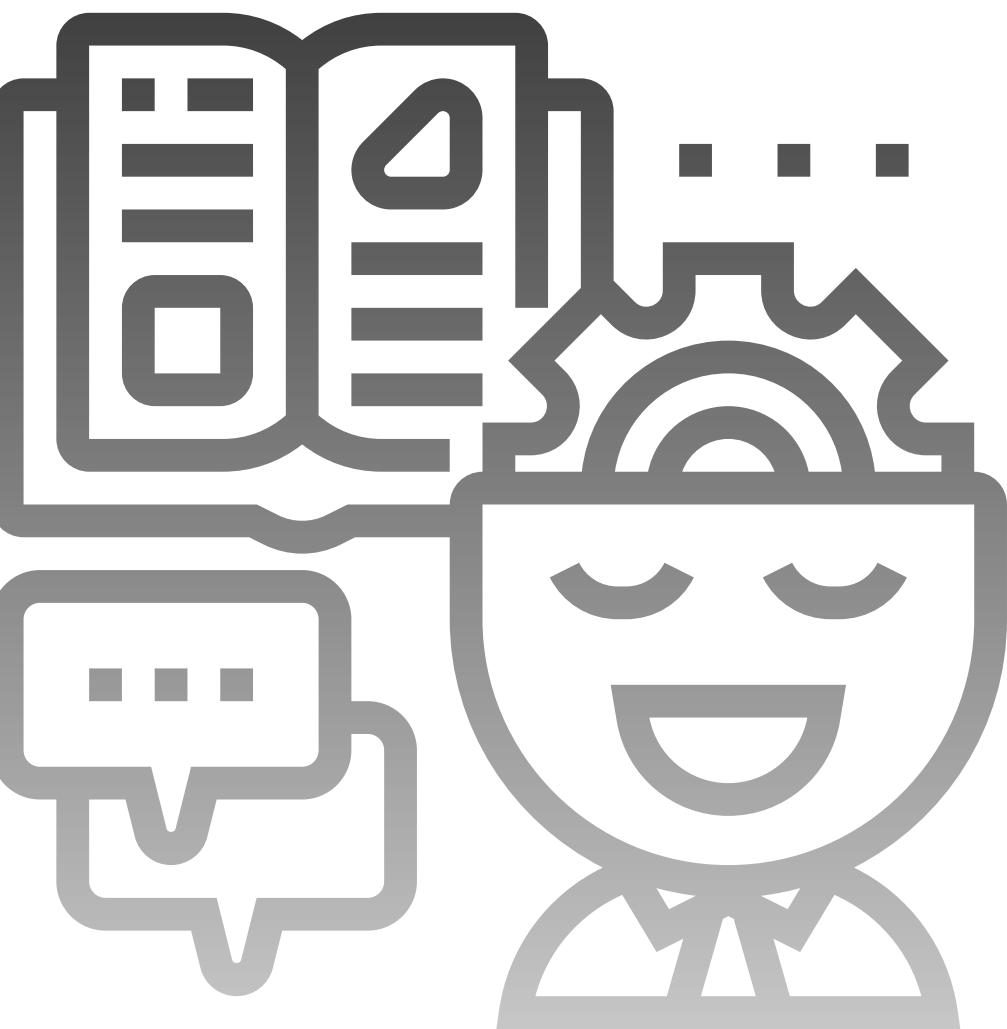




INT395- SUPERVISED ML

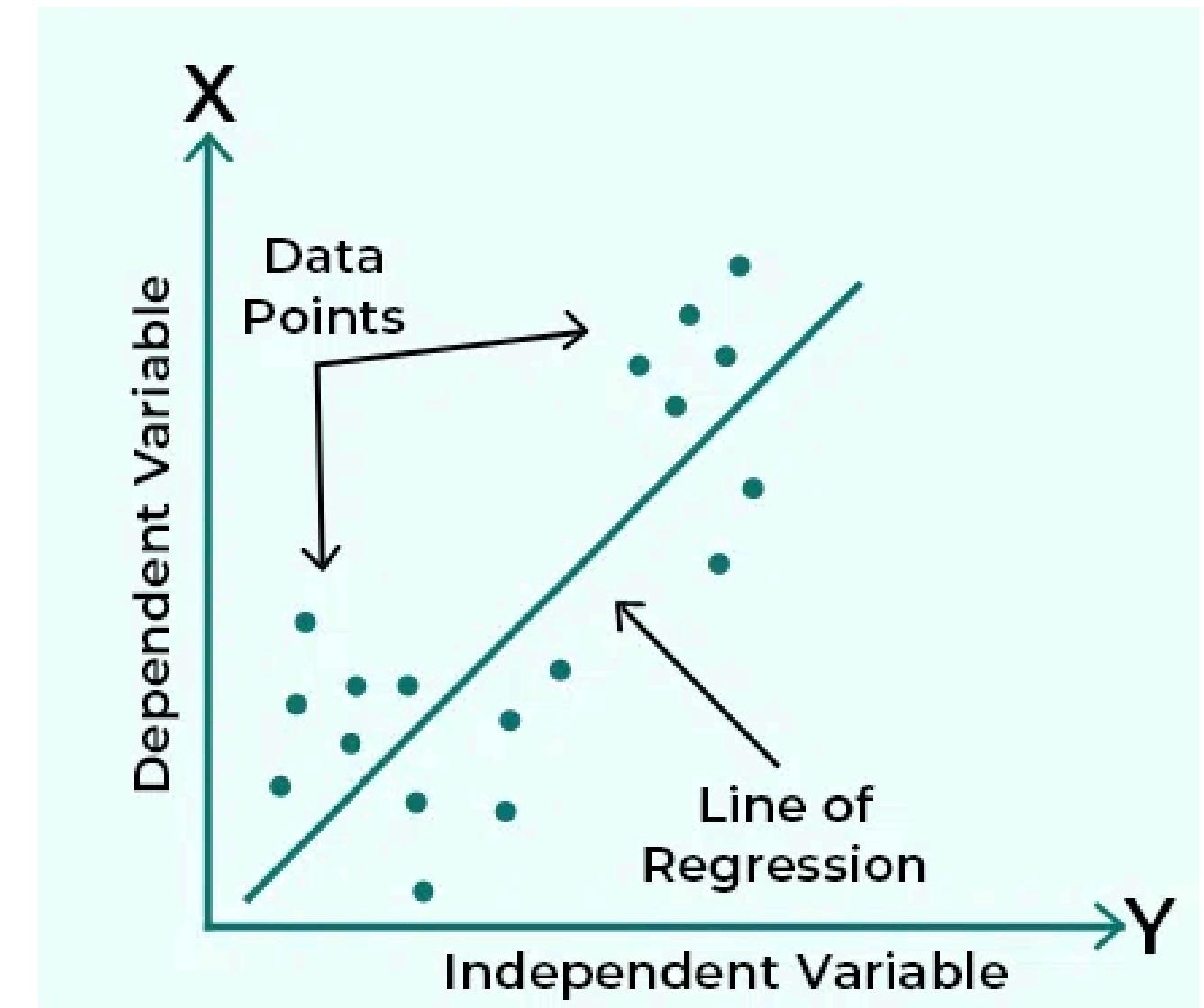
Unit 4: Regression

Presented By: Blossom Kaler
Assistant Professor
SCAI, LPU



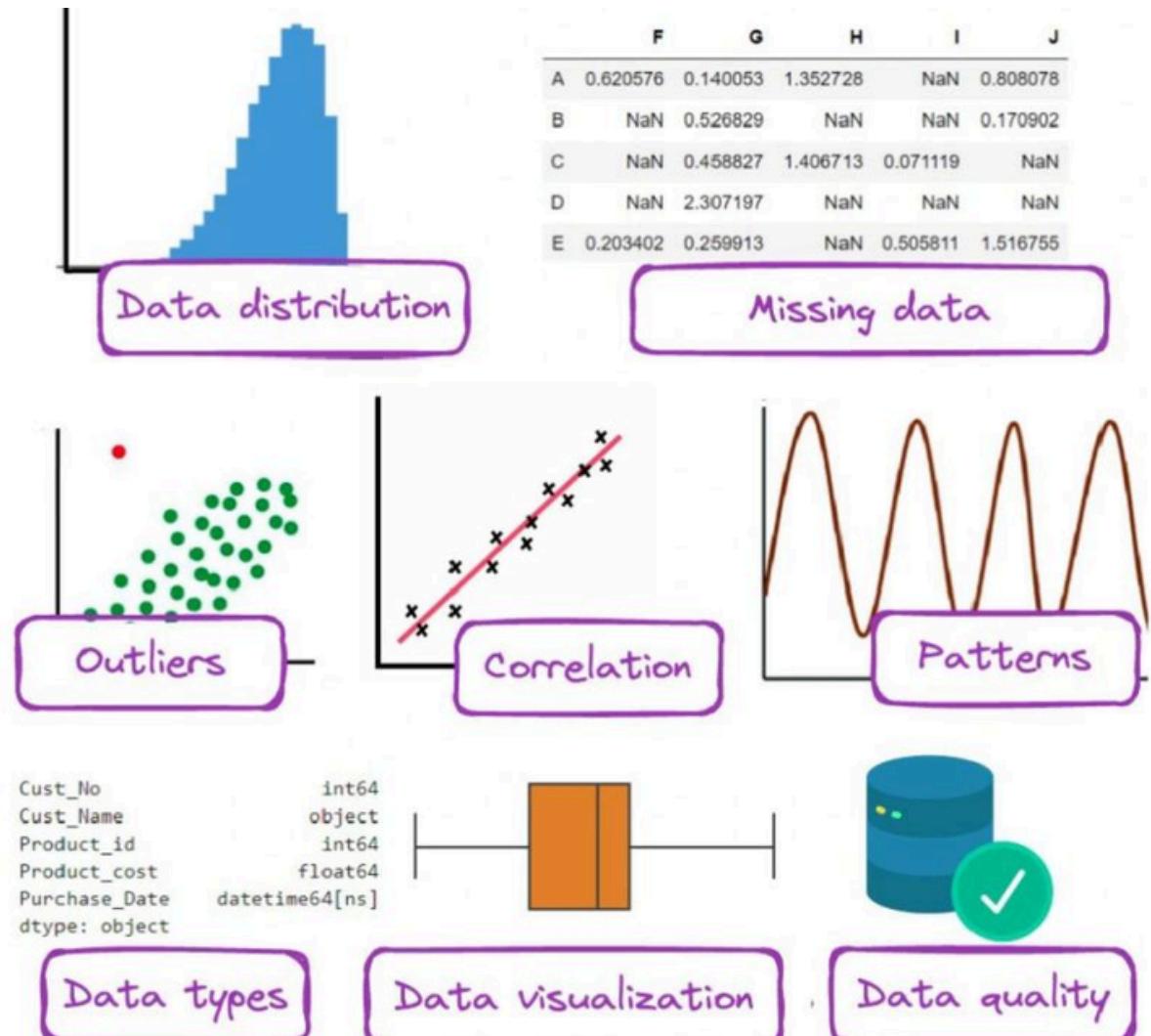
OVERVIEW OF REGRESSION

- Regression predicts a continuous output.
- Models relationship between input features and target variable.
- Used for forecasting, trend analysis, risk estimation.
- Answers "How much?" type questions.
- Works by fitting a best-fit function to data.
- Used in house price prediction, sales forecasting, medical risk scores.
- Focuses on minimizing prediction error.



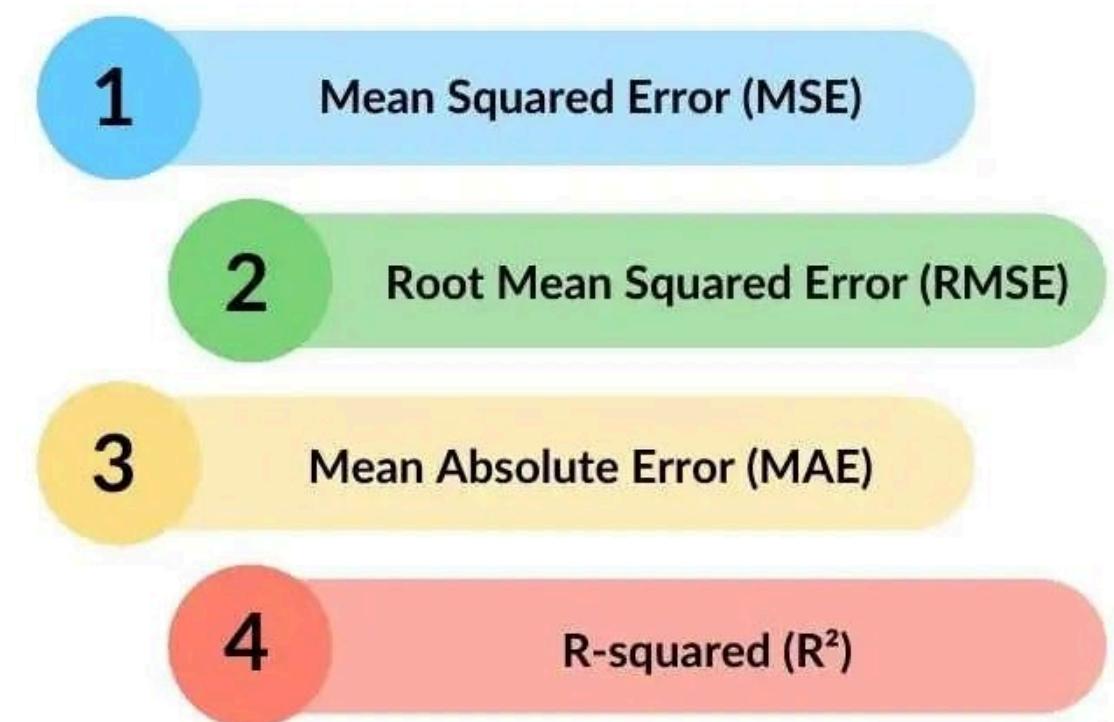
EXPLORATORY DATA ANALYSIS

- Understand data distribution and feature-target relationships.
- Check missing values, outliers, skewness.
- Study correlations between variables.
- Visualize patterns using scatter plots, histograms, boxplots.
- Helps decide preprocessing steps & suitable regression model.
-



EVALUATION METRICS

- Regression models predict continuous numeric values.
- Evaluating their performance requires quantitative metrics.
- Metrics measure how close predicted values are to actual targets.
- **Common objectives:**
 - Minimize errors
 - Understand variance explained by the model
- **Choice of metric depends on:**
 - Sensitivity to outliers
 - Scale of target variable
 - Model interpretability needs



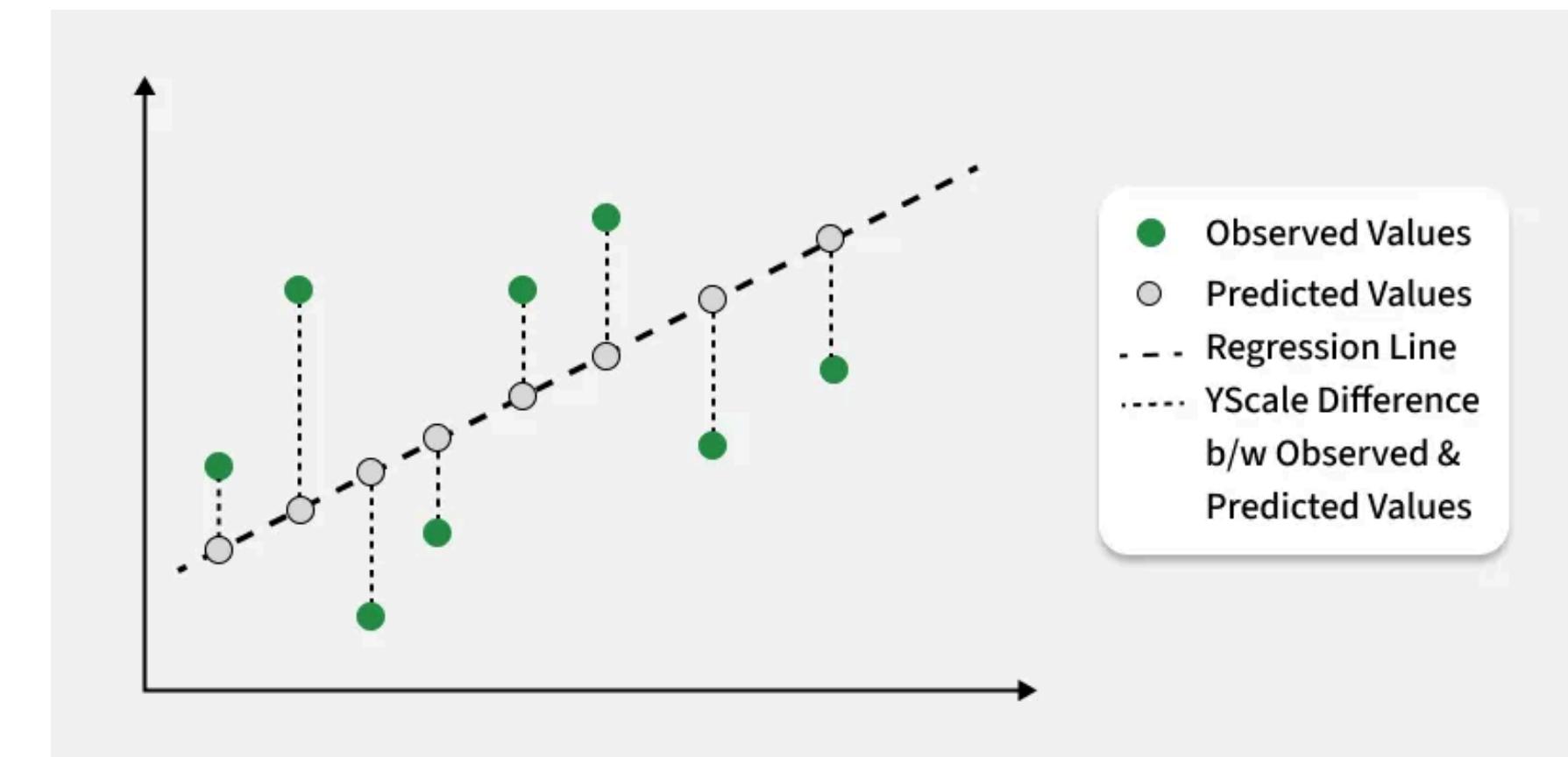
MEAN SQUARED ERROR (MSE)

- Measures average squared difference between: Actual value (y) and Predicted value (\hat{y})
- Used in regression problems
- It penalizes large errors more

$$\text{Mean Squared Error} = \frac{1}{n} \sum_{i=1}^n (Y_i - \hat{Y}_i)^2$$

Where:

- n is the number of observations in the dataset.
- y_i is the actual value of the observation.
- \hat{Y}_i is the predicted value of the i^{th} observation.



MEAN SQUARED ERROR

- Actual Values: [10, 20, 30, 40, 50]
- Predicted Values: [12, 18, 32, 38, 48]

Solution:

To calculate MSE we first compute the squared differences between the each actual and predicted value:

$$\begin{aligned} \text{Squared Differences: } & [(10-12)^2, (20-18)^2, (30-32)^2, (40-38)^2, (50-48)^2] \\ & = [4, 4, 4, 4, 4] \end{aligned}$$

Next, we take the average of these squared differences to the obtain the MSE:

$$\begin{aligned} \text{MSE} &= (4 + 4 + 4 + 4 + 4) / 5 \\ &= 20 / 5 \\ &= 4 \end{aligned}$$

Therefore, the MSE for this regression model is 4.

ROOT MEAN SQUARED ERROR (RMSE)

- Measures the square root of the average squared difference between actual and predicted values.
- Penalizes large errors more heavily due to squaring.
- Sensitive to outliers; best when large deviations matter.
- Expressed in the same units as the target variable.
- Lower RMSE → better model performance.

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

$\hat{y}_1, \hat{y}_2, \dots, \hat{y}_n$ are predicted values

y_1, y_2, \dots, y_n are observed values

n is the number of observations

MEAN ABSOLUTE ERROR (MAE)

- Measures the average absolute difference between actual and predicted values.
- Treats all errors equally; no extra penalty for large errors.
- More robust to outliers compared to RMSE.
- Easy to interpret (average error magnitude).
- Lower MAE → better performance.

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

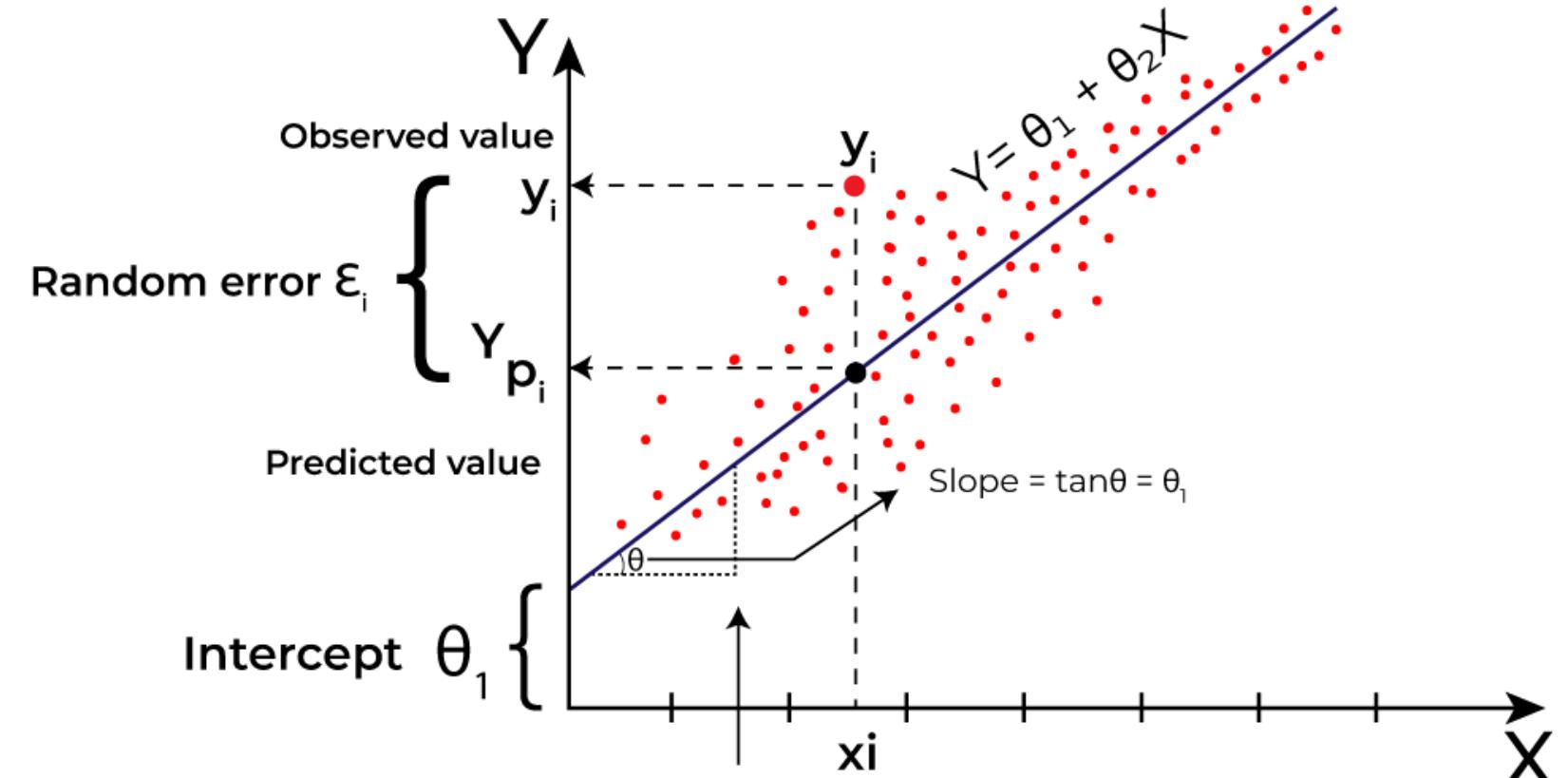
R² SCORE (COEFFICIENT OF DETERMINATION)

- Measures how much variance in the target is explained by the model.
- Ranges from $-\infty$ to 1
- Indicates model's goodness of fit but not error magnitude.
- R^2 measures how much variance in the target variable is explained by the model.
- Higher $R^2 \rightarrow$ model explains more variability \rightarrow better fit.
- $R^2 = 1 \rightarrow$ perfect model
- $R^2 = 0 \rightarrow$ model is no better than predicting the mean
- $R^2 < 0 \rightarrow$ model is worse than predicting the mean

$$R^2 = 1 - \frac{\sum_i^N (y_i - \hat{y}_i)^2}{\sum_i^N (y_i - \bar{y})^2}$$

LINEAR REGRESSION

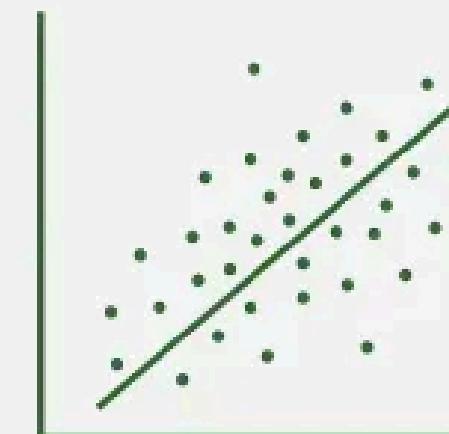
- Predicts continuous output using input features.
- Fits a line to minimize prediction errors.
- Equation: $y=mx+c$ (simple case).
- Intercept is predicted value when features = 0
- Uses least squares to minimize prediction errors.
- Calculates best-fit line for observed data.
- Handles single and multiple features (simple & multiple regression).



MULTILINEAR REGRESSION

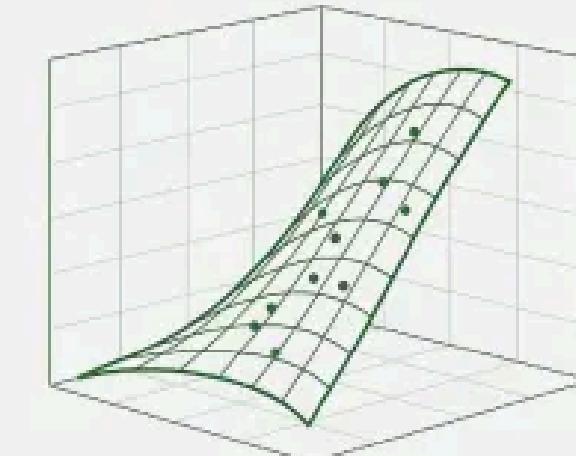
Types of Linear Regression

Simple Linear Regression



Predicts the dependent variable using
a single independent variable.

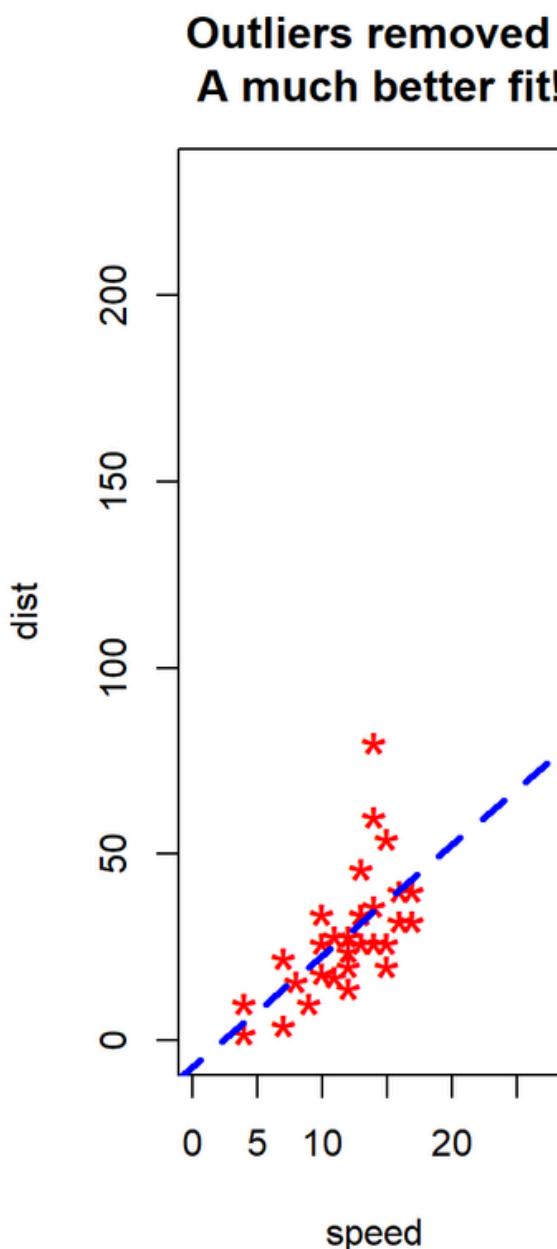
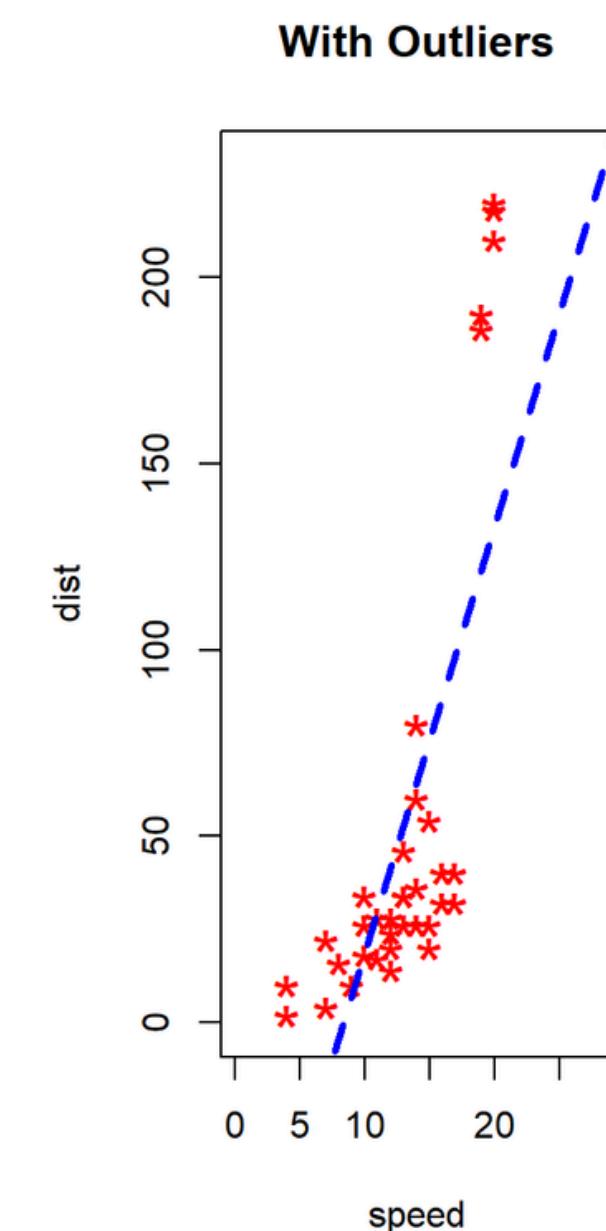
Multiple Linear Regression



Uses two or more independent variables
to predict the dependent variable.

RANSAC MODEL

- RANSAC stands for Random Sample Consensus
- Used for robust regression in presence of outliers
- Assumes data contains inliers + outliers
- Randomly selects a minimum subset of data points
- Fits a model using the selected subset
- Checks how many points agree with the model (inliers)
- Repeats process for fixed number of iterations



RANSAC MODEL

Dataset: (1,2), (2,4), (3,6), (4,8), (5,10), (3,20), (4,25). Threshold: error ≤ 2

- **Iteration 1**

- Random points selected: (1,2), (2,4)
- Fitted model: $y = 2x$
- **Inliers**: (1,2), (2,4), (3,6), (4,8), (5,10)
- **Outliers**: (3,20), (4,25)
- Inlier count = 5

- **Iteration 2**

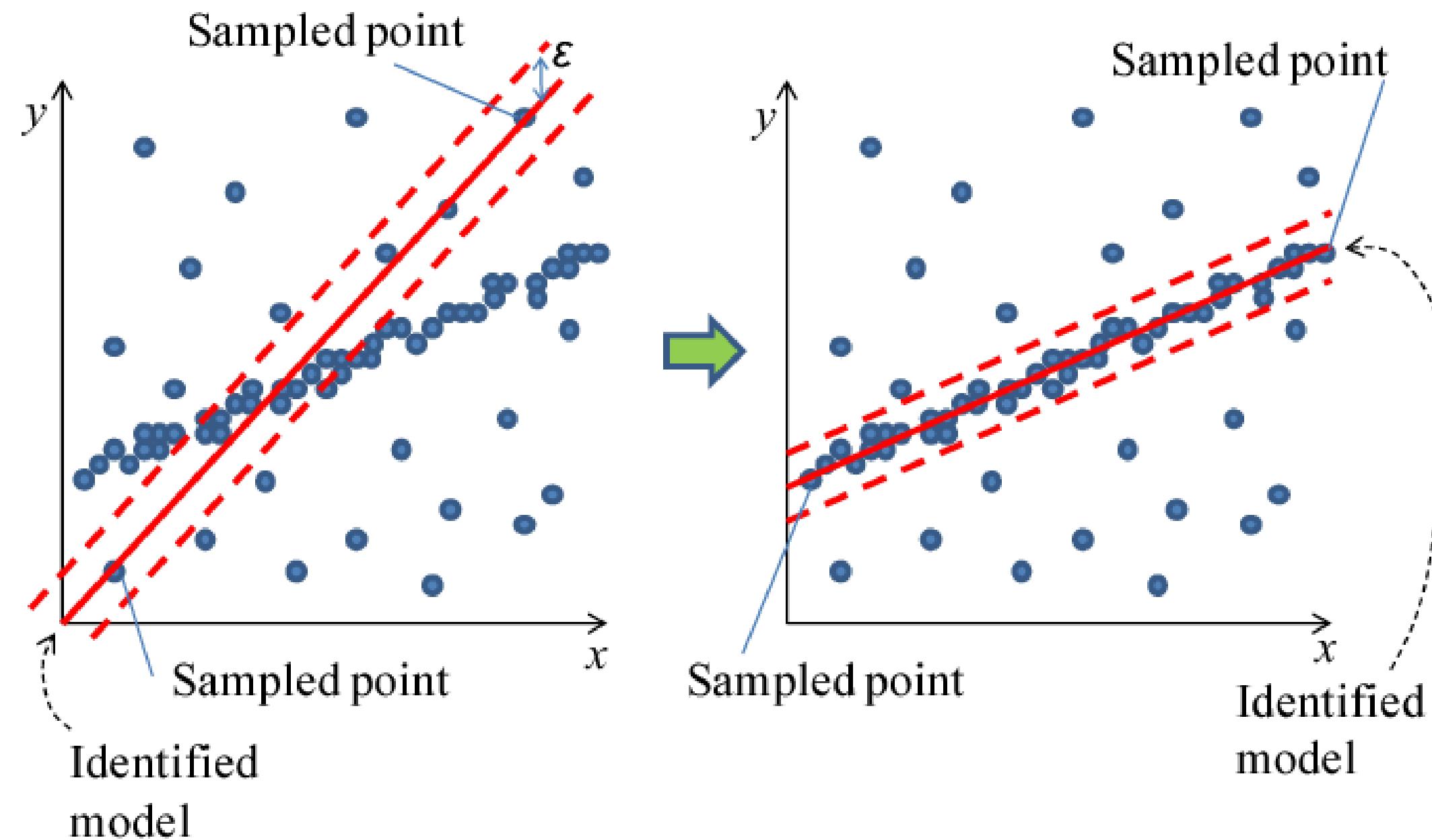
- Random points selected: (3,20), (4,25)
- Fitted model: $y = 5x + 5$
- **Inliers**: (3,20), (4,25)
- **Outliers**: (1,2), (2,4), (3,6), (4,8), (5,10)
- Inlier count = 2

- **Final Decision**

- Model with maximum inliers (5) is selected
- **Final RANSAC model: $y = 2x$**
- Outliers are ignored

Point	Predicted $y (2x)$	Error	Inlier / Outlier
(1,2)	2	0	Inlier
(2,4)	4	0	Inlier
(3,6)	6	0	Inlier
(4,8)	8	0	Inlier
(5,10)	10	0	Inlier
(3,20)	6	14	Outlier
(4,25)	8	17	Outlier

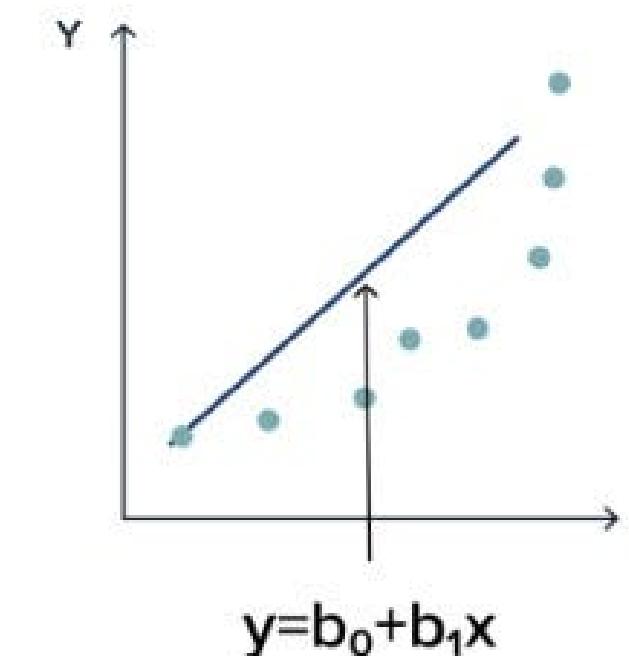
RANSAC MODEL



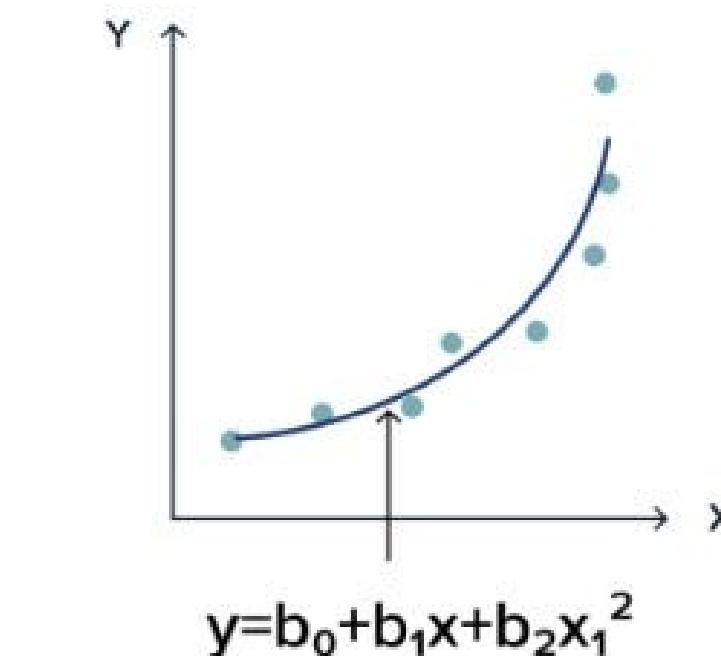
POLYNOMIAL REGRESSION

- Captures non-linear patterns.
- Extends linear regression by adding polynomial terms.
- Example: $y = a + b_1x + b_2x^2 + b_3x^3 \dots$
- Allows flexible curve fitting.
- Controlled by degree of polynomial.
- Higher degrees increase flexibility.
- Risk of overfitting.

Simple linear model

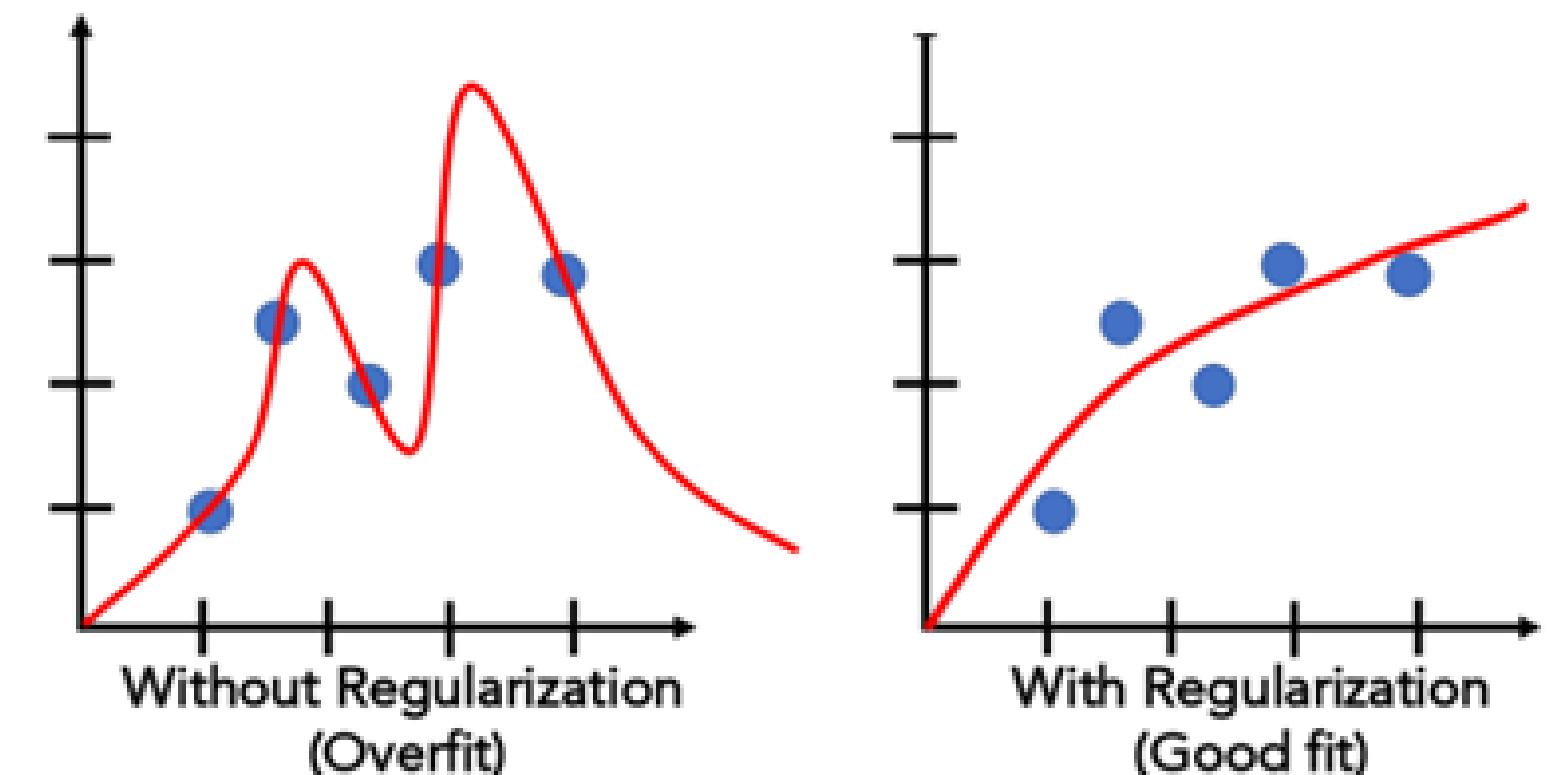


Polynomial model



REGULARIZED REGRESSION

- Regularization is a technique to reduce overfitting by discouraging overly complex models
- Adds a penalty term to the loss function → controls model complexity
- Prevents large weights → improves generalization
- Common types: L1 (Lasso) and L2 (Ridge)
- Helps: smoother decision boundaries, stable predictions, reduced variance
- Balances bias–variance tradeoff using hyperparameter λ / α
- Higher regularization → simpler model; Lower → more flexible model



L1 REGULARIZATION (LASSO)

- Adds penalty = $\lambda \times \sum |\text{weights}|$ (sum of absolute values)
- Encourages sparse models → many weights become exactly zero
- Performs feature selection automatically
- Robust to outliers due to absolute penalty
- Useful when many irrelevant features exist
- Loss function: Loss = MSE + $\lambda \sum |w_i|$

$$\text{Cost} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \lambda \sum_{i=1}^m |w_i|$$

Where

- m - Number of Features
- n - Number of Examples
- y_i - Actual Target Value
- \hat{y}_i - Predicted Target Value

L2 REGULARIZATION (RIDGE)

- Adds penalty = $\lambda \times \sum (\text{weights}^2)$ (sum of squared weights)
- Produces smooth, stable solutions; handles correlated features well
- Reduces variance without sacrificing too much bias
- Loss function: Loss = MSE + $\lambda \sum w_i^2$
- Preferred when most features are useful and multicollinearity exists

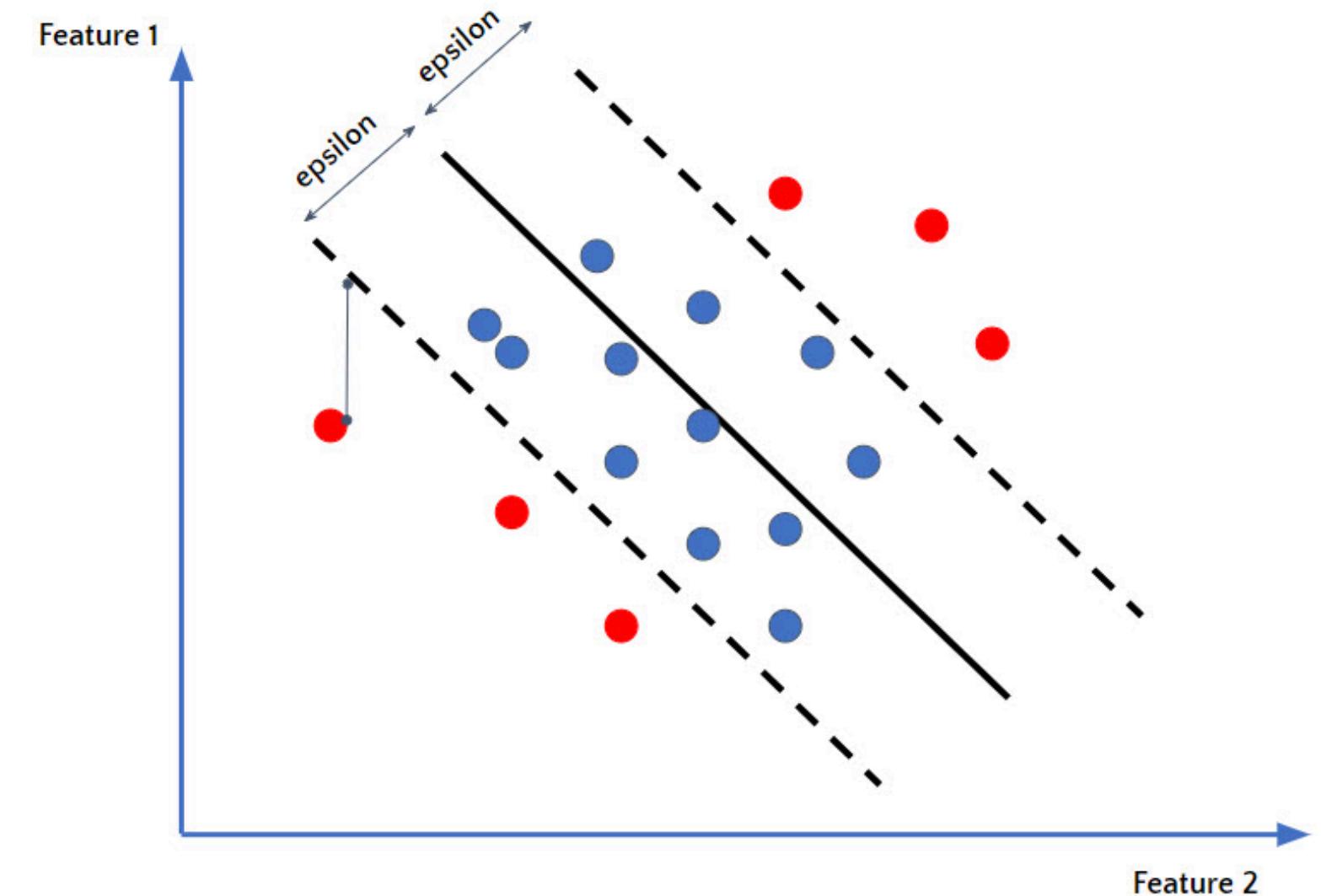
$$\text{Cost} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 + \lambda \sum_{i=1}^m w_i^2$$

Where,

- n = Number of examples or data points
- m = Number of features i.e predictor variables
- y_i = Actual target value for the i^{th} example
- \hat{y}_i = Predicted target value for the i^{th} example
- w_i = Coefficients of the features
- λ = Regularization parameter that controls the strength of regularization

SUPPORT VECTOR REGRESSION

- **Key concept: ϵ -insensitive tube**
- SVR tries to fit a line such that most data points lie within a tube of width 2ϵ around it.
- Errors inside this tube are ignored (considered acceptable).
- Only points outside the tube contribute to the loss.
- SVR also uses the concept of support vectors:
- Only a few crucial data points (those outside or on the boundary of the tube) determine the final model.
- SVR finds a function (line/curve) that approximates numeric outputs while maintaining a margin of tolerance.
- SVR can be:
- Linear (straight lines) and Non-linear (using kernels: RBF, polynomial, etc.)



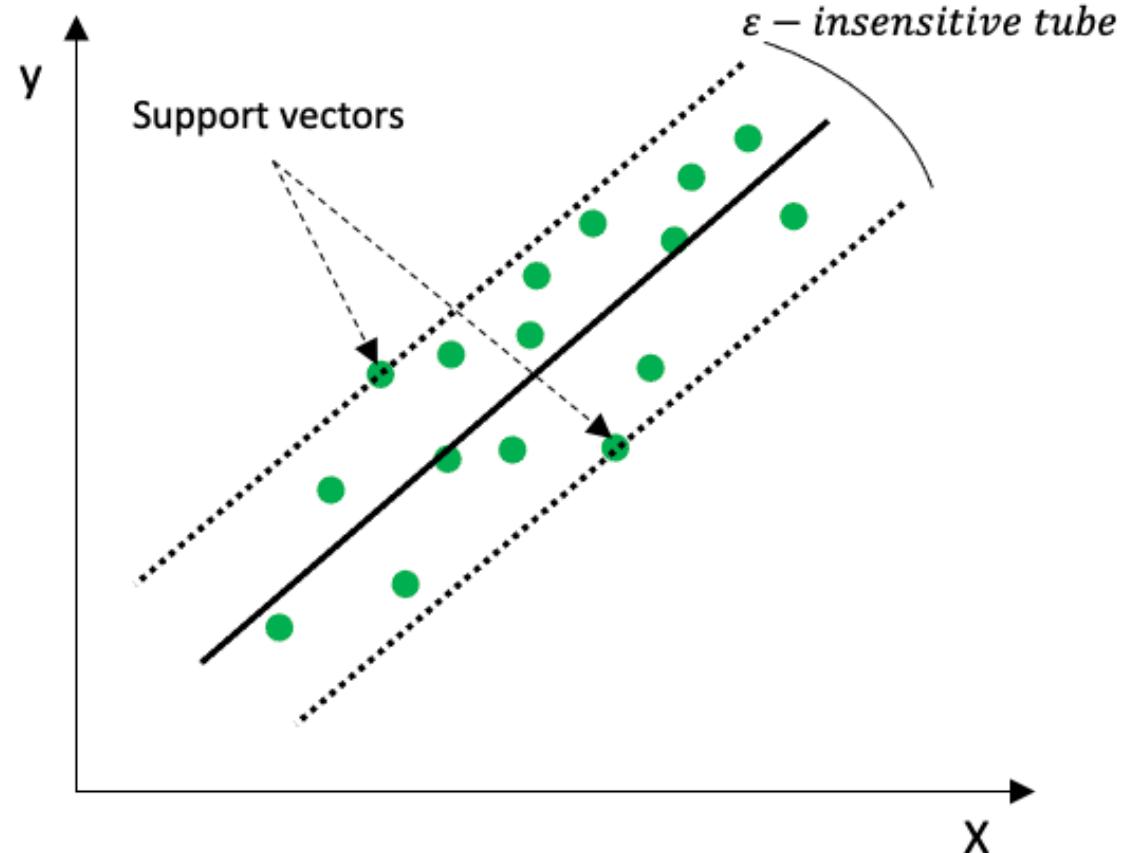
SUPPORT VECTOR REGRESSION

- **ANALOGY:**

- Imagine a bus that is scheduled to arrive at 8:00 AM.
- If bus comes between 7:58 and 8:02, passengers are okay. This is the acceptable margin (ε).
- If it comes at 8:15 or 7:45 → now passengers complain (error is "too big").

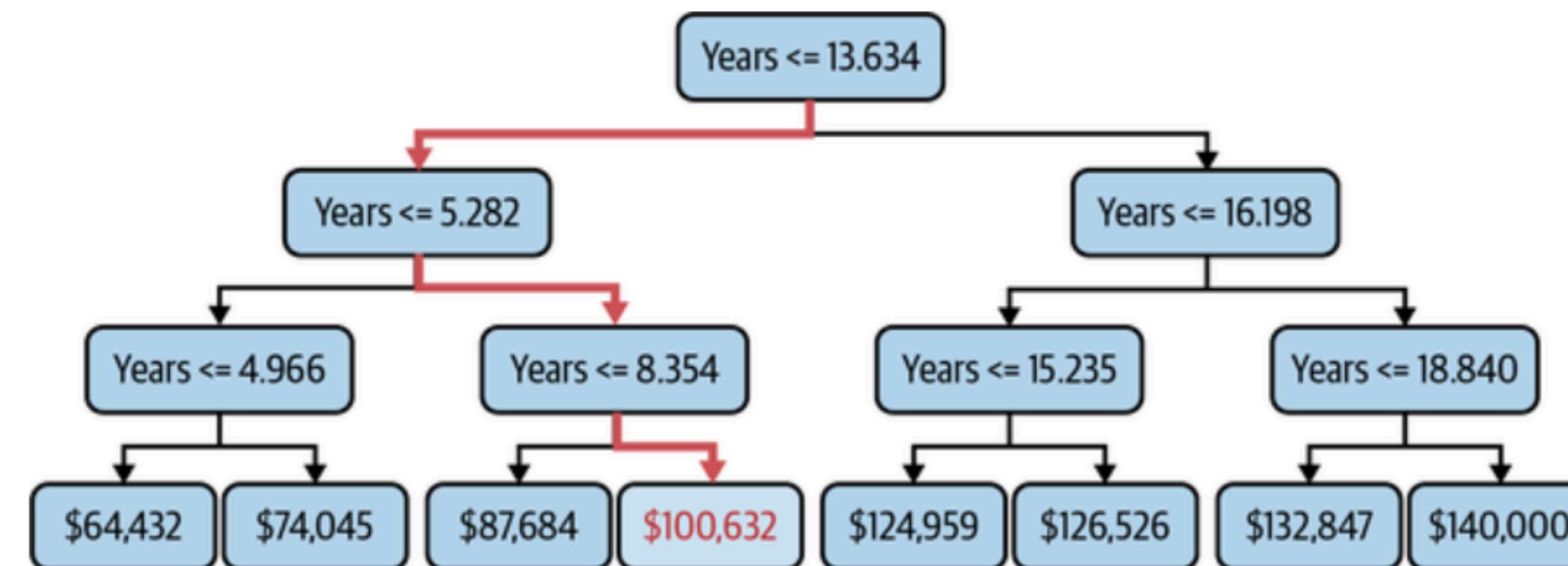
- **SVR works similarly:**

- It draws a prediction curve with a small tolerance band around it (the tube).
- As long as predictions fall inside this band (close enough to real values), it doesn't worry too much.
- Only large deviations are penalized.
- Tube width $\rightarrow \varepsilon$
- Points within tube \rightarrow "on time enough"
- Points outside \rightarrow "significant errors" \rightarrow influence the model.



DECISION TREE REGRESSION

- Decision Tree Regression is the regression version of Decision Trees.
- Instead of predicting a class at each leaf,
- the tree predicts a numeric value
- (typically the average of target values in that leaf).



Refer the numerical shared with this ppt!

DECISION TREE REGRESSION

Imagine a real estate agent with a mental decision tree:

1. “*Is the area > 1200 sq ft?*”

If NO → then price around ₹40–50 lakhs.

If YES →

2. “*Is it in city center?*”

If YES → price around ₹90–100 lakhs

If NO → price around ₹70–80 lakhs

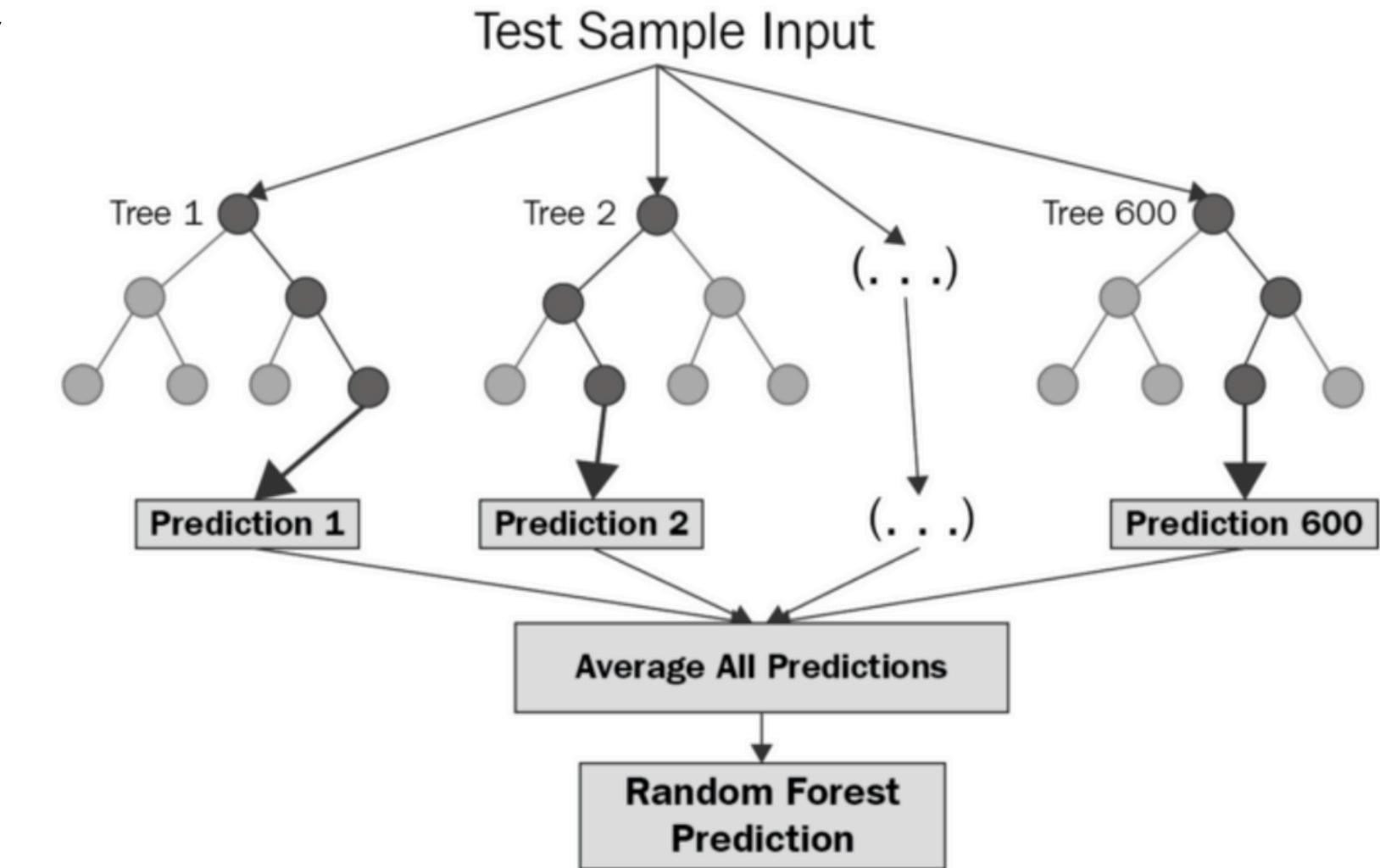
At each step, the agent splits based on a feature and narrows down to a **range of prices**.

This is exactly what a **regression tree** does:

***It splits data into regions where values are similar,
And predicts the average value in each region.***

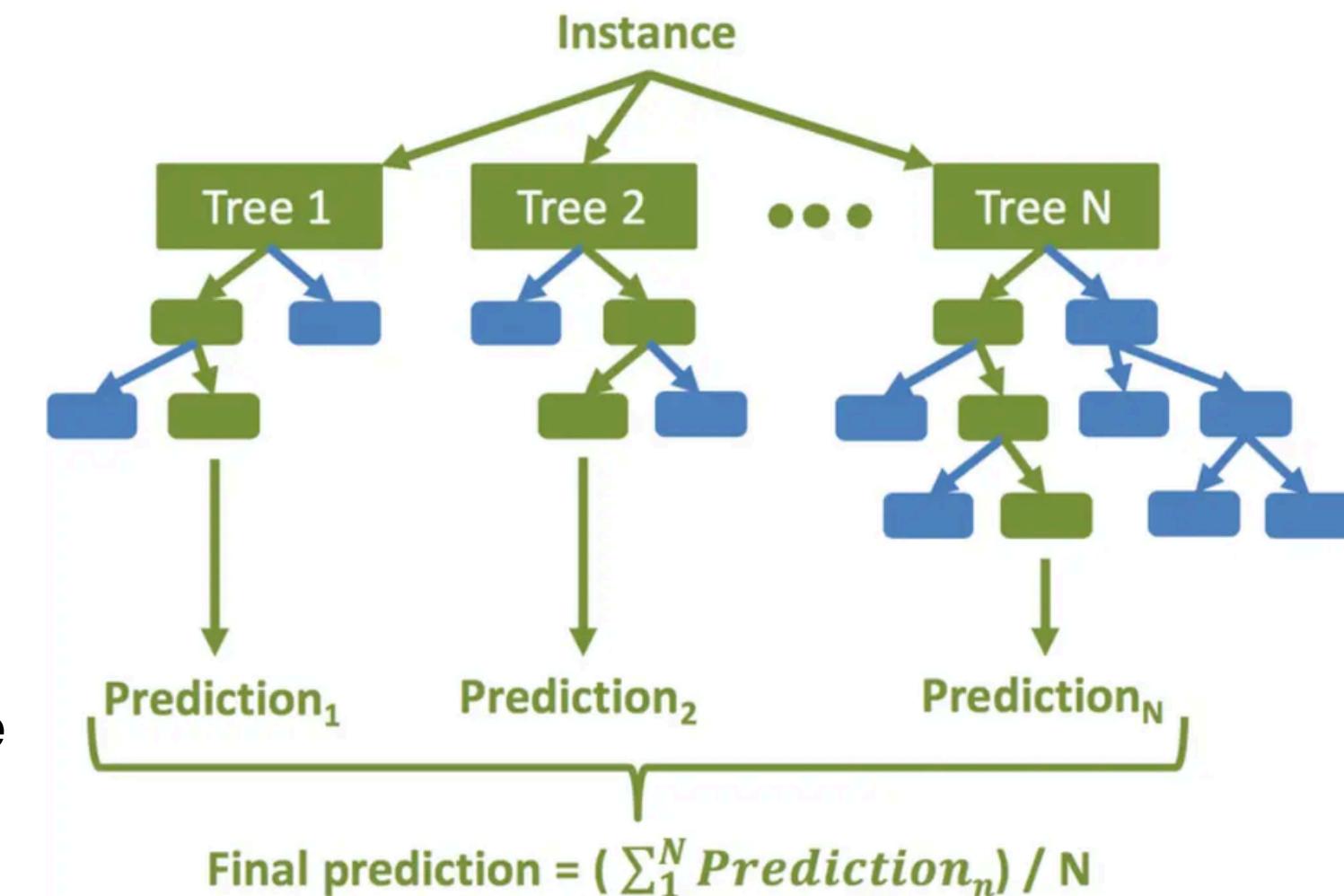
RANDOM FOREST REGRESSION

- Random Forest Regression is an ensemble of many decision trees, used for predicting continuous values.
 - It combines:
 - Bagging (Bootstrap Aggregation), and
 - Random feature selection at splits.
-
- **Steps:**
 - Draw many bootstrap samples from the training data.
 - Train one decision tree regressor on each sample.
 - At each split, each tree considers only a random subset of features (this decorrelates trees).
 - For prediction: Each tree outputs a numeric prediction.
 - The final prediction is the average of all tree predictions.



RANDOM FOREST REGRESSION

- Instead of asking just one property agent for a house price estimate, you ask 30 different agents, each with slightly different experiences and biases.
- Agent 1: ₹70 lakhs
- Agent 2: ₹75 lakhs
- Agent 3: ₹80 lakhs
- ...
- Agent 30: ₹77 lakhs
- **You then average all estimates, say ₹76 lakhs.**
- Individual agents (like individual trees) can be:
- **Biased, Overfitted to specific experiences**
- But when you combine many independent opinions, the result is:
- **More stable, More accurate, Less sensitive to outliers**
- That is exactly what Random Forest does mathematically.





LOVELY
PROFESSIONAL
UNIVERSITY

NAAC
GRADE
A++

THANK YOU