Supervised Learning Introduction

Course:
INFO-6145 Data Science and Machine Learning

Revised by:
Mohammad Noorchenarboo

September 26, 2024

Contents

Current Section

Supervised Learning

Supervised Learning involves training a model using input objects and their corresponding output values. The goal is to build a function f that can map new input data to predicted output values.

$$\hat{y} = f(x)$$

Where:

- $x^{(i)}$ are the input objects (feature vectors).
- $y^{(i)}$ are the output values (labels).
- $\hat{y}$ is the predicted output.

Example:

$$x = (x_1, x_2, \ldots, x_n)$$

Where $x_1, x_2, \ldots, x_n$ are the features, and the model predicts the output y based on these features.

Predictive Models: Regression

In predictive models, there are two main types:

Regression

The outcomes y are continuous values.

$$y \in \mathbb{R}$$

Example: Predicting house prices.

Classification

The outcomes y are discrete values.

$$y \in \{1, 2, \ldots, k\}$$

Example: Classifying emails as spam or not spam.

Predictive Models: Classification

Classification models deal with discrete outcomes. The outputs can belong to different categories or classes.

$$y \in \{1, 2, \ldots, k\}$$

Examples of classification:

- Binary classification: $y \in \{0, 1\}$ (e.g., true/false or spam/not spam).
- Multiclass classification: $y \in \{\text{green}, \text{red}, \text{black}\}$ (e.g., classifying different types of flowers).

Current Section

Regression Overview

Regression models are used for predicting continuous outcomes. The goal is to find a relationship between input features x and a continuous target variable y.

Some common types of regression models include:

- Linear regression
- Polynomial regression
- Support vector regression (SVR)
- Regression trees

Regression

Regression tasks involve predicting continuous output. Examples of regression tasks include:

- Predicting house prices
- Estimating stock prices

Some common types of regression models:

Linear Regression

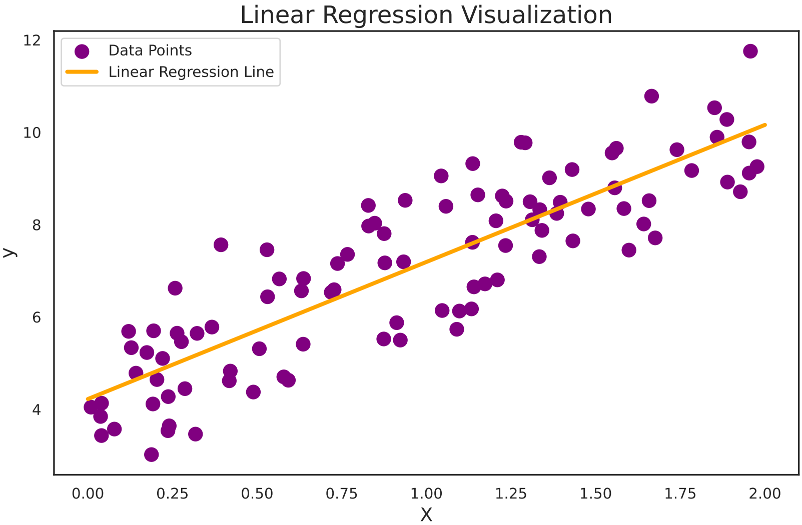
Simple yet highly effective for linear relationships.

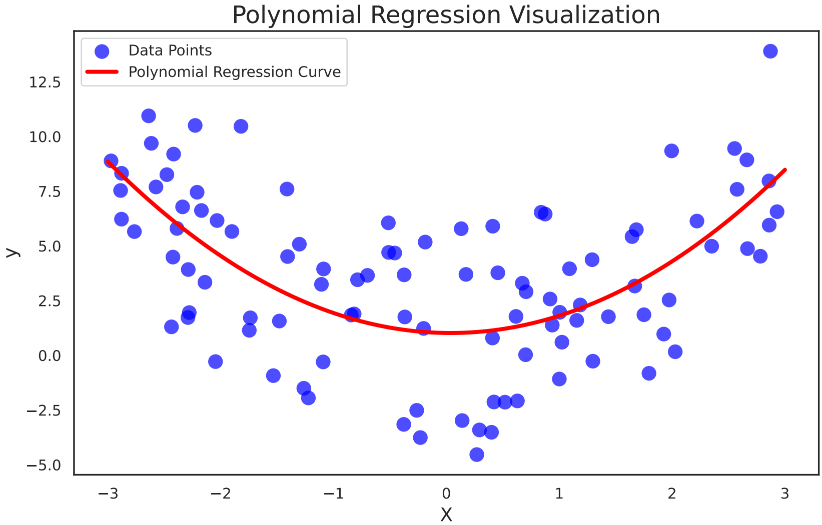
Regression Tree

Uses tree-based structures to partition the data into subgroups.

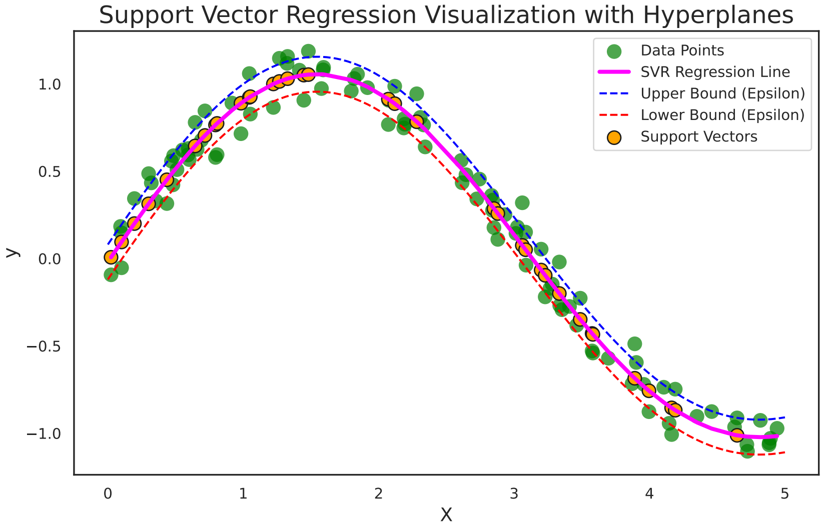
Support Vector Regression (SVR)

Attempts to fit the best hyperplane that minimizes error.

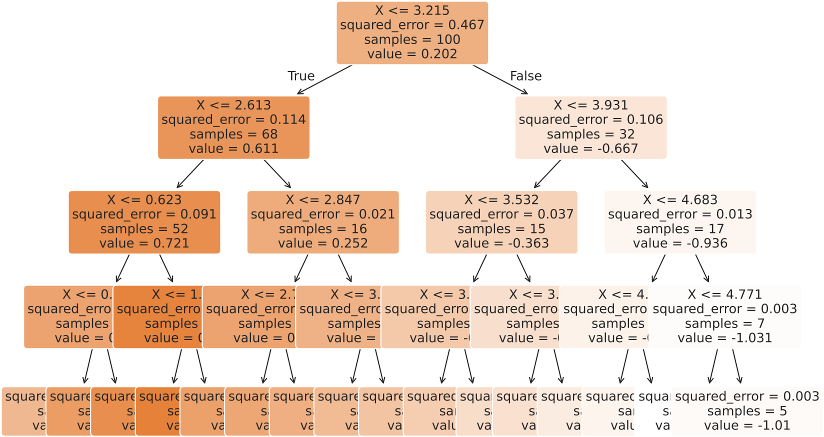

Linear Regression Visualization
Data Points
Linear Regression Line
12
10
8
6
4
y
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
X

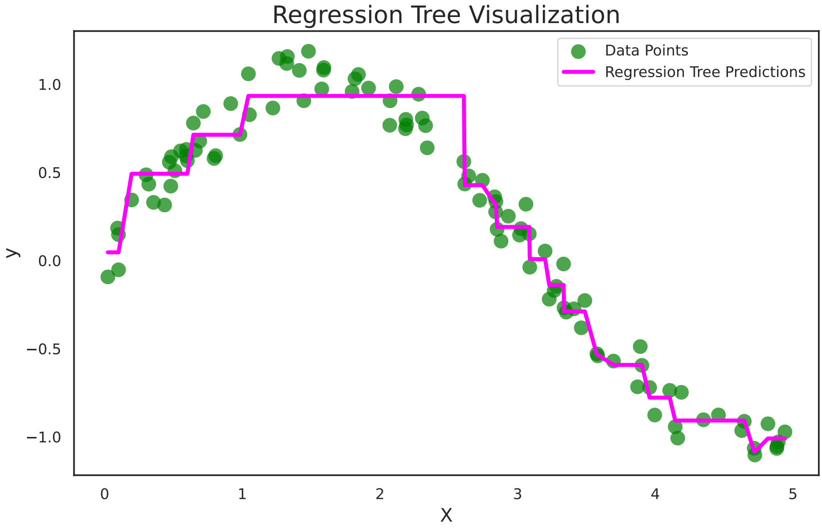


Polynomial Regression Visualization
Data Points
Polynomial Regression Curve
12.5
10.0
7.5
5.0
2.5
0.0
−2.5
−5.0
y
−3
−2
−1
0
1
2
3
X


Support Vector Regression Visualization with Hyperplanes
Data Points
SVR Regression Line
Upper Bound (Epsilon)
Lower Bound (Epsilon)
Support Vectors
1.0
0.5
0.0
−0.5
−1.0
y
0
1
2
3
4
5
X

Decision Tree Structure




Regression Tree Visualization
Data Points
Regression Tree Predictions
1.0
0.5
0.0
−0.5
−1.0
y
0
1
2
3
4
5
X

Linear Regression

Linear regression is the simplest form of regression, where the relationship between the input and output is linear:

$$y = w_0 + w_1 x_1$$

Where:

- y is the predicted output.
- w_0 is the bias term (intercept).
- w_1 is the weight (slope) for the input feature x_1.

Example: If $w_0 = 10$ and $w_1 = 2$, for $x_1 = 5$:

$$y = 10 + 2 \times 5 = 20$$

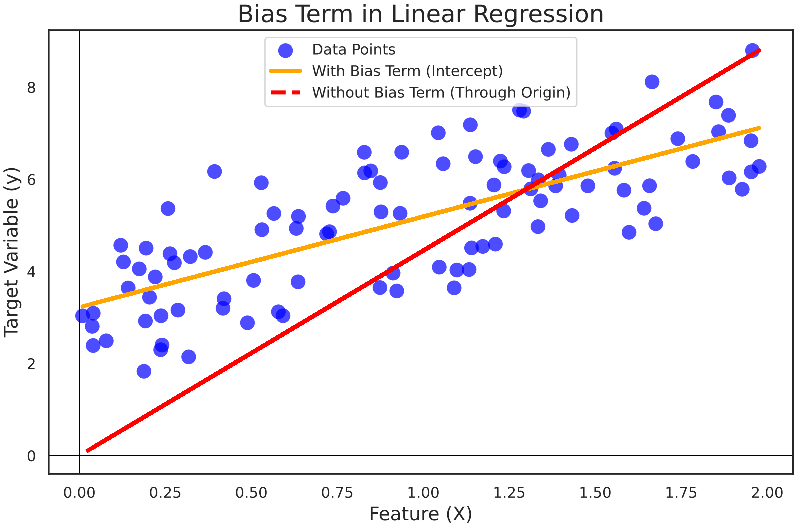
Bias Term in Linear Regression

The bias term w_0 in linear regression is an important constant that shifts the line up or down. It allows the model to fit the data better, even when the target variable y does not intersect the origin.

$$y = w_0 + w_1 x_1$$

If $w_0 = 0$, the regression line will always pass through the origin.

Bias Term in Linear Regression



Question: What is the effect of weight 0?

Q: What happens when an inputâs weight w_1 is 0?
If $w_1 = 0$, the model becomes:

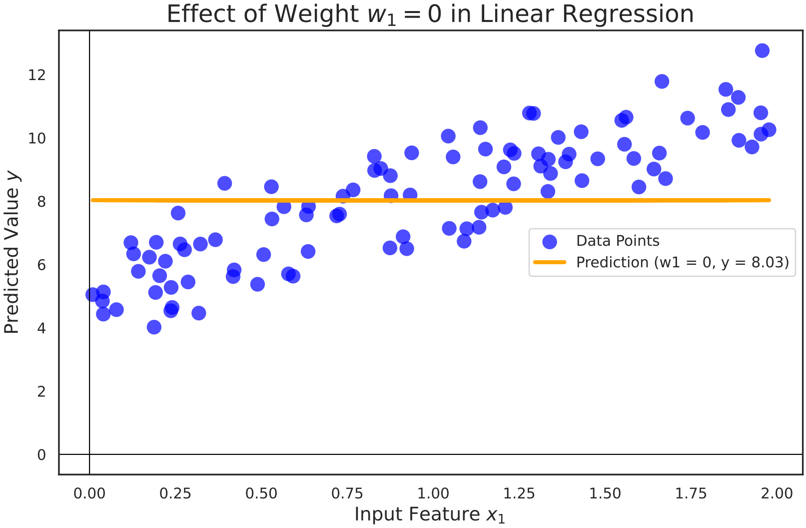
$$y = w_0$$

This means that the modelâs prediction is simply the bias term, and the input feature x_1 has no effect on the outcome.
Example:

$$y = 10 + 0 \times x_1 = 10$$

No matter the value of x_1, y will always be 10.

Question: What is the effect of weight 0?



Current Section

How Good is Our Model?

To evaluate how well a model fits the data, we use a **loss function.** The loss function measures the error between the actual value y and the predicted value $\hat{y}$.

$$\text{Loss} = L(y, \hat{y})$$

Common loss functions include:

- Mean Squared Error (MSE)
- Mean Absolute Error (MAE)
- Huber Loss

Mean Squared Error (MSE)

Mean Squared Error (MSE) is a commonly used loss function that measures the average squared difference between the actual values Y and the predicted values $\hat{Y}$.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2$$

Where:

- n is the number of observations.
- y_i is the actual value.
- $\hat{y}_i$ is the predicted value.

Example: If $y = [3, 5, 7]$ and $\hat{y} = [2.5, 5, 7.5]$:

$$MSE = \frac{1}{3}((3 - 2.5)^2 + (5 - 5)^2 + (7 - 7.5)^2) = 0.083$$

Mean Absolute Error (MAE)

Mean Absolute Error (MAE) is another common loss function that measures the average of the absolute differences between actual values Y and predicted values $\hat{Y}$.

$$MAE = \frac{1}{n}\sum_{i=1}^{n} |y_i - \hat{y}_i|$$

Where:

- n is the number of observations.
- y_i is the actual value.
- $\hat{y}_i$ is the predicted value.

Example: If $y = [3, 5, 7]$ and $\hat{y} = [2.5, 5, 7.5]$:

$$MAE = \frac{1}{3}(|3 - 2.5| + |5 - 5| + |7 - 7.5|) = 0.333$$

Huber Loss

Huber Loss combines elements of both MSE and MAE, offering robustness to outliers. It behaves as MAE when the error is large, and as MSE when the error is small. The Huber loss is defined as:

$$L_\delta(y, \hat{y}) = \begin{cases} \frac{1}{2}(y - \hat{y})^2 & \text{for } |y - \hat{y}| \leq \delta \\ \delta \cdot (|y - \hat{y}| - \frac{1}{2}\delta) & \text{otherwise} \end{cases}$$

Where:

- δ is a threshold that determines when to switch between MSE and MAE.
- y is the actual value.
- $\hat{y}$ is the predicted value.

Example: Let $y = 7$, $\hat{y} = 7.5$, and $\delta = 1$:

$$L_\delta(y, \hat{y}) = \frac{1}{2} \times (7 - 7.5)^2 = 0.125 \quad (\text{as } |7 - 7.5| \leq \delta)$$

Current Section

Visually Evaluating the Model

A visual way to evaluate the model's performance is by plotting the actual values y against the predicted values $\hat{y}$. The closer the points lie to the line $y = \hat{y}$, the better the model fits the data.
If the points scatter far from this line, it indicates poor predictions.