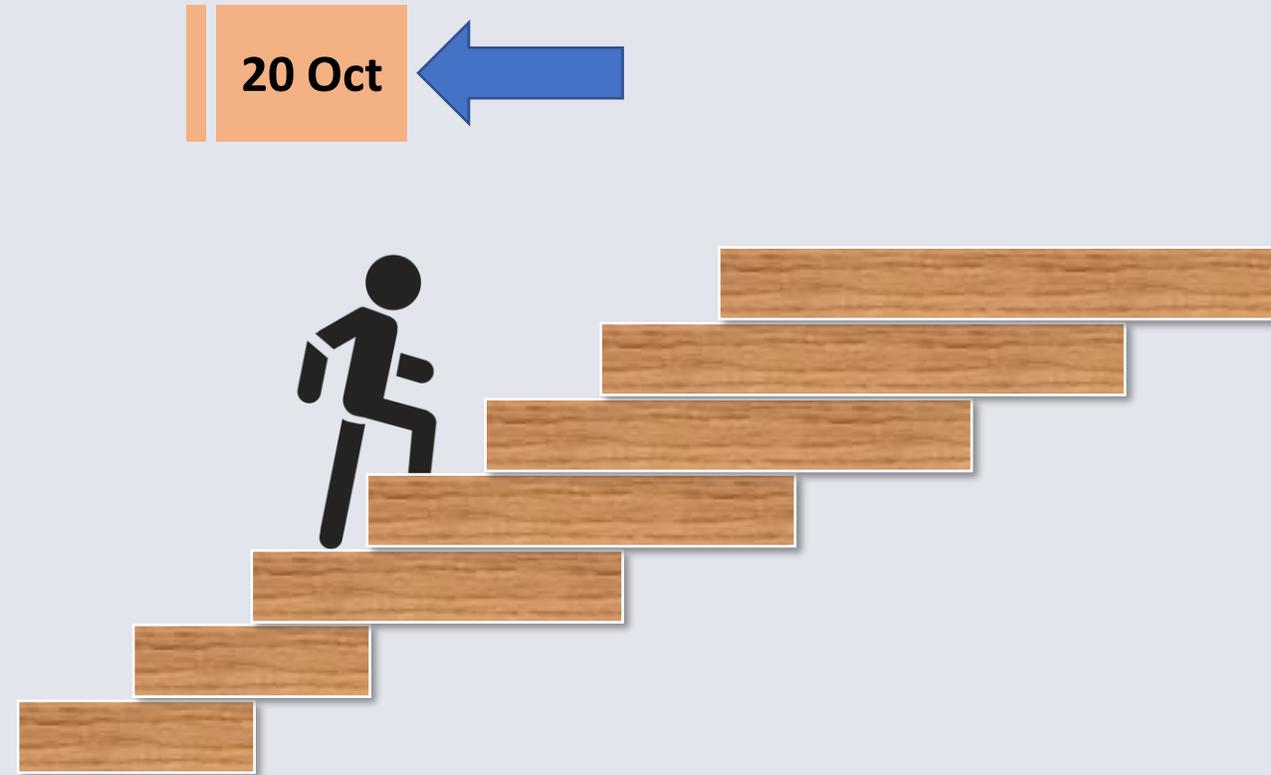


## Lecture 5

# Supervised Learning-I

# Previous Week Recap (..)

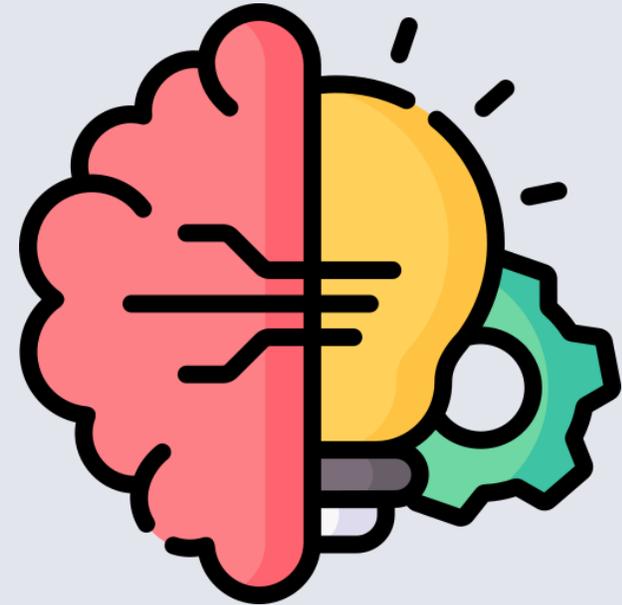
- **Key Challenges/Concepts**
  - **Overfitting vs Underfitting**
  - **Bias-Variance Tradeoff**
- **Supervised Learning Algorithm**
  - **KNN**



# Today's Contents



- **Supervised Learning Algorithms – Classification**
  - **Support Vector Machines (SVM)**
  - **Approaches which SVM works/use**
  - **Underlined concepts.**



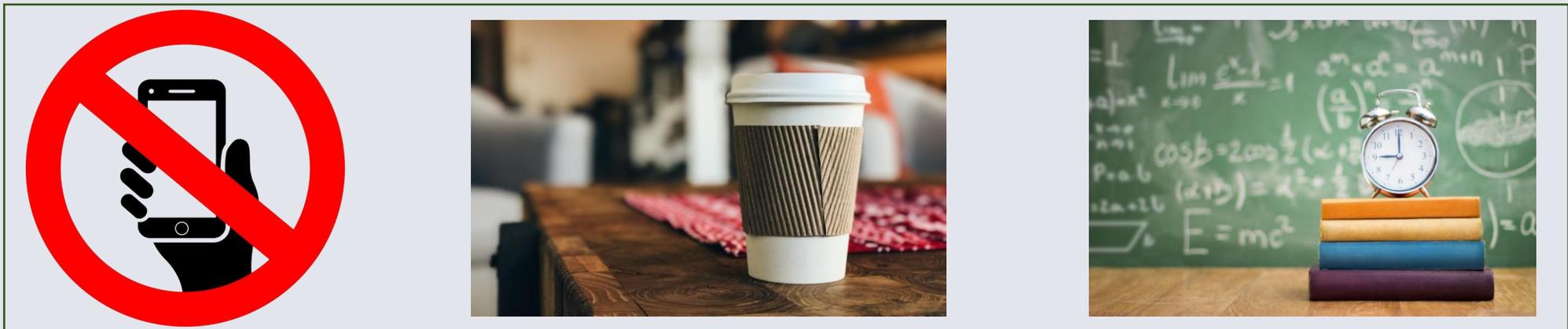
# Today's Contents



## Learning Objectives of Today's Lecture

- Learning about classification algorithms such as Support Vector Machine
- Understand the core concepts behind SVM and why it's useful.
- Approaches used in Support Vector Machine

- **Important Directions**



# Support Vector Machine



## What is SVM?

Developed in the 1990s by Vladimir N. Vapnik and colleagues

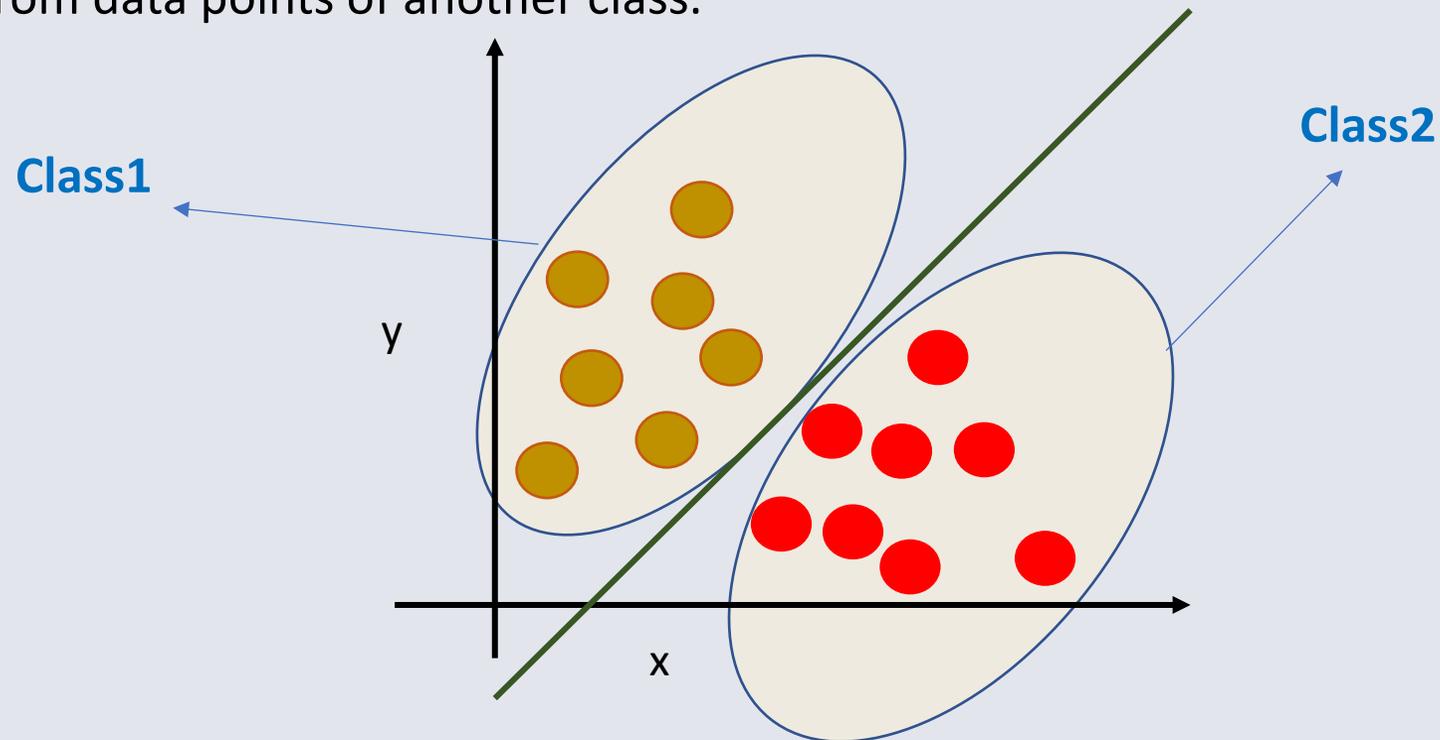
- A powerful machine learning algorithm widely used for both **linear** and **nonlinear** classification, as well as regression and outlier detection tasks.
- SVMs are highly adaptable, making them suitable for various applications such as
  - Text classification, image classification, spam detection, handwriting identification, gene expression analysis, face detection, and anomaly detection.

# Support Vector Machine



## What is SVM?

The idea behind SVM is to find a **boundary** (or decision boundary) that best separates data points of one class from data points of another class.

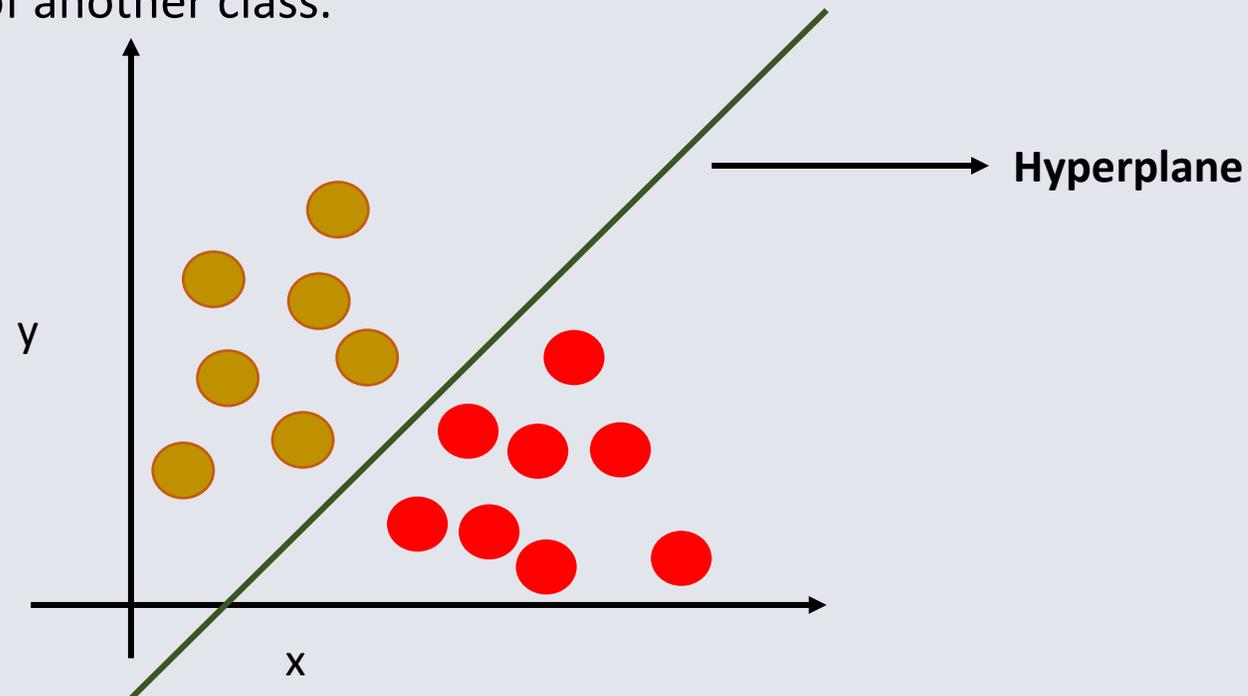


# Support Vector Machine



## What is SVM?

The idea behind SVM is to find a **boundary** (or decision boundary) that best separates data points of one class from data points of another class.



# Support Vector Machine



## Scenario: Email Spam Detection

Imagine you building a system to classify (**Spam, not spam**)

Algorithms to classify incoming emails as either **spam** or **not spam**



Not spam



Spam

Data points

# Support Vector Machine



## Scenario: Email Spam Detection

**Goal:** Model that can learn to distinguish between these two classes using patterns in past emails.

Features for each email:

**X1:** Frequency of the word "**free**"

**X2:** Frequency of the word "**urgent**"

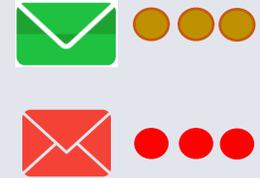
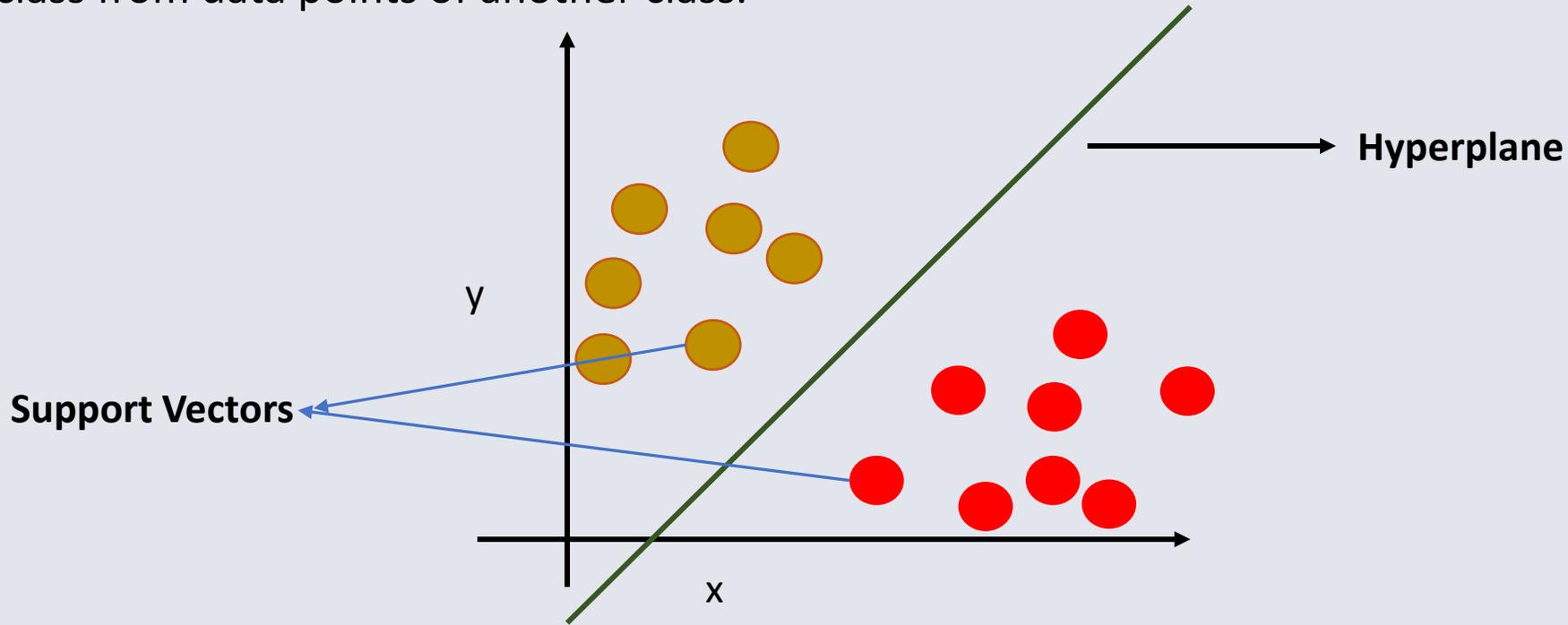
**Other potential features:** Number of links, sender reputation, presence of certain phrases, etc.

# Support Vector Machine



## What is SVM?

The idea behind SVM is to find a **boundary** (or decision boundary) that best separates data points of one class from data points of another class.

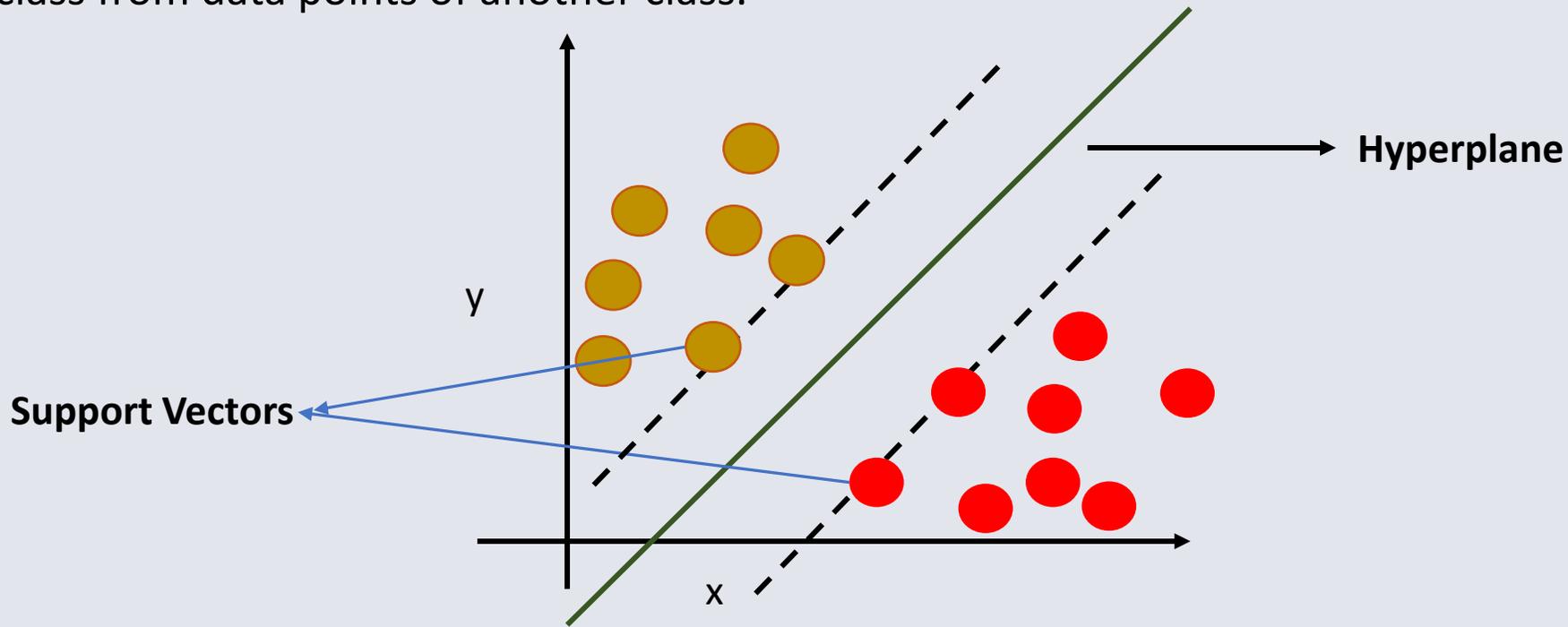


# Support Vector Machine



## What is SVM?

The idea behind SVM is to find a **boundary** (or decision boundary) that best separates data points of one class from data points of another class.

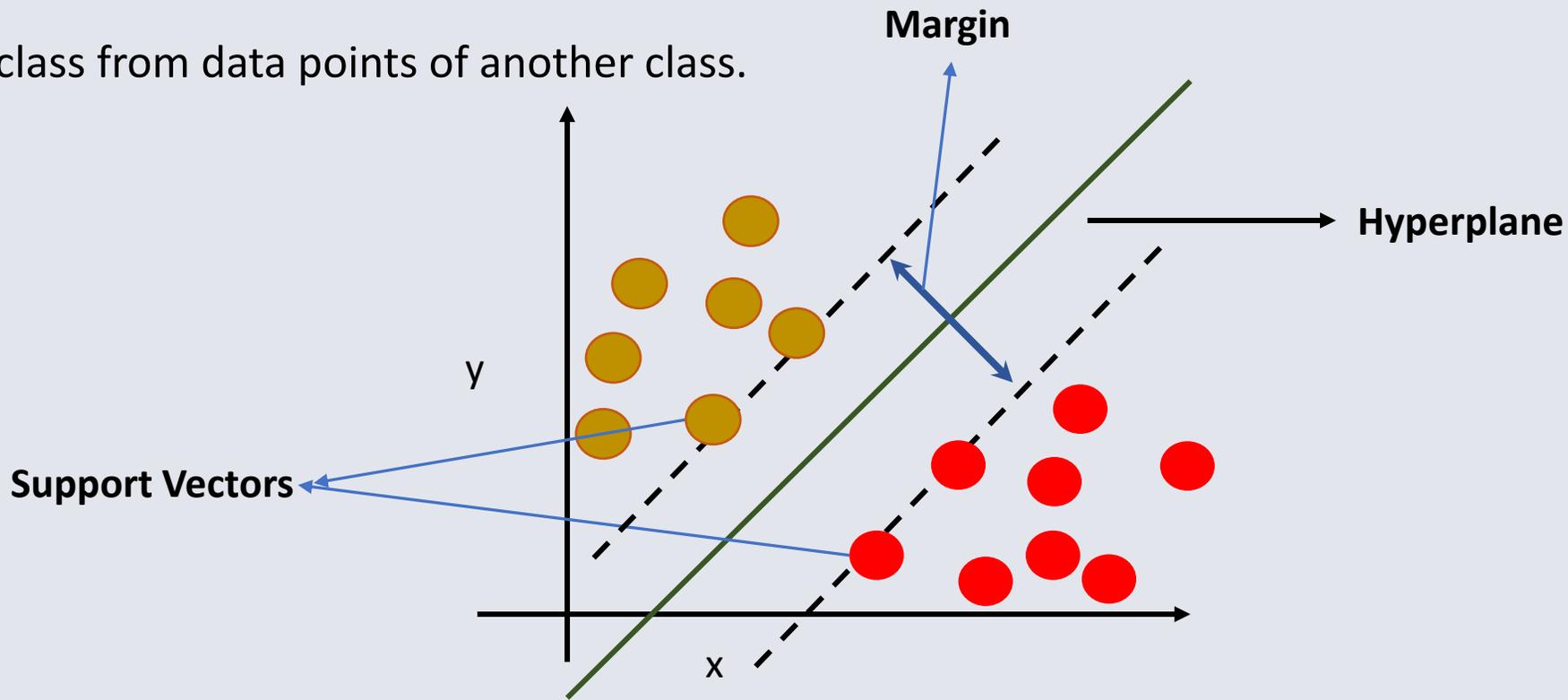


# Support Vector Machine



## What is SVM?

The idea behind SVM is to find a **boundary** (or decision boundary) that best separates data points of one class from data points of another class.

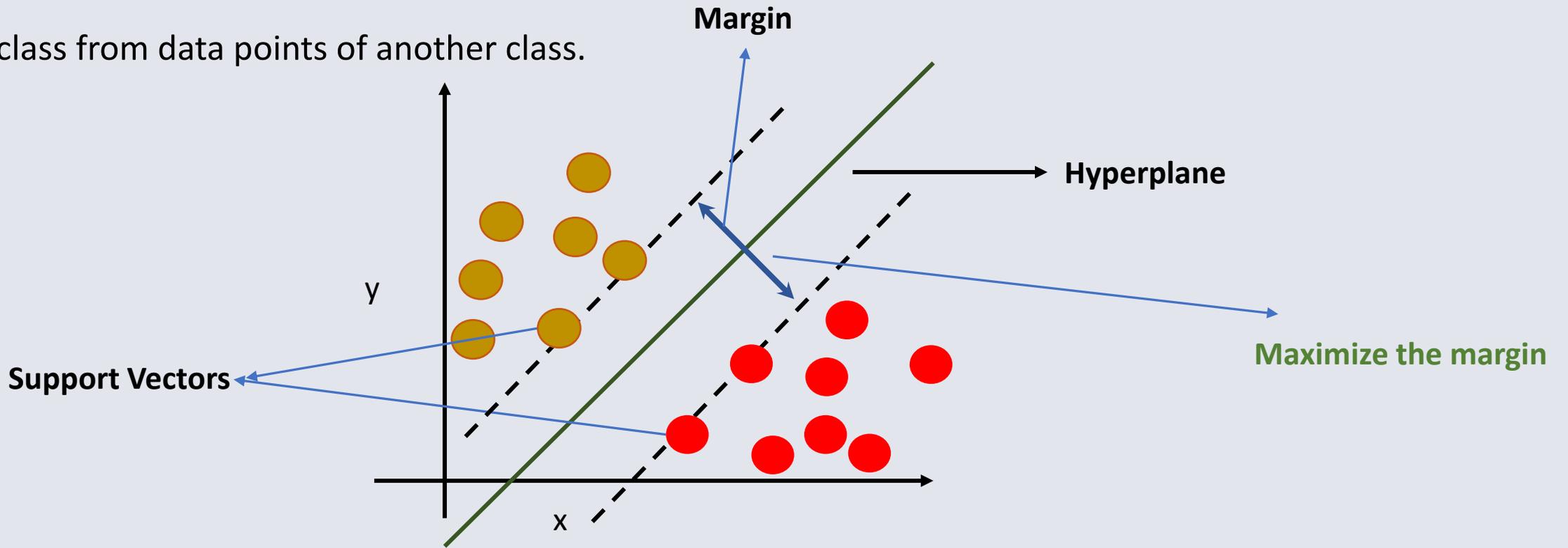


# Support Vector Machine



## What is SVM?

The idea behind SVM is to find a **boundary** (or decision boundary) that best separates data points of one class from data points of another class.



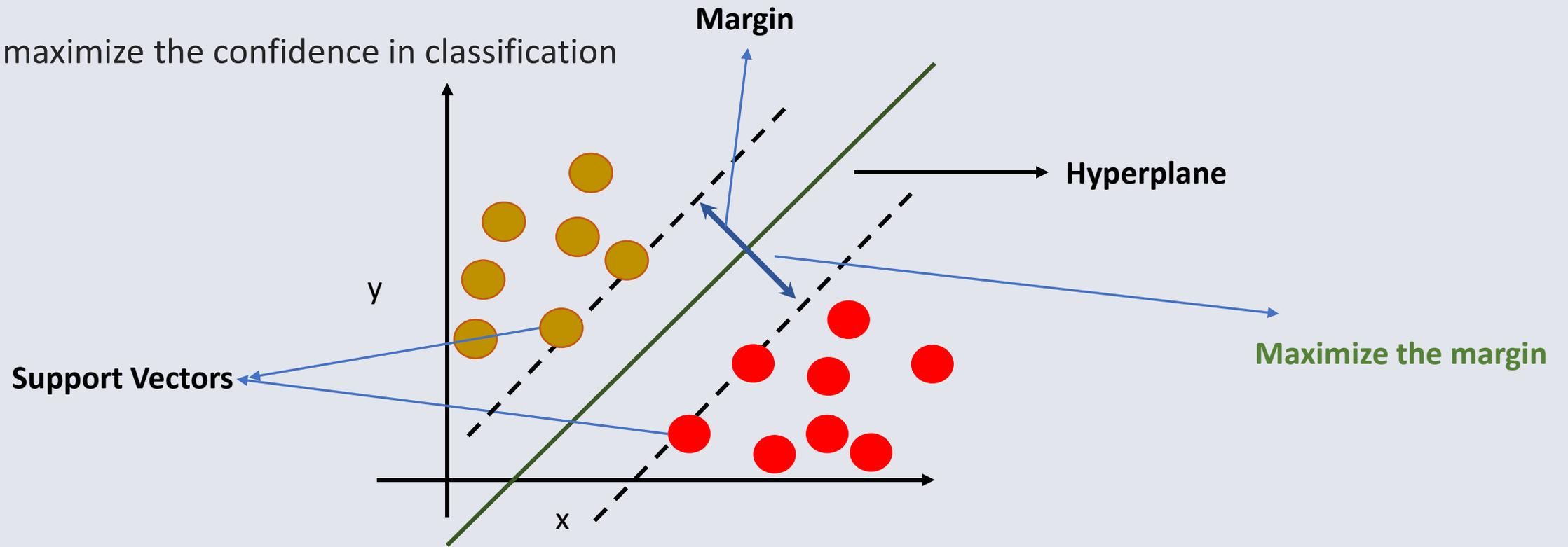
# Support Vector Machine



## Finding the Optimal Hyperplane

SVM selects the line (hyperplane) that **maximizes the margin** between the two classes (Spam, Not spam).

This maximize the confidence in classification



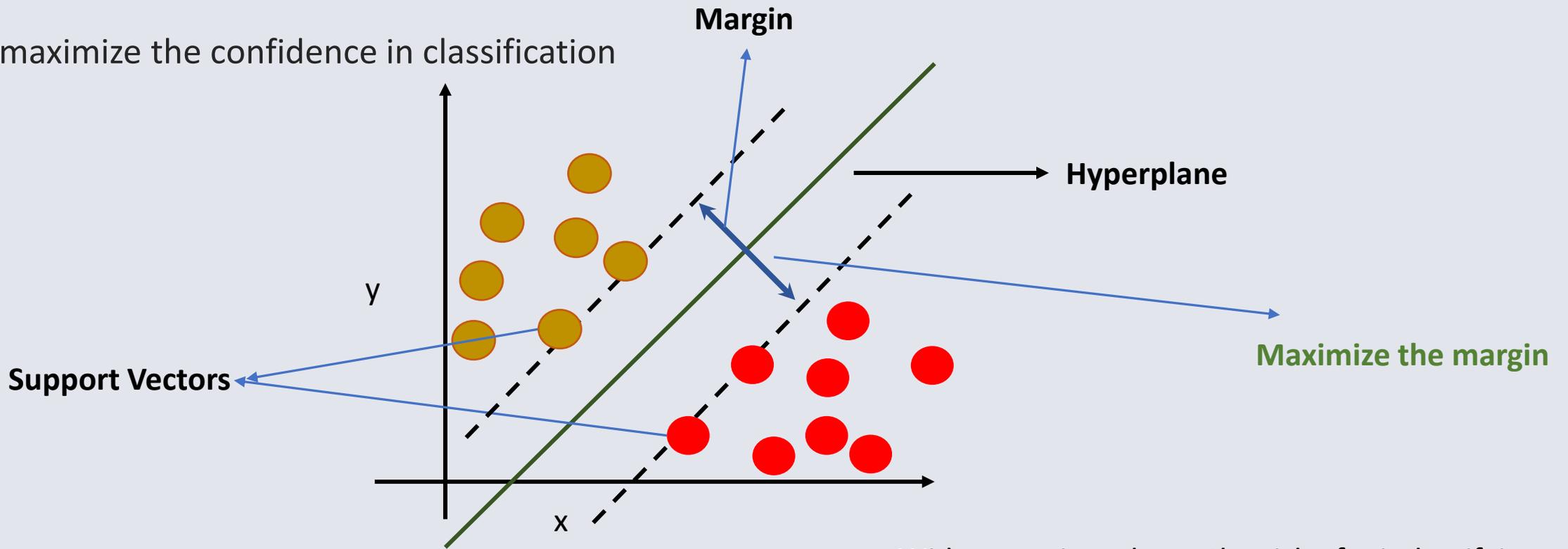
# Support Vector Machine



## Finding the Optimal Hyperplane

SVM selects the line (hyperplane) that **maximizes the margin** between the two classes (Spam, Not spam).

This maximizes the confidence in classification



Wider margin reduces the risk of misclassifying emails.



# Support Vector Machine

## Finding the Optimal Hyperplane

### Why Maximize the Margin?

- A wider margin reduces the risk of misclassifying emails.
- It helps the model generalize better to new emails.

### Equation of Hyperplane

$$w \cdot x + b = 0$$

where  $w$  is the weight vector and  $b$  is the bias.

# Support Vector Machine

## Non-Linearly Separable



### Problem with Linear Boundaries:

**Facial Expression Recognition:** Imagine we're building a model to classify facial expressions into six categories (**Angry**, **Happy**, **Sad**, **Neutral**, **Fear**, **Surprise**).

Each face image can be represented by features like **edge patterns**, **texture**, **brightness**, and **key points on the face** (like the position of eyebrows, mouth, etc.).

**Challenge:** These six expressions may not be linearly separable due to overlapping or similar features.

For example, "Surprise" and "Fear" might have similar wide-eyed expressions.

# Support Vector Machine



## Non-Linearly Separable

### Solution with Kernels:

SVM can use a *kernel function* to map the data into a higher-dimensional space where it becomes easier to separate.

**Radial Basis Function (RBF) Kernel:** Useful for non-linear data; helps separate classes by creating a more flexible decision boundary.

**One-vs-One Strategy:** SVM can classify pairs of expressions at a time. For example, one SVM model might classify "Happy" vs. "Sad," another "Angry" vs. "Neutral," etc., for all combinations.

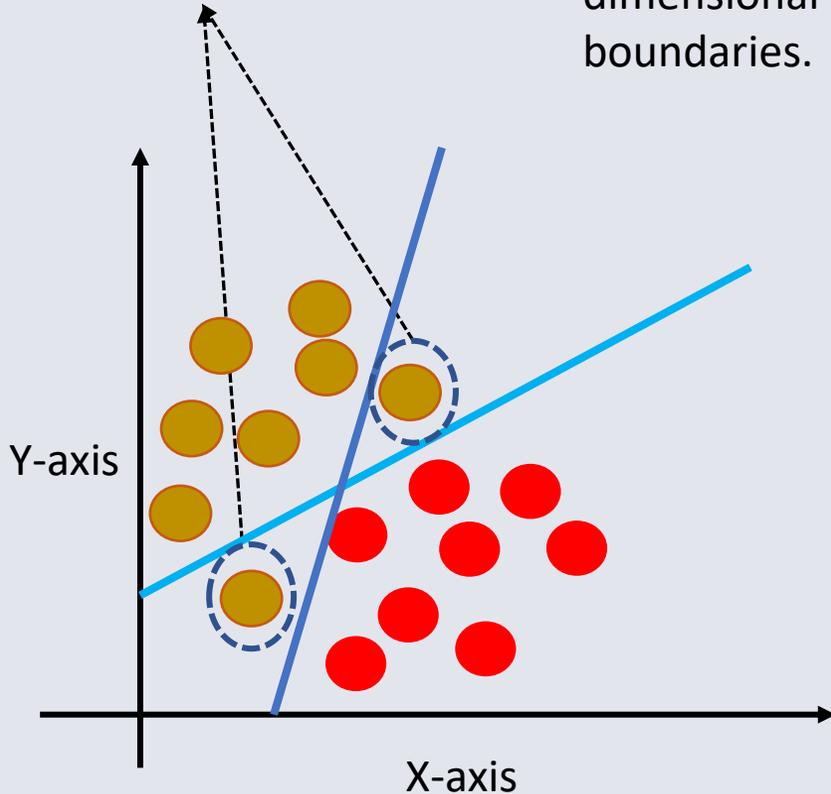
**One-vs-All Strategy:** Score for each class is computed based on the output of the respective binary classifier that was trained to distinguish that class from all others

# Support Vector Machine

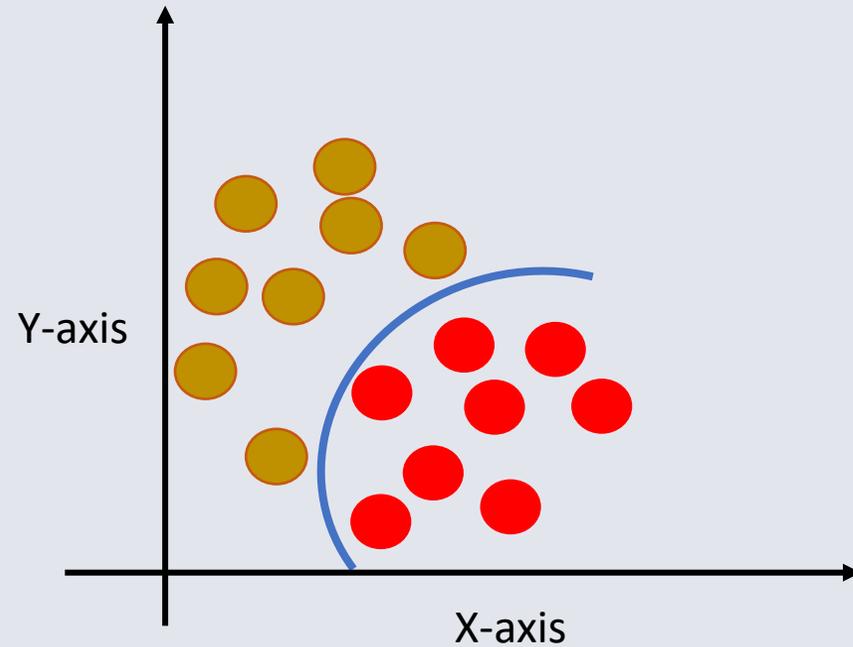


## Non-Linearly Separable

Misclassification



**RBF** maps the features of facial expressions into a higher-dimensional space, making it easier to separate them with curved boundaries.



Using **linear boundary** isn't feasible as the data points overlap

**kernel function**, like the Radial Basis Function (RBF) kernel.

# Support Vector Machine

(Angry, Fear, Happy, Neutral, Sad, Surprise)



## SVM Model for Facial Expression Recognition

### 1. Feature Extraction and Transformation

For new face, key features like **mouth width, eyebrow shapes, eye size, texture patterns**

These features are mapped into higher-dimensional space using RBF kernel so that model can apply non-linear boundaries

### 2. Decision Boundaries

SVM uses the transformed data to predict which expression the new face resembles by placing it in one of six classes.

Each SVM model in the **One-vs-One approach** votes for a class, and the majority vote determines the final label.

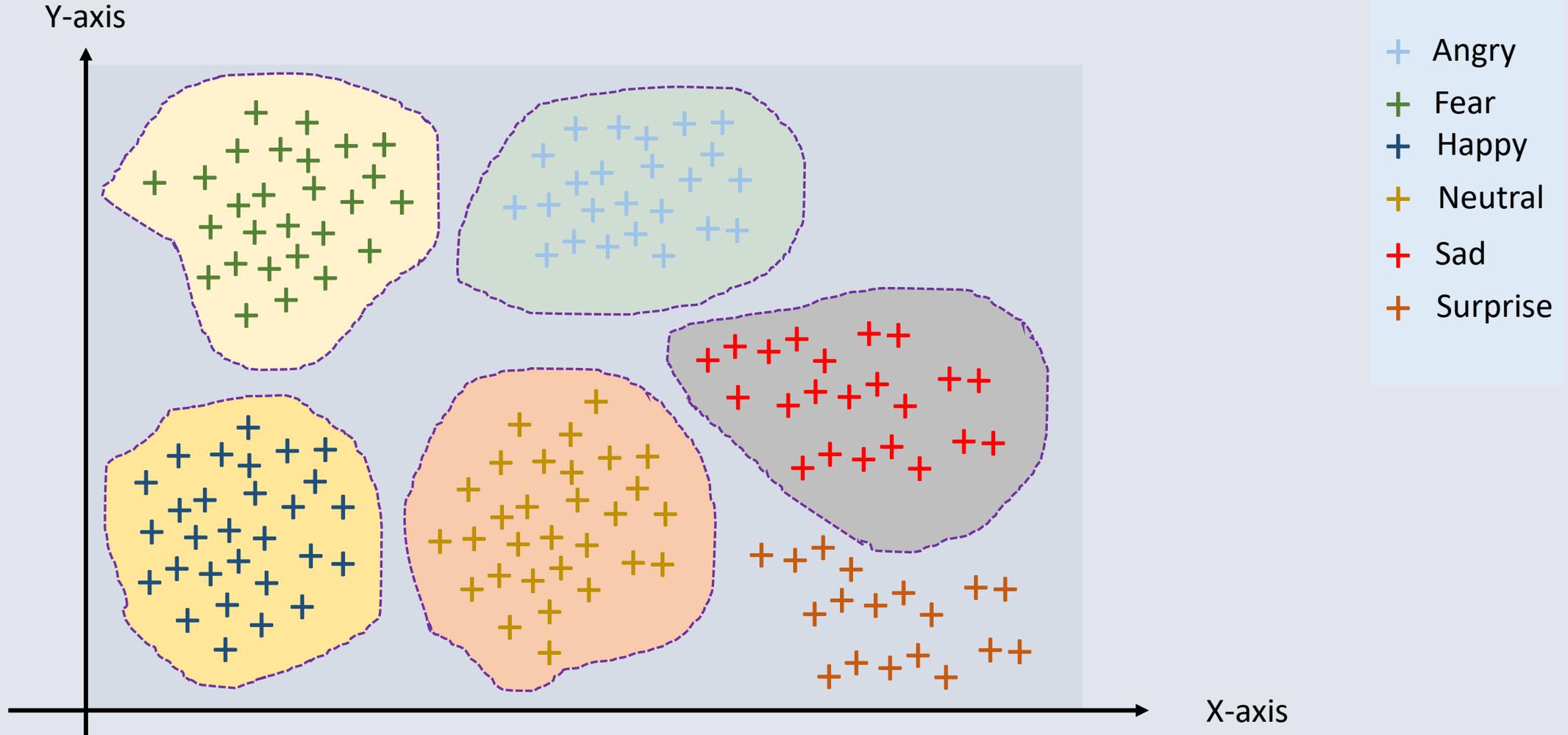
Score for each class is computed **One-vs-All approach** based on the output of the respective binary classifier that was trained to distinguish that class from all others

# Support Vector Machine

(Angry, Fear, Happy, Neutral, Sad, Surprise)



## SVM Model for Facial Expression Recognition: One-vs-One approach

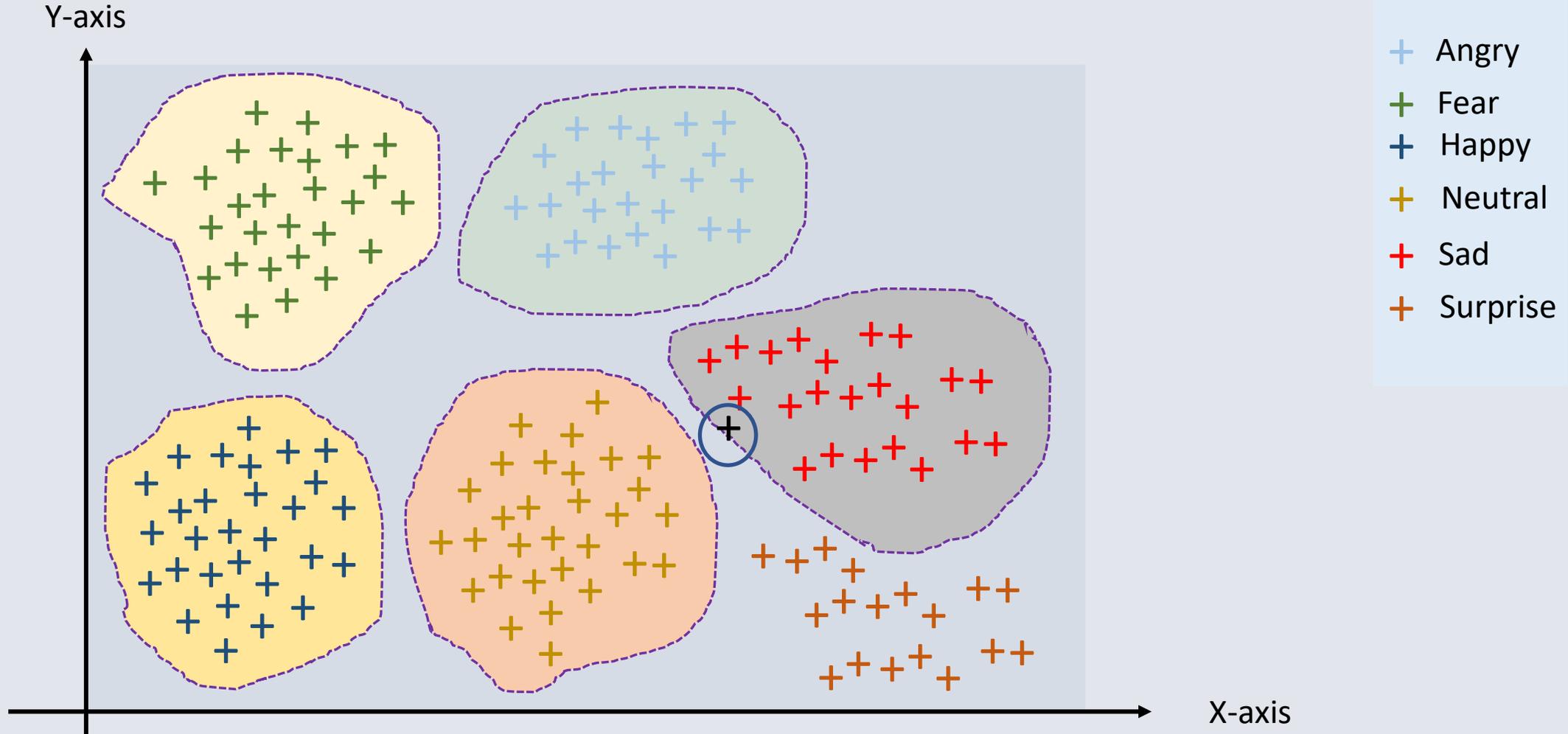


# Support Vector Machine

(Angry, Fear, Happy, Neutral, Sad, Surprise)



## SVM Model for Facial Expression Recognition: One-vs-One approach

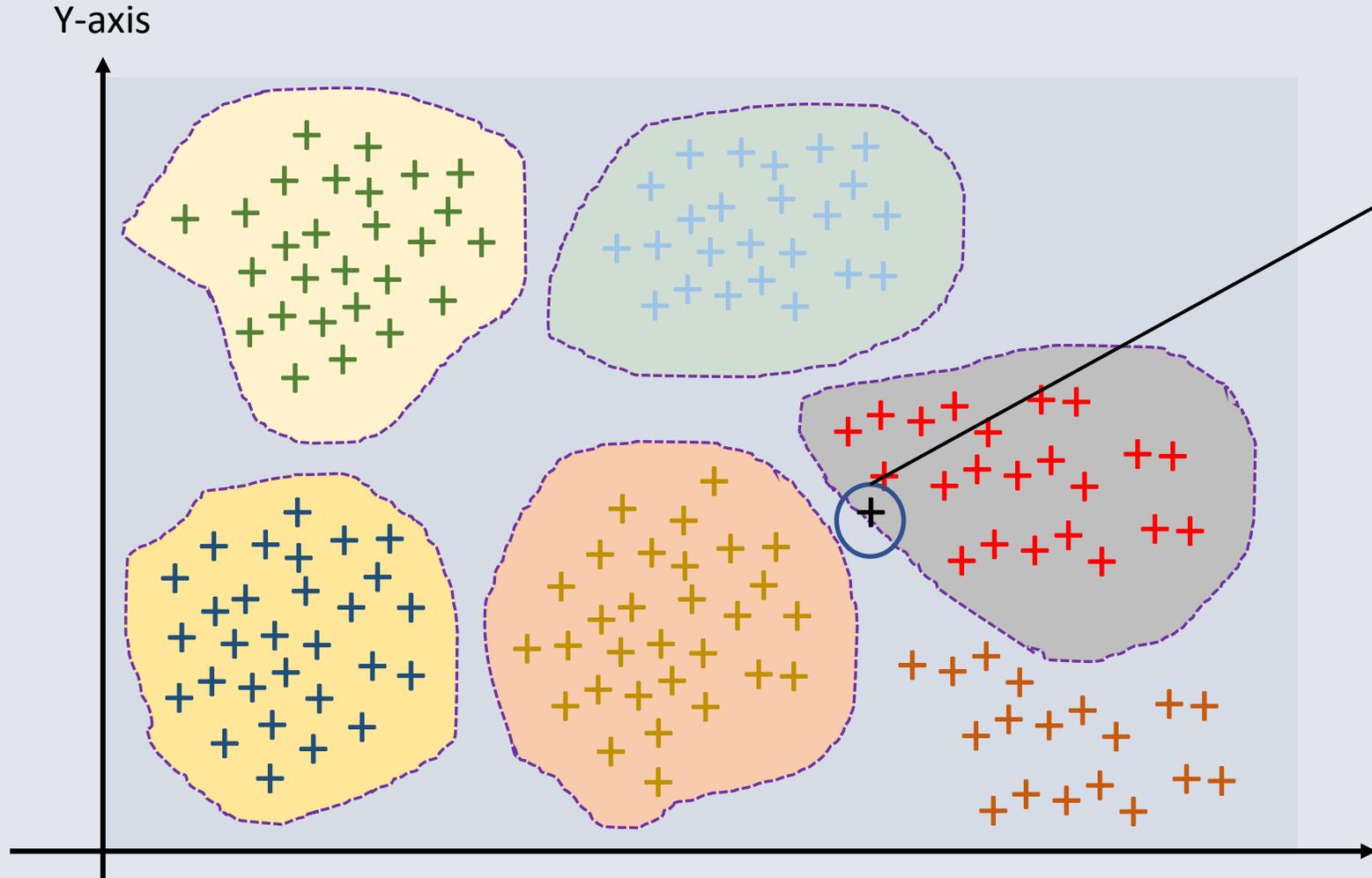


# Support Vector Machine

(Angry, Fear, Happy, Neutral, Sad, Surprise)



## SVM Model for Facial Expression Recognition: One-vs-One approach



**Each classifier** (trained on a pair of classes) will evaluate the image and decide which of the two classes the image belongs to.

The classifier's decision is based on where the image falls relative to the decision boundary between the two classes. The SVM finds the side of the boundary where the image lies.

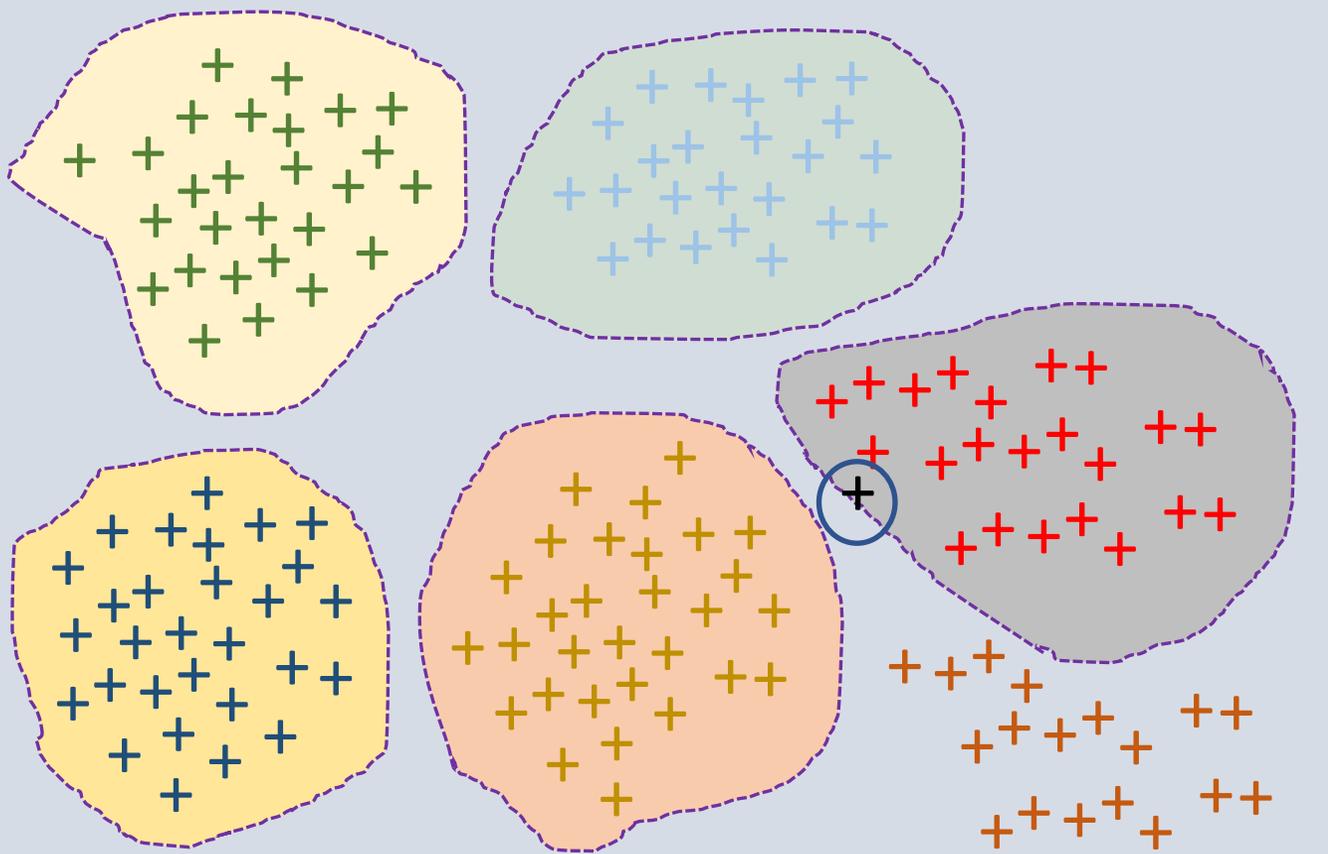
# Support Vector Machine

(Angry, Fear, Happy, Neutral, Sad, Surprise)



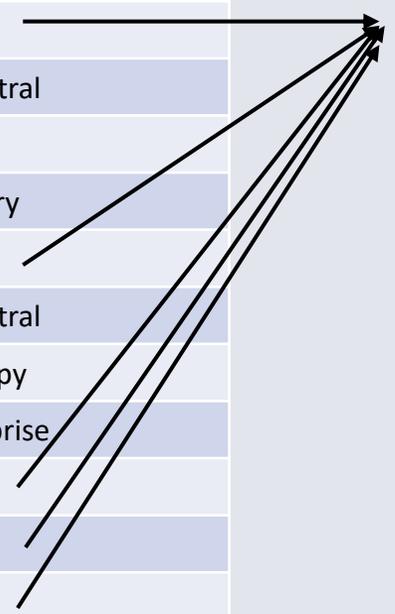
## SVM Model for Facial Expression Recognition: One-vs-One approach

Y-axis



Pair (Classifier)	Prediction
Ang Vs Happy	Angry
Ang Vs Sad	Sad
Ang Vs Neutral	Neutral
Ang Vs Fear	Fear
Ang Vs Surprise	Angry
Hap Vs Sad	Sad
Hap Vs Neutral	Neutral
Hap Vs Fear	Happy
Hap Vs Surprise	Surprise
Sad Vs Neutral	Sad
Sad Vs Fear	Sad
Sad Vs Surprise	Sad
Neutral Vs Fear	Fear
Neutral Vs Surprise	Neutral
Fear Vs Surprise	Fear

5  
Sad



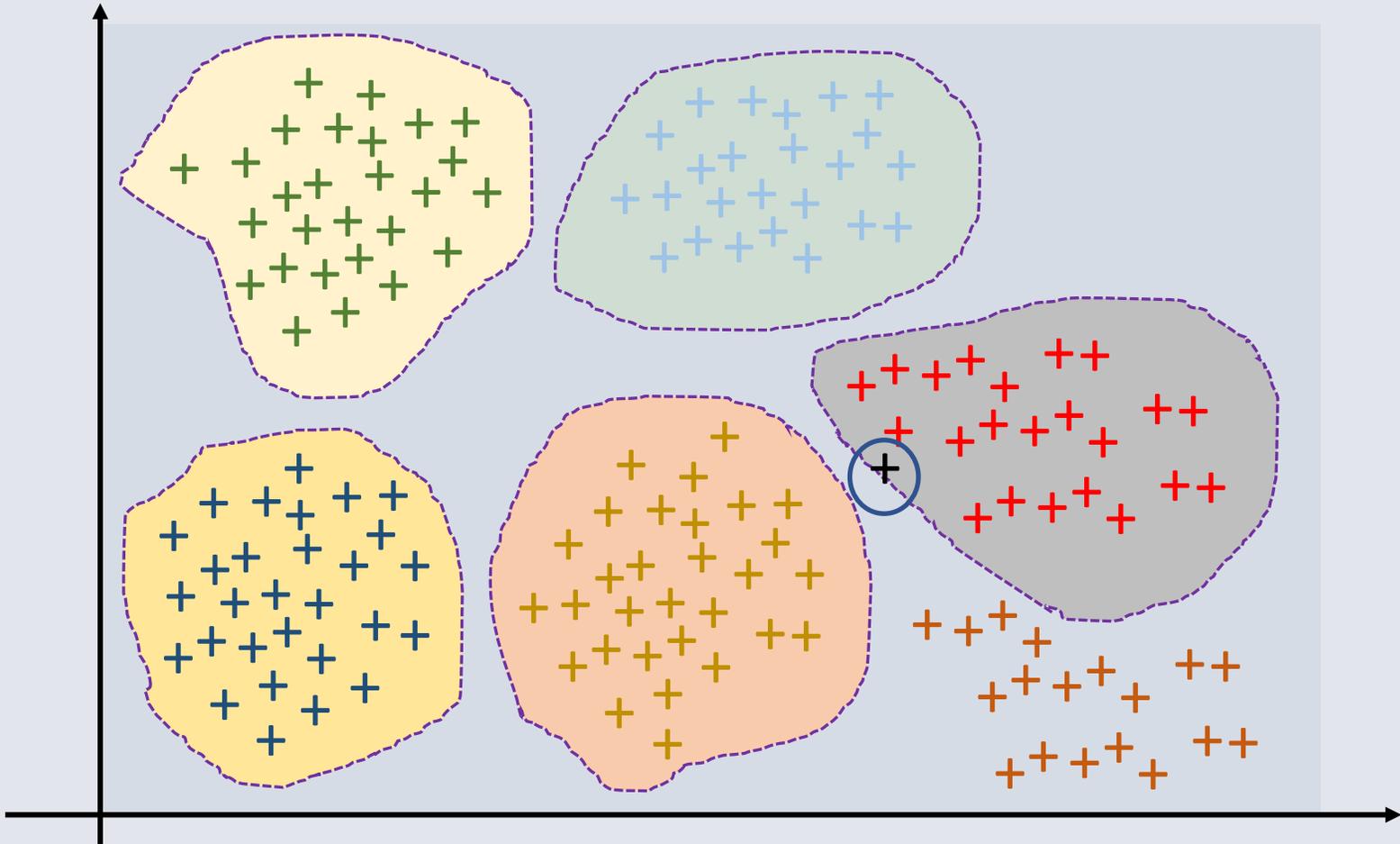
# Support Vector Machine

(Angry, Fear, Happy, Neutral, Sad, Surprise)



## SVM Model for Facial Expression Recognition: One-vs-One approach

Y-axis



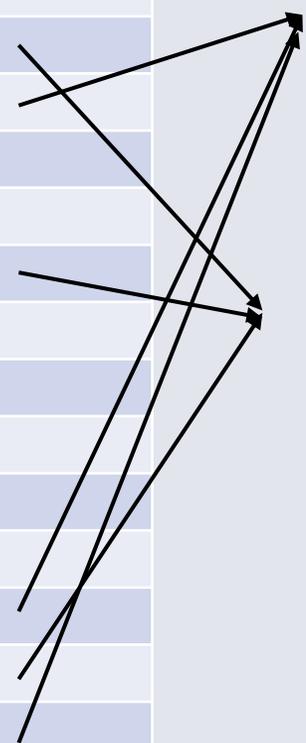
Pair (Classifier)	Prediction
Ang Vs Happy	Angry
Ang Vs Sad	Sad
Ang Vs Neutral	Neutral
Ang Vs Fear	Fear
Ang Vs Surprise	Angry
Hap Vs Sad	Sad
Hap Vs Neutral	Neutral
Hap Vs Fear	Happy
Hap Vs Surprise	Surprise
Sad Vs Neutral	Sad
Sad Vs Fear	Sad
Sad Vs Surprise	Sad
Neutral Vs Fear	Fear
Neutral Vs Surprise	Neutral
Fear Vs Surprise	Fear

3

Fear

3

Neutral



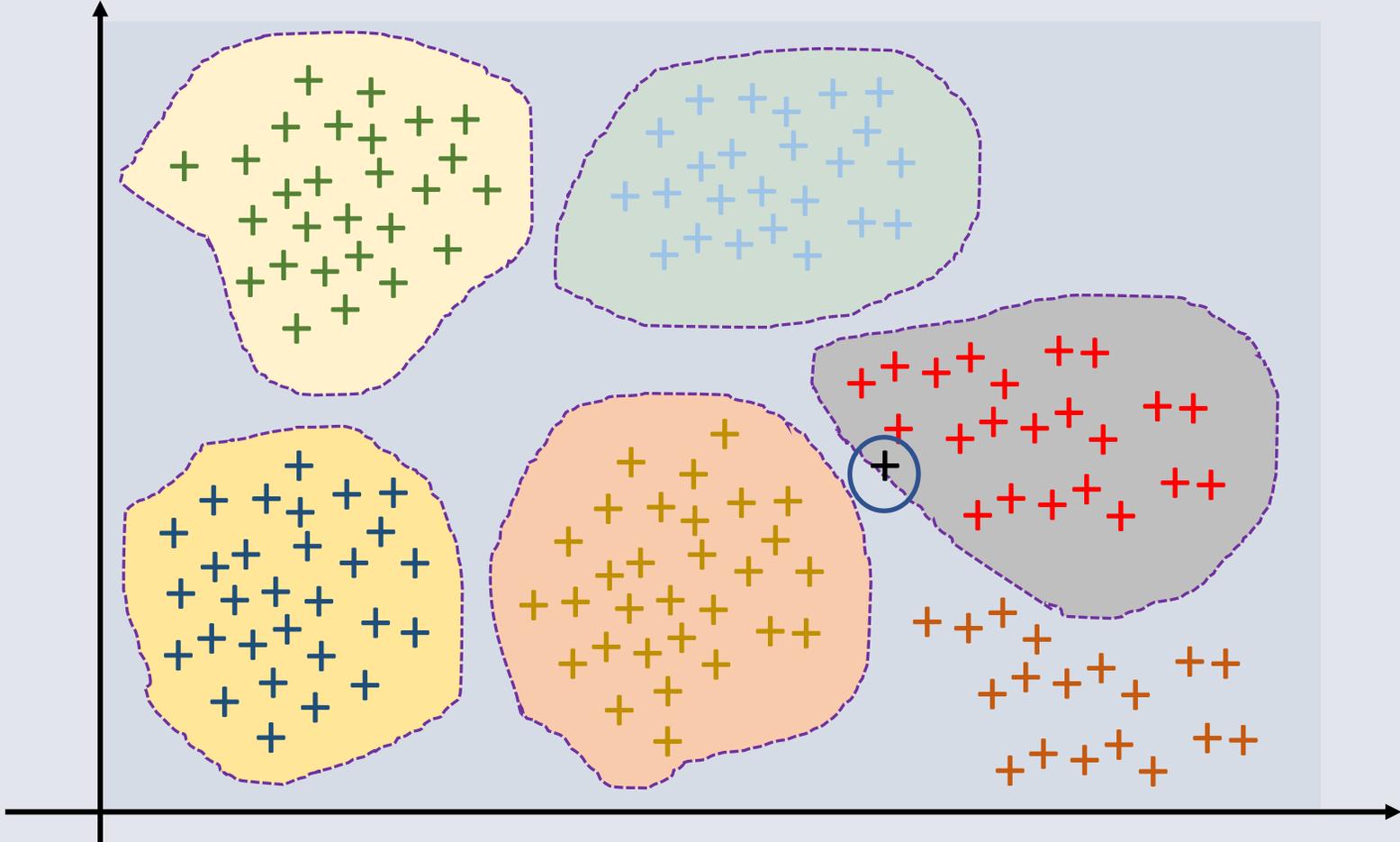
# Support Vector Machine

(Angry, Fear, Happy, Neutral, Sad, Surprise)



## SVM Model for Facial Expression Recognition: One-vs-One approach

Y-axis



Pair (Classifier)	Prediction
Ang Vs Happy	Angry
Ang Vs Sad	Sad
Ang Vs Neutral	Neutral
Ang Vs Fear	Fear
Ang Vs Surprise	Angry
Hap Vs Sad	Sad
Hap Vs Neutral	Neutral
Hap Vs Fear	Happy
Hap Vs Surprise	Surprise
Sad Vs Neutral	Sad
Sad Vs Fear	Sad
Sad Vs Surprise	Sad
Neutral Vs Fear	Fear
Neutral Vs Surprise	Neutral
Fear Vs Surprise	Fear

2

Angry

1

Happy

1

Surprise

X-axis

# Support Vector Machine



## SVM Model for Facial Expression Recognition: One-vs-Many approach

### One classifier per class

We create six separate SVM classifiers.

Each classifier is trained to recognize whether an image **belongs to its specific class or not**

(e.g., Angry vs. not Angry, Happy vs. not Happy).

### Training each classifier

For each classifier, images of the target class are labeled as **positive** (e.g., Angry = 1), while all other

images (of all other expressions) are labeled as **negative** (e.g., not Angry = 0).

Pair (Classifier)
Angry Vs Not Angry
Sad Vs Not Sad
Neutral Vs Not Neutral
Fear Vs Not Fear
Surprise Vs Not Surprise
Happy Vs Not Happy

# Support Vector Machine



## SVM Model for Facial Expression Recognition: One-vs-Many approach

### Example

The Angry classifier is trained to distinguish **Angry** images from any other expression (Fear, Happy, Sad, Neutral, and Surprise).

The Happy classifier is trained to distinguish **Happy** images from any other expression (Angry, Fear, Sad, Neutral, and Surprise), and so forth.

Pair (Classifier)
Angry Vs Not Angry
Sad Vs Not Sad
Neutral Vs Not Neutral
Fear Vs Not Fear
Surprise Vs Not Surprise
Happy Vs Not Happy

# Support Vector Machine



## SVM Model for Facial Expression Recognition: One-vs-Many approach

### Testing New Image

- Each classifier independently decides whether the image belongs to **its specific class** (e.g., Angry or not Angry, Happy or not Happy).
- Each classifier produces a confidence score or distance from the decision boundary for its prediction:
  - A high positive score suggests a strong likelihood that the image belongs to the target class (e.g., the Angry classifier predicts a high score for an image it thinks is Angry).
  - A negative or lower score suggests the image does not belong to that class.

Pair (Classifier)
Angry Vs Not Angry
Sad Vs Not Sad
Neutral Vs Not Neutral
Fear Vs Not Fear
Surprise Vs Not Surprise
Happy Vs Not Happy

# Support Vector Machine



## SVM Model for Facial Expression Recognition: One-vs-Many approach

### Selecting the Final Prediction

Each classifier will produce its score where we:

- Compare the confidence score from all classifiers.
- One with highest confidence score is the final predicted class.

Angry: -0.5

Fear: 0.2

Happy: -2.1

**Sad: 2.8**

Neutral: 0.5

Surprise: 0.1

Pair (Classifier)	
Angry Vs Not Angry	-0.5
Sad Vs Not Sad	2.8
Neutral Vs Not Neutral	0.5
Fear Vs Not Fear	0.2
Surprise Vs Not Surprise	0.1
Happy Vs Not Happy	-2.1

Here, **Sad** has the highest score (2.8), so the model classifies the image as **Sad**

# Support Vector Machine



## SVM Model for Facial Expression Recognition: One-vs-Many approach

Let's say we are predicting the expression for an image where the actual emotion is "Sad."

**How??**

Each classifier scores the image: The Angry classifier gives a low score because it doesn't detect typical Angry features.

The Sad classifier, seeing a eyebrows/lashes down or other "Sad" features, gives a high score.

Other classifiers (Surprise, Fear, etc.) might give neutral or low scores if their features don't align with the image.

Since the Sad classifier has the highest score, the model predicts Sad.

# Support Vector Machine



## SVM Model for Facial Expression Recognition: Comparison

Aspect	One-vs-All	One vs One
# of Classes	Same as of # of classes	Possible pairs of classes
Process	Highest confidence score amongst all the classifier	Voting
Training	Classifier see all the classes	See only two classes

# Coming up



- **Supervised Learning Algorithms – Classification**
  - Decision Trees
  - Random Forest
- **Unsupervised Learning Algorithms – Clustering**
  - K-Means Clustering
  - Principal Component Analysis (PCA)



**Any Question?**