Topics - Supervised Learning

Classification Techniques:

- Naive Bayes Classification
- Fitting Multivariate Bernoulli Distribution
- Gaussian Distribution and Multinomial Distribution
- K- Nearest Neighbours
- Decision tree
- Random Forest
- Ensemble Learning
- Support Vector Machines
- Evaluation metrics for Classification Techniques: Confusion Matrix, Accuracy, Precision, Recall, F1 Score, Threshold, AUC-ROC

Regression Techniques:

- Basic concepts and applications of Regression
- Simple Linear Regression - Gradient Descent and Normal Equation Method
- Multiple Linear Regression
- Non-Linear Regression
- Linear Regression with Regularization
- Overfitting and Underfitting
- Hyperparameter tuning
- Evaluation Measures for Regression Techniques: MSE, RMSE, MAE, R2

What is Regression?

- It is a statistical method that is used for predictive analysis.
- A regression problem is when the output variable is a real or continuous value, such as "salary" or "weight".
- **Regression Analysis** is a statistical process for estimating the relationships between the dependent variables or criterion variables and one or more independent variables or predictors.
- Regression analysis is generally used when we deal with a dataset that has the target variable in the form of **continuous data.**

Applications of Regression

Predictive Modeling and Forecasting

- Sales forecasting
- Demand forecasting
- Stock price prediction
- Weather forecasting

Financial Analysis:

- Credit scoring and risk assessment
- Financial market analysis
- Fraud detection

Energy and Utilities:

- Energy consumption prediction
- Load forecasting
- Energy price modeling

Sports Analytics:

- Player performance analysis
- Outcome prediction

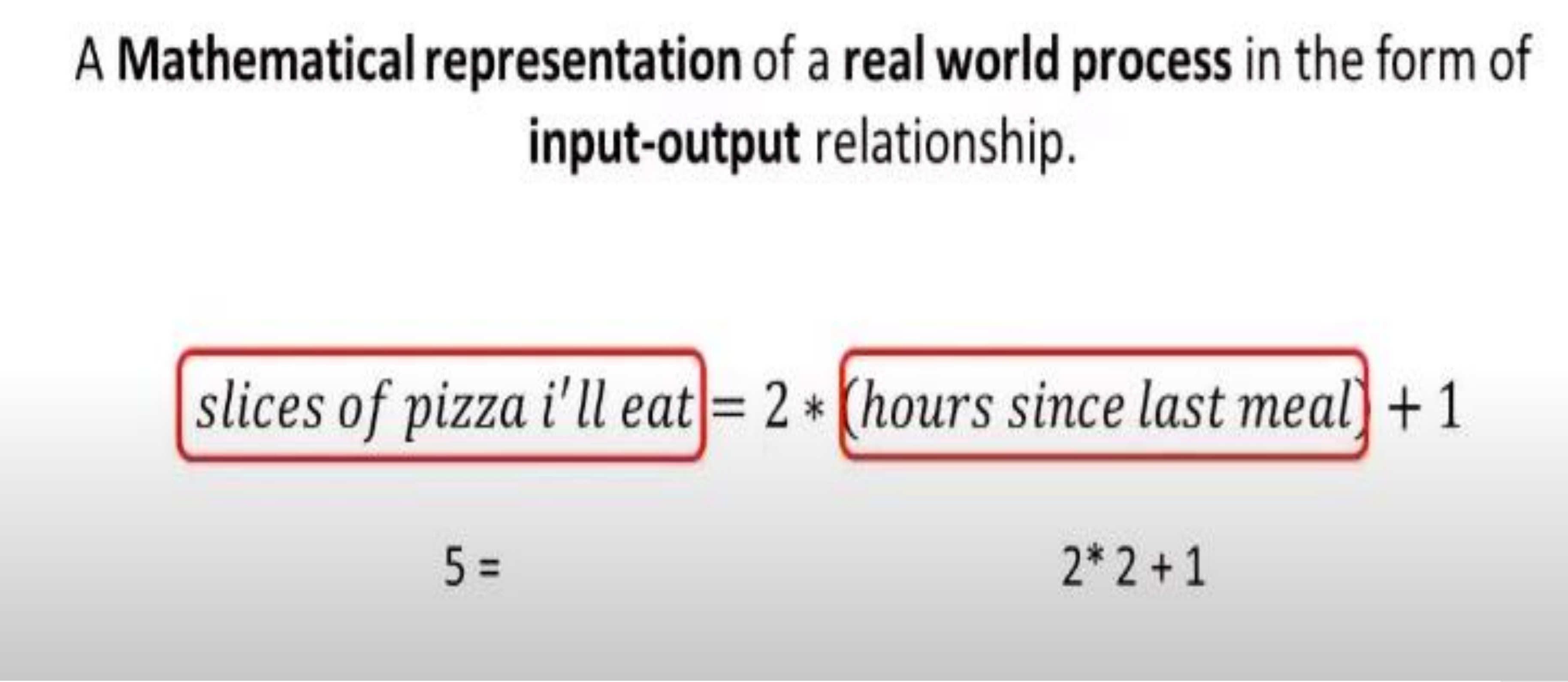
Environmental Analysis:

- Climate Prediction
- Pollution prediction
- Environmental impact assessment

Time Series Analysis:

- Trend analysis
- Seasonal pattern identification

Contd.

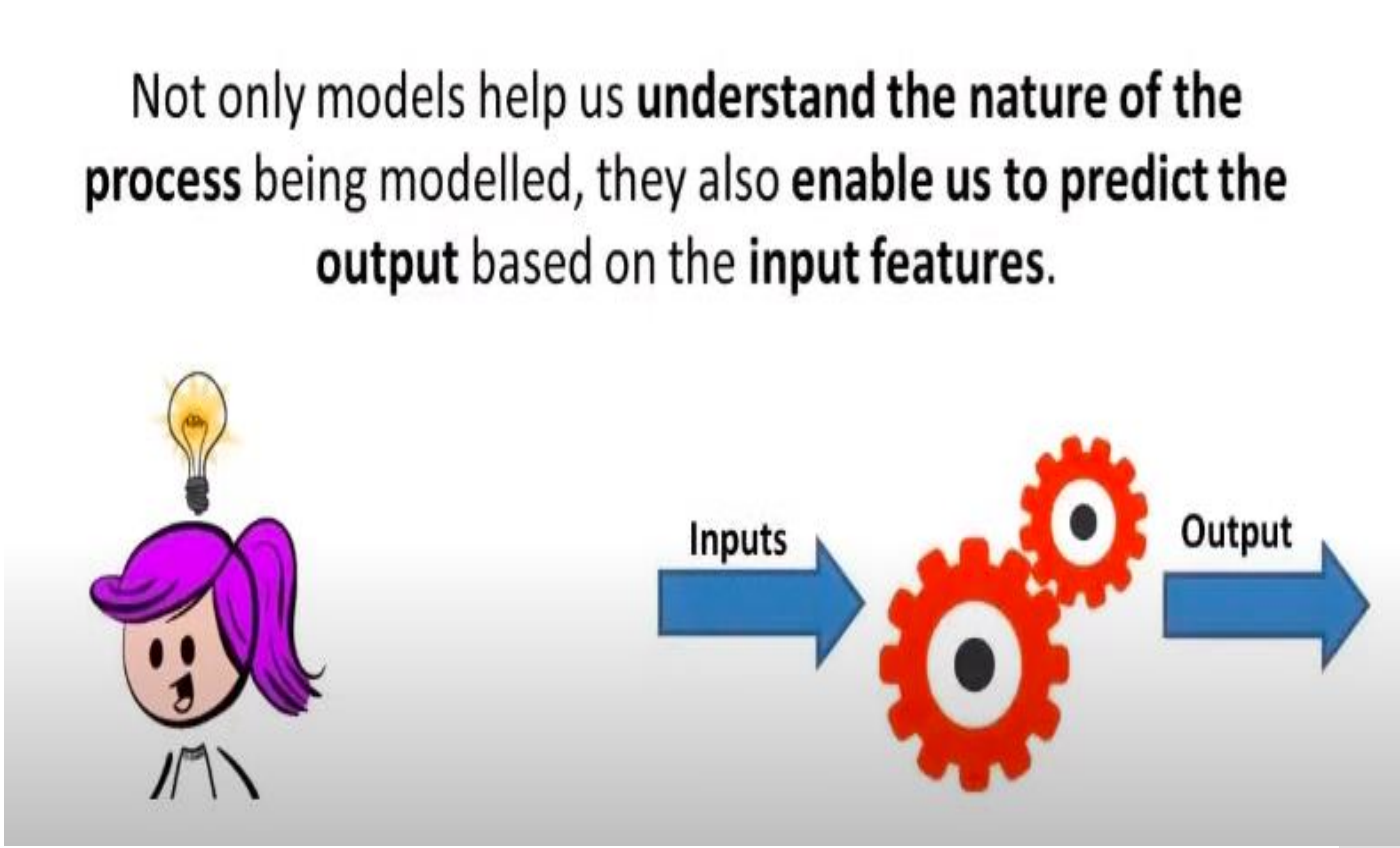


A **Mathematical representation** of a **real world process** in the form of **input-output** relationship.

$$\boxed{\textit{slices of pizza i'll eat}} = 2 * \boxed{(\textit{hours since last meal})} + 1$$

5 = 2* 2 + 1

Contd.



Not only models help us **understand the nature of the process** being modelled, they also **enable us to predict the output** based on the **input features**.

Contd..

- **Linear regression makes predictions for continuous / real or numeric variables such as sales, salary, age, product price, etc.**
- Predictive modelling is a kind of modelling here the possible **output(Y)** for the given **input(X)** is predicted based on the previous data or values.

Example:

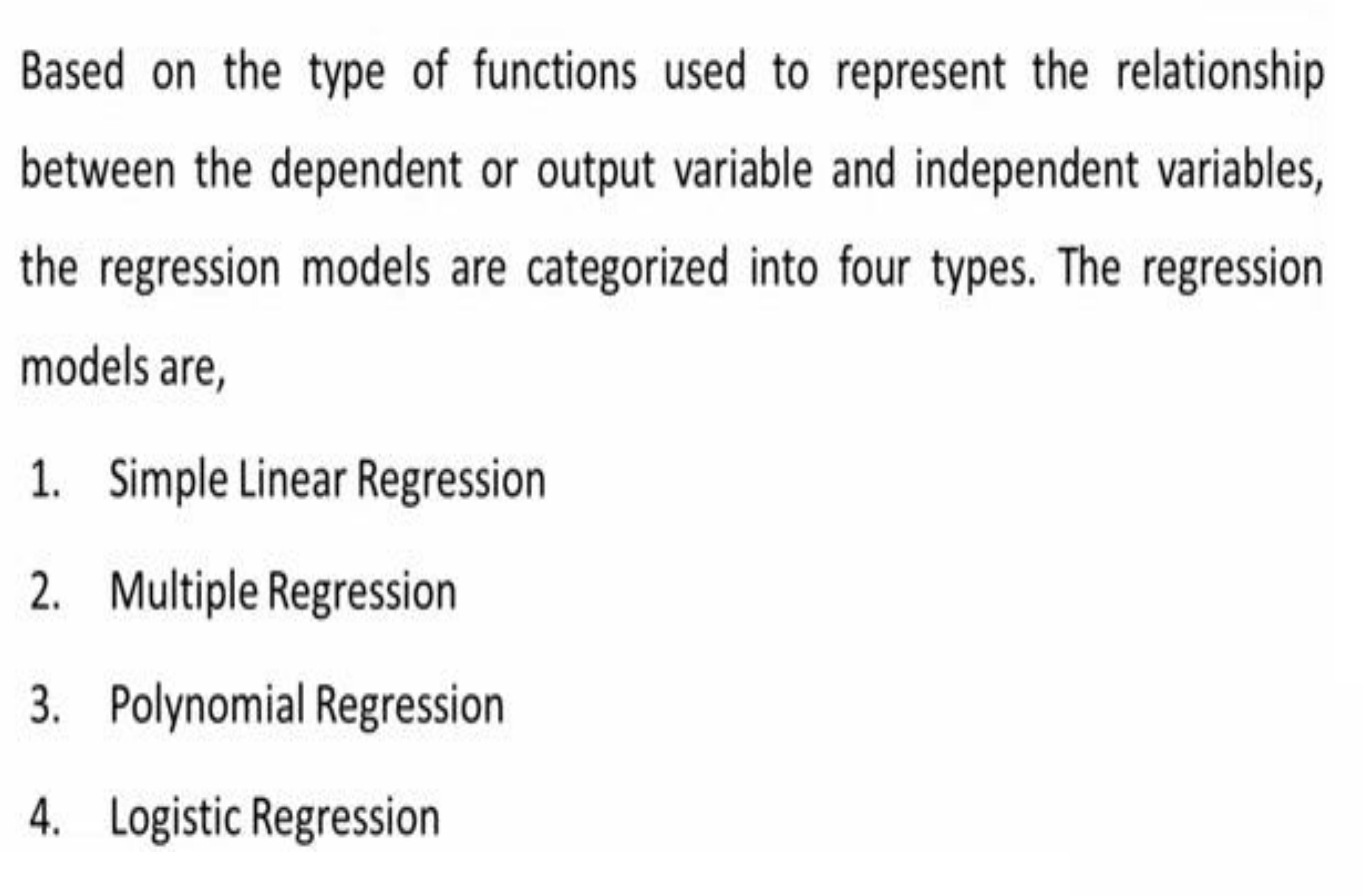
Example: Predicating the height of a person from the Age of the Person which is given

- The age of the Person is Independent variable
- Height is a Dependent Variable

Example: Predicting the Price of the car given the car model, year of Manufacturing, engine capacity etc

- car model, year of Manufacturing, and engine capacity are Independent Variable
- The price of Car is dependent variable

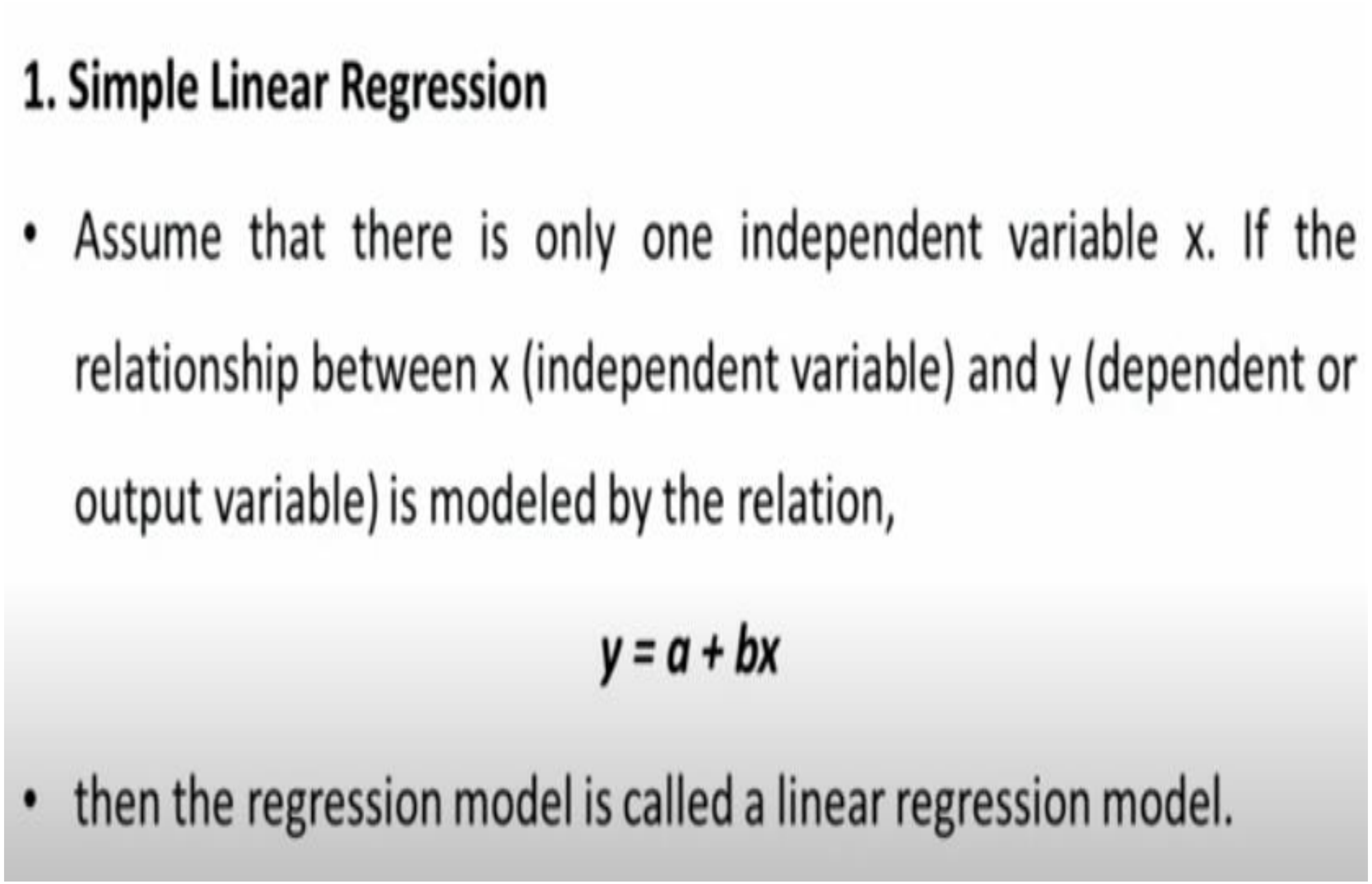
Type of Regression Models



Based on the type of functions used to represent the relationship between the dependent or output variable and independent variables, the regression models are categorized into four types. The regression models are,

1. Simple Linear Regression
2. Multiple Regression
3. Polynomial Regression
4. Logistic Regression

Simple Linear Regression



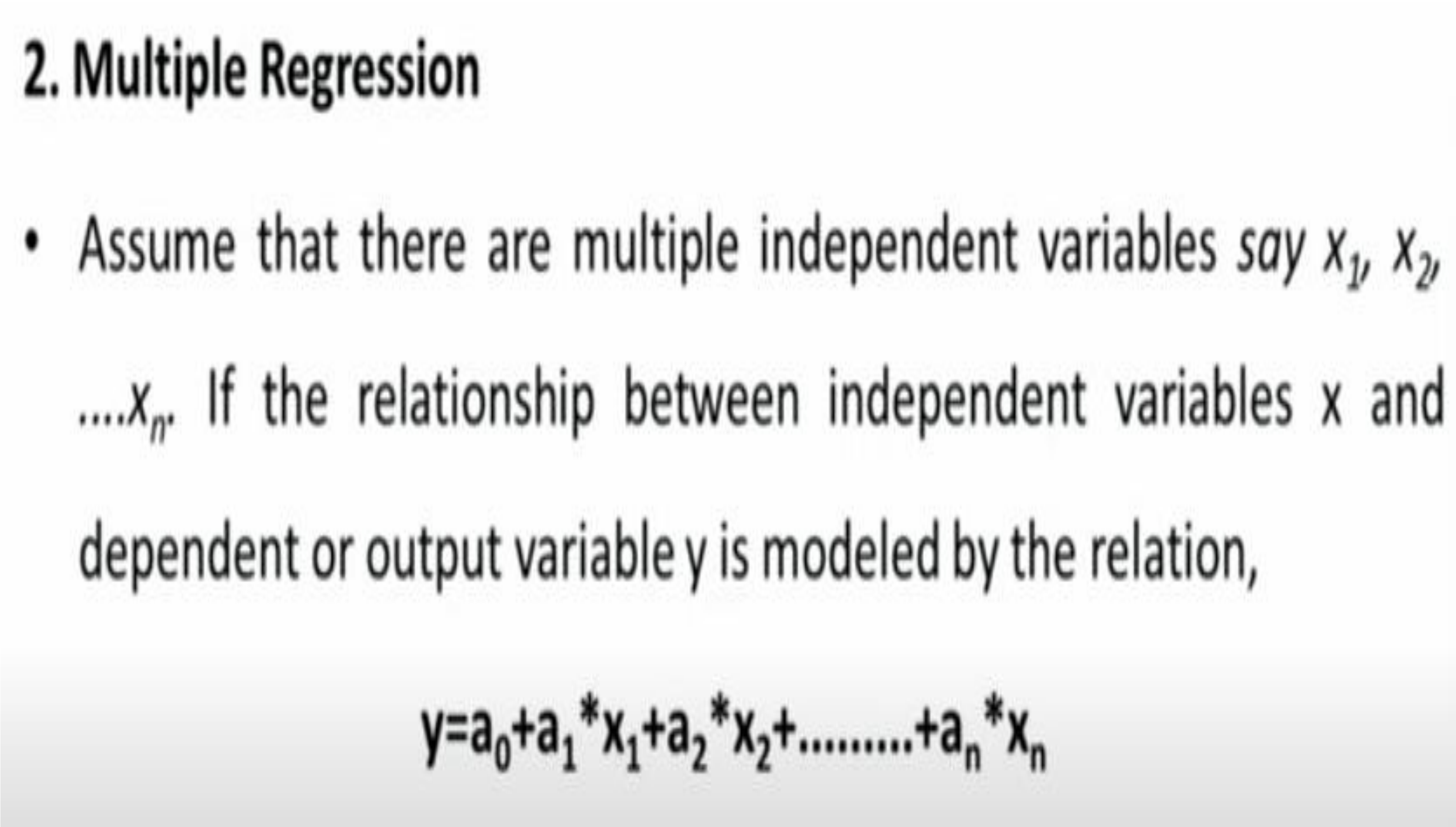
1. Simple Linear Regression

- Assume that there is only one independent variable x. If the relationship between x (independent variable) and y (dependent or output variable) is modeled by the relation,

$$y = a + bx$$

- then the regression model is called a linear regression model.

Multiple Regression

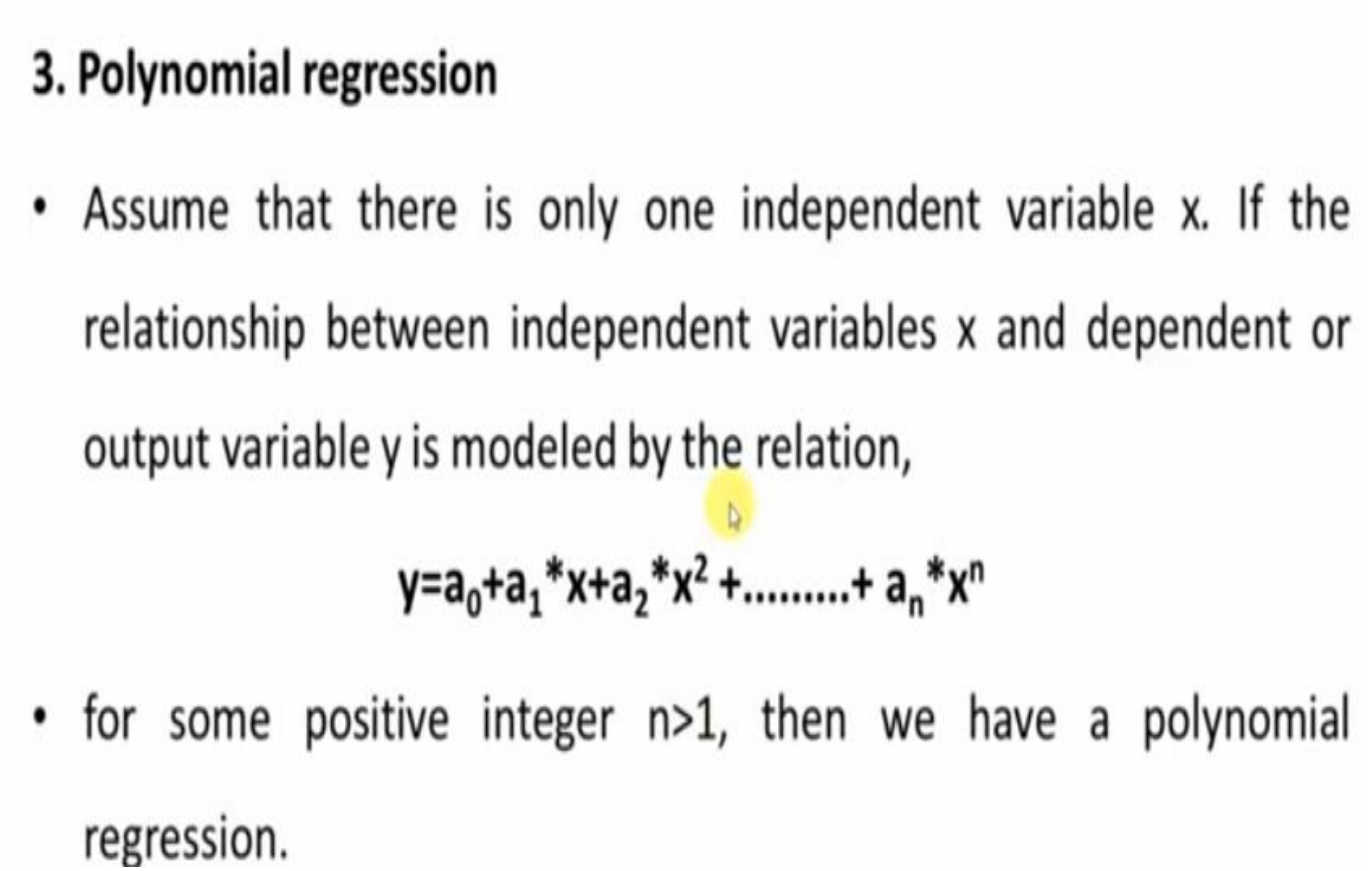


2. Multiple Regression

- Assume that there are multiple independent variables *say* x_1, x_2, ...x_n. If the relationship between independent variables x and dependent or output variable y is modeled by the relation,

$$y=a_0+a_1{*}x_1+a_2{*}x_2+.........+a_n{*}x_n$$

Polynomial Regression



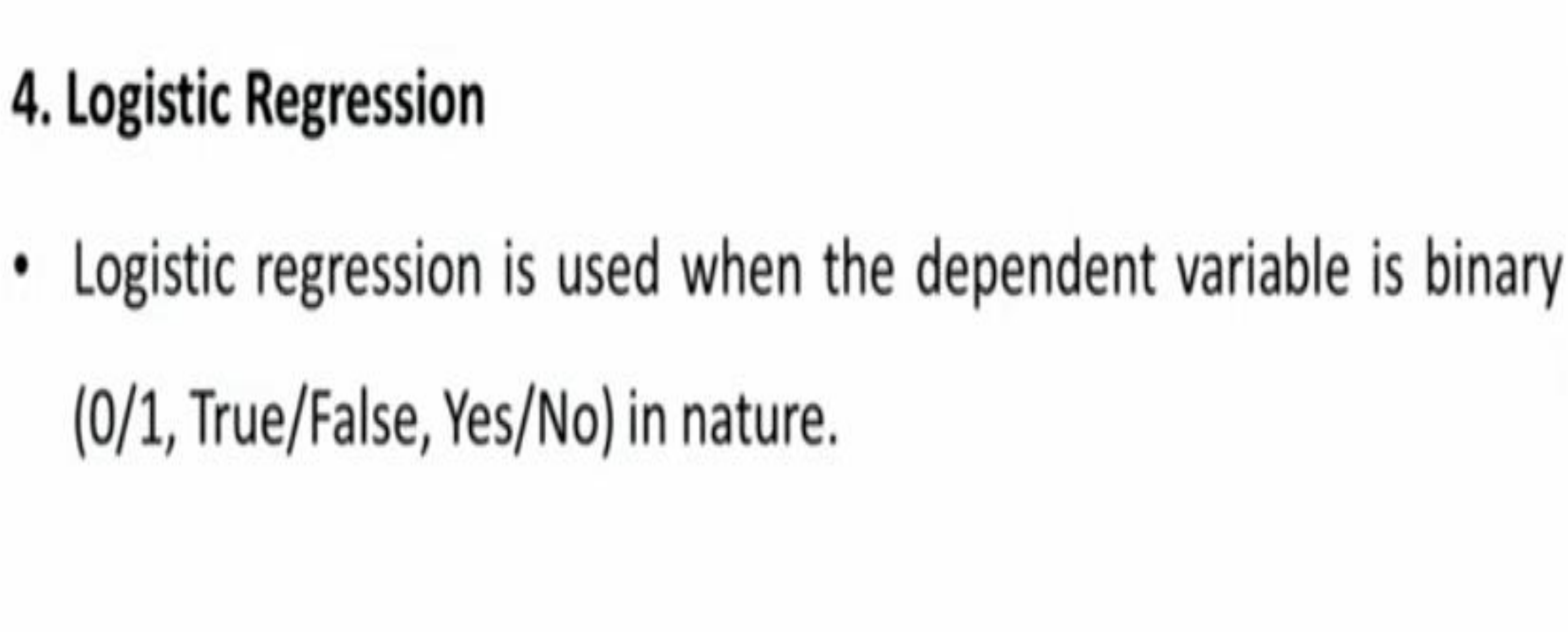
3. Polynomial regression

- Assume that there is only one independent variable x. If the relationship between independent variables x and dependent or output variable y is modeled by the relation,

$$y=a_0+a_1*x+a_2*x^2+........+a_n*x^n$$

- for some positive integer n>1, then we have a polynomial regression.

Logistic Regression



4. Logistic Regression

- Logistic regression is used when the dependent variable is binary (0/1, True/False, Yes/No) in nature.

Linear Regression with one Variable – Simple Linear Regression

Linear Regression with One Variable or Simple Linear Regression

- Linear Regression is an approach to show the relationship between the independent variable or Input Variable x and dependent variable or Output variable y.

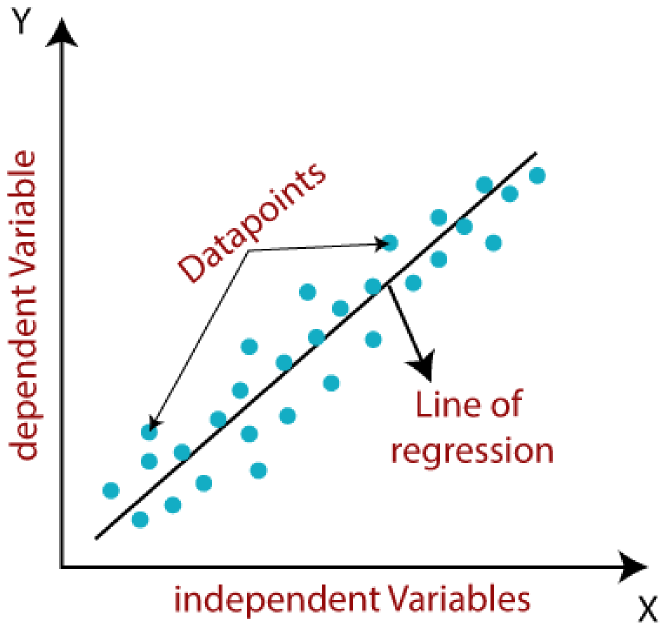
- **y = mx + c**

- **m: slope of line**

- **c: constant**

- **x: Independent variable**

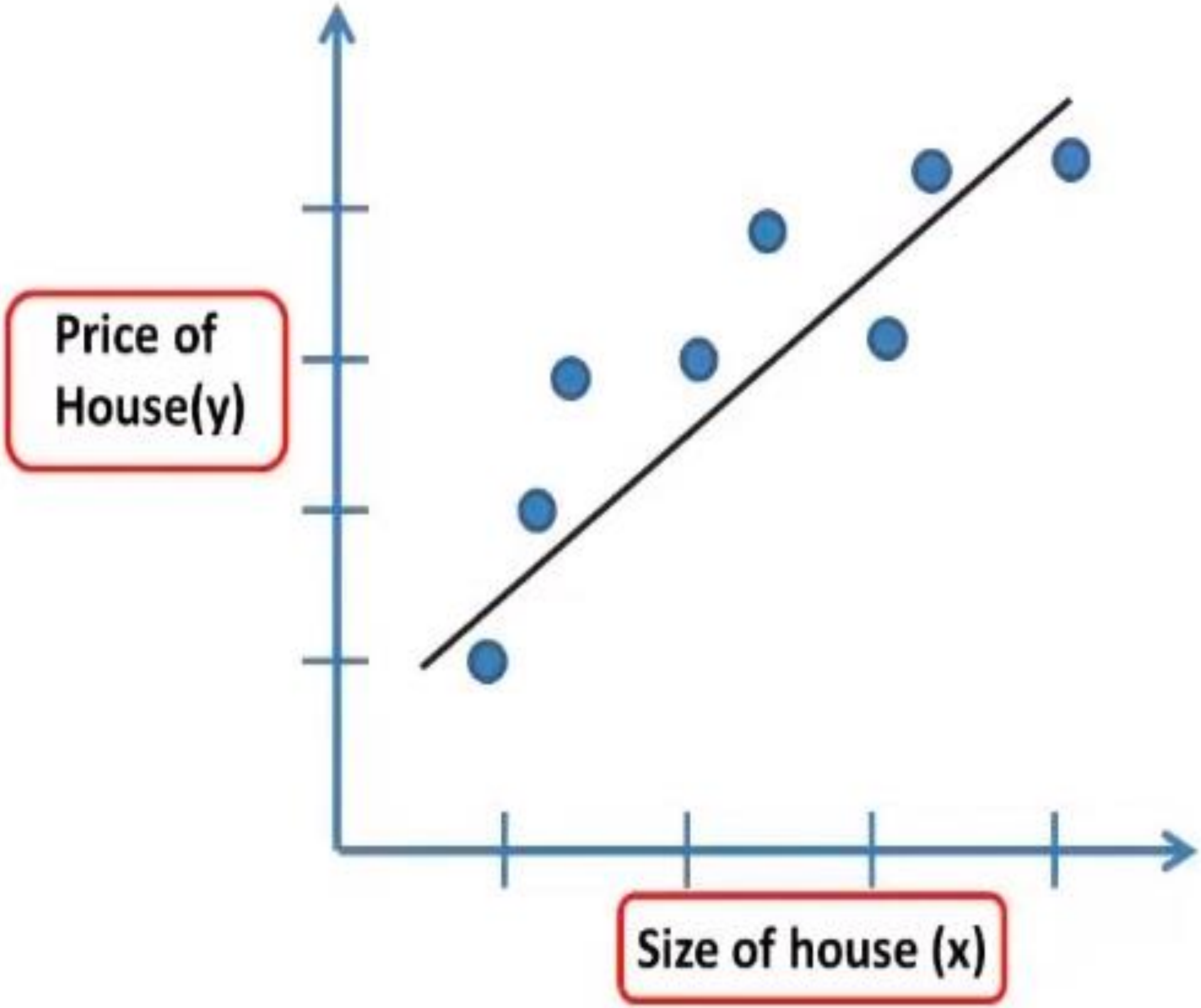
- **y: Dependent variable**



- Our goal is to find the fit of the line. The best fit means where the error is minimum. It can make our prediction more accurate.

Size of house(sqm.)		Price of house
64		$ 200,000
80		$ 250,000
63		$ 230,000
100		$ 320,000
128		$ 300,000
144		$ 450,000
...		....
...		....
81		??

..we want to know price(**output**) of a new house based on its Size(**input**).



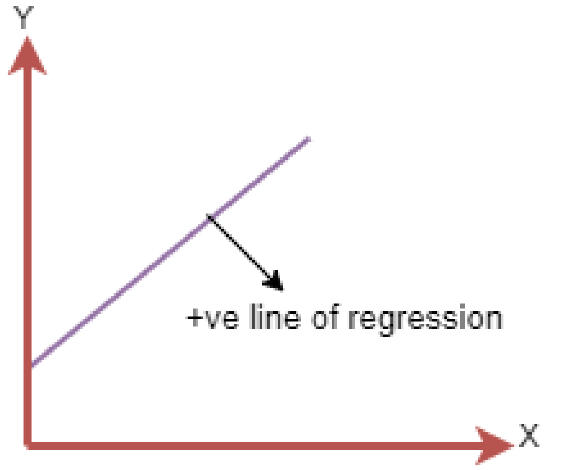
any **linear relationship** between 2 variables can be represented as a **straight line**.

$$\boldsymbol{y = a_0 + a_1 x}$$

*"y" is **target** -> Price of house*
*"x" is **feature** -> Size of house*
a_0 and a_1 are **model parameters**

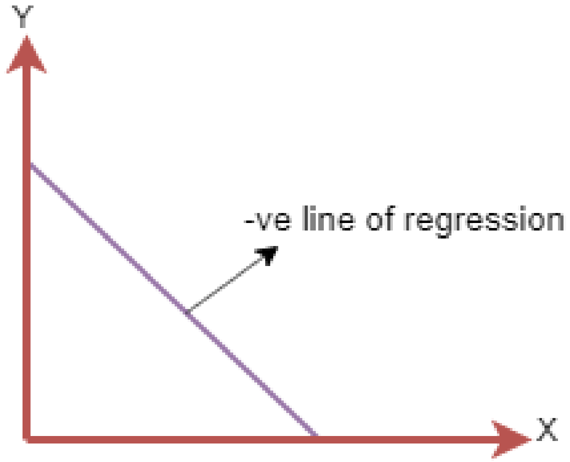
Contd.

- **Linear Regression Line**
- Positive Linear Relationship



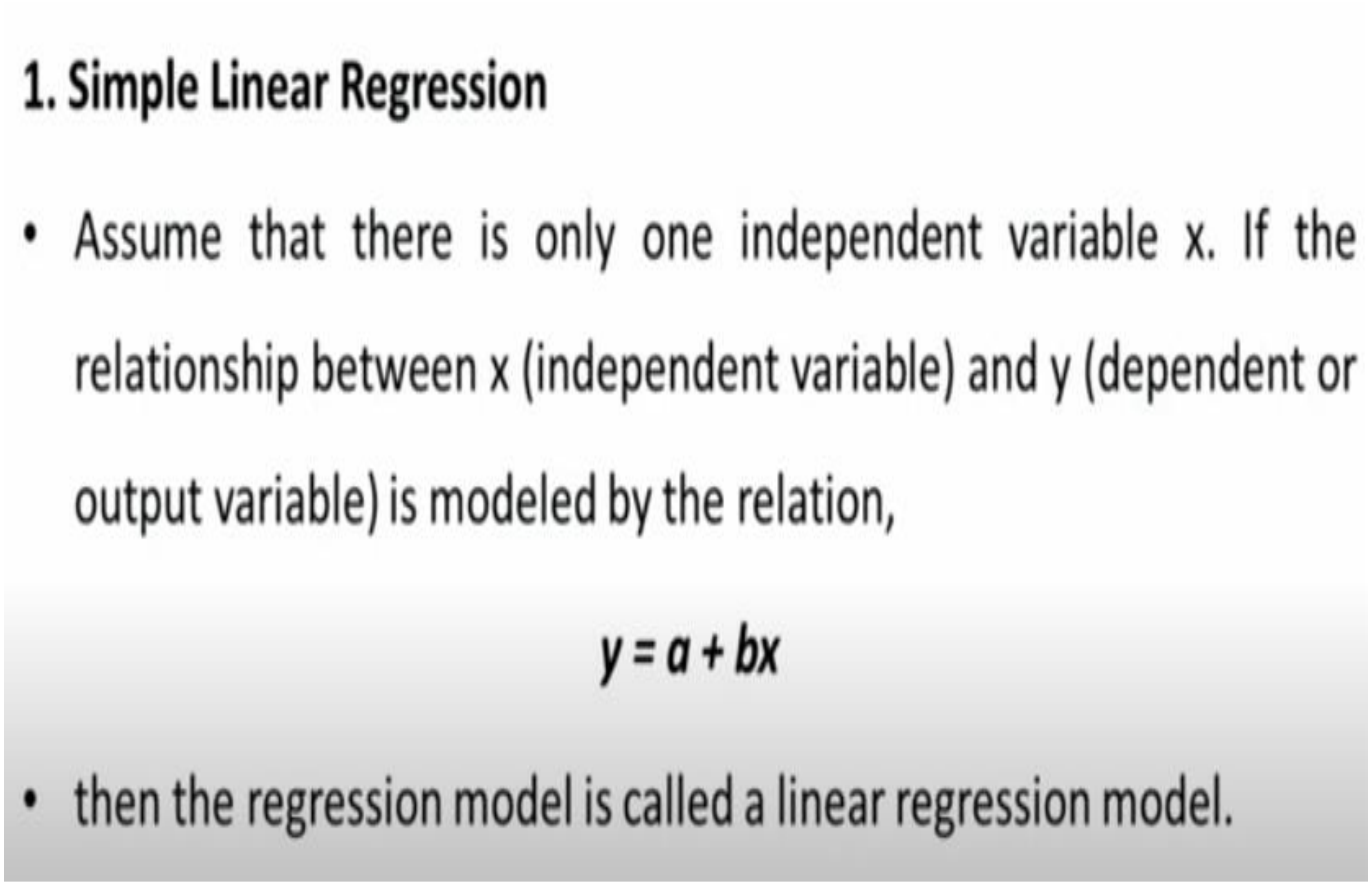
The line equation will be: **$Y = a_0 + a_1 x$**

- Negative Linear Relationship



The line of equation will be: **$Y = -a_0 + a_1 x$**

Contd.



1. Simple Linear Regression

- Assume that there is only one independent variable x. If the relationship between x (independent variable) and y (dependent or output variable) is modeled by the relation,

$$y = a + bx$$

- then the regression model is called a linear regression model.

$y = a + bx$

Method 1:

Steps to find **a** and **b,**

First find the mean and covariance.

Means of x and y are given by

$$\bar{x} = \frac{1}{n}\sum x_i$$

$$\bar{y} = \frac{1}{n}\sum y_i$$

Now the values of a and b can be computed using the following formulas:

Variance of x is given by,

$$\mathrm{Var}\,(x) = \frac{1}{n-1}\sum (x_i - \bar{x}_i)^2$$

The covariance of x and y, denoted by Cov(x, y)is defined as

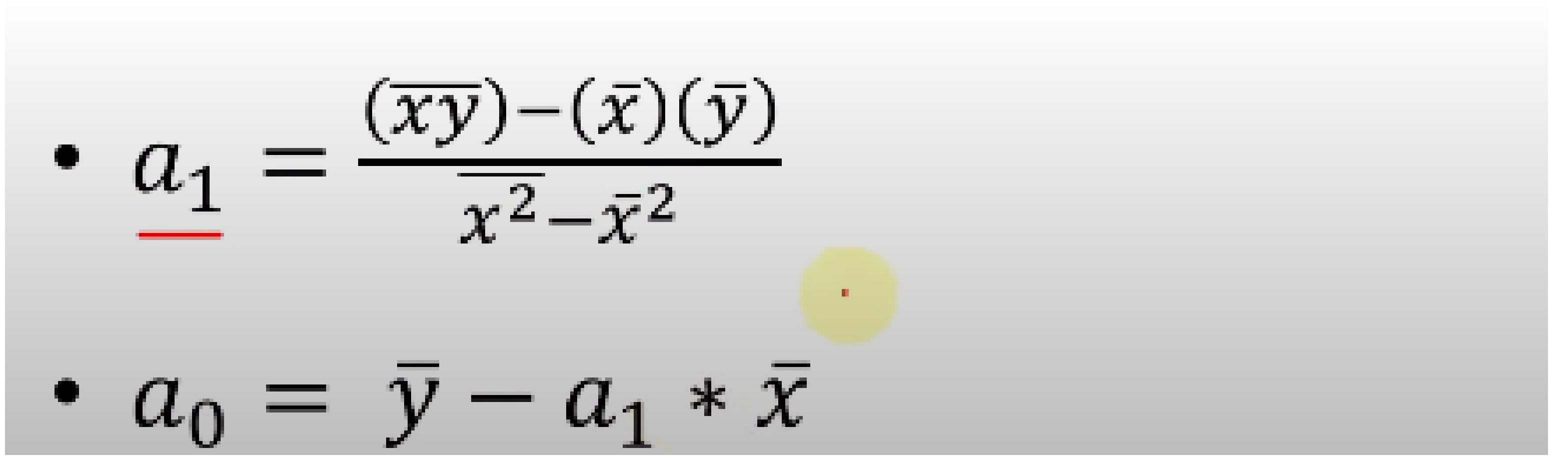
$$\mathrm{v}(x, y) = \frac{1}{n-1}\sum (x_i - \bar{x})(y_i - \bar{y})$$

$$b = \frac{\mathrm{Cov}\,(x, y)}{\mathrm{Var}\,(x)}$$

$$a = \bar{y} - b\bar{x}$$

Method 2:

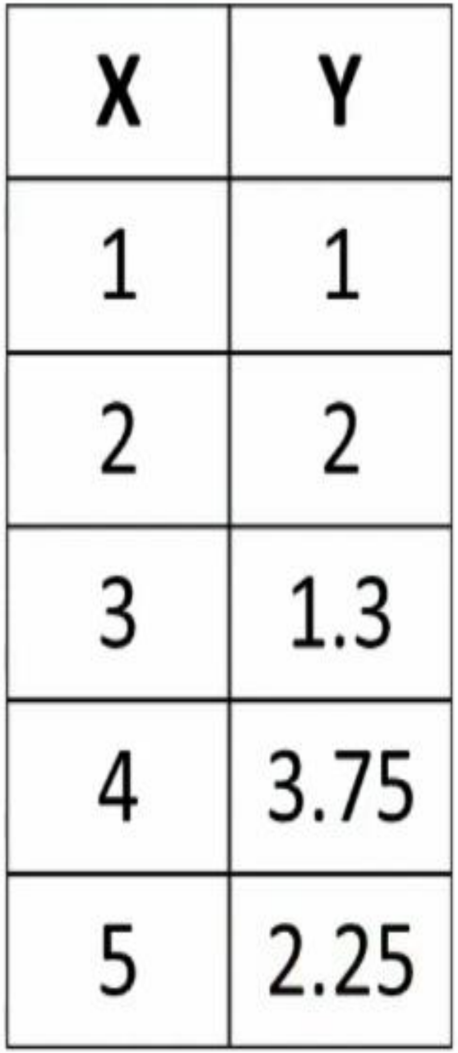
- Linear regression equation is given by
- $y = a_0 + a_1 * \underline{x}$
- $where$



- $\underline{a_1} = \frac{(\overline{xy}) - (\bar{x})(\bar{y})}{\overline{x^2} - \bar{x}^2}$
- $a_0 = \bar{y} - a_1 * \bar{x}$

Example:

- Find out the Equation for Linear Regression line for the following data. Also find out the MAE, MSE, RMSE.



X	Y
1	1
2	2
3	1.3
4	3.75
5	2.25

X	Y
1	1
2	2
3	1.3
4	3.75
5	2.25

$$n = 5$$

$$\begin{aligned}\bar{x} &= \frac{1}{5}(1.0 + 2.0 + 3.0 + 4.0 + 5.0)\\ &= 3.0\end{aligned}$$

$$\begin{aligned}\bar{y} &= \frac{1}{5}(1.00 + 2.00 + 1.30 + 3.75 + 2.25)\\ &= 2.06\end{aligned}$$

$$\mathrm{Cov}(x, y) = \frac{1}{n-1}\sum(x_i - \bar{x})(y_i - \bar{y})$$

$$\begin{aligned}\mathrm{Cov}\,(x, y) &= \frac{1}{4}[(1.0 - 3.0)(1.00 - 2.06) + \cdots + (5.0 - 3.0)(2.25 - 2.06)]\\ &= 1.0625\end{aligned}$$

X	Y
1	1
2	2
3	1.3
4	3.75
5	2.25

$$\mathrm{Var}(x) = \frac{1}{n-1}\sum(x_i - \bar{x_i})^2 \quad \mathrm{Var}(x) = \frac{1}{4}[(1.0-3.0)^2 + \cdots + (5.0-3.0)^2]$$

$$= 2.5$$

$$b = \frac{\mathrm{Cov}(x,y)}{\mathrm{Var}(x)}$$

$$a = \bar{y} - b\bar{x}$$

$$b = \frac{1.0625}{2.5}$$

$$= 0.425$$

$$a = 2.06 - 0.425 \times 3.0$$

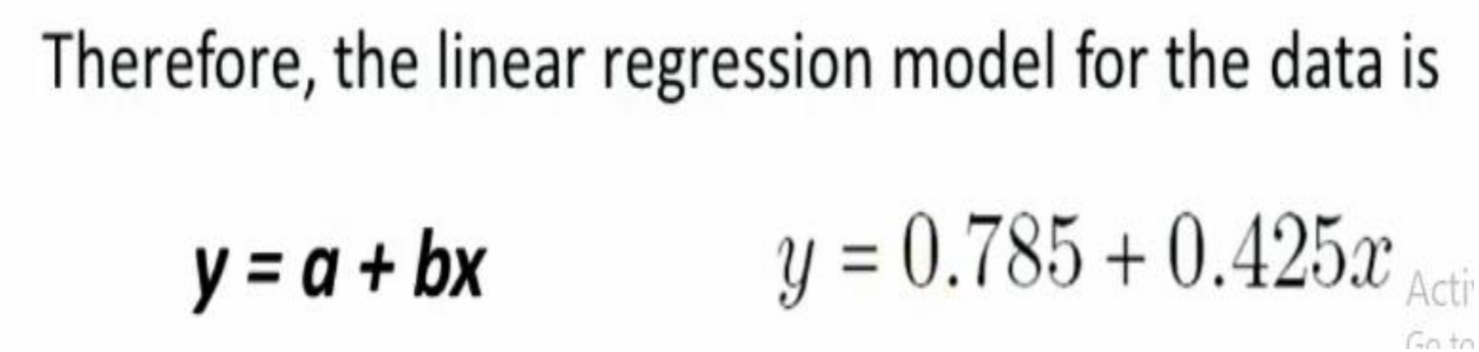
$$= 0.785$$

Therefore, the linear regression model for the data is

y = a + bx

$$y = 0.785 + 0.425x$$

Contd.



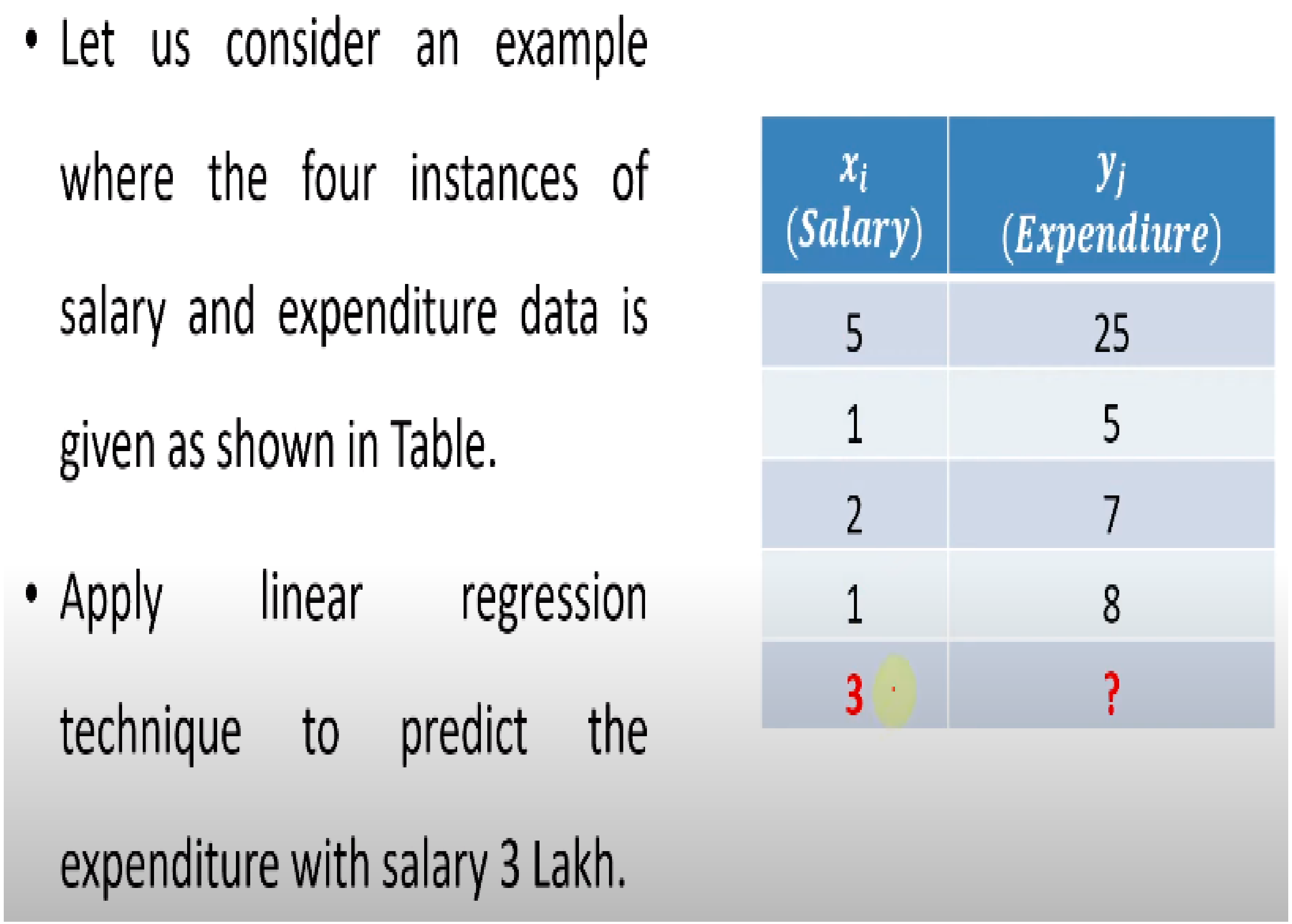
Therefore, the linear regression model for the data is

$$y = a + bx \qquad y = 0.785 + 0.425x$$

X	Y	Predicted Y
1	1	1.21
2	2	1.635
3	1.3	2.06
4	3.75	2.485
5	2.25	2.91

MAE = 0.652
MSE = 0.5582
RMSE = 0.7471

Example



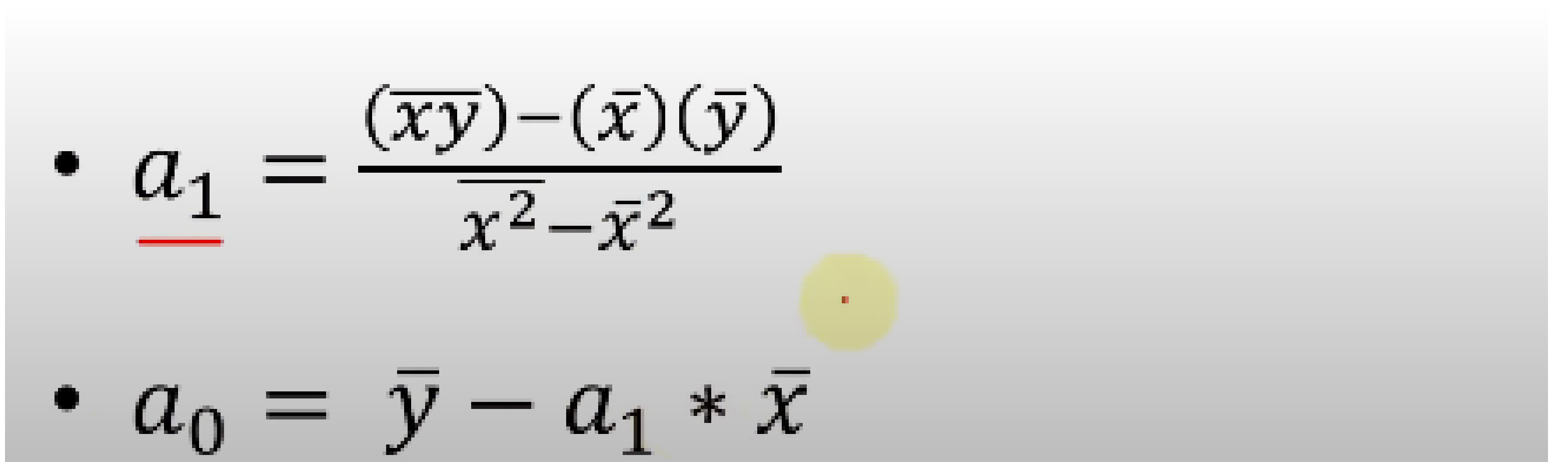
- Let us consider an example where the four instances of salary and expenditure data is given as shown in Table.
- Apply linear regression technique to predict the expenditure with salary 3 Lakh.

x_i $(Salary)$	y_j $(Expendiure)$
5	25
1	5
2	7
1	8
3	?

Also find out the MAE, MSE, RMSE.

Contd.

- Linear regression equation is given by
- $y = a_0 + a_1 * \underline{x}$
- $where$



- $\underline{a_1} = \frac{(\overline{xy}) - (\bar{x})(\bar{y})}{\overline{x^2} - \bar{x}^2}$
- $a_0 = \bar{y} - a_1 * \bar{x}$

	x_i $(Salary)$	y_j $(Expenditure)$	x_i^2	$x_i * y_j$
	5	25	25	125
	1	5	1	5
	2	7	4	14
	1	8	1	8
Sum	9	45	31	152
Average	$\overline{x} = 2.25$	$\overline{y} = 11.25$	$\overline{x^2} = 7.75$	$\overline{xy} = 38$

- $\overline{x} = 2.25$ $\overline{y} = 11.25$ $\overline{x^2} = 7.75$ $\overline{xy} = 38$

- $\underline{a_1} = \frac{(\overline{xy})-(\bar{x})(\bar{y})}{\overline{x^2}-\bar{x}^2} = \frac{38-2.23*11.25}{7.75-2.25^2} = \underline{4.72}$

- $\underline{a_0} = \bar{y} - a_1 * \bar{x} = 11.25 - 4.72 * 2.25 = \underline{0.62}$

- **Regression equation is**

- $\boldsymbol{y = a_0 + a_1 * x}$

- $y = \underline{0.62} + 4.72 * x$

- Regression equation is
- $y = a_0 + a_1 * x$
- $y = 0.62 + 4.72 * x$
- The predicted 5th instance expenditure with salary 3 lakh is,
- $y = 0.62 + 4.72 * 3 = \boxed{14.78}$

Contd.

- Regression equation is
- $y = a_0 + a_1 * x$
- $y = 0.62 + 4.72 * x$

error = (yi - yi pred)

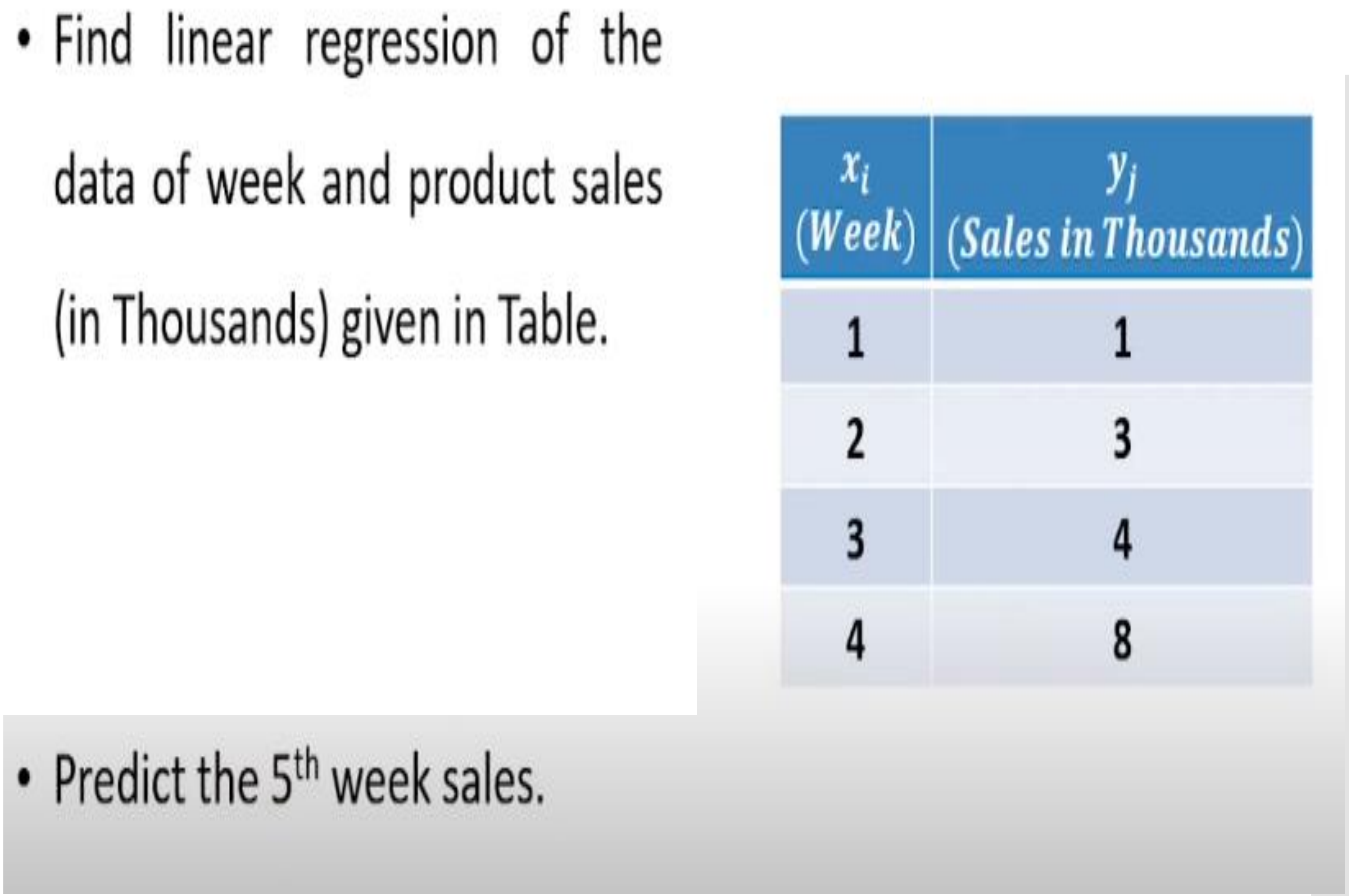
X	Y	Predicted Y	error
5	25	? 24.22	0.78
1	5	? 5.34	0.34
2	7	? 10.06	3.06
1	8	? 5.34	2.66

MAE =? 1.368

MSE = ? 4.29

RMSE = ? 2.07

Example



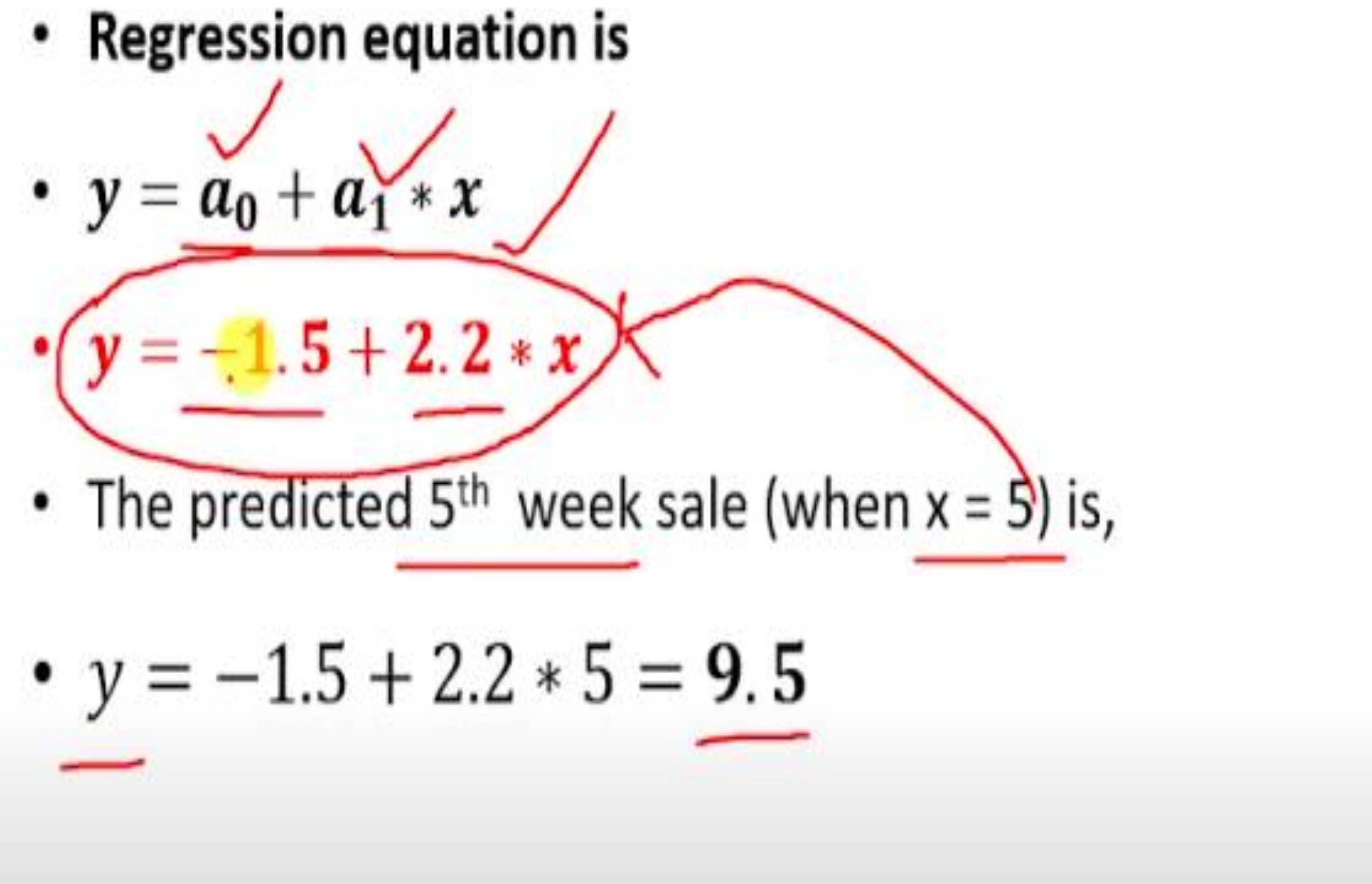
- Find linear regression of the data of week and product sales (in Thousands) given in Table.

x_i *(Week)*	y_j *(Sales in Thousands)*
1	1
2	3
3	4
4	8

- Predict the 5th week sales.

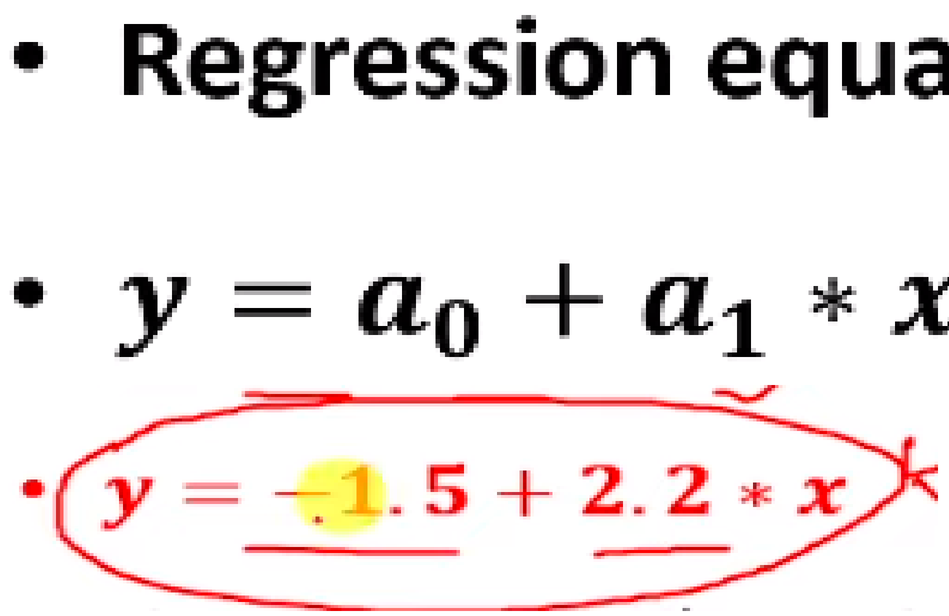
Also find out the MAE, MSE, RMSE.

Solution:



- Regression equation is
- $y = a_0 + a_1 * x$
- $y = -1.5 + 2.2 * x$
- The predicted 5th week sale (when x = 5) is,
- $y = -1.5 + 2.2 * 5 = 9.5$

Contd.



- Regression equation is

- $y = a_0 + a_1 * x$

- $y = \underline{-1.5} + \underline{2.2} * x$

X	Y	Predicted Y	Error
1	1	? 0.7	0.3
2	3	? 2.9	0.1
3	4	? 5.1	1.1
4	8	? 7.3	0.7

MAE =? 0.55

MSE = ? 0.45

RMSE = ? 0.671

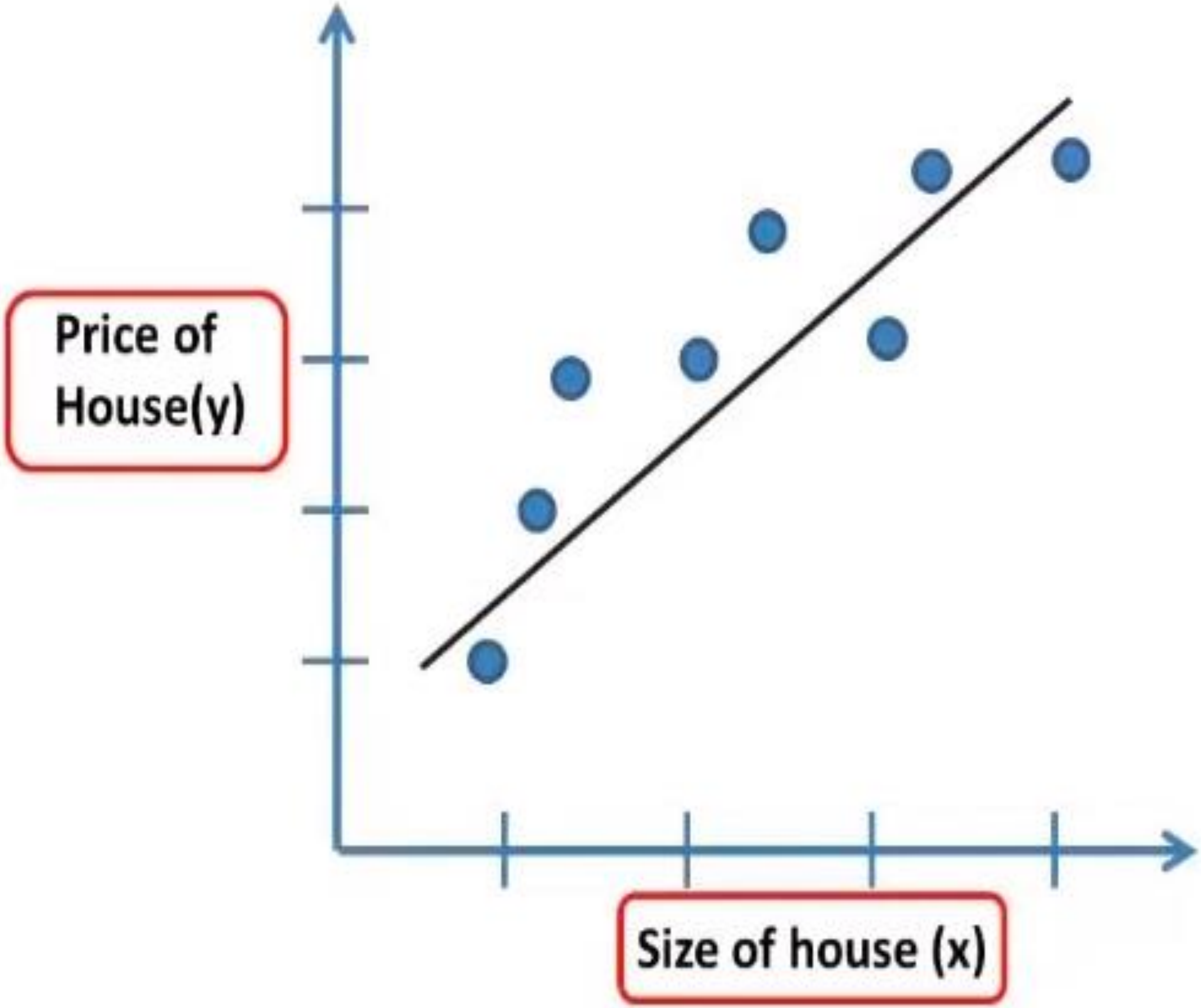
DEMO

Linear Regression with One Variable – Simple Linear Regression

- https://colab.research.google.com/drive/152-OPgXtr72WMTaOhIPT08vbN_vnRZuO?usp=sharing

Size of house(sqm.)		Price of house
64		$ 200,000
80		$ 250,000
63		$ 230,000
100		$ 320,000
128		$ 300,000
144		$ 450,000
...		....
...		....
81		??

..we want to know price(**output**) of a new house based on its Size(**input**).

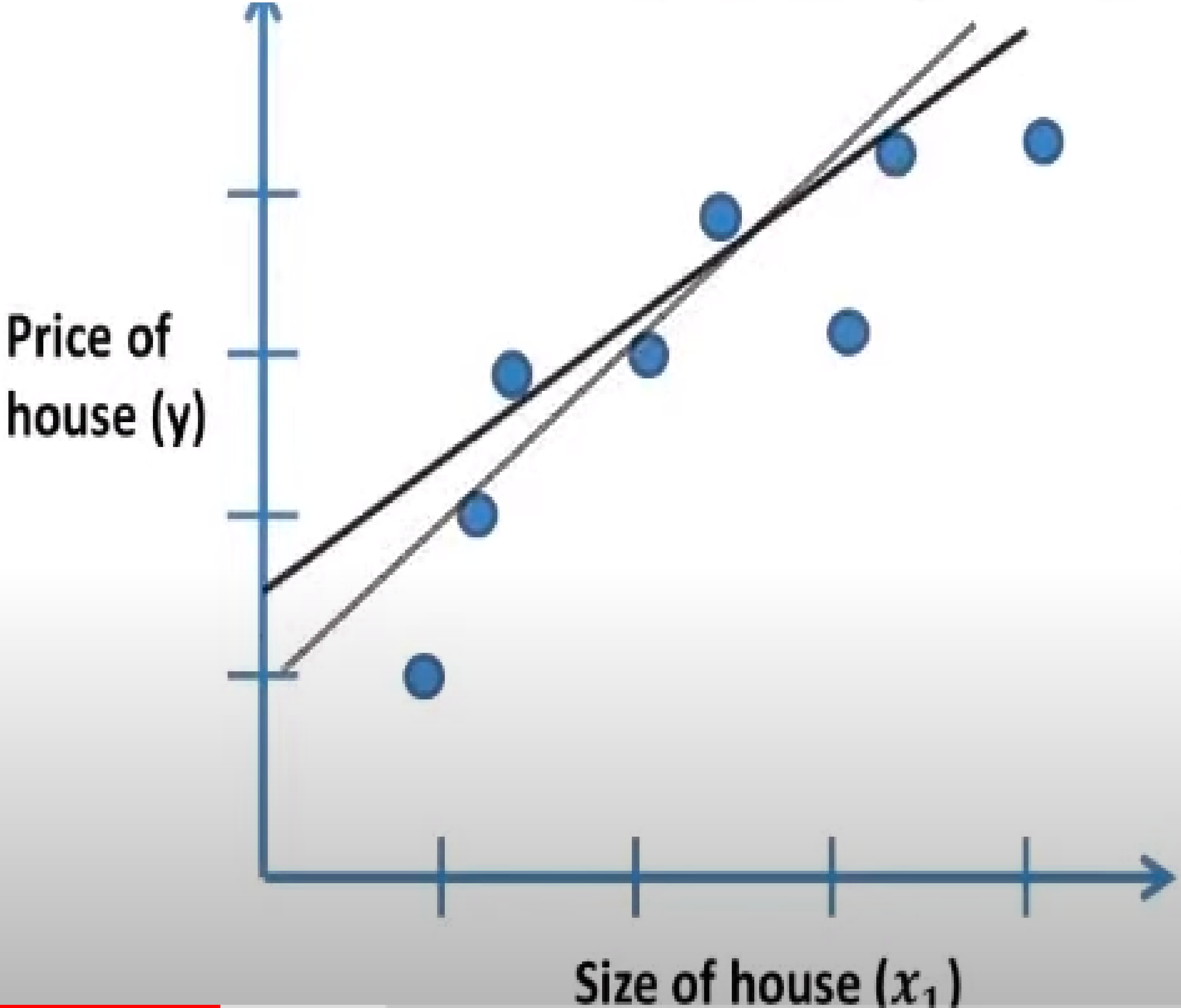


any **linear relationship** between 2 variables can be represented as a **straight line**.

$$\boldsymbol{y = a_0 + a_1 x}$$

*"y" is **target** -> Price of house*
*"x" is **feature** -> Size of house*
a_0 *and* a_1 are **model parameters**

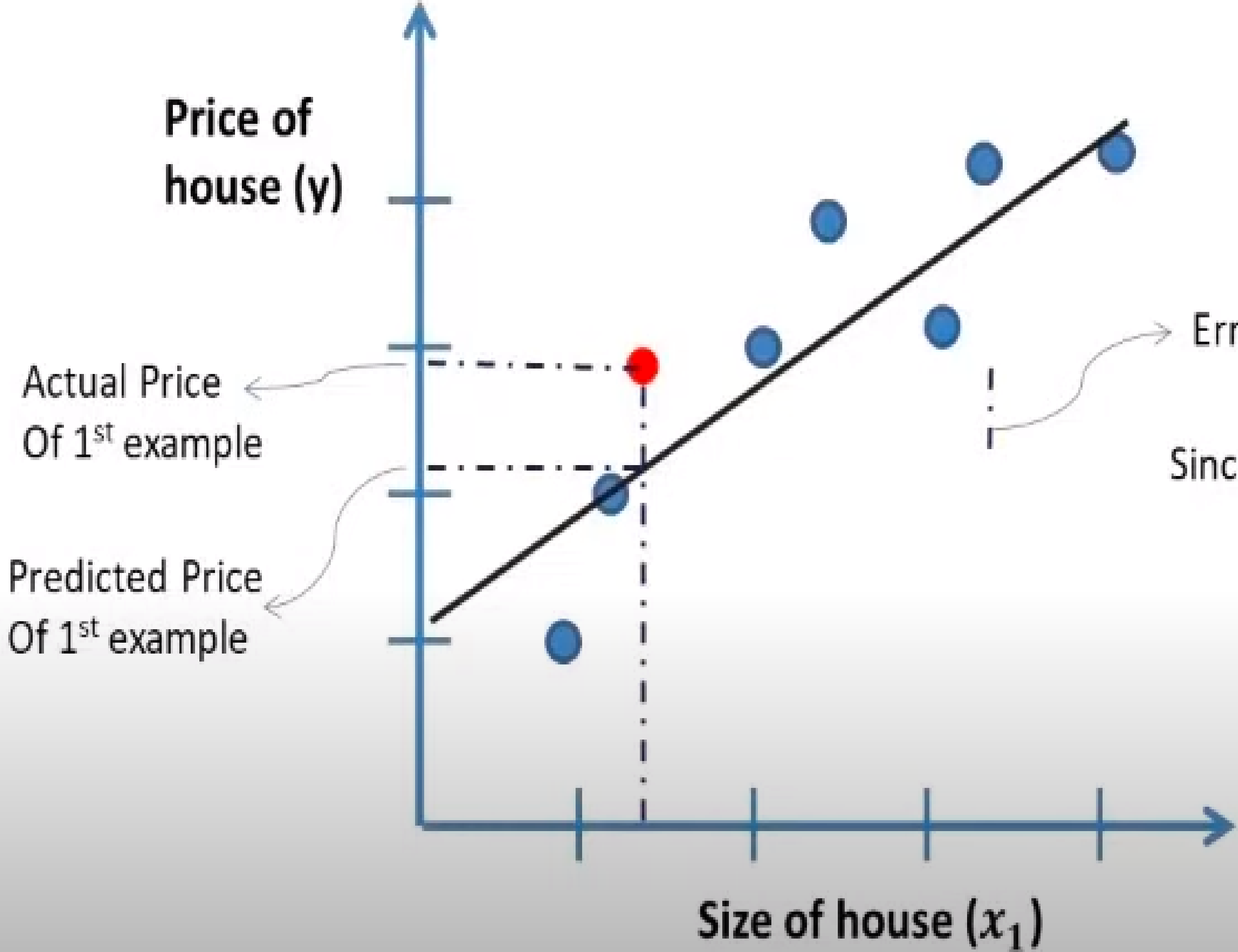
We need to settle the case on a value of parameters $\boldsymbol{a_0}$ and $\boldsymbol{a_1}$ corresponding to which straight line **fits best** to the data.



As per our linear regression model, We need to fit a straight line to it…

$$y = a_0 + a_1 x_1$$

…depending on values of $\boldsymbol{a_0}$ and $\boldsymbol{a_1}$, we can have many possibilities, which look promising.

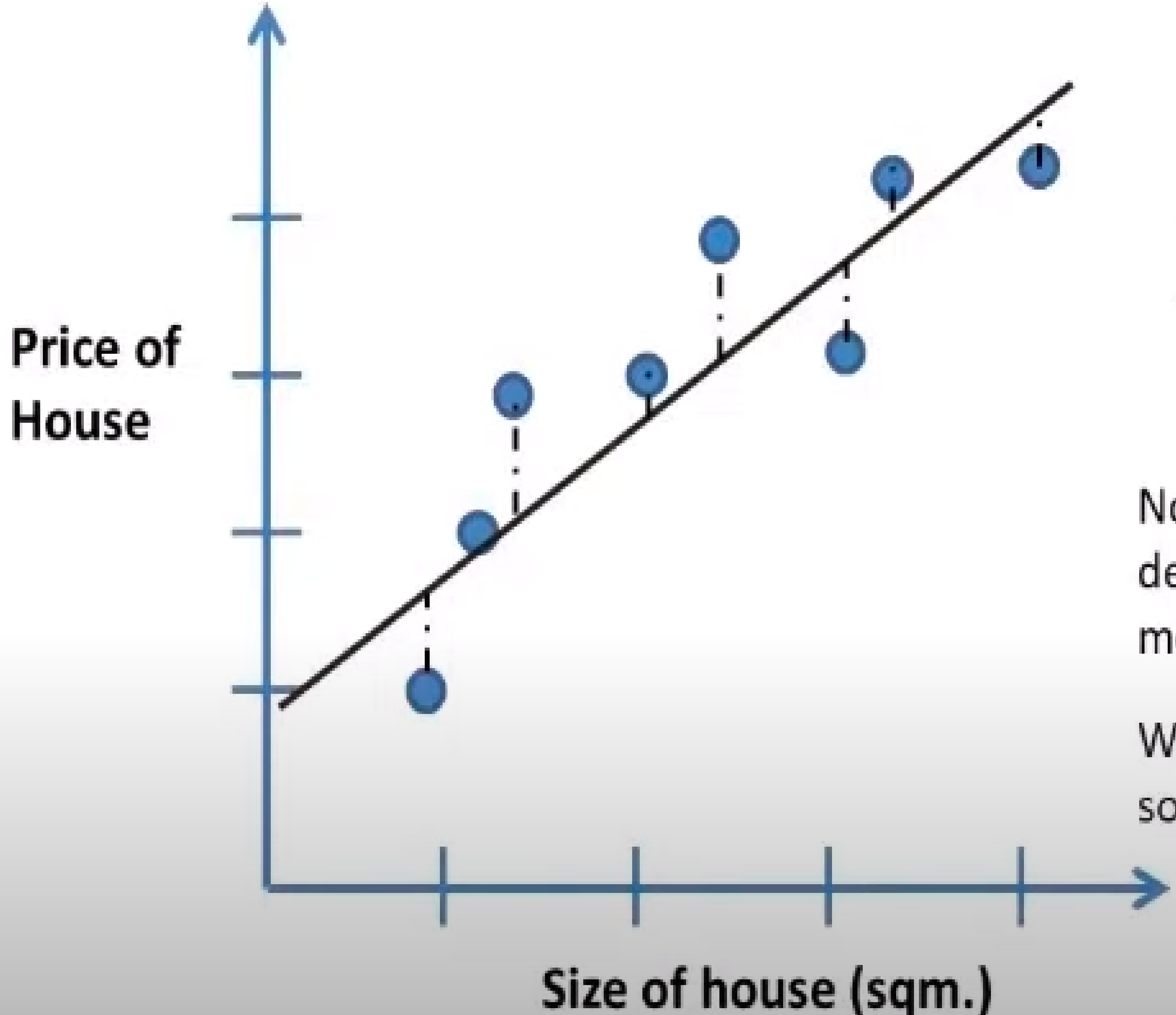


Actual Price Of 1st example

Predicted Price Of 1st example

Error Term(e) : y_{actual} - $y_{predicted}$

Since it is for 1st example lets call it $\boldsymbol{e_1}$

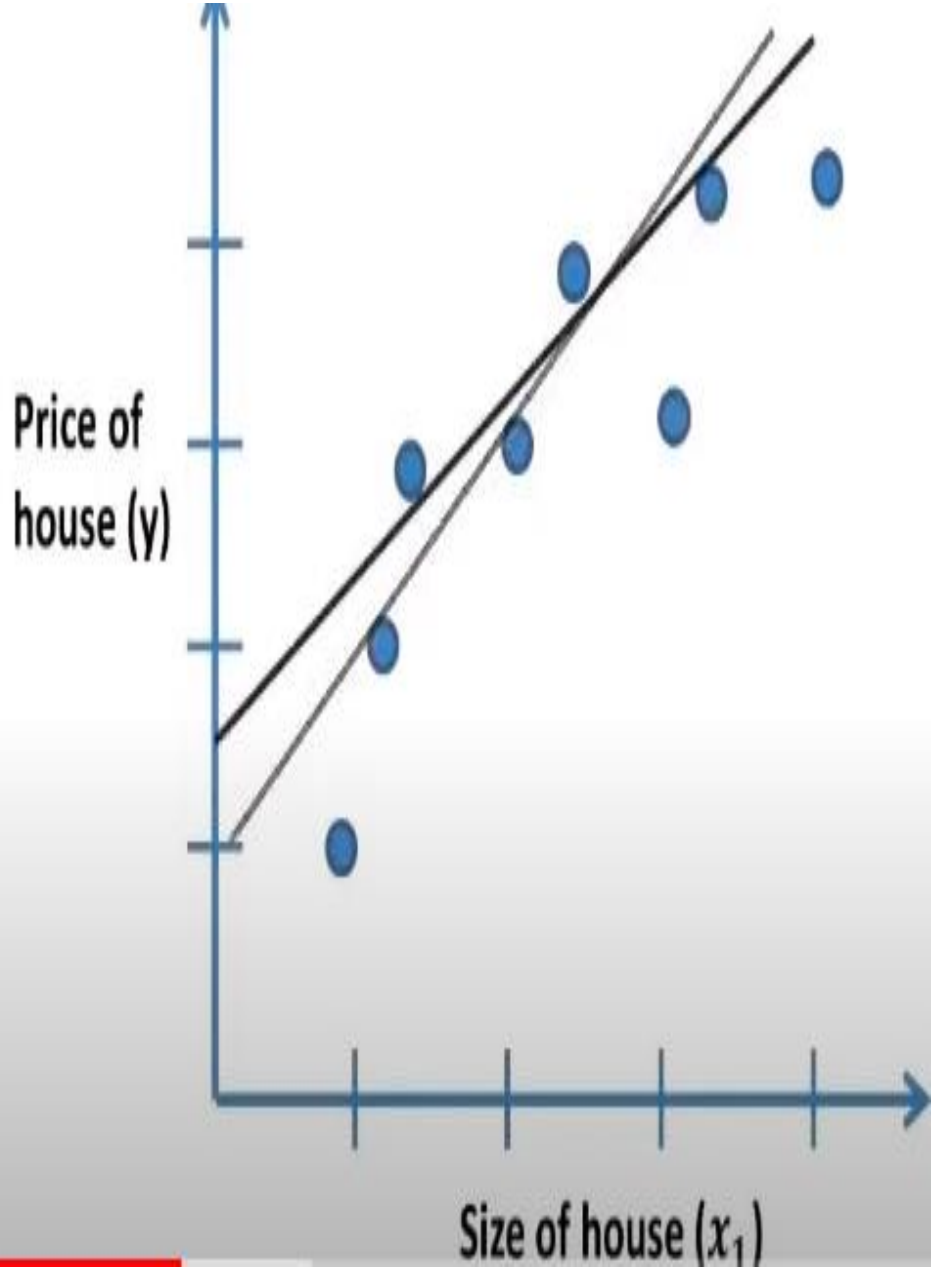


for i^{th} example, we define error term as

$$e_i = Y_{predicted}{}^{i} \ \ - Y_{\ actual}{}^{i}$$

Now $\boldsymbol{e_i}$can be **+ve or –ve**, depending on whether $\boldsymbol{y_{actual}}$ is more or $\boldsymbol{y_{predicted}}$.

We will ***square*** $\boldsymbol{e_i}$to make it positive so the order doesn't matter.



- **How to find “Best Fit” line?**

- Our main goal is to find the best fit line that means the error between predicted values and actual values should be minimized. The best fit line will have the least error.

- The goal of the linear regression algorithm is to get the best values for ao and a1 to find the best fit line. The best fit line should minimize the error between predicted values and actual values, ensuring that the error is minimized.

Gradient descent with Linear Regression

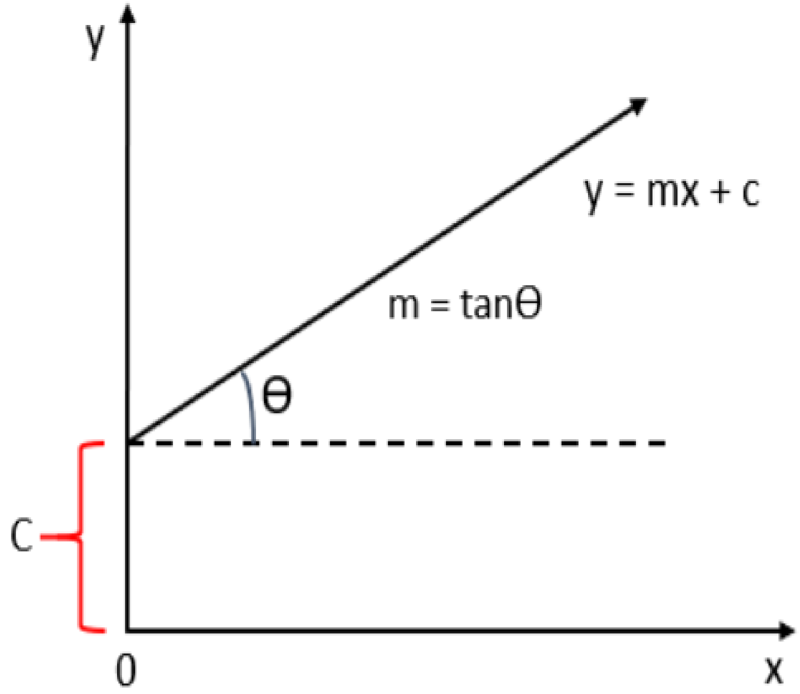
- This straight line is represented using the following formula:
- y = mx +c

Where, y: dependent variable

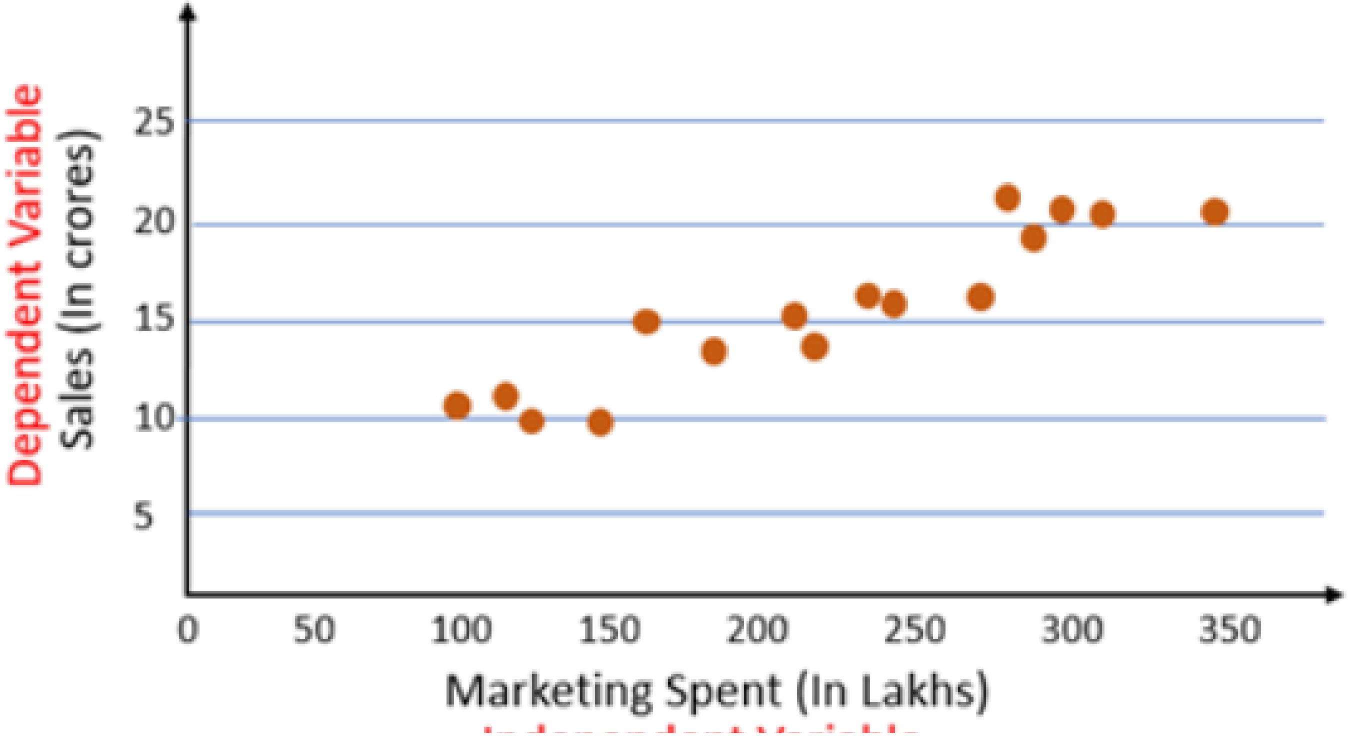
x: independent variable

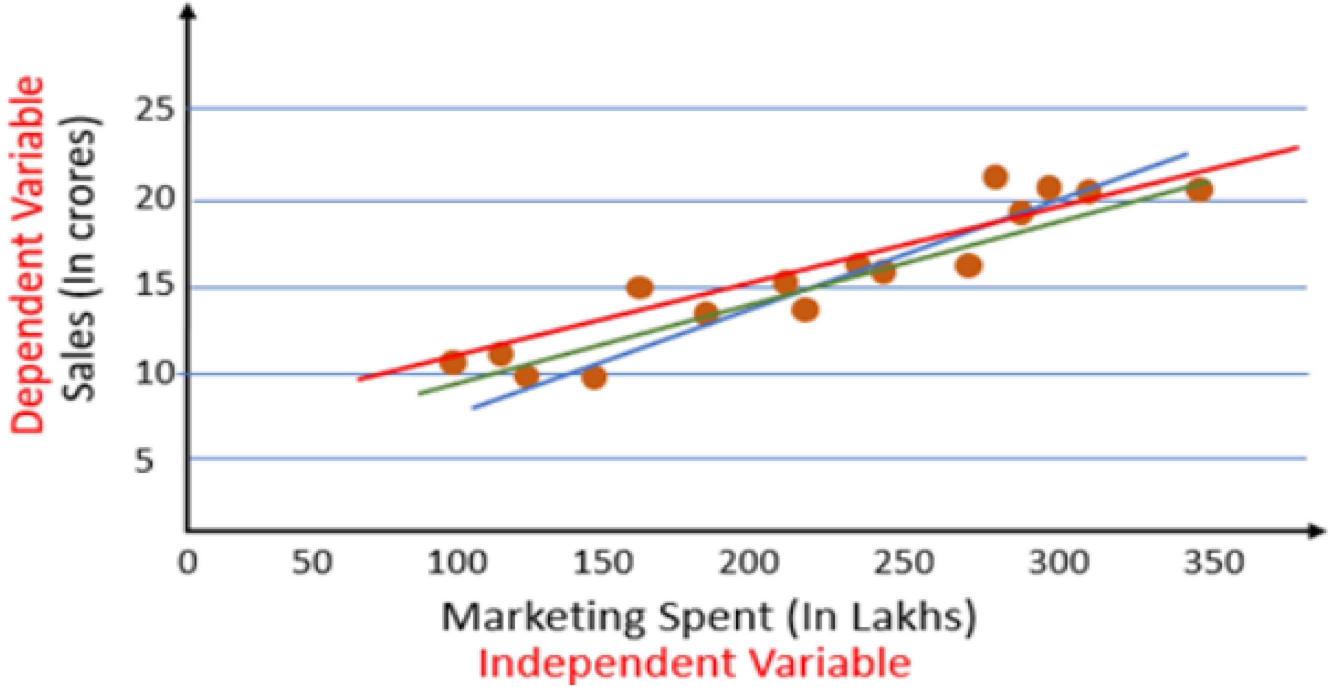
m: Slope of the line

c: y intercept



Contd.


Dependent Variable
Sales (In crores)
25
20
15
10
5
0
50
100
150
200
250
300
350
Marketing Spent (In Lakhs)


Dependent Variable
Sales (In crores)
25
20
15
10
5
0
50
100
150
200
250
300
350
Marketing Spent (In Lakhs)
Independent Variable

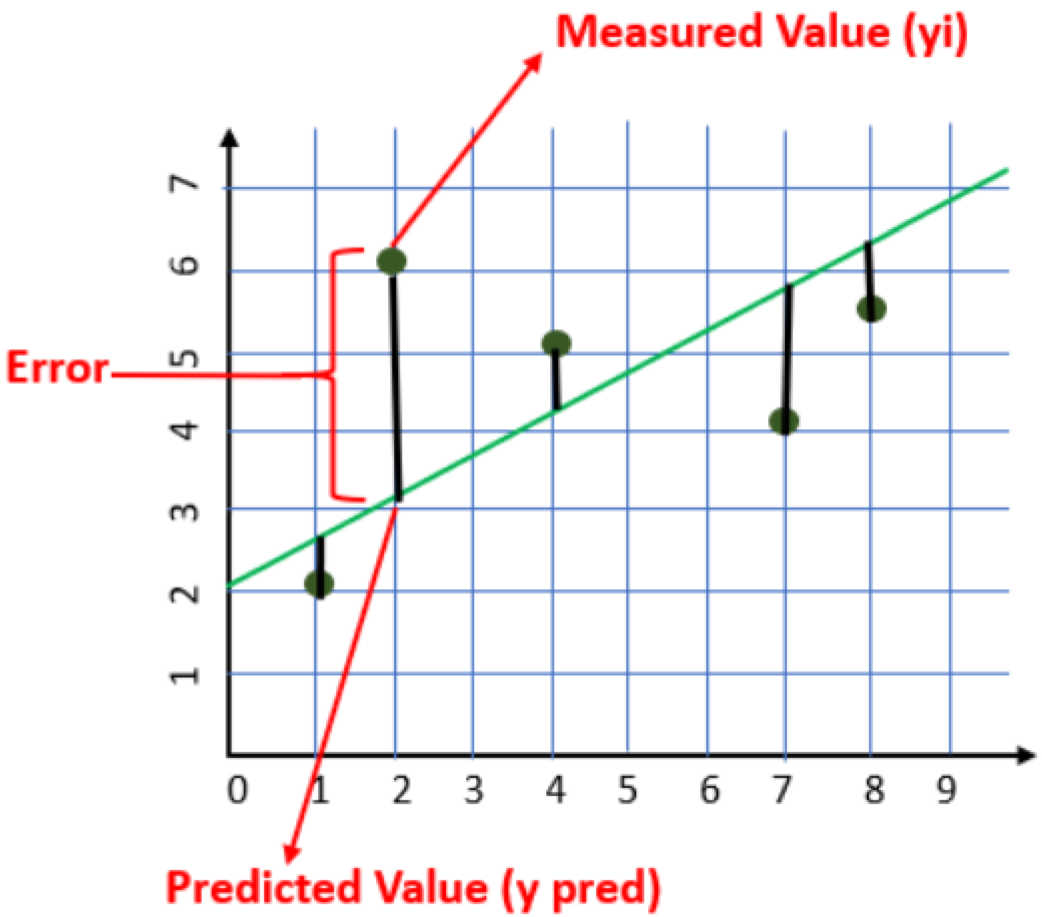
Contd.

- From the scatter plot we can see there is a linear relationship between Sales and marketing spent. The next step is to find a straight line between Sales and Marketing that explain the relationship between them. But there can be multiple lines that can pass through these points.
- So how do we know which of these lines is the best fit line?

Contd.

Cost Function

- The cost is the error in our predicted value. We will use the Mean Squared Error function to calculate the cost.



Contd.

$$Cost\ Function(MSE) = \frac{1}{n}\sum_{i=0}^{n}(y_i - y_{i\,pred})^2$$

Replace $y_{i\,pred}$ with $mx_i + c$

$$Cost\ Function(MSE) = \frac{1}{n}\sum_{i=0}^{n}(y_i - (mx_i + c))^2$$

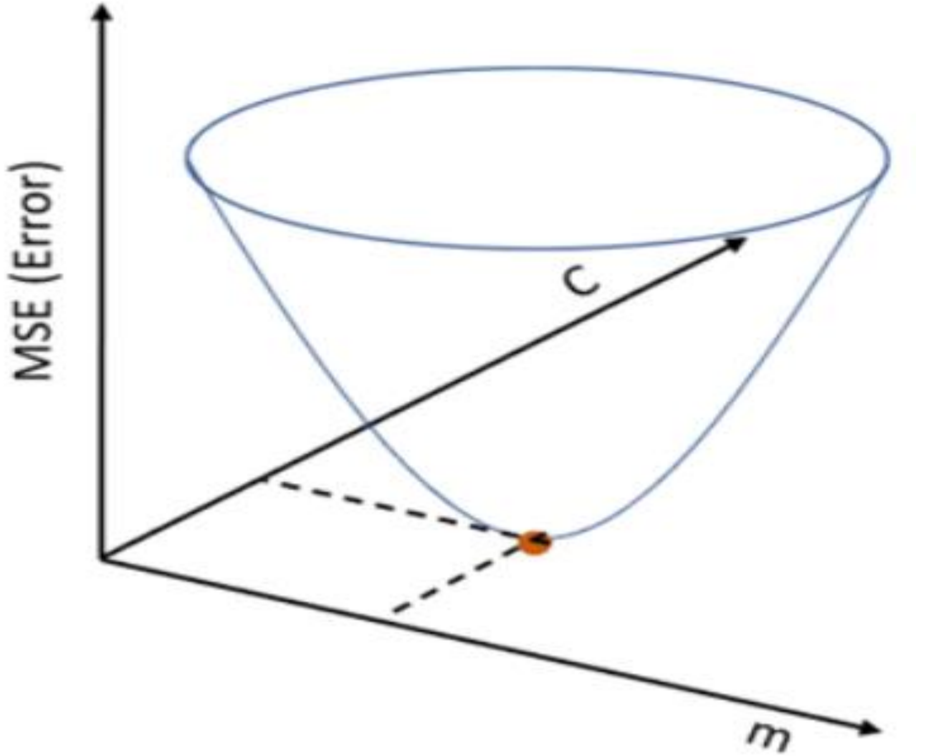
- Our goal is to minimize the cost as much as possible in order to find the best fit line. We are not going to try all the permutation and combination of m and c (inefficient way) to find the best-fit line. For that, we will use Gradient Descent Algorithm.

Contd.

- Gradient Descent is an algorithm that finds the best-fit line for a given training dataset in a smaller number of iterations.

- If we plot m and c against MSE, it will acquire a bowl shape (As shown in the diagram below)

- For some combination of m and c, we will get the least Error (MSE). That combination of m and c will give us our best fit line.



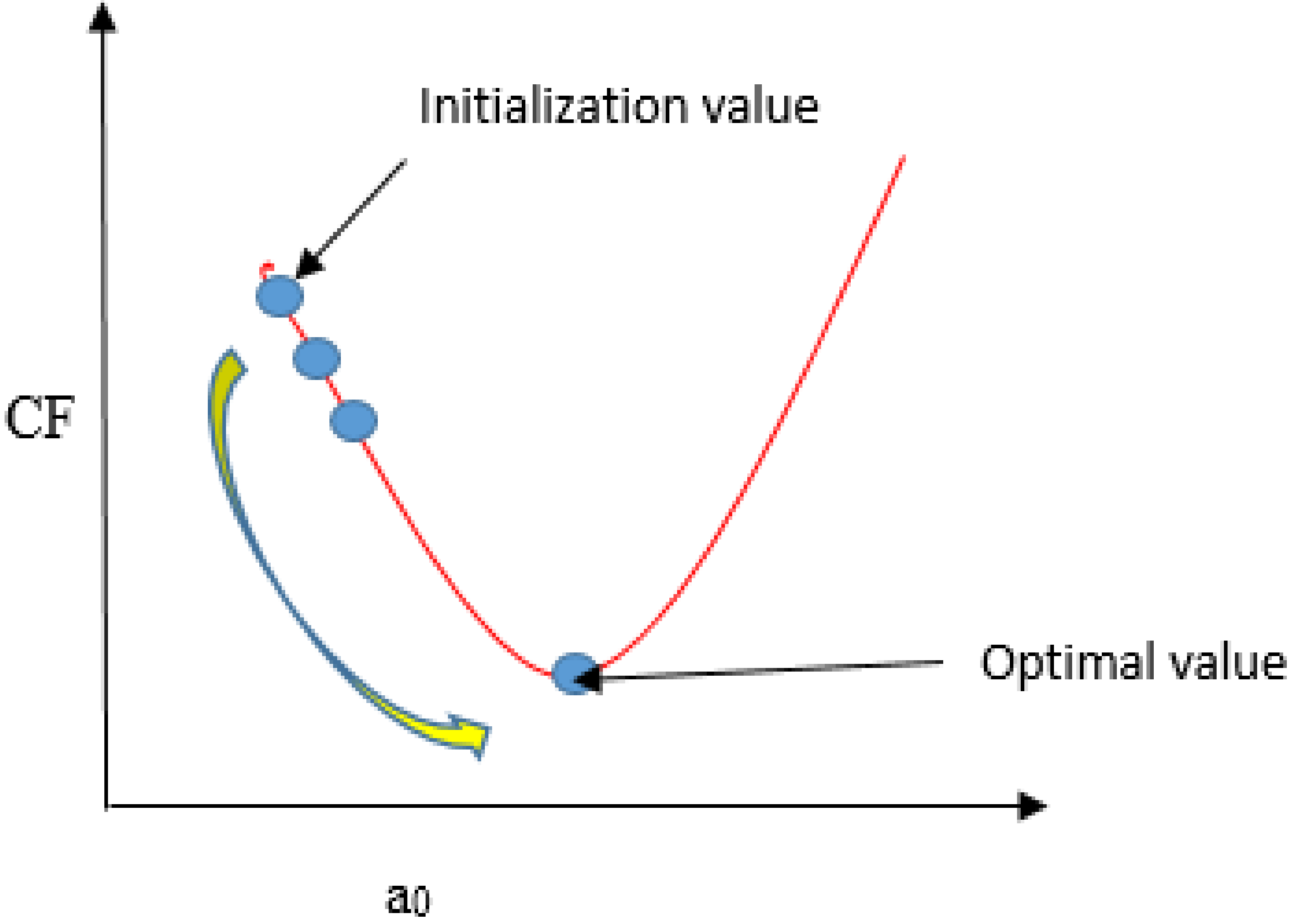
What is Gradient descent ?

$$y = a_0 + a_1 x_1$$

Contd.

- Gradient descent is a method of updating a0 and a1 to minimize the cost function (MSE).
- A regression model uses gradient descent to update the coefficients of the line (a0, a1a) by reducing the cost function by a random selection of coefficient values and then iteratively update the values to reach the minimum cost function.
- The gradients use the partial derivatives to update the values of a0 and a1. Alpha is the learning rate.

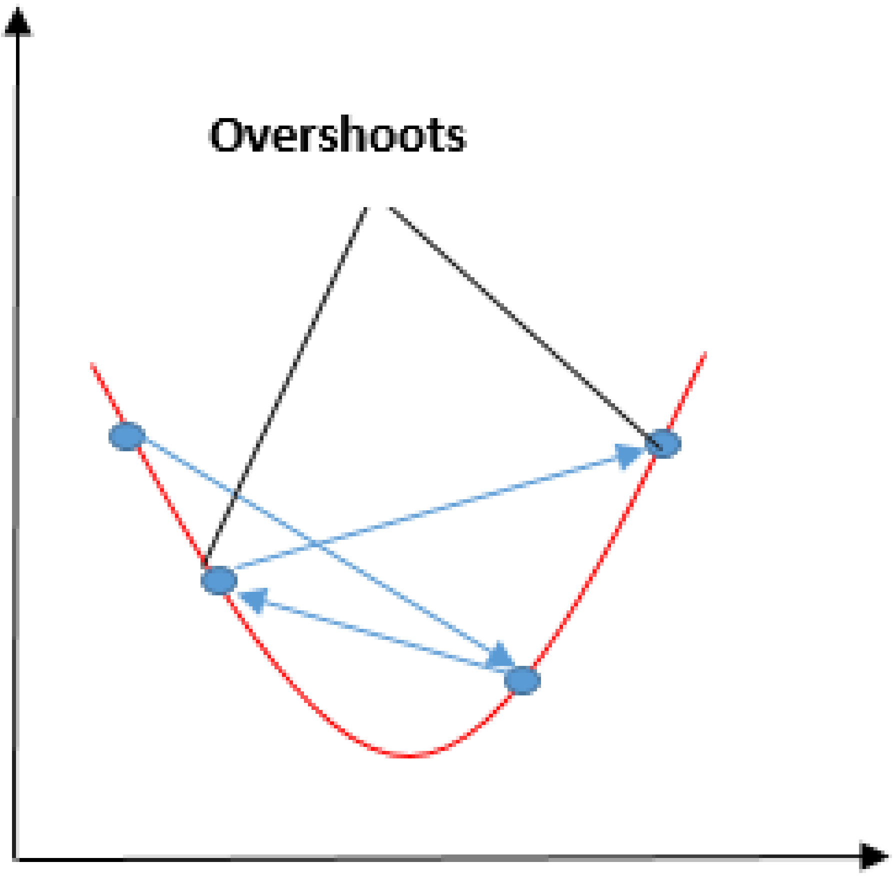
Contd.



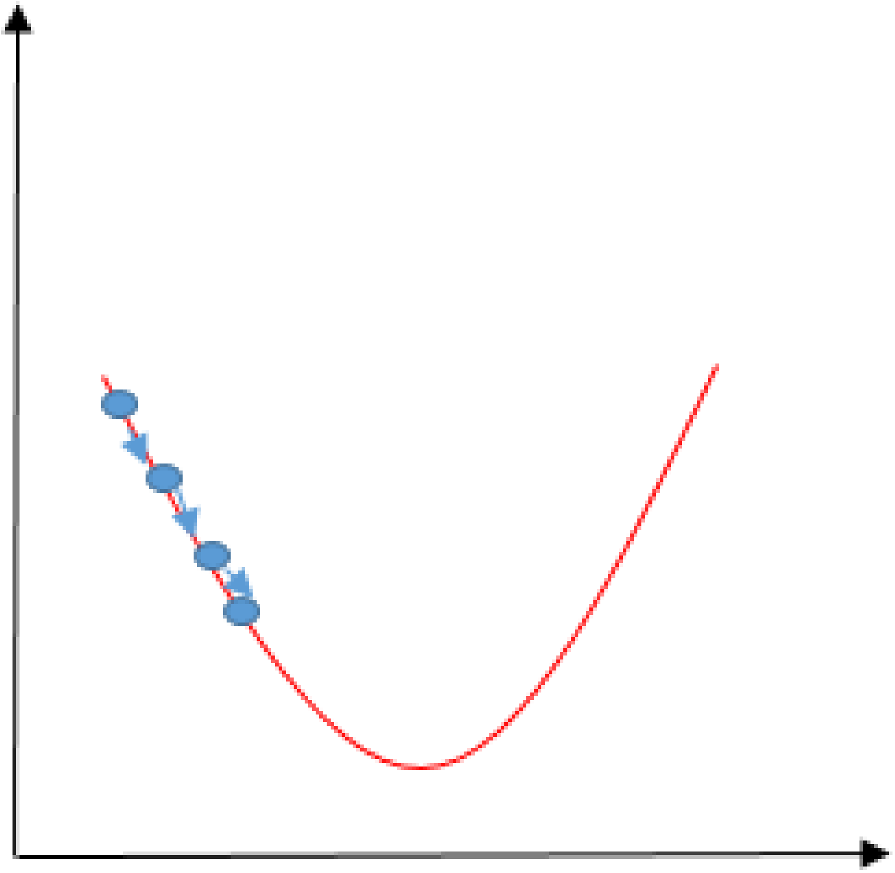
You are standing at the topmost point in the pit, and your objective is to reach the bottom of the pit.

You can only take a discrete number of steps to reach the bottom

In the gradient descent algorithm, **the number of steps you take is the learning rate**



High learning rate



Low learning rate

If you choose to **take longer steps each time**, you may get to sooner but, there is a **chance that you could overshoot the bottom** of the pit and not near the bottom.

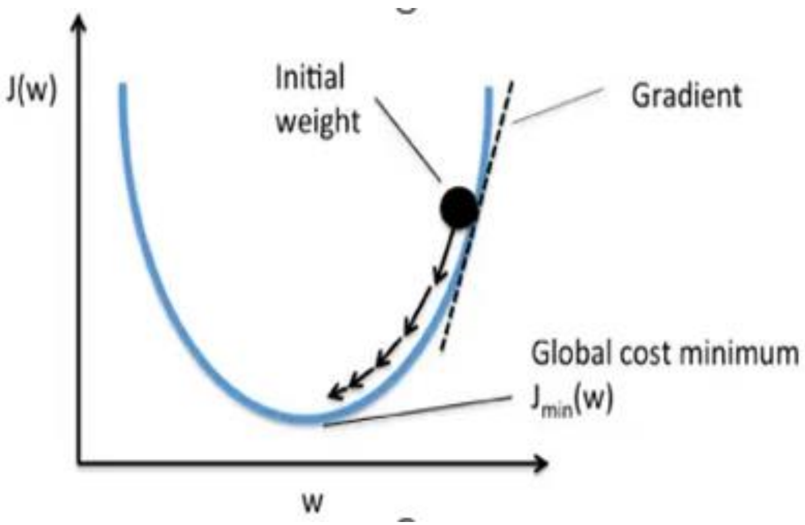
If you decide to **take one footstep at a time**, you would eventually get to the bottom of the pit but**, this would take a longer time**.

Contd.

- Imagine a pit in the shape of U. You are standing at the topmost point in the pit, and your objective is to reach the bottom of the pit.
- There is a treasure, and **you can only take a discrete number of steps to reach the bottom**.
- If you decide to **take one footstep at a time**, you would eventually get to the bottom of the pit but**, this would take a longer time**.
- If you choose to **take longer steps each time**, you may get to sooner but, there is a **chance that you could overshoot the bottom** of the pit and not near the bottom.
- In the gradient descent algorithm, **the number of steps you take is the learning rate**, and this decides how fast the algorithm converges to the minima.

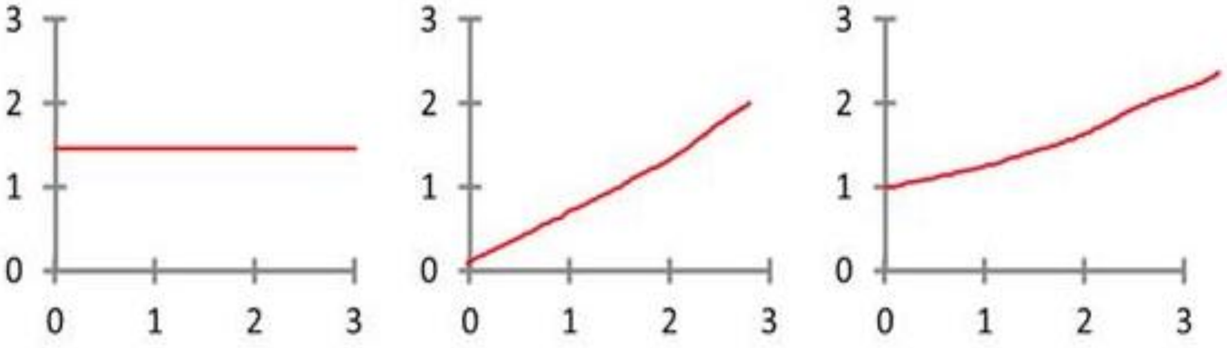
Contd.

- **Gradient Descent**



Standard Learning Rate
LR = 0.01 (Trial & Error method)

Hypothesis Example: $h\theta(x) = \theta_0 + \theta_1 x$ the equation of the line **θ0,θ1 are parameters** and how the effect,

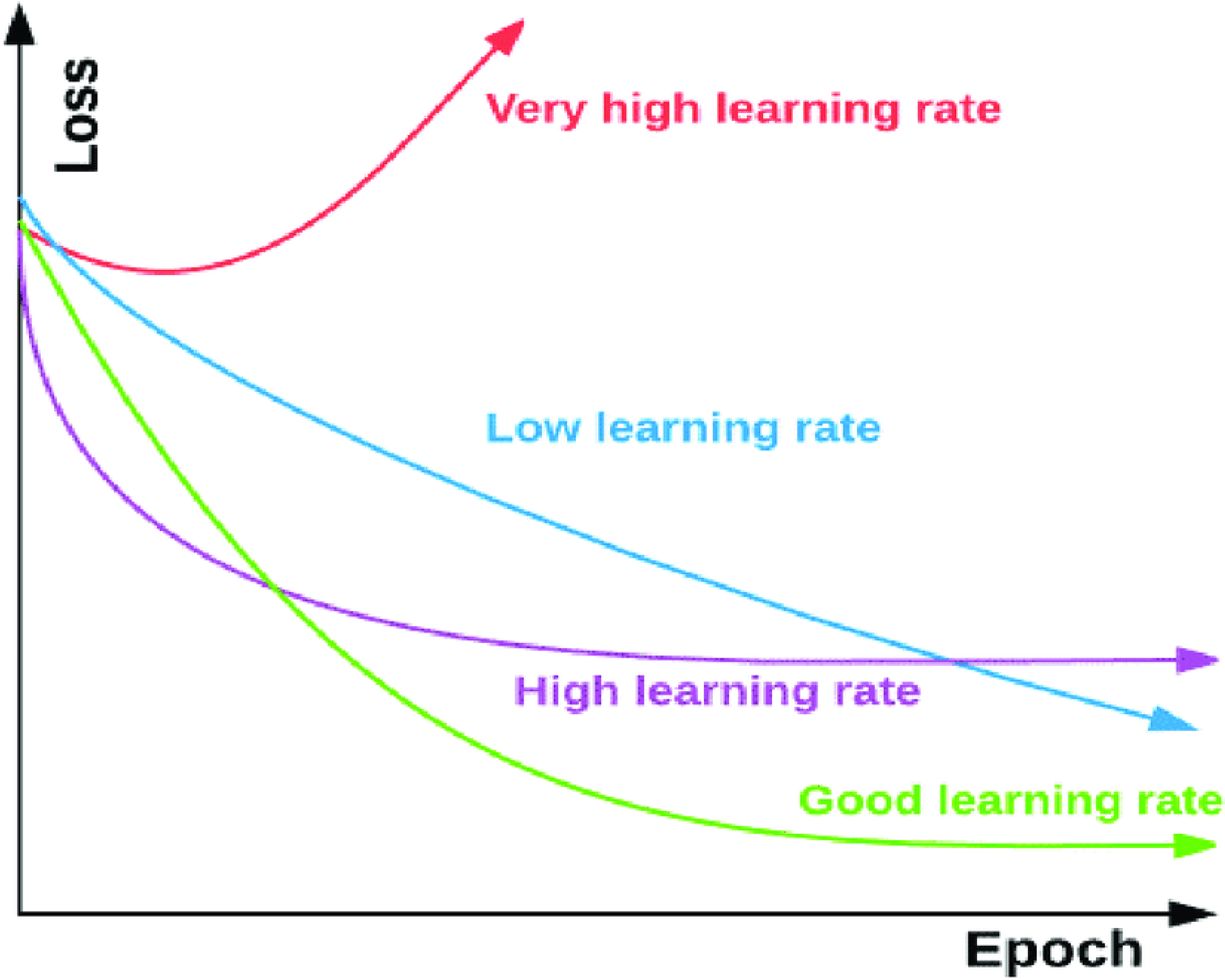


$\theta_0 = 1.5$
$\theta_1 = 0$

$\theta_0 = 0$
$\theta_1 = 0.5$

$\theta_0 = 1$
$\theta_1 = 0.5$

Contd.


Loss
Very high learning rate
Low learning rate
High learning rate
Good learning rate
Epoch

Step by Step Algorithm

Step 1: Let m = 0 and c = 0. Let L be our learning rate. It could be a small value like 0.01 for good accuracy.

- Learning rate gives the rate of speed where the gradient moves during gradient descent. Setting it too high would make your path instable, too low would make convergence slow. Put it to zero means your model isn't learning anything from the gradients.

Contd.

Step 2: Calculate the partial derivative of the Cost function with respect to m. Let partial derivative of the Cost function with respect to m be D_m (With little change in m how much Cost function changes).

$$D_m = \frac{\partial(Cost\ Function)}{\partial m} = \frac{\partial}{\partial m}\left(\frac{1}{n}\sum_{i=0}^{n}(y_i - y_{i\,pred})^2\right)$$

$$= \frac{1}{n}\frac{\partial}{\partial m}\left(\sum_{i=0}^{n}(y_i - (mx_i + c))^2\right)$$

$$= \frac{1}{n}\frac{\partial}{\partial m}\left(\sum_{i=0}^{n}(y_i^2 + m^2x_i^2 + c^2 + 2mx_ic - 2y_imx_i - 2y_ic)\right)$$

$$= \frac{-2}{n}\sum_{i=0}^{n}x_i(y_i - (mx_i + c))$$

$$= \frac{-2}{n}\sum_{i=0}^{n}x_i(y_i - y_{i\,pred})$$

Contd.

- Similarly, let’s find the partial derivative with respect to c. Let partial derivative of the Cost function with respect to c be D$_c$ (With little change in c how much Cost function changes).

$$D_c = \frac{\partial(Cost\ Function)}{\partial c} = \frac{\partial}{\partial c}\left(\frac{1}{n}\sum_{i=0}^{n}(y_i - y_{i\,pred})^2\right)$$

$$= \frac{1}{n}\frac{\partial}{\partial c}\left(\sum_{i=0}^{n}(y_i - (mx_i + c))^2\right)$$

$$= \frac{1}{n}\frac{\partial}{\partial c}\left(\sum_{i=0}^{n}(y_i^2 + m^2x_i^2 + c^2 + 2mx_ic - 2y_imx_i - 2y_ic)\right)$$

$$= \frac{-2}{n}\sum_{i=0}^{n}(y_i - (mx_i + c))$$

$$\frac{-2}{n}\sum_{i=0}^{n}(y_i - y_{i\,pred})$$

Contd.

- **Step 3:** Now update the current values of m and c using the following equation:

$$m = m - LD_m$$

$$c = c - LDc$$

Contd.

Step 4: We will repeat this process until our Cost function is very small (ideally 0).

Gradient Descent Algorithm gives optimum values of m and c of the linear regression equation. With these values of m and c, we will get the equation of the best-fit line and ready to make predictions.

DEMO

Linear Regression with Gradient Descent

With Sample Data:

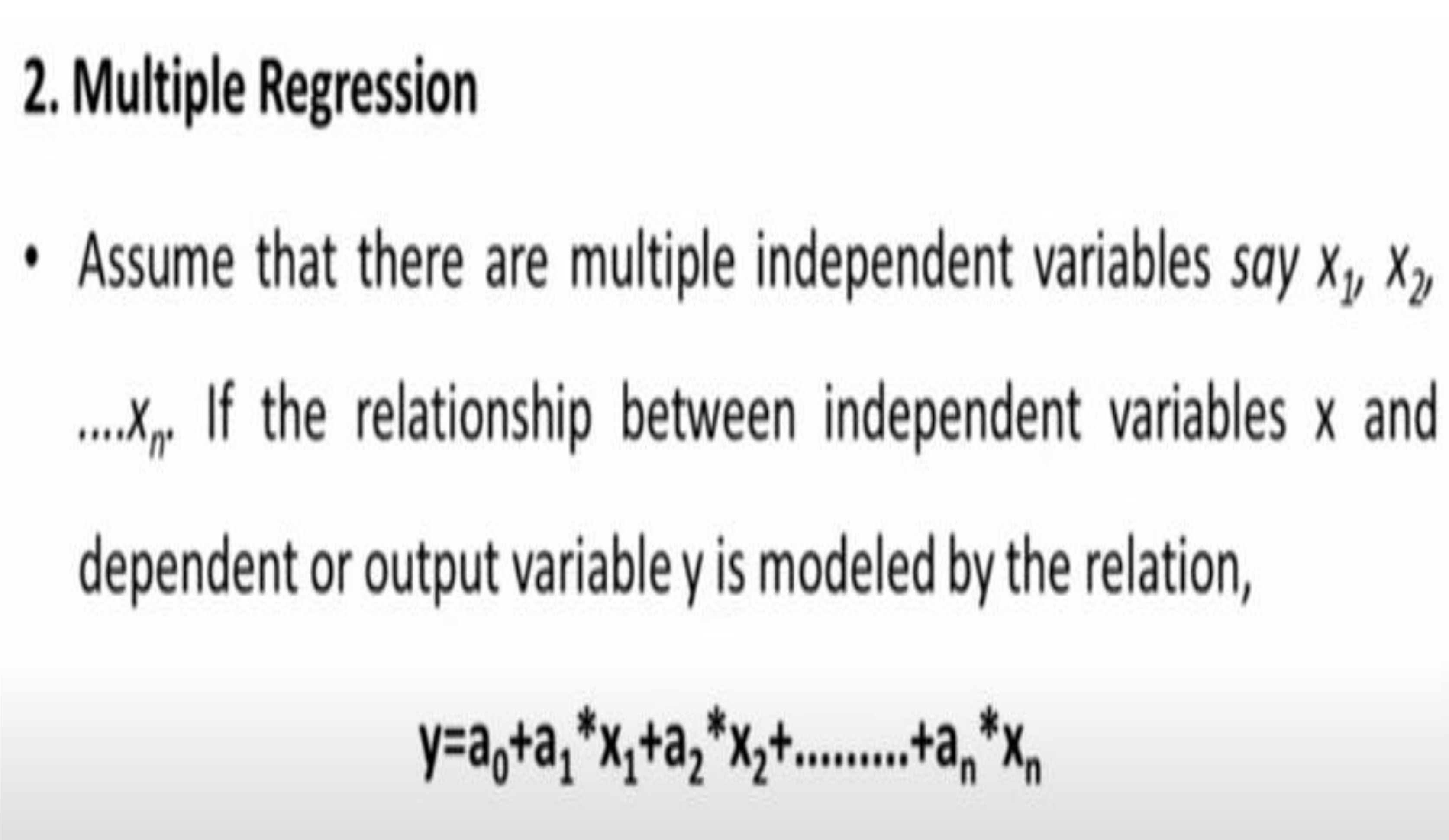
- https://colab.research.google.com/drive/1j57KYPJEkCVCLOOPXdbUX7-SidBlzokG?usp=sharing

With Dataset:

- https://colab.research.google.com/drive/1VidU7gKLSf94BOWV2hF6Z4vQ8crTWKke?usp=sharing

Linear Regression with Multiple Variables - Multiple Regression

Multiple Regression

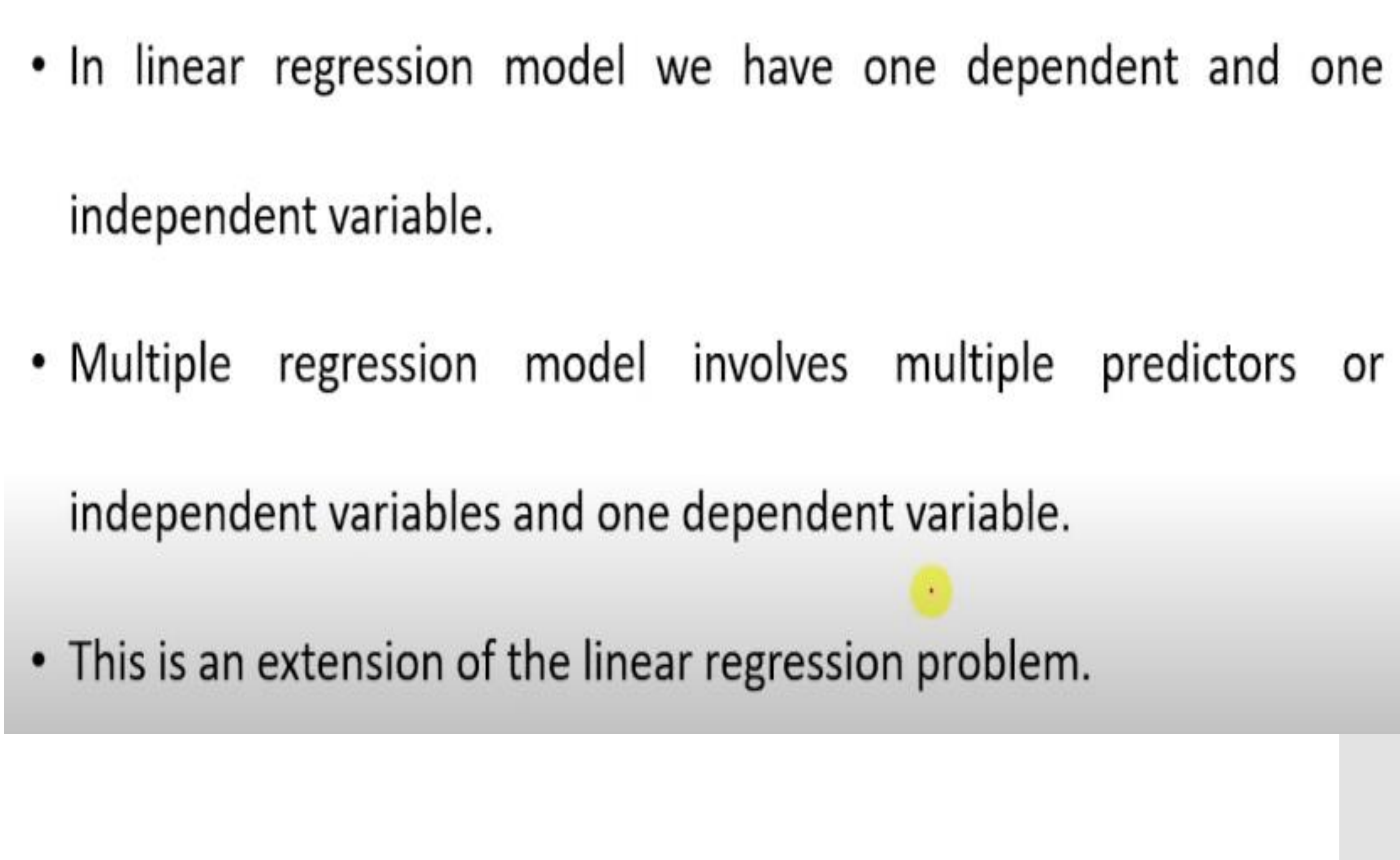


2. Multiple Regression

- Assume that there are multiple independent variables *say* x_1, x_2, ...x_n. If the relationship between independent variables x and dependent or output variable y is modeled by the relation,

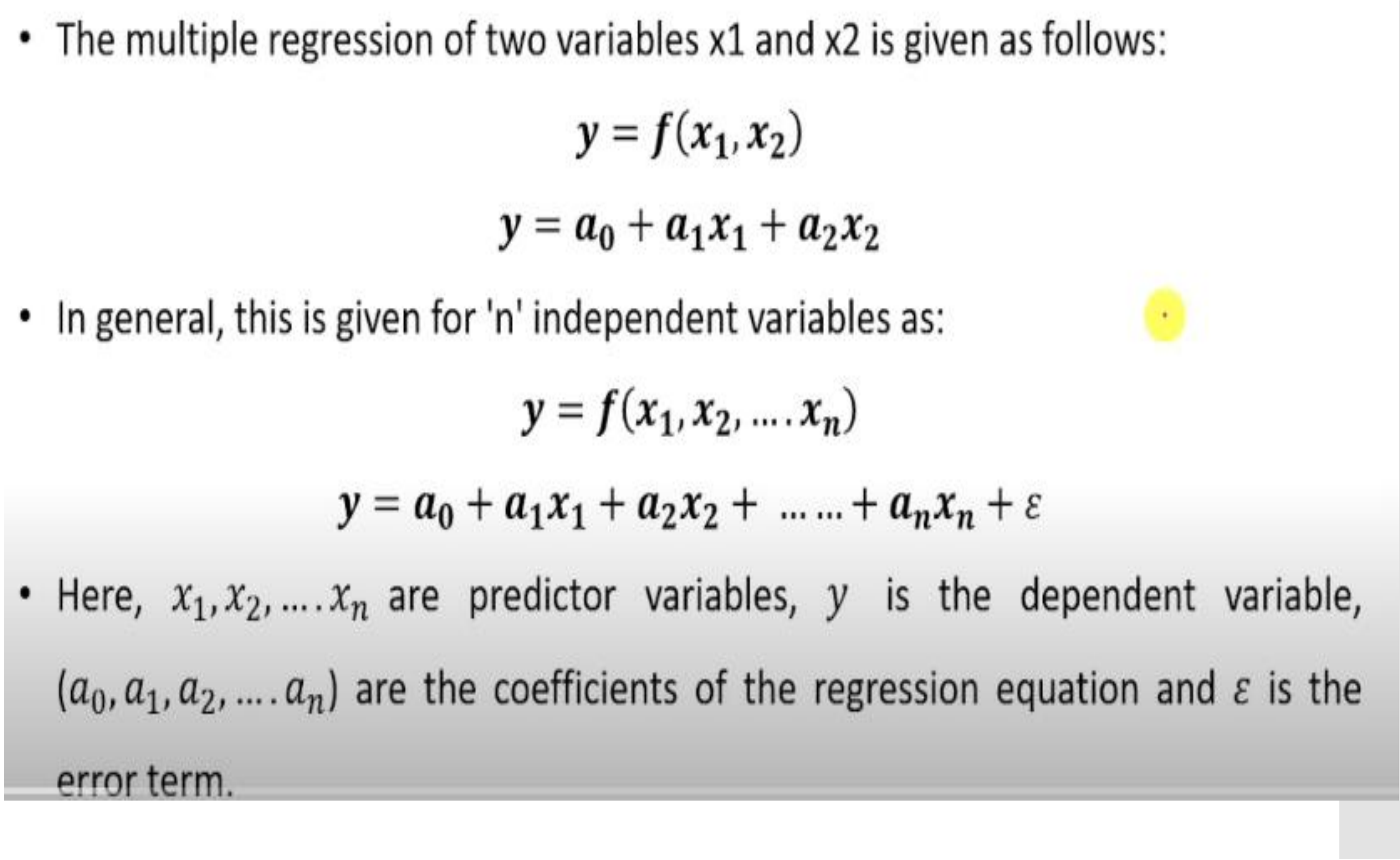
$$y=a_0+a_1{}^*x_1+a_2{}^*x_2+........+a_n{}^*x_n$$

Contd.



- In linear regression model we have one dependent and one independent variable.
- Multiple regression model involves multiple predictors or independent variables and one dependent variable.
- This is an extension of the linear regression problem.

Contd.



- The multiple regression of two variables x1 and x2 is given as follows:

$$y = f(x_1, x_2)$$

$$y = a_0 + a_1x_1 + a_2x_2$$

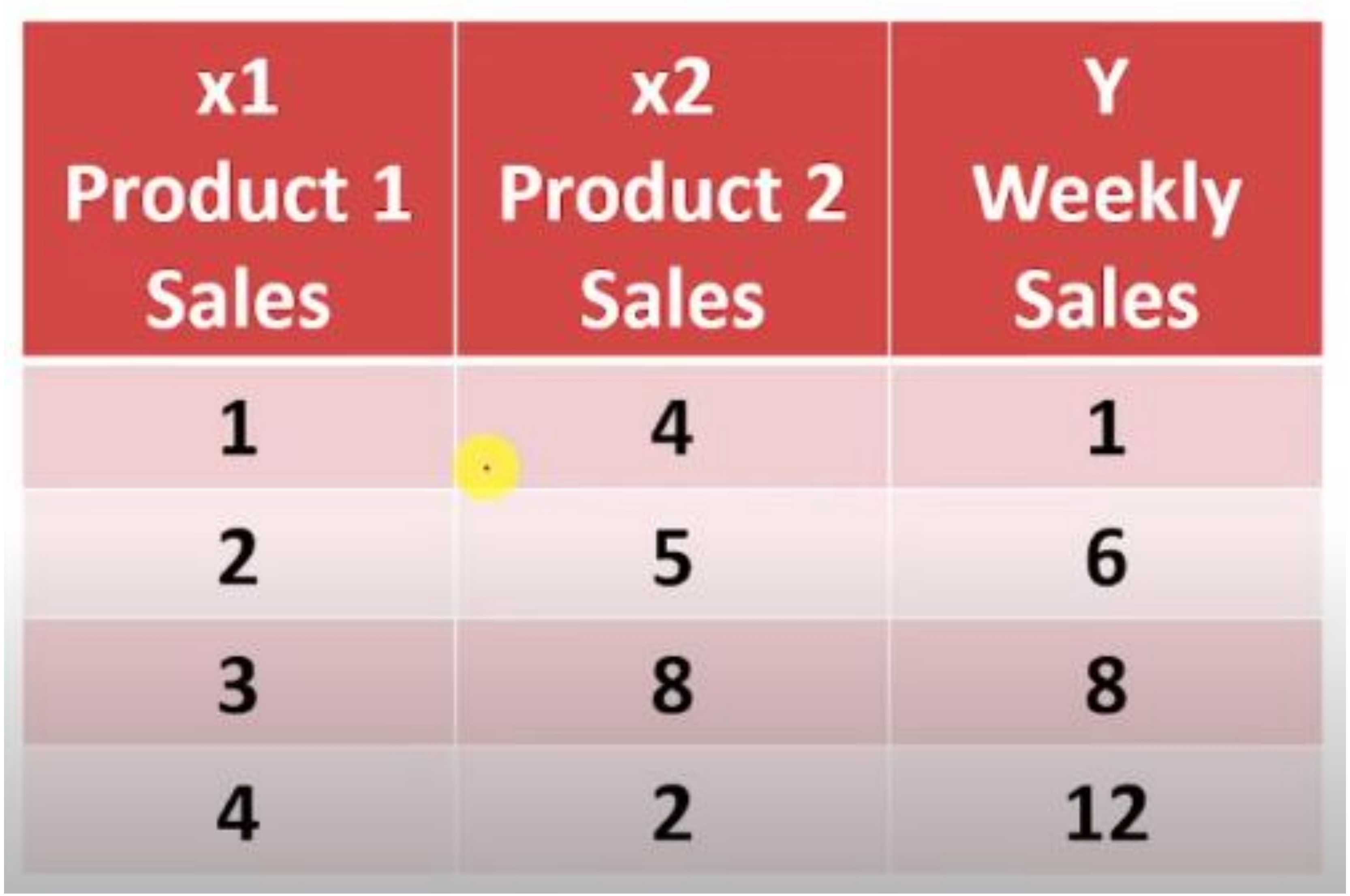
- In general, this is given for 'n' independent variables as:

$$y = f(x_1, x_2, \ldots . x_n)$$

$$y = a_0 + a_1x_1 + a_2x_2 + \ \ldots \ldots + a_nx_n + \varepsilon$$

- Here, $x_1, x_2, \ldots . x_n$ are predictor variables, y is the dependent variable, $(a_0, a_1, a_2, \ldots . a_n)$ are the coefficients of the regression equation and ε is the error term.

Contd.



x1 Product 1 Sales	x2 Product 2 Sales	Y Weekly Sales
1	4	1
2	5	6
3	8	8
4	2	12

Contd.

Size (feet)^2 X1	No. of Bedrooms x2	No. of Floors x3	Age of Home x4	Price ($) y
2104	5	1	45	460
1416	3	2	40	232
1534	3	2	30	315

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_3 + \theta_4 x_4$$

$$y=a_0+a_1{}^*x_1+a_2{}^*x_2+\ldots\ldots\ldots+a_n{}^*x_n$$

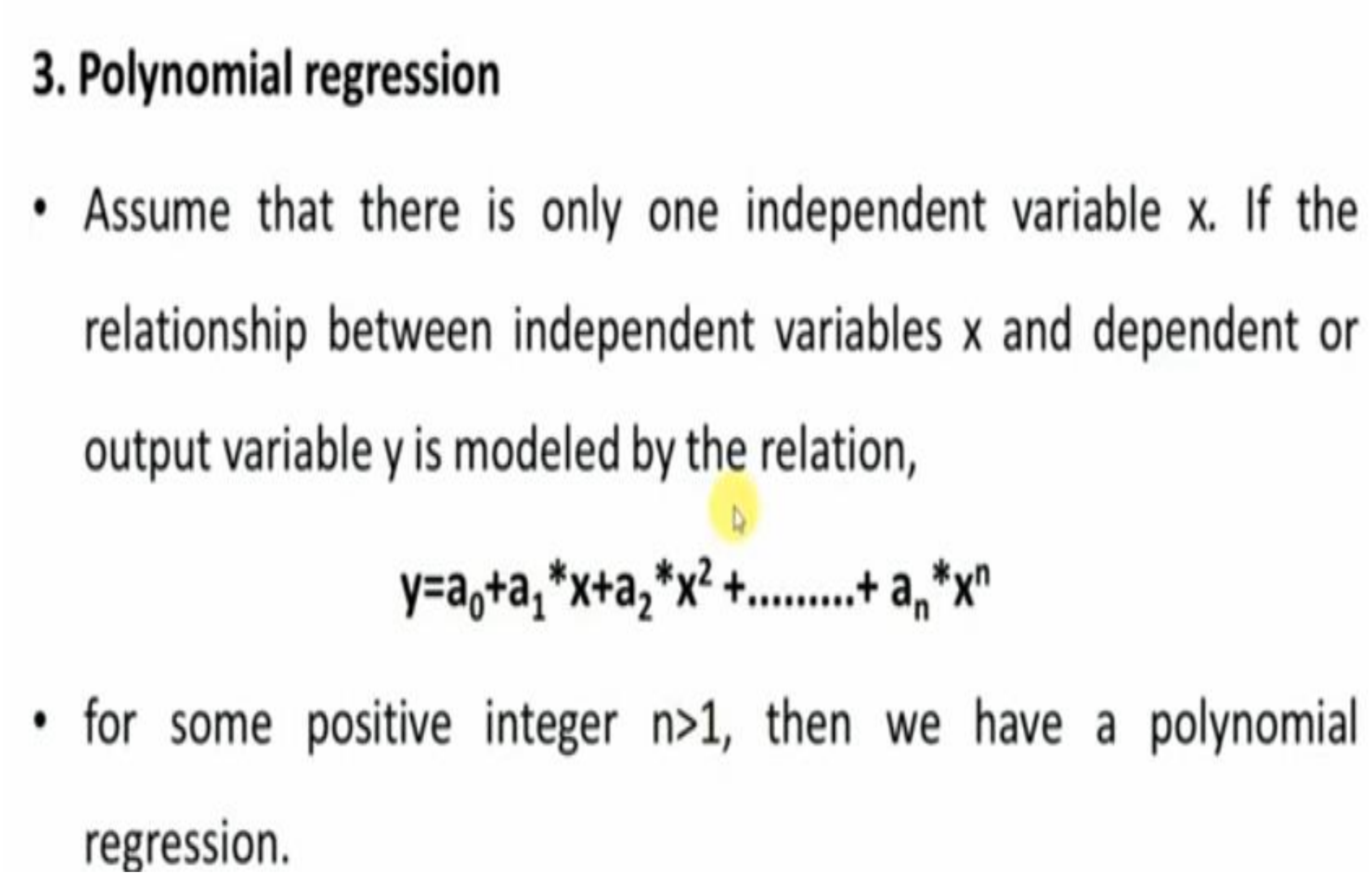
DEMO

Linear Regression with **Multiple** Variables- Multiple Regression

- https://colab.research.google.com/drive/1lc2htHq2sTn5_JqzipHlUbVjbeK4ORAf?usp=sharing

Non-Linear Regression / Polynomial Regression

Polynomial Regression



3. Polynomial regression

- Assume that there is only one independent variable x. If the relationship between independent variables x and dependent or output variable y is modeled by the relation,

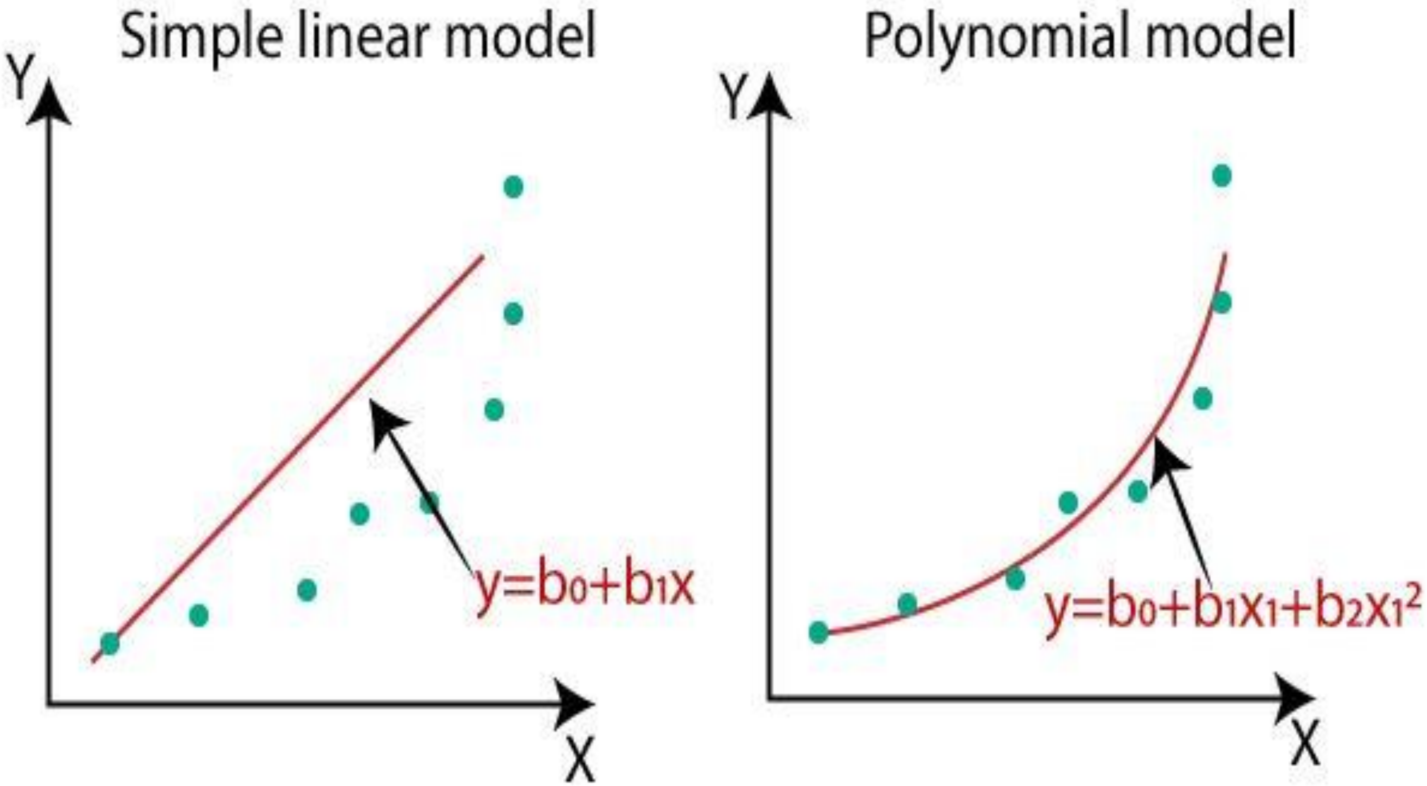
$$y=a_0+a_1*x+a_2*x^2 +.........+ a_n*x^n$$

- for some positive integer n>1, then we have a polynomial regression.

Contd.

- It is also called the special case of Multiple Linear Regression in ML. Because we add some polynomial terms to the Multiple Linear regression equation to convert it into Polynomial Regression.
- Hence, "In Polynomial regression, the original features are converted into Polynomial features of required degree (2,3,..,n) and then modeled using a linear model."

Need for Polynomial Regression



Contd.

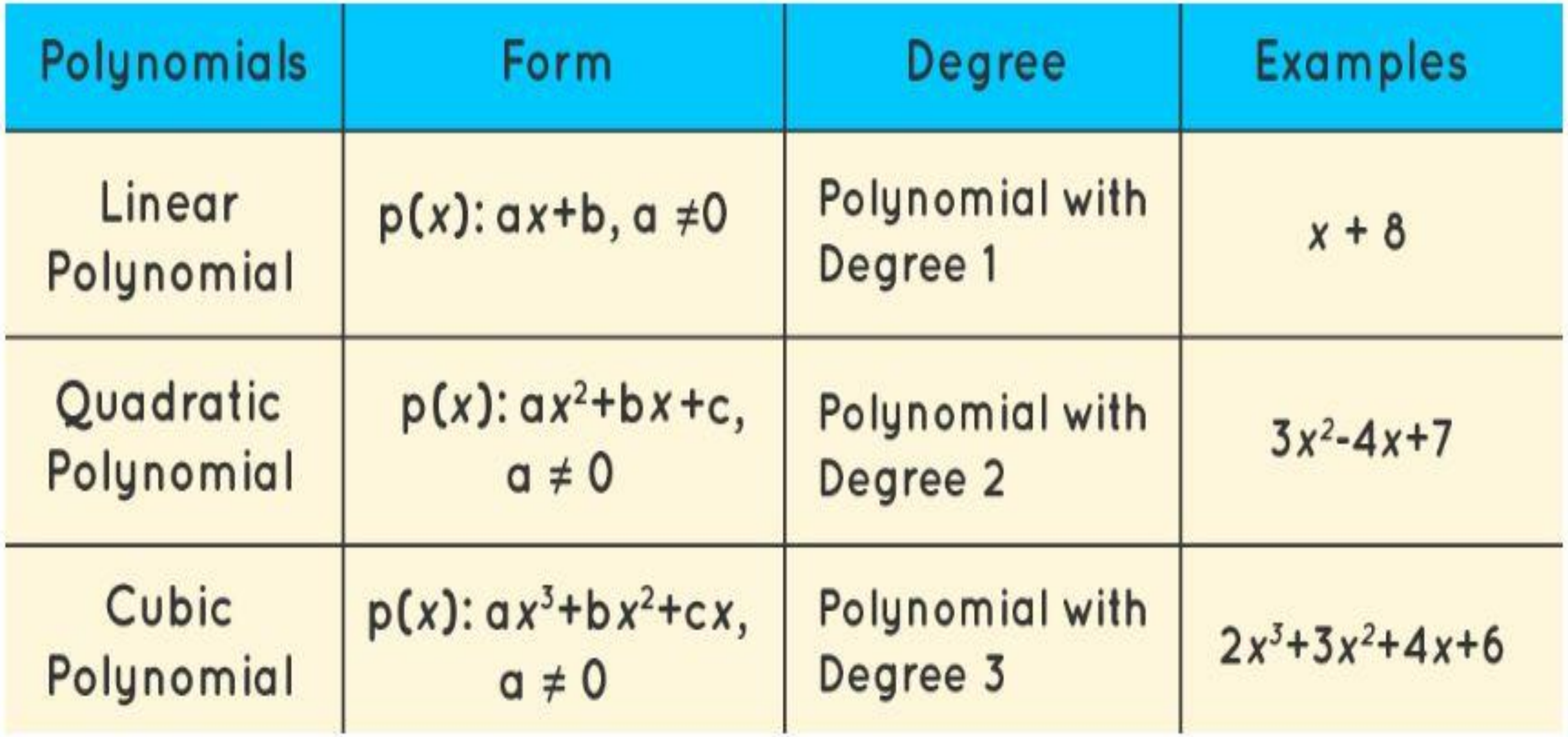
- If we apply a linear model on a **linear dataset**, then it provides us a good result as we have seen in Simple Linear Regression, but if we apply the same model without any modification on a **non-linear dataset**, then it will produce a drastic output. Due to which loss function will increase, the error rate will be high, and accuracy will be decreased.
- So for such cases, **where data points are arranged in a non-linear fashion, we need the Polynomial Regression model**. We can understand it in a better way using the below comparison diagram of the linear dataset and non-linear dataset.

Types of Polynomial Regression

A quadratic equation is a general term for a second-degree polynomial equation. This degree, on the other hand, can go up to nth values. Polynomial regression can so be categorized as follows:

1. Linear – if degree as 1
2. Quadratic – if degree as 2
3. Cubic – if degree as 3 and goes on, on the basis of degree.

Types of Polynomial Regression



Polynomials	Form	Degree	Examples
Linear Polynomial	$p(x): ax+b, a \neq 0$	Polynomial with Degree 1	$x + 8$
Quadratic Polynomial	$p(x): ax^2+bx+c,$ $a \neq 0$	Polynomial with Degree 2	$3x^2-4x+7$
Cubic Polynomial	$p(x): ax^3+bx^2+cx,$ $a \neq 0$	Polynomial with Degree 3	$2x^3+3x^2+4x+6$

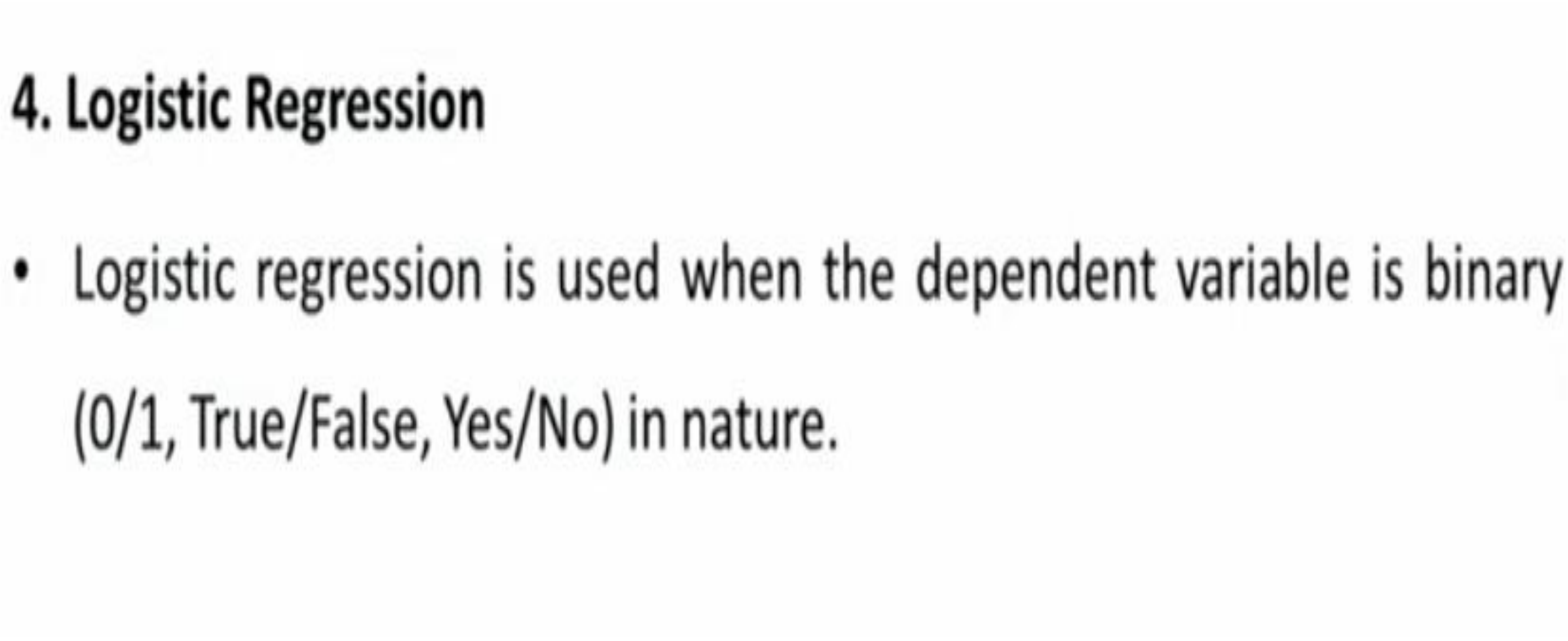
DEMO

Polynomial Regression

- https://colab.research.google.com/drive/1ZJkQLJ4Da9_q8ngcXVCMLgVGxJTjtstc?usp=sharing

Logistic Regression

Logistic Regression



4. Logistic Regression

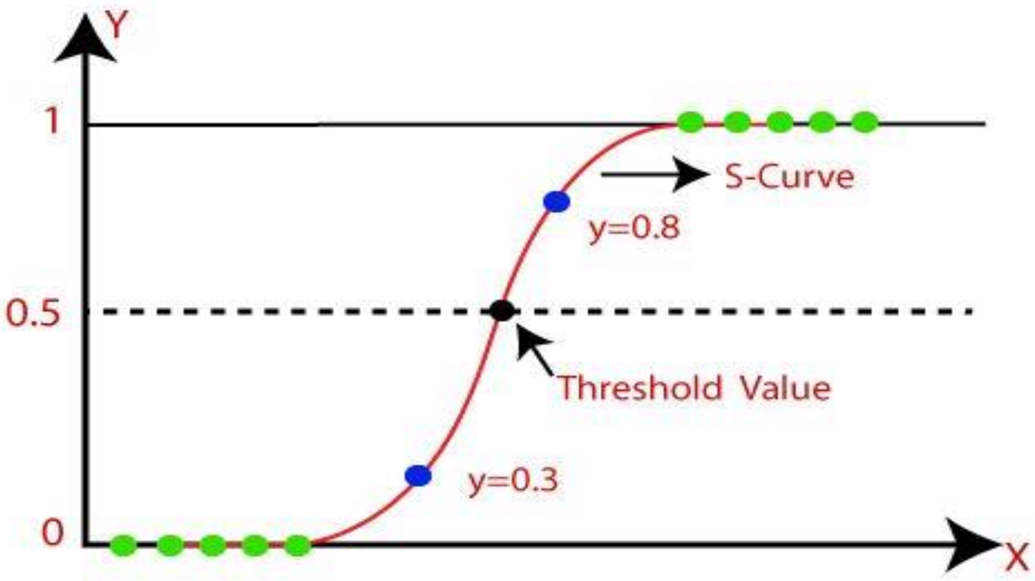
- Logistic regression is used when the dependent variable is binary (0/1, True/False, Yes/No) in nature.

Logistic Regression

- Logistic regression is one of the most popular Machine Learning algorithms, which comes under the supervised Learning technique.
- It is used for predicting the categorical dependent variable using a given set of independent variables.
- Logistic regression predicts the output of a categorical dependent variable. Therefore the outcome must be a categorical or discrete value.
- It can be either Yes or No, 0 or 1, true or False, etc. but instead of giving the exact value as 0 and 1, **it gives the probabilistic values which lie between 0 and 1**.

Logistic Regression

- In Logistic regression, instead of fitting a regression line, we fit an "S" shaped logistic function, which predicts two maximum values (0 or 1)
- The curve from the logistic function indicates the likelihood of something such as whether the cells are cancerous or not, a mouse is obese or not based on its weight, etc



Logistic Function (Sigmoid Function)

- The sigmoid function is a mathematical function used to map the predicted values to probabilities.
- It maps any real value into another value within a range of 0 and 1.
- The value of the logistic regression must be between 0 and 1, which cannot go beyond this limit, so it forms a curve like the "S" form. The S-form curve is called the Sigmoid function or the logistic function.
- In logistic regression, we use the concept of the threshold value, which defines the probability of either 0 or 1. Such as values above the threshold value tends to 1, and a value below the threshold values tends to 0.

Type of Logistic Regression

On the basis of the categories, Logistic Regression can be classified into three types:

1. **Binomial:** In binomial Logistic regression, there can be only two possible types of the dependent variables, such as 0 or 1, Pass or Fail, etc.
2. **Multinomial:** In multinomial Logistic regression, there can be 3 or more possible unordered types of the dependent variable, such as "cat", "dogs", or "sheep“.
3. **Ordinal:** In ordinal Logistic regression, there can be 3 or more possible ordered types of dependent variables, such as "low", "Medium", or "High".

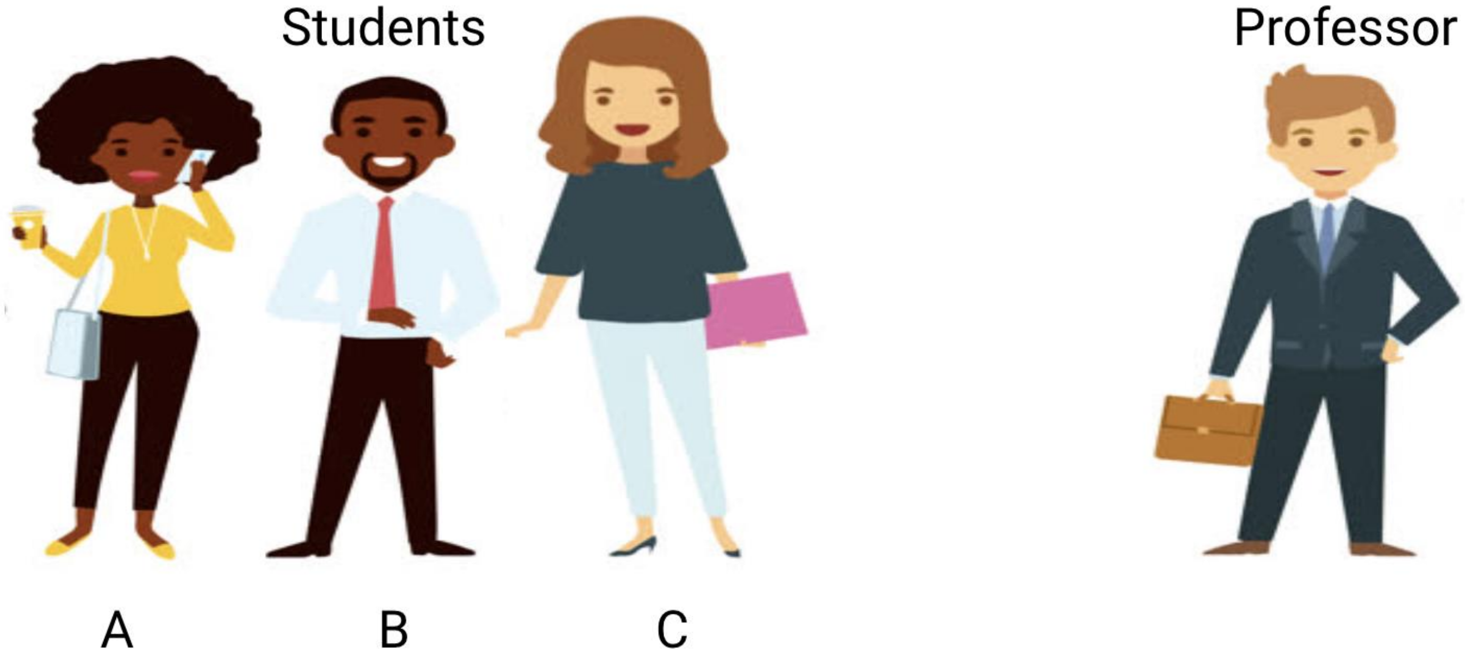
DEMO

Logistic Regression

- https://colab.research.google.com/drive/1Z504ZYibatAsE2_WlHzobJyboSiy4lUk?usp=sharing

Overfitting and Underfitting
&
Bias and Variance

Underfitting and Overfitting in Machine Learning with Example



A

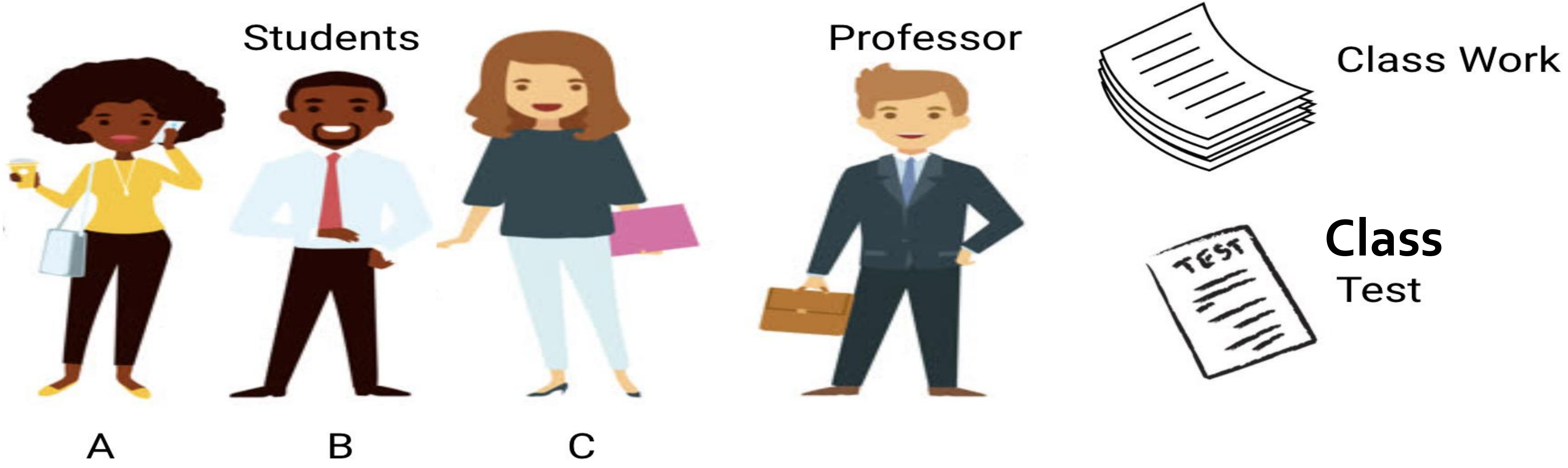
- Hobby = chating
- Not interested in class
- Doesn’t pay much attention to professor

B

- Hobby = to be best in class.
- Mugs up everything professor say
- Too much attention to the class work.

C

- Hobby = learning new things
- Eager to learn concepts.
- Pays attention to class and learns the idea behind solving a problem.



- The professor first delivers lectures and teaches the students about the problems and how to solve them. At the end of the day, the professor simply takes a quiz based on what he taught in the class.

Guessing: ~50%

A

Mr. know it all
~98%

B

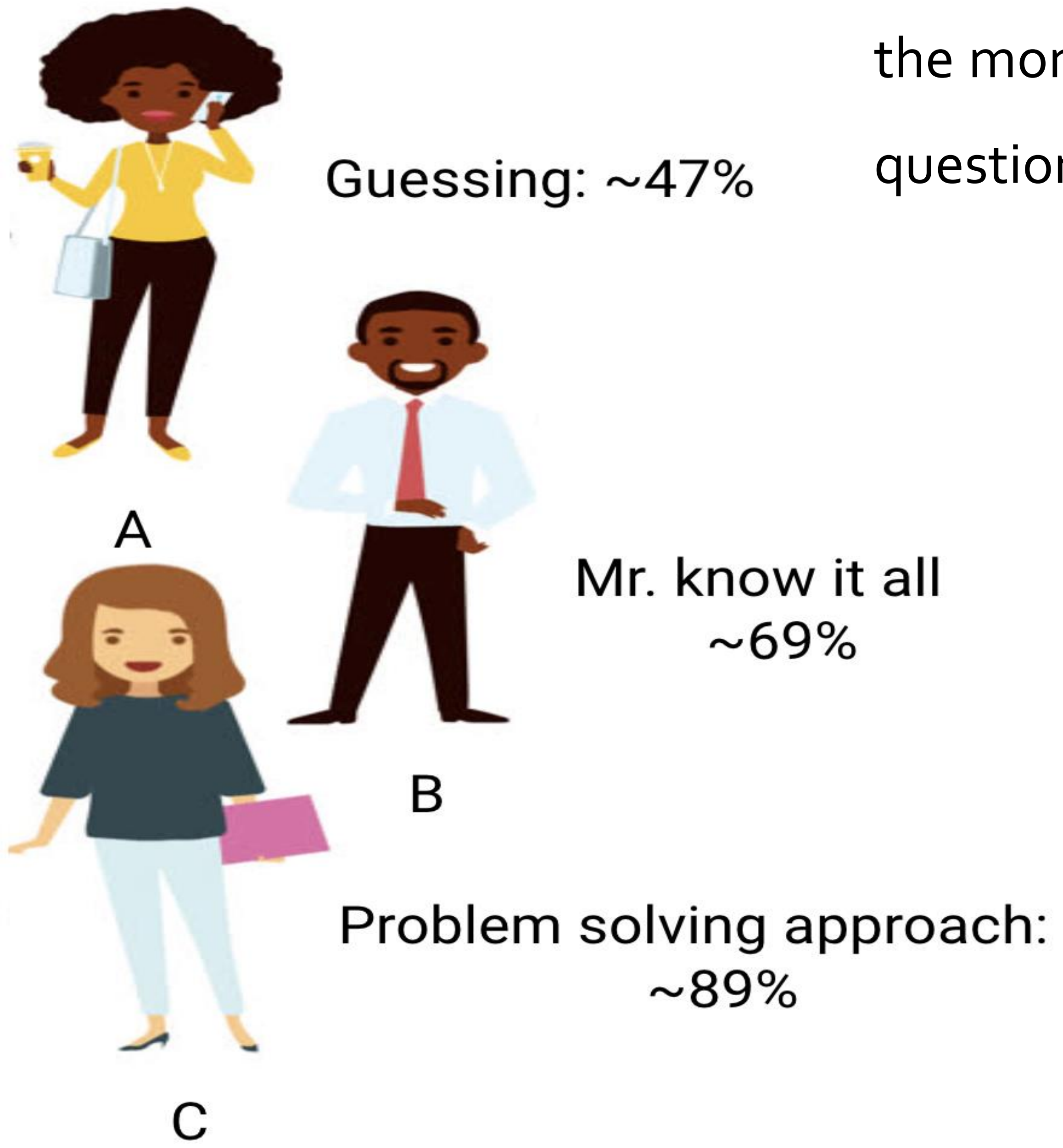
Problem solving approach:
~92%

C

Quiz based on
class work

Professor

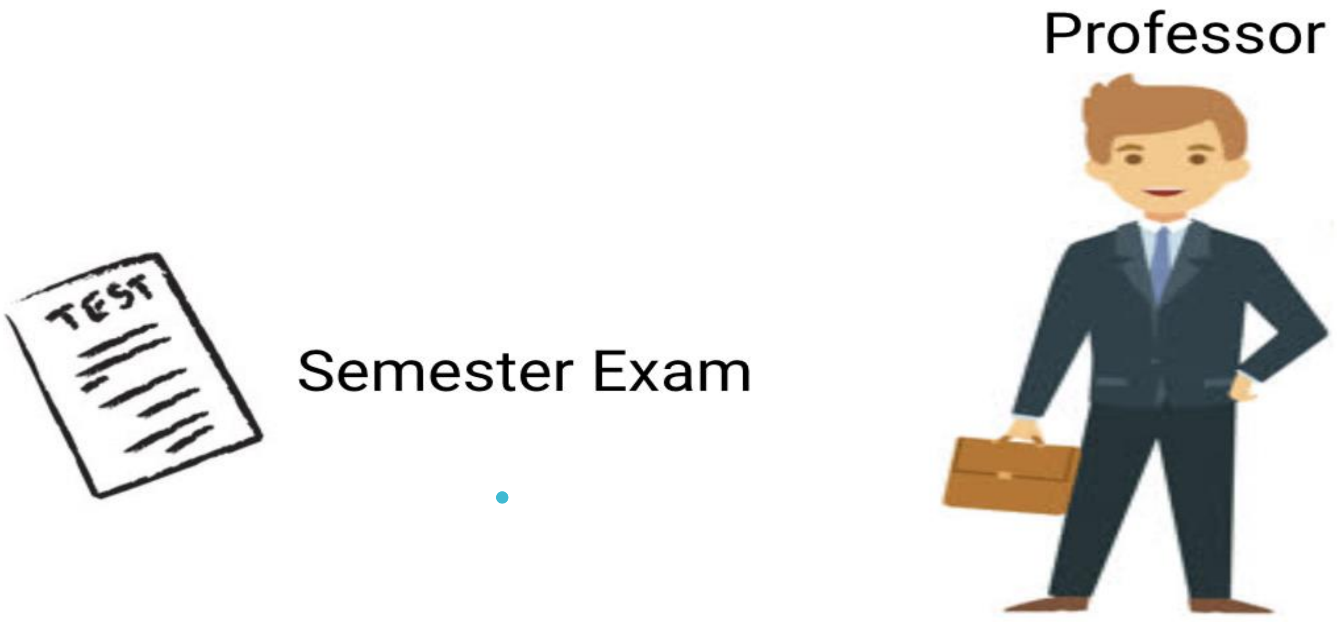
- Now here's the twist. Let's also look at what happens during the monthly test, when students have to face new unknown questions which are not taught in the class by the teacher.



Guessing: ~47%

Mr. know it all
~69%

Problem solving approach:
~89%



Contd.

- In the case of student A, things did not change much and he still randomly answers questions correctly ~50% of the time.
- In the case of Student B, his score dropped significantly. Can you guess why? This is because he always memorized the problems that were taught in the class but this monthly test contained questions which he has never seen before. Therefore, his performance went down significantly
- In the case of Student C, the score remained more or less the same. This is because she focused on learning the problem-solving approach and therefore was able to apply the concepts she learned to solve the unknown questions

A

Not interested in learning

Class test ~50%

Semester Exam ~47%

B

Memorizing the lessons

Class test ~98%

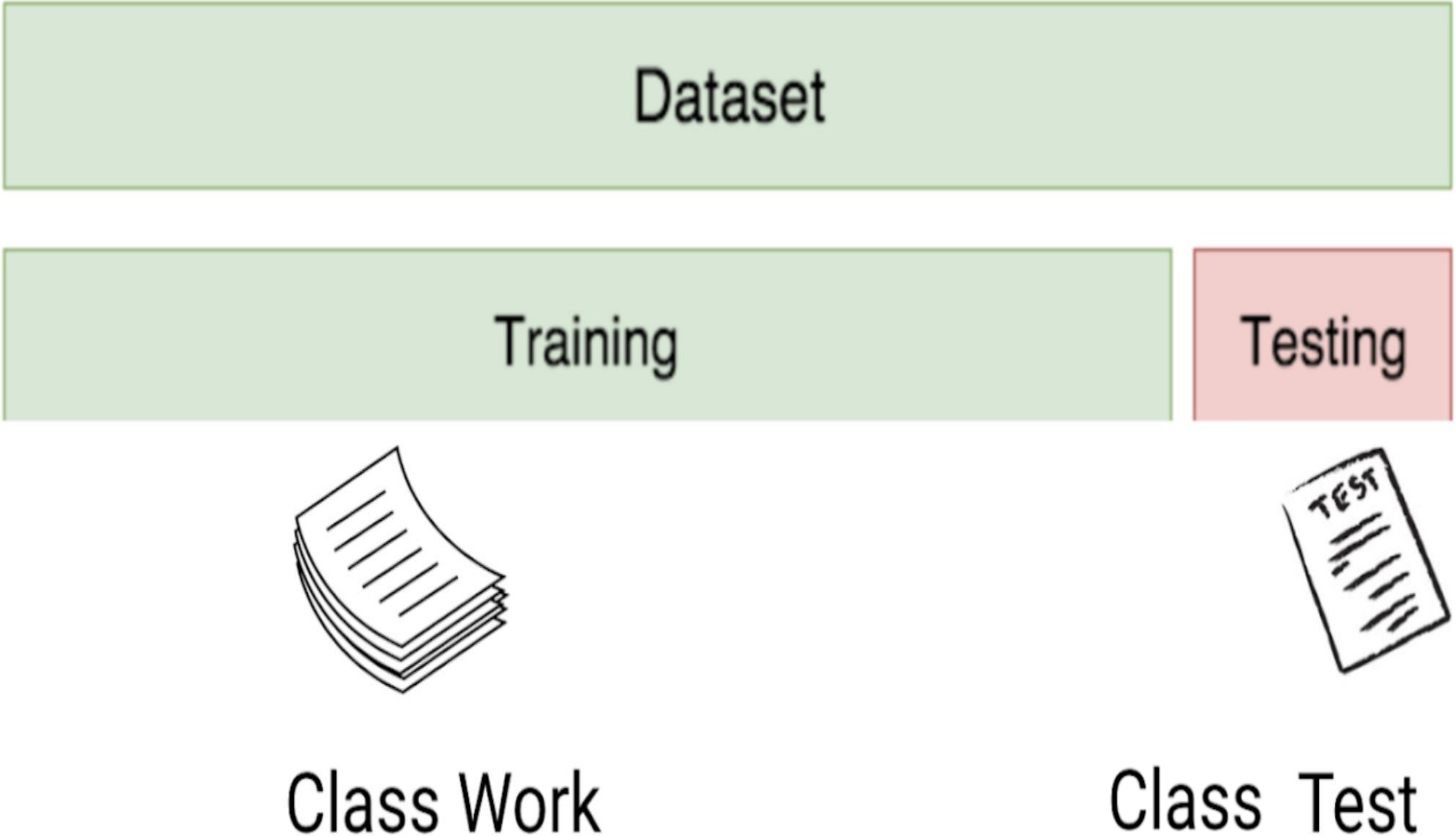
Semester Exam ~69%

C

Conceptual Learning

Class test ~92%

Semester Exam ~89%



Dataset

Training

Testing

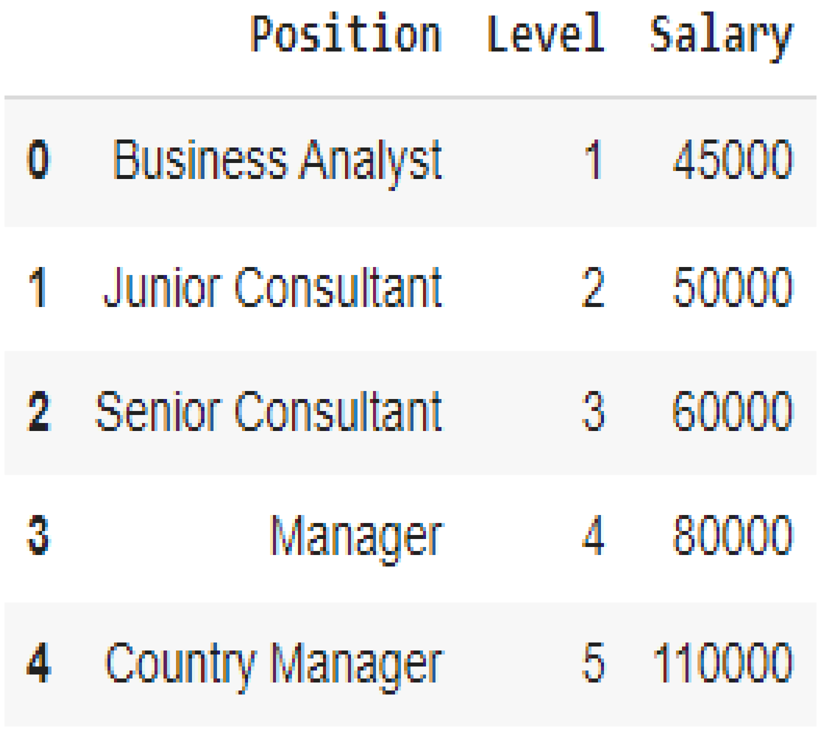
Class Work

Class Test

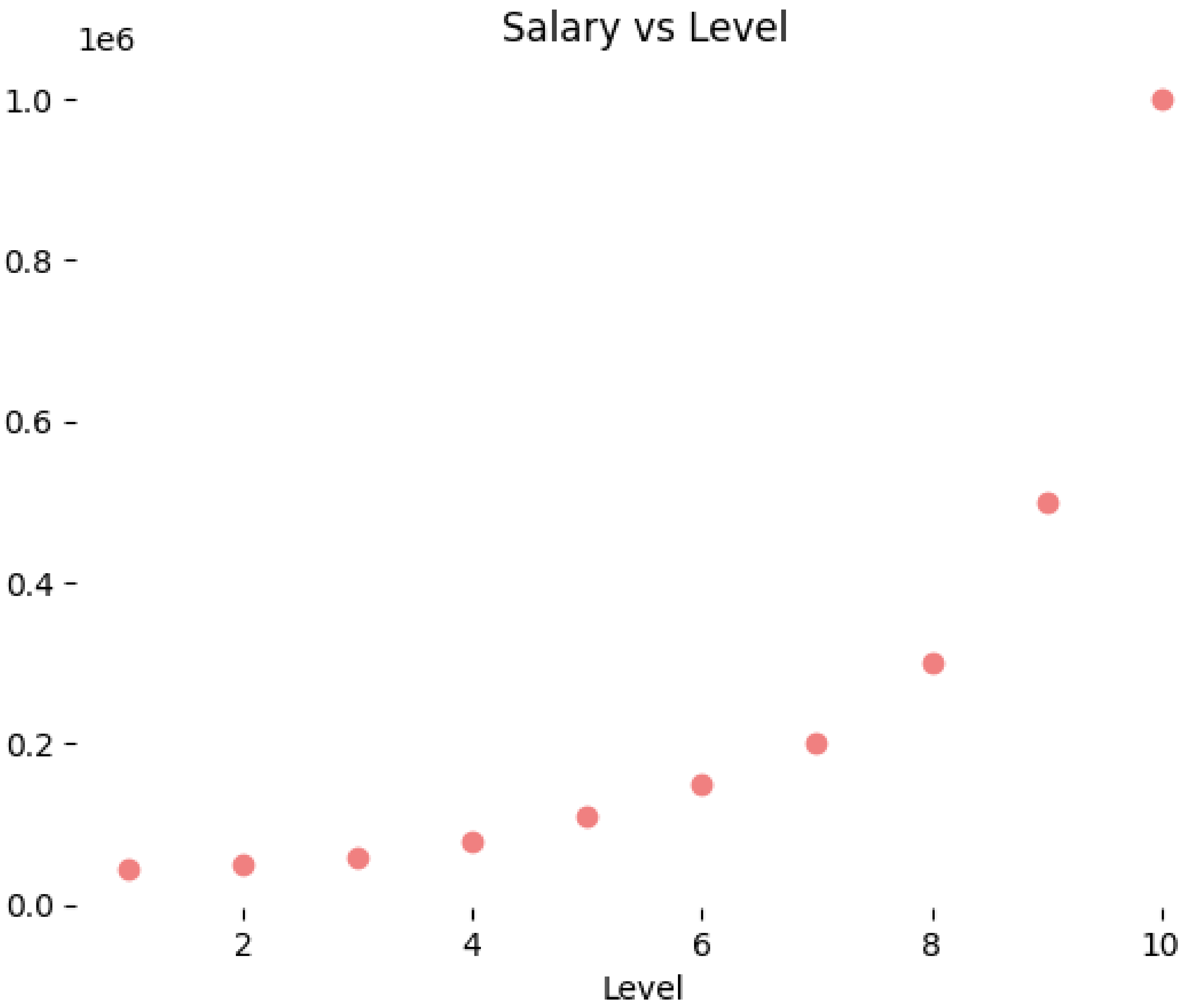
A	B	C
Not interested in learning	Memorizing the lessons	Conceptual Learning
Class test ~50%	Class test ~98%	Class test ~92%
Test ~47%	Test ~69%	Test ~89%
Under-fit/ biased learning	**Over-fit/ Memorizing**	**Best-fit**

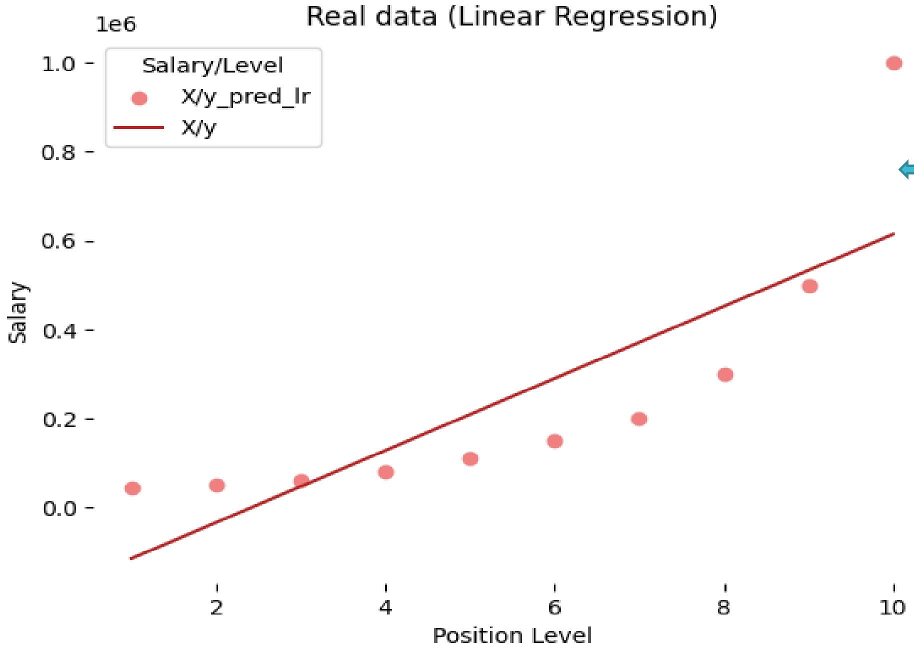
Contd.

- First, the classwork and class test resemble the training data and the prediction over the training data itself respectively.
- On the other hand, the semester test represents the test set from our data which we keep aside before we train our model (or unseen data in a real-world machine learning project).



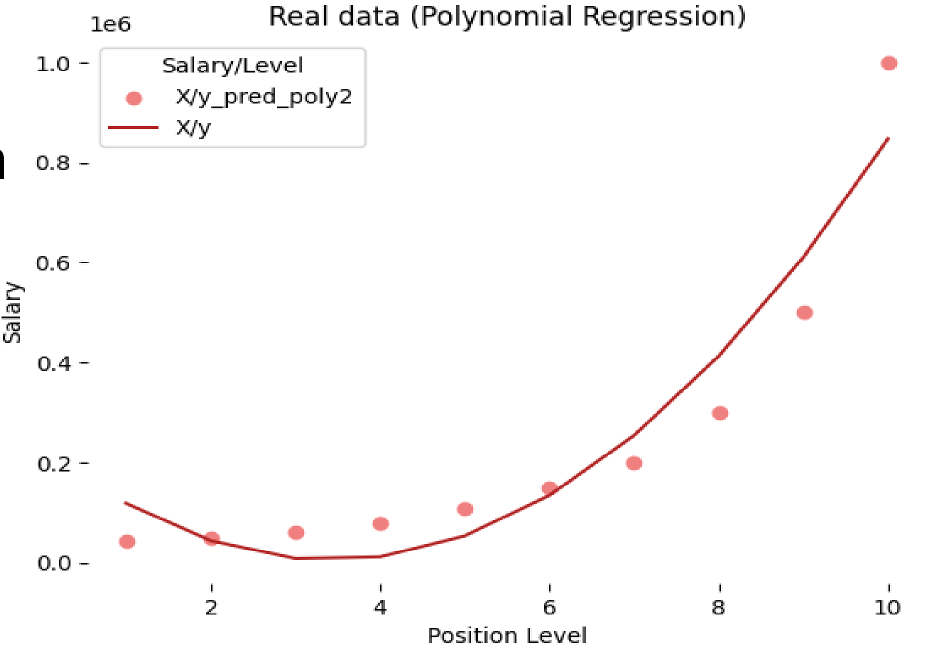
	Position	Level	Salary
0	Business Analyst	1	45000
1	Junior Consultant	2	50000
2	Senior Consultant	3	60000
3	Manager	4	80000
4	Country Manager	5	110000

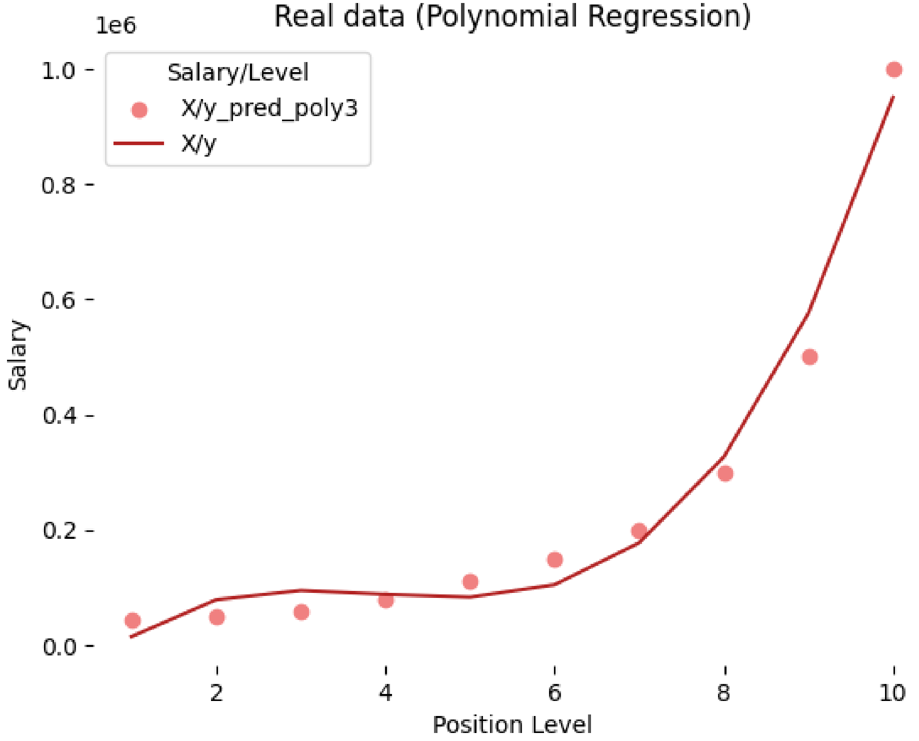




Linear regression

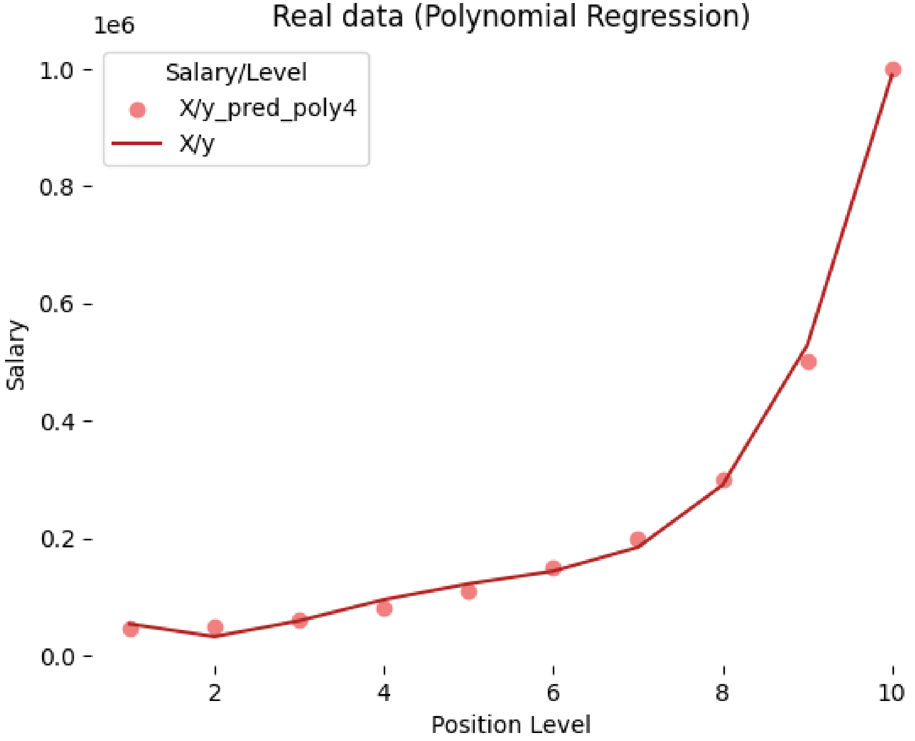
Polynomial Regression
With Degree 2





Polynomial Regression
With Degree 3

Polynomial Regression
With Degree 4



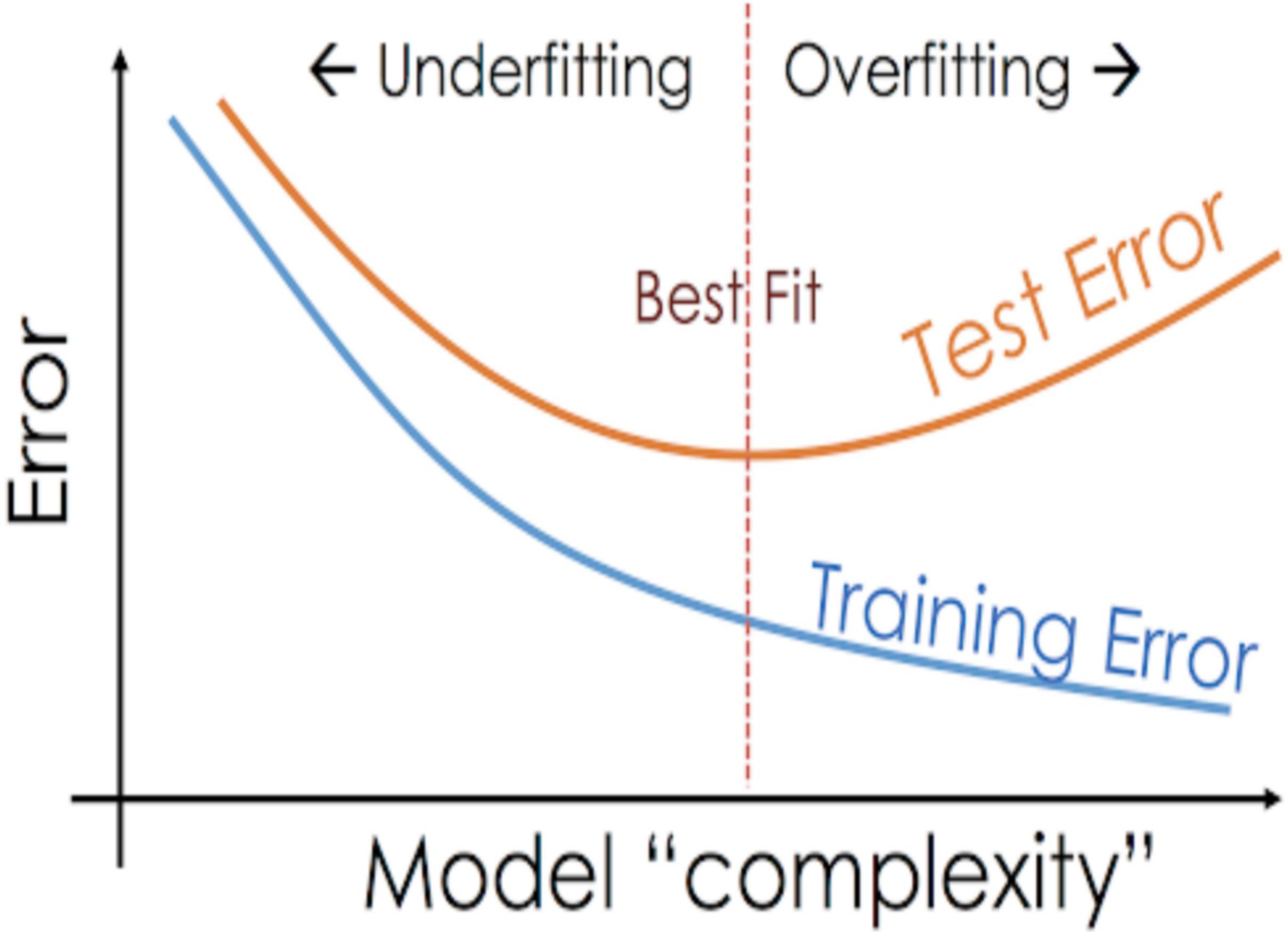
THE BEST WAY TO
EXPLAIN OVERFITTING

DEMO

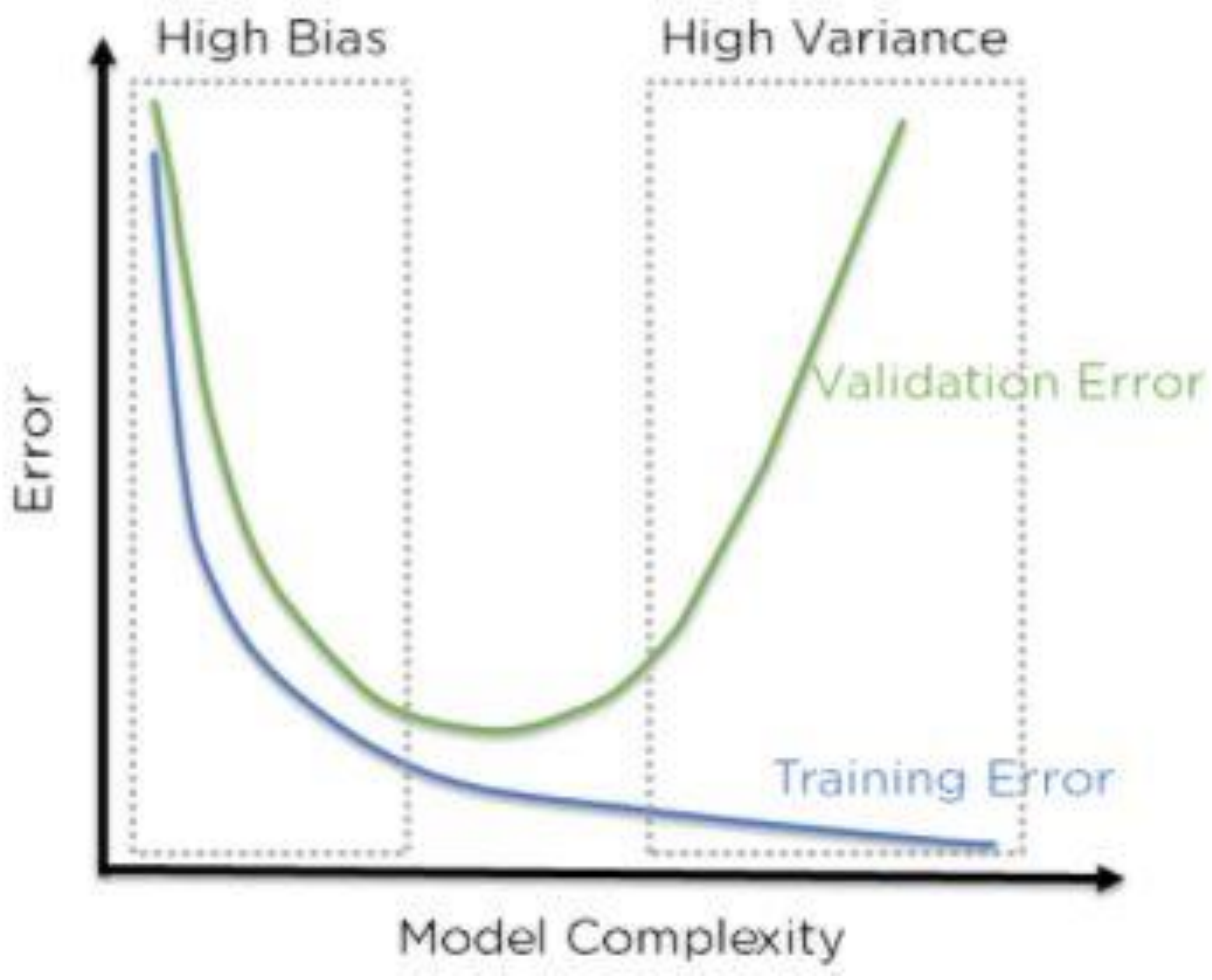
Underfitting and Overfitting in Linear Regression and Polynomial Regression

- https://colab.research.google.com/drive/1ZJkQLJ4Da9_q8ngcXVCMLgVGxJTjtstc?usp=sharing

Contd.


← Underfitting
Overfitting →
Best Fit
Test Error
Training Error
Error
Model "complexity"

What are the Bias and Variance?



Bias

- Bias comes out when the algorithm has limited flexibility to learn from the dataset.
- These models pay little attention to the training data and oversimplify the model, so the validation or prediction errors and training errors follow similar trends.
- Such models always lead to high errors in the training and test data. High bias causes under-adjustment in our model.

Variance

- The variance defines the sensitivity of the algorithm to specific data sets.
- A high-variance model pays close attention to the training data and does not generalize, so the validation or prediction errors are far from each other.
- Such models usually perform very well on the training data but have a high error rate on the test data. High deviation causes an overshoot in our model.

Definition : Underfitting and Overfitting

- This situation where any given **model is performing too well on the training data** but the **performance drops significantly over the test set** is called **an overfitting model**.
- On the other hand, if the **model is performing poorly over the test and the train set**, then we call that **an underfitting model**.

What is Underfitting?

- Underfitting occurs when a model is too simplistic to grasp the underlying patterns in the data. It lacks the complexity needed to adequately represent the relationships present, resulting in poor performance on both the training and new data.

Reasons for Underfitting

- Too few features miss important details.
- Using a simple model for a complex problem.
- Excessive regularization limits model flexibility.

What is Overfitting?

- Overfitting happens when a machine learning model becomes overly intricate, essentially memorizing the training data. While this might lead to high accuracy on the training set, the model may struggle with new, unseen data due to its excessive focus on specific details.

Reasons for Overfitting

- Too many features confuse the model.
- Using a complex model for a simple problem.
- Not enough regularization.

Addressing Overfitting

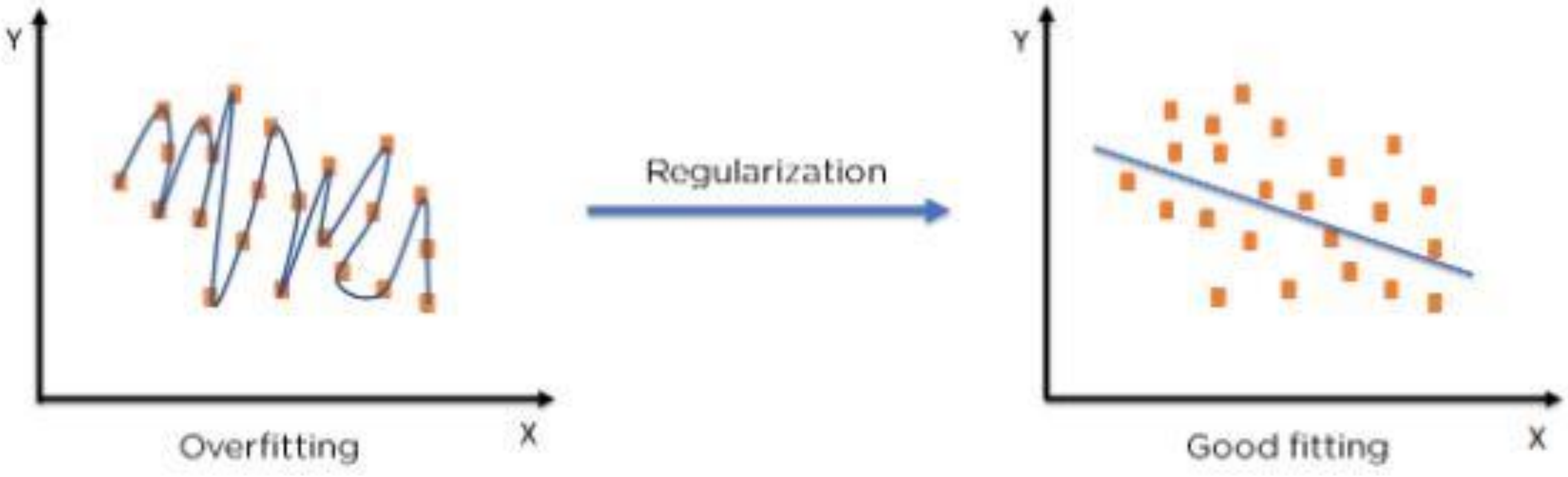
There are two options to avoid the overfitting:

- **Reduce the number of features**: Mostly overfitting is observed when the number of features is really high and the number of training examples are less. In such cases reducing the number of features can help avoid the issue of overfitting. Reducing the number of features can be done manually, or using model selection algorithms which help decide which features to eliminate. **This also presents a disadvantage as removing features is sometimes equivalent to removing information.**
- **Regularization**: Keep all the features but reduce magnitude/values of parameters θjθj. This technique works well, when there exists a lot features and each contributes a bit to the prediction, i.e. **Regularization works well when there are a lot of slightly useful features.**

Regularization & Hyperparameter Tuning

Regularization

- Regularization is a technique used in machine learning to prevent overfitting and improve the generalization performance of models.
- It refers to techniques used to calibrate machine learning models to minimize the adjusted loss function and avoid overfitting or underfitting.



Advantages of regularization

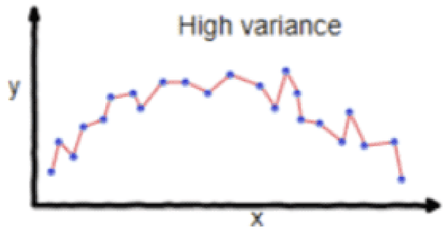
- **Reduces Overfitting:** Regularization techniques help prevent the model from becoming too complex and capturing noise in the training data, leading to better performance on unseen data.
- **Improves Generalization:** By penalizing large coefficients, regularization encourages the model to find simpler patterns that generalize well to new data.
- **Simplifies Models**: Regularization can lead to simpler models with fewer features, which are easier to interpret and understand.
- **Enhances Stability:** Regularized models are often more stable and less sensitive to variations in the training data.
- **Handles High-Dimensional Data:** In cases with many features, regularization helps manage complexity and maintain performance, even when the number of features is large relative to the number of samples.

Techniques of Regularization

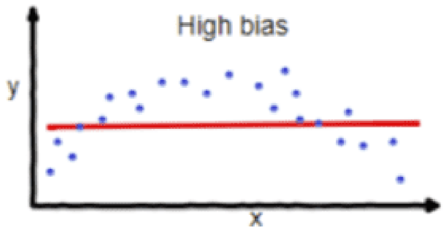
Regularization is a technique used to reduce errors by fitting the function appropriately on the given training set and avoiding overfitting.

The commonly used regularization techniques are :

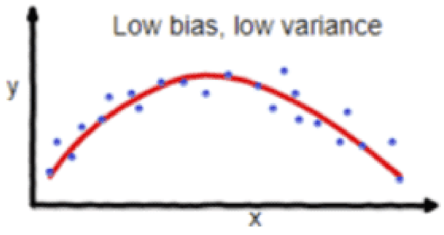
- Lasso Regularization – L1 Regularization
- Ridge Regularization – L2 Regularization
- Elastic Net Regularization – L1 and L2 Regularization



overfitting



underfitting

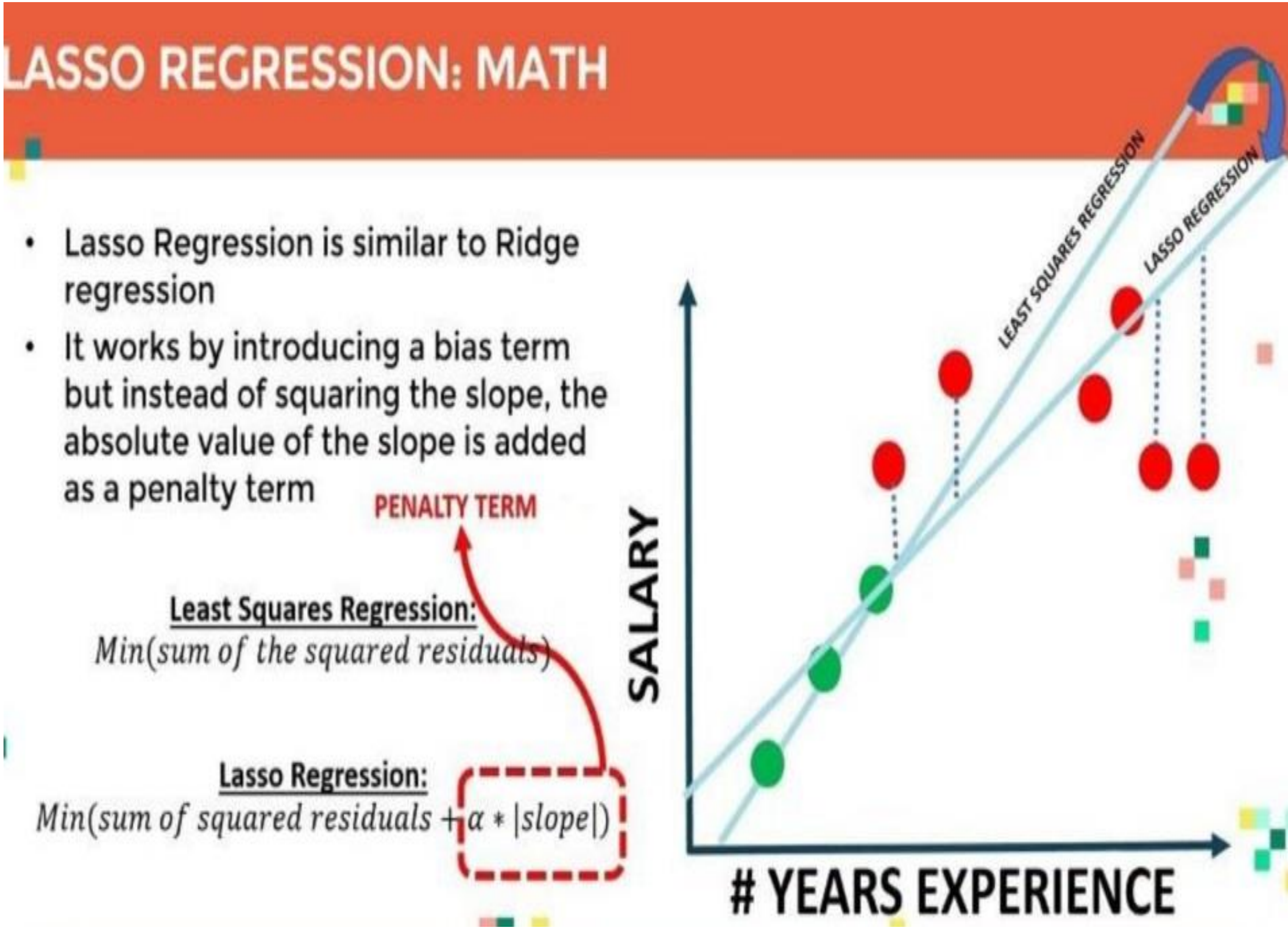


Good balance

Contd.

Regularization Type	Other Name	Key Feature	When to Use
Lasso Regularization	L1 Regularization	Adds the absolute value of coefficients to the loss function	When you want to automatically select features (some coefficients may become exactly zero)
Ridge Regularization	L2 Regularization	Adds the squared value of coefficients to the loss function	When you want to keep all features but shrink their coefficients
Elastic Net Regularization	L1 and L2 Regularization	Combines both L1 and L2 penalties	When you want a balance between feature selection and coefficient shrinkage

Lasso Regularization – L1 Regularization



LASSO REGRESSION: MATH

- Lasso Regression is similar to Ridge regression
- It works by introducing a bias term but instead of squaring the slope, the absolute value of the slope is added as a penalty term

PENALTY TERM

Least Squares Regression:

$$Min(sum\ of\ the\ squared\ residuals)$$

Lasso Regression:

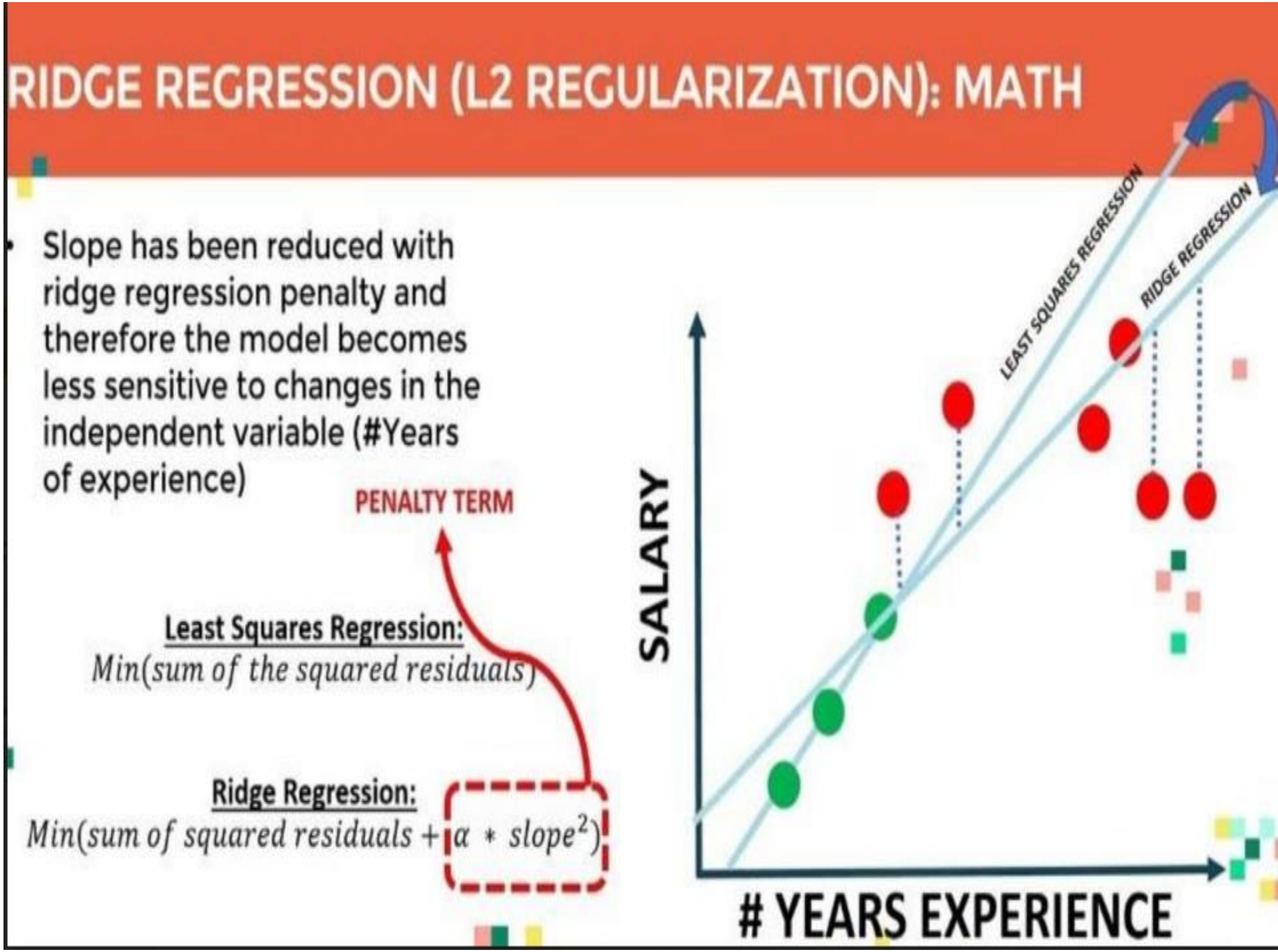
$$Min(sum\ of\ squared\ residuals + \boxed{\alpha * |slope|})$$

Lasso Regularization – L1 Regularization

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

- ***m*** *– Number of Features*
- ***n*** *– Number of Examples*
- ***y_i*** *– Actual Target Value*
- ***y_i(hat)*** *– Predicted Target Value*

Ridge Regularization – L2 Regularization



RIDGE REGRESSION (L2 REGULARIZATION): MATH

- Slope has been reduced with ridge regression penalty and therefore the model becomes less sensitive to changes in the independent variable (#Years of experience)

PENALTY TERM

Least Squares Regression:

Min(sum of the squared residuals)

Ridge Regression:

Min(sum of squared residuals + $\boxed{\alpha * slope^2}$*)*

Ridge Regularization – L2 Regularization

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

- ***m*** *– Number of Features*
- ***n*** *– Number of Examples*
- ***y_i*** *– Actual Target Value*
- ***y_i(hat)*** *– Predicted Target Value*

Elastic Net Regularization – L1 and L2 Regularization

- This model is a combination of L1 as well as L2 regularization. That implies that we add the absolute norm of the weights as well as the squared measure of the weights.
- With the help of an extra hyperparameter that controls the ratio of the L1 and L2 regularization.

$$\mathrm{Cost} = \frac{1}{\mathrm{n}}\sum_{\mathrm{i}=1}^{\mathrm{n}}(\mathrm{y_i} - \hat{\mathrm{y_i}})^2 + \lambda\left((1-\alpha)\sum_{\mathrm{i}=1}^{\mathrm{m}}|\mathrm{w_i}| + \alpha\sum_{\mathrm{i}=1}^{\mathrm{m}}\mathrm{w_i^2}\right)$$

***m** – Number of Features*
***n** – Number of Examples*
***y_i** – Actual Target Value*
***y_i(hat)** – Predicted Target Value*

DEMO

Regularization & Hyperparameter Tuning

Regularization:

- https://colab.research.google.com/drive/1plKBl7P8G7iooiFUddlEhy9vesy7pPhH?usp=sharing

Hyperparameter Tuning:

- https://colab.research.google.com/drive/1Xo_CQD3HashccxjYPP1NxOXegSPt3hPE?usp=sharing

Any Queries..??

Thank you