LECTURER: TAI LE QUY

MACHINE LEARNING – SUPERVISED LEARNNG

UNIT 4

SUPPORT VECTOR MACHINES

STUDY GOALS

- understand the concept of SVMs.
- understand hyperplane and kernel trick in SVMs.
- apply SVMs classification model using Python.

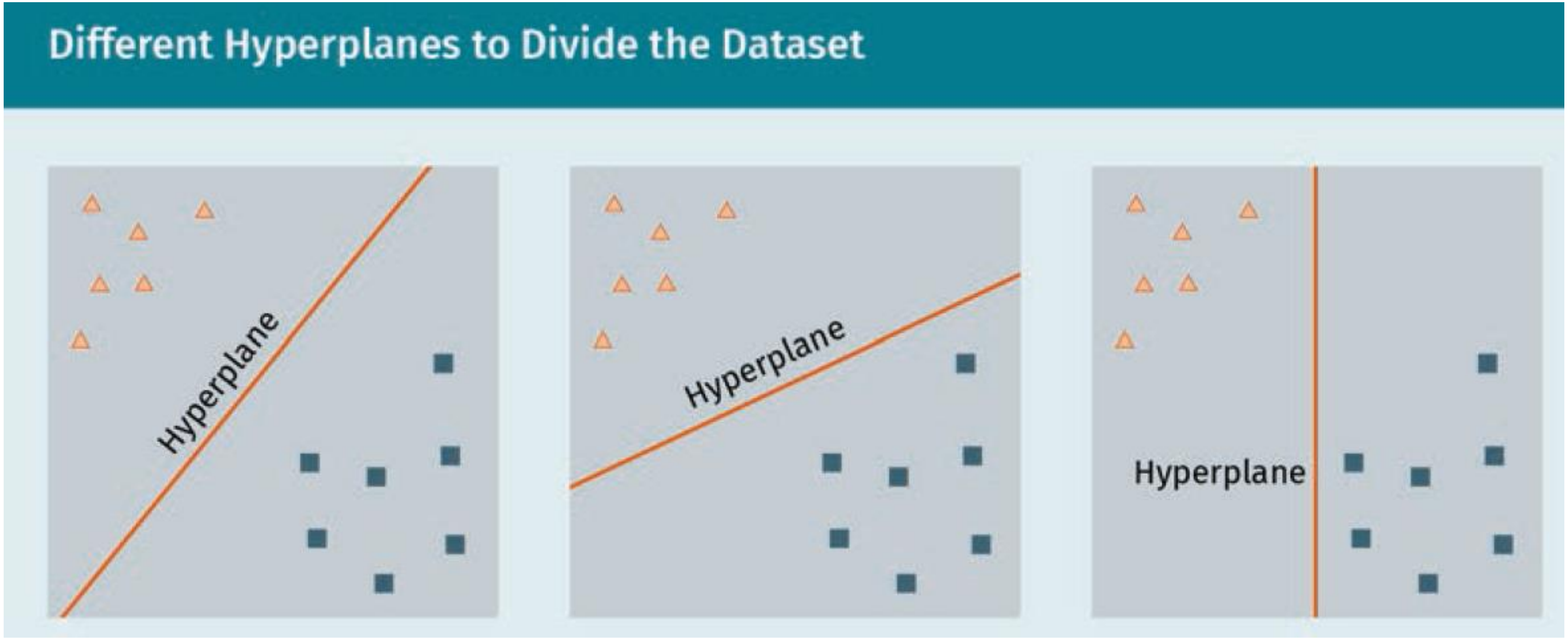
EXPLAIN SIMPLY

1. How can we choose the best hyperplane for classification?
2. How does the kernel trick, which allows SVMs to classify nonlinear data, work?
3. How can we utilize SVMs with the help of Python to solve real-world problems??

- The idea behind SVMs is to divide the data into two classes separated by a classification boundary in an n-dimensional space called a **hyperplane**
- A **hyperplane** is a subspace within a vector with one less dimension
- New observations are assigned to one or the other class depending on which side of the hyperplane they are located.

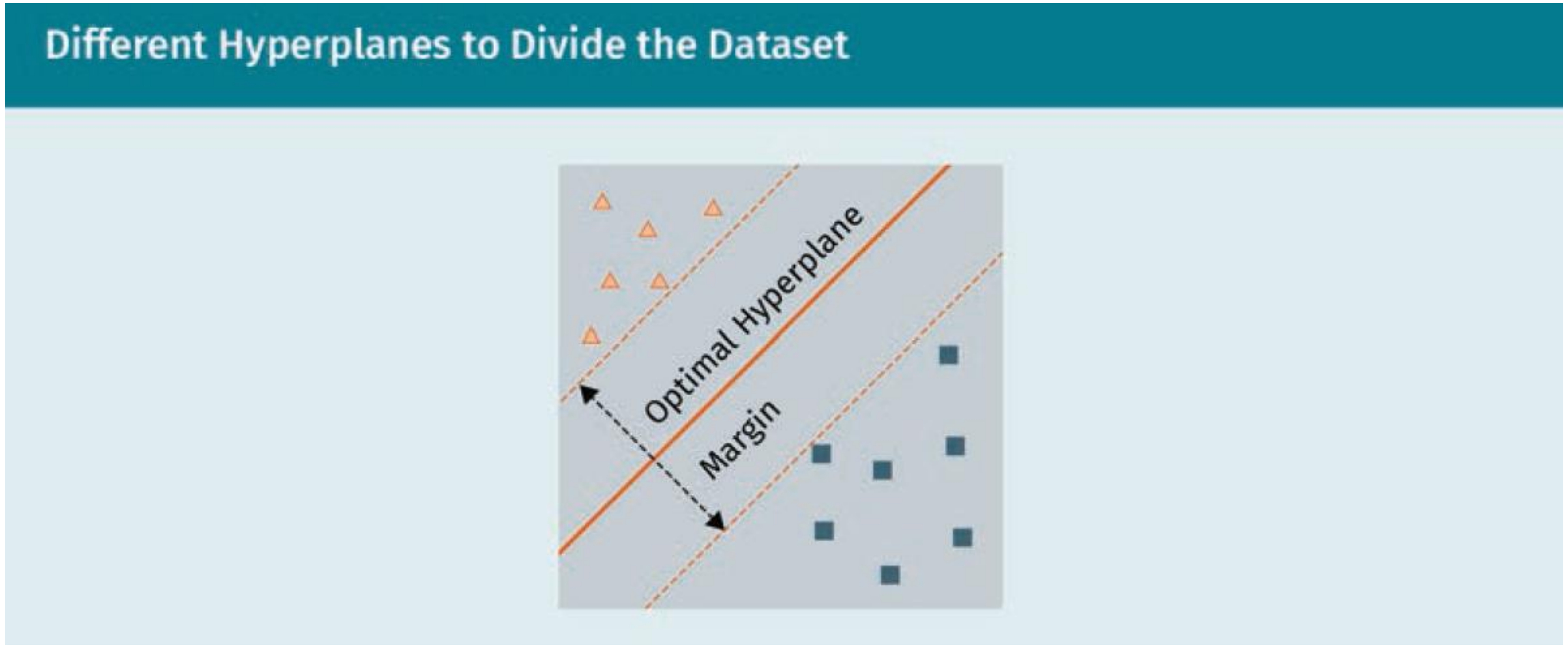
Source of image: Course book

SUPPORT VECTOR MACHINES (SVM)



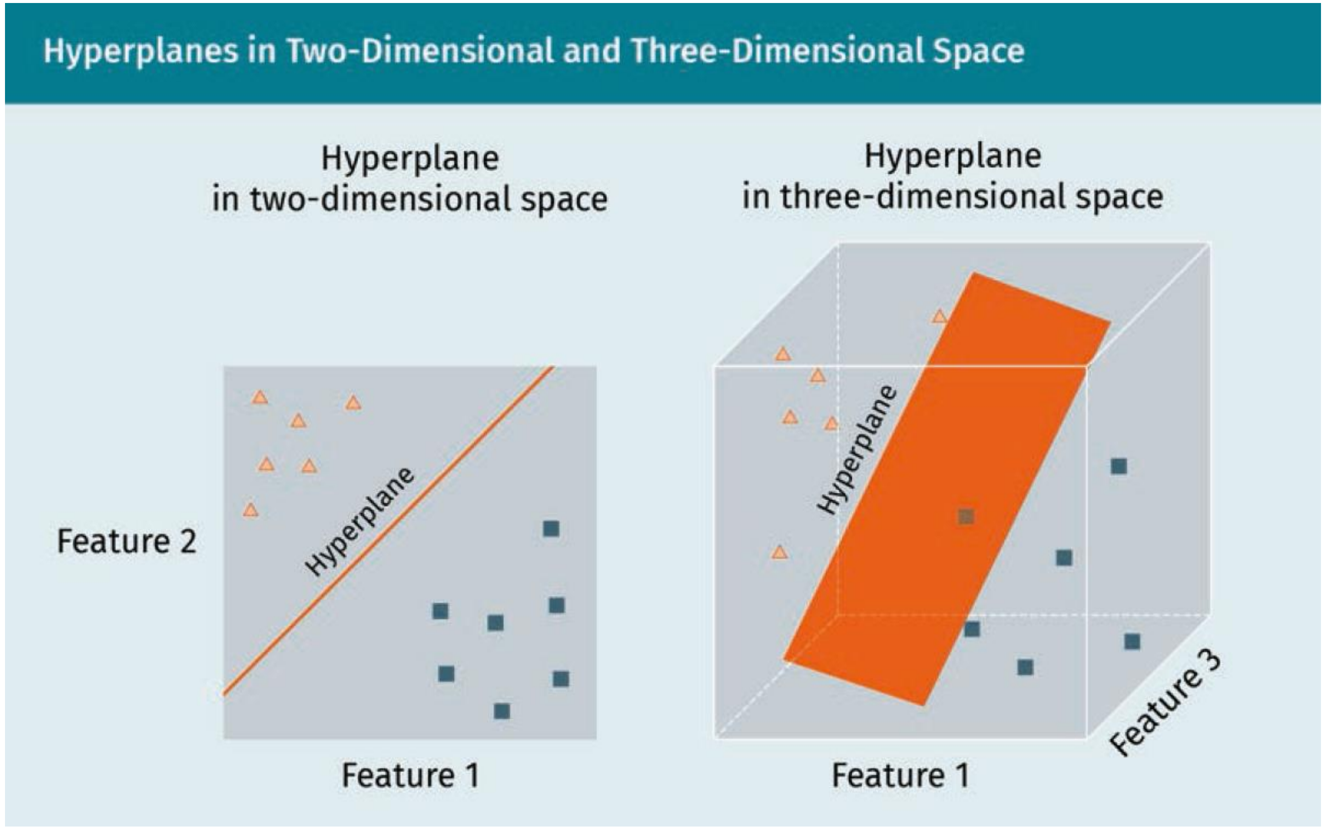
Source of image: Course book

- Choose hyperplane where the distance between the two classes is maximized
- SVMs: large margin classifiers



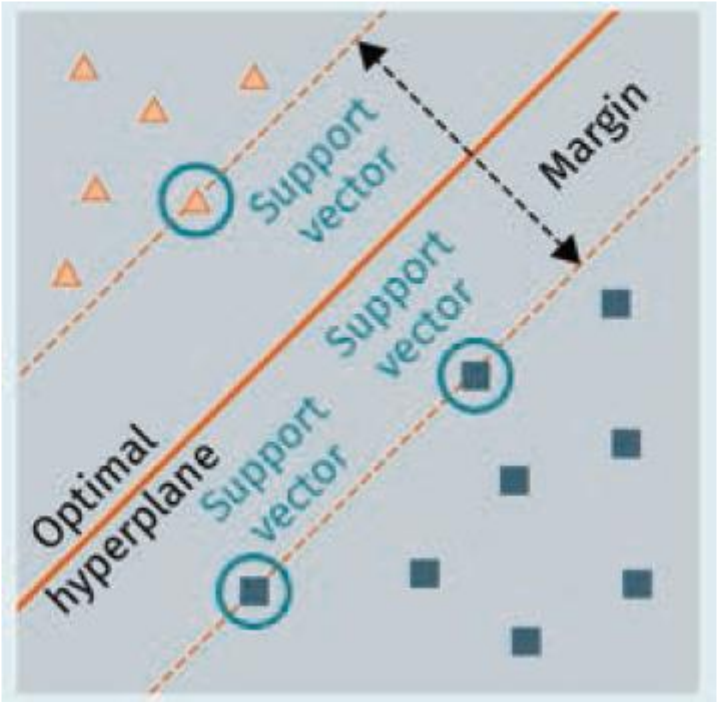
- SVMs work with both linear and nonlinear classification by using the **kernel** trick
- Data that cannot be easily separated in a lower dimensional space are mapped to higher dimensional space (where they can then be separated more easily)

– Given n-dimensional vector space, a hyperplane is a subspace of the same with n-1 dimensions



Source of image: Course book

- Support vectors: data points that lie on the margin of a class and determine the location of the hyperplane
- Two types of SVMs:
  - Linear SVMs are used for data that are linearly separable
  - Nonlinear SVMs are used for data that are not linearly separable
    - The data are mapped into a higher dimensional space, where they are more easily separated using kernel trick



Source of image: Course book

- A hyperplane of p-dimensional space:

$$\beta_0 + \beta_1 X_1 + \ldots + \beta_p X_p$$

- The prediction outcome y_i of an observation I now depends on which side of the hyperplane it is located.
- The goal is to find the hyperplane that maximizes the margin boundaries M

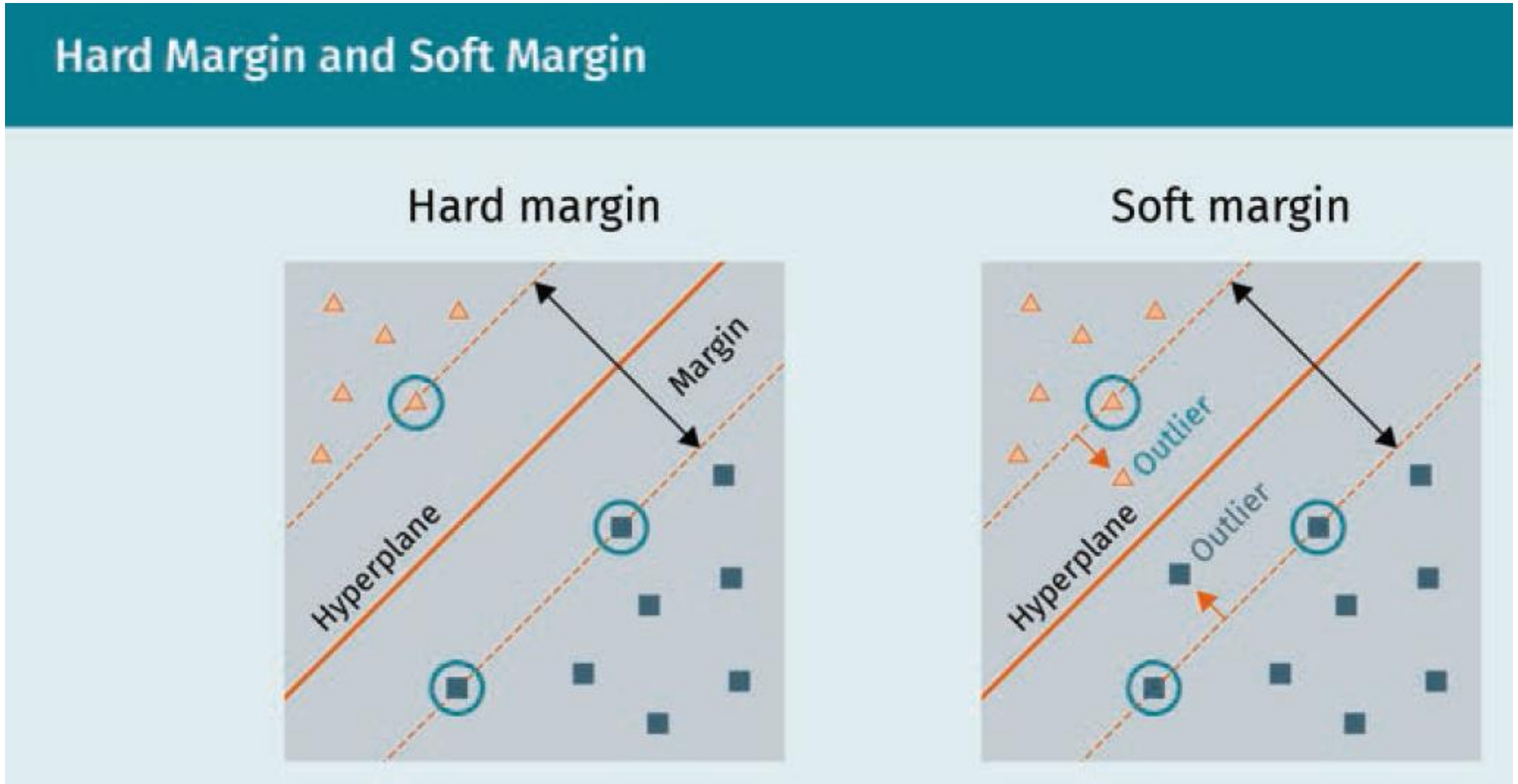
$$\begin{aligned} &\text{maximize M, given } \beta_0, \beta_1, \ldots, \beta_p \\ &\text{subject to } y_i\big(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}\big) \geq \text{M}, \qquad \text{i} = 1,2, \ldots, \text{n} \end{aligned}$$

distance between the i^{th} observation to the decision boundary

→ To make SVMs less sensitive to outliers we need to allow misclassifications (soft margin) → bias-variance-tradeoff

Source of image: Course book

– With a soft margin, the threshold allows misclassification and thus leads to a higher bias on the training data



Source of image: Course book

- Hyperparameter C that allows the algorithm to accept errors (chosen by trial and error or cross-validation)
  - If C is close to zero, the margin is large and many misclassifications are allowed (soft margin). If C takes a high value, the margin is narrow and misclassifications are not allowed (hard margin)
- Slack variable $\varepsilon_I \geq 0$ for each instance, indicating how many instances I can violate the margin

$$\text{maximize } M, \quad \text{given } \beta_0, \beta_1, \ldots, \beta_p$$

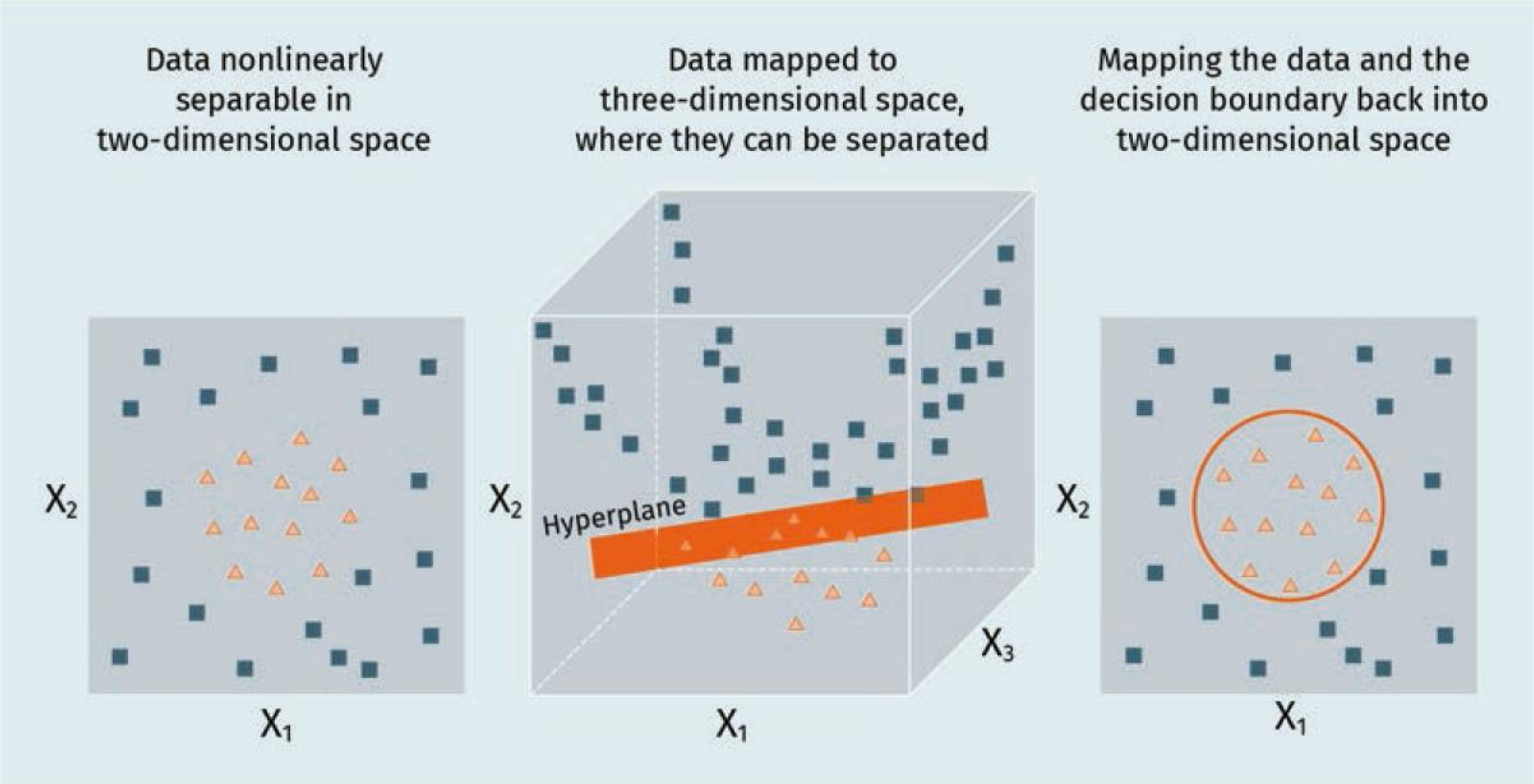
$$\text{subject to } y_i(\beta_0 + \beta_1 x_{i1} + \ldots + \beta_p x_{ip}) \geq M(1 - \varepsilon_i), \qquad i = 1, 2, \ldots, n$$

$$\varepsilon_i \geq 0,$$

$$\sum_{i=1}^{n} \varepsilon_i \leq C$$

Source of image: Course book

THE KERNEL TRICK



$$X_1 = x_1^2, \quad X_2 = x_2^2, \quad X_3 = \sqrt{2}x_1x_2$$

- A transformation of the data into a higher-dimensional space is a computationally intensive task
- Solution: the kernel trick does not perform an actual transformation
  - Merely calculates the relation of the data points to each other as though they were in a higher-dimensional space.
  - This mapping of the data is done with the help of a kernel function

- Directly compute the distance, i.e., the scalar products of the data points for the expanded feature representation, without ever actually computing the expansion.
- Common methods
  - Polynomial kernel: computes all possible polynomials up to a certain degree
  - Radial basis function (RBF) (Gaussian kernel): corresponds to an infinite-dimensional feature space

- Polynomial kernel
  - Computes the decision boundary K via the dot product of the input features X_1 and X_2 by raising the power of the kernel to the degree d.

$$K(X_1, X_2) = (1 + \langle X_1, X_2 \rangle)^d$$

$$\langle X_1, X_2 \rangle = \sum_{i=1}^{n} x_{1i} x_{2i}$$

- Radial basis function kernel
  - Calculates the decision boundary K for the inputs X_1 and X_2 by taking the Euclidean distance between X_1 and X_2 ($||X_1 - X_2||^2$)and scaling it with the help of the hyperparameter determined by cross validation

$$K(X_1, X_2) = exp\left\{-\gamma \ ||X_1 - X_2||^2\right\}$$

  - The radial basis kernel is extremely flexible, and, as a rule of thumb, we generally start with this kernel when fitting SVMs in practice”

- understand the concept of SVMs.
- understand hyperplane and kernel trick in SVMs.
- apply SVMs classification model using Python.

UNIT 4

TRANSFER TASK

Credit Score Classification: Case Study

- The **credit score** of a person determines the creditworthiness of the person. It helps financial companies determine if you can repay the loan or credit you are applying for.

- Create a rough project plan to achieve this goal. For each phase of this plan, explain which classification techniques might be applied.

- When a computer processes an image, it perceives it as a two-dimensional array of pixels. The size of the array corresponds to the resolution of the image, for example, if the image is 200 pixels wide and 200 pixels tall, the array will have the dimensions 200 x 200 x 3. The first two dimensions represent the width and height of the image, respectively, while the third dimension represents the RGB color channels. The values in the array can range from 0 to 255, which indicates the intensity of the pixel at each point.
- Describe the steps when using SVM for image classification task

Example: https://www.geeksforgeeks.org/image-classification-using-support-vector-machine-svm-in-python/

TRANSFER TASK
PRESENTATION OF THE RESULTS

Please present your results.
The results will be discussed in plenary.