Introduction to Machine Learning
PCIT-114

Department of Information Technology
GURU NANAK DEV ENGINEERING COLLEGE
LUDHIANA

Support Vector Regression (SVR)

- No ML method is superior to any other..
- Each method is different
  - Type of problem
  - Prior distribution
- SVR gives us the flexibility to define how much error is acceptable in our model and will find an appropriate line (or hyperplane in higher dimensions) to fit the data
- In contrast to OLS, the objective function of SVR is to minimize the coefficients — more specifically, the *L*2-norm of the coefficient vector — not the squared error.
- The error term is instead handled in the constraints, where we set the absolute error less than or equal to a specified margin, called the maximum error, ϵ (epsilon). We can tune epsilon to gain the desired accuracy of our model.

Support Vector Regression (SVR)

- In SVR, the goal is to find a hyperplane that best fits the data such that the maximum number of points lie within a margin around the hyperplane.
- SVR objective function:
- consider a dataset of N data points, where each data point is a tuple of (x, y) values. Our goal is to find a function f(x) that can predict the corresponding y values for any new input x.
- To derive the SVR objective function, we first define an error term for each data point i as:

$$e\_i = y\_i - f(x\_i)$$

where y_i is the actual output for data point i and f(x_i) is the predicted output by the function f for input x_i.

Support Vector Regression (SVR)

- The goal of SVR is to find a function *f* that minimizes the error on the training data, while also ensuring that the function is not overfitting the data. To achieve this, we introduce a *slack variable ξ_i for each data point i*, which allows some points to be inside the margin or even on the wrong side of the margin.
- We can now formulate the SVR objective function as follows:
  - minimize: $(1/2) * ||w||^2 + C * \Sigma_i (\xi_i + \xi_i^*)$
  - subject to:
  - $y_i - f(x_i) \le \varepsilon + \xi_i$
  - $f(x_i) - y_i \le \varepsilon + \xi_i^*$

Support Vector Regression (SVR)

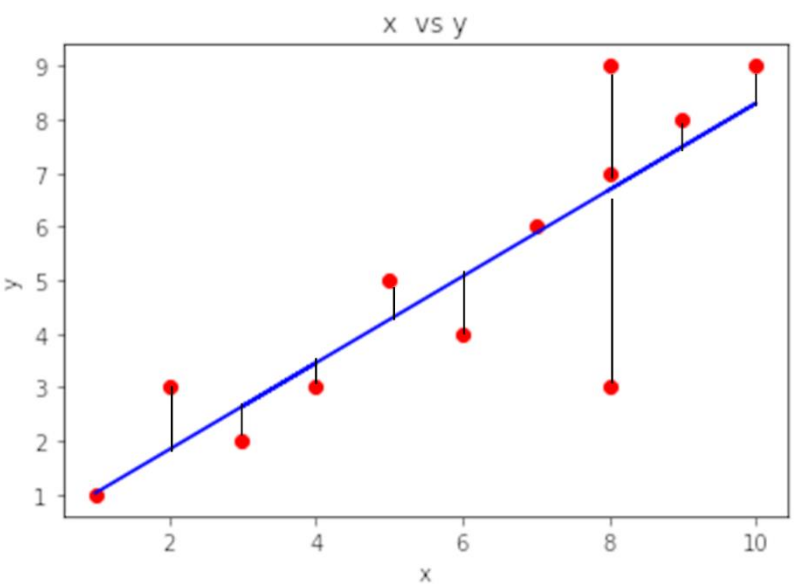
- The squared L2-norm of a vector w is defined as the sum of the squares of its components, i.e., $||w||^2 = w_1^2 + w_2^2 + ... + w_n^2$, where n is the dimensionality of the vector. In the case of SVR, the weight vector w is a vector of coefficients that defines the hyperplane that best fits the data. The L2-norm of w is used as a regularization term in the SVR objective function, to prevent overfitting and to ensure that the hyperplane is as simple as possible. By minimizing $||w||^2$, we are effectively minimizing the complexity of the hyperplane, which helps to reduce the risk of overfitting.

Support Vector Regression (SVR)

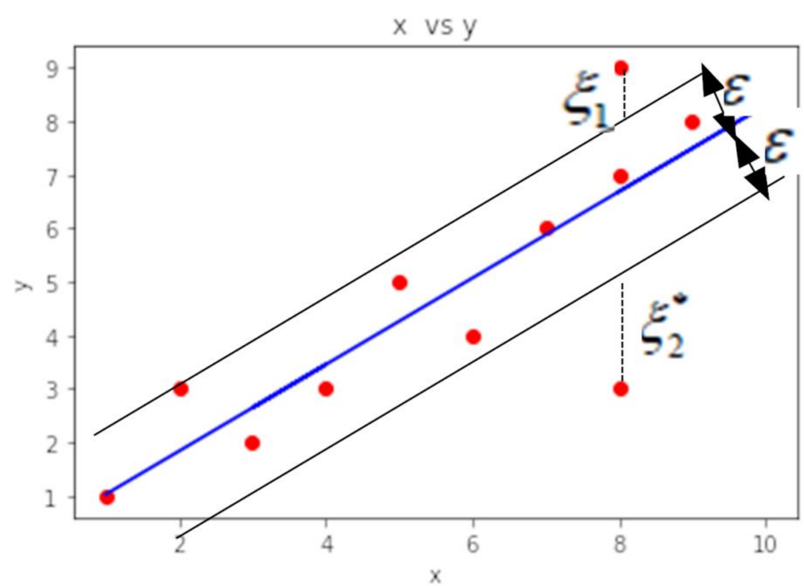
- $||w||^2$ is the squared L2-norm of the weight vector w
- C is a hyperparameter that determines the trade-off between minimizing the error and maximizing the margin
- ε is the margin width, which is a hyperparameter that determines the maximum allowable deviation from the true output or also known as width of the epsilon-insensitive tube
- If the error for a training example is within the tube, the loss is zero. If the error is outside the tube, the loss is proportional to the distance of the error from the tube.
- ξ_i and ξ_i^* are the slack variables for data point i, which are non-negative values that penalize errors and violations of the margin.
- Once we have the optimal values of w and b, we can use the function $f(x) = w^T * x + b$ to predict the output for any new input x.

Support Vector Regression (SVR)

Linear Regression



Support Vector Regression



Ordinary Least Squares

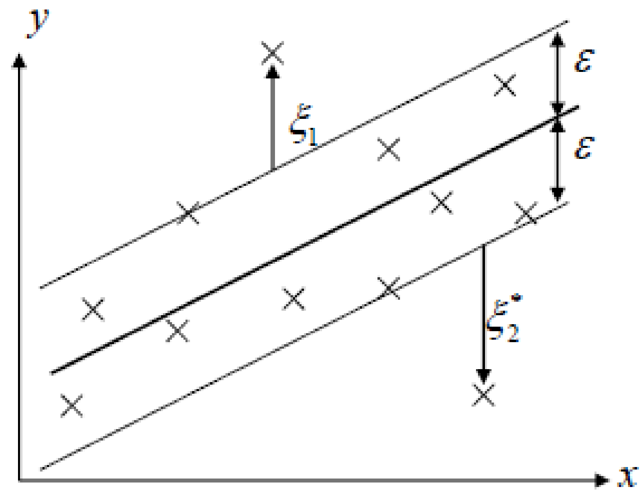
minmize<-sum(y-y')^2

$$\frac{1}{2}\|w\|^2 + C\sum_{i=1}^{m}(\xi_i + \xi_i^*) \rightarrow min$$

Support Vector Regression (SVR)

Given training data

$$(\mathbf{x}_i, y_i) \quad i = 1, \ldots, m$$



Minimize

$$\frac{1}{2} \| \mathbf{w} \|^2 + C \sum_{i=1}^{m} (\xi_i + \xi_i^*)$$

Under constraints

$$\begin{cases} y_i - (\mathbf{w} \cdot \mathbf{x}_i) - b \le \varepsilon + \xi_i \\ (\mathbf{w} \cdot \mathbf{x}_i) + b - y_i \le \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \ge 0, i = 1, \ldots, m \end{cases}$$

Support Vector Regression (SVR)

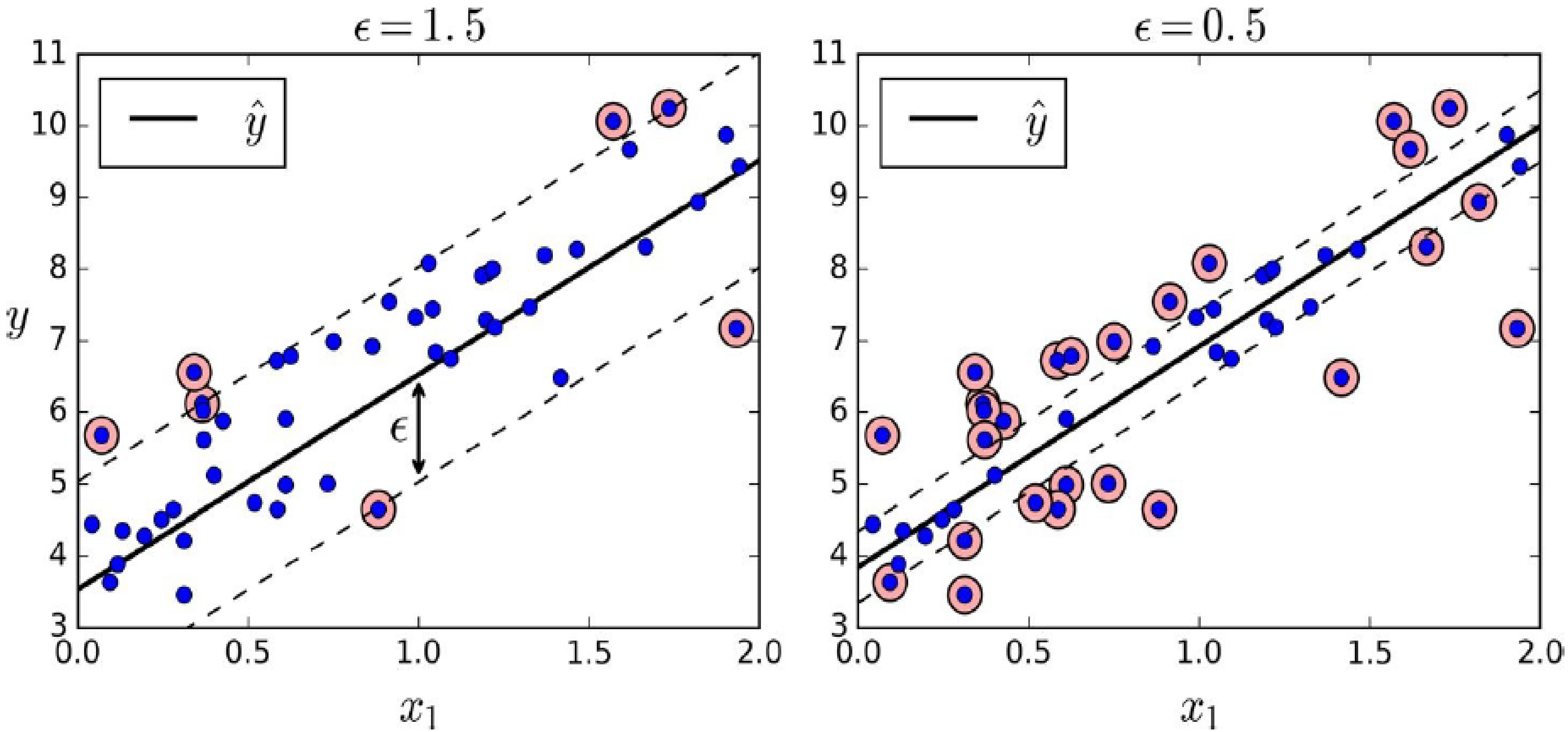


Image Source: Hands-On Machine Learning with Scikit-Learn and TensorFlow Concepts, Tools, and Techniques to Build Intelligent Systems by Aurélien Géron

Support Vector Regression (SVR)

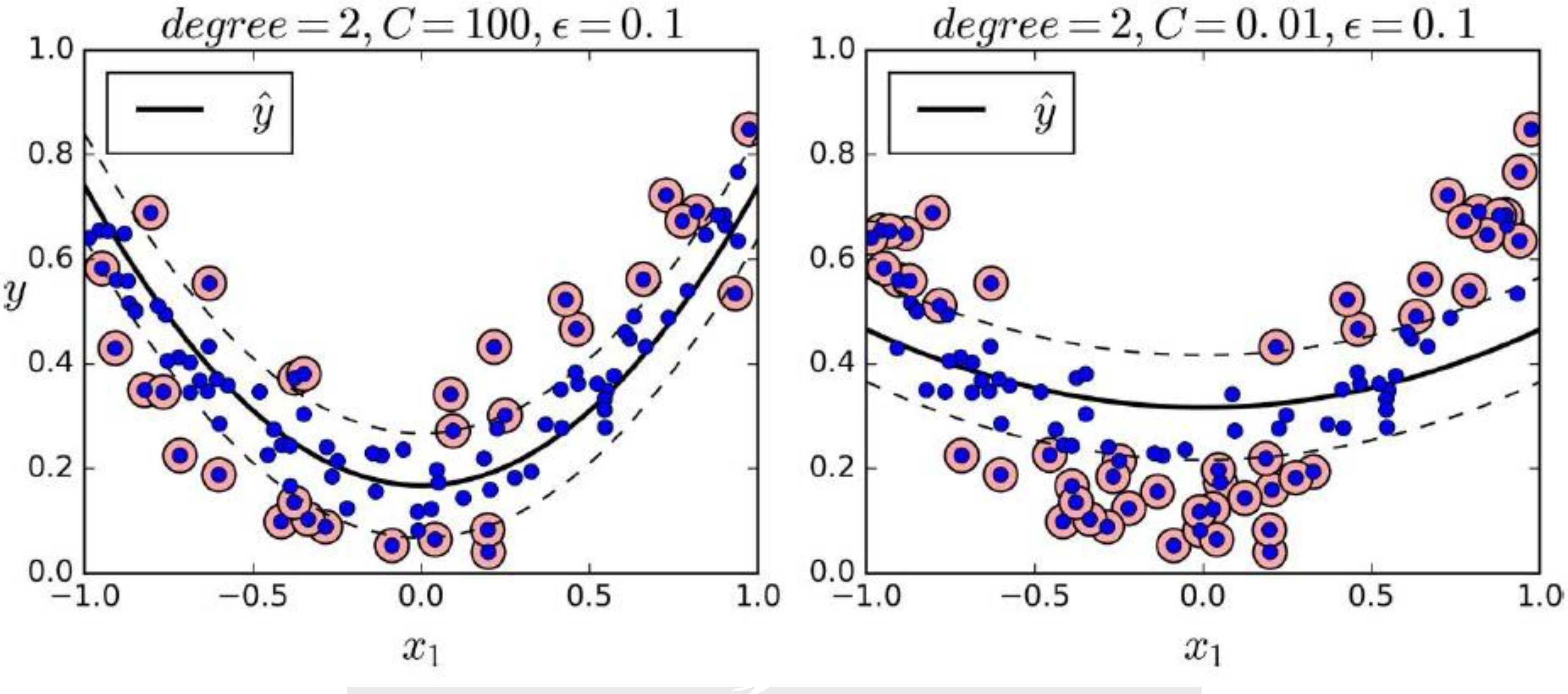


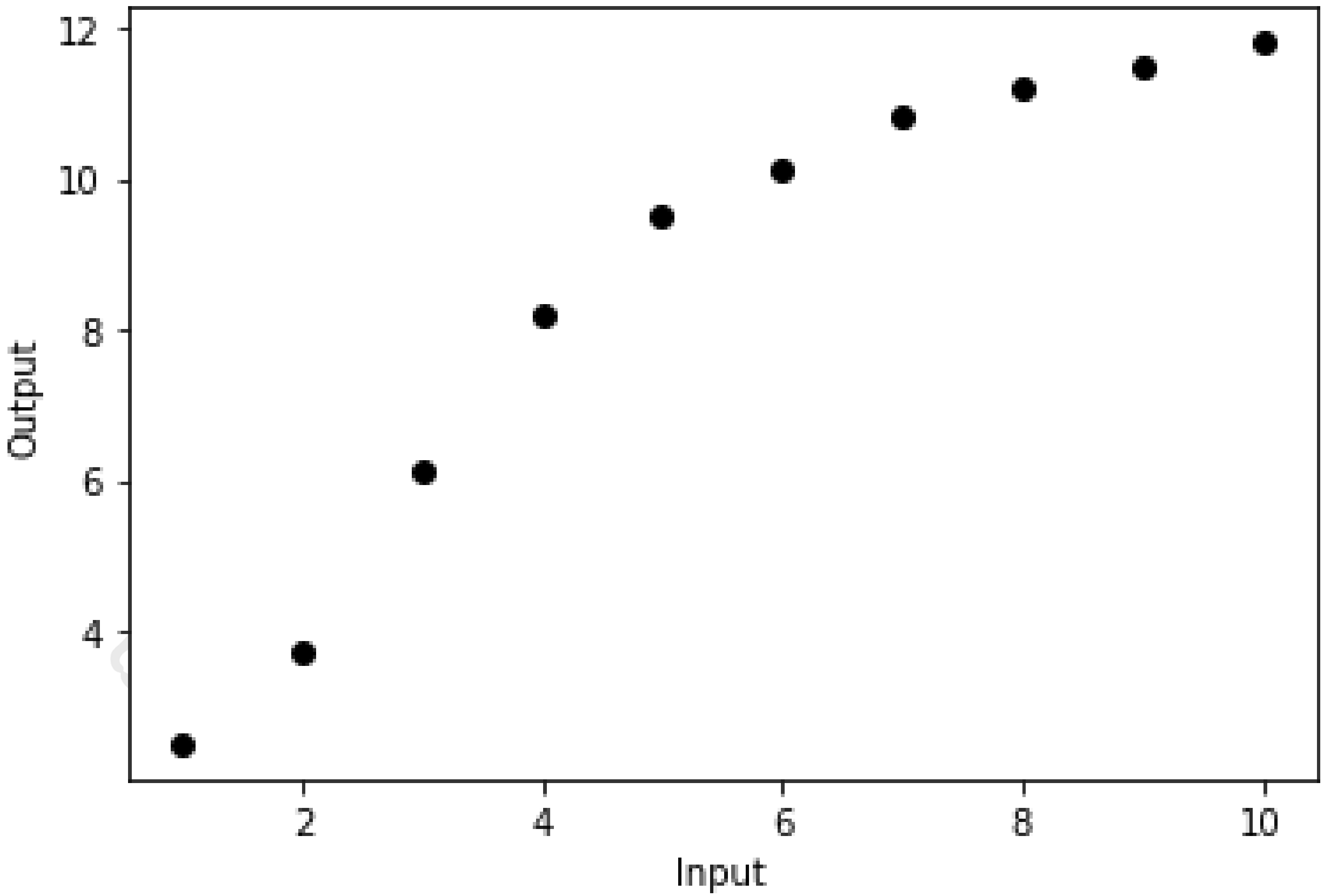
Image Source: Hands-On Machine Learning with Scikit-Learn and TensorFlow Concepts, Tools, and Techniques to Build Intelligent Systems by Aurélien Géron

Support Vector Regression (SVR)

- import numpy as np
- # define the training data
- X = np.array([[1], [2], [3], [4], [5], [6], [7], [8], [9], [10]])
- y = np.array([2.5, 3.7, 6.1, 8.2, 9.5, 10.1, 10.8, 11.2, 11.5, 11.8])

- import matplotlib.pyplot as plt
- # plot the training data
- plt.scatter(X, y, color='black')
- plt.xlabel('Input')
- plt.ylabel('Output')
- plt.show()

Support Vector Regression (SVR)



Support Vector Regression (SVR)

- from sklearn.svm import SVR
- # create an SVR object with linear kernel
- svr = SVR(kernel='linear', C=1.0, epsilon=0.1)
- # train the SVR model on the training data
- svr.fit(X, y)

Support Vector Regression (SVR)

- import numpy as np
- import matplotlib.pyplot as plt
- # define the input values for which we want to make predictions
- X_test = np.linspace(0, 11, 100).reshape(-1, 1)
- # make predictions for the input values using the trained SVR model
- y_pred = svr.predict(X_test)
- # plot the training data
- plt.scatter(X, y, color='black')
- # plot the predicted function
- plt.plot(X_test, y_pred, color='blue', linewidth=2)
- # set the plot limits and labels
- plt.xlim(0, 11)
- plt.ylim(0, 14)
- plt.xlabel('Input')
- plt.ylabel('Output')
- # display the plot
- plt.show()

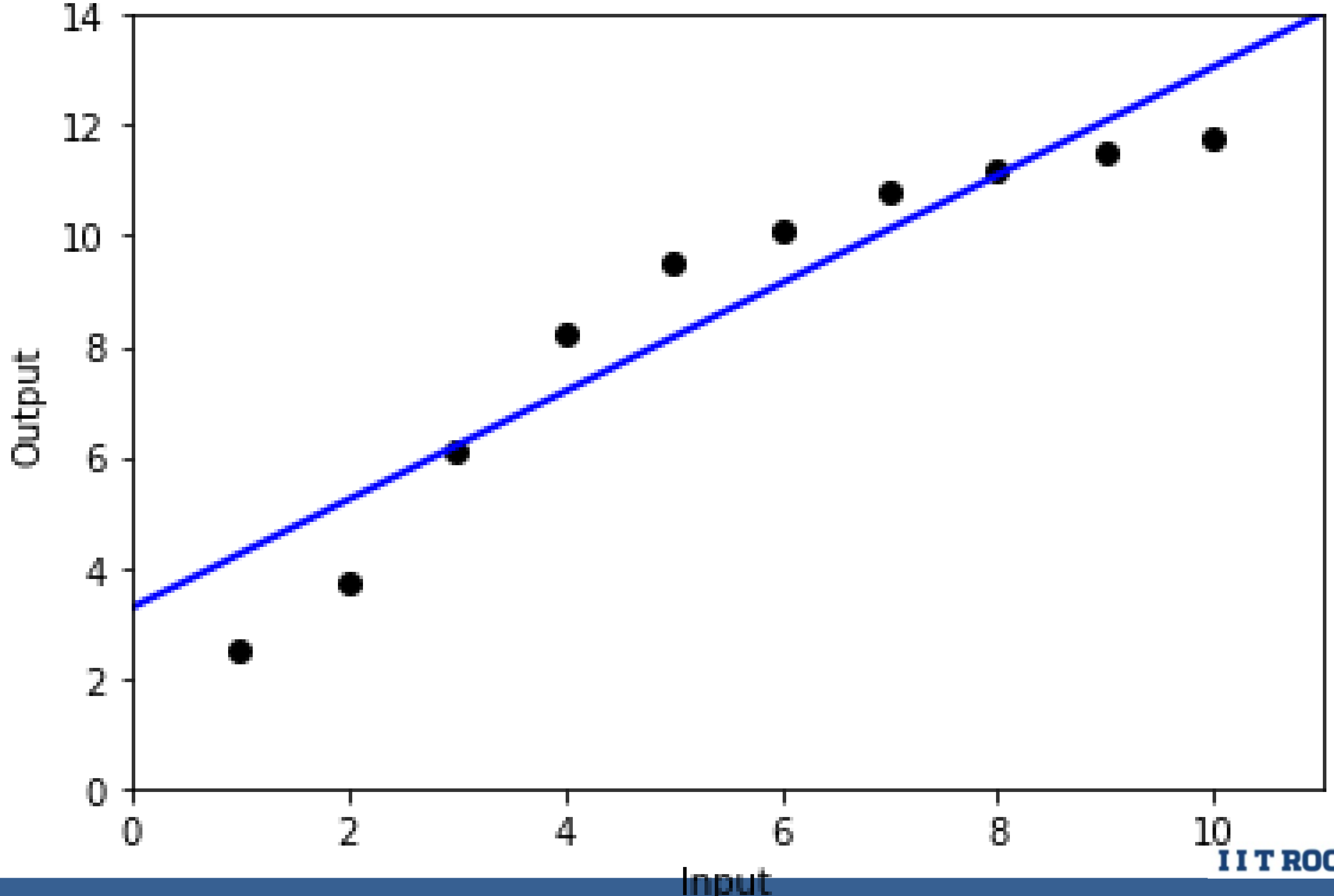
3 Types of Support Vector Regression (SVR)

1. Linear SVR: In this type of SVR, a linear kernel function is used. The linear kernel is the simplest and most computationally efficient kernel function. It can be used when there is a linear relationship between the input and output variables.
2. Polynomial SVR: In this type of SVR, a polynomial kernel function is used. The polynomial kernel function can capture more complex nonlinear relationships between the input and output variables. The degree of the polynomial can be adjusted to control the complexity of the model.
3. Radial Basis Function (RBF) SVR: In this type of SVR, an RBF kernel function is used. The RBF kernel function can capture highly nonlinear relationships between the input and output variables. The hyperparameters of the RBF kernel, such as the gamma parameter, can be adjusted to control the smoothness of the model.

- The choice of kernel function depends on the nature of the data and the complexity of the relationship between the input and output variables. In practice, it is often necessary to try different types of SVR and kernel functions and compare their performance on the test data to select the best model.

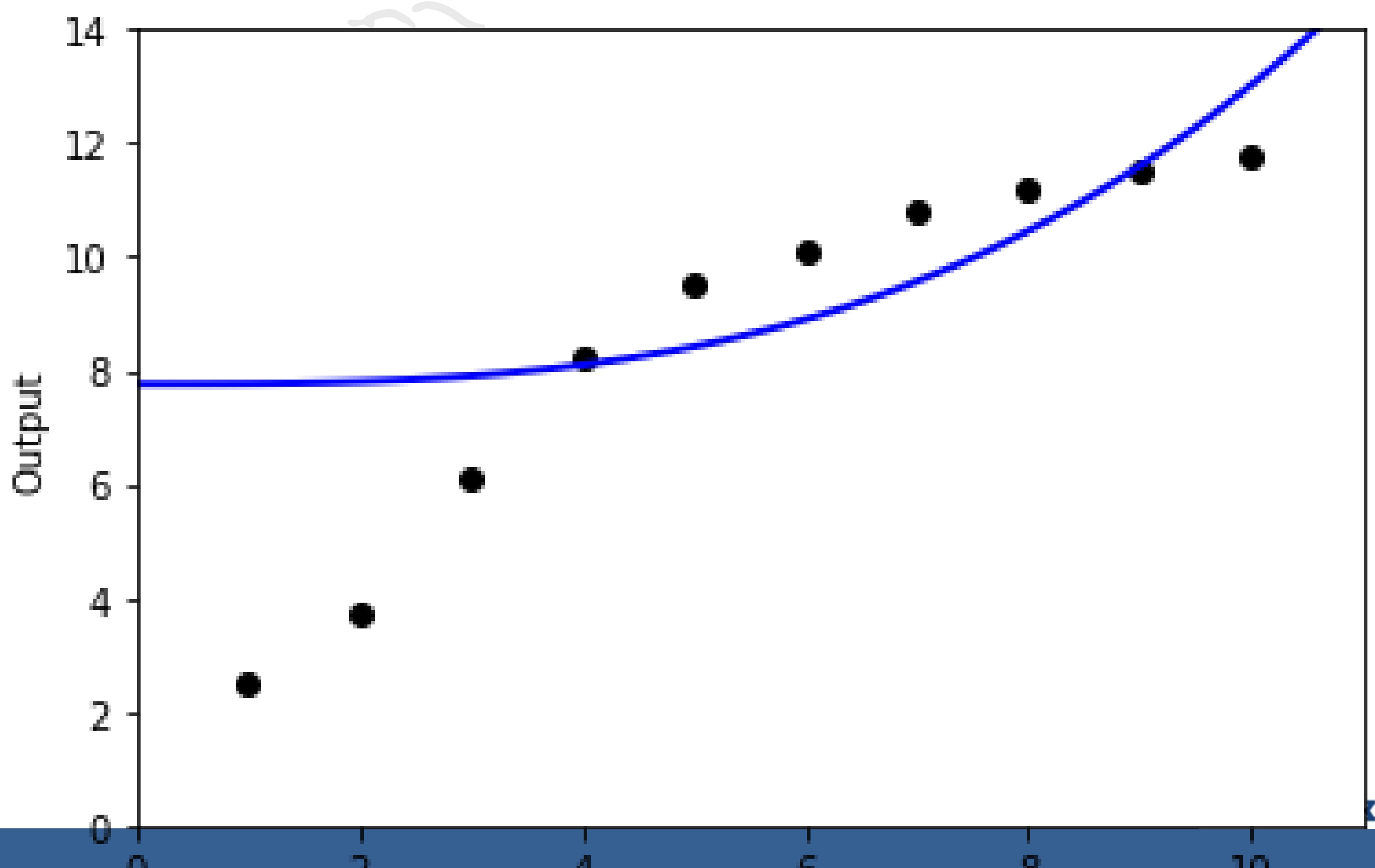
Support Vector Regression (SVR)

- SVR (LINEAR)



Support Vector Regression (SVR)

- SVR (Polynomial)



Support Vector Regression (SVR)

- SVR (RBF)