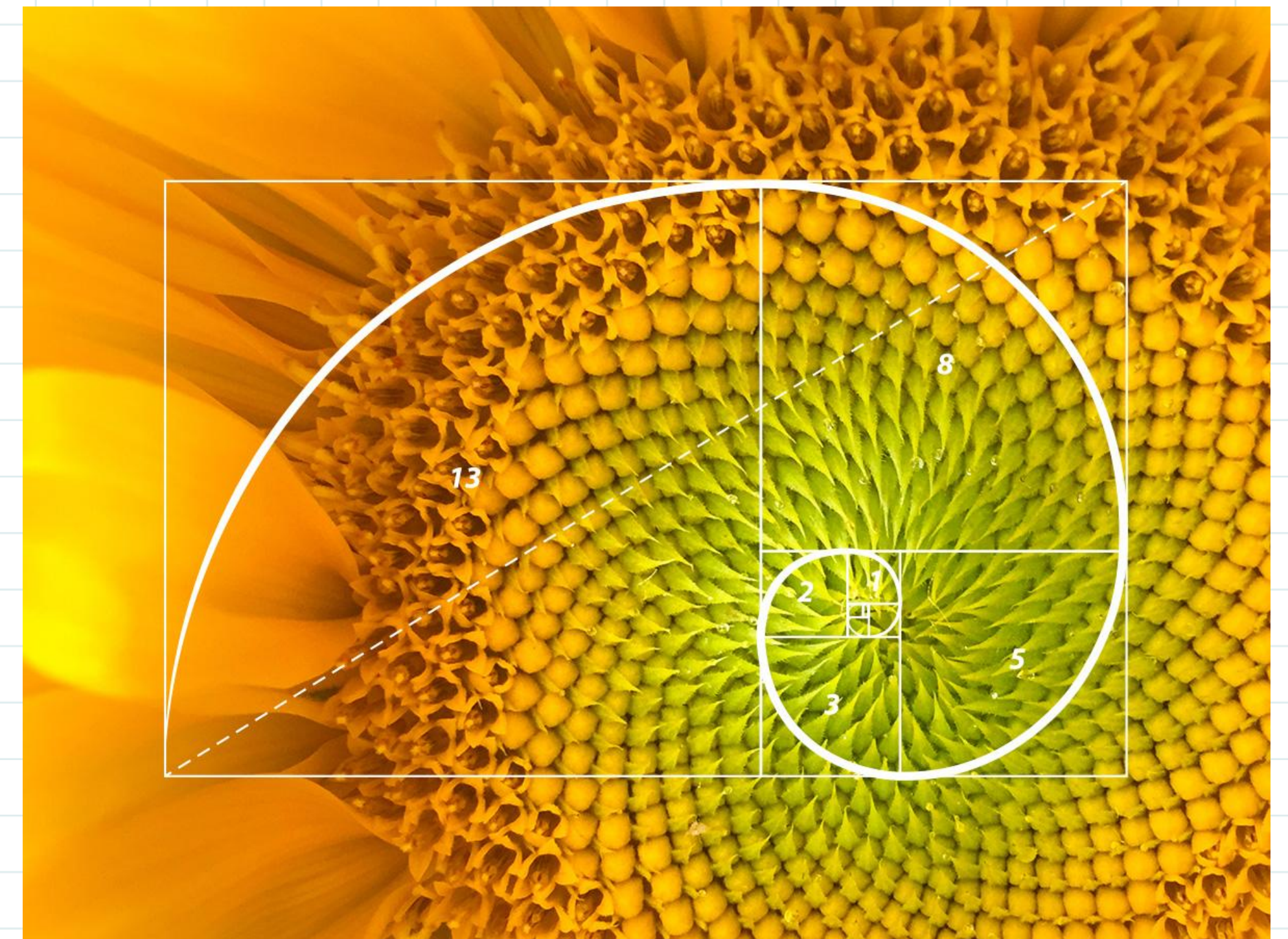


MACHINE LEARNING

Support Vector Machines (SVM)

Dr. Sirojul Munir, S.Si., M.Kom.
rojulman@nurulfikri.ac.id

ARTIFICIAL INTELLIGENCE – INFORMATICS STTNF



Daur ulang Project Data Science



Algoritma Klasifikasi

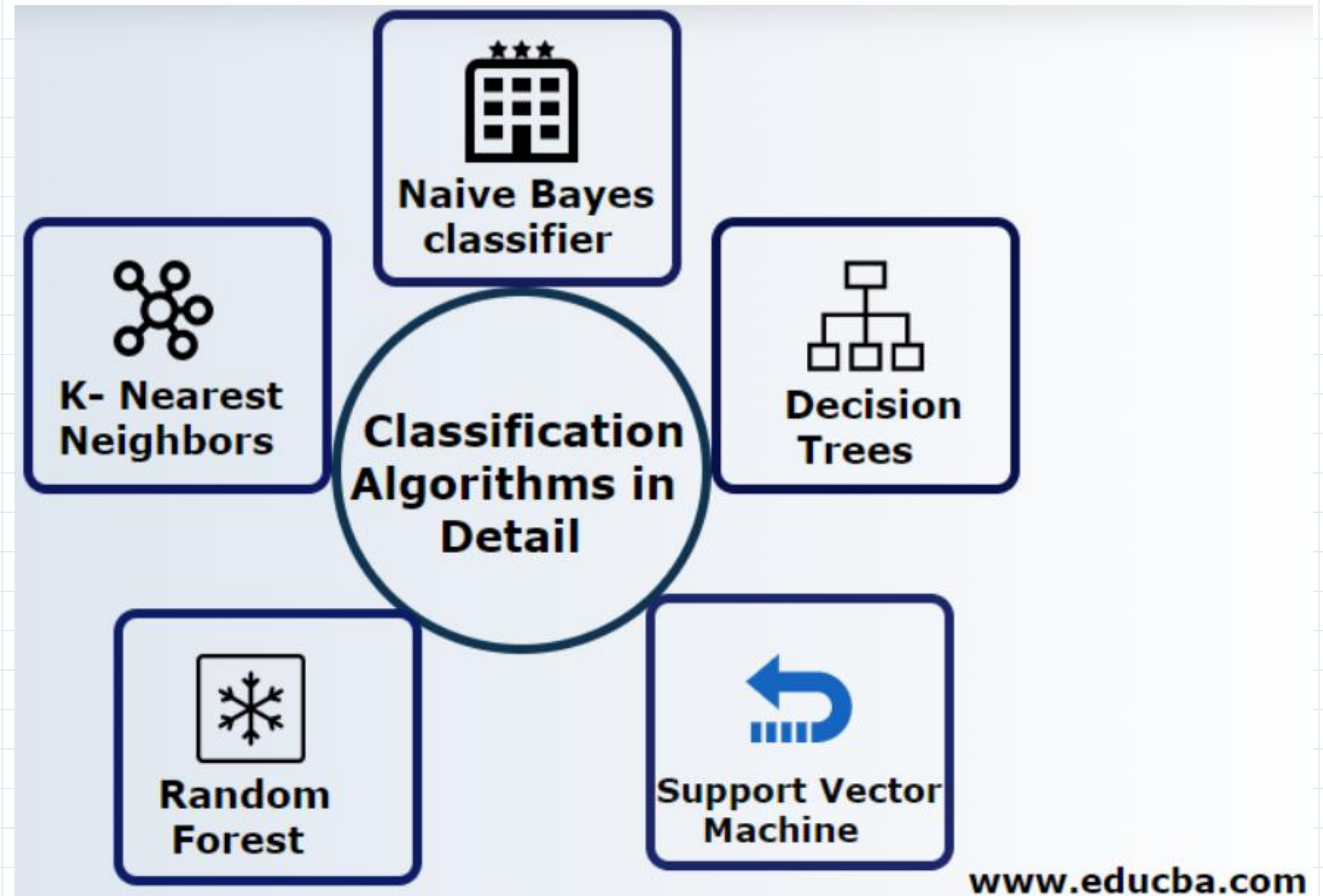
1.K- Nearest Neighbors

2.Naive Bayes classifier

3.Decision Trees

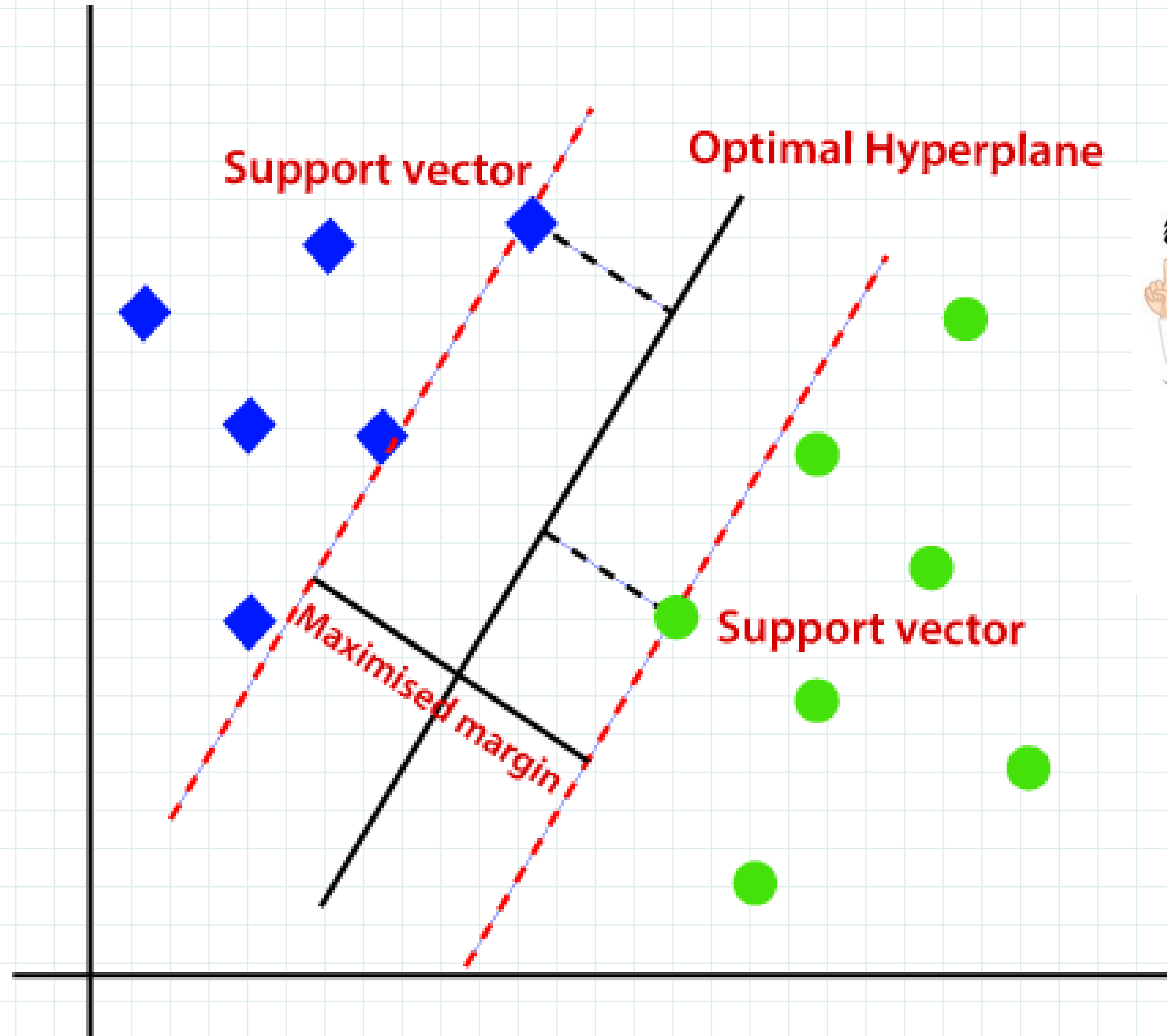
4.Support Vector Machine

5.Random Forest



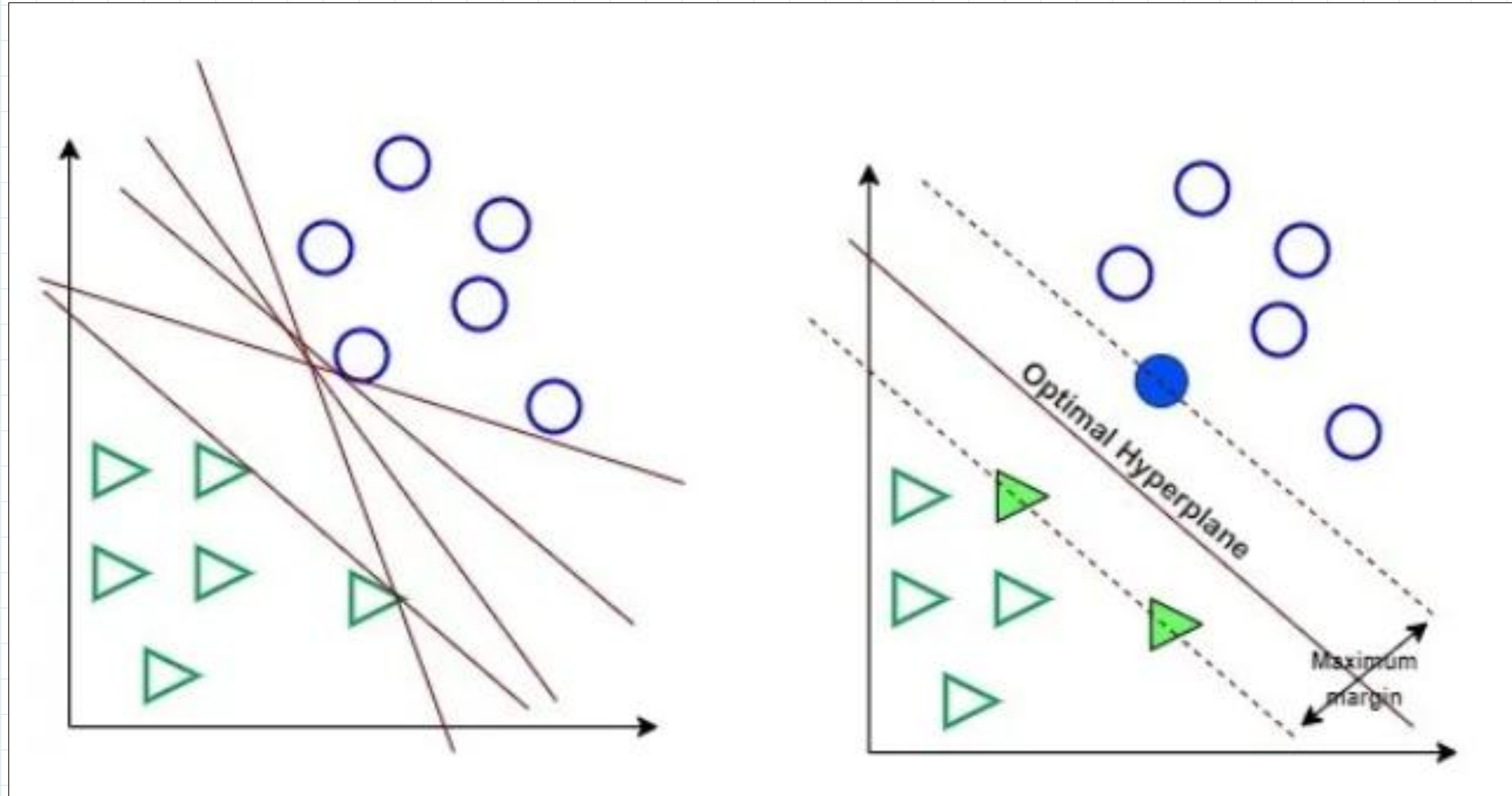
Support Vector Machines

- Klasifikasi data dengan menemukan **garis optimal** atau **hyperplane** terbaik yang **memaksimalkan jarak** antara setiap **kelas** dalam ruang **N-dimensi**
- Vladimir N Vapnik, 1992
Columbia University, USA
- SVM: mode ML Terbimbing untuk solusi masalah **klasifikasi, regresi** dan **deteksi outlier**

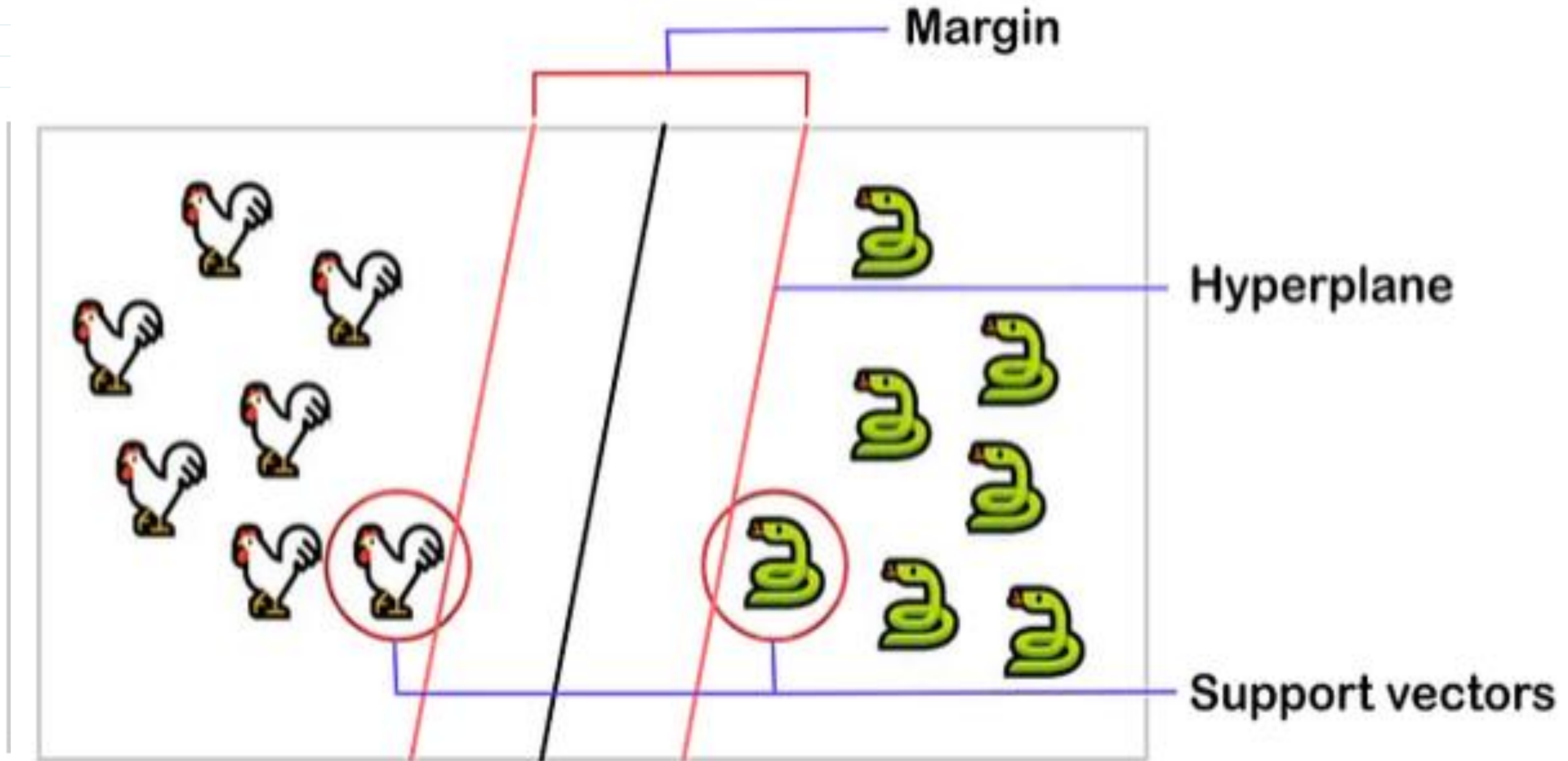
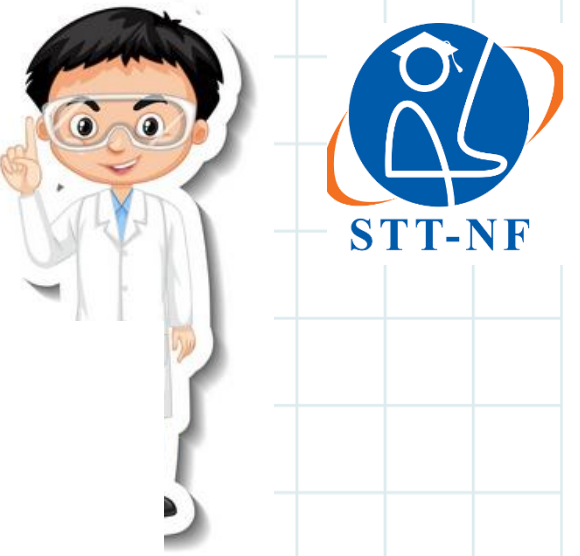


Support Vector Machines - Hyperplane

- Menemukan **hyperplane** terbaik dalam ruang berdimensi-N (N-jumlah fitur) sebagai pemisah data yang jelas bagi titik-titik data input

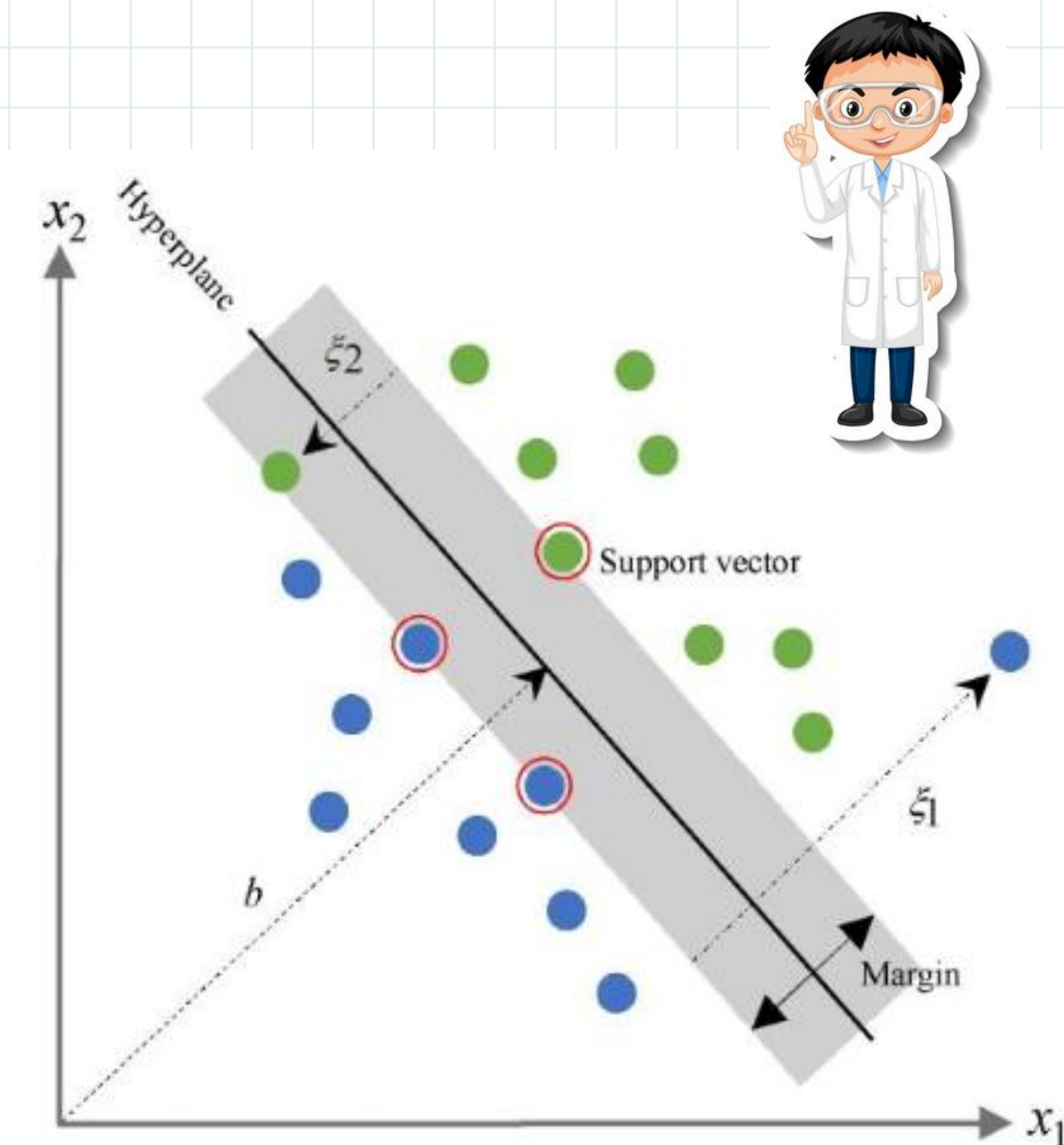


Support Vector Machines - Classifier



Support Vector Machines - Classifier

1)



Sebuah hyperplane dalam ruang berdimensi-n dapat dinyatakan sebagai:

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

di mana:

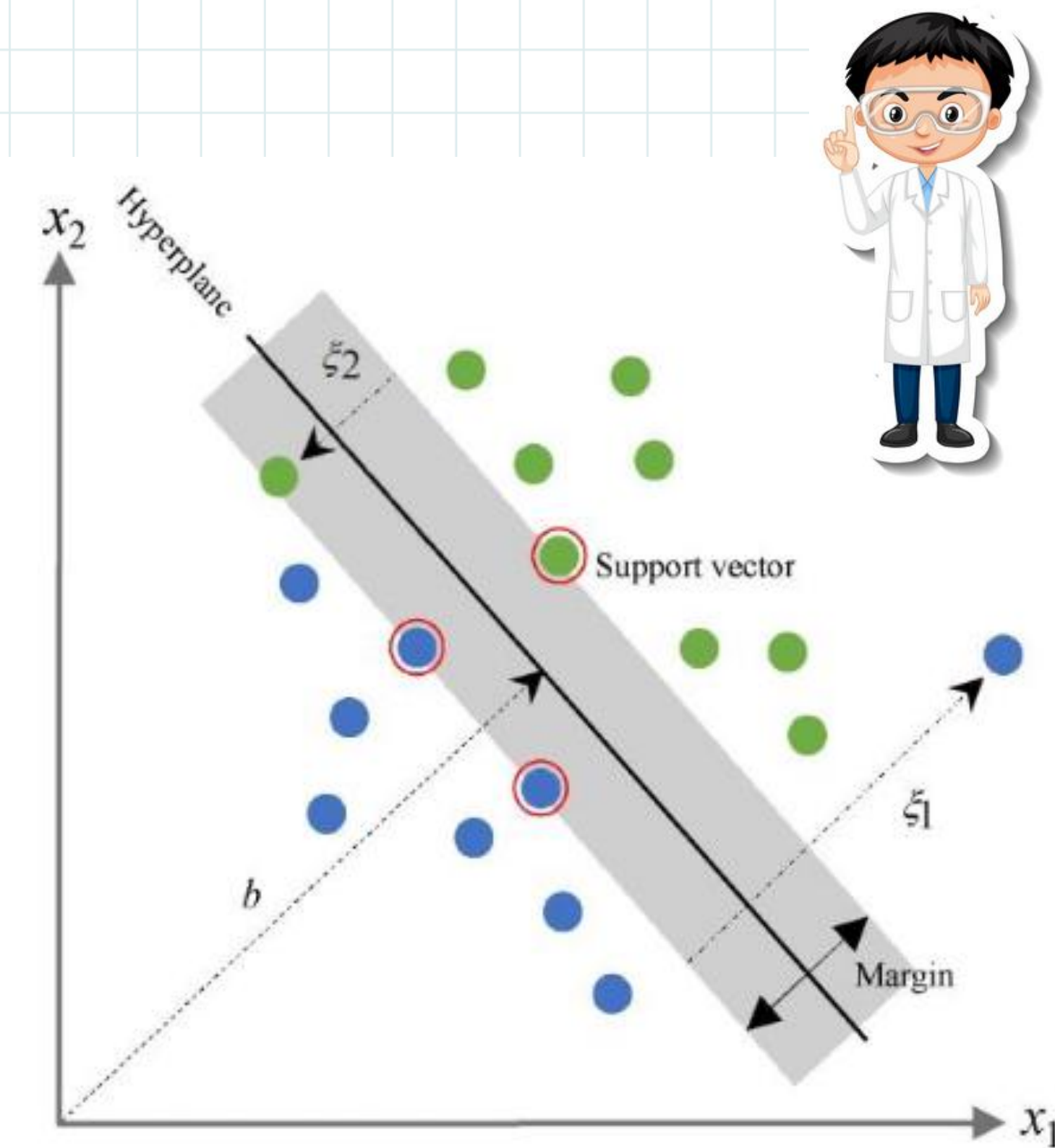
- w = vektor bobot (weight vector)
- x = vektor fitur (data)
- b = bias (intercept)

Hyperplane ini akan memisahkan dua kelas data:

- Jika $w^T x + b > 0$, maka x diklasifikasikan sebagai kelas +1
- Jika $w^T x + b < 0$, maka x diklasifikasikan sebagai kelas -1

Support Vector Machines - Classifier

2)



SVM berusaha memaksimalkan jarak (margin) antara dua kelas. Dua hyperplane pembatas (boundary) dapat ditulis sebagai:

$$w^T x + b = +1$$

$$w^T x + b = -1$$

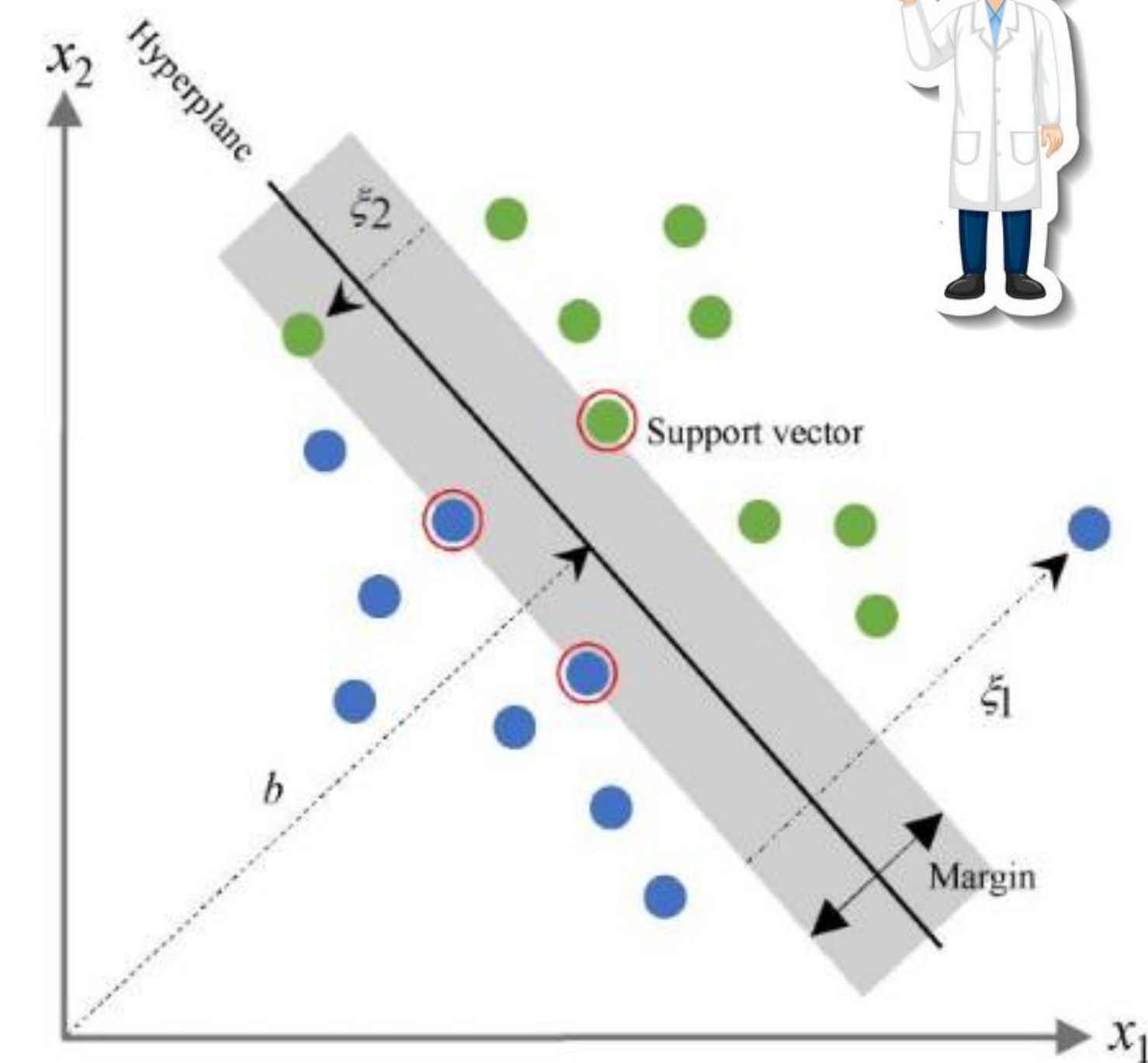
Margin adalah jarak antara kedua hyperplane tersebut:

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

Untuk mendapatkan margin terbesar, kita meminimalkan $||w||$ dengan tetap menjaga data tetap terpisah dengan benar.

Support Vector Machines - Classifier

3)



Untuk data yang separabel secara linear, kita dapat merumuskan masalah optimasi sebagai:

$$\min_{w,b} \frac{1}{2} ||w||^2$$

Dengan kendala (constraint):

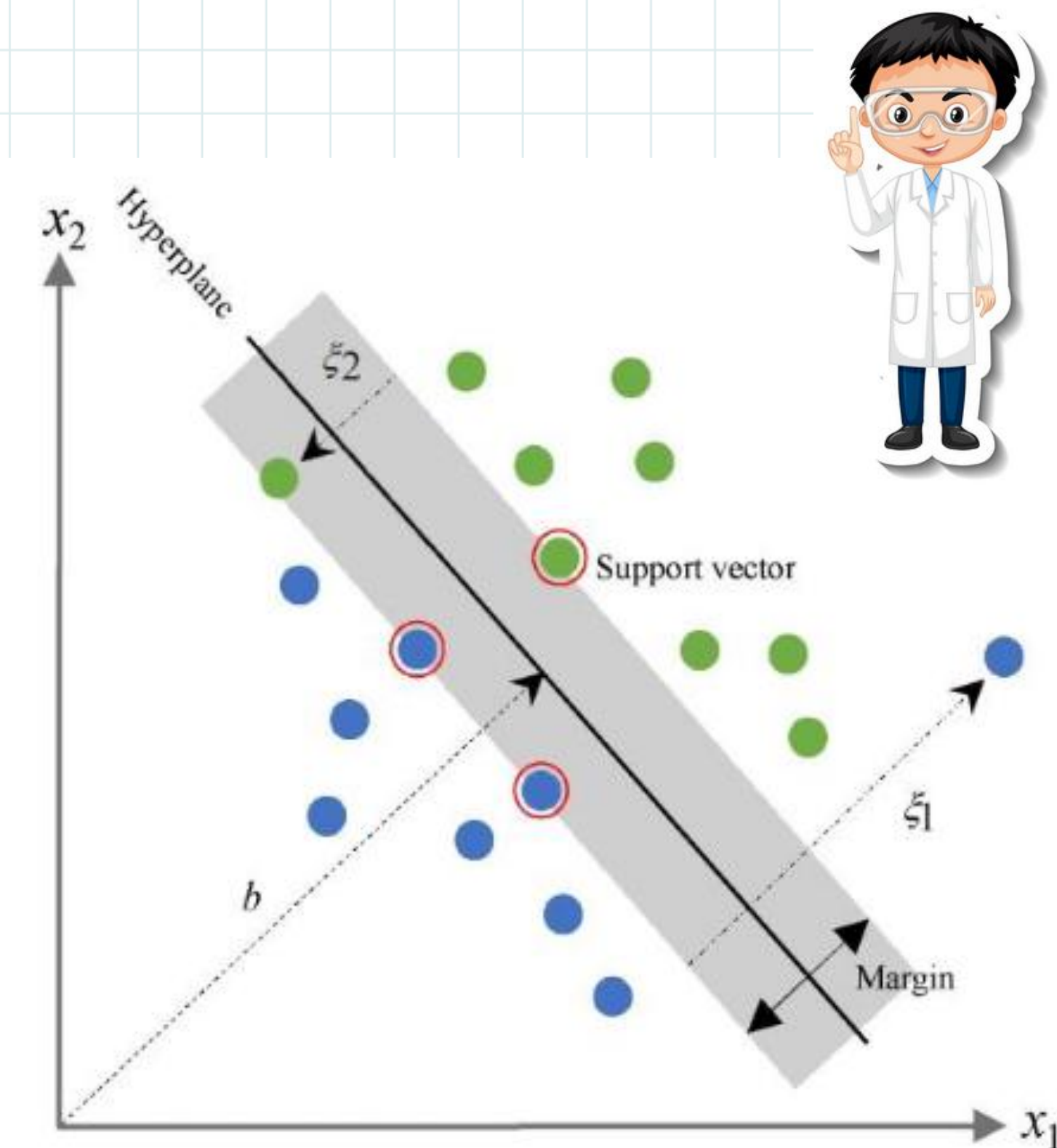
$$y_i(w^T x_i + b) \geq 1, \quad \forall i = 1, 2, \dots, N$$

di mana:

- $y_i \in \{+1, -1\}$ adalah label kelas
- x_i adalah data ke-i

Support Vector Machines - Classifier

4)



Jika data **tidak sepenuhnya terpisah (non-separable)**, digunakan variabel slack $\xi_i \geq 0$ untuk mengizinkan pelanggaran margin:

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

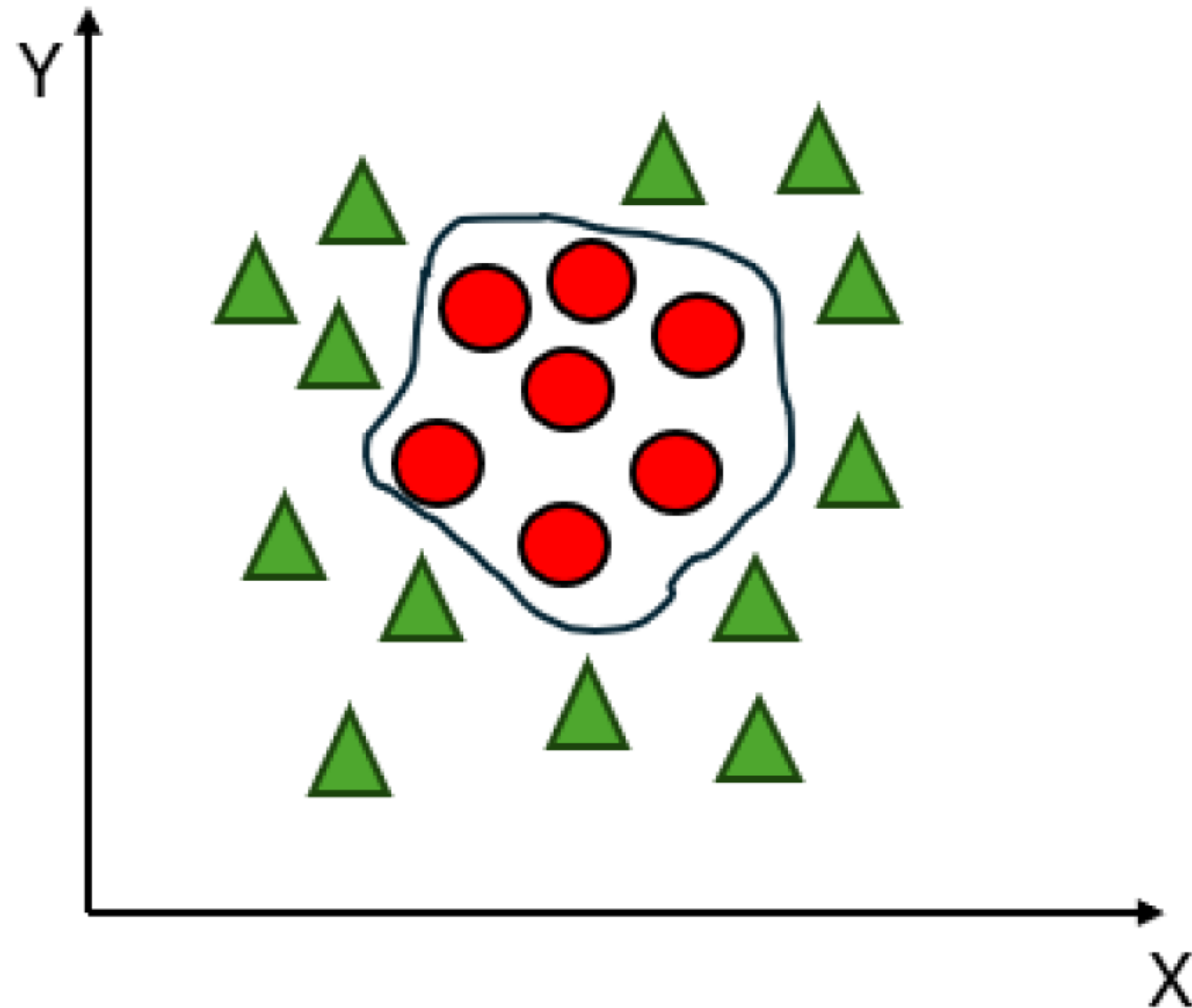
Dengan kendala:

$$y_i(w^T x_i + b) \geq 1 - \xi_i, \quad \xi_i \geq 0$$

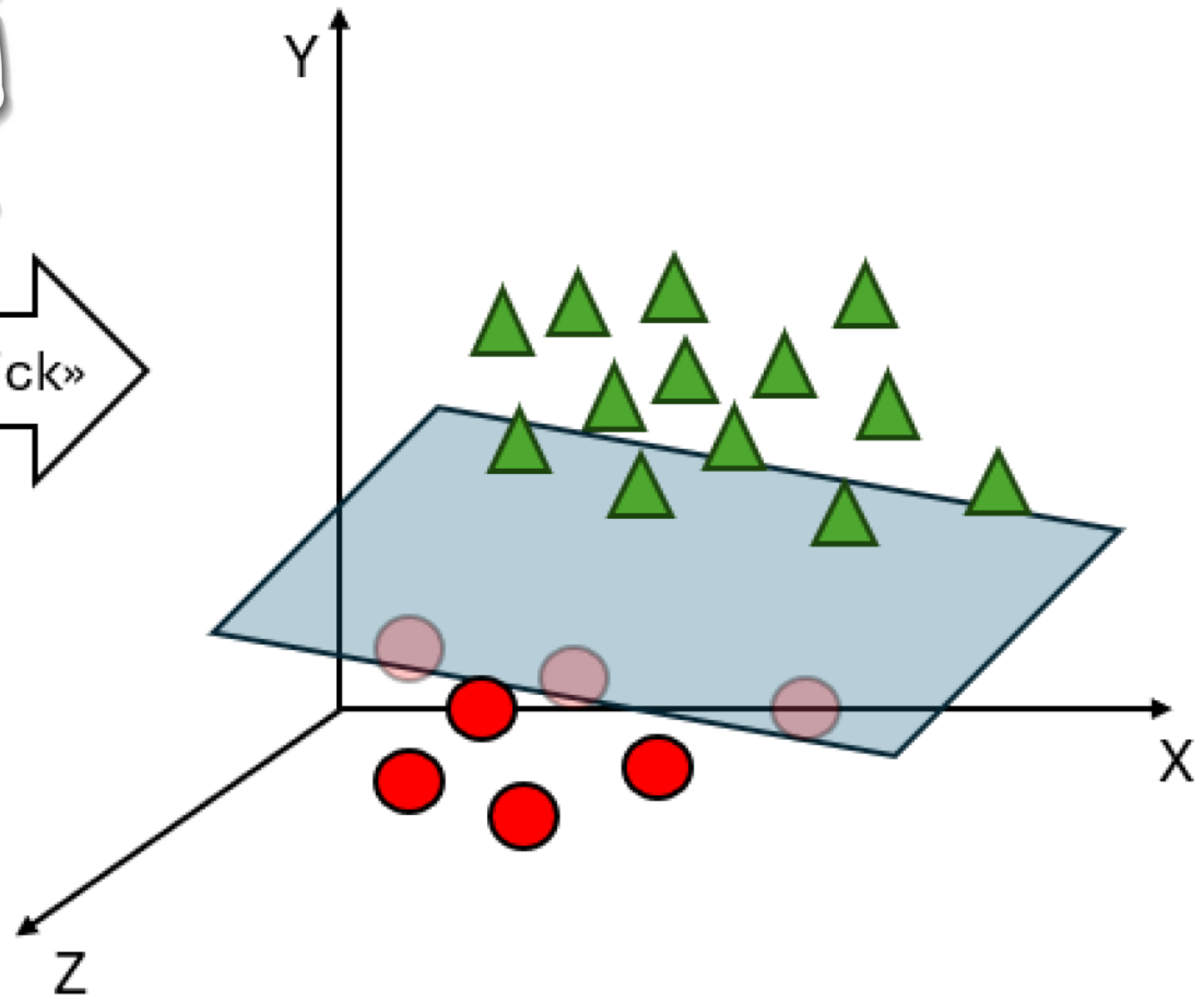
Konstanta C adalah **parameter regularisasi** yang mengatur trade-off antara margin yang lebar dan kesalahan klasifikasi.

Support Vector Machines - Kernel

Bagaimana jika data tidak linear ?

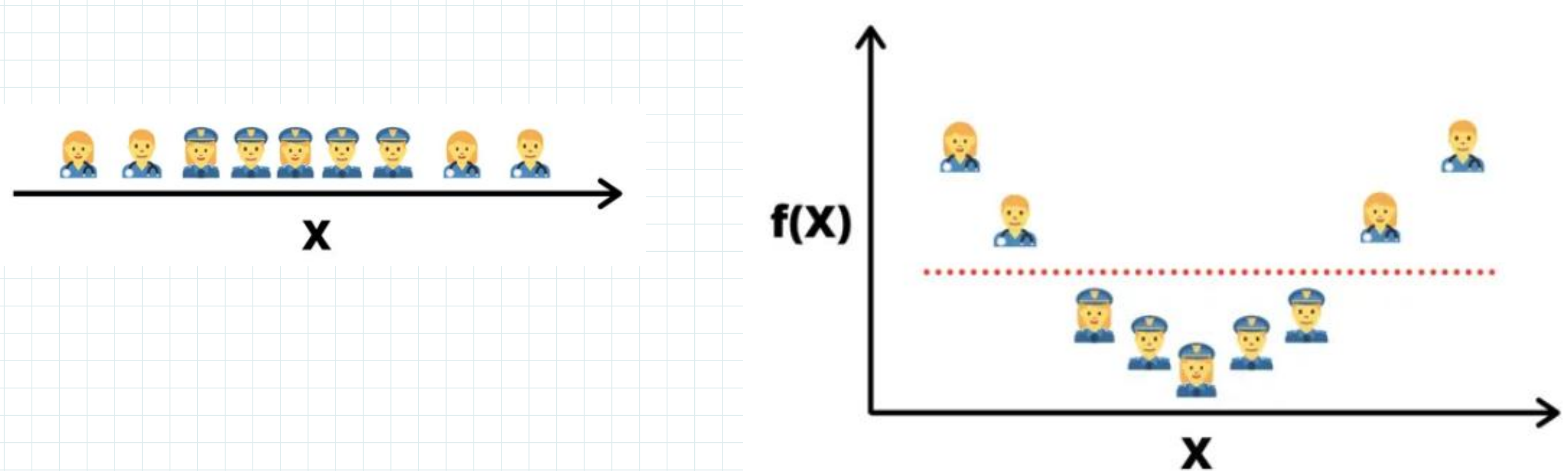
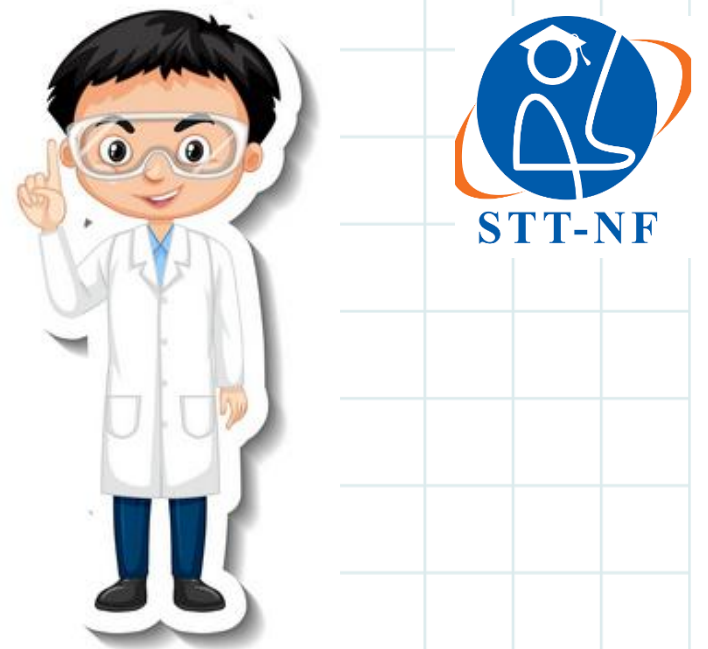


«Kernel trick»



Support Vector Machines - Kernel

Bagaimana jika data tidak linear ?

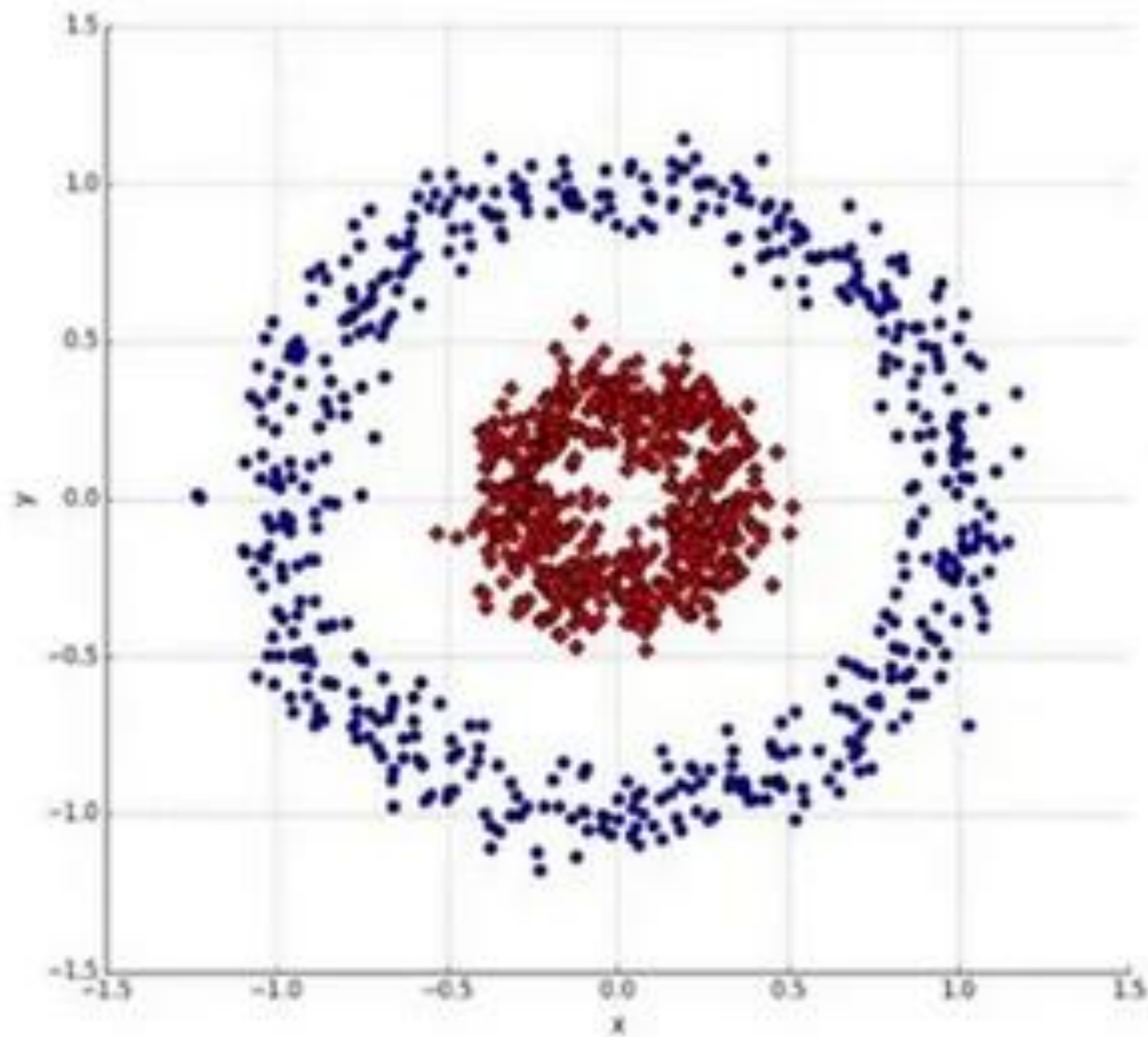


Support Vector Machines - Kernel

Bagaimana jika data tidak linear ?

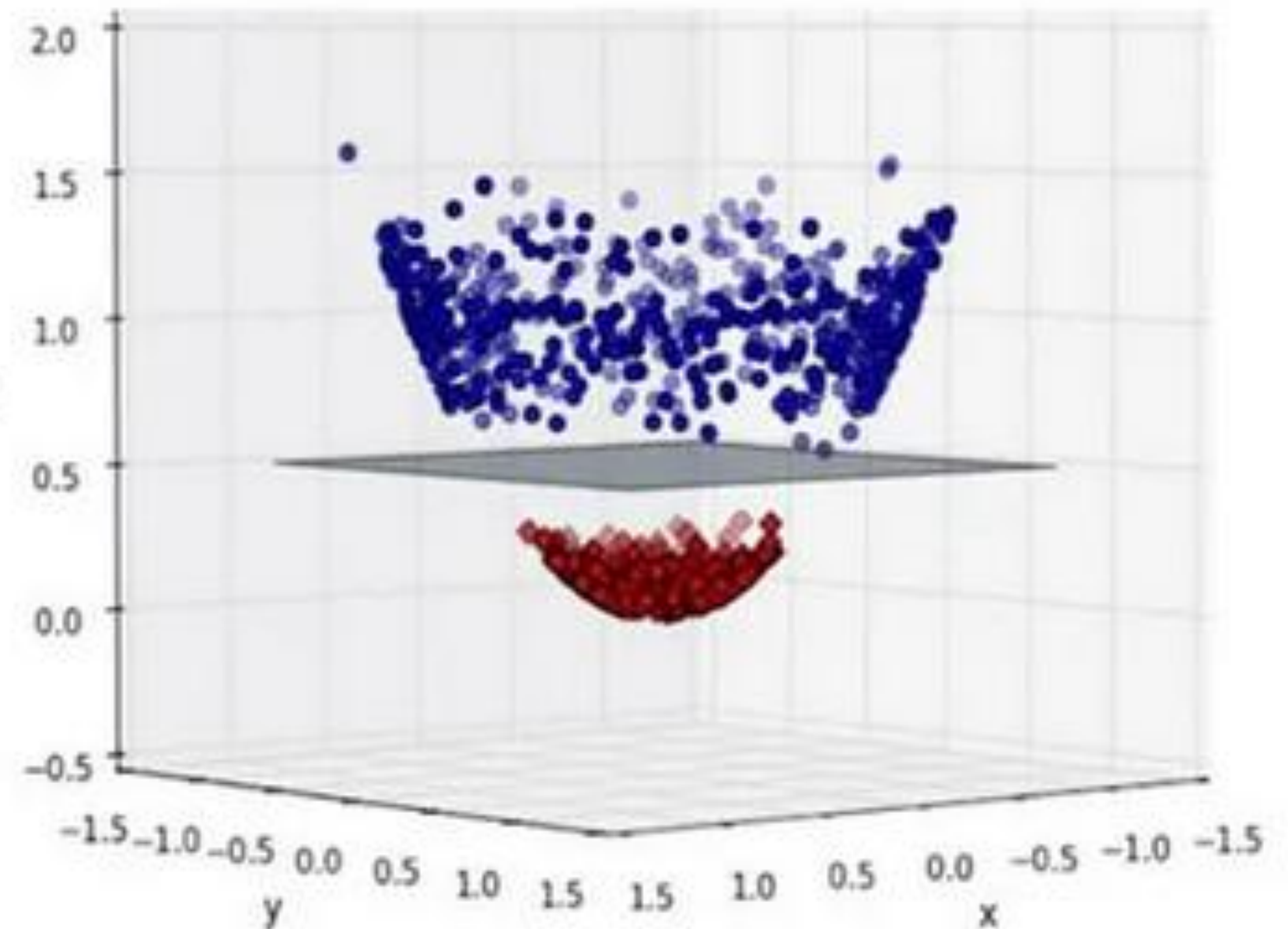


2D



Kernel

3D



Support Vector Machines - Kernel



Untuk kasus non-linear, data dipetakan ke ruang fitur berdimensi lebih tinggi menggunakan fungsi kernel $K(x_i, x_j)$:

$$K(x_i, x_j) = \phi(x_i)^T \phi(x_j)$$

Beberapa kernel yang umum digunakan:

- Linear kernel: $K(x_i, x_j) = x_i^T x_j$
- Polynomial kernel: $K(x_i, x_j) = (x_i^T x_j + 1)^d$
- RBF kernel (Gaussian): $K(x_i, x_j) = \exp(-\gamma ||x_i - x_j||^2)$

Setelah menggunakan kernel, fungsi keputusan menjadi:

$$f(x) = \text{sign} \left(\sum_{i=1}^N \alpha_i y_i K(x_i, x) + b \right)$$

Support Vector Machines - Kernel



Dalam bentuk dual, masalah optimasi dapat ditulis:

$$\max_{\alpha} \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$

dengan kendala:

$$\sum_{i=1}^N \alpha_i y_i = 0, \quad 0 \leq \alpha_i \leq C$$

Hanya data dengan $\alpha_i > 0$ yang menjadi support vectors.

Support Vector Machines - Kernel



Fungsi Kernel:

1. Linear
2. RBF (Radial Basis Function) / Gaussian Kernel
3. Polinomial
4. Sigmoid

Klasifikasi Bunga Iris

iris setosa



petal sepal

iris versicolor

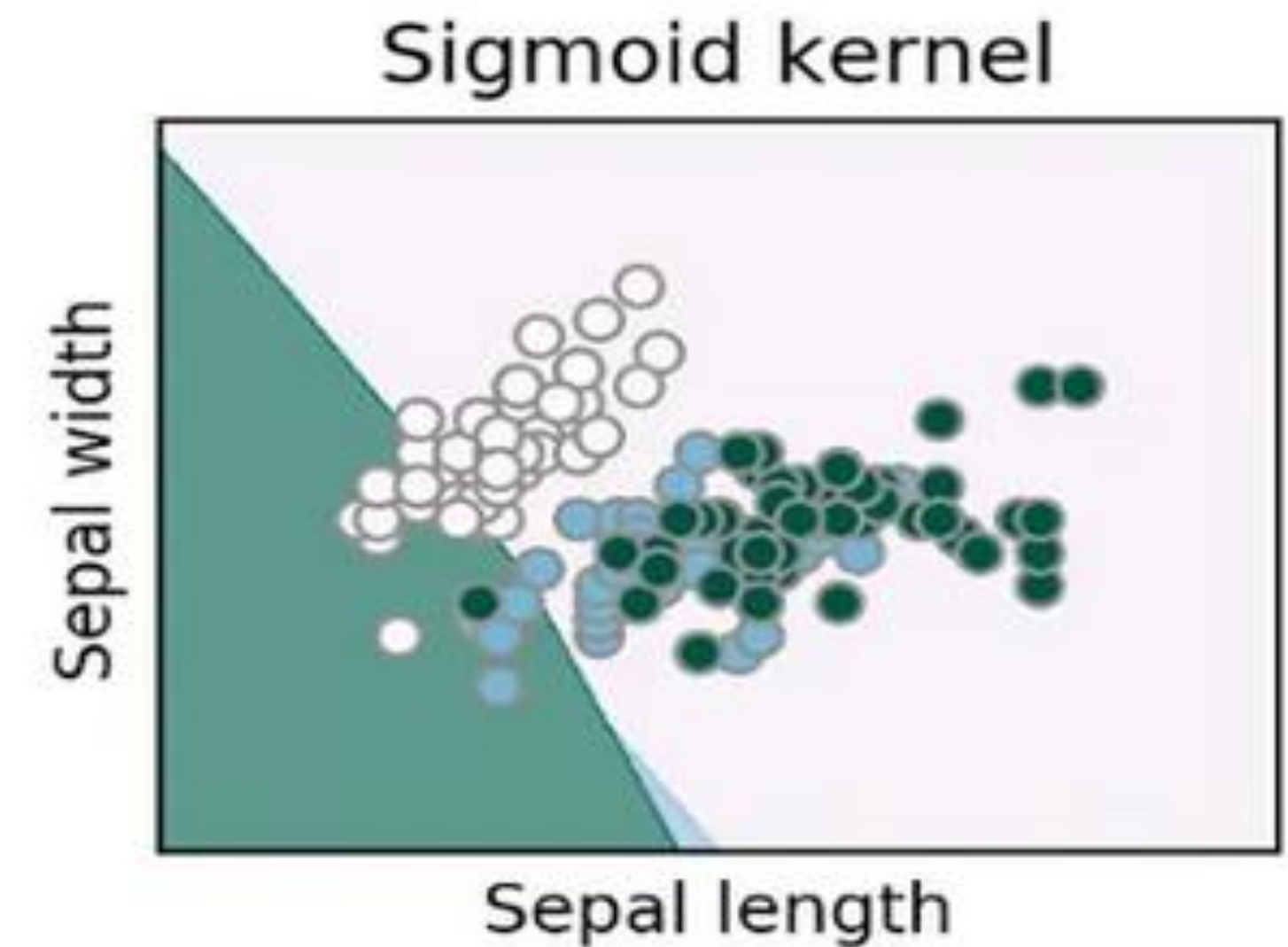
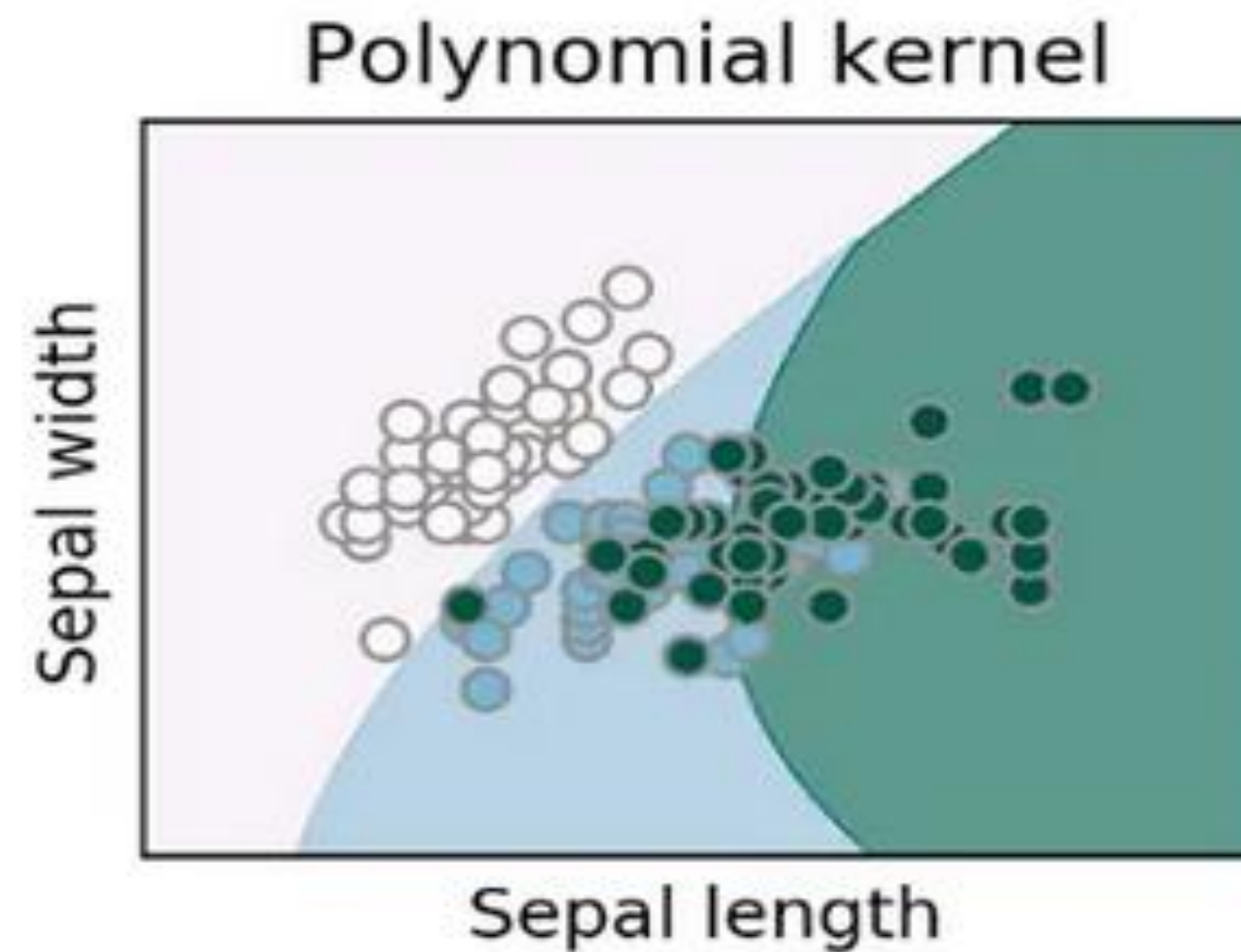
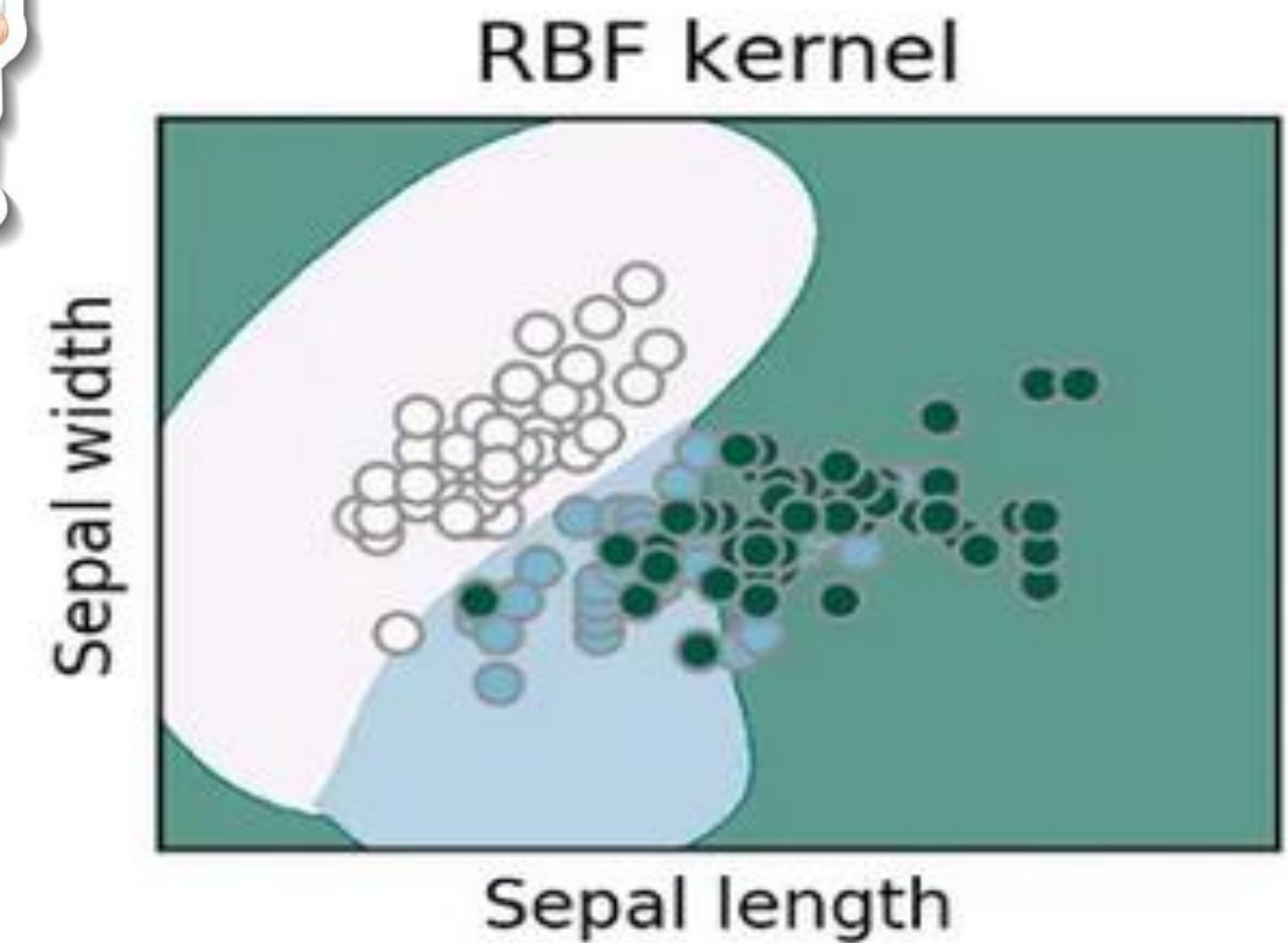
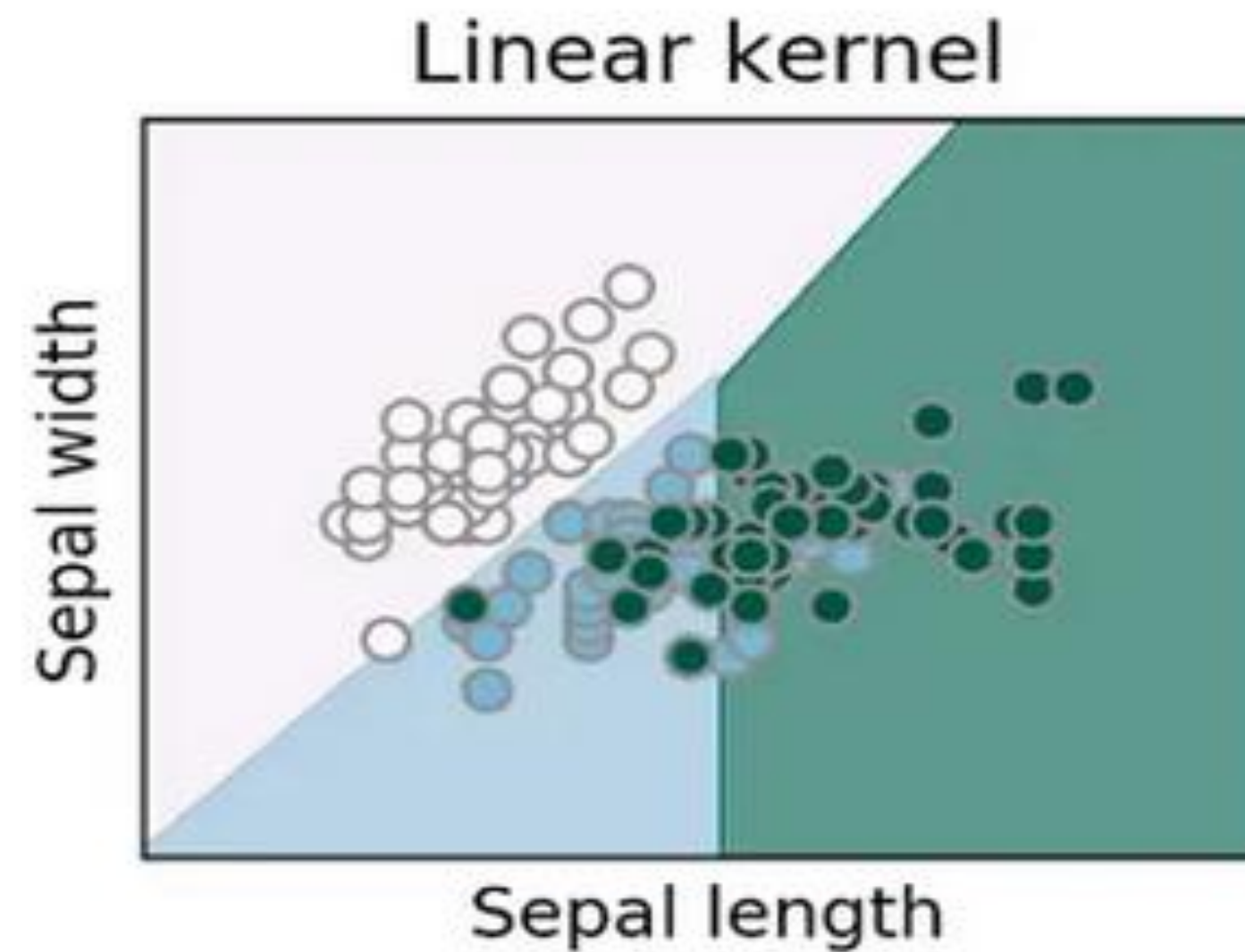


petal sepal

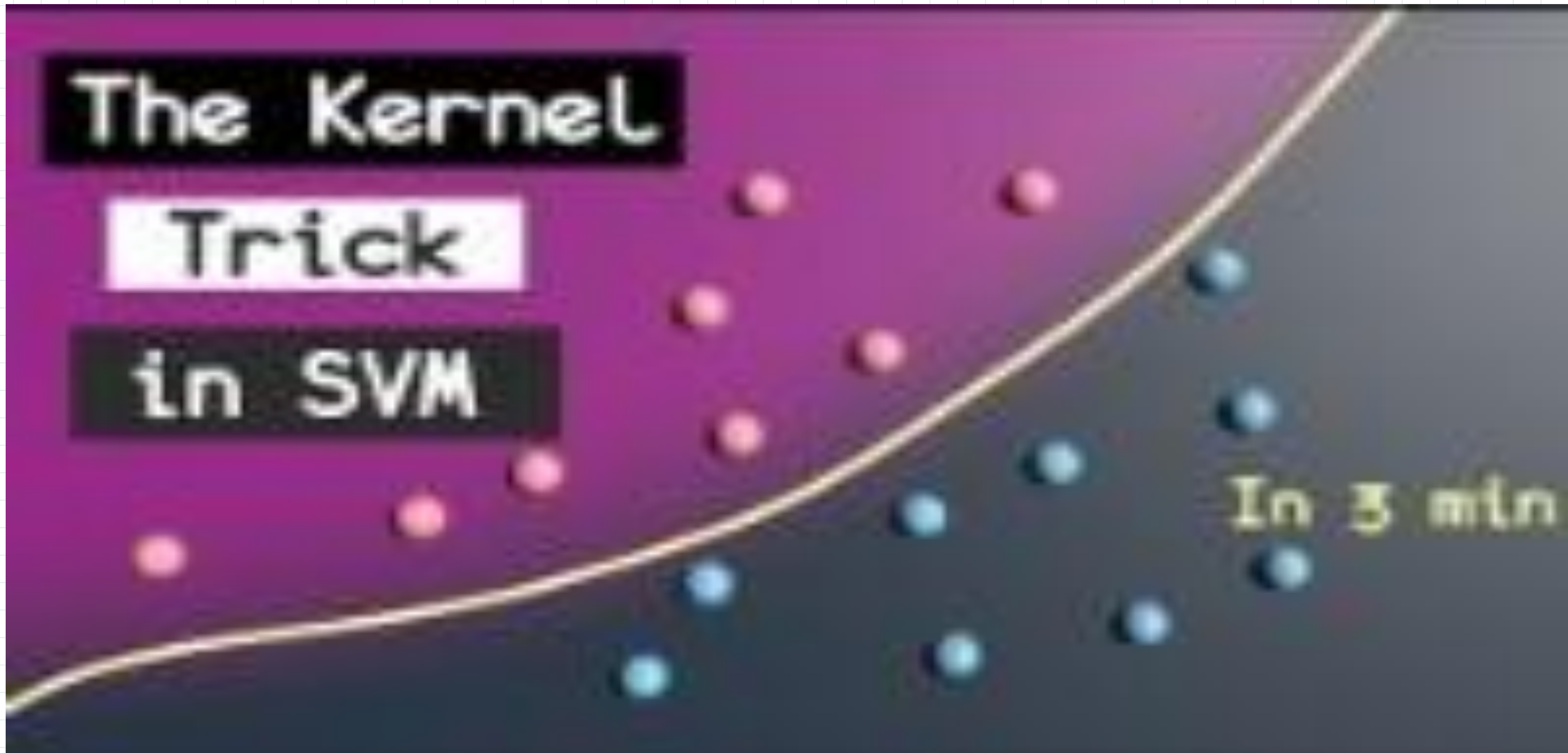
iris virginica



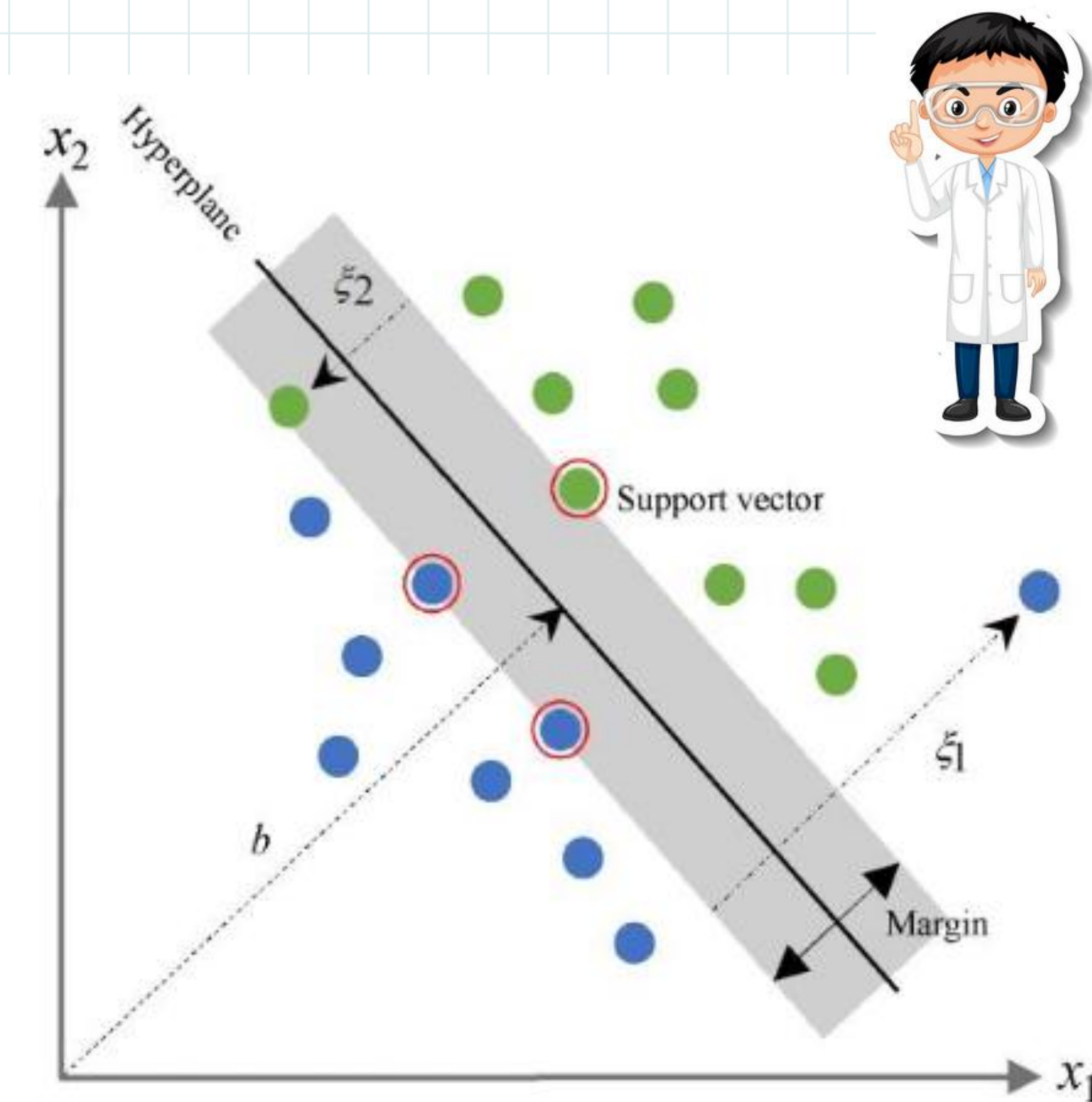
petal sepal



Video – SVM Kernel Trick



Support Vector Machines - Keunggulan



1. SVM efektif pada data berdimensi tinggi (data dengan jumlah fitur atau atribut yang sangat banyak).
2. SVM efektif pada kasus di mana jumlah fitur pada data lebih besar dari jumlah sampel.
3. SVM menggunakan subset poin pelatihan dalam fungsi keputusan (disebut support vector) sehingga membuat penggunaan memori menjadi lebih efisien.

Support Vector Machines – Case Study : iris flowers



iris setosa



petal

sepal

iris versicolor



petal

sepal

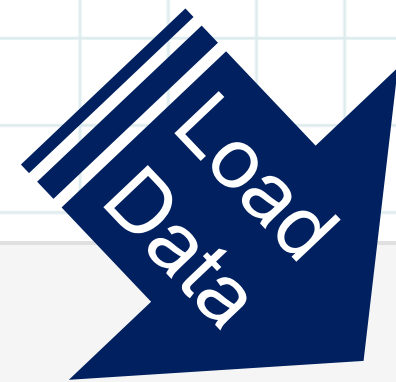
iris virginica



petal

sepal

Support Vector Machines – Case Study : iris flowers



```
import pandas as pd
```

```
# Read the CSV file with a comma delimiter
df = pd.read_csv('../data/iris/iris.csv', sep=',')

# cetak header data (5 baris data) dari file
df.head()
```



	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm	Species
0	1	5.1	3.5	1.4	0.2	Iris-setosa
1	2	4.9	3.0	1.4	0.2	Iris-setosa
2	3	4.7	3.2	1.3	0.2	Iris-setosa
3	4	4.6	3.1	1.5	0.2	Iris-setosa
4	5	5.0	3.6	1.4	0.2	Iris-setosa

```
[2]: df.describe()
```

```
[2]:
```

	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	75.500000	5.843333	3.054000	3.758667	1.198667
std	43.445368	0.828066	0.433594	1.764420	0.763161
min	1.000000	4.300000	2.000000	1.000000	0.100000
25%	38.250000	5.100000	2.800000	1.600000	0.300000
50%	75.500000	5.800000	3.000000	4.350000	1.300000
75%	112.750000	6.400000	3.300000	5.100000	1.800000
max	150.000000	7.900000	4.400000	6.900000	2.500000

```
[3]: df["Species"].unique()
```

```
[3]: array(['Iris-setosa', 'Iris-versicolor', 'Iris-virginica'], dtype=object)
```

```
X = df[['SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm']]
```

```
# Kolom target (label)
```

```
y = df['Species']
```

**Tentukan Variabel
Target & Fitur**

Support Vector Machines – Case Study : iris flowers

```
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
# Membuat model SVM dengan kernel linear
model = SVC(kernel='linear')
model.fit(X_train, y_train)
```

```
y_pred = model.predict(X_test)
# Akurasi
print(f"Akurasi: {accuracy_score(y_test, y_pred) * 100:.2f}%")
# Laporan klasifikasi
print("\nLaporan Klasifikasi:\n", classification_report(y_test, y_pred))
```

Akurasi: 100.00%

Laporan Klasifikasi:

	precision	recall	f1-score	support
Iris-setosa	1.00	1.00	1.00	10
Iris-versicolor	1.00	1.00	1.00	9
Iris-virginica	1.00	1.00	1.00	11
accuracy			1.00	30
macro avg	1.00	1.00	1.00	30
weighted avg	1.00	1.00	1.00	30



SVC		
Parameters		
C		1.0
kernel		'linear'
degree		3
gamma		'scale'
coef0		0.0
shrinking		True
probability		False
tol		0.001
cache_size		200
class_weight		None
verbose		False
max_iter		-1
decision_function_shape		'ovr'
break_ties		False
random_state		None

Support Vector Machines – Case Study : iris flowers

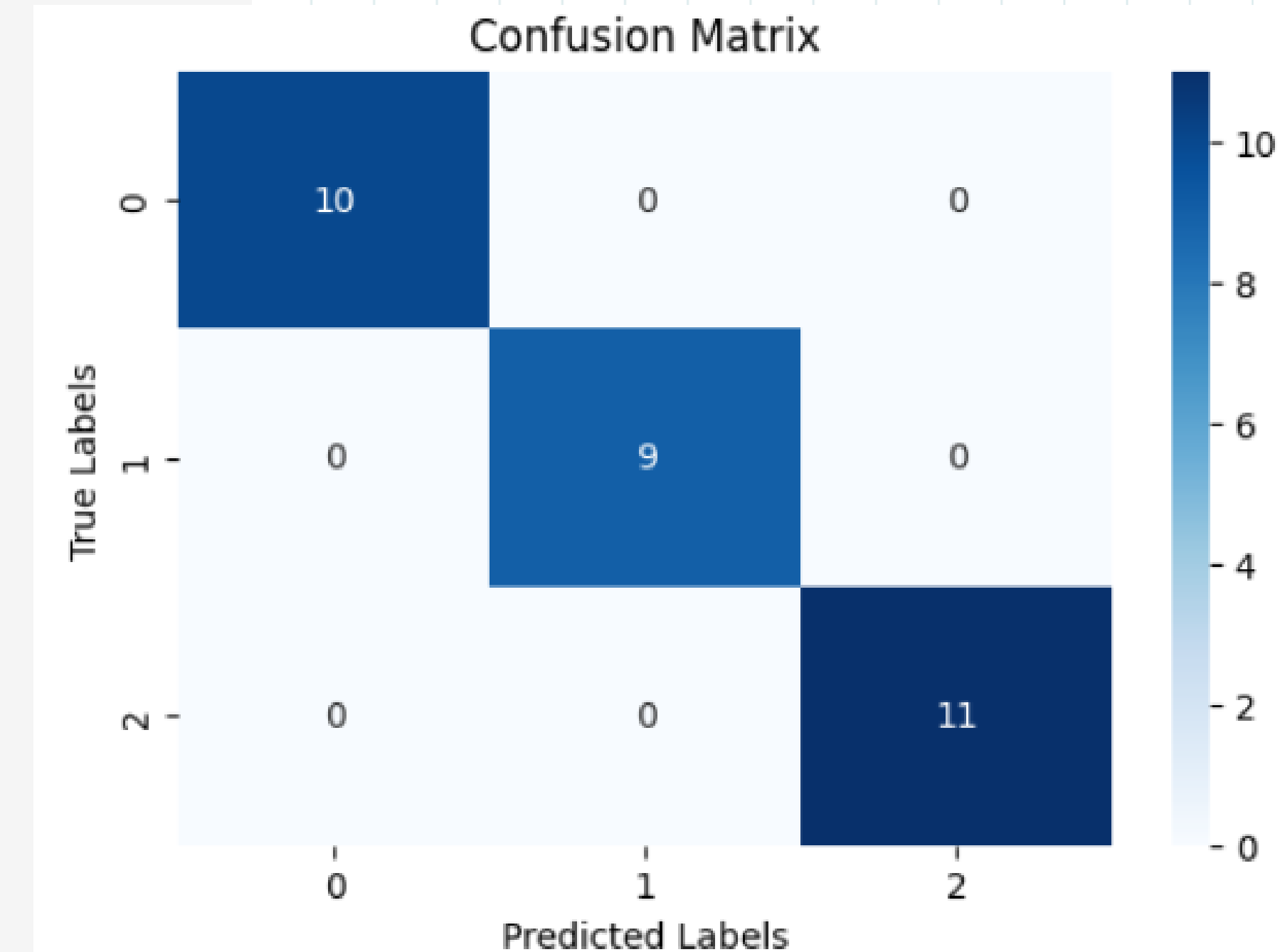
```
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay
import matplotlib.pyplot as plt
import seaborn as sns

print("\nConfusion Matrix:\n", confusion_matrix(y_test, y_pred))

# Buat confusion matrix
cm = confusion_matrix(y_test, y_pred)

# Jika kamu tahu nama kelas (opsional, agar lebih informatif)
# misalnya: class_names = ['Negatif', 'Positif']
# maka tambahkan ke heatmap di bagian "xticklabels" dan "yticklabels"

plt.figure(figsize=(6,4))
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')
plt.title("Confusion Matrix")
plt.xlabel("Predicted Labels")
plt.ylabel("True Labels")
plt.show()
```

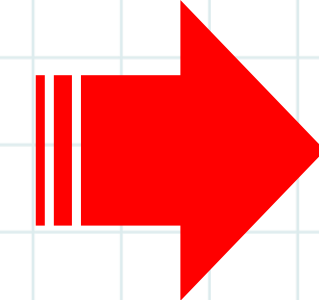
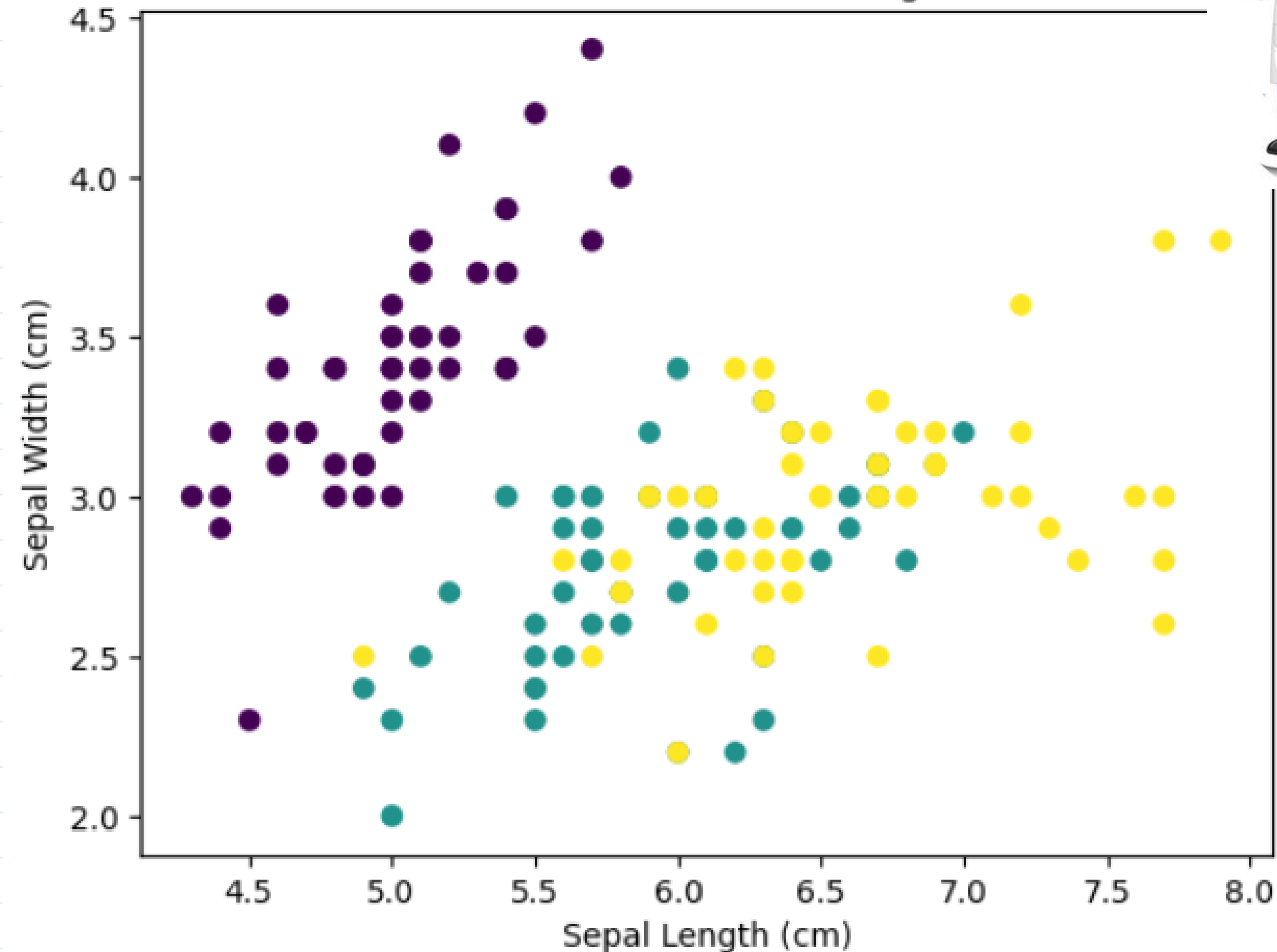


Support Vector Machines – Case Study

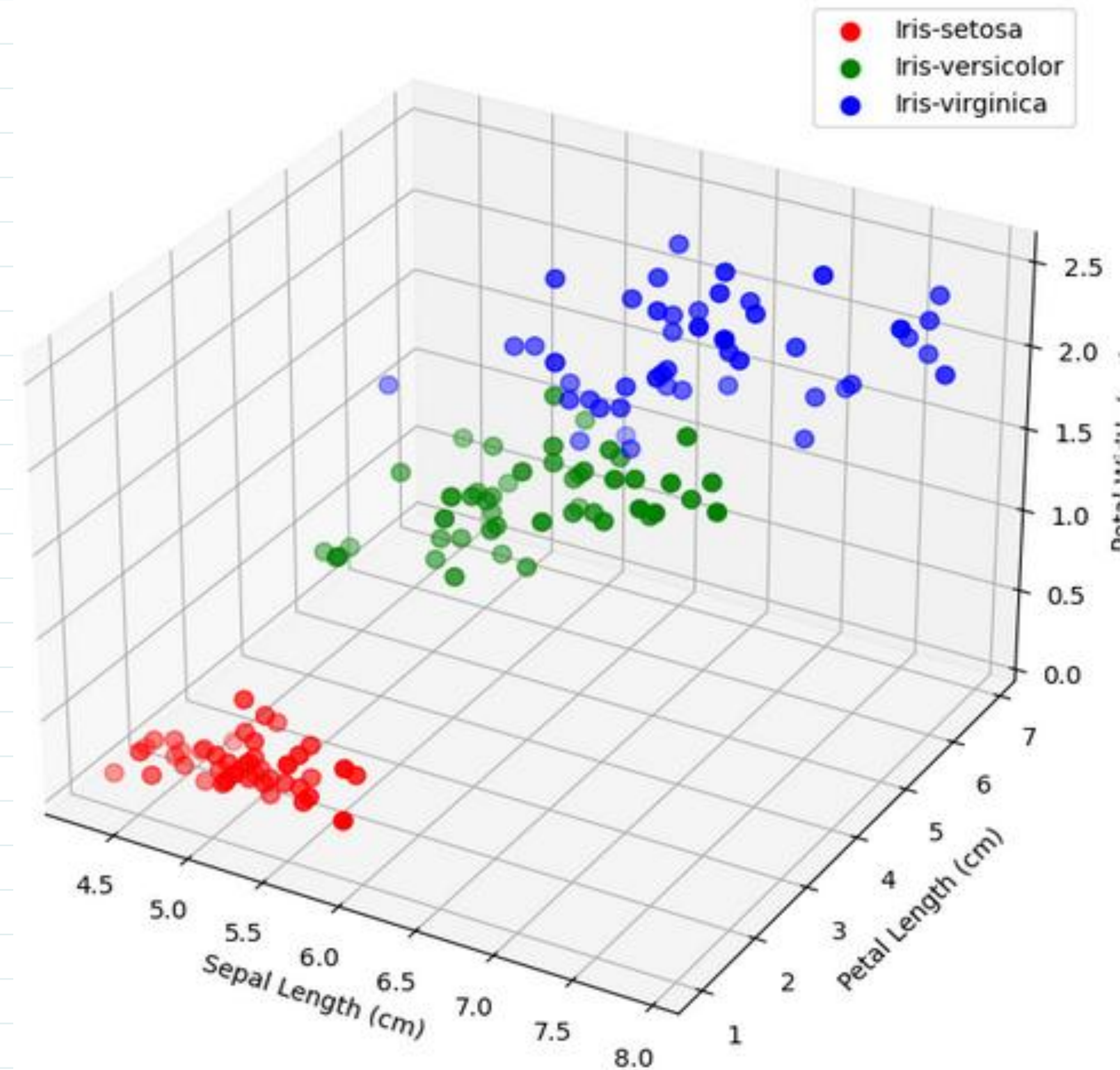
Bagaimana jika data tidak linear ?



Visualisasi Klasifikasi Iris dengan SVM



3D Visualisasi Klasifikasi Iris dengan SVM



Case Study : SVM

1. Explore dataset di platform Kaggle, temukan dataset untuk solusi klasifikasi menggunakan SVM!



Terima Kasih

<http://youtube.com/@rojulman>