# Machine Learning-Session 2

## AI Labs

Gourav Bansal

# Machine Learning Session Two

## Agenda

What is Regression ?

Linear Regression

Multiple Linear Regression

Polynomial Regression

Support Vector Regression (SVR)

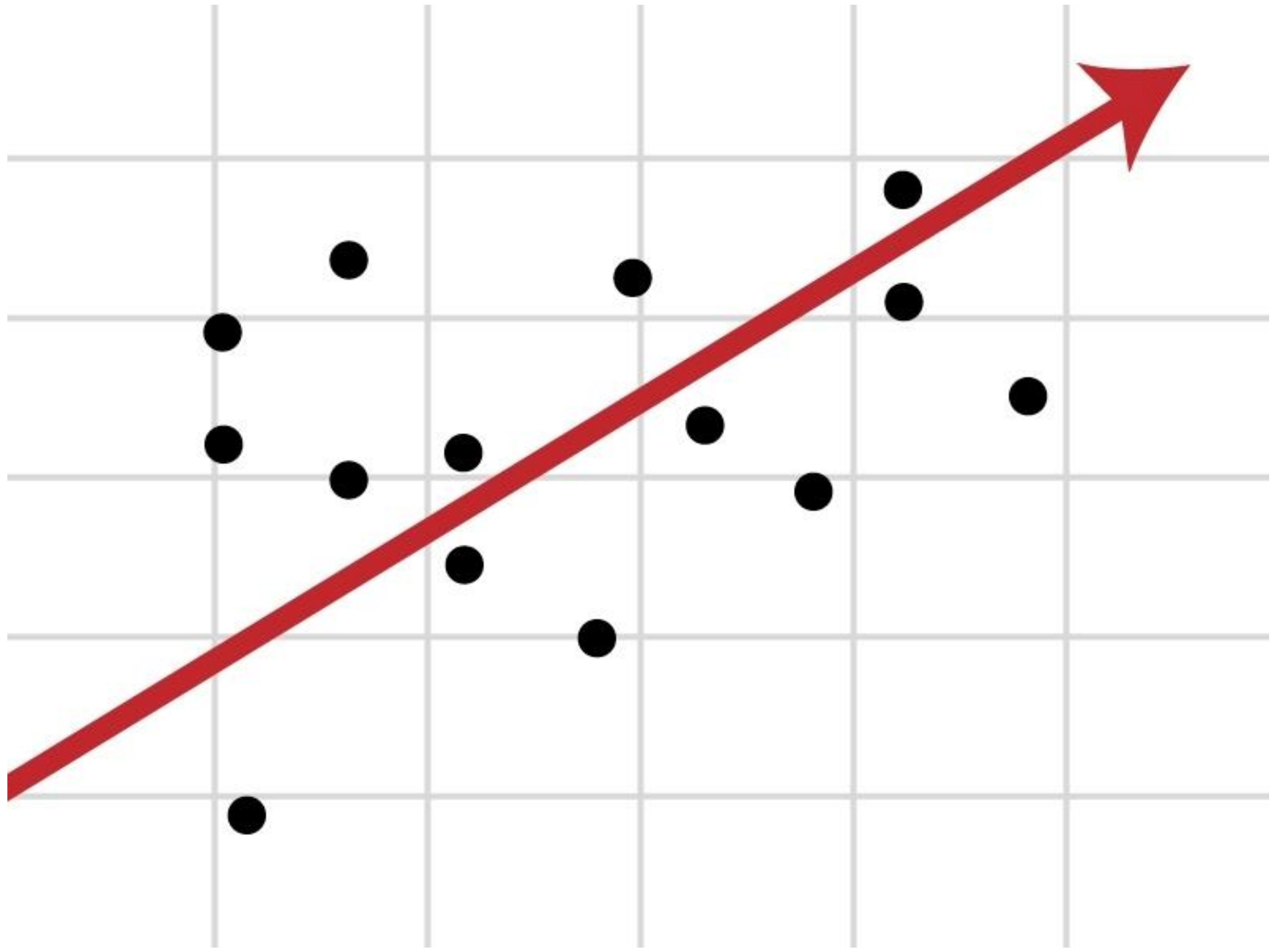
Decision Tree

Random Forest Regression

# What is Regression ?

**Regression** is a way for a computer to **learn how to predict numbers.**

**Regression** is a type of machine learning technique used to **predict a continuous numerical value** based on one or more input variables (features)..

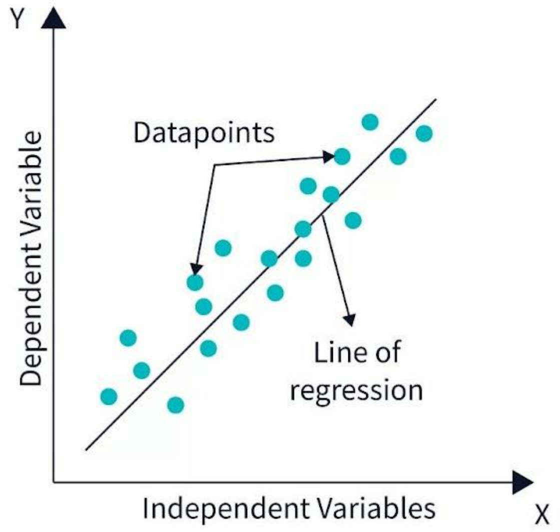


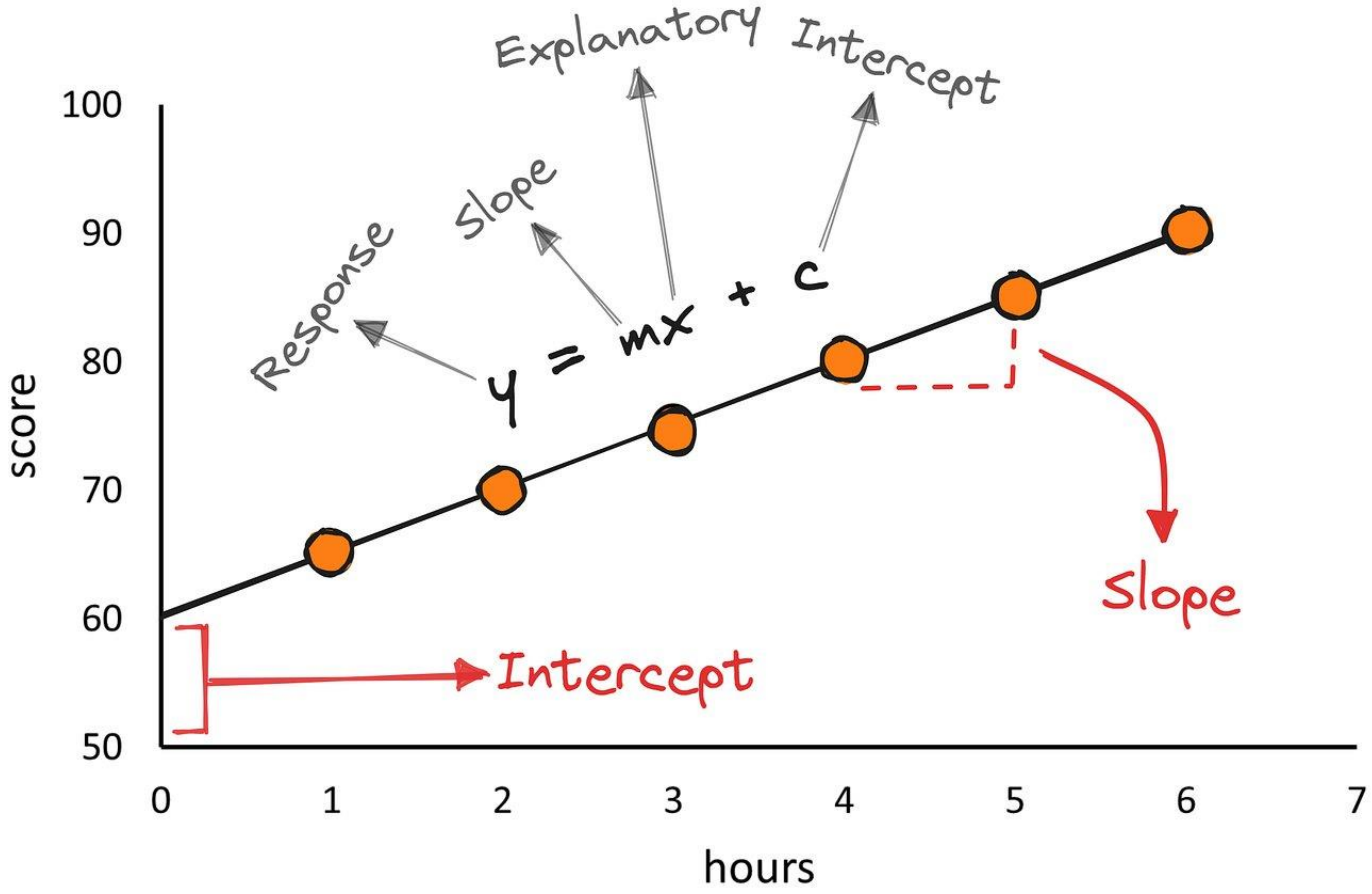
## Why is Regression Used?

- What will the **price of a house** be based on its size?
- How much **rain** will fall tomorrow?
- What is the **salary** based on experience?

# Linear Regression

$$\underbrace{\hat{y}}_{\text{Dependent variable}} = \underbrace{b_0}_{\text{y-intercept (constant)}} + \underbrace{b_1}_{\text{Slope coefficient}} \underbrace{X_1}_{\text{Independent variable}}$$



# Linear Regression Example

Imagine you're an HR manager trying to **predict someone's salary based on their years of experience**.

👇 **Data:**

| Years of Experience (X) | Salary (Y in ₹) |
|---|---|
| 1 | 3,00,000 |
| 2 | 4,00,000 |
| 3 | 5,00,000 |
| 4 | 6,00,000 |
| 5 | 7,00,000 |

Salary = 1,00,000 × Years of Experience + 2,00,000

# Multiple Linear Regression

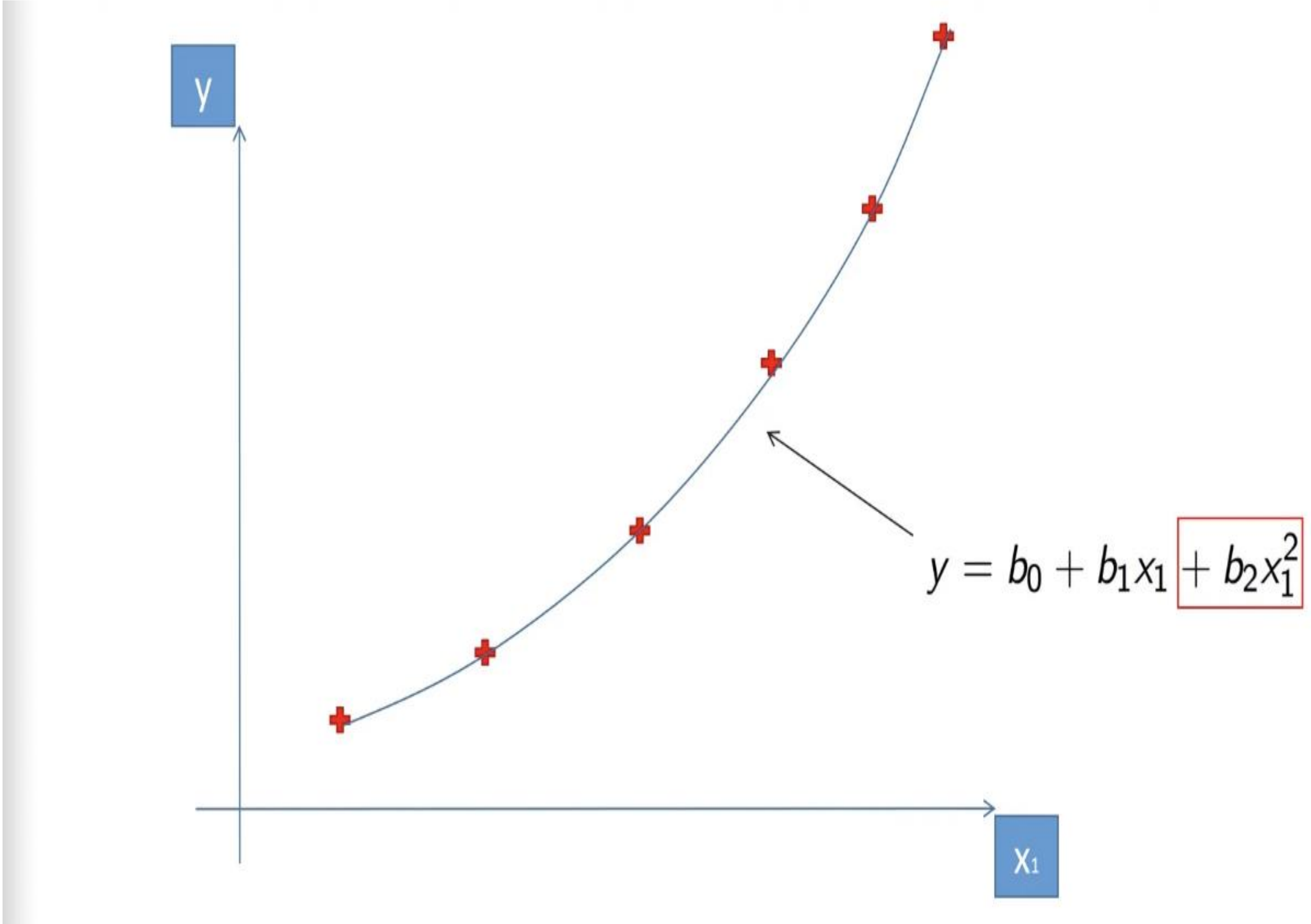
| Years of Experience | Education Level (1-5) | Salary (₹ in Lakhs) |
|---|---|---|
| 1 | 2 | 3.0 |
| 2 | 3 | 4.0 |
| 3 | 2 | 4.5 |
| 4 | 4 | 6.0 |
| 5 | 5 | 7.5 |
| 6 | 3 | 6.8 |

$$\underbrace{\hat{y}}_{\text{Dependent variable}} = \underbrace{b_0}_{\substack{\text{y-intercept} \\ \text{(constant)}}} + \underbrace{b_1}_{\text{Slope coefficient 1}}\underbrace{X_1}_{\text{Independent variable 1}} + \underbrace{b_2}_{\text{Slope coefficient 2}}\underbrace{X_2}_{\text{Independent variable 2}} + \cdots + \underbrace{b_n}_{\text{Slope coefficient n}}\underbrace{X_n}_{\text{Independent variable}}$$

Salary = $b_0$ + b1 × Experience + b2 × Education

# Polynomial Regression

| Years of Experience | Salary (₹ in Lakhs) |
|---|---|
| 1 | 2.0 |
| 2 | 2.5 |
| 3 | 3.2 |
| 4 | 4.5 |
| 5 | 6.5 |
| 6 | 9.0 |
| 7 | 12.5 |
| 8 | 16.8 |



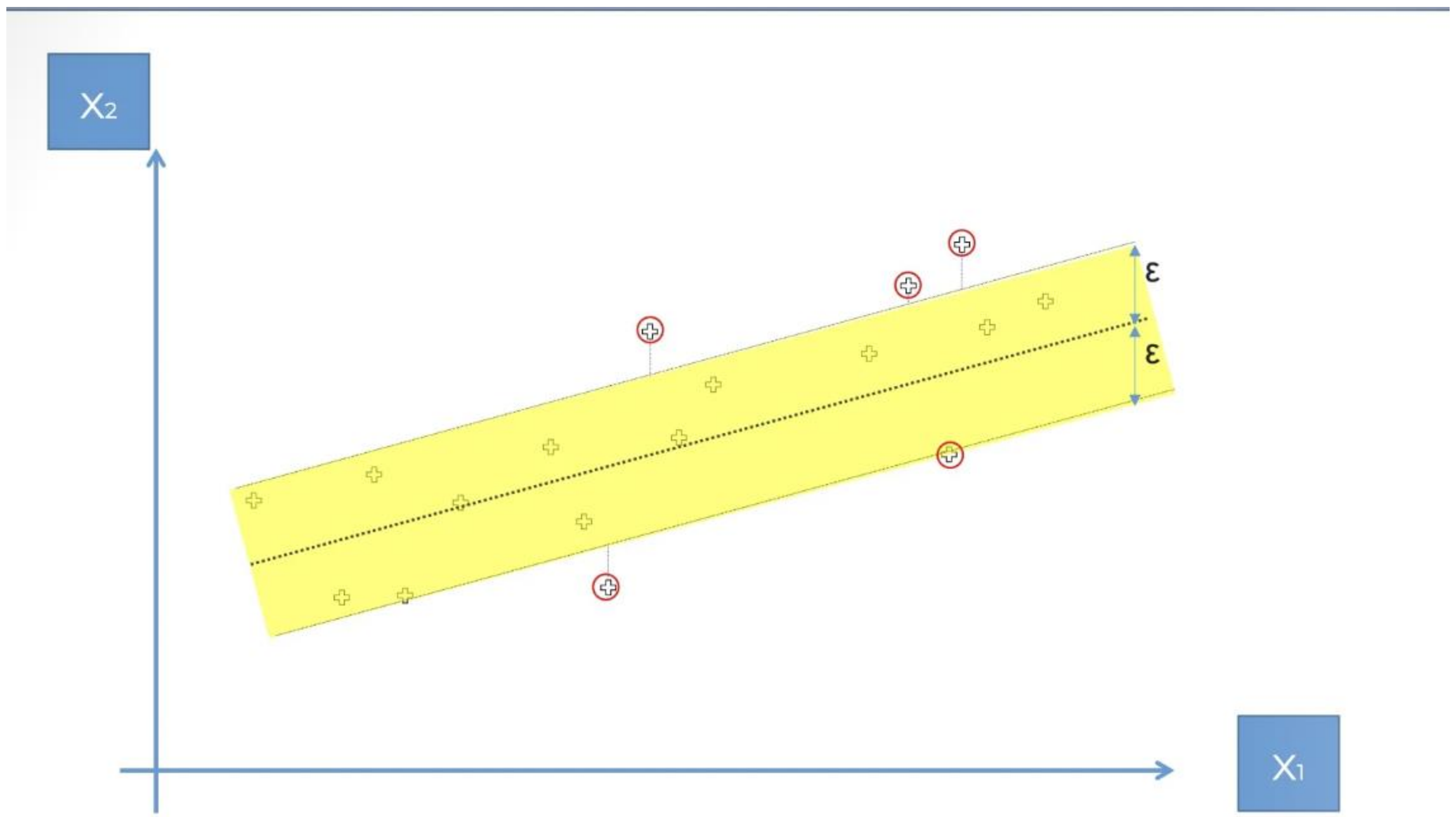

Sometimes, salaries don't increase in a straight line. For example:

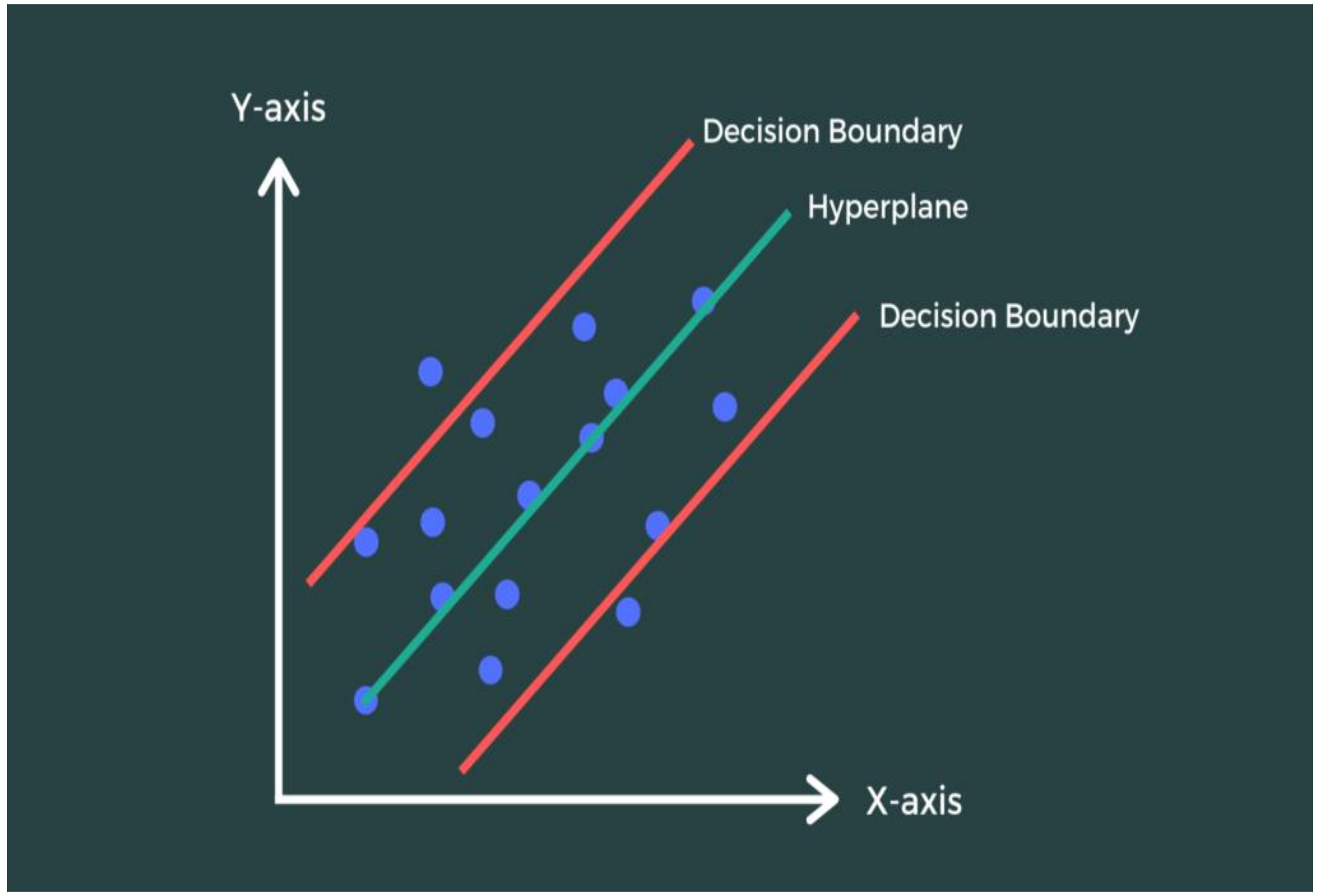
Salary = $b_0$ + $b_1$ × Experience + $b_2$ × (Experience)^2

# Support Vector Regression(SVR)

**SVR** is a type of regression that tries to **predict a continuous value** (like salary), but instead of minimizing the error for each point, it tries to fit the **best line/curve** within a **margin of tolerance (ε)**.

Think of it like:



# Support Vector Regression(SVR)

Minimize:

$$\frac{1}{2}\|w\|^2 + C\sum(\xi_i + \xi_i^*)$$

Subject to:

$$\begin{cases} y_i - w^T x_i - b \le \varepsilon + \xi_i \\ w^T x_i + b - y_i \le \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \ge 0 \end{cases}$$

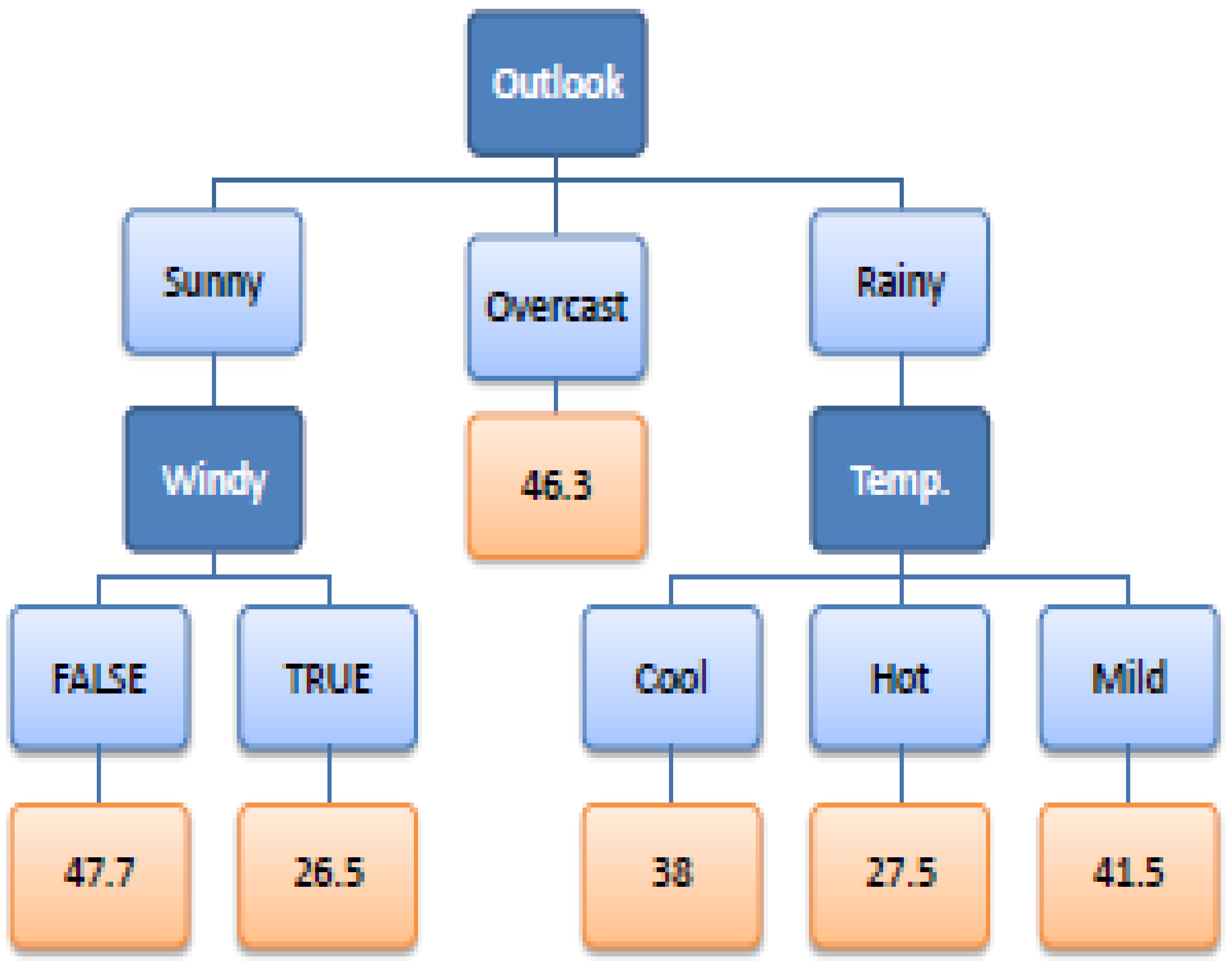
Where:

- $\varepsilon$: tolerance margin
- $\xi_i, \xi_i^*$: slack variables for errors outside ε
- $C$: penalty parameter (controls trade-off between margin and error)

# Decision Tree

Predictors | Target

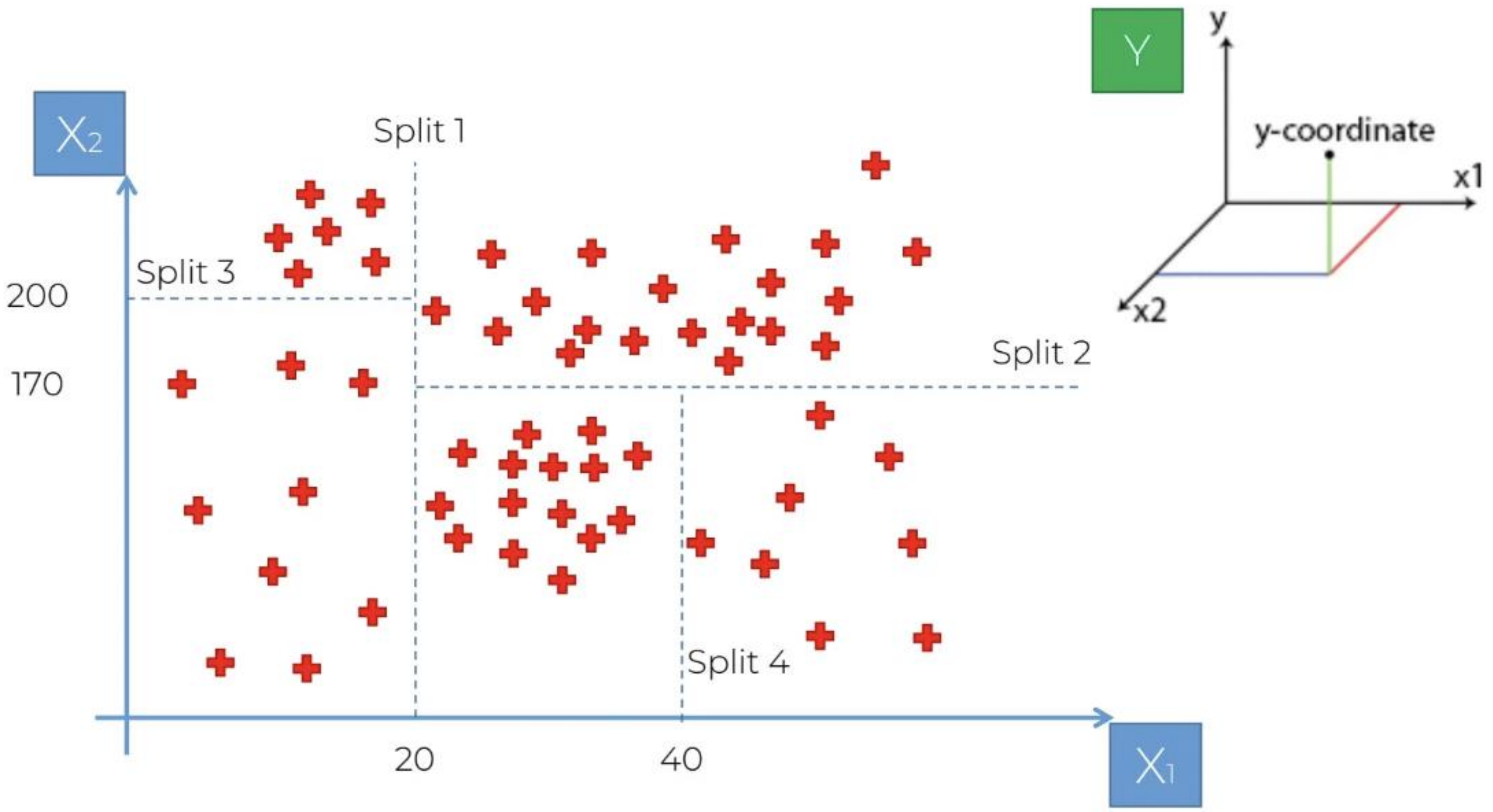
| Outlook | Temp. | Humidity | Windy | Hours Played |
|---|---|---|---|---|
| Rainy | Hot | High | False | 26 |
| Rainy | Hot | High | True | 30 |
| Overcast | Hot | High | False | 48 |
| Sunny | Mild | High | False | 46 |
| Sunny | Cool | Normal | False | 62 |
| Sunny | Cool | Normal | True | 23 |
| Overcast | Cool | Normal | True | 43 |
| Rainy | Mild | High | False | 36 |
| Rainy | Cool | Normal | False | 38 |
| Sunny | Mild | Normal | False | 48 |
| Rainy | Mild | Normal | True | 48 |
| Overcast | Mild | High | True | 62 |
| Overcast | Hot | Normal | False | 44 |
| Sunny | Mild | High | True | 30 |



# Decision Tree

- If **Experience ≤ 2.5**, Salary = ₹3.25 L
- If **2.5 < Experience ≤ 4.5**, Salary = ₹5.25 L
- If **4.5 < Experience ≤ 6.5**, Salary = ₹8.65 L
- If **Experience > 6.5**, Salary = ₹12.0 L

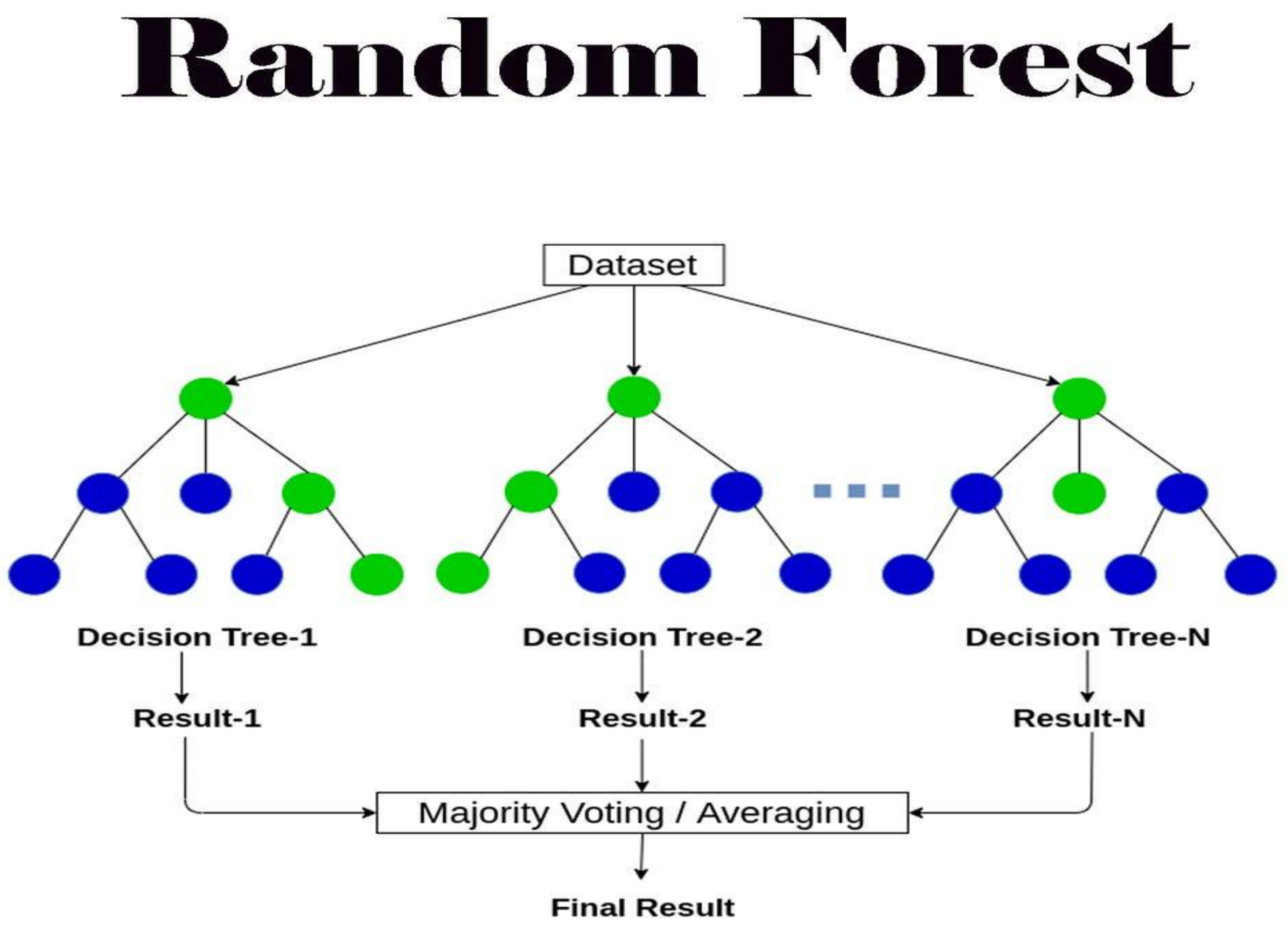
# Decision Tree



# Random Forest Regression

**Random Forest** is a group (or “forest”) of **Decision Trees**.
Instead of relying on just one tree (which might overfit), it:

- Builds **many decision trees** on random subsets of the data.
- Takes the **average** of all tree predictions for better accuracy and **smoother results**.



# Random Forest

Let’s say for a new input:

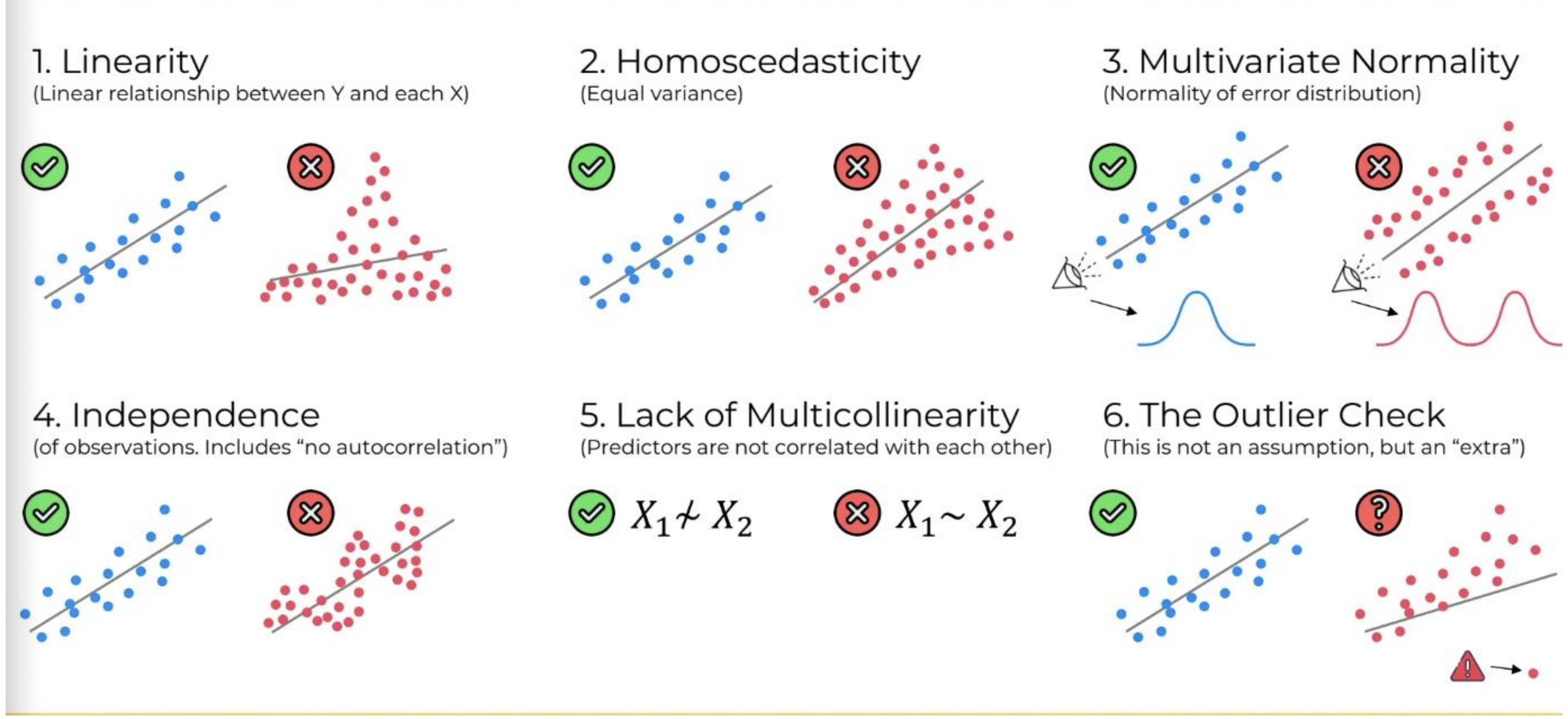
- **Experience = 4.5**, the trees individually predict:
    - Tree 1: ₹6.2 L
    - Tree 2: ₹5.9 L
    - Tree 3: ₹6.0 L

      ... (up to Tree 100)

✅ Final Prediction = **Average of all trees**

→ Salary = ₹6.03 L (more stable than a single tree)

# Assumptions in Linear Regression



# Comparision

| Regression Type | Ideal For | Pros ✅ | Cons ❌ |
|---|---|---|---|
| **Linear Regression** | Predicting a number from **1 input** (linear) | - Simple & fast<br>- Easy to interpret<br>- Good baseline | - Assumes straight-line relationship<br>- Poor for non-linear data |
| **Multiple Linear** | Predicting from **multiple inputs** | - Handles multiple features<br>- Easy to implement | - Still assumes linearity<br>- Sensitive to outliers |
| **Polynomial** | When relationship is **curved** | - Models non-linear trends<br>- Flexible with degree tuning | - Can overfit with high degree<br>- Less interpretable |
| **Support Vector (SVR)** | Data with **noise**, or where **small errors can be ignored** | - Robust to outliers<br>- Ignores small errors (ε margin)<br>- Works with non-linear kernels | - Hard to tune<br>- Slower than linear models<br>- Requires scaling |
| **Decision Tree** | Data with **non-linear splits or categories** | - Easy to understand<br>- No feature scaling needed<br>- Handles non-linear & categorical data | - Overfits easily<br>- Predictions are not smooth (step-like) |
| **Random Forest** | Complex data with noise or many features | - More accurate than single tree<br>- Reduces overfitting<br>- Handles non-linearity well | - Slower<br>- Harder to interpret<br>- Needs more memory |

# Use Cases

| Use-case | Best Regression Type |
|---|---|
| Simple trend prediction (straight line) | Linear Regression |
| Many factors/features involved | Multiple Linear Regression |
| Salary grows slowly, then fast (curved) | Polynomial Regression |
| Ignore small errors and focus on bigger ones | Support Vector Regression |
| Explainable rules and if-else logic | Decision Tree Regression |
| Best performance & generalization | Random Forest Regression |

# More Use Cases

| Regression Type | Industry | Use Case |
|---|---|---|
| **Linear Regression** | HR | Predicting **salary** based on experience |
| | Retail | Forecasting **sales** from advertisement spending |
| | Real Estate | Estimating **house price** from size |
| | Environment | Predicting **temperature** from elevation |
| | Education | Predicting **student scores** from study hours |
| **Multiple Linear** | Real Estate | Predicting **house price** using size, location, and number of rooms |
| | Healthcare | Estimating **medical expenses** from age, BMI, and smoking habits |
| | HR | Forecasting **employee performance** from experience and education |
| | Business | Predicting **monthly revenue** based on sales, marketing, and season |
| | Manufacturing | Estimating **product defect rate** based on materials, speed, and time |
| **Polynomial** | Startups | Modeling **growth rate** over time (e.g., user adoption curve) |
| | Biology | Modeling **enzyme activity** vs temperature |
| | Agriculture | Predicting **crop yield** with weather trends (non-linear) |
| | Education | Modeling **learning curve** over time |
| | Automotive | Predicting **fuel efficiency** at different speeds |
| **Support Vector (SVR)** | Finance | Predicting **stock prices** (with noise tolerance) |
| | IoT / Sensors | Predicting **sensor readings** from noisy inputs |
| | Maintenance | Estimating **machine wear/failure** over time |
| | Transportation | Predicting **travel time** with varying road and traffic conditions |
| | Healthcare | Modeling **patient response** to treatments (ignore small fluctuations) |

# More Use Cases

| | | |
|---|---|---|
| **Decision Tree** | E-commerce | Predicting **purchase value** based on customer profile |
| | Education | Predicting **exam scores** based on attendance and assignment completion |
| | Real Estate | Estimating **rental price** based on location, size, and amenities |
| | Insurance | Predicting **claim amount** from age, vehicle type, and driving history |
| | Agriculture | Estimating **crop disease risk** from soil and weather data |

| | | |
|---|---|---|
| **Random Forest** | Finance | Predicting **loan default** from customer data |
| | Healthcare | Estimating **hospital stay duration** based on condition and history |
| | Marketing | Predicting **customer churn probability** |
| | Real Estate | Predicting **house price** with complex features |
| | Manufacturing | Estimating **product lifespan** from materials, usage, and environment |

# Metrics for Regression – Detailing Next Session

- Mean Absolute Error
- Mean Squared Error
- Root Mean Square Error
- Root Mean Square Logarithmic Error
- R2-Score

# Notebook

Kaggle :

https://www.kaggle.com/code/ohanvi/

GitHub:

https://github.com/Ohanvi/machine-learning-module

Competition:

https://www.kaggle.com/competitions/ml-regression-salary-prediction-challenge

https://www.kaggle.com/competitions/predict-the-closing-stock-price

# Donate to India Army

- [Indian Army]
- [NDF - National Defense Fund]


| | | |
|---|---|---|
| (a) | **Name of Fund** | **: Army Central Welfare Fund.** |
| | Bank Name | : Union Bank of India |
| | Branch | : Chandni Chowk, Delhi – 110006 |
| | IFSC Code | : UBIN0530778 |
| | Account No | : 520101236373338 |
| | Type of Acct | : Saving |
| (b) | **Name of Fund** | **: Armed Forces Battle Casualties Welfare Fund.** |
| | Bank Name | : Canara Bank, |
| | Branch | : South Block, Defence Headquarters, New Delhi - 110011 |
| | IFSC Code | : CNRB0019055 |
| | Account No | : 90552010165915 |
| | Type of Acct | : Saving |

# The End