McMaster University ENGINEERING

W BOOTH SCHOOL OF ENGINEERING PRACTICE AND TECHNOLOGY

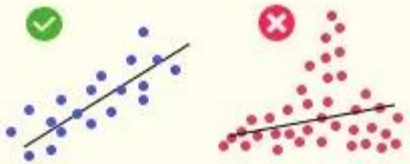
SEP 785: Machine Learning

Lecture 6: Linear Models for Classifications + Kernels + Naïve Bayes

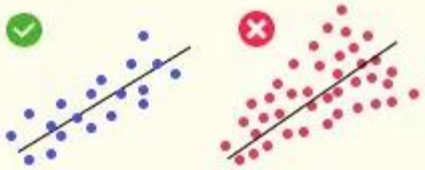
Instructor: Dr. Dalia Mahmoud, PhD
(Mechanical Engineering, McMaster University)
Email: mahmoudd@mcmaster.ca

Recap

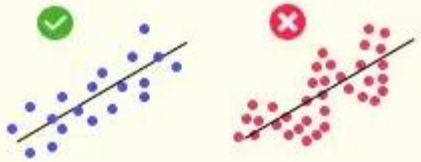
1. Linearity
(Linear relationship between Y and each X)



2. Homoscedasticity
(Equal variance)



4. Independence
(of observations. Includes "no autocorrelation")



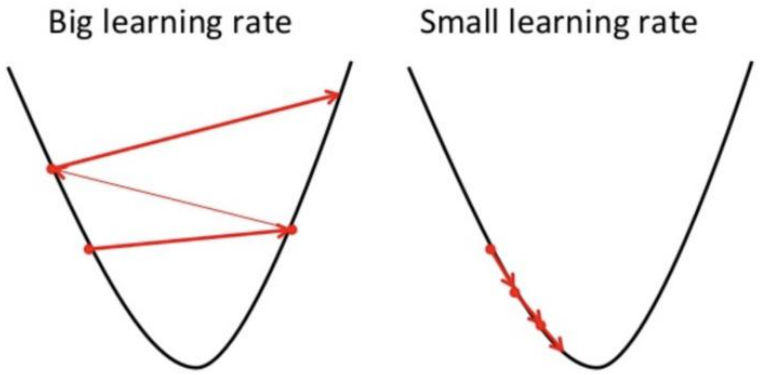
5. Lack of Multicollinearity
(Predictors are not correlated with each other)

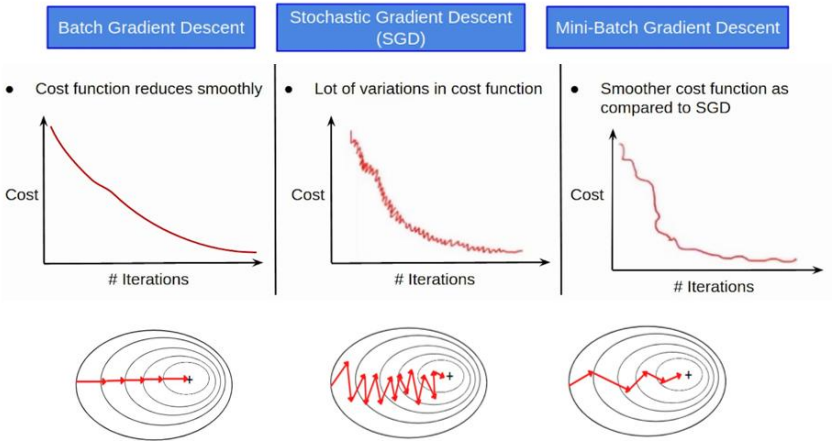
$X_1 + X_2$ $X_1 \sim X_2$

5. Lack of Multicollinearity
(Predictors are not correlated with each other)

$X_1 + X_2$ $X_1 \sim X_2$

6. Absence of endogeneity
(No correlation between predictors and errors.)





$$J(w) = \sum_{i=1}^{n}(y_i - (w_0 + \sum_{j=1}^{p} w_j x_{ij}))^2 + \lambda \sum_{j=1}^{p} |w_j|$$

$$J(w) = \sum_{i=1}^{n}(y_i - (w_0 + \sum_{j=1}^{p} w_j x_{ij}))^2 + \lambda \sum_{j=1}^{p} w_j^2$$

Project and Lecture timeline updated

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

Week	lecture		Project
25th of Feb	Logistic Regression + SVM + Naïve Bayes	In person	Data Collection and Pre-processing
4th of March	Models Evaluation + Miscellanies	In person	
11th of March	Python hands on	Online	
18th of March	Python hands on	Online	Choosing and Applying ML Algorithm(s)
25th of March	Python hands on	Online	
1st of April	No lecture --- Eid Vacation		
8th of April	Ensembles	In person	Video Presentation Deliverable
15th of April	Unsupervised Learning	In person	
22nd of April	Introduction to Neural Networks	In person	Peer Review and Discussion
29th of April	TBD	TBD	

Intended Learning Outcomes

1. Explain the **theory and assumptions** behind logistic regression.
2. Explain the principle of maximum margin classification in linear SVM.
3. Describe the role of support vectors and the hinge loss function.
4. Explain the **Bayes' theorem** and the **assumption of feature independence** in Naïve Bayes.
5. Compare **Naïve Bayes with logistic regression and SVM** in terms of assumptions and applications.

Contents

- Logistic Regression
- SVM
- Naïve Bayes

Logistic Regression

- Up to this point, the methods we have seen have centered around modeling and the prediction of a quantitative response variable.
- Linear regression (and Ridge, LASSO) perform well under these situations.
- When the response variable is categorical, then the problem is no longer called a regression problem but is instead labeled as a classification problem.
- The goal is to attempt to classify each observation into a category (aka, class or cluster) defined by Y, based on a set of predictor variables X

Give Some Examples on Categorical response Variables ?

Categorical response Variables Examples

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

- **Email Classification:** Spam (1) or Not Spam (0)
- **Disease Diagnosis**: Has disease (1) or No disease (0)
- **Loan Approval**: Approved (1) or Rejected (0)
- **Pass/Fail in an Exam**: Pass (1) or Fail (0)

Binary

- **Fruit Type**: Apple, Banana, Orange
- **Car Brand**: Toyota, Ford, Honda
- **Movie Genre**: Comedy, Action, Drama, Horror

Multinomial

- **Education Level**: High School, Bachelor's, Master's, PhD
- **Customer Ratings**: 1 star, 2 stars, 3 stars, 4 stars, 5 stars
- **Economic Status**: Low-income, Middle-income, High-income

Ordinal

Logistic Regression

- Given a dataset:

$$\{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \cdots, (\boldsymbol{x}_N, y_N)\}$$

- where the y are categorical (sometimes referred to as qualitative), we would like to be able to predict which category y takes on given x.

Linear regression does not work well, or is not appropriate at all, in this setting. Why ?

Why Logistic Regression ?

- A categorical variable y could be encoded to be quantitative. For example, if y represents departments of McMaster undergrads, then y could take on the values:

$$y = \begin{cases} 1 & if \text{ Computer Science (CS)} \\ 2 & if \text{ Statistics} \\ 3 & \text{otherwise} \end{cases}$$

- A linear regression could be used to predict y from x.

What would be wrong with such a model?

Why Logistic Regression ?

Issue 1: Implies an Arbitrary Ordering

- Assigning numbers to categories (e.g., 1 = CS, 2 = Statistics, 3 = Other) forces an order that doesn't exist.
- Changing the numbering (e.g., swapping CS and Statistics) would change predictions, making the model unreliable.

Issue 2: Assumes Equal Distance Between Categories

- Linear regression treats y=1 to y=2 (CS → Statistics) the same as y=2 to y=3 (Statistics → Other).
- But there is no meaningful numerical "distance" between categories.

Issue 3: Predictions Can Be Meaningless

- The model may predict fractional or out-of-range values (e.g., y = 2.3), which don't make sense for categorical data.

Binary Classification

- The response variable **Y** has **two categories**, with a natural ordering.
- Example: Categorizing **McMaster students** based on their housing situation

$$y = \begin{cases} 1 & if \text{ lives in the Quad} \\ 0 & \text{otherwise} \end{cases}$$

Using Linear Regression for Classification

- Predict **Y** using features like **sex, athlete status, concentration, GPA, etc.**
- Apply a decision rule:
  - If $\hat{y} \geq 0.5$, predict **Quad residence**
  - If $\hat{y} < 0.5$, predict **other housing**

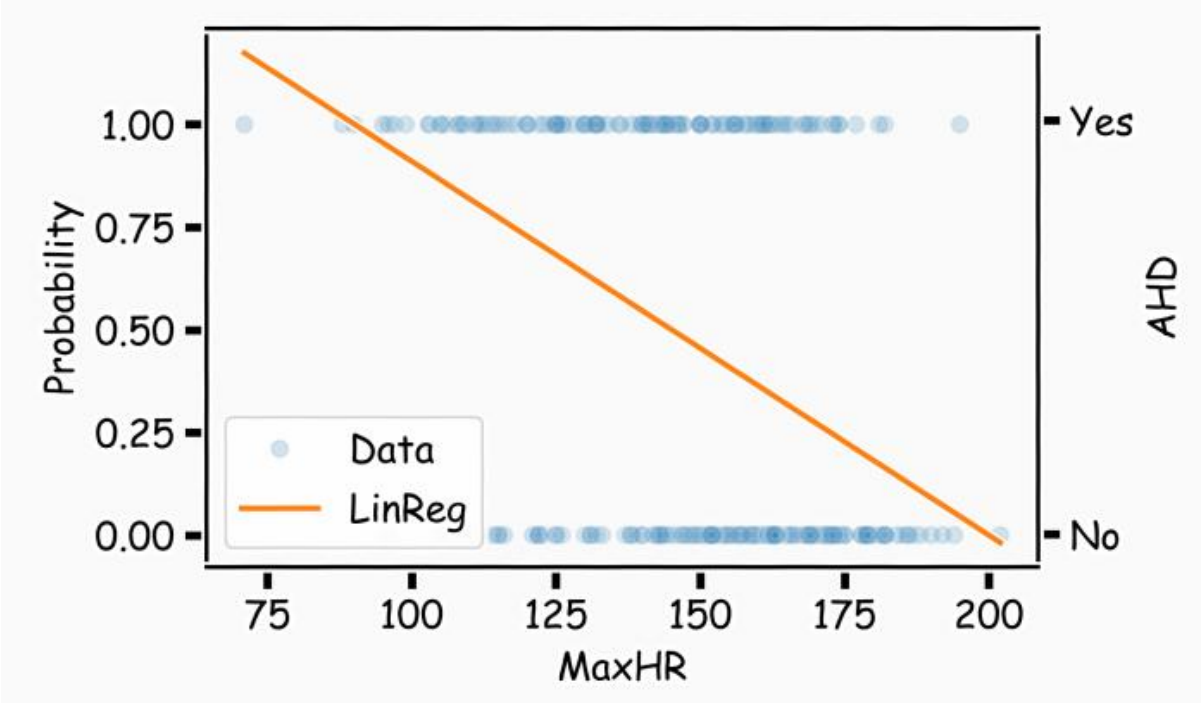
Binary Classification

Issue: Nonsensical Predictions

- Linear regression models **p(y=1)**, the probability of a student living in the Quad.
- However, it can produce predictions below 0 or above 1, which don't make sense for probabilities.

Why This Happens?

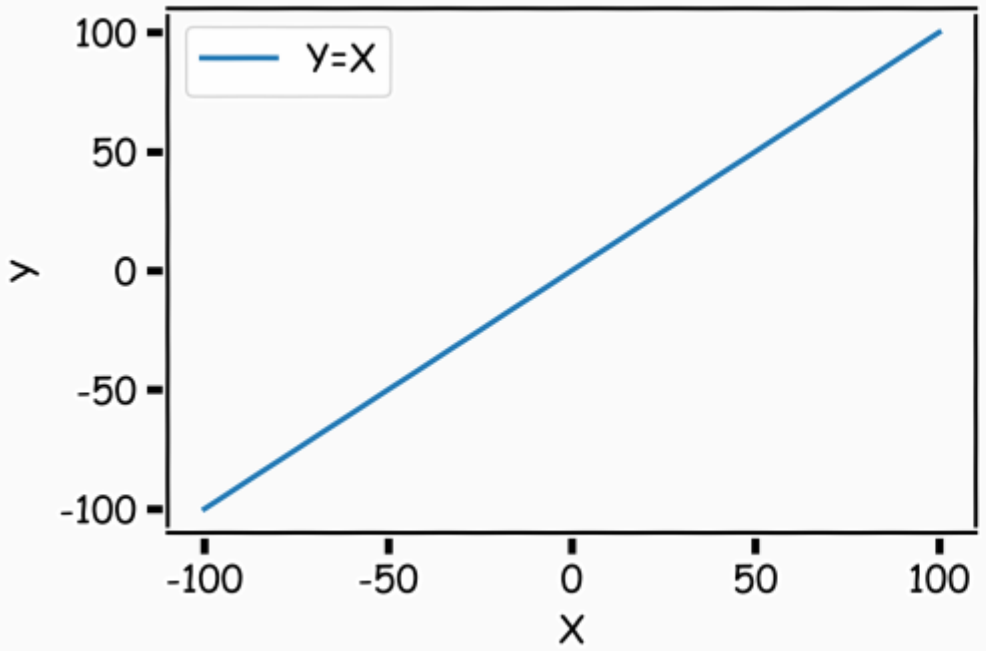
- Linear regression assumes y can take any real value, but probabilities must be between [0,1].
- It doesn't properly handle the S-curve relationship in classification problems.



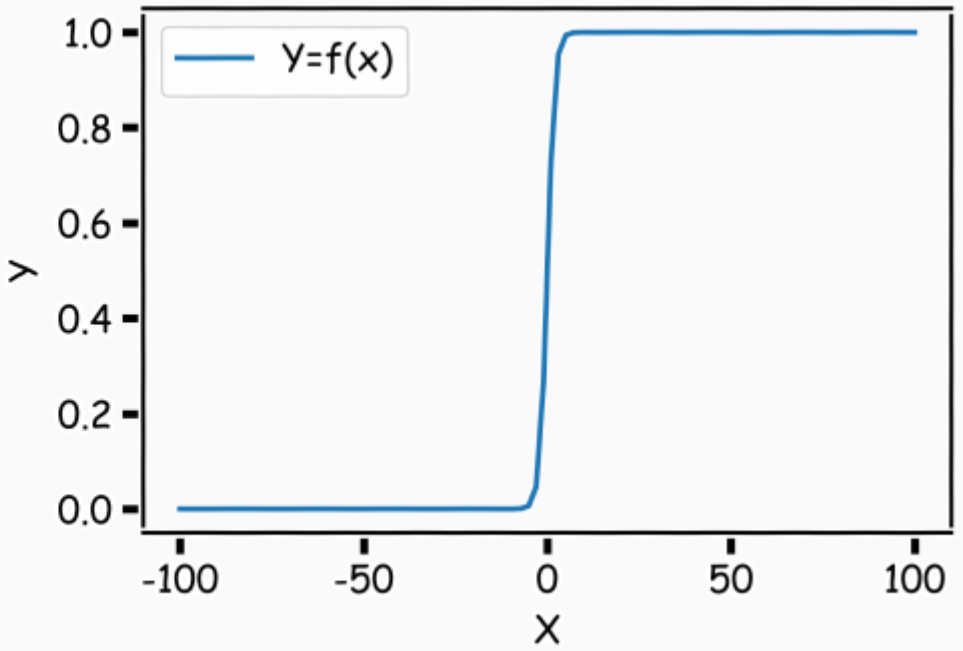
Better Solution: Logistic Regression

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

- Uses a **sigmoid function** to ensure outputs are always between **0 and 1**.
- Interprets predictions as **probabilities**, making classification **more meaningful**.



$Y=f(x)$



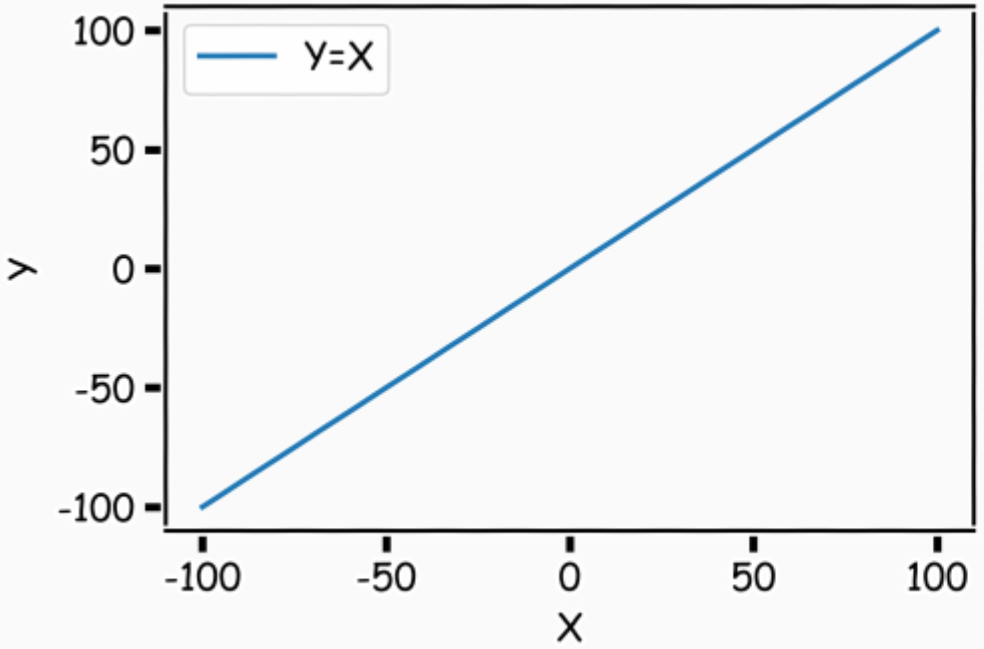
$$z = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n$$

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

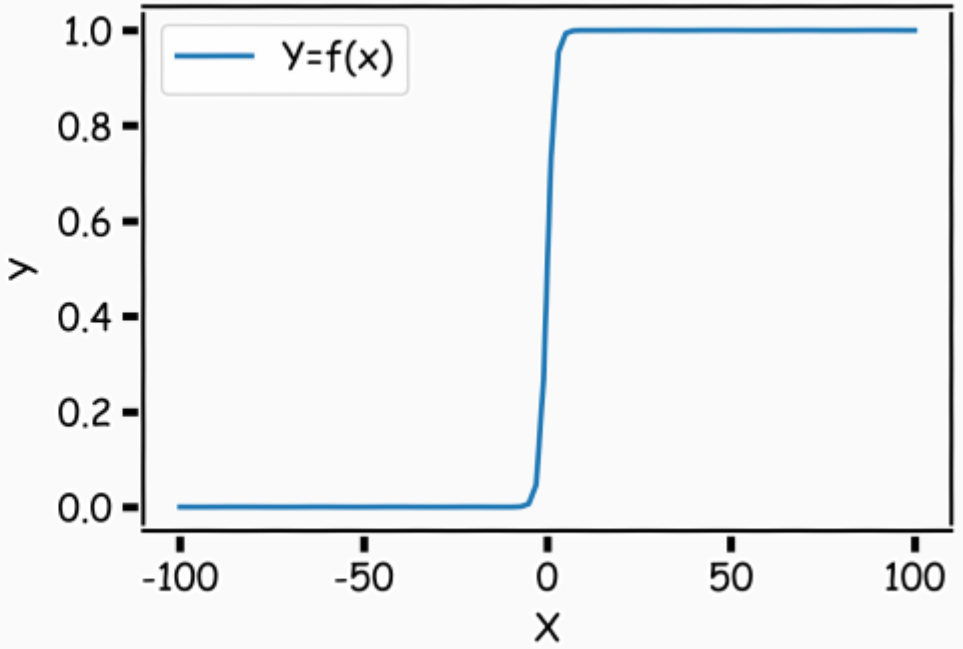
Better Solution: Logistic Regression

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

Logistic Regression solves the issue of probability estimates falling outside the **[0,1]** range. It achieves this by applying the **logistic (sigmoid) function** to model **P(Y=1):**



$Y=f(x)$



$$z = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \ldots + \beta_n X_n$$

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

Estimation in Logistic Regression

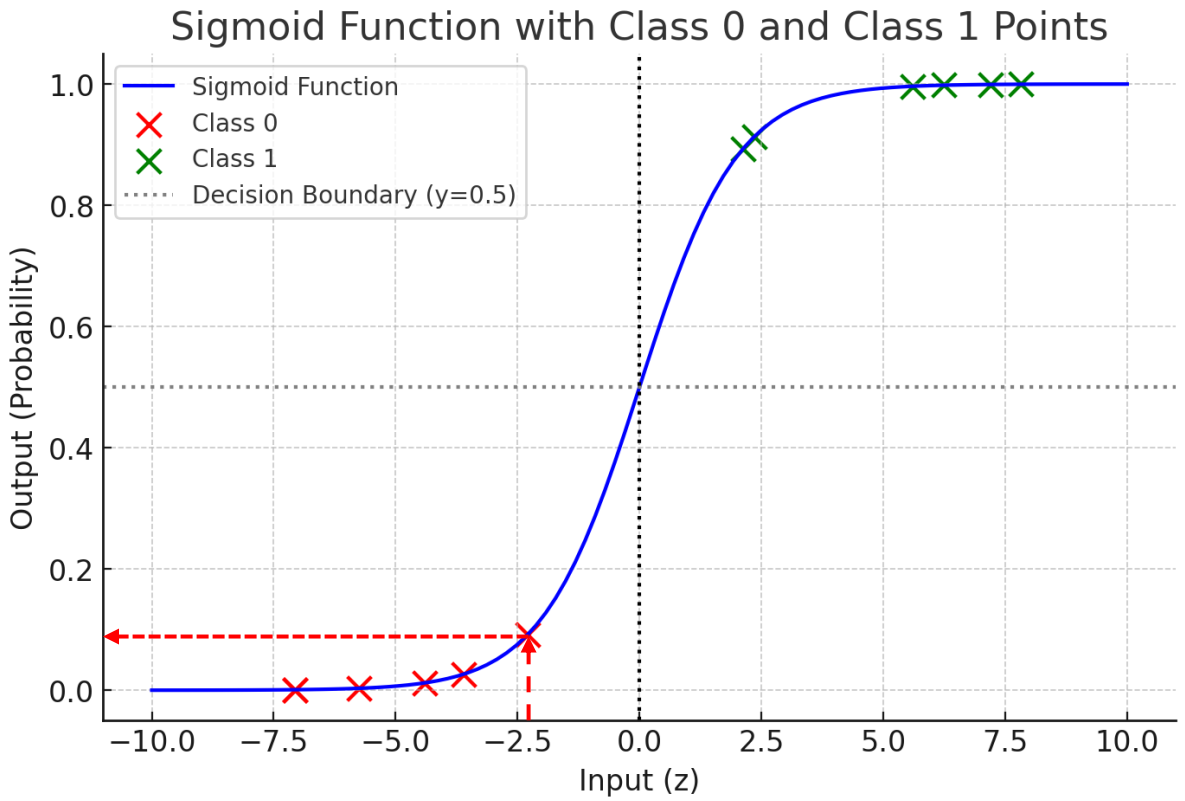
McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

Given a dataset with n observations, where each Y_i is either 0 or 1, and X_i represents the predictor variables, the probability of Y_i=1is given by:

$$P(Y_i = 1|X_i) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_i)}}$$

Similarly, the probability of Y_i=0 is:

$$P(Y_i = 0|X_i) = 1 - \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_i)}}$$



Estimation in Logistic Regression

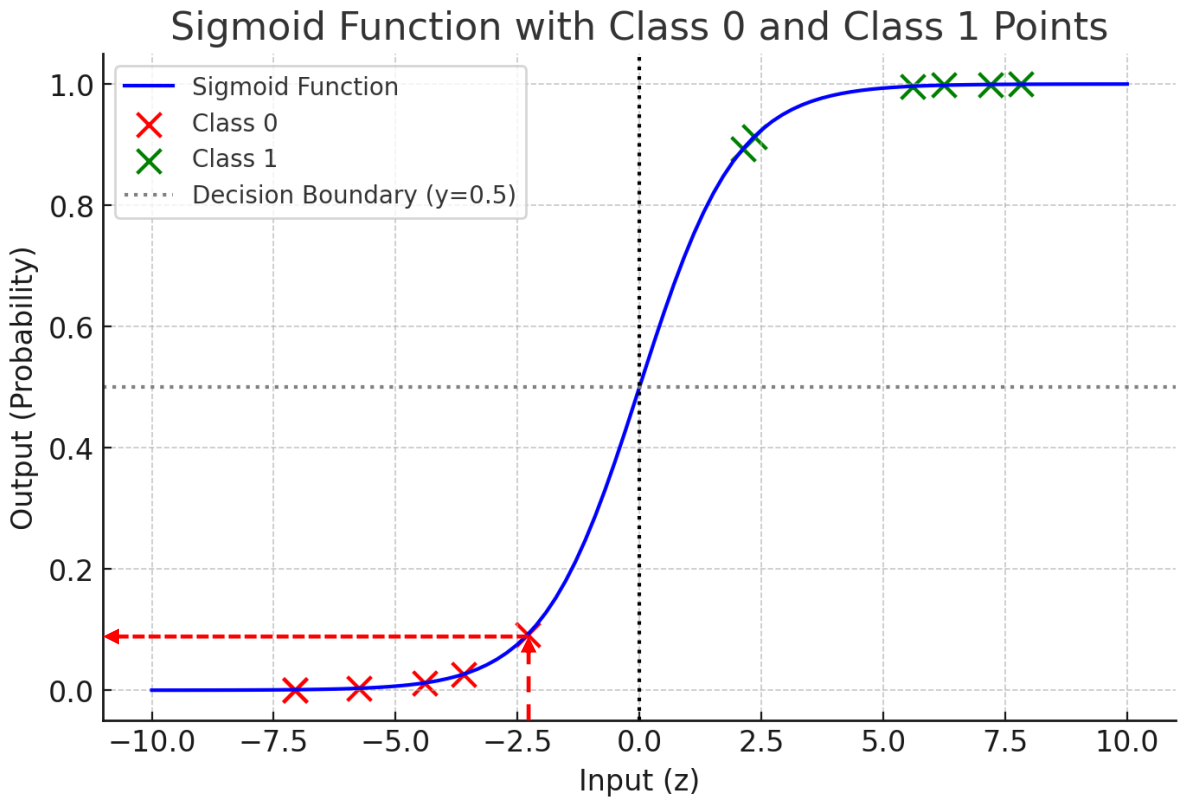
McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

The likelihood function, which gives the probability of **observing the actual data**, is:

$$L(\beta_0, \beta_1) = \prod_{i=1}^{n} P(Y_i|X_i)$$

Since Yi can be either 0 or 1, we can rewrite it as:

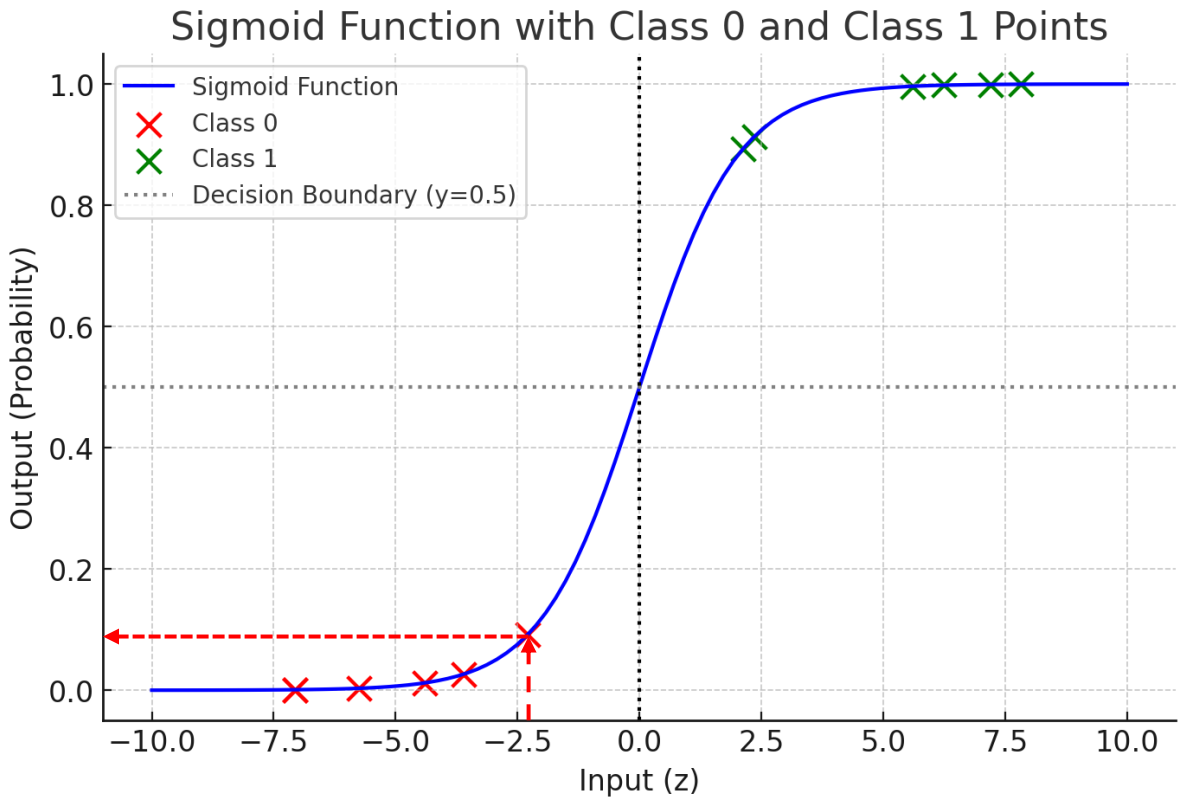
$$L(\beta_0, \beta_1) = \prod_{i=1}^{n} \left(\frac{1}{1 + e^{-(\beta_0 + \beta_1 X_i)}} \right)^{Y_i} \left(1 - \frac{1}{1 + e^{-(\beta_0 + \beta_1 X_i)}} \right)^{(1-Y_i)}$$



Log-Likelihood Function

Since the likelihood function involves a **product**, it's easier to work with the **log-likelihood**, which turns it into a **sum**:

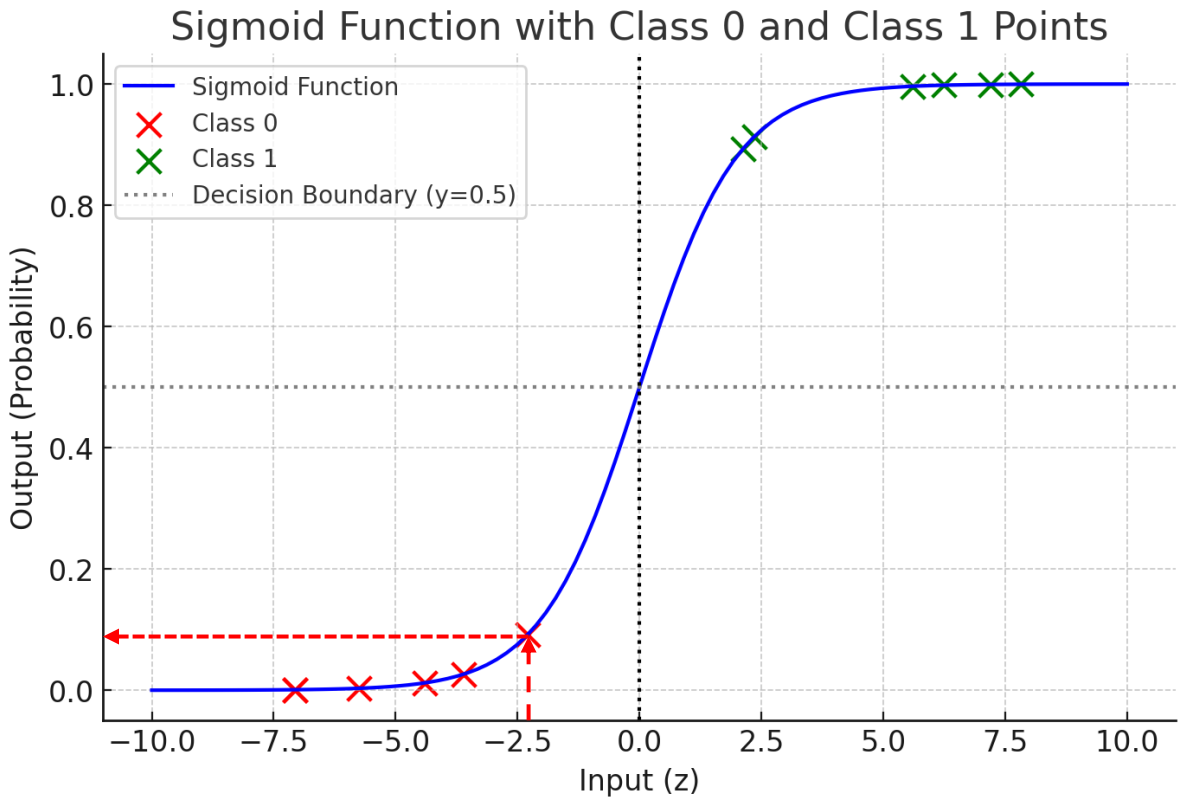
$$\log L(\beta_0, \beta_1) = \sum_{i=1}^{n} [Y_i \log P(Y_i|X_i) + (1 - Y_i) \log(1 - P(Y_i|X_i))]$$



Estimation in Logistic Regression

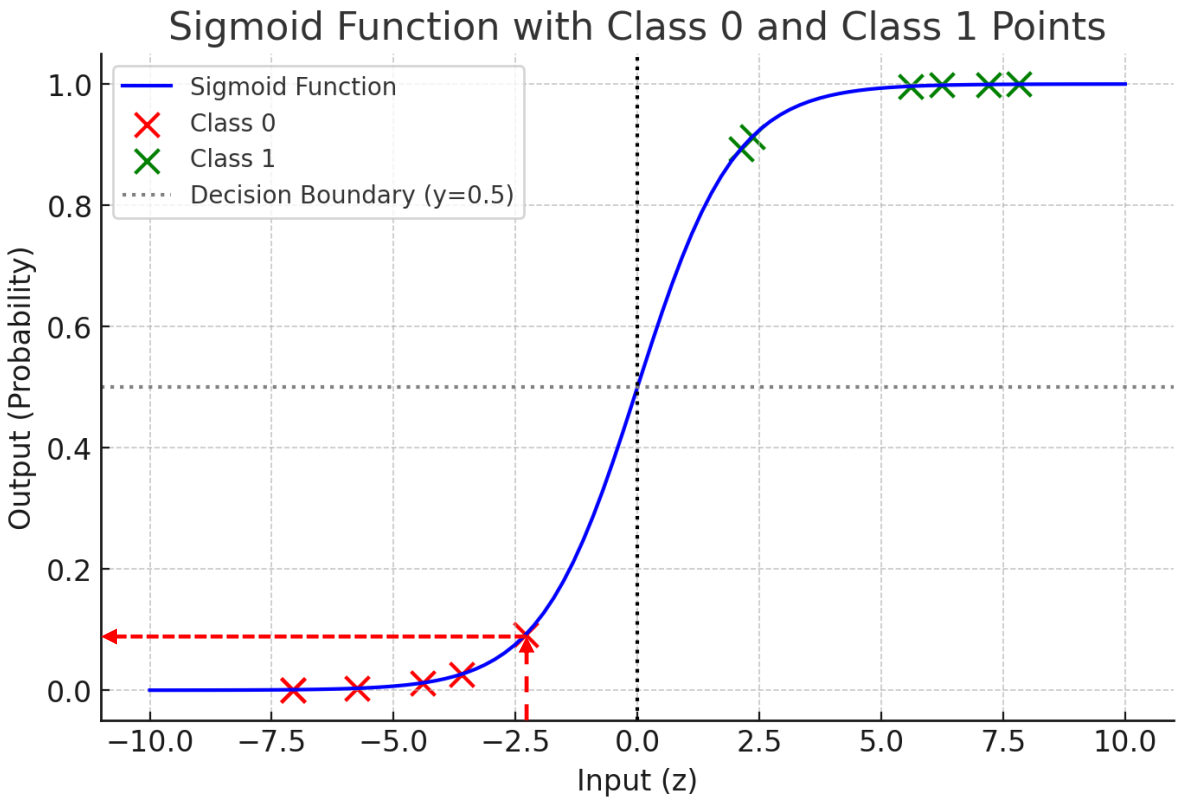
McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

Since logistic regression models the **probability** of an outcome using the logistic function, it cannot be estimated using **ordinary least squares (OLS)** like linear regression. Instead, we use **Maximum Likelihood Estimation (MLE)** to determine the best-fitting parameters.



Maximum Likelihood Estimation (MLE)

- To estimate β_0 and β_1 , we find the values that **maximize** the log-likelihood function.
- Since there is no closed-form solution (like in linear regression), we use **numerical optimization techniques** such as:
  - Gradient Descent
  - Newton-Raphson Method
  - Iteratively Reweighted Least Squares (IRLS)

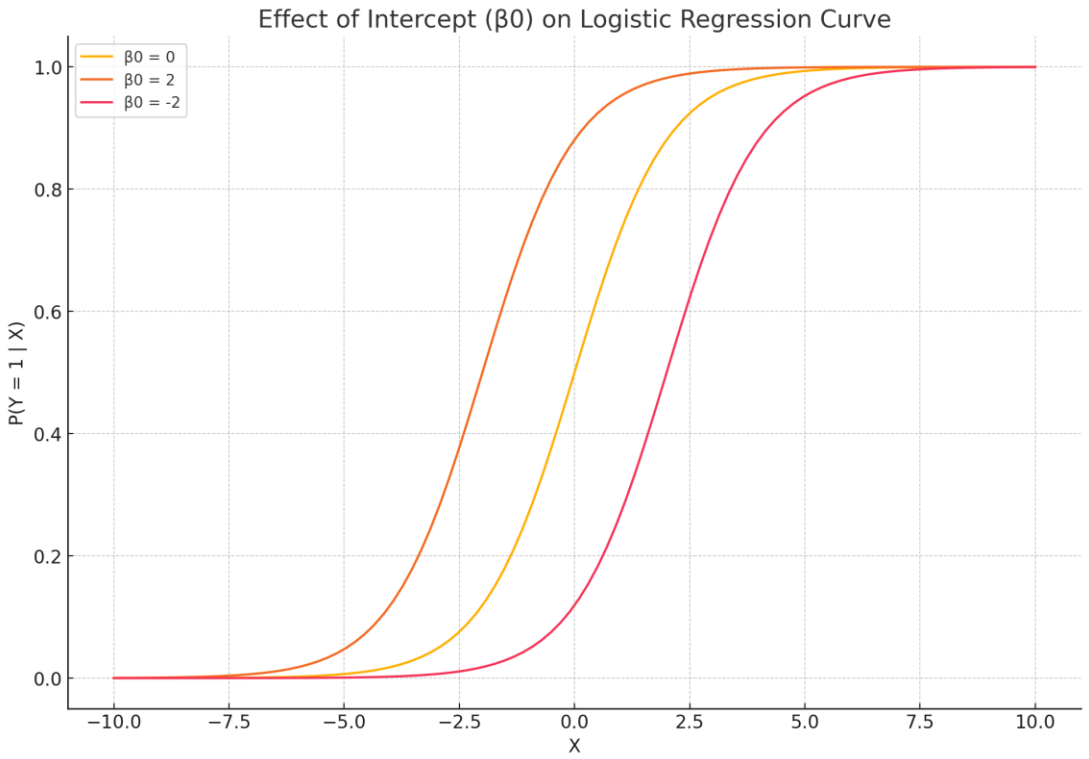


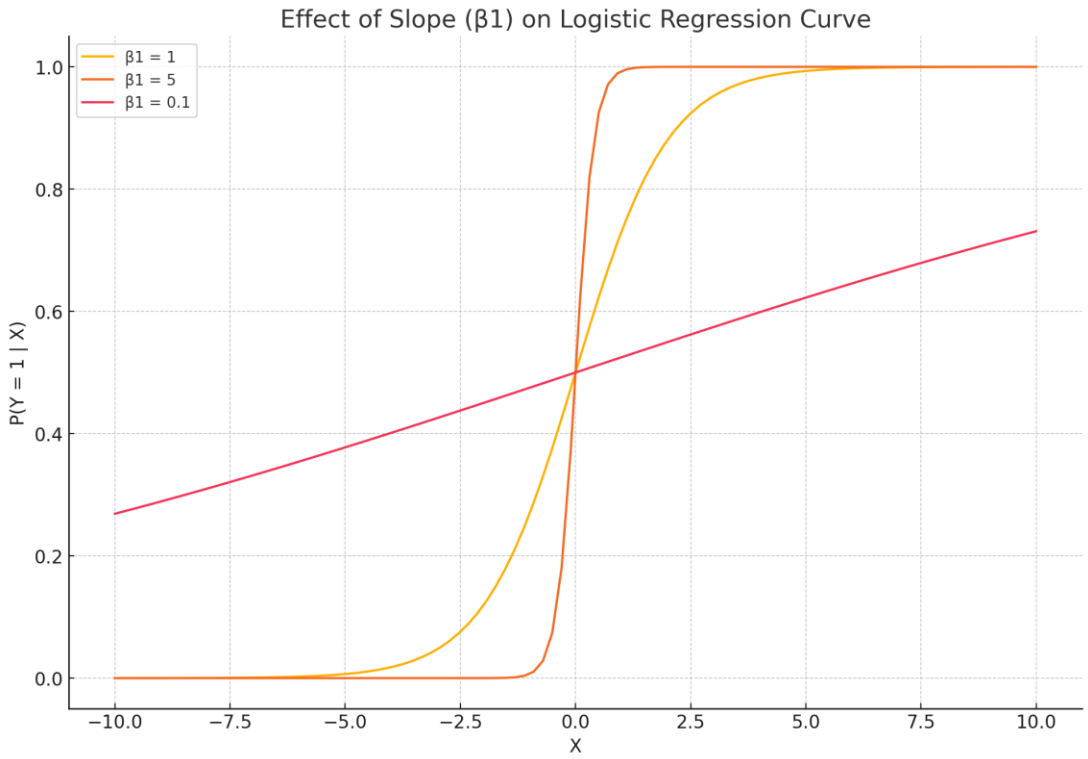
Coefficients in Logistic Regression

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

As a result, the model will predict P(Y=1) with an S-shaped curve, which is the general shape of the logistic function. β_0 shifts the curve right or left. β_1 controls how steep the S-shaped curve is.

$$P(Y=1) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}} = \frac{1}{1+e^{-(\beta_0+\beta_1 X)}}$$





Log Odds in Logistic Regression

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

With a little bit of algebraic work, the logistic model can be rewritten as:

$$P(Y=1) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}} = \frac{1}{1+e^{-(\beta_0+\beta_1 X)}} \quad \Rightarrow \quad \ln\left(\frac{P(Y=1)}{1-P(Y=1)}\right) = \beta_0 + \beta_1 X.$$

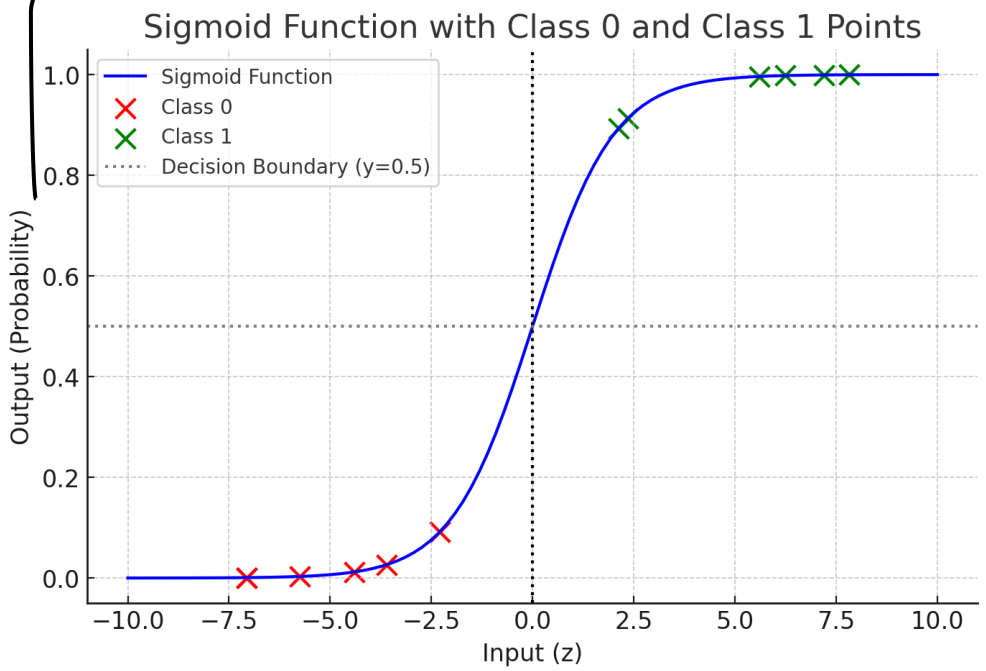
In Logistic Regression, the quantity **P(Y=1) / (1 - P(Y=1))** represents the **odds** of Y=1. This means Logistic Regression models the **log-odds** as a linear function of the predictors **X**.

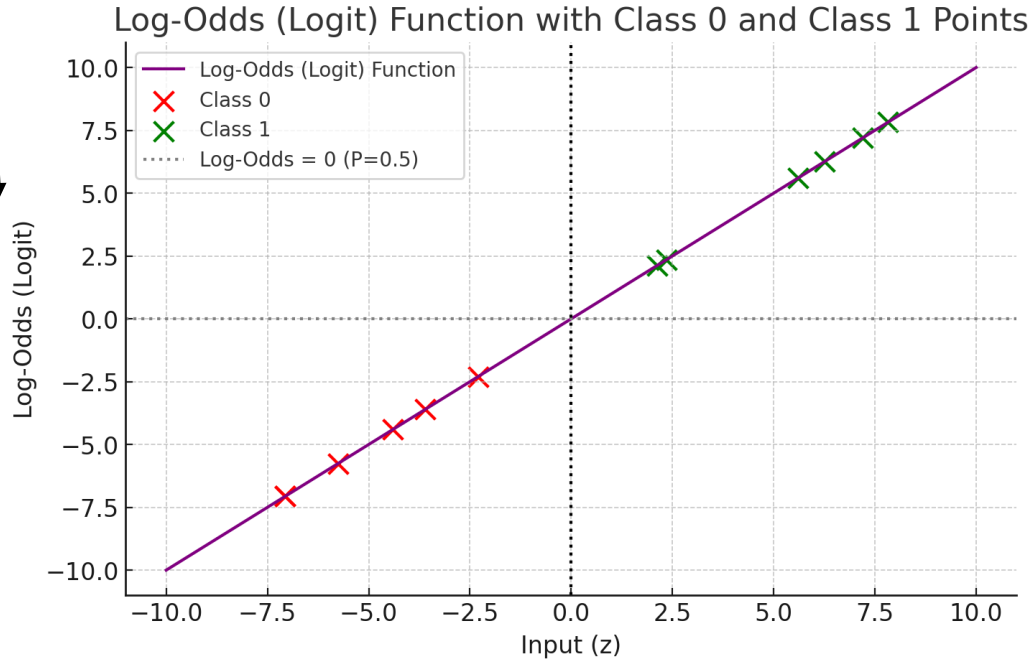
Log Odds in Logistic Regression

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

$$\text{logit}(p) = \ln\left(\frac{p}{1-p}\right)$$

β_0 is the intercept.
$\beta_1,\beta_2,\ldots,\beta_n$ are the coefficients of the predictor variables $x_1,x_2,\ldots,x_n$

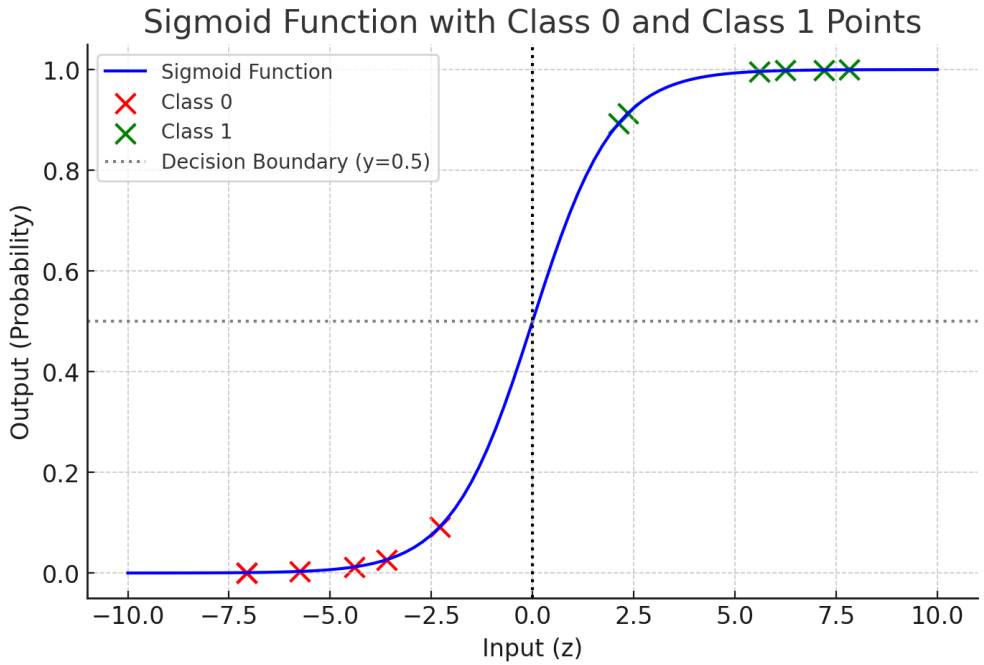


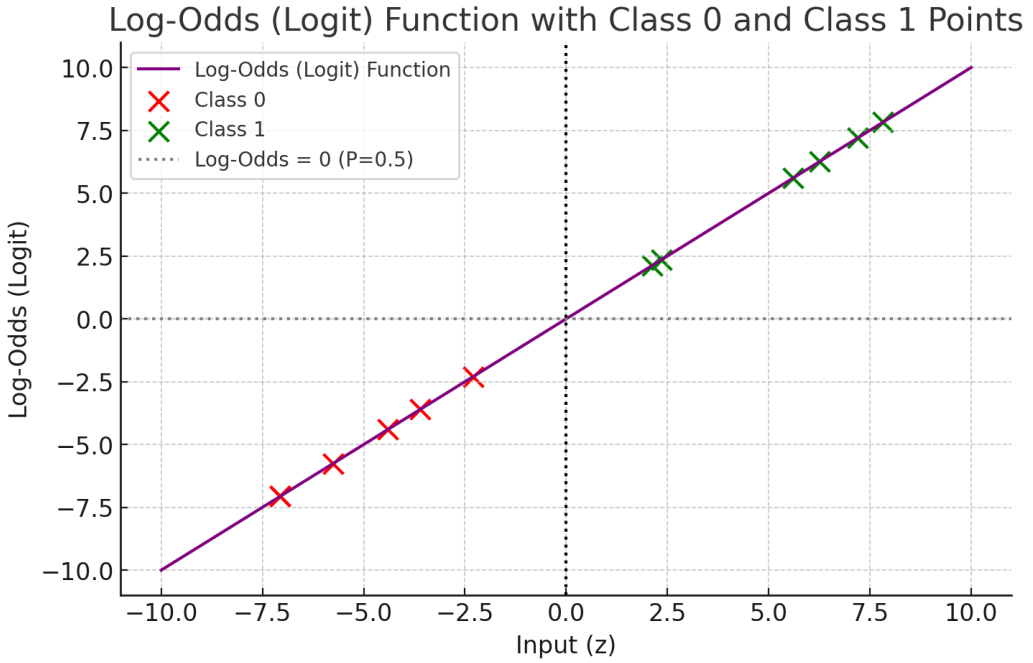


Log Odds in Logistic Regression

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

Linear Relationship: Log-odds enable a linear relationship between predictors and the outcome.
Bounded Probabilities: Ensures predicted probabilities stay between 0 and 1.
Interpretability: Coefficients are easier to interpret and can be converted to odds ratios.
Mapping to Probability: Logistic function converts log-odds back to valid probabilities.





Logistic Regression Steps

1. Define the logistic regression model.
2. Write the likelihood functions.
3. Estimate the parameters using MLE
4. Use the model to predict probabilities
5. Calculate the odds
6. Interpret the coefficients in terms of odds
7. Plot the logistic curve and make prediction

PSA	Prostate Cancer
3.8	cancer
3.4	cancer
2.9	cancer
2.8	cancer
2.7	cancer
2.1	cancer
1.6	cancer
2.5	healthy
2	healthy
1.7	healthy
1.4	healthy
1.2	healthy
0.9	healthy
0.8	healthy

Logistic Regression Steps

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

1. Define the logistic regression model.

$$p(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 \cdot x)}}$$

$p(x)$ is the predicted probability of having cancer (i.e., $y = 1$).

β_0 is the intercept.

β_1 is the coefficient for PSA (x).

x is the PSA value.

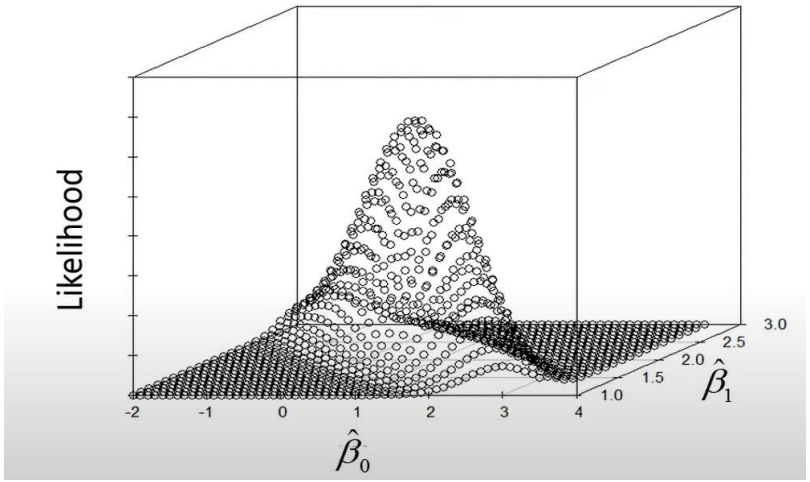
2. Write the likelihood functions

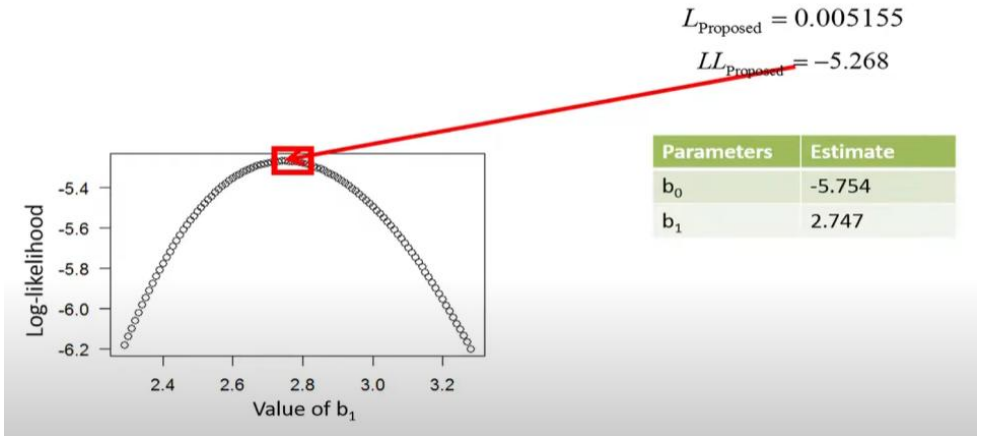
$$L(\beta_0, \beta_1) = \prod_{i=1}^{n} p(x_i)^{y_i} (1 - p(x_i))^{(1-y_i)}$$

PSA	Prostate Cancer	y_i
3.8	cancer	1
3.4	cancer	1
2.9	cancer	1
2.8	cancer	1
2.7	cancer	1
2.1	cancer	1
1.6	cancer	1
2.5	healthy	0
2	healthy	0
1.7	healthy	0
1.4	healthy	0
1.2	healthy	0
0.9	healthy	0
0.8	healthy	0

Logistic Regression Steps

3. Estimate the parameters using MLE





$$\log L(\beta_0, \beta_1) = \sum_{i=1}^{n} \left[y_i(\beta_0 + \beta_1 X_i) - \log(1 + e^{\beta_0 + \beta_1 X_i}) \right]$$

PSA	Prostate Cancer	y_i
3.8	cancer	1
3.4	cancer	1
2.9	cancer	1
2.8	cancer	1
2.7	cancer	1
2.1	cancer	1
1.6	cancer	1
2.5	healthy	0
2	healthy	0
1.7	healthy	0
1.4	healthy	0
1.2	healthy	0
0.9	healthy	0
0.8	healthy	0

Logistic Regression Steps

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

4. Use the model to predict probabilities

$$p(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 \cdot x)}}$$

$p(x)$ is the predicted probability of having cancer (i.e., $y = 1$).

β_0 is the intercept.

β_1 is the coefficient for PSA (x).

x is the PSA value.

Parameters	Estimate
b_0	-5.754
b_1	2.747

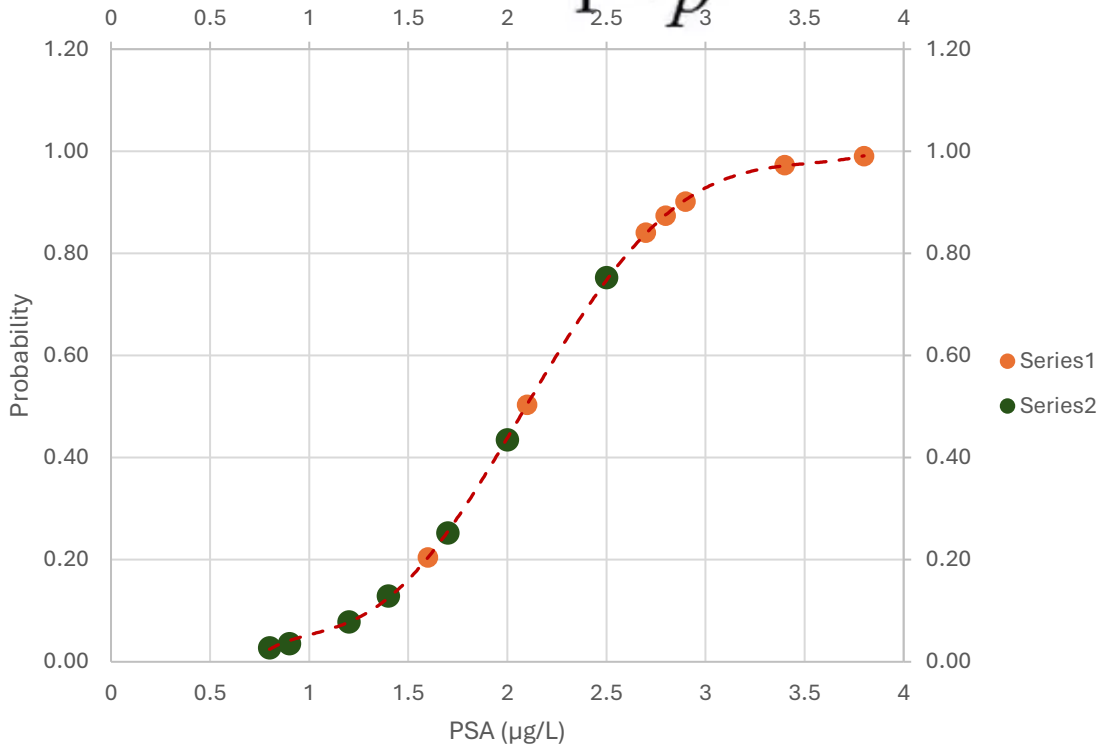
PSA	Prostate Cancer	y_i	P(x)
3.8	cancer	1	???
3.4	cancer	1	???
2.9	cancer	1	???
2.8	cancer	1	???
2.7	cancer	1	???
2.1	cancer	1	???
1.6	cancer	1	???
2.5	healthy	0	???
2	healthy	0	???
1.7	healthy	0	???
1.4	healthy	0	???
1.2	healthy	0	???
0.9	healthy	0	???
0.8	healthy	0	???

Logistic Regression Steps

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

5. Calculate the odds

$$odds = \frac{p}{1-p}$$



PSA	Prostate Cancer	y_i	P(x)	$\frac{p}{1-p}$
3.8	cancer	1	0.99	???
3.4	cancer	1	0.97	???
2.9	cancer	1	0.90	???
2.8	cancer	1	0.87	???
2.7	cancer	1	0.84	???
2.1	cancer	1	0.50	???
1.6	cancer	1	0.20	???
2.5	healthy	0	0.75	???
2	healthy	0	0.44	???
1.7	healthy	0	0.25	???
1.4	healthy	0	0.13	???
1.2	healthy	0	0.08	???
0.9	healthy	0	0.04	???
0.8	healthy	0	0.03	???

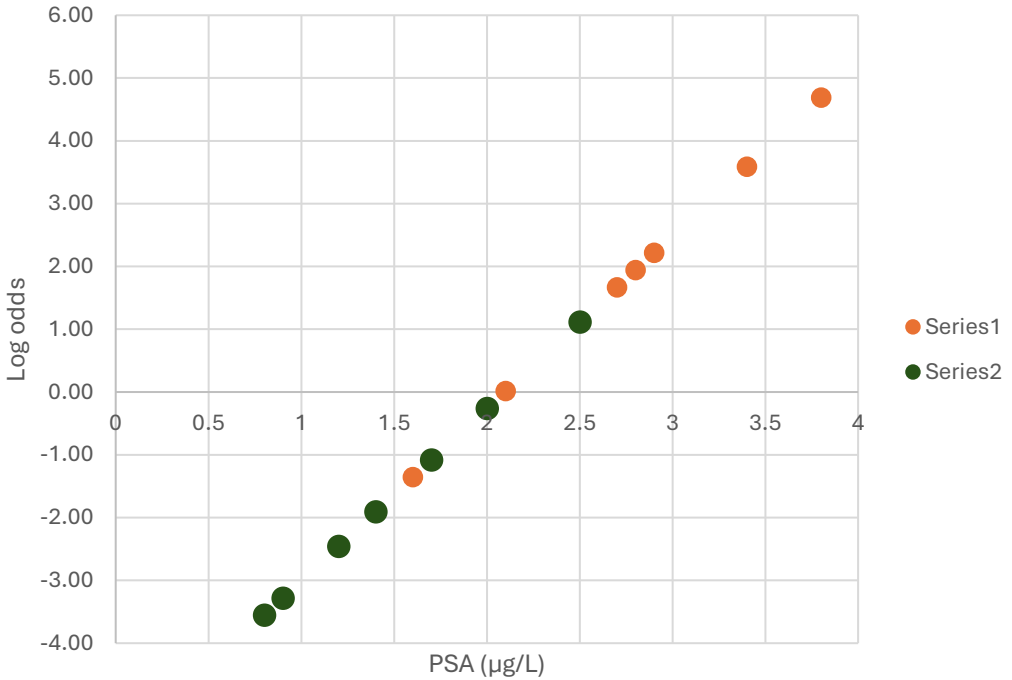
Logistic Regression Steps

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

6. Interpret the coefficients in terms of odds

$$\text{logged odds} = \ln(odds)$$

$$\ln(odds) = -5.75 + 2.75PSA$$



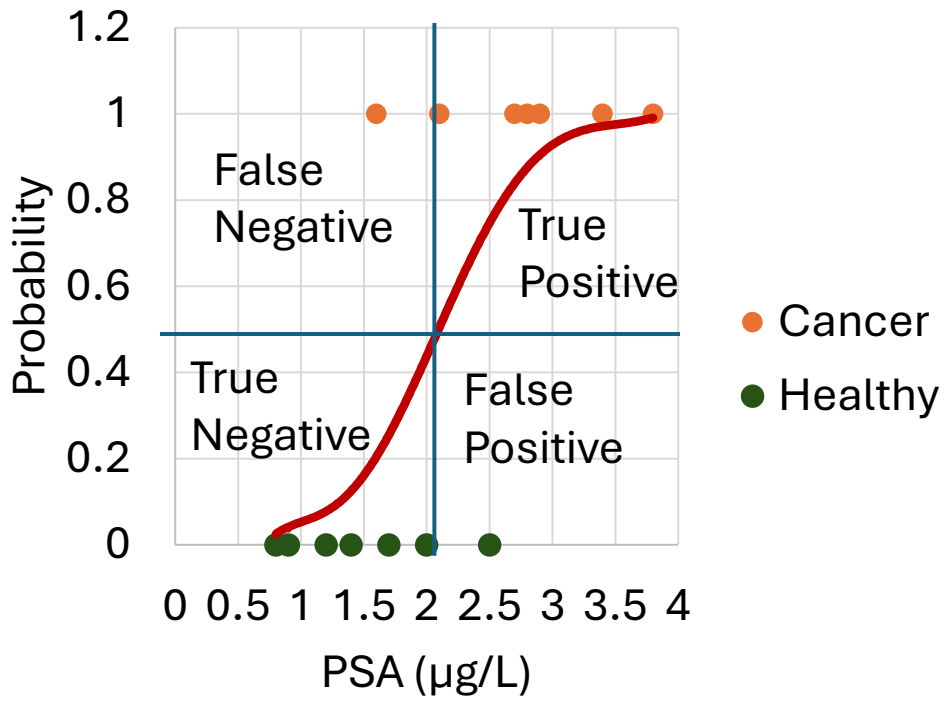
PSA	Prostate Cancer	y_i	P(x)	$\frac{p}{1-p}$	ln(odds)
3.8	cancer	1	0.99	108.27	???
3.4	cancer	1	0.97	36.08	???
2.9	cancer	1	0.90	9.14	???
2.8	cancer	1	0.87	6.94	???
2.7	cancer	1	0.84	5.27	???
2.1	cancer	1	0.50	1.01	???
1.6	cancer	1	0.20	0.26	???
2.5	healthy	0	0.75	3.04	???
2	healthy	0	0.44	0.77	???
1.7	healthy	0	0.25	0.34	???
1.4	healthy	0	0.13	0.15	???
1.2	healthy	0	0.08	0.09	???
0.9	healthy	0	0.04	0.04	???
0.8	healthy	0	0.03	0.03	???

Logistic Regression Steps

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

7. Plot the logistic regression and classify

Parameters	Estimate
b_0	-5.754
b_1	2.747



PSA	Prostate Cancer	P(x)	y(i)	$p(x_i)^{y_i}(1-p(x_i))^{(1-y_i)}$	Prediction
3.8	cancer	0.99	1	0.99	cancer
3.4	cancer	0.97	1	0.97	cancer
2.9	cancer	0.90	1	0.90	cancer
2.8	cancer	0.87	1	0.87	cancer
2.7	cancer	0.84	1	0.84	cancer
2.1	cancer	0.50	1	0.50	cancer
1.6	cancer	0.20	1	0.20	healthy
2.5	healthy	0.75	0	0.25	cancer
2	healthy	0.44	0	0.56	healthy
1.7	healthy	0.25	0	0.75	healthy
1.4	healthy	0.13	0	0.87	healthy
1.2	healthy	0.08	0	0.92	healthy
0.9	healthy	0.04	0	0.96	healthy
0.8	healthy	0.03	0	0.97	healthy

Benefits of using Logistic Regression

- Simple & Interpretable – Easy to understand and explain.
- Great for Binary Classification – Works well for yes/no problems.
- Computationally Efficient – Faster than complex models.
- Provides Probabilities – Outputs the likelihood of an event.
- Works with Small Datasets – Performs well even with limited data.
- Less Overfitting – Stable when features are properly handled.

Challenges of using Logistic Regression

- Assumes a Linear Relationship – Can’t capture complex patterns.
- Sensitive to Outliers – Extreme values can affect accuracy.
- Struggles with Multicollinearity – Highly correlated inputs cause issues.
- Needs Balanced Data – Performs poorly with imbalanced classes.
- Limited to Two Classes (Basic form) – Extensions needed for multi-class problems.

Applications of using Logistic Regression

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

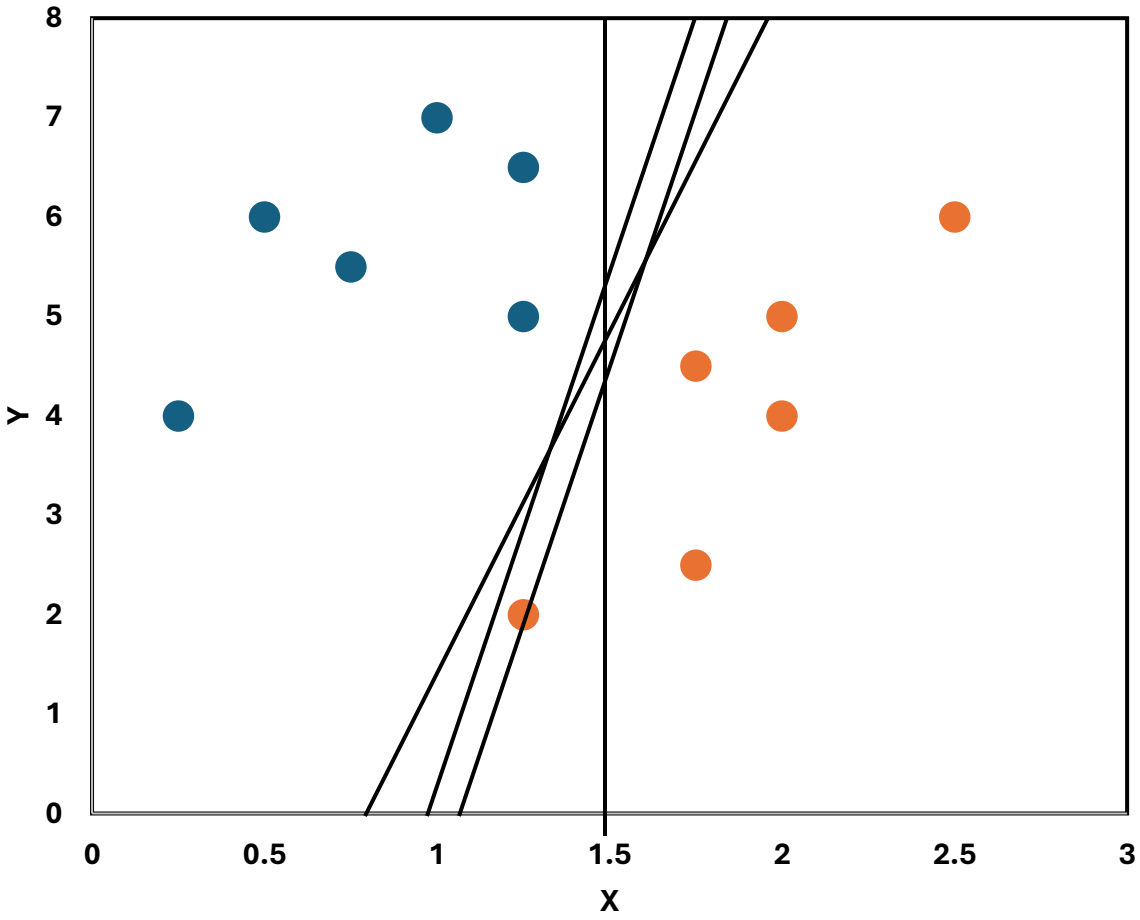
- **Healthcare** – Disease prediction (e.g., cancer detection, diabetes risk).
- **Finance** – Credit scoring, fraud detection.
- **Marketing** – Customer churn prediction, ad click prediction.
- **E-commerce** – Product recommendation, purchase likelihood.
- **HR & Recruitment** – Employee attrition prediction, hiring decisions.
- **Manufacturing** – Predicting machine failure, quality control.
- **Cybersecurity** – Spam detection, malware classification.

Contents

- Linear Models for Classification
  - Logistic Regression
  - Linear SVM
- Kernal SVM
- Naïve Bayes

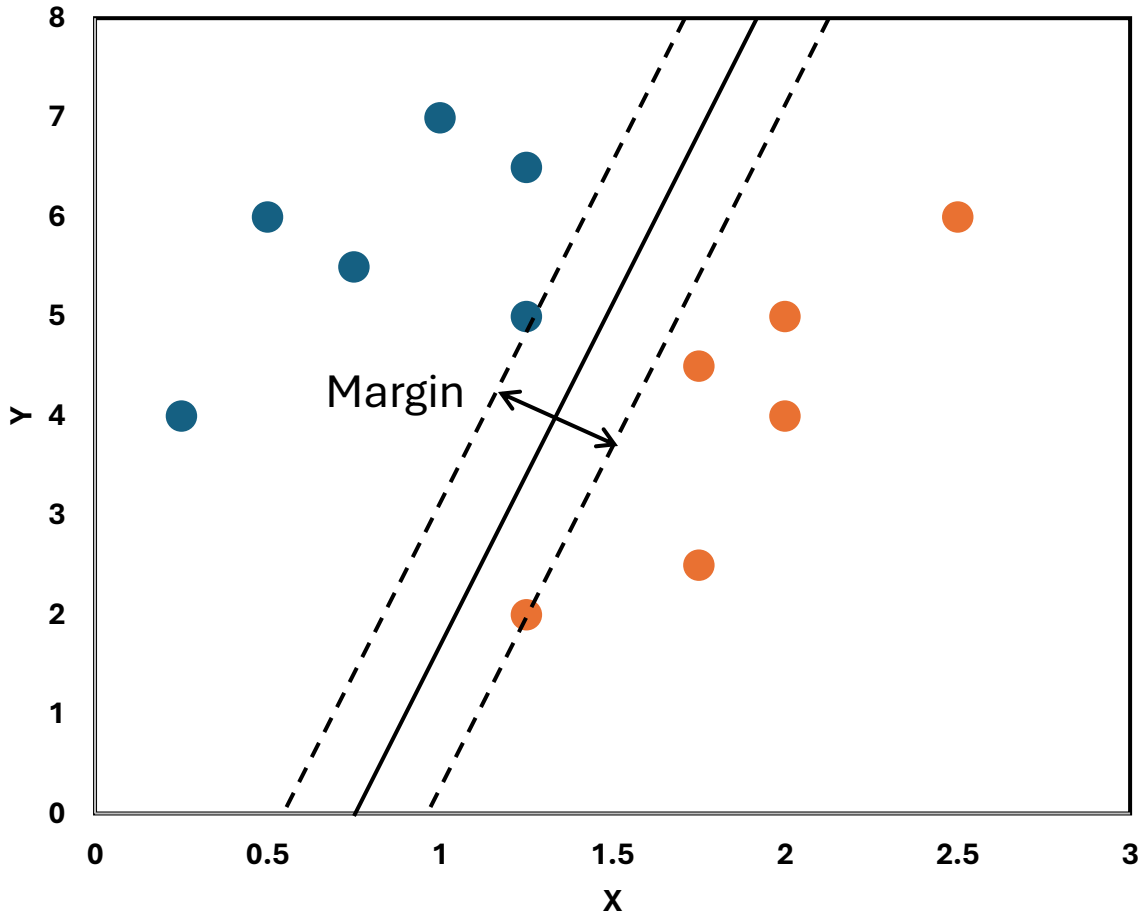
What is SVM

- SVMs (Support Vector Machines) are more commonly used in classification problems
- SVMs are based on the idea of finding a hyperplane that best divides a dataset into two classes, as shown in the image.



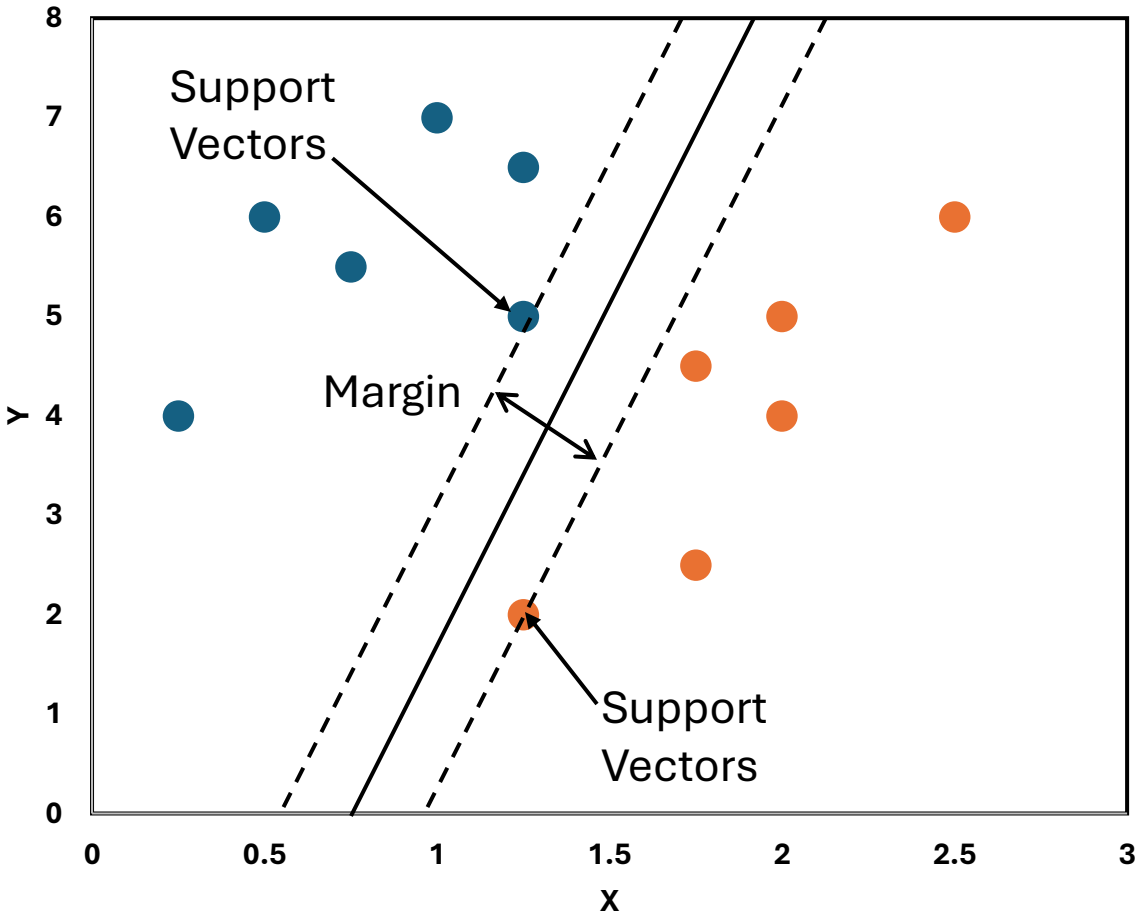
What is SVM

- To separate the two classes of data points, there are many possible hyperplanes that could be chosen.
- SVM objective is to find a plane that has the maximum margin, i.e the maximum distance between data points of both classes.
- Maximizing the margin distance provides some reinforcement so that future data points can be classified with more confidence.



Support Vectors

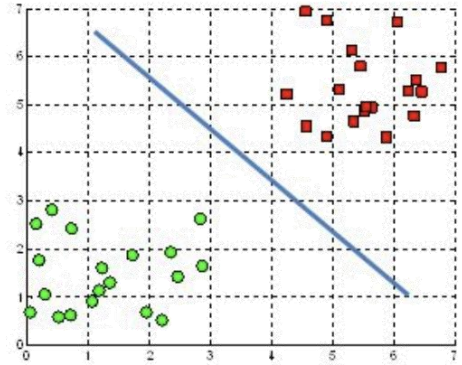
- Support vectors are the data points that are nearest to the hyper-plane and affect the position and orientation of the hyper-plane.
- We have to select a hyperplane, for which the margin, i.e the distance between support vectors and hyper-plane is maximum.
- Even a little interference in the position of these support vectors can change the hyper-plane.



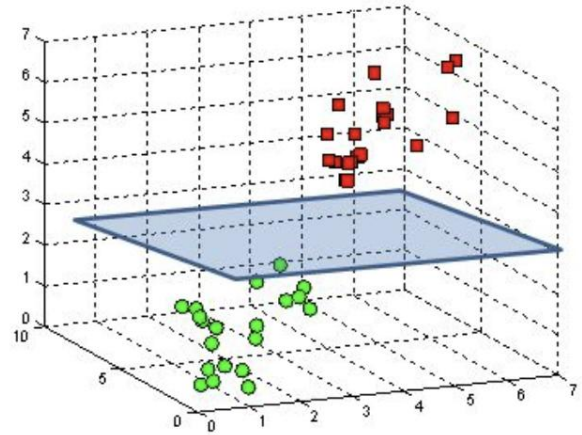
Hyperplanes

- A hyperplane is a decision boundary that differentiates the two classes in SVM.
- A data point falling on either side of the hyperplane can be attributed to different classes.
- The dimension of the hyperplane depends on the number of input features in the dataset.
- If we have 2 input features the hyper-plane will be a line. likewise, if the number of features is 3, it will become a two-dimensional plane.

A hyperplane in $\mathbb{R}^2$ is a line

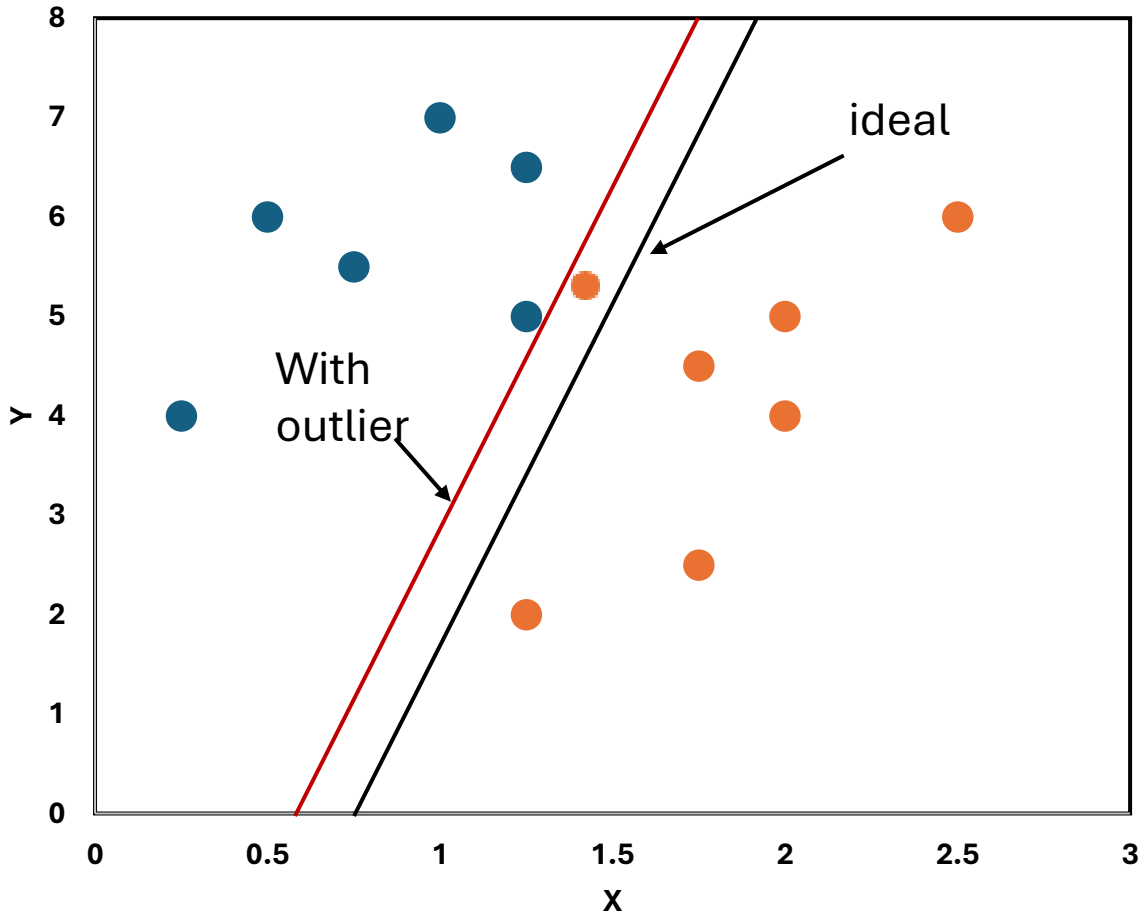


A hyperplane in $\mathbb{R}^3$ is a plane



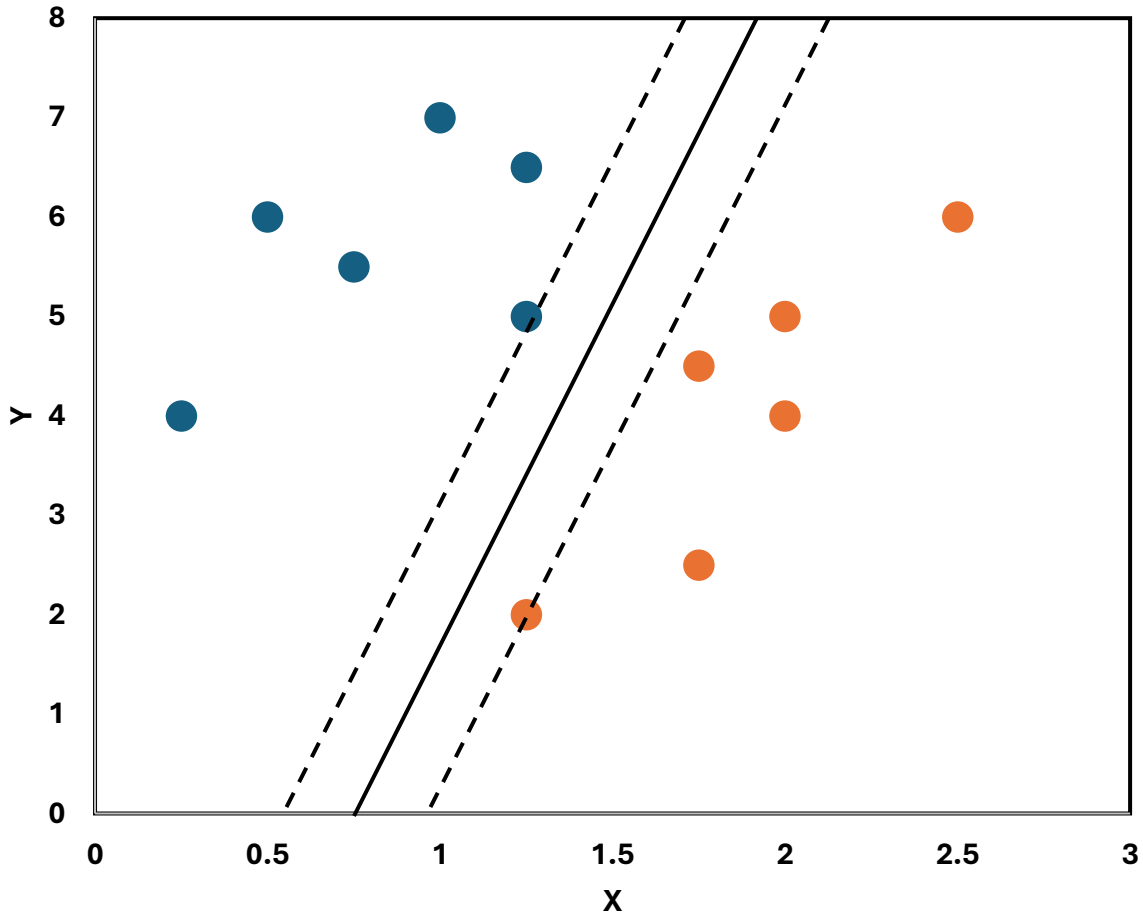
Outliers in SVM

Since SVM tries to maximize the margin while keeping all points correctly classified, **outliers** can significantly affect the placement of the decision boundary.



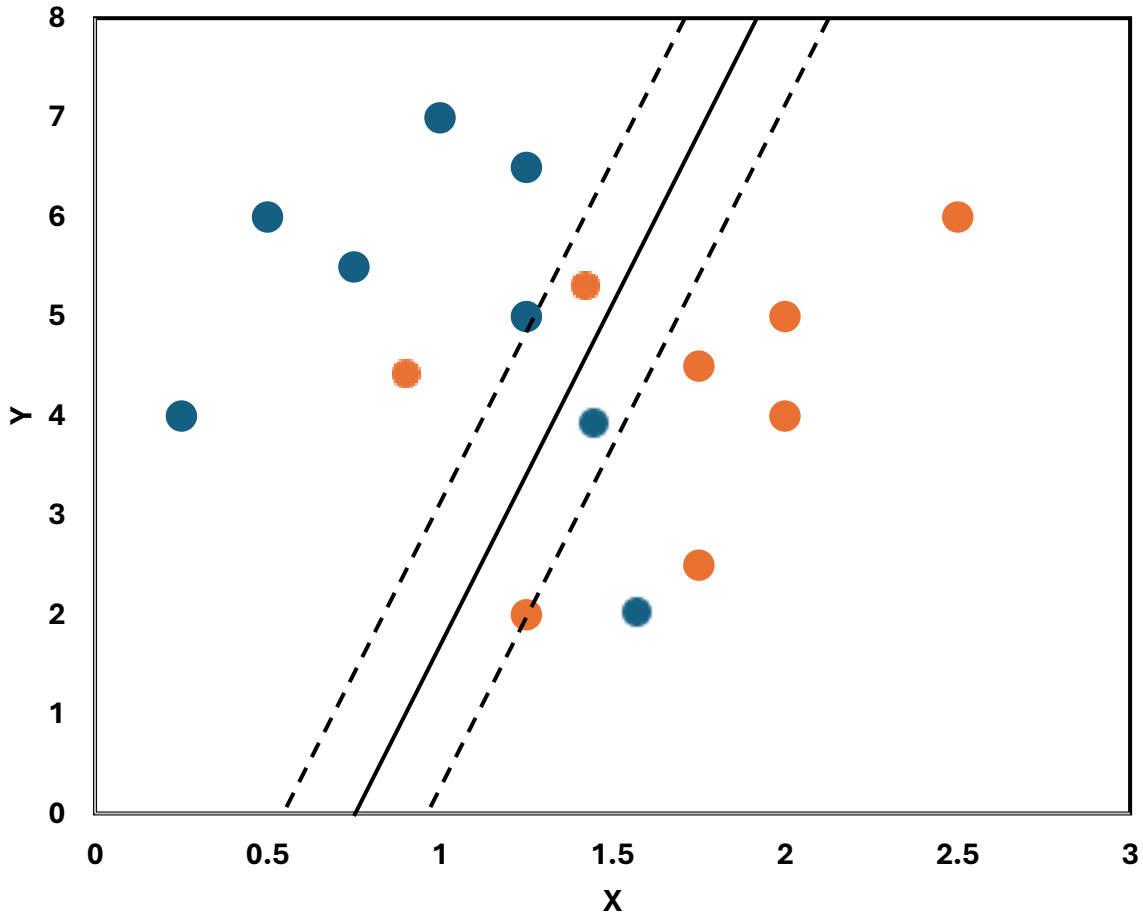
Case 1 : Hard Margin

If SVM is trained without allowing misclassification (i.e., a hard margin), outliers force the hyperplane to shift, reducing generalization.



Case 2 : Soft Margin

Soft margin SVM introduces slack variables to allow some misclassification and make the model more **robust to outliers**.



Mathematical background

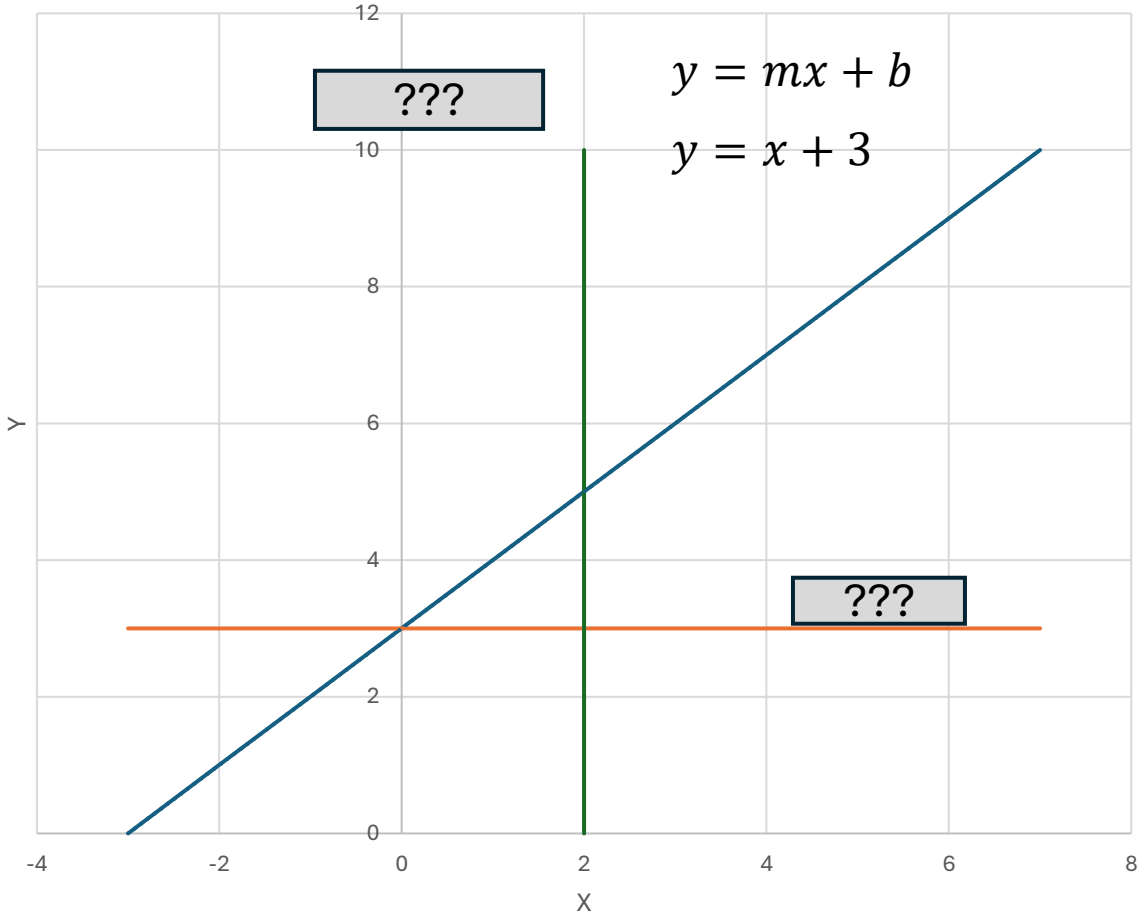
- Need for General Form

$$Ax + By + C = 0$$

$$-x + y - 3 = 0 \quad \times -1$$

$$x - y + 3 = 0 \quad \times -2$$

$$2x - 2y + 6 = 0$$

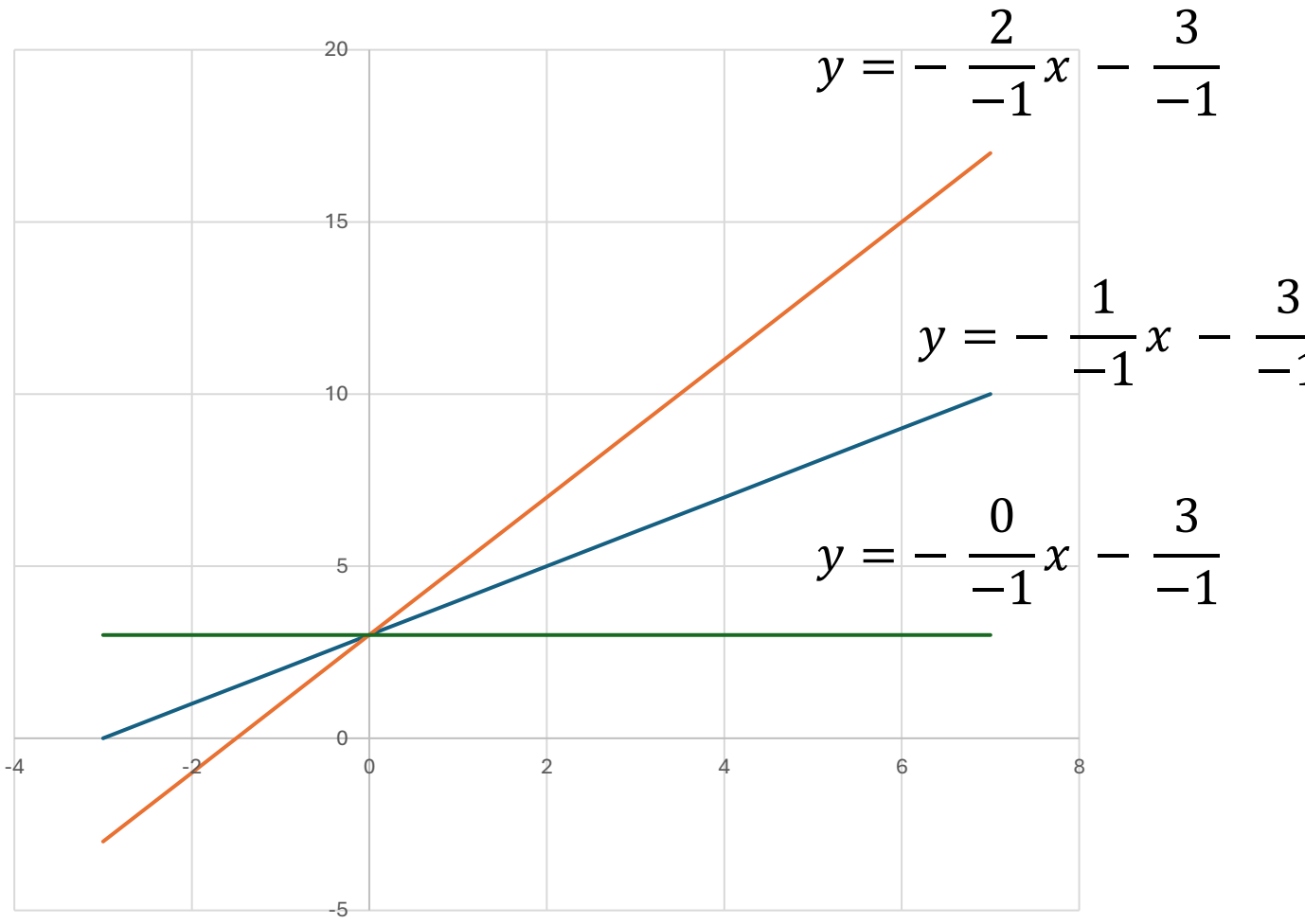


Mathematical background

- Changing A will change the slope around the intercept

$$Ax + By + C = 0$$

$$y = -\frac{A}{B}x - \frac{C}{B}$$



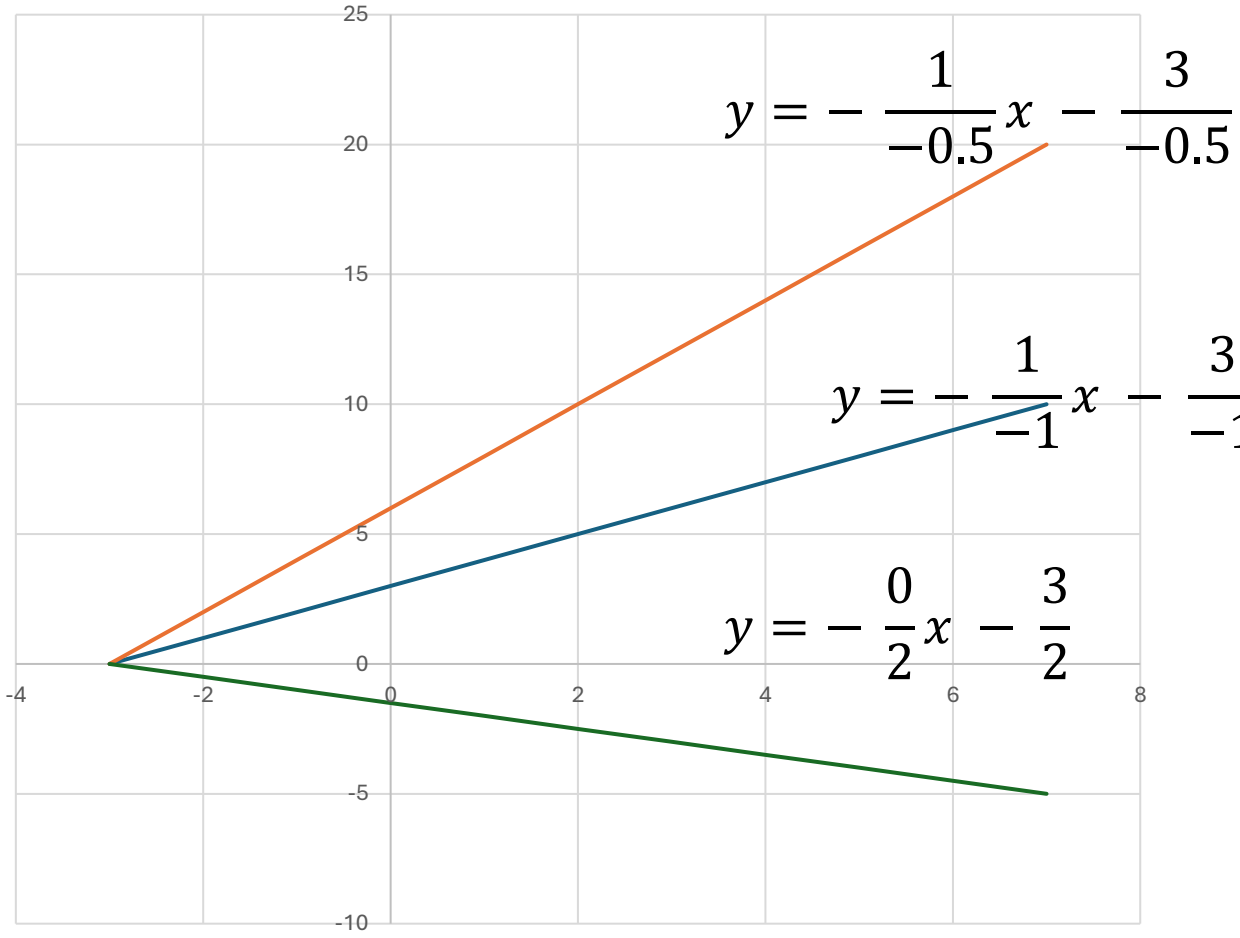
Mathematical background

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

- Changing B will change the slope but with different intercept

$$Ax + By + C = 0$$

$$y = -\frac{A}{B}x - \frac{C}{B}$$



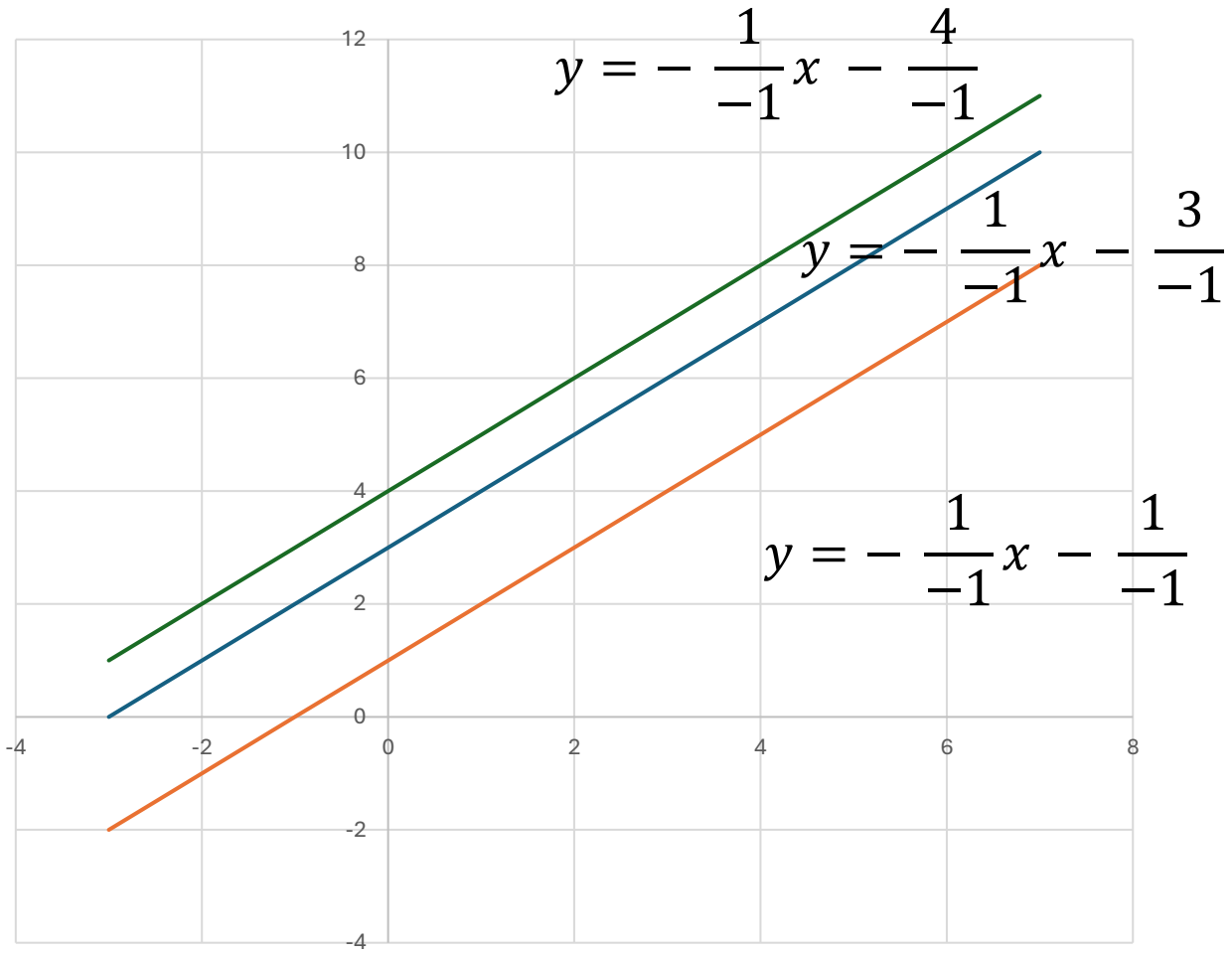
Mathematical background

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

- Changing C will create a parallel line

$$Ax + By + C = 0$$

$$y = -\frac{A}{B}x - \frac{C}{B}$$



Mathematical background

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

- For point X_1 (4,10)

$$Ax + By + C = 0$$

$$-x + y - 3 = 0$$

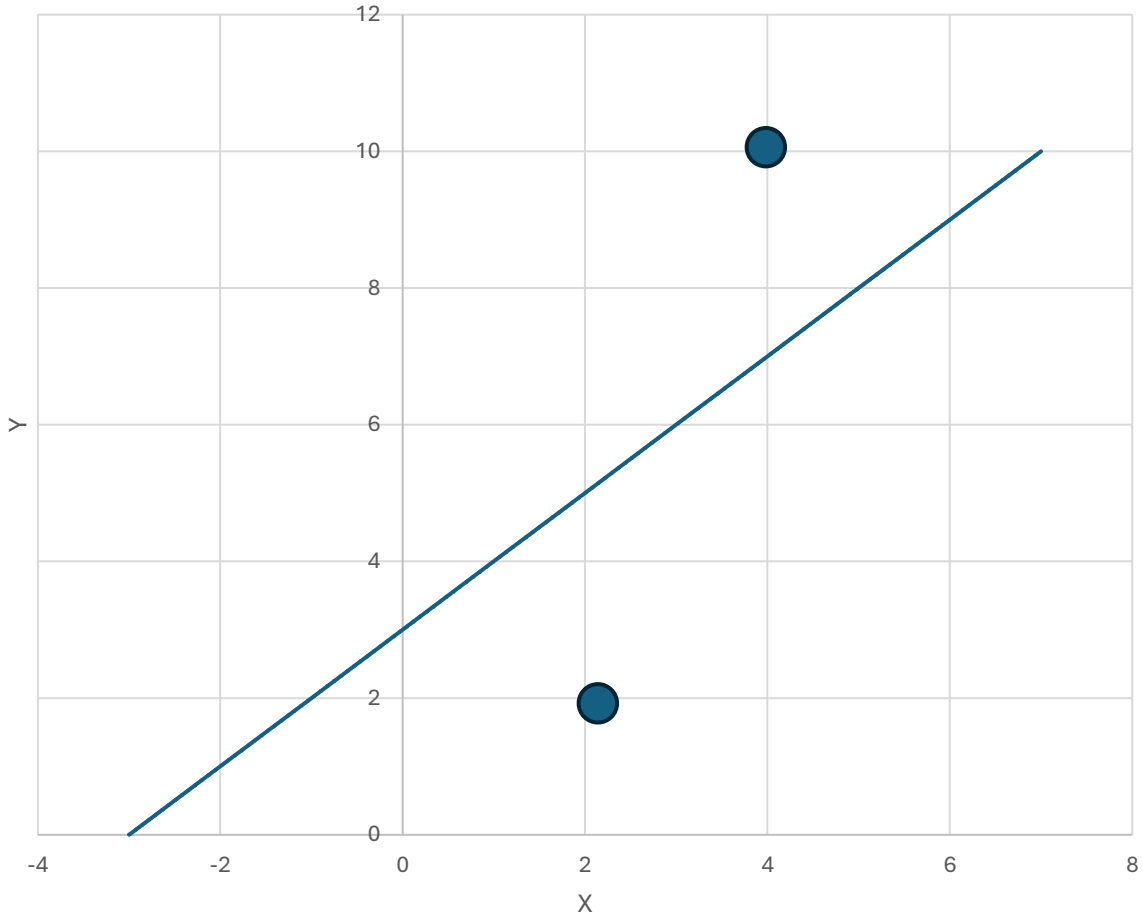
$$-4 + 10 - 3 = \boxed{???}$$

If positive than its above the line

- For point X_1 (2,2)

$$-2 + 2 - 3 = \boxed{???}$$

If negative than its below the line



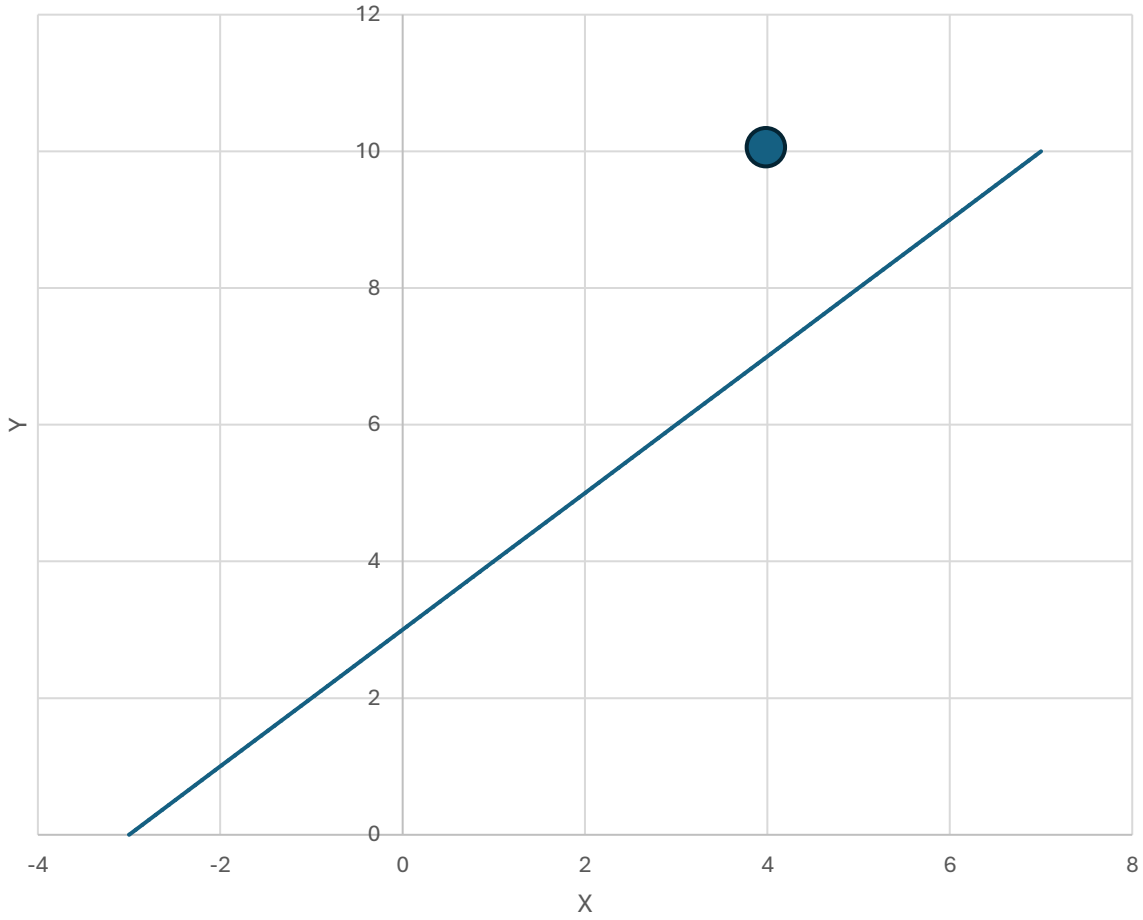
Any data on the lie is equal to?? ???

Mathematical background

- Distance between point X_1 (4,10) and line

$$d = \frac{|Ax_0 + By_o + C|}{\sqrt{A^2 + B^2}}$$

$d =$???

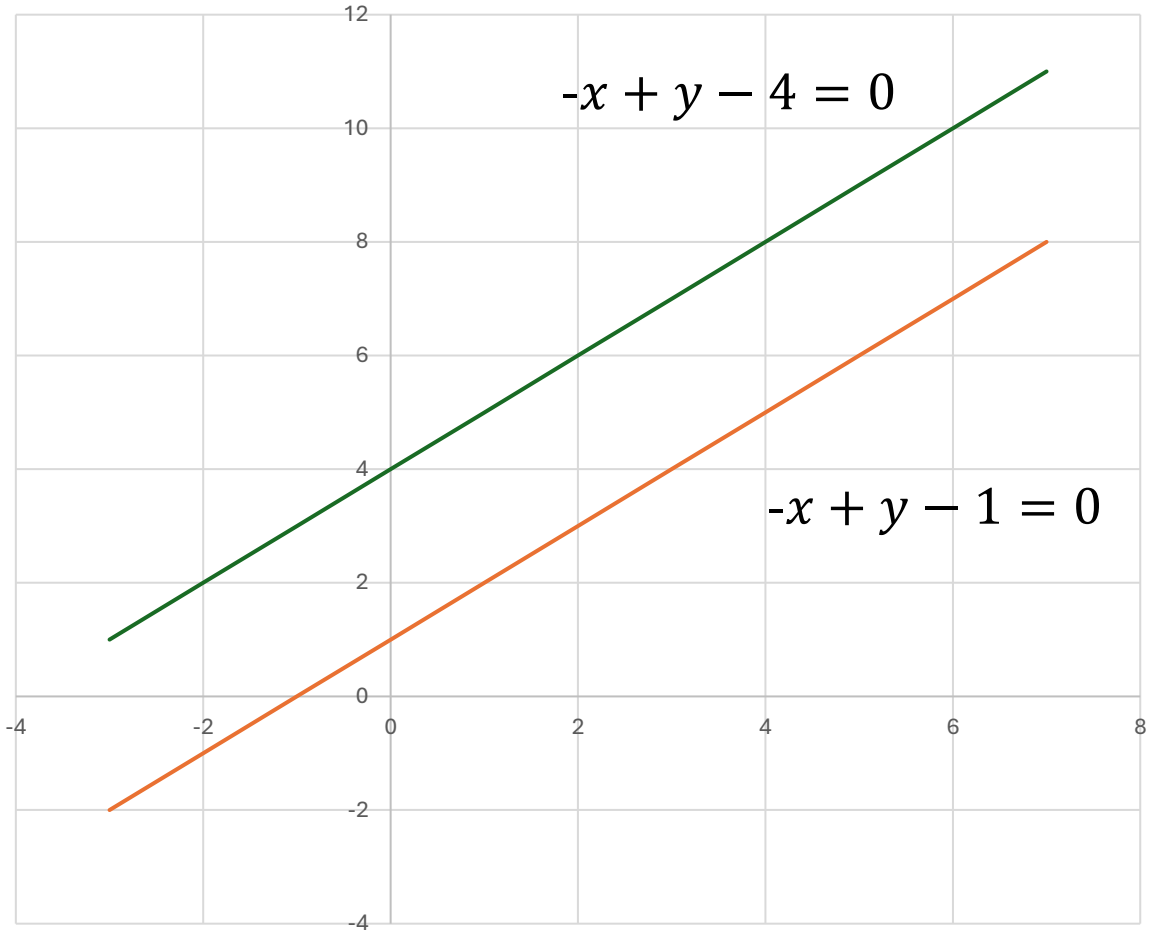


Mathematical background

- Distance between two parallel lines

$$d = \frac{|C_2 - C_1|}{\sqrt{A^2 + B^2}}$$

$d =$???



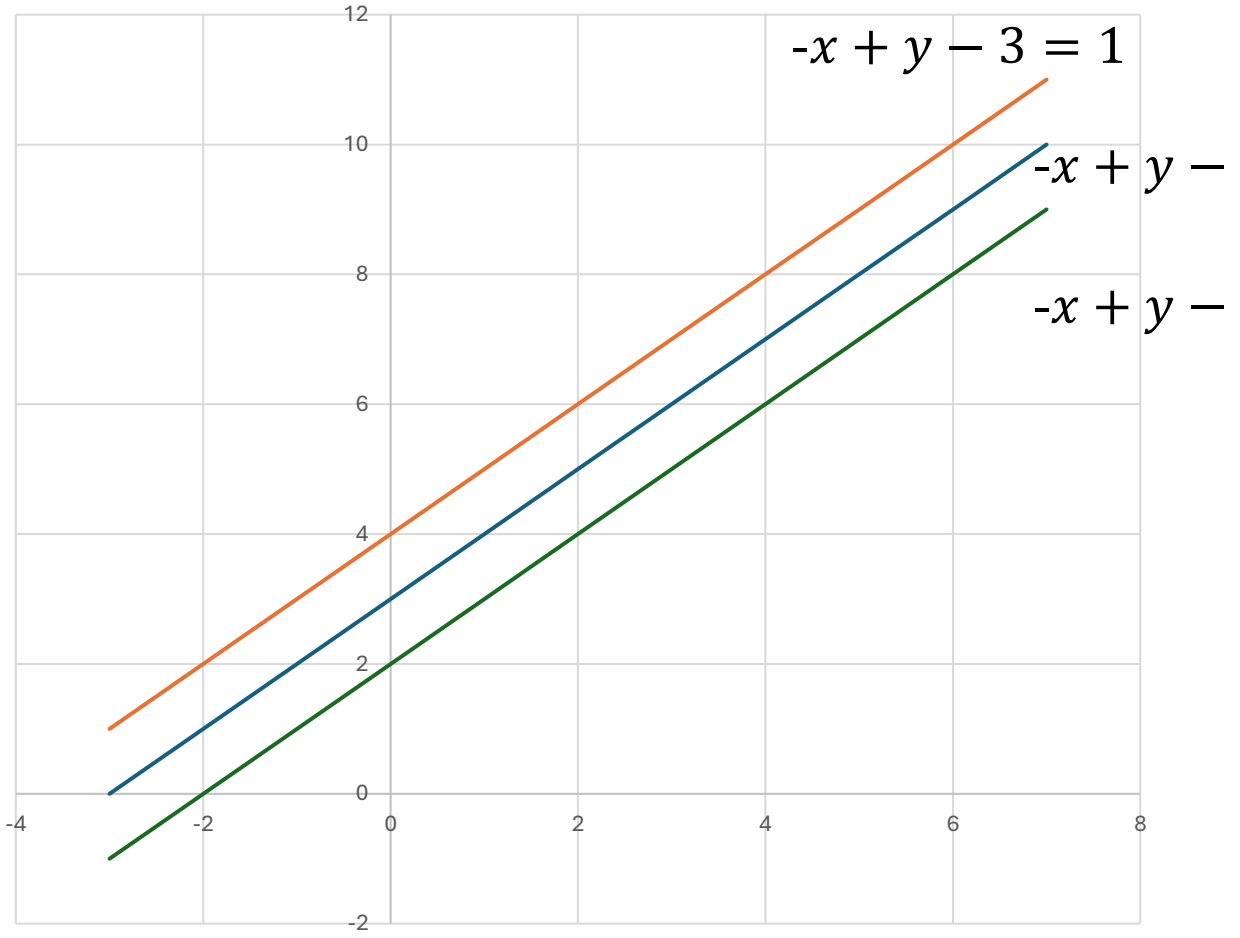
Mathematical background

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

- Distance between two parallel lines

$$d = \frac{|C_2 - C_1|}{\sqrt{A^2 + B^2}}$$

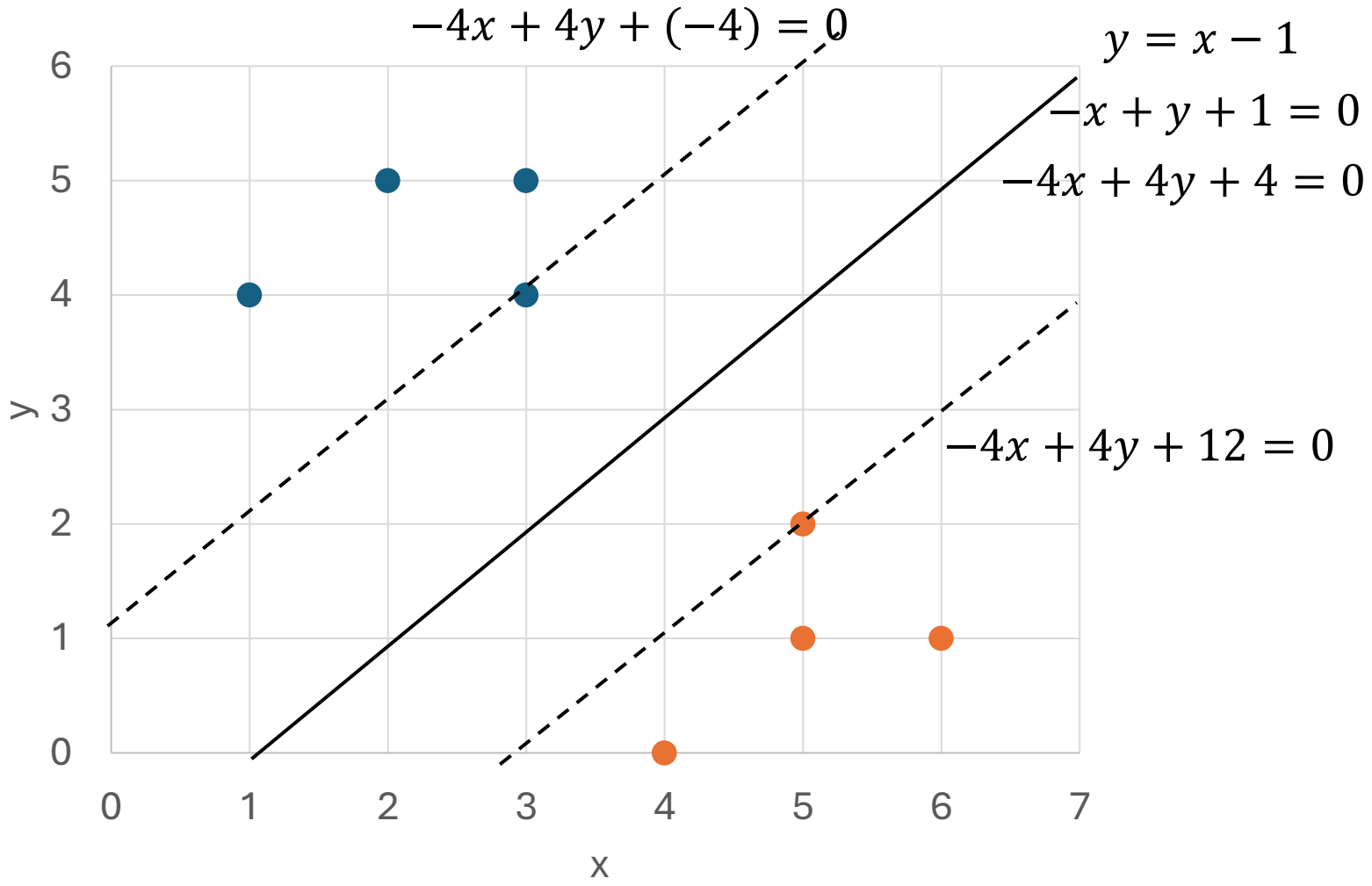
$$d = \frac{2}{\sqrt{A^2 + B^2}}$$



Hard Margin Example

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

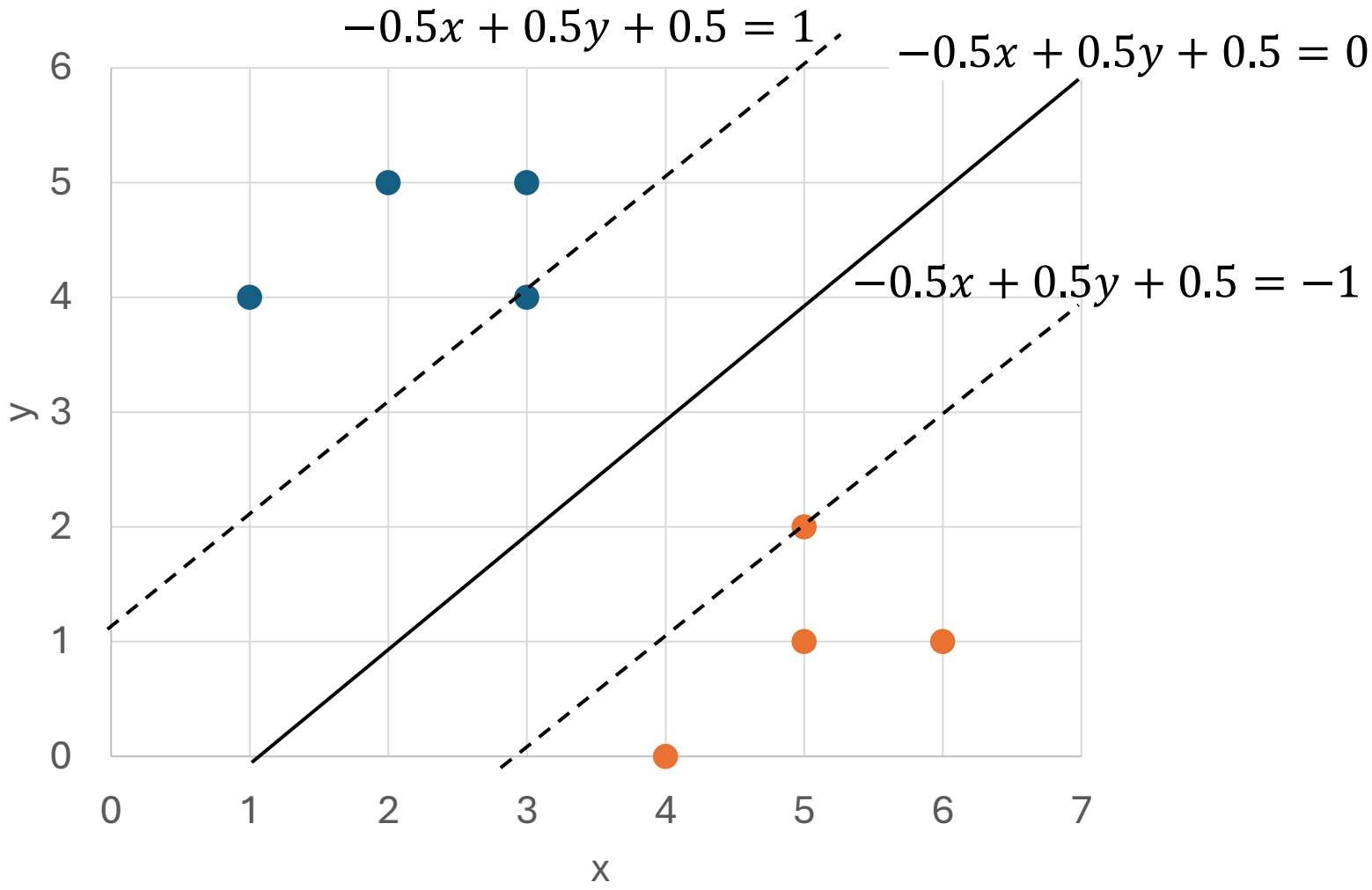
Group	x	y
A	1	4
A	2	5
A	3	5
A	3	4
B	6	1
B	4	0
B	5	2
B	5	1



Hard Margin Example

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

Re-write the equations so that the hyperplane equation is = 0 and the lower boundary is equal to -1 and the upper boundary is =1



Hard Margin Example

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

Re-write the equations so

$$w^T x + b = 0$$

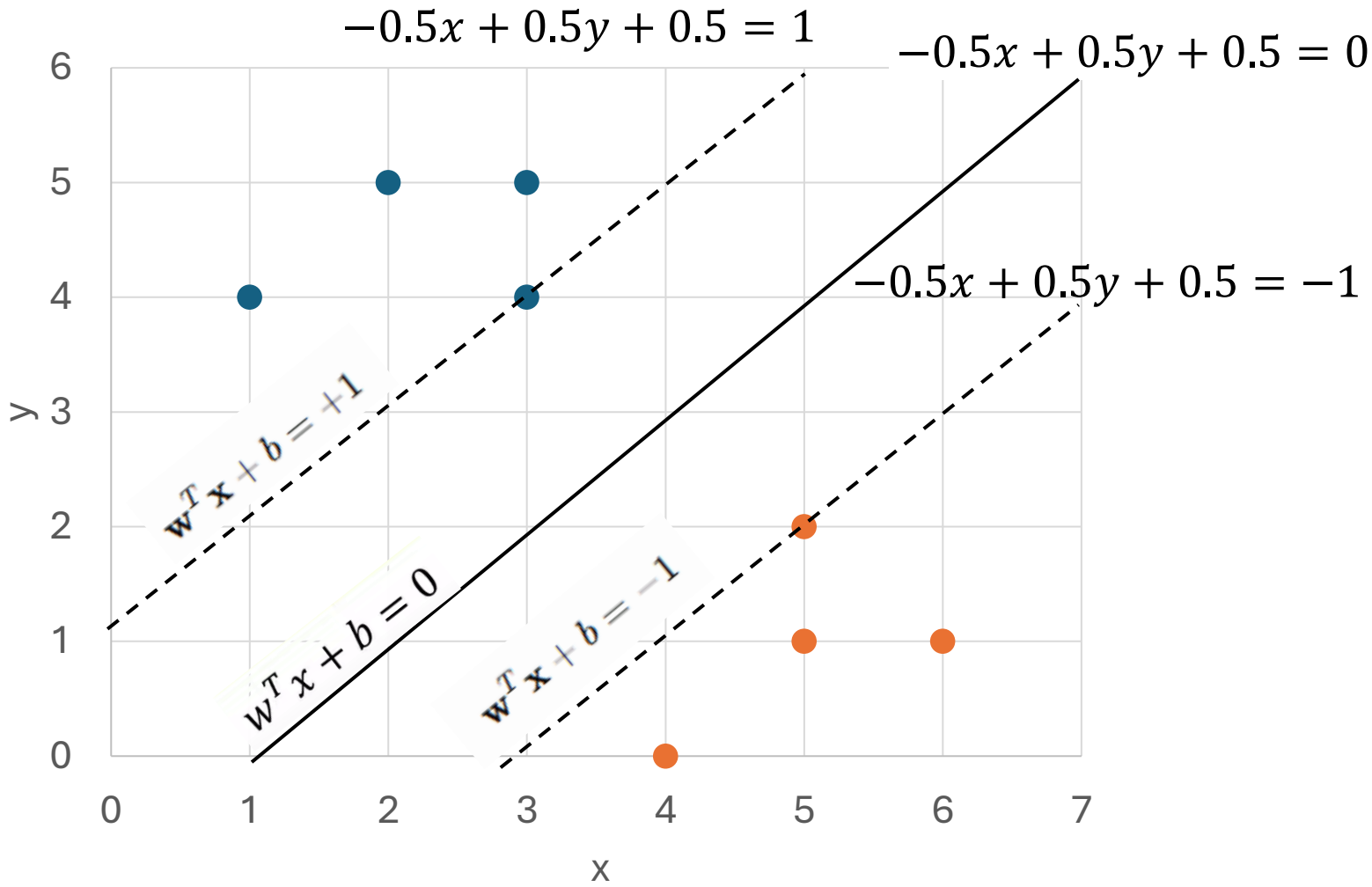
w is the **weight vector** (normal to the hyperplane).

x is a point in the feature space.

b is the **bias term** (offset of the hyperplane from the origin).

$$[-0.5 \quad 0.5] \cdot \begin{bmatrix} x \\ y \end{bmatrix} + b = 0$$

$$-0.5x + 0.5y + b = 0$$



Hard Margin Example

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

SVM Hyperplane Equation

$$w^T x + b = 0$$

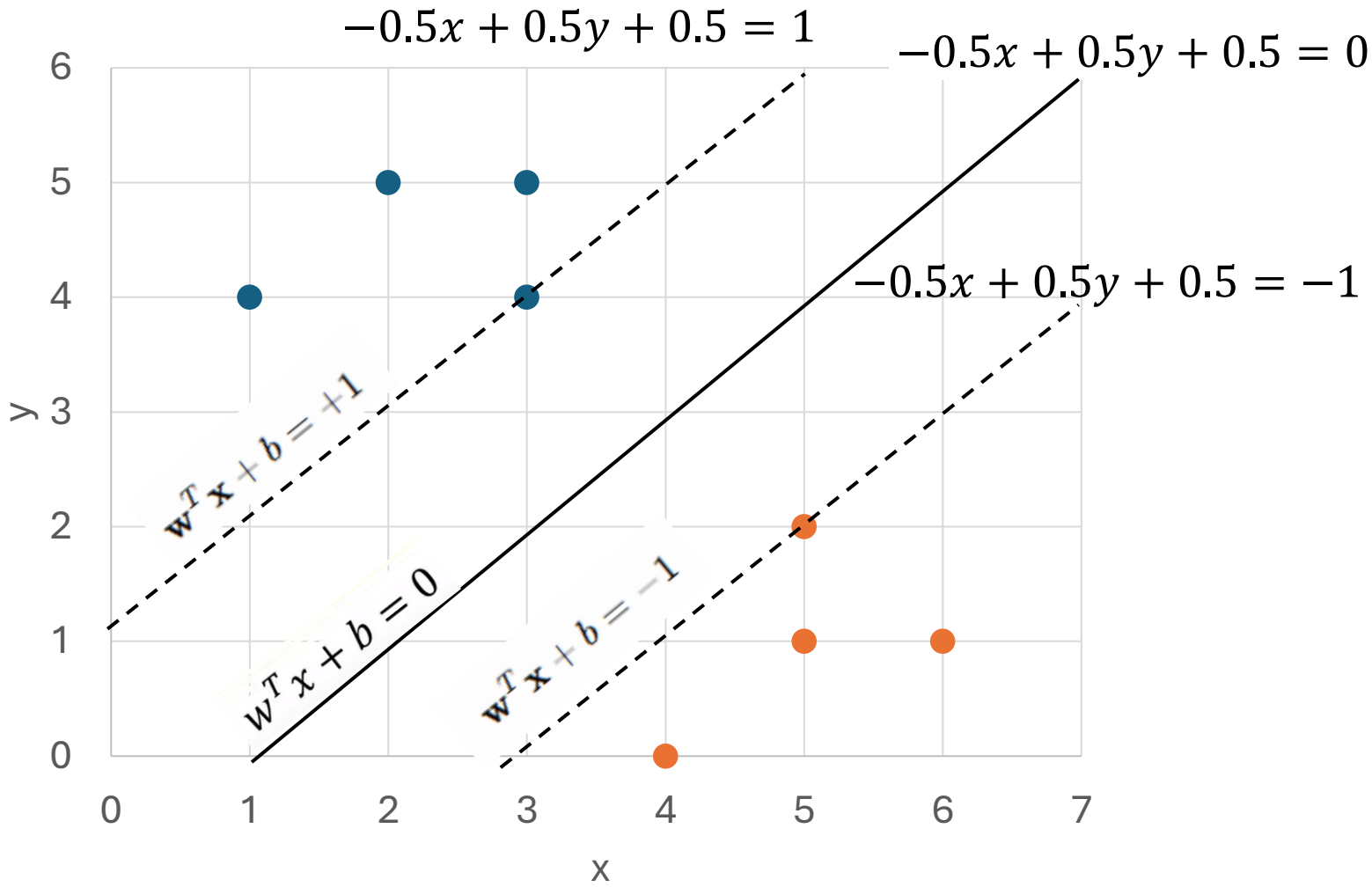
w is the **weight vector** (normal to the hyperplane).

x is a point in the feature space.

b is the **bias term** (offset of the hyperplane from the origin).

$$[-0.5 \quad 0.5] \cdot \begin{bmatrix} x \\ y \end{bmatrix} + b = 0$$

$$-0.5x + 0.5y + b = 0$$



Hard Margin Example

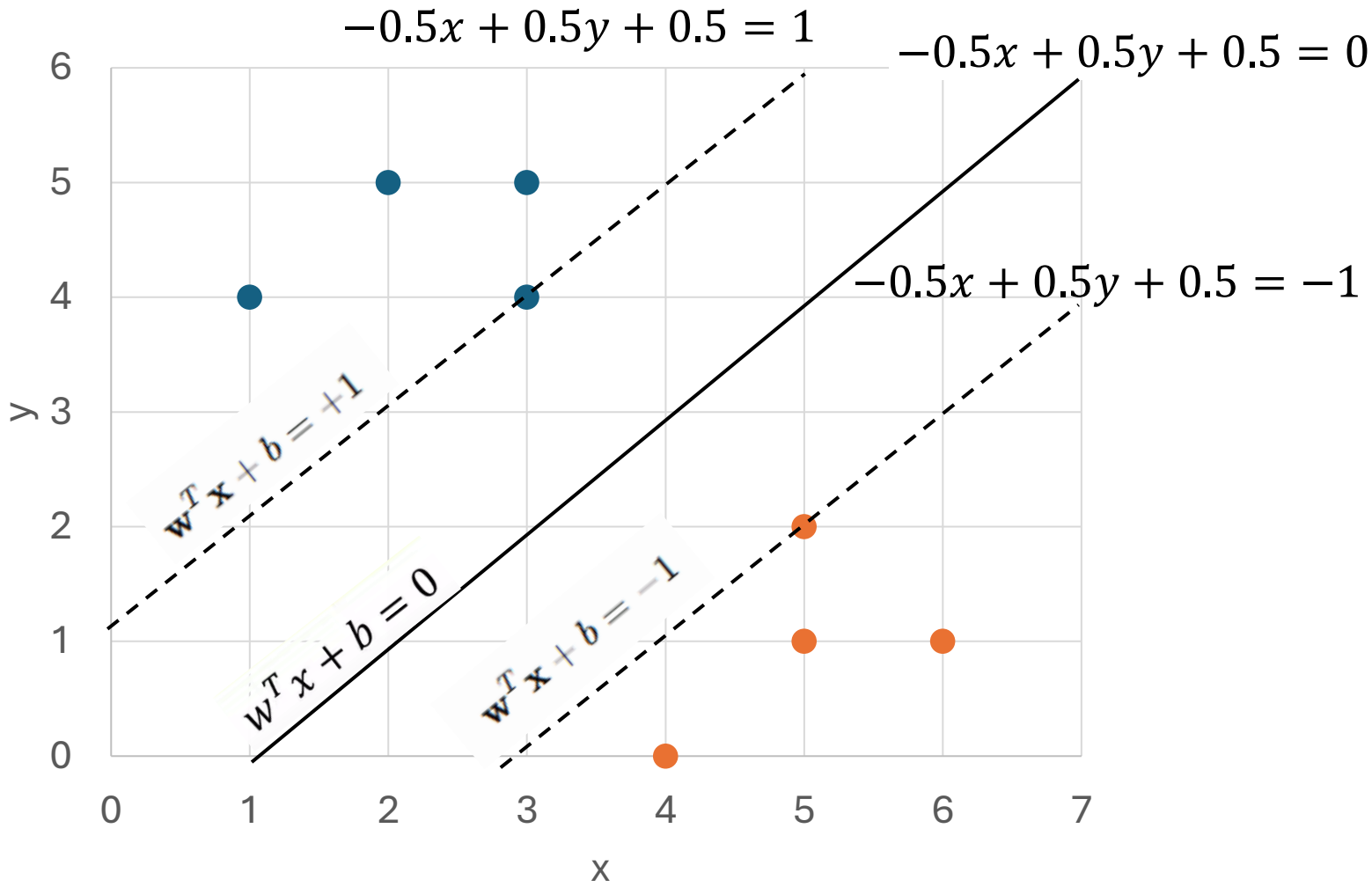
Hyperplanes for Different Classes

For positive class $(+1)$:

$$\mathbf{w}^T\mathbf{x} + b = +1$$

For negative class (-1):

$$\mathbf{w}^T\mathbf{x} + b = -1$$



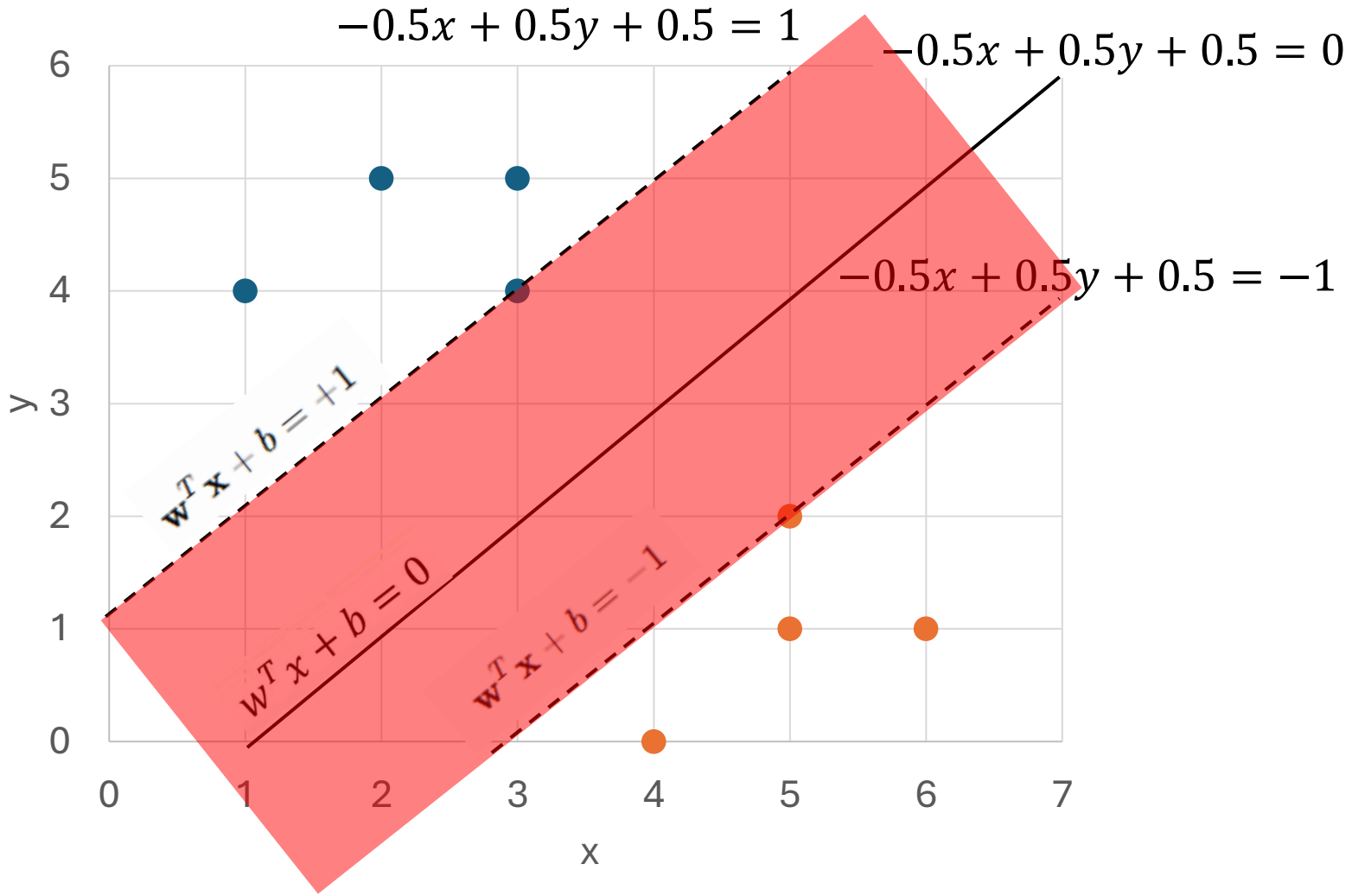
Hard Margin Example

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

The **perpendicular distance** between the two margin boundaries is given by:

$$\text{Margin} = \frac{2}{\|\mathbf{w}\|}$$

SVM tries to **maximize this margin**, which is equivalent to **minimizing ‖w‖**.



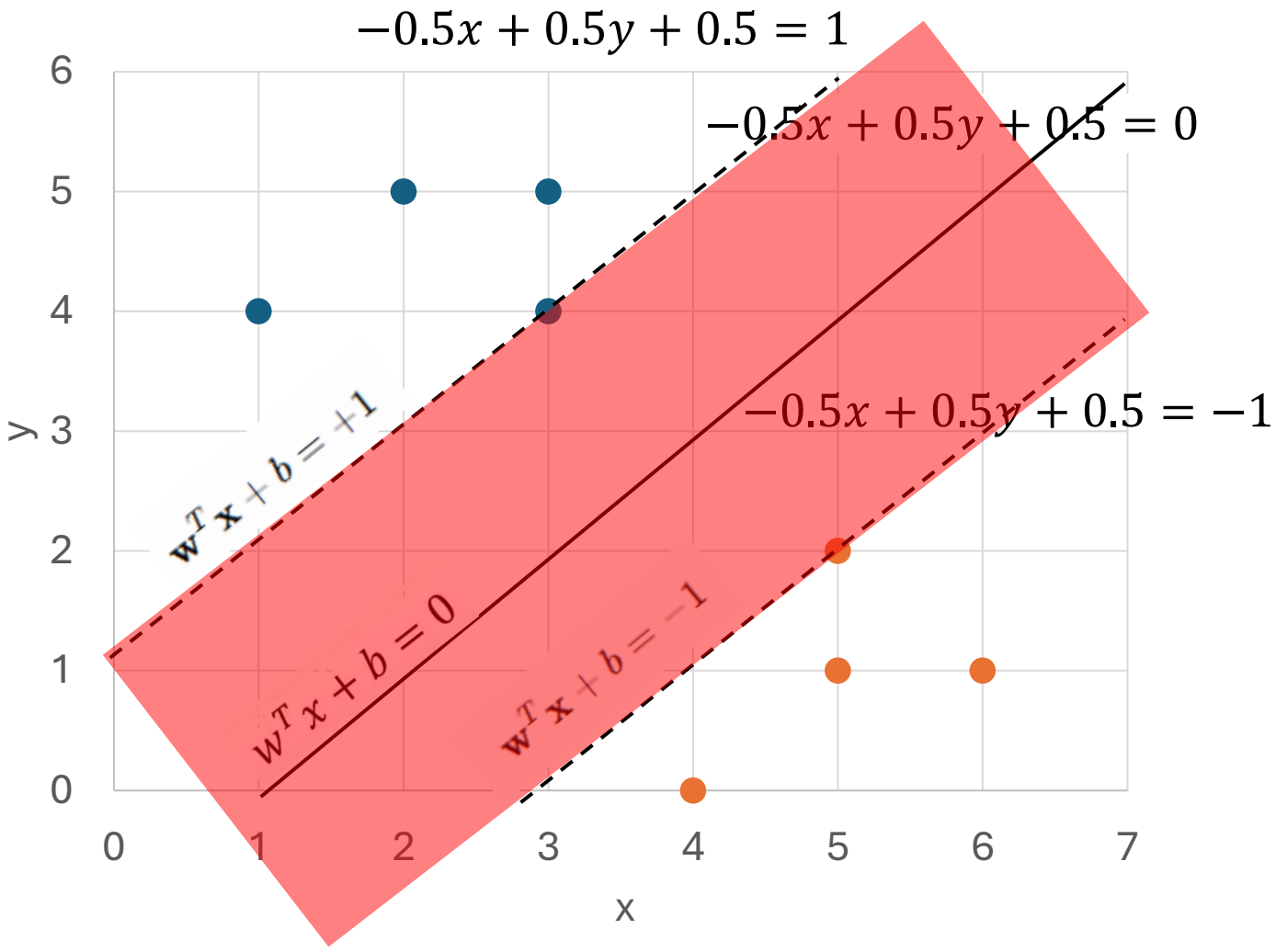
Hard Margin Example

$$Y \begin{cases} +1 & \text{if } w^T x + b \geq 0 \\ -1 & \text{if } w^T x + b < 0 \end{cases}$$

$$\max \frac{2}{\|w\|} \text{ such that } w^T x_i + b \begin{matrix} \geq 1 & \text{if } Y_i = +1 \\ \leq -1 & \text{if } Y_i = -1 \end{matrix}$$

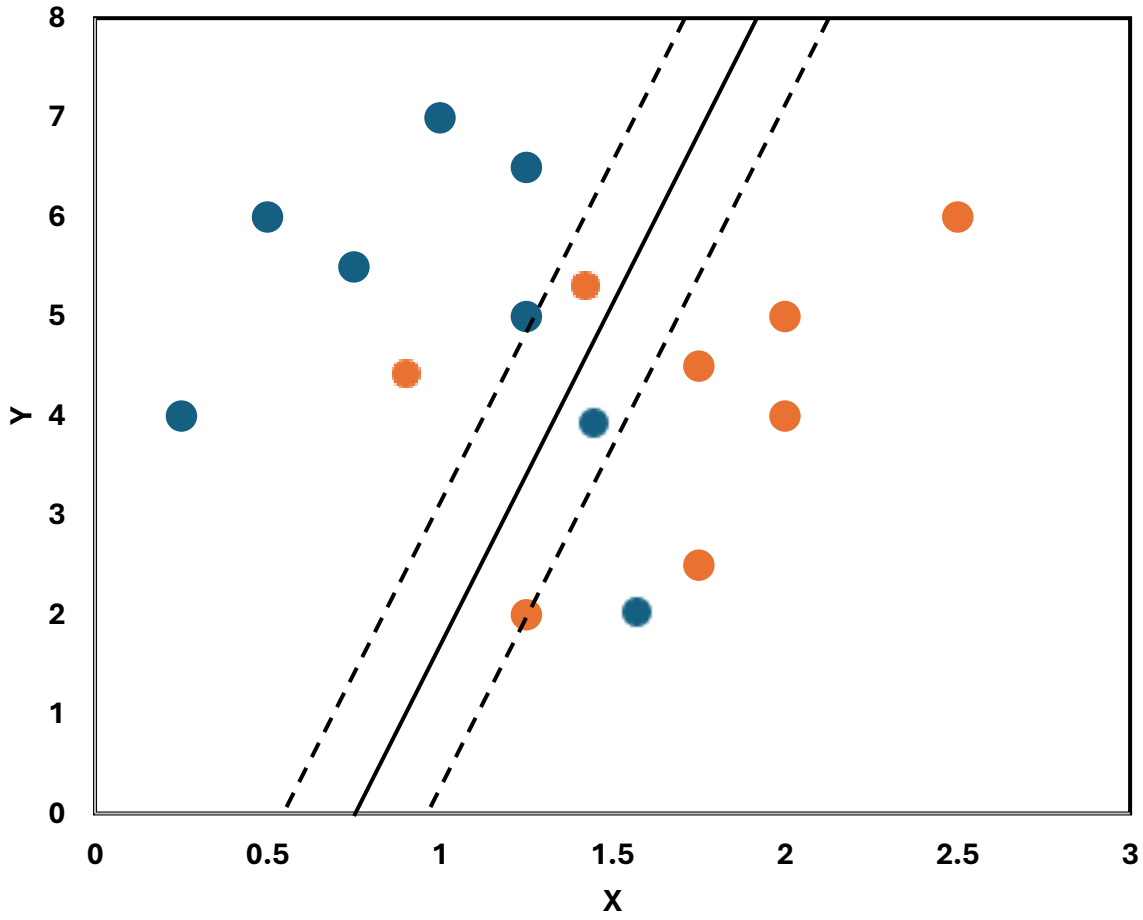
$$Y_i = [\,1, 1, 1, 1, -1, -1, -1, -1]$$

$$\max \frac{2}{\|w\|} \text{ such that } Y_i(w^T x_i + b) \geq 1$$



Soft Margin Example

- In **real-world datasets**, perfect linear separation is often **not possible** due to **overlapping classes** and **outliers**. This is where **Soft Margin SVM** comes in, allowing some misclassification by introducing **slack variables** and using **hinge loss**.



Soft Margin Example

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

- Allows some misclassifications by introducing **slack variables** ξi
- To handle **misclassified points**, we modify the margin constraints:

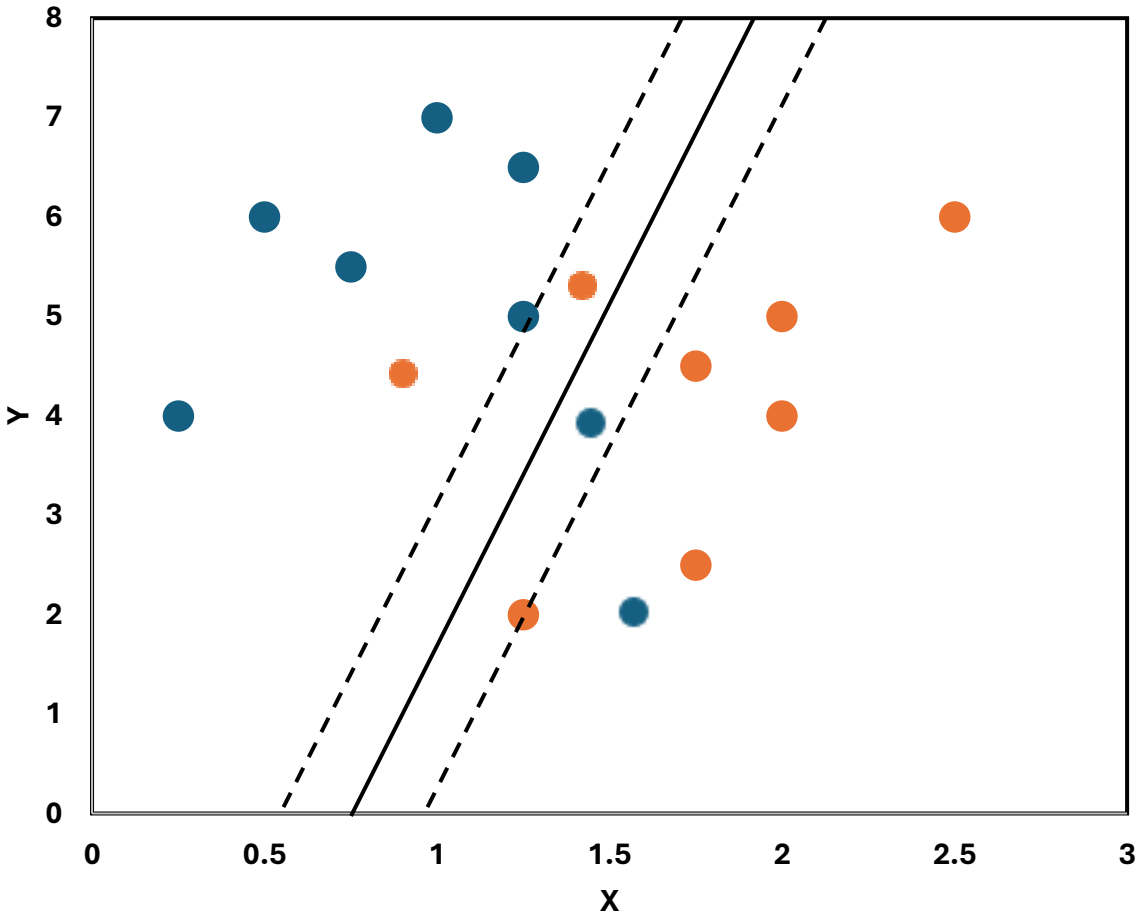
$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i, \quad \forall i$$

$\xi_i \geq 0$ is a **slack variable** for the i-th point, measuring **how much it violates** the margin.

If $\xi_i = 0$, the point is correctly classified **outside** the margin.

If $0 < \xi_i < 1$, the point is **inside** the margin but still correctly classified.

If $\xi_i \geq 1$, the point is **misclassified**.



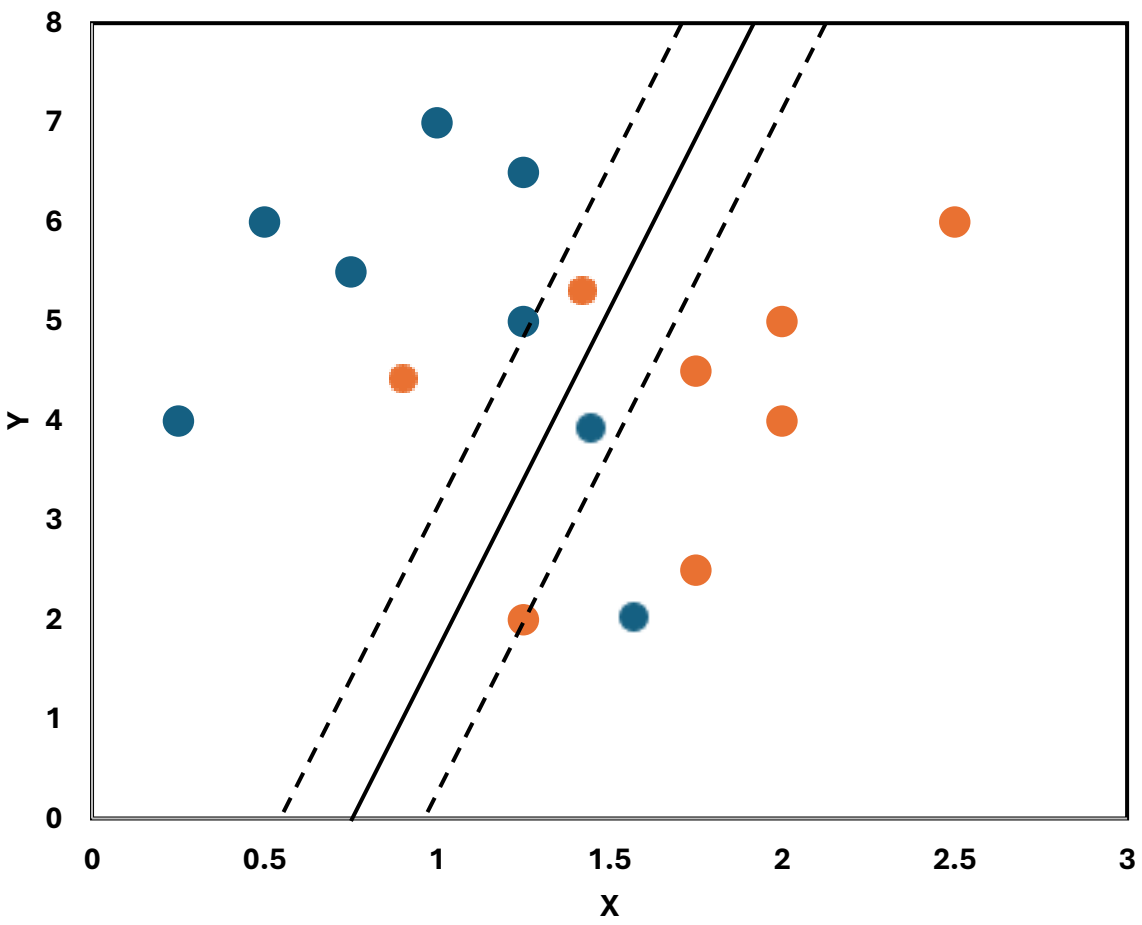
Soft Margin Example

- The **goal** is to **maximize the margin** while minimizing misclassifications. The modified objective function is:

$$\min_{\mathbf{w},b,\boldsymbol{\xi}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N}\xi_i$$

$$y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0, \quad \forall i$$

- C is a **regularization parameter** that controls the trade-off between **margin size** and **misclassification penalty**:
- **Large C** → Penalizes misclassification heavily (tends to fit data tightly).
- **Small C** → Allows more misclassifications (more flexible).



Soft Margin Example

- The **hinge loss** function helps **penalize misclassified points**:

$$L_{\text{hinge}}(y_i, f(x_i)) = \max(0, 1 - y_i(\mathbf{w}^T\mathbf{x}_i + b))$$

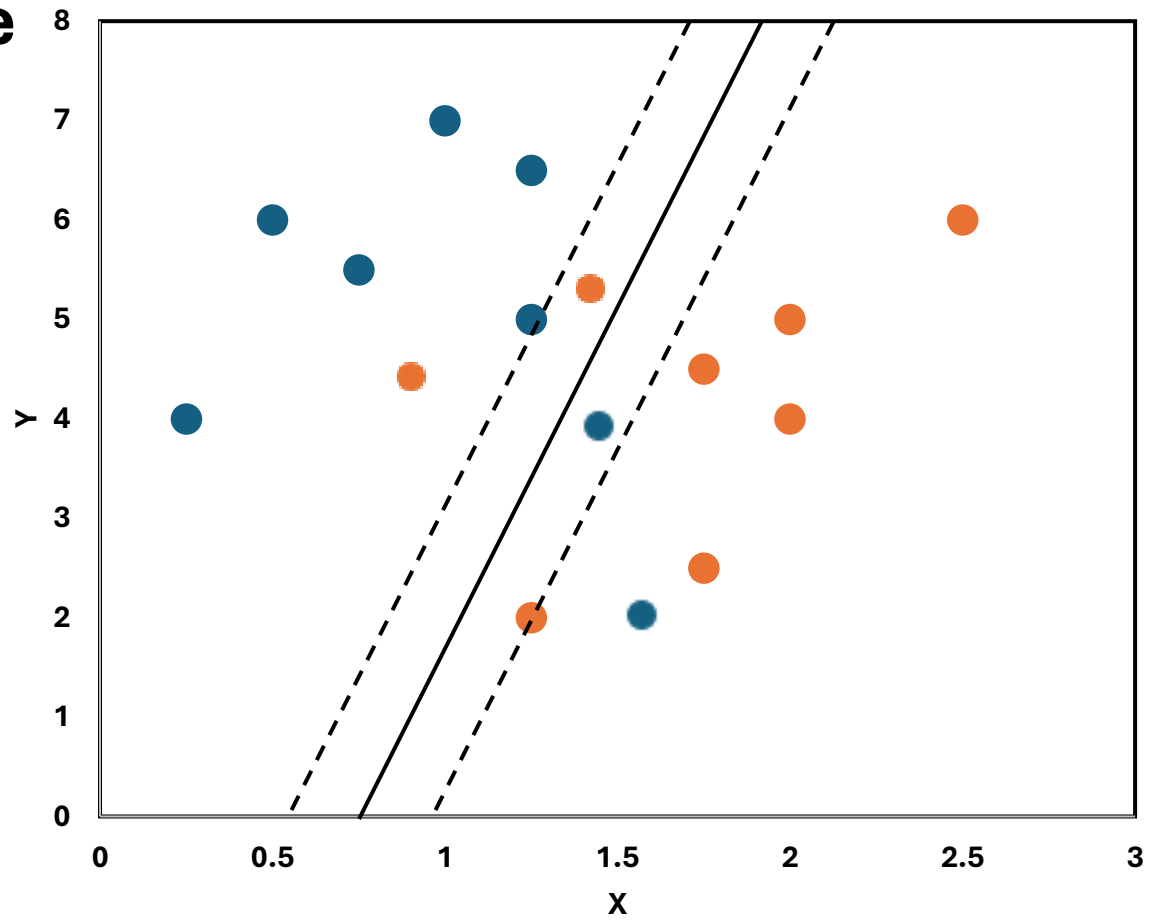
If $y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1$, the loss is **0** (correct classification).

If $y_i(\mathbf{w}^T\mathbf{x}_i + b) < 1$, there is a **penalty**.

- This results in a final optimization problem:

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{i=1}^{N} \max(0, 1 - y_i(\mathbf{w}^T\mathbf{x}_i + b))$$

- which balances **margin maximization** and **misclassification penalties**.



Non-Linear SVM

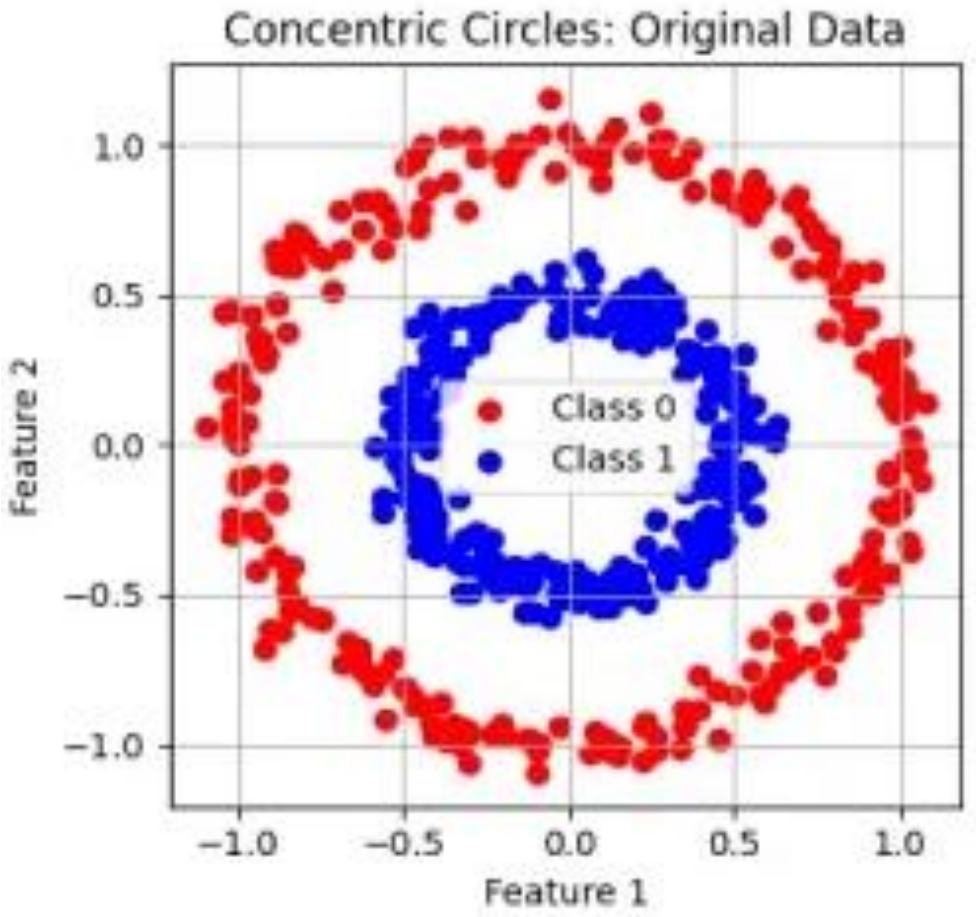
McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

- Non-Linear SVM extends SVM to handle complex, non-linearly separable data using kernels.
- Kernels enable SVM to work in higher dimensions where data can become linearly separable.
- The kernel function computes the similarity between data points allowing SVM to capture complex patterns and nonlinear relationships between features.
- This enables nonlinear SVM to form curved or circular decision boundaries with help of kernel.

Non-Linear SVM

- It's clear that we cannot classify the dataset by a linear decision boundary, but this data can be converted into a linear one using higher dimensions.
- Let's create one more dimension and name it as **z.**

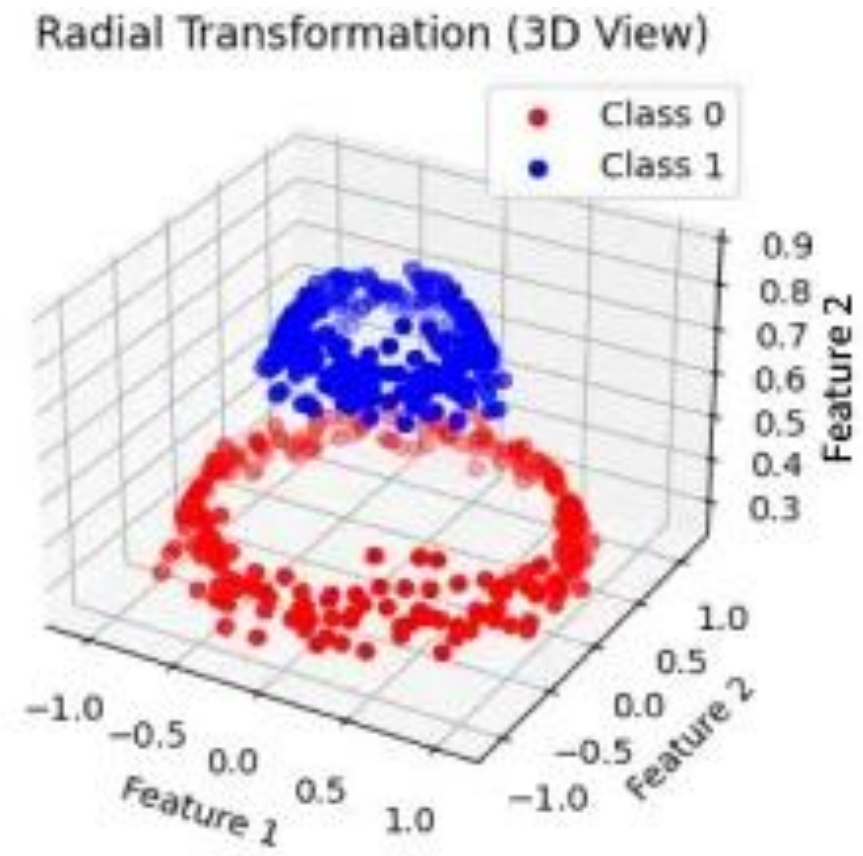
$$z = x^2 + y^2$$



Non-Linear SVM

- By adding this dimension, we will get three-dimensional space.
- Now you can see that the data has become **linearly separable**. As we are in three-dimensions now, the hyperplane we got is parallel to the x-axis at a particular value of z

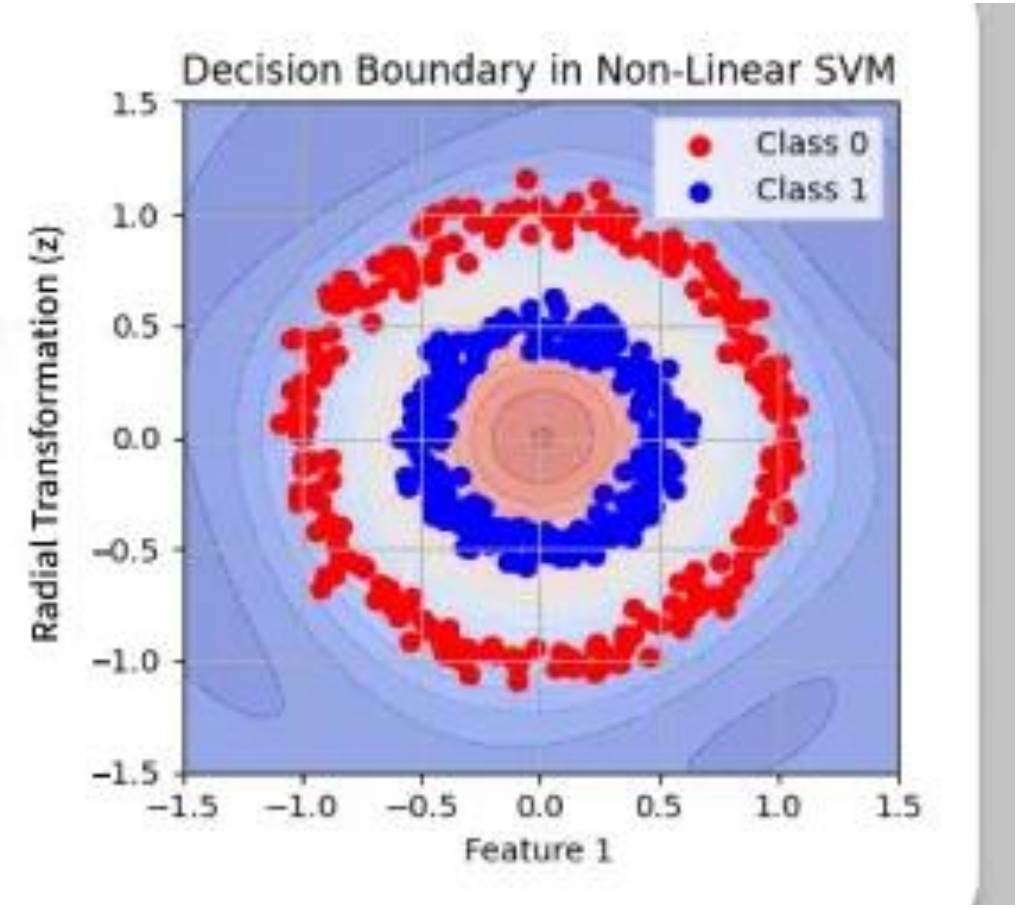
$$z = x^2 + y^2$$



Non-Linear SVM

- We can see that it is the equation of a circle. Hence, we can convert our linear separator in higher dimensions back to the original dimensions using this equation.

$$z = x^2 + y^2$$



Non-Linear SVM

- The linear SVM function $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b$
- Non-Linear SVM function $f(\mathbf{x}) = \mathbf{w}^T\phi(\mathbf{x}) + b$
- a **quadratic transformation**:

$$\phi(\mathbf{x}) = \begin{bmatrix} x_1 \\ x_2 \\ x_1^2 \\ x_2^2 \\ x_1x_2 \end{bmatrix}$$

This can be computationally expensive as the function gets more complicated

- Then the decision function becomes:

$$f(\mathbf{x}) = w_1x_1 + w_2x_2 + w_3x_1^2 + w_4x_2^2 + w_5x_1x_2 + b$$

- This is now a **non-linear function in the original space**, but it is **linear in the transformed space**.

The Kernel Trick

- Instead of working in the **original feature space**, **Kernel SVM maps the data into a higher-dimensional space**, where it **becomes linearly separable**.
- Instead of explicitly transforming the data (which can be computationally expensive), we use a **kernel function** $K(x_i,x_j)$ to compute the dot product **directly** in the higher-dimensional space.

$$K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j)$$

ϕ(x) is a **feature transformation function** (usually unknown).)
$K(x_i,x_j)$ is the **kernel function** that computes dot products in the transformed space **without explicitly computing ϕ(x).**
x_i and x_j are pairs of training points.

$$f(\mathbf{x}) = \sum_{i=1}^{N} y_i K(\mathbf{x}_i, \mathbf{x}) + b$$

Polynomial Kernal

- Captures non-linear relationships using polynomial transformations.
- More effective when the relationship between input features is polynomial.
- Can lead to overfitting if the degree is too high.
- Computationally expensive for large datasets.

$$K(x_i, x_j) = (\gamma x_i^T x_j + r)^d$$

d (degree of the polynomial) – Controls the flexibility of the decision boundary.

γ (scale factor) – Controls how much influence a single training sample has.

r (coefficient) – Controls the influence of higher-order terms.

Radial Basis Function (RBF) Kernal

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

- Handles high-dimensional data well.
- Can capture complex, non-linear relationships.
- More robust than polynomial kernels as it avoids overfitting with proper tuning.
- Computationally efficient compared to polynomial kernels.

$$K(x_i, x_j) = \exp(-\gamma ||x_i - x_j||^2)$$

γ – Determines the spread of the Gaussian function.
A small γ means a smoother decision boundary, while a large γ makes the model focus on individual points.

Sigmoid Kernel

- Inspired by neural networks (acts like a single-layer perceptron).
- Performs well in some cases but is **less popular** because it lacks **Mercer's condition** (does not always define a valid kernel for SVMs)
- Can behave unpredictably with certain hyperparameter values.

$$K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r)$$

γ – Controls the scaling.
r – Shifts the function.

SVM Kernels

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

Kernel	Best for	Pros	Cons	hyperparameters
Linear	Linearly separable data	- Fast training and prediction- Simple and interpretable- Works well with high-dimensional data	- Cannot handle non-linear data- May underfit complex problems	C: Regularization parameter
Polynomial	Data with polynomial-like relationships	Captures interactions between features	High-degree polynomials can overfit	- d: Degree of polynomial- γ (gamma): Scaling factor- r: Coefficient shift- C: Regularization
RBF (gaussian(	Complex and high-dimensional data	Handles non-linearity well, robust	Requires careful tuning of γγ	γ (gamma): Scaling factor-
Sigmoid	Data that can be separated like a neural network	Can approximate neural networks	Unstable, less commonly used	- γ (gamma): Scaling factor- r: Coefficient shift- C: Regularization

Contents

- Linear Models for Classification
  - Logistic Regression
  - Linear SVM
- Kernal SVM
- Naïve Bayes

Naïve Bayes

- Naive Bayes classifiers are supervised machine learning algorithms used for classification tasks, based on Bayes' Theorem to find probabilities.
- The main idea behind the Naive Bayes classifier is to use Bayes' Theorem to classify data based on the probabilities of different classes given the features of the data.
- It is a probabilistic classifier because it assumes that one feature in the model is independent of existence of another feature. In other words, each feature contributes to the predictions with no relation between each other.
- Naïve Bayes Algorithm is used in spam filtration, Sentimental analysis, classifying articles and many more.

Naïve Bayes Example

Consider a car theft problem with attributes Color, Type, Origin, and the target, Stolen can be either Yes or No.

#	Color	Type	Origin	Stolen
1	Red	Sports	Domestic	Yes
2	Red	Sports	Domestic	No
3	Red	Sports	Domestic	Yes
4	Yellow	Sports	Domestic	No
5	Yellow	Sports	Imported	Yes
6	Yellow	SUV	Imported	No
7	Yellow	SUV	Imported	Yes
8	Yellow	SUV	Domestic	No
9	Red	SUV	Imported	No
10	Red	Sports	Imported	Yes

Naïve Bayes Example Steps

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

1. Define the Problem
2. Compute Prior Probabilities.
3. Compute Likelihood Probabilities.
4. Compute Posterior Probabilities Using Bayes' Theorem.
5. Classify the New Data Point

#	Color	Type	Origin	Stolen
1	Red	Sports	Domestic	Yes
2	Red	Sports	Domestic	No
3	Red	Sports	Domestic	Yes
4	Yellow	Sports	Domestic	No
5	Yellow	Sports	Imported	Yes
6	Yellow	SUV	Imported	No
7	Yellow	SUV	Imported	Yes
8	Yellow	SUV	Domestic	No
9	Red	SUV	Imported	No
10	Red	Sports	Imported	Yes

Naïve Bayes Example Steps

1. Define the Problem

- We have three attributes:
  - **Color**: Red, Yellow
  - **Type**: Sports, SUV
  - **Origin**: Domestic, Imported
- **Target Variable (Stolen)**: Yes, No

Our goal is to classify a new car as **Stolen (Yes/No)** based on its **Color, Type, and Origin**.

#	Color	Type	Origin	Stolen
1	Red	Sports	Domestic	Yes
2	Red	Sports	Domestic	No
3	Red	Sports	Domestic	Yes
4	Yellow	Sports	Domestic	No
5	Yellow	Sports	Imported	Yes
6	Yellow	SUV	Imported	No
7	Yellow	SUV	Imported	Yes
8	Yellow	SUV	Domestic	No
9	Red	SUV	Imported	No
10	Red	Sports	Imported	Yes

Naïve Bayes Example Steps

1. Define the Problem

- We have three attributes:
  - **Color**: Red, Yellow
  - **Type**: Sports, SUV
  - **Origin**: Domestic, Imported
- **Target Variable (Stolen)**: Yes, No

Our goal is to classify a new car as **Stolen (Yes/No)** based on its **Color, Type, and Origin**.

#	Color	Type	Origin	Stolen
1	Red	Sports	Domestic	Yes
2	Red	Sports	Domestic	No
3	Red	Sports	Domestic	Yes
4	Yellow	Sports	Domestic	No
5	Yellow	Sports	Imported	Yes
6	Yellow	SUV	Imported	No
7	Yellow	SUV	Imported	Yes
8	Yellow	SUV	Domestic	No
9	Red	SUV	Imported	No
10	Red	Sports	Imported	Yes

Naïve Bayes Example Steps

2. Compute Priors *P(Y)*

The probability of a car being stolen or not:

$$P(\text{Stolen} = Yes) = \frac{\text{Count of Yes}}{\text{Total Samples}} = \boxed{???}$$

$$P(\text{Stolen} = No) = \frac{\text{Count of No}}{\text{Total Samples}} = \boxed{???}$$

#	Color	Type	Origin	Stolen
1	Red	Sports	Domestic	Yes
2	Red	Sports	Domestic	No
3	Red	Sports	Domestic	Yes
4	Yellow	Sports	Domestic	No
5	Yellow	Sports	Imported	Yes
6	Yellow	SUV	Imported	No
7	Yellow	SUV	Imported	Yes
8	Yellow	SUV	Domestic	No
9	Red	SUV	Imported	No
10	Red	Sports	Imported	Yes

Naïve Bayes Example Steps

3. Compute Likelihoods *P(X|Y)*

For Stolen = Yes

Color:

- $P(\text{Red}|\text{Yes}) =$???
- $P(\text{Yellow}|\text{Yes}) =$???

Type:

- $P(\text{Sports}|\text{Yes}) =$???
- $P(\text{SUV}|\text{Yes}) =$???

Origin:

- $P(\text{Domestic}|\text{Yes}) =$???
- $P(\text{Imported}|\text{Yes}) =$???

#	Color	Type	Origin	Stolen
1	Red	Sports	Domestic	Yes
2	Red	Sports	Domestic	No
3	Red	Sports	Domestic	Yes
4	Yellow	Sports	Domestic	No
5	Yellow	Sports	Imported	Yes
6	Yellow	SUV	Imported	No
7	Yellow	SUV	Imported	Yes
8	Yellow	SUV	Domestic	No
9	Red	SUV	Imported	No
10	Red	Sports	Imported	Yes

Naïve Bayes Example Steps

3. Compute Likelihoods *P(X|Y)*

For Stolen = No

Color:

- $P(\text{Red}|\text{No}) =$???
- $P(\text{Yellow}|\text{No}) =$???

Type:

- $P(\text{Sports}|\text{No}) =$???
- $P(\text{SUV}|\text{No}) =$???

Origin:

- $P(\text{Domestic}|\text{No}) =$???
- $P(\text{Imported}|\text{No}) =$???

#	Color	Type	Origin	Stolen
1	Red	Sports	Domestic	Yes
2	Red	Sports	Domestic	No
3	Red	Sports	Domestic	Yes
4	Yellow	Sports	Domestic	No
5	Yellow	Sports	Imported	Yes
6	Yellow	SUV	Imported	No
7	Yellow	SUV	Imported	Yes
8	Yellow	SUV	Domestic	No
9	Red	SUV	Imported	No
10	Red	Sports	Imported	Yes

Naïve Bayes Example Steps

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

4. Compute Posterior Probability for a New Car , Suppose we want to classify a **Red Sports Domestic** car.

#	Color	Type	Origin	Stolen
1	Red	Sports	Domestic	Yes
2	Red	Sports	Domestic	No
3	Red	Sports	Domestic	Yes
4	Yellow	Sports	Domestic	No
5	Yellow	Sports	Imported	Yes
6	Yellow	SUV	Imported	No
7	Yellow	SUV	Imported	Yes
8	Yellow	SUV	Domestic	No
9	Red	SUV	Imported	No
10	Red	Sports	Imported	Yes

For Stolen = Yes:

$$P(\text{Yes}|\text{Red, Sports, Domestic}) \propto P(\text{Red}|\text{Yes}) \cdot P(\text{Sports}|\text{Yes}) \cdot P(\text{Domestic}|\text{Yes}) \cdot P(\text{Yes})$$

$$= \boxed{???}$$

For Stolen = No:

$$P(\text{No}|\text{Red, Sports, Domestic}) \propto P(\text{Red}|\text{No}) \cdot P(\text{Sports}|\text{No}) \cdot P(\text{Domestic}|\text{No}) \cdot P(\text{No})$$

$$= \boxed{???}$$

Naïve Bayes Example Steps

5. Make the Prediction

Since

$$P(\text{Yes}|\text{Red}, \text{Sports}, \text{Domestic}) = 0.1125$$

Is greater than

$$P(\text{No}|\text{Red}, \text{Sports}, \text{Domestic}) = 0.03267$$

We classify the car as stolen

#	Color	Type	Origin	Stolen
1	Red	Sports	Domestic	Yes
2	Red	Sports	Domestic	No
3	Red	Sports	Domestic	Yes
4	Yellow	Sports	Domestic	No
5	Yellow	Sports	Imported	Yes
6	Yellow	SUV	Imported	No
7	Yellow	SUV	Imported	Yes
8	Yellow	SUV	Domestic	No
9	Red	SUV	Imported	No
10	Red	Sports	Imported	Yes

Naïve Bayes Example 2

We have a dataset of students with two numerical features:

We want to predict whether a new student who studies **4.0 hours** and sleeps **6.5 hours** will **Pass (Yes) or Fail (No)**.

Student	Study hours	Sleep Hours	Pass
1	2.5	5.0	No
2	3.0	6.5	No
3	3.5	7.0	No
4	4.5	6.0	No
5	5.0	8.0	Yes
6	5.5	8.5	Yes
7	6.0	7.5	Yes
8	6.5	9.0	Yes
9	7.0	9.5	Yes

Naïve Bayes Example

1. Step 1: Define the Dataset

We have a dataset of students with two numerical features:

1. **Study Hours** (per day)
2. **Sleep Hours** (per day)

Student	Study hours	Sleep Hours	Pass
1	2.5	5.0	No
2	3.0	6.5	No
3	3.5	7.0	No
4	4.5	6.0	No
5	5.0	8.0	Yes
6	5.5	8.5	Yes
7	6.0	7.5	Yes
8	6.5	9.0	Yes
9	7.0	9.5	Yes

Naïve Bayes Example

2. Step 2: Compute Priors P(Y)

The probability of passing or failing:

$$P(\text{Pass} = Yes) = \boxed{???}$$

$$P(\text{Pass} = No) = \boxed{???}$$

Student	Study hours	Sleep Hours	Pass
1	2.5	5.0	No
2	3.0	6.5	No
3	3.5	7.0	No
4	4.5	6.0	No
5	5.0	8.0	Yes
6	5.5	8.5	Yes
7	6.0	7.5	Yes
8	6.5	9.0	Yes
9	7.0	9.5	Yes

Naïve Bayes Example

3. Compute Mean and Standard Deviation for Each Feature Given Class Y.

We assume that **Study Hours** and **Sleep Hours** follow a **normal (Gaussian) distribution** for each class.

Student	Study hours	Sleep Hours	Pass
1	2.5	5.0	No
2	3.0	6.5	No
3	3.5	7.0	No
4	4.5	6.0	No
5	5.0	8.0	Yes
6	5.5	8.5	Yes
7	6.0	7.5	Yes
8	6.5	9.0	Yes
9	7.0	9.5	Yes

Naïve Bayes Example

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

3. Compute Mean and Standard Deviation for Each Feature Given Class Y.

For Pass (Yes)

Student	Study hours	Sleep Hours	Pass
1	2.5	5.0	No
2	3.0	6.5	No
3	3.5	7.0	No
4	4.5	6.0	No
5	5.0	8.0	Yes
6	5.5	8.5	Yes
7	6.0	7.5	Yes
8	6.5	9.0	Yes
9	7.0	9.5	Yes

Mean and Standard Deviation of Study Hours

$$\mu_{\text{Yes, Study}} = \boxed{???}$$

$$\sigma_{\text{Yes, Study}} = \sqrt{\frac{(5.0-6.0)^2 + (5.5-6.0)^2 + (6.0-6.0)^2 + (6.5-6.0)^2 + (7.0-6.0)^2}{5}}$$

$$= \sqrt{\frac{1.0 + 0.25 + 0 + 0.25 + 1.0}{5}} = \sqrt{0.5} = \boxed{???}$$

Mean and Standard Deviation of Sleep Hours

$$\mu_{\text{Yes, Sleep}} = \frac{8.0 + 8.5 + 7.5 + 9.0 + 9.5}{5} = 8.5$$

$$\sigma_{\text{Yes, Sleep}} = \sqrt{\frac{(8.0-8.5)^2 + (8.5-8.5)^2 + (7.5-8.5)^2 + (9.0-8.5)^2 + (9.5-8.5)^2}{5}}$$

$$= \sqrt{\frac{0.25 + 0 + 1.0 + 0.25 + 1.0}{5}} = \sqrt{0.5} = \boxed{???}$$

Naïve Bayes Example

McMaster University
ENGINEERING
W Booth School of Engineering Practice and Technology

3. Compute Mean and Standard Deviation for Each Feature Given Class Y.

For Fail (No)

Mean and Standard Deviation of Study Hours

$$\mu_{\text{No, Study}} = \frac{2.5 + 3.0 + 3.5 + 4.5}{4} = 3.375$$

$$\sigma_{\text{No, Study}} = \sqrt{\frac{(2.5 - 3.375)^2 + (3.0 - 3.375)^2 + (3.5 - 3.375)^2 + (4.5 - 3.375)^2}{4}}$$

$$= \sqrt{\frac{0.7656 + 0.1406 + 0.0156 + 1.2656}{4}} = \sqrt{0.5469} = 0.74$$

Mean and Standard Deviation of Sleep Hours

$$\mu_{\text{No, Sleep}} = \frac{5.0 + 6.5 + 7.0 + 6.0}{4} = 6.125$$

$$\sigma_{\text{No, Sleep}} = \sqrt{\frac{(5.0 - 6.125)^2 + (6.5 - 6.125)^2 + (7.0 - 6.125)^2 + (6.0 - 6.125)^2}{4}}$$

$$= \sqrt{\frac{1.2656 + 0.1406 + 0.7656 + 0.0156}{4}} = \sqrt{0.5469} = 0.74$$

Student	Study hours	Sleep Hours	Pass
1	2.5	5.0	No
2	3.0	6.5	No
3	3.5	7.0	No
4	4.5	6.0	No
5	5.0	8.0	Yes
6	5.5	8.5	Yes
7	6.0	7.5	Yes
8	6.5	9.0	Yes
9	7.0	9.5	Yes

Naïve Bayes Example

4. Compute Likelihoods Using Gaussian Formula

The probability density function of a normal distribution:

$$P(X|Y) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(X-\mu)^2}{2\sigma^2}}$$

For the new student with **Study Hours = 4.0** and **Sleep Hours = 6.5**:

Student	Study hours	Sleep Hours	Pass
1	2.5	5.0	No
2	3.0	6.5	No
3	3.5	7.0	No
4	4.5	6.0	No
5	5.0	8.0	Yes
6	5.5	8.5	Yes
7	6.0	7.5	Yes
8	6.5	9.0	Yes
9	7.0	9.5	Yes

Naïve Bayes Example

4. Compute Likelihoods Using Gaussian Formula

For Pass (Yes)

$$P(4.0|\text{Yes}) = \frac{1}{\sqrt{2\pi(0.71)^2}} e^{-\frac{(4.0-6.0)^2}{2(0.71)^2}}$$

$$= \frac{1}{1.78} e^{-\frac{4}{1.01}}$$

$$= 0.21$$

$$P(6.5|\text{Yes}) = \frac{1}{1.78} e^{-\frac{(6.5-8.5)^2}{1.01}} = 0.21$$

$$P(Yes|X) \propto (0.21) \times (0.21) \times (0.56) = 0.0247$$

Student	Study hours	Sleep Hours	Pass
1	2.5	5.0	No
2	3.0	6.5	No
3	3.5	7.0	No
4	4.5	6.0	No
5	5.0	8.0	Yes
6	5.5	8.5	Yes
7	6.0	7.5	Yes
8	6.5	9.0	Yes
9	7.0	9.5	Yes

Naïve Bayes Example

4. Compute Likelihoods Using Gaussian Formula

For Fail (No)

$$P(4.0|\text{No}) = \frac{1}{1.85} e^{-\frac{(4.0-3.375)^2}{1.09}} = 0.41$$

$$P(6.5|\text{No}) = \frac{1}{1.85} e^{-\frac{(6.5-6.125)^2}{1.09}} = 0.49$$

$$P(No|X) \propto (0.41) \times (0.49) \times (0.44) = 0.088$$

Student	Study hours	Sleep Hours	Pass
1	2.5	5.0	No
2	3.0	6.5	No
3	3.5	7.0	No
4	4.5	6.0	No
5	5.0	8.0	Yes
6	5.5	8.5	Yes
7	6.0	7.5	Yes
8	6.5	9.0	Yes
9	7.0	9.5	Yes

Naïve Bayes Example

5. Predict the Class

Since

$$P(No|X) = 0.088 > P(Yes|X) = 0.0247$$

We predict **Fail (No)**.

Student	Study hours	Sleep Hours	Pass
1	2.5	5.0	No
2	3.0	6.5	No
3	3.5	7.0	No
4	4.5	6.0	No
5	5.0	8.0	Yes
6	5.5	8.5	Yes
7	6.0	7.5	Yes
8	6.5	9.0	Yes
9	7.0	9.5	Yes

Zero Probability Problem

- If a feature **never appears** in the training data for a class, its probability becomes **zero**.
- This leads to **multiplication by zero**, making the entire probability **zero**, even if other features suggest a different outcome.
- Solution: Laplace Smoothing (Additive Smoothing)

$$P(X|Y) = \frac{\text{count}(X, Y) + k}{\text{total count}(Y) + k \times V}$$

- **V** = Total number of unique feature values
- **k** = Smoothing parameter (typically **1** for Laplace Smoothing)

Advantages

- It is not only a simple approach but also a fast and accurate method for prediction.
- Naive Bayes has a very low computation cost.
- It can efficiently work on a large dataset.
- It performs well in case of discrete response variable compared to the continuous variable.
- It can be used with multiple class prediction problems.
- It also performs well in the case of text analytics problems.
- When the assumption of independence holds, a Naive Bayes classifier performs better compared to other models like logistic regression.

Disadvantages

- The assumption of independent features. In practice, it is almost impossible that model will get a set of predictors which are entirely independent.
- If there is no training tuple of a particular class, this causes zero posterior probability. In this case, the model is unable to make predictions. This problem is known as Zero Probability/Frequency Problem.

Summary

Logistic Regression

- Predicts the probability of an outcome using a **sigmoid curve**.
- Works well for **binary classification** and linearly separable data.
- Provides **interpretable probability scores** but struggles with **complex decision boundaries**.

Support Vector Machine (SVM)

- Finds the **optimal boundary (hyperplane)** that maximizes margin between classes.
- Effective for **high-dimensional data** and **non-linear patterns** (via kernel tricks).
- Can be slow for **large datasets** but excels in complex feature spaces.

Naïve Bayes

- A **probabilistic classifier** that assumes features are **independent**.
- Works well for **text classification (spam filtering, sentiment analysis)**.
- Extremely **fast and scalable**, but assumptions may not always hold.