Maximum Likelihood Estimation: MLE

Maximum Likelihood Estimation: MLE

estimate the parameters of a model

Logistic Regression & Binomial Distribution

- For the sample data, a **binomial distribution** is assumed in case of logistic regression, where **each example is one outcome of a Bernoulli** trial & it has a single parameter: the probability of an event or specific class (P)

  P(Y=1) = P

  P(Y=0) = 1 – P

- The expected value (mean) of the Bernoulli distribution can be calculated as

  Mean = P(Y=1) * 1 + P(Y=0) * 0

  Likelihood = $\hat{Y}$ * Y + (1 - $\hat{y}$) * (1- Y)

- It will return a large probability when the model is close to the matching class value, and a small value when it is far away for both the classes

Maximum Likelihood Estimation: MLE

estimate the parameters of a model

objective: wish to maximize the conditional probability of observing the data (X) given specific probability dist^n & its parameters (theta)

Maximum Likelihood Estimation: MLE

estimate the parameters of a model

objective: wish to maximize the conditional probability of observing the data (X) given specific probability dist^n & its parameters (theta)

P(Y= y/X; theta)

P(X1, x2, ...,Xn; theta)

Maximum Likelihood Estimation: MLE

estimate the parameters of a model

objective: wish to maximize the conditional probability of observing the data (X) given specific probability dist^n & its parameters (theta)

P(Y= y/X; theta)

P(X1, x2, ...,Xn; theta)

L (X; theta)

sum of the log for conditional probability

Maximum Likelihood Estimation: MLE

estimate the parameters of a model

objective: wish to maximize the conditional probability of observing the data (X) given specific probability dist^n & its parameters (theta)

P(Y= y/X; theta)

P(X1, x2, ...,Xn; theta)

L (X; theta)

sum of the log for conditional probability

summation i = 1 to n log (P(Xi; theta)

Maximum Likelihood Estimation: MLE

estimate the parameters of a model

objective: wish to maximize the conditional probability of observing the data (X) given specific probability dist^n & its parameters (theta)

P(Y= y/X; theta)

P(X1, x2, ...,Xn; theta)

L (X; theta)

sum of the log for conditional probability

summation i = 1 to n log (P(Xi; theta)

Maximum Likelihood Estimation: MLE

estimate the parameters of a model

objective: wish to maximize the conditional probability of observing the data (X) given specific probability dist^n & its parameters (theta)

P(Y= y/X; theta)

P(X1, x2, ...,Xn; theta)

L (X; theta)

sum of the log for conditional probability

summation i = 1 to n log (P(Xi; theta)

this function to return a large probability when the model is close to the matching class, & a small value when it is far away from the class

Maximum Likelihood Estimation: MLE

estimate the parameters of a model

objective: wish to maximize the conditional probability of observing the data (X) given specific probability dist^n & its parameters (theta)

P(Y= y/X; theta)

-summation i = 1 to n log (P(Xi;theta))

P(X1, x2, ...,Xn; theta)

L (X; theta)

sum of the log for conditional probability

summation i = 1 to n log (P(Xi; theta)

this function to return a large probability when the model is close to the matching class, & a small value when it is far away from the class

Expected mean value of bernoulli dist^n
= P(Y=1) * 1 + P(Y=0) * 0

Expected mean value of bernoulli dist^n
= P(Y=1) * 1 + P(Y=0) * 0

= p * 1 + (1-p) * 0
= y hat * y + (1- yhat) * (1-y)

Expected mean value of bernoulli dist^n
= P(Y=1) * 1 + P(Y=0) * 0

= p * 1 + (1-p) * 0
= y hat * y + (1- yhat) * (1-y)

- sum i = 1 to n (log(yhat) yi + log(1- yhat) (1-yi))

Expected mean value of bernoulli dist^n
= P(Y=1) * 1 + P(Y=0) * 0

= p * 1 + (1-p) * 0
= y hat * y + (1- yhat) * (1-y)

- sum i = 1 to n (log(yhat) yi + log(1- yhat) (1-yi))

= - y log(h(x) - (1-y) log(1-h(x)) = cost ((h(x), y)

Expected mean value of bernoulli dist^n
= P(Y=1) * 1 + P(Y=0) * 0

= p * 1 + (1-p) * 0
= y hat * y + (1- yhat) * (1-y)

- sum i = 1 to n (log(yhat) yi + log(1- yhat) (1-yi))

= - y log(h(x) - (1-y) log(1-h(x)) = cost ((h(x), y)

theta j := theta j - alpha (summation i =1 to m (h(x^i) - y^i) Xj^i

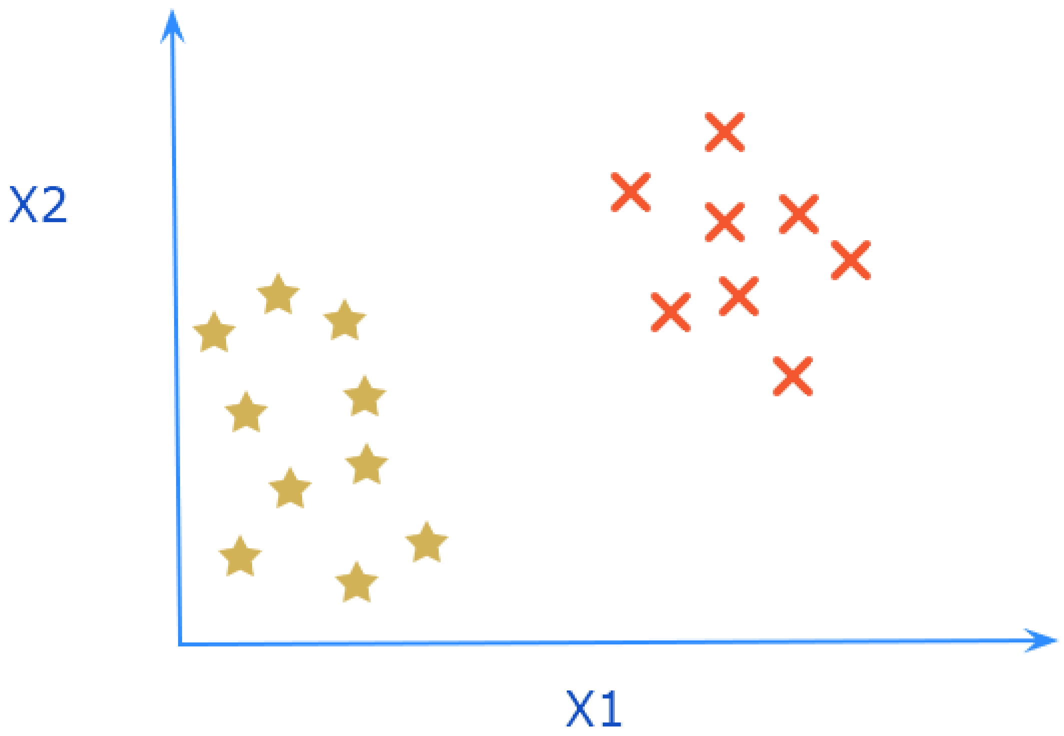
Expected mean value of bernoulli dist^n
= P(Y=1) * 1 + P(Y=0) * 0

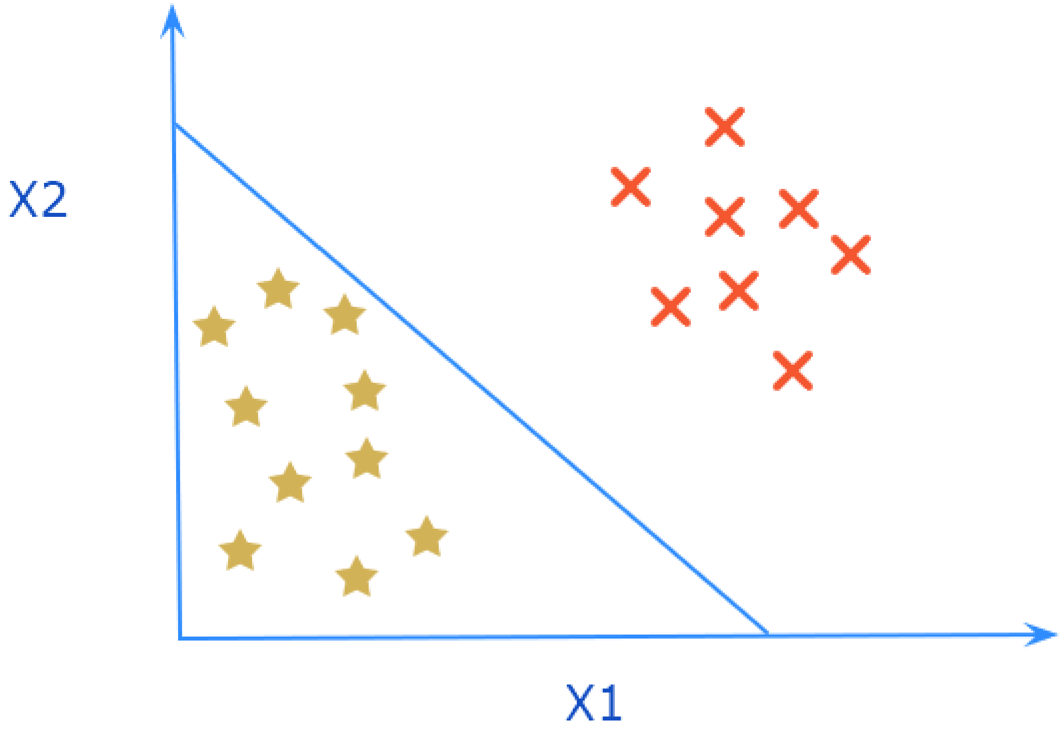
= p * 1 + (1-p) * 0
= y hat * y + (1- yhat) * (1-y)

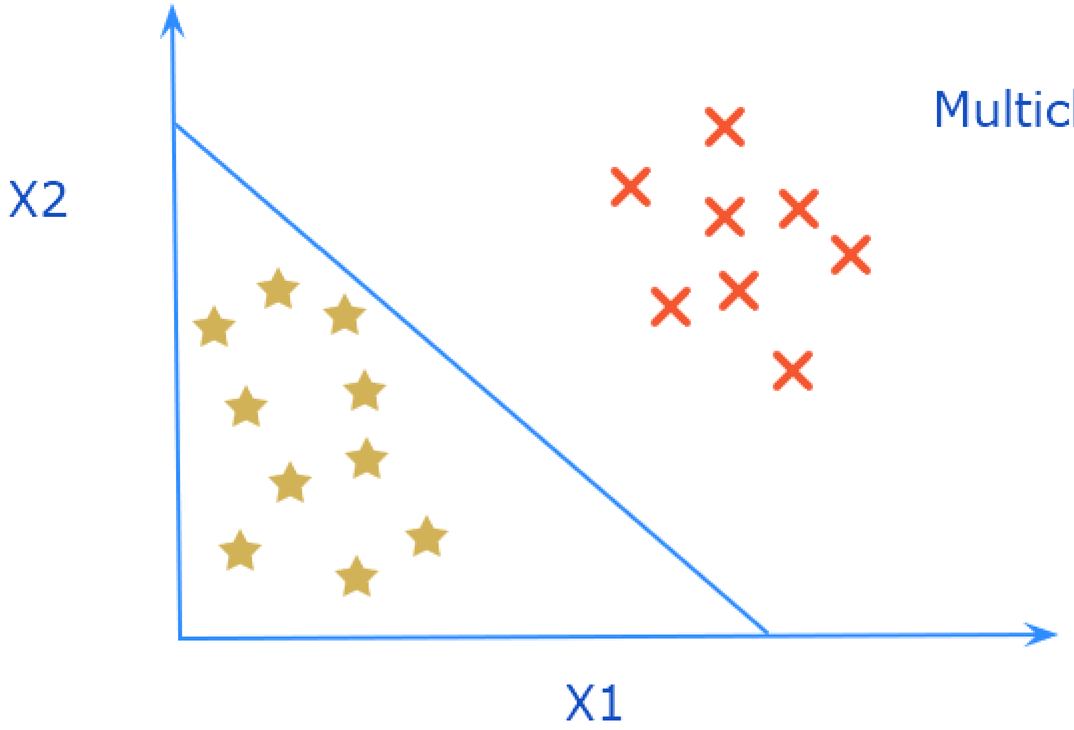
- sum i = 1 to n (log(yhat) yi + log(1- yhat) (1-yi))

= - y log(h(x) - (1-y) log(1-h(x)) = cost ((h(x), y)

theta j := theta j - alpha (summation i =1 to m (h(x^i) - y^i) Xj^i


X2
X1

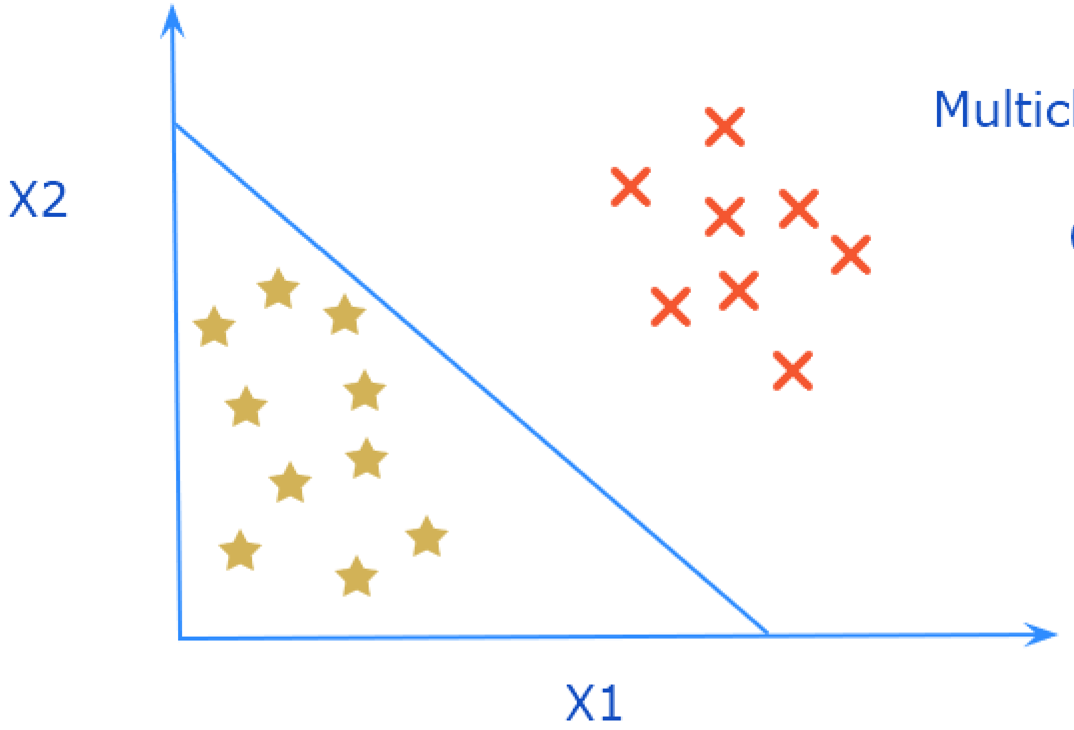

X2
X1



Multiclass classification: y (1, 2, 3,4, 5)

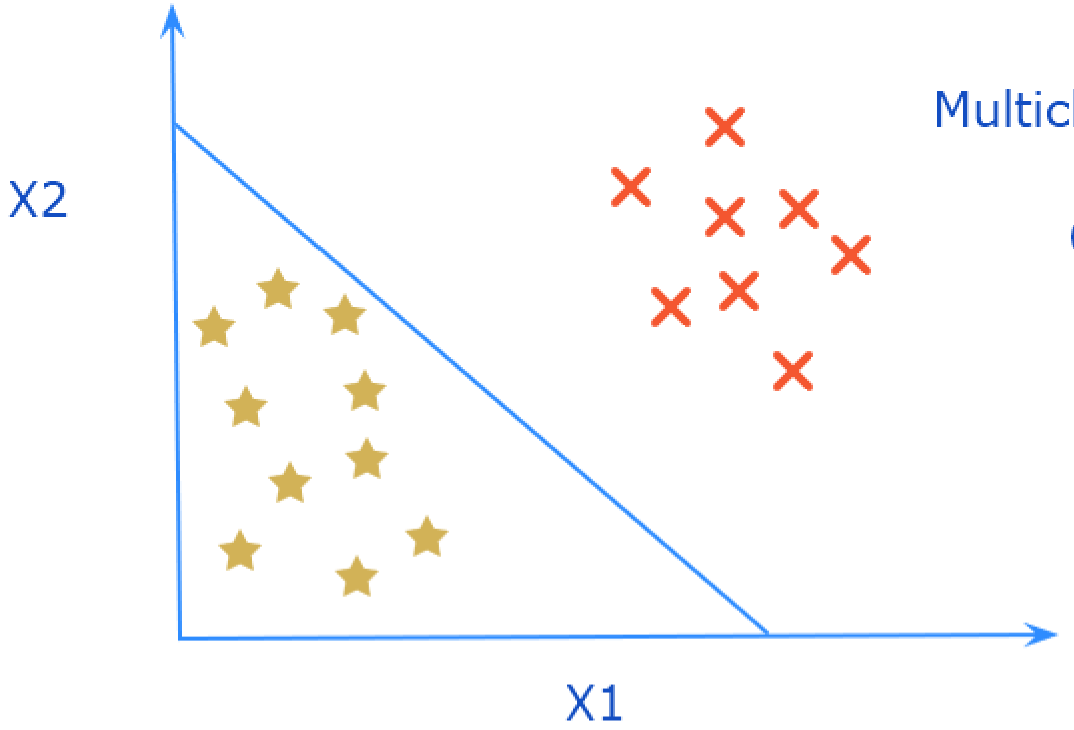
Multiclass classification: y (1, 2, 3,4, 5)

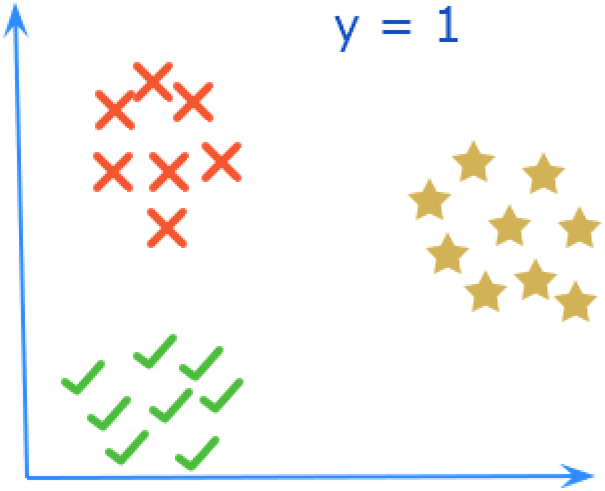
One Vs all (One Vs rest)

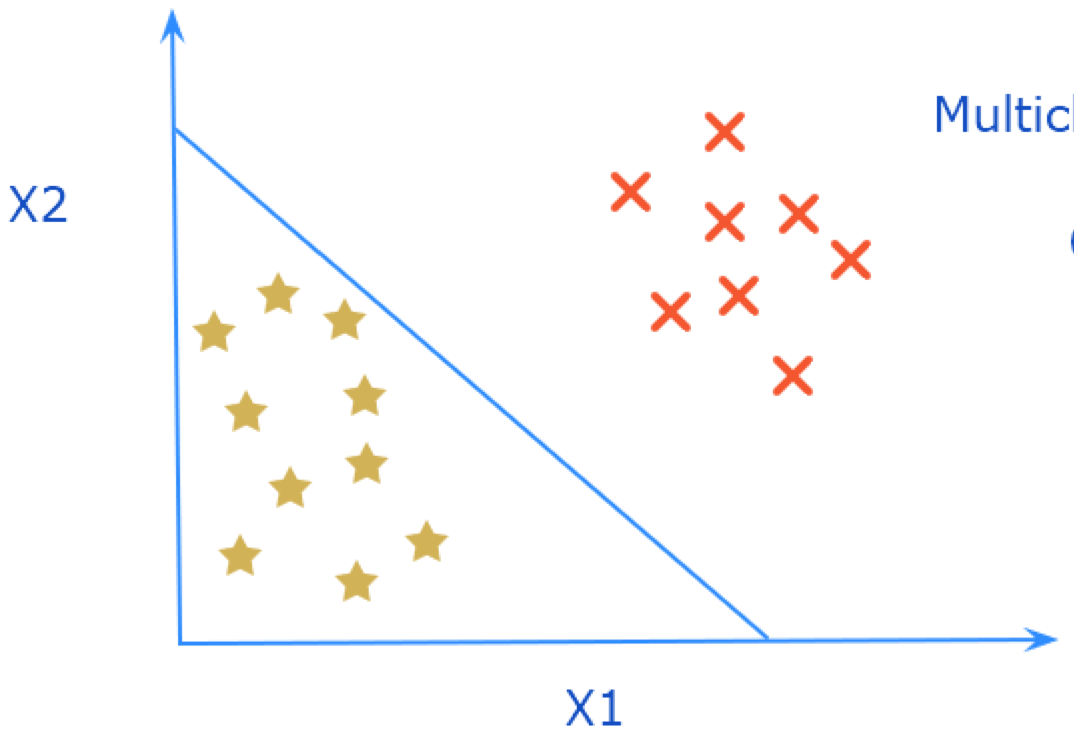


Multiclass classification: y (1, 2, 3,4, 5)

One Vs all (One Vs rest)

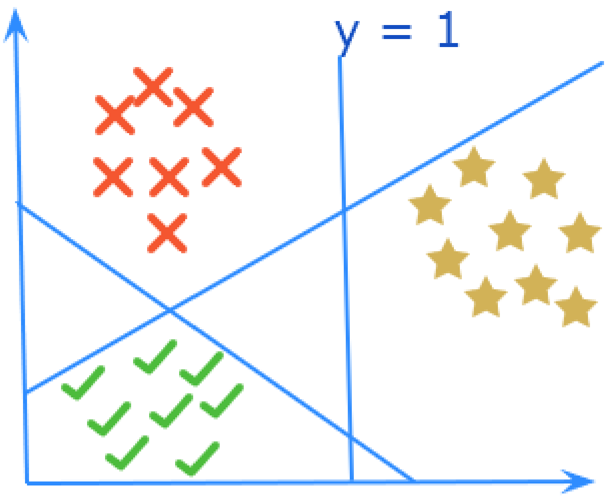






Multiclass classification: y (1, 2, 3,4, 5)

One Vs all (One Vs rest)



Example

- h (x) = g (theta0 + theta1x1 + theta2x2) and the values of theta0 = -3, theta1 = 1 and theta2 = 1; how would you define decision boundary in this case.

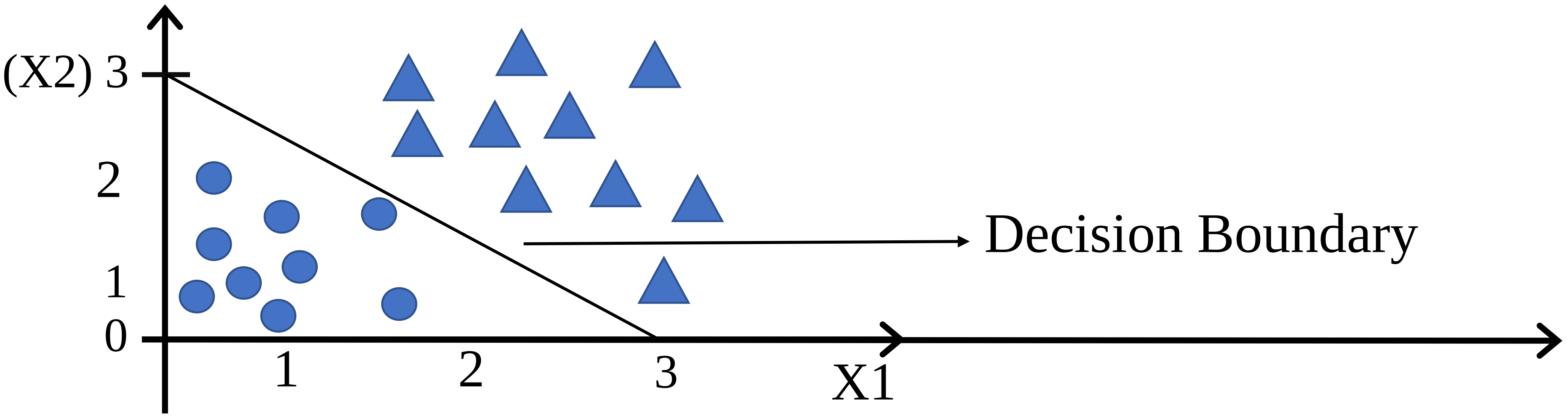
$$h_\theta(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$

- Predict “y = 1” if g(z) ≥ 0.5 and this will happen when “z ≥ 0”

$$-3+x1+x2 \geq 0;\ x1+x2 \geq 3$$

- Predict “y = 0” if g(z) < 0.5 and this will happen when “z < 0”

$$-3+x1+x2 < 0;\ x1+x2 < 3$$



- **Decision boundary** is the property of the hypothesis function; i.e. parameters define the boundary not the training set however, training set is used to find the value of parameters.

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

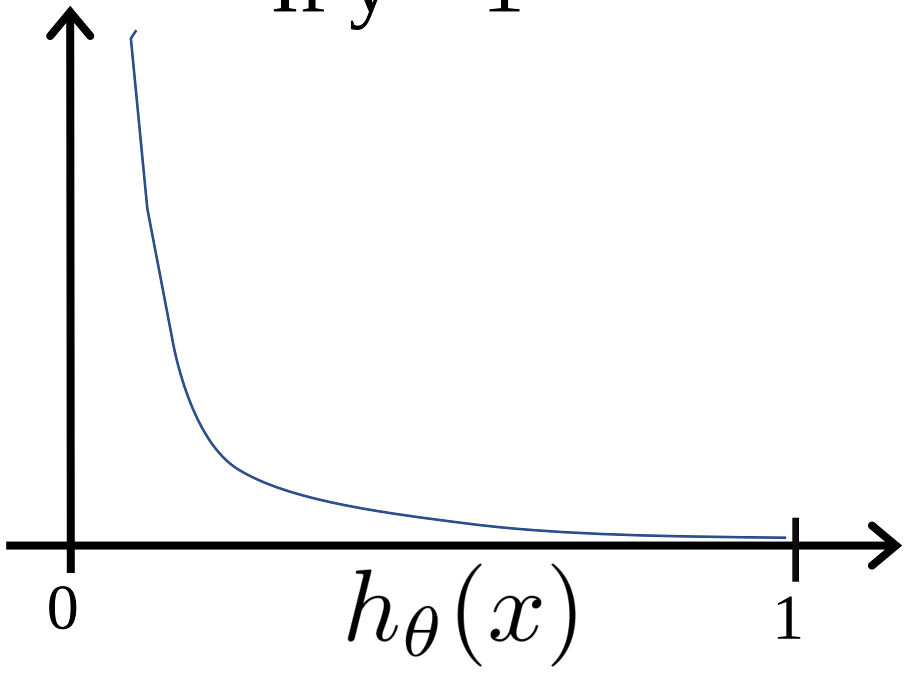
If $h_\theta(x) \geq 0.5$, predict "y = 1"; when x ≥ 0

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

Logistic regression cost function

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

If y = 1



$\mathrm{Cost} = 0$ if $y = 1, h_\theta(x) = 1$

But as $h_\theta(x) \rightarrow 0$

$Cost \rightarrow \infty$

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.

Logistic regression cost function

$x^{(i)}$ = input (features) of i^{th} training example.

$x_j^{(i)}$ = value of feature j in i^{th} training example

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

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

To fit parameters θ:

$$\min_\theta J(\theta)$$

To make a prediction given new x:

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

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

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 $\{$

$$\theta_j := \theta_j - \alpha \frac{\partial}{\partial \theta_j} J(\theta)$$

(simultaneously update all θ_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 $\{$

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

(simultaneously update all θ_j)

$\}$

Algorithm looks identical to linear regression!

Multi-class Classification (One-Vs-All)

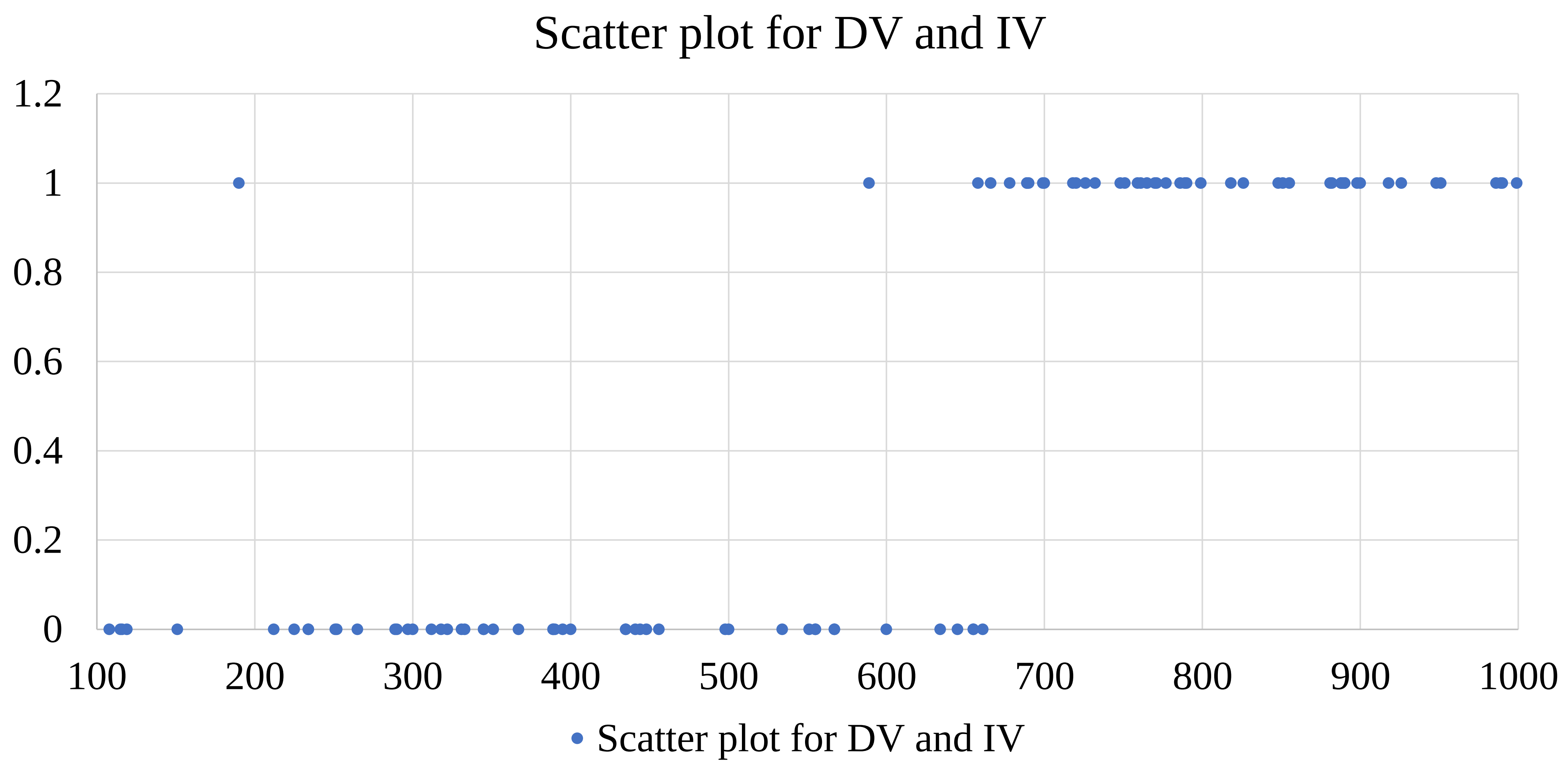
Multiclass classification

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

Medical diagrams: Not ill, Cold, Flu

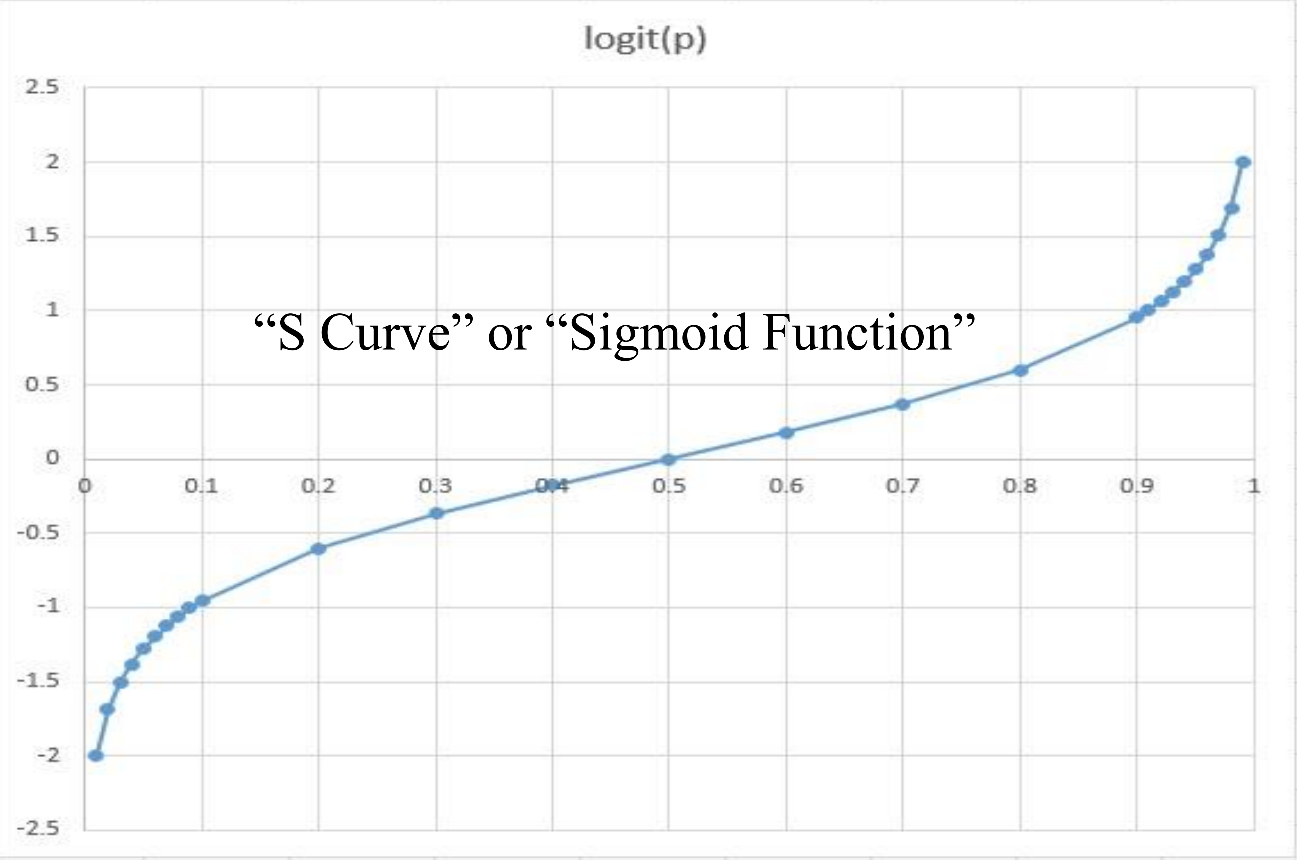
Weather: Sunny, Cloudy, Rain, Snow

Scatter Plot



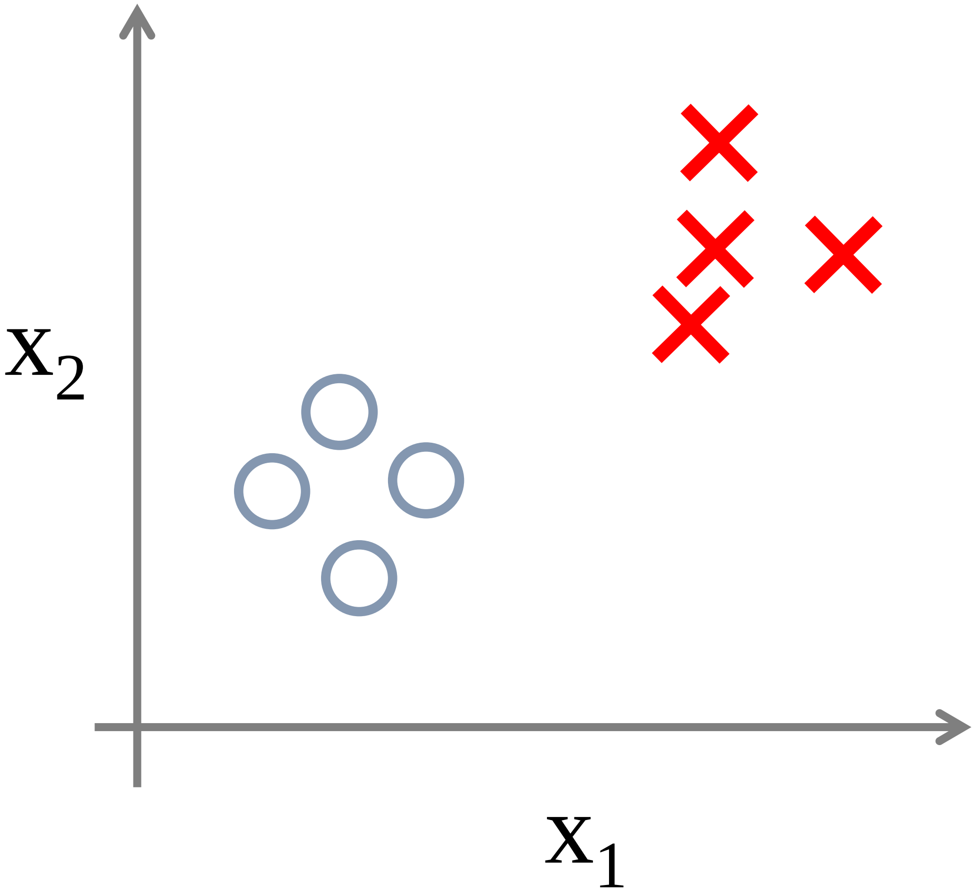
Logit Function Graph

p	logit(p)
0.01	-1.99564
0.02	-1.6902
0.03	-1.50965
0.04	-1.38021
0.05	-1.27875
0.06	-1.19498
0.07	-1.12338
0.08	-1.0607
0.09	-1.0048
0.1	-0.95424
0.2	-0.60206
0.3	-0.36798
0.4	-0.17609
0.5	0
0.6	0.176091
0.7	0.367977
0.8	0.60206
0.9	0.954243
0.91	1.004799
0.92	1.060698
0.93	1.123385
0.94	1.194977
0.95	1.278754
0.96	1.380211
0.97	1.50965
0.98	1.690196
0.99	1.995635

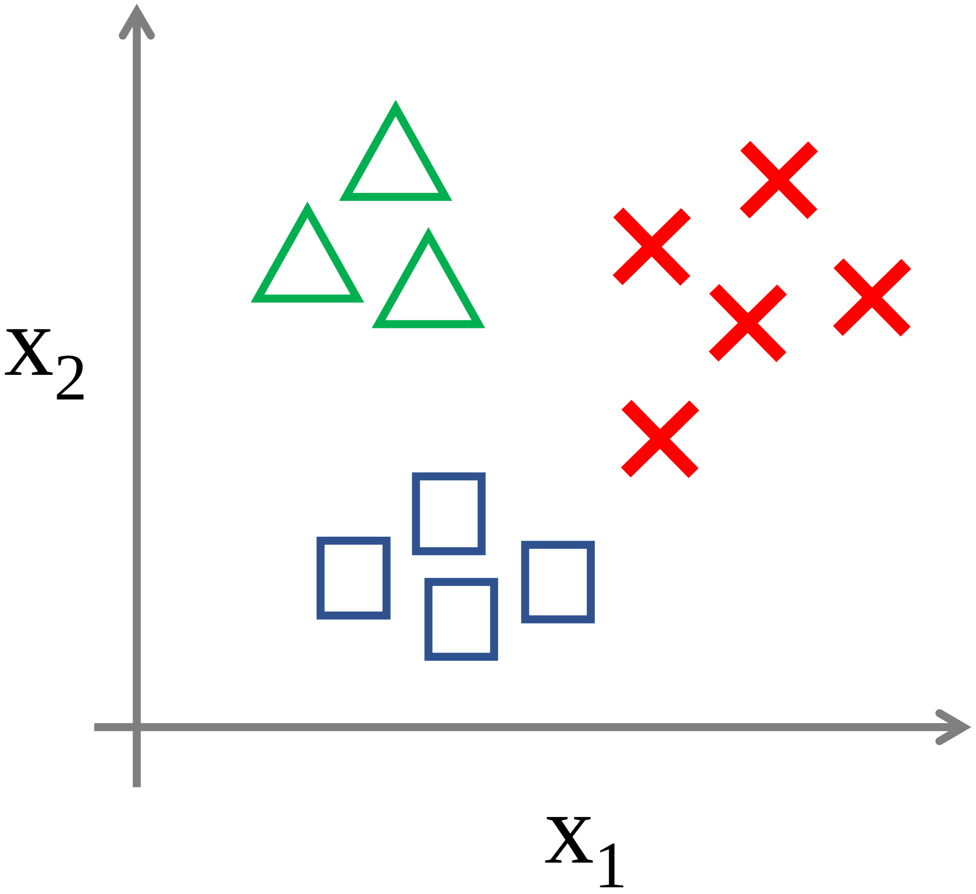


“S Curve” or “Sigmoid Function”

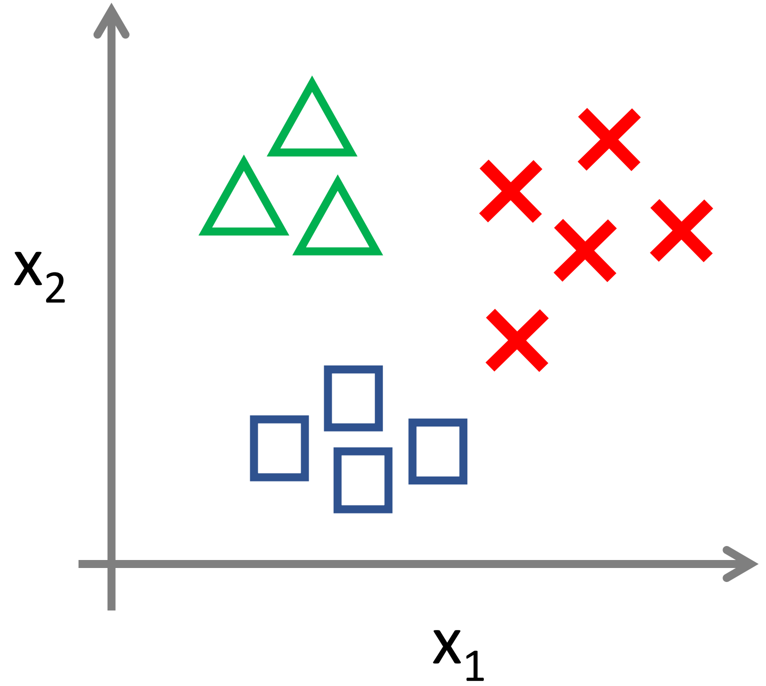
Binary classification:

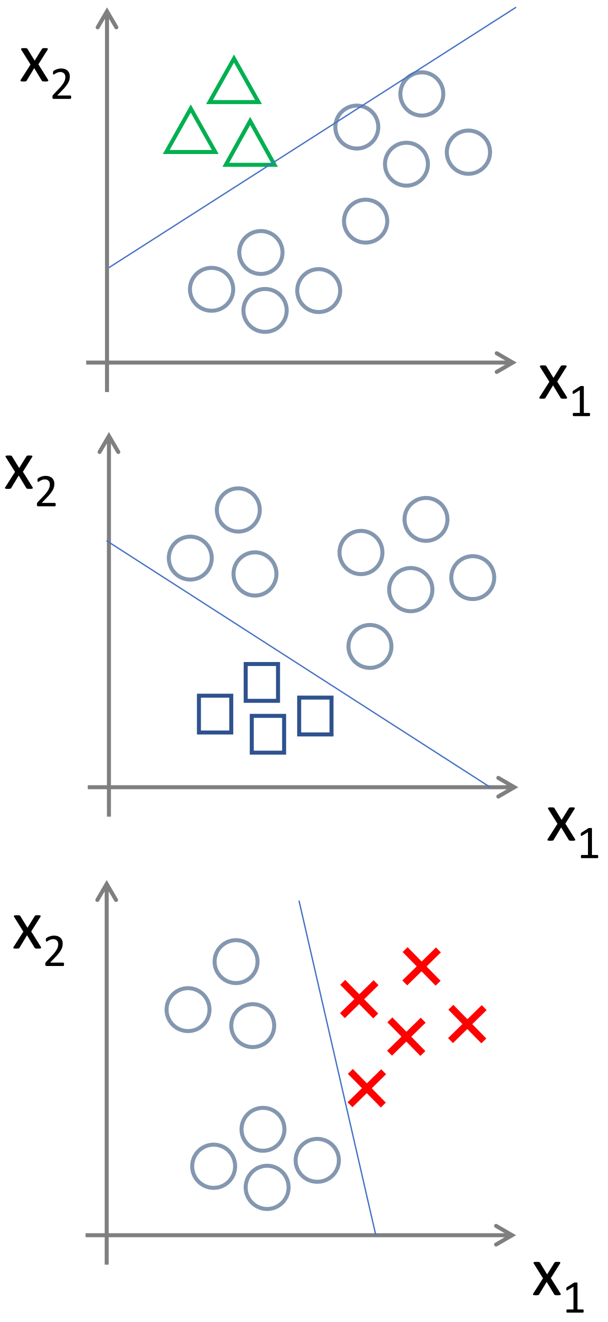


Multi-class classification:



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





Class 1:

Class 2:

Class 3:

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

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

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