SUPERVISED LEARNING ALGORITHMS

-MUKESH KUMAR

AGENDA

- **Regression Algorithms**
  – Linear Regression
  – Polynomial Regression
  – Ridge and Lasso Regression
  – Support Vector Regression (SVR)
- **Classification Algorithms**
  – Logistic Regression
  – k-Nearest Neighbors (k-NN)
  – Support Vector Machine (SVM)
  – Decision Trees
  – Random Forest
  – Naive Bayes
  – Gradient Boosting Machines (GBM), XGBoost, LightGBM

WHAT IS A MODEL

Model

- In machine learning, a **model** is a mathematical representation of a system that learns patterns from data.
- The model is created by training an algorithm on a dataset, where the algorithm finds the relationships between the input features (independent variables) and the output (dependent variable) that we want to predict or classify.
- Once trained, the model can make predictions or decisions based on new data.

Why do we need models?

- **Automation**: Models allow for automation of tasks that would be too complex or time-consuming to do manually, such as real-time decision-making in autonomous vehicles.
- **Generalization**: A well-trained model can generalize from the training data to new, unseen data. This means the model can make accurate predictions or classifications even on data it hasn’t encountered before.
- **Optimization**: Models can optimize outcomes, like maximizing revenue, minimizing cost, or improving efficiency. For instance, a model can help determine the optimal price point for a product by predicting sales at different prices.

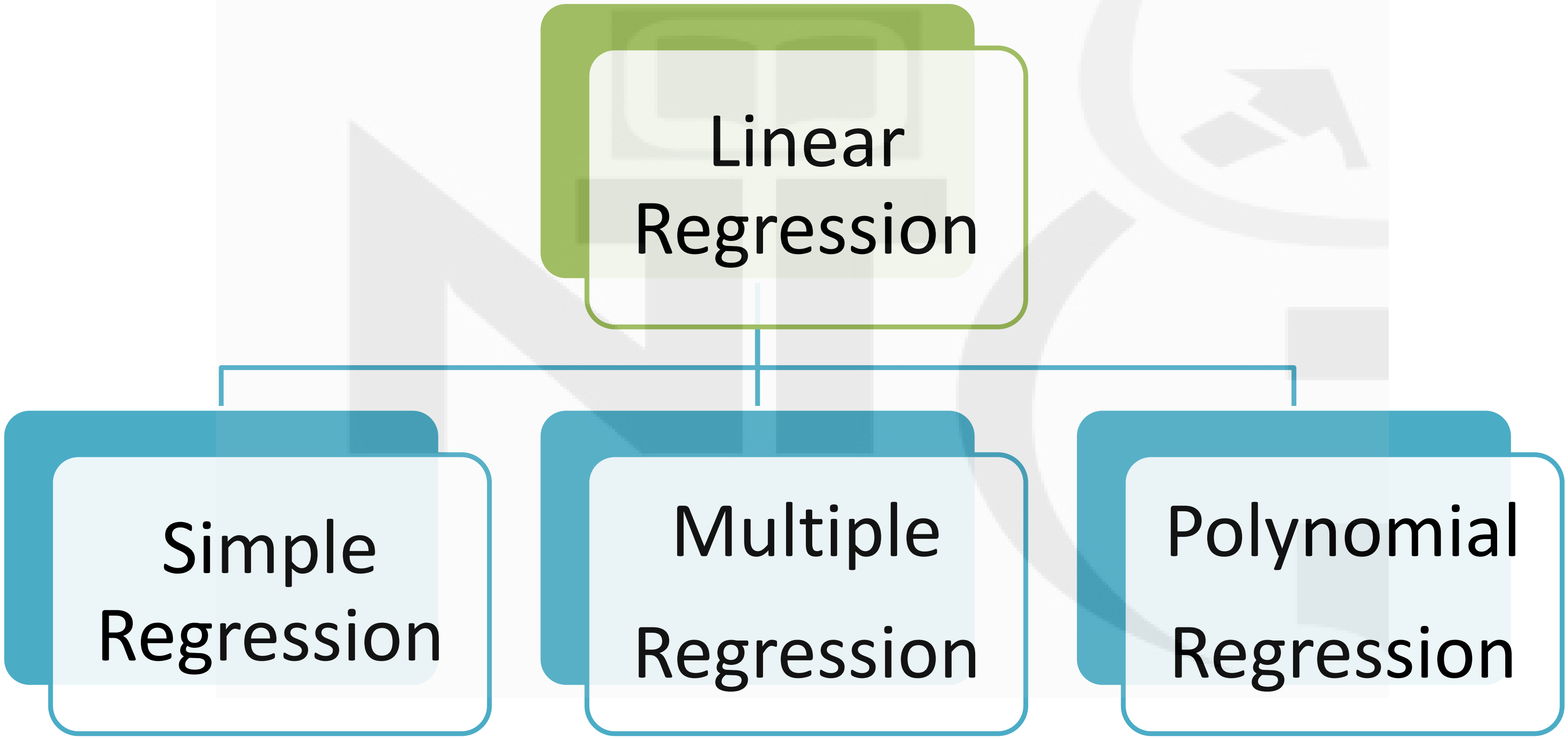
LINEAR REGRESSION

What is Linear Regression

- Linear Regression is a statistical method used to model the relationship between a dependent variable (label) and one or more independent variables (features).

- The main goal of linear regression is to predict the value of the dependent variable based on the values of the independent variables.

LINEAR REGRESSION TYPES


Linear Regression
Simple Regression
Multiple Regression
Polynomial Regression

SIMPLE LINEAR REGRESSION

Simple Linear Regression

- Simple linear regression is the most basic form of linear regression that models the relationship between one independent variable (feature) and one dependent variable (label).

Relation between two variables

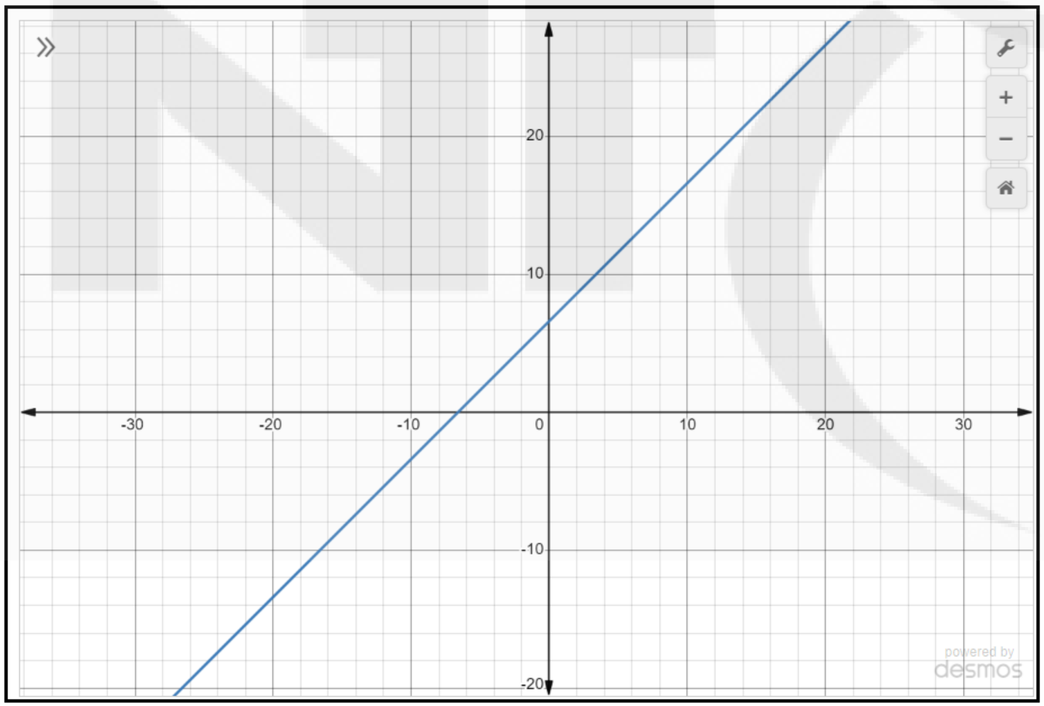
- In a **linear relationship**, two variables, say x and y, have a relationship that can be described by a straight line when plotted on a graph. The general form of a linear equation is:

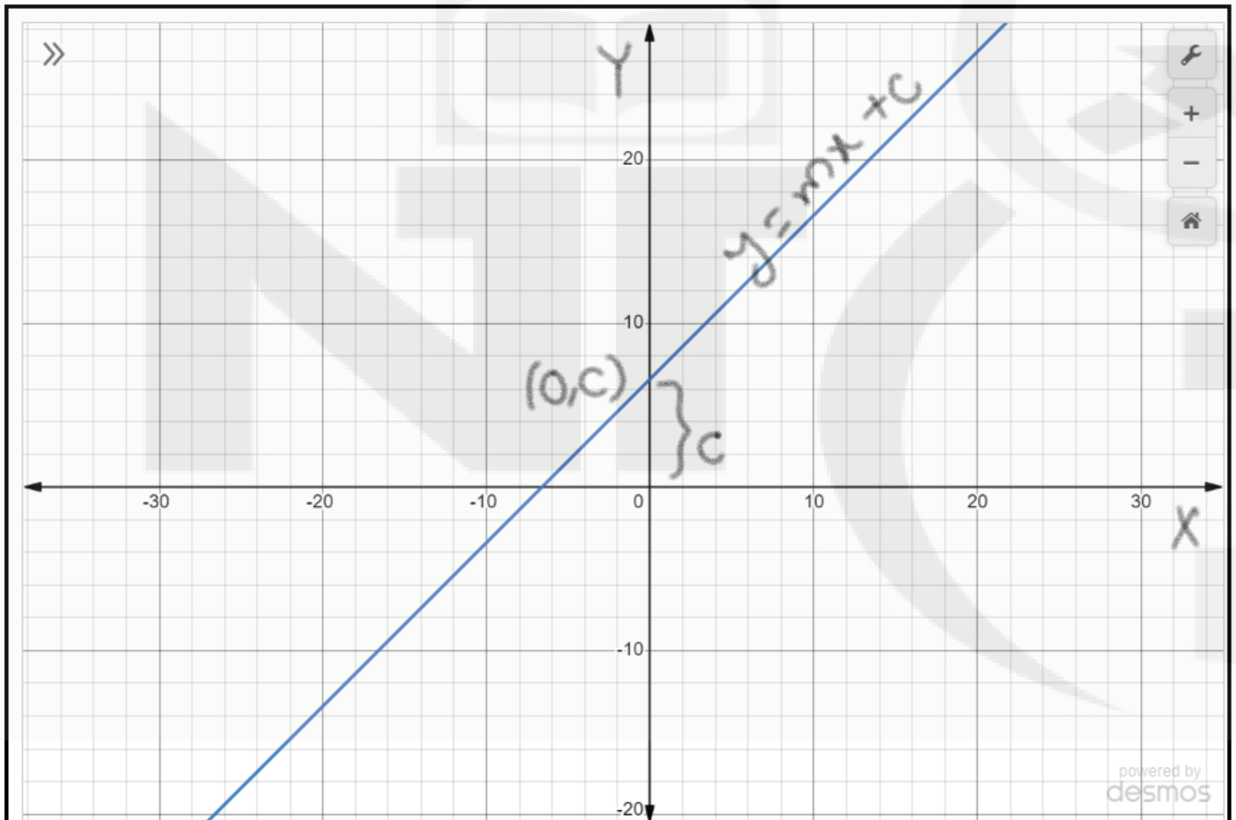
$$\mathbf{y = mx + c}$$

  - **m** is the **slope** of the line, representing the rate of change of y with respect to x.
  - **c** is the **y-intercept**, which is the value of **y** when x=0.

Understanding Slope & Intercept

- https://www.transum.org/Maths/Activity/Graph/Desmos.asp




Y
20
10
0
-10
-20
-30
30
X
y = mx + c
(0,c)
c
powered by
desmos

The equation for a simple linear regression model is:

$$y = mx + b$$

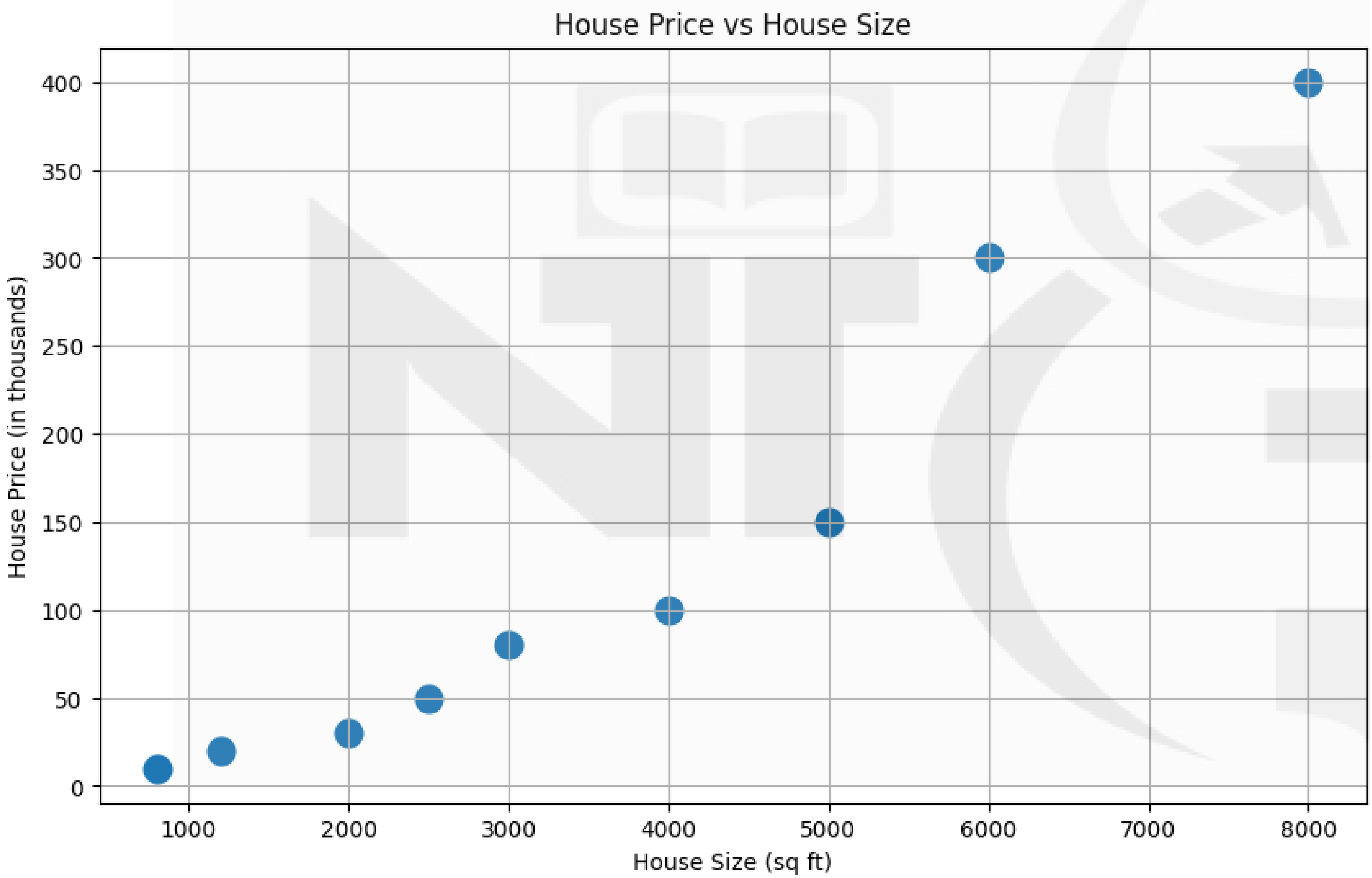
Where:

- y is the predicted value (label).
- m is the slope of the line (coefficient for the feature).
- x is the independent variable (feature).
- b is the y-intercept (constant term).

Simple Linear Regression

HouseSize	HousePrice
800	10
1200	20
2000	30
2500	50
3000	80
4000	100
5000	150
6000	300
8000	400
5500	????

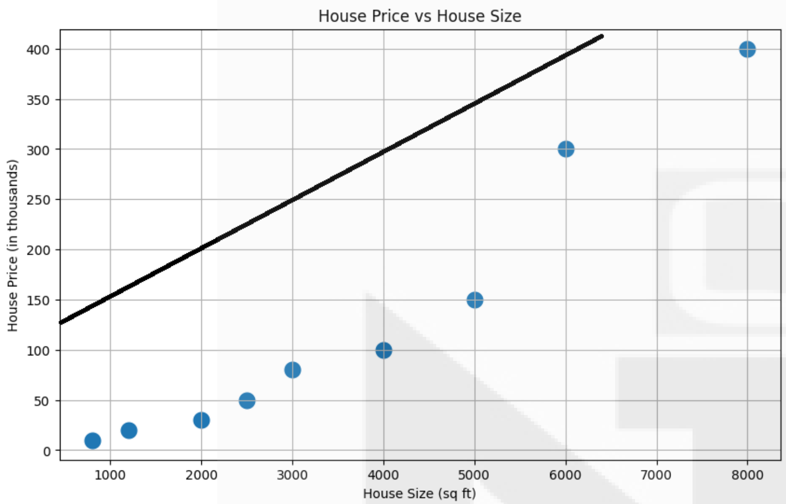
Data points for the house price



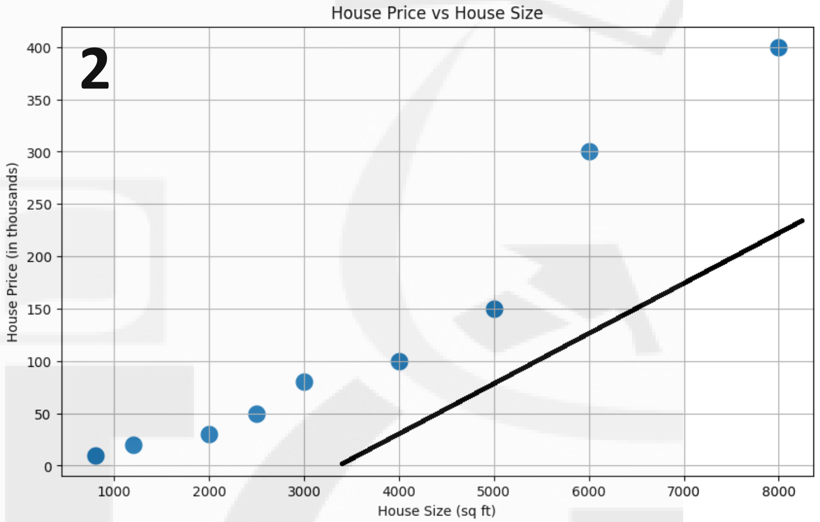
- Simple Linear regression algorithm fits a straight line through the data

Lets say we have 4 different models

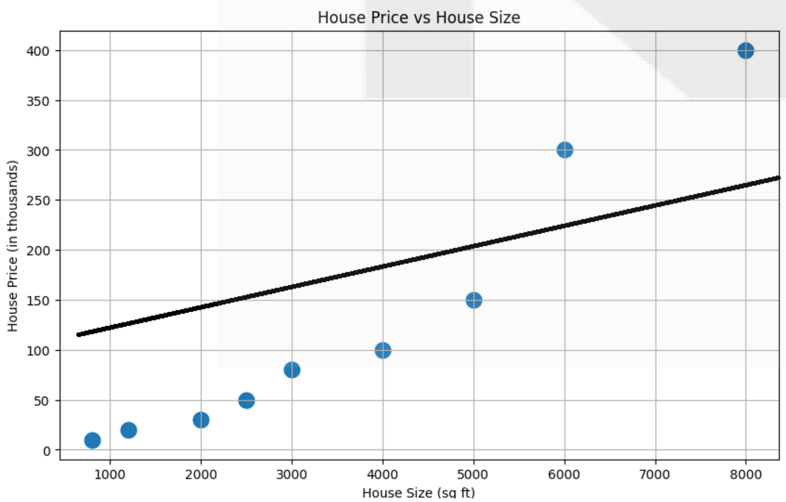
1



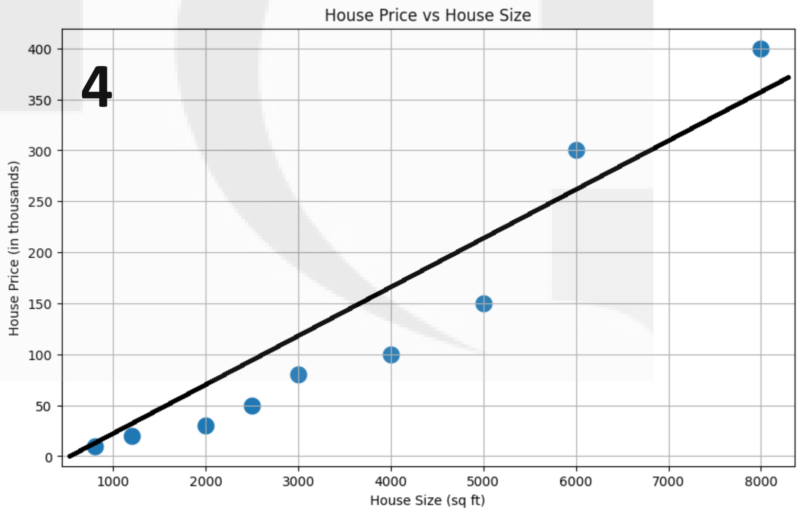
2



3



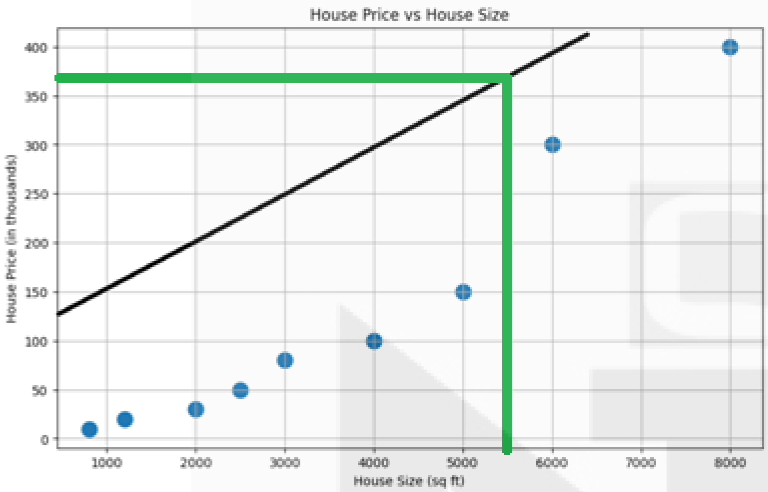
4



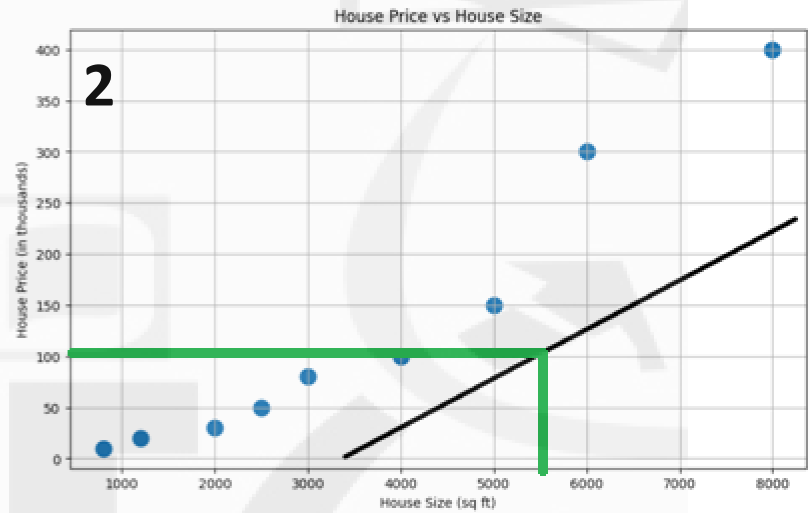
- What the price for housesize = 5500?

HouseSize	HousePrice
800	10
1200	20
2000	30
2500	50
3000	80
4000	100
5000	150
6000	300
8000	400
5500	????

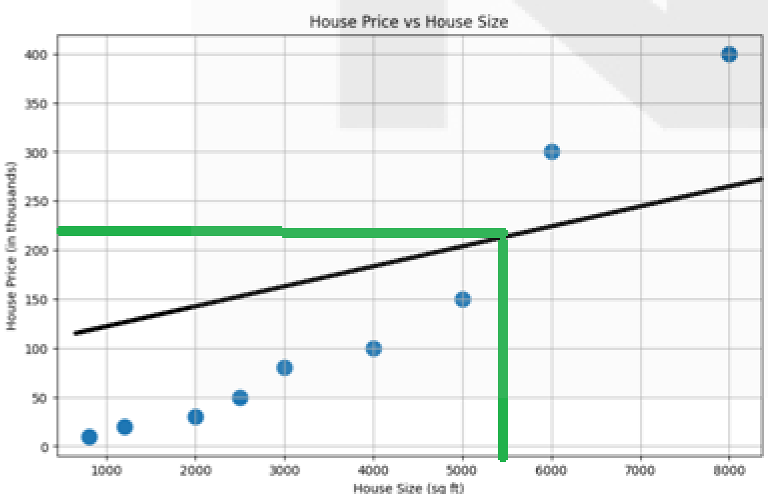
1


House Price vs House Size
House Price (in thousands)
House Size (sq ft)
400
350
300
250
200
150
100
50
0
1000
2000
3000
4000
5000
6000
7000
8000

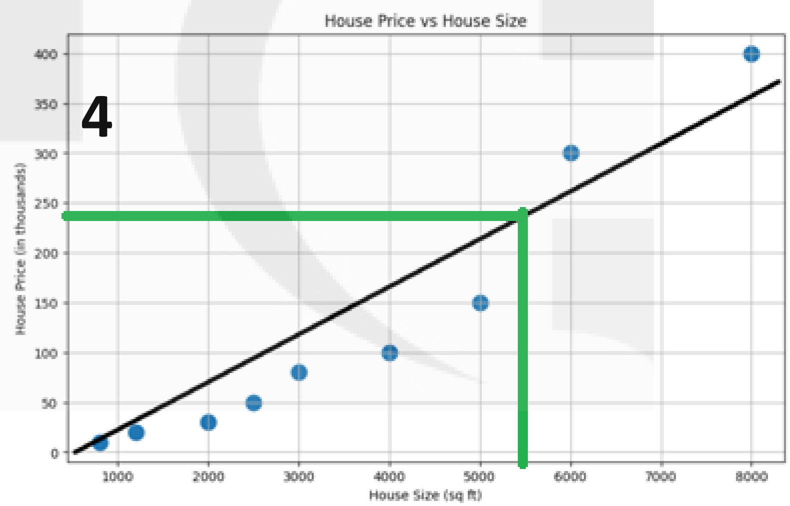
2


House Price vs House Size
House Price (in thousands)
House Size (sq ft)
400
350
300
250
200
150
100
50
0
1000
2000
3000
4000
5000
6000
7000
8000

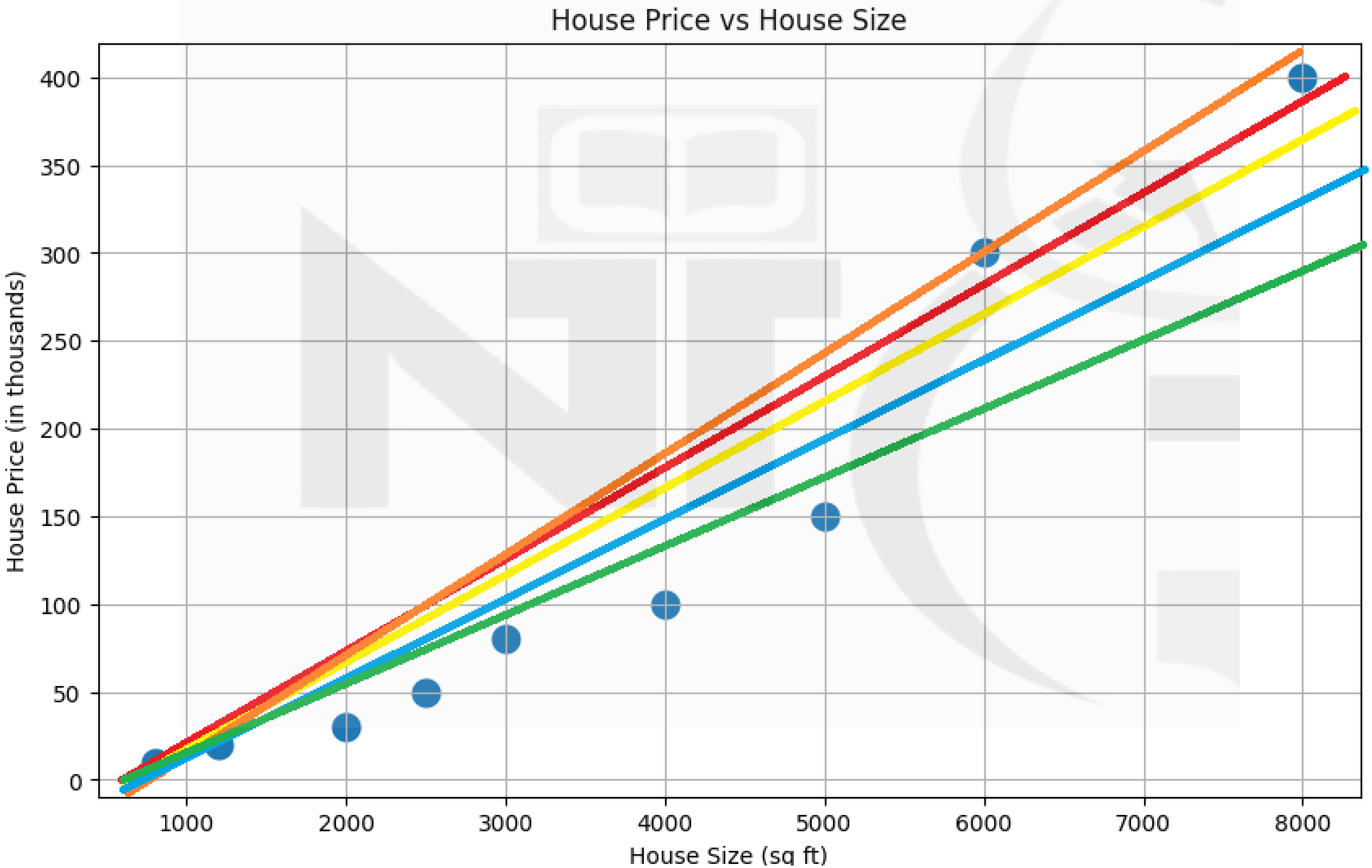
3


House Price vs House Size
House Price (in thousands)
House Size (sq ft)
400
350
300
250
200
150
100
50
0
1000
2000
3000
4000
5000
6000
7000
8000

4

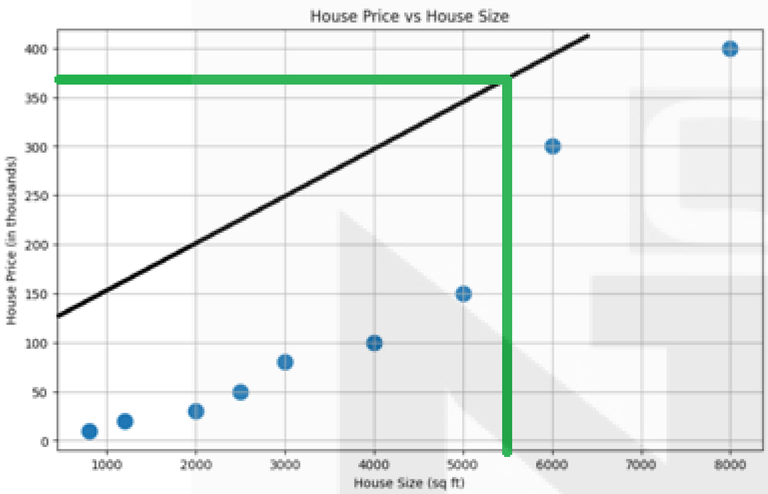

House Price vs House Size
House Price (in thousands)
House Size (sq ft)
400
350
300
250
200
150
100
50
0
1000
2000
3000
4000
5000
6000
7000
8000

Best Fit Line

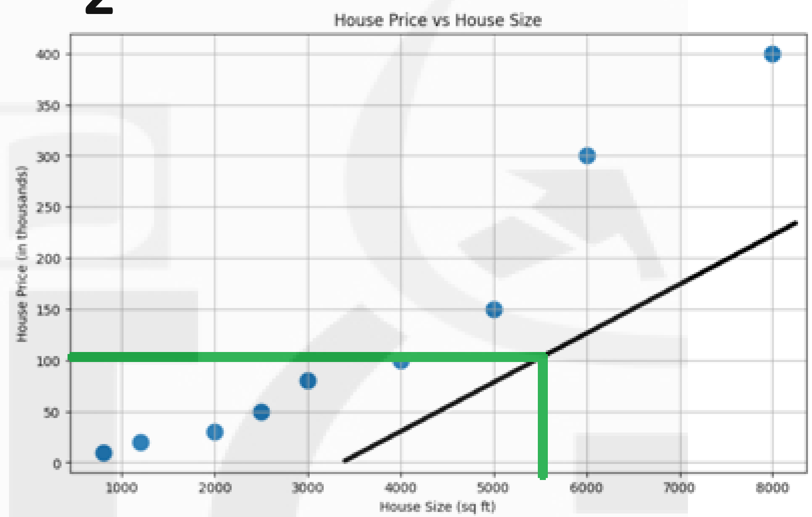


Best Fit line give least error:

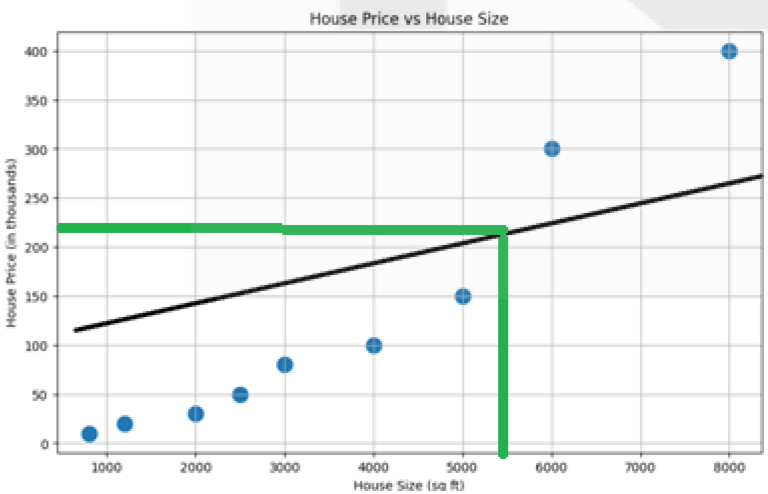
1



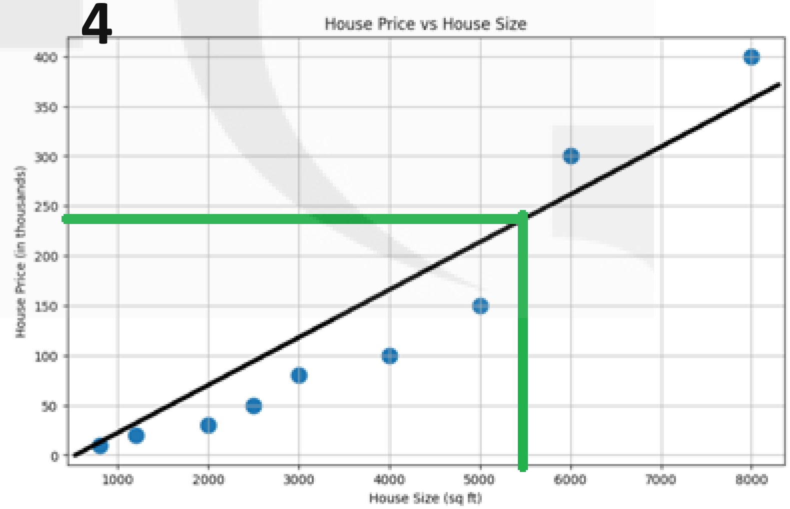
2



3



4



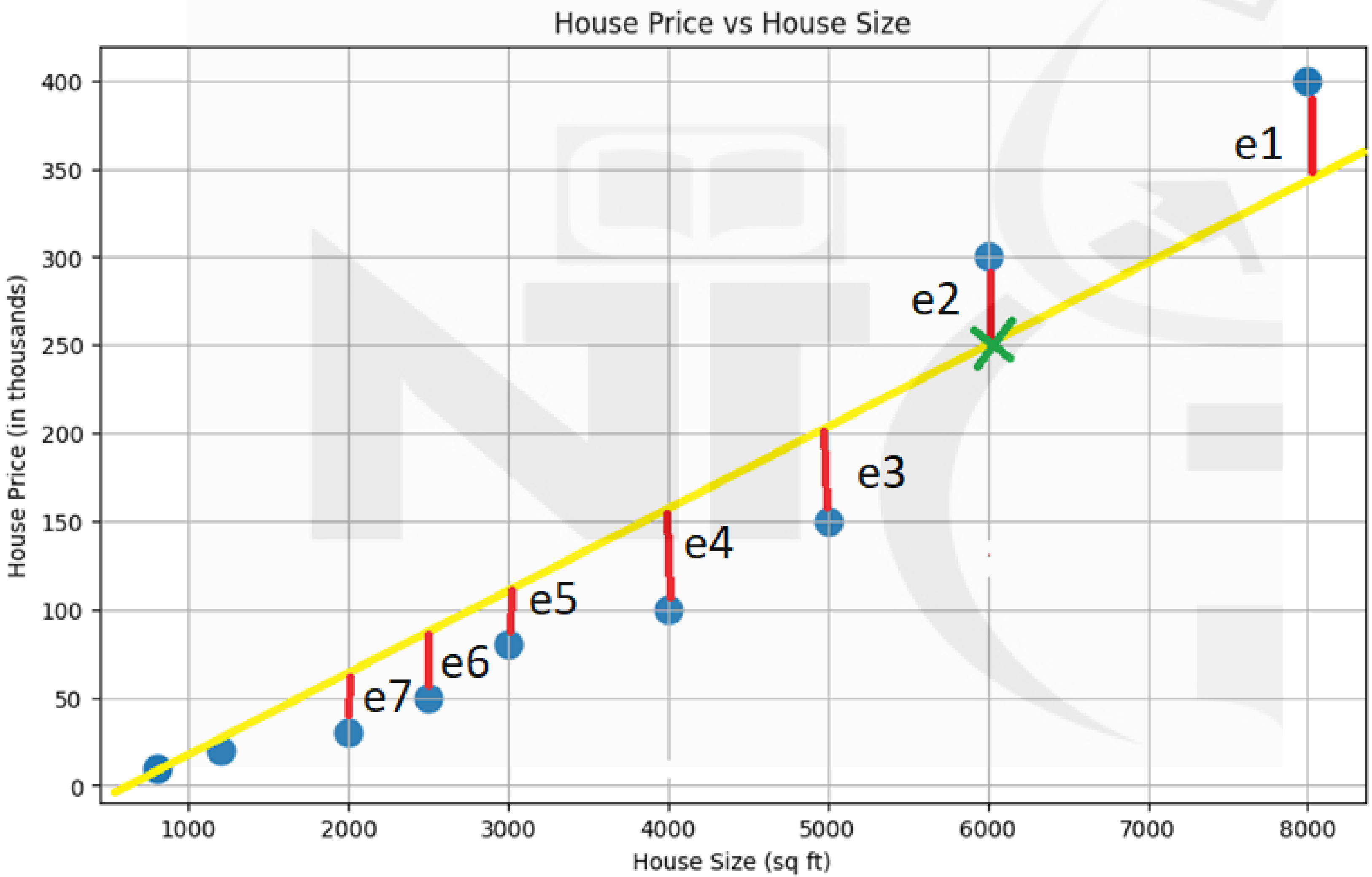
How can we find best fit line?

- By adjusting m and c (explain using plot)

- There are two methods:
  - OLS (Ordinary least squares)
  - Gradient Descent

ORDINARY LEAST SQUARE(OLS)

Best Fit line



Ordinary Least Square(OLS)

- Also known as "Linear least squares", minimizes the sum of the squared differences between the observed values and the predicted value
- Let Yp be the predicted value of Y (the actual value) for a given independent variable value of X

Ypred = mx + c

c = Ypred-mx

$$m = \frac{\sum(x_i - \bar{x})(y_i - \bar{y})}{\sum(x_i - \bar{x})^2}$$

$x = $ *independent variables*

$\bar{x} = $ *average of independent variables*

$y = $ *dependent variables*

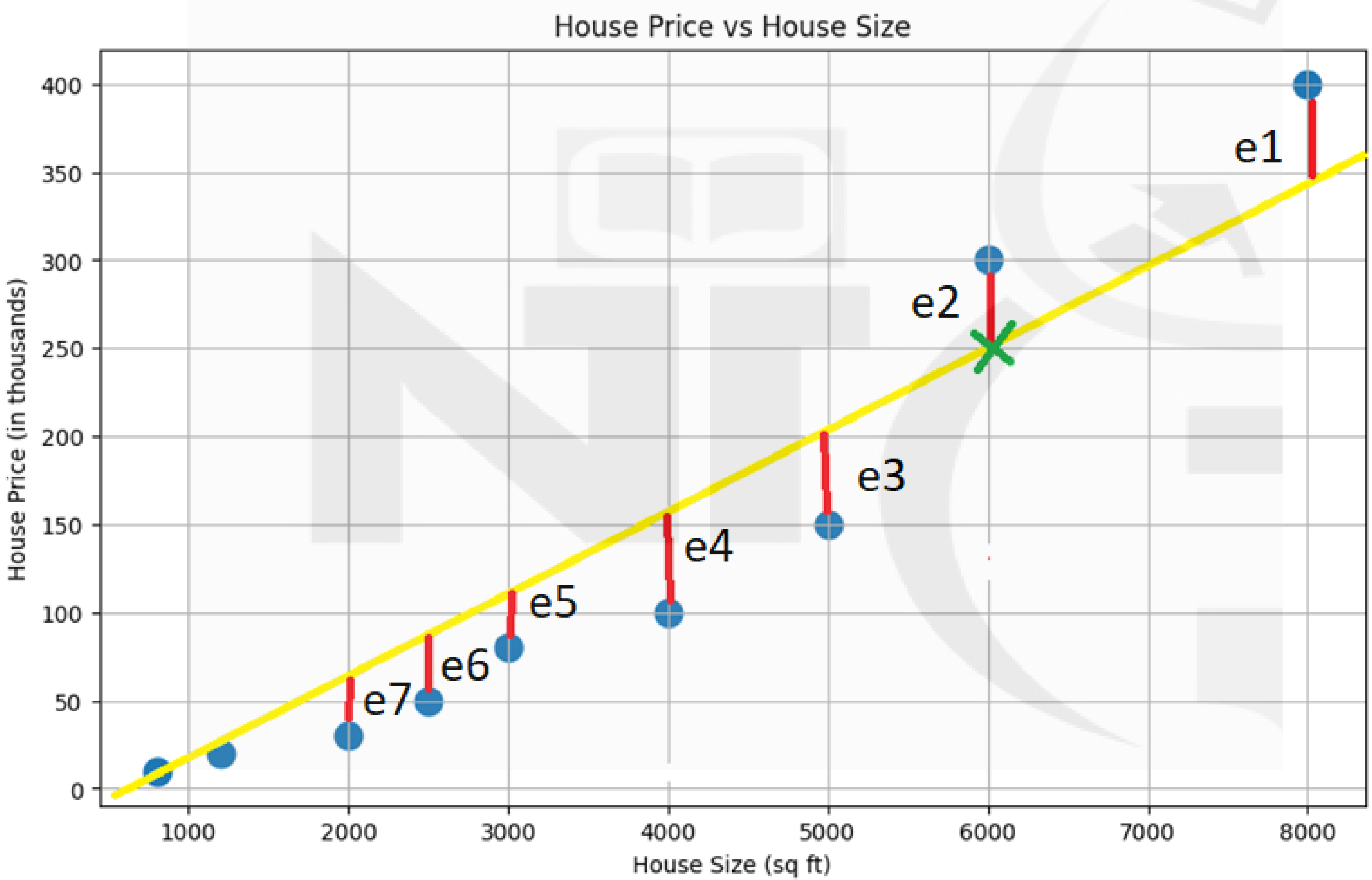
$\bar{y} = $ *average of dependent variables*

- So using OLS , simple Linear Regression is a 2 step process:
  - Find m and c using the formulas
  - Generate the best fit line by substituting the values in **y=mx+c**

Ordinary Least Square(OLS)

- This method is not scalable i.e. with increasing independent variables and increasing data points Hence, we use another method called "Gradient Descent"

Best Fit line



HOW OLS FORMULAS ARE DERIVED

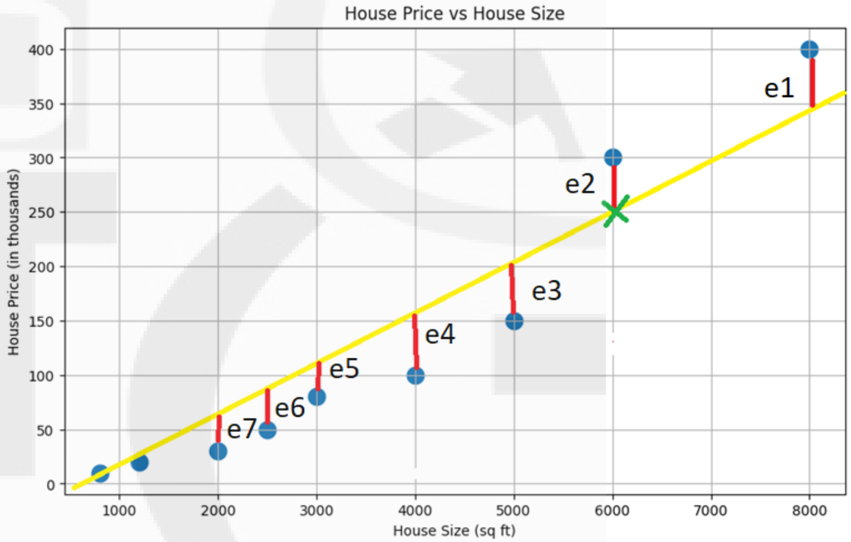
Pre-requisite

- First understand :
  - Derivative
  - Partial Derivative

How OLS formulas are derived

$$E = e_1^2 + e_2^2 + e_3^2 + \ldots + e_n^2$$

$E = \sum_{i=1}^{n} e_i^2$ ← Error function



Notations for Predicted Value

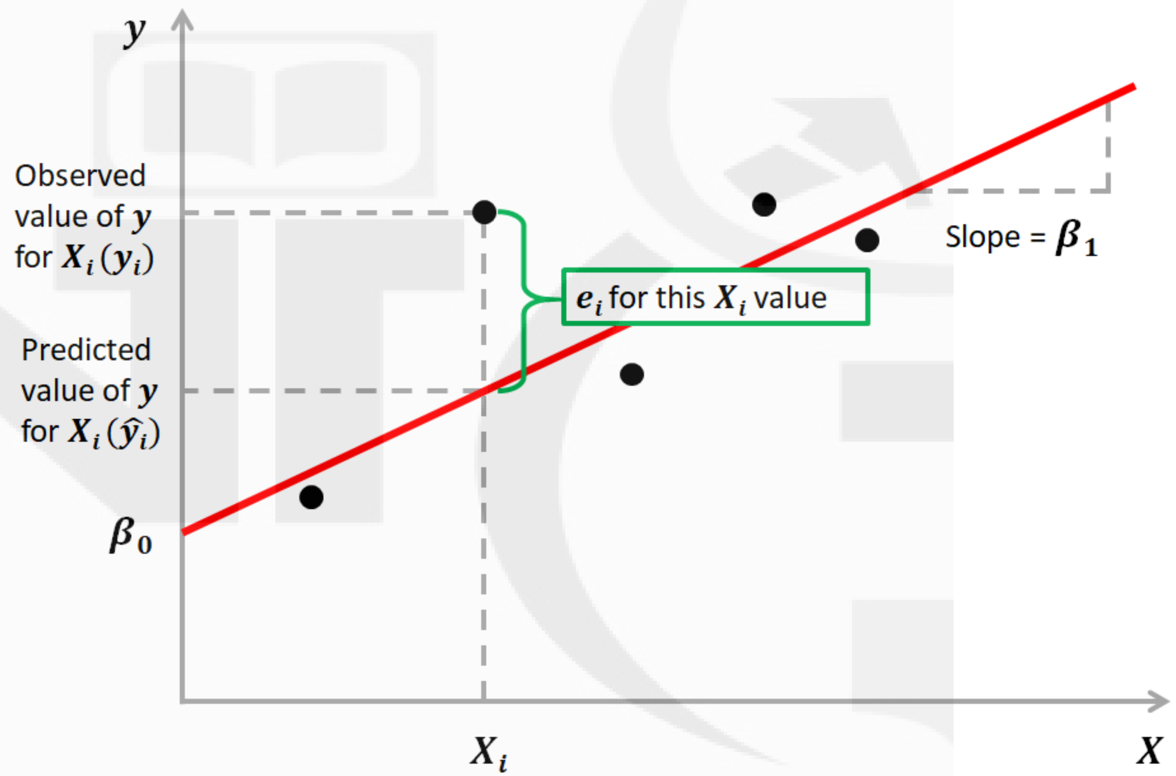
There are a few different notations used to represent the predicted values of the dependent variable (y) in linear regression:

1. $\hat{y}$ (pronounced "y-hat"): This is the most common notation used to represent the predicted values of y. The "hat" symbol indicates that it is an estimate or prediction of the true y value.
2. y^E: Some authors use this notation, where the superscript "E" stands for "expected value". It represents the expected or predicted value of y given the values of the independent variables.
3. $\widehat{y}$: This is similar to $\hat{y}$, but uses an underline instead of a hat. It still represents the predicted value of y.
4. $\widehat{Y}$: When y is a random variable, some authors use this notation with an uppercase Y to emphasize that $\widehat{Y}$ is a random variable representing the predicted value of y.
5. $\widehat{y}_i$: This notation specifies the predicted value of y for the i-th observation in the dataset. The subscript i indexes the individual observations.

The most common and widely used notation is $\hat{y}$. It clearly indicates that it is an estimate or prediction of the dependent variable y based on the linear regression model and the observed values of the independent variables.

What is error:

$$\text{Error} = y_i - \hat{y}_i$$



Error can be rewritten as:

$$e_i = (y_i - \hat{y}_i)^2$$

Where:

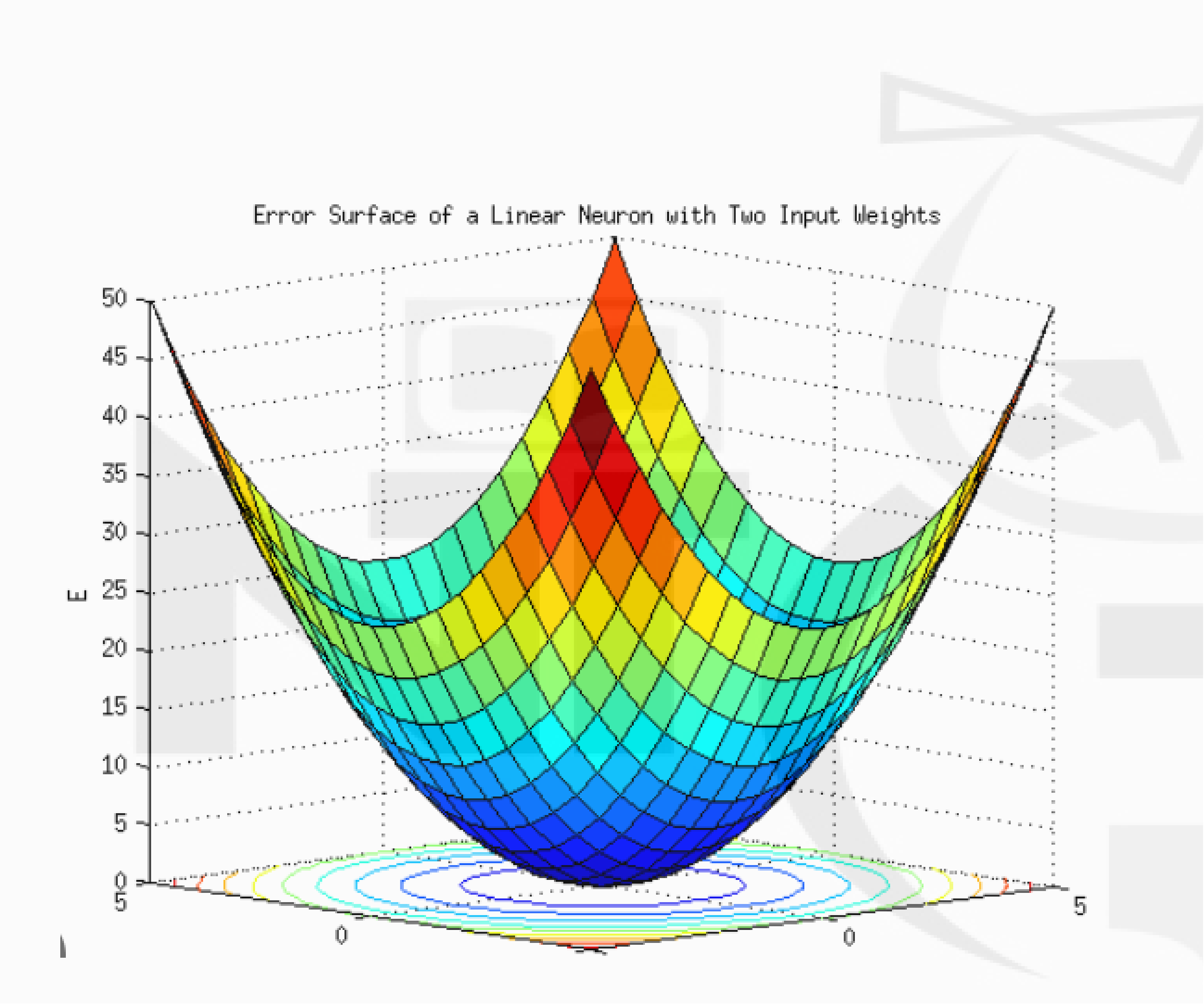
- e_i is the squared error for the i-th data point,
- y_i is the observed value (actual value),
- $\hat{y}_i$ is the predicted value (estimated value) from the model.

After substituting error:

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

Where:

- E is the total error (sum of squared errors),
- n is the total number of data points,
- y_i is the observed value for the i-th data point,
- $\hat{y}_i$ is the predicted value for the i-th data point.


Error Surface of a Linear Neuron with Two Input Weights
E
0
5
10
15
20
25
30
35
40
45
50
5
0
0
5

Reframe in terms of m and b

Reframe this in terms of m, b

Predicted output $\hat{y}_i = mx_i + b$

$$E(m,b) = \sum_{i=1}^{n} (y_i - mx_i - b)^2$$

Only m, b are variables in above equation

After substituting error value:-

$$e_i = (y_i - \hat{y}_i)^2$$

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

Reframe this in terms of m, b

Predicted output $\hat{y}_i = mx_i + b$

$$E(m,b) = \sum_{i=1}^{n} (y_i - mx_i - b)^2$$

Only m, b are variables in above equation

Partial Diff w.r.t b

- To find the optimum values of m and b we need to get to the lowest point in the parabola where slope is zero.

$$\frac{\partial E}{\partial b} = \frac{\partial E}{\partial b} \sum_{i=1}^{n} (y_i - mx_i - b)^2 = 0$$

$$\Rightarrow \sum \frac{\partial}{\partial b} (y_i - mx_i - b)^2 = 0$$

$$\Rightarrow \sum 2(y_i - mx_i - b) = 0$$

$$\Rightarrow \sum (y_i - mx_i - b) = 0$$

$$\Rightarrow \frac{\sum y_i}{n} - \frac{\sum mx_i}{n} - \frac{\sum b}{n} = \frac{0}{n}$$

$$\Rightarrow \bar{y} - m\bar{x} - b = 0$$

$$\boxed{b = \bar{y} - m\bar{x}}$$

Partial Diff w.r.t m

$$E = \sum (y_i - mx_i - \bar{y} + m\bar{x})^2$$

$$\frac{\partial E}{\partial m} = \sum \frac{\partial}{\partial m} (y_i - mx_i - \bar{y} + m\bar{x})^2 = 0$$

$$\sum 2(y_i - mx_i - \bar{y} + m\bar{x}) \cdot (-x_i + \bar{x}) = 0$$

$$\sum -2(y_i - mx_i - \bar{y} + m\bar{x})(x_i - \bar{x}) = 0$$

$$\sum (y_i - mx_i - \bar{y} + m\bar{x})(x_i - \bar{x}) = 0$$

$$\sum \left[(y_i - \bar{y}) - m(x_1 - \bar{x}_1)\right](x_i - \bar{x}) = 0$$

$$\sum (y_i - \bar{y})(x_i - \bar{x}) = m \sum (x_1 - \bar{x})^2$$

$$m = \frac{\sum_{i=1}^{n} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{n} (x_i - \bar{x})^2}$$

COST/LOSS/ERROR FUNCTION

Purpose of Cost Function

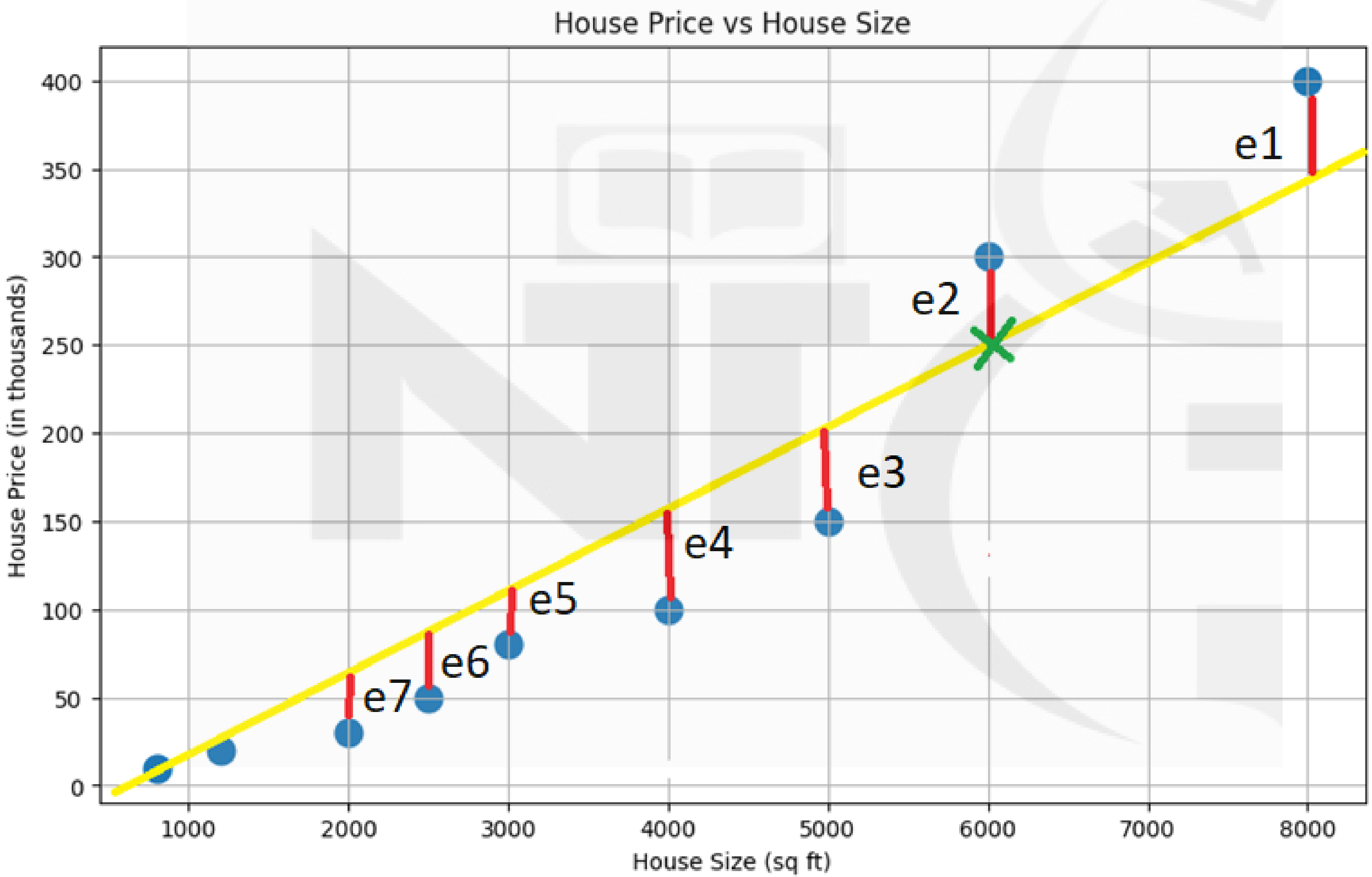
- The primary objective of a cost function is to minimize the errors in predictions.
- By calculating the difference between the predicted output and the actual output, the cost function provides a single numerical value that reflects the model's accuracy.
- This value is essential for guiding the optimization process during model training, allowing the algorithm to adjust its parameters iteratively to improve accuracy.

Notations for Cost Function

1. $J(\theta)$: This is a widely used notation for the cost function, where θ represents the parameters of the model. The function J indicates the cost associated with those parameters.
2. $L(y, \hat{y})$: In this notation, L denotes the loss function, which measures the error between the actual value y and the predicted value $\hat{y}$. This notation emphasizes the relationship between the true and predicted values.
3. C: Some sources may simply denote the cost function as C, representing the overall cost without specifying the parameters or the nature of the loss.

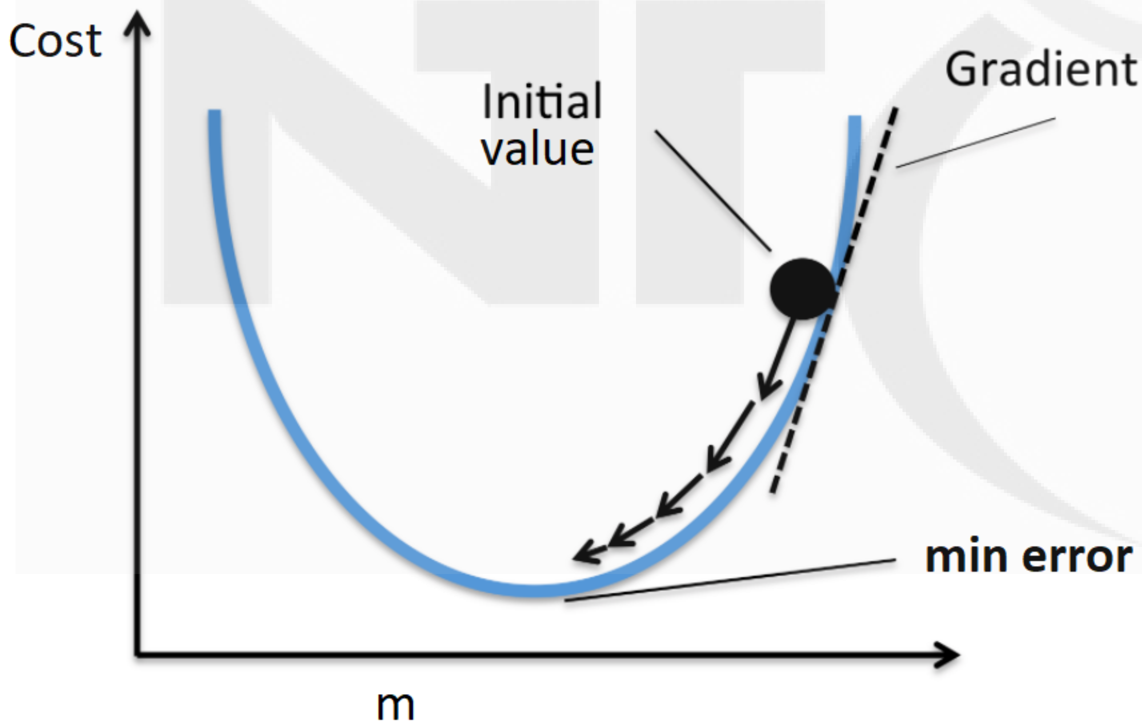
GRADIENT DESCENT METHOD

Best Fit line

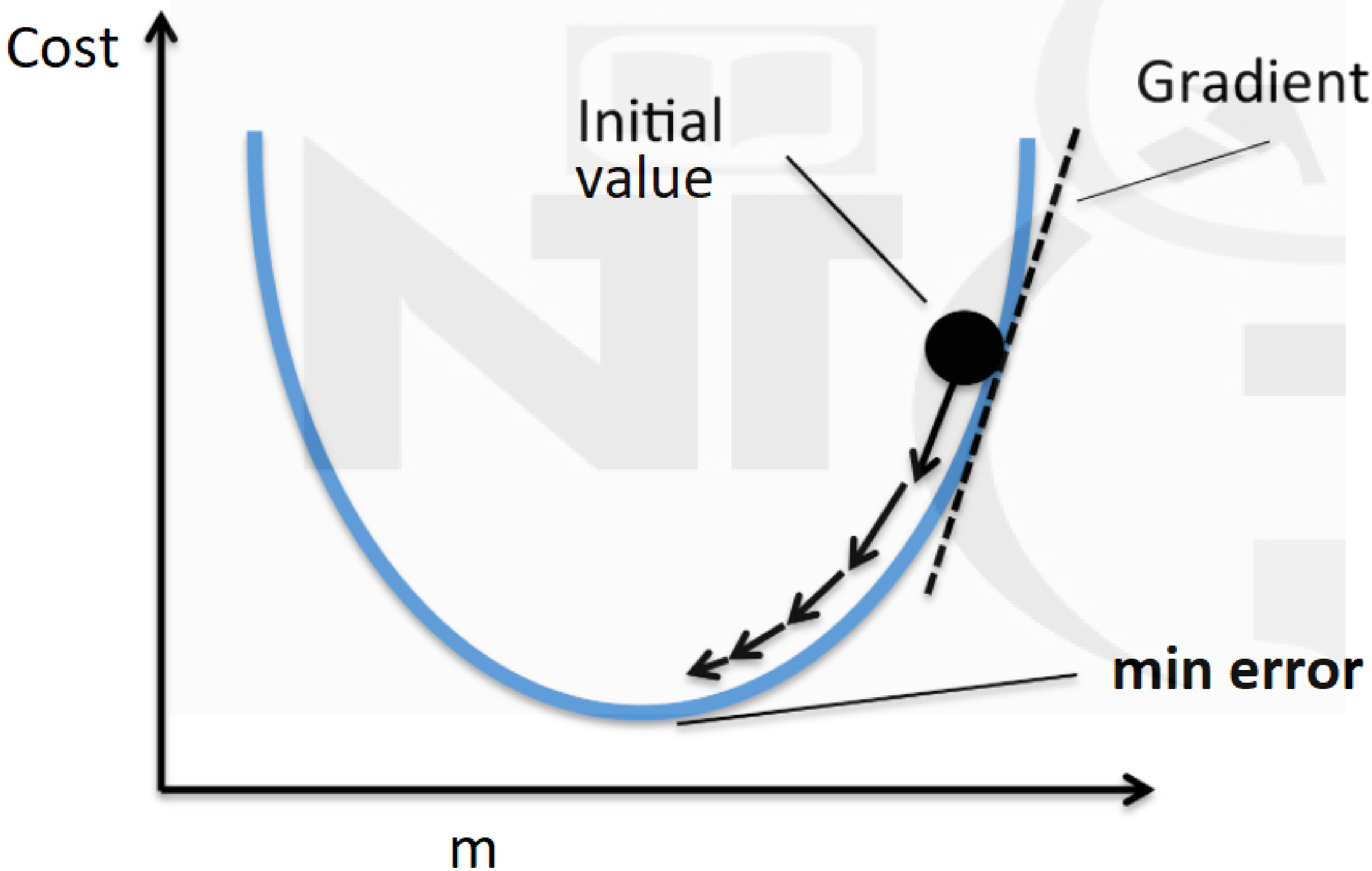


Understanding shape of Error Function

- https://www.transum.org/Maths/Activity/Graph/Desmos.asp



Cost Vs slope



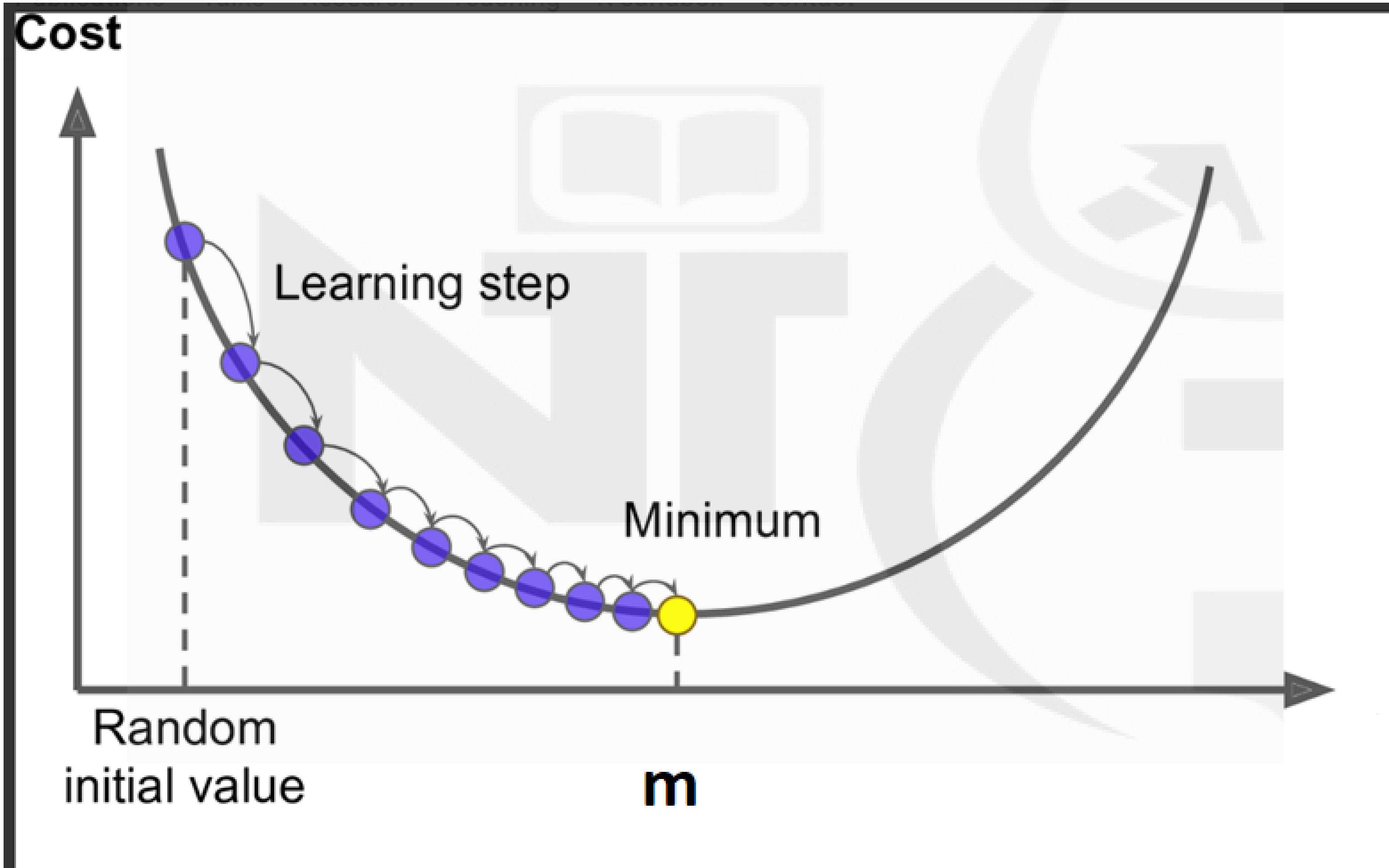
Relation between two variables

- In a **linear relationship**, two variables, say x and y, have a relationship that can be described by a straight line when plotted on a graph. The general form of a linear equation is:

$$\mathbf{y = mx + c}$$

  - **m** is the **slope** of the line, representing the rate of change of y with respect to x.
  - **c** is the **y-intercept**, which is the value of **y** when x=0.

Cost/Error Vs Slope



Gradient Descent

Gradient Descent is an iterative process that adjusts m and c to minimize the cost function $J(m, c)$.

Steps:

1. **Initialize** m and c with random values.
2. **Calculate the gradient** of the cost function with respect to both m and c:

$$\frac{\partial J(m,c)}{\partial m} = \frac{1}{n}\sum_{i=1}^{n}(y_{\text{pred}}^{(i)} - y^{(i)}) \cdot x^{(i)}$$

$$\frac{\partial J(m,c)}{\partial c} = \frac{1}{n}\sum_{i=1}^{n}(y_{\text{pred}}^{(i)} - y^{(i)})$$

3. **Update the parameters** m and c using the gradients:

$$m := m - \alpha \cdot \frac{\partial J(m, c)}{\partial m}$$

$$c := c - \alpha \cdot \frac{\partial J(m, c)}{\partial c}$$

   Here, α is the learning rate that controls the step size in each iteration.

4. **Repeat** the process until convergence, where the changes in m and c become negligible, indicating that the cost function has reached its minimum.

LEARNING RATE

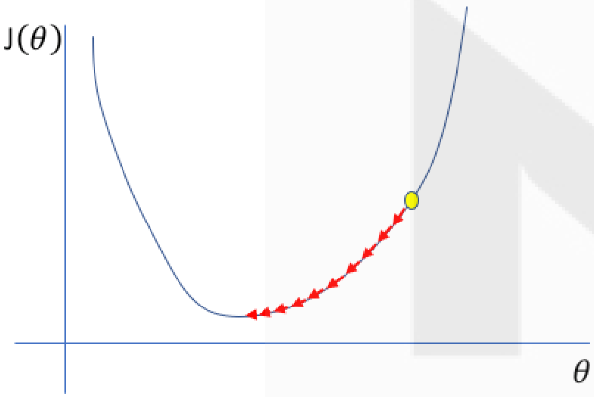
Learning Rate

The learning rate α is critical:

- If α is too small, convergence will be slow.
- If α is too large, the algorithm may overshoot the minimum, failing to converge.

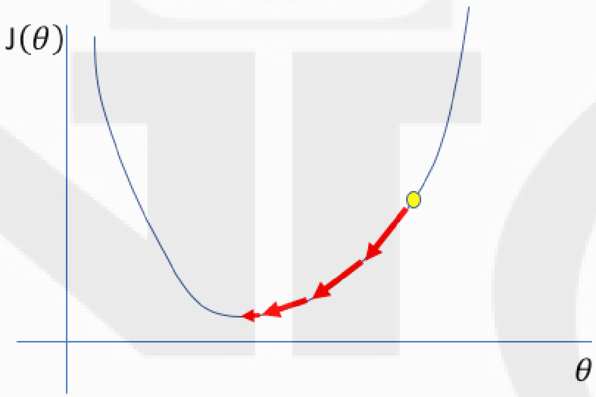
Learning Rate

Too low



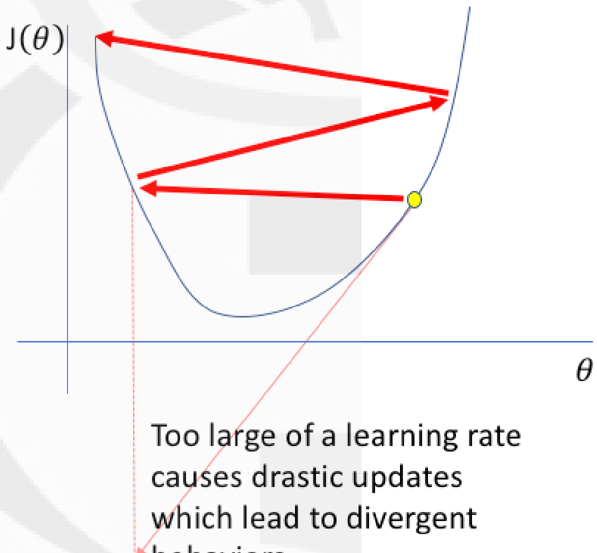
A small learning rate requires many updates before reaching the minimum point

Just right



The optimal learning rate swiftly reaches the minimum point

Too high



Too large of a learning rate causes drastic updates which lead to divergent behaviors

Gradient Descent method

1. In this method the error is represented in squared form i.e. $E = (Y_{expected} - Y_{pred})^2$
2. Expanding Ypred, $E = ((Y_{expected} - (mX + C))^2$
3. Thus, E is a function of m and c given $Y_{expected}$ and X come from data
4. The E function being quadratic (raised to power of 2) when plotted against m and c, will acquire a parabolic shape
5. This guarantees an absolute minima i.e. there will be a unique combination of m and c which will deliver the least error. Let his be the best m and best c
6. Starting from some random m and c, the Gradient Descent method will automatically discover the best m and best c using a mathematical technique called "Partial Derivatives"
7. This method can be applied with any number of independent variables. It will be faster than the algebraic method

Multivariate Linear Regression

1. When more than two predictor variables are used to predict the value in the dependent variable
2. The structure of the model remains same but gets extended to include all the variables instead of just one as in simple linear regression
   a. $Y = m_1X_1 + m_2X_2 + ... + m_nX_n + c + e$
3. Geometrically, the line in simple linear regression model is replaced with a plane (for two predictor variables) and by a hyper plane (planes in higher than three dimensions) to express the relationship between dependent and independent variables
4. The predictor variables are expected to be independent of one another i.e not correlate amongst themselves

LR Advantages/Disadvantages

Advantages –

1. Simple to implement and easier to interpret the outputs coefficients

Disadvantages -

1. Assumes a linear relationships between dependent and independent variables. That is, it assumes there is a straight-line relationship between them
2. Outliers can have huge effects on the regression
3. Linear regression assume independence between attributes
4. Linear regression assumes constant variance of residuals (homoscedasticity).If variance changes (heteroscedasticity), predictions become biased.

Polynomial Regression

Polynomial Regression

- Polynomial Regression is an extension of **Linear Regression** where the relationship between the independent variable (X) and the dependent variable (Y) is **modeled as an n-degree polynomial** rather than a straight line.

- **Key Idea**: Instead of fitting a **straight line,** Polynomial Regression fits a **curved line** to capture non-linear patterns in the data.

Advantages of Polynomial Regression

- **Captures Non-Linearity** → Models curved relationships between X and Y.
- **Better Fit for Some Data** → Works well when data isn’t perfectly linear.
- **More Flexible Than Linear Regression** → Can adjust degree (n) to improve accuracy.

Disadvantages of Polynomial Regression

- **Overfitting Risk** → High-degree polynomials (n too large) may fit noise instead of patterns.
- **Less Interpretable** → Unlike linear regression, coefficients in polynomial regression are harder to interpret.
- **Sensitive to Outliers** → Since polynomial terms amplify values, outliers can distort the curve.