

# Chapter 2: Classification

# Supervised Learning: Logistic Regression

# Logistic regression: Introduction

- In classification, we seek to identify the categorical class  $C_k$  associate with a given input vector  $x$ .

Vehicle features / budget: Buy / Not ?

Online Transactions: Fraudulent (Yes / No)?

$$y \in \{0, 1\}$$

0: “Negative Class”

1: “Positive Class”

- In order to predict correct value of  $Y$  for a given value of  $X$ .
  1. Data (samples, combination of  $X$  and  $Y$ )
  2. Model (function to represent relationship  $X$  &  $Y$ )
  3. Cost function (how well our model approximates training samples)
  4. Optimization (find parameters of model to minimize cost function)

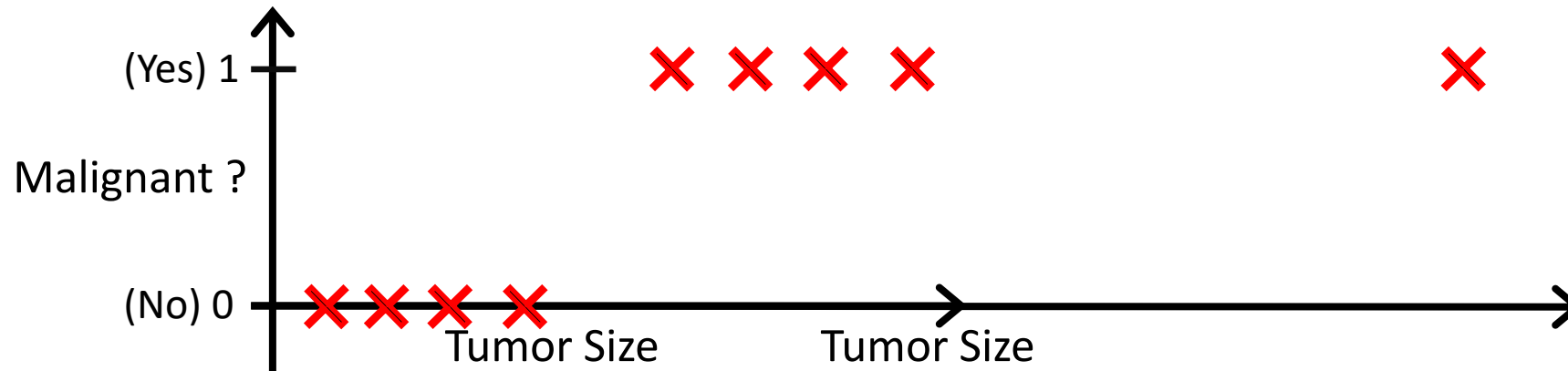
# Logistic regression- data

- In univariate logistic regression the number of independent variables is one and there is a linear relationship between the independent(x) and dependent(y) variable.

Marks scored in entrance examination	Admitted / Not admitted to University
20	Not Admitted
60	Admitted
36	Admitted
32	Not Admitted
30	Not Admitted
80	Admitted
38	Admitted

# Logistic regression: Hypothesis

- Hypothesis used in Linear Regression predicts the continuous values and Logistic regression hypothesis should predict discrete values.



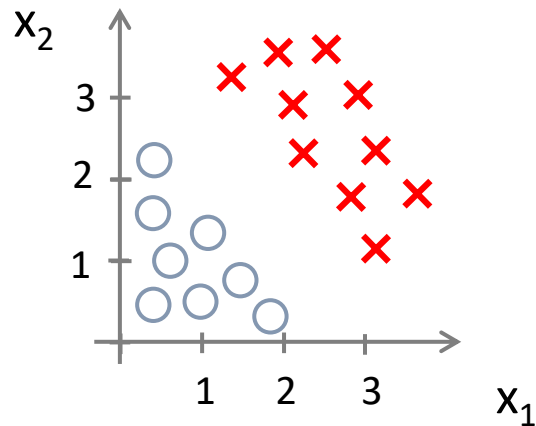
Threshold classifier output  $h_{\theta}(x)$  at 0.5:

If  $h_{\theta}(x) \geq 0.5$ , predict “ $y = 1$ ”

If  $h_{\theta}(x) < 0.5$ , predict “ $y = 0$ ”

# Logistic regression – decision boundary

## Decision Boundary

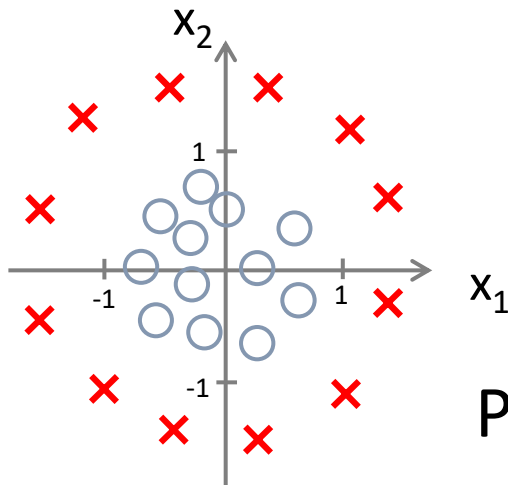


$$h_{\theta}(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$

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

# Logistic regression – decision boundary

## Non-linear decision boundaries



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

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

# Logistic regression - hypothesis

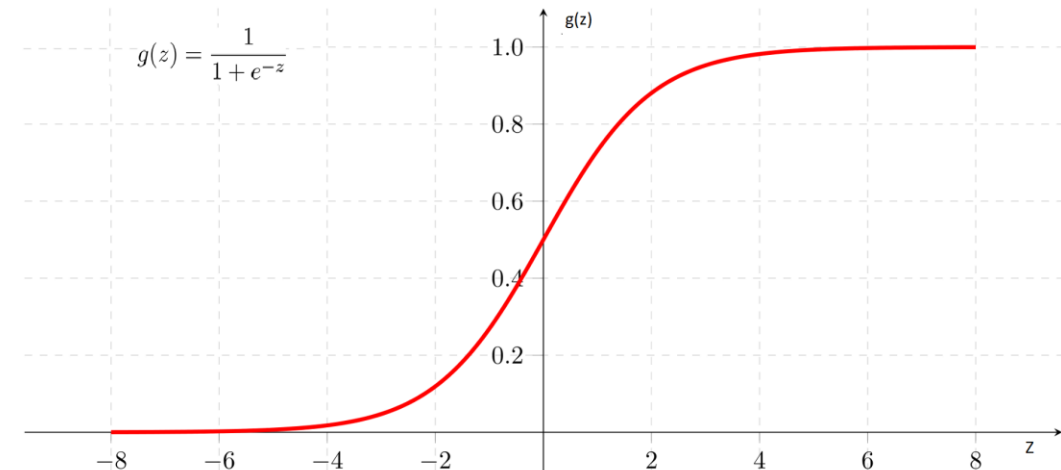
- The sigmoid function is also called a squashing function as its domain is the set of all real numbers, and its range is (0, 1).

Need  $0 \leq h_{\theta}(x) \leq 1$

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

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

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



- For given input, hypothesis always predicts value which is between 0 & 1.

if  $h_{\theta}(x) < 0.5$  then consider  $h_{\theta}(x) = 0$

else if  $h_{\theta}(x) \geq 0.5$  then consider  $h_{\theta}(x) = 1$



# Logistic regression – hypothesis

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

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

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

How to choose parameters  $\theta$  ?

# Logistic regression - hypothesis

## Interpretation of Hypothesis Output

$h_{\theta}(x)$  = estimated probability that  $y = 1$  on input  $x$

Example: If  $x = \begin{bmatrix} x_0 \\ x_1 \end{bmatrix} = \begin{bmatrix} 1 \\ \text{tumorSize} \end{bmatrix}$

$$h_{\theta}(x) = 0.7$$

Tell patient that 70% chance of tumor being malignant

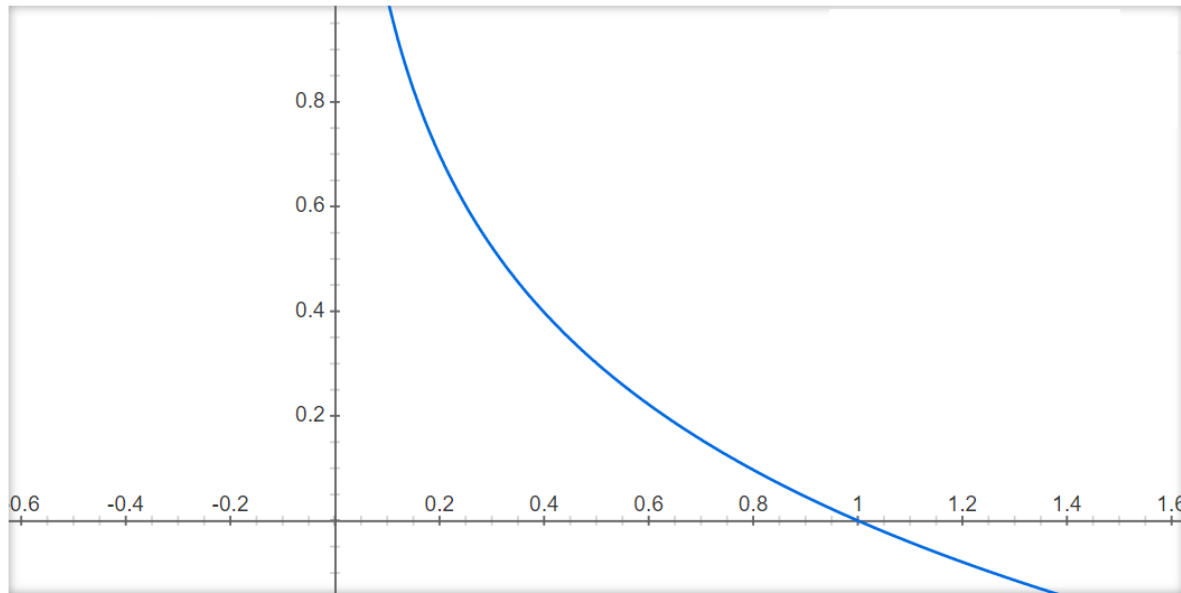
“probability that  $y = 1$ , given  $x$ , parameterized by  $\theta$ ”

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

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

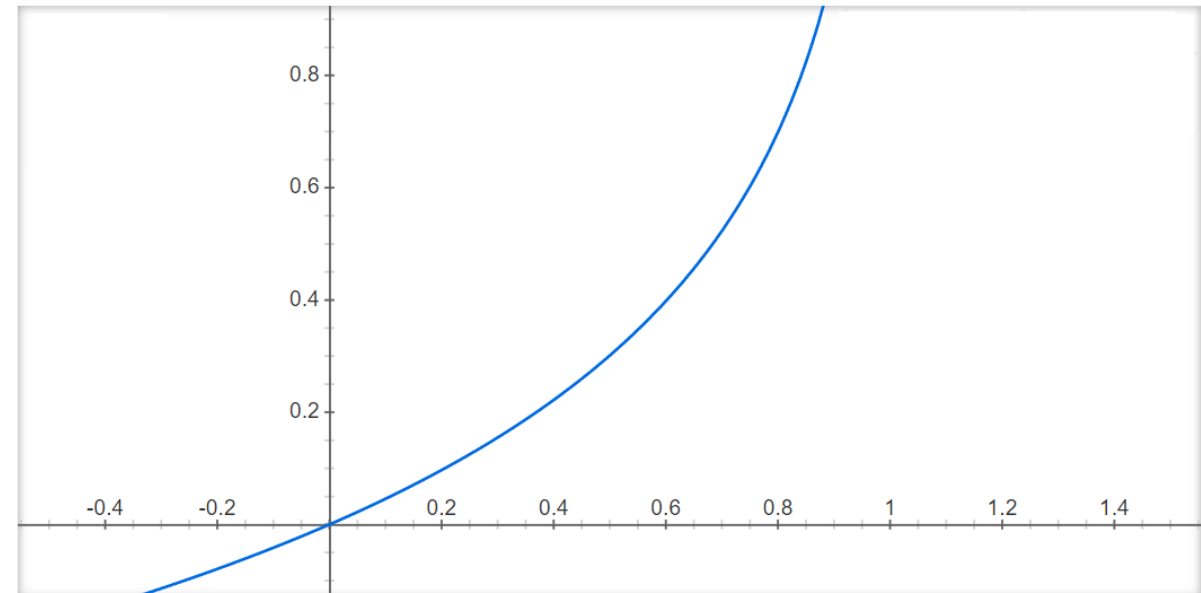
# Logistic regression – cost function

If  $Y = 1$  then,  
 $-\log(z)$  will be zero at  $h_{\theta}(x) = 1$



$-\log(z)$

If  $Y = 0$  then,  
 $-\log(1-z)$  will be zero at  $h_{\theta}(x) = 0$



$-(\log(1-z))$

# Logistic regression – cost function

## Logistic regression cost function

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

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

Note:  $y = 0$  or  $1$  always

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

To fit parameters  $\theta$ :

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

To make a prediction given new  $x$ :

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

# Logistic regression - optimization

## Gradient Descent

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

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

Repeat {

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

} (simultaneously update all  $\theta_j$ )

Repeat {

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

} (simultaneously update all  $\theta_j$ )

# Logistic regression – multiclass

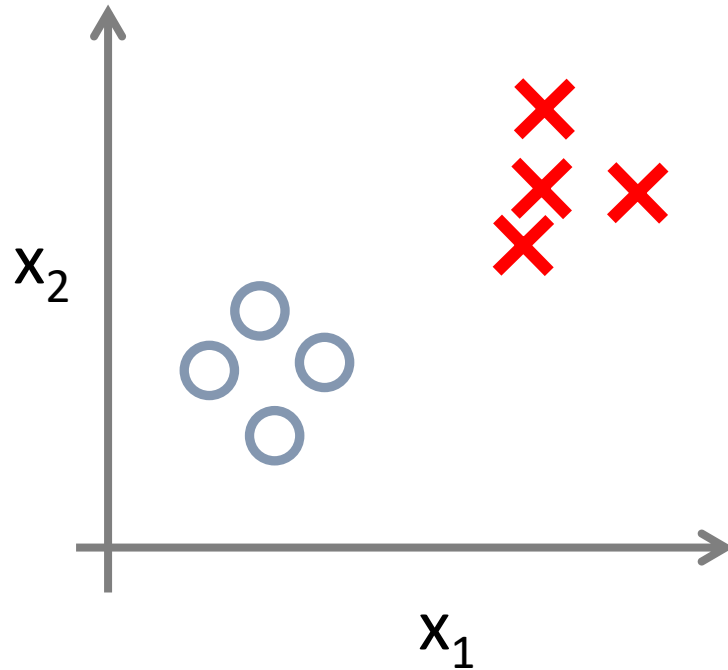
Email foldering/tagging: Work, Friends, Family, Hobby

Medical diagrams: Not ill, Cold, Flu

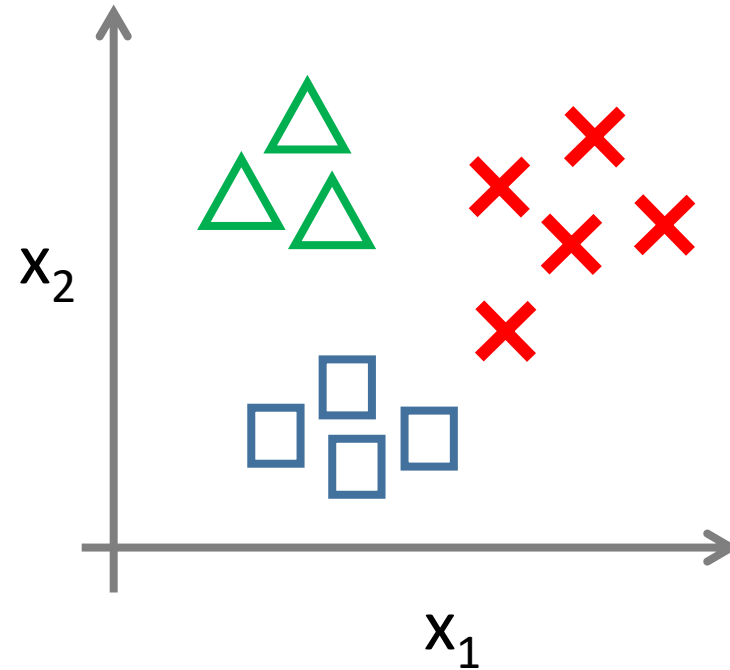
Weather: Sunny, Cloudy, Rain, Snow

# Logistic regression – multiclass

Binary classification:

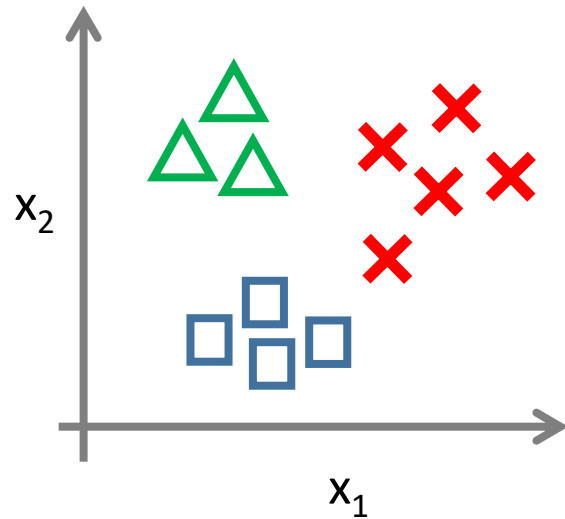



Multi-class classification:





# Logistic regression – multiclass

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

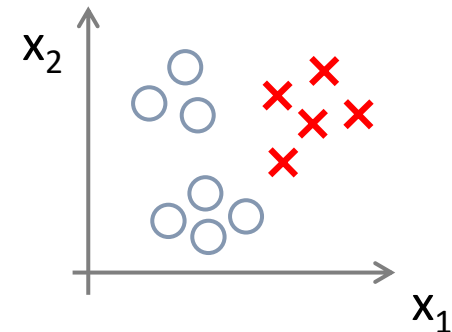
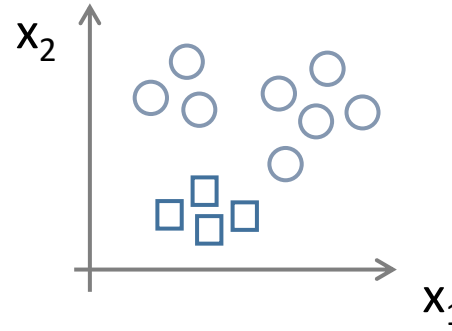
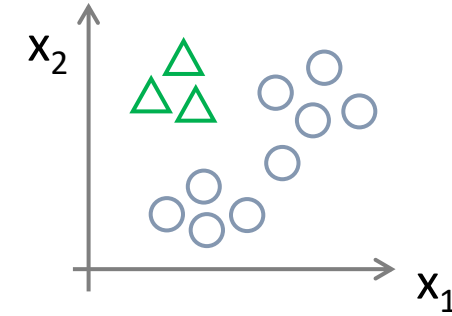


Class 1: 

Class 2: 

Class 3: 

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





# Logistic regression – multiclass

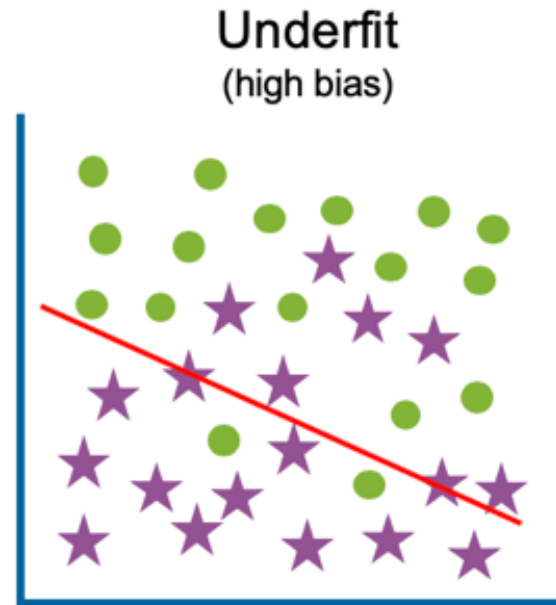
## One-vs-all

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

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

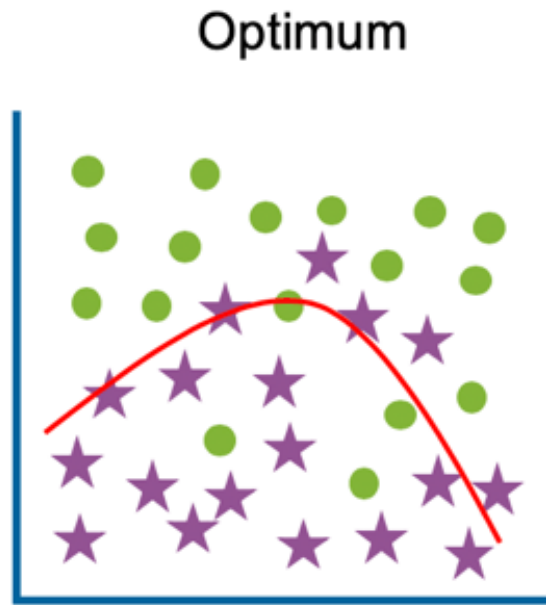
# Regularization - Introduction



High training error  
High test error

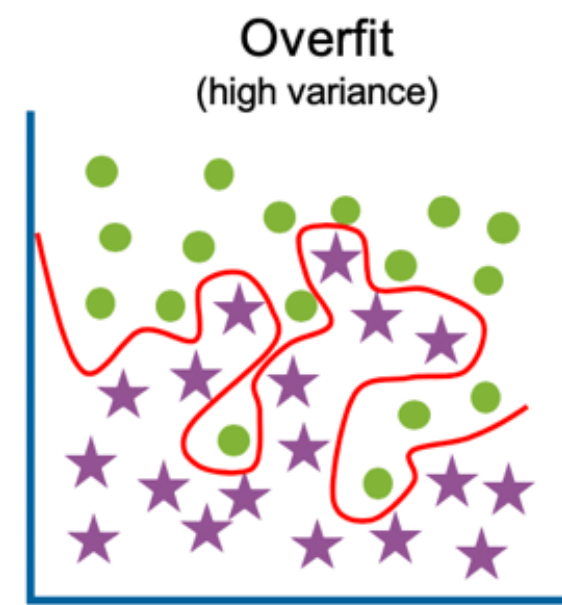
$$h_{\theta}(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$

( $g$  = sigmoid function)



Low training error  
Low test error

$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1^2 + \theta_4 x_2^2 + \theta_5 x_1 x_2)$$



Low training error  
High test error

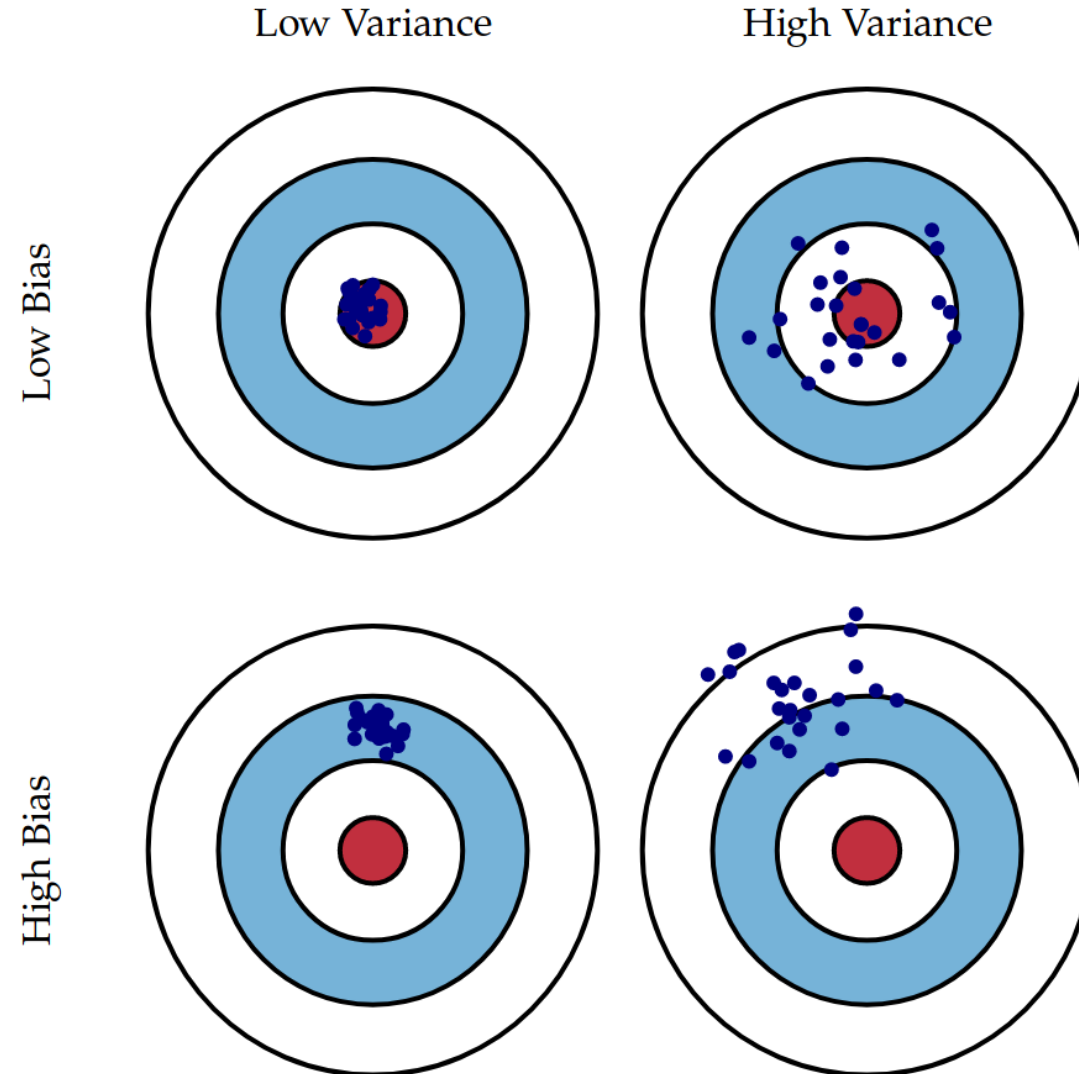
$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2 + \theta_5 x_1^2 x_2^3 + \theta_6 x_1^3 x_2 + \dots)$$

# Regularization – address overfitting

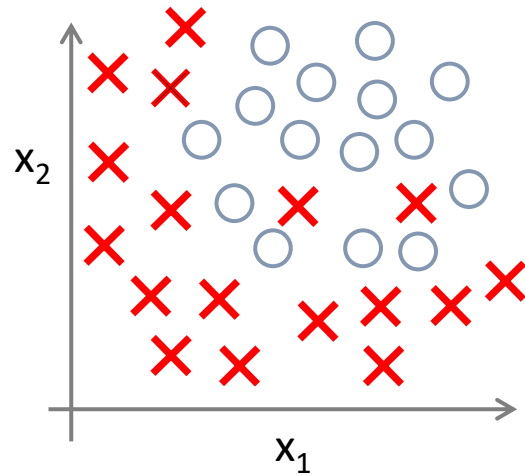
## Options:

1. Reduce number of features.
  - Manually select which features to keep.
  - Model selection algorithm.
2. Regularization.
  - Keep all the features, but reduce magnitude/values of parameters.
  - Works well when we have a lot of features, each of which contributes a bit to predicting.

# Regularization – bias-variance tradeoff



# Regularized Logistic regression

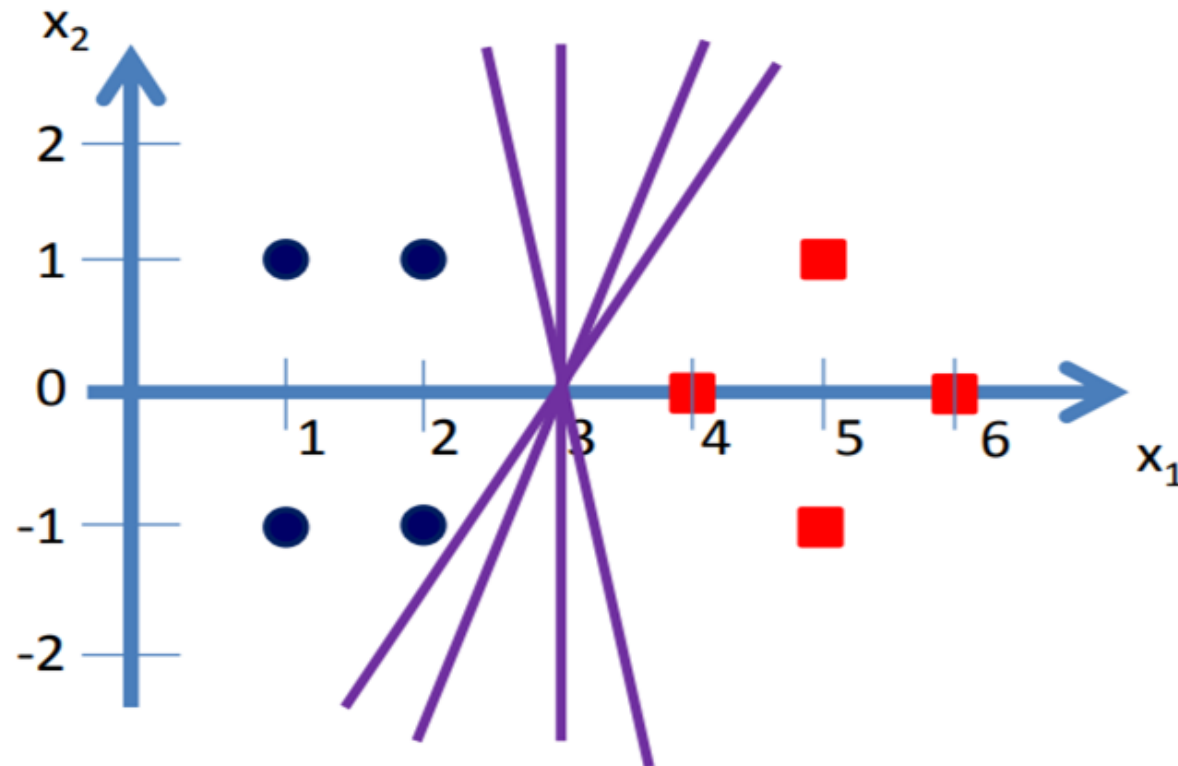


$$h_{\theta}(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_1^2 + \theta_3 x_1^2 x_2 + \theta_4 x_1^2 x_2^2 + \theta_5 x_1^2 x_2^3 + \dots)$$

Cost function: 
$$J(\theta) = \left[ -\frac{1}{m} \sum_{i=1}^m y^{(i)} \log(h_{\theta}(x^{(i)})) + (1 - y^{(i)}) \log(1 - h_{\theta}(x^{(i)})) \right] + \frac{\lambda}{2m} \sum_{j=1}^n \theta_j^2$$

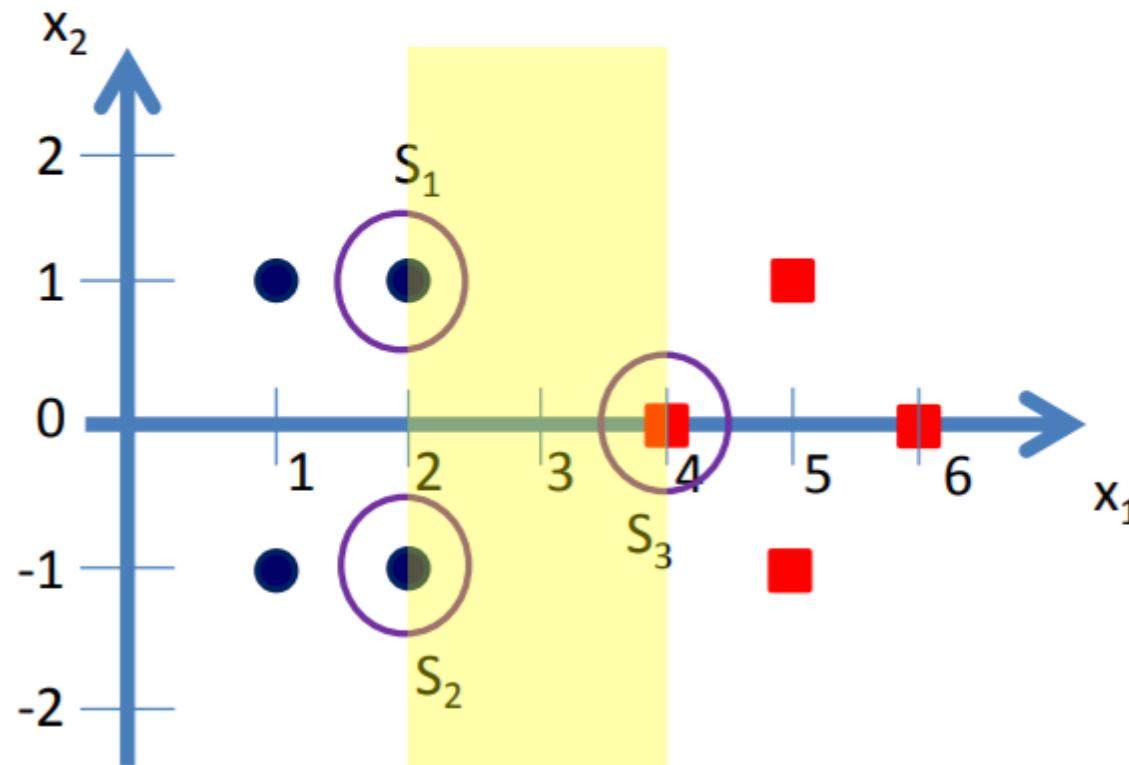
# Support Vector Machines (SVM)

- SVM algorithms are used in classification task of separating classes in feature space.



# Support Vector Machines (SVM)

- Here we select 3 Support Vectors to start with.
- They are  $S_1$ ,  $S_2$  and  $S_3$ .



$$S_1 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$$

$$S_2 = \begin{pmatrix} 2 \\ -1 \end{pmatrix}$$

$$S_3 = \begin{pmatrix} 4 \\ 0 \end{pmatrix}$$

# Support Vector Machines (SVM)

- Here we will use vectors augmented with a 1 as a bias input, and for clarity we will differentiate these with an over-tilde. That is:

$$s_1 = \begin{pmatrix} 2 \\ 1 \end{pmatrix}$$

$$s_2 = \begin{pmatrix} 2 \\ -1 \end{pmatrix}$$

$$s_3 = \begin{pmatrix} 4 \\ 0 \end{pmatrix}$$

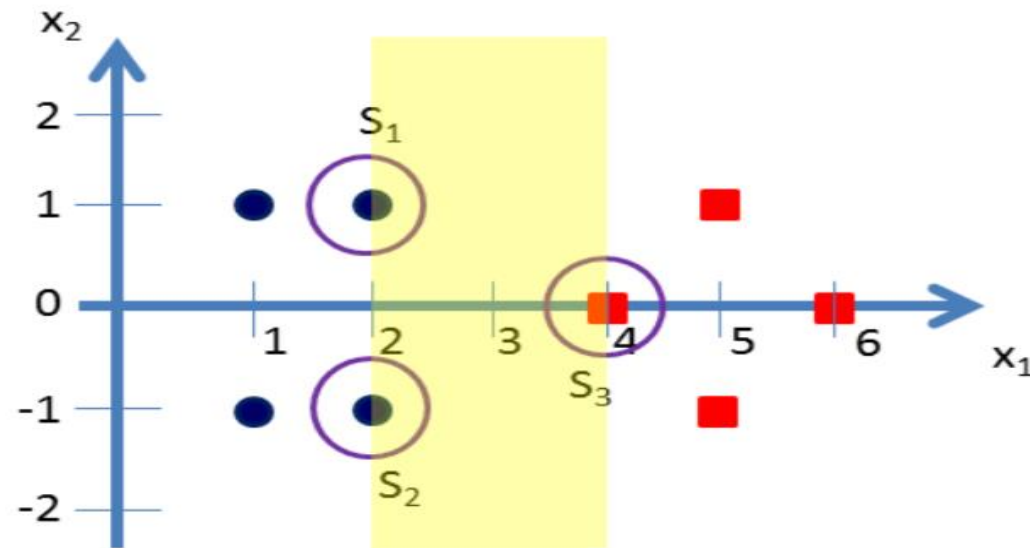
$$\tilde{s}_1 = \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix}$$

$$\tilde{s}_2 = \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix}$$

$$\tilde{s}_3 = \begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix}$$



# Support Vector Machines (SVM)



- Now we need to find 3 parameters  $\alpha_1$ ,  $\alpha_2$ , and  $\alpha_3$  based on the following 3 linear equations:

$$\alpha_1 \widetilde{S_1} \cdot \widetilde{S_1} + \alpha_2 \widetilde{S_2} \cdot \widetilde{S_1} + \alpha_3 \widetilde{S_3} \cdot \widetilde{S_1} = -1 \quad (-ve \text{ class})$$

$$\alpha_1 \widetilde{S_1} \cdot \widetilde{S_2} + \alpha_2 \widetilde{S_2} \cdot \widetilde{S_2} + \alpha_3 \widetilde{S_3} \cdot \widetilde{S_2} = -1 \quad (-ve \text{ class})$$

$$\alpha_1 \widetilde{S_1} \cdot \widetilde{S_3} + \alpha_2 \widetilde{S_2} \cdot \widetilde{S_3} + \alpha_3 \widetilde{S_3} \cdot \widetilde{S_3} = +1 \quad (+ve \text{ class})$$

# Support Vector Machines (SVM)

$$\alpha_1 \widetilde{S}_1 \cdot \widetilde{S}_1 + \alpha_2 \widetilde{S}_2 \cdot \widetilde{S}_1 + \alpha_3 \widetilde{S}_3 \cdot \widetilde{S}_1 = -1 \text{ (-ve class)}$$

$$\alpha_1 \widetilde{S}_1 \cdot \widetilde{S}_2 + \alpha_2 \widetilde{S}_2 \cdot \widetilde{S}_2 + \alpha_3 \widetilde{S}_3 \cdot \widetilde{S}_2 = -1 \text{ (-ve class)}$$

$$\alpha_1 \widetilde{S}_1 \cdot \widetilde{S}_3 + \alpha_2 \widetilde{S}_2 \cdot \widetilde{S}_3 + \alpha_3 \widetilde{S}_3 \cdot \widetilde{S}_3 = +1 \text{ (+ve class)}$$

- Let's substitute the values for  $\widetilde{S}_1$ ,  $\widetilde{S}_2$  and  $\widetilde{S}_3$  in the above equations.

$$\widetilde{S}_1 = \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} \quad \widetilde{S}_2 = \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} \quad \widetilde{S}_3 = \begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix}$$

$$\alpha_1 \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} + \alpha_3 \begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} = -1$$

$$\alpha_1 \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} + \alpha_3 \begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} = -1$$

$$\alpha_1 \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix} + \alpha_3 \begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix} = +1$$

# Support Vector Machines (SVM)

$$\alpha_1 \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} + \alpha_3 \begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} = -1$$

$$\alpha_1 \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} + \alpha_3 \begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} = -1$$

$$\alpha_1 \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix} + \alpha_3 \begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix} \cdot \begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix} = +1$$

- After simplification we get:

$$6\alpha_1 + 4\alpha_2 + 9\alpha_3 = -1$$

$$4\alpha_1 + 6\alpha_2 + 9\alpha_3 = -1$$

$$9\alpha_1 + 9\alpha_2 + 17\alpha_3 = +1$$

- Simplifying the above 3 simultaneous equations we get:  $\alpha_1 = \alpha_2 = -3.25$  and  $\alpha_3 = 3.5$ .

# Support Vector Machines (SVM)

$$\alpha_1 = \alpha_2 = -3.25 \text{ and } \alpha_3 = 3.5$$

- The hyper plane that discriminates the positive class from the negative class is given by:

$$\tilde{w} = \sum_i \alpha_i \tilde{S}_i$$

- Substituting the values we get:

$$\begin{aligned} \tilde{w} &= \alpha_1 \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} + \alpha_2 \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} + \alpha_3 \begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix} \\ \tilde{w} &= (-3.25) \cdot \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} + (-3.25) \cdot \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} + (3.5) \cdot \begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ -3 \end{pmatrix} \end{aligned}$$

$$\begin{aligned} \tilde{S}_1 &= \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} \\ \tilde{S}_2 &= \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} \\ \tilde{S}_3 &= \begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix} \end{aligned}$$

# Support Vector Machines (SVM)

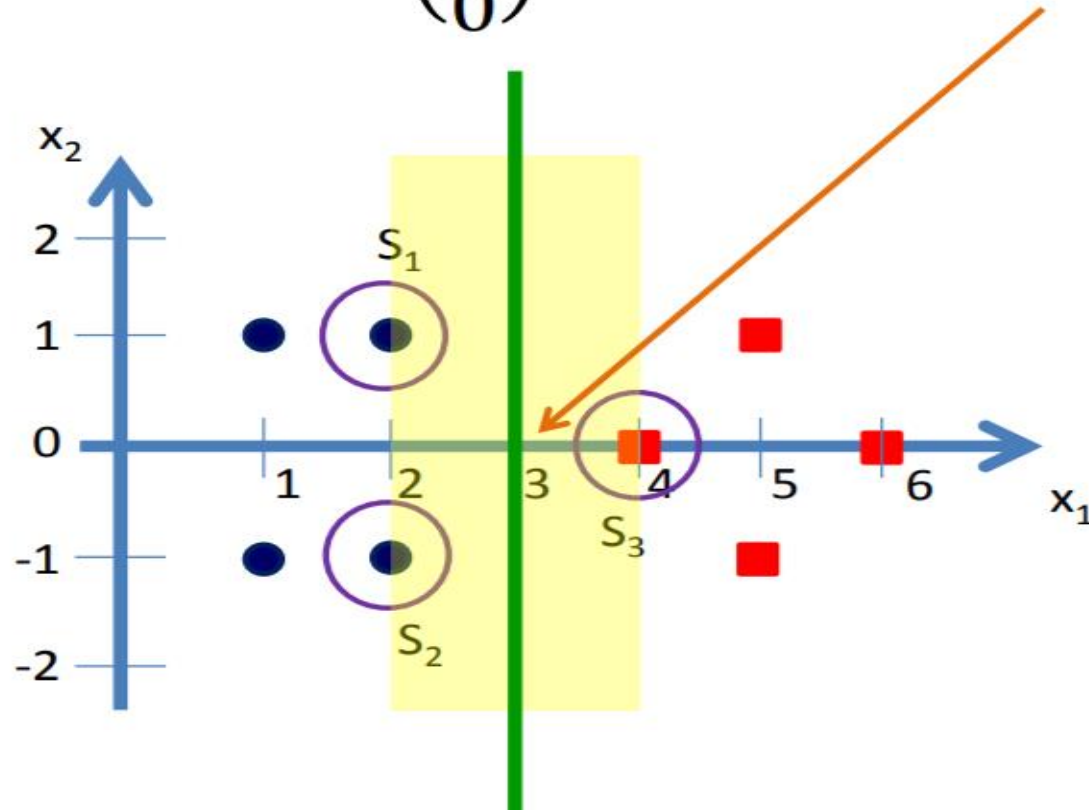
$$\tilde{w} = (-3.25) \cdot \begin{pmatrix} 2 \\ 1 \\ 1 \end{pmatrix} + (-3.25) \cdot \begin{pmatrix} 2 \\ -1 \\ 1 \end{pmatrix} + (3.5) \cdot \begin{pmatrix} 4 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ -3 \end{pmatrix}$$

- Our vectors are augmented with a bias.
- Hence we can equate the entry in  $\tilde{w}$  as the hyper plane with an offset  $b$ .
- Therefore the separating hyper plane equation  $y = wx + b$  with  $w = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$  and offset  $b = -3$ .

# Support Vector Machines (SVM)

## Support Vector Machines

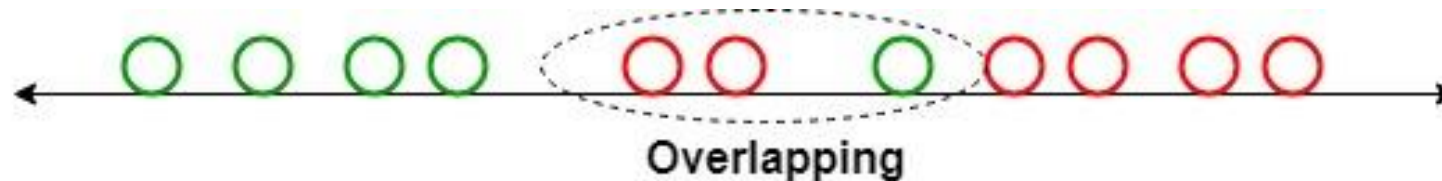
- $y = wx + b$  with  $w = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$  and offset  $b = -3$ .





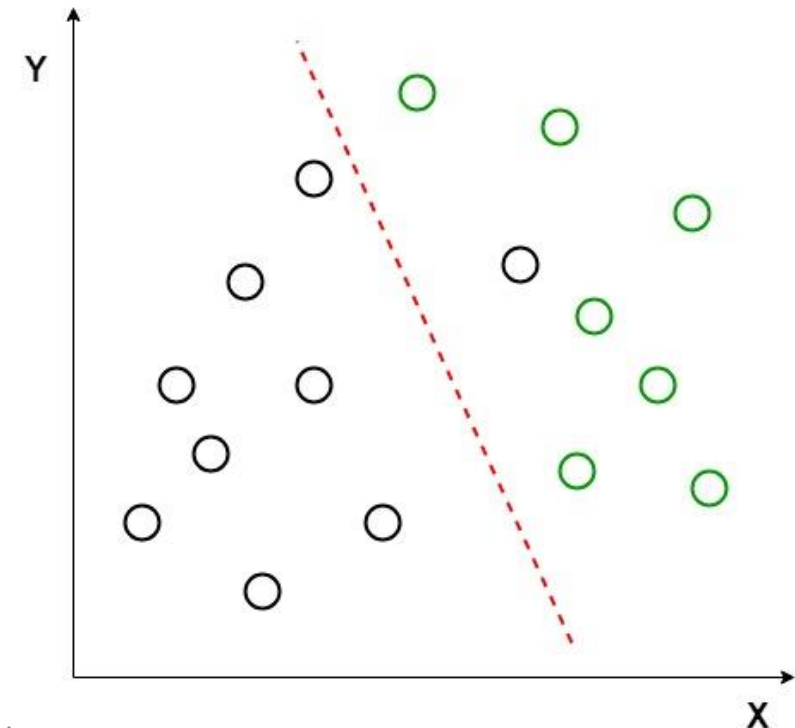
# Linear Discriminant Analysis (LDA)

- LDA is dimensionality reduction technique that is commonly used for supervised classification problems.
- It is used for modelling differences in groups i.e. separating two or more classes.
- It is used to project the features in higher dimension space into a lower dimension space.
- If we have two classes with multiple features and need to separate them efficiently. When we classify them using a single feature, then it may show overlapping.



# Linear Discriminant Analysis (LDA)

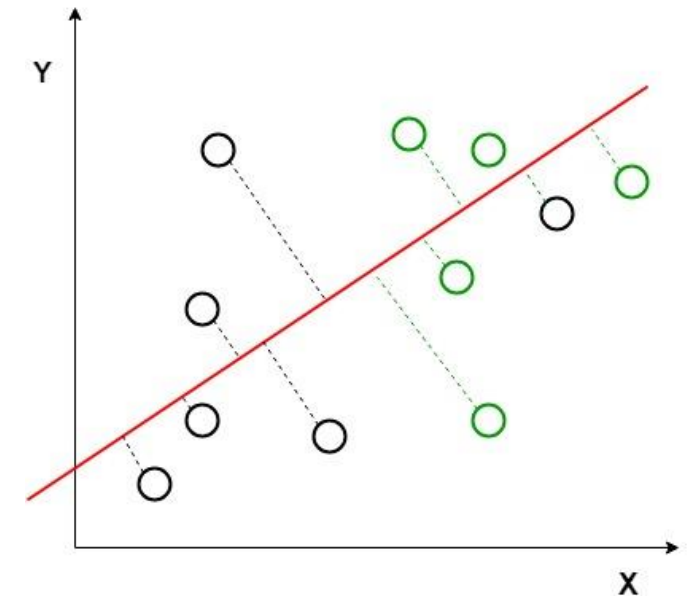
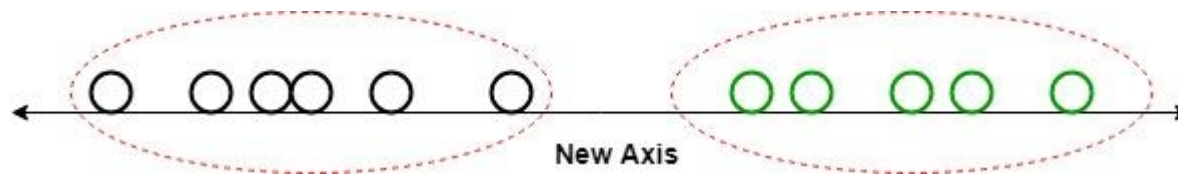
- Let us assume we have to classify two different classes having two sets of data points in a 2-dimensional plane.
- It is impossible to draw a straight line in a 2-d plane that can separate these data points efficiently but using LDA we can dimensionally reduce the 2-D plane into the 1-D plane.





# Linear Discriminant Analysis (LDA)

- LDA uses an X-Y axis to create a new axis by separating them using a straight line and projecting data onto a new axis.
- Hence we can maximize the separation between these classes and reduce the 2-D plane into 1-D.
- To create a new axis, Linear Discriminant Analysis uses the following criteria:
  - maximizes the distance between means of two classes.
  - minimizes the variance within the individual class.



# Linear Discriminant Analysis (LDA)

- Applications of LDA:
  - Face Recognition
  - Medical
  - Customer Identification
  - Predictions
  - Learning

***Thank you***