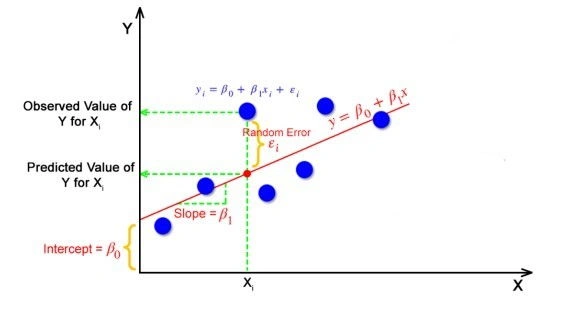
#### **What is Linear Regression?**

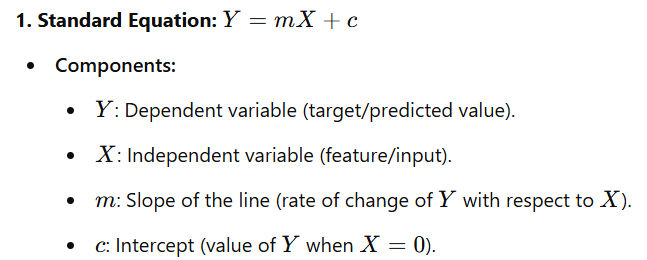
## Linear regression is a statistical method used to model the relationship between a dependent variable (target) and one or more independent variables (features). It is one of the simplest and most widely used regression techniques in machine learning.

## **Simple Linear Regression** involves one independent variable.

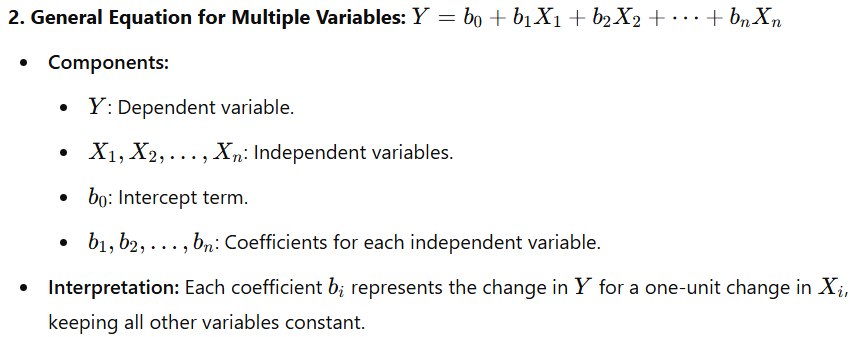
## **Multiple Linear Regression** involves multiple independent variables.

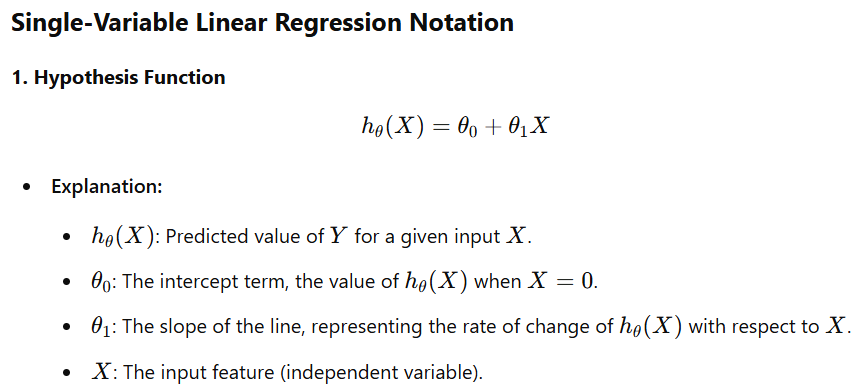
## Linear regression **assumes a linear relationship** between the dependent and independent variables. The goal is to find the line (or hyperplane in multiple dimensions) that best fits the data.





* **Interpretation:** This is the most straightforward representation for a single-variable linear regression (simple linear regression). The slope m indicates how much Y changes for a unit increase in X.

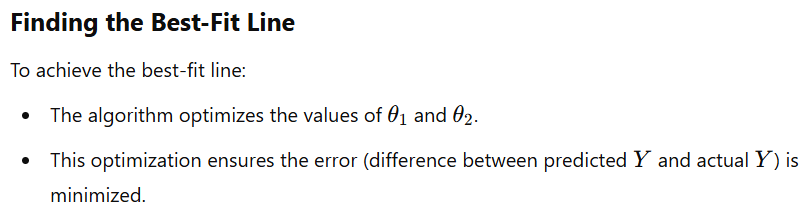


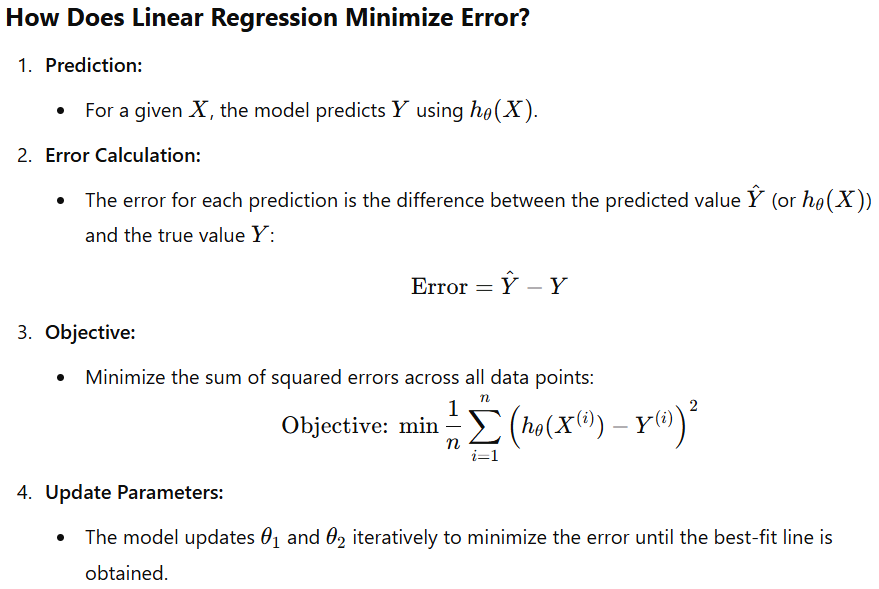


This notation is most commonly used.

### **Goal of Linear Regression**

The primary goal of linear regression is to find a **best-fit line** (linear relationship) that minimizes the error between the predicted values and the true values.





### **Cost Function**

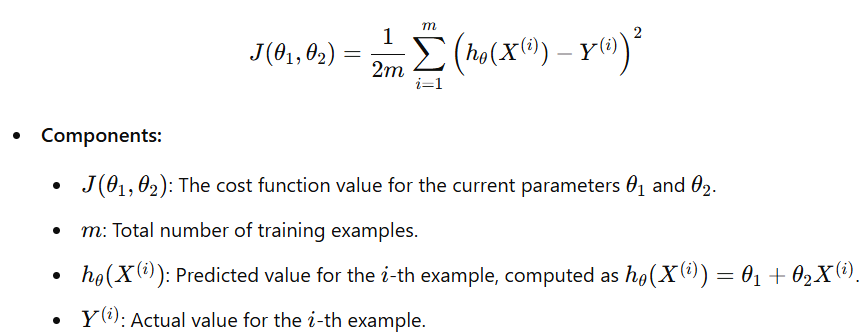
The **cost function or loss function** is a mathematical tool used to measure the error between the predicted values and the actual values. In linear regression, the goal of the cost function is to quantify how well the linear model predicts the target variable.

### **Why Do We Need a Cost Function?**

1. **Evaluate Model Performance:**The cost function helps us determine whether our current choice of parameters (θ1​ and θ2​) provides accurate predictions.
2. **Optimization Goal:**The cost function provides a value that needs to be minimized to achieve the best-fit line.

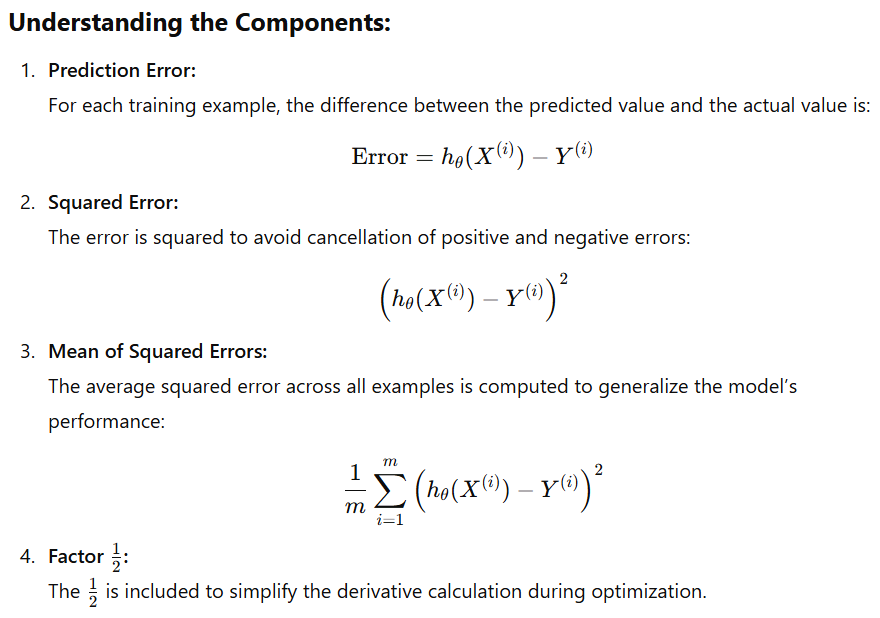
### **Mean Squared Error (MSE)**

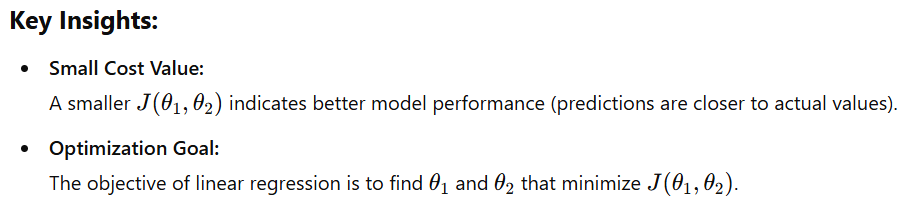
The most commonly used cost function in linear regression is the **Mean Squared Error** (MSE). It measures the average of the squared differences between predicted and actual values.



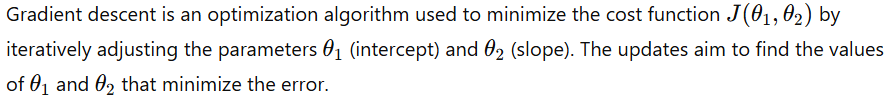
#### **Why Squared Error?**

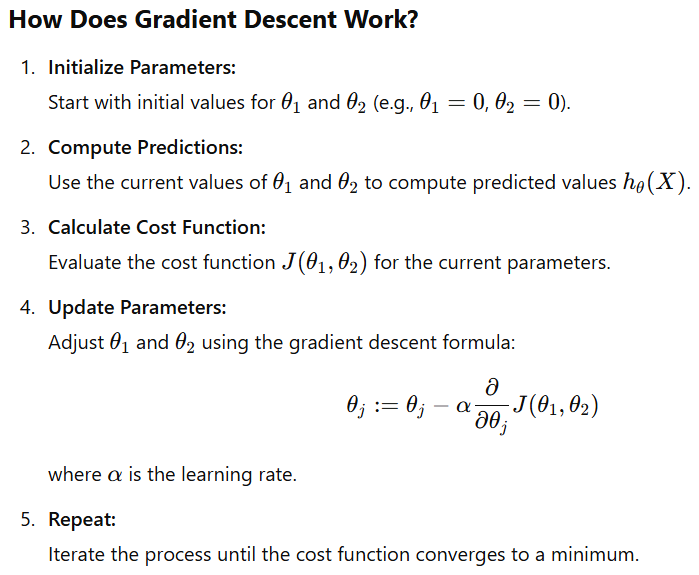
* Squaring the error ensures it is always positive.
* Emphasizes larger errors more than smaller ones, making the model focus on minimizing significant discrepancies.





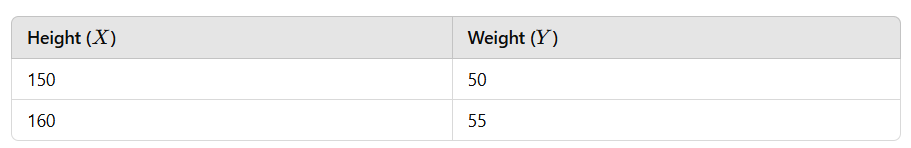
### **Gradient Descent**

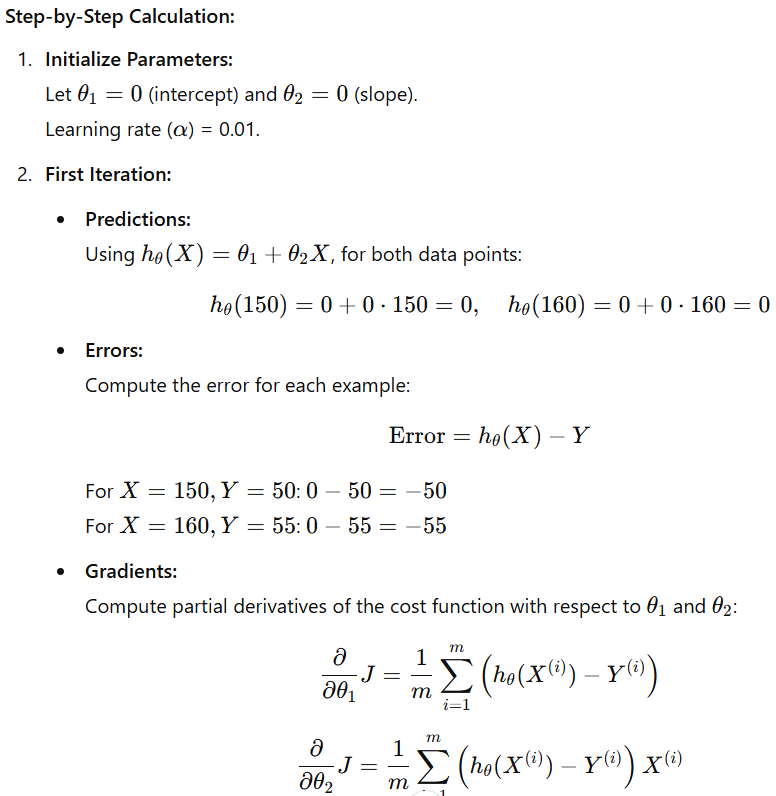


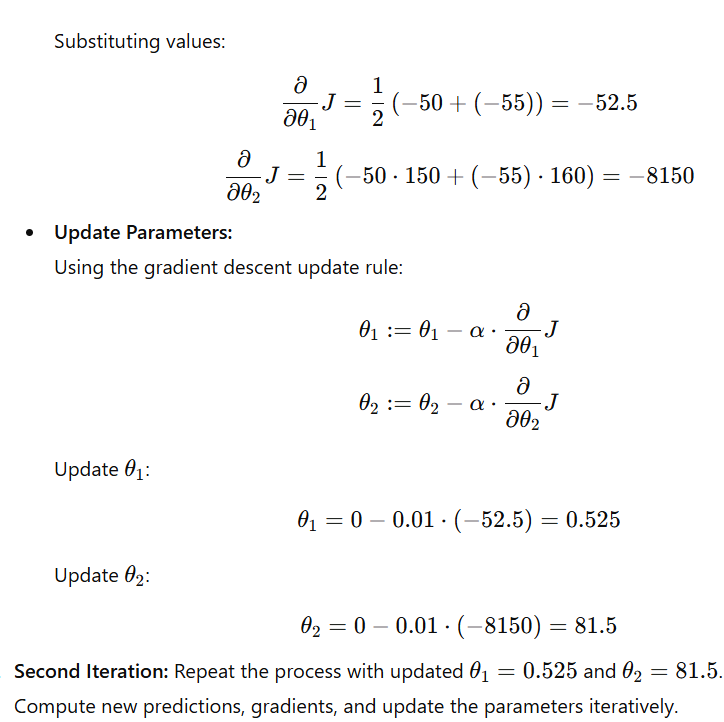


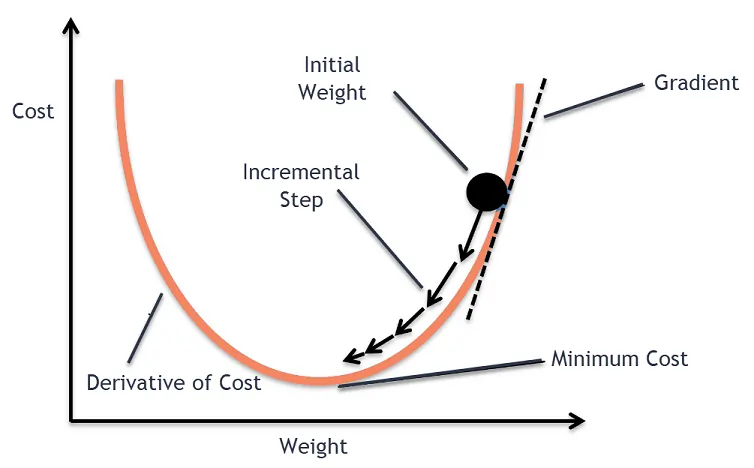
### **Example: Predicting Weight from Height**

#### **Dataset:**

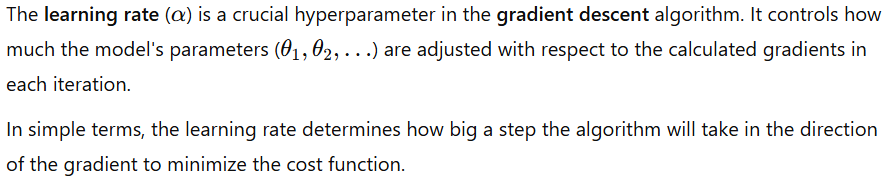








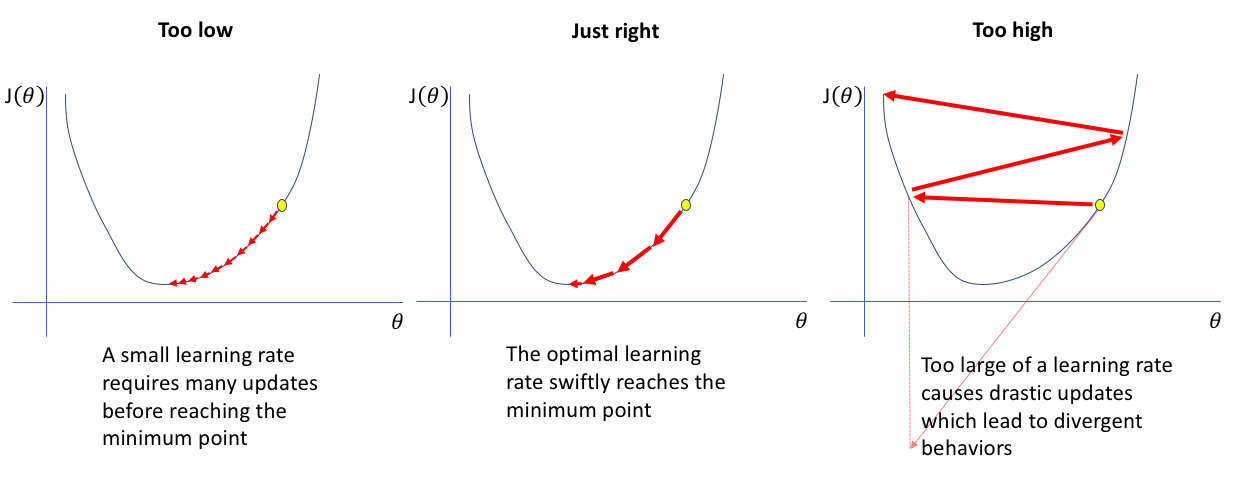
### **Learning Rate in Gradient Descent**



### **Role of the Learning Rate**

The learning rate directly influences the efficiency and success of the gradient descent algorithm:

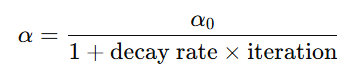
* **Too Large α:**
  + **Overshooting:** If the learning rate is too large, the updates to the parameters may be too big. This causes the gradient descent algorithm to "overshoot" the minimum and may cause it to diverge, instead of converging to the optimal solution.
  + **Oscillations:** The algorithm may keep jumping back and forth across the minimum, never settling down at a good value.
* **Too Small α:**
  + **Slow Convergence:** If the learning rate is too small, the steps will be too small, causing the gradient descent algorithm to converge very slowly. While it might eventually reach the minimum, it may take an impractically long time to do so.
  + **Risk of Stalling:** If the learning rate is too small, it can get stuck in a local minimum and fail to reach the global minimum.
* **Optimal α:**
  + **Balanced Convergence:** An optimal learning rate strikes a balance. It should be large enough to make meaningful progress towards the minimum, but not so large that it overshoots or oscillates. It allows the algorithm to converge to the optimal solution in a reasonable amount of time.



### **Learning Rate Scheduling**

In some cases, instead of using a fixed learning rate, the learning rate is adjusted dynamically during training. This can be done in various ways:

* **Decay:**Gradually decrease the learning rate as the number of iterations increases. This helps the algorithm take large steps early on and smaller steps as it approaches the minimum.



**Adaptive Methods:**

* + Algorithms like **AdaGrad**, **RMSprop**, and **Adam** automatically adjust the learning rate during training, making it more flexible and efficient.

### 

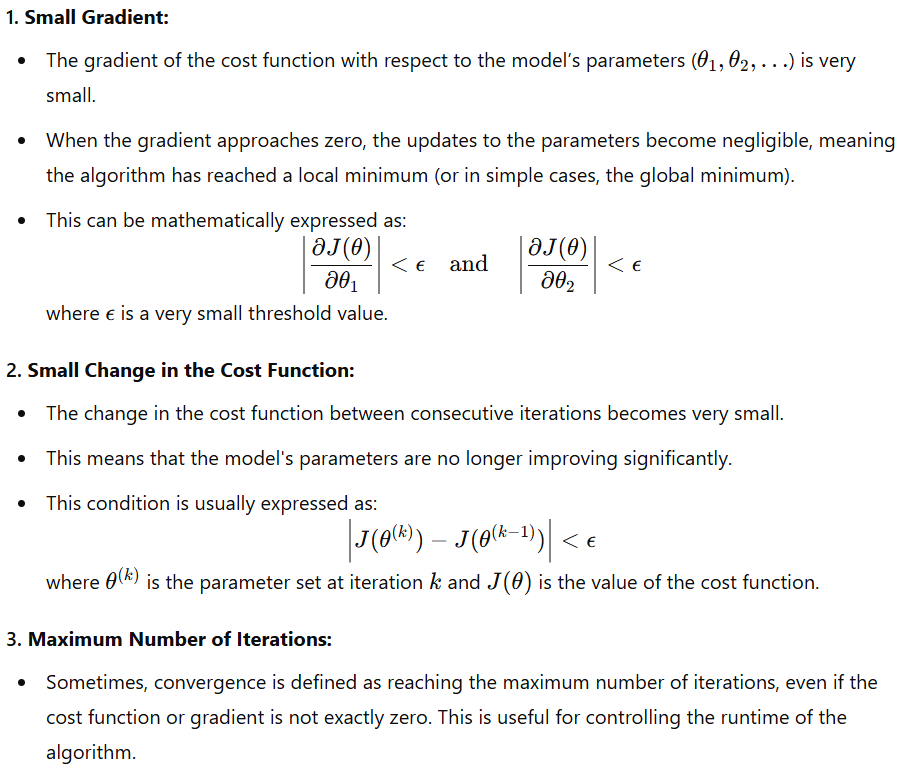
### **Convergence in Optimization Algorithms**

**Convergence** refers to the process in which an optimization algorithm (such as gradient descent) approaches the optimal solution as it iterates. In simpler terms, an algorithm converges when its parameters or variables stop changing significantly, and the solution stabilizes at a point that minimizes the objective (or cost) function.

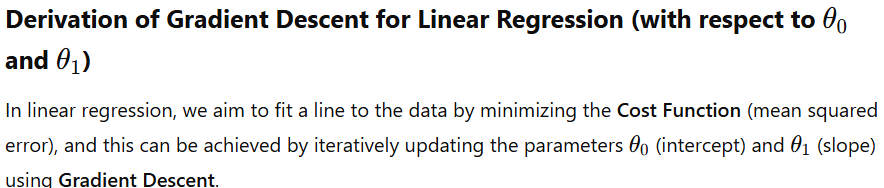
In the context of machine learning, particularly with **gradient descent** for tasks like **linear regression**, convergence is when the algorithm reaches the best-fit model and the changes in the model's parameters become negligible.

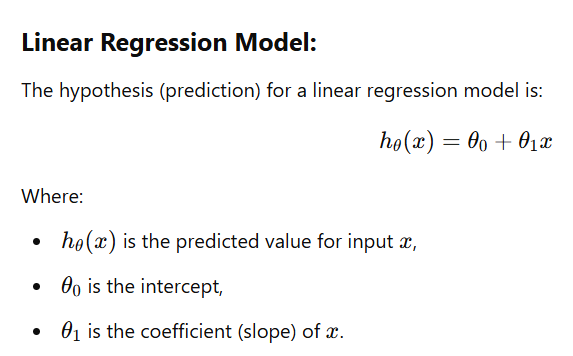
### **Conditions for Convergence in Gradient Descent**

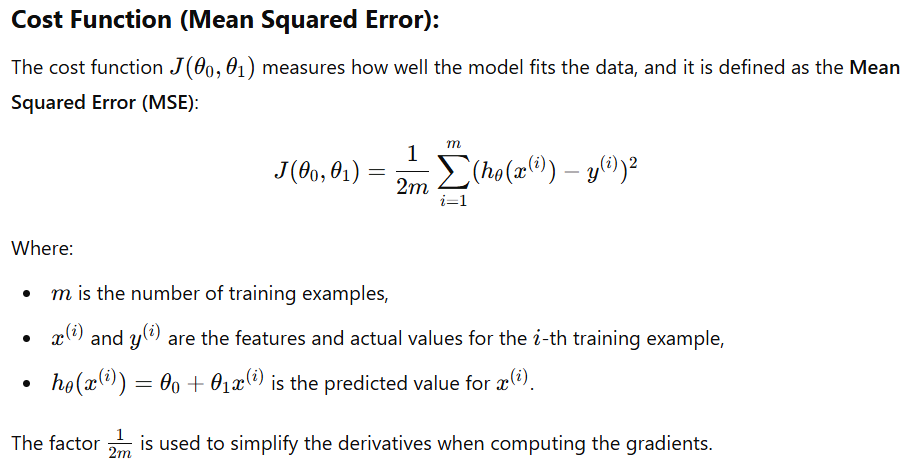
In gradient descent, convergence means that the updates to the parameters are so small that further iterations won’t significantly improve the cost function. There are a few conditions that determine when convergence happens:

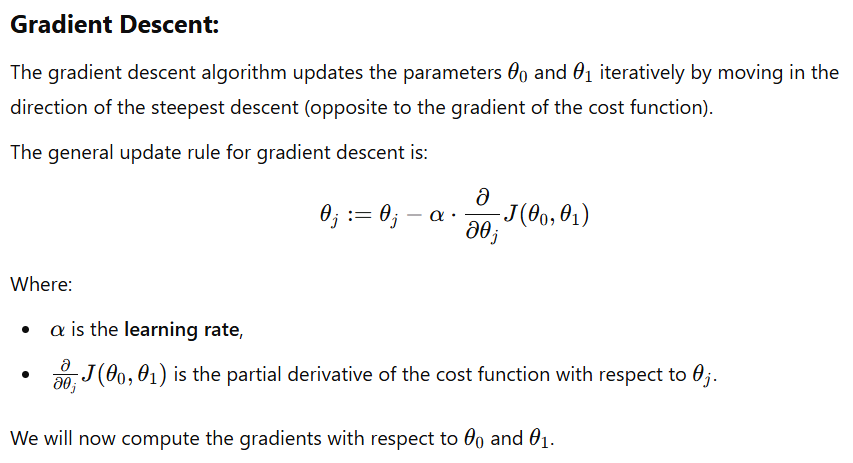


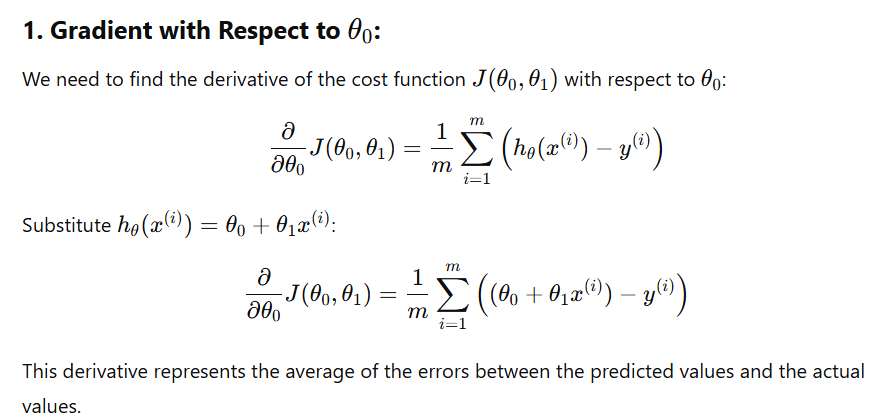
We need to check this for both theta 0 and 1.

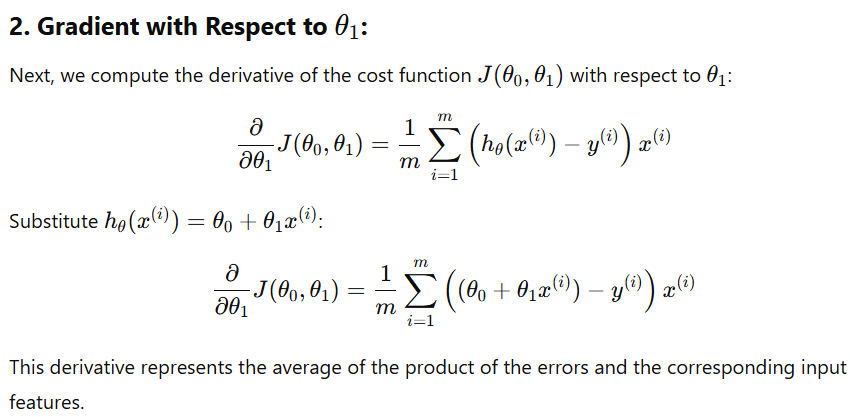


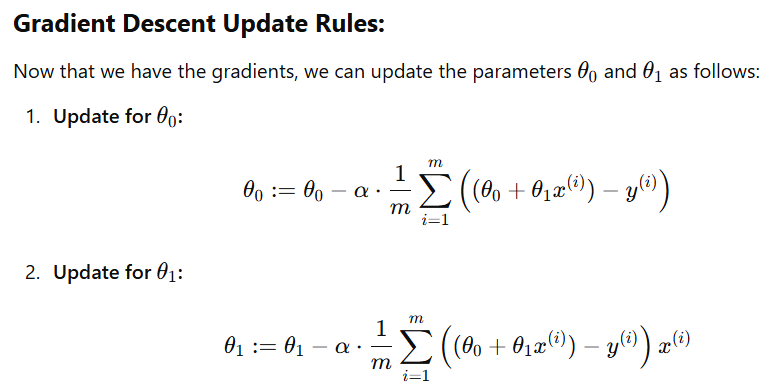


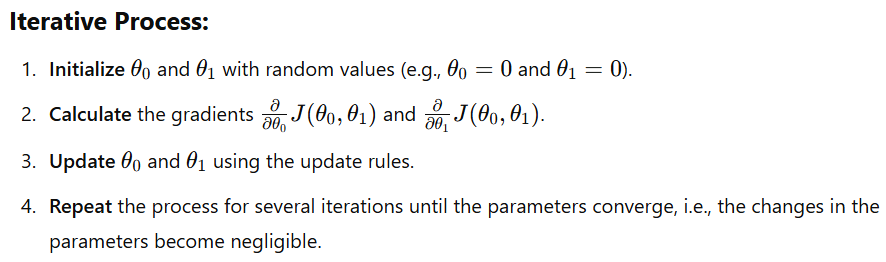












Here are the key **assumptions of Linear Regression** that should be met for the model to produce reliable and accurate results:

### **1. Linearity**

* The relationship between the dependent variable y and the independent variable(s) X should be linear.

### **2. Independence**

* The residuals (errors) should be independent of each other.
* This assumption implies that the error for one data point does not provide information about the error for another data point.
* Independence is especially important when data points are temporally or spatially related. If the data points are correlated (e.g., time-series data), then this assumption is violated.

### **3. Homoscedasticity**

* The variance of the error terms should be constant across all levels of the independent variable(s).
* In other words, the spread (or "scatter") of the residuals should be approximately the same for all values of x.
* If the spread of the residuals changes as a function of x, this is known as **heteroscedasticity** and it violates the assumption of homoscedasticity.

### **4. Normality of Residuals**

* The residuals (errors) should be normally distributed.
* This assumption is particularly important when performing statistical tests (such as hypothesis testing or confidence intervals) on the regression coefficients.
* If the residuals are not normally distributed, the statistical significance of the model may be in question.

### **5. No Multicollinearity (for Multiple Linear Regression)**

* In multiple linear regression, the independent variables should not be highly correlated with each other.
* If two or more independent variables are highly correlated, it leads to multicollinearity, which makes it difficult to isolate the individual effects of the independent variables on the dependent variable.
* This can cause the model's coefficients to be unstable and increase the standard errors of the regression coefficients.

### **6. No Autocorrelation (for Time-Series Data)**

* The residuals (errors) should not exhibit autocorrelation, meaning they should not be correlated with each other over time.
* In time-series data, if the residuals are autocorrelated, it indicates that there may be a missing explanatory variable or that the linear regression model does not capture some underlying time-dependent structure.

### **7. Exogeneity**

* The error term ϵ should not be correlated with the independent variables.
* In other words, the predictors X should be exogenous to the error term ϵ. If the predictors are correlated with the error term, this is known as **endogeneity**, which leads to biased and inconsistent estimates of the model parameters.

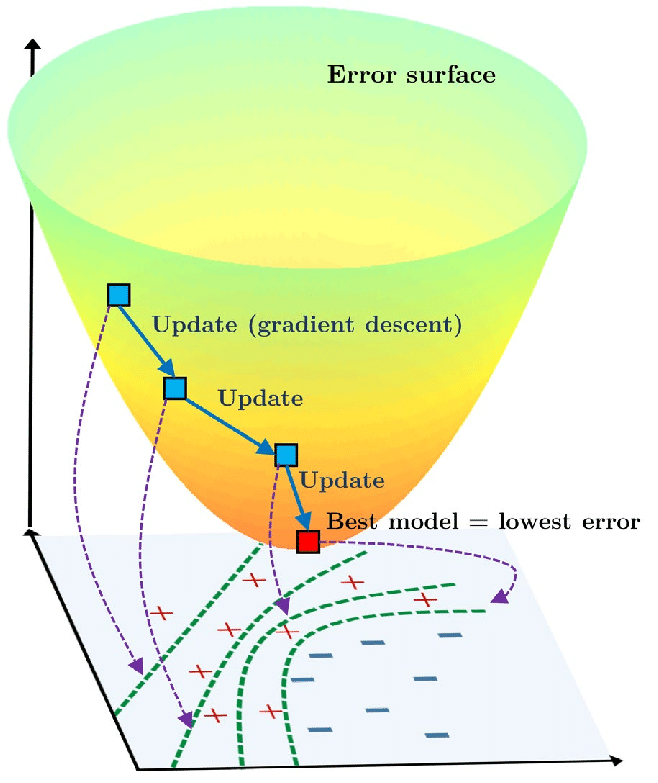
### **Basics of Graph Shapes in Gradient Descent**

#### **1. Univariate (1 Variable) Linear Regression**

* **Cost Function**: Mean Squared Error (MSE)
  + **Graph Shape**: The cost function in terms of θ​ is a **U-shaped parabola**. The goal of gradient descent is to find the minimum point where the cost is lowest.
  + **Process**: Gradient descent adjusts θ​ iteratively, moving toward the minimum of the parabola by calculating the slope (gradient) and adjusting the parameters.

#### **2. Multivariate (2+ Variables) Linear Regression**

* **Cost Function**: Mean Squared Error (MSE)
  + **Graph Shape**: The cost function forms a **bowl-shaped surface** in 3D. Gradient descent updates θ1 and θ2 iteratively to reach the lowest point of the bowl.
  + **Contour Plot**: The contours are **elliptical** or **circular**, representing constant cost values.

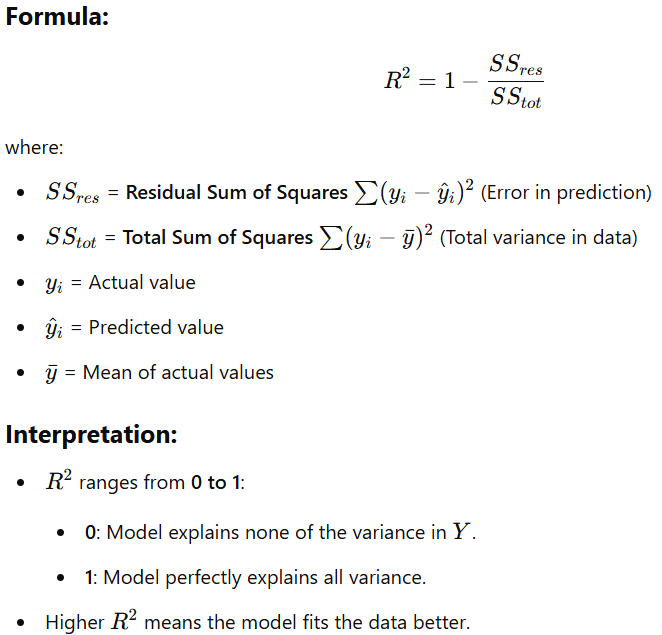


### **Performance Metrics: R² and Adjusted R²**

When evaluating regression models, two common performance metrics are **R² (coefficient of determination)** and **Adjusted R²**. These metrics help assess how well a model explains the variability in the target variable.

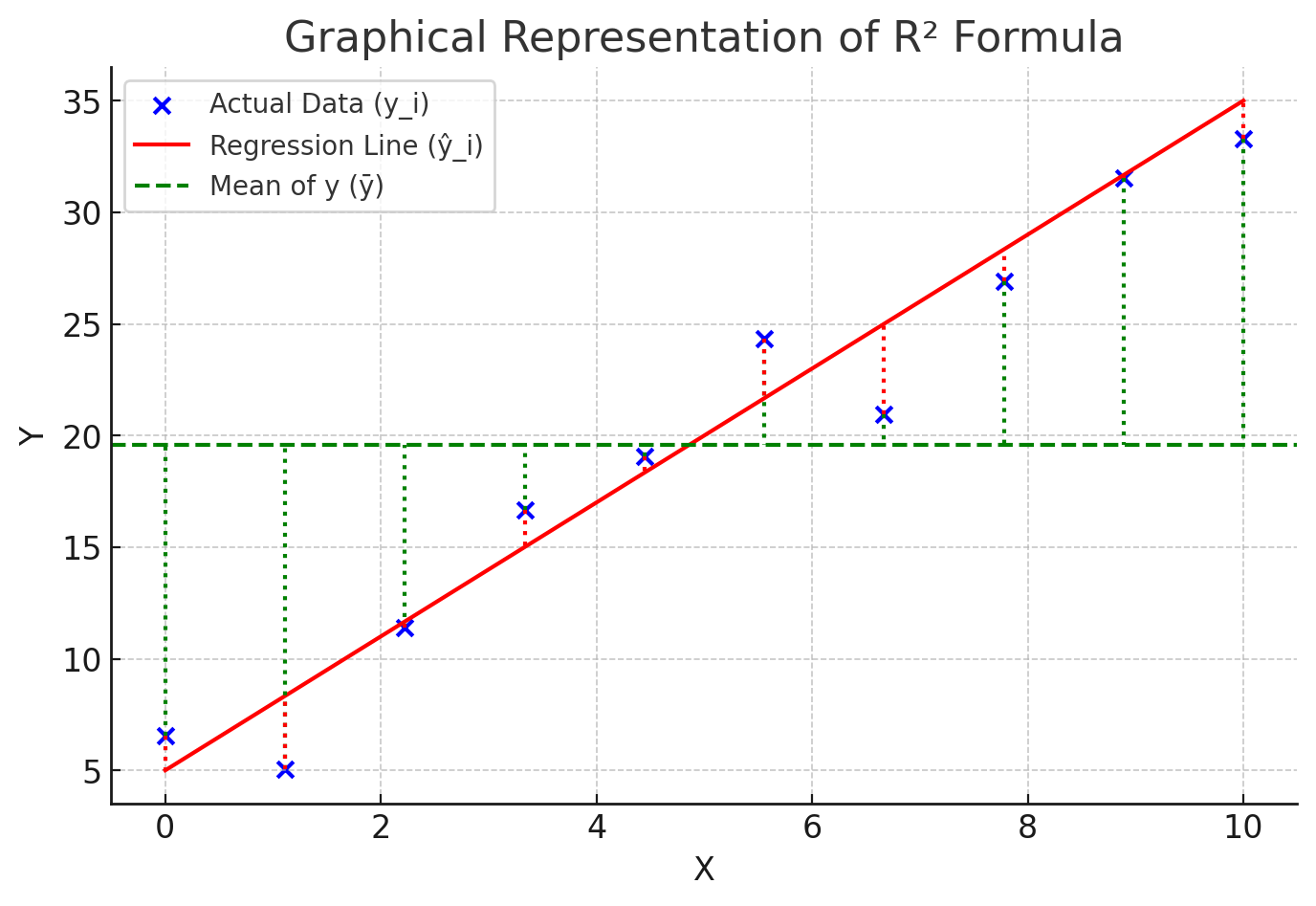
## **1. R² (Coefficient of Determination)**

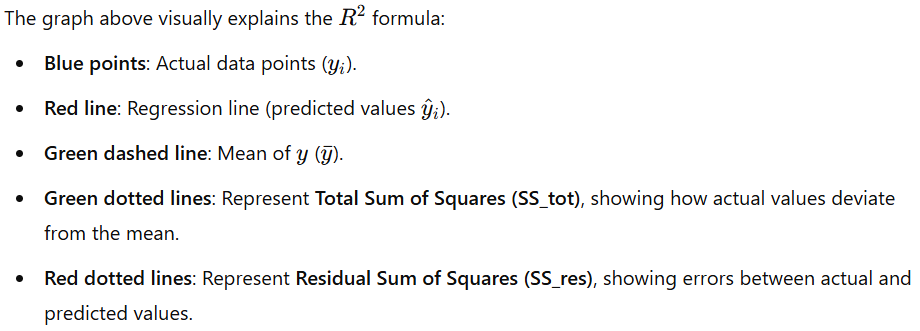
R² measures the proportion of variance in the dependent variable (Y) that is explained by the independent variables (X) in the model.



### **Limitations of R²:**

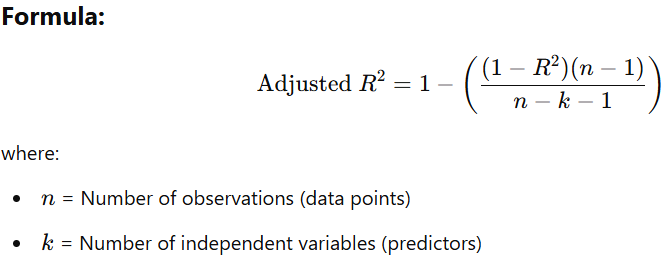
* Adding more independent variables **always increases** R2, even if they are irrelevant.
* It does not penalize unnecessary complexity, leading to overfitting.





## **2. Adjusted R²**

Adjusted R² improves on R² by **penalizing** models that add irrelevant variables. It accounts for the number of predictors and adjusts for model complexity.

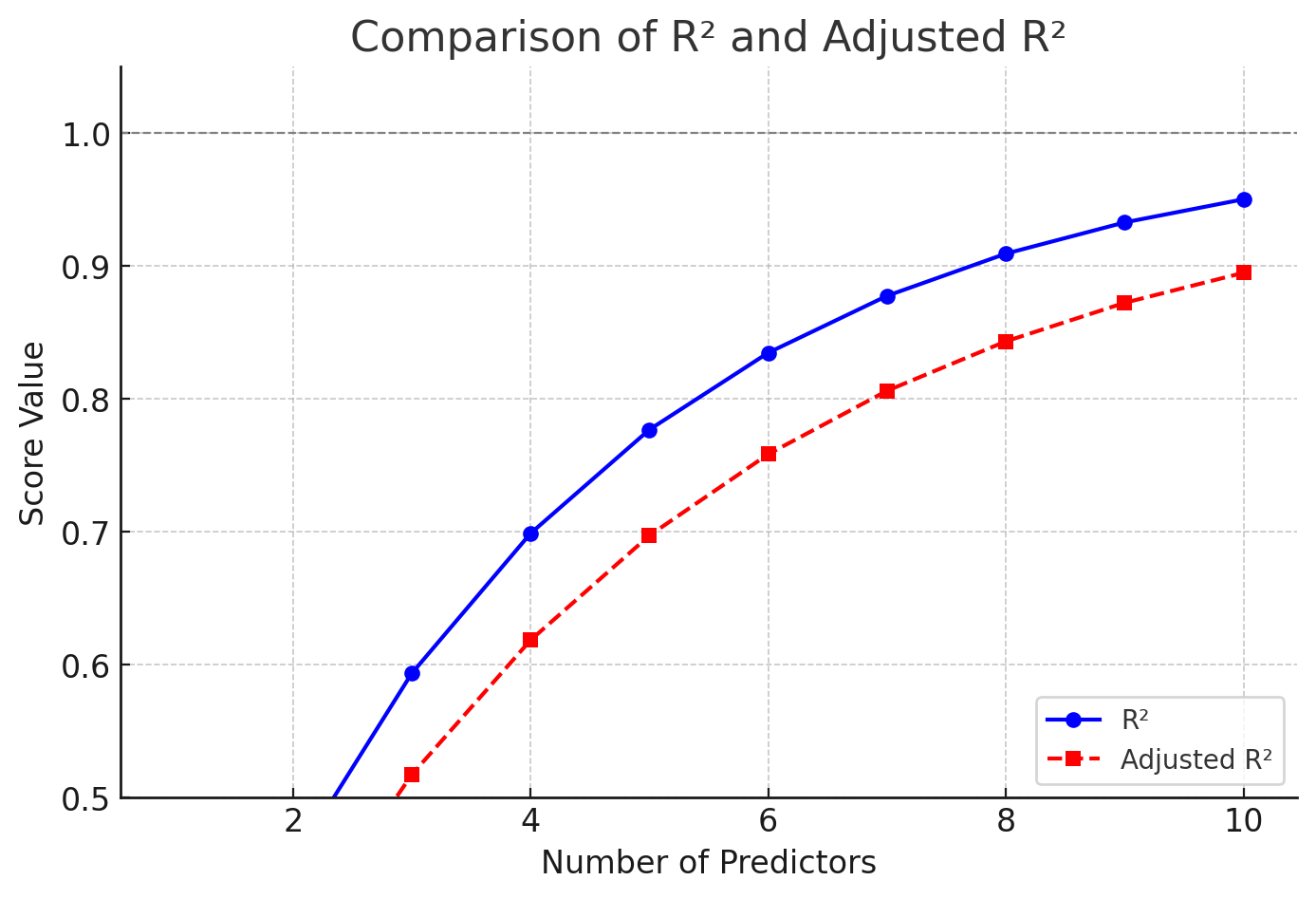


### **Key Differences from R²:**

* If a new variable improves the model, **Adjusted R² increases**.
* If a new variable is irrelevant, **Adjusted R² decreases**.
* Adjusted R² **can be lower than R²**, but it is more reliable for comparing models with different numbers of predictors.

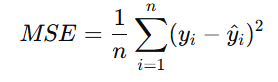
## **Graphical Representation**

Let me generate a visual explanation comparing R2 and Adjusted R2 with an increasing number of features.



### **Error Metrics: MSE, MAE, and RMSE**

## **Mean Squared Error (MSE)**



* **Definition:** MSE is the average of the squared differences between actual and predicted values.
* **Purpose:** It penalizes large errors more than small ones due to squaring.
* **Units:** Squared units of the target variable.

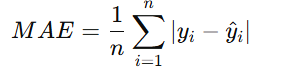
### **✅ Pros**

✔ Differentiable, useful for optimization algorithms (gradient descent).  
✔ Penalizes large errors more, making it useful when large errors need to be avoided.

### **❌ Cons**

✖ Sensitive to **outliers** (as errors are squared, large deviations contribute disproportionately).  
✖ Harder to interpret because the error is squared, making it in different units from the target variable.

## **2. Mean Absolute Error (MAE)**



* **Definition:** MAE is the average of the absolute differences between actual and predicted values.
* **Purpose:** Measures how far predictions are from actual values without squaring.
* **Units:** Same as the target variable.

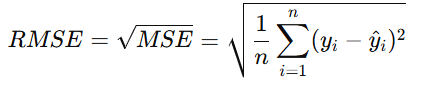
### **✅ Pros**

✔ Less sensitive to outliers than MSE.  
✔ Easier to interpret since the error is in the same unit as the target variable.

### **❌ Cons**

✖ Less useful for gradient-based optimization because the absolute function is not differentiable at zero.  
✖ Treats all errors equally, which may not always be ideal.

## **3. Root Mean Squared Error (RMSE)**



* **Definition:** RMSE is the square root of MSE, making it easier to interpret.
* **Purpose:** Similar to MSE but expressed in the same unit as the target variable.
* **Units:** Same as the target variable.

### **✅ Pros**

✔ Easier to interpret than MSE since it’s in the same unit as the target variable.  
✔ Penalizes large errors while still being useful in optimization.

### **❌ Cons**

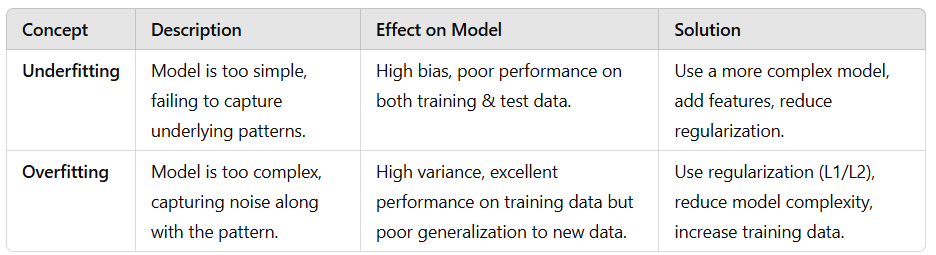
✖ Still sensitive to outliers (like MSE).  
✖ More computationally expensive due to the square root operation.

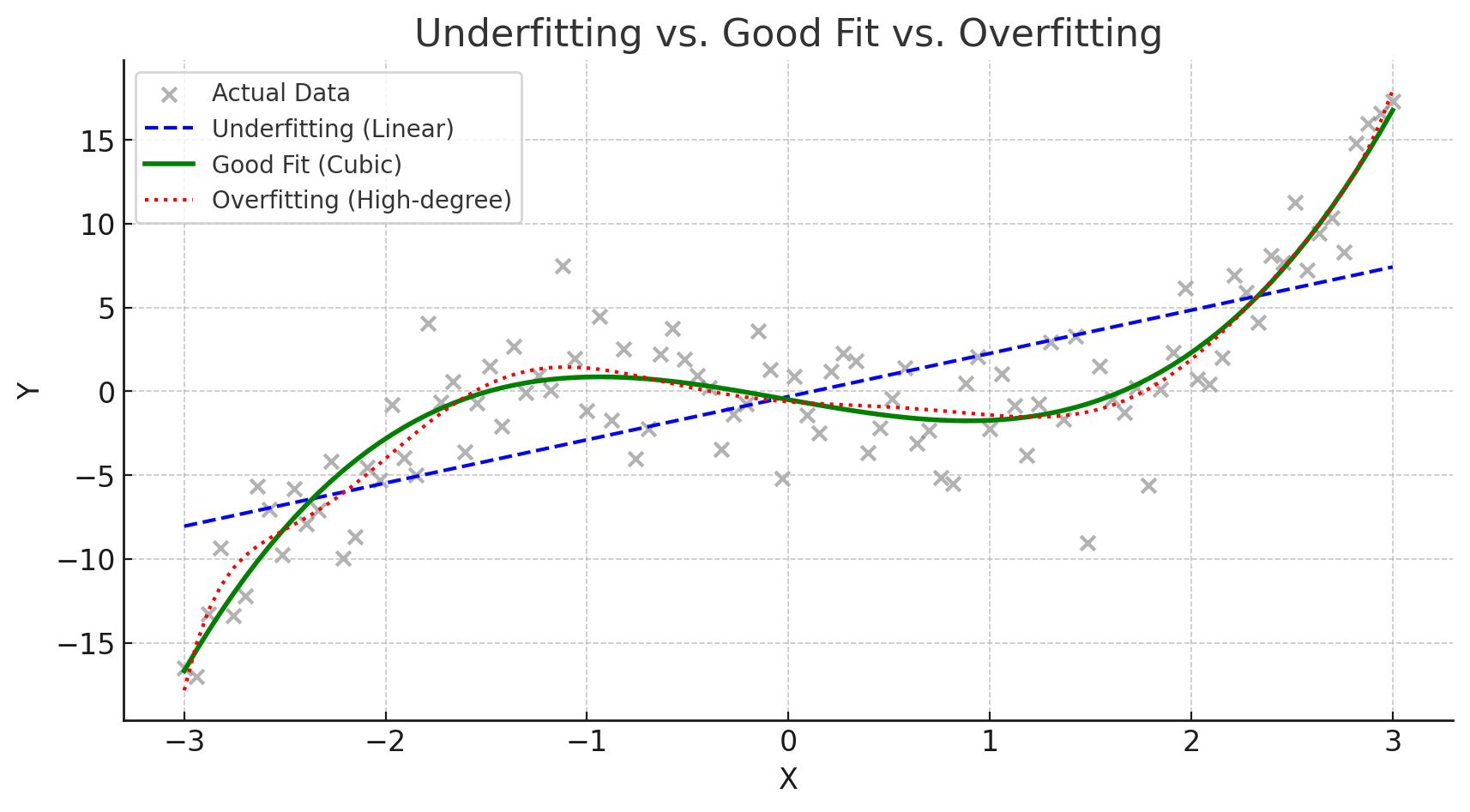
🔹 **MSE** → When large errors should be penalized more.

🔹 **MAE** → When robustness to outliers is needed.

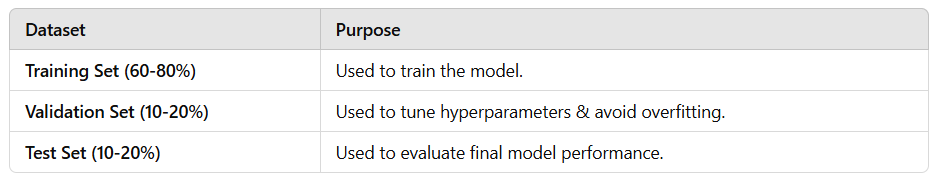
🔹 **RMSE** → When interpretability in the same unit as the target is important.

**Overfitting vs. Underfitting**





**Train-Validation-Test Split**



## **Validation in Machine Learning**

### **Why is Validation Important?**

Validation is used to **fine-tune hyperparameters** and **prevent overfitting**. It helps in selecting the best model before final testing.

## **Types of Validation Methods**

### **1. Hold-Out Validation**

* Splits data into **training (e.g., 80%)** and **validation (e.g., 20%)**.
* Used for **large datasets** when computational efficiency is important.
* **Limitation**: The model's performance depends on how the split is made.

### **2. k-Fold Cross-Validation (CV)**

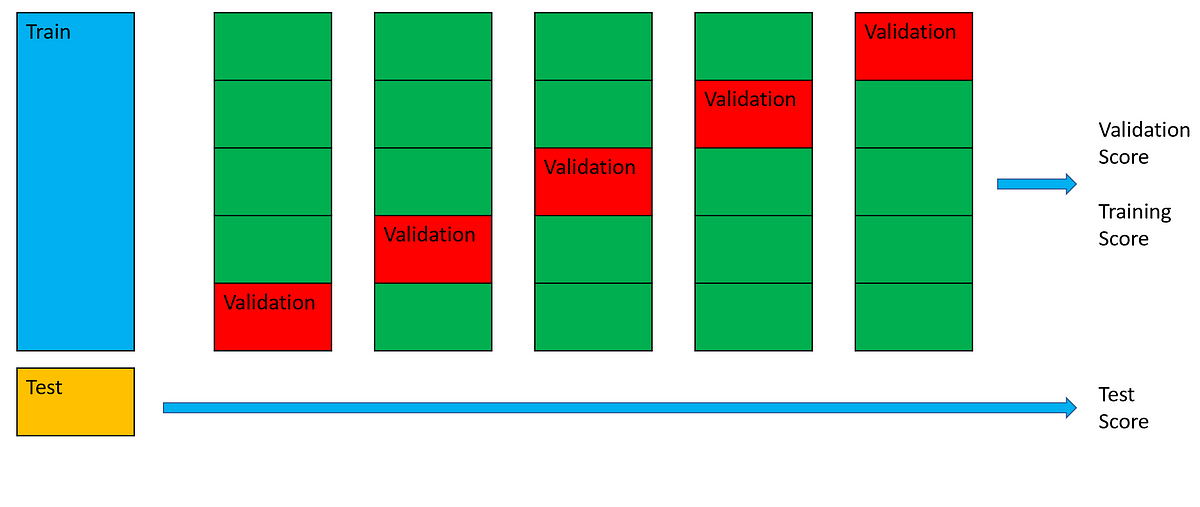
* The dataset is divided into **k equal parts** (e.g., k=5).
* The model trains on **(k-1) folds** and validates on the remaining **1 fold**.
* This process is repeated **k times**, and the final score is the average of all runs.

#### **Advantages**

✔ More reliable than hold-out validation.  
✔ Reduces variance in model performance estimation.

#### **Common k values**

* **k=5 or k=10** is most commonly used.
* **k=2 or k=3** for small datasets to save computation time.



### **3. Stratified k-Fold Cross-Validation**

* Used in **classification problems** where class distribution is imbalanced.
* Ensures each fold has a similar proportion of classes as the original dataset.

### **4. Leave-One-Out Cross-Validation (LOOCV)**

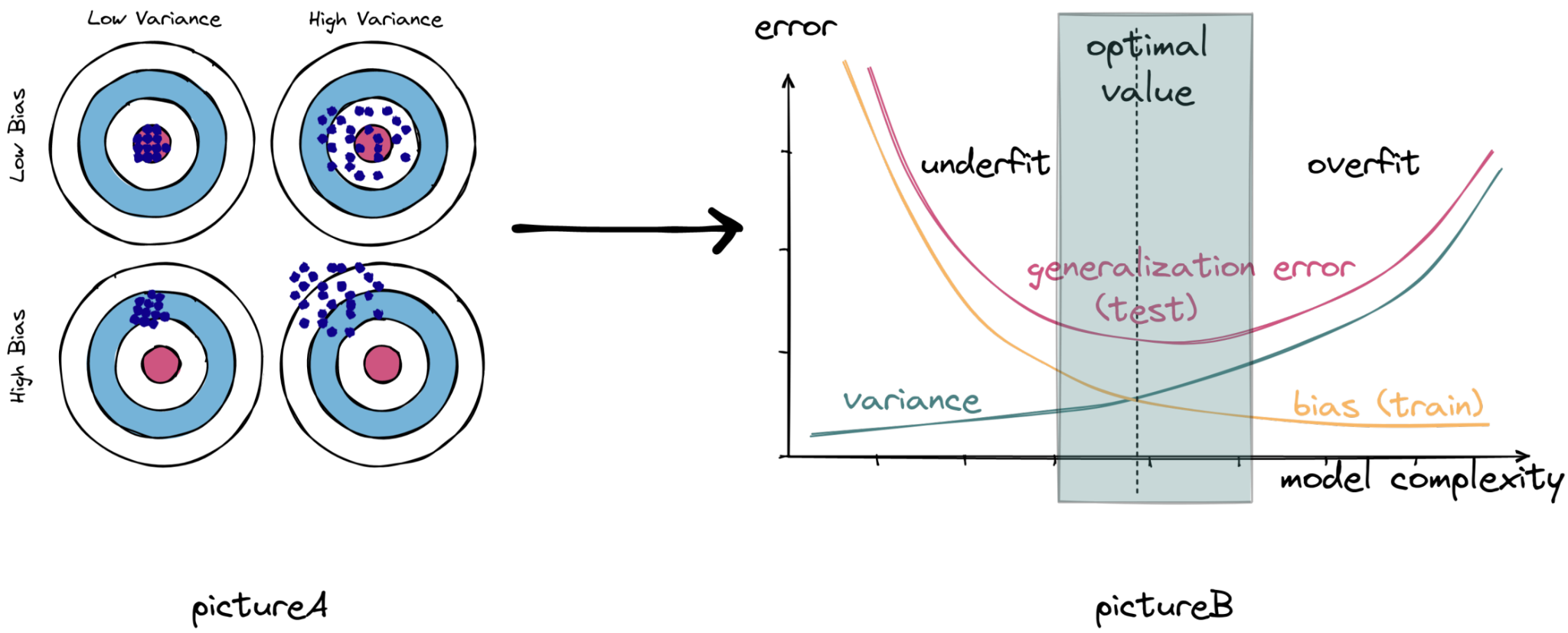
* Each sample is treated as a separate validation set, while the rest are used for training.
* Suitable for **very small datasets**, but computationally expensive.

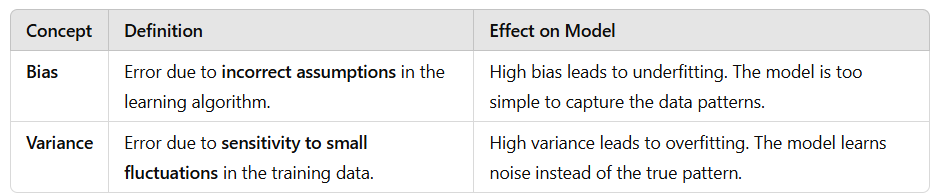
## **When to Use Which Validation Technique?**

|  |  |
| --- | --- |
| **Method** | **Use When** |
| **Hold-Out Validation** | Large datasets with clear separation between training and validation. |
| **k-Fold Cross-Validation** | Moderate-sized datasets where variance in model performance needs to be minimized. |
| **Stratified k-Fold CV** | Classification tasks with imbalanced data. |
| **LOOCV** | Very small datasets where every data point matters. |

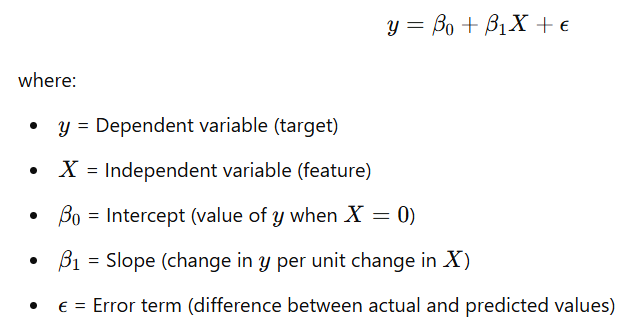
## **How Validation is Used in Model Training?**

1. **Split the dataset** into training and validation sets.
2. **Train the model** using the training set.
3. **Evaluate on the validation set** to tune hyperparameters (e.g., learning rate, number of layers).
4. **Repeat steps 2-3** until the best model is found.
5. **Test the final model** on the test set to check generalization.

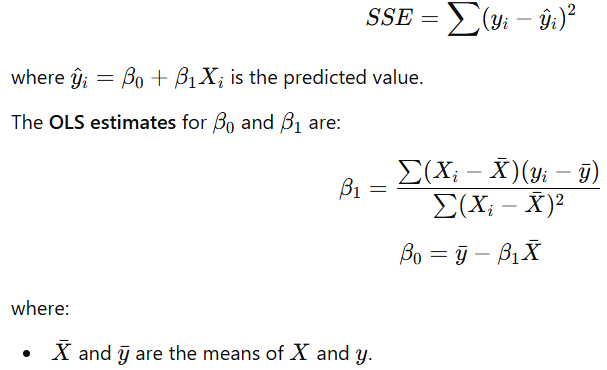


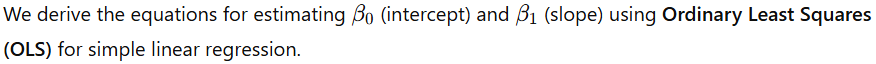


## **Linear Regression using Ordinary Least Squares (OLS)**



The **OLS method** estimates the parameters (β0 and β1​) by minimizing the **sum of squared errors (SSE)**:





### **Deriving the Normal Equations**

