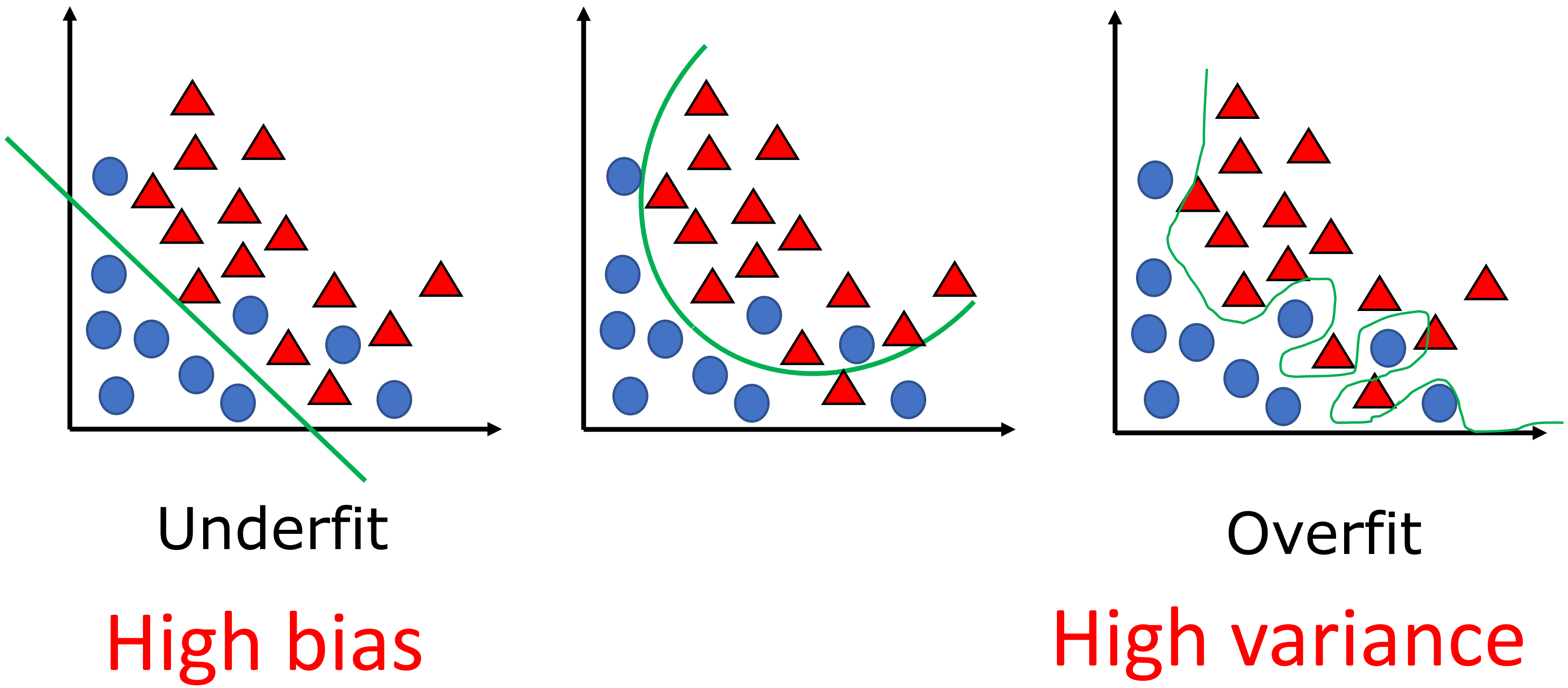
Train each model on D_{train}

Compute $J_{D_{val}}(w)$

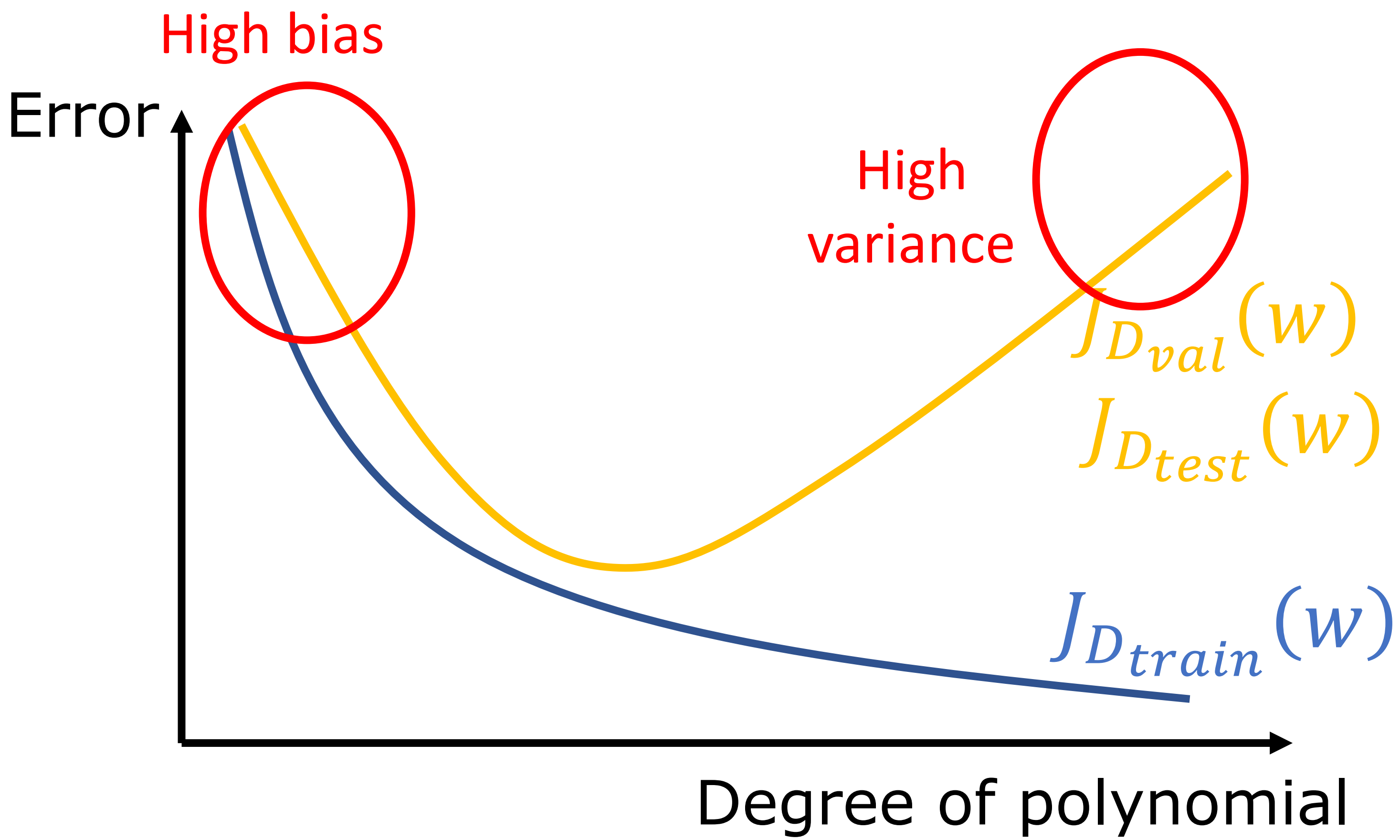
Pick the model with the lowest $J_{D_{val}}(w)$!

Use $J_{D_{test}}(w)$ to estimate performance on unseen samples

Example 1: Bias and Variance



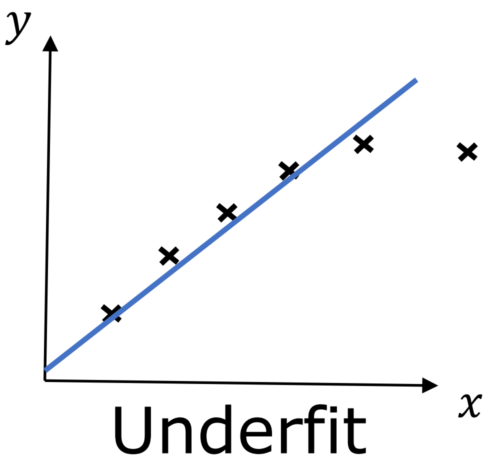
Example 1: Bias and Variance



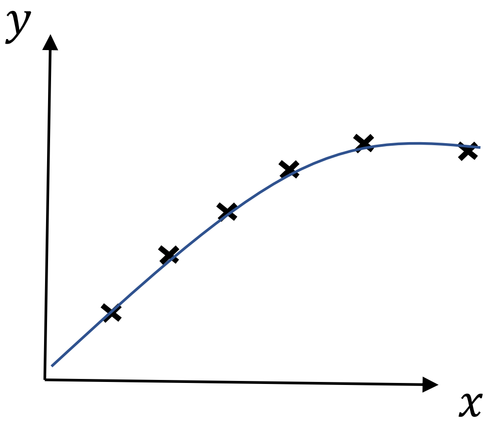
Example 2: Model selection

Polynomial Regression

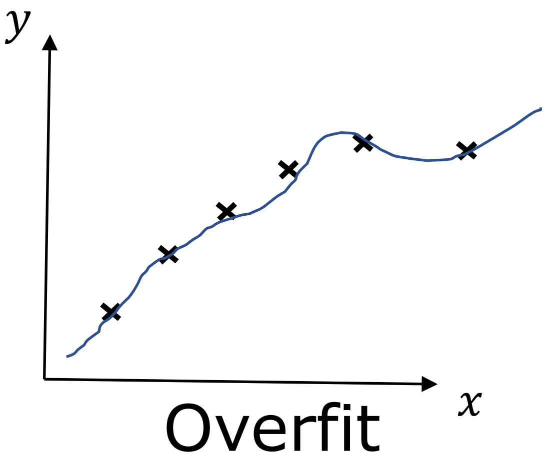
$$J(w) = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_w\left(x^{(i)}\right) - y^{(i)}\right)^2\right]$$



Small degree



Moderate degree



Large degree

Example 2: Model selection

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

$\in D_{train}$

1. Try $degree = 1$
2. Try $degree = 2$
3. Try $degree = 3$
4. Try $degree = 4$
5. Try $degree = 5$
6. …

Train each model on D_{train}

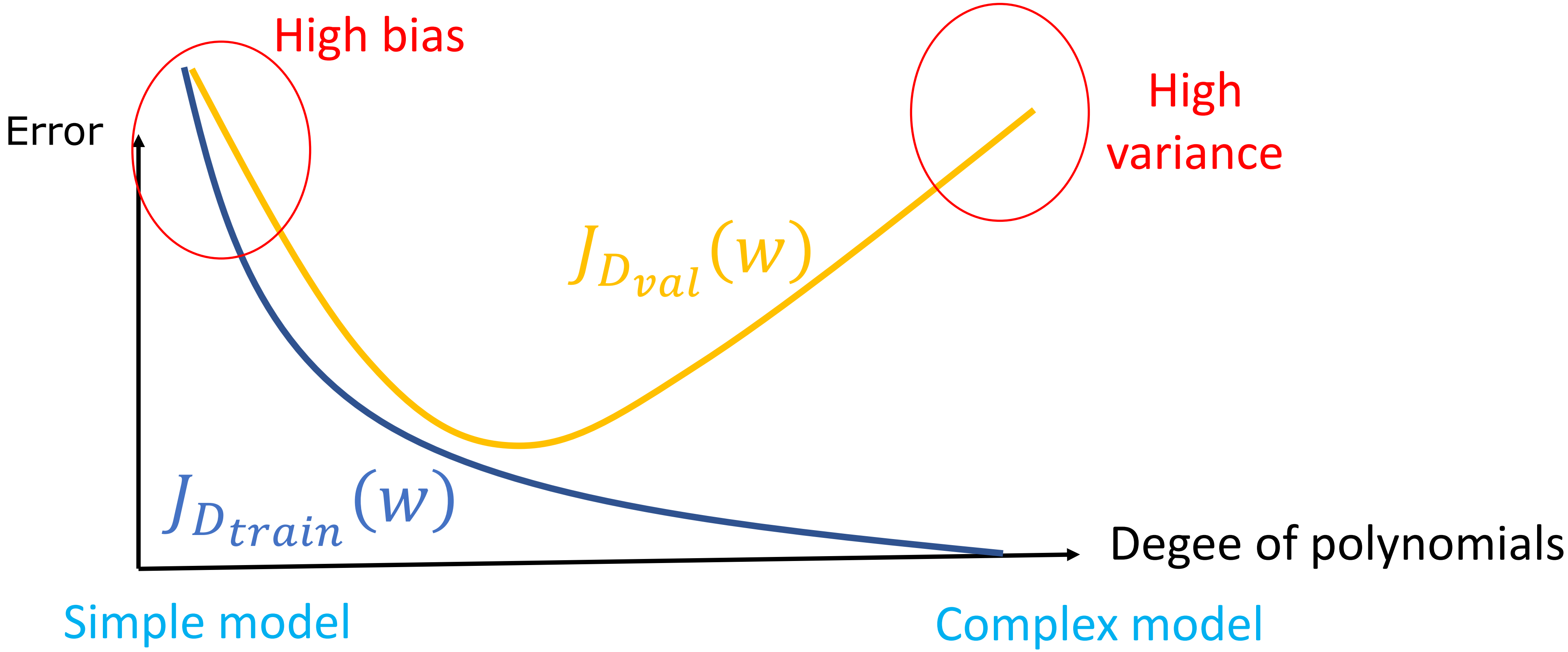
Compute $J_{D_{val}}(w)$

Pick the model with the lowest $J_{D_{val}}(w)$!

Use $J_{D_{test}}(w)$ to estimate performance on unseen samples

Example 2: Bias and Variance

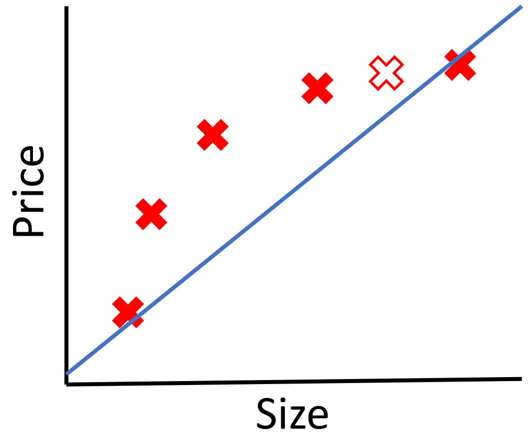
$$J(w) = \frac{1}{2m}\left[\sum_{i=1}^{m}\left(h_w\left(x^{(i)}\right) - y^{(i)}\right)^2\right]$$



Diagnosing Bias and Variance

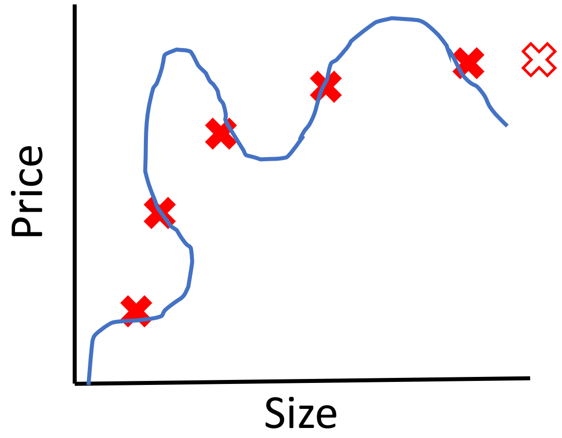
- Bias (undefit):

$$J_{D_{train}}(w) \text{ will be high}$$
$$J_{D_{train}}(w) \approx J_{D_{val}}(w)$$



- Variance (overfit):

$$J_{D_{train}}(w) \text{ will be low}$$
$$J_{D_{val}}(w) \gg J_{D_{train}}(w)$$



Outline

- Logistic Regression
  - Classification with Continuous Inputs
  - Cross-entropy Loss
  - Logistic Regression with Gradient Descent
- Logistic Regression: Challenges and Solutions
  - Logistic Regression with Many Attributes
  - Dealing with Non-Linear Decision Boundary
- Multi-class classification
- (More) Performance Measure
  - Receiver Operating Characteristic (ROC)
  - Area under ROC (AUC)
- Model Evaluation & Selection
  - Bias & Variance
- **Hyperparameter Tuning**

Hyperparameter Tuning

Finding the best model

Loop (Basic):

1. Pick hyperparameters
   - Examples: degree of polynomials, max-depth pruning
2. Train model with the hyperparameters
3. Evaluate model

Hyperparameter Tuning: Methods

- **Grid search (exhaustive search)**
  - Exhaustively try all possible hyperparameters
- **Random search**
  - Randomly select hyperparameters
- **Successive halving**
  - Use all possible hyperparameters but with reduced resources
  - Successively increase the resources with smaller set of hyperparameters
- **Bayesian optimization**
  - Use Bayesian methods to estimate the optimization space of the hyperparameters
- **Evolutionary algorithms**
  - Use evolutionary algorithms (e.g., genetic algo) to select a population of hyperparameters

Many "off-the-shelves" packages available

Summary

- Logistic Regression
  - Classification with Continuous Inputs: **draw a line** that **separates** classes well
  - Cross-entropy Loss: $\frac{1}{m}\sum_{i=1}^{m} BCE\left(h_w\left(x^{(i)}\right), y^{(i)}\right)$, $BCE(y, \hat{y}) = -y\log(\hat{y}) - (1-y)\log(1-\hat{y})$
  - Logistic Regression with Gradient Descent – same as Linear Regression
- Logistic Regression: Challenges and Solutions
  - Logistic Regression with Many Attributes – $\sigma(w^T x)$, similar to Linear Regression
  - Dealing with Non-Linear Decision Boundary – **feature transformations**, similar to Linear Regression
- Multi-class classification: one-vs-all, one-vs-one
- (More) Performance Measure
  - Receiver Operating Characteristic (ROC): TPR vs FPR
  - Area under ROC (AUC): summary of ROC
- Model Evaluation & Selection
  - Bias (model capacity) & Variance (model randomness)
- Hyperparameter Tuning

Coming Up Next Week

Recess Week! … then **Midterm** assessment

Lecture 7:

- Regularizations
- Support Vector Machines
- Kernels

To Do

- **Lecture Training 6**
  - +100 Free EXP
  - +50 Early bird bonus