Introduction to Data Science

(Lecture 12)

Dr. Mohammad Pourhomayoun
Assistant Professor
Computer Science Department
California State University, Los Angeles

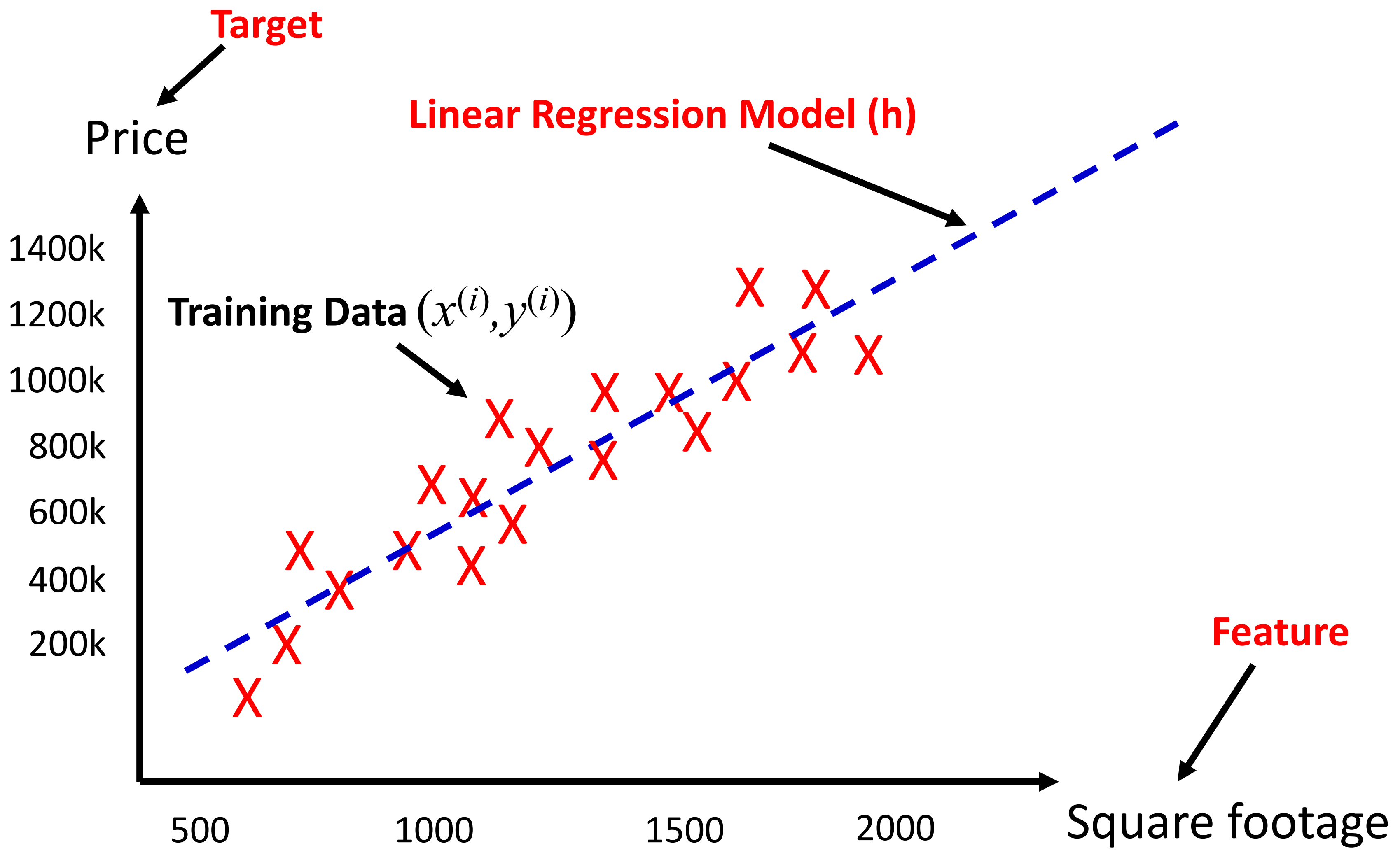
Logistic Regression Classifier

Review

- **Classification**: Predict a discrete valued output for each item.
  – Labels are discrete (categorical)
  – Labels can be binary (e.g., rainy/sunny, spam/non-spam,) or non-binary (e.g., rainy/sunny/cloudy, Setosa/Versicolor/Virginica)

- **Regression:** Predict a continuous valued output for each item.
  – Labels are continuous (numeric), e.g., stock price, housing price
  – Can define ‘closeness’ when comparing prediction with true values

Regression Example: Housing Price



Deriving a Classifier from a Regression Model

1. We learned how to build a linear regression model to predict continuous-valued outputs.

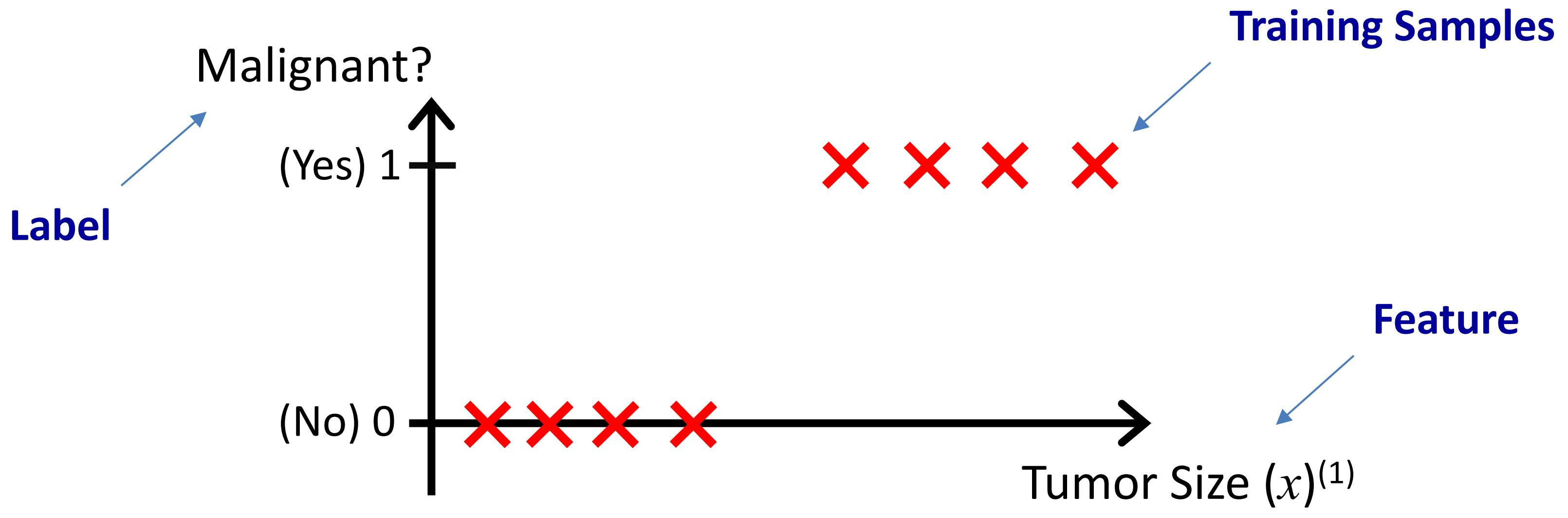
2. Well, in theory we can convert Categorical Labels into Numerical Labels:
   - Sunny → 0 , Rainy → 1 thus: Sunny/Rainy → 0/1
   - Not-Cancer → 0 , Cancer → 1 thus: Not-Cancer / Cancer → 0/1

3. Thus, Let's take advantage of the linear regression model to develop a Classifier! (We will see that discretizing the output of a linear regression is not always a good idea! So, we need to define new hypothesis model and a new cost function)

4. Since it is a classifier built based on linear regression algorithm, we call it: "Logistic Regression"!

Logistic Regression Classifier with One Feature

Example: Predicting if a Tumor is Malignant or Not based on tumor size (so, we need a classifier):

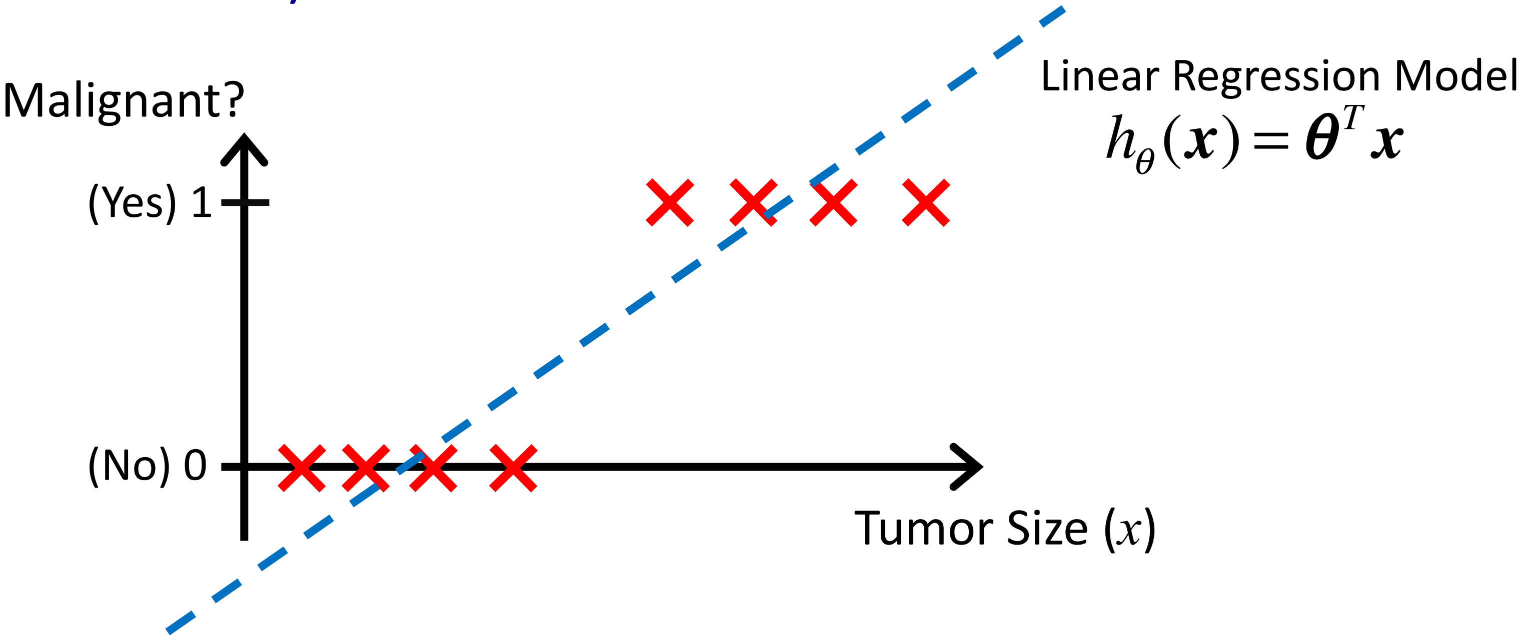


This is a discrete-valued data. Nonetheless, let's apply our Linear Regression model on it!

* Example from Andrew Ng, Machine Learning, Stanford University.

Logistic Regression Classifier with One Feature

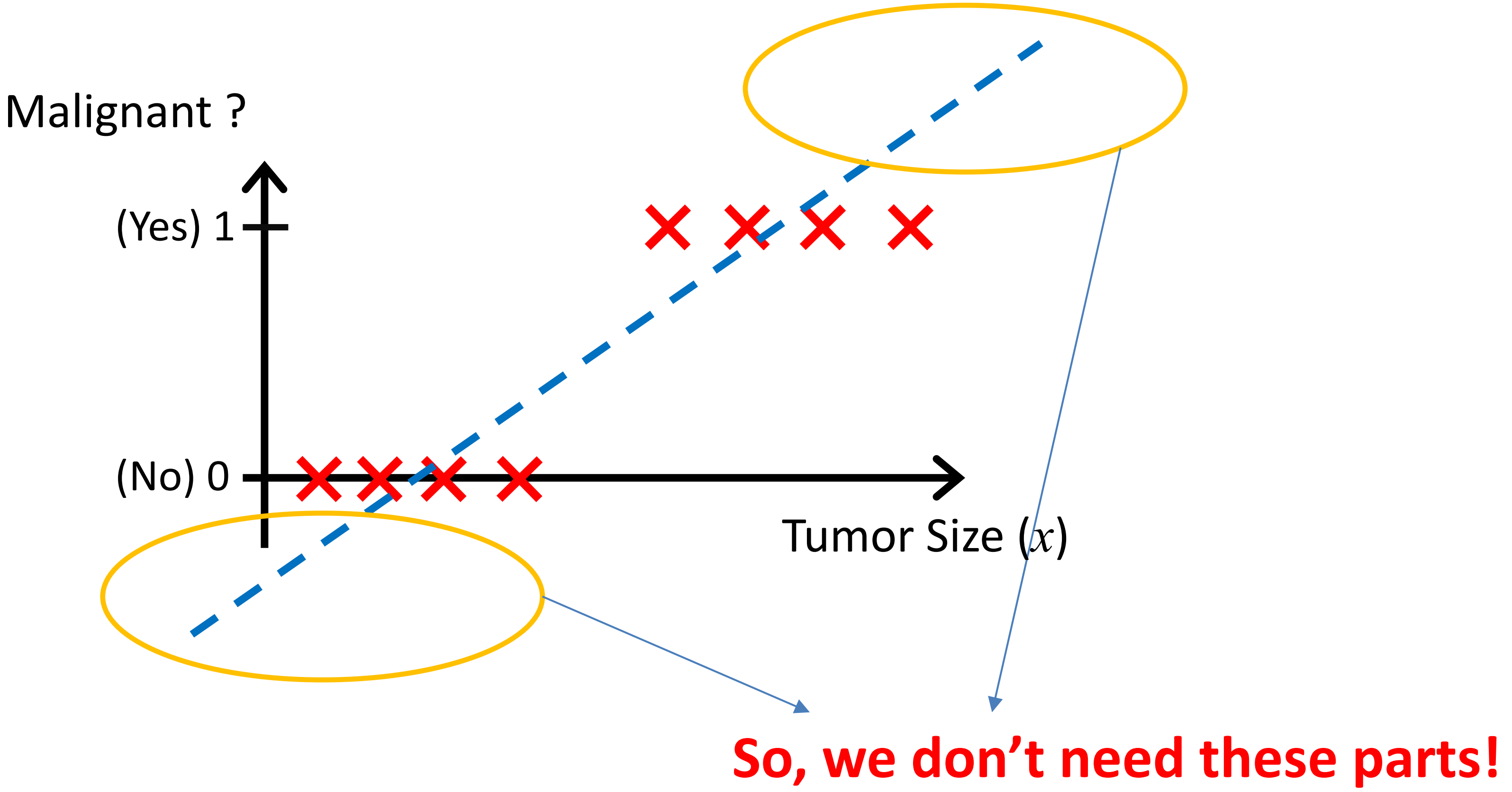
Example: Predicting if a Tumor is Malignant or Not based on tumor size (so, we need a classifier):



This is a discrete-valued data. Nonetheless, let's apply our Linear Regression model on it!

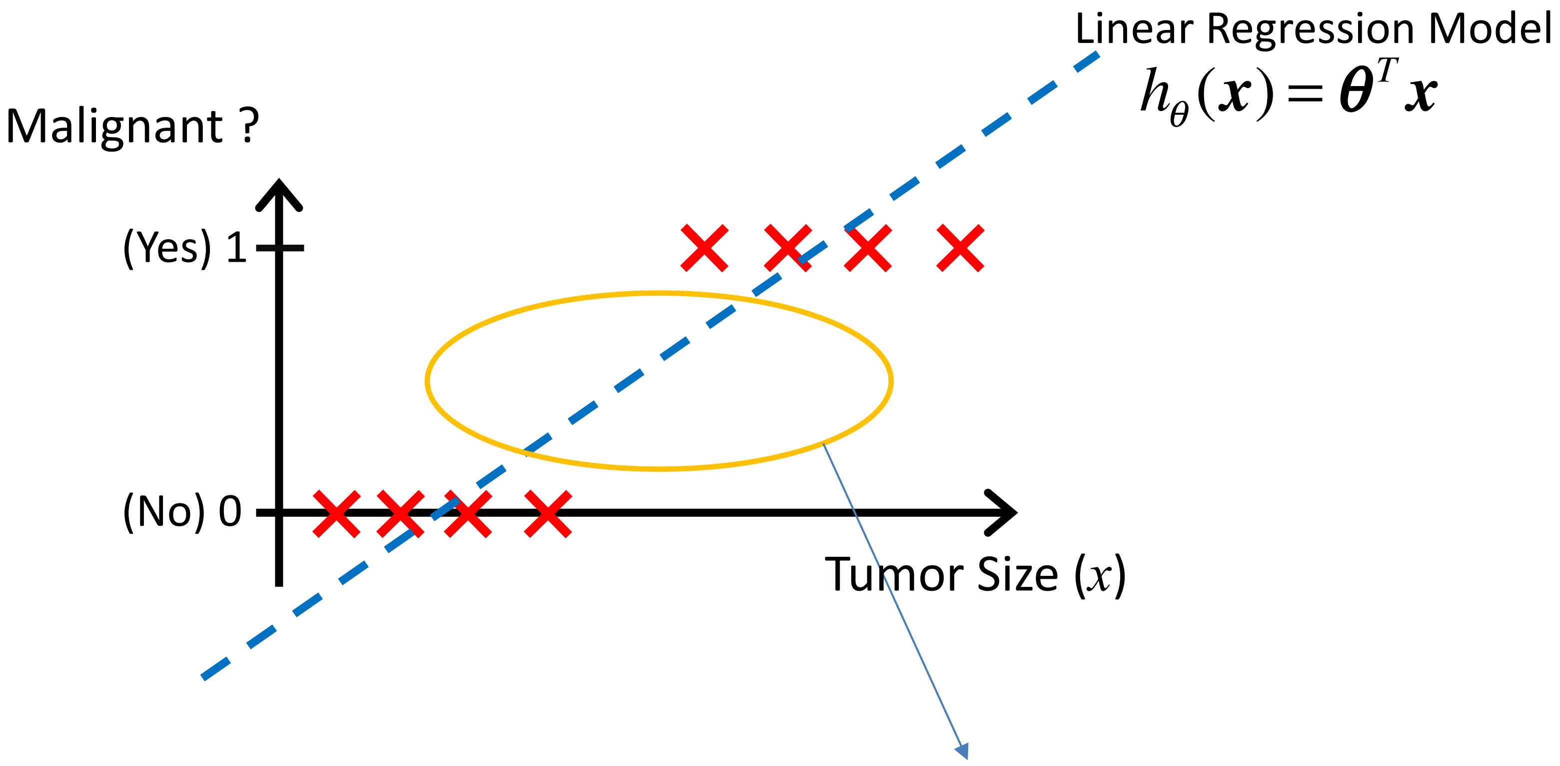
Logistic Regression Classifier with One Feature

- we know that for our classifier, the **output should be either 1 or 0**!



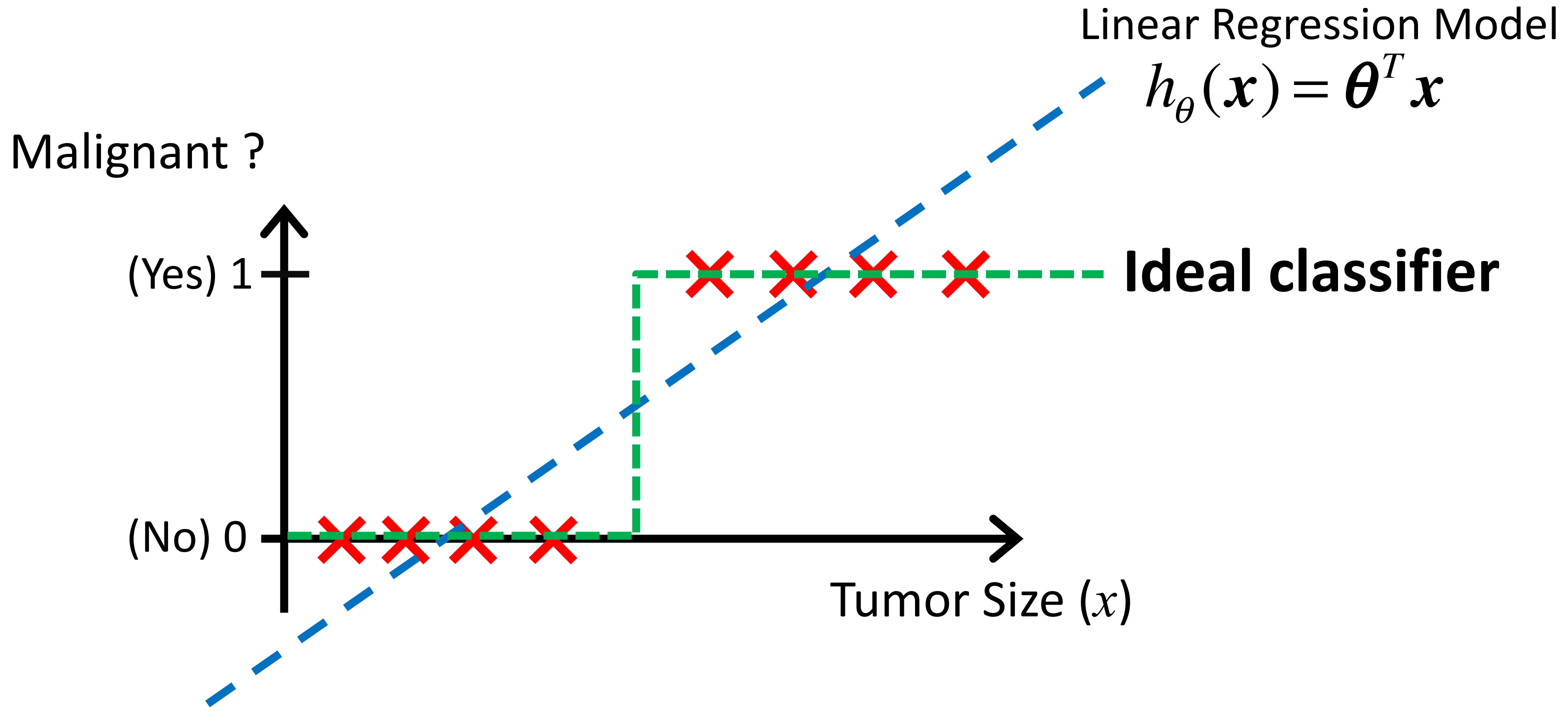
Logistic Regression Classifier with One Feature

- we know that for our classifier, the **output should be either 1 or 0**!



So, We also need a sharp transition here

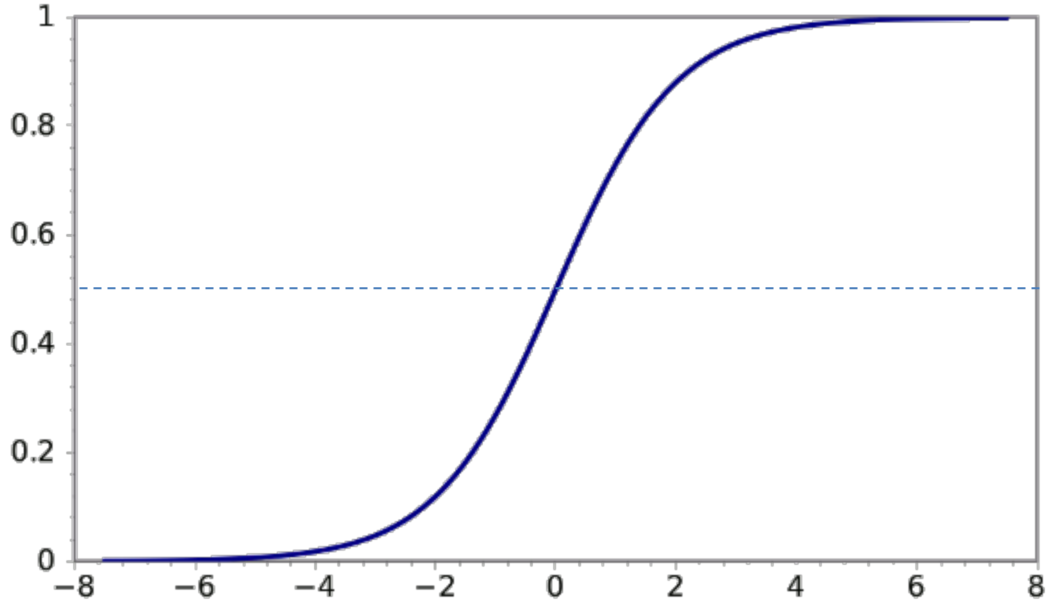
Logistic Regression Classifier with One Feature



Sigmoid Function

- Sigmoid Function (Logistic Function):

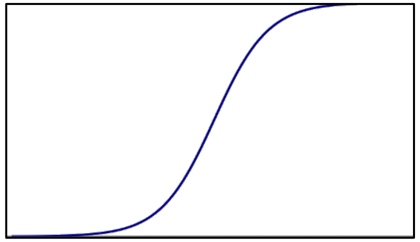
$$g(z) = \frac{1}{1 + e^{-z}}$$



Logistic Regression Model

- Using Sigmoid Function (Logistic Function):

Linear Regression



limited to the range of [0,1]

$$h_\theta(\boldsymbol{x}) = \boldsymbol{\theta}^T \boldsymbol{x}$$

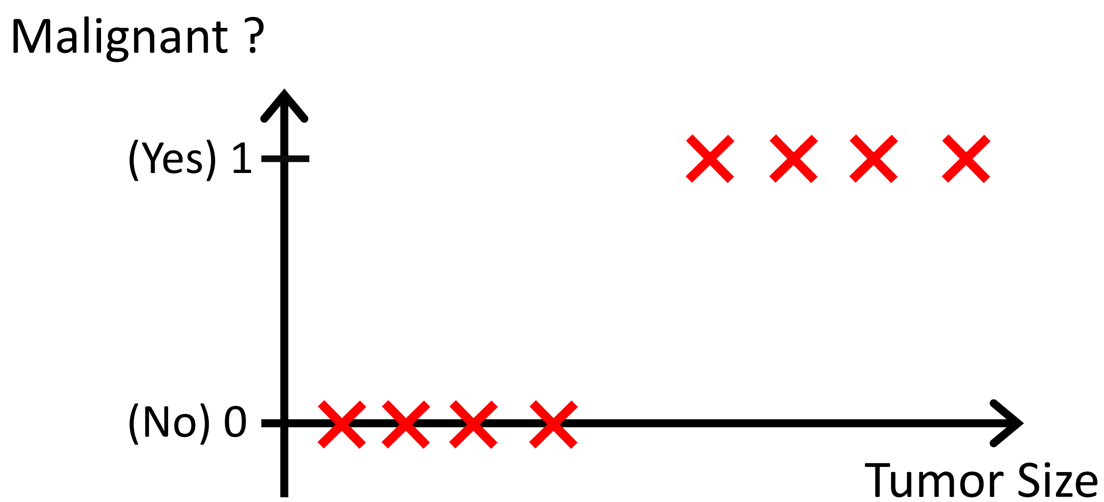
$$g(z) = \frac{1}{1 + e^{-z}}$$

$$g(\boldsymbol{\theta}^T \boldsymbol{x}) = \frac{1}{1 + e^{-\boldsymbol{\theta}^T \boldsymbol{x}}}$$

New approach for output prediction:

$$h_\theta(\boldsymbol{x}) = g(\boldsymbol{\theta}^T \boldsymbol{x}) = \frac{1}{1 + e^{-\boldsymbol{\theta}^T \boldsymbol{x}}}$$

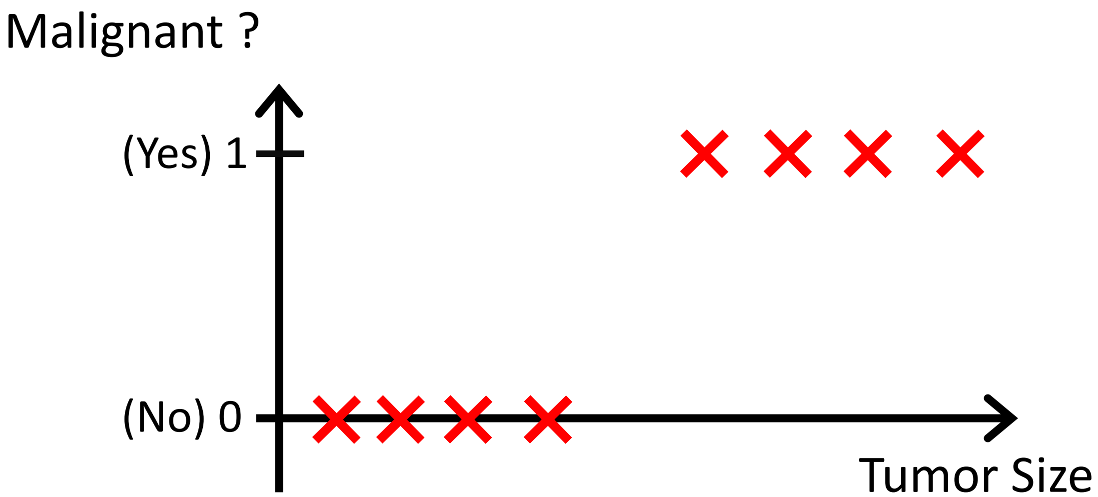
So, Now the NEW $h_\theta(x)$ is limited to the range of [0,1].



- New approach: $h_\theta(\boldsymbol{x}) = g(\boldsymbol{\theta}^T \boldsymbol{x}) = \dfrac{1}{1+e^{-\boldsymbol{\theta}^T \boldsymbol{x}}}$

- After applying the Sigmoid function, $h_\theta(\boldsymbol{x})$ will be limited in the range of [0,1]. But, still can take any value between 0 and 1!

- **So, it is like representing the Probability of happening each label!**
  - **E.g. 30% chance of rain, 5% chance of malignant cancer, …**



- New approach: $h_\theta(\boldsymbol{x}) = g(\boldsymbol{\theta}^T \boldsymbol{x}) = \dfrac{1}{1+e^{-\boldsymbol{\theta}^T \boldsymbol{x}}}$

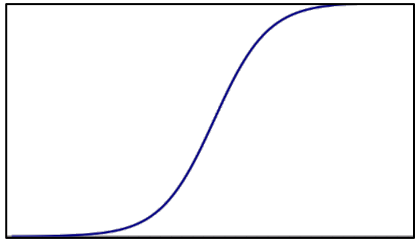
- After applying the Sigmoid function, $h_\theta(\boldsymbol{x})$ will be limited to the range of [0,1]. But, still can take any value in this range!
- **Now, to generate class output (binary output), we can compare the results with a threshold (e.g. 0.5) to discretize the output:**

$$\begin{cases} \text{predict "}y = 1\text{" if } h_\theta(\boldsymbol{x}) \geq 0.5 \\ \text{predict "}y = 0\text{" if } h_\theta(\boldsymbol{x}) \leq 0.5 \end{cases}$$

After limiting the range and discretizing

Linear Regression

$$h_\theta(\boldsymbol{x}) = \boldsymbol{\theta}^T \boldsymbol{x}$$



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

limited to the range of [0,1]

$$g(\boldsymbol{\theta}^T \boldsymbol{x}) = \frac{1}{1 + e^{-\boldsymbol{\theta}^T \boldsymbol{x}}}$$

> 0.5

Y → Malignant Cancer

N → Not Malignant Cancer

How to Select the Parameters?

Training set (m training samples):

$$\left\{(\boldsymbol{x}^{(1)}, y^{(1)}), (\boldsymbol{x}^{(2)}, y^{(2)}), \ldots, (\boldsymbol{x}^{(m)}, y^{(m)})\right\}$$

Feature Vector: $\boldsymbol{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$ $\quad x_0 = 1, y \in \{0, 1\}$

$$h_\theta(\boldsymbol{x}) = g(\boldsymbol{\theta}^T \boldsymbol{x}) = \frac{1}{1 + e^{-\boldsymbol{\theta}^T \boldsymbol{x}}}$$

$$\boldsymbol{\theta} = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{bmatrix}$$

How to choose parameters θ?

Gradient Descent for Logistic Regression

$$h_\theta(\boldsymbol{x}) = g(\boldsymbol{\theta}^T \boldsymbol{x}) = \frac{1}{1+e^{-\boldsymbol{\theta}^T \boldsymbol{x}}} \qquad \boldsymbol{\theta} = \begin{bmatrix} \theta_0 \\ \theta_1 \\ \vdots \\ \theta_n \end{bmatrix} \qquad \boldsymbol{x} = \begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_n \end{bmatrix}$$

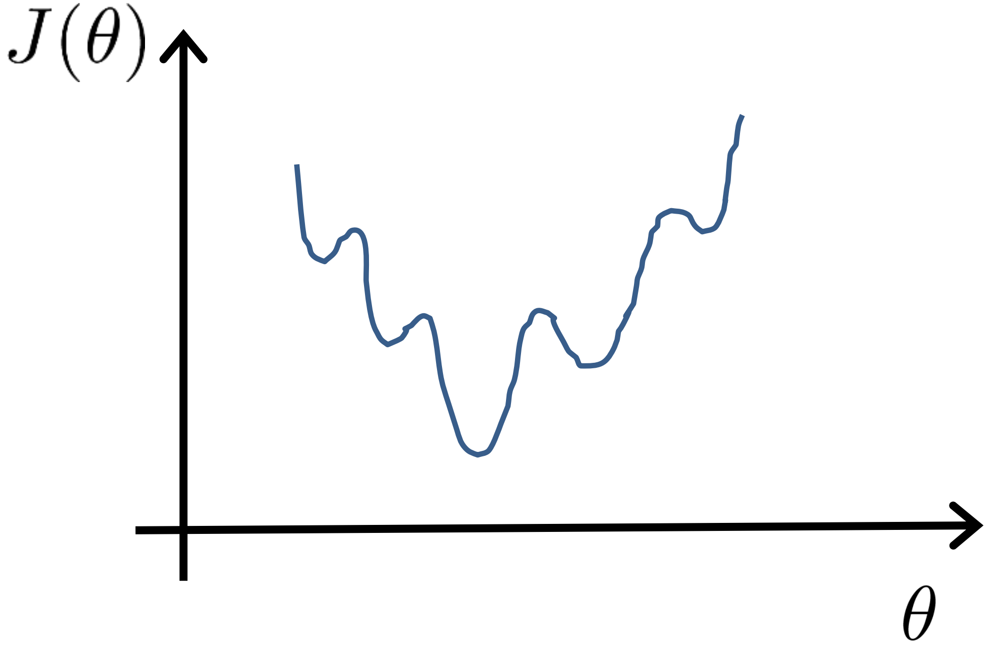
- **Previous definition of Cost function for linear regression:**

$$J(\boldsymbol{\theta}) = J(\theta_0, \theta_1, \theta_2, \ldots, \theta_n) = \frac{1}{2m} \sum_{i=1}^{m} (h_\theta(\boldsymbol{x}^{(i)}) - y^{(i)})^2$$

Logistic Regression Cost Function

- Previous definition of Cost function for linear regression:

$$J(\boldsymbol{\theta}) = J(\theta_0, \theta_1, \theta_2, \ldots, \theta_n) = \frac{1}{2m}\sum_{i=1}^{m}(h_\theta(\boldsymbol{x}^{(i)}) - y^{(i)})^2$$



$$h_\theta(\boldsymbol{x}) = g(\boldsymbol{\theta}^T\boldsymbol{x}) = \frac{1}{1+e^{-\boldsymbol{\theta}^T\boldsymbol{x}}}$$

This Cost Function is not Convex for Logistic Regression (It has many local minimums)!

- Since the Cost Function with this definition **is not convex** for logistic regression, there is **no guarantee** for gradient descent to find the **global minimum** for error. Thus, we take advantage of a "**log**" function to define an alternative **convex** cost function (The mathematical details why *log* is a good option is beyond the scope of the class! But, an intuitive idea is available in next page!).

- **New Cost function:**

$$J(\boldsymbol{\theta}) = J(\theta_0, \theta_1, \dots, \theta_n)$$
$$= -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log h_\theta(\boldsymbol{x}^{(i)}) + (1 - y^{(i)})\log(1 - h_\theta(\boldsymbol{x}^{(i)}))\right]$$

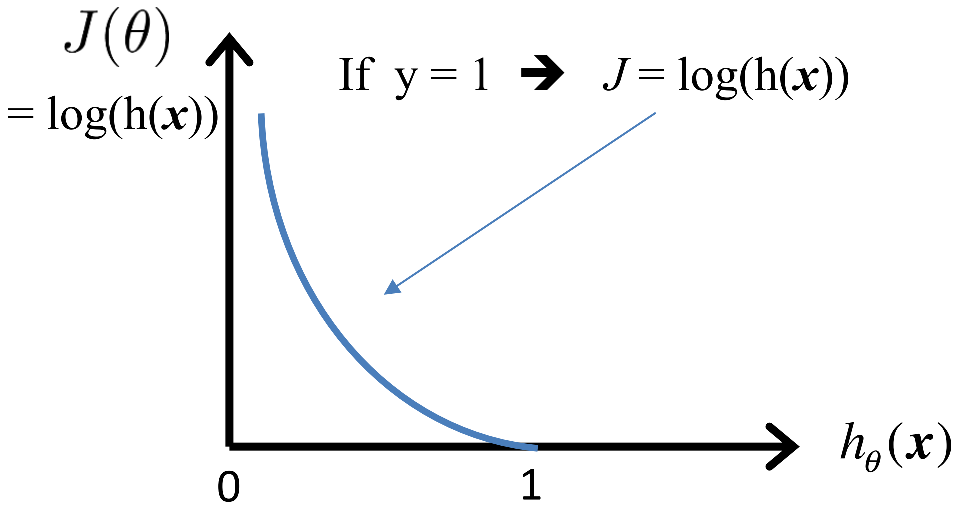
Note: $y = 0$ or 1 always

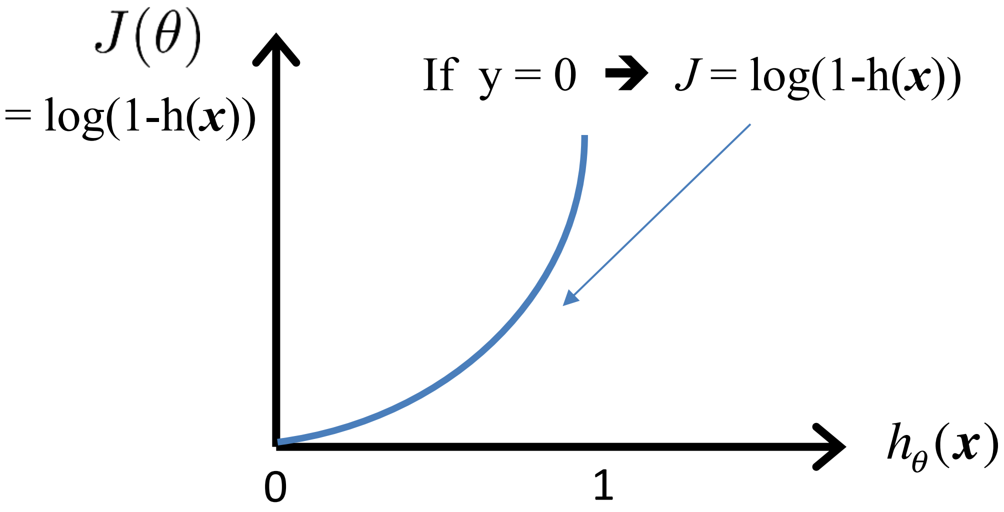
Optional

- **Why does this Cost function work?**

Note: $y = 0$ or 1 always

$$J(\boldsymbol{\theta}) = J(\theta_0, \theta_1, \ldots, \theta_n)$$
$$= -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log h_\theta(\boldsymbol{x}^{(i)}) + (1-y^{(i)})\log(1-h_\theta(\boldsymbol{x}^{(i)}))\right]$$





- when y=1, and h(x) is close to 1, the cost will be very small, and when y=1, and h(x) is close to 0, the cost will be too high.
- when y=0, and h(x) is close to 0, the cost will be very small, and when y=0, and h(x) is close to 1, the cost will be too high.

Logistic Regression Cost Function

Cost Function:

$$J(\boldsymbol{\theta}) = J(\theta_0, \theta_1, \ldots, \theta_n)$$

$$= -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log h_\theta(\boldsymbol{x}^{(i)}) + (1-y^{(i)})\log(1-h_\theta(\boldsymbol{x}^{(i)}))\right]$$

To found the best parameters θ :

$$\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$$

Gradient Descent for Multiple Variables

- Output (Probability): $h_\theta(\boldsymbol{x}) = \dfrac{1}{1+e^{-\boldsymbol{\theta}^T \boldsymbol{x}}}$

- Want $\min_{\boldsymbol{\theta}} J(\boldsymbol{\theta})$ to find $\boldsymbol{\theta}$

- Cost function: $J(\boldsymbol{\theta}) = -\dfrac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log h_\theta(\boldsymbol{x}^{(i)}) + (1-y^{(i)})\log(1-h_\theta(\boldsymbol{x}^{(i)}))\right]$

- **Gradient descent:**

Repeat $\{$

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \ldots, \theta_n)$$

$\}$

(simultaneously update for every $j = 0, \ldots, n$)

Gradient Descent for Multiple Variables

- Output (probability): $h_\theta(\boldsymbol{x}) = \dfrac{1}{1+e^{-\boldsymbol{\theta}^T \boldsymbol{x}}}$
- Cost function: $J(\boldsymbol{\theta}) = -\dfrac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log h_\theta(\boldsymbol{x}^{(i)}) + (1-y^{(i)})\log(1-h_\theta(\boldsymbol{x}^{(i)}))\right]$
- **Gradient descent:**

Repeat {

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta_0, \ldots, \theta_n)$$

$$= \theta_j - \alpha \frac{1}{m}\sum_{i=1}^{m}(h_\theta(\boldsymbol{x}^{(i)}) - y^{(i)})\, x_j^{(i)}$$

} (simultaneously update for every $j = 0, \ldots, n$)

- **Algorithm looks identical to linear regression (except for the definition of $h(x)$)!!!**

Gradient Descent for Multiple Variables

- Output (probability): $h_\theta(\boldsymbol{x}) = \dfrac{1}{1+e^{-\boldsymbol{\theta}^T \boldsymbol{x}}}$
- Cost function: $J(\boldsymbol{\theta}) = -\dfrac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log h_\theta(\boldsymbol{x}^{(i)}) + (1-y^{(i)})\log(1-h_\theta(\boldsymbol{x}^{(i)}))\right]$
- **Gradient descent:**

Repeat $\{$ $\theta_j := \theta_j - \alpha \dfrac{\partial}{\partial \theta_j} J(\theta_0, \ldots, \theta_n)$

$$= \theta_j - \alpha \frac{1}{m}\sum_{i=1}^{m}(h_\theta(\boldsymbol{x}^{(i)}) - y^{(i)})\, x_j^{(i)}$$

$\}$ (simultaneously update for every $j = 0, \ldots, n$)

- **Algorithm looks identical to linear regression (except for the definition of $h(x)$)!!!**

Thank You!

Questions?