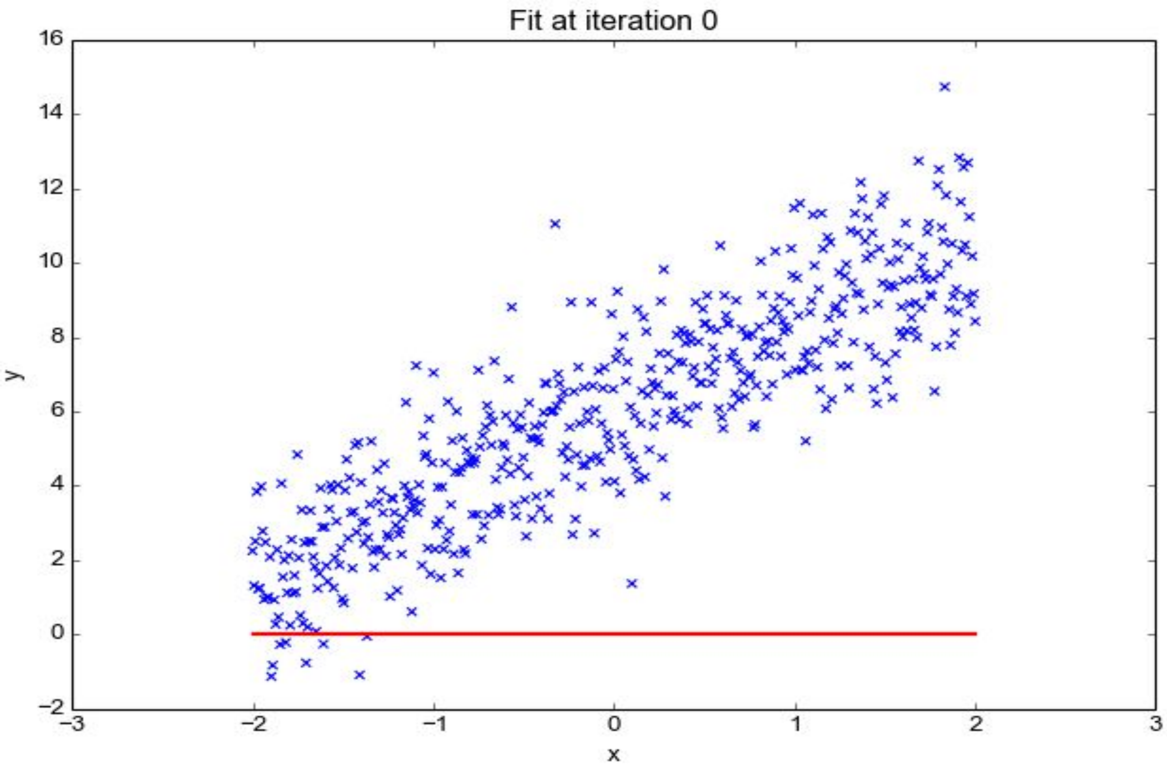
Classification and logistic regression

Aleksanyan Lida

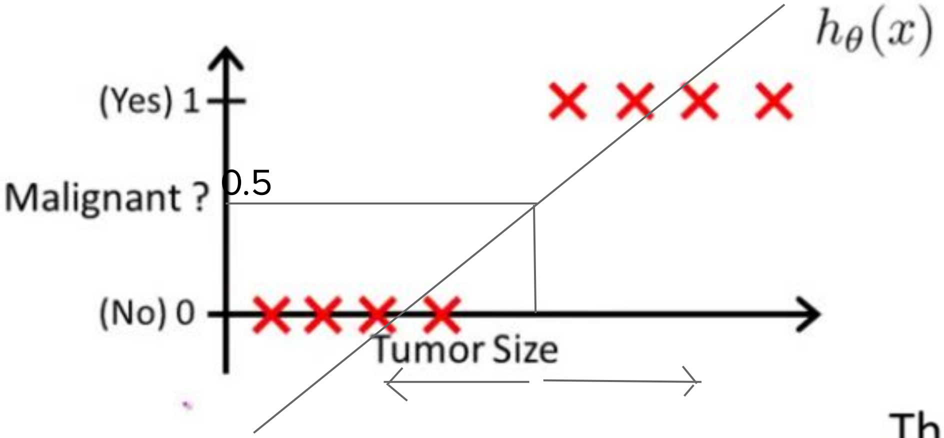
Linear Regression model



Classification

	Two Class Classification	
$y \in \{0, 1\}$	**1 or Positive Class**	**0 or Negative Class**
Email	Spam	Not Spam
Tumor	Malignant	Benign
Transaction	Fraudulent	Not Fraudulent

Example



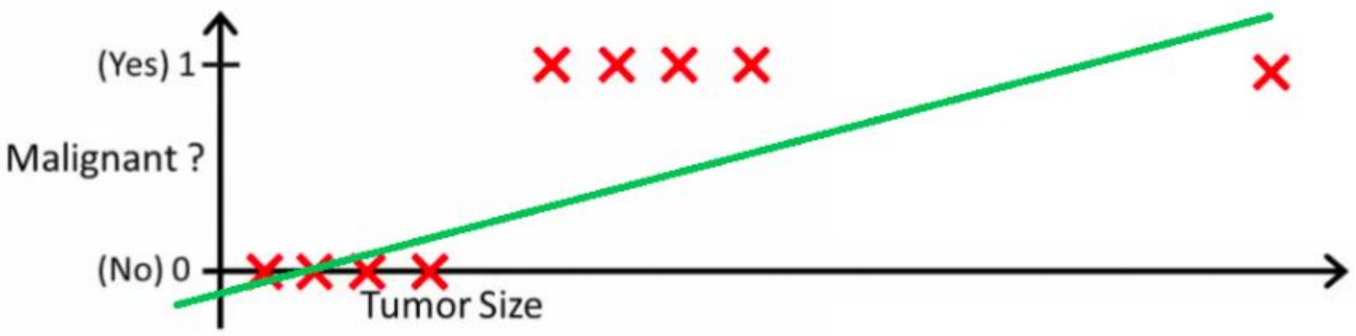
We could approach the classification problem ignoring the fact that y is discrete-valued, and use our old linear regression algorithm to try to predict y given x.

Threshold classifier output $h_\theta(x)$ at 0.5:

If $h_\theta(x) \geq 0.5$, predict "y = 1"

If $h_\theta(x) < 0.5$, predict "y = 0"

Example



Threshold classifier output $h_\theta(x)$ at 0.5:

If $h_\theta(x) \geq 0.5$, predict "y = 1"

If $h_\theta(x) < 0.5$, predict "y = 0"

Classification: y = 0 or 1

$h_\theta(x)$ can be > 1 or < 0

Logistic Regression Model

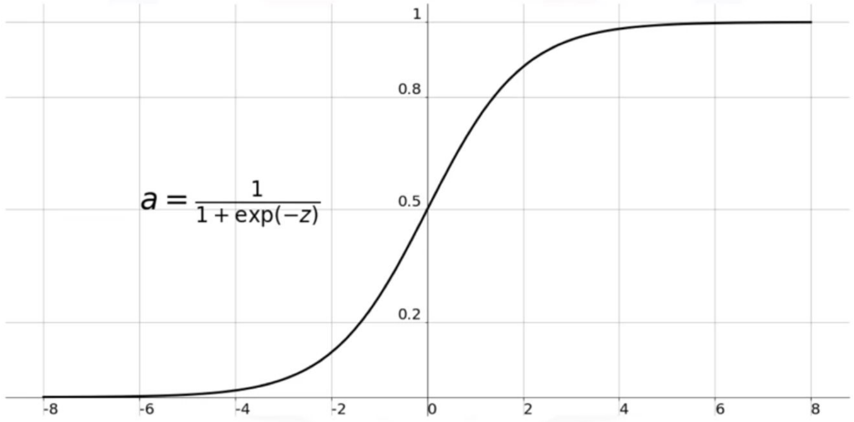
Want $0 \le h_\theta(x) \le 1$

Sigmoid Function

We are going to use this function:

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

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



Good property of sigmoid function:

$$\begin{aligned} g'(z) &= \frac{d}{dz}\frac{1}{1+e^{-z}} \\ &= \frac{1}{(1+e^{-z})^2}\left(e^{-z}\right) \\ &= \frac{1}{(1+e^{-z})}\cdot\left(1-\frac{1}{(1+e^{-z})}\right) \\ &= g(z)(1-g(z)). \end{aligned}$$

Let's assume that:

$$\begin{aligned} P(y=1 \mid x;\theta) &= h_\theta(x) \\ P(y=0 \mid x;\theta) &= 1-h_\theta(x) \end{aligned}$$

We can write it this way too:

$$p(y \mid x;\theta) = (h_\theta(x))^y\,(1-h_\theta(x))^{1-y}$$

So this function gives us the sense of probability

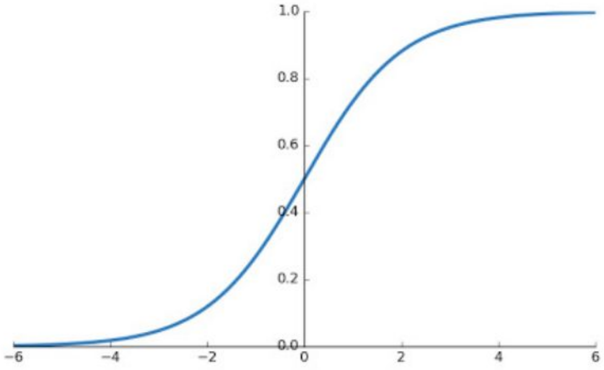
$$h_\theta(x) = P(y=1|x;\theta) = 1-P(y=0|x;\theta)$$

$$P(y=0|x;\theta)+P(y=1|x;\theta) = 1$$

Logistic regression

$$h_\theta(x) = g(\theta^T x) = P(y=1|x;\theta)$$

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



Suppose predict "$y=1$" if $h_\theta(x) \geq 0.5$

$$\theta^T x \geq 0$$

predict "$y=0$" if $h_\theta(x) < 0.5$

$$\theta^T x < 0$$

g(z) $\geq$ 0.5, when z $\geq$ 0

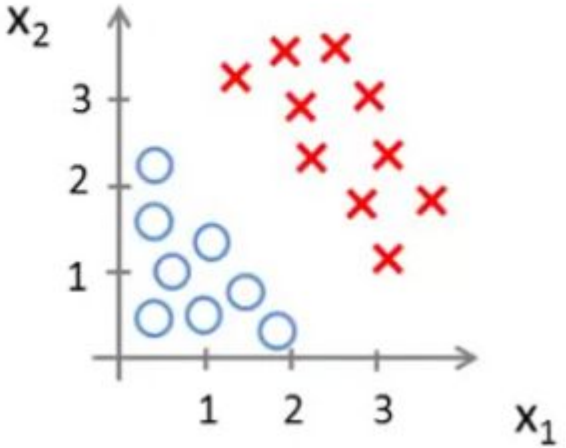
$$h_\theta(x) = g(\theta^T x) \geq 0.5$$

whenever $\theta^T x \geq 0$

$$h_\theta(x) = g(\theta^T x) < 0.5$$

$$\theta^T x < 0$$

Decision Boundary

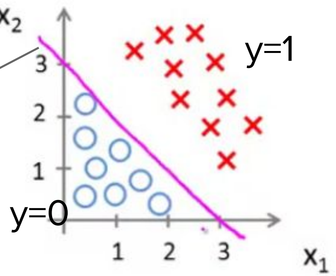


$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$

Now suppose we found somehow parametres
theta1 = -3, theta2 = 1 and theta3 = 1
We know that:

Predict "$y = 1$" if $-3 + x_1 + x_2 \geq 0$

or we can write it as: $x_1 + x_2 \geq 3$

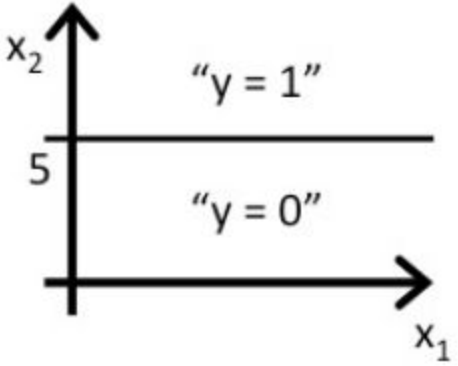


Decision boundary

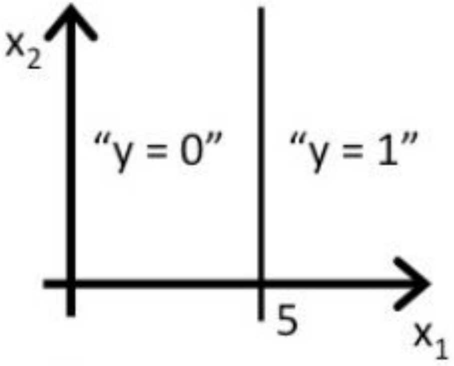
$h_\theta(x) =$ 0.5
when $x_1 + x_2$ = 3

Consider logistic regression with two features x_1 and x_2. Suppose $\theta_0 = 5, \theta_1 = -1, \theta_2 = 0$, so that $h_\theta(x) = g(5 - x_1)$. Which of these shows the decision boundary of $h_\theta(x)$?

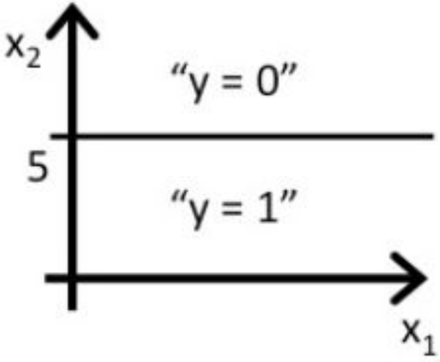
1.



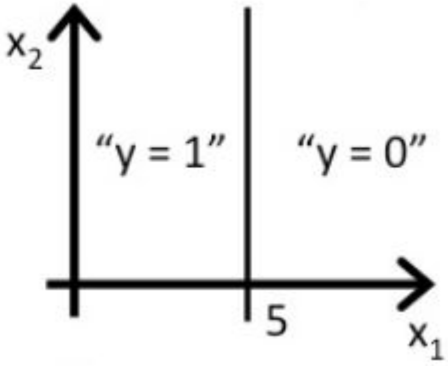
2.



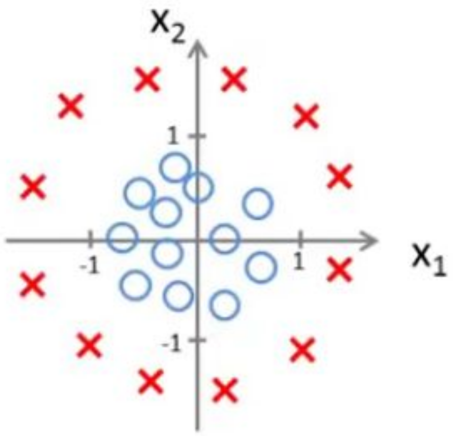
3.



4.



Non-linear decision boundaries

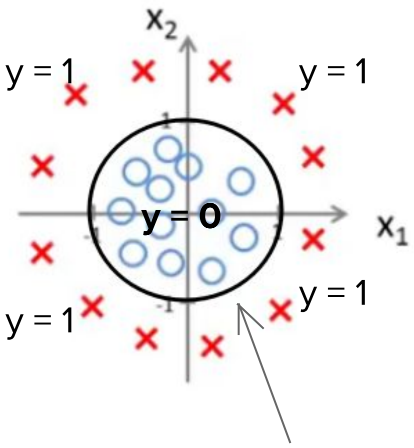


$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2)$$

Suppose we get such values:
theta0 = -1, theta1 = 0, theta2 = 0, theta3 = 1, theta4 = 1

Predict "$y = 1$" if $-1 + x_1^2 + x_2^2 \geq 0$

$$x_1^2 + x_2^2 \geq 1$$



$x_1^2 + x_2^2 = 1$

Decision boundary

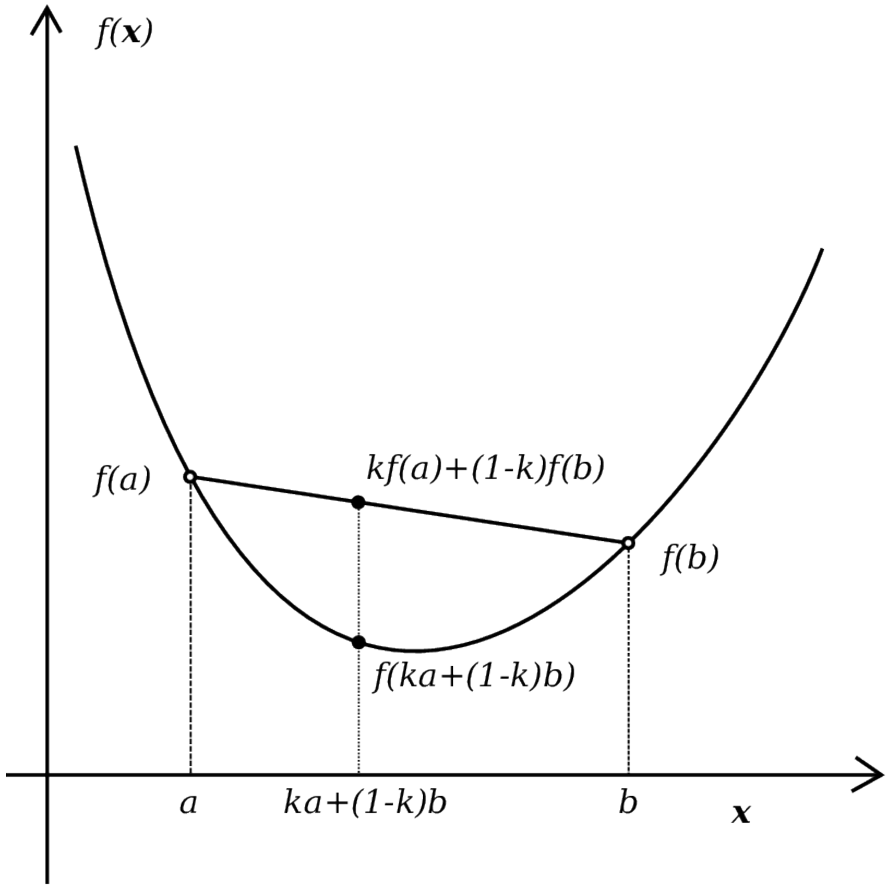
Training set: $\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \cdots, (x^{(m)}, y^{(m)})\}$

m examples $\quad x \in \begin{bmatrix} x_0 \\ x_1 \\ \cdots \\ x_n \end{bmatrix} \quad x_0 = 1, y \in \{0, 1\}$

$$h_\theta(x) = \frac{1}{1 + e^{-\theta^T x}}$$ this is hypothesis

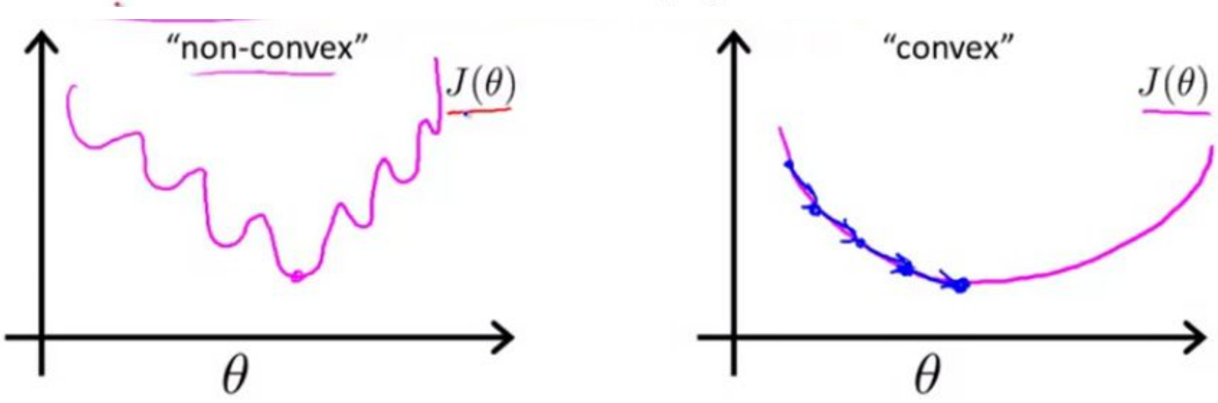
How to choose parameters θ ?

Convex Function



Cost function

Linear regression: $J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\frac{1}{2}\left(h_\theta(x^{(i)}) - y^{(i)}\right)^2$



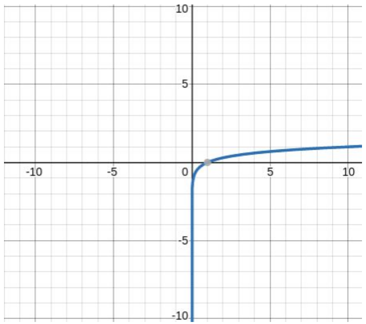
Our sigmoid function isn't convex, so we must change cost function

So we are going to use the following cost function

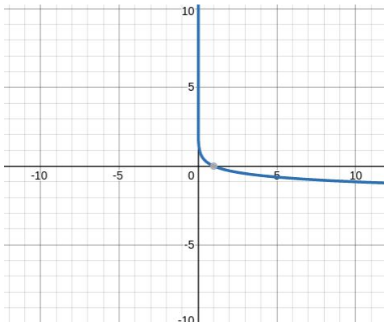
Logistic regression cost function

$$\mathrm{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$

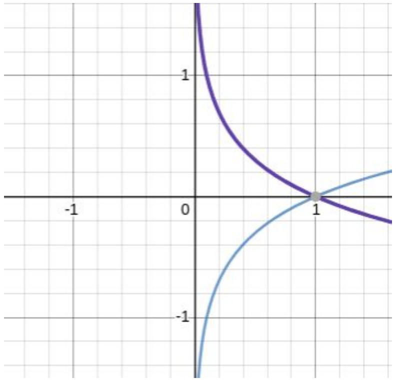
log(x) function



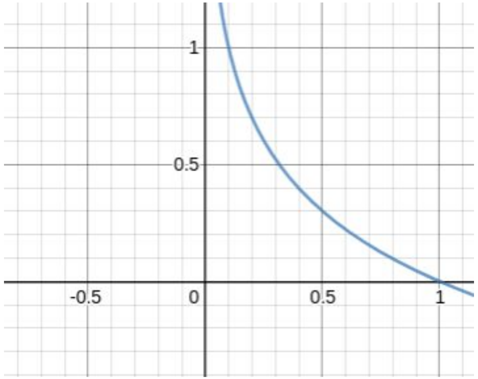
-log(x) function



We're interested only in the range of when this function goes between zero and one



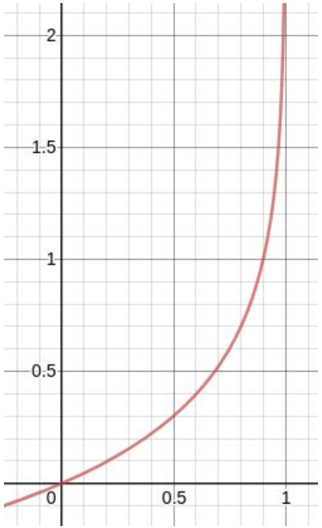
So this is the part that we are interested in



Cost $= 0$ if $y = 1, h_\theta(x) = 1$
But as $h_\theta(x) \to 0$
$Cost \to \infty$

Captures intuition that if $h_\theta(x) = 0$,
(predict $P(y = 1|x;\theta) = 0$), but $y = 1$,
we'll penalize learning algorithm by a very
large cost.

$-\log(1 - h_\theta(x))$



$\text{Cost}(h_\theta(x), y) \to \infty$ if $y = 0$ and $h_\theta(x) \to 1$

Logistic regression cost function

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \mathrm{Cost}(h_\theta(x^{(i)}), y^{(i)})$$

$$\mathrm{Cost}(h_\theta(x), y) = \begin{cases} -\log(h_\theta(x)) & \text{if } y = 1 \\ -\log(1 - h_\theta(x)) & \text{if } y = 0 \end{cases}$$

Note: $y = 0$ or 1 always

Logistic regression cost function

$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} \mathrm{Cost}(h_\theta(x^{(i)}), y^{(i)})$$

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

To fit parameters θ:

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

To make a prediction given new x:

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

We are going to use gradient descent to minimize the cost function J

Gradient Descent

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

Want $\min_\theta J(\theta)$:

Repeat $\{$

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

(simultaneously update all θ_j)

$\}$

The algorithm looks identical to linear regression, but here we have different hypothesis function (sigmoid)

Multiclass classification

Email foldering/tagging: Work, Friends, Family, Hobby

$y = 1$ $y = 2$ $y = 3$ $y = 4$

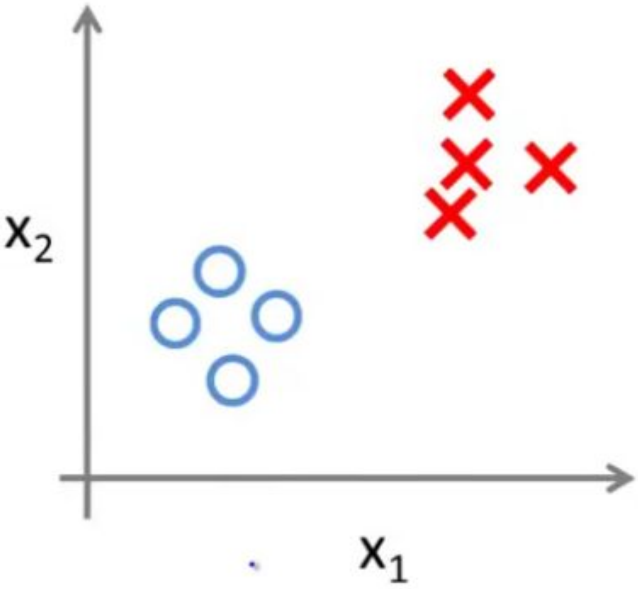
Medical diagrams: Not ill, Cold, Flu

$y = 1$ $y = 2$ $y = 3$

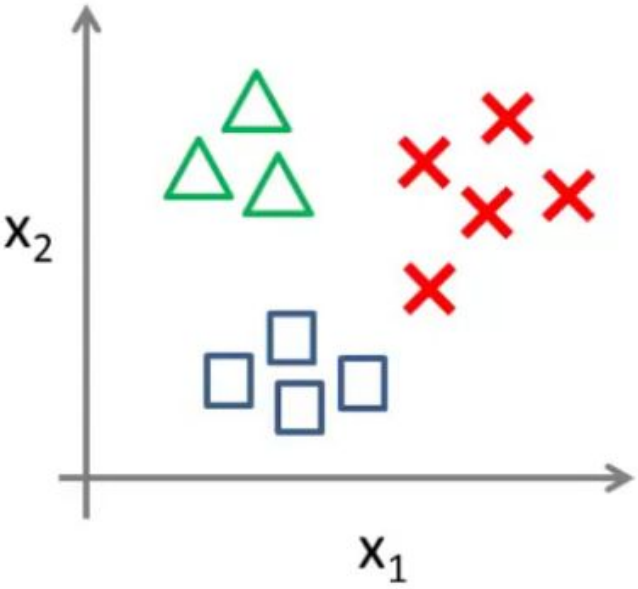
Weather: Sunny, Cloudy, Rain, Snow

$y = 1$ $y = 2$ $y = 3$ $y = 4$

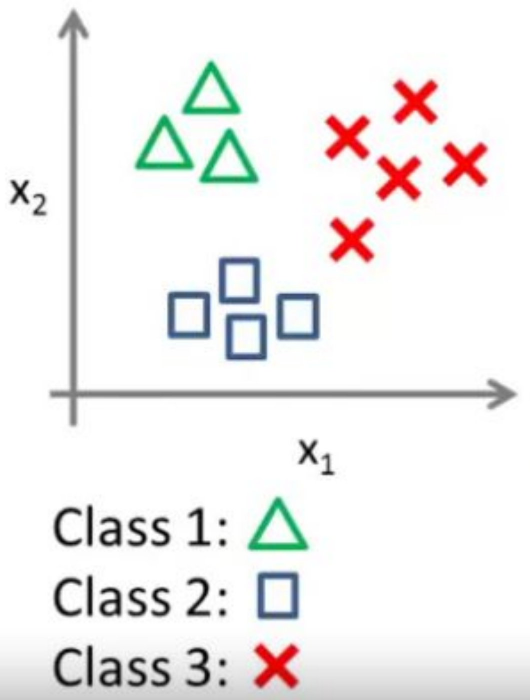
Binary classification:

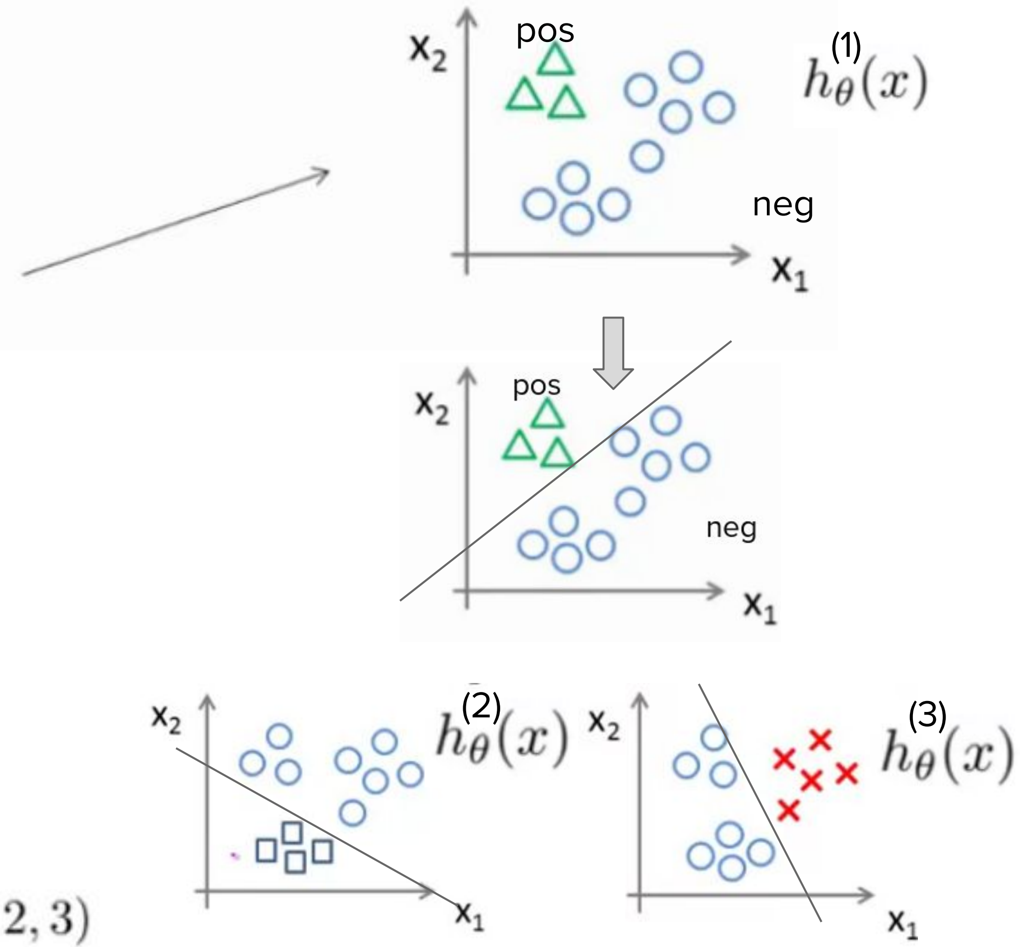


Multi-class classification:



One-vs-all (one-vs-rest):





$$h_\theta^{(i)}(x) = P(y = i|x;\theta) \qquad (i = 1, 2, 3)$$

Summary

One-vs-all

Train a logistic regression classifier $h_\theta^{(i)}(x)$ for each class i to predict the probability that $y = i$.

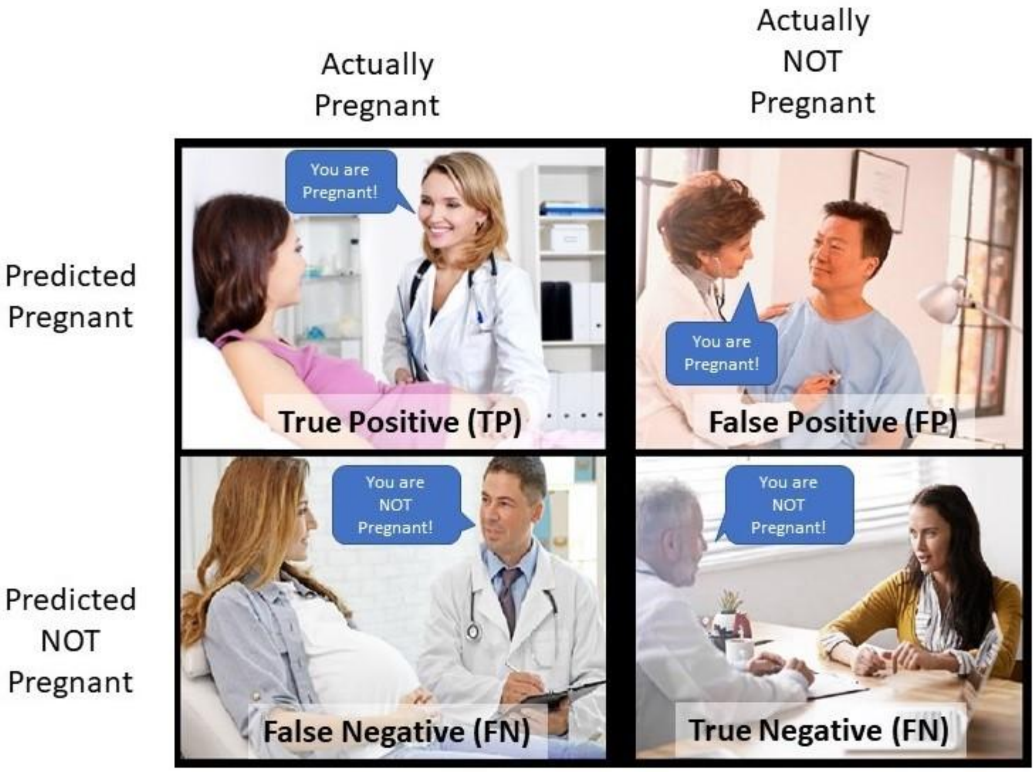
On a new input x, to make a prediction, pick the class i that maximizes

$$\max_i h_\theta^{(i)}(x)$$

Some Metrics

Actual	Predicted		
		Negative	Positive
	Negative	True Negative	False Positive
	Positive	False Negative	True Positive

Some metrics



Confusion Matrix

Confusion Matrix

	Actually Positive (1)	Actually Negative (0)
Predicted Positive (1)	True Positives (TPs)	False Positives (FPs)
Predicted Negative (0)	False Negatives (FNs)	True Negatives (TNs)

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

$$\text{Precision} = \frac{\textit{True Positive}}{\textit{True Positive} + \textit{False Positive}}$$

$$= \frac{\textit{True Positive}}{\textit{Total Predicted Positive}}$$

$$\text{Recall} = \frac{\textit{True Positive}}{\textit{True Positive} + \textit{False Negative}}$$

$$= \frac{\textit{True Positive}}{\textit{Total Actual Positive}}$$

$$\text{F1} = 2 \times \frac{\textit{Precision} * \textit{Recall}}{\textit{Precision} + \textit{Recall}}$$